



## Inhaltsverzeichnis

### Lektion 4

	<u>Seite</u>
1. Der Mikroprozessor in einer anspruchsvolleren Umgebung (Personalcomputer und Entwicklungssystem)	(4)-3
1.1. Beschreibung eines Kleincomputersystems mit Unterscheidungsmerkmalen zu einem Entwicklungssystem	(4)-3
1.2. Standardisierte Schnittstellen	(4)-13
1.3. Integrierte Peripherie-Bausteine	(4)-16
1.3.1. Programmierbarer Peripherie-Schnittstellenbaustein 8255	(4)-17
1.3.2. Programmierbarer Tastatur-/Anzeige-Schnittstellenbaustein 8279	(4)-18
1.3.3. Programmierbarer serieller Schnittstellenbaustein 8251	(4)-20
1.3.4. Prioritäts-Steuerbaustein für Unterbrechungen 8214	(4)-23
2. Erweiterte Anwendungen des "micromaster" über die Anwenderschnittstelle	(4)-25
2.1. Funktions- und Anschlußbeschreibung für den Port-Baustein 8155 mit Anwendungen	(4)-25
2.2. Programmierung der Zählerfunktion im 8155	(4)-32
2.3. Der Umgang mit dem Schnittstellenbaustein 8279	(4)-37
2.4. Möglichkeiten der RAM-Speichererweiterung über die Anwenderschnittstelle des "micromaster" um 1 K	(4)-39
2.5. Anschluß eines EPROM-Programmiergerätes an den "micromaster"	(4)-40
2.6. Unterbrechungs-(Interrupt-)Verarbeitung im "micromaster"	(4)-42
3. Vertiefung der Programmiertechnik an zwei praktischen Beispielen	(4)-49
3.1. Der "micromaster" als Frequenzmeßgerät	(4)-49
3.1.1. Die erforderliche Hardware	(4)-51
3.1.2. Die Software des Frequenzmessers	(4)-52
3.2. Beschreibung für den Anschluß eines Analog/Digital-Wandlers an den 8085	(4)-54
3.2.1. Die Beschreibung des Hardware-Aufbaus	(4)-54
3.2.2. Die Software für das Lesen der AD-Wandler	(4)-56
4. Lösungen zu den Aufgaben in dieser Lektion	(4)-60
	(4)-1



5.	Hausaufgaben	(4)-61
6.	Anhang	(4)-64
6.1.	Bild 27 - Erweiterung mit Port-Baustein 8155	(4)-65
6.2.	Bild 28 - Erweiterung mit 1K-RAM	(4)-66
6.3.	Bild 29 - Erweiterung mit 2K-EPROM und EPROM- Programmierer	(4)-67
6.4.	Bild 30/31 - Analog/Digitalwandlung mit Baustein ZN 427 von Ferranti	(4)-68
6.5.	Belegung der Anwenderschnittstelle	(4)-70
6.6.	Belegung des Speicherraumes mit den zugehörigen $\overline{\text{CS}}$ -Signalen	(4)-71



# 1. Der Mikroprozessor in einer anspruchsvolleren Umgebung (Personalcomputer und Entwicklungssystem)

Wenn man von einer anspruchsvolleren Umgebung für den Mikroprozessor spricht, ist damit gemeint, daß die Peripherie um den Prozessor herum auf größt möglichen Komfort für den Anwender ausgebaut ist. Das bezieht sich sowohl auf die Hard- wie auch Software. Prinzipiell lassen sich mit Mikroprozessoren ausgerüstete Mikro- und Minicomputer in zwei Gruppen aufteilen, einmal in die Kleincomputersysteme, die unter den Bezeichnungen Home-Computer oder Personal-Computer einzuordnen sind, und zum anderen die Entwicklungssysteme, mit denen neue, mit Mikroprozessor gesteuerte Geräte (z.B. Meßgeräte) entwickelt werden können.

## 1.1 Beschreibung eines Kleincomputersystems mit Unterscheidungsmerkmalen zu einem Entwicklungssystem

Der Anwender eines Kleincomputersystems erwartet heute einen hohen Bedienungskomfort, dem von Seiten der Hersteller Rechnung getragen wird. Das sieht man an der schnellen Folge mit der neue Geräte auf dem Markt angeboten werden. Zu einer Mindestausstattung eines solchen Computers, der beispielsweise in einem Kleinbetrieb eingesetzt werden soll, gehören

**M**

- eine Zentraleinheit (CPU)
- ein RAM-Speicherausbau mit 64K oder größer,
- ein Datenmonitor (Bildschirm)
- eine Tastatur
- ein Massenspeicher-Medium (z.B. Floppy-Disk) und
- ein Drucker

Je nach Anforderung und Aufgabe gibt es weitere Peripheriegeräte, die ein solches System ergänzen, wie zum Beispiel

Festplatten-Laufwerke für große Datenmengen  
Grafikdrucker  
Grafikbildschirme  
Typenradschreibmaschinen als Schönschreibdrucker  
Spooler (Druckpuffer)

In dem Bild 1 ist in einem Blockschaltbild ein solcher Ausbau dargestellt. Das Bild 2 zeigt ein entsprechendes Foto eines Computersystems.

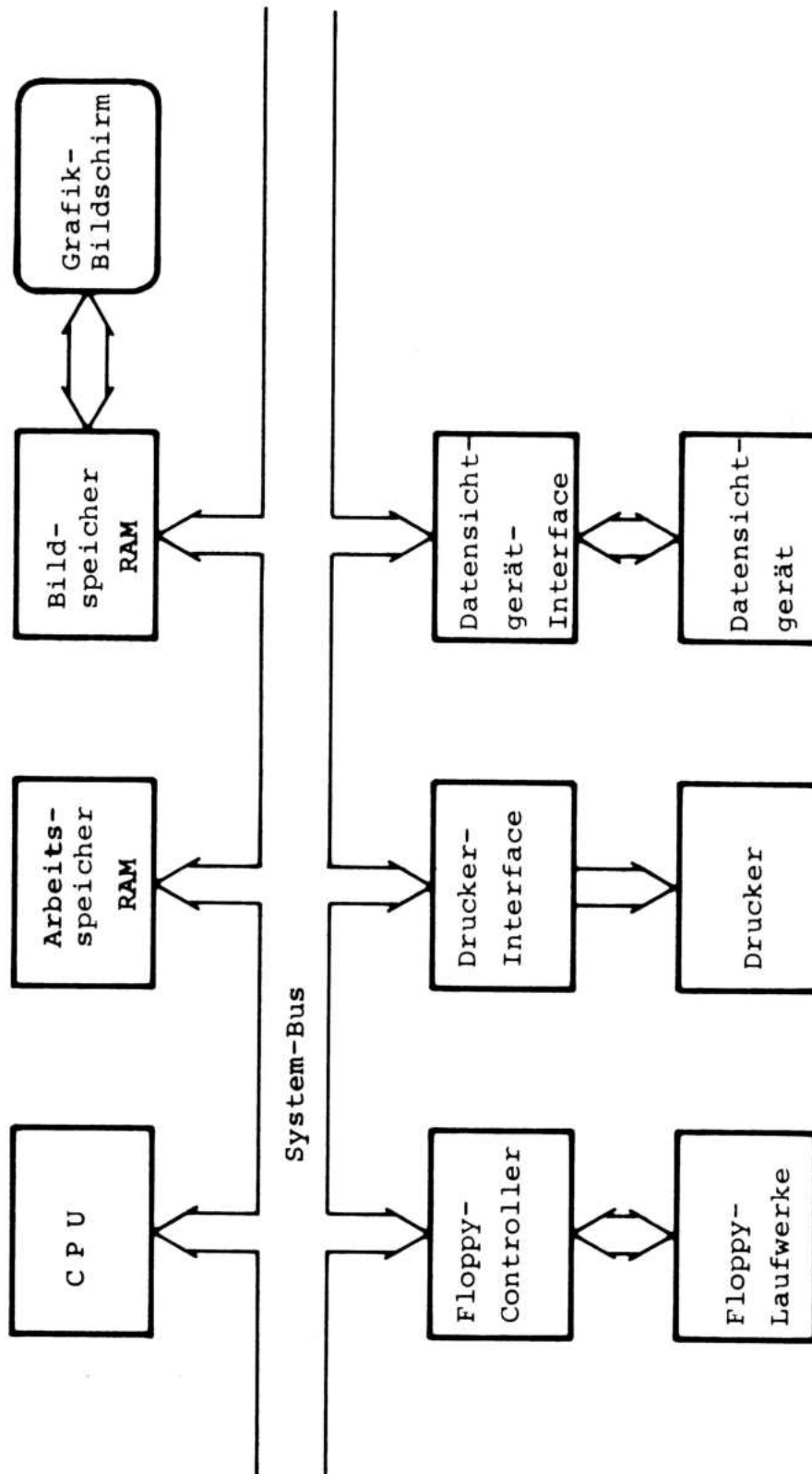


Bild 1

Blockschaltbild eines Computersystems





Bild 2



Sie haben inzwischen gelernt, daß die Hardware eines Computersystems ohne Software nicht arbeitet.

Wir brauchen dazu

- ein Betriebssystem
- einen Compiler oder Interpreter
- die Anwendersoftware

Die eben genannten Hardware-Bausteine sind Ihnen im Prinzip schon bekannt. Sie sind eben nur in einem solchen größeren System für den Anwender mit komplexeren Aufgaben einfacher in der Bedienung. So werden auf einem Bildschirm die Buchstaben und Ziffern nicht mehr im 7-Segment-Code dargestellt, sondern in einer Matrix (z.B. 7x9) aus einzelnen Punkten zusammengesetzt (Bild 3).

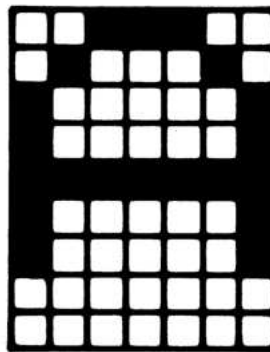


Bild 3: Darstellung des Buchstaben A in einem Punktraster

Alle Matrixdrucker arbeiten nach dem gleichen Prinzip. Aus einem Nadelkopf werden die entsprechenden Nadeln einzeln angestoßen, die zur Darstellung eines Buchstabens oder eines anderen Zeichens erforderlich sind. Nahezu alle Drucker bieten heute die Möglichkeit, zwischen mehreren Zeichensätzen zu wählen. Beispielsweise erfordert der deutsche Zeichensatz die Umlaute, die im amerikanischen nicht enthalten sind.

Eine Tastatur, die nur die Möglichkeit zur Eingabe von Hex-Zeichen bietet, kann natürlich an einem ausgebauten Computersystem nicht mehr verwendet werden. Es wird heute eine vollständig mit allen Ziffern und Buchstaben, sowie Sonderzeichen belegte Tastatur gefordert. Groß- und Kleinschreibung muß möglich sein. Der verwendete Standard-Code zur Übertragung zwischen dem Computer und der Tastatur ist der ASCII-Code, der in der ersten Lektion im Abschnitt über Codes abgedruckt ist.



Als externer Massenspeicher wird bei einigen Geräten die Kassette verwendet. Diese hat den Nachteil, daß die gesamte Organisation der Abspeicherung vom Anwender selbst vorgenommen werden muß. Für Kassetten werden keine Inhaltsverzeichnisse angelegt, d.h. jeder muß seine eigene Liste mit abgespeicherten Programmen und zugehörigem Bandzählerstand führen. Ausnahme davon sind die speziellen Datenkassettenlaufwerke, die ähnlich den Floppydisk-Laufwerken arbeiten. In allen Fällen handelt es sich um ein magnetisches Aufzeichnungsverfahren.

Die Floppydisk oder Diskette, meist kurz Floppy genannt, ist eine flexible Scheibe, auf der einseitig oder beidseitig eine dünne Magnetschicht aufgebracht ist. Diese Schicht ist sehr empfindlich. Deshalb wird sie in eine feste Umhüllung eingeschweißt, die für den Zugriff und Antrieb entsprechende Löcher aufweist. Das Bild 4 zeigt die Schutzhülle der Floppy und die Lage der Floppy in der Hülle.

Die auf einer Floppy zu speichernden Daten werden als elektrische Signale von einem sogenannten Floppy-Controller an das Laufwerk mit dem Schreib-/Lesekopf gegeben. Beim Schreiben wird die Magnetschicht entsprechend der gesendeten Daten beeinflusst. Das erfolgt in Spuren, die wie Ringe um das Antriebsloch angelegt werden. Diese Spuren werden wiederum in Sektoren unterteilt. Der erste Sektor wird eindeutig durch ein Indexloch bestimmt, gleich in welcher Spur sich der Kopf befindet.

Dieses Einteilen der Floppy nennt man formatieren. D.h. jede neue, unbenutzte Floppy muß erst für den Gebrauch vorbereitet, sprich formatiert werden. Das geschieht durch den Start einer speziellen Software, die meist Teil des Betriebssystems ist. Eingesetzt werden heute drei verschiedene Größen, die 8"-Standardfloppy, die 5 1/4"-Mini-Floppy und die 3 1/2"-Micro-Floppy. Diese gibt es sowohl für einfache oder doppelte Schreibdichte (single/double density) und ein- oder zweiseitiger Beschichtung (single/double sided). Beispielsweise lassen sich auf einer 5 1/4"-Floppy mit doppelter Dichte und beiden Seiten unformatiert ca. 2MByte abspeichern. Davon geht jedoch ein Teil für die Formatierung verloren.

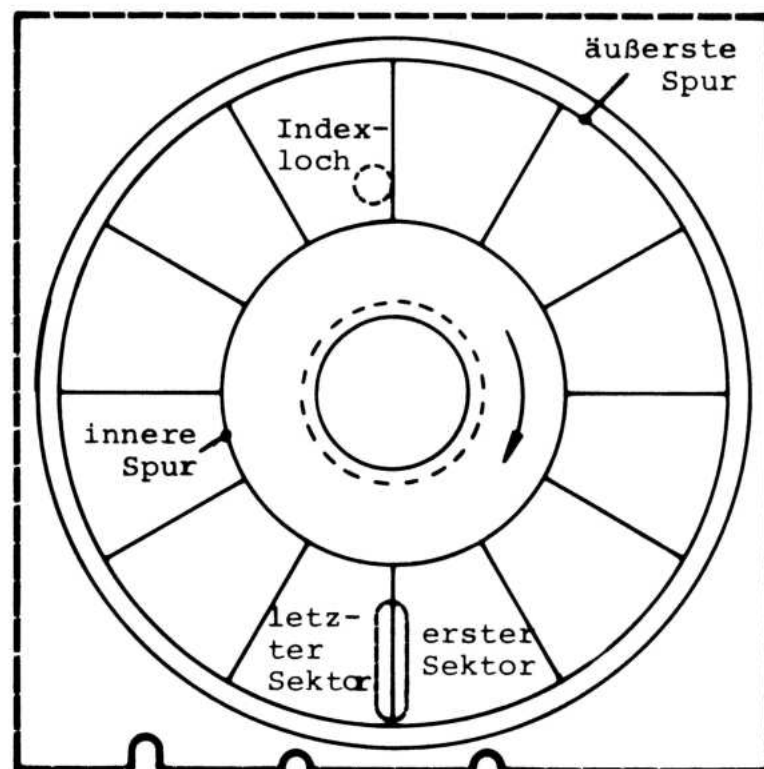
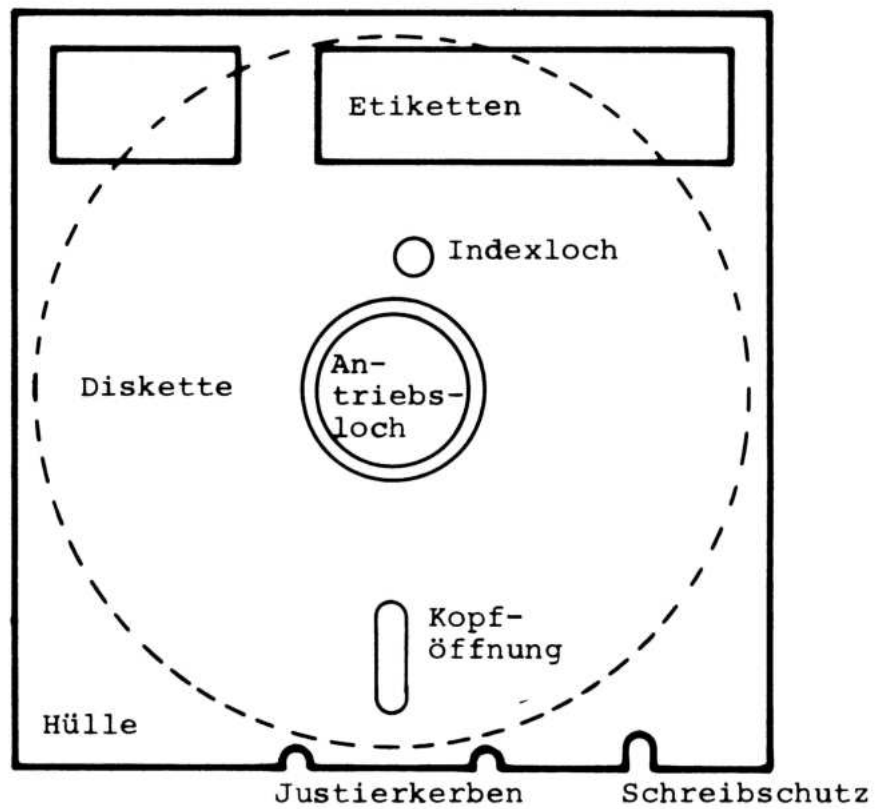


Bild 4



Wesentlich für den Anwender ist, daß er selbst bei der Floppy keinerlei Aufwand für die Speicherorganisation hat. Das System legt auf der Floppy ein Inhaltsverzeichnis an, das ständig abgefragt werden kann. Will man ein Anwenderprogramm von der Floppy in den Speicher holen, geschieht das nur unter Angabe des Namens mit dem das Programm abgelegt wurde. Die auf einer Floppy unter einem Namen abgelegten Daten oder Programme werden mit dem engl. Wort file oder im Deutschen als Datei bezeichnet.

Festplattenlaufwerke arbeiten ähnlich wie die Floppylaufwerke. Sie sind jedoch kein Wechselmedium, d.h. hier ist das Trägermedium eine feste Scheibe, die im Laufwerk montiert ist und im Gegensatz zur Floppy nicht entnommen werden kann. Durch die mechanische Stabilität ist eine wesentlich genauere Kopfpositionierung und somit um ein Vielfaches höheres Speichervolumen (heute bis zu über 100MByte je nach Laufwerk) möglich.

Das Betriebssystem eines Computersystems ermöglicht Ihnen den Datentransport zwischen allen Baustufen und angeschlossenen Geräten. Es ermöglicht die Bedienung des Computers (siehe Lektion 1, Abschnitt 3.3). Weit verbreitete Standard-Betriebssysteme sind z.B. CP/M, MSDOS, UNIX usw. Es gibt eine ganze Reihe weiterer Systeme, die wir hier nicht alle aufzählen können. (Wer sich für CP/M interessiert, mag das Buch "Vom Umgang mit CP/M" von Bernd Pol, erschienen im IWT-Verlag lesen.)

**K** Aufgabe 1: Fassen Sie einige Merkmale zusammen, die den Komfort eines Kleincomputer-Systems ausmachen.

Anhand der Beschreibung einer Programmerstellung wollen wir den Umgang mit einem Kleincomputersystem erläutern. Wir gehen davon aus, daß wir uns, bevor wir uns an den Computer setzen, mit der Aufgabenstellung vertraut gemacht und einen Ablaufplan erstellt haben. Dann ist die erste Frage, wie wir unsere Befehlsfolge (Programm) erstellen und im Computer, bzw. auf der Floppy ablegen können.

Für diesen ersten Schritt ist ein Software-Hilfsmittel erforderlich, das als Editor bezeichnet wird. Der Editor ermöglicht es, von dem Datensichtgerät (Tastatur) Zeichen in den RAM-Speicher einzulesen und auch auf Floppy abzuspeichern. Beim Programmieren werden diese Zeichen sinnvolle Worte aus dem Befehlsvorrat der zu verwendenden Programmiersprache sein.



