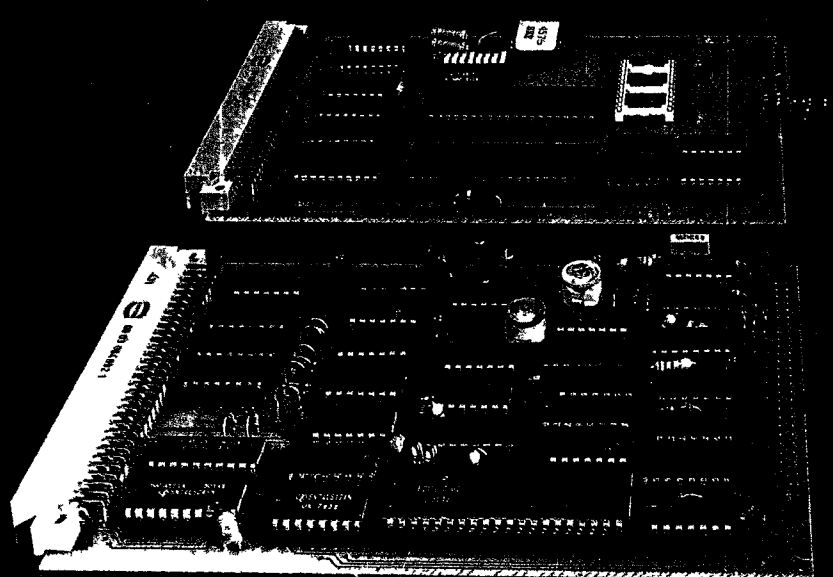
DAS CP/M-SONDERHEFT.

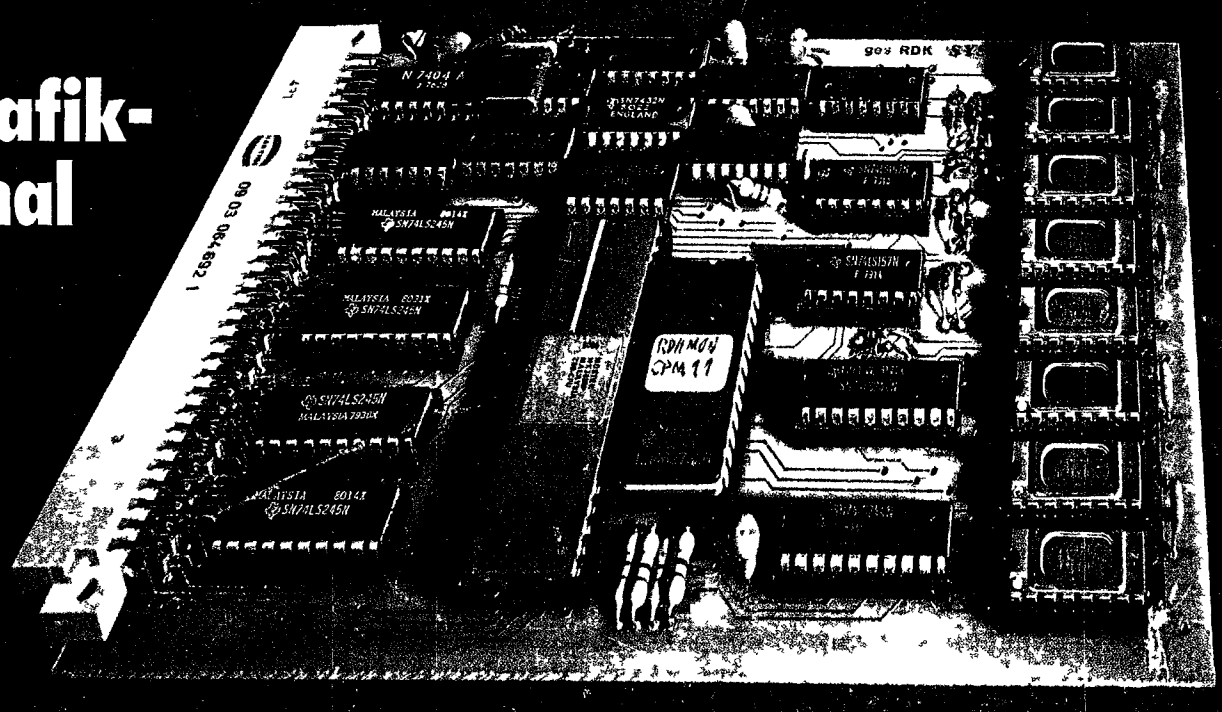
Sonderheft Nr. 81
Preis 26 DM,
198 öS, 26 sfr.
mc

Der mc-CP/M-Computer

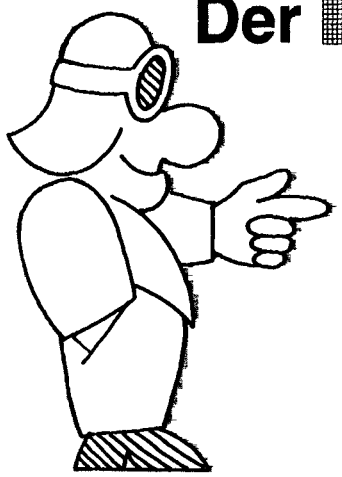
Das mc-Text-verarbeitungs-Terminal

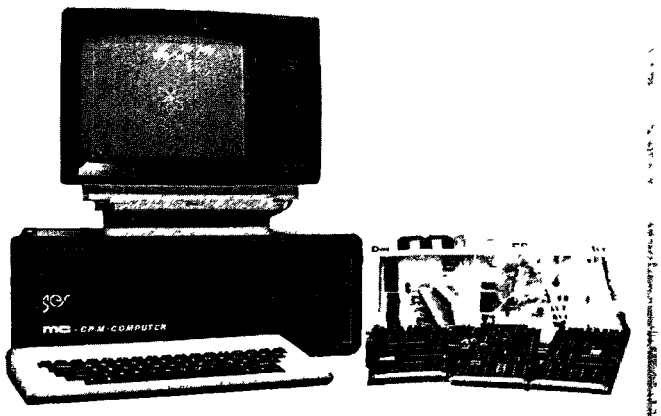
Das mc-Grafik-Terminal





Der mc -CP/M-COMPUTER





Der mc-CP/M⁺-Computer: Fertiggerät und Platinen

GES liefert: Platinen*, Bausätze, Fertiggeräte, Zubehör. Zu gesenkten Preisen!

SYS1

- CPU, 64 K RAM, 4 K EPROM
- Bus: ECB und frei wählbar
- mc September 82
- Bootstrap-Logik für CP/M

Platine + Handbuch	69.–
Komplettbausatz	398.–
Fertiggerät	498.–
Nur Handbuch	20.–
Monitor-Eprom	39.–

FLO1

- Floppy-Disk-Controller
- Alle gängigen Laufwerke
- Controller WD 1797
- Single + double Density
- 5¼" oder 8" (single d.)

Platine + Handbuch	69.–
Komplettbausatz	398.–
Fertiggerät	498.–
Nur Handbuch	20.–
CP/M V 2.2, 5¼" oder 8"	398.–
WD 1797	49.50

- Laufwerk SHUGART SA 860 8" Slime-Line, DS 1798.–
- Laufwerk SHUGART SA 200 5¼" 598.–
- NEU: LW Panasonic 3 Zoll, kompatibel zu 5¼", 2 × 160K .. 598.–
- 10er-Pack 3"-Disketten 199.50
- CP/M-Betriebssystem, komplett mit Dokumentation, auf 3", 5¼" oder 8" 398.–

*Original-mc-CP/M-Computer-Platinen erhalten Sie **nur von uns.** Wir haben das Layout erstellt und können dadurch in Industriequalität liefern.
⁺CP/M ist ein eingetragenes Warenzeichen von Digital Research.

Alle Baugruppen sind Europakarten bzw. als Europakarten trennbar. Alle Preise in DM **inklusive** Mehrwertsteuer ab Kempten. Angebote freibleibend. Umfangreiche Info **kostenlos**. Schutzgebühr für Handbücher, wird bei späterer Bestellung gutgeschrieben. Alle Bausätze nur mit **Markenhalbleitern**, alle Platinen Industriequalität, durchkontaktiert und Lötstopplack.

OUT1

- Serielle und parallele Ausgänge
- 2 V24, 20 parallel
- 2 Baudrate-Generatoren
- Voll gepuffert

Platine + Handbuch	65.–
Komplettbausatz	298.–
Fertiggerät	398.–
Nur Handbuch	20.–

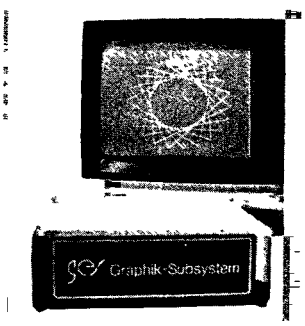
NEU! TERM1

- Terminal + Graphik (256 × 512)
- 2 Prozessoren (Z80 + GDP 9366)
- 64-K + 4-K-RAM, 8-K-ROM
- 4 Bildseiten, umschaltbar
- Intelligente Graphik

Nur Handbuch	30.–
Platine, HB, 8 K ROM	149.–
Komplettbausatz	698.–
Fertiggerät	950.–
GDP 9366	**99.50**

NEU! GSS

Graphik-Subsystem: Dies ist TERM1, komplett, geprüft, mit Netzteil im Gehäuse. Hochauflösende Graphik für jeden Rechner mit V24-Schnittstelle. Betriebsarten: Graphik, Text, Logo, Tektronix-Teilmenge
Preis (ohne Monitor) DM 1298.–



Weiteres Zubehör sowie unsere aktuellen Preise finden Sie in unserer kostenlosen Info. Anruf oder Postkarte genügt.

Und wenn's nicht klappt? Unsere Platinen und Bausätze eignen sich nicht für Anfänger. Falls auch der Profi Schwierigkeiten mit der Inbetriebnahme hat, können wir zu sehr niedrigen Pauschalpreisen reparieren: Bausatz von uns: DM 57.– + Materialkosten; Platine von uns: DM 114.– + Materialkosten. Diese Pausch.-Reparatur kann auch abgelehnt werden. Nicht von uns gelieferte Geräte werden nicht unterstützt.

Wir stellen aus: Systems München, Hobby-Elektronik '83 Stuttgart, Hobbytronic Februar '84 Dortmund

GRAF ELEKTRONIK SYSTEME GMBH
Magnusstraße 13, 8960 Kempten,
Telefon 08 31/6 19 30, Teletex: 831 804 = Graf

TA TRIUMPH-ADLER
sirius COMPUTER
EPSON

Vorwort

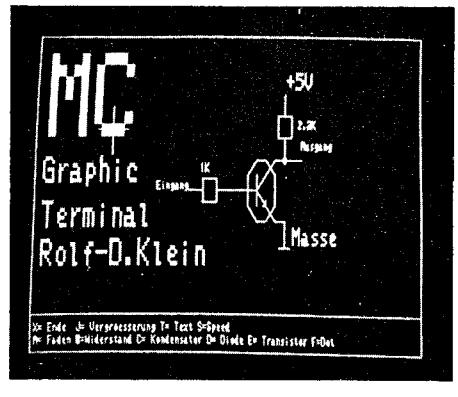
Spektakuläres gibt es in der Computerszene genug. Kritisch wird der Einsatz spektakulärer Systeme immer dann, wenn praktikable Leistungen verlangt werden. Deshalb ist Altbewährtes überall dort zu finden, wo Zuverlässigkeit von Soft- und Hardware verlangt wird. Außerdem ist die Berechenbarkeit im Verhalten eines Systems, also die Bekanntheit aller guten und schlechten Eigenschaften, aller Fehler und Vorzüge, das Wichtigste, wenn es um die Programmierung wirksamer Programme geht. Wir sind deshalb stolz, daß wir mit dem mc-CP/M-Computer ein bewährtes Konzept mit bewährtem Betriebssystem zum Selbstbau anbieten können. Spektakulär ist daran, daß hier zum ersten Mal ein geschlossenes Selbstbausystem mit Floppy-Disk-Anschluß präsentiert wird. Daß sozusagen ein Computer bis zum bitteren (wir hoffen aber: freudigen) Ende aufgebaut werden kann. Da zu einer solchen spektakulären Aktion viel Information gehört – und zwar nicht nur eine reine Bauanleitung, sondern auch das Drum-Herum –, haben wir alles, was zum Bau und ersten Betrieb des mc-CP/M-Computersystems gehört, in einem Sonderheft zusammengefaßt. Dieses Heft ist für all die vielen Professionals gedacht, die unseren Computer schon nach der Artikelserie im Heft nachgebaut haben und nun nach kompaktem Informationsmaterial „mit Allem drin" verlangen, als auch für die, die sich jetzt erst zum Nachbau des in Deutschland bewährtesten Selbstbausystems entscheiden, aber vor allem auch für die, die nur mal hineinriechen wollen in die klare Luft der Bits und Bytes, die die Funktion eines Computers in Hard- und Software kennenlernen wollen, bis auf den Grund.

Ihr

Rolf-D. Klein

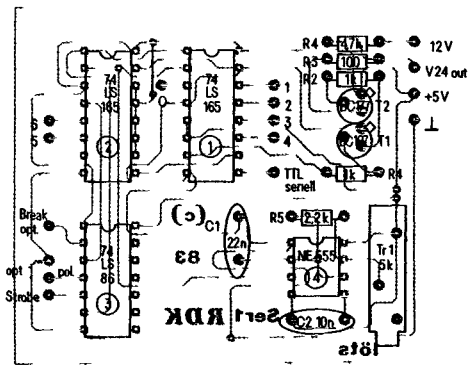
Der mc-CP/M-Computer

Ursprünglich gedacht als Platinensystem aus drei Baugruppen für den Profi, hat sich unser Computer selbständig gemacht, weil ihn viele nachbauen wollten. Es entstand also wegen der großen Nachfrage Schritt für Schritt ein rundum vollständiger Computer. Mit der CPU-Platine geht es los. **Seite 7**



Grafik-Terminal

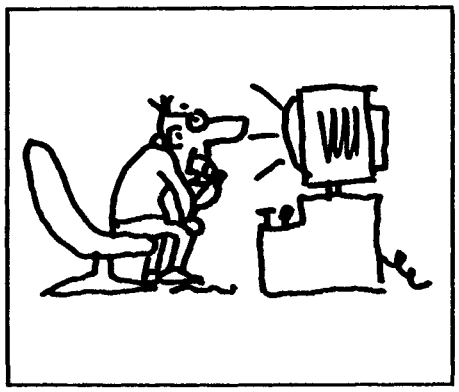
Sofort nach Veröffentlichung der drei Grundplatinen wurden wir bestürmt, auch ein Terminal zu bringen. Jetzt gibt es zwei zur Auswahl. Das alphanumerische mc-Terminal und das mc-Grafik-Terminal **Seiten 51, 73**



Kleinigkeiten

Unser mc-Grafik-Terminal ist auf seriellen Betrieb ausgelegt. Wer daran eine parallel arbeitende Tastatur anschließen möchte, der findet eine kleine Platine zur Lösung dieses Problems. **Seite 70**

Probleme können auch beim Anschluß eines Druckers oder eines „Fremd-Terminals" entstehen. In kleinen Beiträgen finden Sie einiges über die Schnittstelle RS-232 und zum Beispiel über den Anschluß von Centronics-kompatiblen Druckern. **Seiten 71, 112**



Besitzerstolz

Der Weg beim Aufbau des mc-CP/M-Computers ist an manchen Stellen steinig, das darf nicht verschwiegen werden. Vor allem die Inbetriebnahme der Floppys ist mit manchen Schwierigkeiten verbunden. Wer es aber geschafft hat, der wird stolz auf sich sein.

Software-Service

Beim Franzis-Software-Service, Postfach 37 01 20, 8000 München 37, Telefon (0 89) 51 17-3 31 können Sie speziell für den mc-CP/M-Computer folgendes bestellen:

CP/M-Listings, ein Sammelband mit den kommentierten Assemblerlistings vom Monitor (Version 3.4), Mini-BIOS, Minifloppy-Routinen und Formatierer, 18 DM.

Das CP/M-Kurzhandbuch mit Kommandoliste, BDOS-Aufrufen und einer Beschreibung des ASM (in Deutsch), 12 DM.

Die Monitorversion auf EPROM 2732A erhalten Sie fertig programmiert für 28,50 DM.

An Disketten – wahlweise 8" IBM SS/SD oder 5¼" ECMA 70, SS/SD – gibt es derzeit: Speichertestprogramme aus mc 12/82 und Sammeldiskette (mc-Editor nach mc 9/82, Kopierprogramm, Basic-Disassembler, Formatierer usw.). Beide kosten je 19,50 DM.

Für Assemblerprogrammierer haben wir den STRUKTA-Präprozessor, ein Programm das eine strukturierte Assemblerquelle mit IF-THEN-ELSE und WHILE-ENDWHILE in ein ganz normales Assembler-Quellprogramm übersetzt. Auf der STRUKTA-Diskette für 85 DM befinden sich auch die Quellen von STRUKTA und ein Handbuch.

Last, not least, gibt es natürlich noch das Betriebssystem CP/M 2.2 auf Diskette, fertig angepaßt an den mc-CP/M-Computer, wahlweise mit deutschem Kurzhandbuch für 338 DM.

Impressum

1983, Franzis-Verlag GmbH, Karlstraße 37–41, D-8000 München 2.
Bearbeitet von der Redaktion der Zeitschrift mc.
Für den Text verantwortlich: Dipl.-Math. Ulrich Rohde.

ISSN 0722-0022. Druck: Franzis-Druck GmbH, München. Printed in Germany. Imprimé en Allemagne. ZV-Art.-Nr. 81041 · F/ZV/484/811c/3'

Rolf-Dieter Klein:

Die CPU-Platine

mc präsentiert hier den klassischen 8-Bit-Computer zum Eigenbau. Auf der zentralen Karte sind eine Z80-CPU und 64 KByte RAM sowie ein 4-KByte-Urlade-EPROM versammelt! Eine Schnittstellenkarte wird serielle und parallele Datentransfers zu einem Terminal und anderen Peripheriegeräten regeln. Eine Floppy-Steuerkarte wird den Anschluß aller gängigen Disk-Laufwerke erlauben. Damit können Sie dann CP/M laden und haben Zugang zu der Welt der CP/M-Software. Aber auch, wer nur mit Monitor, ohne Disk, zufrieden ist, hat herrliche 64 KByte frei zur Verfügung. Einzige Voraussetzung: etwas Hardware-Erfahrung.

CP/M ist heute jedermann, zumindest vom Namen her, bekannt. Hier soll nun der Selbstbau eines CP/M-Rechners, einschließlich Software, beschrieben werden. CP/M ist ein Betriebssystem für die 80XX-Mikrorechnerfamilie. Es hat die Aufgabe, eine standardisierte Softwareschnittstelle für den Datenverkehr mit einem Terminal und einer Floppy-Disk-Station bereitzustellen. Das CP/M-Betriebssystem wurde schon in mc besprochen [1], so daß wir uns hier auf die technische Realisierung beschränken können. Bei CP/M gibt es zwei unterschiedliche Handelsformen: zum einen das Standard-CP/M, das auch wir verwenden werden, mit dem Benutzerstartbereich auf Adresse 100H, und dann noch eine Version mit der Startadresse der TPA (transient program area) bei 4300H.

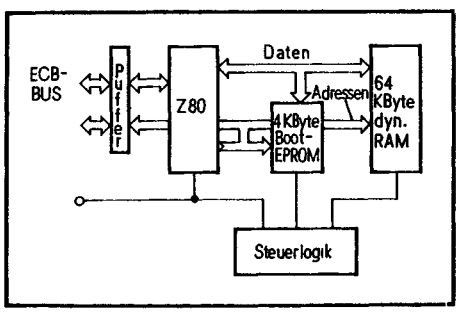


Bild 1. Das Blockschaltbild der CPU-Karte

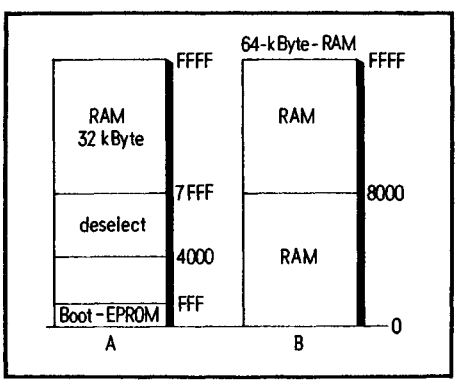


Bild 2. Das sind die beiden Speicheraufteilungen, die sich je nach Situation einstellen. Die Konfiguration A ergibt sich beim Start. Daraus entwickelt sich B, sobald sich das Monitor-Programm nach oben kopiert hat und angelaufen ist

Das Standard-CP/M hat die größte Verbreitung. Für dieses CP/M gibt es eine Vielzahl von fertigen Programmen, so die Programmiersprachen Pascal, Fortran, Basic, PL/1, Forth, Cobol, C, APL, Algol 60, ADA, Lisp, RPG, Mumps, Pilot und viele Dialekte davon. Dann gibt es die unterschiedlichsten Cross-Assembler, zum Beispiel für die Prozessoren 1802, 8048, 8051, Z8000, 68000, 8086, 6809, 6800, 6502. Ferner gibt es Datenbanksysteme, wie MDBS, oder spezielle Anwendersoftware und viele nützliche Hilfsmittel, zu denen der Leser nach Aufbau des CP/M-Computers Zugriff bekommt.
Diese Programme werden von den unterschiedlichsten Anbietern in den Handel gebracht.
Die Anzeigenseiten der in- und ausländischen Fachzeitschriften geben einen aktuellen und guten Überblick über das Angebot.

Die CPU: ein Z80

Grundlage für den CP/M-Computer wird die Z80-CPU sein, so daß sowohl 8080- als auch Z80-Programme gestartet werden können. Viel CP/M-Software, so z. B. APL, ist heute allein in Z80-Code geschrieben.
Unser Z80-Computer arbeitet mit 4 MHz, bei schnellen RAMs auch mit 6 MHz. Das ganze System ist für 6 MHz ausgelegt, um dem neuesten Stand der Technik zu entsprechen.
Der CP/M-Computer besteht aus insgesamt drei Platinen:

- der CPU-Karte, die einen Z80-Prozessor sowie 64 KByte RAM und ein Bootstrap-EPROM beinhaltet;
- der SIO-PIO-Karte, die ein serielles Interface mit zwei Kanälen und ein Parallel-Port enthält, wobei ein serieller Kanal für ein Terminal und der zweite für einen Drucker vorgesehen ist;
- der Floppy-Karte mit dem Steuer-IC 1797, das sowohl Mini- als auch Maxi-Floppys unterstützt, wobei wir mit einfacher Schreibdichte im 8-Zoll-Format arbeiten werden, um das Standard-Disketten-Format von CP/M lesen zu können (IBM-formatiert). Das IC unterstützt aber auch „Double-Density".

Zur CPU-Karte: *Bild 1* zeigt eine Blockschaltung. Die CPU ist über einen internen Bus mit dem EPROM und dem 64-KByte-Speicher verbunden. Das EPROM wird benötigt, um nach dem Einschalten überhaupt ein Programm in den RAM-Bereich bringen zu können (Booten). Als RAM-Bausteine wurden die neuen dynamischen 64-KBit-ICs verwendet. Den Refresh übernimmt der Z80-Prozessor, der einen internen Refresh-Zähler besitzt. Der Z80-Bus ist über Pufferbausteine mit dem externen Bus verbunden. Als Bus-Belegung wurde die des ECB-Busses gewählt. Dieser Bus ist sehr verbreitet.

Die Startprozedur

Bild 2 zeigt die Speicheraufteilung. Nach einem Reset des Systems oder nach dem Einschalten liegt zunächst die Konfiguration A vor: Im oberen Adreßraum sind 32 KByte RAM zugeschaltet, im unteren ist die Bootstrap-Logik eingeblendet. Dabei sind in den unteren 4 KByte das EPROM von Adresse 0 bis 0FFFH und im Bereich 4000H bis 7FFFH ein I/O-Port eingeblendet. Wird in Situation A ein Lesezugriff innerhalb des Bereichs 4000H bis 7FFFH durchgeführt, so wird der untere Bereich gegen den restlichen RAM-Bereich ausgetauscht, damit volle 64 KByte RAM zur

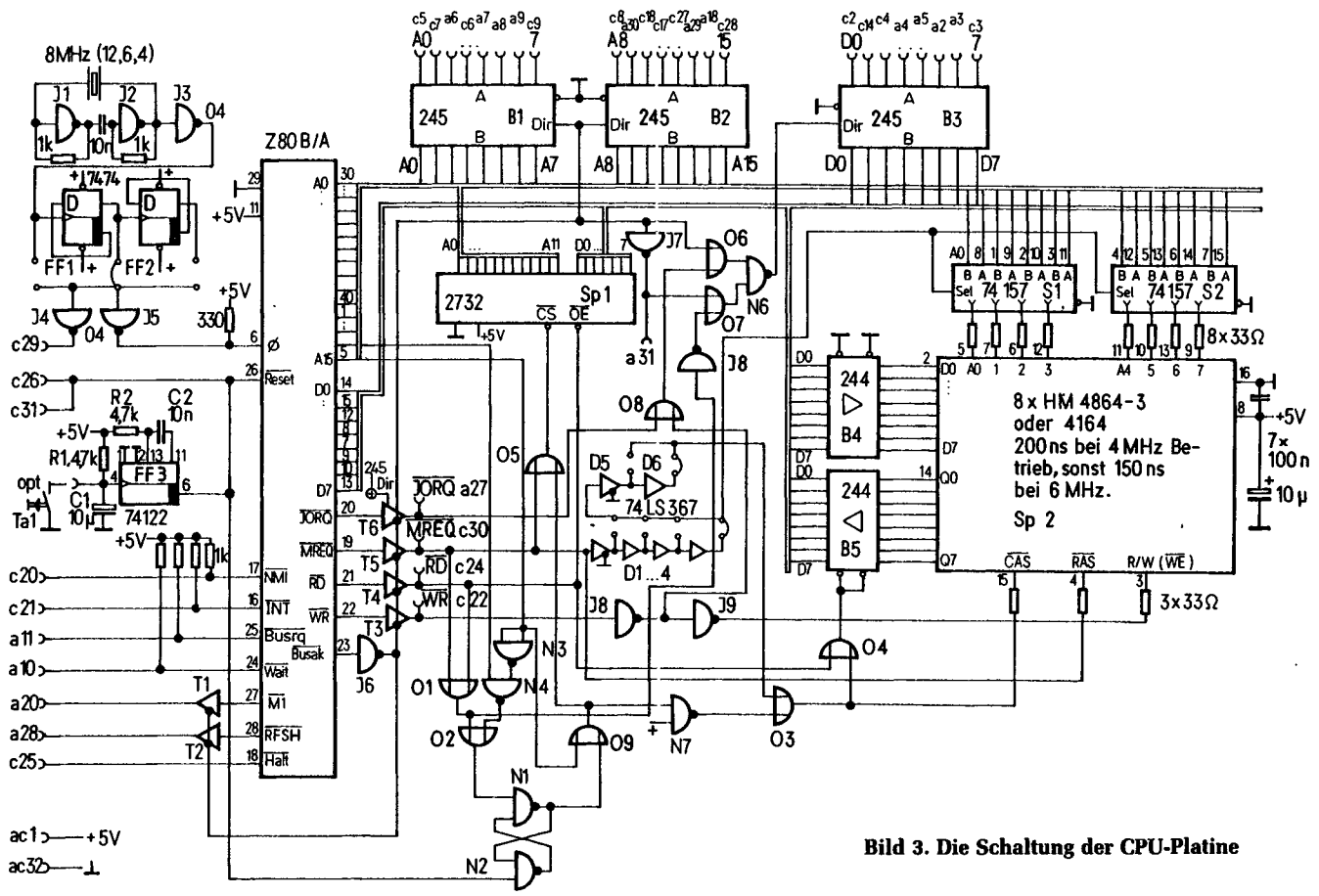


Bild 3. Die Schaltung der CPU-Platine

Verfügung stehen. Dann liegt die Situation B vor.
Der Ablauf beim Systemstart ist wie folgt: Als erstes wird ein im EPROM befindliches Monitorprogramm mit Hilfe eines ebenfalls im EPROM stehenden Blockmove-Befehls in den oberen RAM-Bereich übertragen. Dann erfolgt ein Sprung in das Monitor-Programm im oberen Bereich. Dort wird ein Lade-Befehl z. B. auf die Zelle 7000H durchgeführt. Im unteren Bereich wird jetzt das RAM zugeschaltet und das EPROM ausgeblendet. Der Monitor meldet sich dann über einen der SIO-Kanäle auf der zweiten Karte. Jetzt kann, falls die Floppy schon vorhanden ist, CP/M mit einem Monitorbefehl geladen werden. Der Grund für dieses Verfahren: Der Computer ist so voll softwaredefinierbar.

Die CPU-Karte – sehr schnell

Bild 3 zeigt die Gesamtschaltung der CPU-Karte. Die Takterzeugung der CPU wird von einem einfachen Quarz-Oszillator übernommen. Ein nachgeschalteter Teiler erlaubt es, mit unterschiedlichen Quarzen zu arbeiten. Die Standard-Frequenz ist 8 MHz. Der nachgeschaltete Teiler versorgt die CPU mit 4 MHz. Bei Betrieb mit 6 MHz muß ein schnelles EPROM verwendet werden, wobei 200 ns i. a. genügen, sowie ein schnelles RAM (150 ns). Die Schaltung wurde im Labor bis zu 6,2 MHz getestet. Für die Allgemeinheit empfiehlt es sich aber, nur mit Normalfrequenz zu arbeiten, zudem auch manche Peripherie-Karten Schwierigkeiten mit einer zu hohen Taktrate bekommen würden.

Die Reset-Logik

Die Reset-Logik besteht aus dem Monoflop FF3. Beim Einschalten lädt sich der Kondensator C1 über den Widerstand R1 auf. Erreicht die Spannung an C1 eine bestimmte Schwelle, wird das Monoflop getriggert und es wird ein Impuls am Ausgang ausgelöst, dessen Breite durch C2, R2 bestimmt ist. Es darf dabei kein statisches Signal verwendet werden, da die dynamischen Speicher sonst keinen Refresh bekommen würden – und das ist insbesondere wichtig, wenn ein Reset während des Betriebes über den Schalter ausgelöst wird, der parallel zum Kondensator C1 angebracht werden kann. Dann würde nämlich ein eventuell im Speicher vorhandenes Programm bei statischem Signal gelöscht werden.

Die Reset-Leitung führt außerdem zur Bootstrap-Logik an den Eingang von N2. N1 und N2 bilden ein RS-Flip-Flop. Nach einem $\overline{\text{Reset}}$-Puls (der Z80 verlangt ein invertiertes Signal, deshalb $\overline{\text{Reset}}$) liegt der Ausgang von N2 auf 1 und die Einblend-Logik ist aktiviert. Der eine Eingang von O9 führt damit 0-Signal, und es wird der Zustand von Adresse A15 an den Eingang von O5 und N7 weitergeschaltet. Liegt die Adresse A15 auf 0, so wird O5 freigegeben, und immer wenn ein MREQ-Signal vorliegt, wird das EPROM Sp1 freigegeben. Gleichzeitig wird aber der Ausgang von N7 logisch-1 und O3 wird gesperrt. Das heißt, der Ausgang von O3 liegt auf logisch-1. Damit bleibt der RAM-Speicher abgeschaltet. Ist A15 auf logisch-1, so wird das EPROM gesperrt und das RAM freigegeben.

Nach dem Transfer kann die Boot-Logik abgeschaltet werden. Dies geschieht durch einen Zugriff im Bereich 4000H bis 7FFFH und wird durch die Verknüpfungen N3 und N4 sowie O1 und O2 erreicht. An dem Ausgang von O2 liegt genau dann 0-Signal an, wenn der obengenannte Adreßbereich angesprochen wird.

Wie das RAM angesteuert wird

Der Datenbus wird über den Bustreiber B4 permanent auf die Dateneingänge der Speicher geschaltet. Die Ausgänge des Speichers werden mit B5 auf den Bus geschaltet, aber nur dann, wenn ein Lesezugriff vorliegt. Der Adreßbus für den Speicher muß im Multiplex-Betrieb arbeiten. Er wird von den Multiplexern S1 und S2 erzeugt, die einmal die unteren, einmal die oberen 8 Bit des Adreßbusses

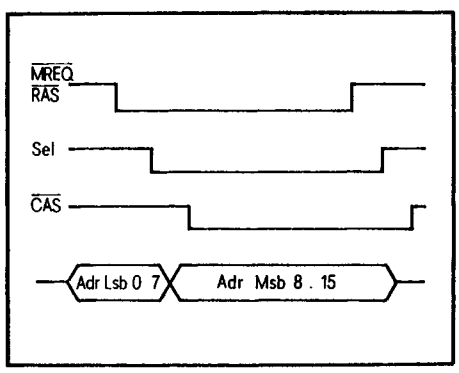


Bild 4. Das Timing für einen Speicherzugriff: Die Multiplexer S1 und S2 haben zunächst wegen SEL = 1 die acht unteren Adreßbits an den Speicher gelegt. Mit Erscheinen von MREQ werden im Speicherbaustein die Datenausgabe aus der adressierten Reihe und der Refresh vorbereitet. Mit Erscheinen von SEL werden die Multiplexer auf die oberen Adreßbits umgeschaltet und mit Erscheinen von CAS werden die jetzt vollständig adressierten Daten ausgegeben. Mit steigender Flanke von RAS ist der Refresh, der auch ohne Auslesen über CAS bewirkt werden kann, abgeschlossen

durchschalten. *Bild* 4 zeigt das Timing-Diagramm des Speichers. Der Refresh wird durch das Signal MREQ ausgelöst, das direkt auf den RAS-Eingang der Speicher führt. Das MREQ-Signal wird außerdem mit den Gattern D1 bis D4 verzögert und den Multiplexern zugeführt. Damit wird die Adresse umgeschaltet, nachdem sie mit RAS in den Speicher übernommen wurde. Nun wird MREQ mit D5 und D6 erneut verzögert und gelangt an den Eingang von O3. Falls O3 freigegeben ist, wird das Signal an O4 gelangen und damit den Speicherzugriff mit CAS auslösen. Über O4 gelangt das Signal noch an den Treiber B5, der bei einem Lese-Zugriff freigegeben wird.

Bei dem Speicher-IC 4116 war die sogenannte Precharge Time noch ein großes Problem beim Betrieb mit dem Z80. Die Precharge Time kommt insbesondere beim M1-Zyklus ins Spiel, bei dem der Speicher vom Z80 sofort wieder angesprochen wird. Die ICs 4164 sind da aber nicht so empfindlich, ferner wird durch die hohe Refreshrate in unserer Schaltung (mehr als einmal alle 2 ms) ein negativer Effekt verhindert.

Der Z80-Bus ist über die bidirektionalen Bustreiber B1 bis B3 vom externen Bus getrennt. Eine Buslogik, bestehend aus den Gattern J7, O6, O7, N5, O8 und N6, übernimmt die Richtungssteuerung dieser Bustreiber. Dies ist nötig, da zwischen externem und internem Adreßraum unterschieden werden muß. Außerdem soll die Karte ja auch DMA-fähig sein. Liegt ein Speicherzugriff der CPU vor, so wird der RAM-Bereich auf der Karte adressiert, und die Bustreiber B1 bis B3 müssen von der CPU „weggeschaltet" bleiben. Bei einem I/O-Zugriff dagegen müssen die Daten beim Lesen über den externen Bus geholt werden, und daher wird beim Lesen der Treiber B3 in Richtung CPU geschaltet. Bei DMA-Betrieb, der über das Signal BUSRQ angezeigt ist, müssen die Adreßtreiber umgekehrt treiben, da nun von außen auf den internen Speicher zugegriffen werden soll, ferner ist die Funktion des Datenbustreibers B3 je nach Anforderung (Ein oder Aus) umzuschalten. Bei DMA-Betrieb wird die Richtung nur dann von der CPU weg nach außen geschaltet, wenn MREQ und RD vorliegen, denn nur dann wird der Speicher vom DMA-Baustein lesend angesprochen. Bei älteren auf dem Markt befindlichen Platinen ist noch ein Layoutfehler bei N5 vorhanden, der dies verhindert. *Bild* 5 zeigt die Bestückungsseite der Platine, *Bild* 6 die Lötseite und *Bild* 7 den Bestückungsplan.

Aufbau-Empfehlungen

Die Platine ist sehr universell verwendbar. Zum einen besitzt sie einen ECB-Bus-kompatiblen Stecker, das ist die auf der Platine innerhalb des Euro-Formates liegende Doppellochreihe, zum anderen aber noch einen frei verdrahtbaren Bus für eigene Systeme. Die Platine besitzt exakt Europa-Format, wenn das Verdrahtungsfeld für den „Eigenbau-Bus" so abgetrennt wird, daß der DIN-Stecker für den ECB-Bus an die Hauptplatine paßt. Es empfiehlt sich beim Aufbau der Platine, alles mit Sockeln zu bestücken, da sonst ein Testen unmöglich ist. Alle passiven Bauteile werden als erstes eingelötet.

Die Reset-Taste wird über ein verdrilltes Kabel parallel zum Kondensator C1 gelötet. Nun beginnt der Test:

1. Einsetzen der ICs bis auf CPU, EPROM und RAMs.
2. Versorgungsspannung anlegen, wobei 5 V genügen.
3. Messen mit einem Oszilloskop: An Pin 6 der CPU muß der Takt anliegen.
4. Messen an Pin 26: Bei Betätigung der Reset-Taste muß an diesem Pin ein sehr kurzer Puls nach 0 V erscheinen; der Ruhepegel ist High.
5. Die Pins 17, 16, 25, 24, 11 müssen an 5 V liegen.
6. Pin 29 muß an 0 V liegen.
7. Nun Überprüfen der RAM-Bausteine. An Pin 16 liegt 0 V und an Pin 8 +5 V (anders als es bei TTL-Bausteinen üblich ist!).
8. Jetzt kann der Rest bestückt werden. Bei den Speichertypen ist etwas aufzupassen, es empfiehlt sich, Hitachi-Speicher Typ HM 4864-3 zu verwenden, da diese am problemlosesten arbeiten: Sie liegen in ihren Daten weit über der Spezifikation. Bei der 200-ns-Version läßt sich unproblematisch noch mit 6 MHz fahren, während sich beim NEC-Typ 4164 mit der 200-ns-Version Schwierigkeiten ergeben. Mitsubishi-RAMs sind jetzt beim neuen Layout ohne Modifikation einsetzbar. Bei diesen Speichern muß (!!!) Pin 1 frei bleiben. Dazu mußte beim alten Layout auf der Platine jede Verbindung zu Pin 1 durchtrennt werden. Dann arbeiten sie einwandfrei. Zu beachten ist bei Typen anderer Hersteller, daß es auch 64-KBit-RAMs mit 256 Refreshzyklen unter der Bezeichnung 4164 gibt, die hier überhaupt nicht zu gebrauchen sind.
9. Um den Monitor in Betrieb nehmen zu können, muß auch die SIO-PIO-Karte bereit sein. Dann muß auf dem Terminal die Meldung des Monitors erscheinen.
10. Die CPU-Karte kann aber auch ohne Peripherie getestet werden. Dazu wird mit dem Oszilloskop nach dem Einschalten an Pin 20 der CPU gemessen. Es ist dies der IORQ-Ausgang. An diesem Ausgang müssen bei ordnungsgemäßer Funktion Pulse erscheinen, die den überwiegenden Teil auf 1 liegen. Dann wird nämlich auf die SIO zugegriffen,

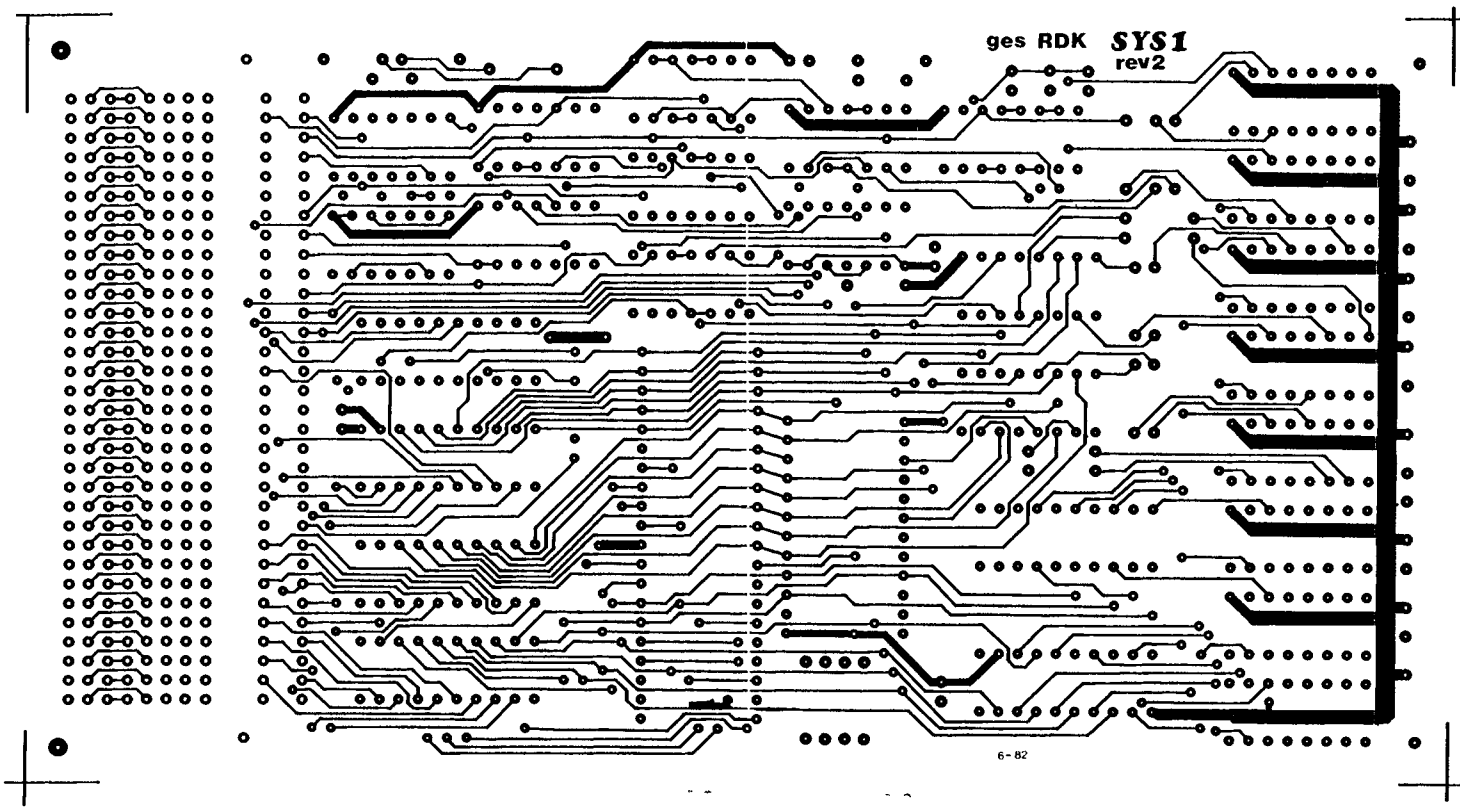


Bild 5. Die Bestückungsseite der Platine

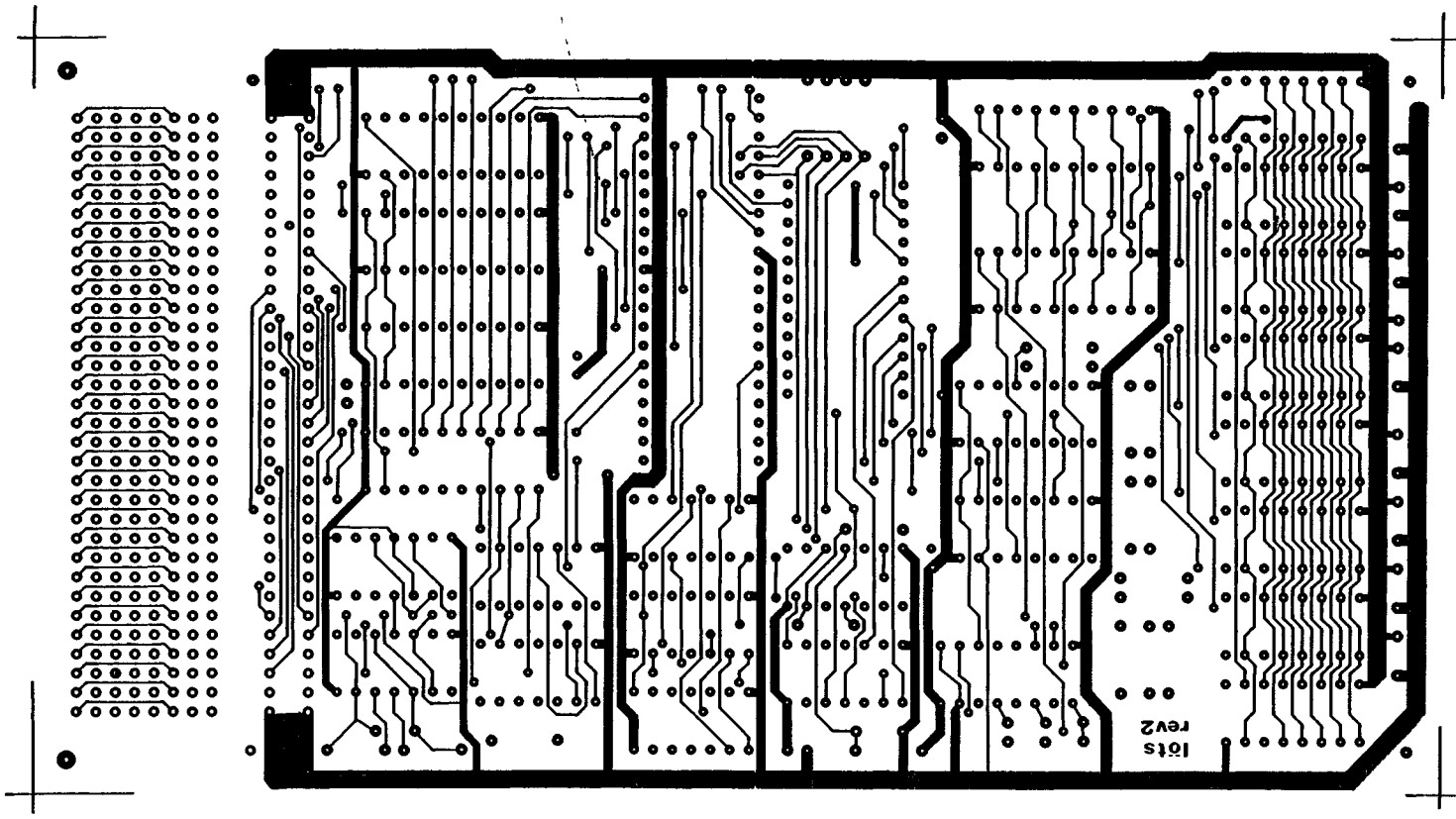


Bild 6. Die Lötseite der Platine. Das Verdrahtungsfeld links kann abgetrennt werden

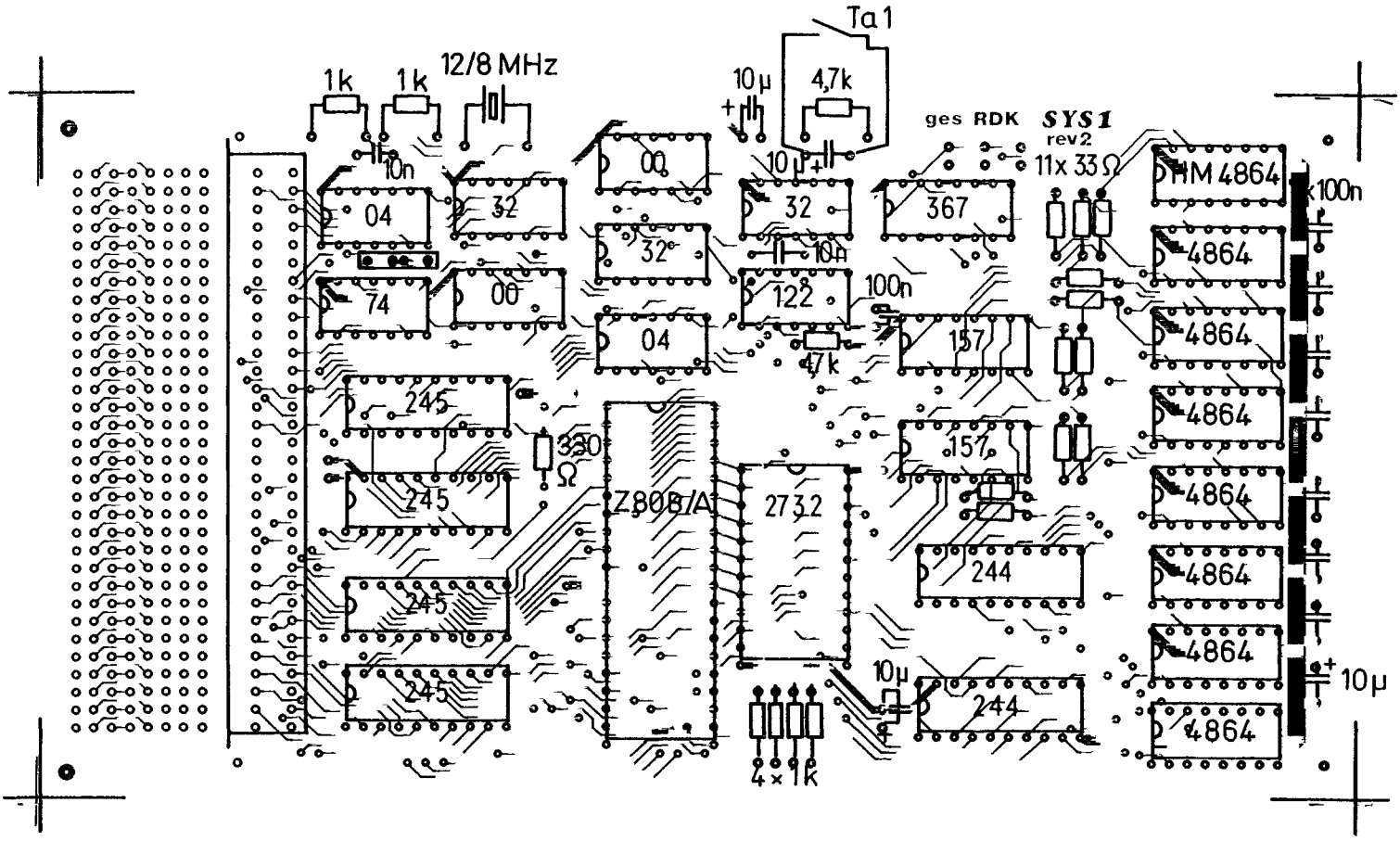


Bild 7. Der Bestückungsplan

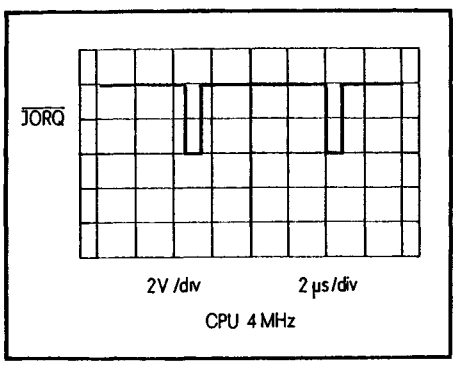


Bild 8. So sieht das Signal an IORQ aus

auch wenn diese noch nicht angeschlossen ist. Fehlen die Pulse, so liegt ein Fehler auf der Platine vor. Siehe auch Pulsdiagramm *Bild 8*.

11. Falls keine Pulse erscheinen, sind alle Lötstellen zu prüfen; wird kein Fehler gefunden, so muß ein Test-EPROM verwendet werden.
12. *Bild 9* zeigt ein kleines Testprogramm, mit dem sich verschiedene Punkte testen lassen. Das Programm hat die Aufgabe, einmal den Wert 0 und dann den Wert FFH in eine Speicherzelle zu schreiben.
13. Nach Reset bei eingestelltem Test-EPROM muß an Pin 22 eine Pulsfolge mit negativen Pulsen erscheinen. Wenn nicht, liegt ein Fehler in der EPROM-Ansteuerung vor, dann sind die Signale am EPROM zu prüfen.
14. Mit einem Zweikanal-Oszilloskop kann nun das Timing der dynamischen RAMs beobachtet werden. Es wird jeweils ein Schreibzugriff mit nachfolgendem Lese-Zugriff durchgeführt. *Bild 10* und *Bild 11* zeigen das Puls-Diagramm, dabei unterscheiden sich die beiden Bilder nur durch den Triggeranfang. Die schraffierten Gebiete hängen von dem gewählten Meßpunkt ab und sind nicht wichtig.

```
                        .phex

                        .pabs
                        ; testprogramm um speicherzugriffe zu
                        ; ueberpruefen.
                        ; 820626 rdk

0000'   3E00            loop:   mvi a,0
0002'   32 8000                 sta 8000h       ;nur eine Zelle
0005'   3A 8000                 lda 8000h       ;als repraesentant
0008'   3EFF                    mvi a,0ffh      ;am scop vergleichen
000A'   32 8000                 sta 8000h
000D'   3A 8000                 lda 8000h       ;eprom belassen
0010'   18EE                    jmpr loop       ;auf adresse 0

                        ;
                                .end
```

Bild 9. Das Speichertestprogramm

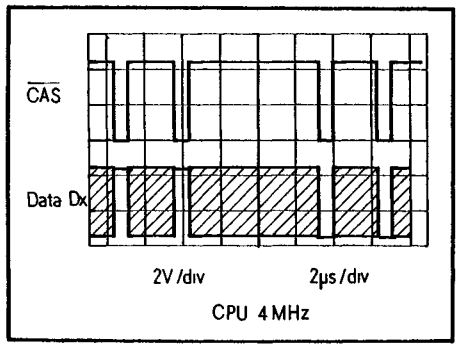


Bild 10. Diese Signale sind bei einwandfreier Funktion während eines Laufes des Testprogrammes zu beobachten

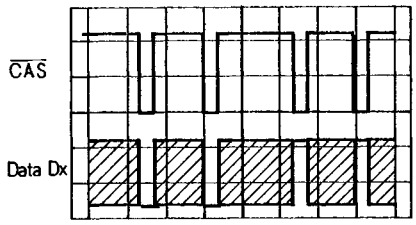


Bild 11. Derselbe Test wie in Bild 10 aber zu einem anderen Triggerzeitpunkt

	a		c	
+5V	o	1	o	+5V
D5	o	2	o	D0
D6	o	3	o	D7
D3	o	4	o	D2
D4	o	5	o	A0
A2	o	6	o	A3
A4	o	7	o	A1
A5	o	8	o	A8
A6	o	9	o	A7
-WAIT	o	10	o	
-BUSRQ	o	11	o	IEI
A18	o	12	o	A19
+12V	o	13	o	
	o	14	o	D1
-5V	o	15	o	-15V
2PHI	o	16	o	IEO
A17	o	17	o	A11
A14	o	18	o	A10
+15V	o	19	o	A16
-M1	o	20	o	-NMI
	o	21	o	-INT
(-12V)	o	22	o	-WR
	o	23	o	
VC MOS	o	24	o	-RD
	o	25	o	-HALT
	o	26	o	-PWRCL
-IORQ	o	27	o	A12
-RFSH	o	28	o	A15
A13	o	29	o	PHI
A9	o	30	o	-MREQ
-BUSAK	o	31	o	-RESET
GND	o	32	o	GND

Bild 12. Die Busbelegung nach ECB

```
                        ;****************************************
                        ;* Boot-Logik fuer die 64K CPU          *
                        ;* Rolf-Dieter Klein 820626             *
                        ;* Dabei ist die Bootsoftware           *
                        ;* unabh(ngig von der eigenen Lage      *
                        ;* und eignet sich daher gut fuer       *
                        ;* Tests im laufenden System            *
                        ;****************************************

EFE3                    .loc 0efe3h     ;wegen adressrechnung
EFE3                    begin:
EFE3    31 FFFF         lxi sp,0ffffh   ;vorlaeufig festlegen
EFE6    3EC9            mvi a,0c9h      ;RET-Befehl laden
EFE8    32 F000         sta 0f000h      ;dort RET ablegen
EFEB    CD F000         call 0f000h     ;um adresse des ROMs
EFEE                    anf:            ;zu bestimmen
EFEE    3B              dcx sp
EFEF    3B              dcx sp          ;rueckkehradresse holen
EFF0    D1              pop d           ;nach DE
EFF1    21 0012         lxi h,main-anf  ;differenz zu hauptprogramm
EFF4    19              dad d           ;dazu rechnen
EFF5    11 F000         lxi d,0f000h    ;ziel
EFF8    01 0FFF         lxi b,0fffh     ;LAENGE DES TRANSFERS 4K-xx
EFFB    EDB0            ldir            ;transport ins RAM ausfuehren
EFFD    C3 F000         jmp 0f000h      ;START ANWENDERPROGRAMM
                        ;
F000                    main:           ;ANWENDERPROGRAMM
F000    3A 7000         lda 7000h       ;RAM in den unteren BEREICH !!!
                        ;
                        ;--------- Anwender software HIER -------------
```

Bild 13. Ein Urlade-Programm zur allgemeinen Verwendung

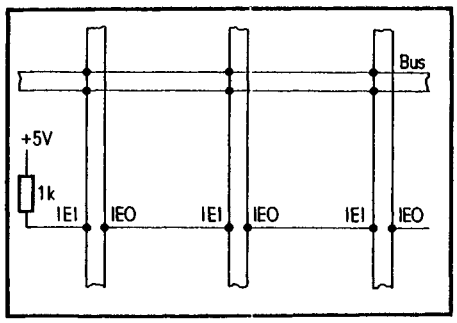


Bild 14. Die ECB-Bus-Verdrahtung. Wichtig: IEI und IEO sind wechselseitig miteinander verbunden

Es werden alle acht Datenleitungen von D0 bis D7 direkt an der CPU mit dem einen Kanal gemessen und der andere Kanal wird an den Anschluß „CAS" der Speicherbausteine geklemmt. Am besten läßt sich „CAS" direkt am 33-Ω-Widerstand abgreifen.

Damit sollte die CPU funktionsfähig werden.

Abschließend sei noch in *Bild 12* die allgemeine Bus-Belegung des ECB-Busses, Elzet-80-Variante, abgedruckt, von dem allerdings nicht alle Signale verwendet wurden.

Die −12-V-Leitung wurde für die SIO auf eine freie Busleitung gelegt, um −15 V zu vermeiden.

Bild 13 ist ein kleines Programm, mit dem gezeigt wird, wie der Boot-Vorgang von der Software-Seite her aussieht. Das Programm lädt den nachfolgenden beliebigen Programmteil in das obere Speichergebiet und schaltet von da aus den unteren EPROM-Bereich gegen den RAM-Bereich um. Das Programm wird dabei, beginnend von EFE3, auf Adresse 0 im ROM gelegt.

Beim Bus werden alle Leitungen, bis auf IEI, IEO parallel verdrahtet, *Bild 14* zeigt das Verbindungsschema.

Literatur

[1] *Sergel, Karl-Heinz:* CP/M – Eine Sache mit Zukunft. mc 1, 1982.

[2] *Oettle, F., Reichle, T.:* Dynamische Speicher. mc 3, mc 4, 1982.

Tabelle: Stückliste für die CPU-Karte

1×	Z80-A-CPU		
8×	HM 4864-3 oder NEC 4164-2		
1×	EPROM 2732 A mit Monitorprogramm		
4×	74LS245		
2×	74LS244		
2×	74LS00		
1×	7404 bei Oszillator einsetzen		
1×	74LS04		
3×	74LS32		
1×	74LS74	11×	33 Ω/⅛ W
1×	74LS122	1×	330 Ω
2×	74LS157	6×	1 kΩ
1×	74LS367	2×	4,7 kΩ
2×	10 nF		
8×	100 nF keram.		
4×	10 µF Tantal		
1×	VG-Leiste 64pol a–c		
1×	Quarz 8 oder 12 MHz (siehe Text)		
1×	Sockel 40polig		
1×	Sockel 24polig		
6×	Sockel 20polig		
11×	Sockel 16polig		
9×	Sockel 14polig		

Platinen und Bausätze liefert die Fa. Graf, GES, Postfach 1610, 8960 Kempten.

Rolf-Dieter Klein:

Die SIO/PIO-Karte

Der mc-CP/M-Computer ist unvollständig, solange er nur aus einer Zentralplatine ohne vernünftiges I/O-Interface besteht. Hier nun eine SIO/PIO-Karte, die zwei serielle Kanäle nach V.24 (RS-232C) und zwei Acht-Bit-Parallel-Eingänge mit Handshake-Möglichkeiten zur Verfügung stellt. Der Monitor zum System wird nur die seriellen Kanäle benutzen. Darüber werden Terminal und Drucker angeschlossen. Die übrigen Ports stehen dem Benutzer zur Verfügung.

Der Name der Karte ist von den Namen der verwendeten Zilog-Bausteine abgeleitet. SIO heißt „Serial Input/Output" und PIO entsprechend „Parallel Input/Output". Das mc-CP/M-System ist schon voll funktionsfähig, wenn nur die serielle Schnittstelle bestückt ist, also nur die SIO eingesetzt ist.
Bild 1 zeigt das Blockschaltbild der SIO/PIO-Karte. Die Bausteine SIO und PIO sind über eine Buslogik mit dem ECB-Bus verbunden. Die SIO erhält von zwei Baudraten-Generatoren den Übertragungstakt.
Bild 2 zeigt den Gesamtschaltplan. Der Datenbus wird über den Bustreiber B1 vom internen Bus getrennt, bei den restlichen Signalen ist dies nicht nötig. Die Platine arbeitet bei einer CPU-Frequenz von 4 MHz mit der A-Version der Bausteine. Sollen 6 MHz verwendet werden, so empfiehlt es sich, die teilweise schon erhältlichen B-Versionen zu verwenden. Allerdings ergab ein Test mit der A-Version auch einwandfreies Funktionieren. Die Karte belegt 8 Adressen, deren Bereich mit Jumpern bei J eingestellt werden kann. Wird das Monitor-Programm aus mc verwendet, so bleiben die Brükken offen und es ergeben sich für SIO und PIO folgende Adressen:

0F0H Daten SIO A,
0F1H Status SIO A,
0F2H Daten SIO B,
0F3H Status SIO B,
0F4H Daten PIO A,
0F5H Command PIO A,
0F6H Daten PIO B,
0F7H Command PIO B.

Mit den Gattern N1 und I1 wird der Bustreiber B1 immer nur dann freigegeben, wenn der obige Adreßbereich angesprochen wurde. Die Richtungsumschaltung von B1 erfolgt mit dem RD-Signal. Der V.24-Bus ist mit den Leistungstreibern B2 und B3 gepuffert und entspricht den gängigen Pegeln. Die Steckerbelegung ist die in Europa gebräuchliche Form. Wegen des geringen Platzes auf der Karte sind die beiden Stecker spiegelbildlich angeordnet. Mit zwei auf unterschiedlichen Seiten sitzenden V.24-Stift-Verbindern können zwei Anschlüsse an der Frontseite einer einzigen Platine bereitgestellt werden. Die Baudrate kann man für jeden Kanal getrennt einstellen. Dazu dienen die beiden Baudraten-Generatoren BD1 und BD2, die mit einem gemeinsamen Quarz-Takt betrieben werden.
Die Anschlüsse der PIO sind frei verwendbar. Sie sind an einen 24poligen Sockel geführt. Es kann auch eine Stiftreihe verwendet werden.
Bild 3 zeigt die Lötseite der Platine, *Bild 4* die Bestückungsseite und *Bild 5* zeigt den Bestückungsplan. Für alle ICs werden dabei Sockel eingelötet.

Aufbau und Test der Schaltung:

1. Es werden alle passiven Bauteile und Sockel eingelötet.

2. Einstellen der Baudrate des verwendeten Terminals. Dazu dient das *Bild 6*, das die Baudraten-Einstellung zeigt. S 0 bis S 3 in Bild 6 entsprechen S 1 bis S 4, wenn es um SIO A geht, S 5 bis S 8, wenn es um SIO B geht.

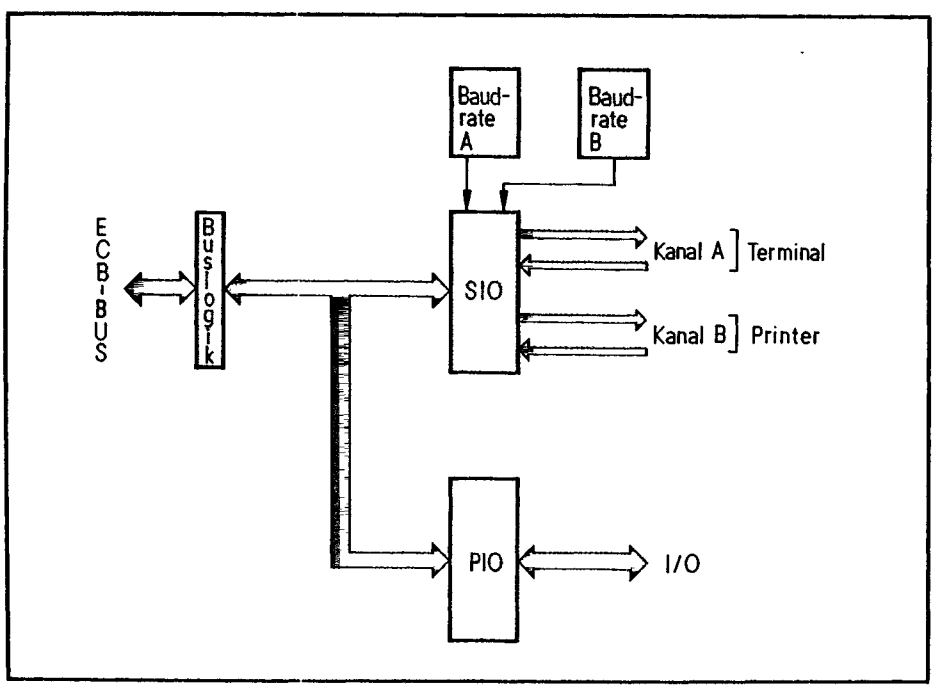


Bild 1. Das Blockschaltbild der SIO/PIO-Karte

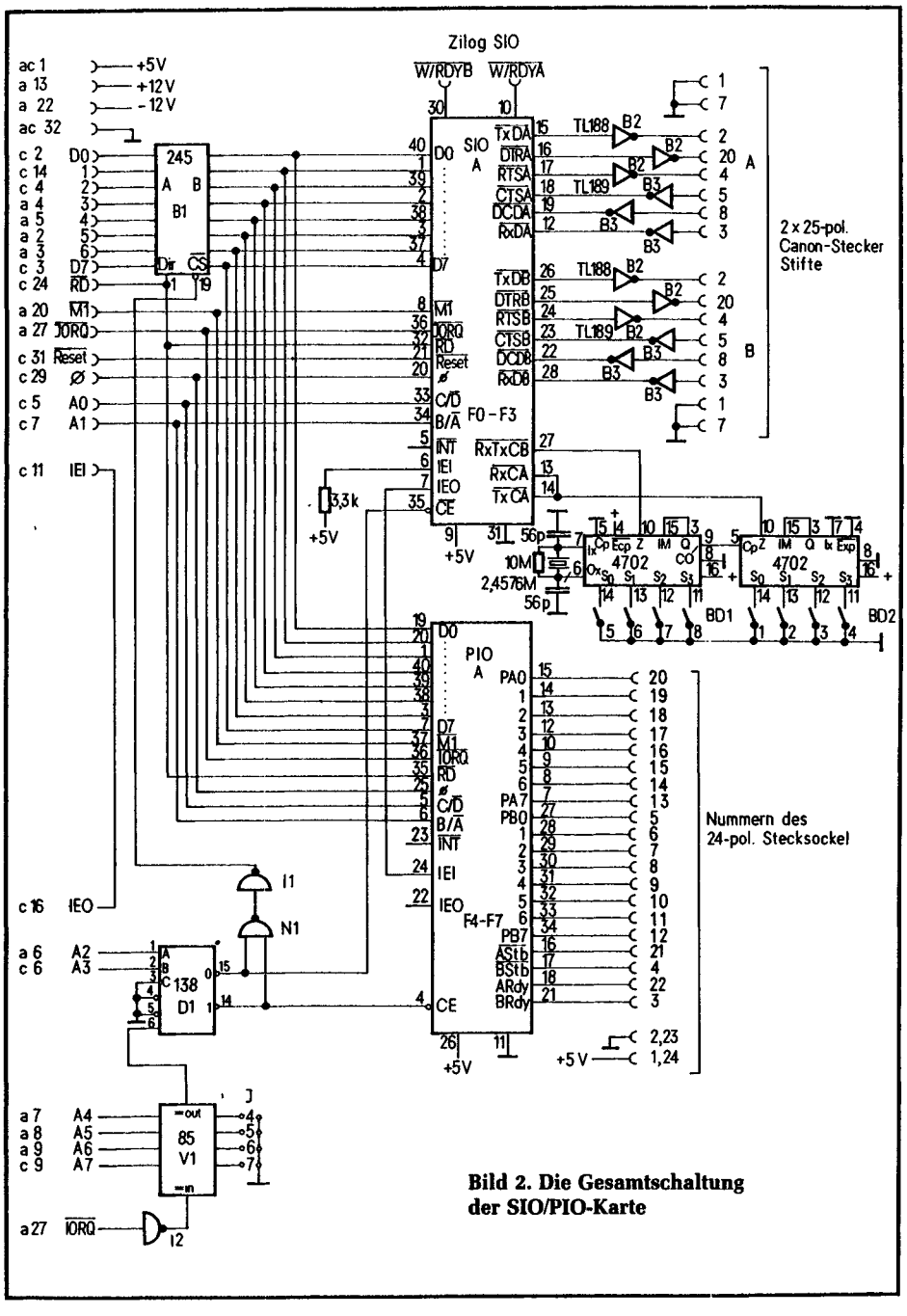


Bild 2. Die Gesamtschaltung der SIO/PIO-Karte

3. Einsetzen aller ICs bis auf SIO, PIO und Bustreiber B1.
4. Einschalten der Versorgungsspannung und Messen mit dem Oszilloskop: Pin 9 an der SIO muß auf +5 V liegen, Pin 31 an Masse. Sind die +12 V- und –12-V-Spannung vorhanden? Am ECB-Bus muß a13 an +12 V liegen und a22 an –12 V.
5. Pin 26 der PIO muß an +5 V liegen, Pin 11 an Masse.
6. An Pin 13, 14, 27 der SIO muß die 16fache Taktrate der gewählten Baudrate anliegen.
7. Pin 20 der SIO und Pin 25 der PIO müssen den CPU-Takt führen.
8. Die Pins 15, 16, 17, 18, 19, 12, 26, 25, 24, 23, 22, 28 der SIO müssen einen Pegel zwischen 0 V und +5 V haben (Test ob V.24-Treiber ok sind).
9. An Pin 35 der SIO muß die in *Bild 7* gezeigte Impulsfolge anliegen.
10. An Pin 19 des Bustreibers B1 ebenfalls.
11. Einsetzen der restlichen ICs
12. Messen an Pin 15 der SIO. Nach Betätigen des RESET-Tasters muß hier eine Pulsfolge erscheinen, falls die CPU mit Monitor läuft.
13. Falls die Impulsfolge da war, kann das Terminal angeschlossen werden. Dort muß dann die Monitormeldung erscheinen. Bei Eingabe eines Zeichens muß eine Reaktion erfolgen.
14. Ist dies nicht der Fall, und kam vorher die Pulsfolge, dann Tastkopf des Oszilloskopes an Pin 12 der SIO. Bei Betätigung einer Taste des Terminals muß an Pin 12 eine Pulsfolge erscheinen. Wenn ja, so stimmen die Baudrateneinstellung oder die Stop-Bit-Einstellung nicht. Es müssen 8 Bits mit einem Stop-Bit (ohne Parität) gesendet und empfangen werden können.
15. Kommen auch die Pulse bei Pin 12 nicht, so muß mit einem Test-EPROM weitergetestet werden.
16. *Bild 8* zeigt ein kurzes Testprogramm. Damit läßt sich die Decodierung des Systems testen. *Bild 9* zeigt das Oszillogramm nach Start des Testprogramms für die beiden Meßpunkte CS an der SIO und CS an der PIO.
17. Ist das Terminal angeschlossen, muß noch die PIO getestet werden. Diesmal kann das Testprogramm mit Hilfe des Monitor-Programms eingegeben werden. *Bild 10* zeigt das Programm.
18. Die beiden Ports der PIO A und B werden zum Zählen veranlaßt. *Bild 11* und *Bild 12* zeigen die Oszillogramme jeweils für Bit 0 von Port A und Port B. Die höherwertigen Bits müssen jeweils die halbe Frequenz des Vorgängers besitzen.

Damit ist der Test abgeschlossen und die Karte müßte funktionieren. Bei der Fehlersuche ist immer zunächst auf Lötzinnbrücken zu achten, da dies der häufigste Fehler ist. Ferner ist der verwendete Bus zu überprüfen; auch eine Kontrolle, ob alle Spannungen (+12 V und insbesondere die extra Leitung –12 V) richtig ankommen, ist sehr nützlich. Mit dem System CPU und SIO/PIO können nun schon Software-Entwicklungen durchgeführt werden. Der Floppy-Anschluß wird mit einer weiteren Karte möglich.

Kommunikation mit Terminal und Drucker

Bild 13 zeigt das Verdrahtungsschema für Terminal und Drucker. Die Steuerleitungen CTS, DCD werden von unserem Monitor voll bedient. Das heißt, mit ihnen ist es möglich, die Datenausgabe zu stoppen. Der Ruhepegel liegt bei +12 V. CTS und DCD müssen diesen Pegel besitzen, sonst meldet sich der Monitor nicht.

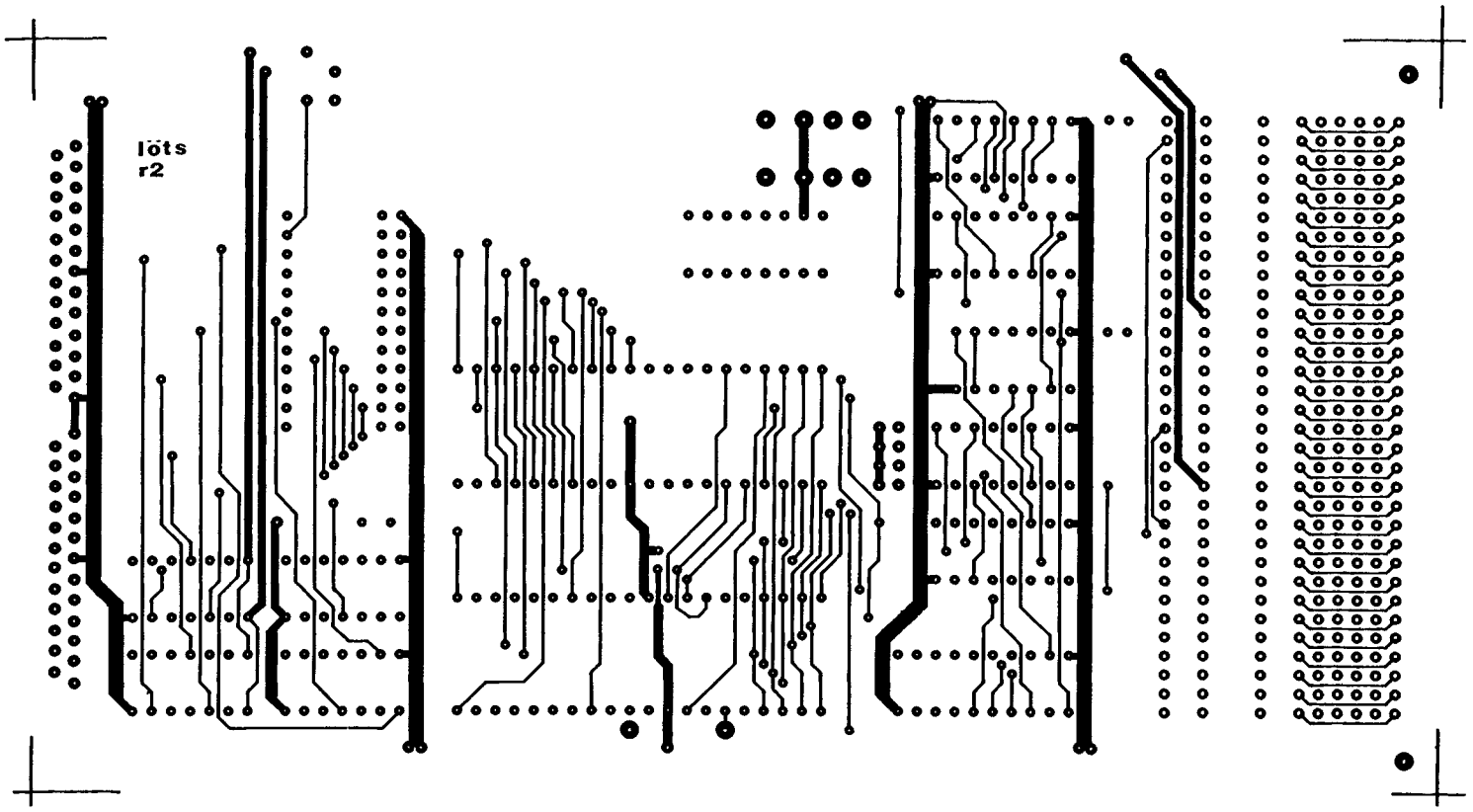


Bild 3. Die Lötseite der Platine

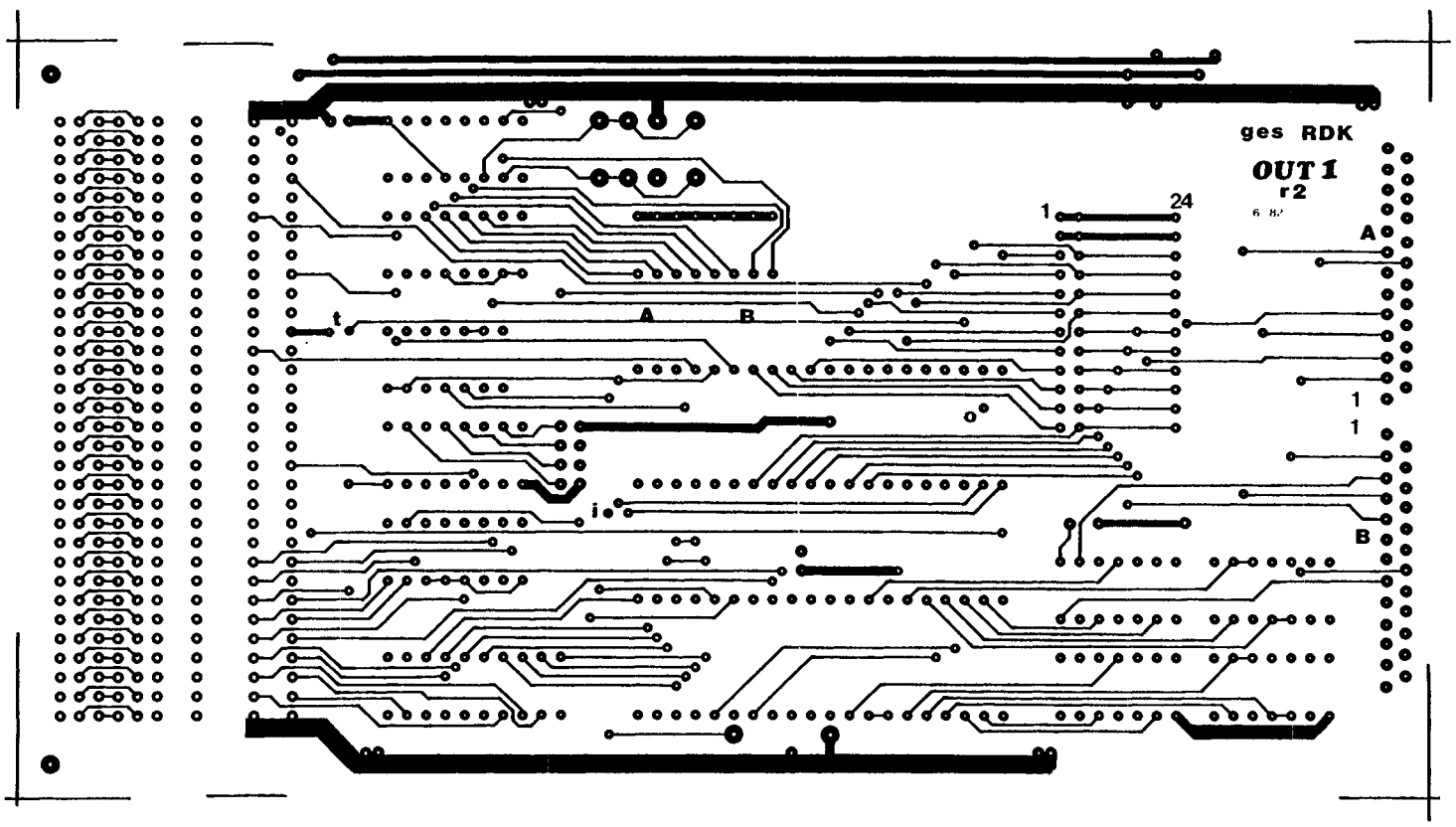


Bild 4. Die Bestückungsseite der Platine

Wird ein Terminal verwendet, das diese Leitungen nicht bedient, so können CTS, DCD und RTS am Stecker der SIO/PIO-Karte miteinander verbunden werden.

RTS liegt normalerweise ständig auf +12 V. Für den Druckeranschluß reicht natürlich die Leitung T × D (2) neben CTS und DCD schon aus. Die Steuerung über die Leitungen CTS und DCD ist beim Terminal und beim Drucker sehr praktisch, denn gerade bei Druckern, die ja langsamer sind, als die Datenübertragung (meistens), ist es nötig, den Rechner von Zeit zu Zeit anzuhalten, damit keine Zeichen verloren gehen. Der Drukker wird über den Vektor in F00F angesprochen, wenn zuvor das Kommando AL = L gegeben wurde.

