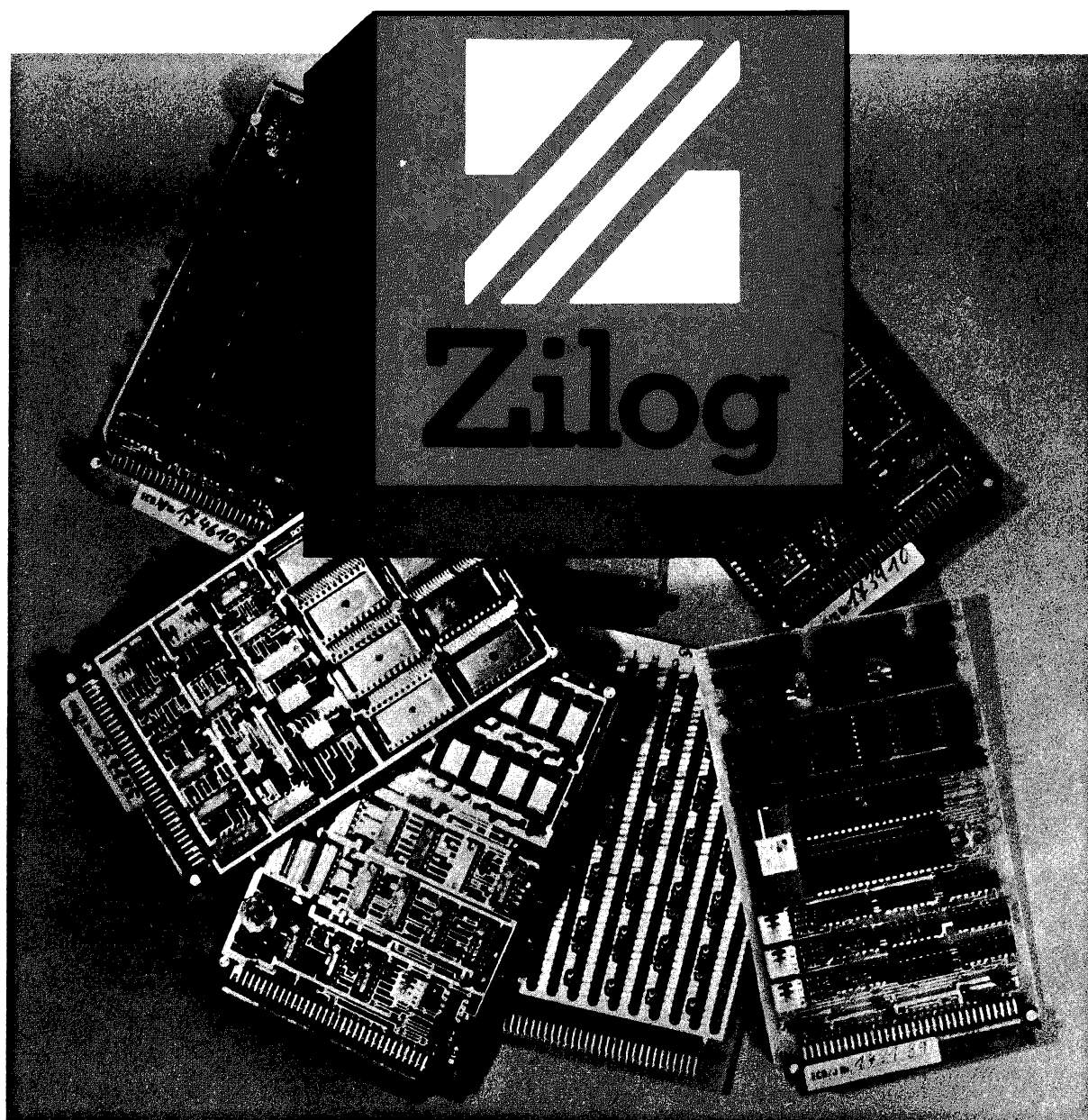


# Z80-ECB- Serie

Hardware  
Software

Anwender-Handbuch





# **Z80-ECB-Serie**

**Hardware**

**Software**

**Anwender-Handbuch**

# Z80-ECB ANWENDER-HANDBUCH

## INHALT

### I. Übersicht

1. Zilog-Baugruppen
  - 1.1. Einführung
  - 1.2. Übersicht über die Z80-Platinensysteme

### II. Hardware-Teil

1. Übersicht über das Z80-ECB-System
2. Die Zentralbaugruppe Z80-ECB/C
  - 2.1. Schaltungsbeschreibung
    - 2.1.1. CPU mit Pufferung
    - 2.1.2. Takt
    - 2.1.3. Programmspeicher
    - 2.1.4. Schreib/Lese-Speicher
    - 2.1.5. Parallel-Ein/Ausgabe
    - 2.1.6. Serien-Schnittstelle
    - 2.1.7. Hinweise zur Interrupt-Behandlung
  - 2.2. Adressenbelegung
    - 2.2.1. Speicher
    - 2.2.2. Ein/Ausgabe
  - 2.3. Pin-Belegung
    - 2.3.1. Ein/Ausgabe-Stecker der ECB/C
    - 2.3.2. Bus-Stecker
3. Die Erweiterungsbaugruppe Z80-ECB/E
  - 3.1. Schaltungsbeschreibung
    - 3.1.1. Pufferung und Decodierung
    - 3.1.2. Programmspeicher
    - 3.1.3. Schreib/Lesespeicher
4. Die Ein/Ausgabebaugruppen Z80-ECB/I
5. Schreib/Lese-Speicher-Erweiterungsbaugruppen Z80-ECB/D
6. Schreib/Lese-Speicher-Erweiterungsbaugruppen (nichtflüchtig) Z80-ECB/V
7. Ein/Ausgabe-Baugruppe für allgemeine Aufgaben und Floppy-Disk-Ansteuerungen Z80-ECB/F

8. Einschubreck mit Rückwandverdrahtung Z80-ECB/R
9. Inbetriebnahme

### III. Softwareteil

1. Monitorprogramme ECB/12 und ECB/1
  - 1.1. Einführung
  - 1.2. Hardware-Erfordernisse des Monitors
    - 1.2.1. Speicherorganisation
    - 1.2.2. Anschluß eines Bedienungs-Terminals
  - 1.3. Beschreibung des Monitor-Programms
    - 1.3.1. Verhältnis zu evtl. Anwenderprogrammen
    - 1.3.2. Formale Regeln
    - 1.3.3. Beschreibung der einzelnen Kommandos
    - 1.3.4. Haltepunkt-Behandlung
2. Monitorprogramm ECB/3
3. Echtzeitbetriebssystem Z80-MTX

Ausgabe 3, April 1978

Copyright by:

The logo for KONTRON ELEKTRONIK GMBH features a stylized 'K' symbol on the left, composed of several vertical bars of varying heights. To the right of this symbol, the word 'KONTRON' is written in a large, bold, sans-serif font. Below 'KONTRON', the words 'ELEKTRONIK GMBH' are written in a smaller, bold, sans-serif font.

GESCHÄFTSBEREICH MIKROCOMPUTER

Breslauer Str. 2  
8057 Eching  
Tel. 089/31 88-1  
Telex 0522122

# 1. Zilog-Z80-Mikrocomputer-Baugruppen

## 1. Einführung

Die weitest verbreitete Anwendung von Mikrocomputern besteht darin, Geräte, Maschinen, Prozesse usw. mit „technischer Intelligenz“ auszurüsten, d.h. in ihnen Programmierbarkeit, Entscheidungsfähigkeit, Rechner-Funktionen und Mensch/Maschine-Dialogfähigkeit zu implementieren.

Dies ist beim heutigen Stand der MOS-Technik zu Preisen möglich, die in den meisten Fällen weit unter technisch gleichwertigen Lösungen mit herkömmlichen oder kundenspezifischen Schaltkreisen liegen.

Um nun dem breiten Kreis von Anwendern diesen Schritt von konventioneller Technik zu Mikrocomputern zu erleichtern und dabei Entwicklungszeit und -kosten möglichst niedrig zu halten, wurden die Z80-Mikrocomputer-Baugruppensysteme entwickelt; in jedem Fall steht standardisierte Hardware (also Baugruppen und Baugruppenträger) zur Verfügung, die gemäß der weitverbreitetsten Normen realisiert wurde, und die zugehörige Grundsoftware (= „Monitoren“) zur Inbetriebnahme dieser Baugruppen. Daher beschränkt sich bei Verwendung der Z80-Baugruppen der gesamte Anlagen-Entwicklungsaufwand in den meisten Fällen auf die Erstellung der Anwender-Software und einer häufig trivialen peripheren Anpassungsschaltung.

Bei der Konzeption dieser Baugruppen wurde von folgenden Voraussetzungen ausgegangen:

- die Leistungsfähigkeit der Z80-Bausteine ist so hoch und die Speichersparnis gegenüber bisherigen Mikrocomputern so weitreichend, daß viele Aufgaben bereits mit einer einzigen Steckkarte zu lösen sind. Jedes der Z80-Baugruppensysteme beinhaltet deswegen einen autonomen 1 Platinen-Computer.
- da das System Z80 andererseits hervorragend zur Lösung sehr komplexer Anwenderprobleme geeignet ist, wurde diese autonome Computerkarte grundsätzlich so ausgelegt, daß ein kompromißloser Ausbau auf die volle Erweiterbarkeit der Z80 ohne spezielle Hardwaremaßnahmen möglich ist.
- die Aufteilung der einzelnen Bausteine auf die Baugruppe (= „Partitioning“) wurde dem hohen Integrationsgrad der Z80-Bausteine angemessen vorgenommen, so daß bei Verwendung dieser Standardbaugruppen praktisch in keinem Fall ungenützte Hardware übrigbleibt (= „Hardware-Overkill“).
- durch den großen Fortschritt in der Floppy-Disk-Technologie und die stark sinkenden Preise der Floppy-Disks einerseits und durch die hohe Verarbeitungsgeschwindigkeit der Z80-CPU andererseits wird es in vielen Fällen interessant und wirtschaftlich, Floppy Disks im Anwendersystem einzusetzen. Aus diesem Grund beinhalten die Z80-Baugruppensysteme komplette FD-Ansteuerungseinheiten.
- Ebenfalls mit dem Ziel, auch komplexe Anwendersysteme realisieren zu helfen, werden Baugruppen mit dynamischen RAM-Bausteinen angeboten.
- Da sämtliche Z80-Bausteine mit einer einzigen 5V-Speisepannung auskommen, wurden auch alle Baugruppen so ausgelegt, daß keine zusätzlichen Speisepannungen erforderlich sind.  
Lediglich bei der Verwendung von 2708-EPROM's geht dieser Vorteil verloren; hier müssen externe +12 V und — 5V-Versorgungsspannungen zugeführt werden.

## 2. Übersicht über die Z80-Platinensysteme

Das System Z80 umfaßt 3 verschiedene Platinensysteme, die aus den Baugruppen, Zubehör und der zugehörigen Standardsoftware bestehen.

Durch die Verfügbarkeit von Platinen unterschiedlicher Abmessungen ist gewährleistet, daß die Einsatzmöglichkeit kostensparender Z80-Standard-Hardware nicht durch mechanische Restriktionen (wie vorgegebene Einbaumaße) eingeschränkt wird:

- Das Baugruppensystem mit dem Z80-MCB als Zentralbaugruppe im amerikanischen Format 7,5'' × 7,7''.
- Das Baugruppensystem mit dem Z80-MCB/E als Zentralbaugruppe im Doppeleuropaformat (233 × 160 mm) mit 64-poligen VG-Steckern, das mit den 7,5'' × 7,7''-Platinen schaltungsmäßig identisch ist und
- das Baugruppensystem mit dem Z80-ECB/C als Zentralbaugruppe im Einzel-Europa-Format (100 × 160 mm) mit einem 64-poligen VG-Stecker.

Sämtliche Platinensysteme umfassen eine autonome voll erweiterbare 1-Platinen-Computerkarte und zusätzlich Ein/Ausgabe-, Speichererweiterungs-, dynamische RAM-, Floppy-Disk-Controller- und Stromversorgungs-Module.

Die Entscheidung, welchem der 3 Formate der Vorzug zu geben ist, hängt von der Einzelsituation des Anwenders ab: ob bei ihm bereits ein Format eingeführt ist und welches Partitioning (also welche Aufteilung der Bestückung auf die Baugruppen) für seine Aktivitäten die günstigere ist.

Die einzelnen Systeme umfassen folgende Baugruppen:

- OEM-Computer im Format 7,7'' × 7,5''
  - Ein-Platinen-Computer
  - Schreib/Lese-Speichererweiterung
  - Festwertspeicher-Erweiterung
  - Parallele Ein/Ausgabe-Baugruppe
  - Serien-Ein/Ausgaben-Baugruppe
  - Floppy-Disk-Controller
  - Video-Board
  - PROM-Programmierboard
  - AD/DA-Wandler
  - Einschub-Rahmen
- OEM-Computer im Doppel-Europa-Format
  - Ein-Platinen-Computer
  - Schreib/Lese-Speichererweiterung
  - Floppy-Disk-Controller
- OEM-Computer im Einzel-Europa-Format
  - Ein-Platinen-Computer
  - Festwertspeicher-Erweiterung
  - Dynamische Schreib/Lesespeicher-Erweiterung
  - Ein/Ausgabe-Baugruppe
  - Stromausfallsichere Schreib/Lese-Speicher-Baugruppe
  - Floppy-Disk-Controller
  - Einschub-Rahmen

An Festwertspeicher-residenter Grundsoftware wird für jedes der 3 Systeme angeboten

- 1/2 kByte-Monitor für Speicherzugriff und Datenverkehr über die Serienschnittstelle
- 1 kByte-Monitor mit zusätzlichen Testmöglichkeiten (Haltepunkten und CPU-Registrierinhalts-Ausgabe)
- 3 kByte-Monitor mit zusätzlicher Floppy-Disk-Grundsoftware

Für die ECB-Familie ist darüberhinaus ein Multi-Task-Echtzeitbetriebssystem verfügbar.

# ECB-HARDWARE

## 1. Übersicht über das ECB-System

In diesem Handbuch werden die Grundbaugruppen des ECB-Systems beschrieben:

Z80-ECB/C	Computerplatine für Stand-Alone-Betrieb, erweiterbar auf vollen Z80-Speicher und I/O-Bereich
Z80-ECB/E	Erweiterungsplatine mit max. 8 kByte Festwertspeicher und 1 kByte statischem Schreib/Lesespeicher
Z80-ECB/I	I/O-Platine mit 4 Byte-paralleler Ein/Ausgabe, Quittungs-(„Handshaking“)-Logik, 4 Zähler/Zeitgeber-Kanälen und Wire wrap Feld für anwenderspezifische Interface-schaltungen.
Z80-ECB/D8	Schreib/Lese-Speichererweiterung um 8 kByte dynamisches RAM (mit 4 kbit-Bausteinen aufgebaut)
Z80-ECB/D16	Schreib/Lese-Speichererweiterung um 16 kByte dynamisches RAM, (mit 16 kbit-Bausteinen aufgebaut) auf 32 kByte ausbaufähig.
Z80-ECB/D32	Schreib/Lese-Speichererweiterung um 32 kByte dynamisches RAM (mit 16 kbit-Bausteinen aufgebaut)
Z80-ECB/V	Schreib/Lese-Speichererweiterung um 4 kByte CMOS-RAM. Dieser Speicher ist durch einen auf der Platine untergebrachten Puffer-Akku nichtflüchtig.
Z80-ECB/F	Floppy-Disk-Controller und allgemeine Ein/Ausgabebaugruppe mit 20 Bit paralleler Ein/Ausgabe, 2 Serien-Duplex-Ein/Ausgabekanälen (SIO, HDLC- und SDLC-fähig z.B. für Rechner/Rechner-Kopplungen) und 4 Zähler/Zeitgeber-Kanälen.
Z80-ECB/R	19 Zoll-Einschubrack mit Rückwandverdrahtung für bis zu 20 ECB-Baugruppen (6 Steckplätze sind bestückt).
Z80-ECB/R6	Nachrüstatz für Z80-ECB/R zur Nachbestückung von weiteren 6 Steckplätzen des ECB/R.

Das Z80-ECB-Grundkonzept ersehen Sie aus Bild 1—1.

Sämtliche Bus-Signale wurden so auf die 64-poligen Standard VG-Stecker herausgeführt, daß alle Baugruppen untereinander über „1:1-Verdrahtung“ verbunden werden, d.h. daß immer gleiche Steckerpins (ohne Kreuzungen) miteinander verdrahtet sind, wodurch sich eine problemlose Rückwandverdrahtung ergibt.

Die Zentralbaugruppe Z80-ECB/C wurde aus folgenden Gründen bereits mit Speicher- und Ein/Ausgabemöglichkeiten ausgerüstet:

- einfachere Anwenderprobleme können bereits mit dieser Konfiguration oder mit geringfügiger Erweiterung gelöst werden.
- eine Serienschmittstelle ist in den meisten Anwendersystemen nur einmal erforderlich; sie wurde daher auf der Zentralplatine untergebracht.
- kleine Systeme kommen bereits mit den auf der Z80-ECB/C vorhandenen zwei 8bit-Ports aus, da diese bereits die gesamte Quittungs- und Interruptsteuerungs-Logik implizieren.  
In größeren Systemen werden dagegen häufig einige wenige parallele Hilfs-Ein/Ausgaben (z.B. für Anzeigen oder Status-Informationen) benötigt, für die die ECB/C-PIO verwendet werden kann.

- Das 1 kByte PROM der Z80-ECB/C wurde bewußt fest durch Programmierung eines Address Decode PROM's auf die Anfangsadresse 0 gelegt (variabel durch Austausch dieses PROM's), da man hier sinnvollerweise die Grundfirmware (= „Monitor“) oder aber Programme für Tests und Wartung bzw. zur Interrupt-Behandlung ablegt; die Platine wird mit Sockeln ohne PROM-Bausteine geliefert, da nicht festgelegt ist, ob Monitor- oder Anwenderfirmware verwendet werden soll.
- Die 256 Bytes statischen RAM's auf der Platine sind als Zwischenspeicher oder Stack einzusetzen.
- Durch Herausführen der Signale  $\overline{\text{BUSRQ}}$  und  $\overline{\text{BUSAk}}$  ist der Aufbau von Multiprozessorsystemen und direkter Speicherzugriff (DMA) möglich.
- Speichererweiterungen werden mit im Verhältnis 8:1 gemischten Festwert/Schreib/Lese-Speicher-Baugruppen durchgeführt, da erfahrungsgemäß bei wachsender Programmgröße auch der Bedarf an Variablen-Speicher proportional wächst. Zum Aufbau des Festwertspeichers können alle Standard-Masken-ROM's „fusible“ PROM's oder EPROM's eingesetzt werden, deren Pinbelegung den Bausteinen HM 7641 (fusible PROM) oder 2708 (EPROM) entspricht und die den Geschwindigkeitsanforderungen der Z80-CPU genügen. Aus diesem Grund wird die Z80-ECB/E-Baugruppe auch lediglich mit Sockeln, nicht aber mit Festwert-Speicher-Bausteinen geliefert, um dem Anwender freie Wahl der Bausteine zu ermöglichen. Beide Speicherbereiche können über DIL-Schalter jedem beliebigen adressierbaren Speicherbereich der Z80-CPU zugeordnet werden. Für Systeme, in denen auch die Anwenderprogramme in Schreib/Lesespeichern abgelegt werden sollen (z.B. bei den meisten Anwender-Systemen, die mit Floppy Disks und der Steuerplatine ECB/F arbeiten), stehen reine RAM-Karten (8, 16 oder 32 kByte dynamisches RAM ECB/D und 4 kByte-CMOS statisches RAM mit Puffer-Batterie und Speicherschutzschalter Z80-ECB/V) zur Verfügung. Zur Erweiterung der Ein/Ausgabe-Möglichkeiten der Zentralplatine dient die Z80-ECB/I mit ihren vier 8bit-Ports und vier Zähler/Zeitgeber-Kanälen inklusive kompletter Quittungs- (= „Handshaking“) und Interrupt-Steuerungslogik. Zusätzliche Ergänzung ECB-basierender Systeme ist über die Serien/Parallel/Echtzeit-, Ein/Ausgabe und Floppy-Disk-Platine ECB/F sowie über die Baugruppen aus dem Z80-KIT-Konzept und Sonderbaugruppen möglich.
- Als Bus-Stecker kommt die weitverbreitete 64-polige VG-Steckerleiste zum Einsatz.  
Hier werden sämtliche Signale herausgeführt, die allen Baugruppen gemeinsam sind und einfach durchverdrahtet werden können.
- Ein/Ausgabe-Signale werden von der Karte an dem dem Bus-Stecker entgegengesetzten Kartenende über 3M-Pfostensteckverbinder verfügbar gemacht, da
  - Ein/Ausgabe-Signale bei sehr vielen Anwendungen an der Frontseite (üblicherweise der „Rückwandverdrahtung“ gegenüberliegend) benötigt werden.
  - auch bei aufwendigerer, den Ein/Ausgabebausteinen nachgeschalteter Peripheriehardware eine Rückführung dieser Signale über eine zweite, kundenspezifische Interface-Baugruppe möglich ist (auch „Huckepack“- bzw. „Sandwich“-Verfahren erwägen!)
  - die Herausführung sämtlicher Signale, also auch der Ein/Ausgabe, auf der Busseite die Verwendung vielpoliger teurer Steckverbinder nötig machen und die komplizierte Leitungsführung auf der Baugruppe zu großem Platzverlust verursachen würde. Um dem Anwender Zeitverlust durch Aufbau- und Verdrahtungsprobleme zu ersparen, wird übrigens passend zu diesem System ein Baugruppenträger (Z80-ECB/R) angeboten.

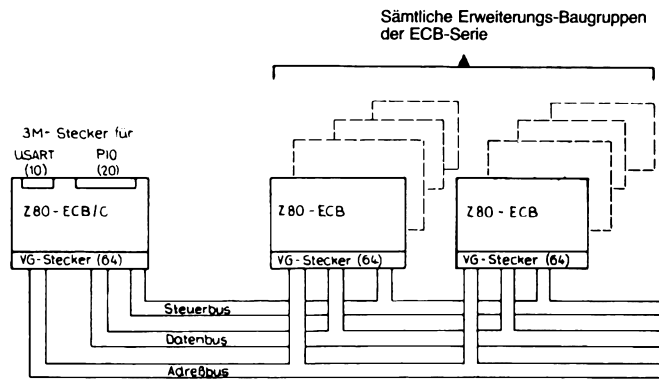


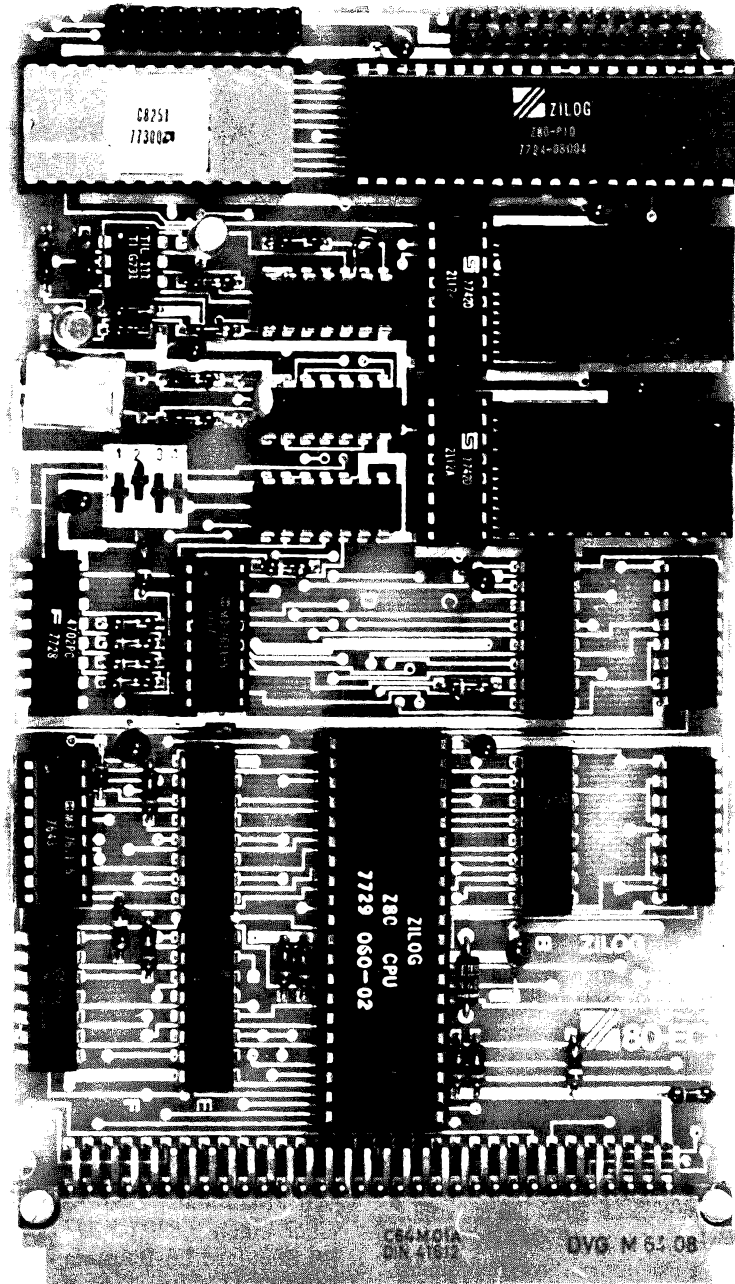
Bild 1—1: Z80-ECB-Grundkonzept.



# Z80-ECB/C



# Zentralbaugruppe



## 2. Die Zentralbaugruppe Z80-ECB/C

### 2.1 Schaltungsbeschreibung

Die Baugruppe umfaßt folgende Funktionseinheiten

- Z80-CPU
- Decoder und Puffer für vollen Speicher-(64 kByte) und I/O-Ausbau (256 Ports)

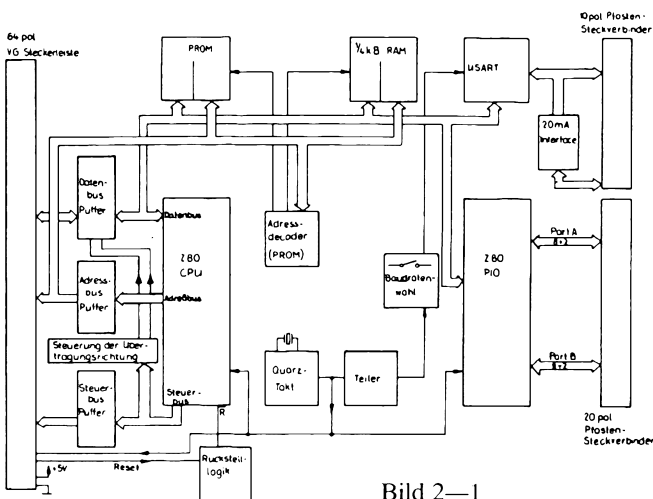


Bild 2—1

- Platz für 2 kByte PROM
- 256 Byte RAM
- 2 Parallelschnittstellen (1 Stück Z80-PIO)
- 1 Serienschnittstelle mit RS 232 und 20 mA Stromschleifen-Interfaces
- Über Schalter 14 Baud-Raten einstellbar
- Busseitiger 64-poliger Stecker nach DIN 41612 (VG 95324)
- I/O-seitig 3M-WWP-Pfostensteckverbinder

Aus der Blockschaltung (Bild 2—1) ist die Verknüpfung der einzelnen Funktionseinheiten zu erkennen; die Busleitungen sind gepuffert. Die Adreßeingänge der auf der Z80-ECB/C untergebrachten Speicherbausteine werden über diese Puffer, die Datenleitungen vom CPU-Datenbus direkt angesprochen. Festwert- und Schreib/Lese-Speicherbereich auf der Platine haben durch den Adreßdecoder vorgegebene Anfangsadressen. Die Baudrate für die Serienschnittstelle wird hardwaremäßig vom Quarz-Systemtakt abgeleitet. Bild 2—2 zeigt die Schaltung im Detail, die in folgenden noch im Einzelnen erläutert wird.

#### 2.1.1 CPU mit Pufferung

Sämtliche Z80-CPU-Signale (siehe Z80-Datenbuch und Z80-CPU-Technical-Manual) bis auf die Signale  $\Phi$ ,  $\overline{\text{BUSRQ}}$ ,  $\overline{\text{WAIT}}$ ,  $\overline{\text{INT}}$  und  $\overline{\text{NMI}}$  sind gepuffert.

Das Rückstellen der Z80-CPU kann über die zwei Standardgatter sowohl durch Aus/Einschalten der Stromversorgung als auch elektronisch oder elektromechanisch über den Eingang  $\overline{\text{RESET}}$  (1 Low-Power-Schottky-Eingangslast) erfolgen. Zur Erhöhung der Störsicherheit und Erreichung voller Speicher-Ausbaufähigkeit wurde im System Z80-ECB eine doppelte Pufferung sämtlicher Busleitungen vorgenommen. Für die bidirektionalen Datenpuffer muß je 1 Steuersignal zur Bausteinaktivierung und zur Festlegung der Datenübertragungsrichtung erzeugt werden (s. Z80-Applikationsschrift über Ein/Ausgabeschaltungen und Pufferung in Z80-Systemen). Diese Steuersignale hängen ihrerseits von folgenden Signalen ab:

- a)  $\overline{\text{INTERNAL MEM DECODE}} (\overline{\text{IMD}})$   
ist aktiv, wenn der auf der Zentralplatine aufgebaute Speicherbereich angesprochen ist.

- b)  $\text{I/O-REQUEST} (\overline{\text{IORQ}})$   
ist aktiv, wenn eine Daten-Ein-Ausgabe eingeleitet wird.
- c)  $\text{READ: Lesesignal} (\overline{\text{RD}})$
- d)  $\text{M1:}$  definiert den Maschinenzklus 1 (= „Fetch-Zyklus“)
- e)  $\text{INTERNAL I/O-DECODE} (\overline{\text{IIODEC}})$   
ist aktiv, sobald Ein/Ausgabebausteine auf der Zentralplatine ECB/C angesprochen sind.
- f)  $\text{INTERNAL INTERRUPT ENABLE IN} (\text{II EI})$   
Interrupt-Freigabe-Eingang für Ein/Ausgabe-Bausteine auf der ECB/C
- g)  $\text{INTERNAL INTERRUPT ENABLE OUT} (\text{II EO})$   
Interrupt-Freigabe-Ausgang von Ein/Ausgabe-Bausteinen auf der ECB/C

Die genannten Signale verknüpft man nun zur Erzeugung der beiden Datenbus-Steuersignale sinnvollerweise mit einem PROM (vgl. Bild 2—3).

Die logischen Gleichungen hierzu lauten wie folgt:

$$\text{CPUIN} = \underbrace{(\overline{\text{IMD}} \wedge \overline{\text{RD}})}_{\text{EXT MEM READ}} \vee \underbrace{(\overline{\text{IORQ}} \wedge \overline{\text{M1}})}_{\text{INTA}} \vee \underbrace{(\overline{\text{IORQ}} \wedge \overline{\text{IIODEC}} \wedge \overline{\text{RD}})}_{\text{EXT IO READ}}$$

$$\overline{\text{CS}} = \underbrace{(\overline{\text{IORQ}} \wedge \overline{\text{M1}} \wedge \text{II EI} \wedge \text{II EO})}_{\text{EXT INTA}} \vee \underbrace{(\overline{\text{IORQ}} \vee \text{II ODEC})}_{\text{EXT MEM TRANS}}$$

Dabei sind:

- $\text{CPUIN:}$  Richtungssteuersignal (High: Richtung zur CPU)
- $\overline{\text{CS:}}$  Aktivierungssignal (High: Puffer gesperrt)
- $\overline{\text{M1:}}$  Maschinenzklus 1
- $\overline{\text{IORQ:}}$  I/O Request
- $\overline{\text{IIODEC:}}$  ECB/C Internal I/O Anforderung
- $\overline{\text{RD:}}$  Read
- $\text{II EO:}$  ECB/C- Internal Interrupt Enable Out
- $\text{II EI:}$  ECB/C Internal Interrupt Enable In
- $\overline{\text{IMD:}}$  ECB/C- Intern Speicheranforderung

Dadurch wird der 1 KByte-Festwertspeicher auf der Z80-ECB/C in den Adreßbereich

0 0 0 0 H bis 0 3 F F H

und der 1/4 kByte Schreib/Lesespeicher in den Adreßbereich

3 C 0 0 H bis 3 C F F H

gelegt.

#### 2.1.2 Takt

Aus Stabilitäts- und Kostengründen wird der Systemtakt über einen 9,8304 MHz-Quarz erzeugt, der zusammen mit 1/3 Standard LS-Baustein einen Oszillator bildet. Der für die Z80-CPU nötige Takt  $\Phi$  von 2,4576 MHz wird über einen D-FlipFlop-Teiler :4 ( $2 \times 1/2 74\text{LS}74$ ) gewonnen, von dem auch das  $2\Phi$ -Signal abgeleitet und auf den Bus herausgeführt wird, das häufig in dynamischen RAM-Speichersystemen benötigt wird (siehe entsprechende Applikationsschriften!). Die für die Serienschnittstelle erforderliche Bezugsfrequenz erzeugt der den D-Flip-Flops nachgeschaltete programmierbare Teilerbaustein 34702, an dessen Eingängen So...S3 4 DIL-Schalter zur Festlegung der gewünschten Datenübertragungsrate angeschlossen sind.

