

Hans Wilhelm Gierlich, Michael Gilge

EPROM-Floppy mit Hardware-Boot

Teil 1: 320 KByte im EPROM für den mc-CP/M-Computer

Selbst bei Verwendung einer RAM-Floppy müssen Programme und Daten zunächst einmal von einer Floppy-Disk geladen werden. Auch jeder Warmstart erfordert einen Floppy-Zugriff mit entsprechenden Wartezeiten. Daher liegt es nahe, oft benötigte Programme wie Editor, Compiler, Assembler und Utilities in Form von Firmware bereitzuhalten. Hinzu kommt, daß EPROMs heutzutage in Kapazitäts- und Preisregionen vorgerückt sind, die einen Einsatz in einer EPROM-Floppy ermöglichen. Hier sind besonders die neuen, billigen Einmal-EPROMs zu erwähnen, die keine Löschenfenster mehr besitzen.

In Verbindung mit einer RAM-Floppy und unter Ausnutzung des DMA-Controllers ist ein Arbeiten fast ohne I/O-Wartezeiten möglich, so wird z. B. der Makroassembler M80 in 70 ms in den Arbeitsspeicher geladen und eine Kopie der gesamten 320-KByte-EPROM-Floppy in die RAM-Floppy erfolgt in ca. 7 s. Die hier vorgestellte ECB-Karte ist in erster Linie für den mc-CP/M-Computer gedacht, kann aber auch in anderen 3-Systemen eingesetzt werden.

Die Schaltung

Bild 1 zeigt die Schaltung der EPROM-Floppy. Fünf Zähler vom Typ 74LS161 generieren die Zugriffsadresse und wählen den entsprechenden EPROM-Baustein aus. Dazu werden die Zähler über den Datenbus mit einer Adresse geladen, die die Startadresse des zu übertragenden Datenblocks darstellt. Dieser Ladevorgang für die Zähler ist für jeden zusammenhängenden Datenblock nur einmal nötig, alle folgenden Lese- oder (im Falle des RAMs) Schreibzugriffe takteten die Zähler um eine Adresse weiter. Die Adressenzuordnung der Zähler sowie auch des DMA-Controllers geschieht in bekannter Weise mit einem 4-Bit-Vergleicher und einem Decoder vom Typ 74LS138. Tabelle 1 zeigt die Zuordnung von Adressen und Zählern. Die eigentli-

che Auswahl der EPROMs ist über je einen Decoder vom Typ 74 159 und 74 156 realisiert. Diese Decoder werden über die Adresse X0 aktiviert und teilen den gesamten Datenbereich in 16 KByte große Blöcke auf. Je nachdem welche EPROM-Typen (27 128 oder 27 256) in welcher Reihenfolge bestückt werden, müssen entsprechende Brücken eingelötet werden. EPROMs vom Typ 27 256 belegen zwei Blöcke. Daher muß Pin 27 dieser EPROMs mit Adresse A14 verbunden werden (Default-Einstellung). Bei Verwendung von EPROMs vom Typ 27 128 wird Pin 27 auf +5 V gelegt. Dazu wird auf der Platine die Verbindung zwischen Pin 27 und A14 aufgetrennt. Tabelle 2 zeigt die Bedeutung der Brücken, die Lage geht aus dem Bestückungsplan (Bild 4) hervor. Um die Stromaufnahme der Platine so gering wie möglich zu halten, werden alle EPROMs im Stand-by-Modus betrieben, bis sie tatsächlich angesprochen

Tabelle 1: Adreßbelegung der Platine

X $\triangleq$ DIL-Schalterstellung
X0: EPROM-Adressen 12–18, Track-Anwahl
X2: EPROM-Adressen 4–11, Sektor-Anwahl
X6: EPROM-Adressen 0–3, Bytes in Sektor
X5: EPROM-Daten
X8: DMA-Controller Adresse

Tabelle 2: Brücken auf der Platine

1. Brücken unter den EPROMs:

Steckplatz	27256 (default)	27128
IC 1	J31 $\leftrightarrow$ J41	J31 $\leftrightarrow$ J51
IC 2	J32 $\leftrightarrow$ J42	J32 $\leftrightarrow$ J52
IC 3	J33 $\leftrightarrow$ J43	J33 $\leftrightarrow$ J53
IC 4	J34 $\leftrightarrow$ J44	J34 $\leftrightarrow$ J54
IC 5	J35 $\leftrightarrow$ J45	J35 $\leftrightarrow$ J55
IC 6	J36 $\leftrightarrow$ J46	J36 $\leftrightarrow$ J56
IC 7	J37 $\leftrightarrow$ J47	J37 $\leftrightarrow$ J57
IC 8	J38 $\leftrightarrow$ J48	J38 $\leftrightarrow$ J58
IC 9	J39 $\leftrightarrow$ J49	J39 $\leftrightarrow$ J59
IC 10	J40 $\leftrightarrow$ J50	J40 $\leftrightarrow$ J60

2. Brücken unter IC 14:

a) Ausschließliche Verwendung von EPROMs vom Typ 27256 (Default-Einstellung):

für IC 1: J61 $\leftrightarrow$ J11 $\leftrightarrow$ J12
für IC 2: J62 $\leftrightarrow$ J13 $\leftrightarrow$ J14
für IC 3: J63 $\leftrightarrow$ J15 $\leftrightarrow$ J16
für IC 4: J64 $\leftrightarrow$ J17 $\leftrightarrow$ J18
für IC 5: J65 $\leftrightarrow$ J19 $\leftrightarrow$ J20
für IC 6: J66 $\leftrightarrow$ J21 $\leftrightarrow$ J22
für IC 7: J67 $\leftrightarrow$ J23 $\leftrightarrow$ J24
für IC 8: J68 $\leftrightarrow$ J25 $\leftrightarrow$ J26
für IC 9: J69 $\leftrightarrow$ J27 $\leftrightarrow$ J28
für IC 10: J70 $\leftrightarrow$ J29 $\leftrightarrow$ J30

b) Ausschließliche Verwendung von EPROMs vom Typ 27128:

für IC 1: J61 $\leftrightarrow$ J11
für IC 2: J62 $\leftrightarrow$ J12
für IC 3: J63 $\leftrightarrow$ J13
für IC 4: J64 $\leftrightarrow$ J14
für IC 5: J65 $\leftrightarrow$ J15
für IC 6: J66 $\leftrightarrow$ J16
für IC 7: J67 $\leftrightarrow$ J17
für IC 8: J68 $\leftrightarrow$ J18
für IC 9: J69 $\leftrightarrow$ J19
für IC 10: J70 $\leftrightarrow$ J20

c) Gemischte Bestückung: Beispiel

IC 1 = 27128: J61 $\leftrightarrow$ J11
IC 2 = 27256: J62 $\leftrightarrow$ J12 $\leftrightarrow$ J13
IC 3 = 27256: J63 $\leftrightarrow$ J14 $\leftrightarrow$ J15
IC 4 = 27128: J64 $\leftrightarrow$ J16

werden. Bei 6-MHz-Betrieb benötigt man dabei jedoch EPROMs mit einer Zugriffszeit von 250 ns. Ein weiterer Decoder vom Typ 74 159 unterteilt den untersten 16-KByte-Bereich in 1 KByte große Teilbereiche. An Hand der Speicheraufteilung (Bild 2) erkennt man die Lage von CP/M und die des Directory. Um Dateien im RAM ablegen zu können, muß auch ein Teil des Directory im RAM liegen. Dieser RAM-Directory-Teil muß dem EPROM-Directory vorangestellt werden, da CP/M den ersten freien Directory-Platz verwendet und die Anzahl der EPROM-Directory-Einträge variabel ist. Daraus erklärt sich die auf den ersten Blick etwas merkwürdige Deco-

Michael Gilge, Hans Wilhelm Gierlich

EPROM-Floppy mit Hardware-Boot

Teil 2

Hier der zweite Teil des Unternehmens. Zusammen mit einer RAM-Floppy wertet dieser mc-Bauvorschlag Ihren guten alten mc-CP/M-Rechner ungeheuer auf. Es ist beinahe so, daß auch ein einfacher PC da nicht mehr mitkommt.