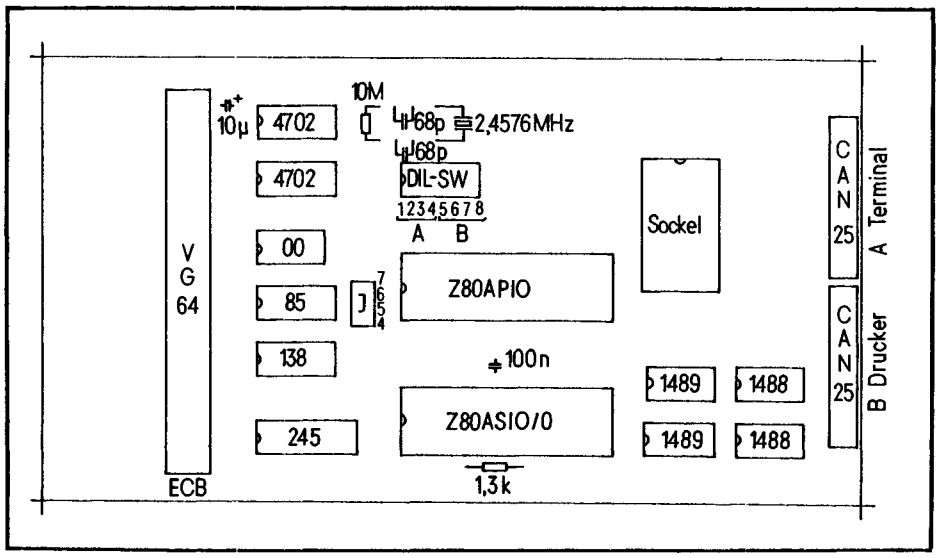


Bild 5. Der Bestückungsplan

I 11	I 12	I 13	I 14	<- Pin	
I S 3	I S 2	I S 1	I S 0	I Baudrate	
I 0	I 0	I 0	I 0	I 19200 (extn)	
I 0	I 0	I 0	I 1	I 19200 (extn)	
I 0	I 0	I 1	I 0	I 50	
I 0	I 0	I 1	I 1	I 75	
I 0	I 1	I 0	I 0	I 134.5	
I 0	I 1	I 0	I 1	I 200	
I 0	I 1	I 1	I 0	I 600	
I 0	I 1	I 1	I 1	I 2400	
I 1	I 0	I 0	I 0	I 9600	
I 1	I 0	I 0	I 1	I 4800	
I 1	I 0	I 1	I 0	I 1800	
I 1	I 0	I 1	I 1	I 1200	
I 1	I 1	I 0	I 0	I 2400	
I 1	I 1	I 0	I 1	I 300	
I 1	I 1	I 1	I 0	I 150	
I 1	I 1	I 1	I 1	I 110	

Bild 6. Baudraten-Tabelle

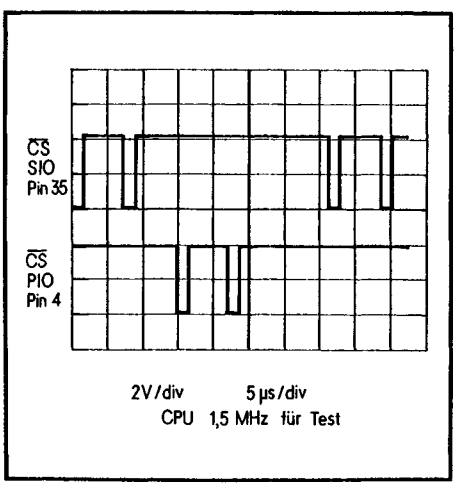


Bild 9. Oszillogramm zum Decoder-Test

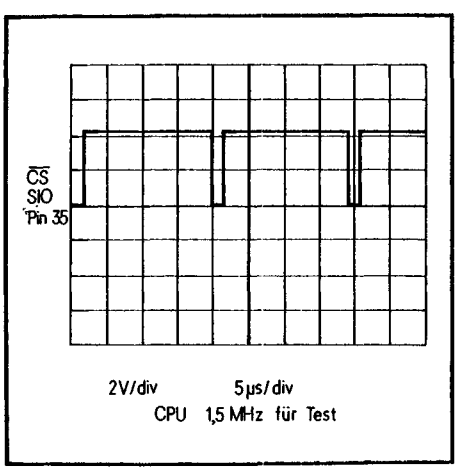


Bild 7. Das Oszillogramm zum Standard-Monitor

```
                        ;*********************************
                        ;* Test Dekodierung SIOPIO       *
                        ;* Rolf-Dieter Klein 820626      *
                        ;*********************************

0000'   DBF0            lp:     in 0f0h         ;sio eingabe
0002'   D3F0                    out 0f0h        ;ausgabe sio
0004'   DBF4                    in 0f4h         ;eingabe pio
0006'   D3F4                    out 0f4h        ;ausgabe pio
0008'   18F6                    jmpr lp

                                .end
```

Bild 8. Das Testprogramm für den Decoder-Test

```
                        ;****************************
                        ;* Test Funktion der PIO    *
                        ;* Rolf-Dieter Klein 820626 *
                        ;****************************

0000'   3E0F                    mvi a,00001111b ;output mode
0002'   D3F5                    out 0f5h        ;port a
0004'   D3F7                    out 0f7h        ;port b
0006'   3C              lp:     inr a           ;daten immer
0007'   D3F4                    out 0f4h        ;auf ports a
0009'   D3F6                    out 0f6h        ;und port b
000B'   18F9                    jmpr lp

                        .end
```

Bild 10. Das Testprogramm für den Test der PIO

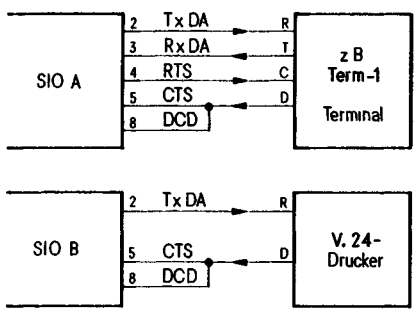


Bild 13. Zum Anschluß eines Terminals oder Druckers

Tabelle:
Stückliste für die SIO/PIO-Karte

1×	SIO-0-A (Zilog!)
1×	PIO-A (optional)
2×	4702BPC von Fairchild oder SE
1×	74LS00
1×	74LS85
1×	74LS138
1×	74LS245
2×	1489 (75189)
2×	1488 (75188)
1×	1,3 kΩ
1×	10 MΩ
2×	68 pF
1×	100 nF ker.
1×	10 µF Tantal
1×	Quarz 2,4576 MHz
1×	DIL-Schalter, 8polig
1×	VG-Leiste, 64polig, a-c
2×	Sockel, 40polig
1×	Sockel, 24polig
5×	Sockel, 14polig (optional)
4×	Sockel, 16polig
1×	Sockel, 20polig
2×	25poliger Cannon-Stecker (Winkelform)

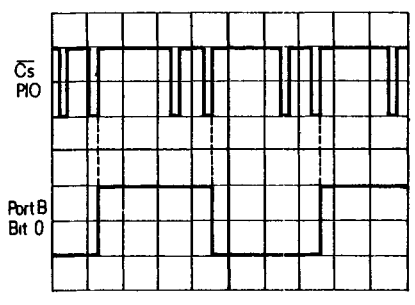


Bild 12. Signal an Bit 0 von PIO B

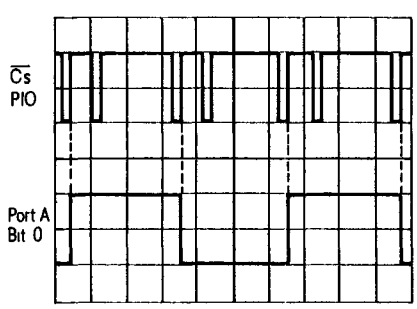


Bild 11. Das Signal an Bit 0 von PIO A

Rolf-Dieter Klein:

Der mc-Monitor

Der hier beschriebene Monitor erweckt Ihren mc-CP/M-Computer zum Leben. Zwar ist die Computerei allein mit Monitor sehr mühsam, aber auch billig und lehrreich. Wer sich also ohne Softwarekenntnisse an den Nachbau gewagt hat, der sollte sich diesen Monitor in ein EPROM schießen lassen und dann in den Computer einsetzen. Vorausgesetzt, es ist ein Terminal zur Hand, können dann die Kenntnisse erworben werden, die später noch bei der Anpassung von CP/M und beim Betrieb des Computers wichtig sind.

Der Monitor ist auf den mc-CP/M-Computer abgestimmt. Er setzt die CPU-Karte und die hier im Heft beschriebene SIO-PIO-Karte voraus. Allerdings ist er änderungsfreundlich geschrieben, so daß er auch ohne weiteres auf andere Z-80-Computer umgeschrieben werden kann. Der Monitor enthält nicht nur die gängigen Monitorbefehle, wie zum Beispiel S⟨XXXX⟩, womit der Inhalt der Speicherzelle XXXX angezeigt (S von substitute) und dann auch geändert werden kann, sondern auch schon Routinen zur Floppy-Verwaltung und Vorkehrungen für einen CP/M-Bootstrap. Er ist nach dem sogenannten TDL-Zapple-Monitor gestaltet.

Das Raffinierte an dem Monitor ist, daß er sich selbst mit Hilfe eines Auto-Boot-Teiles in die oberen 4 KByte des Systemspeichers (Adresse F000 bis FFFF) einschreibt und durch Sprung dorthin startet. Erst durch diese Technik ist es möglich, den gesamten Speicherplatz des Computers nach dem Starten als Schreib-Lesespeicher zur Verfügung zu stellen, denn der Monitor blendet als erstes das EPROM, aus dem er generiert wurde, aus und schaltet den parallel liegenden RAM-Bereich hinzu. Gleichzeitig meldet sich der Monitor dann über den auf Adresse 00F0 liegenden Kanal des SIO-Bausteines der SIO-PIO-Karte.

Bild 1 zeigt den Anfang des Monitors. Im EPROM liegt er auf Adresse 0, damit er bei Reset oder Systemstart anlaufen kann. Zunächst wird der Stackpointer mit LD SP,0FFFFH voreingestellt. Dann wird der Akku mit dem Op-Code für einen Return-Befehl geladen, der dann nach F000 gebracht wird, wo nach Voraussetzung RAM-Speicher zur Verfügung steht. Ein Unterprogrammsprung nach F000 führt sofort zur Rückkehr in den Boot-Teil und hinterläßt auf dem Systemstack die der Absprungadresse folgende Adresse aus dem Boot-Teil (hier 000B), die mit zweimal DEC SP und anschließendem POP DE im Prozessor bereitgestellt wird. Mit zwei Befehlen wird nun Register HL als Pointer auf den zu transferierenden Bereich (DE + HAUPT-ANF) eingestellt. DE wird als Pointer auf den Zielbereich gerichtet, BC wird mit der Länge des zu transferierenden Bereiches geladen (FFF – was eigentlich zuviel ist, denn der Monitor ist nicht volle 4 KByte lang) und damit kann der Blocktransfer mit LDIR gestartet werden. In 001A steht der abschließende Sprung nach F000.

Der Monitor

Leider kann der Monitor nur als Hex-Dump voll abgedruckt werden, alles andere würde zu lang. Ein Sammel-Listing, das auch die Minifloppy- und Formatier-Routinen sowie das CP/M-BIOS als Assembler-Source enthält, ist für 19,40 DM (inkl. 1,40 DM Porto) gegen Einsendung eines Verrechnungsschecks beim Softwareservice (Postfach 37 01 20, 8000 München 37) erhältlich. Der Monitor beginnt mit einer Sprungtabelle in

```
                        ;**********************************
                        ;* RDK MONITOR  1.0  800702       *
                        ;* VERSION PIO,MINI,MAXI          *
                        ;* REV 3.2 830205  ready direct   *
                        ;* rev 3.3 830318  spez umsch     *
                        ;* rev 3.4 830408  dcd cts rts    *
                        ;*                 fuer term1     *
                        ;**********************************

0000'                   CSEG
                        ;
                        ; IM ROM AUF ADRESSE 0
                        ;
0000'                   BEGINN:
0000'  31 FFFF          LD SP,0FFFFH    ;DUMMY-LOAD
0003'  3E C9            LD A,0C9H       ;ABLAGE RET-BEFEHL
0005'  32 F000          LD (0F000H),A   ;AUF ZIELADRESSE
0008'  CD F000          CALL 0F000H     ;FESTSTELLEN
0008'                   ANF:            ;DER EIGENEN ADRESSE,
0008'  3B               DEC SP          ;UM START VON BELIEBIGER
000C'  3B               DEC SP          ;STELLE AUS ZU ERMOEGLICHEN
000D'  D1               POP DE          ;FUER TESTS SEHR GUT
000E'  21 0012          LD HL,HAUPTP-ANF
0011'  19               ADD HL,DE
0012'  11 F000          LD DE,0F000H    ;TRANSPORT ALLER ZELLEN
0015'  01 0FFF          LD BC,0FFFH     ;4K TRANSFER
0018'  ED B0            LDIR            ;EIGENTLICH ZUVIEL,ABER
001A'  C3 F000          JP 0F000H       ;STOERT NICHT WEITER
                        ;
001D'                   HAUPTP:         ;DISTANZ-MERKER
                        ;
                        .PHASE 0F000H   ;START DES MONITORS
                                        ;CODE DENOCH HINTER BOOT BEREICH

                        ; --- START DES EIGENTLICHEN MONITORS ---
                        ;
000D                    CR      EQU     0DH
000A                    LF      EQU     0AH
0007                    BELL    EQU     7
00FF                    RUB     EQU     0FFH
0000                    FIL     EQU     0
0007                    MAX     EQU     7

0038                    RST7    EQU     38H
                        ; ANFANG VERSCHIEBBAR
                        ; der Vektor RST7 wird fuer Breakpoints gebraucht
                        ;

                        ;
00F0                    SIOADAT EQU     0F0H
00F1                    SIOASTS EQU     0F1H
00F2                    SIOBDAT EQU     0F2H
00F3                    SIOBSTS EQU     0F3H
```

Bild 1. Der Anfang des Monitors mit Sprungtabelle, Titel und Init-Teil

```
                        ;* VEKTOR TABELLE *
                        ; CSTS OFFH IN A, FALLS ZEICHEN DA
                        ; IOCHK IN A AKTUELLE IO-KONFIGURATION
                        ; IOSET VERAENDERN, IO IN C-REG
                        ; MEMCK IN B HIGH IN A LOW MEMORY
                        ; TRAP/BREAKPOINT ENTRY


                        ; START DES MONITORPROGRAMMS
F000    C3 F036         JP BEGIN
F003    C3 F039         JP CI
F006    C3 F04C         JP RI
F009    C3 F062         JP CO
F00C    C3 F076         JP POO
F00F    C3 F08D         JP LO
F012    C3 F0A4         JP CSTS
F015    C3 F0B7         JP IOBYTE
F018    C3 F0BB         JP IOSET
F018    C3 F0C0         JP MEMCK
F01E                    TRAP:
F01E    C3 F033         JP RESTART1
F021    C3 F033         JP RESTART1     ;-FLOPPY EXEC VEKTOR HARD
F024    C3 F159         JP MAXI         ;FLOPPY EXEC VEKTOR SOFT
F027    C3 F120         JP MINI         ;FLOPPY EXEC VEKTOR MINI
F02A    C3 F033         JP RESTART1     ;-IMISYS PLATTE EXEC VECTOR
                        ;
                        ;
                        ;
F02D    F0EF            DEFW TABSTART   ;TABELLEN-START, USER-BEREICH
                                        ;SPRUNGTABELLE, IMMER
                                        ;INDIREKT VERWENDEN
F02F    FC91            DEFW LASTMON    ;ADRESSE LETZTE BELEGTE ZELLE
                                        ;DES MONITORS DANACH PATCH FREI
F031    F539            DEFW FREEMEM    ;Lade-Adresse hinter IO-Gebiet
                        ;
F033                    RESTART1:.
F033    C3 F852         JP RESTART      ;kann kurzgeschlossen werden
                                        ;nach freemem ueberschreibung

F036                    BEGIN:
F036    C3 F558         JP BEGIN1       ;kurzschliessen ggf.
                        ;
                        ; ende fixed gebiet

                        ;
                        ;**************************************
                        ;* IO Routinen                        *
                        ;**************************************

                        ;+++ CI MAIN ROUTINE +++
F039                    CI:
F039    3A F101         LD A,(IOBYT)
F03C    E6 03           AND 3   ;CONSOLMASKE
F03E    20 09           JR NZ,CI1       ;ANDERE DEVICES
F040                    TTYIN:
F040    DB F1           IN A,(SIOASTS)
F042    E6 01           AND 1
F044    28 FA           JR Z,TTYIN
F046    DB F0           IN A,(SIOADAT)
F048    C9              RET

F049                    CI1:
F049    C3 F0EF         JP USERCI       ;DORT SPRUNG MOEGLICH
                        ;+++ RI ROUTINE +++

F04C                    RI:
F04C    3A F101         LD A,(IOBYT)
F04F    E6 0C           AND 000011000B
F051    CA F039         JP Z,CI         ;AR=T
F054    FE 04           CP 00000100B
F056    C2 F0F2         JP NZ,USERRI
F059                    LXLPI:          ;AR=P
F059    DB F3           IN A,(SIOBSTS)
F05B    E6 01           AND 1
F05D    28 FA           JR Z,LXLPI
F05F    DB F2           IN A,(SIOBDAT)
F061    C9              RET
                        ;

                        ; +++ MAIN CO ROUTINE +++ MOD
                        ;
F062                    CO:
F062    3A F101         LD A,(IOBYT)
F065    E6 03           AND 3
F067    20 0A           JR NZ,CO1
                        ;
F069                    TTYOUT:
F069    DB F1           IN A,(SIOASTS)
F06B    E6 04           AND 4
F06D    28 FA           JR Z,TTYOUT
F06F    79              LD A,C
F070    D3 F0           OUT (SIOADAT),A
F072    C9              RET
                        ;
F073                    CO1:
F073    C3 F0F5         JP USERCO       ;DORT SPRUNG MOEGLICH
                        ;
                        ;
                        ;
F076                    POO:    ;+++ MAIN POO ROUTINE +++
                                ;HIER ABFRAGE IOBYT MOEGLICH
F076    3A F101         LD A,(IOBYT)
F079    E6 30           AND 001100000B
F07B    CA F062         JP Z,CO         ;AP=T
F07E    FE 10           CP 000100000B
F080    C2 F0F8         JP NZ,USERPOO

F083                    LXLPO:          ;AP=P
F083    DB F3           IN A,(SIOBSTS)
F085    E6 04           AND 4
F087    28 FA           JR Z,LXLPO
F089    79              LD A,C
F08A    D3 F2           OUT (SIOBDAT),A
F08C    C9              RET
                        ;
                        ;
                        ; +++ MAIN LO ROUTINE +++ MOD

F08D                    LO:
F08D    3A F101         LD A,(IOBYT)
F090    E6 C0           AND 0C0H
F092    CA F069         JP Z,TTYOUT     ;AL=T
F095    FE 80           CP 100000000B   ;
F097    C2 F0FB         JP NZ,USERLO
                                        ;AL=L
F09A                    LO1:
F09A    DB F3           IN A,(SIOBSTS)
F09C    E6 04           AND 4
F09E    28 FA           JR Z,LO1
F0A0    79              LD A,C
F0A1    D3 F2           OUT (SIOBDAT),A
F0A3    C9              RET

                        ;
F0A4                    CSTS:   ;+++ MAIN STATUS ROUTINE +++
F0A4    3A F101         LD A,(IOBYT)
F0A7    E6 03           AND 3
F0A9    20 09           JR NZ,CSO
F0AB                    TTYCSTS:
F0AB    DB F1           IN A,(SIOASTS)
F0AD    E6 01           AND 1
F0AF    C8              RET Z
F0B0    3E FF           LD A,0FFH
F0B2    B7              OR A
F0B3    C9              RET
                        ;
F0B4                    CSO:
F0B4    C3 F0FE         JP USERCSTS     ;USER BEREICH
                        ;

                        ; Das IOBYTE dient der Umschaltung von
                        ; physikalischem nach logischem Geraet
                        ; Es hat die gleiche Bedeutung
                        ; wie im CP/M-Bios
                        ;
                        ;IOBYTE
F0B7                    IOBYTE:
F0B7    3A F101         LD A,(IOBYT)
F0BA    C9              RET

                        ; IOSET
                        ; kann z.B. im Bios gesetzt werden
F0BB                    IOSET:
F0BB    79              LD A,C
F0BC    32 F101         LD (IOBYT),A
F0BF    C9              RET

                        ;
F0C0                    MEMCK:  ;prueft den Speicherbereich und gibt
                                ;die hoechste verfuegbare Zelle an
F0C0    E5              PUSH HL
F0C1    CD F0CD         CALL MEMSIZ
F0C4    7D              LD A,L
F0C5    D6 3C           SUB 3CH
F0C7    30 01           JR NC,LXBZ
F0C9    25              DEC H
F0CA                    LXBZ:
F0CA    44              LD B,H
F0CB    E1              POP HL
F0CC    C9              RET

                        ;
                        ;
F0CD                    MEMSIZ: ;Unterprogramm fuer Speicherendebestimmung
F0CD    C5              PUSH BC
F0CE    21 FFFF         LD HL,-1
F0D1                    LXM0:
F0D1    24              INC H
F0D2    7E              LD A,(HL)
F0D3    2F              CPL
F0D4    77              LD (HL),A
F0D5    BE              CP (HL)
F0D6    2F              CPL
F0D7    77              LD (HL),A
F0D8    20 F7           JR NZ,LXM0
F0DA                    LXM1:
F0DA    24              INC H
F0DB    7C              LD A,H
F0DC    FE F0           CP 0F0H         ;ALL RAM
F0DE    28 08           JR Z,LXM2       ;READY 0F000H
F0E0    7E              LD A,(HL)       ;max 0f000h, da dort Monitor startet
F0E1    2F              CPL
F0E2    77              LD (HL),A
F0E3    BE              CP (HL)
F0E4    2F              CPL
F0E5    77              LD (HL),A
F0E6    28 F2           JR Z,LXM1
F0E8                    LXM2:
F0E8    25              DEC H
F0E9    01 FFDD         LD BC,EXIT-ENDX
F0EC    09              ADD HL,BC
F0ED    C1              POP BC
```

verschiedene Routinen. Danach kommen einige Titeleinträge in ASCIIs. Der eigentliche Beginn liegt bei F06B. Dort werden als erstes die beiden Ports der SIO passend initialisiert. In F097 wird mit LDA (7000H) die Freischaltung des RAM bewirkt. Mit der Initialisierung des IOBYTEs ist dann alles so eingestellt, daß nun die Kommandoschleife absolviert werden kann. Von der (hier nicht abgebildeten) Kommandoschleife aus können jetzt die verschiedensten Kommandos angewählt werden:

A⟨log. Gerät⟩ = ⟨phys. Gerät⟩

Nach Tippen des Buchstaben A erwartet der Monitor einen der Buchstaben C (für Console), R (für Reader), P (für Punch), L (für List-Device). Diese Symbole legen den logischen Gebrauch fest, den sie dem nach dem „=" folgenden physikalischen Gerät zuschreiben wollen. Dabei sind folgende Wahlmöglichkeiten, je nach Einstellung des logischen Gebrauches, möglich:

Nach AC= kann T für TTY, V für Video, B für Batch und U für „Eigenbau" folgen.

Nach AR= kann T für TTY, P für Punch-Read, C für Cassette und U für „Eigenbau" folgen.

Nach AP= kann T für TTY, P für Punch-Read, C für Cassette und U für „Eigenbau" folgen.

Nach AL= kann T für TTY, V für Video, L für Zeilendrucker und wieder U für „Eigenbau" folgen.

Wenn der Monitor anläuft sind die Einstellungen AC=T, AR=T, AP=T und AL=T vorgegeben. Das Teletype-Terminal übernimmt also alle Funktionen, es ist die Steuerkonsole, das Lesegerät, das Ausgabegerät und der Protokoll-Drucker in einem. Grob gesagt, beeinflussen Sie mit dem A-Kommando das IOBYTE. Wenn dann später ein anderes Kommando etwas von der Peripherie lesen oder in die Peripherie etwas schreiben will, dann wird das IOBYTE ausgewertet, und entsprechend seiner Einstellung werden die Daten formatiert und über den zugehörigen physikalischen Kanal geleitet. Mit dem Kommando AP=P schaffen Sie sich zum Beispiel einen Kanal, über den Sie vom 2. SIO-Kanal B lesen können. Wenn Sie A verwenden, dann hat das nur einen Sinn, wenn eine passende Routine existiert, die dann automatisch angesprungen wird. Wo diese Routine hingehört und wie der Ansprung funktioniert, das wird bei der Beschreibung der Systemroutinen noch gezeigt.

B

Eingabetastatur blockieren. Kann mit CTRL N (Code 1EH) wieder freigegeben werden.

```
F0EE   C9              RET
                       ;
F0EF                   TABSTART:              ;TABELLE SPRUNG-VEKTOREN
                       ;
F0EF   C3 F040         USERCI:        JP TTYIN
F0F2   C3 F040         USERRI:        JP TTYIN
F0F5   C3 F069         USERCO:        JP TTYOUT
F0F8   C3 F069         USERPOO:       JP TTYOUT
F0FB   C3 F069         USERLO:        JP TTYOUT
F0FE   C3 F0A8         USERCSTS:      JP TTYCSTS
                       ;
F101   00              IOBYT:  DEFB 0          ;IOBYTE-SPEICHER
F102                   OEXEC:  DEFS 3
F105                   IEXEC:  DEFS 3
F108                   KEXEC:  DEFS 3
                       ;
F539                   FREEMEM EQU $   ;dynamischer Platz fuer Buffer etc.

                       ;****************************************
                       ;* Bereich kann geloescht werden unter CP/M *
                       ;* um Platz fuer Puffer etc zu bieten       *
                       ;****************************************
                       ;
                       ;
F539                   MSG:
F539   0D 0A           DEFB 0DH,0AH
F53B   07              DEFB 7 ; BELL
F53C   20 4D 43 20     DEFM " MC COMPUTER"
F540   43 4F 4D 50
F544   55 54 45 52
F548   20 56 33 2E     DEFM " V3.4 "
F54C   34 20
F54E   20 52 44 4B     DEFM " RDK 1982 "
F552   20 31 39 38
F556   32 20
F558   0D 0A 00        DEFM 0DH,0AH,0
0022                   MSGL    EQU     $-MSG


F558                   BEGIN1:         ;nur einmal verwendet
                       ;    ------ INIT ------
F558   0E F1           LD C,SIOASTS
F55D   06 08           LD B,8
F55F   21 F56F         LD HL,TABSIO
F562   ED B3           OTIR
F564   0E F3           LD C,SIOBSTS
F566   06 08           LD B,8
F568   21 F577         LD HL,TABSIB
F56B   ED B3           OTIR
F56D   18 10           JR SKSKL
F56F                   TABSIO:
F56F   01 00           DEFB 1,0
F571   03 E1           DEFB 3,11100001B        ; -CTS UND -DCD ENABLE rev 3.4 term1
F573   04 4C           DEFB 4,01001100B
F575   05 EA           DEFB 5,11101010B        ;dtr rts +12V bedeutet ready rev 3.4
                       ;
F577                   TABSIB:                 ;-CTS UND -DCD ENABLE
F577   01 00           DEFB 1,0
F579   03 E1           DEFB 3,11100001B
F57B   04 4C           DEFB 4,01001100B
F57D   05 EA           DEFB 5,11101010B        ;dtr rts +12V bedeutet ready rev 3.4
                       ;
F57F                   SKSKL:

                       ; pio nicht mehr noetig mit 3.2
                       ;
F57F   3A 7000         LD A,(7000H)    ;Bank-Select umschalten
                       ;


F582   AF              XOR A           ;alles auf Console schalten
F583   32 F101         LD (IOBYT),A

                       ;    ------ END INIT -------

F586   31 F58A         LD SP,AHEAD-4
F589   C3 F0CE         JP MEMSIZ+1
F58C   F58E            DEFW AHEAD
F58E                   AHEAD:
F58E   F9              LD SP,HL
F58F   EB              EX DE,HL
F590   01 0023         LD BC,ENDX-EXIT
F593   21 FC3C         LD HL,EXIT
F596   ED B0           LDIR
F598   EB              EX DE,HL
F599   01 FFA1         LD BC,-5FH
F59C   09              ADD HL,BC
F59D   E5              PUSH HL
F59E   21 0000         LD HL,0
F5A1   06 0A           LD B,10 ;REGISTER
F5A3                   STKIT:
F5A3   E5              PUSH HL
F5A4   10 FD           DJNZ STKIT
F5A6                   HELLO:
F5A6   0E 1A           LD C,1AH
F5A8   CD F062         CALL CO
F5AB   06 22           LD B,MSGL
F5AD   CD F9D9         CALL TOM1
                       ;       Haupteingabeschleife
F5B0                   START:
F5B0   11 F5B0         LD DE,START
F5B3   D5              PUSH  DE
F5B4   CD FA4E         CALL CRLF
F5B7   0E 3E           LD C,'>'
F5B9   CD F062         CALL CO
                       ; ------    U . S . W .
```

C⟨anfadr⟩,⟨endadr⟩
Vergleichen der Eingabe über die Reader-Schnittstelle mit dem Speicher. Beispiel:
C100, 200
D⟨anfadr⟩,⟨endadr⟩
Ausgabe eines Speicherbereichs in Hex-Code. Die Ausgabe kann durch CTRL-C gestoppt werden. Beispiel:
D100, 2000
E[⟨endadr⟩]
Damit wird das Intel-Hexformat EOF erzeugt und über den Punch-Kanal ausgegeben. Eine zusätzliche Adresse kann dabei als Parameter mit angegeben werden ([] bedeutet optional).
F⟨anfadr⟩,⟨endadr⟩,⟨wert⟩
Ein Speicherbereich kann mit einem konstanten Wert gefüllt werden. Beispiel:
F100, E000, 0
G⟨startadr⟩,[⟨break1⟩],[⟨break2⟩]
Start eines Anwenderprogramms. Es können bis zu zwei Breakpoint-Adressen mit angegeben werden.
H⟨wert1⟩,⟨wert2⟩
Hex-Arithmetik. Es werden Summe und Differenz berechnet und ausgegeben.
J⟨anfadr⟩,⟨endadr⟩
Test eines Speicherbereiches. Dabei wird ein nicht zerstörender Schnelltest durchgeführt.
L⟨anfadr⟩
Laden binär vom Reader-Kanal. Der Befehl ist das Gegenstück zu dem U-Befehl. Je 8 Bit bilden eine Einheit.
M⟨anfadr⟩,⟨endadr⟩,⟨zieladr⟩
Transportbefehl. Ein Datenblock wird, beginnend mit anfadr, nach zieladr verschoben.
N
Ausgabe von NULL-Zeichen über den Punch-Kanal z. B. zur Synchronisation eines Kassetten-Interface.
P⟨adr⟩
Eingabe von ASCII-Zeichen in den Speicher beginnend bei adr. Mit dem Zeichen „_" kann das letzte Zeichen gelöscht werden. Mit CTRL-D wird die Eingabe beendet und die letzte Adresse vom Monitor ausgegeben.
QI⟨portadr⟩ oder QO⟨portadr⟩,⟨wert⟩
Es lassen sich die I/O-Ports mit dem QI Befehl abfragen und mit QO auf einen Wert setzen.
R[⟨bias⟩],[⟨reladr⟩]
Mit R kann ein Intel-Hexfile über den Punch-Kanal eingelesen werden. Wird ein Bias (Versatz) angegeben, so wird dieser Wert zu dem im Format angegebenen addiert. Mit reladr können im TDL-Relokating-Format geschriebene Programme verschoben werden, siehe auch [1].
S⟨adr⟩

```
                        ;
F5C5                    CSTS:   ;+++ MAIN STATUS ROUTINE +++
F5C5    3A F87D         LD A,(IOBYT)
F5C8    E6 03           AND 3
F5CA    20 09           JR NZ,CS0
F5CC                    TTYCSTS:
F5CC    DB F1           IN A,(SIOASTS)
F5CE    E6 01           AND 1
F5D0    C8              RET Z
F5D1    3E FF           LD A,0FFH
F5D3    B7              OR A
F5D4    C9              RET
                        ;
F5D5                    CS0:
F5D5    C3 F87A         JP USERCSTS     ;USER BEREICH

                        ;
```

Bild 2. Das Unterprogramm CSTS

```
; IOBYTE ASSIGNE
; 00000XX CONSOLE
; 0000XX00 READER
; 00XX0000 PUNCH
; XX000000 LISTER

; 00= TELETYPE CONS
```

Bild 3. IOBYTE ist der Name einer Speicherzelle, die dem System signalisiert, welcher I/O-Kanal bedient werden soll

Modifizieren von Speicherzellen. Mit Leerzeichen kann zur nächsten Speicherzelle vorangeschritten werden, mit „-" kann die letzte Zelle angesprochen werden und durch Eingabe eines sedezimalen Wertes wird dieser dort abgelegt. Mit CR wird die Eingabe abgeschlossen.
T⟨anfadr⟩,⟨endadr⟩
Ausgabe eines Speicherbereichs in ASCII-Darstellung falls möglich, sonst Ausgabe eines Punktes.
U⟨anfadr⟩,⟨endadr⟩
Binärausgabe über den Punch-Kanal. Ist mit L wieder lesbar.
V⟨anfadr⟩,⟨endadr⟩,⟨zieladr⟩
Vergleich eines Speicherbereichs von anfadr bis endadr mit zieladr. Unterschiede werden auf der Konsole ausgegeben.

W⟨anfadr⟩,⟨endadr⟩
Ausgabe eines Speicherbereichs im Intel-Hex-Format über den Punch-Kanal. Kann mit R wieder geladen werden.
X[⟨register⟩⟨neuer wert⟩]
Registerausgabe durch Angabe von X. Wird der Registername zusätzlich angegeben, so kann dieses modifiziert werden.
Y⟨wert⟩,[⟨wert⟩],[...]
Suchen nach einer Zeichenfolge von maximal 255 Bytes. Beispiel: Y10, 20, 30 sucht nach den Hex-Werten 10, 20, 30.

Z
Gibt die höchste Speicheradresse aus, hier den Wert EFFF, wenn alles ok ist.

I
Floppy-Boot-Start. Es meldet sich der Monitor mit einer Auswahl von verschiedenen Optionen. Nach Eingabe eines geeigneten Wertes wird das CP/M-System von der Floppy gebootet.
Die Floppy-Schnittstelle wird in einem anderen Abschnitt des Heftes ausführlich beschrieben.
Die Monitorkommandos benutzen bei der Ein- oder Ausgabe von Daten einige Unterprogramme, die auch von einem Benutzer-Programm her angesprungen werden können. Gleich zu Beginn des Monitors steht eine Sprungtabelle, die sowohl den Standort der Routine zeigt, als auch deren Namen.

CSTS
(Consol-Status) Diese Routine überprüft, ob vom angeschlossenen Terminal ein Zeichen abgesendet wurde. Hinterläßt sie im Akku den Wert 0, dann wurde seit der letzten Zeichenübernahme kein neues Zeichen abgesendet. Hinterläßt die Routine den Wert FF im Akku, dann wurde ein neues Zeichen abgesendet, das nun auf die Übernahme wartet.
Bild 2 zeigt das Unterprogramm. Es überprüft zunächst das IOBYTE. Hatten Sie dort zum Beispiel mit dem Kommando A oder aus einem eigenen Programm heraus eine andere Einstellung eingetragen, dann hüpft der Computer nach F5D5 und von dort nach F87A, von wo Sie Ihre eigene Routine dann anspringen müssen, die Ihre eigene Konsol-Schnittstelle dann abfragen muß, ob ein Zeichen ansteht. War das nicht der Fall, dann überprüft der Computer das Statusregister der SIO auf der SIO-PIO-Karte und kehrt entweder mit Akku = 0

zurück oder lädt FF und kehrt mit diesem Wert ins Hauptprogramm zurück. *Bild 3* zeigt die möglichen Belegungen des IOBYTE, die sich einfach historisch einmal entwickelt haben. Aus der Auswertung dieses Bytes im CSTS-Unterprogramm ergibt sich, daß im mc-Monitor keine Vorkehrung getroffen wurde, die Konsole umzusteuern. Das A-Kommando sollte man also tunlichst nicht auf die Konsole anwenden. Erfahrene CP/M-Profis wissen aber, daß sie mit wenigen „Handgriffen" eine Routine in das System integrieren können, die zum Beispiel eine Parallel-Schnittstelle so bedient, als wäre dort ein Terminal angeschlossen, und zwar ohne daß die Monitorkommandos umgeschrieben werden müssen.

CI

CI liest ein Zeichen von der Konsole ins Register A. Auch hier wird wieder das IOBYTE bestimmt, um zu entscheiden, ob eine Benutzerroutine (für eine andere Konfiguration) angesprungen werden soll. Wenn nicht, dann wird geprüft, ob überhaupt ein Zeichen ansteht. Wenn keins ansteht, dann wird zurückgesprungen und in der Abfrageschleife so lange gewartet, bis ein Zeichen vom Terminal angekommen ist. Dieses wird dann vom Datenregister der SIO her in den Akku übernommen. Danach wird zurückgesprungen.

CO

Die Konsol-Ausgabe-Routine überprüft IOBYTE, danach den Status der SIO, ob nämlich das angeschlossene Terminal ein Zeichen empfangen kann, und gibt dieses ggf. aus. CO erwartet das auszugebende Zeichen im Register C des Prozessors.

RI

Diese Routine liest ein Zeichen von der vorgesehenen „Einlese-Schnittstelle". Ist IOBYTE vom Benutzer zum Beispiel mit AP=T eingestellt worden, dann liest RI von der Konsol-Schnittstelle (Sprung nach CI). Entspricht IOBYTE der Einstellung AP=P, dann wird vom SIO-B-Kanal gelesen. Das Zeichen befindet sich im Akku.

POO

Die Routine gibt ein Zeichen, das sich in Register C befinden muß, über die mit IOBYTE eingestellte Schnittstelle aus. Entspricht IOBYTE der Einstellung AP=T, dann wird über die SIO, Kanal A ausgegeben (Sprung nach CO). Entspricht IOBYTE der Einstellung AP=P, dann wird über Kanal B ausgegeben.

```
rom  abs                                                        checksum
0000 EFE3 31 FF FF 3E C9 32 00 F0 CD 00 F0 3B 3B                += 068B
000D EFF0 D1 21 12 00 19 11 00 F0 01 FF 0F ED B0 C3 00 F0       += 067D
001D F000 C3 36 F0 C3 39 F0 C3 4C F0 C3 62 F0 C3 76 F0 C3       += 0AD5
002D F010 8D F0 C3 A4 F0 C3 B7 F0 C3 BB F0 C3 C0 F0 C3 33       += 0C15
003D F020 F0 C3 33 F0 C3 59 F1 C3 20 F1 C3 33 F0 EF F0 91       += 0B0D
004D F030 FC 39 F5 C3 52 FB C3 5B F5 3A 01 F1 E6 03 20 09       += 088B
005D F040 DB F1 E6 01 28 FA DB F0 C9 C3 EF F0 3A 01 F1 E6       += 0B1D
006D F050 0C CA 39 F0 FE 04 C2 F2 F0 DB F3 E6 01 28 FA DB       += 0A57
007D F060 F2 C9 3A 01 F1 E6 03 20 0A DB F1 E6 04 28 FA 79       += 0848
008D F070 D3 F0 C9 C3 F5 F0 3A 01 F1 E6 30 CA 62 F0 FE 10       += 0AA0
009D F080 C2 F8 F0 DB F3 E6 04 28 FA 79 D3 F2 C9 3A 01 F1       += 0AB7
00AD F090 E6 C0 CA 69 F0 FE 80 C2 FB F0 DB F3 E6 04 28 FA       += 0BCE
00BD F0A0 79 D3 F2 C9 3A 01 F1 E6 03 20 09 DB F1 E6 01 C8       += 08C0
00CD F0B0 3E FF B7 C9 C3 FE F0 3A 01 F1 C9 79 32 01 F1 C9       += 09C9
00DD F0C0 E5 CD CD F0 7D D6 3C 30 01 25 44 E1 C9 C5 21 FF       += 0927
00ED F0D0 FF 24 7E 2F 77 BE 2F 77 20 F7 24 7C FE F0 28 08       += 0780
00FD F0E0 7E 2F 77 BE 2F 77 28 F2 25 01 DD FF 09 C1 C9 C3       += 07FA
010D F0F0 40 F0 C3 40 F0 C3 69 F0 C3 69 F0 C3 69 F0 C3 AB       += 0AE5
011D F100 F0 00 1B 3E 00 47 32 B6 3D 3C 32 F1 DB 40 FB C9       += 06F3
012D F110 F1 DB 30 FB C9 3E 03 32 23 F5 3E 03 32 24 F5 C9       += 07A0
013D F120 3E D0 D3 40 D3 30 3A 10 00 32 1E F5 3E C3 32 10       += 05F6
014D F130 00 E5 2A 11 00 22 1F F5 21 10 F1 22 11 00 E1 DB       += 0567
015D F140 40 DB 30 FB CD 7E F2 F3 F5 3A 1E F5 32 10 00 E5       += 08DF
016D F150 2A 1F F5 22 11 00 E1 F1 C9 3E D0 D3 40 D3 30 3A       += 076A
017D F160 10 00 32 1E F5 3E C3 32 10 00 E5 2A 11 00 22 1F       += 03F9
018D F170 F5 21 0B F1 22 11 00 E1 DB 40 DB 30 FB CD 8C F2       += 0892
019D F180 F3 F5 3A 1E F5 32 10 00 E5 2A 1F F5 22 11 00 E1       += 06AE
01AD F190 F1 C9 DB 44 E6 20 20 03 3E 04 C9 AF C9 DB 34 E6       += 087A
01BD F1A0 20 20 F8 3E 04 C9 7B D3 42 59 0E 43 CD 92 F1 C6       += 0793
01CD F1B0 88 47 3A 25 F5 B0 D3 40 7B D3 44 DB 44 07 D2 BB       += 082B
01DD F1C0 F1 ED A2 C3 BB F1 7B D3 42 59 0E 43 CD 92 F1 C6       += 0A3F
01ED F1D0 A8 47 3A 25 F5 B0 D3 40 7B D3 44 DB 44 07 D2 DB       += 086B
01FD F1E0 F1 ED A3 C3 DB F1 C5 01 F4 01 0B 78 B1 C2 EA F1       += 0A9C
020D F1F0 C1 C9 CD E6 F1 79 D3 44 7B D3 42 7A D3 43 06 1C       += 0900
021D F200 3A 23 F5 B0 D3 40 76 CD E6 F1 79 E6 0F D3 44 06       += 08BA
022D F210 0C 3A 23 F5 B0 D3 40 76 7B D3 32 59 0E 33 CD 9D       += 071B
023D F220 F1 C6 88 47 3A 25 F5 B0 D3 30 7B D3 34 DB 34 07       += 0825
024D F230 D2 2D F2 ED A2 C3 2D F2 7B D3 32 59 0E 33 CD 9D       += 08E6
025D F240 F1 C6 A8 47 3A 25 F5 B0 D3 30 7B D3 34 DB 34 07       += 0845
026D F250 D2 4D F2 ED A3 C3 4D F2 CD E6 F1 79 D3 34 7B D3       += 0B15
027D F260 32 7A D3 33 06 1C 3A 24 F5 B0 D3 30 76 CD E6 F1       += 07F4
028D F270 79 E6 0F D3 34 06 0C 3A 24 F5 B0 D3 30 76 CB F9       += 07C7
029D F280 78 B7 20 16 7A E6 03 32 24 F5 AF C9 CB B9 78 B7       += 083E
02AD F290 20 08 7A E6 03 32 23 F5 AF C9 AF 32 26 F5 32 25       += 06A0
02BD F2A0 F5 79 E6 C0 0F 0F 32 22 F5 79 E6 30 20 24 79 E6       += 07AD
02CD F2B0 03 D9 21 CA F2 5F 16 00 19 19 7E 32 21 F5 23 5E       += 05A7
02DD F2C0 3A 22 F5 B3 32 22 F5 D9 18 49 00 01 02 02 01 04       += 0491
02ED F2D0 03 08 79 E6 30 FE 10 20 27 79 E6 07 32 21 F5 0F       += 05AC
02FD F2E0 30 05 3E 02 32 25 F5 79 0F E6 03 D9 21 CA F2 5F       += 0647
rom  abs                                                        checksum
030D F2F0 16 00 19 19 23 5E 3A 22 F5 B3 D9 32 22 F5 18 13       += 051A
031D F300 79 E6 07 32 21 F5 0F 30 DE 3A 22 F5 F6 08 32 22       += 066E
032D F310 F5 18 D4 79 CB 7F C2 1C F4 3A 21 F5 4F 3A 27 F5       += 086B
033D F320 B9 CA 73 F3 E5 C5 21 28 F5 3A 27 F5 FE FF 28 0A       += 0956
034D F330 0F E6 03 4F 06 00 09 DB 41 77 C1 E1 E5 C5 21 28       += 067E
035D F340 F5 79 0F E6 03 4F 06 00 09 7E D3 41 C1 E1 79 32       += 06A3
036D F350 27 F5 F5 C5 C5 01 FA 00 0B 78 B1 C2 58 F3 C1 3A       += 08D2
037D F360 22 F5 E6 0F D3 44 01 65 0B 0B 78 B1 C2 69 F3 C1       += 07A7
038D F370 F1 18 06 DB 44 E6 20 20 0D 3A 22 F5 4F CD 11 F4       += 06D3
039D F380 DB 40 E6 80 20 F3 DB 41 FE FF 28 15 BA CA BE F3       += 0A1F
03AD F390 3A 22 F5 4F E5 D5 C5 CD F2 F1 C1 D1 E1 E6 90 28       += 0AE0
03BD F3A0 1D 3A 22 F5 4F E5 D5 C5 CD 07 F2 C1 D1 E1 3A 26       += 08D5
03CD F3B0 F5 3C 32 26 F5 FE 0A 38 D7 3E FF B7 37 C9 78 FE       += 08FF
03DD F3C0 02 CA EC F3 3A 22 F5 4F E5 D5 C5 CD A6 F1 C1 D1       += 0AC0
03ED F3D0 E1 E6 9C C8 E6 10 20 C9 3A 26 F5 3C 32 26 F5 FE       += 08E6
03FD F3E0 05 38 DB FE 0A 38 BA 3E FF B7 37 C9 3A 22 F5 4F       += 07A6
040D F3F0 E5 D5 C5 CD C6 F1 C1 D1 E1 E6 FC C8 E6 10 20 A1       += 0BD7
041D F400 3A 26 F5 3C 32 26 F5 FE 04 DA A1 F3 3E FF B7 37       += 0879
042D F410 C9 E5 D5 C5 CD F2 F1 C1 D1 E1 C9 C9 3A 21 F5 4F       += 089C
043D F420 3A 30 F5 B9 CA 76 F4 E5 C5 21 31 F5 3A 30 F5 FE       += 099A
044D F430 FF 28 0A 0F E6 03 4F 06 00 09 DB 31 77 C1 E1 E5       += 0691
045D F440 C5 21 31 F5 79 0F E6 03 4F 06 00 09 7E D3 31 C1       += 061E
046D F450 E1 79 32 30 F5 F5 C5 C5 01 FA 00 0B 78 B1 C2 5B       += 087C
047D F460 F4 C1 3A 22 F5 E6 0F D3 34 01 65 0B 0B 78 B1 C2       += 0769
048D F470 6C F4 C1 F1 18 06 DB 34 E6 20 20 0D 3A 22 F5 4F       += 0712
049D F480 CD 14 F5 DB 30 E6 80 20 F3 DB 31 FE FF 28 15 BA       += 095A
04AD F490 CA C1 F4 3A 22 F5 4F E5 D5 C5 CD 58 F2 C1 D1 E1       += 0B28
04BD F4A0 E6 90 28 1D 3A 22 F5 4F E5 D5 C5 CD 6D F2 C1 D1       += 0998
04CD F4B0 E1 3A 26 F5 3C 32 26 F5 FE 0A 38 D7 3E FF B7 37       += 0801
04DD F4C0 C9 78 FE 02 CA EF F4 3A 22 F5 4F E5 D5 C5 CD 18       += 09F2
04ED F4D0 F2 C1 D1 E1 E6 9C C8 E6 10 20 C9 3A 26 F5 3C 32       += 0951
04FD F4E0 26 F5 FE 05 38 DB FE 0A 38 BA 3E FF B7 37 C9 3A       += 0859
050D F4F0 22 F5 4F E5 D5 C5 CD 38 F2 C1 D1 E1 E6 FC C8 E6       += 0BDF
051D F500 10 20 A1 3A 26 F5 3C 32 26 F5 FE 04 DA A4 F4 3E       += 0761
052D F510 FF B7 37 C9 E5 D5 C5 CD 58 F2 C1 D1 E1 C9 00 00       += 0A88
053D F520 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF FF       += 08F7
054D F530 FF FF FF FF FF FF FF FF FF 0D 0A 07 20 4D 43 20       += 09E5
055D F540 43 4F 4D 50 55 54 45 52 20 56 33 2E 34 20 20 52       += 040C
056D F550 44 4B 20 31 39 38 32 20 0D 0A 00 0E F1 06 08 21       += 02E8
057D F560 6F F5 ED B3 0E F3 06 08 21 77 F5 ED B3 18 10 01       += 0769
058D F570 00 03 E1 04 4C 05 EA 01 00 03 E1 04 4C 05 EA 3A       += 0481
059D F580 00 70 AF 32 01 F1 31 8A F5 C3 CE F0 8E F5 F9 EB       += 09DB
05AD F590 01 23 00 21 3C FC ED B0 EB 01 A1 FF 09 E5 21 00       += 06B5
05BD F5A0 00 06 0A E5 10 FD 0E 1A CD 62 F0 06 22 CD D9 F9       += 0710
05CD F5B0 11 B0 F5 D5 CD 4E FA 0E 3E CD 62 F0 CD CA FB E6       += 0A83
05DD F5C0 7F 28 F9 D6 41 F8 FE 1A D0 87 21 DA F5 85 6F 7C       += 097E
05ED F5D0 CE 00 67 7E 23 66 6F 0E 02 E9 63 F6 AE F6 DB F6       += 0872
05FD F5E0 FC F6 13 F7 2F F7 3C F7 A0 FA 1E F6 8A F7 08 F1       += 0A7D
060D F5F0 17 FB A8 F7 3B FA 02 F1 BC F6 EB FB B3 F7 6C F8       += 0A7F
rom  abs                                                        checksum
061D F600 95 F8 26 FA 16 FC FD F8 3A F9 B5 F8 0A FA 0D 0A       += 09AF
062D F610 38 20 3D 31 20 35 20 31 2F 34 3D 32 20 3A CD 15       += 037A
063D F620 F1 21 0E F6 06 10 CD DC F9 CD 39 F0 E6 7F 4F CD       += 0945
064D F630 62 F0 FE 49 CA 05 F1 FE 32 28 10 FE 31 CA 43 F6       += 08F3
065D F640 C3 EE F9 CD 57 F6 CD 59 F1 18 06 CD 57 F6 CD 20       += 0A00
```

4. Das Hex-Listing des Monitors. EPROM und Assemblerlisting gibt es beim Software-Service

```
066D F650  F1 DA EE F9 C3 80 00 21 80 00 16 00 1E 01 06 01   += 05D2
067D F660  0E 00 C9 CD CA FB 21 28 FC 01 00 04 11 05 00 BE   += 0587
068D F670  28 06 19 0C 10 F9 18 15 59 CD CA FB FE 3D 20 F9   += 06C8
069D F680  CD CA FB 01 00 04 23 BE 28 06 0C 10 F9 C3 EE F9   += 0765
06AD F690  3E 03 1C 1D 28 08 CB 21 CB 21 17 17 18 F5 2F 57   += 0443
06BD F6A0  CD 02 FB 30 FB 3A 01 F1 A2 B1 4F C3 BB F0 CD 4E   += 094C
06CD F6B0  FA CD C4 FB FE 0E 20 F9 D1 C3 A6 F5 CD 61 FA CD   += 0BCF
06DD F6C0  4E FA E1 CD C4 FB FE 04 CA 11 FA FE 5F 28 08 77   += 0990
06ED F6D0  4F 23 CD 62 F0 18 EC 2B 4E 18 F7 CD 49 FA CD 03   += 07FD
06FD F6E0  FA BE C4 EA F6 CD 90 FA 18 F4 47 CD 14 FA 7E CD   += 0B2C
070D F6F0  B1 FA CD 17 FA 78 CD B1 FA C3 4E FA CD 49 FA CD   += 0B61
071D F700  11 FA CD 17 FA 7E CD B1 FA CD 90 FA 7D E6 0F 20   += 09C8
072D F710  F1 18 EC CD 61 FA CD 1C FA 0E 3A CD 76 F0 AF CD   += 09F7
073D F720  E6 FA E1 CD E1 FA 21 00 00 CD E1 FA C3 3B FA CD   += 0AF7
074D F730  56 FA 71 CD 96 FA 30 FA D1 C3 B0 F5 CD 02 FB 38   += 0A83
075D F740  40 28 10 CD 88 FA D1 21 34 00 39 72 2B 73 78 FE   += 06AC
076D F750  0D 28 2E 16 02 21 35 00 39 E5 CD 61 FA 58 C1 E1   += 0611
077D F760  78 B1 28 0A 71 23 70 23 0A 77 23 3E FF 02 7B FE   += 05DE
078D F770  0D 28 03 15 20 E3 3E C3 32 38 00 21 1E F0 22 39   += 0445
079D F780  00 CD 4E FA D1 21 16 00 39 E9 CD 49 FA 7E 47 2F   += 0743
07AD F790  77 AE 28 0E D5 50 5F CD 14 FA CD FD FB CD 4E FA   += 0994
07BD F7A0  42 D1 70 CD 90 FA 18 E5 CD 56 FA 7E 02 03 CD 90   += 08D4
07CD F7B0  FA 18 F8 CD 61 FA 78 D6 0D 47 4F D1 28 04 CD 61   += 084E
07DD F7C0  FA C1 EB D9 CD 4E FA CD 11 FB D6 3A 47 E6 FE 20   += 0AC8
07ED F7D0  F6 57 CD 4D F8 5F CD 4D F8 F5 CD 4D F8 D9 D1 5F   += 0AE0
07FD F7E0  C5 D5 E5 19 E3 DD E1 D9 E1 CD 4D F8 3D 78 C1 20   += 0A9B
080D F7F0  03 09 DD 09 1C 1D 28 19 3D 28 22 CD 4D F8 CD 60   += 0532
081D F800  F8 20 F8 CD 4D F8 28 BF DD E5 E1 CD AC FA C3 EE   += 0BD0
082D F810  F9 7C B5 C8 EB 21 34 00 39 72 2B 73 C9 2E 01 CD   += 0740
083D F820  3D F8 38 07 CD 60 F8 20 F6 18 D8 4F CD 3D F8 47   += 0837
084D F830  D9 C5 D9 E3 D9 7D CD 60 F8 7C E1 18 E7 2D 20 07   += 08B5
085D F840  CD 4D F8 1D 67 2E 08 CD 4D F8 CB 24 C9 C5 CD CE   += 08F6
086D F850  FA 07 07 07 07 4F CD CE FA B1 4F 82 57 79 C1 C9   += 07D6
087D F860  DD 77 00 DD BE 00 20 A0 DD 23 1D C9 CD 61 FA E1   += 089E
088D F870  7E CD B1 FA CD FD FA D8 28 0F FE 5F 28 14 E5 CD   += 0A14
089D F880  88 FA D1 E1 73 78 FE 0D C8 23 7D E6 07 CC 11 FA   += 0956
08AD F890  18 DE 2B 18 F5 CD 49 FA CD 11 FA 06 40 7E E6 7F   += 083F
08BD F8A0  FE 20 30 02 3E 2E FE 7C 30 FA 4F CD 62 F0 CD 90   += 082B
08CD F8B0  FA 10 EA 18 E3 16 00 CD 61 FA E1 65 E5 33 14 78   += 0817
08DD F8C0  D6 0D 20 F3 47 4F 67 6A 2D 39 E5 C5 C5 CD 4E FA   += 0847
08ED F8D0  C1 E1 DD E1 5A DD 7E 00 ED B1 E2 F8 F8 DD E5 E5   += 0C2C
08FD F8E0  1D 28 0B DD 7E FF BE 20 E8 23 DD 2B 18 F2 E1 E5   += 086B
090D F8F0  2B C5 CD AC FA C1 18 D4 33 1D 20 FC C9 CD 49 FA   += 0955
091D F900  CD 1C FA 01 3A 00 CD 76 F0 D5 E5 D4 CD 96 FA 38   += 08A4
092D F910  24 3E 18 90 20 F5 E1 CD 1D F9 D1 18 E3 57 78 CD   += 084B
093D F920  E6 FA CD E1 FA AF CD E6 FA 7E CD E6 FA 23 10 F9   += 0C3B
094D F930  AF 92 C3 E6 FA E1 D1 AF 18 E3 CD CA FB 21 5F FC   += 0B4E
095D F940  FE 0D 28 5A FE 27 20 0A 21 78 FC CD CA FB FE 0D   += 080E
096D F950  28 4C BE 28 09 CB 7E C2 EE F9 23 23 18 F4 CD 17   += 078B
097D F960  FA 23 7E 47 E6 3F EB 6F 26 00 39 EB 23 1A CD B1   += 0766
098D F970  FA CB 78 28 05 1B 1A CD B1 FA CD FD FA D8 28 19   += 08F4
099D F980  E5 C5 CD 88 FA E1 F1 C5 F5 7D 12 E1 CB 78 28 03   += 0A43
09AD F990  13 7C 12 C1 E1 78 FE 0D C8 CB 7E C0 18 C3 CD 4E   += 088D
09BD F9A0  FA CD 17 FA 7E 23 B7 F8 4F CD 62 F0 0E 3D CD 62   += 0910
09CD F9B0  F0 7E 47 E6 3F 23 EB 6F 26 00 39 EB CB 7D 20 0F   += 070B
09DD F9C0  1A CD B1 FA CB 78 28 09 1B 1A CD B1 FA 18 D2 E5   += 0952
09ED F9D0  1A 67 1B 1A 6F 7E E1 18 F1 21 39 F5 4E 23 CD 62   += 067C
09FD F9E0  F0 10 F9 CD A4 F0 B7 C8 CD C4 FB FE 03 C0 CD CD   += 0BC0
0A0D F9F0  F0 11 EA FF 19 F9 0E 07 CD 62 F0 0E 2A CD 62 F0   += 0887
0A1D FA00  C3 B0 F5 CD 4C F0 38 E6 BA C9 CD CD F0 01 23 00   += 09C0
0A2D FA10  09 CD 4E FA CD AC FA 0E 20 C3 62 F0 0E 0D CD 76   += 0832
0A3D FA20  F0 0E 0A C3 76 F0 CD 49 FA CD C5 FA CD C0 FA 4E   += 0AA2
0A4D FA30  CD 76 F0 CD 96 FA 30 F7 CD C0 FA CD C5 FA C9 E6   += 0C79
0A5D FA40  0F C6 90 27 CE 40 27 4F C9 CD 63 FA D1 E1 E5 06   += 08A0
0A6D FA50  02 CD D9 F9 E1 C9 0C CD 63 FA CD 4E FA C1 D1 E1   += 0B09
0A7D FA60  C9 0E 01 21 00 00 CD CA FB 47 CD D1 FA 38 08 29   += 06D3
0A8D FA70  29 29 29 B5 6F 18 EF E3 E5 78 CD 05 FB 30 02 0D   += 06F2
0A9D FA80  C8 C2 EE F9 0D 20 DC C9 0E 01 21 00 00 C3 69 FA   += 0799
0AAD FA90  CD 96 FA D0 D1 C9 23 7C B5 37 C8 7B 95 7A 9C C9   += 0A09
0ABD FAA0  CD 49 FA E5 19 CD 14 FA E1 B7 ED 52 7C CD B1 FA   += 0AB4
0ACD FAB0  7D F5 0F 0F 0F 0F CD BA FA F1 CD 3F FA C3 62 F0   += 093B
0ADD FAC0  01 FF 08 18 03 01 00 48 CD 76 F0 10 FB C9 CD 11   += 0651
0AED FAD0  FB D6 30 D8 FE 17 3F D8 FE 0A 3F D0 D6 07 FE 0A   += 0901
0AFD FAE0  C9 7C CD E6 FA 7D F5 0F 0F 0F 0F CD 3F FA CD 76   += 08E9
0B0D FAF0  F0 F1 F5 CD 3F FA CD 76 F0 F1 82 57 C9 0E 2D CD   += 0AAA
0B1D FB00  62 F0 CD CA FB FE 20 C8 FE 2C C8 FE 0D 37 C8 3F   += 0A05
0B2D FB10  C9 CD 03 FA E6 7F C9 CD 61 FA E1 CD 4E FA 16 FF   += 0AF4
0B3D FB20  06 04 CD 03 FA 20 F9 10 F9 CD 03 FA 28 FB 77 3E   += 0798
0B4D FB30  07 23 CD 03 FA 28 03 77 18 F7 1E 01 CD 03 FA 20   += 05AE
0B5D FB40  09 1C 3E 07 BB 20 F5 C3 AC FA 72 23 1D 20 FB 77   += 06E7
0B6D FB50  18 DF E5 D5 C5 F5 CD CD F0 EB 21 0A 00 39 06 04   += 084E
0B7D FB60  EB 2B 72 2B 73 D1 10 F9 C1 0B F9 21 25 00 39 7E   += 06C2
0B8D FB70  91 23 20 04 7E 90 28 0C 23 23 7E 91 2D 05 23 7E   += 0435
0B9D FB80  90 28 01 03 21 20 00 39 73 23 72 23 23 71 23 70   += 0388
0BAD FB90  C5 0E 40 CD 62 F0 E1 CD AC FA 21 25 00 39 01 00   += 0706
0BBD FBA0  02 5E 71 23 56 71 23 7B B2 28 02 7E 12 23 10 F1   += 04E9
0BCD FBB0  08 D9 E5 D5 C5 F5 DD E5 FD E5 ED 57 47 ED 5F 4F   += 0B1F
0BDD FBC0  C5 C3 B0 F5 CD 39 F0 E6 7F C9 CD C4 FB C8 3C F8   += 0BD9
0BED FBD0  3D FE 0D C8 FE 4E C8 FE 6E 28 0D C5 4F CD 62 F0   += 08F8
0BFD FBE0  79 C1 FE 40 D8 FE 7B DD E6 5F C9 CD CA FB FE 4F   += 0B86
0C0D FBF0  28 1C FE 49 C2 EE F9 CD 61 FA C1 ED 58 06 08 CD   += 093D
0C1D FC00  17 FA CB 23 3E 18 8F 4F CD 62 F0 10 F5 C9 CD 63   += 0850
0C2D FC10  FA D1 C1 ED 59 C9 CD 56 FA 0A BE 28 05 C5 CD EA   += 0A29
0C3D FC20  F6 C1 03 CD 90 FA 18 F1 43 54 56 42 55 52 54 50   += 0794
0C4D FC30  43 55 50 54 50 43 55 4C 54 56 4C 55 C1 79 ED 4F   += 0631
0C5D FC40  78 ED 47 FD E1 DD E1 F1 C1 D1 E1 08 D9 D1 C1 F1   += 0C10
0C6D FC50  E1 F9 00 21 00 00 C3 00 00 00 00 00 00 00 00 41   += 02FF
0C7D FC60  15 42 13 43 12 44 11 45 10 46 14 48 31 4C 30 4D   += 0305
0C8D FC70  F1 50 B4 53 97 49 03 C1 41 09 42 0B 43 0A 44 00   += 0521
0C9D FC80  45 0C 46 08 48 0F 4C 0E 4D CF 58 87 59 85 52 02   += 047D
0CAD FC90  C1 00 C1 ED 59 C9 CD 56 FA 0A BE 28 05 C5 CD EA   += 091F
0CBD FCA0  F6 C1 03 CD 90 FA 18 F1 43 54 56 42 55 52 54 50   += 0794
0CCD FCB0  43 55 50 54 50 43 55 4C 54 56 4C 55 C1 79 ED 4F   += 0631
0CDD FCC0  78 ED 47 FD E1 DD E1 F1 C1 D1 E1 08 D9 D1 C1 F1   += 0C10
0CED FCD0  E1 F9 00 21 00 00 C3 00 00 00 00 00 00 00 00 41   += 02FF
0CFD FCE0  15 42 13
```

```
                        ;
                        ; BOOT FLOPPY HARD/SOFT/MINI
                        FLPMSG:
F60E
F60E    0D 0A           DEFB 0DH,0AH
F610    38 20 3D 31     DEFM '8 =1 5 1/4=2 :"
F614    20 35 20 31
F618    2F 34 3D 32
F61C    20 3A
0010                    FLPL    EQU $-FLPMSG
                        ;
F61E                    BOOT1:
F61E    CD F115         CALL INIFLP
F621    21 F60E         LD HL,FLPMSG
F624    06 10           LD B,FLPL       ;Meldung ausgeben
F626    CD F90C         CALL TOM        ;dann Antwort abwarten
F629    CD F039         CALL C1
F62C    E6 7F           AND 7FH
F62E    4F              LD C,A
F62F    CD F062         CALL CO
F632    FE 49           CP "I"          ;I-Vektor fuer Benutzerbefehl
F634    CA F105         JP Z,IEXEC
F637    FE 32           CP "2"          ;Mini Laufwerk Boot
F639    28 10           JR Z,M2EX
F63B    FE 31           CP "1"          ;Maxi Laufwerk Boot
F63D    CA F643         JP Z,MEXC
F640    C3 F9EE         JP ERROR
                        ;
F643                    MEXC:
F643    CD F657         CALL FLPSET
F646    CD F159         CALL MAXI       ;SINGLE DENSE BOOT 128 BYTES
F649    18 06           JR M3EX
                        ;
F64B                    M2EX:
F64B    CD F657         CALL FLPSET
F64E    CD F120         CALL MINI       ;SINGLE DENSE BOOT 128 BYTES
F651                    M3EX:
                                        ;AUCH BEI DOUBLE DENSE MINIFLP
                                        ;DA TRK0 IMMER SINGLE DENSE IST
F651    DA F9EE         JP C,ERROR      ;FEHLER
F654    C3 0080         JP 80H          ;UND STARTEN
                        ;
                        ;
F657                    FLPSET:
F657    21 0080         LD HL,80H       ;BOOTEN
F65A    16 00           LD D,0          ;TRK 0
F65C    1E 01           LD E,1          ;SEKTOR 1
F65E    06 01           LD B,1          ;LESEN
F660    0E 00           LD C,0          ;DRIVE 0
F662    C9              RET
```

Bild 6. Der „Boot-Teil" für Floppies

```
F5DA                    TBL:
F5DA    F663            DEFW ASSIGN
F5DC    F6AE            DEFW BYE
F5DE    F6D8            DEFW COMP
F5E0    F6FC            DEFW DISP
F5E2    F713            DEFW EOF
F5E4    F72F            DEFW FILL
F5E6    F73C            DEFW GOTO
F5E8    FAA0            DEFW HEXN
F5EA    F61E            DEFW BOOT1
F5EC    F78A            DEFW TEST
F5EE    F108            DEFW KEXEC
F5F0    FB17            DEFW LOAD
F5F2    F7A8            DEFW MOVE
F5F4    FA3B            DEFW NULL
F5F6    F102            DEFW OEXEC
F5F8    F6BC            DEFW PUTA
F5FA    FBEB            DEFW QUERY
F5FC    F783            DEFW READ
F5FE    F86C            DEFW SUBS
F600    F895            DEFW TYPE
F602    FA26            DEFW UNLO
F604    FC16            DEFW VERIFY
F606    F8FD            DEFW WRITE
F608    F93A            DEFW XAM
F60A    F8B5            DEFW WHERE
F60C    FA0A            DEFW SIZE
```

Bild 5. Die Befehls-Adreßliste

LO

LO (List Out) gibt ein Zeichen, das sich in Register C befindet, auf dem eingestellten Kanal aus. Entspricht IOBYTE der Einstellung AL=T, dann wird über SIO-Kanal A ausgegeben; bei der Einstellung AL=P wird über Kanal B ausgegeben.

Literatur

[1] *Klein, Rolf-Dieter:* Mikrocomputer Hard- und Softwarepraxis. Franzis-Verlag, München.

Rolf-Dieter Klein:

Der Floppy-Controller

Ein CP/M-Computer ist ohne Massenspeicher hilflos. Es ist ja gerade der Sinn eines CP/M-Computers, von einem kleinen Urladeprogramm her sich „hoch-zu-booten", bis ein flexibel reagierender Computer dasteht, der auf Kommando ein Basic-Computer wird – oder ein Fortran-Computer oder eine Abrechnungsmaschine oder was es sonst noch so gibt. Die heute geschilderte Platine steuert die Laufwerke, von welchen Sie Ihr CP/M einlesen können.

Für den kompletten CP/M-Computer fehlt nur noch die Floppy-Karte, die im folgenden besprochen wird. *Bild 1* zeigt die Blockschaltung. Als Floppy-Controller-Chip wurde der Typ 1797 gewählt. Er besitzt einige Vorteile bezüglich seiner Ansteuerung gegenüber anderen Typen. Der FD 1797 wurde als Controller ausgewählt, da er als einziger in der Lage ist, alle bekannten Formate zu lesen. Der FD 1793 kann das nicht, da das Format innerhalb des „Gaps" zwischen Sector-Identifier und Datenblock ein (genau ein!) bestimmtes Muster aufweisen muß (11 × FF, 6 × 00) und das Format 17 × 00, wie es von manchen Geräten produziert wird (IBM-Anlagen), für ihn nicht lesbar ist. Das gleiche gilt auch für die Bausteine FD 2793 und 2797, wobei hier im Datenblatt ausdrücklich auf den Unterschied hingewiesen wird.

Beim NEC 765 treten ähnliche Probleme auf, auch dort gibt es gebräuchliche Diskettenformate, die dieses IC nicht lesen kann.

Die Schaltung wurde mit einem Steuerport zum Selektieren der Drives und zum Einstellen des Aufzeichnungsverfahrens (FM, MFM oder MINI- und MAXI-Floppy) versehen. Es können direkt 4 Laufwerke betrieben werden, mit einem Decoder in den Laufwerken sogar bis zu 16 Drives. Über einen Status-Port können ein paar interne Informationen abgefragt werden, die zum Betrieb des Controllers nötig sind.

Der Controller arbeitet mit Interrupt. Dazu ist ein Bustreiber vorgesehen, der im Z80-Mode 0, der hier verwendet wird, einen Interrupt-Befehl an den Z80 geben kann. Mit einer Auto-Wait-Logik ist der Anschluß von artfremden langsameren CPU-Karten aus der ECB-Serie möglich, wobei dann auch bei 2 MHz 8-Zoll-Floppys mit FM betrieben werden können (dann müssen statische Speicher verwendet werden). Wir arbeiten mit 4 MHz bzw. 6 MHz (bei MFM Voraussetzung), dann ist die Auto-Wait-Schaltung nicht nötig. *Bild 2* zeigt den Gesamtschaltplan des Floppy-Controllers.

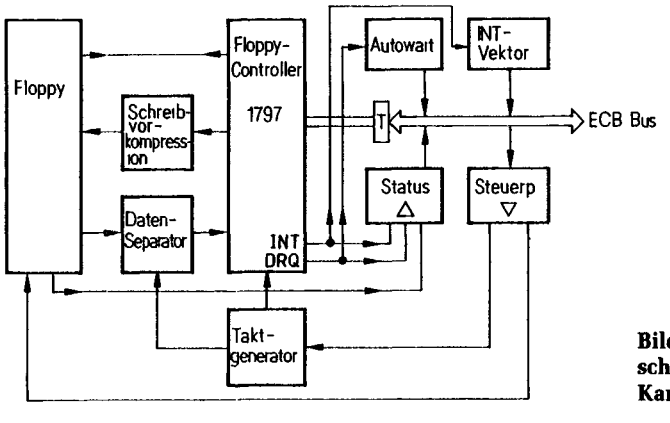


Bild 1. Die Blockschaltung der Floppy-Karte

Die Funktion

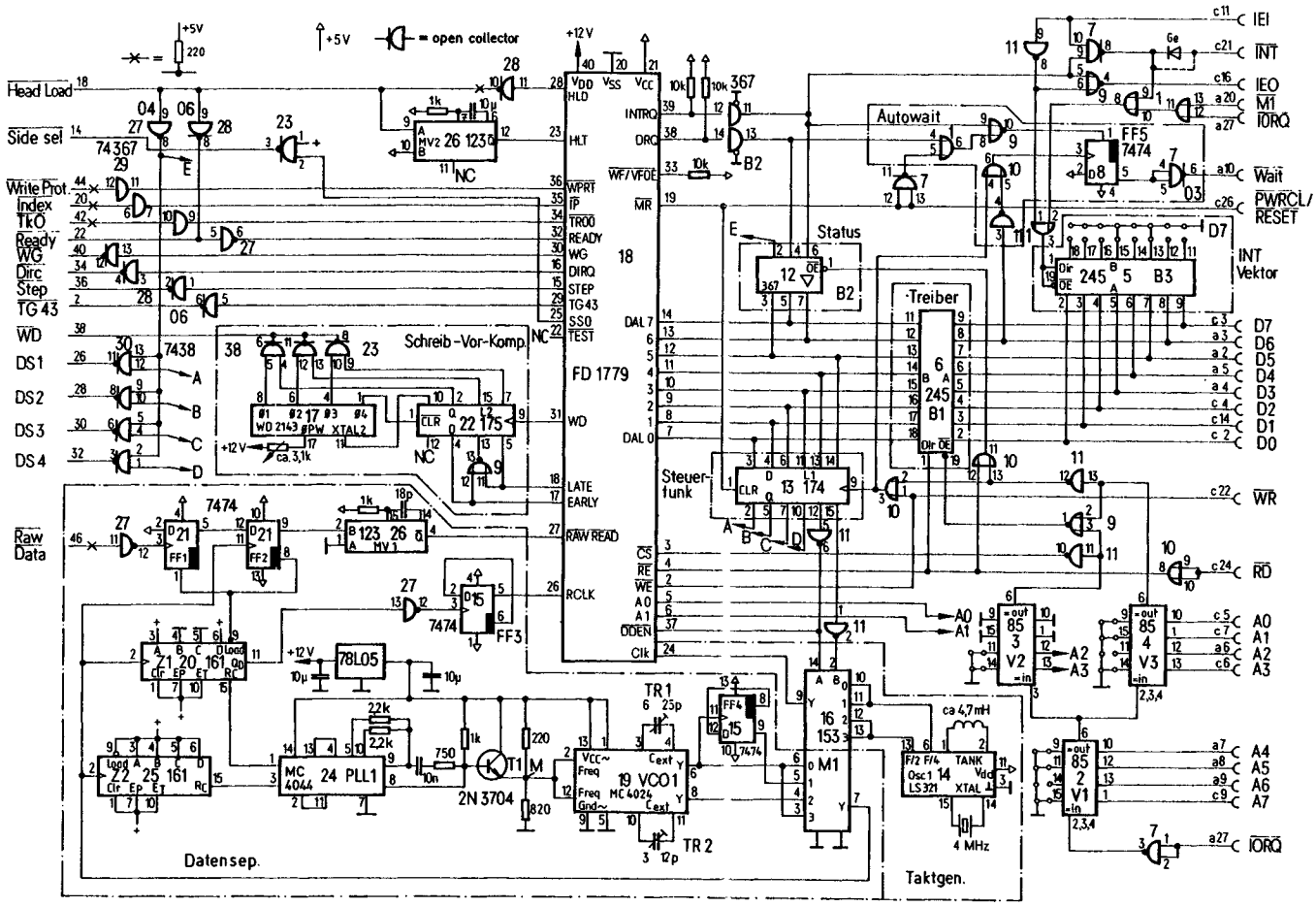
Mehrere ICs 7485 übernehmen die Adreßdecodierung. Der Floppy-Controller muß auf den Bereich 40H...43H gelegt werden. Status- und Steuerport werden auf die Adresse 44H gelegt. Dies ist mit Brücken auf der Floppy-Karte möglich. Die korrekte Stellung der Brücken an V_1, V_2, V_3 ist im Plan eingezeichnet. Ferner ist das Interrupt-Port B3, das der CPU den richtigen Befehl zur geforderten Aktion mitteilt, auf D7H einzustellen, was den RST2-Befehl darstellt. Das Interrupt-Programm befindet sich auf Adresse 10H. Als Interrupt-Mode wird der Z80-Mode 0 verwendet.

Der Floppy-Controller enthält einen recht aufwendigen Datenseparator, der eine hohe Sicherheit garantiert. Er muß, wie später noch gezeigt wird, mit zwei Trimmern abgeglichen werden. Dieser Abgleich ist kritisch und gelingt nur bei mit formatierter Diskette laufender Floppy. Eine weitere Einstellung muß bei der Datenprekompression vorgenommen werden. Dies geschieht mit dem Trimmer TR3, wobei bei FM-Aufzeichnung auch mit einem Festwiderstand von 3,1 kΩ gearbeitet werden kann.

Bild 3 zeigt die Lötseite der Leiterplatte und *Bild 4* zeigt die Bestückungsseite. In *Bild 5* ist der Bestückungsplan abgebildet.

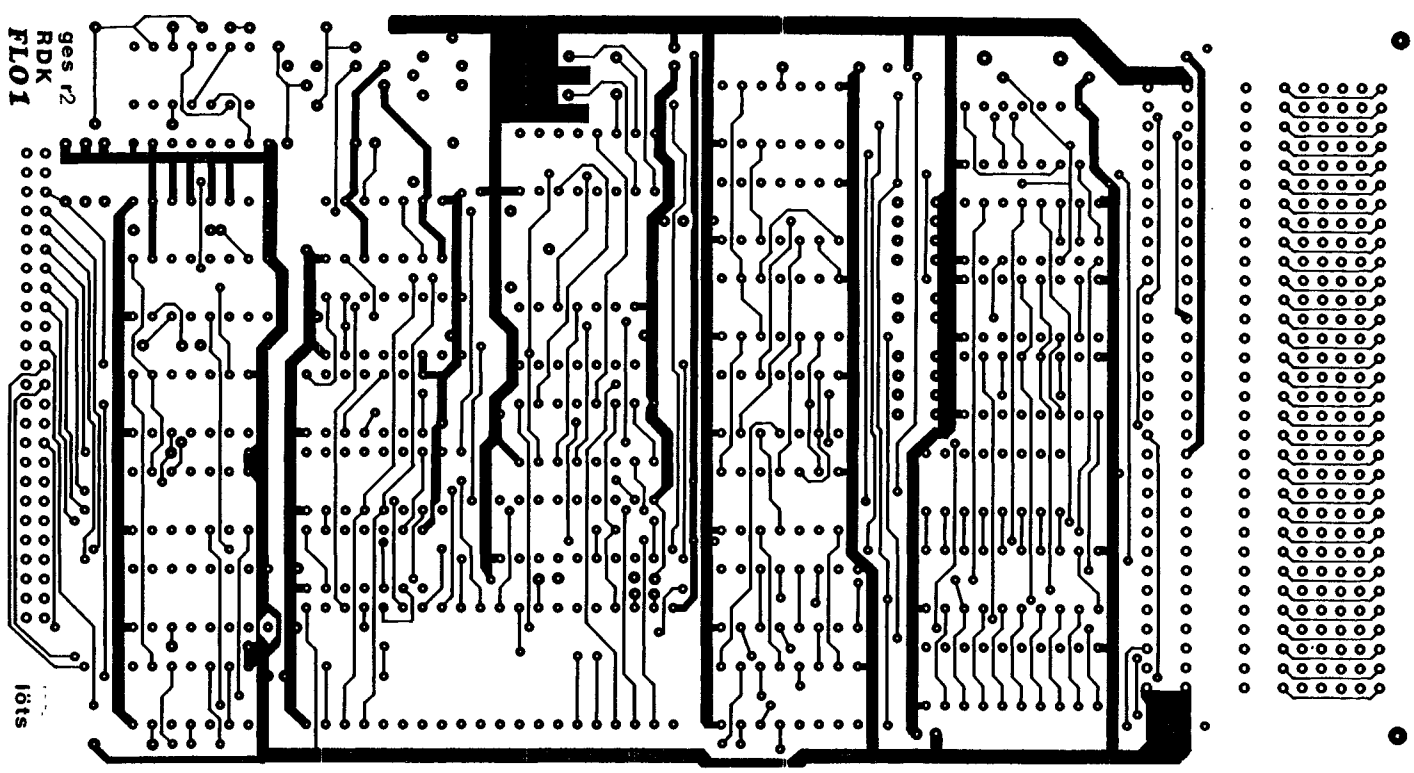
Nun zur Bedeutung der Register und Ports. Die Adressen 40h...43h sind für den Floppy-Controller FD 1797 bestimmt; zur Bedeutung siehe [1]. *Bild 6* zeigt die Bedeutung der einzelnen Bits des Status-Port mit der Adresse 44H. Bit 0 gibt den Status des Anschlusses DRQ an. Mit Bit 1 wird der INT-Ausgang des Controllers sichtbar und mit Bit 2 wird angezeigt, ob der Lese-Schreibkopf der Floppy aufliegt.

Bild 7 zeigt die Belegung des Steuerports. Mit den unteren vier Bits kann ein Drive ausgewählt werden. Dabei wird immer durch ein 1-Signal die Auswahl vorgenommen. Mit den Bits 4 und 5 werden der Floppy-Typ und die Aufzeichnungsdichte eingestellt. Bit 6 aktiviert die Auto-Wait-Schaltung und hält die CPU so lange an, bis ein DRQ-Signal des Floppy-Controllers erscheint. Dieses



▲ Bild 2. Die Floppy-Controller-Gesamtschaltung

Bild 3. Die Lötseite der Platine ▼



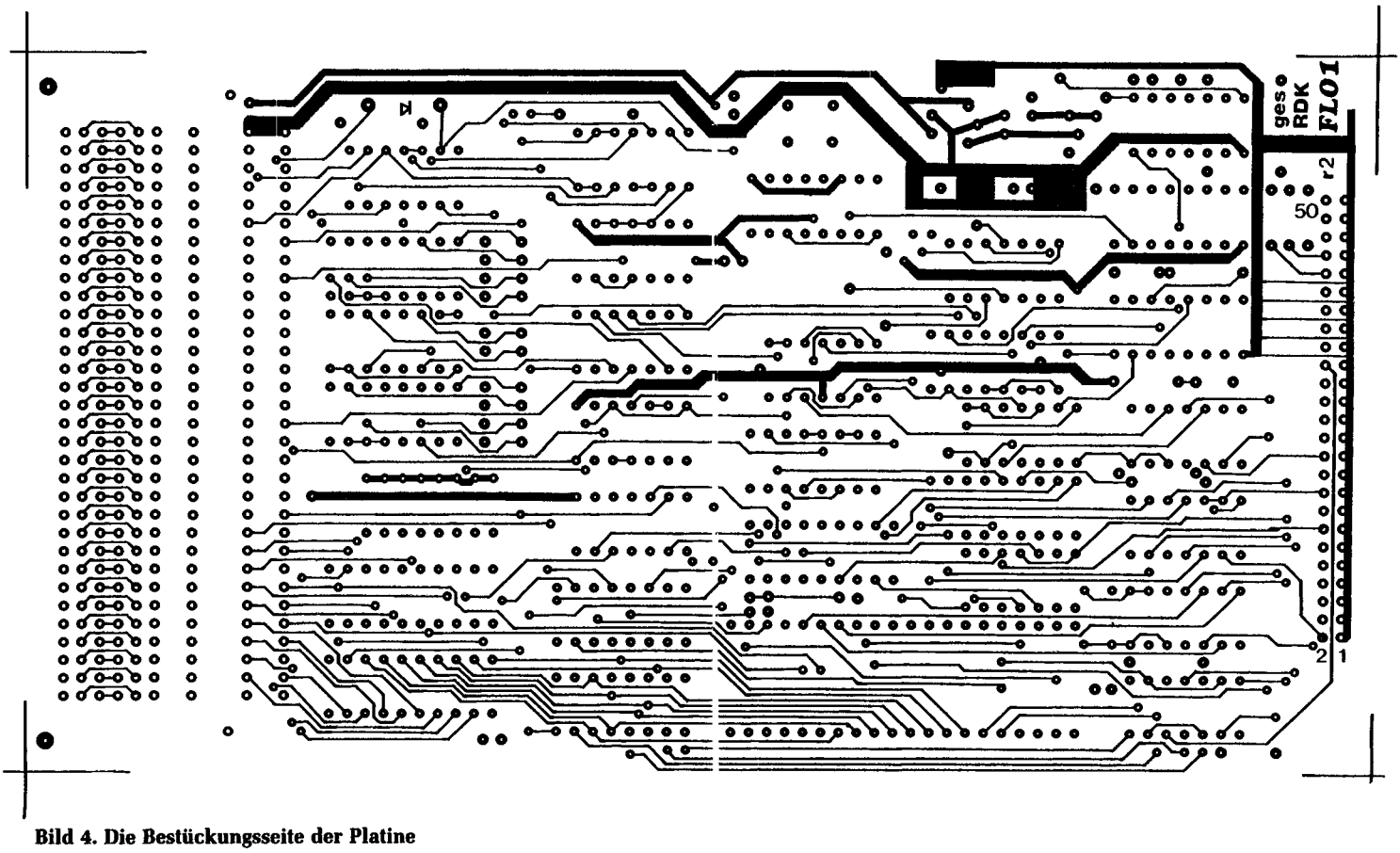


Bild 4. Die Bestückungsseite der Platine

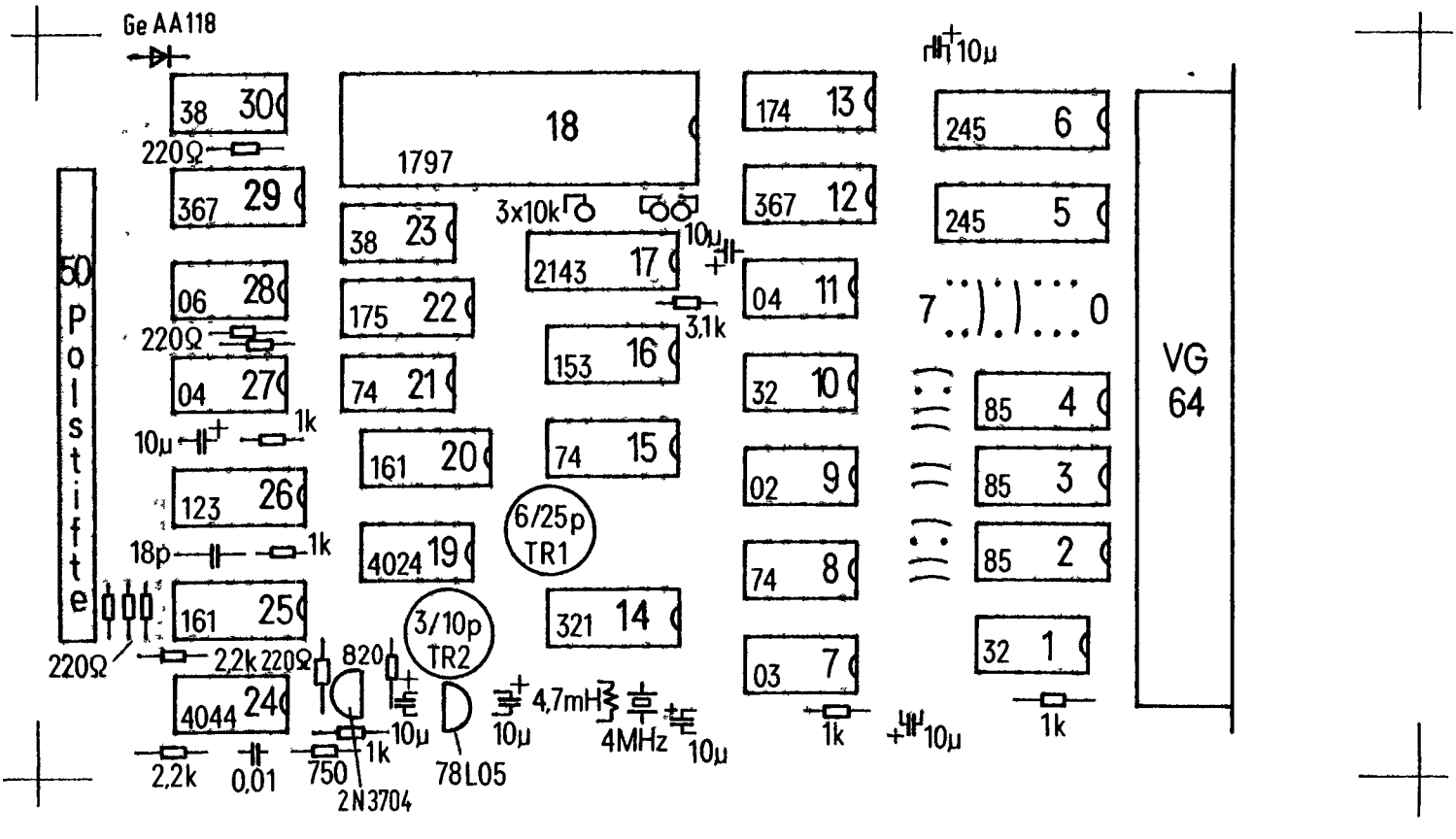


Bild 5. Der Bestückungsplan

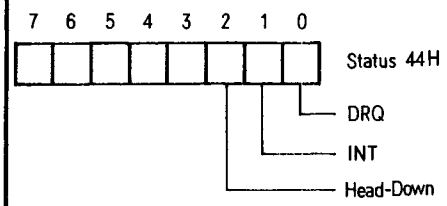


Bild 6. Die Statusport-Belegung

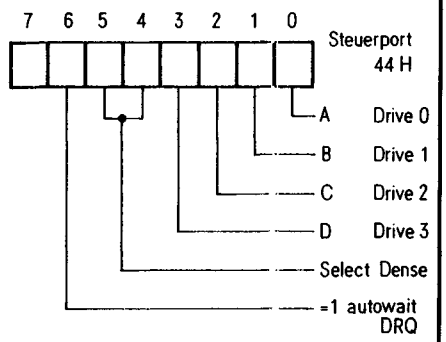


Bild 7. Die Steuerport-Belegung

Select Dense

5	4		Clk (Pin 24 / 1797)	Separator (Pin7 / M1)
0	0	8" FM	2 MHz	8 MHz
0	1	8" MFM	2 MHz	16 MHz
1	0	5¼" FM	1 MHz	4 MHz
1	1	5¼" MFM	1 MHz	8 MHz

Bild 8. Mode-Einstellung

Bit darf bei dynamischen Speichern nicht verwendet werden, da bei einem Lese-Zugriff Wartezeiten bis zu 200 ms auftreten können und damit ein Refresh nicht mehr ordnungsgemäß durchgeführt werden kann. Bei einer 4-MHz-CPU ist dies aber nicht nötig, da 4 MHz für 8 Zoll bei FM ausreichen und bei 6 MHz sogar MFM bearbeitet werden kann, ohne Auto-Wait zu verwenden. *Bild 8* zeigt noch die Bedeutung der Bits 4 und 5 für die Floppy-Auswahl.