Inhalt der PROM's (Datenbussteuerung und Adreßcodierung)

- a) PROM HM 7611 zur Datenbussteuerung  
 D 0 = CS active low für ECB/C  
 D 1 = CPU IN active high für ECB/C  
 D 2 = IOSEL active low für ECB/I  
 D S = IO IN active high für ECB/I

Adresse (hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
001	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
002	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
003	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
004	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
005	F	F	F	F	F	F	F	F	F	F	F	F	F	F	3	3
006	3	3	F	F	F	F	F	F	F	F	F	F	F	F	F	F
007	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
008	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
009	F	F	F	F	3	3	3	3	F	F	F	F	E	E	E	E
00A	E	F	E	E	E	3	E	E	E	F	E	E	E	3	E	E
00B	F	F	F	F	B	B	B	B	F	F	F	F	C	C	C	C
00C	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
00D	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
00E	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
00F	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C

- b) PROM HM 7611 zur Adreßdecodierung (standardmäßig ausgelieferter PROM-Inhalt; bei anderer Adreßbelegung ist ein PROM mit anderem Inhalt zu verwenden).

Adresse (hex)	Inhalt
0	C
1	C
2	A
3	A
3C	6

- c) PROM HM 7611 zur Adreßdecodierung bei Einsatz von HM 7681- oder i 2758 Programmspeicherbausteinen:

Adresse (hex)	Inhalt
0	C
1	C
2	C
3	C
4	A
5	A
6	A
7	A
3C	6

Alle anderen Steuerpromspeicherzellen müssen mit 1111 (FHex) belegt sein!

Dies entspricht einer Speicherorganisation:

0 — 7FFH PROM  
 3C00H — 3CFFH RAM

Die folgende Tabelle zeigt die Beziehung zwischen Schalterstellungen und gewählter Baud-Rate:

S1	S2	S3	S4	
EIN	EIN	AUS	EIN	50 Baud
EIN	EIN	AUS	AUS	75 Baud
EIN	AUS	EIN	EIN	134,5 Baud
EIN	AUS	EIN	AUS	200 Baud
EIN	AUS	AUS	EIN	600 Baud
EIN	AUS	AUS	AUS	2400 Baud
AUS	EIN	EIN	EIN	9600 Baud
AUS	EIN	EIN	AUS	4800 Baud
AUS	EIN	AUS	EIN	1800 Baud
AUS	EIN	AUS	AUS	1200 Baud
AUS	AUS	EIN	EIN	2400 Baud
AUS	AUS	EIN	AUS	300 Baud
AUS	AUS	AUS	EIN	150 Baud
AUS	AUS	AUS	AUS	110 Baud

### 2.1.3 Programmspeicher

Der 1 kByte große Festwertspeicher besteht aus 2 „fusible“ PROM's HARRIS HM 7641, die 512 × 8 bit organisiert sind; bei Verwendung der 1k × 8-Bipolaren HARRIS-PROM's HM 7681 ergibt sich ein Programmspeicherbereich von 2k Byte. Die Adreßleitung A9 ist in diesem Fall zusätzlich mit Pin 22 beider Fassungen zu verbinden und das Adreß-decode-PROM entsprechend zu modifizieren. Stattdessen sind natürlich auch elektrisch und Pin-kompatible Bausteine anderer Hersteller einsetzbar.

Dieser PROM-Bereich ist in erster Linie zur Unterbringung eines Monitors oder von Wartungs- und Test-Programmen gedacht.

Die Bausteinanwahl (= „Chip-Select“) findet durch die höherwertigen 8 Adreßbits statt. Zur Decodierung wurde ein PROM-Baustein HM 7611 verwendet. Dadurch wurde

- die Anzahl der nötigen Decoder-Bausteine minimiert und gleichzeitig
- dem Anwender die Möglichkeit gegeben, durch Auswechseln des PROM's die Speicheranfangsadressen in ¼ kByte-Schritten in jeden beliebigen Speicherbereich zu legen (s. auch Pkt. 2.2. „Adressen-Belegung“).

### 2.1.4 Schreib/Lese-Speicher

Der 256 Byte große Schreib/Lese-Speicher besteht aus 2 statischen 2112A-RAM-Bausteinen, die 256 × 4-bitweise organisiert sind.

Sie werden vom gleichen PROM HM 7611 wie der vorher erwähnte Festwertspeicher-Bereich aktiviert und lassen sich durch Auswechseln dieses PROM's in 256-Byte-Schritten in jeden beliebigen Speicherbereich verlagern.

### 2.1.5 Parallel-Ein/Ausgabe

Für den Datenverkehr zwischen Z80-CPU und der „Außenwelt“ über parallele Schnittstellen wurde 1 Baustein Z80-PIO auf der Baugruppe Z80-ECB/C untergebracht. Dieser Baustein umfaßt zwei 8bit-Ports, die als Eingabe-Port, Ausgabe-Port, bidirektionales Port oder aber in Einzelbit-Schaltung arbeiten können (Einzelheiten siehe Z80-PIO-Produktspezifikation und Z80-PIO-Technical Manual).

Beide Ports verfügen über eine interne Logik, die ohne zusätzliche Hardware Quittungs-(= „Handshaking“)-Betrieb (2 eigene Leitungen pro Port) und priorisierten Vektor-Interrupt ermöglicht. Die die Priorisierung festlegenden Signale IEI (= „Interrupt Enable In“) und IEO (= „Interrupt Enable Out“) sind zur Verkettung mit weiteren Ein/Ausgabebausteinen busseitig herausgeführt.

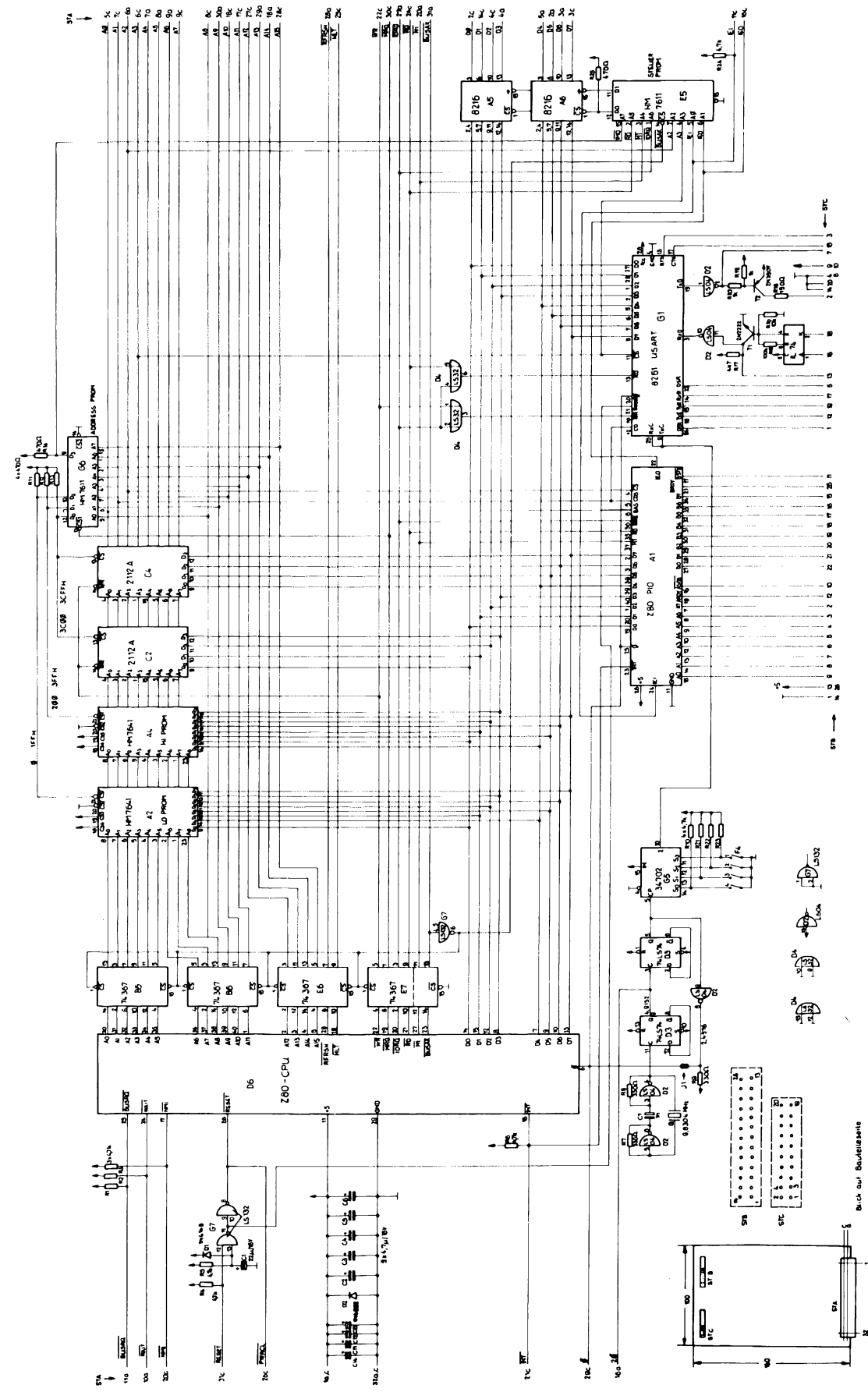


Bild 2-2

### 2.1.6 Serien-Schnittstelle

Die Serienschchnittstelle (1 Duplex-Kanal) ermöglicht den seriellen Datenverkehr zwischen peripherer Elektronik und der Z80-CPU; meist wird man hier Diagnosegeräte für Wartungs- und Testzwecke anschließen.

Bei Verwendung eines Fernschreibers kann die ECB/C zusammen mit den ECB-Monitorprogrammen sogar als einfaches Entwicklungshilfsmittel auf hexadezimaler Basis benutzt werden; es ist mit dieser Konfiguration das Einschreiben von Programmen, Lesen von Speicher- und CPU-Register-Inhalten und sogar die Durchführung von Tests über das Setzen von Haltepunkten (= "Breakpoints") möglich.

Die Serienschchnittstelle wurde mit einem Standard-USART-Baustein realisiert.

Der USART ist übrigens selbst auch interruptfähig, hat aber nicht die Möglichkeit zur automatischen Interrupt-Vektor-Behandlung, die alle Zilog Z80-Ein/Ausgabe-Bausteine aufweisen. Soll die Serienschchnittstelle in Vektor-Interrupt-Betriebsart gefahren werden, müßte deshalb eine zusätzliche Logik zur Interrupt-Vektor-Erzeugung aufgebaut werden.

Die Eingangsschaltung der Stromschleifenschnittstelle (TTY-Schnittstelle) besteht aus einem Optokoppler mit nachfolgendem Impedanzwandler und TTL-Puffer. Die Leuchtdiode des Optokopplers kann in zwei verschiedene Arten vom angeschlossenen Terminal angesteuert werden:

- Liefert das Terminal den Strom für die Stromschleifenschnittstelle, genügt es, die Leuchtdiode in diesen Stromkreis zu schalten.
- Stellt das Terminal nur einen Kontaktschluß zur Verfügung, wie es üblicherweise bei einem Fernschreiber (z. B. Teletype ASR 33) der Fall ist, ist es notwendig, die Anode der Leuchtdiode über einen Widerstand (150 Ohm) auf +5 Volt zu legen und den Kontakt zwischen Leuchtdioden-Kathode und Masse zu legen.

Der Fototransistor auf der Ausgangsseite des Optokopplers ist als Emitterfolger geschaltet, der einen weiteren Transistor schaltet. Die Basis des Fototransistors ist hochohmig mit dem Emitter verbunden, so daß der Fototransistor beim Ein- und Ausschalten an seinen Anschlüssen keine große Spannungsdifferenzen auszugleichen hat. Dadurch wird der Einfluß interner Transistorkapazitäten ausgeglichen und eine stabile Datenübertragung selbst bei hohen Baudraten gewährleistet.

Der dem Treibertransistor folgende TTL-Puffer 74LS04 ist zusätzlich auf den Pfostensteckverbinder der Serienschchnittstelle zur Realisierung einer Spannungsschnittstelle entsprechend RS 232 herausgeführt, wobei allerdings die Normpegel nicht im vollen Umfang eingehalten werden. Dies hat jedoch für Datenübertragungen auf Strecken < 20 m praktisch keine Bedeutung.

**Zur besonderen Beachtung:** Bei Anschluß eines RS 232/V 24 SENDERS mit vollem Spannungshub an die Serienschchnittstelle der ECB/C muß eine Zener Diode 4.7V als Schutz parallel zum Eingang geschaltet werden.

Analog ist auch der Senderausgang des USART 8251 über einen TTL-Puffer und einen PNP-Transistor geschaltet, wobei der Ausgang des TTL-Puffers wieder über eine separate Leitung als RS 232 ähnliche Schnittstelle auf den Serieninterface-Steckverbinder herausgeführt ist. Der PNP-Transistor prägt den Strom der Datenübertragungs-Stromschleife.

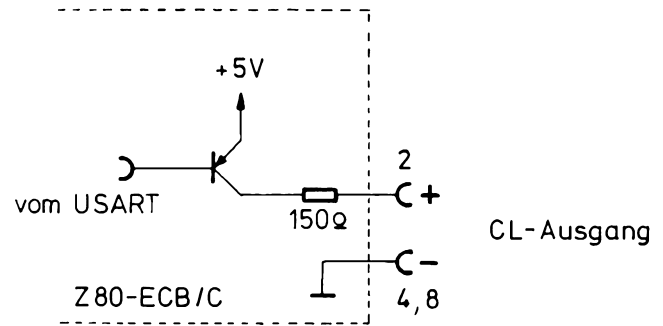
Die restlichen Ausgänge des 8251 sind ungepuffert auf den Serienstecker herausgeführt.

Der Clear to Send Eingang des USART verhindert das Aus-senden von Daten, wenn er auf HIGH steht.

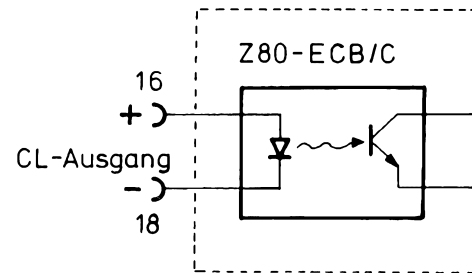
Da dieser Eingang auf der ECB/C nicht abgeschlossen ist, empfiehlt es sich, auf dem Anschlußstecker eine Brücke zwischen CTS (PIN 15 Pfostenstecker) und GND (PIN 14 Pfostenstecker) zu legen.

#### 2.1.6.1 Current Loop 20 mA (CL)

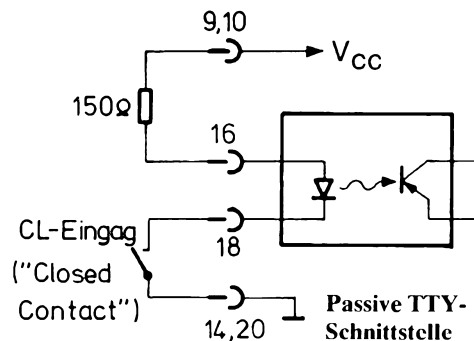
Der CL-Ausgang besteht aus einem PNP-Transistor, der über einen 150Ω-Widerstand den notwendigen Strom liefert.



Der CL-Eingang besteht aus einem Optokoppler mit nachgeschaltetem Verstärker.



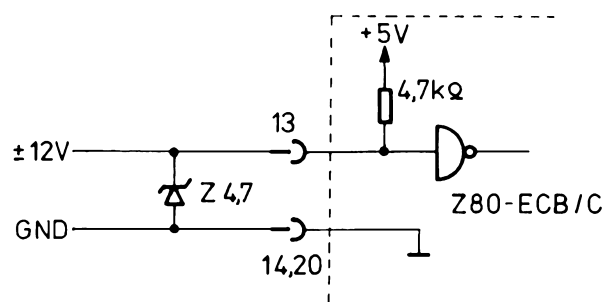
Dieser Eingang ist dafür eingerichtet, daß der Sender den Strom für die Schnittstelle liefert. Ist dies nicht möglich, muß im Verbindungsstecker ein Widerstand eingebaut werden:



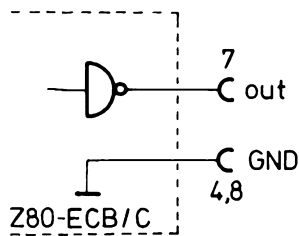
Der CL-Sender liefert dann lediglich ein „closed contact“-Signal, wobei der Strom für die Leuchtdiode über den 150Ω-Widerstand aus den +5V der Zentraleinheit erzeugt wird.

#### 2.1.6.2 RS 232 C/V 24 Schnittstelle

Der RS 232 C Eingang der SST ist ein Low Power Schottky TTL Eingang. Um diesen mit echten RS 232 C Signalen steuern zu können, ist im allgemeinen nur eine Zenerdiode im Verbindungsstecker notwendig.



Der RS 232C Ausgang ist ein Ausgang eines Low Power Schottky Gatters. Dieses liefert zwar die richtigen logischen Signale, nicht jedoch die Normpegel der RS 232C Norm.



Hat ein anzuschließender RS 232C Leitungsempfänger eine Hysterese  $\geq 2V$ , ist ein Interface zwischenzuschalten.

### 2.1.7 Hinweise zur Interrupt-Behandlung

Bei Mehrbaugruppen-Systemen, die unter Interrupt arbeiten, existiert grundsätzlich das Problem, die RETI-Anweisung über die Datenbuspuffer zu bringen.

Da einerseits Interrupt auf der Zentralplatine möglich sein muß, andererseits das Anwenderprogramm vom Entwicklungssystem auf beliebige (also auch außerhalb des 1kByte-Bereichs der ECB/C) Speicherbereiche gelegt werden kann, existieren zur Testbarkeit der Anwender-Programme über den Echtzeit-Testadapter des Z80-Entwicklungssystems nur folgende mögliche Programmstrukturen:

- aa) Im gesamten Programm kommt nur ein einziger RETI-Befehl vor, der im Speicherbereich 0 ... 1 kByte stehen muß (adressiert über eine Marke IRET).
- ab) Der Rücksprung aus weiteren Interrupt-Bedienroutinen erfolgt dann über den Befehl JP IRET.

oder

- bb) alle Interrupt-Service-Routinen liegen im RAM/ROM-Speicherbereich der ECB/C-Baugruppe.

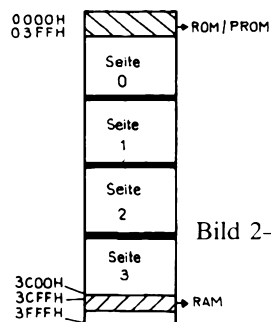
Wird in der Testphase mit dem Z80-Entwicklungssystem-internen Speicher gearbeitet, so ist außerdem sicher zu stellen, daß **keine** Prom's in den Promsockeln der ECB/C stecken. Ist dies nicht der Fall, kommt es bei jeder Ausführung eines RETI-Befehls zu einem Buskonflikt, der in der Regel verhindert, daß die I/O-Bausteine den RETI-Befehl richtig decodieren.

## 2.2 Adressen-Belegung des Z80-ECB/C

### 2.2.1 Speicher

Der Speicher ist in 4 kByte-Seiten eingeteilt, bei Verwendung des 1/4 kByte-Speichers auf der ECB/C-Platine bleibt der Rest der Seite, in der das RAM liegt, unbenutzt.

Die ausgelieferte Standardversion hat folgende Zuweisung (Bild 2—4):



### ROM/PROM

Adresse:	0	0	0	0	} 1 kByte
	.	.	.	.	
	.	.	.	.	
	0	3	F	F	

### RAM

	3	C	0	0	} 1/4 kByte
	.	.	.	.	
	.	.	.	.	
	3	C	F	F	

Der RAM-Bereich wurde an das obere Ende der Seite 3 gelegt, da ECB-Systeme dann problemlos mit der Minimal-Ausrüstung des Z80-Entwicklungssystems hardware- und software-mäßig auszutesten sind.

### 2.2.2 Ein/Ausgabe

Die Ein/Ausgabe-Bausteine auf dem Z80-ECB/C sind in 1 aus 8 Code der Speicheradress-Bits 0—3 dekodiert.

Z80-PIO:

PORT B/A	A0 LOW = A; HI = B
CONTROL DATA	A1 LOW = DATA; HI = CONTROL
CHIP SEL	A2 LOW = CHIP SELECTED

USART:

CONTROL/DATA	A1 LOW = DATA; HI = CONTROL
CHIP SEL	A3 LOW = CHIP SELECTED

Es ist zu beachten, daß bei Erweiterung um weitere I/O-Bausteine die Adreß-Bits 2 und 3 immer HI sind!

Die Adressen für die Ein/Ausgabe-Bausteine ergeben sich damit wie folgt:

PIO:	PORT B:	DATA 09H
	PORT B:	CONTROL 0BH
	PORT A:	DATA 08H
	PORT A:	CONTROL 0AH

USART:	CONTROL:	06H
	DATA:	04H

Grundsätzlich nicht erlaubt sind die I/O Adressen X0 ... X3H, da dann sowohl USART als auch PIO ausgewählt sind (X = 0 ... F).

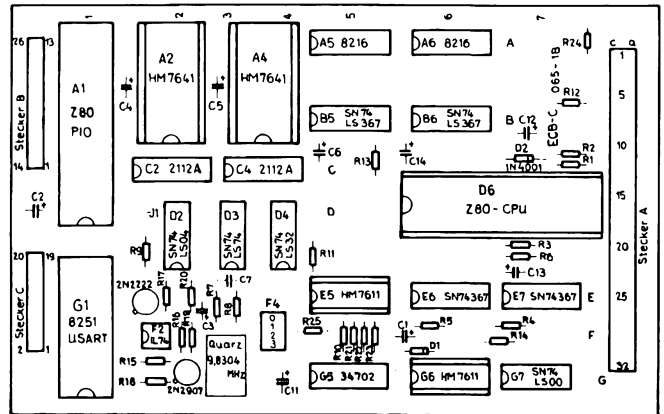
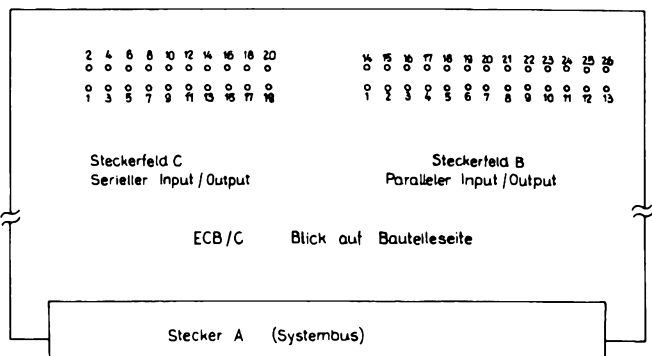


Bild 2—5

## 2.3 Pinbelegungen

### 2.3.1 Ein/Ausgabe-Stecker des ECB/C



Stift-Nr. Steckerfeld C 20 pol. WWP Steckerfeld

1	$\overline{DTR}$ (Data Terminal Ready; 8251)
2	Current loop Out +
3	$\overline{RTS}$ (Request to send Data; 8251)
4	Current loop Out — (GND)
5	$\overline{DSR}$ (Data Set Ready; 8251)
6	NC
7	RS 232 Out +
8	GND
9	+5 V
10	+5 V
11	NC
12	Tx EMPTY (Transmitter Empty; 8251)
13	RS 232 IN +
14	GND
15	$\overline{CTS}$ (Clear to send Data; 8251)
16	Current loop in +
17	RxRDY (Receiver Ready; 8251)

Stift-Nr. Steckerfeld C 20 pol. WWP Steckerfeld  
 18 Current loop in —  
 19 Tx Ready (Transmitter Ready; 8251)  
 20 GND

passendes Gegenstück: 20 pol. Pfosten Verbinder  
 z. B. Scotch 3421—0000

Stift-Nr. Steckerfeld B 26 pol. WWP-Steckerfeld

1 GND  
 2 A7  
 3 A6  
 4 A5  
 5 A4  
 6 A3  
 7 A2  
 8 A1  
 9 A0

} PIO PORTA

10  $\overline{ASTB}$  A Strobe  
 11  $\overline{BSTB}$  B Strobe  
 12 ARDY A Ready  
 13 +5 V  
 14 GND  
 15 B7  
 16 B6  
 17 B5  
 18 B4  
 19 B3  
 20 B2  
 21 B1  
 22 B0  
 23 NC  
 24 NC  
 25 BRDY B Ready  
 26 +5 V

} PIO PORT B

Gegenstück: 26 pol. Pfosten-Verbinder  
 z. B. Scotch 3399—0000

### 2.3.2 Der ECB-Bus.

Die folgende Anschlußbelegung ist für alle Baugruppen der ECB-Serien verschiedener Hersteller genormt. Die einzelnen Platinen sind durch 64polige Steckverbinder (VG 95324) über 1:1 Verdrahtung miteinander verbunden. Die Angaben über Fan In und Fan Out beziehen sich auf die Zentralplatine ECB/C.

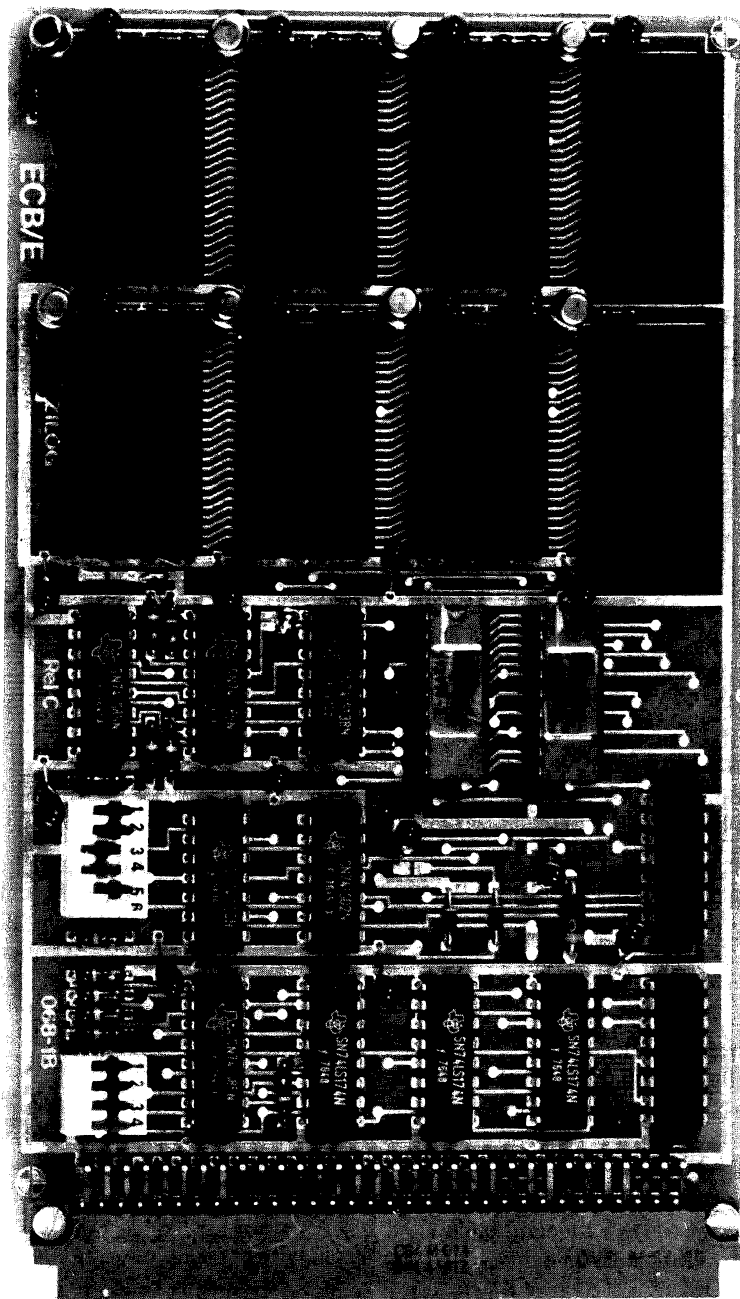
Benennung	Stecker Pin	Bezeichnung	Fan Out (TTL)	Fan In	Kommentar
A 0	5c	Adresse 0	9		Adress-Bus (Tri-State bei BUSAK)
A 1	7c	Adresse 1	9		
A 2	6a	Adresse 2	9		
A 3	6c	Adresse 3	9		
A 4	7a	Adresse 4	9		
A 5	8a	Adresse 5	9		
A 6	9a	Adresse 6	9		
A 7	9c	Adresse 7	9		
A 8	8c	Adresse 8	9		
A 9	30a	Adresse 9	9		
A 10	18c	Adresse 10	9		
A 11	17c	Adresse 11	9		
A 12	27c	Adresse 12	9		
A 13	29a	Adresse 13	9		
A 14	18a	Adresse 14	9		
A 15	28c	Adresse 15	9		
D 0	2c	Data 0	25 mA low	0,25 mA Low	Daten Bus Bidirek- tional; (Tri-State bei BUSAK)
D 1	14c	Data 1	25 mA low	0,25 mA Low	
D 2	4c	Data 2	25 mA low	0,25 mA Low	
D 3	4a	Data 3	25 mA low	0,25 mA Low	
D 4	5a	Data 4	25 mA low	0,25 mA Low	
D 5	2a	Data 5	25 mA low	0,25 mA Low	
D 6	3a	Data 6	25 mA low	0,25 mA Low	
D 7	3c	Data 7	25 mA low	0,25 mA Low	
D 8	10c	Data 8			
D 9	12c	Data 9			
D 10	13c	Data 10			
D 11	14a	Data 11			
D 12	23c	Data 12			
D 13	19c	Data 13			
D 14	21a	Data 14			
D 15	22a	Data 15			
$\overline{M1}$	20a	Maschinenzyklus 1	9		System Steuerbus Ausgänge
$\overline{MRQ}$	30c	Memory Request	9		
$\overline{IORQ}$	27a	IN/OUT Request	9		
$\overline{RD}$	24c	Read	9		
$\overline{WR}$	22c	Write	9		
$\overline{RFRSH}$	28a	Refresh	9		
$\overline{HLT}$	25c	Halt	9		
$\overline{WAIT}$	10a	Wait		1 mA Low 4,7 kPull up	
$\overline{INT}$	21c	Interrupt		1 mA Low 4,7 kPull up	
$\overline{NMI}$	20c	non Mask. Int.		1 mA Low 4,7 kPull up	
$\overline{RESET}$	31c	Reset		1,4 mA Low 4,7 kPull up	
IEI 1	11c	Int. enable in		1 mA Low, 4,7 kPull up	Daisy Chain- Interrupt- Steuerung
IEO 1	16c	Int. enable out	1		
$\overline{PWRCL}$	26c	Power on clear	9		Ausgänge
$\Phi$	29c	Clock 2,45 MHz	1 (330 $\Omega$ Pull up)		
2 $\Phi$	16a	2 $\times$ Clock			
n $\Phi$	25a	n $\times$ Clock			
$\overline{BUSRQ}$	11a	Busrequest		1 mA Low 4,7 kPull up	Bus-Steuerung für DMA- basierende Systeme
BUSAK	31a	Busacknowledge	9		
BAI 1	12a	Busprioritätssteuerung Ein			
BAO 1	17a	Busprioritätssteuerung Aus			
$\overline{WRITE EN}$	26a	Write Enable			Schreibfreigabe für ECB/D Synchronisation für ECB/V
$\overline{DPR}$	23a				
+ 5	1a, c	+ 5 V			Spannungsversor- gungsleitungen
GND	32a, c	Ground			
+ 12	13a	+ 12 V } für EPROMS			
- 5	15a	- 5 V }			
+ 15	19a	+ 15 V } für V24 und			
- 15	15c	- 15 V } AD-Wandler			
VC MOS	24a	+ 5 V Batterie-Spannung (Notstrom)			



# Z80-ECB/E



## Speicher- erweiterungsbaugruppe



# 3. Die Erweiterungsbaugruppe Z80-ECB/E

## 3.1 Schaltungsbeschreibung

Die Baugruppe umfasst folgende Funktionseinheiten:

- 1...8 kByte Festwertspeicher (ROM, PROM oder EPROM)
- 1 kByte statisches RAM
- Decoder
- Schalter zur Speicherbereichsfestlegung
- Busseitiger 64-poliger Stecker nach DIN 41612 (VG 95324)

Bild 3—1 ist das zugehörige Blockschaltbild. Man sieht, daß auch hier die Systembusse gepuffert sind; dadurch ergibt sich zusammen mit den Puffern auf der Zentralplatine Z80-ECB/C eine zweifache Pufferung, was volle Erweiterbarkeit und erhöhte Störsicherheit garantiert.

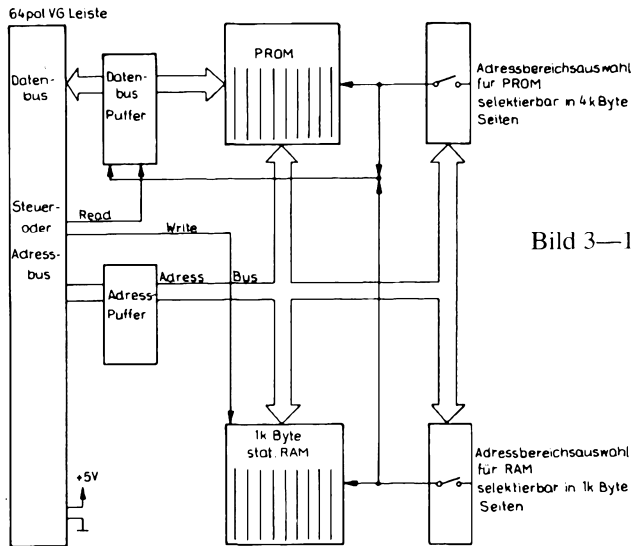


Bild 3—1

Bild 3—2 zeigt die Gesamtschaltung der Baugruppe, die in folgendem näher erläutert wird.

