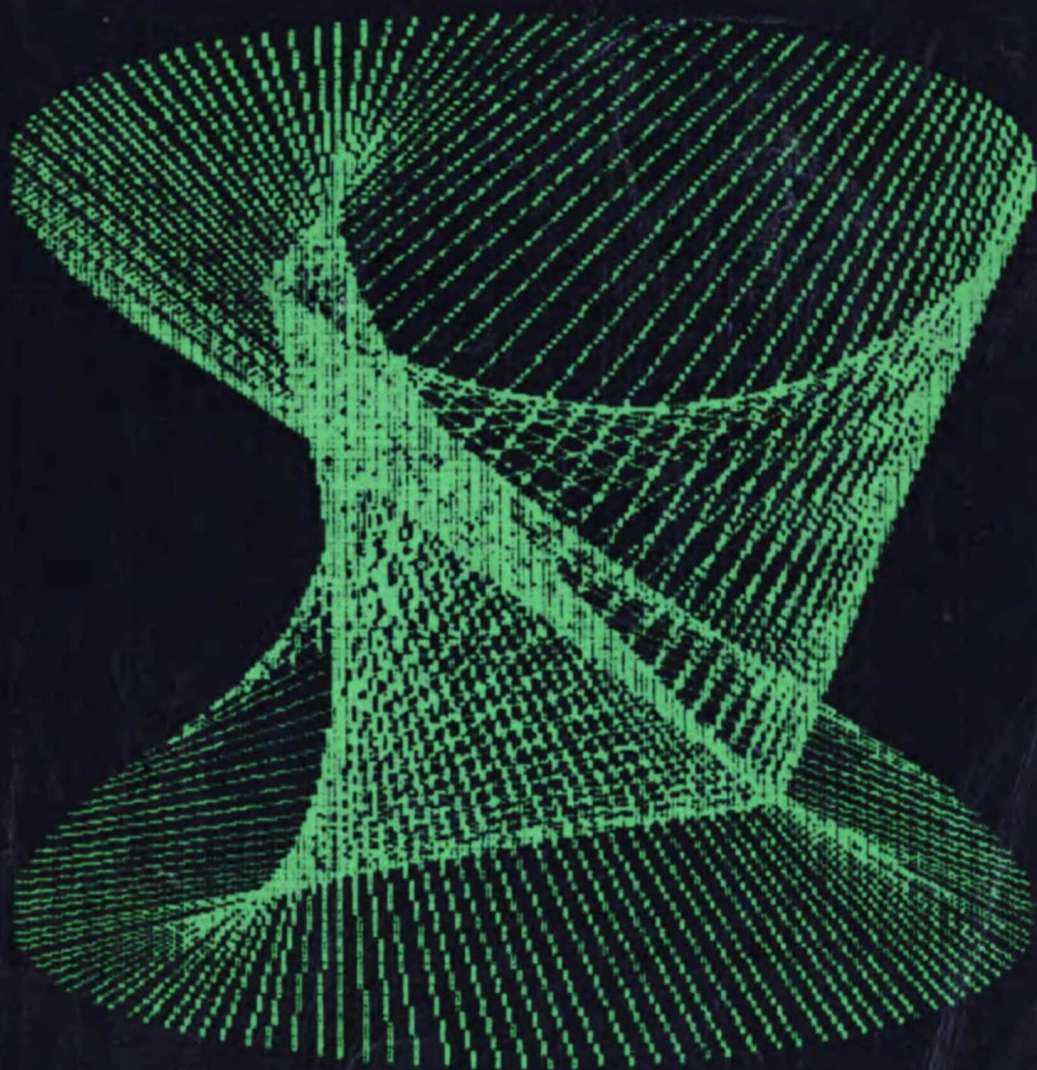


PRAKTISCHE MIKROCOMPUTER TECHNIK



Manfred Kramer



Manfred Kramer · Y2 3VO

Praktische Mikrocomputer- technik



**MILITÄRVERLAG
DER DEUTSCHEN DEMOKRATISCHEN
REPUBLIK**

Kramer, Manfred:
Praktische Mikrocomputertechnik. –
Berlin: Militärverlag der DDR (VEB), 1987. –
272 S. – 190 Bilder – (Amateurbibliothek)

ISBN 3-327-00361-0

1. Auflage
© Militärverlag der Deutschen Demokratischen Republik
(VEB) – Berlin, 1987
Lizenz-Nr. 5
Printed in the German Democratic Republic
Lichtsatz: INTERDRUCK
Graphischer Großbetrieb Leipzig – III/18/97
Druck und buchbinderische Verarbeitung:
Druckerei des Ministeriums für Nationale Verteidigung
(VEB) – Berlin – 32360-6
Lektor: Steffen Würtenberger
Umschlaggestaltung: Rosemarie Lebek
Zeichnungen: Birgit Schepuhl, Christel Meyer, Ingeborg Haendschke
Typografie: Martina Schwarz
Redaktionsschluß: 15.2.1986
LSV 3539
Bestellnummer: 7469367
01450

Inhaltsverzeichnis

Vorwort	
1. Schaltungstechnik	8
1.1. Einführung	8
1.1.1. Historische Entwicklung	8
1.1.2. Programmablaufplan	8
1.1.3. Kodierung von Signalen	9
1.1.4. Allgemeiner Aufbau eines Mikrocomputers	10
1.2. Grundgerät	11
1.2.1. Bauelemente	11
1.2.2. BUS-System	16
1.2.3. Anschluß von Tastatur, Kassettenmagnetbandgerät und Fernschreiber (Drucker)	20
1.3. Bildschirmansteuerung	27
1.3.1. Bildwiederholpeicher	28
1.3.2. Zeitbasis	30
1.3.3. Bildschirmgerät	30
1.4. Stromversorgung	35
1.5. Mechanischer Aufbau	35
1.6. Inbetriebnahme	38
2. Programmierung	41
2.1. Einführung in die Programmiersprache BASIC	41
2.1.1. BASIC-Sprachelemente	41
2.1.2. Bedienung des BASIC-Interpreters	48
2.1.3. Programmbeispiele in BASIC	51
2.2. Maschinenbefehle des Mikroprozessors <i>UB 880 D</i>	52
2.2.1. Lade- und Austauschbefehle	55
2.2.2. Speicherbereichsbefehle	56
2.2.3. Arithmetische und Logikbefehle	56
2.2.4. Verschiebepbefehle	61
2.2.5. Bitbefehle	61
2.2.6. Eingabe-/Ausgabebefehle	63
2.2.7. Befehle für Sprünge, Unterprogrammaufruf und Rücksprung	63
2.3. Monitorprogramm	64
2.3.1. Struktur	65
2.3.2. IN/OUT-Routinen	65
2.3.3. Einsprungtabelle	65
2.3.4. IN/OUT-Zuweisung	66
2.3.5. Debugger	66
2.3.6. RAM-Tabelle	67
2.4. Programmbeispiele in Maschinensprache	67
2.4.1. Bedienung des Monitors	67
2.4.2. Unterprogrammtechnik	70