Übersicht für die Inbetriebnahme der Karte

1. Monitor ändern, assemblieren, linken (oder patchen) und in neues EPROM programmieren
2. Neues oder geändertes BIOS assemblieren, linken und auf Diskette sichern
3. CCP und BDOS mit Befehlen nach Tabelle 5 auf Diskette ablegen
4. Directory editieren oder nach Zusammenstellung der gewünschten Files mit Programm nach Bild 11 lesen
5. Mit Debugger CCP, BDOS, BIOS und das Directory im Speicher zusammenstellen und auf Diskette sichern
6. Dieses File in ein 8-KByte-EPROM programmieren
7. EPROM auf die Platine setzen, RAM 6264 bestücken, Monitor austauschen, Reset des Rechners
8. Mit I-Befehl und nachfolgender '1' aus EPROM-Floppy booten, 'dir'-Befehl sollte alle Directoryeinträge erscheinen lassen
9. Mit Debugger 16 KByte oder 32 KByte (je nach EPROM-Typ) Programmcode in der Reihenfolge des Directory durch Laden und Verschieben zusammenstellen
10. EPROM programmieren und nach Reset testen, ob Programme vorhanden und lauffähig sind
11. Schritte 9. und 10. wiederholen bis alle Programme als Firmware vorliegen

```

MACLIB DISKDEF ;LOAD DEFINITION FOR DISKS
; Gebootet wird aus der EPROM-Floppy, EPROM und RAM-Floppy *
; werden ueber DMA-Zugriffe angesprochen *
; A> EPROM-Floppy 320 kByte - Dir *
; B> RAM-Floppy 1024 kByte - Dir *
; C> 5 1/4 80trk 780 kByte - Dir 4 trk offset *
; D> 5 1/4 80trk 780 kByte - Dir 4 trk offset *
; E> 8 " 243 kByte - Dir 2 trk offset *
; F> Winchester *

ASEG
ORG 100H
.PHASE 0EA00H

SIZE EQU 164D ;GROESSE DER EPROM-FLOPPY
FLOFREE EQU 0F900H ;Bereich fuer Puffer etc.

RFL EQU 0B7H ;RAM-FLOPPY DATEN-ADRESSE
LORFL EQU 0B8H ; LOW-BYTE ADRESSE
HIRFL EQU 0BAH ; HIGH-BYTE ADRESSE
BKRFL EQU 0BCH ; BANK-SELECT
LOSEPR EQU 006H ;EPROM-FLOPPY INSIDE SEKTOR
HISEPR EQU 002H ; SEKTOR-SELECT
TRAEPR EQU 000H ; TRACK-SELECT
EPRFL EQU 005H ; DATEN
DMAON EQU 008H ;DMA-CONTROLLER ADRESSE
STIDAT EQU 0D1H ;STI-PARALLEL-PORT

;
RETRY EQU 00AH ; Anzahl der Versuche
CPMB EQU 0D400H ; Start CP/M-Boot
BDOS EQU 0DC06H ; Beginn des BDOS
CPML EQU $-CPMB ; Laenge des CP/M
BIOS EQU $ ; Beginn des BIOS
BIAS EQU 0
NSECTS EQU CPML/128 ; Anzahl der belegten Sektoren
CDISK EQU 4 ; Speicherstelle, letztes Laufwerk
BUFF EQU 80H ; Default Bufferadresse
BUFFER EQU 0FC00H ; Buffer fuer 1 Sektor im Monitor
OFFSET EQU 4 ; fuer Minilaufwerke

; VEKTORTABELLE

WBOOT: JP BOOT ; Kalt-Start
JP WBOOT ; Warm-Start, z.B. CTRL-C
JP CONST ; Consol Status
JP CONIN ; Consol Eingabe, nach A
JP CONOUT ; Consol Ausgabe von C
JP LIST ; Druckerausgabe
JP PUNCH ; Ausgabe nach PO
JP READER ; Eingabe nach RI
JP HOME ; Track 0 selektieren
JP SELDSK ; Laufwerk selektieren
JP SETTRK ; Spuranwahl
JP SETSEC ; Sektoranwahl
JP SETDMA ; Adresse fuer Datenziel
JP READ ; Sektor lesen
JP WRITE ; Sektor schreiben
JP LISTST ; Drucker bereit?
JP SECTAN ; Sektorumsetzung log.-phys.

.8080 ; Umschalten, da Macros in 8080-Code

DISKS 6 ; EPROMFLOPPY+RAMFLOPPY+3 LAUFWERKE+WINCHEST

DISKDEF 0,0,31,0,2048,SIZE,64,0,2 ;EPROM-FLOPPY
DISKDEF 1,0,31,0,2048,512,128,0,0 ;1 MByte RAM-FLOPPY
DISKDEF 2,0,39,0,2048,390,256,256,4 ;MINI DISK

```

Bild 7. BIOS-Version für EPROM-, RAM-Floppy, 3 Disk-Laufwerke und Winchester

```

DISKDEF 3,0,39,0,2048,390,256,256,4 ;MINI DISK
DISKDEF 4,1,26,6,1024,243,64,64,2 ;8 Zoll DISK
DISKDEF 5,0,255,0,4096,1800,1024,0,1 ;WINCHESTER DISK
.Z80
MON80 EQU OF01EH ; Monitor Neustart
CR EQU 0DH
LF EQU 0AH
SIGNON: DB 0CH,CR,LF ; Meldung bei Kaltstart
DB "Welcome to the world of CP/M"
DB CR,LF,0
INITAB EQU $
DMARD: DB 0C3H ; Tabelle fuer DMA-Controller Ini
DB 061H ; WR6 RESET
DB 07FH ; WR0 B->A
DB 0 ; BLOCKLAENGE - 1
DB 014H ; WR1: A-MEMORY, INC ADDRESS
DB 028H ; WR2: B-1/0-PORT CONST ADDRESS
DB 08AH ; WR5: STOP AM BLOCKENDE
DB 0CFH ; BEFEHL 'LADEN'
L1 EQU $ - INITAB
CONST: JP 0F012H ; Console Status
CONIN: CALL 0F003H ; Console Eingabe
AND 7FH ; Paritaet = 0
RET
CONOUT: JP 0F009H ; Console Ausgabe
LIST EQU 0F00FH ; Drucker
PUNCH EQU 0F00CH ; PO
READER EQU 0F006H ; RI
BOOT: LD SP,BUFF+80H ; Kaltstart, Stack vorbelegen
; INIT RAM-BEREICH IN EPROM FLOPPY FUER DIRECTORY
INIEPR: LD A,0 ; Bereich 9k-10k ist directory
OUT (LOSEPR),A ; nur als geloescht markieren
OUT (HISEPR),A ; damit koennen Files recovered werden
LD A,02H ; setzen des ersten Bytes des
OUT (TRAEP),A ; Directories zu 'ES'
LD HL,1000H ; Dummy Adresse
LD C,EPRFL ; EPROM-Floppy Adresse
LD D,32D ; 32 Directory entries im RAM
CALL CLEAR
; INIT RAM - FLOPPY, ALLE FILES ALS GELOESCHT MARKIEREN
; BEI BATTERIEPUFFERUNG KOENNEN ROUTINEN UEBERSPRUNGEN WERDEN
IN A,(STIDAT) ; Feststellen ob RAM-Floppy-Inhalt
AND 10000000B ; noch gueltig, sonst Initialisier.
JP NZ,INI ; Bit7 = 1 dann Daten ok
INIRFL: LD A,0 ; Auf Directory positionieren
OUT (LORFL),A
OUT (HIRFL),A
OUT (BKRF),A
LD HL,1000H ; Dummy Adresse

```

Label?

```

LD C,RFL ; RAM-Floppy Port Adresse
LD D,128D ; 128 Directory Entries
CALL CLEAR
JP INI
CLEAR: LD A,0ESH ; Loeschen
OUT (C),A
LD B,31D ; 31 mal Dummy read
LD INIR ; danach wieder einmal 'ES' schreiben
DEC D ; um naechstes File zu loeschen
JP NZ,CLEAR
RET
INI: LD HL,SIGNON ; Grip-Karte initialisieren
CALL PRMSG
; INIT SEKTORBUFFER UND MONITOR KURZSCHLUSS
XOR A ; Schreibflag loeschen
LD (WRTFLG),A
LD A,OFFH ; Laufwerk undefiniert
LD (DRVAKT),A ; nach dem Booten
LD HL,WBOOT
LD (OF033H+1),HL ; Monitoreinsprung kurzschliessen
LD (OF036H+1),HL ; da Monitor zerstoeert wird
XOR A ; Anwahl von laufwerk A
LD (CDISK),A ; als Bootlaufwerk
JP GOCPM ; nun Warmstart
WBOOT: LD SP,BUFF ; Stack auf default Wert
PUSH BC
LD A,(WRTFLG) ; zuerst alten track sichern
OR A
JP Z,NOTBAC ; falls noch nicht geschehen
CALL PUTRK
NOTBAC: LD A,OFFH ; Damit ist Bufferinhalt ungueltig
LD (DRVAKT),A ; da alter Track im Buffer
LD HL,CPMB ; Ziel Adresse fuer CP/M-Laden
LD D,NSECTS/2 ; nur CCP und BDOS laden
LD A,0 ; EPROM-Floppy positionieren
OUT (LOSEPR),A ; auf 'Track' und 'Sektor'
OUT (HISEPR),A ; EPROM-Floppy daten
LD C,EPRFL
RDSEC: LD B,0 ; 256 Byte-Zaehler
LD INIR ; Sektor Zaehler
DEC D
JP NZ,RDSEC
GOCPM: LD BC,BUFF
CALL SETDMA
LD A,0C3H ; Sprung auf Warmboot ablegen
LD (0),A ; RAM-Adresse 0000h
LD HL,WBOOTE ; Warmstart Adresse
LD (1),HL
LD (5),A ; Ebenfalls BDOS Startadresse
LD HL,BDOS ; nach Adr. 0005h im RAM ablegen
LD (6),HL
LD (7*8),A ; RST7 definieren, default ist Monitor

```

10RLT

?

