

MFA-MEDIENSYSTEM

# Mikrocomputer- Technik

Fachpraktische Übungen  
Band 6

6

**MFA**  
MEDIENSYSTEM

**VGS Bfz**





---

MFA-MEDIENSYSTEM

---

# Mikrocomputer- Technik

---

Fachpraktische Übungen · Band 6

---





**MFA-Mediensystem Mikrocomputer-Technik** / [Red.: F.

Derriks ...]. – Köln : vgs.

Teilw. redigiert von N. Meyer ... Teilw. hrsg. vom BFZ Essen

NE: Meyer, Norbert [Red.]; Derriks, Franz [Red.];

Berufsförderungszentrum <Essen>; Mikrocomputer-Technik

Fachpraktische Übungen.

Bd. 6. – 1. Aufl. – 1992

ISBN 3-8025-1260-X

Herausgeber: vgs verlagsgesellschaft mbH & Co. KG  
Breite Str. 118-120,  
5000 Köln 1

Redaktion: F. Derriks, B. Liesbrock, Fr.G. Roßmanek,  
H. Sprenger, W. Hild, W. Oehlert, D. Piller

C 1991 vgs verlagsgesellschaft mbH & Co. KG  
Berufsförderungszentrum Essen e.V.

Diese Publikation ist urheberrechtlich  
geschützt. Alle Rechte sind vorbehalten.

Verlag: vgs verlagsgesellschaft, Köln

1. Auflage 1991

Satz und Zeichnungen: BFZ Essen; Teba-Elektronik, Nenderoth

## Hinweise zu den Übungen in diesem Band

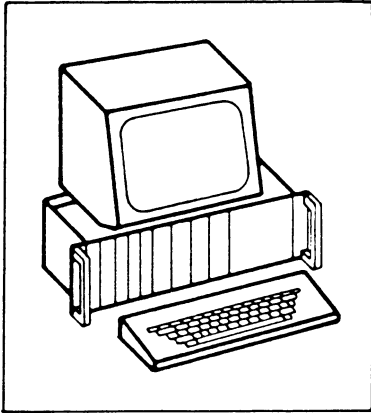
Die in diesem Band enthaltenen Übungen sind nicht durchgehend, sondern einzeln paginiert, und zwar oben rechts auf jeder Seite. Dabei ist jeweils auch die Systemnummer der betreffenden Übung angegeben (z.B. NSC800-Adapter BFZ/MFA 2.2). Hierdurch ist es möglich, den Band auseinanderzunehmen und die Übungen einzeln einzusetzen.

**Dieser Band enthält folgenden Übungen:**

- NSC800-Adapter BFZ/MFA 2.2
- GAL-Programmierer BFZ/MFA 4.14
- Digitale Motorregelung
- Einplatinen-Computer (EPC)



# FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



NSC-800-Adapter







## Inhaltsverzeichnis

---

1. Allgemeines
2. Funktionsbeschreibung
  - 2.1 Die Eingangssignale des NSC800
  - 2.2 Die Ausgangssignale des NSC800
3. Der Register- und Befehlssatz des NSC800
  - 3.1 Das Interrupt-Control-Register
  - 3.2 Der Befehlssatz
4. Inbetriebnahmehinweise
5. Programmier-Beispiele
  - 5.1 Z80-Programmbeispiel
  - 5.2 Turbo-PASCAL-Programmbeispiel

### Anhang:

- A1 Stromlaufplan Adapter-Platine
- A2 Bestückungsplan Adapter-Platine
- A3 Bestückungsplan CPU-Baugruppe
- A4 Bestückungsplan 64-K-RAM-Baugruppe
- A5 Anschlußplan NSC800
- A6 Befehlsliste NSC800/Z80
- A7 Z80-Assemblerprogramm-Listing
- A8 Turbo-PASCAL-Programmlisting
- A9 Anpassen des Programms BFZFORM.COM an den NSC800
- A10 Literaturhinweise

Abhilfe bei Startproblemen

## Funktionsbeschreibung

---

### 1. Allgemeines

-----

Mit dem NSC800-Mikroprozessor liegt ein Prozessor vor, der zur Vereinigung der jeweiligen Vorteile der Prozessoren Z80 und 8085 entwickelt worden ist. Er ist hardware-mäßig dem 8085- und software-mäßig dem Z80-Prozessor weitgehend kompatibel. Damit bietet er die Möglichkeit, den 8085-Prozessor im MFA-Mikrocomputersystem mit geringem Aufwand durch einen Z80-kompatiblen Prozessor zu ersetzen. Da der Befehlssatz des 8085-Prozessors bis auf die Befehle RIM und SIM eine Teilmenge des Z80-Befehlssatzes darstellt, können 8085-Programme in Z80-Systemen ablaufen. Umgekehrt ist das nicht möglich, d.h., Z80-Programme laufen nicht in 8085-Systemen.

Diese Ausbaustufe für den MFA-Mikrocomputer ist für das CP/M-System vorgesehen, weil sich damit die Möglichkeit eröffnet, das gesamte, für CP/M-Systeme verfügbare Software-Angebot nutzen zu können.

Beispielsweise wird es mit diesem Prozessor möglich, Hochsprachen wie Turbo-PASCAL auf dem MFA-Mikrocomputer einzusetzen. Speziell diese Programmiersprache ist im Personal-Computer-Bereich weit verbreitet und bietet auch im MFA-System weitere didaktische Möglichkeiten.

Darüber hinaus kann der Makroassembler M80, der Teil des Microsoft-Entwicklungspaketes (BASIC-Interpreter/-Compiler) ist, durch Ausnutzung der Z80-Mnemonik für die Einführung in die Z80-Assemblersprache verwendet werden.

### 2. Funktionsbeschreibung

-----

Dem Stromlaufplan der Adapter-Platine (Anhang A1) kann entnommen werden, daß die Signale des NSC800-Prozessors denen des 8085-Prozessors weitgehend entsprechen und sich teilweise lediglich in der Bezeichnung unterscheiden. Allerdings ist die Pin-Belegung zum 8085 verschieden und einige Signale müssen invertiert werden.

In der folgenden Beschreibung wird daher im wesentlichen auf die Besonderheiten dieses Prozessors und auf die Unterschiede zum Z80 oder 8085 eingegangen.

Die Analogie der Anschlüsse zwischen dem 8085- und dem NSC800-Prozessor kann dem Schaltbild direkt entnommen werden. Der 8085-Prozessor unterscheidet sich insbesondere durch die Anschlüsse SID (Serial Input DATA, Pin 5) und SOD (Serial OUTPUT DATA, Pin 4) vom NSC800. Dafür besitzt der NSC800 die Anschlüsse PS (Power Save, Pin 39) und RFSH (Refresh, Pin 28). Auf der Adapterplatine sind diese Signale durch offene Lötbrücken vom Platinenstecker getrennt und der Eingang PS (Pin 39) liegt über einen Pull-Up-Widerstand fest auf H-Pegel. Diese Signale werden hier nicht weiter verwendet.

## Funktionsbeschreibung

Darüber hinaus ist der Interrupt-Eingang NMI (Non-Maskable Interrupt, Pin 21) des NSC800 vom Platinenstecker getrennt und ebenfalls fest auf H-Pegel gelegt. Dieser Anschluß entspricht dem Interrupt-Signal TRAP des 8085. Gegenüber dem 8085 ist das NMI-Signal ein aktiv Low-Signal und die zugehörige (Restart-)Adresse ist vom 8085 verschieden.

! 8085- ! Anschluß	! Restart- ! Adresse	! NSC800- ! Anschluß	! Restart ! Adresse	!
! TRAP (Pin 6)	! 0024H	! $\overline{\text{NMI}}$ (Pin 21)	! 0066H	!

Die Anpassung des NSC800-Prozessors an die 8085-Prozessor-Platine erfolgt über einen PAL-Baustein (GAL16V8). Sie wurde so vorgenommen, daß die sich im aktiven Pegel unterscheidenden Signale über Inverter geführt werden. Darüber hinaus wurde auch eine weitgehende Kompatibilität zwischen dem RST7.5-Anschluß des 8085-Prozessors und dem RSTA-Anschluß des NSC800 hergestellt. Beide Anschlüsse unterscheiden sich dadurch, daß der RST7.5-Eingang des 8085 flankengesteuert arbeitet, während der RSTA-Eingang des NSC800 statisch wirkt. Da der flankengesteuerte RST7.5-Eingang Vorteile bei der Anwendung bietet, wurde mittels des PAL-Bausteins diese Anpassung ebenfalls vorgenommen. So lassen sich leichter interrupt-gesteuerte Programme realisieren, die in Verbindung mit der Zähler- und Zeitgeber-Baugruppe (BFZ/MFA 4.6) ausgetestet werden können, indem ein Interrupt über den Handtaster in der Frontplatte der Baugruppe ausgelöst wird. Dabei ist es dann unbedeutend, wie lange der Taster betätigt wird und es brauchen keine programmiertechnischen Abfragen in das Programm eingefügt werden.

Im Anhang ist ein einfaches Beispiel für ein solches interrupt-gesteuertes Programm angegeben, welches manuell getestet werden kann.

## Funktionsbeschreibung

---

### 2.1 Die Eingangssignale des NSC800

---

Im folgenden sind alle NSC800-Bausteinanschlüsse kurz beschrieben. Für eine genaue Funktionsbeschreibung sei auf das zugehörige Datenblatt der Firma National Semiconductor Corporation verwiesen.

#### RESET INPUT (Pin 33):

Aktiv Low-Signal; schaltet die Leitungen A8 - A15 sowie AD0 - AD7 in den Tri-State und löscht die Register PC, I und R; sperrt Interrupts und erzeugt das Signal RESET OUT.

#### BUS REQUEST (Pin 36):

Aktiv Low-Signal; wird verwendet, wenn ein anderes Gerät den BUS anfordert; die CPU erkennt das Signal am Ende des aktuellen Maschinenzyklusses; bewirkt, daß die Leitungen A8 - A15, AD0 - AD7, sowie IO/M, RD, and WR in den Tri-State geschaltet werden; die CPU bestätigt die Anforderung durch das Signal BUS ACKNOWLEDGE.

#### NON-MASKABLE INTERRUPT (Pin 21):

Aktiv Low-Signal; flankengesteuert; nicht-sperrbares Interrupt-Signal mit der höchsten Priorität, d.h. unabhängig vom Interrupt-Enable-Flipflop; wenn aktiv, so sichert die CPU den PC auf dem STACK und verzweigt zur Adresse 0066H.

#### RESTART INTERRUPTS A,B,C (Pin 22,23,24):

Aktiv Low-Signale; sperrbare Interrupt-Signale; Ausführung wie beim NMI-Interrupt; Priorität in der Folge A, B, C; zugehörige Adressen RSTA -> 003CH, RSTB -> 0034H, RSTC -> 002CH.

8085- Anschluß	Restart- Adresse	NSC800- Anschluß	Restart Adresse
RST7.5 (Pin 7)	003CH	RSTA (Pin 22)	003CH
RST6.5 (Pin 8)	0034H	RSTB (Pin 23)	0034H
RST5.5 (Pin 9)	002CH	RSTC (Pin 24)	002CH

## Funktionsbeschreibung

---

### INTERRUPT REQUEST (Pin 25):

Aktiv Low-Signal; Interrupt mit der niedrigsten Priorität; die CPU unterscheidet drei Betriebsarten; in der Betriebsart 0 kann wie beim 8085 über Restart-Vektoren (RST 0 - RST 7) bzw. über einen drei-Byte Call-Befehl zur Interrupt-Service-Routine verzweigt werden; in der Betriebsart 1 verzweigt die CPU fest zur Restart-Adresse 0038H; die Betriebsart 2 erfordert spezielle Ein-/Ausgabe-Bausteine, die nach Anforderung eines Interruptes das Low-Byte einer Adresse liefern können, dieser Teil ergänzt den Inhalt des Interrupt-Vektor-Registers der CPU und bildet eine Speicheradresse, von der die CPU die Adresse der Interrupt-Service-Routine liest (indirekter Sprung zur ISR).

### WAIT (Pin 38):

Aktiv Low-Signal; die CPU verlängert die Maschinenzyklen durch Warte-Zyklen, wenn dieses Signal während eines RD-, WR- oder INTA-Maschinenzyklusses aktiviert wird.

### POWER SAVE (Pin 39):

Aktiv Low-Signal; wenn aktiv, dann stoppt die CPU am Ende des Befehlszyklusses und geht in einen Low-Power-Mode mit geringem Stromverbrauch.

### Xin,Xout (Pin 10,11):

Quarzanschlüsse für den integrierten Taktgenerator.

## 2.2 Ausgangssignale des NSC800

-----

### BUS ACKNOWLEDGE (Pin 35):

Aktiv Low-Signal; zeigt an, daß die CPU den BUS und die Steuer-Signale in den Tri-State geschaltet hat (siehe BUS REQUEST).

### A8 - A15 (Pin 1 - 8):

Aktiv High-Signale; acht höchstwertige Adress-Leitungen.

### AD0 - AD7 (Pin 12 - 19):

Adress- und Daten-Leitungen, die im Betrieb umgeschaltet (gemultiplext) werden; die Leitungen werden beim Datentransport auch als Eingänge verwendet.

### RESET OUT (Pin 37):

Aktiv High-Signal; wenn aktiv, so führt die CPU einen RESET aus.

### INPUT/OUTPUT/MEMORY (Pin 34):

Das High-Signal zeigt einen Schreib- oder Lese-Zugriff der CPU auf Ein-/Ausgabe-Baugruppen an; das Low-Signal entsprechend den Zugriff auf den Speicher.

### REFRESH (Pin 28):

Aktiv Low-Signal; zeigt einen Refresh-Zyklus für dynamische RAM-Speicher an.



---

## Register- und Befehlssatz

---

### ADDRESS LATCH ENABLE (Pin 30):

Aktiv High-Signal; ein High-Low-Signalwechsel signalisiert eine gültige Adresse, das untere Adreß-Byte muß mit diesem Signal gespeichert werden.

### READ STROBE (Pin32):

Aktiv Low-Signal; die CPU liest Daten mit dem Signalwechsel am Ende des Signals.

### WRITE STROBE (Pin 31):

Aktiv Low-Signal; die CPU sendet Daten während des aktiven Signals.

### CLOCK (Pin 9):

Systemtakt mit einer Frequenz, die halb so groß ist wie die Frequenz des Taktgenerators.

### INTERRUPT AKNOWLEDGE (Pin 26):

Aktiv Low-Signal; CPU liest mit diesem Signal den Interrupt-Vektor der Interface-Baugruppe, die den Interrupt ausgelöst hat.

## 3. Der Register- und Befehlssatz des NSC800

-----

Der Registersatz des NSC800 entspricht dem des Z80, d.h. er besitzt gegenüber dem 8085 den doppelten Arbeitsregistersatz sowie zwei Indexregister (IX und IY). Daneben gibt es zwei weitere Register, das Interrupt-Vektor- (I) und das Refresh-Register (R).

Der NSC800 unterscheidet in der Interrupt-Verarbeitung des Interrupt-Request-Signals (INTR, Pin 25) zwischen drei Betriebsarten. Die Betriebsart 2, auch Mode 2 genannt, verlangt spezielle Ein-/Ausgabe-Bausteine als Interrupt-Steuerbausteine. In dieser Betriebsart wird der Inhalt des I-Registers als oberer Adressteil einer Zeigeradresse für das Auffinden der Interrupt-Service-Routine verwendet. Werden dynamische RAM-Bausteine als Speicher verwendet, so müssen die RAM-Zellen in bestimmten Zeitabständen aufgefrischt werden. Der Z80 sowie der NSC800 verwenden dafür ein spezielles Register, welches als Refresh-Zähler verwendet werden kann. Im Gegensatz zum Z80 ist dieses Refresh-Register auf 8-Bit erweitert worden.

```

NSC800 (Z80) Flags
+-----+
! 7 ! 6 ! 5 ! 4 ! 3 ! 2 ! 1 ! 0 !
!---!---!---!---!---!---!---!
! S ! Z ! X ! H ! X !P/V! N ! C !
+-----+

!      !      !      !      !      !
!      !      !      !      !      +-- Übertrags-Flag
!      !      !      !      !
!      !      !      !      !      Additions-/
!      !      !      !      !      Subtraktions-Flag
!      !      !      !      +-----
!      !      !      !      Paritäts-/
!      !      !      !      Überlauf-Flag
!      !      !      +-----
!      !      !      Halbübertrag-Flag
!      !      +-----
!      !      Null-Flag
!
+----- Vorzeichen-Flag

```

Wesentlicher Unterschied zum Z80 ist ein weiteres Register, das Interrupt-Control-Register (ICR), welches in der vorstehenden Darstellung nicht aufgeführt ist. Dieses Register wurde notwendig, um die zum 8085 kompatiblen Interruptsignale RSTA, RSTB und RSTC maskieren, d.h. sperren oder freigeben zu können. Diese drei Interruptsignale, die der Z80 nicht besitzt, entsprechen den 8085-Interruptsignalen RST7.5, RST6.5 und RST5.5. Der 8085 besitzt zum Lesen und Schreiben seines Interrupt-Masken-Registers die Befehle RIM und SIM, die beim NSC800-Prozessor nicht vorhanden sind. Sein Befehlssatz ist voll kompatibel zum Z80. Daher

## Register- und Befehlssatz

wird beim NSC800 das Interrupt-Control-Register wie eine Ausgabe-Einheit mit einer festen Baugruppen-Nummer (BBH) behandelt und kann mit einem OUT-Befehl verändert werden. Im folgenden Bild ist die Zuordnung der einzelnen Masken-Bits zu den Interrupt-Signalen wiedergegeben.

Bit :	7	6	5	4	3	2	1	0
ICR :	! - ! - ! - ! - ! IEA ! IEB ! IEC ! IEI !							
	-- keine Bedeutung --				!	!	!	!
					!	!	!	!
					!	!	!	!
					!	!	!	!
Interrupt-Freigabe-Bit RSTA	----				!	!	!	!
Interrupt-Freigabe-Bit RSTB	-----				!	!	!	!
Interrupt-Freigabe-Bit RSTC	-----				!	!	!	!
Interrupt-Freigabe-Bit INTR	-----				!	!	!	!

Damit eine Interrupt-Anforderung von der CPU angenommen werden kann, muß das zugehörige Freigabe-Bit gesetzt ( = 1 ) sein. Darüber hinaus müssen Interrupts über den Befehl Enable Interrupt (EI) freigegeben sein.

### 3.2 Der Befehlssatz

Im Anhang A6 ist die Befehlsliste des NSC800-/Z80 dargestellt. Um die Liste möglichst kurz und übersichtlich zu gestalten, wird eine Symbolik verwendet, die in solchen Kurzdarstellungen üblich ist. In der Befehlsliste wird neben der Mnemonik und der Wirkung eines Befehls auch eine Erläuterung gegeben, sofern dies erforderlich ist. Die verwendete Symbolik soll im folgenden noch einmal kurz erklärt werden, um die Liste lesbar zu machen.

Ein Datum (oder ein Operand), das von einem Befehl beeinflusst wird, ist durch seine Adresse im Speicher oder durch das Register gekennzeichnet, in welchem es sich befindet. Dieser Sachverhalt wird durch folgende Schreibweise ausgedrückt:

< Adresse > = Datum im Speicher unter Adresse  
 < Register > = Datum im Prozessor-Register

Die spitzen Klammern symbolisieren den Inhalt der Speicherstelle oder des Registers und können wie folgt gelesen werden:

< ... > = "Inhalt von ..."

## Inbetriebnahmehinweise

---

In der Ausführung der Befehle kommt es vor, daß ein Datum im Speicher beeinflußt wird, dessen Adresse in einem Registerpaar, z.B. dem HL-Registerpaar, steht. Dieser Sachverhalt wird wie folgt dargestellt:

<< Registerpaar >> = Datum im Speicher unter der Adresse,  
die im Registerpaar steht

Wird nun durch einen Befehl ein Datentransport ausgeführt, so wird dies durch die in der Programmiersprache PASCAL übliche Wertzuweisung zum Ausdruck gebracht:

< Adresse > := Ergebnis  
< Register > := Ergebnis

Dies bedeutet, daß das Ergebnis unter der Adresse im Speicher oder in dem angegebenen Register abgelegt wird.

Mit dieser Schreibweise (Notation) ist es möglich, die Befehlsliste kurz und übersichtlich zu gestalten. Nach einer Eingewöhnungsphase wird es dem Benutzer keine Probleme mehr machen, mit einer solchen Befehlsliste umgehen zu können.

Im Einzelfall kann es notwendig werden, die exakte Beschreibung eines Befehls nachlesen zu können. Dazu sei auf die Darstellungen in den Datenbüchern verwiesen oder auf die umfangreiche Fachliteratur, die es zum Z80-Prozessor gibt.

## 4. Inbetriebnahmehinweise

-----

Für die Inbetriebnahme der Baugruppe ist lediglich die 8085-CPU auf der Prozessor-Baugruppe BFZ/MFA 2.1 gegen die Adapter-Platine auszutauschen. Dabei ist darauf zu achten, daß alle Steckerstifte korrekt in den 40poligen IC-Stecker auf der Basisplatine eingeführt werden. Im Anhang A3 ist die mit der Adapter-Platine bestückte CPU-Baugruppe abgebildet.

Neben dem Austausch der CPU ist das BOOT-EPROM 2716 mit dem BIOS-Programm für das CP/M-Betriebssystem auf der 64-K-RAM-Baugruppe BFZ/MFA 3.3 bzw. auf der ersten 16-K-RAM/EPROM-Baugruppe BFZ/MFA 3.2 auszutauschen (Anhang A4). Auf den Speicher-Baugruppen bleiben alle Brücken unverändert.

Nach dem Einschalten der Betriebsspannung muß sich das System wie in der ursprünglichen 8085-Konfiguration verhalten und das Betriebsprogramm von der Diskette lesen. Dies ist der Fall, wenn am Laufwerk A der Zugriff durch Aufleuchten der roten LED angezeigt wird. Nach dem Laden des Betriebsprogramms muß das MC-System die Bereitschaft durch Ausgabe der Kommandoanforderung " A > " anzeigen.

## Programmierbeispiele

---

Sollten hier Probleme auftreten, so führen Sie die gleichen Funktionskontrollen und Inbetriebnahmeschritte durch, wie sie in den Fachpraktischen Übungen zur Prozessor-Baugruppe, der Speicher-Baugruppe und zum CP/M-System beschrieben sind.

Sowohl die BUS-Signalanzeige wie auch der BUS-Signalgeber können für den Funktionstest wie in gewohnter Weise verwendet werden.

### 5.1 Z80-Programmbeispiel

---

Im Anhang A7 ist ein Programmbeispiel abgebildet, welches in der Mnemonik des Z80 geschrieben ist und mit dem Makro-Assembler M80 übersetzt wurde. Der Makro-Assembler M80 ist ein universeller Assembler, der von der Firma Microsoft zusammen mit den Hochsprachen vertrieben wird. Er ist beispielsweise in dem Microsoft-Entwicklungspaket BASIC-Interpreter/-Compiler enthalten und für die Einbindung von Assembler-Programnteilen in Hochsprachenprogrammen vorgesehen. Der M80 ermöglicht zusammen mit dem Linker (Binder) L80 die modulare Entwicklung von Programmen. Darüber hinaus bietet er die Möglichkeit, neben der 8080/85-Mnemonik auch Programnteile in der Z80-Mnemonik zu übersetzen, so daß mit dem gleichen Assembler in die Z80-Programmierung eingeführt werden kann.

Das vorliegende Beispiel realisiert einen Zähler an der Ausgabe-Baugruppe (BG-Nr. 00) der über einen Zeit-Interrupt von der Zähler- und Zeitgeber-Baugruppe gesteuert wird. Es wurde deshalb gewählt, um die mit dem 8085-Prozessor vergleichbaren Interrupt-Möglichkeiten durch die Hardware-Anpassung über den PAL-Baustein auf der NSC800-Adapter-Platine zu demonstrieren. Für den Lerner ist es günstig, die Interrupts zunächst manuell über den Handtaster in der Frontplatte der Zähler- und Zeitgeber-Baugruppe auszulösen, um die Funktion und Wirkungsweise des Programms zu testen. Dafür ist es günstig, wenn der Interrupt-Eingang der CPU auf eine Signalflanke reagiert. Das ist beim RSTA-Eingang des NSC800 nicht der Fall, wird aber mit Hilfe eines Schaltwerkes im PAL-Baustein realisiert.

Das Programm ist so dokumentiert, daß es hier nicht weiter beschrieben werden muß. Es wurde wie folgt mit Hilfe des Assemblers M80 und des Linkers L80 übersetzt:

```
A>M80 ZAEHLER,ZAEHLER=ZAEHLER
      !           !           !
      !           !           +--- Name der Quell-Datei
      !           !           z.B. ZAEHLER.MAC
      !           +----- Name der Object-Datei
      !           ZAEHLER.REL
      +----- Name der List-Datei
              ZAEHLER.PRN
```



## Programmier-Beispiele

Der Assembler erzeugt zunächst einen Maschinencode für die Start-Adresse 0000H. Diesen Code nennt man den relativen Maschinencode. Er wird vom M80 in der Object-Datei mit dem Datei-Namen ZAEHLER.REL abgelegt. Erst mit Hilfe des Linkers L80, dem man die wirkliche (absolute) Programmstart-Adresse mitteilen muß, wird der ablauffähige Programmcode erzeugt.

```

A>L80 /P:100,ZAEHLER,ZAEHLER/N/E
      !      !      !      ! !
      !      !      !      ! +-Beenden des LINK-Vorgangs
      !      !      !      !
      !      !      !      +---Abspeichern des Programm-
      !      !      !      codes
      !      !      +-----Name der Programm-Datei
      !      !      hier ZAEHLER.COM
      !      +-----Name der Object-Datei
      !      hier ZAEHLER.REL
      +-----Start-Adresse für den
      Programmcode (CP/M-
      Startadresse 0100H)

```

Das abgebildete Listing stellt den Inhalt der vom Assembler erzeugten List-Datei ZAEHLER.PRN dar.

### 5.2 Turbo-PASCAL-Programmbeispiel

Auf Personal- und Hobby-Computern ist zwar die Programmiersprache BASIC am weitesten verbreitet, weil sie zweifellos leicht erlernbar ist und seit je her als Interpreter ein interaktives Arbeiten erlaubt. Jedoch bieten die heute gebräuchlichen Compiler-Sprachen ebenfalls diese Möglichkeit. Dies verhalf insbesondere der Programmiersprache Turbo-PASCAL zu einem Durchbruch im Personal-Computerbereich. Leistungsfähigkeit, Einfachheit in der Bedienung sowie der günstige Preis führten dazu, daß Turbo-PASCAL heute ein Standard darstellt. Turbo-PASCAL wurde ursprünglich für CP/M-Systeme entwickelt. Die heute auf den IBM- und kompatiblen Personal-Computer eingesetzten Versionen sind bezüglich der Bedienoberfläche weiter verbessert und im Sprachumfang, angepaßt an die PC-Möglichkeiten, erweitert worden. Welche didaktischen Möglichkeiten bietet diese Programmiersprache?

Ein wesentliches Merkmal dieser wie vergleichbarer höherer Programmiersprachen ist das Prozedur-/Unterprogramm-Konzept. Programme werden vergleichbar mit der Assembler-Programmierung in übersichtliche und funktionsbezogene Einheiten, d.h. in Unterprogramme unterteilt, die später durch Angabe des Namens aufgerufen werden.

## Programmier-Beispiele

---

Ein PASCAL-Programm könnte beispielsweise folgende Form haben:

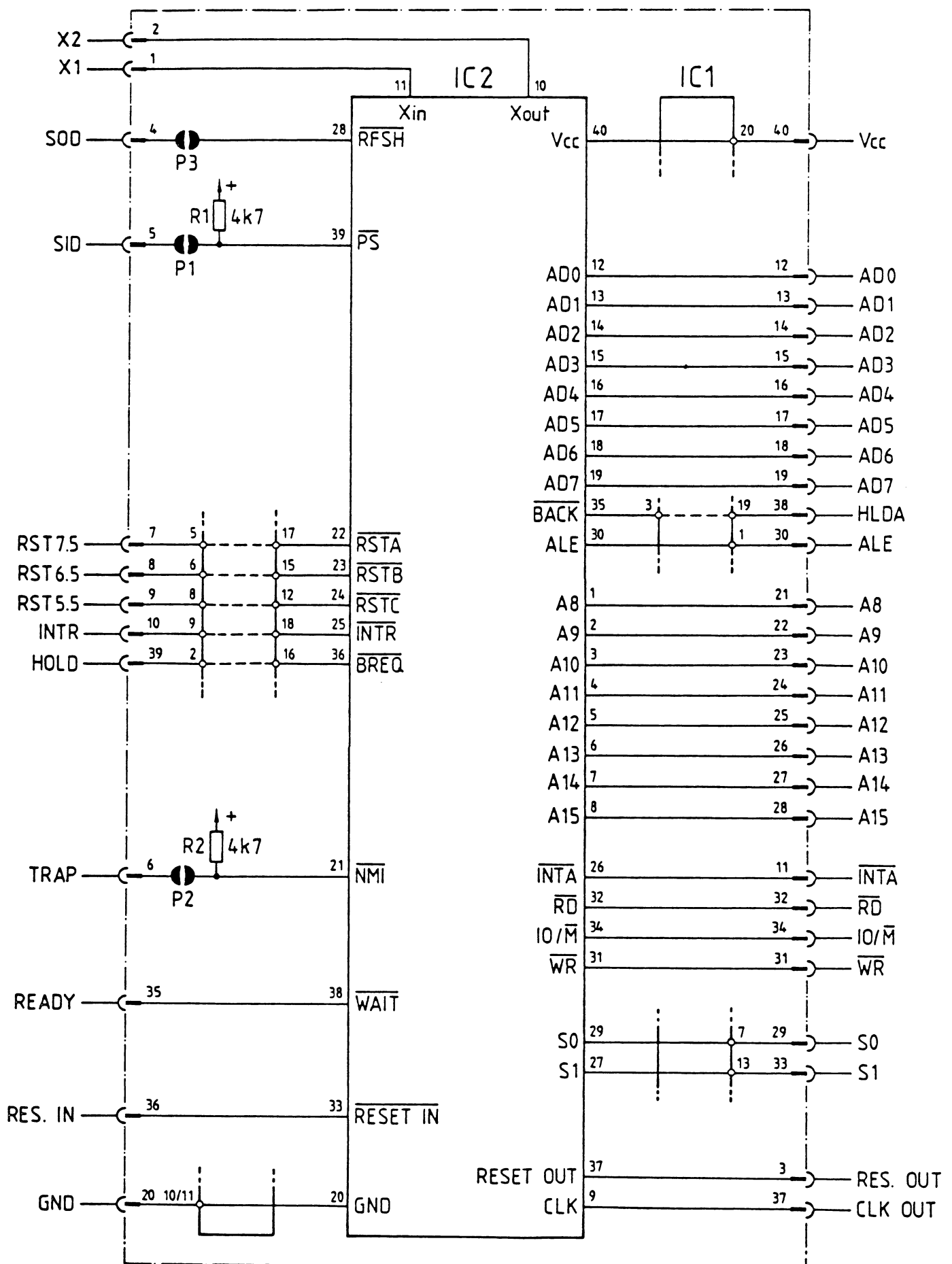
```
program MOTOR_STEUERUNG;  
(*$I UP.PAS *)  
begin  
    WARTE_AUF_START_TASTE;  
    MOTOR_EINSCHALTEN;  
    STARTE_LAUFZEIT;  
    WARTE_AUF_LAUFZEIT_ENDE;  
    MOTOR_AUSSCHALTEN;  
end.
```

Durch die Definition von Unterprogrammen, die in einer Datei (UP.PAS) abgelegt sind, wird die Funktion des Programms direkt lesbar. Hiermit bietet sich die Möglichkeit, dem Lernenden eine Reihe von vordefinierten Programmteilen "unsichtbar" zur Verfügung zu stellen, die er aber nutzen kann. Damit lassen sich Aufgabenbeispiele in die Vermittlung einbinden, die programmier-technisch zu kompliziert, aber für den Auszubildenden von Interesse sind. Verfolgt man dieses Konzept konsequent, so lassen sich auch Lernanfänger über einen anwendungsbezogenen Befehlssatz in Form dieser vordefinierten Unterprogramme in den Einsatz und die Handhabung von Computern einführen.

Turbo-PASCAL hat einen bildschirm-orientierten Editor integriert, der in seinen Funktionen dem Textverarbeitungsprogramm Wordstar entspricht. Das setzt allerdings ein Computer-System voraus, welches mit einem Daten-Terminal ausgestattet ist, das Steuerfunktionen für die Positionierung des Cursors auf dem Bildschirm ermöglicht. Diese Funktionen eines Terminals nennt man "direkte Cursor-Steuerung". Für das MFA-Mikrocomputer-System stehen mit dem Video-Interface BFZ/MFA 8.4 diese Funktionen zur Verfügung. Für den Einsatz von Turbo-PASCAL ist dieses Interface in der CP/M-Ausbaustufe erforderlich.

Bei dem nachfolgend abgebildeten Programmbeispiel, welches in der Programmiersprache Turbo-PASCAL realisiert wurde, handelt es sich wieder um den Zähler. Allerdings wurde hier die Interrupt-Verarbeitung so realisiert, daß die Interrupt-Service-Routine unter der Restart-Adresse 003CH lediglich aus dem RET-Befehl besteht und in dem Unterprogramm "warte\_auf\_interrupt" der Prozessor bis zu einem eintreffenden Interrupt-Impuls über einen HALT-Befehl gestoppt wird.

## Anhang A1: Stromlaufplan Adapter-Platine



## Anhang A1: Stromlaufplan Adapter-Platine

+-----+		
! IC1 !	IC2 !	
+-----+		
! GAL16V8 !	NSC800 !	
+-----+		
! +5 V !	20 !	40 !
+-----+		
! 0 V !	10,11 !	20 !
+-----+		

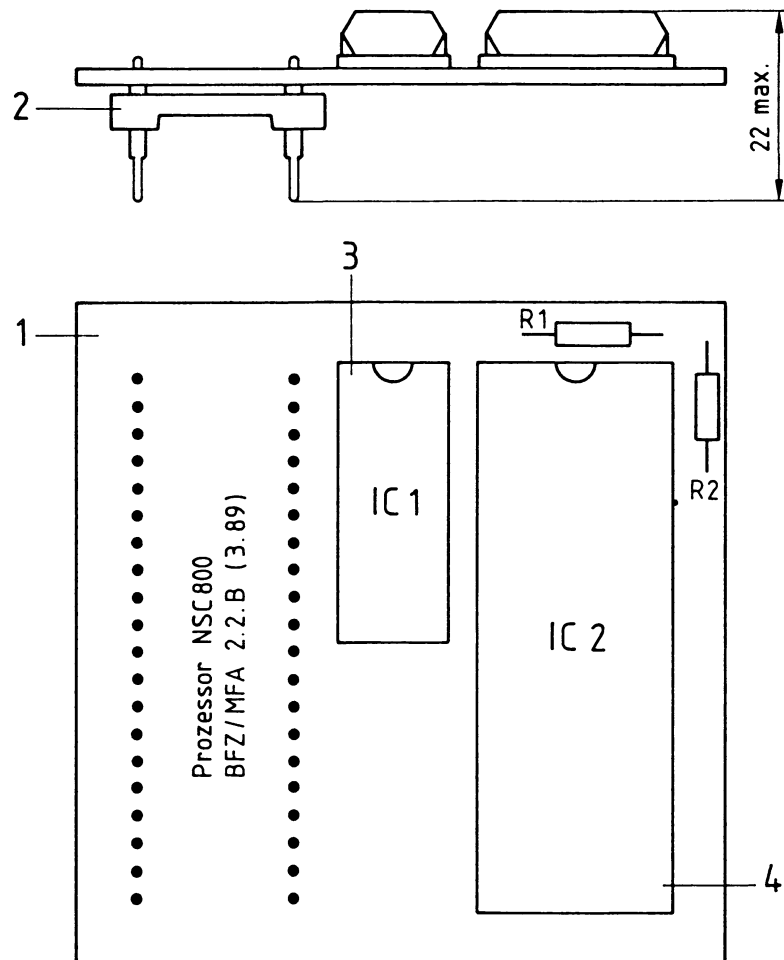
Anm.:

- 1) Der RST7.5-Eingang wirkt flankengesteuert; die Eingangsschaltung zum RSTA-Eingang ist vereinfacht dargestellt!
- 2) Der NMI-Eingang NSC800 ist aktiv Low, so daß gegenüber der 8085-CPU hier die Bezeichnung /TRAP angegeben ist!

## Bauteilliste

Bez.	! Art	! Bemerkung
+-----+		
IC1	! GAL 16V8	!
	!	!
IC2	! Mikroprozessor NSC800	! fmin 2MHz
	!	!
R1,R2	! Widerstand 4,7KOhm	! 0,25W

## Anhang A2: Bestückungsplan Adapter-Platine



## Stückliste Leiterplatte

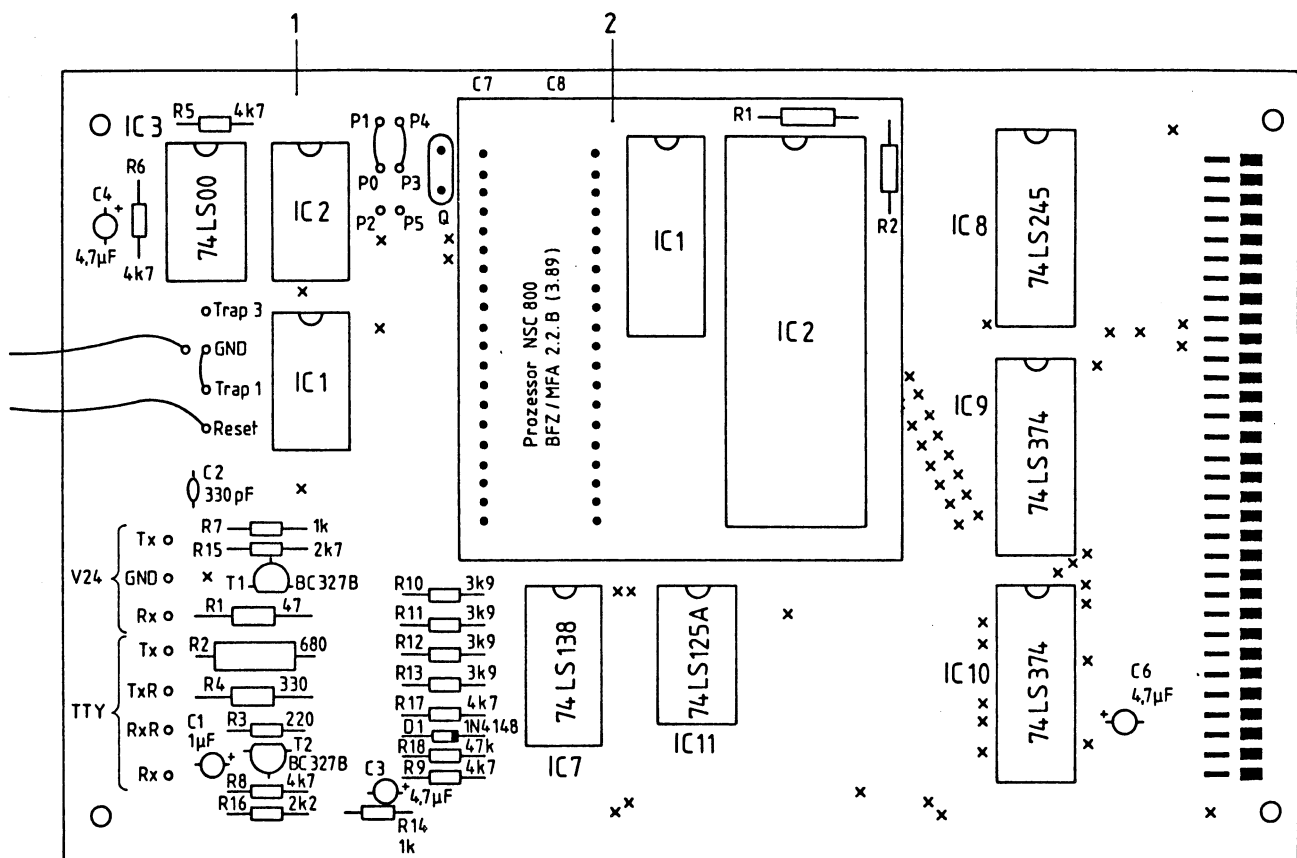
Pos.	Stckz.	Benennung/Daten	Bemerkung
1	1	Leiterplatte BFZ/MFA 2.2	
2	1	Leiterplatte Dual-in-line-Verbindungsstecker 40polig	z.B. DILV4006ZX1 Fa. Fischer Metroplast
3	1	IC-Fassung 20polig	flache Bauform oder
4	1	IC-Fassung 40polig	unbestückt



## Anhang A3: Bestückungsplan CPU-Baugruppe

Siehe auch: Fachpraktische Übung Band 4  
 CP/M-Ausbaustufe 1 BFZ/MFA 7.3.1  
 Aufbau und Inbetriebnahme

Seite 20: A5 - Sichtkontrolle/Modifikation der  
 Baugruppe Prozessor 8085



(IC1 und IC2 sind zu entfernen)

## Stückliste

Pos.	Stckz.	Benennung/Daten	Bemerkung
1	1	Prozessor-Baugruppe BFZ/MFA 2.1	bestückt und geprüft
2	1	NSC800 Aapater-Platine BFZ/MFA 2.2	bestückt und geprüft, alle Lötbrücken ge- öffnet

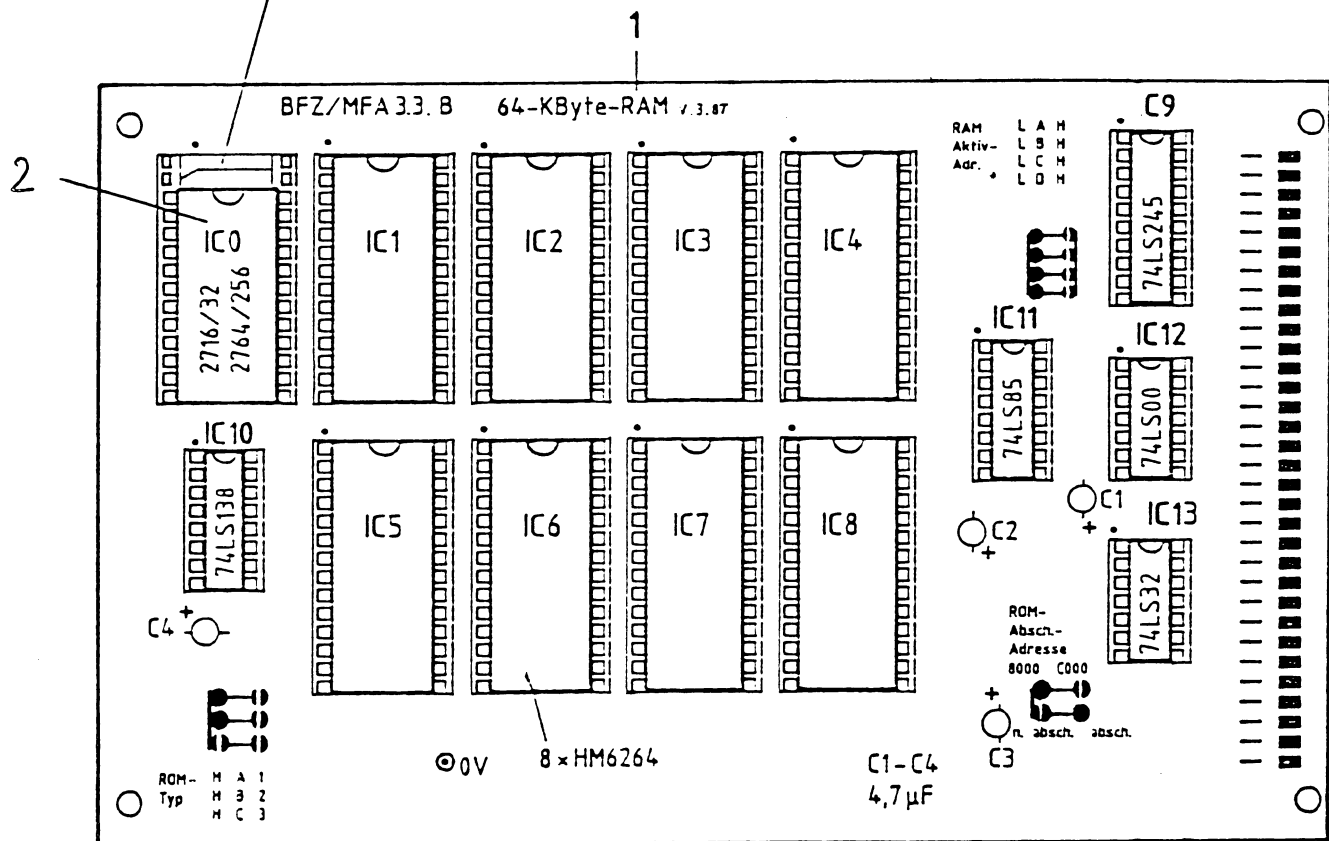
## Anhang A4: Bestückungsplan 64-K-RAM-Baugruppe

Siehe auch: Fachpraktische Übung Band 4  
 CP/M-Ausbaustufe 1 BFZ/MFA 7.3.1  
 Aufbau und Inbetriebnahme

Seite 34: A10 - Sichtkontrolle 64-K-RAM-Baugruppe

Seite 30: A 9 - Sichtkontrolle 16-K-RAM/EPROM-Baugruppe

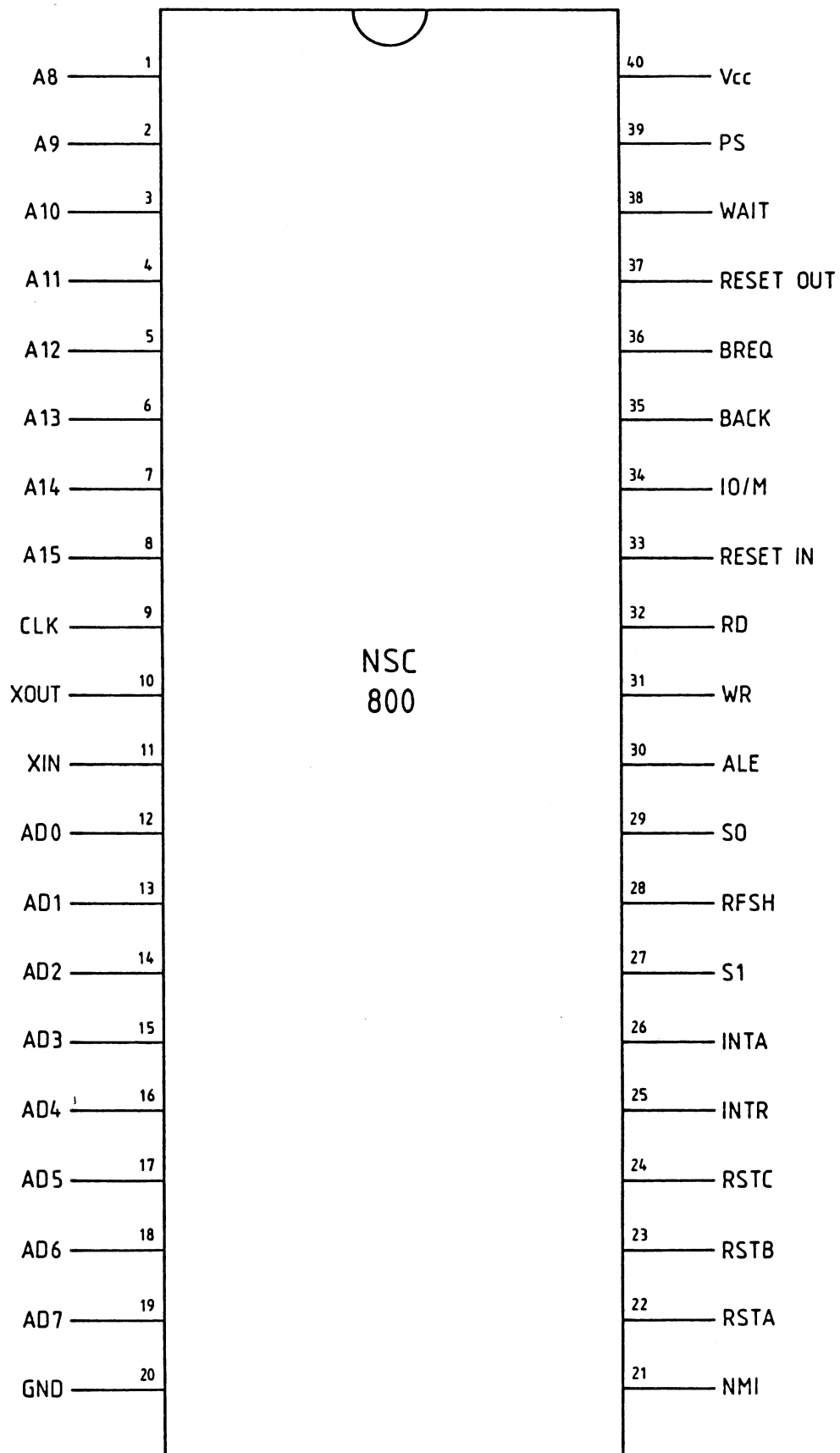
Bei Bestückung mit EPROM 2716 bleiben die  
 Anschlüsse 1,2,27 und 28 der Fassung frei



## Stückliste

Pos.	Stckz.	Benennung/Daten	Bemerkung
1	1	64-K-RAM-Baugruppe BFZ/MFA 3.3	bestückt und geprüft
2	1	BOOT-EPROM 2716 mit BIOS- Programm, angepaßt an NSC800-Prozessor	

## Anhang A5: Anschlußplan NSC800



## Anhang A6: Befehlsliste NSC800/Z80

MNEMOCODE	WIRKUNG	ERLÄUTERUNGEN
=====		
8-Bit-Ladebefehle		
-----		
LD r,s	<r>:=<s>	<s> = <r>,d8,<<HL>>,<<xy+offs>>
LD d,r	<d>:=<r>	d = r,<HL>,<xy+offs>
LD d,d8	<d>:=d8	d = <HL>,<xy+offs>
LD A,s	<A>:=<s>	s = <BC>,<DE>,I,R,adr
LD d,A	<d>:=A	d = <BC>,<DE>,I,R,adr
16-Bit-Ladebefehle		
-----		
LD dd,d16	<dd>:=d16	dd = BC,DE,HL,SP,xy
LD dd,(adr)	<dd>:=<adr+1><adr>	
LD (adr),ss	<adr+1><adr>:=<ss>	ss = BC,DE,HL,SP,xy
LD SP,ss	<SP>:=<ss>	ss = HL,xy
PUSH ss	<<SP>-1>:=hi(<ss>) <<SP>-2>:=lo(<ss>)	ss = BC,DE,HL,AF,xy
POP dd	lo(<dd>):=<<SP>> hi(<dd>):=<<SP>+1>	dd = BC,DE,HL,AF,xy
Austauschbefehle		
-----		
EX DE,HL	<DE> <--> <HL>	
EX AF,AF'	<AF> <--> <AF'>	
EXX	<BCDEHL> <--> <BCDEHL'>	
EX (SP),ss	<<SP>> <--> lo(<ss>) <<SP>+1> <--> hi(<ss>)	ss = HL,xy
Blocktransferbefehle		
-----		
LDI	<<DE>>:=<<HL>>,<ss>:=<ss>+1 <BC>:=<BC>-1	Byte ss = HL,DE
LDIR	<<DE>>:=<<HL>>,<ss>:=<ss>+1 <BC>:=<BC>-1 bis <BC>=0	Block
LDD	<<DE>>:=<<HL>>,<ss>:=<ss>-1 <BC>:=<BC>-1	Byte
LDDR	<<DE>>:=<<HL>>,<ss>:=<ss>-1 <BC>:=<BC>-1 bis <BC>=0	Block
Vergleichsbefehle		
-----		
CPI	<HL>:=<HL>+1,<BC>:=<BC>-1	Vergleich von <A> mit <<HL>>
CPIR	<HL>:=<HL>+1,<BC>:=<BC>-1 bis <BC>=0 oder <A>=<<HL>>	
CPD	<HL>:=<HL>-1,<BC>:=<BC>-1	<A>-<<HL>> beeinflusst Flags <A> bleibt unverändert
CPDR	<HL>:=<HL>-1,<BC>:=<BC>-1 bis <BC>=0 oder <A>=<<HL>>	

## Anhang A6: Befehlsliste NSC800/Z80

## 8-Bit-ALU-Befehle

ADD	s	$\langle A \rangle := \langle A \rangle + \langle s \rangle$	$\langle s \rangle = \langle r \rangle, d8, \langle \langle HL \rangle \rangle$ , auch ADD A,s
ADC	s	$\langle A \rangle := \langle A \rangle + \langle s \rangle + \langle CY \rangle$	$\langle \langle xy+offs \rangle \rangle$ auch ADC A,s
SUB	s	$\langle A \rangle := \langle A \rangle - \langle s \rangle$	auch SUB A,s
SBC	s	$\langle A \rangle := \langle A \rangle - \langle s \rangle - \langle CY \rangle$	auch SBC A,s
AND	s	$\langle A \rangle := \langle A \rangle \text{ and } \langle s \rangle$	bitweise Operation
OR	s	$\langle A \rangle := \langle A \rangle \text{ or } \langle s \rangle$	
XOR	s	$\langle A \rangle := \langle A \rangle \text{ xor } \langle s \rangle$	
CP	s	$\langle A \rangle := \langle A \rangle$	$\langle A \rangle - \langle s \rangle$ beeinflusst Flags
INC	d	$\langle d \rangle := \langle d \rangle + 1$	$d = r, \langle HL \rangle, \langle xy+offs \rangle$
DEC	d	$\langle d \rangle := \langle d \rangle - 1$	

## 16-Bit-ALU-Befehle

ADD	dd,ss	$\langle dd \rangle := \langle dd \rangle + \langle ss \rangle$	dd = xy, ss = BC,DE,SP
ADC	HL,ss	$\langle HL \rangle := \langle HL \rangle + \langle ss \rangle + \langle CY \rangle$	ss = HL,BC,DE,SP
SBC	HL,ss	$\langle HL \rangle := \langle HL \rangle - \langle ss \rangle - \langle CY \rangle$	
INC	ss	$\langle ss \rangle := \langle ss \rangle + 1$	ss = BC,DE,SP,xy
DEC	ss	$\langle ss \rangle := \langle ss \rangle - 1$	

## Konvertierbefehle

DAA		BCD-Korrektur von A	
CPL		$\langle A \rangle := \neg \langle A \rangle$	$\neg \langle A \rangle$ bitweise Negation
NEG		$\langle A \rangle := \neg \langle A \rangle + 1$	
CCF		$\langle CY \rangle := \neg \langle CY \rangle$	
SCF		$\langle CY \rangle := 1$	

## Bitbefehle

BIT	b,s	$\langle Z \rangle := b\text{-tes Bit von } s$	$s = r, \langle HL \rangle, \langle xy+offs \rangle$
SET	b,s	$b\text{-tes Bit von } s := 1$	
RES	b,s	$b\text{-tes Bit von } s := 0$	

## Steuerbefehle

NOP		keine Operation	$\langle P \rangle := \langle PC \rangle + 1$
HALT		HALT-Zustand	
DI		Interrupts sperren	
EI		Interrupts freigeben	
IM0		Interruptmode 0	8080-Modus
IM1		Interruptmode 1	INT --> CALL 0038H
IM2		Interruptmode 2	INT --> CALL $\langle I \rangle \langle d8 \rangle$ $\langle d8 \rangle$ vom IO-Baustein

## Anhang A6: Befehlsliste NSC800/Z80

## Verschiebebefehle

-----			
RLC	s	$\begin{array}{c} +-----+ \\ \text{CY} \leftarrow + \leftarrow / \dots s \dots / \leftarrow + \end{array}$	s = r, <HL>, <xy+offs>
RL	s	$\begin{array}{c} +-----+ \\ + \leftarrow \text{CY} \leftarrow \leftarrow / \dots s \dots / \leftarrow + \end{array}$	
RRC	s	$\begin{array}{c} +-----+ \\ \text{CY} \leftarrow + \rightarrow / \dots s \dots / \rightarrow + \end{array}$	
RR	s	$\begin{array}{c} +-----+ \\ + \leftarrow \text{CY} \leftarrow \rightarrow / \dots s \dots / \rightarrow + \end{array}$	
SLA	s	$\text{CY} \leftarrow \text{---} / \dots s \dots / \leftarrow 0$	
SRA	s	$\begin{array}{c} +-----+ \\ + \rightarrow \text{CY} + \rightarrow / \dots s \dots / \rightarrow + \\ + \text{---} + \end{array}$	
SRL	s	$\begin{array}{c} +-----+ \\ + \rightarrow \text{CY} 0 \rightarrow / \dots s \dots / \rightarrow + \end{array}$	
RLD			Rotieren von Tetraden
RRD			

## Ein- und Ausgabebefehle

-----			
IN	A, (p)	<A> := <p>	
IN	r, (C)	<r> := <<C>>	<C> = Port-Adresse
INI		<<HL>> := <<C>>, <HL> := <HL> + 1	Byte
		<B> := <B> - 1	
INIR		<<HL>> := <<C>>, <HL> := <HL> + 1	Block
		<B> := <B> - 1 bis <B> = 0	
IND		<<HL>> := <<C>>, <HL> := <HL> - 1	Byte
		<B> := <B> - 1	
INDR		<<HL>> := <<C>>, <HL> := <HL> - 1	Block
		<B> := <B> - 1 bis <B> = 0	
OUT	(p), A	<p> := <A>	
OUT	(C), A	<<C>> := <A>	
OUTI		<<C>> := <<HL>>, <HL> := <HL> + 1	Byte
		<B> := <B> - 1	
OUTIR		<<C>> := <<HL>>, <HL> := <HL> + 1	Block
		<B> := <B> - 1 bis <B> = 0	
OUTD		<<C>> := <<HL>>, <HL> := <HL> - 1	Byte
		<B> := <B> - 1	
OUTDR		<<C>> := <<HL>>, <HL> := <HL> - 1	Block
		<B> := <B> - 1 bis <B> = 0	

## Anhang A6: Befehlsliste NSC800/Z80

## Sprungbefehle

JP	adr	<PC>:=adr	
JP	cc,adr	wenn Bedingung erfüllt	cc = NZ,Z,NC,C,PO,PE,
JR	offs	<PC>:=<PC>+offs	P,M
JR	kk,offs	wenn Bedingung erfüllt	kk = NZ,NC,Z,C
JP	(ss)	<PC>:=<ss>	ss = HL,xy
DJNZ	offs	<B>:=<B>-1 und <PC>:=<PC>+offs	
		wenn <B> ungleich 0	
CALL	adr	<<SP>-1><<SP>-2>:=<PC>	
		und <PC>:=adr	
CALL	cc,adr	wenn Bedingung erfüllt	
RST	n	<<SP>-1><<SP>-2>:=<PC>	
		und <PC>:=n*8	n = 0,1,2,3,4,5,6,7
RET		<PC>:=<<SP>><<SP>+1>	
RET	cc	wenn Bedingung erfüllt	
RETI		Rückkehr von Interrupt	
		Int.-Flipflop stellen	
RETN		Rückkehr vom NMI	

## Verwendete Kurzzeichen:

< >	= Inhalt von	d8	= 8-Bit-Wert
:=	= Wertzuweisung	d16	= 16-Bit-Wert
r	= A,B,C,D,E,H,L	adr	= Adresse
xy	= IX oder IY	hi( )	= High-Byte
offs	= Offset	lo( )	= Low-Byte
p	= Port-Adresse		

## Anhang A7: Z80-Assemblerprogramm-Listing

```

;-----
;      NSC800-/Z80 PROGRAMM-BEISPIEL
;      Z A E H L E R
;
; ASSEMBLIERT MIT DEM M80-ASSEMBLER
;
; BENOETIGTE BAUGRUPPEN/KONFIGURATION:
;
; CP/M-AUSBAUSTUFE DES MFA-SYSTEMS
; MIT NSC800-ADAPTER-PLATINE
; SOWIE:
; ZAEHLER UND ZEITGEBER      BFZ/MFA 4.6
; 8-BIT-AUSGABE             BFZ/MFA 4.1
;
; FUNKTION:
; IM RHYTHMUS VON 1 SEKUNDE WIRD DAS
; AUSGABE-BIT-MUSTER AN DER AUSGABE-
; BAUGRUPPE INTERRUPT-GESTEUERT HOCH-
; GEZAEHLT
;
; AUTOR: F. DERRIKS 5/1989
;-----

.Z80                                ;Z80-MNEMONIK

003C      INT75      EQU      003CH      ;RSTA-/RST7.5-EINSPRUNG-
;                                           ;ADRESSE

0063      TIMSTWR    EQU      063H      ;TIMER-BAUGRUPPEN-NUMMERN
0060      TIM0       EQU      060H
0061      TIM1       EQU      061H

0036      TSTW0      EQU      036H      ;TIMER-STEUERWORTE
0076      TSTW1      EQU      076H

;                                           ;TEILER-FAKTOR TIMER 0
00D0      TL0LB      EQU      0DOH      ;07DOH -> 2000:1
0007      TLOHB      EQU      007H      ;1 MSEC - IMPULSE
;                                           ;TEILER-FAKTOR TIMER 1
00E8      TL1LB      EQU      0E8H      ;03E8H -> 1000:1
0003      TL1HB      EQU      003H      ; 1 SEC - IMPULSE

;-----
; DAS HAUPT-PROGRAMM BESTEHT LEDIGLICH AUS DEM
; AUFRUF DES INITIALISIERUNGS-PROGRAMMS, DER
; INTERRUPT-FREIGABE SOWIE EINER PROGRAMM-SCHLEIFE,
; BEI DER DER PROZESSOR BIS ZUM EINTREFFENDEN
; INTERRUPT-IMPULS VERHARRT,
; NACH AUSFUEHRUNG DER INTERRUPT-SERVIVE-
; ROUTINE WIRD DIE SCHLEIFE ERNEUT GESTARTET

0000'     CD 0012     START:      CALL      INIT      ;INITIALISIERUNG
0003'     FB                                EI
0004'     C3 0004     LOOP:      JP        LOOP      ;WARTE BIS INTERRUPT