Der Aufbau der Karte

Begonnen wird mit der Bestückung der IC-Plätze: alle ICs sind mit Sockeln zu versehen. Danach: Einbau aller passiven Bauteile. Dann werden alle ICs, bis auf WD 2143 und FD 1797, eingesetzt.
Tests:

1. Einschalten der Versorgungsspannung im System bei eingestecktem Floppy-Controller. Der Monitor muß sich jetzt nach wie vor auf dem Bildschirm melden. Ist dies nicht der Fall, so liegt irgendein Kurzschluß vor. Zum Einkreisen können auf der Floppy-Platine einmal alle Bustreiber entfernt werden; wenn dies nicht hilft, werden alle ICs herausgezogen.
2. Messen der Versorgungsspannungen am Floppy-Controller: Pin 40 hat +12 V, Pin 20 hat GND und Pin 21 liegt auf +5 V.
3. Messen am WD2143: An Pin 9 muß GND liegen, an Pin 18 müssen +5 V liegen. An Pin 17 liegt eine Spannung von +12 V an (beim Einsetzen des ICs wird diese geringer).
4. An Pin 14 des IC 4024 müssen +5 V liegen.
5. Kleines Testprogramm aus *Bild 9* eingeben. Damit kann die Decodierung getestet werden. Es wird in einer Schleife auf alle Ports der Karte zugegriffen. An Pin 3 des Floppy-Controllers müssen zwei negative Pulse erkennbar sein, an Pin 9 des 74174 muß ein negativer Puls erscheinen; ebenfalls an Pin 1 des 74367 (B2). An B1 (dem 74245) Pin 19 müssen vier negative Pulse erkennbar sein. Wenn nicht, so ist die Dekodierschaltung zu überprüfen.
6. Nach Ausschalten der Versorgungsspannung restliche ICs einsetzen. Mit dem Befehl QI40 kann der Status des Controllers geprüft werden. Er ist 0 oder 1. Beides ist gut. Nun prüfen, ob Schreib-Lese-Befehle möglich sind.
7. Mit QO42 55 wird ein Register auf 55 gesetzt, mit QI42 muß es den gleichen Wert zurückliefern. Mit QO42 AA wird der komplementäre Wert eingeschrieben. QI42 muß diesen Wert wieder zurückliefern. Liegt ein anderer Wert vor, oder fehlen ein paar Bits, so kann ein Kurzschluß zwischen Datenleitungen vorhanden sein.
8. Einlesen von 44H mit QI44, dann müßte der Wert 00011111 ausgegeben werden, oder zumindest ein ähnlicher Wert.
9. Jetzt das Laufwerk mit dem Controller verbinden. Dabei muß im jeweiligen Handbuch nachgesehen werden, ob die Steckerbelegung mit der hier verwendeten, gängigen Belegung übereinstimmt und welche Einstellungen am Laufwerk durchgeführt werden müssen. Wir arbeiten mit einer Steprate von 3 ms, single-dense und softsektorierten Floppys.
10. Erster Test des Anschlusses. Dazu wird ein Restore-Befehl durchgeführt, das heißt, die Floppy muß die Grundstellung einnehmen und den Kopf auf Track 0 positionieren. Befehle dazu: QO44 1 und QO40 0F. Mit dem ersten Kommando wurde der Wert 1 auf das Steuerport gegeben und damit Laufwerk 0 ausgewählt (Einstellung am Laufwerk). Mit dem zweiten Befehl wird der Restore-Befehl an den Controller gegeben. Die Floppy müßte sich nun geregt haben und die Selekt-LED sollte kurz aufgeleuchtet haben. War der Kopf schon in Position 0, so passiert nichts weiter. Wenn nicht, dreht sich der Schrittmotor und bewegt den Kopf in Richtung Track 0.

```
0100                        .loc 100h
0100  DB40          loop:   in 40h          ;floppy port
0102  D340                  out 40h         ;floppy port
0104  DB44                  in 44h          ;status port
0106  AF                    xra a           ;kein select
0107  D344                  out 44h         ;oder autowait
0109  18F5                  jmpr loop
```

Bild 9. Test-I/O-Zugriffe

```
0100                        .loc 100h
0100  3E01          loop:   mvi a,1         ;select drive 0
0102  D344                  out 044h        ;steuerport
0104  3E0F                  mvi a,0fh       ;RESTORE
0106  D340                  out 040h        ;1797 port
0108  18F6                  jmpr loop
```

Bild 10. Test des Datenseparators

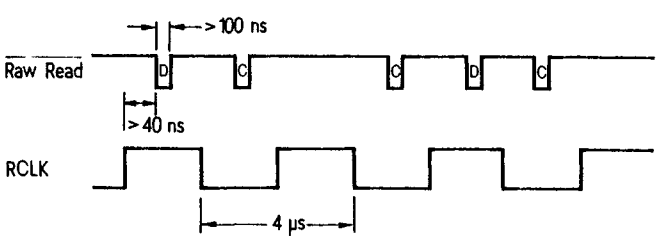


Bild 11. Die Daten- und Takt-Signale

11. Der Datenseparator muß nun abgeglichen werden. Dazu wird erst grob eingestellt. Mit dem Oszilloskop wird Pin 24 des Controllers kontrolliert.

Ausgabe von:	Frequenz:
0 an 44 H	2 MHz,
10 H an 44 H	2 MHz,
20 H an 44 H	1 MHz,
und	
30 H an 44 H	1 MHz.

Nun muß abgeglichen werden. Dazu an Pin 26 messen.
Ausgabe von 0 an 44 H,
Abgleich auf 250 kHz (4 µs) an TR1.
Ausgabe von 10 H an 44 H,
Abgleich auf 500 kHz (2 µs) an TR2.
Der Rest muß nun auch stimmen:
Bei Ausgabe von 20 H an 44 H:
125 kHz (8 µs) an Pin 26.
Bei Ausgabe von 30 H an 44 H:
250 kHz (4 µs) an Pin 26.

Sollten die Werte nicht erreicht werden, liegt das ggf. am verwendeten Trimmer; doch sollte unbedingt noch der nächste Test ausgeführt werden, der mit dem Feinabgleich noch einiges herausholt.

12. *Bild 10* zeigt ein Testprogramm, mit dem das Einlesen getestet werden kann. Dazu wird der Restore-Befehl verwendet, der auch einen Sektor anliest und prüft, ob der Anfang stimmt, um festzustellen, ob Track 0 vorliegt. Dieser Befehl ist sehr praktisch, da die eigentliche Leseroutine noch nicht verwendet werden muß.

13. Dazu muß aber eine schon formatierte Diskette verwendet werden. Sie muß im Standard-IBM-Format Single-Dense formatiert sein. Solche Disketten werden z. B. von BASF im 10er-Pack über den Fachhandel geliefert*). Das Programm in Bild 10 wird gestartet. Es muß sich dann ein Oszillogramm ähnlich *Bild 11* ergeben. *Bild 12* zeigt ein Original-Oszillogramm. Die Lage des Taktes bezüglich positiv oder negativ ist dabei nicht wesentlich. Eine der beiden Takteinheiten muß jedoch immer in der Mitte des Taktes erscheinen. Mit den Trimmern kann die Lage nochmals korrigiert werden, so daß das Bild stabil ist.

Bei Ausführung eines Restore-Befehls muß nach dem Befehl im Statusregister 40H der Wert 0 oder 00100100B stehen, solange der Zugriff noch nicht abgeschlossen ist. Genau abgeglichen werden kann aber auch, wenn am Emitter vom Transistor T1 gemessen wird, dort liegt eine Spannung von ca. 4 V. Mit dem Trimmer kann sie um ein paar hundert mV verändert werden. Der Abgleich ist optimal, wenn die Spannung in der Mitte des Einstellbereichs liegt. Ist der Abgleich nicht durchführbar, schafft ggf. ein kleiner Kondensator (10 pF) parallel zu TR1 oder TR2 Abhilfe.

14. Test SEEK-Befehl. Ausgabe 1 an 44 H und 20 H an 43 H dann 1FH an 40 H. Damit wird Track 20H angefahren. Es muß in 40 H dann der Wert 0 oder 00100000B erscheinen.

15. Nun folgt der Test des Interrupt-Systems: *Bild 13* zeigt die Testroutine. Sie muß durchlaufen werden, und der Monitor muß sich nach dem Restore wieder melden.

16. *Bild 14* zeigt ein kurzes Testprogramm, um den Lesezugriff als ganzes zu prüfen. Nach 80H müssen Datenwerte geschrieben werden. Durch Ändern von Track und Sektor kann dies getestet werden. Im allgemeinen werden bei IBM-Disketten, die leer sind, die Werte E5 oder bei Track 0 40H, 0 oder EBCDIC-Daten gelesen.

17. Einstellen der Schreibprekompensation. *Bild 15* zeigt ein Testprogramm mit laufendem Schreibzugriff und *Bild 16* zeigt das Timing. Mit einem 3,1-kΩ-Widerstand entfällt im allgemeinen ein Abgleich.

*) Das gilt nur für 8-Zoll-Disketten.

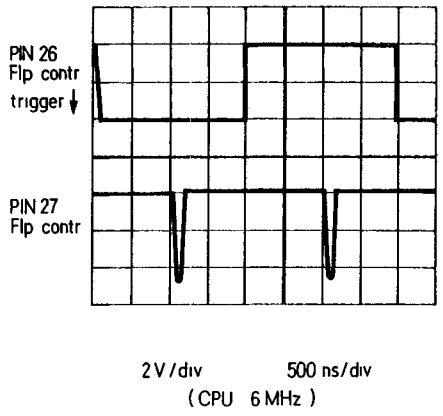


Bild 12. Oszillogramm der Daten und Takt-Signale

```
0010                    .loc 10h
0010  DB40        int:  in 40h          ;bei Minilaufwerken 30h
0012  FB                ei
0013  C9                ret

0100                    .loc 100h
0100  FB                ei            ;Freigabe Interrupt
0101  3E01              mvi a,1       ;drive 0 select
0103  D344              out 44h
0105  3E0F              mvi a,0fh     ;Restore Befehl
0107  D340              out 040h      ;Befehlsport
0109  76                hlt           ;Int abwarten
010A  CD F01E            call 0f01eh   ;nach dem Interrupt
```

Bild 13. Der Interrupt-Test

```
0010                        .loc 10h
0010    DB40        int:    in 040h         ;status einlesen
0012    F1                  pop psw         ;ret adr weg
0013    FB                  ei
0014    C9                  ret
                    ;
0100                        .loc 100h
0100    21 0080     start:  lxi h,80h       ;ziel adresse
0103    11 0001             lxi d,1         ;trk=0 sek=1
0106    01 0100             lxi b,100h      ;lesen drive 0
0109    CD F024             call 0f024h     ;ausfuehren
010C    CD F01E             call 0f01eh     ;monitor
```

Bild 14. Lesen von Daten

Bild 15. Einstellung der Schreib-Präkompensation

```
0010                        .loc 10h
0010    DB40        int:    in 40h  ;status
0012    F1                  pop psw ;ret adr
0013    FB                  ei
0014    C9                  ret     ;eine dr)ber

0100                        .loc 100h
0100    21 0080     loop:   lxi h,80h       ;quelladr
0103    11 0001             lxi d,1         ;tk=0 sek=1
0106    01 0200             lxi b,200h      ;write drv=0
0109    CD F024             call 0f024h     ;FLOPPY CALL
010C    CD F012             call 0f012h     ;abbruch
010F    C4 F01E             cnz 0f01eh      ;monitor dann
0112    18EC                jmpr loop       ;weiter sonst
```

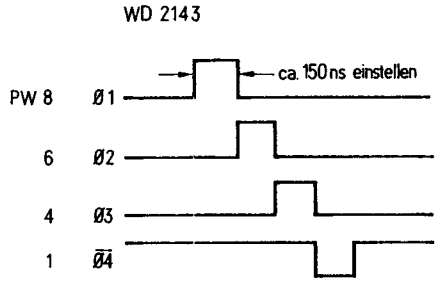


Bild 16. Timing WD 2143

Nun müßten alle Funktionen stimmen – die Karte läuft.
Wie bei den übrigen Komponenten des mc-CP/M-Computers sind Platinen und Bausätze bei der Fa. Graf, Tel. (08 31) 6 19 30, und alle Einzelteile (ICs usw.) bei der Fa. Heninger, Tel. (0 89) 59 19 41, erhältlich.

Literatur

[1] Western Digital: FD 179X-02 Floppy Disk Formatter/Controller Family. Vertrieb: Electronic 2000.
[2] FD 179X Applications.

Tabelle:
Die Stückliste zum Floppy-Controller

1× FD 1797
2× 74 LS 245
3× 74 LS 85
2× 74 LS 32
1× 74 LS 174
2× 74 LS 367
2× 74 LS 04
1× 74 LS 02
3× 74 LS 74
1× 74 03
1× WD 2143
1× 74 LS 153
1× 74 LS 321
2× 74 38
1× 74 LS 175
2× 74 LS 161
1× MC 4024 Motorola
1× MC 4044 Motorola
1× 74 LS 123
1× 7406
1× Transistor 2N 3704
5× 1 kΩ
1× 3,1 kΩ
7× 220 Ω
2× 2,2 kΩ
1× 750 Ω
1× 820 Ω
1× 78L05 oder 7805
7× 10µ Tantal
1× 4 MHz Quarz
1× Spule 100 µH bis 4,7 mH Siemens Minibauform
1× Trimmer 6–25 pF
1× Trimmer 3–10 pF
1× 0,01 µF MMK Kondensator
1× 18 pF
1× 64pol. VG-Leiste
1× 50pol. doppelreihige Stiftleiste
2× 20pol. Sockel
1× 18pol. Sockel
11× 16pol. Sockel
1× 40pol. Sockel
15× 14pol. Sockel

Die Pinbelegungen

Die Pinbelegungen der CPU-, der SIO/PIO- und der Floppy-Controller-Karten-ICs sind hier in der Reihenfolge aus den Stücklisten aufgelistet. Damit sollen Ihre Fehlersuche und Verdrahtungsarbeiten erleichtert werden. Für mehr Einzelheiten über die ICs seien die Datenbücher der Hersteller (Texas Instruments, Motorola, Zilog und so weiter) empfohlen.

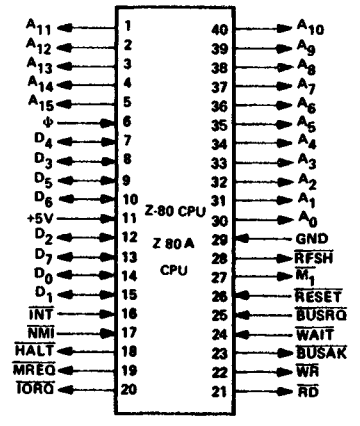


Bild 1. Die Pinbelegung der CPU Z80

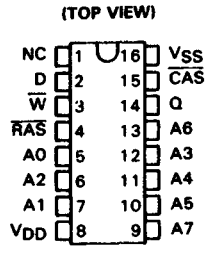


Bild 2. Der Speicherbaustein 4164. VDD = 5 V, VSS = 0 V

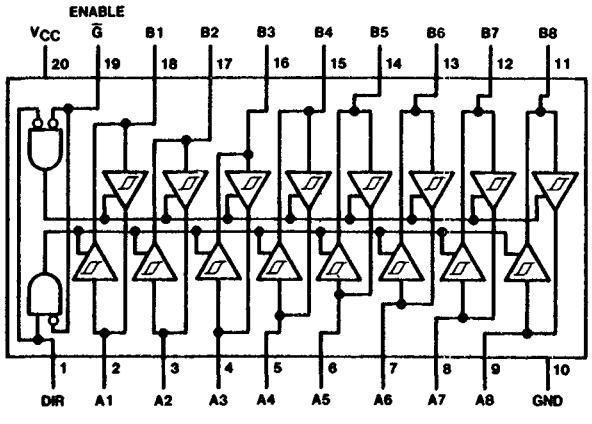


Bild 3. Der „bidirektionale" Datenbus-Puffer-Baustein 74 245

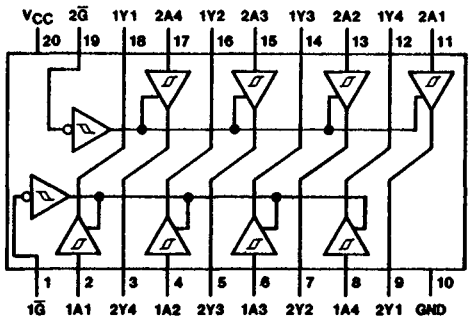
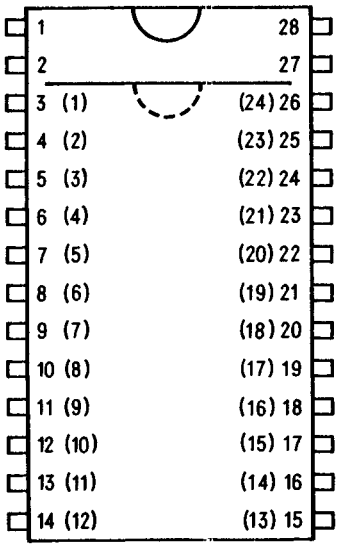


Bild 4. Der Datenbus-Treiber (unidirektional) 74244

EPROM 27256	EPROM 27128	CRAM 5564	PRAM 4864	EPROM 2764	EPROM 2732	PRAM 4816	EEPROM 2816	SRAM 4802	CRAM 6116	EPROM 2716	SRAM 4118
VPP	VPP	NC	$\overline{RFS}$	VPP		$\overline{RFS}$					
A12	A12	A12	A12	A12		NC					
A7	A7	A7	A7	A7	A7	A7	A7	A7	A7	A7	A7
A6	A6	A6	A6	A6	A6	A6	A6	A6	A6	A6	A6
A5	A5	A5	A5	A5	A5	A5	A5	A5	A5	A5	A5
A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4
A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3
A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2
A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
A0	A0	A0	A0	A0	A0	A0	A0	A0	A0	A0	A0
D0	D0	D0	D0	D0	D0	D0	D0	D0	D0	D0	D0
D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1
D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2
GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND

Bild 5. Die Pinbelegungs-Tabelle der modernen EPROMs und statischen RAMs



SRAM 4118	EPROM 2716	CRAM 6116	SRAM 4802	EEPROM 2816	PRAM 4816	EPROM 2732	EPROM 2764	PRAM 4864	CRAM 5564	EPROM 27128	EPROM 27256
					VCC		VCC	VCC	VCC	VCC	VCC
					$\overline{WE}$		$\overline{PGM}$	$\overline{WE}$	R/W	$\overline{PGM}$	A14
VCC	VCC	VCC	VCC	VCC	CS	VCC	NC	CS	CE2	A13	A13
A8	A8	A8	A8	A8	A8	A8	A8	A8	A8	A8	A8
A9	A9	A9	A9	A9	A9	A9	A9	A9	A9	A9	A9
$\overline{WE}$	UPP	$\overline{WE}$	$\overline{WE}$	VPP	NC	A11	A11	A11	A11	A11	A11
$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$/VPP	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$
$\overline{L}$	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7
D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6
D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5
D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4
D3	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3

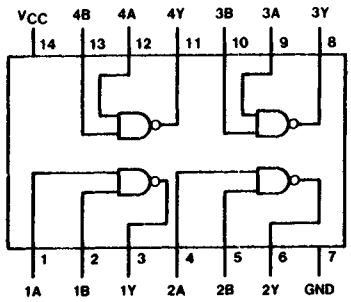


Bild 6. Der Baustein 74LS0. Die Buchstaben-Kombinationen zwischen den Ziffern bezeichnen Leistungsmerkmale, die beachtet werden müssen, aber auf die Pinbelegung keinen Einfluß haben

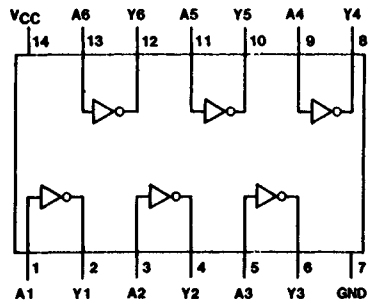


Bild 7. 7404: Sechs Inverter

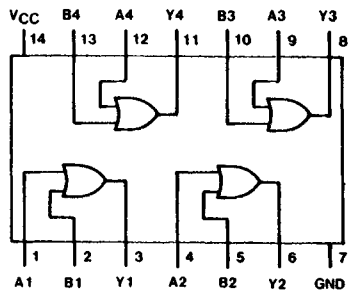


Bild 8. 7432: Viermal Oder mit je zwei Eingängen

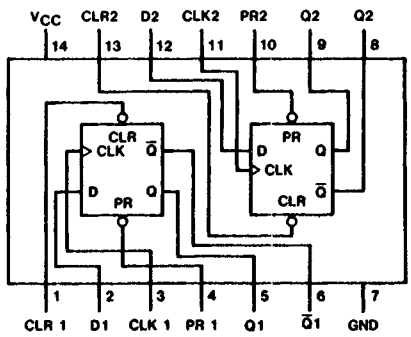


Bild 9. 7474: Zwei D-Flipflops, voreinstellbar, positiv flankentriggerbar, mit Rücksetzeingang

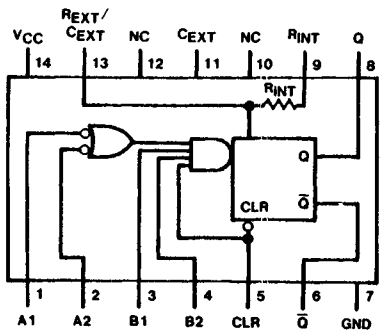


Bild 10. 74 122: Monoflop, retriggerbar, mit Rücksetz-Eingang

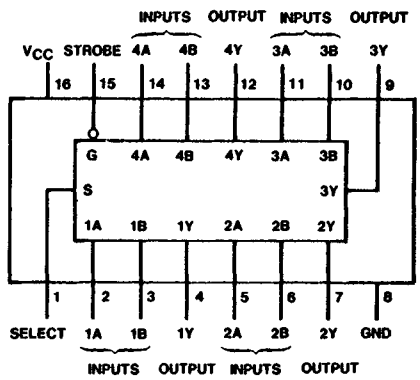


Bild 11. 74 157: Vier Daten-Umschalter/Multiplexer zwei auf eins

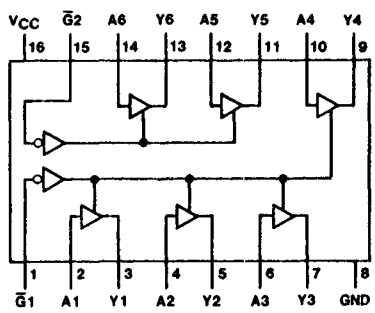


Bild 12. 74 367: Sechs Datenpuffer mit drei Zuständen

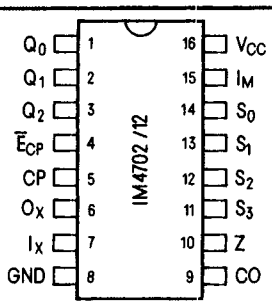


Bild 15. Der Baudraten-Generator ist nicht billig

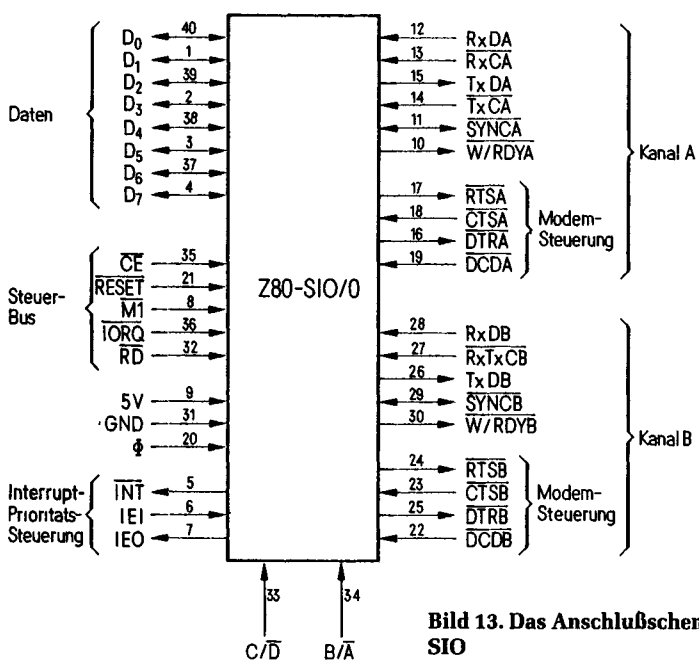


Bild 13. Das Anschlußschema zur SIO

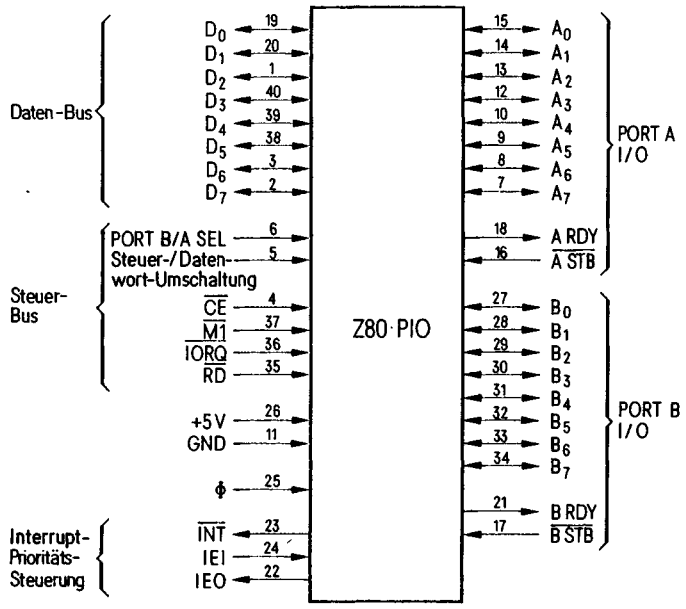


Bild 14. Das Anschlußschema zur PIO

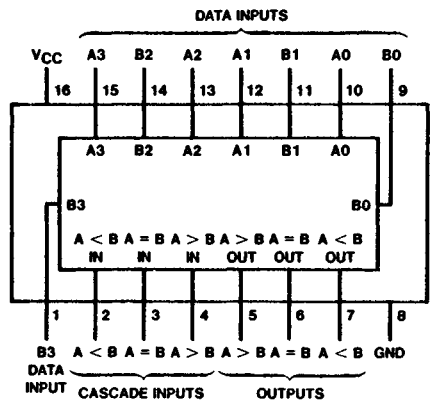


Bild 16. 7485: Vier-Bit-Größen-Komparator

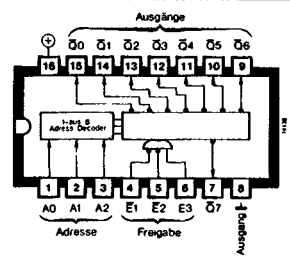
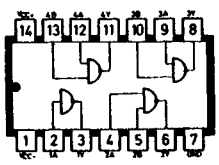


Bild 17. 74 138: Schneller Drei-zu-acht-Decoder



◀ Bild 18. SN 75 188: Vier Leitungstreiber für V.24-Schnittstellen

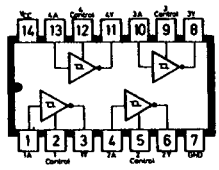
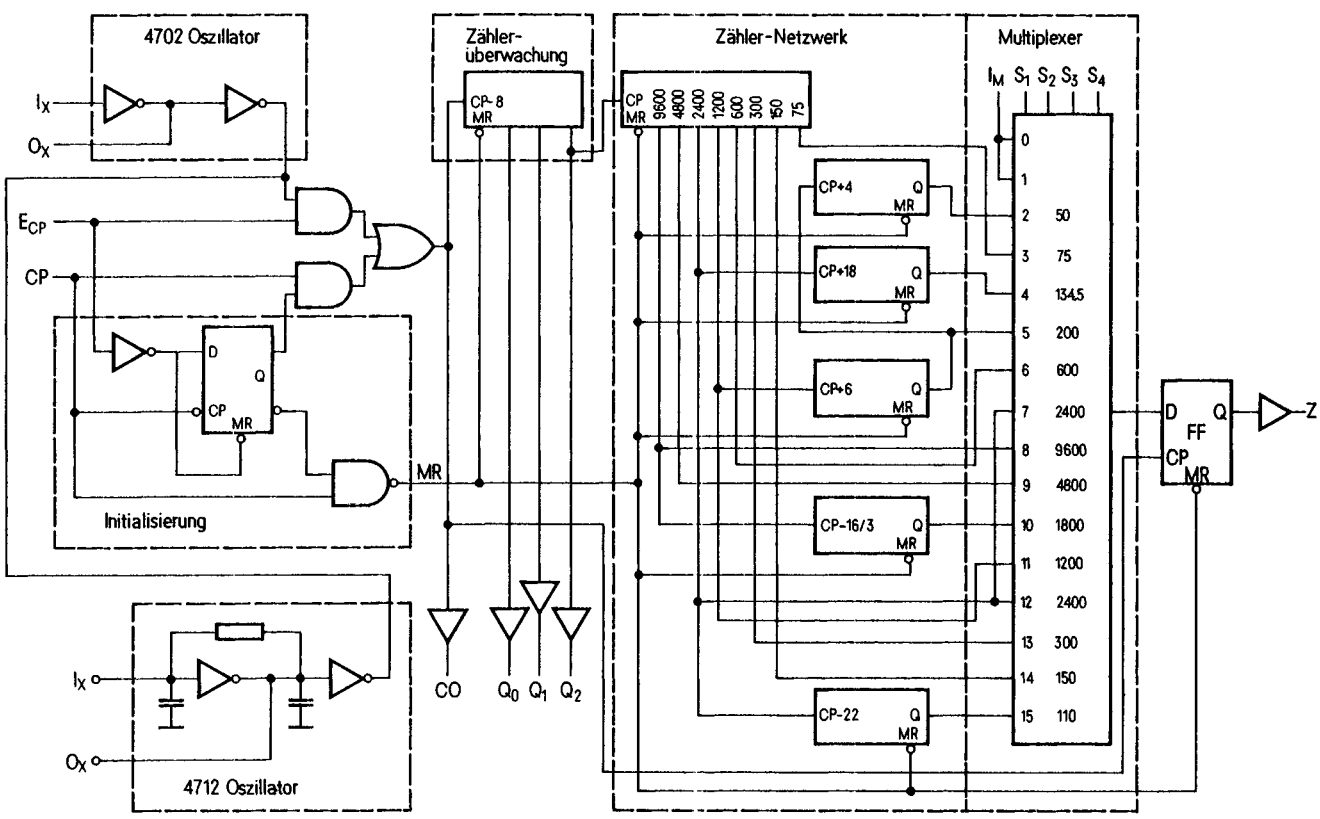
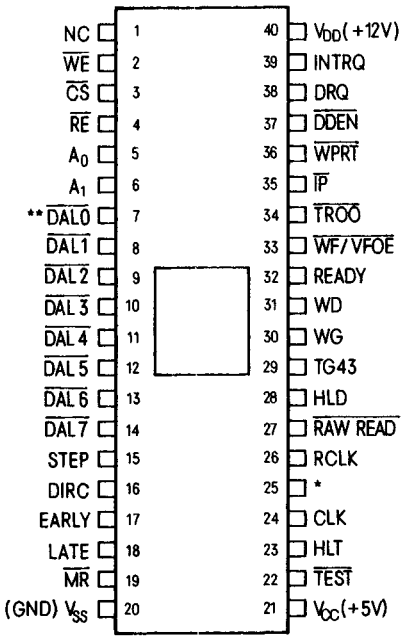


Bild 19. SN 75 189: Vier Leitungsempfänger für V.24-Schnittstellen



Zu Bild 15. Die Innenschaltung



*1791/3=RG 1795/7=SSO
**1793/7 TRUE BUS

Bild 20. Das Anschlußschema des 1797

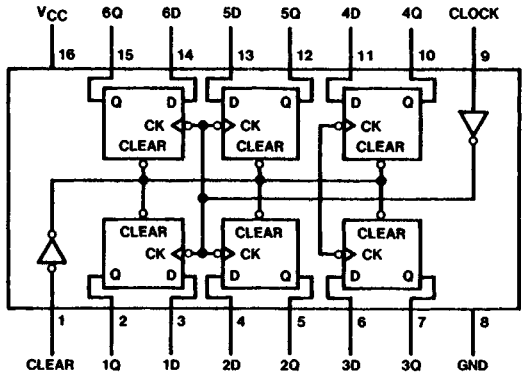


Bild 21. 74 174: Sechs D-Flipflop

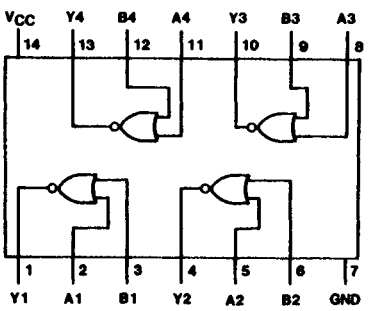


Bild 22. 7402: Vier NOR-Gatter

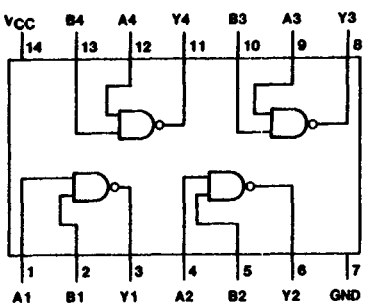
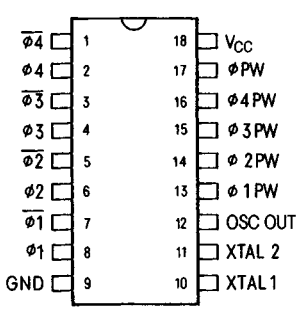


Bild 23. 7403: Vier NAND-Gatter mit offenem Kollektor



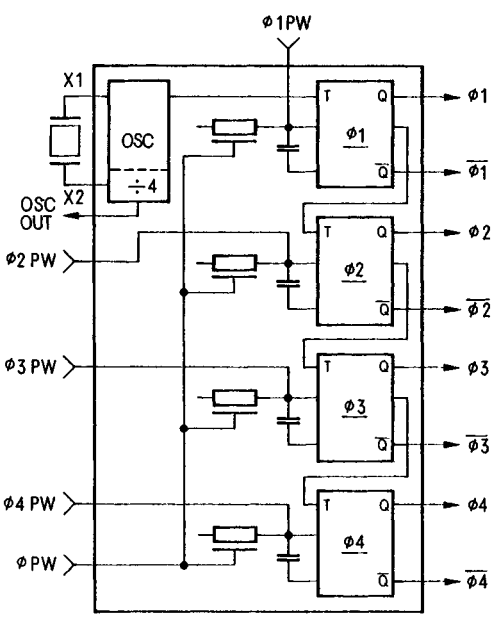
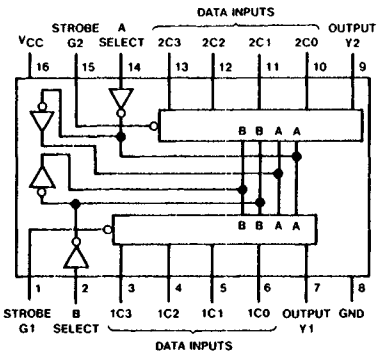
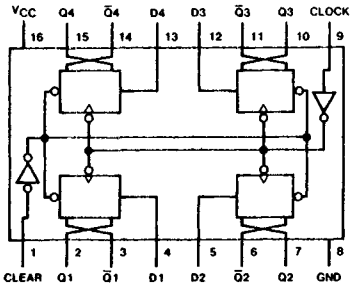


Bild 24. WD2143: Vier-Phasen-Takt-Generator



◀ **Bild 25. 74 153: Daten-Umschalter/ Multiplexer vier auf eins**



◀ **Bild 26. 74 175: Vier D-Flipflops mit Q- und Q̄-Ausgang**

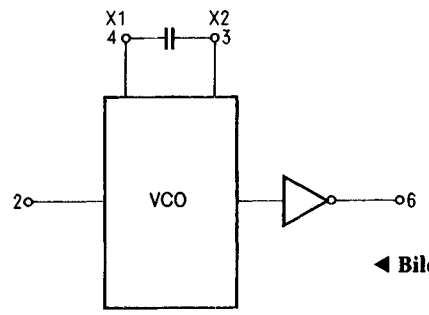
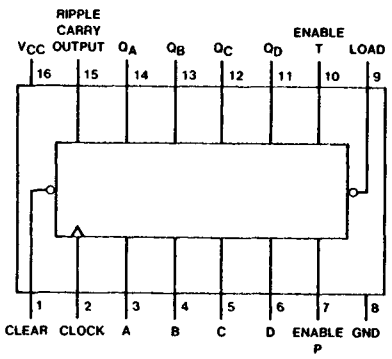
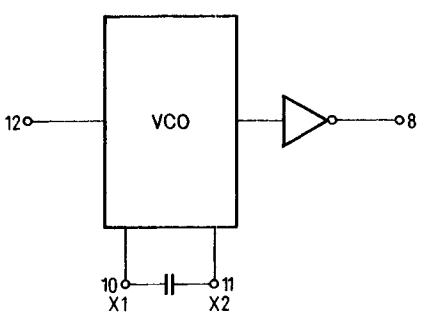


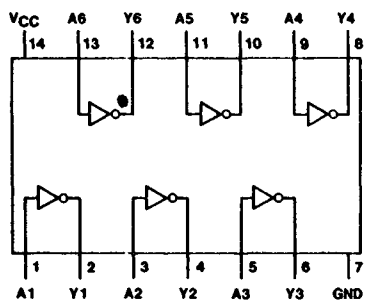
Bild 27. 74 161: ▶ Vier-Bit-Synchron-Zähler (binär)



◀ **Bild 28. Motorola MC 4024: Zwei VCOs**



V_{CC} : VCO_1 = 1, 13
Output-Puffer$_1$ = 14
GND. VCO_2 = 5, 9
Output-Puffer$_2$ = 7



◀ **Bild 30: 7406: Sechs Inverter mit offenem Kollektor, Ausgänge für höhere Spannung geeignet**

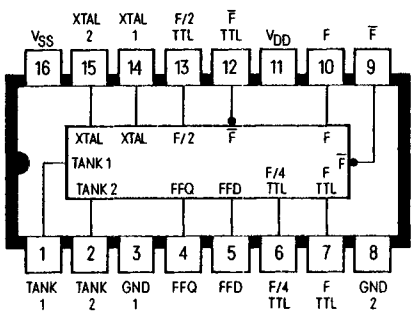


Bild 31. 74 321: Quarzgesteuerter Oszillator

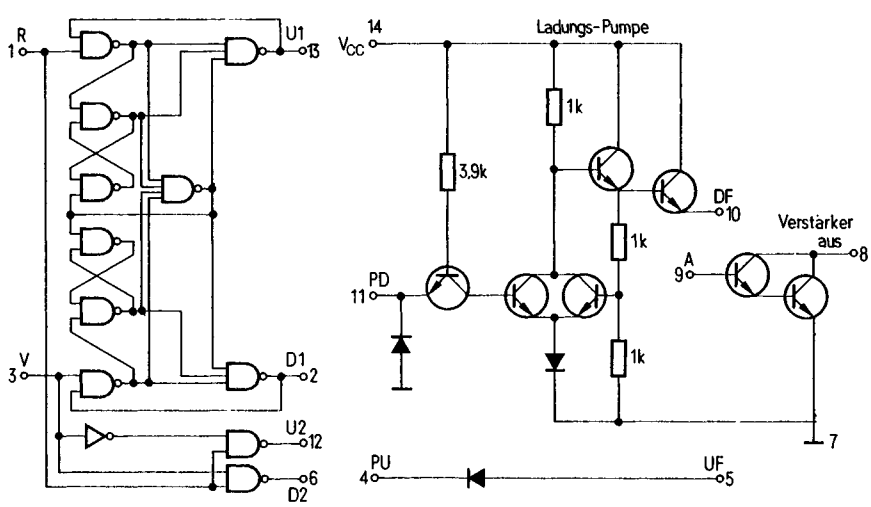


Bild 29. Phasen-Komparator mit „Ladungs-Pumpe" und Verstärker: MC 4044

Rolf-Dieter Klein:

CP/M-Anpassung und Routinen für 8-Zoll-Floppies

Der mc-CP/M-Computer besteht jetzt aus drei Platinen. Damit ist die Bauanleitung für den Kern des Computers abgeschlossen. Allerdings muß noch einiges über die Software gesagt werden, damit das System dann wirklich unter CP/M laufen kann.

Das Betriebssystem CP/M stammt aus den frühen Tagen der Mikrocomputerei. Daß es heute erst richtig aufblüht, spricht für die Weitsicht der Leute, die es programmiert haben und für die geschickte Strategie der Firma Digital Research, die es vermarktet. CP/M ist ein bewährtes und sehr solides Betriebssystem, das zwar auf den Prozessoren der achtziger Modelle beruht, dann aber systemunabhängig konstruiert ist. Es enthält nämlich Teile, die ausdrücklich für die leichte Änderung zur Anpassung programmiert sind. Das macht seinen Erfolg aus. Für den mc-CP/M-Computer gibt es eine angepaßte Version von CP/M im Handel. Wer aber schon CP/M besitzt und selbst anpassen möchte, der muß in Einzelheiten einsteigen, die hier nicht alle geschildert werden.
Zunächst zeigt *Bild 1* die Lage von CP/M im Speicher unseres Computers. Jedes CP/M-System behält sich die ersten Speicherzellen von 0 bis FF vor, um dort Parameter und anderes abzulegen. Danach kommt die sogenannte TPA, das Speichergebiet, in dem ein Benutzer seine Programme und Daten ablegen kann. Oben nun sitzen die CP/M-Module, die das eigentliche Betriebssystem ausmachen. Ab D400 kommen zunächst die Programmteile, die Kommandos von der Konsole bearbeiten (CCP), dann die Teile, die das „Filehandling" durchführen, also (unter anderem) den Betrieb des Massenspeichersystems unterstützen.

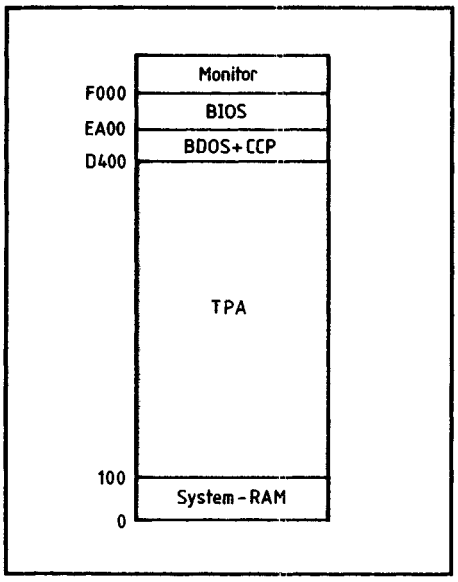


Bild 1. Die Speicheraufteilung. Nach diesem Muster ist der Speicher eines jeden CP/M-Systems aufgeteilt

Dieses sogenannte BDOS (Basic Disk Operation System) greift auch auf Programme zurück, die sich im BIOS, dem Basic Input Output System, befinden. Das BIOS ist der oben angesprochene Programmteil, der speziell für die vorhandene Hardware programmiert werden muß. Unser BIOS greift auf den mc-Monitor zurück.
Das BIOS beinhaltet alle hardwareabhängigen Softwareteile. So zum Beispiel Consol-Routinen, Drucker-Routinen und die Floppy-Schnittstelle. Ferner sind dort Parameter vorhanden, mit welchen die Aufteilung auf der Floppy sowie die Größe der Floppy bestimmt werden.
Bild 2 zeigt das komplette Listing des BIOS für unsere Floppy-Karte. Dabei werden die I/O-Routinen aus dem Monitor verwendet um Platz zu sparen.
Als Laufwerke sind 4 Drives vorgesehen. Dabei können natürlich auch nur 2 verwendet werden. Die 4 Laufwerke sind so organisiert, daß auch zwei doppelseitige Drives verwendet werden können. Die logischen Nummern von CP/M für die Laufwerke sind im Monitor wie folgt aufgeteilt:

A:	Laufwerk 0 Seite 1
B:	Laufwerk 1 Seite 1
C:	Laufwerk 0 Seite 2
D:	Laufwerk 1 Seite 2

Die einzelne Zuordnung muß gemäß der Anleitung, die zu jedem Laufwerk erhältlich ist, mit Brücken eingestellt werden.
Das logische Floppy-Format ist im BIOS so festgelegt, daß die Standard-8-Zoll-single-dense-CP/M-Floppy direkt lesbar ist. Dabei gibt es ein paar wichtige Punkte zu beachten.

Das CP/M-System befindet sich auf den ersten beiden Spuren der Floppy. Vom Monitor wird mit dem I-Befehl der erste Sektor von Spur 0 nach 80H geladen. Dort befindet sich ein weiteres Boot-Programm, das das BDOS und BIOS von Bereich 0D400H ablädt; anschließend wird das BIOS gestartet. *Bild 3* zeigt das Boot-Programm. In *Bild 4* ist nochmals das BIOS als Hexdump abgedruckt.

Die Einsprünge MINI und MAXI

Die Floppy-Routinen sind über zwei Einsprünge erreichbar. Der Eingang MINI ist für Mini-Laufwerke gedacht und adressiert den Floppy-Controller auf Adresse 30H (Einstellen auf der Floppy-Karte). Der Eingang MAXI adressiert den Controller auf 40H.
Das Floppy-Programm arbeitet mit Interrupts, die auf Adresse 10H ausgeführt werden. Unmittelbar nach dem Aufruf der Unterprogramme wird, wie im Listing ersichtlich, auf die Adresse 10H ein Sprung zum Interruptprogrammteil geschrieben. Der alte Inhalt dieser Zellen wird gerettet. Dadurch laufen alle CP/M-Programme, auch APL, das z. B. diesen RESTART-Bereich selbst verwendet.

```
                        MACLIB DISKDEF  ;LOAD DEFINTION FOR DISKS

                ;   VERSION 2.2   CP/M   800823 RDK  ;
                ; VERSION811101    MIXED SOFT UND HARD DRIVE
                ; VERSION 820815 IMI EXT. DIRECTORY
                ;830113
                ;   VERSION RDKCOMP BIOS

0016 =          VERS    EQU     22
FFFF =          TRUE    EQU     OFFFFH
0000 =          FALSE   EQU     NOT TRUE
FFFF =          TEST    EQU     TRUE

003C =          MSIZE   EQU     60      ;CP/M VERSION MEMORY SIZE

A000 =          BIAS    EQU (MSIZE-20)*1024   ;MINIMUM 20K
D400 =          CCP     EQU     3400H+BIAS

DC06 =          BDOS    EQU CCP+806H
EA00 =          BIOS    EQU CCP+1600H
                ;
D400 =          CPMB    EQU CCP

EA00                    ORG BIOS

                ;
1600 =          CPML    EQU $-CPMB

002C =          NSECTS  EQU CPML/128
0002 =          OFFSET  EQU 2   ;START BEI TRACK 3 BEI MAXI
0003 =          MOFFS   EQU 3   ;BEI HARDFLOPPY
0004 =          CDISK   EQU 4   ;ADR 4 SP FUER LAST DRIVE
0080 =          BUFF    EQU 80H ;DEFAULT BUFFER ADRESS
000A =          RETRY   EQU 10  ;10= MAX RETRY



                ; VEKTORTABELLE

EA00 C3DEEA             JMP BOOT
EA03 C3EEEA     WBOOTE: JMP WBOOT
EA06 C3D2EA             JMP CONST
EA09 C3D5EA             JMP CONIN
EA0C C3DBEA             JMP CONOUT
EA0F C30FF0             JMP LIST
EA12 C30CF0             JMP PUNCH
EA15 C306F0             JMP READER

EA18 C37EEB             JMP HOME
EA1B C383EB             JMP SELDSK
EA1E C399EB             JMP SETTRK
EA21 C39EEB             JMP SETSEC
EA24 C3BAEB             JMP SETDMA
EA27 C3CDEB             JMP READ

EA2A C338EC             JMP WRITE
EA2D C37CEB             JMP LISTST
EA30 C3A3EB             JMP SECTRAN
                ;
                        DISKS 5
EA33+=          DPBASE  EQU     $       ;BASE OF DISK PARAMETER BLOCKS
EA33+92EA0000   DPE0:   DW      XLT0,0000H      ;TRANSLATE TABLE
EA37+00000000           DW      0000H,0000H     ;SCRATCH AREA
EA3B+95EC83EA           DW      DIRBUF,DPB0     ;DIR BUFF,PARM BLOCK
EA3F+33ED15ED           DW      CSV0,ALV0       ;CHECK, ALLOC VECTORS
EA43+92EA0000   DPE1:   DW      XLT1,0000H      ;TRANSLATE TABLE
EA47+00000000           DW      0000H,0000H     ;SCRATCH AREA
EA4B+95EC83EA           DW      DIRBUF,DPB1     ;DIR BUFF,PARM BLOCK
EA4F+61ED43ED           DW      CSV1,ALV1       ;CHECK, ALLOC VECTORS
EA53+92EA0000   DPE2:   DW      XLT2,0000H      ;TRANSLATE TABLE
EA57+00000000           DW      0000H,0000H     ;SCRATCH AREA
EA5B+95EC83EA           DW      DIRBUF,DPB2     ;DIR BUFF,PARM BLOCK
EA5F+8FED71ED           DW      CSV2,ALV2       ;CHECK, ALLOC VECTORS
EA63+92EA0000   DPE3:   DW      XLT3,0000H      ;TRANSLATE TABLE
EA67+00000000           DW      0000H,0000H     ;SCRATCH AREA
EA6B+95EC83EA           DW      DIRBUF,DPB3     ;DIR BUFF,PARM BLOCK
EA6F+BDED9FED           DW      CSV3,ALV3       ;CHECK, ALLOC VECTORS
EA73+00000000   DPE4:   DW      XLT4,0000H      ;TRANSLATE TABLE
EA77+00000000           DW      0000H,0000H     ;SCRATCH AREA
EA7B+95ECACEA           DW      DIRBUF,DPB4     ;DIR BUFF,PARM BLOCK
EA7F+4DEECDED           DW      CSV4,ALV4       ;CHECK, ALLOC VECTORS
                        DISKDEF 0,1,26,6,1024,240,64,64,OFFSET
EA83+=          DPB0    EQU     $               ;DISK PARM BLOCK
EA83+1A00               DW      26              ;SEC PER TRACK
EA85+03                 DB      3               ;BLOCK SHIFT
EA86+07                 DB      7               ;BLOCK MASK
EA87+00                 DB      0               ;EXTNT MASK
EA88+EF00               DW      239             ;DISK SIZE-1
EA8A+3F00               DW      63              ;DIRECTORY MAX
EA8C+C0                 DB      192             ;ALLOC0
EA8D+00                 DB      0               ;ALLOC1
EA8E+1000               DW      16              ;CHECK SIZE
EA90+0200               DW      2               ;OFFSET
EA92+=          XLT0    EQU     $               ;TRANSLATE TABLE
EA92+01                 DB      1
EA93+07                 DB      7
EA94+0D                 DB      13
EA95+13                 DB      19
EA96+19                 DB      25
EA97+05                 DB      5
EA98+0B                 DB      11
EA99+11                 DB      17
EA9A+17                 DB      23
EA9B+03                 DB      3
EA9C+09                 DB      9
EA9D+0F                 DB      15
EA9E+15                 DB      21
EA9F+02                 DB      2
EAA0+08                 DB      8
EAA1+0E                 DB      14
EAA2+14                 DB      20
EAA3+1A                 DB      26
EAA4+06                 DB      6
EAA5+0C                 DB      12
EAA6+12                 DB      18
EAA7+18                 DB      24
EAA8+04                 DB      4
EAA9+0A                 DB      10
EAAA+10                 DB      16
EAAB+16                 DB      22
                        DISKDEF 1,0
EA83+=          DPB1    EQU     DPB0    ;EQUIVALENT PARAMETERS
001E+=          AL51    EQU     AL50    ;SAME ALLOCATION VECTOR SIZE
0010+=          CS51    EQU     CS50    ;SAME CHECKSUM VECTOR SIZE
EA92+=          XLT1    EQU     XLT0    ;SAME TRANSLATE TABLE
                        DISKDEF 2,0
EA83+=          DPB2    EQU     DPB0    ;EQUIVALENT PARAMETERS
001E+=          AL52    EQU     AL50    ;SAME ALLOCATION VECTOR SIZE
0010+=          CS52    EQU     CS50    ;SAME CHECKSUM VECTOR SIZE
EA92+=          XLT2    EQU     XLT0    ;SAME TRANSLATE TABLE
                        DISKDEF 3,0
EA83+=          DPB3    EQU     DPB0    ;EQUIVALENT PARAMETERS
001E+=          AL53    EQU     AL50    ;SAME ALLOCATION VECTOR SIZE
0010+=          CS53    EQU     CS50    ;SAME CHECKSUM VECTOR SIZE
EA92+=          XLT3    EQU     XLT0    ;SAME TRANSLATE TABLE
                        DISKDEF 4,0,255,0,8192,1024,1024,1024,0
EAAC+=          DPB4    EQU     $               ;DISK PARM BLOCK
EAAC+0001               DW      256             ;SEC PER TRACK
EAAE+06                 DB      6               ;BLOCK SHIFT
EAAF+3F                 DB      63              ;BLOCK MASK
EAB0+03                 DB      3               ;EXTNT MASK
EAB1+FF03               DW      1023            ;DISK SIZE-1
EAB3+FF03               DW      1023            ;DIRECTORY MAX
EAB5+F0                 DB      240             ;ALLOC0
EAB6+00                 DB      0               ;ALLOC1
EAB7+0001               DW      256             ;CHECK SIZE
EAB9+0000               DW      0               ;OFFSET
0000+=          XLT4    EQU     0               ;NO XLATE TABLE

                ; ENDEF AT END OF ASSEMBLY

F01E =          MON80   EQU 0F01EH
F01E =          RMON80  EQU 0F01EH

000D =          CR      EQU 0DH
000A =          LF      EQU 0AH

                SIGNON:
EABB 0D0A0A             DB 0DH,0AH,0AH
EABE 3630               DB '60'         ;
EAC0 6B2043502F         DB 'k CP/M vers 2.2'
EACF 0D0A00             DB 0DH,0AH,0

                CONST:
EAD2 C312F0             JMP 0F012H
                CONIN:
EAD5 CD03F0             CALL 0F003H


EAD8 E67F               ANI 7FH
EADA C9                 RET

                CONOUT:
EADB C309F0             JMP 0F009H

F00F =          LIST    EQU 0F00FH
F00C =          PUNCH   EQU 0F00CH
F006 =          READER  EQU 0F006H

                BOOT:
EADE 310001             LXI SP,BUFF+80H
EAE1 21BBEA             LXI H,SIGNON
EAE4 CDC0EB             CALL PRMSG
EAE7 AF                 XRA A
EAE8 320400             STA CDISK
EAEB C33AEB             JMP GOCPM

                WBOOT:
EAEE 310080             LXI SP,BUFF
EAF1 0E0A               MVI C,RETRY
EAF3 C5                 PUSH B
EAF4 0100D4     WBOOTO: LXI B,CPMB
EAF7 CD8AEB             CALL SETDMA
EAFA 0E00               MVI C,0
EAFC CD83EB             CALL SELDSK
EAFF 0E00               MVI C,0
EB01 CD99EB             CALL SETTRK
EB04 0E02               MVI C,2
EB06 CD9EEB             CALL SETSEC

EB09 C1                 POP B
EB0A 062C               MVI B,NSECTS
```

Bild 2. BIOS, im 8080-MAC-Assemblerformat. Mit den EQU-Anweisungen für den Assembler werden verschiedene Parameter vereinbart. ORG EA00 legt die Speicherlage unseres BIOS fest. Die Vektortabelle ist eins der Geheimnisse von CP/M. Die Reihenfolge der Sprünge in dieser Tabelle und die Funktion der angesprungenen Routinen sind genormt. Bei uns nutzen die BIOS-Routinen wiederum oft den Monitor, zum Beispiel bei der Bedienung der Floppys. Nach den Sprungvektoren folgen vor den eigentlichen BIOS-Routinen noch viele Parameter, die von BDOS benötigt werden. Sie enthalten Informationen über die angeschlossenen Laufwerke. Benötigt eine der BDOS-Funktionen eine I/O-Schnittstelle, dann wendet sie sich über die Sprungtabelle an die passende Routine, die dann den Kontakt vermittelt

```
EB0C C5         RDSEC:  PUSH B
EB0D CDCDEB             CALL READ
EB10 C264EB             JNZ BOOTERR
EB13 2A8AEC             LHLD IOD
EB16 118000             LXI D,128
EB19 19                 DAD D
EB1A 44                 MOV B,H
EB1B 4D                 MOV C,L
EB1C CDBAEB             CALL SETDMA
EB1F 3A89EC             LDA IOS
EB22 FE1A               CPI 26  ;MAXI 26 SEC/TRK
EB24 DA30EB             JC RD1

EB27 3A88EC             LDA IOT
EB2A 3C                 INR A
EB2B 4F                 MOV C,A
EB2C CD99EB             CALL SETTRK
EB2F AF                 XRA A
EB30 3C         RD1:    INR A
EB31 4F                 MOV C,A
EB32 CD9EEB             CALL SETSEC
EB35 C1                 POP B
EB36 05                 DCR B
EB37 C20CEB             JNZ RDSEC

                GOCPM:
EB3A 018000             LXI B,BUFF
EB3D CDBAEB             CALL SETDMA
EB40 3EC3               MVI A,JMP
EB42 320000             STA 0
EB45 2103EA             LXI H,WBOOTE
EB48 220100             SHLD 1
EB4B 320500             STA 5
EB4E 21060C             LXI H,BDOS
EB51 220600             SHLD 6
EB54 323800             STA 7*8
EB57 211EF0             LXI H,MON80
EB5A 223900             SHLD 7*8+1
EB5D 3A0400             LDA CDISK       ;LAST LOGGED IN DISK
EB60 4F                 MOV C,A
EB61 C300D4             JMP CPMB

                BOOTERR:
EB64 C1                 POP B
EB65 0D                 DCR C
EB66 CA6DEB             JZ BOOTERO
EB69 C5                 PUSH B
EB6A C3F4EA             JMP WBOOTO      ;TRY AGAIN

                BOOTERO:
EB6D 2176EB             LXI H,BOOTMSG
EB70 CDC0EB             CALL PRMSG
EB73 C31EF0             JMP MON80

                BOOTMSG:
EB76 3F424F4F54         DB '?BOOT',0

                LISTST:
EB7C AF                 XRA A
EB7D C9                 RET

                HOME:
EB7E 0E00               MVI C,0
EB80 C399EB             JMP SETTRK

                SELDSK:
EB83 210000             LXI H,0
EB86 79                 MOV A,C
EB87 FE05               CPI NDISKS
EB89 D0                 RNC

                        ;NO INRA DA 0..5
EB8A 3285EC             STA DBANK
EB8D 69                 MOV L,C
EB8E 2600               MVI H,0
EB90 29                 DAD H
EB91 29                 DAD H
EB92 29                 DAD H
EB93 29                 DAD H
EB94 1133EA             LXI D,DPBASE
EB97 19                 DAD D
EB98 C9                 RET
                ;
                SETTRK:
EB99 2188EC             LXI H,IOT
EB9C 71                 MOV M,C
EB9D C9                 RET

                SETSEC:
EB9E 2189EC             LXI H,IOS
EBA1 71                 MOV M,C
EBA2 C9                 RET

                SECTRAN:
EBA3 7A                 MOV A,D
EBA4 B3                 ORA E
EBA5 CAB2EB             JZ SE1
EBA8 0600               MVI B,0
EBAA EB                 XCHG
EBAB 09                 DAD B
EBAC 7E                 MOV A,M
EBAD 3289EC             STA IOS
EBB0 6F                 MOV L,A
EBB1 C9                 RET
EBB2 69         SE1:    MOV L,C
EBB3 79                 MOV A,C
EBB4 3289EC             STA IOS ;AUCH MERKEN
EBB7 2600               MVI H,0 ;SE 0..255 MAX
EBB9 C9                 RET
                ;

                SETDMA:
EBBA 69                 MOV L,C
EBBB 60                 MOV H,B
EBBC 228AEC             SHLD IOD
EBBF C9                 RET

                PRMSG:
EBC0 7E                 MOV A,M
EBC1 B7                 ORA A
EBC2 C8                 RZ
EBC3 E5                 PUSH H
EBC4 4F                 MOV C,A
EBC5 CDDBEA             CALL CONOUT
EBC8 E1                 POP H
EBC9 23                 INX H
EBCA C3C0EB             JMP PRMSG

                ;READ AND WRITE USING
                ; EXEC
                ; HL=DMA ADR
                ; DE=TRACK/SECTOR
                ; B=0 RSTORE
                ;   1 READ
                ;   2 WRITE
                ; C=DRIVE 0...5

                READ:
EBCD 3A85EC             LDA DBANK
EBD0 FE04               CPI 4H
EBD2 C2F3EB             JNZ FLOP
EBD5 3A89EC             LDA IOS
EBD8 6F                 MOV L,A
EBD9 3A88EC             LDA IOT
EBDC 67                 MOV H,A
EBDD 2291EC             SHLD INDADR
EBE0 210000             LXI H,0
EBE3 2293EC             SHLD INDADR2
EBE6 2A8AEC             LHLD IOD
EBE9 1191EC             LXI D,INDADR
EBEC 0E00               MVI C,0
EBEE 0601               MVI B,1
EBF0 C37FEC             JMP IMISYS

EBF3 3A88EC     FLOP:   LDA IOT
EBF6 B7                 ORA A
EBF7 C217EC             JNZ SK1
EBFA 3A89EC             LDA IOS ;AUCH SECTOR TESTEN
EBFD FE02               CPI 2   ;NUR BEI BOOT
EBFF C217EC             JNZ SK1 ;RESTORE DURCHFUEHREN
EC02 210000             LXI H,0
EC05 110000             LXI D,0
EC08 3A85EC             LDA DBANK
EC0B 4F                 MOV C,A
EC0C 0600               MVI B,0
EC0E E5                 PUSH H
EC0F D5                 PUSH D
EC10 C5                 PUSH B
EC11 CD82EC             CALL EXEC       ;RSTORE
EC14 C1                 POP B
EC15 D1                 POP D
EC16 E1                 POP H
                ;
EC17 060A       SK1:    MVI B,RETRY
EC19 C5         LP:     PUSH B
EC1A 2A8AEC             LHLD IOD
EC1D 3A88EC             LDA IOT
EC20 57                 MOV D,A
EC21 3A89EC             LDA IOS
EC24 5F                 MOV E,A
EC25 0601               MVI B,1
EC27 3A85EC             LDA DBANK
EC2A 4F                 MOV C,A
EC2B CD82EC             CALL EXEC
EC2E C1                 POP B
EC2F C8                 RZ
EC30 05                 DCR B
EC31 C219EC             JNZ LP
EC34 3E01               MVI A,1
EC36 B7                 ORA A
EC37 C9                 RET

                WRITE:
EC38 3A85EC             LDA DBANK
EC3B FE04               CPI 4
EC3D C25EEC             JNZ FLOP1
EC40 3A89EC             LDA IOS
EC43 6F                 MOV L,A
EC44 3A88EC             LDA IOT
EC47 67                 MOV H,A
EC48 2291EC             SHLD INDADR
EC4B 210000             LXI H,0
EC4E 2293EC             SHLD INDADR2
EC51 2A8AEC             LHLD IOD
EC54 1191EC             LXI D,INDADR
EC57 0E00               MVI C,0
EC59 0602               MVI B,2
EC5B C37FEC             JMP IMISYS


EC5E 060A       FLOP1:  MVI B,RETRY

EC60 C5         LPP:    PUSH B
EC61 2A8AEC             LHLD IOD
EC64 3A88EC             LDA IOT
EC67 57                 MOV D,A
EC68 3A89EC             LDA IOS
EC6B 5F                 MOV E,A
EC6C 0602               MVI B,2
```

```
EC6E 3A85EC             LDA DBANK
EC71 4F                 MOV C,A
EC72 CD82EC             CALL EXEC
EC75 C1                 POP B
EC76 C8                 RZ
EC77 05                 DCR B
EC78 C260EC             JNZ LPP
EC7B 3E01               MVI A,1
EC7D B7                 ORA A
EC7E C9                 RET

0000 =          MAXI    EQU 0
0001 =          MINI    EQU 1

                ;HARD-DISK SYSTEM

EC7F C32AF0     IMISYS: JMP 0F02AH

                ;EXEC SYSTEM

EC82 C324F0     EXEC:   JMP 0F024H      ;RDKCOMP

                                        ;SOFT SYSTEM
                ;
                ; RAM ZELLEN

EC85 00         DBANK:  DB 0
EC86 80         IOPB:   DB 80H  ;NORM IO
EC87 01         ION:    DB 1    ;SECTOR NR

EC88 02         IOT:    DB OFFSET ;TRK
EC89 01         IOS:    DB 1
EC8A 8000       IOD:    DW BUFF
EC8C 00         TRKBUF: DB 0
EC8D 00                 DB 0
EC8E 00                 DB 0
EC8F 00                 DB 0
EC90 01         ALTDRV: DB 1
EC91 0000       INDADR: DW 0
EC93 0000       INDADR2:DW 0

                        ENDEF
EC95+=          BEGDAT  EQU     $
EC95+           DIRBUF: DS      128     ;DIRECTORY ACCESS BUFFER
ED15+           ALV0:   DS      30
ED33+           CSV0:   DS      16
ED43+           ALV1:   DS      30
ED61+           CSV1:   DS      16
ED71+           ALV2:   DS      30
ED8F+           CSV2:   DS      16
EDA1+           ALV3:   DS      30
EDBF+           CSV3:   DS      16
EDCF+           ALV4:   DS      128
EE4F+           CSV4:   DS      256
EF4F+=          ENDDAT  EQU     $
02B8+=          DATSIZ  EQU     $-BEGDAT

EF4F                    END
```

Die Bedeutung der Register beim Einsprung sind:
HL: Die Ziel- oder Quelladresse des zu ladenden Bereichs. Jedoch werden jetzt nicht nur jeweils feste 128 Bytes übertragen, sondern, je nach Aufzeichnungsformat, automatisch so viele, wie es das Format auf der Diskette vorsieht. Die Floppy-Routinen sind damit transparent für unterschiedliche Verfahren.

D: Enthält die Track-Nummer im Bereich von 0...FF, je nach Laufwerk.
E: Enthält die Sektor-Nummer im Bereich von 0...FF (normalerweise 1...26 oder 1...16).

B: Darin steht der Befehl: 1 = Lesen; 2 = Schreiben, 0 = Setzen der Steprate. Wenn B = 0 ist, dann muß im Register D die Steprate stehen. Dabei gilt folgende Zuordnung:

D	MINI	MAXI
0	6 ms	3 ms

```
                ;* BOOT-PROGRAMM

0080                    ORG     0080H

D400 =          BDOS    EQU     0D400H
EA00 =          BIOS    EQU     0EA00H

                S0080:          ;0080
0080 318000             LXI     SP,80H
0083 210000             LXI     H,0
0086 110000             LXI     D,0     ;STEP RATE
0089 010000             LXI     B,0
008C CD24F0             CALL    0F024H
008F 2100D4             LXI     H,BDOS
0092 0634               MVI     B,34H
0094 0E00               MVI     C,0
0096 110200             LXI     D,2     ;START SEKTOR
                J0099:          ;0099
0099 C5                 PUSH    B
009A 0601               MVI     B,1     ;1=LESEN 2 = SCHREIBEN
009C D5                 PUSH    D
009D E5                 PUSH    H
009E CD24F0             CALL    0F024H
00A1 E1                 POP     H
00A2 118000             LXI     D,80H
00A5 19                 DAD     D
00A6 D1                 POP     D
00A7 C1                 POP     B
00A8 C41EF0             CNZ     0F01EH  ;BOOT ERROR
00AB 05                 DCR     B
00AC CA00EA             JZ      BIOS
00AF 1C                 INR     E
00B0 7B                 MOV     A,E
00B1 FE1B               CPI     1BH
00B3 DA9900             JC      J0099
00B6 1E01               MVI     E,1
00B8 14                 INR     D
00B9 C39900             JMP     J0099

00BC                    END
```

```
0080: 31 80 00 21 00 00 11 00 00 01 00 00 CD 24 F0 21
0090: 00 D4 06 34 0E 00 11 02 00 C5 06 01 D5 E5 CD 24
00A0: F0 E1 11 80 00 19 D1 C1 C4 1E F0 05 CA 00 EA 1C
00B0: 7B FE 1B DA 99 00 1E 01 14 C3 99 00 00 00 00 00
00C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Bild 3. Das Urlade-Programm

```
EA00: C3 DE EA C3 EE EA C3 D2 EA C3 D5 EA C3 DB EA C3
EA10: 0F F0 C3 0C F0 C3 06 F0 C3 7E EB C3 83 EB C3 99
EA20: EB C3 9E EB C3 BA EB C3 CD EB C3 38 EC C3 7C EB
EA30: C3 A3 EB 92 EA 00 00 00 00 00 00 95 EC 83 EA 33
EA40: ED 15 ED 92 EA 00 00 00 00 00 00 95 EC 83 EA 61
EA50: ED 43 ED 92 EA 00 00 00 00 00 00 95 EC 83 EA 8F
EA60: ED 71 ED 92 EA 00 00 00 00 00 00 95 EC 83 EA BD
EA70: ED 9F ED 00 00 00 00 00 00 00 00 95 EC AC EA 4D
EA80: EE CD ED 1A 00 03 07 00 EF 00 3F 00 C0 00 10 00
EA90: 02 00 01 07 0D 13 19 05 0B 11 17 03 09 0F 15 02
EAA0: 08 0E 14 1A 06 0C 12 18 04 0A 10 16 00 01 06 3F
EAB0: 03 FF 03 FF 03 F0 00 00 01 00 00 0D 0A 0A 36 30
EAC0: 6B 20 43 50 2F 4D 20 76 65 72 73 20 32 2E 32 0D
EAD0: 0A 00 C3 12 F0 CD 03 F0 E6 7F C9 C3 09 F0 31 00
EAE0: 01 21 BB EA CD C0 EB AF 32 04 00 C3 3A EB 31 80
EAF0: 00 0E 0A C5 01 00 D4 CD BA EB 0E 00 CD 83 EB 0E
EB00: 00 CD 99 EB 0E 02 CD 9E EB C1 06 2C C5 CD CD EB
EB10: C2 64 EB 2A 8A EC 11 80 00 19 44 4D CD BA EB 3A
EB20: 89 EC FE 1A DA 30 EB 3A 88 EC 3C 4F CD 99 EB AF
EB30: 3C 4F CD 9E EB C1 05 C2 0C EB 01 80 00 CD BA EB
EB40: 3E C3 32 00 00 21 03 EA 22 01 00 32 05 00 21 06
EB50: DC 22 06 00 32 38 00 21 1E F0 22 39 00 3A 04 00
EB60: 4F C3 00 D4 C1 0D CA 60 EB C5 C3 F4 EA 21 76 EB
EB70: CD C0 EB C3 1E F0 3F 42 4F 4F 54 00 AF C9 0E 00
EB80: C3 99 EB 21 00 00 79 FE 05 D0 32 85 EC 69 26 00
EB90: 29 29 29 29 11 33 EA 19 C9 21 88 EC 71 C9 21 89
EBA0: EC 71 C9 7A B3 CA, B2 EB 06 00 EB 09 7E 32 89 EC
EBB0: 6F C9 69 79 32 89 EC 26 00 C9 69 60 22 8A EC C9
EBC0: 7E B7 C8 E5 4F CD DB EA E1 23 C3 C0 EB 3A 85 EC
EBD0: FE 04 C2 F3 EB 3A 89 EC 6F 3A 88 EC 67 22 91 EC
EBE0: 21 00 00 22 93 EC 2A 8A EC 11 91 EC 0E 00 06 01
EBF0: C3 7F EC 3A 88 EC B7 C2 17 EC 3A 89 EC FE 02 C2
EC00: 17 EC 21 00 00 11 00 00 3A 85 EC 4F 06 00 E5 D5
EC10: C5 CD 82 EC C1 D1 E1 06 0A C5 2A 8A EC 3A 88 EC
EC20: 57 3A 89 EC 5F 06 01 3A 85 EC 4F CD 82 EC C1 C8
EC30: 05 C2 19 EC 3E 01 B7 C9 3A 85 EC FE 04 C2 5E EC
EC40: 3A 89 EC 6F 3A 88 EC 67 22 91 EC 21 00 00 22 93
EC50: EC 2A 8A EC 11 91 EC 0E 00 06 02 C3 7F EC 06 0A
EC60: C5 2A 8A EC 3A 88 EC 57 3A 89 EC 5F 06 02 3A 85
EC70: EC 4F CD 82 EC C1 C8 05 C2 60 EC 3E 01 B7 C9 C3
EC80: 2A F0 C3 24 F0 00 80 01 02 01 80 00 00 00 00 00
EC90: 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ECA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ECB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ECC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ECD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ECE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ECF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Bild 4. Das BIOS als Hex-Listing

1	12 ms	6 ms
2	20 ms	10 ms
3	30 ms	15 ms

Über den Einsprung INIFLP wird die langsamste Steprate nach dem Stromeinschalten vom Monitor voreingestellt.