2.4.3.	Unterprogramme aus dem Monitor	71
2.5.	Programmierung mit Assembler	72
2.5.1.	Editor	72
2.5.2.	Assemblerbeschreibung	74
2.5.3.	Bedienung des Assemblers	76
3.	Erweiterungen	77
3.1.	Speicher	77
3.1.1.	Schaltungen	77
3.1.2.	Speicherumschaltung	82
3.2.	EPROM-Programmierzusatz	85
3.3.	Zeitgeberanschluß	89
3.4.	Datenspeicherung auf Magnetband sowie Aus- und Eingabe serieller Daten	97
3.5.	Handler für Lochstreifen	104
3.6.	Anschluß des D/A-Wandlers <i>C 5658 D</i>	106
3.7.	Anschluß des A/D-Wandlers <i>C 520 D</i>	106
4.	Weitere Mikrocomputerbauelemente und Geräte	110
4.1.	Peripheralschaltkreise	110
4.2.	Speicher	113
4.3.	Interface-Schaltkreise	113
4.4.	Einchip-Mikrocomputer	114
4.5.	16-bit-Mikroprozessoren	116
4.6.	Folienspeicher (Floppy-Disk)	116
4.6.1.	Betriebssystem für Floppy-Disk	117
4.6.2.	Floppy-Disk-Schaltkreis <i>U 8272</i>	120
4.6.3.	Anschlußsteuerung für Floppy-Disk-Laufwerk und Inbetriebnahme	123
5.	Anhang	126
5.1.	Funktionserläuterung ausgewählter Maschinenbefehle des Mikroprozessors <i>UB 880 D</i>	126
5.2.	Programme	139
5.2.1.	Programm IN/OUT-Monitor	139
5.2.2.	Programm Debugger	150
5.2.3.	Programm Disassembler in Maschinencode	164
5.2.4.	Zeichengeneratorinhalt	165
5.2.5.	Programm BASIC-Interpreter in Maschinencode	166
5.2.6.	Programm Editor	172
5.2.7.	Programm Assembler	182
5.2.8.	Programm CBIOS	186
5.2.9.	Programm Monitor für Adresse 0F000H	199
5.2.10.	Programm zum Formatieren von Disketten	200
5.3.	Abkürzungen	204
5.4.	Schaltkreisdaten, Anschlüsse, Vergleichsliste	205
5.5.	Leiterplattenlayouts	249
6.	Literaturverzeichnis	265
7.	Sachwortverzeichnis	266

Vorwort

In den letzten Jahren ist bei der Entwicklung von Bauelementen der Mikroelektronik durch deren Programmierbarkeit ein revolutionierender Fortschritt eingetreten. Mit wenigen hochintegrierten Schaltkreisen kann eine kaum übersehbare Vielfalt von Problemen in der Volkswirtschaft, bei der Landesverteidigung, im Bildungswesen, aber auch im Bereich des Amateurs gelöst werden.

Höhere Programmiersprachen, z. B. BASIC, ermöglichen auch dem Laien die Anwendung dieser faszinierenden Technik und erleichtern dem Elektroniker den Start ganz wesentlich.

Bei einer so komplexen Technik, deren Grundlagen heute durch Teams von Ingenieuren gelegt werden, findet der einzelne den Einstieg am effektivsten durch ein konkretes Beispiel bei Selbstbau und Programmierung eines eigenen Computers. Möglichkeiten für kreative Tätigkeit sind durch Programmierbarkeit, Erweiterungen oder Anpassung an eigene Bedingungen, z. B. bei den eingesetzten Bauelementen, ausreichend gegeben.

Das vorliegende Buch führt einerseits in die Schaltungstechnik der Mikrocomputer ein. Konkrete Schaltungen zeigen eine amateurgerechte Lösung für alle wichtigen Bauteile eines Mikrocomputers. Neben den kompletten Baugruppen, wie CPU/PIO-Karte, Bildschirmsteuerung sowie verschiedensten Speicherkarten, geben viele Teilschaltungen Hinweise auf die vielfältigen Möglichkeiten zur Lösung schaltungs-technischer Probleme.

Da die Leistungsfähigkeit eines Computers jedoch andererseits vor allem von den verfügbaren Programmen abhängt, nimmt

die Darstellung von Programmierbeispielen in Maschinen- und Assemblersprache sowie in BASIC breiten Raum ein. Das Monitorprogramm bildet als Assembler-Quell-Listing eine hochwertige, vollständige Dokumentation. Der Amateur wird damit in die Lage versetzt, alle vorhandenen Möglichkeiten dieses Programms zu nutzen. Der vorgestellte BASIC-Interpreter ist zwar mit etwa 12 kByte recht umfangreich, seine Leistungsfähigkeit befriedigt jedoch auch hohe Ansprüche. Er dient vor allem der Einführung in die Programmierung. Mit den Programmen «Editor» und «Assembler» erhält der Amateur professionelle Werkzeuge für die eigene Programmentwicklung. Eine kurze Sammlung von Daten der wichtigsten verwendeten integrierten Schaltkreise bildet den Abschluß des Buches.

Trotz großer Sorgfalt bei der Bearbeitung durch alle Beteiligten sind bei der Fülle des Stoffes Fehler nicht auszuschließen. Mitteilungen dazu sowie sonstige Hinweise können an folgende Anschrift geschickt werden:

Manfred Kramer, 1136 Berlin, Postschließfach 156.

Abschließend dankt der Autor allen Freunden und Bekannten, die mit Rat und Tat bei der Arbeit halfen, vor allem OM Günter, Y2 5WO, der kurzfristig die Leiterplatten fertigte und bei den Fotoarbeiten wertvolle Hilfestellung leistete.

Besonderer Dank gilt Herrn Dipl.-Ing. S. Würtenberger vom Militärverlag der DDR für die ausgezeichnete Zusammenarbeit.

Berlin, im November 1984 *Manfred Kramer*

1. Schaltungstechnik

1.1. Einführung

1.1.1. Historische Entwicklung

Zu Anfang der siebziger Jahre war die Halbleiterindustrie in einer schwierigen Lage. Einerseits war es möglich, monolithische, digitale Halbleiterschaltkreise mit relativ hohem Integrationsgrad herzustellen, die eine große Anzahl einzelner Transistoren, Dioden und Widerstände ersetzen konnten. Auf der anderen Seite erfordert die Technologie zur Entwicklung eines bestimmten Schaltkreises so hohen Aufwand, daß für eine wirtschaftlich vertretbare Fertigung größere Stückzahlen erforderlich sind. Der Ausweg aus dieser Situation war 1971 die Erfindung des Mikroprozessors. Dadurch entstand ein universelles Bauelement, das in Verbindung mit weiteren Schaltkreisen für die unterschiedlichsten Aufgaben anwendbar ist. Die Anpassung an die jeweiligen Anwendungsfälle erfolgt durch wenige Bauelemente geringeren Integrationsgrades sowie vor allem durch programmierbare Speicherschaltkreise. Ständig werden neue, vielfältige Einsatzmöglichkeiten erschlossen. Deshalb bildet die Stückzahl bei der Fertigung von Mikroprozessoren und Speicherschaltkreisen kein Hindernis mehr für die Entwicklung immer leistungsfähigerer und höher integrierter Schaltkreise. Auch hier ist eine Grenze z. Z. schwer abschätzbar. Eine relativ schnelle, jedoch nicht ganz so spektakuläre Entwicklung vollzog sich auch auf dem Gebiet der Rechnerperipherie (Bedien- und Anzeigebaugruppen). Die ersten Schritte der Mikrocomputerentwicklung begannen mit Logikpegelanzeigen (Lampen, LED) und Zifferntastaturen. Die Programme wurden meist auf vorhandenen Groß- oder Minicomputern bearbeitet. Dann ging man zu Fernschreibmaschinen über, die nicht nur das Ausdrucken der Pro-

gramme und die Eingabe in höheren Programmiersprachen direkt mit dem verwendeten Mikrocomputer gestatteten, sondern auch über angebaute Lochbandgeräte auf einfache Weise eine zuverlässige und übersichtliche externe Speicherung von Programmen erlaubten. Ein weiterer wichtiger Vorteil der Fernschreibmaschinen besteht in der seriellen Informationsübertragung über nur 4 Leitungen; Lampen und Tasten erfordern vielpolige Kabel. Nachteilig beim Einsatz von Fernschreibern als Datenperipherie sind die geringe Arbeitsgeschwindigkeit, die Geräuschentwicklung und der Papierverbrauch. Nicht immer wird ein Ausdruck des Programms benötigt, häufiger müssen nur Daten angezeigt werden. Deshalb setzten sich zunächst bei den Mikrocomputern die bereits bei den Groß- und Prozeßrechnern üblichen Bildschirmterminals durch, die außer dem Bildschirm noch eine alphanumerische Tastatur enthalten und wie ein Fernschreiber über 2 bis 4 Leitungen mit dem Computer verbunden sind. In letzter Zeit bilden Tastaturen und Bildschirmgeräte meist elektronisch und mechanisch mit dem Mikrocomputer eine Einheit.

