

JK82 CPU III

Rev. 2.0.

TECHNISCHE BESCHREIBUNG

Bestellnummern:

TIJ-Z-1250	CPU III unbestückt
TIJ-Z-1252	CPU III bestückt (64K, 4MHz)
TIJ-Z-1253	CPU III bestückt (64K, 5MHz, DMA selektiert)
TIJ-Z-1254	CPU III bestückt (64K, 6MHz, DMA selektiert)
TIJ-Z-1255	CPU III bestückt (256K, 4MHz)
TIJ-Z-1256	CPU III bestückt (256K, 5MHz, DMA selektiert)
TIJ-Z-1257	CPU III bestückt (256K, 6MHz, DMA selektiert)
TIJ-Z-1258	PROM/PAL-Satz

Ihr autorisierter Händler: *****
* * * * *

(C) 1984 by Janich & Klass Wuppertal

14.07.84

Inhaltsverzeichnis:

1.	Allgemeine Beschreibung:	Seite 3
1.1	Jumperstellungen:	Seite 4
2.	Speicherverwaltung:	Seite 5
2.1.	Banking:	Seite 5
2.2.	Adreßumrechnung:	Seite 6
2.3.	Schaltungsbeschreibung der Adreßumrechnung:	Seite 9
2.4.	Laden des Bank-Description-Registers:	Seite 10
2.5.	Aufbau eines Bank-Description-Vektors:	Seite 11
3.	Inter-Bank-Move:	Seite 11
3.1.	Inter-Bank-Move mit DMA oder LDIR:	Seite 12
3.1.1	RD/WR-Steuerung beim DMA-Transfer:	Seite 12
3.1.2	RD/WR-Steuerung beim LDIR-Transfer:	Seite 12
4.	Beschaltung des CTC:	Seite 13
5.	I/O-Portbelegung:	Seite 13
5.1.	Die I/O-Ports im einzelnen:	Seite 14
5.1.1	Bank-Latche CWB, TSB und TDB:	Seite 14
5.1.2	BOOT/LDIR-Flipflop:	Seite 14
5.1.3	Basisadresse des Common:	Seite 14
5.1.4	Bank-Description-Vektor: (siehe 2.5.)	Seite 14
5.1.5	Set Watch Latch:	Seite 14
5.1.6	Read Watch Output:	Seite 15
6.	Bussteuerung:	Seite 15
6.1.	Inhalt des PROMs 1208:	Seite 15
7.	Internes Memory:	Seite 16
7.1.	Dynamisches RAM:	Seite 16
7.2.	EPROM der CPU III:	Seite 16
7.3.	Speicheraufteilung:	Seite 17
8.	Schaltungsbesonderheiten:	Seite 18
8.1.	RESET-Generierung:	Seite 18
8.2.	CMOS-Bestückung:	Seite 18
8.3.	Takterzeugung:	Seite 18
9.	Beispielsoftware:	Seite 19
10.	Busbelegung:	Seite 27
11.	Stückliste:	Seite 28
12.	Bestückungsdruck:	Seite 28
13.	Schaltplan:	Seite 29

1. Allgemeine Beschreibung:

Die jk82 CPU III findet ihre Anwendung in größeren Z80-Multiprozessor-systemen, die bereits mit einigen Slave-CPU's oder intelligenten I/O-Karten ausgerüstet sind. Sie besitzt eine Speicherverwaltung über 1MByte mit Adreßumrechnung, 64KB oder 256KB RAM, einen Z80-DMA, eine Z80-CTC sowie eine akkugepufferte Echtzeituhr.

Die CPU III erzeugt vier zusätzliche Adressen (A16 - A19), die den Adreßraum auf 1MByte erweitern. Es wird eine Umrechnung vom logischen Adreßraum in den physikalischen Adreßraum vorgenommen. Logisch existieren 64 Banks mit einer programmierbaren Größe von 0 - 60K (in 4K-Schritten). Die Basisadresse einer Bank wird durch die Hardware-Adreßumrechnung jeweils so verschoben, daß kein einziges Byte verloren geht.

Je nach Bestückung finden 64KB (4164) oder 256KB (41256) dynamisches RAM auf der Platine Platz. Die Refresh-Adresse ist 8Bit breit. Ein unter Software-Kontrolle ausblendbares EPROM von 2KB bis 16KB Größe ermöglicht BOOT-Betrieb.

Auf der Karte befindet sich ein Z80A-DMA Baustein, der einen schnellen Inter-Bank-Move ausführen kann. Auch schnelle I/O-Bausteine können mit dem DMA bedient werden. Der interne DMA besitzt immer die höchste Priorität.

Zusätzliche Logik ermöglicht den Inter-Bank-Move mittels LDIR/LDDR-Befehlen. Dabei ist die Programmierung ebenso transparent, wie bei der Verwendung des DMA.

Eine Echtzeituhr hat ebenfalls Platz gefunden. Die Uhr zeigt das Jahr, das Datum, den Wochentag sowie die Zeit in Stunden, Minuten und Sekunden. Zum Lieferumfang gehören Source-Listings zum Setzen und Lesen der Uhr. Über einen Kanal des Z80-CTC kann die Uhr auch Interrupts auslösen.

Die technischen Daten in Stichworten:

- effektive Speicherverwaltung von max. 1MByte Adreßraum
- 64KByte oder 256KByte Speicher auf der Karte
- Z80-DMA für schnellen Datentransfer
- Inter-Bank-Move mittels LDIR/LDDR-Befehlen
- Echtzeituhr mit Akkupufferung
- Einfach-Europakarte mit ECB-kompatiblen jk82-Bus
- Stromaufnahme typisch: 5V, 1,2A

1.1 Jumperstellungen:**Jumper S1:**

Der Jumper S1 schließt die BAI-BA0-Daisy-Chain. Dies ist erforderlich, falls mehr als ein zusätzlicher DMA im System betrieben wird. Der DMA der CPU III ist nicht in die Daisy-Chain eingebunden. Er liegt zwischen der CPU und der Busleitung BUSAK.

Im Auslieferungszustand ist der Jumper offen.

Jumper S2:

Die Stellung "a" ist erforderlich, falls die Karte mit RAMs des Typs 4164 bestückt ist. Die Stellung "b" ist erforderlich, falls die Karte mit RAMs des Typs 41256 bestückt ist.

Auslieferungszustand ist "a"

Jumper S3:

Die Stellung "a" ist erforderlich, falls die Karte mit einem EPROM des Typs 2732, 2764 bestückt ist. Die Stellung "b" ist erforderlich, falls die Karte mit einem EPROM des Typs 2716 bestückt ist.

Auslieferungszustand ist "a"

Jumper SA4 - SA7:

Die Basisadresse wird mit den Jumpern SA7 bis SA3 eingestellt. Dabei repräsentiert ein gesetzter Jumper eine logische 0. Die Adresse 20H ist also wie folgt einzustellen:

SA7 SA6 SA5 SA4



Auslieferungszustand!

2. Speicherverwaltung:

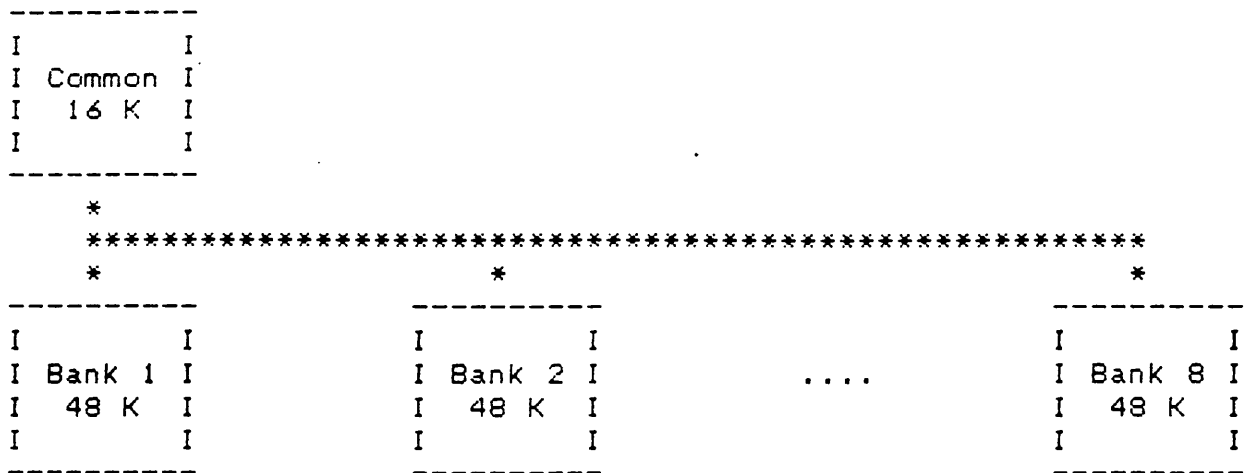
Viele neuere Betriebssysteme (CP/M PLUS) für 8Bit Prozessoren sind sogenannte "Banking" Systeme. Sie setzen das Vorhandensein mehrerer Speicherbanks von <64KB sowie einer Hardware, die das Umschalten zwischen einzelnen Banks gestattet, voraus. Die jk82 CPU III wurde speziell für derartige Betriebssysteme entwickelt.

2.1. Banking:

Definition: Eine "Bank" ist ein Speicherbereich von <64KB, der von einem Programm aus in den Adreßbereich der CPU eingeblendet werden kann.

Definition: Als "Common" wird der Speicherbereich in einem Banking-System benannt, der außerhalb jeder Bank liegt.