60H

[illegible]

```

RET      Z
DEC      B
JP       NZ,RAGAIN
LD       A,1
OR       A
RET

;
WRITE8: LD OR A,(WRITFLG)
JP       Z,WR18N
CALL     PUTTRK
JP       C,ERR10
B,RETRY
WAGAIN: PUSH BC
LD       HL,(I0D)
LD       A,(I0T)
LD       D,A
LD       A,(I0S)
LD       E,A
LD       B,2
LD       A,04H
LD       OR 00010000B
LD       C,A
CALL     FEEXEC
POP      BC
RET      Z
DEC      B
JP       NZ,WAGAIN
LD       A,1
OR       A
RET

;RAM - FLOPPY
;
RORFL: CALL
LD       LD A,0B3H
OUT      (DMACON),A
LD       A,0B3H
OUT      (DMACON),A
LD       A,087H
OUT      (DMACON),A
XOR      A
RET

WRRFL: CALL
LD       LD A,005H
OUT      (DMACON),A
LD       A,0CFH
OUT      (DMACON),A
LD       A,0B3H
OUT      (DMACON),A
LD       A,087H
OUT      (DMACON),A
XOR      A
RET

RFLADR: LD A,(I0S)
LD       B,A
LD       A,0
(LORFL),A
LD       A,B
AND      00001111B
LD       B,A
LD       A,(I0T)

; bei Fehler nochmal
; BAD SEKTOR
;
; Bufferinhalt aktuell?
;
; ggf. zurueckschreiben
; Anzahl der Schreibversuche
; Quellenadresse laden
; Spurnummer
; Sektornummer
; Schreib-Befehlscode
; single dense 8"
; bei Fehler nochmal
; BAD SEKTOR
;
; Berechnung der Zugriffsadresse
; FORCE READY DMA
; ENABLE DMA
;
; Berechnung der Zugriffsadresse
; DMA-DIRECTION: A->B
; LADEN
; FORCE READY DMA
; ENABLE DMA
;
; Sektornummer
; LSB in Carry-Flag speichern
; A loeschen fuer 128 Byte Sektor
; Carry nach A-MSB bildet
; Low-Byte fuer RAM-Floppy
; Rest der Sektornummer
; Maskierung LOW-Nibble
; Tracknummer laden
;
; low nibble nach high schieben
; und maskieren
; zusammen mit rest der Sektornummer
; wird High-Byte fuer RAM-Floppy gebildet
; High nibble von Tracknummer selektiert
; die RAM-bank
;
; Portadresse der RAM-Floppy in DMA-
; Controller Tabelle eintragen
; Anzahl der Ini-Bytes
; und Tabelle zum DMA-Controller ausgeben
;
; Bestimmung Zugriffsadresse
; FORCE READY DMA
; ENABLE DMA
;
; Bestimmung der Zugriffsadresse EPROMS
; DIRECTION: A->B
; LADEN
; FORCE READY DMA
; ENABLE DMA
;
; Sektornummer 0..31 a 128 byte = 4096BYTE
; => 12BIT Zaehler 1 und 2 setzen
; 128 BYTE = 7BIT ->nur noch 5 bit setzen
; untersten Zaehler loeschen
; 3 nach oben schieben
; 2. Zaehler
; Tracknummer Bereich 0...(DSK/2-1)
; 3.Zaehler setzen
;
; port-Adresse in Tabelle eintragen
; Anzahl der Ini-Bytes
; Ziel-Adresse
;
; WR0 B->A
; DMA-BUFFERADRESSE
; WR4: BURST-MODE
; WR4: I/O-PORT ADDRESS
; BEFEHL 'LADEN'

```

L2	EQU	\$ - DMATAB
WINCHES		
WINSET		
B.1		
WISYS		
WINSET		
B.2		
WISYS		
A.(IOS)		
L.A		
A.(IOT)		
H.A		
(INDADR),HL		
HL,0		
(INDAD2),HL		
HL,(IOD)		
DE,INDADR		
C,0		
RET		
FFEXEC:		
EXEC:		
MEEXEC:		
WISYS:		
RAM ZELLEN		
WRTFLG:		
DRVAKT:		
TRAKT:		
SEKAKT:		
ALLOC:		
MODE:		
DBANK:		
IOBP:		
ION:		
IOIOT:		
IOS:		
ALTDIV:		
INDADR:		
INDAD2:		
DEPHASE		
PHASE FLOFREE		
.8080		
ENDEF		
END		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		
EA30		
EA40		
EA50		
EA60		
EA70		
EA80		
EA90		
EA00		
EA10		
EA20		

Franzis-Software-Service

Gut bekannt dürfte er allen regelmäßigen mc-Lesern ja sein, der Franzis-Software-Service (kurz FSS): Viele in mc veröffentlichte Programme kann man hier seit langem auf preiswerten Sammeldisketten erwerben und sich so das mühsame und fehlerträchtige Abtippen sparen. Gleiches gilt übrigens auch für Programme aus unseren Schwesterzeitschriften Elo und Funkschau sowie einigen Büchern aus dem Franzis-Verlag. Doch auch Betriebssystem-Software, etwa das CP/M für den mc-CP/M-Computer oder den NDR-Klein-Computer, wurden vom FSS an diese Rechner angepaßt und sind dort erhältlich.

Inzwischen hat der Software-Service sein Angebot jedoch erheblich erweitert, und bietet – gemäß der Tradition des Franzis-Verlages – Programme für den naturwissenschaftlichen, technischen bzw. industriellen Bereich an. Da es auf diesem Gebiet in Deutschland bislang nur ein recht mageres Angebot gibt, sich ausländische Programme aber oft kaum deutschen Bedürfnissen anpassen lassen, sind Eigenentwicklungen unvermeidlich. Programme, die hohen Ansprüchen genü-

gen, lassen sich natürlich nicht von heute auf morgen entwickeln, aber mit dem im Hause Franzis vorhandenen Know-How sind die ersten Schritte bereits getan: Etwa das Mailbox-Programm PC-Mail, in dessen Entwicklung jahrelange Erfahrungen mit dem mc-Teledaten-Service TEDAS einfließen, und das einen komfortablen Betrieb einer eigenen Mailbox mit MS-DOS-, Atari-ST- oder Epson-PX-4-Computern möglich macht. Schon seit geraumer Zeit auf dem Markt befindet sich das Auto-Router-Programm zur Entflechtung von Platinen-Layouts, das in seiner zweiten Version mit erheblich größerem Leistungsumfang zu haben ist. Preiswerte Programme zum Erstellen von Schaltplänen und Werkzeichnungen werden folgen.

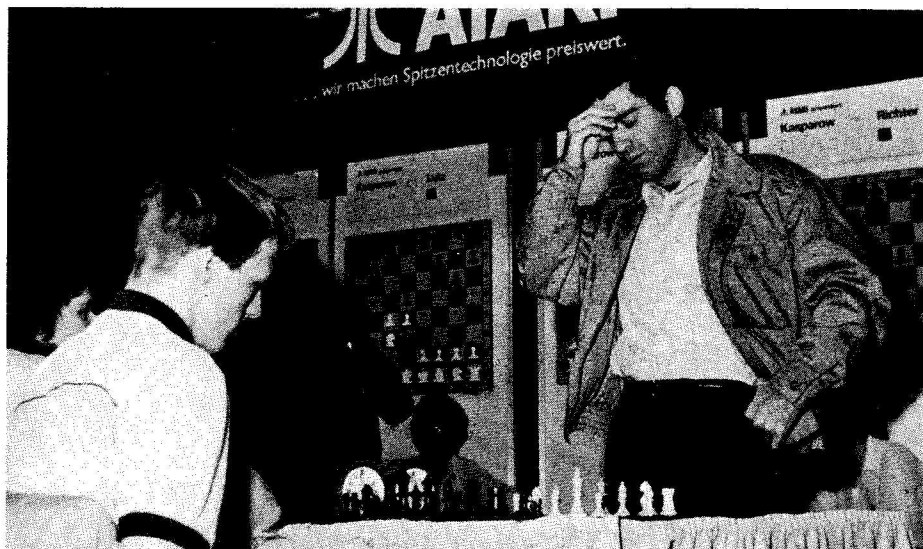
Schon erhältlich ist Franzis-Desk, eine preiswerte Alternative zu schwer überschaubaren großen Programmpaketen, das trotzdem alles in sich vereint, was man täglich braucht: Textverarbeitung, Datenbank, Adreßverwaltung, Serienbriefe, Grafik, Statistik, Dateiverwaltung, Kommunikationsprogramm, ein kleines Zeichenprogramm und ein Mini-Expertensystem.

