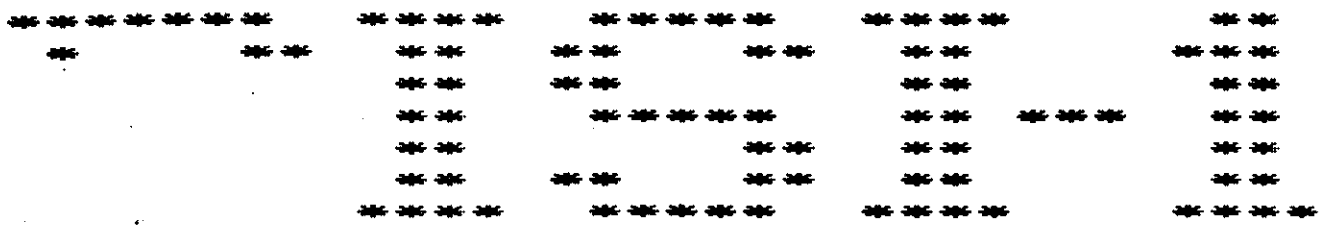


EPROM- / RAM - Floppy

CONITEC
DATENSYSTEME

DISI-1 Inhaltsverzeichnis

Seite

Hardware

Einführung I.1
Funktion I.2
Schaltung I.2
Aufbau Hinweise I.3
Jumper I.4
Programmierung I.7

Software

DISI.ASM II.1
ERASER.COM II.5
GPF.COM II.6

Anhang

CP/M plus Treiber A.1
Blockschaltbild A.2
Schaltplan A.3
Bestückungsplan A.4
Stückliste A.5
ECB-Steckerbelegung A.6

I.2 Hardware

--- Einführung ---

DISI ist als schneller Massenspeicher für ECB-Systeme konzipiert. Die Karte kann sowohl mit EPROM's als auch mit RAM's bestückt werden. Der eingebaute Akku erhält die Daten in den RAM's auch im abgeschalteten Zustand über eine genügend lange Zeit. DISI eignet sich hervorragend zum Einsatz in Datenerfassungssystemen (z.B. mit CEPAC-180) und für den Aufbau diskettenloser CP/M Systeme (z.B. mit PROF-180X). Die Ansteuerung der Karte erfolgt vollständig über vier Portadressen. Diese Lösung bietet gegenüber einer Speicherkarte im Adressbereich der CPU folgende Vorteile:

- Die Karte kann in allen ECB-Systemen eingesetzt werden, da es bei der I/O Ansteuerung keine Herstellerunterschiede gibt.

- Der Arbeitsspeicher Adressbereich wird nicht durch die Karte belegt.

- Die notwendige Software kann für alle Systeme gleich sein.

Der Geschwindigkeitsnachteil beträgt beim reinen Datentransfer zwischen 10% und 20%. Dies fällt jedoch nicht weiter ins Gewicht, da die Verwaltungssoftware im allgemeinen ein vielfaches der Übertragungszeit benötigt.

DISI enthält insgesamt 16 Sockel, die in vier Vierergruppen angeordnet sind, dadurch können maximal vier verschiedene Speichertypen bestückt werden. Meistens sind jedoch nur maximal zwei Speichertypen sinnvoll, nämlich EPROM's gleichen Typs für permanente Daten und RAM's für veränderliche Daten. Die maximale Speicherkapazität beträgt bei reiner RAM-Bestückung 512 K-Byte, bei reiner EPROM-Bestückung z.Z 2048 K-Byte, wobei durch das Intel-Banking-Konzept die obere Grenze hier bei 64 M-Byte liegt. Sollte die Kapazität einer Karte nicht ausreichen, besteht noch die Möglichkeit, mehrere Karten in einem System einzusetzen.

--- Funktionsweise ---

Die Funktionsweise der Karte läßt sich am besten anhand des Blockschaltbildes verdeutlichen.

Die Decoder- und Steuereinheit decodiert die vier Portadressen der Karte und erzeugt die notwendigen Steuersignale. Der Bytezähler erzeugt die unteren Adressleitungen für die Speicherbausteine. Bei jedem Datentransfer (Port xxxxxx1x) wird der Bytezähler um eins erhöht, mit einenem einzigen Z80 INIR- oder OUTIR-Befehl können somit bis zu 256 Byte übertragen werden. In Anlehnung an ein Floppyinterface wurden Z19, Z23 und Z24 als Sektor- bzw. Track-Latch bezeichnet. Das Track-Latch erzeugt die Adressleitungen 7 bis 14 der Speichermatrix. Über Port xxxxxx00 werden die Daten in das Sektor-Latch eingeschrieben. Mit dem Track-Latch werden die einzelnen Speicherbausteine ausgewählt. Sowohl beim Einschreiben ins Sektor-Latch, als auch ins Track-Latch wird der Bytezähler auf Null gesetzt, dadurch ist beim erneuten Einstellen eines Sektors der Bytezähler immer in einem definierten Zustand. Da das Track-Latch lediglich drei Bit, die über die Portadresse xxxxxx01 gesetzt werden, besitzt, ist noch eine kleine Besonderheit zu beachten: Mit drei Bit kann man nur eine von acht Möglichkeiten auswählen, die Speicher werden deshalb immer paarweise selektiert. Die Entscheidung, mit welchem der beiden Bausteine der Datentransfer stattfindet, wird durch die Wahl der Transfer Portadresse vorgenommen. Der Transfer mit den Bausteinen Z1 bis Z8 geschieht über die Portadresse xxxxxx10, der Transfer mit den Bausteinen Z9 bis Z16 über die Portadresse xxxxxx11.

--- Schaltung ---

DISI belegt vier aufeinanderfolgende Ports im 256-Byte I/O-Adressbereich. Z20 vergleicht die an J6 eingestellte Adresse mit dem Adressbus und selektiert die Karte, wenn die oberen 6 Adressleitungen der eingestellten Adresse entsprechen und gleichzeitig ein I/O-Zugriff stattfindet. Die acht Datenleitungen von und zum ECB-Bus werden über Z17 gepuffert.

Z 21 erzeugt die wichtigsten Steuersignale auf der Karte:

Z 21 Pin 4 : Schreibe Daten in das Sektorlatch
Z 21 Pin 5 : Schreibe Daten in das Tracklatch
Z 21 Pin 6 : Datenübertragung mit den Speichern Z 1 bis Z 8
Z 21 Pin 7 : Datenübertragung mit den Speichern Z 9 bis Z 16

Z 21 Pin 11: erhöhe Bytezähler um eins
Z 21 Pin 12: setze Bytezähler auf null

Z19 (Sektor-Latch) und Z18 (Bytezähler) erzeugen die Adressignale der Speicherbausteine. Ein Teil der Ausgänge der beiden Bausteine geht über die Jumper J1 bis J5 auf die Adressleitungen der Speicherbausteinen. Damit hat man die Möglichkeit, die Sektorlänge von 128 Byte bis 4096 Byte zu variieren.

Die ICs Z23 und Z24 bilden das Track-Latch. Die beiden Bausteine erzeugen die Chipselect-Signale für die Speicherbausteine.

Damit die Chipselect-Leitungen nach Ausschalten der Versorgungsspannung auf High-Pegel bleiben, werden Z23 und Z24 aus dem Akku der Karte gespeist. Die Schaltung aus D1 und R9 deaktiviert die Ausgänge von Z23 und Z24 wenn die normale Versorgungsspannung auf unter 4,5V sinkt.