### 3.1.1 Pufferung und Decodierung

Die Übertragungsrichtung der bidirektionalen Datenbus-Puffer wird durch das RD-Signal bestimmt.

Die logischen Verknüpfungen sind wie folgt:

$$\overline{CS} = \overline{MEMSEL} \wedge MRQ$$

$$\overline{RD} = \overline{RD}\text{-Signal der Z80-CPU}$$

Dabei sind:

- $\overline{RD}$  = Richtungssignal (High = zur betreffenden Karte hin)
- $\overline{CS}$  = Aktivierung (High = Puffer gesperrt)
- $\overline{MEMSEL}$  = Anforderung für Speicher auf ECB/E

### 3.1.2 Programmspeicher

Die Baugruppe enthält 8 24-Pin-Sockel, in die wahlweise folgende Festwertspeicherbausteine (bzw. hierzu pinkompatible Bausteine) eingesteckt werden können  
 HARRIS HM 7641 fusible PROM  $\frac{1}{2} k \times 8$  (=  $\frac{1}{2}$  KByte)  
 HARRIS HM 7608 oder 7681 fusible PROM  $1 k \times 8$  (= 1 KByte)  
 2708/8708/2758 EPROM  $1 k \times 8$  (= 1 KByte)  
 Bausteine 2716 können durch Eingriffe in die Verdrahtung eingesetzt werden.

Gelieferte Verdrahtung ist für HM 7641. Sollen HM 7608 oder 2708/8708-Bausteine eingesetzt werden, sind die Drahtbrücken (= "Jumpers") entsprechend Tab. 1 zu verdrahten.

Für 2708 EPROM's müssen zusätzliche externe -5 und +12 V-Versorgungsspannungen am Busstecker angelegt werden. Der auf der Platine befindliche 4-fach-Dil-Schalter dient zur Festlegung der Anfangsadresse des Festwertspeicherbereichs. Bitte beachten Sie, daß die Festwertspeicherkapazität bei Verwendung von HM 7641 4 KByte, bei Verwendung von HM 7608 oder 2708/2758 8 KByte beträgt.

Bestückung und Verdrahtung ist in diesem Fall entsprechend Tabelle 1 zu ändern.

Da die Worst-Case Zugriffszeit der EPROM's ein Arbeiten mit maximaler Z80-CPU-Taktfrequenz nicht erlaubt, können nur selektierte ( $t_c < 380$  nsec.) 2708-Bausteine verwendet werden, oder aber die CPU-Taktfrequenz ist entsprechend zu erniedrigen. Das Einfügen eines Wait-Zyklus ist über einen hardwaremäßigen Zusatz zur Baugruppe möglich. Die entsprechende Schaltung findet man im CPU Manual (Adding a Wait State to any M 1 Cycle). Sie besteht aus lediglich einem D-Flipflop-Baustein. Eine entsprechende Schaltung ist auf der Rückwandverdrahtung der ECB/R vorgesehen. Ein Umstellen der Verdrahtung zum Einsatz von 16 kbit (= 2 kByte) EPROM's (INTEL 2716) ist folgendermaßen möglich:

- a) Änderung am Chipauswahlbaustein (LS 138 IC Nr. D3):
  - Jumper 3 und 4 öffnen
  - IC D3 Pin 1 auf Pin 12 von E3
  - IC D3 Pin 2 auftrennen und auf Pin 13 von IC E3
  - IC D3 Pin 3 auftrennen und auf Pin 15 von IC E3
- b) Änderung des PROM-Bereich-Decoders (LS 85 IC Nr. E5)
  - Jumper 5 öffnen
  - Pin 12 von IC E5 öffnen und mit Pin 10 vom gleichen IC verbinden.
  - Schalterblock F5 (Prom Auswahl) Schalter 3 und 4 immer AUS.

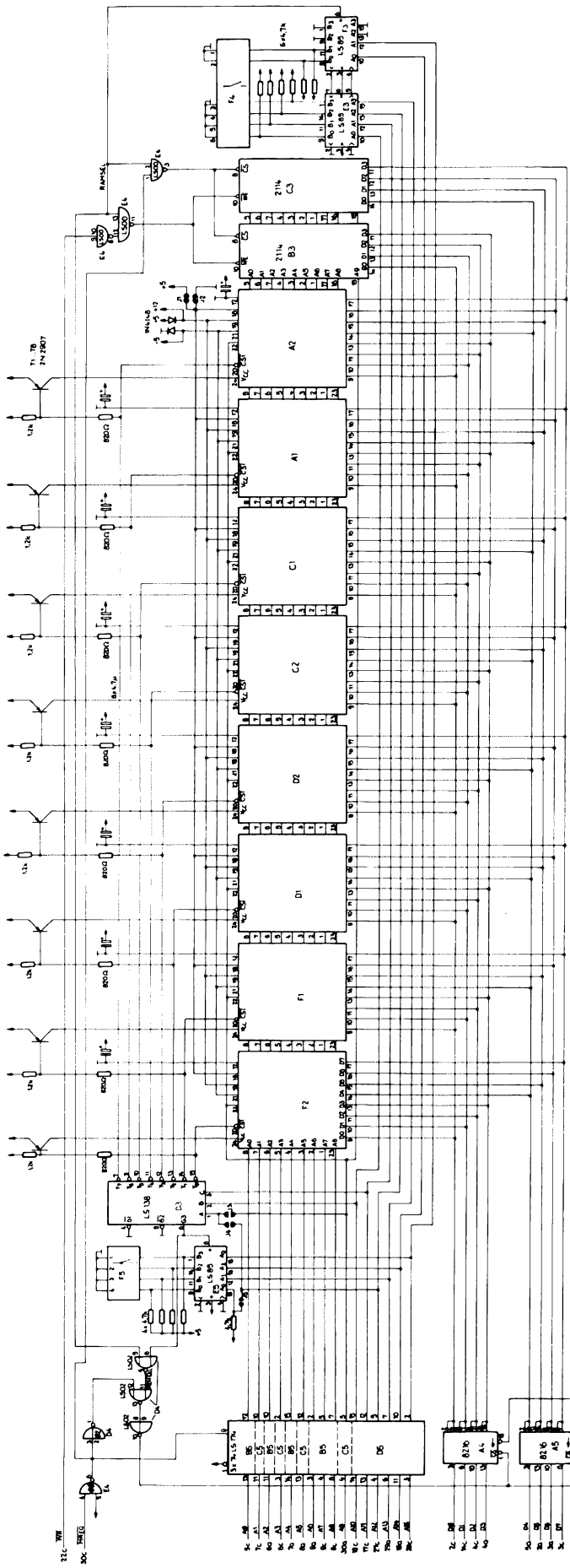
Schalter 1 und 2 ergeben dann 4 mögliche Adreßbereiche:

S1	S2	Adresse
EIN	EIN	0 ..... 3FFF
EIN	AUS	4000 .. 7FFF
AUS	EIN	8000 .. 3FFF
AUS	AUS	C000 .. FFFF

- c) Änderung der Adreßleitungen der PROMS:
  - Jumper J1 öffnen, J2 schließen.
  - Zuleitung +12 V Pin 13a von VG-Leiste auftrennen, zugehörige Block-C (oberhalb D2) und Diode D2 entfernen. Leitung dann mit Adresse 10 IC D5 Pin 15 verbinden.
  - Zuleitung -5V Pin 15a von VG Leiste auftrennen, Diode D3 und Block-C links unterhalb D3 entfernen. Leitung dann mit +5V verbinden.

1/2 kB	Bipolare Proms	J1 geschlossen
		<del>J2</del> geschlossen
1 kB	Bipolare Proms	J5 geschlossen
		J4 geschlossen
1 kB	8708	J1 geschlossen
		J2 geschlossen ✓
		J4 geschlossen ✓

Die Zuweisung der Prom Sockel zu den Adressen ist im Bestückungsplan angegeben. Die Zahlen in eckigen Klammern beziehen sich auf die Verwendung von 1 kB Proms.



**Jumper:**

44	Rechner PROMS (1)	1
	(M7700)	15
45	Rechner PROMS (1)	2
	(M7700)	16
46	Rechner PROMS (1)	3
	(M7700)	17
47	Rechner PROMS (1)	4
	(M7700)	18
48	Rechner PROMS (1)	5
	(M7700)	19
49	Rechner PROMS (1)	6
	(M7700)	20
50	Rechner PROMS (1)	7
	(M7700)	21

41 Rechnet Jumper  
42 Schließen  
43 alle anderen offen

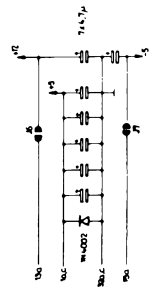
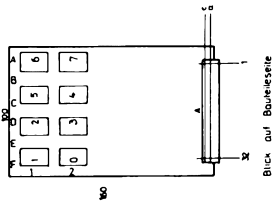


Bild 3-2

Der Seitenwahl-Schalter F5 für die Proms legt die Seite in der die ECB/E arbeitet wie folgt fest:

S1	S2	S3	S4	Seite
AUS	AUS	AUS	AUS	15
AUS	AUS	AUS	EIN	14
AUS	AUS	EIN	AUS	13
AUS	AUS	EIN	EIN	12
AUS	EIN	AUS	AUS	11
AUS	EIN	AUS	EIN	10
AUS	EIN	EIN	AUS	9
AUS	EIN	EIN	EIN	8
EIN	AUS	AUS	AUS	7
EIN	AUS	AUS	EIN	6
EIN	AUS	EIN	AUS	5
EIN	AUS	EIN	EIN	4
EIN	EIN	AUS	AUS	3
EIN	EIN	AUS	EIN	2
EIN	EIN	EIN	AUS	1
EIN	EIN	EIN	EIN	0

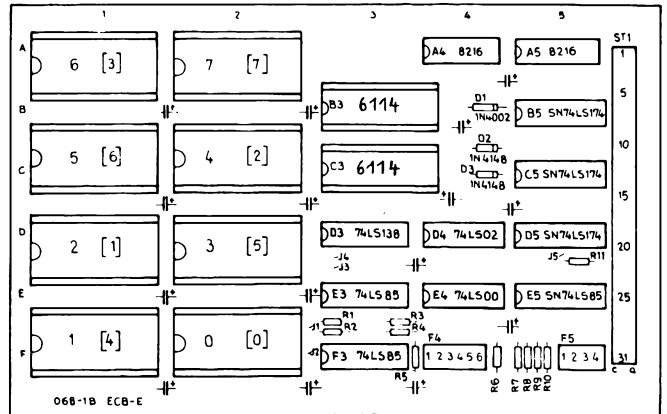
Werden 1 kB Proms verwendet, ist der Schalter S4 offen zu halten. Die Zuweisung ergibt sich dann analog zu obiger Tabelle mit der halben Anzahl von Seiten.

AUS = OPEN

### 3.1.3 Schreib/Lesespeicher

Der Schreib/Lesespeicher ist aus 2 Stück 1 Kbit × 4-Bausteinen aufgebaut.

Seine Anfangsadresse wird durch den 6-fachen DIL-Schalter auf die Platine festgelegt. (Position F4 auf Best.-Plan). Dabei entspricht der Schalter S 1 der Adresse 15, der Schalter S 6 der Adresse 10. Zu beachten ist, daß eine Adresse auf logisch Null verglichen wird, wenn der entsprechende Schalter geschlossen ist.



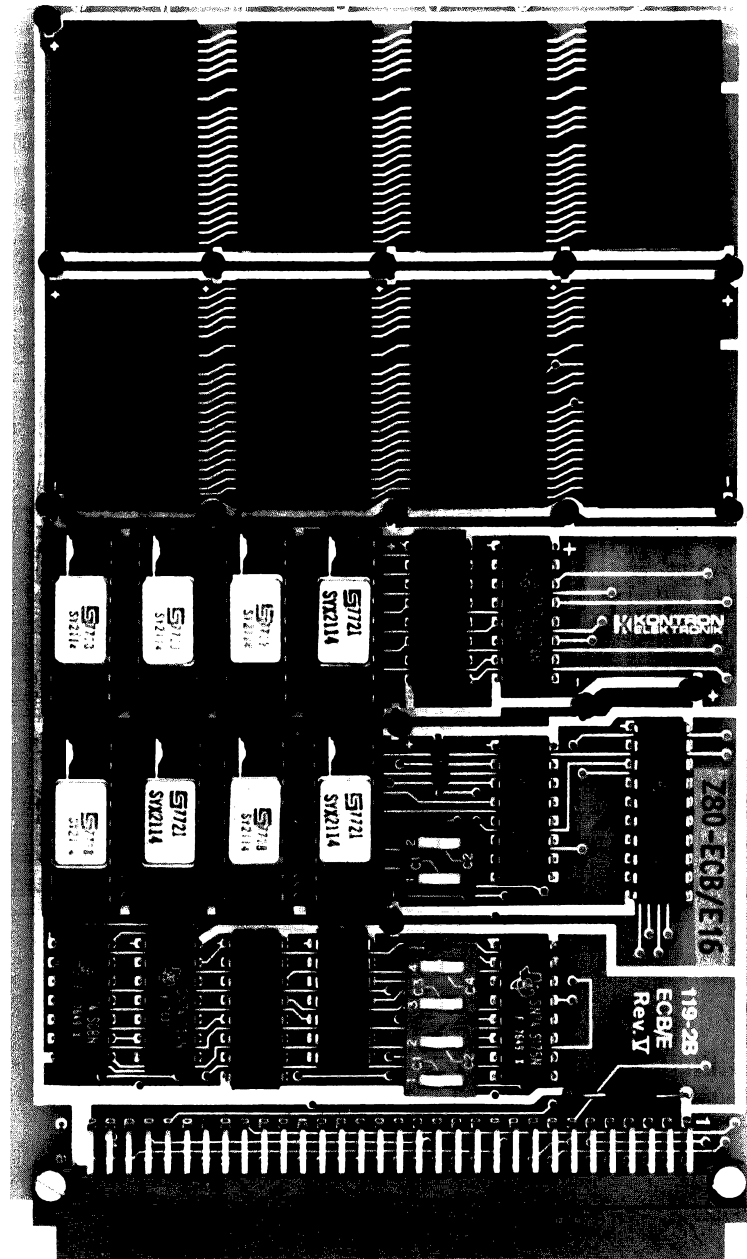
alle Widerstände 4,7kΩ  
alle Elkos 4,7µF

Bild 3—3

# Z80-ECB/E16



Speicher-  
erweiterungsbaugruppe



## 3.2 Die Erweiterungsbaugruppe Z80-ECB/E16

### 3.2.1 Schaltungsbeschreibung

Die Baugruppe umfaßt folgende Funktionseinheiten:

- Bis zu 16 kByte Festwertspeicher (ROM, PROM oder EPROM)
- 1 kByte statisches RAM, bis zu 4 kByte nachrüstbar.
- Decoder
- Schalter zur Speicherbereichsfestlegung
- Busseitiger 64-poliger Stecker nach DIN 41612 (VG 95324)

Bild 3—1 ist das zugehörige Blockschaltbild. Man sieht, daß auch hier die Systembusse gepuffert sind; dadurch ergibt sich zusammen mit den Puffern auf der Zentralplatine Z80-ECB/C eine zweifache Pufferung, was volle Erweiterbarkeit und erhöhte Störsicherheit garantiert.

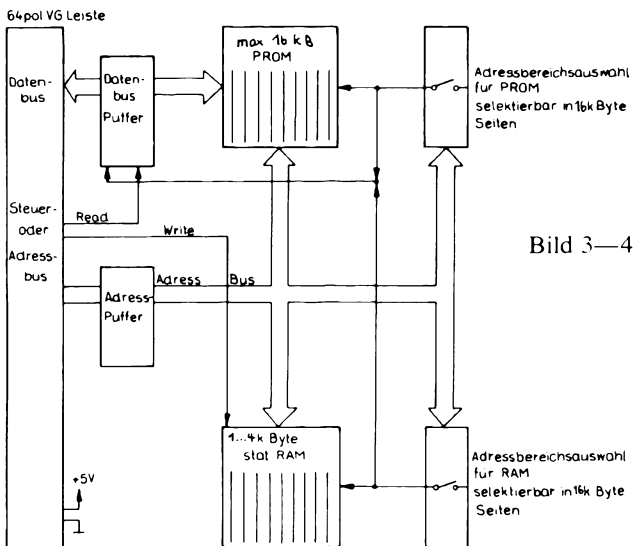


Bild 3—4

Bild 3—4 zeigt die Gesamtschaltung der Baugruppe, die in folgendem näher erläutert wird.

#### 3.2.1.1 Pufferung und Decodierung

Die Übertragungsrichtung der bidirektionalen Datenbus-Puffer wird durch das  $\overline{RD}$ -Signal bestimmt.

Die logischen Verknüpfungen sind wie folgt:

$$\overline{CS} = \overline{MEMSEL} \wedge \overline{MRQ}$$

$$\overline{RD} = \overline{RD}\text{-Signal der Z80-CPU}$$

Dabei sind:

- $\overline{RD}$  = Richtungssignal (High = zur betreffenden Karte hin)
- $\overline{CS}$  = Aktivierung (High = Puffer gesperrt)
- $\overline{MEMSEL}$  = Anforderung für Speicher auf ECB/E

#### 3.2.1.2 Programmspeicher

Die Baugruppe enthält 8 24-Pin-Sockel, in die wahlweise folgende Festwertspeicherbausteine (bzw. hierzu inkompatible Bausteine) eingesteckt werden können

- i 2716 bzw. Z6716 (2 kByte)
- i 2758 (1 kByte)

Gelieferte Verdrahtung ist für i 2716/Z6716.

Bei der Verwendung von 2758-Bausteinen ist zu beachten, daß sich dabei ein nicht zusammenhängender Speicherbereich ergibt (jedes 2. kByte wird physikalisch „doppelt“ adressiert).

Da die Worst-Case Zugriffszeit der EPROM's ein Arbeiten mit maximaler Z80-CPU-Taktfrequenz nicht erlaubt, können nur selektierte ( $t_c < 380$  nsec.) 2716-Bausteine verwendet werden, oder aber die CPU-Taktfrequenz ist entsprechend zu erniedrigen. Das Einfügen eines Wait-Zyklus ist über einen hardwaremäßigen Zusatz zur Baugruppe möglich. Die entsprechende Schaltung findet man im CPU Manual (Adding a Wait State to any M 1 Cycle). Sie besteht aus lediglich einem D-Flipflop-Baustein. Eine entsprechende Schaltung ist auf der Rückwandverdrahtung der ECB/R vorgesehen.

Die Adreßdecodierung erfolgt durch den zweipoligen DIL-Schalter. Sein Schalter 1 schaltet Adreßbit 14, Schalter 2 Adreßbit 15. Schalter auf der den RAM's zugewandten Seite niedergedrückt entspricht dem Wert Null des zugehörigen Adreßbits.

#### 3.1.3 Schreib/Lesespeicher

Der Schreib/Lesespeicher ist aus 2 Stück 1 Kbit  $\times$  4-Bausteinen aufgebaut. Durch Einstecken weiterer 1K $\times$ 4-Bausteinen in die 6 freien Fassungen läßt sich der Speicherbereich auf der Platine in 1 kB-Schritten auf 4 kByte erweitern.

Die Anfangsadresse wird durch den 4-fachen DIL-Schalter auf der Platine festgelegt.

Die RAM-Dekodierung erfolgt mittels des 4-poligen DIP-Schalters.

Die vier höchstwertigen Adreßbits werden wie folgt geschaltet:

Schalter 1 : Adreßbit 12

Schalter 2 : Adreßbit 13

Schalter 3 : Adreßbit 14

Schalter 4 : Adreßbit 15

Schalter auf der den RAM's zugewandten Seite niedergedrückt entspricht dem Wert des Adreßbits von 0.

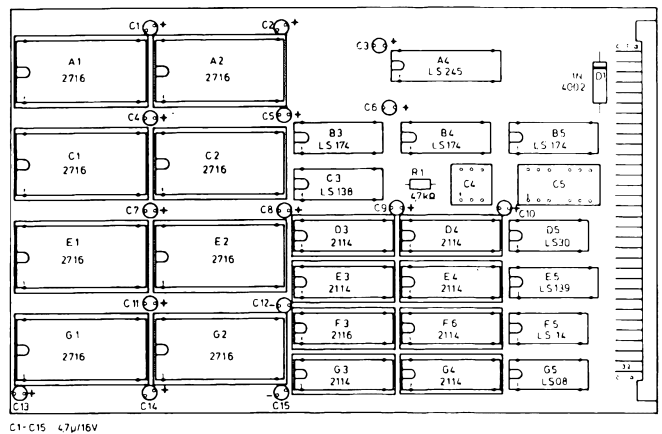


Bild 3—5

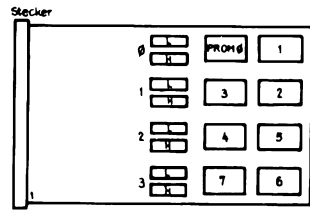
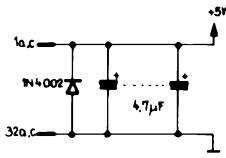
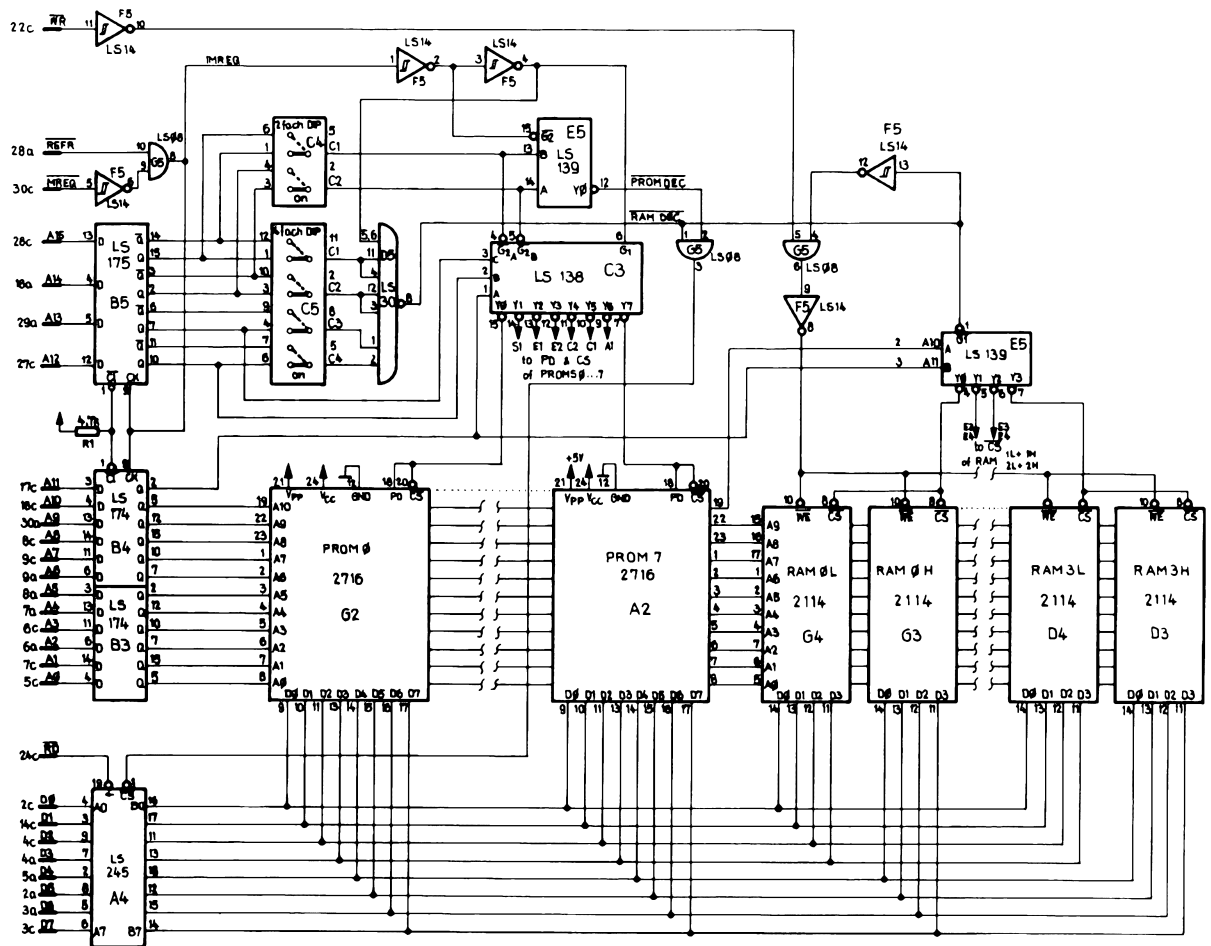


Bild 3-6

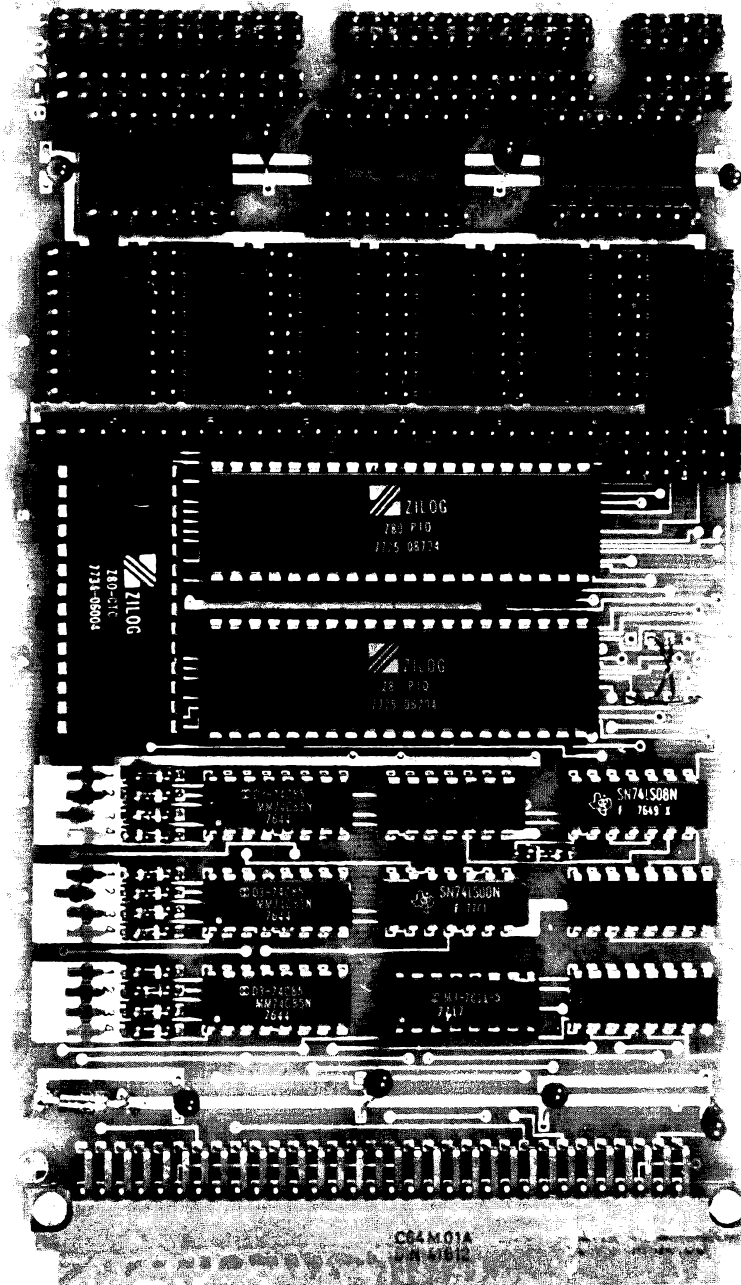




# Z80-ECB/I



Ein/Ausgabe-  
Erweiterungsbaugruppe



## 4. Die Ein/Ausgabebaugruppe Z80-ECB/I

Die Baugruppe umfaßt folgende Funktionseinheiten

- 4 Parallelschnittstellen (2 Stück Z80-PIO)
- 4 Zeitgeber/Ereigniszahlkanäle (1 Stück Z80-CTC)
- 9 Unverdrahtete Fassungen für Standard-Interface-Bausteine
- Busseitig 64-poliger Stecker nach DIN 41612 (VG 95324)
- I/O-seitig 3M-WWP-Pfostensteckverbinder

Bild 4—1 zeigt das zugehörige Blockschaltbild:

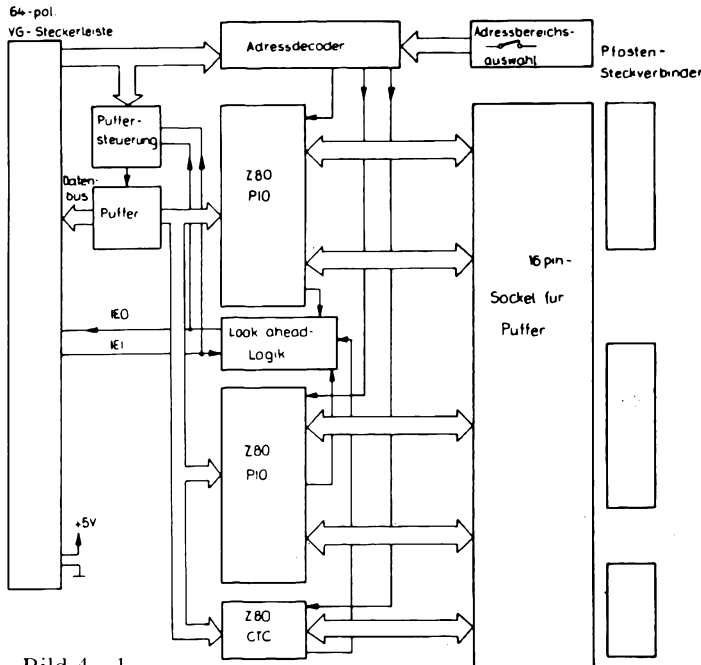


Bild 4—1

Die Baugruppe behandelt mit ihren 2 Bausteinen Z80-PIO und dem 1 Baustein Z80-CTC insgesamt vier 8bit-Ein/Ausgabekanäle, die jeweils über 2 zusätzliche Leitungen für Quittungsbetrieb verfügen, und vier Ereigniszähler/Zeitgeberkanäle. Einzelheiten über den Baustein Z80-PIO finden Sie im Z80-PIO-Technical Manual; der Baustein Z80-CTC ist im Z80-CTC-Technical Manual beschrieben.

Auch bei dieser Baugruppe wurde das Prinzip der doppelten Pufferung weiterverfolgt.

Der hierdurch bedingte Aufwand besteht in den beiden bidirektionalen Puffern 8216 und einem entsprechend Bild 2—3 programmierten Puffer-Steuer PROM HM 7611.

Die zugehörigen logischen Verknüpfungen sind:

$$\overline{IIOIN} = \underbrace{(\overline{IORQ} \wedge \overline{IODEC} \wedge \overline{RD})}_V \wedge \underbrace{(\overline{IORQ} \wedge \overline{M1} \wedge \overline{IEI} \wedge \overline{IEO})}_{INTA}$$

$$\overline{CS} = 0$$

Dabei sind:

- $\overline{IIOIN}$  = Richtungssteuersignal (High = zur betreffenden Karte hin)
- $\overline{CS}$  = Aktivierungssignal (High = Puffer gesperrt)
- $\overline{M1}$ : Maschinenzyklus 1
- $\overline{IORQ}$ : I/O Anforderung
- $\overline{IODEC}$ : I/O Auswahl
- $\overline{RD}$ : Read
- $\overline{IEI}$ : Interrupt Enable IN
- $\overline{IEO}$ : Interrupt Enable OUT

Die Adressen der einzelnen Ports können über DIL-Schalter festgelegt werden.

Es stehen 16 Adressen für jeden PIO bzw. CTC zur Verfügung. Die Schalterstellung ist invertiert zu betrachten:

Adresse	S4	S3	S2	S1
FXH =	AUS	AUS	AUS	AUS
und 0XH =	EIN	EIN	EIN	EIN

Durch direkte Auswahl der I/O-Bausteine auf der ECB/C-Karte müssen die Adreßbits 2 und 3 immer auf log 1 sein.