## M

Der Editor ist ein Hilfsmittel zum Erstellen neuer Text-(Programm-)Dateien und zum Verändern vorhandener Dateien.

Nachdem also das Programm eingetippt wurde, wird das Kommando zum Abspeichern auf die Floppy gegeben. Wir wollen noch erwähnen, daß es zwei verschiedene Typen von Editoren gibt, nämlich den kommandogesteuerten Editor sowie den bildschirmorientierten Editor. Der weniger komfortable ist der kommandogesteuerte Editor. Hier kann eine unsichtbare Schreibmarke durch den geschriebenen Text nur über Kommandos gesteuert werden.

Zum Beispiel

- \*Schreibmarke an Text-Anfang (-Ende) stellen
- \*Schreibmarke vor(zurück)schieben um n Zeilen
- \*Schreibmarke vor(zurück)schieben um n Zeichen
- \*Zeichen bei Schreibmarke löschen
- \*Zeile ab Schreibmarke bis Zeilenende löschen
- \*Datei normal schließen und auf Floppy sichern
- usw.

Jedes dieser Kommandos wird mit einigen Zeichen abgekürzt, die über die Tastatur einzutippen sind. Wesentlich im Gegensatz zum bildschirmorientierten Editor ist, daß hier die Schreibmarke nicht mit Cursor-Tasten beliebig auf dem Bildschirm positioniert werden kann. Beim bildschirmorientierten Editor kann die Schreibmarke sichtbar per Tastendruck an jede Stelle des Textes gebracht werden. An dieser Position kann dann wieder per Tastendruck ein gewünschtes Kommando ausgeführt werden.

Wir gehen jetzt davon aus, daß wir das Programm mit Hilfe eines Editors eingegeben und auf der Floppy gesichert haben. Da es in dieser Form im Computer noch nicht ablaufen kann, muß es erst in die Maschinensprache übersetzt werden. Das geschieht mit Hilfe eines Assemblers, wenn wir beim Programmieren die Mnemonics hingeschrieben haben, oder mit einem Compiler. D.h. dieser Übersetzer wird in den RAM-Speicher geladen und erzeugt von dem ihm mitgeteilten Anwenderprogramm, das auch in den Speicher geholt wird, den Maschinen-Code. Neben dem Maschinen-Code wird in der Regel auch ein sogenanntes Listing erstellt, welches den Quellcode (unser mit dem Editor erstelltes Programm), den Maschinen-Code und die zugehörigen Adressen zusammen wiedergibt. Viele Assembler und Compiler fügen auch eine Zeilennummerierung ein.





Prinzipiell können wir jetzt die Datei, die den Maschinen-Code enthält, aufrufen und somit unser Programm starten. In den meisten Fällen wird das Programm, wenn es etwas umfangreicher ist, aber nicht fehlerlos arbeiten. Diese Fehler werden beim Übersetzungslauf gemeldet und sind mit Hilfe des Editors zu beseitigen. Es handelt sich hier um Syntax-Fehler, d.h. um formale Fehler, die z.B. durch eine falsche Schreibweise eines Befehls verursacht werden.

Sind diese Fehler beseitigt, muß das Programm trotzdem nicht zwingender Weise funktionieren. Jetzt kann es noch logische Fehler enthalten. Um diese herauszufinden, gibt es wieder ein Software-Hilfsmittel, den sogenannten Debugger. Ein solcher Debugger ermöglicht es, ein Programm in den Arbeitsspeicher zu holen und es dann auszutesten. Das Verhalten des Programms kann ständig unter Kontrolle des Debuggers beobachtet werden, indem man es im Einzelschrittverfahren oder mit Haltemarken abarbeitet und dabei Register- und Speicherinhalte anschauen und auch verändern kann.

Nachdem alle Fehler beseitigt sind, werden diese auch im Quellcode korrigiert und das Programm erneut übersetzt. Danach steht ein fehlerfreier Maschinencode zur Verfügung, der beliebig oft gestartet werden kann. Ergänzend soll noch darauf hingewiesen werden, daß es bei den Interpretern, z.B. in der Programmiersprache BASIC, Editormöglichkeiten gibt, die fest in einen solchen Interpreter eingebunden sind. Das hängt mit der grundsätzlich anderen Arbeitsweise eines Interpreters gegenüber einem Compiler zusammen (siehe Lektion 2, Seiten 4/5).

**K** Aufgabe 2:       Geben Sie stichwortartig den Verlauf einer Softwareentwicklung von der Aufgabenstellung bis zum ersten ordnungsgemäßen Programmlauf an.

Die Programmentwicklung an einem Entwicklungssystem unterscheidet sich etwas von dem eben beschriebenen, da die Zielrichtung eine andere ist. Mit Entwicklungssystemen wird eine Software erstellt, die auf eine andere Mikroprozessor-Hardware übertragen wird, z.B. in einem EPROM gespeichert, und die dann dieses neue Gerät steuert. Ein solches Gerät kann z.B. ein Haushaltsgerät (Waschmaschine o.ä.), ein Meßgerät oder eine Maschine in einem Produktionsprozeß sein.

Bevor jedoch die Übertragung der Software erfolgen kann, muß diese zusammen mit der neu entwickelten Hardware ausgetestet werden. Es muß eine Koppplung zwischen dem Entwicklungssystem, das die Software enthält, und der Hardware hergestellt werden.



## M

Das Entwicklungssystem ersetzt den Mikroprozessor und die Software des neu entwickelten Gerätes über einen Testadapter (Bild 5).

Für das Austesten ist in dem Entwicklungssystem eine unterstützende Software erforderlich, die ähnliche Funktionen erfüllt, wie der vorher beschriebene Debugger. Der für den Anwender zugängliche Bus des Systems ist ein weiteres Merkmal eines Entwicklungssystems, obwohl heute auch die Personalcomputer häufig diese Schnittstelle für Erweiterungen zur Verfügung stellen.

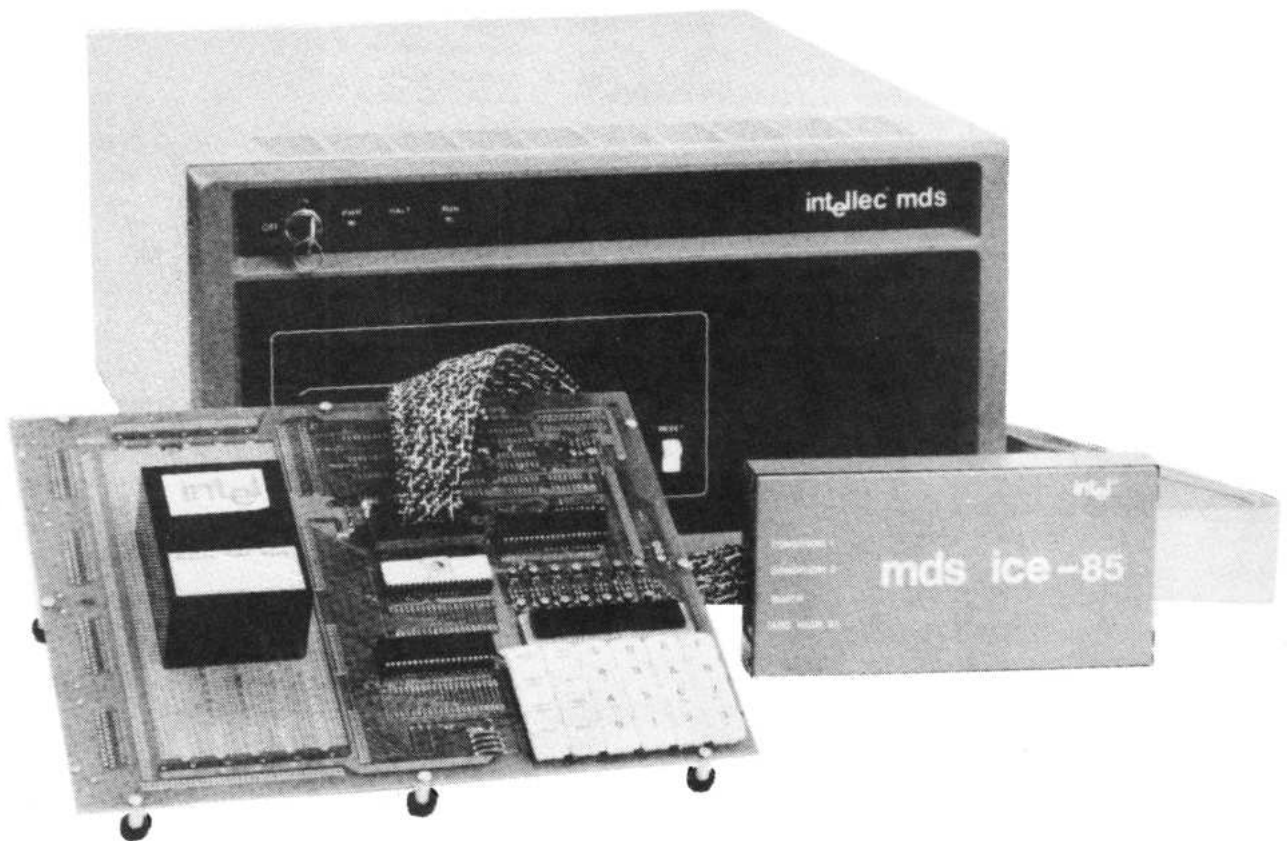


Bild 5

INTEL-Entwicklungssystem mit ICE-85 Testadapter, der auf einem Prototypenboard steckt.





## 1.2 Standardisierte Schnittstellen

Wenn heute Peripheriegeräte an einen Computer angeschlossen werden, verwendet man international übliche Standard-Schnittstellen. Das sind

- die 20 mA-Stromschleife
- die serielle RS232C (V24) Schnittstelle
- die parallele Centronics-Schnittstelle

Die 20 mA-Stromschleife wird überwiegend beim Betrieb von Fernschreibern (TTY, teletypewriter) eingesetzt. Typische Merkmale sind die

- \*Übertragungsgeschwindigkeit 110 Baud  
(1Baud  $\approx$  1 Bit/s)
- \*1 Zeichen wird mit 11 Bits übertragen,  
1 Start-Bit, 7 Daten-Bits, 1 Parity-Bit  
und 2 Stop-Bits.

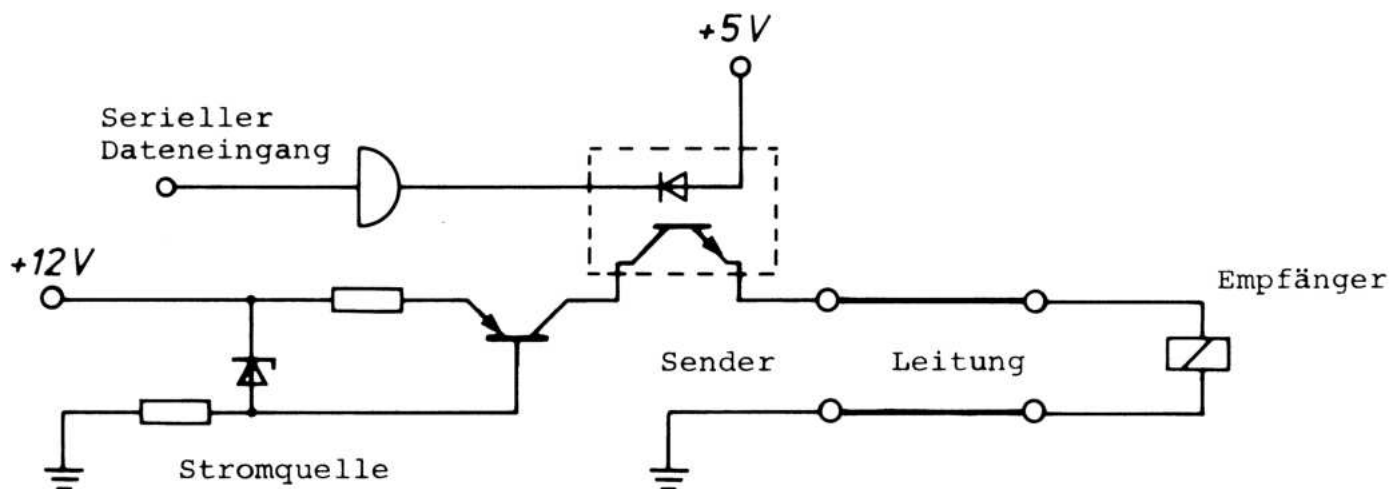


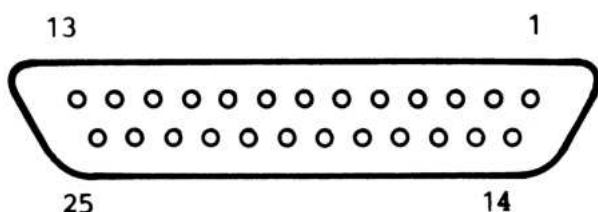
Bild 6: Serielle 20 mA-Stromschleife

Viel häufiger wird die serielle RS 232 C-Schnittstelle eingesetzt. Sie verbindet Drucker, Datensichtgeräte, Plotter, usw. mit dem Computer. Neben den Daten werden auch Steuersignale mit übertragen. Bei einer abgemagerten Belegung des Steckers (25polig, Serie D) spricht man von der V24-Schnittstelle. Die Übertragungsgeschwindigkeit muß bei sendenden und empfangenden Gerät gleich eingestellt werden. Typische Geschwindigkeiten sind



110 Baud	1200 Baud
150 Baud	2400 Baud
300 Baud	4800 Baud
600 Baud	9600 Baud
	19200 Baud

In der Regel wird der ASCII-Code mit Start- und Stop-Bits übertragen. Die Belegung des Steckers ist in Bild 7 dargestellt.



PIN	SIGNAL NAME	DESCRIPTION	Beschreibung
2	T X D	Transmitted Data	gesendete Daten
3	R X D	Received Data	empfangene Daten
4	R T S	Request to Send	Bereitschaft zum Daten senden
5	C T S	Clear to Send	Sendebereitschaft
6	D S R	Data Set Ready	Empfangsbereitschaft
7	G N D	Ground	Masse (Bezugspotential)
15	T X C	Transmitter Clock	Sendetakt
17	R X C	Receiver Clock	Empfangstakt
20	D T R	Data Terminal Ready	Datenterminal bereit (online)

Bild 7

Lötseite des Steckers RS 232 C

Leider sind diese Schnittstellen häufig ein Problem. Beim Eigenbau von Verbindungsleitungen sind die Erfordernisse in den Gebrauchsanleitungen der zu verbindenden Geräte genau nachzulesen. Die Hersteller verwenden zwar die entsprechenden Signale auf den vorgegebenen Anschlußpins der Verbinder, aber der Belegungsgrad ist unterschiedlich. Die Minimalbelegung auf dem 25poligen Stecker



kann folgendermaßen sein

- 2 Datenausgang (gesendete Daten)
- 3 Dateneingang (empfangene Daten)
- 7 Masse

Will man beispielsweise die Funktionsfähigkeit eines Datensichtgerätes prüfen, das sich nicht in den Local-Modus (off-line-Betrieb, d.h. Betrieb ohne am Rechner angeschlossen zu sein) schalten läßt, so muß man am Datenleitungsstecker die Pins 2 und 3 verbinden. Dann sendet das Gerät die Daten an sich selbst. Was auf der Tastatur eingegeben wird, zeigt sich sofort auf dem Bildschirm, wenn das Gerät in Ordnung ist.

Mit einer parallelen Schnittstelle sind hohe Datenübertragungsraten erreichbar, da alle 8 Bits eines Datenbytes gleichzeitig übertragen werden. Außerdem sind Steuersignale erforderlich, die eine einwandfreie Übergabe ermöglichen. Entsprechend viele Leitungen müssen zwischen sendendem und empfangenden Gerät geschaltet werden.

Neben dem Datenbyte wird ein Übergabeimpuls (Strobesignal) übertragen. Wurde das Byte übernommen, sendet das empfangende Gerät einen Impuls als Empfangsbestätigung (Acknowledge, ACK) zurück. Bleibt das ACK-Signal nach einem Strobe aus, ist das gesendete Byte nicht angekommen. Diese Betriebsart zwischen Sender und Empfänger bezeichnet man als handshake-Verfahren.

Die wichtigsten Belegungen der 36-poligen Steckverbindung Centronics 702 sind in der folgenden Tabelle enthalten. Beim Zusammenschalten ist auch die Beachtung der Gerätebeschreibungen wichtig.



Centronics 702 Pin	Funktion
1	Data Strobe
2	Data Bit 1
3	Data Bit 2
4	Data Bit 3
5	Data Bit 4
6	Data Bit 5
7	Data Bit 6
8	Data Bit 7
9	Data Bit 8
10	Acknowledge ACK
11	Busy

Bei einer Parallelschnittstelle sind alle signalführenden Leitungen als Schutz gegen Störungen mit einer Masseleitung verdreht auszuführen. Es kann z.B. ein paarverseiltes Kabel mit Schirmung verwendet werden. Die Masseanschlüsse liegen meist ab Pin 20.

### 1.3 Integrierte Peripherie-Bausteine

Bei den heutigen Mikroprozessor-Konzepten kann man von Mikroprozessor-Familien sprechen. Die Halbleiter-Hersteller haben eine Vielzahl von Funktionsbausteinen entwickelt, die bezogen auf die zu verbindenden Signale zu dem jeweiligen Mikroprozessor "passen". Es handelt sich dabei um Bausteine, die meistens als Interface eingesetzt werden, teilweise wieder über Kommandos programmierbar (umfunktionierbar) sind und den Prozessor von Peripherie-Steuerfunktionen entlasten. Vergleichen Sie die Anschlußbelegung des Mikroprozessors mit der der Peripheriebausteine, so finden Sie heraus, daß alle gleiche Anschlüsse, auch bei den Steuerleitungen verwenden, so daß eine Zusammenschaltung einfach möglich ist. (Siehe Beschreibung des Bus-Konzeptes in Lektion 1.)

Wie Sie sehen werden, ist dem Mikroprozessor-Anwender mit diesen hochintegrierten Funktionsbausteinen sehr viel Hardware-Entwicklung abgenommen, zwar nicht alles, aber eben viel.