Trotzdem sind auch die zuvor genannten Möglichkeiten weiterhin im Gebrauch, und viele übliche Begriffe und Bezeichnungen entstammen dieser Entwicklung.

1.1.2. Programmablaufplan

Wie bereits erwähnt, wird die Mikrocomputerschaltung an eine Aufgabenstellung vor allem durch das Programm angepaßt. Ein wichtiges Hilfsmittel bei der Erarbeitung von Programmen ist die grafische Darstellung im Programmablaufplan. Sowohl für die grundsätzliche Lösung einer Aufgabenstellung als auch für die Festlegung von Einzelheiten

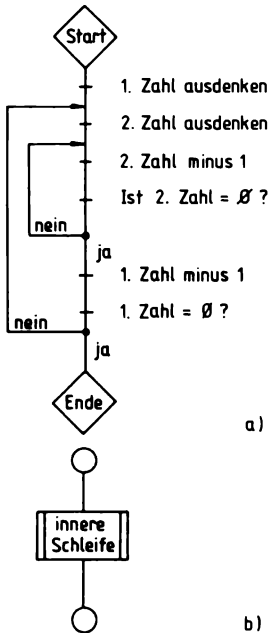


Bild 1.1 Programmablaufplan Zeitschleife

kann man damit die Arbeitsweise eines Computers übersichtlich und anschaulich darstellen. Ein Beispiel soll dies näher erläutern:

Recht häufig sind Verzögerungszeiten erforderlich, wie sie in der Digitaltechnik von monostabilen Multivibratoren erzeugt werden. Bekanntlich prellen bei der Betätigung eines Schalters die Kontakte noch mehrmals auseinander, bevor sie sich endgültig schließen. Bei der Abfrage eines Tastenkontaktes durch einen Computer muß deshalb die ansonsten recht schnelle Abfrage nach einigen Millisekunden wiederholt werden, um eine sichere Information zu gewährleisten. Sonst wäre zusätzlicher materieller Aufwand für eine Entprellschaltung erforderlich. Ein paar Speicherplätze für ein kurzes Programm sind in diesem Fall jedoch billiger.

Die Verzögerungszeit kann man einfach durch eine Zählschleife erzeugen. Man denkt sich eine Zahl und zieht von dieser so lange 1 ab, bis sie den Wert 0 hat. Dabei ver-

geht Zeit. Einen Programmablaufplan für diese Aufgabe zeigt Bild 1.1a, wobei wegen der geforderten Länge der Zeit zwei Zahlen benutzt werden. Man erkennt schnell noch einige andere Lösungen für dieses Problem. Kommen Aufgaben mehrmals vor, so kann man die Lösung mehrfach verwenden und im Programmablaufplan als Teilprogramm vereinfacht darstellen (Bild 1.1b).

Die Aufstellung eines Programmablaufplans kann die Lösung einer Aufgabe sehr erleichtern.

1.1.3. Kodierung von Signalen

Bekanntlich nehmen die Signalpegel in digitalen Schaltungen nur zwei Wertebereiche ein, die man meist mit H (high) und L (low) bezeichnet.

Bei der Kombination mehrerer Leitungen gibt es dann $m = 2^n$ Möglichkeiten, wobei n die Anzahl der Leitungen ist. Um diese Signale mit geringem Aufwand eindeutig zu beschreiben, verwendet man in der Mikrocomputertechnik die sogenannte hexadezimale Schreibweise. Dabei beschreibt die Darstellung zwar jeden einzelnen Pegel, je-

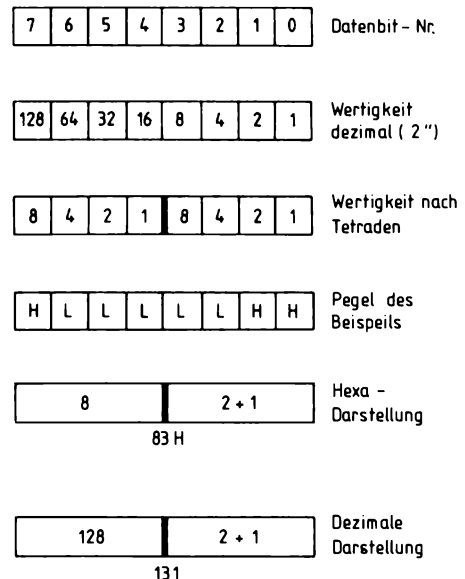


Bild 1.2 Kodierung von Signalen

doch in Gruppen von je 4 Leitungen (Tetraden), denen entsprechend ihrer Anordnung innerhalb dieser Gruppen eine Wertigkeit zugeordnet ist. Bild 1.2 zeigt als Beispiel eine 8-bit-Anordnung, auf der als Inhalt eines Speicherplatzes die Zahl 131 nach dezimaler, die Zahl 83 nach hexadezimaler Schreibweise anliegt. Der Umgang mit hexadezimalen Zahlen erfordert etwas Übung. Diese Form der Darstellung bietet jedoch in der Mikrocomputertechnik viele Vorteile.

1.1.4. Allgemeiner Aufbau eines Mikrocomputers

Bild 1.3 zeigt den Aufbau eines Computers. Die Funktionsweise der Schaltung beruht im wesentlichen auf dem Zugriff zum Speicher. Jeder Speicherplatz hat eine Nummer, die als Adresse bezeichnet wird und auf dem ein Dateninhalt steht. Adressen und Daten werden über Gruppen von Leitungen übertragen, die an alle Bauelemente parallel führen. Sollen z. B. die Daten eines bestimmten Speicherplatzes gelesen werden, so ist dazu das Steuersignal «Speicher lesen» zusätzlich zur Adresse erforderlich. Die Leitungen für Adressen, Daten und Steuerung in ihrer Gesamtheit bezeichnet man als BUS-System. Die Anzahl der Leitungen sowie die Arbeitsgeschwindigkeit auf dem BUS bestimmen entscheidend die Leistungsfähigkeit des Computers. Am weitesten verbreitet sind z. Z. Mikrocomputer mit 4, 8 und 16 bit breitem Datenbus. Abschnitt 1.2.2. erläutert einen 8-bit-BUS am konkreten Beispiel.

Mikroprozessor und Speicher bilden zusammen den eigentlichen Computer. Die Bauelemente zur Kopplung mit der «Um-

welt» (Anschluß von Tasten, Lampen, Bildschirm usw.) bezeichnet man als «Peripherie». Für viele dieser Kopplungsaufgaben gibt es Schaltkreise mit höherem Integrationsgrad, die dann meist auf einen bestimmten Prozessortyp und dessen BUS-System zugeschnitten sind. Im Mikroprozessor erfolgt die Verknüpfung der vom Speicher oder von der Peripherie gelesenen Daten entsprechend dem Programm. Ebenso wie im üblichen Sprachgebrauch versteht man darunter eine Handlung, die in Einzelschritten nacheinander abläuft und eine festgelegte Zielstellung verfolgt. Die Möglichkeiten der Signalverknüpfung im Prozessor sind durch dessen internen schaltungstechnischen Entwurf festgelegt und stehen im sogenannten Befehlssatz dem Anwender zur Verfügung.