Grundsätzlich kann eine Z80-CPU nur einen Adreßraum von 64KB Größe adressieren. Nun könnte ein System aber beispielsweise 8 Speicherbanks mit je 48KB besitzen, von denen immer jeweils eine in den Adreßraum der CPU eingeblendet wird. Das folgende Bild zeigt dies schematisch.



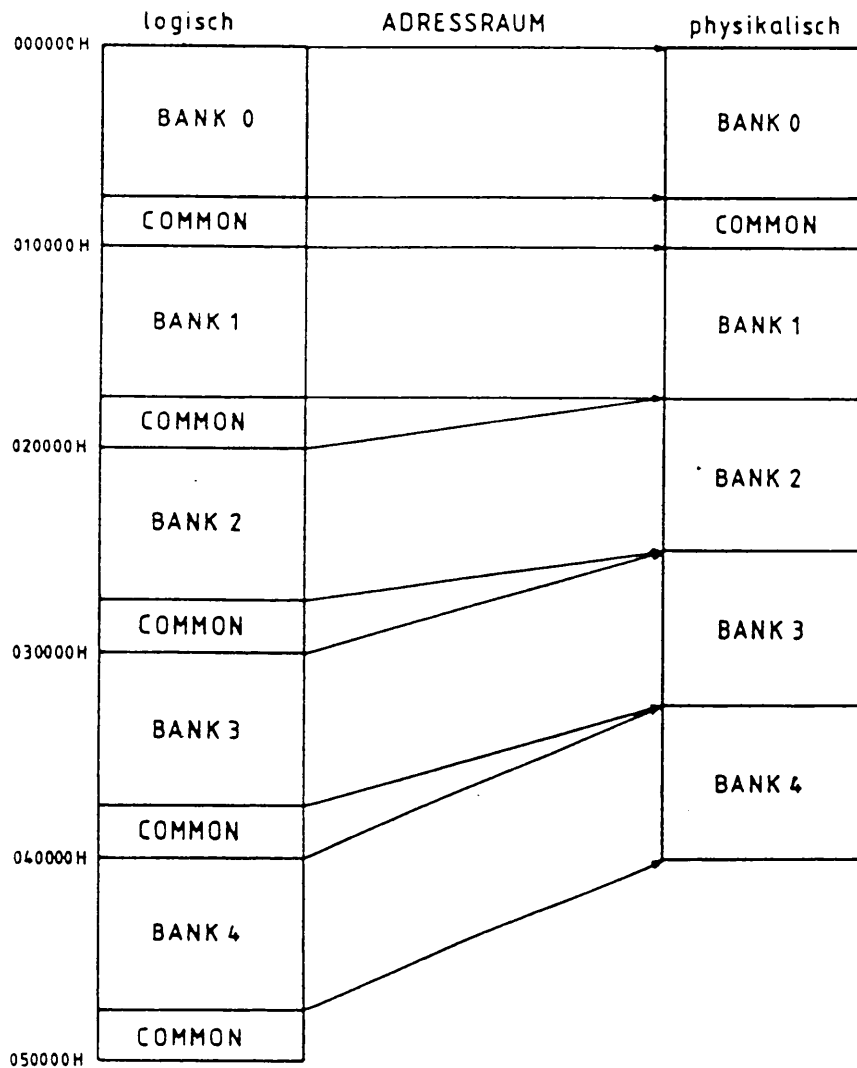
Die 16KB Common-Memory kann die CPU immer adressieren. Dort steht das Betriebssystem und auch das Unterprogramm zur Umschaltung auf eine andere Bank. Die einzelnen Banks können nun spezielle Programme (Multi-Tasking) oder Dateien enthalten. Müssen Übergabeparameter zwischen den Banks ausgetauscht werden, so kann das in den oberen 16KB des Common-Memory geschehen. Es wäre jedoch auch denkbar, daß ein DMA-Baustein einen "Inter-Bank-Move", also einen direkten Datentransfer zwischen zwei Banks durchführt.

Es tritt nun ein Problem auf. Unter der Voraussetzung, daß der Speicher mit RAMs vom Typ 4164 mit 64Kx1 aufgebaut wird, gehen jeder, außer der Bank 0, 16KB Speicherbereich verloren, da die einzelnen 64KB Blöcke adreßmäßig hintereinander liegen. Bei 5 Banks sind somit schon 64KB Speicher nicht mehr adressierbar.

Um dieses Problem zu umgehen, wurde auf der CPU III eine im folgenden beschriebene Adreßumrechnung auf Hardware-Ebene vorgenommen.

2.2. Adreßumrechnung:

Die grundsätzliche Überlegung zeigt das folgende Bild:

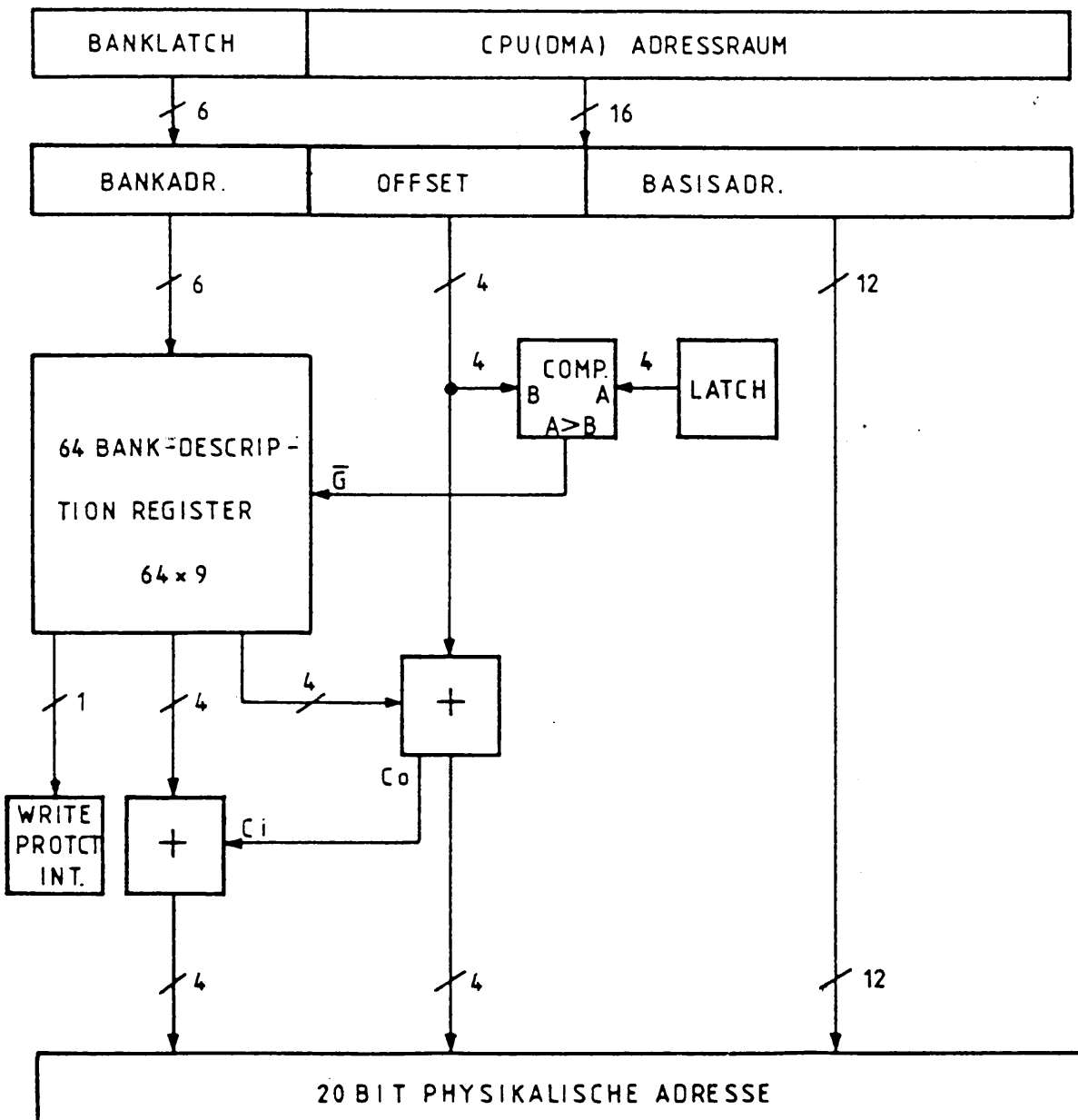


Der logische Adreßraum wird mit einer Abbildungsfunktion, dargestellt durch die Pfeile, auf den physikalischen Adreßraum abgebildet. Dabei wird der Common-Bereich nur einmal in den physikalischen Adreßraum abgebildet. Das Common-Memory ist normalerweise am oberen Ende des 64KB Adreßraums der CPU zu finden.

Die Größe des eingesparten Speicherraums ist nach folgender Formel zu berechnen:

- A = Anzahl der Banks
 - B = Speichergröße des Common-Memories (KB)
 - C = eingesparter Speicherraum (KB)
- $$C = (A - 1) \times B$$

Das folgende Bild zeigt die prinzipielle Schaltung der Adreßumrechnung auf der CPU III.



Der 16Bit breite Adreßbus der CPU wird in zwei Teile aufgeteilt. Die unteren 12Bit werden **Basisadresse** genannt, die oberen 4Bit werden **Offset** genannt. Zusätzlich existiert ein "Bank-Latch" in dem die **Bankadresse** gespeichert ist. Die Bankadresse ist 6Bit breit. Es können also maximal 64 Banks adressiert werden.

Das Bank-Latch ist von der CPU über I/O-Befehle zu laden. Die detaillierte Funktion ist der Schaltungsbeschreibung zu entnehmen.