C enthält den kombinierten Drive-Dichte-Code:
Bereich: 0, 1, 2, 3:
Laufwerk 0 ist mit Laufwerk 2 gekoppelt und Laufwerk 1 mit Laufwerk 3. Das bedeutet, daß es sich um zwei Doppelkopf-Floppys handelt, die adressiert werden sollen. Der Wert 0 bewirkt ein Select auf 0, 1 auf 1, 2 auf 2 und 3 auf 3. Der Side-Select-Ausgang (SSO) wird in diesem Modus nicht verwendet.
Bereich 10H, 11H...17H:
Die Nummern 10H, 11H und 12H, 13H usw. sind jeweils gekoppelt. Bei 10H wird Select 0 aktiviert und Side-Select (SSO) ist auf 0 (die Leitung an Pin 14 liegt dabei auf 1). Es wird die Oberfläche 0 ausgewählt, die auch bei einseitigen Floppys vorhanden ist. Bei 11H wird ebenfalls Select 0 aktiviert, jedoch SSO = 1 gesetzt, also wird die zweite Seite des Laufwerks ausgewählt. Bei 12H, 13H wird das Gerät 1 adressiert und so weiter. Damit lassen sich insgesamt 8 Oberflächen auf 4 Geräten adressieren. Die meisten Mini-Laufwerke besitzen jedoch Select 3 nicht. Dann sind nur 6 Oberflächen erreichbar.

```
                        C     INCLUDE MINIMAX         ;floppy routinen von da an
                        C
                        C     ;**********************************
                        C     ;* FLOPPY INTERFACE PAKET         *
                        C     ;* ROLF-DIETER KLEIN 820730       *
                        C     ;* DRIVES  A   B                  *
                        C     ;*           C   D                *
                        C     ;* rev 821127 fehler dop stg      *
                        C     ;* MINIMAX   kombinierte Routinen *
                        C     ;* fuer den Betrieb von Mini und  *
                        C     ;* Maxilaufwerken                 *
                        C     ;* Es wird immer der gesamte      *
                        C     ;* Sektor uebertragen             *
                        C     ;* PIO nicht mehr noetig          *
                        C     ;* rev 830318 select fehler doppel*
                        C     ;* laufwerke                      *
                        C     ;**********************************
                        C
                        C     ;***** Maxi Floppy Eigenschaften *****
                        C     ; MOTOR
                        C     ;*****                        ******
                        C     ; WICHTIG :  DIE PORTS 30H..34H  UND 40H..44H
                        C     ;            DUERFEN IN MC-COMPUTER NICHT MIT ANDEREN
                        C     ;            BAUGRUPPEN BELEGT WERDEN, DA AUF BEIDE
                        C     ;            SCHREIBEND ZUGEGRIFFEN WIRD AUCH WENN NUR
                        C     ;            MINI ODER MAX-FLOPPY VERWENDET WERDEN
                        C
                        C     ;
                        C     ; HAUPTPROGRAMM EINSPRUNG
                        C     ; DIE INTERRUPTROUTINE MUSS DURCH
                        C     ; DEN MONITOR BEREITS IN DIE ENSTPRECHENDEN
                        C     ; ZELLEN GESCHRIEBEN WORDEN SEIN
                        C     ; DIES GESCHIENT BEI EINEM CP/M BOOT
                        C     ; DURCH DEN MONITOR
                        C
                        C     ; Interruptroutine befindet sich nun mit im
                        C     ; Monitor
F108                    C     INTEX:
F108    F1              C     POP AF
F10C    DB 40           C     IN A,(40H)      ;Maxi Floppy
F10E    FB              C     EI
F10F    C9              C     RET
                        C     ;
F110                    C     INTEX1:
F110    F1              C     POP AF
F111    DB 30           C     IN A,(30H)      ;Mini Floppy
F113    FB              C     EI
F114    C9              C     RET
                        C     ;
                        C     ; Der Vektor wird dynamisch bei Anforderung an
                        C     ; die Stelle 10H geladen um fuer alle CP/M
                        C     ; Programme verwendbar zu sein
                        C     ;
F115                    C     INIFLP:
F115    3E 03           C     LD A,11b        ;maxi steprate
F117    32 F523         C     LD (MAXSTP),A   ;Voreinstellung
F11A    3E 03           C     LD A,11b
F11C    32 F524         C     LD (MINSTP),A   ;mini steprate
F11F    C9              C     RET
                        C     ;
                        C
                        C     ;
F120                    C     MINI:
F120    3E D0           C     LD A,0D0H       ;TERMINATE IF BUSY
F122    D3 40           C     OUT (040H),A    ;WICHTIG UM INTERRUPT ZU
F124    D3 30           C     OUT (030H),A    ;VERHINDERN Z.B. BEI RESET
F126    3A 0010         C     LD A,(10H)      ;RETTEN SPRUNGSTELLE
F129    32 F51E         C     LD (LOC10),A
F12C    3E C3           C     LD A,0C3H
F12E    32 0010         C     LD (10H),A
F131    E5              C     PUSH HL
F132    2A 0011         C     LD HL,(11H)
F135    22 F51F         C     LD (LOC10+1),HL
F138    21 F110         C     LD HL,INTEX1    ;JP INTEX1 AUF 10H
F13B    22 0011         C     LD (11H),HL
F13E    E1              C     POP HL
F13F    DB 40           C     IN A,(040H)
F141    DB 30           C     IN A,(030H)
F143    FB              C     EI
F144    CD F27E         C     CALL MINISOFT   ;MINIFLP
F147    F3              C     DI
F148    F5              C     PUSH AF
F149    3A F51E         C     LD A,(LOC10)
F14C    32 0010         C     LD (10H),A
F14F    E5              C     PUSH HL
F150    2A F51F         C     LD HL,(LOC10+1)
F153    22 0011         C     LD (11H),HL
F156    E1              C     POP HL
F157    F1              C     POP AF
F158    C9              C     RET
                        C     ;
                        C
F159                    C     MAXI:
F159    3E D0           C     LD A,0D0H       ;TERMINATE IF BUSY
F15B    D3 40           C     OUT (040H),A    ;
F15D    D3 30           C     OUT (030H),A
F15F    3A 0010         C     LD A,(10H)      ;RETTEN SPRUNGSTELLE
F162    32 F51E         C     LD (LOC10),A
F165    3E C3           C     LD A,0C3H
F167    32 0010         C     LD (10H),A
F16A    E5              C     PUSH HL
F16B    2A 0011         C     LD HL,(11H)
F16E    22 F51F         C     LD (LOC10+1),HL
F171    21 F108         C     LD HL,INTEX     ;JP INTEX AUF 10H
F174    22 0011         C     LD (11H),HL
F177    E1              C     POP HL
F178    DB 40           C     IN A,(040H)     ;INT LOESCHEN FALLS NOCH DA
F17A    DB 30           C     IN A,(030H)     ;BEIDE FLOPPYS GGF.
F17C    FB              C     EI
F17D    CD F28C         C     CALL MAXISOFT
F180    F3              C     DI
F181    F5              C     PUSH AF
F182    3A F51E         C     LD A,(LOC10)
F185    32 0010         C     LD (10H),A
F188    E5              C     PUSH HL
F189    2A F51F         C     LD HL,(LOC10+1)
F18C    22 0011         C     LD (11H),HL
F18F    E1              C     POP HL
F190    F1              C     POP AF
F191    C9              C     RFT
                        C     ;
                        C     ;
F192                    C     SETUP:          ;TESTET OB HEAD DOWN UND SETZT BIT
F192    DB 44           C     IN A,(44H)      ;DIREKT BIT 5 ABFRAGE IST HEADLOAD
F194    E6 20           C     AND 20H         ;=1 FALLS HEAD DOWN
F196    20 03           C     JR NZ,MSKSK
F198    3E 04           C     LD A,4          ;E=1 SETZEN
F19A    C9              C     RET
F19B                    C     MSKSK:
F19B    AF              C     XOR A           ;E=0
F19C    C9              C     RET
                        C     ;
                        C
F19D                    C     MSETUP:         ;TESTET OB HEAD DOWN UND SETZT BIT
F19D    DB 34           C     IN A,(34H)      ;DIREKT BIT 5 ABFRAGE IST HEADLOAD
F19F    E6 20           C     AND 20H         ;=1 FALLS HEAD DOWN
F1A1    20 F8           C     JR NZ,MSKSK
F1A3    3E 04           C     LD A,4          ;E=1 SETZEN
F1A5    C9              C     RET
                        C     ;
                        C
                        C
F1A6                    C     RDFLP:
F1A6    78              C     LD A,E          ;IN E SEKTOR
F1A7    D3 42           C     OUT (42H),A     ;SEKTORSPEICHER
F1A9    59              C     LD E,C          ;IN C DRIVECODE + DENSITY
F1AA    0E 43           C     LD C,43H        ;KANAL DATEN
F1AC    CD F192         C     CALL SETUP
F1AF    C6 88           C     ADD A,88H       ;LESE BEF E=X
F1B1    47              C     LD B,A
F1B2    3A F525         C     LD A,(MSIDE)    ;CODE FUER DENSITY
F1B5    B0              C     OR B
F1B6    D3 40           C     OUT (40H),A     ;CNTRL 1771 BEF
F1B8    78              C     LD A,E
F1B9    D3 44           C     OUT (44H),A     ;ENABLE NO AUTOWAIT BUT DENSE ETC
F1BB                    C     ..LP8:          ;LAENGE UNBESTIMMT BIS INT KOMMT
F1BB    DB 44           C     IN A,(44H)      ;4MHZ WAIT ON READY
F1BD    07              C     RLCA            ;1=READY DRQ
F1BE    D2 F1BB         C     JP NC,..LP8
F1C1    ED A2           C     INI
F1C3    C3 F1BB         C     JP ..LP8
```

Bild 5. Für Spezialisten: Die Floppy-Routinen aus dem Monitor isoliert aufgelistet. Für alle, die ihr eigenes BIOS konstruieren wollen

```
                        C
F1C6                    C       WRFLP:          ;C AUF DRIVE 1...4 HL AUF BUFFER
F1C6    78              C       LD A,E
F1C7    D3 42           C       OUT (42H),A     ;IN E SEKTOR
F1C9    59              C       LD E,C          ;DRIVE CODE
F1CA    0E 43           C       LD C,43H        ;LAENGE UNBESTIMMT BIS INT KOMMT
F1CC    CD F192         C       CALL SETUP      ;E=X
F1CF    C6 A8           C       ADD A,0A8H
F1D1    47              C       LD B,A
F1D2    3A F525         C       LD A,(MSIDE)
F1D5    B0              C       OR B
                        C                       ;SCHREIB BEF
F1D6    D3 40           C       OUT (40H),A
F1D8    78              C       LD A,E
F1D9    D3 44           C       OUT (44H),A     ;rev 821127
F1DB                    C       ..LLP:
F1DB    DB 44           C       IN A,(44H)
F1DD    07              C       RLCA
F1DE    D2 F1DB         C       JP NC,..LLP
F1E1    ED A3           C       OUTI
F1E3    C3 F1DB         C       JP ..LLP        ;BIS INT KOMMT
                        C
F1E6                    C       WAWA:           ;WARTESCHLEIFE
F1E6    C5              C       PUSH BC         ;6YS.12YS/DURCHL
F1E7    01 01F4         C       LD BC,500
F1EA                    C       ..LP17:
F1EA    0B              C       DEC BC
F1EB    78              C       LD A,B
F1EC    B1              C       OR C
F1ED    C2 F1EA         C       JP NZ,..LP17
F1F0    C1              C       POP BC
F1F1    C9              C       RET             ;
                        C
                        C
F1F2                    C       SEEK:           ;C DRIVE 1...4 D TRK E SEKTOR
F1F2    CD F1E6         C       CALL WAWA       ;!!!
F1F5    79              C       LD A,C          ;+ CODING AUSGEBEN
F1F6    D3 44           C       OUT (44H),A
F1F8    7B              C       LD A,E
F1F9    D3 42           C       OUT (42H),A     ;SEKTOR
F1FB    7A              C       LD A,D
F1FC    D3 43           C       OUT (43H),A     ;SUCHEN AUSFUEHREN AUF LOGIN DRIVE
F1FE    06 1C           C       LD B,1CH        ;BEFEHL SEEK
F200    3A F523         C       LD A,(MAXSTP)
F203    B0              C       OR B            ;step rate dynamisch
F204    D3 40           C       OUT (40H).A     ;BEFKANAL
F206    76              C       HALT
                        C       ;
                        C
F207                    C       RSTORE:         ;RESTORE IN CASE OF ERROR
F207    CD F1E6         C       CALL WAWA       ;!!!
F20A    79              C       LD A,C          ;DRIVE 1,2,4,8
F20B    E6 0F           C       AND 0FH
F20D    D3 44           C       OUT (44H),A
F20F    06 0C           C       LD B,0CH
F211    3A F523         C       LD A.(MAXSTP)
F214    B0              C       OR B
F215    D3 40           C       OUT (40H),A
F217    76              C       HALT
                        C       ;
                        C
F218                    C       MRDFLP:
F218    7B              C       LD A,E          ;IN E SEKTOR
F219    D3 32           C       OUT (32H),A     ;SEKTORSPEICHER
F21B    59              C       LD E,C          ;IN C DRIVECODE + DENSITY
F21C    0E 33           C       LD C,33H        ;KANAL DATEN
F21E    CD F19D         C       CALL MSETUP
F221    C6 88           C       ADD A,88H       ;LESE BEF E=X
F223    47              C       LD B,A
F224    3A F525         C       LD A.(MSIDE)    ;CODE FUER DENSITY
F227    B0              C       OR B
F228    D3 30           C       OUT (30H),A     ;CNTRL 1771 BEF
F22A    7B              C       LD A,E
F22B    D3 34           C       OUT (34H),A     ;ENABLE NO AUTOWAIT BUT DENSE ETC
F22D                    C       ..MLP8:         ;LAENGE UNBESTIMMT BIS INT KOMMT
F22D    DB 34           C       IN A,(34H)      ;4MHZ WAIT ON READY
F22F    07              C       RLCA            ;1=READY DRQ
F230    D2 F22D         C       JP NC,..MLP8
F233    ED A2           C       INI
F235    C3 F22D         C       JP ..MLP8
                        C
F238                    C       MWRFLP:         ;C AUF DRIVE 1...4 HL AUF BUFFER
F238    7B              C       LD A,E
F239    D3 32           C       OUT (32H),A     ;IN E SEKTOR
F23B    59              C       LD E,C          ;DRIVE CODE
F23C    0E 33           C       LD C,33H        ;LAENGE UNBESTIMMT BIS INT KOMMT
F23E    CD F19D         C       CALL MSETUP     ;E=X
F241    C6 A8           C       ADD A,0A8H
F243    47              C       LD B,A
F244    3A F525         C       LD A,(MSIDE)
F247    B0              C       OR B
                        C                       ;SCHREIB BEF
F248    D3 30           C       OUT (30H),A
F24A    7B              C       LD A,E
F24B    D3 34           C       OUT (34H),A     ;rev 821127
F24D                    C       ..MLLP:
F24D    DB 34           C       IN A,(34H)
F24F    07              C       RLCA
F250    D2 F24D         C       JP NC,..MLLP
F253    ED A3           C       OUTI
F255    C3 F24D         C       JP ..MLLP       ;BIS INT KOMMT
                        C       ;
                        C
F258                    C       MSEEK:          ;C DRIVE 1...4 D TRK E SEKTOR
F258    CD F1E6         C       CALL WAWA       ;!!!
F25B    79              C       LD A.C          ;+ CODING AUSGEBEN
F25C    D3 34           C       OUT (34H),A
F25E    7B              C       LD A,E
F25F    D3 32           C       OUT (32H),A     ;SEKTOR
F261    7A              C       LD A,D

F262    D3 33           C       OUT (33H).A     ;SUCHEN AUSFUEHREN AUF LOGIN DRIVE
F264    06 1C           C       LD B,1CH        ;BEFEHL SEEK
F266    3A F524         C       LD A,(MINSTP)
F269    B0              C       OR B            ;step rate dynamisch
F26A    D3 30           C       OUT (30H),A     ;BEFKANAL
F26C    76              C       HALT
                        C       ;
                        C
F26D                    C       MRSTORE:        ;RESTORE IN CASE OF ERROR
F26D    CD F1E6         C       CALL WAWA       ;!!!
F270    79              C       LD A,C          ;DRIVE 1,2,4,8
F271    E6 0F           C       AND 0FH
F273    D3 34           C       OUT (34H),A
F275    06 0C           C       LD B,0CH
F277    3A F524         C       LD A,(MINSTP)
F27A    B0              C       OR B
F27B    D3 30           C       OUT (30H),A
F27D    76              C       HALT
                        C       ;
                        C
                        C       ;
F27E                    C       MINISOFT:
F27E    CB F9           C       SET 7,C
                        C                       ; B=0 dann D=Steprate voreinst ist langsamste
                        C                       ; D=0 high speed c=3 slowest rate
F280    78              C       LD A,B          ; C = drivecode wird ignoriert
F281    B7              C       OR A
F282    20 16           C       JR NZ,XX1
                        C       ; steprate einstellen
F284    7A              C       LD A,D
F285    E6 03           C       AND 03H
F287    32 F524         C       LD (MINSTP),A ; kein restore mehr ausf, sonst ggf hangup
F28A    AF              C       XOR A
F28B    C9              C       RET
                        C
                        C       ;
F28C                    C       MAXISOFT:
F28C    CB B9           C       RES 7,C         ;GEMEINSAMER CODE
                        C                       ;B= COMMAND 0=RSTORE 1=READ 2=WRITE
                        C                       ;C= DRIVE + Coding
                        C                       ;HL POINTER AUF BUFFER
                        C                       ;D=TRACK
                        C                       ;E=SEKTOR
                        C                       ; UMRECHNEN
                        C
                        C                       ; B=0 dann D=Steprate voreinst ist langsamste
                        C                       ; D=0 high speed c=3 slowest rate
F28E    78              C       LD A,B          ; C = drivecode wird ignoriert
F28F    B7              C       OR A
F290    20 08           C       JR NZ,XX1
                        C       ; steprate einstellen
F292    7A              C       LD A,D
F293    E6 03           C       AND 03H
F295    32 F523         C       LD (MAXSTP),A   ;kein restore mehr ausf, sonst ggf hangup
F298    AF              C       XOR A
F299    C9              C       RET
                        C
                        C       ;
F29A                    C       XX1:
                        C
                        C                       ;B=1 lesen B=2 schreiben
                        C                       ;C=  0  1  2  3      0,2  1,3 gekoppelt
                        C                       ;
                        C                       ; =  10h,11h,12h,13h,14h,15h,16h,17h
                        C                       ;    mit sideselekt (10h, 11h) jeweils gekoppe
                        C                       ;
                        C                       ; =  20h,21h,22h,23h,24h,25h,26h,27h
                        C                       ;    mit selekt 3 als side sel
                        C                       ;    0,1,2 normal 20h,21h jeweils gekoppelt
                        C                       ;Bit 7,6   codierung dense,mini,maxi
                        C                       ;    1= minilaufwerk(auch bei maxieinsprg)
                        C                       ;       1= double dense
F29A    AF              C       XOR A
F29B    32 F526         C       LD (FEHZA),A
F29E    32 F525         C       LD (MSIDE),A    ;side select = 0   std
F2A1    79              C       LD A,C
                        C                       ;minimax,densecode
F2A2    E6 C0           C       AND 11000000b
F2A4    0F              C       RRCA
F2A5    0F              C       RRCA
F2A6    32 F522         C       LD (CCODE),A    ;kombinierter drive dense code
                        C                       ;codierung fuer drvcod,ccode bestimmen
                        C                       ;ccode ist c register binaerform
                        C                       ;drvcode 0..7  bit 0 ist seite des laufwerks
F2A9    79              C       LD A,C
F2AA    E6 30           C       AND 00110000B   ;=0 DANN FALL1
F2AC    20 24           C       JR NZ,SKPV1
                        C                       ;CODE 0,1,2,3    0,2   1,3 GEKOPPELT
F2AE    79              C       LD A,C
F2AF    E6 03           C       AND 000000110   ;CODING
F2B1    D9              C       EXX
F2B2    21 F2CA         C       LD HL,TABM1
F2B5    5F              C       LD E,A
F2B6    16 00           C       LD D,0
F2B8    19              C       ADD HL,DE
F2B9    19              C       ADD HL,DE
F2BA    7E              C       LD A,(HL)       ;DRVCODE
F2BB    32 F521         C       LD (DRVCODE),A
F2BE    23              C       INC HL
F2BF    5E              C       LD E,(HL)
F2C0    3A F522         C       LD A,(CCODE)
F2C3    B3              C       OR E
F2C4    32 F522         C       LD (CCODE),A    ;NEUER DRIVECODE
F2C7    D9              C       EXX
F2C8    18 49           C       JR FISKP
F2CA                    C       TABM1:
F2CA    00 01           C       DEFB 0,0001B
F2CC    02 02           C       DEFB 2,0010B
```

```
F2CE  01 04      C    DEFB 1,0100B
F2D0  03 08      C    DEFB 3,1000B
                 C    ;
F2D2             C    SKPV1:
F2D2  79         C    LD A,C
F2D3  E6 30      C    AND 00110000B
F2D5  FE 10      C    CP  00010000B     ;BEREICH 10H..17H
F2D7  20 27      C    JR NZ.SKPV2
F2D9  79         C    LD A,C
F2DA  E6 07      C    AND 000001118     ;0..7 BIT 0=SEL
F2DC  32 F521    C    LD (DRVCODE),A    ;CODING
F2DF  0F         C    RRCA              ;=CARRY DANN SIDE SEL=1
F2E0  30 05      C    JR NC,SKPV12
F2E2  3E 02      C    LD A,00000010B    ;SSO = 1 SETZEN
F2E4  32 F525    C    LD (MSIDE),A
F2E7             C    SKPV12:
F2E7  79         C    LD A,C
F2E8  0F         C    RRCA
F2E9  E6 03      C    AND 000000118     ;DRVCODE TABM1
F2EB  D9         C    EXX
F2EC  21 F2CA    C    LD HL,TABM1
F2EF  5F         C    LD E,A
F2F0  16 00      C    LD D,0
F2F2  19         C    ADD HL,DE
F2F3  19         C    ADD HL,DE
F2F4  23         C    INC HL
F2F5  5E         C    LD E,(HL)
F2F6  3A F522    C    LD A,(CCODE)
F2F9  B3         C    OR E
F2FA  D9         C    FXX
F2FB  32 F522    C    LD (CCODE),A      ;CODIERUNG
F2FE  18 13      C    JR FISKP
                 C    ;
F300             C    SKPV2:
F300  79         C    LD A,C
F301  E6 07      C    AND 000001118     ;0..7 BIT 0=SEL
F303  32 F521    C    LD (DRVCODE),A    ;CODING
F306  0F         C    RRCA              ;=CARRY DANN SIDE SEL=1
F307  30 DE      C    JR NC,SKPV12
F309  3A F522    C    LD A,(CCODE)
F30C  F6 08      C    OR 00001000B      ;SELEKT SIDE
F30E  32 F522    C    LD (CCODE),A
F311  18 D4      C    JR SKPV12
                 C    ;
                 C    ;
                 C
F313             C    FISKP:            ;DRVCODE,CCODE BESTIMMT  C REGISTER
                 C
F313  79         C    LD A,C
F314  CB 7F      C    BIT 7,A           ;<>0 dann start
F316  C2 F41C    C    JP NZ,MINEXX      ;minifloppy teil
F319  3A F521    C    LD A,(DRVCODE)
F31C  4F         C    LD C,A            ;code of drive 0..7  bit 0=side sel
                 C    ;
F31D  3A F527    C    LD A,(DRVAT)
F320  B9         C    CP C              ;GLEICHE DRIVE
F321  CA F373    C    JP Z,..SK2        ;WEITER DANN
                 C                      ; hier aber immer mit abb
                 C                      ; end rev 821127
F324  E5         C    PUSH HL
F325  C5         C    PUSH BC
F326  21 F528    C    LD HL,DRVTAB      ;TABELLEN INDEX BERECHNEN
F329  3A F527    C    LD A,(DRVAT)
F32C  FE FF      C    CP 0FFH           ;UNDEF
F32E  28 0A      C    JR Z.NOD
F330  0F         C    RRCA              ;
F331  E6 03      C    AND 00000011b
F333  4F         C    LD C,A
F334  06 00      C    LD B.0
F336  09         C    ADD HL,BC
F337  DB 41      C    IN A.(41H)        ;ALTER TRACK
F339  77         C    LD (HL),A         ;RETTEN VOR RSTORE AUSFUEHRUNG NOETIG
F33A             C    NOD:              ;UNDEF                          AUFR
F33A  C1         C    POP BC
F33B  E1         C    POP HL            ;ERSTMAL ALTE DRIVE RETTEN
F33C  E5         C    PUSH HL
F33D  C5         C    PUSH BC
F33E  21 F528    C    LD HL,DRVTAB
F341  79         C    LD A,C
F342  0F         C    RRCA              ;
F343  E6 03      C    AND 00000011b
F345  4F         C    LD C,A
F346  06 00      C    LD B,0            ;IN C NEUE DRIVE
F348  09         C    ADD HL,BC
F349  7E         C    LD A,(HL)
F34A  D3 41      C    OUT (41H),A       ;NEUER TRACK HOLEN
F34C  C1         C    POP BC
F34D  E1         C    POP HL
F34E  79         C    LD A,C
F34F  32 F527    C    LD (DRVAT),A      ;NUN NEUE DRIVE
F352  F5         C    PUSH AF           ;NEU BUG 810917
F353  C5         C    PUSH BC           ;BETREFF HEADLOAD
F354  C5         C    PUSH BC           ;rev 821127
F355  01 00FA    C    LD BC,1000*3/12   ;1MS  BEI 6MHZ AUCH
F358             C    ..LPPD:
F358  0B         C    DEC BC
F359  78         C    LD A,B
F35A  B1         C    OR C
F35B  C2 F358    C    JP NZ,..LPPD      ;WARTEN TUNNELERASE COMPLETED
F35E  C1         C    POP BC            ;rev 821127
F35F  3A F522    C    LD A,(CCODE)      ;rev 830318 vorher falsches reg
                 C                      ;NEUE DRIVE CODED 1,2,4,8
F362  E6 0F      C    AND 0FH           ;AUSGEBEN
F364  D3 44      C    OUT (44H),A       ;OHNE AUTOWAIT
                 C    ;
F366  01 0865    C    LD BC,2917        ;ca 35 MS WARTEN UNKRIT. (10MS..35MS)
F369             C    ..LP4:            ;WICHTIG ABER BEI SCHREIBEN NACH LESEN
F369  0B         C    DEC BC            ;FALLS DRIVES HEADLOAD BESITZEN
F36A  78         C    LD A,B
F36B  B1         C    OR C
F36C  C2 F369    C    JP NZ...LP4       ;DA SONST SETTLE NICHT OK
F36F  C1         C    POP BC

F370  F1         C    POP AF            ;END 810917
                 C                      ;rev 3.2
F371  18 06      C    JR ..SKKK1        ;bei laufwerkswechsel kann head load down
                 C                      ;sein aber laufwerk nicht ready
                 C                      ;daher immer dummy seek ausfuehren
                 C
F373             C    ..SK2:            ;rev 3.2 ohne PIO
F373  DB 44      C    IN A,(44H)        ;wenn head load vorliegt kann
F375  E6 20      C    AND 20H           ;ready nicht ungueltig sein wegen reihenfolge
F377  20 0D      C    JR NZ,..OV        ;fertig dann
                 C
F379             C    ..SKKK1:          ;warte schleife seek solange bis status
                 C                      ;ein ready angibt
                 C
F379  3A F522    C    LD A,(CCODE)
F37C  4F         C    LD C,A            ;drive code internal
F37D  CD F411    C    CALL SELDUM       ;SELEKT DUMMY AUSFUEHREN mit seek
                 C
F380  DB 40      C    IN A,(40H)        ;bit 7 entscheidet
F382  E6 80      C    AND 80H
F384  20 F3      C    JR NZ,..SKKK1     ;bis ausgefuehrt
                 C                      ;drive wird nicht not ready ohne
                 C                      ;das headload weggeht im normal fall
                 C                      ;da nur MOTOR OFF automatik hier
                 C                      ;wesentlich ist
                 C
F386             C    ..OV:
F386  DB 41      C    IN A,(41H)        ;TRACK HOLEN
F388  FE FF      C    CP 0FFH           ;AUS TRACKTABELLE
F38A  28 15      C    JR Z,..TRY1       ;ERST RESTORE
F38C  BA         C    CP D              ;GLEICH DANN WEITER
F38D  CA F3BE    C    JP Z,..SK11       ;KEIN SUCHEN NOETIG
                 C                      ;SUCHEN NOETIG
F390             C    ..TRY:
F390  3A F522    C    LD A,(CCODE)
F393  4F         C    LD C,A
F394  E5         C    PUSH HL
F395  D5         C    PUSH DE
F396  C5         C    PUSH BC
F397  CD F1F2    C    CALL SEEK
F39A  C1         C    POP BC
F39B  D1         C    POP DE
F39C  E1         C    POP HL
F39D  E6 90      C    AND 10010000B
F39F  28 1D      C    JR Z,..SK11       ;OK WEITER
F3A1             C    ..TRY1:           ;EINGANG BEI LESEFEHLER UND SCHR8FEHLER
F3A1  3A F522    C    LD A,(CCODE)
F3A4  4F         C    LD C.A
                 C                      ;FEHLER AUFGETRETEN
                 C                      ;RSTORE AUSFUEHREN
F3A5  E5         C    PUSH HL
F3A6  D5         C    PUSH DE
F3A7  C5         C    PUSH BC
F3A8  CD F207    C    CALL RSTORE
F3AB  C1         C    POP BC
F3AC  D1         C    POP DE
F3AD  E1         C    POP HL
F3AE  3A F526    C    LD A,(FEHZA)      ;FEHLER COUNT
F3B1  3C         C    INC A
F3B2  32 F526    C    LD (FEHZA),A
F3B5  FE 0A      C    CP 10
F3B7  38 07      C    JR C,..TRY
F3B9  3E FF      C    LD A,0FFH
F3BB  B7         C    OR A
F3BC  37         C    SCF
F3BD  C9         C    RET               ;FEHLER
                 C
                 C    ;
F3BE             C    ..SK11:           ;KEIN SEEK WAR NOETIG
F3BE  78         C    LD A,B            ;BEFEH
F3BF  FE 02      C    CP 2
F3C1  CA F3EC    C    JP Z...WRTEX
F3C4  3A F522    C    LD A,(CCODE)
F3C7  4F         C    LD C,A
F3C8  E5         C    PUSH HL
F3C9  D5         C    PUSH DE
F3CA  C5         C    PUSH BC
F3CB  CD F1A6    C    CALL RDFLP
F3CE  C1         C    POP BC
F3CF  D1         C    POP DE
F3D0  E1         C    POP HL
F3D1  E6 9C      C    AND 100111008
F3D3  C8         C    RET Z
F3D4  E6 10      C    AND 00010000b     ;record not found then seek
F3D6  20 C9      C    JR NZ...TRY1      ;restore first
F3D8  3A F526    C    LD A,(FEHZA)
F3DB  3C         C    INC A
F3DC  32 F526    C    LD (FEHZA),A
F3DF  FE 05      C    CP 5
F3E1  38 DB      C    JR C,..SK11
F3E3  FE 0A      C    CP 10
F3E5  38 BA      C    JR C,..TRY1
F3E7  3E FF      C    LD A,0FFH
F3E9  B7         C    OR A
F3EA  37         C    SCF
F3EB  C9         C    RET
                 C    ;
F3EC             C    ..WRTEX:
F3EC  3A F522    C    LD A,(CCODE)
F3EF  4F         C    LD C,A
F3F0  E5         C    PUSH HL
F3F1  D5         C    PUSH DE
F3F2  C5         C    PUSH BC
F3F3  CD F1C6    C    CALL WRFLP
F3F6  C1         C    POP BC
F3F7  D1         C    POP DE
F3F8  E1         C    POP HL
F3F9  E6 FC      C    AND 11111100B
F3FB  C8         C    RET Z
F3FC  E6 10      C    AND 00010000b     ;3.2 seek err on RNF half errcount
F3FE  20 A1      C    JR NZ,..TRY1      ;restore seek first
F400  3A F526    C    LD A,(FEHZA)
```

```
F403  3C          C     INC A
F404  32 F526     C     LD (FEHZA).A
F407  FE 04       C     CP 4
F409  DA F3A1     C     JP C,..TRY1     ;bei write gleich rstore
F40C  3E FF       C     LD A,0FFH       ;r/o error
F40E  B7          C     OR A
F40F  37          C     SCF
F410  C9          C     RET
                  C     ;
                  C     ;
F411              C     SELDUM:         ;WIE SEEK AUSFUEHREN
F411  E5          C     PUSH HL
F412  D5          C     PUSH DE
F413  C5          C     PUSH BC
F414  CD F1F2     C     CALL SEEK       ;IGNORE FEHLER
F417  C1          C     POP BC
F418  D1          C     POP DE
F419  E1          C     POP HL
F41A  C9          C     RET
F41B  C9          C     RET
                  C     ;
                  C
                  C     ;
                  C
F41C              C     MINEXX:
F41C  3A F521     C     LD A,(DRVCODE)
F41F  4F          C     LD C,A          ;code of drive 0..7  bit 0=side sel
                  C     ;
F420  3A F530     C     LD A,(MDRVAT)
F423  B9          C     CP C            ;GLEICHE DRIVE
F424  CA F476     C     JP Z,..MSK2     ;WEITER DANN
                  C                     ; hier aber immer mit abb
                  C                     ; end rev 821127
F427  E5          C     PUSH HL
F428  C5          C     PUSH BC
F429  21 F531     C     LD HL,MDRVTAB   ;TABELLEN INDEX BERECHNEN
F42C  3A F530     C     LD A,(MDRVAT)
F42F  FE FF       C     CP 0FFH         ;UNDEF
F431  28 0A       C     JR Z,MNOD
F433  0F          C     RRCA            ;
F434  E6 03       C     AND 00000011b
F436  4F          C     LD C,A
F437  06 00       C     LD B,0
F439  09          C     ADD HL,BC
F43A  DB 31       C     IN A,(31H)      ;ALTER TRACK
F43C  77          C     LD (HL),A       ;RETTEN VOR RSTORE AUSFUEHRUNG NOETIG
F43D              C     MNOD:           ;UNDEF NICHT ZURUECK            ABER
F43D  C1          C     POP BC
F43E  E1          C     POP HL          ;ERSTMAL ALTE DRIVE RETTEN
F43F  E5          C     PUSH HL
F440  C5          C     PUSH BC
F441  21 F531     C     LD HL,MDRVTAB
F444  79          C     LD A,C
F445  0F          C     RRCA            ;
F446  E6 03       C     AND 00000011b
F448  4F          C     LD C,A
F449  06 00       C     LD B,0          ;IN C NEUE DRIVE
F44B  09          C     ADD HL,BC
F44C  7E          C     LD A,(HL)
F44D  D3 31       C     OUT (31H),A     ;NEUER TRACK HOLEN
F44F  C1          C     POP BC
F450  E1          C     POP HL
F451  79          C     LD A,C
F452  32 F530     C     LD (MDRVAT),A   ;NUN NEUE DRIVE
F455  F5          C     PUSH AF         ;NEU BUG 810917
F456  C5          C     PUSH BC         ;BETREFF HEADLOAD
F457  C5          C     PUSH BC         ;rev 821127
F458  01 00FA     C     LD BC,1000*3/12 ;1MS  BEI 6MHZ AUCH
F458              C     ..MLPPD:
F458  0B          C     DEC BC
F45C  78          C     LD A,B
F45D  B1          C     OR C
F45E  C2 F458     C     JP NZ,..MLPPD   ;WARTEN TUNNELERASE COMPLETED
F461  C1          C     POP BC          ;rev 82112
F462  3A F522     C     LD A,(CCODE)    ;rev 830318 vorher falscher code
                  C                     ;NEUE DRIVE CODED 1,2,4,8
F465  E6 0F       C     AND 0FH         ;AUSGEBEN
F467  D3 34       C     OUT (34H),A     ;OHNE AUTOWAIT
                  C     ;
F469  01 0865     C     LD BC,2917      ;ca 35 MS WARTEN UNKRIT. (10MS..35MS)
F46C              C     ..MLP4:         ;WICHTIG ABER BEI SCHREIBEN NACH LESEN
F46C  0B          C     DEC BC          ;FALLS DRIVES HEADLOAD BESITZEN
F46D  78          C     LD A,B
F46E  B1          C     OR C
F46F  C2 F46C     C     JP NZ,..MLP4    ;DA SONST SETTLE NICHT OK
F472  C1          C     POP BC
F473  F1          C     POP AF          ;END 810917
                  C                     ;rev 3.2 830205
F474  18 06       C     JR ..MSKKK1     ;bei laufwerkswechsel kann head load
                  C                     ;down sein aber dennoch drive not ready
                  C                     ;daher immer dummy seek ausfuehren
F476              C     ..MSK2:
                  C                     ;rev 3.2 ohne PIO
F476  DB 34       C     IN A,(34H)      ;wenn head load vorliegt kann
F478  E6 20       C     AND 20H         ;ready nicht ungueltig sein wegen
F47A  20 0D       C     JR NZ,..MOV     ;fertig dann          reihenfolge
                  C
F47C              C     ..MSKKK1:       ;warte schleife seek solange bis status
                  C                     ;ein ready angibt
                  C
F47C  3A F522     C     LD A,(CCODE)
F47F  4F          C     LD C,A          ;drive code internal
F480  CD F514     C     CALL MSELDUM    ;SELEKT DUMMY AUSFUEHREN mit seek
                  C
F483  DB 30       C     IN A,(30H)      ;bit 7 entscheidet
F485  E6 80       C     AND 80H
F487  20 F3       C     JR NZ,..MSKKK1  ;bis ausgefuehrt
                  C                     ;drive wird nicht not ready ohne
                  C                     ;das headload weggeht in normal fall
                  C                     ;da nur MOTOR OFF automatik hier
                  C                     ;wesentlich ist
F489              C     ..MOV:
F489  DB 31       C     IN A,(31H)      ;TRACK HOLEN
F48B  FE FF       C     CP 0FFH
F48D  28 15       C     JR Z,..MTRY1    ;UNDEF DANN RESTORE
F48F  BA          C     CP D            ;GLEICH DANN WEITER
F490  CA F4C1     C     JP Z,..MSK11    ;KEIN SUCHEN NOETIG
                  C                     ;SUCHEN NOETIG
F493              C     ..MTRY:
F493  3A F522     C     LD A,(CCODE)
F496  4F          C     LD C,A
F497  E5          C     PUSH HL
F498  D5          C     PUSH DE
F499  C5          C     PUSH BC
F49A  CD F258     C     CALL MSEEK
F49D  C1          C     POP BC
F49E  D1          C     POP DE
F49F  E1          C     POP HL
F4A0  E6 90       C     AND 10010000B
F4A2  28 10       C     JR Z,..MSK11    ;OK WEITER
F4A4              C     ..MTRY1:              ;EINGANG BEI LESEFEHLER UND
F4A4  3A F522     C     LD A,(CCODE)                        SCHREIBFEHLER
F4A7  4F          C     LD C,A
                  C                     ;FEHLER AUFGETRETEN
                  C                     ;RSTORE AUSFUEHREN
F4A8  E5          C     PUSH HL
F4A9  D5          C     PUSH DE
F4AA  C5          C     PUSH BC
F4AB  CD F26D     C     CALL MRSTORE
F4AE  C1          C     POP BC
F4AF  D1          C     POP DE
F4B0  E1          C     POP HL
F4B1  3A F526     C     LD A,(FEHZA)    ;FEHLER COUNT
F4B4  3C          C     INC A
F4B5  32 F526     C     LD (FEHZA),A
F4B8  FE 0A       C     CP 10
F4BA  38 D7       C     JR C,..MTRY
F4BC  3E FF       C     LD A,0FFH
F4BE  B7          C     OR A
F4BF  37          C     SCF
F4C0  C9          C     RET             ;FEHLER
                  C
                  C
                  C
F4C1              C     ..MSK11:        ;KEIN SEEK WAR NOETIG
F4C1  78          C     LD A,B          ;BEFEHL
F4C2  FE 02       C     CP 2
F4C4  CA F4EF     C     JP Z,..MWRTEX
F4C7  3A F522     C     LD A,(CCODE)
F4CA  4F          C     LD C,A
F4CB  E5          C     PUSH HL
F4CC  D5          C     PUSH DE
F4CD  C5          C     PUSH BC
F4CE  CD F218     C     CALL MRDFLP
F4D1  C1          C     POP BC
F4D2  D1          C     POP DE
F4D3  E1          C     POP HL
F4D4  E6 9C       C     AND 100111008
F4D6  C8          C     RET Z
F4D7  E6 10       C     AND 000100008   ;3.2 rstore seek
F4D9  20 C9       C     JR NZ,..MTRY1
F4DB  3A F526     C     LD A,(FEHZA)
F4DE  3C          C     INC A
F4DF  32 F526     C     LD (FEHZA),A
F4E2  FE 05       C     CP 5
F4E4  38 DB       C     JR C,..MSK11
F4E6  FE 0A       C     CP 10
F4E8  38 BA       C     JR C,..MTRY1
F4EA  3E FF       C     LD A,0FFH
F4EC  B7          C     OR A
F4ED  37          C     SCF
F4EE  C9          C     RET
                  C
                  C
                  C     ;
F4EF              C     ..MWRTEX:
F4EF  3A F522     C     LD A,(CCODE)
F4F2  4F          C     LD C,A
F4F3  E5          C     PUSH HL
F4F4  D5          C     PUSH DE
F4F5  C5          C     PUSH BC
F4F6  CD F238     C     CALL MWRFLP
F4F9  C1          C     POP BC
F4FA  D1          C     POP DE
F4FB  E1          C     POP HL
F4FC  E6 FC       C     AND 111111008
F4FE  C8          C     RET Z
F4FF  E6 10       C     AND 000100008   ;3.2 restore seek
F501  20 A1       C     JR NZ,..MTRY1   ;on seek error RNF no double count here
F503  3A F526     C     LD A,(FEHZA)
F506  3C          C     INC A
F507  32 F526     C     LD (FEHZA),A
F50A  FE 04       C     CP 4
F50C  DA F4A4     C     JP C,..MTRY1    ;bei write gleich rstore
F50F  3E FF       C     LD A,0FFH       ;r/o error
F511  B7          C     OR A
F512  37          C     SCF
F513  C9          C     RET
                  C     ;
                  C     ;
F514              C     MSELDUM:        ;WIE SEEK AUSFUEHREN
F514  E5          C     PUSH HL
F515  D5          C     PUSH DE
F516  C5          C     PUSH BC
F517  CD F258     C     CALL MSEEK      ;IGNORE FEHLER
F51A  C1          C     POP BC
F51B  D1          C     POP DE
F51C  E1          C     POP HL
F51D  C9          C     RET
                  C
                  C
F51E  00 00 00    C     LOC10:  DEFB 0,0,0      ;SPRUNG AUSTAUSCHSTELLE
F521  00          C     DRVCODE: DEFB 0         ;temp phys drive code
```

```
F522  00        C     CCODE:  DEFB 0          ;code fuer ausgabeport
F523  00        C     MAXSTP: DEFB 0
F524  00        C     MINSTP: DEFB 0
F525  00        C     MSIDE:  DEFB 0          ;flags fuer adressierung etc
F526  00        C     FEHZA:  DEFB 0
F527  FF        C     DRVAT:  DEFB 0FFH       ;0..7 STD WERT ZWSPEICHER
                C                             ;DANN IST RSTORE NOETIG
                C

F528  FF FF FF FF   C     DRVTAB: DEFB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
F52C  FF FF FF FF   C
F530  FF            C     NDRVAT: DEFB 0FFH       ;1...8 STD WERT ZWSPEICHER
                    C                             ;DANN IST RSTORE NOETIG
                    C
F531  FF FF FF FF   C     NDRVTAB: DEFB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
F535  FF FF FF FF   C
                    C     ;
                    C
                    C
                    C
                          ,
```

Bereich 20H, 21H...27H:

Es wird anstelle von SSO der Ausgang Select 3 zur Seitenwahl verwendet. Wichtig ist dabei, daß auf der zweiten Seite nun ein anderes Format stehen muß, wie es zum Beispiel beim Alphatronik oder beim ITT-3030 der Fall ist. Dort ist das Side-Select-Bit auf 0 gesetzt. Die Werte 26H und 27H sind hier nicht mehr sinnvoll, da Select 3 nicht mehr als Drive-Select zur Verfügung steht.

Mit Bit 6 des Registers C kann eingestellt werden, ob einfache Dichte oder doppelte Dichte verwendet werden soll. Ist das Bit auf 0 gesetzt, so wird einfache Schreib-Dichte verwendet, ist es auf 1 gesetzt, dann wird doppelte Dichte gefahren. Bit 7 ist als internes Flag reserviert.

Wie ein Sektor adressiert wird

Es soll ein Sektor von Adresse 2000H auf die Floppy, Gerät 1, Seite 0 in Double Dense auf Track 3, Sektor 5 geschrieben werden:

```
LD HL, 2000H    ; Adresse 2000H
LD D, 3         ; Track 3
LD E, 5         ; Sektor 5
LD B, 2         ; Schreiben
LD C, 12H + 40H ; (52H) double
                  dense LW2
CALL MINI       ; oder MAXI
```

Es werden dabei 256 Bytes geschrieben, wenn die Floppy so formatiert wurde.

Steprate: programmierbar

Wie wir vorher gesehen haben, ist die Steprate programmierbar. Beim Booten des CP/M-Systems wird zunächst die langsamste Rate verwendet, so daß alle Laufwerke verwendet werden können. Im Boot-Programm kann dann die Steprate neu definiert werden. Auf Adresse 86H des Cold-Boots wird das Registerpaar DE mit 0 geladen, also eine Steprate von 3 ms verwendet. Bei langsameren Laufwerken muß hier der entsprechende Wert verwendet werden. Nach dem Warm-Boot wird auf Adresse EC05H des BIOS ebenfalls das Registerpaar DE mit 0 geladen, also auch 3 ms. Auch hier muß bei langsamen Laufwerken ins Register D ein größerer Wert (0...3) geladen werden, sonst wird nach CTRL-C eine andere Steprate verwendet als nach dem Kalt-Start.

Bild 6 zeigt noch ein Programm, das im CP/M-TPA-Bereich abläuft. Es formatiert eine Diskette. Alle wichtigen Werte werden dabei vom Programm interaktiv abgefragt. Das Programm verwendet Monitor-Einsprünge und ist daher nur in unserem System lauffähig.

```
rom  abs                                                        checksum
0100 C3 CA 01 C3 09 F0 CD 03 F0 E6 7F C9 7E B7 C8 4F   += 0984
0110 23 CD 03 01 18 F6 0E 0D CD 03 01 0E 0A C3 03 01   += 03CD
0120 0D 0A 4D 61 78 69 2D 46 6C 6F 70 70 79 20 46 6F   += 0522
0130 72 6D 61 74 74 65 72 20 20 52 6F 6C 66 2D 44 2E   += 0571
0140 4B 4C 65 69 6E 20 32 2E 31 0D 0A 00 73 69 6E 67   += 044C
0150 6C 65 20 64 65 6E 73 65 20 3D 20 31 20 64 6F 75   += 0516
0160 62 6C 65 20 3D 20 32 20 20 74 72 6B 30 20 73 64   += 049A
0170 20 72 65 73 74 20 64 62 20 3D 20 33 20 00 0D 0A   += 03AB
0180 41 6E 7A 61 68 6C 20 53 70 75 72 65 6E 20 37 37   += 0589
0190 3D 31 20 20 00 0D 0A 46 6F 72 6D 61 74 69 65 72   += 046E
01A0 65 6E 20 20 44 52 49 56 45 20 41 2C 42 2C 43 2C   += 03F7
01B0 44 20 20 65 6E 64 65 20 3D 20 45 20 00 4A 41 20   += 03AD
01C0 3D 20 4A 20 00 F1 DB 40 FB C9 DB 30 DB 40 3E C3   += 07BE
01D0 32 10 00 21 C5 01 22 11 00 21 20 01 CD 0C 01 21   += 0299
01E0 4C 01 CD 0C 01 CD 06 01 4F CD 03 01 FE 31 38 DA   += 055C
01F0 FE 34 30 D6 E6 0F 32 4F 06 21 7E 01 CD 0C 01 CD   += 05F8
0200 06 01 4F CD 03 01 FE 31 20 04 3E 4D 18 02 18 BA   += 03F1
0210 32 50 06 21 95 01 CD 0C 01 CD 06 01 4F CD 03 01   += 0400
0220 FE 45 CA 1E F0 FE 41 DA CA 01 FE 45 D2 CA 01 D6   += 09B5
0230 41 F5 CD 16 01 21 BD 01 CD 0C 01 F1 57 CD 06 01   += 05EF
0240 4F CD 03 01 FE 4A C2 CA 01 CD 16 01 4A CD 53 02   += 0645
0250 C3 13 02 3E 00 32 C0 04 79 3C FE 01 28 0E FE 02   += 04F6
0260 28 0A FE 03 20 04 3E 04 18 02 3E 08 4F C5 79 E6   += 046C
0270 0F F6 10 D3 44 CD 95 03 CD 95 03 DB 40 E6 80 20   += 0797
0280 F7 3A 4F 06 FE 01 CA F2 02 FE 03 CA F2 02 21 C1   += 07E4
0290 04 11 52 06 01 20 00 ED B0 C1 16 00 1E 01 21 72   += 03B4
02A0 06 E5 D5 C5 EB 21 E1 04 01 6E 01 ED B0 C1 D1 E1   += 08F6
02B0 E5 DD E1 D5 DD 72 10 DD 73 12 3A C0 04 DD 77 11   += 089C
02C0 11 6E 01 19 D1 1C 7B FE 1B C2 A1 02 C5 01 58 02   += 059F
02D0 36 4E 23 0B 79 B0 C2 D0 02 C1 E5 D5 C5 CD 7D 03   += 07FC
02E0 C1 D1 E1 14 3A 50 06 3C BA CA 6D 03 CD A5 03 C3   += 077F
02F0 9C 02 21 F9 03 11 72 06 01 10 00 ED B0 C1 16 00   += 04C9
0300 1E 01 21 82 06 E5 D5 C5 EB 21 09 04 01 B7 00 ED   += 0605
0310 B0 C1 D1 E1 E5 DD E1 D5 DD 72 07 DD 73 09 3A C0   += 0A44
0320 04 DD 77 08 11 B7 00 19 D1 1C 7B FE 1B C2 05 03   += 058C
0330 C5 01 2C 01 36 FF 23 0B 79 B0 C2 34 03 C1 E5 D5   += 06F3
0340 C5 CD 89 03 C1 D1 E1 14 3A 50 06 3C BA CA 6D 03   += 0765
0350 CD A5 03 3A 4F 06 FE 03 C2 00 03 C5 21 C1 04 11   += 0586
0360 52 06 01 20 00 ED B0 C1 16 01 C3 9C 02 CD 95 03   += 05B4
0370 C9 C5 01 05 0D 0B 78 B1 C2 75 03 C1 C9 CD 71 03   += 06DA
0380 FB 21 52 06 CD BA 03 F3 C9 CD 71 03 FB 21 72 06   += 078F
0390 CD D9 03 F3 C9 DB 30 CD 71 03 FB CD A0 03 F3 C9   += 09D8
03A0 3E 0A D3 40 76 CD 71 03 FB CD AE 03 F3 C9 79 E6   += 08A6
03B0 0F F6 10 D3 44 3E 5A D3 40 76 3A C0 04 07 E6 02   += 063A
03C0 F6 F4 D3 40 79 E6 0F F6 10 0E 43 D3 44 DB 44 E6   += 08DE
03D0 80 CA CD 03 ED A3 C3 CD 03 3A C0 04 07 E6 02 F6   += 0820
03E0 F4 D3 40 79 E6 0F F6 00 0E 43 D3 44 DB 44 E6 80   += 08S8
03F0 CA EC 03 ED A3 C3 EC 03 00 FF FF FF FF FF FF FF   += 0BF4
0400 FF FF FF FF FF FF FF FF FF FF 00 00 00 00 00 FE   += 09F5
0410 00 00 00 00 F7 FF FF FF FF FF FF FF FF FF FF FF   += 0BEC
0420 00 00 00 00 00 00 FB E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0908
0430 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0440 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0450 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0460 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0470 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0480 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0490 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
04A0 E5 E5 E5 E5 E5 E5 E5 F7 FF FF FF FF FF FF FF FF   += 0F32
04B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   += 0FF0
04C0 00 4E 4E 4E 4E 4E 4E 4E 0E 4E 4E 4E 4E 4E 4E 4E   += 0452
04D0 0E 4E 4E 4E 4E 4E 4E 4E 0E 4E 4E 4E 4E 4E 4E 4E   += 0460
04E0 DE 00 00 00 00 00 00 00 00 00 00 00 00 F5 F5 F5   += 02ED
04F0 FE 00 00 00 01 F7 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   += 0502
0500 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 00 00 00 00   += 03A8
0510 00 00 00 00 00 00 00 00 F5 F5 F5 FB E5 E5 E5 E5   += 076E
0520 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0530 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0540 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0550 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0560 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0570 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0580 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0590 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
05A0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
05B0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
05C0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
05D0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
05E0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
05F0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0600 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0610 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 F7 4E 4E 4E   += 0C9D
0620 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   += 04E0
0630 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   += 04E0
0640 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 00   += 0492
0650 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   += 0000
0660 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   += 0000
0670 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   += 0000
0680 00 00 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0C86
0690 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 F7 4E 4E 4E   += 0C9D
06A0 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   += 04E0
06B0 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   += 04E0
06C0 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 00   += 0492
06D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   += 0000
06E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   += 0000
06F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   += 0000
0700
```

Bild 6. Dieses Programm ist nur für Ihre Werkstatt gedacht. Es läuft zwar in der CP/M-TPA, ist aber nicht mit anderen CP/Ms verträglich, weil es die CP/M-Funktionen umgehen muß und sich direkt an unseren Monitor und Floppyports wendet. Es formatiert 8-Zoll-Disketten

Tabelle 2
ANSI-Standard bei 5¼-Zoll-Laufwerken

Signal-Pin	Masse-Pin	Bedeutung
2	1	nicht verwendet
4	3	(In User Control)
6	5	Drive Select 3
8	7	Index/(Sektor)
10	9	Drive Select 0
12	11	Drive Select 1
14	13	Drive Select 2
16	15	Motor on
18	17	Direction Select
20	19	Step
22	21	Composite Write Data
24	23	Write Gate
26	25	Track 0
28	27	Write protected
30	29	Composite Read Data
32	31	Side One Select
34	33	(Disk Change)

Bei Doppelkopf-Laufwerken ist der Pin „Side One Select" wichtig. Man kann ihn vom Ausgang SSO am Floppy Controller her mit Zwischenschalten eines Gatters ansteuern. Siehe überarbeiteten Schaltplan in diesem Heft. Das RDK-Bios, Version Mini 1, unterstützt die Floppy-Seiten-Ansteuerung über SSO. Bei unserem Bios kann auch Ausgang Select 3 zur Seitenanwahl benutzt werden. Man muß diesen Ausgang dann mit Side One Select verbinden. Die meisten 8-Zoll-Laufwerke können die Seitenanwahl auch direkt von den Select-Anschlüssen her ableiten. Anhand des Benutzerhandbuches muß man bei jedem einzelnen Laufwerk herausfinden, wie das geht. Auf alle Fälle sieht dann eine doppelseitig beschreibbare Diskette vom Benutzer her wie zwei Stationen aus (deren Disketten aber nicht separat gewechselt werden können).

Da die Mini-Floppy-Laufwerke und die großen Laufwerke weitgehend mit identischen Signalen versorgt werden müssen, kann man eine Adapterplatine entwerfen, die die Signale vom 50poligen Stecker, wie ihn die mc-Floppy-Controller-Platine erwartet, auf den 34poligen Stecker bringt. *Bild 1* zeigt einen Vorschlag von der Bestückungsseite her, *Bild 2* die Lötseite. Von der Lötseite her wird eine 50polige Buchse eingelötet, von der Bestückungsseite her eine 34polige Stiftleiste. Die Platine ist durchkontaktiert. Auf der Platine ist Pin 16 gegen Masse geschaltet, was den Motor des Laufwerkes permanent laufen läßt. Wer das ändern will, der benötigt ein Bios mit „Timern", das in der Lage ist, den Motor sinnvoll ein- und auszuschalten – was nicht ganz einfach herzustellen ist. Darüber hinaus muß auf der Platine die Verbindung von Pin 16 zur Masse aufgetrennt werden.

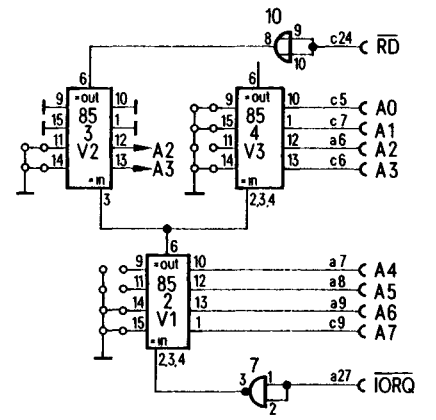


Bild 3. Die Adreß-Jumpen an V 1 müssen geändert werden, damit der Controller sich bei 30H meldet

Der Floppy-Controller für Minilaufwerke enthält eine neue Basisadresse 30H (bis 34H). Dazu muß auf der Floppy-Karte eine andere Adresse eingestellt werden als bisher angegeben. *Bild 3* zeigt die Brückenbelegung, die von der

```
rom  abs                                                        checksum
0100 C3 D0 01 C3 09 F0 CD 03 F0 E6 7F C9 7E B7 C8 4F   += 098A
0110 23 CD 03 01 18 F6 0E 0D CD 03 01 0E 0A C3 03 01   += 03CD
0120 0D 0A 4D 69 6E 69 20 46 6C 6F 70 70 79 20 46 6F   += 0520
0130 72 6D 61 74 74 65 72 20 20 52 6F 6C 66 2D 44 2E   += 0571
0140 4B 4C 65 69 6E 20 31 2E 31 0D 0A 00 73 69 6E 67   += 044B
0150 6C 65 20 64 65 6E 73 65 20 3D 20 31 20 64 6F 75   += 0516
0160 62 6C 65 20 3D 20 32 20 20 45 43 4D 41 20 37 30   += 03B8
0170 20 3D 20 33 20 00 0D 0A 41 6E 7A 61 68 6C 20 53   += 0433
0180 70 75 72 65 6E 20 33 35 3D 31 20 34 30 3D 32 20   += 0336
0190 38 30 3D 33 20 39 36 3D 34 20 00 0D 0A 46 6F 72   += 052F
01A0 6D 61 74 69 65 72 65 6E 20 20 44 52 49 56 45 20   += 03E7
01B0 41 2C 42 2C 43 2C 44 20 20 65 6E 64 65 20 3D 20   += 0597
01C0 45 20 00 4A 41 20 3D 20 4A 20 00 F1 DB 30 FB C9   += 04AA
01D0 DB 30 DB 40 3E C3 32 10 00 21 CB 01 22 11 00 21   += 0433
01E0 20 01 CD 0C 01 21 4C 01 CD 0C 01 CD 06 01 4F CD   += 063C
01F0 03 01 FE 31 38 DA FE 34 30 D6 E6 0F 32 71 06 21   += 0498
0200 76 01 CD 0C 01 CD 06 01 4F CD 03 01 FE 31 20 04   += 03C4
0210 3E 23 18 16 FE 32 20 04 3E 28 18 0E FE 33 20 04   += 04AB
0220 3E 50 18 06 FE 34 20 A8 3E 60 32 72 06 21 9B 01   += 06E7
0230 CD 0C 01 CD 06 01 4F CD 03 01 FE 45 CA 1E F0 FE   += 07E3
0240 41 DA D0 01 FE 45 D2 D0 01 D6 41 F5 CD 16 01 21   += 0622
0250 C3 01 CD 0C 01 F1 57 CD 06 01 4F CD 03 01 FE 4A   += 055F
0260 C2 D0 01 CD 16 01 4A CD 6D 02 C3 2D 02 3E 00 32   += 053A
0270 E2 04 79 3C FE 01 28 16 FE 02 28 12 FE 03 20 07   += 0421
0280 3E 01 32 E2 04 18 07 3E 01 32 E2 04 3E 02 4F C5   += 089A
0290 79 E6 0F F6 30 D3 34 CD B7 03 CD B7 03 DB 30 E6   += 060A
02A0 80 20 F7 3A 71 06 FE 01 CA 14 03 FE 03 CA 14 03   += 0447
02B0 21 E3 04 11 74 06 01 20 00 ED B0 C1 16 00 1E 01   += 071C
02C0 21 94 06 E5 D5 C5 EB 21 03 05 01 6E 01 ED B0 C1   += 09E8
02D0 D1 E1 E5 DD E1 D5 DD 72 10 DD 73 12 3A E2 04 DD   += 05E5
02E0 77 11 11 6E 01 19 D1 1C 7B FE 11 C2 C3 02 C5 01   += 082F
02F0 90 01 36 4E 23 0B 79 B0 C2 F2 02 C1 E5 D5 C5 CD   += 07C1
0300 9F 03 C1 D1 E1 14 3A 72 06 3C BA CA 8F 03 CD C7   += 04E0
0310 03 C3 BE 02 21 1B 04 11 94 06 01 10 00 ED B0 C1   += 0572
0320 16 00 1E 01 21 A4 06 E5 D5 C5 EB 21 2B 04 01 B7   += 0A37
0330 00 ED B0 C1 D1 E1 E5 DD E1 D5 DD 72 07 DD 73 09   += 0696
0340 3A E2 04 DD 77 08 11 B7 00 19 D1 1C 7B FE 11 C2   += 0585
0350 27 03 C5 01 2C 01 36 FF 23 0B 79 B0 C2 56 03 C1   += 08F3
0360 E5 D5 C5 CD AB 03 C1 D1 E1 14 3A 72 06 3C BA CA   += 068B
0370 8F 03 CD C7 03 3A 71 06 FE 03 C2 22 03 C5 21 E3   += 0575
0380 04 11 74 06 01 20 00 ED B0 C1 16 01 C3 BE 02 CD   += 0742
0390 B7 03 C9 C5 01 05 0D 0B 78 B1 C2 97 03 C1 C9 CD   += 0813
03A0 93 03 FB 21 74 06 CD DC 03 F3 C9 CD 93 03 FB 21   += 091C
03B0 94 06 CD FB 03 F3 C9 DB 30 CD 93 03 FB CD C2 03   += 0937
03C0 F3 C9 3E 0A D3 30 76 CD 93 03 FB CD D0 03 F3 C9   += 06D3
03D0 79 E6 0F F6 30 D3 34 3E 5A D3 30 76 3A E2 04 07   += 088C
03E0 E6 02 F6 F4 D3 30 79 E6 0F F6 30 0E 33 D3 34 DB   += 08A8
03F0 34 E6 80 CA EF 03 ED A3 C3 EF 03 3A E2 04 07 E6   += 07CA
0400 02 F6 F4 D3 30 79 E6 0F F6 20 0E 33 D3 34 DB 34

rom  abs                                                        checksum
0410 E6 80 CA 0E 04 ED A3 C3 0E 04 00 FF FF FF FF FF   += 09A2
0420 FF FF FF FF FF FF FF FF FF FF 00 00 00 00 00 00   += 0AF5
0430 00 FE 00 00 00 00 F7 FF FF FF FF FF FF FF FF FF   += 0AEC
0440 FF FF 00 00 00 00 00 00 FB E5 E5 E5 E5 E5 E5 E5   += 093C
0450 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0460 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0470 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0480 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0490 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
04A0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
04B0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
04C0 E5 E5 E5 E5 E5 E5 E5 E5 F7 FF FF FF FF FF FF FF   += 0EFE
04D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   += 0FF0
04E0 FF FF 00 4E 4E 4E 4E 4E 4E 4E 0E 4E 4E 4E 4E 4E   += 05B4
04F0 4E 4E 0E 4E 4E 4E 4E 4E 4E 4E 0E 4E 4E 4E 4E 4E   += 0460
0500 4E 4E 0E 00 00 00 00 00 00 00 00 00 00 00 00 F5   += 019F
0510 F5 F5 FE 00 00 00 01 F7 4E 4E 4E 4E 4E 4E 4E 4E   += 0650
0520 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 00 00   += 0444
0530 00 00 00 00 00 00 00 00 00 00 00 F5 F5 F5 FB E5 E5   += 05A4
0540 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0550 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0560 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0570 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0580 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0590 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
05A0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
05B0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
05C0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
05D0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
05E0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
05F0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 F7   += 0DCB
0600 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0610 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0620 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0E50
0630 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 F7   += 0DCB
0640 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E. 4E 4E   += 04E0
0650 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   += 04E0
0660 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   += 04E0
0670 4E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   += 004E
0680 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   += 0000
0690 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   += 0000
06A0 00 00 00 00 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   += 0ABC
06B0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 F7   += 0DCB
06C0 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   += 04E0
06D0 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   += 04E0
06E0 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   += 04E0
06F0 4E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   += 004E
0700
```

Bild 4. Mini-Single-Double – all das kann dieses Formatierprogramm. Assembler-Listing auf Anfrage

```
                        MACLIB DISKDEF  ;LOAD DEFINTION FOR DISKS

                ; VERSION 2.2 BIOS   FUER MINILAUFWERKE
                ; VARIANTE 1.0 FUER ITT3030 DIREKTORYKOMPATIBEL
                ;



0016 =          VERS    EQU     22
FFFF =          TRUE    EQU     0FFFFH
0000 =          FALSE   EQU     NOT TRUE
FFFF =          TEST    EQU     TRUE

EA00                    ORG 0EA00H      ;60K CP/M BIOS
0400 =          CPMB    EQU 0D400H
DC06 =          BDOS    EQU 0DC06H
1600 =          CPML    EQU $-CPMB
EA00 =          BIOS    EQU $
0000 =          BIAS    EQU 0

002C =          NSECTS  EQU CPML/128    ;128 BYTES PRO SEKTOR
0004 =          OFFSET  EQU 4   ;START BEI TRACK 4 BEI ITT 2 BEI ALPHATRONIK
                                ;                               BEI MINI DOUBLE
0004 =          CDISK   EQU 4   ;ADR 4 SP FUER LAST DRIVE
0080 =          BUFF    EQU 80H ;DEFAULT BUFFER ADRESS


                ; VEKTORTABELLE

EA00 C3B5EA             JMP BOOT
EA03 C3D7EA     WBOOTE: JMP WBOOT
EA06 C3A9EA             JMP CONST
EA09 C3ACEA             JMP CONIN
EA0C C3B2EA             JMP CONOUT
EA0F C30FF0             JMP LIST
EA12 C30CF0             JMP PUNCH
EA15 C306F0             JMP READER

EA18 C35BEB             JMP HOME
EA1B C360EB             JMP SELDSK
EA1E C376EB             JMP SETTRK
EA21 C37BEB             JMP SETSEC
EA24 C397EB             JMP SETDMA
EA27 C3CCEB             JMP READ
EA2A C325EC             JMP WRITE
EA2D C359EB             JMP LISTST
EA30 C380EB             JMP SECTRAN
                ;
008F =          DISKKAP EQU     143     ;TRUNC (16 * 256 * (40-4)/1024) - 1
                ;
                        DISKS 4         ;NUR ZWEI LAUFWERKE HIERBEI
EA33+=          DPBASE  EQU     $       ;BASE OF DISK PARAMETER BLOCKS
EA33+00000000   DPE0:   DW      XLT0,0000H      ;TRANSLATE TABLE
EA37+00000000           DW      0000H,0000H     ;SCRATCH AREA
EA3B+03ED73EA           DW      DIRBUF,DPB0     ;DIR BUFF,PARM BLOCK
EA3F+95ED83ED           DW      CSV0,ALV0       ;CHECK, ALLOC VECTORS
EA43+00000000   DPE1:   DW      XLT1,0000H      ;TRANSLATE TABLE
EA47+00000000           DW      0000H,0000H     ;SCRATCH AREA
EA4B+03ED73EA           DW      DIRBUF,DPB1     ;DIR BUFF,PARM BLOCK
EA4F+B7EDA5ED           DW      CSV1,ALV1       ;CHECK, ALLOC VECTORS
EA53+00000000   DPE2:   DW      XLT2,0000H      ;TRANSLATE TABLE
EA57+00000000           DW      0000H,0000H     ;SCRATCH AREA
EA5B+03ED73EA           DW      DIRBUF,DPB2     ;DIR BUFF,PARM BLOCK
EA5F+D9EDC7ED           DW      CSV2,ALV2       ;CHECK, ALLOC VECTORS
EA63+00000000   DPE3:   DW      XLT3,0000H      ;TRANSLATE TABLE
EA67+00000000           DW      0000H,0000H     ;SCRATCH AREA
EA6B+03ED73EA           DW      DIRBUF,DPB3     ;DIR BUFF,PARM BLOCK
EA6F+FBEDE9ED           DW      CSV3,ALV3       ;CHECK, ALLOC VECTORS
                        DISKDEF 0,0,31,0,1024,DISKKAP,64,64,OFFSET
EA73+=          DPB0    EQU     $               ;DISK PARM BLOCK
EA73+2000               DW      32              ;SEC PER TRACK
EA75+03                 DB      3               ;BLOCK SHIFT
EA76+07                 DB      7               ;BLOCK MASK
EA77+00                 DB      0               ;EXTNT MASK
EA78+8E00               DW      142             ;DISK SIZE-1
EA7A+3F00               DW      63              ;DIRECTORY MAX
EA7C+C0                 DB      192             ;ALLOC0
EA7D+00                 DB      0               ;ALLOC1
EA7E+1000               DW      16              ;CHECK SIZE
EA80+0400               DW      4               ;OFFSET
0000+=          XLT0    EQU     0               ;NO XLATE TABLE
                        DISKDEF 1,0
EA73+=          DPB1    EQU     DPB0    ;EQUIVALENT PARAMETERS
0012+=          ALS1    EQU     ALS0    ;SAME ALLOCATION VECTOR SIZE
0010+=          CSS1    EQU     CSS0    ;SAME CHECKSUM VECTOR SIZE
0000+=          XLT1    EQU     XLT0    ;SAME TRANSLATE TABLE
                        DISKDEF 2,0
EA73+=          DPB2    EQU     DPB0    ;EQUIVALENT PARAMETERS
0012+=          ALS2    EQU     ALS0    ;SAME ALLOCATION VECTOR SIZE
0010+=          CSS2    EQU     CSS0    ;SAME CHECKSUM VECTOR SIZE
0000+=          XLT2    EQU     XLT0    ;SAME TRANSLATE TABLE
                        DISKDEF 3,0
EA73+=          DPB3    EQU     DPB0    ;EQUIVALENT PARAMETERS
0012+=          ALS3    EQU     ALS0    ;SAME ALLOCATION VECTOR SIZE
0010+=          CSS3    EQU     CSS0    ;SAME CHECKSUM VECTOR SIZE
0000+=          XLT3    EQU     XLT0    ;SAME TRANSLATE TABLE

                ; ENDEF AT END OF ASSEMBLY

F01E =          MON80   EQU 0F01EH
F01E =          RMON80  EQU 0F01EH

000D =          CR      EQU 0DH
000A =          LF      EQU 0AH

                SIGNON:
EA82 1A                 DB 26
EA83 5765726636F        DB 'Welcome to 60 K  CP/M  Mini Version'
EAA6 0D0A00             DB CR,LF,0

                CONST:
EAA9 C312F0             JMP 0F012H
                CONIN:
EAAC CD03F0             CALL 0F003H
EAAF E67F               ANI 7FH
EAB1 C9                 RET

                CONOUT:
EAB2 C309F0             JMP 0F009H

F00F =          LIST    EQU 0F00FH
F00C =          PUNCH   EQU 0F00CH
F006 =          READER  EQU 0F006H

                BOOT:
EAB5 310001             LXI SP,BUFF+80H
EAB8 2182EA             LXI H,SIGNON
EABB CD9DEB             CALL PRMSG
                ; INIT SEKTORBUFFER UND MONITOR KURZSCHLUSS
                ;
EABE AF                 XRA A
EABF 32EFEC             STA WRTFLG      ;NO WRITE PERFORMED
EAC2 3EFF               MVI A,0FFH
EAC4 32F0EC             STA DRVAKT      ;DRIVE UNDEF
EAC7 21D7EA             LXI H,WBOOT
EACA 2234F0             SHLD 0F033H+1   ;MONITOR ++
EACD 2237F0             SHLD 0F036H+1   ;MONITOR ++
                ;
EAD0 AF                 XRA A
EAD1 3204D0             STA CDISK
EAD4 C317EB             JMP GOCPM


                WBOOT:
EAD7 318000             LXI SP,BUFF
EADA C5                 PUSH B
                WBOOT0:
                ; ALTER TRACK ZURUECK FALLS DIESER BESCHRIEBEN WAR
EADB 3AEFEC             LDA WRTFLG
EADE B7                 ORA A
EADF CAE5EA             JZ NOTBAC
EAE2 CDBAEC             CALL PUTTRK     ;LOESCHEN WRTFLG DORT UND ZURUECK
                                        ;BEI LAUFWERKWECHSEL AUFPASSEN AN
                                        ;STELLE, DASS NICHT VERSEHENTLICH
                                        ;WARMBOOT AUF NEUEM LAUFWERK DURCH
                                        ;WIRD DA SONST DATEN VERNICHTET WE
                                        ;
EAE5 3EFF       NOTBAC: MVI A,0FFH
EAE7 32F0EC             STA DRVAKT      ;DAMIT BUFFER UNDEFINIERT SONST
                                        ;ALTER  TRACK BEI DISKWECHSEL ALS
                                        ;BUFFER VERWENDET UND DAMIT FEHLER
EAEA 2100D4             LXI H,CPMB      ;START ADRESSE
EAED 0616               MVI B,NSECTS/2  ;ANZAHL DER SEKTOREN  A 256 BYTES
EAEF 1601               MVI D,1         ;START BEI TRACK 1 WEGEN DOUBLE DE
EAF1 1E01               MVI E,1         ;1..16 SEKTOR
                RDSEC:
EAF3 E5                 PUSH H
```

Bild 8. Das Assemblerlisting des BIOS

```
EBD9 BA             CMP D
EBDA C200EC         JNZ RLOAD
EBDD 3AF2EC         LDA SEKAKT
EBE0 BB             CMP E
EBE1 C200EC         JNZ RLOAD
                R1RD:
                ; OK IST SCHON IN BUFFER
EBE4 2100FC         LXI H,BUFFER    ;ADRESSE BERECHNEN
EBE7 3AF7EC         LDA IOS         ;0..32   * 128 + BUFFER
EBEA E607           ANI 00000111B   ;0,1,2,3,4,5,6,7
EBEC 57             MOV D,A
EBED 1E00           MVI E,0         ;SCHIEBEN MIT Z80 BEFS
EBEF CB2A           DB 0CBH,2AH     ;SRA D
EBF1 CB1B           DB 0CBH,1BH     ;RR E   = *256/2
EBF3 19             DAD D           ;+BUFFER
EBF4 EB             XCHG            ;NACH DE IST ZIEL
EBF5 2AF8EC         LHLD IOD        ;DMA ADRESSE QUELLE HIER
EBF8 EB             XCHG            ;ZIEL DE
EBF9 018000         LXI B,128       ;LAENGE
EBFC EDB0           DB 0EDH,0B0H    ;LDIR
EBFE AF             XRA A
EBFF C9             RET             ;OK ENDE


                RLOAD:              ;NEUEN LADEN, GGF ALTEN ZURUECK
EC00 3AEFEC         LDA WRTFLG
EC03 B7             ORA A
EC04 CA0DEC         JZ R1LOAD
EC07 CDBAEC         CALL PUTTRK     ;ALTEN ZURUECKSCHREIBEN
EC0A DA82EC         JC ERRIO
EC0D CDAAEB     R1LOAD: CALL CALC   ;BERECHNEN
EC10 79             MOV A,C
EC11 32FDEC         STA DRVAKT
EC14 7B             MOV A,E
EC15 32F2EC         STA SEKAKT
EC18 7A             MOV A,D
EC19 32F1EC         STA TRKAKT
EC1C CD86EC         CALL GETTRK
EC1F DA82EC         JC ERRIO
EC22 C3E4EB         JMP R1RD
                ,
                WRITE:
EC25 CDAAEB         CALL CALC       ;FUER VERGLEICH
EC28 3AFDEC         LDA DRVAKT
EC2B B9             CMP C           ,
EC2C C25DEC         JNZ WLOAD
EC2F 3AF1EC         LDA TRKAKT
EC32 BA             CMP D
EC33 C25DEC         JNZ WLOAD
EC36 3AF2EC         LDA SEKAKT
EC39 BB             CMP E
EC3A C25DEC         JNZ WLOAD
                W1WR:
                , OK IST SCHON IN BUFFER
EC3D 2100FC         LXI H,BUFFER    ;ADRESSE BERECHNEN
EC40 3AF7EC         LDA IOS         ;0..32   * 128 + BUFFER
EC43 E607           ANI 00000111B   ;0,1,2,3,4,5,6,7
EC45 57             MOV D,A
EC46 1E00           MVI E,0         ;SCHIEBEN MIT Z80 BEFS
EC48 CB2A           DB 0CBH,2AH     ;SRA D
EC4A CB1B           DB 0CBH,1BH     ;RR E   = *256/2
EC4C 19             DAD D           ;+BUFFER
EC4D EB             XCHG            ;NACH DE IST ZIEL
EC4E 2AF8EC         LHLD IOD        ;DMA ADRESSE QUELLE HIER
EC51 018000         LXI B,128       ;LAENGE ZIEL DE
EC54 EDB0           DB 0EDH,0B0H    ;LDIR
EC56 3E01           MVI A,1
EC58 32EFEC         STA WRTFLG      ;NUN BESCHRIEBEN
EC5B AF             XRA A
EC5C C9             RET             ;OK ENDE


                WLOAD:              ;NEUEN LADEN, GGF ALTEN ZURUECK
EC5D 3AEFEC         LDA WRTFLG
EC60 B7             ORA A
EC61 CA6AEC         JZ W1LOAD
EC64 CDBAEC         CALL PUTTRK     ;ALTEN ZURUECKSCHREIBEN
EC67 DA82EC         JC ERRIO
EC6A CDAAEB     W1LOAD: CALL CALC   ;BERECHNEN
EC6D 79             MOV A,C
EC6E 32FDEC         STA DRVAKT
EC71 7B             MOV A,E
EC72 32F2EC         STA SEKAKT
EC75 7A             MOV A,D
EC76 32F1EC         STA TRKAKT
EC79 CD86EC         CALL GETTRK
EC7C DA82EC         JC ERRIO
EC7F C33DEC         JMP W1WR

EC82 3E01       ERRIO.  MVI A,1
EC84 B7             ORA A
EC85 C9             RET



                ; BUFFERVERWALTUNG

                GETTRK:             ;TRKAKT,SEKAKT,DRVAKT ENTHALTEN NEUE
                                    ;BUFFERADRESSE
EC86 AF             XRA A
EC87 32EFEC         STA WRTFLG
EC8A 2100FC         LXI H,BUFFER
EC8D 3AF2EC         LDA SEKAKT
EC90 5F             MOV E,A
EC91 3AFDEC         LDA DRVAKT
EC94 F6D0           ORI 11010000B   ;LAUFWERK PHYS 0,1,2,3 DOUBLE DENSE
EC96 4F             MOV C,A
EC97 0601           MVI B,1         ;READ
EC99 1604           MVI D,4         ;VIER MAL
                GETLP1:
EC9B E5             PUSH H
EC9C D5             PUSH D
EC9D C5             PUSH B
EC9E 3AF1EC         LDA TRKAKT
ECA1 57             MOV D,A
ECA2 CDECEC         CALL EXEC
ECA5 C1             POP B
ECA6 D1             POP D
ECA7 E1             POP H
ECA8 DAB8EC         JC ERRX
ECAB D5             PUSH D
ECAC 110001         LXI D,256
ECAF 19             DAD D
ECB0 D1             POP D
ECB1 1C             INR E           ;NEXT SEKTOR
ECB2 15             DCR D
ECB3 C29BEC         JNZ GETLP1      ;ALLE VIER SEKTOREN
ECB6 AF             XRA A
ECB7 C9             RET
                ;
ECB8 37         ERRX:   STC
ECB9 C9             RET



                PUTTRK.             ;TRKAKT,SEKAKT,DRVAKT ENTHALTEN NEUE
                                    ;BUFFERADRESSE
ECBA AF             XRA A           ;WIRD ZURUECKGESCHRIEBEN
ECBB 32EFEC         STA WRTFLG
ECBE 2100FC         LXI H,BUFFER
ECC1 3AF2EC         LDA SEKAKT
ECC4 5F             MOV E,A
ECC5 3AFDEC         LDA DRVAKT
ECC8 F6D0           ORI 11010000B   ,LAUFWERK PHYS 0,1,2,3 DOUBLE DENSE
ECCA 4F             MOV C,A
ECCB 0602           MVI B,2         ,WRITE
ECCD 1604           MVI D,4         ;VIER MAL
                PUTLP1:
ECCF E5             PUSH H
ECD0 D5             PUSH D
ECD1 C5             PUSH B
ECD2 3AF1EC         LDA TRKAKT
ECD5 57             MOV D,A
ECD6 CDECEC         CALL EXEC
ECD9 C1             POP B
ECDA D1             POP D
ECDB E1             POP H
ECDC DAB8EC         JC ERRX
ECDF D5             PUSH D
ECE0 110001         LXI D,256
ECE3 19             DAD D
ECE4 D1             POP D
ECE5 1C             INR E           ;NEXT SEKTOR
ECE6 15             DCR D
ECE7 C2CFEC         JNZ PUTLP1      ;ALLE VIER SEKTOREN
```

Rolf-Dieter Klein:

Das mc-Grafik-Terminal

Bisher war es nur wenigen möglich, von hochauflösender Grafik Gebrauch zu machen: Entweder ist sie umständlich zu bedienen oder zu teuer. Mit dem mc-Grafik-Terminal soll das anders werden. Das Terminal besteht aus einer Europakarte, die neben einem Z80-Prozessor auch einen Grafikprozessor beinhaltet.

Die Software zur Bedienung des Terminals ist knapp 8 KByte groß und ermöglicht neben einer Teilemulation des TVI-950 (wordstarfähige Teilmenge) die Emulation des T4014 (Tektronix-kompatibel) und einen eigenen Grafik-Modus mit vielen einfach zu bedienenden Befehlen der mc-Grafik-Sprache. Die Auflösung des Grafikbildschirms ist 512 × 256 Punkte, die bei Auswahl eines anderen Grafikprozessors auch 512 × 512 Punkte sein kann. Bei 512 × 256 Punkten stehen vier Bildschirmebenen zur Verfügung, die von der Software zum Scrollen gebraucht werden oder im Grafikmodus für bewegte Grafik verwendet werden können.

Bild 1 zeigt die Anschlüsse zur Außenwelt. Das Terminal besitzt zwei serielle Schnittstellen, die eine ist gedacht zum Anschluß des Computers, die andere für den Anschluß einer Tastatur. Beide sind bidirektional, da die Tastatur auch programmiert werden soll. Am Ausgang steht ein BAS-Signal zur Verfügung, das direkt an einen Standard-Monitor angeschlossen werden kann. Ferner sind die Ausgänge HSYNC, VSYNC und Video für Spezialanwendungen und für die Farberweiterung vorhanden.