Bei einer Schalterstellung

S4	S3	S2	S1
AUS	EIN	AUS	EIN

am Schalter E 3 muß der PIO links wie folgt adressiert werden:

$$\left. \begin{array}{l} S4 \hat{=} \text{I/O Adreßbit 7} \\ S3 \hat{=} \text{I/O Adreßbit 6} \\ S2 \hat{=} \text{I/O Adreßbit 5} \\ S1 \hat{=} \text{I/O Adreßbit 4} \end{array} \right\} \begin{array}{l} SX = \text{AUS} \rightarrow \text{Adresse} = 1 \\ SX = \text{EIN} \rightarrow \text{Adresse} = 0 \end{array}$$

I/O Adresse (hex)

- ACH = PIO L Port A Data
- ADH = PIO L Port B Data
- AEH = PIO L Port A Control
- AFH = PIO L Port B Control

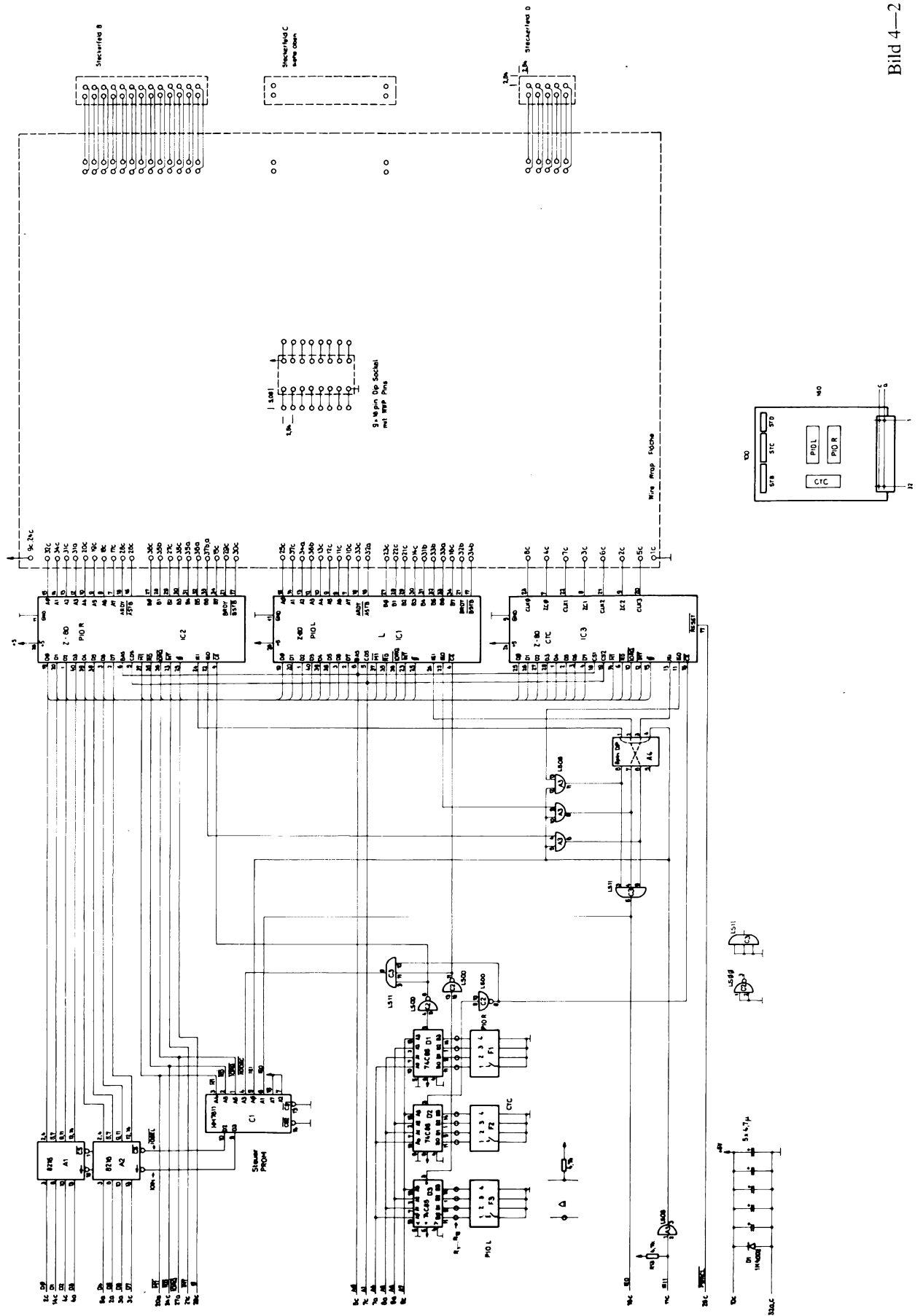
Auf diese Weise ist es möglich das ECB-System auf insgesamt 5 I/O-Karten zu erweitern. Es stehen dann 20 I/O-Ports = 200 Leitungen und 5 Zähler/Zeitgeberbausteine zur Verfügung.

Zur Realisierung anwenderspezifischer Interfaces wurden auf der Baugruppe neun 16-polige Fassungen untergebracht, deren Ausgänge über Wire-Wrap-Stifte entsprechend der im Anwendungssystem gewünschten Steckerbelegung mit den 3 Pfostensteckverbindern verdrahtet werden.

Für den Fall, daß komplexere Interface-Schaltungen oder Verbindungen zur Busseite der Platinen hergestellt werden sollen, können diese Pfostensteckverbinder als Verbindungen zu „Huckepack“- (= „Sandwich“-) Platinen verwendet werden. Sollen die Ausgänge der PIO- bzw. CTC-Bausteine ohne zwischengeschaltete Puffer benutzt werden, sind die Grenzwerte bzw. Charakteristika zu beachten:

**Z80-PIO:**  $I_{OL} = 1,8 \text{ mA}$  bei  $V_{OL} = \text{max. } 0,4 \text{ V}$  (Port A)  
 $I_{OHD} = 1,5 \dots 3,8 \text{ mA}$  bei  $V_{OH} = 1,5 \text{ V}$  und  $R_{ext} = 390 \Omega$  (Darlington-Ausgänge der Ports B)

**Z80-CTC**  $I_{OL} = 1,8 \text{ mA}$  bei  $V_{OL} = \text{max. } 0,4 \text{ V}$   
 $I_{OHD} = 1,5 \dots 3,8 \text{ mA}$  bei  $V_{OH} = \text{max. } 1,5 \text{ V}$  und  $R_{ext} = 390 \Omega$



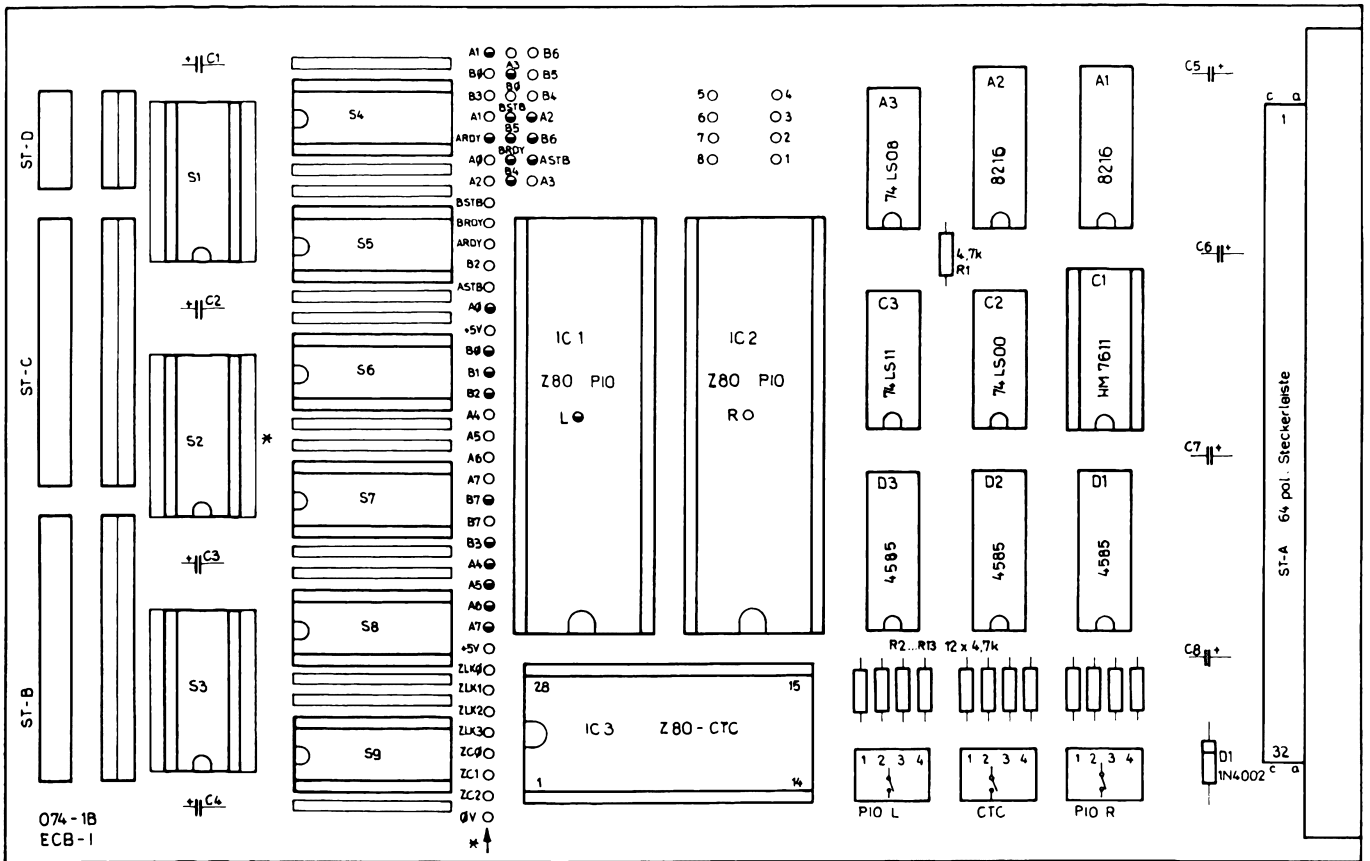
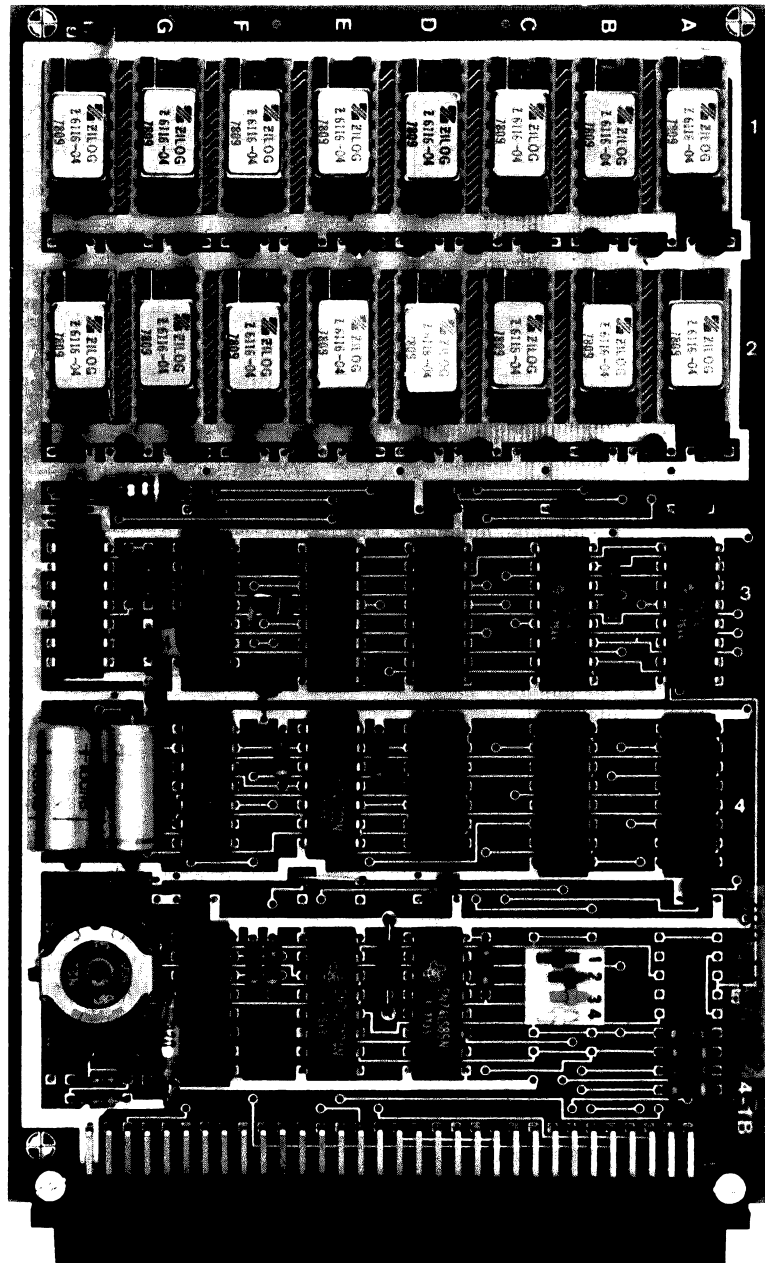


Bild 4—3

# Z80-ECB/D



Dynamische Speicher-  
erweiterungsbaugruppen



## 5. Speichererweiterungsbaugruppen der Serie Z80-ECB/D (/D8, /D16 und /D32)

Die Baugruppen der Serie ECB/D sind Baugruppen zur Schreib/Lesespeichererweiterung für RAM-intensive Systeme, die mit der Familie ECB aufgebaut sind. Sie sind in 3 Versionen erhältlich:

- Z80-ECB/D8 ist eine 8 kByte-Speichererweiterungseinheit, die mit dynamischen 4 kbit- 16 pin-RAM-Bausteinen realisiert ist. Die Anfangsadresse des gesamten Speicherbereichs auf der Platine läßt sich mit einem DIL-Schalter auf Speicheradressen zwischen 0 und 56k in 8k-Schritten einstellen.

Bild 5—1: ECB/D8 Adreßauswahl-Schalterstellungen 4-fach DIP Schalter B5

S1	S2	S3	S4	ADRESSE (HEX)
AUS	EIN	EIN	EIN	0 ... 1FFFFH
AUS	AUS	EIN	EIN	2000 ... 3FFFFH
AUS	EIN	AUS	EIN	4000 ... 5FFFFH
AUS	AUS	AUS	EIN	6000 ... 7FFFFH
AUS	EIN	EIN	AUS	8000 ... 9FFFFH
AUS	AUS	EIN	AUS	A000 ... BFFFFH
AUS	EIN	AUS	AUS	C000 ... DFFFFH
AUS	AUS	AUS	AUS	E000 ... FFFFFH

- Z80-ECB/D16 ist eine 16 kByte-Speichererweiterungseinheit, die mit dynamischen 16 kbit-16-pin-ZILOG-RAM-Bausteinen Z6116 (oder Äquivalenztyp) realisiert ist. Die Baugruppe ist durch Einstecken weiterer 8 Bausteine Z6116 auf 32 kByte erweiterbar.

Bei Bestückung mit nur 16 kByte ist die Anfangsadresse wie bei der Z80-ECB/D32 nur auf die Anfangsadressen 0H und 8000 H zu legen.

S4 = EIN : 0H  
 S4 = AUS : 8000H  
 S2, 3 geschlossen  
 S1 offen.

- Z80-ECB/D32 ist eine 32 kByte-Speichererweiterungseinheit, die mit dynamischen 16 kbit-16pin-ZILOG-RAM-Bausteinen Z6116 (oder Äquivalenztyp) realisiert ist. Die Anfangsadresse des Speicherbereichs läßt sich auf 0 oder auf 8000H festlegen.

S4 = EIN : 0H      S2, 3 geschlossen  
 = AUS : 8000H    S1 offen

In allen Fällen ist die Zugriffszeit im Temperaturbereich von 0 ... 50°C zu 350 nsec. garantiert, so daß die Baugruppe zusammen mit der Z80-ECB/C mit der maximalen Betriebsfrequenz von 2,5 MHz ohne das Einfügen von WAIT-Zyklen arbeiten kann. Die für die dynamischen RAM-Bausteine zusätzlich benötigten Speisespannungen -5 V und +12 V werden mit einem Gleichspannungswandler auf der Platine erzeugt, so daß der gesamten Platine von außen her lediglich eine einzige +5 V-Versorgungsspannung (0,6 A max.) zuzuführen ist.

Die gesamte Logik zum Demultiplexen und zur Erzeugung des Speicherrefresh in Z80-CPU-gesteuerten Systemen ist mit auf der Baugruppe untergebracht. Der Refresh erfolgt abhängig vom laufenden Programm max. alle 0,7 msec unter

der Kontrolle der Z80-CPU vollautomatisch ohne zusätzliche Software und ohne Verlust von Verarbeitungszeit, d.h. für den Anwender verhält sich die Baugruppe nach außen hin genauso wie jede statische Speichererweiterung.

Folgende Punkte sind jedoch zur Vermeidung von Informationsverlusten im dynamischen Speicher zu beachten:

- Bei Verwendung des Signals RESET ist dafür Sorge zu tragen, daß die Dauer dieses Signals unter 2 msec liegt, da die CPU kein Refresh-Signal erzeugt, solange ihr RESET-Eingang auf LOW liegt. Ein einfacher Schalter genügt an dieser Stelle also nicht, falls Informationsverluste im dynamischen RAM beim Systemrückstellvorgang unerwünscht ist; die Problematik kann z. B. mit einem Differenzierglied umgangen werden, das der RESET-Leitung der CPU vorgeschaltet wird.

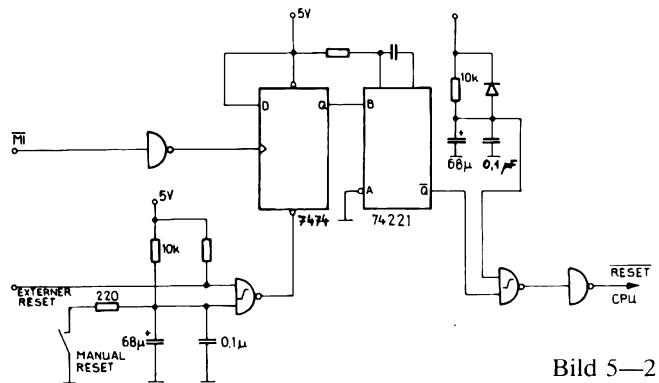


Bild 5—2

- Bei direktem Speicherzugriff (DMA) ist die Länge der DMA-Zugriffe so kurz zu wählen, daß immer noch alle 2 msec sämtliche Speicherstellen des dynamischen RAM's „aufgefrischt“ werden.

Die Z80-CPU schaltet bei jedem M1-Zyklus die Refresh-Adresse um 1weiter. Der Zeitraum zwischen zwei Refreshvorgängen ist abhängig von der Art der Befehle, die in einem Programm vorkommen. Geht man von den längsten Befehlen (23 Taktzyklen) aus, so beinhalten diese zwei M1-Zyklen.

Damit ergibt sich:

$$1 \text{ Refreshzyklus} = \frac{23}{2} \text{ Taktzyklen} \cdot 128$$

$$\approx 0,7 \text{ ms bei einer Taktfrequenz von } 2,5 \text{ MHz.}$$

Da die RAMS alle 2 ms einen vollständigen Refreshzyklus durchlaufen haben müssen, ergibt sich eine Restzeit von 1,3 msec für andere Aktivitäten.

- Das eben über DMA Gesagte ist sinngemäß auch auf die Benützung des WAIT-Eingangs der Z80-CPU anzuwenden, da auch WAIT-Zyklen den Refresh-Vorgang vorübergehend unterbinden (s. Z80-CPU-Technical Manual).

Die Dauer eines periodisch anliegenden WAIT-Signals muß also unter 1,3 msec liegen. Die Baugruppe Z80-ECB/D8 läßt sich durch Austausch der 4 kbit-16 pin-Bausteine gegen pinkompatible 16 kbit-16 pin-Bausteine (z.B. ZILOG Z6116) und durch Verlegung der folgenden Drahtbrücken auf den Leistungsumfang der Z80-ECB/D16 oder Z80-ECB-D32 erweitern (s. Bild 5—3):

Die Pinbelegung ist identisch mit der Zentralbaugruppe ECB/C, so daß 1:1 Verdrahtung möglich ist. Einzig zusätzlich belegte Leitung ist Pin 26a mit dem in Anschluß beschriebenen WRITE EN-Signal.

Auf den Baugruppen wurde außerdem ein Signal implementiert, das das Einschreiben in die RAM's verhindert ("WRITE EN"). Dieses Signal wurde auf den Systembus auf Pin 26a herausgeführt. Ausgeliefert wird die Baugruppe mit geschlossener Drahtbrücke S1, so daß der

Schreibzugriff freigegeben ist. Soll WRITE EN von externer Logik kontrolliert werden, ist diese Drahtbrücke zu entfernen und das entsprechende Bus-Signal zur Verfügung zu stellen.

- a) Drahtbrückenfeld A 5 (J1)  
 auftrennen: J1/1—10  
                   J1/9—6  
                   J1/8—7  
                   J1/2—8  
                   J1/3—10  
                   J1/5—9
- b) Drahtbrückenfeld B 5 (J2)  
 auftrennen: J2/2—8  
                   J2/6—9  
                   J2/1—14  
 verbinden: J2/2—14.

Bild 5—3. Drahtbrücken für 16 K RAMS:

Weitere Eingriffe in die Schaltung sind nicht erforderlich. Die Schaltung der Baugruppen finden Sie im Bild 5—6.

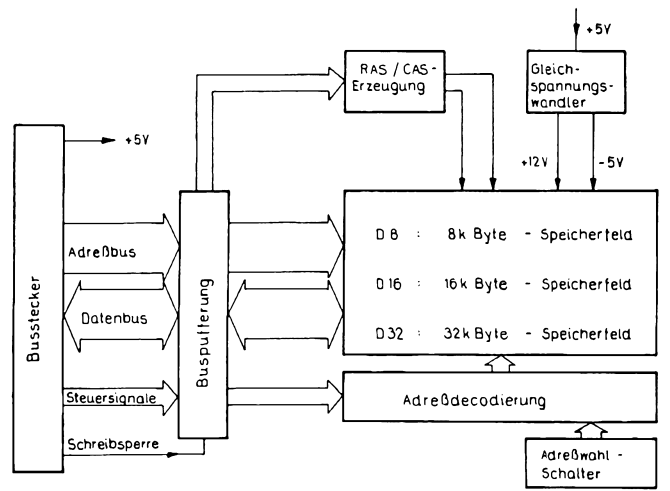


Bild 5—4

**Zur Beachtung:**

Dynamischer Speicher sind grundsätzlich störempfindlich und kritisch im Zeitverhalten.

Es empfiehlt sich daher, Baugruppen der ECB/D-Serie möglichst in Steckplätzen nahe der Zentralbaugruppe einzusetzen.

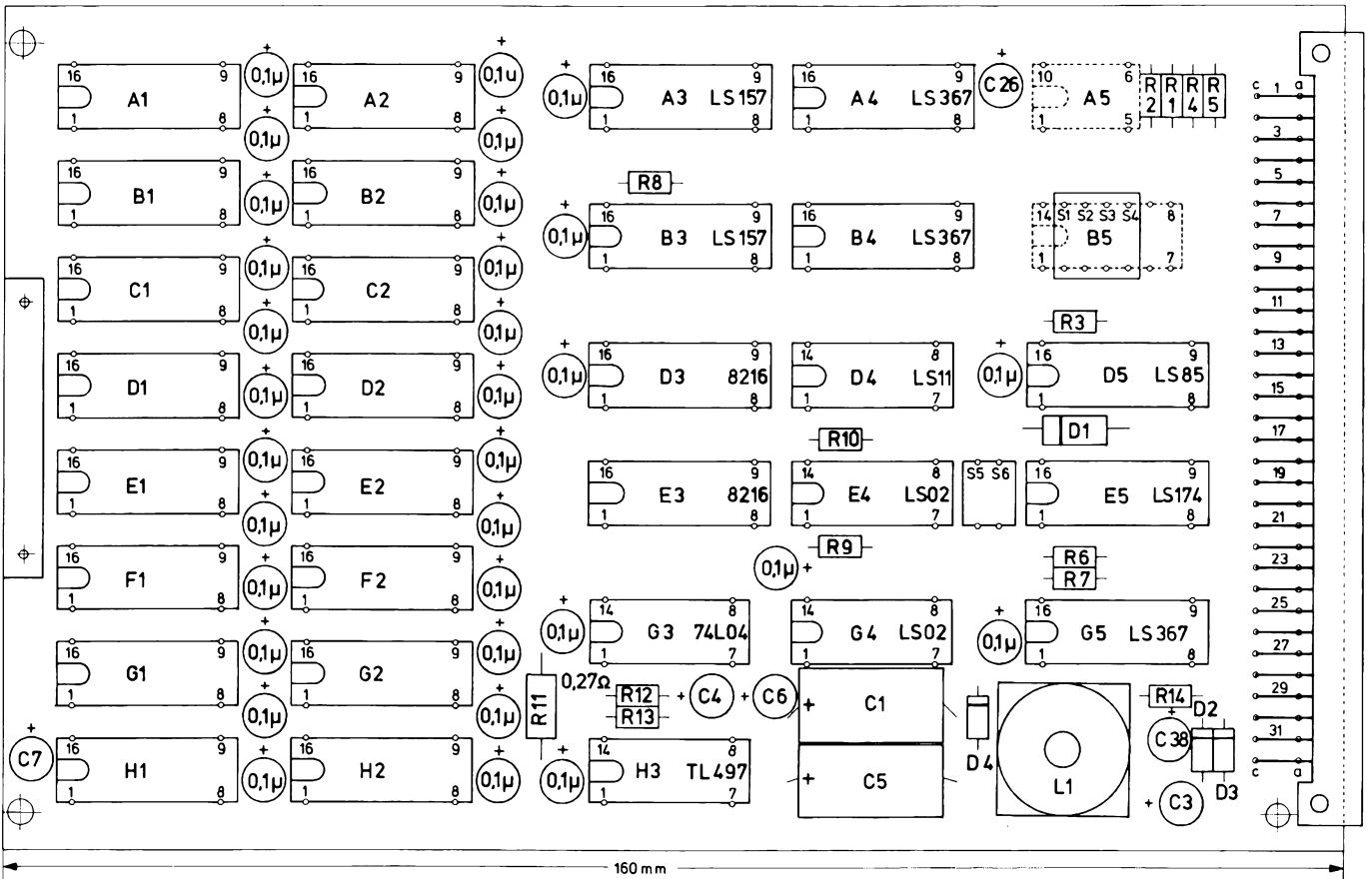
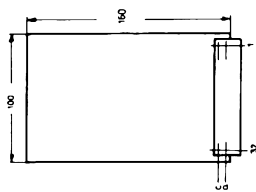
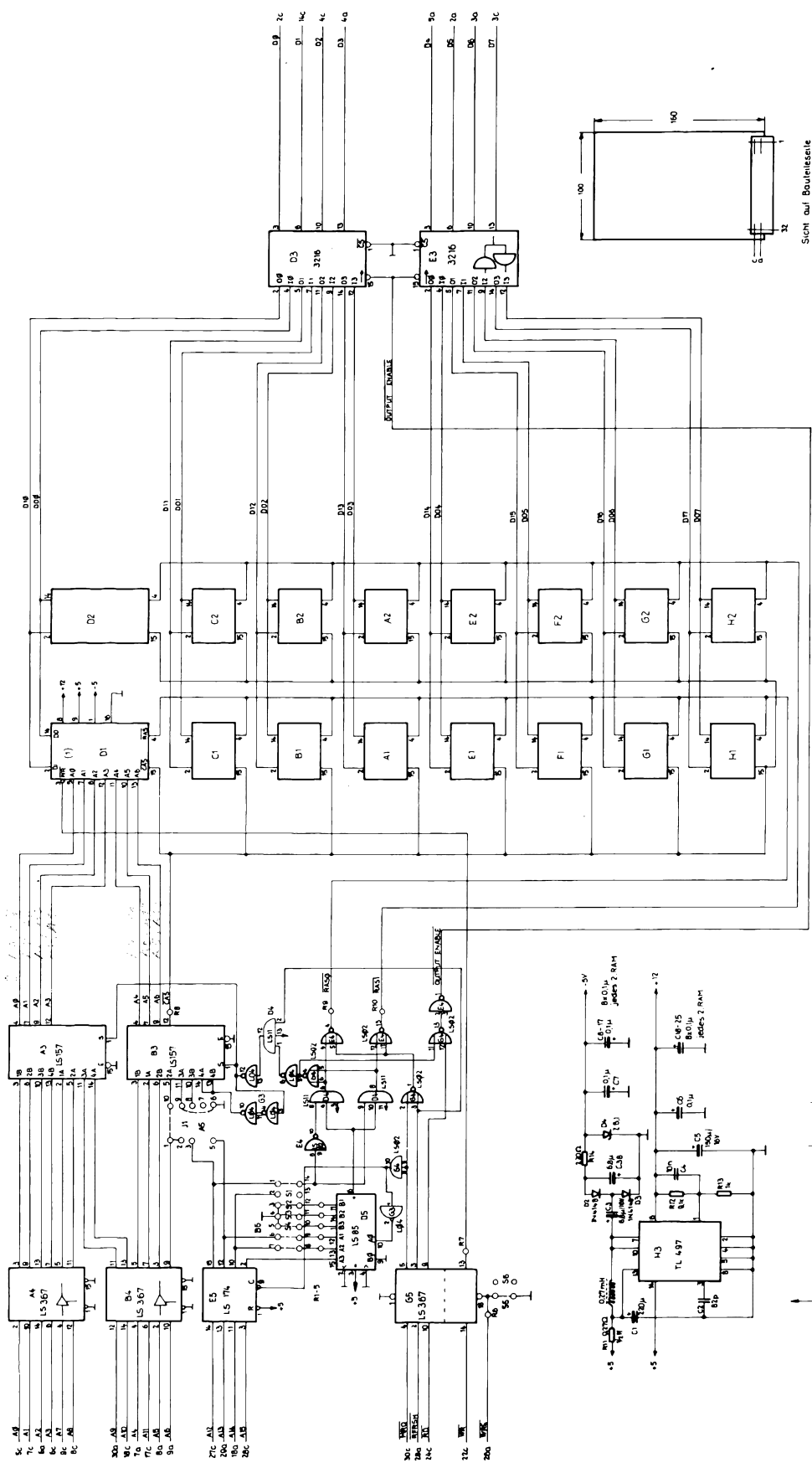


Bild 5—5



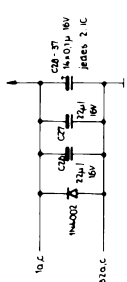
Sicht auf Bauelemente

Alle RAM Sockel verdrahtet wie (1)

Jumper für 16K RAMS  
 Jumperfeld BS (J2)  
 auftrennen J1 / 1-10  
 verbinden J1 / 1-10

Jumperfeld AS (J1)  
 auftrennen J1 / 1-10  
 verbinden J1 / 1-10

S1	S2	S3	S4	ADRESSE
EN	EN	EN	EN	0
AUS	EN	EN	EN	1FFF
EN	AUS	EN	EN	2000
AUS	AUS	EN	EN	3FFF
EN	EN	EN	EN	4000
AUS	AUS	EN	EN	5FFF
EN	EN	AUS	AUS	6000
AUS	EN	AUS	AUS	7FFF
EN	EN	AUS	AUS	8000
AUS	EN	AUS	AUS	9FFF
EN	EN	AUS	AUS	A000
AUS	EN	AUS	AUS	BFFF
EN	EN	AUS	AUS	C000
AUS	EN	AUS	AUS	DFFF
EN	EN	AUS	AUS	E000
AUS	EN	AUS	AUS	FFFF

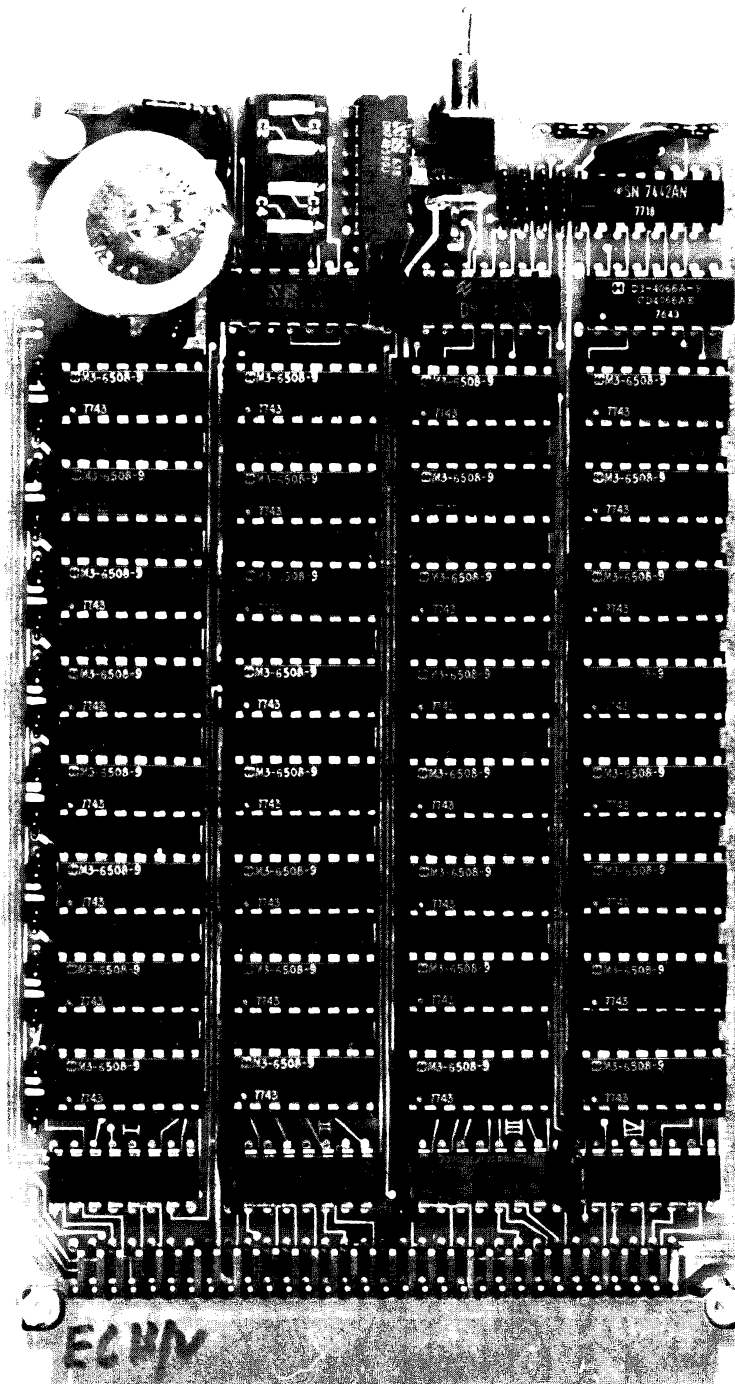




# Z80-ECB/V



# CMOS-Speicher- erweiterungsbaugruppe



## 6. 4 kByte-CMOS-RAM-Karte Z80-ECB/V

### 6.1 Beschreibung

Die Einfach-Europakarte ECB/V ist eine voll statische 4 kByte-CMOS-Speicherbaugruppe hoher Flexibilität und Packungsdichte. Wegen der extrem niedrigen Stromaufnahme ist sie besonders für folgende zwei Anwendungen geeignet:

- Große und schnelle Arbeitsspeicher: z.B. 64 kByte = 16 Karten mit 2,5 A-Netzteil in einem 19"-Einschub evtl. über Zentral-Akku zu puffern: 20 mA Ruhestrom. Diese Packungsdichte ist nur durch die vernachlässigbare Wärmeentwicklung der CMOS-RAMS möglich.
- Nicht-flüchtige Speicher. Programme und Daten können beim Austesten gegen Überschreiben geschützt werden (Speicher-Schutz-Schalter). Beim Abschalten des Systems oder bei Stromausfall bleiben die eingegebenen Daten erhalten.