Die Schaltung um T1 läßtdas Write-Signal nur dann durch, wenn die Versorgungsspannug gegnügend hoch ist. Im Schaltplan sind zwei alternative Beschaltungen der T1-Basis angegeben (R15 oder R12/R14/D2). Wir empfehlen die Beschaltung mit R15, da sich gezeigt hat, daß bei ungenügender Verstärkung von T1 die Bestückung mit R12/R14/D2 die Flanken der Write-Signals zu stark verschleifen.

--- Aufbau Hinweise ---

Der Aufbau der Karte ist unproblematisch, allerdings sollten Erfahrungen im Aufbau solch eng bestückter Karten vorhanden sein. Hier einige Hinweise, die man beachten muß:

- Unter den IC-Sockeln befinden sich zwei Widerstände und die meisten Abblockkondensatoren. Beim Beschaffen der Bauteile ist darauf zu achten, daß die Abblockkondensatoren klein genung sind und die IC-Sockel in der Mitte ausgespart sind.

- R15 ist im Bestückungsplan nicht eingezeichnet. Er wird so eingelötet, daß er jeweils das zur VG-Leiste zeigende Lötauge von R12 und D2 (beide nicht bestückt) verbindet (am besten vorher mit dem Ohm-Meter prüfen).

- Falls die Karte nur mit EPROM's bestückt werden soll, können der Akku, R10, R11, D3, R13, R15 und T1 entfallen. Anstelle von D3 und T1 werden Drahtbrücken eingelötet (bei T1 Drahtbrücke zwischen Kolektor und Emitter).

- R10 darf nur dann bestückt werden, wenn der Akku die UBAT-Leitung des ECB-Busses speisen soll. UBAT darf in diesem Fall von keinem zweiten Akku versorgt werden.

--- Jumper ---

Portadresse:

Zum Einstellen der Portadressen dienen 6 Steckbrücken, die mit J6 bezeichnet sind. Da die Karte vier Portadressen belegt, kann jede Adresse im 256 Byte I/O-Adressbereich eingestellt werden. Die Karte wird dann ausgewählt, wenn ein I/O Zugriff erfolgt und sich die Adressleitungen A2 bis A7 des ECB-Busses mit der J6 Jumperstellung decken. Jumper offen bedeutet: zugehörige Adressleitung muß High-Pegel haben; Jumper gebrückt bedeutet: zugehörige Adressleitung muß Low-Pegel haben. Es gilt die folgende Zuordnung der Jumper zu den Adressleitungen:

J6 1- 2	ECB-Adressleitung 2
J6 3- 4	ECB-Adressleitung 3
J6 5- 6	ECB-Adressleitung 4
J6 7- 8	ECB-Adressleitung 5
J6 9- 10	ECB-Adressleitung 6
J6 11- 12	ECB-Adressleitung 7

Beispiel:

Portadresse	60h-63h	F8h-FBh	B8h-BBh
J6 1- 2	gebrückt	gebrückt	gebrückt
J6 3- 4	gebrückt	offen	offen
J6 5- 6	gebrückt	offen	offen
J6 7- 8	offen	offen	offen
J6 9-10	offen	offen	gebrückt
J6 11-12	gebrückt	offen	offen

Sektorlänge:

Mit den Jumpern J1 bis J5 können Sektorlängen von 128 Byte bis 4096 Byte eingestellt werden:

	J1	J2	J3	J4	J5
Sektorlänge 128 Byte	2-3	2-3	2-3	2-3	2-3
Sektorlänge 256 Byte	1-2	2-3	2-3	2-3	2-3
Sektorlänge 512 Byte	1-2	1-2	2-3	2-3	2-3
Sektorlänge 1024 Byte	1-2	1-2	1-2	2-3	2-3
Sektorlänge 2048 Byte	1-2	1-2	1-2	1-2	2-3
Sektorlänge 4096 Byte	1-2	1-2	1-2	1-2	1-2

Falls EPROM's vom Typ 27512 eingesetzt werden, ist die Sektorlänge von 128 Byte nicht möglich, da in diesem Fall die Adressleitungen 7 und 15 identisch sind.

Schreibschutz:

J7 offen: Der Schreibschutz für alle Speicherbausteine ist aktiv.

J7 gebrückt: Die Speicherbausteine können beschrieben werden.

Achtung: Wenn gebankte EPROM's Verwendung finden, muß J7 immer gebrückt sein, da die Bankumschaltung einen Schreibzugriff erfordert.

Konfiguration der Speicherbausteine:

Die 16 Speicherbausteine sind in vier Vierergruppen aufgeteilt. Die Bausteinkonfiguration kann für jede Gruppe unabhängig eingestellt werden. Auf der Karte befinden sich insgesamt 16 Jumper zur Bausteineinstellung. Die Jumper sind mit jeweils zwei Ziffern bezeichnet. Die erste Ziffer gibt an, für welche Gruppe der Jumper zuständig ist, die zweite Ziffer gibt die Funktion an.

Zuordnung der Jumper zu den Vierergruppen:

J1x erste Gruppe (Z1-Z4)
J2x zweite Gruppe (Z5-Z8)
J3x dritte Gruppe (Z9-Z11)
J4x vierte Gruppe (Z12-Z16)

Funktionszuordnung der Jumper:

Jumper	Speicherpin	mögliche Signale
Jx1	Pin 1	A14, A15, High-Pegel
Jx2	Pin 27	A14, /Write, High-Pegel von Akku
Jx3	Pin 26	A13, 5V Versorgung
Jx4	Pin 28	5V Versorgung, Akkuversorgung

Die nachfolgende Tabelle gibt mögliche Jumperstellungen für verschiedene Speicherbausteine an:

xxx : Jumperstellung unwichtig
--- : kein Jumper
1-2 / 2-3 : entsprechende Pins brücken

Type	Jx1	Jx2	Jx3	Jx4	Anmerkung
EPROM 2716	xxx	xxx	2-3	xxx	1)
EPROM 2732	xxx	xxx	2-3	xxx	
EPROM 2764	---	---	xxx	2-3	
EPROM 27128	---	---	1-2	2-3	
EPROM 27256	---	1-2	1-2	2-3	
EPROM 27512	1-2	1-2	1-2	2-3	2)
EPROM 27513	---	2-3	1-2	2-3	
EPROM 27011	---	2-3	1-2	2-3	
RAM 6264	xxx	2-3	1-2	1-2	3)
RAM 43256	2-3	2-3	1-2	1-2	

Anmerkungen:

1) A11 muß vor dem Zugriff über das Trackregister auf High-Pegel gesetzt werden. Sektorgröße maximal 2048 Byte.

2) Sektorgröße mindestens 256 Byte.

3) Vor dem Zugriff muß A13 über das Trackregister auf High-Pegel gesetzt werden. Nach dem Zugriff muß A13 auf Low-Pegel gesetzt werden. Diese Maßnahme ist notwendig, damit die Daten im RAM beim Ausschalten der Versorgungsspannung nicht zerstört werden.