Zwei Prozessoren

Bild 2 zeigt das Blockschaltbild der Platine. Ein Z80A übernimmt die Steuerung des Terminals. 8 KByte EPROM sind für die Steuersoftware vorhanden, die im nächsten Heft vorgestellt wird. 4 KByte RAM werden für einen internen Bildwiederholspeicher und einen Ringpuffer für die Zeichenebene gebraucht, wobei etwa ½ KByte frei bleibt für Anwendersoftware, die auch ladbar ist. Mit den neuen 8 K × 8-Bit-RAMs, die leider noch sehr teuer sind, kann der Speicher auf 16 KBytes erweitert werden, und dann läßt sich dort auch ein programmierbarer Zeichengenerator unterbringen. Doch vorerst genügen die 4 KByte. Zwei STI-Bausteine (MK 3801), die eine serielle Schnittstelle, Timer, Interruptcontroller und Parallelports beinhalten, ermöglichen die Steuerung und Kommunikation. Als Grafikprozessor wurde der EF-9366 der Firma Thomson gewählt, da er der einzige ist, mit dem sich auf einer Europakarte ein komplettes Grafikterminal unterbringen läßt und der außerdem den schnellsten Vektorgenerator (1 Mio. Bildpunkte/s) besitzt. Dem Grafikprozessor steht ein vom Hauptspeicher getrennter Bildwiederholspeicher von 64 KByte zur Verfügung. Beim EF-9366 ist er in vier Bildebenen von je 512 × 256 Punkte aufgeteilt. Es wird immer nur eine Bildebene ausgelesen, aber in den anderen kann gleichzeitig geschrieben werden, wodurch ein unsichtbarer Bildaufbau möglich ist. Die Auswahl der Bildebenen erfolgt über Steuerbits der STI-A. Die STI-B kann auch entfallen, wenn keine Tastatur benötigt wird und das Terminal als Peripheriegerät betrieben wird: Die Software merkt dies automatisch und die Baudrate ist dann fest auf 9600 Baud eingestellt. Der EF-9365 erlaubt die Darstellung von 512 × 512 Punkten, dieser

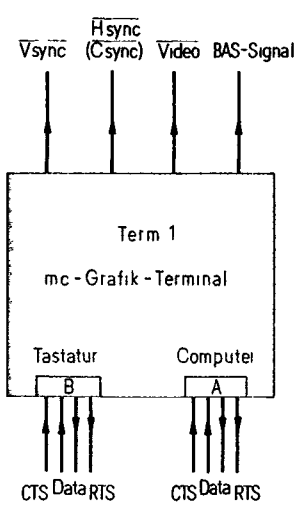


Bild 1. Diese Signale verbinden die Grafik-Terminal-Platine mit der Außenwelt

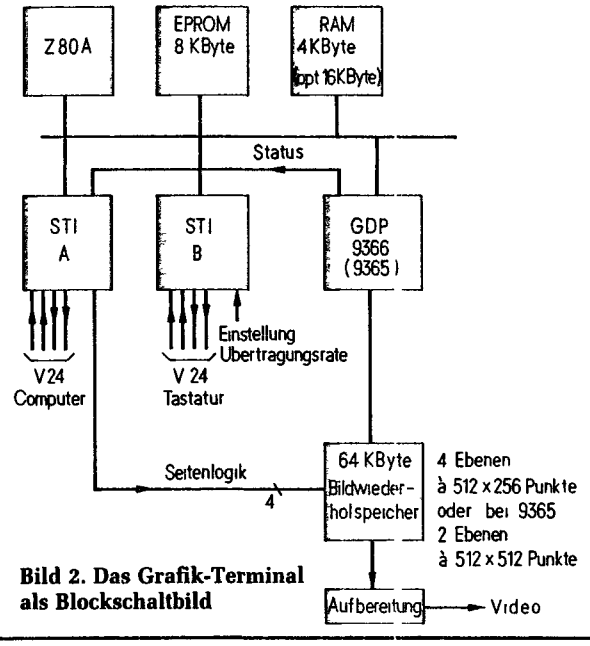


Bild 2. Das Grafik-Terminal als Blockschaltbild

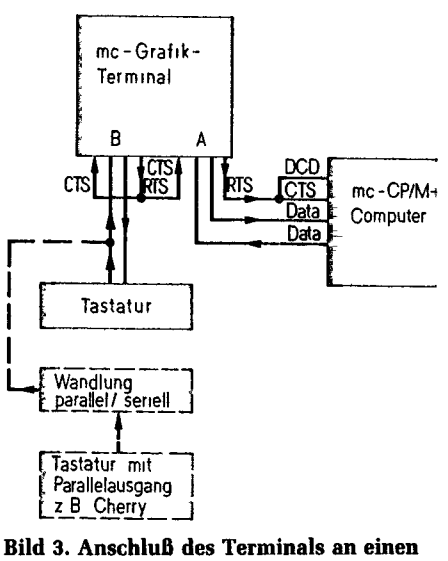


Bild 3. Anschluß des Terminals an einen Computer (beispielsweise der mc-CP/M-Computer) und der Anschluß der Tastatur

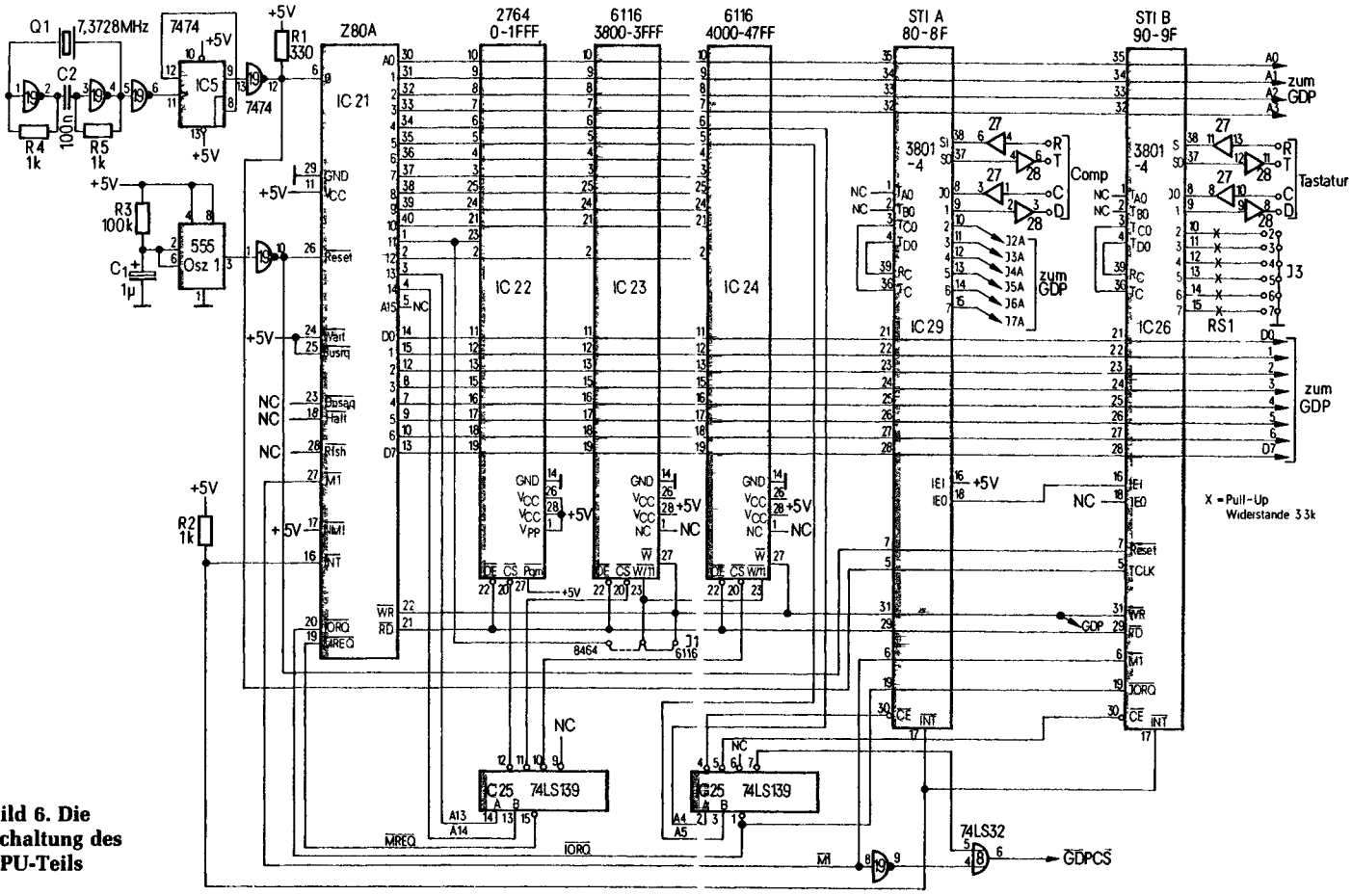


Bild 6. Die Schaltung des CPU-Teils

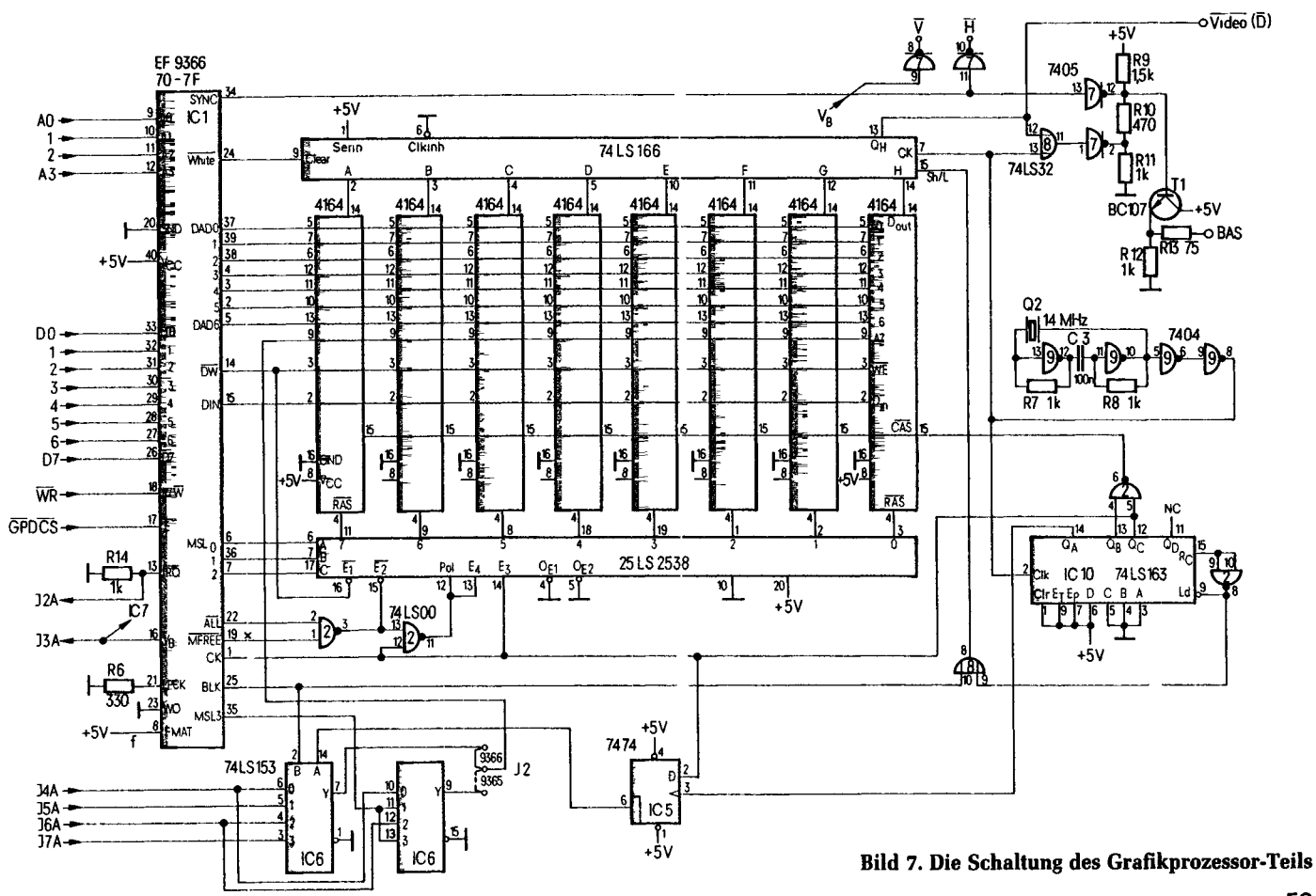


Bild 7. Die Schaltung des Grafikprozessor-Teils

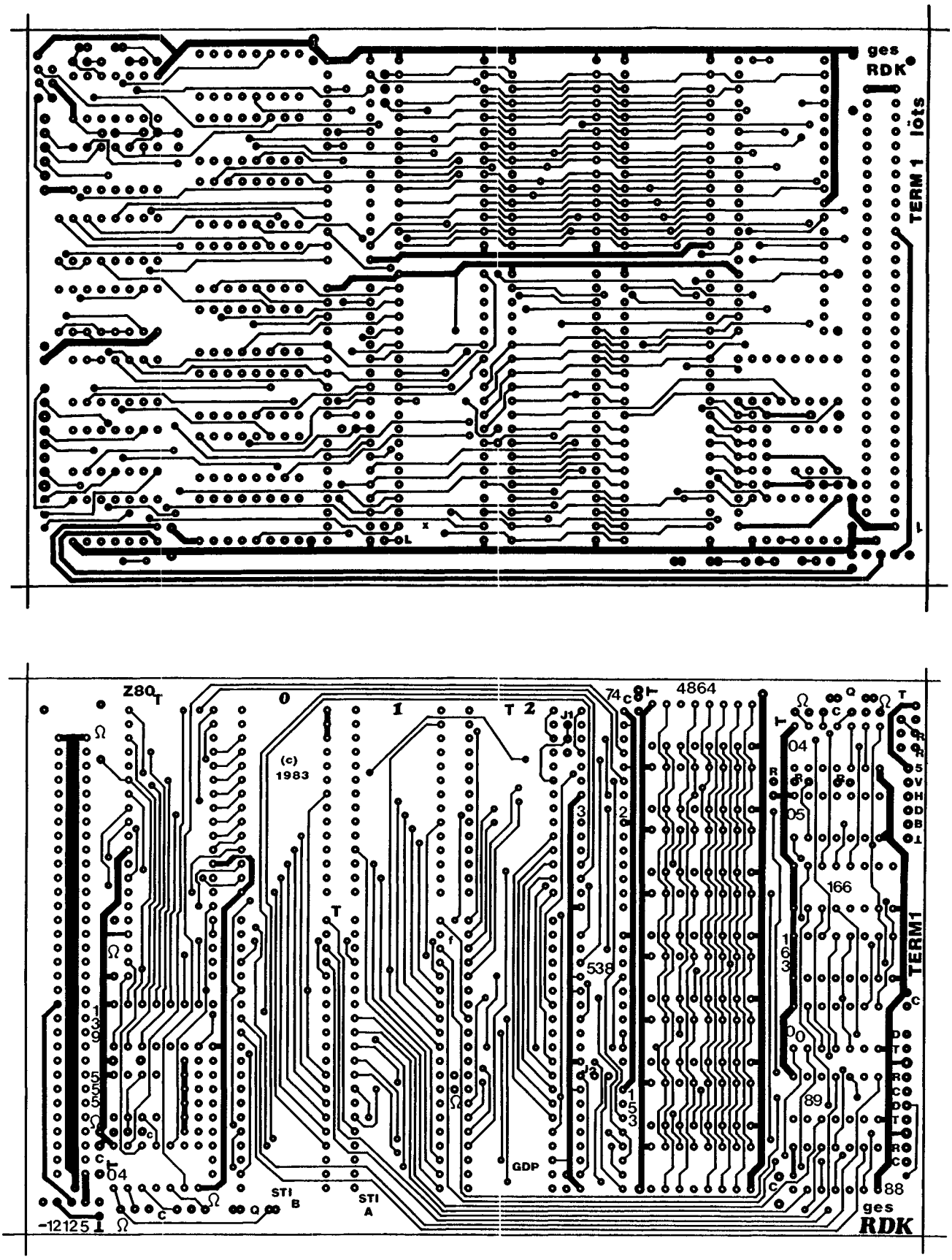


Bild 9. Die Schaltung befindet sich auf einer zweiseitigen Leiterplatte, die Lötseite oben und die Bestückungsseite unten

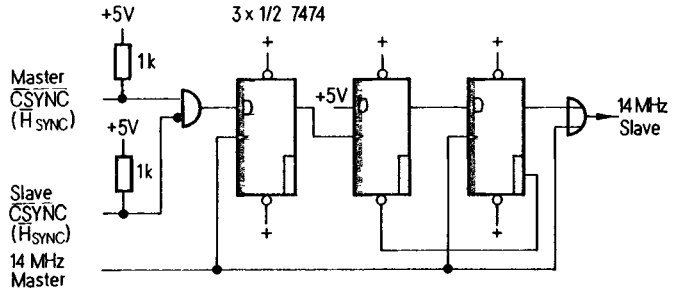


Bild 11. Eine Schaltung zur Synchronisation mehrerer Einheiten

gleich verwendet, da es HSYNC- und VSYNC-Informationen enthält. Das Signal ist in der Schaltung auch als HSYNC bezeichnet. Alle Slave-Platinen werden ohne Taktoszillator (14 MHz) aufgebaut, wobei es genügt, den 7404 (IC 9) herauszulassen. An Pin 8 des nun nicht mehr vorhandenen ICs wird dann ein neuer Takt angeschlossen, der von der Zusatzschaltung kommt. Stimmt CSYNC des Masters nicht mit CSYNC des Slave überein, so wird ein Taktpuls beim Slave ausgelassen, dadurch verschiebt sich das Timing. Nach einer Weile (einige Sekunden) sind beide CSYNC-Signale identisch und der Takt ist von da an gleich.

Dieses Verfahren hat sich gut bewährt und ist einfach zu realisieren. Beim Nachbau ist darauf zu achten, daß 14 MHz schon sehr kritisch sind und kurze Leitungen wie auch gute Entblokkung Voraussetzung für das Funktionieren der Schaltung sind. Es empfiehlt sich auch, den Mastertakt nochmals getrennt auf der Zusatzplatine zu erzeugen und dem eigentlichen Master über ein Oder-Gatter zuzuführen, so daß alle Takte der Platinen genau synchron anfangen. Das ist aber nicht Voraussetzung der Schaltung, da Master und Slave nur auf einen Taktzyklus genau synchronisiert werden. Sind die Verzögerungszeiten aber gleich, so sind die beiden Teilbilder absolut deckungsgleich.

Am Schluß sei noch darauf hingewiesen, daß die Originalplatine (TERM 1) sowie das EPROM nur bei der Firma GES/Graf (Postfach 1610, 8960 Kempten) zu erhalten sind. Alle Original-Platinen werden mit Lötstopplack und alle Bausätze mit hochwertigen IC-Sockeln geliefert. *(Fortsetzung folgt)*

Literatur

[1] *Klein, Rolf-Dieter:* CRT-Controller unterstützt Grafikfunktionen. ELEKTRONIK 1981, Heft 8, Seite 63.

[2] *Klein, Rolf-Dieter:* Umwandlung. In diesem Heft

EF 9366 und dem neueren EF 9367. Wir wollen standardmäßig den EF 9366 verwenden, der auch der preisgünstigste ist. Mit ihm erhält man eine Auflösung von 512 x 256 Pixel in je 4 Bildebenen. *Bild 2* zeigt das Arbeitsergebnis des EF 9365, *Bild 3* das des EF 9366 und *Bild 4* das für den EF 9367. Der EF 9367 erzeugt ein Rechteckfenster, während die anderen Prozessoren quadratische Bildfenster besitzen. Beim EF 9367 tritt ein Vektor unter Umständen gespiegelt wieder ins Bild, wenn die Endpunkte außerhalb liegen, da in unserer Schaltung der X9-Ausgang nicht ausgewertet wird.

Die Bilddarstellung des EF 9366

Wir arbeiten also mit dem EF 9366 und erhalten eine Aufteilung nach *Bild 5*. Es zeigt vier Bildseiten mit je 512 x 256 Bildpunkten. Es kann immer nur eine Seite auf dem Bildschirm dargestellt werden, daher gibt es einen Lesezeiger. Geschrieben werden kann aber auch in eine unsichtbare Seite. Dafür ist der Schreibzeiger gedacht. Wie wir später noch genauer sehen, gibt es die Möglichkeit der quasisimultanen Darstellung zweier Seiten, bei der die Seiten synchron (durch einen Grafik-Befehl veranlaßt) mit der Bildwechselfrequenz geschaltet werden. Es ergibt sich dann ein leicht flimmerndes Bild, ähnlich wie beim Zeilensprungverfahren. Dafür sind aber zwei Seiten „gleichzeitig" sichtbar und es läßt sich dann mit Grauabstufung arbeiten (wenn auf zwei Seiten das gleiche Bild ist, wirkt es heller als wenn nur eine Seite das Bild zeigt). Damit können auch Fadenkreuze und vieles mehr dargestellt werden.

Die Textdarstellung

Nach dem Einschalten der Term-Platine erscheint an der linken oberen Ecke im Bild ein blinkender Cursor. Die Platine ist nun bereit zur Texteingabe. Bevor wir auf die Befehle eingehen, zunächst etwas über die interne Darstellung von Texten. Wer schon einmal mit dem Grafik-Prozessor gearbeitet hat, wird wissen, daß er keine Befehle für die Verschiebung des Textfensters (Scrolling) besitzt. Daher muß das Scrolling, was auch für das Einfügen und Löschen von Zeilen gilt, per Software durch Neueinschreiben des gesamten Bildes erfolgen. Die Besonderheit hierbei ist, daß jeder Punkt einzeln neu erzeugt werden muß. Mit einem konventionellen Verfahren ohne Grafik-Prozessor wäre das recht aussichtslos, denn dazu wäre eine beträchtliche Zeit notwendig. Hier ist der GDP von Thomson jedoch so schnell, daß ein Wiedereinschreiben eines Bildes im wesentlichen nur davon abhängt, wie schnell die einzelnen Zeichen angeliefert werden. Um das Löschen des alten Bildes und das Wiedereinschreiben ohne Flimmern zu realisieren, wurden zwei Bildebenen verwendet. Eine Bildebene ist sichtbar, während die andere für das neue Bild vorbereitet wird, dann wird umgeschaltet und die neue Ebene dargestellt. *Bild 6* zeigt die Situation. Der Cursor entsteht dadurch, daß er auf der einen Bildebene geschrieben wird, auf der anderen nicht. Die Seiten werden nach dem Bildaufbau mit einer konstanten Frequenz hin- und hergeschaltet.

Um die Textausgabe optimal zu gestalten, ist ein zusätzlicher Bildwiederholspeicher nötigt, denn es ist nicht möglich, die Bildinformation, oder gar die Zeichencodes aus dem Bildspeicher wieder rückzulesen. *Bild 7* zeigt die Speicherorganisation im CPU-Speicher. Ein Speicher mit 80 x 24 Zeichen beinhaltet die Zeicheninformationen. Daneben gibt es einen Speicher der Tiefe 24, in dem in einem Byte die jeweilige Anzahl der tatsächlich in dieser Zeile vorhandenen Zeichen abgelegt ist. Damit das Scrolling, Löschen und Einfügen von Zeilen möglichst schnell geht, gibt es einen weiteren Speicherraum von 24 Zeigern, die auf den Bildwiederholspeicher gerichtet sind. Eine Zeile wird gelöscht, indem die Zeigerkette verändert wird. Nach Veränderung wird das Bild auf einer Seite gelöscht und neu aufgebaut. Das Löschen geschieht durch Einschreiben des alten Inhalts im Löschmodus um Zeit zu sparen, und außerdem ist es dadurch möglich, Texte und Grafik zu mischen und beim Scrolling die Grafik nicht zu zerstören, solange sie beim Scrollvorgang nicht von Zeichen durchkreuzt wird.

Die Befehle der Grafik-Karte

Nun zu den einzelnen Befehlen. *Bild 8* zeigt eine Tabelle aller vorhandenen Befehle. Der Alpha-Modus ist der Grundmodus, der auch nach Reset vorliegt. Die Befehle wurden so gewählt, daß Wordstar mit der TVI-950-Installation direkt läuft. Es sind natürlich nicht alle TVI-950-Befehle realisiert, so fehlen zum Beispiel Invers, Underline und Blinkend. Alle diese Eigenschaften können aber über Grafik-Befehle viel effektiver dargestellt werden (und außerdem wäre dazu ein weiterer Attributspeicher notwendig gewesen). Hinzugefügt wurde die Umschaltung auf den deutschen Zeichensatz mit ESC z 1. Dann werden die Zeichen „ÄÖÜäöüß" geplottet. Die Scrollgeschwindigkeit verringert sich dadurch leider, und zwar um so mehr, je mehr dieser Zeichen auf dem Bildschirm sind. Im Mittel scrollt das Terminal jedoch so schnell, daß es nicht möglich ist, die Zeichen zu lesen.

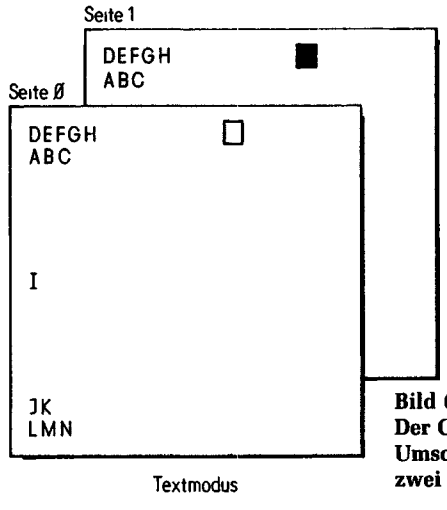


Bild 6. Der Textmodus. Der Cursor wird durch Umschalten zwischen zwei Seiten betrieben

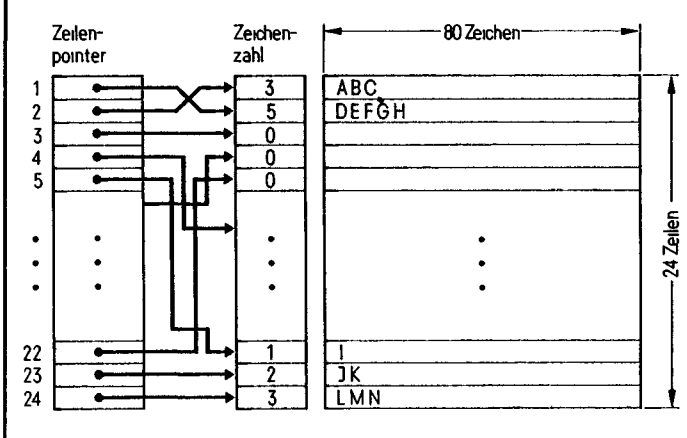


Bild 7. Die internen Bildwiederholspeicher werden über Zeiger verwaltet

n = 3*4 + 1 = 13. Befehl:
P 13 CR

S n CR — Seite synchron anwählen.
n wird wird wie oben bestimmt

X n CR — Die Seiten 0,1,2,3 werden zyklisch angezeigt, n gibt die Sichtdauer einer Seite an.
Die Anzeigedauer beträgt
n * 20ms
n = 0 beendet den Wechsel

Y n CR — Es werden jeweils nur zwei Seiten zyklisch angezeigt und zwar die Seiten 0 und 1, wenn eine dieser beiden Seiten als Leseseite definiert war, sonst 2 und 3

C — Löschen der aktuellen Schreibseite

Z — Löschen aller Seiten 0 bis 3.
Seite 3 ist danach als Lese- und Schreibseite angewählt

G n1 n2 CR — Befehl an GDP
n1 = Nummer des GDP-Ports (0..15),
n2 = Datenwert an diesen Port

B text CR — Text (20h..7fh) an GDP-Port 0 senden.
Der Text wird wird ab aktueller Koordinate des GDP ausgegeben

V binär 00h — Binärdaten an den GDP-Port 0 senden
Das ASCII-Zeichen NUL beendet die Obertragung

O n1 n2 n3 n4 CR — Ellipsenabschnitte zeichnen
Mit n1 wird die Länge der Halbachse in x-Richtung angegeben, mit n2 die Länge der Halbachse in y-Richtung.
Mit n3 wird der Startwinkel bezüglich der x-Achse in Grad angegeben. Mit n4 der Endwinkel des Ellipsenabschnittes.
Der Ellipsenabschnitt wird von der aktuellen x,y-Koordinate bis zum Erreichen des Endwinkels gezeichnet. Der Ellipsenmittelpunkt wird aus der Startwinkel- und Halbachsenangabe vor Beginn des Zeichnens automatisch errechnet

O n1 n2 n3 n4 1 CR — Wie oben oben, jedoch der vom Kurvenstück und den Radien zum Mittelpunkt begrenzte Raum gefüllt (Torte)

R n1 n2 CR — Rechteck ab aktueller x,y-Koordinate.
n1=dx und n2=dy geben die Breite und die Höhe des Rechtecks an

L n1 n2 n3 n4 ... nn nm CR — Polygon zeichnen, mit absoluten Koordinaten.
x0=n1 y0=n2 gibt die Startposition an, alle weiteren Paare geben die Eckpunkte des Polygons an.
Der letzte Eckpunkt wird wieder mit dem Startpunkt verbunden

L n1 n2 n3 n4 n5 n6 CR — Dreieck gefüllt zeichnen
x0 = n1 y0 = n2
x1 = n3 y1 = n4
x2 = n5 y2 = n6

F n1 n2 n3 CR — Fadenkreuz zeichnen,
an Position x=n1 y=n2,
auf Seite n3 (0..3).
Altes Fadenkreuz wird gelöscht
Die Schreib- und Leseseite bleiben erhalten

WA string CR — Symbol für den Fadenkreuz-Befehl umdefinieren.
string ist eine Zeichenkette mit Zeichen im ASCII-Bereich (30h,31h,40h..5fh).
0 (30h) = Schreibstift hoch,
1 (31h) = Schreibstift runter.
Der Code für die Schreibstiftbewegung berechnet sich wie folgt:
Richtung + (8 * Länge) + 40h
Es stehen die Richtungen von 0 bis 7 (* 45 Grad) zur Verfügung

WA CR — Symbol für den Fadenkreuzbefehl auf das Symbol "Fadenkreuz" zurücksetzen

WB — Fadenkreuzsymbol an aktueller x,y-Koordinate auf der aktuellen Schreibseite setzen

WC n1 n2 CR — Fadenkreuzsymbol vergrößern und drehen
n1 = Vergrößerungsfaktor (1..255)
n2 = Drehung (0..7)

WC CR — Rückstellen auf n1 = 1,
n2 = 0

WD n1 n2 n3 — nn CR
Download. Es können die Daten n2 bis nn ab Adresse n1 in den Arbeitsspeicher der Term1 geladen werden

WE n1 CR — Programm auf Adresse = n1 starten

WF n1 CR — Byte mit Adresse n1 aus dem Speicher der Term1 lesen
Ergebnis wird binär über den Port der STI-A übertragen

Doppelescape

1bh 1bh 41h	ESC ESC A	Alpha-Modus (oder Logo,T4010)
1bh 1bh 43h	ESC ESC C	Durchgangsmodus
1bh 1bh 50h	ESC ESC P	Parallel-Modus
1bh 1bh 52h	ESC ESC R	Rücksetzen

LOGO-MODUS

A — Alpha-Modus

Z — Lösche Bildschirm, alte Koordinaten bleiben

M n1 n2 n3 CR — Turtle Positionieren für Initialsierung.
x=n1 y=n2 Startwinkel=n3 , Angabe in Grad (0..359)

F n1 CR — Vorwärts (bei negativer Zahl Rückwärts).
n1 ist die Schrittzahl

P n1 CR — Drehung der Schildkröte von dem aktuellen Winkel aus um den Winkel n1 (in Grad, + und - möglich).

D — Pen Down

U — Pen Up

T0 — Turtle unsichtbar

T1 — Turtle sichtbar

1bh 1bh 41h	ESC ESC A	Alpha-Modus
1bh 1bh 47h	ESC ESC G	Grafik-Modus aufrufen
1bh 1bh 43h	ESC ESC C	Durchgangsmodus
1bh 1bh 50h	ESC ESC P	Parallel-Modus
1bh 1bh 52h	ESC ESC R	Rücksetzen

TEKTRONIX-4010-MODUS

Es sind implementiert.
Alpha-, Grafik-, Incremental-, Pointplot-Modus

Nicht implementiert:
Gin-Modus nicht implementiert
(Definitionen siehe Tektronix 4010-Manual)

T4010-Alpha-Modus·

20h bis 7fh — Textzeichen im Tektronix-Alpha-Modus

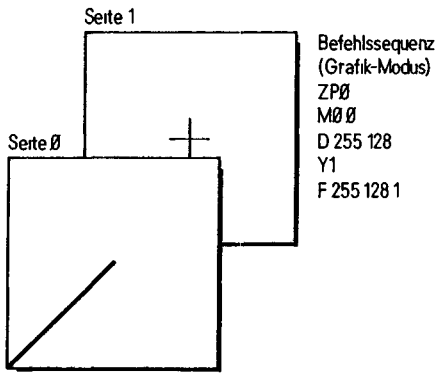


Bild 9. Der Fadenkreuzmodus

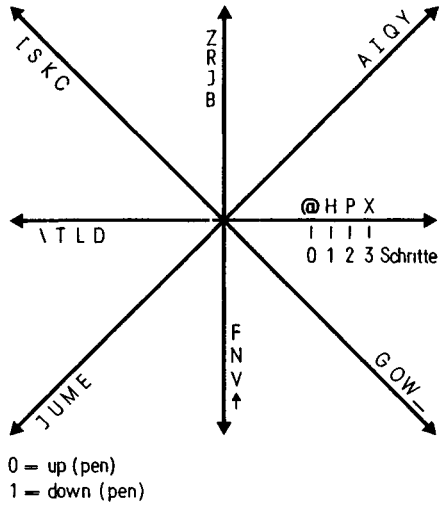
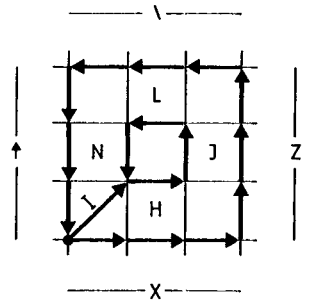


Bild 10. Die Codierung von Figuren



" X Z \ ↑ Ø I 1 H J L N "

Bild 11. Ein Beispiel zu einer Figurcodierung

```
 '10 CLEAR 2000
20 REM Beispiele fuer TERM 1
30 REM Rolf-Dieter Klein 830508
40 REM TERMINAL-TVI-950 Funktionen
50 PRINT CHR$(26); 'Clear Screen
60 FOR X=0 TO 79 'Alle Spalten
70 Y = SIN(X/79*2*PI)*11+11
80 Y = INT(Y)
90 PRINT CHR$(27);"=";CHR$(32+Y);CHR$(32+X);"*"
100 NEXT X
110 REM Wegschieben durch Scrollen
120 FORI=1TO40:PRINT:NEXT I
130 REM
140 REM verschiedene Zeichensaetze
150 PRINT CHR$(27);"=";CHR$(32+5);CHR$(32)
160 PRINT CHR$(27);"z0" 'amerikan Zeichen
170 PRINT
180 FORI=32 TO 127:PRINT CHR$(I);:NEXT I
190 PRINT
200 PRINT CHR$(27);"z1" 'deutsche Zeichen
210 FORI=32 TO 127:PRINT CHR$(I);:NEXT I
220 PRINT
230 FORI=1TO1000:NEXT I
240 REM Einfuegen, Loeschen etc.
250 FOR J=1 TO 2
260 PRINT CHR$(26);CHR$(27);"=$ "
270 PRINT "Testsatz fuer Einfuegen etc."
280 PRINT CHR$(11); ' cursor up
290 FORI=1TO10:PRINT CHR$(27);"E";:NEXT I
300 FORI=1TO10:PRINT CHR$(27);"R";:NEXT I
310 FORI=1TO10:PRINT CHR$(27);"Q";:NEXT I
320 FORI=1TO10:PRINT CHR$(27);"W";:NEXT I
330 PRINT 'cr ausgeben, wegen autocr von Basic
340 NEXT J
```

Bild 12. Ein Programmbeispiel für den Textmodus

```
10 REM Graphic Modus
20 REM Rolf-Dieter Klein 830508
30 PRINT CHR$(27);CHR$(27);"GZPO" 'Graphic Modus
40 PRINT "MO 0;R511 255" 'Rechteck-Rahmen zeichnen
50 PRINT "M50 200;G3 $22;BGraphic-Befehle"
60 PRINT "M200 50;0100 50 0 0 1"
70 PRINT "l300 0 511 20 400 200"
80 REM Symbol definieren
90 PRINT "WA";"HHHHIJKKLLMMNO"
100 PRINT "Y1" 'zwei Seiten quasisimultan
110 PRINT "WC2 0" 'VERGROESSERUNG 2 DREHUNG 0
120 FOR X=0 TO 511
130 PRINT "F";X;INT(SIN(X/100*2*PI)*120+127);1
140 NEXT X
150 PRINT "A" 'in Alpha zurueck
999 PRINT "A"
```

Bild 13. Ein Programmbeispiel für den Grafik-Modus

```
10 REM Statuszeile ueber Graphic Modus
20 REM Rolf-Dieter Klein 830508
30 PRINT CHR$(27);CHR$(27);"G" 'ohne Loeschen
40 PRINT "PO;MO 0;R511 15" ' mit Rahmen
50 PRINT "P5;MO 0;R511 15" ' beide Seiten
60 PRINT "PO;G3 $22;M40 -1;BStatus Zeile doppelt hoch"
70 PRINT "P5;M40 -1;BStatus Zeile hoch"
80 PRINT "A" 'wieder zurueck in Textmodus
90 LIST
```

Bild 14. Einblendung der Statuszeile

```
10 REM Kleine LOGO-Demonstration
20 REM Rolf-Dieter Klein
30 PRINT CHR$(27);CHR$(27);"L" 'std mode
40 PRINT "M255 255 90" 'umgerechnet auf 512x256
50 FOR I=0 TO 360/10
60 GOSUB 130 'Muster 1
70 PRINT "P-10"
80 NEXT I
90 PRINT "TO" 'Turtle off
100 FOR I=1 TO 1000:NEXT I
110 PRINT "A"
120 END
130 REM Unterprogramm Figur
140 FOR N=1 TO 4
150 PRINT "F150;P";360/4
160 NEXT N
170 RETURN
```

Bild 15. Logo-Demo

```
10 REM Tektronix Demo
20 REM Rolf-Dieter Klein 830508
30 PRINT CHR$(27);CHR$(27);"TO 0 1 2 1"
40 FOR X=0 TO 1023 STEP 4
50 Y=(SIN(X/1023*2*PI)*500+512)
55 Y=INT(Y)
60 GOSUB 100 'Tektronix  Ausgabe
70 NEXT X
80 FORI=1TO1000:NEXT I
90 PRINT CHR$(27);CHR$(27);"A":END
100 REM x,y trennen und Ausgeben
105 PRINT CHR$(28);CHR$(7); 'Pointplot mode
110 PRINT CHR$(INT(Y/32)+32);CHR$(Y MOD 32 + 96);
999 RETURN
CHR$(INT(X/32)+32);CHR$(X MOD 32 + 64)
```

Bild 16. Der Tektronix-Modus

```
; initialisieren der Peripheriebausteine

;
siolist::

defb 2*18                ;Anzahl der Eintraege
                         ;STI-A
defb stia+12,10001000b   ;/16 1 Stop 8 Bit
defb stia+13,00000001b   ;receive enable
defb stia+14,00000101b   ;transmitter enable
defb stia+8,1            ;pointer to timer d
defb stia+0,3            ;/3
defb stia+8,2            ;pointer to timer c
defb stia+0,3            ;/3  9600 baud
defb stia+8,7            ;pointer to control c,d
defb stia+0,10011001b    ;/4 /4
defb stia+8,6            ;pointer
defb stia+0,11110010b,   ;I/O definieren Bit 1 =
                         dtr Ausgang
defb stia+1,0            ;page 0 dtr high
defb stia+8,4
defb stia+0,00000000b    ;disable rest
defb stia+8,5            ;iera
defb stia+0,00010000b    ;bit 4=1 enable receive
                         interupt
defb stia+7,00010000b    ;imra freigeben
defb stia+8,01000110b    ;Vektor xx40

sti1list::               ;STI-B
defb 2*16                ;adjust for

defb stib+12,10001000b   ;/16 1 Stop 8 Bit
defb stib+13,00000001b   ;receive enable
defb stib+14,00000101b   ;transmitter enable
defb stib+8,1            ;pointer to timer d
```

Bild 18. Die STI-Initialisierungstabelle

```
defb stib+0,3            ;/3
defb stib+8,2            ;pointer to timer c
defb stib+0,3            ;/3  9600 baud
defb stib+8,7            ;pointer to control c,d
defb stib+0,10011001b    ;/4 /4
defb stib+8,6            ;pointer
defb stib+0,00000010b    ;I/O definieren Bit 1 =
                         dtr Ausgang
defb stib+1,00000000b    ;dtr high Freigabe
defb stib+8,4
defb stib+0,00000000b    ;disable rest
defb stib+8,5            ;iera
defb stib+0,00000000b    ;disable
;
;
; STI-B Bits   7  6  5  4  3  2  1  0
;              sw sw sw sw sw sw dtr cts
;

; Baudraten Zuordnung
bdtab::
defb 10011001b,1         ;/4 /1   28800 baud
defb 10011001b,2         ;/4 /2   14400 baud
defb 10011001b,3         ;/4 /3    9600 baud
defb 10011001b,3         ;/4 /3    9600 baud
defb 10011001b,3         ;/4 /3    9600 baud
defb 10011001b,3         ;/4 /3    9600 baud
defb 10011001b,3         ;/4 /3    9600 baud
defb 10101010b,230       ;/10 /230   50 baud
defb 10101010b,154       ;/10 /154   75 baud
defb 10101010b,105       ;/10 /105  110 baud
defb 10011001b,96        ;/4 /96    300 baud
defb 10011001b,48        ;/4 /48    600 baud
defb 10011001b,24        ;/4 /24   1200 baud
defb 10011001b,12        ;/4 /12   2400 baud
defb 10011001b,6         ;/4 /6    4800 baud
defb 10011001b,3         ;/4 /3    9600 baud
```

ge, in welcher das Bild gezeichnet werden soll. Darunter ist die Codierung gezeigt. Mit WAXZ... wird der Code übertragen. Das Zeichen cr (Wagenrücklauf) dient der Trennung. Mit dem Befehl WCn r kann die Vergrößerung und Drehung dieses Symbols bestimmt werden. Der Befehl F schließlich macht es sichtbar.

Ein paar Programmbeispiele

Bild 12 zeigt ein Programm in Basic, das ein paar Terminalfunktionen testet. *Bild 13* zeigt ein Beispiel für den Grafik-Modus. *Bild 14* zeigt die Möglichkeit, eine Statuszeile in den Alpha-Modus einzublenden. Diese Textzeile bleibt so lange erhalten, bis der Bildschirm (z. B. mit CTRL-Z) gelöscht wird. Scrolling stört sie nicht. Als Statuszeilenhöhe stehen zwei Textzeilen (25 und 26) zur Verfügung. Die Statuszeile wird mit Hilfe des Grafik-Befehls eingeblendet, weshalb es auch möglich ist, Zeichnungen darin unterzubringen.
Bild 15 zeigt, wie man im Logo-Modus arbeitet. Hierbei ist übrigens zu beachten, daß automatisch eine Umsetzung in ein quadratisches Raster erfolgt (512 x 256 Bildpunkte ergeben nur ein Rechteck). Der Logomodus kann auch mit einem Zusatzparameter aufgerufen werden, so daß 512 x 512 Bildpunkte verwendet werden, um den Einsatz des EF 9365 zu ermöglichen. *Bild 16* schließlich zeigt den Aufruf des Tektronix-Modus. Es ist möglich, einen Bildausschnitt zu wählen und eine Vergrößerung (eigentlich Verkleinerung) anzugeben. Damit lassen sich Bilder mit 1024 x 1024 Punkten auf einem Schirmbild unterbringen und auch Ausschnitte zeigen. Der letzte Parameter gibt an, ob ein Cursor (hier als Fadenkreuz) verwendet werden soll. Wenn ja, so wird durch das schnelle Hin- und Herschalten eine quasisimultane Darstellung erreicht. Zu beachten ist, daß bei allen diesen Darstellungen der EF 9366 verwendet werden sollte. Falls nicht, so muß über den Grafik-Modus der Befehl X2 gegeben werden, um nur alle 40 ms umzuschalten. Der EF 9365 arbeitet selbst schon mit dem Zeilensprungverfahren und ist dementsprechend langsamer. Ferner ist damit unbedingt ein lang nachleuchtender Schirm zu verwenden.

Das Assemblerprogramm

Bild 17 zeigt Ausschnitte aus dem Assemblerlisting. Für eigene Programme, die ab Adresse 4700h bis 47FFh abgelegt werden können, gibt es eine Sprungtabelle, um nützliche Unterprogramme verwenden zu können.
Bild 18 zeigt den Ausschnitt der Programmiertabelle der STIs. *Bild 19* zeigt die Belegung des RAM-Speichers. Mit dem Auslesebefehl WF können damit zum Beispiel der Inhalt des Bildwiederholspeichers oder andere Parameter ausgelesen werden.

Die Arbeit mit dem Terminal

Für das Arbeiten mit diesem umfangreichen Software-Paket sei geraten, einmal jeden Befehl getrennt in einem kleinen Programm zu verwenden und zu testen. Wichtig ist ferner, daß manche Basic-Versionen automatisch CRLF nach einer bestimmten Anzahl von Zeichen einfügen und daher Vorsicht mit allen Binär-Befehlen geboten ist (V-Befehl, m oder d oder Tektronix-Modus). Der mc-Grafik-Modus ignoriert zusätzliche CRLFs, außer bei den Binärbefehlen, der Tektronix-Modus nicht.
(Der Bausatz und das PROM mit Handbuch sind von der Firma GES, Graf Elektronik Systeme, Kempten, unter der Bezeichnung TERM1 lieferbar. Hexdump und EPROM vom Franzis-Software-Service, Tel. 0 89/51 17-3 31.)

```
 rom  abs                                                          checksum
0000  C3 AC 1F C3 CF 00 C3 EF 00 C3 FC 05 C3 83 01 C3   += 08A0
0010  93 01 C3 48 01 C3 C0 00 C3 C7 00 C3 B9 03 C3 C1   += 07B0
0020  03 C3 CA 03 C3 DA 03 C3 E7 03 C3 F5 03 C3 58 04   += 07BA
0030  C3 63 00 C3 8F 08 C3 16 08 C3 04 1A 00 00 00 00   += 0442
0040  60 00 60 00 60 00 60 00 60 00 60 00 60 00 60 00   += 0300
0050  60 00 60 00 60 00 60 00 63 00 60 00 60 00 60 00   += 0303
0060  FB ED 4D E5 D5 C5 F5 CD B9 03 FE FF C2 AB 00 CD   += 0B69
0070  C1 03 2A 8A 38 11 8C 38 19 77 2A 88 38 ED 5B 8A   += 05D1
0080  38 AF ED 52 7C E6 03 67 7C B7 C2 92 00 7D B7 CA   += 0877
0090  A0 00 11 64 00 AF ED 52 CB 7C CA A0 00 CD B9 00   += 073A
00A0  2A 8A 38 23 7C E6 03 67 22 8A 38 F1 C1 D1 E1 FB   += 081E
00B0  ED 4D DB 81 CB 8F D3 81 C9 DB 81 CB CF D3 81 C9   += 0B20
00C0  DB 70 E6 04 28 FA C9 F5 CD C0 00 F1 D3 70 C9 F5   += 0A94
00D0  CD C0 00 F1 E6 F0 47 DB 81 E6 0F B0 D3 81 3E 07   += 0935
00E0  D3 71 3E 04 CD C7 00 CD C0 00 3E 03 D3 71 C9 3E   += 0733
00F0  00 CD CF 00 3E 50 CD CF 00 3E A0 CD CF 00 3E F0   += 076E
0100  CD CF 00 21 BC 3C 11 BD 3C 01 7F 07 36 20 ED B0   += 0639
0110  3E 00 32 84 38 32 85 38 21 A4 3C 0E 00 06 18 71   += 03B9
0120  0C 23 05 C2 1F 01 21 8C 3C 0E 00 06 18 36 00 0C   += 026D
0130  23 05 C2 2D 01 C9 3A 87 38 07 07 E6 0C 47 3A 86   += 04E1
0140  38 E6 03 B0 07 07 07 07 F5 CD C0 00 F1 C5 E6 F0   += 07FB
0150  47 DB 81 E6 0F B0 D3 81 C1 C9 E5 D5 3A 84 38 4F   += 0925
0160  69 06 00 60 29 09 29 E5 3A 85 38 4F 69 06 00 60   += 0424
0170  29 29 09 29 11 F6 00 EB AF ED 52 EB E1 CD 83 01   += 0781
0180  D1 E1 C9 CD C0 00 7C D3 78 7D D3 79 7A D3 7A 7B   += 09DA
0190  D3 7B C9 22 02 38 ED 53 04 38 CD C0 00 DB 78 CB   += 079A
01A0  5F 28 02 F6 F0 57 DB 79 5F ED 53 06 38 2A 02 38   += 065B
01B0  AF ED 52 22 0A 38 DB 7A CB 5F 28 02 F6 F0 57 DB   += 0813
01C0  7B 5F ED 53 08 38 2A 04 38 AF ED 52 22 0C 38 ED   += 0601
01D0  5B 0A 38 CB 7C 28 03 CD 1A 0F CB 7A 28 05 EB CD   += 062F
01E0  1A 0F EB 7C B7 C2 ED 01 7A B7 CA 1C 02 2A 02 38   += 0674
01F0  ED 5B 04 38 E5 D5 2A 0A 38 CB 2C CB 1D ED 5B 06   += 06D7
0200  38 19 E5 2A 0C 38 CB 2C CB 1D ED 5B 08 38 19 EB   += 060F
0210  E1 CD 93 01 D1 E1 CD 93 01 C3 26 02 2A 0A 38 ED   += 0799
0220  5B 0C 38 CD 27 02 C9 06 11 CB 7C 28 05 CD 1A 0F   += 04DF
0230  CB C8 CB 7A 28 07 EB CD 1A 0F EB CB D0 CD C0 00   += 08FB
0240  7D D3 75 7B D3 77 78 CD C7 00 C9 3E 00 18 02 3E   += 06F5
0250  01 CD C7 00 3E 02 C3 C7 00 F5 CD C0 00 F1 D3 72   += 0817
0260  C9 F5 CD C0 00 F1 D3 71 C9 DB 70 E6 02 CA 86 02   += 09D0
0270  3A 81 38 B7 C2 81 02 3C 32 81 38 37 3F C9 C3 83   += 069B
0280  02 37 C9 C3 99 02 3A 81 38 B7 C2 92 02 37 C9 C3   += 0723
0290  99 02 3E 00 32 81 38 37 C9 C9 24 8C 88 8D 01 8E   += 05E1
02A0  05 88 01 80 03 88 02 80 03 88 07 80 99 88 06 80   += 04D4
02B0  F2 81 00 88 04 80 00 88 05 80 10 87 10 88 46 20   += 0521
02C0  9C 88 9D 01 9E 05 98 01 90 03 98 02 90 03 98 07   += 055D
02D0  90 99 98 06 90 02 91 00 98 04 90 00 98 05 90 00   += 0543
02E0  AF 32 62 38 32 81 38 32 6E 38 32 7A 38 32 7E 38   += 050A
02F0  32 82 38 32 67 38 32 6F 38 32 68 38 3E 01 32 83   += 03FC
0300  38 32 7D 38 21 07 45 22 3C 44 21 00 00 22 8A 38   += 0333
0310  22 88 38 21 9A 02 46 23 4E 23 EB A3 05 C2 18 03   += 04EB
0320  3E 02 D3 98 DB 98 E6 03 FE 02 C2 3F 03 3E 01 D3   += 071D
0330  98 DB 98 E6 03 FE 01 C2 3F 03 3E 01 32 0F 38 21   += 0500
0340  BF 02 46 23 4E 23 ED A3 05 C2 44 03 3A 0F 38 B7   += 0571
0350  CA 79 03 DB 91 0F 0F 0F 0F E6 0F 21 99 03 4F 06   += 04F5
0360  00 09 09 3E 47 D3 88 7E D3 80 23 3E 41 D3 88 7E   += 063E
0370  D3 80 3E 42 D3 88 7E D3 80 3E 07 CD C7 00 3E 00   += 0716
0380  CD C7 00 3E 02 CD C7 00 CD EF 00 21 00 00 22 00   += 0567
0390  38 ED 5E 3E 00 ED 47 FB C9 99 01 99 02 99 03 99   += 0723
03A0  03 99 03 99 03 99 03 AA E6 AA 9A AA 69 99 60 99   += 0750
03B0  30 99 18 99 0C 99 06 99 03 DB 8D E6 80 C8 3E FF   += 0794
03C0  C9 DB 80 E6 80 28 FA DB 8F C9 DB 81 E6 01 20 FA   += 0A49
03D0  DB 8E E6 80 28 F4 79 D3 8F C9 3A 0F 38 B7 C8 DB   += 096A
03E0  9D E6 80 C8 3E FF C9 3A 0F 38 B7 C8 DB 9D E6 80   += 09AF
03F0  28 F5 DB 9F C9 3A 0F 38 B7 C8 DB 91 E6 01 20 F5   += 08C8
0400  DB 9E E6 80 28 EF 79 D3 9F C9 CD DA 03 B7 C8 CD   += 0AA0
0410  E7 03 FE 80 C2 1F 04 3E 01 32 68 38 C3 25 04 C5   += 060F
0420  4F CD CA 03 C1 C9 3A 68 38 FE 01 C2 39 04 CD DA   += 07F2
0430  03 37 CA 36 04 3F C3 57 04 E5 D5 CD 0A 04 F3 2A   += 064D
0440  8A 38 ED 5B 88 38 FB AF ED 52 7D B4 C2 53 04 37   += 0834
0450  C3 55 04 37 3F D1 E1 C9 CD 26 04 D8 3A 68 38 FE   += 07B4
0460  01 C2 75 04 CD E7 03 FE 81 C2 72 04 AF 32 68 38   += 072B
0470  37 C9 C3 AB 04 F3 E5 D5 2A 88 38 ED 5B 8A 38 AF   += 08C2
0480  ED 52 7C E6 03 67 11 96 00 AF ED 52 CB 7C C2 94   += 083D
0490  04 CD B2 00 2A 88 38 11 8C 38 19 7E F5 2A 88 38   += 05B6
04A0  23 7C E6 03 67 22 88 38 F1 D1 E1 37 3F F5 3A 62   += 077B
04B0  38 FE 01 C2 BE 04 F1 F5 C5 4F CD F5 03 C1 F1 FB   += 0A27
04C0  C9 3A 85 38 47 CD B4 06 3A 84 38 5F 16 00 19 C9   += 05DB
04D0  CD C1 04 E5 3A 83 38 E6 0F B7 CA 71 05 FE 01 CA   += 0821
04E0  E7 04 FE 02 C2 90 05 F5 4E C5 CD 4B 02 3E 01 32   += 0675
04F0  87 38 CD 36 01 CD 5A 01 3E 0A CD C7 00 CD 5A 01   += 05EF
0500  CD 4F 02 C1 79 CD EE 05 F1 FE 02 C2 2D 05 C5 3E   += 0800
0510  00 32 87 38 CD 36 01 CD 5A 01 CD 4B 02 3E 0A CD   += 054C
0520  C7 00 CD 5A 01 CD 4F 02 C1 79 CD EE 05 C3 71 05   += 0740
0530  FE 03 CA 3A 05 FE 04 C2 71 05 F5 3E 01 32 87 38   += 0669
0540  CD 36 01 CD 5A 01 CD 4B 02 3E F8 CD C7 00 3E A8   += 06F6
0550  CD C7 00 F1 FE 04 C2 71 05 3E 00 32 87 38 CD 36   += 06F1
0560  01 CD 5A 01 CD 4B 02 3E F8 CD C7 00 3E A8 CD C7   += 0787
0570  00 CD 26 04 D2 AB 05 CD 69 02 DA A5 05 3A 82 38   += 0629
0580  FE 0A C2 A1 05 3E 00 32 82 38 3A 86 38 B7 C2 96   += 06A1
0590  05 3E 01 C3 98 05 3E 00 32 86 38 CD 36 01 C3 A5   += 053E
05A0  05 3C 32 82 38 CD 26 04 C3 74 05 E1 3E 00 32 87   += 0538
05B0  38 3E 00 32 86 38 CD 36 01 CD 5A 01 CD 4F 02 3E   += 04EE
05C0  0A CD C7 00 CD 5A 01 CD 4B 02 7E F5 CD EE 05 3E   += 0751
05D0  01 32 87 38 3E 00 32 86 38 CD 36 01 CD 5A 01 CD   += 0519
05E0  4F 02 3E 0A CD C7 00 CD 5A 01 CD 4B 02 F1 B7 F2   += 0709
05F0  F8 05 CD C6 07 C3 FB 05 CD C7 00 C9 CD 4F 02 3E   += 0813
0600  00 CD 61 02 3E 00 CD 59 02 11 00 00 06 04 21 00   += 02D2
0610  00 CD 83 01 0E 08 3E 0B CD C7 00 0D C2 16 06 21   += 0450
0620  40 00 19 EB 05 C2 0E 0E 3E 11 C3 61 02 3A A4 3C   += 04AE
0630  F5 DD 21 A4 3C 06 17 DD 7E 01 DD 77 00 DD 23 05   += 06A5
0640  C2 37 06 F1 DD 77 00 C9 3E 00 32 86 38 3E 01 32   += 05AC
0650  87 38 CD 36 01 CD 4F 02 21 00 00 11 F6 00 CD 83   += 0559
0660  01 CD 39 07 CD 4B 02 21 00 00 11 F6 00 CD 83 01   += 04A1
0670  CD 28 07 3E 01 32 86 38 3E 00 32 87 38 CD 36 01   += 045E
0680  CD 4F 02 21 00 00 11 F6 00 CD 83 01 CD 39 07 CD   += 0571
0690  4B 02 21 00 00 11 F6 00 CD 83 01 CD 28 07 CD 2D   += 04BC
06A0  06 06 17 CD B4 06 3E 00 02 54 5D 13 01 4F 00 36   += 0334
06B0  20 ED B0 C9 D5 58 16 00 21 A4 3C 19 5E 21 8C 3C   += 062A
06C0  19 44 4D 62 6B 29 29 29 29 54 5D 29 29 19 11 BC   += 0404
06D0  3C 19 D1 C9 3E 4F 32 7F 38 18 05 3E 50 32 7F 38   += 04F9
06E0  CD C1 04 3A 84 38 4F DB 70 E6 04 CA E7 06 7E B7   += 07F8
06F0  F2 F9 06 CD C6 07 C3 FB 06 D3 70 23 0C 3A 7F 38   += 07B2
0700  B9 C2 E7 06 C9 3E 17 32 80 38 18 05 3E 18 32 80   += 0595
0710  38 CD C1 04 0A 32 83 38 3A 85 38 47 C5 3A 83 38   += 0579
0720  47 3A 84 38 4F C3 48 07 3E 18 32 80 38 06 01 C5   += 04AA
0730  CD B4 06 0A 47 0E 00 18 0F 3E 18 32 80 38 06 00   += 0353
0740  C5 CD B4 06 0A 47 0E 00 78 B7 CA 68 07 DB 70 E6   += 0744
0750  04 CA 4D 07 7E B7 F2 5F 07 CD C6 07 C3 61 07 D3   += 0747
0760  70 23 0C 79 B8 C2 4D 07 CD CD 00 3E 00 D3 78 D3   += 06CF
0770  79 DB 7A 67 DB 7B 6F 11 0A 00 AF ED 52 7C D3 7A   += 07CC
0780  7D D3 7B C1 04 3A 80 38 B8 C2 40 07 C9 5B 5C 5D   += 0720
0790  7B 7C 7D 7E 7D 0A 09 0A 7D 3D 42 42 42 3D 7D 40   += 0506
07A0  40 40 7D 71 54 54 78 41 00 39 44 44 39 3D 40 40   += 0486
07B0  7D 40 00 7F 01 4D 32 21 8D 07 01 07 00 ED B1 C0   += 04D7
07C0  3E 06 91 F6 80 C9 C5 D5 E5 E6 7F 5F 16 00 62 6B   += 083A
07D0  29 29 19 11 94 07 19 CD ED 0F D9 D5 E5 2A 25 38   += 0613
07E0  ED 5B 27 38 D9 06 05 7E D9 D5 47 0E 08 CB 00 D2   += 06B1
07F0  F7 07 3E 80 CD C7 00 13 CD 83 01 0D C2 ED 07 D1   += 0748
0800  23 CD 83 01 D9 23 05 C2 E7 07 D9 23 CD 83 01 E1   += 0753
0810  D1 D9 E1 D1 C1 C9 CD 25 08 D2 22 08 E6 7F 37 C3   += 093B
0820  24 08 E6 7F C9 CD 58 04 D2 8E 08 3A 6B 38 B7 CA   += 0749
0830  59 08 CD 69 02 DA 59 08 3A 6A 38 FE 01 C2 55 08   += 05CE
0840  3A 6B 38 32 6A 38 3A 86 38 3C E6 03 32 86 38 CD   += 0588
0850  36 01 C3 59 08 3D 32 6A 38 3A 6D 38 B7 CA 88 08   += 055C
0860  CD 69 02 DA 88 08 3A 6C 38 FE 01 C2 84 08 3A 6D   += 0674
0870  38 32 6C 38 3A 86 38 EE 01 E6 03 32 86 38 CD 36   += 05D1
0880  01 C3 88 08 3D 32 6C 38 CD 58 04 C3 28 08 C9 3E   += 058A
0890  00 32 69 38 21 00 00 CD 16 08 FE 20 CA 97 08 FE   += 0564
08A0  2D C2 AC 08 3E 01 32 69 38 CD 16 08 FE 24 C2 24   += 05A8
08B0  09 CD 16 08 FE 3A 30 05 FE 30 D2 D1 08 FE 47 30   += 06AF
08C0  05 FE 41 D2 D1 08 FE 67 30 05 FE 61 D2 D1 08 37   += 07CA
08D0  C9 FE 3A 30 05 FE 30 D2 ED 08 FE 47 30 05 FE 41   += 07E4
08E0  D2 ED 08 FE 67 D2 21 09 FE 61 DA 21 09 29 29 29   += 0706
08F0  29 FE 47 D2 02 09 FE 41 DA 02 09 D6 41 C6 0A C3   += 0719
0900  15 09 FE 67 D2 13 09 FE 61 DA 13 09 D6 61 C6 0A   += 06CD
0910  C3 15 09 D6 30 E6 0F 5F 16 00 19 CD 16 08 C3 D1   += 05E9
0920  08 C3 4B 09 FE 3A 30 05 FE 30 D2 2F 09 37 C9 FE   += 06C2
0930  3A D2 4B 09 FE 30 DA 4B 09 54 5D 29 29 19 29 E6   += 05E7
0940  0F 5F 16 00 19 CD 16 08 C3 2F 09 F5 3A 69 38 FE   += 0551
0950  00 CA 5A 09 DA 5A 09 CD 1A 0F F1 37 3F C9 3E 00   += 05CE
0960  32 6D 38 32 6B 38 CD 4B 02 AF 32 87 38 32 86 38   += 05S6
0970  CD 36 01 CD 16 08 FE 41 CA 97 0C FE 4D C2 95 09   += 0746
0980  CD 8F 08 DA 92 09 E5 CD 8F 08 D1 DA 92 09 EB CD   += 0920
0990  83 01 C3 91 0C FE 44 C2 AF 09 CD 8F 08 DA AC 09   += 0793
09A0  E5 CD 8F 08 D1 DA AC 09 EB CD 93 01 C3 91 0C FE   += 0953
09B0  50 C2 CE 09 CD 8F 08 DA CB 09 7D E6 03 32 86 38   += 0751
09C0  7D 0F 0F E6 03 32 87 38 CD 36 01 C3 91 0C FE 53   += 062A
09D0  C2 F3 09 CD 8F 08 DA F0 09 7D E6 03 32 86 38 7D   += 07C8
09E0  0F 0F E6 03 32 87 38 CD 69 02 DA E7 09 CD 36 01   += 05FE
09F0  C3 91 0C FE 59 C2 08 0A CD 8F 08 DA 05 0A 7D 32   += 0687
0A00  6D 38 32 6C 38 C3 91 0C FE 58 C2 1D 0A CD 8F 08   += 067E
0A10  DA 1A 0A 7D 32 6B 38 32 6A 38 C3 91 0C FE 43 C2   += 0687
0A20  2B 0A CD FC 05 CD 4B 02 C3 91 0C FE 5A C2 39 0A   += 06DA
0A30  CD EF 00 CD 4B 02 C3 91 0C FE 47 C2 6A 0A CD 8F   += 080D
0A40  08 DA 67 0A E5 CD 8F 08 D1 DA 67 0A 7B FE 10 D2   += 0813
0A50  67 0A FE 00 DA 67 0A 7C B7 C2 67 0A CD C0 00 7B   += 0728
0A60  E6 0F F6 70 4F ED 69 C3 91 0C FE 42 C2 8D 0A CD   += 08C6
0A70  16 08 FE 0D CA 8A 0A FE 80 D2 84 0A FE 20 DA 84   += 07E1
0A80  0A CD C7 00 CD 16 08 C3 72 0A C3 91 0C FE 56 C2   += 073E
0A90  A5 0A CD 25 08 B7 CA A2 0A CD C7 00 CD 25 08 C3   += 0727
0AA0  95 0A C3 91 0C FE 6D C2 C6 0A CD 25 08 67 E5 CD   += 080F
0AB0  25 08 E1 6F E5 CD 25 08 57 D5 CD 25 08 D1 5F E1   += 0793
0AC0  CD 83 01 C3 91 0C FE 64 C2 E7 0A CD 25 08 67 E5   += 080C
0AD0  CD 25 08 E1 6F E5 CD 25 08 57 D5 CD 25 08 D1 5F   += 077F
0AE0  E1 CD 93 01 C3 91 0C FE 4F C2 21 0B CD 8F 08 22   += 0763
0AF0  15 38 CD 8F 08 22 17 38 CD 8F 08 22 11 38 CD 8F   += 054D
0B00  08 22 13 38 FE 20 C2 0F 0B CD 8F 08 C3 10 0B 37   += 04E8
0B10  D2 17 0B AF C3 18 0B 7D 32 3D 38 CD 05 10 C3 91   += 05E3
0B20  0C FE 52 C2 64 0B CD 8F 08 22 31 38 CD 8F 08 22   += 0602
0B30  33 38 CD ED 0F 2A 27 38 ED 4B 33 38 09 EB 2A 25   += 05A3
0B40  38 E5 D5 CD 93 01 D1 E1 ED 4B 31 38 09 E5 CD 93   += 08F4
0B50  01 E1 ED 5B 27 38 D5 CD 93 01 D1 2A 25 38 CD 93   += 0777
0B60  01 C3 91 0C FE 4C C2 A1 0B CD 8F 08 E5 CD 8F 08   += 07C6
0B70  EB E1 CD 83 01 CD ED 0F E5 21 94 0B E3 CD 8F 08   += 08D2
0B80  FE 20 C0 E5 CD 8F 08 D1 EB E5 CD 93 01 F1 FE 20   += 0A48
0B90  C0 C3 7D 0B 2A 25 38 ED 5B 27 38 CD 93 01 C3 91   += 06EE
0BA0  0C FE 6C C2 AC 0B CD 11 11 C3 91 0C FE 4A C2 D7   += 081F
0BB0  0B CD 8F 08 22 31 38 CD 8F 08 22 33 38 CD ED 0F   += 05B4
0BC0  2A 27 38 ED 4B 33 38 09 EB 2A 25 38 ED 4B 31 38   += 0548
0BD0  09 CD 93 01 C3 91 0C FE 57 C2 E2 0B CD A3 0C C3   += 080D
0BE0  91 0C FE 46 C2 55 0C CD 8F 08 DA 47 0C E5 CD 8F   += 07D6
0BF0  08 D1 DA 44 0C ED 53 74 38 22 76 38 CD 8F 08 DA   += 06FD
0C00  44 0C 3A 87 38 F5 7D E6 03 32 78 38 CD D1 0D DA   += 0708
0C10  3D 0C CD 0B 0E ED 5B 76 38 ED 53 71 38 2A 74 38   += 05E4
0C20  22 8F 38 CD 83 01 3A 78 38 32 73 38 32 87 38 CD   += 059F
0C30  36 01 CD 4B 02 CD 38 0E 3E 01 32 6E 38 F1 32 87   += 0525
0C40  38 CD 36 01 C3 52 0C CD 0B 0E 3E 00 32 6E 38 CD   += 0526
0C50  4B 02 C3 91 0C FE 1B C2 91 0C CD 16 08 FE 1B C2   += 06EB
0C60  91 0C CD 16 08 FE 41 C2 70 0C C3 97 0C C3 91 0C   += 06CB
0C70  FE 43 C2 7B 0C CD C2 19 C3 91 0C FE 52 C2 87 0C   += 0837
0C80  AF 32 62 38 C3 91 0C FE 50 C2 91 0C 3E 01 32 62   += 065B
0C90  38 CD 16 08 C3 76 09 CD C0 00 3E 11 D3 73 3E 00   += 05C5
0CA0  D3 72 C9 CD 16 08 FE 41 C2 3F 0D CD 0B 0E 3E 00   += 066A
0CB0  32 6E 38 CD 16 08 FE 60 30 05 FE 40 D2 CF 0C FE   += 0739
0CC0  30 CA C9 0C FE 31 C2 29 0D 21 07 45 22 7B 38 FE   += 0636
0CD0  30 C2 DA 0C 3E 03 77 C3 02 0D FE 31 C2 E5 0C 3E   += 0682
0CE0  02 77 C3 02 0D E6 1F 47 DD 21 C1 0D E6 07 5F 16   += 05C5
0CF0  00 DD 19 78 E6 18 DD B6 00 47 07 07 E6 60 F6 80   += 0710
0D00  80 77 23 E5 CD 16 08 E1 FE 60 30 05 FE 40 D2 CF   += 086D
0D10  0C FE 30 CA CF 0C FE 31 CA CF 0C 36 00 23 3E 01   += 064B
0D20  32 7A 38 22 3C 44 C3 3C 0D 3A 7A 38 FE 01 C2 3C   += 057B
0D30  0D 3E 00 32 7A 38 2A 7B 38 22 3C 44 C3 C0 0D FE   += 053C
0D40  42 C2 4A 0D CD 38 0E C3 C0 0D FE 43 C2 7F 0D CD   += 075A
0D50  0B 0E 3E 00 32 6E 38 CD 8F 08 DA 72 0D E5 CD 8F   += 062D
0D60  08 D1 DA 6F 0D 7B 32 7D 38 7D E6 07 32 7E 38 C3   += 06A6
```

Bild 20. Der Hex-Dump

rom	abs																checksum
1B10	67	3A	C3	68	1B	FE	1B	C2	68	1B	CD	04	1A	FE	47	C2	+= 0735
1B20	28	1B	CD	5E	09	C3	68	1B	FE	54	C2	33	1B	CD	2E	14	+= 062E
1B30	C3	68	1B	FE	43	C2	3E	1B	CD	80	19	C3	68	1B	FE	50	+= 079C
1B40	C2	4B	1B	3E	01	32	62	3A	C3	68	1B	FE	52	C2	57	1B	+= 05FD
1B50	AF	32	62	3A	C3	68	1B	FE	4C	C2	62	1B	CD	D0	11	C3	+= 07BB
1B60	68	1B	FE	52	C2	68	1B	00	C9	3E	00	32	86	3A	3E	01	+= 054E
1B70	32	87	3A	CD	36	01	CD	5A	01	CD	4F	02	CD	DB	06	3E	+= 0627
1B80	01	32	86	3A	3E	00	32	87	3A	CD	36	01	CD	5A	01	CD	+= 0519
1B90	4F	02	CD	DB	06	CD	C1	04	3A	84	3A	3C	02	3A	84	3A	+= 05BB
1BA0	47	3E	50	90	47	36	20	23	05	C2	A5	1B	C9	3E	00	32	+= 04E5
1BB0	86	3A	3E	01	32	87	3A	CD	36	01	CD	5A	01	CD	4F	02	+= 0538
1BC0	CD	DC	07	3E	01	32	86	3A	3E	00	32	87	3A	CD	36	01	+= 0442
1BD0	CD	5A	01	CD	4F	02	CD	DC	07	CD	C1	04	11	3B	44	EB	+= 0633
1BE0	AF	ED	52	E5	CD	C1	04	54	5D	13	36	20	C1	78	B1	CA	+= 0833
1BF0	F4	1B	ED	B0	C9	3E	00	32	86	3A	3E	01	32	87	3A	CD	+= 06A0
1C00	36	01	CD	5A	01	CD	4F	02	CD	DB	06	CD	5A	01	CD	4B	+= 0668
1C10	02	3E	20	CD	C7	00	3A	84	3A	FE	4F	D2	21	1C	CD	D4	+= 06E7
1C20	06	3E	01	32	86	3A	3E	00	32	87	3A	CD	36	01	CD	5A	+= 048F
1C30	01	CD	4F	02	CD	DB	06	CD	5A	01	CD	4B	02	3E	20	CD	+= 063A
1C40	C7	00	3A	84	3A	FE	4F	D2	4D	1C	CD	D4	06	CD	C1	04	+= 077E
1C50	0A	FE	50	D2	58	1C	3C	02	3A	84	3A	47	0E	20	7E	71	+= 0536
1C60	4F	23	04	78	FE	50	C2	5E	1C	C9	3E	00	32	86	3A	3E	+= 05AD
1C70	01	32	87	3A	CD	36	01	CD	5A	01	CD	4F	02	CD	DB	06	+= 05EA
1C80	CD	5A	01	CD	4B	02	3A	84	3A	F5	3C	FE	50	D2	96	1C	+= 0738
1C90	32	84	3A	CD	DB	06	F1	32	84	3A	3E	01	32	86	3A	3E	+= 05E8
1CA0	00	32	87	3A	CD	36	01	CD	5A	01	CD	4F	02	CD	DB	06	+= 05E9
1CB0	CD	5A	01	CD	4B	02	3A	84	3A	F5	3C	FE	50	D2	C6	1C	+= 076B
1CC0	32	84	3A	CD	DB	06	F1	32	84	3A	CD	C1	04	3A	84	3A	+= 0703
1CD0	47	E5	DD	E1	78	FE	4F	D2	E6	1C	DD	7E	01	DD	77	00	+= 0933
1CE0	DD	23	04	C3	D4	1C	DD	36	00	20	C9	3E	00	32	84	3A	+= 05DF
1CF0	3E	00	32	86	3A	3E	01	32	87	3A	CD	36	01	CD	4F	02	+= 0480
1D00	CD	5A	01	CD	0C	07	CD	4B	02	3A	85	3A	FE	17	D2	20	+= 0620
1D10	1D	F5	3C	32	85	3A	CD	5A	01	F1	32	85	3A	CD	05	07	+= 061E
1D20	3E	01	32	86	3A	3E	00	32	87	3A	CD	36	01	CD	4F	02	+= 0480
1D30	CD	5A	01	CD	0C	07	CD	4B	02	3A	85	3A	FE	17	D2	50	+= 0650
1D40	1D	F5	3C	32	85	3A	CD	5A	01	F1	32	85	3A	CD	05	07	+= 061E
1D50	21	A4	3C	3A	85	3A	4F	06	00	09	3A	85	3A	47	3A	BB	+= 0489
1D60	3C	4F	78	FE	18	D2	70	1D	7E	71	4F	23	04	C3	62	1D	+= 061F
1D70	3A	85	3A	47	CD	B4	06	3E	00	02	54	5D	13	01	4F	00	+= 0419
1D80	36	20	ED	B0	C9	3E	00	32	84	3A	3E	00	32	86	3A	3E	+= 0554

rom	abs																checksum
1D90	01	32	87	3A	CD	36	01	CD	4F	02	CD	5A	01	CD	0C	07	+= 051C
1DA0	CD	4B	02	CD	5A	01	3A	85	3A	FE	17	D2	BA	1D	F5	3C	+= 0728
1DB0	32	85	3A	CD	0C	07	F1	32	85	3A	3E	01	32	86	3A	3E	+= 051C
1DC0	00	32	87	3A	CD	36	01	CD	4F	02	CD	5A	01	CD	0C	07	+= 051B
1DD0	CD	4B	02	CD	5A	01	3A	85	3A	FE	17	D2	EA	1D	F5	3C	+= 0758
1DE0	32	85	3A	CD	0C	07	F1	32	85	3A	21	A4	3C	3A	85	3A	+= 05A7
1DF0	4F	06	00	09	7E	F5	E5	DD	E1	3A	85	3A	47	78	FE	17	+= 073F
1E00	D2	0F	1E	DD	7E	01	DD	77	00	DD	23	04	C3	FD	1D	F1	+= 0781
1E10	DD	77	00	06	17	CD	B4	06	3E	00	02	54	5D	13	01	4F	+= 044C
1E20	00	36	20	ED	B0	C9	FE	08	C2	53	1E	3A	84	3A	B7	C2	+= 0764
1E30	4C	1E	3A	85	3A	B7	CA	49	1E	3E	4F	32	84	3A	3A	85	+= 0583
1E40	3A	B7	CA	49	1E	3D	32	85	3A	C3	5D	1E	3D	32	84	3A	+= 05A8
1E50	C3	F9	1E	FE	1A	C2	5E	1E	CD	EF	00	C3	F9	1E	FE	1E	+= 08E2
1E60	C2	6E	1E	3E	00	32	84	3A	32	85	3A	C3	F9	1E	FE	07	+= 0648
1E70	C2	7B	1E	0E	07	CD	F5	03	C3	F9	1E	FE	0B	C2	8E	1E	+= 0786
1E80	3A	85	3A	B7	CA	8B	1E	3D	32	85	3A	C3	F9	1E	FE	0A	+= 072F
1E90	C2	A8	1E	3A	85	3A	FE	17	D2	A2	1E	3C	32	85	3A	C3	+= 0714
1EA0	A5	1E	CD	48	06	C3	F9	1E	FE	16	C2	BC	1E	3A	85	3A	+= 075F
1EB0	FE	17	D2	B9	1E	3C	32	85	3A	C3	F9	1E	FE	0C	CA	C6	+= 085D
1EC0	1E	FE	09	C2	EF	1E	3A	84	3A	FE	4F	C2	E8	1E	3E	00	+= 073D
1ED0	32	84	3A	3A	85	3A	FE	17	C2	E1	1E	CD	48	06	C3	E5	+= 077E
1EE0	1E	3C	32	85	3A	C3	EC	1E	3C	32	84	3A	C3	F9	1E	FE	+= 0718
1EF0	0D	C2	F9	1E	3E	00	32	84	3A	C9	E6	7F	47	3A	67	3A	+= 0660
1F00	FE	01	C2	0C	1F	78	CD	B7	07	C3	0D	1F	78	F5	3E	00	+= 0689
1F10	32	87	3A	CD	36	01	CD	5A	01	CD	C1	04	0A	5F	3A	84	+= 05D6
1F20	3A	3C	BB	CA	2A	1F	DA	2A	1F	02	7E	32	66	3A	FE	20	+= 05D3
1F30	CA	3E	1F	CD	4F	02	3E	0A	CD	C7	00	CD	5A	01	CD	4B	+= 0661
1F40	02	F1	F5	77	B7	F2	4E	1F	CD	C6	07	C3	51	1F	CD	C7	+= 0806
1F50	00	3E	01	32	87	3A	CD	36	01	CD	5A	01	3A	66	3A	FE	+= 0532
1F60	20	CA	72	1F	CD	4F	02	3E	0A	CD	C7	00	CD	5A	01	CD	+= 066A
1F70	4B	02	F1	B7	F2	7D	1F	CD	C6	07	C3	80	1F	CD	C7	00	+= 0813
1F80	3A	84	3A	3C	32	84	3A	FE	4F	CA	AB	1F	DA	AB	1F	3E	+= 06E3
1F90	00	32	84	3A	3A	85	3A	3C	32	85	3A	FE	17	CA	AB	1F	+= 05B9
1FA0	DA	AB	1F	3E	17	32	85	3A	CD	48	06	C9	31	06	45	CD	+= 0615
1FB0	E0	02	E5	21	E9	1F	E3	CD	58	04	D2	C6	1F	CD	D0	04	+= 0854
1FC0	CD	58	04	C3	BA	1F	E6	7F	FE	1B	C2	D3	1F	CD	1F	1A	+= 07FD
1FD0	C3	E6	1F	FE	20	D2	DE	1F	CD	26	1E	C3	E6	1F	FE	20	+= 08AC
1FE0	DA	E6	1F	CD	FA	1E	C3	B7	1F	76	00	00	00	00	00	00	+= 05D3
1FF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	+= 0000

Rolf-Dieter Klein:

Centronics-Schnittstelle mit Software

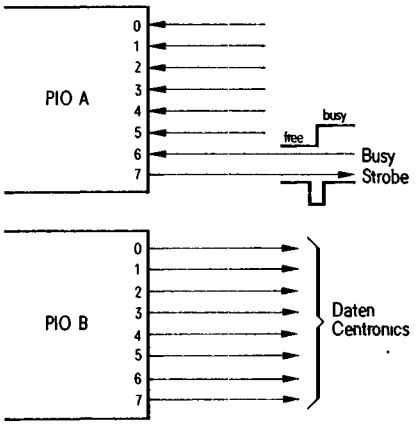
Im Monitor 3.4 für den mc-CP/M-Computer sind nur Routinen zur Bedienung einer seriellen Schnittstelle vorhanden, es gibt jedoch eine Vielzahl von parallel anzusteuernden Druckern und Peripheriegeräten auf dem Markt, die meist preiswerter sind, als die mit einer seriellen Schnittstelle ausgerüsteten.

Als Parallelport wird die PIO auf der SIO/PIO-Karte verwendet. Das Programm zur Bedienung wird einfach unter CP/M als Kommando aufgerufen und lädt die Routine in den Speicher. *Bild 1* zeigt den hardwaremäßigen Anschluß. Die PIOB dient dabei als Datenport, die PIOA als Steuerport. Ein negativer Strobe an Bit 7, PIO A gibt die Daten aus und lädt sie in den Drucker. Ein Busy-Eingang kann dies verhindern, wenn er auf High-Pegel liegt. Dann wartet das Interfaceprogramm auf die Freigabe. Damit ist es möglich, die Ausgabegeschwindigkeit des Computers mit der des Peripheriegerätes zu synchronisieren. *Bild 2* zeigt das Assemblerlisting des Programms und *Bild 3* einen Hexdump. Das Programm wird auf Adresse 100H gestartet. Es verschiebt das eigentliche Programm auf die Adresse FB00. Dort wird der Monitor überschrieben. Ebenfalls wird der Sprungvektor LO am Anfang des Monitors durch einen neuen ausgetauscht.
Das kleine Programm kann als .COM-Datei mit der Sequenz SAVE 1 CENT-.COM abgelegt werden, nachdem es zuvor von Hand auf die Adresse 100H eingegeben und danach CP/M gebootet wurde.

```
                        .z80
                        ;**************************************
                        ;* Centronics Installationsroutine    *
                        ;* Port B ist Data 0..7               *
                        ;* Port A bit 7 = -strobe             *
                        ;* Port A bit 6 = busy                *
                        ;* Rolf-Dieter Klein 1982 821218      *
                        ;**************************************
00F4                    pioa    equ     0f4h
00F5                    pioasts equ     pioa + 1
00F6                    piob    equ     pioa + 2
00F7                    piobsts equ     pioa + 3

0000'                   start:
0000'  3E CF                    ld a,0cfh       ;bit mode
0002'  D3 F5                    out (pioasts),a
0004'  3E 7F                    ld a,01111111b  ;bit 7 ist output
0006'  D3 F5                    out (pioasts),a ;bit 0,1 floppy ready
0008'  3E CF                    ld a,0cfh
000A'  D3 F7                    out (piobsts),a ;als output
000C'  3E 00                    ld a,0          ;definieren
000E'  D3 F7                    out (piobsts),a
0010'  3E 80                    ld a,80h        ;strobe = 1
0012'  D3 F4                    out (pioa),a    ;init
                        ;
0014'  21 FB00                  ld hl,lo        ;adresse vektor umstellen
0017'  22 F010                  ld (0f00fh+1),hl
                        ;
001A'  21 0077'                 ld hl,lorel
001D'  11 FB00                  ld de,lo        ;reale adresse
0020'  01 0019                  ld bc,loend-lo  ;laenge
0023'  ED B0                    ldir            ;transport
                        ;
0025'  21 002E'                 ld hl,msg
0028'  CD 006D'                 call print
002B'  CD 0000                  call 0          ;warmboot
                                ;
```

Bild 2. Assemblerlisting des Programms „Centronics“



◀ Bild 1. Anschlußbelegung


```
002E'  43 65 6E 74      msg:    defm 'Centronics Routine installiert'
0032'  72 6F 6E 69
0036'  63 73 20 52
003A'  6F 75 74 69
003E'  6E 65 20 69
0042'  6E 73 74 61
0046'  6C 6C 69 65
004A'  72 74
004C'  0D 0A                    defb 0dh,0ah
004E'  56 65 6B 74              defm 'Vektor LO (LST:) adr 0ff20h '
0052'  6F 72 20 4C
0056'  4F 20 28 4C
005A'  53 54 3A 29
005E'  20 61 64 72
0062'  20 30 66 66
0066'  32 30 68 20
006A'  0D 0A                    defb 0dh,0ah
006C'  00                       defb 0

     MACRO-80 3.43  27-Jul-81       PAGE    1-1

006D'  7E               print:  ld a,(hl)
006E'  B7                       or a    ;
006F'  C8                       ret z
0070'  4F                       ld c,a
0071'  CD F009                  call 0f009h     ;monitor co
0074'  23                       inc hl
0075'  18 F6                    jr print
                        ;


0077'                   lorel:                  ;adresse lo in ram

                        ; Monitor wird dort ueberlagert
                        ;
                        .phase 0fb00h           ;
FB00   79               lo:     ld a,c          ;je nach Druckertyp
FB01   FE 0A                    cp 0ah          ;linefeeds ueberlesen
FB03   C8                       ret z           ;da sonst cr lf
                                                ;als nicht vorhanden gilt
                                                ;und vorschuebe verloren gehen
FB04                    loi:
FB04   DB F4                    in a,(pioa)     ;test status
FB06   E6 40                    and 40h         ;=1 dann busy
FB08   20 FA                    jr nz,loi       ;warten dann
FB0A   79                       ld a,c
FB0B   D3 F6                    out (piob),a    ;daten definieren
FB0D   DB F4                    in a,(pioa)     ;ggf alter wert
FB0F   CB BF                    res 7,a         ;strobe auf 0
FB11   D3 F4                    out (pioa),a
FB13   CB FF                    set 7,a         ;wieder auf 1
FB15   D3 F4                    out (pioa),a
FB17   79                       ld a,c          ;konventionell data nach a
FB18   C9                       ret
                        ;
FB19                    loend:

                        end start
```

```
0100: C3 03 01 3E CF D3 F5 3E 7F D3 F5 3E CF D3 F7 3E ...>...>...>...>
0110: 00 D3 F7 3E 80 D3 F4 21 00 FB 22 10 F0 21 7A 01 ...>...!..".!z.
0120: 11 00 FB 01 19 00 ED B0 21 31 01 CD 70 01 CD 00 ........!1..p...
0130: 00 43 65 6E 74 72 6F 6E 69 63 73 20 52 6F 75 74 .Centronics Rout
0140: 69 6E 65 20 69 6E 73 74 61 6C 6C 69 65 72 74 0D ine installiert.
0150: 0A 56 65 6B 74 6F 72 20 4C 4F 20 28 4C 53 54 3A .Vektor LO (LST:
0160: 29 20 61 64 72 20 30 66 66 32 30 68 20 0D 0A 00 ) adr 0ff20h ...
0170: 7E B7 C8 4F CD 09 F0 23 18 F6 79 FE 0A C8 DB F4 ~..O...#..y.....
0180: E6 40 20 FA 79 D3 F6 DB F4 CB BF D3 F4 CB FF D3 .@ .y...........
0190: F4 79 C9 .y
```

Bild 3. Hexdump ▶ zu „Centronics “

Reinhard Jäger, Hans-Joachim Regge, Günter Tobergte:

Das mc-Terminal

Im folgenden Artikel wird eine universelle Terminalkarte beschrieben, die in Verbindung mit einem Video-Monitor und einer Tastatur für den mc-CP/M-Computer verwendet werden kann (wie auch für jeden anderen Computer, der eine serielle Schnittstelle besitzt). Es wurde besonderer Wert gelegt auf die Verwendung handelsüblicher Teile. Die Leistung des Gerätes kann sich mit der üblicher Terminals durchaus messen und übertrifft sie sogar. Durch eine entsprechend vorbereitete Software ist eine Anpassung an verschiedene Tastaturen und Rechner möglich.

Die Hardware ist auf einer Europakarte mit DIN 41 612-Steckverbinder untergebracht. Für den Aufbau eines kompakten Terminals ist die Karte mit einer leicht abtrennbaren Verlängerung versehen, auf der die Stromversorgung für +5 V und ± 12 V Platz findet. Es ist dann lediglich noch ein Transformator erforderlich, der etwa 7,5 V bei 1,5 A liefern muß. Außerdem ist eine genormte Terminal-Steckverbindung (25polig) vorgesehen; für die Verbindung zur Tastatur ist ebenfalls eine Steckverbindung auf der Verlängerung vorhanden (*Tabelle 1*). Wer die Stromversorgung mit einem vorhandenen Netzteil übernehmen möchte, findet natürlich eine Klemmleiste. Dem Wandler können noch ± 12 V/20 mA für andere Zwecke entnommen werden.

Hardware-Eigenschaften

Bei dem hier beschriebenen Terminal ist besonders hervorzuheben, daß durch die hierfür ausgelegte Software jeder Taste jedes Zeichen zugeordnet werden kann. Es müssen nur die Tabellen im Programm (EPROM) geändert werden. Selbstverständlich sind auch acht verschiedene Übertragungsraten bis 19 200 Baud (!) sowie ein oder zwei Stopbits, keine, gerade oder ungerade Parität mittels eines DIL-Schalters einstellbar. Es können 7 oder 8 Datenbits übertragen werden. Zudem kann noch zwischen fünf Zeichensätzen gewählt werden. Mit einem weiteren Schalter läßt sich ein Autolinefeed zuschalten, das jedoch nur im Online-Betrieb wirksam ist. Besonders hervorzuheben ist hier natürlich die hohe Übertragungsrate von 19 200 Baud, die durch die Verwendung des 16-Bit-Prozessors TMS-9995 von Texas Instruments möglich wurde.