### 1.3.1 Programmierbarer Peripherie-Schnittstellenbaustein 8255 (8155)

(Kurzbeschreibung, Anschlußbelegung, Anschlußbeschreibung, Blockschaltbild)

SAB 8255A-5kompatibel mit dem System SAB 8085

24 programmierbare E/A-Anschlüsse

Vollständig TTL-kompatibel

Völlig kompatibel mit allen Siemens Mikroprozessoren

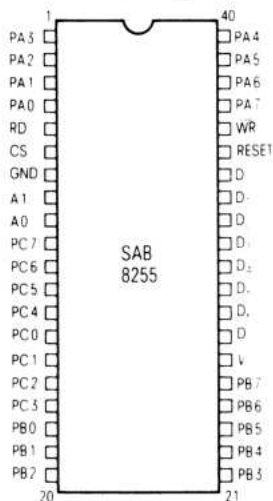
Direkte Bit-Setz- und Rücksetz-Möglichkeiten zur Vereinfachung der Schnittstellen bei Steuerungsanwendungen

Verbesserte Ausgangstreiberleistung

Verbessertes Zeitverhalten

Nur eine Versorgungsspannung (5 V)

#### Anschlußbelegung



#### Anschlußbezeichnungen

D <sub>0</sub> -D <sub>7</sub>	Daten-BUS (Zweiweg)
RESET	Rücksetz-Eingang
CS	Baustein-Auswahl
RD	Lese-Eingang
WR	Schreib-Eingang
A <sub>0</sub> , A <sub>1</sub>	Kanal-Adresse
PA <sub>0</sub> -PA <sub>7</sub>	Kanal A (Bit 0 bis 7)
PB <sub>0</sub> -PB <sub>7</sub>	Kanal B (Bit 0 bis 7)
PC <sub>0</sub> -PC <sub>7</sub>	Kanal C (Bi 0 bis 7)
V <sub>cc</sub>	Versorgungsspannung (+5V)
GND	Masse (0 V)

#### Blockschaltbild 8255

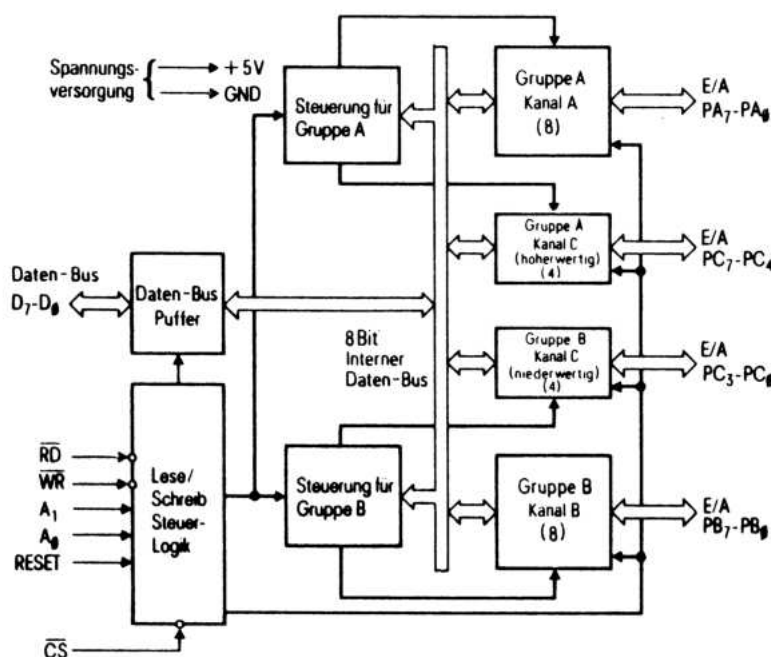
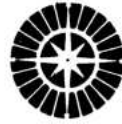


Bild 8



### 1.3.2 Programmierbarer Tastatur-/Anzeige-Schnittstellenbaustein 8279

(Kurzbeschreibung, Anschlußbelegung, Anschlußbezeichnung, Blockschaltbild)

SAB 8279-5 kompatibel mit SAB 8085

Gleichzeitige Tastatur und Anzeige-Operationen

Tastatur-Abtastbetrieb

Sensor-Abtastbetrieb

Getaktete Eingabe

Tastatur-FIFO für 8 Zeichen

2-Key Lockout oder N-Key Rollover mit Kontakt-Entprellung

Zweifache 8- oder 16-stellige numerische Anzeige

Einfache Anzeige mit 16 Zeichen

16 Byte-Anzeige für Eingabe von links oder von rechts

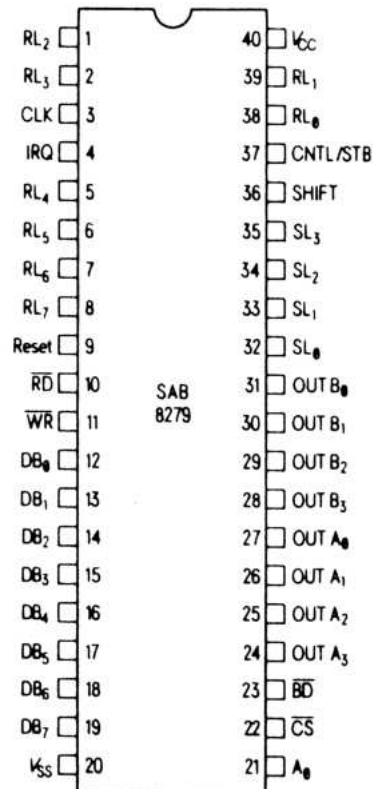
Betriebsart vom Prozessor programmierbar

Programmierbare Abtastzeiten

Unterbrechungs-Ausgang bei Tasten-Eingabe

Nur eine Versorgungsspannung (5 V)

**Anschlußbelegung**



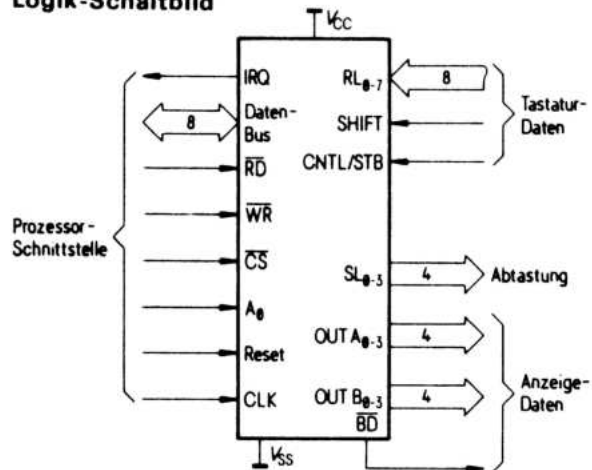
**Anschlußbezeichnungen**

DB <sub>0-7</sub>	Datenbus (bidirektional)
CLK	Takt-Eingang
RESET	Rücksetz-Eingang
$\overline{CS}$	Baustein-Auswahl
$\overline{RD}$	Lese-Eingang
$\overline{WR}$	Schreib-Eingang
A <sub>0</sub>	Puffer-Adresse
IRQ	Unterbrechungs-Anforderungs-Ausgang
SL <sub>0-3</sub>	Abtast-Anschlüsse
RL <sub>0-7</sub>	Rück-Anschlüsse
SHIFT	SHIFT-Eingang
CNTL/STB	CONTROL-/Übernahme-Eingang
OUT A <sub>0-3</sub>	Anzeige-(A)-Ausgänge
OUT B <sub>0-3</sub>	Anzeige-(B)-Ausgänge
$\overline{BD}$	Ausgang für Dunkelastung der Anzeige
V <sub>CC</sub>	Versorgungsspannung (+ 5 V)
V <sub>SS</sub>	Masse (0 V)

Bild 10



### Logik-Schaltbild



### Blockschaltbild

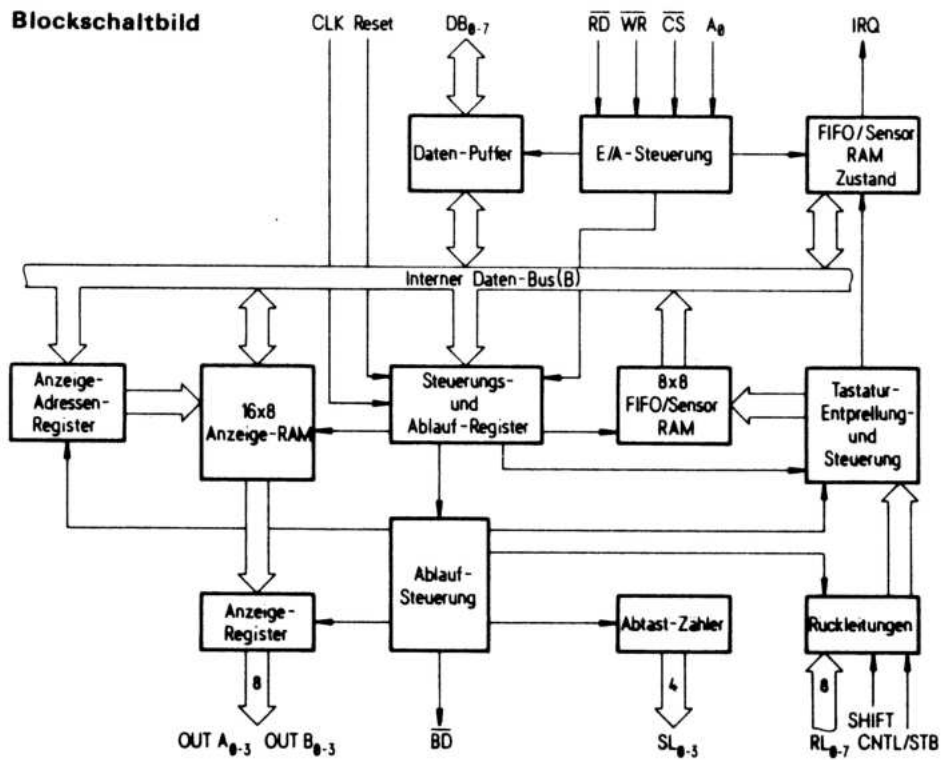


Bild 11



### 1.3.3 Programmierbarer serieller Schnittstellenbaustein 8251

(Kurzbeschreibung, Anschlußbelegung, Anschlußbezeichnung, Blockschaltbild)

Funktionen:

serieller Sender und Empfänger

zwei kaskadierte Register in Sender und Empfänger

Vollduplex-Betrieb

Betriebsarten:

Asynchronbetrieb

Synchronbetrieb

Eigenschaften im Asynchronbetrieb:

Taktfrequenz wahlweise das 1-, 16- oder 64-fache der Übertragungsgeschwindigkeit, Störimpulse werden nicht als Anlaufschritte interpretiert

Länge des Sperrschritts wahlweise 1, 1 1/2 oder 2 Bit

Erzeugung des BREAK-Zustands

automatische Erkennung und Anzeige des BREAK-Zustands

Übertragungsgeschwindigkeit 0 - 19,2 KBaud

Eigenschaften im Synchronbetrieb:

wahlweise interne oder externe Zeichensynchronisation

automatisches Einschleiben von Synchronzeichen

Übertragungsgeschwindigkeit 0 - 64 KBaud

Zeichenlänge:

5 bis 8 Bit

Fehlererkennung: Paritäts-, Überlauf- und Sperrschritt-Fehler

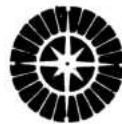
Alle Ein- und Ausgänge TTL-kompatibel

Nur ein TTL-Takt

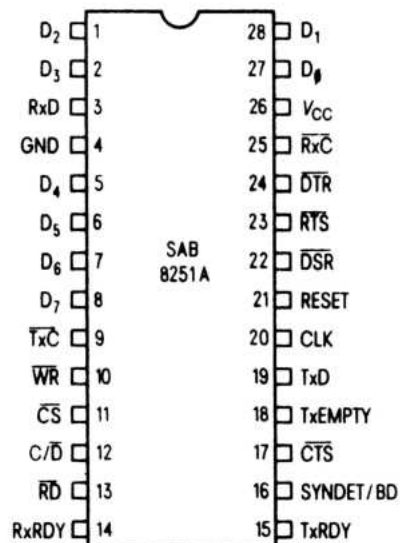
Voll-kompatibel mit den Mikroprozessoren SAB 8080 und SAB 8085

Nur eine Versorgungsspannung (5 V)





### Anschlußbelegung



### Anschlußbezeichnungen

D0 bis D7	Datenbus (8 Bit)
RESET	Rücksetzen
CLK	Systemtakt (TTL)
Cs	Baustein-Auswahl
C/D	Kennzeichnung der Parallel-Daten als Steuer- bzw. Status-Wort/Zeichen
RD	Daten oder Zustand lesen
WR	Daten oder Steuerinformationen schreiben
DSR	Betriebsbereitschaft
DTR	DE-Einrichtung betriebsbereit
RTS	Sendeteil einschalten
T x D	Serielle Sendedaten
T x C	Sendetakt
CTS	Sender-Freigabe
T x EMPTY	Senderegister leer
T x RDY	Sender bereit (kann Daten annehmen)
R x D	Serielle Empfangsdaten
R x C	Empfangstakt
SYNDET/BD	Synchronisations-signal/-erkennung und BREAK-Erkennung
R x RDY	Empfänger bereit (kann Daten liefern)
V <sub>cc</sub>	Versorgungsspannung (+ 5 V)
GND	Masse (0V)

Bild 12



**Blockschaltbild**

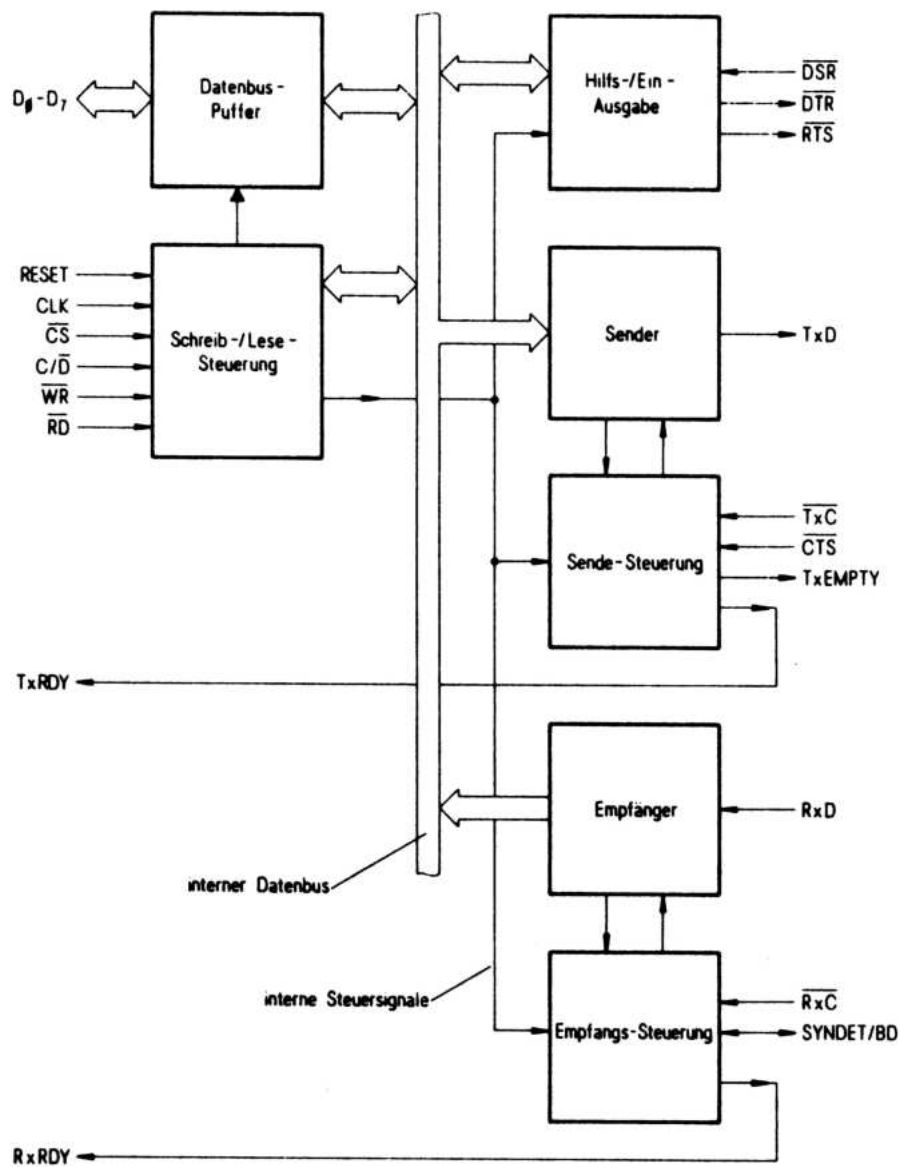


Bild 13



#### 1.3.4 Prioritäts-Steuerbaustein für Unterbrechungen 8214

(Kurzbeschreibung, Anschlußbelegung, Anschlußbezeichnung, Blockschaltbild)

Acht Prioritätsebenen

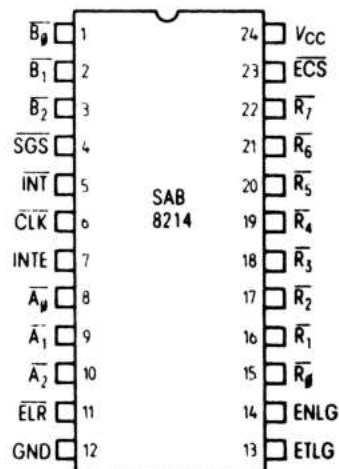
Register für die jeweils laufende Prioritätsebene

Prioritätsvergleicher

Erweiterungsmöglichkeit

Nur eine Versorgungsspannung (5 V)

##### Anschlußbelegung



##### Anschlußbezeichnungen

###### Eingänge

$\overline{R_0}-\overline{R_7}$	Anforderungseingänge ( $\overline{R_7}$ hat höchste Priorität)
$\overline{B_0}-\overline{B_2}$	laufende Programmebene
$\overline{SGS}$	Ausschalten des Vergleichers
$\overline{ECS}$	Übernahme der laufenden Programmebene
$\overline{INTE}$	Unterbrechungsfreigabe
$\overline{CLK}$	Takt (für $\overline{INT}$ -Flipflop)
$\overline{ELR}$	Leserfreigabe für Unterbrechungsebene <sup>2)</sup>
$\overline{ETLG}$	Freigabe dieser Ebenengruppe <sup>2)</sup>

###### Ausgänge

$\overline{A_0}-\overline{A_2}$	Anforderungsebenen (kodiert) <sup>1)</sup>
$\overline{INT}$	Unterbrechungsimpuls (aktiv mit L-Pegel) <sup>1)</sup>
$\overline{ENLG}$	Freigabe der nächsten Ebenengruppe
$V_{CC}$	Versorgungsspannung (+ 5 V)
$\overline{GND}$	Masse (0 V)

<sup>1)</sup> offener Kollektor

<sup>2)</sup> zur Kaskadierung

**Bild 14**



### Blockschaltbild

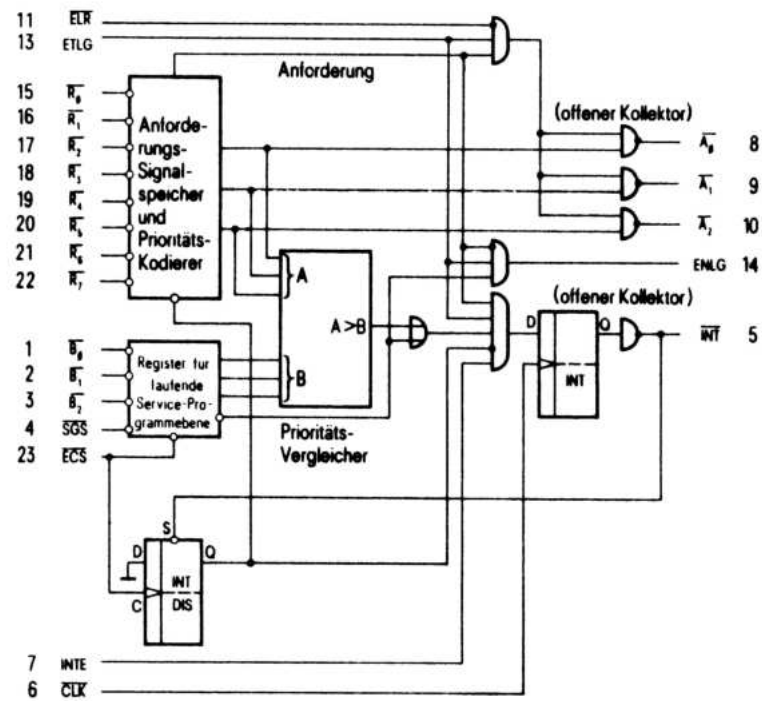
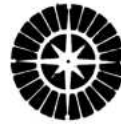


Bild 15



## 2. Erweiterte Anwendungen des "micromaster" über die Anwenderschnittstelle

Im folgenden werden wir ausführlich die Erweiterungsmöglichkeiten des "micromaster" über die Anwenderschnittstelle behandeln. Die in dem Gerät enthaltene Software ist weitgehend dafür vorbereitet. Für kleinere Versuchsaufbauten lassen sich Experimentierboards vom Typ protoboard von Firma AP-Products (Typenreihe ACE), die in entsprechenden Fachgeschäften erhältlich sind, sehr gut verwenden. Über eine 64polige Messerleiste (Stecker), an die die erforderlichen Leitungen angelötet werden, kann die Versuchsschaltung mit dem "micromaster" verbunden werden. Ansonsten sind kleinere Aufbauten auch in Fädertechnik auf Lochrasterplatten einfach herzustellen.

