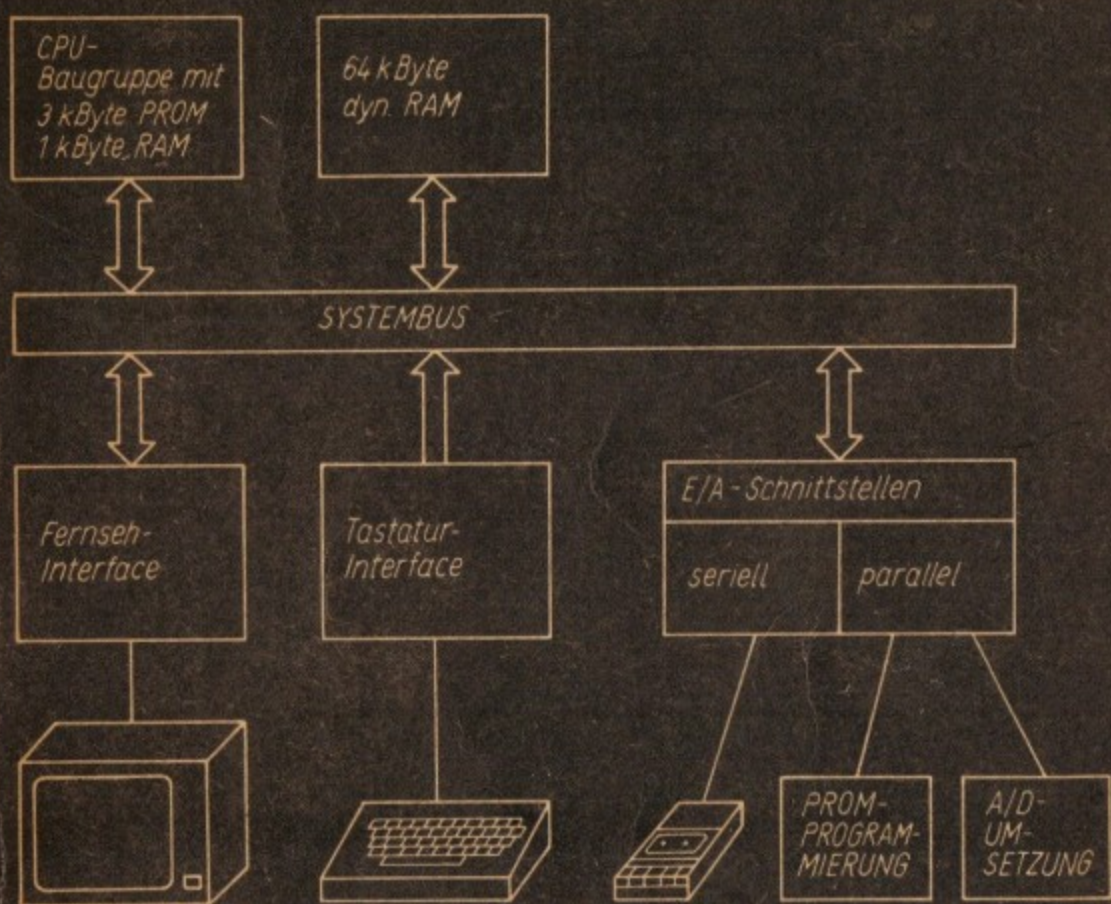


amateurreihe

electronica



Bernd Hübler
Klaus-Peter Evert

Ausbaufähiger
Mikrocomputer
mit dem U 880

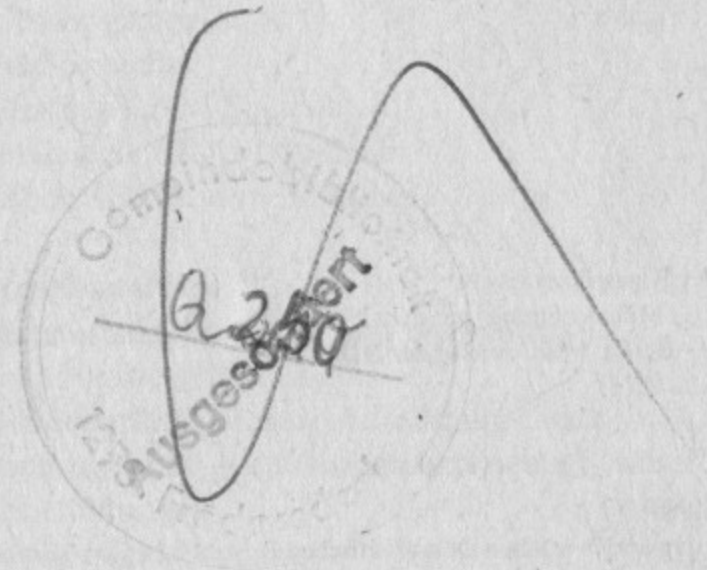
227/

228

electronica · Band 227/228

BERND HÜBLER · KLAUS-PETER EVERT †

Ausbaufähiger Mikrocomputer mit dem U 880



MILITÄRVERLAG
DER DEUTSCHEN DEMOKRATISCHEN
REPUBLIK

BERND HÜBLER · KLAUS-PETER EVERT †

Inhaltsverzeichnis

1.	Vorwort	5
2.	Systembeschreibung	7
3.	CPU-Baugruppe	10
3.1.	Taktversorgung, Buspufferung	10
3.2.	Speicher	14
3.3.	Zentrale Funktionseinheiten	15
3.3.1.	Speicherselektierung	15
3.3.2.	I/O-Portselektierung	17
3.3.3.	Startlogik	18
3.4.	Aufbau und Inbetriebnahme	22
4.	Speichererweiterung	25
4.1.	ROM-Speicher	26
4.2.	RAM-Speicher	30
4.2.1.	Statischer RAM-Speicher	30
4.2.2.	Dynamischer RAM-Speicher	32
4.3.	Aufbau und Inbetriebnahme	38
5.	Fernsehinterface	42
5.1.	Zeichendarstellung auf dem Bildschirm	42
5.2.	Schaltungsbeschreibung	46
5.2.1.	Bildwiederholungspeicher, Adreßmultiplexer	48
5.2.2.	Synchron- und Austastsignalerzeugung, Adressen- bereitstellung	52
5.2.3.	Videosignalerzeugung	55
5.2.4.	BAS-Signalerzeugung	61
5.3.	Erweiterungsmöglichkeiten	61
5.3.1.	Zeichengenerator mit Graphiksymbolen	62
5.3.2.	RAM-Speicher als Zeichengenerator	62
5.3.3.	Vollgraphik	62
5.4.	Aufbau und Inbetriebnahme	63
6.	Alphanumerische Tastatur	65

Bernd Hübler/Klaus-Peter Evert:
Ausbaufähiger Mikrocomputer mit dem U 880.
- 1. Auflage - Berlin: Militärverlag der DDR, 1985 -
(electronica 227/228)

1. Auflage, 1985
© Militärverlag der Deutschen Demokratischen Republik (VEB) - Berlin, 1985
Lizenz-Nr. 5
Printed in the German Democratic Republic
Gesamtherstellung: Druckerei Märkische Volksstimme Potsdam
Lektor: Rainer Erlekampf
Zeichnungen: Angelika Ulsamer
Typografie: Martina Schwarz
Redaktionsschluß: 12. November 1984
LSV 3539
Bestellnummer: 746728 1
00380

6.1.	Schaltungsbeschreibung	65
6.2.	Aufbau und Inbetriebnahme	70
7.	Ein-/und Ausgabebaugruppen	74
7.1.	Universelle I/O-Karte	74
7.2.	EPROM-Programmiergerät	79
7.2.1.	Programmiervorschriften	79
7.2.2.	Die Schaltung	81
7.2.3.	Aufbau und Codierung	83
7.2.4.	Ein Beispielprogramm	85
7.3.	Analschnittstellen	87
7.3.1.	Digital/Analog-Wandler	87
7.3.2.	Analog/Digital-Wandler	91
8.	Kassetteninterface	95
8.1.	Aufzeichnungsverfahren	95
8.2.	Realisierung	97
9.	Monitorprogramm	104
9.1.	Beschreibung der Monitorkommandos	104
9.2.	Schnittstellen und Erweiterbarkeit	110
9.3.	Das Programm	113
10.	Hardwareaufbau und Inbetriebnahme	177
10.1.	Stromversorgung	177
10.2.	Systemaufbau	179
11.	Anhang	184
12.	Literaturverzeichnis	188

1. Vorwort

Die Möglichkeiten und Erfordernisse der modernen Mikroelektronik zeigen sich im Bereich der Mikrocomputertechnik besonders deutlich.

Der Anwendungsbereich des Mikrocomputers reicht von der Steuerung und Regelung industrieller Prozesse – genannt sei hier nur das Schlagwort Industrieroboter – über den wissenschaftlich-technischen und den Bürocomputer bis hin zum Heimcomputer. Immer stärker dringt der Mikrocomputer auf Grund seiner ständig wachsenden Leistungsfähigkeit in die Bereiche der traditionellen EDV ein.

Für ein modernes Industrieland ist es unumgänglich, die Möglichkeiten dieser neuen Technik zu nutzen.

Wissen über Computertechnik im allgemeinen und Mikrocomputertechnik im speziellen sowie computerspezifische Denkweisen dürfen deshalb nicht einem kleinen Kreis von Spezialisten vorbehalten bleiben. Diese Erkenntnis hat international zu einer bemerkenswerten Verbreitung und Popularisierung des Heimcomputers bzw. des Personalcomputers besonders bei jugendlichen Interessenten geführt.

Das vorliegende Heft wendet sich deshalb an den elektronisch versierten Amateur, aber auch an den Elektroniker, der noch keine Gelegenheit hatte, auf dem Gebiet der Mikrocomputertechnik praktische Erfahrungen zu sammeln. Grundkenntnisse der Digitaltechnik sowie der Mikroprozessortechnik, wie sie beispielsweise [1] vermittelt, werden vorausgesetzt.

Das gegenwärtig in der DDR verfügbare Schaltkreisangebot gestattet dem potentiellen Interessenten den Aufbau eines leistungsfähigen Mikrocomputers. Nachbaufähige Schaltungen für die Realisierung der dazu benötigten Hardware sowie erprobte Betriebssoftware sollen den Weg zum fertigen Gerät erleichtern.

Der Aufbau des Systems ist mit einiger Mühe verbunden und erfordert ein gewisses Durchstehvermögen.

Beim eigenen Realisieren begreift man jedoch die relativ komplizierte Hardware am besten. Gerade für den Bereich des Mikro-

computers sind gründliche Hardwarekenntnisse von besonderem Wert.

Die Autoren hoffen, daß einem möglichst breiten interessierten Personenkreis der Einstieg in die zukunftssträchtige Problematik der Mikrocomputertechnik ermöglicht wird.

Berlin, im Januar 1984

Bernd Hübler
Klaus-Peter Evert †

2. Systembeschreibung

Das in der Beschreibung vorgestellte Mikrocomputerkonzept basiert auf der Schaltkreisfamilie des Mikroprozessors *U 880*. Diese leistungsfähige 8-Bit-CPU einschließlich der dazugehörigen Peripherieschaltkreise sowie der bereits verfügbaren Speicher erlaubt bei vertretbarem Aufwand den Aufbau eines universell einsetzbaren Mikrocomputers.

Bild 2.1 zeigt den Übersichtsschaltplan der Hardware des Systems. Danach besteht das voll ausgebaute System aus folgenden Komponenten:

- CPU-Baugruppe (realisiert als Einplatinenrechner) mit 3-kByte-PROM, 1-kByte-RAM
- 64-kByte-RAM-Speicher
- Bildschirm mit 24 Zeichenzeilen zu je 64 Zeichenpositionen
- alphanumerische Tastatur (maximal 64 Tasten)
- Kasette als externes Speichermedium
- parallele und serielle Ein-/Ausgabeschnittstelle
- PROM-Programmiergerät

Als Bildschirm eignet sich ein handelsüblicher Schwarzweißfernsehempfänger (möglichst Koffergerät), und zur Bandaufzeichnung nutzt man einen Kassettenrecorder.

Die Hardware kann schrittweise, beginnend mit Aufbau und Inbetriebnahme des Einplatinenrechners bis hin zum kompletten System, ausgebaut werden.

Das Systemkonzept des beschriebenen Mikrocomputers soll einen möglichst universellen Anwendungsbereich sichern. Dementsprechend kann man ein breites Softwarespektrum einsetzen. Es reicht vom Maschinensprachniveau bis zu Assemblerprogrammen und zu höheren Programmiersprachen (z. B. BASIC). Aus diesem Grund steht der RAM-Speicher ab Adresse 0 zur Verfügung. Ein möglichst weitgehender Ausbau des RAM-Bereichs erweist sich unter den genannten Aspekten als günstig.

Der Computer soll aber auch als Entwicklungssystem genutzt werden. Für diesen Anwendungsfall wird ein Gerät vorgestellt, das die Programmierung aller wichtigen, derzeit bekannten PROM-Typen (≥ 1 kByte) gestattet.

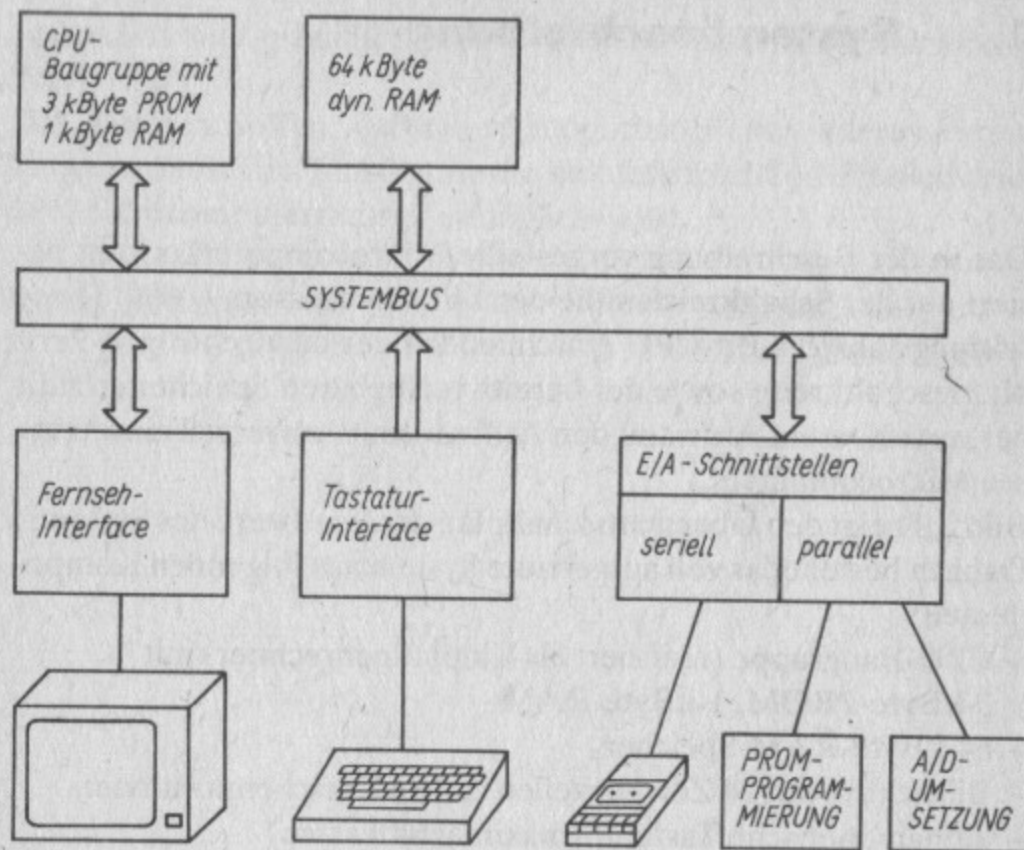


Bild 2.1 Hardwareübersicht

Voraussetzung für die Arbeit mit dem Computer ist das Vorhandensein eines (im allgemeinen) systemresidenten Betriebssystems. Als Minimalforderung muß ein Urlader auf dem System abgespeichert sein, der das Laden eines Betriebssystems von einem externen Speichermedium ermöglicht. Für den vorgestellten Mikrocomputer wurde ein den Belangen des Systems angepaßtes Monitorprogramm entwickelt. Mit Hilfe des Monitors wird der Datenverkehr zwischen CPU und Peripherie abgewickelt und das Arbeiten auf Maschinensprachniveau ermöglicht. Es lassen sich also Programme sowie Datenein- und -ausgaben testen. Darüber hinaus verfügt das Monitorprogramm über Schnittstellen, die es gestatten, Teile desselben in Anwenderprogrammen mitzunutzen. Beispielsweise können die auch beim Betrieb mit höheren Programmiersprachen benötigten Treiberrountinen (Bildschirm, Tastatur, Kassettenaufzeichnung) so in einfacher Weise angesprochen werden.

Unter Berücksichtigung des Systemkonzepts sollte der Monitor das einzige systemresidente, d. h. auf PROM abgespeicherte, Pro-

gramm sein. Alle weitere Software kann vom externen Speichermedium (Kassette) bei Bedarf in den Arbeitsspeicher des Mikrocomputers geladen werden. Abhängig von den Möglichkeiten (z. B. hinsichtlich des RAM-Speicherausbaus) lassen sich jedoch auch individuelle Lösungen realisieren. Beispielsweise kann ein geeigneter BASIC-Interpreter auch auf PROM abgespeichert werden. Die schaltungsmäßigen Lösungen für den in diesem Fall zusätzlichen Bedarf an PROM-Speicherkapazität sind im vorgestellten Hardwareumfang enthalten.

Die frei verfügbaren E/A-Schnittstellen gestatten das Koppeln des Mikrocomputers mit externen Geräten. Für die in diesem Zusammenhang häufig benötigten A/D-Wandler bzw. D/A-Wandler werden Lösungsvorschläge unterbreitet.

Im voll ausgebauten Zustand steht damit ein leistungsfähiger Mikrocomputer zur Verfügung, der darüber hinaus hard- und softwaremäßig für individuelle Erweiterungen vorbereitet ist.

3. CPU-Baugruppe

Die CPU-Baugruppe stellt, wie bereits der Name sagt (*Central-Processing-Unit* – zentrale Verarbeitungseinheit), das Herzstück des Mikrocomputersystems dar. Basierend auf dem Mikroprozessor *U 880*, bestimmt sie in wesentlichem Umfang die Leistungsfähigkeit des Systems.

Als Voraussetzung zur Erfüllung ihrer Hauptfunktion hat die CPU-Baugruppe die Aufgabe der Taktversorgung des Mikroprozessors sowie der Verteilung der Systeminformationen (Daten, Adressen, Steuersignale) im Computer. Ihre spezielle Konfiguration ist vorrangig vom gewünschten Einsatzzweck (z. B. Heimcomputer, komfortabler Bürocomputer, Steuercomputer usw.) abhängig und damit auch vom Umfang und Niveau der Software, die man beabsichtigt, auf dem Computer lauffähig zu machen (fachmännisch ausgedrückt: zu implementieren). Letztlich ist die CPU-Platine auch eine Widerspiegelung der Systemphilosophie des Entwicklers.

Bild 3.1 zeigt die für diese Broschüre gewählte Variante.

3.1 Taktversorgung, Buspufferung

Als Taktgenerator wird ein quarzstabilisierter TTL-Generator verwendet. Er schwingt mit 10 MHz und liefert der CPU, nachdem diese Frequenz 1 : 4 unternimmt und durch 2 Leistungsgatter *D 240* gepuffert wurde, den Standardtakt von 2,5 MHz. Außerdem wird der Takt am Systembus bereitgestellt.

Wer über einen geeigneten Schwingquarz verfügt, hat die Möglichkeit, mit der angegebenen Schaltung oder entsprechend anders eine Taktfrequenz von genau 2,4576 MHz zu erzeugen. Damit lassen sich standardisierte Datenübertragungsraten (Baudraten) für die entsprechende Peripherie (Kassettenaufzeichnung, Datenfernübertragung usw.) realisieren.

Die an den $\overline{\text{RESET}}$ -Eingang des Mikroprozessors angeschlossene Kombination aus RC-Glied und den Gattern *D3.1* und *D3.2* bewirkt das Einschalt-RESET (Power-On-RESET) und damit den

Urstart des Systems. Diese $\overline{\text{RESET}}$ -Schaltung wird häufig auch mit einfachen NAND-Gattern bzw. ohne Gatter ausgeführt. Die Autoren empfehlen die Verwendung des Triggerschaltkreises *MH 7413*. Auch das $\overline{\text{RESET}}$ -Signal steht am Systembus zur Verfügung. Zum Realisieren einer Reset-Taste kann man eine Schaltungslösung verwenden, die einen Datenverlust bei eventuell eingesetzten dynamischen RAM (siehe Abschnitt 4.2.2.) verhindert. Eine mögliche Variante ist [2] zu entnehmen. Die restlichen CPU-Eingänge $\overline{\text{INT}}$, $\overline{\text{NMI}}$, $\overline{\text{BUSRQ}}$, $\overline{\text{WAIT}}$ – ihre Bedeutung wird als bekannt vorausgesetzt und ist z. B. in [1], [2] nachzulesen – werden, da sie L-aktiv sind, für den Fall des nicht angesteuerten Zustandes mit den entsprechenden Widerständen auf ein sicheres H-Potential gebracht.

Die Daten- und Adreßausgänge der CPU *U 880* sind als Tri-State-Ausgänge ausgeführt. Sie sind ebenso wie die L-aktiven Steuerausgänge – von denen nur die Signale $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$ über Tri-State-Ausgänge verfügen – in der Lage, eine TTL-Lasteinheit zu treiben. Für das Vergrößern der Busbelastbarkeit aus der Sicht der CPU sowie zur Leistungstreibung, d. h. zum Unterdrücken des Einflusses von Störimpulsen und parasitären Kapazitäten, muß man die genannten Signale vor dem Bereitstellen auf den geeigneten Treiberschaltkreis puffern.

Der Datenverkehr zwischen CPU und Peripherie ist grundsätzlich bidirektional, d. h., er erfolgt in beiden Richtungen.

Dementsprechend werden die Datenausgänge *D0* ... *D7* mit 2 bidirektionalen Treiberschaltkreisen des Typs *8216* (*D11*, *D12*) gepuffert. Dazu ist es notwendig, gleichwertige Datenein- und -ausgänge (*DI* und *DO*) des *8216* parallelzuschalten. Für das Durchschalten in der jeweils gewünschten Richtung, d. h. für das Bereitstellen des Datenrichtungssignals $\overline{\text{DIEN}}$ am Schaltkreis *8216*, ist die Datenrichtungslogik, bestehend aus dem Gatter *D4.3*, zuständig. Wenn das Lesesignal $\overline{\text{RD}}$ aktiv ist, ist also der Datenfluß in der Richtung Peripherie \rightarrow CPU möglich. Für den Fall $\overline{\text{RD}} = \text{H}$ sind die Verhältnisse umgekehrt.

Das $\overline{\text{MI}}$ -Signal muß bei der Datenrichtungsumschaltung berücksichtigt werden, damit der Interruptmode 2 (Vektorinterrupt) abgesichert ist. Dabei muß die CPU einen Interruptvektor von der Peripherie lesen. In diesem Fall ist $\overline{\text{RD}}$ nicht aktiv, sondern $\overline{\text{MI}}$ und $\overline{\text{IORQ}}$.

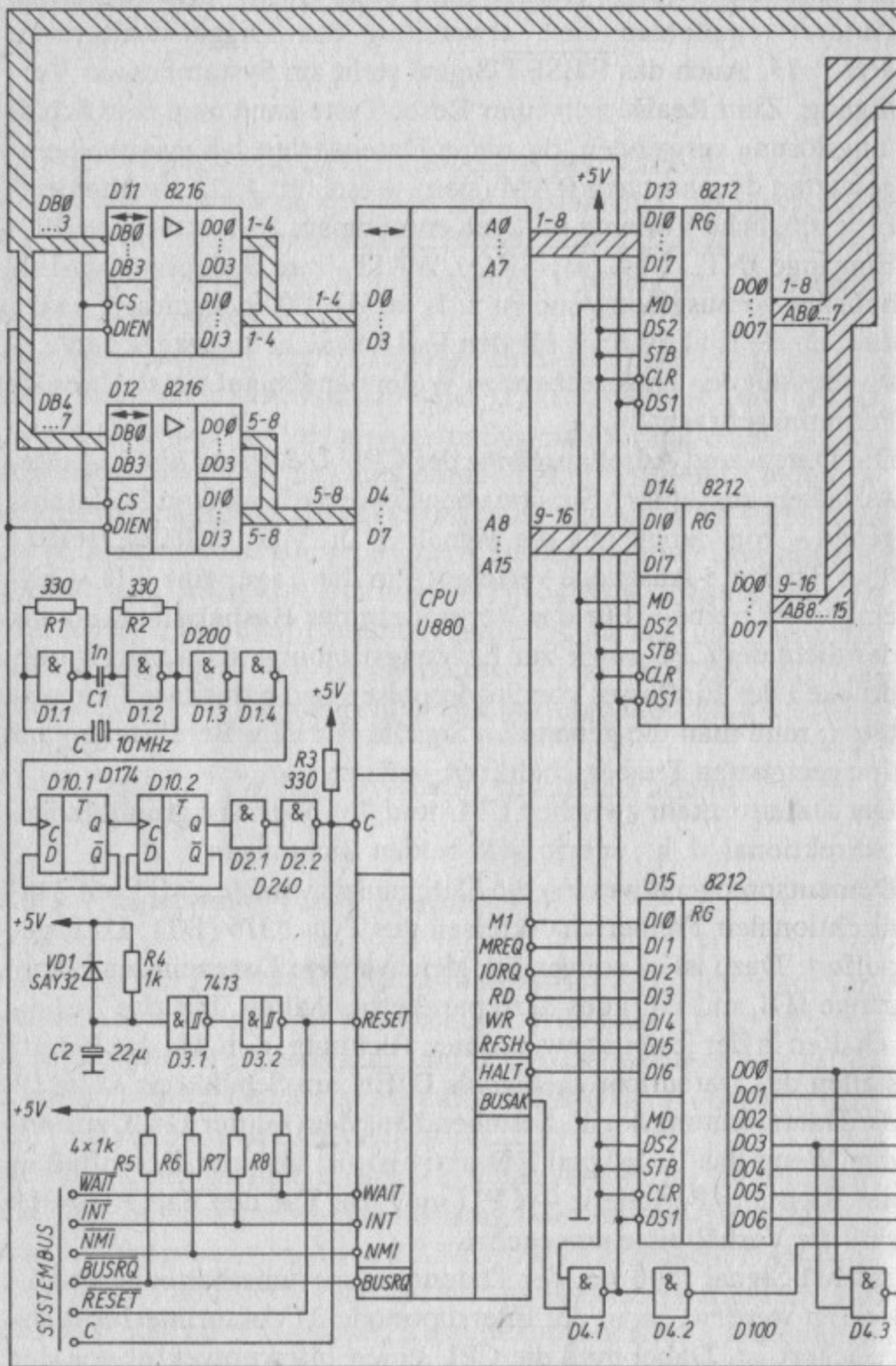
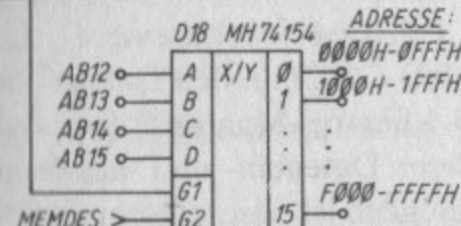
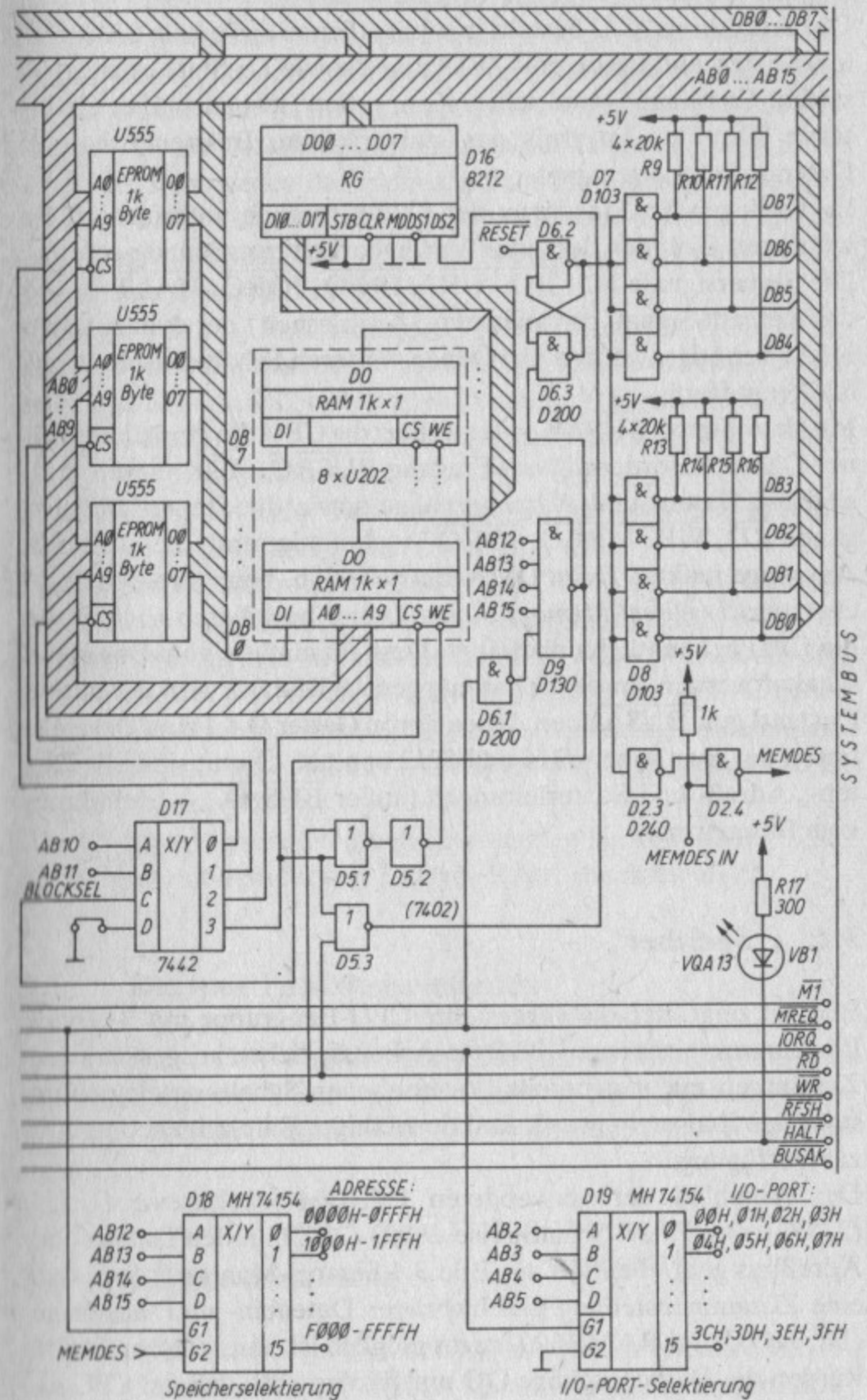
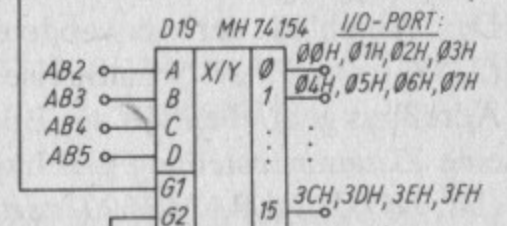


Bild 3.1 CPU-Baugruppe



Speicherselektierung



I/O-PORT-Selektierung

Als Treiber für die Adreßausgänge A0 ... A15 der CPU setzt man Schaltkreise des Typs 8212 (D13, D14) ein.

Der Schaltkreis 8212 besteht im wesentlichen aus einem 8-Bit-Datenregister und -treiber mit Zusatzlogik sowie Tri-State-Ausgangsstufen. Er ist auf Grund seiner Konzeption (löst auch über ein internes Flip-Flop Interrupt aus) universell zur Bussteuerung und Datenpufferung geeignet.

Im vorliegenden Einsatz ist der 8212 eigentlich unterfordert. Er wurde aus Gründen der guten Verfügbarkeit trotzdem eingesetzt. Die Steuersignale $\overline{M1}$, \overline{RD} , \overline{WR} , \overline{MREQ} , \overline{IORQ} , \overline{HALT} sowie das Refresh-Signal, das man zum „Auffrischen“ der dynamischen RAM benötigt, werden über einen weiteren Schaltkreis des Typs 8212 gepuffert.

Mit dem Signal $\overline{BUSAK} = L$ quittiert die CPU die Anmeldung einer DMA-Anforderung am Eingang \overline{BUSRQ} . Für diesen Fall sind die Daten- und Adreßausgänge sowie die Steuersignalausgänge \overline{RD} , \overline{WR} , \overline{MREQ} , \overline{IORQ} hochohmig und die weiteren Ausgänge inaktiv. Beim DMA-Betrieb, d. h. beim direkten Speicherzugriff (*Direct Memory Access*) einer peripheren Einheit, ist die CPU bekanntlich unbeteiligt. Deshalb muß sie vom Bus abgeschaltet werden. In der Schaltung gemäß Bild 3.1 wird der aktive Zustand von \overline{BUSAK} am Ausgang von Gatter D 4.1 zum Deselektieren der Bustreiber 8216 und 8212 benutzt. Damit sind alle Daten-, Adreß- und Steuerleitungen (außer \overline{BUSAK}) hochohmig vom Bus getrennt.

3.2. Speicher

Bild 3.1 zeigt, daß die vorgestellte CPU-Baugruppe mit 3-kByte-Programmspeicher und 1-kByte-Arbeitsspeicher ausgerüstet ist. Zusammen mit den bereits beschriebenen Schaltungselementen steht damit also ein in sich funktionsfähiger Einplatinencomputer zur Verfügung.

Der Anschluß der verwendeten EPROM-Schaltkreise U 555 (2708) sowie RAM-Schaltkreise U 202 (2102) an den Daten- und Adreßbus geht ebenfalls aus Bild 3.1 hervor. Man stellt fest, daß eine Zusammenstellung gleichwertiger Datenein- und -ausgänge (DI, DO) beim RAM U 202 nicht möglich ist. Aus diesem Grund werden die Datenausgänge DO nur für den Fall, daß die CPU ei-

nen Speicherlesezyklus zum selektierten RAM-Block durchführt, über die Tri-State-Ausgänge des Schaltkreises 8212 (D15) auf den Datenbus geschaltet. Andernfalls sind sie hochohmig vom Bus getrennt. Günstig, besonders wenn man größere Speicherblöcke auf der Basis statischer RAM des Typs U 202 verwendet, ist das Puffern sowohl der Dateneingänge als auch der Datenausgänge. Hierfür bietet sich wieder der bidirektionale Treiberschaltkreis 8216 an. Die Verdrahtung der benötigten Schaltkreise 8216 mit den Ein- bzw. Ausgängen des U 202 ergibt sich aus den jeweiligen Pin-Belegungen (siehe auch Bild 4.3).

Die Lese/Schreib-Umschaltung am \overline{WE} -Eingang des U 202 sowie die Bereitstellung des für die gewünschte Funktion des 8212 notwendigen Signals DS2 realisiert man mit geringem Aufwand über die NOR-Gatter D 5.1, D 5.2, D 5.3. Dazu werden die CPU-Signale \overline{RD} , \overline{WR} verwendet.

Beim Einsatz des 8216 steht als Datenrichtungssignal \overline{DIEN} das \overline{RD} -Signal zur Verfügung (siehe auch Bild 4.3).

Für die Chipselektierung der 3 EPROM U 555 sowie des 1-kByte-RAM-Blocks ist eine 1-aus-4-Decodierung notwendig. Sehr einfach und übersichtlich läßt sich dies mit dem 1-aus-10-Decoder MH 7442 lösen, der das jeweilige Chipselektsignal aus den Adressen AB10, AB11 sowie einem 4-kByte-Blockselektsignal decodiert. Aber auch jeder andere 1-aus-n-Decoder ($n \geq 4$) ist geeignet (z. B. 1-aus-8-Decoder 8205).

Das CS-Signal für die eventuell eingesetzten bidirektionalen Treiber 8216 ist mit dem für den 1-kByte-RAM-Block identisch.

3.3. Zentrale Funktionseinheiten

Die 3 auf der CPU-Platine untergebrachten Funktionseinheiten haben in engerem Sinne mit der Funktion der CPU-Baugruppe nichts zu tun, sie sind aber für das Systemkonzept von Bedeutung.

3.3.1. Speicherselektierung

Die Speicher werden bei üblichen Mikrocomputersystemen (z. B. dem K 1520 vom VEB Kombinat Robotron) häufig dezentral se-

lektiert, d. h., mit entsprechenden Decodierschaltungen erzeugt sich jede Speicherplatine ihren Adreßbereich im Adreßraum des Computers (64 kByte beim *U 880*) gewissermaßen selbst. Dieser Adreßbereich ist meistens vom Anwender über Drahtbrücken oder DIL-Schalter auswählbar.

Im Gegensatz dazu wurde für das vorgestellte System eine Schaltungsvariante zur zentralen Speicherselektierung entwickelt. Als elegante Lösung bietet sich der 1-aus-16-Decoderschaltkreis *MH 74154* an, der es erlaubt, aus den Adressen AB12 ... AB15 16 4-kByte-Blockselektionsignale zu decodieren. Damit ist man in der Lage, den 64-kByte-Adreßbereich der CPU *U 880* in 16 4-kByte-Speicherblöcke aufzuteilen. Tabelle 3.1. verdeutlicht dies.

Für die praktische Auswahl eines gewünschten Speicherbereichs ist es lediglich notwendig, den jeweiligen Ausgang des *MH 74154* mit dem Blockselektionsingang der entsprechenden Speicherplatine galvanisch zu verbinden.

Die 4-kByte-Blöcke werden dann dezentral mit Hilfe der angegebenen Chipselektionschaltungen in die gewünschte Größe unterteilt. In der Schaltung gemäß Bild 3.1 wird der Eingang G1 des *MH 74154* als notwendige logische Verknüpfung der Speicherselektierung mit dem $\overline{\text{MREQ}}$ -Signal der CPU verwendet. Den Eingang G2 nutzt man zum Deselektieren des gesamten Speicherbereichs (G2 = H). Der Grund für diese Maßnahme wird im Zusammenhang mit der Startlogik erläutert.

Tabelle 3.1. Zentrale Speicherselektierung

Ausgang <i>MH 74154</i>	decodierter Speicherbereich (HEXA)	selektierter Speicherblock (kByte)
0	0000 - 0FFF	0 - 4
1	1000 - 1FFF	4 - 8
2	2000 - 2FFF	8 - 12
.	.	.
.	.	.
15	F000 - FFFF	60 - 64

3.3.2. I/O-Portselektierung

Der Datenverkehr der CPU mit peripheren Geräten wird, sofern diese nicht wie gewöhnliche Speicherbereiche ansprechbar sind, über I/O-Ports (Ein-/Ausgabe-Tore) abgewickelt. Die Adressierung der I/O-Ports ergibt sich aus dem speziellen Verhalten der CPU *U 880*. Diese liefert während eines I/O-Zyklus ($\overline{\text{IORQ}} = \text{L}$) die 8 niederwertigen Adreßbit A0 ... A7 als I/O-Portadresse. Ob es sich um einen Eingabe- (Portlesen) oder Ausgabezyklus (Portschreiben) handelt, richtet sich nach der gleichzeitigen Aktivität des entsprechenden CPU-Signals $\overline{\text{RD}}$ oder $\overline{\text{WR}}$ (siehe [1], [2]).

Mit den 8 zur Verfügung stehenden Adreßbit A0 ... A7 können also maximal 256 I/O-Ports durch die CPU angesprochen werden. Die I/O-Ports werden im vorliegenden System folgendermaßen selektiert:

Als Decoderschaltkreis verwendet man wieder den von der Speicherselektierung bekannten 1-aus-16-Decoder *MH 74154*. Die Adreßbit AB0, AB1 werden im allgemeinen für spezielle Selektierungsfunktionen im Zusammenhang mit den I/O-Schaltkreisen *PIO U 855* bzw. *SIO U 856* benötigt (siehe Abschnitte 7. und 8.) und sind damit bereits belegt. Aus den in der Wertigkeit nächstfolgenden Adressen AB2, AB3, AB4, AB5 wird das I/O-Portselektionsignal mit dem *MH 74154*, wie in Tabelle 3.2. angegeben, decodiert. Analog zur Speicherselektierung wird dabei die notwendige logische Verknüpfung mit dem Signal $\overline{\text{IORQ}}$ über den Toreingang G1 erreicht.

Damit verfügt man über 16 ($\times 4$) I/O-Portadressen. Die Unterscheidung der an jedem Ausgang vorliegenden 4 Einzeladressen ist

Tabelle 3.2. I/O-Portselektierung

Ausgang <i>MH 74154</i>	decodierte I/O-Ports (HEXA)
0	00, 01, 02, 03
1	04, 05, 06, 07
2	08, 09, 0A, 0B
.	.
.	.
15	3C, 3D, 3E, 3F

mit zusätzlichem Decodieraufwand (jeweils 1 aus 4) möglich. Für das System sind 16 I/O-Ports ausreichend, zumal bei Verwendung der systemspezifischen I/O-Schaltkreise PIO U 855 bzw. SIO U 856 über jedes der 16 Ports weitere schaltkreisinterne I/O-Kanäle angesprochen werden können (Port B/ \bar{A} Select über CPU-Adresse A0).

Ein gewünschtes I/O-Port wird der jeweiligen peripheren Einheit wieder durch Verbinden des entsprechenden Ausgangs des MH 74154 mit dem Chipselekteingang der peripheriespezifischen Interfacebaugruppe (z. B. Chip-Enable-Eingang beim PIO- oder SIO-Schaltkreis) zugeordnet.

3.3.3. Startlogik

Der Mikroprozessor U 880 kann grundsätzlich über $\overline{\text{RESET}}$ oder über Interrupt gestartet werden (siehe [1]). Für den Urstart des Systems verwendet man üblicherweise das bereits erwähnte „Power-On-RESET“. Hierbei beginnt die CPU die Befehlsbearbeitung bei der Speicherzelle 0. Aus diesem Grund ist es häufig so, daß der Beginn des Betriebssystems des Mikrocomputers (z. B. eines Monitorprogramms) auf die Speicheradresse 0 gelegt wird. Diese Methode ist mit einigen Nachteilen verbunden, die den Ausschlag für die Wahl einer anderen Lösung gaben. Die genannten Nachteile sind vor allem softwarebedingt und ergeben sich daraus, daß im oben beschriebenen Fall (d. h. Monitorbeginn bei Adresse 0) ab Adresse 0 kein frei verfügbarer Speicherbereich vorhanden ist. Dies soll im folgenden kurz erläutert werden.

Für Mikrocomputer auf der Basis des U 880 ist heute ein breites Softwareangebot bekannt, das von Assembler- und Disassemblerprogrammen bis zu Interpreter- bzw. Compilerprogrammen für höhere Programmiersprachen (z. B. BASIC, PASCAL, FORTRAN usw.) reicht. In vielen Fällen läuft diese Standardsoftware ab Adresse 0. Ein Umschreiben bewährter Programme für andere Speicherbereiche ist mit großem Aufwand und entsprechend vielen Fehlermöglichkeiten verbunden. Als weiterer Aspekt ist zu beachten, daß einige der genannten Programme nur auf RAM-Speichern lauffähig sind, da sich in diesem Fall die notwendigen Arbeitszellen innerhalb des jeweiligen Programmspeicherbereichs befinden (also nicht in andere RAM-Bereiche ausgelagert sind).