Weltmeisterliches Schacharchiv

Eine für Schachfreunde vielversprechende Ehe sind die Firma Atari und der junge Schachweltmeister Garry Kasparow eingegangen. Das Computerunternehmen will den Weltmeister nicht nur für Werbewecke einspannen, sondern hat mit seiner Unterstützung auch ein Schacharchiv erstellt, das für die eigenen Systeme künf-

tig angeboten wird. Schachfreunden steht damit ein riesiges Reservoir an Partien zur Verfügung.

Kasparow selbst, der im Jahr zwischen 10 000 und 20 000 Partien nachspielt, verspricht sich davon ebenfalls einen großen (Zeit-)Gewinn.



Schachweltmeister Garry Kasparow bei einem Simultanturnier für Atari

mc kostet 7 DM

Das vorliegende Heft von mc kostet am Kiosk noch 6.50 DM. Ab Heft 10 wird mc 7.– DM kosten. Die Preisanhebung wurde notwendig, weil wir Sie weiter mit gewohnter mc-Qualität – interessant und immer nützlich – bedienen wollen.

Nachträge

EPROM-Floppy mit Hardware-Boot

mc 1986, Heft 7, Seite 54

Hier haben sich eine ganze Reihe von Fehlern eingeschlichen: Im Schaltplan (Seite 55, Bild 1) wird der Ausgang 4 des IC18 (74LS156) als Pin 4 angegeben, tatsächlich ist es aber Pin 7.

Im Layout der Lötseite (Seite 56, Bild 3b) darf der Pin 1 von IC16 (in der oberen rechten Ecke) nicht mit +5 V verbunden sein. Pin 7 des IC17, bisher nicht angeschlossen, muß mit Masse verbunden werden; und der bislang ebenfalls nicht angeschlossenen Pin 2 des IC15 (DMA-Controller) ist mit der Adreßleitung A4 zu verbinden, z. B. Anschluß 7a des Bus-Steckers oder Pin 15 von IC27.

Im DMA-Testprogramm (Seite 58, Bild 5) muß der Wert in der Speicherstelle 14F (Label IOT) von 4 auf 2 geändert werden, damit man auf das RAM zugreifen kann. Gleich daneben im Text sind die Adressen der Debugger-Befehle etwas durcheinander geraten, richtig muß es heißen: Dann startet man mit „G11C,129“ auf Adresse 11CH und legt einen Brakpoint auf Adresse 129. ... Die Leseoperation „G100,10D“ stellt den alten Pufferinhalt wieder her. Schließlich sollte die Versorgungs-Spannung von IC21 durch einen zusätzlichen Kondensator von 100 nF neben dem IC-Sockel abgeblockt werden.

EPROMs in MS-DOS programmiert

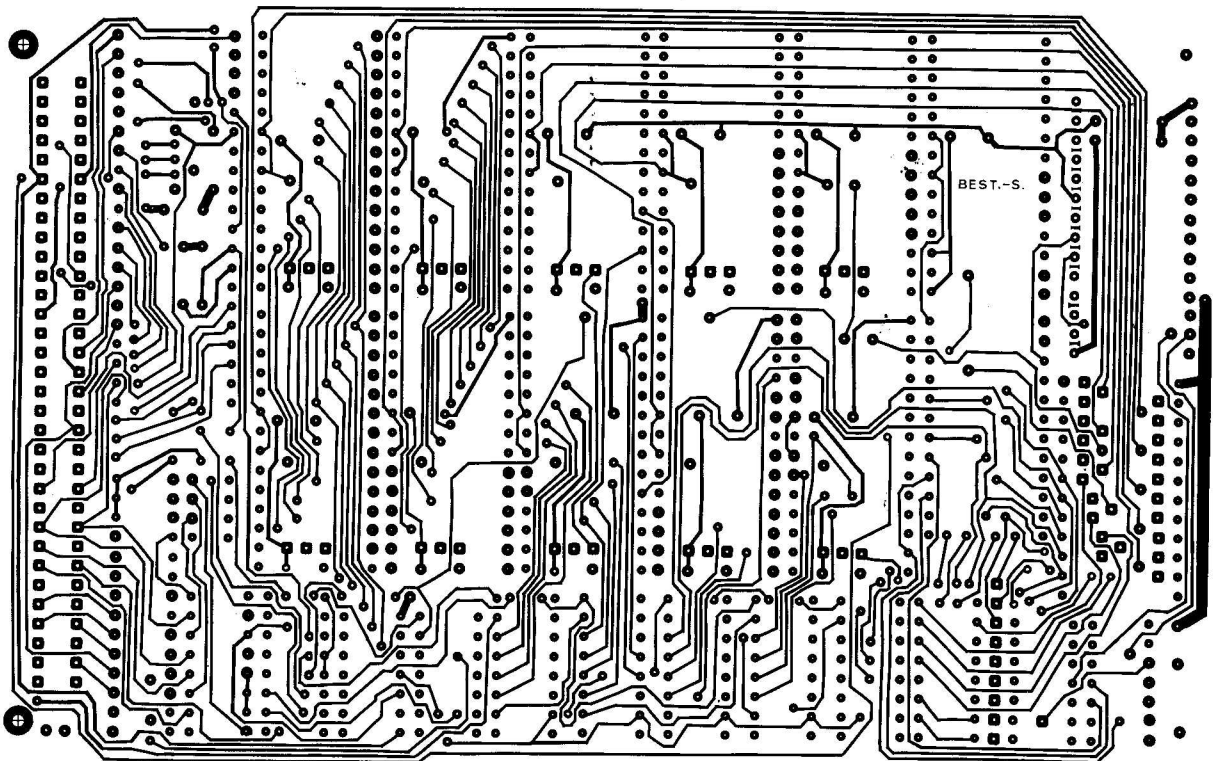
mc 1986, Heft 3, Seite 90

In dem Pascal-Programm (das auch auf der MS-DOS-Sammeldisk 3 zu beziehen ist) sind bei Verwendung des Druckerports LPT1 der Monochrom- oder Herkules-Karte die Druckerport-Adressen PORT-OUT = \$378 in \$3BC und PORT-IN = \$379 in \$3BD zu ändern, um die Schnittstelle richtig ansprechen zu können. Sonst kommt es trotz angeschlossenem EPROMmer zu der Fehlermeldung „no Access to programmer hardware“.