Die grundsätzliche Arbeitsweise eines Computers beruht auf der zeitlich aufeinanderfolgenden Verknüpfung von Signalen: Nach dem Einschalten gibt der Prozessor die Speicherplatzadresse 0 aus und «liest» den als Antwort vom Speicher kommenden «Befehl». Nach Ausführung des Befehls durch den Prozessor erhöht dieser seinen internen Befehlszähler und gibt die nächste Speicheradresse aus ...

Auch der Umfang des Befehlssatzes eines Mikroprozessors bestimmt in hohem Maße das Leistungsvermögen einer Anwenderschaltung. Die Computerbefehle werden im Abschnitt 2. erläutert.

Die Mikrocomputertechnik ist durch den Einsatz besonders hoch integrierter Schaltkreise gekennzeichnet.

Dadurch ergeben sich trotz vieler Verbindungsleitungen übersichtliche, fast an »Blockschaltpläne« erinnernde Darstellungen. Eine zentrale Rolle spielt dabei die bereits erwähnte, zusammengefaßte Darstel-

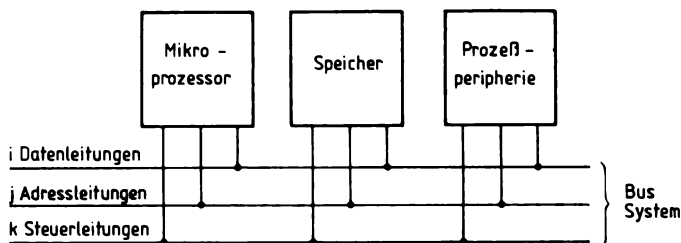


Bild 1.3
Struktur eines Computers

lung der Verbindungsleitungen, das BUS-System. Die Anzahl der Leitungen jeder Gruppe, die Logikpegel und die zeitliche Folge von Signalen sind ausschließlich vom eingesetzten Mikroprozessor abhängig. So hat der *UB 880 D* z. B. einen Adreßbus mit 16 Leitungen, der inzwischen veraltete *U 808* kam mit maximal 14 Adreßleitungen aus. Dementsprechend kann der *UB 880 D* maximal $2^{16} = 65\,536 = 64\text{ kByte}$ Speicher direkt adressieren, der *U 808* nur 16 kByte.

Durch Zwischenspeicherung und Umschaltung der Adressen kann der Speicher aber auch über diese Grenze hinaus anwachsen. Für eine problemlose Kopplung unterschiedlicher Baugruppen mit Speicher- oder Peripherieschaltkreisen ist auch die einheitliche Konstruktion der Leiterplatten sinnvoll, d. h., die Art der eingesetzten Steckverbinder sowie deren Belegung sind in der industriellen Praxis für einen bestimmten Mikrocomputer verbindlich festgelegt.

Es ist zweckmäßig, wenn sich der Amateur beim Aufbau eines seinen Möglichkeiten und Vorstellungen entsprechenden Mikrocomputers an die Industriestandards weitestgehend anlehnt. Man hat dann die Möglichkeit, verfügbare industrielle Baugruppen ebenfalls einzusetzen, was vor allem bei der Inbetriebnahme neuer Schaltungen und bei der Fehlersuche Vorteile bringen kann.

1.2. Grundgerät

Beim Aufbau eines Computers kann man ein Minimum an Funktionen definieren, die schon einen Betrieb des Gerätes erlauben. Wichtig für den Amateur ist hierbei die Möglichkeit, seine Technik schrittweise zu erweitern, ohne daß teure Bauelemente später überflüssig werden.

Bild 1.4 zeigt die Schaltung eines erweiterungsfähigen Mikrocomputers mit Prozessor, Speicher und Peripherieschaltkreis. Sie ist auf einer Leiterplatte mit den Maßen $95\text{ mm} \times 170\text{ mm}$ untergebracht und bildet den Grundstein für einen komfortablen Amateurcomputer. Obwohl die in Abschnitt 1.5. dargestellte Leiterplatte auch die

Bestückung mit anderen Schaltungen gestattet, soll dieses einfache Beispiel die Grundlagen der Arbeitsweise eines Mikrocomputers erläutern helfen.

1.2.1. Bauelemente

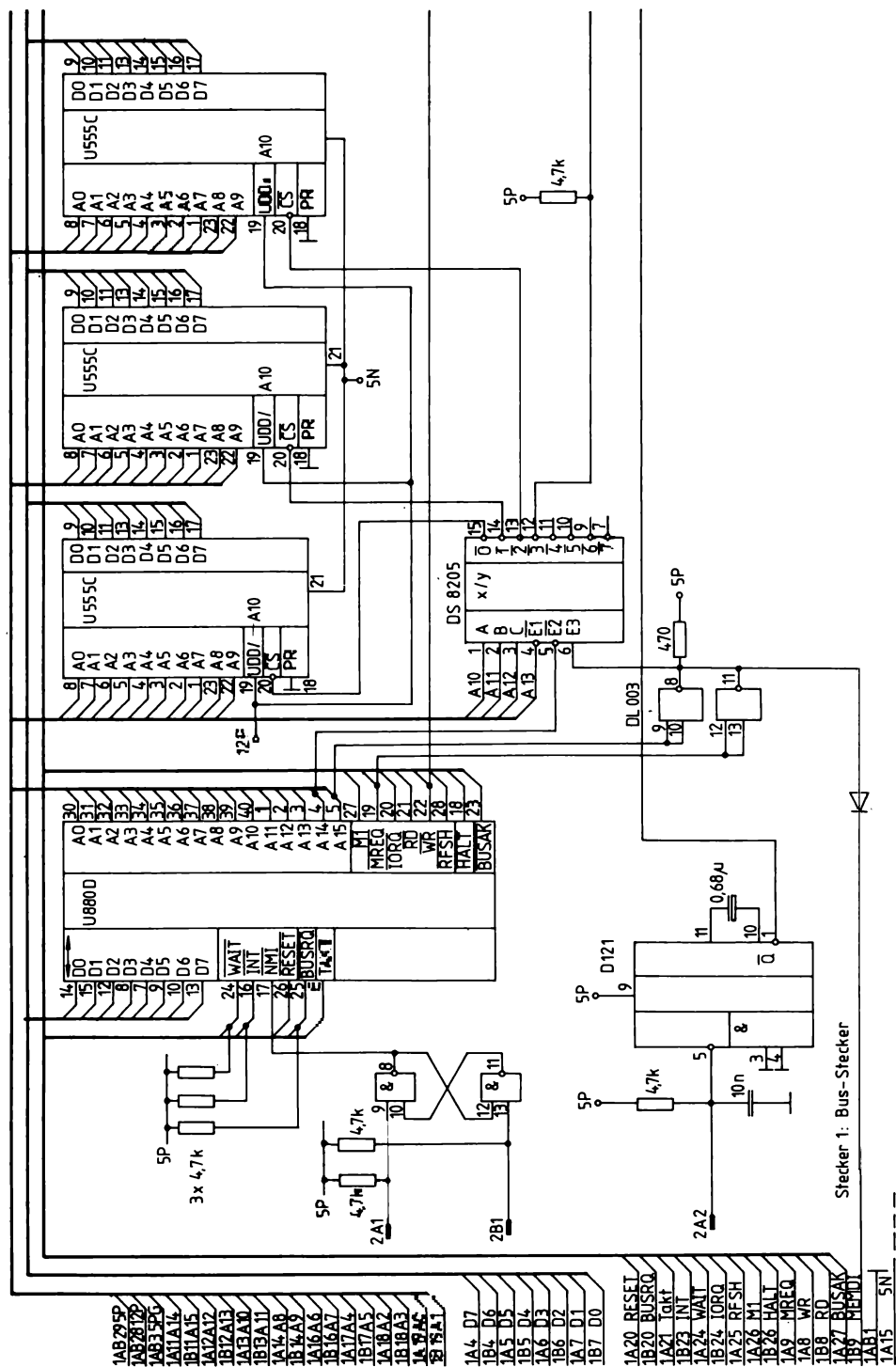
Die Funktion von Schaltungen ist nur verständlich, wenn die der eingesetzten Bauelemente bekannt ist. Leider sind die Funktionsbeschreibungen der hier verwendeten Bauelemente zusammen umfangreicher als dieses Buch, deshalb können sie nur kurz vorgestellt werden. Sofern die Erläuterungen und die Technischen Daten im Anhang nicht ausreichen, findet der Leser im Literaturverzeichnis Hinweise auf Ausführlicheres.