--- Programmierung ---

Die Programmierung der Karte ist denkbar einfach: Zuerst wird der gewünschte Sektor über das Sektor- und Track-Register angwählt, danach erfolgt der Datentransfer mit INIR- bzw. OUTIR-Befehlen. Die einzelnen Bits des Sektor- und Trackregister haben folgende Bedeutung:

Sektor-Register:

Adresse: xxxxxx00

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A14	A13	A12	A11	A10	A9	A8	A7/ A15

Track-Register:

Adresse: xxxxxx01

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X	X	X	X	X	S2	S1	S0

X: ohne Bedeutung

Die drei Bits S0, S1 und S2 wählen aus den 16 Speicherbausteinen zwei aus:

S2	S1	S0	ausgewählte Bausteine
0	0	0	Z1 / Z9
0	0	1	Z2 / Z10
0	1	0	Z3 / Z11
0	1	1	Z4 / Z12
1	0	0	Z5 / Z13
1	0	1	Z6 / Z14
1	1	0	Z7 / Z15
1	1	1	Z8 / Z16

Sektor-Register und Track-Register können nur beschrieben, nicht aber gelesen werden. Beim Zugriff auf eins der beiden Register wird automatisch der Bytezähler zurückgesetzt. Der Datentransfer mit Z1 bis Z8 erfolgt über die Portadresse xxxxxx10, der Datentransfer mit Z9 bis Z16 erfolgt über die Adresse xxxxxx11.

II.2 Software für CP/M plus

Dieser Teil des Handbuchs beschreibt die CP/M plus Software für die DISI-Karte. Wir behalten uns vor, die Software weiterhin zu verbessern, so daß Teile dieser Beschreibung ungültig werden können. Auf der DISI-Diskette finden Sie die jeweils aktuelle Beschreibung der Software in der Datei DISI.DOC.

--- DISI.ASM ---

Diese Datei enthält den Source-Code des DISI-Treiber-Programms für CP/M plus. Bevor die DISI-Karte in Ihrem System verwendet werden kann müssen Sie DISI.ASM für Ihre Anwendung konfigurieren und in Ihr BIOS einbinden.

DISI.ASM unterstützt bis zu vier DISI-Karten, maximal zwei logische Laufwerke und folgende Bausteine:

RAM's : 8K*8 (z.B. 6264), 32K*8 (z.B. 63256)
EPROM's : 2716, 2732, 2764, 27128, 27256, 27513, 27011

Der Treiber ist so ausgelegt, daß er ohne Änderung auf jedem CP/M plus System verwendet werden kann, sofern der BIOS-Kern den Empfehlungen von Digital Research entspricht.

Konfiguration

Am Anfang von DISI.ASM befinden sich 14 EQU-Anweisungen, mit denen der Treiber konfiguriert wird. Die Anweisungen haben folgende Bedeutung:

disia, disib, disic, disid

Geben Sie hinter diesen Labels die Basisadressen der verwendeten DISI-Karten an. Werden weniger als vier Karten verwendet, dann sind die verbleibenden Labels auf den Wert OFFh zu setzen.

nr0, nr1

Diese beiden Labels definieren die Anzahl der verwendeten Speicherbausteine (max. 64 zusammen) für das erste und zweite Laufwerk. Ist nr1 auf null gesetzt, so wird nur ein Laufwerk definiert; alle sonstigen Angaben für das zweite Laufwerk sind dann ohne Bedeutung. Bei RAM-Bausteinen sollte immer die Anzahl der Bausteine angegeben werden, die auch physikalisch vorhanden sind. Bei EPROM's ist es hingegen sinnvoll, einige Bausteine mehr zu definieren als tatsächlich benötigt werden. Neue Bausteine können dann einfach hinzugesteckt werden, ohne daß das BIOS geändert werden muß. Die Speicherbausteine innerhalb eines Laufwerks müssen fortlaufend bestückt sein, es gilt folgende Reihenfolge: Karte1/Z1, Karte1/Z2, ... , Karte1/Z16, Karte2/Z1 ... usw.

offs1

Die Zahl hinter diesem Label gibt die Nummer des ersten Speicherbausteins vom zweiten Laufwerk an. Beachten Sie bitte, daß die Zählweise bei Null beginnt. Falls die Bausteine für das zweite Laufwerk anders als beim Ersten gejumpert werden, muß offs1 durch vier teilbar sein. Das erste Laufwerk beginnt immer bei Baustein Nummer Null, deshalb ist hier kein Offset notwendig.

bkap0, bkap1

Diese beiden Labels definieren die Bausteinkapazität für jedes DISI-Laufwerk in K-Byte. Die Kapazität für das erste Laufwerk ergibt sich aus nr0 * bkap0, die Kapazität für das zweite Laufwerk aus nr1 * bkap1. Ist die Bausteinkapazität größer als 32 K-Byte, wird automatisch angenommen, daß es sich um ein gebanktes EPROM handelt.

rw0, rw1

Diese beiden Labels geben für jedes Laufwerk an, ob es sich um eine RAM-Floppy oder um eine EPROM-Floppy handelt. "ja" definiert RAM-Floppy, "nein" definiert EPROM-Floppy.

amsk0, amsk1

Je nach Bausteintyp muß noch eine Odermaske definiert werden, die vor dem Zugriff auf die DISI-Karte mit dem Trackregister verknüpft wird. In der Jumpertabelle für die Speicherbausteine ist die zugehörige Odermaske angegeben.

prof180

Dieses Label wird auf "ja" gesetzt, falls der Treiber in einem PROF-180X System eingesetzt wird. Bei allen anderen Computern ist dieses Label auf "nein" zu setzen.

Beispiel

Sie wollen zwei DISI-Karten mit RAM- und EPROM-Laufwerk auf einem beliebigen CP/M plus Rechner einsetzen. Die RAM-Floppy umfaßt 20 Bausteine vom Typ 63256, die EPROM-Floppy maximal 12 Bausteine vom Typ 27513. Als Basisadresse wählen Sie B8 (Hex) und BB (Hex) aus. Das erste Laufwerk soll die EPROM-Floppy enthalten, das zweite Laufwerk die RAM-Floppy. Der Treiber muß demnach folgendermaßen konfiguriert sein.

```
prof180   equ     nein

disia     equ     0b8h
disib     equ     0bbh
disic     equ     0ffh
disid     equ     0ffh

bkap0     equ     64
amsk0     equ     00h
nr0       equ     12
rw0       equ     nein

offs1     equ     12
bkap1     equ     32
amsk1     equ     00h
nr1       equ     20
rw1       equ     ja
```

Auf der ersten Karte Z1 bis Z12 werden die EPROM's bestückt, auf der zweiten Karte und den restlichen Sockeln der ersten Karte die RAM's.

Jumper

DISI.ASM arbeitet grundsätzlich mit einer Sektorlänge von 128 Byte, J1 bis J5 müssen demnach in Stellung 2-3 stehen.

Die Bausteinjumperung ergibt sich aus folgender Tabelle:

xxx : Jumperstellung unwichtig
--- : kein Jumper
1-2 / 2-3 : entsprechende Pins brücken

Type	Jx1	Jx2	Jx3	Jx4	Odermaske (amsk0/amsk1)
EPROM 2716	xxx	xxx	2-3	xxx	10h
EPROM 2732	xxx	xxx	2-3	xxx	00h
EPROM 2764	---	---	xxx	2-3	00h
EPROM 27128	---	---	1-2	2-3	00h
EPROM 27256	---	1-2	1-2	2-3	00h
EPROM 27513	---	2-3	1-2	2-3	00h
EPROM 27011	---	2-3	1-2	2-3	00h
RAM 6264	xxx	2-3	1-2	1-2	40h
RAM 43256	2-3	2-3	1-2	1-2	00h

Wenn RAM's oder gebankte EPROM's bestückt werden muß J7 unbedingt gebrückt sein.