Ferner kann man bei der System-Entwicklung auf die Verwendung von löschbaren PROM's verzichten, weil z.B. Programme bis zu ihrer endgültigen Festlegung auf der Karte erhalten bleiben (Speicherschutzschalter in Stellung "ROM"). Löschungen und Modifikationen können selektiv durch die normalen WRITE-Operationen vorgenommen werden, dazu muß lediglich der Speicherschutzschalter auf "RAM" geschaltet werden.

In gewissem Sinne bietet also diese CMOS-RAM-Karte eine preisgünstige, schnelle und platzsparende Alternative zu Kernspeicher-Systemen.

Eine oft willkommene Erleichterung für die Microcomputer-Entwickler bieten auch die auf der Karte angebrachten Basis-Adreß-Umschalter. Durch Betätigung eines 4 bit-DIL-Schalters kann so der Ansprechbereich der Karte z.B. im 64 kByte Adreßraum beliebig gewählt werden.

Die Worstcase-Zugriffszeit der Baugruppe ist im Temperaturbereich 0 ... 50°C zu 350 nsec garantiert, so daß die Verwendung zusammen mit der Baugruppe Z80-ECB/C ohne das Einfügen von WAIT-Zyklen bei einer max. Taktfrequenz von 2,5 MHz möglich ist.

Die Blockschaltung zeigt Bild 6—1.

Die vollständige Schaltung der Baugruppe finden Sie in Bild 6—2; die Pinbelegung ist identisch mit der Zentral-Baugruppe ECB/C, so daß 1:1-Verdrahtung möglich ist.

### 6.2 Bedienungsanleitung

a) Speicherschutzschalter (Write-Protect).  
Der Schalter in Stellung Leiterbahnseite ergibt Betriebsart "Quasi-ROM", in Richtung Bestückungsseite können Informationen in die RAM-Bausteine eingeschrieben werden.

b) Adreß-Vorwahlschalter (Basis-Adresse)  
Der DIL-Schalter dient zur Vorwahl der höchstwertigen 4 Adreßbits. Nr. 4 schaltet das höchstwertige bit A15, Nr. 3, 2, 1 entsprechen A14, A13, A12. Schalter in Richtung Pufferbatterie: Adreßbit muß 1 sein, damit die Karte angesprochen wird. Schalter in Richtung des Kippschalters: Adreßbit = 0 für die Selektion.

Beispiel:

DIL-Schalter Nr. 1 und 2 zum Akku, Nr. 3, 4 in Richtung Kippschalter, Write-Protect-Schalter zeigt zur Lötseite.

A 15, 14, 13, 12, 11 ..... 0  
0 0 1 1 X ..... X (HEX : 3XXX)

Die HEX Adresse 3000 — 3FFF ist für die Betriebsart "Quasi-ROM" reserviert.

### □ Zur Beachtung:

Die Baugruppe ECB/V wird mit geladenem Akkumulator und aufgetrennter Verbindung zum Akkumulator geliefert, um seine Entladung während der Lagerungszeit zu vermeiden.

Ebenso ist vom Anwender dafür Sorge zu tragen, daß der Akkumulator vor Einsatz der Baugruppe als Netzausfallsicherung aufgeladen ist und zu berücksichtigen, daß sich der Akkumulator bei ausgeschaltetem Gerät entlädt.

Die Information im Speicher der Baugruppe bleibt nach Stromausfall oder Abschaltung ca. 5 Tage erhalten, vorausgesetzt, daß der Akkumulator zu diesem Zeitpunkt voll aufgeladen war.

Der Eingang  $\overline{\text{DPR}}$  dient zur Synchronisation des  $\overline{\text{CS}}$ -Eingangs bei Spannungsausfall.

Ist dies irrelevant, wirkt  $\overline{\text{DPR}}$  als Schreibschutz und ist über einen 1 k $\Omega$ -Pull-Up-Widerstand auf +5V zu legen. Nähere Einzelheiten über Verwendung von  $\overline{\text{DPR}}$  s. Abschnitt 6.3.

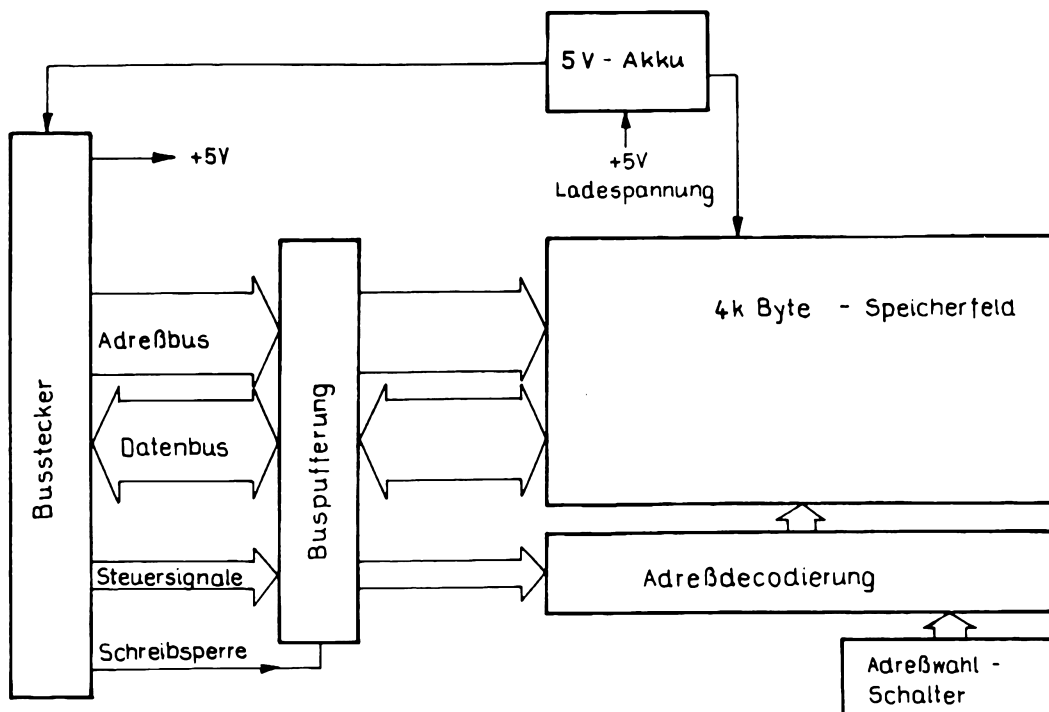


Bild 6—1

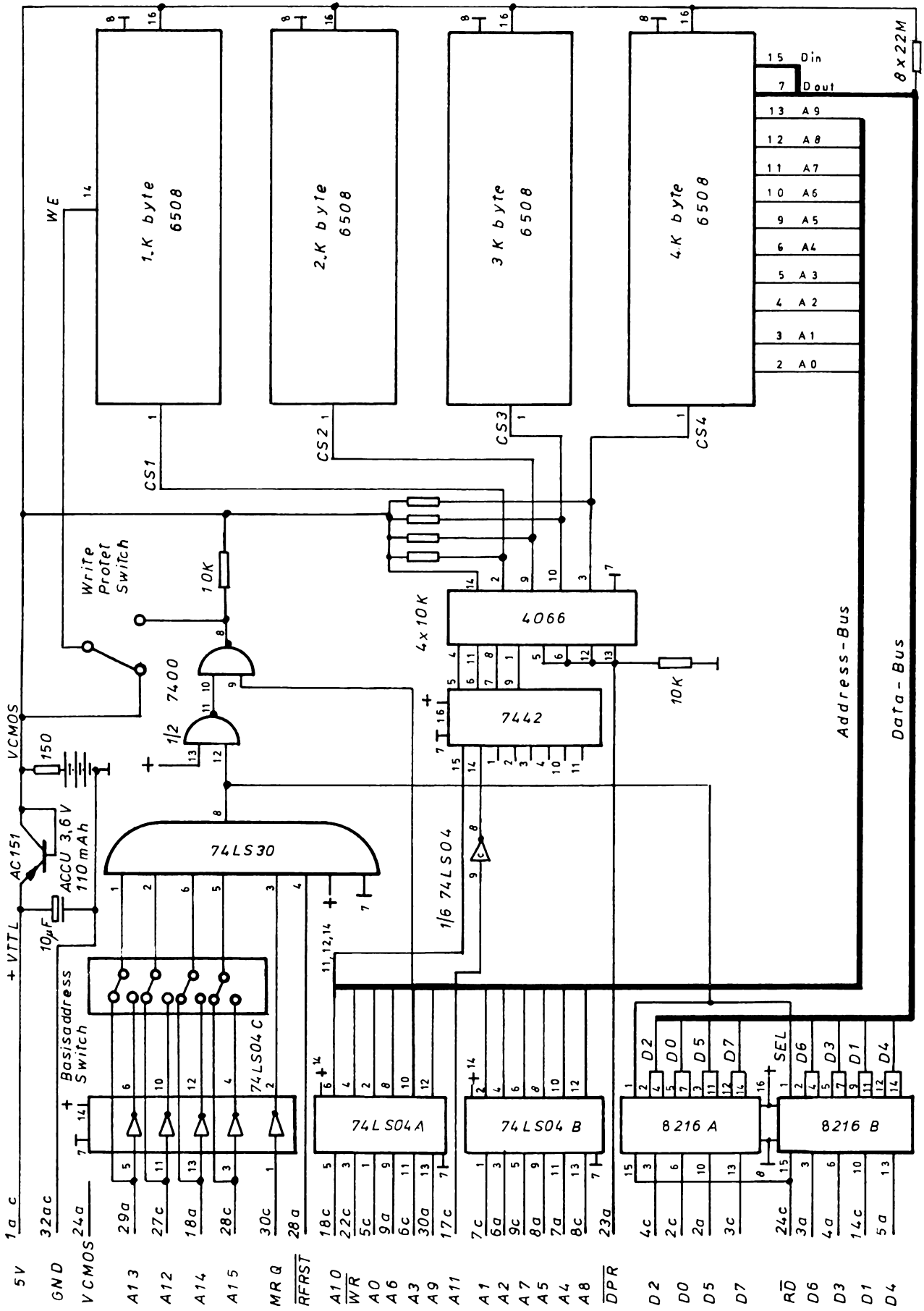


Bild 6-2

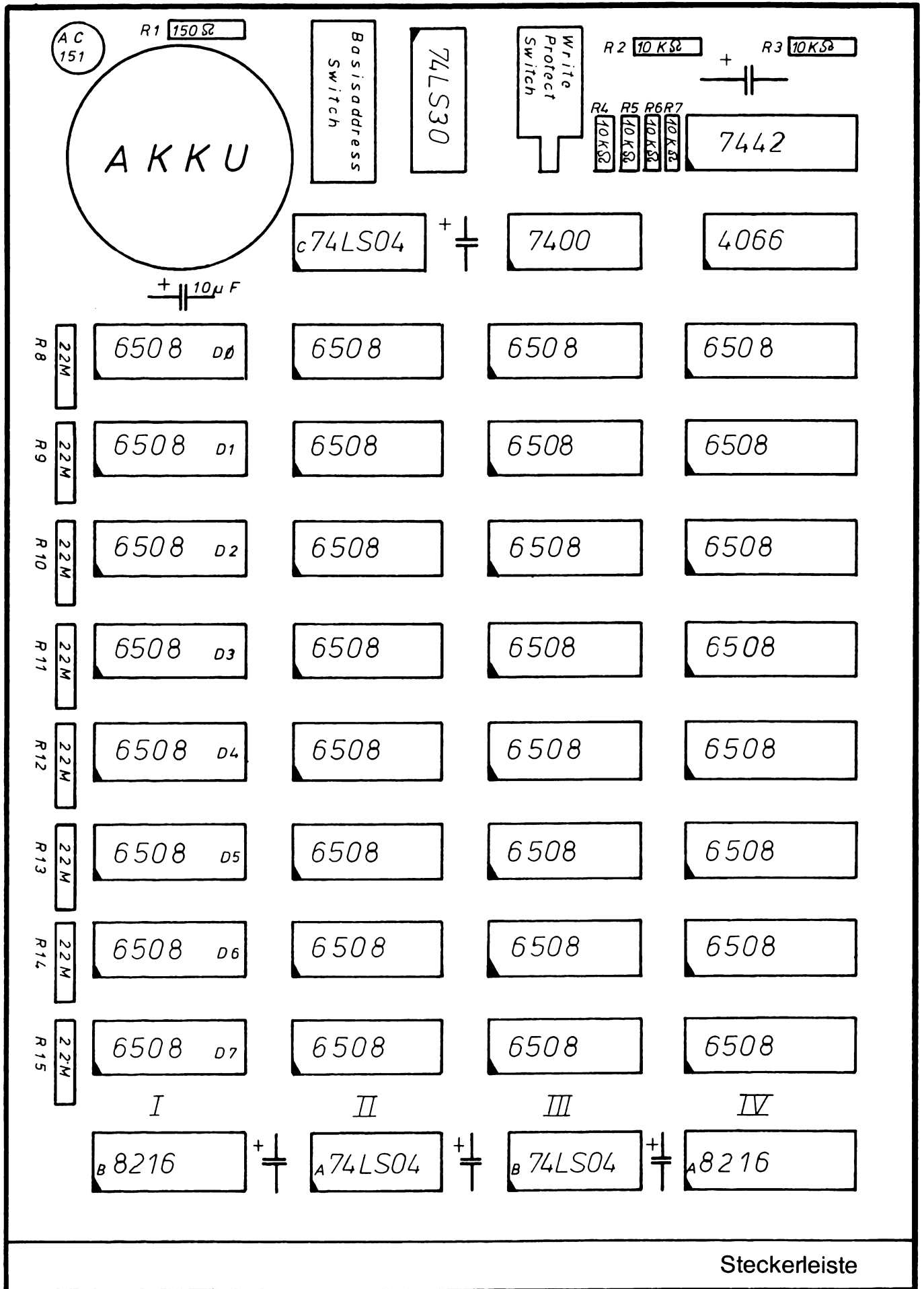
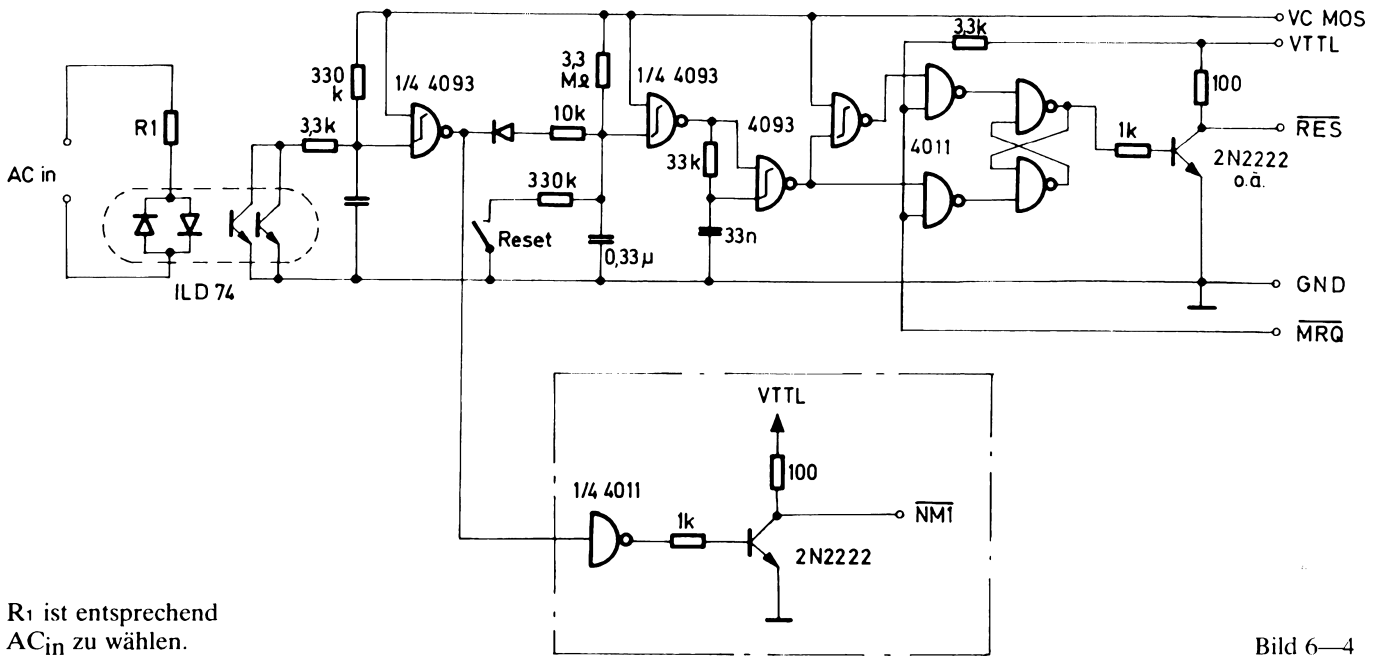


Bild 6—3

Powerfail- und Reset-Synchronisation für ECB/V



R<sub>1</sub> ist entsprechend  
AC<sub>in</sub> zu wählen.

Bild 6—4

V <sub>CMOS</sub>	pin 24 a	$\overline{\text{MRQ}}$	30c
V <sub>TTL</sub>	1 a, 1 c	$\overline{\text{RES}}$	31 c und 23 a bei ECB/V
NMI*	20c	GND	32 a, c

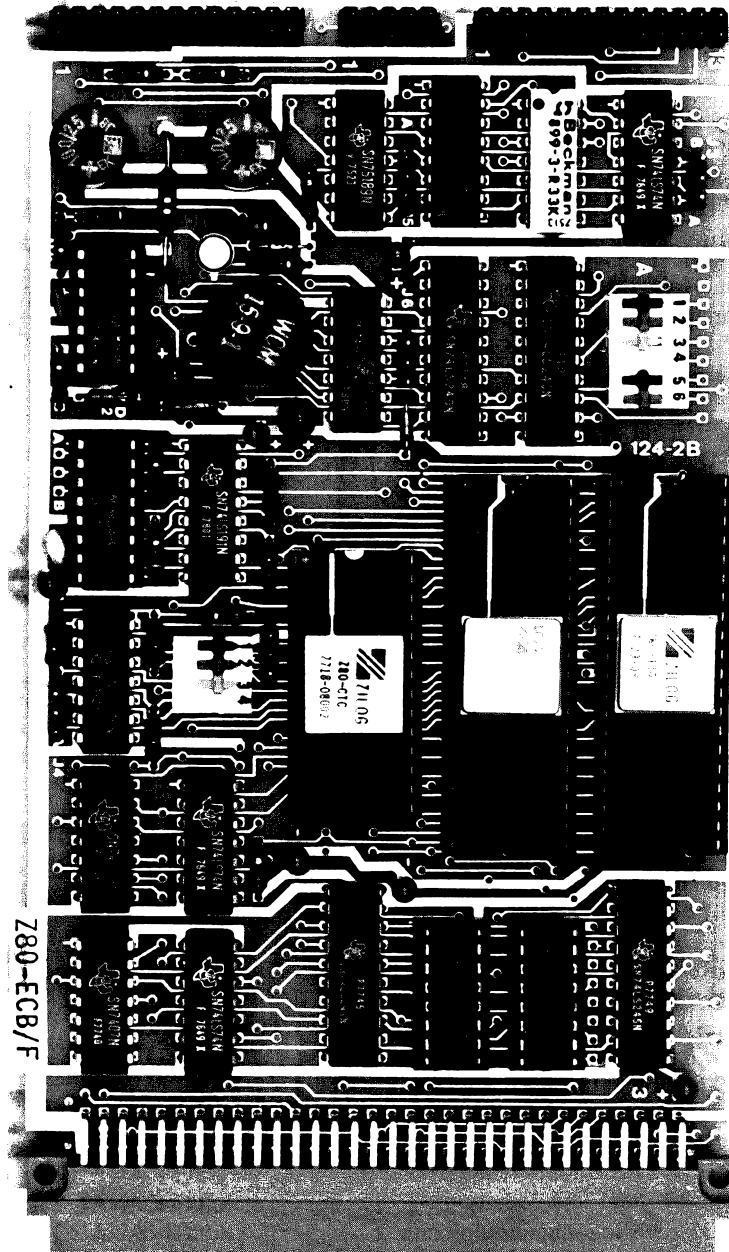
\* nur für Interrupt-Auslösung bei Stromausfall



# Z80-ECB/F



Ein/Ausgabe-Baugruppe für  
allgemeine Aufgaben und  
Floppy-Disk-Steuerung



Vorläufige Beschreibung

# 7. Ein/Ausgabe-Baugruppe für allgemeine Aufgaben und Floppy-Disk-Steuerung Z80-ECB/F

## 7.1 Übersicht

Die Baugruppe ECB/F ist eine universell einsetzbare Serien- ein- bzw. Ausgabeeinheit. Sie ist kompatibel zur Z80-ECB-Baugruppenserie im Einfach-Europaformat und vereint die Leistungsfähigkeit von drei Z80-Peripheriebausteinen, nämlich des Z80-CTC, des Z80-PIO und des Z80-SIO. Zwei separate Voll-Duplex-Kanäle erlauben serielle Datenübertragung in nahezu allen bekannten asynchronen und synchronen Verfahren. Über den Z80-CTC und einen PLL-Frequenzsynthesizer sind alle gebräuchlichen Baudraten für den Asynchronkanal bzw. alle Übertragungsraten von 0—550 kbit/s für den Synchronkanal programmierbar.

Die Baugruppe ermöglicht den unmittelbaren Anschluß von bis zu acht hardsektorierten Floppy-Disk-Laufwerken verschiedener Größen und Speicherkapazitäten, sowie asynchron arbeitenden Datenendgeräten mit genormten RS232C- oder 20 mA Current Loop-Schnittstellen. Durch den Einsatz des Z80-SIO ist die ECB/F in hervorragender Weise auch für Rechner-Rechnerkopplungen geeignet. International genormte Übertragungsverfahren wie SDLC und HDLC werden vom Z80-SIO abgehandelt.

Zum Betrieb der Baugruppe genügt eine einzige Versorgungsspannung von +5 Volt. Die Betriebsspannungen für das RS232C-Interface werden durch Gleichspannungswandler auf der ECB/F erzeugt.

Eine leistungsfähige, in zwei Stufen implementierte Floppy-Disk Betriebssoftware steht ebenfalls zur Verfügung. Zusammen mit der Zentralsbaugruppe ECB/C stellt die ECB/F somit das Kernstück eines Floppy-Disk basierenden Anwendersystems dar.

## 7.2 Eigenschaften und Hardwarekomponenten

Die Baugruppe enthält folgende Funktionseinheiten:

- a) CPU-Interface
  - Schmitt Trigger Pufferung sämtlicher Bussignale
  - Adreßdekoder für CTC, PIO und SIO
  - Look Ahead Logik für Interruptprioritätssteuerung (Daisy Chain)
  - 64-poliger Busstecker nach DIN 41612
- b) SIO-Synchronkanal (Floppy Disk Interface)
  - Voll programmierbarer PLL-Frequenzsynthesizer zur Erzeugung der Übertragungsrate (2 CTC-Kanäle)
  - Programmierbarer, rein digitaler Datenseparator mit 16 Zeitkonstanten
  - Datenencoder zur Daten-/Clock-Gemischerzeugung
  - Diskstatussignale über Z80-PIO
  - Schmitt Trigger Pufferung aller Ein-/Ausgänge
- c) SIO-Asynchronkanal
  - Voll programmierbare Baudratenerzeugung von 50—38400 Baud (1 CTC-Kanal)
  - RS 232C und 20mA Current Loop Interface
  - Gleichspannungswandler für ±12 Volt
  - 6 Modem Signale im RS 232C Pegel
- d) Sonstiges
  - 6 DIP Schalter (über PIO) zur Programmierung von Systemvariablen
  - Steckbare Jumper zur Umstellung des Synchronkanals an andere Peripherie
  - I/O seitig drei Standardstecker zum Anschluß der Peripheriegeräte

### 7.2.1 CPU-Interface

Sämtliche Bussignale der ECB/F sind über die 64-polige VG-Messerleiste entsprechend der PIN-Belegung der übrigen Z80-ECB-Baugruppen herausgeführt (siehe Abschnitt 1).

Bild 7—1 zeigt das Blockschaltbild der Baugruppe

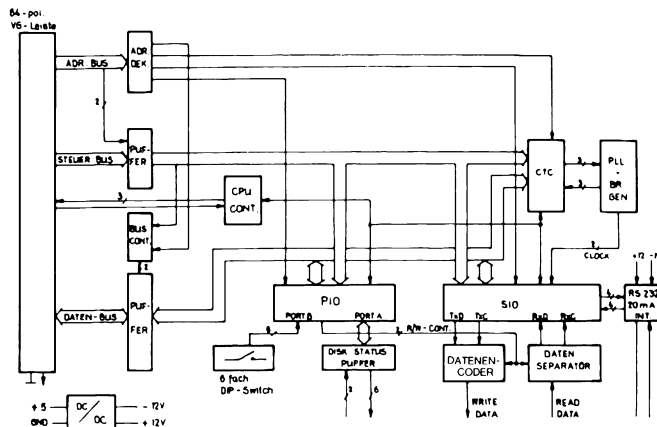


Bild 7—1: Blockschaltbild der ECB/F

Alle Leitungen sind gepuffert und stellen pro Eingang nicht mehr als eine LS TTL-Last ( $\hat{=} 0,25 \text{ mA}$  bei LOW) dar. Schmitt Trigger Bausteine erhöhen hierbei die Störsicherheit. Alle Ausgänge erlauben den Anschluß von kapazitiv hochbelasteten Bussystemen.

Die Leitungen  $\overline{\text{WAIT}}$  und  $\overline{\text{INT}}$  sind Open Collector-Ausgänge ohne Pull up Widerstände, da diese bereits auf der Zentralbaugruppe ECB/C vorhanden sind. Eine "Look ahead Logik" (Baustein F5: 74LS08) sorgt für die schnelle Durchschaltung der Z80-Interruptprioritätssteuerung (Daisy Chain). Die ECB/F-interne Priorität ist: CTC, SIO, PIO.

### 7.2.2 Adressierung

Die Adressierung der Bausteine Z80-CTC, Z80-PIO und Z80-SIO erfolgt über das Dekoderprom B6 (HM7611). Da die Floppy-Disk-Treibersoftware feststehende I/O-Adressen verwendet, wurde darauf verzichtet, einstellbare Adressierung zu ermöglichen. Das Dekoderprom dekodiert aus den Adressen A2—A7 folgende I/O-Adressen:

	A7	A6	A5	A4	A3	A2	A1	A0
Z80-CTC	→ 0	0	0	0	1	1	X	X
Z80-SIO	→ 0	0	0	1	1	1	X	X
Z80-PIO	→ 0	0	1	0	1	1	X	X

Es liefert außerdem ein  $\overline{\text{I/O-DEC}}$  Signal, welches immer dann aktiv ist, wenn einer der drei Z80-Bausteine auf der ECB/F adressiert wird. Dieses Signal wird zur Datenbussteuerung benötigt.

Die Adreßleitungen A0 und A1 sind in der üblichen Weise mit den Port- bzw. Channel-Select-Eingängen von PIO, SIO und CTC verbunden.

Tabelle 7.1 enthält sämtliche I/O-Adressen der ECB/F in hexadezimaler Form.

CTC-KANAL 0	0CH
CTC-KANAL 1	0DH
CTC-KANAL 2	0EH
CTC-KANAL 3	0FH
SIO PORTA DATA	1CH
SIO PORTB DATA	1DH
SIO PORTA CONTROL	1EH
SIO PORTB CONTROL	1FH
PIO PORTA DATA	2CH
PIO PORTB DATA	2DH
PIO PORTA CONTROL	2EH
PIO PORTB CONTROL	2FH

Tabelle 7.1: Kanal- und Portadressen von CTC, SIO und PIO



Das Dekoderprom kann selbstverständlich durch ein anders programmiertes ausgetauscht werden. Zu beachten ist, daß Promausgang D3 (PIN 9) ein I/O-DEC Signal führt.

### 7.3 Floppy Disk Interface

#### 7.3.1 Allgemeines

Die ECB/F erlaubt den unmittelbaren Anschluß von bis zu 8 Floppy-Disk-Laufwerken (mit hardsektorierten Disketten) in gemultiplexten Systemen entsprechend Bild 7—2. Durch die hohe Flexibilität der Hardware sind Mini- oder Normal-laufwerke mit einfacher oder doppelter Speicherdichte anschließbar.

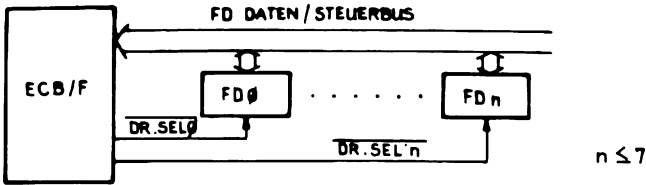


Bild 7—2: Blockschaltbild eines Systems mit mehreren Laufwerken

Zum Betrieb der unterschiedlichen Laufwerke ist lediglich eine geringfügige Modifikation in der Betriebssoftware nötig.

#### 7.3.2 Diskettenaufbau und Speicherkapazität

Das Speichermedium Diskette ist in verschiedene Spuren und Sektoren eingeteilt. Die Spuren bilden konzentrische Kreise um den Mittelpunkt einer Diskette. Ihre Anzahl ist durch den Durchmesser einer Diskette festgelegt; ihr Abstand voneinander durch die Mechanik eines Schrittmotors, der den Schreib-/Lesekopf eines Laufwerks von Spur zu Spur bewegt. Jede Spur wiederum ist unterteilt in mehrere kleinere Abschnitte, den Sektoren. Der Sektor ist die kleinste adressierbare Einheit auf einer Diskette und faßt einen Datenblock von 128 Byte (256 Byte bei double density Laufwerken).

Der Beginn eines jeden Sektors ist durch einen physikalischen Sektorimpuls gekennzeichnet. Eine derartige Einteilung wird als "hardsektoriert" bezeichnet. Im Gegensatz dazu erfolgt beim "softsektorierten" Verfahren die Kennzeichnung eines Sektoranfangs durch bestimmte Bitfolgen (Kennungsmarken). Dafür gibt es verschiedene Normen.

#### ⚠ Achtung:

Die ECB/F gestattet **nicht** das Schreiben und Lesen von softsektorierten Disketten. Hardsektorierte Disketten sind leicht zu erkennen:

In einer Öffnung unterhalb des Aufklebers erscheinen beim Drehen in regelmäßigen kurzen Abständen Löcher, die zur Erzeugung der Sektorimpulse abgetastet werden. Bei softsektorierten Disketten ist dies nicht der Fall. Hier ist nur ein Loch zur Kennzeichnung von Sektor 0 vorhanden.

In Tabelle 7.2. sind für verschiedene FD-Laufwerke charakteristische Größen zusammengestellt.

	Spuren	Sekt/ Spur	Sektoren	Bytes/ Sek.	Total (kbyte)
Mini-FD	32	16	512	128	65,5
Mini-FD (double d.)	32	16	512	256	130
Normal-FD	77	32	2464	128	315
Normal-FD (double d.)	77	32	2464	256	630

Tabelle 7.2: Charakteristische Werte verschiedener FD-Laufwerke

#### 7.3.3. Aufzeichnungsformat

Daten werden in einem speziellen Synchronverfahren auf Disketten aufgezeichnet. Der serielle Datenstrom wird hierbei mit einem Referenztakt so gemischt, daß jedes Datenbit in der sogenannten „Bitzelle“ durch zwei Taktbits eingeraht ist (Bild 7.3 a).

Dieses Verfahren erlaubt beim Rücklesen von der Diskette auf einfache Art die Wiedergewinnung von Daten und Referenztakt (im sogenannten Daten-/Taktseparator).

Der Vorteil liegt auf der Hand:

Für Daten und Takt ist nur eine Spur notwendig.

Die Aufzeichnung des Referenztakts ist prinzipiell notwendig, da sonst an die Konstanz der FD-Drehzahl in der Praxis unhaltbare Forderungen gestellt werden müßten.