```



## Anhang A7: Z80-Assemblerprogramm-Listing

```

;-----
; INTERRUPT-SERVIVE-ROUTINE, DIE IM RHYTHMUS DER
; INTERRUPT-IMPULSE VOM TIMER ANGESPRUNGEN UND
; AUSGEFUEHRT WIRD

0007'  3A 004A  ISR:      LD      A,(ZAEHLER)      ;ALTEN ZAEHLWERT
000A'  3C              INC      A              ;INKREMENTIEREN
000B'  32 004A      LD      (ZAEHLER),A      ;UND SICHERN
000E'  D3 00        OUT      (0),A          ;SOWIE ANZEIGEN
0010'  FB          EI              ;INTERRUPTS FREIGEBEN
0011'  C9          RET              ;UND ZURUECK ZUM HAUPT-
;PROGRAMM

;-----
; INITIALISIERUNGS-PROGRAMM FUER INTERRUPT, TIMER-
; BAUGRUPPE UND RAM-ZAEHLER

0012'  CD 0021  INIT:      CALL    INTIN      ;INTERRUPT-INITIALISIERUNG
0015'  CD 0031      CALL    TIMIN      ;TIMER-INITIALISIERUNG
0018'  CD 001C      CALL    INITZ      ;ZAEHLER-INITIALISIERUNG
001B'  C9          RET

;-----
; INITIALISIERUNG DES RAM-ZAEHLERS, DER DAS
; AUSGANGS-BITMUSTER SPEICHERT

001C'  AF          INITZ:   XOR      A
001D'  32 004A      LD      (ZAEHLER),A
0020'  C9          RET

;-----
; INITIALISIERUNG DES RSTA-/RST7.5-INTERRUPTS
; DURCH SETZEN DES MASKEN-BITS SOWIE DURCH EIN-
; TRAGEN DES SPRUNGBEFEHLS ZUR INTERRUPT-SERVIVE-
; ROUTINE IM SPEICHER UNTER DER ZUM RSTA GEHOEREN-
; DEN RESTART-ADRESSE 003CH

0021'  3E 08  INTIN:      LD      A,08H      ;INT.-MASKE FUER
0023'  D3 BB      OUT      (0BBH),A      ;RSTA/RST7.5
0025'  3E C3      LD      A,0C3H      ;JMP-BEFEHL NACH
0027'  32 003C      LD      (INT75),A      ;003CH
002A'  21 0007      LD      HL,ISR
002D'  22 003D      LD      (INT75+1),HL
0030'  C9          RET

```

## Anhang A7: Z80-Assemblerprogramm-Listing

```

;-----
; INITIALISIERUNG DER ZAEHLER- UND ZEITGEBER-BAU-
; GRUPPE, VERWENDET WERDEN DER TIMER 0 UND 1, UM
; DEN 2 MHZ - SYSTEMTAKT AUF 1 HZ HERUNTERZUTEILEN,
; DAZU MUESSEN BEIDE TEILER HINTEREINANDER GESCHAL-
; TET WERDEN, INDEM DER AUSGANG DES TIMERS 0 (OUT0)
; MIT DEM EINGANG DES TIMER 1 (CLK 1) VERBUNDEN
; WIRD
; DARUEBER HINAUS MUSS DER AUSGANG DES TIMERS 1 UEBER
; STECKER St1 UND St2 AUF DER BAUGRUPPE MIT DER
; INTERRUPT-LEITUNG RST7.5 (STIFT 25a) DES BUS-
; SYSTEMS VERBUNDEN WERDEN
; IM TESTFALL KANN DER TIMER-IMPULS VON HAND
; AUSGELOEST WERDEN, INDEM UEBER DEN STECKER St3
; DER HAND-INTERRUPT SELEKTIERT WIRD

0031' 3E 36      TIMIN:  LD      A,TSTWO      ;TIMER 0 AUF 1 MSEC-
0033' D3 63      OUT      (TIMSTWR),A      ;IMPULSE
0035' 3E D0      LD      A,TLOLB
0037' D3 60      OUT      (TIM0),A
0039' 3E 07      LD      A,TLOHB
003B' D3 60      OUT      (TIM0),A
003D' 3E 76      LD      A,TSTW1      ;TIMER 1 AUF 1 SEC-
003F' D3 63      OUT      (TIMSTWR),A      ;IMPULSE
0041' 3E E8      LD      A,TL1LB
0043' D3 61      OUT      (TIM1),A
0045' 3E 03      LD      A,TL1HB
0047' D3 61      OUT      (TIM1),A
0049' C9        RET

;-----
; RAM-SPEICHERSTELLE, IN DER DIE ANZAHL DER EINGEGAN-
; GENEN INTERRUPT-IMPULSE FESTGEHALTEN WIRD

004A' 00        ZAEHLER:  DEFB      0

                        END

```

## Symbols:

INIT	0012'	INITZ	001C'	INT75	003C	INTIN	0021'
ISR	0007'	LOOP	0004'	START	0000'	TIM0	0060
TIM1	0061	TIMIN	0031'	TIMSTW	0063	TLOHB	0007
TL0LB	00D0	TL1HB	0003	TL1LB	00E8	TSTW0	0036
TSTW1	0076	ZAEHLE	004A'				

No Fatal error(s)

## Anhang A8: Turbo-PASCAL-Programmlisting

```

(*-----*)
(* Turbo-PASCAL Programm-Beispiel: Interrupt-gesteuerter *)
(* Zaehler, im 1 Sekunden-Rhythmus ueber Timer-Interrupt- *)
(* Impuls am RSTA-(RST7.5) Eingang des Prozessors *)
(* Autor: F.Derriks 5/1989 *)
(*-----*)
program zaehler(input,output);
(*-----*)
    const
        output_0      = 0;
    var
        zaehler_stand : byte;

(*----- U n t e r p r o g r a m m e -----*)
    procedure interrupt_init;
(*-----*)
        const
            int_contl_reg  = $BB;
            int_maske      = $08;
            restart_adr     = $003C;
            ret_instr_code  = $C9;
        begin
            mem[restart_adr] := ret_instr_code;
            port[int_contl_reg] := int_maske;
        end;

(*-----*)
    procedure timer_init;
(*-----*)
        const
            timer_contl_reg = $63;
            timer_0         = $60;
            timer_1         = $61;
            timer_stw_0     = $36;
            timer_stw_1     = $76;
            teiler_0        = 2000;
            teiler_1        = 1000;
        begin
            port[timer_contl_reg] := timer_stw_0;
            port[timer_0]         := lo(teiler_0);
            port[timer_0]         := hi(teiler_0);
            port[timer_contl_reg] := timer_stw_1;
            port[timer_1]         := lo(teiler_1);
            port[timer_1]         := hi(teiler_1);
        end;

(*-----*)
    procedure warte_auf_interrupt;
(*-----*)
        begin
            inline ($FB/      (* EI  *)
                    $76);    (* HALT *)
        end;

```

Anhang A8: Turbo-PASCAL-Programmlisting

---

```
(*----- H a u p t p r o g r a m m -----*)
begin;
  timer_init;
  interrupt_init;
  zaehler_stand := 0;
  repeat
    warte_auf_interrupt;
    port[output_0] := zaehler_stand;
    zaehler_stand := zaehler_stand + 1;
  until keypressed;
end.
```

---

Anhang A9: Anpassen des Programms BFZFORM.COM an den NSC800

---

Das Formatierungs-Programm BFZFORM.COM zum MFA-CP/M-System muß an den Prozessor NSC800 angepaßt werden. Diese Anpassung kann mit Hilfe des CP/M-Debuggers DDT durchgeführt werden. Die auszuführenden Schritte sind im folgenden protokolliert, wobei zur Änderung des Programms das S-Kommando verwendet wurde.

```
A>DDT BFZFORM.COM      <---- Aufruf des DDT und Laden des Programms
DDT VERS 2.2           BFZFORM.COM in den TPA-Bereich
NEXT PC
0D80 0100
-S066A                <---- Änderungen bei Adresse 066A beginnen
066A 3E CD            <---- der jeweilige Inhalt der Speicherstelle
066B 0E 5D            wird vom DDT angezeigt, die Eingabe-
066C 30 0C            Daten überschreiben diesen
066D E1 .             <---- Abbruch der Änderung
-S0C4C                <---- erneute Änderung ab Adresse 0C4C
0C4C 46 4E
0C4D 4F 49
0C4E 52 43
0C4F 4D 48
0C50 41 54
0C51 54 20
0C52 49 46
0C53 45 4F
0C54 52 52
0C55 45 4D
0C56 4E 41
0C57 20 54
0C58 4E 49
0C59 49 45
0C5A 43 52
0C5B 48 54
0C5C 54 24
0C5D 20 3E
0C5E 4D 02
0C5F 4F D3
0C60 45 BB
0C61 47 C9
0C62 4C 00
0C63 49 00
0C64 43 00
0C65 48 00
0C66 24 00
0C67 5A              <---- Abbruch der Änderung
-^C                  <---- Rückkehr zum Betriebssystem
A>SAVE 13 NSCFORM.COM <---- Speichern des geänderten Programms
A>                  unter dem Namen NSCFORM.COM
```

Anhang A10: Literaturhinweise

---

NSC800 Microprocessor Family  
Databook  
National Semiconductor Corporation

MSC-80/85 Family User's Manual  
Intel Corporation

Mikrocomputer Bausteine  
Mikroprozessor-System SAB 8085  
Siemens Datenbuch

Microsoft Utility Software Package  
for CP/M 80  
Assembler, Linker, Cross Referenz  
Facility, Library Manager

Z80 Assembly Language Subroutines  
L.A. Leventhal, W. Saville  
Osborne/McGraw-Hill

Turbo-PASCAL Reference Manual  
Borland International  
Scotts Valley, California

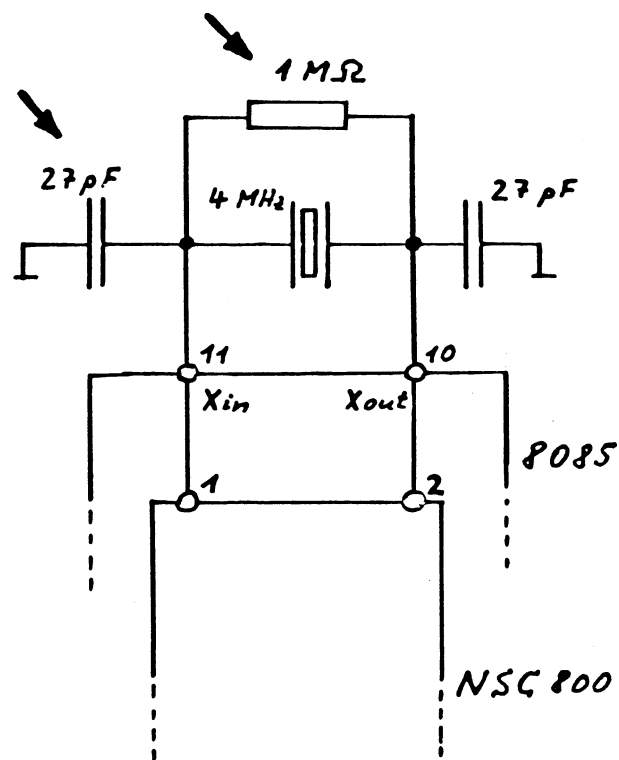
Programmierung des Z80  
Rodnay Zaks  
Sybex-Verlag, Düsseldorf

Turbo-PASCAL Top Training  
Jürgen Handke  
Klett-Verlag, Stuttgart

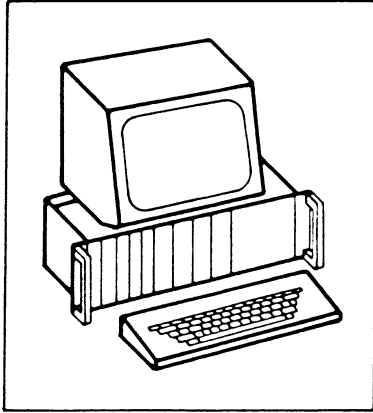
### Abhilfe bei Startproblemen

In Einzelfällen hat es sich gezeigt, daß der Oszillator des NSC800-Prozessors nach dem Einschalten der Betriebsspannung Anschwingprobleme hat, so daß zum Start nochmals die Reset-Taste betätigt werden muß.

In hartnäckigen Fällen ist Abhilfe auf folgende Art möglich: Die beiden mitgelieferten Teile (ein Kondensator mit 27 pF und ein Widerstand mit 1 MOhm) werden direkt am 2-MHz-Quarz auf die Prozessor-Hauptplatine gelötet, wie in der Skizze dargestellt:



# FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



GAL-Programmierer







## Inhalt

1.0	<b>Einführung</b>	1
2.0	<b>Funktionsbeschreibung</b>	2
2.1.0	Aufgabe des bidirektionalen BUS-Treibers	2
2.2.0	Der elektronische Schalter für die Versorgungsspannung	4
2.3.0	Erzeugung der Editierspannung	5
2.3.1	Der Rechteckgenerator	5
2.3.2	Die Spannungsvervielfachung	8
2.3.3	Elektronischer Schalter und Stabilisierung für die Editierspannung	9
3.0	<b>Bereitstellungsliste</b>	11
4.0	<b>Überprüfung der Baugruppe GAL-Programmierinterface</b>	15
4.1	Überprüfung der benötigten An- schlüsse PORT A	16
4.2	Überprüfung der benötigten An- schlüsse PORT B	17
4.3	Überprüfung der Funktionen, die durch das PORT C des Schnittstellenbausteins gesteuert werden	18
4.3.1	Schalten der Betriebsspannung VCC	18
4.3.2	Erzeugen und Schalten der Editier- spannung	19
4.3.3	Funktion der Schreib-/Lese-Umschaltung	19
4.3.4	Erzeugung des Strobe-Signals	19
4.3.5	Überprüfung der Eingangsanschlüsse von PORT C	20
5.0	<b>Einsatz der Programmiersoftware</b>	21
5.1	Hardwarevoraussetzungen	21
5.2	Programmaufruf	21

6.0	Beschreibung der GAL-Programmier-Kommandos	22
6.1	Das Editierkommando EDIT	22
6.1.1	Editieren der UND-Verknüpfungen	22
6.1.2	Editieren des Ausgangs-Steuerwortes	23
6.1.3	Editieren der Bausteinbezeichnung	25
6.1.4	Editieren des Securitybit	26
6.2	Das Programmierkommando PROG	27
6.3	Das Kommando DELETE	27
6.4	Das HELP-Kommando	28
6.5	Das TEST-Kommando	28
6.6	Das READ-Kommando	29
6.7	Das Kommando LPRINT	29
6.8	Das VERIFY-Kommando	30
6.8.1	Die Fehlermeldungen beim VERIFY-Kommando	30
6.8.1.1	Unterschied in der Bausteinbezeichnung	30
6.8.1.2	Unterschied im Ausgangssteuerwort	31
6.8.1.3	Unterschied im AND-ARRAY	31
6.9	Das Kommando CHANGE	32
6.10	Das Kommando NEW	32
6.11	Das Kommando QUIT	33

**Wichtiger Hinweis !**

Als Ergänzung zu diesen Unterlagen empfehlen wir das

FRANZIS - Arbeitsbuch 'GALs'

mit der Bestell-Nr. 3-7723-5902-7 (28,- DM).

Dieses Buch beschreibt die theoretischen Grundlagen der 'GALs' und eine Reihe praktischer Übungen.

## 1.0 Einführung

In der letzten Zeit findet man immer häufiger Schaltungen, die programmierbare Logikbausteine, kurz PLD (programmable logic devices) enthalten. Bei diesen handelt es sich um vom Anwender zu programmierende Bausteine, deren Funktion durch einen Programmervorgang festgelegt wird. Inzwischen gibt es solche Bausteine von den verschiedensten Herstellern in unterschiedlichen Ausführungen und Bezeichnungen. Die gebräuchlichsten Bezeichnungen sind z.Z.:

PLD = Programmable Logic Device

PAL = Programmable Array Logic

GAL = Generic Array Logik

Allen gemeinsam ist die Eigenart, daß sie für die kompliziertesten Logikschaltungen programmiert werden können, die sich aus den Grundverknüpfungen UND, ODER und NICHT realisieren lassen. Die meisten der am Markt erhältlichen PLD enthalten programmierbare UND-, ODER- sowie ANTIVALENZ-Verknüpfungen.

Um solche Bausteine sinnvoll programmieren zu können, sind gute Kenntnisse in der Digitaltechnik erforderlich.

Für die Programmierung benötigt man ein entsprechendes Programmiergerät und die für den jeweiligen Baustein erforderliche Software. Im folgenden wird der Aufbau eines Programmierinterfaces als zusätzliche Schnittstelle zum MFA-Mikrocomputer zur Programmierung von Bausteinen des Typs GAL 16 V 8 beschrieben.

**GAL-Programmierinterface**

---

**2.0 Funktionsbeschreibung.**

Das GAL-Programmierinterface basiert auf der Schaltung der Programmierbaren Parallelschnittstelle Bfz/MFA 4.3. Die Funktion der Programmierbaren Parallelschnittstelle ist in der zugehörigen Fachpraktischen Übung beschrieben, so daß hier lediglich auf die für den Einsatz als GAL-Programmierinterface notwendigen Schaltungserweiterungen eingegangen wird.

Diese Erweiterungen der Parallelschnittstelle sind:

1. Ein bidirektionaler Bus-Treiberbaustein 74LS245.
2. Einen Schaltungsteil, mit dem die Spannung von 5V elektronisch geschaltet werden kann.
3. Einen Schaltungsteil zur Erzeugung der notwendigen Editierspannung.
4. Für den Programmiervorgang die benötigte mechanische Aufnahme.

**2.1.0 Aufgabe des bidirektionalen Bus-Treibers.**

Aufgabe des bidirektionalen Bus-Treibers ist es, den Datenbus einerseits, sowie die Datenanschlüsse des programmierbaren Parallelschnittstellenbausteins andererseits beim Schreib- bzw. Lesevorgang zu entlasten. Seine Funktion ist im Zusammenhang mit den Baugruppen 8-K-RAM/EPROM Speicherkarten bereits beschrieben und wird als bekannt vorausgesetzt.

Der Baustein wird durch das Adreßvergleichersignal ausgewählt. Die Übertragungsrichtung für die Daten -entweder vom Datenbus zum Schnittstellenbaustein, oder umgekehrt- wird mittels des Steuersignals IOW an dem Richtungssteuerungsanschluß DIR bestimmt.

CON1:

Sign	Textoc
STR	1 11
GND	2 10
SDD	3 12
SD1	4 9
5	13
SC	6 8
7	14
8	15
9	16
10	17
11	18
12	19
13	20
14	1
15	2
16	3
17	4
18	5
19	6
20	7

CON2:

IC1	IC2	IC3	IC4	IC5
8255	74LS00	95	245	NE555

CON3:

STR	SDD	SD1	SC	AA5	AA4	AA3	AA2	AA1	P/V	VCC
1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33
34	35	36	37	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63	64	65	66
67	68	69	70	71	72	73	74	75	76	77
78	79	80	81	82	83	84	85	86	87	88
89	90	91	92	93	94	95	96	97	98	99
100	101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120	121
122	123	124	125	126	127	128	129	130	131	132
133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154
155	156	157	158	159	160	161	162	163	164	165
166	167	168	169	170	171	172	173	174	175	176
177	178	179	180	181	182	183	184	185	186	187
188	189	190	191	192	193	194	195	196	197	198
199	200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219	220
221	222	223	224	225	226	227	228	229	230	231
232	233	234	235	236	237	238	239	240	241	242
243	244	245	246	247	248	249	250	251	252	253
254	255	256	257	258	259	260	261	262	263	264
265	266	267	268	269	270	271	272	273	274	275
276	277	278	279	280	281	282	283	284	285	286
287	288	289	290	291	292	293	294	295	296	297
298	299	300	301	302	303	304	305	306	307	308
309	310	311	312	313	314	315	316	317	318	319
320	321	322	323	324	325	326	327	328	329	330
331	332	333	334	335	336	337	338	339	340	341
342	343	344	345	346	347	348	349	350	351	352
353	354	355	356	357	358	359	360	361	362	363
364	365	366	367	368	369	370	371	372	373	374
375	376	377	378	379	380	381	382	383	384	385
386	387	388	389	390	391	392	393	394	395	396
397	398	399	400	401	402	403	404	405	406	407
408	409	410	411	412	413	414	415	416	417	418
419	420	421	422	423	424	425	426	427	428	429
430	431	43								

**Bild 1 : Gesamtschaltung des GAL-Programmierinterface**

### 2.2.0 Der elektronische Schalter für die Versorgungsspannung

Bild 2 zeigt den Schaltungsausschnitt, der die Versorgungsspannung von 5V zum Anschlußstift 20 des Aufnahmesockels für den zu programmierenden GAL-Baustein schaltet. Die so geschaltete Versorgungsspannung wird auch über einen Pull-up-Widerstand an den Anschlußstift 12 des Sockels gelegt.

Über den Portanschluß PC5 wird die Basis von Transistor V2 angesteuert. Führt PC5 einen L-Pegel, so sperrt der Transistor. Als Folge davon fließt kein Kollektorstrom durch die Teilerwiderstände R7 und R8 und der Transistor V1 wird ebenfalls sperren. An seinem Kollektoranschluß stellt sich somit ein hochohmiger Zustand ein.

Wird später das GAL-Programmierprogramm gestartet, so wird nach der Initialisierung des Schnittstellenbausteins an dem Portanschluß PC5 ein H-Pegel ausgegeben und damit der Transistor V2 über den Basisspannungsteiler R5 und R6 leitend geschaltet. Bedingt durch den jetzt fließenden Kollektorstrom von V2, wird die Basis von V1 angesteuert. V2 schaltet ebenfalls durch und damit ist die Versorgungsspannung an den o.g. Anschlüssen vorhanden. Dieser Zustand wird auch durch die Leuchtdiode V3 angezeigt.

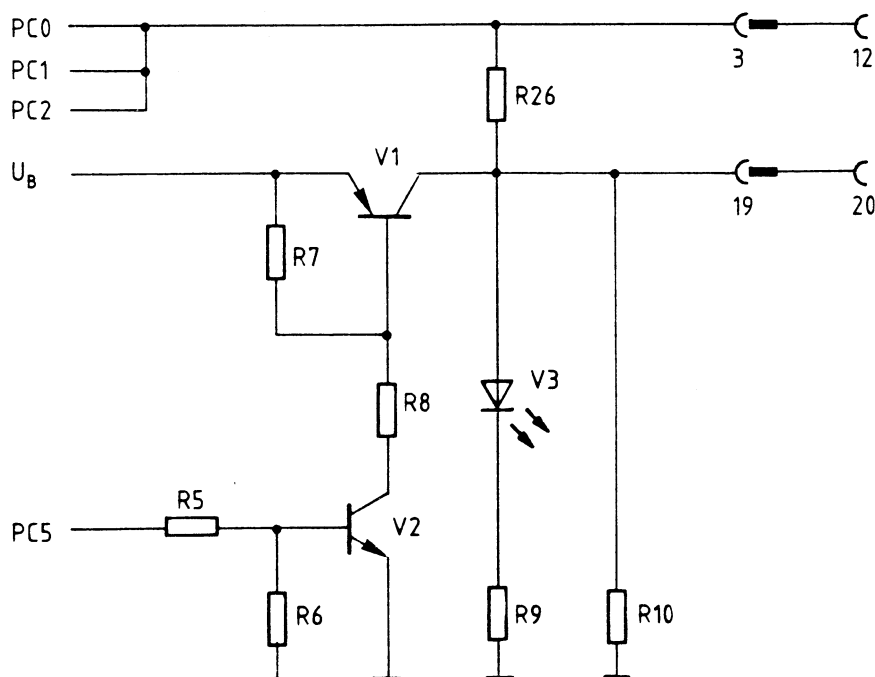


Bild 2: Elektronisches Schalten der Versorgungsspannung

### 2.3.0 Erzeugung der Editierspannung

Zur Gewinnung der beim Programmieren des GAL-Bausteins erforderlichen Editierspannung von 16,5V dient der nachfolgend beschriebene Schaltungsteil. Er besteht im wesentlichen aus drei Funktionsblöcken:

- Einem Rechteckgenerator
- einer Spannungsvervielfachungsschaltung
- einem elektronischen Schalter

#### 2.3.1 Der Rechteckgenerator

Der Rechteckgenerator wird mit dem Timer-Baustein 555 und der im Bild 3 dargestellten Beschaltung aufgebaut. Die Innenschaltung des Timers zeigt, daß er aus zwei Komparatoren aufgebaut ist, wobei von Komparator N1 der invertierende Eingang auf zwei Drittel und der nichtinvertierende Eingang von Komparator N2 auf ein Drittel der Versorgungsspannung intern vorgespannt sind. Die Ausgänge der Komparatoren wirken auf ein SR-Flipflop, dessen Ausgangszustand nach außen geführt ist. Am Q-Nicht-Ausgang des Flipflops ist die Basis des Entladetransistors angeschlossen, dessen Open-Collektor-Anschluß ebenfalls außen zur Verfügung steht.

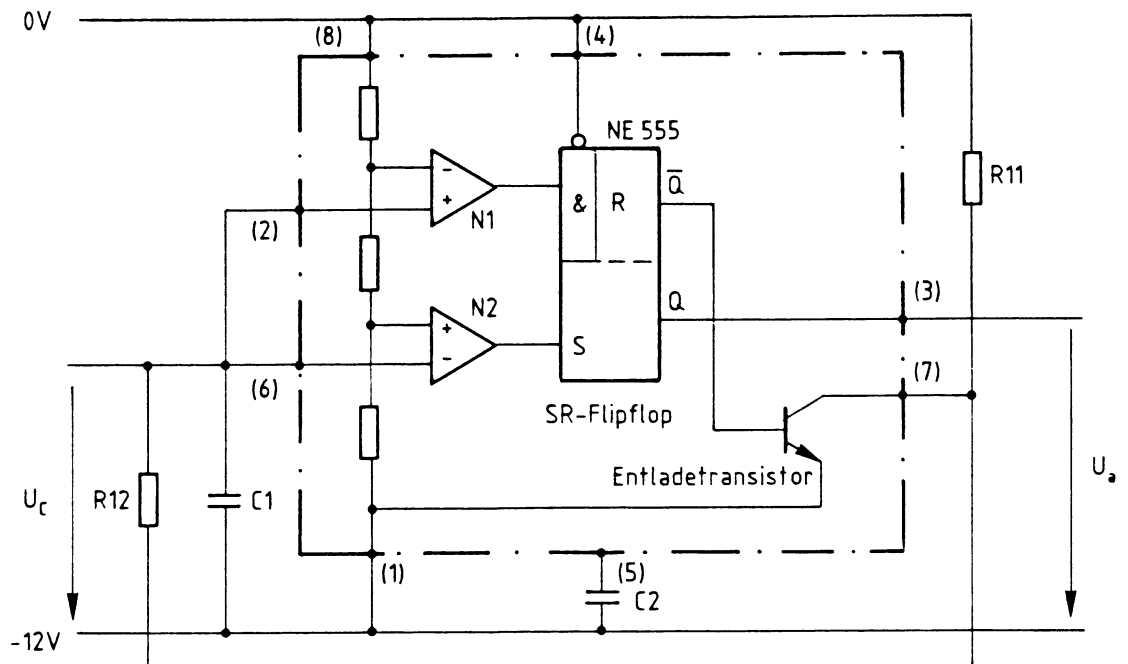


Bild 3: Innenschaltung und äußere Beschaltung des Timerbausteins 555



GAL-Programmierinterface

---

Durch die äußere Beschaltung mit den Widerständen R11 und R12 sowie dem Kondensator C1 wird die Gesamtschaltung zum Rechteckgenerator. Die Versorgungsspannung für diesen Schaltungsteil wird aus der vorhandenen negativen Spannung von -12V als niedrigem und 0V als hohem Versorgungspotential gewonnen.

Für die folgende Betrachtung gilt, daß der statische Rücksetzeingang (4) mit hohem Potential verbunden und damit wirkungslos ist. Geht man von dem Anfangszustand aus, daß das Flipflop zurückgesetzt sei, so ist der Entladetransistor im leitenden Zustand und damit der Ausgang (7) niederohmig nach dem niedrigen Versorgungspotential geschaltet. Hierdurch stellt sich an den beiden Komparatoren folgender Zustand ein: Der Ausgang von Komparator N1 wird einen L-Pegel annehmen, weil an seinem invertierenden Eingang ein höheres Potential anliegt, als an seinem nichtinvertierenden Eingang und bei Komparator N2 steht -umgekehrt- ein H-Pegel an. An den Flipflopeingängen ergibt sich daraus die Signalkombination für das Setzen. Damit wird der Ausgang (3) einen H-Pegel führen. Die Basis des Entladetransistors wird nicht mehr angesteuert, der Transistor sperrt.

Die Folge davon ist, daß ein Ladestrom über die Widerstände R11 und R12 in den Kondensator C1 fließen kann. Der Kondensator lädt sich mit der Zeitkonstanten  $(R11 + R12) * C1$  auf. Übersteigt die Spannung am Kondensator den Wert von  $1/3$  der Versorgungsspannung, so wird der Komparator N2 seinen Zustand wechseln und nun L-Pegel führen. Das Flipflop geht in den Speicherzustand, d.h. es bleibt gesetzt. Der Ladestrom fließt weiter und die Spannung am Kondensator steigt weiter an. Nachdem dann die Kondensatorspannung  $2/3$  der Versorgungsspannung überschritten hat, schaltet Komparator N1 ebenfalls um. Er führt jetzt einen H-Pegel, der bewirkt, daß das Flipflop zurückgesetzt wird. Ausgang (3) führt nun L-Pegel und der Entladetransistor wird durchgeschaltet. Dadurch wird der Kondensator C1 über den Widerstand R12 und die Kollektor-Emitterstrecke des Entladetransistors entladen. Die hierbei wirksame Zeitkonstante ist  $R12 * C1$ . Die Entladung des Kondensators wird solange fortgesetzt, bis die Kondensatorspannung wieder  $1/3$  der Versorgungsspannung unterschreitet.

GAL-Programmierschnittstelle

---

Das Ausgangssignal am Anschlußstift (3) des Timerbausteins bildet einen rechteckförmigen Spannungsverlauf. Die Pulszeit für den H-Pegel läßt sich aus dem Produkt

$$T_i = 0,66 * (R_{11} + R_{12}) * C_1$$
$$T_i = 0,66 * (33k + 3k3) * 22nF = \underline{\underline{0,527ms}}$$

berechnen und die Pausenzeit aus:

$$T_p = 0,66 * R_{12} * C_1$$
$$T_p = 0,66 * 33k * 22nF = \underline{\underline{0,479ms}}$$

Hierdurch läßt sich auch die Periodendauer des Rechtecksignals, entweder aus der Summe von  $T_i + T_p$ , oder aus der Gleichung

$$T = 0,66 * (R_{11} + 2 * R_{12}) * C_1$$
$$T = 0,66 * (3k3 + 66k) * 22nF = \underline{\underline{1,006ms}}$$

bestimmen und damit ergibt sich auch die Frequenz der Rechteckspannung als Kehrwert der Periodendauer  $T$  zu ca. 1kHz. Siehe hierzu auch Bild 4.

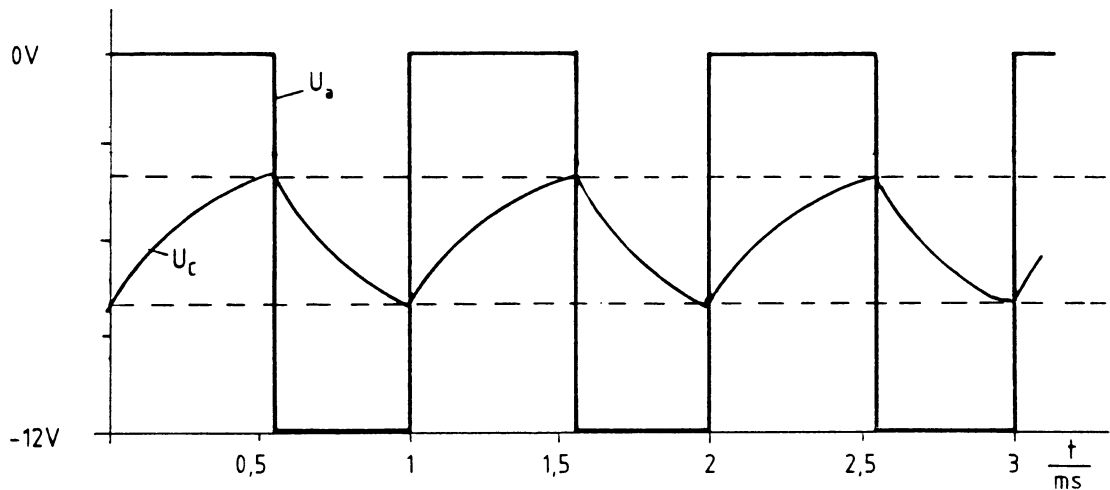


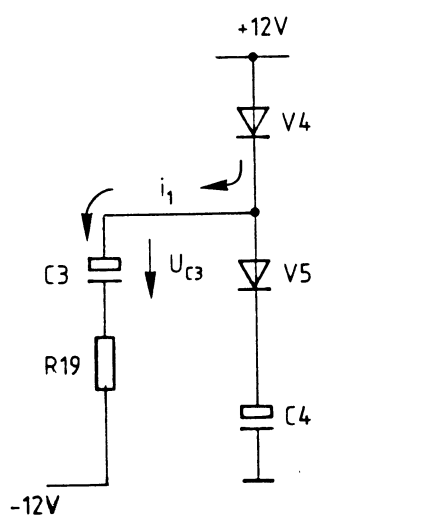
Bild 4 : Impulsverlauf am Timerausgang (3), Leerlauf

### 2.3.2 Die Spannungsvervielfachung

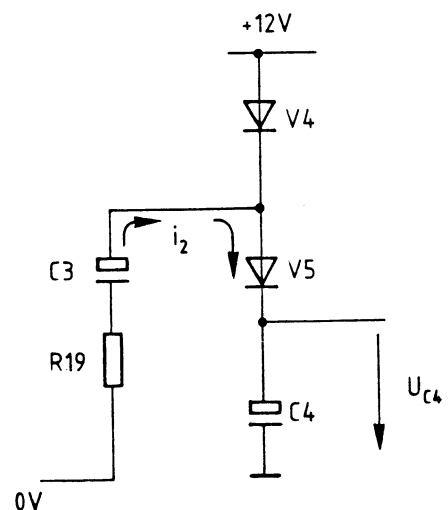
Die Erzeugung einer höheren Spannung, wie sie später für den Programmiervorgang der GAL-Bausteine benötigt wird, erfolgt im wesentlichen durch die beiden Kondensatoren C3 und C4, sowie den Dioden V4 und V5. Die Spannungsquelle liefert eine Spannung von 12V, die über die Diode V4 den Kondensator C3 auf diese Spannung auflädt. Ebenso wird der Kondensator C4 über die weitere Diode V5 aufgeladen.

Der Kondensator C3 wird zudem über den Widerstand R19 durch den Rechteckgenerator angesteuert. Der Rechteckgenerator liefert, wie bereits betrachtet, eine Rechteckimpulsfolge, deren beide Pegel 0V bzw. -12V betragen. Während der Ansteuerung mit -12V fließt also ein zusätzlicher Ladestrom über die Diode V4 in den Kondensator C3 und lädt ihn auf nahezu 24V auf. Wechselt alsdann der Pegel des Generators auf 0V, so kann der Kondensator sich nur über die Diode V5 zum Kondensator C4 entladen. Dieser Kondensator wird also auf die Spannung von ca. 24V aufgeladen. An ihm kann sie auch später abgenommen werden.

Bild 5a zeigt die Stromrichtung beim Aufladevorgang für Kondensator C3 während der Ansteuerung mit -12V über R19. Bild 5b zeigt die Stromrichtung bei der Entladung von C3, die die Aufladung von C4 zur Folge hat.



$i_1$  = Ladestrom für C3  
C3 wird auf  $U_{C3} \approx 24V$  aufgeladen



$i_2$  = Entladestrom von C3  
C4 wird auf  $U_{C4} \approx 24V$  aufgeladen

Bild 5a : Aufladung von C3

Bild 5b : Entladung von C3

### 2.3.3 Elektronischer Schalter und Stabilisierung für die Editierspannung.

Die durch die Spannungsvervielfachung gewonnene Gleichspannung von ca. 24V am Kondensator C4 wird mittels des Längstransistors V6 zum entsprechenden Anschluß des GAL-Bausteins durchgeschaltet. Transistor V6 kann diese Spannung nur dann schalten, wenn seine Basis angesteuert wird. Hierzu dient der Transistor V7, der von den Portanschlüssen PC3 und PC4 des Schnittstellenbausteins angesteuert wird. Bild 6 zeigt diesen Teil der Schaltung. Liefern die Portanschlüsse einen L-Pegel, so wird V7 und damit auch V6 sperren - die Editierspannung ist abgeschaltet. Bei H-Pegel an den Portanschlüssen wird über den Basis-Spannungsteiler R13, R14 der Transistor V7 in den leitenden Zustand gebracht und damit auch V6 durchgesteuert. Durch den Widerstand R18 fließt nun ein Strom über die Emitter-Kollektorstrecke des Transistors V6. Die Kondensatorspannung von nahezu 24V wird durch den Widerstand R18 und R17 geteilt und zudem durch die beiden in Reihe geschalteten Zenerdioden V8 und V9 auf 16,8V stabilisiert.

Diese so gewonnene Gleichspannung steht, gesteuert durch die Portanschlüsse des Schnittstellenbausteins am Anschluß (2) des Stecksockels für den GAL-Baustein zur Verfügung.

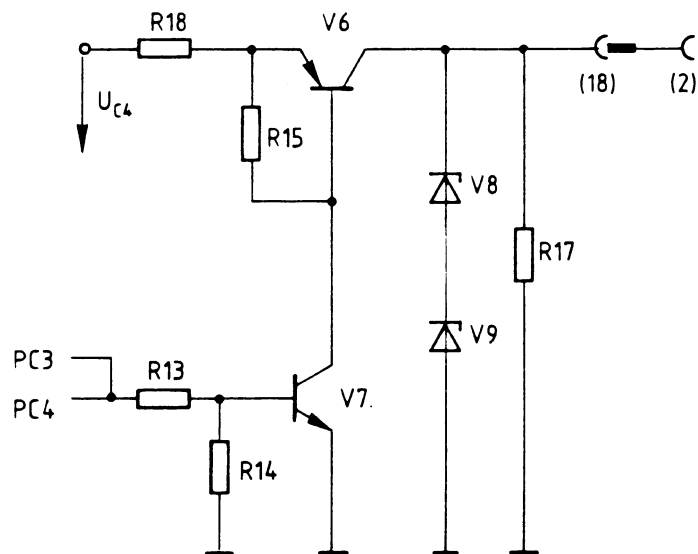
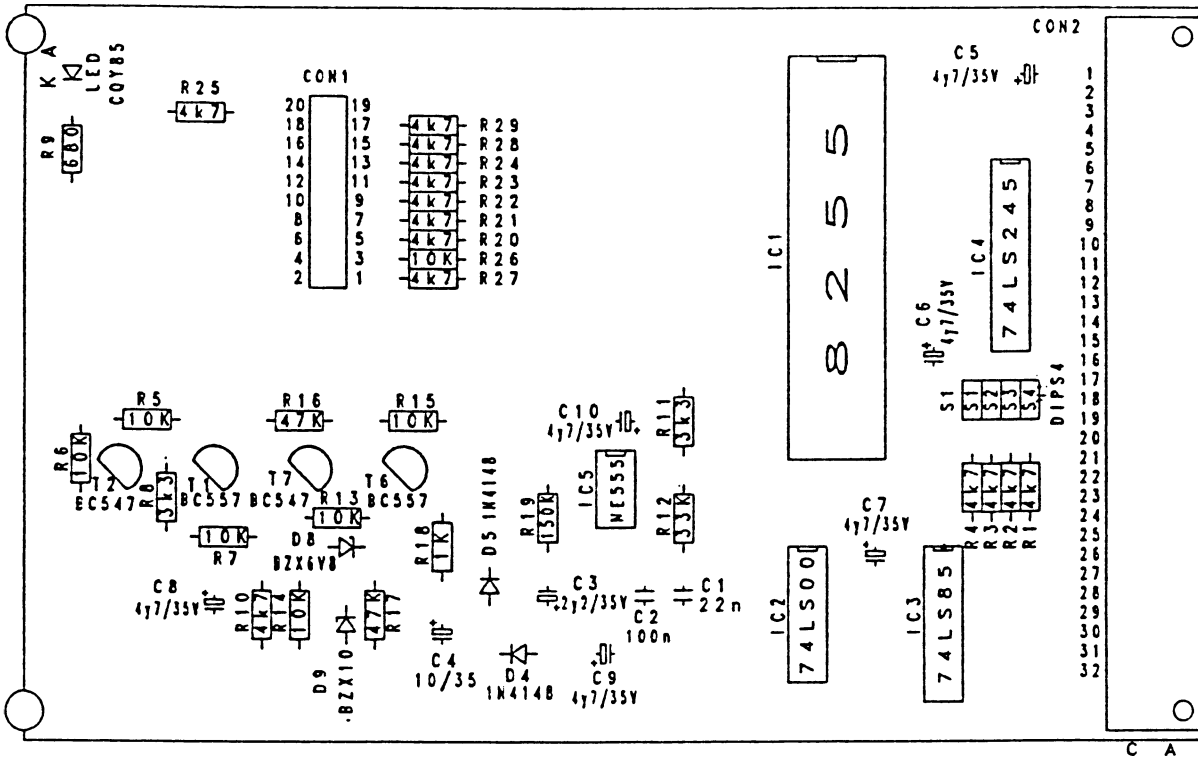


Bild 6 : Elektronischer Schalter und Stabilisierung für die Editierspannung

GAL-Programmierschnittstelle



## GAL-Programmierinterface

Stckz.	Benennung / Daten	Bemerkung
1	Leiterplatte, ca. 110x170 mm Mat.: Epoxid-Glashartgewebe (Hgw 2372)	doppelseitig Cu- kaschiert (35 um) Fotolackbeschichtet
1	Leiterplatte, ca. 25x80 mm Mat.: wie vor	wie vor
je 1	Filmvorlage Bfz/MFA 4.14 L und 4.14 B zum Belichten	je nach Ätzverfahren Pos.- oder Neg.-Film
je 1	Filmvorlage BFZ/MFA 4.14a L und 4.14a B zum Belichten	wie vor
1	Frontplatte, Teilung L-C 05, Alu, 2 mm dick, 25,1 mm br.	z.B. Intermas Nr. 409-017 665
1	Griff kpl. mit Abdeckung T03	z.B. Intermas Nr. 409-017 927
1	Frontverbinder 1,6 FEE	z.B. Intermas Nr. 409-024 830
1	Messerleiste 64pol., DIN 41612	z.B. Erni STV-P- 364 a/c
3	Zylinderschraube M 2,5 x 8	DIN 84
2	Zylinderschraube M 2,5 x 10	DIN 84
1	Zylinderschraube M 2,5 x 12	DIN 84
2	Zylinderschraube mit Schaft BM 2,5 x 10/5	DIN 84
2	Senkschraube mit Kreuzschlitz M 2,5 x 8	DIN
5	Federscheibe A 2,7	DIN 137
1	Federring B 2,5	DIN 127
8	Sechskantmutter M 2,5	DIN 439
2	Schraubensicherung, Kunststoff	z.B. Intermas Nr. 409-026 748

## GAL-Programmierinterface

Stckz.	Benennung / Daten	Bemerkung
1	Kondensator 22 nF	Keramik
1	Kondensator 100 nF	Keramik
1	Tantal-Elko 2,2 uF / 35 V	Tropfenform
1	Tantal-Elko 10 uF / 35 V	Tropfenform
6	Tantal-Elko 4,7 uF / 35 V	Tropfenform
1	Widerstand 680	0,25 W / 5% Toleranz
1	Widerstand 1 k	0,25 W / 5% Toleranz
2	Widerstand 3,3 k	0,25 W / 5% Toleranz
14	Widerstand 4,7 k	0,25 W / 5% Toleranz
7	Widerstand 10 k	0,25 W / 5% Toleranz
1	Widerstand 33 k	0,25 W / 5% Toleranz
2	Widerstand 47 k	0,25 W / 5% Toleranz
1	Widerstand 150 k	0,25 W / 5% Toleranz
1	Leuchtdiode rot, 3 mm Durchm.	
1	Befestigungshülse sw für LED	
1	Befestigungsring sw für LED	
2	Diode 1 N 4148	
1	Zenerdiode ZPD 6 V 8	
1	Zenerdiode ZPD 10 V	
2	NPN-Kleinsignaltransistor	BC 547
2	PNP-Kleinsignaltransistor	BC 557
1	IC 8255 Parallelschnittstellenbaustein	

## GAL-Programmierinterface

Stckz.	Benennung / Daten	Bemerkung
1	IC 74LS00, 4 NAND-Gatter	
1	IC 74LS85, 4-Bit-Vergleicher	
1	IC 74LS245, 8-Bit-Bustreiber	Tri-State
1	IC NE 555, Timerbaustein	
1	IC-Fassung 8-polig	
1	IC-Fassung 14-polig	
1	IC-Fassung 16-polig	
1	IC-Fassung 20-polig	
1	IC-Fassung 40-polig	
1	Miniaturschiebeschalter 4pol.	DIL
1	Steckverbinder 20pol. 2reihig	STV20
1	20pol. Leiterplattenstecker	z.B. KK20 Bürklin
1	20pol. Buchsenstecker	z.B. Flakafix Bürklin
1	Sockel 20pol. mit Klemmfassung	z.B. Textool Nr. 220 3342
n.B.	Flachbandlitze 20adrig	ca. 50 mm
n.B.	Löt draht	
n.B.	Lötlack	
n.B.	Reinigungsmittel	
n.B.	Klarlacksspray	
n.B.	Beschriftungsmaterial	



## GAL-Programmierinterface

Zum Prüfen der Baugruppe GAL-Programmierinterface benötigen Sie zusätzlich:

Stckz.	Benennung / Daten	Bemerkung
1	Baugruppenträger mit Busverdrahtung BFZ/MFA 0.1	
1	Bus-Abschluß BFZ/MFA 0.2	
1	Trafo-Einschub BFZ/MFA 1.1	
1	Spannungsregler BFZ/MFA 1.2	
1	Prozessor 8085 BFZ/MFA 2.1	
* 1	8-K-EPROM BFZ/MFA 3.1	bestückt mit MAT 85
* 1	8-K-RAM BFZ/MFA 3.1	bestückt mit mindestens 2-K-RAM ab Adr. F800
1	Videointerface	
1	Tastatur	
1	Datensichtgerät	
	Meß- und Prüfmittel:	
1	Spannungsmesser	
1	Widerstand 10 kOhm 0,25 W	
ggf.	Logiktester	
1	Floppy-Interface mit Laufwerk (3,5 oder 5,25 Zoll)	

\* 1 ersatzweise 64K-RAM-Karte mit MAT 32K

**Name:**

---

**GAL-Programmierinterface****Datum:**

---

#### **4.0 Überprüfung der Baugruppe GAL-Programmierinterface**

1. Stellen Sie auf der Baugruppe die Basisadresse "D" ein, das bedeutet: S4, S3 und S1 in Stellung OFF und S2 in Stellung ON schalten.
2. Stecken Sie die Baugruppe GAL-Programmierinterface in das Mikrocomputersystem ein und schalten Sie die Betriebsspannung ein.
3. Mit Hilfe des "OUT"-Kommandos wird das Interface zunächst initialisiert. Rufen Sie deshalb das "OUT"-Kommando auf und wählen Sie die Portadresse D3H aus. D3H ist die Adresse des Steuerwortregisters. Durch Ausgabe des Datums 81H in das Steuerwortregister wird der Parallelschnittstellenbaustein initialisiert. Hiermit werden die Portanschlüsse PC0...PC3 zu Eingabekanälen und alle anderen Portanschlüsse zu Ausgabekanälen.
4. Geben Sie mittels des OUT-Kommandos nacheinander die folgenden Daten aus und überprüfen Sie an den Portanschlüssen die jeweiligen Spannungen bzw. Pegel:

Name: \_\_\_\_\_

GAL-Programmierinterface

Datum: \_\_\_\_\_

**4.1 Überprüfung der benötigten Anschlüsse von PORT A****Die Daten müssen an die Portadresse D0 übergeben werden.**

Tragen Sie in die untenstehende Tabelle die gemessenen IST-Pegel unter der Spalte I ein, und vergleichen Sie sie mit den vorgegebenen SOLL-Pegeln. Messen Sie zunächst, ob der jeweilig richtige Wert an den Anschlüssen des Aufnahmesockels für den GAL-Baustein vorhanden ist und gehen Sie erst im Fehlerfall bis zu den Portanschlüssen des Schnittstellenbausteins zurück. Es werden lediglich die Anschlüsse PA0 bis PA5 benötigt, sie dienen beim Programmierungsvorgang zur Erzeugung der 64 internen Adressen des GAL-Bausteins an den Bausteinanschlüssen A0 bis A5.

Ausg.- wert	7 PA5		6 PA4		5 PA3		4 PA2		3 PA1		18 PA0		Sockel Port Pegel
	S	I	S	I	S	I	S	I	S	I	S	I	
15	L		H		L		H		L		H		
2A	H		L		H		L		H		L		

Name:

GAL-Programmierinterface

Datum:

#### 4.2 Überprüfung der benötigten Anschlüsse von Port B

Die Daten müssen an die Portadresse D1 übergeben werden.

Gehen Sie bei dieser Messung genau so vor, wie bei der vorhergehenden. Da nur die beiden Portanschlüsse PB0 und PB1 für den Betrieb als GAL-Programmierinterface benötigt werden, beschränkt sich der Funktionstest auch auf diese beiden Anschlüsse. PB0 dient zur Übergabe der Programmierdaten an den seriellen Dateneingang des GAL 16 V 8, SDI. Über PB1 wird der Schiebetakt beim Programmervorgang an den Anschluß SC des GAL geliefert.

Ausg.- wert	8 PB1	9 PB0	Sockel Port Pegel
	S I	S I	
01	L	H	
02	H	L	

Name: \_\_\_\_\_

GAL-Programmierinterface

Datum: \_\_\_\_\_

#### 4.3 Überprüfung der Funktionen, die durch das Port C des Schnittstellenbausteins gesteuert werden.

Hierzu ist folgende Vorbereitung zu treffen :

Schliessen Sie zwischen den Anschlüssen GND', Pin 10 und SDO, Pin 12 der Fassung einen 10-kOhm Widerstand an.

Die Daten müssen an die Portadresse D2 übergeben werden.

Zur ersten Überprüfung messen Sie die Pegel an den Portanschlüssen PC4 bis PC7 nachdem Sie jeweils die folgenden Daten ausgegeben haben :

Ausg.-wert	PC7		PC6		PC5		PC4	
	S	I	S	I	S	I	S	I
50	L		H		L		H	
A0	H		L		H		L	

##### 4.3.1 Schalten der Betriebsspannung VCC.

Die Betriebsspannung VCC von 5V wird über die Transistoren V1 und V2 zur Fassung geschaltet, wenn über den Portanschluß PC5 ein H-Pegel ausgegeben wird.

Überprüfen Sie diese Funktion, indem Sie das Datum 20H an das Port C, Adresse D2H, ausgeben.

Messen Sie das Vorhandensein der Betriebsspannung am Anschluß 20 der Fassung. Der Sollwert muß zwischen 4,75V und 5,25V liegen. Die Leuchtdiode V3 des Programmierinterface leuchtet. Am Anschluß 12 der Fassung muß, bedingt durch den 10 kOhm-Teilerwiderstand, die halbe Betriebsspannung gemessen werden.

**Name:**

---

**GAL-Programmierinterface****Datum:**

---

Entfernen Sie nun den Teilerwiderstand, da er für die weiteren Prüfschritte nicht mehr benötigt wird.

#### **4.3.2 Erzeugen und Schalten der Editierspannung.**

Die Editierspannung von ca. 16V wird durch die Schaltungskomponenten Rechteckgenerator, Spannungsvervielfacher und elektronischem Schalter erzeugt, wenn am Port PC4 ein H-Pegel eingestellt wird. Sie steht dann am Anschluß 2 der Fassung zur Verfügung.

Überprüfen Sie diese Funktion, indem Sie das Datum 10H an das Port C, Adresse D2H, ausgeben.

Messen Sie das Vorhandensein der Editierspannung am Anschluß 2 der Fassung. Der Sollwert liegt zwischen 16,25V und 16,75V.

#### **4.3.3 Funktion der Schreib-/Leseumschaltung.**

Die Schreib-/ Leseumschaltung wird über den Portanschluß PC7 angesteuert. Geben Sie deshalb das Datum 80H an das Port C, Adresse D2H aus.

Prüfen Sie den Pegel des Signals am Anschluß 19 der Fassung, er muß jetzt H-Pegel führen.

#### **4.3.4 Erzeugung des Strobe-Signals.**

Das Strobe-Signal wird über den Portanschluß PC6 geschaltet. Geben Sie das Datum 40H an das Port C, Adresse D2H aus.

Prüfen Sie den Pegel des Signals am Anschluß 11 der Fassung, er muß jetzt H-Pegel führen.

**Name:**

---

**GAL-Programmierinterface****Datum:**

---

**4.3.5 Überprüfung der Eingangsanschlüsse von Port C.**

Geben Sie nun nochmals das Datum 20H an das Port C aus und lesen Sie anschließend mit dem IN-Kommando den Zustand des Port C ein. Er muß 27H sein, da an den Portanschlüssen PC0, PC1 und PC2 über den Widerstand R26 die Betriebsspannung anliegt. Die oberen 4 Bit des Port C sind ja noch als Ausgabekanäle programmiert und daher wird hier auch der eben ausgegebene Wert aus dem Register des Schnittstellenbausteins gelesen.

**Name:**

---

**GAL-Programmierinterface****Datum:**

---

## **5.0 Einsatz der Programmiersoftware**

### **5.1 Hardwarevoraussetzungen**

Die Software liegt in Form einer Diskette (3,5 oder 5,25") vor und muß ab Adresse C000 H betrieben werden.

Zur Anwendung der Software benötigt man 4-K-Byte RAM ab Adresse D000 H. Günstig ist hierbei die Verwendung der 64-K-Speicherkarte mit dem MAT 32K.

### **5.2 Programmaufruf**

Die GAL-Programmiersoftware wird mittels des GO-Kommandos gestartet. Hierzu rufen Sie das GO-Kommando auf und geben die Startadresse der Programmiersoftware, C000 H, ein.

Hierdurch wird folgende Meldung auf dem Bildschirm erzeugt:

**GAL 16V8, 16V8A V2.3**

**EDIT  
PROG  
DELETE  
HELP  
TEST  
READ  
LPRINT  
VERIFY  
CHANGE  
NEW  
QUIT**

**KMD\*>**

Dieses ist die Liste der unter dem GAL-Programmierer möglichen Kommandos. Das Prompt ist an dem zusätzlichen Zeichen "\*" von den Kommandoaufrufen KMD> (MAT 85) und KMD+ (MAT 85+) zu unterscheiden.





**Name:**  
\_\_\_\_\_**GAL-Programmierinterface****Datum:**  
\_\_\_\_\_

Die Kennzeichnung "X" bedeutet hierbei, daß eine Verbindung programmiert werden soll. Ein "-" bedeutet, daß hier später keine Verbindung programmiert wird. Die Software schlägt also die Programmierung aller möglichen Verknüpfungen vor. Der Vorschlag für jede Verbindung kann durch Überschreiben mit dem Zeichen "-" geändert werden und auch, durch erneute Eingabe von "X" wieder zurückgeändert werden. Hierzu muß die Schreibmarke, das ist der blinkende Unterstrich, mit den Pfeiltasten auf die Stelle in der Tabelle bewegt werden, die geändert werden soll.

**Steuerfunktionen innerhalb des Editiermodus:**

Betätigung der Taste(n) :

**Pfeiltasten**                   bewegen die Schreibmarke**Buchstabe "P"**               führt zu den Produktterm-Verbindungen (PTD = Produktterm disable). Ein "X" an dieser Stelle sperrt die UND-Verknüpfung, unabhängig davon, wie deren übrige Eingänge programmiert sind.**Leerzeichen**               führt zum nächsten AND-Array, bzw. nach AND-Array 7 zum Ausgangs-Steuerwort.**Carriage Return**       führt zurück zur KMD\*>-Kommandoebene**6.1.2 Editieren des Ausgangs-Steuerwortes**

Mit dem Ausgangs-Steuerwort wird die Makrozelle des GAL-Bausteins beeinflusst. Die Bedeutung der einzelnen Bits in dem Steuerwort wird hier nur kurz aufgeführt.

Name:  

---

GAL-Programmierinterface

Datum:  

---

Um das Steuerwort zu editieren geben Sie , nach dem Aufruf des Editierkommandos, den Buchstaben A gefolgt von (CR) ein.

KMD\*> EDIT? A(CR)      Hierauf erscheint die folgende Meldung auf dem Bildschirm :

AUSGANGS-STEUERWORT

```
SYN AC0 0  AC1  7 0  XOR  7
      0   0  0000 0000 0000 0000
```

Im Rechnervorschlag sind alle Bits des Ausgangs-Steuerwortes auf 0 eingestellt. Änderungen können durch das Überschreiben mit dem Wert 1, Rückänderungen durch Überschreiben mit 0 durchgeführt werden. Die einzelnen Bits können mit den Pfeiltasten für rechts, bzw. links ausgewählt werden.

Kurzbeschreibung der Bedeutung der Bits :

SYN            =            Synchronbit. 0 bedeutet, der Baustein arbeitet mit einem Takt, 1 bedeutet, der Baustein arbeitet rein kombinatorisch.

AC0 und AC1 =            Architecture Control Bit. Die Kombination dieser Bits beeinflußt die Makrozelle des GAL-Bausteins so, daß z.B. Ausgänge zu Eingängen geschaltet werden, bzw. festgelegt wird, womit die Ausgangsfreigabe (Output Enable) erfolgen soll, woher ggf. Rückführungen kommen sollen. AC0 existiert nur einmal für den gesamten Baustein, während AC1 für jede der 8 Ausgangsmakrozellen programmierbar ist.

Name:  
\_\_\_\_\_

GAL-Programmierinterface

Datum:  
\_\_\_\_\_

XOR            =            Exclusive OR. Hiermit können die Ausgangszustände der ODER-Verknüpfungen invertiert werden. Eine 1 an der entsprechenden Position invertiert das Signal, eine 0 invertiert das Signal nicht.

**Steuerfunktionen innerhalb des Editiermodus:**

**Pfeiltasten**                      Schreibmarke nach rechts bzw. links

**Leerzeichen**                      führt zur Bausteinauswahl

**Carriage Return (CR)** führt zurück zur KMD\*>-Kommandoebene

**6.1.3 Editieren der Bausteinbezeichnung**

Als Bausteinbezeichnung kann ein Name oder eine Bezeichnung der Bausteinfunktion sowie eine Nummer dienen. Insgesamt darf eine solche Bezeichnung maximal 8 Zeichen lang sein.

Um eine Bausteinbezeichnung zu editieren geben Sie, nach dem Aufruf des Editierkommandos, den Buchstaben B gefolgt von (CR) ein.

KMD\*> EDIT? B(CR) Hierauf erhalten Sie die Meldung :

BAUSTEIN-BEZEICHNUNG  
.....

Jeder dieser 8 Punkte kann durch einen Buchstaben, eine Ziffer oder ein Leerzeichen überschrieben werden. Die Bausteinbezeichnung wird beim Programmiervorgang des GAL-Bausteins mit programmiert.

**Steuerfunktionen innerhalb des Editiermodus:**

**Pfeiltasten**                      Schreibmarke nach rechts bzw. links

**Carriage Return (CR)** führt zurück zur KMD\*>-Kommandoebene

Name:  
\_\_\_\_\_

GAL-Programmierinterface

Datum:  
\_\_\_\_\_

#### 6.1.4 Editieren des Securitybit

Das Securitybit dient zur Programmierung eines Leseschutzes. Ist dieses Bit im Baustein gesetzt, so kann sein Inhalt nicht gelesen werden. Alle Funktionen, die einen solchen Lesevorgang erfordern (READ, VERIFY) sind dann nicht mehr durchführbar.

Um das Securitybit zu editieren geben Sie, nach dem Aufruf des Editierkommandos, den Buchstaben S gefolgt von (CR) ein.