Die Hardware im einzelnen

Kernstück des Prozessorteils ist der Prozessor TMS-9995. Es handelt sich hierbei um eine Weiterentwicklung des bekannten 9900 mit externem 8-Bit-Datenbus. Intern arbeitet der Prozessor wie der 9900 mit einer Datenbreite von 16 Bit, beide sind weitgehend kompatibel. Zu beachten ist, daß bei TI die Wertigkeit der Bits anders ist: MSB = Bit 0, LSB = Bit 15 beim Adreßbus; MSB = Bit 0 und LSB = Bit 7 beim Datenbus. Bei den sonst üblichen CPUs wie Z80, 6502 oder 6809 ist es umgekehrt.
Kernstück des Video-Teils ist ein Video-Controller MC-6845 von Motorola (1-MHz-Version). Dieser Baustein enthält interne Register für das Video-Timing (Bildformat), ein Cursor-Register und ein Light-Pen-Register. Die Benutzung eines Light-Pen ist ohne erhebliche Software- und Hardwareänderungen nicht möglich. Ferner liefert der 6845 Ausgangssignale für die Horizontal- und Vertikalsynchronisation, die über ein Exklusiv-Oder verknüpft sind.
Nähere Angaben zu beiden Bausteinen sind den Datenblättern der Hersteller zu entnehmen.
Der Zeichenspeicher kann alternativ vom CRT-Controller 6845 oder von der CPU 9995 adressiert werden. Wird der Zeichenspeicher von der CPU adressiert, so wird gleichzeitig mit dem Umschalten des Adreßmultiplexers auch der Datenbus umgeschaltet, so daß der Zeichenspeicher von der CPU beschrieben oder gelesen werden kann. Adressiert der CRT-Controller den Zeichenspeicher, so wird ein ASCII-Zeichen an der aktuellen Position auf dem Bildschirm ausgegeben. Zusammen mit der entsprechenden Reihenadresse wird aus dem Zeichengenerator eine Reihe einer 6 × 10-Matrix gelesen und über ein Schieberegister von einem parallelen in ein serielles Signal gewandelt, das Video-TTL-Signal.
Das Video-TTL-Signal wird zunächst mit dem Cursor-Signal des 6845 verknüpft und dadurch die Cursor-Position auf dem Bildschirm festgelegt. Das bei ASCII-Zeichen nicht verwendete höchstwertige Bit im Video-RAM wird zur inversen Darstellung auf dem Schirm genutzt und dementsprechend mit dem Video-Signal über ein Exklusiv-Oder verknüpft. Das Display-Enable-Signal sorgt dafür, daß nur der aktive Teil der Zeile auf dem Bildschirm dargestellt wird. Das Ausgangssignal für den Monitor entsteht dann durch die Verknüpfung von Video- und Synchronisationssignal.
Das Ansprechen des Programm-EPROMs, des 6845, des Zeichengenerators und der parallelen Tastaturschnittstelle erfolgt „memory mapped", dies gilt gleichfalls für das Video-RAM und das prozessor-interne RAM. Der UART-Baustein 9902 hingegen wird über den CRU-Bus (Communication-Register-Unit) angesprochen, einen TI-eigenen I/O-Bus. Durch die serielle Struktur dieses CRU-Busses ergeben sich für den UART-Baustein günstige Abmessungen (nur 18 Pins). An diesen Baustein sind über entsprechende Treiber die RS-232-Schnittstelle sowie die 20-mA-Schnittstelle (über Opto-Koppler) angeschlossen.
Das Sendesignal (XMT) des 9902 wird auf einen RS-232-Treiber (SN 75188) geschaltet. Die Versorgung dieses Bausteines erfordert ± 12 V. Parallel hierzu wird das XMT-Signal dem Optokoppler vom Typ 4N33 zugeführt, der das Signal galvanisch getrennt zum Schalten der 20-mA-Schleife bereitstellt. Das 20-mA-Empfangssignal wird ebenfalls über einen Optokoppler geführt. Mit Hilfe einer Steckbrücke B2 kann entweder das RS-232-Signal oder der Ausgang des Opto-

kopplers an den Schmitt-Trigger gelegt werden, der aus einem der beiden Quellsignale ein Empfangssignal (RCV) mit TTL-Pegel formt und dieses dem 9902 zuführt.
Das Terminal kann also über beide Schnittstellen gleichzeitig senden, aber abhängig von der Steckbrücke B2 nur über eine Schnittstelle empfangen. Bei Betrieb der 20-mA-Schnittstelle ist das Terminal immer passiv, d. h. der Rechner muß den Strom liefern.
Bild 1 zeigt die Gesamtschaltung der Terminalkarte, jedoch ohne die Stromversorgung auf der Platinenverlängerung, diese Schaltung ist in *Bild 2* gezeigt. Die verwendeten Bauteile sind in *Tabelle 2* aufgelistet.

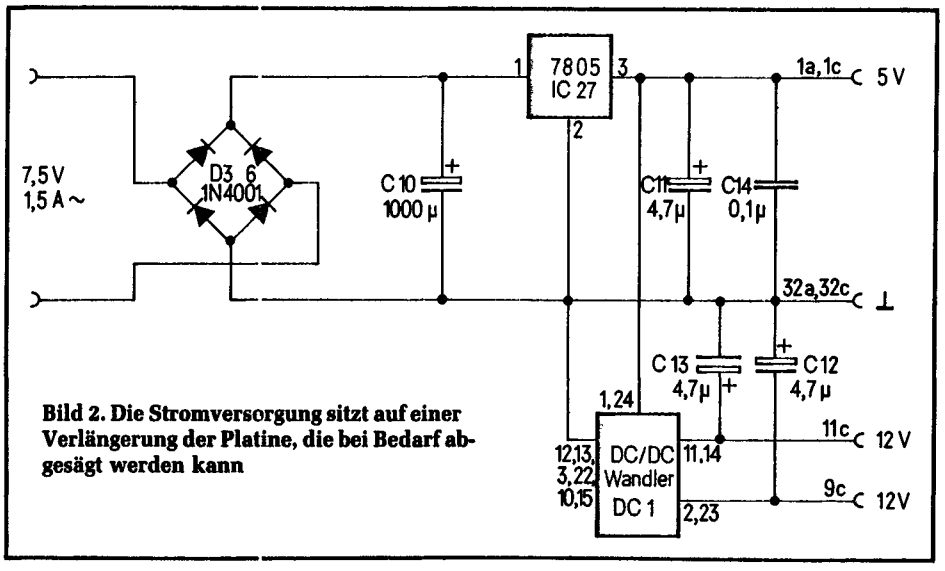


Bild 2. Die Stromversorgung sitzt auf einer Verlängerung der Platine, die bei Bedarf abgesägt werden kann

Hinweise zum Aufbau

Der Aufbau der Platine kann nur Lesern empfohlen werden, die im Löten geübt sind und im Aufbau und Bestücken von gedruckten Schaltungen bereits Erfahrung haben. Die Platine ist relativ gut bestückt, wie man *Bild 3, 4* und *Bild 5* unschwer entnehmen kann.

Trotz Lötstopmaske sollte ein Lötkolben mit feiner Spitze und einer Leistung von nicht mehr als 30 W sowie möglichst dünnes Lötzinn verwendet werden, um ungewollte Lötbrücken zu vermeiden. Nach Bestücken der Platine mit den IC-Fassungen (es sollten alle ICs auf nicht zu billige Fassungen gesetzt werden) ist die Lötseite sorgfältig auf Zinnspritzer und ungewollte Verbindungen zu prüfen. Zur Probe kann die Platine dann an die Versorgungsspannungen angeschlossen werden. Die Spannungen dürfen nur an den entsprechenden Pins der ICs meßbar sein, die Stromaufnahme ist dabei gleich Null.
Nach dem Einlöten der Steckverbindung(en) sowie der übrigen passiven Bauteile und nochmaligem Vergleich mit dem Bestückungsplan (sind die Tantal-Elkos und die Dioden richtig gepolt?) wird die Platine erneut an die Betriebsspannung angeschlossen (ohne ICs!). Die Stromaufnahme darf nur wenige mA betragen. Vorsichtige Leute prüfen noch mit einem Ohmmeter bei abgeschalteter Spannung die Daten- und Adreßleitungen auf etwaige Kurzschlüsse gegeneinander, gegen Masse oder eine der Betriebsspannungen. Nun können alle integrierten Bausteine in die Fassungen gesetzt werden.
Zu beachten ist die unterschiedliche Lage der ICs (Bild 4). Die Stromaufnahme der Karte beträgt ohne Prozessor und ohne CRT-Controller etwa 800 mA für die 5-V-Spannung, die beiden 12-V-Spannungsquellen werden mit je 20 mA belastet. Ist bis dahin alles in Ordnung, so können die „teuren" Bausteine eingesetzt werden. Das Einsetzen und Entfernen von ICs darf selbstverständlich nur bei abgeschalteter Betriebsspannung erfolgen. Die gesamte Stromaufnahme beträgt dann 1 A.
Zunächst kann man die Adreß- und Datenbusleitungen auf „Leben" untersuchen. Die Taktleitung muß ein sauberes Rechtecksignal von 2,5 MHz liefern, was sich auch mit einem Zähler oder einem Oszilloskop an Pin 3 des Prozessors

Tabelle 1: Die Steckerbelegung auf einen Blick

ST 1	ST 2		
4a	11	Bit 0 (LSB)	Tastatur
5c	4	Bit 1	
6c	5	Bit 2	
5a	12	Bit 3	
2a	9	Bit 4	
3c	2	Bit 5	
4c	3	Bit 6	
3a	10	Bit 7 (MSB)	
17c	15	Strobe (neg./pos.)	
	1	+5 V/100 mA	
	6	+12 V/20 mA	
	7	−12 V/20 mA	
	8	Masse	
	ST 3		
10c	2	XMT (Sender)	RS-232
32a	7	GND (Masse)	
12c	3	RCV (Empfänger)	
31c	25	Sender-Kollektor	20 mA
31a	13	Sender-Emitter	
28c	12	Empfänger-Anode	
29c	24	Empfänger-Katode	
1		+5 V/1 A	Betriebsspannungen
32a/32c		Masse	
11c		+12 V/30 mA	
9c		−12 V/30 mA	
8c		Video-Ausgang 1 V, 75 Ω	
27c		Signal, neg. Impuls 450 ns	
14c		Terminal Reset (nur f. Tests)	

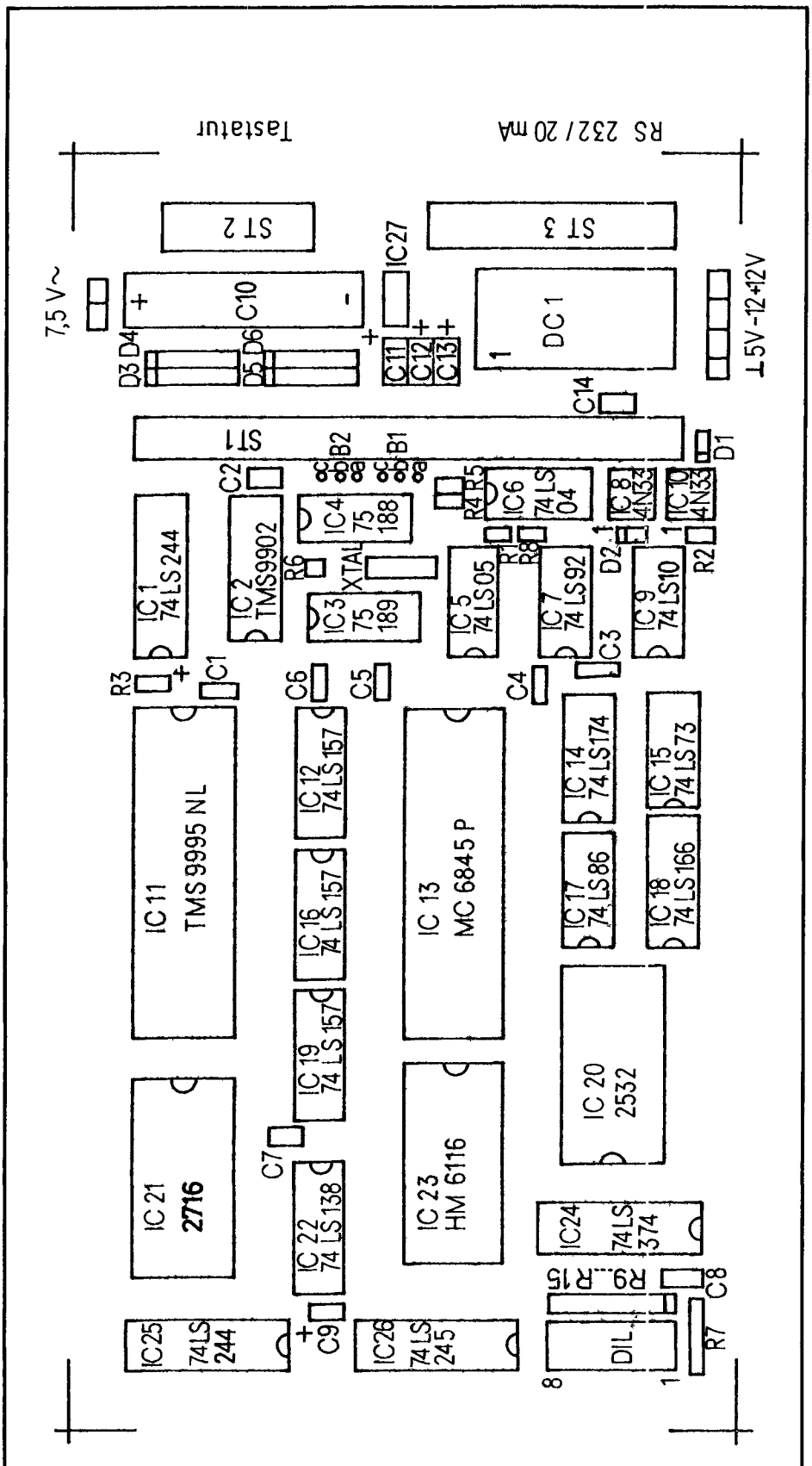


Bild 4. Aus dem Bestückungsplan ist die Lage der einzelnen Bauelemente ersichtlich

Beide Verfahren haben Vor- und Nachteile. Hat der Hauptrechner direkten Zugriff auf den Bildschirm, so ist ein schnelles Screen-Editing oder auch Grafik möglich. Der Bildschirm muß sich jedoch unmittelbar am Rechner befinden und die Rechengeschwindigkeit sinkt durch die zusätzliche Arbeit.
Ist der Rechner mit dem Terminal über eine serielle Schnittstelle verbunden, so können abgesetzte Terminals (auch über Telefonleitung) betrieben werden, ein Multi-User-Betrieb ist ohne weiteres möglich. Der Nachteil ist, daß besonders bei niedrigen Übertragungsraten die Zeit zum Übertragen eines Bildschirminhaltes relativ groß ist. Ein Screen-Editing ist meist nur eingeschränkt möglich.
Für den Betrieb eines seriellen Terminals an einem Computer ist eine Anpassung erforderlich. Denn leider sind sich die Hersteller nicht einig geworden über das Übertragungsformat, die Übertragungsrate oder auch nur über die Funktion von Steuerzeichen. Alle Terminals haben deshalb die Möglichkeit, Übertragungsraten und Datenformate einzustellen. Keinen Einfluß hat der Anwender darauf, welche Steuerzeichen welche Funktion auslösen, beispielsweise die Cursorsteuerung. Aus diesem Grund empfehlen die meisten Hersteller gleich bestimmte Terminals, um Überraschungen von vornherein auszuschließen. Das mc-Terminal ermöglicht es, jeder Taste eine beliebige Funktion zuzuordnen.
Sowohl die Eingabe von der Tastatur als auch die Datenübertragung vom Rechner lösen beim Prozessor TMS9995 einen Interrupt aus. Um eine einwandfreie Datenübertragung zu gewährleisten, wird der Interrupt-Routine bei Datenempfang vom Rechner eine höhere Priorität als der Service-Routine für die Tastatureingabe zugeordnet. Die Interruptanforderung der Tastatur wird jedoch gespeichert und einige Mikrosekunden später abgearbeitet. Der Interrupt wird durch den Strobe der Tastatur ausgelöst.

Umcodierung per Tabelle

Nachdem das Zeichen von der Tastatur empfangen wurde, wird es gegebenenfalls umcodiert. Bei Standard-ASCII-Tastaturen ist keine Umcodierung erforderlich.
Für jeden der beiden möglichen Zeichensätze ist eine separate Umcodierungstabelle im Programm-EPROM vorhanden (hex 00 bis 7F). Dadurch ist es möglich, bei gleicher Tastatur zwei verschiedene Belegungen zu realisieren. Jeder Taste kann somit jedes darstellbare

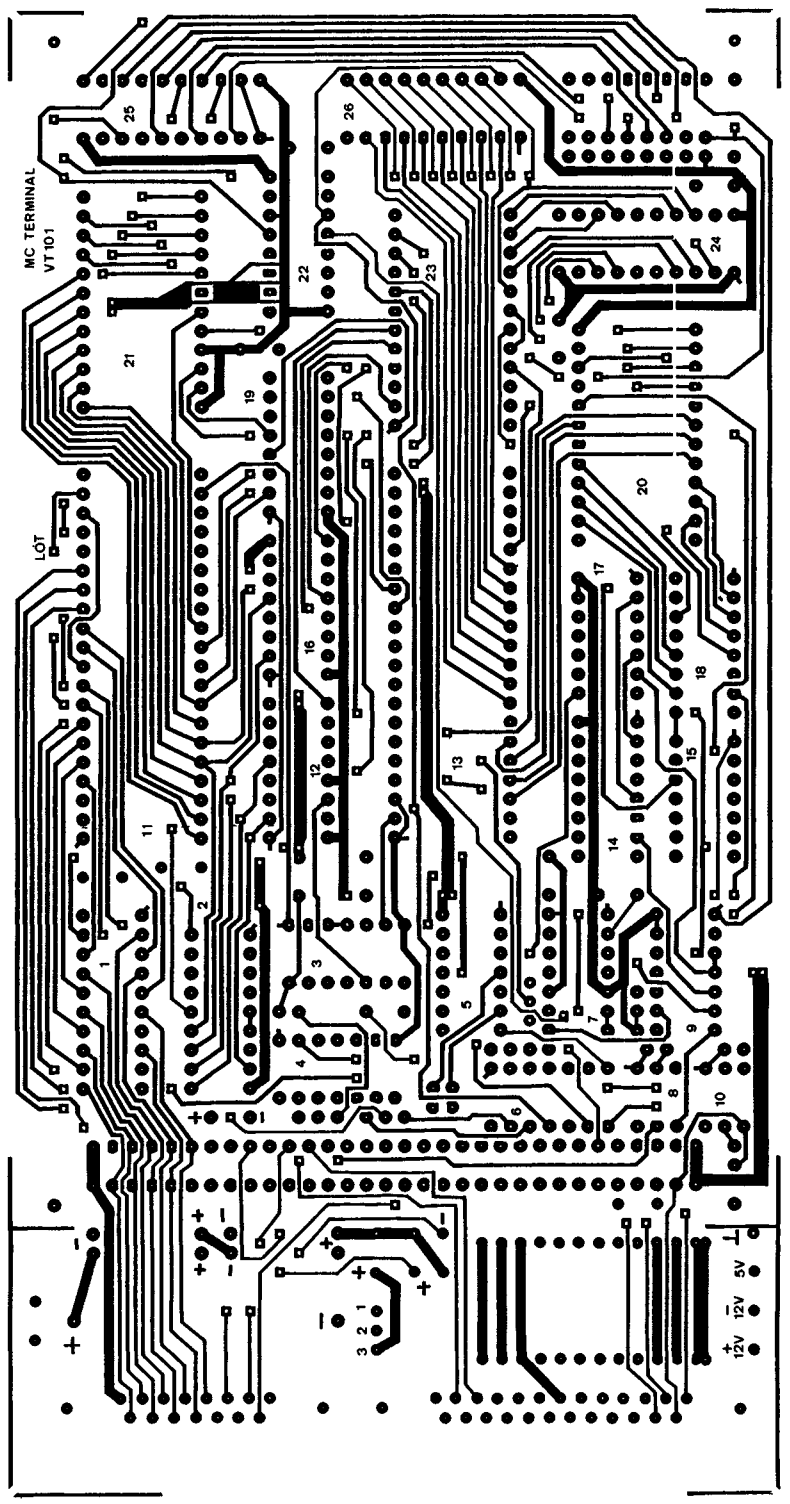


Bild 5. Leiterbahnen auf der Lötseite der Terminal-Platine

derzeichenbearbeitung das 16-Bit-Wort unter Adresse Tabellenbeginn + hex 0E (= 2 × hex 07) mit dem 16-Bit-Wort unter Tabellenbeginn + hex 18 (= 2 × hex 0C) getauscht werden.

Da die hier verwendete Cherry-Tastatur für die Funktionen „Cursor rechts" und „Cursor Home" etwas ungewöhnliche Steuerzeichen ausgibt, wird in der vorliegenden Software eine Umcodierung in den Tabellen 1 und 2 vorgenommen: Die Taste „Cursor rechts" wird zu Control L (hex 0C) und „Cursor Home" (hex 0F) wird zu Control ∧ (hex 1E). Die Tabellen beginnen jeweils bei den folgenden Adressen:

Tabelle 1: hex 0A00 (Zeichensatz 1)
Tabelle 2: hex 0900 (Steuerfunktionen)
Tabelle 3: hex 0A80 (Zeichensatz 2)

Diese Umcodiererei mag auf den ersten Blick verwirrend und unnötig erscheinen, wer sich aber schon über die Unterschiede zwischen Tastenaufschrift und Wirkung geärgert hat, wird diese Möglichkeit zu schätzen wissen.

Implementierte Sonderfunktionen

Control G (hex 07) = Bell
Control H (hex 08) = Backspace/ Cursor links
Control J (hex 0A) = Line Feed/Cursor nach unten
Control K (hex 0B) = Cursor hoch
Control L (hex 0C) = Cursor rechts
Control M (hex 0D) = Carriage Return
Control Z (hex 1A) = Clear Screen
Control ∧ (hex 1E) = Cursor Home
Control _ (hex 1F) = New Line

Alle weiteren Funktionen werden über Escape-Sequenzen angesprochen und sind mit einer Ausnahme sowohl im Local Modus mit der Tastatur als auch im Online-Modus vom Rechner aus zu bedienen. Die Escape-Sequenzen werden durch aufeinanderfolgende Betätigung der Taste ESC und den zugehörigen Tasten ausgelöst. Wird nach ESC kein gültiges Zeichen eingegeben, wird die begonnene Sequenz abgebrochen.

– Keyboard Disable: ESC #
Durch diesen Befehl wird die Tastatur vom Terminal getrennt, es werden keine Eingaben mehr angenommen.

– Keyboard Enable: ESC "
Diese Sequenz bewirkt, daß die vorher gesperrte Tastatur wieder freigegeben

```
0900                    *
0900                    * TABELLE INDEX FUER SPRUNG SONDERZEICHENBEARBEITUNG
0900                    *  (ZEICHEN VOM RECHNER)
0900                    *
0900 0246 TABEL  DATA NOFUNC       (0)                        Ctl $
0902 0246        DATA NOFUNC       (1)                        Ctl A
0904 0246        DATA NOFUNC       (2)                        Ctl B
0906 0246        DATA NOFUNC       (3)                        Ctl C
0908 0246        DATA NOFUNC       (4)                        Ctl D
090A 0246        DATA NOFUNC       (5)                        Ctl E
090C 0246        DATA NOFUNC       (6)                        Ctl F
090E 0718        DATA BELSE        (7) PIEPSER EIN            Ctl G
0910 04AC        DATA BSSERV       (8) BACKSPACE              Ctl H
0912 0246        DATA NOFUNC       (9)                        Ctl I
0914 0518        DATA LFSERV       (A) LINEFEED               Ctl J
0916 04EE        DATA CUHOSE       (B) CURSOR HOCH            Ctl K
0918 04CC        DATA CURENS       (C) CURSOR RECHTS          Ctl L
091A 046C        DATA CRSERV       (D) CARRIAGE RETURN        Ctl M
091C 0246        DATA NOFUNC       (E)                        Ctl N
091E 0246        DATA NOFUNC       (F)                        Ctl O
0920 0246        DATA NOFUNC       (10)                       Ctl P
0922 0246        DATA NOFUNC       (11)                       Ctl Q
0924 0246        DATA NOFUNC       (12)                       Ctl R
0926 0246        DATA NOFUNC       (13)                       Ctl S
0928 0246        DATA NOFUNC       (14)                       Ctl T
092A 0246        DATA NOFUNC       (15)                       Ctl U
092C 0246        DATA NOFUNC       (16)                       Ctl V
092E 0246        DATA NOFUNC       (17)                       Ctl W
0930 0246        DATA NOFUNC       (18)                       Ctl X
0932 0246        DATA NOFUNC       (19)                       Ctl Y
0934 05E2        DATA CLRSCS       (1A) CLEAR SCREEN          Ctl Z
0936 0246        DATA NOFUNC       (1B)                       Esc
0938 0246        DATA NOFUNC       (1C)                       Ctl Ä
093A 0246        DATA NOFUNC       (1D)                       Ctl Ü
093C 05F4        DATA HOME         (1E) HOME                  Ctl ^
093E 070C        DATA NEWLIN       (1F) NEW LINE              Ctl _
0940                    *
0940                    *
0940                    *  TABELLE ESCAPE-SEQUENZEN
0940                    *
0940                    *
0940 0246 ESCTAB DATA NOFUNC       ( )
0942 0246        DATA NOFUNC       (!)
0944 062E        DATA AKTTAS       (") TASTATUR AKTIVIEREN
0946 0634        DATA DISTAS       (#) TASTATUR SPERREN
0948 0246        DATA NOFUNC       ($)
094A 0246        DATA NOFUNC       (%)
094C 0246        DATA NOFUNC       (&)
094E 0246        DATA NOFUNC       (')
0950 0246        DATA NOFUNC       (()
0952 0246        DATA NOFUNC       ())
0954 0602        DATA CLRSCN       (*) CLEAR SCREEN (NULL)
0956 05E2        DATA CLRSCS       (+) CLEAR SCREEN (SPACE)
0958 0246        DATA NOFUNC       (,)
095A 0246        DATA NOFUNC       (-)
095C 0664        DATA CURDAR       (.) CURSORDARSTELLUNG
095E                    * ESC.0 = CURSOR NICHT DARSTELLEN
095E                    * ESC.1 = BLOCKCURSOR BLINKEND
095E                    * ESC.2 = BLOCKCURSOR STATISCH
095E                    * ESC.3 = UNTERSTREICHENDER CURSOR BLINKEND
095E                    * ESC.4 = UNTERSTREICHENDER CURSOR STATISCH
095E 0246        DATA NOFUNC       (/)
0960 0246        DATA NOFUNC       (0)
0962 0246        DATA NOFUNC       (1)
0964 0246        DATA NOFUNC       (2)
0966 0246        DATA NOFUNC       (3)
0968 0246        DATA NOFUNC       (4)
096A 0246        DATA NOFUNC       (5)
096C 02EE        DATA SENDLS       (6) SEND LINE
096E 0282        DATA SENDSC       (7) SEND SCREEN
0970 0246        DATA NOFUNC       (8)
0972 0246        DATA NOFUNC       (9)
0974 0246        DATA NOFUNC       (:)
0976 05E2        DATA CLRSCS       (;) CLEAR SCREEN (SPACE)
0978 0246        DATA NOFUNC       (<)
097A 0690        DATA POSCUR       (=) CURSOR POSITIONIEREN
097C 0246        DATA NOFUNC       (>)
097E 06DC        DATA ASKCUR       (?) CURSORPOSITION ABFRAGEN
0980 0246        DATA NOFUNC       (@)
0982 0246        DATA NOFUNC       (A)
0984 0246        DATA NOFUNC       (B)
0986 0246        DATA NOFUNC       (C)
0988 0246        DATA NOFUNC       (D)
098A 03D8        DATA INLINS       (E) INSERT LINE
098C 0246        DATA NOFUNC       (F)
098E 05BE        DATA ZEIDAR       (G) ZEICHENDARSTELLUNG
0990                    * ESC G0 = ZEICHEN HELL AUF DUNKEL
0990                    * ESC G4 = ZEICHEN DUNKEL AUF HELL
0990 07E2        DATA MOUT         (H) AUSGABE HILFSTABELLE
0992 0246        DATA NOFUNC       (I)
0994 0246        DATA NOFUNC       (J)
0996 0246        DATA NOFUNC       (K)
0998 0246        DATA NOFUNC       (L)
099A 0246        DATA NOFUNC       (M)
099C 0246        DATA NOFUNC       (N)
099E 0246        DATA NOFUNC       (O)
09A0 0246        DATA NOFUNC       (P)
09A2 035E        DATA INCHSE       (Q) CHARACTER INSERT
09A4 0434        DATA DELINS       (R) DELETE LINE
09A6 0246        DATA NOFUNC       (S)
09A8 0330        DATA CLRLIS       (T) ERASE TO END OF LINE (SPACE)
09AA 0616        DATA SETRMS       (U) TRANSPARENTMODE EIN
09AC 0246        DATA NOFUNC       (V)
09AE 03A8        DATA DECHSE       (W) DELETE CHARACTER
09B0 061C        DATA RETRMS       (X) TRANSPARENTMODE AUS
09B2 030C        DATA CENDSS       (Y) ERASE TO END OF SCREEN
09B4 0246        DATA NOFUNC       (Z)                    (SPACE)N
09B6 0246        DATA NOFUNC       ([)
09B8 0246        DATA NOFUNC       (\)
09BA 0246        DATA NOFUNC       (])
09BC 0246        DATA NOFUNC       (^)
09BE 0246        DATA NOFUNC       (_)
09C0 0246        DATA NOFUNC       ( )
09C2 0246        DATA NOFUNC       (a)
09C4 0246        DATA NOFUNC       (b)
09C6 0246        DATA NOFUNC       (c)
09C8 0246        DATA NOFUNC       (d)
09CA 0246        DATA NOFUNC       (e)
09CC 0246        DATA NOFUNC       (f)
09CE 0246        DATA NOFUNC       (g)
09D0 0246        DATA NOFUNC       (h)
09D2 0246        DATA NOFUNC       (i)
09D4 0246        DATA NOFUNC       (j)
09D6 0622        DATA SETLOC       (k) LOCAL MODE EIN
09D8 0628        DATA RESLOC       (l) LOCAL MODE AUS
09DA 0246        DATA NOFUNC       (m)
09DC 0246        DATA NOFUNC       (n)
09DE 0246        DATA NOFUNC       (o)
09E0 0246        DATA NOFUNC       (p)
09E2 0246        DATA NOFUNC       (q)
09E4 0246        DATA NOFUNC       (r)
09E6 0246        DATA NOFUNC       (s)
09E8 035A        DATA CLRLIN       (t) ERASE TO END OF LINE (NULL)
09EA 061C        DATA RETRMS       (u) TRANSPARENTMODE AUS
09EC 0246        DATA NOFUNC       (v)
09EE 0246        DATA NOFUNC       (w)
09F0 0246        DATA NOFUNC       (x)
09F2 032C        DATA CENDSN       (y) ERASE TO END OF SCREEN (NULL)
09F4 06F8        DATA NOFUNC       (z)
09F6
09F6
09F6 0246        DATA NOFUNC       (ä)
09F8 0246        DATA NOFUNC       (ö)
09FA 0246        DATA NOFUNC       (ü)
09FC 0246        DATA NOFUNC       (ß)
09FE 0246        DATA NOFUNC       ( )
0A00                    *
0A00                    *
0A00                    *  TABELLE ZEICHEN UMCODIEREN (ASCII)
0A00                    *
0A00                    *
0A00 0001 TABEL1 BYTE >00,>01,>02,>03,>04,>05,>06,>07
0A02 0203
0A04 0405
0A06 0607
0A08 080C        BYTE >08,>0C,>0A,>0B,>09,>0D,>0E,>1E
0A0A 0A0B
0A0C 090D
0A0E 0E1E
0A10 1011        BYTE >10,>11,>12,>13,>14,>15,>16,>17
0A12 1213
0A14 1415
0A16 1617
0A18 1819        BYTE >18,>19,>1A,>1B,>1C,>1D,>0F,>1F
0A1A 1A1B
0A1C 1C1D
0A1E 0F1F
0A20 2021        BYTE >20,>21,>22,>23,>24,>25,>26,>27
0A22 2223
0A24 2425
0A26 2627
0A28 2829        BYTE >28,>29,>2A,>2B,>2C,>2D,>2E,>2F
0A2A 2A2B
0A2C 2C2D
0A2E 2E2F
0A30 3031        BYTE >30,>31,>32,>33,>34,>35,>36,>37
0A32 3233
0A34 3435
0A36 3637
0A38 3839        BYTE >38,>39,>3A,>3B,>3C,>3D,>3E,>3F
0A3A 3A3B
0A3C 3C3D
0A3E 3E3F
0A40 4041        BYTE >40,>41,>42,>43,>44,>45,>46,>47
0A42 4243
0A44 4445
0A46 4647
0A48 4849        BYTE >48,>49,>4A,>4B,>4C,>4D,>4E,>4F
0A4A 4A4B
0A4C 4C4D
0A4E 4E4F
0A50 5051        BYTE >50,>51,>52,>53,>54,>55,>56,>57
0A52 5253
0A54 5455
0A56 5657
0A58 5859        BYTE >58,>59,>5A,>5B,>5C,>5D,>5E,>5F
0A5A 5A5B
0A5C 5C5D
0A5E 5E5F
0A60 6061        BYTE >60,>61,>62,>63,>64,>65,>66,>67
0A62 6263
0A64 6465
0A66 6667
0A68 6869        BYTE >68,>69,>6A,>6B,>6C,>6D,>6E,>6F
0A6A 6A6B
0A6C 6C6D
0A6E 6E6F
0A70 7071        BYTE >70,>71,>72,>73,>74,>75,>76,>77
0A72 7273
0A74 7475
0A76 7677
0A78 7879        BYTE >78,>79,>7A,>7B,>7C,>7D,>7E,>7F
0A7A 7A7B
0A7C 7C7D
0A7E 7E7F
```

Bild 6. In den Tabellen des Programmes steht, welche Funktion beim Drücken einer bestimmten Taste ausgelöst wird

```
0000  f0 24 00 4a f0 64 01 bc  ff ff ff ff ff ff ff ff
0010  f0 44 00 d2 6a 50 59 07  1e 02 18 1b 00 09 40 09
0020  00 00 00 00 20 40 00 48  08 a2 62 b2 72 82 52 83
0030  43 00 00 16 00 2b 00 57  00 ae 01 5b 04 ae 05 d9
0040  06 b6 f8 00 00 50 ff fe  0d 0a 03 00 00 00 02 0c
0050  00 80 1d 1f d0 60 70 00  0a 11 09 d1 32 21 00 29
0060  1e 0d d0 60 70 00 09 11  0a 51 09 c1 33 21 00 32
0070  d0 60 70 00 0a 71 09 f1  c8 01 f0 12 02 01 f0 00
0080  04 f1 02 81 f0 24 16 fc  02 01 80 80 c8 01 f0 08
0090  c8 01 50 00 04 c1 d8 01  20 00 06 c1 d8 21 00 14
00a0  28 00 05 81 06 c1 02 81  10 00 16 f5 d0 60 70 00
00b0  09 f1 c8 01 f0 0a 02 01  f0 a4 c8 01 f0 16 06 a0
00c0  08 5e 02 01 f0 a4 06 a0  08 c4 1d 12 03 00 00 04
00d0  10 fd 03 00 00 01 02 0c  00 80 02 09 02 68 c0 60
00e0  f0 18 16 51 02 08 02 5c  06 98 d0 60 f0 12 13 03
00f0  d0 26 0a 80 10 02 d0 26  0a 00 c1 80 06 c6 0a 16
0100  02 80 1b 00 16 06 07 20  f0 22 c0 60 f0 04 16 3b
0110  10 17 d0 60 f0 22 13 11  02 80 75 00 13 0b 02 80
0120  58 00 13 08 02 80 6b 00  16 08 06 a0 06 e0 04 e0
0130  f0 22 03 80 06 a0 06 da  10 fa d0 60 f0 04 16 02
0140  06 99 03 80 d0 60 f0 10  13 03 06 a0 07 6e 03 80
0150  d0 60 f0 1a 16 1f d0 60  f0 20 16 16 d0 60 f0 1c
0160  16 16 d0 60 f0 0e 16 19  d0 60 f0 22 16 19 d0 60
0170  f0 06 13 03 06 a0 07 e4  03 80 02 80 20 00 14 fa
0180  c0 26 09 00 06 90 03 80  06 a0 06 fe 10 4c 06 a0
0190  06 82 10 49 06 a0 07 28  10 46 06 a0 07 54 10 43
01a0  02 26 ff c0 02 88 02 5c  16 03 04 e0 f0 22 10 02
01b0  04 e0 f0 0c c0 66 09 40  06 91 10 35 02 0c 00 80
01c0  1e 12 04 c0 35 c0 02 0c  00 80 c1 80 06 c6 0a 16
01d0  02 09 07 e4 d0 60 f0 06  16 1e d0 60 f0 10 13 03
01e0  06 a0 07 6e 10 20 d0 60  f0 1a 16 d4 d0 60 f0 1c
01f0  16 ce d0 60 f0 0e 16 d1  d0 60 f0 20 16 c5 02 80
0200  1b 00 16 03 07 20 f0 0c  10 0e d0 60 f0 0c 16 c8
0210  d0 60 f0 06 13 02 06 99  10 06 02 80 1f ff 1b fb
0220  c0 66 09 00 06 91 02 a1  02 81 f0 84 13 15 03 00
0230  00 00 d0 60 f0 1e 13 04  02 01 f0 a4 06 a0 08 c4
0240  1f 15 16 09 c0 60 f0 16  35 f1 1e 12 c8 01 f0 16
0250  07 20 f0 1e 10 f1 1d 12  03 80 04 5b 04 c0 d0 20
0260  60 00 c1 80 06 c6 04 5b  1d 10 1f 16 16 fe 32 00
0270  1e 10 02 80 0d 00 16 06  c0 60 f0 0a 13 03 02 00
0280  0a 00 10 f2 04 5b c2 8b  c2 60 f0 02 02 29 40 00
0290  c1 89 02 26 07 80 02 05  f0 a4 c0 60 f0 16 cd 79
02a0  02 85 f0 f6 16 fc 06 45  06 05 04 cb d2 d5 0a 1b
02b0  09 1b 02 8b 20 00 16 04  02 85 f0 a4 16 f5 10 01
02c0  05 85 d5 60 00 48 02 05  f0 a4 d0 15 06 a0 02 68
02d0  1f 15 16 02 35 f1 1e 12  98 35 00 48 16 f6 06 49
02e0  81 89 1a d9 02 00 04 00  06 a0 02 68 07 20 f0 1e
02f0  04 5a c2 8b c0 20 f0 00  06 a0 05 42 c2 60 f0 00
0300  c8 00 f0 00 02 29 40 00  c1 89 02 26 00 50 10 c3
0310  02 09 20 20 c2 8b c1 60  f0 00 c1 a0 f0 02 02 26
0320  08 00 02 25 40 00 02 26  40 00 e2 60 f0 08 c0 60
0330  f0 16 c1 c6 61 c5 41 e0  00 46 13 01 dd 49 81 85
0340  14 06 cd 49 1f 15 16 fb  35 f1 1e 12 10 f8 c8 01
0350  f0 16 07 20 f0 1e 04 5a  04 c9 10 dc 02 09 20 20
0360  c2 8b c1 20 f0 00 06 a0  05 42 c1 a0 f0 00 c8 04
0370  f0 00 02 26 00 50 02 24  40 00 02 26 40 00 c1 44
0380  10 d4 04 c9 10 ed c2 8b  c0 60 f0 00 06 a0 05 42
0390  c0 a0 f0 00 02 22 40 4f  c1 42 06 05 c1 81 02 26
03a0  40 00 c0 e0 f0 16 d4 95  1f 15 16 02 35 f3 1e 12
03b0  06 05 06 02 81 82 16 f7  02 05 20 20 e1 60 f0 08
03c0  d5 85 c8 01 f0 00 c8 03  f0 16 07 20 f0 1e 04 5a
03d0  c0 60 f0 00 c2 8b 06 a0  05 42 c0 a0 f0 00 02 22
03e0  40 4f c1 81 02 26 40 00  c1 46 05 85 c0 e0 f0 16
03f0  dd b5 1f 15 16 02 35 f3  1e 12 81 82 16 f9 10 dc
0400  c2 8b 06 a0 05 42 c1 60  f0 00 02 25 40 00 c1 a0
0410  f0 02 02 26 47 2e c2 c5  c2 46 02 29 00 50 c0 60
0420  f0 16 1f 15 16 02 35 f1  1e 12 c6 56 06 46 06 49
0430  c6 56 06 46 06 49 c6 56  06 46 06 49 c6 56 06 46
0440  06 49 c6 56 06 46 06 49  c6 56 06 46 06 49 c6 56
0450  06 46 06 49 c6 56 06 46  06 49 c6 56 06 46 06 49
0460  c6 56 06 46 06 49 c6 56  06 46 06 49 c6 56 06 46
0470  06 49 c6 56 06 46 06 49  c6 56 06 46 06 49 c6 56
0480  06 46 06 49 c6 56 06 46  06 49 c6 56 06 46 06 49
0490  c6 56 06 46 06 49 c6 56  06 46 06 49 c6 56 06 46
04a0  06 49 81 46 14 be 02 2b  00 50 02 06 20 20 e1 a0
04b0  f0 08 cd 46 cd 46 cd 46  cd 46 cd 46 cd 46 cd 46
04c0  cd 46 cd 46 cd 46 1f 15  16 02 35 f1 1e 12 82 c5
04d0  1a f0 06 a0 08 38 c8 01  f0 16 07 20 f0 1e 04 5a
04e0  c2 8b 06 a0 05 42 c1 60  f0 00 02 25 40 00 c1 85
04f0  02 26 00 50 c2 e0 f0 02  02 2b 47 7f c0 60 f0 16
0500  cd 76 cd 76 cd 76 cd 76  cd 76 cd 76 cd 76 cd 76
0510  cd 76 cd 76 1f 15 16 02  35 f1 1e 12 82 c6 1a f0
0520  c1 60 f0 02 02 25 47 30  10 c0 c2 8b 06 a0 05 42
0530  d0 60 f0 04 13 03 d0 60  f0 0a 16 14 06 a0 08 14
0540  04 5a 68 20 f0 02 f0 00  c1 a0 f0 00 04 c5 3d 60
0550  00 44 04 c6 39 60 00 44  c8 06 f0 00 a8 20 f0 02
0560  f0 00 04 5b 06 a0 05 d6  10 eb c2 8b 88 20 f0 00
0570  f0 02 16 06 c0 a0 f0 00  02 22 07 80 c8 02 f0 00
0580  06 20 f0 00 06 a0 08 38  04 5a c2 8b c1 60 f0 00
0590  61 60 f0 02 02 85 07 7f  1a 04 c8 20 f0 02 f0 00
05a0  10 01 05 a0 f0 00 06 a0  08 38 04 5a c2 8b 68 20
05b0  f0 02 f0 00 c0 a0 f0 00  02 82 00 4f 1b 03 02 22
05c0  07 30 10 02 02 22 ff b0  a0 a0 f0 02 c8 02 f0 00
05d0  06 a0 08 38 04 5a c1 cb  d0 60 f0 04 16 3b c0 a0
05e0  f0 00 02 22 00 50 c1 20  f0 02 c1 42 61 44 02 85
05f0  07 80 1a 29 02 24 00 50  02 84 08 00 1a 04 41 20
0600  00 42 40 a0 00 42 c1 44  02 25 47 30 c1 85 02 26
0610  00 50 02 09 20 20 e2 60  f0 08 cd 49 cd 49 cd 49
0620  cd 49 cd 49 cd 49 cd 49  cd 49 cd 49 cd 49 cd 49
0630  cd 49 cd 49 cd 49 cd 49  cd 49 cd 49 cd 49 cd 49
0640  cd 49 81 85 11 ea c8 02  f0 00 c8 04 f0 02 06 a0
0650  08 14 04 57 68 20 f0 02  f0 00 c0 a0 f0 00 02 82
0660  07 2f 1b 03 02 22 00 50  10 02 02 22 f8 d0 a0 a0
0670  f0 02 c8 02 f0 00 06 a0  08 14 04 57 07 20 f0 1c
0680  04 5b 02 80 34 00 16 02  04 e0 f0 08 02 80 30 00
0690  16 04 02 00 80 80 c8 00  f0 08 04 e0 f0 1c 04 5b
06a0  c2 8b c8 20 f0 02 f0 00  06 a0 08 38 06 a0 08 5e
06b0  04 5a c2 8b c8 20 f0 02  f0 00 06 aQ 08 38 04 5a
06c0  c2 8b c8 20 f0 02 f0 00  06 a0 08 38 04 c9 06 a0
06d0  08 62 04 5a 07 20 f0 06  04 5b 04 e0 f0 06 04 5b
06e0  07 20 f0 04 04 5b 04 e0  f0 04 04 5b 04 e0 f0 18
06f0  04 5b 07 20 f0 18 04 5b  07 20 f0 20 04 5b 02 80
0700  30 00 13 08 02 80 31 00  16 09 c8 01 50 00 07 20
0710  f0 12 10 04 c8 01 60 00  04 e0 f0 12 04 e0 f0 20
0720  04 5b 07 20 f0 1a 04 5b  06 c0 02 20 ff d0 1a 0c
0730  02 80 00 04 15 09 c0 40  d0 61 00 24 02 02 0a 00
0740  d8 02 20 00 d8 01 28 00  04 e0 f0 1a 04 5b 07 20
0750  f0 0e 04 5b 06 c0 02 20  ff e0 02 80 00 17 12 02
0760  02 00 00 17 60 f0 0e 07  20 f0 10 04 5b c2 8b
0770  06 c0 02 20 ff e0 02 80  00 4f 12 02 02 00 00 4f
07B0  c0 a0 f0 0e 38 a0 00 44  a0 c0 a0 e0 f0 02 c8 03
0790  f0 00 06 a0 08 38 04 e0  f0 0e 04 e0 f0 10 04 5a
07a0  c2 8b c0 60 f0 00 04 c0  60 60 f0 02 3c 20 00 44
07b0  02 20 00 20 02 21 00 20  06 c0 06 a0 02 68 06 c1
07c0  c0 01 06 a0 02 68 02 00  0d 00 06 a0 02 68 04 5a
07d0  c0 cb 06 a0 05 2a 06 a0  05 d6 04 53 c8 20 30 00
07e0  30 00 04 5b c2 8b e0 20  f0 08 c0 a0 f0 00 d8 80
07f0  40 00 60 a0 f0 02 02 82  07 7e 15 05 05 a0 f0 00
0800  06 a0 08 14 04 5a c0 60  f0 04 13 fa c8 20 f0 02
0810  f0 00 10 f6 02 05 0c 00  d8 05 20 00 d8 20 f0 02
0820  28 00 06 e0 f0 02 02 25  01 00 d8 05 20 00 d8 20
0830  f0 02 28 00 06 e0 f0 02  02 05 0e 00 d8 05 20 00
0840  d8 20 f0 00 28 00 06 e0  f0 00 02 25 01 00 d8 05
0850  20 00 d8 20 f0 00 28 00  06 e0 f0 00 04 5b 02 09
0860  20 20 c1 60 f0 02 02 25  40 00 c1 85 02 26 08 00
0870  e2 60 f0 08 c0 60 f0 16  cd 49 cd 49 cd 49 cd 49
0880  cd 49 cd 49 cd 49 cd 49  cd 49 cd 49 cd 49 cd 49
0890  cd 49 cd 49 cd 49 cd 49  cd 49 cd 49 cd 49 cd 49
08a0  81 85 14 05 1f 15 16 e8  35 f1 1e 12 10 e5 c8 01
08b0  f0 16 07 20 f0 1e 04 5b  02 01 0b 00 02 02 0f 3d
08c0  c8 02 f0 16 1f 15 16 06  c1 a0 f0 16 35 f6 1e 12
08d0  c8 06 f0 16 88 01 f0 16  13 07 04 e0 f0 84 d8 31
08e0  f0 84 04 20 08 f6 10 ee  04 e0 f0 1e 02 01 f0 a4
08f0  c8 01 f0 16 04 5b f0 84  01 c6 ff ff ff ff ff ff
0900  02 5a 02 5a 02 5a 02 5a  02 5a 02 5a 02 5a 07 dc
0910  05 6a 02 5a 05 d6 05 ac  05 8a 05 2a 06 ec 06 f2
0920  02 5a 02 5a 02 5a 02 5a  02 5a 02 5a 02 5a 02 5a
0930  02 5a 02 5a 06 a0 02 5a  02 5a 02 5a 06 b2 07 d0
0940  02 5a 02 5a 06 ec 06 f2  02 5a 02 5a 02 5a 02 5a
0950  02 5a 02 5a 06 c0 06 a0  02 5a 02 5a 07 22 02 5a
0960  02 5a 02 5a 02 5a 02 5a  02 5a 02 5a 02 f2 02 86
0970  02 5a 02 5a 02 5a 06 a0  02 5a 07 4e 02 5a 07 a0
0980  02 5a 02 5a 02 5a 02 5a  02 5a 04 00 02 5a 06 7c
0990  08 b8 02 5a 02 5a 02 5a  02 5a 02 5a 02 5a 02 5a
09a0  02 5a 03 86 04 e0 02 5a  03 5c 06 d4 02 5a 03 d0
09b0  06 da 03 10 02 5a 02 5a  02 5a 02 5a 02 5a 02 5a
09c0  02 5a 02 5a 02 5a 02 5a  02 5a 02 5a 02 5a 02 5a
09d0  02 5a 02 5a 02 5a 06 e0  06 e6 02 5a 02 5a 02 5a
09e0  02 5a 02 5a 02 5a 02 5a  03 82 06 da 02 5a 02 5a
09f0  02 5a 03 58 06 f8 02 5a  02 5a 02 5a 02 5a 02 5a
0a00  00 01 02 03 04 05 06 07  08 09 0a 0b 0c 0d 0e 0f
0a10  10 11 12 13 14 15 16 17  18 19 1a 1b 1c 1d 1e 1f
0a20  20 21 22 23 24 25 26 27  28 29 2a 2b 2c 2d 2e 2f
0a30  30 31 32 33 34 35 36 37  38 39 3a 3b 3c 3d 3e 3f
0a40  40 41 42 43 44 45 46 47  48 49 4a 4b 4c 4d 4e 4f
0a50  50 51 52 53 54 55 56 57  58 59 5a 5b 5c 5d 5e 5f
0a60  60 61 62 63 64 65 66 67  68 69 6a 6b 6c 6d 6e 6f
0a70  70 71 72 73 74 75 76 77  78 79 7a 7b 7c 7d 7e 7f
0a80  00 01 02 03 04 05 06 07  08 09 0a 0b 0c 0d 0e 0f
0a90  10 11 12 13 14 15 16 17  18 19 1a 1b 1c 1d 1e 1f
0aa0  20 21 22 23 24 25 26 27  28 29 2a 2b 2c 2d 2e 2f
0ab0  30 31 32 33 34 35 36 37  38 39 3a 3b 3c 3d 3e 3f
0ac0  40 41 42 43 44 45 46 47  48 49 4a 4b 4c 4d 4e 4f
0ad0  50 51 52 53 54 55 56 57  58 59 5a 5b 5c 5d 5e 5f
0ae0  60 61 62 63 64 65 66 67  68 69 6a 6b 6c 6d 6e 6f
0af0  70 71 72 73 74 75 76 77  78 79 7a 7b 7c 7d 7e 7f
0b00  1a 52 45 56 2e 34 2e 35  1b 3d 21 33 1b 47 34 20
0b10  53 20 50 20 45 20 43 20  49 20 41 20 4c 20 20 20
0b20  46 20 55 20 4e 20 43 20  54 20 49 20 4f 20 4e 20
0b30  53 20 1b 3d 23 20 1b 47  30 43 54 4c 20 47 20 3a
0b40  20 42 45 4c 4c 1b 3d 23  48 43 54 4c 20 48 20 3a
0b50  20 42 41 43 4b 53 50 41  43 45 0d 0a 43 54 4c 20
0b60  4a 20 3a 20 4c 49 4e 45  46 45 45 44 1b 3d 24 48
0b70  43 54 4c 20 4b 20 3a 20  43 55 52 53 4f 52 20 55
0b80  50 0d 0a 43 54 4c 20 4c  20 3a 20 43 55 52 53 4f
0b90  52 20 52 49 47 48 54 1b  3d 25 48 43 54 4c 20 4d
0ba0  20 3a 20 43 41 52 52 49  41 47 45 20 52 45 54 55
0bb0  52 4e 0d 0a 43 54 4c 20  5a 20 3a 20 43 4c 45 41
0bc0  52 20 53 43 52 45 45 4e  1b 3d 26 48 43 54 4c 20
0bd0  5e 20 3a 20 43 55 52 53  4f 52 20 48 4f 4d 45 0d
0be0  0a 43 54 4c 20 5f 20 3a  20 4e 45 57 20 4c 49 4e
0bf0  45 0d 0a 45 53 43 20 22  20 3a 20 4b 45 59 42 4f
0c00  41 52 44 20 45 4e 41 42  4c 45 1b 3d 28 48 45 53
0c10  43 20 23 20 3a 20 4b 45  59 42 4f 41 52 44 20 44
0c20  49 53 41 42 4c 45 0d 0a  45 53 43 20 2a 20 3a 20
0c30  43 4c 45 41 52 20 53 43  52 45 45 4e 20 54 4f 20
0c40  4e 55 4c 4c 1b 3d 29 48  45 53 43 20 2b 20 3a 20
0c50  43 4c 45 41 52 20 53 43  52 45 45 4e 20 54 4f 20
0c60  53 50 41 43 45 0d 0a 45  53 43 20 2e 30 3a 20 43
0c70  55 52 53 4f 52 20 44 49  53 41 42 4c 45 1b 3d 2a
0c80  48 45 53 43 20 2e 31 3a  20 42 4c 49 4e 4b 49 4e
0c90  47 20 42 4c 4f 43 4b 20  43 55 52 53 4f 52 0d 0a
0ca0  45 53 43 20 2e 32 3a 20  53 54 45 41 44 59 20 42
0cb0  4c 4f 43 4b 20 43 55 52  53 4f 52 1b 3d 2b 48 45
0cc0  53 43 20 2e 33 3a 20 42  4c 49 4e 4b 49 4e 47 20
0cd0  55 4e 44 45 52 4c 49 4e  45 0d 0a 45 53 43 20 2e
0ce0  34 3a 20 53 54 45 41 44  59 20 55 4e 44 45 52 4c
0cf0  49 4e 45 1b 3d 2c 48 45  53 43 20 36 20 3a 20 53
0d00  45 4e 44 20 4c 49 4e 45  0d 0a 45 53 43 20 4c 49
0d10  3a 20 53 45 4e 44 20 53  43 52 45 45 4e 1b 3d 2d
0d20  48 45 53 43 20 3b 20 3a  20 43 4c 45 41 52 20 53
0d30  43 52 45 45 4e 20 54 4f  20 53 50 41 43 45 0d 0a
0d40  45 53 43 3d 52 2c 43 3a  20 57 52 49 54 45 20 43
0d50  55 52 53 2e 20 50 4f 53  49 54 49 4f 4e 1b 3d 2e
0d60  48 45 53 43 20 3f 20 3a  20 52 45 41 44 20 43 55
0d70  52 53 4f 52 20 50 4f 53  49 54 49 4f 4e 0d 0a 45
0d80  53 43 20 45 20 3a 20 49  4e 53 45 52 54 20 4c 49
0d90  4e 45 1b 3d 2f 48 45 53  43 20 47 30 3a 20 4e 4f
0da0  52 4d 41 4c 20 56 49 44  45 4f 0d 0a 45 53 43 20
0db0  47 34 3a 20 49 4e 56 45  52 53 45 20 56 49 44 45
0dc0  4f 1b 3d 30 48 45 53 43  20 51 20 3a 20 49 4e 53
0dd0  45 52 54 20 43 48 41 52  41 43 54 45 52 0d 0a 45
0de0  53 43 20 52 20 3a 20 44  45 4c 45 54 45 20 4c 49
0df0  4e 45 1b 3d 31 48 45 53  43 20 54 20 3a 20 45 52
0e00  41 53 45 20 54 4f 20 45  4e 44 20 4f 46 20 4c 49
0e10  4e 45 20 28 53 50 41 43  45 29 0d 0a 45 53 43 20
0e20  55 20 3a 20 4d 4f 4e 49  54 4f 52 20 4d 4f 44 45
0e30  20 4f 4e 1b 3d 32 48 45  53 43 20 57 20 3a 20 44
0e40  45 4c 45 54 45 20 43 48  41 52 41 43 54 45 52 0d
0e50  0a 45 53 43 20 58 20 3a  20 4d 4f 4e 49 54 4f 52
0e60  20 4d 4f 44 45 20 4f 46  46 1b 3d 33 48 45 53 43
0e70  20 59 20 3a 20 45 52 41  53 45 20 54 4f 20 45 4e
0e80  44 20 4f 46 20 53 43 52  45 45 4e 20 28 53 50 41
0e90  43 45 29 0d 0a 45 53 43  20 6b 20 3a 20 4c 4f 43
0ea0  41 4c 20 45 44 49 54 1b  3d 34 48 45 53 43 20 6c
0eb0  20 3a 20 44 55 50 4c 45  58 20 45 44 49 54 0d 0a
0ec0  45 53 43 20 74 20 3a 20  45 52 41 53 45 20 54 4f
0ed0  20 45 4e 44 20 4f 46 20  4c 49 4e 45 20 28 4e 55
0ee0  4c 4c 29 1b 3d 35 48 45  53 43 20 75 20 3a 20 4d
0ef0  4f 4e 49 54 4f 52 20 4d  4f 44 45 20 4f 46 46 0d
0f00  0a 45 53 43 20 79 20 3a  20 45 52 41 53 45 20 54
0f10  4f 20 45 4e 44 20 4f 46  20 53 43 52 45 45 4e 20
0f20  28 4e 55 4c 4c 29 1b 3d  36 48 45 53 43 20 48 20
0f30  3a 20 4d 45 53 53 41 47  45 20 4f 55 54 00 20 08
0f40  20 08 20 08 20 08 20 08  ff ff ff ff ff ff ff ff
0f50  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
0f60  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
0f70  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
0f80  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
0f90  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
0fa0  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
0fb0  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
0fc0  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
0fd0  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
0fe0  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff
0ff0  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff 04 05
Fuer die Cherry-Tastatur:
0a89    0c    statt   09
0a09    0c    statt   09
0a8c    09    statt   0c
0a0c    09    statt   0c
0a8f    1e    statt   0f
0a0f    1e    statt   0f
0a9e    0f    statt   1e
0a1e    0f    statt   1e
```

Bild 8. Das vollständige Programm des mc-Terminals als Hex-Dump

Grundsätzlich lassen sich alle Zeichen auf folgende Weise finden: Die ersten acht Zeilen sind zu finden ab ASCII-Wert × 8 aufwärts bis (ASCII-Wert × 8) + 8. Die zweiten acht Zeilen findet man analog unter den Adressen (ASCII-Wert × 8) + hex 400 aufwärts. Von diesen Zeilen werden aber nur die ersten beiden abgebildet. In hex 00 bis FF sind die Zeichen abgespeichert, die im Transparent Mode für die Darstellung der Steuerzeichen benutzt werden. Es sind dies die Zeichen 0 bis o, die zur Unterscheidung von den normalen Zeichen dann unterstrichen sind (die Unterstreichung selbst ist im Zeichengenerator ab hex 400 abgelegt, siehe oben).

Änderung der Betriebsart

Soll die Betriebsart des Terminals, beispielsweise Übertragungsformat oder -rate, geändert werden, so muß nach Umschalten des DIL-Schalters ein Reset durchgeführt werden, da die Schalterstellung nur beim Reset abgefragt werden. Spätere Änderungen der Schalterstellungen werden nicht erkannt.

Bessere Störaustastung

Bei einer Revision der Platine wurde eine verbesserte Störaustastung vorgesehen. Dazu wird der Takt des Zwischenspeichers IC24 während der Zeit unterbrochen, in der die CPU auf das Video-RAM zugreift. Die Schaltung ist in *Bild 11* dargestellt. Der Widerstand R18 wird auf der Lötseite der Platine montiert. Die Leiterbahn zum Anschluß 11 des Zwischenspeichers muß dabei aufgetrennt werden (unter R16).

Bezugsquellen für die Platine, Teilbausätze und programmierte EPROMs sind: H.-J. Regge, Fesenfeld 57, 2800 Bremen, sowie r+r electronic, Adlerstr. 55, 6500 Heidelberg.
Die Halbleiter sind auch bei Heninger, Landwehrstr. 39, 8000 München 2, erhältlich.

Im ersten Fall wird die Datei NAME.TXT von der Betriebsdiskette gelesen, im zweiten Fall die Datei BEISPIEL.MAC von der Diskette in Laufwerk B. Wird der Dateityp weggelassen, muß der Dateiname unbedingt mit einem Punkt enden, andernfalls erfolgt die Fehlermeldung SYNTAX ERROR und der Miniassembler verlangt eine neue Eingabe. Ein fehlender Dateityp wird durch den Typ MAC ergänzt. Die Dateibezeichnung muß nicht in Großbuchstaben eingegeben werden, der Miniassembler versteht auch Kleinschrift. Existiert die verlangte Datei nicht, erscheint die Meldung NO INPUT FILE auf dem Bildschirm und der Assembler fordert eine erneute Eingabe der Dateibezeichnung.

Ein kleines Beispiel

In *Bild 1* ist Sourcecode zu sehen, wie er vom Miniassembler akzeptiert wird. Das kleine Programm gibt auf dem Bildschirm zwei Textzeilen aus und übergibt dann wieder die Kontrolle an das Betriebssystem. In *Bild 2* ist der Assemblierlauf protokolliert. Nach der Eingabe der Dateibezeichnung – die Datei heißt in diesem Fall BILD3,MAC – meldet sich der Miniassembler erneut mit „**FIRST PASS**". Jetzt wird die erste Aanalyse des Sourcecodes durchgeführt. Bei längeren Quelltexten gibt der Miniassembler in der folgenden Zeile nach jeder 100sten Zeile ein Sternchen aus. Ist der erste Teil fehlerfrei überstanden, folgt der zweite Durchlauf, in dem das Opcode-File auf die Betriebsdiskette geschrieben und das Assemblerlisting ausgegeben wird. In *Bild 3* schließlich ist ein Hexdump des erzeugten Opcodes zu sehen.

Zum Assemblerlisting sind noch einige Anmerkungen erforderlich: Der Miniassembler zählt die Zeilen, die er sukzessive verarbeitet, die Zeilennummer steht in der ersten Spalte. In der zweiten Spalte ist die Opcode-Adresse zu finden. Dabei handelt es sich in der Regel um die Adresse des ersten Codes, der in dieser Zeile steht, jedoch gibt es drei Ausnahmen: Die Direktiven BYTE, WORD und die Textdarstellung. In diesen Fällen zeigt die aufgelistete Adresse auf die erste Speicherstelle nach dem Opcode, der der jeweiligen Zeile entspricht. In der folgenden Spalte wird der Opcode aufgelistet – wobei die drei eben genannten Ausnahmen wieder eine Sonderstellung einnehmen: Da Texte ebenso wie Byte- oder Wortlisten oft länger sind als die vier maximal zur Verfügung stehenden Bytes, wird hier kein Opcode ausgegeben. In der nächsten Spalte folgen die Labels, falls vorhanden, dann kommt der Opcode samt Kommentaren.

Direktiven

Könnte der Miniassembler nur Mnemonics übersetzen, wäre es zum Beispiel kaum möglich eine Tabelle oder Text in den Opcode einzuarbeiten. Auch für einige andere Zwecke ist es sinnvoll, dem Miniassembler während der Übersetzung des Sourcecodes diverse Anweisungen geben zu können. Solche Anweisungen nennt man Pseudo-Befehle oder Assembler-Direktiven. Der Miniassembler kennt zum Beispiel die Direktive BYTE. Taucht dieses Wort im Sourcecode auf, dann werden die darauf folgenden Ausdrücke als Bytes interpretiert und in den Opcode eingefügt. Beispiele sind die Zeilen 18 und 19 in Bild 2. Der Miniassembler akzeptiert dabei alle zulässigen Formen der Zahlendarstellung. So können auch mathematische Ausdrücke verwendet werden:

```
** INPUT-FILENAME? B:BILD3.MAC
** FIRST PASS **

Z80 ASM PROG 2.1 (C) BY B. WIEDEMANN

0001 0100                          ;*** Beispielprogramm
0002 0100
0003 0100                          ORIG #100
0004 0100
0005 0100           :CP/M-JUMP:= 05
0006 0100           :Print-Befehl:= #9
0007 0100
0008 0100  #110E01                 LD DE,:Ausgabetext:
0009 0103  #0E09                   LD C,:Print-Befehl:
0010 0105  #CD0500                 CALL :CP/M-JUMP:
0011 0108  #C30000                 JP 0
0012 010B
0013 010B           :Ausgabete.t:
0014 010C                          BYTE 012 ;Form Feed / Schirm loschen
0015 0111                          "Text:"
0016 0113                          WORD #0D0A
0017 013B                          "Dieser Text erscheint auf dem Bildschirm"
0018 013D                          BYTE #D,#A
0019 013E                          BYTE '$'
0020 013E                          ;Ende des Programmes "Bild 3"
```

Bild 2. Das Assemblerlisting des Programmes von Bild 1

BYTE :LABEL: +#20,#40+08
Nach Byte können mehrere Werte oder Ausdrücke stehen, die dann durch je ein Komma getrennt sein müssen. Ähnliches gilt für die Direktive WORD. Der Miniassembler bewertet die folgenden Ausdrücke oder Zahlen als 16-Bit-Worte. Sie werden in den Opcode eingefügt, wobei das niederwertige Byte zuerst im Speicher abgelegt wird. Ein Vergleich der Bilder 2 und 3 macht dies deutlich: In der Zeile 16 (Bild 2) soll das Wort #0D0A in den Opcode eingefügt werden. Betrachtet man die Speicherstellen #0111 und #0112 in Bild 3, fällt auf, daß im Opcode zunächst #0A, dann erst #0D steht. Diese Umkehrung der Reihenfolge wurde wegen der Eigenart der 80er CPUs gewählt, Adressen und Inhalte der 16-Bit-Register in eben dieser Weise im Speicher zu erwarten. Auch nach WORD können mehrere Ausdrücke oder Zahlen durch Kommata getrennt verwendet werden.

Assembler-Direktiven sind im Grunde genommen auch die „Gänsefüßchen". Die Zeilen 15 und 17 in Bild 2 sind Beispiele dafür. Der Miniassembler fügt in den Opcode die ASCII-Äquivalente des Textes ein, der von den Anführungszeichen eingeschlossen wird. Enthält eine Zeile nur Text, kann des zweite Anführungszeichen entfallen. In die gleiche Kategorie fällt das Gleichheitszeichen nach einem Label. Es sagt dem Miniassembler, daß dem Label der folgende Wert zugewiesen werden soll. Dabei kann es sich wiederum um eine Zahl oder einen mathematischen Ausdruck handeln. Beispiele finden sich in den Zeilen 5 und 6 von Bild 2.

Eine weitere wichtige Direktive ist ORIG. Die Zahl nach dieser Direktive wird in den internen Programmzähler des Assemblers geladen. Der Programmzähler gibt an, an welcher Speicherstelle der Opcode letztlich stehen soll. Er ist für die Berechnung von Sprungadressen bei der Verwendung von Labels und bei

der Berechnung relativer Sprungdistanzen erforderlich. Beim Starten des Miniassemblers ist der Programmzähler auf #100 eingestellt. In Bild 2 wurde der Programmzähler nur demonstrationshalber geladen. Da das Programm bei #100 beginnt, hätte die Zeile 3 ebensogut entfallen können. Der Programmzähler ist über das Dollarzeichen erreichbar und kann somit im Sourcecode als Wert verwendet werden.

Zahlendarstellung

Zahlen können im Sourcecode in verschiedener Weise geschrieben werden. Zum einem ist es möglich, Dezimalzahlen zu verwenden. Diese müssen immer mit einer führenden Null beginnen. Den Hexadezimalzahlen muß ein # vorangestellt werden. Schließlich können die ASCII-Werte von Zeichen verarbeitet werden. Dazu setzt man das entsprechende ASCII-Zeichen zwischen zwei Apostrophe, wie dies in Bild 2, Zeile 19 geschehen ist. Allerdings gibt es von dieser Regel eine Ausnahme: Das Komma kann auf diese Weise *nicht* verwendet werden. Hier muß statt dessen der ASCII-Wert direkt angegeben werden. Im übrigen können zur Zahlendarstellung auch Labels verwendet werden. Davon wird später noch ausführlich die Rede sein.
Statt einzelner Zahlen versteht der Miniassembler auch einfache mathematische Ausdrücke. Zulässig sind innerhalb eines solchen Ausdruckes die mathematischen Operatoren + und −. Multiplikationen und Divisionen sowie Klammern sind nicht zulässig. Innerhalb eines Ausdruckes können dagegen alle Zahlendarstellungen gleichzeitig verwendet und miteinander verknüpft werden (siehe auch *Bild 4*).
Weiterhin können Zahlen stellvertretend durch Labels ausgedrückt werden. Einem Label wird einmal innerhalb des Sourcecodes ein Wert zugewiesen, den er dann behält. Eine erneute Wertzuweisung ist nicht mehr möglich. Labels erfordern eine besondere Schreibweise: Sie müssen mit einem Doppelpunkt beginnen und mit einem Doppelpunkt enden. Dazwischen können bis zu 13 Zeichen stehen. Im Bedarfsfall verarbeitet der Miniassembler auch längere Labels, doch werden diese unter Umständen in der Labelspalte des Assemblerlistings nicht vollständig ausgedruckt. Der Miniassembler arbeitet intern trotzdem mit dem vollständigen Label. Die Anzahl der signifikanten Stellen ist dabei unbegrenzt.

Sprungzielberechnungen

Die Verwendung von Labels besitzt zwei große Vorteile: So ist es nicht mehr erforderlich, Sprungziele oder relative Sprungdistanzen zu berechnen, zum anderen können Zahlen durch einen anschaulichen Namen ersetzt werden, der es gestattet, selbstdokumentierende Programme zu schreiben. Die Zeilen 5 und 6 in Bild 2 bieten ein Beispiel dafür: Das Label :CP/M-JUMP: wird hier auf den Wert 5 festgelegt. Immer wenn im Operandenteil eines Mnemonics – natürlich auch innerhalb eines mathematischen Ausdruckes – dieses Label auftaucht, repräsentiert es die Zahl 5. Dies ist in Bild 2, Zeile 10 der Fall. Bei der Wertzuweisung an ein Label können auch weitere Labels verwendet werden. Einzige Voraussetzung: Die Labels, die rechts vom Gleichheitzeichen stehen, müssen bereits zuvor definiert worden sein:

```
:LABEL1:=#5
:LABEL2:=:LABEL1:+0100
```

Würde man die beiden Zeilen vertauschen, käme es zu einer Fehlermeldung. Bei der Verwendung eines Labels im Operandenteil eines Mnemonics gilt diese Einschränkung nicht, das Label kann in diesem Fall auch später definiert werden, wie dies bei Sprungzielangaben ja oft erforderlich ist.

Prinzipiell können Labels an zwei Stellen einer Zeile auftauchen: Entweder gleich zu Beginn der Zeile oder im Operandenteil eines Mnemonics bzw. einer Assemblerdirektive. Im ersten Fall weist der Miniassembler dem Label einen Wert zu, wie wir es bereits am Beispiel der Label-Definition gesehen haben. Folgt dem Label kein Gleichheitzeichen, dann erhält es den Wert, den der Programmzähler an dieser Stelle besitzt. Dabei spielt es keine Rolle, ob das Label alleine in einer Zeile steht, oder ob ihm noch ein Kommentar, eine Direktive oder ein Mnemonic folgt. Labels, die im Operandenfeld auftauchen, repräsentieren dort den Wert, der ihnen an anderer Stelle des Sourcecodes zugewiesen wurde. Da es sich um einen Zwei-Pass-Assembler handelt, ist es nicht erforderlich, daß das Label bereits vorher definiert wurde. Es kann ebensogut einige Zeilen weiter oder am Schluß des Programmes seinen Wert erhalten. Die einzige Ausnahme von dieser Regel wurde bereits bei der Beschreibung der Labeldifinition erwähnt.

Wenn ein direkter Sprung oder ein Unterprogrammaufruf programmiert wird, kann das Sprungziel natürlich direkt angegeben werden. Dies Möglichkeit wird man jedoch bestenfalls dann anwenden, wenn das Sprungziel außerhalb des eigentlichen Programmes liegt. Sprünge und Unterprogrammaufrufe innerhalb des Programmes sollten über ein Label erfolgen. Dabei schreibt man das Label an den Anfang der Zeile, die das Ziel des Sprunges ist oder mit der die Subroutine beginnt. Wie oben bereits gesagt, erhält das Label den Wert des Programmzählers an dieser Stelle. Im Operandenteil des Sprungbefehls bzw. Unterprogrammaufrufes setzt dann der Miniassembler diesen Wert in den Opcode ein.

```
     0 1  2 3  4 5  6 7  8 9  A B  C D  E F
0100 110B 010E 09CD 0500 C300 000C 5465 7874 ▯.....▯▯.C...Text
0110 3A0A 0D44 6965 7365 7220 5465 7874 2065 ▯:..Dieser Text e
0120 7273 6368 6569 6E74 2061 7566 2064 656D ▯rscheint auf dem
0130 2042 696C 6473 6368 6972 600D 0A24 0000 ▯ B:ldschirm..$..
```

Bild 3. Der Hexdump des nun assemblierten Programmes

Das gleiche gilt für relative Sprünge, wie sie beim Z80 möglich sind. Dabei kann das Sprungziel wieder in Form eines Labels angegeben werden. Der Miniassembler berechnet die relative Distanz zum Sprungziel und verwendet diesen Wert im Opcode. Die Angabe der relativen Distanz im Sourcecode ist nicht möglich. Dies wäre auch sehr unpraktisch: Bei einer Änderung des Programmes zwischen relativem Sprung und Sprungziel müßte der Programmieren die Distanz „per Hand" neu berechnen

```
C0>>prt 1 99
:  muß vor und nach einem Label stehen
"  damit beginnen und enden Zeichenketten
;  damit beginnen Kommentare
0  damit beginnen Dezimalzahlen
#  damit beginnen Hexzahlen
'  schließt Zeichenoperanden ein (z.B.: LD A,'B'
   bedeutet, daß der Akku mit $42 geladen wird)
,  trennt zwei Operanden, wie in (1) definiert
=  Wertzuweisung für Labels
+  Pluszeichen in Ausdrücken
-  Minuszeichen in Ausrücken
$  Programmzähler
```

Bild 4. Für den Miniassembler haben verschiedene Zeichen eine bestimmte Bedeutung

und eintragen. Bei der Angabe des Sprungzieles nimmt ihm der Miniassembler diese Arbeit ab. Allerdings ist es indirekt möglich, eine relative Distanz anzugeben: Dabei verwendet man den Programmzähler:
JR$+09
Der Miniassembler nimmt als Sprungziel die Adresse an, die 9 Byte oberhalb des derzeitigen Programmzählerstandes liegt. Dabei ist zu berücksichtigen, daß der Programmzähler in diesem Fall noch auf den Beginn des Sprungbefehles zeigt, während beim relativen Sprung die Distanz ab dem Byte nach dem Sprungbefehl berechnet wird. Übersetzt wird der Befehl somit in die Bytefolge #18, #07. Die Angabe der Distanz nach dem Programmzähler muß also immer um zwei größer sein als die relative Distanz im Opcode.
Ein weiteres Beispiel für die Verwendung von Labels findet sich in den Zeilen 8 und 13 von Bild 2. Das Label :Ausgabetext: wird in Zeile 13 definiert, indem es den Wert des Programmzählers erhält. An dieser Stelle des Programmes beginnt der Text, der über das Terminal ausgegeben werden soll. In Zeile 8 wird der Wert des Labels in das Register DE geladen, in dem CP/M bei der Ausführung der Routine „Print String" die Anfangsadresse des Stringes erwartet.

Fehlermeldungen

Einen längeren Sourcecode auf Anhieb ohne syntaktische Fehler zu schreiben, ist fast unmöglich. Der Assembler erkennt Syntaxfehler und einige logische Fehler und gibt während der beiden Durchläufe Fehlermeldungen aus. Diese Meldungen bestehen aus der Bezeichnung des Fehlers, der Zeilennummer und dem Zeileninhalt. Folgende Fehlermeldungen existieren:

UNKNOWN OPERATOR
Der Operator ist dem Assembler nicht bekannt. Entweder ist er falsch geschrieben oder es wurde bei einer Kommentarzeile das Semicolon, bei einem Text das Anführungszeichen oder bei einem Label der erste Doppelpunkt vergessen. In all diesen Fällen wertet der Miniassembler die ersten Zeichen innerhalb einer Zeile als Operatoren, die ihm jedoch nicht bekannt sind.
OPERAND ERROR
Im Operandenteil eines Mnemonics oder einer Direktive wurde ein Syntaxfehler entdeckt. Dabei kann es sich auch um einen falsch aufgebauten mathematischen Ausdruck oder um eine nicht richtig geschriebene Zahl handeln.
REDEFINED LABEL
Es wurde versucht, einem bereits definierten Label ein zweites Mal einen Wert zuzuweisen.
UNDEFINED LABEL
Im Operandenfeld wurde ein bisher nicht definiertes Label gefunden. Dabei kann es natürlich auch sein, daß dieses Label lediglich falsch geschrieben wurde.
INCORRECT LABEL
Der zweite Doppelpunkt wurde vergessen.
REL.DIST.TOO FAR
Das Sprungziel ist für einen relativen Sprung zu weit entfernt.
VALUE> 1BYTE
Dieser Fehler tritt auf, wenn ein 8-Bit-Register geladen werden soll, der aus dem Operandenteil berechnete Wert jedoch größer ist als 8 Bit.
NOT ENOUGH RAM
Der RAM-Bereich reicht nicht aus, um alle Labels in einer Tabelle unterzubringen.
Ein Teil der Fehlermeldungen tritt während des ersten Durchlaufes auf, ein anderer Teil erst im zweiten Durchlauf. Dabei stehen die Fehlermeldungen zwischen den Zeilen des Assemblerlistings. Wurde irgendwo ein Fehler entdeckt, schließt der Miniassembler auf jeden Fall seine Arbeit mit der Meldung ab:
INCORRECT SOURCECODE – NO OBJECT CODE GENERATED
Auf der Diskette ist dann kein entsprechendes COM-File zu finden. Wo der Fehler genau zu suchen ist, muß den vorangegangenen Fehlermeldungen entnommen werden. Bleibt nur noch zu wünschen, daß der Programmierer diese Meldung nur sehr selten zu Gesicht bekommt...
Der Miniassembler ist beim Franzis-Software-Service (Tel 089/5117-331) auf Diskette erhältlich.

Literatur

[1] *Feichtinger, Herwig:* Ist Assemblerprogrammierung veraltet? mc 1982, Heft 8, S. 22.
[2] MK 3880 CPU Technical Manual, Mostek
[3] *Klein, Rolf-Dieter:* Mikrocomputer Hard- und Softwarepraxis. Franzis-Verlag, München.

Rolf-Dieter Klein:

BIOS

In den letzten Abschnitten wurde in der Beschreibung des mc-CP/M-Computers schon oft das Betriebssystem CP/M angesprochen. Hier soll etwas ausführlicher darauf eingegangen werden. CP/M wurde von der Firma Digital Research eingeführt und fand im Laufe der Jahre eine sehr große Verbreitung.

Zu Beginn der Mikrocomputerei, als es schon einige fertige Geräte gab, hatte jeder Hersteller sein eigenes Betriebssystem gebaut und den Rechner damit angeboten. Das Betriebssystem war meist Firmengeheimnis. Über die Funktionen im System oder den Aufbau wurde nichts bekanntgegeben. Eine Anpassung des Systems an einen anderen Rechner oder gleichen Rechner mit anderen Erweiterungen war nicht möglich. (Solche Systeme gibt es auch heute noch.) Mit CP/M kam allerdings ein Betriebssystem als eigenständiges Software-Produkt auf den Markt, das daher eine Beschreibung über die Verbindung von Hardware und Betriebssystem enthalten mußte. Nun war es möglich, dazu passende Rechner zu bauen.
Viele Firmen haben in der Folge CP/M-fähige Rechner entwickelt. Dadurch fand das Betriebssystem sehr schnell große Verbreitung. Aufgrund dieser hohen Verbreitung gibt es auch sehr viele Programme, die mit CP/M verträglich sind. 1982 erlebte CP/M einen großen Boom, da nun auch große Rechnerhersteller mit CP/M-fähigen Computern auf den Markt kamen. So ist es heute üblich, auch bei 16-Bit-Rechnern, z. B. durch einen zweiten Prozessor, CP/M-2.2-Kompatibilität einzubauen.
CP/M wurde zunächst für den 8080-Prozessor angelegt. Daher gibt es die meiste Software auch für diesen Prozessor. Da der leistungsfähigere Z80 aufwärtskompatibel zum 8080 ist, hat sich CP/M-Software, die den Z80-Befehlsatz voll einsetzt, inzwischen weit verbreitet. Es empfiehlt sich daher, heute ein Z80-System zu verwenden, um auch neueste Software starten zu können. Neben CP/M für den 8080 bzw. Z-80, auch CP/M-80 genannt, wurde jüngst CP/M-86 entwickelt, das einen 8086 als Prozessor benötigt. Für diese CP/M gibt es ebenfalls schon eine Reihe von Software, jedoch ist CP/M-86 natürlich noch nicht sehr verbreitet. Neuerdings gibt es sogar CP/M 68 K für den 68000.

Was ist CP/M?

CP/M besteht aus einer Reihe von logisch trennbaren Teilen, dem BIOS, dem BDOS und dem CCP. Was diese Teile leisten, welche Aufgaben sie haben, wird nach und nach dargestellt. Es soll als erstes das BIOS, das Basic Input Output System, vorgestellt werden (und die Überlegungen, die zu diesem Programmteil geführt haben könnten).
Ein Mikrocomputer besteht gewissermaßen aus zwei Teilen, nämlich aus Rechner-Hardware und aus Software. Der Softwareteil hat die Aufgabe, die jeweilige Problemstellung mit der Hardware lösbar zu machen. Bei einem kleinen Rechnersystem besteht der System-Softwareteil zum Beispiel oft nur aus einem Monitor-Programm, mit dessen Hilfe Programme im „Hex-Code" (sedezimal) eingegeben werden können. Dieses Steuerprogramm ermöglicht es auch, Speicherbereiche anzuschauen und eingegebene Programme zu starten. Die Eingabe kann über eine Tastatur (alphanumerisch oder Hex-Tastatur) erfolgen. Die Ausgabe geschieht über einen Bildschirm oder eine Siebensegmentanzeige. Oft gehören zum Monitor auch Routinen, um Daten und Programme auf einem Kassettenrecorder zu speichern und auch wieder einzulesen. Mit dieser Anordnung lassen sich dann schon kleinste Programmentwicklungen durchführen.
Die dabei erzeugten Programme sind speziell auf die vorhandene Hardware abgestimmt, manchmal verwenden sie auch Unterprogramme des jeweiligen Monitors. Solche Programme sind weder portabel noch gut zu warten. Ändert sich zum Beispiel einmal der Monitor, so laufen meist die Programme nicht mehr. Für größere Programme ist das Verfahren also nicht praktikabel.
Um Programme zu entwickeln, die nicht nur auf einen Computer abgestimmt sind, ist es zunächst einmal nötig, definierte und festgelegte Schnittstellen zu schaffen. Während einfache Systeme Unterprogramme im Monitor direkt verwenden, indem direkt an eine Stelle des Monitors gesprungen wird, ist es bei wirklich komfortablen Systemen günstiger, alle nützlichen Unterprogramme in Form einer Liste von Sprüngen am Anfang des Monitors zusammenzufassen. Diese Liste, lauter Befehle von der Form „JMP Adresse", heißt dann Sprungtabelle (Vector Table). Nun kann der Benutzer des Systems Unterprogramme wie Ausgabe-Routinen und Einlese-Programme verwenden, sofern sie in der Sprungtabelle aufgeführt werden. Ändert sich jetzt die Version eines Monitors, was sich meist durch neu hinzugefügte Routinen bemerkbar macht, so können dennoch alte Programme verwendet werden, da die Sprungtabelle nur so geändert wird, daß auch alte Programme ihren Einsprung noch finden. Wenn zum Beispiel neue Einträge hinzukommen, dann werden sie immer am Ende der Tabelle angehängt. Man spricht hier von Aufwärtskompatibilität, da alte Programme, die nur den unteren Teil der Sprungtabelle verwenden, mit der erweiterten Sprungtabelle immer noch laufen, umgekehrt neuere Programme unter einem alten Monitor aber nicht.

Die Hardware-Voraussetzungen

Nun bleibt die Frage, welche Art von Unterprogrammen in eine solche Liste aufgenommen werden sollen. Das hängt natürlich vom Umfeld und von der Aufgabe des Systems ab. Bei der Konzeption von CP/M gab es dazu klare Vorstellungen. Es sollte ein universelles System geschaffen werden, bei dem diese Schnittstelle (Sprungtabelle) relativ ein-

fach zu verwirklichen sein, aber auch den verschiedensten anspruchsvollen Aufgaben gerecht werden sollte. Dabei wurde natürlich von der Verwendung von Siebensegmentanzeigen und Hex-Tastaturen Abstand genommen.
Die Hardware eines CP/M-Rechners muß heute also dazu ein paar allgemeine Voraussetzungen liefern. Zum Beispiel muß ein Datensichtgerät vorhanden sein. Außerdem muß der Rechner die Möglichkeit besitzen, Daten langfristig auf einen Massenspeicher abzulegen, um Programme nach Abschalten des Rechners auch nach Tagen wieder zur Verfügung zu haben – ohne sie neu eintippen zu müssen.
Dafür gibt es verschiedene Möglichkeiten, jedoch wurde bei CP/M vor allem an die Verwendung eines Floppy-Laufwerks gedacht. Plattenlaufwerke für hohe Kapazitäten (10 MByte) sollten aber genauso anschließbar sein. Da die Hardware sehr verschieden konstruiert sein kann, mußte eine möglichst allgemeine Form der Software gefunden werden. Zum Beispiel gibt es bei Datensichtgeräten sehr unterschiedliche Systeme. Im Prinzip ist aber eines allen gemein: Es sollen Zeichen über eine Tastatur eingegeben und zum Rechner transportiert werden können. Bei der Ausgabe vom Rechner zum Datensichtgerät sollen ebenfalls Zeichen transportiert werden können. Das gleiche gilt übrigens auch für einen eventuell anzuschließenden Drucker.