Im Anfangsbereich des Adreßraums des U 880 befinden sich einige Adressen mit besonderer Bedeutung (00H, 08H, 10H ... 38H, 66H: Restart bzw. Interruptmode 1 und NMI), die möglichst allgemein verfügbar sein sollten.

Aus den genannten Gründen wurde im System die Voraussetzung dafür geschaffen, daß – beginnend bei der Adresse 0 – ein möglichst großer, zusammenhängender RAM-Bereich realisiert werden kann. In diesem Zusammenhang müssen noch ein paar Überlegungen zur Speicherbereichsaufteilung angestellt werden:

Man benötigt für das System unbedingt ein Monitorprogramm (siehe Abschnitt 9.), welches das Arbeiten auf Maschinensprachniveau sowie die Handhabung aller weiteren Software ermöglicht. Als wichtiges, den Dialog zwischen Mensch und Computer unterstützendes peripheres Gerät wird ein Fernsehinterface benötigt (siehe Abschnitt 5.). Wie sich zeigen wird, kann man dieses durch den Computer wie einen normalen RAM-Speicherbereich ansprechen.

Im Ergebnis der Überlegungen kommt man zu der im Bild 3.2 angegebenen Speicherbereichsaufteilung.

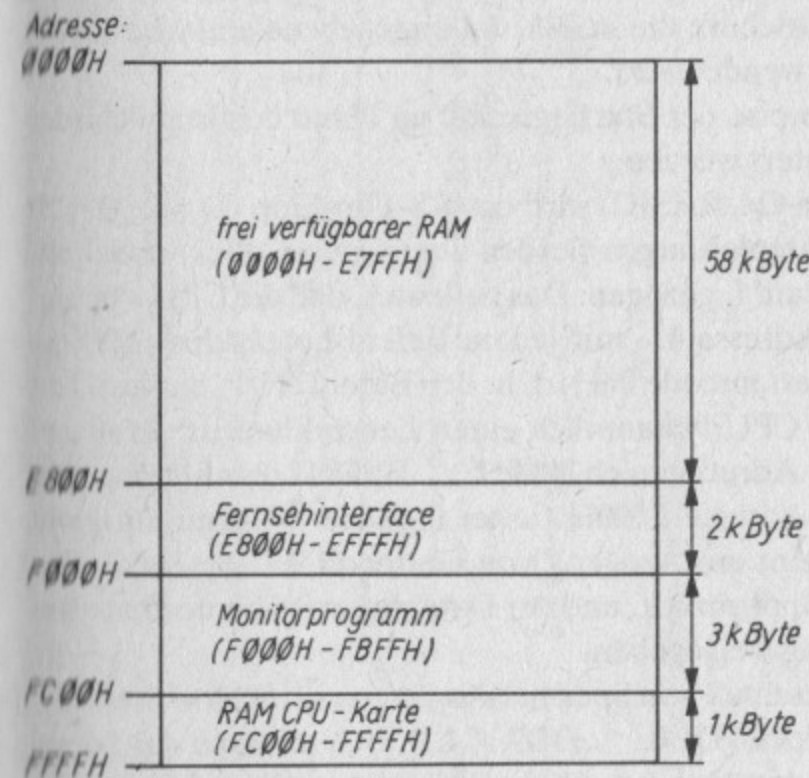


Bild 3.2 Speicherbereichsaufteilung

Für den Bildwiederholpeicher des Fernsehinterface werden 2-kByte im Adreßbereich E800H . . . EFFFH reserviert. Das in Abschnitt 9. beschriebene Monitorprogramm wird im Bereich F000H FC00H untergebracht. In den bis zum Ende des Adreßraums verbleibenden 1-kByte-Bereich paßt genau der Arbeitsspeicher (1-kByte-RAM) der CPU-Platine.

Damit sind die oberen 6 kByte des 64-kByte-Adreßbereichs des Computers belegt, und es steht ein zusammenhängender Adreßraum von 58 kByte (0000H . . . E7FFH) für Anwenderprogramme zur freien Verfügung.

Bei der gewählten Speicheraufteilung muß der Beginn der Programmbearbeitung bei der vereinbarten Startadresse des Monitorprogramms F000H gewährleistet werden. Hierfür gibt es hard- und softwareorientierte Möglichkeiten. Beispielsweise kann man den Monitor einschließlich einer einfachen Laderoutine ab Adresse 0 auf PROM unterbringen. Die Laderoutine dupliziert dann nach Netzeinschalten den Monitor zur Adresse F000H, wo in diesem Fall natürlich ein RAM-Speicher zur Verfügung stehen muß. Nach Beendigung dieses Vorgangs wird der PROM deselektiert, damit ab Adresse 0 ein frei verfügbarer Speicher vorhanden ist. Auf weitere Einzelheiten dieser Variante soll nicht eingegangen werden, da in dieser Broschüre die in Bild 3.1 angegebene einfache Hardwarelösung verwendet wird.

Die Funktionsweise der Startlogik soll an Hand des Impulsbildes (Bild 3.3) erläutert werden.

Mit dem Power-On-RESET wird das RS-Flip-Flop (D 6.2, D 6.3) gesetzt, alle Datenleitungen werden über Open-Collektor-Schaltkreise D 7, D 8 auf L gezogen. Das bedeutet, daß die CPU – beginnend bei der Adresse 0 – mit jedem Befehlsholezyklus (M1-Zyklus) den Operationscode 00H, d. h. den Befehl NOP, einliest. Mit NOP führt die CPU bekanntlich einen Leerzyklus aus. In dieser Weise wird der Adreßbereich 0000H . . . EFFFH durchlaufen. Bei Erreichen der Adresse F000H (unter logischer Verknüpfung mit \overline{MREQ}) erscheint am Ausgang von Gatter D 9 L-Potential, das RS-Flip-Flop kippt zurück, und der Datenbus wird für normale Befehlsbearbeitung freigegeben.

Um einen Kurzschluß von Speicherausgängen beim Hochstartvorgang zu vermeiden (DB0 . . . DB7 = L!), benutzt man das Signal MEMDES am Ausgang von Gatter D 2.4 dazu, über den Toreingang G 2 der zentralen Speicherselektierung sämtliche angeschlos-

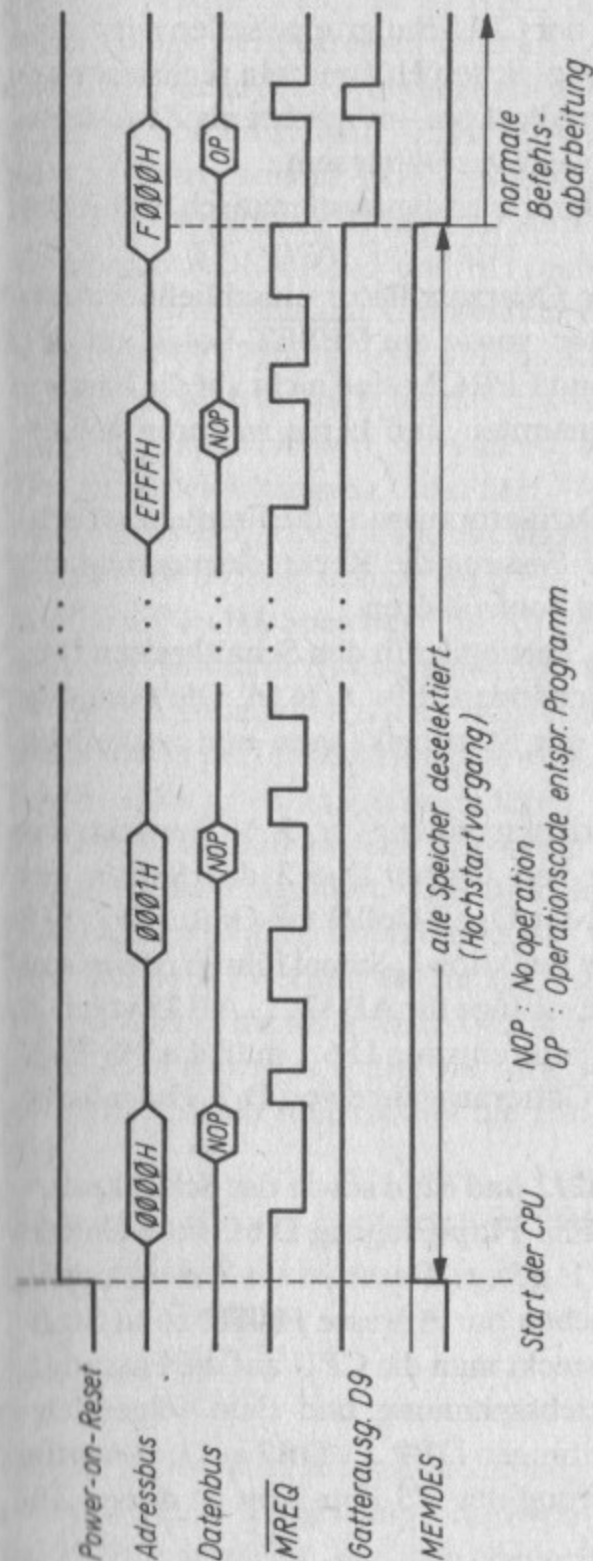


Bild 3.3 Impulsdiagramm Startlogik

senen Speicherschaltkreise zu deselektieren (siehe Abschnitt 3.3.1.).

3.4. Aufbau und Inbetriebnahme

Aufbau und Inbetriebnahme der CPU-Baugruppe sollen mit einfachen, auch dem Amateur zugänglichen Hilfsmitteln realisiert werden. Es müßte jedoch ein Oszilloskop – möglichst als Zweistrahlenausführung – für diesen Zweck verfügbar sein.

Die CPU-Platine kann man dann wie folgt systematisch bestücken und erproben:

1. Zunächst werden nur der Quarzoszillator einschließlich 1:4-Untersetzer und Puffergatter sowie die RESET-Logik auf der Leiterplatte bestückt. CPU und EPROM sind nicht auf die Fassungen gesteckt. Außer den genannten sind keine weiteren Schaltkreise eingelötet.

Mit dem Oszilloskop ist am Oszillatorausgang die Frequenz von 10 MHz und am Pin 6 der CPU-Fassung die Rechteckspannung mit der Frequenz von 2,5 MHz zu kontrollieren.

2. Danach bestückt man die Startlogik mit den Schaltkreisen D 6, D 7, D 8, D 9 und den Widerständen R 9 . . . R 16. Die Funktion der RESET-Schaltung und der Startlogik kann nun zusammen kontrolliert werden.

Nach Einschalten der Betriebsspannung + 5 V bewirkt das RESET-Signal am Eingang von Gatter D 6.2 das Setzen des RS-Flip-Flop, alle Ausgänge der Open-Collektor-Gatter D 7, D 8 müssen bei ordnungsgemäßer Funktion L-Signal führen. Beim Anlegen eines H-Signals an die Eingänge für AB 12 . . . AB 15 von D 9 und eines L-Signals an den Gattereingang D 6.1 muß das RS-Flip-Flop zurückkippen, und die Gatterausgänge von D 7, D 8 müssen auf H liegen.

3. Einlöten der Bustreiber 8212 und 8216 sowie des Schaltkreises D 4. Anschließend wird der Flip-Flop-Eingang D 6.3 vom Gatterausgang D 9 getrennt und auf H gelegt. Damit ist ein Zurückkippen des RS-Flip-Flop nach Erreichen der Adresse F000H beim Startvorgang verhindert. Zuletzt steckt man die CPU auf die Fassung.

Nach Einschalten der Betriebsspannung und dem folgenden RESET werden alle Datenleitungen DB0 . . . DB7 auf L-Potential gezogen. Durch die Blockierung des RS-Flip-Flop ist dieser Zustand permanent vorhanden.

Infolge dieser Maßnahme liest die CPU ausschließlich NOP-Befehle ein, und der gesamte Adreßbereich wird zyklisch durchlaufen. Mit dem Oszilloskop kontrolliert man jetzt die Adreßleitungen AB0 . . . AB15. Die an AB0 anliegende Pulsfolge weist die höchste Frequenz auf, die an AB15 die niedrigste. In ansteigender Reihenfolge der Adressen muß an der jeweiligen Adreßlinie die halbe Frequenz der vorhergehenden anliegen. Obwohl die Impulsfolge an den einzelnen Adreßausgängen wegen des Refresh-Zyklus unterbrochen ist, muß der beschriebene Sachverhalt deutlich auf dem Oszilloskop erkennbar sein. Auch an den Steuersignalausgängen RD, MREQ und M1 muß eine Impulsfolge anstehen.

4. Nun wird noch die Chipselektierung für die Speicherschaltkreise der CPU-Baugruppe überprüft.

Dazu lötet man als nächstes die Schaltkreise D 17 und D 5 ein. Die Datenleitungen bleiben, wie unter 3. beschrieben, auf L-Potential. Der Blockselekteingang C des MH 7442 (D 17) wird vom Ausgang des Gatters D 9 getrennt und auf Masse gelegt. Nach dem Einschalten der Betriebsspannung muß ein Impuls an den CS-Eingängen der EPROM-Speicher (Pin 20) bzw. des RAM-Blocks (Pin 13) anliegen. Mit dem Zweistrahlenszilloskop kann überprüft werden, ob die 4 CS-Signale – wie es notwendig ist – zeitlich versetzt auftreten. Man darf natürlich nur jeweils 1 Speicherschaltkreis (bzw. RAM-Block) gleichzeitig selektieren.

5. Zum Abschluß werden alle noch fehlenden Schaltkreise und Bauelemente eingelötet. Die Blockierung des RS-Flip-Flop der Startlogik wird aufgehoben, und alle Verbindungen gemäß Schaltung in Bild 3.1 werden wieder hergestellt.

Wer jetzt über die Möglichkeit verfügt, einen PROM programmieren zu lassen (was er später zur Inbetriebnahme des Monitor-Programms ohnehin muß), kann die Ein-/Ausgabefunktion noch testen.

Dazu programmiert man einen EPPROM mit:

```
M1:    IN A
        OUT A
        INC C
        JR M1
```

Der EPROM wird auf den Steckplatz 1 der CPU-Platine gesteckt und das kleine Programm mit Power-On-RESET und Startlogik bei F000H gestartet. Auf den Steuerleitungen IORQ, RD und

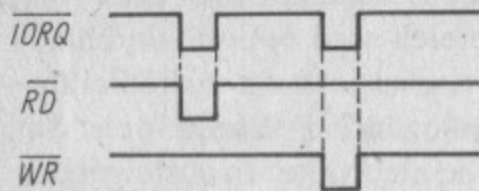


Bild 3.4 Impulsdiagramm Ein-/Ausgabefunktion

\overline{WR} muß der in Bild 3.4 dargestellte Impulsverlauf nachweisbar sein.

Die Ausgänge der I/O-Portdecodierung müssen I/O-Selektsignale aufweisen.

Diesen Zeitverlauf kontrolliert man mit dem Zweistrahloszilloskop.

Der Test ermöglicht gleichzeitig noch einmal den Nachweis für die einwandfreie Funktion der Startlogik.

Nach positivem Abschluß der Inbetriebnahme gemäß 1. bis 5. sind wesentliche Funktionen der CPU-Baugruppe überprüft. Man kann jetzt mit hoher Wahrscheinlichkeit davon ausgehen, daß ihre ordnungsgemäße Arbeitsweise abgesichert ist. Eine Gesamtüberprüfung stellt dann noch einmal die später folgende Inbetriebnahme des Monitorprogramms im Zusammenhang mit den peripheren Baugruppen dar.

Es soll hier aber auch nicht unerwähnt bleiben, daß manchmal bestimmte versteckte Fehler erst nach längerer Betriebszeit entdeckt werden.

4. Speichererweiterung

Wie bereits erwähnt, stellt die in Abschnitt 3. beschriebene CPU-Baugruppe einen in sich funktionsfähigen Einplatinenrechner dar. Zu einem leistungsfähigen Mikrocomputersystem gehört selbstverständlich mehr, vor allem eine ausreichend aufgerüstete Speicherbaugruppe.

Im Zusammenhang mit der Startlogik wurde darauf hingewiesen, daß im vorliegenden Systemkonzept ein möglichst großer, zusammenhängender RAM-Bereich bevorzugt wird. Voraussetzung für die Abarbeitung häufig benötigter Programme im RAM-Bereich ist aber ihr vorheriges Laden von einem externen Speichermedium (Kassette, Lochband, Floppy-Disk) bzw. von einem Festwertspeicher (PROM bzw. EPROM). Es ist leicht einzusehen, daß es nicht möglich ist, beispielsweise einen 8 . . . 12 kByte langen BASIC-Interpreter jedesmal vor einer gewünschten Benutzung über die Tastatur einzugeben.

Weiterhin ist es sinnvoll, ein häufig benötigtes Programm, wie z. B. den erwähnten Monitor, direkt auf PROM-Speicher laufen zu lassen. Voraussetzung ist natürlich, daß er dafür geeignet ist.

Im folgenden werden die Schaltungen für die zur Speichererweiterung notwendigen ROM- und RAM-Platinen angegeben und diskutiert. Als RAM-Speicher werden je eine Schaltungsvariante auf der Basis des statischen RAM U 202 sowie des für einen effektiven Speicherausbau günstigen dynamischen RAM U 256 (4116) vorgestellt.

An dieser Stelle ist es erforderlich, auf das Problem der Pufferung der Busleitungen auf Speicher- und Peripheriebaugruppen kurz einzugehen.

Der Datenverkehr im Computer ist grundsätzlich bidirektional. Das bedeutet also, daß alle Datenquellen (außer der CPU auch Speicher und Peripherie) in der Lage sein müssen, den Datenbus mit allen angeschlossenen Systemkomponenten hinsichtlich Lastfaktor und kapazitiver Belastung zu treiben.

Aus der Sicht der CPU ist dies durch die eingesetzten Treiberschaltkreise abgesichert. Für Speicher- und Peripheriebaugruppen in größeren Mikrocomputersystemen können sich aber in diesem

Zusammenhang Probleme ergeben. Besonders beachten muß man die Kapazitäten der Ein- und Ausgänge von Speicherschaltkreisen, die sich durch die übliche Parallelschaltung der Adreß- und Datenleitungen entsprechend vergrößern.

Aus den genannten Gründen ist es bei allgemein verwendbaren OEM-Computern wie z. B. K 1520 (OEM: *Original Equipment Manufactures*) sowie in großen Systemen üblich und notwendig, die Leitungen auf der jeweiligen Speicher- bzw. Peripheriebaugruppe zu puffern. Aus Gründen des Schaltkreisbedarfs ist es im vorliegenden System vertretbar, auf diese Pufferung zu verzichten. Dabei muß man darauf achten, daß die Belastung des Busses kleiner als eine TTL-Last bleibt und sich die oben erwähnten Lastkapazitäten in annehmbaren Grenzen halten.

Wer über ausreichend Treiberschaltkreise verfügt, sollte die Busleitungen auf den Speicherplatinen (z. B. PROM mit 8212, RAM mit 8216) bzw. Peripheriebaugruppen jedoch puffern.

4.1. ROM-Speicher

Bild 4.1 zeigt die Schaltung für eine 8-kByte-Prom-Leiterplatte auf der Basis des Schaltkreises U 555 (2708).

Mit AB0 . . . AB9 werden die 1k 8-Bit-Speicherzellen auf dem jeweiligen Chip adressiert. Das Chipselektsignal für die 8 auf der Leiterplatte platzierten PROM wird durch 1-aus-8-Decodierung der Adressen AB10, AB11, AB12 sowie des in Abschnitt 3.3.1. erläuterten Blockselektsignals der zentralen Speicherselektierung erzeugt. Für diesen Zweck nutzt man den 1-aus-10-Decoder MH 7442. Dem Problem direkt angepaßt ist natürlich der eventuell schwieriger beschaffbare 1-aus-8-Decoder 8205.

Die Pufferung der Datenausgänge bleibt, wie oben ausgeführt, dem einzelnen überlassen. Falls sie gepuffert werden sollen, eignet sich dazu der Schaltkreis 8212. Dieser Schaltkreis bietet sich deshalb an, weil vom PROM-Speicher selbstverständlich nur Daten gelesen werden können (der Fall des PROM-Programmierens ist in diesem Zusammenhang nicht von Interesse) und damit nur Datenfluß in der einen Richtung PROM → CPU möglich ist.

Mit H-Pegel am STR-Eingang wird die 8-Bit-Information der jeweiligen Speicherzelle des gerade selektierten PROM in den D-Latches des 8212 gespeichert. Wenn $\overline{DS1} = L$ (entsprechend

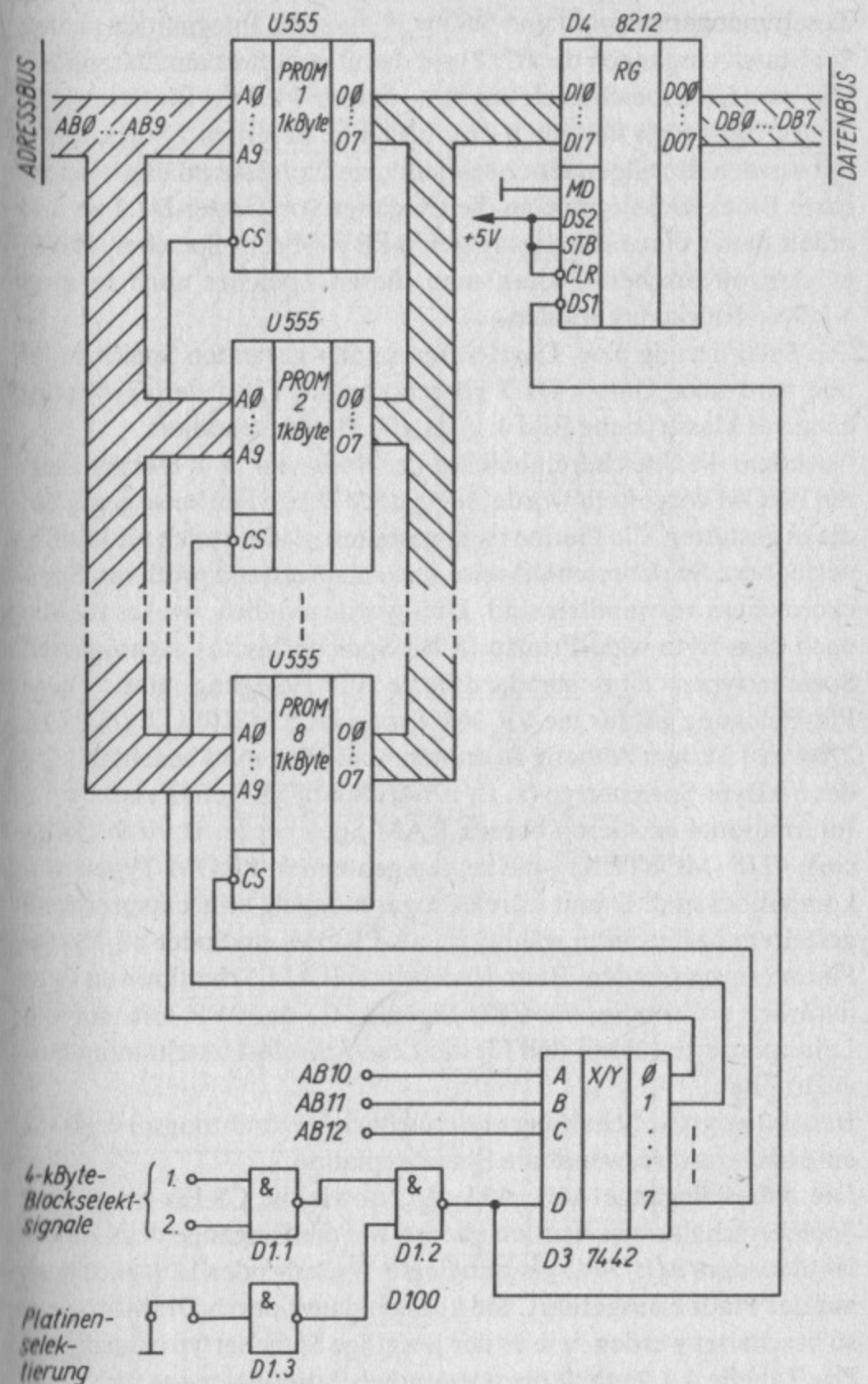


Bild 4.1 8-kByte-PROM-Platine mit U 555

dem Blockselektsignal) und $DS2 = H$, liegt die Information an den Tri-State-Ausgängen des 8212 und damit gepuffert am Datenbus. Die zentrale Speicherselektierung liefert 4-kByte-Blockselekt-signale, von denen für eine 8-kByte-Platine natürlich 2 Stück benötigt werden. Im allgemeinen schließt man 2 adressenmäßig benachbarte Blockselekt-signale an die Eingänge von Gatter D1.1 an und erhält damit einen durchgängigen 8-kByte-Prom-Speicher. Wenn es sinnvoll erscheint, kann man diesen Speicher auch in zwei 4-kByte-Blöcke unterteilen.

Die Selektierung bzw. Deselektierung der gesamten Speicherplatine wird über Gatter D1.3 vorgenommen. Liegt der Gattereingang auf Masse (siehe Bild 4.1), ist die Platine selektiert.

Nachdem die Speichereinheit auf der Basis von $1k \times 8$ -organisierten PROM vorgestellt wurde, seien noch einige Hinweise gegeben, die es gestatten, die Platine so zu gestalten, daß sie auch für künftig verfügbare Speicherschaltkreise mit entsprechend größeren Speicherdichten verwendbar sind. Dies wurde möglich, weil es für alle nach dem Byte-wide-Prinzip (8 Bit Speicherbreite) organisierten Speichertypen eine standardisierte Pin-Belegung gibt. Diese Pin-Belegung gilt für die PROM-Typen U 555 (2708), 2716, 2732, 2764 und ist dem Anhang zu entnehmen. Man muß beachten, daß der 8-kByte-Speichertyp (z. B. 2764) 28polig ausgeführt ist.

International existieren bereits RAM-Speicher [z. B. 6116 (Hitachi), 4118 (MOSTEK)], die zu den genannten PROM-Typen pin-kompatibel sind. Damit wäre es sogar möglich, eine entsprechend gestaltete Leiterplatte wahlweise als PROM- und/oder als RAM-Platine zu verwenden. Beim Einsatz von RAM-Schaltkreisen ist es natürlich notwendig, die CPU-Signale \overline{RD} und \overline{WR} mit auf die Leiterplatte zu führen und für die Lese/Schreib-Umschaltung heranzuziehen.

Bild 4.2 zeigt die Schaltung einschließlich Verdrahtungsprinzip für eine universell verwendbare Speicherplatine.

Die Adreßeingänge A10, A11, A12 sowie die \overline{CS} -Eingänge der Speicherschaltkreise werden ebenso wie die Eingänge des Chipselektdecoders MH 7442 als unbelegte Wickel- oder Lötanschlüsse auf der Platine ausgeführt. Sie können damit durch Drahtbrücken so beschaltet werden, wie es der jeweilige Speichertyp erfordert. Die Tabelle 4.1. enthält die notwendige Adreßbelegung für vier 8-Bit-organisierte Speichertypen. Die Verdrahtung für den 1-kByte-Speicher (z. B. U 555) entspricht der in Bild 4.1, wobei man 2 Ein-

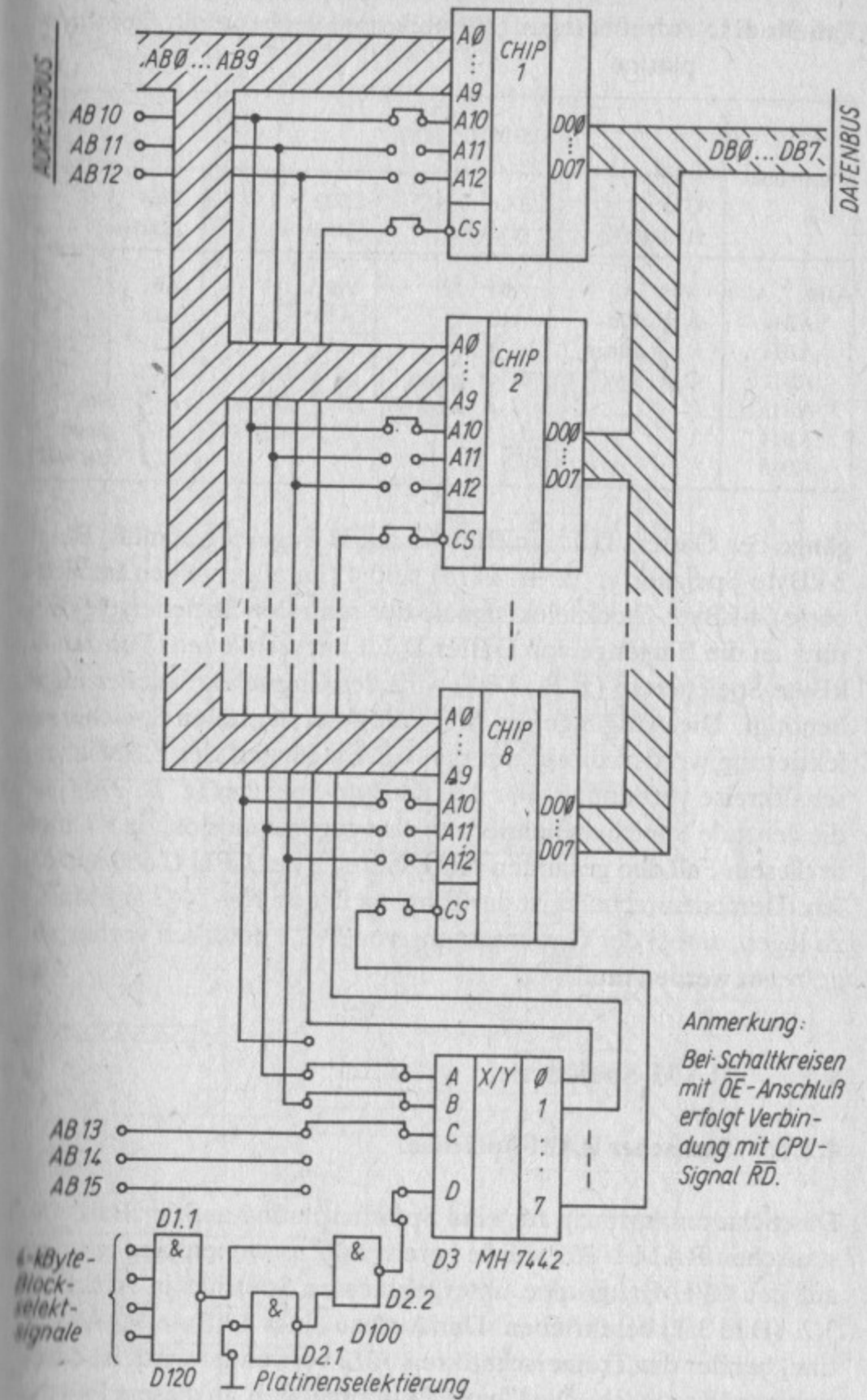


Bild 4.2 Universell verwendbare Platine für byte-wide-organisierte Speichertypen

Tabelle 4.1. Adreßbelegung für universell verwendbare Speicherplatine

Adreßbus	SPEICHERTYP						
	U 555 (1 kByte)	2716 (2 kByte)	2732 (4 kByte)	2764 (8 kByte)			
AB0...AB9	A0...A9	A0...A9	A0...A9	A0...A9			
AB10	A	A10	A10	A10			
AB11	B	A	A11	A11			
AB12	C	B	A	A12			
AB13	} Eingänge MH7442	} Eingänge MH7442	} Eingänge MH7442	} Eingänge MH7442			
AB14					/	/	/
AB15					/	/	/

gänge des Gatters D 1.1 in Bild 4.2 auf H-Pegel legen muß. Beim 2-kByte-Speichertyp (z. B. 2716) sind 4 (im allgemeinen benachbarte) 4-kByte-Blockselekt-signale der zentralen Speicherselektierung an die Eingänge von Gatter D 1.1 anzuschließen. Für den 4-kByte-Speichertyp (z. B. 2732) wird der Chipselektdecoder nicht benötigt. Die Ausgänge des MH 74154 der zentralen Speicherselektierung werden direkt mit den CS-Eingängen der 8 Speicherschaltkreise verbunden. Für den 8-kByte-Speicher (z. B. 2764) ist die zentrale Speicherselektierung also gegenstandslos, da 8 Chips in diesem Fall den gesamten Adreßbereich der CPU U 880 ausfüllen. Dementsprechend ist der Eingang D des MH 7442 auf Masse zu legen, wobei der Gatterausgang von D 2.2 natürlich vorher abgetrennt werden muß.

4.2. RAM-Speicher

4.2.1. Statischer RAM-Speicher

Das Schaltungsprinzip für eine Speicherplatine auf der Basis des statischen RAM U 202 wurde bereits im Zusammenhang mit dem auf der CPU-Baugruppe untergebrachten Speicher in Abschnitt 3.2. (Bild 3.1) beschrieben. Der Aufbau einer 4-kByte-RAM-Platine, bei der der Treiberschaltkreis 8212 verwendet wird, ist damit sinngemäß möglich. Die Chipselektierung, d. h. in diesem Fall das Selektieren der vier 1-kByte-RAM-Blöcke, sowie die Lese/

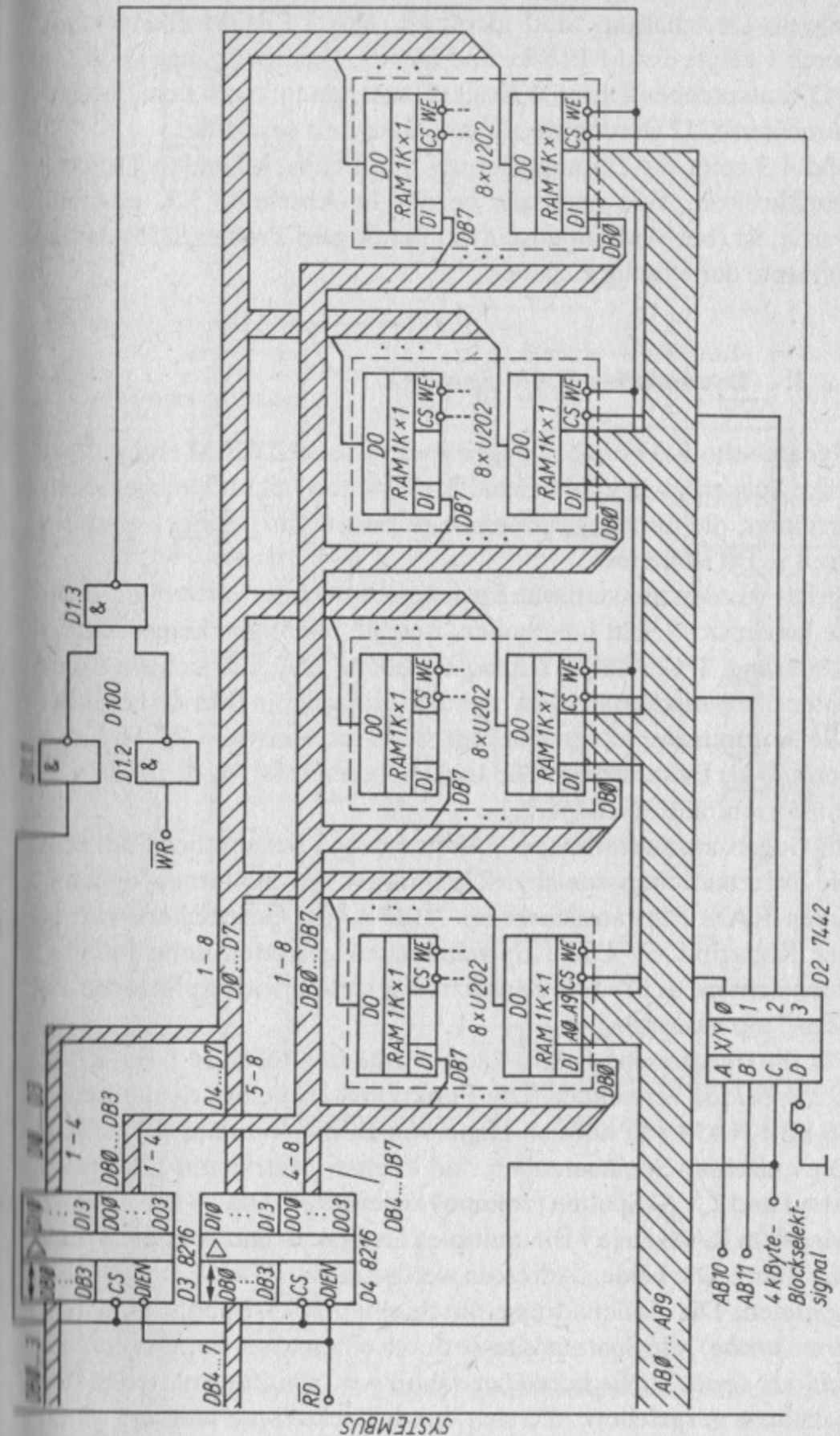


Bild 4.3 4-kByte-RAM-Speicherbaugruppe mit U 202

Schreib-Umschaltung sind identisch. Die 3 PROM ersetzt man durch 1-kByte-RAM-Blöcke und faßt die Datenausgänge DO0 . . . DO7 entsprechend ihrer Wertigkeit zusammen. Nach dem Puffern durch den 8212 werden sie auf den Datenbus geschaltet. Bild 4.3 zeigt die Datenpufferung mit 2 bidirektionalen Treiberschaltkreisen. Wie ebenfalls bereits in Abschnitt 3.3. erwähnt wurde, ist (bei Vorhandensein der benötigten Treiber 8216) dieser Variante der Vorzug zu geben.

4.2.2. Dynamischer RAM-Speicher

Dynamische RAM haben gegenüber statischen RAM einige Vorteile: So werden in einem Schaltkreis weitaus mehr Speicherzellen vereinigt, die Leistungsaufnahme ist bedeutend geringer, und der Preis je Bit ist kleiner.

Bisher wurden nur statische Speicherschaltkreise verwendet, denn sie lassen sich leicht handhaben, und sie benötigen keine externe Steuerung. Da es kaum Timing-Probleme gibt, läßt sich ein Computer mit statischen RAM wesentlich einfacher in Betrieb nehmen. Die Autoren empfehlen den Bau dieser dynamischen RAM-Karte deshalb als Erweiterung. Die Inbetriebnahme ist für den Amateur sonst eventuell zu schwierig.

Im Gegensatz zu statischen RAM, in denen bekanntlich Flip-Flop die Informationen speichern, enthalten die modernen dynamischen RAM Eintransistorzellen (Bild 4.4a). Gespeichert wird in der Kapazität C_s . Diese Speicherzellen gestatten hohe Integrationsdichten, d. h., es lassen sich sehr viele Speicherplätze auf einem Chip unterbringen.

Für die dynamische RAM-Karte wurde der 16384×1 -Bit-RAM U256 (4116) verwendet. Die Funktionsweise eines dynamischen 16-kBit-RAM soll kurz an Hand von Bild 4.4b erläutert werden. Die einzelnen Speicherzellen sind in einer Matrix von 128 Zeilen (row) und 2×64 Spalten (column) angeordnet. Die 14-Bit-Adresse wird dem RAM zu je 7 Bit multiplex als Zeilen- und Spaltenadresse zugeführt. Die beiden Adressen werden in zwei 7-Bit-Latches übernommen: Die Zeilenadresse durch einen \overline{RAS} -Impuls (row address strobe), die Spaltenadresse durch einen \overline{CAS} -Impuls (column address strobe). Wie bereits erwähnt, wird die Information in Kapazitäten gespeichert, die sich durch Leckströme langsam entla-

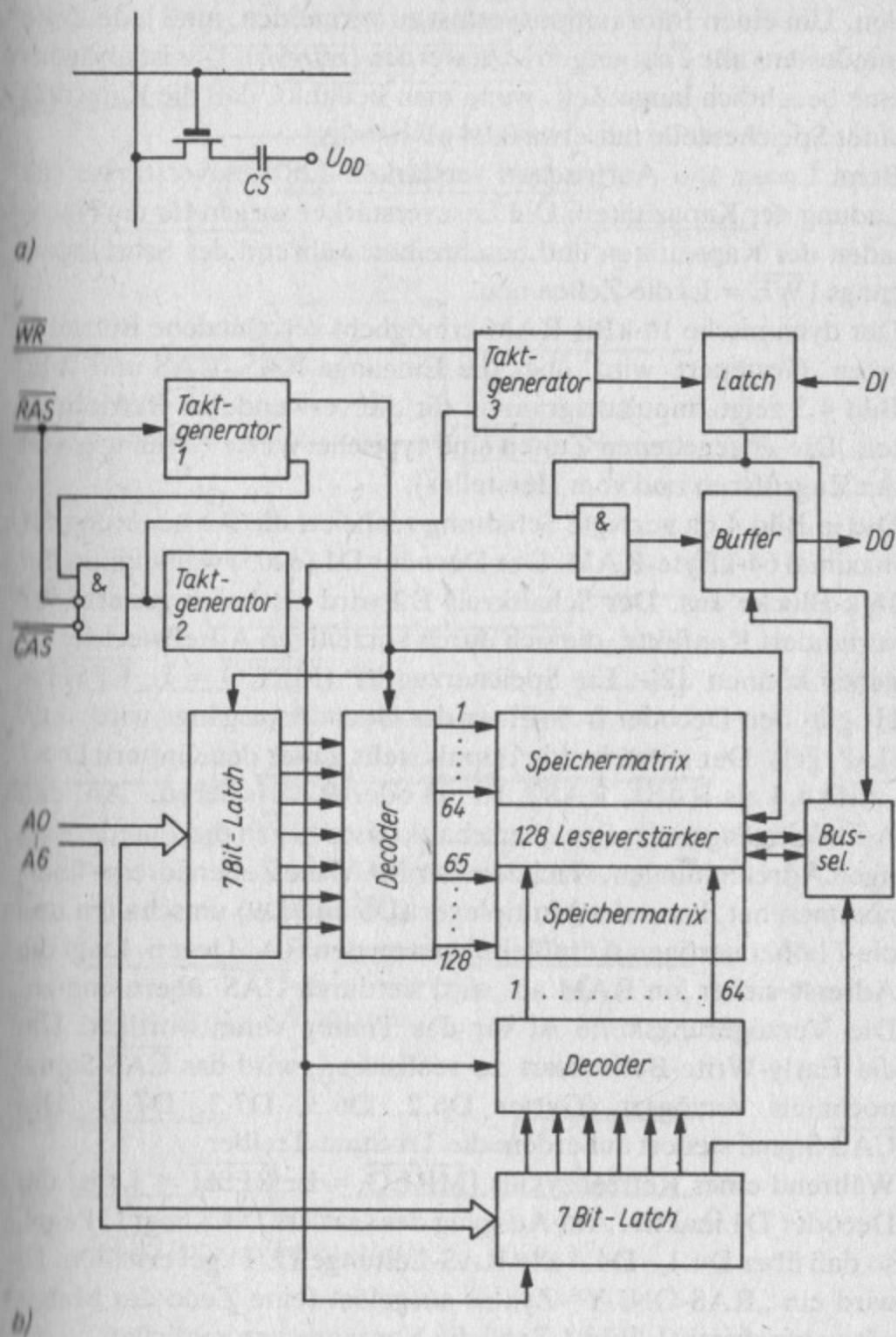


Bild 4.4 a - dynamischer RAM U 256 (4116), Eintransistorzelle, b - Übersichtsplan

den. Um einen Informationsverlust zu vermeiden, muß jede Zeile mindestens alle 2 ms aufgefrischt werden (*refresh*). Das ist übrigens eine beachtlich lange Zeit, wenn man bedenkt, daß die Kapazität einer Speicherzelle nur etwa 0,04 pF beträgt!