Die **Bankadresse**, der **Offset** und die **Basisadresse** stellen zusammen die logische Adresse der CPU dar.

Diese wird in der folgenden Hardware in eine physikalische Adresse umgewandelt.

Die logische Basisadresse ist gleich der physikalischen Basisadresse.

Der Offset zerschneidet den logischen Adreßraum in 4KB große Kacheln.

Die Bankadresse führt auf ein bipolares RAM, das Bank-Description-Register. Dieses RAM ist durch ein Programm beschreibbar. Die Bankadresse adressiert nun eins der 64 9Bit breiten Worte dieses RAMs.

Die unteren 8Bit dieses Wortes geben die **Verschiebung** der Basisadresse einer Bank in 4KB Schritten im physikalischen Adreßraum an. Die folgende Additionsstufe addiert den jeweiligen Offset dazu, so daß die Abbildung der logischen in die physikalische Adresse innerhalb einer Bank **linear** verläuft.

Das neunte Bit ist als Speicherzugriffsindikator definiert. Ist es 0, so erzeugt jeder Speicherzugriff auf diese Bank ein Triggersignal für den CTC.

Im obigen Bild ist noch ein Komparator zu finden, der den Wert des Latches mit der Offsetadresse vergleicht. In dem Latch stehen die oberen 4Bit der unteren Adresse des Common-Bereichs. Ist die Offsetadresse größer oder gleich diesem Wert, so befindet sich die CPU im Common-Memory. Das Bank-Description-Register wird gesperrt, (die Ausgänge werden hier zu logisch 0 angenommen) und die unteren 16Bit der physikalischen Adresse sind gleich der Logischen. Die oberen 4Bit sind dann immer logisch 0. Die CPU kann also die Page 0 linear adressieren.

Festzuhalten ist, daß der logische größer als der physikalische Adreßraum ist.

2.3. Schaltungsbeschreibung der Adreßumrechnung:

Die ICs 31 und 32 (siehe Schaltplan) stellen an den Ausgängen B0 bis B5 die jeweilige Bankadresse zur Verfügung. Die Erzeugung dieser Bankadresse ist dem Abschnitt Inter-Bank-Move zu entnehmen.

Die 6Bit breite Bankadresse liegt an den Adreßleitungen des 64x9 bipolaren RAMs IC24. Die unteren 8 Datenausgänge führen auf die Addierer IC27 und IC28. Diese addieren das im RAM gespeicherte Wort zu den oberen 4 Adreßbits der CPU oder des DMA um daraus die Adressen A12 bis A19 zu bilden.

Pin 15 des bipolaren RAMs ist der CE. Ist dieses Signal High, so sind alle Ausgänge ebenfalls High. Das IC27 bekommt am Anschluß "Carry In" ebenfalls ein High-Signal. IC27 bildet dann die Summe $S = B + FH + 1$. Die Summenausgänge nehmen die Werte der B-Inputs an, zusätzlich wird "Carry Out" High. Das IC28 hat dann alle Ausgänge auf Low. Der Adreßbereich der CPU wird linear auf den Bus geführt und die Obergruppenadressen A16 - A19 sind Low.

Dieser Fall tritt immer dann ein, wenn der Ausgang A>B des IC36 Low ist. Nach PWCLR ist das immer der Fall, da die Ausgänge des IC36 alle Low sind. Nach dem Laden des Latches IC36 mit der **unteren Basisadresse des Common-Bereiches** wird das Bank-Description-Register nur gesperrt, falls die CPU oder der DMA im Common-Bereich arbeiten.

Im BOOT-Betrieb ist die Adreßumrechnung in den unteren 4KB ohne Bedeutung, da das EPROM die Adressen A12 bis A19 nicht zur Dekodierung verwendet. Im BOOT-Betrieb sollten die Latche IC31 und IC32 definiert geladen werden, so daß die CPU nach dem BOOT auf einer definierten Bank arbeitet. Dies kann aber auch nach dem Laden des Betriebssystems geschehen, falls das entsprechende Programm im Common-Memory steht.

Werden 8KB EPROMs verwendet, so ist darauf zu achten, daß die oberen 4KB durch die Adresse A12 selektiert werden. Diese Adresse führt über die Adreßumrechnung. In den unteren 4KB des BOOT-EPROMs muß also das Programm zum Laden des Bank-Description-RAMs liegen.

2.4. Laden des Bank-Description-Registers:

Die ICs 31 und 32 stellen zusammen 4 einzelne 6Bit Latches dar. Sie werden im Kapitel "Inter-Bank-Move" noch ausführlich beschrieben.

Das Register CWB (CPU-Work-Bank) wird zur Adressierung des Bank-Description-Registers IC24 benutzt.

Es hält somit die Adresse des zu schreibenden Bank-Description-Vektors. Dieser Vektor wird mit einem I/O-Befehl geladen (s. I/O-Portbelegung) Man beachte, daß die Datenspeicherung in diesem RAM invertiert erfolgt.

Es existieren zwei Ports. Der Port mit A0 = 0 setzt das neunte Bit des RAMs auf 1, der Port mit A0 = 1 setzt das Bit auf 0, so daß der CTC mit jedem Speicherzugriff auf diese Bank getriggert wird. Eine sinnvolle Verwendung dieses Bits ist noch offen.

Das Laden des Vektors kann nur aus dem Common-Memory heraus erfolgen. Das 64x9 RAM muß ein aktives CE-Signal (Pin 15) vor dem Write-Zugriff bekommen. Dies kann beim Beschreiben der Banks nur erfolgen, falls der Ausgang A>B des IC35 High wird. Hier wird folgende Eigenschaft der 280-CPU ausgenutzt:

Bei einem indirekten OUT-Befehl wird der Inhalt des Registers B auf die obere Hälfte des Adreßbus gelegt. Ist diese 0 und ist der Inhalt des Latches IC36 ungleich 0, so wird während des Schreibzugriffs auf das bipolare RAM das Signal A>B des IC35 aktiv. Es wird also das geforderte CE-Signal erzeugt.

Das Laden eines Bank-Description-Registers muß mit einem indirekten OUT-Befehl aus dem Common-Memory heraus erfolgen, wobei der Inhalt des CPU-Registers B 0 sein muß.

Im BOOT-Betrieb sollte mindestens die Bank 0 konfiguriert worden sein. D.h. in diesem Fall, daß der Vektor der Bank 0 auch den Wert 0 besitzen muß. Außerdem muß die untere Adresse des Common-Bereichs definiert geladen werden. Sie muß wie bereits erwähnt, ungleich 0 sein.

2.5. Aufbau eines Bank-Description-Vektors:

Der zu ladende Vektor stellt den invertierten Wert der oberen 8Bits der Basisadresse einer bestimmten Bank im physikalischen Speicherbereich dar.

```

*****
* --- * --- * --- * --- * --- * --- * --- * --- *
* A19 * A18 * A17 * A16 * A15 * A14 * A13 * A12 *
*      *      *      *      *      *      *      *      *
*****
*      *      *      *      *      *      *      *      *
* D7  * D6  * D5  * D4  * D3  * D2  * D1  * D0  *
*      *      *      *      *      *      *      *      *
*****
    
```

3. Inter-Bank-Move:

Sollen Daten zwischen zwei Banks transferiert werden, so bestehen im Prinzip zwei Möglichkeiten:

- 1) Die CPU kann die Daten aus einer Bank in reservierte Speicherbereiche des Common-Memories transferieren, die Bank wechseln und dann die Daten aus dem Common in die neue Bank transferieren.
- 2) Ein DMA-Baustein kann Daten direkt zwischen den Banks transferieren. Eine Lösung dazu wurde bereits auf der Address-Extension-Karte verwirklicht.

Die erste Möglichkeit ist zeitaufwendig. Der Transfer kann nur in kleinen Teilen erfolgen, will man den dafür reservierten Speicherplatz im Common nicht zu groß werden lassen.

Daher fand die zweite Möglichkeit Anwendung. Die Schaltung der Address-Extension-Karte wurde auf der CPU III in modifizierter und erweiterter Form realisiert. Zusätzlich wurde noch die Möglichkeit eines Inter-Bank-Move mittels LDIR- oder LDDR-Befehlen geschaffen.

3.1. Inter-Bank-Move mit DMA oder LDIR:

Auf der Karte ist ein 4x6Bit Latch untergebracht. In diesem Latch wird die Bankadresse gespeichert.

Latch 0: CWB (CPU-Work-Bank)

Enthält die Adresse der Bank, auf der die CPU gerade arbeitet. Ein externer DMA kann ebenfalls nur auf diese Bank zugreifen.

Latch 1: TSB (Transfer-Source-Bank)

Enthält die Bank, von der der DMA oder die CPU mit einem LDIR/LDDR bei einem Inter-Bank-Move liest.

Latch 2: TDB (Transfer-Destination-Bank)

Enthält die Bank, auf die der DMA oder die CPU mit einem LDIR/LDDR bei einem Inter-Bank-Move schreibt.

Latch 3: nicht benutzt.

3.1.1. RD/WR-Steuerung beim DMA-Transfer:

Die Initialisierung des DMA ist im Programmbeispiel im Anhang nachzulesen. Zuerst müssen die Latche TSB und TDB geladen werden. Dann wird der DMA mittels eines OTIR-Befehls geladen.

Das RD/WR Flipflop IC26 steuert den Transfer. Mit jeder positiven Flanke des MRQ-Signals wechselt es seinen Zustand. Ein M1-Signal setzt es zurück. Bei einem Memory zu Memory Transfer mit dem DMA ist der erste Zugriff immer ein Lese- der zweite ein Schreibzugriff usw. Aus dem Timing eines OTIR ist ersichtlich, daß der Ausgang des RD/WR Flipflop beim ersten DMA-Zugriff, also beim Lesen, High ist. Dies gilt nur, falls der DMA mit einem OTIR initialisiert wird. Das PAL (1258) setzt dies jedoch voraus.