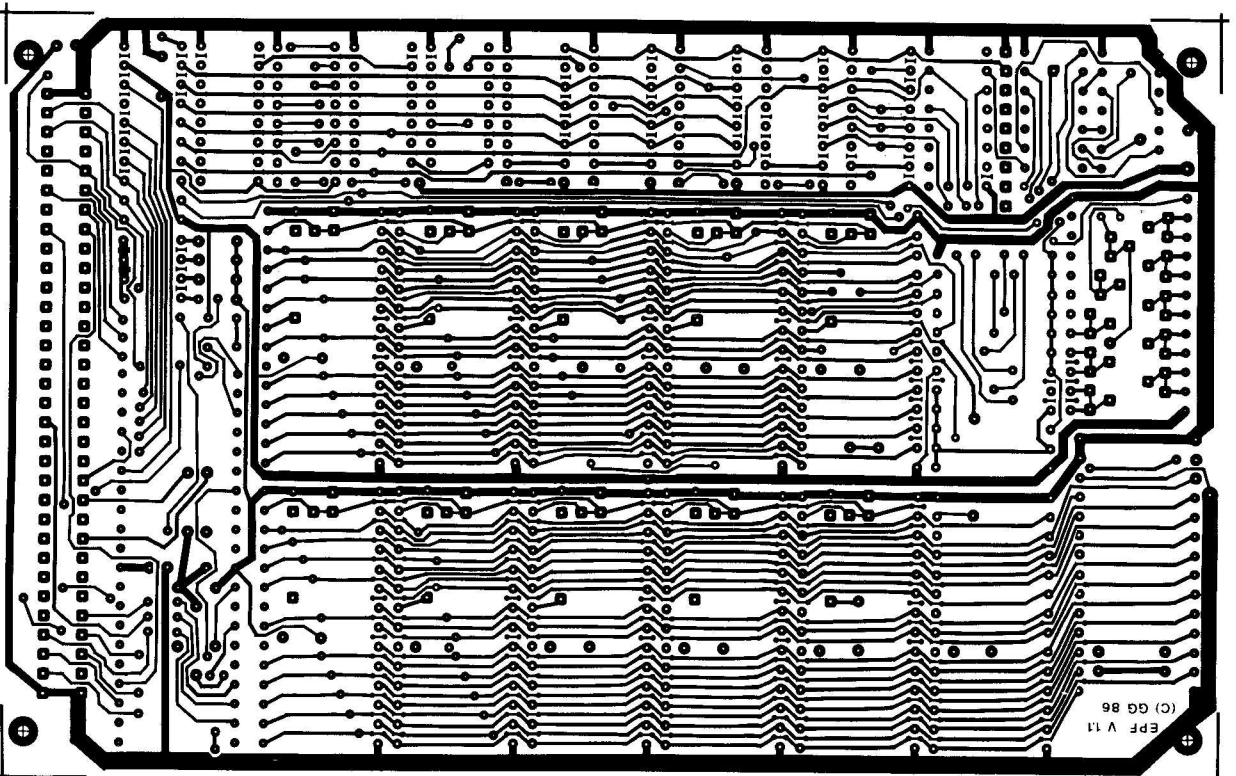


dierung von RAM und System-EPROM, und die Adressenumsetzung für das System-EPROM mit Hilfe der 4 Gatter. Diese Umsetzung bildet das EPROM-Directory, das auf Grund der Decodierung eigentlich außerhalb des EPROM-Bereiches liegt, wieder in das EPROM ab. Der DMA-Controller wird über Adresse X8 angesprochen. Erinnert sei an das Daisy-Chain-Prinzip der Interrupt-Priorisierung, in welches der DMA-Controller auch einbezogen werden muß (Leitungen IEI und IEO). Der DMA-Baustein wird leider nur als A-Version (bis zu

Bild 2. Die Speicherbelegung der Baugruppe



3a



3b

Bild 3. Platinenlayout, Bestückungsseite (3a) und Lötseite (3b)

4 MHz) angeboten. Unser Baustein (SGS) arbeitet jedoch auch tadellos bei 6 MHz, unter Umständen sollte man verschiedene Fabrikate ausprobieren. Der Refresh des Hauptspeichers ist bei DMA-Betrieb durch die sektorweise Datenanforderung durch das BDOS immer gewährleistet. Stört der DMA-Betrieb, wird dieser Baustein in der Platine nicht bestückt; ein entsprechendes BIOS ohne DMA ist im Software-Teil zu finden.

Der Aufbau der Schaltung

Bild 3 zeigt Löt- und Bestückungsseite der doppelseitig kaschierten und durchkontaktierten Europakarte. Wie aus dem Bestückungsplan (Bild 4) hervorgeht, liegen einige Widerstände und Kondensatoren unter den ICs. Daher müssen Fassungen im Boden verwendet werden, zum Beispiel Präzisionssockel. Alle notwendigen Bauteile sind in Tabelle 3 zusammengestellt. Nach dem Löten sollte die obligatorische Kontrolle auf Kurzschlüsse durch Lötbrücken erfolgen. Dann bestückt man alle TTL-ICs, das RAM und den DMA-Controller. Die so bestückte Karte wird nun, nachdem man mit dem DIP-Schalter einen freien I/O-Adreßbereich ausgewählt hat (unsere

0000		ASEG		
		ORG	100H	
		.Z80		
1000	BUFF	EQU	1000H	;DMA-BUFFERADRESSE
0006	LOSEPR	EQU	006H	
0002	HISEPR	EQU	002H	
0000	TRAEPR	EQU	000H	
0005	EPRFL	EQU	005H	
				;EPROM - FLOPPY
0100	CD 013A	REPROM: CALL	EPROAD	;UMRECHNUNG UND ADRESSIERUNG
0103	06 0E	LD	B,14D	;EIN SEKTOR
0105	0E 08	LD	C,08H	;ADRESSE DES DMA-CONTROLLERS
0107	21 010E	LD	HL,DMARD	;DMA-ADRESSE
010A	ED B3	OTIR		;BLOCK-AUSGABE
010C	AF	XOR	A	
010D	C9	RET		
010E	C3	DMARD: DB	0C3H	;WR6 RESET
010F	79	DB	079H	;WR0 B->A
0110	1000	DW	BUFF	;DMA-BUFFERADRESSE
0112	7F	DB	07FH	;BLOCKLAENGE - 1
0113	00	DB	0	
0114	14	DB	014H	;WR1: A-MEMORY, INC ADDRESS
0115	28	DB	028H	;WR2: B-I/O-PORT CONST ADDRESS
0116	C5	DB	0C5H	;WR4: BURST-MODE
0117	05	DB	005H	;WR4: I/O-PORT ADDRESS
0118	82	DB	082H	;WR5: STOP AM BLOCKENDE
0119	CF	DB	0CFH	;BEFEHL 'LADEN'
011A	B3	DB	0B3H	;FORCE READY
011B	87	DB	087H	;ENABLE DMA
011C	CD 013A	WEPROM: CALL	EPROAD	;UMRECHNUNG UND ADRESSIERUNG
011F	06 10	LD	B,16D	;EIN SEKTOR
0121	0E 08	LD	C,08H	;ADRESSE DES DMA-CONTROLLERS
0123	21 012A	LD	HL,DMAWR	;DMA-ADRESSE
0126	ED B3	OTIR		;BLOCK-AUSGABE
0128	AF	XOR	A	
0129	C9	RET		