Zeichen ein – Zeichen aus

Also kann man sich auf eine Schnittstelle einigen, die einzelne Zeichen übergibt. Aufgabe einer solchen Schnittstelle ist es also, ein Zeichen, das in einem Register des Prozessors übergeben wird, an die Außenwelt abzugeben und umgekehrt ein Zeichen, das von der Außenwelt ankommt, bei Bedarf abholen zu lassen. Damit entstehen ein paar zu beachtende Einschränkungen. Bei CP/M mußte ein Zeichensatz, also eine bestimmte Code-Darstellung, aus den vielen möglichen ausgewählt werden. Es wurde die ASCII-Darstellung dazu gewählt (ISO-7-Bit-Code nach DIN 66003). Ein EBDIC-Terminal verwendet eine andere Codierung und muß also vor dem Anschluß an einen CP/M-Rechner zuerst mit einer Anpassung versehen werden (zum Beispiel durch eine Umcodiertabelle).
Eine weitere Bedingung, die CP/M insgeheim stellt, ist, daß alle ASCII-Zeichen, also auch die Steuerzeichen, erzeugbar, also auf dem verwendeten Terminal eingebbar sein mussen. Da das Paritätsbit von CP/M immer auf 0 gesetzt wird, sind dies 128 verschiedene Zeichen. Bei Einschränkungen kann es passieren, daß ein Zeichen, das von einem CP/M-Anwenderprogramm verlangt wird, nicht erzeugbar ist.
Eine weitere Bedingung ist, daß bei einer Eingabe in den Rechner immer nur ein Zeichen nach dem anderen, gewissermaßen in Tippgeschwindigkeit, anfallen sollte. Also sind zum Beispiel Terminals mit Blockmode, bei denen eine ganze „Seite" übertragen wird, mit CP/M nicht verträglich. Das Zeichen zu einer Taste, die betätigt wurde, wird nicht sogleich auf dem Bildschirm des Terminals ausgegeben. Dort erscheinen nur die Zeichen, die vom Rechner kommen. Damit ein eingegebener Text mitgelesen werden kann, wird durch die Einlese-Routinen des CP/M ein sogenanntes Echo erzeugt, das ein eingegebenes Zeichen sogleich wieder ausgibt.

Die Ausgabe auf eine Konsole wird wieder in ASCII durchgeführt. Von CP/M gibt es darüber hinaus nur sehr vage Grundforderungen an das Terminal. Im Prinzip muß, wie gesagt, das Terminal zunächst einmal alle Zeichen und Buchstaben des ASCII-Satzes darstellen können. Kleinbuchstaben müssen aber nicht unbedingt bereitgehalten werden. Es genügt auch allein der Satz der Versalien (Teletype-Besitzer freuen sich hierbei immer). Als Steuerzeichen werden zunächst von CP/M nur der Wagenrücklauf, auch Carriage Return, kurz CR genannt, und der Zeilenvorschub, Line Feed, auch LF genannt, benötigt. Dann kann damit bereits gearbeitet werden. Ein Bildschirmeditor wird aber meist doch mehr verlangen.

Da CP/M-Software immer hardwareunabhängig sein sollte, ist es bei den meisten Programmen möglich, diese auf verschiedene Terminals einzustellen. Mindestens sollten aber bei einem Datensichtgerät noch folgende Funktionen als Grundausstattung vorhanden sein:

- Ein Zeichen zurück, BS genannt (Back Space);
- Löschen des Bildschirms (manchmal CLS genannt);
- Positionieren auf die linke obere Ecke (HOME);
- Cursor nach rechts ohne Löschen des darunterliegenden Zeichens.

Damit kann eine Anpassung an die meisten Programme vorgenommen werden. Cursor-Direkt-Steuerung zum Beispiel könnte mit den obigen Funktionen umschrieben werden. Soweit der Abstecher bei der Zeichenschnittstelle. Hier nun in Kurzform die Funktion aller Schnittstellen zur Außenwelt (Kommunikationsteil):

CONST: Status der Eingabeschnittstelle. Wurde ein Zeichen eingegeben, so wird im Register A der Wert 0FFH übergeben, sonst 0. Das Zeichen wird aber noch nicht eingelesen.

CONIN: Das nächste Zeichen wird in das Register A eingelesen. Die Routine wartet so lange, bis ein Zeichen eingegeben wurde.

CONOUT: Das Zeichen, das auszugeben ist, befindet sich in Register C.

LIST: Die Schnittstelle führt auf den Drucker. Das Zeichen ist in Register C zu übergeben. Ist kein Drucker vorhanden, so kann die Ausgabe auch auf die Konsole umgesteuert werden.

LISTST: Eine Routine, die erst später hinzugekommen ist. Die Routine übergibt im Akkumulator den Wert FF, wenn der Drucker (Schnittstelle LIST) bereit ist, ein neues Zeichen zu empfangen, sonst 0. Damit ist es Programmen wie DESPOOL möglich, Ausgaben auf den Drucker zu leiten, während der Bediener quasi gleichzeitig z. B. im Editor arbeitet.

PUNCH: Register C enthält ein Zeichen zur Ausgabe, z. B. auf einen Lochstreifenstanzer.

READER: Eingabe von einem Lese-Gerät, z. B. Lochstreifen, in den Akkumulator. Die letzten beiden Routinen werden praktisch von keiner Software mehr benötigt. READER und PUNCH können zur Kopplung zweier Computer verwendet werden.

Floppies und CP/M

Nun fehlt noch die Schnittstelle zum Massen-Speicher. Eine Speichereinheit wie ein Floppy-Laufwerk überträgt seine

Daten üblicherweise in Blöcken. Ein solcher Speicher besitzt auch eine interne Adressierung, über die ein solcher Block angesprochen werden kann. Bei Floppy-Laufwerken gibt es darüber hinaus die Laufwerknummer. Mit ihr wird das Floppy-Laufwerk selbst aus mehreren ausgewählt. Die Laufwerknummer heißt auch DRIVE-Nummer. Innerhalb einer Floppy-Einheit unterscheidet man zwischen Sektornummern und Tracknummern.

Damit hat es folgende Bewandtnis: Ein Floppy-Laufwerk verwendet eine flexible Magnetplatte als Datenträger. Diese Magnetplatte ist in konzentrische Kreise eingeteilt, die Tracks (Spuren) genannt werden. Ein solcher Track ist dann nochmals in einzelne Abschnitte aufgeteilt, die Sektoren genannt werden. Die Anzahl der Sektoren, wie auch die Anzahl der Tracks sind je nach Typ eines Laufwerks (Minifloppy oder Standard-Floppy) unterschiedlich groß. Ebenfalls hängt dies von der Art der Aufzeichnung und vom gewählten Format ab.

Man unterscheidet FM- und MFM-Aufzeichnungen. FM ist auch unter dem Namen „Single Density" oder „einfache Schreibdichte" bekannt, MFM unter dem Namen „Double Dense" oder „doppelte Schreibdichte". Bei der MFM-Aufzeichnung wird die erhöhte Schreibdichte aber nicht etwa nur durch einfache Verdoppelung der Aufzeichnungsfrequenz, sondern durch Änderung der Bedeutung einzelner Bits im „Magnetisierungsmuster" erreicht. Bei FM folgt auf ein Taktbit immer ein Datenbit. Bei MFM wird unter bestimmten Voraussetzungen der Takt weggelassen, und daher haben mehr Daten Platz. Bei einer Hard-Disk, die eine sehr hohe Kapazität besitzt, ist dies im Prinzip ähnlich.

Man sieht, daß es sehr schwierig ist, den unterschiedlichen Systemen gerecht zu werden. Daher ist erst seit Erscheinen der Version CP/M 2.2 soviel Allgemeinheit gegeben, daß es möglich ist, alle bekannten Floppies und Hard-Disks anzupassen. Es gibt aber eine Einschränkung des Adreßbereiches innerhalb eines Laufwerks. Er beträgt 8 MByte. Eine Harddisk besitzt meist höhere Kapazität. Um sie unter CP/M ausnützen zu können, wird ein physikalisches Laufwerk dann in mehrere logisch adressierte Laufwerke aufgeteilt. Beispiel: Das Hard-Disk-Laufwerk besitzt 20 MByte. Dann wird es in drei Drives aufgespalten, die jeweils 6 MByte Speicherraum besitzen. Die restlichen 2 MByte können noch auf ein viertes Laufwerk gebracht werden, denn unter CP/M können bis zu 16 Drives adressiert werden. Die verschiedenen Speichersysteme verwenden die unterschiedlichsten Blockgrößen. CP/M verlangt an der Schnittstelle 128 Bytes pro Block. Das heißt, eine Übertragungseinheit pro Adresse (Drive, Track, Sektor) muß 128 Byte groß sein. Kennt der verwendete Floppy-Controller nur ein anderes Format, so ist im BIOS die Anpassung per Software auf die 128 Bytes zu erledigen.

Wie die Übertragungslogik arbeitet

Zur Peripherie werden also drei Adreßgrößen übertragen: Laufwerk, Spur und Sektor. Damit ist auf dem Speichermedium eindeutig ein Block mit einer Größe von 128 Bytes bestimmt. Nun muß noch die Information gegeben werden, ob dieser Block vom Laufwerk in das CP/M-System hinein übertragen werden soll, oder umgekehrt aus CP/M auf das Laufwerk ausgegeben werden soll. Dazu gibt es zwei Einsprünge, READ und WRITE, in CP/M. Ferner muß natürlich noch zuvor intern eine Adresse angegeben werden, woher der Block aus dem Haupt-Speicher kommt oder wohin er, bei einem Lesezugriff von der Peripherie, gebracht werden soll.

Da die Laufwerke unterschiedlich viel Sektoren pro Track besitzen können, wie auch die Anzahl der Sektoren unterschiedlich ist, müssen diese Daten dem CP/M zur Anpassung mitgeteilt werden. Dazu wird in BIOS eine Tabelle angelegt, die alle wichtigen Daten eines verwendeten Laufwerks enthält. Ebenfalls möglich ist es, unterschiedliche Laufwerke gleichzeitig zu verwenden, da für jedes Laufwerk eine solche Tabelle im CP/M bereitgehalten wird. Die Adresse der Tabelle wird dabei bei Aufruf eines der Voreinstellungs-Unterprogramme (Drive einstellen) an das CP/M übertragen, weshalb die Lage der Tabelle keine große Rolle spielt. In der Tabelle werden die Kapazität des Laufwerks festgelegt, der erste und letzte Sektor eines Tracks sowie die Anzahl der Tracks. Ferner werden noch Parameter definiert, die für das CP/M-Betriebssystem interessant sind, wie zum Beispiel Lage und Umfang eines Inhaltsverzeichnisses auf dem Speichermedium.

Ein anderer wichtiger Punkt bei der allgemeinen Betrachtung von Floppy-Laufwerken ist die Geschwindigkeitsoptimierung.

Wir wissen bereits, daß eine Spur einer Floppy in einzelne Sektoren eingeteilt ist. Ein Sektor kann z. B. aus 128 Bytes (optimal für CP/M) bestehen. Nun soll ein Programm eingelesen werden. Dazu werden im allgemeinen mehrere Sektoren benötigt. Das Programm sei z. B. 1024 Bytes lang, belegt also 8 Sektoren ($8 \times 128 = 1024$). Wird das Programm abgespeichert, so soll das z. B. einmal in aufeinanderfolgenden Sektoren geschehen. Das Laufwerk habe 18 Sektoren (Minifloppy z. B.) pro Spur.
Wenn der erste Sektor frei war, so werden die Sektoren 1, 2, 3, 4, 5, 6, 7, 8 belegt. Nun ist aber CP/M ein nicht ganz kleines Betriebssystem (ca. 6 KByte). Es gibt ja auch allerhand zu berechnen und zu prüfen. CP/M ist also im Normalfall nicht in der Lage, in einem Zuge alle Sektoren hintereinander einzulesen oder zu beschreiben, da zwischen dem Einlesen eines Blockes und des nächsten eine gewisse Zeit zum Rechnen benötigt wird. Da eine Diskette nur relativ langsam rotiert, muß zum Beispiel zwischen dem Einlesen zweier Sektoren eine volle Umdrehung gewartet werden, wenn die Zeit nicht ausreicht, sie direkt nacheinander zu lesen. Die „Gesamtladezeit" geht also drastisch herauf.

Trennung „logisch" – „physikalisch"

Um diesen Nachteil zu vermeiden, kann ein Trick angewendet werden: Die Sektoren werden nicht mehr nacheinander beschrieben oder gelesen, sondern es werden andere dazwischen geschoben.

Beispiel:
Alte Anordnung:
1 2 3 4 5 6 7 8 9 10 11 12
13 14 15 16 17 18

Neue Anordnung:
1 10 6 15 2 11 7 16 3 12 8
17 4 13 9 18 5 14

Nun hat sich der Abstand zwischen den Sektoren erhöht, und damit kann innerhalb einer Umdrehung mehr als nur ein Sektor gelesen werden, wenn die Rechenzeit kleiner als der Zeit-Abstand zwischen den jetzt logisch aufeinander folgenden Sektoren ist. Es gibt nun mehrere Möglichkeiten, eine solche Anordnung zu erreichen. Zum einen könnte mit der verwendeten Floppy durch spezielle Formatierung eine solche Numerierung erreicht werden. In CP/M aber gibt es im BIOS eine Übersetzungstabelle, die jedem ankommenden logischen Sektor des CP/M-Systems einen physika-

lischen Sektor zuweist, wie ihn das Floppy-System versteht.

Eine solche Tabelle könnte wie folgt aussehen:

1 5 9 13 17 3 7 11 15 2 6
10 14 18 4 8 12 16

Nehmen wir einmal an, wir wollten zunächst Sektor 1 schreiben oder lesen (logischer Sektor). Dann nehmen wir den ersten Eintrag unserer Tabelle und erhalten ebenfalls den Sektor 1 als physikalischen Sektor. Nun wollen wir den nächsten logischen Sektor schreiben oder lesen, also Sektor 2. In der Tabelle steht an der zweiten Stelle der Wert 5. Wir greifen damit also in Wirklichkeit auf den Sektor 5 (selbe Spur) auf der Diskette zu.

Zwischen Sektor 1 und Sektor 5 liegt der Abstand 3 Sektoren, es bleibt also Rechenzeit zwischen den Sektoren 1 und 2 (logische Sektoren), da wir in Wirklichkeit die Sektoren 1 und 5 verwendet haben. Mit den anderen Sektoren verhält sich dies ganz analog. Diese Tabelle wird auch Sektorsprungtabelle genannt. Der Abstand der Sektoren ist der sogenannte Interleaving-Faktor.

In unserem Beispiel war es der Wert 4. Aus dem Wert 4 läßt sich die Tabelle eindeutig aufbauen. Dazu wird bei Sektor 1 begonnen. Dann wird 4 addiert, und es ergibt sich als zweiter Eingang der Wert 5; dann nochmals, und es ergibt sich 9, dann 13, dann 17 – und was nun? 21 gibt es nicht, also minus 18 rechnen, damit ergibt sich der Wert 3.

Jetzt geht es weiter mit 7, 11, 15 und dann 19, 19–18 ergibt 1, aber den Sektor 1 gab es schon in der Tabelle. Nun wird nach dem nächsten nicht in der Tabelle schon vorhandenem Wert gesucht, und es ergibt sich der Sektor 2. Dann wird wieder fortgefahren, bis schließlich alle Sektor-Zuordnungen ermittelt sind. Zur Konstruktion dieser Tabelle (wie auch der anderen Tabellen) gibt es zu CP/M einen Satz von Makros für den Assembler MAC. Diese Programmstücke führen die Berechnung der Tabelleneinträge selbst durch. Für unsere Tabelle reicht dann die Angabe des Interleaving-Faktors.

Das BDOS

„BDOS" heißt der Programmteil von CP/M, der für alle unter CP/M laufenden Programme den Kontakt zum Computer und seiner Peripherie herstellt. BDOS kommt von Basic Disk-Operation System.

Das BDOS besitzt genau einen Einsprung, der in Adresse 5 des Hauptspeichers steht. Parameter, die in Registern oder Speicherteilen an BDOS übertragen werden, geben an, was BDOS tun soll.

Beim Kaltstart wird die Adresse 5 von BIOS aus mit dem Sprungbefehl belegt.

Alle CP/M-Programme können also über diesen Sprung an das BDOS gelangen, und zwar unabhängig von der eigentlichen Lage des BDOS. Leider gibt es auch Ausnahmen. Manche Systeme verwenden eine andere Adresse für den Einsprung, die oberhalb von 4000H liegt.

Solche Systeme sind allerdings eine Minderheit. Es gibt daher auch nicht allzu viele Programme dafür. Auf dem mc-CP/M-Computer lassen sich im Prinzip beide Systeme fahren. Jedoch sei von der Verwendung des 4000H-Systems abgeraten, da ja auch der Adreßraum für Anwenderprogramme sehr stark eingeschränkt ist (16 KBytes weniger).

Auf der Adresse 0 befindet sich ebenfalls ein Sprung, der auf den Warm-Boot-Teil des BIOS zeigt. Wird dieser angesprungen, so wird das CP/M-Betriebssystem neu geladen und gestartet.

Die Aufgabe von BDOS

Das BDOS hat die Aufgabe, die Dateiverwaltung vorzunehmen. Was versteht man aber darunter? Beim BIOS konnte man nur über Sektor- und Spur-Nummer (= Adresse) auf einen Sektor der Diskette zugreifen. Wenn zum Beispiel das Programm „Basic-Interpreter" 8 KByte umfaßt, ein Programm „Pascal" 32 KByte, und beide Programme auf der Diskette festgehalten werden sollen, dann muß man zuerst jeweils die benötigte Anzahl von Sektoren ermitteln. Bei 128 Bytes pro Sektor wären das beim Basic-Interpreter 64 Sektoren und beim Programm Pascal 256. Ein Track habe nun zum Beispiel 26 Sektoren und es gebe insgesamt 77 Spuren. Der Adreßbereich der Sektoren liegt dann zwischen 1 und 26 und bei den Tracks bei 0 bis 76. Dann wäre eine Zuweisung wie folgt möglich:

Basic Interpreter:	Start	Track 0	Sektor 1
	Ende	Track 2	Sektor 13
Pascal:	Start	Track 2	Sektor 14
	Ende	Track 12	Sektor 11

Man müßte dann eine Liste führen, auf der genau diese Tabelle steht. Daß das ganze sehr mühsam und fehlerträchtig ist, kann man sich denken.

Besser ist es, wenn der Computer selbst in der Lage ist, eine solche Liste zu führen. Die Liste nennt man dann Directory oder Inhaltsverzeichnis oder auch Datei-Katalog.

Wir haben hier also eine Möglichkeit gesehen, ein Direktory aufzubauen. Dazu wird einmal der Name der Datei notiert und zum anderen der Start (Track, Sektor) und das Ende (Track, Sektor). Und so wird mit jeder Datei verfahren.

Dieses Verfahren wird oft genauso angewendet. Doch es hat auch Nachteile. Was passiert zum Beispiel, wenn man eine Datei mitten aus einer Gruppe von weiteren Dateien herauslöschen will? Es bleibt ein Loch übrig. Beispiel:

Vor dem Löschen:

Name	Start Track	Start Sektor	Ende Track	Ende Sektor
Datei 1	1	1	4	5
Datei 2	4	6	10	3
Datei 3	10	4	23	16
Datei 4	23	17	40	2

Datei 2 soll beispielsweise gelöscht werden. Dann sieht das Inhaltsverzeichnis danach wie folgt aus:

Name	Start Track	Start Sektor	Ende Track	Ende Sektor
Datei 1	1	1	4	5
Datei 3	10	4	23	16
Datei 4	23	17	40	2

Der Bereich Track 4, Sektor 6, bis Track 10, Sektor 3, ist leer. Eine neue Datei kann nur dann diesen Platz belegen, wenn sie genausoviele oder weniger Blöcke benötigt. Um den restlichen Platz wieder uneingeschränkt verfügbar zu machen, muß ein Kompressionsvorgang gestartet werden. Die Datei 3 wird mit ihrem Beginn an die Stelle Track 4, Sektor 6 geschafft, und jeder der Blöcke von Datei 3 entsprechend nach unten verschoben. Dann muß Datei 4 nach unten verschoben werden usw. Dies ist ein sehr langwieriger und sogar gefährlicher Prozeß. Denn wird er versehentlich unterbrochen, so ist die betroffene Datei zerstört.

CP/M macht es anders

CP/M geht einen anderen Weg. In CP/M wird die gesamte Diskette in grobe Blökke unterteilt. Ein solcher Block ist normalerweise 1024 Bytes groß. Er enthält dann 8 Sektoren, wenn ein Sektor 128 Bytes umfaßt. Die Größe von 128 Bytes wird bei CP/M auch als Record bezeichnet. Es ist die kleinste adressierbare Einheit.

Die Blockgröße ist bei CP/M im BIOS definierbar und wird bei großen Hard-Disk-Systemen auch auf 8 KByte oder 16 KByte gesetzt. Jeder dieser Blöcke bekommt nun in CP/M eine Nummer. Diese Nummer bestimmt den Block eindeutig und ist mit der Adresse aus Track, Sektor-Nummer vergleichbar, jedoch gröber. Nun bekommt eine Datei bei Neuanlage eine Reihe solcher Nummern zugewiesen. Es werden dabei aber nicht Anfang- und Ende-Nummer festgehalten, sondern alle Nummern der Blöcke, über die die Datei sich erstreckt. Dazu wieder ein Beispiel. Es sollen drei Dateien abgelegt werden: BASIC 8 KByte, ASM 5 KByte und ED 4 KByte. Der erste freie Block sei der mit der Nummer 2, da das Inhaltsverzeichnis selbst natürlich auch Platz auf der Diskette belegt. Es ergibt sich folgendes Bild:

BASIC	2, 3, 4, 5, 6, 7, 8, 9
ASM	10, 11, 12, 13, 14
ED	15, 16, 17, 18

Die Blocknummern werden mit im Inhaltsverzeichnis abgespeichert. In CP/M belegt ein Inhaltsverzeichniseintrag deshalb 32 Byte. In Standard-8-Zoll-CP/M können 64 Einträge abgelegt werden (durch BIOS einstellbar).

Wenn jetzt die Datei ASM gelöscht werden soll, dann bleibt danach folgendes Inhaltsverzeichnis übrig:

BASIC	2, 3, 4, 5, 6, 7, 8, 9
ED	15, 16, 17, 18

Nun soll eine neue Datei mit dem Namen USER abgelegt werden, die 10 KByte groß ist. Es werden jetzt erst die Blökke 10, 11, 12, 13, 14 verwendet und dazu einfach die Blöcke 19, 20, 21, 22, 23.

Damit ergibt sich im Inhaltsverzeichnis:

BASIC	2, 3, 4, 5, 6, 7, 8, 9
USER	10, 11, 12, 13, 14, 19, 20, 21, 22, 23
ED	15, 16, 17, 18

Bei dieser Verwaltungstechnik bleiben also keine Blöcke ungenutzt. Nur, nach einer gewissen Zeit sind die Blöcke einer Datei über die Diskette verstreut, was im Mittel die Zugriffszeit herabsetzt.

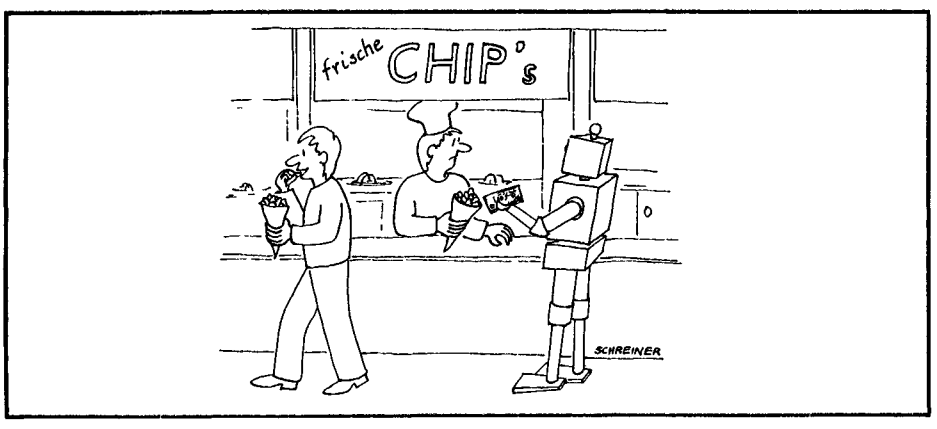
Es gibt aber noch ein weiteres Problem. In CP/M stehen nur 32 Byte pro Directory-Eintrag zur Verfügung. Davon werden 16 Byte für Dateinamen und bestimmte Kennungen benötigt, die anderen 16 Byte stehen für die Blocknummern zur Verfügung. Damit könnten also nur höchstens 16 KByte große Dateien dargestellt werden. CP/M löst dieses Problem aber sehr einfach. Wird eine Datei mit mehr als 16 KByte angelegt, so bekommt sie einfach einen weiteren Directory-Eintrag zugewiesen. Dort steht erneut der Dateiname – jedoch mit der Kennung, daß es sich um die Fortsetzung eines Directory-Eintrags handelt. Umgekehrt erhält der erste Eintrag noch einen Vermerk, daß er noch eine Fortsetzung besitzt. Mit diesem Verfahren lassen sich nun im Prinzip unbegrenzt große Dateien anlegen. CP/M beschränkt jedoch die Dateigrößen auf maximal 8 MByte.

Wieder ein anderes Problem ist die Blocknummer, die bei Verwendung von einem Byte als Zähler nur von 0 bis 255 laufen kann. Also können bei einer Blockgröße von 1024 Byte pro Block nur 256 KBytes erreicht werden. Durch Vergrößern der Blockgröße kann jedoch auch ein größerer Bereich adressiert werden. CP/M kann aber auch mit zwei Byte pro Blocknummer arbeiten, was durch Definitionen im BIOS möglich ist.

Nun aber ein Beispiel zu den sogenannten Extensions. Es soll eine Datei PASCAL mit 44 KByte angelegt werden.

BASIC	2, 3, 4, 5, 6, 7, 8, 9
USER	10, 11, 12, 13, 14, 19, 20, 21, 22, 23
ED	15, 16, 17, 18
PASCAL	24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39 (0)
PASCAL	40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, (1)
PASCAL	56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 67, 68, 69 (2)

Die Nummer, die in Klammer steht, gibt die Extension-Nummer an.



Die CCP-Befehle

Nach dem Start meldet sich jedes CP/M-System mit einer im BIOS definierten Überschrift, zum Beispiel:
60K CP/M VER 2.2
Danach erscheint das sogenannte Prompt-Symbol:
A⟩
Der Buchstabe A signalisiert, daß auf dem Disketten-Laufwerk A gearbeitet wird (was bei Systemstart immer automatisch eingestellt wird). CP/M kann Laufwerke von A bis P adressieren. Nach Erscheinen des Prompt-Symbols wartet CCP auf eine Befehlszeile.
CCP heißt Console Command Prozessor.

Die Steuerzeichen

Es gibt ein paar wichtige Steuerzeichen zur Unterstützung der Konsolbedienung.
Die Taste RUBOUT löscht das zuletzt eingegebene Zeichen und gibt es nochmals auf der Console aus. Gedacht ist diese Taste eigentlich für Teletype-Besitzer.
Komfortabler ist die Taste CTRL-H oder BACKSPACE, die ebenfalls ein Zeichen löscht, jedoch die Sequenz BACKSPACE SPACE BACKSPACE auf dem Bildschirm ausgibt, um das Zeichen auch dort zu löschen.
Mit der Taste CTRL-U kann eine ganze Eingabezeile gelöscht werden.
CTRL-X wirkt genauso, beim Löschen der Zeile wird jedoch der Cursor an den Anfang der Zeile geführt und die Zeichen in der Zeile werden auch auf dem Bildschirm getilgt.
CTRL-U ist wieder für Teletype-Besitzer gedacht, genauso wie CTRL-R, das die Eingabezeile erneut ausgibt.
Mit CTRL-E kann das Zeichen CR zu Verschönerungszwecken auf dem Bildschirm verwendet werden, ohne die Eingabe von Befehlen zu beenden, die ja normalerweise mit CR (RETURN-Taste) abgeschlossen wird.
CTRL-J (LINEFEED) kann neben der Return-Taste ebenfalls zum Abschluß einer Eingabezeile verwendet werden.
CTRL-C bewirkt Neuladen des CP/M-Betriebssystems. Dieser „Befehl" wird auch bei Diskettenwechsel benötigt.
CTRL-Z zeigt das Ende einer Consoleingabe an und wird von manchen Programmen (PIP, ED) als Ende-Zeichen erwartet.
CTRL-P schaltet den Drucker (LST-Schnittstelle) parallel zur Konsole. Damit kann man Protokolle von Ein- und Ausgaben auf dem Drucker erstellen.
CTRL-S stoppt die Ausgabe auf der Console. Damit können zum Beispiel schnell vorüberlaufende Listings angehalten werden. Ein weiteres CTRL-S gibt die Ausgabe wieder frei.
Eine Eingabezeile kann bis zu 255 Zeichen lang sein.

Die Befehle

Hier zunächst einmal die wichtigsten, in CCP direkt eingebauten Kommandos:
DIR
Das Kommando DIR erlaubt es, das Inhaltsverzeichnis einer Diskette auf die Konsole zu bringen. *Bild 1* zeigt ein Beispiel dafür.
Es zeigt, daß Dateinamen aus zwei Teilen bestehen, einem Hauptteil, der aus bis zu acht Zeichen besteht und einem Zusatz. Buchstaben und Ziffern sind zugelassen. Der Zusatz, der aus höchstens drei Zeichen bestehen kann, darf zur besseren Unterscheidung der verschiedenen Dateitypen hinzugefügt werden. Es haben sich bestimmte Standard-Zusätze herausgebildet, die aufzeigen, welches Programmsystem die Datei erzeugt hat. Der Zusatz BAS zeigt Basic-Dateien an, der Zusatz PAS Dateien von Pascal, mit ASM werden die Assemblerdateien gekennzeichnet und MAC bezeichnet die Dateien, die der Makroassembler MAC erzeugen und verstehen kann. Bis auf eine gleich zu schildernde Ausnahme dienen die Zusätze meist nur der Unterrichtung des Benutzers, oder des anfordernden Programmes. Ein Basic-Programm benötigt innerhalb einer CP/M-Datei natürlich andere Steuerinformationen als etwa ein Pascalprogramm und kann deshalb mit PAS gekennzeichnete Dateien nicht lesen. Übrigens werden diese Konventionen nicht einheitlich in jedem System eingehalten.
Beim Eintippen werden Name und Zusatz durch einen Punkt getrennt. Beispiel:

MCCPM.003
Die Ausnahme, bei der ein Zusatz zum Dateinamen von CCP auch ausgewertet wird, bildet der Fall, in dem ein Maschinenprogramm in das CP/M-System als Kommando integriert werden soll. Solche CCP-externen Kommandos müssen als Maschinenprogramm auf der Diskette abgelegt sein und die Namen dieser Dateien müssen den Zusatz COM besitzen. Wird an der Konsole ein Kommando eingetippt, dann versucht CCP zunächst dieses Kommando unter den eingebauten Standard-Kommandos zu finden. Wenn das mißlingt, dann werden auf der angesprochenen Diskette alle Dateinamen mit dem Zusatz COM untersucht. Wenn der Hauptname einer solchen „COM-Datei" mit dem gegebenen Kommando übereinstimmt, dann wird das zugehörige Maschinenprogramm in den Hauptspeicher gebracht und dort vom Rechner abgearbeitet.
Jedem der CCP-Kommandos können jeweils bestimmte Parameter mitgegeben werden, die genauer präzisieren, was zu tun ist. Beim CCP-Kommando DIR kann ein Parameter angegeben werden, der ein (unvollständiger) Dateiname sein muß. Ist im System zum Beispiel ein Laufwerk B vorhanden, dann kann man mit DIR B: den Inhalt der dort eingelegten Diskette besichtigen.
Möchte man zum Beispiel wissen, wieviele Programme man mit den Namen PROG1, PROG2, PROG3, PROG4...
schon abgelegt hat, dann kann man mit DIR PROG?
alle Dateinamen auf den Bildschirm bringen, die fünf Zeichen lang sind und mit PROG beginnen. Dabei wird in diesem Fall auf der momentan eingestellten

```
B>dir
B: MCCPM    003 : CPM3     BAK : CPM3     PRN : CPM3     TXT
B>
```

Bild 1. So wird der Inhalt einer Diskette angezeigt

Diskette gesucht, da keine Laufwerkbezeichnung angegeben wurde. Man nennt solch einen unvollständig angegebenen Namen im Slang „wildcard".
Eine Variante davon ist folgende:
DIR * . COM . Sie listet alle Kommandos auf.
DIR A*.* listet alle Dateinamen auf, die mit A beginnen.
DIR *.* bewirkt dasselbe, wie DIR.

Ein Stern bewirkt also, daß an seiner Stelle beliebige Buchstabenkombinationen im Namen auftauchen dürfen. Ein Fragezeichen bewirkt, daß an seiner Stelle ein beliebiger Buchstabe stehen darf.

Soweit zunächst zum Kommando DIR, das mithilfe der Wildcard-Technik sehr flexibel gehandhabt werden kann.

B:
Ein ganz einfaches Kommando veranlaßt die Neueinstellung des gerade aktuellen Laufwerkes: Mit dem Kommando B: wird das Laufwerk B aktuelles Laufwerk und es erscheint
B⟩
als Prompt-Symbol.

Wird nun der Befehl DIR ohne Parameter eingegeben, so erscheinen die Dateinamen der Dateien, die sich auf der Diskette des Laufwerks B befinden.

ERA

Ein weiterer wichtiger eingebauter Befehl ist das Kommando ERA. Damit können Dateien gelöscht werden. Als Parameter wird wieder ein Dateiname verlangt. Sollen mehrere Dateien gelöscht werden, so wird der Name mit „Wildcards" definiert. Beispiel: ERA *.* löscht alle Dateien auf der aktuellen Diskette. Bei diesem Befehl fragt CCP, ob wirklich alles gelöscht werden soll. ERA *.COM löscht alle Dateien, deren Name mit .COM endet und ERA BASIC.COM löscht das Programm mit dem Namen BASIC.COM. Der Befehl ERA schreibt in das Inhaltsverzeichnis der Diskette allerdings nur eine Kennung dafür, daß die Datei gelöscht wurde. Die Daten der Datei selbst werden dabei nicht entfernt (dennoch kann die Datei ohne zusätzliche Hilfsmittel nicht wieder beschafft werden). Erst wenn nach dem Löschen eine neue Datei angelegt wird, überschreibt diese dann normalerweise die Daten der alten. ERA kann dieselben Parameter verarbeiten, wie DIR. Zum Beispiel löscht ERA B:*.* alle Dateien auf Laufwerk B.

Wenn der Prompt C⟩ lauten würde, dann würde ERA *.* alle Dateien auf Laufwerk C löschen.

REN

Der Befehl REN dient zum Umbenennen von Dateien.

Das Kommando erhält dazu zwei Parameter und hat die Form
REN neuername = altername
Beispiel: Soll die Datei TEST.ASM in FERTIG.ASM umbenannt werden, geschieht das mit dem Kommando
REN FERTIG.ASM=TEST.ASM.
Wildcards sind hier nicht zugelassen, und es darf keine existierende Datei den neu einzuführenden Namen schon besitzen. Eine Laufwerksangabe kann ebenfalls erfolgen. Beispiel:

REN D:HALLO.DOK=D:HELLO.DOC.
Die Laufwerksangabe muß aber bei beiden Dateien identisch sein, sie darf bei einem der Namen auch wegfallen.

TYPE

Der Befehl TYPE dient der Ausgabe von Textdateien auf die Konsole. Damit können Dateien mit ASCII-Zeichen schnell angesehen werden. TYPE LIES.DAS gibt die Datei LIES.DAS auf die Konsole aus.

Mit CTRL-S kann die Ausgabe momentan gestoppt werden, um die Information bei einer schnell arbeitenden Konsole auch lesen zu können. Ein weiteres CTRL-S startet die Ausgabe wieder.

Durch Eingabe des Zeichens CTRL-C kann der gesamte Ausgabevorgang gestoppt werden. Ein Laufwerk kann auch hier wieder mit angegeben werden, also zum Beispiel TYPE B:LISTING.PRN gibt die Datei LISTING.PRN aus.

USER

Ab CP/M-Version 2.0 gibt es den Befehl USER. Damit ist es möglich, ein Inhaltsverzeichnis für mehrere Gruppen von Benutzern aufzuteilen. Dieser Befehl ist aus dem Multiuser-System MP/M abgeleitet, ist aber auch bei Hard-Disk-Laufwerken mit sehr vielen möglichen Datei-Einträgen für einen Benutzer allein nützlich, um die Dateien in übersichtliche Gruppen aufzuteilen. Der Befehl USER erhält als Parameter eine Zahl von 0 bis 15. 0 ist der Wert, der nach dem Systemstart voreingestellt ist. Mit dem Befehl USER 1 gelangt man ins Inhaltsverzeichnis 1 der aktuellen Diskette. Mit dem Befehl DIR kann man also nur die Dateien des aktuellen Benutzers ansehen. ERA *.* löscht auch nur alle Dateien des aktuellen Benutzers. Für den Normalgebrauch empfiehlt es sich aber, die Programme und Dateien als USER 0 zu halten.

SAVE

Ein sehr nützlicher Befehl ist das Kommando SAVE. Damit können Maschinenprogramme aus dem Hauptspeicher auf Diskette abgelegt werden – leider nicht unter Angabe von bestimmten Adreßbereichen, sondern nur von Adresse 100H ab, da dies bei CP/M die normale Start-Adresse von Benutzerprogrammen ist. Das Gebiet von 100H bis zum Anfang von CCP wird auch als TPA (transient program area) bezeichnet. SAVE muß zwei Parameter mitgeteilt bekommen. Zum einen die Anzahl der aufeinanderfolgenden 256-Byte-Blöcke (in dezimaler Schreibweise), die abgelegt werden sollen, zum anderen den Dateinamen, unter dem das Programm abgelegt werden soll.

Beispiel: SAVE 50 PROGRAM.COM legt 50*256 Byte, beginnend bei der Adresse 100H bis Adresse 32FFH, auf die Diskette. Der Name der neuen Datei ist PROGRAM.COM, also ein neues Kommando des CCP. Wenn man zum Beispiel ein neues Kommando EXIT konstruieren wollte, das in das mc-Monitorprogramm springt, etwa beim mc-CP/M-Computer mit großen Floppys, bei dem der Monitor nicht vom BIOS gelöscht wird, dann könnte man nach dem RESET des Computers mit Hilfe des Monitors beginnend bei Adresse 100H folgende Sequenz ablegen:

CD 1E F0
Das ist ein CALL, ein Unterprogrammsprung zur RESTART-Adresse des Monitors. Danach muß man das CP/M-System starten. Der Speicher-Bereich 100H bis CCP-Beginn, also die TPA, wird dabei nicht zerstört. Das CP/M-System meldet sich mit A⟩. Dann wird der Befehl SAVE 1 EXIT.COM eingegeben. Das kurze Programm aus 3 Byte wird zusammen mit 253 weiteren Byte auf die Diskette gerettet. Nun kann CP/M durch Eingabe des Kommandos EXIT wieder in den Monitor zurückspringen.
Dieses kurze Programm ist übrigens ein schlechtes Beispiel für ein CP/M-Programm, da es nur auf dem mc-CP/M-Computer funktioniert.

Die Disk-Befehle

Alle Dateien auf der aktuellen Diskette, deren Extension mit „COM" endet, sind direkt ausführbare Befehle. Es handelt sich dabei um Programme, die ab Adresse 100H in die sogenannte TPA (transient program area) geladen werden und bei 100H gestartet werden. Die Befehle, die sich dadurch ergeben, werden daher auch als „transient commands" bezeichnet.
Durch die .COM-Dateien kann also der Befehlssatz des CP/M erweitert werden. Auf der Systemdiskette des Standard-CP/M befinden sich einige solche Dateien, die nützliche Erweiterungen des Systems darstellen:

STAT.COM, ASM.COM, LOAD.COM, DUMP.COM, PIP.COM, ED.COM, SYSGEN.COM, MOVCPM.COM, SUBMIT.COM, XSUB.COM.

Genauso wie bei den eingebauten Befehlen können auch hier je nach Art des Kommandos Parameter wie Dateinamen usw. hinzugefügt werden.

Wieviel Platz ist auf der Disk?

Der Befehl STAT gibt Auskunft über das Inhaltsverzeichnis einer Diskette, ähnlich wie bei DIR, jedoch kann hier auch die Größe von Dateien festgestellt werden. Wird der Befehl STAT ohne weiteren Parameter angegeben, so erfolgt auf der Konsole z. B. folgende Ausgabe:
A: R/W, SPACE: 25K
oder auch
B: R/O, SPACE: 3K
Im ersten Fall wurde der Befehl vom Laufwerk A aus gegeben. Die Angabe R/W gibt an, daß auf diesem Laufwerk derzeit Lese- und Schreiboperationen erlaubt sind. Dies kann sich ändern, wenn zum Beispiel beim Laufwerk A die Diskette gewechselt wurde, ohne daß die Taste CTRL-C betätigt wurde, um einen Warm-Boot zu erreichen. Beim zweiten Beispiel war dies bei Diskette B der Fall. Dann gibt STAT R/O aus, und die Diskette ist nur noch zum Lesen freigegeben.

Warum aber die Unterscheidung von R/W und R/O?
Beim Warmboot wird das Inhaltsverzeichnis der Diskette A eingelesen, ebenfalls bei einem Zugriff auf ein neues Laufwerk, z. B. durch den Befehl „B:" wird dessen Inhaltsverzeichnis gelesen.

```
A>stat *.*

 Recs  Bytes  Ext Acc
   64     8k    1 R/W A:ASM.COM
  146    19k    2 R/W A:BASIC.COM
   96    12k    1 R/W A:BIOS.ASM
    4     1k    1 R/W A:BOOT.ASM
   35     5k    1 R/W A:CBIOS.ASM
   16     2k    1 R/W A:COP.COM
   38     5k    1 R/W A:DDT.COM
   80    10k    1 R/W A:DEBLOCK.ASM
   49     7k    1 R/W A:DISKDEF.LIB
   33     5k    1 R/W A:DUMP.ASM
    4     1k    1 R/W A:DUMP.COM
   52     7k    1 R/W A:ED.COM
    5     1k    1 R/W A:FORMAT.COM
   14     2k    1 R/W A:LOAD.COM
  121    16k    1 R/W A:MINIMAX.MAC
  153    20k    2 R/W A:MONI3.MAC
   76    10k    1 R/W A:MOVECPM.COM
   58     8k    1 R/W A:PIP.COM
    8     1k    1 R/W A:PRTHEX.BAS
   41     6k    1 R/W A:STAT.COM
   10     2k    1 R/W A:SUBMIT.COM
    8     1k    1 R/W A:SYSGEN.COM
  104    13k    1 R/W A:TDLASM.COM
    6     1k    1 R/W A:XSUB.COM
Bytes Remaining On A: 75k
```

Bild 1. Ausgabe des Disk-Inhaltsverzeichnisses mit STAT

Daraus wird ein Bitmuster abgeleitet, in dem verzeichnet ist, welche Blöcke auf der Diskette noch nicht verwendet wurden. Diese „Bitmap" wird beim Anlegen von neuen Dateien benötigt, um ihnen einen Platz auf der Diskette zuzuweisen. Wird nun die Diskette gewechselt, so bleibt die Bitmap erhalten. Das CP/M-Betriebsystem prüft nun aber beim Anlegen einer neuen Datei aus Sicherheitsgründen nochmals das Inhaltsverzeichnis und vergleicht die Bitmap. Liegt eine Differenz vor, so wird die Diskette als R/O gekennzeichnet. Es ist dann nicht mehr möglich, dort eine Datei anzulegen, bis durch einen Warmboot die gespeicherte Bitmap neu konstruiert wird. Die Bitmap kann aber auch durch ein Anwenderprogramm über einen BDOS-Aufruf selbst erzwungen werden, ohne einen Warmstart durchführen zu müssen: In manchen Basic-Interpretern gibt es dazu den Befehl RESET.

STAT kann aber auch noch eine Reihe anderer Parameter erhalten. Wird der Laufwerksname mit einem nachfolgenden Doppelpunkt angegeben, so wird der freie Speicher dieses Laufwerks angegeben, z. B. „STAT B:".

Länge einzelner Dateien

Ebenfalls kann ein Dateiname angegeben werden, der auch „Wildcards" enthalten darf; z. B. gibt STAT *.* alle Dateinamen einer Diskette aus. Im Gegensatz zu DIR werden die Dateinamen hier mit einer Längeninformation ausgegeben. *Bild 1* zeigt den Inhalt der CP/M-Diskette, wie sie zum mc-CP/M-Computer geliefert wird.

Durch den Befehl „STAT B:=R/O" wird das Laufwerk B in den R/O-Zustand versetzt. Der STAT-Befehl kann aber noch mehr. So kann auch einzelnen Dateien ein R/O-Attribut zugewiesen werden, oder sie können als Systemdateien gekennzeichnet werden und erscheinen bei einem DIR-Befehl nicht mehr. Mit „STAT*.*$S" werden sie wieder ausgegeben. Mit dem Kommando „STAT VAL:" kann man sich eine Übersicht über die mögliche Syntax bei Zuweisungen verschaffen. *Bild 2* zeigt das Ergebnis dieses Befehls. Dort erkennt man auch die Möglichkeit, den logischen Geräten wie „CON:" (Console), „RDR:" (Lesekanal), „PUN:" (Schreibkanal), „LST:" (Druckerkanal) physikalische Geräte wie „TTY:", „CRT:" usw. zuzuweisen. Durch diese Befehle wird ein Byte im Speicher auf Adresse 3 abgelegt. Die Belegung der einzelnen Bits entspricht im wesentlichen der im Monitor des mc-Computers verwendeten, jedoch werden die Bits im mc-Computer nicht ausgewertet, da der Monitor einen eigenen Speicherplatz dafür verwendet. Das IOBYTE (Adresse 3) wird normalerweise im BIOS ausgewertet. Über die aktuelle Zuweisung kann man sich durch den Befehl STAT DEV: informieren, *Bild 3* zeigt den Ausdruck.

In *Bild 4* ist gezeigt, wie man sich mit Hilfe des „STAT DSK:"-Befehls über die Laufwerkseigenschaften informieren kann. Dabei sind es genau die Informationen, die im BIOS eingetragen sind.

Laufwerk A ist in diesem Beispiel eine Standard-8"-Floppy (Single Density), E ist ein Plattenlaufwerk.

Schließlich gibt es noch den Befehl „STAT USR:", dessen Ergebnis in *Bild 5* sichtbar ist. Unter CP/M 2.2 ist es möglich, mit verschiedenen Benutzernummern Dateien anzulegen. Durch den Befehl „STAT USR:" erfährt man, welche Gebiete belegt sind. Die Benutzernummern helfen auch, Dateien nach logischen Gesichtspunkten zu gruppieren, was insbesondere bei Plattenlaufwerken wichtig ist. Mit dem Befehl USER n, der ein eingebauter Befehl ist, wird die Umschaltung vorgenommen. n kann 0 bis 15 sein. USER 2 schaltet z. B. auf den Benutzer 2 um. Nun können dort ganz normal Dateien angelegt und bearbeitet werden. Mit DIR erscheinen nur die Dateien des aktuellen Benutzers. ERA*.* löscht auch nur die Dateien eines Benutzers.

Der Assembler ASM

Auf der Diskette befindet sich auch ein Assembler, mit dem Maschinenprogramme erzeugt werden können; er ist allerdings aber nur für 8080-Mnemonics geeignet. Die Programme laufen aber natürlich auch auf einem Z80-System. Der Aufruf erfolgt z. B. mit ASM TEST, wobei die Datei TEST.ASM gelesen und die Dateien TEST.HEX und TEST.PRN erzeugt werden: In TEST.HEX steht im Intel-Hex-Format der Objectcode und in TEST.PRN wird das Listing abgelegt. *Bild 6* zeigt den Ausdruck der Datei TEST.PRN. In *Bild 7* ist der Dump der Hex-Datei dargestellt.

LOAD

Soll eine .COM-Datei erzeugt werden, so muß das Intel-Hex-Format noch geladen werden. Dazu gibt es den Befehl LOAD. Mit LOAD TEST wird eine .COM-Datei erzeugt, wie in *Bild 8* zu sehen ist.

DUMP

Mit dem Befehl DUMP kann der Inhalt einer .COM-Datei auf dem Bildschirm in lesbarer Hex-Form ausgegeben werden. *Bild 9* zeigt ein Beispiel. In *Bild 10* schließlich ist das Ergebnis des neuen Befehls gezeigt. *Bild 11* zeigt alle nun auf der Diskette vorhandenen Dateien. Die Programme ASM und LOAD befanden sich auf Diskette A.

PIP: Kopieren von Dateien

Ein universelles Programm zum Kopieren von Dateien ist PIP. Damit lassen sich Dateien auf den Drucker ausgeben,

```
A>stat val:

Temp R/O Disk: d:=R/O
Set Indicator: d:filename.typ $R/O $R/W $SYS $DIR
Disk Status  : DSK: d:DSK:
User Status  : USR:
Iobyte Assign:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
```

Bild 2. Kommandoformate von STAT

```
A>stat dev:
CON: is UC1:
RDR: is UR2:
PUN: is UP2:
LST: is UL1:
```

Bild 3. Gerätezuordnung

```
A>stat usr:

Active User : 0
Active Files: 0
```

Bild 5. User-Gebiete

```
E>a:stat dsk:

    A: Drive Characteristics
 1920: 128 Byte Record Capacity
  240: Kilobyte Drive  Capacity
   64: 32  Byte Directory Entries
   64: Checked  Directory Entries
  128: Records/ Extent
    8: Records/ Block
   26: Sectors/ Track
    2: Reserved Tracks

    E: Drive Characteristics
65536: 128 Byte Record Capacity
 8192: Kilobyte Drive  Capacity
 1024: 32  Byte Directory Entries
 1024: Checked  Directory Entries
  512: Records/ Extent
   64: Records/ Block
  256: Sectors/ Track
    0: Reserved Tracks
```

Bild 4. Diskettencharakteristik

```
B>type test.prn

                ; Kleines Testprogramm fuer den ASM
                ;
0100                    ORG 100H        ;start in TPA
0005 =          BDOS    EQU     5
0100 110D01     START:  LXI D,TEXT      ;Adresse des Textes
0103 0E09               MVI C,9         ;Funktion STRING AUS
0105 CD0500             CALL BDOS       ;aufruf ueber CP/M
0108 0E00               MVI C,0         ;Warm Start
010A CD0500             CALL BDOS
                        ;kommt nicht hierher

010D 5374617274TEXT:    DB 'Start eines Anwenderprogramms'
012A 0D0A               DB 0DH,0AH
012C 24                 DB '$'          ;ende des Strings

012D                    END
```

Bild 6. Ausgabe der mit ASM erzeugten .PRN-Datei

```
B>type test.hex
:100100000110D010E09CD05000E00CD0500537461DF
:1001100072742065696E657320416E77656E6465E3
:0D0120007270726F6772616D6D730D0A244D
:00000000000

B>
```

Bild 7. Intel-Hex-Datei

```
B>a:load test

FIRST ADDRESS 0100
LAST  ADDRESS 012C
BYTES READ    002D
RECORDS WRITTEN 01

B>
```

Bild 8. Ausführung des LOAD-Befehls

Dateien von einem Laufwerk auf das andere kopieren oder aneinanderhängen. PIP läßt eine Vielzahl von Parametern zu, um die Daten bei der Übertragung zu manipulieren.
PIP kann auf zwei Arten aufgerufen werden: einmal ohne die Angabe von Parametern, dann meldet sich PIP mit dem Zeichen „*" und ist für eine Eingabe bereit. Wird ein Parameter angegeben, so wird dieser sofort interpretiert und ausgeführt. Die Grundform lautet:
zieldateiname = quelldateiname
oder
zieldateiname = quelldateiname 1, quelldateiname 2...
Als Quell- und Zielnamen können neben normalen Dateien auch logische Geräte angegeben werden. Beim Ziel sind dies z. B. „CON:", „PUN:", „LST:", „PRN:" (mit expandierten Tabulatoren und Seitenvorschub alle 60 Zeilen) und „OUT:" (eigene Anpassung auf 106H), als Quelle CON:, RDR: und INP: (eigene Anpassung auf 109H). Hinter den Dateinamen kann ein weiterer Parameter in eckigen Klammern folgen:
B (Block mode), Dn (Löschen nach ntem Zeichen), E (echo mode), F (Formfeeds entfernen), H (Intelhexcheck) I (:00 ignorieren), L (in Kleinbuchstaben wandeln), N (Zeilennummern ausgeben), O (Objektdatei-Kopie), Pn (Seitenvorschub alle n Zeilen), Qs↑z (Kopierstop, wenn String s bis CTRL-Z gefunden), Ss↑z (Kopierstart, wenn String gefunden), Tn (Tabulatoren auf n expandieren), U (Kleinbuchstaben in große umwandeln), V (Prüflesen bei Diskettenbetrieb), Z (Paritätsbit auf 0 setzen).
Mit „PIP B:*.*=A:*.*" werden alle Dateien von Laufwerk A auf Laufwerk B kopiert. *Bild 12* zeigt ein weiteres Beispiel.

Der CP/M-Texteditor

Auch ein kleiner Texteditor wird auf der Systemdiskette mitgeliefert. Damit lassen sich die Quellprogramme von der Tastatur eingeben. Zum Editor gehört ein eigenes Handbuch und er kann hier nur grob beschrieben werden. Der Aufruf erfolgt mit ED name, also z. B. ED TEST.ASM. Es wird, falls nicht schon vorhanden, eine neue Datei angelegt. Dann meldet sich der Editor mit dem Zeichen „*" und wartet auf Befehle. Es kann nun z. B. der Befehl I (Insert = Einfügen), gefolgt von beliebigen Textzeilen, eingegeben werden. Mit CTRL-Z wird der I-Befehl beendet.
Der Editor besitzt intern eine Textmarke; dort merkt er sich, an welcher Stelle des Textes er gerade arbeiten soll. Nach dem I-Befehl steht dieser Zeiger am Ende der gerade eingegebenen Daten. Soll nun der Text ausgegeben werden, so kann z. B. mit dem Befehl B an den Anfang des Textes zurückgegangen werden. Mit T30 lassen sich von hier an 30 Zeilen auf der Console ausgeben. L30 verschiebt den Textzeiger um 30 Zeilen; mit –L30 kann er auch wieder zurückgesetzt werden.

Bild 13 zeigt ein paar Befehle. Verläßt man den Editor mit E, so werden automatisch die eingegebenen Texte auf Disk abgespeichert. Soll erneut editiert werden, so genügt es, ED TEST.ASM aufzurufen, und mit dem Befehl 200A werden die ersten 200 Zeilen (oder soviel wie existieren, falls es weniger sind) eingelesen. Dann kann der Text modifiziert werden, und mit E wird der Editor wieder verlassen. Die Datei, die vor der Editierung existierte, hat nun die Extension .BAK erhalten und die neuerzeugte .ASM, wie es mit ED TEST.ASM angegeben wurde. Die Datei TEST.BAK dient dazu, einem Datenverlust durch fehlerhafte Bedienung vorzubeugen, so daß immer eine Vorgängerversion erhalten bleibt. Sie kann mit dem Befehl REN in TEST.ASM umbenannt werden, nachdem diese zuvor gelöscht wurde.

```
B>a:dump test.com

0000 11 0D 01 0E 09 CD 05 00 0E 00 CD 05 00 53 74 61
0010 72 74 20 65 69 6E 65 73 20 41 6E 77 65 6E 64 65
0020 72 70 72 6F 67 72 61 6D 6D 73 0D 0A 24 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

B>
```

Bild 9. Erzeugte .COM-Datei mit DUMP ausgegeben

Bild 10. Start der Datei TEST.COM

```
B>test
Start eines Anwenderprogramms
B>
```

Bild 11. Dateien, die für Bild 10 benötigt wurden

```
B>a:stat test.*
 Recs  Bytes  Ext Acc
    3     1k    1 R/W B:TEST.ASM
    1     1k    1 R/W B:TEST.COM
    2     1k    1 R/W B:TEST.HEX
    5     1k    1 R/W B:TEST.PRN
Bytes Remaining On B: 202k
B>
```

```
B>a:pip lst:=test.asm[nt8]
     1: ; Kleines Testprogramm fuer den ASM
     2: ;
     3:          ORG 100H          ;start in TPA
     4: BDOS     EQU     5
     5: START:   LXI D,TEXT        ;Adresse des Textes
     6:          MVI C,9           ;Funktion STRING AUS
     7:          CALL BDOS         ;aufruf ueber CP/M
     8:          MVI C,0           ;Warm Start
     9:          CALL BDOS
    10:          ;kommt nicht hierher
    11:
    12: TEXT:    DB 'Start eines Anwenderprogramms'
    13:          DB 0DH,0AH
    14:          DB '$'            ;ende des Strings
    15:
    16:          END
    17:
    18:

B>
```

Bild 12. Anwendung des PIP-Befehls

```
B>a:ed test.asm
     : *200a
    1: *5t
    1:  ; Kleines Testprogramm fuer den ASM
    2:  ;
    3:          ORG 100H          ;start in TPA
    4:  BDOS    EQU     5
    5:  START:  LXI D,TEXT        ;Adresse des Textes
    1: *-b
     : *-10t
    9:          CALL BDOS
   10:          ;kommt nicht hierher
   11:
   12:  TEXT:   DB 'Start eines Anwenderprogramms'
   13:          DB 0DH,0AH
   14:          DB '$'            ;ende des Strings
   15:
   16:          END
   17:
   18:
     : *e
B>
```

Bild 13. Eingabe eines kurzen Assembler-Quellenprogramms mit dem Texteditor ED

Spezielle Hilfsmittel

Mit dem Kommando PIP kann man nur normale Dateien kopieren, es ist nicht möglich, damit die Systemspuren zu kopieren. Dies wird aber bei der Erzeugung von Sicherheitskopien benötigt. Mit dem Kommando „SYSGEN" kann man aber das BIOS auf die Diskette an die richtige Stelle bringen. Man kann damit sogar ein neues BIOS abspeichern.

Kopieren der Systemspuren

SYSGEN besitzt normalerweise keine Parameter. Wird es aufgerufen, so meldet es sich wie folgt:

SYSGEN VERSION 2.0
SOURCE DRIVE NAME (OR RETURN TO SKIP)

Darauf kann als Antwort eine Laufwerksbezeichnung (A, B, C oder D) angegeben werden, normalerweise A. Dort sollte eine Diskette mit „System" montiert sein. Falls im Speicher schon ein System an die richtige Stelle geladen wurde (z. B. durch MOVCPM), dann muß hier nur mit einem Wagenrücklauf (CR) geantwortet werden. Der Bootsektor wird dabei auf der Adresse 900H abgelegt. Auf 980H beginnt das CP/M-Betriebssystem und auf Adresse 1F80H beginnt das BIOS – wie gesagt, nur für die Kopierzwecke.

Wurde ein Laufwerk angegeben so erscheint folgende Meldung:

SOURCE ON x THEN TYPE RETURN,

wobei x für das angegebene Laufwerk steht. Es kann dann die Diskette mit dem zu ladenden System eingelegt werden und anschließend ein CR getippt werden. Danach erfolgt der Diskettenzugriff und es erscheint die Meldung:

FUNCTION COMPLETE.

Jetzt erscheint (auch wenn am Anfang kein Laufwerkname eingegeben wurde) folgende Meldung:

DESTINATION DRIVE NAME (OR RETURN TO REBOOT).

Nun kann die Zieldiskette angegeben werden. Es erscheint:

DESTINATION ON x THEN TYPE RETURN.

Nun wird die Zieldiskette eingelegt und nach dem Wagenrücklauf wird das System auf die unteren beiden Spuren des Laufwerks zurückgeschrieben.

Danach erscheint

FUNCTION COMPLETE.

Und wieder die Meldung:

DESTINATION DRIVE NAME (OR RETURN TO REBOOT),

um weitere Kopien zu erlauben. SYSGEN gestattet es, wie schon gesagt, ein eigenes BIOS (oder auch Boot-Programm) auf die Diskette zu bringen. Dazu muß ein lauffähiger CP/M-Computer zur Verfügung stehen. Es wird zuerst das alte System mit SYSGEN geladen und danach RESET betätigt. Nun kann der eben geladene Bereich, der bei CP/M-Computern durch RESET nicht zerstört wird, z. B. in einen höheren Bereich gerettet werden. Dann könnte man mit DDT das neue BIOS oder BDOS laden und anschließend kann mit Move-Befehlen das alte System wieder in den Originalbereich kopiert werden. Am Schluß muß das neue BOOT-Programm auf 900H bis 97FH stehen, das CP/M auf 980H bis 1F7F und das neue BIOS beginnend bei 1F80H. Nun kann das CP/M wieder gebootet werden und SYSGEN wird erneut aufgerufen. Diesmal wird keine Quelle als Laufwerk angegeben, sondern nur die neue Zieldiskette.

Verschieben des CP/M Betriebssystems

Mit dem Befehl MOVCPM ist es möglich, das CP/M-Betriebssystem für unterschiedliche Speicherbereiche zu konfektionieren. Man könnte auch ein „kleines" CP/M erzeugen, um den höheren Speicherbereich für andere Programme fest zu reservieren. MOVCPM kann dazu mehrere Parameter erhalten.

MOVCPM ohne Parameter erzeugt ein maximales CP/M-System. Es wird dazu nach einem durchgehenden RAM-Bereich von 0 an gesucht. Das ist aber bei den wenigsten Computern nützlich, da im Speicher befindliche Routinen, wie z. B. ein Monitorprogramm, dann überschrieben werden können.
Besser ist es, die neue Größe anzugeben:

MOVCPM n

Wobei n die „Größe" des CP/Ms, angegeben im KByte, ist. Für den mc-CP/M-Computer beträgt sie 60 KByte, also:

MOVCPM 60

Damit wird ein 60-KByte CP/M generiert. Auf dem mc-CP/M-Computer läßt sich auch ein 64-KByte-System fahren, wenn die Floppy-Routinen aus dem Monitor genommen werden und in das Bios verlagert werden.

MOVCPM hat noch einen zweiten Parameter. Wird dort das Zeichen * angegeben, so wird das System nur „auf der Stelle" verschoben, jedoch noch nicht in den echten Bereich transportiert. Es wird für den SYSGEN-Befehl vorbereitet und kann damit auf die Diskette zurückgeschrieben werden. Auch hier ist es möglich, noch das BIOS durch ein eigenes auszutauschen, ehe es verwendet wird.

Bei dem im Handel befindlichen Original-CP/M-Betriebssystem wird nach MOVCPM das BIOS für das Intel MDS 800 System geladen. MOVCPM darf aber von lizensierten Händlern konfektioniert werden. Das gilt zum Beispiel für den mc-CP/M-Computer. Dort wird nach dem MOVCPM sofort ein ablauffähiges BIOS bereitgestellt. Das Gleiche gilt für das BOOT-Programm. Das CP/M wird

normalerweise als 20-KByte-System ausgeliefert, um alle Möglichkeiten auch für kleinere Systeme offen zu halten.

Hier nun der Ablauf zur Erzeugung eines 60-KByte-Systems.

Zuerst der Befehl

MOVCPM 60 *

Danach gibt das Programm MOVCPM folgende Meldung aus:

READY FOR „SYSGEN" OR

„SAVE 32 CPM60.COM"

Es kann hier also auch eine .COM – Datei gebildet werden, die sich aber nicht direkt starten läßt. Besser aber ist es, das System sofort nach dem Booten zu haben. Daher wird dann SYSGEN aufgerufen. Die Quelle wird mit CR beantwortet und die Zieldiskette angegeben. Danach befindet sich dort das 60-KByte-System.

Die Meldung beim Booten, die die Größe des CP/Ms angibt, wird im BIOS automatisch vom MOVCPM-Programm aktualisiert, so daß nach dem Kaltstart die neue Größe des Systems auch gemeldet wird.

MOVCPM kann auch mit zwei Sternen aufgerufen werden

MOVCPM **

Dann wird ein maximales CP/M erzeugt und für die SYSGEN-Operation vorbereitet.

Die Kommandodateien .SUB

Werden häufige Kommandosequenzen immer wieder benötigt, so können diese Kommandos in eine Datei geschrieben werden und mit dem Programm SUBMIT werden sie dann automatisch nacheinander ausgeführt. Die Datei muß dabei die Extension .SUB besitzen. SUBMIT kann mehrere Parameter erhalten. Der erste Parameter ist der Name der abzuarbeitenden Datei. Dann können noch Parameter angegeben werden, die an die Kommandos in der Datei als lokale Parameter weitergereicht werden können. Die Parameter haben in dieser Datei die feste Bezeichnung $1, $2, $3, $4, ... $n. Nach dem Start der SUBMIT-Funktion wird eine Zwischendatei $$$.SUB erzeugt. Danach wird ein Reboot (Warm Start) durchgeführt. Nun sucht der CCP (Consol Command Prozessor) auf dem Laufwerk A nach einer solchen Datei. Findet er diese, so werden die einzelnen darin befindlichen Befehle ausgeführt. Die Datei $$$.SUB muß also auf dem Laufwerk A abgelegt worden sein. Die Ausführung kann durch Eingabe des Zeichens „Rubout", während der CCP ein neues Kommando aus dieser Datei liest und dabei auf dem Bildschirm anzeigt, abgebrochen werden.

Beispiel: Es soll eine Programm-Übersetzung mit nachfolgendem Laden durchgeführt werden. Dabei soll die zu übersetzende Datei beim Start dieser Kommandodatei angegeben werden.

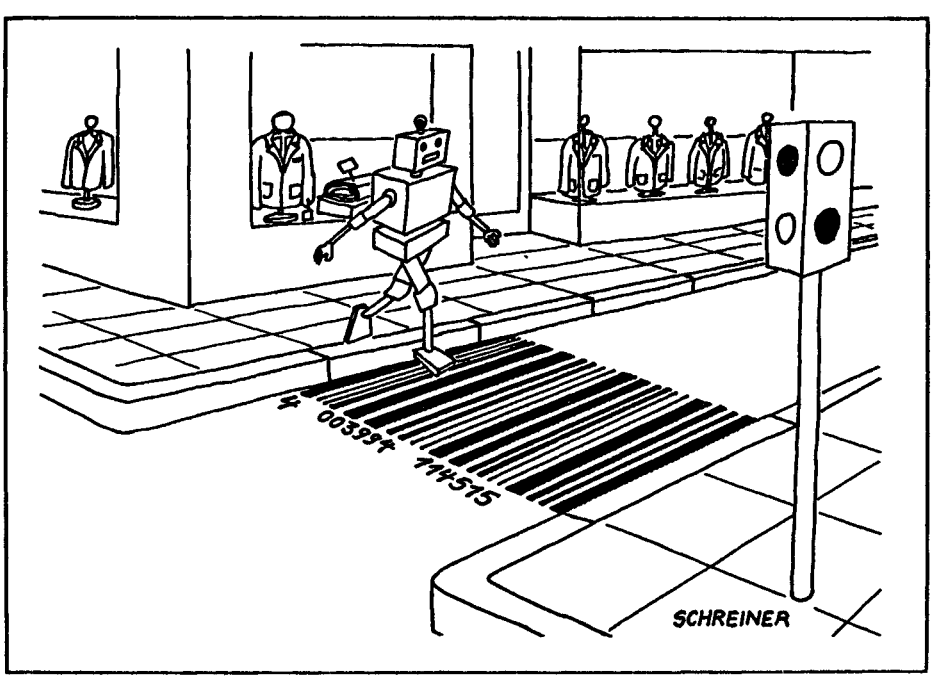
Inhalt der Kommandodatei:
ASM $1
LOAD $1
ERA $1.HEX

Die Datei soll den Namen UEBER.SUB erhalten. Nun kann z. B. mit der Sequenz SUBMIT UEBER TEST die (schon existente) Datei TEST.ASM übersetzt werden und es wird eine Datei TEST.COM erzeugt. Die Zwischendatei TEST.HEX wird gelöscht. Im Betriebssystem 2.2 gibt es noch eine weitere Datei, die dazugehört, XSUB.COM. Mit SUBMIT war es nämlich nicht möglich, Konsoleingaben in die Kommando-Datei einzutragen. XSUB behebt diesen Mangel. Dazu wird XSUB als erster Befehl in eine Kommandodatei geschrieben. Nun ist folgende Sequenz möglich:

XSUB
DDT
I$1.HEX
R
G0
SAVE 2 $2.COM.

Wenn die Datei mit dem Namen SICHERE.SUB angelegt wurde, so kann durch die Sequenz
SUBMIT SICHERE TEST NEUTEST
eine Datei TEST.HEX in eine Datei NEUTEST.COM gewandelt werden, wobei diese Sequenz nicht das LOAD-Kommando ersetzen kann, da nur konstant ein Block abgelegt wird.

XSUB lädt sich selbst unter den Bereich des CCP und meldet sich von dort mit (xsub active) nach jedem Warm-Start. So lange, bis ein Kaltstart durchgeführt wird.



Einfache BDOS-Befehle

Von Leuten, die sich über die Uneinheitlichkeit der verschiedensten Computersysteme ärgern, weil eben nichts kompatibel ist, wird auch dem Betriebssystem CP/M der Vorwurf gemacht, daß es doch nicht genügend Kompatibilität biete, denn Disketten (oder Software mit besonderen Steuerzeichen für das Terminal) könnten nicht ohne weiteres von einer Konfiguration auf die andere gebracht werden. Die Leute haben zwar sachlich recht, aber daß sie sich ärgern, mag daran liegen, daß sie etwas von CP/M verlangen, was CP/M nie versprochen hat: Normierung der Diskettenformate oder der Terminalprotokolle. Was dagegen den Wert von CP/M wirklich ausmacht, das kann gerade an BDOS gut gezeigt werden.

BDOS – einheitliche Schnittstelle von CP/M-Software zur Hardware

Um die einzelnen BDOS-Fähigkeiten, die BDOS-Funktionen, für ein Anwenderprogramm nutzbar zu machen, wird ein standardisiertes Verfahren zum Aufruf benutzt. Und zwar wird von CP/M beim Systemstart in Zelle 5 des Speichers ein Sprung geschrieben, der ins BDOS führt. Ein Benutzerprogramm kann deshalb mit CALL 5 ins BDOS verzweigen. Damit man bestimmte Dienstleistungsroutinen gezielt anspringen kann, sind diese durchnumeriert. Das Benutzerprogramm muß diese Nummer vor dem Absprung ins BDOS im Register C des Prozessors ablegen. In BDOS wird als erstes diese Nummer ausgewertet und dann entsprechend weiterverzweigt. In manchen Fällen verlangt eine BDOS-Routine noch weitere Informationen, eine bestimmte Speicheradresse zum Beispiel, an der ein auszugebender Text stehen könnte. Solche Adressen werden BDOS über die beiden zu einem 16-Bit-Zeiger zusammengefaßten Register D und E mitgeteilt.
Das Ergebnis eines „BDOS-Calls" wird in den meisten Fällen in Register A an das Benutzerprogramm zurückgeliefert. Zum Beispiel das Ergebnis einer Tastaturabfrage. Das eingegebene Zeichen steht dem Benutzerprogramm im Akkumulator A zur Weiterverarbeitung zur Verfügung. Zwei-Byte-Rückmeldungen werden mit den Registern H und L von BDOS ans Programm übertragen. In diesen Fällen ist das Register A zusätzlich identisch mit dem Register L und das Register B enthält den Wert des Registers H (um mit frühen Versionen von CP/M kompatibel zu bleiben).

Die BDOS-Funktionen für die Kommunikation mit der Außenwelt

Funktion 0: System-Neustart
Mit dem Code 0 in Register C wird mit CALL 5 ein Warm-Boot des CP/Ms veranlaßt. Die Wirkung ist identisch mit einem Sprung auf die Adresse 0, da dort der Sprung zum WARM-Boot des BIOS stattfindet. Beispiel eines Aufrufs an geeigneter Stelle im Anwenderprogramm:

```
...
MVI C,0
CALL 5
...
```

Natürlich erfolgt nach diesem Unterprogrammaufruf kein Rücksprung, sondern es meldet sich CCP mit dem aktuellen Laufwerk und wartet auf ein Kommando.

Funktion 1: Konsolen-Eingabe

Bei CALL 5 mit dem Inhalt 1 in Register C gilt: Das nächste Zeichen wird in das Register A eingelesen. Dabei wartet die Routine in BDOS solange, bis ein Zeichen eingegeben wurde. Alle Zeichen werden auf der Konsole auch mit ausgegeben (Echo-Betrieb). Wenn das eingegebene Zeichen CTRL-I lautet (TAB), werden bei der Ausgabe so viele Leerzeichen gegeben, daß der Cursor in einer der Spalten 8, 16, 24... (immer in der nächst erreichbaren) erscheint. Das Zeichen LF (0A) bei der Eingabe setzt den internen Tabulatorzähler auf 0 zurück.
Beispiel: Warten auf das Zeichen CR (0D).

```
WARTE: MVI C,1
       CALL 5
       CPI 0DH
       JNZ WARTE
```

Funktion 2: Konsolen-Ausgabe

Mit der Funktion 2 ist es möglich, ein Zeichen auf der Konsole auszugeben. Wie bei der Funktion 1 werden Tabulatoren (CRTL-I) expandiert. Das auszugebende Zeichen wird im Register E an das BDOS übergeben. Wird ein CTRL-S an der Konsole eingegeben, so wird die Ausgabe bis zur Eingabe von CTRL-Q angehalten.

Beispiel einer Ausgabesequenz:

```
MVI C,2
MVI E,'A'
CALL 5
```

Durch diese Sequenz wird das Zeichen „A" auf der Konsole ausgegeben.

Funktion 3: Reader-Eingabe

Vielleicht erinnern Sie sich noch an die BIOS-Funktion zum Reader-Kanal. Dieser Kanal kann mit Funktion 3 angesprochen werden. Beim mc-CP/M-Computer ist dies der Eingabekanal der SIO B. Im Monitor, vor dem CP/M-Start, muß der Kanal mit AR = P auch logisch zugewiesen worden sein, sonst werden die Zeichen fälschlich von der Konsole erwartet.
Ein ankommendes Zeichen gelangt in das Register A. Es wird solange gewartet, bis wirklich ein Zeichen angekommen ist.

Beispiel: Warten auf das Zeichen „A"

```
WARTE: MVI C,3
       CALL 5
       CPI 'A'
       JNZ WARTE
```

Bei der Eingabe vom Reader wird kein Echo erzeugt.

Funktion 4: Punch-Ausgabe

Damit kann der Punch-Kanal angesprochen werden. Beim mc-CP/M-Computer

ist dies die Ausgabe auf SIO B, wenn im Monitor vor dem Start von CP/M der Befehl AP.= P gegeben wurde, um den Kanal auch logisch zuzuordnen. Das Zeichen muß an BDOS in Register E übergeben werden. Die Ausgabe erfolgt transparent, das heißt, es werden die Zeichen genauso übertragen wie sie im Register E stehen (die Konsol-Ausgabe mit der Funktion 2 z. B. ist dagegen nicht transparent, denn das Zeichen CTRL-I (9) wird als Sequenz von Leerzeichen ausgegeben).

Beispiel: Ausgabe des Zeichens „B".

```
MVI C,4
MVI E,'B'
CALL 5
```

Funktion 5: Drucker-Ausgabe

Mit Funktion 5 kann ein Zeichen, das in Register E steht, an den Drucker gegeben werden. Beim mc-CP/M-Computer ist dies der Kanal SIO B, wenn zuvor im Monitor AL = L eingestellt wurde. Die Ausgabe ist ebenfalls transparent.

Funktion 6: Direkte Konsolen-Ein-/Ausgabe

In vielen früheren Anwenderprogrammen für CP/M finden sich seltsame Konstruktionen, die versuchen die Anfangsadresse des BIOS zu finden, um dann Code zu modifizieren, um so direkt irgendwohin zu gelangen. Dadurch sollte die Ein-/Ausgabe des BIOS verwertbar gemacht werden, da dort die I/O-Geräte direkt zugänglich sind. Ab CP/M, Version 2.0, gibt es die Funktion 6, mit der die Konsol-Vektoren direkt angesprungen werden können. Dazu erhält das Register E einen Parameter. Ist der Wert des Registers mit E = FF eingestellt, so wird eine Eingabe verlangt, ist der Wert in E ungleich FF, so soll eine Ausgabe dieses Wertes in E erfolgen. Bei einer Eingabe erscheint nach dem Aufruf im Register A entweder das Zeichen oder, falls keines da war, der Code 00. Damit wartet also die Routine nicht so lange, bis ein Zeichen eingegeben wurde, sondern dies muß man, wenn gewünscht, als Anwender selbst im Programm so einrichten.

Beispiel: Ausgabe eines Zeichens.

```
MVI C,6
MVI E,'Z'
CALL 5
```

Beispiel: Eingabe eines Zeichens.

```
WARTE: MVI C,6
       MVI E,OFFH
       CALL 5
       CPI 0
       JZ WARTE
       ; In A steht nun das Zeichen
```

Die Routine ist wegen der Parameterkonvention leider nicht ganz transparent. So kann das Zeichen FF nicht ausgegeben werden und das Zeichen 00 darf nicht als Eingabewert vorkommen.

Funktion 7: I/O-Byte holen

Bei CP/M ist auf Adresse 3 ein I/O-Byte definiert, das vom BIOS ausgewertet werden kann. Mit dieser Funktion wird der Wert dieser Speicherzelle in das Register A geholt.

Im mc-CP/M-Computer wird dieses Byte im BIOS nicht ausgewertet, da der Monitor selbst über ein solches Byte verfügt.

Beispiel: Holen des I/O-Bytes

```
MVI C,7
CALL 5
; in Register A steht der Wert
```

Funktion 8: Setzen des I/O-Bytes

Mit dieser Funktion wird der Inhalt der Zelle 3 verändert. Der neue Wert muß in Register E stehen. Mancher Leser wird sich fragen, warum es hier nicht einfacher ist, den Wert direkt in die Speicherzelle zu schreiben oder, wie im vorherigen Fall, einfach daraus zu lesen. Dies hängt mit der Philosophie von CP/M zusammen. Der BDOS-Aufruf ist eine klare Schnittstelle. Es gibt nämlich auch CP/M-Systeme, bei denen die TPA nicht bei 100 H anfängt. Der BDOS-Vektor und auch das I/O-Byte liegen dann nicht an den normalen Stellen. Programme die direkt Speicherzellen modifizieren, können dann nur mit Vorsicht an die neue Umgebung angepaßt werden. Ein BDOS-Aufruf dagegen läßt sich leicht umstellen, da nur eine einzige Adresse (neben der Programmanfangsadresse) geändert werden muß.

Beispiel: Setzen auf 0.

```
MVI C,8
MVI E,0
CALL 5
```

Beim mc-CP/M-Computer wird das I/O-Byte nicht ausgewertet. Ein Programm, das dies verlangt, läuft also auf dem mc-CP/M-Computer nicht wie vorgesehen; dies ließe sich aber leicht ändern.

Funktion 9: Ausgabe eines Strings

Mit Funktion 9 kann eine Zeichenkette auf der Konsole ausgegeben werden. Die Anfangs-Adresse der Zeichenkette wird im Registerpaar DE übergeben. Das Ende des Strings wird durch das Zeichen „$" angezeigt. Tabulatoren werden expandiert.

Beispiel: Ausgabe eines Textes.

```
       MVI C,9
       LXI D,TEXT
       CALL 5
       ...
TEXT: DB 'MC-Computer$'
```

Auch Zeichen wie CR, LF dürfen im Text vorkommen. Nur das Zeichen „$" bestimmt das Ende der Sequenz.

Funktion 10: Eingabe des Konsolen-Puffers

Sehr komfortabel ist die Funktion 10. Damit ist die Eingabe einer ganzen Textzeile möglich. Im Registerpaar DE wird die Startadresse des Textpuffers angegeben. Das erste Byte in diesem Puffer gibt an, wieviele Zeichen maximal eingelesen werden dürfen. Der Bereich liegt zwischen 1 und 255 (der Wert 0 bedeutet „kein Zeichen" und ist sinnlos). Nach dem Aufruf wird im zweiten Byte des Puffers die tatsächliche Zahl eingegebener Zeichen von der Funktion 10 eingetragen. Sie kann die Zahl im ersten Byte nicht übersteigen. An der dritten Position des Puffers beginnt der Textspeicher. Dahin werden die eingelesenen Zeichen transportiert. Der Puffer wird nicht aufgefüllt, so daß hinter dem letzten eingegebenen Zeichen zufällig gesetzte Zeichen folgen. Die Eingabe einer Zeile wird entweder bei Erreichen der Maximalzahl abgebrochen, oder wenn das Zeichen CR oder LF eingegeben wurde. Für Eingabe selbst gibt es eine Reihe von Kontrollzeichen, die zum zeilenorientierten Editieren der Eingabe dienen können:

RUB/ DEL	Das zuletzt eingegebene Zeichen wird gelöscht und auf der Konsole erneut ausgegeben.
CTRL-H	Das zuletzt eingegebene Zeichen wird gelöscht und auch auf der Konsole mit einem Leerzeichen überschrieben.

CTRL-C Wird diese Taste am Zeilenanfang eingegeben, so wird das CP/M-System neu gestartet (Warm-Boot).

CTRL-E Auf der Console wird ein CR-LF ausgegeben, diese Sequenz erscheint nicht im Puffer.

CTRL-R Die Zeile wird erneut auf dem Bildschirm ausgegeben. Diese Taste wird in Verbindung mit RUB/DEL gebraucht um die Zeile wieder in lesbare Form zu bringen, doch bei Datensichtgeräten ist CTRL-H besser.

CTRL-U Die aktuelle Zeile wird gelöscht, sie kann neu eingegeben werden.

CTRL-X Die aktuelle Zeile wird gelöscht, der Cursor geht hier auf die erste Position zurück. Diese Taste empfiehlt sich wieder bei Datensichtgeräten. Die erste Position ist dabei die Position bei der der Cursor zum Zeitpunkt des BDOS-Aufrufs mit dieser Funktion war.

CTRL-P Der Drucker-Kanal wird parallel zur Konsolen-Ausgabe geschaltet. Er kann durch erneutes CTRL-P wieder abgeschaltet werden. CTRL-P, einmal gesetzt, bleibt nach verlassen von Funktion 10 wirksam. Darüber hinaus berücksichtigen Funktionen 2 und 9 ein zuvor mit Funktion 10 gegebenes CTRL-P und geben ihre Ausgaben auch auf den Drucker-Kanal.

Beispiel für gepufferte Eingabe:

```
        MVI C,0AH
        LXI D,BUFFER
        CALL 5
        ; nach Eingabe wird hier fortgefahren

PUFFER: DB 40 ; MAX 40 Zeichen
        DB 0  ; HIER NACH AUFRUF ANZAHL
        DS 40 ; PUFFER FUER EINGABE
```

Funktion 11: Konsolen-Status holen

Eine Funktion, die mit den Aufrufen 1 und 2 zusammenarbeitet. Damit kann geprüft werden ob ein Zeichen eingegeben wurde. Im Register A erscheint der Code 0, wenn kein Zeichen anstand, sonst ein Wert ungleich 0 (i. a. der Wert 1 und nicht FF, wie im Handbuch angegeben). Das Zeichen wird vom BDOS schon eingelesen, jedoch wird es erst mit der Funktion 1 (oder 10) aus dem BDOS dem Aufrufer übermittelt. Die Funktion 11 kann aber mehrere Male hintereinander aufgerufen werden. Solange das Zeichen nicht durch die Funktion 1 oder 10 eingelesen wurde, bleibt der Status ungleich 0. Es gehen also keine Zeichen verloren. Beispiel: Warten auf ein Zeichen.

```
WARTE: MVI C,0BH
       CALL 5
       CPI 0
       JZ WARTE
       ...ggf. andere Operationen
       MVI C,1
       CALL 5
       ; nun ist das Zeichen im Akku
```

Damit sind die I/O-Funktionen für die Ein- und Ausgabe von Zeichen unter Programmkontrolle besprochen.

Die Floppy-Disk-Bedienung

Als Disk-Grundfunktionen kann man im Prinzip bei jedem Disk-Betriebssystem vier Funktionen unterscheiden: OPEN, READ, WRITE und CLOSE. Die Funktion OPEN dient dazu, im Betriebssystem die Zuordnung von einem Dateinamen zu den eigentlichen Blöcken dieser Datei herzustellen. Mit READ und WRITE erfolgen die Datenzugriffe, und mit CLOSE wird dem Betriebssystem gesagt, daß die Bearbeitung der Datei abgeschlossen ist, und die verwendeten Zwischenpuffer (wenn das CP/M-Directory verändert wurde) werden auf die Diskette zurückgeschrieben.