3.1.2. RD/WR-Steuerung beim LDIR-Transfer:

Ist der DMA auf der Karte nicht bestückt, so kann derselbe Transfer auch mit einem LDIR/LDDR-Befehl vorgenommen werden. Ein Programmbeispiel dazu befindet sich ebenfalls im Anhang. Das LDIR-Flipflop IC10 wird vor dem LDIR mit einem OUT-Befehl gesetzt. Das Flipflop wird nach dem LDIR wieder zurückgesetzt.

Es wird hier vorausgesetzt, daß der Code im Common-Bereich abgearbeitet wird. Erfolgt ein Interrupt während des LDIR, so darf die Service-Routine keinesfalls außerhalb des Commons irgendeinen Speicherzugriff machen. Während das LDIR-Flipflop gesetzt ist, werden nämlich die Bank-Latche TSB und TDB abwechselnd selektiert. Nur im Common ist immer das CWB-Latch selektiert. Es ist zu empfehlen, die Interrupts vor Aufruf des Inter-Bank-Move Programmes zu sperren. Dies erscheint auch deshalb sinnvoll, da bei einem DMA-Transfer ebenfalls keine Interrupts verarbeitet werden.

4. Beschaltung des CTC:

Der Z80-CTC auf der CPU III kann mehrere Funktionen erfüllen. Die meisten Multi-Tasking oder Multi-User Betriebssysteme benötigen z.B. eine feste Interrupt-fähige Zeitbasis. Dieser Interrupt kann mit dem CTC-Baustein verwirklicht werden.

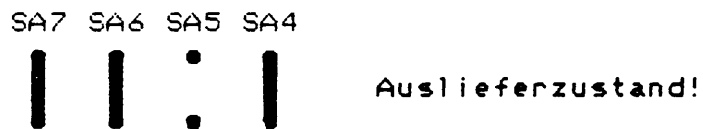
Der Triggereingang des Kanals 0 erhält mit jedem MRQ auf eine Bank, deren Vektor mit der höheren Adresse (s. I/O-Port Belegung) in das bipolare RAM geschrieben wurde, einen positiven Puls.

Der Triggereingang des Kanals 1 ist mit der Hardware-Uhr IC38 verbunden. Er erhält eine Frequenz von 32768Hz. Diese kann für Zeitsteuerungen herangezogen werden.

Der Triggereingang des Kanals 2 ist mit dem Ausgang des Kanals 1 verbunden um auch lange Zeiten im Sekundenbereich realisieren zu können.

5. I/O-Portbelegung:

Die Basisadresse wird mit den Jumpern SA7 bis SA3 eingestellt. Dabei repräsentiert ein gesetzter Jumper eine logische 0. Die Adresse 20H ist also wie folgt einzustellen:



A7	A6	A5	A4	A3	A2	A1	A0	RD/WR	Name	Funktion
X	X	X	X	0	0	0	0	RD/WR	CTC0	Kanal 0 des CTC
X	X	X	X	0	0	0	1	RD/WR	CTC1	Kanal 1 des CTC
X	X	X	X	0	0	1	0	RD/WR	CTC2	Kanal 2 des CTC
X	X	X	X	0	0	1	1	RD/WR	CTC3	Kanal 3 des CTC
X	X	X	X	0	1	0	0			nicht benutzt
X	X	X	X	0	1	0	1	WR	TSB	Transfer-Source-Bank
X	X	X	X	0	1	1	0	WR	TDB	Transfer-Destination-Bank
X	X	X	X	0	1	1	1	WR	CWB	CPU-Work-Bank
X	X	X	X	1	0	0	0	WR	CFF	Clock-Flip-Flop (BOOT and LDIR)
X	X	X	X	1	0	0	1	WR	BAC	Base-Address-Common
X	X	X	X	1	0	1	0	WR	BDV	Bank-Desc.-Vek. (ohne Attribut)
X	X	X	X	1	0	1	1	WR	BDVA	Bank-Desc.-Vek. mit Attribut
X	X	X	X	1	1	0	0	WR	SWL	Set-Watch-Latch
X	X	X	X	1	1	0	1	RD	RWO	Read-Watch-Output
X	X	X	X	1	1	1	0	RD/WR	DMA	DMA-Controller

5.1. Die I/O-Ports im einzelnen:

5.1.1. Bank-Latche CWB, TSB und TDB:

```

    D7      D6      D5      D4      D3      D2      D1      D0
*****
*      *      *      *      *      *      *      *
*  X   *  X   *  B5   *  B4   *  B3   *  B2   *  B1   *  B0   *
*      *      *      *      *      *      *      *
*****
    
```

B0 - B5 ist die Bankadresse.

5.1.2. BOOT/LDIR-Flipflop:

```

    D7 - D2      D1      D0
*****
*      *      *      *      *
*  X   *      LDIR   *      BOOT
*      *      *      *      *
*****
    
```

```

        LDIR = 0   LDIR-F.F. AUS   BOOT = 0   BOOT-EIN
        LDIR = 1   LDIR-F.F. EIN   BOOT = 1   BOOT-AUS
    
```

5.1.3. Basisadresse des Common:

```

    D7      D6      D5      D4      D3      D2      D1      D0
*****
*      *      *      *      *      *      *      *
*  X   *  X   *  X   *  X   *  A15 *  A14 *  A13 *  A12 *
*      *      *      *      *      *      *      *
*****
    
```

Die Basisadresse des Common-Memory setzt sich dann so zusammen:

A15 A14 A13 A12 X X X X

5.1.4. Bank-Description-Vektor: (siehe 2.5.)

5.1.5. Set Watch Latch:

Die Bedienung der Hardware-Uhr erfolgt seriell. Die Ausgänge des Watch-Latches IC39 sind nach einem PWCLR alle High (invertierte Ausgänge) und damit inaktiv.

Das Auslesen der Uhr geschieht relativ langsam. Aus dem Datenblatt der Uhr kann entnommen werden, daß die maximale Clockfrequenz von der Spannung des AKKUs abhängig ist. Diese beträgt mind. 2,2 Volt, so daß die Clockfrequenz max. 100kHz betragen darf. Die minimale Periodendauer beträgt demnach 10µs. Die Beispielsoftware arbeitet sicherheitshalber mit ca. 14µs.

D3 erzeugt den invertierten Clock der Uhr, wobei zu beachten ist, daß die Clockperiodendauer 10µs nicht unterschreiten darf.

D2 erzeugt den invertierten Chip-Select der Uhr.

D1 steuert den Lese- oder Schreibzugriff. Dabei gilt:

- 0 = Lesen
- 1 = Schreiben

D0 enthält den invertierten Wert des zu Schreibenden Bits.

5.1.6. Read Watch Output:

Beim Lesen der Uhr ist die Zugriffszeit nach Clock von min. 5µs zu beachten. Das zu Lesende Bit wird auf die Datenleitung D0 geschaltet.

6. Bussteuerung:

Die Steuerung der Daten- Adreß- und Steuerbustreiber wird im wesentlichen durch das PROM 1208 IC6 vorgenommen. Es wird nur aufgetaktet, falls ein gültiger Speicher- oder I/O-Zugriffszyklus stattfindet. Die Steuerung ist vollständig bezüglich aller vorkommender Zustände in einem Z80-System. So wird z.B. der RETI immer an alle I/O-Bausteine gelegt, unabhängig davon, ob der OPCODE-Fetch aus dem internen oder dem externen Memory erfolgt.

6.1. Inhalt des PROMs 1208:

Adresse:	Inhalt:
00	F F F F F F F F F F F F F F F F
10	F F F F 1 1 1 D 3 F 3 F B B F B
20	F F F F 9 D 9 D B F B F F 3 F 3
30	F F F F 9 D 9 D F F B F F B F 3
40	F F F F F F F F F F F F F F F
50	F F F F 1 1 1 D 3 F 3 F B B F B
60	F F F F 9 D 9 D B F B F F 3 F 3
70	F F F F 9 D 9 D F F B F F F F 3
80	F F F F F F F F F F F F F F F
90	F F F F 1 1 1 D 3 F 3 F B B F B
A0	F F F F 9 D 9 D B F B F F 3 F 3
B0	F F F F 9 D 9 D F F B F 3 F F 3
C0	F F F F F F F F F F F F F F F
D0	F F F F 1 1 1 D 3 F 3 F B B F B
E0	F F F F 9 D 9 D B F B F F 3 F 3
F0	F F F F 9 D 9 D F F B F F B F 3

7. Internes Memory:

Die CPU III besitzt 64K/256KB RAM und einen Steckplatz für ein EPROM von 2KB - 8KB Speichergröße.

7.1. Dynamisches RAM:

Die 8 Sockel für dynamische RAMs können wahlweise mit 64Kx1 oder mit 256Kx1 bestückt werden. Der Schalter S2 zeigt dem PAL1259 die Speichergröße an. Bei Verwendung von 256K Chips müssen die Adressen A16 und A17 gemultiplext auf Pin 1 der RAMs geführt werden. Dies geschieht im PAL (1259).

Die 64Kx1 Chips gibt es in zwei Versionen, mit 128 Refresh Zyklen also 7Bit und 256 Refresh Zyklen entsprechend 8Bit. Die Z80-CPU besitzt aber bekanntlich nur 7 Refresh-Adressen. Das Adreßbit A7 ist beim Refresh immer 0.