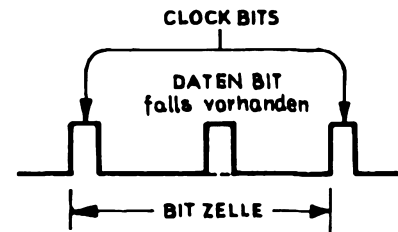


Bild 7-3 a: Bitzelle

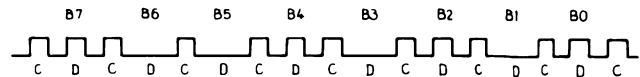


Bild 7—3 b: Impulsdigramm der Hexadezimalzahl 95

Bild 7—3: Das Daten/Takt-Gemisch

Jeder Sektor enthält neben dem eigentlichen Datenblock noch weitere Informationen, die der Adreßbestimmung auf der Diskette und dem Dateihandling dienen. In Bild 7—4 ist der gesamte Informationsblock eines Sektors dargestellt.

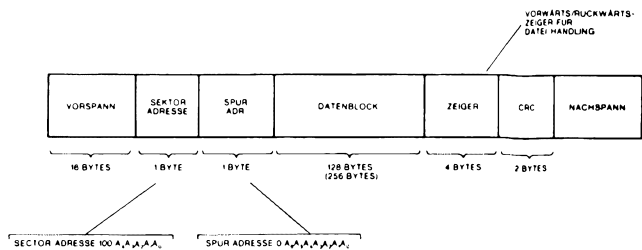


Bild 7—4: Sektorformat

Nach einem Vorspann von Nullen sind 16 Bit zur Kennzeichnung der Sektor- und Spuradresse aufgezeichnet. Wie Bild 7—4 zeigt, muß die Sektoradresse an der höherwertigsten Stelle eine log. „1“ führen. Diese „erste Eins“ in einem Sektor wird von der Hardware erkannt (Flipflop E5, Blatt 3) und für Synchronisationszwecke verwendet. Der Spuradresse folgt der eigentliche Datenblock, der abhängig von der Speicherdichte des Laufwerks 128 oder 256 Byte enthält. Die folgenden vier Byte dienen in einem Disk Operating System (DOS) zum Dateihandling. Jeweils 16 Bit bilden einen Zeiger auf den vorhergehenden bzw. nachfolgenden Sektor einer Datei. Schließlich folgt noch ein 16 Bit CRC-Wort (Cyclic Redundancy Check). Dieses CRC-Wort wird beim Schreiben mit übertragen und erlaubt beim Lesen das Erkennen von Übertragungsfehlern.

#### 7.3.4 Datenseparator

Der Datenseparator erzeugt aus dem Daten-/Takt-Gemisch-Daten und Referenztakt. Diese Signale werden anschließend unmittelbar den entsprechenden SIO-Eingängen zugeführt (Receiver Data — RxDA bzw. Receiver Clock — RxCA).

Die Schaltung besteht aus den beiden D-Flipflops A1 und dem programmierbaren Zählerbaustein 74LS191 (E3). Die Funktionsweise entspricht dem Impulsdigramm in Bild 7—5.

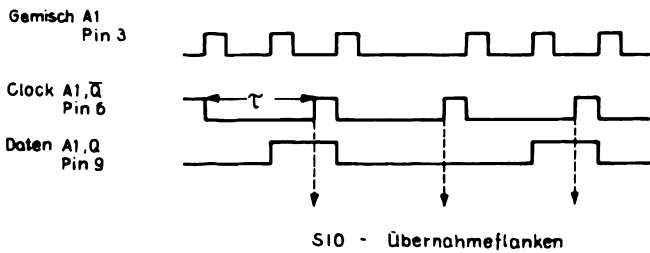


Bild 7—5: Impulsdigramm des Datenseparators

Die Zeitkonstante  $\tau$  ist durch 4 DIP Schalter in Inkrementen von 400 ns auf 16 verschiedene Werte einstellbar. Der erforderliche Wert hängt ab von der Art des verwendeten Laufwerks bzw. dessen Übertragungsrate. Die üblichen Übertragungsraten und die dafür empfohlenen Zeitkonstanten sind in Tabelle 7.3 für verschiedene FD-Laufwerke zusammengestellt.

Laufwerk	Übertragungsrate [kbit/s]	Zeitkonstante $\tau$ [ $\mu$ s]
Mini-FD	125	4,8—6,4
Mini-FD (double dens.)	250	2,4—3,2
Normal-FD	250	2,4—3,2
Normal-FD (double dens.)	500	1,2—1,6

Tabelle 7.3: Zusammenhang zwischen Übertragungsrate und Zeitkonstante  $\tau$

Der Zusammenhang zwischen Zeitkonstante und Schalterstellung (DIP Switch E4 ist in Tabelle 7.4 dargestellt).

S4	S3	S2	S1	Zeitkonstante $\tau$ [ $\mu$ s]
ON	ON	ON	ON	6,4
ON	ON	ON	OFF	6,0
ON	ON	OFF	ON	5,6
ON	ON	OFF	OFF	5,2
ON	OFF	ON	ON	4,8
ON	OFF	ON	OFF	4,4
ON	OFF	OFF	ON	4,0
ON	OFF	OFF	OFF	3,6
OFF	ON	ON	ON	3,2
OFF	ON	ON	OFF	2,8
OFF	ON	OFF	ON	2,4
OFF	ON	OFF	OFF	2,0
OFF	OFF	ON	ON	1,6
OFF	OFF	ON	OFF	1,2
OFF	OFF	OFF	ON	0,8
OFF	OFF	OFF	OFF	0,4

Tabelle 7.4: Schalterstellung und Zeitkonstante

### 7.3.5 Datenencoder

Der Datenencoder erzeugt aus der Verknüpfung von seriellen Daten und doppeltem sowie einfachem Übertragungstakt das zur Floppy-Disk Ansteuerung notwendige Daten/Takt-Gemisch (Baustein F4).

### 7.3.6. Erzeugung der Übertragungsrate

Die Übertragungsrate ist über 2 CTC-Kanäle und einen PLL-Schaltkreis (F3: CD4066) in weiten Grenzen programmierbar. Grundsätzlich muß der CTC so programmiert werden, daß am Ausgang des PLL-Bausteines das Doppelte der ge-

wünschten Übertragungsrate entsteht. Flipflop E6 (Sheet 1) erzeugt daraus den für den SIO erforderlichen symmetrischen Transmittertakt ( $\overline{\text{TxCA}}$ ).

Kanal 1 des Z80-CTC bestimmt die eine Eingangsfrequenz der PLL-Schaltung. Die zweite Eingangsfrequenz erzeugt Kanal 2 des Z80-CTC durch Teilung des rückgekoppelten PLL-Ausgangssignals. Es gilt näherungsweise folgender Zusammenhang:

$$f_{\text{out}} = f_{\text{in}} \times n$$

Hierbei ist:

- $f_{\text{out}}$  Ausgangsfrequenz des PLL ( $\text{VCO}_{\text{out}}$ -Pin 4)
- $f_{\text{in}}$  Eingangsfrequenz des PLL ( $\text{PCA}_{\text{in}}$ -Pin 14)
- $n$  Zeitkonstante CTC-Kanal 2 (Countermode)

Die erzeugbare Ausgangsfrequenz hängt wesentlich von der Dimensionierung des RC-Gliedes R9, C1 ab. Bild 7—6 zeigt den Zusammenhang zwischen  $\text{VCO}_{\text{in}}$  (Eingangsspannung des spannungsgesteuerten Oszillators) und der Ausgangsfrequenz des VCO's.

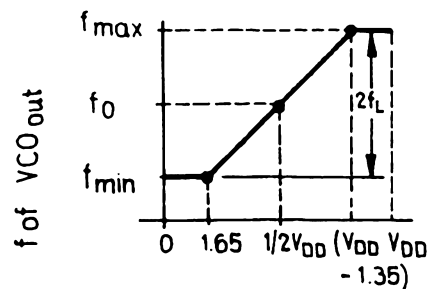


Bild 7—6

$\text{VCO}_{\text{in}}$

Die maximale Frequenz (ca. 1,4 MHz) tritt demnach bei  $\text{V}_{\text{DD}} = 1,65 \text{ V}$  ( $\text{V}_{\text{DD}} = 12 \text{ V}$ ) die minimale bei 0 V auf. Die ausgelieferte RC-Dimensionierung erlaubt durch die Modifikation des Vorteilers  $m_2$  von Kanal 2 des CTC das Überstreichen des gesamten Frequenzbereiches von 0 bis  $f_{\text{max}}$ . Voraussetzung ist ein Vorteiler  $m_1$  von ca. 50H für Kanal 1 des Z80-CTC.

Es gilt dann der bereits erwähnte Zusammenhang:

$$f_{\text{out}} = m_2 \times f_{\text{in}} \quad \text{mit } f_{\text{in}} = 1/2 \Phi : m_1$$

Durch Änderung von  $m_1$  und  $m_2$  sind theoretisch insgesamt  $2^{16} = 65536$  verschiedene Ausgangsfrequenzen programmierbar.

### 7.3.7 Zusammenschluß ECB/F-FD-Laufwerke

Der Anschluß eines Laufwerkes erfolgt über die im Bild 7—7 dargestellten Leitungen.

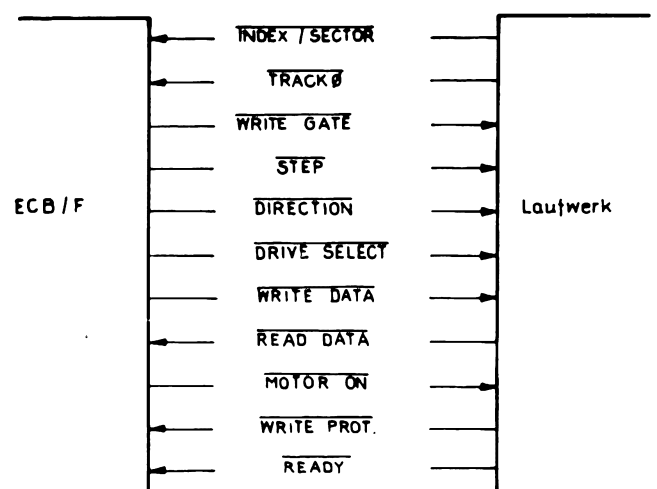


Bild 7—7: ECB/F-FD-Laufwerk-Verbindungen

**Signalbedeutungen:**

<u>INDEX/SECTOR</u>	ist aktiv wenn die Sektor- oder Indexöffnung einer Diskette abgetastet wird. Der Indexpuls tritt nur einmal pro Umdrehung auf und kennzeichnet Sektor 0.
<u>TRACK0</u>	ist aktiv, wenn der Schreib-/Lesekopf auf Spur 0 steht.
<u>WRITE GATE</u>	ermöglicht das Schreiben auf die Diskette und verhindert die Ansteuerung des Schrittmotors.
<u>STEP</u>	bewegt den Schreib-/Lesekopf um eine Spur weiter.
<u>DIRECTION</u>	bestimmt, in welche Richtung ein STEP erfolgt.
<u>DRIVE SELECT</u>	aktiviert das angesteuerte Laufwerk (Laufwerkadresse).
<u>WRITE DATA</u>	Clock/Datengemisch zum Laufwerk.
<u>READ DATA</u>	Clock/Datengemisch vom Laufwerk
<u>MOTOR ON*</u>	startet die Motoren.
<u>WRITE PROTECT</u>	die Diskette ist mechanisch schreibgeschützt.
<u>READY*</u>	Ist 5 Umdrehungen nach Einlegen einer Diskette aktiv.

\* Die Signale MOTOR ON und READY sind nicht bei allen Laufwerken vorhanden.

Alle Disk-Status-Signale (außer MOTOR ON und WRITE PROTECT) sind über PORT A des Z80-SIO mit der CPU verbunden, wobei alle Signale Schmitt Trigger gepuffert sind (außer DRIVE SELECT).

Selbstverständlich kann PORT A auch für andere Aufgaben als zur Floppy-Disk-Steuerung verwendet werden. PORT A muß dann im BIT Mode (Steuerwort CFH) mit der Maske 30H (A4, A5 sind Eingänge) betrieben werden. Die DRIVE SELECT-Signale werden durch einen 1:8 Decoder aus den PIO-Anschlüssen A0, A1, A2 erzeugt. Die Freigabe dieses Decoderbausteins (C1:7442) erfolgt über B0 von PORT B des PIO's.

Ebenfalls an PORT B befindet sich ein 6fach DIL Schalter, der beispielsweise zur Einstellung von Systemvariablen Baudrate etc.) abgefragt werden kann. Dazu muß auch PORT B im BIT Mode betrieben werden. Es gilt folgende Zuordnung:

Schalter	PIO-Anschluß PORT B
S1	B2
S2	B3
S3	B4
S4	B5
S5	B6
S6	B7

Die Schalterstellung „ON“ entspricht einer log. Null.

**7.4. Der Asynchronkanal**

**7.4.1. Schnittstellenwahl**

Kanal B des Z80-SIO ist zum Betrieb als asynchroner Serienkanal beschaltet. Es steht sowohl eine 20 mA-Stromschleifenschnittstelle, als auch eine RS232C Schnittstelle zur Verfügung. Die dazu notwendigen Spannungen werden auf der Baugruppe durch Gleichspannungswandler erzeugt. Die Auswahl der gewünschten Schnittstelle erfolgt über die beiden Kurzschlußstecker J5 und J6 entsprechend Tabelle 7.5:

Stecker	Stellung	Signal	→	Schnittstelle
J5	A	Receiver Data (RxD)	→	RS 232 C
J5	B	Receiver Data (RxD)	→	20 mA
J6	A	Transmitter Data (TxD)	→	RS 232 C
J6	B	Transmitter Data (TxD)	→	20 mA

Tabelle 7.5: Festlegung der Schnittstellen

**7.4.2. Baudrateerzeugung**

Über Kanal 0 des Z80-CTC sind alle gebräuchlichen Baudraten von 50 bis 38400 Baud programmierbar. Dies geschieht entweder im Counter Mode durch Teilung des halben Systemtaktes (= 1,2288 MHz) oder im Timer Mode.

Die Zeitkonstante des CTC muß jeweils so gewählt werden, daß das Doppelte der gewünschten Baudrate entsteht. Flipflop E5 erzeugt aus diesem Signal ein symmetrisches Signal, das unmittelbar dem Eingang RxTxCB des Z80-SIO zugeführt wird. Der Vorteiler m des Z80-CTC läßt sich abhängig von der programmierten Betriebsart nach folgenden Formeln berechnen:

$m = \frac{1/2 \cdot \Phi}{2 \cdot x \cdot BR}$	Counter Mode
$m = \frac{\Phi}{2 \cdot x \cdot y \cdot BR}$	Timer Mode

mit:

x = 16 oder 64 (SIO-Übertragungsrate × 16 oder × 64)

y = 16 oder 256 (CTC-Vorteiler im Timer Mode)

BR = gewünschte Baudrate

Φ = Frequenz des Systemtakts (Z80-ECB/C: 2,4576 MHz)

Beispiel: BR = 300; Timer Mode für CTC

x = 64

$$m = \frac{1/2 \cdot 2,4576 \cdot 10^6}{2 \cdot 64 \cdot 300} = 32$$

Tabelle 7.6 enthält eine Zusammenstellung aller möglichen Baudraten und der dazu erforderlichen CTC-Zeitkonstanten.

Baudrate BR	m für Counter Mode		m für Timer Mode (y = 16)	
	× 16	× 64	× 16	× 64
50	—	192	96	24
75	—	128	64	16
110	—	87	44	11
134,5	—	72	36	9
150	256	64	32	8
200	192	48	24	6
300	128	32	4	1
600	64	16	2	—
1200	32	8	4	—
2400	16	4	2	—
4800	8	2	1	—
9600	4	1	—	—
19200	2	—	—	—
38400	1	—	—	—

Tabelle 7.6: Zusammenhang zwischen Baudrate und CTC-Vorteiler m.

**7.4.3. Anschluß von Terminals**

Abhängig von der ausgewählten Schnittstelle (Tabelle 7.5) erfolgt der Anschluß eines Terminals (Datensichtgerät, Fernschreiber) über eine 3- bzw. 4-Drahtverbindung entsprechend den Prinzipschaltungen in Bild 7—8.

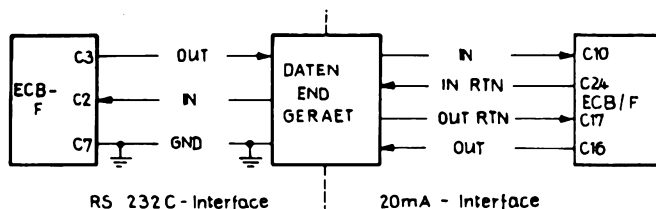


Bild 7—8: Anschluß von Terminals

Alle übrigen Steuer-Ein- bzw. -Ausgänge des Z80-SIO stehen nur im RS 232 C Pegel zur Verfügung (siehe auch Abschnitt 7.5 — Steckerbelegung).

#### 7.4.4. Programmierhinweise

Bevor die asynchrone Serienschnittstelle betriebsbereit ist, müssen Z80-CTC (Kanal 0) und Z80-SIO (Kanal B) entsprechend programmiert werden.

a) CTC-Programmierung (Kanal 0)  
 LD A, CTRMOD ;  
 OUT (KANAL 0), A ; KANAL 0 = Counter Mode  
 LD A, m  
 OUT (KANAL 0), A ; Zeitkonstante m Laden  
 ; (Tabelle 5)  
 CTRMOD EQU 0 1 0 0 0 1 1 1 B; (= 47 H)

b) SIO-Programmierung (Kanal B)  
 Der Z80-SIO bietet sehr komplexe Programmiermöglichkeiten. Einzelheiten sind der SIO-Produkt-Spezifikation bzw. dem Z80-Datenbuch zu entnehmen. Im folgenden einige Programmierhinweise zur Initialisierung und zur Kommunikation zwischen SIO und CPU.

Beispiel einer Initialisierungsroutine nach einem Hard- oder Softwarereset. (Anwendung der Z80 Block-I/O-Transferbefehle).

LD HL, TABELLE ; Tabellenadresse  
 LD C, KANAL B + 2 ; Kanaladresse (Controlport)  
 LD B, ANZAHL ; Anzahl der zu übertragenden Bytes  
 OTIR ; Blockausgabe

TABELLE:

DEFB 003H ; Zeiger auf WRITE REG. 3  
 DEFB 0C1H ; 8 Bits pro empfangenes Zeichen  
 ; Empfänger enabled  
 DEFB 004H ; Zeiger auf WRITE REG. 4  
 DEFB 044H ; × 16 Clock, 1 Stop Bit/Zeichen,  
 ; Kein Paritybit  
 DEFB 005H ; Zeiger auf WRITE REG. 5  
 DEFB 068H ; 8 Bits pro gesendetes Zeichen  
 ; Sender enabled  
 DEFB 001H ; Zeiger auf WRITE REG. 1  
 DEFB 008H ; Empfängerinterrupt nur beim  
 ; ersten Zeichen, kein Sender-  
 ; interrupt  
 DEFB 002H ; Zeiger auf WRITE REG. 2  
 DEFB VECTOR ; Interruptvektor laden

ANZAHL EQU \$-TABELLE

WRITE REG. 1 und 2 müssen natürlich nur programmiert werden, wenn mit Interrupts gearbeitet werden soll.

Die Kommunikation zwischen CPU und SIO erfolgt oftmals im Polling-Verfahren. Hierbei muß beim Lesen zunächst das Bit D0 (Receive Character Available) von READ REG. 0 in einer Schleife solange abgefragt werden, bis es gesetzt ist. Entsprechendes gilt für den Sendevorgang. (Bit D2 von READ REG. 0).

Beispiele: Unterprogramme zur Ein-/Ausgabe über den Akkumulator.

SIO IN: IN A, (KANAL B + 2) ; READ REG. 0 lesen  
 BIT D0, A ; Zeichen vorhanden?  
 JR Z, SIO IN ; Endlosschleife bis Zeichen  
 ; vorhanden  
 IN A, (KANAL B) ; Zeichen einlesen  
 RET

SIO OUT: PUSH AF ; Akku retten  
 IN A, (KANAL B + 2) ; READ REG. 0 lesen  
 BIT D2, A ; Senderpuffer leer?  
 JR Z, SIO OUT+1  
 POP AF  
 OUT (KANAL B), A ; Akku ausgeben  
 RET

#### 7.5. Ein-Ausgabe-Steckerbelegung

Alle Ausgangssignale der ECB/F sind an drei Pfostensteckverbindern herausgeführt:

Stecker A	A1	INDEX/SECTOR
(26polig)	A2	DRIVE 0
	A3	DRIVE 2
	A4	DRIVE 2
	A5	MOTOR ON
	A6	DIRECTION
	A7	STEP
	A8	DISK WRITE DATA
	A9	WRITE GATE
	A10	TRACK 0
	A11	WRITE PROTECT
	A12	DISK READ DATA
	A13—A26	GND

Stecker B	B1	DRIVE 3
(10polig)	B2	DRIVE 4
	B3	DRIVE 5
	B4	DRIVE 6
	B5	DRIVE 7
	B6	EXT. CLOCK
	B7	SPARE INPUT.
	B8	SPARE OUTPUT

Stecker C	C1	—
	C2	RS 232 DATA IN
	C3	RS 232 DATA OUT
	C4	DCD
	C5	CTS
	C6	RTS
	C7	COMMON GND
	C8	DTR
	C9	—
	C10	20 mA DATA IN
	C11—C15	—
	C16	20 mA DATA OUT
	C17	20 mA OUT RTN
	C18—C23	—
	C24	20 mA IN RTN
	C25—C26	—

#### Zählweise:

Die Zählweise der Steckerleistenpins entspricht Bild 7.9.

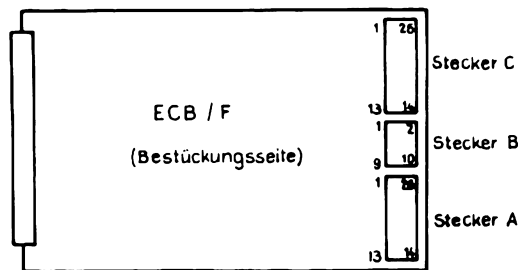


Bild 7—9: Steckeranordnung und Zählweise. Beachten Sie bitte die abweichende Zählweise bei Stecker B.

#### 7.6 Jumperstellungen

Die Baugruppe enthält insgesamt 6 steckbare Kurzschlußbügel (Jumper J1—J6), von denen J5 und J6 bereits erwähnt wurden (Tabelle 7.5).

Die Jumper J2 bis J4 dienen dazu, andere synchrone Peripherie an Kanal A des Z80-SIO anzuschließen. Durch Umstecken der Stecker J2, J3, J4 auf Stellung B wird die Floppy Disk spezifische Hardware (Datenencoder, Datensseparator) umgangen. Die SIO-Eingänge R×CÄ und R×DA führen durch Umstecken von J2 bzw. J3 unmittelbar die an den

Steckerpins B6 bzw. A12 anliegenden Signale (EXT. CLOCK RECEIVE DATA). Wird J4 umgesteckt, so ist der Anschluß T×DA über den Puffer B2 mit Stecker A, Pin 8 (TRANSMIT DATA) verbunden.

Mit Jumper J1 wird der WRDY-Ausgang des SIO-Kanals B mit dem entsprechenden Ausgang von Kanal A verbunden

(beides Open Drain Anschlüsse) und über einen Open-Collector-Puffer (F6: 7407) an die CPU-WAIT-Leitung angeschlossen. Damit läßt sich eine Synchronisation zwischen CPU und SIO erreichen (siehe SIO-Produktspezifikation).

Bild 7—9 zeigt die Lage der einzelnen Jumper auf der Bau-  
gruppe (von der Bestückungsseite gesehen).

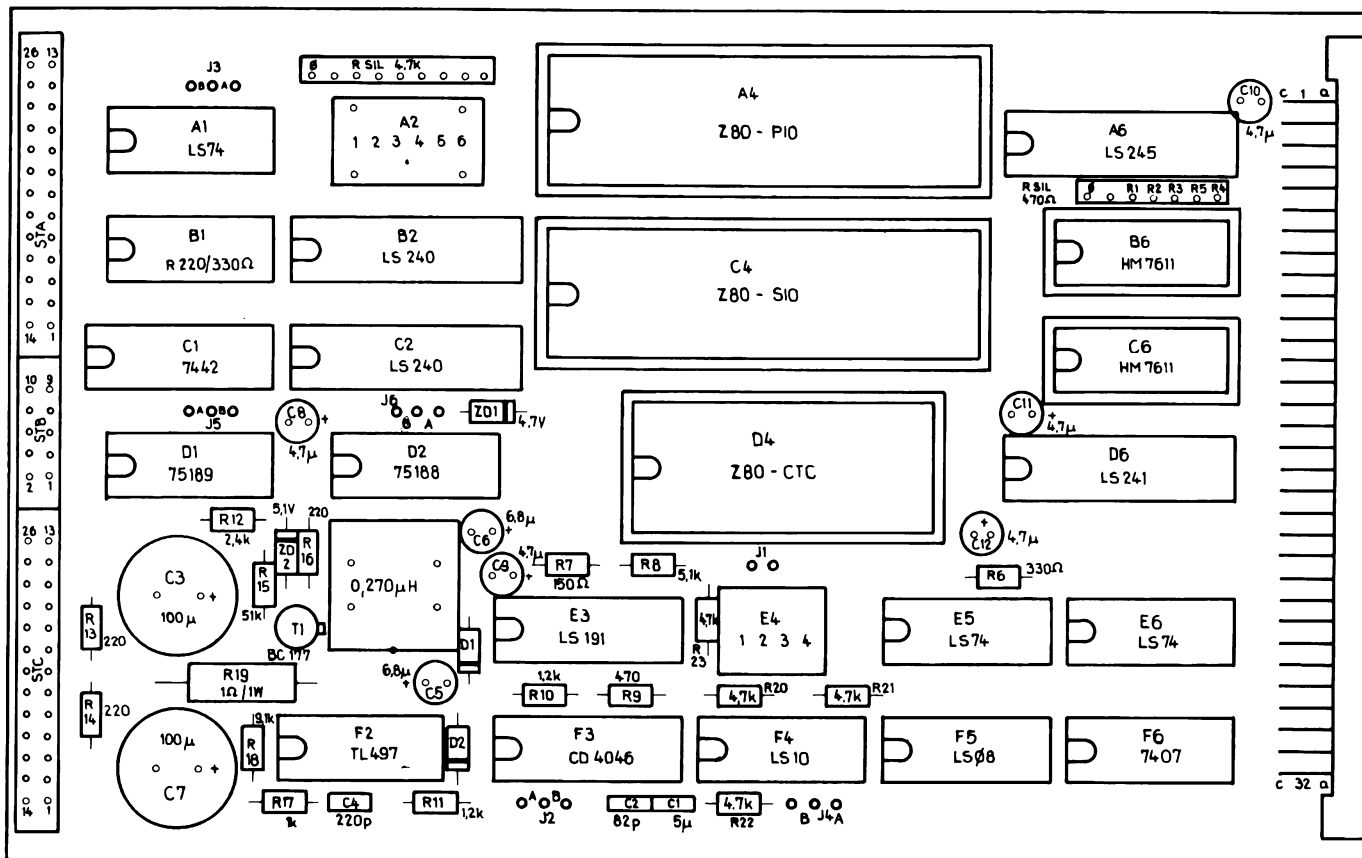


Bild 7—10

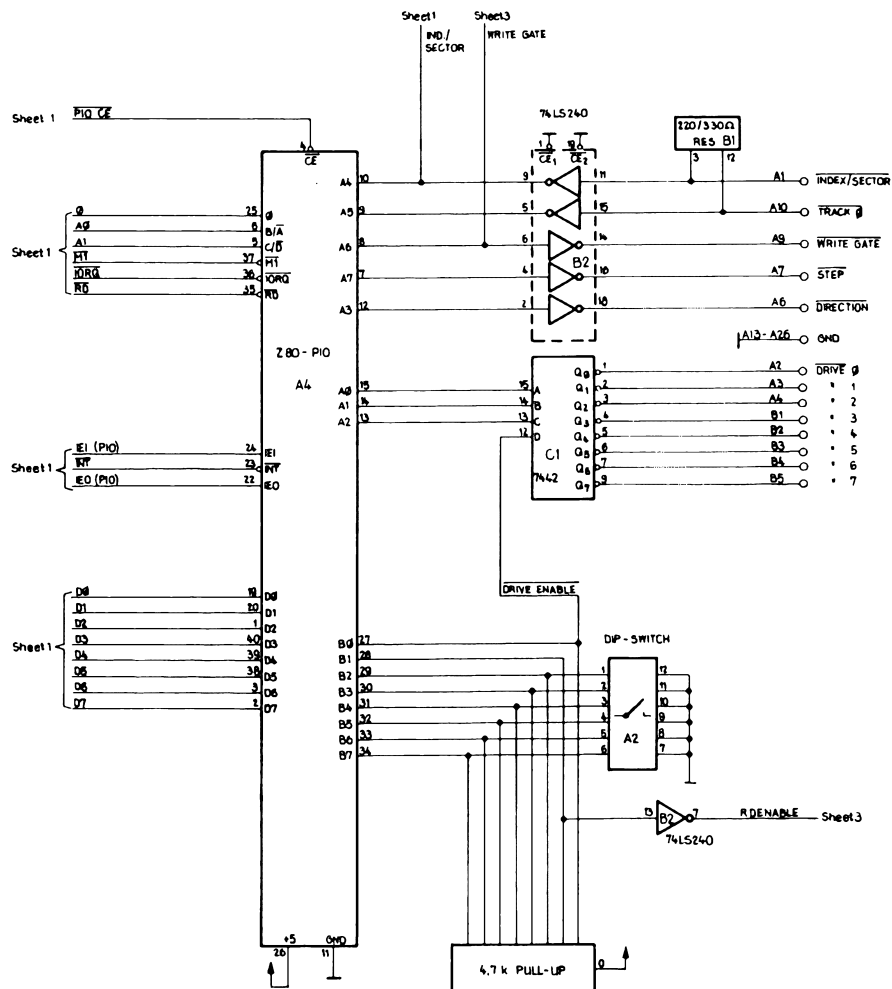
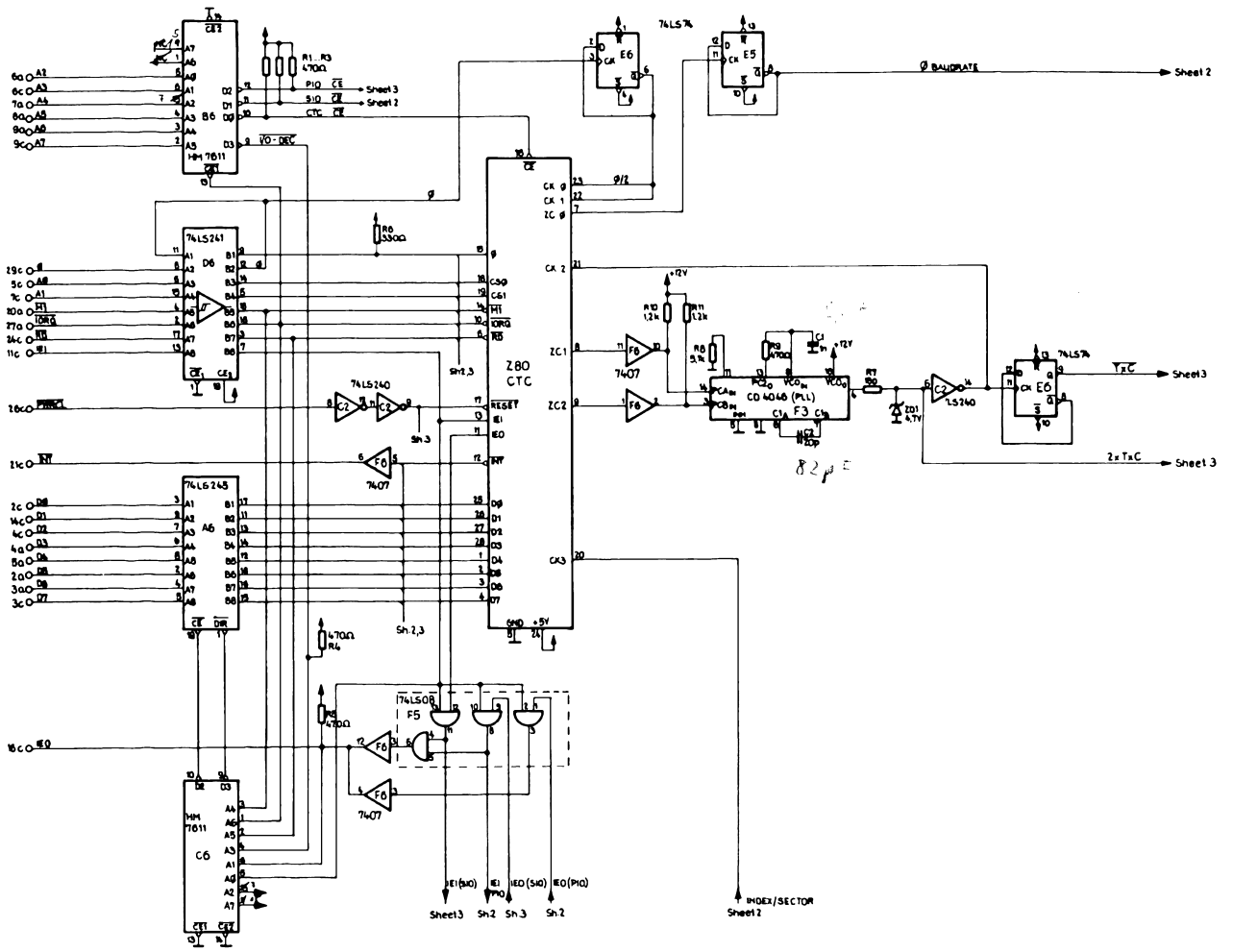
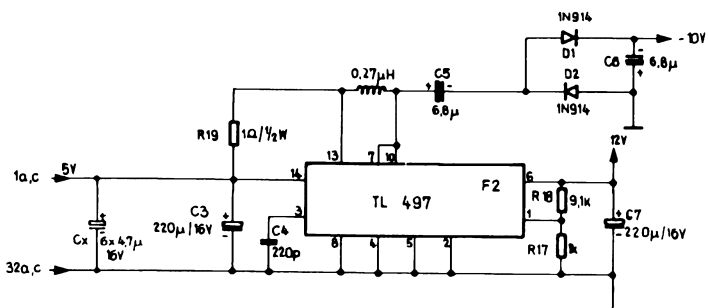
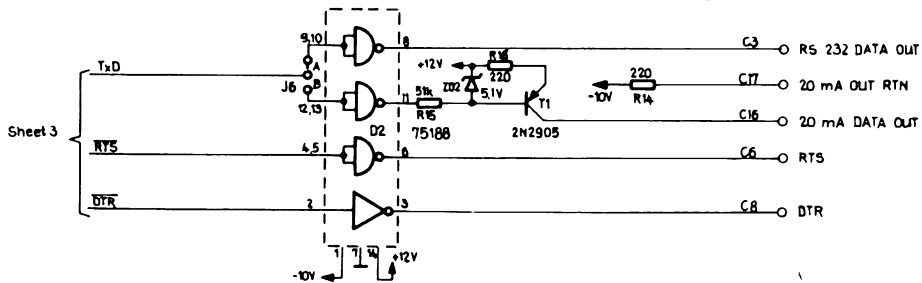
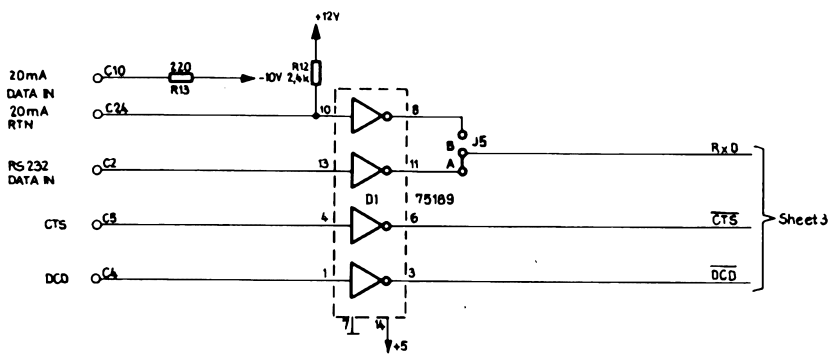
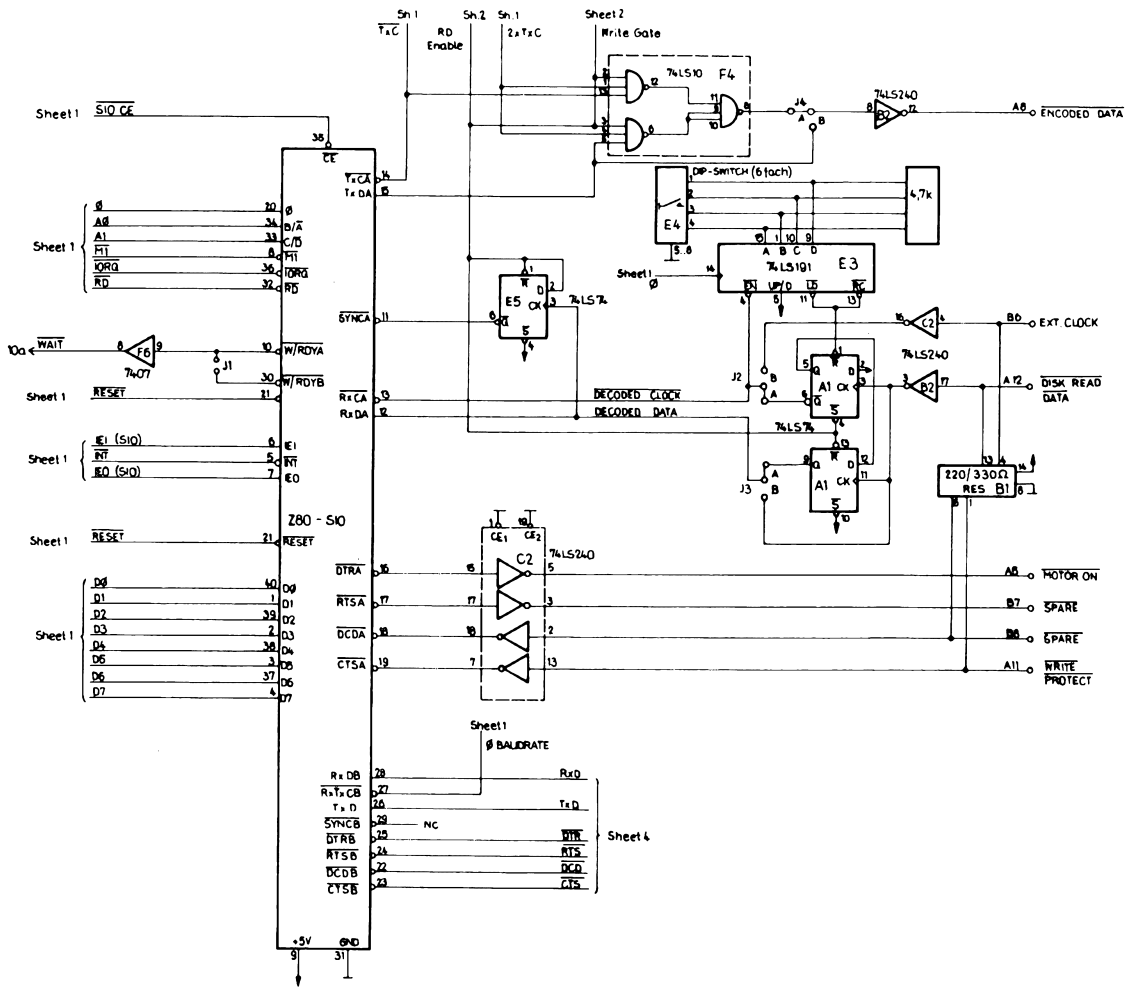