Beim Lesen und Auffrischen verstärken 128 Leseverstärker die Ladung der Kapazitäten. Die Leseverstärker sorgen für ein Nachladen der Kapazitäten und beschreiben während des Schreibvorgangs ($\overline{WE} = L$) die Zellen neu.

Der dynamische 16-kBit-RAM ermöglicht verschiedene Betriebsarten. Gesteuert wird über die Eingänge \overline{RAS} , \overline{CAS} und \overline{WE} . Bild 4.5 zeigt Impulssdiagramme für die verwendeten Betriebsarten. Die angegebenen Zeiten sind typische Werte (abhängig von der Zugriffszeit und vom Hersteller).

Die in Bild 4.6a gezeigte Schaltung realisiert die Ansteuerung für maximal 64-kByte-RAM. Der Decoder D1 (8205) wählt einen der 16-k-Blöcke aus. Der Schaltkreis D2 wird als Latch genutzt. Er verhindert Konflikte, die sich durch vorzeitigen Adreßwechsel ergeben können [2]. Ein Speicherzugriff ($\overline{MREQ} = L$, $\overline{RFSH} = H$) gibt den Decoder frei. Einer der Decoderausgänge wird aktiv (L-Pegel). Der entstehende Impuls steht hinter den Gattern D 4.1 ... D 4.4 als $\overline{RAS1}$, $\overline{RAS2}$, $\overline{RAS3}$ oder $\overline{RAS4}$ bereit. An den Adreßeingängen des Speicherschaltkreises liegen die 7 niederwertigen Adreßleitungen. Nachdem der RAM die Zeilenadresse übernommen hat, kann der Multiplexer (D8 und D9) umschalten und die 7 höherwertigen Adreßleitungen an den RAM legen. Liegt die Adresse sicher am RAM an, wird sie durch \overline{CAS} übernommen. Die Verzögerungskette ist für das Timing verantwortlich. Um die Early-Write-Betriebsart zu realisieren, wird das \overline{CAS} -Signal nochmals verzögert (Gatter D6.2, D6.3, D7.3, D7.4). Das \overline{CAS} -Signal steuert außerdem die Tri-State-Treiber.

Während eines Refreshzyklus ($\overline{MREQ} = L$, $\overline{RFSH} = L$) ist der Decoder D1 inaktiv. Am Ausgang des Gatters D3.3 liegt L-Pegel, so daß über D4.1 – D4.4 alle RAS-Leitungen L-Pegel erhalten. Es wird ein „RAS-ONLY“-Zyklus ausgelöst (eine Zeile der Matrix wird aufgefrischt). Bild 4.7 soll die Vorgänge verdeutlichen.

Die kleine Logik in Bild 4.6b mit den Gattern D13.1, D13.2 und D5.2 hat die Aufgabe, die oberen 6 kByte der 64-kByte-RAM-Karte auszublenden. Der Bereich von E8000H ... FFFFH wird bereits von den Speichern auf der CPU-Karte und dem Bildwiederholpeicher belegt.

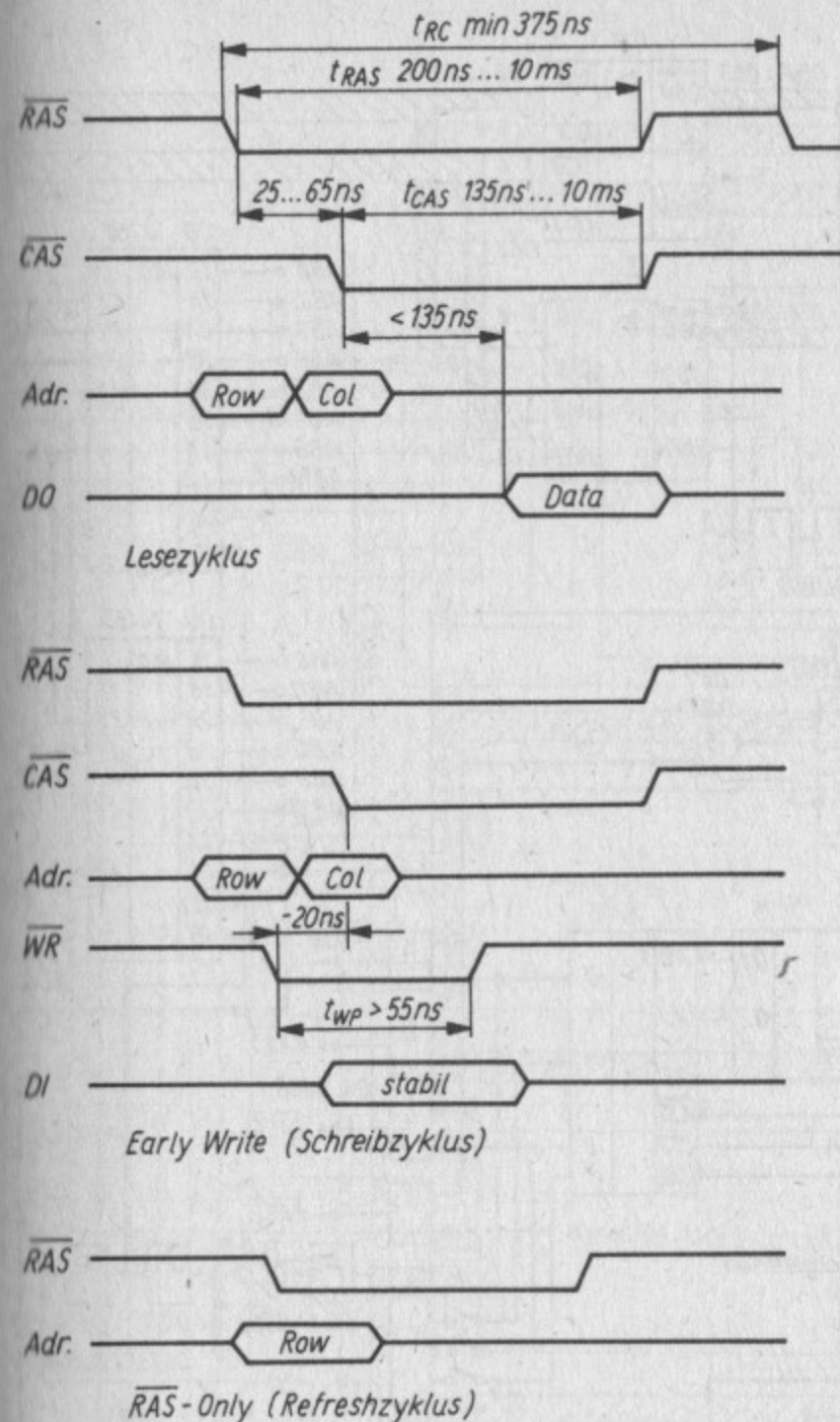
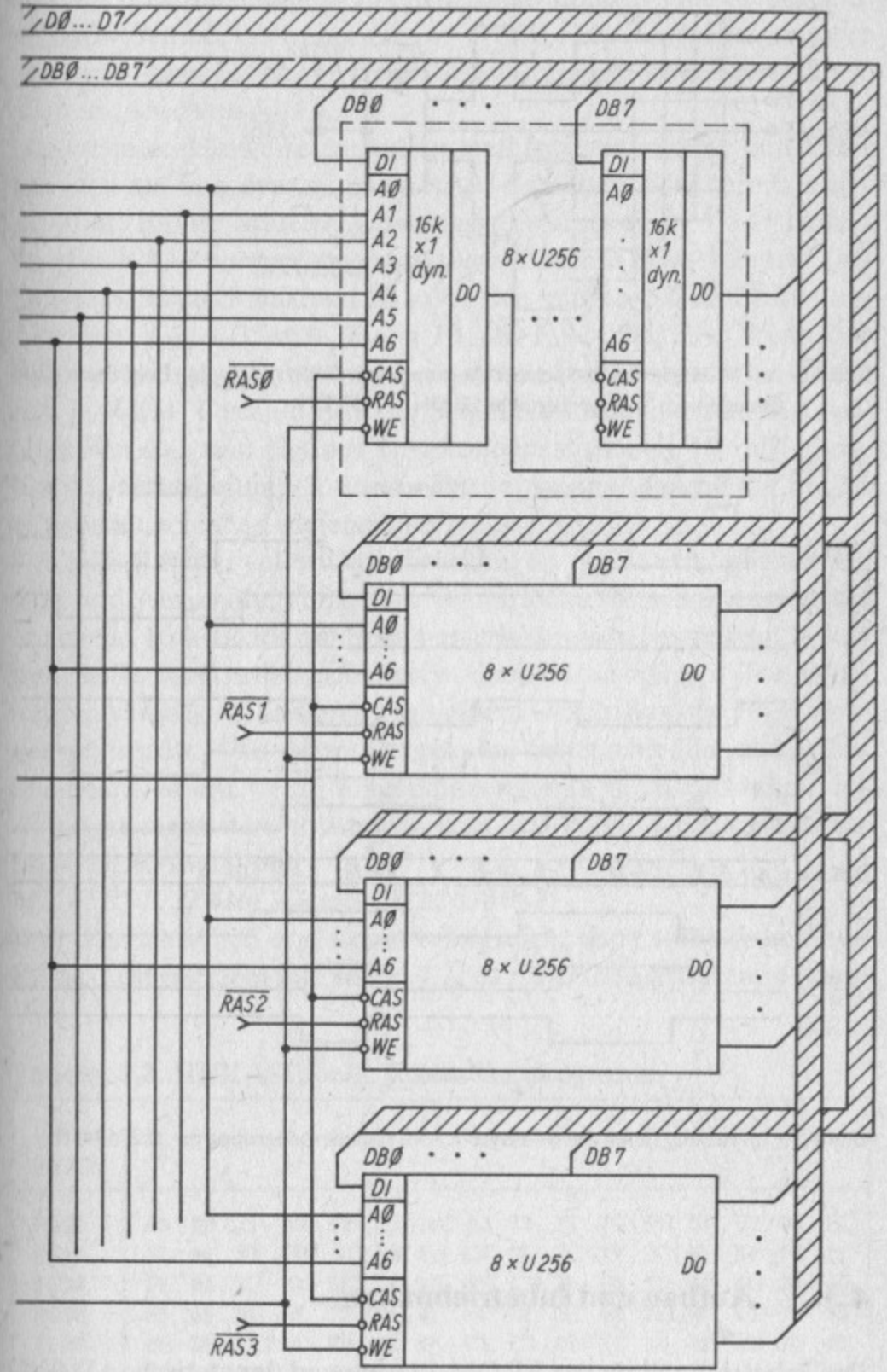
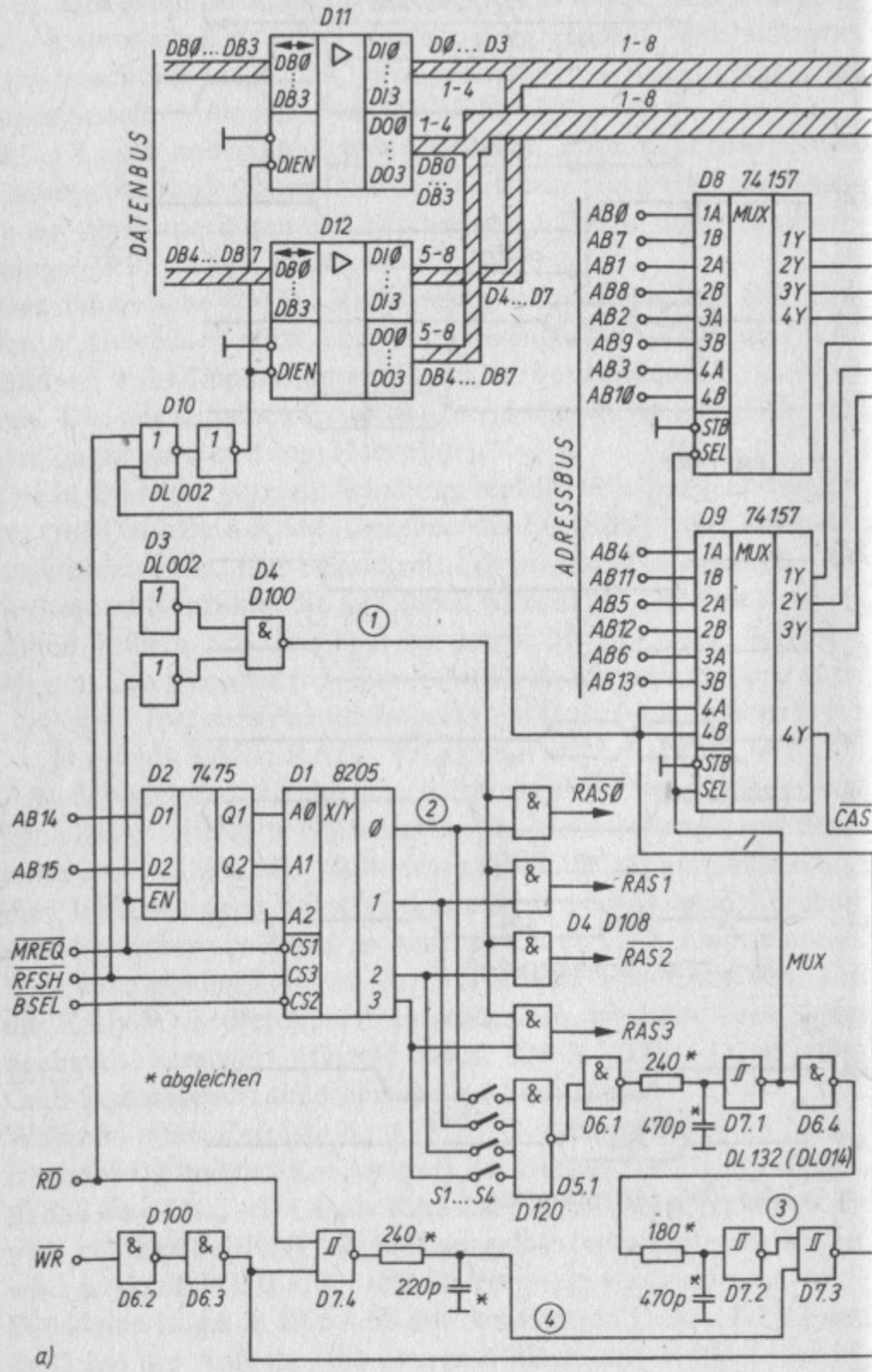
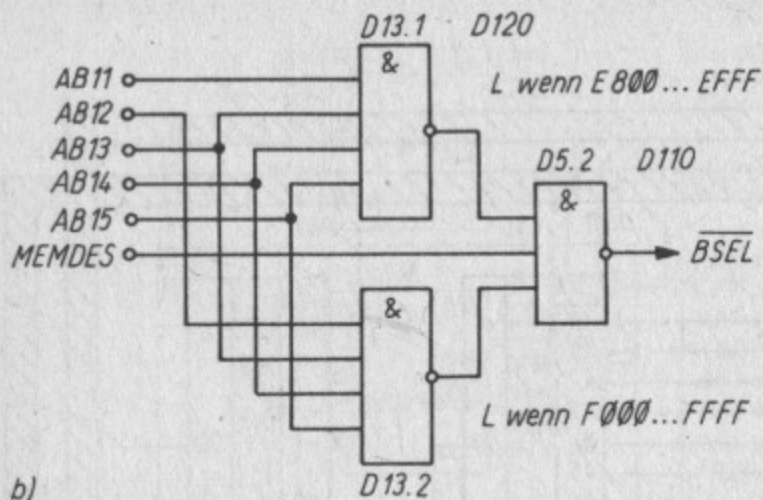


Bild 4.5 Impulssdiagramm der dynamischen RAM U 256 (4116)





b) Bild 4.6 a – 64-kByte-RAM-Speicherbaugruppe mit U 256 (4116), b – Logik zum Ausblenden des Speicherbereichs E800H ... FFFFH

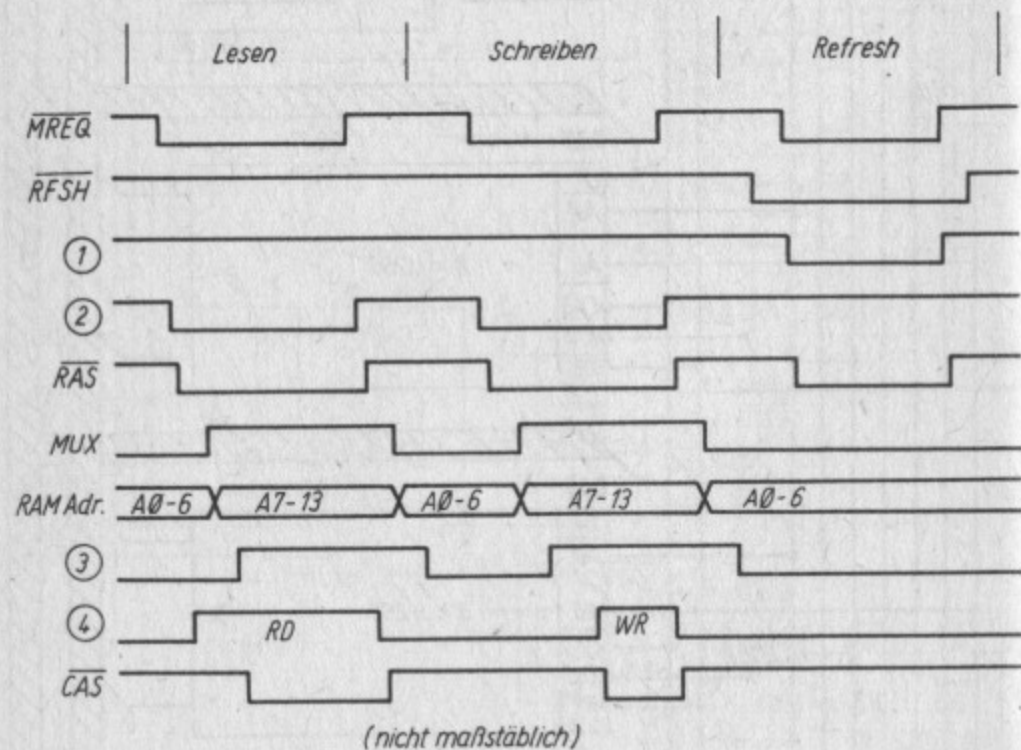


Bild 4.7 Impulssdiagramm der 64-kByte-RAM-Speicherbaugruppe mit U 256 (4116)

4.3. Aufbau und Inbetriebnahme

Die Inbetriebnahme der PROM-Platine und des statischen RAM-Speichers ist bei sauberem, übersichtlichem Aufbau problemlos.

Für den Einsatz des EPROM U 555 (2708) sowie des dynamischen RAM U 256 (4116) ist jedoch folgendes zu beachten:

Um ein Zerstören der genannten Speichertypen auszuschließen, muß die Betriebsspannung $U_{BB} = -5 V$ beim Einschalten zuerst und beim Ausschalten zuletzt am Schaltkreis anliegen (Einzelheiten dazu im Abschnitt 10.1.).

Die weiteren Hinweise zu Aufbau und Inbetriebnahme beschränken sich auf den dynamischen RAM-Speicher. Wie bereits oben erwähnt, ist der mittlere Leistungsverbrauch eines dynamischen 16-kBit-RAM nur gering: im aktiven Zustand etwa 420 mW, im Standby-Zustand maximal 20 mW. Die mittleren Betriebsströme betragen: $I_{DD} = 35 \text{ mA}$, $I_{CC} = 10 \mu\text{A}$, $I_{BB} = 200 \mu\text{A}$. Wenn der RAM jedoch angesteuert wird, entstehen Spitzenströme bis zu 100 mA je RAM. Deshalb ist jeder Schaltkreis besonders an den Anschlüssen U_{DD} und U_{BB} mit Stützkondensatoren (je 100 nF) abzublocken. Die Leitungen für die Betriebsspannung und die Masse müssen ausreichend dimensioniert werden.

Die DIL-Schalter (oder Drahtbrücken) S1 ... S4 ermöglichen das Ein- und Ausschalten einzelner 16-k-Blöcke. Man sollte zunächst nur einen 16-k-Block mit Speicherschaltkreisen bestücken. Wenn sich die Speicherzellen nicht beschreiben lassen oder gar ihre Information verändern, so werden mit einem Oszilloskop die Impulsbilder überprüft. Vor allen Dingen bei unterschiedlich schnellen Speichern ist ein wenig Fingerspitzengefühl beim Einstellen der Verzögerungszeiten notwendig. Die RC-Werte sind erprobt für folgende RAM-Typen:

MK 4116-3, D 416C-2 und TMM 416P-3.

War das Schreiben und Lesen erfolgreich, dann sollte man einen RAM-Test mit dem in Tabelle 4.2. dargestellten Programm (nach

Tabelle 4.2. HEXA-Tabelle RAM-Testprogramm

Adresse HEXA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
FC00	06	00	21	00	ST	7D	AC	A8	77	23	7C	FE	SP	C2	05	FC
FC10	21	00	ST	7D	AC	A8	BE	C4	25	FC	23	7C	FE	SP	C2	13
FC20	FC	04	C3	02	FC	C5	0E	05	CD	21	F0	52	41	4D	20	46
FC30	65	68	6C	65	72	20	41	64	72	2E	00	0E	04	CD	21	F0
FC40	0E	05	CD	21	F0	1E	00	C1	C9	ED	FF	52	41	4D	54	45
FC50	53	54	00	C3	00	FC										

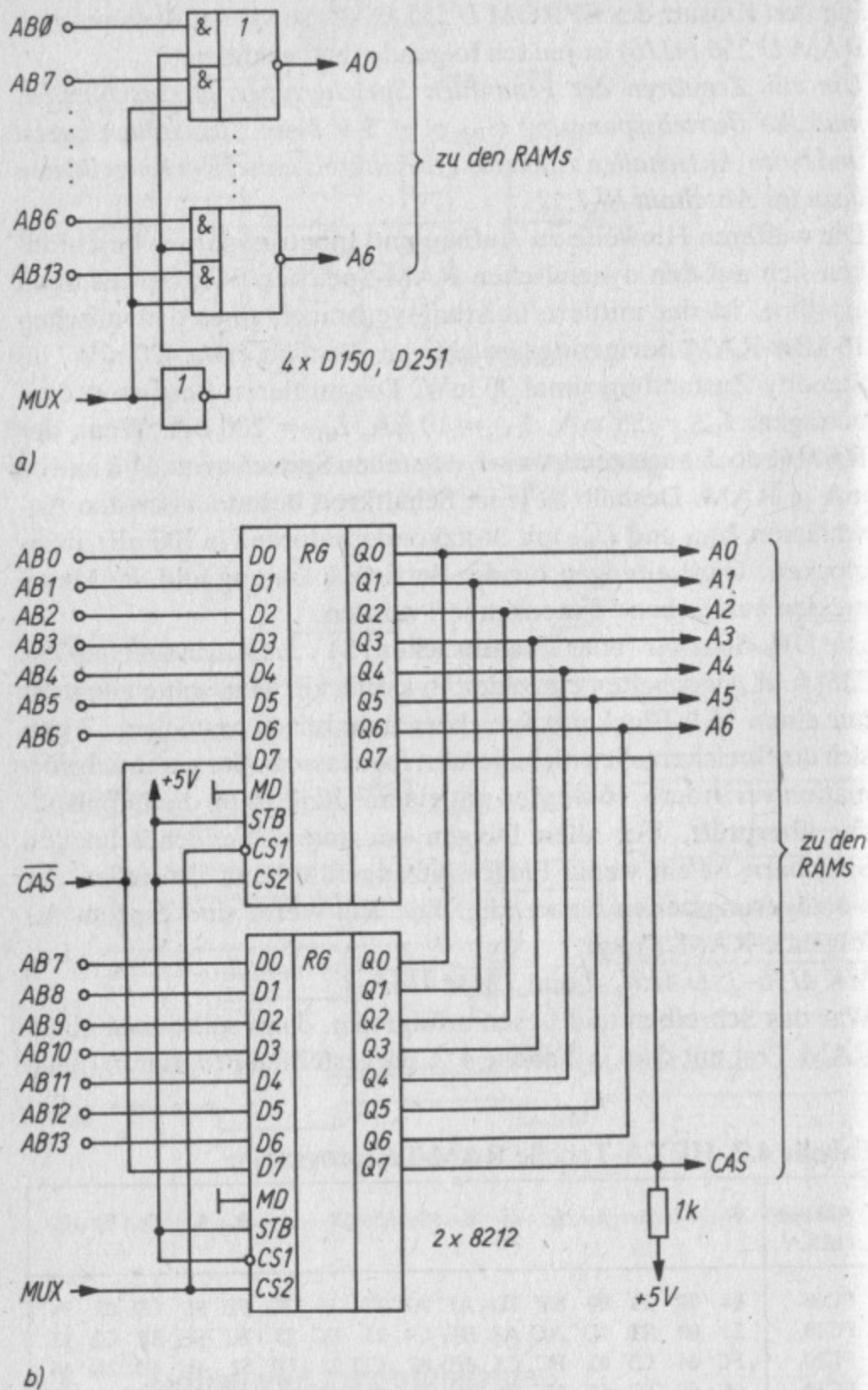


Bild 4.8 Adreßmultiplexer; a – mit D 150, b – mit 8212

MOSTEK) anschließen. Das Programm arbeitet mit dem in Abschnitt 9. beschriebenen Monitor zusammen. Einzutippen ist das Programm ab Adresse FC00H. ST bedeutet H-Teil der Anfangsadresse des zu testenden Speicherbereichs, SP bedeutet H-Teil der Endadresse. Die 4 Byte muß man selbst eintragen.

Gestartet wird das Testprogramm ab Adresse FC00H oder durch Eingabe des Befehls RAMTEST. Der Test eines 16-kByte-Speichers dauert etwa 4 Minuten. Der Speicher läßt sich beliebig lange testen (Abbruch durch RESET oder NMI). Alle fehlerhaften Speicheradressen werden ausgegeben. Auch wenn dieser Test erfolgreich war, ist noch nicht hundertprozentig sicher, daß der Speicher funktioniert. Es kann sein, daß die Zugriffszeit der RAM-Karte während des M1-Zyklus nicht klein genug ist. Dann müssen die Zeitkonstanten der RC-Glieder der Verzögerungskette geringfügig verkleinert werden.

Ein Hinweis noch zu den verwendeten Bauelementen: Als Multiplexer wurden Schaltkreise vom Typ 74157 eingesetzt. Bild 4.8 zeigt 2 Alternativen. Um den Datenbus nicht zu belasten, benutzt man Treiberschaltkreise vom Typ 8216.

5. Fernsehinterface

Die Kommunikation zwischen Mensch und Computer setzt geeignete Ein-/Ausgabemedien voraus. Von besonderer Bedeutung in diesem Zusammenhang sind Geräte bzw. Systemkomponenten, die den visuell unterstützten Dialog ermöglichen. Aus der Vielzahl der Displayvarianten hat sich als universelles Anzeigemedium für Ziffern- und Zeichendarstellung bis hin zur Graphik der Bildschirm durchgesetzt. Das gilt in besonderem Maße für den Bereich der Mikrocomputer. Die Industrie bietet hierfür spezielle Bildschirmgeräte (vom Schwarzweiß- bzw. einfarbigen Monitor bis hin zum Farbmonitor) an.

Für viele Belange, vor allem natürlich aus der Sicht des Amateurs, erweist sich für den genannten Zweck der Einsatz eines handelsüblichen Fernsehgeräts als technisch und ökonomisch sinnvoll. Im folgenden wird ein relativ komfortables, problemlos nachbaubares Fernsehinterface beschrieben.

5.1. Zeichendarstellung auf dem Bildschirm

Das vom Fernsehen bekannte Darstellungsprinzip beruht auf der Helligkeitssteuerung des Elektronenstrahls in der Bildröhre. Der abhängig von der Videoinformation getastete Strahl überstreicht dabei in einem Zeilenraster den Bildschirm von oben nach unten. Das geschieht nach CCIR-Norm mit einer Zeilenfrequenz von 15625 Hz (d. h. einer Zeilendauer von $64 \mu\text{s}$) und 625 Zeilen je Bild. Daraus ergibt sich eine Bildwechselfrequenz von 25 Hz. Durch den Trick des Zeilensprungverfahrens erzeugt man eine Rasterwechselfrequenz von 50 Hz. Durch das Verkoppeln der Bildfrequenz wird ein flimmerfreies Bild abgesichert.

Zum Darstellen der Zeichen auf dem Bildschirm sollte man auf das Zeilensprungverfahren verzichten und statt dessen 2 identische Halbbilder mit etwa der Hälfte der 625 Fernsehzeilen erzeugen. Damit ist die für eine ausreichende Flimmerfreiheit des Bildes notwendige Bildfrequenz von etwa 50 Hz garantiert.

Die Anzahl der Fernsehzeilen (gewählt wurden 320 je Bild) ermög-

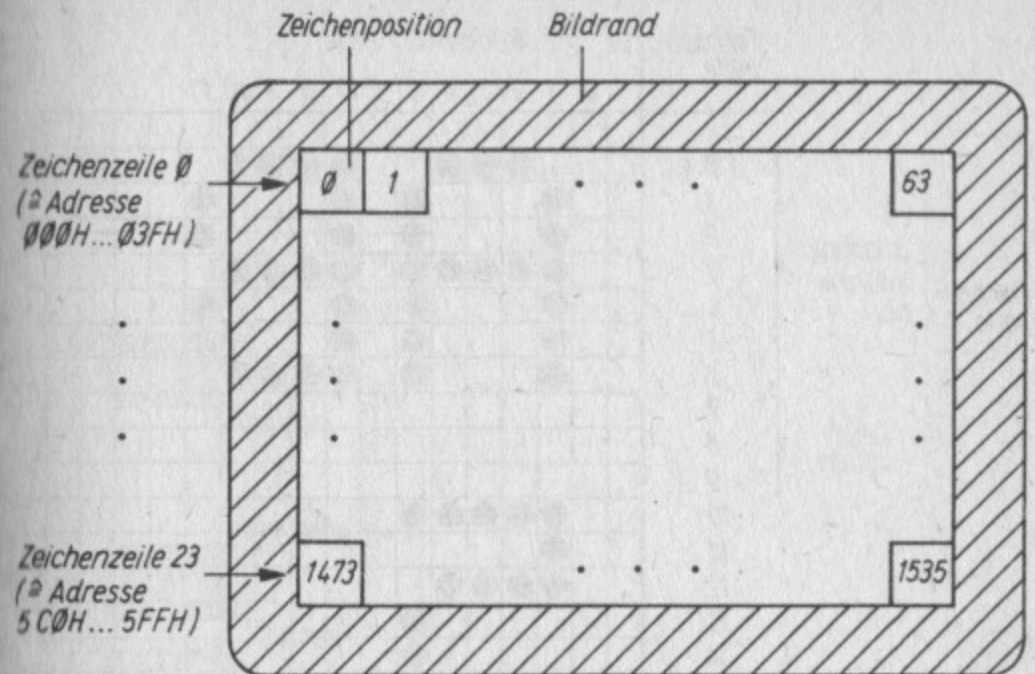


Bild 5.1 Bildschirmformat

licht die Darstellung einer relativ großen Anzahl von Zeichen auf dem Bildschirm (einschließlich des oberen und des unteren Bildrands).

Bild 5.1 zeigt das verwendete Bildschirmformat. Ein Bild besteht danach aus 24 Zeichenzeilen, jede Zeichenzeile aus 64 Zeichenpositionen. Damit ist die Darstellung von maximal 1536 Zeichen (d. h. 1,5 k) je Bild möglich.

Mit dieser Lösung wurde ein sinnvoller Kompromiß zwischen den technischen Möglichkeiten eines handelsüblichen TV-Geräts und der Forderung nach einer Abbildungsmöglichkeit für eine ausreichend große Zeichenzahl in gut lesbarer Form gefunden. Dabei nimmt man bewußt in Kauf, daß 0,5 kByte des 2-kByte-Bildwiederholerspeichers nicht ausgenutzt sind.

Jeder Zeichenposition auf dem Bildschirm ist eine Adresse im Bildwiederholerspeicher in steigender Reihenfolge fest zugeordnet. Der Bildschirmposition 0 entspricht also die Anfangsadresse CRTBG, der Position 1535 die Adresse CRTBG + 1535 des Bildwiederholerspeichers.

Ein Zeichen wird auf dem Bildschirm nach dem häufig benutzten Prinzip der 5×7 -Punktmatrix aufgebaut. Bild 5.2 zeigt die Zeichenzusammensetzung für das verwendete Bildschirminterface auf der Grundlage dieses Prinzips. Man benötigt zum Darstellen der

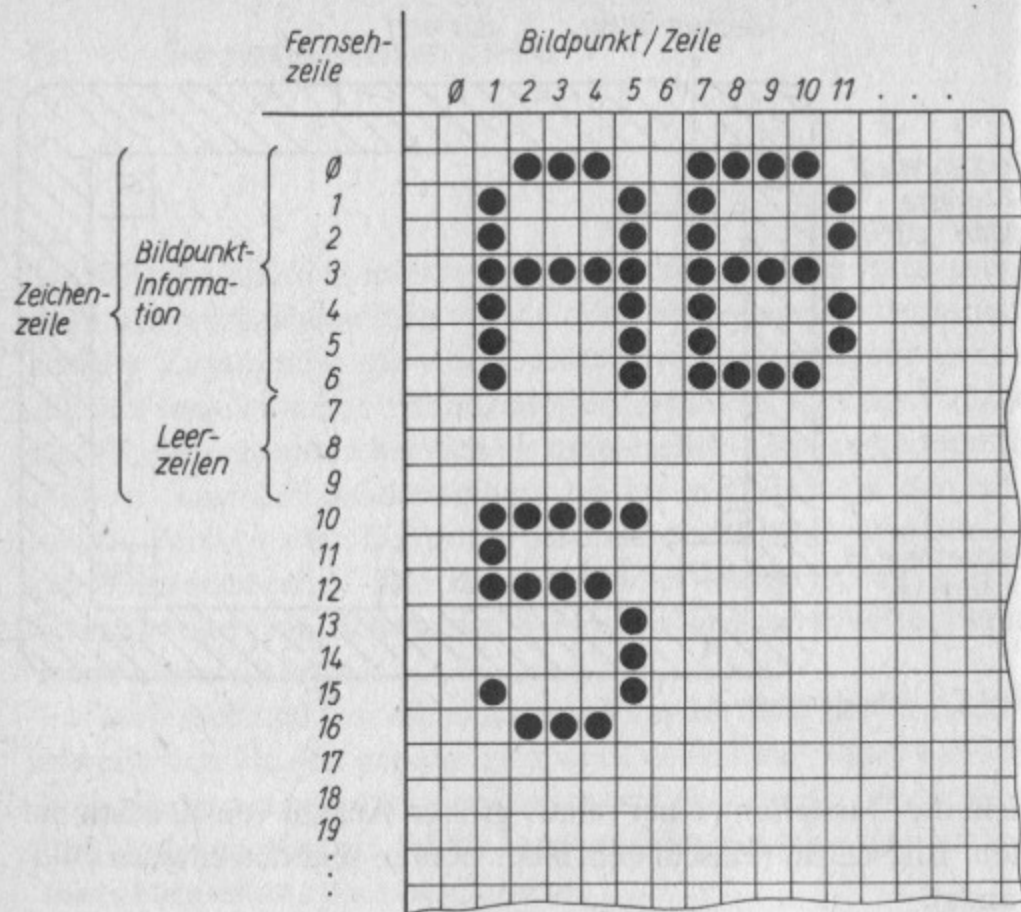


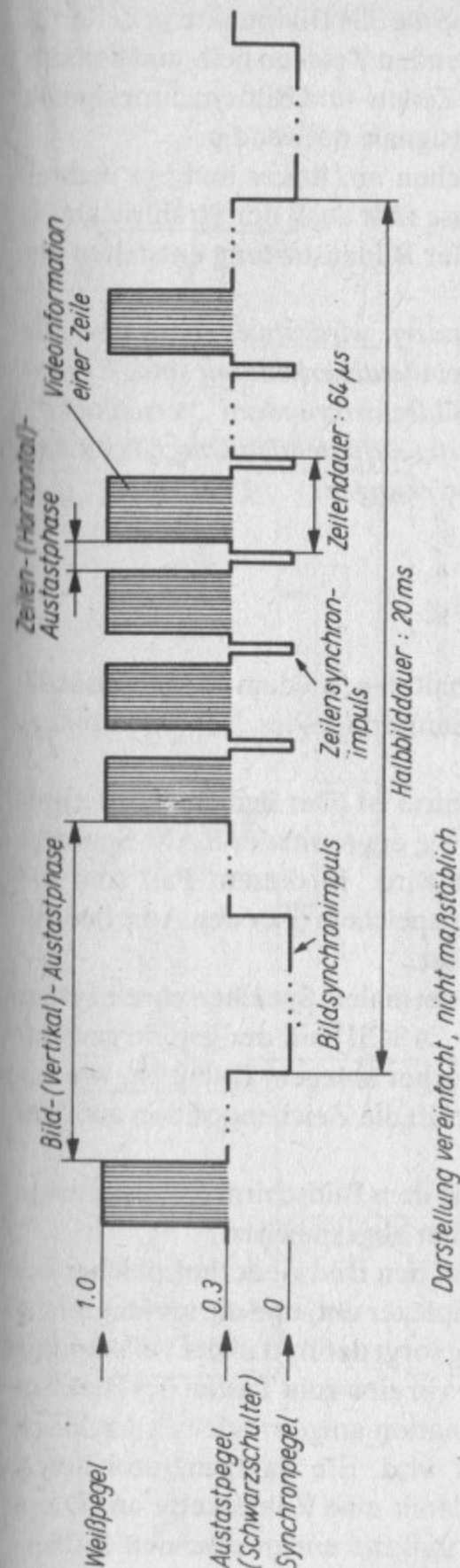
Bild 5.2 Zeichenaufbau nach dem Prinzip der 5×7 -Punktmatrix

Information einer Zeichenzeile also 7 Fernsehzeilen. Zur Gewährleistung eines übersichtlichen, gut lesbaren Bildes wird jede Zeichenzeile durch 3 weitere Fernsehzeilen von der benachbarten getrennt. Insgesamt besteht damit eine Zeichenzeile aus 10 Fernsehzeilen.

Für jedes Einzelzeichen stehen in der Horizontalen, d. h. je Fernsehzeile, 6 Bildpunkte zur Verfügung. Jeweils der 1. oder 6. Bildpunkt (abhängig von der gewählten Bitposition auf dem Zeichengenerator) wird dunkelgetastet und bildet den horizontalen Zeichenabstand.

Unter Berücksichtigung von Bildschirmformat und Zeichenaufbau setzt sich das Bildfeld damit aus 240 Fernsehzeilen zusammen. Eine Zeile besteht aus 384 Bildpunkten.

Bild 5.3 zeigt das BAS-Signal (Bild-, Austast- und Synchronsignal) nach CCIR, das mit dem Bildschirminterface erzeugt werden muß, um ein handelsübliches TV-Gerät ansteuern zu können. Danach



Darstellung vereinfacht, nicht maßstäblich

Bild 5.3 BAS-Signal nach CCIR

ist neben der Videoinformation, die die 384 Bildpunkte je Zeile abhängig von dem jeweils darzustellenden Zeichen hell- und dunkeltastet, auch die Bereitstellung der Zeilen- und Bildsynchrosignale sowie der entsprechenden Austastsignale notwendig.

Durch das Zeilenaustasten entstehen ein linker und ein rechter Bildrand. In die Zeilenaustastphase fällt auch der Strahlrücklauf, der damit nicht sichtbar ist. Bei der Bildaustastung entstehen ein oberer und ein unterer Bildrand.

Durch die Festlegung von Bildrändern wird abgesichert, daß alle 24 Zeichenzeilen zu je 64 Zeichen eindeutig erkennbar sind. Es werden also keine Zeichen von den Bildschirmrändern „verschluckt“. Die an den Bildschirmrändern häufig auftretenden Unschärfen und Verzeichnungen haben keine Auswirkungen.

5.2. Schaltungsbeschreibung

Bild 5.4 zeigt einen Übersichtsschaltplan, an dem das grundsätzliche Funktionsprinzip des Fernsehinterface des Mikrocomputers erklärt werden soll.

Der Zugriff der CPU zum Bildschirm ist über den Bildwiederhol-speicher sichergestellt, da dieser wie ein normaler RAM-Speicher (siehe Abschnitt 4.2.) verwaltet wird. In diesem Fall sind die Adreßleitungen des Bildwiederholerspeichers über den Adreßmultiplexer auf den Systembus geschaltet.

Die CPU kann jetzt mit einem normalen Speicherschreibzyklus den Zeichencode (ISO-7-Bit bzw. ASCII) auf der gerade gewählten Adresse im Bildwiederholer Speicher ablegen. Damit ist, wie bereits in Abschnitt 5.1. erwähnt, auch die Zeichenposition auf dem Bildschirm festgelegt.

In dieser Art wird der gesamte auf dem Bildschirm darzustellende Bildinhalt im Bildwiederholer Speicher abgespeichert.

Nachdem der Zugriff der CPU auf den Bildwiederholer Speicher beendet ist, schaltet der Adreßmultiplexer um, und die umfangreiche Elektronik der Interfaceschaltung sorgt dafür, daß der vollständige Inhalt des Bildwiederholerspeichers in eine zum Tasten des Elektronenstrahls geeignete Videoinformation umgewandelt und zyklisch mit der Bildfrequenz wiederholt wird. Ein Taktgenerator liefert den Bildpunkt-takt und steuert damit eine Zählerkette an. Diese Zählerkette wurde so gestaltet, daß die entsprechenden Zähler-

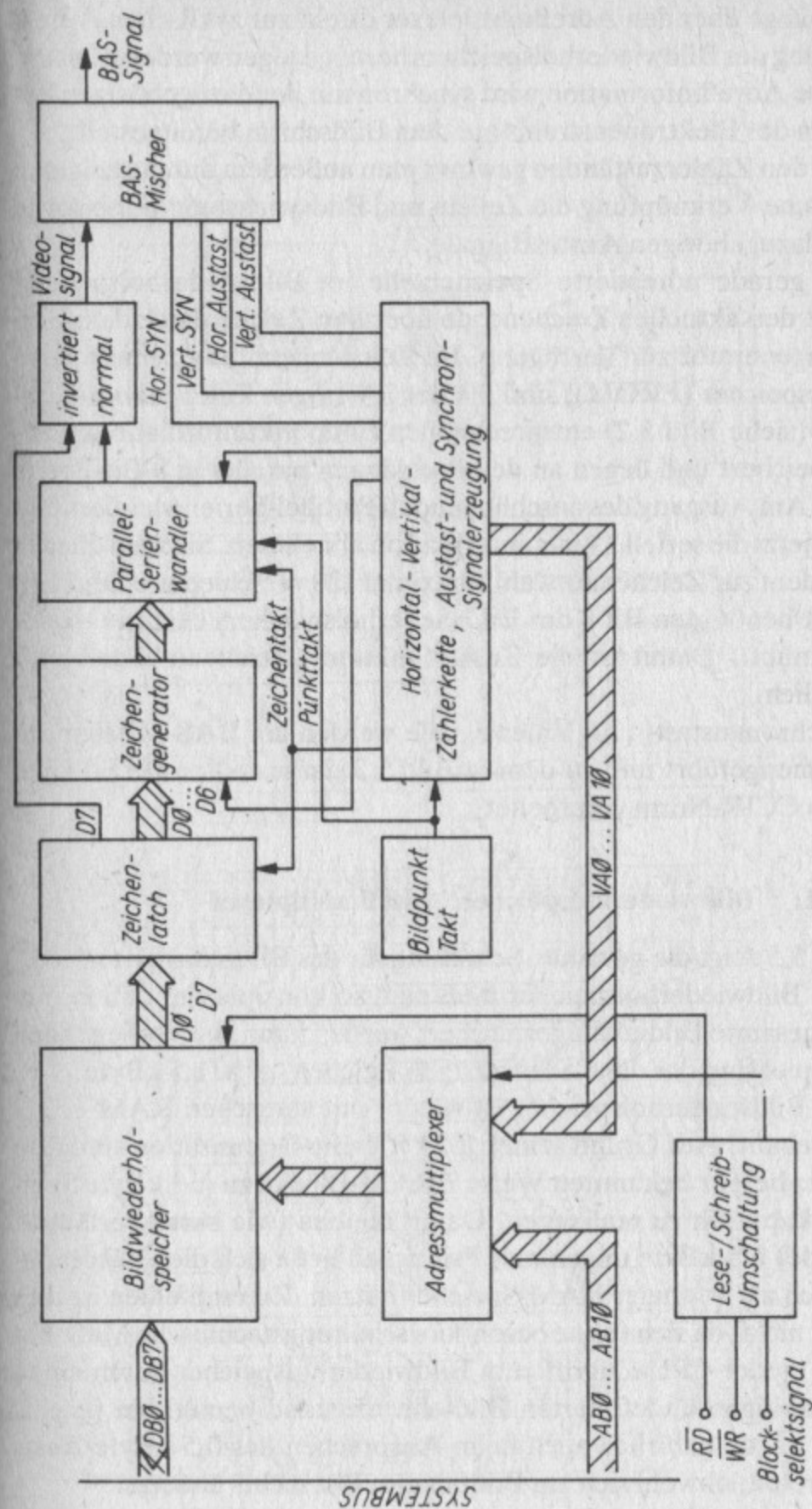


Bild 5.4 Übersichtsschaltplan des Fernsehinterface

ausgänge über den Adreßmultiplexer direkt zur zyklischen Adressierung des Bildwiederholers herangezogen werden können. Diese Adreßinformation wird synchron mit der dazugehörigen Position des Elektronenstrahls auf dem Bildschirm bereitgestellt.