### 2.1 Funktions- und Anschlußbeschreibung für den Port-Baustein 8155 mit Anwendungen

Die in dem Anhang enthaltene Schaltung zeigt eine Möglichkeit, wie 2 Port-Bausteine vom Typ 8155 an den Mikroprozessor über die Anwenderschnittstelle angeschlossen werden können. Jeder Baustein enthält drei Funktionen

3 Ein-/Ausgabe-Kanäle (Ports)  
1/4 K RAM-Speicher und  
eine Zählereinheit (Timer)

Die Funktion (Mode) des Bausteins ist vom Mikroprozessor aus programmierbar und wird durch die Übergabe eines 8-Bit-Kommandowortes in das Kommandoregister festgelegt. Rückwärts läßt sich der Zustand des Bausteins durch Auslesen eines Statuswortes abfragen. Der RAM-Bereich wird normal wie jeder andere Speicherbereich auch angesprochen. Die Ports und der Zähler werden über sogenannte Kanal-Adressen erreicht (siehe Tabelle). Es werden dabei die IN- und OUT-Befehle aus dem Befehlssatz verwendet.

Der 8155 verfügt über 22 Leitungen zum Anschluß von Peripherie, die zu 3 Gruppen (Kanäle) zusammengefaßt sind:

Kanal A	8 Bit
Kanal B	8 Bit
Kanal C	6 Bit

Alle lassen sich durch ein Kommandowort so programmieren, daß sie als Ein- oder Ausgänge betrieben werden können. Werden die Kanäle A und/oder B im Hand-



shake-Betrieb gefahren, so werden 3 Anschlüsse des C-Kanals als Steuersignale verwendet. Daraus ergeben sich 4 Betriebsarten für die Kanäle A und B:

a. Eingabe ohne Rückmeldung

Der Mikroprozessor liest die zum Abfragezeitpunkt anstehende Information an den Peripherieanschlüssen in den Akkumulator ein (IN-Befehl).

b. Ausgabe ohne Rückmeldung

Der Mikroprozessor übergibt aus dem Akku das Datenwort in ein Zwischenregister im 8155. Von hier wird es an die Peripherieanschlüsse zur Ausgabe gegeben (OUT-Befehl).

c. Eingabe mit Rückmeldung

Das Datenwort von den Peripherieanschlüssen wird mit einem Übergabesignal (Strobe) in ein Zwischenregister 8155 geschrieben. An die Peripherie wird ein Steuersignal über Port C geliefert, welches anzeigt, wann das Datenwort vom Mikroprozessor abgeholt wurde (acknowledge  $\hat{=}$  Quittung). Es gibt außerdem noch das Steuersignal "Interrupt anfordern". Wird dieses von dem Peripherieanschluß auf den Interrupt-Eingang des Prozessors gelegt, kann dieser veranlaßt werden, das Datenwort sofort aus dem Zwischenregister des 8155 abzuholen.

d. Ausgabe mit Rückmeldung

Diese Betriebsart funktioniert wie bei der Ausgabe ohne Rückmeldung. Zusätzlich werden jedoch über Port C drei Steuersignale erzeugt. Mit dem "Strobe"-Signal wird der Peripherie mitgeteilt, daß ein neues Datenwort im Zwischenregister zur Abholung zur Verfügung steht. Über das Steuereingangssignal "acknowledge" (Quittung) erklärt die Peripherie dem Mikroprozessor, daß sie das Datenwort aus dem Zwischenregister abgeholt hat.

Das Steuerausgangssignal "Interrupt anfordern" wird erzeugt, wenn die Abholung eines Datenwortes quittiert wurde. Damit kann eine Programmunterbrechung erzeugt und der Prozessor zum Bereitstellen eines neuen Datenwortes veranlaßt werden.

Der Aufbau der Kommandoworte, die die Funktion des 8155 bestimmen sind in der Tabelle 1 und die erforderlichen Adressen in der Tabelle 2 zusammengestellt.

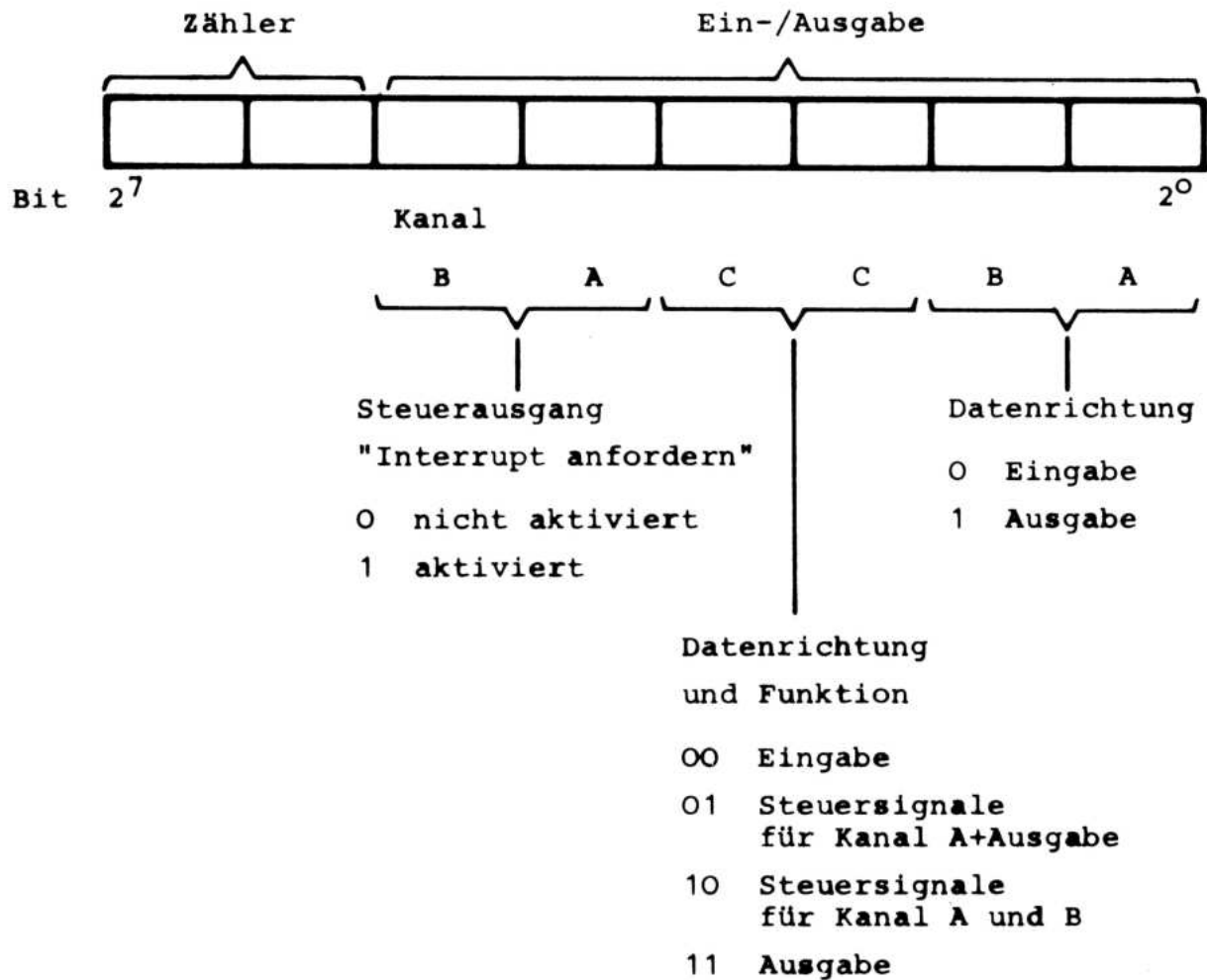


Tabelle 1

Aufbau eines Kommandowortes  
für die Port-Funktionen



A d r e s s e	Informationsfluß		
	Ausgabe zum 8155	Eingabe vom 8155	
F 8	Kommandowort	Status	8 1 5 5
F 9	Datenwort Kanal A	Datenwort Kanal A	
F A	Datenwort Kanal B	Datenwort Kanal B	
F B	Datenwort Kanal C	Datenwort Kanal C	
F C	Modewort niederwertiges Byte		
F D	Modewort höherwertiges Byte		
F E			
F F			

Tabelle 2

Adressen für Baustein 8155  
nach Schaltung im Anhang 6.1

Der Zustand (Status) der Ports eines 8155 kann jederzeit durch den Mikroprozessor abgefragt werden. Die möglichen Zustände werden in der Tabelle 3 erläutert.





Tabelle 3.0: Port-Status 8155

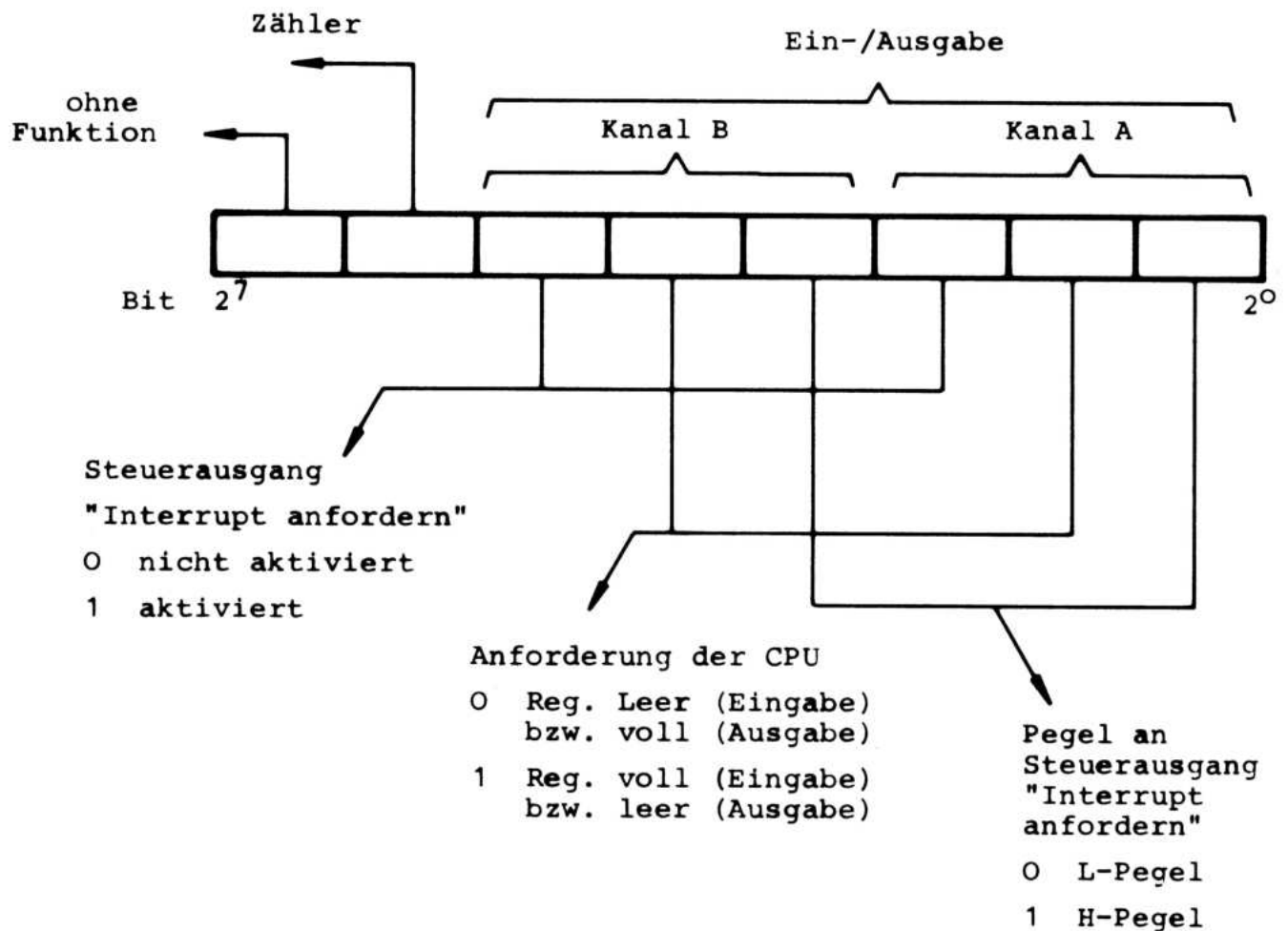
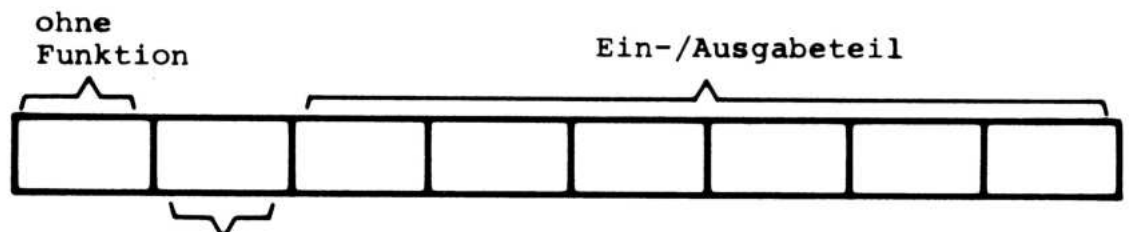


Tabelle 3.1: Zähler-Status 8155



**Zähler-Status:**

Wird Log. "1" mit der letzten fallenden Flanke des Eingangssignales während des Zählzyklusses.

Wird Log. "0" nach dem Einlesen des Statuswortes.



Beispiel: Zur Demonstration einer Ein- bzw. Ausgabe sei folgende Aufgabenstellung gegeben. Über den Peripherieanschluß 0 des Kanals A (8155) soll die Stellung eines Schalters (offen/geschlossen) abgefragt werden (Eingabefunktion). Die Schalterstellung soll eine Leuchtdiode (LED) anzeigen (Ausgabefunktion). Die erforderliche Hardware zeigt das folgende Bild 16. Die LED wird über ein NAND-Gatter 7400 von Pin 0 des Kanals B angesteuert.

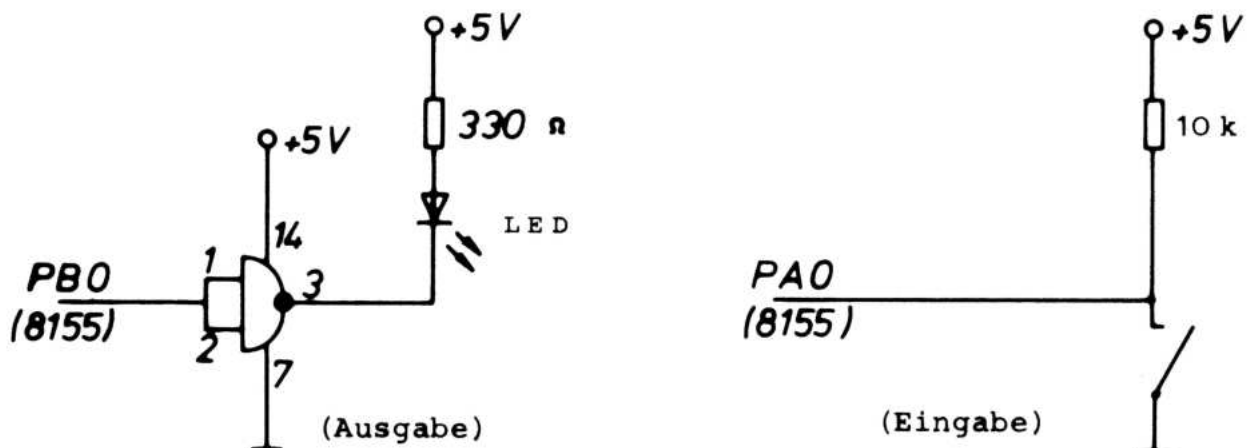


Bild 16

Schaltung zum Port-Ein/Ausgabebeispiel

Das erforderliche Flußdiagramm sieht folgendermaßen aus (Bild 17):

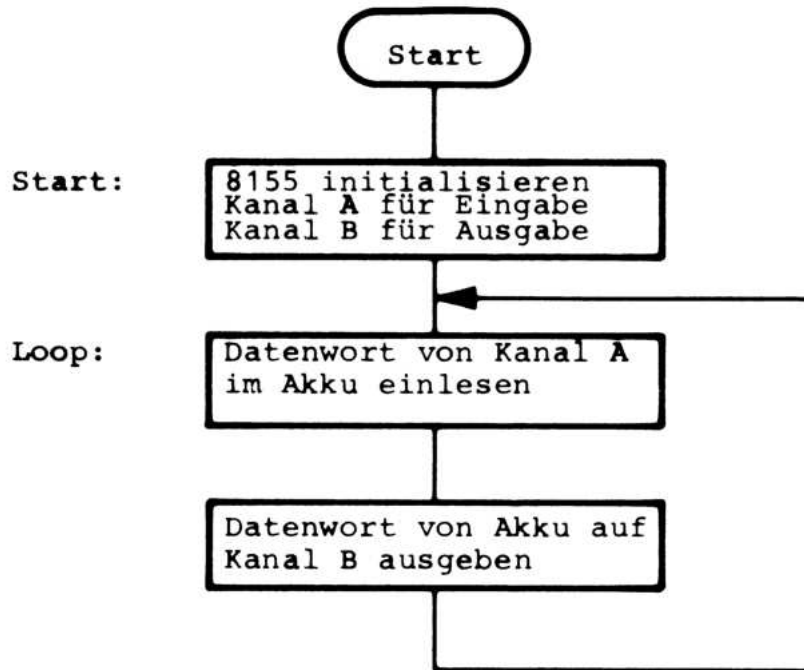


Bild 17

Das Programm ist äußerst einfach und besteht nur aus 5 Befehlen:

```
*****
*****
; DIESES PROGRAMM IST EIN DEMONSTRATIONSBEISPIEL FÜR
; DIE EIN-/AUSGABE-FUNKTION DES BAUSTEINS 8155. ES SOLL
; EIN SCHALTER ÜBER PORT A ABGEFRAGT WERDEN. SEIN ZU-
; STAND OFFEN/GESCHLOSSEN WIRD ÜBER PORT B AUSGEGEBEN
; UND ÜBER EINE LEUCHTDIODE ANGEZEIGT. NÄHERES SIEHE
; SCHALTUNG UND BESCHREIBUNG IM TEXT.
*****
*****
1800      ORG 1800H
1800 3E02  START:  MVI      A,02H      ;KOMMANDOWORT ZUR INITIALISIE-
1802 D3F8                OUT      0F8H  ;RUNG DES 8155
1804 DBF9  LOOP:   IN       0F9H      ;AUSGABE DES KOMMANDOWORTES AN
1806 D3FA                OUT      0FAH  ;DEN 8155 NR.1
1808 C3041B              JMP      LOOP ;SCHALTERSTELLUNG ÜBER KANAL A
;EINLESEN, DATENWORT IN AKKU
;DATENWORT AUS AKKU ÜBER KANAL
;B WIEDER ZUR ANSTEUERUNG DER
;LED AUSGEBEN
;SPRUNG ZUM SCHLEIFENANFANG
;DER MICROMASTER KANN NUR DURCH
;EINEN RESET AUS DIESER SCHLEI-
```



## K

Aufgabe 3: Wie lautet der Befehl, wenn vom 8155 Port A ein Datenwort in den Akku eingelesen werden soll?