KMD\*> EDIT? S(CR) Sie erhalten die Meldung :

KMD\*> EDIT? SECURITYBIT - \_

Der Vorschlag "-" bedeutet, daß kein Leseschutz programmiert wird. Ein "X" bedeutet Leseschutz wird programmiert.

#### Steuerfunktionen innerhalb des Editiermodus:

**Delete**                                löscht die Eingabe vor Abschluß

**Carriage Return (CR)** führt zurück zur KMD\*>-Kommandoebene

Name:  

---

GAL-Programmierinterface

Datum:  

---

## 6.2 Das Programmierkommando PROG

Mit diesem Kommando wird der Programmiervorgang ausgelöst. Der durch das EDIT-Kommando in den RAM-Bereich für die "Fuse-Map" eingetragene Inhalt für die UND-Verknüpfungsfelder, das Ausgangs-Steuerwort, die Bausteinbezeichnung und das Securitybit wird in den GAL-Baustein programmiert.

Ein GAL-Baustein des Typs 16 V 8 oder 16 V 8 A muß hierzu in die Fassung des GAL-Programmierinterface gesteckt sein.

Aufruf des Kommandos :

KMD\*> P(CR)

Es erfolgt die Meldung :

KMD\*> PROG

CHANGE, PROG, VERIFY : bei fehlerfreier Programmierung.

Bei fehlerhaftem Programmierversuch werden Fehlermeldungen erzeugt.

## 6.3 Das Kommando DELETE

Dieses Kommando dient zum Löschen des Inhaltes eines programmierten GAL-Bausteins. Es werden alle programmierten Verbindungen, sowie Bausteinbezeichnung und Leseschutz zurückgesetzt.

Aufruf des Kommandos :

KMD\*> D(CR)

Es erfolgt die Meldung :

KMD\*> DELETE

Bei fehlerhaftem Versuch werden Fehlermeldungen erzeugt.

**Name:**

---

**GAL-Programmierinterface****Datum:**

---

#### **6.4 Das HELP-Kommando**

Mit Aufruf dieses Kommandos wird die Kommandoliste nochmals ausgedruckt.

Aufruf des Kommandos

KMD\*> H(CR) Es erfolgt die Meldung :

Edit

.

.

Quit

KMD\*>

#### **6.5 Das TEST-Kommando**

Das Test-Kommando dient dazu, daß die Baustein-Bezeichnung aus dem GAL ausgelesen und auf dem Bildschirm dargestellt wird.

Aufruf des Kommandos

KMD\*> T(CR) Es erfolgt die Meldung :

KMD\*> TEST

BAUSTEIN-BEZEICHNUNG

AMPEL 03

KMD\*> \_

Hierin ist die Bezeichnung AMPEL 03 ein Beispiel. Verfügt der Baustein über keine Bezeichnung, so wird diese Zeile als Leerzeile dargestellt.

Name:

---

GAL-Programmierinterface

Datum:

---

## 6.6 Das READ-Kommando

Mit diesem Kommando wird der Inhalt aus einem im GAL-Interface eingesteckten Baustein als Fuse-Map ins RAM eingelesen. Der eingelesene Inhalt kann jetzt z.B. bearbeitet oder auch in einen weiteren Baustein programmiert werden.

Das READ-Kommando kann nicht verwendet werden, wenn im Baustein das SECURITYBIT gesetzt ist. In diesem Fall können die Daten aus dem GAL-Baustein nicht gelesen werden !

Aufruf des Kommandos

KMD\*> R(CR) Es erfolgt die Meldung :

KMD\*> READ

KMD\*> \_

## 6.7 Das Kommando LPRINT

Mit diesem Kommando können alle Programmierdaten ausgedruckt werden.

Aufruf des Kommandos

KMD\*> L(CR) Es erfolgt die Meldung :

KMD\*> LPRINT? \_ An dieser Stelle kann, genau wie beim Editieren, ein Teil des AND-Array (0...7), die Baustein-Bezeichnung bzw. das Ausgangssteuerwort ausgewählt werden. Der jeweils ausgewählte Bereich wird auf Drucker oder Bildschirm ausgedruckt. Gibt man nichts weiteres an, sondern betätigt lediglich (CR), so wird der komplette Inhalt des RAM-Bereichs ausgedruckt also die Bausteinbezeichnung, alle acht AND-Arrays und das Ausgangs-Steuerwort.



Name: \_\_\_\_\_

GAL-Programmierinterface

Datum: \_\_\_\_\_

## 6.8 Das VERIFY-Kommando

Dieses Kommando dient zum Vergleich des RAM-Bereichinhaltes mit dem Inhalt eines im GAL-Programmierinterface eingesteckten GAL-Bausteins. Unterschiede werden hierbei gekennzeichnet.

Das VERIFY-Kommando kann nicht verwendet werden, wenn im Baustein das SECURITYBIT gesetzt ist. In diesem Fall können die Daten aus dem GAL-Baustein nicht gelesen werden und damit auch nicht verglichen werden !

Aufruf des Kommandos

KMD\*> V(CR)            Es erfolgt die Meldung :

KMD\*> VERIFY            bei fehlerfreiem Vergleich, d.h. es besteht an keiner Stelle ein Unterschied zwischen RAM- und Bausteininhalt.

### 6.8.1 Die Fehlermeldungen beim VERIFY-Kommando

Werden durch das Verify-Kommando Unterschiede erkannt, so werden sie wie folgt gemeldet :

#### 6.8.1.1 Unterschied in der Bausteinbezeichnung

Nach Aufruf des VERIFY-Kommandos erfolgt die Meldung :

KMD\*> VERIFY

BAUSTEIN-BEZEICHNUNG

AMPEL 03  
IC ....RE 2

RAM-Inhalt  
IC-Inhalt

Die Meldung zeigt erstens, daß in der Baustein-Bezeichnung Unterschiede gefunden wurden und zweitens, worin diese Unterschiede bestehen. Die Zeile, die mit "IC" beginnt, symbolisiert den Inhalt des GAL-Bausteins. In dem Beispiel sind bei den ersten vier Buchstaben der Baustein-Bezeichnung keine Unterschiede (...), aber bei den

Name:  
\_\_\_\_\_

GAL-Programmierinterface

Datum:  
\_\_\_\_\_

letzten vier sind im Baustein die Zeichen RE 2 statt L 03 (RAM) gefunden worden. Übereinstimmungen werden also durch Punkte und Unterschiede durch die entsprechenden Zeichen gekennzeichnet.

#### 6.8.1.2 Unterschied im Ausgangs-Steuerwort

Unterschiede im Ausgangs-Steuerwort werden genauso wie die in der Bausteinbezeichnung dargestellt. Eine solche Fehlermeldung kann z.B. wie folgt aussehen :

KMD\*> VERIFY

##### AUSGANGS-STEUERWORT

SYN	AC0	0	AC1	7	0	XOR	7	
1	0	0000	0000	1111	1111			RAM-Inhalt
IC	.	.	....	....	0...	....		IC-Inhalt

#### 6.8.1.3 Unterschied im AND-Array

Nach Aufruf des VERIFY-Kommandos erscheint zunächst die Fehlermeldung :

KMD\*> VERIFY

\*\*\* AND-ARRAY UNGLEICH \*\*\*

KMD\*> \_

Jetzt gilt es, die Unterschiede im AND-Array ausfindig zu machen. Hierzu muß das EDIT-Kommando aufgerufen werden und die einzelnen Arrays müssen nun auf Fehlermeldungen durchsucht werden. Die Unterschiede werden in den UND-Verknüpfungsfeldern durch die Kennzeichnungen "0" und "1" angezeigt.

**Name:**  
\_\_\_\_\_**GAL-Programmierinterface****Datum:**  
\_\_\_\_\_

Eine "0" besagt hierbei, daß im RAM eine Verbindung (X) eingetragen ist und im Baustein stattdessen keine Verbindung (-) programmiert ist.

Eine "1" signalisiert, im RAM ist keine Verbindung (-) eingetragen und im Baustein ist eine Verbindung (X) programmiert.

An den Positionen, an denen kein Unterschied besteht, sind die herkömmlichen Verbindungszeichen "-" und "X" eingetragen.

## **6.9 Das Kommando CHANGE**

Mit diesem Kommando werden die Fehlerkennzeichnungen, die zuvor erzeugt wurden wieder gegen die ursprünglichen RAM-Programmierdaten ausgetauscht. Das Kommando wird bei einem Programmiervorgang automatisch aufgerufen, um evtl. noch vorhandene Fehlermeldungen nicht irrtümlich mit zu übertragen. Ebenfalls werden die alten Fehlermeldungen beim erneuten Aufruf des Kommandos VERIFY durch das CHANGE-Kommando rückgängig gemacht.

Aufruf des Kommandos

KMD\*> C(CR)            Es erfolgt die Meldung :

KMD\*> CHANGE

KMD\*> \_

## **6.10 Das Kommando NEW**

Dieses Kommando dient dazu, die Programmierdaten in der Fuse-Map im RAM zurückzusetzen.

**Name:**  
\_\_\_\_\_**GAL-Programmierinterface****Datum:**  
\_\_\_\_\_

Aufruf des Kommandos

KMD\*> N(CR)            Es erfolgt die Meldung :

GAL 16V8, 16V8A V2.3

KMD\*> \_

### 6.11    Das Kommando QUIT

Durch Aufruf dieses Kommandos wird der GAL-Programmierer verlassen und zur Kommandoebene des Betriebssystems zurückgekehrt.

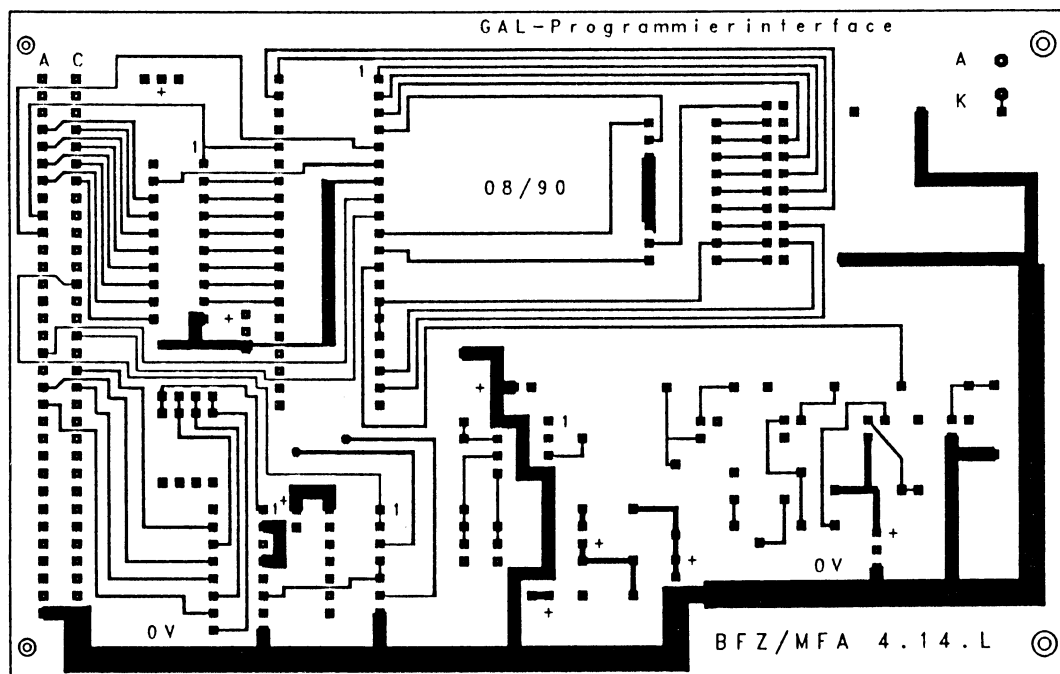
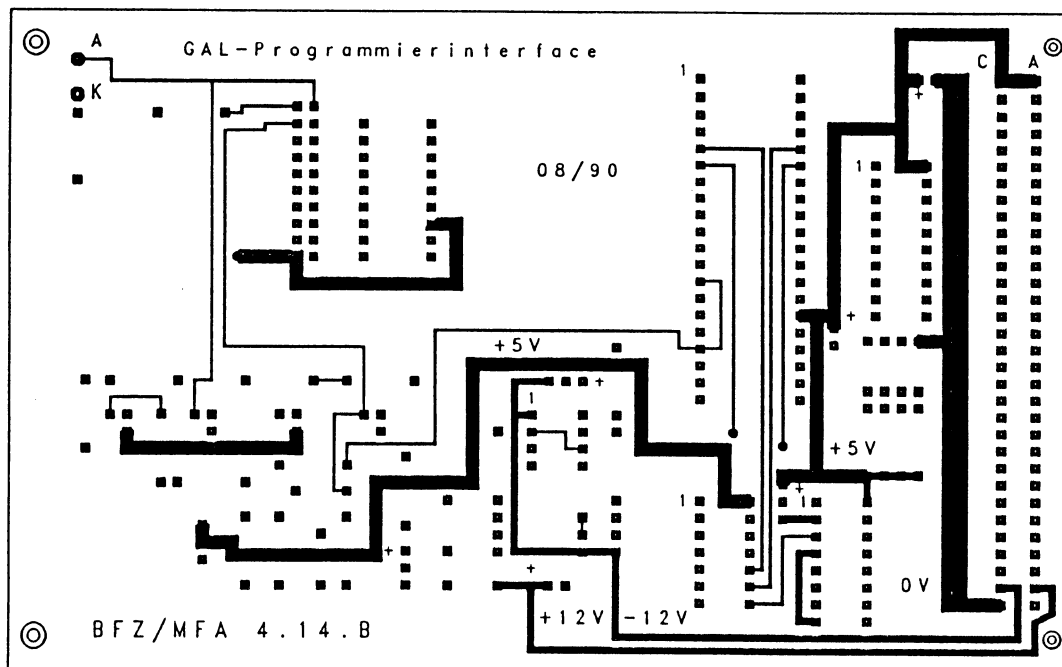
Aufruf des Kommandos

KMD\*> Q(CR)            Es erfolgt die Meldung :

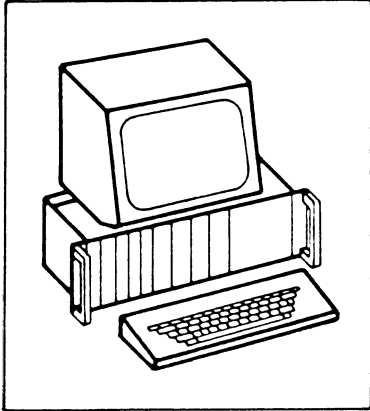
KMD\*> QUIT

KMD > \_

<p><b><u>Hinweis</u> :</b> Die Programmierdaten können mittels des SAVE und des LOAD-Kommandos des Betriebssystems gespeichert und zurückgelesen werden. Hierbei gilt als Startadresse D000 und als Stopadresse D89F</p>
--



# FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



Digitale  
Motor-Regelung





I n h a l t

## 1. Einleitung

1.1 Aufgabenstellung 3

1.2 Aufbau der Steuerungs- und Regelstrecke 3

## 2. Methodisch-didaktische Hinweise

2.1 Lernziele 5

2.2 Voraussetzungen 5

2.3 Bereitstellungsliste 6

## 3. Hinweise zur Lösung der Übung

3.1 Einfache Motorsteuerung mit dem Baustein L6203 7

3.2 Beispiellösung für Motor-Linkslauf 9

3.3 Verbesserte Steuerung mit digitaler  
Einstellung der Drehzahl 11

3.4 Beeinflussung der Motordrehzahl durch  
die Belastung des Generators 15

4. Aufbau und Wirkungsweise eines digitalen Zwei-  
punktreglers zur Drehzahlregelung eines DC-Motors

4.1 Grundlagen 19

4.2 Aufbau des Regelkreises 19

4.3 Unterschied zwischen analogen und digitalen  
Regeleinrichtungen 20



---

4.4	Ansteuerung des A/D-Wandlers durch die Generatorspannung	21
4.5	Beispiellösung für einen digitalen Zweipunktregler	23
4.6	Messungen an einem digitalen Zweipunktregler	27
4.7	Diskussion der Meßergebnisse	28
5.	Realisierung der Regeleinrichtung mit Hilfe des MFA- Einplatinen-Computers	
5.1	Vorbereitung des MFA-Systems zur Anpassung des Programms an den EPC	29
5.2	Anpassung des Programms an den EPC	30
5.3	Probelauf und Programmieren des EPROMs	31
6.	Anhang	
6.1	Überprüfung der Regelstrecke	34
6.2	Lösungsvorschläge für die Programmieraufgaben	35

## 1. Einleitung

### 1.1 Aufgabenstellung

Die vorliegende Übung "Digitale Motor-Regelung" ist in zwei Teile untergliedert:

- Digitale Steuerung eines Gleichstrommotors (im folgenden als DC-Motor bezeichnet)

In diesem Teil soll die Drehzahl eines 12-Volt-DC-Motors durch das MFA-Mikrocomputersystem (im folgenden kurz als MC-System bezeichnet) gesteuert werden. Die Drehzahl wird durch eine 8-Bit-Eingabe-Baugruppe indirekt vorgegeben.

- Digitale Regelung eines DC-Motors

Hier soll die Drehzahl des DC-Motors in Abhängigkeit von der Belastung der Motorwelle durch ein MC-System geregelt werden. Die Nenndrehzahl wird wiederum durch eine 8-Bit-Eingabe-Baugruppe indirekt vorgegeben.

Hinweis:

Falls die Regelstrecke noch nicht überprüft ist, befolgen Sie vor Durchführung der Übung die Anweisungen auf Seite 34.

### 1.2 Aufbau der Steuerungs- und Regelstrecke

Die Anlage besteht aus einer starr gekoppelten Motor-Generator-Kombination. Dabei handelt es sich um DC-Motoren, von denen einer als Motor und der andere als Generator betrieben wird.

Der Baustein L6203 ist ein integrierter Motorsteuerbaustein mit den TTL-kompatiblen Eingängen IN1, IN2 und EN. Diese Eingänge können direkt an eine 8-Bit-Ausgabe-Baugruppe des MC-Systems angeschlossen werden oder an eine 8-Bit-Ausgabe des MFA-Einplatinen-Computers (im folgenden kurz als EPC bezeichnet).

An den Anschluß Uext der Regelstrecke muß eine Gleichspannung von 12 Volt angeschlossen werden, die zur Versorgung des Steuerbausteins L6203 dient. Aus Gründen der Störsicherheit muß diese Spannung einem separaten Netzteil entnommen werden.

An den Eingang 5V muß eine Spannung von 5 Volt angeschlossen werden. Sie dient zur Einstellung von H- oder L-Pegeln über die Schalter S1 bis S8, wenn die Regelstrecke mit dem EPC betrieben wird.

Am Anschluß Ug liegt eine zur Drehzahl der Motor-Generator-Kombination proportionale Spannung an, die vom Generator induziert wird. Störspannungen, die sich der Spannung Ug überlagern könnten, werden durch das RC-Glied R2/C1 unterdrückt.

Über den LAST-Anschluß kann die vom Generator induzierte Spannung mit einem Widerstand belastet werden.

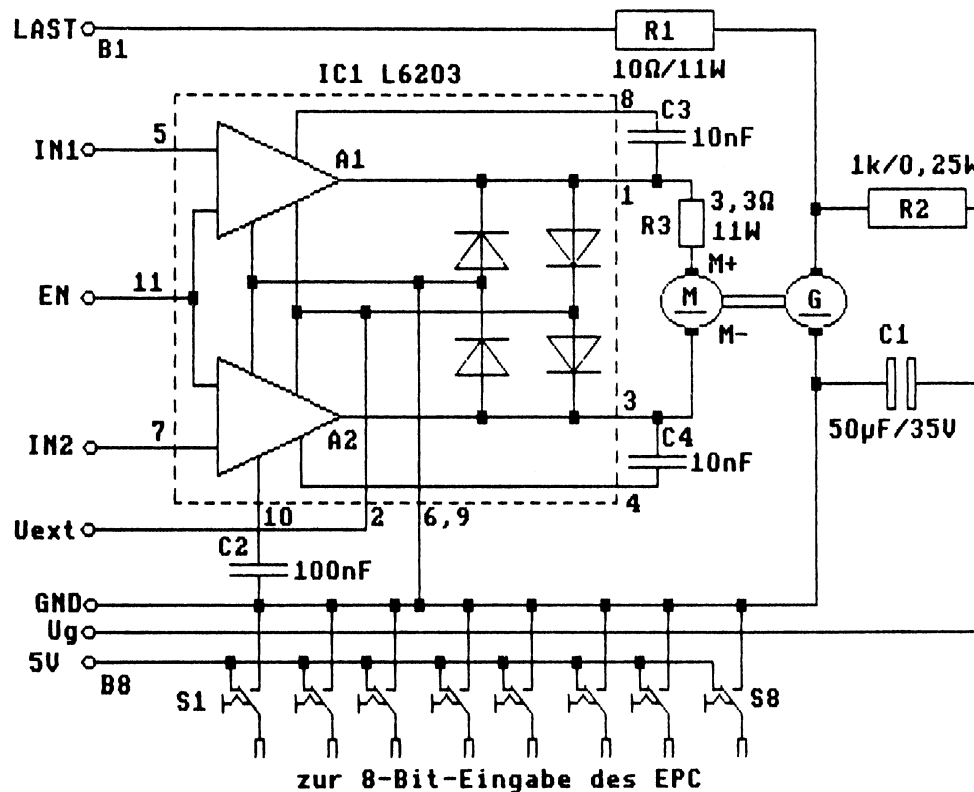


Bild 1: Der Schaltplan der Regelstrecke "Digitale Motor-Regelung".

## 2. Methodisch-didaktische Hinweise

### 2.1 Lernziele

Nach Bearbeitung der Übung kann der Lernende

- den Unterschied zwischen analoger und digitaler Steuerung und Regelung erklären;
- die Schritte einer Programmentwicklung aufzählen;
- einfache Programme zur Steuerung eines DC-Motors entwickeln;
- komplexere Programme zur Drehzahlregelung eines DC-Motors lesen;
- die Wirkungsweise einer Pulsbreiten-Steuerung erklären;
- die Einsatzgebiete von MC-Entwicklungssystemen und Einplatinen-Computern nennen.

### 2.2 Voraussetzungen

Die Lösung der vorliegenden Übung ist nur möglich, wenn der Lernende fundierte Kenntnisse sowohl der Steuerungs- und Regelungstechnik als auch der Mikrocomputer-Technik besitzt.

Es müssen die Inhalte der Fachtheoretischen Übungen 10.1 bis 10.4 sowie die Wirkungsweise einer A/D-Wandlung und der Unterschied zwischen stetigen und unstetigen Reglern bekannt sein.

Wird zum Aufbau der Regeleinrichtung zusätzlich der EPC mit der zugehörigen Ein/Ausgabe-Baugruppe eingesetzt, so werden auch die Kenntnisse der EPC-Unterlagen vorausgesetzt.

Die hier genannten Kenntnisse sollen durch die vorliegende Übung vertieft und erweitert werden. Außerdem sollen die Fähigkeiten der Problemanalyse und Programmentwicklung mit Hilfe von Flußdiagrammen und die Assemblerprogrammierung geschult werden.

Daneben bietet die Übung die Möglichkeit, die entwickelten Programme auf den EPC des MFA-Mediensystems zu übertragen. Dadurch kann der Unterschied zwischen der Problemlösung auf einem Entwicklungssystem und dem Einsatz des EPCs verdeutlicht werden.

### 2.3 Bereitstellungsliste

Die Übung erfordert – neben einem funktionsfähigen MC-System mit Monitor und Tastatur sowie einem EPC – keinen hohen technischen Aufwand, da die Regelstrecke nur aus handelsüblichen Bauteilen besteht.

Folgende Gerätekonfiguration ist erforderlich:

- MFA-Baugruppenträger mit Bus, Busabschluß, Trafo-Einschub und Spannungsregelung
- CPU-Baugruppe 8085
- \* - 8-K-RAM-Baugruppe (Basisadresse E000H)
- \* - EPROM-Baugruppe mit MAT85 und Software-Paket SP1 (Basisadresse 0000H)
- \* - EPROM-Baugruppe mit Mini-DOS (Basisadresse 4000H)
- Floppy-Interface mit Floppy-Disk-Laufwerk und formatierter Diskette (320 KByte)
- zwei Stück 8-Bit-Parallel-Ausgabe; Adressen: 00H, 01H
- 8-Bit-Parallel-Eingabe; Adresse :01H
- AD/DA-Wandler-Baugruppe; Adresse: 00H
- Video-Interface, Tastatur, Monitor
- Netzteil 0 - 15V/3A
- komplett aufgebaute Regelstrecke zur digitalen Motor-Regelung
- Lastpotentiometer 100 Ohm / 100 Watt
- drei analoge Vielfach-Meßinstrumente
- Zweikanal-Oszilloskop 10 MHz

Wenn die Regeleinrichtung zusätzlich mit Hilfe des EPCs realisiert werden soll, müssen der EPC und die zugehörige Ein/Ausgabe-Karte I/O-EPC mit Adapterkarte funktionsfähig bereitgestellt werden. Außerdem wird dann die EPROM-Programmierer-Baugruppe des MFA-Systems benötigt.

- \* **ersatzweise 64K-RAM-Karte mit MAT 32K**

### 3. Hinweise zur Lösung der Übungen

#### 3.1 Einfache Motorsteuerung mit dem Baustein L6203

##### 3.1.1 Funktionsbeschreibung

Der Baustein L6203 (siehe Bild 2) enthält zwei Leistungsverstärker, die TTL-kompatible Eingänge besitzen. Über den Enable-Eingang EN können die Verstärkerausgänge in den hochohmigen Zustand geschaltet werden, oder sie übernehmen den logischen Pegel der Eingänge IN1 und IN2. Die Ausgangsspannung und der Ausgangsstrom der Verstärker ist dabei von der externen Betriebsspannung  $U_{ext}$  abhängig. Um die Verstärker bei induktiven Lasten vor Zerstörung zu schützen, sind zusätzlich vier Dioden im Baustein integriert.

Die Ausgangsverstärker können einen maximalen Laststrom von 3A treiben, wobei die Gesamtverlustleistung des Bausteins 25W nicht überschreiten darf.

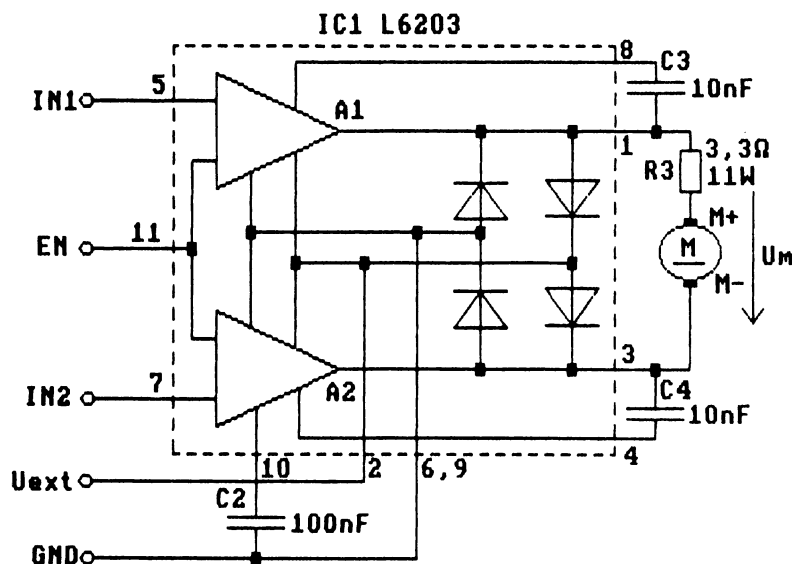


Bild 2: Der Aufbau des Bausteins L6203, hier mit angeschlossenem Gleichstrom-Motor (M) dargestellt.

### 3.1.2 Die Steuerung des Motors

Erhält der Eingang EN einen H-Pegel, so liegen an den Verstärkerausgängen die logischen Pegel der Eingänge IN1 und IN2 an. Die Spannung  $U_m$  zwischen den Motorklemmen M+ und M- ist dann entweder positiv oder negativ. Der Motor dreht entweder rechts oder links.

Erhält der Eingang EN dagegen einen L-Pegel, so werden die Verstärkerausgänge hochohmig: Die Spannung  $U_m$  sinkt auf null Volt ab, und die Drehzahl des Motors sinkt. Da die Selbstinduktionsspannung des Motors aber keinen Strom treiben kann (hochohmige Verstärkerausgänge), dreht der Motor langsam aus. Würde man dagegen den Eingang EN mit H-Pegel beschalten und die Eingänge IN1 und IN2 mit L-Pegel, so würde der Motor abrupt stehenbleiben, da die gesamte Selbstinduktionsspannung der ursprünglichen Drehrichtung entgegenwirkt.

In beiden Fällen besitzt diese Art der Motorsteuerung aber einen erheblichen Nachteil:

Die Drehzahl des Motors kann nur den Minimalwert oder den Maximalwert annehmen (siehe Tabelle 1). Zwischenwerte für die Drehzahl sind nur über eine Veränderung der externen Betriebsspannung möglich. In der Praxis ist diese Art der Drehzahlbeeinflussung aber nicht üblich.

EN	IN1	IN2	A1	A2	$U_m$	Drehzahl	Richtung
L	X	X	Z	Z	0V	sinkt	Y
H	L	L	L	L	0V	null	Y
H	L	H	L	H	- $U_{ext}$	Max.	Links
H	H	L	H	L	$U_{ext}$	Max.	Rechts

Tabelle 1: Die Pegel am Baustein L6203 und ihre Wirkung auf den angeschlossenen Motor. Erläuterung zu den Pegelangaben: X = beliebiger Pegel, Z = hochohmig, Y = beliebige Richtung.

Bevor die folgenden Seiten durchgearbeitet werden, muß das MC-System mit allen auf Seite 6 aufgeführten Baugruppen bestückt und die Regelstrecke bereitgestellt werden. Außerdem benötigen Sie das ebenfalls auf Seite 6 aufgeführte Netzteil 0 - 15V/3A.

### 3.2 Beispiellösung für Motor-Linkslauf

Mit Hilfe des in Bild 4 abgedruckten Programms kann der Motor bei maximaler Drehzahl links drehen oder gestoppt werden. Die Entscheidung wird über das Bit B7 der 8-Bit-Eingabe-Baugruppe (Adr. 01H) getroffen. Ist B7 = L, so läuft der Motor langsam aus; ist dagegen B7 = H, dann läuft der Motor an.

Bevor Sie das Programm nach Bild 4 ab der Adresse E000H eingeben, müssen MC-System und Regelstrecke miteinander verbunden werden. Außerdem muß das externe Netzteil an die Regelstrecke angeschlossen werden, wie in Bild 3 gezeigt.

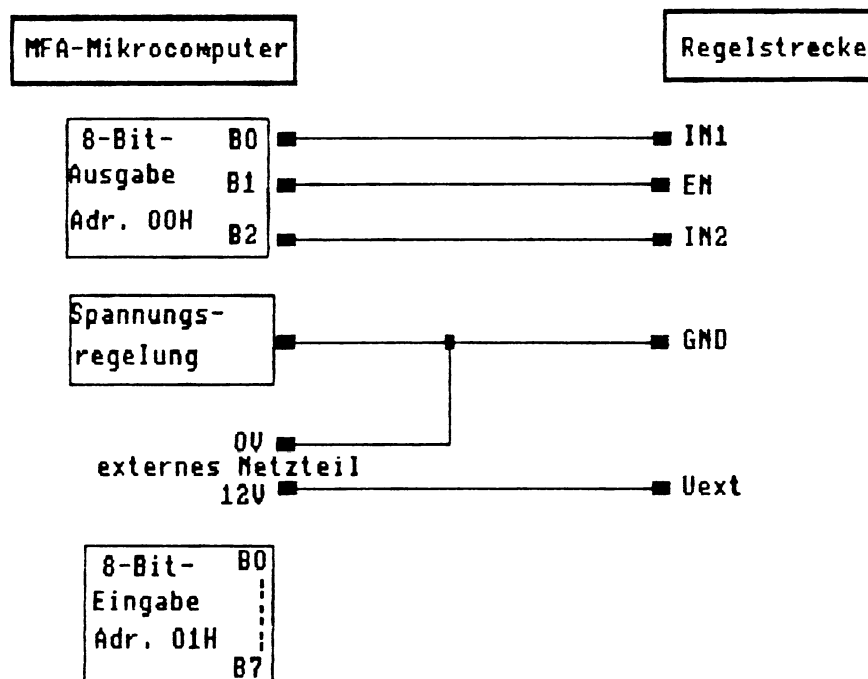


Bild 3: Die Verbindung von MC-System und Regelstrecke.



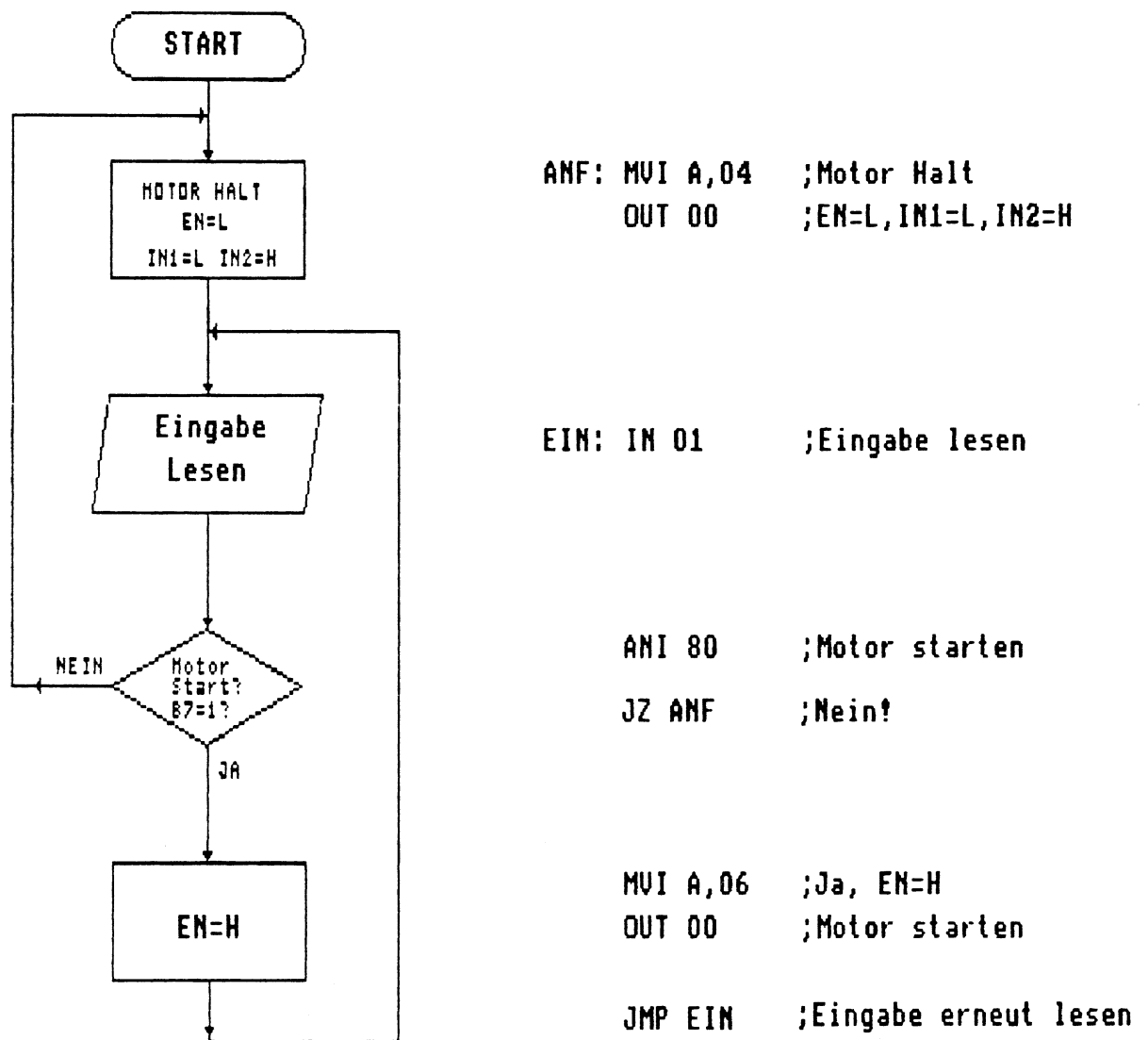


Bild 4: Flußdiagramm und Assemblerprogramm für Motor-Linkslauf.

### 3.2.1 Programmieraufgabe

Entwickeln Sie in Anlehnung an das eben vorgeführte Beispiel ein Flußdiagramm und ein Assemblerprogramm für Rechtslauf des Motors. Außerdem soll der Motor nicht langsam ausdrehen, sondern abrupt

stoppen, wenn das Bit B7 der 8-Bit-Eingabe-Baugruppe (Adresse 01H) auf null gesetzt wird.

Hinweis: Wenn Sie ihre Lösung überprüfen wollen, finden Sie im Anhang einen Lösungsvorschlag.

### 3.3 Verbesserte Steuerung

#### mit digitaler Einstellung der Drehzahl

Da die Betriebsspannung  $U_{ext}$  der Leistungsverstärker und damit des Motors in der Praxis konstant ist, kann die Drehzahl des Motors nur über die Ansteuerung beeinflusst werden.

Dazu werden an die Verstärkereingänge konstante Pegel für Rechts- bzw. Linkslauf angelegt. Der Eingang EN erhält jetzt jedoch keinen konstanten Pegel, sondern wird mit einer Rechteckspannung angesteuert, deren Impulszeit oder Pausenzeit geändert werden kann.

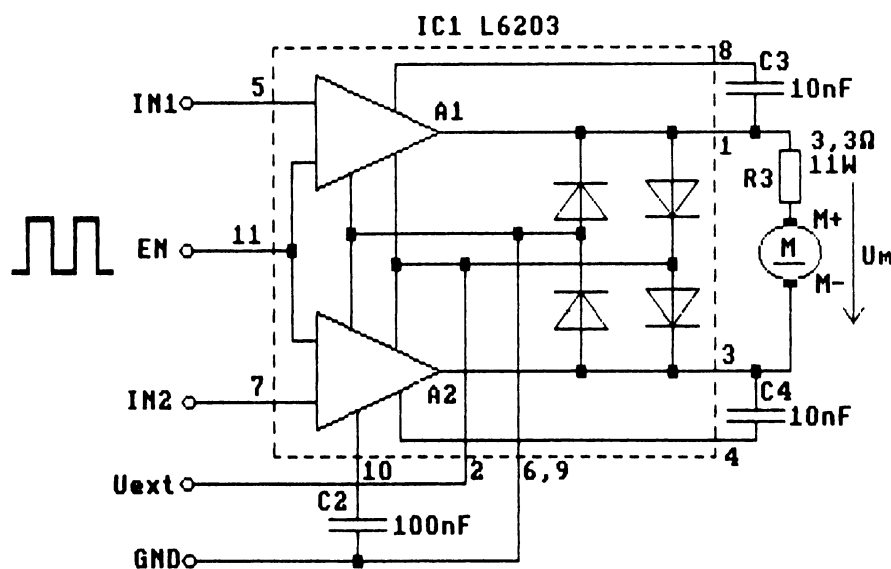


Bild 5: Die Beschaltung des Eingangs EN mit Rechteckimpulsen.

IN1 und IN2 erhalten gleichbleibende Pegel, z.B. für Linkslauf.

### 3.3.1 Motor-Spannung und -Drehzahl in Abhängigkeit vom Impuls/Pausen-Verhältnis der Steuerspannung am Eingang EN

Je nach dem Verhältnis der Impulszeit zur Pausenzeit der Steuerspannung am Eingang EN sinkt oder steigt der zeitliche Mittelwert der Motorspannung und damit die Drehzahl. In diesem Zusammenhang wird von einer Pulsbreiten-Steuerung gesprochen.

Zu beachten ist bei dieser Art der Steuerung, daß die Frequenz der Steuerspannung nicht zu groß und die Impulszeit nicht zu klein gewählt wird, da sonst der Motor nicht anläuft. Damit der Motor ohne Probleme läuft, darf das Impuls/Pausen-Verhältnis nicht unter 20 % absinken. Die Frequenz der Steuerspannung an EN darf einen Maximalwert von 2,8 kHz nicht überschreiten.

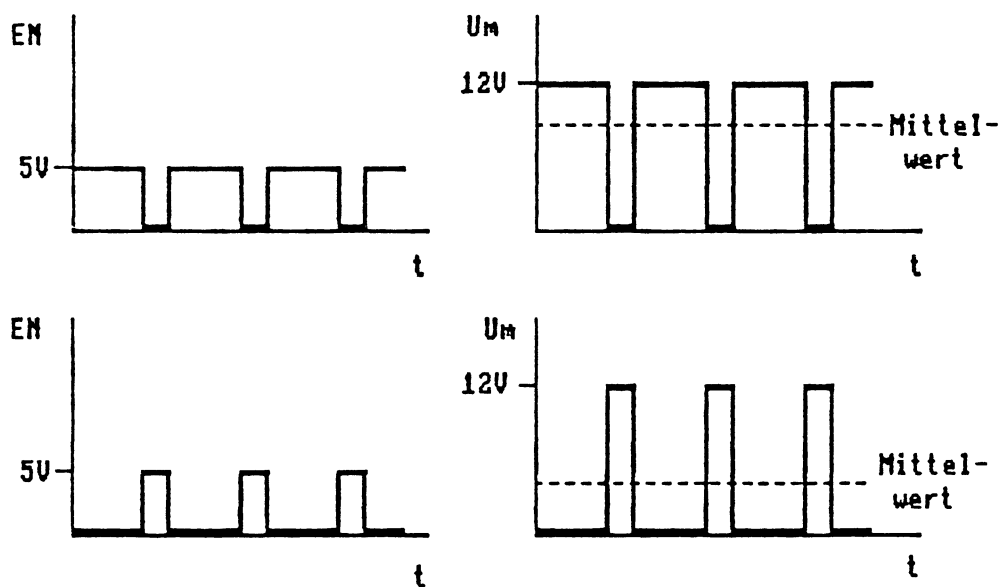


Bild 6: Der Verlauf der Steuerspannung, der Motorspannung und deren Mittelwert für zwei unterschiedliche Impulszeiten.

### 3.3.2 Beispiel-Lösung

Die Drehzahl des Motors wird in dem folgenden Programm indirekt durch zwei Zeitschleifen festgelegt; durch ihren Wert wird die Impulszeit der Rechteckspannung am Eingang EN bestimmt, deren

Werte nur vor dem Programmstart durch das MEMORY-Kommando des MFA-Systems eingestellt werden können. Dazu müssen in den Zeitschleifen Z1 und Z2 die Werte geändert werden, mit denen das Register B geladen wird. Zur Vereinfachung wird in diesem Beispiel die Pausenzeit konstant auf ca. 0,014ms gehalten.

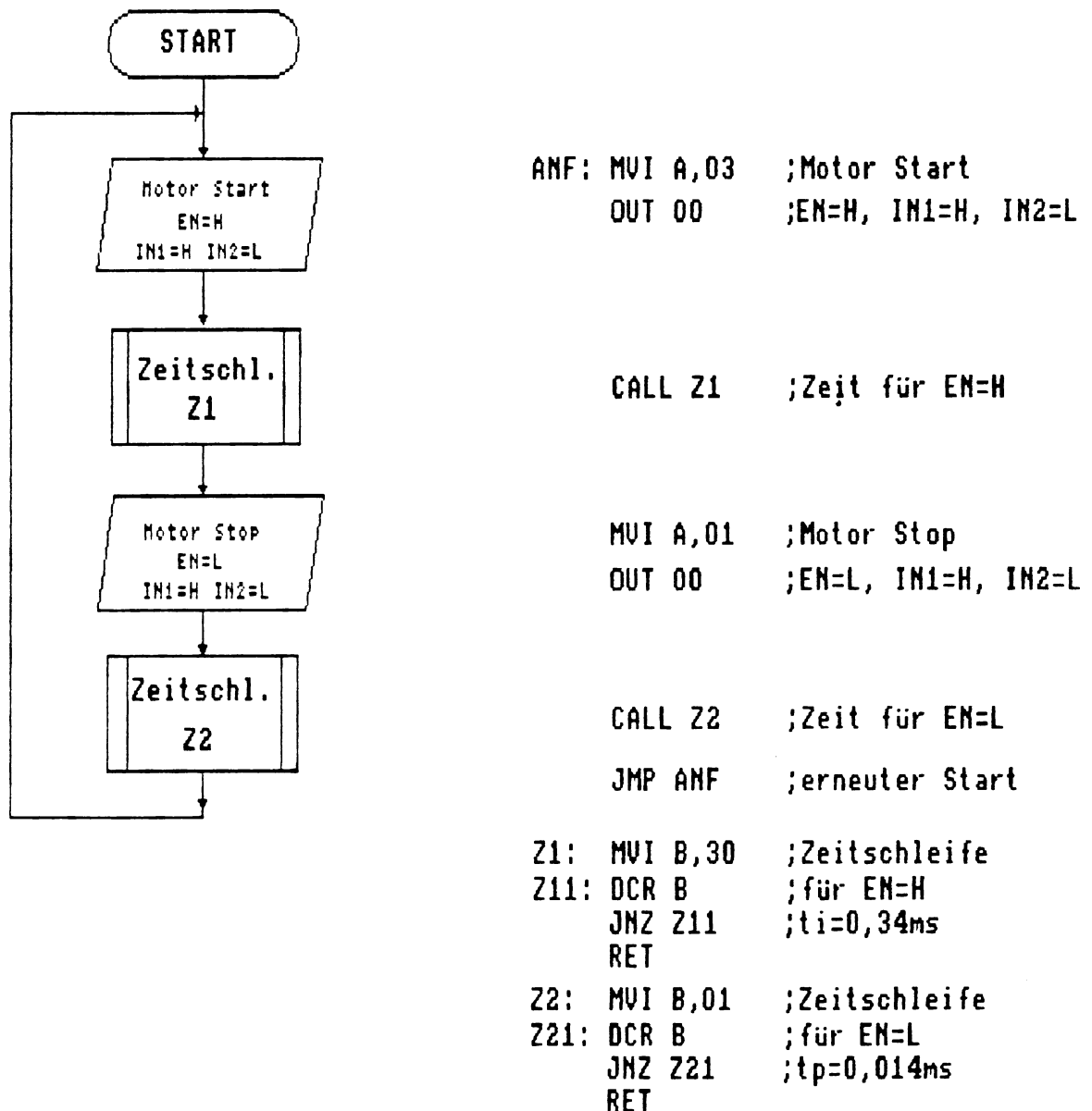


Bild 7: Flußdiagramm und Assemblerprogramm zur Beispiel-Lösung  
mit zwei Zeitschleifen.

Ändern Sie den Wert für das Register B in der Zeitschleife Z1 nach der folgenden Tabelle, messen Sie jeweils die Motorspannung  $U_m$  und tragen Sie Ihre Meßergebnisse in Tabelle 2 ein.

Tabelle 2:

Reg. B	$U_m/V$	Reg.B	$U_m/V$
10H		80H	
20H		D0H	
40H		FFH	
60H		00H	

Zeichnen Sie in das Koordinatensystem von Bild 8 die eben ermittelten Funktionswerte ein.

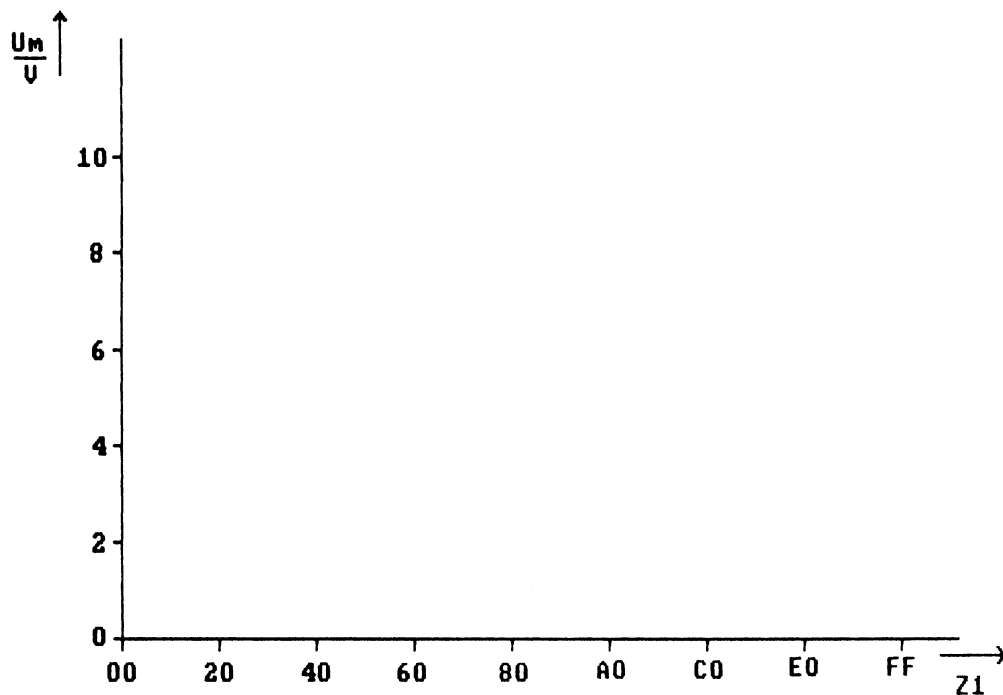


Bild 8: Die Motorspannung in Abhängigkeit vom Wert der Zeitschleife Z1.

Diskussion der Meßergebnisse:

Erklären Sie, warum bei dem Wert 00H im Register B die Drehzahl nicht geringer ist als bei den anderen Eingabewerten!

Antwort: \_\_\_\_\_

\_\_\_\_\_

### 3.3.3 Programmieraufgabe

Entwickeln Sie in Anlehnung an das vorherige Beispiel je ein Flußdiagramm und ein Assemblerprogramm für Rechts- und für Linkslauf des Motors. Die Impulszeit  $t_i$  der Steuerspannung am Eingang EN soll mit Hilfe der Zeitschleife Z1 durch die 8-Bit-Eingabe-Baugruppe (Adresse 01H) beeinflußt werden.

Hinweis:

Wenn Sie Ihre Lösung überprüfen wollen, finden Sie im Anhang einen Lösungsvorschlag.

Wenn Sie die Programme auf Richtigkeit überprüft haben, speichern Sie sie auf einer Floppy-Disk ab, da sie im folgenden noch benötigt werden.

## 3.4 Beeinflussung der Motordrehzahl durch die Belastung des Generators

### 3.4.1 Grundlagen

Durch die starre Verbindung von Motor und Generator auf der Regelstrecke erzeugt der vom Motor angetriebene Generator eine Spannung, die von der Drehzahl des Motors abhängt.

Messen Sie die Generatorspannung  $U_g$  in Abhängigkeit vom eingestellten Wert an der 8-Bit-Eingabe-Baugruppe (Adr. 01H). Tragen Sie Ihre Meßergebnisse in Tabelle 3 ein. Zur Durchführung des Versuchs benötigen Sie die beiden im vorigen Schritt entwickelten Programme.

Tabelle 3:

Rechtslauf		Linkslauf	
8-Bit-Eingabe	U <sub>g</sub> /V	8-Bit-Eingabe	U <sub>g</sub> /V
10H		10H	
20H		20H	
30H		30H	
50H		50H	
80H		80H	
A0H		A0H	
C0H		C0H	
FFH		FFH	

Übertragen Sie die eben ermittelten Funktionswerte in das Koordinatensystem von Bild 9.

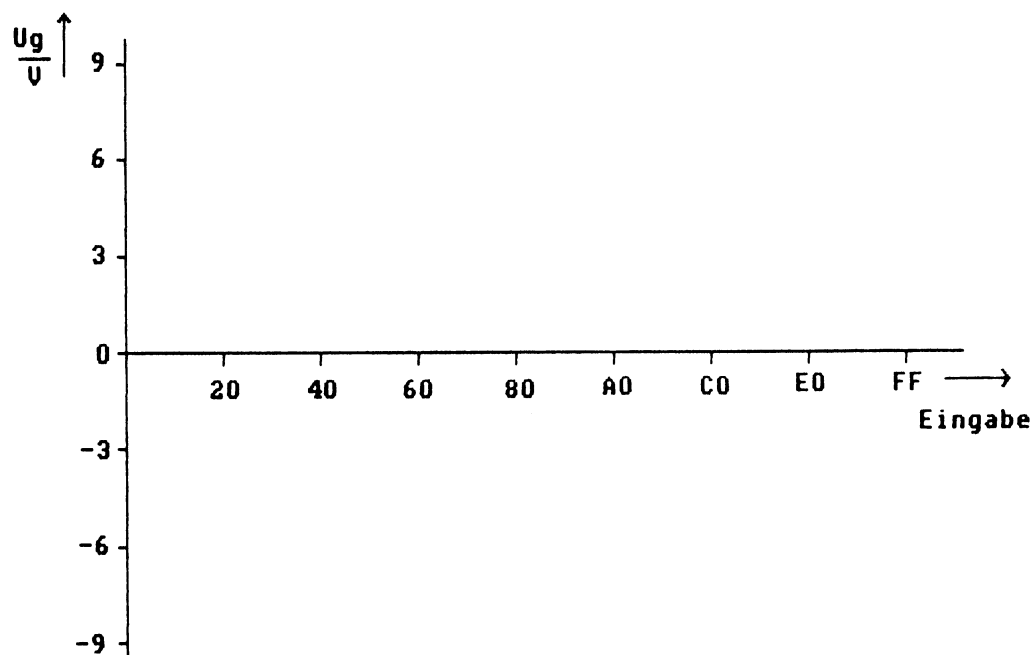


Bild 9: Die Generatorspannung in Abhängigkeit vom Wert an der 8-Bit-Eingabe-Baugruppe.

Erklären Sie hier mit einigen Worten, welchen Einfluß die Drehzahl des Motors auf die Generatorspannung hat, und wie sich eine Drehrichtungsumkehr auswirkt.

---

---

---

#### 3.4.2 Beeinflussung von Motordrehzahl und Generatorspannung durch eine ohmsche Belastung

Wird die im Generator G (siehe Bild 1) induzierte Spannung belastet, so muß sie einen Strom treiben. Dem Generator wird Leistung entnommen. Der durch die Generatorspannung getriebene Strom erzeugt aber seinerseits in den Generatorwicklungen ein Magnetfeld, das der Drehung entgegenwirkt. Als Folge davon nimmt die Drehzahl mit steigender Belastung ab. Außerdem wird der vom treibenden Motor M aufgenommene Strom ansteigen, da ja an der Motorwelle mehr mechanische Leistung abgegriffen wird und ein stärkeres Drehmoment erzeugt werden muß.

Die folgende Messung soll diesen Zusammenhang verdeutlichen.

Belasten Sie, wie in Bild 10 gezeigt, die Generatorspannung mit einem 100-Ohm/100-Watt-Potentiometer am Anschluß LAST. Messen Sie die vom Generator erzeugte Spannung  $U_g$ , den fließenden Laststrom  $I_L$  und den vom Motor aufgenommenen Strom  $I_m$ . Benutzen Sie eines der Programme von Abschnitt 3.4.1 und stellen Sie auf der 8-Bit-Eingabe-Baugruppe die Adresse 80H ein.

Die Meßergebnisse tragen Sie in Tabelle 4 auf der folgenden Seite ein.



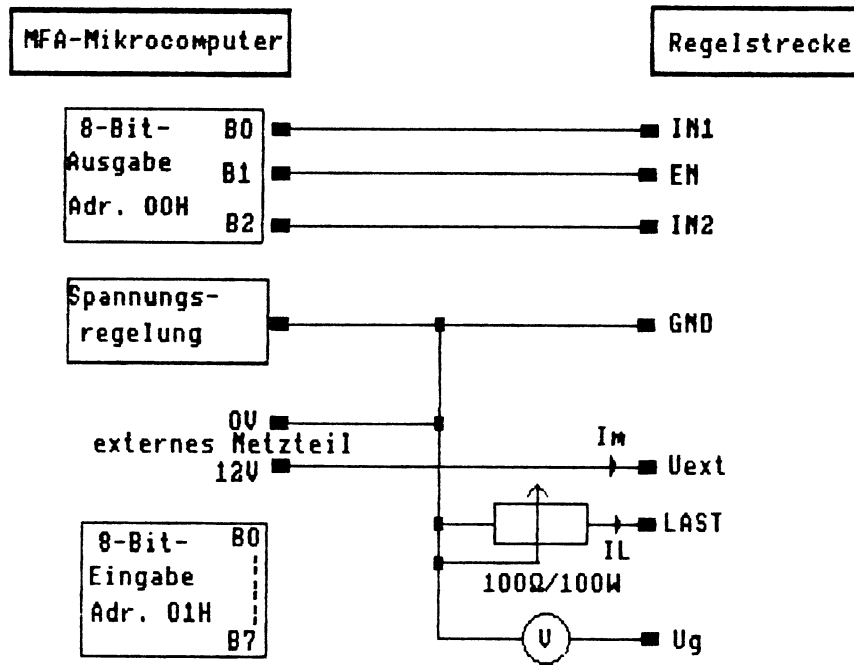


Bild 10: Meßaufbau zur Erfassung des Motorstromes  $I_m$ ,  
des Laststromes  $I_L$  und der Generatorspannung  $U_g$ .

Tabelle 4:

$I_L/A$	$U_g/V$	$I_m/A$
0,1		
0,15		
0,2		
0,25		

Übertragen sie die ermittelten Meßwerte auch in das Koordinatensystem von Bild 11.

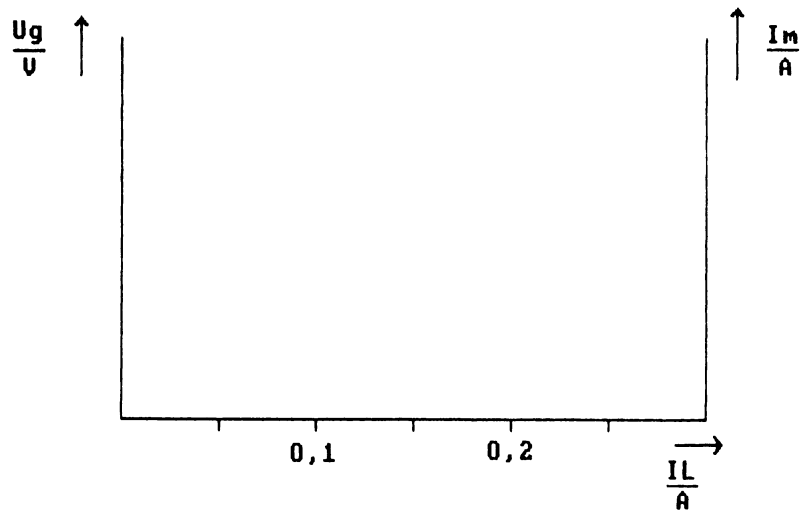


Bild 11: Generatorspannung und Motorstrom in Abhängigkeit vom Laststrom.

#### 4. Aufbau und Wirkungsweise eines digitalen Zweipunktreglers zur Drehzahlregelung eines DC-Motors

##### 4.1 Grundlagen

Zweipunktregler zählen zu den unstetigen Reglern, denn sie haben eine mehr oder weniger große Schalthysterese. Je größer die Hysterese, desto kleiner ist die Schalthäufigkeit des Stellgliedes, und desto größer sind die Schwankungen der Regelgröße.

##### 4.2 Der Aufbau des Regelkreises

Die Regelstrecke wird in dem Aufbau von Bild 1 durch die Einheiten Motor und Generator gebildet.

Die Regelgröße (Drehzahl) soll weitgehend unabhängig von der Störgröße (Belastung der Generatorspannung) konstant gehalten

werden. Ein Maß für den momentanen Wert der Regelgröße (Istwert), ist die vom Generator erzeugte Spannung  $U_g$ . Verändert sich der Istwert gegenüber dem Sollwert, so muß die Stellgröße (der dem Motor zugeführte Strom  $I_m$ ) in entgegengesetzter Richtung beeinflußt werden; steigt die Drehzahl an, so muß der Motorstrom abgesenkt werden und umgekehrt.

Die Regeleinrichtung, welche die Drehzahl des Motors regelt, wird in dieser Übung durch den Mikrocomputer realisiert, der ein entsprechendes Programm abarbeitet.

Die Regeleinrichtung muß folgende Aufgaben erfüllen:

- Erfassung des Istwertes;
- Vorgabe des Sollwertes;
- Vergleich zwischen Soll- und Istwert;
- Falls Soll- und Istwert nicht übereinstimmen, muß die Stellgröße entsprechend beeinflußt werden.

#### 4.3 Unterschied zwischen analogen und digitalen Regeleinrichtungen

Bei analog arbeitenden Regelkreisen kann die erzeugte Generatorspannung direkt zur Beeinflussung des Stellgliedes und damit der Stellgröße benutzt werden.

Bei dem hier beschriebenen Regelkreis müssen zwischen der analog arbeitenden Regelstrecke und der digital arbeitenden Regel-Einrichtung Schnittstellen eingebaut werden. Der analog vorliegende Istwert (Generatorspannung) muß in ein 8-Bit-Datenwort umgewandelt werden. Diese Aufgabe wird durch einen A/D-Wandler erfüllt. Die Stellgröße (d.h. der dem Motor zugeführte Strom) muß indirekt aus einem 8-Bit-Datenwort erzeugt werden. Diese Aufgabe fällt der 8-Bit-Ausgabe-Baugruppe in Verbindung mit dem Steuerbaustein L6203 zu, denn über die 8-Bit-Ausgabe-Baugruppe wird das Impuls/-Pausen-Verhältnis der Steuerspannung am Eingang EN beeinflußt. Diese Einheiten stellen damit einen D/A-Wandler dar.