Bild 5. Ein Testprogramm für den DMA-Controller

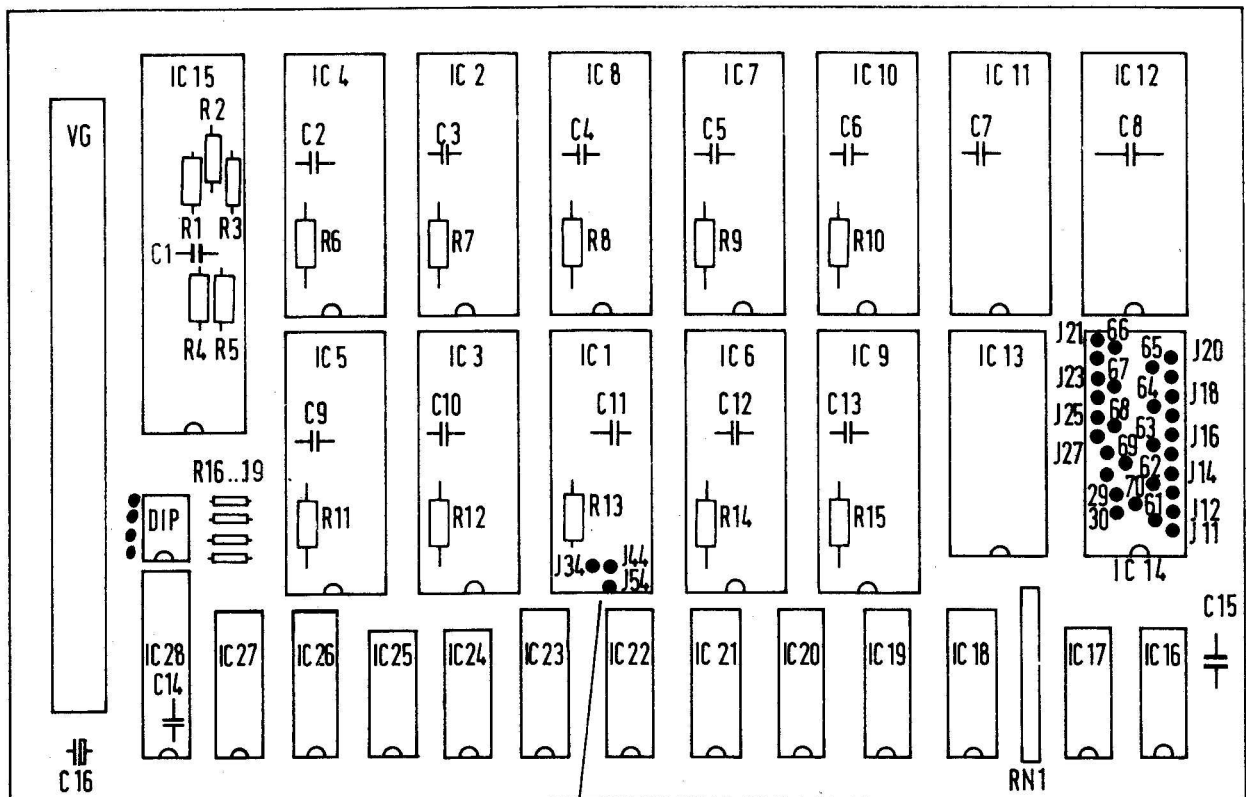


Bild 4. So wird die Platine bestückt; die Brücken unter IC 2...9 werden entsprechend denen unter IC 1 verbunden

```

012A C3      DB      0C3H ;WR6 RESET
012B 79      DB      079H ;WR0 B->A
012C 1000    DW      BUFF ;DMA-BUFFERADRESSE
012E 7F      DB      07FH ;BLOCKLAENGE - 1
012F 00      DB      0
0130 14      DB      014H ;WR1: A-MEMORY, INC ADDRESS
0131 28      DB      028H ;WR2: B-I/O-PORT CONST ADDRESS
0132 C5      DB      0C5H ;WR4: BURST-MODE
0133 05      DB      005H ;WR4: I/O-PORT ADDRESS
0134 82      DB      082H ;WR5: STOP AM BLOCKENDE
0135 CF      DB      0CFH ;BEFEHL 'LADEN'
0136 05      DB      005H ;WR0: A->B
0137 CF      DB      0CFH ;BEFEHL LADEN
0138 B3      DB      0B3H ;FORCE READY
0139 87      DB      087H ;ENABLE DMA
;
013A 3E 00 EPROAD: LD      A,0 ;SEKTORNUMMER 0...31a128 byte=4096
013C D3 06      OUT     (LOSEPR),A ;= 12BIT ZAEHLER 1 UND 2 SETZEN
013E 3A 0151    LD      A,(IOS) ;128 BYTE = 7BIT ->NUR NOCH 5 BIT
0141 CB 27      SLA     A ;UNTERSTEN ZAEHLER LOESCHEN
0143 CB 27      SLA     A ;3 NACH OBEN SCHIEBEN
0145 CB 27      SLA     A
0147 D3 02      OUT     (HISEPR),A ;2. ZAEHLER
0149 3A 014F    LD      A,(IOT) ;TRACKNUMMER BEREICH 0...(DSK/2-1)
014C D3 00      OUT     (TRAEP),A ;3.ZAEHLER SETZEN
014E C9         RET
014F 0004 IOT:   DW      4
;1 0001 IOS:    DW      1

```

Programms (DDT oder ZSID) und des Testprogramms aus Bild 5 kann nun der DMA-Controller getestet werden: Nachdem man das Testprogramm eingegeben hat, legt man 128 Byte, die man per DMA-Zugriff in das 6264-RAM transferieren möchte, ab der Bufferadresse 1000H im Speicher ab. Die Variablen IOT und IOS sind so eingestellt, daß ein Bereich innerhalb des RAMs angesprochen wird. Dann startet man mit „G100,10D“ auf Adresse 100H und legt einen Breakpoint auf Adresse 10DH. Beobachtet man mit einem Oszilloskop gleichzeitig die Busleitung 11a (BSREQ), sollte dort ein Impuls aufgetreten sein, der DMA-Aktivität verrät. Nun wird der Puffer überschrieben; die Leseoperation „G11C,129“ stellt den alten Pufferinhalt wieder her. Dieser Test sollte zunächst mit 4 MHz durchgeführt werden, da nicht alle DMA-Controller mit einer Taktfrequenz von 6 MHz arbeiten. Eine weitere Fehlerquelle kann eine falsche I/O-Adresseneinstellung sein. Ist die Karte soweit fehlerfrei, kann mit der Programmierung der EPROMs begonnen werden.

Software verwendet die Adressen 00-08), auf den Bus gesteckt. Der Rechner sollte sich wie bisher melden. Ist das nicht der Fall, überprüfe man die Richtung des Bustreibers 74LS245: Pin 1 muß Low-Pegel haben, oder man entfernt den DMA-Controller noch einmal, um den Fehler zu lokalisieren. Ist soweit alles klar, kann man mit Moni-

torbefehlen an Hand von Tabelle 4 Schreib- und Lesezugriffe ins RAM durchführen. Mit Hilfe eines Debugger-

Tabelle 3: Die Stückliste

IC 1...IC 10	27256-25 oder 27128-25
IC 11	HN6264
IC 12	2764-25 oder 27128-25
IC 13, IC 14	74159
15	Z80-A DMA
IC 16	74LS02
IC 17	74LS04
IC18	74LS156
IC 19...IC 23	74LS161
IC 24	74LS02
IC 25	74LS04
IC 26	74LS138
IC 27	74LS85
IC 28	74LS245
VG	64polige Messerleiste
DIP	4poliger DIP-Schalter
RN1	Widerstandsnetzwerk 8 x 3,9 kΩ
C1...C15	Kondensatoren 100 nF
R1...R19	Widerstände 3,9 kΩ
C16	Kondensator 47 µF
12x 28poliger Sockel	
1x 40poliger Sockel	
2x 24poliger Sockel	
1x 20poliger Sockel	
8x 16poliger Sockel	
4x 14poliger Sockel	

```

ASEG
ORG 0F7CCH
;*****
;*      MONITOR - PATCH      *
;*      *                      *
;* BOOT-Programm fuer EPROM - Floppy *
;* Ersetzt den Code im neuen Monitor *
;* ab Adresse F7CCH          *
;*      MICHAEL GILGE        *
;*****
;
FLPMSG:
DEFB 0DH,0AH
DEFM "EPROM-Boot = 1, FLO2-Boot = 2, "
DEFM "Winchester = 3, FLO1-Boot = 4 : "
FLPL EQU $-FLPMSG
;
BOOT1:
CALL INIFLP ;Vektoren etc definieren
LD HL,FLPMSG
LD B,FLPL ;Meldung ausgeben
CALL TOM ;dann Antwort abwarten
CALL C1
AND 7FH ;ohne Paritaet
LD C,A
CALL CO
CP "I" ;I-Vektor fuer Benutzerbefehl
JP Z,IEXEC
CP "1" ;EPROM-Boot
JP Z,EPRBOO
CP "3" ;Winchester Boot
JP Z,WIEX
CP "4" ;Boot mit FLO1
JP NZ,FLO2BO
LD A,1 ;1 = FLO1
LD (FLOC0),A
JP BOOTX

FLO2BO: CP '2' ;Boot mit FLO2
JP NZ,NOBOOT
XOR A

```

Bild 6. Zur Änderung des Monitorprogramms

schon den Platz für das Directory von 2 KByte und 6 KByte RAM. Nun muß noch das Directory als File auf der Diskette angelegt werden. Hier gibt es zwei Möglichkeiten:

1. Man gibt an Hand von Bild 9 das Directory von Hand ein; dabei dient Bild 9 nur als Anhaltspunkt für die Form des Directory, schließlich will man ja seine eigenen Lieblingsprogram-

me in die EPROM-Floppy schießen. Die ersten 3 Blocks sind dabei für den RAM-Bereich reserviert (File DUMMY). 2. Ein anderer Weg besteht darin, alle Files, die in die EPROM-Floppy geschossen werden sollen, auf einer neuen Diskette zusammenzustellen, z. B. auf Laufwerk B. Das Programm aus Bild 10 liest nun das Directory dieses Laufwerks. Beachten muß man allerdings, daß zuerst ein Dummy-File (6 KByte) auf der Diskette angelegt wird (siehe unter 1) und daß die Blockanzahl von Laufwerk B kleiner als 255 ist. Andernfalls werden im Directory 2 Byte pro Blockzuweisung statt nur 1 Byte verwendet (die EPROM-Floppy bleibt auf jeden Fall unter 255 Blöcken). Ausweg, falls > 255 Blöcke/Disk: Vorübergehende Änderung der Diskparameter für Laufwerk B.

2

Endlich ist es soweit. CCP, BDOS und BIOS werden mit dem Debugger zusammengestellt und in ein EPROM vom Typ 2764 programmiert. Es kann auch ein EPROM vom Typ 27128 verwendet werden, dann muß allerdings noch Pin 26 = A13 entsprechend auf Low- oder High-Pegel gelegt werden. So können zum Beispiel zwei verschiedene Betriebssystemvarianten (mit/ohne DMA) durch Umlöten von Pin 27 am System-EPROM ausgewählt werden. Nun muß es möglich sein, von der EPROM-Floppy aus zu booten. Mit dem DIR-Befehl sollten alle in das Directory eingegebenen Programmnamen erscheinen, das Programm STAT muß mindestens noch 6 KByte Platz ausweisen. Ein weiterer Test ist das Kopieren eines kleinen (≤ 6 KByte) Files in den RAM-Bereich der EPROM-Floppy und die anschließende Überprüfung, ob der Name des Files richtig in das Directory eingetragen wur-

Tabelle 6: Übersicht über Laufwerkszuordnungen

Laufwerk:	BIOS nach Bild 7	BIOS nach Bild 9
A>	EPROM-Floppy, max. 328 KByte + 7 KByte für CPM-System DMA-Control. oder Block-I/O	5¼-Zoll-Laufwerk 780 KByte, 4 Track Offset 1024 Byte/Sector, 160 Tracks
B>	RAM-Floppy, 1 MByte Zugriff auf Files mittels DMA-Control. oder Block-I/O	5¼-Zoll-Laufwerk 780 KByte, 4 Track Offset 1024 Byte/Sector, 160 Tracks
C>	5¼-Zoll-Laufwerk 780 KByte, 4 Track Offset 1024 Byte/Sector, 160 Tracks	Eprom-Floppy, max. 328 KByte + 7 KByte für CPM-System DMA-Control. oder Block-I/O
D>	5¼-Zoll-Laufwerk 780 Byte, 4 Track Offset 1024 Byte/Sector, 160 Tracks	RAM-Floppy, 1 MByte Zugriff auf Files mittels DMA-Control. oder Block-I/O
E>	8-Zoll-Laufwerk, Single density, Single side, IBM-Format	
F>	Winchester-Laufwerk	