Einbinden ins BIOS

Nachdem die Konfiguration, wie oben angegeben, durchgeführt wurde (Texteditor), kann DISI.ASM mit dem Assembler RMAC assembliert werden:

```
RMAC DISI
```

Auf der Diskette müssen sich allerdings noch die Macro-Libraries CPM3.LIB und Z80.LIB (beim PROF-180X statt Z80.LIB HD64180.LIB) befinden. Der Assembler liefert als Ergebnis die Datei DISI.REL, die über die global (public) definierten Labels fdisi0 und fdisi1 (fdisi1 nur wenn zwei Laufwerke konfiguriert wurden) mit dem restlichen BIOS verbunden werden kann. Diese Verbindung geschieht über das Drive Table Modul DRVTBL.ASM (beim PROF-180X heißt diese Modul P6.ASM). Tragen Sie also fdisi0 (und fdisi1, wenn zwei Laufwerke konfiguriert wurden) in DRVTBL.ASM ein. Durch die Stellung des Eintrags in DRVTBL.ASM ergibt sich

die Laufwerksbezeichnung. Conitec schlägt als nichtflüchtige RAM-Floppy den Laufwerskbuchstaben F und als EPROM-Floppy den Buchstaben G vor. Nachdem DRVTBL.ASM neu assembliert wurde, können die BIOS-Module nach der Angabe Ihres Computerherstellers gelinkt werden (mit zusätzlichem Modul DISI). Für den PROF-80 von Conitec sieht der Linkvorgang z.B. so aus:

```
LINK
BNKBIOS3ABU=BIOSKRNL,SCB,BOOT,CHARIO,MOVE,DRVTBL,FDPROFBK,DISI
```

Der Linker erzeugt ein neues BIOS, daß, wie gewohnt, mit GENCPM zu einem neuen CPM3.SYS zusammengebunden wird.

--- ERASER.COM ---

Bevor zum ersten Mal auf eine nichtflüchtige statische RAM-Floppy zugegriffen wird, muß das Directory so mit Daten gefüllt werden, daß das CP/M die RAM-Floppy als gelöscht erkennt.

Das Programm ERASER.COM löscht auf diese Weise die RAM-Floppy. Nach dem Start fragt ERASER nach dem Laufwerksbuchstaben der RAM-Floppy.

*** Achtung ***

ERASER löscht unter Umständen auch normale Disketten, wenn deren Laufwerksbuchstabe angegeben wird. Als kleinen Schutz haben wir ERASER so konzipiert, daß bei Angabe der Laufwerksbuchstaben A-D mit einem Fehler abgebrochen wird. ERASER liegt im Turbo-Pascal Source-Code vor, wir empfehlen dringend, ERASER so abzuändern, daß nur die RAM-Floppy auf der DISI-Karte gelöscht werden kann (die Änderung ist im Source-Code beschrieben).

*** Wichtig ***

Da die DISI-Laufwerke im Treiber mit festem Medium definiert wurden, bekommt CP/M plus das Löschen der RAM-Floppy erst nach einem erneuten Start mit. Nach dem Aufruf von ERASER muß CP/M deshalb neu gebootet werden.

--- GPF.COM ---

Wie jedes andere Betriebssystem benötigt CP/M die Daten auf den Massenspeichern in einer bestimmten Struktur. Da diese Struktur beim Schreiben auf die Floppy automatisch angelegt wird, braucht man sich im allgemeinen nicht darum zu kümmern. Anders ist dies bei einer EPROM-Floppy, da das Betriebssystem von einer EPROM-Floppy nur lesen kann.

Die Dateien, die in einer EPROM-Floppy untergebracht werden, müssen also die gleiche Struktur haben, die entstehen würde, wenn das CP/M direkt in die EPROM's schreiben könnte. Das Programm GPF.COM erzeugt genau diese Struktur.

GPF.COM legt ein virtuelles Laufwerk an, das durch eine Anzahl von Ziel-Dateien representiert wird. Diese Ziel-Dateien werden dann in EPROM's gebrannt und mit Hilfe der DISI-Karte als EPROM-Laufwerk benutzt.

GPF arbeitet mit zwei Disketten-Laufwerken, dem Quell-Laufwerk und dem Ziel-Laufwerk. Nach dem Start liest GPF vom Quell-Laufwerk die CP/M-Dateien, die in die EPROM-Floppy kopiert werden sollen und legt dabei auf dem Ziel-Laufwerk das virtuelle Laufwerk an, representiert durch die Ziel-Dateien.

Die Ziel-Dateien sind durchgehend nummeriert und heißen DISI000.HXC ..... DISI063.HXC. Die Größe der Ziel-Dateien entspricht der Größe der verwendeten EPROM's und können deshalb direkt in EPROM's gebrannt werden.

Stellt GPF fest, daß auf der Ziel-Diskette nicht mehr genügend Platz für eine neue Ziel-Datei ist, fordert es zum Diskettenwechsel im Ziel-Laufwerk auf. Zum Schluß wird dann allerdings noch einmal die erste Ziel-Diskette angefordert, da in die Ziel-Datei DISI000.HXC noch das Directory geschrieben werden muß. Soll eine Datei kopiert werden, die sich nicht auf dem Quell-Laufwerk befindet, fordert DISI ebenfalls einen Diskettenwechsel im Quell-Laufwerk an. Durch diesen Mechanismus ist GPF in der Lage, ein virtuelles Laufwerk zu verwalten, daß in seiner Kapazität die der Disketten weit übersteigt.

GPF.COM erhält seine Anweisungen aus einer Datei namens GPF.CTR. Es handelt sich dabei um eine gewöhnliche Textdatei, die mit Wordstar, ED oder ähnlichen Editoren erstellt werden kann. GPF.CTR muß sich beim Aufruf von GPF.COM auf der Default-Disk befinden. GPF.CTR hat folgenden Aufbau:

Beispiel für GPF.CTR

```
-------------------------------------------------------------

16                      ; Größe der EPROMS in K-Byte
20                      ; Maximale Anzahl der EPROMS
A                       ; Quell-Laufwerk
B                       ; Ziel-Laufwerk
2                       ; logische CP/M Blockgröße in K-Byte
256                     ; Anzahl der Directory-Einträge
TURBO.COM               ; Quell-Datei Nr.1
TURBO.OVR               ; Quell-Datei Nr.2
TURBOMSG.MSG S          ; Quell-Datei Nr.3, setze SYS-Atribut
PIP.COM SAR             ; Quell-Datei Nr.4, setze alle Atribute
DATE.COM                ;
DIR.COM                 ;
SID.COM                 ;
usw.

-------------------------------------------------------------
```

Die beiden Parameter <log.Blockgröße> und <Anzahl der Directory-Einträge> müssen mit den Werten im CP/M Treiber übereinstimmen. Am einfachsten ist es, wenn beide Parameter weggelassen werden (geht nur gemeinsam). GPF.COM setzt diese Werte dann automatisch so, wie sie auch vom Treiber DISI.ASM gesetzt werden.

Es besteht die Option, die Atribute Read-Only, System und Archiviert, beim Kopieren ins virtuelle Laufwerk zu setzen, alle eventuell vorhandenen Atribute der Quell-Datei werden automatisch gelöscht. Die Angabe der Atribute erfolgt durch die Buchstaben 'R' für Read-Only, 'S' für System und 'A' für Archiviert. Die Atributzeichen werden durch ein Blank getrennt hinter den Dateinamen geschrieben.

Erscheint in einer GPF.CTR Zeile ein Semikolon, so wird der der Rest der Zeile einschließlich des Semikolons als Kommentar betrachtet.