Dem offensichtlich höheren Schaltungsaufwand im Vergleich zu einem analogen Regelkreis steht eine wesentlich einfachere Beeinflussung der Reglereigenschaften gegenüber. Bei digitalen Regeleinrichtungen muß nur das Programm geändert werden, um andere Reglereigenschaften zu erhalten.

Die Ansteuerung des Motors durch die 8-Bit-Ausgabe-Baugruppe in Verbindung mit dem Steuerbaustein wurde schon in Kapitel 3 beschrieben. Auf der folgenden Seite finden Sie eine Kurzbeschreibung zur Ansteuerung des A/D Wandlers. Weitergehende Informationen entnehmen Sie bitte den Unterlagen zur AD/DA-Baugruppe.

#### 4.4 Ansteuerung des A/D-Wandlers durch die Generatorspannung

Im Kapitel 3.4.1 haben Sie die Abhängigkeit der Generatorspannung  $U_g$  von der Drehrichtung und dem an der 8-Bit-Eingabe-Baugruppe eingestellten Wert ermittelt. Dabei stellte sich heraus, daß bei Rechtsdrehung des Motors eine positive und bei Linksdrehung eine negative Generatorspannung erzeugt wurde.

Der A/D-Wandler muß also je nach Drehrichtung positive oder negative Spannungswerte umwandeln. Bei der vorliegenden AD/DA-Wandler-Baugruppe besteht folgender Zusammenhang zwischen Eingangsspannung und 8-Bit-Datenwort am Ausgang:

$U_e$	8-Bit-Datenwort
0 Volt	80H
10 Volt	FFH
-10 Volt	00H

Innerhalb eines Programms zur Drehzahlregelung muß also beachtet werden, daß die Sollwertvorgabe bei Rechtsdrehung größer als 80H, bei Linksdrehung kleiner als 80H sein muß.

Gleiches gilt für die digitale Verarbeitung des vom A/D-Wandler erfaßten Istwertes der Generatorspannung.

Aus den Unterlagen des verwendeten A/D-Wandlers (ZN 427) geht außerdem hervor, daß für den Beginn einer Umwandlung ein Startimpuls erzeugt werden muß, der aus dem IN-Befehl des Mikroprozessors generiert wird. Die Adresse dieses IN-Befehls muß mit der auf der Baugruppe eingestellten Adresse 00H übereinstimmen.

Neben den in Bild 10 gezeichneten Verbindungen zwischen Regelstrecke und MC-System müssen Sie jetzt zusätzlich die Generatorspannung  $U_g$  dem A/D-Wandlereingang zuführen und den Masse-Anschluß der A/D-Wandler-Baugruppe mit dem GND-Anschluß der Regelstrecke verbinden. Außerdem wird jetzt die zweite 8-Bit-Ausgabe-Baugruppe mit der Adresse 01H eingesetzt.

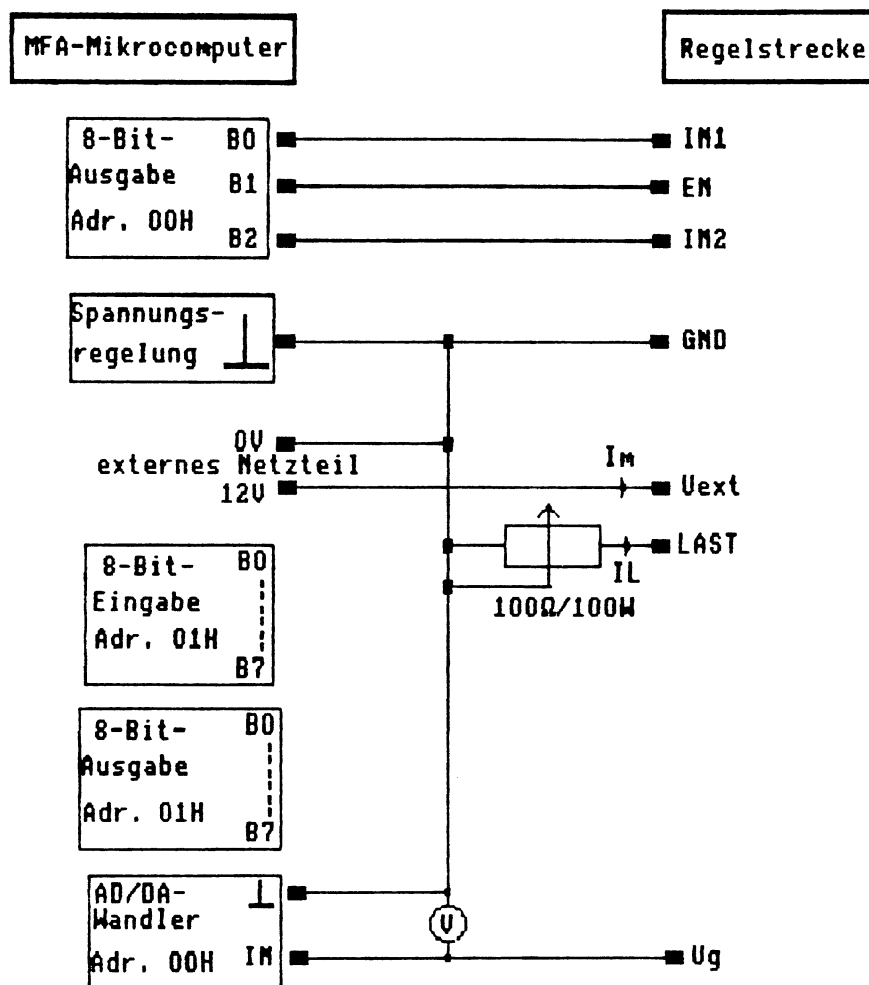


Bild 12: Gesamt-Blockschaltbild zur Verbindung von MC-System und Regelstrecke.

#### 4.5 Beispiellösung für einen digitalen Zweipunktregler

Bei der Entwicklung der Beispiellösung wurden folgende Punkte berücksichtigt:

- Der Motor soll nur rechts drehen.
- Der Sollwert wird über die 8-Bit-Eingabe-Baugruppe mit der Adresse 01H vorgegeben und mit dem 8-Bit-Wort der durch den A/D-Wandler umgewandelten Generatorspannung verglichen.
- Ist der Sollwert größer als der Istwert, wird der Programmteil GROSS durchlaufen; ist der Sollwert kleiner als der Istwert, wird der Programmteil KLEIN durchlaufen.
- In diesen beiden Programmteilen wird der Inhalt zweier Zählregister gegenläufig beeinflusst; d.h. wenn das eine Register incrementiert wird, wird das andere Register decrementiert.
- Je ein Register wird anschließend in je einer Zeitschleife benutzt, welche den Eingang EN mit H-Pegel und danach mit L-Pegel beschalten.

Dazu ein Beispiel:

Wird festgestellt, daß der Sollwert größer ist als der Istwert, so wird im Programmteil GROSS das Zählregister für H-Pegel an EN incrementiert und das für L-Pegel an EN decrementiert. Danach werden die beiden Zeitschleifen durchlaufen und ein erneuter Soll-Ist-Vergleich durchgeführt. Ist der Sollwert immer noch größer als der Istwert, so wird die H-Zeit weiter erhöht und die L-Zeit weiter verringert. Durch diese Maßnahme sind Impuls- und Pausenzeit an EN belastungs-abhängig.

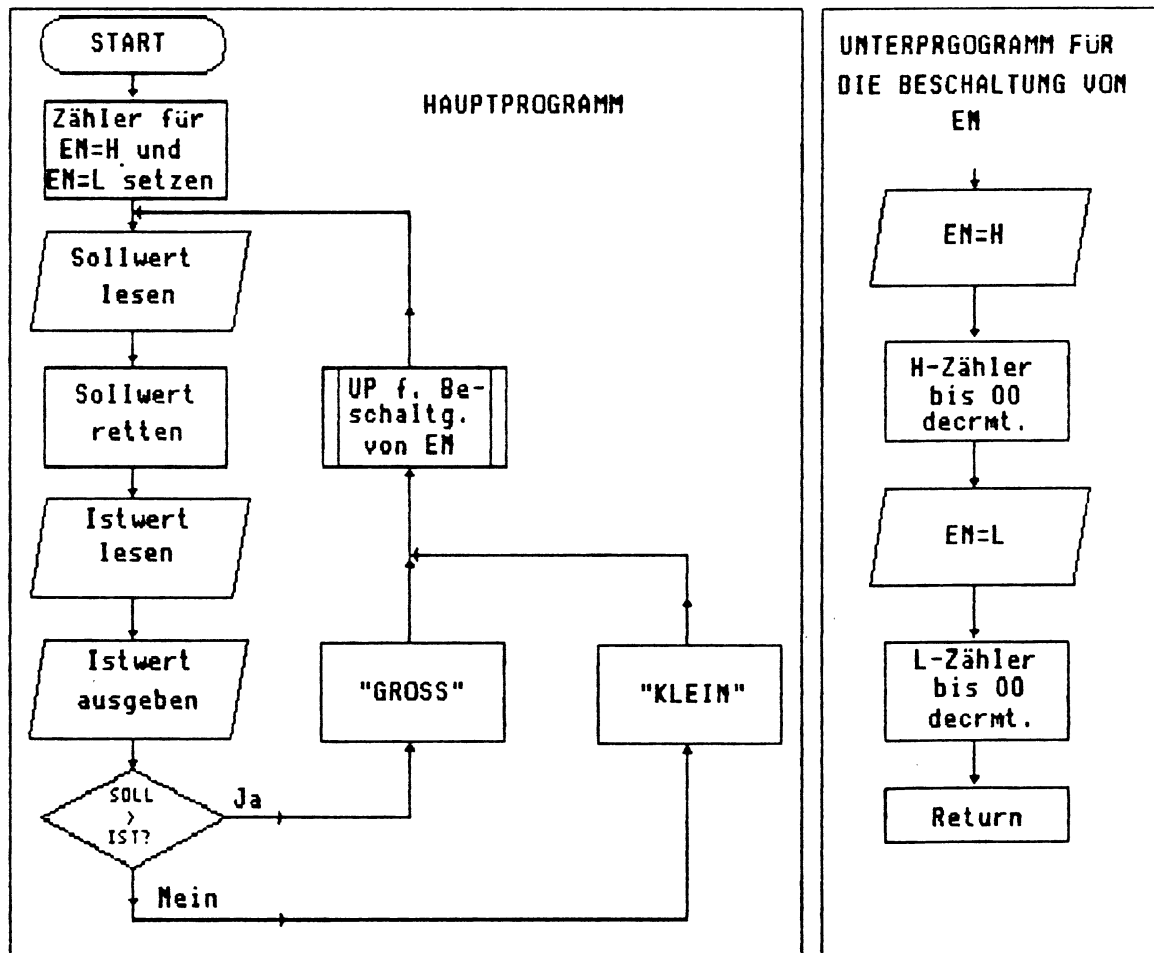


Bild 13: Flußdiagramme (Hauptprogramm und Unterprogramm) zur Beispiellösung. In Bild 14 finden Sie die Flußdiagramme für die Programmteile KLEIN und GROSS.

## Programmteil "GROSS"

## Programmteil "KLEIN"

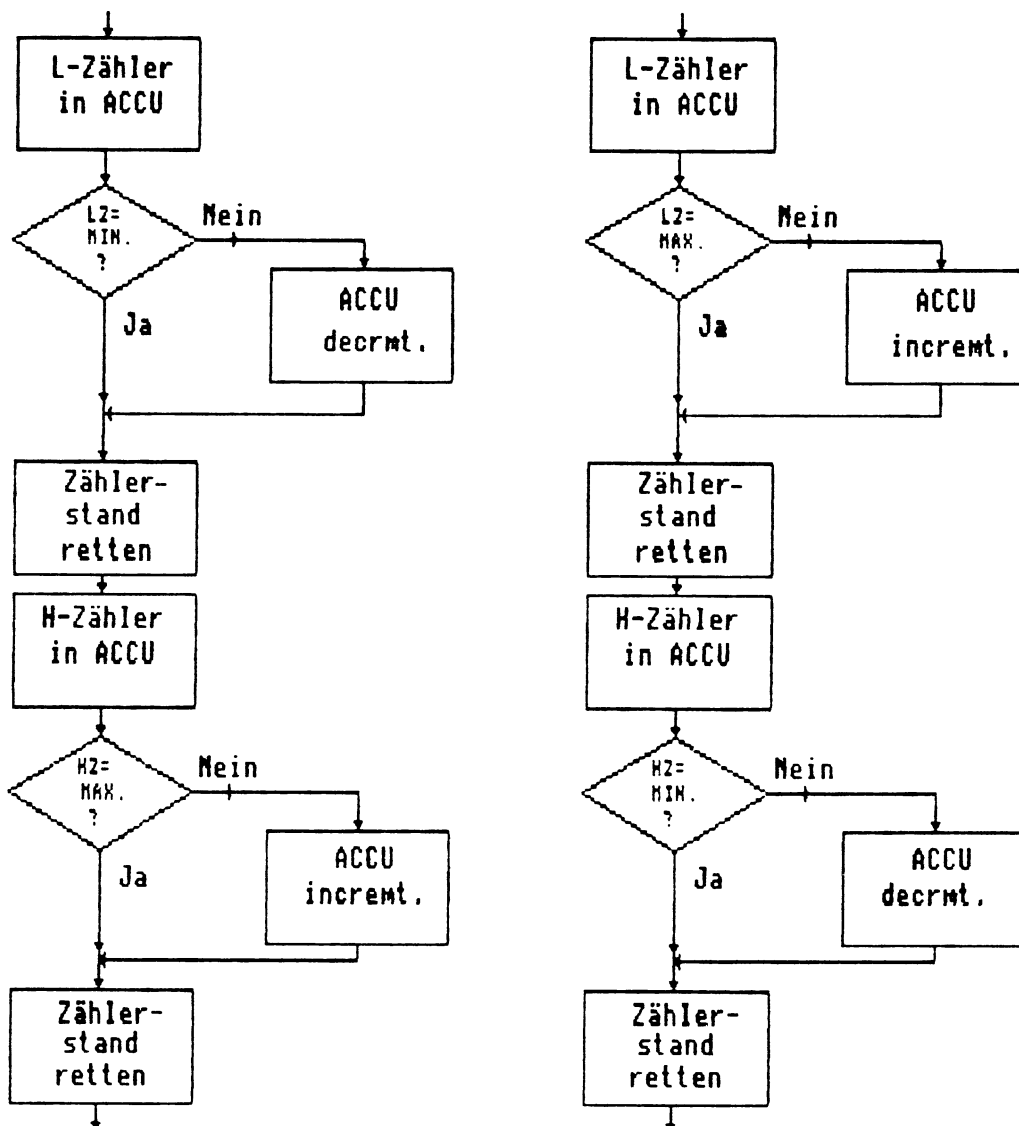


Bild 14: Flußdiagramme für die Programmteile  
GROSS (links) und KLEIN (rechts).

Erläuterungen zur den dargestellten Flußdiagrammen

Die auf Seite 23 erwähnten Zählregister werden durch zwei Register realisiert (E für H und D für L), deren Inhalte in den Programmteilen GROSS und KLEIN mit den Konstanten MIN und MAX verglichen werden. Diese Konstanten stellen das absolute Minimum



bzw. Maximum von Impuls- und Pausenzeit der Steuerspannung an EN dar. Durch den Vergleich wird sichergestellt, daß die Werte MIN bzw. MAX nicht unter- bzw. überschritten werden und somit die Impulszeit nicht zu groß bzw. die Pausenzeit nicht zu klein wird. Diese Maßnahme ist erforderlich, weil sonst die Impuls- bzw. die Pausenzeit zu null werden könnten.

#### Assemblerprogramm für den digitalen Zweipunktregler:

```

E003      MVI D,05      ;Zähler für EN = L
E005      MVI E,05      ;Zähler für EN = H
E007 EIN:  IN 01        ;Sollwert lesen
E009      MOV B,A       ;Sollwert retten
E00A      IN 00         ;Istwert lesen
E00C      NOP
E00D      OUT 01        ;Istwert ausgeben
E00F      CMP B         ;Soll>Ist?
E010      JNC KLEIN     ;Nein
E013      JMP GROSS     ;Ja
E016 KLEIN: MOV A,D      ;L-Zähler in Accu
E017      CPI 20        ;L-Zähler = Max.?
E019      JZ K1         ;Ja
E01C      INR A         ;Nein, LZ incremt.
E01D K1:   MOV D,A      ;Zählerstand retten
E01E      MOV A,E       ;H-Zähler in Accu
E01F      CPI 01        ;H-Zähler = Min.?
E021      JZ K2         ;Ja
E024      DCR A         ;Nein, HZ decremt.
E025 K2:   MOV E,A      ;Zählerstand retten
E026      CALL LAUF     ;EN mit H bzw. L beschalten
E027      JMP EIN       ;Soll. und Ist. erneut vergleichen
E02C GROSS: MOV A,D      ;L-Zähler in Accu
E02D      CPI 01        ;L-Zähler = Min.?
E02F      JZ G1         ;Ja
E032      DCR A         ;Nein, LZ decremt.
E033 G1:   MOV D,A      ;Zählerstand retten
E034      MOV A,E       ;H-Zähler in Accu
E035      CPI 50        ;H-Zähler = Max.?
E037      JZ G2         ;Ja
E03A      INR A         ;Nein, HZ incremt.
E03B G2:   MOV E,A      ;Zählerstand retten
E03C      CALL LAUF     ;EN mit H bzw. L beschalten
E03F      JMP EIN       ;Soll. und Ist. erneut vergleichen
E042 LAUF: MVI A,03      ;Eingang EN = H
E044      OUT 00
E046      MOV C,E       ;H-Zähler in C-Register
E047 L1:   DCR C        ;H-Zähler decremt.
E048      JNZ L1        ;H-Zähler = 00H?
E04B      MVI A,01      ;Ja, Eingang EN = L
E04D      OUT 00
E04F      MOV C,D       ;L-Zähler in C-Register
E050 L2:   DCR C        ;L-Zähler decremt.
E051      JNZ L2        ;L-Zähler = 00H?
E054      RET          ;Ja, zurück zum Hauptprog.

```

#### 4.6 Messungen an einem digitalen Zweipunktregler

Geben Sie das Programm von der vorigen Seite ab der Startadresse E003H ein. Danach muß das Programm zuerst auf einer Floppy-Disk gesichert werden, damit bei einem Programmabsturz keine komplette Neueingabe nötig ist. Belasten Sie die Generatorspannung mit dem 100-Ohm/100-Watt-Potentiometer am Ausgang LAST. Die 8-Bit-Eingabe-Baugruppe (Adresse 01H) ist auf AFH einzustellen. Starten Sie das Programm und messen Sie die Generatorspannung  $U_g$  sowie den Motorstrom  $I_m$  in Abhängigkeit vom Laststrom  $I_L$ . Tragen Sie die Ergebnisse in die nachfolgende Tabelle sowie in das Koordinatensystem von Bild 15 ein.

$I_L/A$	$U_g/V$	$I_m/A$
0,1		
0,15		
0,2		
0,25		

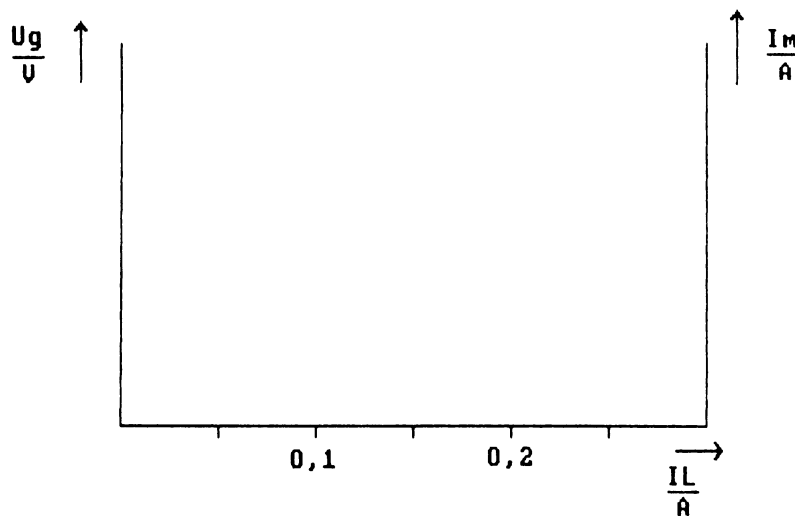


Bild 15: Generatorspannung und Motorstrom in Abhängigkeit vom Laststrom.

Messen Sie jetzt noch (mit einem Oszilloskop) den Spitze-Spitze-Wert des Wechselspannungs-Anteils der Generatorspannung  $U_g$

$$u_{gss} = \underline{\hspace{2cm}}$$

#### 4.7 Diskussion der Meßergebnisse

Mit steigendem Laststrom am Ausgang LAST nimmt der vom Motor aufgenommene Strom ebenfalls zu. Dadurch steigt das Drehmoment an der Welle an. Die Generatorspannung bleibt nahezu konstant, was auf eine fast konstante Drehzahl des Motors hinweist.

Der Arbeitsbereich der Regel-Einrichtung läßt sich durch einfache Programmänderung erweitern, so daß die Motordrehzahl auch bei größeren Lastströmen noch relativ konstant bleibt. Hierzu muß nur der Wert, mit dem das Register E bei der Programmadresse E035 verglichen wird, erhöht werden. Das Register E stellt, wie auf Seite 25 erwähnt, den Zähler für  $EN = H$  dar. Bei der Programmadresse E035 findet ein Vergleich mit der Konstanten MAX statt. Erhöht man den Wert dieser Konstanten, so kann die Impulszeit der Steuerspannung größere Werte annehmen, wodurch der Motorstrom und damit das Drehmoment größer werden kann.

#### 5. Realisierung der Regel-Einrichtung mit Hilfe des EPC

Das Programm von Seite 26 ist Teil einer Regeleinrichtung, die zur Drehzahlregelung von DC-Motoren eingesetzt werden kann. Das Entwicklungssystem (hier das MFA-System) ist aber nicht für den Praxiseinsatz vor Ort geeignet, da hier die Forderung nach möglichst geringem Platzbedarf besteht.

Dazu bietet sich ein Einplatinen-Computer mit entsprechender Ein/Ausgabe in idealer Weise an, da er in einer einzigen Baugruppe die gesamte Regeleinrichtung vereint, die bisher durch das gesamte MFA-System realisiert wurde.

### 5.1 Vorbereitung des MFA-Systems zur Anpassung des Programms an den EPC

Entfernen Sie aus dem abgeschalteten MFA-System folgende Baugruppen:

- die 8-Bit-Ausgabe-Baugruppen
- die 8-Bit-Eingabe-Baugruppe
- die AD/DA-Wandler-Baugruppe

Anstatt dieser Baugruppen setzen Sie die EPROM-Programmierer-Baugruppe und den I/O-Adapter mit der Ein/Ausgabe-Baugruppe (I/O-EPC) des EPC ein. Verbinden Sie dann die Regelstrecke mit Hilfe des Blockschaltbildes (Bild 16) mit der Ein/Ausgabe I/O-EPC.

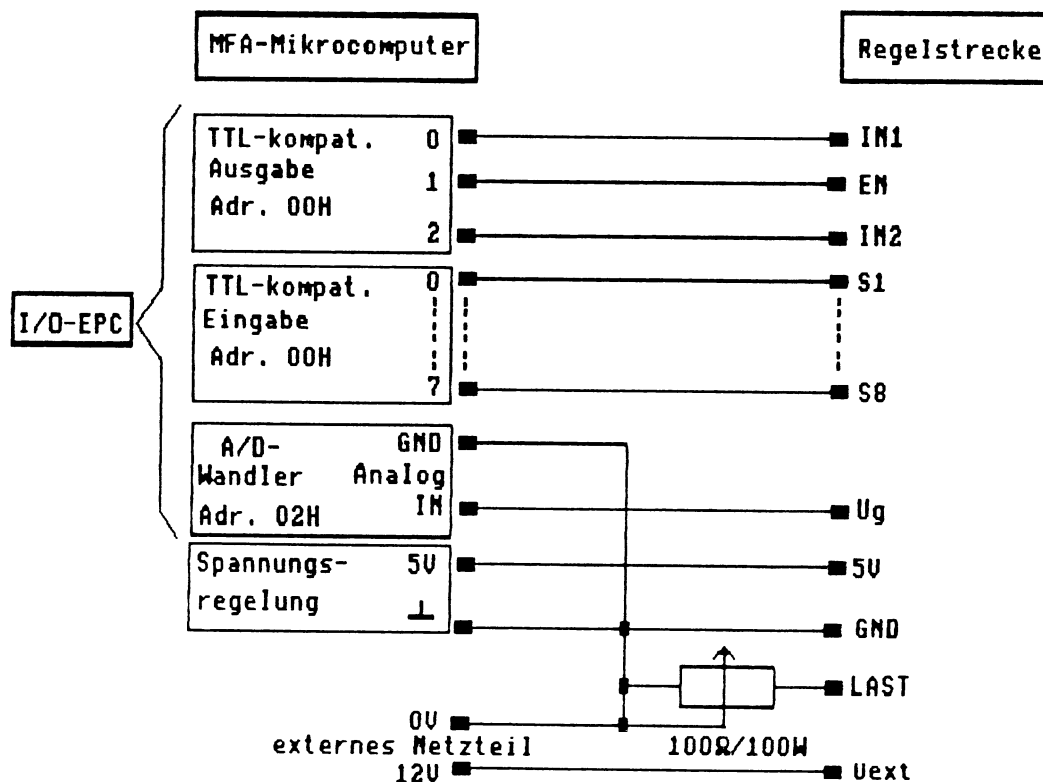


Bild 16: Verbindung zwischen Ein/Ausgabe-Baugruppe I/O-EPC und Regelstrecke

## 5.2 Anpassung des Programms an den EPC

- Der A/D-Wandler ist nur für positive Eingangsspannungen zwischen 0 Volt und 10 Volt ausgelegt. Der Wandler hat die Adresse 02H.
- Das 8-Bit-Datenwort am Ausgang des A/D-Wandlers liegt zwischen 00H = 0 Volt und FFH = 10 Volt. Die Vorgabe des Sollwertes, die jetzt über die Schalter S1 bis S8 auf der Regelstrecke erfolgt, muß deshalb entsprechend gewählt werden.
- Zum Lesen des 8-Bit-Datenwortes am Ausgang des A/D-Wandlers genügt nicht mehr ein einfacher IN-Befehl. Lesen Sie dazu die entsprechenden Informationen in den Unterlagen zur Ein/Ausgabe-Baugruppe I/O-EPC.
- Der erste Programmbefehl muß den Stackpointer des EPC auf die höchste RAM-Adresse 27FF setzen. Dieser Befehl darf allerdings bei einem Probelauf im MFA-System nicht abgearbeitet werden.
- Die Vorgabe des Sollwertes erfolgt über die TLL-kompatible 8-Bit-Eingabe, welche die Adresse 00H besitzt.

Nachfolgend ist das für den EPC-Einsatz modifizierte Programm wiedergegeben; dabei wurden zur besseren Übersicht nur die neuen bzw. geänderten Befehle kommentiert:

```
E000      LXI SP,27FF      ;Stackpointer des EPC initialis.
E003      MVI D,05
E005      MVI E,05
E007 EIN:  IN 00           ;Sollwert lesen
E009      MOV B,A
E00A      CALL ADW         ;A/D-Wandler initialisieren
E00D      IN 02            ;gewandelten Istwert lesen
E00F      CMP B
E010      JNC KLEIN
E013      JMP GROSS
E016 KLEIN: MOV A,D
E017      CPI 20
E019      JZ K1
E01C      INR A
E01D K1:  MOV D,A
E01E      MOV A,E
E01F      CPI 01
E021      JZ K2
```

---

```

E024      DCR A
E025 K2:   MOV E,A
E026      CALL LAUF
E029      JMP EIN
E02C GROSS: MOV A,D
E02D      CPI 01
E02F      JZ G1
E032      DCR A
E033 G1:   MOV D,A
E034      MOV A,E
E035      CPI 50
E037      JZ G2
E03A      INR A
E03B G2:   MOV E,A
E03C      CALL LAUF
E03F      JMP EIN
E042 LAUF: MVI A,03
E044      OUT 00
E046      MOV C,E
E047 L1:   DCR C
E048      JNZ L1
E04B      MVI A,01
E04D      OUT 00
E04F      MOV C,D
E050 L2:   DCR C
E051      JNZ L2
E054      RET
E055 ADW:  OUT OFF      ;Start der A/D-Wandlung
E057      MVI C,04      ;Zeitverzögerung
E059 AD1:  DCR C        ;für
E05A      JNZ AD1       ;A/D-Wandler
E05D      RET          ;zurück zum Hauptprogramm

```

### 5.3 Probelauf und Programmieren des EPROMs

Geben Sie das vorstehende Programm ab der Startadresse E000H ein. Dabei können Sie das auf Floppy-Disk gesicherte Programm von Seite 26 als Grundlage benutzen. Dieses Programm muß nur an einigen Adressen ergänzt bzw. geändert werden:

Ergänzungen:

Bei E000 muß ein LXI-Befehl eingegeben werden, und ab der Adresse E055 muß das Unterprogramm zur A/D-Wandlung eingegeben werden.

Änderungen:

```

E007: IN-Befehl ändern
E00A: CALL-Befehl eingeben
E00D: IN-Befehl eingeben

```

Sichern Sie auch das modifizierte Programm auf einer Floppy-Disk.

Bei dem jetzt erforderlichen Probelauf zur Überprüfung des Programms müssen Sie beachten, daß der erste Befehl nicht abgearbeitet werden darf. Dieser Befehl (LXI SP,27FF) setzt den Stackpointer der CPU auf die Adresse 27FF. Da Sie im Augenblick noch mit dem MFA-System arbeiten, ist ein Setzen des Stackpointers nicht erforderlich, da der Stackpointer durch das Betriebsprogramm MAT85 automatisch gesetzt wird.

Würde der Stackpointer trotzdem gesetzt, so hätte das zwangsläufig einen Absturz des Programms zur Folge, da sich bei der angegebenen Adresse (27FF) kein RAM-Speicher befindet.

Starten sie das Programm deshalb unbedingt ab der Adresse E003

Wenn das Programm richtig arbeitet, muß die Regelstrecke die gleichen Eigenschaften aufweisen, wie bei Bild 15 ermittelt. Eine indirekte Veränderung der Drehzahl ist jetzt über die Schalter S1 bis S8 auf der Regelstrecke möglich, deren Stellung den Sollwert darstellen. Die Schalterstellungen werden dann im Programm mit dem Istwert verglichen und die Drehzahl in Abhängigkeit von diesem Vergleich verändert.

### 5.3.1 Programmieren des EPROMs

Bevor das EPROM für den Einsatz im EPC programmiert wird, muß das Programm an den Adressbereich des EPC angepaßt werden. Dazu müssen diverse Sprungadressen auf den EPROM-Bereich des EPC zugeschnitten werden. Die Änderung ist am einfachsten über das MEMORY-Kommando des MFA-Systems möglich. Richten Sie sich dabei nach der folgenden Tabelle. Die zu ändernden Bytes sind dabei unterstrichen.

Programmadresse des zu ändernden Befehls	Alter Befehl	Neuer Befehl
E00A	CALL <u>E</u> 055	CALL 0055
E010	JNC <u>E</u> 016	JNC <u>00</u> 16
E013	JMP <u>E</u> 02C	JMP <u>00</u> 2C
E019	JZ <u>E</u> 01D	JZ <u>00</u> 1D
E021	JZ <u>E</u> 025	JZ <u>00</u> 25

---

E026	CALL	<u>E042</u>	CALL	<u>0042</u>
E029	JMP	<u>E007</u>	JMP	<u>0007</u>
E02F	JZ	<u>E033</u>	JZ	<u>0033</u>
E037	JZ	<u>E03B</u>	JZ	<u>003B</u>
E03C	CALL	<u>E042</u>	CALL	<u>0042</u>
E03F	JMP	<u>E007</u>	JMP	<u>0007</u>
E048	JNZ	<u>E047</u>	JNZ	<u>0047</u>
E05A	JNZ	<u>E059</u>	JNZ	<u>0059</u>

Gehen Sie bei der Änderung besonders sorgfältig vor, da das geänderte Programm nicht mehr auf dem MFA-System lauffähig ist, sondern nur mit dem EPC getestet werden kann.

Sie sollten deshalb auch das geänderte Programm zur Sicherheit auf einer Floppy-Disk abspeichern. Nach Sicherung des Programms kann das EPROM programmiert werden.

Schalten Sie dann das MFA-System und das externe Netzteil ab, entfernen außer der Ein/Ausgabe I/O-EPC, dem Trafo und der Spannungs-Regelung alle Einschübe, stecken den mit dem programmierten EPROM bestückten EPC in das System und schalten die Spannungsversorgung sowie das externe Netzteil wieder ein.

Wenn Sie das Programm richtig angepaßt haben, muß die Regel-Einrichtung die gleichen Eigenschaften wie beim Probelauf aufweisen.

Die Regel-Einrichtung ist jetzt auf einer Baugruppe zusammengefaßt und kann an beliebiger Stelle mit entsprechender Spannungsversorgung eingesetzt werden.

Damit ist die Übung beendet.



## 6. Anlagen

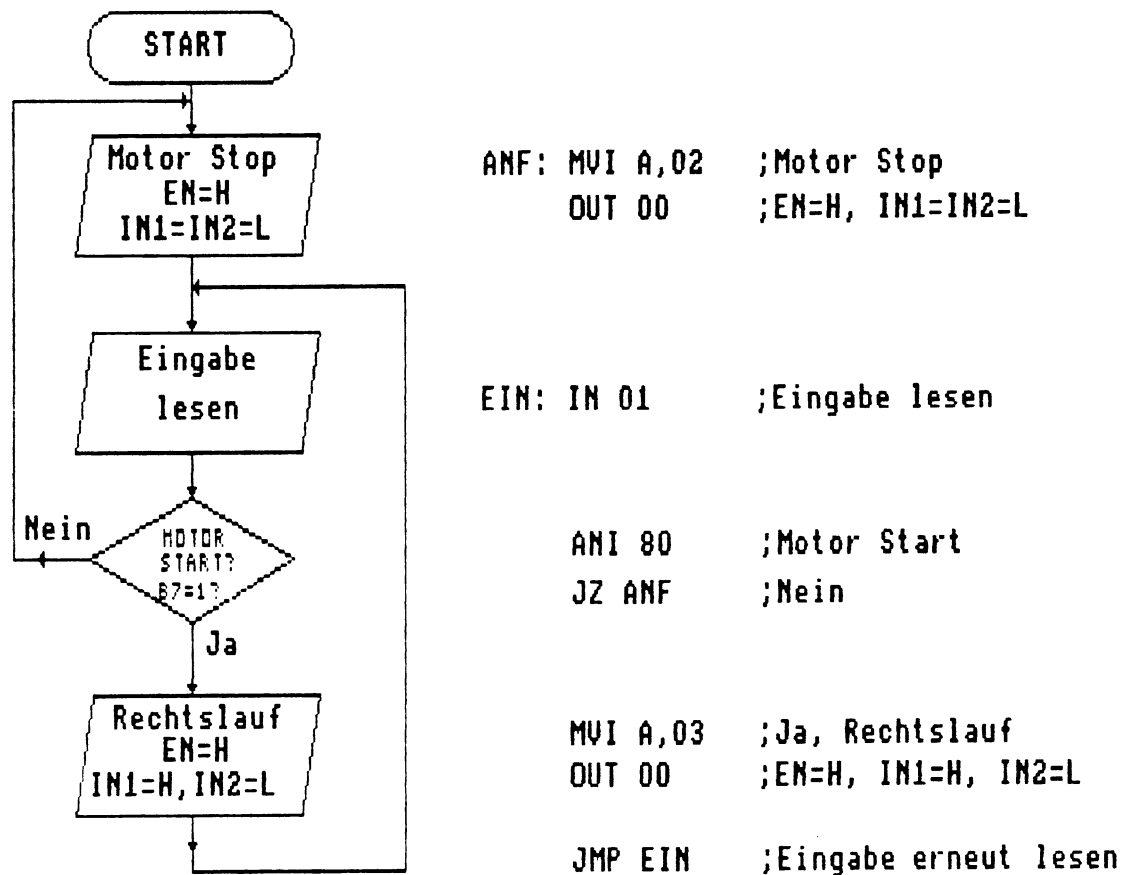
### 6.1 Überprüfung der Regelstrecke

Bevor die Übung durchgeführt wird, muß eine kurze Funktionsprüfung erfolgen. Verbinden Sie ein Netzteil ( $U = 12 \text{ Volt}$ ) mit den Steckbuchsen Uext und GND. Ein zweites Netzteil ( $U = 5 \text{ Volt}$ ) ist mit den Anschlüssen 5 V und GND zu verbinden. Die Steckbuchsen IN1 und EN sind mit 5V zu verbinden, die Steckbuchse IN2 ist mit GND zu verbinden. Wenn Sie beide Netzteile einschalten, muß sich die Motor-Generator-Kombination drehen. Messen Sie jetzt die Generatorspannung von der Klemme G+ zur Klemme G-. Es muß sich eine positive Spannung ergeben. Ist das nicht der Fall, sind die beiden Leitungen, welche den Generator mit der Platine verbinden, zu vertauschen.

Danach können Sie mit der Durchführung der Übung beginnen.

## 6.2 Lösungsvorschläge für die Programmieraufgaben

Zur Programmieraufgabe Seite 10:

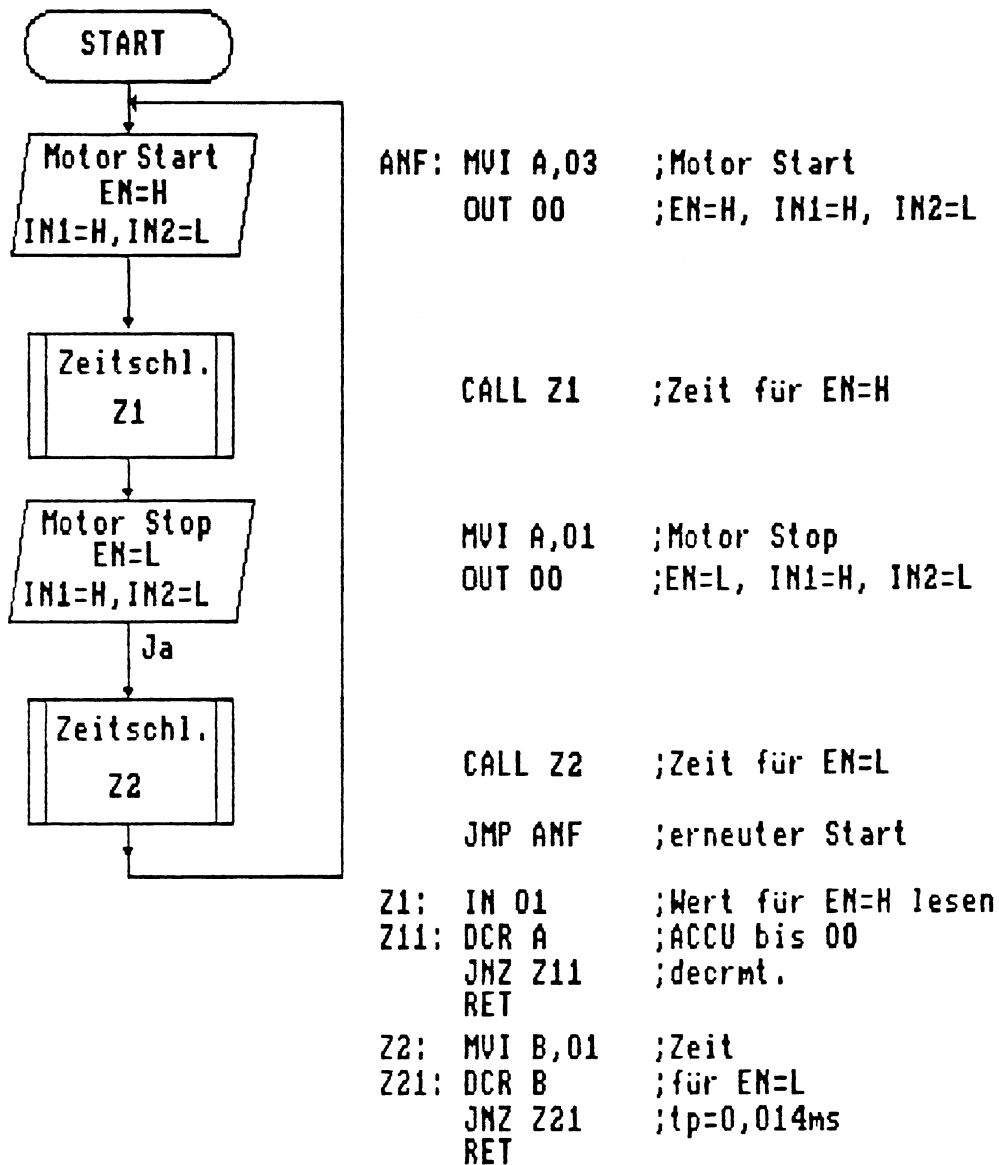


Antwort zur Frage auf Seite 15:

Wird das Register B mit dem Wert 00H geladen, so hat es nach dem ersten Decrementieren den Inhalt FFH. Die Zeitschleife wird also in diesem Fall 256-mal durchlaufen.

Zur Programmieraufgabe auf Seite 15 (Rechtslauf):

Für Linkslauf des Motors muß der ACCU bei "Motor Start" mit 06H und bei "Motor Stop" mit 04H geladen werden.

Antwort zur Frage auf Seite 17:

Die Generatorspannung steigt mit wachsender Drehzahl des Motors an. Eine Drehrichtungsumkehr hat eine Umkehr der Polarität der Generatorspannung zur Folge.

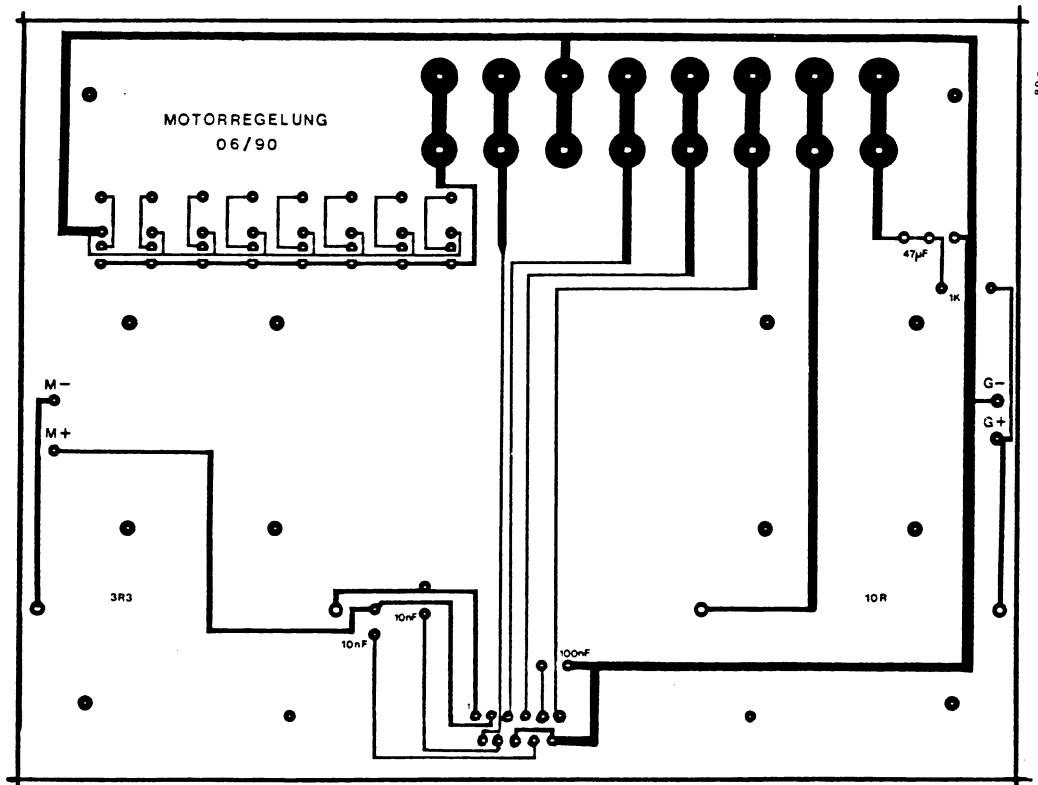


Bild 17: Layout Leiterplatte

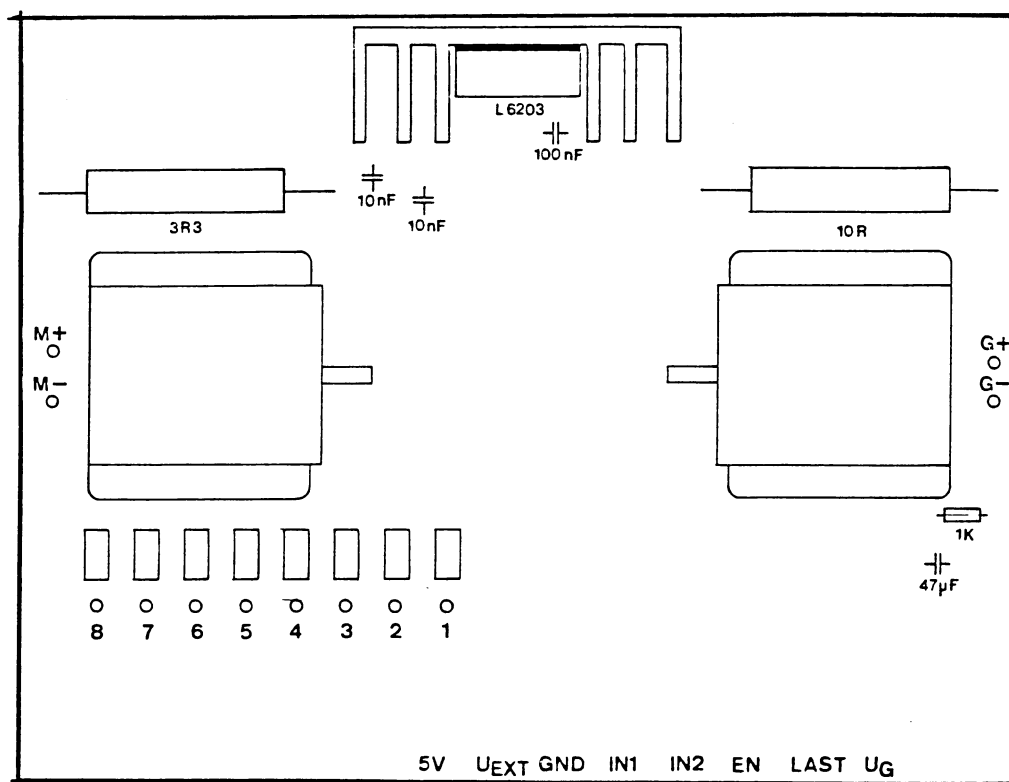


Bild 18: Bestückungsplan Leiterplatte

## =====

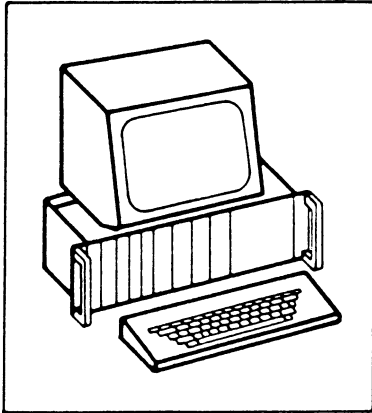
## M F A - COMPUTERSYSTEM S T Ü C K L I S T E

## =====

## M O T O R R E G E L U N G

POS	ARTIKEL	BAUFORM	EINZEL STCKZ.
1	Leiterplatte	Stärke 2,5mm 200*150	1
2	Bodenplatte	Stärke 1,5mm 200*150	1
3	Aufklebefüße		4
4	Distanzbolzen	M 3, 20 lang DI 641/20	4
5	Zylinderkopfschraube	M 3*8	8
6	"	M 3*10	3
7	"	M 4*10	2
8	Federscheibe	A 2,7	17
9	Sechskantmutter	M 3	11
10	Gleichstrommotor	GRAUPNER, SPEED 500 E 1788	2
11	Befestigungshalter	passend hierzu	2
12	Kupplung	passend hierzu	1
13	Litze	blau, 40mm lang	2
14	"	rot, 40mm lang	2
15	Kühlschelle	PR 42/37,5/SE	1
16	Befestigungswinkel	BE 631	2
17	Steuer-IC	L 6203	1
18	Widerstand	3R3 11 Watt	1
19	"	10 R 11 Watt	1
20	"	1 K 1/4 Watt	1
21	Kondensator	10 nF, RM 5, Ker.	2
22	"	100 nF, RM 5, Ker.	1
23	Elco	bipolar, 47uF/40V, radial	1
24	Miniatur-Schiebeschalter		8
25	Steckerstifte	1,2mm	8
26	Einbau-Buchse	Hirschmann 2mm blank, Mbu 1	8
27	Einbau-Buchse	Hirschmann 4mm blank, Bo 10	8

# FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



Einplatinen-Computer  
(EPC)





## Einplatinencomputer

Fachtheoretischer Teil	Seite
1. Einleitung	3
1.1. Blockaufbau des Einplatinencomputers	4
2. Zentraleinheit (CPU)	6
2.1. RESET-Erzeugung	7
2.2. Adreßsignal-Erzeugung und -Decodierung	8
2.3. Datensignale	9
2.4. Steuersignaldecodierung	10
2.5. Prüfplatine	11
2.5.1. Einzelschrittbetrieb der CPU	12
2.6. Interrupteingänge der CPU 8085	14
2.6.1. Restarteingänge RST5.5, 6.5 und 7.5	14
2.6.2. Interrupt-Eingang INTR und -Ausgang INTA	15
2.7. Steuersignale S0 und S1	15
2.8. Ein- und Ausgänge zur seriellen Datenübertragung	16
2.9. Speicherbereich des Einplatinencomputers	16
2.9.1. EPROM-Speicherbereich	16
2.9.2. RAM-Speicherbereich	17



## Einplatinencomputer

Fachpraktischer Teil	Seite
Flußdiagramm für den Arbeitsablauf	18
Bereitstellungsliste Einplatinencomputer	19
Bereitstellungsliste Prüfplatine	21
Übungsteil	22
Arbeitsblätter	
A1. Bestücken der Leiterplatte Einplatinencomputer	24
A2. Bestücken der Leiterplatte Prüfplatine	28
A3. Inbetriebnahme der Prüfplatine	32
A4. Sichtkontrolle des Einplatinencomputers	33
A5. Inbetriebnahme des Einplatinencomputers	36
A6. Entwickeln und Testen eines Prüfprogramms	52

## Einplatinencomputer

## 1. Einleitung

Unter einem Einplatinencomputer wird allgemein ein Computersystem verstanden, das auf einer einzigen Platine neben dem Mikroprozessor mit seinen erforderlichen Steuerbausteinen zusätzlich EPROM- und RAM-Speicher mit bestimmter Kapazität besitzt.

Die Kapazität der Speicherbausteine ist dabei wesentlich geringer, als die durch den Mikroprozessor maximal adressierbare Kapazität. Diese Tatsache hängt mit dem Einsatzgebieten eines Einplatinencomputers zusammen. Er soll in diesen Bereichen, fest installiert, ein Programm abarbeiten, das eine bestimmte Aufgabe löst. Diese Programme sind wesentlich kleiner als z.B. 64 KByte. Das Programm wird auf einem Entwicklungssystem geschrieben und in allen Funktionen getestet. Danach wird es in einem EPROM mit entsprechendem Speicherbereich fest abgelegt. Der Einplatinencomputer wird dann mit diesem EPROM bestückt und kann das Programm abarbeiten.

Neben den genannten Blöcken findet man sehr oft noch Ein- und Ausgabebausteine auf Einplatinencomputern. Diese E/A-Einheiten sind allerdings meistens auf spezielle Aufgaben zugeschnitten. Um die Einsatzmöglichkeiten dieses Einplatinencomputer universeller zu gestalten, wurde eine zusätzliche Ein- Ausgabekarte entwickelt, die direkt adaptierbar ist.

Dadurch lassen sich fast alle in der Praxis vorkommenden Problemstellungen mit geringem Aufwand lösen, da nur das Programm im EPROM geändert werden muß.

## Einplatinencomputer

## 1.1. Blockaufbau des Einplatinencomputers

Der Einplatinencomputer ist in seinem Aufbau an das MFA-System angelehnt.

Er wurde deshalb mit dem Mikroprozessor 8085 aufgebaut und ist daneben mit der für diesen Prozessortyp erforderlichen Adreßzwischen­speicherung, einem Datenbustransceiver und einem Decoder für die Steuersignale RD, WR und IO/M bestückt.

Zusätzlich befinden sich auf der Platine ein EPROM- und ein RAM-Baustein. Die Kapazität des RAM-Bausteins beträgt 2 KByte, für den EPROM-Bereich können wahlweise Bausteine mit einer Kapazität von 2 KByte oder 4 KByte benutzt werden. Die Unterscheidung wird durch eine Drahtbrücke auf der Platine vorgenommen. Die Pinbelegung der 64-poligen Steckerleiste ist vollkommen kompatibel zum Bussystem des MFA-Mikrocomputers.

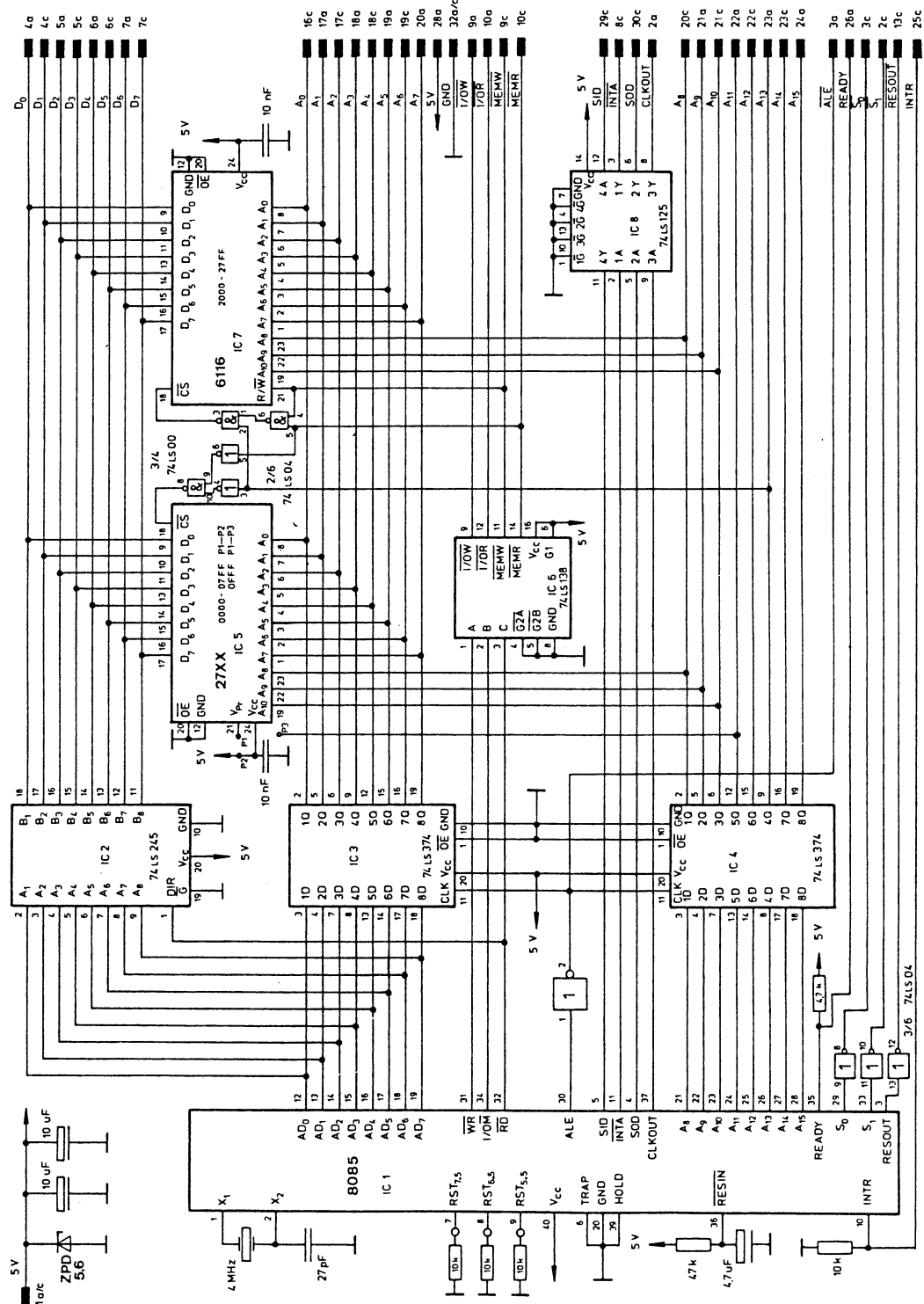
Die Z-Diode ZPD 5,6 dient als Verpolungs- und Überspannungsschutz.

Diese Übung setzt die Kenntnisse der Fachtheoretischen Übungen BFZ / MFA 10.1. - 10.4. voraus, da hier nicht auf die grundsätzliche Arbeitsweise eines Mikroprozessors eingegangen wird. Außerdem können weitergehende Informationen der Fachpraktischen Übung BFZ / MFA 2.1. entnommen werden.

## 1.2. Stromlaufplan des Einplatinencomputers

Bild 1 zeigt den Stromlaufplan des Einplatinencomputers. Dieser Stromlaufplan muß bei der Durcharbeitung der folgenden Seiten und bei der Inbetriebnahme immer benutzt werden.

Bild 1: Stromlaufplan Einplatinencomputer



## Einplatinencomputer

## 2. Zentraleinheit ( CPU )

Die Zentraleinheit des Einplatinencomputers wird durch den Mikroprozessor 8085 gebildet. Der Systemtakt dieser CPU wird durch einen 4-MHz-Quarz erzeugt und im Baustein auf 2 MHz heruntergeteilt. Am PIN 37 des Bausteins, sowie am PIN 2a der Steckerleiste, kann dieser Systemtakt gemessen werden. Dabei ist zwischen die beiden Anschlüsse der Pufferbaustein IC8 geschaltet.

Bild 2 verdeutlicht diesen Zusammenhang.

Auf den folgenden Seiten sind alle auf dem Einplatinencomputer benutzten Anschlüsse der Zentraleinheit beschrieben.

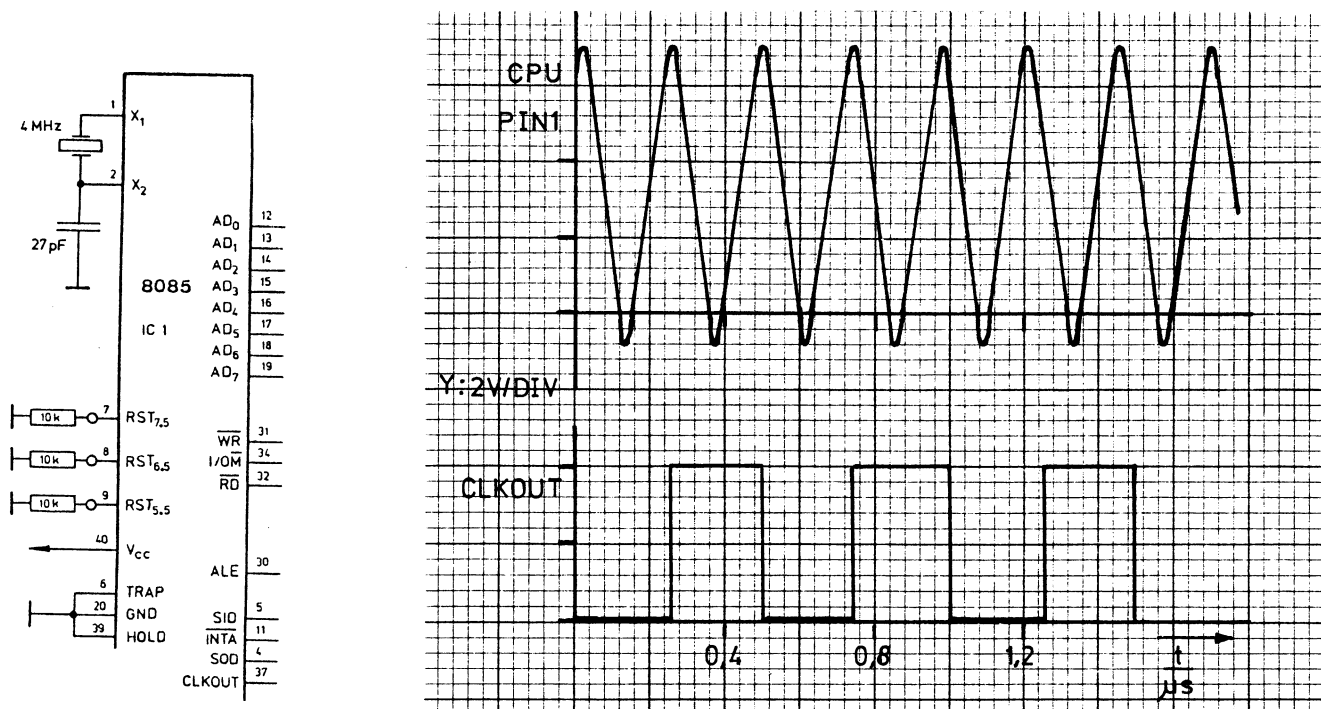


Bild 2: Systemtakterzeugung des Einplatinencomputers

## Einplatinencomputer

## 2.1. RESET-Erzeugung

Jede CPU muß nach dem Anlegen der Betriebsspannung definiert zurückgesetzt werden. Das bedeutet, der Adreßzähler wird auf die Adresse 0000H gesetzt.