Tabelle 7: Speichergröße für die Eintragung im BIOS

		Variable SIZE für DISKDEF im BIOS											
Anzahl 32-KByte-EPROMs:		0	1	2	3	4	5	6	7	8	9	10	
Anzahl 16 KByte EPROMs:	0	4	20	36	52	68	84	100	116	132	148	164	
	1	12	28	44	60	76	92	108	124	140	156		
	2	20	36	52	68	84	100	116	132	148			
	3	28	44	60	76	92	108	124	140				
	4	36	52	68	84	100	116	132					
	5	44	60	76	92	108	124						
	6	52	68	84	100	116							
	7	60	76	92	108								
	8	68	84	100									
	9	76	92										
	10	84											

;EPROM - FLOPPY			;RAM - FLOPPY		
REPROM:	CALL	EPROAD	RDRFL:	CALL	RFLADR
	LD	B,128		LD	B,128
	LD	C,EPRFL		LD	C,0B7H
	LD	HL,(IOD)		LD	HL,(IOD)
	INIR			INIR	
	XOR	A		XOR	A
	RET			RET	
;UMRECHNUNG UND ADRESSIERUNG			;BERECHNUNG ADRESSE		
;EIN SEKTOR			;KOPIEREN VON 128 BYTE		
;ADRESSE DER EPROM-FLOPPY			;ADRESSE DER RAM-FLOPPY		
;DMA-ADRESSE			;DMA-ADRESSE		
;BLOCK-EINGABE			;BLOCK EINGABE		
WEPROM:	CALL	EPROAD	WRRFL:	CALL	RFLADR
	LD	B,128		LD	B,128
	LD	C,EPRFL		LD	C,0B7H
	LD	HL,(IOD)		LD	HL,(IOD)
	OTIR			OTIR	
	XOR	A		XOR	A
	RET			RET	
;UMRECHNUNG UND ADRESSIERUNG			;BERECHNUNG DER ADRESSE		
;EIN SEKTOR			;KOPIEREN VON 128 BYTE		
;ADRESSE DER EPROM-FLOPPY			;ADRESSE DER RAM-FLOPPY		
;DMA-ADRESSE			;DMA-ADRESSE		
;BLOCK-AUSGABE			;BLOCK EINGABE		

Bild 7. EPROM- und RAM-Floppy-Routinen ohne DMA-Betrieb

EA00	C3	2E	EB	C3	8D	EB	C3	22	EB	C3	25	EB	C3	2B	EB	C3	".%.+..	
EA10	0F	F0	C3	0C	F0	C3	06	F0	C3	33	EC	C3	38	EC	C3	4E	3..8.N	
EA20	EC	C3	53	EC	C3	6F	EC	C3	82	EC	C3	A4	EC	C3	31	EC	S.o.....I	
EA30	C3	58	EC	00	00	00	00	00	00	00	00	00	00	F9	93	EA	B1	X.....
EA40	F9	80	F9	00	00	00	00	00	00	00	00	00	00	F9	93	EA	22	
EA50	FA	F1	F9	00	00	00	00	00	00	00	00	00	00	F9	A2	EA	77	w
EA60	FA	62	FA	00	00	00	00	00	00	00	00	00	00	F9	B1	EA	B7	b
EA70	FA	77	FA	CF	EA	00	00	00	00	00	00	00	00	F9	C0	EA	D6	w
EA80	FA	B7	FA	00	00	00	00	00	00	00	00	00	00	F9	E9	EA	C7	
EA90	FB	E6	FA	28	00	04	0F	01	A3	00	3F	00	80	00	00	00	02	s
EA00	04	20	00	04	0F	01	A3	00	3F	00	80	00	00	00	00	00	02	?
EA10	00	20	00	04	0F	01	FF	01	7F	00	C0	00	00	00	00	00	00	
EA20	01	A0	03	07	00	F2	00	3F	00	C0	00	10	00	02	00	01	01	?
EA30	1A	00	13	19	05	0B	11	17	03	09	0F	15	02	08	0E	14	01	
EA40	07	0D	13	19	05	0B	11	17	03	09	0F	15	02	08	0E	14	01	
EA50	1A	06	C2	18	04	10	16	00	01	05	1F	01	07	01	07	07	01	
EA60	FF	03	FF	00	00	01	00	00	0D	A4	57	65	6C	63	6F	07	01	Welco
EA70	6D	65	20	74	6F	20	74	68	65	20	77	6F	72	6C	64	20	me	to the world
EA80	EB	6F	20	43	50	2F	4D	0D	0A	00	C3	61	7F	00	14	28	of	CP/M...a..
EA90	6A	CF	C3	12	F0	C3	0F	C3	E6	7F	C9	63	09	F0	31	00		
EA00	01	3E	00	D3	06	D3	02	3E	02	D3	00	21	00	10	0E	05	>.....>.....!	
EA10	20	CD	61	EB	DB	D1	E6	80	C2	6E	EB	3E	00	D3	6E	a.....n>.....		
EA20	D3	BA	D3	BC	21	00	10	0E	B7	16	80	CD	61	EB	C3	8E	a.....n>.....	
EA30	EB	3E	5E	ED	79	06	1F	ED	B2	15	C2	61	EB	C9	21	F8	>.....a.....!	
EA40	EA	C1	7D	EC	AF	32	04	00	AF	32	BE	EE	3E	FF	32	BF	>.....a.....!	
EA50	EE	21	85	EB	22	34	07	22	37	F0	C3	E6	EB	3A	BE	EE	!....."4..7..>2..	
EA60	B7	CA	97	EB	CD	61	ED	3E	FF	32	BF	EE	31	80	00	0E	a>2..1..	
EA70	05	C5	01	00	D4	CD	6F	EC	0E	00	CD	38	EC	0E	00	CD	o.....8..	
EA80	4E	EC	0E	08	00	D4	CD	6F	EC	0E	00	CD	38	EC	0E	00	N.....S.....	
EA90	EC	2A	83	EE	11	80	00	19	22	83	EE	3A	C8	EE	FE	27	*....."	
EA00	DA	DD	EB	3A	7C	EE	3C	32	C7	EE	3C	FF	3C	32	C6	<2..>2..		
EA10	EE	C1	05	C2	BA	EB	01</											

Bild 8. BIOS zum Booten von Diskette

EDC0	CA	C9	ED	CD	61	ED	DA	54	ED	06	05	C5	2A	83	EE	3A	a..T..**.
EDD0	EE	C1	E7	3A	C8	EE	5F	06	02	B7	C4	F6	10	4F	CD	B2	..W:..->..>..O.
EDE0	EE	C1	C8	05	C2	EE	3E	01	B7	C9	CD	0D	EE	3E	B3		..>..>..>..>..
EDF0	D3	08	3E	87	D3	08	AF	C9	CD	0D	EE	3E	05	D3	08	3E	..>..>..>..>..>..
EE00	CD	D3	08	3E	B3	D3	08	3E	87	D3	08	AF	C9	3A	CA	EE	..>..>..>..>..>..
EE10	1F	47	3E	00	1F	D3	B8	78	E6	0F	47	3A	C7	EE	17	17	..G>..x..G>..>..
EE20	17	17	E6	F0	B0	D3	BA	3A	C7	EE	E6	F0	D3	BC	3E	B7	..:..:..:..:..:..>..
EE30	32	86	EE	06	06	06	08	21	82	EE	ED	B3	C9	CD	5F	EE	2....!..>..>..>..>..>..
EE40	3E	B3	D3	08	3E	87	D3	08	AF	C9	5F	EE	3E	05	D3		..>..>..>..>..>..>..
EE50	08	3E	CF	D3	08	3E	B3	D3	08	3E	87	D3	06	AF	C9	3E	..>..>..>..>..>..>..
EE60	00	D3	06	3A	CA	EE	CB	27	CB	27	CB	27	D3	02	CA	C7	..:..:..:..:..:..>..2..
EE70	EE	D3	00	3E	05	32	86	EE	06	06	08	21	82	EE	ED		..:..:..:..:..:..>..>..>..>..>..>..
EE80	B3	C9	19	00	C5	05	CF	CD	98	EE	06	01	C3	BB	EE		..:..:..:..:..:..>..>..>..>..>..>..
EE90	CD	98	EE	06	02	C3	BB	EE	3A	CA	EE	6F	3A	C7	EE	67	..:..:..:..:..:..>..>..>..>..>..>..
EEA0	22	CA	EE	21	00	00	22	CC	EE	2A	83	EE	11	CA	EE	0E	..:..:..:..:..:..>..>..>..>..>..>..
EEB0	00	C9	C3	21	F0	C3	24	F0	C3	27	F0	C3	2A	F0	00	00	..:..:..:..:..:..>..>..>..>..>..>..
EEC0	00	00	00	00	00	80	01	04	01	00	00	00	00	00	00	00	..:..:..:..:..:..>..>..>..>..>..>..
EEF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	..:..:..:..:..:..>..>..>..>..>..>..