Das achte Adreßbit wird folgendermaßen erzeugt. IC7 ist ein Teiler 1:256, der lediglich die Refresh-Zyklen zählt. Sein Ausgang zeigt dem PAL die Wertigkeit des Adreßbits A7 beim Refresh an. Das PAL muß diese Adresse mit der CPU-Adresse A7 multiplexen.

Es bleibt anzumerken, daß die nun vorhandene 8Bit Refresh-Adresse zu RAMs mit nur 7Bit-Refresh aufwärtskompatibel ist. Die zukünftigen 256Kx1 RAMs benötigen jedoch mindestens eine 8Bit Refresh-Adresse.

7.2. EPROM der CPU III:

Der 28polige Sockel IC15 kann mit drei verschiedenen EPROM-Typen bestückt werden.

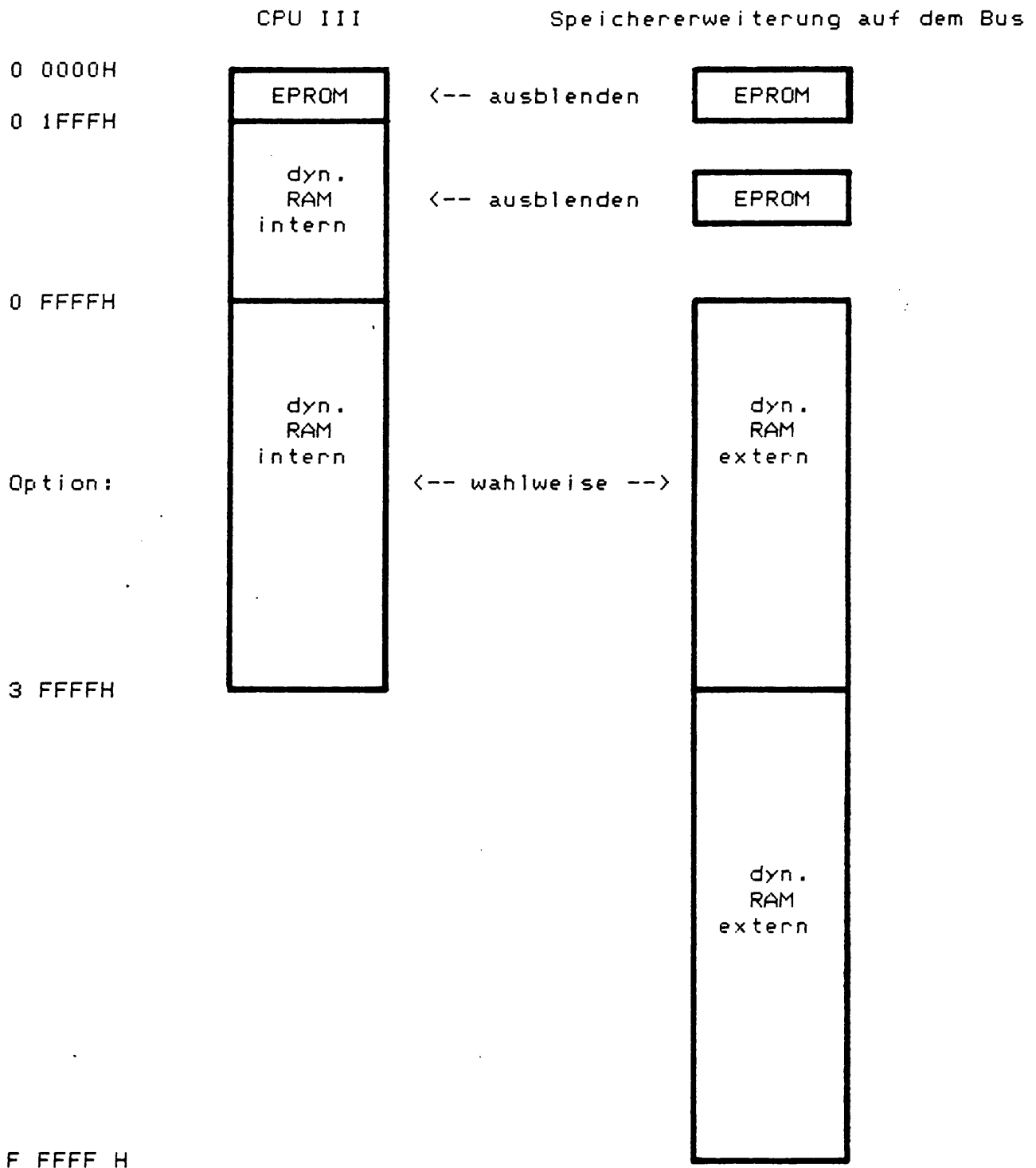
2716	S3 = (b)	
2732	S3 = (a)	
2764	S3 = (a)	Auslieferungszustand!

Dabei ist zu beachten, daß im BOOT-Betrieb, also mit eingeblendetem EPROM, immer die unteren 16KB des dynamischen RAM ausgeblendet werden. Da das EPROM dieser Karte praktisch nur als abschaltbarer BOOT sinnvolle Verwendung findet, ist diese Einschränkung tragbar.

7.3. Speicheraufteilung:

Im Adreßbereich von 1MB liegt das EPROM ab 00000H aufwärts. Das dynamische RAM liegt ebenfalls unten im Speicherbereich. Je nach Bestückung reicht es bis 0FFFFH oder 3FFFFH. Eine Speichererweiterung auf dem Bus (64/256K RAM-Karte) darf also erst bei 0FFFFH bzw. 3FFFFH beginnen.

Das DESELECT-Signal wird ebenfalls dekodiert. Durch DESELECT zeigt eine externe Speicherkarte (EPROM oder CMOS-RAM) an, das der jeweilige Speicherbereich des dynamischen RAMs ausgeblendet werden soll.



8. Schaltungsbesonderheiten:

8.1. RESET-Generierung:

Das IC41 ist ein Spannungskomparator, dessen Ausgang erst inaktiv wird, falls die Betriebsspannung 4,7V überschritten hat. Dadurch erhält man ein unter allen Umständen funktionierendes Power-On-Signal. Es ist jedoch zu beachten, daß das Netzteil tatsächlich 5V liefert. Einstellbare Netzteile sollten auf 5,1V abgeglichen werden. Ein 100Hz Brumm von mehr als 300mV ist nicht zu vertreten. Die CPU wird in einem solchen Fall ständig zurückgesetzt.

8.2. CMOS-Bestückung:

Die beiden Komparatoren IC35 und IC40 sind Bausteine des Typ 74HC85. Dies ist erforderlich, da sonst das zulässige Fan-In des internen Bus überschritten wird. Der Typ 74LS85 hat leider ein Fan-In von 3 auf den betreffenden Leitungen.

8.3. Takterzeugung:

Der Quarz Q1 schwingt auf der doppelten Systemtaktfrequenz. Das Signal wird durch das IC26 geteilt und anschließend durch den Transistor T1 auf den erforderlichen Pegel angehoben. Das ϕ Signal ist das einzige nicht TTL-kompatible Signal in der 280 Familie. Der Spannungshub muß hier mindestens 4V betragen.

9. Beispielsoftware:

```

;
; *** I/O-PORTS ***
;
BASIS EQU 20H ; BASE ADDRESS CPU III
CTC0 EQU BASIS ; CHANEL 0 OF CTC
CTC1 EQU BASIS+1 ; CHANEL 1 OF CTC
CTC2 EQU BASIS+2 ; CHANEL 2 OF CTC
CTC3 EQU BASIS+3 ; CHANEL 3 OF CTC

;
TSB EQU BASIS+5 ; TRANSFER SOURCE BANK
TDB EQU BASIS+6 ; TRANSFER DESTINATION BANK
CWB EQU BASIS+7 ; CPU WORK BANK

;
CFF EQU BASIS+8 ; CLOCK BOOT- AND LDIR F.F.

;
BAC EQU BASIS+9 ; BASE ADDRESS COMMON MEMORY

;
BDV EQU BASIS+0AH ; BANK DESCRIPTION VECTOR
BDVA EQU BASIS+0BH ; BANK DESCRIPTION VECTOR WITH ATTRIBUT

;
SWL EQU BASIS+0CH ; SET WATCH LATCH
RWO EQU BASIS+0DH ; READ WATCH OUTPUT

;
DMA EQU BASIS+0EH ; DMA CONTOLLER
;
;
; *****
; *
; * THE FOLLOWING PROGRAMM CONFIGURATES A MEMORY OF 320KB INTO 6 BANKS, EACH *
; * HAVING A SIZE OF 48KB. THE COMMON MEMORY IS DEFINED TO BE 16KB LARGE. *
; * THIS PROGRAMM CAN EITHER RUN IN THE BOOTSTRAPLOADER OR IN THE COMMON *
; * MEMORY. RECKOGNIZE, THAT AFTER A RESET THE BASE ADDRESS OF THE COMMON *
; * MEMORY IS ALWAYS ZERO. *
; *
; *****
;
;
;
; *** ENTRY: BKING ***
;
; *** DESTROYS: AF,BC,DE ***
;
;
;
BGR EQU 0CH ; SIZE OF BANKS: NUMBER OF 4K BLOCKS PER BANK
; (HEX) C = 12
;
;