Der Prozessor 8085 wird durch ein L-Signal am PIN 3 zurückgesetzt. Dies wird durch eine R-C-Kombination aus 47 kOhm und

4,7  $\mu$ F bewirkt. Der Kondensator wird über den Widerstand auf die Betriebsspannung aufgeladen. Solange die Kondensatorspannung unterhalb eines bestimmten Wertes liegt, erkennt die CPU L-Signal. Für diesen Zeitraum gibt die CPU außerdem am PIN 3 H-Signal ab. Dieses Signal kann zum Rücksetzen externer Bausteine benutzt werden. Danach beginnt die CPU mit der Abarbeitung des Programms. Im Bild 3 ist dieser Zusammenhang dargestellt.

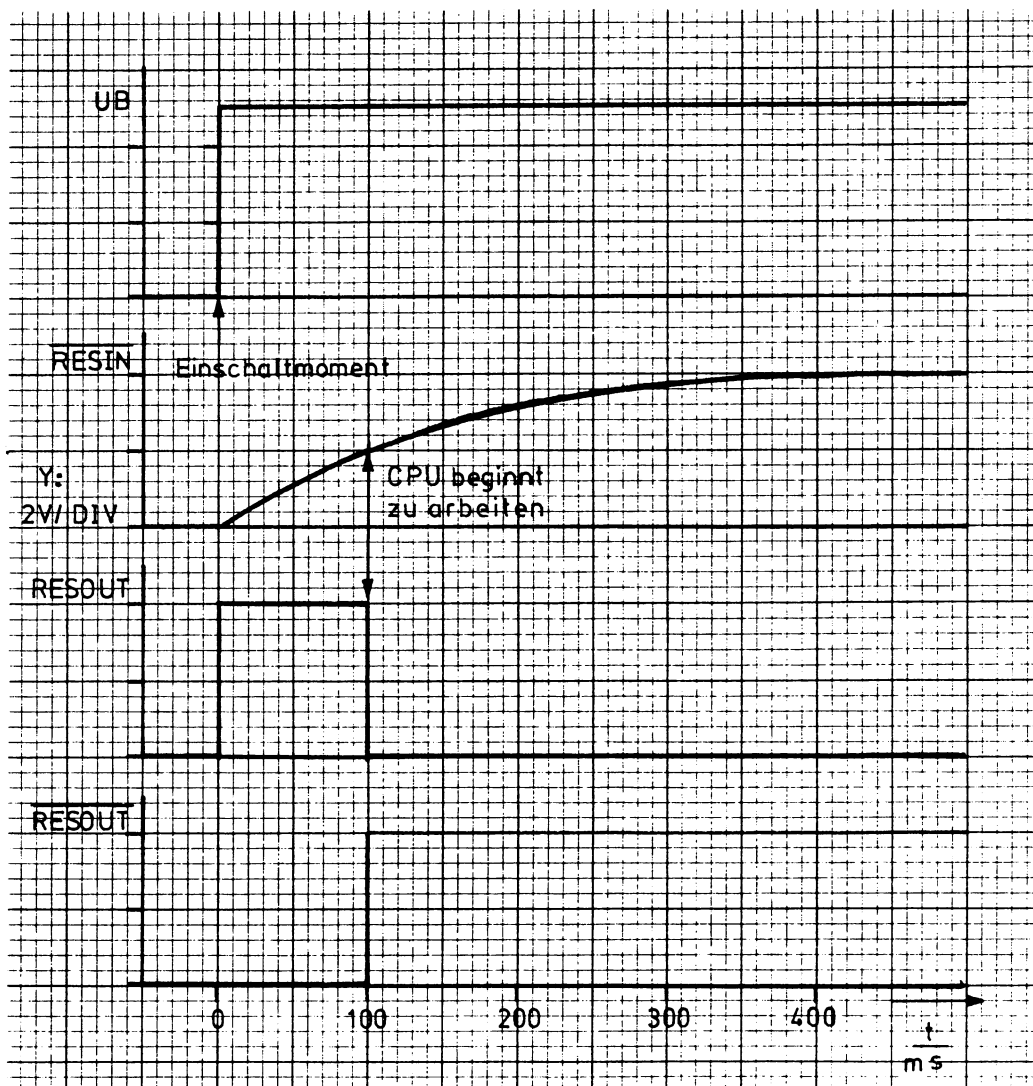


Bild 3: RESET-Signalverlauf

## Einplatinencomputer

## 2.2. Adreßsignal-Erzeugung und -Decodierung

Die CPU 8085 erzeugt insgesamt 16 Adreßsignale, die zum Adreßbus des Einplatinencomputers gesandt werden.

Die sechzehn Adreßsignale liegen an den Anschlüssen 12-19 und 21-28 an.

Dabei entsprechen die Anschlüsse 21-28 den Adreßleitungen A8-A15. An den Anschlüssen 12-19 liegen im Multiplexbetrieb die Adreßleitungen A0-A7 und die Datenleitungen D0-D7 an. Sie werden deshalb mit AD0-AD7 bezeichnet. Die Steuerung erfolgt dabei durch ein spezielles Signal, das mit "ADRESS-LATCH-ENABLE" ( ALE ) bezeichnet wird.

Ist ALE = H, so liegt das untere Adreßbyte an den Anschlüssen 12 - 19 an, ist ALE = L , so werden diese Anschlüsse als Datenleitungen benutzt.

Damit die komplette 16-Bit-Adresse stabil auf dem Adreßbus des Einplatinencomputers anliegt, werden alle 16 Adreßsignale zwischengespeichert. Dies erfolgt mit Hilfe des ALE-Signals und der beiden Bausteine IC3 und IC4. Die beiden Bausteine stellen jeweils acht D-Flip-Flops dar, die mit einer positiven Taktflanke gesteuert werden. Die Taktflanke wird aus dem invertierten ALE-Signal gewonnen.

Außerdem erfolgt durch diese beiden Bausteine eine Pufferung der o.g. CPU-Anschlüsse.

Bild 4 verdeutlicht die beschriebenen Zusammenhänge.

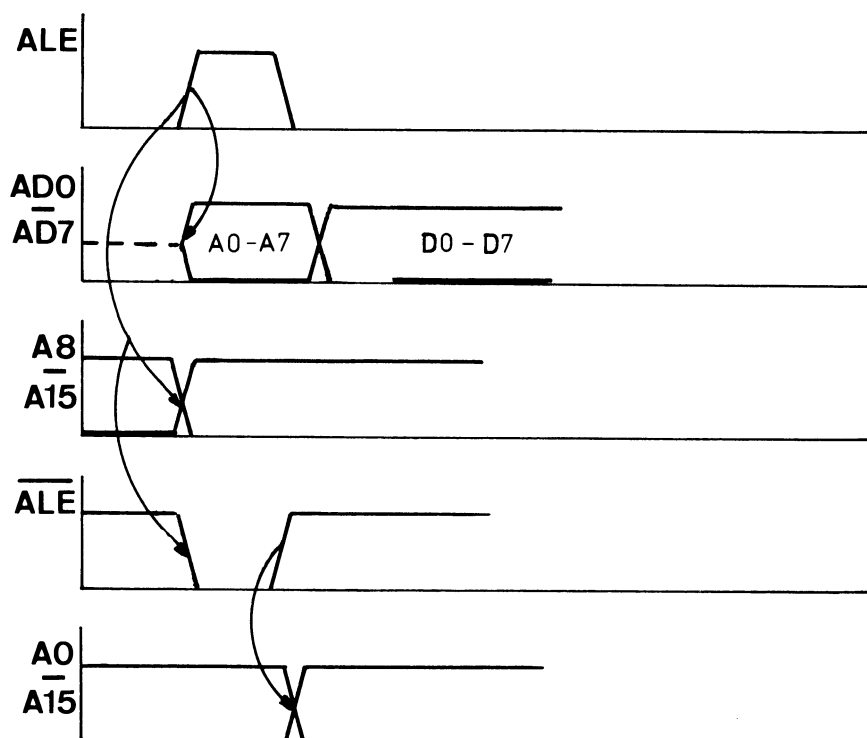


Bild 4: Adreßzwischenspeicherung

## Einplatinencomputer

## 2.3. Datensignale

Die Datenleitungen der CPU werden durch den Baustein IC2 gepuffert. Da der Datenbus aber immer bidirektional ist, muß dieser Baustein in der Datenflußrichtung umschaltbar sein. Die Umschaltung erfolgt durch einen entsprechenden Pegel am Pin 1 dieses Bausteins. Liegt L-Signal an, so kann die CPU Daten zum Datenbus des Einplatinencomputers senden. Liegt H-Signal an, so kann die CPU Daten vom Datenbus empfangen. Die Steuerung erfolgt durch das Steuersignal  $\overline{RD}$  der CPU.

Bild 5 zeigt den entsprechenden Schaltungsauszug.

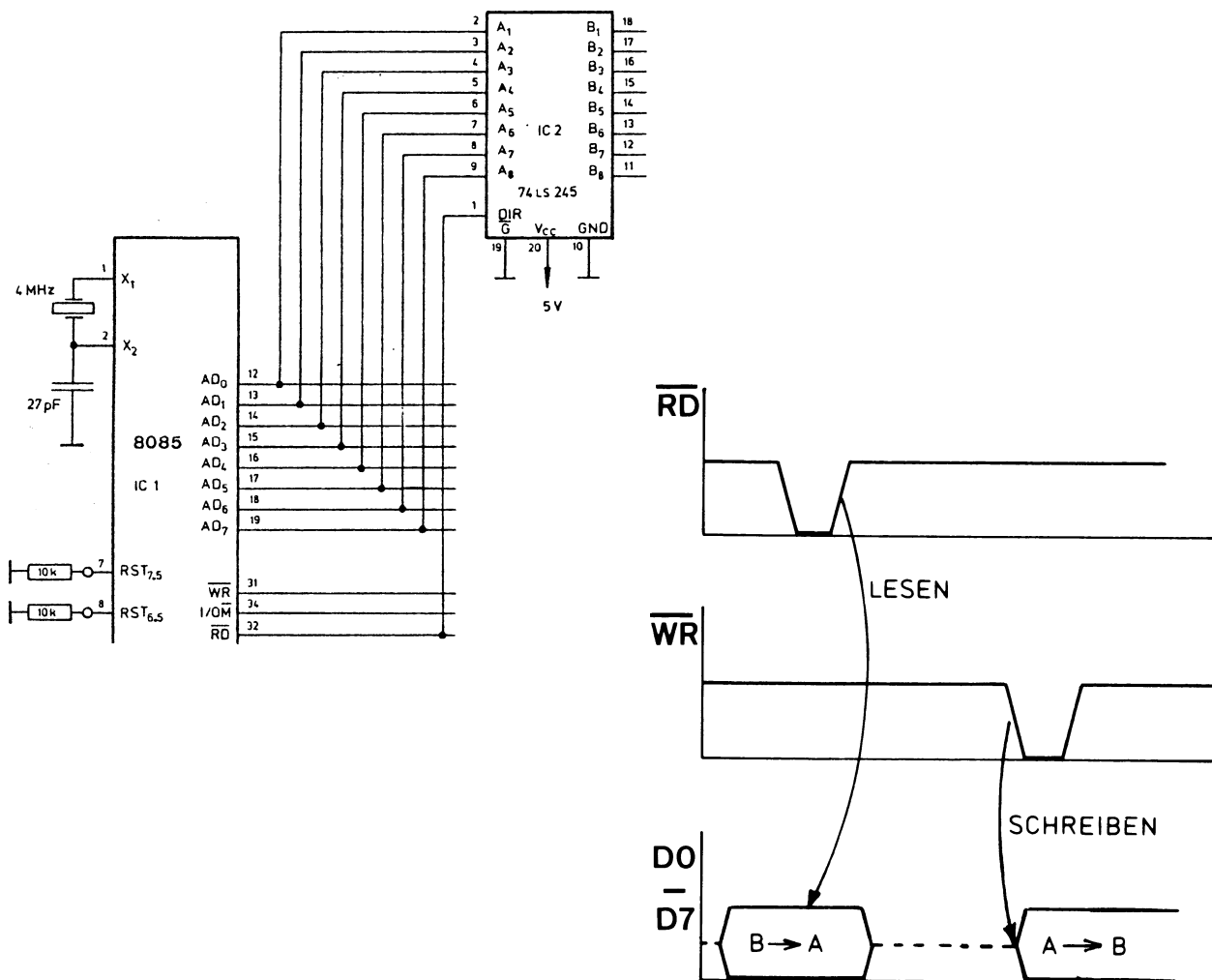


Bild 5: Datensignale des Einplatinencomputers



## 2.4. Steuersignaldecodierung

Die CPU 8085 liefert drei Steuersignale. Die Bezeichnung und Bedeutung dieser Steuersignale ist in der folgenden Tabelle 1 zusammengestellt.

		Stereusignale		
		$\overline{RD}$	$\overline{WR}$	$IO/\overline{M}$
Tabelle 1: Speicher	Lesen	L	H	L
	Schreiben	H	L	L
Ein-Ausgabe	Lesen	L	H	H
	Schreiben	H	L	H

Auf dem Steuerbus des Einplatinencomputers werden aber die vier Steuersignale  $\overline{MEMR}$ ,  $\overline{MEMW}$ ,  $\overline{IOR}$  und  $\overline{IOW}$  benötigt. Die o.g. Steuersignale der CPU müssen also decodiert werden. Diese Decodierung erfolgt durch den Baustein IC6. Es handelt sich dabei um einen 1 aus 4 - Decoder, dessen logische Funktion aus der folgenden Tabelle 2 hervorgeht. Die vier Steuersignale werden außerdem durch den Baustein IC8 gepuffert.

		$\overline{RD}$	$\overline{WR}$	$IO/\overline{M}$	$\overline{IOW}$	$\overline{IOR}$	$\overline{MEMW}$	$\overline{MEMR}$
Speicher	Lesen	L	H	L	H	H	H	L
	Schreiben	H	L	L	H	H	L	H
Ein-Ausgabe	Lesen	L	H	H	H	L	H	H
	Schreiben	H	L	H	L	H	H	H

Tabelle 2: Decodierung der CPU-Steuersignale

## Einplatinencomputer

## 2.5. Prüfkarte

Bei der Inbetriebnahme des Einplatinencomputers ist es sinnvoll, die CPU nicht dynamisch zu betreiben, sondern ein vorgegebenes Prüfprogramm in einzelnen Schritten abzuarbeiten. Dabei wird die Schrittweite nicht durch den Takt der CPU bestimmt, sondern durch die Betätigung eines externen Tasters. Zu diesem Zweck wurde eine direkt adaptierbare Prüfkarte entwickelt, die neben der Einzelschrittsteuerung der CPU die Messung der Adreßsignale, der acht Datensignale und der vier Steuersignale erlaubt.

Außerdem besteht die Möglichkeit, über einen 8-poligen Dilschalter ein festes Datenwort einzustellen, das von der CPU gelesen wird.

Bild 6 zeigt den kompletten Stromlaufplan dieser Prüfkarte.

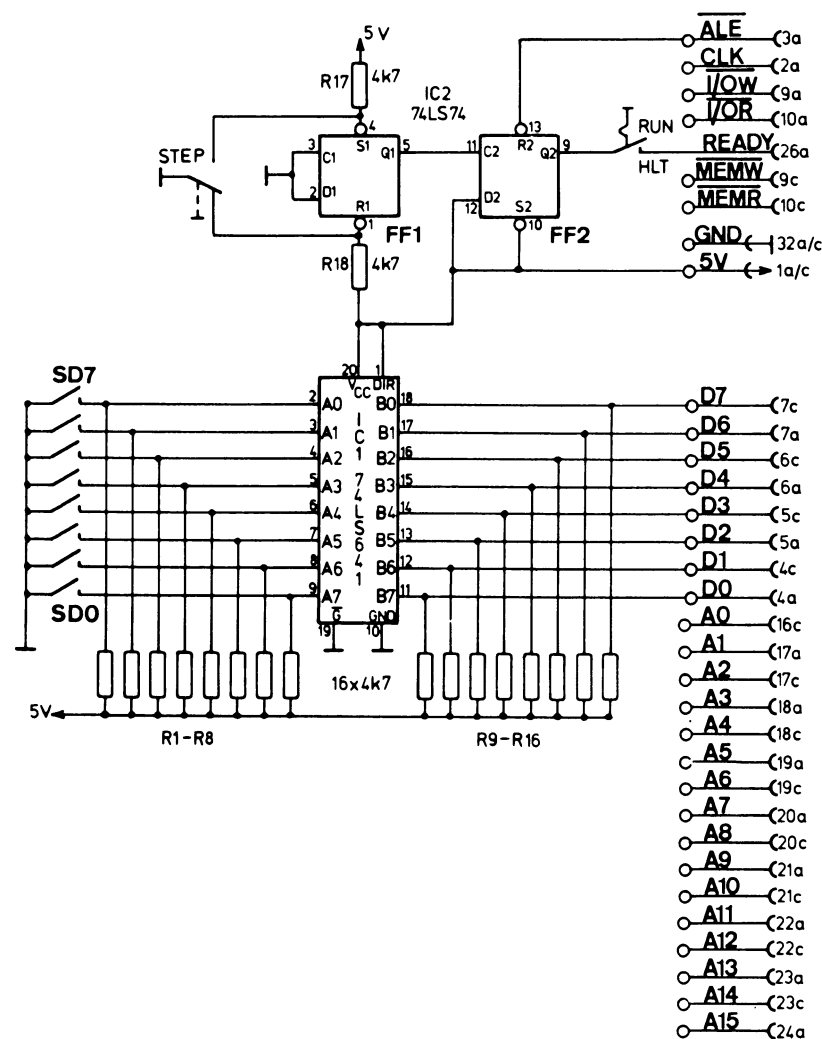


Bild 6: Stromlaufplan der Prüfkarte

## Einplatinencomputer

## 2.5.1. Einzelschrittbetrieb der CPU

Hinweis: Benutzen Sie zu der folgenden Beschreibung den Stromlaufplan von Seite 11!

Die CPU 8085 besitzt einen sog. "READY"-Anschluß, der folgende Funktion hat:

Liegt dieser Anschluß auf konstantem H-Pegel, so arbeitet die CPU ein vorgegebenes Programm dynamisch ab, d.h. die Arbeitsgeschwindigkeit wird durch den Systemtakt bestimmt.

Liegt der READY-Anschluß dagegen auf L-Pegel, so stoppt die CPU die Programmabarbeitung und wartet so lange, bis READY wieder H-Pegel führt.

Ist der Schalter RUN/HLT auf der Prüfkarte geöffnet, so erhält der READY-Eingang über den Widerstand  $R_4$  auf dem Einplatinencomputer konstanten H-Pegel.

Wird der Schalter geschlossen, so wird der READY-Eingang mit dem Q-Ausgang von FF2 verbunden. Führt jetzt das ALE-Signal einen H-L-Sprung durch, so wird FF2 zurückgesetzt und der READY-Eingang liegt auf L-Pegel. Damit wird die CPU in den Wartezustand versetzt.

Der H-L-Sprung des  $\overline{\text{ALE}}$ -Signals bedeutet aber, daß die CPU die nächst höhere Adresse ausgesandt, das unter dieser Adresse stehende Datenbyte gelesen und die entsprechende Steuerleitung aktiviert hat.

Mit einem Logiktester können jetzt die Daten-, Adreß- und Steuer-signale auf der Prüfkarte gemessen werden.

Wird der STEP-Taster danach gedrückt, so wird FF1 gesetzt, Q1 führt einen L-H-Sprung aus, FF2 wird ebenfalls gesetzt, Q2 nimmt H-Pegel an und der READY-Eingang ebenso.

Jetzt arbeitet die CPU das vorher gelesene Datenbyte ab. Nach der Abarbeitung führt  $\overline{\text{ALE}}$  wieder einen H-L-Sprung durch (die nächst höhere Adresse wird ausgesandt) und setzt FF2 zurück, wodurch der READY-Eingang wieder L-Pegel führt.

Wird der STEP-Taster losgelassen, so wird FF1 ebenfalls zurückgesetzt, und bei nochmaligem Betätigen des Tasters läuft der beschriebene Vorgang erneut ab.

## Einplatinencomputer

Im Bild 7 sind die auf der Seite 12 beschriebenen Zusammenhänge nochmals im Liniendiagramm dargestellt.

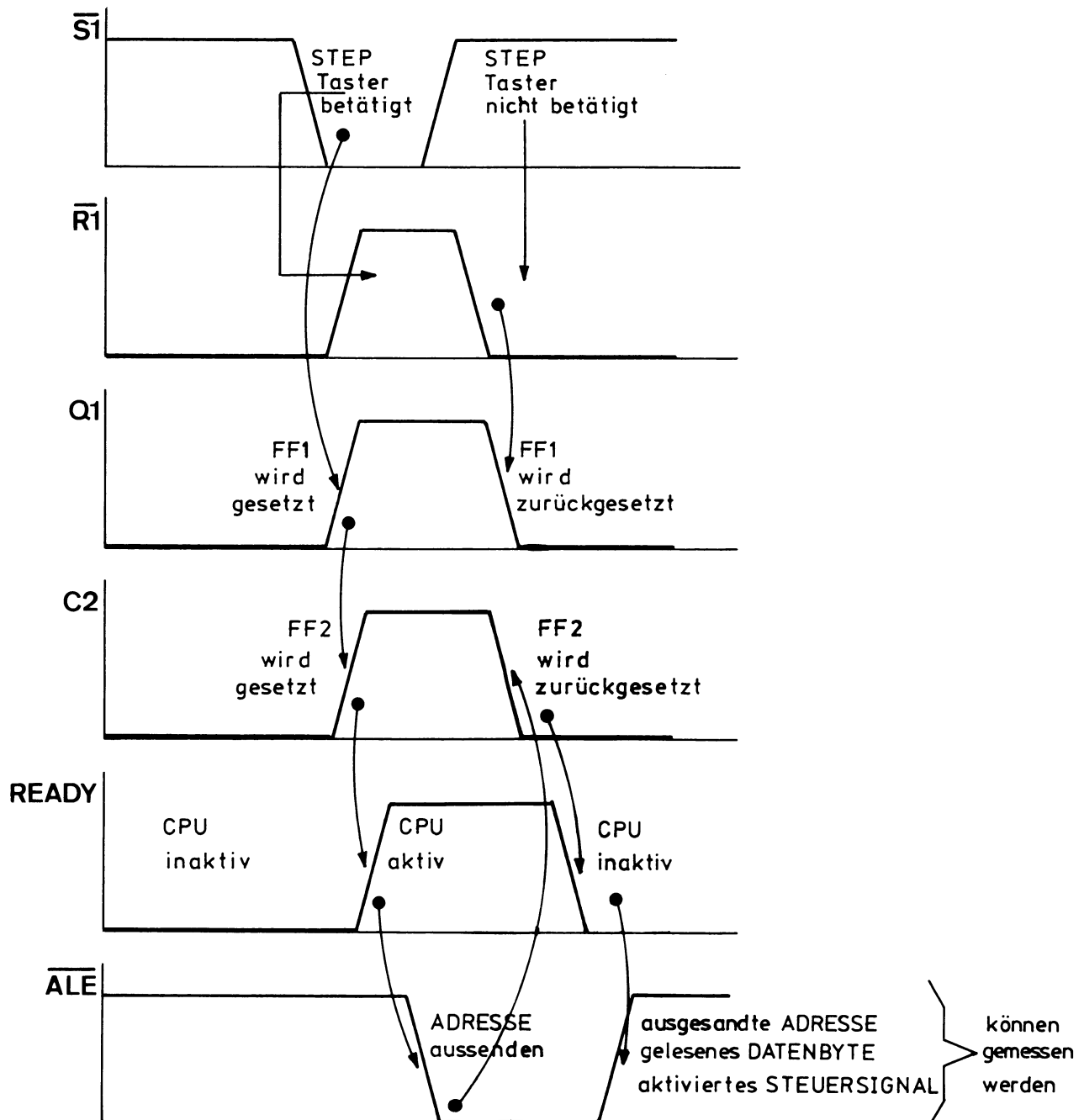


Bild 7: Einzelschrittbetrieb der CPU 8085

## Einplatinencomputer

## 2.6. Interrupteingänge der CPU 8085

Der Mikroprozessor 8085 besitzt insgesamt fünf Interrupteingänge, von denen beim Einplatinencomputer vier benutzt werden können. Grundsätzlich ist bei einem Interrupt folgendes zu beachten: Der Mikroprozessor unterbricht die laufende Programmabarbeitung und speichert die aktuelle Programmadresse ab. Danach wird die sog. Interruptroutine abgearbeitet. Ist die Abarbeitung erfolgt, fährt der Prozessor in dem vorher unterbrochenen Programm fort.

## 2.6.1. Restarteingänge RST5.5, 6.5 und 7.5

Diese drei Eingänge sind H-aktiv, wobei der Eingang RST 7.5 auf die positive Flanke eines Signals reagiert, während die Eingänge RST 6.5 und RST 5.5 zustandsgesteuert sind. Per Befehl läßt sich bestimmen, auf welchen der drei Eingänge der Mikroprozessor reagieren soll, d.h. einer oder mehrere Eingänge können ausgeblendet werden. Dieser Befehl lautet SIM (Set Interrupt Mask). Man spricht in diesem Zusammenhang auch von maskierbaren Interrupteingängen. Nach Annahme eines "Restart-Interrupts" wird durch den Mikroprozessor eine vom Hersteller festgelegte Adresse ausgesandt, unter der dann ein Sprungbefehl zur Interruptroutine stehen muß. In Tabelle 3 sind diese Adressen zusammengefaßt. Da diese drei Eingänge auf dem Einplatinencomputer offen liegen, wurden drei 10-kOhm-Widerstände als Pull-Down-Widerstände eingesetzt, damit ein ungewollter Interrupt nicht ausgelöst werden kann. Außerdem können die drei Eingänge durch den Befehl DI (Disable Interrupt) gesperrt und durch den Befehl EI (Enable Interrupt) freigegeben werden.

Tabelle 3:

RST 5.5	002C H
RST 6.5	0034 H
RST 7.5	003C H

## Einplatinencomputer

2.6.2. Interrupt-Eingang INTR und -Ausgang  $\overline{\text{INTA}}$ 

Dieser Interrupteingang wird bei einem RESET der CPU automatisch gesperrt. Außerdem besteht die Möglichkeit, diesen Eingang durch den Befehl DI (Disable Interrupt) zu sperren, bzw. durch den Befehl EI (Enable Interrupt) freizugeben. Der Eingang ist, wie die Eingänge RST 5.5 und 6.5, H-aktiv.

Im Gegensatz zu den RESTART-Eingängen muß nach Annahme dieses Interrupts vom Anwender ein CALL-Befehl vorgegeben werden, dessen Adresse auf die Interrupt-Routine weist.

Der Prozessor gibt nach Annahme des Interrupts am Ausgang  $\overline{\text{INTA}}$  ein L-Signal ab. Der INTR-Eingang wurde auf dem Einplatinencomputer ebenfalls mit einem Pull-Down-Widerstand beschaltet. Außerdem wurde der INTR-Eingang und der  $\overline{\text{INTA}}$ -Ausgang auf die Steckerleiste des Einplatinencomputers geführt. Der Ausgang ist dabei durch IC8 gepuffert.

## 2.7. Steuersignale S0 und S1

Diese Steuersignale stehen in engem Zusammenhang mit dem Signal  $\text{IO}/\overline{\text{M}}$ . Die nachfolgende Tabelle gibt Auskunft über den Zusammenhang der drei Signale.

Tabelle 4:

$\text{IO}/\overline{\text{M}}$	S1	S0	
L	L	H	Speicher schreiben
L	H	L	Speicher lesen
H	L	H	E/A schreiben
H	H	L	E/A lesen
L	H	H	Befehlsbyte lesen

## Einplatinencomputer

## 2.8. Ein- und Ausgänge zur seriellen Datenübertragung

Der Prozessor 8085 besitzt neben den acht Datenleitungsanschlüssen, die zur parallelen Datenübertragung dienen, einen Ein- und einen Ausgang zur seriellen Datenübertragung.

Die serielle Datenübertragung bietet gegenüber der parallelen den Vorteil, daß nur zwei Leitungen benötigt werden. Sie hat allerdings den Nachteil, daß sie langsamer als die parallele Datenübertragung ist, da die acht Bit nur zeitlich hintereinander übertragen werden können. Außerdem ist ein höherer Softwareaufwand erforderlich. Die serielle Datenübertragung wird auf Grund ihres Vorteils sehr häufig zur Datenfernübertragung benutzt.

Der serielle Eingang der CPU wird mit SID bezeichnet, der serielle Ausgang mit SOD.

Die beiden Anschlüsse sind über einen Pufferbaustein auf die Steckerleiste des Einplatinencomputers geführt ( IC8 ).

## 2.9. Speicherbereich des Einplatinencomputers

## 2.9.1. EPROM-Speicherbereich

Für diesen Speicherbereich kann entweder der Baustein 2716 mit 2 KByte Speicherkapazität oder der Baustein 2732 mit 4 KByte Speicherkapazität benutzt werden. Die Unterscheidung erfolgt durch eine isolierte Drahtbrücke auf der Platine.

Es gilt folgende Zuordnung:

Baustein	Drahtbrücke
2716	P1 - P2
2732	P1 - P3

Die Adreßdecodierung erfolgt durch die Bausteine 74LS00 (IC9) und 74LS04 (IC10).

Allerdings muß auf dem Einplatinencomputer nicht die gesamte 16-Bit-Adresse decodiert werden, sondern nur die Adreßleitung A13. Solange diese Adreßleitung L-Signal führt, wird das EPROM angesprochen. Bei H-Signal auf dieser Leitung wird der RAM-Baustein angesprochen. Dadurch bedingt kann sich z.B. der Adreßbereich des EPROMs entweder von 0000 - 07FF (2716) oder von 0000 - 0FFF (2732).

## Einplatinencomputer

## 2.9.2. RAM-Speicherbereich

Der RAM-Speicherbereich wird durch den Baustein 6116 mit einer Kapazität von 2 KByte realisiert. Dieser Baustein wird immer dann angesprochen, wenn die Adreßleitung A13 H-Signal führt. Der Adreßbereich kann sich also z.B. von 2000 - 27FF. Bild 8 zeigt nochmals einen entsprechenden Schaltungsauszug, und Tabelle 5 stellt die Adreßdecodierung der beiden Speicherbausteine dar.

		$\overline{\text{CS}}$ -Eingang	
Adresse	A13	EPROM	RAM
0000	L	L	H
1FFF	L	L	H
2000	H	H	L
2FFF	H	H	L

Tabelle 5: EPROM- und RAM-Adressierung

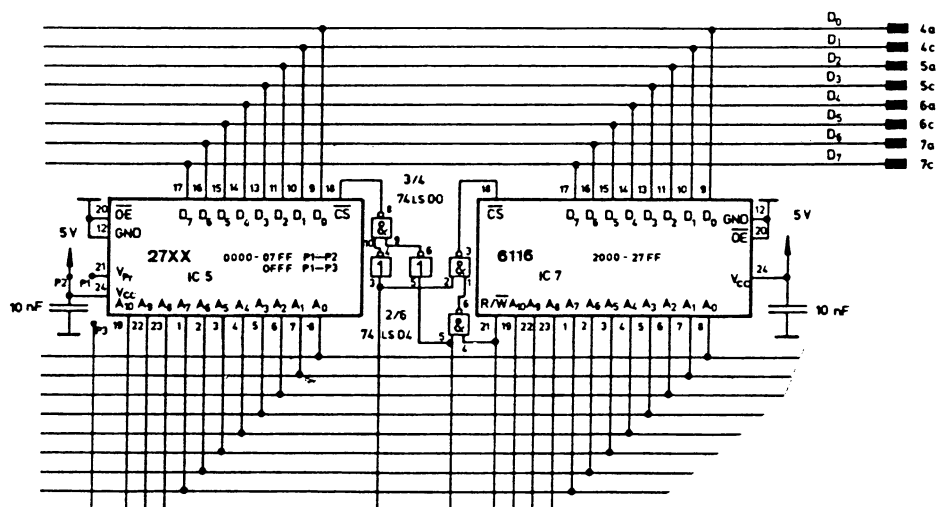


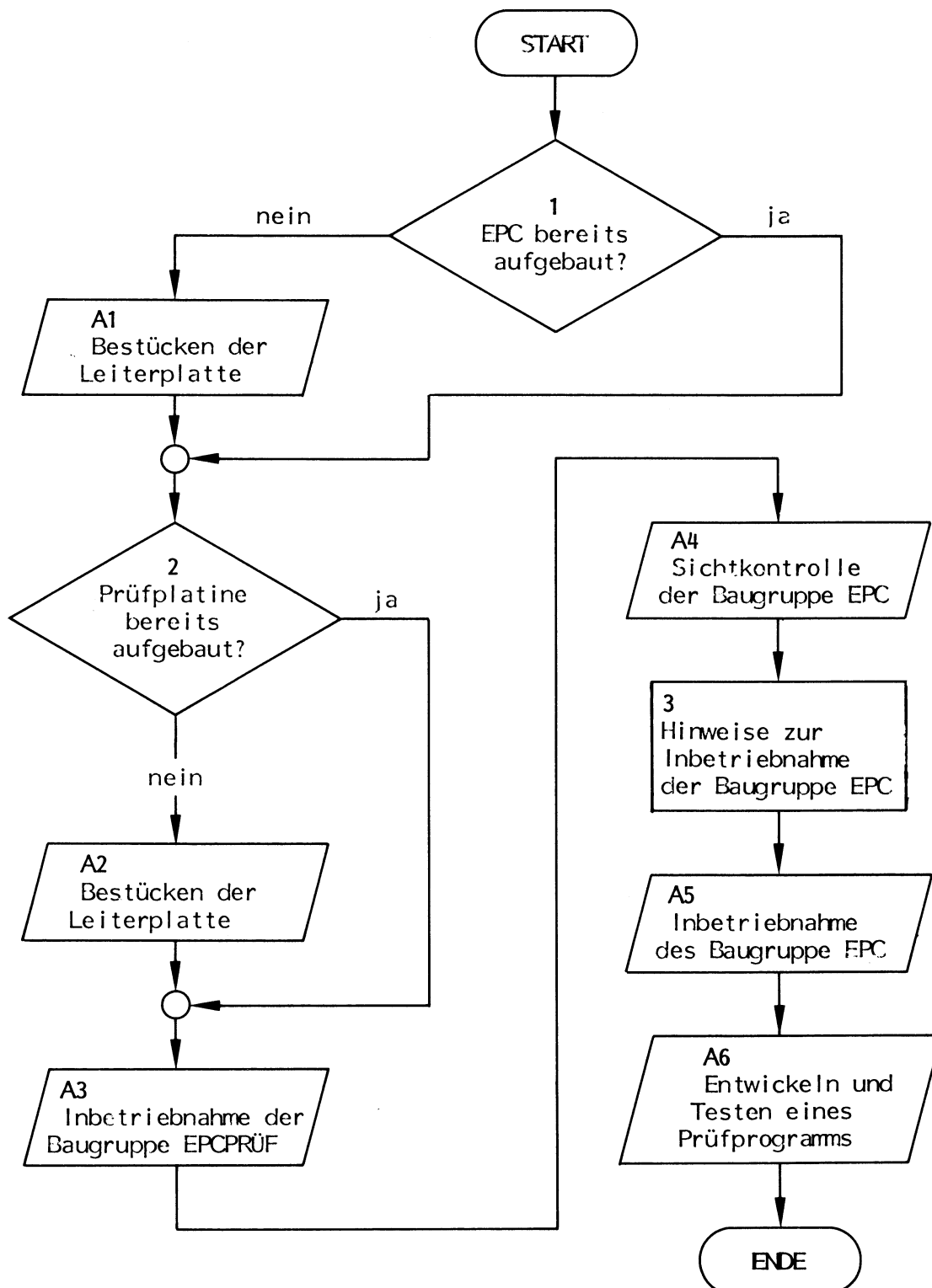
Bild 8: Speicherbereich des Einplatinencomputers



## Einplatinencomputer

## Flußdiagramm für den Arbeitsablauf

Hinweis: Auf den folgenden Seiten wird der Einplatinencomputer kurz als EPC bezeichnet!



## Einplatinencomputer

Stckz.	Benennung/Daten	Bemerkung
1	fertig geätzte, durchkontaktierte Epoxid-Glashartgewebeplatte mit der Bezeichnung EPC	
1	Messerleiste 64polig, DIN 41 612 mit abgewinkelten Anschlüssen für Platinenmontage	z.B. Erni STV-P-364a/c Nr.9722.333.401
2	Zylinderschraube M2,5x10 DIN 84	
2	Federscheibe A2,7 DIN 137	
2	Sechskantmutter M2,5 DIN 439	
1	Widerstand 4,7 kOhm/0,25 +/- 5%	
4	Widerstand 10 kOhm/0,25W +/- 5%	
1	Widerstand 47 kOhm/0,25W +/- 5%	
1	Ker. Scheibenkondensator 27 pF/63 V	RM ca. 2,5 mm
2	Ker. Scheibenkondensator 10 nF/63 V	RM ca. 2,5 mm
1	Tantal-Elko 4,7 uF/16 V	Tropfenform
2	Tantal-Elko 10 uF/16 V	Tropfenform
1	Z-Diode ZPD 5,6	
1	Quarz 4 MHz	HC 18 W
1	IC CPU 8085	
1	IC 2716 od. 2732, EPROM	
1	IC 6116, RAM	
1	IC 74LS125	
1	IC 74LS245	
2	IC 74LS374	
1	IC 74LS138	
1	IC 74LS04	
1	IC 74LS00	

## Einplatinencomputer

Stckz.	Benennung/Daten	Bemerkung
1	IC-Fassung 40polig, DIL	
2	IC-Fassung 24polig, DIL	
3	IC-Fassung 20polig, DIL	
1	IC-Fassung 16polig, DIL	
3	IC-Fassung 14polig, DIL	
5	Stecklötösen	für Bohrung 1,3 mm
1	Silikon-Isolierscheibe ca. 10x12 mm	zur Isolierung des Quarzes

## Einplatinencomputer

Für die Prüfkarte werden folgende Bauelemente benötigt:

Stckz.	Benennung/Daten	Bemerkung
1	fertig geätzte Epoxid-Glashartgewebeplatte mit der Bezeichnung <b>EPCPRÜF</b>	
1	Federleiste 64polig, DIN 41612 mit abgewinkelten Anschlüssen für Platinenmontage	z.B. Panduit Nr. 100-964-553
2	Zylinderschraube M2,5x10 DIN 84	
2	Federscheibe A2,7 DIN 137	
2	Sechskantmutter B2,5 DIN 127	
1	Einzel­taster 2polig um für Platinenmontage	Typ Digitast RM 5,08 mm
1	DIL Schiebeschalter 8polig	RM 2,56 mm
1	DIL Schiebeschalter 1polig	RM 2,56 mm
18	Widerstand 4,7 kOhm	0,25 W
1	IC 74LS641	
1	IC 74LS74	
1	IC-Fassung 20polig	
1	IC-Fassung 14polig	
32	Stecklötösen	für Bohrung 1,3 mm

## Einplatinencomputer

1

In dieser Übung werden Sie den Einplatinencomputer aufbauen und in Betrieb nehmen. Falls Sie bereits eine fertig aufgebaute Baugruppe erhalten haben, besteht Ihre Aufgabe darin, sie zu prüfen und in Betrieb zu nehmen.

Entscheiden Sie nun, wie Sie vorgehen:

Aufbau der Platine nach Arbeitsunterlagen ---> A1

Aufbau der Prüfplatine ---> 2

---

2

In dieser Übung werden Sie die Prüfplatine zum Einplatinencomputer bestücken und in Betrieb nehmen. Falls Sie bereits eine fertig aufgebaute Baugruppe erhalten haben, besteht Ihre Aufgabe darin, sie zu prüfen und in Betrieb zu nehmen.

Entscheiden Sie nun, wie Sie vorgehen:

Aufbau der Prüfplatine nach Arbeitsunterlagen ---> A2

Inbetriebnahme der fertigen Baugruppe ---> A3

---

3

Mit Hilfe der folgenden Seiten werden Sie den Einplatinencomputer in Betrieb nehmen.

Dazu benötigen Sie:

- 1 Prüfplatine  
fertig aufgebaut und in Betrieb genommen
- 1 Netzteil 5 V/500 mA
- 1 Zweikanal-Oszilloskop 15 MHz
- 1 Logiktester
- 1 Digital- oder Analog-Vielfachinstrument
- 2 Tastköpfe mit einem Teilerverhältnis von 10:1
- 1 40poligen Meßadapter für Messungen an der CPU

## Einplatinencomputer

## 3.1

Die Inbetriebnahme ist folgendermaßen aufgeteilt:

- statische Funktion des EPC

hier wird die CPU im Einzelschritt betrieben. Der Schalter RUN/HLT auf der Prüfplatine muß dabei auf HLT stehen.

- dynamische Funktion der EPC

die CPU arbeitet jetzt mit der durch den Systemtakt vorgegebenen Geschwindigkeit. Der Schalter RUN/HLT der Prüfplatine muß dabei auf RUN stehen.

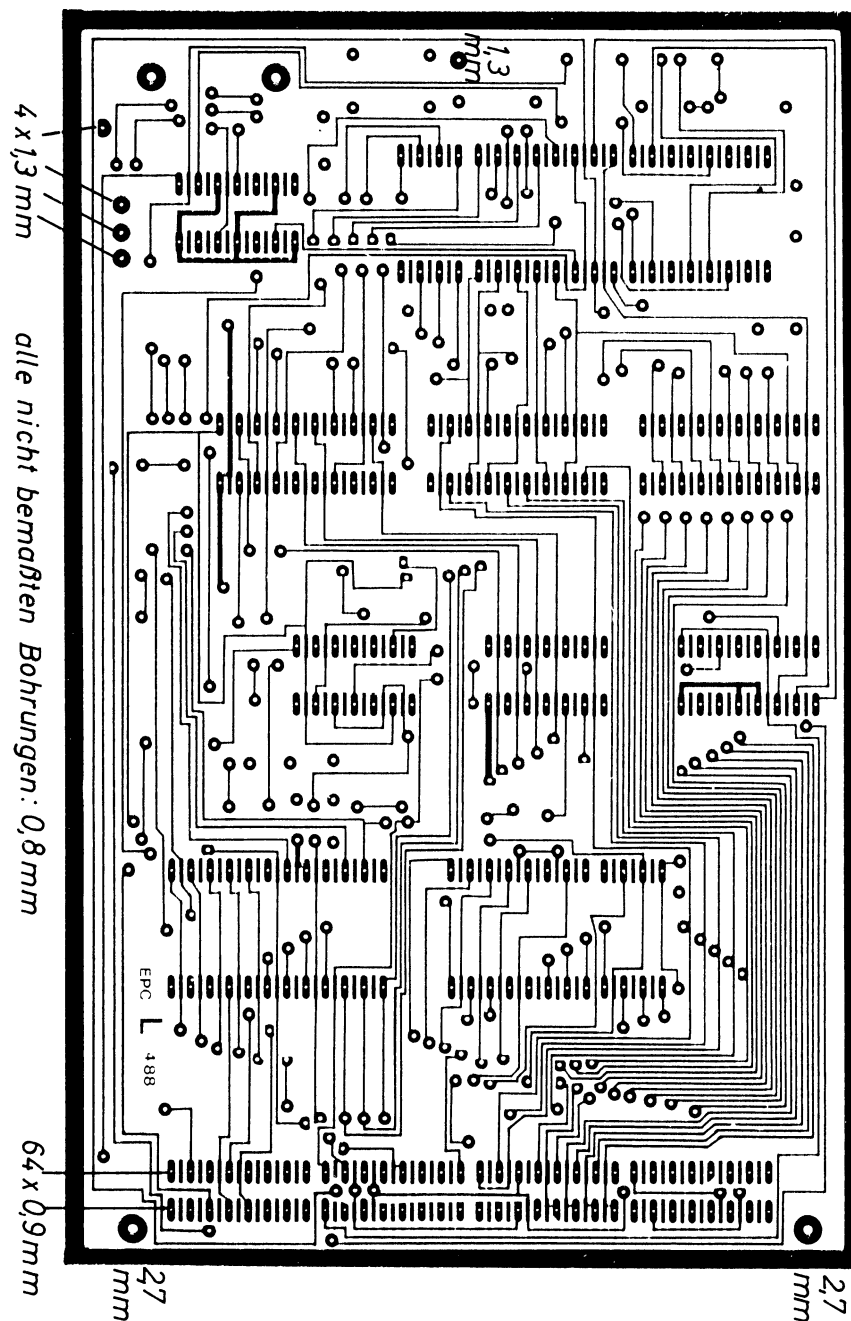
---> A5

## Einplatinencomputer

## A1.1

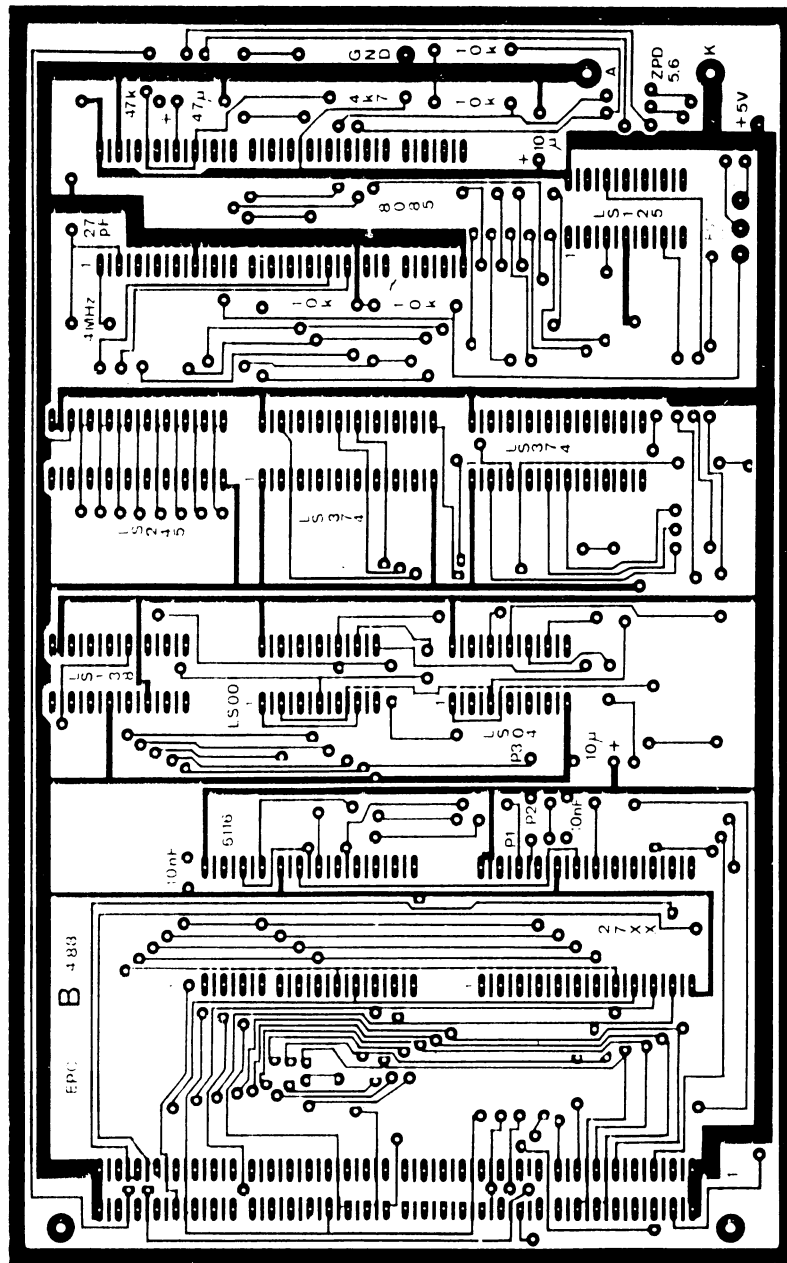
Für den Einplatinencomputer benötigen Sie eine zweiseitig-kupferkaschierte und durchkontaktierte Leiterplatte.

Leiterbahnseite der Platine



A1.2

Die folgende Abbildung zeigt das Layout der Bestückungsseite



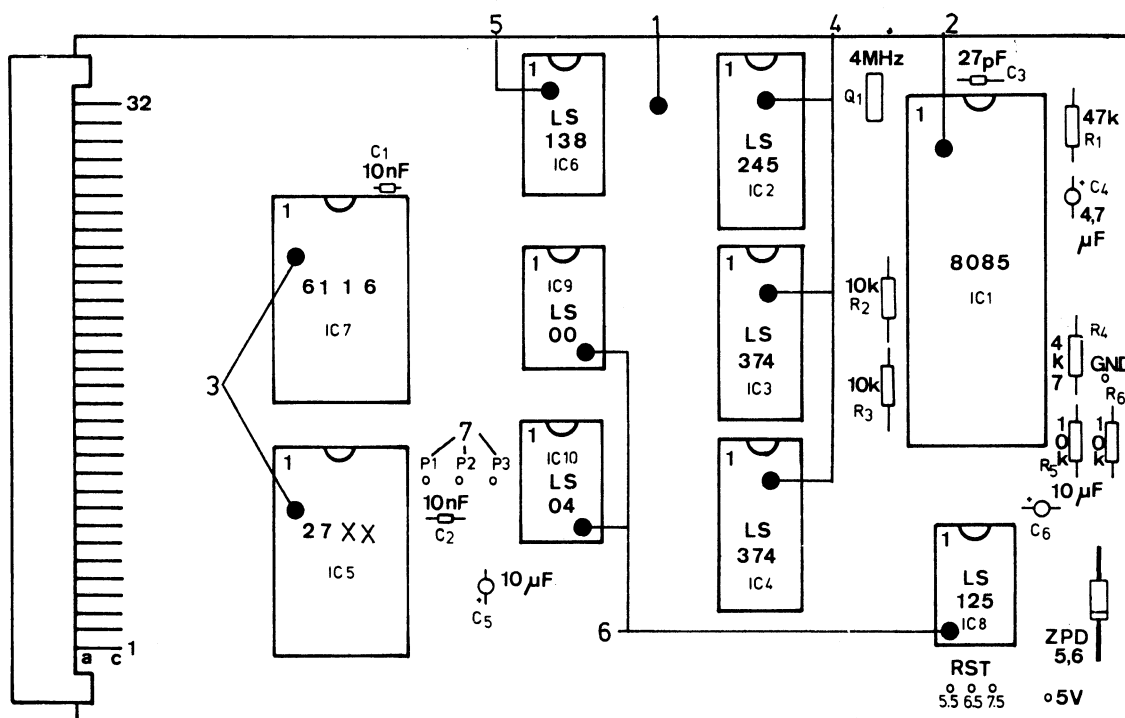


## Einplatinencomputer

## A1.3

Bestücken Sie die Leiterplatte mit Hilfe des Bestückungsplans, der Stückliste und der Bauteilliste.  
Vorher sollten Sie die Leiterplatte mit einer Lupe nach Rissen und Kurzschlüssen untersuchen und Fehler beseitigen.

## BESTÜCKUNGSPLAN Einplatinencomputer



---&gt;

## Einplatinencomputer

A1.4

## Stückliste Einplatinencomputer

Pos.	Stckz.	Benennung/Daten	Bemerkung
1	1	Leiterplatte EPC	
2	1	IC-Fassung 40polig	
3	2	IC-Fassung 24polig	
4	3	IC-Fassung 20polig	
5	1	IC-Fassung 16polig	
6	3	IC-Fassung 14polig	
7	1	Lötbrücke aus isol. Schaltdraht 0,5 mm Cu-Ag	2716: P1 - P2 2732: P1 - P3

## Bauteilliste Einplatinencomputer

Kennz.	Benennung/Daten	Bemerkung
R1	Widerstand 47 kOhm/0,25 W	
R2-R3	Widerstand 10 kOhm/0,25 W	
R4	Widerstand 4,7 kOhm/0,25 W	
R5-R6	Widerstand 10 kOhm/0,25 W	
C1-C2	ker. Scheibenkondensator 10 nF	
C3	ker. Scheibenkondensator 27 pF	
C4	Tantal-Elko 4,7 uF	
C5-C6	Tantal-Elko 10 uF	
D1	Z-Diode ZPD 5,6	
Q1	Quarz 4 MHz	
IC1	CPU 8085	
IC2	74LS245	IC's nicht
IC5	EPROM 2716 od. 2732	eingesteckt
IC6	74LS138	

---&gt;

## Einplatinencomputer

A1.5

## Bauteilliste Einplatinencomputer

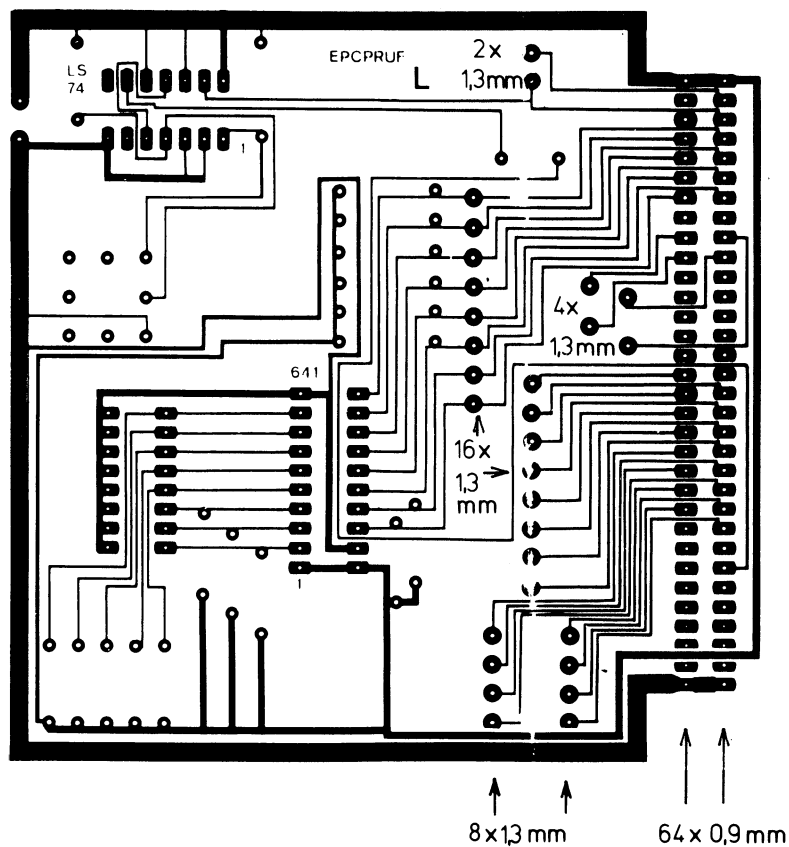
Kennz.	Benennung/Daten	Bemerkung
IC3-IC4	74LS374	
IC7	RAM 6116	
IC8	74LS125	IC's nicht eingesteckt
IC9	74LS00	
IC10	74LS04	

---&gt; 2

A2.1

Für die Prüfkarte benötigen Sie eine einseitig-kupferkaschierte Leiterplatte.

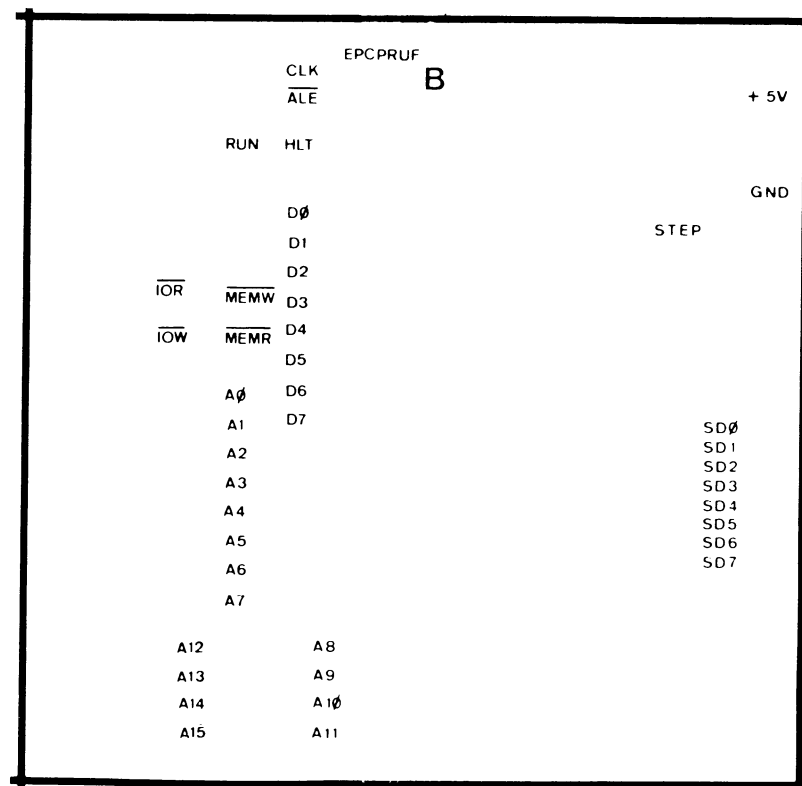
## Leiterbahnseite der Prüfplatine



---&gt;

## A2.2

## Bestückungsseite der Prüfkarte



Bestücken Sie die Leiterplatte mit Hilfe des Bestückungsplans, der Stückliste und der Bauteilliste. Vorher sollten Sie die Leiterplatte mit einer Lupe nach Rissen und Kurzschlüssen untersuchen und Fehler beseitigen.

---

## Einplatinencomputer

A2.4

## Stückliste Prüfkarte

Pos.	Stckz.	Benennung/Daten	Bemerkung
1	1	Leiterplatte EPCPRÜF	
2	1	IC-Fassung 20polig	
3	1	IC-Fassung 14polig	
4	1	DIL-Schiebeschalter 8polig	
5	1	DIL-Schiebeschalter 1polig	Einbau wie auf dem Bestückungsplan
6	1	Digitaster 2polig-um	
7	32	Stecklötösen	

## Bauteilliste Prüfkarte

Kennz.	Benennung/Daten	Bemerkung
R1-R18	Widerstand 4,7 kOhm/0,25 W	
IC1	74LS641	IC's nicht eingesteckt
IC2	74LS74	

---&gt; A3

## Einplatinencomputer

## A3.1

## Sichtkontrolle

Für die Sichtkontrolle sowie die Inbetriebnahme der Prüfkarte muß der Stromlauf- und Bestückungsplan bereitliegen. Kontrollieren Sie alle Lötstellen auf der Leiterbahnseite und achten Sie auf Kurzschlüsse, die beim Löten eventuell zwischen den Leiterbahnen entstanden sein könnten.

Achten Sie bei der Kontrolle der Bauteile auf folgende Punkte:

- Sind die beiden DIL-Schiebeschalter richtig eingebaut?
- Ist der Digitaster richtig eingebaut?
- Die IC's dürfen noch nicht gesteckt sein

Auf den folgenden Seiten werden Sie die Prüfkarte in Betrieb nehmen. Sie benötigen dazu:

- 1 5V-Netzteil
- 1 Digital- oder Analog-Vielfachinstrument
- 1 Logiktester

Messen Sie an den entsprechenden Anschlüssen der nichtbestückten IC-Sockel die Versorgungsspannung.

Dazu schließen Sie ein 5V-Netzteil an die Stecklötösen +5V und GND an. Suchen Sie sich aus dem Stromlaufplan die entsprechenden Anschlüsse heraus und tragen Sie Ihre Meßergebnisse in die folgende Tabelle ein.

	IC1	IC2
Positive Versorgungsspannung an PIN Nr.		
Masse an PIN Nr.		
$U_B/V$		
GND		

--->

## Einplatinencomputer

## A3.2

Wenn die Meßergebnisse korrekt sind, können Sie die Betriebsspannung ausschalten, die IC's in die Fassungen stecken (Achten Sie auf den richtigen Einbau) und die Betriebsspannung wieder einschalten.

Bei dem Baustein 74LS641 (IC1) handelt es sich um einen bidirektionalen Treiberbaustein mit Open-Collector-Ausgängen. Durch die feste Beschaltung des DIR-Steuereingangs mit +5V ist der Baustein so voreingestellt, daß er nur Daten von den Anschlüssen A0-A7 zu den Anschlüssen B0-B7 senden kann. Die Signale werden dabei durch den 8-poligen DIL-Schiebeschalter vorgegeben.

Überprüfen Sie jetzt mit Hilfe eines Logiktesters die Funktion des Bausteins und tragen Sie Ihre Meßergebnisse in die folgende Tabelle ein. Überlegen Sie dabei, welche Pegel bei der entsprechenden Schalterstellung des 8-poligen Schiebeschalters am IC zu messen sein müssen.

		IC1 PIN															
		2	3	4	5	6	7	8	9	18	17	16	15	14	13	12	11
SD7-SD0	01010101																
SD7-SD0	10101010																



## A3.3

Der Baustein 74LS74 (IC2) enthält zwei taktzustandsgesteuerte D-FLIP-FLOPS mit L-aktiven, statischen Setz- und Rücksetzeingängen. Das FLIP-FLOP 1 soll den Step-Taster entprellen, daß FLIP-FLOP 2 steuert den READY-Eingang der CPU 8085 an.