1C00	E544554D	4D592020	2048494C	00000030	.DUMMY	FILE...
1C10	01020300	00000000	00000000	00000000		
1C20	00444920	20202020	20C34F4D	00000016	.DI	OM...
1C30	04050000	00000000	00000000	00000000		
1C40	00504950	20202020	20C34F4D	0000003A	.PIP	OM...
1C50	06070809	00000000	00000000	00000000		
1C60	00535441	54202020	20C34F4D	00000029	.STAT	OM...
1C70	0A0B0C00	00000000	00000000	00000000		
1C80	00535542	4D495420	20C34F4D	0000000A	.SUBMIT	OM...
1C90	0D000000	00000000	00000000	00000000		
1CA0	00535745	4E503330	20C34F4D	0100004F	.SWEEP30	OM...
1CB0	0E0F1011	12131415	16171819	1A000000		
1CC0	005A5349	44202020	20C34F4D	00000050	.ZSID	OM...P
1CD0	1B1C1D1E	1F000000	00000000	00000000		
1CE0	00444454	5A202020	20C34F4D	0000004F	.DDTZ	OM...O
1CF0	20212223	24000000	00000000	00000000	!#\$	
1D00	00444553	504F4F4C	20C34F4D	00000014	.DESPOOL	OM...
1D10	25260000	00000000	00000000	00000000	%&.	
1D20	004C3830	20202020	20C34F4D	0000003B	.L80	OM...;
1D30	2728282A	00000000	00000000	00000000	'(*)	
1D40	004D3830	20202020	20C34F4D	0100000E	.M80	OM...
1D50	2B2C2D2E	2F303132	33000000	00000000	+,-./0123	
1D60	00404241	53494320	20C34E3F	0100003B	.MBASIC	OM...;
1D70	34353637	38393A3B	3C3D3E3F	00000000	456789;,<=>?	
1D80	00575320	20202020	20C34F4D	0100000A	.WS	OM...
1D90	40414243	44454647	48000000	00000000	\$%&@DEFH	
1DA0	0057534F	564C5931	20CF5652	0100006B	.WSMSG	VR...k
1DB0	494A4B4C	4D4E4F50	51525354	55585700	IJKLMNOPQRSTUWV	
1DC0	0057534F	564C5931	20CF5652	01000080	XYZOVL^	VR...
1DD0	5959595B	5ED5E5F	60616263	64656667	XZYAOUL^	abcdefg
1DE0	0057534F	564C5931	20CF5652	02000006	.WSOVL^	VR....
1DF0	68000000	00000000	00000000	00000000	h.....	
1E00	00463830	20202020	20C34F4D	01000055	.F80	OM...U
1E10	636A6B6C	6D6E6F70	71727374	75760000	ijklmnopqrstuv...	
1E20	0046465F	4C494220	70F0454C	0100004F	.FORLIB	EL...O
1E30	7778797A	7B7C7D7E	7F808182	83000000	wxyzau8.	
1E40	00544C49	53542020	20C34F4D	00000077	.TLIST	OM...w
1E50	84858687	88898A8B	00000000	00000000		
1E60	00545552	424F2020	20C34F4D	0100006E	.TURBO	OM...r

Bild 9. Ein Beispiel-Directory der EPROM-Floppy


```

1E70 8C8D8E8F 90919293 94959697 98999A00 .....
1E80 00545552 424F2020 20CD5347 0000000C .TURBO .SG....
1E90 9B000000 00000000 00000000 00000000 .....
1EA0 00545552 424F2020 20CF5652 00000008 .TURBO .VR....
1EB0 9C000000 00000000 00000000 00000000 .....
1EC0 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1ED0 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1EE0 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1EF0 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1F00 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1F10 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1F20 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1F30 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1F40 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1F50 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1F60 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1F70 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1F80 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1F90 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1FA0 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1FB0 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1FC0 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1FD0 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1FE0 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....
1FF0 E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5 .....

```

```

;*****
; PROGRAMM BRINGT DAS DIRECTORY DER *
; DISKETTE IM LAUFWERK B NACH 100H *
; BIS 4FFH ENTSPR. 1K DIRECTORY *
;*****
;
.Z80
CONOUT EQU 0F009H
;
LD HL,100H ; NACH 100 LADEN
LD D,2 ; DIREKTORY AUF TRACK 2
LD E,1 ; 1 SEKTOR a 1024 Bytes
RDSEC: PUSH HL
PUSH DE
PUSH BC
LD C,11010010B ;LAUFWERK 1 DOUBLE DENSE MINI
LD B,1 ;READ
CALL 0F027H
POP BC
POP DE
POP HL
LD HL,MSG
CALL PRMSG
JP 0
PRMSG: LD A,(HL)
OR A
RET Z
PUSH HL
LD C,A
CALL CONOUT
POP HL
INC HL
JP PRMSG
;
MSG: DB 0DH,0AH
DB "LOADING DIRECTORY OF DRIVE B TO 100h - 04FFh"
DB 0DH,0AH
DB "SAVE IT WITH 'SAVE 4 DIR.DAT' "
DB 0
;
END

```

D2 11000010 E2

C = 11010010

doppelschlägig LW

```

0500 21 00 01 16 02 1E 01 E5 D5 C5 0E D2 06 01 CD 27 !.....
0510 F0 C1 D1 E1 21 2A 05 CD 1D 05 C3 00 00 7E B7 C8 ....!*.....B..
0520 E5 4F CD 09 F0 E1 23 C3 1D 05 0D 0A 4C 4F 41 44 .O....#.....LOAD
0530 49 4E 47 20 44 49 52 45 43 54 4F 52 59 20 4F 46 ING DIRECTORY OF
0540 20 44 52 49 56 45 20 42 20 54 4F 20 31 30 30 68 DRIVE B TO 100h
0550 20 2D 20 30 34 46 46 68 0D 0A 53 41 56 45 20 49 - 04FFh..SAVE I
0560 54 20 57 49 54 48 20 27 53 41 56 45 20 34 20 44 T WITH 'SAVE 4 D
0570 49 52 2E 44 41 54 27 20 00 04 00 00 00 00 00 00 IR.DAT' .....

```

Bild 10. Das Programm liest das Directory von Laufwerk B

de und das File selber richtig abgespeichert wurde.

Aus Software wird Firmware

Ist soweit alles zufriedenstellend verlauten, kann mit der Fleißarbeit begonnen werden, die gewünschten Programme in EPROMs abzulegen. Dazu stellt man die Programme hintereinander mit dem Debugger zusammen. Es ist zu beachten, daß die kleinste Einheit 1 Block, also 2 KByte ist. Beispiel:

Programm A, 12 KByte: 1000H-3FFFH
 Programm B, 2 KByte: 4000H-47FFFH
 Programm C, 30 KByte: 4800H-9000H

Das ist mehr als $1 \times 27\,256$, der Rest des Programmes C wird zu Beginn des folgenden EPROMs programmiert.

Zu erwähnen bleibt noch, daß der DMA-Controller natürlich systemweit verwendet werden kann, also auch z. B. für Floppyzugriffe. Dies erfordert jedoch einige Änderungen im Monitor. Insgesamt gesehen ist die EPROM-Floppy eine echte Alternative zu 7- bis 8mal teureren Festplattenlaufwerken und diesen in puncto Geschwindigkeit weit überlegen, immerhin fördert der DMA-Controller die Daten mit 1 MByte/s. Aber auch ohne DMA ist die Arbeitsgeschwindigkeit atemberaubend und wird nur durch die Bildschirmausgabe und durch die echte Rechenzeit des Prozessors begrenzt. Durchkontaktierte Platinen mit Lötstopmmaske sind von der Firma GES, Kempten, erhältlich.

Im nächsten Heft das BIOS

Literatur

- [1] Klein, R.-D.: Mini-Floppy-Anschluß. mc 1983, Heft 2, S. 66.
- [2] Das mc-CP/M-Sonderheft. Franzis-Verlag.
- [3] Brunner, Chr.: Berechnung der Disk-Parameter-Tabellen für CP/M-Rechner. mc 1985, Heft 7, S. 48.
- [4] Sternberg, G.: Eine RAM-Floppy für den mc-CP/M-Computer. mc 1985, Heft 6, S. 86.
- [5] CP/M-Alteration Guide. Digital Research.
- [6] Z80 DMA, Technical Manual. Zilog Inc.
- [7] Datenblätter der Firmen NEC und Hitachi für EPROMs bzw. RAMs der Typen 2764, 27128, 27256, 6264.
- [8] Klein, R.-D.: Das BIOS für FLO-2. mc 1985, Heft 3, S. 90.