Die Anzahl der von GPF.COM erzeugten Ziel-Dateien richtet sich nach der Anzahl und Größe der Quell-Dateien. In der Steuer-Datei GPF.CTR sollte aber immer die maximal mögliche EPROM Anzahl angegeben werden. Die von GPF.COM erzeugten Zieldateien sind

nämlich so aufgebaut, daß jeweils das erste und letzte EPROM überprogrammiert werden können, falls neue Dateien in die EPROM-Floppy gebracht werden sollen.

```
        title   'CP/M plus DISI-EPROM/RAM-Disk driver'

;       (c) Copyright Conitec 1986 JH

;       Letzte Aenderung am 08.12.1986 (Joachim)

ja      equ     -1              ; nicht veraendern
nein    equ     not ja          ;

; DISI-Konfiguration
;
; Der Index gibt an, ob sich die Daten auf das erste oder das
; zweite Laufwerk beziehen.

prof180 equ     ja              ; PROF-180X = ja, Sonstige = nein

disia   equ     0b8h            ; DISI 1 Basisadresse
disib   equ     0ffh            ; DISI 2 Basisadresse
disic   equ     0ffh            ; DISI 3 Basisadresse
disid   equ     0ffh            ; DISI 4 Basisadresse
                                ; Werden weniger als 4 Karten eingesetzt,
                                ; so erhalten die nicht benutzten Karten
                                ; die Portadresse 0FFh

bkap0   equ     64              ; Bausteinkapazitaet in KB
amsk0   equ     00h             ; Odermaske fuer Sektorregister beim Lesen/Schreiben
nr0     equ     2               ; Anzahl der Bausteine
rw0     equ     nein            ; RAM = ja, EPROM = nein

offs1   equ     8               ; zweites Laufwerk beginnt ab diesem Baustein
bkap1   equ     32              ; Bausteinkapazitaet in KB
amsk1   equ     00h             ; Odermaske fuer Sektorregister beim Lesen/Schreiben
nr1     equ     8               ; Anzahl der Bausteine, wenn 0 nur ein Laufwerk
rw1     equ     ja              ; RAM = ja, EPROM = nein

;
; Ende der Konfiguration, ab hier bleibt die Datei unveraendert
;
;-----------------------------------------------------------------------------

kap0    equ     bkap0*nr0       ; Kapazitaet des ersten Laufwerks
kap1    equ     bkap1*nr1       ; Kapazitaet des zweiten Laufwerks

bnks0   set     1               ; bestimme Anzahl der EPROM Bank's
sf0     set     0               ; und Shiftfaktor
        if      bkap0 gt 32     ; Wenn EPROM groesser als 32 KB, dann
bnks0   set     bkap0/16        ; dann muss es gebankt sein.
sf0     set     2               ; hier 27513
        endif                   ;
        if      bkap0 gt 64     ;
sf0     set     3               ; hier 27011
        endif                   ;

        if      nr1 ne 0        ;
bnks1   set     1               ; bestimme Anzahl der EPROM Bank's
sf1     set     0               ; und Shiftfaktor
        if      bkap1 gt 32     ; Wenn EPROM groesser als 32 KB, dann
bnks1   set     bkap1/16        ; dann muss es gebankt sein.
sf1     set     2               ; hier 27513
        endif                   ;
```

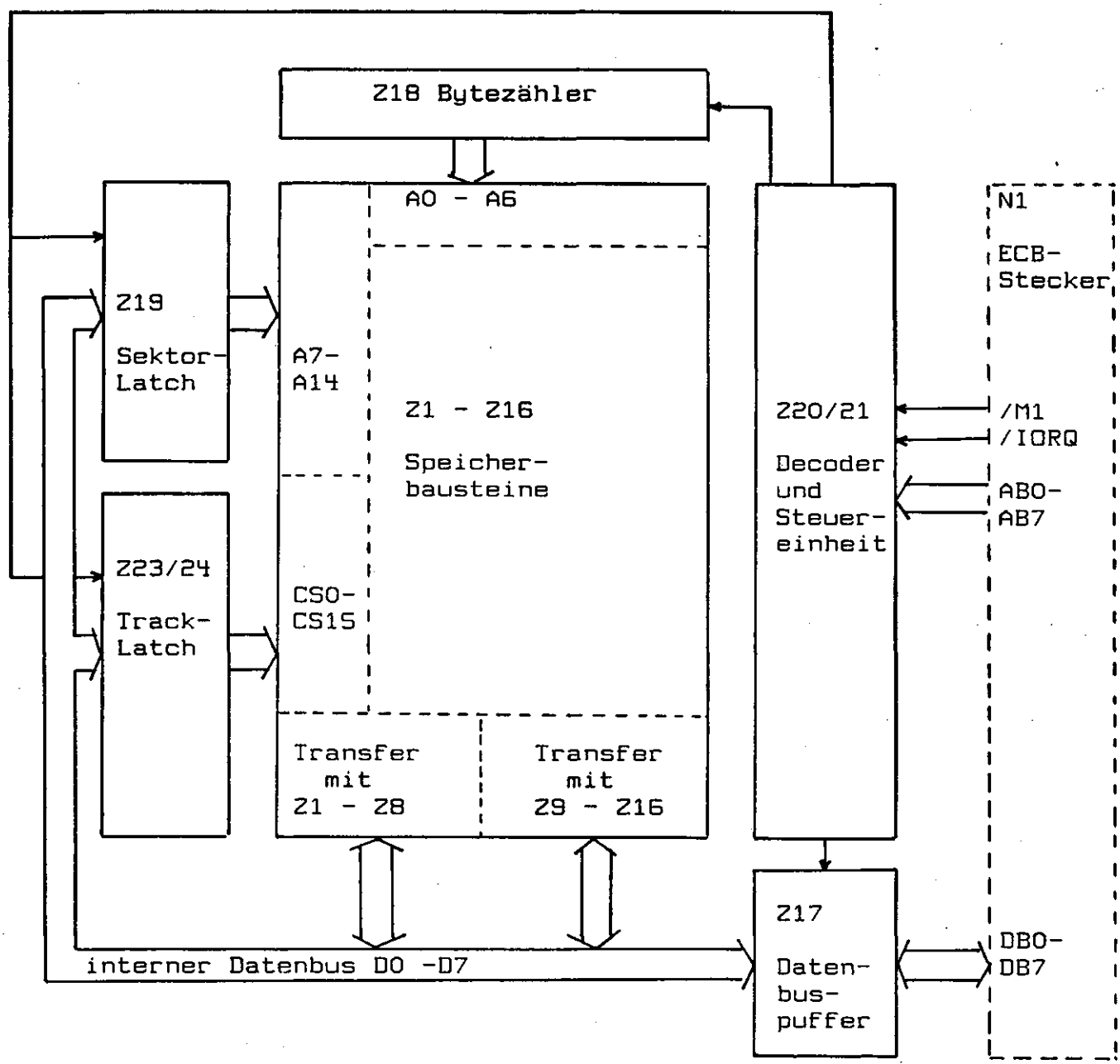
```
        if      bkap1 gt 64     ;
        set     3               ; hier 27011
        endif                   ;
        endif


nrcard  set     0               ; bestimme anzahl der karten
        if      disia ne 0ffh   ;
nrcard  set     nrcard+1        ;
        endif                   ;
        if      disib ne 0ffh   ;
nrcard  set     nrcard+1        ;
        endif                   ;
        if      disic ne 0ffh   ;
nrcard  set     nrcard+1        ;
        endif                   ;
        if      disid ne 0ffh   ;
nrcard  set     nrcard+1        ;
        endif                   ;

        dseg

        maclib  cpm3

        if      prof180
        maclib  hd64180
        else
        maclib  z80
        endif

        public  fdisi0

        if      nr1 ne 0
        public  fdisi1
        endif

        extrn   §adrv,§rdrv,§dma,§trk,§sect,§dbnk,§cbnk
        extrn   ?bank

; bestimme Blockgroesse und Dir-Eintraege aus Kapazitaet

block0  set     1024
dirs0   set     32
        if      kap0 gt 96
dirs0   set     64
        endif
        if      kap0 gt 256
block0  set     2048
dirs0   set     128
        endif
        if      kap0 gt 512
dirs0   set     256
        endif
        if      kap0 gt 1024
dirs0   set     512
        endif

        if      nr1 ne 0
block1  set     1024
```

```
dirs1   set     32
        if      kap1 gt 96
dirs1   set     64
        endif
        if      kap1 gt 256
block1  set     2048
dirs1   set     128
        endif
        if      kap1 gt 512
dirs1   set     256
        endif
        if      kap1 gt 1024
dirs1   set     512
        endif
        endif

        ; extended disk parameter header

        if      rw0
        dw      fwrite
        else
        dw      wprot
        endif
        dw      fread
        dw      flogin
        dw      finit
        db      0,0
fdisi0  dph     0,dpbr0,0,kap0/(block0/1024)/4

        if      nr1 ne 0
        if      rw1
        dw      fwrite
        else
        dw      wprot
        endif
        dw      fread
        dw      flogin
        dw      finit
        db      1,0
fdisi1  dph     0,dpbr1,0,kap1/(block1/1024)/4
        endif

        cseg

dpbr0   dpb     128,(bkap0*8)/bnks0,nr0*bnks0,block0,dirs0,0,8000h

        if      nr1 ne 0
dpbr1   dpb     128,(bkap1*8)/bnks1,nr1*bnks1,block1,dirs1,0,8000h
        endif


fread:  pushix                  ; save indexregister
        call    setadr          ; berrechne adresse in disi-floppy
        call    ?bank           ;
        inir                    ;
        XRA     A               ;
        CALL    ?BANK           ;
        xra     a               ;
        ldx     c,0             ;
        outp    a               ;
```