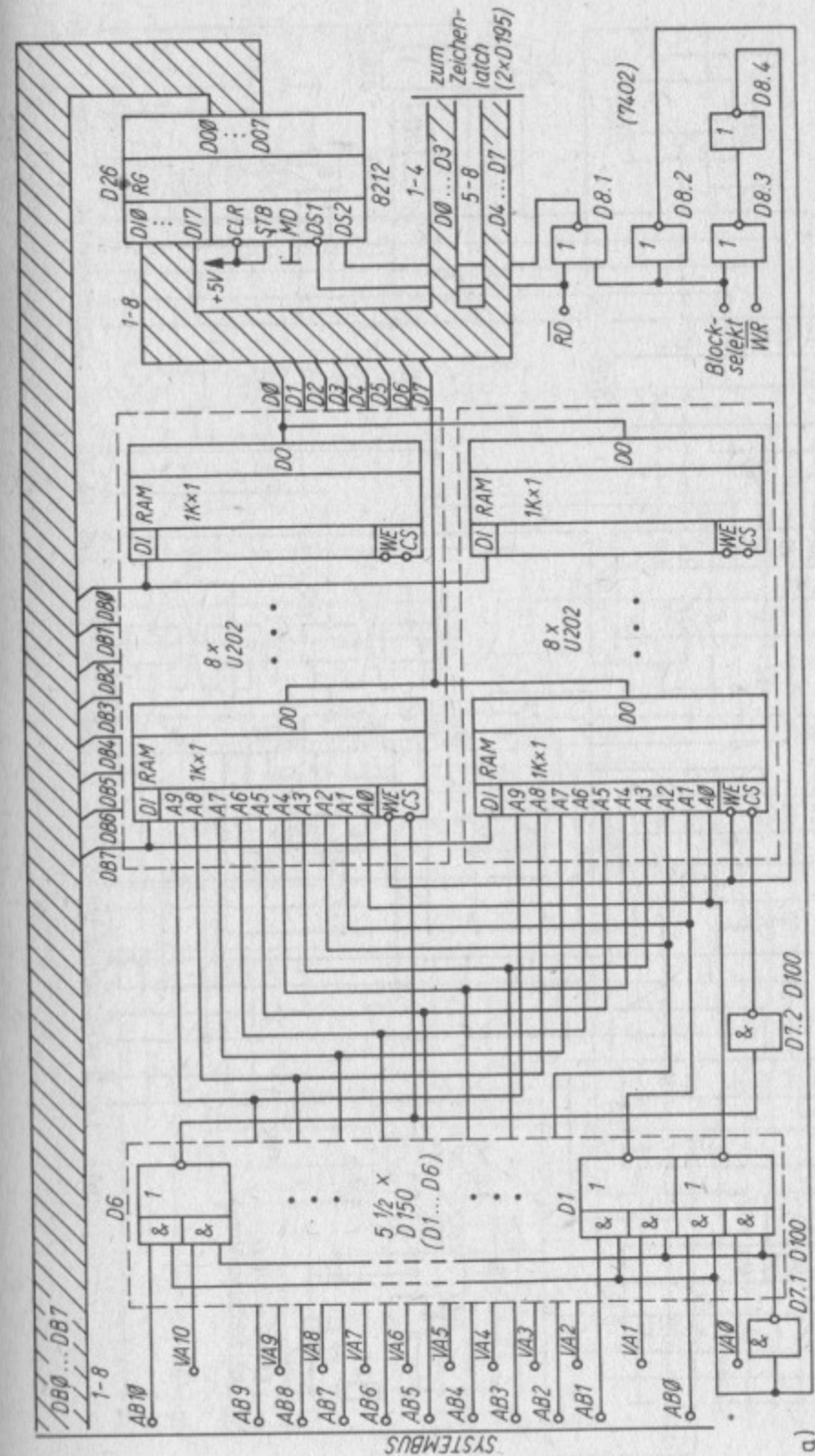
Aus den Zählerzuständen gewinnt man außerdem durch geeignete logische Verknüpfung die Zeilen- und Bildsynchronimpulse sowie die dazugehörigen Austastsignale.

Die gerade adressierte Speicherzelle im Bildwiederholer stellt den aktuellen Zeichencode über den Zeichenlatch dem Zeichengenerator zur Verfügung. Im Zeichengenerator, einem Festwertspeicher (PROM), sind die der jeweiligen Zeichenkonfiguration (siehe Bild 5.2) entsprechenden Bildpunktinformationen abgespeichert und liegen an den Ausgängen parallel in 8-Bit-Breite vor. Am Ausgang des anschließenden Parallel-Serienwandler läßt sich jetzt die serielle Videoinformation abnehmen. Sie wird danach mit dem zur Zeichenauswahl (maximal 128 verschiedene Zeichen) nicht benötigten Bit 8 des Bildwiederholers exklusiv – oder verknüpft. Damit ist die Zusatzfunktion „Zeicheninvertierung“ möglich.

Synchronaustast- und Videosignale werden am BAS-Mischer zusammengeführt und zu dem in Bild 5.3 dargestellten BAS-Signal nach CCIR-Norm verarbeitet.

5.2.1. Bildwiederholer, Adreßmultiplexer

Bild 5.5 zeigt die gewählte Schaltung für das Bildschirminterface. Den Bildwiederholer muß man so konzipieren, daß in ihm der gesamte Bildinhalt gespeichert werden kann. Im vorliegenden Beispiel sind das die Codes für 1536 Zeichen, d. h. 1,5 kByte. Der Bildwiederholer ist wieder mit statischen RAM U 202 aufgebaut. Auf Grund seiner 1024×1 -Bit-Organisation sind also in der bereits bekannten Weise 2 RAM-Blöcke zu je 1 kByte Speicherkapazität zu realisieren. Damit bleiben (wie bereits erläutert wurde) 0,5 kByte ungenutzt. Prinzipiell ließe sich dieser Rest natürlich als normaler RAM-Speicher nutzen. Zu empfehlen ist dies aber nicht, da sich (ohne besondere schaltungstechnische Maßnahmen) jeder CPU-Zugriff zum Bildwiederholer durch einen kurzzeitigen undefinierten Bildschirmzustand bemerkbar macht. Der würde natürlich auch beim Ansprechen des 0,5-kByte-Rests auftreten, obwohl sich am Bildinhalt selbst nichts änderte.



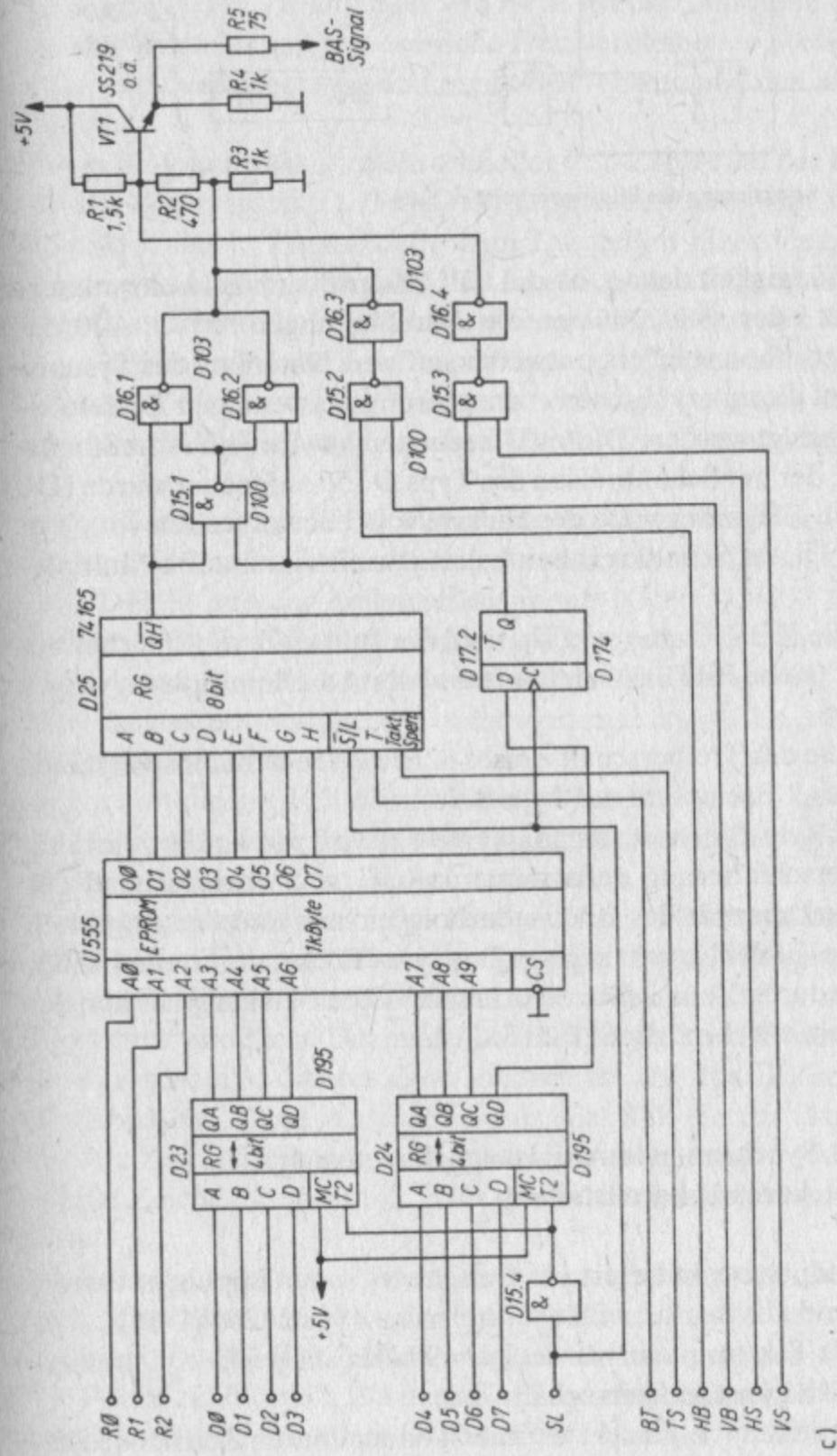
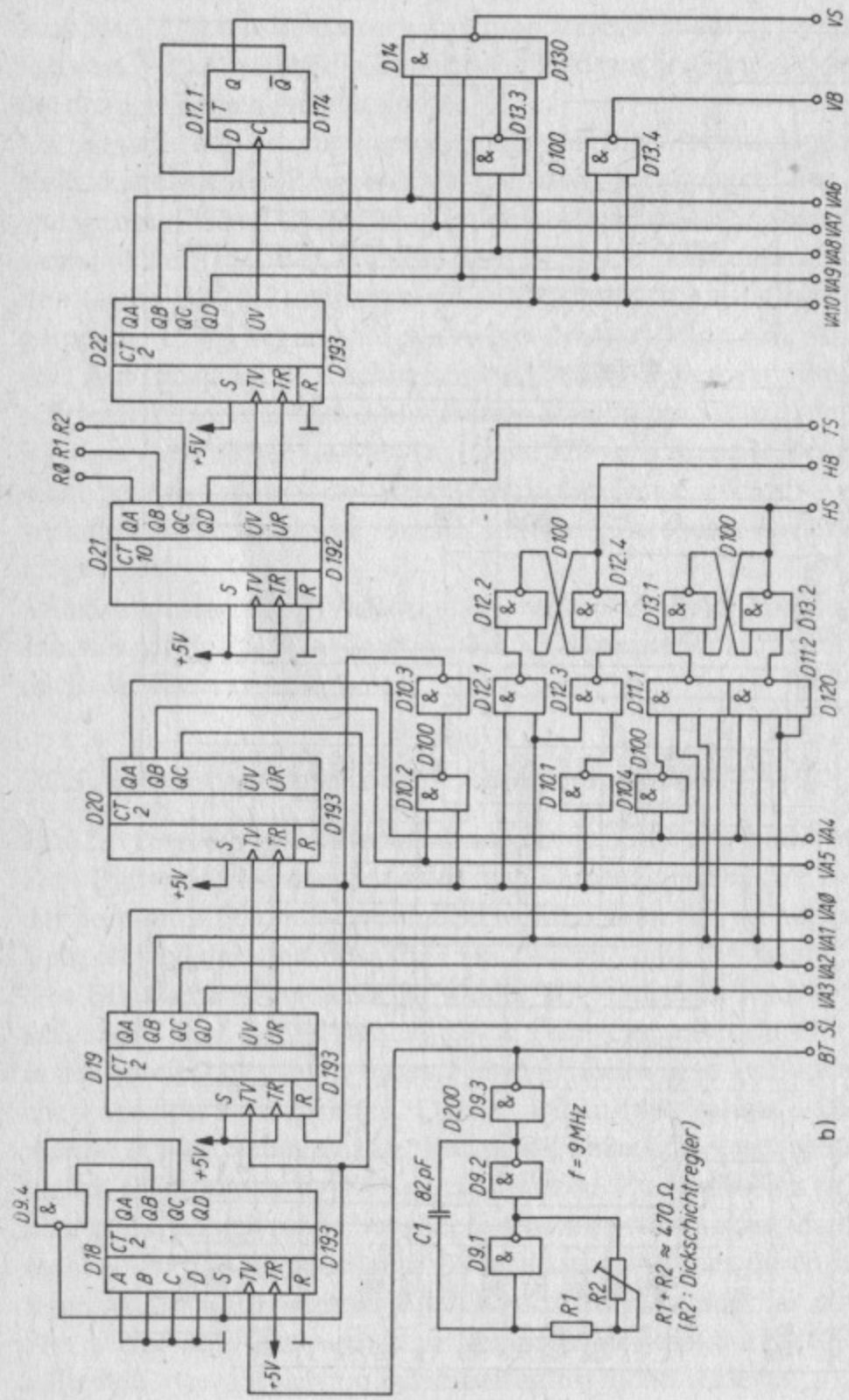


Bild 5.5 a – Bildwiederholer, b – Synchron- und Auslastungssteuerung, Adressbereitstellung, c – Steuergerät

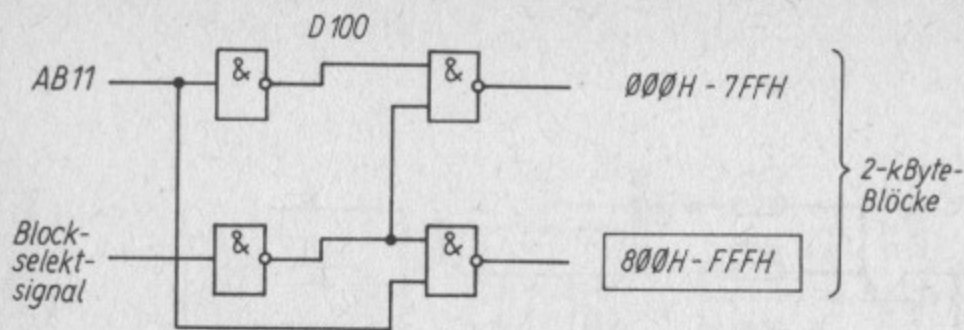


Bild 5.6 Selektierung des Bildwiederholerspeichers

In Abhängigkeit davon, ob die CPU Zugriff zum Bildschirminterface hat oder nicht, müssen die Adreßleitungen A0 ... A10 des Bildwiederholerspeichers entweder auf den Datenbus des Systems oder auf die zur zyklischen Adressierung notwendigen Zählerausgänge gelegt werden. Dieses Umschalten bewirkt ein Adreßmultiplexer, der mit Schaltkreisen des Typs *D 150* aufgebaut wurde (D1 ... D6). Eleganter wäre der Einsatz von 3 Schaltkreisen vom Typ *74157*. Dieser Schaltkreis beinhaltet jeweils vier 2-auf-1-Multiplexer [3].

Über die NOR-Gatter von D6 wird der Bildwiederholerspeicher bei *E800H* (siehe Bild 3.3) selektiert und der Adreßmultiplexer umgeschaltet.

An Stelle des Treiberschaltkreises *8212* eignen sich selbstverständlich auch 2 Exemplare des Typs *8216*.

Das 4-kByte-Blockselekt-signal (*E000H*) wird wieder der zentralen Speicherselektierung entnommen, wobei man entsprechend der Speicherkapazität des Bildwiederholerspeichers und der gewählten Adresse *E800H* noch in zwei 2-kByte-Blöcke unterteilen muß. Dies ist durch 1-aus-2-Decodierung des Blockselekt-signals und des Adreßbit *AB11* möglich (Bild 5.6).

5.2.2. Synchron- und Austast-signal-erzeugung, Adressenbereitstellung

Den Bildpunkt-takt liefert ein freischwingender Impuls-generator, bestehend aus den schnellen Gattern des Typs *D 200* (D 9.1 ... D 9.3). Als Taktfrequenz wurde $f_{BP} = 9 \text{ MHz}$ ausgewählt. Damit ergeben sich günstige Zeitverhältnisse.

Bildpunkt-dauer: $t_{BP} = 111 \text{ ns}$; mit 64 darstellbaren Zeichen je Zei-

chenzeile und 6 Bildpunkten je Zeichen und Fernsehzeile erhält man 384 Bildpunkte je Fernsehzeile. Daraus folgt eine genutzte Zeilendauer von $384 \cdot 111 \text{ ns} = 42,6 \mu\text{s}$. Das bedeutet also, daß $42,6 \mu\text{s}$ der verfügbaren Zeilendauer von $64 \mu\text{s}$ für das Schreiben einer Zeile des Bildfelds genutzt werden. Der verbleibende Rest von $21,4 \mu\text{s}$ wird dunkel getastet und ergibt den rechten und den linken Bildrand.

Mit dem Bildpunkt-takt wird ein schneller 6 : 1-Zähler auf der Basis des 4-Bit-Binärzählers *D 193* (D18) angesteuert. Er zählt die 6 Bildpunkte, die je Fernsehzeile zum Darstellen eines Zeichens notwendig sind, und gibt danach den Impuls an den folgenden Zeichenpositions-zähler ab. Dieser besteht aus 2 Schaltkreisen *D 193* (D19, D20) und zählt die Zeichenpositionen je Zeichenzeile bis 64. Die Gatter *D10.2* und *D10.3* übernehmen das Rücksetzen dieses Zählers nach 96 Eingangsimpulsen (d. h. 576 Bildpunktimpulsen).

Aus den Zählerzuständen des Zeichenpositions-zählers werden in geeigneter Form der Zeilensynchronimpuls (Gatter *D10.4*, *D11*, *D13.1*, *D13.2*) und der Zeilenaustastimpuls (Gatter *D12*, *D10.1*) decodiert. Für den Zeilensynchronimpuls ergeben sich eine Impulsbreite von $t_{IH} = 8 \mu\text{s}$ und die geforderte Periode von $T_H = 64 \mu\text{s}$. Die Austastzeit beträgt, wie bereits gezeigt, $t_{AH} = 21,4 \mu\text{s}$. Bild 5.7 zeigt die zeitlichen Abläufe.

Man erkennt die um 2 Zeichenpositionen (12 Bildpunkte entsprechend $1,33 \mu\text{s}$) verzögerte Helltastung. Damit werden keine undefinierten Zeichen auf dem Bildschirm sichtbar.

Der Ausgangsimpuls der bisher beschriebenen Horizontalzählerkette, d. h. der Zeilensynchronimpuls, gelangt auf den Eingang der sich anschließenden Vertikalzählerkette. Der am Beginn dieser Zählerkette befindliche Dezimalzähler *D 192* (D21) erhält also mit jeder Fernsehzeile, die der Elektronenstrahl auf den Bildschirm geschrieben hat, einen Ansteuerimpuls. Er zählt die zur Darstellung einer Zeichenzeile notwendigen 10 Fernsehzeilen (7 für die eigentliche Zeicheninformation, 3 für den vertikalen Zeichenzeilenabstand).

Nach Beendigung dieses Vorgangs wird ein Impuls an den nachfolgenden Zeichenzeilen-zähler abgegeben. Dieser besteht aus dem Schaltkreis *D 193* (D22) sowie einem Binäruntersetzer *D 174* (D17.1) und zählt bis 32. Da nach 32 Eingangsimpulsen der Zählvorgang wieder bei 0 beginnt, ist kein Rücksetzen erforderlich.

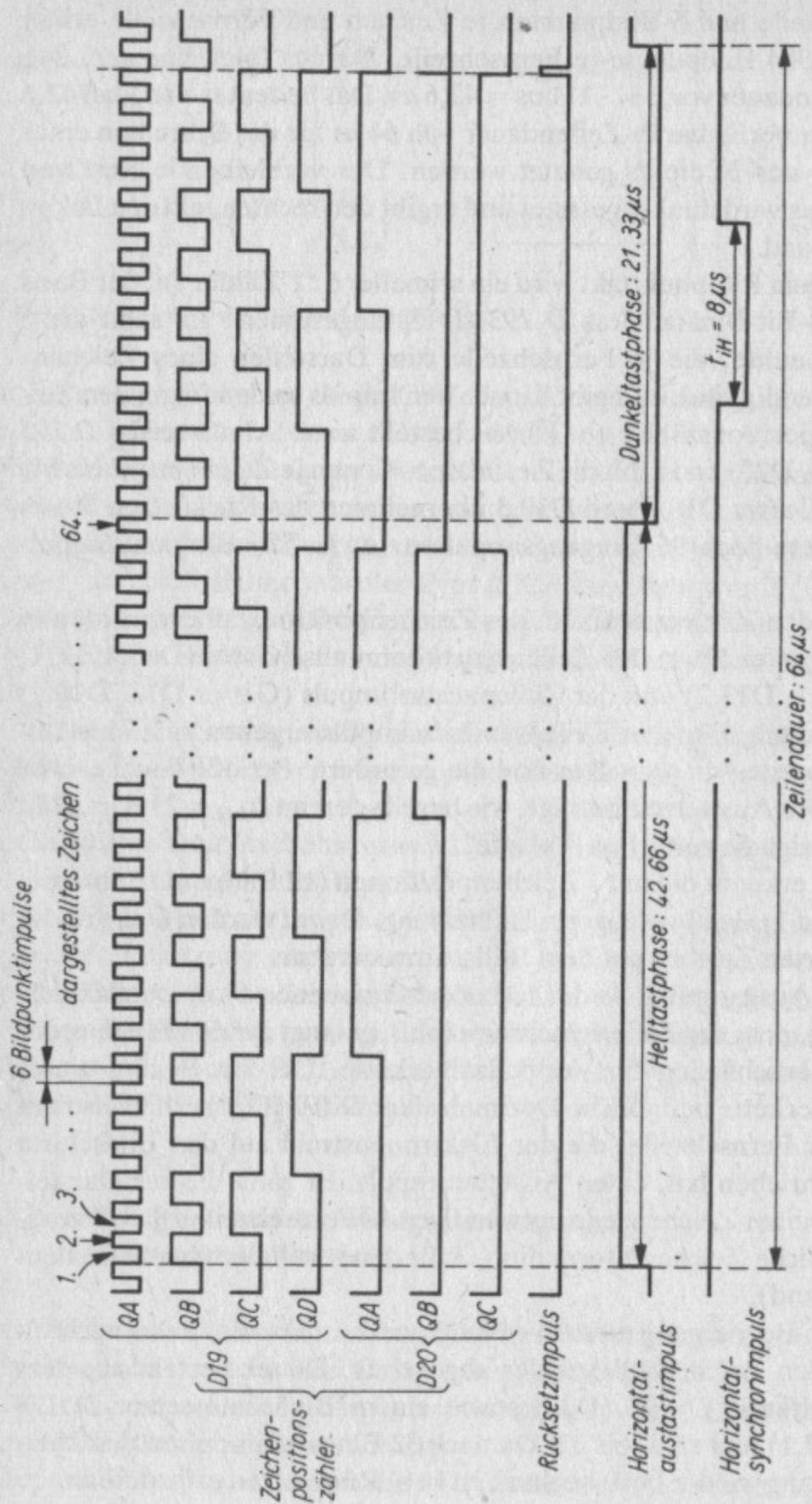


Bild 5.7 Impulsdiagramm der Horizontalablenkung

Das gesamte Bild besteht damit aus 32 Zeichenzeilen, entsprechend also 320 Fernsehzeilen. Davon entfallen 24 Zeichenzeilen auf das Bildfeld, die verbleibenden 8 (d. h. 80 Fernsehzeilen) bilden den oberen und den unteren Bildrand. Es ergibt sich daraus eine Bildfrequenz von

$$\frac{1}{320 \cdot 64 \mu\text{s}} = 48,8 \text{ Hz.}$$

Damit ist eine ausreichende Flimmerfreiheit des Bildes gewährleistet.

Aus den Zählerzuständen des Zeichenzeilenzählers werden mit der Gatterkombination D13.3, D14 der Vertikalsynchronimpuls ($t_{IV} = 640 \mu\text{s}$) sowie der Vertikalaustastimpuls ($t_{AV} = 80 \cdot 64 \mu\text{s} = 5,12 \text{ ms}$) mit dem Gatter D13.4 decodiert.

Bild 5.8 zeigt die Zeitverhältnisse für die Vertikalimpulserzeugung.

Über die zyklische Adressierung durch die entsprechenden Zählerausgänge werden dem Bildwiederholtspeicher die *Zeichenposition* ($0 \dots 63$) und die *Zeichenzeile* ($0 \dots 23$) signalisiert, in der sich der Elektronenstrahl gerade befindet. Genau aus diesem Sachverhalt ergibt sich die eindeutige Zuordnung zwischen Zeichenpositionen auf dem Bildschirm und Adresse im Bildwiederholtspeicher. Die Ausgänge QA, QB, QC des Dezimalzählers D 192 sind mit den Adreßeingängen A0, A1, A2 des im weiteren noch zu beschreibenden Zeichengenerators verbunden. Damit wird dem Zeichengenerator mitgeteilt, in welcher Fernsehzeile des gerade darzustellenden Zeichens sich der Elektronenstrahl befindet. Der Ausgang QD des D 192 sorgt über den Taktsperreingang des Parallel-Serienwandlers 74165 dafür, daß in den letzten 2 Fernsehzeilen (d. h. während der Zählerzustände 8, 9) jeder Zeichenzeile keine Videoinformation ausgegeben wird. Diese beiden sowie eine weitere, auf dem Zeichengenerator erzeugte, Leerzeile trennen die Zeichenzeilen in der Vertikalen.

5.2.3. Videosignalerzeugung

Der aktuelle Zeichencode, d. h. der Inhalt der gerade im Bildwiederholtspeicher adressierten Speicherzelle, wird mit jedem Ausgangsimpuls des Bildpunktzählers in den Zeichenlatch übernommen und steht gleichzeitig an dessen Ausgang zur Verfügung.

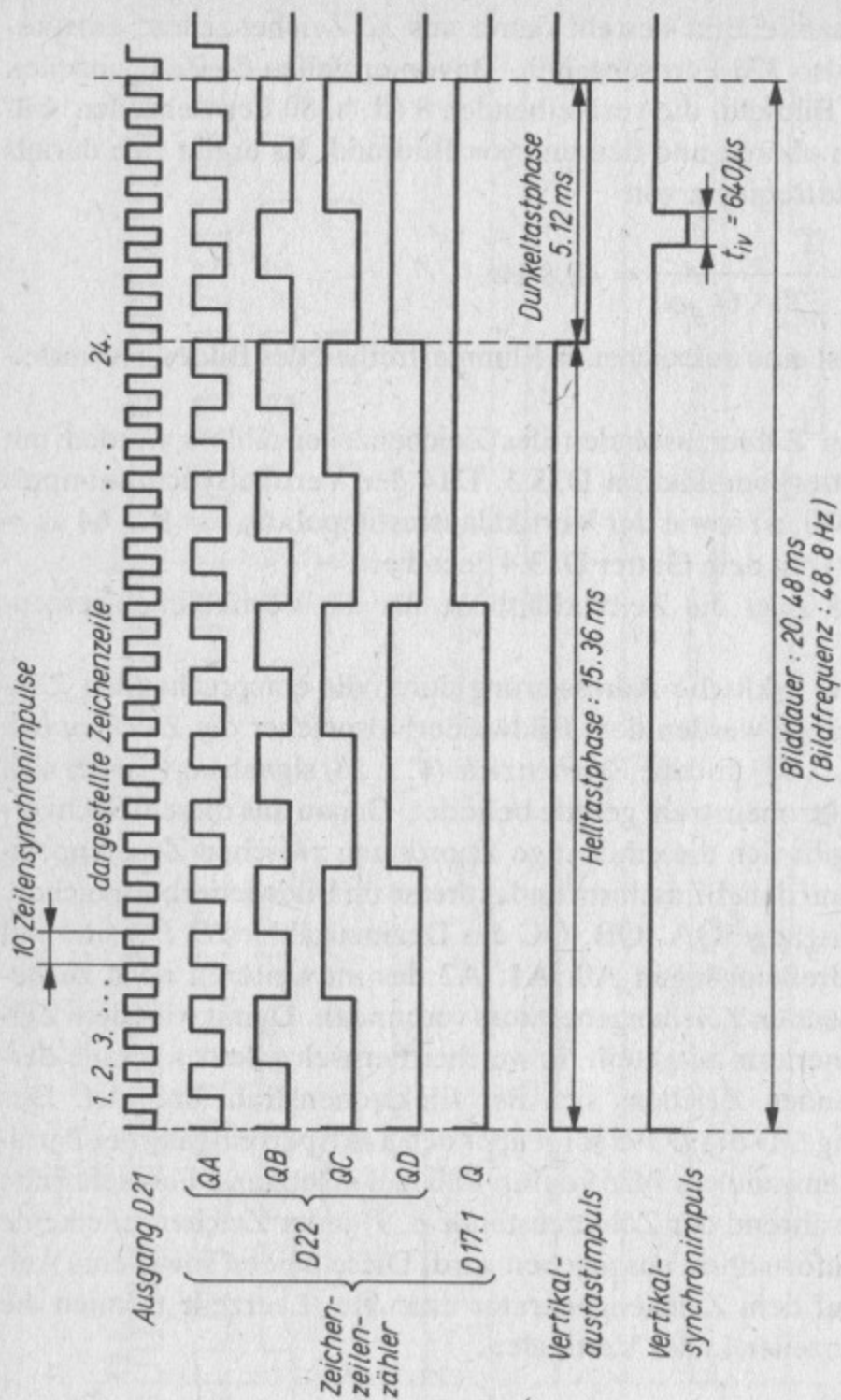


Bild 5.8 Impulsdiagramm der Vertikalablenkung

Der Zeichenlatch wurde mit zwei 4-Bit-Schieberegistern des Typs *D 195* aufgebaut. Er stellt dem Zeichengenerator unter Berücksichtigung der Speicherzugriffszeiten den Zeichencode als stabile Adresse zur Verfügung.

Der Zeichencode bildet also über die Adreßeingänge *A3 ... A9* eine von 128 möglichen Adressen für den Zeichengenerator-PROM. Auf genau dieser Adresse sind die Bildpunktinformationen für die 7 Zeilen des darzustellenden Zeichens in 7 aufeinanderfolgenden Speicherzellen als Datenbyte abgespeichert. Die 8. Speicherzelle in dieser Reihenfolge ist mit *00H* belegt und bildet damit die 1. der 3 Leerzeilen für den vertikalen Zeichenzeilenabstand.

Es werden nur 6 Bit der 8-Bit-Speicherbreite des PROM genutzt (5 Bit Bildpunktinformation, 1 Bit horizontaler Zeichenabstand). Insgesamt benötigt man damit zum Darstellen eines Zeichens 8 aufeinanderfolgende Speicherzellen des Zeichengenerator-PROM. Die 8 Speicherzellen werden, wie in Abschnitt 5.2.2. bereits erwähnt, durch die Zählerausgänge *QA, QB, QC* des Dezimalzählers *D 192* an den Adreßeingängen *A0, A1, A2* des Zeichengenerators adressiert.

Tabelle 5.1. enthält die Bildpunktinformation für den Buchstaben

Tabelle 5.1. Zuordnung: Zeichencode – Speicherzelle – Bildpunktinformation auf dem Zeichengenerator-PROM

Zeichencode (A3 ... A9)	PROM-Speicherzelle (A0 ... A9)	Zeichen							Datenbytes	
		D7	...	D0	nicht belegt					
41H	208H			•	•	•			38H	Bildpunktinformation
	209H	•					•		44H	
	20AH	•							44H	
	20BH	•	•	•	•	•			7CH	
	20CH	•						•	44H	
	20DH	•							44H	
	20EH	•						•	44H	
	20FH								00H	Leerzeile

Beispiel: Adressbitbelegung Zeichengenerator - PROM

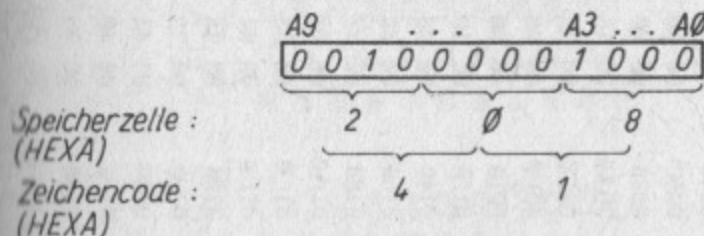


Tabelle 5.2. HEXA-Tabelle Zeichengenerator

0000H/03FFH

ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
0000	00	08	18	38	78	38	18	08	00	28	28	28	28	28	28	28	0010	00	7C	00	44	28	10	28	44	00	00	00	00	7E	00	00		
0020	00	7C	00	58	64	44	44	44	00	00	00	34	48	48	48	34	0030	00	30	48	48	58	44	58	40	00	18	20	10	38	44	44	38	
0040	00	00	00	00	10	28	44	7C	00	00	40	70	48	48	08	04	0050	00	38	44	44	7C	44	44	38	00	00	40	20	10	28	44	44	
0060	00	00	00	24	24	24	38	40	00	04	38	68	28	28	28		0070	00	00	00	3C	48	48	48	30	00	7C	20	10	08	10	20	7C	
0080	00	00	00	04	38	50	10	10	00	10	38	54	54	38	10		0090	00	38	44	44	44	44	28	60	00	10	00	38	44	7C	44	44	
00A0	00	10	00	38	48	48	48	34	00	28	00	38	44	7C	44	44	00B0	00	28	00	38	48	48	34	00	28	38	44	44	44	44	44	38	
00C0	00	28	00	38	44	44	44	38	00	28	00	44	44	44	38		00D0	00	00	28	00	44	44	44	38	00	00	10	08	7C	08	10	00	
00E0	00	00	3C	20	20	20	60	20	00	30	48	10	20	78	00	00	00F0	00	18	24	20	70	20	20	7C	00	44	28	10	10	38	10	10	
0010	00	00	00	00	00	00	00	00	00	20	20	20	20	20	00	20	0110	00	28	28	00	00	00	00	00	00	28	28	7C	28	7C	28	28	28
0120	00	10	3C	50	38	14	78	10	00	60	64	08	10	20	4C	0C	0130	00	20	50	20	54	48	34	00	30	30	10	10	20	00	00	00	
0140	00	08	10	10	10	10	10	08	00	20	10	10	10	10	10	20	0150	00	00	10	54	38	54	10	00	00	00	10	10	7C	10	10	00	00
0160	00	00	00	00	30	30	10	20	00	00	00	00	7C	00	00	00	0170	00	00	00	00	30	30	00	00	00	00	04	08	10	20	40	00	00
0180	00	38	44	4C	54	64	44	38	00	10	30	10	10	10	38		0190	00	38	44	04	18	20	40	7C	00	38	44	04	18	04	44	38	
01A0	00	08	18	28	48	7C	08	08	00	7C	40	78	04	04	44	38	01B0	00	18	20	40	78	44	44	38	00	7C	04	08	10	20	20	20	
01C0	00	38	44	44	38	44	44	38	00	38	44	44	3C	04	08	30	01D0	00	00	30	30	00	30	00	00	00	30	30	00	30	30	10	20	20
01E0	00	04	08	10	20	10	08	04	00	00	00	7C	00	7C	00	00	01F0	00	40	20	10	08	10	20	40	00	38	44	04	08	10	00	10	00
0200	00	38	44	5C	54	5C	40	38	00	38	44	44	7C	44	44	44	0210	00	78	44	44	78	44	44	78	00	38	44	40	40	40	44	38	
0220	00	78	44	44	44	44	44	78	00	7C	40	40	78	40	40	7C	0230	00	7C	40	40	78	40	40	40	00	38	44	40	40	4C	44	3C	3C
0240	00	44	44	44	7C	44	44	44	00	38	10	10	10	10	38		0250	00	04	04	04	04	04	44	38	00	44	48	50	60	50	48	44	44
0260	00	40	40	40	40	40	40	7C	00	44	6C	54	54	44	44	44	0270	00	44	44	64	54	4C	44	44	00	38	44	44	44	44	44	44	38
0280	00	78	44	44	78	40	40	40	00	38	44	44	44	54	48	34	0290	00	78	44	44	78	50	48	44	00	38	44	40	38	04	44	38	
02A0	00	7C	10	10	10	10	10	10	10	00	44	44	44	44	44	38	02B0	00	44	44	44	28	28	10	10	00	44	44	44	44	54	54	6C	6C

Tabelle 5.2. HEXA-Tabelle Zeichengenerator

0000H/03FFH

ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
02C0	00	44	44	28	10	28	44	44	00	44	44	28	10	10	10	10	02D0	00	7C	04	08	10	20	40	7C	00	1C	10	10	10	10	10	1C	
02E0	00	00	40	20	10	08	04	02	00	70	10	10	10	10	70		02F0	00	10	38	54	10	10	10	10	00	00	00	00	00	00	00	7C	
0300	00	30	30	20	10	00	00	00	00	00	00	38	48	48	34		0310	00	40	40	58	64	44	44	78	00	00	38	40	40	44	38		
0320	00	04	04	34	4C	44	44	3C	00	00	00	38	44	78	40	38	0330	00	10	28	20	70	20	20	20	00	00	00	3C	44	3C	04	18	
0340	00	40	40	58	64	44	44	44	00	10	00	30	10	10	38		0350	00	08	00	18	08	08	48	30	00	20	24	28	30	28	24	24	
0360	00	30	10	10	10	10	10	38	00	00	00	28	54	54	44	44	0370	00	00	00	58	64	44	44	44	00	00	38	44	44	44	44	38	
0380	00	00	00	78	44	64	58	40	00	00	00	38	48	58	28	0C	0390	00	00	00	2C	30	20	20	20	00	00	38	40	38	04	78		
03A0	00	00	20	70	20	20	28	10	00	00	00	44	44	44	4C	34	03B0	00	00	00	44	44	44	28	10	00	00	44	44	54	54	28	28	
03C0	00	00	00	44	28	10	28	44	00	00	00	44	28	10	10	20	03D0	00	00	00	7C	08	10	20	7C	00	08	10	10	20	10	10	08	
03E0	00	10	10	10	00	10	10	10	00	20	10	10	08	10	10	20	03F0	00	00	00	20	54	08	00	00	00	28	54	28	54	28	54	28	28

A sowie die entsprechenden Datenbyte, die auf dem Zeichengenerator abzuspeichern sind. Auch die Zuordnung zwischen Speicherzellenadressierung (A0 ... A9) und Zeichencode (A3 ... A9) ist aus der Tabelle ersichtlich.

Als Zeichengenerator wird der 1-kByte-PROM *U 555* verwendet. Über dessen Adreßeingänge A3 ... A9 lassen sich also 128 Zeichen zu je 8 aufeinanderfolgenden Speicherzellen (das entspricht genau 1 kByte) auswählen.

Für die notwendigen Zeichencodereinigungen gilt der ISO-7-Bit- bzw. ASCII-Code (siehe Tabellen im Anhang). Die Auswahl der 128 Zeichen und das Programmieren des Zeichengenerators-PROM sind unter Berücksichtigung der gegebenen Hinweise (Zeichenaufbau, Adreßbelegung, Codereinigung) nach eigenem Geschmack möglich. Beispielsweise können selbst zu gestaltende Pseudographikelemente mit vorgesehen werden.

In Tabelle 5.2. ist ein bewährter Zeichengenerator in Form einer Hexatabelle angegeben. Er enthält neben den Ziffern, den großen und kleinen Buchstaben alle Umlaute, die gebräuchlichsten griechischen Buchstaben sowie alle aus der Sicht der Thematik relevanten Sonderzeichen.

Entsprechend der Adressierung durch den jeweiligen Zeichencode liegt also die aktuelle Bildpunktinformation einer Zeile des gerade darzustellenden Zeichens an den Ausgängen 00 ... 07 des Zeichengenerators *U 555* parallel vor. Der sich anschließende Parallel-Serienwandler wandelt diese Information in ein serielles Videosignal um, wie es zum Tasten des Elektronenstrahls benötigt wird. Zur Parallel-Serienwandlung wird das 8-Bit-Schieberegister *74165* verwendet [3]. Für die Übernahme der Parallelinformation vom Zeichengenerator (d. h. zum Laden des Schieberegisters) ist ein Nadelimpuls erforderlich, der in dem Moment an den Shift-Load-Eingang des *74165* geliefert wird, wenn der Elektronenstrahl die 6 Bildpunkte des vorhergehenden Zeichens in der jeweiligen Fernsehzeile auf den Bildschirm geschrieben hat.

Der benötigte Ladeimpuls steht am Ausgang des Bildpunktzählers zur Verfügung. Er wird durch Gatter D15.4 invertiert und zum Laden des Zeichenlatch *D 195* verwendet.

Die Videoinformation muß selbstverständlich synchron mit dem Bildpunkttakt seriell ausgegeben werden. Zu dem Zweck ist der Takteingang des Schaltkreises *74165* direkt mit dem Ausgang des Taktgenerators verbunden.