Falls die Bezeichnung der Bauelemente in der DDR von der international üblichen stark abweicht, wird auch die in den Katalogen der Hersteller (z. B. [26]) genannte angegeben.

Der Speicher besteht aus maximal 3 Schaltkreisen *U 555 C* (2708) für Festwerte sowie aus 2 Schaltkreisen *U 214 D* (2114) als Schreib-/Lesespeicher.

Die Arbeitsweise dieser Speicher erlaubt es besonders gut, das scheinbare Wirrwarr von Daten, Adressen und Steuersignalen übersichtlich darzustellen. Der Schaltkreis *U 555 C* gehört zu den löschbaren, elektrisch programmierbaren Festwertspeichern; in der Fachsprache hat sich für diese Art Speicher die Bezeichnung EPROM eingebürgert. Die Kapazität beträgt 8 192 Speicherplätze. Der Übersichtsschaltplan (nach [5]) zeigt die Struktur des internen Aufbaus (Bild 1.5). Wie bereits erwähnt, liegt die aktuelle Speicherplatznummer binär codiert auf den Adreßleitungen A0 bis A9. Die Speicherplätze sind intern in einer Matrix mit 128 Spalten und 64 Zeilen angeordnet. Jedoch merkt der Anwender davon nichts. Mit einer Adresse erscheint der Inhalt von je 8 Speicherplätzen an den Ausgängen. Man sagt, der Speicher ist mit $1024 \times 8\text{ bit}$ ($= 8\,192\text{ bit}$) organisiert. Die Adresse durchläuft die Dekoder für Zeilen und Spalten sowie den Spaltenschalter, der als Demultiple-

1. Schaltungstechnik



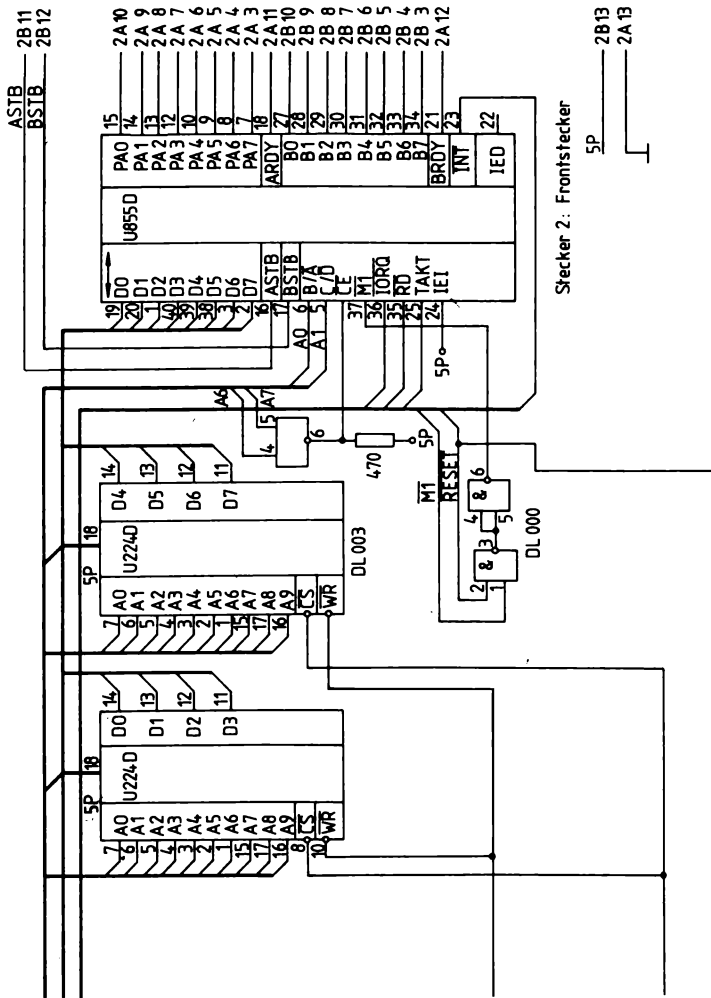


Bild 1.4 Stromlaufplan der Prozessorbaugruppe

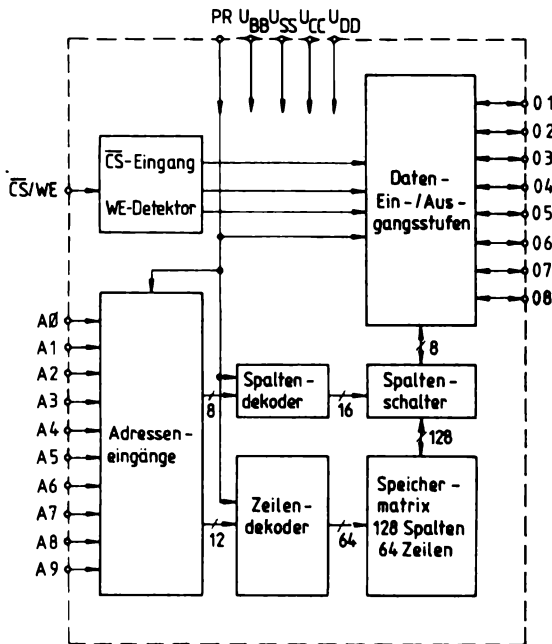


Bild 1.5
Struktur des EPROM U 555 C

xer die zugeordneten 8 bit auswählt. An die Ausgänge O1 bis O8 gelangt der Dateninhalt nur, wenn der Steuersignaleingang «Chip select/Write enable» ($\overline{CS}/WR$) auf Low-Pegel liegt. Anderenfalls nehmen die Datenausgangsstufen einen hochohmigen Zustand ein, so als ob sie am Datenbus gar nicht vorhanden wären. Diese Schaltungsart ist bei Parallelschaltung mehrerer Ausgänge, wie sie beim Datenbus vorliegt, unbedingt erforderlich und für die Mikrocomputertechnik charakteristisch. Bei einem herkömmlichen TTL-Schaltkreis liegt dagegen an den Ggengtaktausgängen entweder hoher oder niedriger Pegel, d. h., es gibt nur diese zwei Zustände der Ausgangssignale.

Beim Anschluß an das BUS-System mit seinen parallelen Leitungen an allen Schaltkreisen ist noch der genannte hochohmige Zustand erforderlich, damit nicht ein Ausgang mit niedrigem Pegel einen mit hohem beeinflußt. Man sagt, die Bauelemente haben Tristate-Ausgänge (3-Zustands-Ausgang).