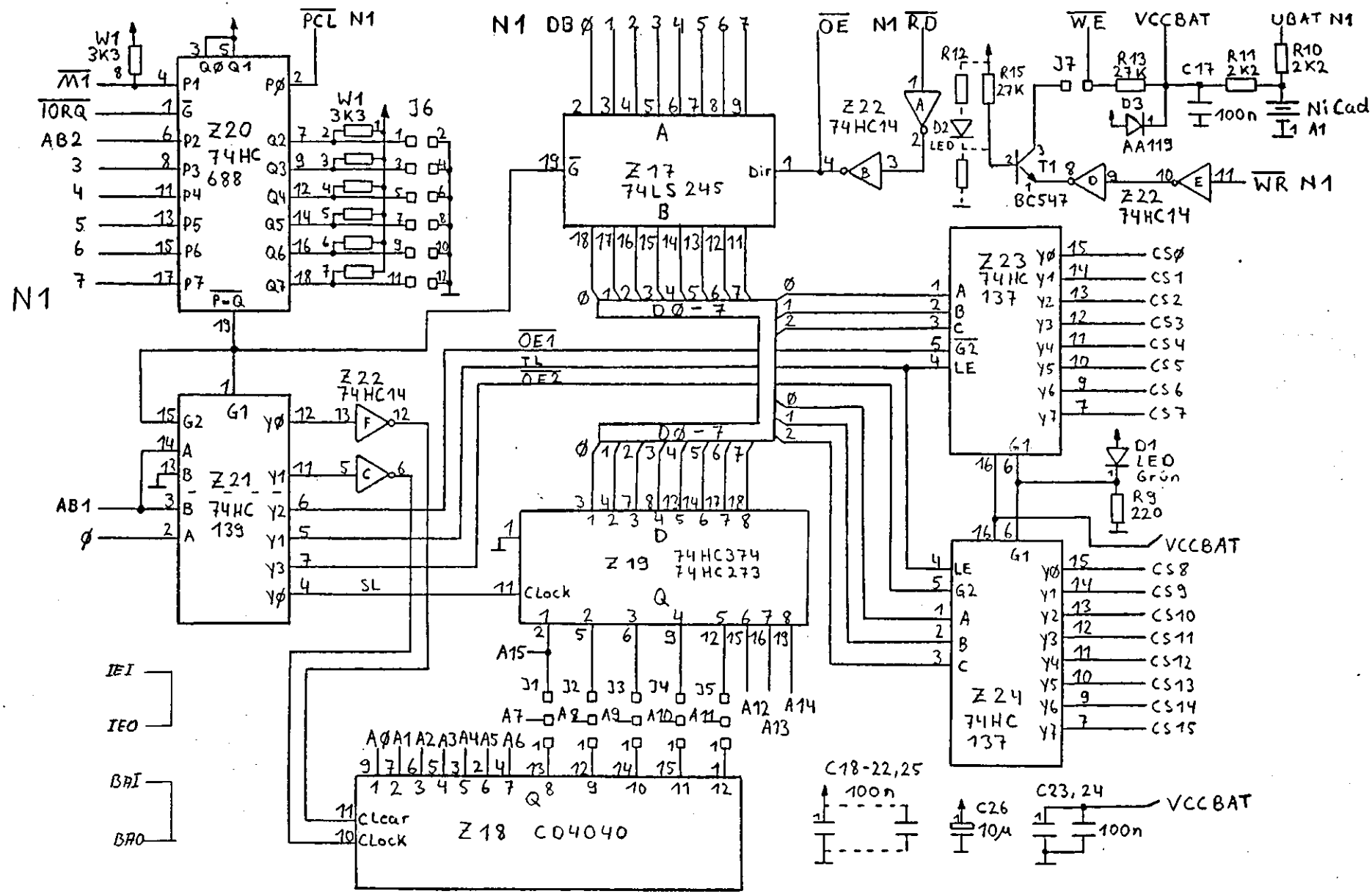
```
        popix                   ;
        ret                     ;

fwrite: pushix                  ;
        call    setadr          ; berechne adresse in disi-floppy
        call    ?bank           ;
        outir                   ;
        XRA     A               ;
        CALL    ?BANK           ;
        xra     a               ;
        ldx     c,0             ;
        outp    a               ;
        popix                   ;
        ret                     ;


        dseg                    ; Rest kann gebankt sein
wprot:
        mvi     a,02h           ; write routine bei EPROM's beschraenkt
                                ; sich auf fehlermeldung.

finit:                          ; kein init notwendig
flogin:                         ; kein login notwendig
        ret


setadr:                         ; setze dmabank und disi-adresse
                                ; eingang: §trk,§sekt
                                ;
        if      nr1 ne 0        ; wenn zwei laufwerke vorhanden sind
        lda     §rdrv           ; dann erfolgt die adressberechnung
        ora     a               ; abhaengig vom laufwerk
        jrz     setad3          ;
                                ;
                                ; ADRESSBERRECHNUNG FUER LAUFWERK 1
                                ;
        lda     §trk            ; lade spur nummer
                                ;
        if      sf1 ne 0        ; code nur notwendig, wenn banking EPROM
        mvi     b,sf1           ; teile spur durch anzahl
setad4: srlr    a               ; der EPROM Bank's
        djnz    setad4          ;
        endif                   ;
                                ; ok, in Akku steht Bausteinnummer
        adi     offs1           ; addiere offset des zweiten lw's dazu
        mov     e,a             ; rette Bausteinnummer nach E
                                ;
        if      nrcard ne 1     ; code nur notwendig wenn mehr als eine karte
        srlr    a               ; teile durch 16
        srlr    a               ; um kartennummer
        srlr    a               ; zu ermitteln
        srlr    a               ;
        mov     l,a             ; setze ix
        mvi     h,0             ; auf portliste
        dad     h               ;
        dad     h               ;
        lxix    portlist        ;
        xchg                    ;
        dadx    d               ;
        xchg                    ;
```

```
                                        ;
        else                            ; code nur notwendig wenn genau eine karte
        lxix    portlist                ;
        endif                           ;
                                        ; ok, ix zeigt auf portliste
        ldx     c,1                     ; waehle baustein aus (einer von acht)
        outp    e                       ;
        mov     a,e                     ; ermittle port fuer datenuebertragung
        ani     00001000b               ;
        ldx     a,2                     ;
        jrz     setad5                  ;
        ldx     a,3                     ;
setad5: mov     c,a                     ;
        push    b                       ; rette portadresse
                                        ;
        if      sf1 ne 0                ; code nur notwendig, wenn gebankte EPROM's
        lda     §trk                    ; setze Bank im EPROM
        outp    a                       ; *** Achtung die oberen Bits muessen fuer EPROM
        endif                           ;     bedeutungslos sein
                                        ;
        ldx     c,0                     ; setze sektor (setze implizit Bytecounter auf 0)
        lda     §sect                   ;
        ori     amsk1                   ; setze ungueltige adressleitungen
        outp    a                       ;
        pop     b                       ; lade Datenportadresse
                                        ;
        mvi     b,128                   ;
        lhld    §dma                    ;
        lda     §dbnk                   ;
        ret                             ;
        endif

                                        ; ADRESSBERRECHNUNG FUER LAUFWERK 0
setad3:                                 ;
        lda     §trk                    ; lade spur nummer
                                        ;
        if      sf0 ne 0                ; code nur notwendig, wenn banking EPROM
        mvi     b,sf0                   ; teile spur durch anzahl
setad1: srlr    a                       ; der EPROM Bank's
        djnz    setad1                  ;
        endif                           ;
                                        ; ok, in Akku steht Bausteinnummer
        mov     e,a                     ; rette Bausteinnummer nach E
                                        ;
        if      nrcard ne 1             ; code nur notwendig wenn mehr als eine karte
        srlr    a                       ; teile durch 16
        srlr    a                       ; um kartennummer
        srlr    a                       ; zu ermitteln
        srlr    a                       ;
        mov     l,a                     ; setze ix
        mvi     h,0                     ; auf portliste
        dad     h                       ;
        dad     h                       ;
        lxix    portlist                ;
        xchg                            ;
        dadx    d                       ;
        xchg                            ;
                                        ;
        else                            ; code nur notwendig wenn genau eine karte
        lxix    portlist                ;
```