Bevor Sie diesen Übungsteil durchführen, machen Sie sich die Arbeitsweise des Stepbetriebs mit Hilfe von Seite 12 noch einmal klar.

Arbeitsweise von FF1:

Ist der Step-Taster nicht betätigt, so ist das FLIP-FLOP zurückgesetzt und Q1 muß L-Pegel führen. Bei Betätigung des Step-Tasters wird FF1 gesetzt und Q1 muß H-Pegel führen.

Arbeitsweise von FF2:

Nimmt der Clock-Eingang C2 H-Pegel an (gesteuert durch Q1), so kann der Ausgang Q2 nur dann H-Pegel annehmen, wenn S2 und R2 ebenfalls H-Pegel führen. Führt der Eingang R2 dagegen L-Pegel, so wird FF2 zurückgesetzt und Q2 nimmt L-Pegel an.

Überprüfen Sie die Funktion von FF1 und FF2 mit Hilfe eines Logiktesters und tragen Sie Ihre Meßergebnisse in die folgende Tabelle ein. Der angegebene Pegel des  $\overline{\text{ALE}}$ -Signals kann durch eine Leitung zwischen +5 Volt bzw. GND und dem  $\overline{\text{ALE}}$ -Meßpunkt auf der Prüfkarte hergestellt werden.

		IC2					
		$\overline{\text{R1}}$	$\overline{\text{S1}}$	Q1	C2	$\overline{\text{R2}}$	Q2
$\overline{\text{ALE}}=\text{H}$	Step-Taster nicht betätigt						
	Step-Taster gedrückt						
$\overline{\text{ALE}}=\text{L}$	Step-Taster nicht betätigt						
	Step-Taster gedrückt						

A3.4

## ERGEBNISSE DER MESSUNGEN VON SEITE 34

		$\overline{R1}$	$\overline{S1}$	Q1	C2	$\overline{R2}$	Q2
$\overline{ALE}=H$	Step-Taster nicht betätigt	L	H	L	L	H	H
	Step-Taster betätigt	H	L	H	H	H	H
$\overline{ALE}=L$	Step-Taster nicht betätigt	L	H	L	L	L	L
	Step-Taster betätigt	H	L	H	H	L	L

Damit ist die Inbetriebnahme der Prüfkarte beendet!

---> A4

A4.1

## Sichtkontrolle

Für die Sichtkontrolle sowie die Inbetriebnahme des EPCs muß der Stromlauf- und Bestückungsplan bereitliegen.

Kontrollieren Sie alle Lötstellen auf der Leiterbahnseite und achten Sie auf Kurzschlüsse, die beim Löten eventuell zwischen den Leiterbahnen entstanden sein könnten.

Achten Sie bei der Kontrolle der Bauteile auf folgende Punkte:

- Sind alle Widerstände mit Ihrem richtigen Wert eingebaut?
- Sind alle Elkos richtig gepolt?
- Die IC's dürfen nicht gesteckt sein
- Die Brücke P1-P2 oder P1-P3 muß vorhanden sein
- Die Isolierscheibe muß sich auf der Bestückungsseite unter dem Quarz befinden

---> 3

## Einplatinencomputer

## A5.1

Adaptieren Sie die fertige Prüfkarte an den Einplatinencomputer und schließen Sie die Versorgungsspannung von 5V über die Prüfkarte an. Messen Sie die Spannungsversorgung an allen IC-Sockeln. Suchen Sie sich aus dem Stromlaufplan die entsprechenden Anschlüsse heraus und tragen Sie Ihre Meßergebnisse in die folgende Tabelle ein.

	IC1	IC2	IC3	IC4	IC5	IC6	IC7	IC8	IC9	IC10
TYP										
positive Versorgungsspannung an PIN										
Masse an PIN										
$U_B$										
GND										

Nachdem Sie die Betriebsspannung an allen IC-Sockeln überprüft haben, werden die Daten- und Steuerleitungen auf Kurzschlüsse und Unterbrechungen untersucht. Da bei der Vielzahl der Leitungen und der sich daraus ergebenden Dichte auf der Platine, eine Sichtkontrolle nur schwer möglich ist, werden sie elektrisch überprüft.

--->

## Einplatinencomputer

## A5.2

## Überprüfung der Datenleitungen

Stellen Sie mit Hilfe der Schalter SD7-SD0 auf der Prüfkarte das Datenwort 01010101 ein und überprüfen Sie, ob die gleichen Pegel an den entsprechenden Anschlüssen der beiden Speicherbausteine sowie an den B-Anschlüssen von IC2 des EPCs anliegen. Danach stellen Sie das Datenwort 10101010 ein und überprüfen wiederum die Pegel an den drei IC's. Tragen Sie Ihre Meßergebnisse in die Tabelle auf der folgenden Seite ein, überprüfen Sie die Richtigkeit und beseitigen Sie eventuell auftretende Fehler.

eingestelltes Datenwort: 01010101

EPROM								RAM								74LS245							
D0	D1	D2	D3	D4	D5	D6	D7	D0	D1	D2	D3	D4	D5	D6	D7	B1	B2	B3	B4	B5	B6	B7	B8

eingestelltes Datenwort: 10101010

EPROM								RAM								74LS245							
D0	D1	D2	D3	D4	D5	D6	D7	D0	D1	D2	D3	D4	D5	D6	D7	B1	B2	B3	B4	B5	B6	B7	B8

**ACHTUNG: BEI DEN NÄCHSTEN ARBEITSSCHRITTEN MUß DIE BETRIEBSSPANNUNG ABGESCHALTET WERDEN !!!**

Überprüfen Sie jetzt mit einem Ohmmeter, ob die Leitungen zwischen den A-Anschlüssen von IC2 und dem CPU-Sockel keine Unterbrechungen bzw. Kurzschlüsse aufweisen. Niederohmige Verbindungen kennzeichnen Sie in der folgenden Tabelle mit "l", hochohmige mit "h".

## Kontrolle der Adreßleitungen

## Überprüfung auf Unterbrechungen

IC2 (PIN)	2	3	4	5	6	7	8	9
CPU (PIN)	12	13	14	15	16	17	18	19
Meßergebnis								

## Überprüfung auf Kurzschluß zwischen den A-Anschlüssen von IC2

IC2 (PINs)	2-3	3-4	4-5	5-6	6-7	7-8	8-9
Meßergebnis							

--->

## A5.3

Die Steuerleitungen werden ebenfalls mit einem Ohmmeter auf Unterbrechungen und Kurzschlüsse untersucht. Niederohmige Verbindungen werden mit "l", hochohmige mit "h" gekennzeichnet.

## Überprüfung auf Unterbrechungen

Dazu wird das Ohmmeter auf der einen Seite an die Stecklötösen der Prüfkarte angeschlossen, auf der anderen Seite an die entsprechenden Anschlüsse von IC6.

IC6 (EPC)	PIN 9	PIN 12	PIN 11	PIN 14
Prüfkarte	IOW	IOR	MEMW	MEMR
Meßergebnis				

## Überprüfung auf Kurzschluß

Es wird der Widerstand zwischen jeweils zwei Leitungen auf der adaptierten Prüfkarte gemessen

IOW-IOR	IOR-MEMW	MEMW-MEMR

Auf die gleiche Weise werden die Verbindungen zwischen IC6 und den CPU-Anschlüssen überprüft.

## Untersuchung auf Unterbrechungen

IC6 (PIN)	1	2	3
CPU (PIN)	31	34	32
Meßergebnis			

## Untersuchung auf Kurzschluß an den Anschlüssen von IC6

IC6 (PINs)	1-2	2-3
Meßergebnis		

## Einplatinencomputer

## A5.4

Wenn die Meßergebnisse der letzten beiden Seiten keine Fehler aufwiesen, kann der Einplatinencomputer jetzt mit der CPU, dem Datenbustransceiver IC2, den Adreßzwischen Speichern IC3 und IC4, dem Steuersignaldecoder IC6 sowie dem Pufferbaustein IC8 bestückt werden. Die beiden Speicherbausteine werden noch nicht in die Sockel gesteckt.

Nach Einschalten der Betriebsspannung können jetzt die Adreßleitungen auf Kurzschlüsse und Unterbrechungen untersucht werden. Danach wird die Arbeitsweise der CPU in Verbindung mit den anderen Bausteinen überprüft.

Zuerst muß aber die Quarzfrequenz und die Taktfrequenz der CPU gemessen werden. Machen Sie sich mit Hilfe von Seite 6 den entsprechenden Sachverhalt noch einmal klar. Oszilloskopieren und zeichnen Sie dann folgende Signale:

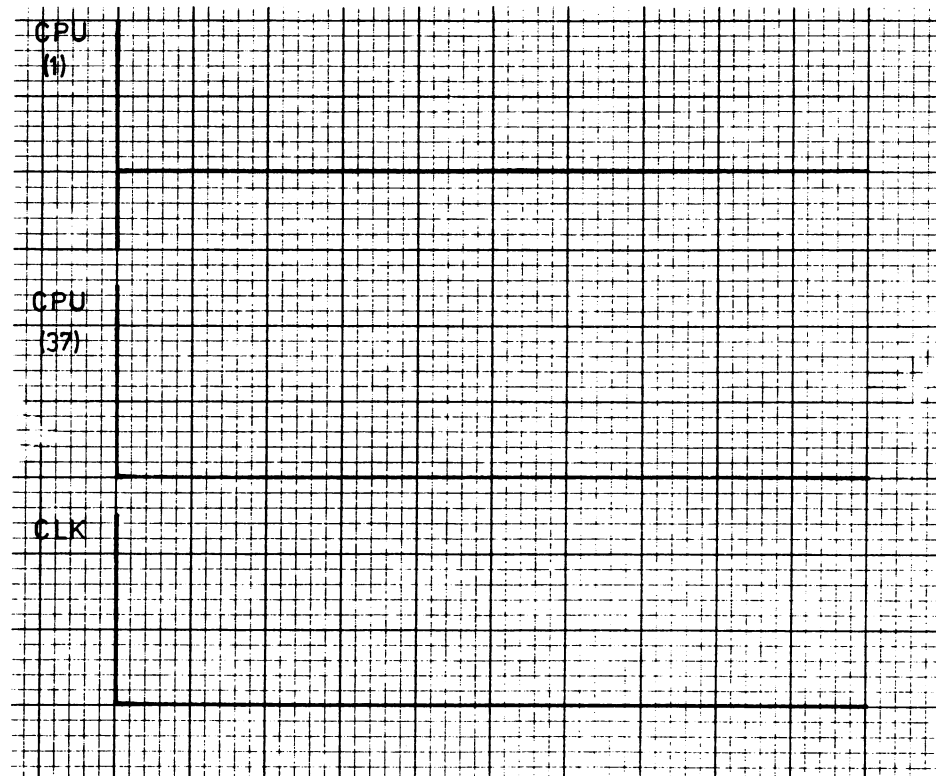
$$U_{PIN1} = f(t) \quad \text{---> an der CPU}$$

$$U_{PIN37} = f(t)$$

$$U_{CLK} = f(t) \quad \text{---> auf der Prüfplatine}$$

Benutzen Sie für die Messungen an der CPU den 40poligen Adapter.

Die Messungen an der CPU müssen mit einem 10:1-Tastkopf durchgeführt werden !



## A5.5

## RESET-Erzeugung

Der Programmzähler der CPU wird bei L-Pegel am Eingang  $\overline{\text{RESIN}}$  auf 0000 zurückgesetzt. Außerdem werden die Adreß-, Daten- und Steuerleitungsanschlüsse in den hochohmigen Zustand geschaltet. Der Ausgang RESOUT gibt in dieser Phase H-Pegel ab.

Überprüfen Sie die beschriebene Funktion, indem Sie über den 40poligen Meßadapter an den PIN 36 der CPU L-Pegel anlegen. Messen Sie mit einem Logiktester den Pegel an den in der folgenden CPU-Anschlüssen.

$\overline{\text{RESIN}}$	RESOUT	AD0	AD1	AD2	AD3	AD4	AD5	AD6	AD7
L									

$\overline{\text{RESIN}}$	A8	A9	A10	A11	A12	A13	A14	A15	$\overline{\text{WR}}$	$\overline{\text{RD}}$	IO/ $\overline{\text{M}}$
L											

**HINWEIS:** Bei richtiger Funktion der CPU muß der Ausgang RESOUT "H-Pegel" führen, während alle anderen Ausgänge hochohmig sein müssen!

## Überprüfung der Adreßsignale

Der Mikroprozessor arbeitet ein vorgegebenes Programm immer sequentiell ab, d.h. es wird Befehl nach Befehl abgearbeitet. Bei der Überprüfung der Adreßleitungen wird die CPU durch das fest eingestellte Datenbyte auf der Prüfkarte (SD0-SD7) dazu gezwungen, immer den gleichen 1-Byte-Befehl abzuarbeiten. Das hat zur Folge, daß der gesamte Adreßbereich von 0000-FFFF durchlaufen wird und alle Adressen hintereinander auf dem Adreßbus erscheinen. Dieser Vorgang wiederholt sich so lange, bis die Betriebsspannung der CPU abgeschaltet wird. Der Programmzähler der CPU wird also ständig um 1 erhöht. Mißt man die Adreßsignale, so müssen zwei benachbarte Leitungen ein Frequenzverhältnis von z.B. fA0:fA1 = 2:1 aufweisen. Um diesen Sachverhalt zu verdeutlichen, wird die CPU zuerst im STEP-Betrieb betrieben. Stellen Sie den Schalter RUN/HLT der Prüfkarte auf HLT. Mit den Schaltern SD7-SD0 wird das Datenbyte 01111111 = 7F H auf der Prüfkarte eingestellt. Danach wird die CPU durch kurzzeitiges Ausschalten der Betriebsspannung zurückgesetzt.

Messen Sie jetzt die Pegel auf den Adreßleitungen A0, A1 und A2 auf der Prüfkarte. Danach betätigen Sie einmal die STEP-Taste und messen wiederum den Pegel auf den genannten Adreßleitungen. Ihre Meßergebnisse tragen Sie in die Tabelle auf der nächsten Seite ein.

--->

## Meßwerttabelle für den Stepbetrieb

Step	A2	A1	A0
0			
1			
2			
3			
4			
5			

Schalten Sie jetzt den Schalter RUN/HLT der Prüfkarte auf RUN. Die CPU arbeitet den gesamten Adreßbereich jetzt mit der Geschwindigkeit der Taktfrequenz ab (siehe Erklärung auf Seite 40). Messen Sie die Adreßsignale an den Stecklötösen der Prüfkarte und tragen Sie die Meßergebnisse in die folgende Tabelle ein. Achten Sie darauf, daß die Signale immer symetrisch sein müssen.

## Meßwerttabelle für den dynamischen Betrieb

$T_{A0} =$ $\mu s$	$T_{A1} =$ $\mu s$
$T_{A2} =$ $\mu s$	$T_{A3} =$ $\mu s$



## ERGEBNISSE DER MESSUNGEN VON SEITE 41

Meßergebnisse zum Stepbetrieb

Step	A2	A1	A0
0	L	L	L
1	L	L	H
2	L	H	L
3	L	H	H
4	H	L	L
5	H	L	H

Meßergebnisse zum dynamischen Betrieb

$T_{A0} = 4 \mu s$	$T_{A1} = 8 \mu s$
$T_{A2} = 16 \mu s$	$T_{A3} = 32 \mu s$

## A5.8

## Adreßzwischenspeicherung mit Hilfe des ALE-Signals

Wie auf Seite 8 schon beschrieben, besitzt die CPU 8085 nur 16 Anschlüsse für Adressen- und Datensignale. Darum werden die Anschlüsse AD0-AD7 (12-19) im Multiplexbetrieb betrieben. Die 16-Bit-Adresse wird deshalb durch die Bausteine IC3 und IC4 zwischengespeichert. Die Zwischenspeicherung erfolgt immer dann, wenn das  $\overline{\text{ALE}}$ -Signale von H auf L wechselt.

Betreiben Sie die CPU wie auf der vorhergehenden Seite dynamisch und messen Sie mit Hilfe des Oszilloskops folgende Signale an den Stecklötösen der Prüfplatine:

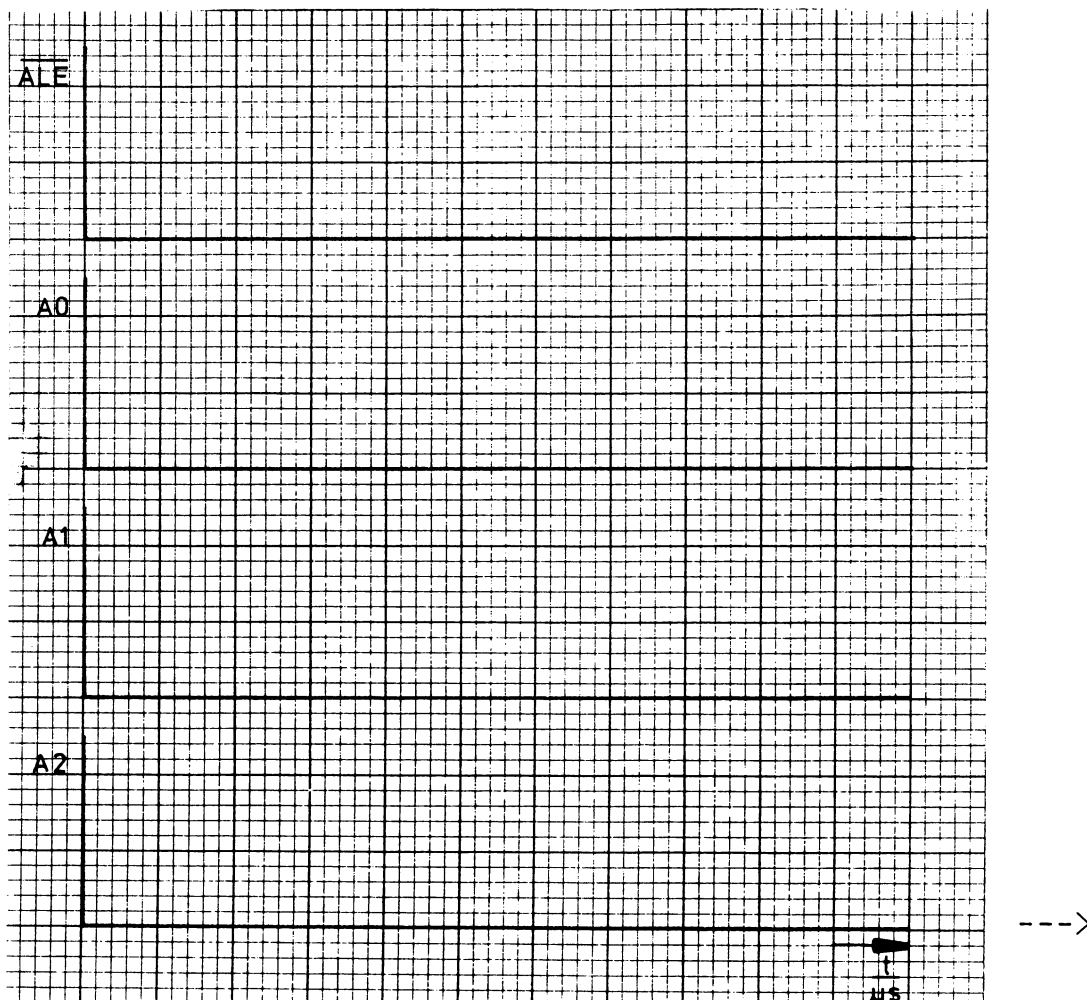
$$\overline{\text{ALE}} = f(t)$$

$$\text{A0} = f(t)$$

$$\text{A1} = f(t)$$

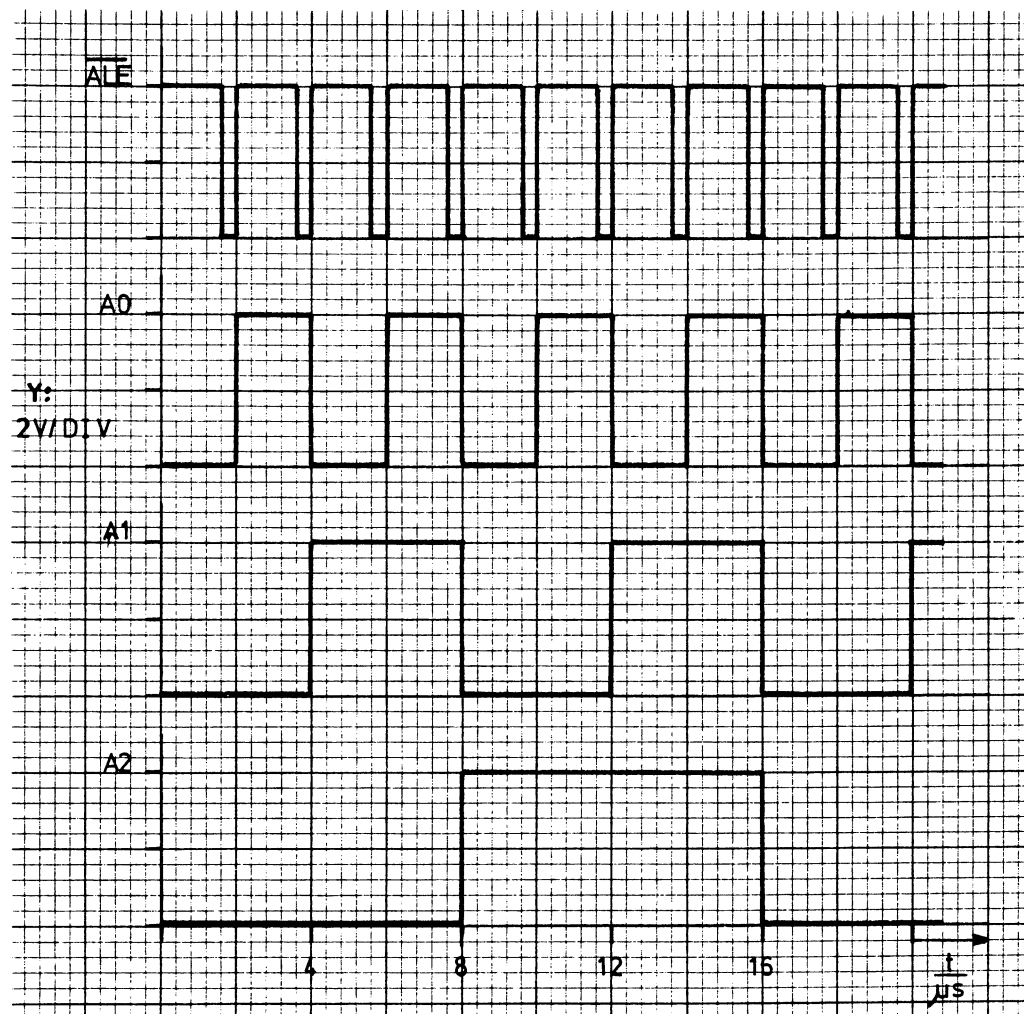
$$\text{A2} = f(t)$$

Die Meßergebnisse tragen Sie in die Koordinatensysteme auf dieser Seite ein.



A5.9

## ERGEBNISSE ZU DEN MESSUNGEN VON SEITE 43



## HINWEIS ZU DEN MESSERGEBNISSEN

Ein Signalwechsel auf dem Adreßbus findet immer dann statt, wenn das ALE-Signal einen L-H-Sprung durchführt. Die Triggerung des Oszilloskops erfolgte bei den dargestellten Diagrammen auf der negativen Flanke des Signals A13.

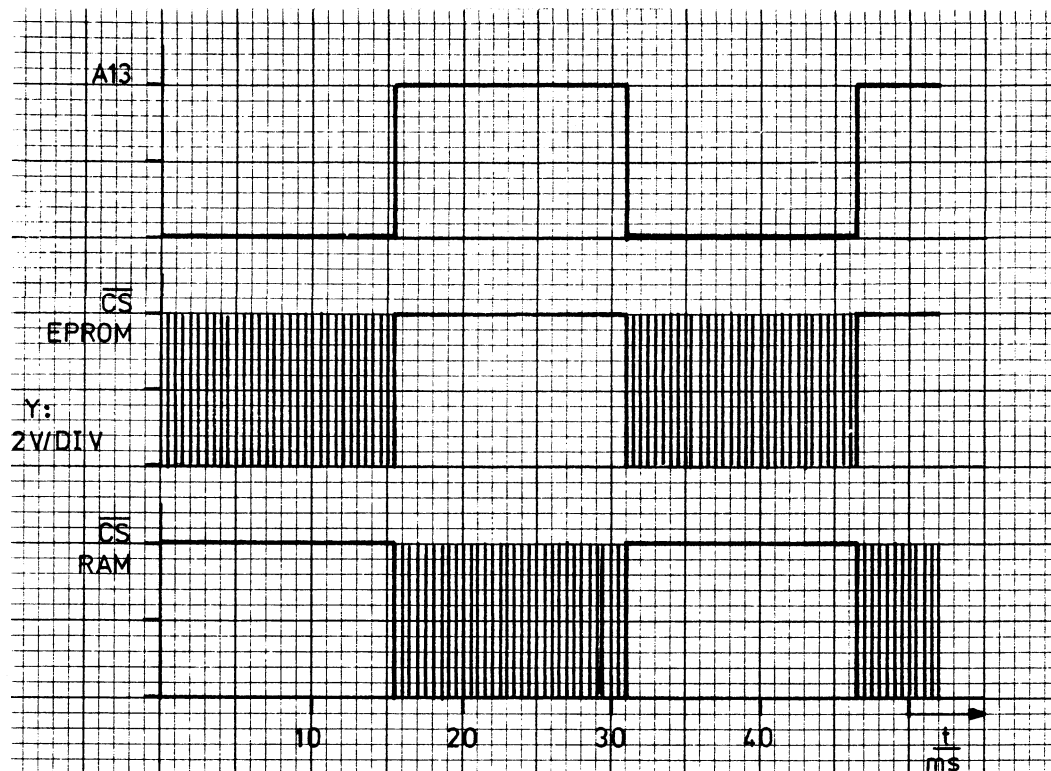
---&gt;

## Adreßdecodierung für die Speicherbausteine

Wie auf den Seiten 16 und 17 schon beschrieben, wird der Speicherbereich des Einplatinencomputers in ein EPROM mit 2 oder 4 KByte und ein RAM mit 2 KByte Speicherkapazität unterteilt. Das EPROM wird immer dann angesprochen, wenn die Adreßleitung A13 L-Pegel führt, das RAM wird angesprochen, wenn A13 H-Pegel führt. Überprüfen Sie diesen Sachverhalt, indem Sie die Signale der Adreßleitung A13 und der beiden  $\overline{CS}$ -Eingänge der Speicherbausteine oszilloskopieren. Die CPU wird genauso betrieben wie auf der vorhergehenden Seite. Die Meßergebnisse tragen Sie in die Koordinatensysteme auf dieser Seite ein.



## ERGEBNISSE ZU DEN MESSUNGEN VON SEITE 45



## HINWEIS ZU DEN MESSUNGEN

Durch die Decodierung der Adreßleitung A13 wird das EPROM bei L-Signal dieser Leitung angesprochen, das RAM bei H-Signal auf dieser Leitung. Innerhalb der aktiven Zeit wird der  $\overline{CS}$ -Eingang des jeweiligen Bausteins insgesamt 8191mal angesprochen. Das entspricht z.B. dem Adreßbereich 0000-1FFF oder 2000-2FFF. Die Triggerung des Oszilloskops erfolgte auf der negativen Flanke von A13.

--->

## Einplatinencomputer

A5.12

## Befehls-Abarbeitung durch die CPU

Die Befehlsabarbeitung kann grundsätzlich in zwei Schritte eingeteilt werden:

## 1. Befehlsholphase

In dieser Phase liest die CPU das Befehlsbyte aus dem Speicher. Das gelesene Byte wird anschließend in der CPU decodiert. Sind zur Abarbeitung noch weitere Daten erforderlich, so bewirkt die Ablaufsteuerung, daß diese Daten ebenfalls gelesen werden.

## 2. Ausführungsphase

Sind alle Daten zur Ausführung des Befehls bekannt, dann wird der Befehl ausgeführt und anschließend das nächste Befehlsbyte gelesen.

Mit Hilfe der Schalter SD7-SD0 auf der Prüfkarte werden jetzt verschiedene Befehlsbytes fest eingestellt, so daß die CPU immer den gleichen Befehl ausführen muß. Mit Hilfe des Oszilloskops können dann die Steuersignale gemessen und die Arbeitsweise der CPU überprüft werden.

Stellen Sie zuerst das Datenbyte 00101111 = 2F H ein. Dieses Datenbyte entspricht dem 1-Byte-Befehl CMA = Complement Accu.

Oszilloskopieren Sie die auf der nächsten Seite genannten Signale und übertragen Sie die Meßergebnisse in die entsprechenden Koordinatensysteme.

**HINWEIS:** Die Triggerung sollte auf der negativen Flanke von  $\overline{\text{MEMR}}$  erfolgen.

---&gt;

## A5.13

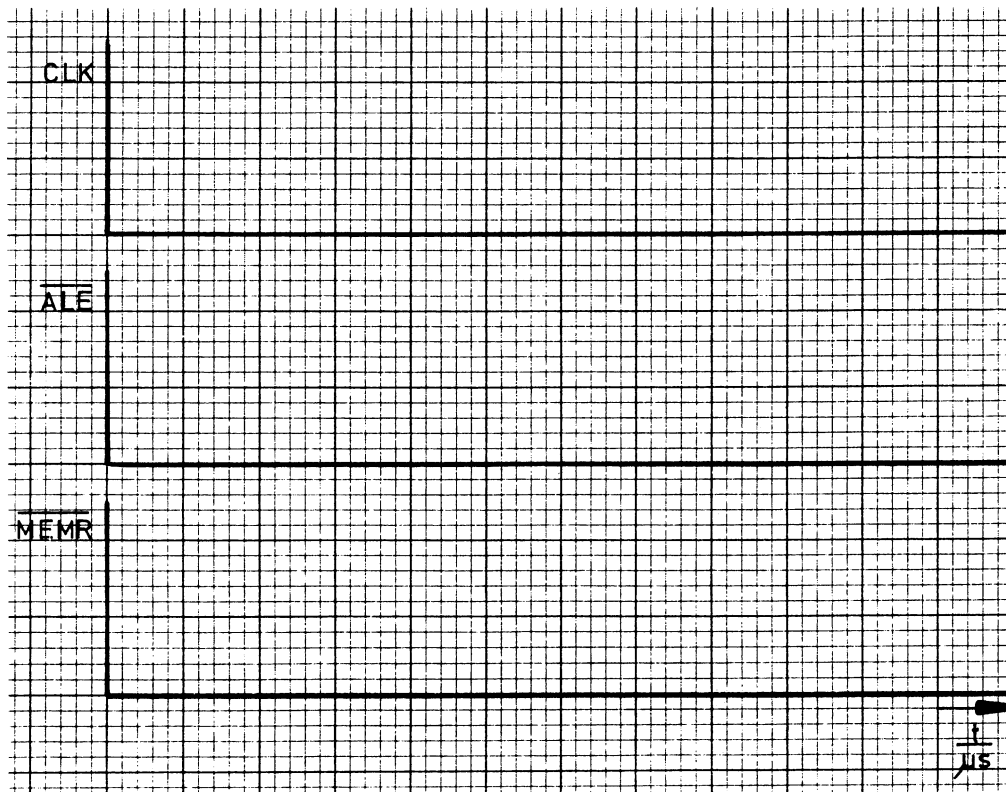
Oszilloskopieren und zeichnen Sie folgende Signale:

$$\text{CLK} = f(t)$$

$$\overline{\text{ALE}} = f(t)$$

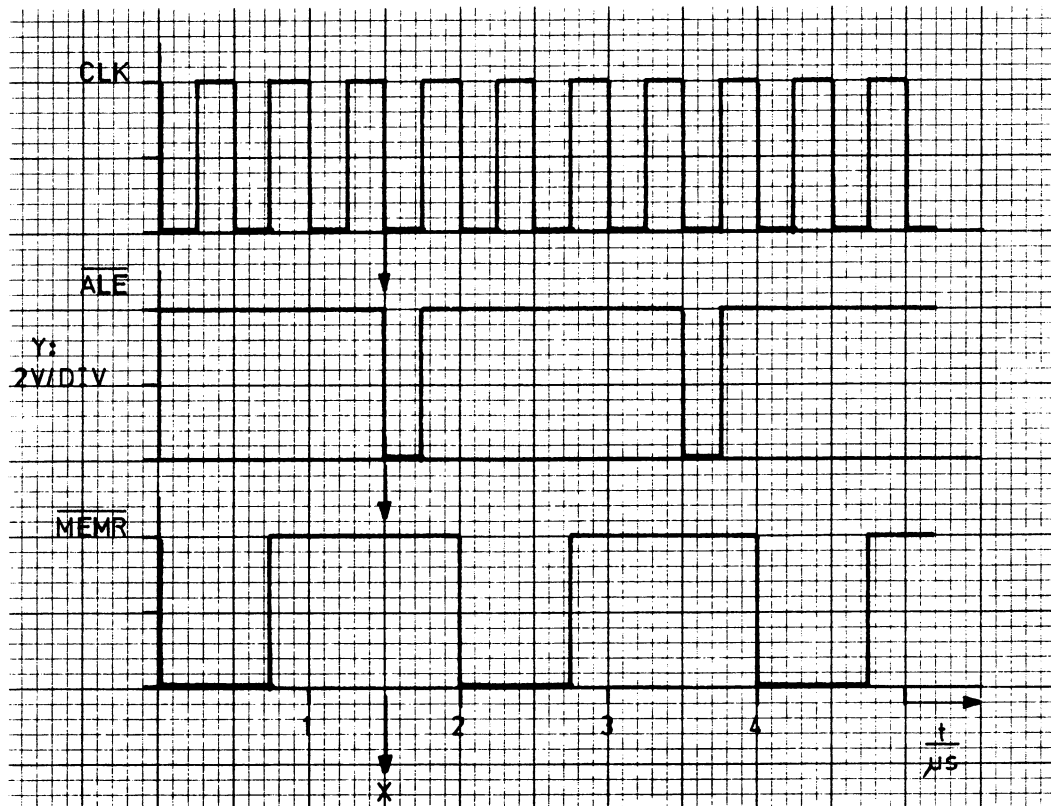
$$\overline{\text{MEMR}} = f(t)$$

Alle Signale können auf der Prüfplatine gemessen werden.



--->

## ERGEBNISSE ZU DEN MESSUNGEN VON SEITE 48



## HINWEIS ZU DEN MESSUNGEN

Die Beschreibung der Signale bezieht sich auf den im Diagramm gekennzeichneten Zeitpunkt "X" !

Zu diesem Zeitpunkt sendet die CPU eine neue Adresse aus, die mit der darauf folgenden positiven Flanke von ALE auf dem Adreßbus ansteht. Eine halbe Taktperiode danach wird MEMR aktiviert und das unter der vorher ausgesandten Adresse stehende Befehlsbyte (2F) wird gelesen, entschlüsselt und ausgeführt. Danach beginnt der beschriebene Vorgang erneut.

--->



A5.15

Stellen Sie jetzt das Datenbyte 11011011 = DB H ein. Dieses Datenbyte entspricht dem IN-Befehl.

Oszilloskopieren und zeichnen Sie folgende Signale:

$$\text{CLK} = f(t)$$

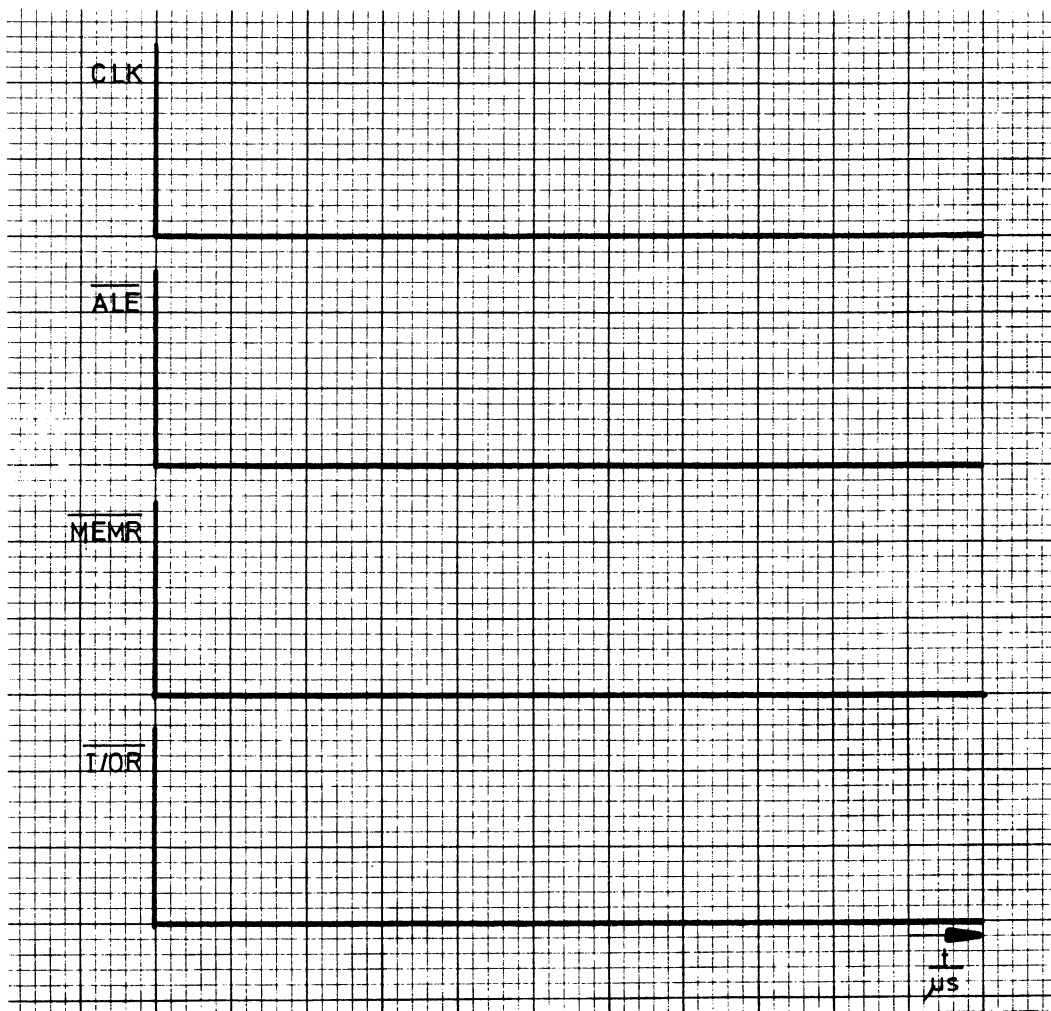
$$\overline{\text{ALE}} = f(t)$$

$$\overline{\text{MEMR}} = f(t)$$

$$\overline{\text{T/OR}} = f(t)$$

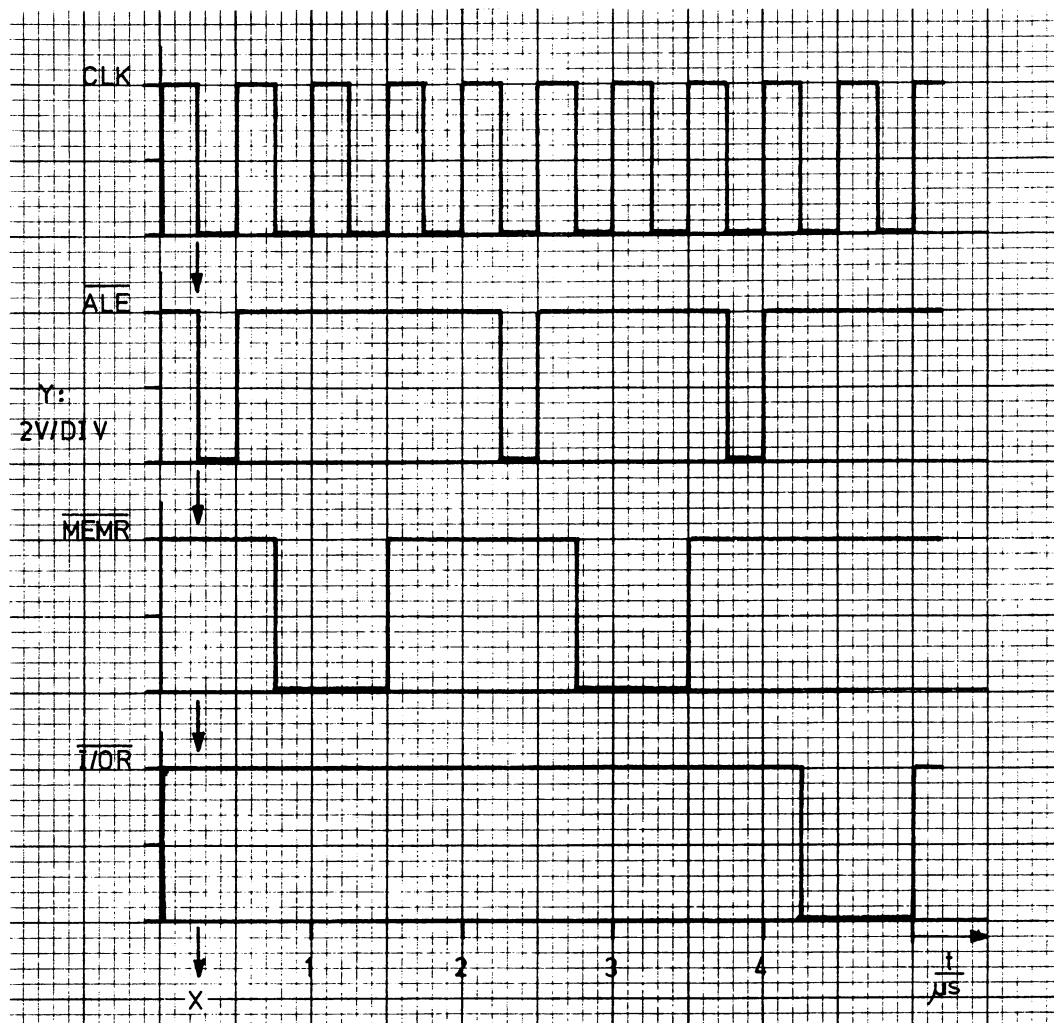
Alle Signale können auf der Prüfplatine gemessen werden.

HINWEIS: Die Triggerung sollte auf der positiven Flanke von  $\overline{\text{T/OR}}$  erfolgen!



---&gt;

## ERGEBNISSE ZU DEN MESSUNGEN VON SEITE 50



## HINWEIS ZU DEN MESSUNGEN

Die Beschreibung der Signale bezieht sich auf den im Diagramm gekennzeichneten Zeitpunkt "X" !

Zu diesem Zeitpunkt sendet die CPU eine neue Adresse aus, die mit der darauf folgenden positiven Flanke von  $\overline{ALE}$  auf dem Adreßbus ansteht. Danach wird  $\overline{MEMR}$  aktiviert und das Befehlsbyte "DB" wird gelesen und entschlüsselt. Da es sich um den IN-Befehl handelt, muß die CPU noch die Adresse der Eingabeeinheit aus dem Speicher lesen.  $\overline{ALE}$  wird aktiviert (die Portnummer steht immer auf der nächst höheren Adresse des Befehlsbyte), danach wird  $\overline{MEMR}$  aktiviert und damit sind alle Daten zur Ausführung des Befehls bekannt. Die CPU aktiviert jetzt nochmals das  $\overline{ALE}$ -Signal, um die Adresse der Eingabeeinheit auszusenden. Danach wird die Steuerleitung  $\overline{IOR}$  aktiviert und mit der positiven Flanke dieses Signals dann das Eingabebyte gelesen.

## Einplatinencomputer

## A6.1

Jetzt muß noch überprüft werden, ob der Einplatinencomputer ein im EPROM abgelegtes Programm richtig abarbeitet und der RAM-Baustein richtig angesprochen wird.

Sie benötigen dazu ein komplett aufgebautes MFA-System, das als Entwicklungssystem für das Prüfprogramm dient und das auch zum Programmieren des EPROM's dienen soll.

Das MFA-System muß folgende Einschübe enthalten:

- Prozessor 8085
- 8 KByte EPROM mit MAT 85 (Basisadresse 0000)
- 8 kByte RAM (Basisadresse E000)
- Ausgabekarte (Adresse 00)
- Video-Interface
- EPROM-Programmierkarte
- Spannungsregelung
- Netzteil

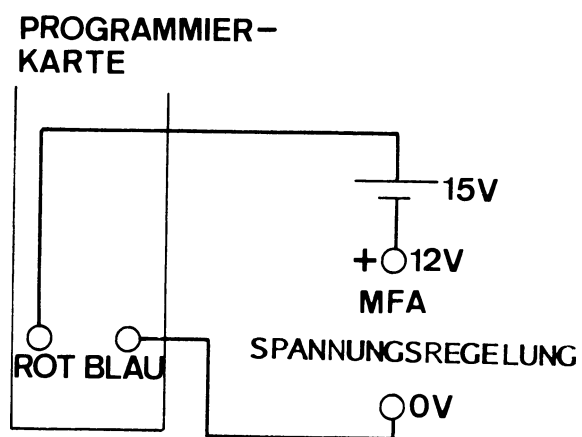
zusätzlich benötigen Sie:

- Tastatur
- Monitor
- Gleichspannungsquelle 15 V
- 1 gelöscht EPROM 2716

Beschalten Sie die Gleichspannungsquelle, die Spannungsregelung und die Programmierkarte nach dem Stromlaufplan auf der nächsten Seite.

--->

Stromlaufplan für die Zusammenschaltung von MFA-Spannungsregelung, Gleichspannungsquelle und EPROM-Programmierkarte.



## Einplatinencomputer

## A6.3

Wie am Anfang dieser Übung bereits beschrieben wurde, soll der Einplatinencomputer ein vorgegebenes Anwenderprogramm abarbeiten. Dieses Programm muß im EPROM des Einplatinencomputers stehen. Die folgenden Punkte sollen Ihnen deutlich machen, welche Schritte erforderlich sind, bis dieses Anwenderprogramm auf dem Einplatinencomputer lauffähig ist.

1. Formulierung der Problemstellung für das Anwenderprogramm

- Der Einplatinencomputer soll mit Hilfe des Programms seinen eigenen RAM-Speicherbereich testen und über die Ausgabebaugruppe des MFA-Systems das Testergebnis anzeigen.

2. Das Anwenderprogramm wird auf einem Entwicklungssystem geschrieben

- Das MFA-System dient hier als Entwicklungssystem. Das Anwenderprogramm wird mit Hilfe des Assemblercodes in den RAM-Bereich des MFA-Systems eingegeben.

3. Testphase des Anwenderprogramms auf dem Entwicklungssystem

- Über das GO-Kommando wird das Anwenderprogramm gestartet. Der überprüfte RAM-Bereich muß zum einen außerhalb des Anwenderprogramms liegen (das Programm würde sonst durch den Testlauf zerstört), zum anderen muß der überprüfte RAM-Bereich dem des Einplatinencomputers entsprechen (2 KByte). Sollten bei dieser Testphase Fehler im Programmablauf auftreten, lassen sie sich problemlos korrigieren, da das Programm im RAM-Speicher steht.

4. Das Anwenderprogramm muß auf den EPROM-Adreßbereich des Einplatinencomputers zugeschnitten werden

- Das Anwenderprogramm wird im MFA-System ab der Adresse E000 abgelegt. Sämtliche Sprungbefehle und Unterprogramme beziehen sich auf diese Anfangsadresse. Auf dem Einplatinencomputer muß das Anwenderprogramm aber mit der Startadresse 0000 beginnen.

5. Das Anwenderprogramm wird mit Hilfe eines EPROM-Programmiergerätes in einem EPROM abgelegt.

- Das MFA-System wird hier zum Programmieren des EPROMs verwendet.

6. Der Einplatinencomputer wird mit dem programmierten EPROM bestückt. Mit Hilfe einer Anzeige wird überprüft, ob das Anwenderprogramm korrekt abgearbeitet wird.

- Als Anzeige dient hier die Ausgabebaugruppe des MFA-Systems.

--->

**Formulierung der Problemstellung für das Anwenderprogramm**

Das Anwenderprogramm soll einen RAM-Speicherbereich von 2 KByte überprüfen. Es soll außerdem über eine Ausgabebaugruppe anzeigen, ob der getestete Speicherbereich funktionsfähig ist oder nicht.

Der Aufbau des Programms sieht folgendermaßen aus:

- Zuerst wird das Datenwort 55H in die niedrigste Speicheradresse geschrieben und danach aus der gleichen Adresse gelesen. Dann werden geschriebenes und gelesenes Datenwort miteinander verglichen. Fällt der Vergleich positiv aus, wird das Programm fortgesetzt, ansonsten erfolgt ein Programmabbruch mit entsprechender Anzeige.
- Bei Fortsetzung des Programms wird dann das Datenwort AAH in die gleiche RAM-Adresse geschrieben, gelesen und verglichen. Die Reaktion des Programms auf positives oder negatives Ergebnis ist dabei die gleiche wie im ersten Schritt.
- Auf diese Weise werden sämtliche Speicheradressen getestet. Taucht bei einer Adresse ein Fehler auf, wird das Programm abgebrochen und eine Fehlermeldung ausgegeben. Sind alle Speicheradressen in Ordnung, dann erfolgt ebenfalls eine entsprechende Anzeige.

## Einplatinencomputer

## A6.5

**Das Anwenderprogramm wird auf einem Entwicklungssystem geschrieben**

Überprüfen Sie vor dem Einschalten des MFA-Systems, ob die auf Seite 52 beschriebenen Einschübe vorhanden sind. Schalten Sie das System ein und rufen Sie nach dem Start des Betriebsprogramms den Assembler auf. Danach geben Sie, beginnend bei der Adresse E000 folgendes Programm ein.

	LXI	SP,27FF	;Stackpointer setzen
	LXI	H,E100	;Anf.adr. RAM-Bereich
ANF:	MVI	A,55	;Datenwort laden
	MOV	B,A	; " sichern
	MOV	M,A	;RAM-Adr. schreiben
	MOV	A,M	;RAM-Adr. lesen
	CMP	B	;RAM-Adr. OK ?
	JNZ	STOP	;Nein !
	MVI	A,0AA	;Ja, inv. Datenwort laden
	MOV	B,A	;inv. Datenwort sichern
	MOV	M,A	;RAM-Adr. schreiben
	MOV	A,M	;RAM-Adr. lesen
	CMP	B	;RAM-Adr. OK ?
	JNZ	STOP	;Nein !
	INX	H	;Ja, Adr.zähler erhöhen
	MOV	A,H	;Zählerstand auf
	CPI	E8	;höchster RAM-Adresse ?
	JNZ	ANF	;Nein,nächste Adr. prüfen
ANF1:	MVI	A,0FF	;Alle
	OUT	00	;RAM-Adressen
	CALL	ZEIT	;sind in
	MVI	A,00	;Ordnung
	OUT	00	
	CALL	ZEIT	
	JMP	ANF1	
STOP:	MVI	A,55	;Eine RAM-Adr. ist
	OUT	00	;nicht in
	JMP	STOP	;Ordnung
ZEIT:	MVI	B,0F	;Zeitschleife
Z1:	LXI	D,1999	;ca. 1sec
Z2:	DCX	D	
	MOV	A,E	
	ORA	D	
	JNZ	Z2	
	DCR	B	
	JNZ	Z1	
	RET		

--->

**Testphase des Anwenderprogramms auf dem Entwicklungssystem**

Bei diesem Testlauf ist folgendes zu beachten:

Der erste Befehl LXI SP,27FF soll nachher den Stackpointer des Einplatinencomputers definiert auf die höchste RAM-Adresse setzen. Dieser Befehl darf beim Testlauf im MFA-System nicht abgearbeitet werden!

**DAS PROGRAMM IST DESHALB IM MFA-SYSTEM AB DER ADRESSE E003 ZU STARTEN!!!!**

Außerdem wurde das H-L-Register so geladen, daß sich das Programm beim Testlauf nicht selber zerstört.

Starten Sie jetzt das Programm mit dem GO-Befehl ab der Adresse

**E003**

Ist das Programm in Ordnung (die LED's der Ausgabebaugruppe müssen im Sekundentakt blinken), kann das EPROM für den Einplatinencomputer programmiert werden.

--->



## Einplatinencomputer

## A6.7

**Das Anwenderprogramm muß auf den EPROM-Adreßbereich des Einplatinencomputers zugeschnitten werden**

Das Anwenderprogramm muß für den Einplatinencomputer ab der Adresse 0000h lauffähig sein. Deshalb sind die in folgender Liste aufgeführten Adressen zu ändern. Die Änderung läßt sich am einfachsten mit dem "Memory"-Kommando des MFA-Betriebssystems durchführen.

Liste der zu ändernden Befehle

ADRESSE	ALTER BEFEHL	GEÄNDERTER BEFEHL
E003	LXI H,E100	LXI H,2000
E00C	JNZ E030	JNZ 0030
E015	JNZ E030	JNZ 0030
E01A	CPI E8	CPI 28
E01C	JNZ E006	JNZ 0006
E023	CALL E037	CALL 0037
E02A	CALL E037	CALL 0037
E02D	JMP E01F	JMP 001F
E034	JMP E030	JMP 0030
E03F	JNZ E03C	JNZ 003C
E043	JMP E039	JMP 0039

---&gt;

## Einplatinencomputer

## A6.8

**Das Anwenderprogramm wird mit Hilfe eines EPROM-Programmiergerätes in einem EPROM abgelegt**

Überprüfen Sie, ob das EPROM-Programmiergerät im MFA-System und die zusätzliche Spannungsquelle richtig miteinander verbunden sind.

Danach wird ein gelöscht EPROM 2716 in die Fassung des Programmiergerätes gesteckt und das Programmierprogramm mit der Adresse 1E06 gestartet. Nach dem Aufruf erscheint auf dem Bildschirm folgender Ausdruck:

```
TEST
READ
PROC
COMP
QUIT
```

Zuerst muß das eingesteckte EPROM auf richtige Funktion getestet werden. Dazu "T" eingeben. Ist das EPROM in Ordnung, erscheint auf dem Bildschirm der Ausdruck "READY".

Danach "P" eingeben. Jetzt wird die Startadresse der zu programmierenden Daten abgefragt. Geben Sie "E000" ein.

Das EPROM wird jetzt programmiert. Dieser Vorgang dauert ca. 2 Min.

Der Einplatinencomputer wird mit dem programmierten EPROM bestückt. Mit Hilfe einer Anzeige wird überprüft, ob das Anwenderprogramm korrekt abgearbeitet wird

Schalten Sie jetzt die Betriebsspannung am Einplatinencomputer ab und entfernen Sie die Prüfplatine. Danach wird der Einplatinencomputer mit EPROM und RAM bestückt.

Um die richtige Funktion des Einplatinencomputers zu überprüfen, wird er mit der Ausgabe-Baugruppe zusammen im MFA-System betrieben. Schalten Sie das MFA-System und die zusätzliche Gleichspannungsquelle aus und entfernen Sie außer der Ausgabebaugruppe, der Spannungsregelung und dem Netzteil alle Einschübe. Danach schieben Sie den Einplatinencomputer in das MFA-System und schalten das System wieder ein.

Der Einplatinencomputer arbeitet jetzt das im EPROM befindliche Programm ab. Da die Abarbeitung sehr schnell erfolgt, werden Sie schon nach kurzer Zeit eine entsprechende Reaktion an der Ausgabebaugruppe feststellen.

Damit ist die Übung beendet!



## EPC-Ein/Ausgabe

Fachtheoretischer Teil	Seite
1. Konzept der Ein/Ausgabekarte	2
2. Adressierung der Ein/Ausgabekarte	5
3. TTL-kompatible Eingabe	5
4. Optokoppler-Eingabe	7
5. A/D-Wandler	8
6. TTL-kompatible Ausgabe	9
7. Optokoppler-Ausgabe	9
8. D/A-Wandler	10
 Fachpraktischer Teil	
Flußdiagramm für den Arbeitsablauf	12
Bereitstellungsliste Ein/Ausgabekarte	13
Bereitstellungsliste I/O-Adapterkarte	15
Übungsteil	16
Arbeitsblätter	
A1. Bestücken der Leiterplatte I/O-EPC	18
A2. Bestücken der Leiterplatte I/O-Adapter	23
A3. Funktionsprüfung der I/O-Adapterkarte	26
A4. Sichtkontrolle der Ein/Ausgabekarte	27
A5. Statische Funktionsprüfung der Ein/Ausgabekarte	27
A6. Dynamische Funktionsprüfung der Ein/Ausgabekarte	38

## EPC-Ein/Ausgabe

## 1. Konzept der Ein/Ausgabekarte

Diese Zusatzkarte ist als Erweiterung zum Einplatinencomputer entwickelt worden. Der Einplatinencomputer wird in der Übung EPC ausführlich beschrieben. Im folgenden wird die Ein/Ausgabekarte beschrieben. Die Karte ist über eine 64polige Federleiste direkt an den Einplatinencomputer adaptierbar. Beide Systeme werden dabei von einer Betriebsspannung (5 V) versorgt. Dadurch steht dem Anwender ein vollkommen autonomes Computersystem zur Verfügung, mit dem sich fast alle in der Praxis vorkommenden Problemstellungen lösen lassen.

Alle Ein- und Ausgänge der Karte können über Lötstecker und entsprechende Verbinder direkt belegt werden.

Es sind folgende Ein- bzw. Ausgänge vorhanden:

**TTL-kompatible Ein/Ausgabe**

**Ein/Ausgabe über Optokoppler**

**Analog-Eingabe und -Ausgabe  
über A/D- bzw. D/A-Wandler**

Für die TTL-kompatible Ein/Ausgabe werden konventionelle Bausteine in LS-Technologie benutzt. Beide Kanäle sind für parallele 8Bit-Datenverarbeitung ausgelegt.

Die Ein/Ausgabe über Optokoppler erfolgt jeweils mit 4fach-Optokopplern. Auch hier erfolgt eine parallele Datenverarbeitung mit 8 Bit. Um eine vollkommene galvanische Trennung zu erreichen, wurden Anschlüsse für externe Betriebsspannungen vorgesehen. Dadurch können auch Systeme bedient werden, die mit Netzwechselspannung arbeiten.

Die Analog-Eingabe und -Ausgabe wurde mit 8Bit-A/D- bzw. D/A-Wandlern realisiert. Der A/D-Wandler arbeitet nach dem Annäherungsverfahren und benötigt für eine Wandlung eine Zeit von 20  $\mu$ s. Die Laufzeit des D/A-Wandlers für eine Wandlung beträgt zirka 800 ns.

Die Bilder 1.1. und 1.2. stellen die kompletten Stromlaufpläne der Ein/Ausgabekarte dar. Sie müssen für die Funktionsbeschreibung und die Inbetriebnahme der Karte benutzt werden.