Wie bereits erwähnt, führen an jedes EPROM die Adreßleitungen A0 bis A9. Damit nur eines ein $\overline{CS}$ -Signal erhält, muß ein

Dekodiert die verbleibenden Leitungen A10 bis A15 entsprechend auswertet. Die vollständige Dekodierung aller 6 Leitungen erscheint sinnvoll, weil anderenfalls der verfügbare direkt adressierbare Speicherbereich des Mikroprozessors beim späteren Ausbau des Computers nicht nutzbar ist. Es müssen jedoch nicht alle 64 möglichen Ausgangszustände auf der Leiterplatte verfügbar sein, weil ja maximal nur 4 kByte an Speicher körperlich vorhanden sind. Der Dekoder muß zwar über 6 Adreßeingänge, jedoch nur über 4 Ausgänge verfügen.

Am Schaltkreis **DS 8205** liefern nur die Signale der Eingänge 1-3 dekodierte Ausgangssignale, die anderen dienen lediglich der Freigabe des Dekoders. Auch die Peripherieschaltkreise werden mittels Adressen ausgewählt. Deshalb erzeugt der Prozessor zusätzlich die Steuersignale **MREQ** (Memory request = Speichieranforderung) und **IORQ** (In/Out request). **MREQ** ist dabei für die eindeutige Dekodierung von Speicheradressen erforderlich.

Da die Zahl der Eingänge des 8205 zu klein ist, erweitert sie die ODER-Schaltung mit den Gattern am Stift 6. Hier kann auch

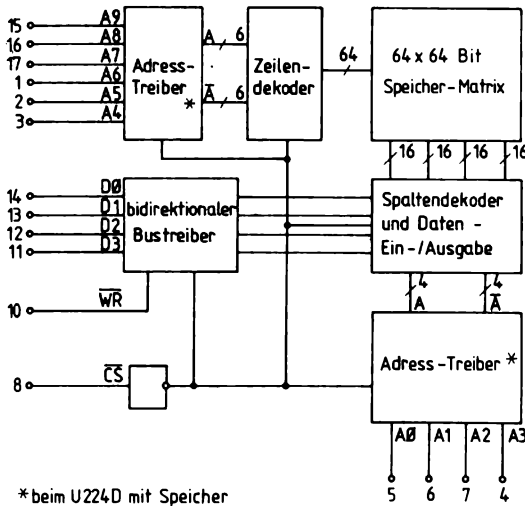


Bild 1.6
Struktur des RAM-Schaltkreises
U 214 D/U 224 D

eine äußere Schaltung die Arbeit des Dekoders sperren.

Die Schreib-/Lesespeicher U 214 D sind mit 1024 Plätzen zu je 4 bit organisiert, für den 8-bit-Datenbus sind deshalb 2 Schaltkreise notwendig.

Schreib-/Lesespeicher werden mit der Abkürzung RAM (Random Access Memory) bezeichnet. Bild 1.6 zeigt die interne Struktur des U 214 in der Übersicht.

Außer den Adreßleitungen und dem Chip-Select-Signal, welches parallel an beide Schaltkreise führt, benötigen diese Schaltkreise noch eine Information darüber, ob Daten aus dem Speicher gelesen oder in ihn geschrieben werden sollen (Stift 10). Auch das Signal WR ist mit L-Pegel aktiv. In Abhängigkeit von diesen beiden Signalen arbeiten die Datenleitungen des Schaltkreises entweder als Eingänge, Ausgänge oder sind hochohmig (Tristate).

- $\overline{CS}$ und $\overline{WR}$ aktiv (low): Daten schreiben
- $\overline{CS}$ low, $\overline{WR}$ high: Daten lesen
- $\overline{CS}$ high, $\overline{WR}$ beliebig: Tristate

Ebenso wie die Betriebsart «Tristate» ist auch der Betrieb der Datenleitungen in beide Richtungen charakteristisch für die meisten am BUS eines Mikrocomputers arbeitenden Peripherieschaltkreise sowie für den Prozessor selbst.

Der Schaltkreis U 855-PIO ist ein typischer Peripherieschaltkreis. Er verfügt über einen hohen Integrationsgrad und zeichnet sich durch universelle Anwendbarkeit aus. Nach den Erläuterungen zu den Datenleitungen der RAM-Schaltkreise überrascht es den Leser sicher nicht mehr, daß man bei diesem Schaltkreis mit Hilfe interner Speicherplätze die Anschlüsse als Eingänge oder Ausgänge programmieren kann.

Auch einige weitere Eigenschaften sind wählbar und können sich damit sogar im laufenden Programm verändern. Da der Betrieb des Schaltkreises im nächsten Abschnitt am konkreten Beispiel noch näher betrachtet wird, folgen hier nur kurz einige Erläuterungen zur Schaltung am BUS.

Wie bereits erwähnt, erfolgt die Auswahl der Peripherieschaltkreise ebenfalls über Adreßleitungen. Die Anzahl der Schaltkreise ist jedoch im allgemeinen wesentlich geringer als die der Speicher, so daß nur die unteren 8 bit des Adreßbusses dafür nötig sind. Damit sind je $2^8 = 256$ IN/OUT-Anschlüsse möglich.

Während der direkt adressierbare Speicherbereich des Prozessors von 64 kByte durchaus auch für den Amateur von Interesse ist, kann man bei den IN/OUT-Adressen auf einen großen Teil der Möglichkeiten ohne Bedenken verzichten. Die Dekodier-

rung des $\overline{CS}$ -Signals erfolgt deshalb hier nicht vollständig, sondern stark vereinfacht. Von den 8 Adreßleitungen liegen 2 direkt am Schaltkreis, der dafür einen internen Dekoder enthält. Von den restlichen 6 Leitungen wertet das Gatter am $\overline{CS}$ -Eingang nur die oberen 2 aus, so daß der Schaltkreis auf alle 64 Adressen oberhalb 0C0H reagiert, obwohl er nur 4 Adressen unterscheiden kann. Davon dienen 2 der Steuerung, die anderen 2 als IN/OUT-Kanäle. Mit Adreßbit 0 erfolgt die Auswahl der Kanäle A und B, mit bit 1 die Auswahl Steuerung oder Datenverkehr. Die Datenleitungen können selbstverständlich auch bidirektional arbeiten, der Pegel des Steuersignals $\overline{RD}$ (READ) kennzeichnet ihre Betriebsweise.

Den Unterschied zu den Speicheradressen erfährt der Peripherieschaltkreis aus dem Signalpegel des Steuersignals $\overline{IORQ}$ (aktiv Low). Bei der Steuerung industrieller Prozesse, wo Mikrocomputer in letzter Zeit zunehmend eingesetzt werden, besteht eine wichtige Forderung darin, auf die Signale des Prozesses mit festgelegter Rangordnung zu reagieren. Ein Gefahrensignal muß das gerade laufende Programm unterbrechen und sofort die nötige Reaktion der Steuerung auslösen. Dazu kann der Peripherieschaltkreis ein Signal $\overline{INT}$ (Interrupt) an den Prozessor geben. Da auch mehrere Schaltkreise interruptfähig sein können, der Prozessor jedoch die Programme nur nacheinander abarbeitet, muß die Rangordnung der Unterbrechungen durch Reihenschaltung der Steuerleitungen IEI (Interrupt enable input) und IEO (Interrupt enable output) festgelegt werden. Eine nähere Erläuterung zum Interruptverhalten erfolgt am konkreten Beispiel in Abschnitt 3.3. Der Eingang $\overline{MI}$ des PIO dient dem Rücksetzen interner Speicher. Das vollständige Rücksetzen mit dem Signal «Reset» erfolgt bei diesem Schaltkreis nicht über eine eigene Leitung wie sonst üblich. Wenn der Takt anliegt, $\overline{MI}$ über mindestens 2 Perioden aktiv (low) bleibt sowie die Signale $\overline{RD}$ und $\overline{IORQ}$ nicht auf aktiven Pegel schalten, bewertet der Schaltkreis PIO diesen Zustand als Rücksetz-Signal. «Reset» ist deshalb mit $\overline{MI}$ über zwei Gatter UND-verknüpft. Sind im Computer besonders

viele PIO-Schaltkreise vorhanden, empfiehlt es sich, die Schaltung nur einmal zentral aufzubauen.

