



## Inhaltsverzeichnis

1.	Einleitung	3
2.	I/O-Adressraum	4
3.	Memory-Adressraum	5
	3.1 Memory-Bereiche	5
	3.2 Adress-Erweiterung (Banking)	5
4.	Schnittstellen	8
	4.1 ECB-Bus	8
	4.2 V24-Schnittstellen	9
	4.3 Centronics-Schnittstelle	11
5.	Programmbeispiele	12
	5.1 SIO für seriell I/O	12
	5.2 PIO als Centronics-Schnittstelle	14
	5.3 CTC als Baudratengenerator	15
	5.4 Adress-Erweiterung (Banking)	16
6.	Bestückung	17
	6.1 Bestückungsliste	17
	6.2 Bestückungsplan	18
	6.3 Inhalt der I/O Steuerproms	19
	6.4 Jumperfunktion	21
7.	Schaltplan	22

1. Einleitung

Die CPU II ist eine stand-alone CPU-Karte im Europaformat. Es wurde bei der Entwicklung dieser Karte auf ein breites Einsatzspektrum vom Ein-Platinen-Computer bis zum Floppy-Disk-System Wert gelegt. Auf der Karte sind alle Komponenten vereinigt, die zum Betrieb notwendig sind.

## Versorgung:

5 Volt ca. 1 A  
12 Volt ca. 50 mA

## Systemtakt:

4 MHz (CP/M PLUS oder ZDOS)  
5 MHz (CP/M PLUS)  
6 MHz (CP/M PLUS)

## Abmessungen:

Einfach-Europa-Karte, minimale Mechanik

## Komponenten:

280A(B) CPU als Zentraleinheit  
280A(B) PIO als Centronicsschnittstelle  
280A(B) SIO oder DART für 2 V24-Schnittstellen (RxD, TxD, CTS, RTS)  
DC/DC-Wandler für -12 Volt  
280A(B) CTC zur Baudratenerzeugung  
Baudraten softwaremäßig einstellbar von 50 bis 9600 oder 19200 Baud)  
2716-2764 EPROM als Urlader ausblendbar  
64 KByte dyn. RAM, DMA-fähig, DESELECT-Eingang, BANKING Adressen - A16 A17 A18 A19, 1MByte in Blöcken zu 16K beliebig in den Arbeitsspeicher einblendbar  
vollständige Buspufferung  
IM2-fähig, IEI/IEO Look-Ahead-Logik  
alle Schnittstellen gepuffert und 1:1 auf Normstecker quetschbar  
voll erweiterbar (ECB-Bus)

2. I/O-Adressraum

Die CPU II erzeugt 256 I/O-Adressen (Z80-CPU). Die Karte selbst belegt davon insgesamt 16 Adressen (von 0E0 bis 0EF). Alle anderen I/O-Adressen stehen dem Anwender für Erweiterungen zur Verfügung. Da die Karte über eine Look-ahead-Logik für IEI / IEO verfügt, sind auch weitere ZILOG-ICs im Interrupt Mode 2 verwendbar. Selbstverständlich sind auf Wunsch auch andere Adressen wählbar. Im folgenden sind die belegten Adressen im einzelnen dargestellt:

## I/O-Adressen:

0E0	SIO Data A	V24-Schnittstelle
0E1	SIO Command A	
0E2	SIO Data B	V24-Schnittstelle
0E3	SIO Command B	
0E4	PIO Data A	Centronics-
0E5	PIO Command A	Schnittstelle
0E6	PIO Data B	
0E7	PIO Command B	
0E8	CTC Channel 0	Baudrate V24 A
0E9	CTC Channel 1	frei
0EA	CTC Channel 2	Baudrate V24 B
0EB	CTC Channel 3	frei
0EC	Paging-Register Block 1	0000H-3FFFH
0ED	Paging-Register Block 2	4000H-7FFFH
0EE	Paging-Register Block 3	8000H-BFFFH
0EF	Paging-Register Block 4	C000H-FFFFH

### 3. Memory-Adressraum

#### 3.1 Memory-Bereiche

Auf der Karte sind 64-KByte dynamisches RAM enthalten. Dekodiert und erzeugt werden 20 Bit Adressen. Das Eprom belegt im eingblendeten Zustand ( nach RESET ) die Adressen von 00000 bis 07FFFH, das RAM belegt die Adressen 00000...0FFFFH und ist von außerhalb im DMA-Betrieb erreichbar. Die Bussteuerung ermöglicht den Interrupt Mode 2. ( RETI kann gelesen werden.)

Durch den DESELECT-Eingang ist es möglich, parallel zum Hauptspeicher weitere Speicherkarten (z.B. EPROM- oder Memory-Mapped-Video-Karten) zu betreiben. Das Select-Signal dieser Karten ist dann mit dem DESELECT der CPU II zu verbinden.

Die meisten Karten verfügen über einen gepufferten Datenbus mit entweder 74LS245 oder 2 \* 8216 als Treibern. Diese Treiber werden beim Zugriff auf die Karte selektiert. Der SELECT-Eingang der Treiber (Pin 1 beim 8216 bzw. Pin 19 beim 74LS245) wird bei Zugriff auf die Karte LOW. Dieses Signal kann fast immer (evtl. Timing beachten!) direkt mit DESELECT der CPU II verbunden werden. Bei mehreren Karten ist zu beachten, daß DESELECT ein OPEN-COLLECTOR-Signal ist (aktiv low!). Karten, die die Treiber immer selektiert haben, und nur die Richtung (Direction) der Treiber steuern, sind nicht so einfach zu verwenden. Bei ihnen muß in ihrer Schaltung ein CARD-SELECT-Signal gesucht werden. Dieses kann dann für DESELECT verwendet werden.

#### 3.2 Adress-Erweiterung (Banking)

Mit Hilfe der IC's 1, 8 und 14 wird eine 4-Bit-Adresse (A16 bis A19) generiert, die es ermöglicht, 1 MByte Speicher zu adressieren. Zu diesem Zweck werden mit IC 8 (4X4 Bit Register-File) 4 Pseudoadressen A16 bis A19 erzeugt. Um die Memory, die mit diesen Adressen selektiert wird, auch benutzen zu können, muß sie in kleinen Portionen zerlegt in den Arbeitsspeicher eingblendet werden. Deshalb werden die Adressen A14' und A15' mit IC 14 (4X4 Bit Register-File) erzeugt. Die Adressen A14 und A15 von der CPU selektieren nur noch die entsprechende 6 Bit Zelle der Register-Files und legen deren Inhalt auf den Bus. (A14', A15', A16, A17, A18 und A19) Beim Beschreiben der Register-Files wird das Register mit D2 bis D7 geladen. Welches der 4 Register beschrieben wird bestimmt die entsprechende I/O-Adresse. Bei Boot-Betrieb (nach Reset) werden die Adressen A16 bis A19 mit Hilfe von IC1 auf Ground gelegt und die CPU-Adressen (A14 und A15) auf den Bus gelegt. Es muß im Boot sichergestellt werden, daß das Register-File für Banking vorbe-

Realitäts!  
! 84 EPROM!

Das Registerfile besteht aus 4 6 Bit breiten Speicherzellen, die unabhängig voneinander beschrieben und gelesen werden können. Zum Lesen wird mit den CPU-Adressen A14 und A15 selektiert. Zum Schreiben wird mit A0 und A1 selektiert. Gelesen wird immer, geschrieben mit entsprechenden Portadressen. Somit hat man den logischen 64K-Adressraum der CPU in 4 Kacheln zu je 16 KByte aufgeteilt:

I/O-Adresse	CPU (logisch)	RAM (physikalisch)
0EC	0000H..3FFFH	
0ED	4000H..7FFFH	
0EE	8000H..BFFFH	
0EF	C000H..FFFFH	

Die Zuordnung zwischen logischer und physikalischer Adresse geschieht durch das Registerfile. Für jede Kachel ist in das zugehörige Register die physikalische Adresse (A19..A14 = D7..D2) einzutragen.

Einige Beispiele:

1. normale Aufteilung (zusammenhängendes RAM in Page 0)

I/O-Adresse	CPU (logisch)	RAM (physikalisch)
0EC = 00H	0000H..3FFFH	00000H..03FFFH
0ED = 04H	4000H..7FFFH	04000H..07FFFH
0EE = 08H	8000H..BFFFH	08000H..0BFFFH
0EF = 0CH	C000H..FFFFH	0C000H..0FFFFH

2. CP/M PLUS Bank 1 selektiert (16 KByte COMMON)

I/O-Adresse	CPU (logisch)	RAM (physikalisch)
0EC = 10H	0000H..3FFFH	10000H..13FFFH TPA
0ED = 14H	4000H..7FFFH	14000H..17FFFH TPA
0EE = 18H	8000H..BFFFH	18000H..1BFFFH TPA
0EF = 0CH	C000H..FFFFH	0C000H..0FFFFH COMMON

3. 4 mal die gleiche 16 KByte Kachel

I/O-Adresse	CPU (logisch)	RAM (physikalisch)
0EC = 00H	0000H..3FFFH	00000H..03FFFH
0ED = 00H	4000H..7FFFH	00000H..03FFFH
0EE = 00H	8000H..BFFFH	00000H..03FFFH
0EF = 00H	C000H..FFFFH	00000H..03FFFH

## 4. Overlays tauschen (Overlay-Area von 4000h bis BFFFh)

I/O-Adresse	CPU (logisch)	RAM (physikalisch)	
0EC = 00H	0000H..3FFFH	00000H..03FFFH	COMMON
0ED = 04H	4000H..7FFFH	04000H..07FFFH	OVERLAY 1
0EE = 08H	8000H..BFFFH	08000H..0BFFFH	OVERLAY 1
0EF = 0CH	C000H..FFFFH	0C000H..0FFFFH	COMMON
0EC = 00H	0000H..3FFFH	00000H..03FFFH	COMMON
0ED = 04H	4000H..7FFFH	10000H..13FFFH	OVERLAY 2
0EE = 08H	8000H..BFFFH	14000H..17FFFH	OVERLAY 2
0EF = 0CH	C000H..FFFFH	0C000H..0FFFFH	COMMON
0EC = 00H	0000H..3FFFH	00000H..03FFFH	COMMON
0ED = 04H	4000H..7FFFH	18000H..1BFFFH	OVERLAY 3
0EE = 08H	8000H..BFFFH	1C000H..1FFFFH	OVERLAY 3
0EF = 0CH	C000H..FFFFH	0C000H..0FFFFH	COMMON

Es lassen sich also beliebige 6 Bit Kombinationen in die Register-Files einschreiben. Somit ergibt sich ein frei, aus 16K Blöcken, zusammenstellbarer Arbeitsspeicher. Es ist auf diese Weise leicht möglich, auf einfache Art Overlays schnellstmöglich in den Arbeitsspeicher einzublenden. Anhand der Darstellung erkennt man, daß die CPU-Adressen A14 und A15 nicht den gleichen logischen Zustand haben müssen, wie A14' und A15'. Daraus ergibt sich eine völlig freie Memory-Zuweisung, die es im Extremfall ermöglicht, viermal den gleichen 16K-Block einzublenden (Beispiel 3).

Bei Verwendung von Video-Karten, die mittels I/O-Befehlen ihre Memory adressieren (High-Adress-Byte=Inhalt von Register B und Low-Adress-Byte=Portadresse), muß vor dem Out- oder In-Befehl sichergestellt werden, daß A14=A14' und A15=A15' sind.

Die Banking-Register sind für eine DMA-Karte nicht existent! Das heißt, der DMA schreibt immer in die physikalischen Blöcke. In Verbindung mit der FDC 8/5 ist zu beachten, das ihr DMA immer in die Page schreibt, die im Register (mit der I/O-Adresse) 0EFH steht. Hierbei werden nur die Adressen A16 bis A19 aus dem Register genommen, die Adressen A0 bis A15 erzeugt der 280 DMA der FDC 8/5-Karte.