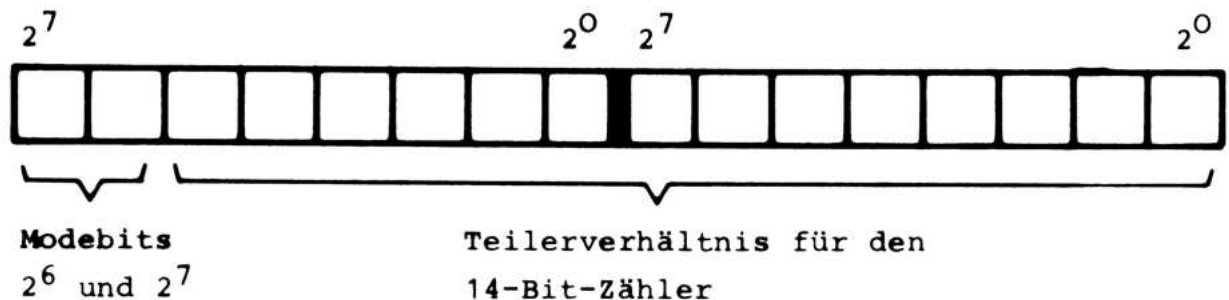
### 2.2 Programmierung der Zählerfunktion im 8155

Im 8155 ist ein 14-Bit-Zähler integriert, der als Ereigniszähler oder periodischer Teiler arbeiten kann. Zu dieser Einheit gehören auch ein 16-Bit-Zwischenregister, ein 16-Bit-Zählerregister und eine Logikschaltung zur Erzeugung des Ausgangssignals.

Die Betriebsarten des Zählers sind wieder über ein Kommandowort und ein Modewort (Bits 6 und 7) von der CPU aus programmierbar. Im Modewort, das ins Zählerregister übergeben wird, bestimmen die zwei höchstwertigen Bits die Funktion, die restlichen 14 Bits das Teilerverhältnis. Über das Kommandowort erfolgt das Starten und Stoppen des Zählers. Weitere Funktionsbeschreibungen finden Sie in der folgenden Tabelle 4, die zugehörigen Adressen in Tabelle 2:

Tabelle 4:

Modewort des 8155 (16 Bit)

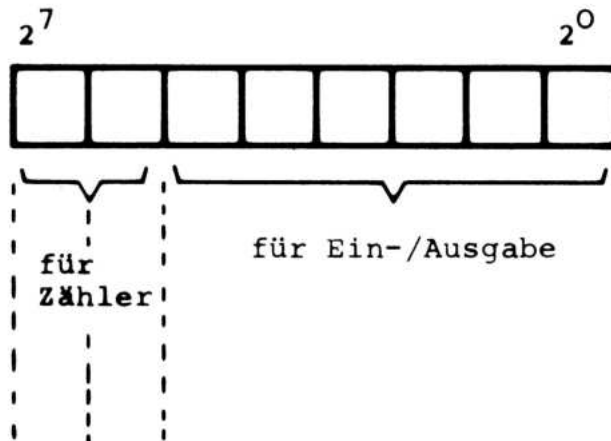


	Funktion Bit $2^6$	Funktion Bit $2^7$
0	einmaliges Zählen	Symmetrischer Teiler. Der Zählerausgang ist während der ersten halben Periode auf High-Pegel, während der zweiten auf Low-Pegel
1	periodisches Zählen	Zählerausgang auf High-Pegel während des Zählvorgangs. Am Ende für eine Periode des Eingangssignals auf Low-Pegel



Tabelle 5:

Kommandowort des 8155 (8 Bit, 2 Bit für Zählerteil)



0	0	Keine Funktion
0	1	Zähler stoppen
1	0	Zähler am Ende des gerade durchlaufenden Zählvorgangs stoppen
1	1	a. Zähler arbeitet nicht: Aus Zählerregister Zähler mit Mode und Teilverhältnis laden und starten. b. Zähler arbeitet: Nach Ende des Zählerdurchlaufs wie unter a. starten.

Auch für die Zählerfunktion im 8155 sei ein einfaches Beispiel angegeben:  
Der "micromaster" soll zusammen mit einem 8155 als Frequenzgenerator arbeiten.  
Hardware-Aufwand ist kaum erforderlich. Den Anschluß des 8155 kann man der  
Schaltung im Anhang entnehmen. Die nachfolgende Schaltung in Bild 18 zeigt,  
welche Pegel abgenommen werden können.

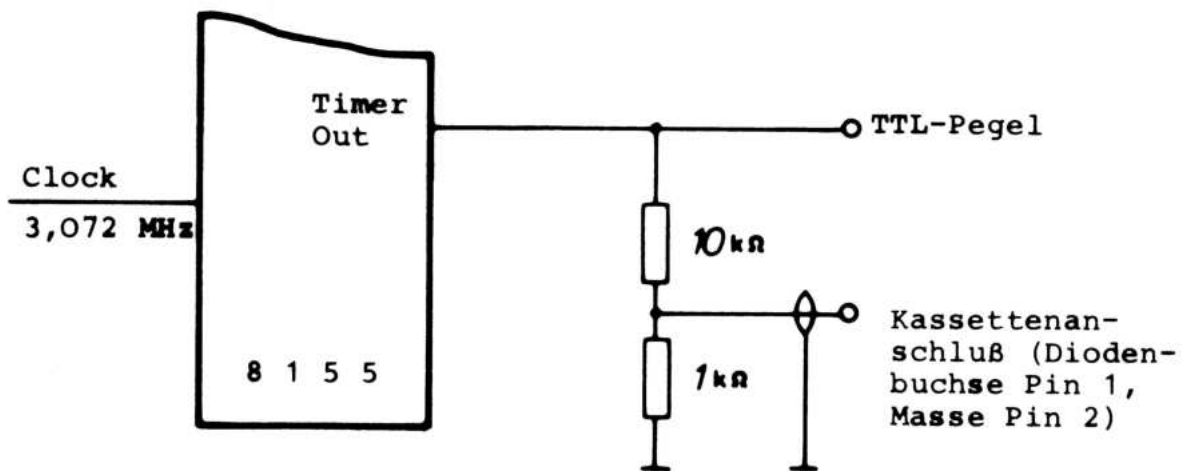


Bild 18

Über Tasteneingabe soll die ausgesendete Frequenz veränderbar sein: Ausgangsfrequenz = 3 kHz geteilt durch Wert des Tastenwortes.

Taste 1	3 000 Hz
Taste 2	1 500 Hz
Taste 3	1 000 Hz
.	.
.	.
.	.
.	.
Taste E	214 Hz
Taste F	200 Hz

In Bild 19 finden Sie das dazugehörige Flußdiagramm und danach folgend das Programmlisting.

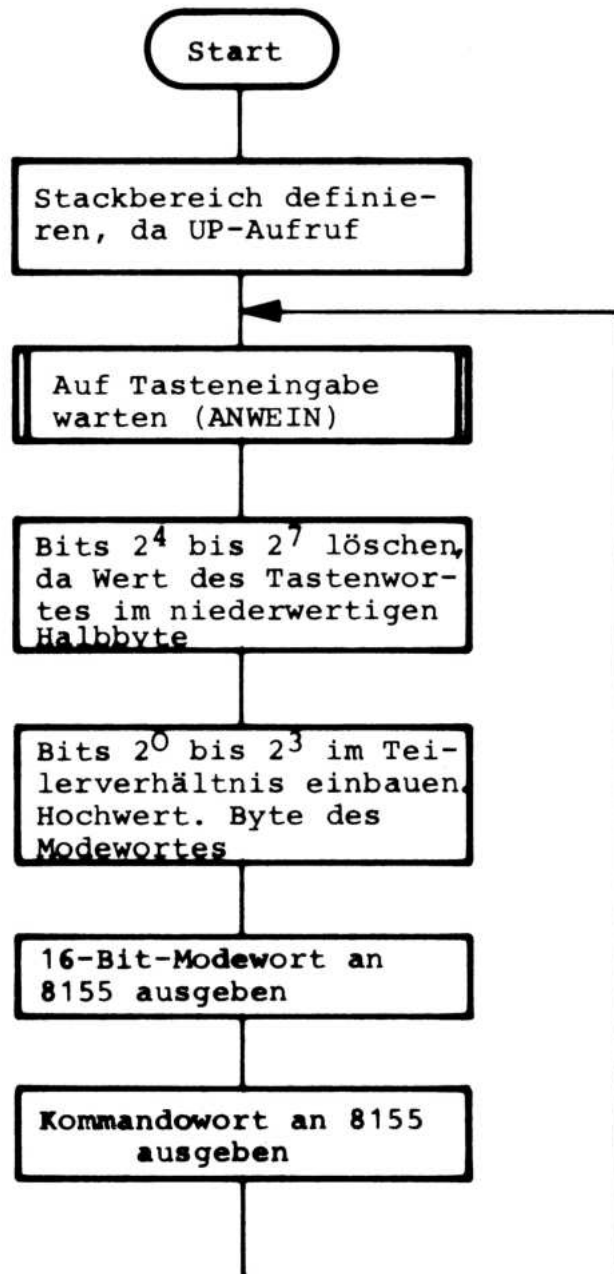


Bild 19



```

; *****
; *****
; DIESES IST EIN PROGRAMM ZUR DEMONSTRATION DER ZAEH-
; LERFUNKTION IM 8155. ES SOLL EIN FREQUENZGENERATOR
; PROGRAMMIERT WERDEN, DER AM ZAEHLERAUSGANG EINE
; VON EINER TASTENEINGABE ABHAENGIGE FREQUENZ AUSGIBT.
; NAEHERE ANGABEN UND SCHALTUNG SIEHE TEXT IN DER LEK-
; TION.
; *****
; *****
1800      ORG      1800H

07D1 =    ANWEIN  EQU 07D1H

1800 31EB1B  START: LXI      SP,1BEBH;STACKBEREICH FESTLEGEN
1803 CDD107      CALL     ANWEIN  ;AUF TASTENBETAETIGUNG WARTEN
                                ;WENN TASTE GEDRUECKT, TASTEN-
                                ;WORT IN AKKU EINLESEN
1806 E60F      ANI      0FH      ;BIT 4 BIS 7 AUSBLENDEN, DA
                                ;WERT DER TASTE IM NIEDERWER-
                                ;TIGEN HALBBYTE
1808 07      RLC      ;BIT 0 BIS 3 IN TEILERVERHAELT-
1809 07      RLC      ;NIS EINBAUEN, HOCHERT. BYTE
                                ;DES MODEWORTES
180A C640      ADI      40H      ;
180C D3FD      OUT     0FDH      ;16-BIT-MODEWORT AUSGEBEN
180E 3E00      MVI     A,00H      ;AN 8155
1810 D3FC      OUT     0FCH      ;
1812 3EC0      MVI     A,0C0H      ;KOMMANDOWORT AUSGEBEN AN
1814 D3FB      OUT     0FBH      ;8155
1816 C30018     JMP     START    ;SPRUNG AN DEN PROGRAMMANFANG
                                ;DIESES PROGRAMM KANN NUR UEBER
                                ;DIE RESERT-TASTE VERLASSEN
                                ;WERDEN, DA DER PROZESSOR IN
                                ;EINER ENDLOSSCHLEIFE ARBEITET

```

## M

Durch Aussenden von Kommando- bzw. Modeworten sind die Funktionen des 8155 veränderbar. Sein aktueller Zustand kann über das Statuswort abgefragt werden.





### 2.3 Der Umgang mit dem Schnittstellenbaustein 8279

Das Anzeigefeld und das Bedienungstastenfeld des "micromaster" sind über den Schnittstellenbaustein 8279 mit dem Bus des Mikroprozessors 8085 verbunden und entlasten ihn von Ein-/Ausgabefunktionen mit diesen Einheiten. Auch der 8279 ist ähnlich wie der 8155 in seinen Funktionen programmierbar und der jeweilige Status kann vom Prozessor abgefragt werden.

Prinzipiell enthält der 8279 die zwei Betriebsfunktionen "Tasten abfragen" und "Anzeigen steuern". Diese lassen durch Übergabe von Kommandoworten wieder verschiedene Betriebszustände zu. Der Zusammenhang des Informationsflusses über die beiden Adressen des 8279 im "micromaster" zeigt die Tabelle 6:

Tabelle 6:

Adresse	Funktion Ausgabe an 8279	Funktion Eingabe an 8279
EE	Datenwort	Datenwort
EF	Kommandowort	Statuswort



Tabelle 7:

<b>Tastenbezeichnung</b>	<b>vom 8279 an den Mikroprozessor über- gebener Wert (Tastenwort)</b>
0	00
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	08
9	09
A	0A
B	0B
C	0C
D	0D
E	0E
F	0F
L	10
R	11
-	12
+	13

Zusammenhang zwischen Tastenbezeichnung  
und Tastenwort



Sollen die Segmente der 7-Segment-Anzeigen in beliebigen Mustern angesteuert werden, sind in dem zur Anzeige übergebenden Datenwort die entsprechenden Bits mit 1 zu setzen, Bild 20.

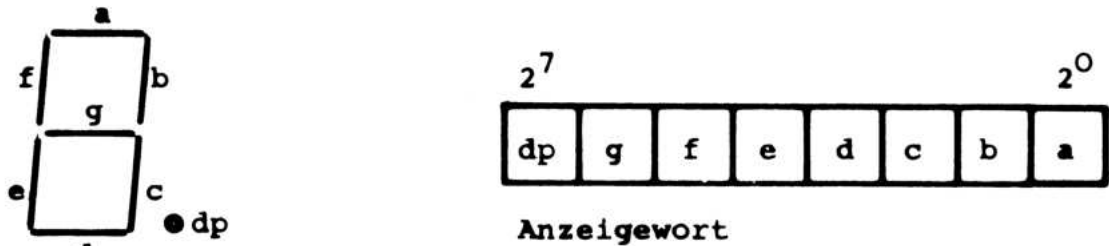


Bild 20

Die Ausgabe des Anzeigewortes erfolgt durch Aufruf des Unterprogramms ANWAUS. Sehen Sie sich bitte die Beschreibung dieses Programms in Lektion 3 nochmals an. Da die Funktionen des 8279 teilweise recht kompliziert sind, müssen wir an dieser Stelle für Einzelheiten wieder in das Datenblatt verweisen. Um Ihnen den Umgang mit dem 8279, bzw. Anzeige und Tastatur möglichst einfach zu machen, stehen Ihnen die in Lektion 3 abgedruckten Unterprogramme zur Verfügung.

#### 2.4 Möglichkeiten der RAM-Speichererweiterung über die Anwenderschnittstelle des "micromaster" um 1KByte

Im Grundausbau des "micromaster" stehen auf der Platine 1K RAM-Speicher zur Verfügung. Das sind die Bausteine 2114 mit einer Speicherorganisation von 1K x 4 Bit, d.h. zwei Stück parallel adressiert, ermöglichen den Aufbau 1K x 8 Bit.

Da der "micromaster" bereits für den Anschluß von peripheren Funktionsbausteinen vorgesehen ist, stehen alle Steuer-, Daten- und Adreßsignale an der Anwenderschnittstelle zur Verfügung. Wird beispielsweise ein 8155 angeschlossen, so steht zusätzlich 1/4K RAM-Speicher zur Verfügung. Der Adreßbereich liegt von FF00H bis FFFFH.

Betrachten wir jetzt zuerst die Möglichkeit den gesamten adressierbaren Speicherbereich von 64K extern außerhalb des "micromaster" aufzubauen. Zu diesem Zweck ist es erforderlich in dem Gerät RAM- und EPROM-Bereich abzuschalten.



Das erfolgt durch Auftrennen der beiden kleinen Brücken (Mitte, links neben dem Baustein 74S472). Da diese Speicherbereiche dann kein chip select-Signal (CS) mehr erhalten, sind sie nicht mehr ansprechbar. In der Schaltung (Lektion 2, Anhang) sind diese Brücken mit BR1 und BR2 gekennzeichnet.

Wie ist nun ein externer RAM-Speicherbereich aufzubauen? Wir brauchen dazu folgende Hilfsmittel: Die Anschlußbelegung des zu verwendenden Speicherbausteines (hier 2114), die Belegung der Anwenderschnittstelle und eine Tabelle der auf dem "micromaster" bereits erzeugten CS-Signale. Alles können Sie den Schaltungsunterlagen, bzw. dem Anhang dieser Lektion entnehmen.

Für einen 1K-RAM-Baustein sind 10 ( $2^{10} = 1024$ ) Adreßleitungen erforderlich, A0 bis A9. Es müssen 8 Datenleitungen zur Verfügung stehen, d.h. der Speicher muß eine Breite von 8 Bit haben, entsprechend einem Datenwort. Beim 2114 erfolgt die Ansteuerung zum Lesen (read,  $\overline{RD}$ ) oder Schreiben (write,  $\overline{WR}$ ) an den Anschlüssen (write enable,  $\overline{WE}$ ) und Bausteinanwahl (chip select,  $\overline{CS}$ ).

Auf die einzelnen Probleme, wie z.B. zeitliche Erfordernisse beim Speichern oder lesen, bei der Entwicklung von Speicher-Hardware können wir hier nicht eingehen und verweisen auf die Datenbücher.

Das Bild 28 (Anhang) zeigt die Schaltung des externen Zusatz-RAM-Speichers. Eine solche Erweiterung wird für ein Lern- und Übungsgerät jedoch nur in Ausnahmefällen erforderlich sein, da in der Regel der Bereich mit 1K ausreicht. Erst wenn neben einem umfangreichen Programm auch größere Datenmengen gespeichert werden sollen, wird eine Erweiterung sinnvoll.

## 2.5 Anschluß eines EPROM-Programmiergerätes an den "micromaster"

Manchmal kann es erforderlich sein, Anwenderprogramme nicht erst von der Kassette laden zu müssen, sondern diese bereits im Speicher verfügbar zu haben. Solche Programme müssen dann in einem EPROM abgelegt werden.

Im folgenden wird ein Schaltungsaufbau erläutert, der es ermöglicht, Informationen in ein EPROM (2716) zu schreiben und auch auszulesen. Der "micromaster" ist für den Anschluß dieser Schaltung vorbereitet. Zum Programmieren des EPROM kann das Kommando 2 verwendet werden. In einen solchen Speicherbaustein kann der Schreibvorgang natürlich nur einmal vorgenommen werden. Dabei werden die beim Löschvorgang erzeugten 1-Zustände, falls das entsprechende Bit programmiert werden soll, in den 0-Zustand gebracht. Das EPROM kann dann beliebig oft gelesen werden.



Bild 29 (Anhang) enthält die vollständige Schaltung des Programmiergerätes. Alle Adreß-, Daten- und Steuerleitungen sind so bezeichnet, wie sie an der Anwenderschnittstelle wieder zu finden sind. Wenn das Netzteil, an dem der "micromaster" betrieben wird, mindestens 1 A bei 5 V liefert, kann diese Schaltung aus diesem mit versorgt werden. Wird ein zweites Netzteil verwendet, muß auf alle Fälle die Masse durchverbunden werden.

Als Fassung für diesen EPROM Nr. 2 (Nr. 1 ist der Monitor) hat sich ein Wechselsockel bewährt, bei dem die Kontakte beim Stecken und Ziehen des Bausteins mittels eines kleinen Hebels entriegelt werden können. Im gesteckten Zustand werden diese wieder verriegelt.

## M

Während des Steckvorganges ist immer die Reset-Taste zu drücken, damit durch Störungen der RAM-Speicherinhalt nicht verändert wird!

Zum Programmieren des EPROM's ist eine zusätzliche Programmierspannung von 25 V am Pin 21 erforderlich. Im Lese-Betrieb muß dieser Anschluß an +5 V liegen. Die Umschaltung erfolgt mit dem Schalter PROG. Die 25 V-Spannung kann aus einer Batterie oder einem Netzgerät erzeugt werden.

Der eigentliche Programmiervorgang läuft folgendermaßen ab: Vor dem Beschreiben des EPROM's muß die Programmierspannung eingeschaltet werden. Dann kann ein Speicherbereich aus dem RAM in das EPROM kopiert werden. Danach wird die Programmierspannung wieder abgeschaltet und PIN 21 an +5 V gebracht.

Das EPROM kann in beliebigen Teilbereichen beschrieben werden, d.h. der Anwender muß sich Notizen darüber machen, welche Bereiche er bereits belegt hat. Es ist sinnvoll, ein Programm erst dann im EPROM abzulegen, wenn es zuvor im RAM ausgetestet wurde. In diesem Zusammenhang muß auch auf den Gebrauch des Kommandos 3 hingewiesen werden. Wenn ein Programm im RAM ausgetestet wurde und Sprungbefehle enthält, so sind diese vor dem Übertragen in das EPROM umzurechnen auf den neuen Bereich.