Der File-Control-Block (FCB)

Bei den meisten bisherigen BDOS-Funktionen wurden Parameter einfach in Register übertragen, oder es wurde eine einfache Pufferstruktur verwendet. Bei den Disk-Befehlen kommt ein spezieller Parameterblock, der File-Control-Block (kurz FCB, *Bild 1*) hinzu. Darin ist z. B. der Name der Datei festgelegt, und das BDOS hat dort seine Zwischenspeicher für Directory-Informationen usw. *Bild 1* zeigt die Aufteilung des FCB. Er besteht aus 36 Bytes (0...35). Das erste Byte gibt an, auf welchem Laufwerk die Datei liegen soll. Wird dort der Wert 0 abgelegt, so wird das aktuelle Laufwerk verwendet. Es kann entweder durch einen speziellen Aufruf des BDOS eingestellt werden oder ist einfach das zuletzt verwendete Laufwerk.

Wird an die Stelle 0 der Wert 1 gelegt, so wird auf jeden Fall das Laufwerk A verwendet, beim Wert 2 das Laufwerk B bis 16 für Laufwerk P.
Die Positionen 1 bis 8 sind die ersten acht Buchstaben des Dateinamens. Bit 7 muß auf Wert 0 gesetzt sein. 9 bis 11 stellen die Extension dar (z. B. „.COM"). Dabei besitzt die Position 9 und 10 eine Doppelfunktion: Ist in Position 9 das Bit 7 gesetzt, so ist die Datei als Nur-Lese-Datei schreibgeschützt; ist Bit 7 der Position 10 gesetzt, so wird die Datei bei DIR nicht mit ausgegeben, da sie eine SYS-Datei geworden ist.
Die Bytes 12 bis 15 werden vom System verwendet, genauso wie 16 bis 31. In Position 32 steht die aktuelle Record-Position, die beim nächsten Diskettenzugriff verwendet wird, wobei auch die Position 12 eine Rolle spielt: Vor dem Eröffnen müssen beide Positionen auf

den Wert 0 gesetzt werden, wenn sequentiell zugegriffen werden soll. Die Positionen 33 bis 35 sind für Random-Zugriffe vorgesehen.
Wird ein Programm vom CCP aus gestartet, indem der Name einer .COM-Datei angegeben wurde, gibt es die Möglichkeit, die Parameter, die hinter dem Befehl angegeben wurden, zu erhalten, um dem Anwenderprogramm zu ermöglichen, diese Parameter zu verarbeiten.
Ein Beispiel:
PROGRAM A:T.DAT B:A1.BIN /H
Die Datei mit dem Namen PROGRAM .COM wird auf die Stelle 100H (TPA) geladen. Nun gibt es mehrere Puffer, die für die Parameter verwendet werden. Zunächst einmal steht alles hinter dem Programmnamen ab Adresse 80H, dem voreingestellten Disk-Puffer. An der ersten Position steht die Anzahl der Zeichen, die als Parameter übertragen wurden; dann folgen die Zeichen, beginnend mit dem ersten Zeichen nach den Programmnamen, also bei uns:

80H	81H	82H	83H	84H	85H	86H	87H
14H	' '	'A'	':'	'T'	'.'	'D'	'A'

88H	89H	8AH	8BH	8CH	8DH	8EH	8FH
'T'	' '	'B'	':'	'A'	'1'	'.'	'B'

90H	91H	92H	93H	94H
'I'	'N'	' '	'/'	'H'

Eine Besonderheit aber ist, daß auch ein FCB aufgebaut wird. Der erste Parameter, also hier A:T.DAT, wird nämlich als FCB auf Adresse 5CH abgelegt; falls wie hier ein zweiter Parameter vorhanden war, wird dieser, als Dateiname interpretiert, auf Adresse 6CH abgelegt (gemäß dem FCB-Format). Es werden dazu die Buchstaben des Namens in die Felder 1 bis 11 verteilt und mit Leerzeichen angefüllt. Die Position 0 wird mit der Laufwerksnummer belegt, falls eine angegeben war. Da der Bereich 6CH den Bereich 5CH überlappt, muß der Bereich 6CH vom Benutzer in einen anderen Speicherbereich geschafft werden, bevor eine Disk-Funktion mit 5CH als FCB ausgeführt werden kann.
Nun aber zu den FCB-Funktionen, wobei auch noch ein paar andere Grundfunktionen enthalten sind.

Byte	Abkürzg.	Beschreibung
0	dr	Drivecode. 0=default Laufwerk 1=Laufwerk A, 2=B ... 16=P
1 bis 8	f1..f8	ASCII-Name der Datei. Bit 7 = 0
9 10 11	t1,t2,t3	Datei-Name (Extension) Bit 7 hat eine Spezialbedeutung. t1' gesetzt dann Read/Only-Datei t2' gesetzt dann SYS-Datei
12	ex	Die aktuelle Erweiterungsnummer bei Dateien die mehr als einen Direktoryeintrag benötigen. Normalerweise wird der Wert auf 0 gesetzt, durch das System nach Zugriffen aber im Bereich 0..31.
13	s1	für interne Verwendung durch das System
14	s2	für interne Verwendung reserviert,
15	rc	Record Zähler. Im Bereich 0..128.
16 bis 31	d0..dn	wird durch das System definiert und ist für internen Gebrauch reserviert.
32	cr	Die aktuelle Recordnummer die bei einem sequenziellen Lese- oder Schreibzugriff verwendet wird. Vom Benutzer am Anfang auf den Wert 0 zu setzen.
33 34 35	r0,r1,r2	optionale Random-Record-Position im Bereich 0 bis 65535 (Überlauf nach r2). Lsb ist r0.

Funktion 12: Versionsnummer angeben

Da es verschiedene CP/M-Ausführungen auf dem Markt gibt (MP/M, CP/M 2.2 oder 1.4, neuerdings 3.0) ist es für Anwenderprogramme ganz praktisch, die Version abzuprüfen und so z. B. bei der Version 1.4 einen RANDOM-Zugriff zu emulieren, der erst ab Version 2.0 verfügbar wurde. Die Versionsnummer erscheint im Register HL; bei MP/M ist H=1, sonst H=0. Bei früheren Versionen als 2.0 ist L=0. Bei CP/M 2.2 ist L=22H.
Der Aufruf erfolgt mit:

```
MVI C,0CH
CALL BDOS
```

Das Ergebnis steht dann in HL.

Funktion 13: Rücksetzen des Disk-Systems

Programme, bei denen die Disketten ohne Warm-Boot ausgetauscht werden müssen, benötigen diese Funktion. Sie wird beim Einlegen der neuen Diskette angewendet und verhindert den R/O-Fehler, der sonst auftreten würde.

Funktion 14: Laufwerk selektieren

Im Register E steht das neue Laufwerk, das als aktuelles Laufwerk ausgewählt werden soll (E=0 bei Laufwerk A. E=1 bei Laufwerk B usw.). Steht im FCB bei Diskzugriffen der Wert 0 als erstes Byte, so wird das hier eingestellte Laufwerk verwendet.

Funktion 15: Eröffnen einer Datei

Im Registerpaar DE steht die Adresse des verwendeten FCBs. Dort müssen der Name der Datei und das Laufwerk eingetragen sein. EX und S1 müssen normalerweise den Wert 0 besitzen. Das Fragezeichen (?) kann als Ersatzzeichen an allen Stellen des Namens vorkommen. Wird eine Datei gefunden, die mit dem Namen übereinstimmt, so wird der FCB mit den entsprechenden Daten gefüllt. War die OPEN-Funktion erfolgreich, so wird im Register A der Wert 0, 1, 2 oder 3 geliefert. Wurde die Datei nicht gefunden, so wird der Wert FFH im Akku geliefert.

Funktion 16: Schließen einer Datei

Das Registerpaar DE hat die Adresse des FCB. Die Directory-Information wird nach diesem Befehl auf die Diskette zurückgeschrieben. Nach dem Aufruf gibt es im Register A die Werte 0 bis 3 bei Erfolg; wenn der Dateiname nicht gefunden wurde, erscheint der Wert FFH.

```
0000'   C3 0010'            jmp start

0005                        bdos    equ     5
005C                        fcb     equ     5ch     ;default buffer
0080                        dma     equ     80h     ;default dma

0003'                       co:
0003'   C5                          push b
0004'   D5                          push d
0005'   E5                          push h
0006'   59                          mov e,c
0007'   0E 02                       mvi c,2
0009'   CD 0005                     call bdos
000C'   E1                          pop h
000D'   D1                          pop d
000E'   C1                          pop b
000F'   C9                          ret



0010'                       start:                  ;angabe mit wildcards ist moeglich
0010'   0E 11                       mvi c,11h       ;search for first
0012'   11 005C                     lxi d,fcb
0015'   CD 0005                     call bdos       ;dma ist default ziel
0018'   FE FF                       cpi 0ffh        ;dann keine daten mehr da
001A'   CA 0000                     jz 0            ;ende des programms
                                    ;
001D'                       loop:                   ;a=0,1,2,3  daraus adresse berechnen
001D'   21 0080                     lxi h,dma       ;repeat
0020'   11 0020                     lxi d,32        ;displacement
0023'                       loop1:                  ;while a<>0
0023'   B7                          ora a
0024'   CA 002C'                    jz skip
0027'   19                          dad d
0028'   3D                          dcr a
0029'   C3 0023'                    jmp loop1       ;endwhile
002C'                       skip:
                                    ; ausgabe des namens und extension
002C'   23                          inx h           ;name
002D'   06 08                       mvi b,8
002F'                       loop2:                  ;do b,8
002F'   4E                          mov c,m
0030'   CD 0003'                    call co
0033'   23                          inx h
0034'   05                          dcr b
0035'   C2 002F'                    jnz loop2       ;enddo
                                    ; extension
0038'   0E 2D                       mvi c,'-'       ;hier mit minus trennen
003A'   CD 0003'                    call co
003D'   06 03                       mvi b,3         ;ausgeben
003F'                       loop3:                  ;do b,3
003F'   4E                          mov c,m
0040'   CD 0003'                    call co
0043'   23                          inx h
0044'   05                          dcr b
0045'   C2 003F'                    jnz loop3       ;enddo
0048'   0E 0D                       mvi c,0dh
004A'   CD 0003'                    call co
004D'   0E 0A                       mvi c,0ah
004F'   CD 0003'                    call co
0052'   0E 12                       mvi c,12h       ;search for next
0054'   11 005C                     lxi d,fcb
0057'   CD 0005                     call bdos
005A'   FE FF                       cpi 0ffh        ;until a=0ffh
005C'   C2 001D'                    jnz loop        ;warm boot dann
005F'   C3 0000                     jmp 0

                                    end
```

Bild 2. Ein neues DIR-Kommando

Funktion 17: Suche nach dem ersten Eintrag

Um die Directory-Information auch dem Benutzer zugänglich zu machen, gibt es die Funktion 17. DE zeigt auf den FCB. Nach dem Aufruf steht im Akku der Wert FFH, wenn keine Datei gefunden wurde, sonst der Wert 0, 1, 2 oder 3. Dieser Wert wird mit 32 multipliziert und ergibt das Displacement (relative Versatzadresse) auf den Puffer, an dessen Position die Directory-Information geladen wurde. Der Puffer ist normalerweise auf 80H voreingestellt und kann mit einer BDOS-Funktion auch beliebig gesetzt werden. Ein Fragezeichen im Namen läßt jeden Buchstaben zu; an der Position 0 im FCB erlaubt es, auf dem aktuellen Laufwerk zu suchen. Alle User-Bereiche werden durchsucht.

```
B>
B>c:dirneu *.STR
EXAMPLE -STR
STRUZ80 -STR
STRU8080-STR
EX1     -STR

B>c:dirneu S*.COM
STRUZ80 -COM
STRU8080-COM

B>
```

Bild 3. Ausgabeformat des neuen DIR-Programms

```
0000'   C3 0010'            jmp start              ;hauptprogramm


0005                        bdos    equ     5

                            ; hier werden die defaultbuffer verwendet

005C                        fcb     equ     5ch    ;buffer durch ccp angelegt
0080                        dma     equ     80h    ;lese buffer default


0003'                       co:                    ;zeichen in c-register ausgeben
0003'   C5                          push b
0004'   D5                          push d
0005'   E5                          push h         ;retten sonst undef
0006'   59                          mov e,c
0007'   0E 02                       mvi c,2                ;funktion co
0009'   CD 0005                     call bdos
000C'   E1                          pop h
000D'   D1                          pop d
000E'   C1                          pop b
000F'   C9                          ret


0010'                       start:
0010'   0E 0F                      mvi c,0fh      ;open funktion
0012'   11 005C                    lxi d,fcb
0015'   CD 0005                     call bdos      ;0ffh bedeutet fehler
0018'   3C                          inr a          ;trick
0019'   C2 0024'                    jnz start1     ;ok weiter
001C'   0E 3F                      mvi c,'?'
001E'   CD 0003'                    call co        ;fehler datei nicht da
0021'   C3 0000                     jmp 0          ;warm start
                            ;
0024'                       start1:
0024'   0E 14                      mvi c,14h      ;lese befehl
0026'   11 005C                    lxi d,fcb      ;
0029'   CD 0005                     call bdos      ;dma ist default
002C'   B7                          ora a          ;=0 dann kein fehler
002D'   C2 0041'                    jnz ende1      ;keine daten mehr da
0030'   06 80                      mvi b,128      ;ausgabe des buffers auf die console
0032'   21 0080                    lxi h,dma
0035'   4E                  loop:   mov c,m
0036'   CD 0003'                    call co        ;ausgeben
0039'   23                          inx h
003A'   05                          dcr b
003B'   C2 0035'                    jnz loop
003E'   C3 0024'                    jmp start1     ;aussenschleife
                            ;
0041'   0E 10               ende1:  mvi c,10h      ;datei schliessen
0043'   11 005C                    lxi d,fcb      ;bei lesezugriffen nicht
0046'   CD 0005                     call bdos      ;unbedingt noetig
0049'   C3 0000                     jmp 0          ;warm boot

                                    end
```

Bild 4. Lesen aus einer Datei

Funktion 18: Suchen nach nächstem Eintrag

Ein Name mit Fragezeichen kann mehreren gültigen Namen auf der Diskette entsprechen. Durch die vorhergehende Funktion wird das erste Auftreten gesucht, hier wird nun der nächste Eintrag geliefert. Werden keine Einträge mehr gefunden, so wird im Akku der Wert FFH geliefert. *Bild 2* zeigt ein Beispiel für ein Programm, das das Inhaltsverzeichnis einer Diskette ausgeben soll, wie es auch der Befehl DIR tut. Das Programm habe z. B. den Namen DIRNEU .COM. Dann werden durch Eingabe des Befehls DIRNEU *.* alle Dateien auf der Diskette ausgegeben. Die Angabe *.* wird vom CCP durch ? ersetzt und im FCB abgelegt. *Bild 3* zeigt ein Ausgabebeispiel.

Funktion 19: Löschen einer Datei

Im DE-Registerpaar steht wieder die Adresse des FCB. Die Datei mit dem gefundenen Namen wird entfernt. Im Akku steht FFH, wenn es keine Datei mit dem angegebenen Namen, der auch Fragezeichen enthalten kann, gab.

```
0000'   C3 0003'            jmp start

0005                        bdos    equ     5


                            ; sektoren mit A..Z beschreiben
                            ; pro sektor ein Buchstabe

0003'                       start:
0003'   0E 13                       mvi c,13h       ;loeschen falls schon da
0005'   11 004E'                    lxi d,fcb
0008'   CD 0005                     call bdos
000B'   0E 16                       mvi c,16h       ;creieren
000D'   11 004E'                    lxi d,fcb
0010'   CD 0005                     call bdos
0013'   FE FF                       cpi 0ffh
0015'   CA 0000                     jz 0            ;direktory full
0018'   0E 1A                       mvi c,1ah
001A'   11 0072'                    lxi d,dma       ;dma adresse neu setzen
001D'   CD 0005                     call bdos
0020'   0E 41                       mvi c,'A'       ;start fuell wert
0022'   21 0072'           loop:    lxi h,dma       ;repeat
0025'   06 80                       mvi b,128
0027'   71                 loop1:   mov m,c         ;do b,128
0028'   23                          inx h
0029'   05                          dcr b
002A'   C2 0027'                    jnz loop1       ;enddo
002D'   C5                          push b          ;retten
002E'   0E 15                       mvi c,15h
0030'   11 004E'                    lxi d,fcb
0033'   CD 0005                     call bdos
0036'   FE FF                       cpi 0ffh
0038'   CA 0000                     jz 0            ;disk full
003B'   C1                          pop b
003C'   0C                          inr c
003D'   79                          mov a,c
003E'   FE 5B                       cpi  'Z'+1      ;until c='Z'+1
0040'   C2 0022'                    jnz loop        ;danach schliessen
0043'   0E 10                       mvi c,10h
0045'   11 004E'                    lxi d,fcb
0048'   CD 0005                     call bdos       ;close
004B'   C3 0000                     jmp 0           ;ende



004E'   00                  fcb:    db 0            ;aktuelles laufwerk verwenden
004F'   44 41 54 45                 db 'DATEN   DAT'
0053'   4E 20 20 20
0057'   44 41 54
005A'   00 00 00 00                 db 0,0,0,0
005E'                               ds 16           ;d0..dn
006E'   00                          db 0            ;cr
006F'   00 00 00                    db 0,0,0        ;r0,r1,r2 hier dummy


0072'                       dma:    ds 128          ;buffer
                                                            end
```

Bild 5. Schreiben in eine Datei

Funktion 20: Sequentielles Lesen

Im DE-Register steht die Adresse des FCB. Es werden 128 Byte zur aktuellen DMA-Adresse gelesen. Die DMA-Adresse ist auf 80H voreingestellt. Die Position CR im FCB gibt den nächsten zu lesenden Record an. Der Wert wird automatisch erhöht; bei Überlauf wird das Feld EX erneuert. Ist das Dateiende erreicht, so erscheint der Wert FFH im Register nach dem Aufruf, bei erfolgreichem Lesevorgang dagegen Null.
Bild 4 zeigt ein Beispielprogramm. Es soll die Funktion TYPE nachgebildet und der Inhalt einer Datei in Textform auf der Konsole ausgegeben werden. Mit TYPE DATEI.TXT wird die Datei DATEI.TXT ausgegeben. Existiert die angegebene Datei nicht, so wird ein Fragezeichen auf der Konsole ausgegeben.

Funktion 21: Sequentielles Schreiben

Im Registerpaar DE steht die Adresse des FCB. Hier wird der Inhalt des Puffers auf der Diskette abgelegt, die Felder CR und EX werden automatisch auf neuen Stand gebracht. Bei Erfolg wird im Akku Null stehen, bei einer vollen Diskette ein Wert ungleich Null. Nach erfolgtem Schreiben muß die CLOSE-Funktion (16) aufgerufen werden, um die Directory-Information auch zurückzuschreiben.

Funktion 22: Datei erzeugen

Soll eine Datei neu angelegt werden, so wird anstelle der Open-Funktion, die ja nur bei einer schon existierenden Datei arbeitet, Funktion 22 aufgerufen. Hier darf aber die Datei zuvor nicht vorhanden sein, und es empfiehlt sich, vor diesem Aufruf sicherheitshalber die Lösch-Funktion aufzurufen. DE wird wieder mit der FCB-Adresse vorbelegt. FFH steht im Akku, wenn kein Directory-Platz mehr vorhanden war, sonst der Wert 0 bis 3. Ein nachfolgender OPEN-Befehl ist nicht mehr nötig.
Bild 5 zeigt ein Beispiel, bei der eine Datei mit Daten gefüllt werden soll. Dazu wird die Datei vorher gelöscht (falls sie schon da war) und anschließend neu erzeugt. Dann werden dort Daten abgelegt. Hier wird die DMA-Adresse mit der Funktion 26 (die wir noch nicht besprochen hatten) auf einen anderen Wert als die Voreinstellung von 80H gesetzt. Nach dem Aufruf des Programms steht im ersten Sektor das Zeichen A, im zweiten das Zeichen B usw., im letzten das Zeichen Z.

Funktion 23: Datei umbenennen

Mit diesem BDOS-Aufruf ist es möglich, den Namen einer Datei neu festzulegen. Dazu wird im Registerpaar DE die Adresse eines FCBs angegeben. Die ersten 16 Bytes enthalten im üblichen FCB-Format den alten Namen der Datei. Das erste Byte enthält den Code für das Laufwerk (0 = aktuelles 1 = Laufwerk A etc.). In den nächsten 16 Bytes wird der neue Name gegeben. Bild 6 zeigt ein Beispiel für den Aufruf dieses Befehls. Im Akku wird ein Wert 0, 1, 2, 3 geliefert, wenn die Operation erfolgreich war, sonst der Wert 0FFh, wenn die Datei nicht existierte.

Funktion 24: LOGIN-Vektor holen

Damit läßt sich feststellen, welche Laufwerke schon angesprochen wurden. Das Registerpaar HL enthält 16 Bit. Bit 0 des Registers L entspricht dem Laufwerk A, Bit 1 dem Laufwerk B, ..., Bit 7 von Register H dem Laufwerk P.
Ist das Bit auf 0 gesetzt, so wurde das Laufwerk noch nicht angesprochen, sonst ja. Register A und L enthalten denselben Wert um mit alten Versionen aufwärtskompatibel zu sein.

Funktion 25: Aktuelles Laufwerk melden

Im Register A steht anschließend der Code für das aktuelle Laufwerk. Der Bereich liegt zwischen 0 und 15, was den Laufwerken A bis P entspricht.

Funktion 26: DMA-Adresse festlegen

Bei Zugriffen, wie READ, WRITE, wird für den Datentransfer immer ein Buffer verwendet. Die Adresse dieses Buffers (DE) läßt sich mit Funktion 26 festlegen. Der Buffer wird von CP/M auf die Adresse 80h voreingestellt.

Funktion 27: Bit-Map-Adresse holen

Dies ist eine Funktion, die es ermöglicht, die Adresse der Bit Map des aktuellen Laufwerks zu holen. HL enthält den Adreßwert. Diese Funktion wird nur von ein paar Spezialprogrammen benötigt und wird von normalen Benutzerprogrammen nicht gebraucht.

Funktion 28: Diskette auf Schreibschutz setzen

Schreiben auf das aktuelle Laufwerk wird damit verhindert. Ein Versuch wird mit der Fehlermeldung BDOS ERR ON ×: R/O quittiert, wobei × für das aktuelle Laufwerk steht.
Ein Schreibschutz erfolgt automatisch bei einem Schreibversuch, wenn zwischendurch die Diskette gewechselt wurde und damit die Bitmap nicht mit der neuen Diskette übereinstimmt. Nur durch die Funktion RESET nach dem Einlegen der neuen Diskette läßt sich dies verhindern.

Funktion 29: READ/ONLY – Vektor holen

Im HL-Register wird ein Bit-Code übergeben, der genauso wie Funktion 24 die Laufwerke codiert, also Bit 0 Register L ist Laufwerk A. Das Bit ist gesetzt, wenn das Laufwerk mit R/O gesperrt wurde, dies kann dabei durch die Funktion 28 oder durch einen Diskettenwechsel geschehen sein.

Funktion 30: Datei-Attribute setzen

Das Registerpaar DE zeigt auf den FCB. Dieser muß einen gültigen Dateinamen enthalten. Nun haben wir schon früher einmal von den Bits t1', t2' gehört, die im Namen gesetzt sein können, um System oder R/O-Dateien zu kennzeichnen. Mit diesem Befehl können sie auf die Diskette geschrieben werden. Die Bits f1' bis f4' können zusätzlich vom Benutzer verwendet werden, f5' bis f8' und t3' sind für weitere CP/M-Versionen reserviert. Zur Erinnerung, f1' kennzeichnet die Byteposition 1 im FCB und dabei das Bit 7.

```
0000'
0000'   11 000B'                    ld de,fcbn      ;adresse file control
0003'   0E 17                       ld c,17h        ;rename funktion
0005'   CD 0005                     call 5          ;ausfuehren
                                                    ;akku =0,1,2,3 wenn ok
                                                    ;sonst 0ffh
0008'   CD 0000                     call 0          ;ende des programms

000B'                       fcbn:
000B'   00 41 4C 54         defb 0,'ALT     ','DAT',0,0,0,0
000F'   20 20 20 20
0013'   20 44 41 54
0017'   00 00 00 00
001B'   00 4E 45 55         defb 0,'NEU     ','NEU',0,0,0,0
001F'   20 20 20 20
0023'   20 4E 45 55
0027'   00 00 00 00
```

Bild 6. Umbenennen einer Datei

```
jp start

bdos    equ     5

; versch. Unterprogramme
;

ci:     push hl
        ld c,1
        call bdos       ;zeichen lesen
        pop hl
        ret
co:     push hl
        ld c,2          ;e=wert
        call bdos
        pop hl
        ret

print:  push hl
        ld c,9          ;de -> string
        call bdos
        pop hl
        ret

meldg:
        defb 0dh,0ah,'Eingabe der Recordnr:$'

zahlein:
        ld de,meldg
        call print
        ld hl,0         ;start wert
        call ci         ;erstes
        cp '0'
        jr c,zahlein    ;neuer versuch
        cp '9'+1
        jr nc,zahlein
wdh:    ld d,h
        ld e,l
        add hl,hl
        add hl,hl
        add hl,de
        add hl,hl       ;hl :=  hl * 10
        and 0fh         ;zahl
        ld c,a
        ld b,0
        add hl,bc       ;hl := hl + neue zahl
        call ci         ;zahl
        cp '0'
        jr c,finzahl
        cp '9'+1
        jr nc,finzahl
        jr wdh          ;erneut ausfuehren
finzahl:ret             ;hl = zahlenwert


msgb:
defb 0dh,0ah,'Alter Inhalt:',0dh,0ah,'$'

bufaus:                 ;alten Inhalt ausgeben
        ld de,msgb
        call print
        ld hl,buffer    ;zeichenbuffer
        ld b,128        ;max count
lopbuf: ld e,(hl)       ;zeichen holen
        push bc
        call co
        pop bc
        inc hl
        djnz lopbuf     ;alle 128 zeichen
        ret             ;dann fertig

msgb1:
defb 0dh,0ah,'Neuen Inhalt eingeben',0dh,0ah,'$'

bufein:                 ;neuen Inhalt in
                        Buffer einlesen
        ld de,msgb1
        call print
        ld hl,buffer    ;erst mit leerzeichen
                        vorbelegen
        ld de,buffer+1
        ld bc,128-1
        ld (hl),' '     ;leerzeichen
        ldir
        ld hl,buffer    ;nun neueingabe
        call ci
        cp 0dh
        jr z,carset
        ld b,128        ;max count
lopein: and 7fh
        cp 1            ;CTRL-A beendet
        jr z,carset
        cp ' '          ;> dann ok sonst ende
        jr c,einfin
        ld (hl),a
        inc hl
        push bc
        call ci
        pop bc
        djnz lopein
einfin: xor a           ;kein carry
        ret             ;ok fertig
carset: scf
        ret             ;ende bedingung

msgw:
defb 0dh,0ah  Buffer neu belegt wird
geschrieben',0dh,0ah,'$'


start:
        ld sp,stack
        ld c,26         ;neue dma - adresse
        ld de,buffer
        call bdos
        ld c,15         ;eroeffnen der Datei
        ld de,fcbn      ;wenn schon da
        call bdos       ;dann ok sonst anlegen
        cp 0ffh         ;=ff dann nicht da
        jr nz,weiter
        ld c,22         ;anlegen noetig
        ld de,fcbn
        call bdos       ;nun muss es klappen
        cp 0ffh         ;bei fehler verlassen
        call z,0        ;hier meldung moeglich
weiter: call zahlein    ;nach hl fuer rndpos
        ld (rndpos),hl  ;nur r0,r1 belegen
        ld c,33         ;versuch zu lesen
        ld de,fcbn
        call bdos       ;wenn schon da anzeigen
        cp 0            ;=0 dann ok
        jr nz,neuan     ;neu anlegen
        call bufaus     ;ausgabe des alten buffers
neuan:  call bufein     ;neuer eingeben nach buffer
        cp 0dh
        jr z,weiter     ;nicht zurueckschreiben
        cp 1
        jr z,finale
        ld de,msgw
        call print
        ld c,34         ;wenn nicht cr
        ld de,fcbn      ;zurueckschreiben
        call bdos
        cp 0
        jr z,weiter     ;das ganze immer weiter so.
finale:
        ld c,16         ;close
        ld de,fcbn
        call bdos
        call 0          ;bei fehler write



fcbn:
        defb 0,'RNDDAT  ','TXT',0,0,0,0
        defs 16
        defb 0          ;cur rec
rndpos: defb 0,0,0              ;rnd zeiger hier
;               lsb .. msb
buffer: defs 128                ;zeichenbuffer
        defs 200                ;stack
stack:  defs 1
end
```

Bild 7. Beispiel für die RND-Befehle

Funktion 31: Adresse der Laufwerksparameter holen

Im Registerpaar HL wird die Adresse geliefert. Es ist die Adresse des Disk-Parameter-Blocks vom aktuellen Laufwerk im BIOS. Spezialprogramme können somit darauf direkt zugreifen.

Funktion 32: USER-Code setzen, holen

Steht im Register E der Wert OFFh, so steht nach dem Aufruf des BDOS im Register A der aktuelle USER-Code. Steht im Register E vor dem Aufruf ein Wert ungleich OFFh, so wird der neue Benutzerbereich auf diesen Wert gesetzt. Der Wert kann dabei im Bereich 0 bis 31 liegen. Der Benutzerbereich ermöglicht es, Dateien zu Gruppen zusammenzufassen und dennoch auf einer Diskette zu halten. Mit dem CCP-Befehl USER kann der Benutzerbereich auch gewechselt werden.

Funktion 33: Lesen mit wahlfreiem Zugriff

Diese Funktion arbeitet ähnlich zu der Funktion des sequentiellen Lesens, jedoch kann mit einer 24-Bit-Adresse eine RND-Position vorgegeben werden (dabei darf der Bereich jedoch nur zwischen 0 und 65535 liegen. Das Registerpaar DE enthält die Adresse des FCBs. Im FCB wird die gewünschte RND-Position bei der Byte-Position 33, 34, 35 abgelegt, wobei die Position 33 das LSB enthält und 35 normalerweise auf 0 liegt. Der FCB faßt dann 36 Bytes. Bei mehrfachen Zugriffen auf die Datei ohne den FCB zu ändern, wird immer der gleiche Record angewählt, dies unterscheidet diese Funktion vom sequentiellen Lesen. Bei einem Wechsel zum SEQ-Lesen wird der letzte Record erneut gelesen. Interessant bei der RND-Betriebsart ist die Möglichkeit, virtuelle Dateigrößen zu verwenden, so müssen nicht alle Recordpositionen zwischen niedrigstem und höchstem Wert auch tatsächlich existieren. Im Akku steht nach dem Aufruf ein Fehlercode:
0 = keine Fehler
1 = Record mit unbeschriebenen Daten wurde gelesen
3 = aktuelle Extension kann nicht geschlossen werden
4 = Versuch eine unbeschriebene Extension anzuwählen
6 = Versuch über das Diskende zu positionieren
(bei CP/M Vers 2.0, wenn r2 (Bytepos 35) ungleich 0 ist.)

Funktion 34: Schreiben mit wahlfreiem Zugriff

Im DE-Register ist die FCB-Adresse. Der Record mit der angegebenen Position (33, 34, 35) wird geschrieben und angelegt, Extensionen werden ggf. eröffnet. Die Fehlermeldungen sind identisch zum READ-Befehl, mit folgenden Erweiterungen:
5 = eine neue Extension konnte nicht angelegt werden.
Bild 7 zeigt ein Beispielprogramm mit den RND-Funktionen. Es ist damit möglich eine RND-Datei anzulegen und einzelne Record-Positionen zu beschreiben. Das Programm wird durch CTRL-A bei der Eingabe einer neuen Zeile gestoppt. *Bild 8* zeigt ein Aufrufbeispiel. Wird eine leere Zeile eingegeben, so wird der alte Inhalt nicht gelöscht.

Zu beachten ist der Umstand, daß nicht alle nicht beschriebenen Record-Positionen auch als solche erkannt werden. Im CP/M werden normalerweise immer 1024 Bytes auf einmal angelegt. Wird also ein Record in diesem Bereich angelegt, so ist natürlich der ganze Bereich gültig, obwohl darin evtl. ungültige Daten stehen. In einem Benutzerprogramm muß dieser Tatsache natürlich Rechnung getragen werden, in unserem Programm wird der Inhalt einfach auf die Console ausgegeben, auch wenn es sich um einen ungültigen Record handelt. Ist die Diskette vorher formatiert worden, und nur die RND.COM-Funktion vorhanden, so sind alle Records mit 0e5h gefüllt, die nicht gültig sind.

Funktion 35: Berechnen der Datei-Größen

Das Registerpaar DE enthält die Adresse des FCBs. Nach dem Aufruf sind die Positionen 33, 34, 35 im FCB auf den Record nach dem zuletzt vorhandenen eingestellt. Mit diesem Befehl ist es z. B. möglich direkt hinter einem geschriebenen Datenfeld anzuschließen und von da an weiterzuschreiben.

Funktion 36: Random-Record setzen

Registerpaar DE enthält die Adresse des FCBs. Die Record-Nr. ist zuvor gesetzt worden. Nach diesem Aufruf werden die anderen Felder definiert, so daß z. B. mit einem sequentiellen Zugriff weitergearbeitet werden kann.

Damit sind alle BDOS-Operationen besprochen. Über CP/M gibt es noch viel zu sagen und in der Zeitschrift mc wird das auch nach und nach noch getan werden. So gibt es bereits Versionen CP/M 86 (für den 8086/8088) und CP/M 68K (für den 68000). Ferner gibt es die Multiuser-Betriebssysteme MP/M und CP/NET als Netzwerk-System. Neuerdings gibt es auch eine GSX-Schnittstelle für ein Graphik-Interface (mit einem GDOS und GIOS-Teil) über die es auch zu berichten lohnt. An das GSX-System kann dann die GKS-Schnittstelle aufgesetzt werden, die im Level OA bereits als GKS-KERNEL auf CP/M verfügbar ist.

Ein CP/M-Kurzhandbuch, bestehend aus einer Übersicht über alle CP/M-Kommandos und der Beschreibung der BDOS-Aufrufe, ist vom Franzis-Software-Service erhältlich. Das Handbuch besitzt 42 Seiten und kostet 12,– DM. Es ist in deutscher Sprache geschrieben.

```
B>
B>rnd

Eingabe der Recordnr:1
Neuen Inhalt eingeben
test daten rec 1
Buffer neu belegt wird geschrieben

Eingabe der Recordnr:5000
Neuen Inhalt eingeben
rec nr 5000
Buffer neu belegt wird geschrieben

Eingabe der Recordnr:1
Alter Inhalt:
test daten rec 1
Neuen Inhalt eingeben

Eingabe der Recordnr:1
Alter Inhalt:
test daten rec 1
Neuen Inhalt eingeben

Eingabe der Recordnr:200
Neuen Inhalt eingeben
rec 200
Buffer neu belegt wird geschrieben

Eingabe der Recordnr:0
Alter Inhalt:
0
Neuen Inhalt eingeben
rec 0
Buffer neu belegt wird geschrieben

Eingabe der Recordnr:1
Alter Inhalt:
test daten rec 1
Neuen Inhalt eingeben

B>a:stat rnddat.txt

 Recs  Bytes  Ext Acc
   84     3k    3 R/W B:RNDDAT.TXT
Bytes Remaining On B: 161k

B>
B>
```

Bild 8. Aufruf des RND-Programms

Siegfried Langer:

Die Schnittstelle RS-232 – Beschreibung und Anwendung

Sobald ein Rechner zum System wachsen soll, besteht die Notwendigkeit, weitere Einheiten (Input/Output Units und Speicher) anzuschließen. Die universellste Möglichkeit stellen genormte Schnittstellen dar. Über solch ein verbindliches Interface können auch Einheiten verschiedener Hersteller angeschlossen werden. Neue Dienste der Deutschen Bundespost und anderer Postverwaltungen wie Bildschirmtext (BTX) werden auch den Amateuren und kleinen kommerziellen Anwendern die Möglichkeiten der Datenfernverarbeitung zu interessanten Preisen ermöglichen. Zugriff zu Datenbanken und Programmen, die bisher Großrechnern vorbehalten waren, rücken in den Bereich des möglichen, ja sie werden wahrscheinlich übliche Erweiterungen der eigenen Rechnermöglichkeiten darstellen.

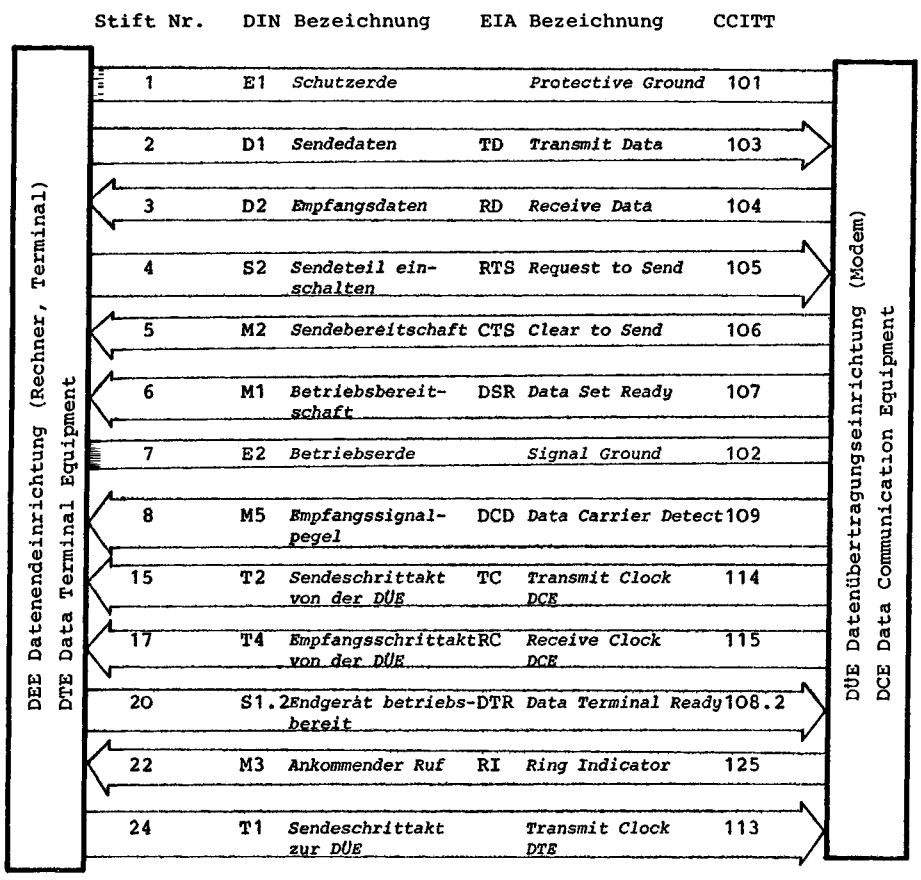


Bild 1. Schnittstellenleitungen nach DIN 66020, EIA-RS-232C und CCITT-V.24. Es sind nicht alle Leitungen aufgeführt

Der erste Teil dieses Aufsatzes beschreibt die stark verbreitete Schnittstelle RS-232 bzw. V.24. In der kommerziellen Datenverarbeitung verbindet sie meist eine Datenstation mit einem Modem zur Datenfernübertragung. Der zweite Teil beschreibt Möglichkeiten, verschiedene Geräte über diese Schnittstelle zu verbinden, ohne Eingriffe in die Geräte vornehmen zu müssen.

Die Schnittstelle RS-232C

Schnittstellen sind definierte Übergänge zwischen Komponenten eines Systems. Für den Prozessor bilden sie die Verbindung zur Außenwelt. Zahlreiche Kleincomputer sind bereits mit einer Schnittstelle RS-232C ausgerüstet, oder diese kann nachgerüstet werden. Es handelt sich um eine serielle Schnittstelle, die üblicherweise zwischen Datenendeinrichtung (DEE) und Datenübertragungseinrichtung (DÜE), auch Modem genannt, liegt.
Die Norm RS-232C ist eine US-Industrienorm nach EIA (Electronic Industries Associates). Das internationale Pendant nach CCITT (Comite Consultativ International Telegrafique et Telefonique) ist die Schnittstellendefinition V.24/V.28. V.24 beschreibt die funktionellen, V.28 die elektrischen Eigenschaften.
Bild 1 führt einige der Schnittstellenleitungen und deren Bezeichnung in den verschiedenen Normen auf. Nachstehend werden die EIA-Bezeichnungen verwendet.

Die Funktion der Schnittstellenleitungen

Die Kommunikation zwischen DEE und DÜE geschieht über Daten-, Takt-, Melde- und Steuersignale.
TD Sendedaten: Der DÜE werden über diese Leitung die digitalen Datensignale zugeführt.
RD Empfangsdaten: Datensignale von der DÜE zur DEE.

RTS Sendeteil einschalten: Mit diesem Signal wird die DÜE aufgefordert auf Sendebetrieb umzuschalten (Sender einschalten).
CTS Sendebereitschaft: Sobald die DÜE sendebereit ist, meldet sie dies mit dieser Leitung (Antwort auf RTS).
DSR Betriebsbereitschaft (DÜE): Ein aktives Signal auf dieser Leitung zeigt an, daß die DÜE betriebsbereit und mit dem Datenübertragungskanal verbunden ist.
DCD Empfangssignalpegel: Die DÜE empfängt gültige Signale (ausreichender Pegel). Der DEE wird angezeigt, daß empfangen wird.
TC Sendeschrittakt von der DÜE: Mit dieser Leitung wird der DEE der Sendeschrittakt zugeführt, falls dieser in der DÜE erzeugt wird.
RC Empfangsschrittakt von der DÜE: Der Empfangsschrittakt wird der DEE mit dieser Leitung zugeführt, falls die Taktsynchronisation in der DÜE geschieht (Taktgenerator in der DÜE).
DTR Endgerät betriebsbereit: Die DEE signalisiert mit dieser Leitung, daß sie bereit ist, Daten auszusenden. Gleichzeitig kontrolliert dieses Signal die Anschaltung der DÜE an den Übertragungskanal.

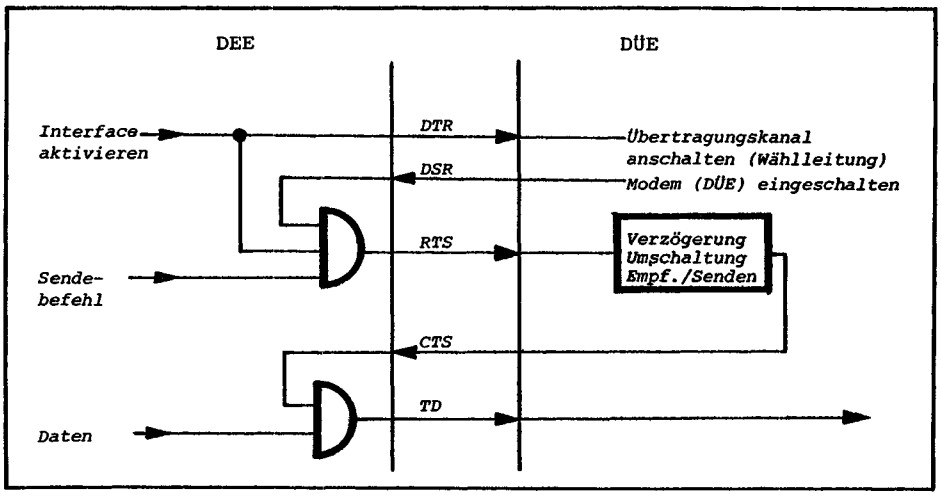


Bild 2. Verknüpfung der Melde- und Steuersignale

RI Ankommender Ruf: Bei geschalteten Rufleitungen wird mit diesem Signal ein ankommender Ruf angezeigt.
Sendeschrittakt zur DÜE: Wird der Schrittakt in der DEE erzeugt, so steht das Taktsignal auf dieser Leitung zur Verfügung.
Da die Verbindung zwischen zwei Datenstationen nur aus 2 Drähten besteht (2-Draht-Leitung), muß ein ausgefeiltes Protokoll für die Koordination der DÜE sorgen.
Empfängt eine DÜE z. B. DCD (Träger auf der Leitung), so wird dies an die DEE gemeldet. Deren Programm kann nun berücksichtigen, daß die Gegenstation sendet (eigene Sendeanforderungen werden zurückgestellt).
Häufig werden statt 2-Draht-Leitungen auch 4-Draht-Leitungen eingesetzt. Das

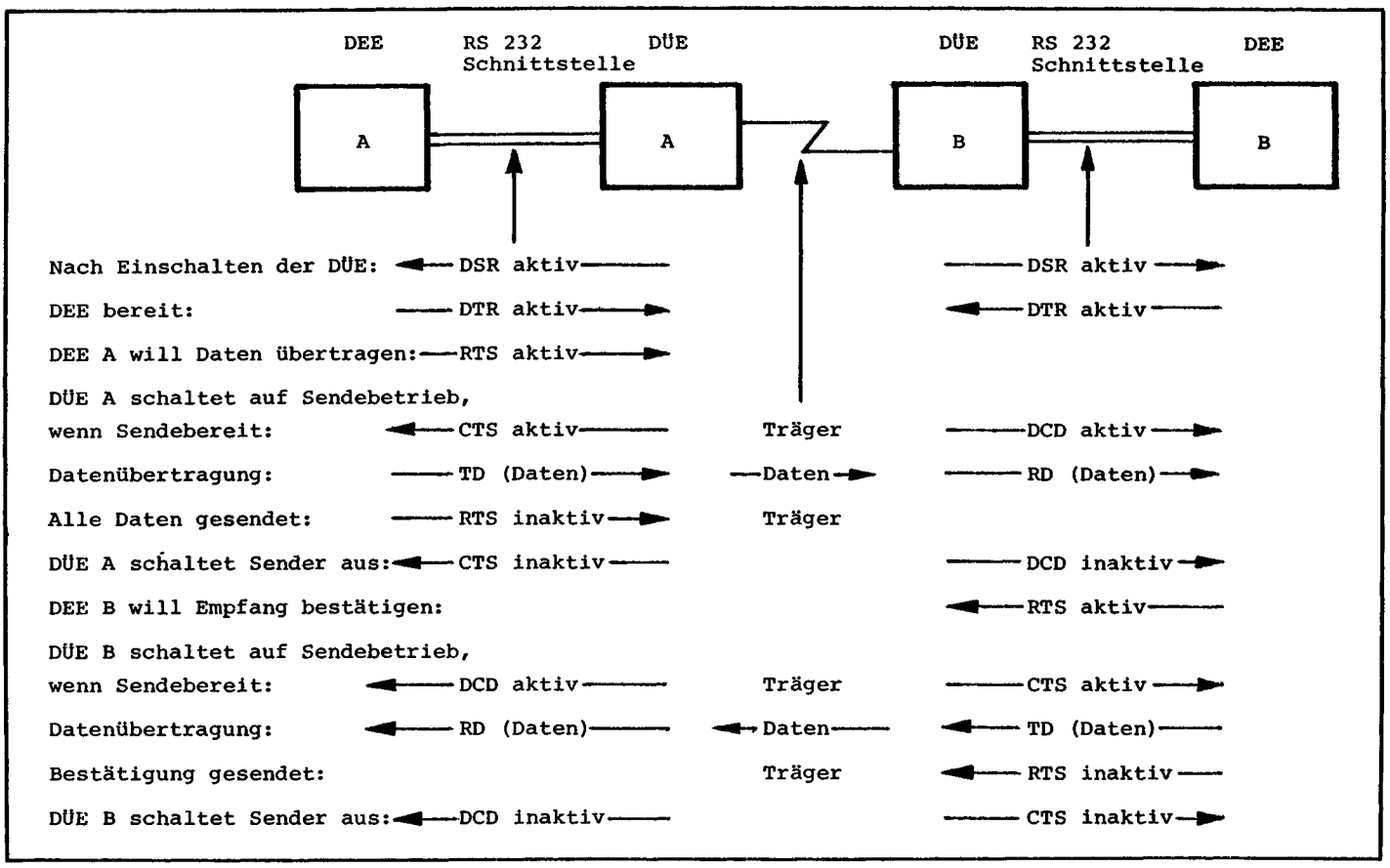


Bild 3. Beispiele für den logischen Ablauf der Interfacesignale während einer Übertragung. Halbduplex-Verkehr (Wechsel-Verkehr bei 2-Draht-Verbindung), Standleitung

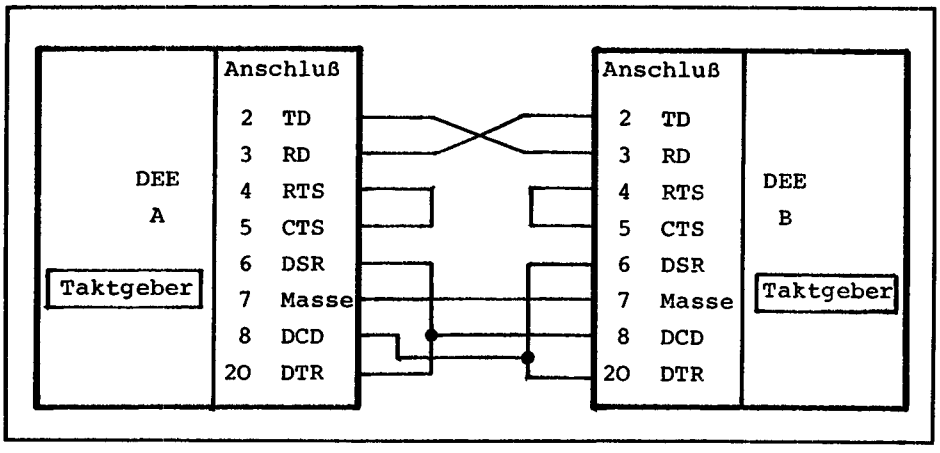


Bild 4. Beide DEEs verfügen über eigene Taktgeber

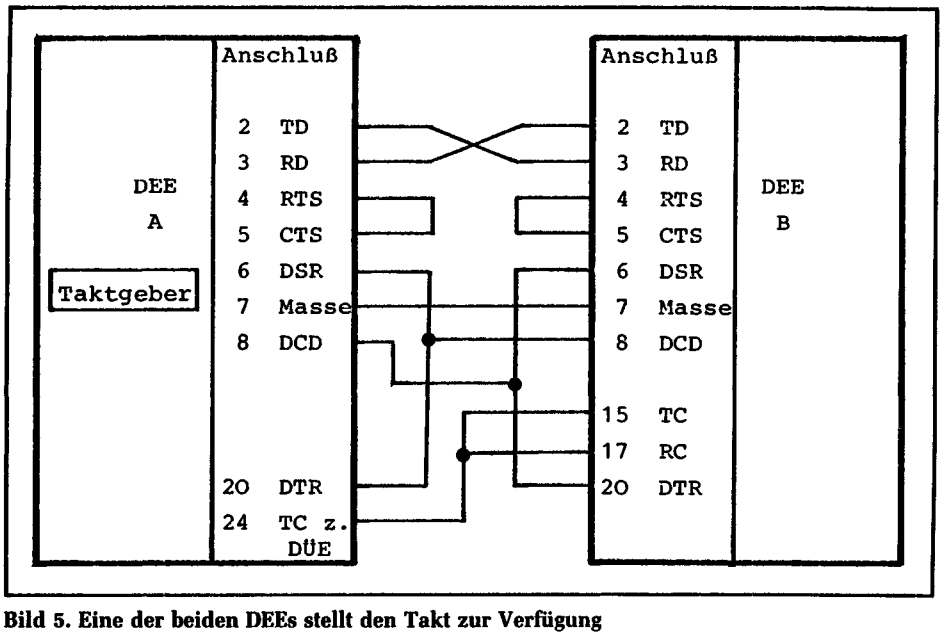


Bild 5. Eine der beiden DEEs stellt den Takt zur Verfügung

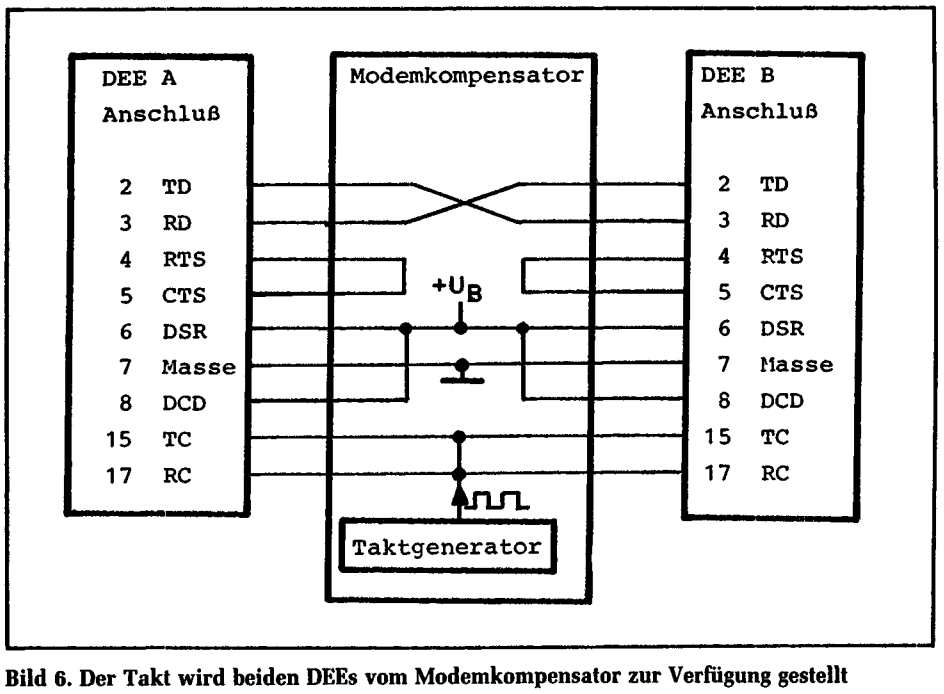


Bild 6. Der Takt wird beiden DEEs vom Modemkompensator zur Verfügung gestellt

hat folgenden Vorteil: Die DÜE benötigt eine bestimmte Zeit, um vom Empfangsbetrieb auf Sendebetrieb umzuschalten. Die Zeit ist abhängig von der Übertragungsleitung, ein durchaus typischer Wert ist 20 ms im Fernmeldenetz. Diese für den Rechner sehr lange Zeit (Instruktionen werden in µs ausgeführt), kann durch den Mehraufwand zweier zusätzlicher Drähte eingespart werden. Ein Leitungspaar verbindet ständig den Sender der einen DEE mit dem Empfänger der anderen und umgekehrt. Beide Sender sind ständig aktiv (Träger oder Daten auf den Leitungspaaren).

Elektrische Eigenschaften

Ein aktives Bit (L) auf den Datenleitungen der Schnittstelle (RD und TD) wird durch eine Spannung von −3V bis −25V repräsentiert. Kein Bit (0) entspricht +3V bis +25V. Die Melde- und Steuersignale sind aktiv, wenn die Spannung +3V bis +25V ist, inaktiv bei −3V bis −25V. Der Spannungsbereich von +3V bis −3V (inklusive 0V) ist undefiniert.

Die Verbindung zweier Datenstationen mit RS-232-Schnittstelle

In der Regel ist es nicht möglich, zwei Einheiten (z. B. zwei Rechner) mit RS-232-Anschluß direkt durch ein einfaches Kabel zu verbinden. Auf der anderen Seite ist der Einsatz von Modems (DÜE) für Entfernungen von wenigen Metern unsinnig. Hier hilft ein Modemkompensator.
Ein solches Gerät simuliert zwei DÜEs (Modem) und die Übertragungsleitung. Im einfachsten Fall kann eine solche Schaltung aus einigen Drahtbrücken bestehen. Der Selbstbau bereitet keine Schwierigkeiten.
Die einfachste Schaltung ergibt sich, wenn eine der zwei DEEs über einen eigenen Taktgeber verfügt. Dies kann kontrolliert werden, indem man an Anschlußpunkt 24 das Taktsignal mißt. Ist es nicht vorhanden, so verfügt das Gerät nicht über einen eingebauten Taktgeber. Falls keine der beiden DEEs über einen Taktgeber verfügt, muß dieser im Modemkompensator eingebaut werden. Es ist üblich bei Übertragungsgeschwindigkeiten bis 1200 Bps (Bit pro Sekunde) das Taktsignal in der DEE zu erzeugen, bei höheren Geschwindigkeiten erzeugt die DÜE den Takt.
Gebräuchliche Datenübertragungsgeschwindigkeiten:

300 Bps	2400 Bps	19 200 Bps
600 Bps	4800 Bps	usw.
1200 Bps	9600 Bps	

Kurze Maschinenprogramme lassen sich mit unserem Monitor schnell und einfach mit den hexadezimalen Operationscodes schreiben. Heinz-Peter Curdts hat sich die Mühe gemacht, die Z80-Maschinenbefehle übersichtlich zusammenzustellen

8-bit-Ladebefehle

	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(nn)	n
LD A,.	7F	78	79	7A	7B	7C	7D	7E	0A	1A	3AXXXX	3EXX
LD B,.	47	40	41	42	43	44	45	46				06XX
LD C,.	4F	48	49	4A	4B	4C	4D	4E				0EXX
LD D,.	57	50	51	52	53	54	55	56				16XX
LD E,.	5F	58	59	5A	5B	5C	5D	5E				1EXX
LD H,.	67	60	61	62	63	64	65	66				26XX
LD L,.	6F	68	69	6A	6B	6C	6D	6E				2EXX
LD (HL),.	77	70	71	72	73	74	75					36XX
LD (BC),.	02											
LD (DE),.	12											
LD (nn),.	32XXXX											

	A	B	C	D	E	H	L
LD .,(IX+d)	DD7EXX	DD46XX	DD4EXX	DD56XX	DD5EXX	DD66XX	DD6EXX
LD .,(IY+d)	FD7EXX	FD46XX	FD4EXX	FD56XX	FD5EXX	FD66XX	FD6EXX
LD (IX+d),.	DD77XX	DD70XX	DD71XX	DD72XX	DD73XX	DD74XX	DD75XX
LD (IY+d),.	FD77XX	FD70XX	FD71XX	FD72XX	FD73XX	FD74XX	FD75XX

LD (IX+d),n DD36XXXX LD (IY+d),n FD36XXXX

		S	Z	H	P/V	N	C
LD A,I	ED57	*	*	0	*	0	-
LD A,R	ED5F	*	*	0	*	0	-
LD I,A	ED47	-	-	-	-	-	-
LD R,A	ED4F	-	-	-	-	-	-

16-bit-Ladebefehle

	BC	DE	HL	SP	IX	IY
LD ..,nn	01XXXX	11XXXX	21XXXX	31XXXX	DD21XXXX	FD21XXXX
LD ..,(nn)	ED4BXXXX	ED5BXXXX	2AXXXX	ED7BXXXX	DD2AXXXX	FD2AXXXX
LD (nn),..	ED43XXXX	ED53XXXX	22XXXX	ED73XXXX	DD22XXXX	FD22XXXX
LD SP			F9		DDF9	FDF9

	BC	DE	HL	AF	IX	IY
PUSH ..	C5	D5	E5	F5	DDE5	FDE5
POP ..	C1	D1	E1	F1	DDE1	FDE1

EX (SP),HL	E3	EX DE,HL	EB
EX (SP),IX	DDE3	EX AF,AF'	08
EX (SP),IY	FDE3	EXX	D9 BC-BC' DE-DE' HL-HL'

Blocktransfer- und Suchbefehle

		S	Z	H	P/V	N	C	
LDI	EDA0	-	-	0	*	0	-	LD(DE),(HL);INC HL;INC DE;DEC BC
LDIR	EDB0	-	-	0	0	0	-	wie LDI,wiederholen bis BC=0
LDD	EDA8	-	-	0	*	0	-	LD(DE),(HL);DEC HL;DEC DE;DEC BC
LDDR	EDB8	-	-	0	0	0	-	wie LDD,wiederholen bis BC=0
CPI	EDA1	*	*	*	*	1	-	LD A,(HL);INC HL;DEC BC
CPIR	EDB1	*	*	*	*	1	-	wie CPI,wiederholen bis BC=0 oder Zeichen gefunden
CPD	EDA9	*	*	*	*	1	-	LD A,(HL);DEC HL;DEC BC
CPDR	EDB9	*	*	*	*	1	-	wie CPD,wiederholen bis BC=0 oder Zeichen gefunden

Sprungsbefehle

	Z	NZ	C	NC	PE	PO	M	P
JP	CAXXXX	C2XXXX	DAXXXX	D2XXXX	EAXXXX	E2XXXX	FAXXXX	F2XXXX
CALL	CCXXXX	C4XXXX	DCXXXX	D4XXXX	ECXXXX	E4XXXX	FCXXXX	F4XXXX
RET	C8	C0	D8	D0	E8	E0	F8	F0
JR	28XX	20XX	38XX	30XX				

	unbedingt	(HL)	(IX)	(IY)
JP	C3XXXX	E9	DDE9	FDE9
CALL	CDXXXX			
RET	C9			
JR	18XX			

RST 00 08 10 18 20 28 30 38
C7 CF D7 DF E7 EF F7 FF

DJNZ	10XX	DEC B;JR NZ,d
RETI	ED4D	zurueck vom Interrupt
RETN	ED45	zurueck vom nicht maskierbaren interrupt

CPU-Steuerbefehle

		S	Z	H	P/V	N	C	
NOP	00	-	-	-	-	-	-	Leerbefehl
HALT	76	-	-	-	-	-	-	
CCF	3F	-	-	*	-	0	*	Komplementiere Carry-Flag
SCF	37	-	-	0	-	0	1	Setze Carry-Flag
EI	FB	-	-	-	-	-	-	Interrupts freigeben
DI	F3	-	-	-	-	-	-	Interrupts sperren
IM 0	ED46	-	-	-	-	-	-	Interrupt-Modus 0
IM 1	ED56	-	-	-	-	-	-	Interrupt-Modus 1
IM 2	ED5E	-	-	-	-	-	-	Interrupt-Modus 2

Ein-/Ausgabebefehle

	A	B	C	D	E	H	L	S	Z	H	P/V	N	C
IN .,(C)	ED78	ED40	ED48	ED50	ED58	ED60	ED68	*	*	0	*	0	-
OUT (C),.	ED79	ED41	ED49	ED51	ED59	ED61	ED69	-	-	-	-	-	-

		S	Z	H	P/V	N	C	
IN A,n	DBXX	-	-	-	-	-	-	
OUT n,A	D3XX	-	-	-	-	-	-	
INI	EDA2	?	*	?	?	1	-	IN (HL),(C);INC HL;DEC B
INIR	EDB2	?	1	?	?	1	-	wie INI,wiederholen solange B<>0
IND	EDAA	?	*	?	?	1	-	IN (HL),(C);DEC HL;DEC B
INDR	EDBA	?	1	?	?	1	-	wie IND,wiederholen solange B<>0
OUTI	EDA3	?	*	?	?	1	-	OUT (C),(HL);INC HL;DEC B
OTIR	EDB3	?	1	?	?	1	-	wie OUTI,wiederholen solange B<>0
OUTD	EDAB	?	*	?	?	1	-	OUT (C),(HL);DEC HL;DEC B
OTDR	EDBB	?	1	?	?	1	-	wie OUTD,wiederholen solange B<>0

8-bit Arithmetische und Logische Befehle

	B	C	D	E	H	L	(HL)	A	n	(IX+d)	(IY+d)	S	Z	H	P/V	N	C
ADD .	80	81	82	83	84	85	86	87	C6XX	DD86XX	FD86XX	*	*	*	*	0	*
ADC .	88	89	8A	8B	8C	8D	8E	8F	CEXX	DD8EXX	FD8EXX	*	*	*	*	0	*
SUB .	90	91	92	93	94	95	96	97	D6XX	DD96XX	FD96XX	*	*	*	*	1	*
SBC .	98	99	9A	9B	9C	9D	9E	9F	DEXX	DD9EXX	FD9EXX	*	*	*	*	1	*
AND .	A0	A1	A2	A3	A4	A5	A6	A7	E6XX	DDA6XX	FDA6XX	*	*	1	*	0	0
XOR .	A8	A9	AA	AB	AC	AD	AE	AF	EEXX	DDAEXX	FDAEXX	*	*	1	*	0	0
OR .	B0	B1	B2	B3	B4	B5	B6	B7	F6XX	DDB6XX	FDB6XX	*	*	1	*	0	0
CP .	B8	B9	BA	BB	BC	BD	BE	BF	FEXX	DDBEXX	FDBEXX	*	*	*	*	1	*
INC .	04	0C	14	1C	24	2C	34	3C		DD34XX	FD34XX	*	*	*	*	0	-
DEC .	05	0D	15	1D	25	2D	35	3D		DD35XX	FD35XX	*	*	*	*	1	-

		S	Z	H	P/V	N	C	
DAA	27	*	*	*	*	-	*	BCD-Korrektur im Akku
CPL	2F	-	-	1	-	1	-	Komplementiere Akku (1er-Komplement)
NEG	ED44	*	*	*	*	1	*	Komplementiere Akku (2er-Komplement)

16-bit Arithmetische und Logische Befehle

	BC	DE	HL	SP	IX	IY	S	Z	H	P/V	N	C
INC .	03	13	23	33	DD23	FD23	-	-	-	-	-	-
DEC .	0B	1B	2B	3B	DD2B	FD2B	-	-	-	-	-	-
ADD HL,.	09	19	29	39			-	-	*	-	0	*
ADC HL,.	ED4A	ED5A	ED6A	ED7A			*	*	*	*	0	*
SBC HL,.	ED42	ED52	ED62	ED72			*	*	*	*	1	*
ADD IX,.	DD09	DD19		DD39	DD29		-	-	*	-	0	*
ADD IY,.	FD09	FD19		FD39		FD29	-	-	*	-	0	*

Rotations- und Schiebebefehle

	B	C	D	E	H	L	(HL)	A	(IX+d)	(IY+d)
RR	CB18	CB19	CB1A	CB1B	CB1C	CB1D	CB1E	CB1F	DDCBXX1E	FDCBXX1E
RL	CB10	CB11	CB12	CB13	CB14	CB15	CB16	CB17	DDCBXX16	FDCBXX16
RRC	CB08	CB09	CB0A	CB0B	CB0C	CB0D	CB0E	CB0F	DDCBXX0E	FDCBXX0E
RLC	CB00	CB01	CB02	CB03	CB04	CB05	CB06	CB07	DDCBXX06	FDCBXX06
SRA	CB28	CB29	CB2A	CB2B	CB2C	CB2D	CB2E	CB2F	DDCBXX2E	FDCBXX2E
SLA	CB20	CB21	CB22	CB23	CB24	CB25	CB26	CB27	DDCBXX26	FDCBXX26
SRL	CB38	CB39	CB3A	CB3B	CB3C	CB3D	CB3E	CB3F	DDCBXX3E	FDCBXX3E

	S	Z	H	P/V	N	C	
RR/RL	*	*	0	*	0	*	Rotiere Reg rechts/links durch Carry
RRC/RLC	*	*	0	*	0	*	Rotiere Reg rechts/links
SRA/SLA	*	*	0	*	0	*	Shift Reg rechts/links arithmetisch
SRL	*	*	0	*	0	*	Shift Reg rechts/links logisch

		S	Z	H	P/V	N	C	
RRCA	0F	-	-	0	-	0	*	Rotiere Akku rechts
RLCA	07	-	-	0	-	0	*	Rotiere Akku links
RRA	1F	-	-	0	-	0	*	Rotiere Akku rechts durch Carry
RLA	17	-	-	0	-	0	*	Rotiere Akku links durch Carry
RLD(HL)	ED6F	*	*	0	*	0	-	Rotiere Ziffer links zwischen Akku und (HL)
RRD(HL)	ED67	*	*	0	*	0	-	Rotiere Ziffer rechts zwischen Akku und (HL)

Einzelbitbefehle

	B	C	D	E	H	L	(HL)	A	(IX+d)	(IY+d)
BIT 0	CB40	CB41	CB42	CB43	CB44	CB45	CB46	CB47	DDCBXX46	FDCBXX46
BIT 1	CB48	CB49	CB4A	CB4B	CB4C	CB4D	CB4E	CB4F	DDCBXX4E	FDCBXX4E
BIT 2	CB50	CB51	CB52	CB53	CB54	CB55	CB56	CB57	DDCBXX56	FDCBXX56
BIT 3	CB58	CB59	CB5A	CB5B	CB5C	CB5D	CB5E	CB5F	DDCBXX5E	FDCBXX5E
BIT 4	CB60	CB61	CB62	CB63	CB64	CB65	CB66	CB67	DDCBXX66	FDCBXX66
BIT 5	CB68	CB69	CB6A	CB6B	CB6C	CB6D	CB6E	CB6F	DDCBXX6E	FDCBXX6E
BIT 6	CB70	CB71	CB72	CB73	CB74	CB75	CB76	CB77	DDCBXX76	FDCBXX76
BIT 7	CB78	CB79	CB7A	CB7B	CB7C	CB7D	CB7E	CB7F	DDCBXX7E	FDCBXX7E
RES 0	CB80	CB81	CB82	CB83	CB84	CB85	CB86	CB87	DDCBXX86	FDCBXX86
RES 1	CB88	CB89	CB8A	CB8B	CB8C	CB8D	CB8E	CB8F	DDCBXX8E	FDCBXX8E
RES 2	CB90	CB91	CB92	CB93	CB94	CB95	CB96	CB97	DDCBXX96	FDCBXX96
RES 3	CB98	CB99	CB9A	CB9B	CB9C	CB9D	CB9E	CB9F	DDCBXX9E	FDCBXX9E
RES 4	CBA0	CBA1	CBA2	CBA3	CBA4	CBA5	CBA6	CBA7	DDCBXXA6	FDCBXXA6
RES 5	CBA8	CBA9	CBAA	CBAB	CBAC	CBAD	CBAE	CBAF	DDCBXXAE	FDCBXXAE
RES 6	CBB0	CBB1	CBB2	CBB3	CBB4	CBB5	CBB6	CBB7	DDCBXXB6	FDCBXXB6
RES 7	CBB8	CBB9	CBBA	CBBB	CBBC	CBBD	CBBE	CBBF	DDCBXXBE	FDCBXXBE
SET 0	CBC0	CBC1	CBC2	CBC3	CBC4	CBC5	CBC6	CBC7	DDCBXXC6	FDCBXXC6
SET 1	CBC8	CBC9	CBCA	CBCB	CBCC	CBCD	CBCE	CBCF	DDCBXXCE	FDCBXXCE
SET 2	CBD0	CBD1	CBD2	CBD3	CBD4	CBD5	CBD6	CBD7	DDCBXXD6	FDCBXXD6
SET 3	CBD8	CBD9	CBDA	CBDB	CBDC	CBDD	CBDE	CBDF	DDCBXXDE	FDCBXXDE
SET 4	CBE0	CBE1	CBE2	CBE3	CBE4	CBE5	CBE6	CBE7	DDCBXXE6	FDCBXXE6
SET 5	CBE8	CBE9	CBEA	CBEB	CBEC	CBED	CBEE	CBEF	DDCBXXEE	FDCBXXEE
SET 6	CBF0	CBF1	CBF2	CBF3	CBF4	CBF5	CBF6	CBF7	DDCBXXF6	FDCBXXF6
SET 7	CBF8	CBF9	CBFA	CBFB	CBFC	CBFD	CBFE	CBFF	DDCBXXFE	FDCBXXFE

Flagbeeinflussung:

	S	Z	H	P/V	N	C
BIT	?	*	1	?	0	-
SET	-	-	-	-	-	-
RES	-	-	-	-	-	-

Flag-Register

Bit	7	6	5	4	3	2	1	0
	S	Z	X	H	X	P/V	N	C

	gesetzt	nicht gesetzt	wird gesetzt bei
C Carry-Flag	C	NC	Uebertrag von Bit 7
N Add-/Subtract-Flag			Subtraktionen
P/V Parity-/Overflow-Flag	PE	PO	gerader Paritaet
H Half-Carry-Flag			Uebertrag von Bit 3
Z Zero-Flag	Z	NZ	Ergebnis 0
S Sign-Flag	M	P	neg. Ergebnis
X nicht verwendet			

Beeinflussung:

- 1 gesetzt
- 0 zurueckgesetzt
- * abhaengig vom Ergebnis einer Operation
- - nicht beeinflusst
- ? unbestimmt

Reservierte Bereiche in Seite 0

Die Hauptspeicherseite 0, zwischen den Adressen 0000H und 00FH enthält verschiedene Code- und Datensegmente, die vom CP/M verwendet werden. Diese Segmente werden im folgenden erläutert:

Adresse von	Adresse bis	Inhalt
0000H	0002H	Sprungbefehl zur Warmstartroutine (4A03H+b). Das erlaubt einen einfach programmierbaren Restart (JMP 0000H) oder einen manuellen Neustart von der Bedientastatur.
0003H	0003H	Enthält das IOBYTE nach Intel-Standard (wird nicht ausgewertet)
0004H	0004H	Voreingestellte Laufwerke (0 = A, ..., 15 = P)
0005H	0007H	Sprung in das BDOS. Der Sprung dient zwei Gründen: JMP 00005H ist der primäre Eintrittspunkt in das BDOS, wie in Kapitel 3.3 beschrieben wird. LHLD 00006H bringt die Adresse der Instruktion in Register HL. Dieser Wert ist die niedrigste von CP/M verwendete Adresse (wenn man annimmt, daß der CCP überschrieben wird). Der DDT ändert beispielsweise diesen Wert, um anzuzeigen, daß im Debugmodus der Speicher reduziert ist.
0008H	00027H	Interrupt-Bereiche 1–5; nicht verwendet.
0030H	0037H	Interruptbereich 6; nicht benutzt; reserviert.
0038H	003AH	Restart 7; enthält einen Sprung in den DDT oder SID für Programm-Unterbrechungspunkte (Breakpoints), wird sonst nicht verwendet.
003BH	003FH	nicht benutzt; reserviert
0040H	004FH	16-Byte Arbeitsbereich für das CBIOS, wird von der Handelsversion des CP/M nicht verwendet.
0050H	005BH	nicht benutzt; reserviert
005CH	007CH	voreingestellter Bereich für den ersten FCB vom CCP aus
007DH	007FH	Optionaler FCB-Anhang für Random-Zugriff.
0080H	00FFH	Voreingestellter DMA-Bereich (wird vom CCP auch mit der Kommandozeile gefüllt, wenn ein Programm geladen wird).

Eintrittspunkte des BIOS

Die Eintrittspunkte des BIOS und BDOS werden unten aufgeführt. In das BIOS wird immer über einen „Sprungvektor" ab 4A00H+b eingesprungen. Die Anordnung der Vektoren ist standardisiert. Die BIOS-Routinen können für einige Funktionen während einer Rekonfiguration des CP/M auch leer sein (also z. B. nur einen RET-Befehl enthalten), der Sprungvektor muß jedoch immer vollständig sein.

Die Sprungvektor-Tabelle ab 4A00H+b hat die unten gezeigte Form, wobei die individuellen Adressen links stehen.

```
4A00H+b   JMP BOOT     ; EINSPRUNG KALTSTART
4A03H+b   JMP WBOOT    ; WARMSTART-EINSPRUNG
4A06H+b   JMP CONST    ; KONSOLENSTATUS (IST ZEICHEN DA?)
4A09H+b   JMP CONIN    ; KONSOLENZEICHEN EINLESEN
4A0CH+b   JMP CONOUT   ; KONSOLENAUSGABE
4A0FH+b   JMP LIST     ; ZEICHEN AUF LIST AUSGEBEN
4A12H+b   JMP PUNCH    ; ZEICHEN AUF PUNCH AUSGEBEN
4A15H+b   JMP READER   ; ZEICHEN VOM READER EINLESEN
4A18H+b   JMP HOME     ; LESEKOPF AUF AKTUELLEM LAUFWERK AUF
                       ; SPUR 00 POSITIONIEREN
4A1BH+b   JMP SELDSK   ; LAUFWERK SELEKTIEREN
4A1EH+b   JMP SETTRK   ; SETZE SPURNUMMER
4A21H+b   JMP SETSEC   ; SETZE SEKTORNUMMER
4A24H+b   JMP SETDMA   ; SETZE DMA-ADRESSE
4A27H+b   JMP READ     ; LIES SELEKTIERTEN SEKTOR
4A2AH+b   JMP WRITE    ; SCHREIB SELEKTIERTEN SEKTOR
4A2DH+b   JMP LISTST   ; STATUS DES LISTERS
4A30H+b   JMP SECTRAN  ; UNTERPROGRAMM ZUR SEKTORÜBERSETZUNG
```

Jede Sprungadresse korrespondiert mit einem Unterprogramm, das die entsprechende Funktion erbringt. In der Sprungtabelle gibt es drei Gruppen: Reinitialisierung (BOOT, WBOOT), einfache Zeichen-E/A (CONST, CONIN, CONOUT, LIST, PUNCH, READER und LISTST) und die Disketten-Funktionen (HOME, SELDSK, SETTRK, SETSEC, SETDMA, READ, WRITE und SECTRAN).

Das mc-CP/M-Abenteuer

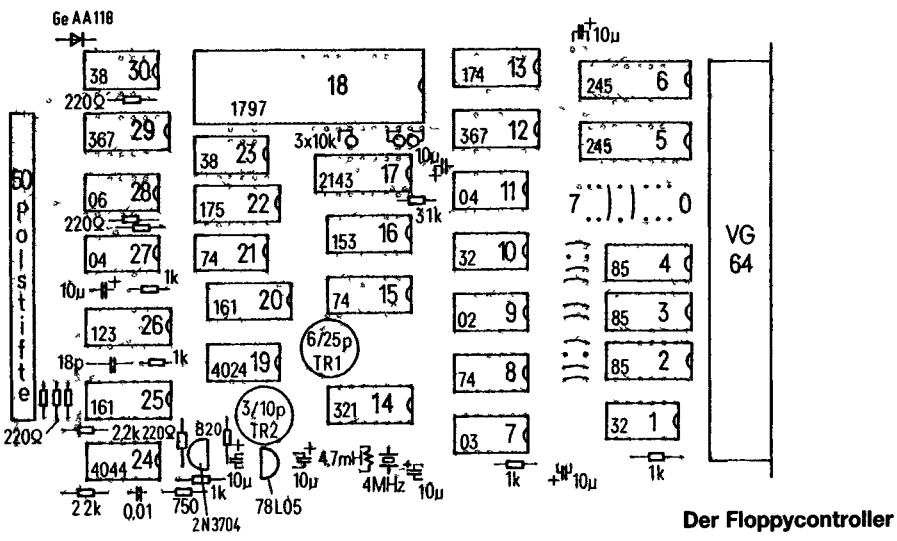
Als in mc die Serie mit einem Selbstbau-CP/M-System erschien, besorgte ich sofort die jeweiligen Platinen, baute sie auf, und soweit funktionierte auch alles auf Anhieb.
Da ich inzwischen ein CP/M mit angepaßtem BIOS besaß, war die Enttäuschung groß, als sich endlich nach vielem Einstellen der Trimmer TR 2 und TR 1 zwar CP/M auf dem Bildschirm meldete – aber das war auch schon alles. Der restliche Teil des Systems ist nämlich in doppelter Schreibdichte auf der Disk abgelegt und da hat die PLL-Schaltung ihre liebe Not. Entweder war das Signal bei doppelter Schreibdichte einigermaßen jitterfrei, dann war aber ein Reboot mit CTRL-C nicht mehr möglich oder umgekehrt.
Kurzerhand entfernte ich alle zur PLL-Schaltung gehörenden ICs und ersetzte diese durch ein einziges achtbeiniges! Lieferbar ist dieses IC m. W. von zwei Firmen: Western Digital und Standard Microsystem Corp. Typbezeichnung: FDC 9216. Für Verwendung von sowohl 8-Zoll- wie 5¼-Zoll-Laufwerken empfiehlt sich der Typ FDC 9216 01 bei WD bzw. FDC 9216 B bei SMC.
Der Umbau des bestehenden Controllers ist sehr einfach. Zuerst entfernt man folgende ICs:
Nr. 15, 74 LS 74; Nr. 19 MC 4024; Nr. 20 74 LS 161; Nr. 21 74 LS 74; Nr. 24 MC 4044; Nr. 25 74 LS 161.

Der restliche Teil der PLL (wie T1 und 7805) ist sicherlich eingelötet, so daß ein Entfernen nicht unbedingt nötig wird. Nun werden bei Sockel Nr. 25 an Pin 3, 4, 5, 6 die Verbindungen zu +5 V auf der Lötseite unterbrochen. Dieser Sockel hat somit keine weiteren Verbindungen mehr, so daß er nicht einmal ausgelötet werden muß. Sodann sind folgende Verbindungen zu schaffen:

Sockel-Nr.	Pin-Nr.	zu	Sockel-Nr.	Pin-Nr.	Bezeichnung	Bezeichnung (FDC)	IC Pin
25	3	–	21	3	(RAW DATA)	(DSKD)	1
25	4	–	15	5	RCLK	SEPCLK	2
25	5	–	14	9	4 MHz f0	REFCLK	3
25	6	gnd	–	–	0	GND	4
25	11	–	11	6	DDEN	CDO	5
25	12	gnd	–	–	0	CDI	6
25	13	–	21	9	B Ein. v. IC26	SEPD	7
25	14	+5 V	–	–	+5 V	VDD	8

Damit lief dann die Mini-Floppy einwandfrei.
Der Ausgang SEPD ist ein invertiertes Signal, welches bewirkt, daß Monoflop 26, 74LS123, auf der hinteren Flanke getriggert wird. Damit erscheint dann der Datenpuls an Pin 26 des FD 1797 nicht mehr genau in der Mitte des Clockpulses an Pin 27. Auch dies kann mit wenigen Änderungen abgeschafft werden. Verbindung Sockel 25 Pin 13 nicht an IC 21 Pin 9, sondern an IC 26 Pin 1, wobei auf der Oberseite die Verbindung zur Masse unterbrochen werden muß. Die Verbindung IC 21 Pin 9 muß jetzt an +5 V gelegt werden (B-Eingang von IC-26). Nun triggert das IC an der vorderen Flanke und der Puls erscheint dann dort, wo er hingehört. Dies ist aber ein Job für Perfektionisten, denn das Laufwerk arbeitet auch ohne letztere Änderungen einwandfrei.

Günter Kuhn



Der Floppycontroller

Umrechnung dezimal/hexadezimal und ASCII-Zeichen

hex	dez.	ASCII	hex	dez.	ASCII
00	0	NUL	20	32	Space
01	1	SOH	21	33	!
02	2	STX	22	34	"
03	3	ETX	23	35	#
04	4	EOT	24	36	$
05	5	ENQ	25	37	%
06	6	ACK	26	38	&
07	7	BEL	27	39	'
08	8	BS	28	40	(
09	9	HT	29	41	)
0A	10	LF	2A	42	*
0B	11	VT	2B	43	+
0C	12	FF	2C	44	,
0D	13	CR	2D	45	–
0E	14	SO	2E	46	.
0F	15	SI	2F	47	/
10	16	DLE	30	48	0
11	17	DC1	31	49	1
12	18	DC2	32	50	2
13	19	DC3	33	51	3
14	20	DC4	34	52	4
15	21	NAK	35	53	5
16	22	SYN	36	54	6
17	23	ETB	37	55	7
18	24	CAN	38	56	8
19	25	EM	39	57	9
1A	26	SUB	3A	58	:
1B	27	ESC	3B	59	;
1C	28	FS	3C	60	<
1D	29	GS	3D	61	=
1E	30	RS	3E	62	>
1F	31	US	3F	63	?

hex	dez.	ASCII	hex	dez.	ASCII
40	64	@	60	96	\
41	65	A	61	97	a
42	66	B	62	98	b
43	67	C	63	99	c
44	68	D	64	100	d
45	69	E	65	101	e
46	70	F	66	102	f
47	71	G	67	103	g
48	72	H	68	104	h
49	73	I	69	105	i
4A	74	J	6A	106	j
4B	75	K	6B	107	k
4C	76	L	6C	108	l
4D	77	M	6D	109	m
4E	78	N	6E	110	n
4F	79	O	6F	111	o
50	80	P	70	112	p
51	81	Q	71	113	q
52	82	R	72	114	r
53	83	S	73	115	s
54	84	T	74	116	t
55	85	U	75	117	u
56	86	V	76	118	v
57	87	W	77	119	w
58	88	X	78	120	x
59	89	Y	79	121	y
5A	90	Z	7A	122	z
5B	91	[	7B	123	{
5C	92	\	7C	124	|
5D	93	]	7D	125	}
5E	94	↑	7E	126	~
5F	95	←	7F	127	DEL