Die sinnvolle Verwendung des Taktsperreingangs am Schieberegister *74165* wurde bereits beschrieben.

Im Abschnitt 5.2. wurde darauf hingewiesen, daß das Bit 8 des Bildwiederholungspeichers zur Auswahl der 128 auf dem Zeichengenerator programmierten Zeichen nicht benötigt wird. Man realisiert deshalb damit die Zusatzfunktion „Zeicheninvertierung“, indem das Bit 8 nach Zwischenspeicherung im D-Latch *D 174* (D17.2) mit dem Videosignal exklusiv – oder verknüpft (Gatter D15.1, D16.1, D16.2) wird. Für den Fall, daß Bit 8 H-Pegel führt, ist eine invertierte Zeichendarstellung (dunkel auf hellem Grund) sichergestellt.

5.2.4. BAS-Signalerzeugung

Ein TV-Gerät läßt sich nur mit BAS-Signal nach CCIR gemäß Bild 5.3 ansteuern. Dieses erzeugt man, indem das Videosignal mit den vorher beschriebenen Synchron- und Austastsignalen zusammengeführt wird. Hierzu werden die Horizontal- und Vertikalsynchronsignale am Gatter D15.3 und die entsprechenden Austastsignale am Gatter D15.2 und verknüpft. An den Ausgängen der Open-Collektor-Gatter *D 103* (D16) liegen jetzt alle Signale in richtiger Polarität vor. Der BAS-Mischer, Transistor VT 1, mischt sie und liefert am Ausgang polaritäts- und potentialgerecht das BAS-Signal. Das BAS-Signal wird dem Video-Eingang des TV-Geräts zugeführt.

Für bestimmte Fernsehgeräte (z. B. *Combivision*) hat es sich als günstig erwiesen, Video- und Synchronsignale getrennt den entsprechenden Stufen im TV-Gerät zuzuführen.

Verwendet man einen geeigneten HF-Modulator, kann das BAS-Signal auch direkt über den Antenneneingang eingespeist werden. Hierbei sind die Störstrahlungsbestimmungen der Deutschen Post zu beachten!

5.3. Erweiterungsmöglichkeiten

Das Fernsehinterface kann durch die mögliche Darstellung graphischer Informationen (z. B. in Verbindung mit den in Abschnitt 7. vorgestellten AD-Wandlern) auf dem Bildschirm sinnvoll erweitert werden. Dazu bieten sich mehrere Möglichkeiten.

5.3.1. Zeichengenerator mit Graphiksymbolen

Auf dem Zeichengenerator werden neben dem ASCII- bzw. ISO-7-Bit-Zeichensatz Graphikelemente (z. B. Blockgraphik) untergebracht. Diese Blockelemente gestaltet man so, daß sich aus ihnen die gewünschten graphischen Darstellungen auf dem Bildschirm zusammensetzen lassen. Als Zeichengenerator eignet sich ein PROM mit einer Speicherkapazität ≥ 2 kByte (z. B. 2716, 2732). Dabei wird der Ausgang QD des Zählers *D 192* (D21) vom Taktsperringang des Parallel-Serienwandlers 74165 (D25) abgetrennt und zum Adressieren des PROM herangezogen. Jetzt können alle 10 Fernsehzeilen einer Zeichenposition zur Informationsdarstellung genutzt werden (z. B. auch bei Buchstaben zum Erzeugen von Unterlängen).

5.3.2. RAM-Speicher als Zeichengenerator

Als Zeichengenerator wird ein RAM-Speicher verwendet. Die gewünschten Graphiksymbole (z. B. Kurvenabschnitt u. ä.) lassen sich jetzt nach Bedarf programmieren. Zu diesem Zweck werden die Datengänge der RAM (ggf. über Treiberschaltkreise) auf den Datenbus gelegt. Die Datenausgänge sind unmittelbar mit den Eingängen des Schieberegisters 74165 verbunden.

Die Adreßeingänge werden über 2-auf-1-Multiplexer bei CPU-Zugriff auf den Adreßbus und andernfalls auf die Latchausgänge *D 195* (D23, D24) bzw. die Zählerausgänge QA, QB, QC des *D 192* (D21) gelegt.

Den Zeichengenerator kann man auch in kombinierter RAM- bzw. PROM-Ausführung realisieren. Beispielsweise ließ sich der ASCII-Zeichensatz auf PROM abspeichern, und Graphikelemente in der oben geschilderten Art könnte man im RAM-Speicher programmieren.

Für den Taktsperringang des 74165 gilt das in Abschnitt 5.3.1. Gesagte.

5.3.3. Vollgraphik

In diesem Fall kann jeder auf dem Bildschirm darstellbare Bildpunkt softwaremäßig angesprochen werden. Der Zeichengenera-

tor entfällt. Als Bildwiederholtspeicher nutzt man einen RAM-Block mit entsprechend großer Speicherkapazität (z. B. 256×256 Bit). Hierzu wird entweder dem Fernsehinterface ein Bildwiederholtspeicher zugeordnet oder ein Teil des Arbeitsspeichers des Computers genutzt. Im letzteren Fall liegen die Eingänge des Parallel-Serienwandlers dann direkt am Datenbus. Die Videoadressen (d. h. die Zählerausgänge zur zyklischen Adressierung) werden über Bustreiber an den Adreßbus gelegt. Die für den Speicherzugriff nötigen Steuersignale muß eine entsprechende Logik erzeugen. Der DMA-Vorgang, d. h. das Auslesen des Bildwiederholtspeichers, liegt in der Vertikalhellstastphase. Das Vertikalaustastsignal ist gleichzeitig das $\overline{\text{BUSRQ}}$ -Signal (in der Darstellung gemäß Bild 5.8 in invertierter Form). Die CPU kann damit nur in der Vertikalaustastphase arbeiten.

Mit dem Signal $\text{BUSAK} = \text{L}$ werden die Adreßtreiber freigegeben und damit die aktuelle CRT-Adresse auf den Adreßbus gelegt. Verwendet man dynamische RAM, dann muß das Refresh aufrechterhalten werden.

5.4. Aufbau und Inbetriebnahme

Bei Inbetriebnahme des Bildschirminterface werden zunächst die RAM-Bestückung des Bildwiederholtspeichers und die CPU-Baugruppe ausgeklammert.

Man beginnt mit dem Einstellen des Taktgenerators auf 9 MHz. Damit wäre bereits der einzig notwendige Abgleichvorgang an der Schaltung erledigt. Hierzu noch ein Hinweis: Der Bildpunkttakt wird mit dem Oszilloskop zweckmäßig durch exaktes Einstellen der Periode des Zeilensynchronimpulses mit $T_H = 64 \mu\text{s}$ fein justiert.

Anschließend werden sämtliche Synchron- und Austastsignale und, wenn diese in Ordnung sind, das komplette BAS-Signal oszilloskopisch kontrolliert. Genügen diese Signale den angegebenen Bedingungen (vergleiche Bild 5.3, Bild 5.7, Bild 5.8), legt man die Multiplexerausgänge $A_0 \dots A_7$ auf die entsprechenden Eingänge des Zeichenlatch. Auf dem Bildschirm muß jetzt der komplette Zeichensatz abgebildet sein (normal und invertiert).

Nach positivem Abschluß dieser Kontrollvorgänge werden die Verbindungen Multiplexer – Zeichenlatch beseitigt und der Bild-

wiederholpeicher mit den RAM-Schaltkreisen bestückt. Im Anschluß hieran verbindet man die Bildschirmplatine über den Systembus mit der CPU-Baugruppe und hat jetzt die Möglichkeit, ein geeignetes Testprogramm zu starten.

Beispielsweise kann der Zeichensatz auf den Bildschirm geschrieben werden. Das kleine Programm dazu ist im folgenden angegeben:

```
LD HL, E800H
M1: LD (HL), L
    LD L, (HL)
    INC HL
    LD A, H
    CMP F0H
    JRNZ M1
    HALT
```

Das Bildschirminterface funktioniert dann vorschriftsmäßig, wenn der vollständige Zeichensatz 6mal in steigender Reihenfolge der Zeichen gemäß ISO-7-Bit bzw. ASCII auf dem Bildschirm geschrieben wird.

6. Alphanumerische Tastatur

Für den Dialog mit dem beschriebenen Mikrocomputer ist eine geeignete Tastatur die wichtigste Voraussetzung der Eingabe. Die im folgenden vorgestellte, einfach nachzubauende alphanumerische Tastatur läßt sich mit bis zu 64 Tasten auslegen. Im Gegensatz zu anderen denkbaren und auch üblichen Lösungen wird der jeweilige Tastaturcode hardwaremäßig erzeugt.

6.1. Schaltungsbeschreibung

Die Tastatur ist so konzipiert, daß beim Drücken einer Taste der entsprechende Tastaturcode auf dem Datenbus zur Verfügung gestellt wird. Dabei muß natürlich der Tastaturcode identisch mit dem auch im Bildschirminterface verwendeten Zeichencode gemäß ISO-7-Bit bzw. ASCII sein. Alle gewünschten Tastaturcodes werden auf einem PROM (U 555) gespeichert. Das Tastaturinterface sorgt nun dafür, daß beim Drücken einer Taste dem PROM gerade die Adresse bereitgestellt wird, auf der der dazugehörige Tastaturcode abgespeichert ist. Für die Meldung an das System über das Vorliegen eines gültigen Tastaturcodes steht ein Konsolstatussignal zur Verfügung.

Mit Hilfe eines Fertigmeldeimpulses kann man auch die softwaremäßige Übernahme des gültigen Tastaturcodes durch die CPU z. B. über nicht maskierbaren Interrupt ($\overline{\text{NMI}}$) einleiten.

Funktionsweise (Bild 6.1):

Ein Taktgenerator ($f \approx 1,5 \text{ KHz}$) steuert den 4-Bit-Binärzähler *D 193* an, dessen Ausgänge dem 1. Schaltkreis *D 195* (D7) eines insgesamt 6-Bit-Adreßlatch zugeführt werden. Die Zählerzustände des *D 193* (D6) bilden damit nach Übernahme in den Latch die 4 niederwärtigen Adreßbit $A_0 \dots A_3$ für den Tastatur-PROM. Gleichzeitig wird aus ihnen mit Hilfe des 1-aus-16-Decoders *MH 74154* das Spaltenauswahlsignal für die Kontaktmatrix der Tastatur gewonnen. Die Decoderausgänge werden dabei durch die Dioden *VD1 \dots VD16* geschützt.

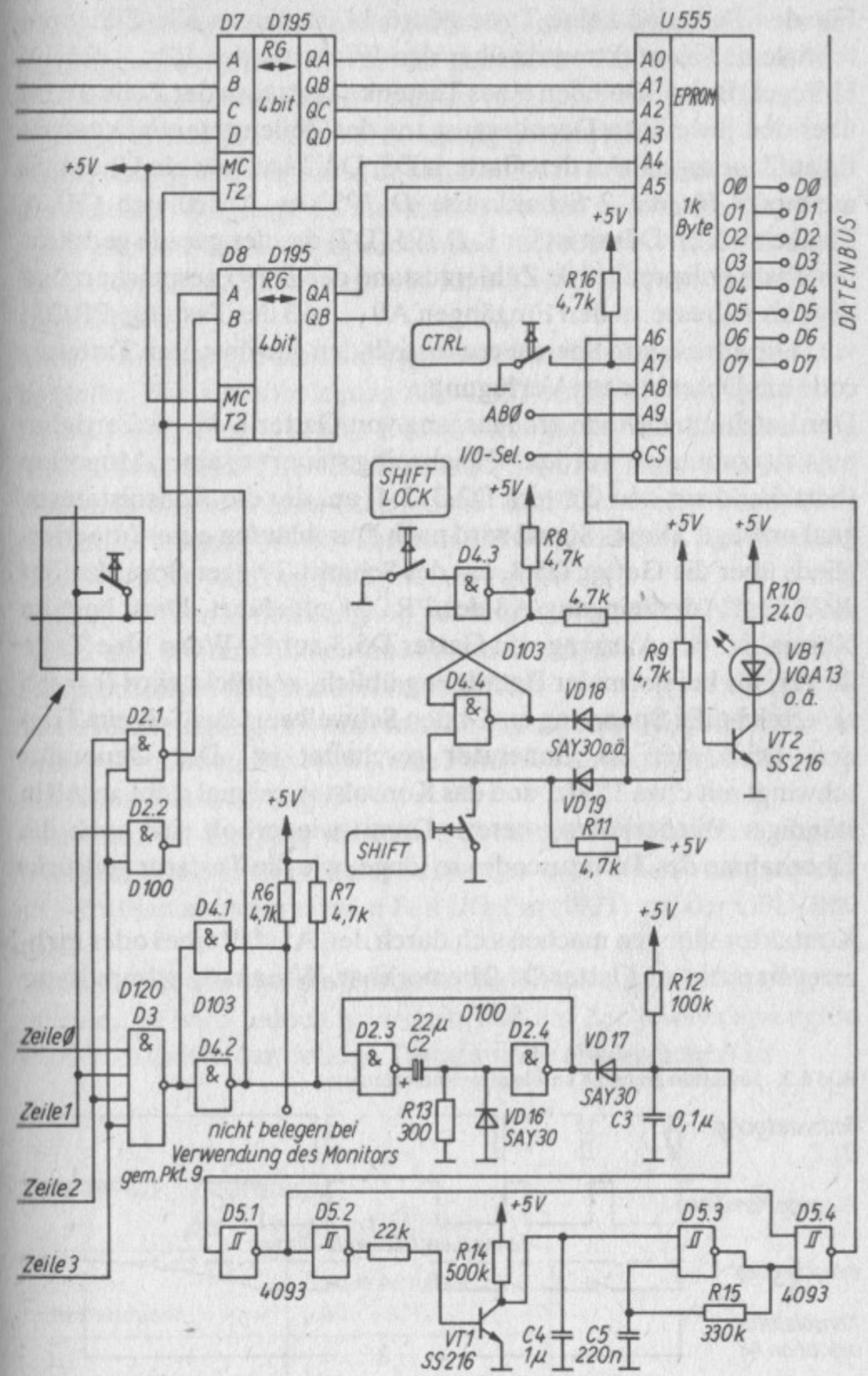
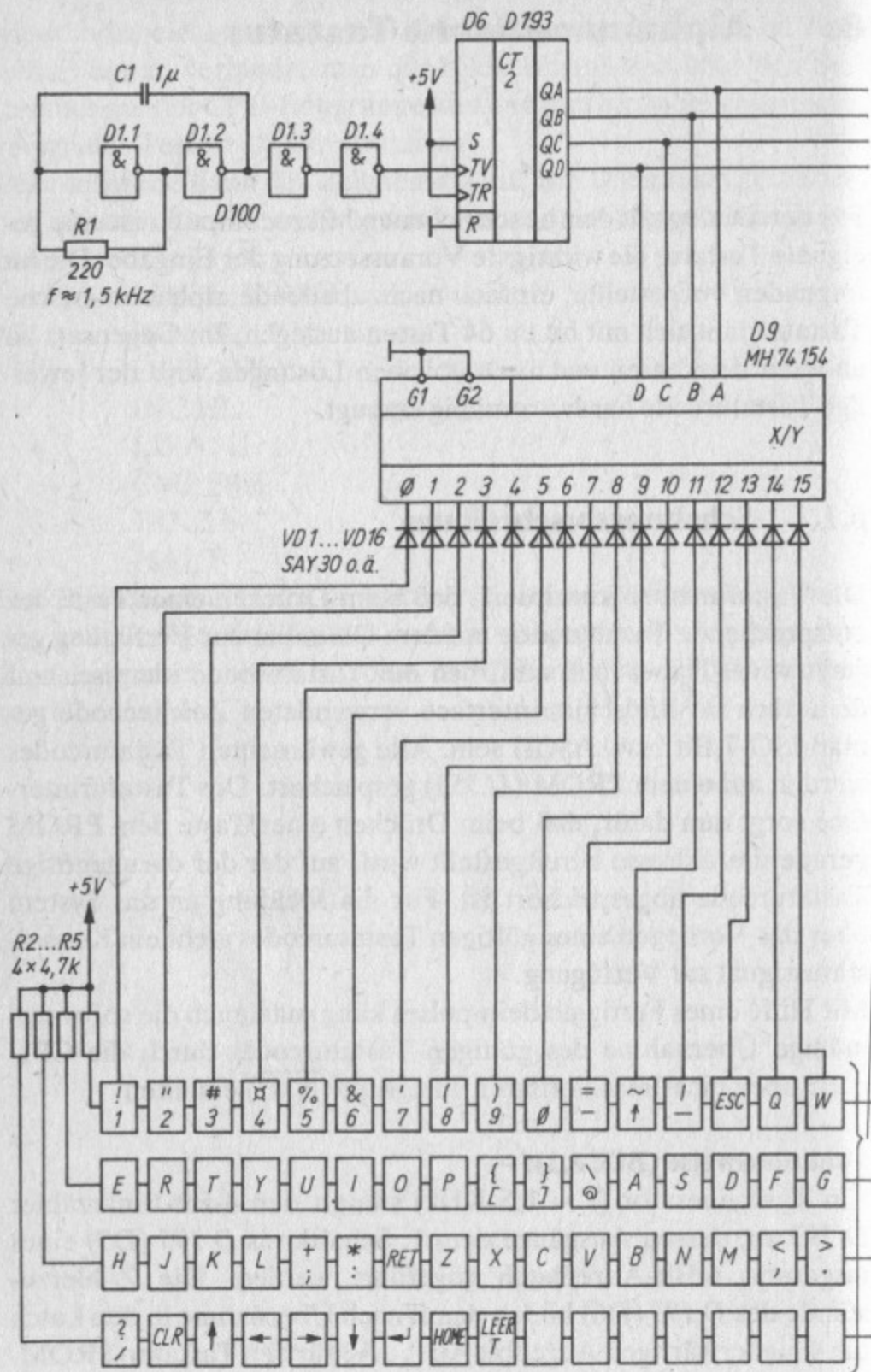


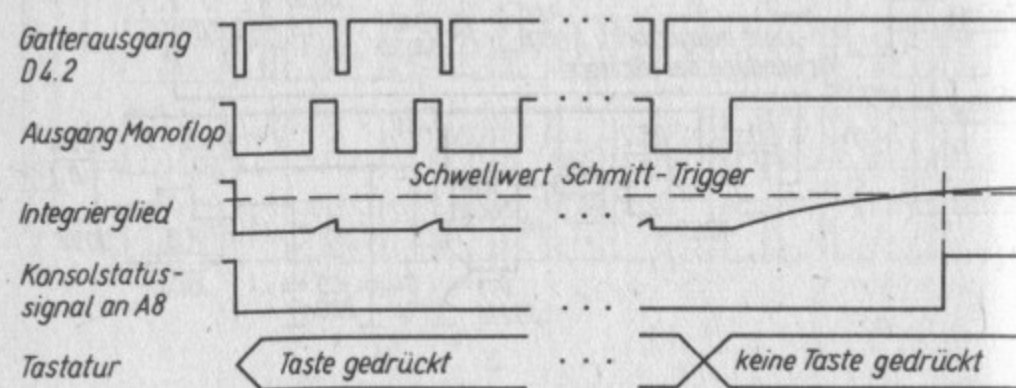
Bild 6.1 Alphanumerische Tastatur

Für den Fall, daß keine Taste gedrückt ist, liegen alle Zeilenpotentiale der Kontaktmatrix über den Widerständen R2 ... R5 auf H-Pegel. Beim Schließen eines Tastenkontaktes in der Zeile 0 wird über den jeweiligen Decoderausgang das Zeilenpotential kurzzeitig auf L gezogen. An den Gattern D3, D4.2 entsteht ein Übernahmeimpuls für die 2 Schaltkreise D 195 des Adreßlatch (HL - Flanke an T2). Damit ist im 1. D 195 (D7) der gerade gedrückten Taste entsprechende Zählerzustand des D 193 gespeichert und liegt als Adresse an den Eingängen A0 ... A3 des Tastatur-PROM vor. Die adressierte Speicherzelle stellt den gewünschten Tastaturcode am Datenbus zur Verfügung.

Der Latchimpuls kann am Ausgang von Gatter D4.1 als Fertigimpuls abgenommen werden. Gleichzeitig steuert er einen Monoflop (bestehend aus den Gattern D2.3, 2.4) an, der das Konsolstatussignal erzeugt. Dieses Signal wird nach Durchlaufen eines Integrierglieds über die Gatter D5.1, 5.4 des Schmitt-Trigger-Schaltkreises 4093 dem Adreßeingang A8 des PROM zugeführt. Dazu liegt im Normalfall der Ausgang von Gatter D5.3 auf H. Wenn eine Taste länger, als bei normaler Betätigung üblich, gedrückt wird ($t = 0,5$ s), erreicht die Spannung an C4 den Schwellwert des Schmitt-Triggers D5.3, der als Generator geschaltet ist. Der Generator schwingt mit etwa 15 Hz, und das Konsolstatussignal steht an A8 in ständiger Wiederholung bereit. Damit wiederholt sich auch die Übernahme des Tastaturcodes so lange, wie die Tastatur gedrückt ist.

Kontaktprellungen machen sich durch den Ausfall eines oder mehrerer Impulse am Gatter D4.2 bemerkbar. Wie aus dem Impulsver-

Bild 6.2 Impulsdiagramm Konsolstatussignalerzeugung



Darstellung vereinfacht, nicht maßstäblich

lauf in Bild 6.2 zu ersehen ist, wählt man die Zeitkonstante des Integrierglieds deshalb so, daß im genannten Fall der Schwellwert des nachfolgenden Schmitt-Triggers nicht erreicht wird. Damit ist eine ausreichende Prellunterdrückung gewährleistet.

Beim Betätigen einer Taste in den Zeilen 1 ... 3 laufen prinzipiell die gleichen Vorgänge ab. Zusätzlich werden jedoch die der höheren Tastenwertigkeit entsprechenden Adreßbit A4 und A5 durch die Gatter D2.1, 2.2 erzeugt und im 2. D 195 (D8) des Adreßlatch gespeichert.

Tabelle 6.1. enthält die sich ergebende Zuordnung der Tastenwertigkeiten, die der Bitbelegung A0 ... A5 am PROM entspricht.

Das RS-Flip-Flop (D4.3, D4.4) ermöglicht die Umschaltung SHIFT bzw. SHIFTLÖCK. Damit wird die bei Tastaturen übliche Zeitfunktion der Tasten (z. B. Groß- und Kleinbuchstaben) über den Adreßeingang A6 des Tastatur-PROM realisiert. Der Zustand SHIFTLÖCK wird mit der LED VB1 angezeigt.

Mit der an den Adreßeingang A7 angeschlossenen CTRL-Taste lassen sich weitere Tastenfunktionen erzeugen, natürlich nur bei entsprechenden programmierten PROM.

Der Adreßeingang A9 wird mit der Adreßlinie AB0 des Systembusses verbunden und erfüllt eine Funktion im Zusammenhang mit der Konsolstatusabfrage.

Über den CS-Eingang wird die I/O-Portzuordnung der Tastatur realisiert. Dementsprechend verbindet man einen Ausgang des I/O-Portdecoders (in diesem Fall I/O-Port 08H) auf der CPU-Platine.

Prinzipiell lassen sich die Tasten in der Tastaturmatrix beliebig zuordnen. Es wird jedoch gefordert, daß auf der jeweils erzeugten PROM-Adresse der richtige Tastaturcode abgespeichert ist.

Tabelle 6.1. Zuordnung der Tastenwertigkeiten

Zeile der Kontaktmatrix	Tastenwertigkeit (A0 ... A5)
0	0 ... 15
1	16 ... 31
2	32 ... 47
3	48 ... 63

Für die in Bild 6.1 gewählte Tastenzuordnung muß man den Tastatur-PROM U 555 gemäß ISO-7-Bit-bzw. ASCII-Code entsprechend Tabelle 6.2. programmieren. In Tabelle 6.2. wurden neben den eigentlichen Tastaturcodes auch die zur Realisierung der Konsolestatusfrage notwendigen Randbedingungen berücksichtigt.

6.2. Aufbau und Inbetriebnahme

Der Aufbau einer alphanumerischen Tastatur erfordert eine sinnvolle, d. h. bedienungsgerechte, Tastenanordnung (ähnlich einer Schreibmaschinentastatur).

Bild 6.3 zeigt eine derartige Tastenanordnung, zugeschnitten auf das vorgestellte Mikrocomputersystem. Als Tasten eignen sich beispielsweise die im Amateurhandel erhältlichen Mikrotaster des VEB *Elektroschaltgerätewerk* Auerbach.

Die Tastatur wird mit einem Kabel an den Computer angeschlossen. Bringt man den Tastatur-PROM unmittelbar in der Tastatur unter, dann ergibt sich damit eine unerwünschte Verlängerung des Systembusses. Deshalb ist es sinnvoll, den Tastatur-PROM im Computereinschub, d. h. in Systembusnähe, zu plazieren.

Zur Inbetriebnahme – vorerst ohne PROM-Bestückung – wird zunächst die Taktfrequenz von etwa 1,5 kHz am Ausgang von Gatter D1.2 bzw. D1.4 mit dem Oszilloskop kontrolliert. Anschließend führt man eine oszillographische Signalkontrolle an folgenden Stellen der Schaltung durch:

- Zählerausgänge D 193 (D6) einschließlich Eingänge D 195 (D7), MH 74154 (D9)

- Decoderausgänge MH 74154

Das Decoderausgangssignal muß bei gedrückter Taste auch auf der jeweiligen Zeile erscheinen.

- Gatterausgang D3, D4.1, D4.2, Monoflop (D2.3, D2.4) Repeaterschaltung (D5.2, D5.3), Adreßeingang A8 am Tastatur-PROM.

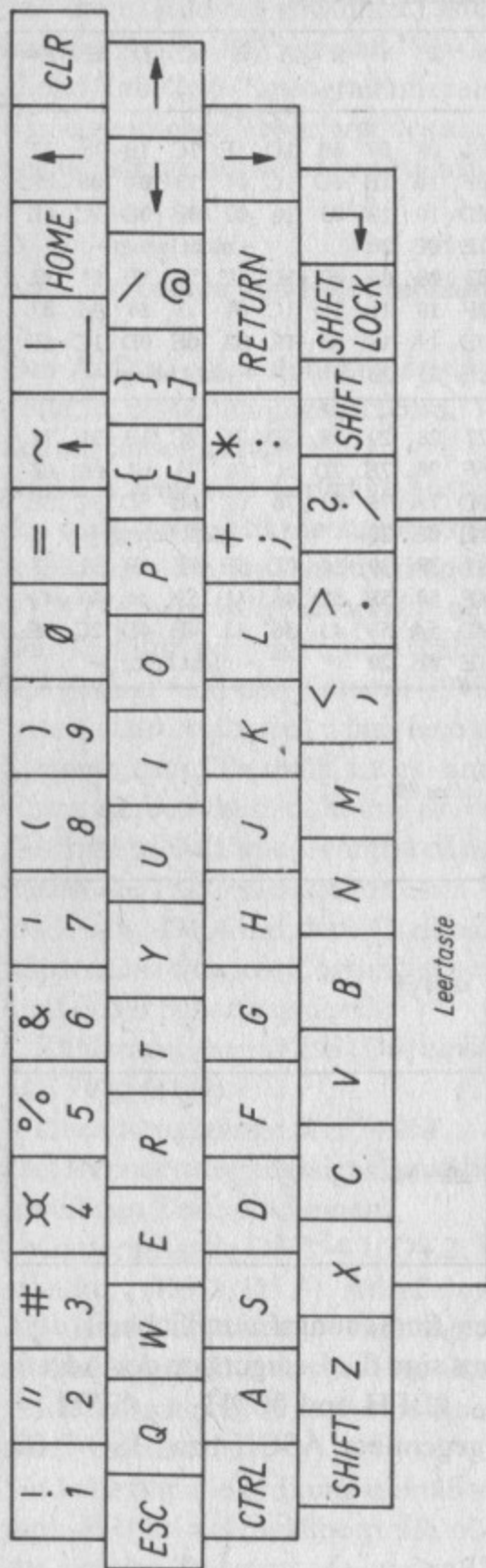
An diesen Punkten der Schaltung erscheint nur bei gedrückter Taste ein Signal. Damit schwingt also der Repeatgenerator, und am Gatterausgang D5.3 muß eine Rechteckspannung von etwa 15 Hz nachweisbar sein.

Sind alle Signale ordnungsgemäß vorhanden, steckt man den Tastatur-PROM auf und überprüft, ob nach dem Drücken einer Taste der richtige Tastaturcode an den Datenausgängen des PROM an-

Tabelle 6.2. HEXA-Tabelle Tastatur-PROM

Adresse HEXA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																
000	01	02	03	04	05	06	07	08	09	00	1D	1E	1C	1B	11	17																
010	05	12	14	19	15	09	0F	10	1B	1D	1C	01	13	04	06	07																
020	08	0A	0B	0C	0B	0A	0D	10	18	03	16	02	0E	0D	1C	1E																
030	1F	0C	0B	08	09	0A	1E	0E	20							– nicht belegt –																
040	01	02	03	04	05	06	07	08	09	00	1D	1E	1C	1B	11	17																
050	05	12	14	19	15	09	0F	10	1B	1D	1C	01	13	04	06	07																
060	08	0A	0B	0C	0B	0A	0D	1A	18	03	16	02	0E	0D	1C	1E																
070	1F	0C	0B	0C	09	0A	1E	05	20							– nicht belegt –																
080	21	22	23	24	25	26	27	28	29	30	3D	7E	7C	1B	71	77																
090	65	72	74	79	75	69	6F	70	7B	7D	5C	61	73	64	66	67																
0A0	68	6A	6B	6C	2B	2A	0D	7A	78	63	76	62	6E	6D	3C	3E																
0B0	3F	0C	0B	08	09	0A	1E	0E	20							– nicht belegt –																
0C0	31	32	33	34	35	36	37	38	39	30	2D	5E	5F	1B	51	57																
0D0	45	52	54	59	55	49	4F	50	5B	5D	40	41	53	44	46	47																
0E0	48	4A	4B	4C	3B	3A	0D	5A	58	43	56	42	4E	4D	2C	2E																
0F0	2F	0C	0B	08	09	0A	1E	0E	20							– nicht belegt –																
100	alles 00/																															
1FF																																
200																	alles FF															
2FF																																
300	alles 00																															
3FF																																

Anmerkung: Aus Gründen der Bedienungsfreundlichkeit des Monitorprogramms sind die Belegungen der Adreßbereiche 080H ... 0BFH und 0C0H ... 0FFH (= SHIFT/Normal) gegenüber ASCII bzw. ISO-7-Bit vertauscht.



Hinweis: Für die Tasten SHIFT, RETURN, Leertaste sollten möglichst 2 parallel geschaltete Mikroaster verwendet werden.

Bild 6.3 Tastenanordnung für alphanumerische Tastatur

liegt. Gemäß ISO-7-Bit bzw. ASCII (siehe Anhang) muß sich z. B. für den Buchstaben A der Code 41 H ergeben, wenn nicht gleichzeitig die Taste SHIFT gedrückt ist (bzw. SHIFTLOCK nicht vorher gedrückt wurde). In dieser Art lassen sich alle Tastenfunktionen bzw. -codes kontrollieren.

Als praktisches Hilfsmittel für diese Kontrolle erweist sich die Verwendung von 8 LED mit entsprechenden Treibertransistoren. Notfalls erfüllt natürlich auch ein einfacher Vielfachmesser den gewünschten Zweck.

Das Zusammenspiel der Tastatur mit dem Computer kann mit einem kleinen Testprogramm bzw. dem in Abschnitt 9. behandelten Monitorprogramm überprüft werden.

Vor dem Start des im folgenden angegebenen Testprogramms sind der CS-Eingang des Tastatur-PROM auf das I/O-Port 08H und der Adreßeingang A9 auf die Adreßlinie AB0 des Systembusses zu legen. Der Bildschirm muß auf E800H selektiert sein.

M1: LD HL, E7FFH

M2: INC HL

LD A, H

CMP EEH

JRZ M1

M3: IN A, (09H)

OR A

JRNZ M3

M4: IN A, (09H)

OR A

JRZ M4

IN A, (08H)

LD (HL), A

JR M2

Bei richtiger Funktion der Tastatur werden die der jeweils gedrückten Taste entsprechenden Zeichen nacheinander, links oben beginnend, auf den Bildschirm geschrieben. Bei längerem Drücken einer Taste wirkt die Repeatfunktion. Solange die Taste gedrückt ist, wird ein Zeichen in ständiger Wiederholung ausgegeben.

7. Ein-/Ausgabebaugruppen

Der Computer kann bisher nur über die Tastatur und den Bildschirm Kontakt mit der Außenwelt aufnehmen. Mit geeigneten Ein-/Ausgabeschaltungen wird der Computer zu einem universell verwendbaren Gerät, mit dem sich auch die vielfältigsten Meß-, Steuer- und Regelaufgaben sowie eine Datenübertragung realisieren lassen. Einige Interfacetechniken sollen in diesem Abschnitt beschrieben werden.

7.1. Universelle I/O-Karte

Das Herz vieler Ein-/Ausgabeschaltungen ist eine universelle I/O-Leiterkarte. Die Leiterkarte enthält 2 PIO-Bausteine, 1 SIO-Baustein und 1 CTC-Baustein. Auf der Leiterkarte sind zusätzlich noch ein Kassetteninterface (siehe Abschnitt 8.), ein Port zum Abschalten des Computers und ein Tongenerator für Signalzwecke untergebracht.

Bild 7.1 und Bild 7.2 zeigen die Schaltung der Leiterkarte. Will man das Monitorprogramm aus Abschnitt 9. verwenden, dann müssen die IOSEL-Leitungen so mit dem zentralen I/O-Decoder verbunden werden, daß sich folgende Adressen ergeben (siehe auch Tabelle 3.2.):

00H	Power-off-Port
04H	Tongenerator aus
05H	Tongenerator an
08H	Daten Tastatur
09H	Status Tastatur
0CH	Daten PIO 1 Port A
0DH	Steuerwort PIO 1 Port A
0EH	Daten PIO 1 Port B
0FH	Steuerwort PIO 1 Port B
10H	Daten PIO 2 Port A
11H	Steuerwort PIO 2 Port A
12H	Daten PIO 2 Port B
13H	Steuerwort PIO 2 Port B

14H	Kanal 0	CTC
15H	Kanal 1	CTC
16H	Kanal 2	CTC
17H	Kanal 3	CTC
18H	Daten	SIO Port A
19H	Steuerwort	SIO Port A
20H	Daten	SIO Port B
21H	Steuerwort	SIO Port B

Die PIO 1 wird im System zum Anschluß eines Lochbandlesers genutzt, Port A überträgt die Daten, Port B die Steuersignale. Mit der PIO 2 stehen weitere 16 Ein-/Ausgabeleitungen zur Verfügung.

Zur seriellen Ein-/Ausgabe ist eine SIO vorgesehen. Die SIO wandelt 8-Bit-parallele in 1-Bit-serielle Informationen um. Somit sind nur sehr wenige Verbindungsleitungen zur Datenübertragung notwendig. Weiterhin arbeiten viele externe Geräte von Natur aus seriell, beispielsweise Fernschreiber, Floppys und Magnetbandspeicher. Ein Interface zur Datenaufzeichnung auf Magnetband wird in Abschnitt 8. beschrieben. Die Datenaufzeichnung auf Magnetband ermöglicht das SIO Port A.

Zum seriellen Informationsaustausch zwischen dem Computer und anderen Geräten ist eine V24-Schnittstelle vorgesehen (SIO Port B). Anpaßstufen sorgen für die Erzeugung der notwendigen Spannungspegel. Die V24-Schnittstelle arbeitet mit den Pegeln +12 V (0-Bit) und -12 V (1-Bit). Durch die Wahl dieser Pegel wird eine sichere Datenübertragung auch über größere Entfernungen garantiert. Außer den seriellen Sende- und Empfangsdaten (TxD und RxD) werden noch folgende Steuersignale auf die Schnittstelle geführt:

RTSB	Sendeaufforderung	(Ausgang)
DTRB	Datenstation bereit	(Ausgang)
DCDB	Empfängerfreigabe	(Eingang)
CTSB	Senderfreigabe	(Eingang)

Mit diesen Steuersignalen können verschieden schnelle Geräte miteinander synchronisiert werden. Beispielsweise läßt sich mit Hilfe des CTS-Eingangs das Senden von Daten verhindern, wenn das periphere Gerät die eingetroffenen Daten noch nicht verarbeitet hat. Umgekehrt kann das Signal RTS verhindern, daß das periphere Gerät weitere Daten sendet, solange der Computer die empfangenen Informationen noch nicht verarbeitet hat. Dazu muß

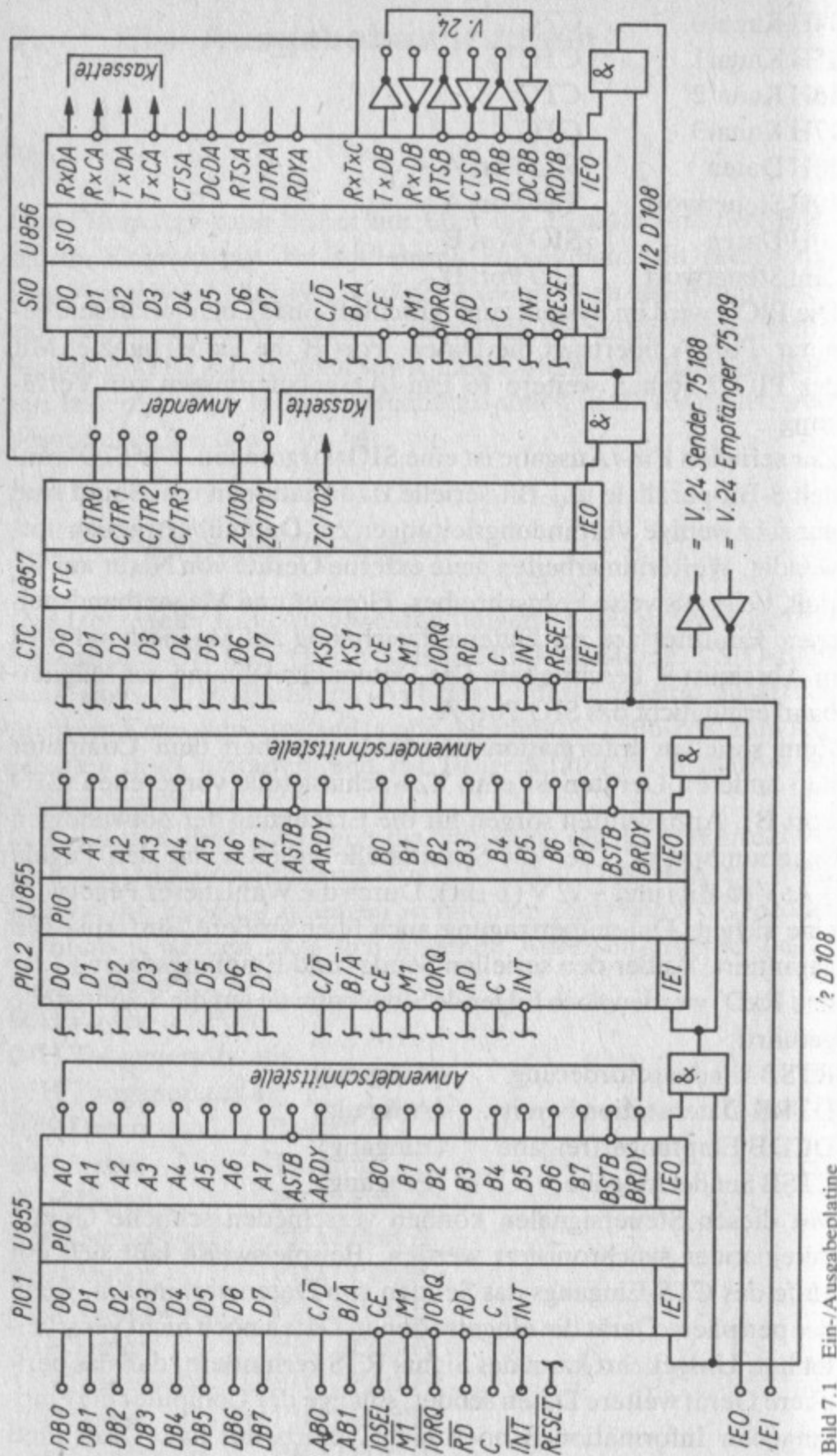


Bild 7.1 Ein-/Ausgabeplatine

man die Anschlußpunkte CTS und RTS des Computers und des peripheren Geräts kreuzweise miteinander verbinden. Die Daten werden mit genormten Geschwindigkeiten übertragen. Üblich sind z. B. folgende Übertragungsraten (Baudraten, 1 baud = 1 Bit/s):

45.45, 50, 110, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200. Die Sende- und Empfangstakte erzeugt ein CTC-Baustein. Durch entsprechende Programmierung des CTC können die verschiedenen Baudraten eingestellt werden (siehe auch Abschnitt 8.). Der CTC-Kanal 2 ist mit dem Kassetteninterface und der CTC-Kanal 1 mit dem Sende-/Empfangstakteingang \overline{RxTxCB} des SIO-Schaltkreises verbunden. Die anderen beiden CTC-Kanäle stehen dem Anwender frei zur Verfügung.

Bei systemseitigem Ansteuern der I/O-Leiterkarte wurde auf den Einsatz von Treiberschaltkreisen verzichtet. Beim Aufbau eines sehr großen Systems wird jedoch der Einsatz von Treiberschaltkreisen (z. B. 8216) empfohlen. Die AND-Gatter verringern die Einschwingzeiten der Interruptprioritätskette [2].

Auf der I/O-Karte fanden noch 2 kleine Schaltungen Platz, die in Bild 7.2 zu sehen sind. Die Schaltkreise D1, D2, D4.3 und N1 bilden einen Tongenerator, der mit dem Befehl OUT 5 angeschaltet und mit OUT 4 wieder abgeschaltet werden kann. Das JK-Flip-Flop D2 arbeitet als Frequenzteiler und als Tor. Das Rechtecksignal wird vom Transistor VT1 verstärkt und dem (hochohmigen) Lautsprecher zugeführt.

Wenn wie im Mustergerät die Netzzuschaltung mit einer Relais-selbsthalteschaltung erfolgt, so kann mit der 2. Schaltung der Computer softwaregesteuert abgeschaltet werden. Der Kontakt des Relais K unterbricht nach der Befehlssequenz OUT 0, OUT 0 und nach einer Zeit von etwa 18 s die Spannungszuführung des Netzrelais. Die Zeitverzögerung wird vom RC-Glied am Eingang des Triggerschaltkreises N2 bestimmt. Das Schieberegister D3.1 und D3.2 verzögert das Aufladen des Kondensators um einen OUT-Befehl. Mit $\overline{IN0}$ -Befehlen, \overline{RESET} oder \overline{NMI} kann die Abschaltlogik zurückgestellt werden. Der Transistor VT2 beschleunigt in diesen Fällen das Entladen des Kondensators.