EPC-Ein/Ausgabe

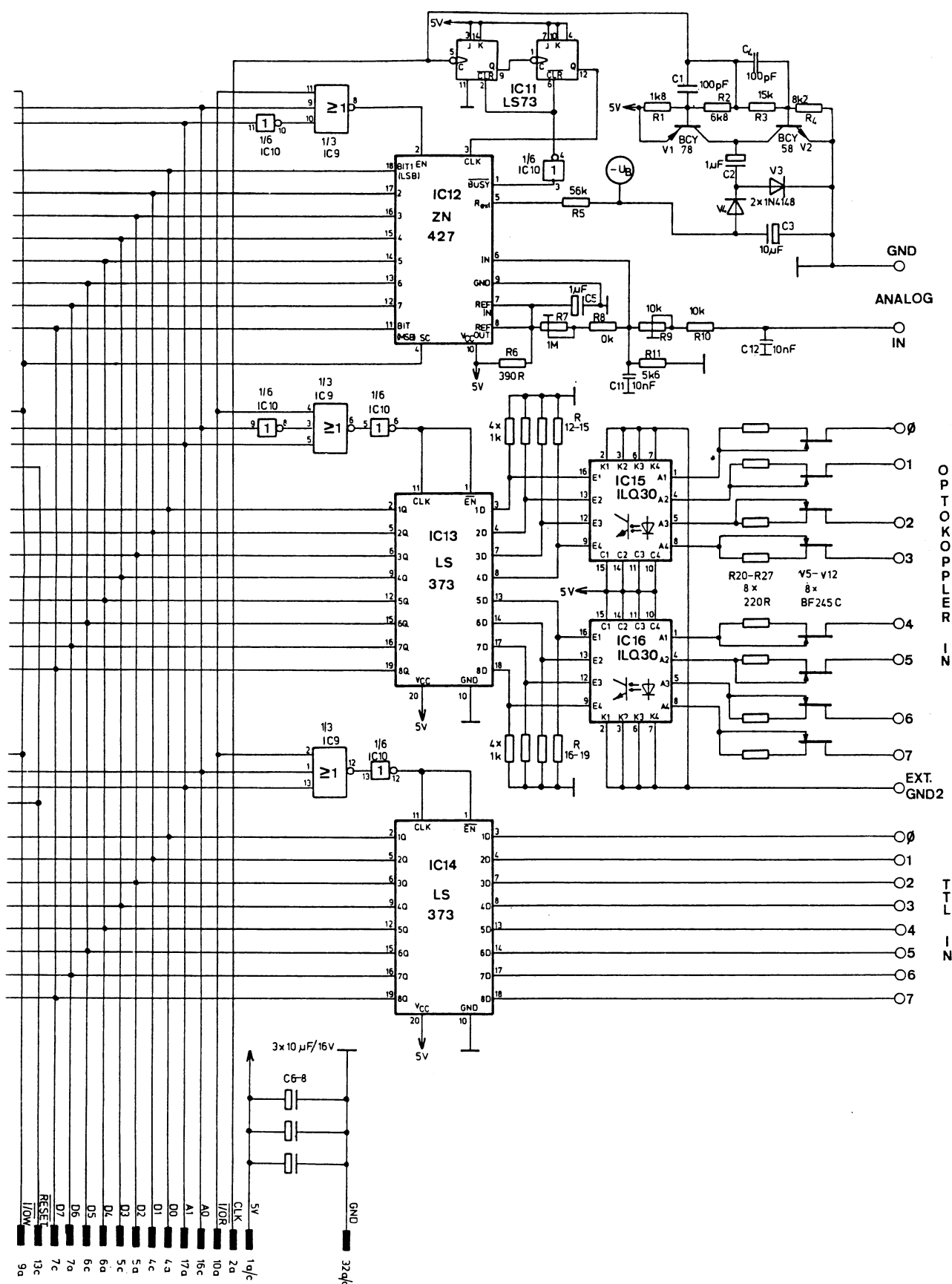


Bild 1.1.: Stromlaufplan der Eingabekanäle

EPC-Ein/Ausgabe

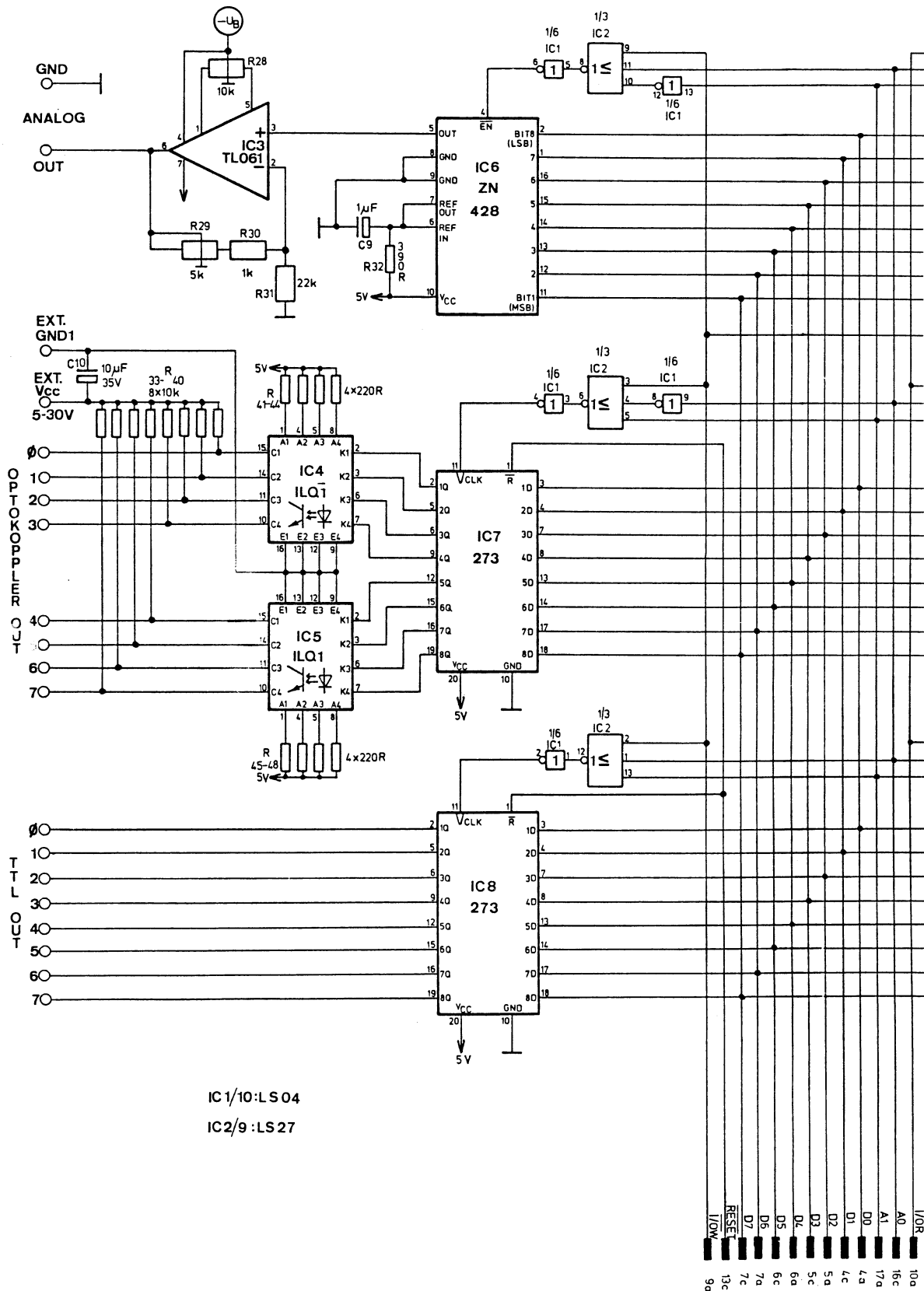


Bild 1.2.: Stromlaufplan der Ausgabekanäle

## EPC-Ein/Ausgabe

## 2. Adressierung der Ein- und Ausgabekanäle

In jedem Mikroprozessorsystem müssen den Eingabe- und Ausgabebaugruppen definierte Adressen zugeordnet werden können, damit sie gezielt ansprechbar sind. Den Ein- und Ausgabekanälen dieser Karte wurden je nach Typ gleiche Adressen zugeordnet. Die Adressierung ist fest "verdrahtet" und kann nicht geändert werden. Zur Unterscheidung zwischen Eingabe und Ausgabe dienen die beiden Steuerleitungen  $\overline{\text{IOR}}$  und  $\overline{\text{IOW}}$ .

Tabelle 1 stellt alle Adressen für Ein- und Ausgabekanäle dar.

	ADRESSE	KANAL			ADRESSE	KANAL
$\overline{\text{IOR}} = \text{L}$	00	TTL-EINGABE		$\overline{\text{IOW}} = \text{L}$	00	TTL-AUSGABE
	01	OPTO-EINGABE			01	OPTO-AUSGABE
	02	A/D-WANDLER			02	D/A-WANDLER

Tabelle 1: Adressierung der Ein- und Ausgabekanäle

## 3. TTL-kompatible Eingabe (IC14)

Der Baustein 74LS373 ist ein taktzustandsgesteuertes 8Bit-D-Flip-Flop. Der Takteingang ist H-aktiv, d.h. die internen D-Flip-Flops übernehmen immer dann die Eingangsdaten, wenn der Takteingang H-Pegel führt. Da es sich um einen Eingabebaustein handelt, sind die Ausgänge in Tristatetechnik ausgeführt. Die Ausgänge werden immer dann aktiv, wenn der Enable-Eingang L-Pegel führt.

Tabelle 2 stellt die Wahrheitstabelle dieses Bausteins dar

CLK	$\overline{\text{EN}}$	D	Q
L	L	H	$t_{n-1}$
L	H	H	hochohmig
H	L	H	H
H	H	H	hochohmig

Tabelle 2: Wahrheitstabelle TTL-Eingabe



### 3.1. Funktionsweise des D-Flip-Flops

Bild 2 zeigt den internen Aufbau des Bausteins 74LS373. Zur übersichtlicheren Darstellung wurde nur ein D-Flip-Flop gezeichnet.

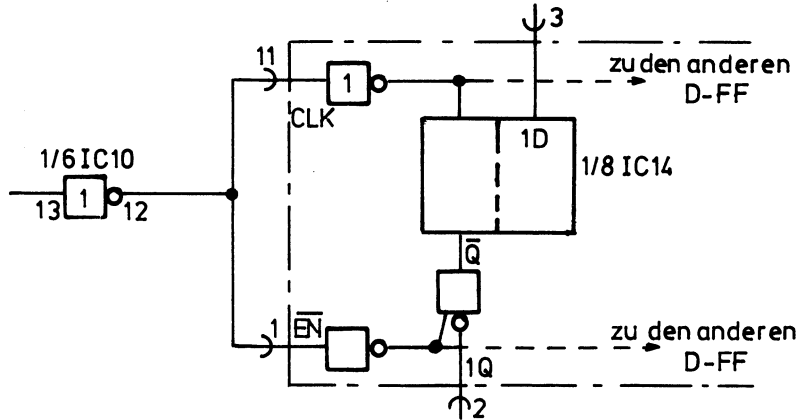


Bild 2: Beschaltung des D-Flip-Flops

Bei der folgenden Beschreibung wird davon ausgegangen, daß auf dem Adreßbus die Adresse 00 anliegt!

Aus dem Stromlaufplan geht hervor, daß die beiden Steuereingänge parallelgeschaltet sind. Dadurch wird folgendes bewirkt:

Führt die Steuerleitung IOR H-Pegel, so liegt dieser Pegel ebenfalls an den beiden Steuereingängen des D-Flip-Flops an. Der Ausgang 1Q befindet sich dann im hochohmigen Zustand. Das D-Flip-Flop ist über den CLK-Eingang aktiviert und übernimmt ständig die Eingangsinformation, welche dann am internen Q-Ausgang anliegt. Führt die Steuerleitung IOR L-Pegel, so wird der gerade anliegende Eingangspegel gespeichert und zum Ausgang 1Q durchgeschaltet. Weitere Wechsel des Eingangspegels haben keinen Einfluß auf die Information am Ausgang, da das D-Flip-Flop gesperrt ist.

Dies ist wichtig, da der Mikroprozessor zu dem zuletzt beschriebenen Zeitpunkt die Information am Q-Ausgang des D-Flip-Flops liest. Würde das D-Flip-Flop nicht gesperrt, so könnte während des Lesezyklus noch ein Signalwechsel stattfinden, was keinen eindeutigen Pegel auf dem Datenbus zur Folge hätte.

Tabelle 3 zeigt die Wahrheitstabelle des Bausteins im Zusammenhang mit dem Adreßbus und der Steuerleitung IOR.

Adresse	IOR	CLK/EN	Wirkung
00	H	H	D-FF aktiv, Ausgang 1Q hochohmig
00	L	L	D-FF gesperrt, Ausgang 1Q niederohmig
01	X	H	Baustein inaktiv
02	X	H	Baustein inaktiv

Tabelle 3: Steuerung der TTL-Eingabe

#### 4. Optokoppler-Eingabe (IC15 und IC16)

Optokoppler bieten den Vorteil, daß zwischen Ein- und Ausgängen eine galvanische Trennung herrscht. Als Sender dient eine Leuchtdiode, deren abgestrahltes Licht den Empfänger, einen Fototransistor, ansteuert. Beide Bauelemente sind zum Schutz vor Fremdlicht in einem lichtdichten Gehäuse untergebracht.

Bei der hier vorliegenden Karte werden aus Platzgründen zwei Vierfach-Optokoppler 1LQ30 verwendet. Die Leuchtdioden (Sender) werden durch Konstantstromquellen mit einem Nennwert von 5,5 mA gespeist. Das hat den Vorteil, daß die Eingangspegel zwischen 5 V und 30 V liegen können, ohne daß sich die Lichtstärke der Dioden ändert. Die Eingangsspannung muß zwischen dem jeweiligen Eingang und dem Anschluß EXT.GND2 angelegt werden. Ein offener Eingang erzeugt bei diesen Bausteinen einen L-Pegel am Ausgang, da die Signale am Emitter der Fototransistoren (Empfänger) abgegriffen werden.

Bild 3 zeigt die Beschaltung eines Optokopplers.

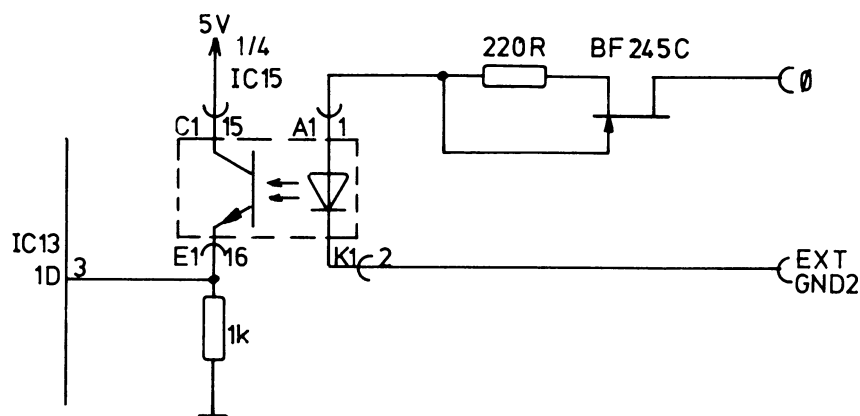


Bild 3: Beschaltung der Optokoppler für die Eingabe

Da die Optokoppler keine Tristate-Ausgänge besitzen, wurde ihnen ein achtfach-D-Flip-Flop 74LS373 nachgeschaltet, dessen Funktionsweise auf den Seiten 5 und 6 beschrieben wurde.

Für diesen Optokoppler wurde ein Typ gewählt, dessen Stromübertragungsverhältnis  $I_C/I_F > 100$  ist. Die relativ hohe Kollektor-Emitter-Sättigungsspannung von 0,8 V wirkt sich bei der Beschaltung des Optokopplers nicht aus, da die Eingangsspannung für den nachgeschalteten Baustein 74LS373 am Emitterwiderstand abgegriffen wird.

## 5. A/D-Wandler (IC12)

Der A/D-Wandler setzt einen analogen Eingangsspannungswert, der zwischen den Anschlüssen "Analog IN" und "GND" anliegt, in ein 8Bit-Datenwort um. Für die Umwandlung benötigt der Baustein ein Taktsignal, das aus dem Mikroprozessortakt (2 MHz) des Einplatinencomputers erzeugt wird und eine Frequenz von 500 kHz hat. Die Frequenzteilung erfolgt durch IC11. Für eine Umwandlung eines Eingangsspannungswertes in ein entsprechendes Datenwort benötigt der A/D-Wandler zehn Taktzyklen, was einer Wandlungszeit von 18  $\mu$ s entspricht. Außerdem wird neben der positiven Betriebsspannung eine negative Spannung benötigt. Um den Anschluß einer externen Betriebsspannung zu sparen, ist auf der Karte ein Spannungswandler eingesetzt.

### 5.1. Spannungs-Zeit-Diagramm für die A/D-Wandlung

Für den Beginn der Umwandlung benötigt der A/D-Wandler einen Startimpuls am PIN 4 (SC = Start Conversion). Mit der darauf folgenden negativen Taktflanke beginnt die Umwandlung, welche nach neun Taktzyklen beendet ist. Das so erzeugte 8Bit-Datenwort wird intern gespeichert und kann nur dann gelesen werden, wenn der EN-Anschluß H-Pegel führt, da die Ausgänge ansonsten hochohmig bleiben. Der Startimpuls wird durch das ~~TOW~~<sup>TOR</sup>-Signal des Mikroprozessors erzeugt (OUT-Befehl). Nach einer Wartezeit von 18  $\mu$ s nimmt der ~~BUSY~~<sup>TOR</sup>-Anschluß des Bausteins H-Pegel an, wodurch angezeigt wird, daß die Umwandlung beendet ist. Mit Hilfe des ~~TOR~~<sup>TOR</sup>-Signals (IN-Befehl) wird der EN-Anschluß mit H-Pegel beschaltet, wodurch daß 8-Bit-Datenwort gelesen wird.

Bild 4 zeigt den beschriebenen Vorgang im Spannungs-Zeit-Diagramm.

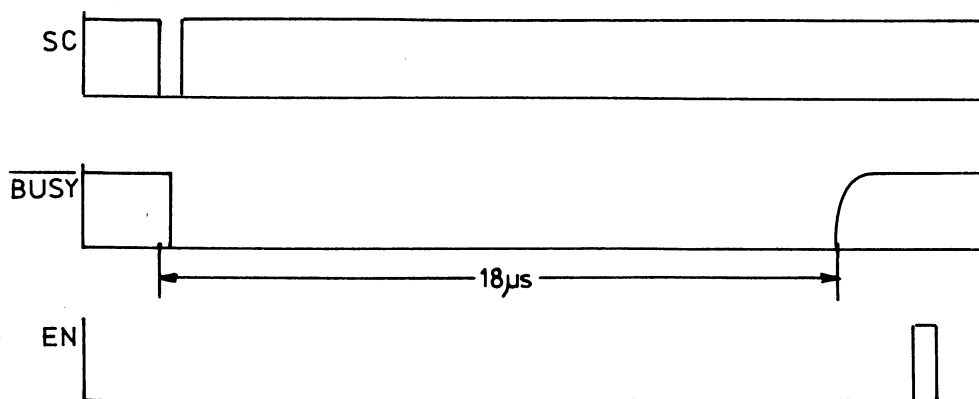


Bild 4: Spannungs-Zeit-Diagramm für die A/D-Wandlung

## EPC-Ein/Ausgabe

## 5.2. Erzeugung der negativen Betriebsspannung

Die negative Betriebsspannung wird durch die Transistoren  $V_1$  und  $V_2$  erzeugt. Führt das Taktsignal des A/D-Wandlers L-Pegel, dann ist  $V_1$  leitend und lädt den Kondensator  $C_2$  auf einen Spannungswert auf, der sich nach der Formel  $U_C = U_B - U_{FV3} - U_{CE1}$  berechnet. Führt das Taktsignal H-Pegel, so ist  $V_2$  leitend. Die Diode  $V_3$  verhindert eine Entladung des Kondensators über den leitenden Transistor. Gleichzeitig ist  $V_4$  leitend und der Kondensator  $C_3$  wird auf eine Spannung von  $U_{C3} = - (U_{C2} - U_{FV4} - U_{CE2})$  aufgeladen. Die so erzeugte Spannung hat einen Wert von ca. - 3,5 Volt.

## 6. TTL-kompatible Ausgabe (IC8)

Die TTL-Ausgabe erfolgt ebenfalls über D-Flip-Flops. Hier sind aber gegenüber der Eingabe keine Tri-State-Ausgänge erforderlich. Die Flip-Flops können allerdings über den Eingang R zurückgesetzt werden. Außerdem sind sie im Gegensatz zur TTL-Eingabe nicht zustandsgesteuert, sondern die Eingangsinformation (vom Datenbus) wird mit der positiven Taktflanke in die D-Flip-Flops übernommen. Tabelle 4 stellt die Wahrheitstafel dieses Bausteins dar.

CLK	R	D	Q
L	L	X	L
L	H	X	$t_{n-1}$
H	H	H	H
H	H	L	L

Tabelle 4: Wahrheitstabelle 74LS273

## 7. Ausgabe über Optokoppler (IC4 und IC5)

Für die Ausgabe werden ebenfalls vierfach-Optokoppler verwendet. Es handelt sich dabei um Bausteine, deren Fototransistoren eine möglichst geringe Kollektor-Emitter-Sättigungsspannung aufweisen. Die Betriebsspannung für die Transistoren wird über die Anschlüsse EXT.V<sub>CC</sub> und EXT.GND1 zugeführt. Die Fotodioden werden bei L-Pegel am Ausgang von IC7 (74LS273) leitend, wodurch die Fototransistoren ebenfalls leitend werden, was an den Optokoppler-Ausgängen L-Pegel zur Folge hat. Bild 5 zeigt die Beschaltung eines Optokopplers. Für diese Optokoppler wurde ein Typ mit einem Stromübertragungsverhältnis von >25 gewählt.

## EPC-Ein/Ausgabe

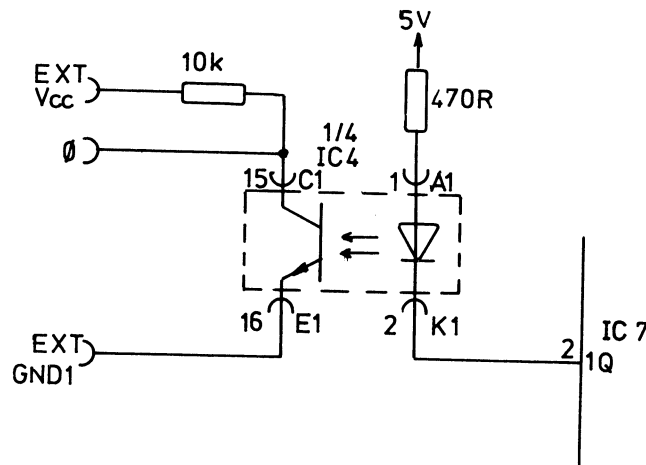


Bild 5: Optokoppler-Ausgabe

## 8. D/A-Wandler (IC6)

Der D/A-Wandler setzt ein 8Bit-Datenwort an seinen Eingängen in einen analogen Spannungswert am Ausgang um. Das Datenwort wird in internen D-Flip-Flops gespeichert. Die Übernahme in die D-Flip-Flops erfolgt bei L-Pegel am EN-Eingang. Ein Rücksetzen der Flip-Flops ist nur möglich, wenn das Datenwort 00000000 angelegt wird und der EN-Eingang L-Pegel annimmt.

Um einen möglichst geringen Ausgangswiderstand der Analogausgabe zu erhalten, wurde dem D/A-Wandler ein Operationsverstärker nachgeschaltet, der im nichtinvertierenden Betrieb arbeitet.

Dadurch wird der relativ hohe Ausgangswiderstand des D/A-Wandlers in den geringen Ausgangswiderstand des Operationsverstärkers umgewandelt. Außerdem kann durch die Beschaltung des Operationsverstärkers eine Spannungsverstärkung  $> 1$  eingestellt werden, wodurch die Ausgangsspannung des D/A-Wandlers verstärkt wird.

Bei der D/A-Wandlung muß beachtet werden, daß der Baustein ZN 428 für die Umwandlung eine Zeit von zirka 800 ns benötigt.

Außerdem benötigt der Operationsverstärker eine bestimmte Zeit für die Umsetzung seines Eingangsspannungswertes. Die Bilder auf der nächsten Seite zeigen die Spannungs-Zeit-Diagramme für zwei D/A-Wandlungen. Im Bild 6.1 ändert sich die Ausgangsspannung des D/A-Wandlers (D/A-OUT = IC6 PIN 5) und die Ausgangsspannung des Operationsverstärkers (ANALOG-OUT = IC3 PIN 6) vom kleinsten zum größten Wert, im Bild 6.2 ändern sich die Ausgangsspannungen in umgekehrter Richtung. Aus den Diagrammen ist ersichtlich, daß der Operationsverstärker bei einer Spannungsänderung vom kleinsten zum größten Wert eine zusätzliche Zeitverzögerung erzeugt.

## EPC-Ein/Ausgabe

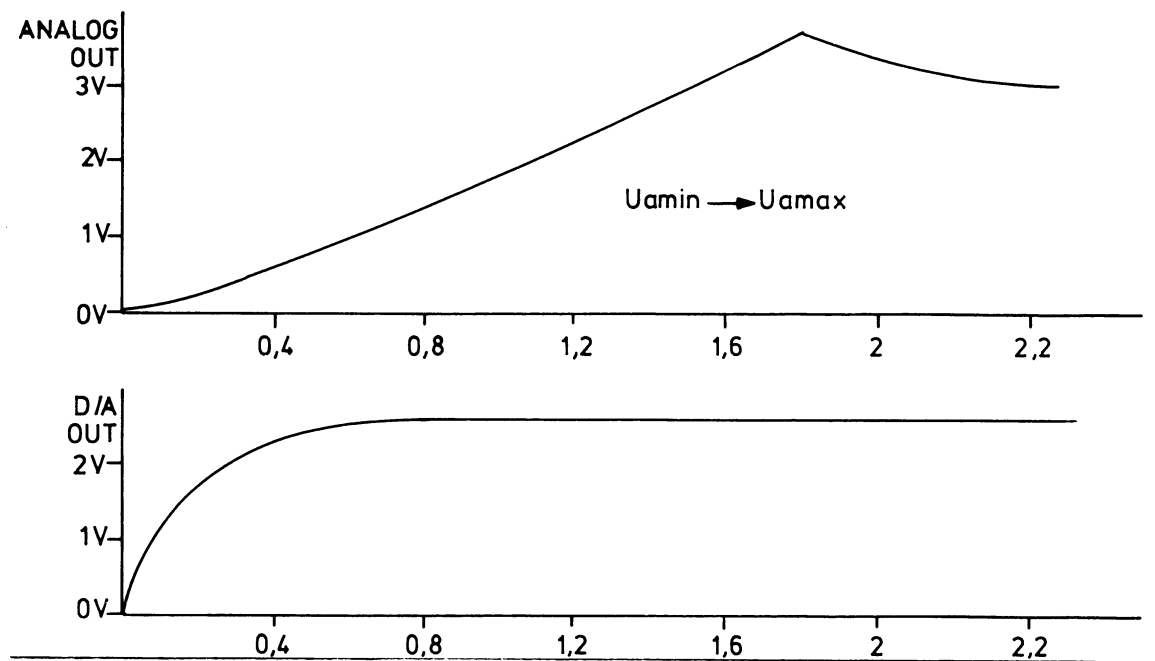


Bild 6.1

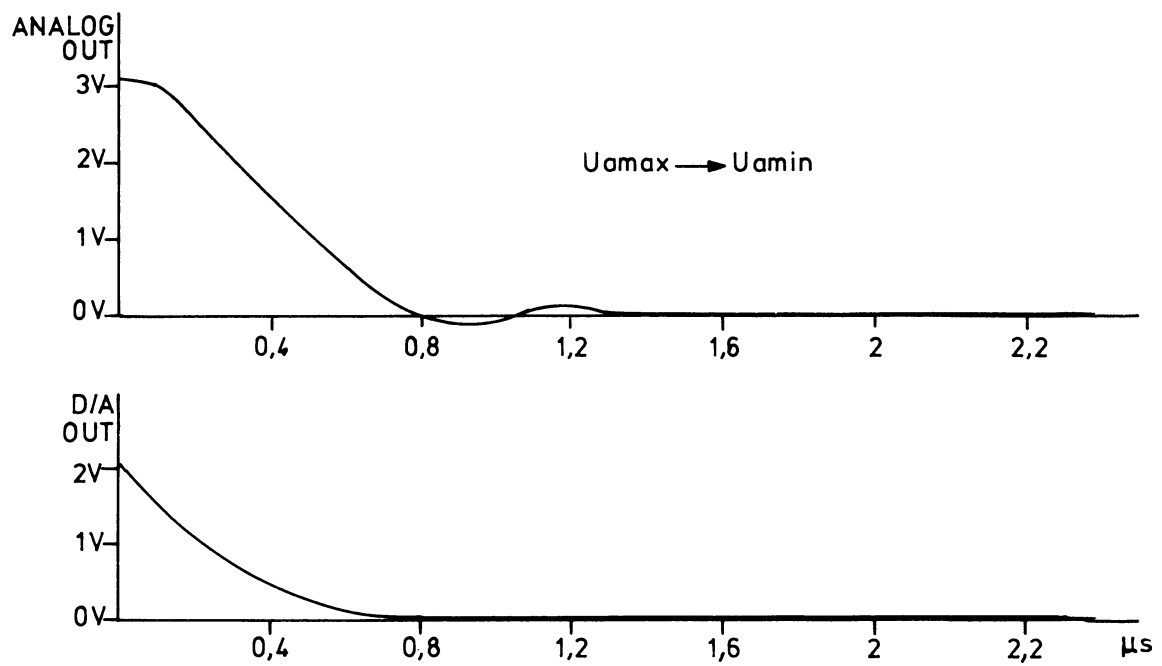
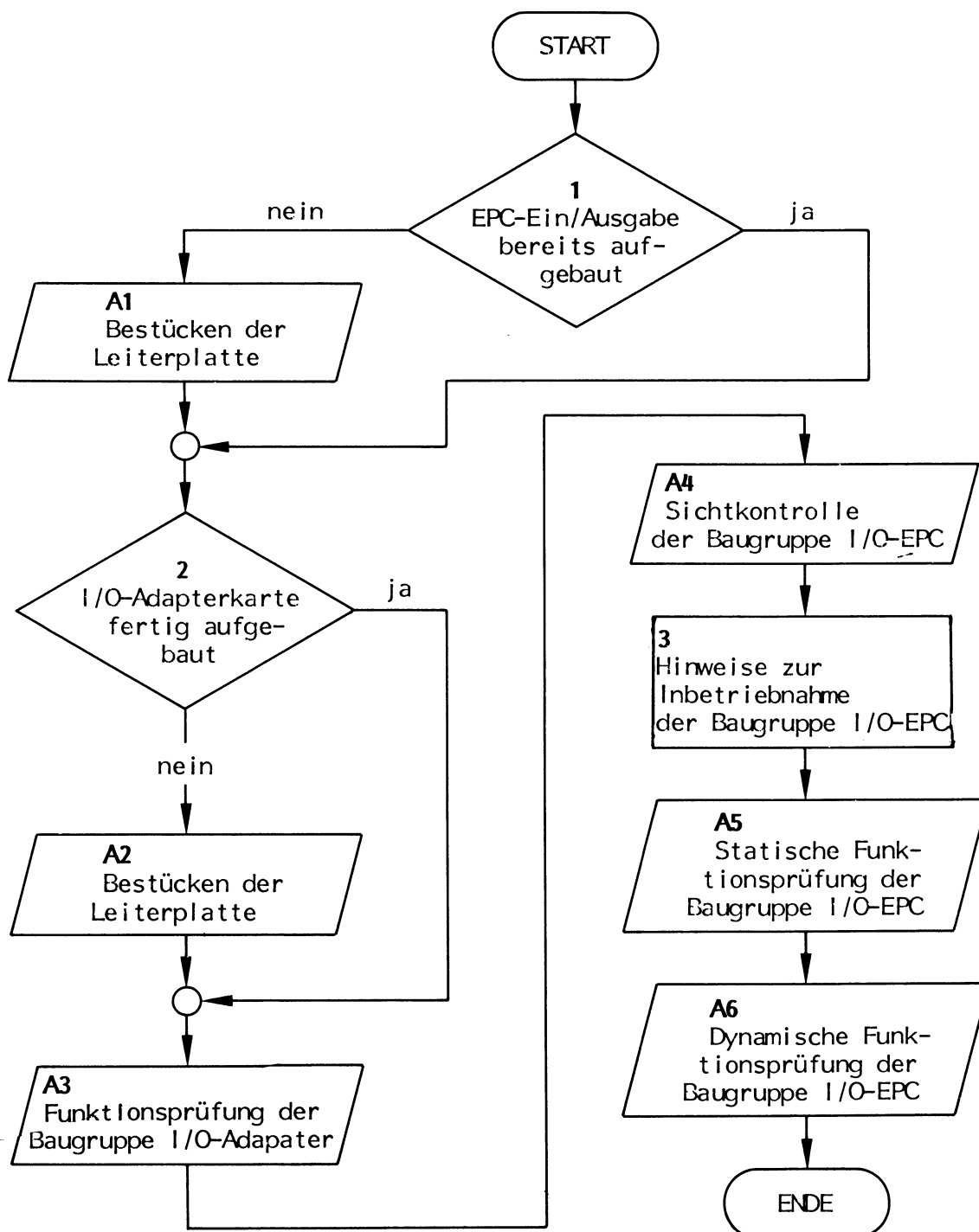


Bild 6.2

## Flußdiagramm für den Arbeitsablauf

Hinweis: Auf den folgenden Seiten wird die EPC-Ein/Ausgabe kurz als **I/O-EPC** bezeichnet. Die zur Inbetriebnahme erforderliche Adapterkarte als **I/O-Adapter!**



## EPC-Ein/Ausgabe

Stckz.	Benennung/Daten	Bemerkung
1	fertig geätzte, durchkontaktierte Epoxid-Glashartgewebeplatte mit der Bezeichnung I/O-EPC	
1	Federleiste 64polig, DIN 41612 mit abgewinkelten Anschlüssen für Platinenmontage	z.B. Panduit Nr. 100-964-553
2	Zylinderschraube M2,5x10 DIN 84	
2	Federscheibe A2,7 DIN 137	
2	Sechskantmutter B2,5 DIN 127	
1	Widerstandsnetzwerk 8x220 Ohm in 16poligem DIL-Gehäuse Gesamtbelastung 1,75 W	alle Widerstände einzeln herausgeführt
2	Widerstand 390 Ohm/0,25 W	
1	Widerstandsnetzwerk 8x470 Ohm für stehende Montage Gesamtbelastung 0,9 W	RM 2,56 mm mit einem gemeinsamen Anschluß
1	Widerstand 1 kOhm/0,25 W	
1	Widerstandsnetzwerk 8x1 kOhm für stehende Montage Gesamtbelastung 0,9 W	RM 2,56 mm mit einem gemeinsamen Anschluß
1	Widerstand 1,8 kOhm/0,25 W	
1	Widerstand 5,6 kOhm/0,25 W	
1	Widerstand 6,8 kOhm/0,25 W	
1	Widerstand 8,2 kOhm/0,25 W	
1	Widerstand 10 kOhm/0,25 W	
1	Widerstandsnetzwerk 8x10 kOhm für stehende Montage Gesamtbelastung 0,9 W	RM 2,56 mm mit einem gemeinsamen Anschluß
1	Widerstand 15 kOhm/0,25 W	
1	Widerstand 22 kOhm/0,25 W	

---&gt;



## EPC-Ein/Ausgabe

Stckz.	Benennung/Daten	Bemerkung
1	Widerstand 56 kOhm/0,25 W	
1	Widerstand 680 kOhm/0,25 W	
1	Trimmpoti 5 kOhm/0,2 W	stehende Montage
2	Trimmpoti 10 kOhm/0,2 W	stehende Montage
1	Trimmpoti 1 MOhm/0,2 W	stehende Montage
3	Tantal-Elko 1 uF/16 V	Tropfenform
4	Tantal-Elko 10 uF/16 V	Tropfenform
1	Tantal-Elko 10 uF/35 V	Tropfenform
2	ker. Scheibenkondensatoren 100 pF/63 V	
2	ker. Scheibenkondensatoren 10 nF/63 V	
8	JFET BF 245 C	
1	NPN Transistor BCY 58	
1	PNP Transistor BCY 78	
2	Si-Diode 1N4148 o.ä.	
2	IC 74LS04	
2	IC 74LS27	
2	IC 74LS273	
2	IC 74LS373	
1	IC 74LS73	
2	IC ILQ 1, 4fach Optokoppler	Fa. Siemens
2	IC ILQ 30, 4fach Optokoppler	Fa. Siemens
1	IC ZN 427, A/D-Wandler	
1	IC ZN 428, D/A-Wandler	
1	IC TL 061, Operationsverstärker mit JFET-Eingang	
1	IC-Fassung, 8polig, DIL	--->

## EPC-Ein/Ausgabe

Stckz.	Benennung/Daten	Bemerkung
5	IC-Fassung, 14polig, DIL	
6	IC-Fassung, 16polig, DIL	
1	IC-Fassung, 18polig, DIL	
4	IC-Fassung, 20polig, DIL	
36	Lötnägel für Bohrung 1,3 mm	
8	Lötfedern für 1,3 mm	z.B. Stocko Nr.:RF7796.004
n.B.	Löt draht	

## Bereitstellungsliste I/O-Adapterkarte

Stckz.	Benennung/Daten	Bemerkung
1	fertiggeätzte, durchkontaktierte Epoxid Glashartgewebeplatte mit der Bezeichnung <b>I/O-Adapter</b>	
2	Messerleiste 64pol., DIN 41612	z.B. Erni STV-P-364a/c Nr.9722.333.401
4	Zylinderschraube M2,5x10 DIN 84	
4	Federscheibe A2,7 DIN 137	
4	Sechskantmutter M2,5 DIN 439	

## EPC-Ein/Ausgabe

1

In dieser Übung werden Sie die Ein/Ausgabekarte zum Einplatinencomputer aufbauen und in Betrieb nehmen. Falls Sie bereits eine fertig aufgebaute Karte erhalten haben, besteht Ihre Aufgabe darin, sie zu prüfen und in Betrieb zu nehmen.

Entscheiden Sie nun, wie Sie vorgehen:

Aufbau der Ein/Ausgabeplatine nach Arbeitsunterlagen ---> A1

Aufbau der I/O-Adapterplatine ---> 2

---

2

In dieser Übung werden Sie die I/O-Adapterkarte zur Ein/Ausgabekarte aufbauen und überprüfen. Falls Sie bereits eine fertig aufgebaute Karte erhalten haben, besteht Ihre Aufgabe darin, sie zu überprüfen.

Entscheiden Sie nun, wie Sie vorgehen:

Aufbau der I/O-Adapterplatine nach Arbeitsunterlagen ---> A2

Funktionsprüfung der fertigen Baugruppe ---> A3

---

3

Mit Hilfe der folgenden Seiten werden Sie die Baugruppe I/O-EPC Betrieb nehmen. Die Karte wird mit Hilfe des MFA-Systems statisch und dynamisch in Betrieb genommen.

--->

## 3.1

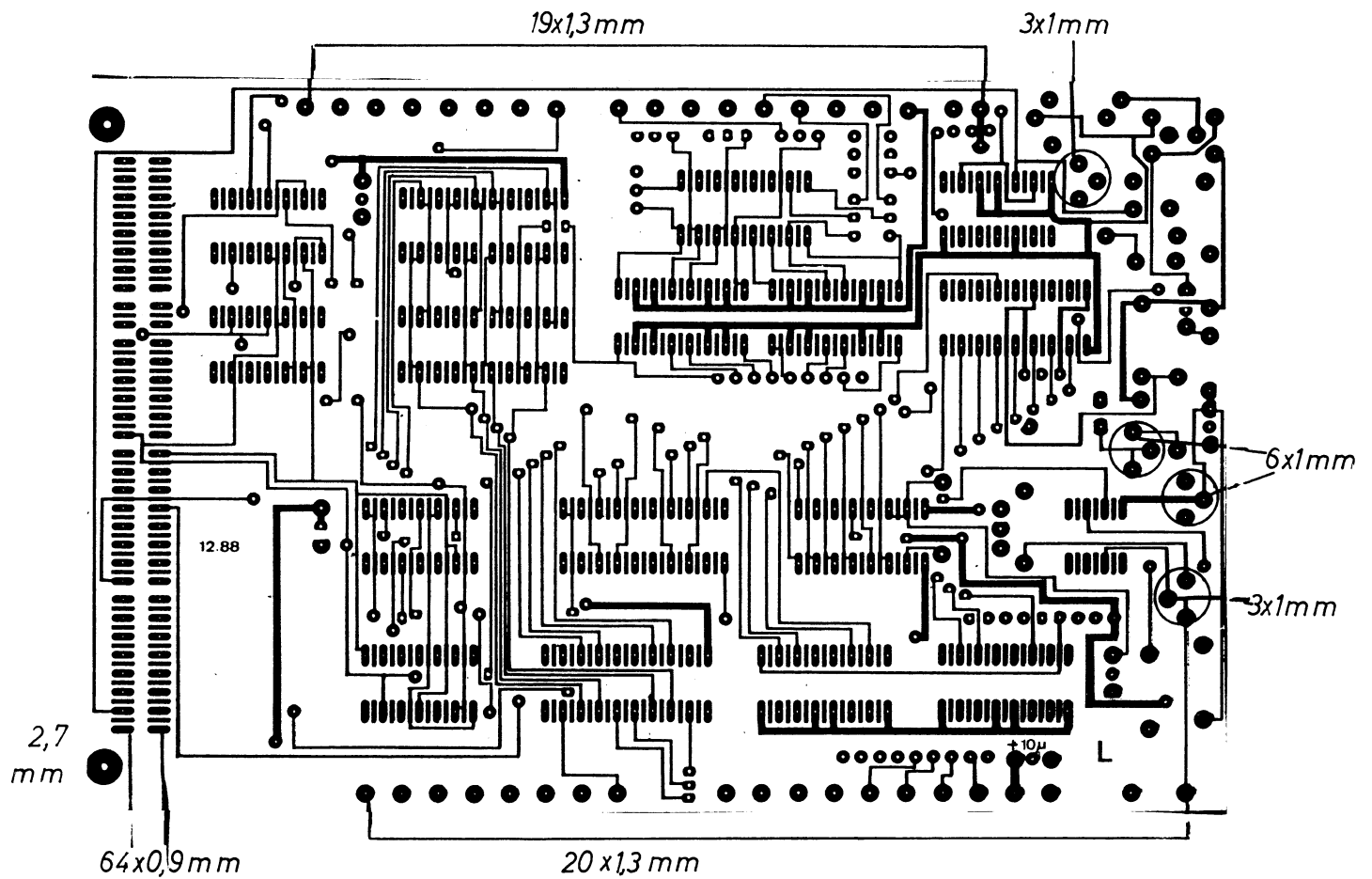
Für die Inbetriebnahme der Ein/Ausgabekarte benötigen Sie ein MFA-System mit Netzteil und Spannungsregelung  
Außerdem benötigen Sie:

- 1 Prozessorkarte CPU 8085
  - \* - 1 8 KByte EPROM mit MAT 85 (Basisadresse 0000)
  - \* - 1 8 KByte RAM (Basisadresse E000)
  - 1 Bus-Signalgeber
  - 1 Bus-Signalanzeige
  - 1 8-Bit-Parallelausgabe
  - 1 Video-Interface
  - 1 Tastatur
  - 1 Monitor
  - 1 Gleichspannungsquelle einstellbar 0-15 V/500 mA
  - 1 I/O-Adapterkarte
  - 1 Oszilloskop zweikanalig, 15 MHz
  - 2 BNC-Meßleitungen mit Tastkopf 10:1
  - 8 Leitungen aus isol. Schaltdraht  $\varnothing$  0,8mm  
beidseitig mit Lötfedern 1,3 mm versehen
  - 1 Vielfachmeßinstrument
  - 1 IC Meßadapter 16polig
  - 1 IC Meßadapter 18polig
  - 1 Logiktester
- 
- \* ersatzweise 64K-RAM-Karte mit MAT 32K

A1.1

Für die Ein/Ausgabekarte zum Einplatinencomputer benötigen Sie eine zweiseitig kupferkaschierte und durchkontaktierte Leiterplatte mit der Bezeichnung I/O-EPC

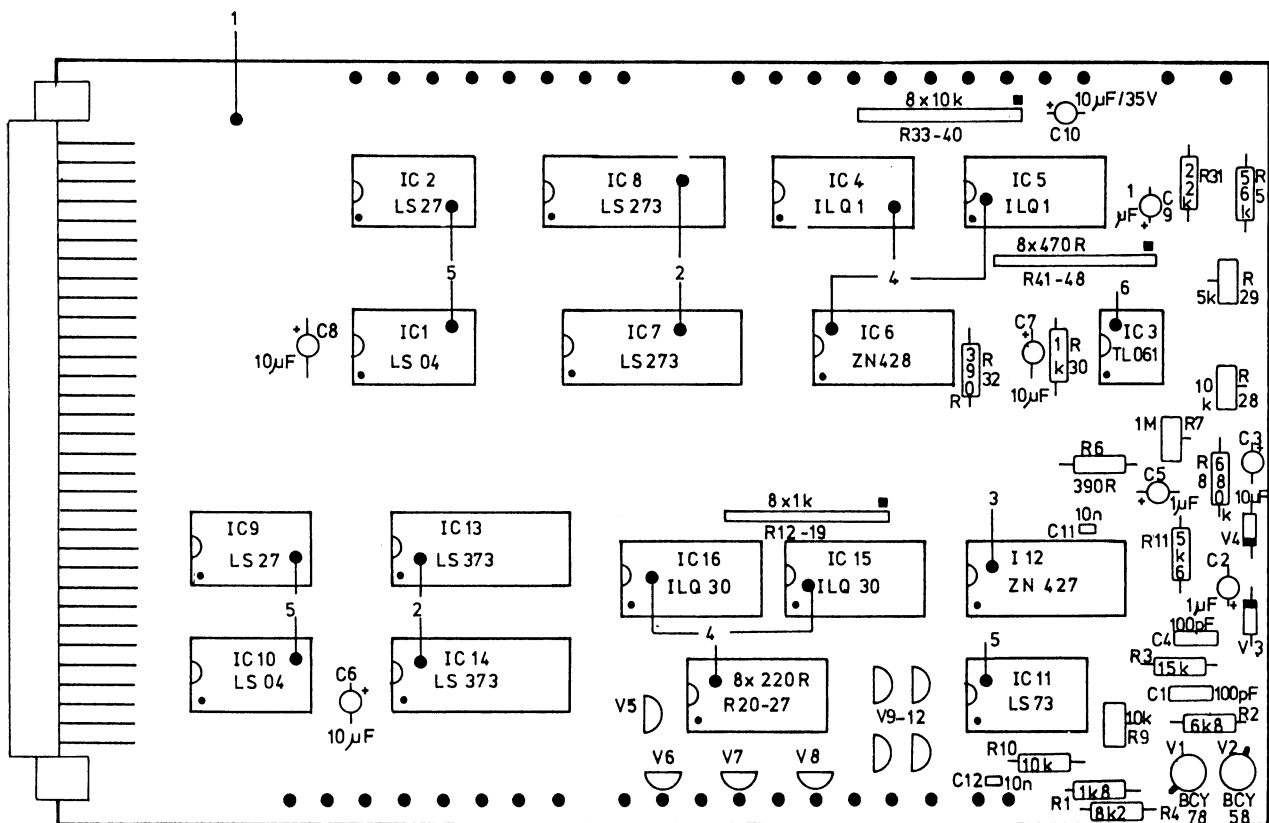
Leiterbahnseite



---&gt;



## BESTÜCKUNGSPLAN Ein/Ausgabekarte



Bestücken Sie die Leiterplatte mit Hilfe des Bestückungsplans, der Stückliste und der Bauteilliste.  
Vorher sollten Sie die Leiterplatte mit Hilfe einer Lupe auf Risse und Kurzschlüsse untersuchen und Fehler beseitigen.

## EPC-Ein/Ausgabe

## A1.4

## Stückliste Ein/Ausgabekarte

Pos.	Stckz.	Benennung/Daten	Bemerkung
1	1	Leiterplatte EPC-I/O	
2	4	IC-Fassung 20polig	
3	1	IC-Fassung 18polig	
4	6	IC-Fassung 16polig	
5	5	IC-Fassung 14polig	
6	1	IC-Fassung 8polig	

## Bauteilliste Ein/Ausgabekarte

Kennz.	Benennung/Daten	Bemerkung
R1	Widerstand 1,8 kOhm/0,25 W	
R2	Widerstand 6,8 kOhm/0,25 W	
R3	Widerstand 15 kOhm/0,25 W	
R4	Widerstand 8,2 kOhm/0,25 W	
R5	Widerstand 56 kOhm/0,25 W	
R6	Widerstand 390 Ohm/0,25 W	
R7	Trimpoti 1 MOhm/0,2 W	stehende Mont.
R8	Widerstand 680 Kohm/0,25 W	
R9	Trimpoti 10 kOhm/0,2 W	stehende Mont.
R10	Widerstand 10 kOhm/0,25 W	
R11	Widerstand 5,6 kOhm/0,25 W	
R12-R19	Widerstandsnetzwerk 8x1 kOhm	stehende Mont.
R20-R27	Widerstandsnetzwerk 8x220 Ohm	DIL-Gehäuse
R28	Trimpoti 10 kOhm/0,2 W	stehende Mont.
R29	Trimpoti 5 kOhm/0,2 W	stehende Mont.
R30	Widerstand 1 kOhm/0,25 W	
R31	Widerstand 22 kOhm/0,25 W	--->



## A1.5

## Bauteilliste Ein/Ausgabekarte

Kennz.	Benennung/Daten	Bemerkung
R32	Widerstand 390 Ohm/0,25 W	
R33-R40	Widerstandsnetzwerk 8x10 kOhm	stehende Mont.
R41-R48	Widerstandsnetzwerk 8x470 Ohm	stehende Mont.
C1	ker. Scheibenkondensator 100 pF	
C2	Tantal-Elko 1 uF/16 V	Tropfenform
C3	Tantal-Elko 10 uF/16 V	Tropfenform
C4	ker. Scheibenkondensator 100 pF	
C5	Tantal-Elko 1 uF/16 V	Tropfenform
C6-C8	Tantal-Elko 10 uF/16 V	Tropfenform
C9	Tantal-Elko 1 uF/16 V	Tropfenform
C10	Tantal-Elko 10 uF/35 V	Tropfenform
C11,C12	ker. Scheibenkondensator 10 nF	
V1	PNP-Transistor BCY 78	
V2	NPN-Transistor BCY 58	
V3-V4	Si-Diode 1N4148	
V5-V12	JFET BF 245 C	
IC1	74LS04	
IC2	74LS27	
IC3	TL 061, OPV	
IC4-IC5	ILQ 1, 4fach-Optokoppler	
IC6	ZN 428, D/A-Wandler	IC's nicht ein- gesteckt
IC7-IC8	74LS273	
IC9	74LS27	
IC10	74LS04	
IC11	74LS73	

---&gt;

A1.6

## Bauteilliste Ein/Ausgabekarte

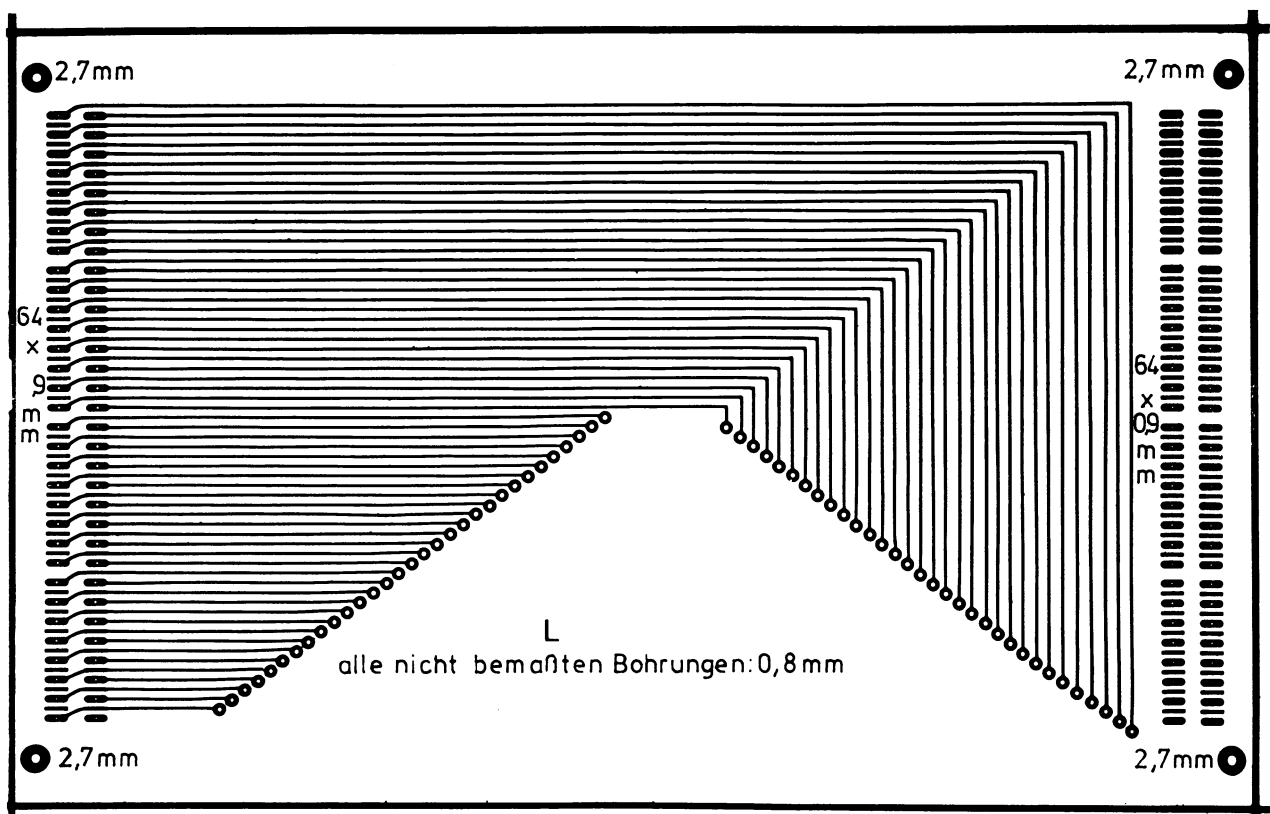
Kennz.	Benennung/Daten	Bemerkung
IC12	ZN 427, A/D-Wandler	
IC13-IC14	74LS373	
IC15-IC16	1LQ 30, 4fach-Optokoppler	IC's nicht eingesteckt

---&gt; 2

A2.1

Für die I/O-Adapterkarte benötigen Sie eine zweiseitig kupferkaschierte und durchkontaktierte Leiterplatte mit der Bezeichnung **I/O-Adapter**.

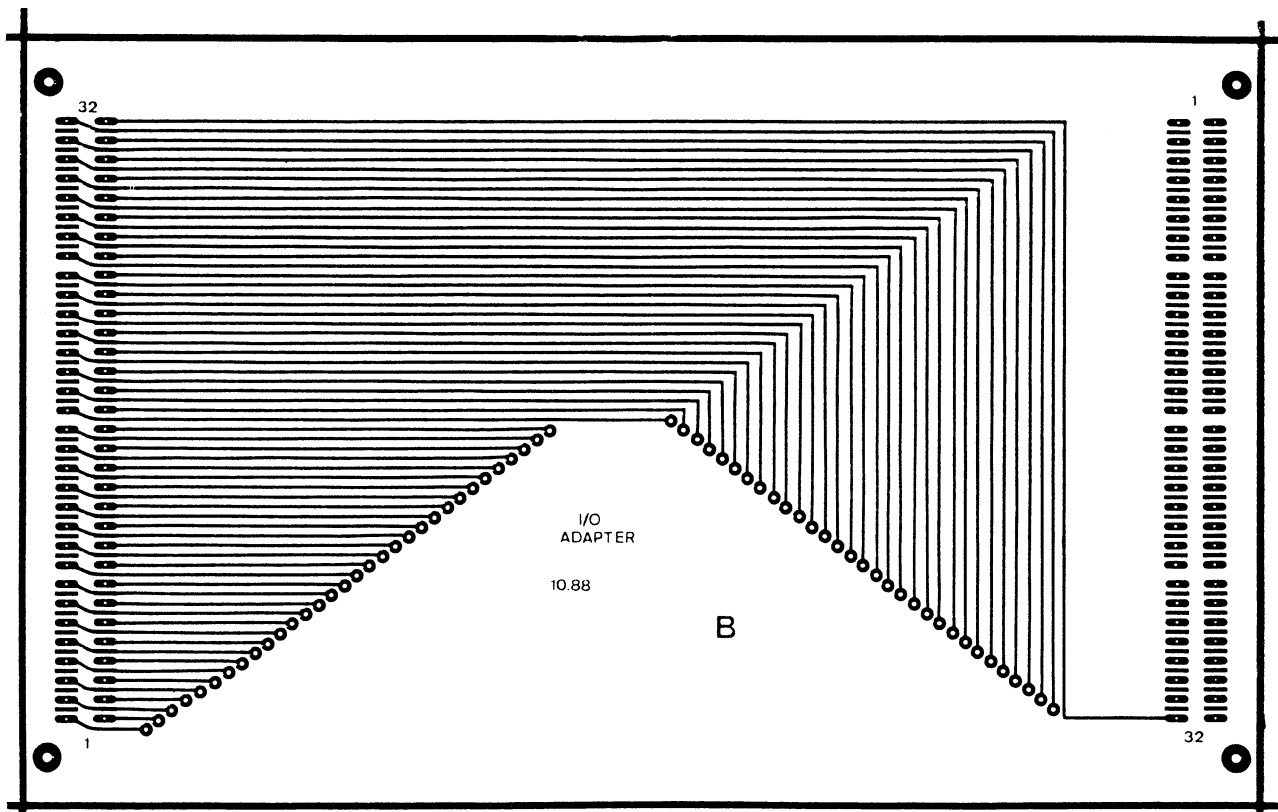
Leiterbahnseite



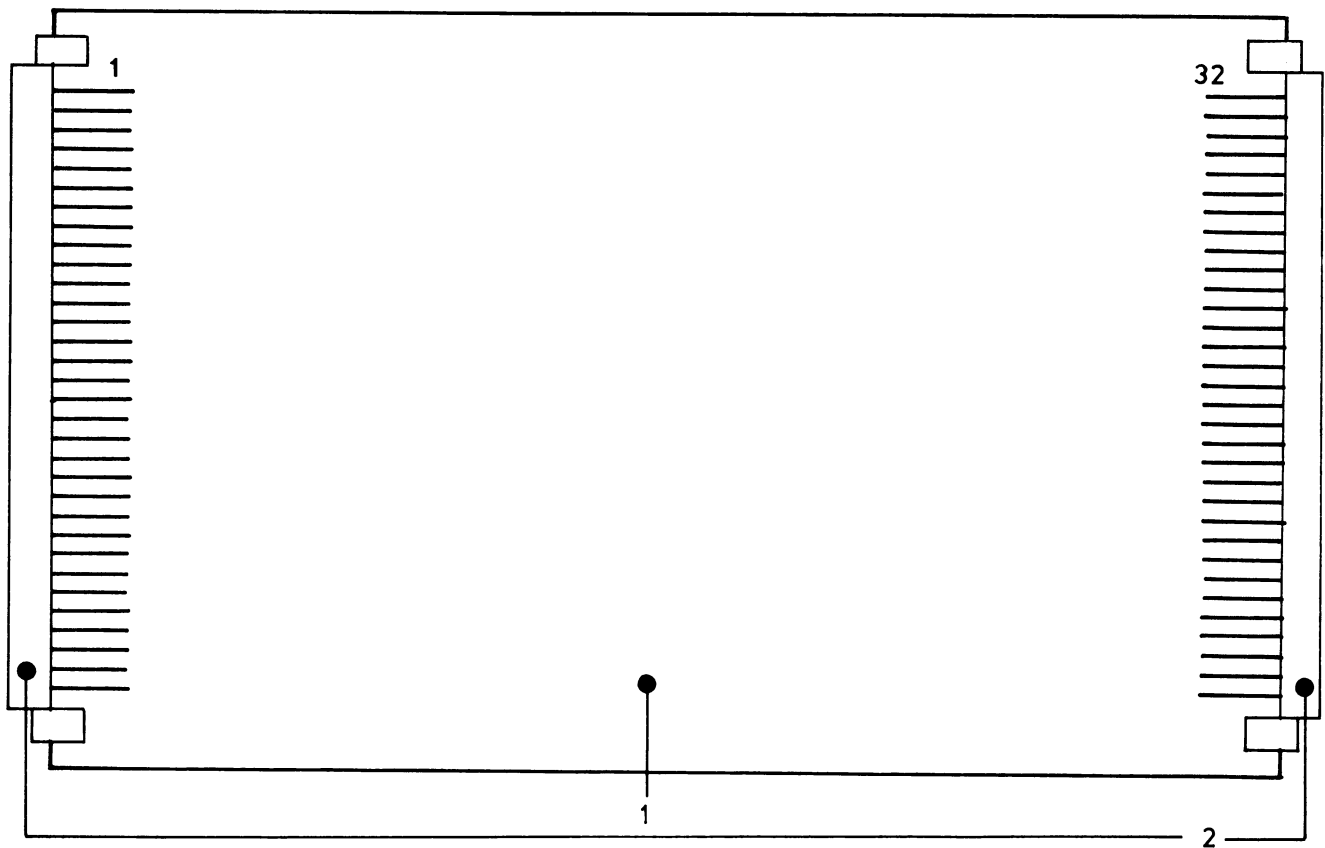
---&gt;

## A2.2

Die folgende Abbildung zeigt das Layout der Bestückungsseite der I/O-Adapterplatine



## BESTÜCKUNGSPLAN I/O-Adapterkarte



Bestücken Sie die Leiterplatte der I/O-Adapterkarte mit Hilfe des Bestückungsplans und der Stückliste.  
Vorher sollten Sie die Leiterplatte mit einer Lupe auf Risse und Kurzschlüsse untersuchen und Fehler beseitigen.

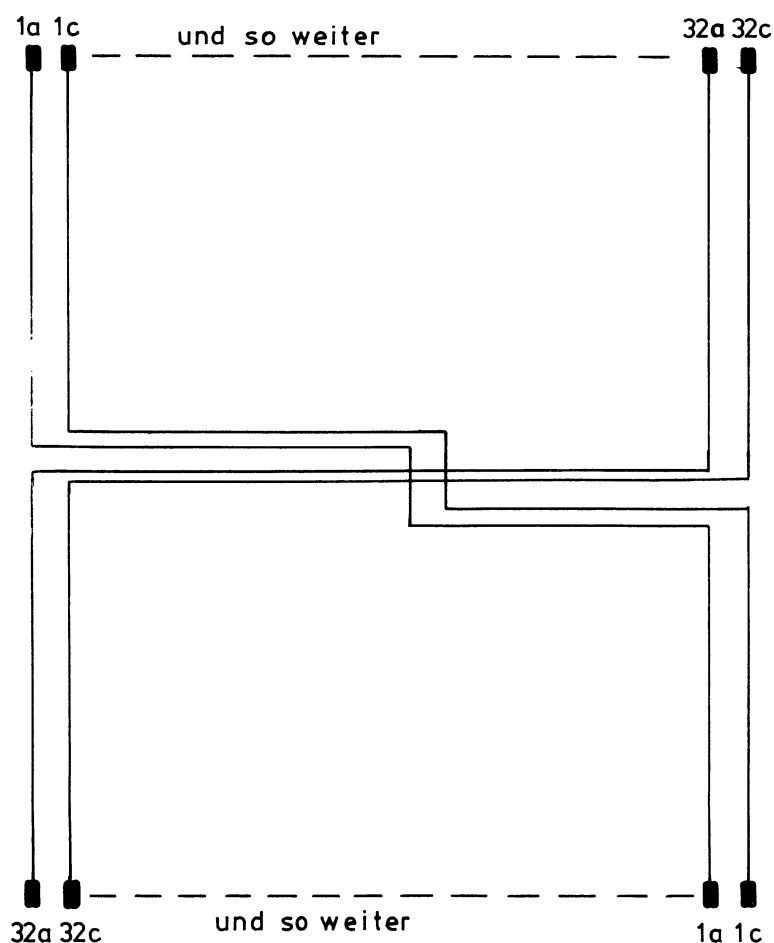
## Stückliste I/O-Adapterkarte

Pos.	Stckz.	Benennung/Daten
1	1	Leiterplatte I/O-Adapterkarte
2	2	Messerleiste 64polig

## A3.1

Zur Funktionsprüfung der I/O-Adapterkarte benötigen Sie die auf dieser Seite abgebildete Schemazeichnung. Kontrollieren Sie vor der Funktionsprüfung die Leiterplatte auf Kurzschlüsse, die durch Lötspritzer entstanden sein können.