## M

Ein fehlerhaft programmiertes EPROM kann nur durch völliges Löschen unter UV-Licht und Neuprogrammierung in den gewünschten Zustand gebracht werden!

Mit dieser Schaltung ist es auch möglich, ein bereits programmiertes EPROM zu kopieren. Dazu wird der Bereich in 4 Bereiche zu je 1/2KByte geteilt und dann in vier Schritten übertragen.



Adresse	0 8 0 0 - 0 9 F F	1. Teilbereich
	0 A 0 0 - 0 B F F	2. Teilbereich
	0 C 0 0 - 0 D F F	3. Teilbereich
	0 E 0 0 - 0 F F F	4. Teilbereich

Die Vorgehensweise ist folgende:

Zuerst das erste halbe KByte des Original-EPROMs in den RAM-Bereich kopieren. Dann wird das neue EPROM eingesteckt und der Bereich wieder zurückkopiert. Dabei muß auf das Umschalten der Programmierspannung und das Drücken der Reset-Taste beim EPROM-Wechsel geachtet werden. Ist das Original nur teilweise beschrieben, braucht natürlich auch nur dieser Bereich kopiert werden.

Als Anwendungsbeispiel kann man z.B. das Abspeichern der WARTE-Schleifen aus der Lektion 3 in ein EPROM betrachten. Diese Zeitschleifen werden in den Übungen ja häufig verwendet. Nach dem Ändern der Adressen in dem Programm WARTE kann es in das EPROM übertragen und beliebig oft aufgerufen werden ohne es jedes Mal neu laden zu müssen. Bei den aufrufenden Programmen ist dann aber auch die neue Zieladresse einzugeben.

- K** Aufgabe 4: Worauf müssen Sie achten, wenn Sie im Programm, welches ab Adresse 1800H lauffähig ist und Sprungadressen enthält, in einen Bereich ab Adresse 0800H verschieben wollen?

## 2.6 Unterbrechungs-(Interrupt-)Verarbeitung im "micromaster"

Bevor wir uns mit den speziellen Unterbrechungsmöglichkeiten im "micromaster" beschäftigen, wollen wir kurz das Interruptsystem des 8085 selbst erläutern. Es ist etwas umfangreicher als das des 8080.

**M**

Unter einer Unterbrechung versteht man, daß der Mikroprozessor beim Auftreten einer Unterbrechungsanforderung das gerade laufende Programm verläßt und der Programmzähler mit einer fest vorgegebenen Adresse geladen wird. Ab dieser Adresse steht die Unterbrechung-Bedienungsroutine (interrupt service routine) selbst oder es wird mit einem unbedingten Sprung auf diese an anderer Stelle stehenden Routine verzweigt.



Das Unterbrechungssystem des 8085 besteht aus den 5 Eingängen

## M

TRAP, RST 5.5, RST 6.5, RST 7.5 und INTR

Zu jedem Eingang gehören unterschiedliche Prioritäten und Sprungziele. D.h. wenn z.B. nach einem RST 5,5 ein RST 6.5 folgt, wird die Interruptroutine des RST 5.5 verlassen und die des RST 6.5 bearbeitet. Bei jedem Interrupt wird der Programmzähler in den Stackbereich geladen damit der Rücksprung nach Erledigung der Interruptroutine gewährleistet werden kann. Die folgende Tabelle enthält die Einzelheiten.

Tabelle 8:

Interrupt-Eingang	Priorität	Verzweigt bei Interrupt-anforderung nach Adresse
TRAP	1	0024H
RST 7.5	2	003CH
RST 6.5	3	0034H
RST 5.5	4	002CH
INTR	5	siehe Anmerkung

### Interruptprioritäten

Anmerkung: Der INTR-Eingang ist ein allgemeiner Eingang wie er auch beim 8080 vorhanden ist. Er wird mit dem letzten Taktzyklus einer Befehlsausführung oder im Halt-Zustand abgefragt. Während der Quittung (interrupt acknowledge, INTA) kann ein Restart- oder Call-Befehl zur Verzweigung in die Interruptroutine erfolgen.

Der TRAP-Eingang kann nicht maskiert, d.h. abgeschaltet werden. Eine Anforderung an diesem Eingang löst immer einen Sprung nach 0024H aus. Dagegen können RST 5.5, 6.5 und 7.5 durch den SIM-Befehl aus dem Vorrat des 8085 gesperrt werden. Man spricht vom maskieren dieser Interrupts. Der Akkumulatorinhalt wird von dem SIM-Befehl wie folgt verwendet:



- Bit 7: Serielles Ausgangsdatenbit (0 oder 1)  
am Anschluß SOD des 8085
- Bit 6: Eine 1 gibt die seriellen Ausgangsdaten  
frei, der Ausgangspuffer wird geladen.
- Bit 5: Nicht belegt
- Bit 4: Eine 1 setzt RST 7.5 zurück. Das Inter-  
rupt-Flip-Flop wird gelöscht.
- Bit 3: Wenn 1 können die Maskenbits für RST 7.5,  
6.5 und 5.5 geladen werden
- Bit 2: RST 7.5
- Bit 1: Interruptmaskierung für RST 6.5
- Bit 0: RST 5.5
- Wird eine 0 gesetzt, ist der Interrupt frei,  
eine 1 sperrt ihn.

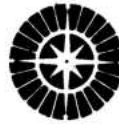
Der RIM-Befehl ermöglicht es den Interrupt-Status und den seriellen Eingang  
SID in den Akkumulator zu laden. Die einzelnen Bits haben folgende Bedeutung:

- Bit 7: Serielles Eingangsdatenbit (0 oder 1)  
vom Anschluß SID des 8085
- Bit 6: bestehende Interruptanforderung an RST 7.5
- 5: 0 = keine Anforderung RST 6.5
- 4: 1 = Anforderung RST 5.5
- Bit 3: Wenn 1, dann ist das gesamte Interruptsystem  
(außer TRAP) gesperrt.
- Bit 2: aktueller Interruptzustand RST 7.5
- 1: 0 = frei RST 6.5
- 0: 1 = gesperrt RST 5.5

Eine Anwendungsmöglichkeit der SID- bzw. SOD-Anschlüsse des 8085 ist zum Beispiel  
die Realisierung einer seriellen Schnittstelle, deren Übertragungsgeschwindig-  
keit von zu programmierenden Zeitschleifen zwischen Ausführung des RIM- bzw.  
SIM-Befehls abhängt.

Eine Anwendung des TRAP-Interrupteingangs liegt beispielsweise in der Überwa-  
chung von Versorgungsspannungen, da es in jedem Fall zu einer Interruptbearbei-  
tung kommt. Wenn der Prozessorzustand zum Zeitpunkt des Ausfalls unter allen Um-





ständen gesichert werden muß, um nach Behebung der Ursache das System ab da weiter arbeiten zu lassen, wo der Ausfall auftrat, sind folgende Voraussetzungen zu schaffen:

Batteriepufferung des RAM-Speichers  
(zumindest teilweise)

Halten der Versorgungsspannung am  
Prozessorsystem bis alle Register  
gesichert sind, z.B. durch Batterie  
oder großen Elko.

Programmroutine, die nach einem Ein-  
schalt-Reset die gesicherten Inhalte  
aus dem Stack in die Register lädt.

Wenn Sie dieses komplizierte Thema mit seinen vielfältigen Möglichkeiten noch näher interessiert, bitten wir Sie in dem Datenbuch des 8085 nachzulesen. Da sind insbesondere auch die zeitlichen Anforderungen an die Interruptsignale beschrieben. Eines beschreiben wir im Demonstrationsbeispiel zur Bearbeitung eines Interrupts RST 6.5.

Beispiel: In einem Hauptprogramm soll auf der Anzeige des "micromaster" der Buchstabe H (für Hauptprogramm) in der 8. Anzeigestelle ausgegeben werden. Nach Betätigung der Interrupttaste (RST 6.5) soll in eine Interruptroutine gesprungen werden. In dieser soll in der 1. Anzeigestelle der Buchstabe I (für Interrupt) angegeben werden.

Es ist dabei zu beachten, daß das Interruptprogramm in einem EPROM (siehe Abschnitt vorher) liegen muß, da von der Interruptsprungleiste für den TRAP, den RST 6.5 und 5.5, sowie die Software RST-Befehle in den Anwender-EPROM-Speicherbereich durch unbedingte Sprünge verzweigt wird (siehe Tabelle 9 ).



Tabelle 9

Interrupt	Zieladresse in Anwender- EPROM Nr. 2
TRAP	0824H
RST 6.5	0834H
RST 5.5	0832H
RST 6	0830H
RST 5	0828H
RST 4	0820H
RST 3	0818H
RST 2	0810H
RST 1	0808H

Bild 21 zeigt die Schaltung, mit der ein Interrupt RST 6.5 hervorgerufen werden kann.

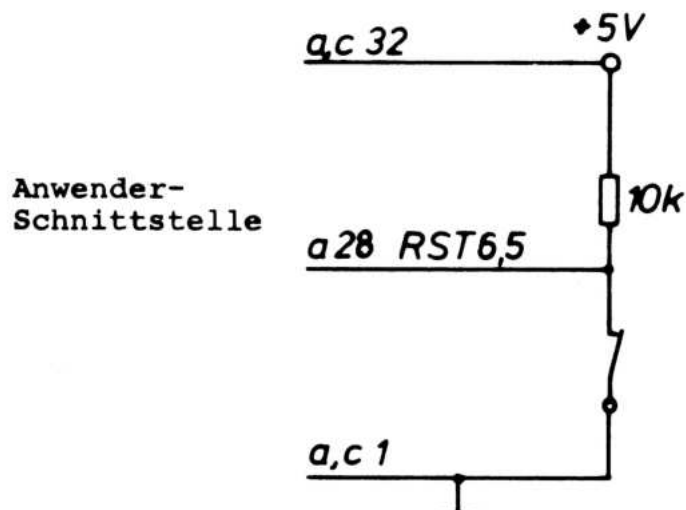
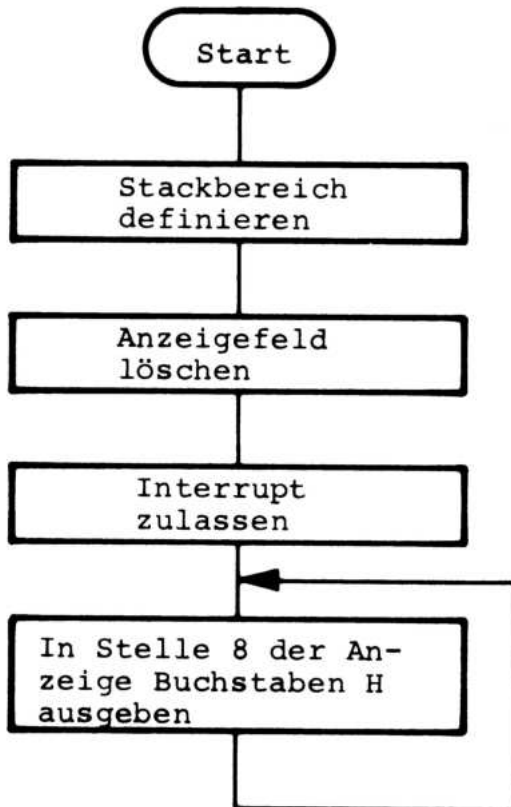


Bild 21

Für die angegebene Aufgabenstellung sind zwei Flußdiagramme erforderlich, je eins für das Hauptprogramm und die Interruptroutine (Bild 22).



Flußdiagramm  
Hauptprogramm



Flußdiagramm  
Interruptroutine

Bild 22



```
*****
*****
;DIESES IST EIN PROGRAMM ZUR DEMONSTRATION DER VERAR-
;BEITUNG DES INTERRUPTS RST 6.5. IN DEM HAUPTPROGRAMM
;SOLL IN DER 8. STELLE DES ANZEIGEFELDES DER BUCHSTA-
;BE H, MIT DER INTERRUPT-ROUTINE IN DER 1. STELLE DER
;BUCHSTABE I AUSGEGEBEN WERDEN. NAEHERES SIEHE TEXT IN
;DER LEKTION.
*****
*****

07BC =      ANWAZL EQU      07BCH
07C7 =      ANWAUS EQU      07C7H
1BE8 =      STACK EQU      1BE8H
1800 =      HAUPROG EQU     1800H
0834 =      INTR EQU       0834H

;-----
;HAUPTPROGRAMM
;-----
1800      ORG      HAUPROG

1800 31E81B  START: LXI      SP,STACK;STACKBEREICH DEFINIEREN
1803 CDBC07          CALL    ANWAZL  ;ANZEIGE WIRD GELOESCHT
1806 3E0E          MVI      A,0EH   ;WORT FUER INTERRUPT-FREIGABE
1808 30          SIM          ;IN AKKU LADEN UND AUSGEBEN
1809 FB          EI            ;INTERRUPT ZULASSEN
180A 3E76  LOOP:  MVI      A,76H   ;ANZEIGEWORT FUER BUCHSTABEN H
180C 0607          MVI      B,07H   ;IN AKKU LADEN UND AUSGEBEN
180E CDC707          CALL    ANWAUS ;
1811 C3091B          JMP     LOOP   ;WEITER BUCHSTABEN H AUSGEBEN
                                         ;UND AUF INTERRUPT WARTEN

;-----
;INTERRUPT-ROUTINE
;-----
0834      ORG      INTR

0834 3E06          MVI      A,06H   ;ANZEIGEWORT FUER BUCHSTABEN I
0836 0600          MVI      B,00H   ;IN AKKU LADEN UND AUSGEBEN
0838 CDC707          CALL    ANWAUS ;
083B FB          EI            ;INTERRUPT ZULASSEN
083C C9          RET              ;RUECKSPRUNG ZUM HAUPTPROGRAMM
```



Den Aufwand mit dem zweiten EPROM zur Aufnahme der Interruptroutine kann man sich sparen, wenn man die Möglichkeit hat, ein EPROM zu brennen. Z.B. im Betrieb o.ä. Die Interruptroutine kann auch im RAM ab 1900H eingegeben werden, wenn ab Adresse 0034H, der Zieladresse des RST 6.5, die drei Bytes

Adresse		
0 0 3 4 H	C 3 H	
0 0 3 5 H	0 0 H	(statt 3 4 H)
0 0 3 6 H	1 9 H	(statt 0 8 H)

einprogrammiert werden. Dann wird nicht in das Anwender-EPROM, sondern in den RAM-Bereich ab Adresse 1 9 0 0 H verzweigt.

- K** Aufgabe 5: Welcher Interrupteingang am 8085 muß verwendet werden, wenn z.B. ein Alarmsignal in einer Anlage zum sofortigen Maschinenstop führen soll?

### 3. Vertiefung der Programmierertechnik an zwei praktischen Beispielen

Anhand zweier praktischer Beispiele soll die Übung in der Mikroprozessortechnik weiter vertieft werden. Außerdem sei an dieser Stelle auch noch einmal auf das Buch "8080/8085 - Programmieren in Assembler" von L.A. Leventhal hingewiesen, das eine Vielzahl von weiteren Beispielen enthält. Es ist im tewi-Verlag, München, erschienen.

#### 3.1 Der "micromaster" als Frequenzmeßgerät

Frequenzmessungen werden in der Regel mit Meßgeräten durchgeführt, die durch eine Torschaltung die zu messende Frequenz in einen Zähler einlesen. Das Tor wird durch eine Zeitbasis gesteuert. Der Zählerinhalt wird angezeigt. Eine Ablaufsteuerung sorgt für die periodische Meßwiederholung und, falls vorhanden, die Übergabe des Ergebnisses an eine Rechnerschnittstelle.

Es ist jedoch auch eine Software-Lösung denkbar, bei der der 8085 zusammen mit einem 8155 und dem Interruptsystem arbeitet. Dabei wird der Zählereingang mit dem Taktausgang (clock) des 8085 verbunden und der Zählerausgang an den Interrupteingang RST 6.5 angeschlossen. Der Meßeingang kann einer der Porteingänge des Kanals A sein und muß durch 1 Bit im D-Register vor dem Meßbeginn bestimmt sein. Also muß für Eingang PA0 im D-Register der Inhalt 01 stehen (Bild 23).

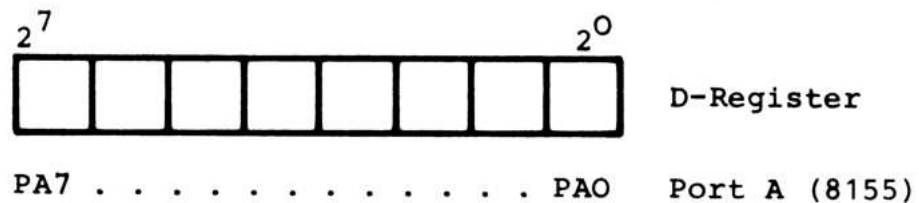


Bild 23

Da der "micromaster" mit einer Quarzfrequenz 6,144 MHz arbeitet, steht am Clock-Ausgang des 8085 die Frequenz 3,072 MHz. Wird der Zähler als periodischer Teiler mit einem Teilungsverhältnis von 15 360 eingestellt, so ergeben sich 200 Hz als Zählerausgangsfrequenz,  $3,072 \text{ MHz} : 15360 = 200 \text{ Hz}$ . Es können Frequenzen bis 20 kHz gemessen werden.

Das Programm kann selbsttätig ablaufen oder von einem anderen Programm aufgerufen werden. Das D-Register muß nicht mit dem MVI-Befehl geladen werden, sondern kann auch im aufrufenden Programm bestimmt werden. Der MVI ist dann an der jetzigen Stelle zu löschen.

In dem erforderlichen Programm sind zuerst vor der eigentlichen Meßroutine der 8155 und das Interruptsystem des 8085 zu initialisieren. Unter initialisieren versteht man das Programmieren der Bausteine auf die gewünschte Funktion durch Laden entsprechender Kommandoworte. Register E wird als Interruptzählregister verwendet und von 200 auf 0 durch jeden Interrupt aus dem Zählerausgang heruntergezählt. Daraus ergibt sich ein Meßintervall von 1 Sekunde entsprechend 1 Hz. Nach Ablauf eines Intervalls wird das Ergebnis im BC-Registerpaar übergeben.

Mit diesem Software-Frequenzmeßgerät lassen sich TTL-Signale bis zu 20 kHz messen, wobei in jedem Fall die Zeit eines H- oder L-Pulses 25 µs betragen muß. Es können bis zu 8 verschiedene Frequenzen an den 8 Anschlüssen des Port A gemessen werden, wenn eine entsprechende Programmierung im D-Register das gesetzte Bit weiterschaltet. Auch die Erweiterung auf 16 Meßkanäle ist unter Einbeziehung des Port B möglich.

Das Bild 24 zeigt den einfachen Hardware-Aufbau. Wenn ein 8155 bereits angeschlossen ist, sind nur zwei zusätzliche Leitungen und der Anschluß des Meßsignals erforderlich.