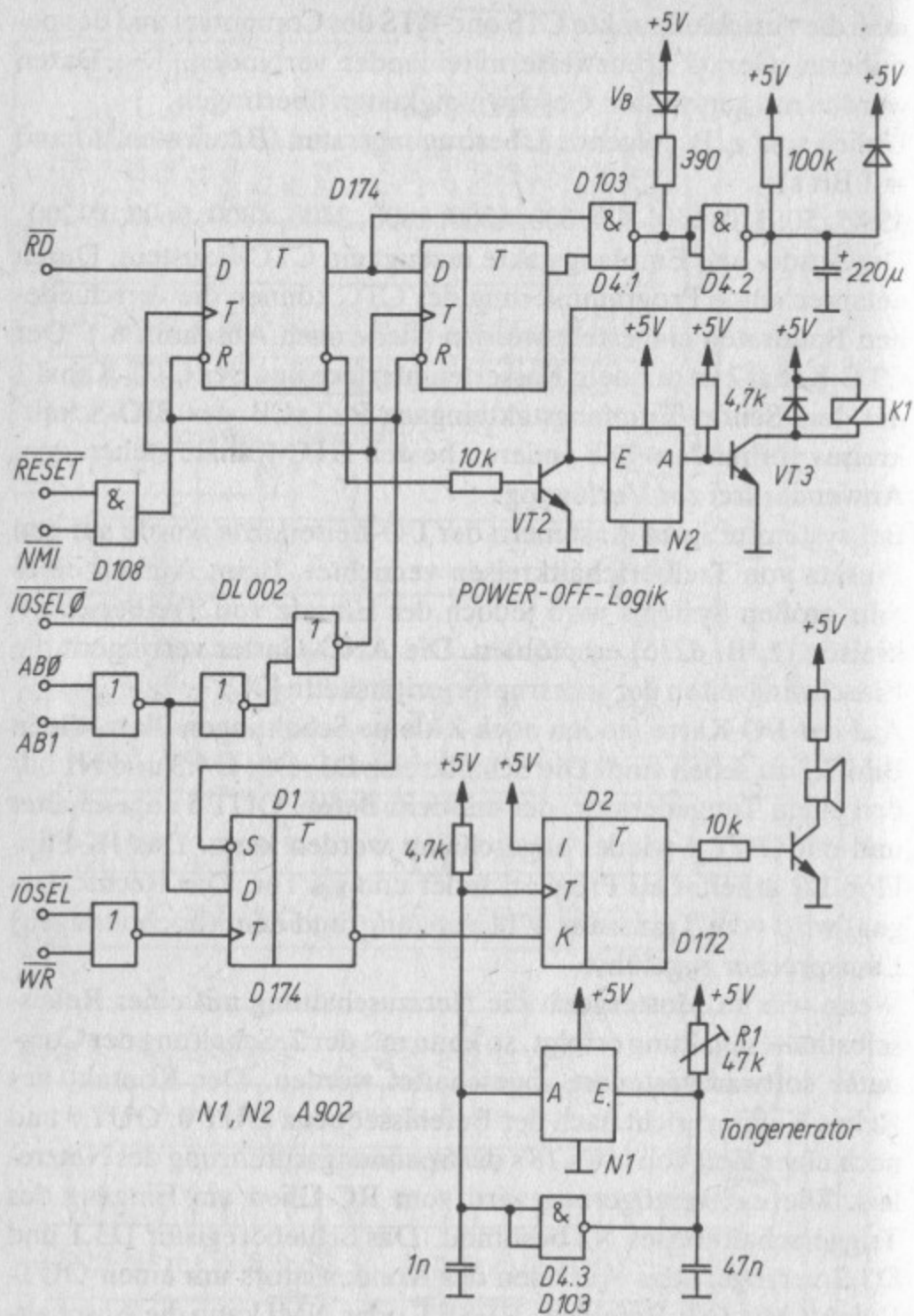


Bild 7.2 Power-off-Schaltung, Tongenerator

7.2. EPROM-Programmiergerät

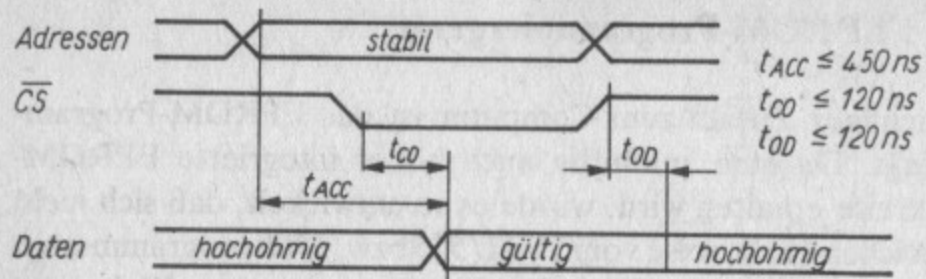
Ein wichtiger Zusatz zum Computer ist das EPROM-Programmiergerät. Da man zukünftig auch höher integrierte EPROM-Schaltkreise erhalten wird, wurde es so entwickelt, daß sich nicht nur Speicherschaltkreise vom Typ U 555 bzw. 2708 programmieren lassen, sondern das gesamte Typenspektrum: vom „Kleinsten“ 2704 bis zum 27256. Die Anschlußbelegungen der verschiedenen EPROM-Typen sind im Anhang enthalten. Ein Vergleich der Pin-Belegungen zeigt, daß es nur wenige Unterschiede gibt. Bei EPROM-Schaltkreisen im 24poligen DIL-Gehäuse sind es die Pins 18 ... 21 und 24. Über diese Pins werden \overline{CS} , Programmierspannung und Programmierimpulse, Betriebsspannungen und/oder weitere Adreßbit zugeführt. EPROM-Schaltkreise ab 2764 befinden sich im 28poligen Gehäuse. Bei diesen Schaltkreisen kommen noch 4 Anschlüsse hinzu, für die gleiches gilt. Die genannten Signale werden über Codierstecker zugeführt. Mit Hilfe dieser Codierstecker wird die Hardware an den jeweiligen EPROM-Typ angepaßt.

7.2.1. Programmiervorschriften

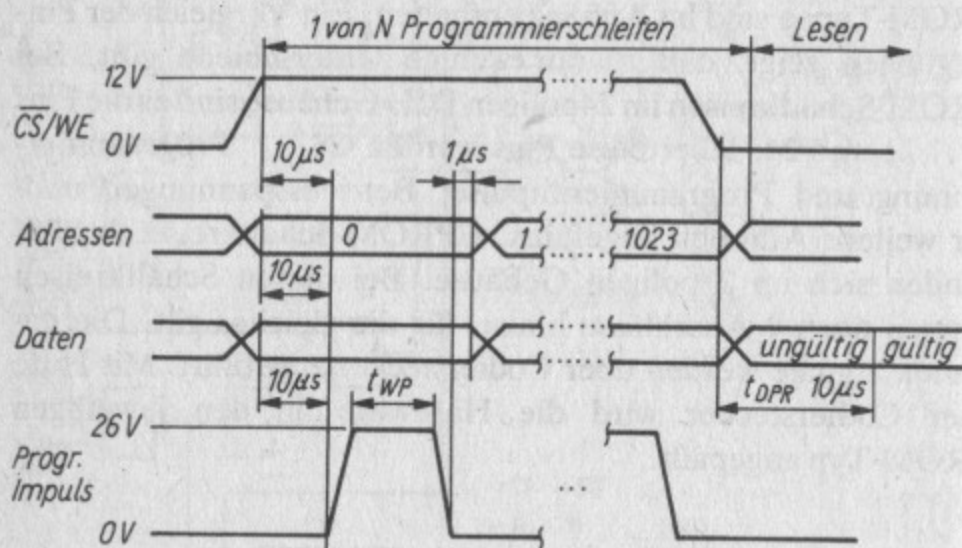
Im unprogrammierten Zustand des EPROM-Schaltkreises sind alle Bit 1 (Ausgänge H-Pegel). Programmiert wird somit durch Einschreiben von 0 Bit in die adressierten Zellen. Eine programmierte 0 kann nicht mehr umprogrammiert werden.

Bild 7.3 verdeutlicht das Programmieren des U 555 an Hand des Impulsdiagramms. Den $\overline{CS}/\overline{WE}$ -Eingang legt man auf +12 V, anschließend die 1. Adresse und das zugehörige Datenwort (Ausgänge $0_0 \dots 0_7$) an den EPROM. Es folgt ein Programmierimpuls (0,1 ... 1 ms, 26 V). Danach wird die Adresse um 1 erhöht und der Vorgang wiederholt, bis alle Adressen abgearbeitet sind. Das muß N-mal für alle 1024 Adressen wiederholt werden. Die Anzahl der Programmierschleifen N hängt von der Breite des Programmierimpulses t_{pw} ab. Die integrale Programmierdauer je Speicherplatz beträgt 100 ms, so daß bei $t_{wp} = 1$ ms 100 Programmierschleifen notwendig sind (Intel-Programmierschrift).

Moderne EPROM lassen sich einfacher programmieren. Der Schaltkreis 2716 wird mit 50-ms-TTL-Impulsen programmiert. Er



a)



b)

Bild 7.3 a – Impulsdiagramm U555, b – Impulsdiagramm 2708

gestattet das Programmieren einzelner Speicherzellen und das Prüfllesen während des Programmiervorgangs. Das Programmieren kann folgendermaßen ablaufen: Anlegen der Programmierspannung U_{pp} (25 V) an Pin 21, $\overline{OE} = H$ -Pegel, Anlegen Adresse und Datenbyte, Programmierimpuls (max. 55 ms) dem Eingang \overline{CE} zuführen. Ein Prüfllesen ist möglich, wenn der \overline{OE} -Eingang auf L gelegt wird, dabei muß U_{pp} nicht abgeschaltet werden. Bild 7.4 zeigt das Impulsdiagramm.

Der Schaltkreis 2732 wird ähnlich programmiert (Bild 7.5). Die Programmierspannung (25 V) wird dem Eingang \overline{OE}/V_{pp} zugeführt, der Programmierimpuls (50 ms; L-Pegel) dem Eingang \overline{CE} .

Die genauen Programmiervorschriften sind unbedingt den Datenblättern des Herstellers zu entnehmen. Beispielsweise wird für den U 555 eine integrale Programmierzeit von 50 ms je Speicherzelle

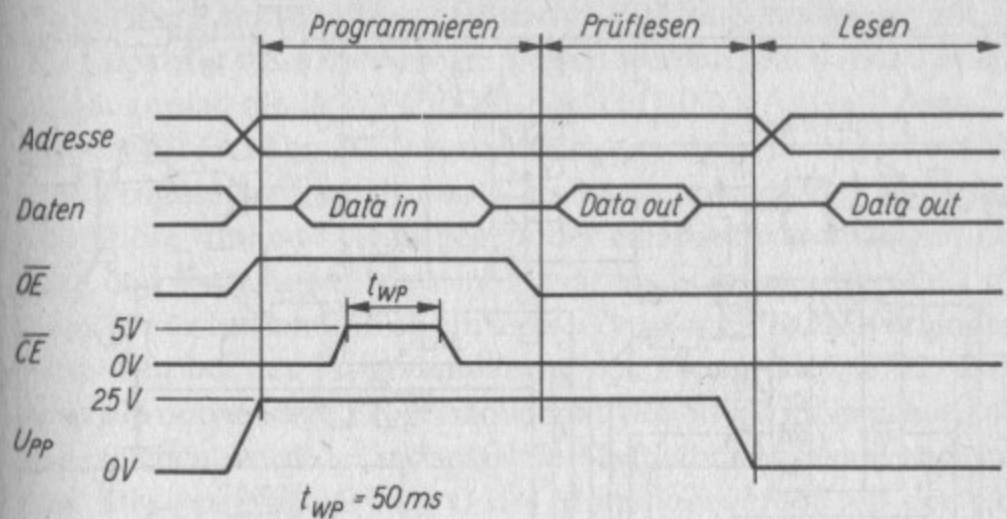


Bild 7.4 Impulsdiagramm 2716

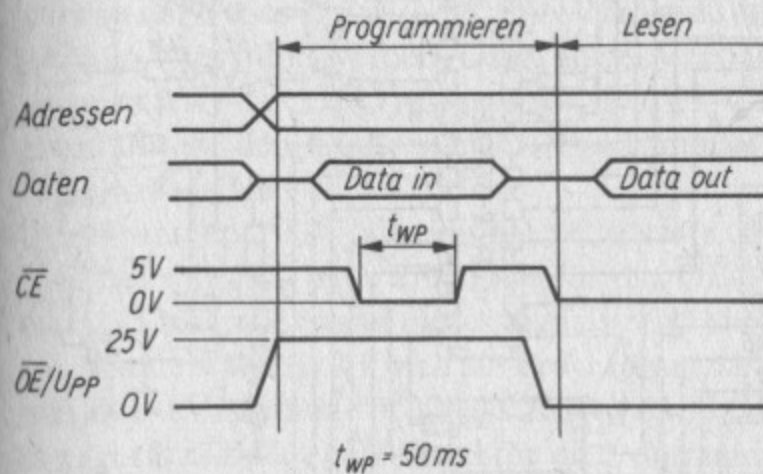


Bild 7.5 Impulsdiagramm 2732

[6], für den Intel 2708 aber 100 ms [7] angegeben. Ein weiterer Hinweis in diesem Zusammenhang: Die EPROM-Schaltkreise TMS 2716 und i 2716 sind nicht kompatibel.

7.2.2. Die Schaltung

Bild 7.6 zeigt die Schaltung des Programmiergeräts. Ein PIO-Baustein überträgt die Daten und Steuersignale von und zum Programmiergerät. Die Daten werden über Port A übertragen. Um einen PIO-Baustein einzusparen, erzeugt man die Adressen hardwaremäßig. Dafür sind die Zähler D1 ... D4 vorgesehen. Der Zähler

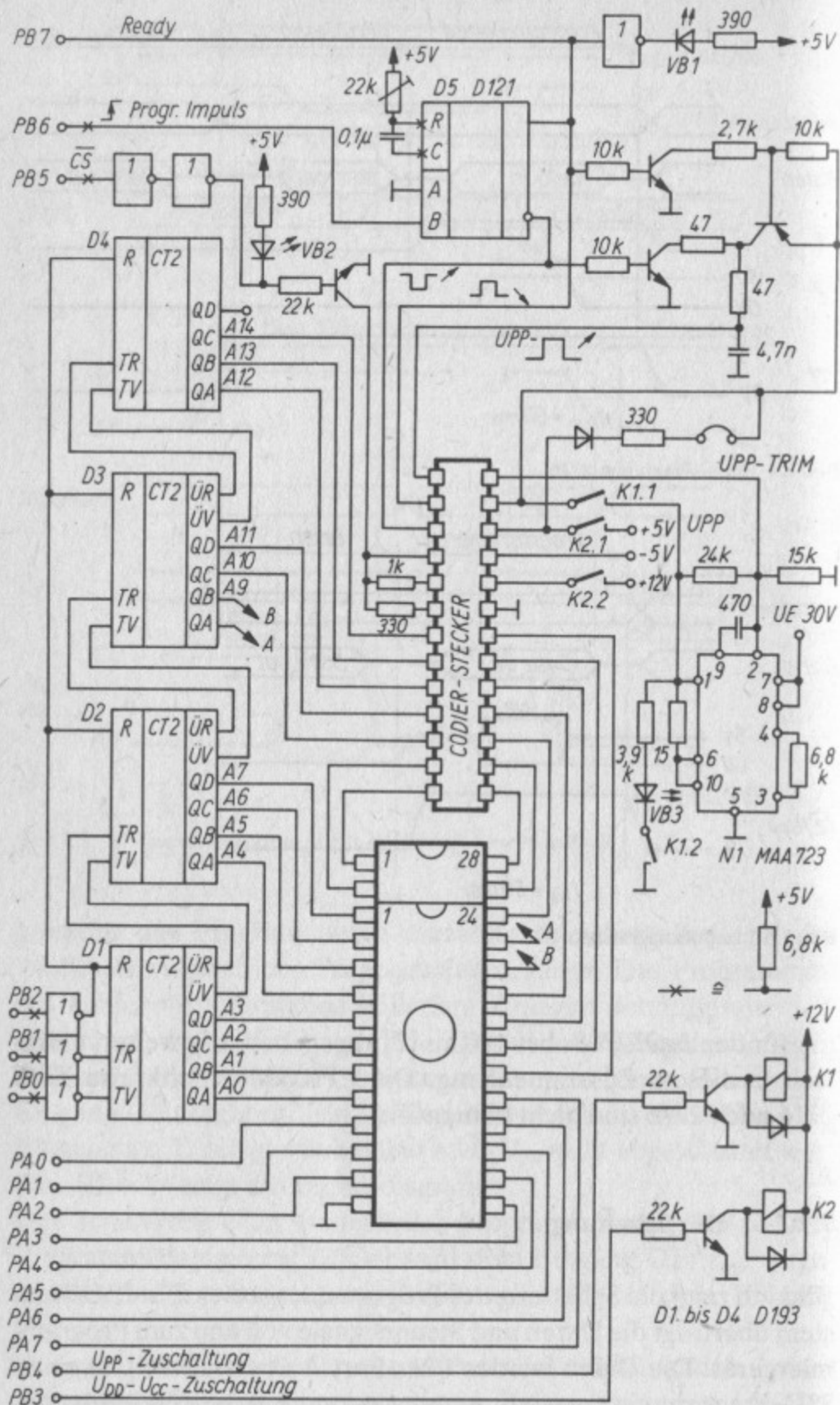


Bild 7.6 EPROM-Programmiergerät

D4 wird erst notwendig, wenn auch EPROM-Schaltkreise mit einer Kapazität > 4 kByte programmiert werden sollen. Die Zählerausgänge sind mit dem EPROM-Sockel (A0 bis A9) und dem Codierstecker (A10 bis A14) verbunden.

Den Programmierimpuls von 1 ms Länge erzeugt der Monoflop D5. Diese Variante bietet gegenüber einer softwaremäßigen Lösung den Vorteil, daß auch im Havariefall die Impulsbreite 1 ms nicht überschritten und so ein Zerstören des EPROM verhindert wird. Um bei der Programmierung der Typen 2716, 2732, 2764 usw. die notwendige Programmierzeit von 50 ms je Speicherplatz zu erreichen, wird der monostabile Multivibrator 50mal angestoßen. Die Ausgänge Q und \bar{Q} des Monoflop werden auf den Codierstecker geführt. Diese beiden Signale nutzt man bei den Schaltkreisen 2716, 2732 usw. als Programmierimpuls. Beim Programmieren der Typen U 555, 2708, TMS 2716 muß der 1-ms-Impuls auf etwa 26 V verstärkt werden. Dafür ist der Verstärker mit den Transistoren VT1 . . . VT3 vorgesehen. Die 26-V-Impulse gelangen ebenfalls an den Codierstecker. Programmierimpulse zeigt die Leuchtdiode VB1 an.

Der Transistor VT4 arbeitet als Schaltstufe für das $\overline{CS}/\overline{WE}$ -Signal beim U 555 bzw. 2708 oder für das \overline{OE} -Signal beim 2716 und 2732. Den Zustand dieses Signals zeigt die Leuchtdiode VB2 an. Mit dem Relais K1 wird die Programmierspannung zugeschaltet. Die Leuchtdiode VB3 hat Kontrollfunktion. Das Programmiergerät enthält ein Netzteil für die Programmierspannung U_{pp} . Die Spannung stabilisiert man mit dem Schaltkreis N1 (MAA 723). Der Knotenpunkt der Widerstände $R1/R2$ wird auf den Codierstecker geführt, um dort die genaue Programmierspannung mit einem Zusatzwiderstand festzulegen. Das Relais K2 schaltet die Betriebsspannung des EPROM-Schaltkreises softwaregesteuert zu.

7.2.3. Aufbau und Codierung

Die Programmierereinrichtung wurde als externes Gerät aufgebaut und läßt sich bei Bedarf über ein Kabel mit der PIO 2 verbinden. Das hat den Vorteil, daß die PIO 2 auch für andere Ein-/Ausgabearbeiten genutzt werden kann.

Als EPROM-Fassung sollte ein 28poliger Nullkraftsockel eingesetzt werden. Ein 26poliger direkter Steckverbinder nimmt die Co-

dierstecker auf. Sie bestehen aus 30 mm × 35 mm großen doppelt kaschierten Leiterplatten. Bild 7.7 zeigt die Anschlußbelegung der Codiereinrichtung. Bild 7.8 bis Bild 7.10 zeigen die Beschaltung der Pins 18...21 sowie die Verdrahtung der Codierstecker für die EPROM-Typen U 555 (2708), 2716 und 2732 (vergleiche auch Bild 7.3 bis Bild 7.5).

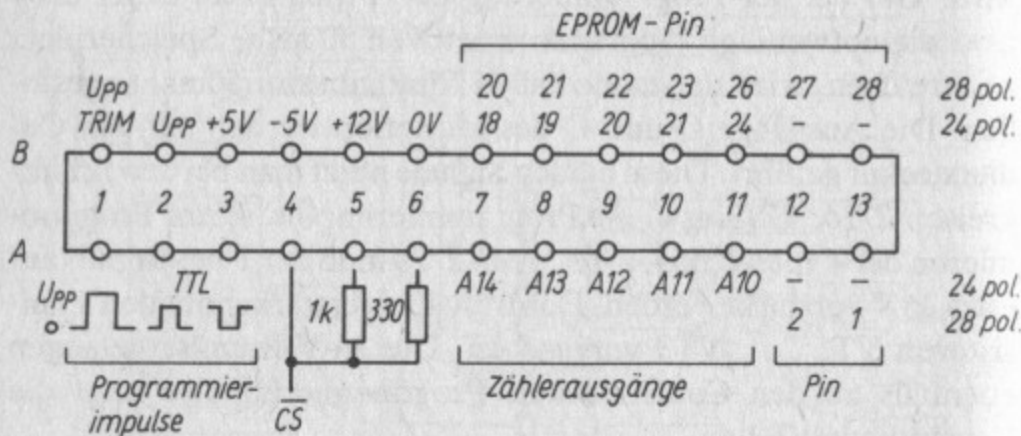


Bild 7.7 Anschlußbelegung der Codiereinrichtung

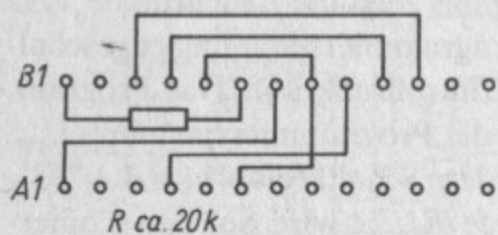


Bild 7.8 Codierstecker für U 555 (2708)

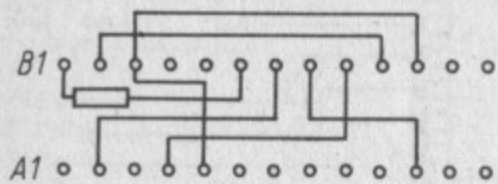


Bild 7.9 Codierstecker für 2716

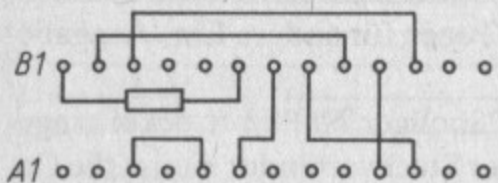


Bild 7.10 Codierstecker für 2732

Die Verbindung des Programmiergeräts mit dem Computer übernimmt die PIO 2. Dabei sind die beiden Ports wie folgt belegt:

- Port A Daten
- Port B
- 0 Vorwärtszählen (positiver Impuls)
 - 1 Rückwärtszählen (positiver Impuls)
 - 2 Rücksetzen des Zählers (negativer Impuls)
 - 3 Zuschalten U_{cc} und U_{DD} (H)
 - 4 Zuschalten U_{pp} (H)
 - 5 $\overline{CS/WE}$ bzw. \overline{OE} (H = Lesen)
 - 6 Programmierimpuls (L/H-Flanke)
 - 7 Fertigsignal (H für die Dauer des Impulses)

7.2.4. Ein Beispielprogramm

An einem BASIC-geschriebenen Programm soll das Programmieren eines U 555 demonstriert werden. Dabei wird abweichend von der Intel-Vorschrift programmiert [8], [9].

```

10 PRINT
20 PRINT „U 555/2708 PROGRAMMER“
30 PRINT
60 DA = 16:CA = 17:DB = 18:CB = 19:REM
  Ports festlegen
70 OUT CB, & CF:OUT CB, & 80:REM Bitmode, Bit
  7 Eingabe
80 OUT CA, & 7F:REM Byteeingabe
90 OUT DB, & 21:REM Spannungen aus, Zählerreset.
100 INPUT „LESEN (1) PROG (2) VERGLEICHEN (3)
  LÖSCHTEST (4)“; M
110 IF M > 4 THEN END
120 ON M GOSUB 250, 320, 550, 140
130 GOTO 90
140 REM Löschkontrolle
150 OUT CA, & 7F:REM Byteeingabe
160 OUT DB, & 29:OUT DB, & 2D:REM Zählerrest
170 F = 0:REM Flag
180 FOR I = 1 TO 1024:IF INP (DA) <> & FF THEN F =
  1:I = 1024
190 OUT DB, & 2c:OUT DB, & 2D:REM weiterzählen
200 NEXT I
  
```

```

210 PRINT „EPROM“; : IFF = 1 THEN PRINT „NICHT“;
220 PRINT „GELÖSCHT“ : RETURN
250 REM Dublizieren des EPROM-Inhalts ins RAM
260 INPUT „ADRESSE“; A
270 OUT CA, & 7F : REM Byteeingabe
280 OUT DB, & 29 : OUT DB, & 2D : REM Reset
290 FOR I = A TO A + 1023 : POKE I, INP (DA)
300 OUT DB, & 2c : OUT DB, & 2D : REM weiterzählen
310 NEXT I : RETURN
320 REM programmieren
330 INPUT „ADRESSE“; A : GOSUB 150 : IFF = 1 THEN
INPUT „WEITER (J/N)“; W$ : IF W$ <> „J“ THEN
RETURN
340 PRINT : Z = 0 : REM Zykluszähler = 0
350 N = 2 : GOSUB 430 : REM 2 Programmiersversuche
360 REM Prüfllesen
370 OUT CA & 7F : OUT DB, & 2D : OUT DB, & 29 : OUT
DB, & 2D
380 F = 0 : FOR I = A TO A + 1023 : IF INP (DA) <> PEEK
(I) THEN F = 1 : I = A + 1024
390 OUT DB, & 2C : OUT DB, & 2D : NEXT I
400 IFF = 1 AND Z < 32 THEN 350
410 IFF = 1 THEN PRINT „NICHT PROGRAMMIERT“; :
GOTO 520
420 PRINT : N = Z/2 : GOSUB 430 : GOTO 520 : REM
Sicherheitszyklen
430 OUT DB, & 29 : OUT DB, & 19 : OUT DB, & 1D : REM
Zählerreset und Spannungen anlegen
440 OUT CA, & F : REM Byteausgabe
450 FOR I = 1 TO N : Z = Z + 1 : PRINT CHR$( &B);
USING „PROG ZYKLUS # # #“; Z
460 OUT DB, & 19 : OUT DB, & 1D : REM Zähler
rücksetzen
470 FOR I = A TO A + 1023 : OUT DA, PEEK (I)
475 REM Programmierimpuls erzeugen und weiterzählen
480 OUT DB, & 5D : OUT DB, & 1D : OUT DB, & 1C :
OUT DB, & 1D
490 NEXT I, J : RETURN
500 REM Auf Fertigsignal (Bit 7) wird nicht gewartet,
weil Schleifenlaufzeit > 1 ms

```

```

510 OUT CA & 7F : REM Byteeingabe
520 OUT DB, & 2D : REM Auf Lesen schalten
525 REM Prüfllesen, anschließend Signalton
530 GOSUB 560 : OUT 5, 0 : FOR I = 1 TO 500 : NEXT
I : OUT 4, 0
540 RETURN
550 INPUT „ADRESSE“; A
560 OUT CA, & 7F : REM Byteeingabe
570 OUT DB, & 29 : OUT DB, & 2D : REM Zähler
rücksetzen
580 E = 0 : FOR I = A TO A + 1023
590 IF INP (DA) <> PEEK (I) THEN E = E + 1
600 OUT DB, & 2C : OUT DB, & 2D : REM weiterzählen
610 NEXT I : PRINT E; „FEHLER“ : RETURN

```

7.3. Analogschnittstellen

Will man mit dem Computer auch Analogsignale oder Meßwerte erfassen und verarbeiten, so sind geeignete Analog/Digital-Wandler und Digital/Analog-Wandler erforderlich. In diesem Abschnitt werden einige Analogschnittstellen beschrieben.

7.3.1. Digital/Analog-Wandler

International gibt es eine große Anzahl hybrider und monolithischer D/A-Wandlerbausteine. Ein besonders günstiger Baustein ist der sowjetische CMOS-Wandler *K 572 ПA1*. Es handelt sich bei diesem Schaltkreis um einen TTL-kompatiblen multiplizierenden 10-Bit-Wandler mit einer Einschwingzeit von 500 ns. Die Betriebsspannung kann 5 . . . 15 V ($P_{vmax} = 20$ mW), die Referenzspannung maximal ± 10 V betragen. Bild 7.11 zeigt das Funktionsprinzip des *K 572 ПA1* und Bild 7.12 die Standardbeschaltung. Die Ausgangsspannung beträgt

$$U_A = U_{ref} \sum_{i=1}^n \frac{S_i}{2^n - i + 1}$$

Dabei gilt: $S_i = 1$ bei H und $S_i = 0$ bei L am Digitaleingang E_i und $n = 10$.

Bild 7.13 zeigt die Beschaltung des D/A-Wandlers für bipolare

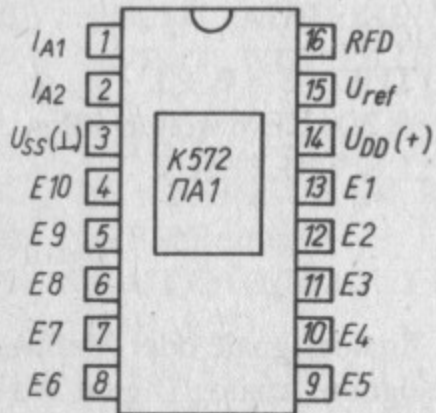
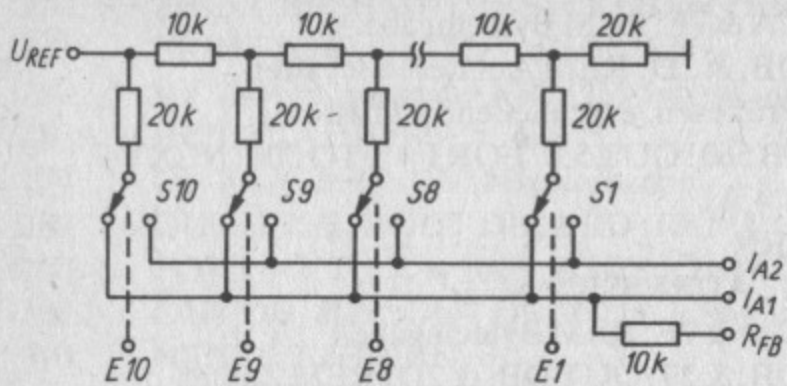


Bild 7.11 D/A-Wandler K 572 П A1

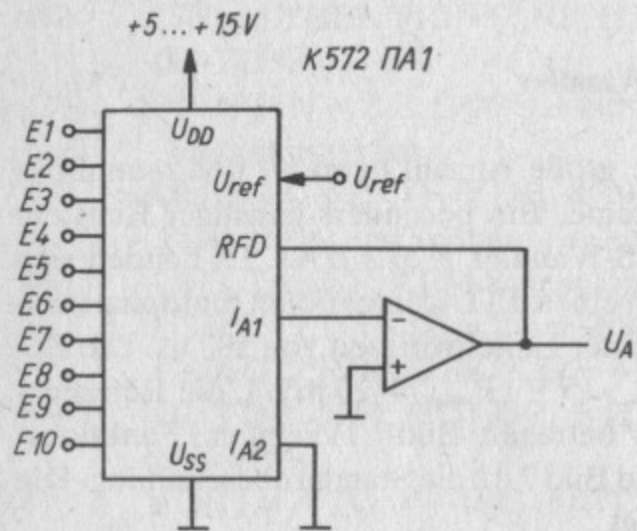


Bild 7.12 Standardbesetzung des K 572 П A1 für unipolare Ausgangsspannung

Ausgangsspannung. Die erreichbare Umsetzungsgeschwindigkeit ist von der Slew-Rate der verwendeten Operationsverstärker abhängig. Der A/D-Wandler K 572 П A1 gestattet einfache und elegante Problemlösungen. Er läßt sich beispielsweise in den Gegenkopplungszweig eines Verstärkers schalten, um eine digitale Verstär-

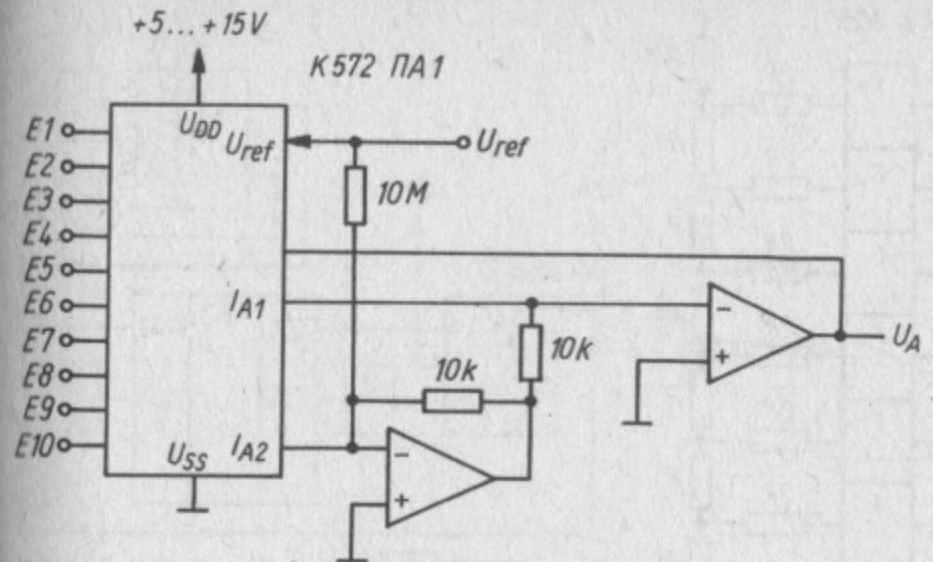


Bild 7.13 Standardbesetzung des K 572 П A1 für bipolare Ausgangsspannung

kereinstellung zu ermöglichen, oder man verwendet ihn als digital steuerbaren Spannungsteiler. Dabei können von dem D/A-Wandler auch Wechselspannungen bis zu $U_{ss} = 20\text{ V}$ verarbeitet werden. Der Wandler läßt sich über einen PIO-Baustein an den Computer an koppeln. Sind keine speziellen D/A-Wandlerschaltkreise verfügbar, so bieten sich auf der Grundlage von R-2R-Netzwerken verschiedene Lösungsvarianten an. Bild 7.14 und Bild 7.15. zeigen

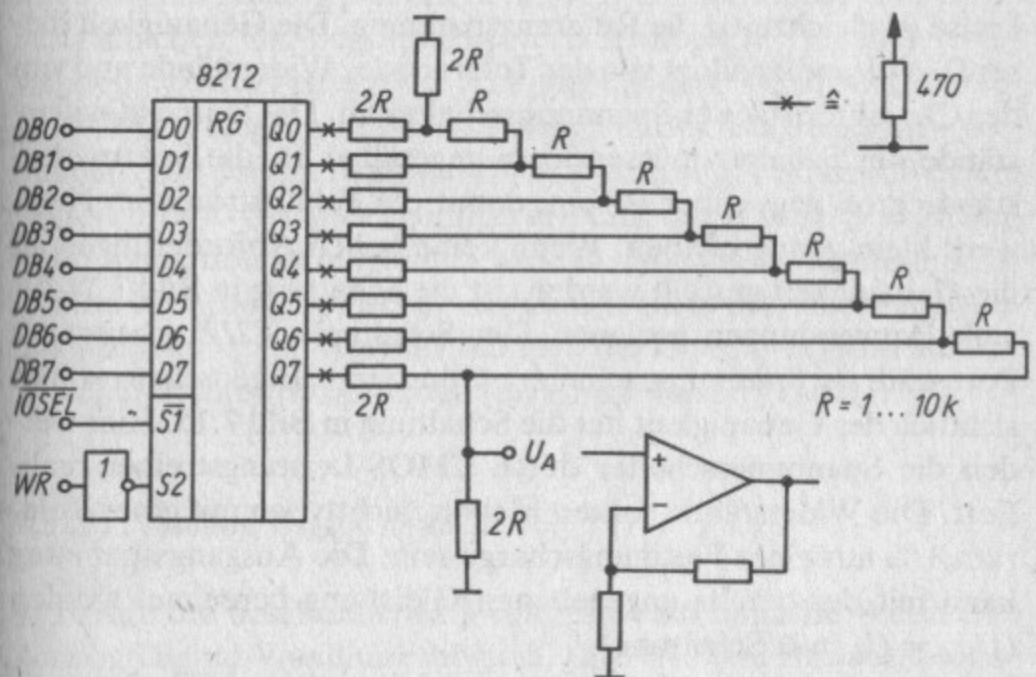


Bild 7.14 D/A-Wandler mit 8212

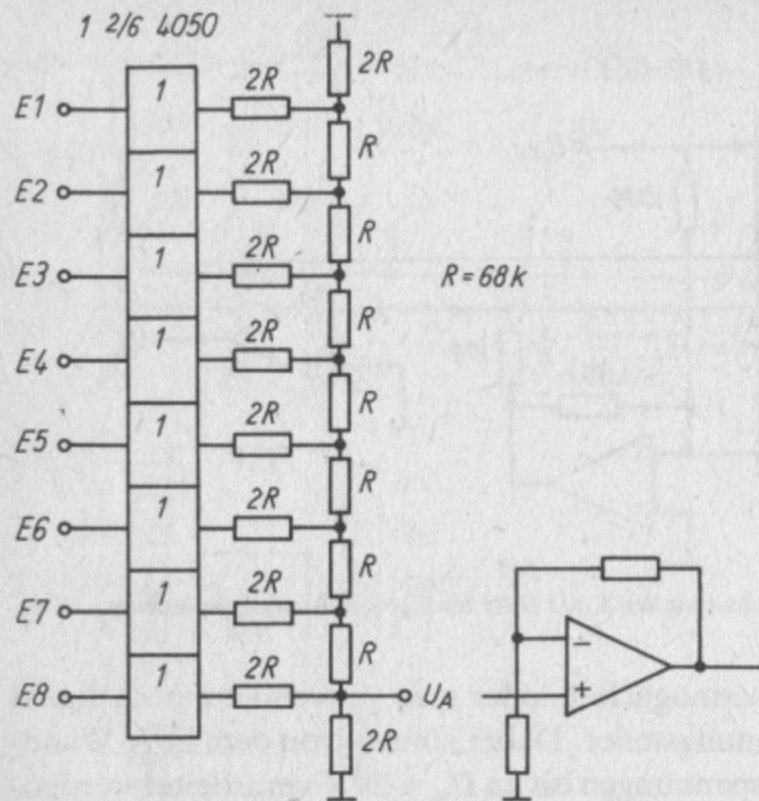


Bild 7.15 D/A-Wandler mit CMOS-Schaltkreisen 4050

2 einfache Schaltungen. Die Ausgangsstufen der Schaltkreise arbeiten als Spannungsschalter. Die Betriebsspannung der Schaltkreise ist gleichzeitig die Referenzspannung. Die Genauigkeit dieser D/A-Wandler hängt von der Toleranz der Widerstände und von der Charakteristik der Spannungsschalter ab. Die Durchlaßwiderstände der Schalter müssen klein gegenüber R , die Sperrwiderstände groß gegenüber R sein, damit die Fehlerströme im Netzwerk klein genug bleiben. Wenn keine hohen Anforderungen an die Genauigkeit gestellt werden, ist die Schaltung in Bild 7.14 für viele Anwendungen geeignet. Der Schaltkreis 8212 arbeitet als Port und als Spannungsschalter. Günstigere Eigenschaften hinsichtlich der Genauigkeit hat die Schaltung in Bild 7.15. Hier werden die Spannungsschalter durch CMOS-Leistungstreiber realisiert. Die Widerstände sollten Metallschichttypen mit einer Toleranz 1% aus einer Fertigungscharge sein. Die Ausgangsspannung kann mit der bereits angegebenen Gleichung berechnet werden ($U_{Ref} = U_{cc}$, $n \triangleq$ Schaltung).

In Anwendungsfällen, bei denen es nicht so sehr auf hohe Umsetzgeschwindigkeiten ankommt, bieten sich D/A-Wandler an, die

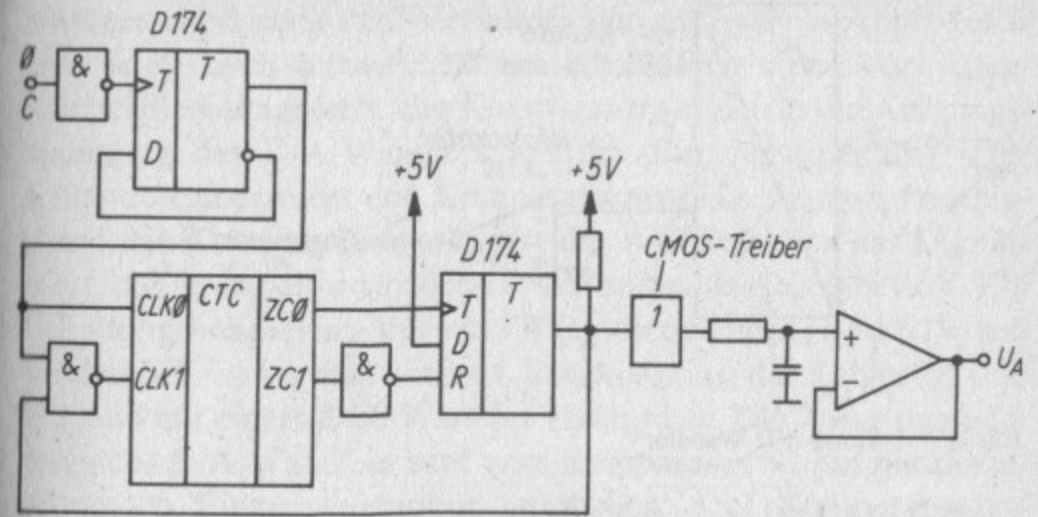


Bild 7.16 D/A-Wandler mit CTC-Baustein

durch die Pulsbreitenmodulation und anschließende Integration sehr genau umsetzen. Diese D/A-Wandler haben den Vorteil, daß sie sich ohne Präzisionsbauelemente aufbauen lassen [10]. Bild 7.16 zeigt einen 8-Bit-Digital/Analog-Umsetzer mit einem CTC-Baustein (siehe auch [2]). Beide CTC-Kanäle arbeiten im Zählermode. Der CTC-Kanal 0 erzeugt den Referenztakt ($TC = 256$). Der Kanal 1 wird mit dem zu wandelnden Wert geladen. Mit 2 Digital/Analog-Wandlern (z. B. Bild 7.14) ist der Aufbau eines einfachen Oszillographen-Interface möglich. Bei dieser Anwendung wird der eine DAC-Ausgang mit dem X-Eingang, der andere DAC-Ausgang mit dem Y-Eingang des Oszillographen verbunden. Hat der Oszillograph einen Z-Eingang, so kann mit einem weiteren D/A-Wandler (geringerer Auflösung) auch eine Helligkeitsmodulation durchgeführt werden. Mit dieser Konfiguration ist eine einfache graphische Ausgabe möglich. Mit einem geeigneten Analog/Digital-Wandler läßt sich der Computer dann als NF-Speicheroszillogoskop betreiben (bis einige hundert Hertz).