## 4. Schnittstellen

## 4.1 Der ECB-Bus (Stecker S1 = UG 64)

## Busbelegung:

	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
$\overline{\text{WAIT}}$	o	10	o	A16 **
$\overline{\text{BUSRQ}}$	o	11	o	IEI
$\overline{\text{BAI}}$	o	12	o	A17 **
+12V	o	13	o	A18 **
** A19	o	14	o	D1
	o	15	o	
2xCLOCK	o	16	o	IE0
$\overline{\text{BA0}}$	o	17	o	A11
A14	o	18	o	A10
	o	19	o	
$\overline{\text{M1}}$	o	20	o	$\overline{\text{NMI}}$
	o	21	o	$\overline{\text{INT}}$
	o	22	o	$\overline{\text{WR}}$
	o	23	o	
	o	24	o	$\overline{\text{RD}}$
	o	25	o	$\overline{\text{HALT}}$
** $\overline{\text{DESEL}}$	o	26	o	$\overline{\text{PWRCLR}}$
$\overline{\text{IORQ}}$	o	27	o	A12
$\overline{\text{RFSH}}$	o	28	o	A15
A13	o	29	o	CLOCK
A9	o	30	o	$\overline{\text{MRQ}}$
$\overline{\text{BUSAk}}$	o	31	o	$\overline{\text{RESET}}$
GND	o	32	o	GND

nicht bezeichnete Pins des Steckers sind nicht belegt  
mit \*\* bezeichnete Pins weichen vom ECB-Bus ab



Signalgruppen, Fan - In / Fan - Out

Signal	von IC-Typ	Fan-Out	Fan-In
D0 ... D7	245	60	1
A0 ... A15	244	60	
$\overline{\text{RFSH}}$	244	60	
$\overline{\text{RD}}$	244	60	
$\overline{\text{WR}}$	244	60	
$\overline{\text{MRQ}}$	244	60	
$\overline{\text{IORQ}}$	244	60	
$\overline{\text{MI}}$	244	60	
CLOCK	S04 + T	60	
2 x CLOCK	321	60	
$\overline{\text{RESET}}$	132		1 + 10K Pull up
$\overline{\text{PWRCLR}}$	S04	60	
$\overline{\text{DESELECT}}$			1 + 2K2 Pull-up
$\overline{\text{NMI, INT, WAIT}}$			1 + 2K2 Pull-up
$\overline{\text{BUSRQ}}$			1 + 2K2 Pull-up
$\overline{\text{BUSAK}}$	S04	60	
IE1			1 + 2K2 Pull-up
IE0	21	20	
A16 .. A19	670	20	
+ 5V			ca. 1000 mA
+12V			ca. 50 mA (V24)

4.2 V24-Schnittstellen

Die beiden seriellen Schnittstellen S2 sind softwaremäßig einstellbar zwischen 50 und 9600 bzw. 19200 (bei 5 oder 6 MHz) Baud. Empfohlen wird folgende Einstellung des angeschlossenen Terminals:

- V24, keinesfalls Current-Loop
- Remote bzw. Online-Betrieb
- (19200), 9600, 4800, 2400 oder 1200 baud
- 8 bits / character
- 1 stopbit
- parity off

RTS / CTS - Handshake ist prinzipiell möglich  
DTR / DCD wird nicht benutzt, Pins sind nicht angeschlossen

Erzeugt werden + / - 8 Volt Signalpegel; beim Empfang werden -12 ... 0 V als Minus und +2 ... +12 V als Plus erkannt.

Die beiden seriellen Kanäle sind auf der Karte an einen gemeinsamen Pfostenstecker geführt. Die Pins 1 bis 13 gehören zum Kanal A, die Pins 14 bis 26 zum Kanal B. Beide Kanäle können direkt auf CANNON-Stecker gequetscht werden, hierbei wird das 26-polige Quetschkabel auf der Pfosten-seite gequetscht, dann auf der CANNON-Steckerseite zwischen den Leitungen 13 und 14 aufgetrennt und dann Pin 1 des Pfostens auf Pin 1 des CANNON-Steckers A und Pin 14 des Pfostens auf Pin 1 des CANNON-Steckers B gequetscht. Dann ergibt sich folgende Belegung der CANNON-Stecker (male oder female ist möglich):

S2			Cannon	
-	o 1	2o	-	1
RXDA	o 3	4o	-	14
TXDA	o 5	6o	-	Receive Data
CTSA	o 7	8o	-	2
RTSA	o 9	10o	-	15
-	o11	12o	-	3
GND	o13	14o	-	16
-	o15	16o	RXDB	4
-	o17	18o	TXDB	17
-	o19	20o	CTSB	Request to send
-	o21	22o	RTSB	5
-	o23	24o	-	18
-	o25	26o	GND	6
			Ground	19
				7
				20
				8
				21
				9
				22
				10
				23
				11
				24
				12
				25
				13

nicht bezeichnete Pins sind frei

Diese Belegung der V24-Schnittstelle stimmt mit fast allen Terminals überein, d.h. es kann ein Quetschkabel zwischen Rechner und Terminal verwendet werden.

## 4.3 Centronics-Schnittstelle

Die Centronics-Parallel-Schnittstelle S3 ist zum Anschluß von einem Drucker gedacht. Sie ist auf den 40-poligen Pfostenstecker C gelegt. Auch hier besteht die Möglichkeit Quetschkabel zu verwenden. Es ist ein 36-poliges Kabel so zu quetschen, daß Pin 1 des 40-poligen Pfostensteckers mit Pin 1 des Centronics-Steckers (36-polig) verbunden wird. Belegung des 36-pin-Steckers (z. B. AMPHENOL 57-30360):

Strobe not	0	1	19	G	Ground
Data 0	0	2	20	G	Ground
Data 1	0	3	21	G	Ground
Data 2	0	4	22	G	Ground
Data 3	0	5	23	G	Ground
Data 4	0	6	24	G	Ground
Data 5	0	7	25	G	Ground
Data 6	0	8	26	G	Ground
Data 7	0	9	27	G	Ground
Acknlg not	I	10	28	G	Ground
Busy	I	11	29	G	Ground
Paper end	I	12	30	G	Ground
Selected	I	13	31	0	Init not
not connected	0	14	32	I	Error not
not connected	-	15	33	G	Ground ***
Ground	G	16	34	-	not connected
not connected	-	17	35	-	not connected
not connected	-	18	36	0	not connected

## Steckerbelegung S3

Strobe not	0	1	2	G	Ground
Data 0	0	3	4	G	Ground
Data 1	0	5	6	G	Ground
Data 2	0	7	8	G	Ground
Data 3	0	9	10	G	Ground
Data 4	0	11	12	G	Ground
Data 5	0	13	14	G	Ground
Data 6	0	15	16	G	Ground
Data 7	0	17	18	G	Ground
Acknlg not	I	19	20	G	Ground
Busy	I	21	22	G	Ground
Paper end	I	23	24	G	Ground
Selected	I	25	26	0	Init not
not connected	0	27	28	I	Error not
not connected	-	29	30	G	Ground ***
Ground	G	31	32	-	not connected
not connected	-	33	34	-	not connected
not connected	-	35	36	0	not connected
.....					
not connected	-	37	38	x	ARDY
ASTB	x	39	40	x	BRDY

```

hierbei bedeutet:  0   Ausgang vom Rechner
                   1   Eingang des Rechners
                   -   nicht belegt
                   x   nicht benutzt
                   G   Masse (Ground)
                   ***  nicht Centronics

```

Alle Signale sind über die PIO geführt. Die Daten sind mit den ICs 32 und 33 gepuffert soweit nötig und möglich.

## 5. Programmbeispiele

### 5.1 SIO für seriell I/O

Die SIO-0 bzw. der DART wird als serielle V24-Schnittstelle verwendet. Folgende drei Routinen sind hierfür erforderlich:

(Es wird angenommen, daß der CTC - wie in Abschnitt 5.3 beschrieben - bereits programmiert ist!)

#### 1. Initialisieren für Polling-Betrieb:

```

;
;   Register C = Portadresse Command
;               = 0E1H für Kanal "A"
;               = 0E3H für Kanal "B"
;
INIT:  LD   HL, TABLE
       LD   A, 18H           ; Channel Reset
       OUT (C), A
       LD   B, 10           ; LENGTH of TABLE
       OTIR
       RET
TABLE:  DEFB 1, 0            ; WR-REG. 1, INT disabled
       DEFB 2, 0            ; WR-REG. 2, INT-VECTOR
       DEFB 3, 0C1H        ; WR-REG. 3, Rx 8 Bits
                               ; Rx enable
       DEFB 4, 044H        ; WR-REG. 4, X16-Clock
                               ; 1 Stop-Bit no Parity
       DEFB 5, 06AH        ; WR-REG. 5, DTR off,
                               ; Tx 8 Bits, Tx enable
                               ; RTS on

```

## 2. Read-Routine (ein Zeichen lesen)

```

;
;       Register C = Portadresse Command
;                   = 0E1H für Kanal "A"
;                   = 0E3H für Kanal "B"
;
;
READ:   IN   (A),C           ; Status lesen
        AND  01H           ; RxRdy ?
        JR   Z,READ        ; nein: warten
        DEC  C             ; Data-Port
        IN   A,(C)         ; Zeichen lesen
        INC  C
        RET
;
;       Register A = gelesenes Zeichen
;

```

## 3. Write-Routine (ein Zeichen schreiben)

```

;
;       Register C = Portadresse Command
;                   = 0E1H für Kanal "A"
;                   = 0E3H für Kanal "B"
;
;
;       Register A = zu schreibendes Zeichen
;
;
WRITE:  PUSH AF
WRL:    IN   A,(C)         status lesen
        AND  04H           ; Tx empty
        JR   Z,WRL        ; nein: warten
        DEC  C             ; Datenport
        POP  AF           ; Zeichen
        OUT  (C),A        ; senden
        INC  C
        RET

```

## 4. Testprogramm für Kanal "A" der SIO

```

;
; dieses Programm initialisiert die SIO (Kanal "A") und
; liest dann Zeichen und sendet sie als Echo auf das
; Terminal zurück.
;
TEST:   LD   C,0E1H       ; SIO A Command
        CALL INIT        ; initialisieren
TLOP:   CALL READ        ; Zeichen lesen
        CALL WRITE       ; und als Echo senden
        JR   TLOP        ; und von vorne

```

5.2 PIO als Centronics-Schnittstelle

Die folgenden Routinen beschreiben die Programmierung der PIO als Centronics-Schnittstelle im Polling-Betrieb. Zuerst ist die PIO zu initialisieren, dann folgt eine Routine zum Senden eines Zeichen an den Drucker.

1. Portadressen für die folgenden Routinen:

```

PIOA      EQU      0E4H          ; "A"-DATA      STEUERPORT
PIOA+1    EQU      0E5H          ; "A"-CONTROL
PIOB      EQU      0E6H          ; "B"-DATA      DATENPORT
PIOB+1    EQU      0E7H          ; "B"-CONTROL

```

2. Belegung der PIO

```

PIO Port A      Bit 0 = Selected (Input)
(Steuer-Port)  Bit 1 = Busy      (Input)
               Bit 2 = Paper end (Input)
               Bit 3 = Error not (Input)
               Bit 4 = Strobe not (Output)
               Bit 5 = Select not (Output)
               Bit 6 = Autolf not (Output)
               Bit 7 = Init not  (Output)
PIO Port B      Bit 0 bis 7 = Daten (Output)

```

3. Initialisierungsroutine

```

PINIT:  LD      A,0FH          ; OUTPUT MODE 0
        OUT     (PIOB+1),A    ; PORT B
        LD      A,0CFH        ; CONTROL MODE 3
        OUT     (PIOA+1),A    ; PORT A
        LD      A,0FH          ; BIT 7 .. 4 = OUT
                                   ; BIT 0 .. 3 = IN
        OUT     (PIOA+1),A
        XOR     A
        OUT     (PIOA),A      ; CLEAR CONTROL-PORT
        RET

```

4. Senderoutine (ein Zeichen senden)

Register A = zu sendendes Zeichen

```

PIOOUT: PUSH   AF
PIOI:   IN     A,(PIOA)       ; Status lesen
        AND    0EH           ; ready
        CP     0EH
        JR     NZ,PIOI        ; nein: warten
        POP    AF            ; Zeichen
        OUT    (PIOB),A      ; senden
        LD    A,10H          ; STROBE ON
        OUT    (PIOA),A
        XOR    A              ; STROBE OFF
        OUT    (PIOA),A
        RET

```

5.3 CTC als Baudratengenerator

Die folgende Routine initialisiert einen Kanal des CTC als Baudratengenerator für die SIO (Prescaler in der SIO = 16). Für 4800 Baud ist also HL mit TABLE.+2 zu laden. In Register C wird 0E8H als Kanaladresse für SIO Port "A" eingetragen.

```

;
; Register HL points to Baudrate-Factor
; Register C = Portadresse CTC
;           = 0E8H für Kanal "A" der SIO
;           = 0EAH für Kanal "B" der SIO
CINIT: LD B,2
      OTIR
      RET
;
; Baudraten-Tabelle (4 MHz Systemtakt)
;
TABLE4: DEFB      4DH,13    ; 9600 BAUD (Counter 2MHz
      DEFB      4DH,26    ; 4800 BAUD  Clock)
      DEFB      4DH,52    ; 2400 BAUD
      DEFB      4DH,104   ; 1200 BAUD
      DEFB      4DH,208   ; 600 BAUD
      DEFB      0DH,52    ; 300 BAUD (Timer 4 MHz
      DEFB      0DH,104   ; 150 BAUD  Systemtakt)
      DEFB      0DH,142   ; 110 BAUD
      DEFB      0DH,208   ; 75 BAUD
;
; Baudraten-Tabelle (5 MHz Systemtakt)
;
TABLE5: DEFB      4DH,8    ; 19K2 BAUD (Counter 2,5
      DEFB      4DH,16    ; 9600 BAUD  MHz Clock)
      DEFB      4DH,32    ; 4800 BAUD
      DEFB      4DH,64    ; 2400 BAUD
      DEFB      4DH,128   ; 1200 BAUD
      DEFB      4DH,0     ; 600 BAUD
      DEFB      0DH,64    ; 300 BAUD (Timer 5 MHz
      DEFB      0DH,128   ; 150 BAUD  Systemtakt)
      DEFB      0DH,174   ; 110 BAUD
      DEFB      0DH,0     ; 75 BAUD
;
; Baudraten-Tabelle (6 MHz Systemtakt)
;
TABLE6: DEFB      4DH,10   ; 19K2 BAUD (Counter 3MHz
      DEFB      4DH,20   ; 9600 BAUD  Clock)
      DEFB      4DH,40   ; 4800 BAUD
      DEFB      4DH,80   ; 2400 BAUD
      DEFB      4DH,160  ; 1200 BAUD
      DEFB      0DH,39   ; 600 BAUD (Timer 6 MHz
      DEFB      0DH,78   ; 300 BAUD  Systemtakt)
      DEFB      0DH,156  ; 150 BAUD
      DEFB      0DH,213  ; 110 BAUD