```

```

BANK0: LD      A,00001100B      ; BASE ADDRESS COMMON
        ; FORMAT: (0,0,0,0,A15,A14,A13,A12)
        ;
        OUT     (BAC),A          ; 16K COMMON
        LD      B,0              ; DON'T CHANGE, REFER TO DESCRIPTION
        LD      C,BDV           ; BANK DESCRIPTION REGISTER ADDRESS
        ;
BANK0: LD      A,0              ; BANK NUMBER
        OUT     (CWB),A         ; CWB-PORT ADDRESSES VECTOR RAM
        LD      A,0              ; BASE ADDRESS OF BANK
        CPL     ; STORAGE IS INVERTED
        OUT     (C),A           ;
        ;
BANK1: LD      A,1              ; BANK NUMBER
        OUT     (CWB),A         ; CWB-PORT ADDRESSES VECTOR RAM
        LD      A,00010000B     ; BASE ADDRESS OF BANK
        ; FORMAT: (A19,A18,A17,A16,A15,A14,A13,A12)
        ;
        LD      E,A              ; SAVE FOR CALCULATION
        CPL     ; STORAGE IS INVERTED
        OUT     (C),A           ;
        ;
BANK2: LD      A,2              ; BANK NUMBER
        OUT     (CWB),A         ; CWB-PORT ADDRESSES VECTOR RAM
        LD      A,E              ;
        ADD     A,BGR            ; ADD SIZE OF BANK
        LD      E,A              ; SAVE FOR CALCULATION
        CPL     ; STORAGE IS INVERTED
        OUT     (C),A           ;
        ;
BANK3: LD      A,3              ; BANK NUMBER
        OUT     (CWB),A         ; CWB-PORT ADDRESSES VECTOR RAM
        LD      A,E              ;
        ADD     A,BGR            ; ADD SIZE OF BANK
        LD      E,A              ; SAVE FOR CALCULATION
        CPL     ; STORAGE IS INVERTED
        OUT     (C),A           ;
        ;
BANK4: LD      A,4              ; BANK NUMBER
        OUT     (CWB),A         ; CWB-PORT ADDRESSES VECTOR RAM
        LD      A,E              ;
        ADD     A,BGR            ; ADD SIZE OF BANK
        LD      E,A              ; SAVE FOR CALCULATION
        CPL     ; STORAGE IS INVERTED
        OUT     (C),A           ;
        ;
BANK5: LD      A,5              ; BANK NUMBER
        OUT     (CWB),A         ; CWB-PORT ADDRESSES VECTOR RAM
        LD      A,E              ;
        ADD     A,BGR            ; ADD SIZE OF BANK
        LD      E,A              ; SAVE FOR CALCULATION
        CPL     ; STORAGE IS INVERTED
        OUT     (C),A           ;
        ;
        XOR     A
        OUT     (CWB),A         ; CPU WORK BANK IS ZERO
        RET

```

```

; *****
; *
; *          INTER BANK MOVE USING THE DMA CONTROLLER          *
; *          DMAINI SHOULD ONLY BE CALLED ONCE AFTER POWER UP  *
; *
; *****
;
; *** ENTRIES: DMAINI - DMA INIT ***
;           DMANOV - DMA MOVE
;
; *** INPUT REGISTERS: ***
;
; HL - SOURCE ADDRESS
; DE - DESTINATION ADDRESS
; BC - BLOCKLENGTH
; H' - SOURCE BANK
; L' - DESTINATION BANK
;
; *** DESTROYS: AF,BC,DE,HL,HL' ***
;
DMAINI: LD    C,DMA      ; PORT ADDRESS
        LD    B,LEN1    ; BLOCK LENGTH OF TABLE
        LD    HL,TAB1   ; START OF TABLE
        OTIR                ; INITIALISE DMA
        RET

TAB1:  DEFB  01111101B  ; SOURCE ADDRESS & BLOCKLENGTH FOLLOWS,
        ; TRANSFER FROM PORT A TO B
        DEFW  0        ; SOURCE ADDRESS
        DEFW  0        ; BLOCKLENGTH - 1
        DEFB  00010100B ; PORT A IS MEMORY, ADDRESS INCREMENTS
        DEFB  00010000B ; PORT B IS MEMORY, ADDRESS INCREMENTS
        DEFB  10101101B ; CONTINUE MODE, NO INT, PORT B ADDRESS FOLLOWS
        DEFW  0        ; DESTINATION ADDRESS
        DEFB  10011010B ; STOP ON END OF BLOCK, CE/WAIT, READY AKTIV HIGH
        DEFB  0CFH     ; LOAD STARTING ADDRESS FOR BOTH PORTS, CLEAR
        ; BYTE COUNTER
        DEFB  083H     ; DISABLE DMA, (CONTROLL REGISTERS ARE NOT AFFECTED)
LEN1  EQU  $-TAB1

;
;
DMANOV: LD    (SOURCE),HL
        LD    (DESTIN),DE
        DEC   BC      ; BLOCK LENGTH FOR DMA HAS TO BE BLOCK LENGTH - 1
        ; AND SHOULD NOT BE GREATER THAN 0400H = 1K.
        LD    (LENGTH),BC
        EXX
        LD    A,H      ; SOURCE BANK
        OUT   (TSB),A
        LD    A,L      ; DESTINATION BANK
        OUT   (TDB),A
        EXX

        LD    C,DMA    ; PORTADRESSE
        LD    B,LEN2   ; BLOCK LENGTH OF TABLE
        LD    HL,TAB2  ; START OF TABLE
        OTIR                ; INITIALIZE DMA
        RET

```

```

TAB2:  DEFB  01111101B  ; SOURCE ADDRESS & BLOCK LENGTH FOLLOWS,
                                ; TRANSFER FROM PORT A TO B
SOURCE: DEFW  0          ; SOURCE ADDRESS
LENGTH: DEFW  0          ; BLOCKLENGTH - 1
        DEFB  10101101B ; CONTINUE MODE, NO INT, PORT B ADDRESS FOLLOWS
DESTIN: DEFW  0          ; DESTINATION ADDRESS
        DEFB  0CFH      ; LOAD STARTING ADDRESS FOR BOTH PORTS, CLEAR
                                ; BYTE COUNTER
        DEFB  087H      ; ENABLE DMA, (CONTRALL REGISTERS ARE NOT AFFECTED)
LEN2   EQU   $-TAB2

```

```

; *****
; *
; *          INTER BANK MOVE USING THE LDIR F.F.
; *
; *****
;
;
; *** ENTRIES: LDIR - INTER BANK MOVE USING THE LDIR ***
;
; *** INPUT REGISTERS: ***
;
; HL - SOURCE ADDRESS
; DE - DESTINATION ADDRESS
; BC - BLOCKLENGTH
; H' - SOURCE BANK
; L' - DESTINATION BANK
;
; *** DESTROYS: AF,BC,DE,HL,HL' ***
;
LDIR:  EXX
        LD   A,H          ; SOURCE BANK
        OUT  (TSB),A
        LD   A,L          ; DESTINATION BANK
        OUT  (TDB),A
        EXX

;BOOT: LD   A,0          ; IF USED INSIDE THE BOOTSTRAPLOADER
NBOOT: LD   A,0FFH       ; IF USED OUTSIDE OF THE BOOTSTRAPLOADER
        SET  1,A
        OUT  (CFF),A     ; LDIR F.F. ON
        LDIR
        RES  1,A
        OUT  (CFF),A     ; LDIR F.F. OFF
        RET

```



```

RTIME: LD      B,00000111B      ; ADDRESS FOR SECONDS
      CALL    RREG
      LD      B,C                ; B = SECOND
      PUSH   BC
      LD      B,00000011B      ; ADDRESS FOR MINUTES
      CALL    RREG
      POP    DE
      LD      B,D                ; C = MINUTE
      PUSH   BC
      LD      B,00000101B      ; ADDRESS FOR HOURS
      CALL    RREG
      LD      D,C                ; D = HOUR
      POP    BC
      RET

      ;
      ;
RDATE: LD      B,00000001B      ; ADDRESS FOR DAYS
      CALL    RREG
      LD      E,C                ; E = DAY
      LD      B,00000110B      ; ADDRESS FOR MONTH
      CALL    RREG
      LD      H,C                ; H = MONTH
      LD      B,00000100B      ; ADDRESS FOR JEARS
      CALL    RREG
      LD      L,C                ; L = JEAR
      RET

      ;
      ;
;
;
; *** WREG ***
;
; THIS PROGRAM WRITES TO A REGISTER OF THE CMOS WATCH.
;
; INPUT REGISTER: B = REGISTER ADDRESS
;                  (FORMAT: 0,0,0,0,1,A0,A1,A2; A0-A2 ARE INVERTED)
;
;                  C = DATA TO BE WRITTEN
;                  (FORMAT: TWO WORD BCD)
;
;
;
WREG:  PUSH   BC
      PUSH   DE
      PUSH   AF
      LD     A,C
      CPL
      LD     C,A                ; WRITE DATA INVERTED
      LD     D,4                ; BIT COUNTER
WREG0: LD     A,B
      OR     0EH                ; CE = LOW, CLOCK = LOW, WRITE MODE SELECTED
      CALL   WCLOCK            ; BIT WRITE
      RRC   B                  ; NEXT BIT
      DEC   D
      JR    NZ,WREG0          ; IF ZERO THAN 4BITS DONE

```



```

LD      D,8          ; BIT COUNTER
WREG1: RLC      C          ; MSB FIRST
LD      A,C
OR      0EH          ; CE = LOW, CLOCK = LOW, WRITE MODE SELECTED
CALL   WCLOCK       ; BIT WRITE
DEC     D
JR      NZ,WREG1     ; IF ZERO THAN 8BITS DONE
XOR     A            ; DISABLE WATCH
OUT     (SWL),A
POP     AF
POP     DE
POP     BC
RET

```

```

;
;
;

```

```

; *** RREG ***
;

```

```

; THIS PROGRAM READS FROM A REGISTER OF THE CMOS WATCH.
;