7.3.2. Analog/Digital-Wandler

Mit Hilfe der beschriebenen Wandler ist auf einfache Weise eine Analog/Digital-Wandlung möglich. Bild 7.17 und Bild 7.18 zeigen 2 mögliche Schaltungsvarianten, wobei die Schaltung in Bild 7.18 gleichzeitig 8 Analogkanäle erfassen kann. Das Signal wird soft-

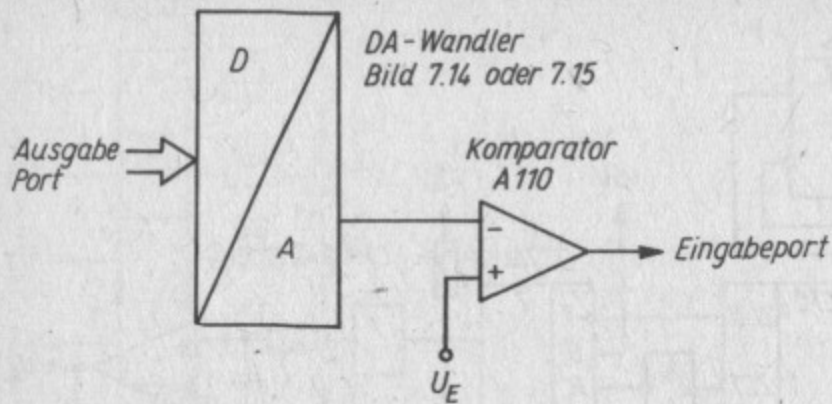


Bild 7.17 1-Kanal-A/D-Wandler

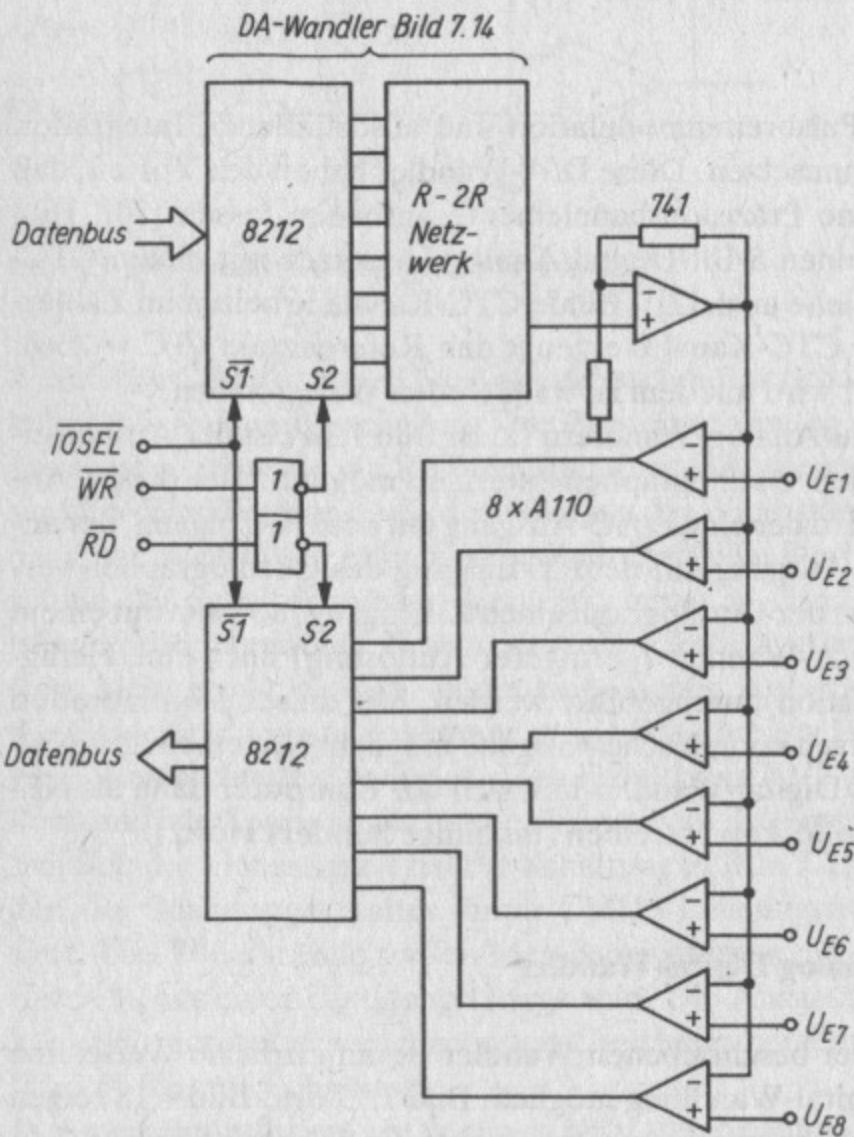


Bild 7.18 8-Kanal-A/D-Wandler

waregesteuert nach dem Verfahren der sukzessiven Approximation oder durch softwaremäßiges Nachbilden eines Vor-/Rückwärtszählers umgesetzt. Ein Komparator vergleicht die Ausgangsspannung des D/A-Wandlers mit der Eingangsspannung. Der Computer überwacht den Komparatorausgang. Je nach Pegelzustand des Komparators verändert die A/D-Software das Digitalwort, das anschließend an den D/A-Wandler ausgegeben wird. Ein Schaltungsbeispiel mit Vor- und Rückwärtszählern (Bild 7.19) soll verdeutlichen, wie man vorgeht. Die Ausgänge der Zähler D1 und D2 sind mit einem D/A-Wandler verbunden. Die Ausgangsspannung des D/A-Wandlers wird vom Komparator N1 mit der umzusetzenden Eingangsspannung verglichen. Das Komparatorausgangssignal, auf ein D-Flip-Flop geführt, bestimmt die Zählrichtung. Bei veränderter Eingangsspannung wird, je nach Vorzeichen der Änderung, vor- oder rückwärts gezählt, bis die Eingangsspannung und DAC-Ausgangsspannung gleich sind. Die Umsetzzeit ist

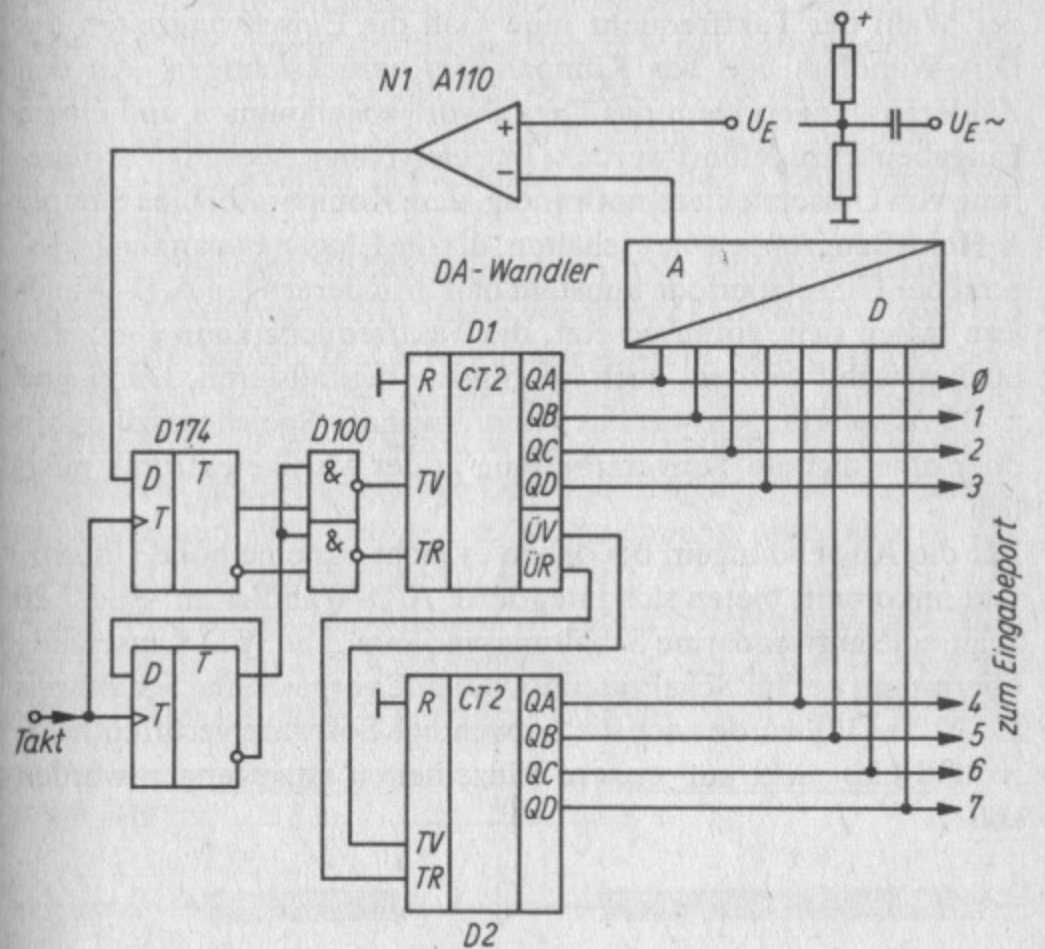


Bild 7.19 A/D-Wandler mit Vor-/Rückwärtszähler

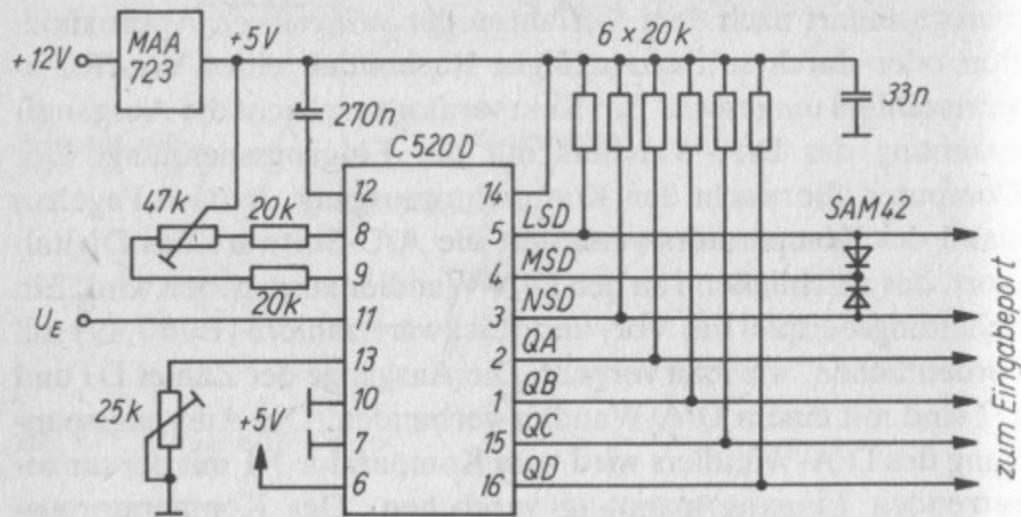


Bild 7.20 A/D-Wandler mit C 520

von der Größe der Änderung und der Taktfrequenz abhängig. Bei der Wahl der Taktfrequenz muß man die Einschwingzeiten des D/A-Wandlers und des Komparators berücksichtigen. An den Zählerausgängen kann das Digitalwort abgenommen und einem Eingabeport zugeführt werden. In vielen Fällen ist es zur Verringerung von Umsetzfehlern notwendig, dem Komparator eine Sample & Hold-Baugruppe vorzuschalten, die die Eingangsspannung während der Umsetzperiode konstant hält. Mit derartigen A/D-Wandlern lassen sich, vorausgesetzt, die Taktfrequenz kann genügend hoch gewählt werden, auch NF-Signale digitalisieren. Dann sind z. B. Anwendungen wie das oben erwähnte Speicheroszilloskop oder eine digitale Tonverarbeitung in der Musikelektronik möglich.

Für die Anwendungen, bei denen es nicht auf eine hohe Umsetzrate ankommt, bieten sich integrierte A/D-Wandler an. Bild 7.20 zeigt eine aufwandarme Schaltungsvariante. Die A/D-Umsetzung übernimmt der für Schalttafelinstrumente vorgesehene Schaltkreis C 520. In [11] wurde auch die notwendige Software veröffentlicht, so daß hier nicht auf weitere Einzelheiten eingegangen werden soll.

8. Kassetteninterface

Als externe Daten- und Programmspeicher setzt der Amateur vorwiegend Lochbänder und Magnetbänder ein. Das Magnetband ist ein preiswerter Datenspeicher hoher Kapazität. Im folgenden wird ein Kassetteninterface beschrieben, das die Aufzeichnung und die Wiedergabe von Daten und Programmen mit einem handelsüblichen Kassettenrecorder ermöglicht.

8.1. Aufzeichnungsverfahren

Ein Magnetbandgerät hat näherungsweise eine Bandpaßcharakteristik. Gleichspannungen, die bei einer langen Folge von L-Bit oder H-Bit auftreten, lassen sich nicht aufzeichnen. Digitale Daten kann man mit einem Audiokassettengerät nur aufzeichnen, wenn sie codiert sind. Das Codierungsverfahren sollte dabei den Übertragungseigenschaften des Speichermediums gut angepaßt sein. Einige Möglichkeiten der Codierung werden nachfolgend kurz aufgezeigt.

Ein sehr einfaches Verfahren ist die Amplitudenmodulation. Die Datenbit tasten einen NF-Träger von etwa 2 kHz. Bild 8.1 zeigt das modulierte Signal. Die Information gewinnt man durch Gleichrichten und Sieben des wiedergegebenen Signals zurück. Eine praktische Realisierung wurde in [2] veröffentlicht. Nachteil dieses Verfahrens ist die nur geringe Datenübertragungsrate von (meist) nur 110 baud (110 Bit/s). Die Aufzeichnung eines Blocks von 1 kByte Länge dauert etwa 102 s.

Auch die Frequenzmodulation ist zum Codieren geeignet. Beispiele dafür sind das KIM-Interface und der Kansas-City-Standard. Bild 8.2 zeigt den Kansas-City-Standard. Ein L-Bit besteht

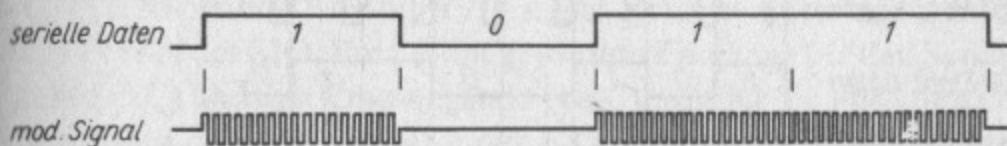


Bild 8.1 Aufzeichnung mittels Amplitudenmodulation

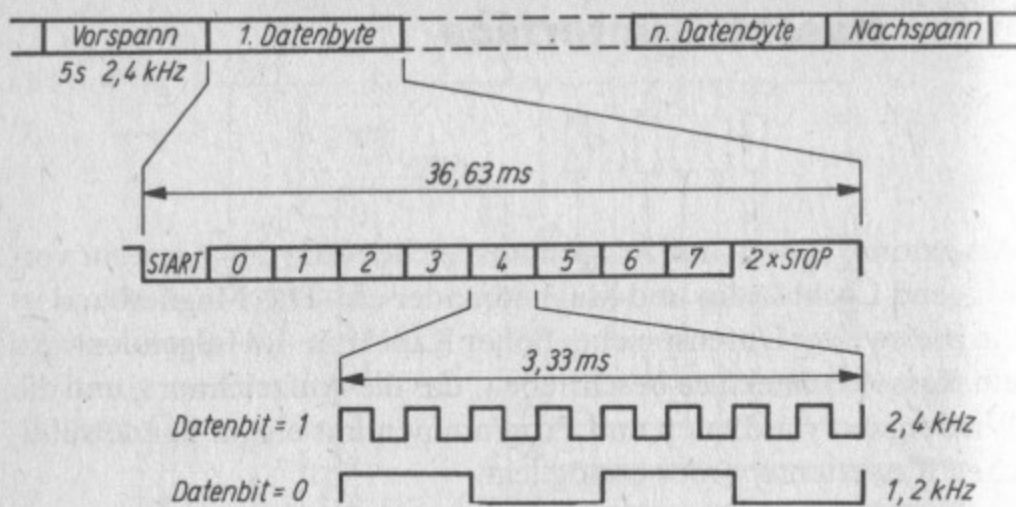


Bild 8.2 Aufzeichnung nach dem Kansas-City-Verfahren

aus 4 Schwingungen (1,2 kHz), ein H-Bit aus 8 Schwingungen (2,4 kHz). Moduliert wird z. B. mit einem FSK-Generator. Die Demodulation kann mit einem Impulsbreitendiskriminator durchgeführt werden. Ein Demodulator wurde in [5] beschrieben. Der Kansas-City-Standard ermöglicht eine recht sichere Datenaufzeichnung. Allerdings beträgt die Übertragungsgeschwindigkeit nur 300 baud, so daß ein Kansas-City-Interface für die Aufzeichnung großer Datenmengen ebenfalls nicht geeignet ist. Das Aufzeichnen eines 1-kByte-Blocks dauert etwa 38 s.

Weitaus höhere Datendichten erreicht man mit den verschiedenen PCM-Verfahren. Im Beispiel wurde Phase-Encoding (Richtungstaktschrift) verwendet. Bild 8.3 zeigt den Code sowie die Modula-

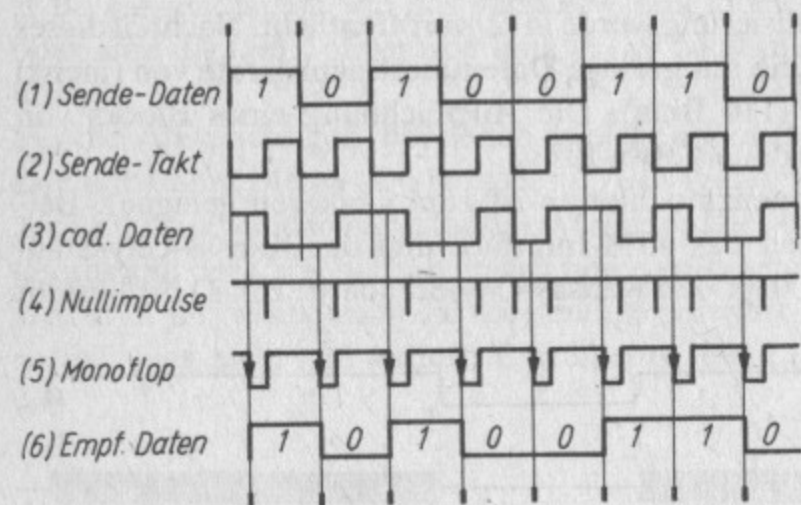


Bild 8.3 Aufzeichnung nach dem Phase-Encoding-Verfahren

tion und Demodulation. Eine Exklusiv-Oder-Verknüpfung von Takt (1) und Daten (2) bewirkt die Modulation. Die Impulsfolge (3) zeigt die codierten Daten: Ein H-Bit wird durch einen H/L-Sprung und ein L-Bit durch einen L/H-Sprung in der Mitte des Bit gekennzeichnet. Bei aufeinanderfolgenden gleichen Bit entstehen redundante Flußwechsel, die bei der Rückgewinnung der Information von den eigentlichen Flußwechseln unterschieden werden müssen. Das wird durch die Aufzeichnung von Synchronzeichen am Beginn eines Datenblocks erreicht. Bei der Wiedergabe gelangt das vom Kassettenrecorder kommende Signal an einen Nulldurchgangsdetektor. Die Nulldurchgänge (4) triggern einen monostabilen Multivibrator, der nach 3/4 der Bitlänge wieder zurückkippt (5). Mit der entstehenden Flanke wird die Information in das Empfangsschieberregister (6) geschoben.

Das Kassetteninterface arbeitet mit einer Bitrate von 4800 baud. 1 kByte läßt sich in nur 2,25 s aufzeichnen. Das von den Autoren realisierte Interface wird mit dem Kassettenrecorder *Geracord GC-6020 portable* betrieben. Dieses Gerät verfügt über eine genau arbeitende Bandzähluhr, die das schnelle Auffinden von Datenaufzeichnungen ermöglicht. Auch andere Geräte wie *MIRA, MK 21* und *ZK 246* wurden an das Interface angeschlossen. Die Fehlerrate war bei Einsatz von hochwertigem Bandmaterial sehr gering. Treten bei der Wiedergabe dennoch Lesefehler auf, so sorgt die in Abschnitt 9. beschriebene Software für eine selbständige Fehlerkorrektur. Somit steht ein zuverlässiger, schneller und billiger Massenspeicher zur Verfügung.

8.2. Realisierung

Mit dem Interface lassen sich unterschiedliche Baudraten realisieren. Der etwas höhere Bauelementeaufwand bringt den zusätzlichen Vorteil, daß das Interface ohne komplizierten Abgleich in Betrieb genommen werden kann. Bild 8.4 zeigt die Schaltung des Kassetteninterface. Zentrale Bausteine sind die SIO U 856 und der CTC U 857 (siehe Abschnitt 7.1.). Das Kassetteninterface benutzt das Port A der SIO. Port A hat getrennte Eingänge für den Sendetakt (TxC) und den Empfangstakt (RxC), was für die Funktion der Schaltung nötig ist. Kanal 2 der CTC erzeugt den Takt. Gatter D1.3 arbeitet als Takttreiber für den Takt T.

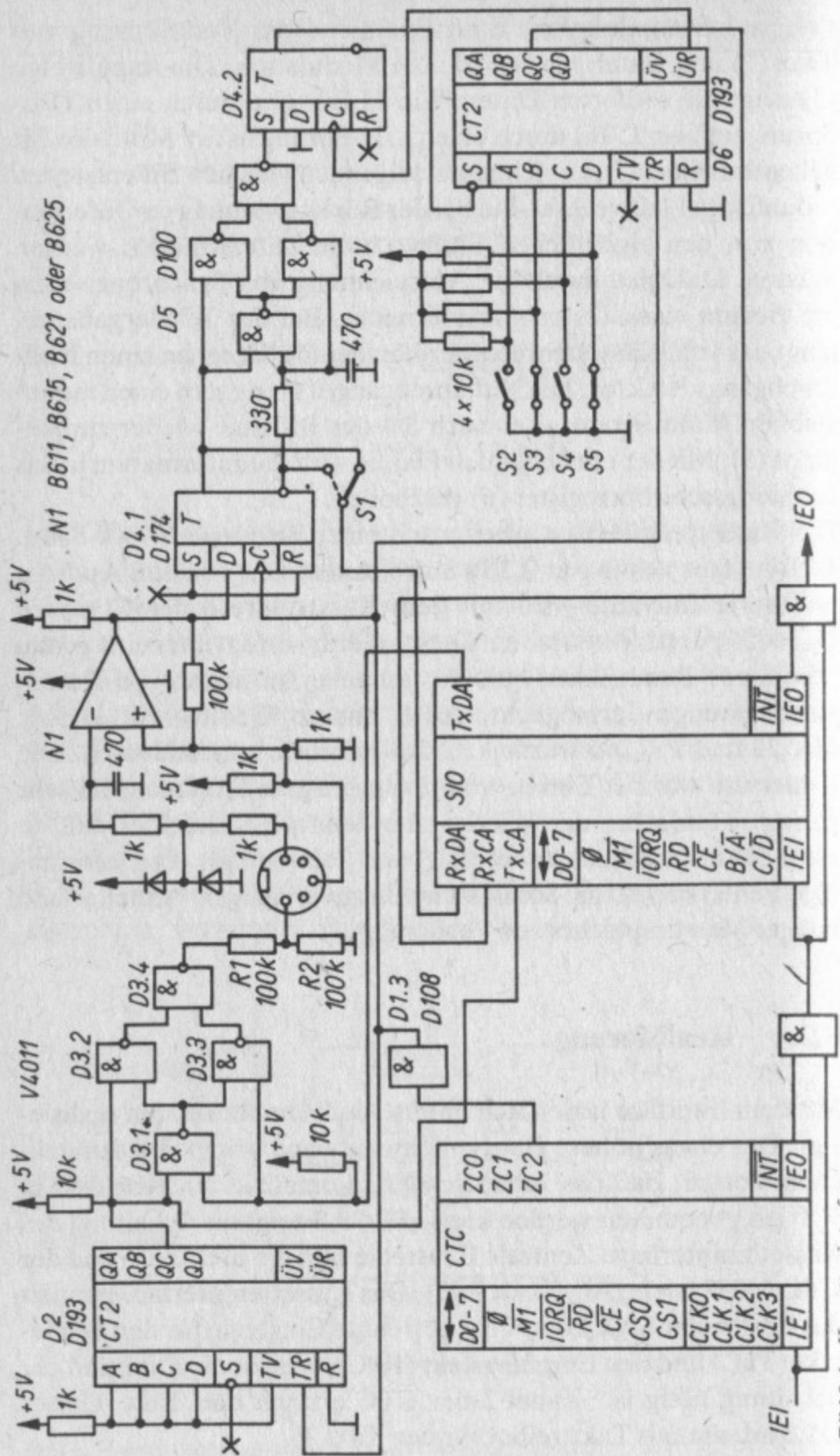


Bild 8.4 Kassetteninterface

Der Modulator besteht aus dem Zähler D2 (D 193) und einem Exklusiv-Oder, das aus den Gattern D3.1 . . . D3.4 gebildet wird. Der Zähler D2 teilt den Takt T durch 8. An QC des Zählers wird der Sendetakt TxCA abgenommen und der SIO zugeführt. Die Baudrate entspricht der Frequenz des Sendetaktes TxCA. Jede fallende Flanke des Sendetaktes schiebt 1 Bit aus dem Senderegister der SIO. Dieses Bit wird mit dem Takt TxCA exklusiv-oder-verknüpft. Am Ausgang von Gatter D3.4 steht das codierte Signal zur Verfügung. Über R1 und R2 gelangt es an den Kassettenrecorder. Der Demodulator ist etwas aufwendiger. Bild 8.5 zeigt das Impulsdiagramm.

Das Eingangssignal wird dem Operationsverstärker N1 zugeführt. An seinem Ausgang liegt das begrenzte Wiedergabesignal, welches durch das D-Flip-Flop D4.1 mit dem Takt T synchronisiert wird. Die synchronisierten Daten werden über den Schalter S1 an den Eingang RxDA der SIO gelegt. Der Schalter S1 legt die Phasenlage der Daten fest (je nach Kassettenrecorder). Bei falscher Phasenlage werden invertierte Daten gelesen. Die Stellung des Schalters muß man bei der Inbetriebnahme durch Versuch ermitteln. Den Empfangstakt RxCA erzeugen die Schaltkreise D4.2, D5 und D6. Der Zähler D6 und das D-Flip-Flop D4.2 bilden einen monostabilen Multivibrator. Die Gatter D5.1 . . . D5.4 erzeugen bei jedem Nulldurchgang des Eingangssignals einen kurzen L-Impuls. Dieser Impuls triggert den monostabilen Multivibrator. An Hand von Bild 8.6 soll die Funktionsweise erläutert werden. Am Ausgang Q des Flip-Flop D4.2 sei L-Pegel. Dieser Eingang ist mit dem Ladeeingang des Zählers D6 verbunden. Der an den Dateneingängen (mit den DIL-Schaltern S2 . . . S5) vorgewählte Wert wird in den Zähler übernommen. Der Nulldurchgangsimpuls setzt das Flip-Flop D4.2, Ausgang Q wird H. Mit dem nächsten Taktim-

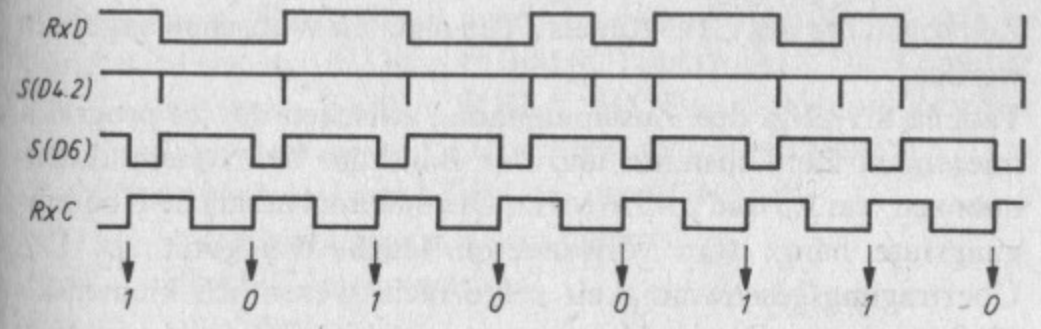
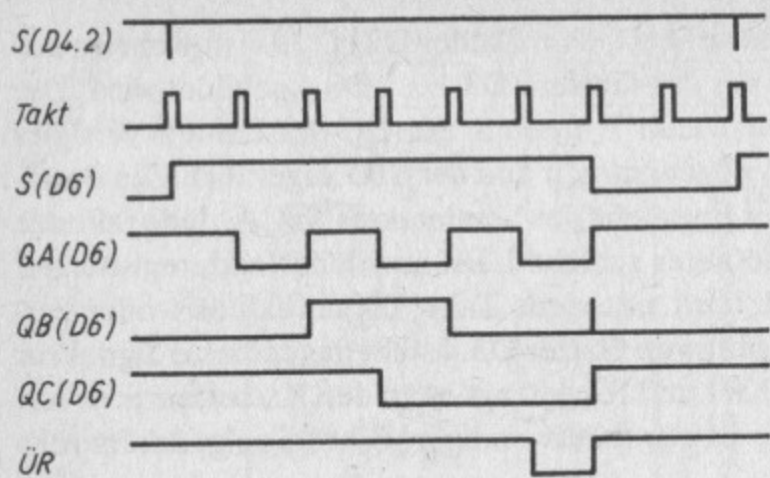


Bild 8.5 Impulsdiagramm des Demodulators



(nicht maßstäblich)

Bild 8.6 Impulsdiagramm des Monoflop

puls T beginnt der Zähler rückwärts zu zählen. Bei Erreichen des Zählerstandes „0“ entsteht ein Übertragsimpuls, der den Ausgang Q des Flip-Flop D4.2 auf L schaltet. Somit ist die Ausgangsstellung wieder erreicht. Am Ausgang Q des Flip-Flop entsteht die schon in Bild 8.3 dargestellte Impulsfolge (5). Den Empfangstakt RxCA für die SIO entnimmt man QC des Zählers D6. Mit jeder steigenden Flanke des Empfangstaktes übernimmt die SIO die an RxDA liegende Information in das Empfangsschieberegister. Die Lage dieser Flanke und damit die Impulsdauer des Monoflop werden mit den DIL-Schaltern S2 . . . S5 auf 3/4 der Bitlänge eingestellt. Im Mustergerät lädt sich der Zähler mit dem Wert 5.

Die Inbetriebnahme ist mit der in Abschnitt 9. beschriebenen Software nicht schwierig. Es wird ein kurzer Datenblock aufgezeichnet. Werden die Daten bei Wiedergabe nicht gelesen, muß man die Stellung von Schalter S1 (Phasenlage) ändern. Der optimale Abtastzeitpunkt läßt sich dann mit S2 . . . S5 einstellen (Richtwert: 5). Die Übertragungsgeschwindigkeit kann durch Verändern der Zeitkonstante des CTC-Kanals 2 den eigenen Wünschen angepaßt werden.

Tabelle 8.1 zeigt den Zusammenhang zwischen der zu programmierenden Zeitkonstante und der Baudrate bei Systemtaktfrequenzen von 2,5 und 2,4596 MHz. Die maximal mögliche Übertragungsgeschwindigkeit hängt vom verwendeten Magnetbandgerät ab. Die Übertragungsgeschwindigkeit sollte nicht wesentlich kleiner als 1200 baud sein. Die im Mustergerät gewählte Baudrate von 4800 baud stellt einen guten Kompromiß zwischen möglichst geringer

Tabelle 8.1. Zusammenhang: CTC-Zeitkonstante-Baudrate

TC	Baudrate	
	bei $f_c = 2,5$ MHz	bei $f_c = 2,4576$ MHz
1	19531	19200
2	9766	9600
3	6510	6400
4	4883	4800
5	3906	3840
6	3255	3200
8	2441	2400
12	1628	1600
16	1220	1200
32	610	600

Fehlerzahl und möglichst hoher Geschwindigkeit dar (1 kByte wird in 2,25 s aufgezeichnet). Hochwertiges Bandmaterial ist Grundvoraussetzung für die erreichten Werte.

Wie bereits erwähnt, werden eventuell auftretende Lesefehler selbständig korrigiert. Ermöglicht wird das durch eine geeignete Formatierung der Daten. Wie man die Daten formatiert, ist in Bild 8.7 dargestellt. Die aufzuzeichnenden Daten werden in einzelne Blöcke zu je 128 Byte (oder weniger, falls die Anzahl der zu übertragenden Byte < 128 beträgt) aufgeteilt. Ein Datenblock beginnt mit 6 Synchronzeichen. Es folgt das Zeichen GS (1 DH), das den Datenblockbeginn kennzeichnet. Das nächste Byte enthält die Anzahl der folgenden Datenbyte. Nach der Übertragung der n Datenbyte folgt das Prüfsummenbyte (die Übertragung von nur einem Prüfsummenbyte hat sich als ausreichend erwiesen).

Zu Beginn jeder Datei wird ein Block übertragen, der alle wichtigen Datenparameter enthält (Kopf). Der Kopf beginnt mit 20 Synchronzeichen. NL (01EH) kennzeichnet den Beginn der Aufzeichnung, wird also nur in Datei 1 (Bild 8.7) übertragen. Die Kennung des Kopfes ist das Zeichen SOH (01H). Die nächsten 8 Zeichen enthalten den Dateinamen. Die weiteren Parameter sind:

- TT Dateityp (maximal 2 Zeichen), zum Beispiel:
 - 'P' Maschinenprogramm mit Autostart
 - 'BA' BASIC-Programme
 - 'A' Textdatei
 - 'BI' Sonstige

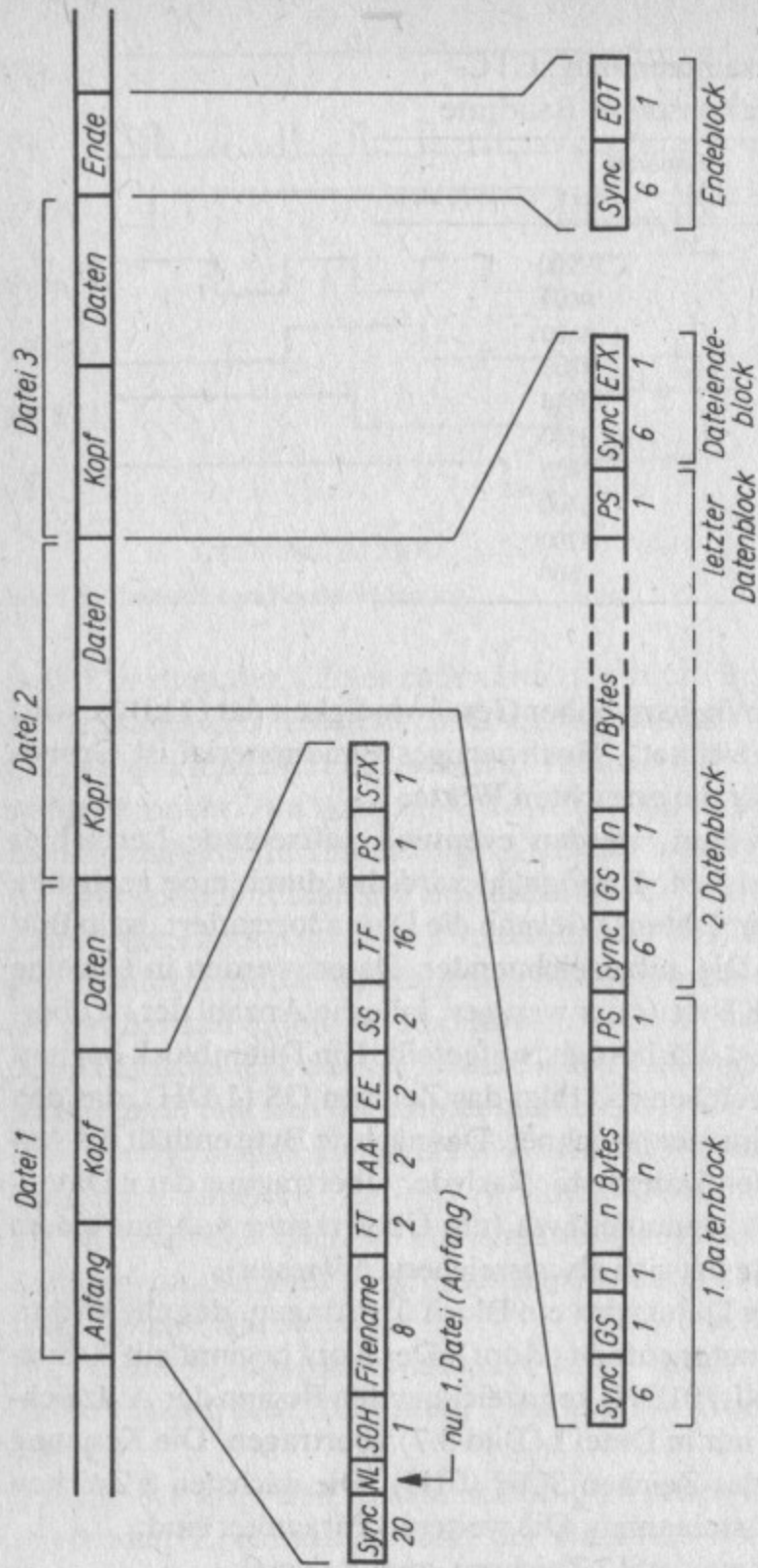


Bild 8.7 Verwendetes Dateiformat für Kassettenaufzeichnung

- AA Anfangsadresse (2 Byte)
- EE Endadresse (2 Byte)
- SS Startadresse für Autostart
- TF Textfeld (16 Zeichen)

Im Textfeld kann beispielsweise das Erstellungsdatum oder der Bearbeitungszustand eingetragen werden. Anschließend folgt die Prüfsumme des Kopfes. STX (02H) beendet den Kopfblock, zugleich zeigt STX den Beginn der Datenblöcke an. Sind alle Datenblöcke übertragen, erscheinen 6 Synchronzeichen und ETX (03H). Dieser kurze Block beendet die Datei. Die gesamte Datei wird danach noch 2mal übertragen. Die Aufzeichnung endet mit dem Endblock, der aus 6 Synchronzeichen und dem Zeichen EOT (04H) besteht.

Nicht nur die Hardware, auch die Software ist für die Wiedergabe umfangreicher als für den Aufzeichnungsvorgang. Eine Datei wird eingelesen, wenn der Dateiname der Aufzeichnung mit dem eingegebenen Dateinamen übereinstimmt. Dabei sind alle 8 Zeichen des Namens signifikant. Bei der Wiedergabe werden die einzelnen Datenblöcke mitgezählt. Traten keine Lesefehler auf, so ist nach Erreichen des Dateiendblocks (ETX) die Wiedergabe beendet. Lagern jedoch Lesefehler vor, so wird die nächste Datei gelesen und alle mit fehlerhafter Prüfsumme „gemerkten“ Datenblöcke werden neu geladen. Bei mangelhaftem Bandmaterial oder starken Gleichlaufschwankungen des Kassettensrecorders kann die Anzahl der Lesefehler so groß werden, daß eine Korrektur durch Neuladen der fehlerhaften Datenblöcke nicht mehr möglich ist. In diesem Fall lädt man die gesamte Datei neu. Da jede Datei 3mal aufgezeichnet wurde, ist in der Regel (auch bei weiteren Fehlern) das Laden (und Korrigieren) der Datei möglich. Wird beim Lesen der EOT-Block erreicht, so war keine fehlerfreie Wiedergabe möglich.

Die Software (Abschnitt 9.) organisiert den Datenverkehr. Details sind der Softwarebeschreibung und dem Listing zu entnehmen.

9. Das Monitorprogramm

Bisher wurde vor allen Dingen die Computerhardware beschrieben. Damit der Computer sinnvolle Funktionen ausführen kann, muß ihm ein geeignetes Programm eingegeben werden. Das in diesem Abschnitt beschriebene Monitorprogramm soll den Computer „zum Leben erwecken“.

Das Monitorprogramm benötigt als minimale Hardwarevoraussetzung die CPU-Karte (mit 3-kByte-ROM und 1-kByte-RAM), die Tastatur und die Bildschirmsteuerung. Wenn die I/O-Karte eingesetzt wird, sind alle vorhandenen Monitorkommandos ausführbar. Das Monitorprogramm wurde im wesentlichen unter folgenden Gesichtspunkten geschrieben:

- einfache Bedienung, Fehlertoleranz;
- einfache Ergänzung durch zusätzliche Programmmodule;
- einfache Arbeit mit externen Speichermedien;
- Schnittstellen zu höheren Programmiersprachen (Assembler, BASIC).

Der Monitor wird als ausführliches Assemblerlisting veröffentlicht. Dadurch lernt der Anwender zusätzlich die Funktionsweise kennen, er kann Änderungen einbauen, (zum Beispiel andere Adressen) oder einzelne Teile des Programms anderweitig verwenden.

9.1. Beschreibung der Monitorkommandos

Das Monitorprogramm erwartet nach der Ausgabe des Promptzeichens (&) die Eingabe einer Kommandozeile. Eine Kommandozeile besteht aus dem Kommando selbst und den dazugehörigen Angaben. Die einzelnen Angaben müssen durch ein Komma oder mindestens ein Leerzeichen getrennt werden. Die Eingabe wird durch RETURN (CR) abgeschlossen, der Computer führt nun das eingegebene Kommando aus.

Im folgenden sind die einzelnen Kommandos kurz beschrieben. Den Kommandonamen braucht man nicht auszuschreiben. So kann beispielsweise an Stelle von MEMORY auch MEMOR,

MEMO, MEM, ME oder M geschrieben werden. Alle angegebenen Zeichen sind signifikant. Beginnen verschiedene Kommandonamen mit dem gleichen Buchstaben und ist nur ein Buchstabe angegeben, so wird das zuerst gefundene Kommando angesprungen. M entspricht MEMORY, aber nicht MOVE, für MOVE muß mindestens MO eingegeben werden. Klarheit über die Reihenfolge der Kommandos schafft das Kommando HELP.

- HELP Anzeige aller im Speicher vorhandenen Kommandos HELP CR

Sowohl die Kommandos des Monitors als auch die vom Anwender definierten werden gelistet. Im gesamten Speicher wird nach dem Kommandorahmen (0EDH, 0FFH, Kommandozeichenkette, 00H) gesucht, und die Kommandozeichenketten werden ausgegeben (siehe auch Abschnitt 9.2.).

- MEMORY Anzeigen und Modifizieren von Speicherbereichen/
MEMORY [Option] Startadresse CR

Es wird ein Speicherbereich ab Startadresse und 1/4 kByte Länge in hexadezimaler Form und in Textdarstellung ausgegeben. Bei der Textdarstellung erscheinen nicht darstellbare Codes als Punkt. Bei Angabe der Option @ A beginnt jede neue Zeile zusätzlich mit der entsprechenden Adresse. Dabei wird die Textdarstellung unterdrückt.

Mit dem MEMORY-Befehl lassen sich auch Speicherzellen modifizieren. Ein spezieller Cursor zeigt auf die zu modifizierende Speicherzelle. Die Cursorposition und damit die Adresse können mit den 4 Cursortasten (links, rechts, hoch, tief) verändert werden. Die Eingabe erfolgt hexadezimal (Mode H) oder als Textzeichen (Mode A). Zum Umschalten in den jeweils anderen Modus benutzt man die Escape-Taste (ESC).

Eine neue Startadresse läßt sich nach Drücken der Taste Q (Quit) eingeben. Folgt an Stelle einer Adresse ein weiteres Q, so wird das Programm MEMORY nach Drücken der RETURN-Taste verlassen.