```

5.4 Anwendung Adress-Erweiterung (Banking)

Das Beispiel blendet auf Adresse 4000H den 3. 16K-Block aus Page 1 (18000H..1BFFFFH) ein und auf Adresse 8000H den 1. 16K-Block aus Page 3 (30000H..33FFFFH) ein. Voraussetzung hierfür ist eine bankingfähige Memory-Erweiterung.

```

ADR1      EQU      0E0H      ; BANKADRESSE 0000H-3FFFFH
ADR2      EQU      0EDH      ; BANKADRESSE 4000H-7FFFFH
ADR3      EQU      0EEH      ; BANKADRESSE 8000H-BFFFFH
ADR4      EQU      0EFH      ; BANKADRESSE C000H-FFFFH
START:    LD       A,18H     ; D0 = DON'T CARE    x
                                     ; D1 = DON'T CARE    x
                                     ; D2 = A14'         0
                                     ; D3 = A15'         1
                                     ; D4 = A16          1
                                     ; D5 = A17          0
                                     ; D6 = A18          0
                                     ; D7 = A19          0
                                     ; SETZEN DES ENTSPRECHEN-
OUT       (0EDH),A          ; DEN REGISTERS
LD       A,30H             ; D0 = DON'T CARE    x
                                     ; D1 = DON'T CARE    x
                                     ; D2 = A14'         0
                                     ; D3 = A15'         0
                                     ; D4 = A16          1
                                     ; D5 = A17          1
                                     ; D6 = A18          0
                                     ; D7 = A19          0
OUT       (0EEH),A          ; SETZEN DES ENTSPRECHEN-
                                     ; DEN REGISTERS

```

somit ergibt sich die folgende Speicheraufteilung:

I/O-Adresse	CPU (logisch)	RAM (physikalisch)
0E0 = 00H	0000H..3FFFFH	00000H..03FFFFH
0ED = 18H	4000H..7FFFFH	18000H..1BFFFFH
0EE = 30H	8000H..BFFFFH	30000H..33FFFFH
0EF = 0CH	C000H..FFFFH	0C000H..0FFFFFH




6.1 Stückliste

## aktive Bauelemente:

IC 1	SN 74 LS 367
IC 2	SN 74 LS 245
IC 3	SN 74 LS 32
IC 4	BYTEWIDE-SOCKEL FÜR EPROM
IC 5, 13, 18, 33	SN 74 LS 244
IC 6	SN 74 LS 74
IC 7	TBP 24 SA 10 prog. (BUSSTEUER-ROM)
IC 8, 14	SN 74 LS 670 (REGISTER-FILE)
IC 9	SN 74 LS 132
IC 10	SN 74 LS 21
IC 11	NEC D780C-1(2) bzw. Z80A(B)-CPU
IC 12	TBP 24 SA 10 prog. (MEMORY-ROM)
IC 15	SN 74 S 04
IC 16, 17	SN 74 S 158 oder S 157
IC 19 - IC 26	64K DYN. MEMORY 150ns 128 REFRESHZYKLEN
IC 27	SN 74 LS 321 (OSZILLATOR)
IC 28	TBP 24 SA 10 prog. (I/O-ROM)
IC 29	Z80A(B)-DART/SIO 0
IC 30	ICL 7660 CPA (DC-DC WANDLER)
IC 31	Z80A(B)-CTC
IC 32	SN 74 LS 240
IC 34	SN 75 189 (V24-EMPFÄNGER)
IC 35	SN 75 188 (V24-SENDER)
IC 36	Z80A(B)-PIO
T	TRANSISTOR 2N 2907 (POWER-CLOCK)
Q	Quarz 8, 9,8304 oder 12 MHz
L	Spule 47..100uH
D 1	1 N 4148 oder ähnlich
Z 1	LED grün (bei Dioden ist das Symbol = Kathode)
X	Diagnose-LED (rot) X1= DMA-Zugriff X2= Internal Memory Zugriff

## Kondensatoren:

	3x	10 uF 16V Tantal (unterh. IC 30, + = <math>\langle \rangle</math>)
C	1x	10 uF Tantal (neben R24, + zu R24)
C	1x	10 uF Tantal (neben R22, + zu IC29)
C1	1x	33 pF keramisch
C	16x	100 nF Vielschichtkondensatoren
C	1x	47 uF (zwischen R6 und R7)

## Widerstände:

R 1	470R
R 2, 3, 4	1K5
R 5	82K
R 6	4K7
R 7	10K
R 8 .. 10	1K5
R 11 .. 16	680R
R 17, 18	330R
R 19 .. 22	680R
R 23 .. 29	1K5
SIL	8*1K2, gemeinsamer Anschluß = PIN 1

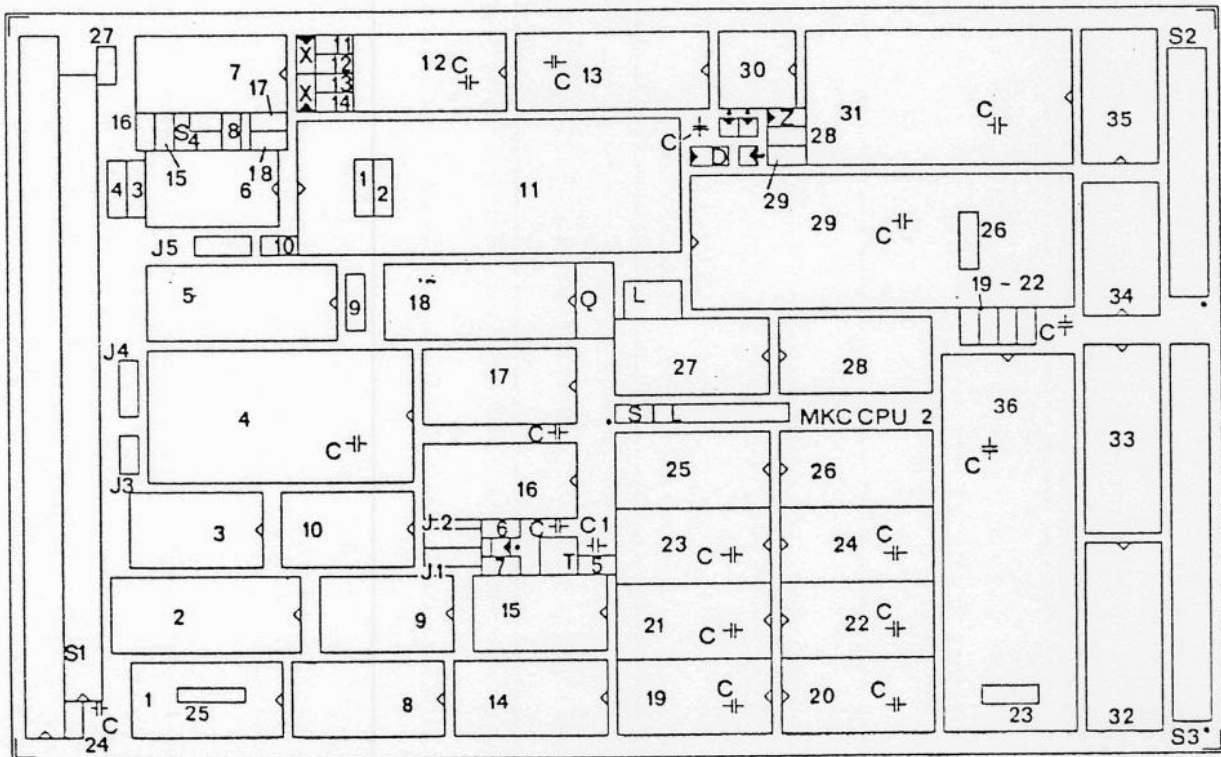
## Stecker:

S1	UG 64-Leiste a und c bestückt
S2	26 pol. Pfofen (2 x 13)
S3	40 pol. Pfofen (2 x 20)
S4	2 pol. Pfofen für Boot-LED
J1, J2, J4, J5	3 pol. Pfofen (1 x 3)
J3	2 pol. Pfofen (1 x 2)

## Fassungen:

1	DIL 8
7	DIL 14 offen
17	DIL 16 offen
6	DIL 20 offen
2	DIL 28 offen
3	DIL 40 offen

## 6.2 Bestückungsplan



### 6.3 Inhalt der Steuer-Proms

#### 1. I/O Steuerprom IC 28

Sig	Eingang								Ausgang			
	A7	A6	A5	A4	A3	A2	A1	M1	CE4	CTC	SIO	P10
Pin	15	1	2	3	4	7	6	5	9	10	11	12
Nam	H	G	F	E	D	C	B	A	04	03	02	01
SIO	1	1	1	0	0	0	0	1	1	1	0	1
SIO	1	1	1	0	0	0	1	1	1	1	0	1
P10	1	1	1	0	0	1	0	1	1	1	1	0
P10	1	1	1	0	0	1	1	1	1	1	1	0
CTC	1	1	1	0	1	0	0	1	1	0	1	1
CTC	1	1	1	0	1	0	1	1	1	0	1	1
Bank	1	1	1	0	1	1	0	1	0	1	1	1
Bank	1	1	1	0	1	1	1	1	0	1	1	1

Daraus ergibt sich: Adresse E1H Inhalt 0DH

E3H	0DH
E5H	0EH
E7H	0EH
E9H	0BH
EAH	0BH
EDH	07H
EFH	07H

#### 2. Memory Steuer-Rom IC 12

Sig	A17	A18/19 Boot	Deselect A16	A15	A14	A13	M1	IMD		Eprom			
								Bootr	Ram	01	00		
Pin	14	15	1	2	3	4	7	6	5	9	10	11	12
Nam	CS	H	G	F	E	D	C	B	A	03	02	01	00
	0	0	0	1	0	0	0	0	1	0	1	0	1
	0	0	0	1	0	0	0	0	0	0	1	0	1
	0	0	0	1	0	0	0	1	1	0	1	1	0
	0	0	0	1	0	0	1	0	1	0	1	1	0
	:	:	:	:	:	:	:	:	:	:	:	:	:
	0	0	0	1	0	1	1	1	1	0	1	1	0
	0	0	0	1	0	0	0	1	0	0	0	1	0
	0	0	0	1	0	0	1	0	0	0	0	1	0
	:	:	:	:	:	:	:	:	:	:	:	:	:
	0	0	0	1	0	1	1	1	0	0	0	1	0
	0	0	1	1	0	0	0	0	0	0	0	1	0
	:	:	:	:	:	:	:	:	:	:	:	:	:
	0	0	1	1	0	1	1	1	1	0	0	1	0

Sig = Signal am Prom Pin = Pin-Nummer Nam = Pin-Name

Prominhalt IC 12: Adresse	Daten
00	F F F F F F F F F F F F F F F F
10	F F F F F F F F F F F F F F F F
20	5 5 2 6 2 6 2 6 2 6 2 6 2 6 2 6
30	F F F F F F F F F F F F F F F F
40	F F F F F F F F F F F F F F F F
60	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
70	F F F F F F F F F F F F F F F F
:	
F0	F F F F F F F F F F F F F F F F

Wie sich aus der Aufstellung ergibt, läßt sich der Eprom-Bereich auch fest, unabhängig vom Boot-Signal, einblenden.  
Einschränkung: Einblendung in 8K Schritten und 8K Größe.