## SCHEMAZEICHNUNG Adapterkarte



Für die Funktionsprüfung der I/O-Adapterkarte benötigen Sie einen Durchgangsprüfer mit akustischer Anzeige. Kontrollieren Sie jetzt die einzelnen Leiterbahnen zwischen den entsprechenden Anschlüssen der Messerleisten auf Durchgang. Danach sollten Sie jeweils zwei benachbarte Leiterbahnen auf Kurzschluß überprüfen.

## A4.1

## Sichtkontrolle der Ein/Ausgabekarte

Für die Sichtkontrolle sowie für die Inbetriebnahme muß der Stromlauf- und Bestückungsplan bereitliegen. Kontrollieren Sie alle Lötstellen auf der Leiterbahnseite und achten Sie auf Kurschlüsse, die beim Löten eventuell zwischen den Leiterbahnen entstanden sein könnten.

Achten Sie bei der Kontrolle der Bauteile auf folgende Punkte:

- Sind alle Widerstände mit ihrem richtigen Wert eingebaut?
- Sind alle Elko's richtig gepolt?
- Der Elko C10 muß eine Spannungsfestigkeit von **35 Volt** haben
- Die IC's dürfen nicht gesteckt sein

---> 3

## A5.1

Bestücken Sie das MFA-System, welches mit Netzteil und Spannungsregelung ausgerüstet sein muß, mit folgenden Einschüben:

- Bus-Signalgeber
- Bus-Signalanzeige
- I/O-Adapterkarte
- Prozessorkarte CPU 8085

Stecken Sie die Ein/Ausgabekarte auf den Adapter und schalten Sie die Betriebsspannung des Systems ein.

Kontrollieren Sie jetzt die Spannungsversorgung an allen IC-Sockeln und tragen Sie Ihre Meßergebnisse in die Tabelle auf der nächsten Seite ein.

Der Schalter "HLT/RUN" der Bus-Signalanzeige muß auf "HLT" stehen!

--->

## A5.2

Meßwerttabelle zur Überprüfung der Betriebsspannung

AUSGABEKANÄLE

	IC1	IC2	IC3	IC4	IC5	IC6	IC7	IC8
TYP								
positive Versorgungsspannung an PIN Nr.								
Masse an PIN Nr.								
$U_B/V$								
GND								

EINGABEKANÄLE

	IC9	IC10	IC11	IC12	IC13	IC14	IC15	IC16
TYP								
positive Versorgungsspannung an PIN Nr.								
Masse an PIN Nr.								
$U_B/V$								
GND								

Wenn die Spannungsversorgung an allen IC-Sockeln richtig ist, können Sie das MFA-System ausschalten und die Ein/Ausgabekarte mit allen ICs bestücken.

--->

## A5.3

Der Operationsverstärker (IC3) und der A/D-Wandler (IC12) benötigen neben der positiven auch eine negative Versorgungsspannung. Überprüfen Sie deshalb mit Hilfe eines Oszilloskops die Erzeugung der negativen Betriebsspannung. Machen Sie sich vorher die Funktionsweise der Schaltung mit Hilfe von Seite 9 noch einmal klar. Tragen Sie Ihre Meßergebnisse in die Koordinatensysteme auf dieser Seite ein.

Messen Sie folgende Spannungen:

$$IC11 \text{ PIN } 12 = f(t)$$

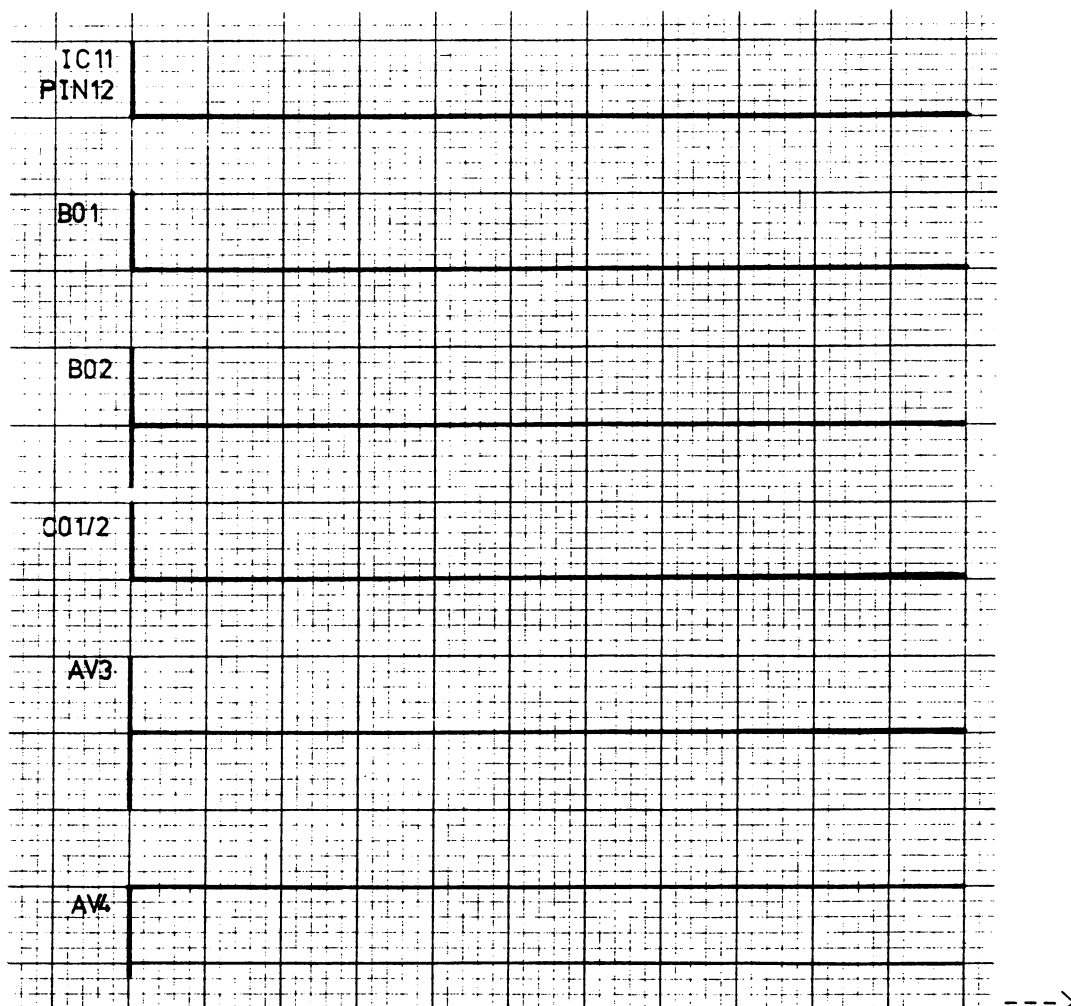
$$U_{B01} = f(t)$$

$$U_{B02} = f(t)$$

$$U_{CO1/2} = f(t)$$

$$U_{AV3} = f(t)$$

$$U_{AV4} = f(t)$$





## A5.4

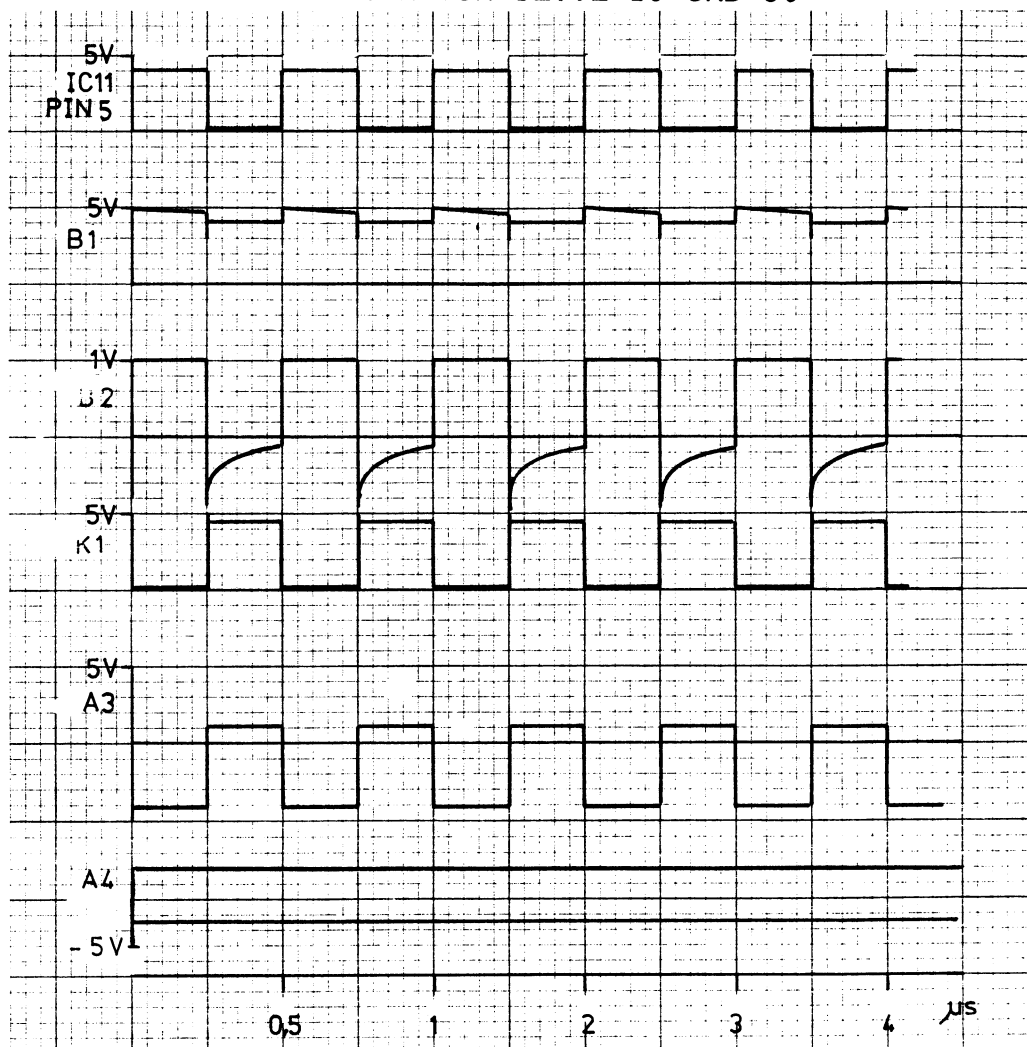
Messen Sie jetzt mit einem Vielfachmeßinstrument die Spannung an folgenden Punkten:

Minuspol von C3: U= \_\_\_\_\_

IC12 PIN5 : U= \_\_\_\_\_

IC3 PIN4 : U= \_\_\_\_\_

### ERGEBNISSE ZU DEN MESSUNGEN VON SEITE 29 UND 30



### HINWEIS ZU DEN MESSUNGEN

Die Triggerung des Oszilloskops erfolgt auf der positiven Flanke der Spannung an IC11 PIN12.

Die Gleichspannungen müssen ungefähr folgende Werte aufweisen:

Minuspol von C3= -3,3V bis -3,8V      IC12 PIN5= -300mV bis -800mV  
 IC3 PIN4= -3,3V bis -3,8V

--->

## A5.5

Wenn die vorhergehenden Messungen keine Fehler aufwiesen, können Sie jetzt die Adreß- und die Datenleitungen überprüfen. Entfernen Sie dazu nach Abschalten der Betriebsspannung die Prozessorkarte CPU 8085 aus dem MFA-System.

Auf der Ein/Ausgabekarte werden nur die Adreßleitungen A0 und A1 benutzt.

Stellen Sie mit Hilfe des Bussignalgebers die in folgender Tabelle angegebenen Adressen ein und überprüfen Sie mit der Bus-Signalanzeige die richtige Übertragung der eingestellten Adreßsignale. Mit einem Logiktester können Sie die Pegel an den einzelnen IC-Anschlüssen messen. Beachten Sie dabei die invertierende Wirkung von IC1 und IC10.

### Meßwerttabelle zur Überprüfung der Adreßleitungen

#### Eingabekanäle

ADRESSE SIGNALGEBER	ADRESSE SIGNALANZEIGE	IC10 PIN				IC9 PIN					
		9	8	11	10	1	13	3	5	9	10
0000											
0001											
0002											

#### Ausgabekanäle

ADRESSE SIGNALGEBER	ADRESSE SIGNALANZEIGE	IC1 PIN				IC2 PIN					
		9	8	13	12	1	13	4	5	11	10
0000											
0001											
0002											

--->

## ERGEBNISSE ZU DEN MESSUNGEN VON SEITE 31

Eingabekanäle

			IC10 PIN				IC9 PIN					
ADRESSE	A1	A0	9	8	11	10	1	13	3	5	9	10
0000	0	0	L	H	L	H	L	L	H	L	L	H
0001	0	1	H	L	L	H	H	L	L	L	H	H
0002	1	0	L	H	H	L	L	H	H	H	L	L

Ausgabekanäle

			IC1 PIN				IC2 PIN					
ADRESSE	A1	A0	9	8	13	12	1	13	4	5	11	10
0000	0	0	L	H	L	H	L	L	H	L	L	H
0001	0	1	H	L	L	H	H	L	L	L	H	H
0002	1	0	L	H	H	L	L	H	H	H	L	L

## EPC-Ein/Ausgabe

A5.7

Überprüfung der Datenleitungen

Die Datenleitungen der Ein/Ausgabekarte werden ebenfalls mit Bus-Signalgeber und -anzeige überprüft. Zusätzlich werden Messungen mit dem Logiktester an den einzelnen IC's durchgeführt. Dabei kann nur der interne Datenbus der Karte überprüft werden. Eine Überprüfung der Ein- und Ausgabedaten an den Lötnägeln der Karte erfolgt zu einem späteren Zeitpunkt.

Eingabekanäle

DATEN SIGN. GEB.	DATEN SIGN. ANZ.	IC12 PIN								IC13 PIN								IC14 PIN							
		11	12	13	14	15	16	17	18	19	16	15	12	9	6	5	2	19	16	15	12	9	6	5	2
55																									
AA																									

Ausgabekanäle

DATEN SIGN. GEB.	DATEN SIGN. ANZ.	IC6 PIN								IC7 PIN								IC8 PIN							
		11	12	13	14	15	16	1	2	18	17	14	13	8	7	4	3	18	17	14	13	8	7	4	3
55																									
AA																									

---&gt;

A5.8

## ERGEBNISSE ZU DEN MESSUNGEN VON SEITE 33

Eingabekanäle

DATEN SIGN. GEB.	DATEN SIGN. ANZ.	IC12 PIN								IC13 PIN								IC14 PIN							
		11	12	13	14	15	16	17	18	19	16	15	12	9	6	5	2	19	16	15	12	9	6	5	2
55	55	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H
AA	AA	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L

Ausgabekanäle

DATEN SIGN. GEB.	DATEN SIGN. ANZ.	IC6 PIN								IC7 PIN								IC8 PIN							
		11	12	13	14	15	16	1	2	18	17	14	13	8	7	4	3	18	17	14	13	8	7	4	3
55	55	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H
AA	AA	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L

Überprüfen Sie die Richtigkeit Ihrer Meßergebnisse. Falls bei einigen Messungen keine Übereinstimmung vorhanden ist, beseitigen Sie den entsprechenden Fehler auf der Karte.

---&gt;

Überprüfung der Adreßdecodierung

Wie auf Seite 5 schon beschrieben, ist die Adreßdecodierung fest verdrahtet. Die einzelnen Kanäle haben je nach Typ die gleiche Adresse, wobei die beiden Steuerleitungen  $\overline{\text{TOR}}$  und  $\overline{\text{TOW}}$  die Unterscheidung zwischen Eingabe und Ausgabe ermöglichen.

Für die folgenden Messungen benötigen Sie das MFA-System mit den gleichen Einschüben wie bei den vorhergegangenen Messungen.

**Achten Sie darauf, daß die CPU-Karte nicht im System steckt!**

Adreßdecodierung der Eingabekanäle

Stellen Sie die in der folgenden Tabelle angegebenen Adressen am Bus-Signalgeber ein und notieren Sie die an der Bus-Signalanzeige angezeigte Adresse. Mit einem Logiktester messen Sie an den vorgegebenen Anschlüssen der einzelnen Bausteine. Die Angabe  $\overline{\text{TOR}} = \text{H}$  bedeutet, daß die entsprechende Taste des Bus-Signalgebers nicht gedrückt ist;  $\overline{\text{TOR}} = \text{L}$  bedeutet, daß die entsprechende Taste gedrückt ist.

ADRESSE SIGNAL- GEBER	$\overline{\text{TOR}}$	ADRESSE SIGNAL- ANZEIG.	$\overline{\text{TOR}}$	PIN	IC9 8	IC12 2	IC9 6	IC10 5   6	IC13 1   11	IC9 12	IC10 13   12	IC14 1   11
0000	H											
0000	L											
0001	H											
0001	L											
0002	H											
0002	L											

A5.10

Adreßdecodierung der Ausgabekanäle

Diese Überprüfung erfolgt mit den gleichen Mitteln wie auf Seite 35. Zu beachten ist bei diesem Übungsteil, daß die Steuerleitung TOW und nicht die Steuerleitung TOR aktiviert werden muß.

ADRESSE SIGNL. GEBER	<u>TOW</u>	ADRESSE SIGNL. ANZEIG.	<u>TOW</u>	PIN	IC2 8	IC1 5   6	IC6 4	IC2 6	IC1 3   4	IC7 11	IC2 12	IC1 1   2	IC8 11
0000	H												
0000	L												
0001	H												
0001	L												
0002	H												
0002	L												

---&gt;

## ERGEBNISSE ZU DEN MESSUNGEN VON SEITE 35 UND 36

Eingabekanäle

ADRESSE SIGNAL- GEBER	$\overline{\text{TOR}}$	ADRESSE SIGNAL- ANZEIG.	$\overline{\text{TOR}}$	PIN	IC9 8	IC12 2	IC9 6	IC10 5   6	IC13 1   11	IC9 12	IC10 13   12	IC14 1   11
0000	H	0000	H		L	L	L	L H	H H	L	L H	H H
0000	L	0000	L		L	L	L	L H	H H	H H	L L	L
0001	H	0001	H		L	L	L	L H	H H	L	L H	H H
0001	L	0001	L		L	L	H H	L L	L L	L	L H	H H
0002	H	0002	H		L	L	L	L H	H H	L	L H	H H
0002	L	0002	L		H H		L	L H	H H	L	L H	H H

Ausgabekanäle

ADRESSE SIGNAL. GEBER	$\overline{\text{IOW}}$	ADRESSE SIGNAL. ANZEIG.	$\overline{\text{IOW}}$	PIN	IC2 8	IC1 5   6	IC6 4	IC2 6	IC1 3   4	IC7 11	IC2 12	IC1 1   2	IC8 11
0000	H	0000	H		L	L H	H	L	L H	H	L	L H	H
0000	L	0000	L		L	L H	H	L	L H	H	H H	L L	
0001	H	0001	H		L	L H	H	L	L H	H	L	L H	H
0001	L	0001	L		L	L H	H	H H	L L	L	L	L H	H
0002	H	0002	H		L	L H	H	L	L H	H	L	L H	H
0002	L	0002	L		H H	L L		L	L H	H	L	L H	H



## A6.1

Dynamische Funktionsprüfung

Für die dynamische Funktionsprüfung muß das MFA-System mit folgenden Einschüben bestückt werden:

- Prozessorkarte CPU 8085
- \* - 8 KByte EPROM mit MAT 85 (Basisadresse 0000)
- \* - 8 KByte RAM (Basisadresse E000)
- Video-Interface

Die Einschübe "Bus-Signalgeber" und "Bus-Signalanzeige" können Sie entfernen.

Außerdem benötigen Sie:

- Tastatur
- Monitor

Schalten Sie die Betriebsspannung ab und bestücken Sie das System mit den oben genannten Einschüben.

Verbinden Sie danach auf der Ein-Ausgabekarte die acht Anschlüsse 0 - 7 der TTL-Ausgabe mit den entsprechenden Anschlüssen der TTL-Eingabe.

Danach können Sie die Betriebsspannung wieder einschalten.

Sie müssen jetzt zuerst mit Hilfe des Assemblers ein Prüfprogramm für die TTL-Ein/Ausgabe schreiben. Rufen Sie deshalb den Assembler auf und geben Sie das folgende Programm ab der Adresse E000 ein:

```
ANF:      MVI    A,55          ;Accu mit 55H laden
          MOV    C,A          ;Wert retten
          OUT    00          ;55H über TTL-Ausgabe ausgeben
          IN     00          ;TTL-Eingabe lesen
          CMP    C          ;ausgegebener Wert=gelesener Wert?
          JNZ    STOP        ;NEIN !
          MVI    A,0AA        ;Ja, Accu mit AAH laden
          MOV    C,A          ;Wert retten
          OUT    00          ;AAH über TTL-Ausgabe ausgeben
          IN     00          ;TTL-Eingabe lesen
          CMP    C          ;ausgegebener Wert=gelesener Wert?
          JNZ    STOP        ;NEIN !
          JMP    ANF         ;Ja, zurück zum Anfang
STOP:     HLT                ;Fehler!: Prozessor hält an!
```

- \* -ersatzweise 64K-RAM-Karte mit MAT 32K

--->

## A6.2

Durch das auf Seite 38 aufgelistete Programm werden erstens die beiden Bausteine IC8 und IC14 auf richtige Funktion überprüft. Zweitens werden die Datenleitungen auf Unterbrechung bzw. Kurzschluß überprüft. Starten Sie das Programm mit dem GO-Kommando und messen Sie mit Hilfe des Oszilloskops folgende Signale. Tragen Sie ihre Meßergebnisse in die Koordinatenkreuze auf der nächsten Seite ein.

TTL-Ausgabe

$$U_{A0} = f(t)$$

$$U_{A1} = f(t)$$

$$U_{A2} = f(t)$$

$$U_{A3} = f(t)$$

$$U_{A4} = f(t)$$

$$U_{A5} = f(t)$$

$$U_{A6} = f(t)$$

$$U_{A7} = f(t)$$

$$IC8 \text{ PIN11} = f(t)$$

$$IC10 \text{ PIN12} = f(t)$$

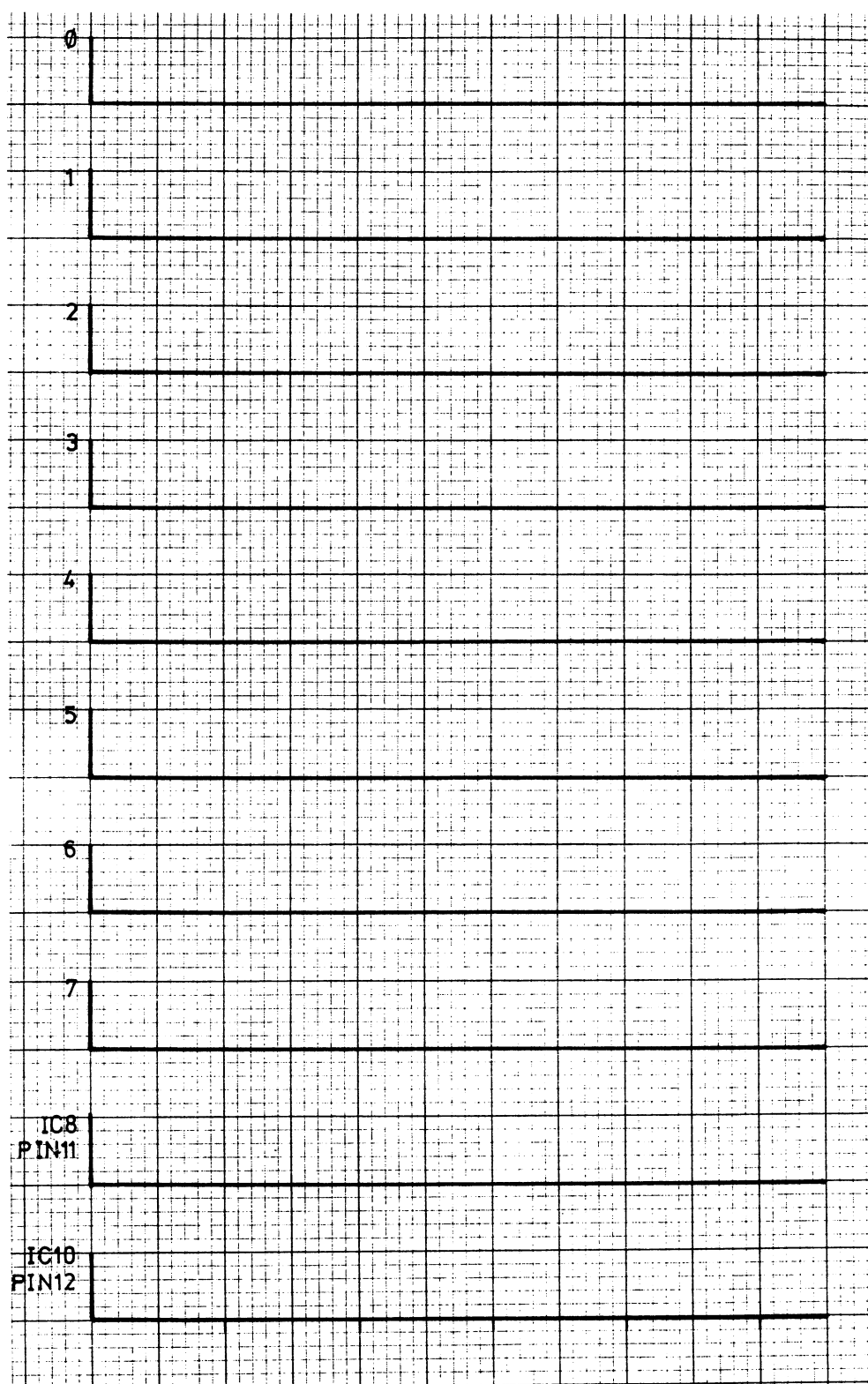
Als Bezugspunkt für alle Messungen können Sie die GND-Anschlüsse der Analog-Ein/Ausgabe benutzen.

Bei richtiger Funktion beider Bausteine müssen sich an allen Anschlüssen (0-7) rechteckförmige Spannungen ergeben, wobei zwei benachbarte Leitungen immer gegenphasige Verläufe aufweisen.

--->

## A6.3

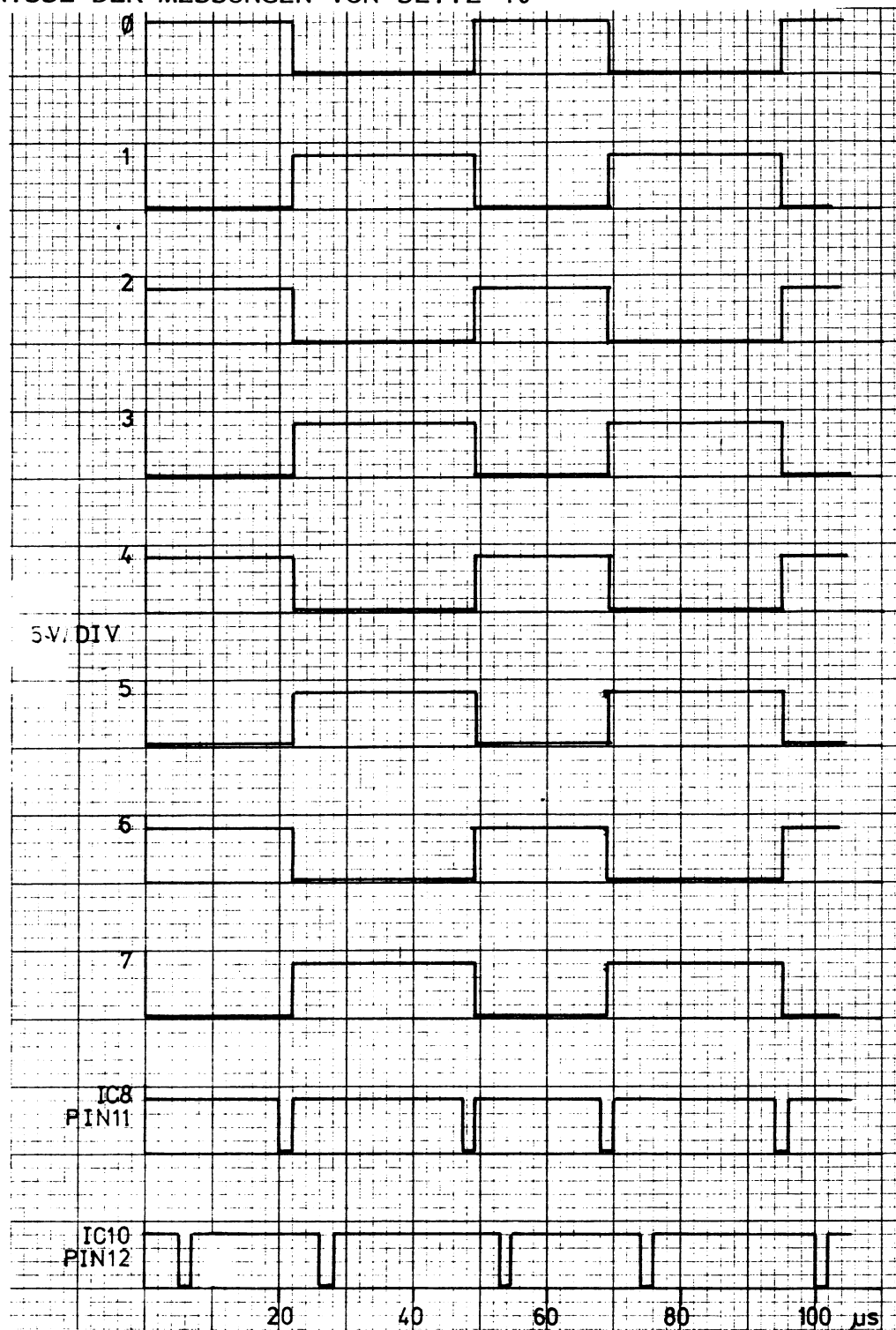
Spannungs-Zeit-Diagramme für die dynamische Funktionsprüfung der TTL-kompatiblen Ein/Ausgabe.



---&gt;

A6.4

## ERGEBNISSE DER MESSUNGEN VON SEITE 40



## HINWEIS ZU DEN MESSUNGEN

Die Triggerung des Oszilloskops erfolgte auf der Ausgabeleitung 0. Die Signale an IC8 PIN11 und IC10 PIN12 entsprechen den Steuersignalen IOW und IOR.

----&gt;

Dynamische Funktionsprüfung der Optokoppler-Ein/Ausgabe

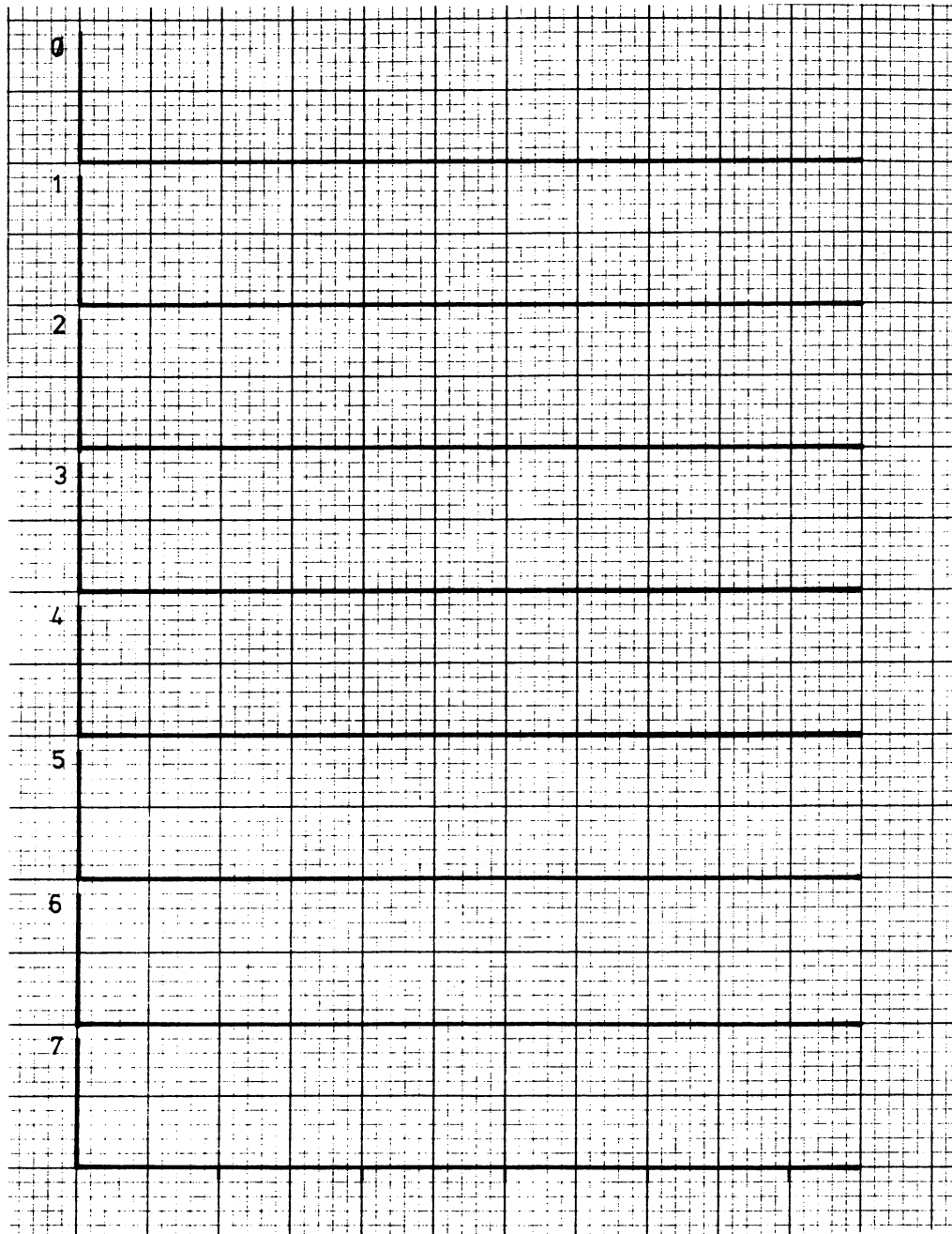
Schalten Sie die Betriebsspannung des Systems ab. Benutzen Sie die acht Leitungen zur Verbindung der Punkte 0-7 der Optokoppler-Ausgabe mit den entsprechenden Punkten der Optokoppler-Eingabe. Danach schließen Sie eine externe Gleichspannungsquelle von 15V an die Anschlüsse "EXT.Vcc" und "EXT.GND1" an. Außerdem müssen Sie die Anschlüsse "EXT.GND1" und "EXT.GND2" miteinander verbinden. Sie müssen jetzt mit Hilfe des Assemblers ein neues Programm eingeben, das zwar im Prinzip die gleiche Wirkung wie das auf Seite 38 hat. Es muß bei der Überprüfung der Optokoppler aber eine Zeitverzögerung zwischen Ausgeben und Lesen des Datenwortes eingefügt werden, da die Optokoppler wesentlich größere Schaltverzögerungszeiten aufweisen als die TTL-Bausteine. Geben Sie, beginnend ab der Adresse E000, folgendes Programm ein:

```
ANF:      MVI  A,55      ;Accu mit 55H laden
          MOV  C,A       ;Wert retten
          OUT  01        ;55H ausgeben
          CALL ZEIT      ;Zeitverzögerung
          IN   01        ;Opto-Eingabe lesen
          CMP  C         ;gelesener Wert=ausgegebener Wert
          JNZ  STOP      ;NEIN!
          MVI  A,0AA     ;Accu mit AAH laden
          MOV  C,A       ;Wert retten
          OUT  01        ;AAH ausgeben
          CALL ZEIT      ;Zeitverzögerung
          IN   01        ;Opto-Eingabe lesen
          CMP  C         ;gelesener Wert=ausgegebener Wert
          JNZ  STOP      ;NEIN!
          JMP  ANF       ;JA, zurück zum Anfang
STOP:     HLT           ;FEHLER!: Prozessor hält an
ZEIT:     MVI  B,8F      ;Zeitverzögerung
Z1:       DCR  B
          JNZ  Z1
          RET
```

Messen Sie jetzt die Signale an den Anschlüssen 0 - 7 der Optokoppler-Ausgabe. Benutzen Sie für die Zeichnungen die Koordinatenkreuze auf der nächsten Seite.

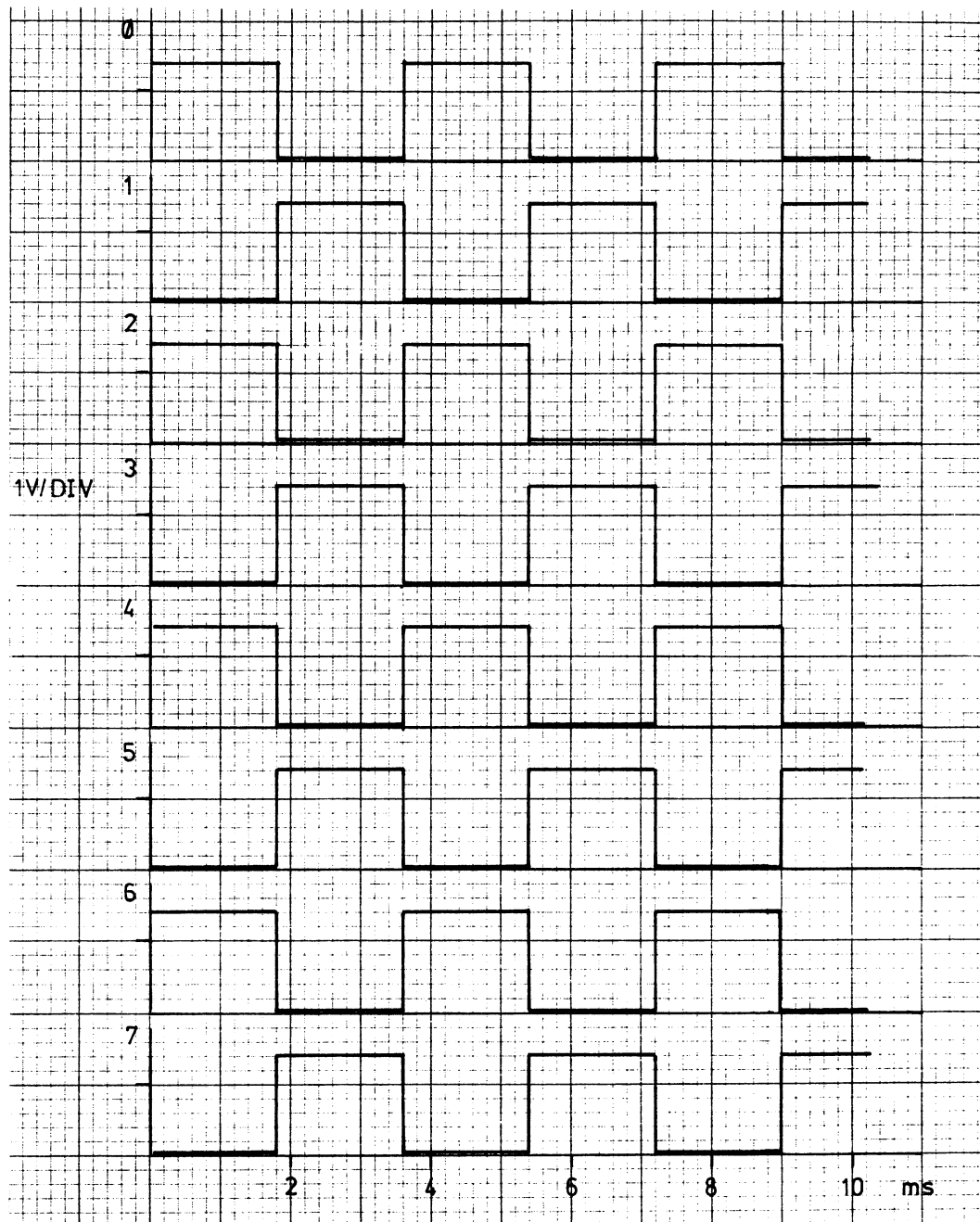
## A6.6

Spannungs-Zeit-Diagramme für die dynamische Funktionsprüfung der Optokoppler-Ein/Ausgabe.



A6.7

## ERGEBNISSE DER MESSUNGEN VON SEITE 43



## HINWEIS ZU DEN MESSUNGEN

Der Konstantstrom von 5,5 mA der Optokoppler-Eingabe stellt für die hochohmigen Kollektorwiderstände der Optokoppler-Ausgabe eine hohe Belastung dar. Bedingt dadurch erreicht die Ausgangsspannung nur einen Wert von ca. 2 Volt.

---&gt;

Funktionsprüfung des D/A-Wandlers

Im ersten Schritt werden der Minimal- und der Maximalwert der Ausgangsspannung des D/A-Wandlers gemessen. Dazu geben Sie folgendes Programm ab der Adresse E000 ein:

```
ANF:      MVI  A,00      ;Accu mit 00 laden
          OUT  02        ;Wert wandeln
ANF1:     CALL 0043      ;Tastatur lesen
          CPI  20        ;Space-Taste betätigt?
          JNZ  ANF1      ;Nein, Tastatur erneut lesen
          MVI  A,0FF     ;Accu mit FFH laden
          OUT  02        ;Wert wandeln
ANF2:     CALL 0043      ;Tastatur abfragen
          CPI  20        ;Space-Taste betätigt?
          JNZ  ANF2      ;Nein, Tastatur erneut lesen
          JMP  ANF       ;Ja, zum Anfang zurück
```

Das Programm wirkt folgendermaßen:

Im ersten Schritt wird das Datenwort 00 umgewandelt. Danach erfolgt eine Abfrage der Tastatur. Solange auf der Tastatur die Space-Taste nicht betätigt wird, kann am Ausgang des D/A-Wandlers der minimale Spannungswert gemessen werden. Wird die Space-Taste betätigt, so wird das Datenwort FF umgewandelt, und am Ausgang des D/A-Wandlers kann der maximale Spannungswert gemessen werden. Wird die Space-Taste erneut betätigt, so beginnt der Vorgang wieder von vorn. Starten Sie das Programm mit dem GO-Kommando und messen Sie mit einem Vielfachinstrument die minimale und die maximale Ausgangsspannung des D/A-Wandlers.

Die gemessenen Spannungen müssen ungefähr folgende Werte haben:

$U_{Amin}$ : 0,3 Volt

$U_{Amax}$ : 2,5 Volt

Jetzt muß der Operationsverstärker IC3 noch abgeglichen werden, damit am Analogausgang der Karte definierte Minimal- und Maximalspannungen auftreten. Für den Operationsverstärker muß ein "OFFSET-ABGLEICH" und ein "VERSTÄRKUNGS-ABGLEICH" durchgeführt werden. Starten Sie das o.g. Programm erneut ab der Adresse E000 und stellen Sie mit R28 am Analogausgang eine Spannung von 0 Volt ein. Nach Betätigung der Space-Taste stellen Sie mit R29 eine Spannung von 3,06 Volt ein. Die beiden Einstellungen müssen mehrmals hintereinander wiederholt werden, da sie sich gegenseitig beeinflussen.

--->



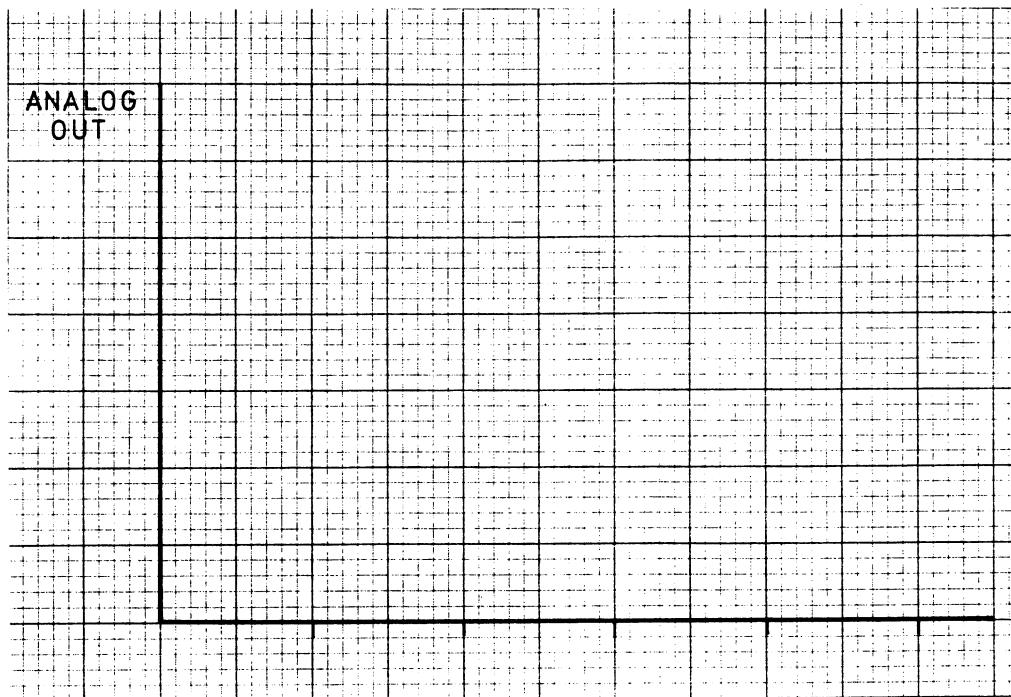
A6.9

Dynamische Funktionsprüfung des D/A-Wandlers

Bei der dynamischen Funktionsprüfung werden durch den D/A-Wandler fortlaufend alle 256 Bit-Kombinationen umgewandelt. Dazu müssen Sie folgendes Programm ab der Adresse E000 eingeben:

```
ANF:      MVI  A,00      ;Accu mit 00H laden
          OUT  02        ;D/A-Wandlung
          INR  A          ;Accu incrementieren
          JMP  ANF        ;nächst höheren Wert wandeln
```

Starten Sie das Programm mit dem GO-Kommando und messen Sie mit dem Oszilloskop den Spannungsverlauf am Analogausgang der Karte. Den Spannungsverlauf zeichnen Sie in das Koordinatenkreuz auf dieser Seite ein.

**HINWEIS ZU DER OBIGEN MESSUNG**

Die Ausgangsspannung muß einen sägezahnförmigen Verlauf haben. Die Minimal- und die Maximalwerte müssen den Spannungswerten des Operationsverstärkerabgleichs entsprechen. Die Zeitablenkung des Oszilloskops muß auf 0,5ms/Div eingestellt sein.

---&gt;

Funktionsprüfung des A/D-Wandlers

Schalten Sie die Betriebsspannung des MFA-Systems aus und fügen Sie die 8Bit-Ausgabekarte mit der Adresse 03 in das MFA-System ein.

An den Analogeingang der Ein-Ausgabekarte schließen Sie die einstellbare Gleichspannungsquelle an. Danach können Sie die Betriebsspannung wieder einschalten und folgendes Programm ab der Adresse E000 eingeben:

```

ANF:      OUT  03          ;Startimpuls A/D-Wandlung
          MVI  B,04        ;Zeitverzögerung
S1:      DCR  B            ;für Umwandlung
          JNZ  S1
          IN   02          ;gewandelten Wert lesen
          JMP  ANF         ;und ausgeben

```

Wirkung des Programms

Durch den ersten OUT-Befehl wird der A/D-Wandler gestartet ( $\overline{IOW}$  nimmt kruzzeitig L-Pegel an). Durch die danach folgenden drei Befehle wird eine Wartezeit für die Umwandlung erzeugt. Mit dem IN-Befehl wird der gewandelte Spannungswert gelesen, und dann beginnt die Umwandlung erneut, wobei der gewandelte Wert auch ausgegeben wird.

Entscheidend für die richtige Funktion des Wandlers ist die Wartezeit, die mindestens neun Taktimpulse betragen muß.

Spannungs-Zeit-Diagramme der A/D-Wandlung

Oszilloskopieren Sie folgende Spannungsverläufe:

$$U_{SC} = f(t) \quad (\text{IC12 PIN4})$$

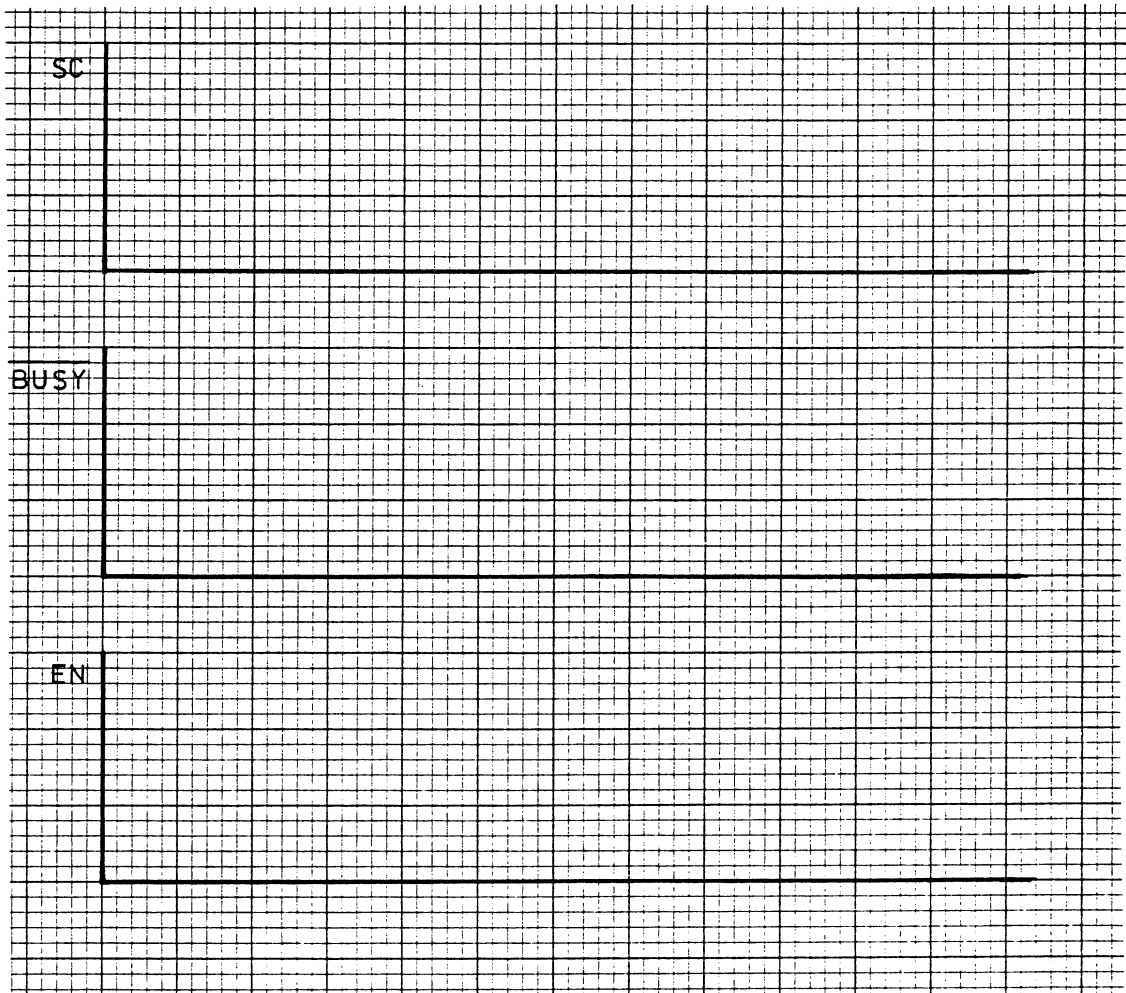
$$U_{\overline{BUSY}} = f(t) \quad (\text{IC12 PIN1})$$

$$U_{EN} = f(t) \quad (\text{IC12 PIN2})$$

Die Oszillogramme tragen Sie in die Koordinatenkreuze auf der nächsten Seite ein!

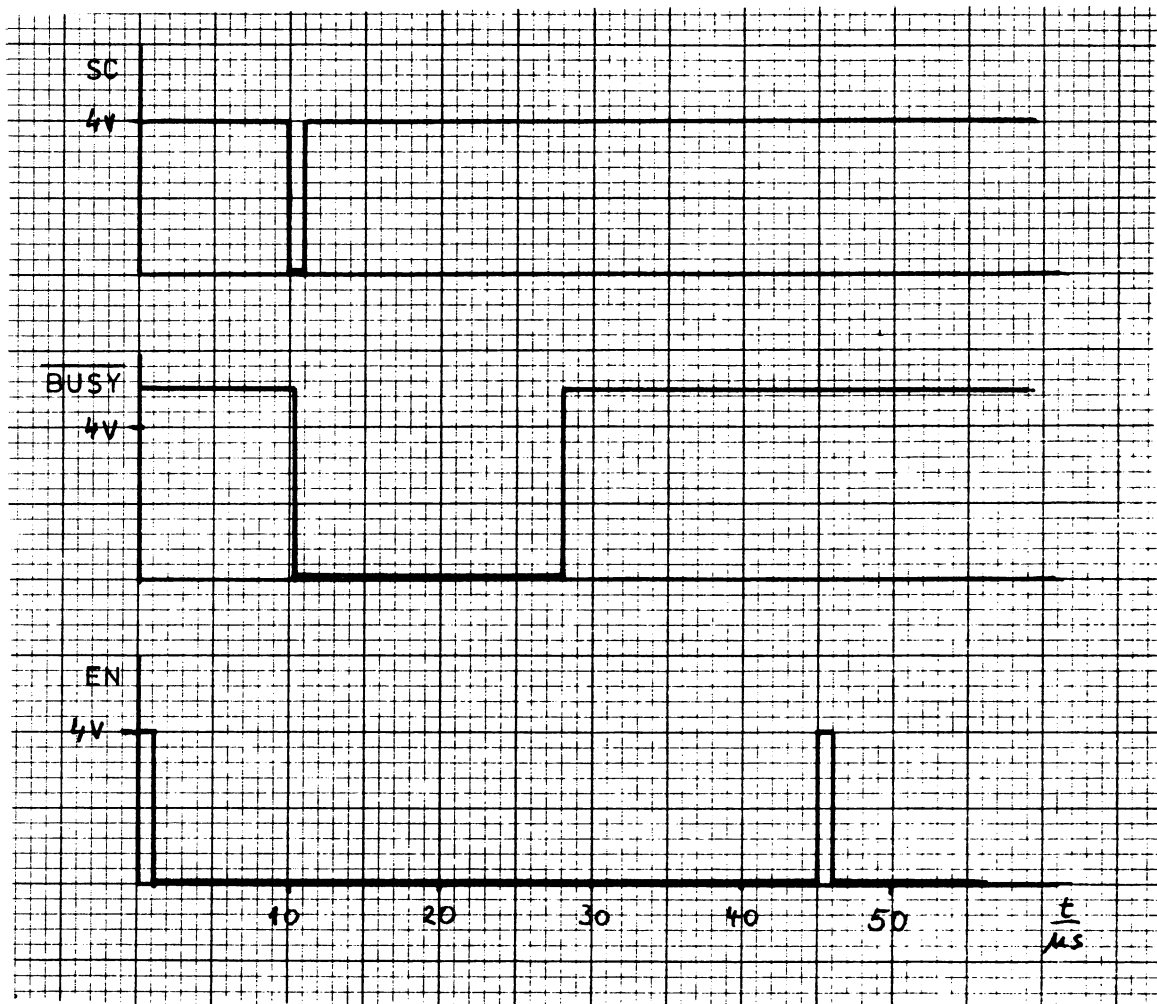
---->

A6.11

Koordinatenkreuze für die Messungen von Seite 47

-----&gt;

## ERGEBNISSE ZU DEN MESSUNGEN VON SEITE 48



## HINWEIS ZU DEN MESSUNGEN

Die Triggerung des Oszilloskops erfolgte auf der positiven Flanke von  $U_{EN}$ . Mit dem ersten Impuls am PIN4 (SC) wird der A/D-Wandler gestartet. Das BUSY-Signal nimmt gleichzeitig L-Pegel an. Nachdem BUSY wieder H-Pegel angenommen hat, ist die Wandlung beendet. Durch den IN-Befehl wird danach der PIN2 (EN) kurzzeitig auf H-Pegel gelegt, der gewandelte Spannungswert gelesen und mit dem OUT-Befehl ausgegeben.

---&gt;

Abgleich des A/D-Wandlers

Schließen Sie die Gleichspannungsquelle an den Analogeingang an und stellen Sie null Volt ein.

Verstellen Sie jetzt das Poti R7 solange, bis an der Ausgabe-Baugruppe die LED B0 gerade aufleuchtet. Danach das Poti R7 in umgekehrter Richtung verstellen, bis die LED B0 gerade erlischt.

Im nächsten Schritt stellen Sie eine Gleichspannung von 10 Volt ein. Jetzt wird das Poti R9 solange verstellt, bis alle LED's der Ausgabe-Baugruppe aufleuchten.

Führen Sie die beiden Abgleicharbeiten mehrmals hintereinander durch, da sie sich gegenseitig beeinflussen.

Damit ist die Übung beendet!

MFA-MEDIENSYSTEM

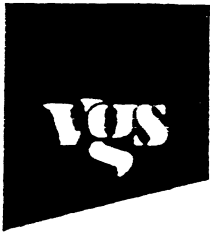
# Mikrocomputer- Technik

Fachpraktische Übungen  
Band 6

6

**MFA**  
MEDIENSYSTEM

**VGS Bfz**



# – Ihr Partner für Ausbildungssysteme

Neue Produkte im Bereich Mediensysteme

Mit Medien, Aus- und Weiterbildung begann sich die vgs bereits vor 20 Jahren mit dem Begleitbuch zur Fernsehreihe „Einführung in die Elektronik“ von Jean Pütz zu beschäftigen. Über die Jahre hinweg entstanden in Zusammenarbeit mit ARD und ZDF weitere Bücher und Bausätze aus den Bereichen Elektronik, Chemie und Biotechnologie. Inzwischen ist die vgs der Spezialist für Medienverbund auf allen Gebieten. Die Zusammenarbeit mit dem Berufsförderungszentrum Essen e.V. begann 1983, als die vgs den Zuschlag für die Produktion und Verbreitung des dort entwickelten Mikrocomputer-Ausbildungssystems MFA erhielt. Inzwischen umfaßt das Programm der vgs für den Bereich der Aus- und Weiterbildung folgende Produkte:

- **MFA** – ein Mikrocomputer-Ausbildungssystem, das in den industriellen Metall- und Elektro-Berufen sowie in den handwerklichen Elektro-Berufen im In- und Ausland eine zentrale Rolle spielt. Das vom BFZ in Essen entwickelte Grundsystem wird von der vgs ständig zukunftsorientiert weiterentwickelt, produziert und vertrieben, so daß inzwischen ca. 90 Baugruppen zur Verfügung stehen.
- **40 900 NORMCOMPUTER** – ein Computer-Lehrsystem für die Schulung im Bereich Digitaltechnik, bei dem Wert auf die Einhaltung der gültigen internationalen Digitalnorm nach DIN 40 900 gelegt wurde (in Zusammenarbeit mit dem BBZ Köln). Mit diesem, aus 8 Lehrplatten bestehendem Lehrsystem, kann die immer noch bestehende Lücke zwischen Digitaltechnik und Mikrocomputertechnik geschlossen werden. Basis des NORMCOMPUTERS ist der in der Aus- und Weiterbildung nach wie vor optimal einzusetzende Mikrocomputer 8085.
- **PC-Modelle** für die Schulung – hier bietet die vgs Anwendungsmodelle aus den Bereichen Logikanalyse, Meßtechnik und Robotik an.
- **PTQ** (steht für Produktionstechnische Qualifikation im Lernverbund). An dieser neuen Entwicklung vom BFZ Essen ist die vgs als Werkvertragsnehmer über die Konstruktion und Dokumentation sowie Fertigungsarbeiten für „Portallader-Komponenten“ beteiligt.

Die vgs liefert in das Inland (inzwischen mit Schwerpunkt neue Bundesländer) und über Vertragspartner in das Ausland an Industrie, Handwerk, Ausbildungszentren und Schulen.

Bitte fordern Sie weitere Informationen an bei:

vgs verlagsgesellschaft mbH & Co. KG  
Postfach 180269  
Breite Straße 118-120, 5000 Köln 1

Telefon 02 21/2 08 11-12  
Telefax 02 21/24 57 99  
Telex 888 2202 vgs d