- MOVE Umladen von Speicherbereichen
MOVE, Startadresse, Endadresse, Zieladresse CR

Es wird der Bereich von Startadresse bis Endadresse nach Zieladresse kopiert. Dabei können sich die Speicherbereiche auch überlappen. Nach Ausführung des MOVE-Befehls wartet das Programm auf eine Tastatureingabe: Q (Quit) oder RETURN (oder irgendeine andere Taste). Im letzteren Fall springt das Programm

nach MEMORY und zeigt den Speicherbereich ab Startadresse an.

- FILL Füllen eines Speicherbereichs mit einer Konstanten
FILL, Startadresse, Endadresse [Byte] CR

Das angegebene Byte füllt den entsprechenden Speicherbereich. Fehlt das Füllbyte, so wird das Byte als 00H angenommen (schnelles Löschen eines Speicherbereichs).

- IN, OUT Bedienen von Ports
IN, Portadresse CR
OUT, Portadresse, Datenbyte CR

Mit dem IN-Befehl lassen sich Ports direkt lesen. Das gelesene Byte wird auf dem Consolkanal (Bildschirm) ausgegeben.

Mit dem OUT-Befehl können zum Beispiel Ports initialisiert werden.

Beispiel: OUT 0 CR
OUT 0 CR entspricht dem Kommando OFF

- OFF Ausschalten des Computers
OFF CR

Nach etwa 20 s schaltet sich der Computer ab. Den Abschaltvorgang kann man durch RESET, NMI oder durch die Sequenz IN 0 CR, IN 0 CR abbrechen.

- SAVE Abspeichern auf Magnetband
SAVE, Filename, Filetyp, Anfangsadr., Endadr.
[Eintrittspunkt] [Textfeld] CR

SAVE zeichnet den Speicherbereich von Anfangs- bis Endadresse auf Magnetband auf. Der Name des Programms kann beliebig lang sein, es werden aber nur die ersten 8 Zeichen als Filename verwendet. Durch Angabe eines Filetyps läßt sich die Datei besonders kennzeichnen, beispielsweise können BASIC-Programme durch @ BA oder Textdateien durch @ A gekennzeichnet werden. Bei der Angabe eines Filetyps sind 2 Zeichen signifikant. Ist kein Filetyp angegeben, so wird vom SAVE-Programm automatisch P eingesetzt. P kennzeichnet die Datei als lauffähiges Maschinenprogramm. Dateien vom Typ P erfordern noch einen weiteren Parameter, nämlich den Eintrittspunkt des Programms. Bei nicht gegebenem Eintrittspunkt wird vom SAVE-Programm automatisch der Restart-Eintrittspunkt des Monitors eingetragen. Schließlich können im Textfeld noch zusätzliche Informationen, z. B. das Erstellungsdatum, folgen. Es werden 16 Textzeichen (nach;) aufgezeichnet. Dieses Textfeld werten die hier beschriebenen Monitor-kommandos nicht aus. Der Anwender kann den Monitor aber

durch weitere Kommandos ergänzen, die bei Bedarf dieses Textfeld auch auswerten (von den Autoren wurde ein Programm CAT geschrieben, das alle auf Band befindlichen Dateien mit ihren Eigenschaften einschließlich Textfeld auflistet).

- LOAD Laden von Magnetband
LOAD, Filename [Option] [Offset] CR

Ein mit SAVE aufgezeichneter Speicherbereich wird wieder eingelesen. Wenn der angegebene Filename mit dem aufgezeichneten Namen übereinstimmt, wird die Datei geladen. Man erkennt das an der Aufschrift „FILE FOUND“. Nach dem Laden, falls es sich um ein abarbeitbares Maschinenprogramm (Filetyp P) handelt, startet das Programm automatisch (Eintrittsadresse), es sei denn, es wurde die Option @ Q angegeben. @ Q unterdrückt den Autostart. Bei Bedarf kann noch ein Offsetwert angegeben werden. Der Wert addiert sich zur Ladeadresse und ermöglicht das Laden auf beliebige Speicherbereiche. Dabei ist natürlich der eventuelle Autostart unterdrückt. Nach dem Laden werden die eventuell aufgetretenen Lesefehler angezeigt. Diese Fehler wurden aber vom LOAD-Programm korrigiert. Nur wenn „BAD FILE“ erscheint, war keine Korrektur möglich, d. h., die im RAM befindlichen Daten sind fehlerhaft.

- READ Lochstreifen lesen
READ [Option] [Offset] CR

Dieser Befehl ermöglicht das Lesen von MC-Lochstreifen (Intel-Hex-Format) vom Lochstreifenlesekanal RI. Wird die Option @ A angegeben, dann stoppt das Programm nach Einlesen 1. Adresse, gibt diese Adresse aus und wartet auf eine Eingabe. Drückt man ohne Eingabe RETURN, so wird, beginnend bei der angezeigten Startadresse, eingelesen. Es kann aber auch eine andere Adresse eingegeben werden. Nach dem Drücken der RETURN-Taste wird dann das Einlesen, beginnend bei der neuen Adresse, fortgesetzt. Eine andere Möglichkeit, den Lochstreifen „verschoben“ einzulesen, besteht darin, daß man einen Offsetwert angibt.

Beim Auftreten eines Lesefehlers stoppt das Programm. Mit Hilfe der Taste Q kann nun das READ-Programm verlassen werden. Es ist aber auch möglich, den fehlerhaften Block noch einmal einzulesen (Lochstreifen zum Blockanfang [:] zurücktransportieren, dann RETURN).

- NMI Vektor eintragen
NMI CR

Es wird auf die Adresse 066H ein Rücksprung zum Monitor eingetragen. Den nichtmaskierten Interrupt kann man dann als „Notbremse“ nutzen. Die CPU-Register werden gerettet. Ein ununterbrochenes Programm wird durch GO PC mit den geretteten Registerwerten wieder gestartet. Dabei ist zu beachten, daß sich nur Programme wieder starten lassen, die nicht den Systemstack verwenden.

- GO Start von Programmen
GO, Startadresse [Breakpoint] CR
GO PC CR

Mit GO adr. läßt sich ein Programm starten. Der Stackpointer wird auf die Adresse USTCK (Userstack) gesetzt. Wenn das Anwenderprogramm diesen Stack benutzt, kann es durch RET (0C9H) in den Monitor zurückkehren. Es ist möglich, zusätzlich den Breakpoint zu setzen, wenn das Anwenderprogramm zu Testzwecken unterbrochen werden soll. Der Breakpoint muß auf den Anfang eines Befehls gesetzt werden. Das GO-Programm rettet das dort stehende Byte und trägt 0FFH (Restart 38 H) ein. Bei Erreichen des Breakpoints wird das ursprüngliche Byte wieder zurückgeladen, anschließend werden alle CPU-Register angezeigt. Sie lassen sich nun gegebenenfalls verändern (REG). Das Anwenderprogramm wird durch GO PC fortgesetzt. Bei der Anwendung von GO PC wird der RAM-Bereich der Userregister in die CPU kopiert (siehe REG). Jedes Programm kann statt mit GO adr. auch mit GO PC gestartet werden (SP berücksichtigen!). GO ohne weitere Angaben führt keinen Programmstart aus, sondern eignet sich dazu, einen nicht erreichten Breakpoint zu löschen.

- REG Anzeigen und Modifizieren der CPU-Register
REG [reg Byte [reg Byte . . .]] CR

Das Kommando REG ohne weitere Angaben zeigt die CPU-Register an. Die Register verändert man durch die Eingabe des Registers und des gewünschten Wertes. Beispielweise bewirkt das Kommando

REG A0, SP1000, PC100, H'0

die Belegung der Register A, H', SP und PC mit den angegebenen Werten. Genaugenommen werden die entsprechenden RAM-Zellen (Userregister UREG) verändert.

- CHECKSUM Prüfsummenberechnung
CHECKSUM, Startadr., Endadr. CR

Dieses Kommando berechnet von dem angegebenen Speicherbereich eine 2-Byte-Prüfsumme.

- = Adreßrechnung
= , x, y, CR

Diese Eingabe bewirkt folgende Berechnung und Ausgabe:

SUM : x + y DIF : x - y REL: **

Dabei bedeutet REL die relative Distanz bei Sprungbefehlen (JR, DJNZ usw.). Ist die Distanz zu groß, so werden an Stelle eines Byte 2 Sterne ausgegeben.

- FIND Suchen von Bytes oder Zeichenketten
FIND, Startadr., Endadr., Byte 1, Byte 2 . . . Byte n
CR FIND, A, Startadr., Endadr., Zeichenkette CR

FIND ermöglicht das Suchen nach einer Bytefolge oder nach einer Zeichenkette in dem Speicherbereich von Start- bis Endadresse. Auf dem Bildschirm werden alle Adressen ausgegeben, auf denen die Zeichenkette oder Bytefolge gefunden wurden.

- SIZE Speichertest
SIZE CR

SIZE führt einen einfachen, nicht zerstörenden RAM-Test durch, gibt den höchsten als RAM verfügbaren Speicherplatz aus und erwartet eine Eingabe. Die ermittelte Adresse oder eine jetzt einzugebende Adresse wird nach RETURN in die Zelle MSIZE geladen. Der Monitor selbst benötigt die Angabe über den höchsten Speicherplatz nicht. Aber verschiedene Programme (z. B. BASIC-Interpreter) brauchen in der Initialisierungsphase die Angabe über den höchsten zur Verfügung stehenden Speicherplatz, der entweder das RAM-Ende oder eine andere Adresse sein kann.

- EOF Dateiendezeichen definieren
EOF CR

Verschiedene Programme (BASIC-Interpreter, Assembler usw.) benötigen beim Einlesen von Daten und Programmen ein End-of-File-Zeichen. Fehlt dieses Zeichen auf dem Datenträger (z. B. Lochstreifen), so läßt sich die Lesefunktion dieser Programme nicht beenden. Um dieses Problem zu lösen, wurde der Monitor so ausgelegt, daß nach Drücken von CTRL C EOF-Zeichen gesendet werden. Da die verschiedenen Programme unterschiedliche EOF-Zeichen voraussetzen, wurde dieses Kommando geschaffen. Es

zeigt das derzeit gültige EOF-Zeichen an. Das Zeichen kann wieder übernommen oder durch Neueingabe verändert werden.

- ASN Gerätezuweisung
- ASNRI = RI CR
- ASNRI = CRI CR
- ASNRI = USR, Adresse CR

Mit diesem Kommando kann man die jeweilige Gerätezuweisung verändern (siehe auch Abschnitt 9.2.). Hat beispielsweise ein Anwenderprogramm nur eine Lochstreifenschnittstelle, so lassen sich die Daten bei der Zuweisung RI = RI vom Lochstreifenleser oder bei RI = CRI vom Kassettenmagnetbandgerät einlesen.

Im vorliegenden Monitor sind nur wenige Zuweisungen fest vorhanden. Eine Erweiterung ist leicht möglich (siehe Abschnitt 9.2.). Man muß allerdings beim Verändern der Consolzuweisung aufpassen, sonst wird das Bedienen des Computers in Frage gestellt (Die Gerätezuweisungen wirken auch auf den Monitor!). In solchen Fällen hilft dann nur noch RESET.

9.2. Schnittstellen und Erweiterbarkeit

Der Monitor hat ab Adresse 0F000H einige Eintrittspunkte, die im folgenden kurz erläutert werden:

- JMP BEGIN Neustart des Monitors
- JMP CI Consoleingabe (Tastatur, Zeichen in A)
- JMP RI Lesekanal (Lochstreifenleser, Zeichen in A)
- JMP COE Consolausgabe (Bildschirm, Zeichen C)
- JMP POE Punchkanal (Lochstreifenstanzer, Zeichen in C)
- JMP LOE Listkanal (Drucker, Zeichen in C)
- JMP CSTS Consolstatus (A = 0 keine Taste, A = 0FFH gedrückt)
- JMP CRI Kassettenlesekanal (Zeichen in A)
- JMP CPOE Kassettenpunchkanal (Zeichen in C)
- JMP MEMSI höchster Speicherplatz (A = Low, B = High-Teil der Adresse)
- JMP MAIN Restart des Monitors
- JMP EXT Aufruf von Monitorunterprogrammen (C = Nr. des Programms entsprechend ETAB)
- DA FINA Beginn RAM-Bereich des Monitors.

Ein Teil dieser Sprünge führt in den RAM-Bereich des Monitors.

Von dort wird dann zur eigentlichen Geräteroutine gesprungen. Dadurch ist eine einfache Gerätezuweisung möglich. Beim Neustart benutzt man folgende Zuordnung (ATAB):

Consoleingabe	CI	—	CI	Tastatur
Lesekanal	RI	—	RI	Lochstreifenleser
Consolausgabe	COE	—	CO	Bildschirm
Punchkanal	POE	—	CPO	Kassettenausgabe
Listkanal	LOE	—	CO	Bildschirm
Kassettenlesekanal	CRI	—	CRI	Kassetteneingabe
Kassettenpunchkanal	CPO	—	CPO	Kassettenausgabe

Bei Veränderung durch ASN wird auf den entsprechenden RAM-Zellen nur die jeweilige Gerätetreiberadresse eingetragen. Weitere Einsprungstellen (wenn Zeichen im Akkumulator übergeben werden sollen) sind:

- FINA + 0B6H Listkanal
- FINA + 0BAH Punchkanal
- FINA + 0BEH Kassettenpunchkanal
- FINA + 0C2H Consolausgabekanal

Wie man die Schnittstelle benutzt und den Monitor erweitert, belegen 3 einfache Beispiele.

Im 1. Fall soll eine Schreibmaschinenfunktion realisiert werden. Der Aufruf erfolgt durch Eingabe Typ und RETURN

- | | | |
|-------|-------------|------------------------------|
| | DB 0EDH | ; Kennzeichnungsrahmen |
| | DB 0FFH | ; wird von MAIN gesucht |
| | DB 'TYP' | ; Kommandoname |
| | DB 00H | ; Ende des Rahmens |
| LOOP: | CALL 0F003H | ; Zeichen von Tastatur holen |
| | CMP 03H | ; CTRL C gedrückt? |
| | Rz | ; Ja, zum Monitor zurück |
| | LD C, A | |
| | CALL 0F009H | ; Zeichen ausgeben |
| | JR LOOP - # | |

Der Monitor (Routine MAIN) sucht den Kennzeichnungsrahmen. Wenn die Bytefolge 0EDH und 0FFH gefunden wurde, wird die folgende Zeichenkette mit der im Textpuffer abgelegten Kommandozeichenkette verglichen. Sind beide Zeichenketten identisch und folgt im Speicher das Byte 00H, so wird die dem Rahmenendebite folgende Adresse von MAIN aus angesprungen. In diesem Beispiel handelt es sich um die Marke LOOP. Dabei ist es gleichgültig, wo die Folge 0EDH, 0FFH, Zeichenkette, 00H im Speicher

steht. Der gesamte Speicher wird nach dem eingegebenen Kommando durchsucht. Die Routine MAIN beginnt die Suche ab Marke START - 1000H (0E000H). Das bedeutet, daß beispielsweise eine veränderte MEMORY-Routine ab Adresse 0E000H abgelegt und bei Eingabe des Befehls MEMORY aufgerufen werden könnte. Die im Monitor-EPROM enthaltene Routine MEMORY wird in diesem Fall nicht mehr gefunden. Jedes Anwenderprogramm kann man mit dem (im Listing HEAD genannten) Rahmen, dem Namen und anschließenden Byte 00H versehen. Auf diese einfache Art und Weise ist es möglich, jedes Programm mit seinem Namen vom Monitor aus aufzurufen. Die Rückkehr zum Monitor erfolgt durch einen RET-Befehl.

Im 2. Beispiel sollen alle Zeichen, die über den Listkanal an den Monitor gelangen, auf Magnetband aufgezeichnet werden. Benötigt wird dazu eine neue Gerätezuweisung LO=CPO.

```
DB 0EDH
DB 0FFH
DB 'ASNLO=CPO'
DB 00H
LD HL, CPO1           ; Adresse Kassettentreiber
LD (LO+1), HL        ; ins RAM laden
RET                   ; zurück zum Monitor
```

Im 3. Beispiel soll der Hexadezimalcode eines von der Tastatur eingegebenen Zeichens auf dem Bildschirm ausgegeben werden.

```
ENTRY: CALL 0F003H ; Zeichen von Tastatur holen
        PUSH AF    ; retten
        LDC, A
        CALL 0F009H ; Zeichen ausgeben
        LDC, 5     ; Code für PRINT
        CALL 0F021H ; EXT anspringen
        DB '='     ; Zeichenkette ausgeben
        DB 00H    ; bis 0
        POP AF    ; Zeichen zurück
        LDC, 3    ; Code für HEX-Ausgabe
        CALL 0F021H ; EXT anspringen
        RET       ; zurück zum Monitor
```

Dieses kurze Programm zeigt, wie man den Programmteil EXT anwendet. Die Codes enthält Tabelle ETAB (siehe Listing). Auch die Funktionsweise der einzelnen Unterprogramme ist dem Listing zu entnehmen.

Die 3 Beispielprogramme sollten die Anwendung der Monitorroutinen demonstrieren. Weitere Einzelheiten findet man im ausführlichen Assemblerlisting.

9.3. Das Programm

NOTA: ENTRE ESTA PAGINA Y LA SIGUIENTE, VA EL LISTADO FUENTE DEL MONITOR DEL U880, QUE NO HE INCLUIDO EN EL PDF, PERO SI EN FORMATO "TXT" EN FICHERO APARTE

```
;TASTATUR
;LOCHBANDLESER
;BILDSCHIRM
;LOCHBANDSTANZER
```

EBAS-88: R

PASS: P	EBAS-88	26.1.84	LOC	CODE	LINE	SOURCE
					00001	PN MON
					00002	
					00003	*****
					00004	* MONITORPROGRAMM
					00005	* * * * *
					00006	* V2.1 20.12.83
					00007	* ROM-TEIL
					00008	*****
					00009	
					00010	
					00011	ORG 0F000H
					00012	
					00013	;SPRUNGTABELLE:
					00014	
					00015	START: JMP BEGIN
					00016	JMP CI
					00017	JMP RI
					00018	JMP COE
					00019	JMP POE
F000	C326F0					
F003	C3CBFF					
F006	C3C5FF					
F009	C3C1FF					
F00C	C3B9FF					

; DARSTELLMODE BEI MEMORY

MEMMO: EQU DATA
SYSBG: EQU CRTBG

01820
01821
01822
01823
01824

END

FFD1

NEXT: E

PASS: E

**HASTA AQUI LLEGABA EL
LISTADO FUENTE COMENTADO
EN LA PAGINA ANTERIOR**

Anmerkungen zum Programm

Die in Spalte Code nicht ausgedruckten Byte können der folgenden Aufstellung entnommen werden.

FO38 20 48 2E 4D 4F 4E 20 56 32 2E 31 20 31 32 2F 38 33
FO5C 49 4C 4C 45 47 41 4C 20 43 4F 4D 4D 41 4E 44
F1C2 4D 45 4D 4F 52 59
F200 20 20 43 55 52 53 4F 52 3A
F237 20 20
F309 52 41 4D 20 45 52 52 4F 52 20
F31A 4D 4F 56 45
F369 4C 4F 41 44
F3CD 46 49 4C 45 20 46 4F 55 4E 44 20
F490 20 42 41 44 20 46 49 4C 45 20
F4A4 20 45 52 52 4F 52 53
F4D5 53 41 56 45
F599 49 4E
F5A7 4F 55 54
F5B7 46 49 4C 4C
F5EE 53 49 5A 45
F5F6 53 49 5A 45 3A
F605 20 52 41 4D 54 4F 50 3A
F621 4E 4D 49
F642 42 52 45 41 4B 20 41 54 20
F696 47 4F
F71C 20 41 46 20 20 20 42 43 20 20 20 44 45 20 20 20 48
F72D 4C 20 20 20 41 46
F734 20 20 42 43
F739 20 20 44 45
E73E 20 20 48 4C
F743 20 20 49 58 20 20 20 49 59 20 20 49 20 20 20 20 53
F754 50 20 20 20 20 50 43
F780 20 20
F79B 52 45 47
F809 46 41 43 42 45 44 4C 48
F813 48 45 4C 50
F826 20 20 20
F852 52 45 41 44
F8AE 43 48 45 43 4B 53 55 4D 20 45 52 52 4F 52 20
F8E9 43 48 45 43 4B 53 55 4D
F91B 3D

F92C 53 55 4D 3A
F93B 20 20 44 49 46 3A
F94F 20 20 52 45 4C 3A
F964 2A 2A
F96A 46 49 4E 44
F9B9 20 20
F9D7 45 4F 46
F9EC 4F 46 46
F9F7 41 53 4E 52 49 3D 52 49
FA07 41 53 4E 52 49 3D 43 52 49
FA1A 41 53 4E 52 49 3D 55 53 52

10. Hardwareaufbau und Inbetriebnahme

10.1. Stromversorgung

Bild 10.1 zeigt die Schaltung für ein Netzteil, das alle für den Mikrocomputer benötigten Betriebsspannungen liefert.

Zusätzlich wird die Gleichspannung -12 V bereitgestellt, die für den Betrieb analoger Schaltungselemente (vor allem Operationsverstärker) erforderlich ist. Die Einzelschaltungen wurden im wesentlichen [12] entnommen und erfüllen hinsichtlich Strombelastbarkeit und Konstanz der Ausgangsspannungen alle Anforderungen der vorliegenden Anwendung.

Wie Bild 10.1 zeigt, wurden die Regelstrecken für die Ausgangsgleichspannungen auf der Basis integrierter Spannungsregler aufgebaut. Die Betriebsspannung wird damit mit minimalem Aufwand erzeugt. Für das Regelteil der bezüglich des Ausgangsstroms am meisten belasteten Betriebsspannung + 5 V wurde der Spannungsreglerschaltkreis *MAA 723* eingesetzt. Die weiteren Betriebsspannungen erzeugen die integrierten Festspannungsregler *MA 7805* bzw. *MA 7812*. Auf Grund der verwendeten Spannungsregler (bei + 5 V in Verbindung mit geeigneter externer Beschaltung) sind alle Regelstrecken kurzschlußfest und temperaturkompensiert.

Ohne Schutzmaßnahmen können die Schaltkreise *MA 7805* bzw. *MA 7812* bei entsprechender kapazitiver Belastung zerstört werden, denn beim Abschalten entladen sich Lastkapazitäten über die genannten Regelschaltkreise. Zum Schutz der Schaltkreise wurden deshalb die Dioden *VD9*, *VD14*, *VD19* vorgesehen.

Der Einfluß netzbedingter Störungen läßt sich durch Netzfilter wirkungsvoll vermindern.

Im Monitorprogramm ist das softwaremäßige Abschalten des Computers über die Tastatur vorgesehen (Monitorbefehl: OFF). Wer von dieser Möglichkeit Gebrauch machen will, muß natürlich einen kleinen zusätzlichen Aufwand im Netzteil in Kauf nehmen. Bild 10.2 zeigt die für diesen Zweck benötigte Relaischaltung.

Im Zusammenhang mit der Stromversorgung ist es notwendig,

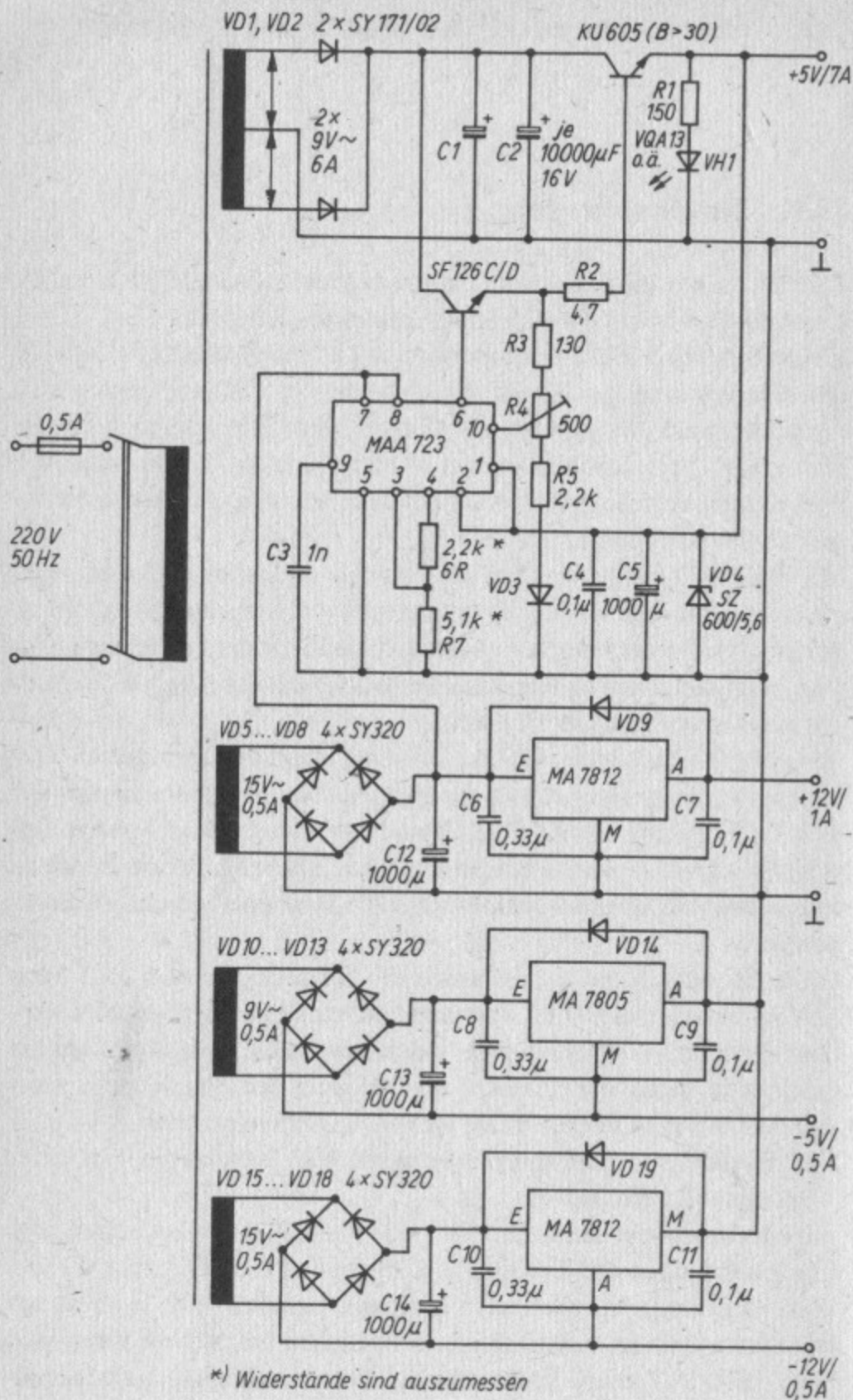


Bild 10.1 Netzteil für 5 V, - 5 V, 12 V, - 12 V

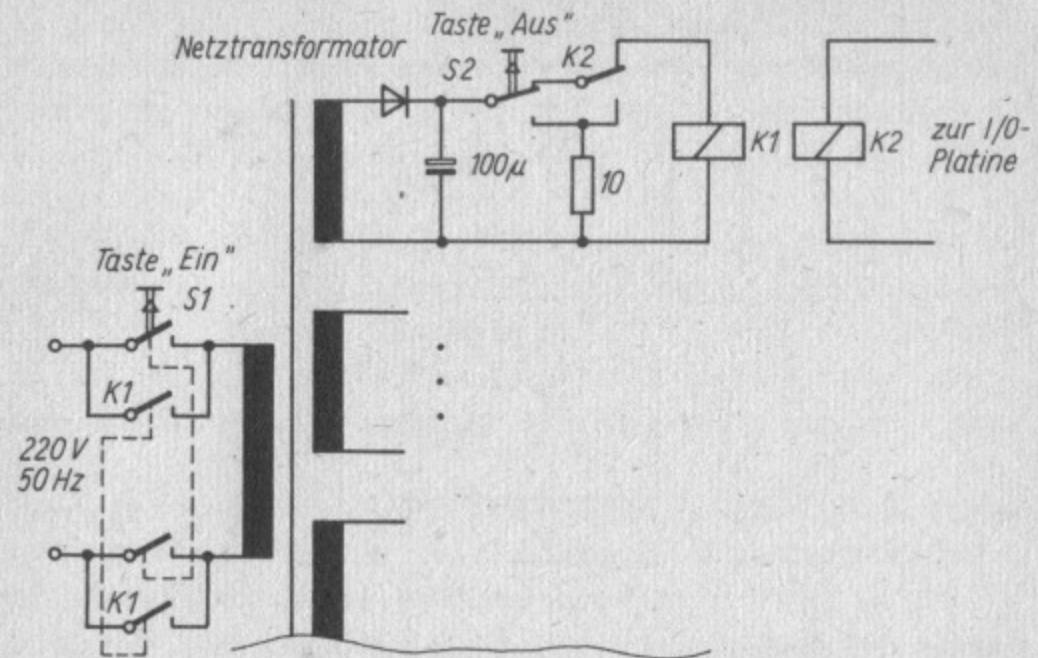


Bild 10.2 Relaischaltung für Power-off

noch auf die bereits im Abschnitt 4.3.5 erwähnte wichtige Randbedingung für den Einsatz der Speicherschaltkreise *U 555 (2708)* bzw. *U 256 (4116)* hinzuweisen:

Eine Vorrangschaltung muß absichern, daß die Betriebsspannung $U_{BB} = -5 \text{ V}$ (Substratvorspannung für oben genannte Speichertypen) beim Einschalten zuerst und beim Ausschalten zuletzt am Schaltkreis anliegt. Beim Ausfall von U_{BB} muß man die Betriebsspannungen $U_{CC} = 5 \text{ V}$ und $U_{DD} = 12 \text{ V}$ sofort abschalten. Beachtet man diese Vorschrift nicht, können die genannten Schaltkreise zerstört werden. Nach den Erfahrungen der Autoren ist die Beachtung der genannten Randbedingung dringend anzuraten.

Bild 10.3 zeigt eine für diesen Zweck bewährte Vorrangschaltung. Beim praktischen Aufbau dieser Schaltung ist darauf zu achten, daß die Spannung -5 V unmittelbar am Bus abgegriffen wird (d. h. in der Nähe der gefährdeten Speicherschaltkreise).

10.2. Systemaufbau

Sämtliche Daten-, Adreß- und Steuerleitungen einschließlich Taktsignal C müssen auf den Systembus geführt werden. Das gilt auch für die Betriebsspannungen (+ 5 V, - 5 V, + 12 V, - 12 V) und die Masseleitung.

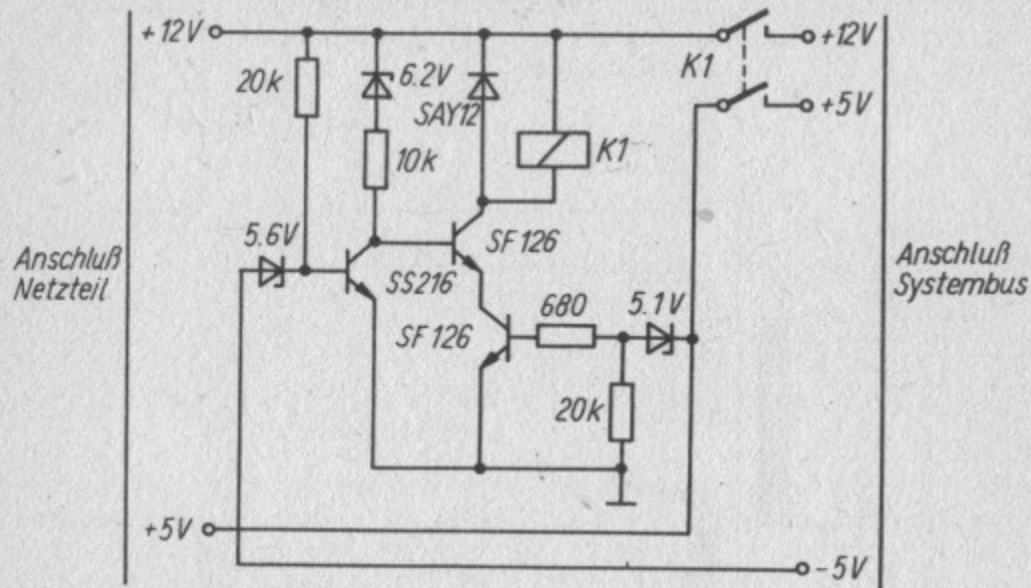


Bild 10.3 Vorrangschaltung für -5 V

Die konkrete Gestaltung des Systembusses ist abhängig von den Gegebenheiten, die dem einzelnen zur Verfügung stehen. Das bezieht sich besonders auf Steckverbinder, Einschub- und damit Leiterkartenabmessungen sowie Möglichkeiten zum Anfertigen der Leiterkarten. Unabhängig von diesen individuellen Voraussetzungen sind noch einige allgemeingültige Hinweise notwendig.

So muß vor allem auf Systematik und Übersichtlichkeit geachtet werden. Von der Funktion her zusammengehörige Signale sollten auch auf dem Bus benachbart und in geordneter Reihenfolge untergebracht werden, sofern übergeordnete Gesichtspunkte dem nicht entgegenstehen. Weiter achtet man möglichst darauf, daß Kurzschlüsse benachbarter Leitungen nicht zu Zerstörungen führen. Betriebsspannungen und Masse werden zweckmäßigerweise doppelt kontaktiert.

Besonders wichtig ist es, daß man die Störungsfreiheit auf dem Systembus absichert. Störungen sind sowohl unerwünschte Beeinflussungen von Bussignalen untereinander als auch von außen kommende Einwirkungen (z. B. vom Netz oder durch Störstrahlung verursacht). Unter dem zuletzt genannten Aspekt sind alle Busleitungen so kurz wie möglich zu halten. Netzspannungen führende Leitungen dürfen grundsätzlich nicht in Busnähe angeordnet werden.

Eine Orientierung für die Systembusgestaltung bieten industrielle Busausführungen, z. B. die des K 1520 [13]. Aus der Sicht des

Amateurs sind jedoch individuelle von Industriestandards abweichende Busgestaltungen zulässig. Wenn dabei die oben genannten Gesichtspunkte beachtet werden, ist bei der Systemtaktfrequenz von 2,5 MHz ein problemloser Betrieb im allgemeinen abgesichert.

Für die praktische Realisierung erweist es sich am günstigsten, den Bus in gedruckter Schaltungstechnik als Rückverdrahtungsplatine aufzubauen. Dabei werden die Steckverbinder direkt auf die Busplatine gelötet. Betriebsspannungs- und Masseleitungen versieht man entsprechend ihrer Strombelastung mit größerem Querschnitt.

Als Ergänzung zu dieser Problematik sei noch darauf hingewiesen, daß gemäß den Erfahrungen der Autoren ein konventionell verdrahteter Systembus in sauberer Ausführung, unter Beachtung oben genannter Gestaltungsgesichtspunkte, ebenfalls einwandfrei funktioniert.

Der vorgestellte Mikrocomputer wurde in Einschubbauweise realisiert. Im Computereinschub werden die CPU-Platine sowie Speicher- und Interfacebaugruppen untergebracht und über den Systembus miteinander verbunden. Eine Ausnahme macht das Tastaturinterface, das (abgesehen vom Tastatur-PROM) direkt in der Tastatur platziert wird. Das ist deshalb sinnvoll, weil andernfalls für das Verbindungskabel zwischen Tastatur und Computer wesentlich mehr Leitungen benötigt werden (etwa 32 an Stelle von 11).

Der Computereinschub enthält auch die Stromversorgung. Reservesteckplätze ermöglichen eine spätere individuelle Erweiterung. Abmessung und Gestaltung der Leiterkarten für die im Computereinschub befindlichen Baugruppen sind, wie bereits erwähnt, von den individuellen Möglichkeiten des einzelnen abhängig. Aus diesem Grund werden in dieser Broschüre auch keine Leiterplattenentwürfe veröffentlicht.

Als günstiges Leiterkartenformat haben sich die EGS-Abmessungen 170 mm × 210 mm erwiesen. Frei verdrahtbare Universalplatinen sollte man nicht verwenden. Der Aufwand, der mit dem Entwurf und der Herstellung sauberer, übersichtlicher Leiterkarten verbunden ist, zahlt sich spätestens bei Inbetriebnahme und Fehlersuche sowie durch hohe Betriebszuverlässigkeit wieder aus. Es sei davor gewarnt, die doch vergleichsweise komplizierte Hardware eines Mikrocomputers auf der Basis von „Drahtverhau“-

Schaltungen zu realisieren. Bei einem derartigen Schaltungsaufbau wird außer dem permanenten Mißerfolg am Computer nichts programmierbar sein.

Auf Grund der computerspezifischen Schaltungen stellt man die Platinen nur aus doppelt kaschiertem Leiterplattenmaterial her. Zusammen mit den Leiterkartenabmessungen ergibt sich damit die Möglichkeit, alle wichtigen Baugruppen jeweils auf einer Leiterkarte unterzubringen.

Als Verbindungselemente zwischen Bus und Leiterkarten werden geeignete Steckverbinder benötigt. Zu empfehlen sind 58polige Steckverbinder indirekter Ausführung. Bei Verwendung direkter Steckverbinder sollten die Leiterplattenkontakte möglichst oberflächenveredelt sein. Dies ist dem Amateur aber nicht immer möglich.

Zum Aufbau des Mikrocomputers werden relativ viele Schaltkreise benötigt. Der Einsatz von qualitätsgeminderten Bastlerschaltkreisen kann zu erheblichen Problemen führen und bedeutet letztlich keine Kostenersparnis. Eine CPU U 880 mit verminderter Taktfrequenz (z. B. $f = 1$ MHz) kann man einsetzen, wenn der Taktgenerator dann entsprechend dimensioniert wird.

Für das Abblocken der Betriebsspannungszuführungen mit Stützkondensatoren gelten, wenn keine speziellen Hinweise gegeben wurden (siehe Abschnitt 4.2.2), die für TTL-Schaltkreise gültigen Richtlinien [3].

Wichtiger Hinweis zum Bildschirm:

Grundsätzlich darf man nur Fernsehgeräte verwenden, die galvanisch vom Netz getrennt sind. Das trifft im allgemeinen für moderne voll halbleiterbestückte Geräte zu. Empfohlen seien vor allem Koffergeräte. Besonders geeignet ist z. B. der Kofferempfänger *Combivision* wegen seiner guten Wiedergabequalität.

Unter Berücksichtigung aller Hinweise sollte der Mikrocomputer nun schrittweise in folgender Reihenfolge aufgebaut und in Betrieb gesetzt werden:

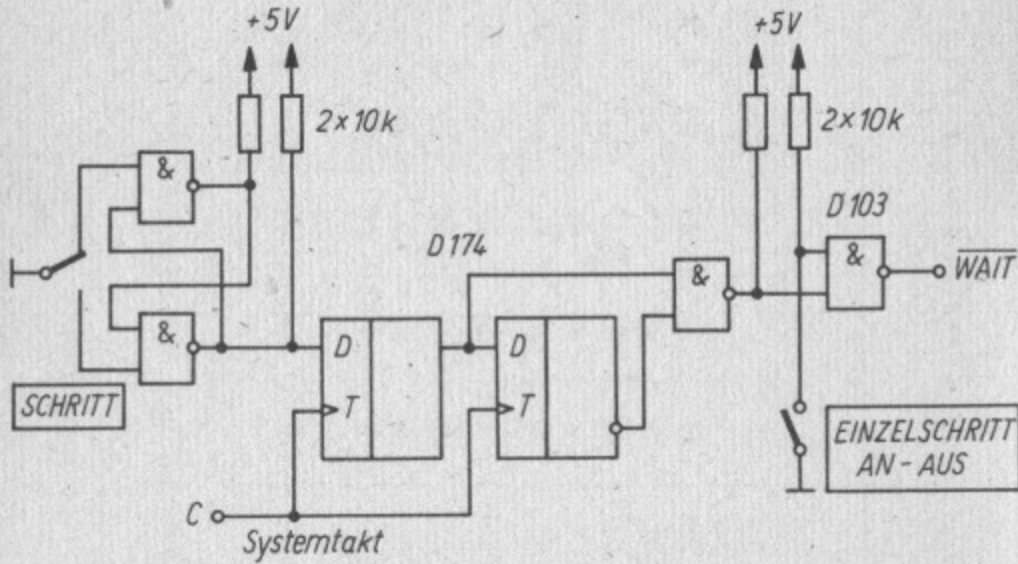
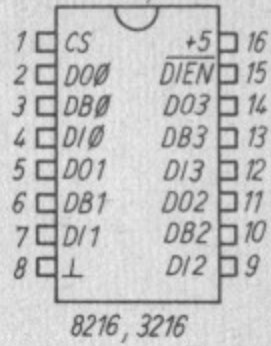
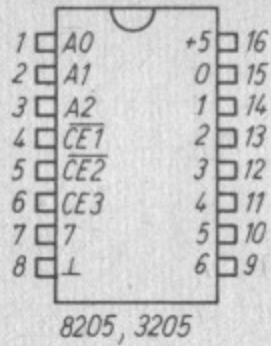
1. Einschub mit Systembus und zugehörigen Steckverbindern; für jede einzelne Leitung müssen der galvanische Durchgang über den gesamten Bus bis hin zum einzelnen Steckkontakt sowie die Kurzschlußfreiheit überprüft werden
2. Netzteil unter Beachtung der einschlägigen Schutzgütebestimmungen

3. CPU-Baugruppe
4. Fernsehinterface
5. Tastatur
6. Programmieren des Monitors auf PROM-Schaltkreise U 555, Inbetriebnahme des Monitorprogramms
7. Ein-/Ausgabeschchnittstellen
8. Kassetteninterface
9. Speichererweiterung nach individuellen Möglichkeiten und Bedarf (möglichst viel RAM)
10. Weitere Hardware nach Bedarf (z. B. PROM-Programmiergerät)

Die in 10. angeführte Hardware sowie darüber hinausgehende individuelle Erweiterungen sollten erst in Angriff genommen werden, wenn die Punkte 1. bis 9. des Systemaufbaus zufriedenstellend absolviert sind. Voraussetzung für die Verwirklichung eigener Ideen ist natürlich auch die vorherige Aneignung ausreichender Hardware- und Softwarekenntnisse.

Zum Schluß sei noch der Hinweis gegeben, daß zur erfolgreichen Arbeit mit dem Mikrocomputer eine übersichtliche und ständig aktualisierte Hard- und Softwaredokumentation gehört. Erfahrungsgemäß wird nämlich genau das mühsam erarbeitete und eingetippte Programm versehentlich gelöscht, das man vergessen hat, ordentlich zu dokumentieren.

Einige Pinbelegungen



Einfache Einzelschrittschaltung, mit dessen Hilfe Hardwarefehler aufgespürt werden können. Die Signale auf dem Datenbus und auf dem Adressbus können durch Leuchtdioden oder Hexadezimalanzeigen sichtbar gemacht werden.

HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII
00	NUL	10	DLE	20	SPACE	30	0	40	@	50	P	60	\	70	p	80	DEL
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q	81	
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r	82	
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s	83	
04	EOT	14	DC4	24	\$	34	4	44	D	54	T	64	d	74	t	84	
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u	85	
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v	86	
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w	87	
08	BS	18	CAN	28	(38	8	48	H	58	X	68	h	78	x	88	
09	HT	19	EM	29)	39	9	49	I	59	Y	69	i	79	y	89	
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z	8A	
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[6B	k	7B	{	8B	
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C	}	8C	
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D]	6D	m	7D	~	8D	
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	DEL	8E	
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F		8F	

Umwandlungstabelle Hexadezimal ↔ ASCII-Code