### 3. Bussteuer-Rom IC 7

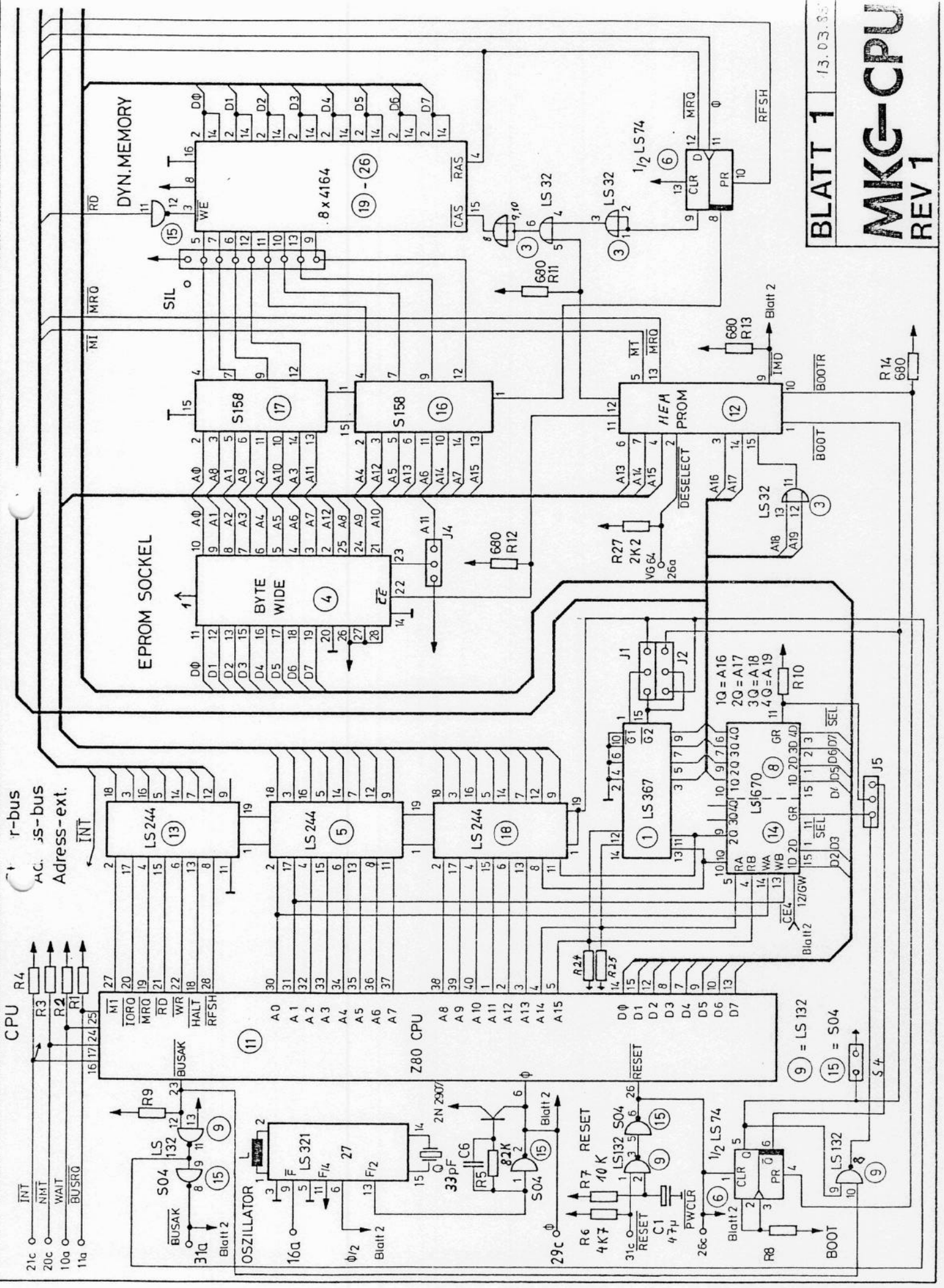
Adresse	Daten
10	F F F F F F F F F F F F F F F F
20	F F F F F F F F F F F F F F F F
30	F F F F F F F F F F F F F F F F
40	F F F F F F F F 2 2 2 2 3 3 3 3
50	F F F F F F F F 6 6 6 6 3 3 3 3
60	F F F F F F F F F F F F F F F F
70	F F F F F F F F F F F F F F F F
80	F F F F F F F F F F F F F F F F
90	F F F F F F F F F F F F 7 7 0 7
A0	F F F F 1 1 1 1 F F F F 7 7 7 7
B0	F F F F 1 1 1 1 F F F F 3 3 3 3
C0	F F F F F F F F F F F F 7 7 7 7
D0	F F F F F F F F F F F F 3 3 3 3
E0	2 2 2 2 1 1 1 1 F F F F 7 7 7 7
F0	6 6 6 6 1 1 1 1 F F F F 3 3 3 3

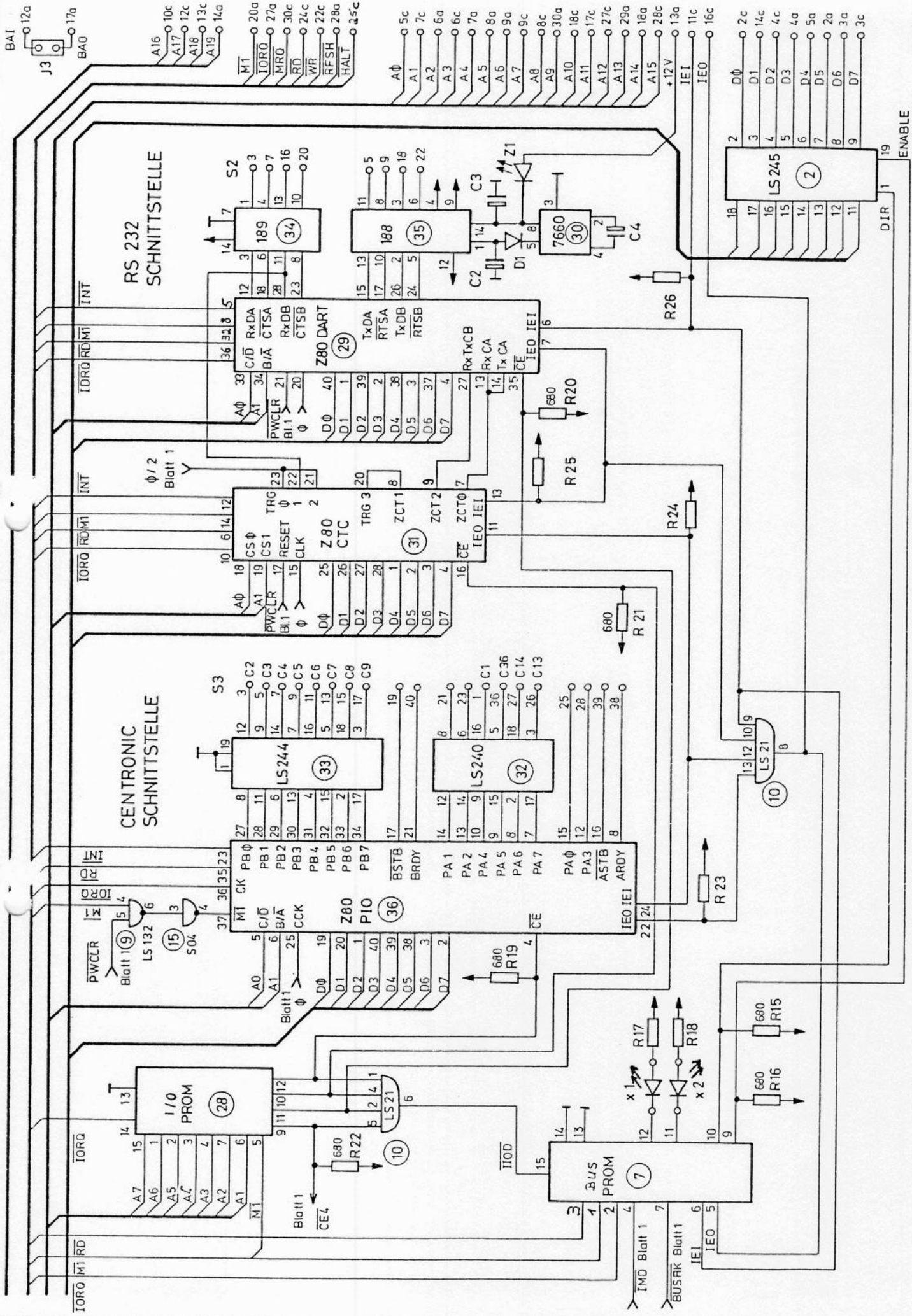
#### 6.4 Jumperfunktionen

Auf der CPU II Karte befinden sich drei Jumper, die es ermöglichen, die Banking-Verwaltung verschieden zu nutzen.

Sämtliche Angaben für das Stecken der Jumper gilt für die Draufsicht auf die Bestückungsseite mit der VG-Leiste nach rechts.

- J1** Nach links gesteckt (nur in Verbindung mit J2 nach rechts) werden die Adressen A16..A19 während des Boot-Betriebes auf Ground gezogen.  
Nach rechts gesteckt werden die Adressen A16..A19 während eines DMA-Zugriffs disabled. Sonst werden sie auf Ground gezogen. (J2 beliebig)  
Standard: nach links gesteckt.
- J2** Nach rechts gesteckt werden die Adressen A14, A15 während des Boot-Betriebes auf den Bus gelegt.  
Nach links gesteckt werden die Adressen immer wenn kein DMA-Zugriff erfolgt auf den Bus gelegt.  
Achtung dabei muß IC 14 entfernt werden!  
Standard: nach rechts gesteckt.
- J5** Nach rechts gesteckt wird das Register-File für A16 .. A19 nur während des Boot-Betriebes disabled. (in Verbindung mit der FDC 8/5 erforderlich)  
Nach links gesteckt wird das Register-File für A16 .. A19 während des Boot-Betriebes und während eines DMA-Zugriffs disabled. (Die DMA-Karte darf nur bei DMA A16..A19 auf den Bus legen.)  
Standard: nach rechts gesteckt.
- J3** Geschlossen wird BAI mit BAO verbunden.
- J4** Nach oben gesteckt wird Pin 23 des Bytewise-Sockels IC 4 mit +5V verbunden. (Eprom 2716)  
Nach unten gesteckt wird Pin 23 mit A11 verbunden. (Eprom 2732, 2764)
- S4** S4 ist zum Anschluß einer BOOT-LED gedacht, es ist eine LED mit eingebautem Vorwiderstand (ca. 220 Ohm) vorgesehen. + 5 Volt liegen an dem weiter von der VG-Leiste entfernten Pin des Steckers.





	MMMM	MMMM	KKKK	KKKK	CCCCCCCCCC	
	MMMMMMMMMMMMMMMM		KKKK	KKKK	CCCCCCCCCCCCCC	
	MMMM	MMMMMM	MMMM	KKKK	KKKK	CCCC
MMMM	MMMM	MMMM	KKKK	KKKK	CCCC	
MMMM	MMMM	MMMM	KKKKKKKKKKKK		CCCC	
MMMM	MMMM	MMMM	KKKKKKKKKK		CCCC	
MMMM	MMMM	MMMM	KKKKKKKKKKKK		CCCC	
MMMM	MMMM	MMMM	KKKK	KKKK	CCCC	
MMMM	MMMM	MMMM	KKKK	KKKK	CCCC	
MMMM	MMMM	MMMM	KKKK	KKKK	CCCC	
MMMM	MMMM	MMMM	KKKK	KKKK	CCCC	
MMMM	MMMM	MMMM	KKKK	KKKK	CCCCCCCCCC	
MMMM	MMMM	MMMM	KKKK	KKKK	CCCCCCCCCC	

Michels & Kleberhoff Computer GmbH  
Hauptstrasse 78, 5600 Wuppertal 12

technisches Handbuch

M. K. C. FDC II

Lieferversionen:

MKC-Z-3042	FDC II fertig 4 MHz
MKC-Z-3043	FDC II fertig 6 MHz
MKC-Z-3040	FDC II Leerplatine
MKC-Z-3048	Sonderbauteile-Satz inkl. Proms (fuer 3040)
MKC-Z-3049	Promsatz (nur für MKC-Z-3040)



## Inhaltsverzeichnis

1.	Übersicht	3
2.	allgemeine Beschreibung	4
3.	technische Daten	5
	- Businterface	5
	- Floppyteil	7
	- Winchester-(SASI)-teil	9
4.	I/O-Adressen	10
5.	Prominhalte	11
	- Bussteuerprom	11
	- Memoryprom	11
	- I/O-Dekoderprom	11
6.	Bestückung	12
	- Stückliste	12
	- Bestückungsplan	13
	- Aufbauhinweise	13
7.	Jumper-Einstellungen	14
8.	Schaltplan	15
9.	Programmbeispiele	18

## 1. Übersicht

Die M.K.C. FDC II Platine ist ein universelles Interface zwischen einem ECB-280-Computer und Massenspeichern. Anschließbar sind bis zu 7 Floppy-Disk-Laufwerke für 5,25" und 8" Disketten und über die SASI (SCSI)-Schnittstelle bis zu 2 Winchester-Laufwerke an einem XEBEC S1410(A)-Controller.

Ferner verfügt die FDC II Karte über 64 K-Byte Speicher und eine akkugepufferte Uhr.