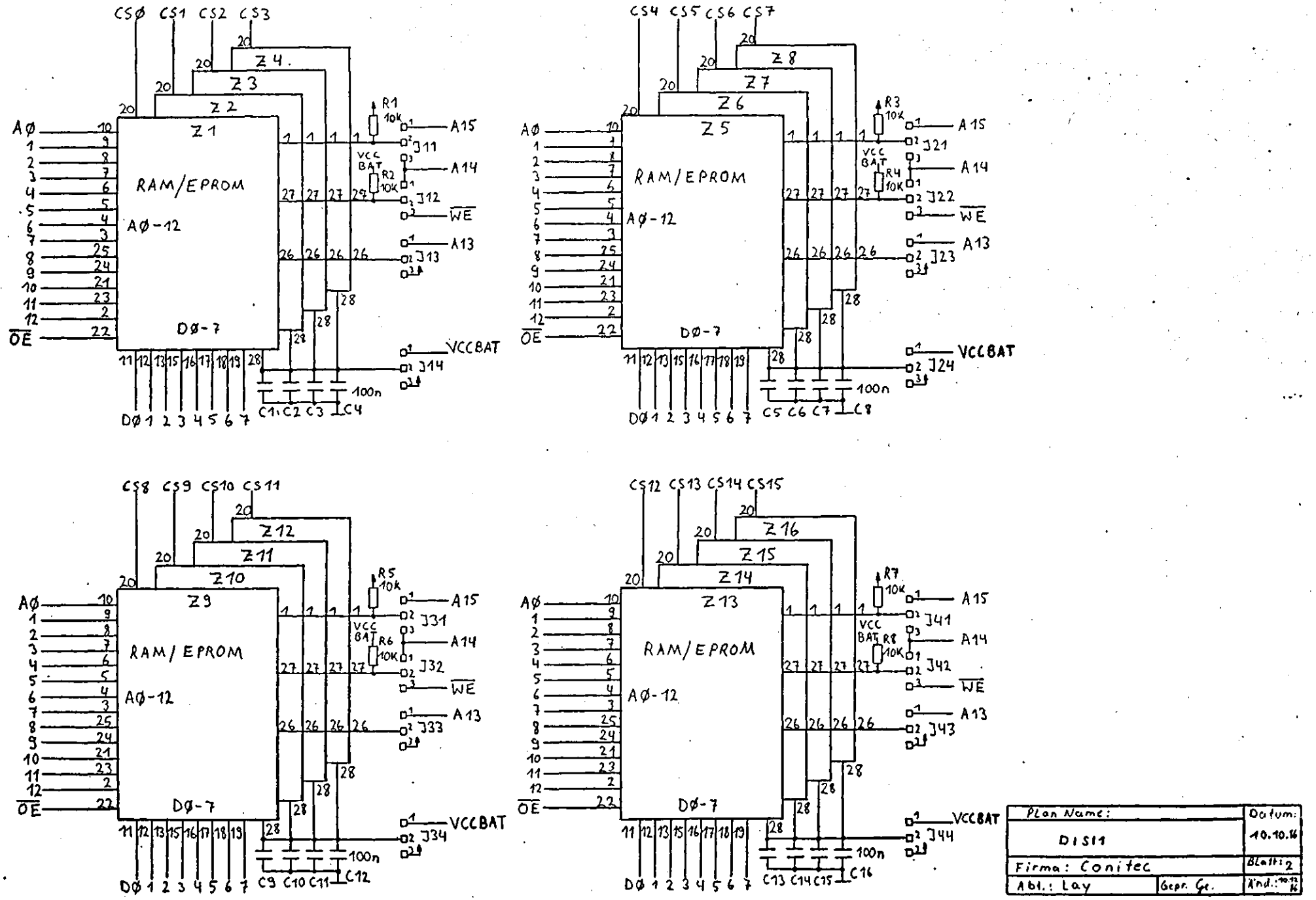
```
        endif                   ;
                                ; ok, ix zeigt auf portliste
        ldx     c,1             ; waehle baustein aus (einer von acht)
        outp    e               ;
        mov     a,e             ; ermittle port fuer datenuebertragung
        ani     00001000b       ;
        ldx     a,2             ;
        jrz     setad2          ;
        ldx     a,3             ;
setad2: mov     c,a             ;
        push    b               ; rette portadresse
                                ;
        if      sf0 ne 0        ; code nur notwendig, wenn gebankte EPROM's
        lda     §trk            ; setze Bank im EPROM
        outp    a               ; *** Achtung die oberen Bits muessen fuer EPROM
        endif                   ;     bedeutungslos sein
                                ;
        ldx     c,0             ; setze sektor (setze implizit Bytecounter auf 0)
        lda     §sect           ;
        ori     amsk0           ; setze ungueltige adressleitungen
        outp    a               ;
        pop     b               ; lade Datenportadresse
                                ;
        mvi     b,128           ;
        lhld    §dma            ;
        lda     §dbnk           ;
        ret                     ;

portlist:                       ; portadressen fuer 4 disi karten
        if      disia ne 0ffh   ;
        db      disia           ;
        db      disia+1         ;
        db      disia+2         ;
        db      disia+3         ;
        endif                   ;
        if      disib ne 0ffh   ;
        db      disib           ;
        db      disib+1         ;
        db      disib+2         ;
        db      disib+3         ;
        endif                   ;
        if      disic ne 0ffh   ;
        db      disic           ;
        db      disic+1         ;
        db      disic+2         ;
        db      disic+3         ;
        endif                   ;
        if      disid ne 0ffh   ;
        db      disid           ;
        db      disid+1         ;
        db      disid+2         ;
        db      disid+3         ;
        endif                   ;

        end
```