Der Takt wird aus der quarzstabilisierten Frequenz der Bildschirmansteuerung abgeleitet und über den Steckverbinder 1 an alle Baugruppen geführt.

1.2.2. BUS-System

Die Schaltung nach Bild 1.4 zeigt auch die Belegung des Steckers mit dem BUS-System für die Verbindung mit weiteren Baugruppen. Alle Erweiterungen erfolgen über diesen Stecker. Bei zusätzlichen Leiterplatten, wie vor allem Speicher und Bildschirmansteuerung, werden Stecker mit gleicher Belegung eingesetzt. Dadurch liegen Signale mit einer Bezeichnung auf allen Leiterplatten stets auf dem gleichen Steckerstift, z. B. Datenbit 8 erscheint immer auf Stift B7. Die Verbindung der einzelnen Leiterkarten ist so auf einfache Art und Weise mit einer gedruckten Verdrahtung möglich, die Kontakte gleicher Bezeichnung miteinander verbindet (Bild 1.13), sofern die Funktion der Schaltung dies zuläßt. Die Anschlußbelegung der Steckverbinder mit dem System-BUS folgt im wesentlichen dem industriellen Vorbild des Mikrorechners K 1520 (Robotron), dessen Gestaltung in [6] ausführlich beschrieben ist. Tabelle 1.1. zeigt die Belegung der Kontakte. Es wird ein 58poliger Steckverbinder benutzt, wobei sowohl indirekte als auch direkt auf den Leiterplatten geätzte Kontakte üblich sind. Die Bestückungsseite hat die Bezeichnung «B», die Lötseite erscheint mit der Bezeichnung «L», der Stecker befindet sich auf der Leiterplatte, die Buchse in der Rückverdrahtung. Tabelle 1.2 enthält die Signalbezeichnung und kurze Erläuterungen. In der Tabelle erscheinen einige Signale, die nicht unmittelbar vom Mikroprozessor stammen. Neben der Stromversorgung ist vor allem das Signal $\overline{MEMDI}$ (Memory disable) auf Kontakt B9 von großer Bedeutung. Es erlaubt die Abschaltung des Speichers der jeweiligen Leiterplatte. Durch Low-Pegel an diesem Stift kann der Speicher komplett dem Zugriff des Mikroprozessors

Tabelle 1.1. Steckerbelegung System-BUS

Signal-name	B/C	A	Signal-name
5P	29	29	5P
12P	28	28	12P
BAI	27	27	BAO
HALT	26	26	MI
RDY	25	25	RFSH
IORQ	24	24	WAIT
INT	23	23	NMI
00	22	22	IODI
00	21	21	TAKT
BUSRQ	20	20	RESET
AB1	19	19	AB0
AB3	18	18	AB2
AB5	17	17	AB4
AB7	16	16	AB6
5N	15	15	5N
AB9	14	14	AB8
AB11	13	13	AB10
AB13	12	12	AB12
AB15	11	11	AB14
IEI	10	10	IEO
MEMDI	09	09	MREQ
RD	08	08	WR
DB0	07	07	DB1
DB2	06	06	DB3
DB4	05	05	DB5
DB6	04	04	DB7
5PG	03	03	5PG
00	02	02	00
00	01	01	00

Tabelle 1.3. Signalbezeichnungen Koppel-BUS

	Signal-name	B/C	A	Signal-name	
	00	29	29	00	
	00	28	28	00	
	12N,			12N	
1)	12NR	27	27	12NR	1)
2)	IEI1	26	26	IEO1	
	C/TRGQ	25	25	ZC/TO0	
3)	C/TRG1	24	24	ZC/TO1	3)
	C/TRG2	23	23	ZC/TO2	
4)	SUE	22	22	C/TRG3	5)
5)	MEMDI2	21	21	MEMDI1	
6)	SA	20	20	RDYAS	7)
7)	MEMPR	19	19	MEMAS	
	IOPR	18	18	IOAS	
	PASTB	17	17	PBSTB	
	PA1	16	16	PA0	
	PA3	15	15	PA2	
	PA5	14	14	PA4	
	PA7	13	13	PA6	
8)	PARDY	12	12	PBRDY	8)
	PB1	11	11	PB0	
	PB3	10	10	PB2	
	PB5	9	9	PB4	
	PB7	8	8	PB6	
10)	IEP	7	7	UM	7)
	PRDY	6	6	PSTB	
8)	RESET	5	5	00	
	TAKT	4	4	00	8)
9)		3	3		9)
		2	2		
	5P	1	1	5P	

entzogen werden und ein anderer oder ein externes Gerät an dessen Stelle treten. Das BUS-System nach [6] verfügt noch über einen zweiten Steckverbinder, der Signale mit der Bezeichnung MEMDI1 und MEMDI2 führt (Tabelle 1.3.). Diese Signale erlauben es, den Speicher baugruppenweise ein- und abzuschalten.

Auf dem eigentlichen System-BUS fehlen sie leider, so daß beim Einsatz von Leiterplatten einfacher Breite mit nur einem Steckverbinder andere, für den Amateur weniger wichtige Leitungen hierfür dienen müssen. Die Leitungen BAI und BAO arbeiten als Eingang und Ausgang für die Steuerung eines von der CPU unabhängigen, von einer Baugruppe gesteuerten direkten Speicherverkehrs. Diese Steuerleitungen werden

- 1) 12N: – 12 V; 12NR: Rohspannung zur Erzeugung von Sonderspannungen im Modul
- 2) Zweite Prioritätenkette zur Realisierung verschiedener Prioritäten
- 3) Zähler-/Zeitgebersignal
- 4) Spannungsüberwachung (bei CMOS-Speichern)
- 5) Speichererweiterung über 64 kByte
- 6) Signal zur Netzein- und -ausschaltung
- 7) Anschluß Prüfrechner
- 8) Mehrrechnerkopplung (bei Realisierung durch PIO-Schaltkreis auf der ZVE)
- 9) Sonderspannung oder Anwendersignal
- 10) Interrupt enable – paralleles Signal für externe Beschleunigungsschaltung Interrupt-prioritätskette.