Somit kann mit nur 2 Platinen (der FDC II und der CPU II) ein vollständiges banked CP/M Plus System aufgebaut werden. Dieses System enthält 128 K Speicher, Anschlussmöglichkeiten für die üblichen Massenspeicher, zwei V24- und eine CENTRONICS-Schnittstelle. Es ist ausbaufähig bis zu 1 MByte Speicherkapazität.

Die M.K.C. FDC II Karte enthält den NEC uPD 765 als Floppy-Controller. Als Floppy-Support Chip wird der SMC FDC 9229 eingesetzt. Der FDC 9229 enthält einen monolithischen Datenseparator, die Takterzeugung und die Write-Precompensations-Logik. 8 Precompensationszeiten sind softwaremäßig einstellbar. Der Floppyteil läuft im Vector-Interrupt-Mode. Die M.K.C. FDC II Karte muß an höchster Priorität stecken. Niedrigere Interrupts werden bei Datentransfers über die IEI/IEO-Daisy-Chain gesperrt. Auch 8" MFM (Double Density) ist in einem System mit 4 MHz Systemtakt problemlos realisierbar! Ein Datentransfer darf nicht durch Interrupts oder DMA-Zugriffe gestört werden. Ein Transfer in RAM-Karten, die WAIT erzeugen, ist problematisch. Die synchrone Taktumschaltung für 5,25" und 8" ermöglicht gemischten Betrieb von 5,25" und 8" Laufwerken.

Das SASI-Interface ist interruptfähig (über die PIO). Der Datentransfer ist zeitunkritisch, d.h. hier können auch Interrupts bzw. DMA-Zugriffe oder WAITs während eines Datentransfers zugelassen werden.

Als Uhr wird das MEM E050-16 Uhrenchip verwendet. Mit dem Trimmer T ist ein genauer Abgleich der Uhr möglich. Der Akku puffert die Uhr für mindestens 500 Stunden. Somit muß die Uhr nicht nach jedem Einschalten neu gestellt werden.

Der RAMteil enthält 64K \* 1 dynamische RAMs, der REFRESH erfolgt durch die 280 CPU. Dekodiert werden 20 bit Adressbus. Die Seite (zu 64KByte) ist einstellbar.

## 2. allgemeine Beschreibung

### - Floppyteil

Vorgesehen ist der Anschluß von bis zu drei 5,25" und vier 8" Laufwerken. Als Aufzeichnungsverfahren wird MFM oder FM (double oder single density) mit IBM-kompatiblen Formaten verwendet. (uPD765 als Floppy-Controller).

Der Anschluß von ein- oder zweiseitigen Laufwerken mit Shugart-kompatiblem Bus ist möglich. (Wir empfehlen TEAC FD55A bis F bzw. NEC1165 Laufwerke!)

Der Floppy-Controller wird im Interrupt (IM2 über CTC bzw. PIO) betrieben. Damit ist bei einem Systemtakt von mindestens 4 MHz auch 8" MFM problemlos realisierbar.

Andere Interrupts können (müssen) während des Datentransfers blockiert werden.

Der CTC wird weiterhin zur Motorsteuerung der Laufwerke (falls möglich) verwendet. Der Motor wird ca. 20 sec nach dem letzten Zugriff auf das Laufwerk abgestellt. Durch Auslesen der noch verbleibenden Zeit aus dem CTC könnten auch Laufwerke ohne READY-Signal verwendet werden. Dies wird allerdings von unserer Software nicht unterstützt. Über die PIO kann die Write-Precompensation-Time per Software eingestellt werden.

### - Winchesterteil

Zum Anschluß von Winchester-Laufwerken ist auf der FDC II ein SASI (SCSI)-Interface vorhanden. Für den SASI-Bus gibt es von mehreren Herstellern Controller, an die i.a. bis zu 2 Winchester-Laufwerke angeschlossen werden können. Die hier vorhandene SASI-Schnittstelle ist nur für den Anschluß von Winchester-Controllern vorgesehen, an sie kann keinesfalls ein zweiter MASTER (Rechner) angeschlossen werden. Die im weiteren angegebenen Routinen sind für den XEBEC S1410(A) Controller geschrieben. Sie müssen für andere Controller adaptiert werden. Unser CP/M PLUS unterstützt nur den XEBEC-Winchester-Controller!

Die SASI-Schnittstelle ist voll interrupt-fähig (IM2 über die PIO); dies wird von unserer Software jedoch nicht benötigt.

Benutzt wird die OPEN-COLLECTOR-Version der SASI-Norm.

Die optionelle PARITY-Prüfung ist nicht implementiert.

### - RAMteil

Die M.K.C. FDC II Platine enthält 64 K-Byte dynamisches RAM. Dekodiert werden 20 Bit Adressen. Die obersten 4 Bit (Speicherseite) sind über Jumper einstellbar. (Normal-einstellung: 10000H..1FFFFH) Ein DESELECT-Eingang ist nicht vorhanden. Der REFRESH wird von der Z80-CPU gesteuert.

### - die Uhr

Als Uhr wird das MEM-IC E050-16 verwendet. Die Uhr ist akkugepuffert. Setzen und Lesen erfolgt über die PIO.

### 3. technische Daten

#### 3.0 Businterface

Das Businterface der FDC II Karte ist kompatibel zum ECB-Bus. Abweichend vom ECB-Bus sind nur die Adressen A16 bis A20. Sie belegen die Pins 10c, 12c, 13c und 14a der VG-Leiste.

#### Achtung!

Bei Betrieb von Mini-Floppy-Laufwerken kann der uPD 765 nicht ohne Wait mit einer höheren Systemtakt rate als 4 MHz betrieben werden. Bei 6 MHz-Systemen ist der Wait-Jumper unbedingt zu setzen, wenn mit 5,25"-Laufwerken gearbeitet werden soll. (Sonst erfolgt die Fehlermeldung OVERRUN)

#### Versorgung

Die FDC-II-Karte benötigt voll bestückt bei 5 Volt ca. 1A bei 4 MHz Systemtakt. Andere Spannungen werden nicht benötigt.

#### fan-in/fan-out

Signal	LS-fan-in	LS-fan-out
D0 bis D7	1	60
A0 bis A1	8	
A2 bis A7	7	
A8 bis A15	6	
A16 bis A19	5	
IEI	2	
IEO		20
INT		4
WAIT		25
CLK	4	
MRQ, IORQ, M1, RD	3	
WR, PWRCLR	1	

## Steckerbelegung VG 64 (ECB-kompatibel) S1

	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
<u>WAIT</u>	o	10	o	A16
<u>BUSRQ</u>	o	11	o	IEI
	o	12	o	A17
	o	13	o	A18
A19	o	14	o	D1
	o	15	o	
	o	16	o	IE0
	o	17	o	A11
A14	o	18	o	A10
	o	19	o	
<u>M1</u>	o	20	o	
	o	21	o	<u>INT</u>
	o	22	o	<u>WR</u>
	o	23	o	
	o	24	o	<u>RD</u>
	o	25	o	
	o	26	o	<u>PWRCLR</u>
<u>IORQ</u>	o	27	o	A12
	o	28	o	A15
A13	o	29	o	<u>CLOCK</u>
<u>A9</u>	o	30	o	<u>MRQ</u>
<u>BUSAK</u>	o	31	o	
GND	o	32	o	GND
	a		c	

nicht bezeichnete Pins des Steckers sind nicht belegt

### 3.1 Floppyteil

Das Floppyinterface basiert auf dem Floppy-Controller uPD765 und dem FDC-Support-Chip SMC FDC 9229B. Als Ausgangstreiber zu den Laufwerken werden OPEN-COLLECTOR ICs 7406, 7407 und 7445 verwendet. Im Eingang sind LS-TTL-Bausteine mit 220/330 Ohm Pull-up/Pull-Down angeordnet. Zwei kaskadierte Kanäle des CTC werden zur Motorsteuerung benutzt. Dies ermöglicht es, Laufwerke ohne READY-Signal zu verwenden.

Der uPD765 wird im DMA-Mode betrieben. Die für einen höheren Systemtakt als 4 MHz benötigte Wait-Logik kann mittels Jumper aktiviert werden. Die DRQ-Leitung (Anforderung) wird über die PIO geführt. DACK (Quittung) erfolgt über das I/O-Dekoderprom. Damit kann auch bei nur 4 MHz Systemtakt 8" MFM gelesen und geschrieben werden. Am Ende der Transfer-Phase erzeugt der uPD765 über den CTC einen Interrupt; hierdurch wird die Ein- oder Ausgabeschleife verlassen. Über einen weiteren Kanal des CTC kann die IEI/IEO-Daisy-Chain gesetzt werden. Dies verhindert, daß andere Peripheriebausteine während der kritischen Transfer-Phase eine Unterbrechung veranlassen können. Aus diesem Grund sollte der Floppycontroller auf dem Steckplatz mit der höchsten Priorität stecken. Es ist sicherzustellen, daß evtl. im System vorhandene DMAs während eines Datentransfers inaktiv bleiben.

Die Eingänge P0 bis P2 und MINI des FDC 9229 werden über die PIO gesteuert. Dies ermöglicht das Umschalten zwischen 5,25" und 8" und die Einstellung der Write-Precompensation-Time.

#### Write-Precompensation-Zeiten

P2	P1	P0	PIO Port B	5"	8"	
B1	B2	B3				
0	0	0	..m.000.	0	0	ns
0	0	1	..m.100.	125	62.5	ns
0	1	0	..m.010.	250	125	ns
0	1	1	..m.110.	375	187.5	ns
1	0	0	..m.001.	500	250	ns
1	0	1	..m.101.	500	250	ns
1	1	0	..m.011.	625	312.5	ns
1	1	1	..m.111.	625	312.5	ns

Die Punkte im Port B der PIO stehen für Bits, die für andere Funktionen benutzt werden, sie dürfen keinesfalls geändert werden. m ist das MINI/MAXI-Bit; eine 1 bedeutet MINI, eine 0 MAXI.

Belegung der Floppy-Steckers

pin	8" SHUGART S3	5,25" SHUGART S2	pin
2	LOW CURRENT	HEAD LOAD	2
4			4
6		READY ++++++	6
8		INDEX	8
10	TWO SIDED	SELECT 1	10
12		SELECT 2	12
14	SIDE SELECT	SELECT 3	14
16		MOTOR ON	16
18	HEAD LOAD	DIRECTION	18
20	INDEX	STEP	20
22	READY	WRITE DATA	22
24		WRITE GATE	24
26	SELECT 1	TRACK 00	26
28	SELECT 2	WRITE PROTECT	28
30	SELECT 3	READ DATA	30
32	SELECT 4	SIDE SELECT	32
34	DIRECTION	READY ++++++	34
36	STEP	-	36
38	WRITE DATA	-	38
40	WRITE GATE	-	40
42	TRACK 00	-	42
44	WRITE PROTECT	-	44
46	READ DATA	-	46
48		-	48
50		-	50

alle Signale von und zur Floppy sind aktiv LOW

nicht bezeichnete Pins sind nicht belegt. Bei 5,25" Laufwerken gibt es zwei Möglichkeiten für das READY-Signal: pin 34 z.B. bei TEAC oder pin 6 z.B. bei BASF. Dies wird mit den Jumpfern J4 bzw. J5 eingestellt. Somit ist für TEAC-Laufwerke (READY = pin 34) J5 und für BASF-Laufwerke (READY = pin 6) J4 zu setzen.

Alle ungeraden Pins liegen auf Masse. Bei Verwendung eines gemeinsamen Netzteiles für die Laufwerke und den Rechner entsteht hierdurch eine Brummschleife. Dies kann zu Störungen auf den inneren Spuren der Diskette führen. Notfalls muß Ground von den ungeraden Pins abgetrennt werden. Weiterhin ist der maximal zulässige Störpegel der Versorgungsspannungen für die Laufwerke zu beachten (meist 50 millivolt Spitze-Spitze).

### 3.2 Winchester-(SASI)-teil

Das SASI-Interface ist in diskreter Logik aufgebaut. Es benutzt das Port A der PIO für die Control-Signale und TTL-Treiber/register für den Datenbus. Als Ausgangstreiber werden 7406 und 74LS642 benutzt. Im Eingang sind LS-TTL oder die PIO mit 220/330 Ohm Pull-up/Pull-down angeordnet. Das SASI-Interface ist voll interruptfähig, es wird jedoch von der Beispiels- und System-Software im Polling betrieben.

Da der Winchestercontroller XEBEC 1410 einen Sektorpuffer besitzt, ist die Datenübertragung nicht zeitkritisch. Es dürfen also fremde Interrupts auch beim Lesen oder Schreiben eines Blocks zugelassen sein!

Das SASI-Interface ist die OPEN-Collector-Version ohne Parity-Prüfung. ARBITRATION und RESELECTION sind nicht implementiert. Es darf also nur einen INITIATOR (Bus-Master) geben.