Bild 7-11  
Blatt 1



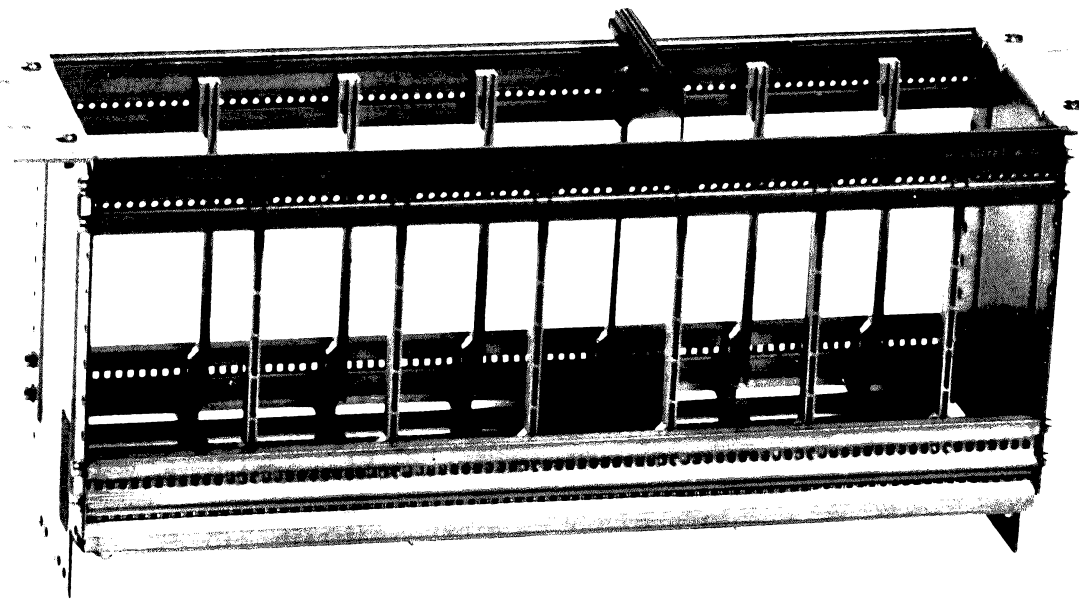




**Z80-ECB/R**



**Baugruppenträger**



## 8. Der Baugruppenträger Z80-ECB/R

Das 19 Zoll-Einbaureck ist zur weiteren Vereinfachung des Aufbaus ECB-basierender anwenderspezifischer Computerschaltungen gedacht. Es wird fertig montiert geliefert und besteht aus dem Baugruppenträger selbst und einer zweiseitig kaschierten 1:1-Rückwandverdrahtungsplatine, die insgesamt 20 Steckplätze umfaßt. Von diesen 20 möglichen Steckplätzen sind bei der Auslieferung 6 mit VG-Messerleisten bestückt. Nachrüstätze zur Bestückung von je 6 weiteren Steckplätzen sind unter der Bezeichnung Z80-ECB/R6 lieferbar. Auf dieser Platine ist auch bereits die folgende Schaltung vorgesehen, die das Einfügen eines WAIT-Zyklus nach jedem M1-Zyklus besorgt (Bild 8-1). Dadurch ist ohne weitere Zusätze das Einsetzen von Baugruppen möglich, die mit langsamen (und damit preisgünstigeren) Speicherbausteinen bestückt sind.

Die Verarbeitungsgeschwindigkeit des Gesamtsystems wird dadurch natürlich in diesem Fall entsprechend herabgesetzt. Abschließend soll hier noch die Verdrahtungsweise der einzelnen Pins angegeben werden:

Pin Nr.	Verdrahtet mit	Bedeutung
16c bzw. 17a	verdrahtet mit Pin 11c bzw. 12a des nächstfolgenden Steckplatzes. Der nächstfolgende Steckplatz ist der nächste, links vom aktuellen Steckplatz liegende (von der Lötseite der Rückwandverdrahtung aus gesehen), in aufsteigender Reihenfolge der alphabetischen Steckplatzkennzeichnung.	Daisy Chain für Interrupt-IEI (11c)/IEO (16c) bzw. für DMA-Priorisierung BAI (12a)/BAO (17a). Zur Beachtung: Werden Steckplätze zwischen zwei Baugruppen frei gelassen oder aber Baugruppen eingefügt, die keine Verbindung zwischen Chain-Eingang und Chain-Ausgang haben, so muß die so entstandene Unterbrechung der Daisy Chain durch eine Drahtbrücke auf der Verdrahtungsplatine überbrückt werden.
Pin 26a	reserviert für ECB/D	Schreib/Lesesignal WRITE EN
Pin 19a, Pin 16c	reserviert für AN- $\mu$ P80/E 16SE	Analog-Ein/Ausgabeplatine, $\pm 15$ V
Pin 24c		führt die Batteriespannung des ECB/V auf den Bus heraus. Die Power-Fail-Logik auf der Busverdrahtungsplatine wird mit dieser Spannung versorgt.
Pin 23a	reserviert für ECB/V	Auf dieser Leitung liegt das von der auf der Busverdrahtungsplatine erzeugte Signal zum Abschalten des CMOS-Speichers im Power-Fail-Fall.

Außer Pin Nummer 16c, 17a sind alle Pins gleicher Nummer sämtlicher Steckplätze miteinander verbunden (= 1:1 verdrahtet).

74 LS 112

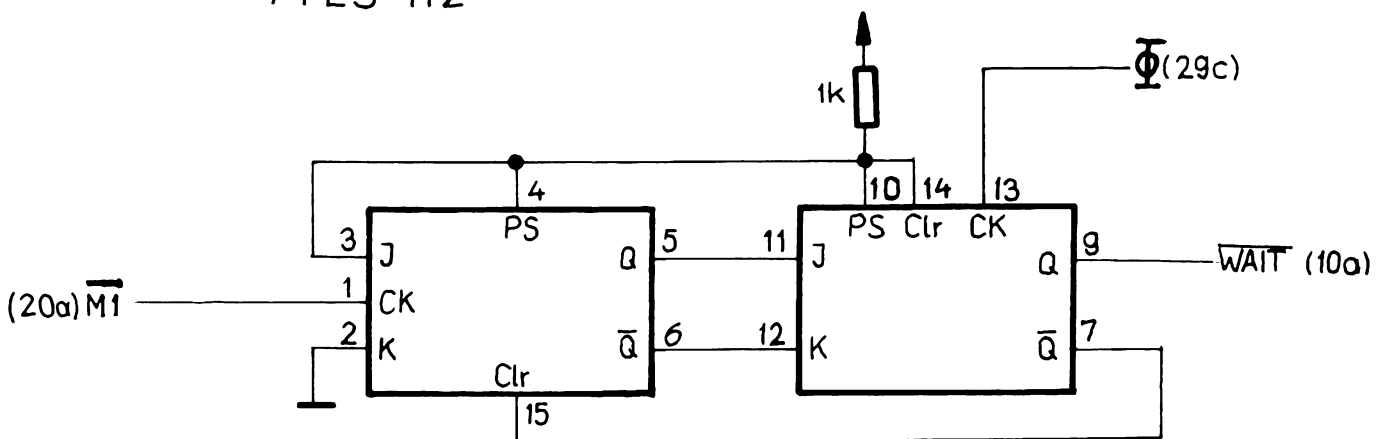


Bild 8-1

## 9. Inbetriebnahme

Zur Inbetriebnahme einer Z80-ECB-Hardware sind lediglich folgende Maßnahmen nötig:

- Zusammenstecken der ausgewählten Baugruppen
- Einsetzen der Monitor-PROM's
- Einstellen der Speicherbereichs-Wahlschalter
- Anschließen der Systemkonsole
- Anlegen der Speisespannung
- evtl. Zuschaltung von Peripherie an die Parallel-Schnittstellen.

Danach kann der Computer bereits in Betrieb genommen bzw. programmiert werden.

### III. Softwareteil

## 1. Betriebsprogramme ECB/12 und ECB/1

### 1.1 Einführung

Für die Z80-Baugruppen im Einfacheuropaformat (160 × 100 mm) sind Monitor-Programme im Umfang von 1/2 kByte und 1 kByte verfügbar, wobei die 1/2 kByte-Version über 5 Kommandos, die 1 kByte-Version über 8 Kommandos verfügt (zusätzliche Möglichkeit der Registeranzeige/Modifikation und des Setzens von Haltepunkten).

Das Programm ECB/3 dient zur Ansteuerung von Floppy-Disk-Systemen mit ECB/F. Die Monitor-Programme stellen beide jeweils ein einfaches Betriebssoftwarepaket dar, das als Firmware (also in einem Festwertspeicher gespeichertes Programm) geliefert wird; das PROM kann direkt in die hierfür auf der ECB-Baugruppe montierte Fassung eingesteckt werden. Der Monitor erlaubt die sofortige Inbetriebnahme des ECB-Computers und erspart dem Anwender die Erstellung gewisser Basis-Routinen, wie Serien-Ein/Ausgabe usw. Darüber hinaus ist mit dem Monitor sogar die Durchführung von Programm-Entwicklungen und -Tests möglich. Die verfügbaren Monitor-Kommandos sind in Punkt 3 beschrieben; die nur in der 1 kByte zulässigen Kommandos sind GO, BREAK und REGISTER.

Z80-ECB/12: 1/2 kByte-Monitorprogramm  
Z80-ECB/1 : 1 kByte-Monitorprogramm  
Z80-ECB/3 : 3 kByte-Monitorprogramm  
Z80-MTX : Echtzeit-Multitask-Betriebssystem

Darüberhinaus sind Standardsoftwarepakete mit arithmetischen Operationen und Wandlungsroutinen verfügbar (Z80-SAP, Z80-FPA, Z80-EAP).

### 1.2 Hardware-Erfordernisse des Monitors

#### 1.2.1 Speicherorganisation

Der 1 kByte Monitor ist in dezimal 1024 (bzw. hexadezimal 400) Festwertspeicherstellen abgelegt. Er benötigt darüber hinaus 35 Byte Schreib/Lesespeicher zuzüglich einem Stack-Bereich im RAM.

Dieser Schreib/Lesespeicher liegt am oberen Ende der vierten Seite, also von Adresse 3CFFH abwärts. (Eine Seite entspricht 4 kByte). Der darauffolgende Stack-Speicherbereich beginnt ab Adresse 3CD8H und wächst in Richtung niederwertiger Speicheradressen. Über den übrigen Speicherbereich kann frei verfügt werden.

Der 1/2 kByte-Monitor belegt entsprechend 512 (= 200H) Speicherstellen; er benötigt abgesehen vom Stack-Bereich, der von Speicherstelle 3CFFH ab abwärts wächst, kein RAM.

#### 1.2.2 Anschluß eines Bedienungs-Terminals

Über die 20 mA-Schnittstelle des ECB/C kann das Monitorprogramm mit jedem beliebigen Terminal verkehren, das 8 bit-ASCII-Code asynchron mit oder ohne Parity-Bit seriell überträgt. Es sind 14 verschiedene Übertragungsraten zugelassen, die auf der ECB/C-Baugruppe über die 4 Schalter einstellbar sind.

Soweit die Fähigkeit des Monitors zur Dateiausgabe bzw. Dateieingabe genutzt werden soll, muß das Terminal über ein Datenspeichermedium verfügen, das softwaremäßig einer Lochstreifen-Leser/Stanzer-Kombination entspricht. Als Start- und Stoppsignal muß dieses Kommunikationsmedium die ASCII-Zeichen 11H und 13H erkennen können, oder an geeigneter Stelle manuell zu starten oder zu stoppen sein.

### 1.3 Beschreibung des Monitor-Programms

#### 1.3.1 Verhältnis zu evtl. Anwenderprogrammen

Anwenderprogramme, die zusammen mit dem Monitor eingesetzt werden, können auf eine Reihe von dessen Unterprogrammen zugreifen.

#### 1.3.2 Formale Regeln

Sämtliche Kommandos können durch ihren ersten Buchstaben abgekürzt oder auch bis zu jeder beliebigen Länge ausgeschrieben werden, da lediglich der erste Buchstabe in einer neuen Eingabe-Zeile als Schlüsselzeichen für ein empfangenes Kommando benutzt wird. Nicht verstandene Kommando-Eingaben werden vom Monitor durch Ausgabe des ASCII-Zeichens für ? quittiert, worauf eine neue Kommando-Eingabe erwartet wird. Sämtliche Felder werden durch Leerzeichen begrenzt (= "Delimiters").

Außer den Daten im SET-Kommando können alle Zahlen formatfrei (= "free form") hexadezimal (auch ohne führende Nullen) eingegeben werden. Die als ein Datum eingebare Zahl ist auf 4 Zeichen begrenzt; werden mehr als 4 Zeichen eingegeben, werden die letzten 4 Zeichen als gültige Zahl interpretiert.

#### 1.3.3 Beschreibung der einzelnen Kommandos

**DISPLAY** adr n

veranlaßt die Ausgabe des Inhalts der n folgenden Speicherstellen ab (inklusive) der Speicherstelle „adr“.

**SET** adr dat1 dat2 . . . . datm

speichert die angegebenen Datenwörter (jedes als 2 Hexadezimalzeichen einzugeben!!) ab (inklusive) der Speicheradresse „adr“ ab.

Unmittelbar auf das letzte Zeichen von datm muß ein Wagenrücklaufzeichen ("carriage return") eingegeben werden.

**PUNCH** anfadr endadr

gibt den Speicherinhalt ab (inklusive) „anfadr“ bis (inklusive) „endadr“ auf den Lochstreifenstanzer der Kommandokonsole aus, wobei der Stanzvorgang automatisch durch Ausgabe eines „Tape-on-Zeichens“ (12 H ASCII) gestartet und durch Ausgabe von „Tape-off“ (14 H ASCII) beendet wird. Nach „Tape-off“ wird noch das Stanzen eines Stücks Leerstreifen veranlaßt.

**LOAD**

Einlesen von Daten von der Lochstreifeneinheit der Kommandokonsole in den Speicher, wobei die Anfangsadresse auf dem Lochstreifen eingestanzt sein muß und der Lochstreifen natürlich vorher (und zwar ein kleines Stück vor Vorkommen des ersten lesbaren Zeichens) in den Leser eingelegt werden muß.

Durch ein "Reader-On"-Signal wird der Leser vom Programm automatisch gestartet und am Ende der eingelesenen Datei wieder abgeschaltet.

Im Fehlerfall (= Check-Summe nicht richtig) wird der Leser ausgeschaltet und eine Nachricht ausgegeben, da anzunehmen ist, daß Daten verloren gegangen sind.

**JUMP** adr

veranlaßt einen unbedingten Sprung auf Adresse adr und Ausführung des dort befindlichen Programms.

**GO\***

veranlaßt Fortsetzung des Programms nach einem Breakpoint.

**BREAK** adr\*

setzt einen Haltepunkt bei Speicheradresse adr, wobei zuvor evtl. vorhandene andere Haltepunkt-Bedingungen außer Kraft gesetzt werden.

## REGISTER reg\*

Mit diesem Kommando können CPU-Registerinhalte ausgegeben und modifiziert werden.

Auf die Angabe des Registernamens hat statt des Wagenrücklaufzeichens ein Leerzeichen (Blank) zu folgen; vom Monitor wird daraufhin der Registerinhalt in der gleichen Zeile ausgegeben, worauf der Benutzer die Ausführung des Kommandos mit der Eingabe eines Wagenrücklaufzeichens abschließen kann.

Soll stattdessen das nächstfolgende Register ausgegeben werden, muß der Benutzer statt des Wagenrücklauf- ein Zeilenvorschub-Zeichen eingeben.

Soll der Inhalt eines eben angezeigten Registers modifiziert werden, ist unmittelbar nach einem weiteren Leerzeichen der gewünschte Registerinhalt einzugeben. Danach beendet wieder ein Wagenrücklauf- oder Zeilenvorschub-Zeichen den Vorgang.

Die Registerinhalte werden von RAM-Speicherstellen aus angezeigt, in denen sie vorher vom Monitor-Programm abgelegt werden.

Die Reihenfolge, in der die Register bei fortlaufender Eingabe von Zeilenvorschubzeichen in REGISTER-Kommandos behandelt werden ist folgende:

A, B, C, D, E, F, H, L, I, A', B', C', D', E', F', H', L', PC, SP, IX, IY.

### 1.3.4 Haltepunkt-Behandlung (nur beim 1K-Monitor!)

Die Software-Haltepunkte des Z80-Monitors dienen zur Fehlersuche und -Beseitigung in Anwenderprogrammen. Wird eine Haltepunktbedingung bei der Ausführung des Anwenderprogramms als gültig erkannt, wird diese Ausführung abgebrochen, alle Registerinhalte in den hierfür vom

Monitor reservierten Speicherbereich kopiert und eine Meldung ausgegeben, daß und bei welcher Speicherstelle der Haltepunkt erreicht wurde.

Besonders zu beachten ist, daß man den Stackpointer initialisieren muß, wenn man ein Anwenderprogramm ablaufen lassen und durch das Kommando BREAKPOINT behandeln will.

Dies kann geschehen

- durch Initialisierung im Anwenderprogramm selbst oder
- durch direktes Setzen des CPU-Stackpointers durch das Kommando REGISTER

Eine beliebige Anzahl von Haltepunkten kann manuell gesetzt werden, indem als Haltepunktbedingung auf der gewünschten Adresse die Daten 0FFH eingegeben werden.

Damit der Haltepunkt richtig behandelt wird, ist es notwendig, den Stackpointer im User-Programm an eine Stelle im gültigen RAM-Bereich zu setzen, die **nicht** durch die Monitor-Routinen benützt werden. Dies geschieht entweder durch Setzen des SP am Anfang des User-Programms, oder durch Modifizieren des Registerinhalts SP. Günstige Adressen für den ECB 1K-Monitor sind ab 3CD0H abwärts.

Beliebig viele Haltepunkte können manuell durch Angabe der Haltepunktadresse 0 FFH gesetzt werden. Dabei muß die Haltepunktadresse auf das erste Byte einer Anweisung zeigen, und die ursprüngliche Anweisung muß von Hand wieder auf diesen Speicherplatz geschrieben werden, sobald dieser Haltepunkt nicht mehr benötigt wird.

Hintergrund dieses Verfahrens ist die Tatsache, daß der Monitor grundsätzlich anhält, wenn er annimmt, daß er auf eine physikalisch nichtexistierende Speicherstelle (die den Inhalt FFH = RST 38H-Befehl hat) zugreift.

---

\* Bitte beachten Sie: Diese Kommandos sind nur im 1K-Monitor enthalten.

## **2. Floppy-Disk-Treibersoftware Z80-ECB/3**

Zur Ansteuerung von Floppy-Disk-Laufwerken über die Baugruppe Z80-ECB/F ist eine Treibersoftware in Vorbereitung, die die Verwendung von Hardsektorierten Floppy-Disks (und zwar Single-Density, Double Density und Minifloppy) erlaubt.

Nähere Informationen sind in einer getrennten Produktspezifikation erhältlich.



# Z80-MTX



# Echtzeit-Multitask- Betriebsprogramm

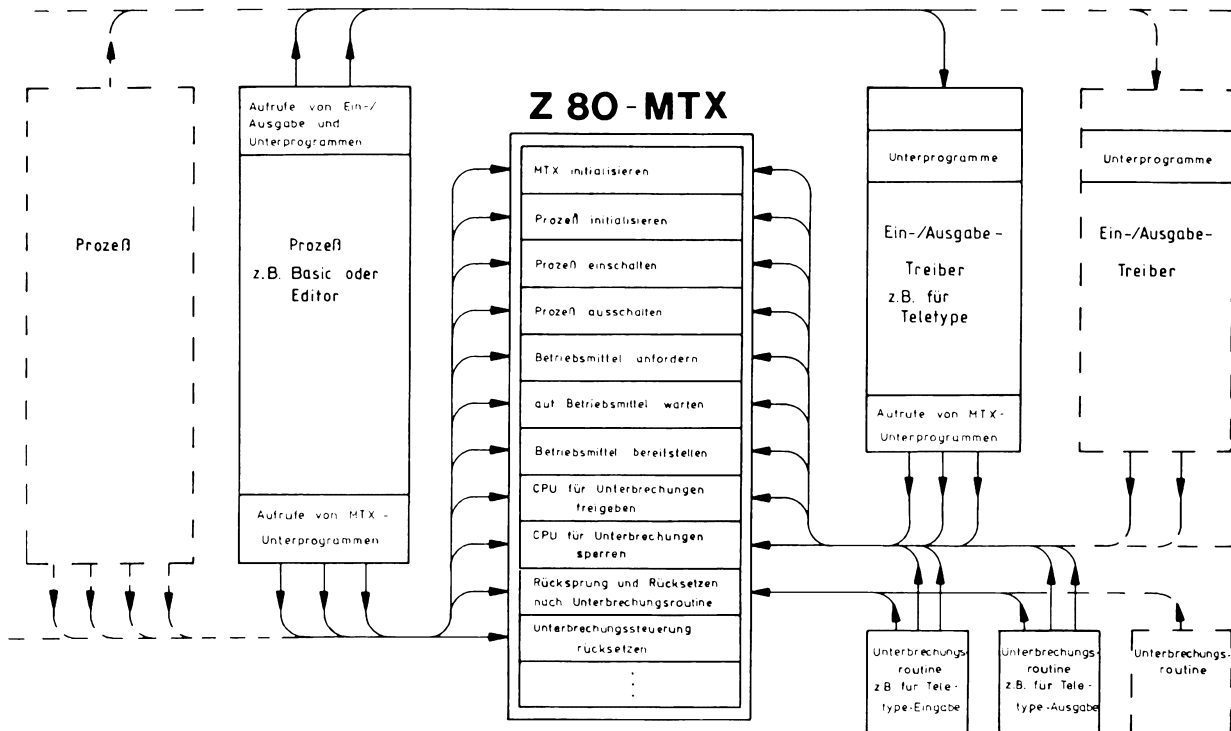


Bild 3—1

# 3. Echtzeit-Multitask-Betriebsprogramm

Für die ständig erweiterte Z80-ECB-Baugruppen-Serie ist jetzt ein Betriebssystem lieferbar, das auf der besonders leistungsfähigen Interrupt-Architektur des Z80-Konzepts basiert und sie ausnutzt.

Z80-MTX ist ein Betriebssystem zur Verwaltung einer beliebigen Anzahl voneinander abhängiger und asynchron verlaufender Vorgänge. Jeder Vorgang (= „Prozeß“) kann Ein/Ausgabe-Aktivitäten, Rechnungen und logische Entscheidungen enthalten.

Das Betriebssystem überwacht den Ablauf der durch Unterbrechungen (Interrupts) gesteuerten Ein- und Ausgaben und weist den Prozessor nach einer Prioritätenliste Rechenzeit zu. Dadurch können z.B. Ein/Ausgabe-bedingte Wartezeiten eines bestimmten Prozesses zur Fortführung anderer Prozesse genutzt werden.

Ebenso koordiniert Z80-MTX logische Verknüpfungen der Prozesse untereinander.

Das Betriebssystem hat folgende hervorstechende Eigenschaften:

- **Multitask**  
Mehrere Aufgaben, die asynchron anfallen und per Interrupt aufgerufen werden, werden unter Z80-MTX bearbeitet.
- **Erweiterbar**  
Jede Benutzerroutine kann in einfacher Weise an das Betriebssystem angeschlossen werden.
- **Priorisierbar**  
Die Rangordnung der Aufgaben wird durch den Benutzer festgelegt.
- **Eindeutigkeit**  
Sämtliche Benutzer- und Prozeß-Parameter sind in definierbaren Tabellen festgelegt.
- **Hohe Geschwindigkeit**  
Standardisierte Interrupt-Service-Routinen führen lediglich Eintragung in Prozeßtabellen aus. Die Verarbeitungsgeschwindigkeit kann daher sehr groß sein.
- **Standards**  
Fertige Ein/Ausgabe-Routinen, wie Terminals-Drucker, Printer-Driver, Floppy-Disk-Driver, sind jetzt oder in Kürze verfügbar.
- **Kompakt**  
Z80-MTX benötigt durch den leistungsfähigen Befehlsatz der Z80-CPU einen wesentlich geringeren Programm-Speicherbereich als vergleichbare Programme, die auf anderen Prozessoren basieren. Das gesamte Betriebssystem findet in 1 kByte Platz.

Das Betriebssystem ist auf Diskette oder als Firmware (= PROM-resident) lieferbar.

## 3.1. Allgemeines

Z80-MTX ist ein Multitask-Echtzeit-Betriebssystem, das speziell für Mikroprozessor- und Minicomputer-Anwendungen entwickelt wurde. Multitask-Manager werden sonst im allgemeinen nur für Großrechner und größere Prozeßrechner angeboten. Trotzdem konnte hier ein System geschaffen werden, das auch zur Lösung kleiner und kleinster Probleme geeignet ist.

MTX beschränkt sich auf wichtige Funktionen und ist vor allem benutzerfreundlich konzipiert (nur einige wenige Systemaufrufe müssen vom Benutzer beherrscht werden). Sie sind an Hand dieser Beschreibung leicht zu erlernen.

Oft ist der Wunsch nach kurzen Reaktionszeiten ein entscheidender Grund für die Anwendung eines Mikroprozessors. Daher wurde MTX vor allem zeitoptimiert. Nur etwa 0,2 Millisekunden (bei 4 MHz Takt) werden für die längste Systemaktion benötigt, wobei der Durchschnitt wesentlich

niedriger liegt. Dennoch ist der Speicherbedarf – 1/2 KByte für Programm und 1/4 KByte für Systemtabellen – äußerst gering.

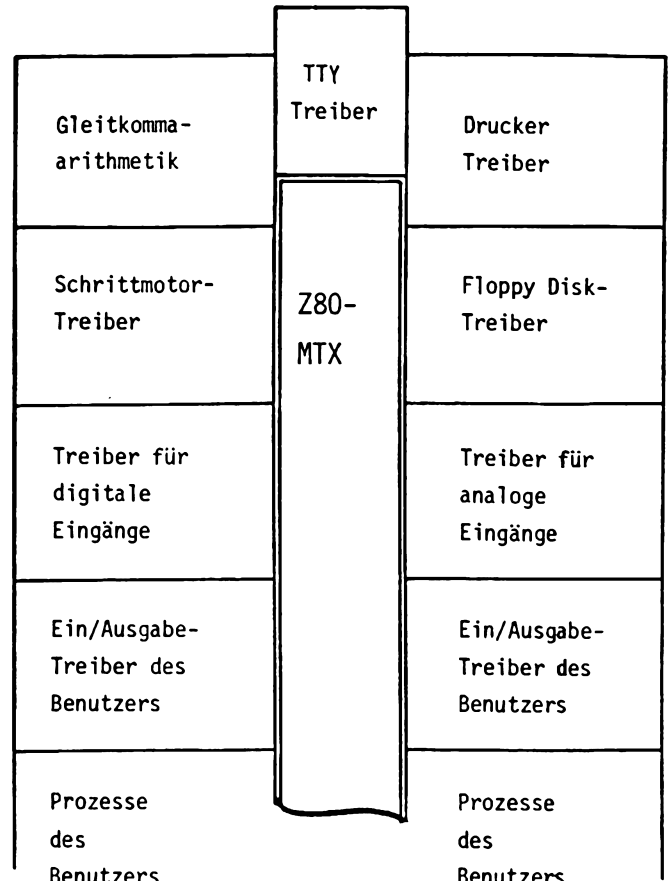


Bild 3-2

## 3.2. Aufgabe

Das Multitask-Echtzeit-Betriebssystem Z80-MTX (Multi-Task-Executive) ermöglicht die simultane Abarbeitung mehrerer Teilaufträge unter Berücksichtigung ihrer Priorität und Rechenwilligkeit. Diese Teilaufträge werden im folgenden Prozesse genannt; es kann sich dabei um Benutzer- oder Systemprogramme handeln. Der Systemprogrammierer, d.h. der Benutzer von MTX, muß seine Gesamt-Aufgabe in geeignete Prozesse untergliedern und die Abhängigkeiten der Prozesse untereinander und von Ein/Ausgabe-Aktivitäten analysieren. Entsprechend diesen logischen Abhängigkeiten sind Betriebsmittel vorzusehen. (Der Begriff „Prozeß“ wird in Abschnitt 3.4.2. näher erläutert, der Begriff „Betriebsmittel“ in 3.4.2.5).