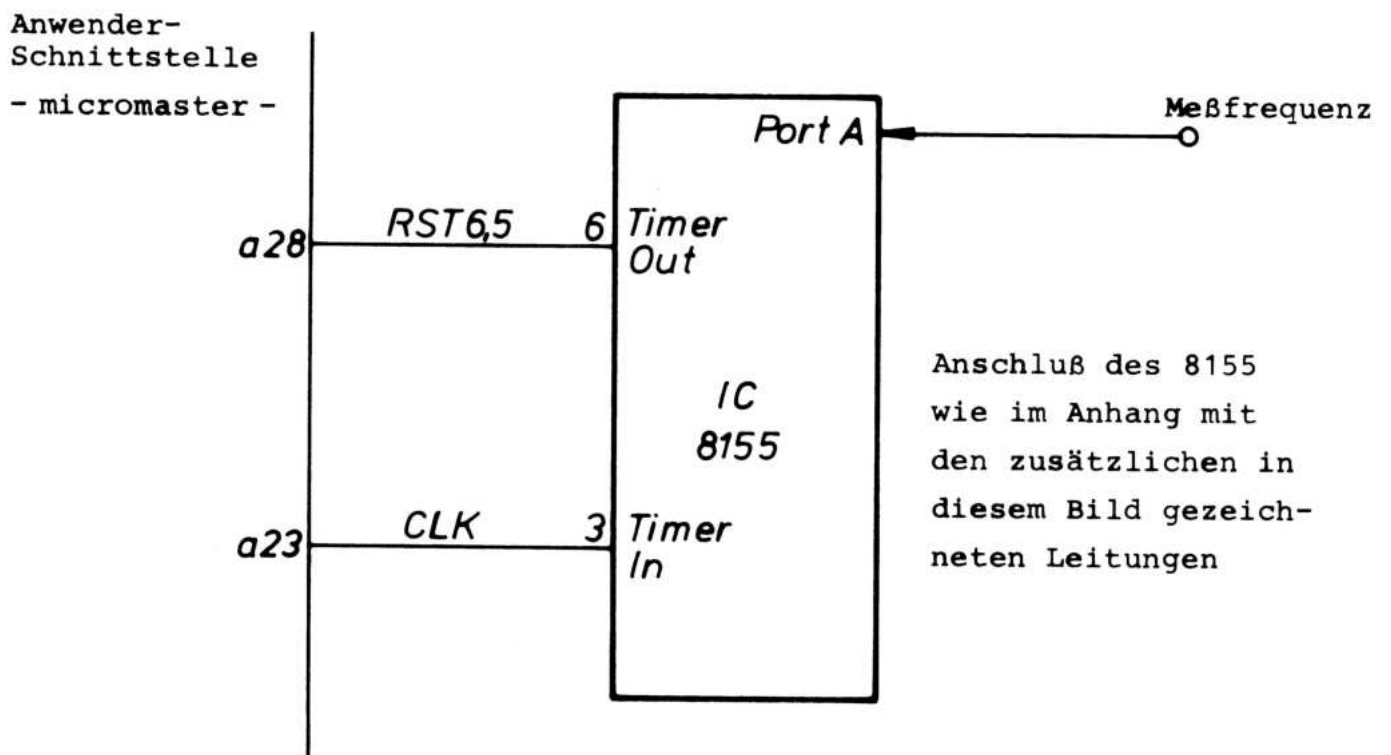
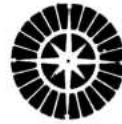


Bild 24

Frequenzmessung mit dem 8155



```

; *****
; *****
; DAS PROGRAMM *FMESS* IST EINE SOFTWARE-ROUTINE ZUR
; MESSUNG VON FREQUENZEN BIS 20 KHZ MIT TTL-PEGEL. ALS
; HARDWARE IST EIN 8085 ZUSAMMEN MIT EINEM 8155 ERFOR-
; DERLICH. DIE BEIDEN EINZIGEN ERFORDERLICHEN VERBIN-
; DUNGEN SIND ZWISCHEN DEM TAKT (CLK, 8085) UND ZAEH-
; LEREINGANG (8155), BZW. ZWISCHEN ZAEHLERAUSGANG (8155)
; UND INTERRUPTTEINGANG (RST6.5, 8085) ZU VERDRAHTEN.
;
; ALS MESSEINGANG KANN JEWEILS EINER DER EINGAENGE DES
; PORT A VOM 8155 VERWENDET WERDEN. DURCH SETZEN EINES
; BITS IM D-REGISTER WIRD DER EINGANG BESTIMMT. DER IM
; E-REGISTER EINGESCHRIEBENE WERT BESTIMMT DIE LAENGE
; DES MESSINTERVALLS. DAS ERGEBNIS WIRD IM BC-REGISTER-
; PAAR UEBERGEHEN.
;
; VERAENDERTE REGISTER: PSW,A,E
;
; WEITERE EINZELHEITEN SIEHE TEXT IN DER LEKTION
; *****
; *****

```

```

0001 = PAEIN EQU 01H
0008 = MESINT EQU 200D
1800 = FMESS EQU 1800H
0834 = INTR EQU 0834H
00F8 = KW8155 EQU 0F8H
00FD = MWHWB EQU 0FDH
00FC = MWNWB EQU 0FCH
00F9 = PA EQU 0F9H

```

```

; -----
; INITIALISIERUNG
; -----

```

```

1800 ORG FMESS

```

```

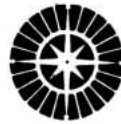
1800 1601 FMESS: MVI D,PAEIN ; EINEN DER 8 EINGAENGE DES
; PORT A AUSWAERHLEN
1802 1EC8 MVI E,MESINT; MESSINTERVALL FESTLEGEN
1804 010000 LXI B,0000H ; ERGEBNISREGISTER LOESCHEN
1807 3E40 MVI A,40H ; KOMMANDOWORT AN 8155
1809 D3F8 OUT KW8155 ; AUSGEBEN FUER ZAEHLERSTOP
180B 3E7C MVI A,07CH ; MODEWORT FUER SYMETRISCHE TEI-
180D D3FD OUT MWHWB ; LUNG MIT VERHAELTNIS 15360
; UND ALS HOCHWERT. MODEBYTE
; AUSGEBEN
180F AF XRA A ; AKKU LOESCHEN
1810 D3FC OUT MWNWB ; NIEDERWERT. BYTE DES MODE-
; WORTES AUSGEBEN
1812 3E1B MVI A,1BH ; BYTE FUER INTERRUPT-MASKE
1814 00 NOP ; LADEN
SIM ; INTERRUPT-MASKE SETZEN UND
; RST6.5 ZULASSEN ***KONTROLLE**
1815 3EC0 MVI A,0C0H ; KOMMANDOWORT FUER ZAEHLER
; STARTEN, PORT A AUF EINGABE
1817 D3F8 OUT KW8155 ; LADEN UND AUSGEBEN

```





```
;-----  
;MESSROUTINE  
;-----  
  
FMO:                                ;AUF LOW-PULS ABFRAGEN  
  
1819 FB          EI                ;INTERRUPT ZULASSEN  
181A DBF9        IN                ;PORT A LESEN  
181C F3          DI                ;ACHTUNG: BEI INTERRUPT WERDEN  
                                ;IN ROUTINE INTR ZUSTANDSBITS  
                                ;VERAENDERT  
181D A2          ANA               ;MIT MASKE VERKNUEPFEN  
181E C21918      JNZ              FMO ;WARTEN BIS MASKIERTES BIT=0  
  
FM1:                                ;AUF HIGH-PULS ABFRAGEN  
  
1821 FB          EI                ;WENN NICHT, INTERRUPT ZULASSEN  
1822 DBF9        IN                ;PORT A LESEN  
1824 F3          DI                ;SIEHE VORHER  
1825 A2          ANA               ;  
1826 CA2118      JZ               FM1 ;WARTEN BIS MASKIERTES BIT=1  
1829 03          INX              B  ;WECHSEL VON 0 NACH 1 IST AUF-  
                                ;GETRETEN, ERGEBNISREGISTER  
                                ;UM 1 ERHOEHEN  
182A C31918      JMP              FMO ;WIEDER AUF LOW-PULS ABFRAGEN  
  
;-----  
;INTERRUPTROUTINE  
;-----  
  
0834             ORG              INTR  
  
0834 1D          INTR:   DCR       E      ;MESSINTERVALL-ZAEHLER UM 1  
                                ;VERRINGERN  
0835 C0          RNZ                                ;IN MESSROUTINE SPRINGEN, WENN  
                                ;NOCH NICHT 0  
0836 33          INX       SP            ;SONST MESSUNG BEENDET  
0837 33          INX       SP            ;  
0838 3E40        MVI       A,040H      ;BYTE FUER ZAEHLERSTOP LADEN  
083A D3F8        OUT      KW8155      ;UND AUSGEBEN  
083C C9          RET                                ;RUECKSPRUNG INS AUFRUFENDE  
                                ;PROGRAMM
```



### 3.2 Beschreibung für den Anschluß eines Analog/Digital-Wandlers an den 8085

Da der Mikroprozessor nicht nur, wie bereits beschrieben, in Personalcomputern eingesetzt wird, sondern sein wesentlicher Einsatz in der Kontrolle und Steuerung von Prozessen ist, muß er über periphere Einheiten in der Lage sein, eine Vielzahl physikalischer Größen zu verarbeiten. Diese können nach entsprechender Wandlung analog oder digital vorliegen. Sollen sie im System verarbeitet werden, muß ein analoges Signal zuvor in ein gleichwertiges digitales Signal umgewandelt werden.

Als typisches Anwendungsbeispiel für den Mikroprozessor sei die Steuerung bzw. Regelung einer Heizungsanlage genannt. Die den Brenner beeinflussenden Größen sind die Temperaturen, die außen und in den Räumen über Temperaturfühler gemessen werden. Diese gemessenen Signale gelangen über Analog/Digital-Wandler in das Prozessorsystem. Wie der Anschluß eines solchen AD-Wandlers an ein System erfolgen kann, soll im folgenden beschrieben werden (nach Unterlagen der Fa. Ferranti GmbH., München).

#### 3.2.1 Beschreibung des Hardware-Aufbaus

Die nachfolgend beschriebene Schaltung finden Sie im Anhang als Bild 30. Sie zeigt ein 8085-Mikroprozessor-System, an dem über einen 8155 jeweils vier AD-Wandler vom Typ ZN 427 (Ferranti) angeschlossen sind.

Der 8155 übernimmt dabei als peripherer Schnittstellenbaustein die Entkopplung zwischen den Wandlern und dem System. Die Systemseite des 8155 wird mit dem 8085 so verbunden, wie es vorher im Abschnitt über Erweiterungsmöglichkeiten beschrieben wurde. Auf der Peripherieseite arbeitet der Port A als Eingabekanal für die gewandelten digitalen Signale. Port B und C übernehmen die zeitliche Steuerung der ZN 427. Der Systemtakt muß über einen Teiler durch 4 (z.B. zwei Flipflops) heruntergeteilt werden.

Der ZN 427 ist ein 8-Bit AD-Wandler, der nach dem Verfahren der sukzessiven Approximation (schrittweise Annäherung) arbeitet. Bei diesem Verfahren wird ein Zähler durch einen Takt hochgezählt, dessen Inhalt an einem Digital/Analog-Wandler liegt. Dieses Analogsignal wird mit dem zu messenden Analogsignal in einem Komparator (Vergleicher) verglichen. Wurde der Zähler so weit gezählt, daß die beiden Signale gleich groß sind, wird das nach außen signalisiert (EOC, end of conversion, Ende der Wandlung) und der digitale Zählerinhalt kann als eine dem



Analogsignal gleichwertige Information ausgelesen werden. Der Zähler wird gestoppt. Eine neue Wandlung wird eingeleitet mit dem Signal SC (start of conversion, Start der Wandlung), welches den Zählerinhalt rücksetzt (Bild 25).

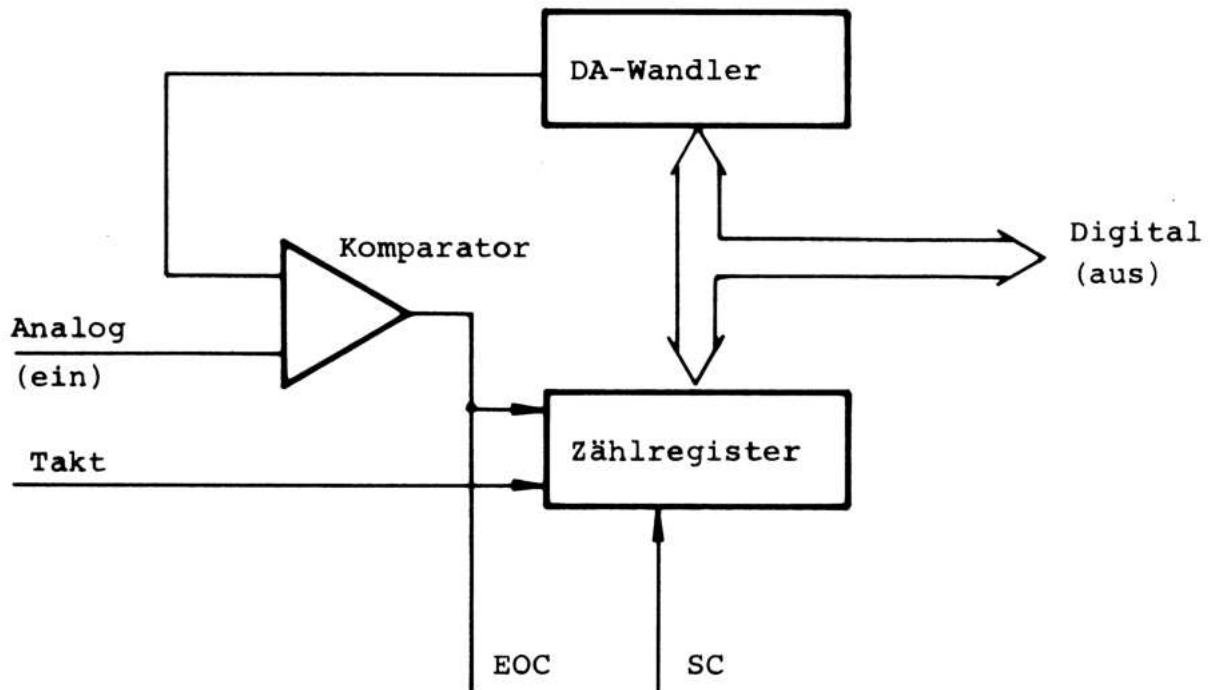


Bild 25

Analog/Digital-Wandlungsprinzip

Alle Hersteller von AD- und DA-Wandler haben heute Bausteine im Programm, die problemlos mit Mikroprozessoren zusammengeschaltet werden können, d.h. sie bieten gute Interface-Möglichkeiten. In unserem Beispiel werden die digitalen Ausgänge der ZN 427 mit dem Port A des 8155, der als Eingang programmiert wird, verbunden. Port B wird als Ausgang programmiert, wobei die 4 niederwertigen Bits die 4 SC-Signale und die 4 höherwertigen das Übernahmesignal (OE, output enable, Ausgang verfügbar) liefern. Vom Port C, der als Eingang programmiert wird, werden auf 4 Bits die EOC-Signale gelegt. Das Herunterteilen des Taktes und die D-Flipflops sind für das timing (zeitlicher Ablauf) der Wandlung erforderlich.

Die Wandlung wird eingeleitet, indem über Port B für das SC-Signal eine 0 gefolgt durch eine 1 ausgegeben wird. Nach 9 negativen Taktflanken am AD-Wandler geht das EOC-Signal nach log. 1 und wird durch Port C eingelesen. Natürlich kann die angegebene Schaltung auch nur mit einem ZN 427 betrieben werden. Der Software-Aufwand verringert sich ebenfalls entsprechend.



### 3.2.2 Die Software für das Lesen der AD-Wandler

Bevor der Start-Puls (SC) an die AD-Wandler ausgegeben werden kann, müssen zuerst der 8155 und der Stackbereich initialisiert werden. Die Einstellung der Ports wurde vorher beschrieben. Nach dem Start müssen die EOC-Signale über Port C solange abgefragt werden, bis alle AD-Wandler dieses Signal geliefert haben.

Danach werden der Reihe nach alle Wandler abgefragt und das Ergebnis eingelesen. Es steht dann im RAM-Speicher des Systems zur Verfügung und kann weiter verarbeitet werden. Dieses Weiterverarbeiten kann z.B. bedeuten, daß das Ergebnis in eine Anzeige geschrieben wird und von dem Anwender abgelesen wird.

Nehmen wir nun folgendes Beispiel an: Ein Temperaturfühler mit nachgeschaltetem Verstärker liefert für einen Temperaturbereich 0 bis 100°C eine Spannung von 0 bis 10 V. Dann entsprechen

$$\begin{aligned} 0^{\circ}\text{C} &\cong 0 \text{ V} \\ 100^{\circ}\text{C} &\cong 10 \text{ V} \end{aligned}$$

Wenn wir von unserem 8-Bit-AD-Wandler ausgehen, so wird eine Auflösung des 10 V-Bereiches von

$$\frac{10 \text{ V}}{256 \text{ Stufen}} = 0,039 \frac{\text{V}}{\text{Stufe}}$$

erreicht ( $2^8 = 256$ ). Das Meßergebnis im Speicher muß mit dieser Auflösung bewertet werden und kann dann in ein dezimal richtiges Anzeigewort gewandelt und ausgegeben werden. Damit hat man ein Thermometer. Lautet das 8-Bit-Ergebnis im Speicher z.B. 0 1 1 0 0 1 1 1, so ist als Ergebnis

$$103_{\text{Dez}} \text{ Stufen} \cdot 0,039 \frac{\text{V}}{\text{Stufe}} = 4,0 \text{ V}$$

auszugeben. Auf dieses Ergebnis kommt man, wenn man mit dem Prozessor 103 mal 0,039 addiert, da im Befehlssatz des 8085 keine Multiplikationsbefehle vorhanden sind.

Das nachfolgende Bild zeigt das Flußdiagramm zu dem Programmbeispiel, welches die Initialisierung, Start und Ende der Wandlung und das Abspeichern beschreibt (Bild 26).