#### Steckerbelegung S4

pin	Signal	
2	DB(0)	DATA 0 NOT
4	DB(1)	DATA 1 NOT
6	DB(2)	DATA 2 NOT
8	DB(3)	DATA 3 NOT
10	DB(4)	DATA 4 NOT
12	DB(5)	DATA 5 NOT
14	DB(6)	DATA 6 NOT
16	DB(7)	DATA 7 NOT
18	DB(P)	DATA PARITY NOT (nicht implementiert)
20		
22		
24		
26		
28		
30		
32	ATN	ATTENTION (nicht implementiert)
34		
36	BSY	BUSY NOT
38	ACK	ACKNOWLEDGE NOT
40	RST	RESET
42	MSG	MESSAGE NOT
44	SEL	SELECT NOT
46	C/D	CONTROL NOT/DATA
48	REQ	REQUEST NOT
50	I/O	INPUT NOT/OUTPUT

Alle ungeraden pins liegen auf Masse-Potential, nicht bezeichnete pins sind nicht belegt.



4. I/O-Adressen

Die M.K.C. FDC II belegt insgesamt 14 I/O-Adressen von 0F0h bis 0FDh. (Die Adresseinstellung erfolgt im I/O-PROM und ist prinzipiell auch für andere Adressen möglich; es erfolgt hierfür aber keine Softwareunterstützung durch uns.)

0F0	uPD765	Statusregister
0F1	uPD765	Datenregister
0F2		sperrt über den CTC die IEI/IEO-Kette
0F3	uPD765	Data-Acknowledge
0F4	Z80-CTC	Kanal 0 (uPD765 Interrupt-Controller)
0F5	Z80-CTC	Kanal 1 (Motorlaufzeit-Vorteiler)
0F6	Z80-CTC	Kanal 2 (Motorlaufzeit-Teiler)
0F7	Z80-CTC	Kanal 3 (sperrt die IEI/IEO-Kette)
0F8	Z80-PIO	Port A Datenregister (SASI und Uhr)
0F9	Z80-PIO	Port A Kommandoregister
0FA	Z80-PIO	Port B Datenregister (Floppy und Uhr)
0FB	Z80-PIO	Port B Kommandoregister
0FC	SASI	Daten lesen
0FD	SASI	Daten schreiben

**Belegung der PIO-Ports:**

A0	INPUT	SASI-REQUEST
A1	INPUT	SASI-BUSY
A2	INPUT	SASI-IO
A3	INPUT	SASI-CD
A4	INPUT	SASI-MSG
A5	OUTPUT	SASI-SEL
A6	OUTPUT	SASI-RESET
A7	OUTPUT	MEM E050 Chip-Select
B0	INPUT	uPD765 DATA-REQUEST
B1	OUTPUT	FDC9229B P2 Precompensation
B2	OUTPUT	FDC9229B P1 Precompensation
B3	OUTPUT	FDC9229B P0 Precompensation
B4	OUTPUT	Motor triggern
B5	OUTPUT	5,25" / 8" Umschaltung
B6	OUTPUT	MEM E050 Clock
B7	IN/OUTPUT	MEM E050 Data In/Output

5. Prominhalte5.1 Bussteuer-Prom (IC 10) TBP24SA10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
bis																
70	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
80	F	F	F	F	F	F	E	F	F	F	F	F	F	F	F	F
bis																
B0	F	F	F	F	F	F	E	F	F	F	F	F	F	F	F	F
C0	F	E	F	E	F	F	F	F	F	F	F	F	F	F	F	F
D0	F	E	F	E	E	F	F	F	F	F	F	F	E	F	F	F
E0	F	E	F	E	F	F	F	F	F	F	F	F	F	F	F	F
F0	F	E	F	E	F	F	F	F	F	F	F	F	F	F	F	F

5.2 Memory-Prom (IC 11) TBP24SA10

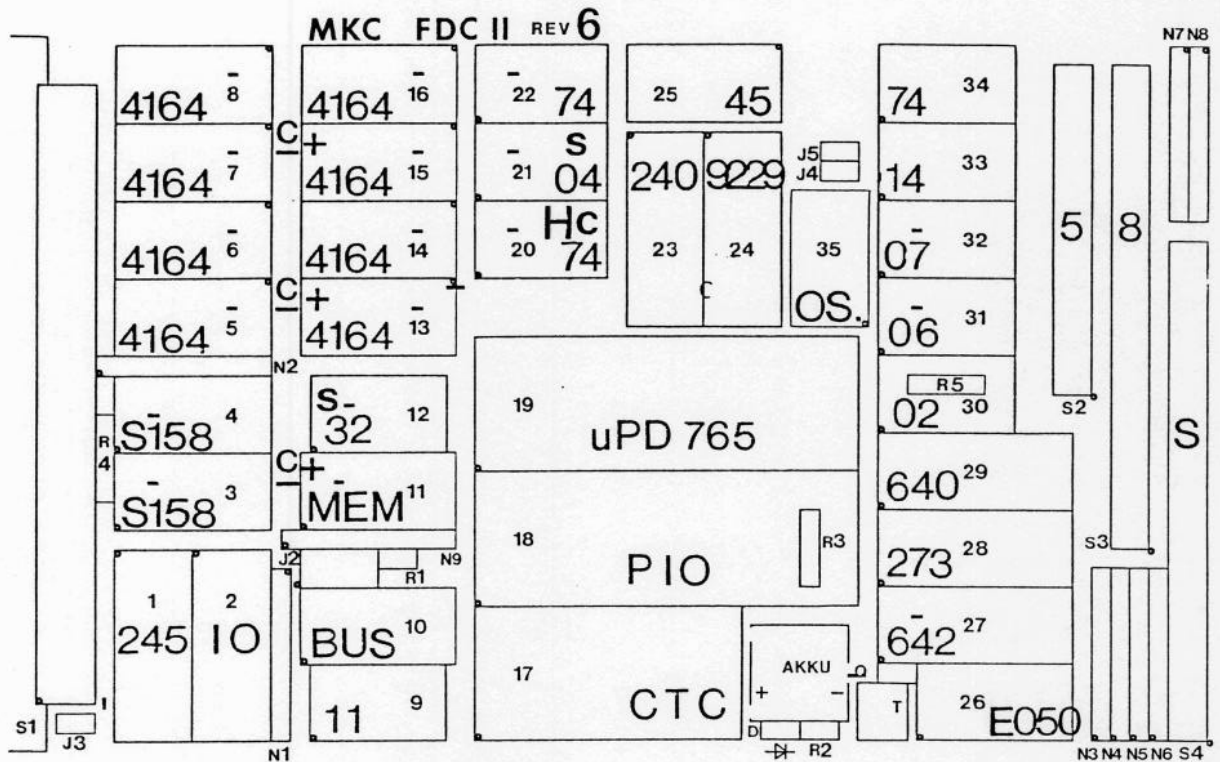
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
10	F	0	F	F	F	F	F	F	F	F	F	F	F	F	F	F
20	F	F	0	F	F	F	F	F	F	F	F	F	F	F	F	F
30	F	F	F	0	F	F	F	F	F	F	F	F	F	F	F	F
40	F	F	F	F	0	F	F	F	F	F	F	F	F	F	F	F
50	F	F	F	F	F	0	F	F	F	F	F	F	F	F	F	F
60	F	F	F	F	F	F	0	F	F	F	F	F	F	F	F	F
70	F	F	F	F	F	F	F	0	F	F	F	F	F	F	F	F
80	F	F	F	F	F	F	F	F	0	F	F	F	F	F	F	F
90	F	F	F	F	F	F	F	F	F	0	F	F	F	F	F	F
A0	F	F	F	F	F	F	F	F	F	F	0	F	F	F	F	F
B0	F	F	F	F	F	F	F	F	F	F	F	0	F	F	F	F
C0	F	F	F	F	F	F	F	F	F	F	F	F	0	F	F	F
D0	F	F	F	F	F	F	F	F	F	F	F	F	F	0	F	F
E0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	F
F0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0

5.3 I/O-Dekoder-Prom (IC 2) TBP28S42

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
bis																
150	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
160	FF	FF	FF	E7	FF	FF	FF	DF	FF	FF	FF	F6	FF	FF	FF	F6
170	FF	FF	FF	E7	FF	FF	FF	B7	FF	FF	FF	F6	FF	FF	FF	F6
180	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
bis																
1D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
1E0	FF	FF	FF	77	FF	FF	FF	77	FF	FF	FF	F5	FF	FF	FF	FF
1F0	FF	FF	FF	77	FF	FF	FF	77	FF	FF	FF	F3	FF	FF	FF	FF

6.1 Stückliste (die fett gedruckten Teile sind in MKC-Z-3048)

IC 1	74 LS 245 * == selekt. SCHOTTKY ==
IC 2	TBP 28 S 42 prog. (I/O-Prom)
IC 3, 4	74 S 158 o. 74 S 157 == SCHOTTKY ==
IC 5..8, 13..16	NEC 4164 o.ä. (64 K * 1 dyn. Ram, 128-REFRESH-Zyklen, 4Mhz 150ns / 5 oder 6MHz 120ns)
IC 9	74 LS 11
IC 10	TBP 24 SA 10 prog. (Bus-Prom)
IC 11	TBP 24 SA 10 prog. (MEM-Prom)
IC 12	74 S 32 * == selekt. SCHOTTKY ==
IC 17	280 A(B) CTC
IC 18	280 A(B) PIO
IC 19	uPD 765
IC 20	74 HC 74 == CMOS ==
IC 21	74 S 04 == SCHOTTKY ==
IC 22, 34	74 LS 74
IC 23	74 LS 240
IC 24	SMC FDC 9229B
IC 25	7445 == <b>Achtung: LAGE</b> ==
IC 26	MEM E050-16 Uhr
IC 27	74 LS 642
IC 28	74 LS 273
IC 29	74 LS 640
IC 30	74 LS 02
IC 31	7406 oder 7416
IC 32	7407 oder 7417
IC 33	74 LS 14
IC 35	TTL - Oszillator 16 MHz
Q	Quarz 32,768 KHz
D 1	1N4148 o.ä.
R 1	680 R
R 2, 3, 4	2K2
R 5, 6	470 R
N 1	SIL 8 * 680 R, pin 1 = common
N 2	SIL 8 * 1K2, pin 1 = common
N 3, 5, 8	SIL 8 * 220 R, pin 1 = common
N 4, 6, 7	SIL 8 * 330 R, pin 1 = common
N 9	SIL 8 * 2K2, pin 1 = common
T	3 bis 18 pF Trimmer
AKKU	2,4 Volt
C 1 bis 18	100 nF Vielschicht-C RM 2,54 mm
C 19..21	10 uF / 16 V Tantal
S 1	VG 64 a,c bestückt
S 2	34 pin Pfostenleiste
S 3, 4	50 pin Pfostenleiste
J 1, 3, 4, 5	2 pin Pfostenleiste
J 2	2 * 4 pin Pfostenleiste
14 Fassungen DIL-16	offen
10 Fassungen DIL-14	offen
7 Fassungen DIL-20	offen
2 Fassungen DIL-40	offen
1 Fassung DIL-28	offen
6 zweifach-Kurzschlußstecker	

6.2 Bestückungsplan6.3 Aufbauhinweise

Die Kondensatoren C1 bis C18 liegen fast alle unter den ICs. Daher sind offene Fassungen zu verwenden. Kondensatoren und Fassungen sollten gemeinsam bestückt werden. Bei den Widerstandsnetzwerken markiert der Punkt den gemeinsamen Anschluß (pin 1). Bei der Montage des Quarzoszillators ist darauf zu achten, daß das Metallgehäuse die danebenliegenden Pins der IC-Fassungen nicht berühren darf. Der AKKU sollte erst nach Fertigstellung der Karte eingelötet werden. Die Kondensatoren C19 bis C21 fehlen leider im Bestückungsplan auf der Platine, in obigem Bestückungsplan sind sie richtig eingezeichnet. Falls ein CMOS-Quarzoszillator verwendet wird, muß in den pin 11 des FDC9229B ein Widerstand von 1K5 eingebaut werden. Ein TTL-Oszillator ist jedoch wesentlich besser.

Das Uhren-IC MEM E050-16 darf erst nach Einbau des AKKUS eingesetzt werden.

Die Anschlußkabel zu den Laufwerken werden sehr eng nebeneinander gesteckt. Dies ist relativ einfach, wenn alle Kabel vom vorderen Rand der Karte kommen und man die Reihenfolge SASI, 8" und erst dann 5,25" einhält. Es sollten nur Quetschstecker ohne Polarisierung und Zugentlastung benutzt werden.