Zur Definition der Prozeß- und Betriebsmittelkonfiguration fügt der Benutzer standardisierte Unterprogramm-Aufrufe für MTX-Routinen in seine Programme ein (siehe auch Bild 3-1).

Durch MTX werden in optimaler Weise die I/O-Wartezeiten von Prozessen hoher Priorität genutzt, um Prozesse niedriger Priorität zu bearbeiten. MTX regelt außerdem die Kommunikation und Synchronisation der Prozesse untereinander. Es ermöglicht sogar die gleichzeitige Benutzung eines oder mehrerer Programme durch mehrere Benutzer. Durch die Benutzung von MTX ist es also gegenüber herkömmlicher Interrupt-organisierter Programmierung möglich,

1. die Ausführungszeit von Benutzeraufträgen wesentlich zu verkürzen.



2. die Programmierung entschieden zu vereinfachen,
3. verschiedenste Aufgaben gleichzeitig zu erledigen, z.B. mehrere Benutzer auf derselben Anlage gleichzeitig rechnen zu lassen.

### 3.3. Beispiel

Zwei Benutzer möchten auf derselben Anlage gleichzeitig rechnen (z.B. Benutzer 1 mit Basic-Interpreter, Benutzer 2 mit Editor). Die hierzu nötige Systemkonfiguration ist in Bild 3-3 dargestellt. Der Benutzer 1 erhält die höhere Priorität; ihm wird der Prozeß 1 zugeordnet, dem Benutzer 2 der Prozeß 2. Die Warteschleife wird zum Prozeß 3 erklärt.

Die Prozesse 1 und 2 sind in herkömmlicher Weise und unabhängig voneinander programmiert. Nur wenige Unterprogrammaufrufe genügen, um MTX mit den nötigen Informationen zu versorgen. MTX regelt dann automatisch und für den Benutzer unsichtbar die zeitliche Verzahnung der Prozesse.

Am Beispiel Bild 3-4 ist ersichtlich, wie kompliziert — in Abhängigkeit der speziellen Betriebssituation — der Ablauf für die CPU sein kann, obwohl die Benutzerprozesse unabhängig voneinander programmiert wurden. Die hierzu nötigen Aufrufe von System-Programmen sind aus dem nachfolgenden Listing zu entnehmen. Dabei bedeutet „;...“: Hier können Benutzerbefehle eingefügt werden.

#### BEISPIEL FÜR SYSTEMKONFIGURATION

#### MIT Z80 - MTX

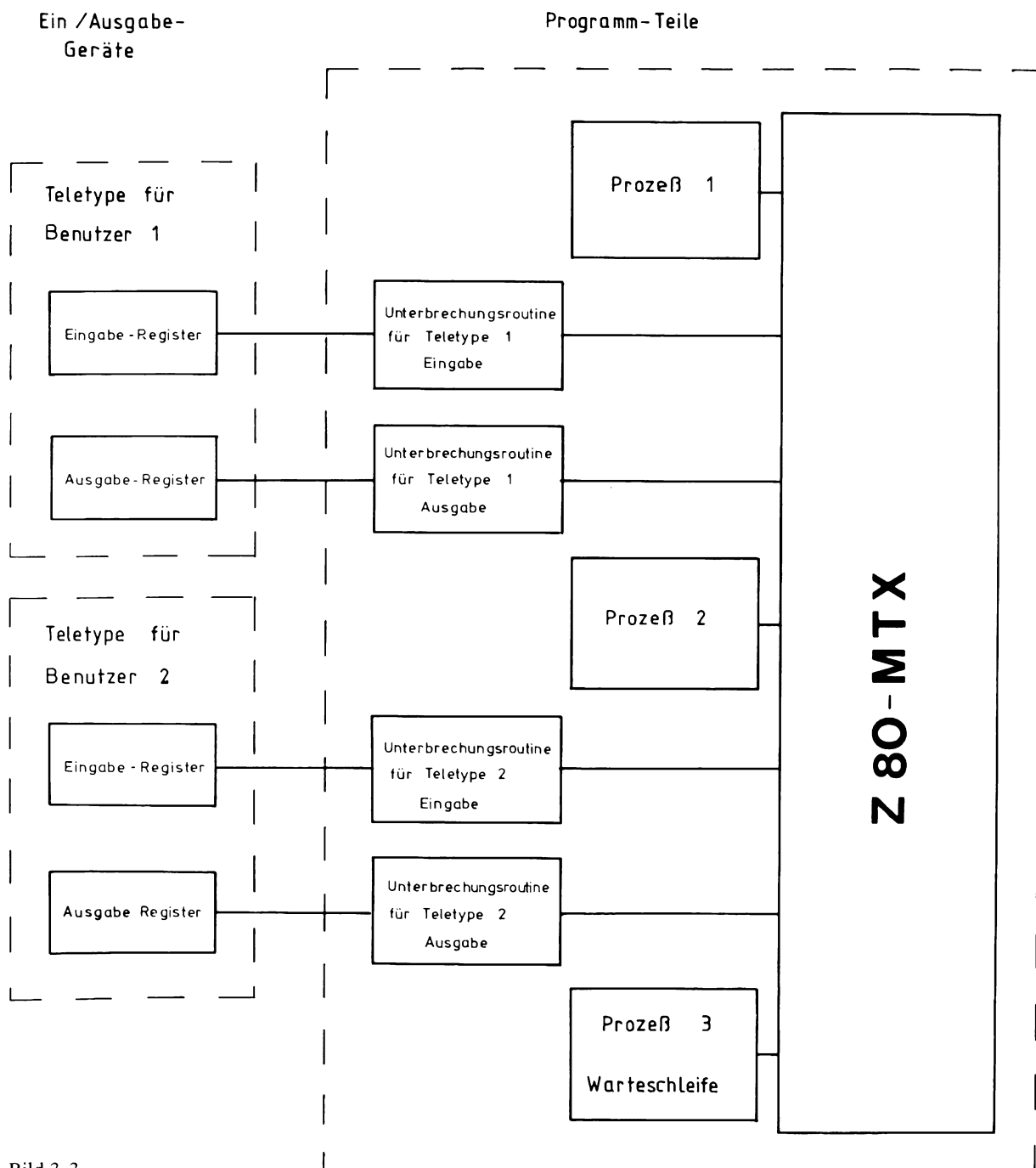
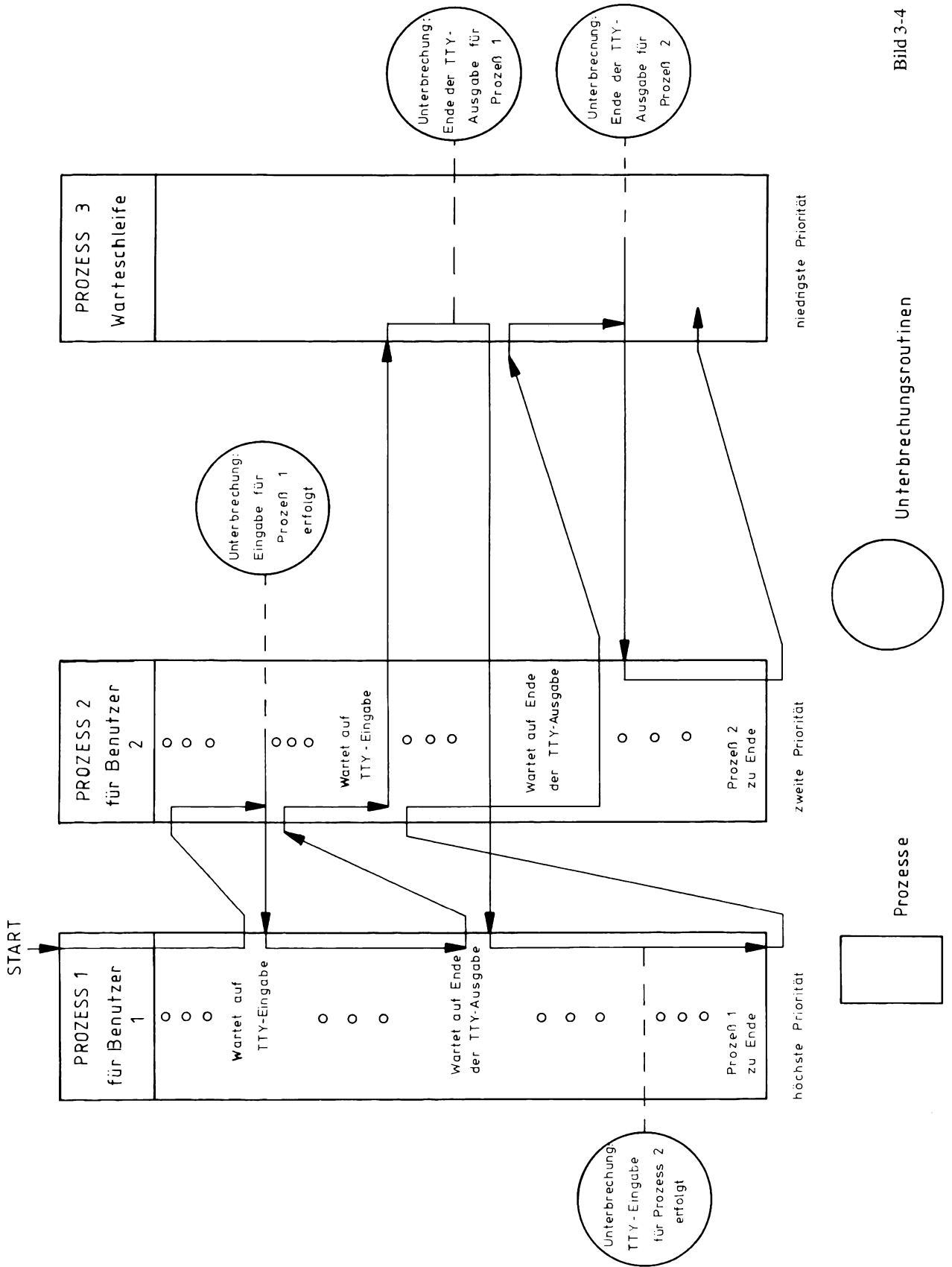


Bild 3-3

BEISPIEL FÜR EINE ABLAUFSTEUERUNG

durch Z80-MTX



```

; BENUTZERHAUPTPROGRAMM
;
; SYSTEM-INITIALISIERUNG
EXTERNAL CLSYS, PROIN, PROEIN, PROAUS, ANBM, BEBM,
EXTERNAL WABM, PUREG, EINT, DINT, UBRE, RETIN;
; JEDEM BETRIEBSMITTEL, HIER DIE VIER TTY-KANAELE,
; MUSS EINE ZAHL >0 UND <126 ZUGEORDNET WERDEN.
TTY1E EQU 1 ; TTY-1 EINGABE KANAL
TTY1A EQU 2 ; TTY-1 AUSGABE KANAL
TTY2E EQU 3 ; TTY-2 EINGABE KANAL
TTY2A EQU 4 ; TTY-2 AUSGABE KANAL
;
LD SP,STACK1 ; LADE STACK-POINTER FUER PROZESS 1
CALL CLSYS ; LOESCHE DEN SYSTEMSPEICHER
CALL PROIN ; INITIALISIERE PROZESS 1
;
LD SP,STACK2 ; LADE STACK-POINTER FUER PROZESS 2
CALL PROIN ; INITIALISIERE PROZESS 2
;
LD SP,STACK3 ; LADE STACK-POINTER FUER PROZESS 3
CALL PROIN ; INITIALISIERE PROZESS 3 (WARTEPROZESS)
;
;
;
LD BC,1 ; LADE PROZESS-NR.
LD DE,PROZESS1 ; LADE STARTADRESSE
CALL PROEIN ; SCHALTE PROZESS 1 EIN
;
LD BC,2
LD DE,PROZESS2
CALL PROEIN ; SCHALTE PROZESS 2 EIN
;
LD BC,3
LD DE,PROZESS3
CALL PROEIN ; SCHALTE PROZESS 3 EIN
;
;
CALL UBRE ; SPRUNG INS SYSTEM
; MTX STARTET PROZESS 1
JP PROZESS3
;
; ENDE SYSTEM INITIALISIERUNG
;
;
;
PROZESS1: ; ANFANG PROZESS 1
; PROGRAMM DES BENUTZERS 1
CALL EINT ; ENABLE INTERRUPT IM PROZESS 1
; ...
; ...
; ...
CALL TY1EIN ; TTY-EINGABE-PROGRAMM
LD BC, TTY1E ; MITTEILUNG AN MTX.
CALL ANBM ; DASS TTY-EINGABE ERWARTET WIRD
; ...
LD BC, TTY1E ; FUEHRE PROZESS 1 FORT,
CALL WABM ; WENN EINGABE ERFOLGT IST
; ...
; ...
; ...
CALL TY1AUS ; TTY-AUSGABE-PROGRAMM
LD BC, TTY1A ; MITTEILUNG AN MTX.
CALL ANBM ; DASS TTY AUSGABE ERWARTET WIRD
; ...

```

```

LD BC, TTY1A ; FUEHRE PROZESS 1 FORT,
CALL WABM ; WENN AUSGABE ERFOLGT IST
;
;
;
LD BC, 1 ; SCHALTE PROZESS 1 AUS
CALL PROAUS
CALL UBRE ; SPRINGE INS SYSTEM
; ENDE PROZESS1
;
;
; TTY-EINGABE-UNTERPROGRAMM :
TY1EIN:
;
;
; RET
;
; TTY-AUSGABE-UNTERPROGRAMM :
TY1AUS:
;
;
; RET
;
;
;
PROZESS2: ; WIE PROZESS 1 (ERSETZE JEDE 1 DURCH 2)
;
;
;
PROZESS3: ; WARTE-PROZESS
CALL EINT ; ENABLE INTERRUPT IM PROZESS 3
;
;
WARTE: ; WARTE-SCHLEIFE
JR WARTE
;
; ENDE PROZESS 3
;
;
;
; UNTERBRECHUNGSBEARBEITUNG FUER TTY1-EINGABE
CALL PUREG ; MTX SPEICHERT ALLE REGISTER
;
;
;
LD BC, TTY1E ; MITTEILUNG AN MTX
CALL BEBM ; DASS EINGABE ERFOLGT IST
JR RETIN ; EINSPRUNG INS MTX, ANSTATT DES
; RETI-BEFEHLS
;
;
;
; UNTERBRECHUNGSBEARBEITUNG FUER TTY1-AUSGABE
; WIE OBEN NUR TTY1A ANSTATT TTY1E
;
;
;
; UNTERBRECHUNGSBEARBEITUNG FUER TTY2-EINGABE
; WIE OBEN NUR TTY2E
;
;
;
; UNTERBRECHUNGSBEARBEITUNG FUER TTY2-AUSGABE
; WIE OBEN NUR TTY2A
;
;

```

```

;
;
;   ORG RAM           ; RAM-ANFANGSADRESSE
SYSRAM           ; MTX BRAUCHT 1/4 K-RAM
;   DEFS 100H        ; UNTER DIESEM GLOBAL-NAMEN
GLOBAL SYSRAM
STACK:           ; ANFANG STACK-BEREICH
;   DEFS STACK-BEREICH PROZESS 1
STACK1:          ; STACK ANFANG PROZESS 1
;   DEFS STACK-BEREICH PROZESS 2
STACK2:          ; STACK ANFANG PROZESS 2
;   DEFS 52          ; STACK BEREICH WART-PROZESS
STACK3:          ; STACK ANFANG PROZESS 3
;
;
;
;   ENDE BENUTZER PROGRAMM
;

```

### 3.4. Programmorganisation

Das Benutzungsprogramm teilt sich auf in vier Klassen: Systeminitialisierung, Prozesse, Prozeßunabhängige Unterprogramme und Unterbrechungsbehandlungen.

#### 3.4.1 Systeminitialisierung

In der Systeminitialisierung werden die Systemtabellen rückgesetzt (durch CALL CLSYS) und die Prozesse eingerichtet (durch CALL PROIN). Hierbei wird ihre Priorität festgelegt (durch CALL PROIN). Hierbei wird ihre Priorität festgelegt. Außerdem können bereits Prozesse eingeschaltet werden.

Der letzte Prozeß (in der Regel die Warteschleife) muß eingeschaltet werden. Damit immer wenigstens ein rechenwilliger Prozeß existiert. Er hat die niedrigste Priorität, der Interrupt muß in diesem Prozeß in der Regel „Enabled“ sein (durch CALL EINT).

Im einfachsten Fall hat der Warte-Prozeß folgende Gestalt:  
 WARTEPROZESS: ; Prozeß mit höchster Prozeß-Nr.  
 CALL EINT  
 WARTE:  
 ; JR WARTE  
 ; ENDE WARTE PROZESS

Die Systeminitialisierung muß nach folgendem Schema aufgebaut werden:

```

Urstart:
;
;   LD SP, STACK1    ; INITIALISIERE PROZESS-NR.1
;   CALL CLSYS       ; LOESCHE DEN SYSTEMSPEICHER
;   CALL PROIN       ; MIT HOECHSTER PRIORITAET
;
;   LD SP, STACK2    ; INITIALISIERE PROZESS-NR.2
;   CALL PROIN       ; MIT 2.HOECHSTER PRIORITAET
;
;   USW.
;
;   LD SP, STACKM    ; INITIALISIERE PROZESS-NR.M
;   CALL PROIN       ; MIT NIEDRIGSTER PRIORITAET
;                   ; IM ALLGEMEINEN DER WARTE-PROZESS
;
;   ; HIER KOENNEN PROZESSE EINGESCHALTET WERDEN
;   ; DER WARTE-PROZESS MUSS EINGESCHALTET WERDEN
;   LD BC,M          ; LADE PROZESS NR.M
;   LD DE,WARTEPROZESS ; LADE PROZESS START ADRESSE
;   CALL PROEIN      ; SCHALTE WARTE PROZESS EIN
;
;   CALL UBRE        ; WENN NICHT SOFORT DER WARTEPROZESS
;                   ; AUSGEFUEHRT WERDEN SOLL, MUSS DIESER
;                   ; AUFRUF EINGEFUEGT WERDEN
;   JP WARTEPROZESS ; IMMER DER LETZTE BEFEHL DER
;                   ; INITIALISIERUNG

```

### 3.4.2 Prozesse

Alle Benutzer-Programmteile die nicht unter 3.4.1, 3.4.3 und 3.4.4 fallen, sind Prozessen zuzuordnen. Ein Prozeß kann eingeschaltet (durch CALL PROEIN) und ausgeschaltet (durch CALL PROAUS) werden. Er ist durch folgende Daten charakterisiert.

#### 3.4.2.1 Priorität

Sie ist gleich der Prozeß-Nr. Diese wird bei der Systeminitialisierung festgelegt. Die Prozeß-Nr. bzw. die Priorität muß  $>0$  sein. Maximal 28 Prozesse sind erlaubt. Die kleinste Prozeß-Nr. (nämlich 1) hat die größte Priorität.

#### 3.4.2.2 Stack

Jeder Prozeß hat seinen eigenen Stack-Bereich. Seine Größe berechnet sich aus der Summe folgender vier Posten:

- 1) Prozeß-bedingte Stack-Tiefe.
- 2) 26 Bytes für Systemaktionen.
- 3) Wenn Interrupts erlaubt sind, größte Stack-Tiefe aller möglichen Unterbrechungsbehandlungen (eine mögliche Schachtelung der Unterbrechungsbehandlungen braucht nicht mit einkalkuliert zu werden).
- 4) Wenn Interrupts erlaubt sind, 26 weitere Bytes für Systemaktionen.

Bei der Initialisierung wird dem System die Stack-Anfangsadresse mitgeteilt. MTX sorgt dafür, daß beim Einschalten und Wechseln von Prozessen die Stack-Pointer stimmen.

#### 3.4.2.3 Interrupt-Status

Der Interrupt Status darf nur in einem Prozeß ein- oder ausgeschaltet werden, und zwar nur mit den zugehörigen Unterprogrammen durch CALL EINT und CALL DINT. Die Befehle „EI“ und „DI“ sind also für den Benutzer verboten. EINT und DINT wirken nur für den Prozeß, in dem sie aufgerufen werden. MTX sorgt dafür, daß beim Wechseln von Prozessen der Interrupt-Status stimmt. Nach dem Start eines Prozesses ist der Interrupt-Status ausgeschaltet („Disabled“).

#### 3.4.2.4 Startadresse

Die Ausführung eines Prozesses beginnt zu gegebener Zeit bei dieser Adresse. Sie wird dem System beim Einschalten eines Prozesses mitgeteilt. Wurde ein Prozeß ausgeschaltet, kann er mit der gleichen oder einer neuen Startadresse erneut eingeschaltet werden.

#### 3.4.2.5 Rechenwilligkeit

Ein Prozeß ist rechenwillig, wenn er eingeschaltet ist und nicht auf die Bereitstellung eines Betriebsmittels wartet. Unter einem Betriebsmittel sind diejenigen Aktivitäten zu verstehen, die ein Prozeß zu seiner Fortführung benötigt. Ein Betriebsmittel ist gekennzeichnet durch seine Betriebsmittelnummer. Es tritt erst dann in Erscheinung, wenn es angefordert wird. Damit wird einem Prozeß die Möglichkeit eröffnet, auf dieses Betriebsmittel zu warten. Zum Betriebsmittel gehört weiterhin eine Fertigmeldung an das System unter seiner Betriebsmittel-Nr. (CALL BEBM). Die den darauf wartenden Prozeß rechenwillig macht.

Unter einem Betriebsmittel kann sowohl ein Prozeß als auch eine Ein/Ausgabe-Aktivität verstanden werden. Im letzteren Fall steht die Fertigmeldung in der Unterbrechungsbehandlung. Ein Prozeß hat also die besondere Eigenschaft, daß er warten kann (CALL WABM), d.h. seine Fortführung wird unterbrochen und andere Prozesse werden weitergerechnet, bis das betreffende Betriebsmittel bereitgestellt ist. Also die Fertigmeldung — CALL BEBM — abgegeben hat. Unter Berücksichtigung seiner Priorität wird der wartende Prozeß dann fortgesetzt.

#### 3.4.3 Prozeßunabhängige Unterprogramme

Es handelt sich hier um Unterprogramme, die durch mehrere Prozesse benützt werden können. Auch in Unterbrechungsbehandlungen können diese Unterprogramme aufgerufen werden (z.B. Arithmetik-Unterprogramme wie Floating-Point-Package Z80 FPA/EAP).

Möchte der Benutzer solche Programme schreiben, muß er eine spezielle Programmierweise verwenden (Eintrittsinvariant) oder sicherstellen, daß, solange das Unterprogramm von einem Prozeß benutzt wird, kein anderer Prozeß oder keine Unterbrechungsbehandlung auf dieses Unterprogramm zugreift.

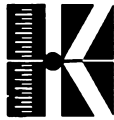
#### 3.4.4 Unterbrechungsbehandlung

Unterbrechungsbehandlungen (auch Interrupt-Service-Routinen genannt), müssen mit einem System-Aufruf (CALL PUREG) beginnen. MTX sorgt dann dafür, daß die Register und der Status des unterbrochenen Prozesses gespeichert werden. Es stehen also dem Benutzer in der Unterbrechungsbehandlung alle Register zur freien Verfügung. Jedoch darf der Interrupt auf CPU-Ebene nicht eingeschaltet werden (Befehl „EI“). Anstelle des „RETI“-Befehls muß in jedem Falle ein Sprung ins System erfolgen (JPRETIN), damit die Interruptlogik des MTX nicht gestört wird. Das System hebt für kurze Zeit die Interruptsperre auf, um Interrupts mit höherer Priorität (siehe PIO/CTC Manual) zuzulassen. Eine wesentliche Aufgabe von Unterbrechungsbehandlungen kann es sein, dem System die Bereitstellung eines Betriebsmittels durch die entsprechende Fertigmeldung (CALL BEBM) anzuzeigen.



**DEUTSCHLAND**

**KONTRON ELEKTRONIK GmbH**  
8057 Eching b. München  
Breslauer Straße 2  
Tel. (0 89) 31 88-1  
Telex 05 22 122



**ÖSTERREICH**

**Kontron Ges mbH & Co KG**  
A-1140 Wien  
Ameisgasse 49  
Telefon 94 56 46  
Telex 11 699



**SCHWEIZ**

**STOLZ AG**  
CH 8968 Mutschellen  
Telefon 0575 4655  
Telex 54 070



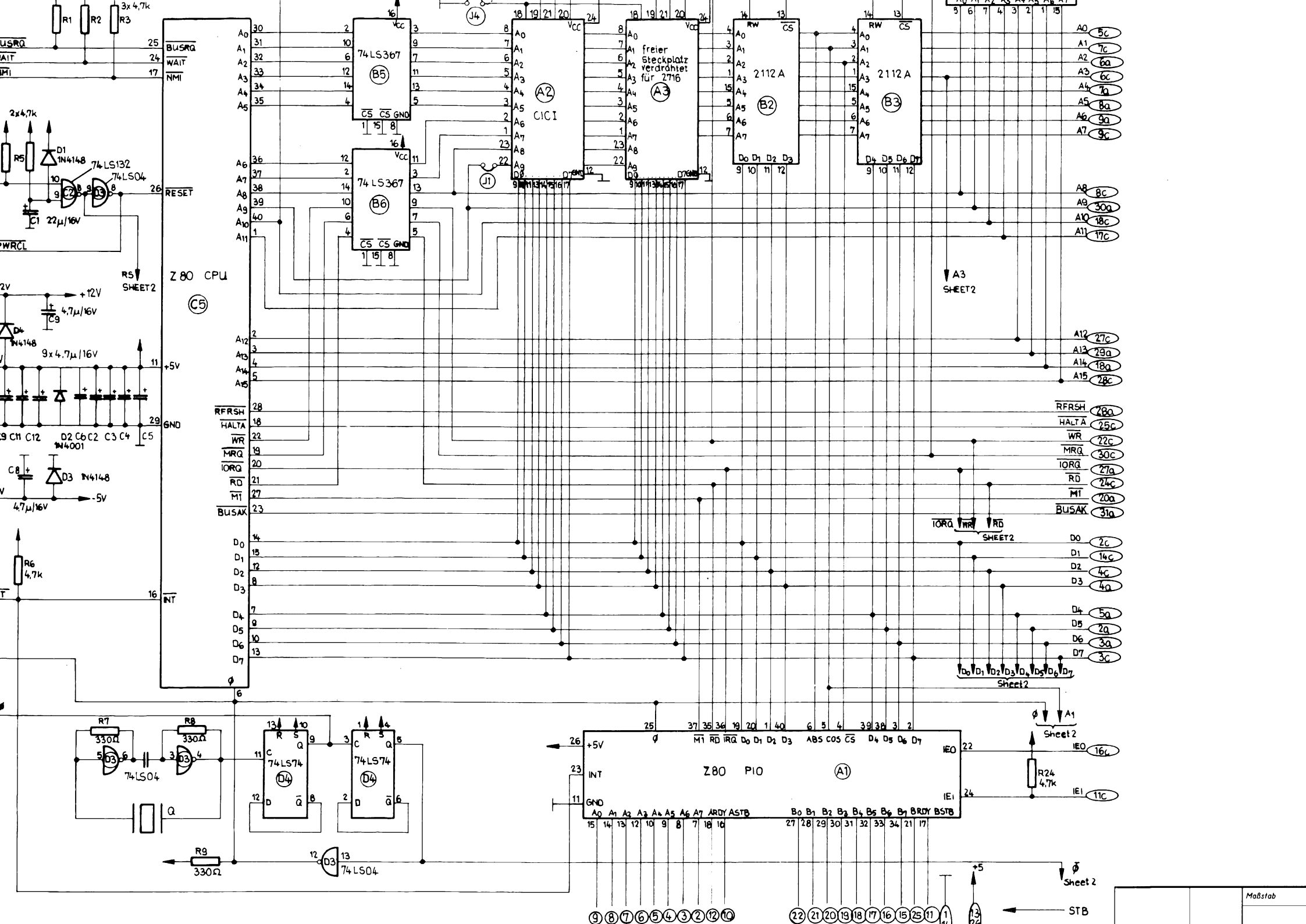
**EUROPA**

**ZILOG (UK) LTD.**  
Nicholson House,  
Maidenhead, Berkshire,  
**UNITED KINGDOM**  
Telefon 0628-36131/2/3  
Telex 848 609



**USA**

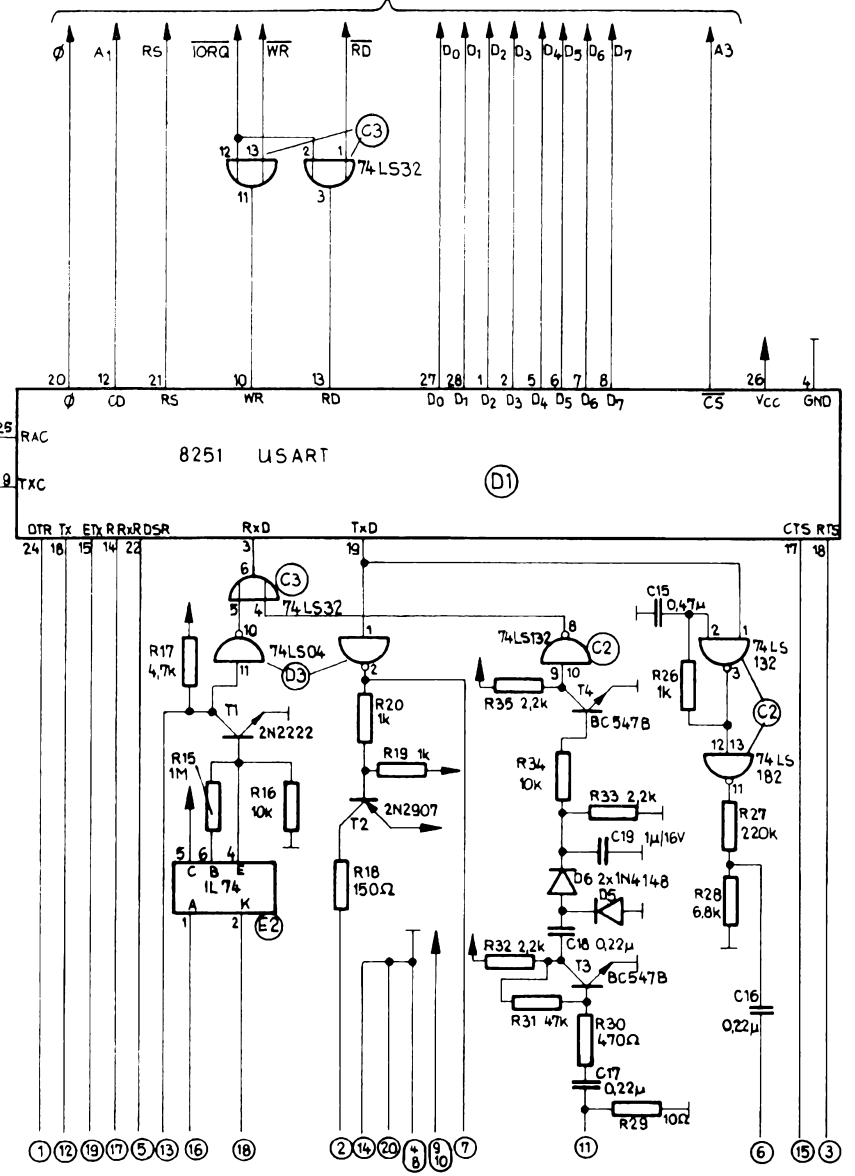
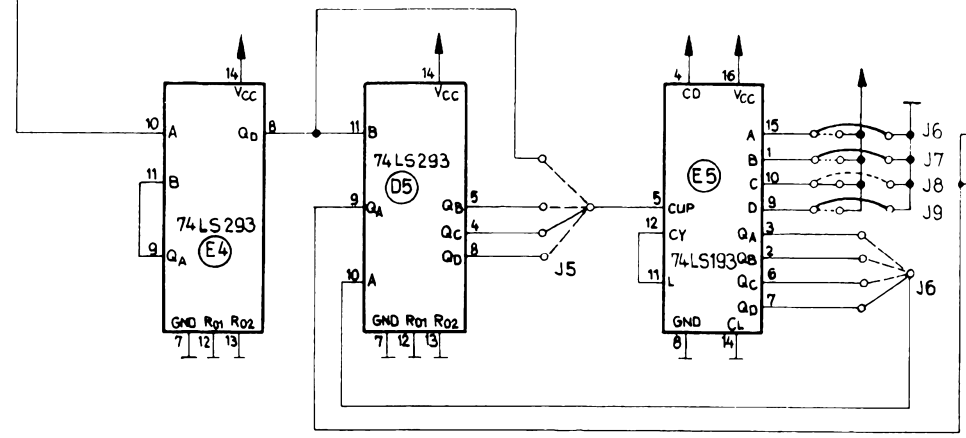
**ZILOG INC.**  
10460 Bubb Road,  
Cupertino, Cal. 95014  
Telefon (408) 446-4666  
Telex 910-338-7621





Sheet 1

Sheet 1



STC →