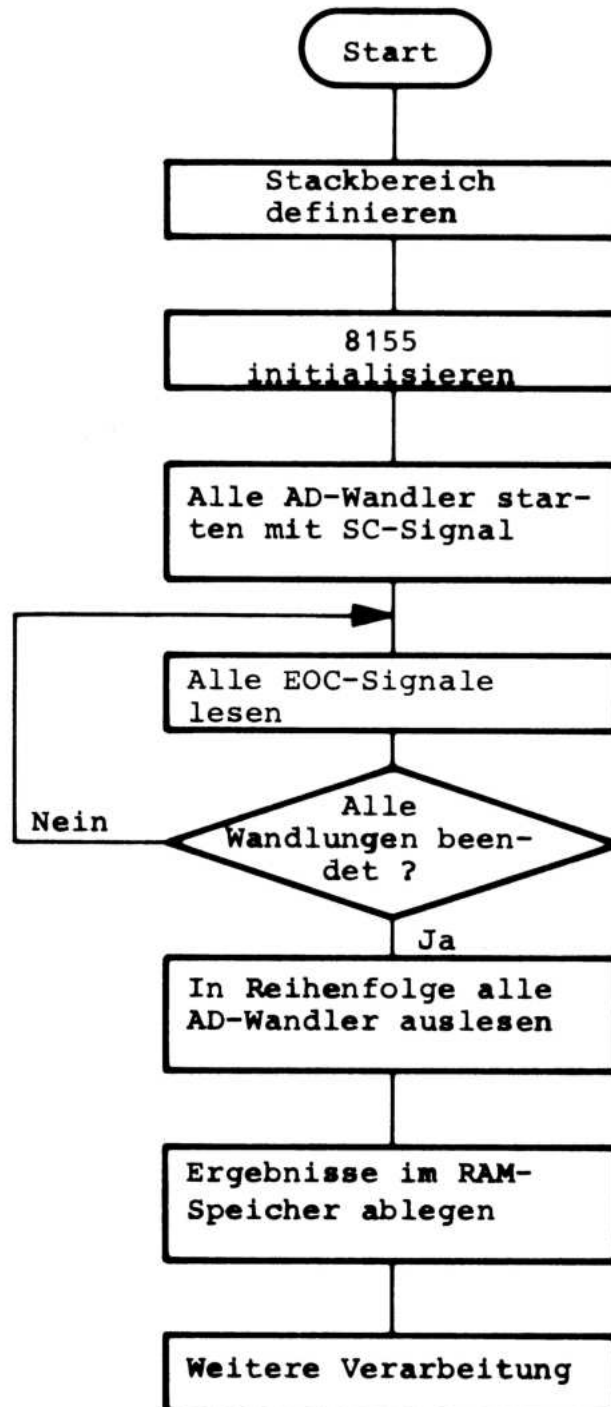


Bild 26



```
; *****  
;  
;  
; DIESES PROGRAMM-BEISPIEL DEMONSTRIERT DAS INITIALI-  
; SIEREN DER HARDWARE FÜR DIE ANALOG/DIGITAL-WAND-  
; LUNG MIT EINEM SYSTEM 8085/8155/ZN427. (NACH DATEN-  
; BUCH DER FIRMA FERRANTI, MUENCHEN). ES WERDEN  
; VIER WANDLER ABGEFRAGT. BEIM EINSATZ WENIGER WAND-  
; LER REDUZIERT SICH DER SOFTWARE-AUFWAND ENTSPRE-  
; CHEND. DIE VON DEN AD-WANDLERN ABGEHOLTEN DATEN  
; WERDEN IM RAM-SPEICHER AB ADRESSE 1900H ABGELEGT.  
;  
; DER KURSTEILNEHMER SOLLTE JETZT IN DER LAGE SEIN,  
; DIE SOFTWARE ZUM BEISPIEL SO ZU VERFEINERN, DASS  
; DIESE DATEN AUCH AUF DIE ANZEIGE GEBRACHT WERDEN  
; BZW. DIE ABFRAGE AUCH IN EINER SCHLEIFE LAEUFT.  
; ES WERDEN ALLE PORTS DES 8155 ENTSPRECHEND DER  
; TABELLE 2 BELEGT. WEITERE HINWEISE SIND DEM TEXT  
; UND DER SCHALTUNG IN DER LEKTION ZU ENTNEHMEN.  
;  
; *****  
; *****
```

ORG 1800H

```
DATA      EQU      1900H  
CSREG     EQU      F8H  
PORTA     EQU      F9H  
PORTB     EQU      FAH  
PORTC     EQU      FBH  
STACK     EQU      18E8H
```

```
; INITIALISIERUNG  
; *****
```

```
        LXI        SP,STACK; STACKBEREICH SETZEN  
        MVI        A,02H    ; EIN/AUS-PORTS DEFINIEREN  
        OUT        CSREG  
        MVI        A,00H  
        OUT        PORTB    ; STARTZEICHEN AN AD-WANDLER  
                                ; SENDEN  
        MVI        A,0FH  
        OUT        PORTB  
        MVI        B,0FH
```



```
;ABFRAGE DER WANDLER
;*****

LOOP:  IN      PORTC
      ANA      B      ;OBERE 4 BITS ABSPALTEN
      CMP      B      ;VERGLEICH, OB ALLE EOC'S
                        ;LOGISCH 1
      JNZ      LOOP
      LXI      H,DATA
MVI    A,1FH
      OUT      PORTB   ;ENABLE AD-WANDLER 1
      IN       PORTA   ;AD-WANDLER 1 LESEN
      MOV      M,A     ;IM SPEICHER SICHERN
      INX      H       ;SPEICHERADRESSE ERHOEHEN
      MVI      A,2FH
      OUT      PORTB   ;ENABLE AD-WANDLER 2
      IN       PORTA   ;AD-WANDLER 2 LESEN
      MOV      M,A     ;IM SPEICHER SICHERN
      INX      H       ;SPEICHERADRESSE ERHOEHEN
      MVI      A,4FH
      OUT      PORTB   ;WIE VORHER FUER WANDLER 3
      IN       PORTA
      MOV      M,A
      INX      H
      MVI      A,8FH
      OUT      PORTB   ;WIE VORHER FUER WANDLER 4
      IN       PORTA
      MOV      M,A

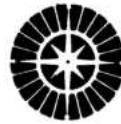
      HLT
```



## K 4. Lösungen zu den Aufgaben in dieser Lektion

- Aufgabe 1 von Seite (4)- 9: Zum Komfort in einem solchen System gehören u.a. ein Betriebssystem, das die Verwendung aller Einheiten, insbesondere auch von Floppy-Laufwerken, ermöglicht, eine Schreibmaschinen ähnliche Tastatur und ein gut lesbares Ausgabemedium (Bildschirm). Weiter muß der Anschluß von Peripheriegeräten über Standardschnittstellen möglich sein.
- Aufgabe 2 von Seite (4)-11:
- a. Flußdiagramm erstellen
  - b. Programm mit Hilfe eines Editors eingeben/ändern.
  - c. Programm auf Floppy sichern
  - d. Programm assemblieren
  - e. Maschinencode mit einem Debug-Programm austesten.
  - f. Wenn alles ok → Ende, sonst ab b. wiederholen.
- Aufgabe 3 von Seite (4)-32: Der Befehl muß für einen 8155, der am "micro-master" angeschlossen ist,
- IN 0F9H
- heißen. (Siehe Beispiel Schalterabfrage und Liste der Adressen des 8155.)
- Aufgabe 4 von Seite (4)-42: Die Sprungadressen müssen vor dem Verschieben umgerechnet werden, wenn dieser neue Zielbereich z.B. EPROM-Speicher ist und das Programm dort fest abgelegt werden soll.
- Aufgabe 5 von Seite (4)-49: Es muß der TRAP-Eingang benutzt werden, da er die höchste Priorität hat.





**A** 5.

<p>Hausaufgaben zur Lektion Nr. 4</p> <p>Name:</p> <p>Anschrift:</p> <p>Teiln.-Nr.:</p>	<p>Korrektur-Rand</p> <p>Nicht beschreiben</p>
<p>1. Beschreiben Sie stichwortartig Unterschiede zwischen einem Computersystem für den Anwender und einem Entwicklungssystem.</p> <p>2. Was versteht man unter einem handshake-Verfahren?</p> <p>3. Wodurch wird dem Anwender das Zusammenschalten von Bausteinen einer Mikroprozessorfamilie erleichtert?</p> <p>4. Welches sind die wesentlichen Aufgaben von Peripheriebausteinen in einer Mikroprozessorfamilie!</p>	



	Korrektur-Rand Nicht beschreiben
<p>5. Welcher Hardware-Unterschied besteht zwischen einer seriellen und einer parallelen Schnittstelle. Welchen Vorteil bietet letztere?</p> <p>6. Beschreiben Sie kurz, was passiert wenn während eines Programmlaufs ein Interrupt RST 5.5, und während dann die Interruptroutine läuft, ein Interrupt RST 6.5 auftritt.</p> <p>7. Erstellen Sie ein Flußdiagramm, welches die Programmierung folgender Aufgabenstellung ermöglicht: Es sollen zwei 8-Bit-Hex-Zahlen miteinander multipliziert werden. Stellen Sie die Eingabe, das Rechenverfahren und die Ausgabe dar.</p>	



	Korrektur-Rand Nicht beschreiben
<p>8. Welche Betriebsarten gibt es für die Ein-/Ausgabekanäle des 8155? Beschreiben Sie diese.</p> <p>9. Wieviel Adressleitungen sind erforderlich, um ein 4K-EPROM anzusteuern?</p> <p>10. Wie hoch ist die <u>Taktfrequenz</u> der CPU im "micromaster"?</p> <p>11. In dem Versuchsaufbau Seite (4)-56 haben Sie einen 12-Bit-AD-Wandler zur Verfügung. Welche Auflösung ist nun erreichbar?</p> <p>Zensierung:</p> <p>Lehrgangsleiter:</p>	



6. Anhang

6.1 Bild 27 - Erweiterung mit Port-Baustein 8155

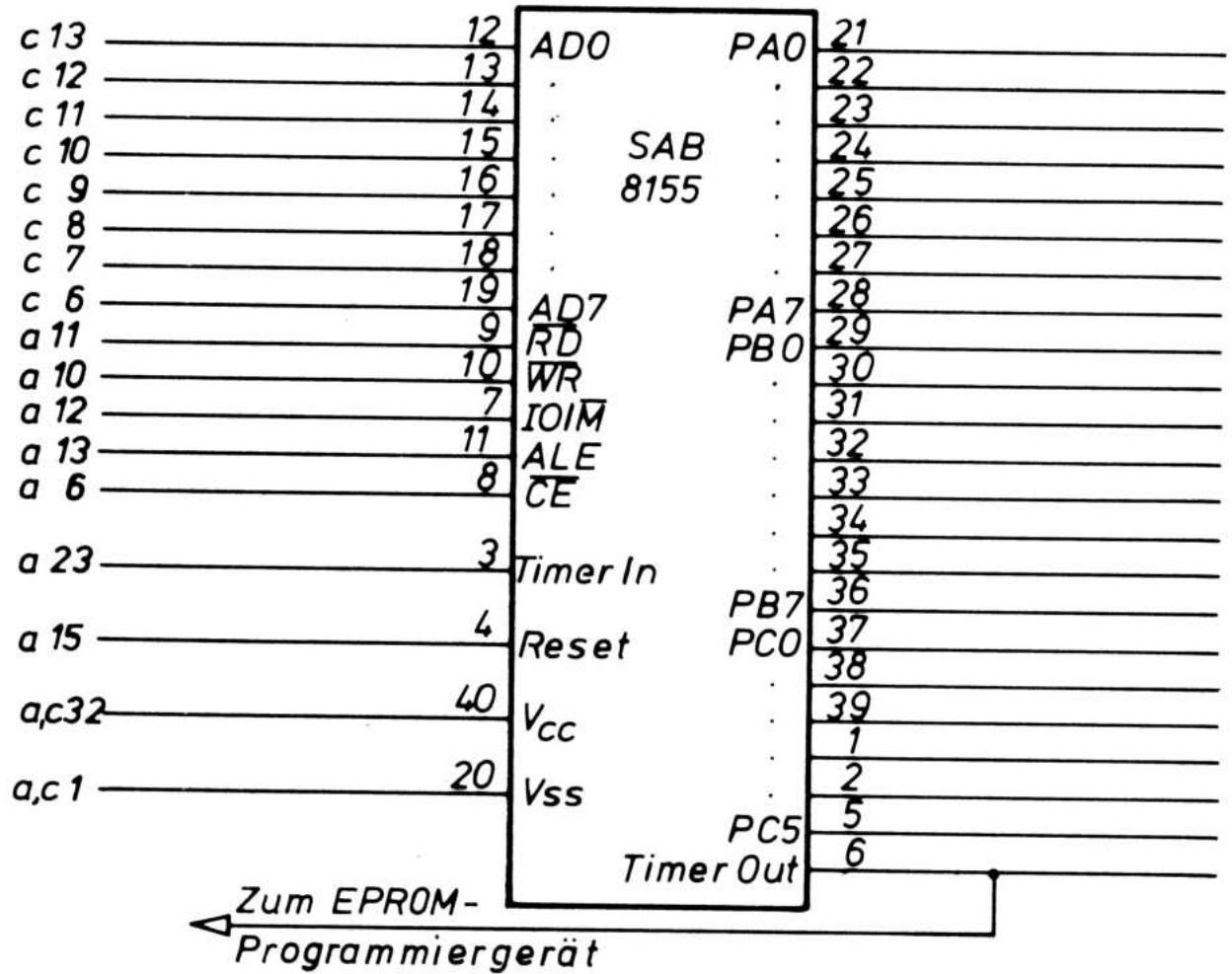
6.2 Bild 28 - Erweiterung mit 1K-RAM

6.3 Bild 29 - Erweiterung mit 2K-EPROM und EPROM-Programmierer

6.4 Bild 30/31 - Analog/Digitalwandlung mit Baustein ZN 427  
von Ferranti

6.5 Belegung der Anwenderschnittstelle

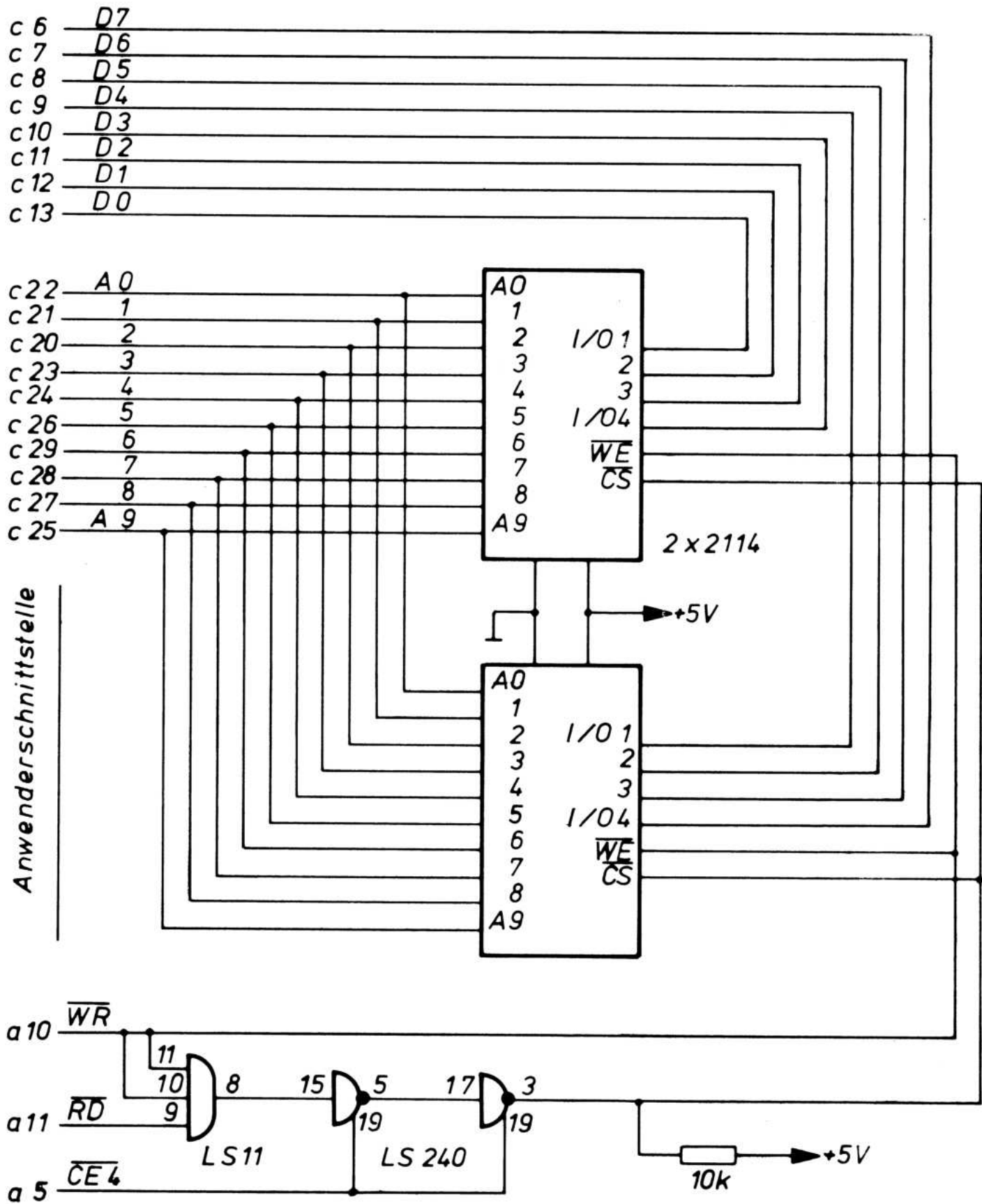
6.6 Belegung des Speicherraumes mit den zugehörigen  $\overline{CS}$ -Signalen

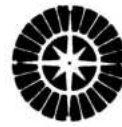
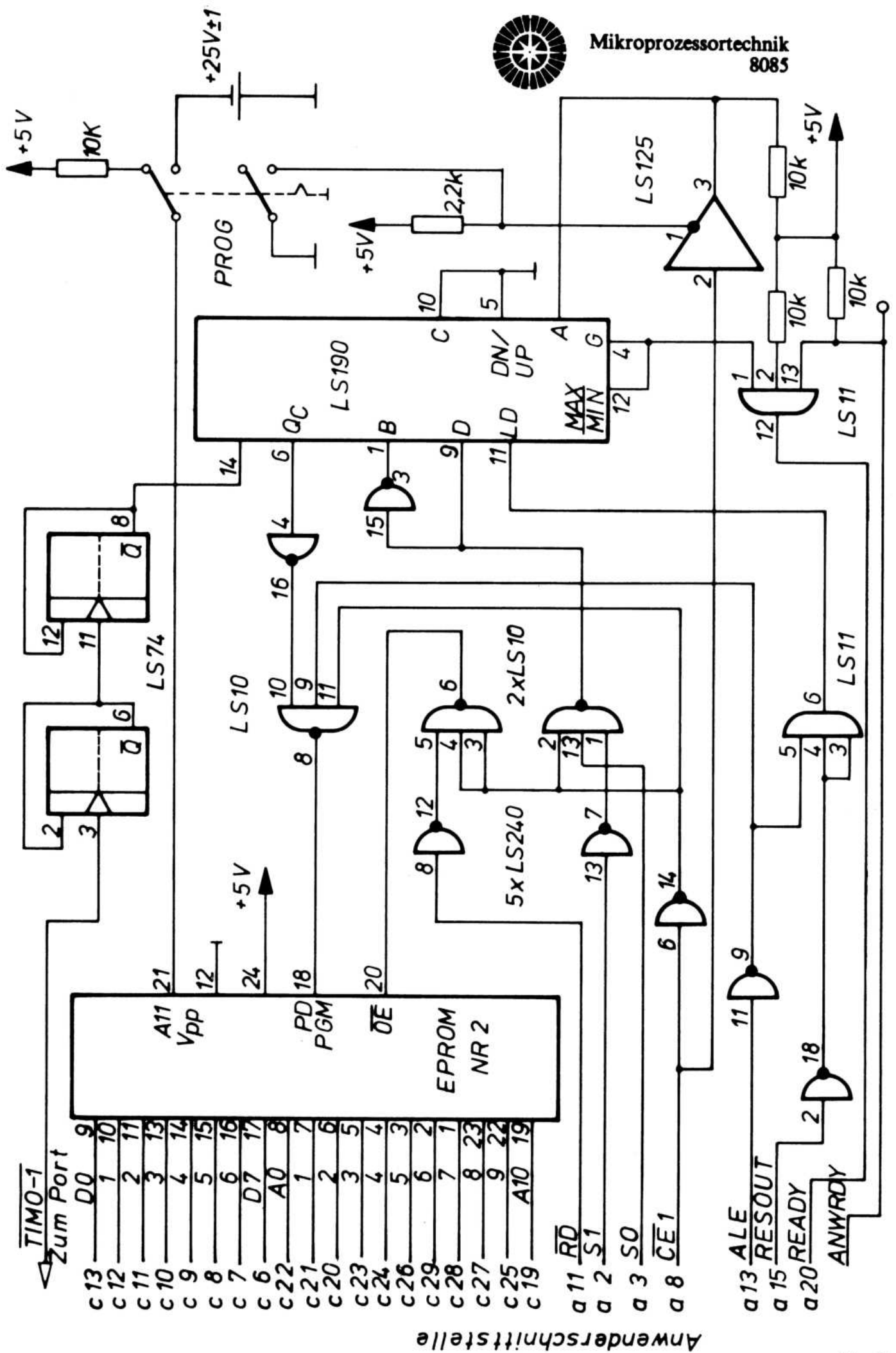


6.1

Bild 27

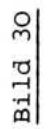
Erweiterung mit Port-Baustein 8155





Mikroprozessortechnik  
8085

Bild 29 - Erweiterung mit 2K-EPROM und EPROM-Programmierer









## 6.5 Belegung der Anwenderschnittstelle

Anschluß Pin-Nr.	Reihe a	Reihe c
1	OV	OV
2	CE7	S1
3	CE6	SO
4	CE5	SOD
5	CE4	SID
6	CE3	D7
7	CE2	D6
8	CE1	D5
9	CEO	D4
10	$\overline{WR}$	D3
11	$\overline{RD}$	D2
12	IO/ $\overline{M}$	D1
13	ALE	DO
14	$\overline{RESIN}$	A15
15	RESOUT	A14
16		A13
17		A12
18		A11
19		A10
20	READY	A2
21	INTR	A1
22	$\overline{INTA}$	A0
23	CLK	A3
24	HOLD	A4
25	HLDA	A9
26		A5
27		A8
28	RST 6.5	A7
29	TRAP	A6
30	RST 5.5	
31		
32	+5V	+5V



## 6.6 Belegung des Speicherraumes mit den zugehörigen $\overline{CS}$ -Signalen