## 7. Jumper-Einstellungen

Auf der M.K.C. FDC II rev. 6 gibt es folgende Jumper:

J2	8 pin	binäre Einstellung der Pageadressen für den 64KByte Speicher
	1-2	A19 (geschlossen: = 0)
	3-4	A18 (geschlossen: = 0)
	5-6	A17 (geschlossen: = 0)
	7-8	A16 (geschlossen: = 0)
J3	2 pin	geschlossen: WAIT wird bei 6 MHz Systemtakt und 5,25" Floppylaufwerken benötigt.
J4	2 pin	geschlossen: READY = pin 6 MiniFloppy (für z.B. BASF)
J5	2 pin	geschlossen: READY = pin 34 Minifloppy (für z.B. TEAC)
S2	Anschlußstecker für 5,25" Floppy	
S3	Anschlußstecker für 8" Floppy	
S4	Anschlußstecker für SASI-Controller	

Auf dem XEBEC-Controller S1410 ist 512 Byte Sektorlänge zu jumpern (W3 nach 5 stecken). Beim Adress-Jumper muß S-0 verbunden sein (Voreinstellung). Das Winchester-Laufwerk muß Unit 0 sein! Der Terminator muß im XEBEC-Controller sein.

Beim XEBEC S1410A ist W1 nach 5 zu stecken (Sektorgröße 512 Byte). Achtung: J5 muß in Stellung 0 stehen!

Bei Anschluss von 5,25" und 8" Laufwerken muß der Terminator in dem jeweils letzten Laufwerk sein!

### Achtung:

Es dürfen nur maximal 2 Widerstandsnetzwerke in den Floppy-Laufwerken gesteckt sein

### Auslieferungszustand:

J2:	1-2 gesteckt	
	3-4 gesteckt	
	5-6 gesteckt	
	7-8 offen	=> Adresse: 10000 bis 1FFFF
J3:	bei 4 MHz offen	
	bei 6 MHz gesteckt	=> kein WAIT bei 4 MHz
J4:	offen	
J5:	gesteckt	=> TEAC - Laufwerke

```

TITLE   FDC II VERSION 1.2 (25.09.84 L.K.)
        .280
        .PHASE 100H          ; ORG 100H
;
;   Testroutinen fuer den M.K.C. FDC II
;
00F0    HLT   EQU    240      ; HEADLOADTIME IN MS (8 MHZ CLOCK)
000E    SRT   EQU    14      ; STEPRATETIME IN MS (8 MHZ CLOCK)
00F0    HUT   EQU    240      ; HEADUNLOADTIME IN MS (8 MHZ CLOCK)
;
00EE    HLTS  EQU    (<HLT-1>/2)*2
000F    HUTS  EQU    (HUT+15)/16
0002    SRTS  EQU    16-SRT
;
;   Mini-Disketten (400 OR 800 KBYTE)
;
;   DOUBLE DENSITY
;   40 or 80 TRACKS / DISK (each side)
;   10 SECTORS / TRACK
;   512 BYTE / SECTOR
;   DOUBLE SIDED
;
000A    MAXSEC EQU    10      ; 10 SECTORS
001E    FMTGAP EQU    1EH     ; END GAP LENGTH
;
;   RETRY COUNTER (NORM. 10 TIMES)
;
000A    RETRY  EQU    10
;
;   I/O PORTS
;
00F3    DACK   EQU    0F3H    ; DMA ACKNOWLEDGE
00F0    FDC    EQU    0F0H    ; STATUS PORT
00F1    FDD    EQU    FDC+1   ; DATA PORT
00F2    FTC    EQU    FDD+1   ; SECTOR TERMINATION
00F4    ctc0   equ    0f4h    ; uPD 765 interrupt
00F6    TIMER  EQU    0F6H    ; MOTOR-TIMER
00F8    P10A   EQU    0F8H    ; SASI AND CLOCK
00FA    P10B   EQU    0FAH    ; FLOPPY TIMING AND CLOCK
00F5    PRESC  EQU    0F5H
0047    CTCMOD EQU    01000111B
;
0100 025B 0000 vector: defw  fdint,0,0,0
0104 0000 0000
0108 0000 0000      defw  0,0,0,0
010C 0000 0000

```

```

;
; Command Table for uPD765
;
0110 00 CMTAB::DEFB 0 ; COMMAND BYTE
0111 01 UNIT:: DEFB 1 ; UNIT (0...3)
0112 1E TRACK:: DEFB 30 ; TRACK (0...39/77)
0113 00 HEAD:: DEFB 0 ; HEAD (0 for SIDE 0, 1 for SIDE 1)
; single sided is SIDE 0 only!
0114 01 SECTOR::DEFB 1 ; first SECTOR to read (1...10)
0115 02 DEFB 2 ; N = 512 Bytes/Sector
0116 0A LSTSEC::DEFB MAXSEC ; last SECTOR to read
0117 10 DEFB 10H ; READ/WRITE-GAP LENGTH
0118 80 DEFB 128 ; DATA-LENGTH (128 BYTE SECTORS only!)
;
; RESULT-TABLE FOR READ, WRITE AND READ ID
;
0119 00 REST: DEFB 0 ; STATUS 0 (01x00xxx ist kein Fehler)
; D7 & D6 = 00 normal Termination
; = 01 started, not succesful
; = 10 Invalid command
; = 11 Ready-Line Changes
; D5 = 1 Seek completed
; D4 = 1 Fault from Floppy or
; Track 0 not found
; D3 = 1 Floppy not ready
; D2 = Head Address at Interrupt
; D1 & D0 = Unit-Number at Interrupt
011A 00 DEFB 0 ; STATUS 1
; D7 = 1 Cylinderend (no error!)
; D6 = 0 not used
; D5 = 1 CRC-Error
; D4 = 1 Overrun, not serviced
; (READ, WRITE or FORMAT)
; D3 = 0 not used
; D2 = 1 ID not found
; D1 = 1 write protected
; D0 = 1 Missing Address Mark
011B 00 DEFB 0 ; STATUS 2
; D7 = 0 not used
; D6 = 1 deleted Address-Mark
; D5 = 1 CRC-Error in Data-Field
; D4 = 1 Wrong Cylinder
; D3 = 1 Scan equal hit
; D2 = 1 Scan not satisfied
; D1 = 1 Bad cylinder
; D0 = 1 Missing Address-Mark
011C 00 DEFB 0 ; TRACK
011D 00 DEFB 0 ; HEAD
011E 00 DEFB 0 ; SECTOR
011F 00 DEFB 0 ; N (SECTOR-LENGTH)
0120 00 errflg::defb 0 ; 0 if no error

```

```

;
;      INITIALIZE UPD 765 C and ZILOG-Chips, set IM2
;      ( SOFT RESET )
;*****
0121  INIFDC:      ; NO PARAMETERS
;*****
0121  F3          di
0122  21 0100     ld      hl,vector
0125  7C          ld      a,h
0126  ED 47       ld      i,a
0128  ED 5E       im      2
012A  7D          ld      a,1
012B  D3 F4       out     (ctc0),a      ; interrupt vector for ctc
012D  3E D7       ld      a,11010111b  ; counter to generate interrupt
012F  D3 F4       out     (ctc0),a
0131  3E 01       ld      a,1
0133  D3 F4       out     (ctc0),a
0135  3E 27       LD      A,00100111b    ; SET MOTOR-PRESCALER-CHANNEL
0137  D3 F5       OUT     (PRESC),A
0139  3E 00       LD      A,00
013B  D3 F5       OUT     (PRESC),A
013D  3E CF       LD      A,0CFH        ; SET PIO
013F  D3 F9       OUT     (PIOA+1),A
0141  3E 1F       LD      A,1FH
0143  D3 F9       OUT     (PIOA+1),A
0145  3E CF       LD      A,0CFH
0147  D3 FB       OUT     (PIOB+1),A
0149  3E 01       LD      A,1
014B  D3 FB       OUT     (PIOB+1),A
014D  3E 38       LD      A,38H          ; 125 ns 5 1/4"
;               LD      A,18H          ; 125 ns 8"
014F  D3 FA       OUT     (PIOB),A
0151  FB         ei
0152  06 0A       ini2:  ld      b,10
0154  10 FE       LOOP:  DJNZ   LOOP
0156  DB F0       IN      A,(FDC)
0158  FE 80       CP      80H          ; REQUEST FOR MASTER
015A  28 04       JR      Z,INI1        ; OK
015C  DB F1       IN      A,(FDD)        ; ELSE FETCH DATA
015E  18 F2       JR      INI2
0160  C9         INI1:  RET            ; DONE
;
;      SPECIFY DISC PARAMETERS pseudo-DMA-MODE !!
;      head-load-time, step-rate-time and head-unload-time
;****
0161  SPEC::      ; NO PARAMETERS
;****
0161  2A 0111     LD      HL,(UNIT)
0164  E5         PUSH   HL
0165  21 EE2F     LD      HL,HLTS*256+SRTS*16+HUTS    ; dma-mode
0168  01 0303     LD      BC,0303H
016B  22 0111     LD      (UNIT),HL
016E  CD 023E     CALL   CMFD
0171  E1         POP    HL
0172  22 0111     LD      (UNIT),HL
0175  C9         SPE2:  RET

```



```

;
;   move head to track 0, Track-Register is 0 after RESET
;   necessary one time for each Unit after RESET
;*****
0176 RECAL:: ; PARAMETERS: UNIT
;*****
0176 3E FF          ld      a,0ffh          ; track not reached
0178 32 0289       ld      (skdone),a
017B 01 0207       LD      BC,0207H          ; RECALIBRATE
017E CD 021E       CALL     MOTO            ; WAIT FOR DISK
0181 3A 0289       reclop: ld      a,(skdone)
0184 3C            inc      a
0185 28 FA         jr      z,reclop          ; wait until track reached
0187 C9           REC1:  RET              ; a = 20h ok, if not: aborted
;
;   SEEK TRACK
;   move the Head to a specified Track
;****
0188 SEEK:: ; PARAMETERS: UNIT, TRACK
;****
0188 3E FF          ld      a,0ffh
018A 32 0289       ld      (skdone),a
018D 01 030F       LD      BC,030FH          ; SEEK TRACK
0190 CD 021E       CALL     MOTO
0193 3A 0289       seklop: ld      a,(skdone)
0196 3C            inc      a
0197 28 FA         jr      z,seklop
0199 C9           SK1:   RET              ; YES => ALL DONE
;
;   READ ID-FIELD
;   to find the actual Track, read double density
;*****
019A RDIDF:: ; PARAMETERS: UNIT
;*****
019A 11 01A6       ld      de,rdidl
019D D5           push   de
019E 01 024A       LD      BC,024AH          ; READ ID
01A1 CD 021E       CALL     MOTO
01A4 18 FE       RDIDL: JR      RDIDL          ; WAIT FOR RESULTS
01A6 C9           RDID1: RET
;
;   SENSE DRIVE STATUS
;****
01A7 SDRS:: ; PARAMETERS: UNIT
;****
01A7 01 0204       LD      BC,0204H          ; SENSE DRIVE
01AA CD 023E       CALL     CMFD
01AD CD 0253       CALL     NEXT          ; A=STATUS 3
01B0 C9           SD1:   RET              ; D7 = Fault
; D6 = Write Protected
; D5 = Ready
; D4 = Track 0
; D3 = two sided
; D2 = Head-Address
; D1,D0 = Unit

```

```

;
; WRITE 512 SECTOR TO DISK DMA MODE !!
;*****
01B1 WR512: ; PARAMETERS: UNIT, TRACK, first SECTOR, last SECTOR
;*****
01B1 06 0A LD B,RETRY ; RETRY COUNTER
01B3 C5 WR5: PUSH BC ; SAVE COUNTER
01B4 01 0945 LD BC,0945H ; WRITE
01B7 CD 021E CALL MOTO
01BA 21 1000 LD HL,DBUF ; .BUFFER
01BD 0E F3 LD C,dack ; 256 BYTES, DATA-dma PORT
01BF 11 01CD LD DE,WR99 ; EXEC-END-ADDRESS
01C2 D5 PUSH DE ; ON STACK
01C3 DB FA WR6: IN A,(piob) ; REQUEST FOR MASTER ?
01C5 0F rrca
01C6 30 FB jr nc,wr6
01C8 ED A3 OUTI ; YES: TRANSFER BYTE
01CA C3 01C3 JP WR6
01CD C1 WR99: POP BC ; UNSAVE COUNTER
01CE 3A 0120 ld a,(errflg)
01D1 B7 or a
01D2 28 02 JR Z,WR8 ; NO ERRORS
01D4 10 DD DJNZ WR5 ; ERROR: TRY AGAIN
; FATAL ERROR
; RETURNS WITH A = 0 IF NO ERROR
; AND WITH A .NE. 0 IF ERRORS OCCURED
01D6 C9 WR8: RET
;
; READ 512 SECTOR DMA MODE !
;*****
01D7 RD512: ; PARAMETERS: UNIT, TRACK, first SECTOR, last SECTOR
;*****
01D7 06 0A LD B,RETRY ; COMPARE WR512
01D9 C5 RD4: PUSH BC
01DA 01 0946 LD BC,0946H ; READ
01DD CD 021E CALL MOTO
01E0 21 1000 LD HL,DBUF
01E3 11 01F3 ld de,rd99
01E6 D5 push de
01E7 0E F3 LD C,dack
01E9 DB FA RD5: IN A,(piob)
01EB 0F RRCA
01EC 30 FB JR NC,RD5
01EE ED A2 INI
01F0 C3 01E9 JP RD5
01F3 C1 rd99: POP BC
01F4 3A 0120 ld a,(errflg) ; error?
01F7 B7 or a
01F8 28 02 JR Z,RD7
01FA 10 DD DJNZ RD4
; RETURNS WITH A = 0 IF NO ERROR
; AND WITH A .NE. 0 IF ERRORS OCCURED
01FC C9 RD7: RET
;