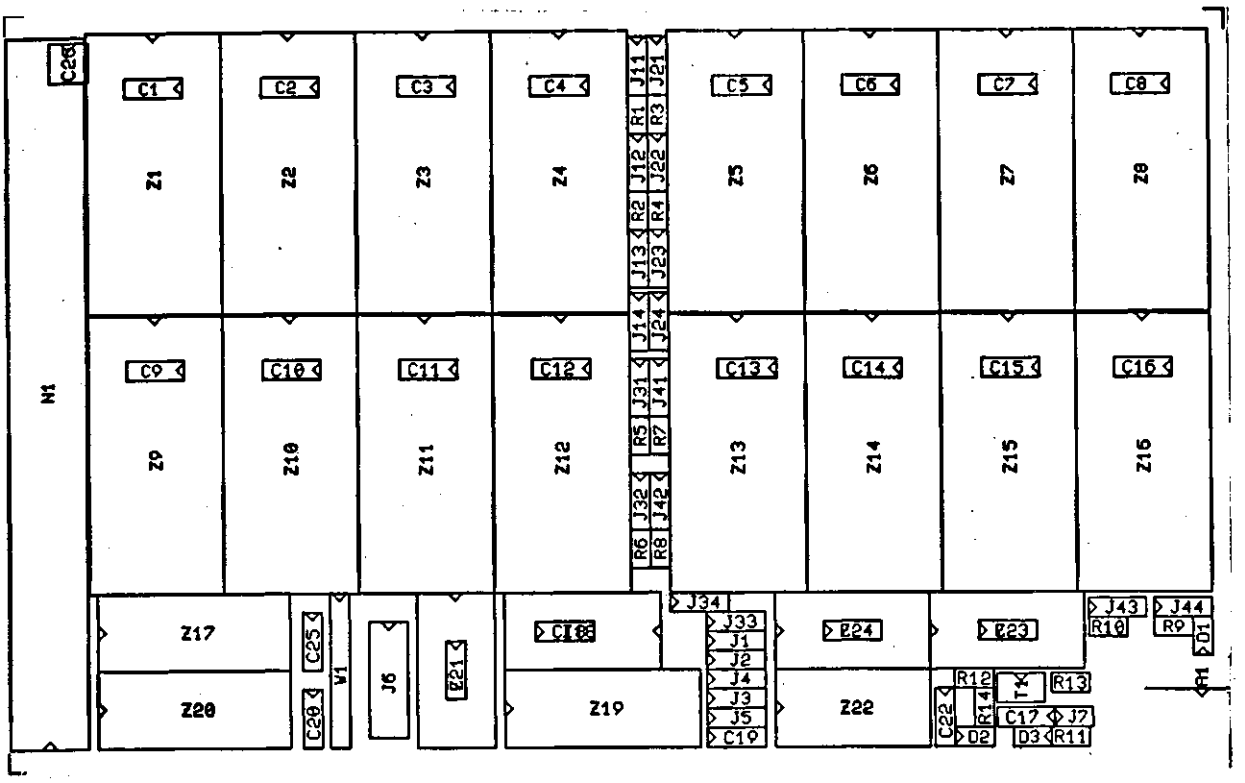
DISI Blockschaltbild



Plan Name: D1S11		Datum: 10.10.86
Firma: Conitec		Blatt: 1
Abt.: Lay	Gepr. Ge.	Änd. 10.12.86



Plan Name: DIS11		Datum: 10.10.86
Firma: Conitec		Blatt: 2
Abt.: Lay	Gepr. Ge.	Änd.:



DIBI Stückliste

-- ICs ----------

Z1 - Z16	Speicherbausteine (RAM oder EPROM)	
Z17	74 HC 245	8-fach Bustreiber
Z18	CD 4040	Binärzähler
Z19	74 HC 374	8-fach Latch
Z20	74 HC 688	8-Bit Vergleicher
Z21	74 HC 139	2-fach 2 zu 4 Dekoder
Z22	74 HC 14	6-fach Inverter
Z23/Z24	74 HC 137	3 zu 8 Dekoder mit Latch

-- sonstige Halbleiter ----------

T1	BC 547	NPN Silizium Transistor
D1	LED grün	
D2	entfällt	
D3	AA 119	Germaniumdiode

-- Kondensatoren ----------

C1 - C25	100 nF Keramik (muß unter IC-Sockel passen)
C26	10 uF / 6,3V Tantal

-- Widerstände ----------

W1	Netzwerk 7 mal 4,7 K-Ohm
R1 - R8	10 K-Ohm
R9	220 Ohm
R10	2,2 K-Ohm (siehe Text)
R11	2,2 K-Ohm
R12	entfällt
R13	27 K-Ohm
R14	entfällt
R15	27 K-Ohm

-- Präzisionssockel ----------

16x	28-polig
3x	20-polig
4x	16-polig
1x	14-polig

-- sonstiges ----------

1x	NiCad-Akku 3,6V
N1	64-polige VG-Leiste
J6	12-polige Doppelpfostenleiste
21x	3-polige Jumper
1x	2-polige Jumper
max 28x	Steckbrücken für Jumper

DISI ECB-Steckerbelegung

N1: ECB-Bus-Stecker

	a	c	Funktion	
1:	+5V	+5V	+5V:	Betriebsspannung
2:	D5	D0	GND:	Masse
3:	D6	D7		
4:	D3	D2	D0-D7:	8-Bit-Datenbus
5:	D4	A0	A0-A7:	8-Bit-Adressbus
6:	A2	A3		
7:	A4	A1	/WR:	Schreiben
8:	A5	-	/IORQ:	Ein/Ausgabe
9:	A6	A7	/PCL:	Reset
10:	-	-	UBAT:	Akkupufferung
11:	-	IEI	IEI:	Int.-Ketteneingang
12:	-	-	IEO:	Int.-Kettenausgang
13:	-	-	BAI:	DMA-Ketteneingang
14:	-	D1	BAO:	DMA-Kettenausgang
15:	-	-		
16:	-	IEO		
17:	-	-	-:	nicht benutzt
18:	-	-		
19:	-	-		
20:	-	-		
21:	-	-		
22:	-	/WR		
23:	BAI	-		
24:	UBAT	-		
25:	BAO	-		
26:	-	/PCL		
27:	/IORQ	-		
28:	-	-		
29:	-	-		
30:	-	-		
31:	-	-		
32:	GND	GND		