```

```

; INPUT REGISTER: B = REGISTER ADDRESS
;                 (FORMAT: 0,0,0,0,1,A0,A1,A2; A0-A2 ARE INVERTED)
;

```

```

; OUTPUT REGISTER: C = DATA READ
;                 (FORMAT: TWO WORD BCD)
;
;
;

```

```

RREG:  PUSH  DE
      PUSH  AF
LD      D,4          ; BIT COUNTER
RREG0: LD      A,B
OR      0EH          ; CE = LOW, CLOCK = LOW, WRITE MODE SELECTED
CALL   WCLOCK       ; BIT WRITE
RRC     B            ; NEXT BIT
DEC     D
JR      NZ,RREG0    ; IF ZERO THAN 4BITS DONE
;
LD      D,8          ; BIT COUNTER
LD      C,0FFH      ; ALL BITS ON
;
RREG1: CALL   RCLOCK ; BIT READ
      BIT   0,A
JR      NZ,RREG2
RES     0,C
RREG2: RRC     C            ; LSB FIRST
DEC     D
JR      NZ,RREG1    ; IF ZERO THAN 8BITS DONE
XOR     A            ; DISABLE WATCH
OUT     (SWL),A
POP     AF
POP     DE
RET

```

```

WCLOCK: PUSH    BC
            OUT    (SWL),A      ; CLOCK OF WATCH DURING WRITE
            LD     B,XMHZ      ; T = 1/2*CLOCK
L1WCLK: DJNZ   L1WCLK          ; CA. 7YS
            RES    3,A         ; CLOCK = HIGH
            OUT    (SWL),A
            LD     B,XMHZ
L2WCLK: DJNZ   L2WCLK
            SET    3,A         ; CLOCK = LOW
            OUT    (SWL),A
            POP    BC
            RET

;
RCLOCK: PUSH    BC            ; CLOCK OF WATCH DURING READ
            LD     A,00001100B ; CE = LOW, CLOCK = LOW, READ SELECTED
            OUT    (SWL),A
            PUSH   AF         ; SAVE CONTROLL BYTE
            LD     B,XMHZ      ; T = 1/2*CLOCK
L1RCLK: DJNZ   L1RCLK          ; CA. 7YS
            RES    3,A         ; CLOCK = HIGH
            OUT    (SWL),A
            IN     A,(RMO)     ; READ WATCH
            LD     C,A         ; SAVE DATA
            LD     B,XMHZ
L2RCLK: DJNZ   L2RCLK
            POP    AF         ; RESTORE CONTROLL BYTE
            SET    3,A         ; CLOCK = LOW
            OUT    (SWL),A
            POP    AF         ; RESTORE DATA
            LD     A,C         ; RESTORE DATA
            POP    BC
            RET

```

10. Busbelegung

Name	Signal	LS-Fan-Out/In
A0	Adresse 0	60 1
A1	Adresse 1	60 1
A2	Adresse 2	60 1
A3	Adresse 3	60 1
A4	Adresse 4	60 1
A5	Adresse 5	60 1
A6	Adresse 6	60 1
A7	Adresse 7	60 1
A8	Adresse 8	60 1
A9	Adresse 9	60 1
A10	Adresse 10	60 1
A11	Adresse 11	60 1
A12	Adresse 12	60 1
A13	Adresse 13	60 1
A14	Adresse 14	60 1
A15	Adresse 15	60 1
A16	Adresse 16	20 --
A17	Adresse 17	20 --
A18	Adresse 18	20 --
A19	Adresse 19	20 --
D0	Data 0	60 1
D1	Data 1	60 1
D2	Data 2	60 1
D3	Data 3	60 1
D4	Data 4	60 1
D5	Data 5	60 1
D6	Data 6	60 1
D7	Data 7	60 1
<u>MRQ</u>	Memory Request	60 1
<u>IORQ</u>	I/O Request	60 1
<u>RD</u>	Read	60 1
<u>WR</u>	Write	60 1
<u>HALT</u>	Halt	60 --
<u>MI</u>	Maschinenzyklus 1	60 --
<u>RFSH</u>	Refresh	60 --
<u>WAIT</u>	Wait	-- 3,2mA
ϕ	Clock	24mA --
2ϕ	2 x Clock	20 --
<u>PWCLR</u>	Power On Clear	60 --
<u>RESET</u>	Reset	-- 4mA
<u>DESELECT</u>	Deselect	3,2mA
<u>DMARDY</u>	DMA Ready	3,2mA
<u>IEI</u>	Int. Enable In	-- 2
<u>IEO</u>	Int. Enable Out	20 --
<u>INT</u>	Interrupt	-- 3,2mA
<u>NMI</u>	Non Maskable Int	-- 3,2mA
<u>BAI</u>	Busacknowledge In	S1
<u>BAO</u>	Busacknowledge Out	
<u>BUSAK</u>	Busacknowledge	60 --
<u>BUSRQ</u>	Busrequest	-- 3,2mA

Belegung der VG-64 Leiste

	a	c
+5V	1	1 +5V
D5	2	2 D0
D6	3	3 D7
D3	4	4 D2
D4	5	5 A0
A2	6	6 A3
A4	7	7 A1
A5	8	8 A8
A6	9	9 A7
<u>WAIT</u>	10	10 A16
<u>BUSRQ</u>	11	11 IEI
<u>BAI</u>	12	12 A17
	13	13 A18
A19	14	14 D1
	15	15
2ϕ	16	16 IEO
<u>BAO</u>	17	17 A11
A14	18	18 A10
	19	19
<u>MI</u>	20	20 <u>NMI</u>
	21	21 <u>INT</u>
<u>DMARDY</u>	22	22 <u>WR</u>
	23	23
	24	24 <u>RD</u>
	25	25 <u>HALT</u>
<u>DESELECT</u>	26	26 <u>PWCLR</u>
<u>IORQ</u>	27	27 A12
<u>RFRSH</u>	28	28 A15
A13	29	29 ϕ
A9	30	30 <u>MRQ</u>
<u>BUSAK</u>	31	31 <u>RESET</u>
GND	32	32 GND

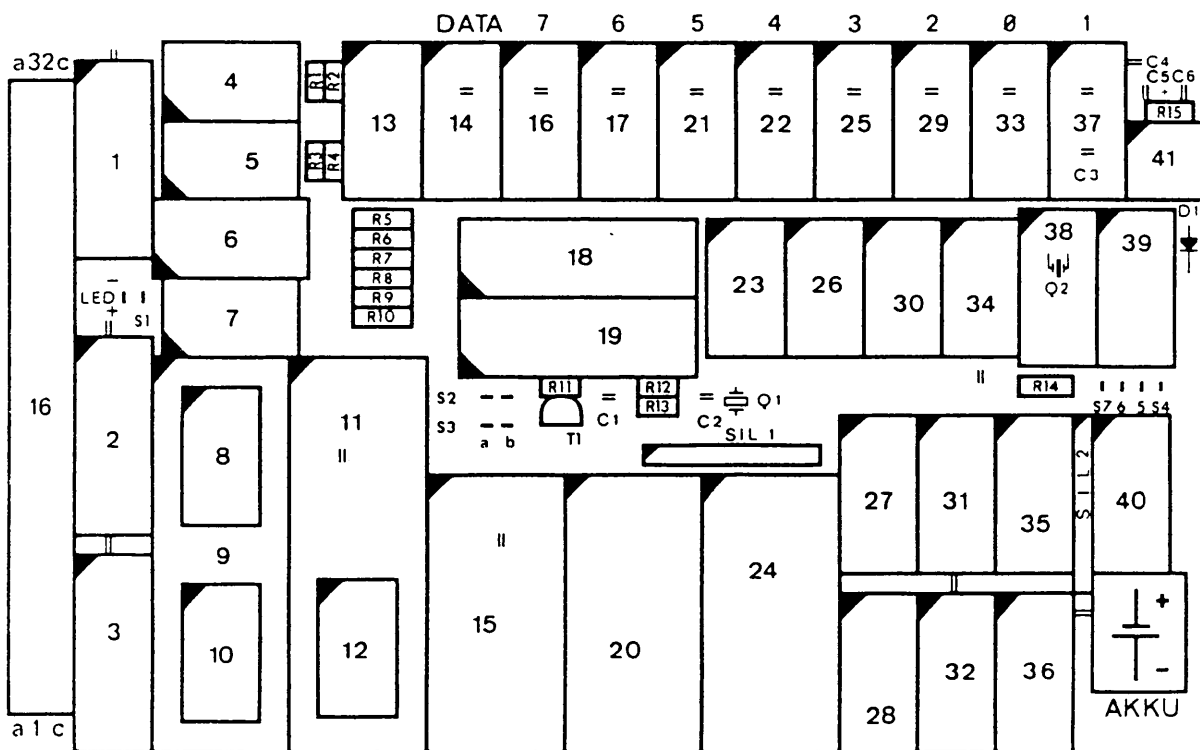
Die Stromaufnahme beträgt ca. 1,2A auf den 5V bei Bestückung mit 64KB und einem EPROM 2764.