```

```

01FD          RESULT: ; FETCH RESULTS OF READ/WRITE OPERATION, store in REST
                ; AND CHECK THEM FOR R/W-ERRORS
                ; DESTROYES: AF,BC,HL
                ; RETURNS WITH: Z & A=0, IF NO ERRORS
                ; AND WITH: NZ & A=? IF ANY ERROR
                ;
01FD 06 07     LD      B,7          ; 7 RESULTS
01FF 21 0119   LD      HL,REST     ; RESULT-TABLE
0202 CD 0253   RESLOP: CALL     NEXT
0205 77        LD      (HL),A      ; STORE RESULT
0206 23        INC      HL
0207 10 F9     DJNZ     RESLOP
0209 3A 0119   ld      a,(rest)
020C E6 98     and     10011000b
020E 4F        ld      c,a        ; save
020F 3A 011A   ld      a,(rest+1)
0212 E6 37     and     00110111b
0214 B1        or      c
0215 4F        ld      c,a
0216 3A 011B   ld      a,(rest+2)
0219 E6 7F     and     01111111b
021B B1        or      c          ; a = 0 if no error
021C 77        ld      (hl),a     ; store error-flag
021D C9        RET
                ;
021E          MOTO:  ; WAITS UNTIL DISK READY AND THEN
                ; TRANSMITS COMMAND TO FD-CONTROLLER
                ; DESTROYES: AF,BC,HL
                ;
021E 3E 47     LD      A,CTCMOD    ; start timer
0220 D3 F6     OUT     (TIMER),A
0222 AF        XOR     A
0223 D3 F6     OUT     (TIMER),A
0225 DB FA     in     a,(piob)     ; start motor
0227 CB A7     res     4,a
0229 D3 FA     out     (piob),a
022B CB E7     set     4,a
022D D3 FA     out     (piob),a
022F C5        PUSH    BC          ; SAVE COMMAND
0230 01 0204   MOTO1: LD      BC,0204H ; SENSE DRIVE
0233 CD 023E   CALL    CMFD
0236 CD 0253   CALL    NEXT        ; FETCH RESULT
0239 CB 6F     BIT     5,A         ; DISK READY ?
023B 28 F3     JR      Z,MOTO1    ; NO => WAIT
023D C1        POP     BC         ; YES => FETCH COMMAND
                ;

```

```

023E          CMFD:  ; TRANSMITS COMMAND TO FD-CONTROLLER
                ; DESTROYES: AF,BC,HL
                ;
023E 21 0110   LD    HL,CMDTAB      ; POINT TO COMMANDTABLE
0241 71       LD    (hl),C        ;
0242 0E F1    ld    c,fdd        ; data port
0244 3E 05    ld    a,5
0246 3D       cmdlop: dec    a
0247 20 FD    jr    nz,cmdlop
0249 DB F0    in    a,(fdc)      ; uPD ready for command
024B 07       rlca
024C 30 F8    jr    nc,cmdlop
024E ED A3    outi
0250 20 F4    jr    NZ,CMDLOP    ; SEND NEXT BYTE
0252 C9       RET                ; ALL BYTES TRANSFERRED

                ;
0253          NEXT:  ; READ NEXT RESULT-BYTE FROM FD-CONTROLLER
                ; DESTROYES: AF
                ;
0253 DB F0    next1: IN    A,(FDC)    ; REQUEST FOR MASTER &
0255 07       rlca
0256 30 FB    JR    nc,NEXT1      ; NO => WAIT
0258 DB F1    IN    A,(FDD)      ; YES => FETCH RESULT
025A C9       RET

                ;
                ; ***** interrupt service routine *****
                ;
025B 08       fdint: ex    af,af'
025C D9       exx
025D DB F0    fdintl: in    a,(fdc)    ; read status
025F CB 7F    bit    7,a          ; request for master
0261 28 FA    jr    z,fdintl      ; no: wait
0263 CB 77    bit    6,a
0265 28 06    jr    z,sense       ; not result phase
                ; ***** result phase *****
0267 CD 01FD  call    result
026A E1       pop    hl          ; don't return into loop
026B 18 17    jr    sensend
026D          sense: ;***** sense interrupt status *****
026D 01 0108  ld    bc,0108h
0270 CD 023E  call    cmfd
0273 CD 0253  call    next
0276 F5       push   af
0277 CD 0253  call    next
027A F1       pop    af
027B E6 E0    and    0e0h
027D FE C0    cp    0c0h
027F 28 03    jr    z,sensend

                ; ***** seek or recal done *****
0281 32 0289  ld    (skdone),a
0284          sensend:
0284 08       sensel: ex    af,af'
0285 D9       exx
0286 FB       ei
0287 ED 4D    reti

                ;

```

```
0289 FF      SKDONE: DEFB    0FFH          ; NOT DONE, 20H = DONE
;
;          SECTOR BUFFER
;
1000         DBUF    EQU    1000H          ; SECTOR BUFFER
;
;          .DEPHASE
;
;          END
```

```

        title    winchester-driver v.1.6
;
;   aufrufbare Routinen:
;       INIT:    initialisieren der Hardware
;       HDINIT:  reset XEBEC 1410
;       HDREAD:  einen Block lesen
;       HDWRIT:  einen Block schreiben
;
;*****
;*   Non-interrupt SASI-driver           *
;*****
        .z80
;*****
;*   i/o-ports                           *
;*****
00F8    fpioa    equ    0f8h           ; pio a-port
00FC    sasiin  equ    0fch           ; sasi read
00FD    sasiout equ    0fdh           ; sasi write
;*****
;*   SASI-Status bits                     *
;*****
0003    cd      equ    3              ; command/data
0002    io      equ    2              ; input/output
0000    req     equ    0              ; request
0001    busy    equ    1              ; busy
;*****
;*   initialize FDC II SASI-part         *
;*****
0000'   3E CF   init:   ld      a,0cfh
0002'   D3 F8           out    (fpioa),a
0004'   3E 1F           ld      a,1fh
0006'   D3 F8           out    (fpioa),a
0008'   C9           ret
;*****
;*   reset hard-disk-controller         *
;*****
0009'   DB F8   hdinit: in     a,(fpioa)   ; reset SASI-controller
000B'   CB F7           set    6,a         ; reset on
000D'   D3 F8           out    (fpioa),a
000F'   CB B7           res    6,a
0011'   D3 F8           out    (fpioa),a   ; reset off

```

```

;*****
;*      read sector from hard-disk      *
;*****
0013' CD 005F' hhread: call   hdrw           ; compute block-number
0016' CD 008A' hhdr1: call   select        ; select controller
0019' 3E 08          ld     a,8           ; read-command
001B' CD 00A2'          call   cmdout       ; send command
001E' 2A 00EA'          ld     hl,($dma)    ; buffer
0021' 0E FC          ld     c,sasiin      ;
0023' CD 0083' hhdr1p: call   rqwait       ; byte ready?
0026' CB 5F          bit    cd,a          ;
0028' 28 05          jr     z,hrdone      ; end of sector
002A' ED A2          ini     ; fetch byte
002C' C3 0023'          jp     hhdr1p      ; next byte
002F' CD 00AF' hrdone: call   getstat      ; fetch status
0032' C8          ret     z              ;
0033' CD 00BE'          call   herror      ; hard-disk-error
0036' 20 DE          jr     nz,hdrd1      ; try again
0038' C9          ret

;*****
;*      write sector to hard-disk      *
;*****
0039' CD 005F' hdwrit: call   hdrw           ; compute block-number
003C' CD 008A' hdwrl: call   select        ; select controller
003F' 3E 0A          ld     a,10          ;
0041' CD 00A2'          call   cmdout       ; send command
0044' 2A 00EA'          ld     hl,($dma)    ; buffer
0047' 0E FD          ld     c,sasiout     ;
0049' CD 0083' hdwrlp: call   rqwait       ; byte ready?
004C' CB 5F          bit    cd,a          ;
004E' 28 05          jr     z,hwrone      ; end of sector
0050' ED A3          outi          ; send byte
0052' C3 0049'          jp     hdwrlp      ; next byte
0055' CD 00AF' hwrone: call   getstat      ; fetch status
0058' C8          ret     z              ;
0059' CD 00BE'          call   herror      ; hard-disk-error
005C' 20 DE          jr     nz,hdwrl      ; try again
005E' C9          ret

;*****
;*      internal subroutine hdrw      *
;*****
005F' CD 0077' hdrw: call   seldrv
0062' 3A 00E9'          ld     a,($sect)
0065' 32 00DF'          ld     (la),a          ; lowest block-address (8 bit)
0068' 2A 00E7'          ld     hl,($trk)
006B' 7D          ld     a,l
006C' 32 00DE'          ld     (ma),a          ; middle block-address
006F' 3A 00DD'          ld     a,(drive)
0072' B4          or     h
0073' 32 00DD'          ld     (drive),a          ; highest block-address
0076' C9          ret

```

```

;*****
;*      internal subroutine seldrv      *
;*****
0077' 3A 00E6' seldrv: ld      a,(Sdrv)      ; relativ drive
007A' 06 05      ld      b,5
007C' 87      hdrwl: add     a,a
007D' 10 FD      djnz    hdrwl      ; compute unit
007F' 32 00DD'      ld      (drive),a
0082' C9      ret

;*****
;*      internal subroutine wait for request  *
;*****
0083' DB F8      rqwait: in     a,(fpioa)
0085' CB 47      bit     req,a
0087' 20 FA      jr      nz,rqwait
0089' C9      ret

;*****
;*      internal subroutine select      *
;*****
008A' DB F8      select: in     a,(fpioa)
008C' CB 4F      bit     busy,a
008E' 28 FA      jr      z,select
0090' CB EF      set     5,a      ; select
0092' 0E FD      ld      c,sasiout
0094' 06 01      ld      b,1
0096' ED 41      out     (c),b      ; controller-address
0098' D3 F8      out     (fpioa),a      ; select on
009A' CB AF      res     5,a      ; select off
009C' D3 F8      out     (fpioa),a
009E' 05      dec     b      ; b = 0
009F' 10 FE      sloop: djnz   sloop
00A1' C9      ret

;*****
;*      internal subroutine command-output  *
;*****
00A2' 01 06FD      cmdout: ld     bc,600h+sasiout ; 6 bytes command
00A5' 21 00DC'      ld     hl,task      ; command buffer
00A8' 77      ld     (hl),a      ; store command
00A9' CD 0083'      call   rqwait      ; wait for request
00AC' ED B3      otir      ; send command-bytes
00AE' C9      ret

;*****
;*      internal subroutine getstat      *
;*****
00AF'      getstat:
00AF' CD 0083'      call   rqwait      ; wait for request
00B2' DB FC      in     a,(sasiin)      ; fetch result
00B4' F5      push   af
00B5' CD 0083'      call   rqwait      ; wait for request
00B8' DB FC      in     a,(sasiin)      ; fetch second result
00BA' F1      pop    af
00BB' E6 02      and    2      ; nz = error
00BD' C9      ret

```



```

;*****
;*      internal subroutine hard-disk-error      *
;*****
00BE' CD 008A' herror: call  select
00C1' 3E 03          ld    a,3          ; request sense status
00C3' CD 00A2'          call  cmdout
00C6' 0E FC          ld    c,sasiin
00C8' 21 00E2'          ld    hl,hdres          ; result area
00CB' CD 0083' herr1: call  rqwait
00CE' CB 5F          bit    cd,a
00D0' 28 04          jr    z,herr9
00D2' ED A2          ini
00D4' 18 F5          jr    herr1
00D6' 3A 00E2' herr9: ld    a,(hdres)      ; result-byte
00D9' E6 7F          and   7fh          ; error?
00DB' C9          ret

;*****
;*      command-table for SCSI-controller      *
;*****
00DC' 00          task:: defb  0          ; command
00DD'          drive:
00DD' 00          ha:    defb  0          ; drive and highest address
00DE' 00          ma:    defb  0          ; middle address
00DF' 00          la:    defb  0          ; lowest address
00E0' 01          blkcnt: defb  1          ; block-count
00E1' 80          defb  80h          ; control-byte

;*****
;*      result-table for SCSI-controller      *
;*****
00E2' 0000 0000 hdres: defb  0,0,0,0      ; result area
;*****
;*      unit, track, sector, dma-address      *
;*****
00E6' 00          $rdrv: defb  0          ; unit 0
00E7' 0000          $trk:  defw  0          ; track 0
00E9' 00          $sect: defb  0          ; sector 0
00EA' 00EC'          $dma:  defw  buffer      ; dma-buffer
00EC'          buffer: defb  512          ; sector-size
; = 512 byte

end

```