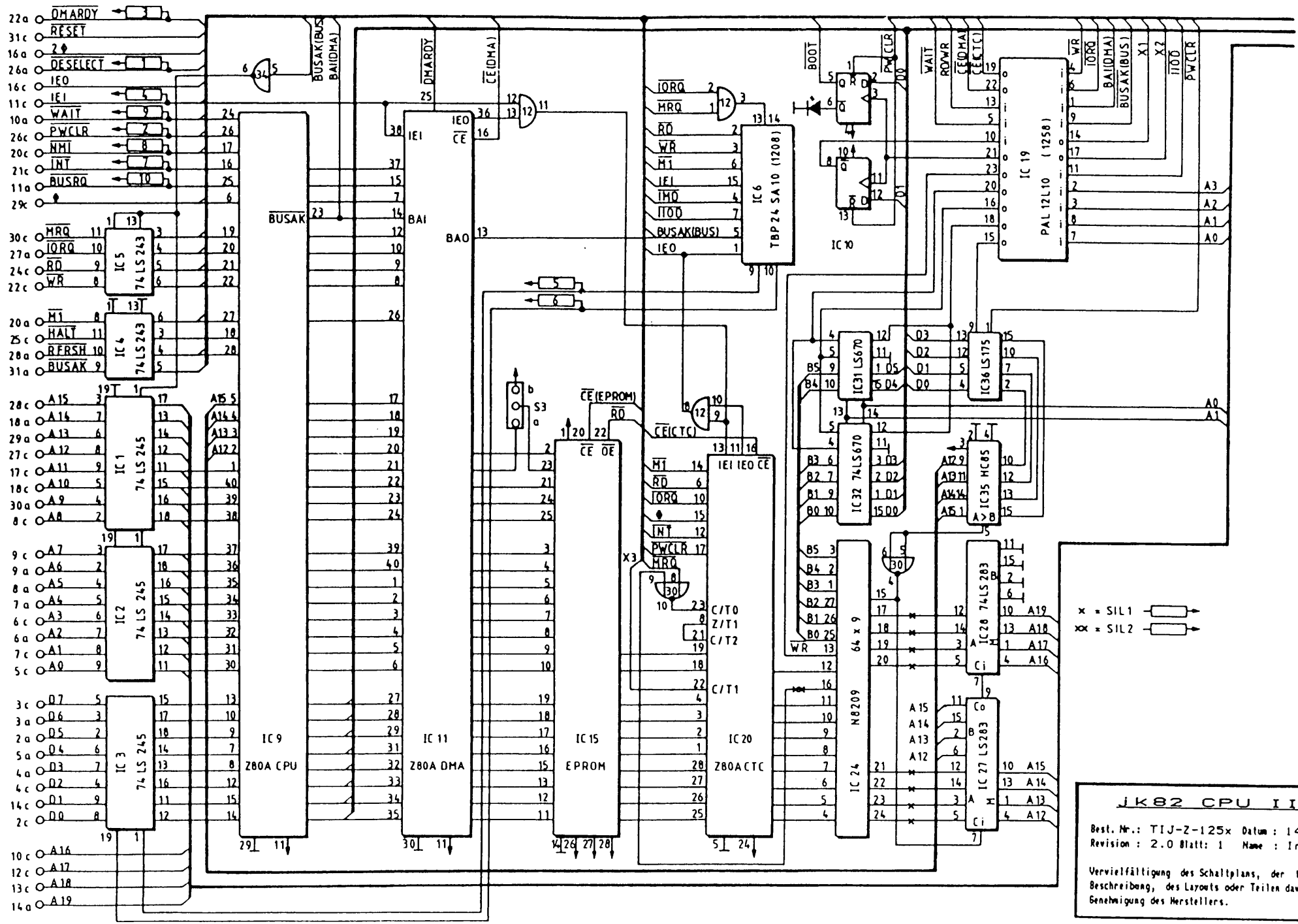
11. Stückliste:

IC 1 - 3	74LS245	R 1 - 3,	
IC 4 - 5	74LS243	7 - 10	1K5
IC 6	Prom (1208)	R 4, 14	4K7
IC 7	74LS393	R 5, 6	680R
IC 8	74LS125	R 11	12K
IC 9	Z80A-CPU	R 12, 13	330R
IC 10, 26	74LS74	R 15	22K
IC 11	Z80(A/B)-DMA(selekt.)	SIL 1, 2	8 x 1K2
IC 12	74LS08		
IC 13, 14	74157	C 1	33pF
IC 15	BYTEWIDE-Sockel	C 2 - 5	100nF RM2,5
IC 16,17,21,22, 25,29,33,37	4164/41256 128/256 R.C. (150ns)	C 6	10yF 16V
IC 18	PAL14L8 (1259)	C 17 Stück a'	100nF RM2,5
IC 19	PAL12L10 (1258)		
IC 20	Z80(A/B)-CTC	D 1	1N914
IC 23	74S04 Schottky !		
IC 24	N82S09/19 (64x9)	T 1	2N2906
IC 27, 28	74LS283		
IC 30	74LS02	Q 1	8MHz/10MHz/12MHz
IC 31, 32	74LS670	Q 2	32768Hz
IC 34	74LS04		
IC 35, 40	74HC85 CMOS !		UG-64 Messerleiste
IC 36, 39	74LS175		Pfostenleiste 18 pol.
IC 38	MEM E050-16		Steckbrücken 5 Stück
IC 41	TL7705A CPA		AKKU 2DK60

Socket 8polig 300mil.	1 Stück	Socket 24polig 300mil.	2 Stück
Socket 14polig 300mil.	7 Stück	Socket 28polig 600mil.	3 Stück
Socket 16polig 300mil.	20 Stück	Socket 40polig 600mil.	2 Stück
Socket 20polig 300mil.	3 Stück		

12. Bestückungsdruck:



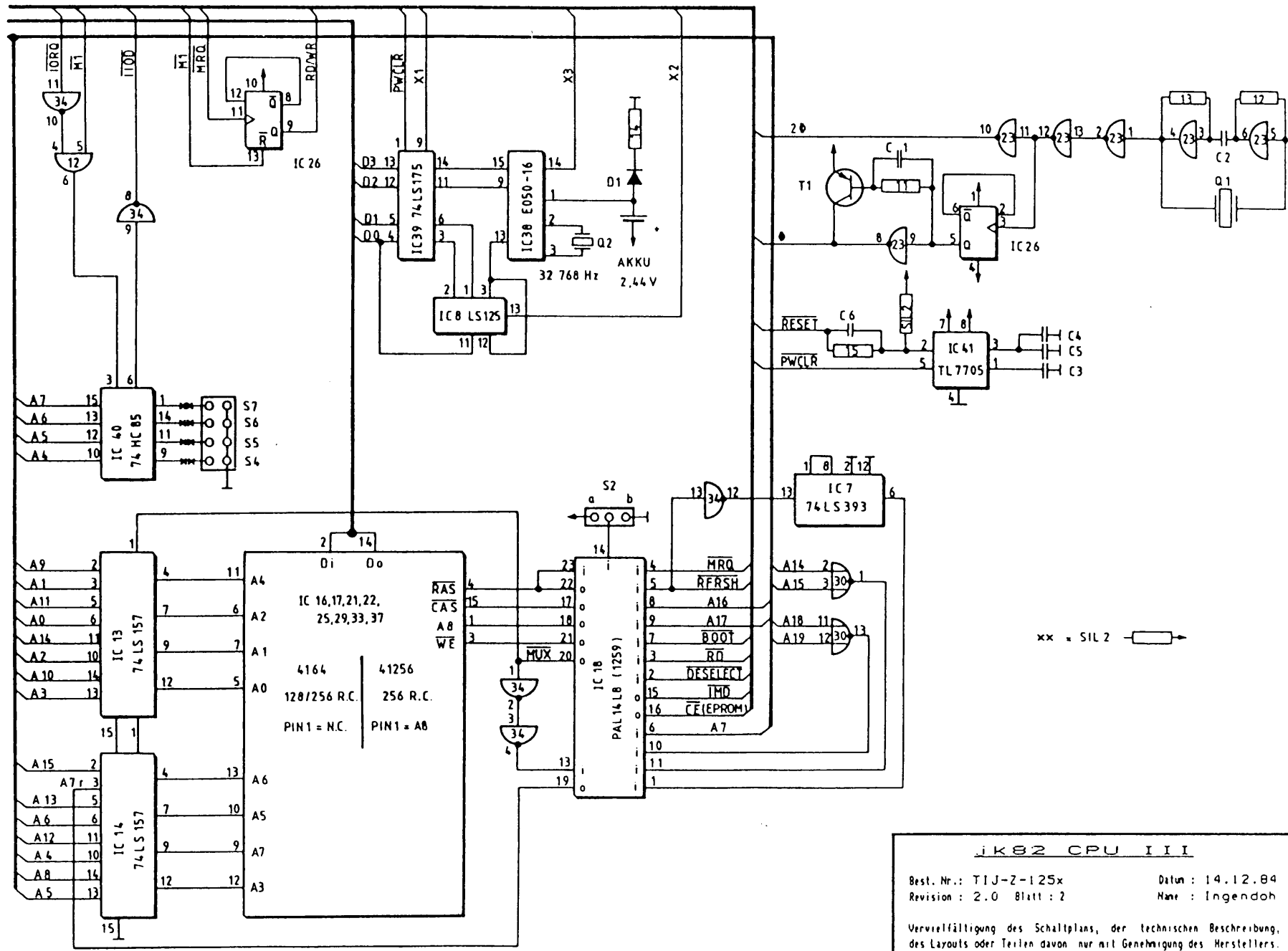


13. Schaltplan:

JK82 CPU III

Best. Nr.: TIJ-2-125x Datum: 14.12.84
 Revision: 2.0 Blatt: 1 Name: Ingendoh

Vervielfältigung des Schaltplans, der technischen Beschreibung, des Layouts oder Teilen davon nur mit Genehmigung des Herstellers.



JK82 CPU III

Best. Nr.: TIJ-2-125x Datum: 14.12.84
Revision: 2.0 Blatt: 2 Name: Ingendoh

Vervielfältigung des Schaltplans, der technischen Beschreibung,
des Layouts oder Teilen davon nur mit Genehmigung des Herstellers.