1. Schaltungstechnik

Tabelle 1.2. Signalbezeichnung System-BUS

Signalname	Signalbedeutung
DB0 bis DB7	Datenbus: Leitungen führen beim Datenaustausch die Befehls- bzw. Dateninformation.
AB0 bis AB15	Adreßbus: Leitungen führen die Adresse für den Informationsaustausch mit dem Speicher bzw. E/A-Gerät. AB0 bis AB15 sind mit $\overline{MREQ}$ als Speicheradresse gültig (64 kByte). AB0 bis AB7 sind mit $\overline{IORQ}$ als E/A-Geräteadresse gültig (256 Eingabe- und 256 Ausgabeadressen). AB8 bis AB15 sind mit $\overline{IORQ}$ gültig und enthalten bei Ein- und Ausgabebefehlen den Inhalt vom CPU-Register A bzw. B (je nach Befehlsart) bzw. den Inhalt des oberen Byteregisters des DMA-Kanals. AB0 bis AB6 sind mit $\overline{RFSH}$ als Refreshadresse für das Auffrischen dynamischer RAM gültig.
$\overline{MREQ}$	Speichertransfergusuch: Signal zeigt an, daß der Adreßbus eine gültige Adresse für eine Speicherlese- bzw. -schreiboperation führt.
$\overline{IORQ}$	Ein-/Ausgabetransfergusuch: Signal zeigt an, daß der Adreßbus im unteren Byte (AB0 bis AB7) eine gültige E/A-Geräteadresse führt. Das Signal wird von der CPU auch erzeugt, wenn ein Interruptgesuch von der CPU akzeptiert wurde und der neue Befehl bzw. Interruptvektor auf den Datenbus gelegt werden darf. (siehe M1)
$\overline{RD}$	Lesen: Signal zeigt an, daß durch die CPU bzw. den DMA-Kanal Informationen (Daten oder Befehle) vom Speicher bzw. E/A-Gerät gelesen werden sollen.
$\overline{WR}$	Schreiben: Signal zeigt an, daß von der CPU bzw. von der DMA-Einheit gültige Daten auf den Datenbus gelegt wurden, die im Speicher einzutragen bzw. vom E/A-Gerät zu übernehmen sind.
$\overline{RFSH}$	Auffrischen: Signal zeigt an, daß die unteren 7 bit des Adreßbusses eine Refreshadresse zum Auffrischen dynamischer RAM bilden. In Verbindung mit $\overline{MREQ}$ kann dieses Signal benutzt werden, jeweils eine Refresh-Leseoperation durchzuführen.
$\overline{MI}$	Befehlslesezyklus: Signal zeigt an, daß der laufende CPU-Zyklus ein Befehlslesezyklus des auszuführenden Befehls ist. In Verbindung mit dem Signal $\overline{IORQ}$ zeigt es an, daß ein Interruptgesuch akzeptiert wurde und der Interruptvektor bzw. Interruptbefehl auf den Datenbus zu legen ist.
$\overline{HALT}$	CPU-Halt: Das Signal zeigt an, daß sich die CPU im Halt-Zustand befindet und zur weiteren Arbeit auf einen Interrupt wartet.
$\overline{BUSRQ}$	Gesuch auf direkten Speicherverkehr: Das Signal zeigt der CPU an, daß ein Gesuch auf direkten Speicherverkehr durch eine DMA-Einheit gestellt wurde. Das Signal besitzt bezüglich der CPU-Unterbrechung höchste Priorität.
$\overline{INT}$	Maskierbares Interruptgesuch: Signal kennzeichnet ein Interruptgesuch eines peripheren Gerätes in die CPU.
$\overline{NMI}$	Nichtmaskierbares Interruptgesuch: Signal kennzeichnet ein Interruptgesuch an die CPU. Es besitzt höhere Priorität als $\overline{INT}$ und wird unabhängig vom Zustand des CPU-internen Interrupt-Enable-Flip-Flops akzeptiert.
$\overline{WAIT}$	Warten: Signal zeigt der CPU an, daß der Speicher bzw. das E/A-Gerät nicht für einen Datenaustausch „bereit“ ist. (Die CPU tritt so lange in einen Wartezustand ein, wie das Signal aktiv ist.)

Signalname	Signalbedeutung
RDY	Bereit: Signal zeigt an, daß der angesprochene Speicher oder das angesprochene E/A-Gerät am Bus vorhanden ist und für Lese- oder Schreiboperationen zur Verfügung steht.
RESET	Rücksetzen: Signal dient als zentrales Rücksetzsignal im Rechner.
MEMDI	Speichersperrung: Das Signal dient zur Sperrung aller Speichermodule für Lese- und Schreiboperationen. Die Signale $\overline{\text{MREQ}}$ und $\overline{\text{MI}}$ werden davon nicht beeinflusst. Das Signal wird bei Simulation des Speichers durch z. B. eine Bedieneinheit bzw. einen externen Speicher oder zur CPU-unabhängigen Beendigung eines Speicherzyklus benutzt.
IODI	Ein-/Ausgabesperre: Das Signal dient zur Sperrung aller Ein-/Ausgabebaugruppen für Schreib- und Leseoperationen. Die Signale $\overline{\text{IORQ}}$ und $\overline{\text{MI}}$ werden davon nicht beeinflusst. Das Signal wird bei der Simulation des E/A-Systems verwendet. Die Verbindung des Signals auf der Steckeinheit der höchsten Prioritätsstufe mit $\overline{\text{IEI}}$ verhindert zusätzlich die automatische Eingabe von Interruptvektoren bzw. -befehlen nach Interruptanerkennung und erlaubt ihre Handeingabe.
IEI	Interrupt-Freigabe-Eingang: Das Signal kennzeichnet, daß die Funktionseinheiten mit der höheren Priorität sich nicht im Interrupt-Behandlungsstatus befinden oder bei Interruptanerkennung keinen Interrupt anfordern.
IEO	Interrupt-Freigabe-Ausgang: Bedeutung wie $\overline{\text{IEI}}$, jedoch mit Einbeziehung der vorliegenden Steckeinheit. Die Leitung ist direkt mit $\overline{\text{IEI}}$ der nachfolgenden Steckeinheit zu verbinden.
$\overline{\text{BAI}}$ und $\overline{\text{BAO}}$ (BUSAK)	Anerkennung des direkten Speicherverkehrs (Eingang/Ausgang): Die Signale $\overline{\text{BAI}}$ und $\overline{\text{BAO}}$ bilden am Bus eine prioritätsbestimmende Kette zur Durchschaltung des Signals $\overline{\text{BUSAK}}$. Dabei bildet $\overline{\text{BAI}}$ das Eingangssignal (BUSAK-Input), $\overline{\text{BAO}}$ (BUSAK-Output) das weitergeleitete Anerkennungssignal am Ausgang der Steckeinheit. $\overline{\text{BUSAK}}$ ist ein von der CPU erzeugtes Signal, mit dem sie anzeigt, daß $\overline{\text{BUSERQ}}$ akzeptiert wurde und Datenbus, Adreßbus und 3-state-Steuerausgänge in den hochohmigen Zustand geschaltet sind.
TAKT	Systemtakt: Die zugeordnete Leitung führt den zentralen Systemtakt. Er entspricht dem Grundtakt der CPU und steht den Busteilnehmern zur zeitlichen Synchronisation interner Vorgänge zur Verfügung.
5P	+ 5 V $\pm 0,25$ V im Betriebsfall; = 2 V bei externer Batteriestützung von RAM-Speichern im Zustand Datensicherung nach Netzspannungsausfall; wenn keine externe Speicherstützung erfolgt, ist „5PG“ mit 5P zu verbinden.
5N	- 5 V $\pm 0,25$ V; Versorgungsspannung ungestützt
12P	+ 12 V $\pm 0,6$ V; Versorgungsspannung ungestützt
00	Zentrales Bezugspotential (Masse)

schon in der Industrie so gut wie nicht benutzt und sind für den Amateur geradezu bedeutungslos.

In den meisten Fällen kann der Amateur jedoch mit dem Signal MEMDI zur Speicherumschaltung auskommen.

Einige der BUS-Leitungen erfordern beim Signalgeber keinen Tristate-Ausgang, sondern nur einen offenen Kollektor, so daß auch mehrere Baugruppen gleichzeitig aktiv sein können («Verdrahtetes ODER-Glied»). Es sind dies die Anschlüsse für BUSRQ, INT, NMI, WAIT, RESET, RDY, MEMDI und IODI. Die Zeitfolge der Signalpegel auf den Leitungen des BUS-Systems folgt der des Mikroprozessors. Sie wird durch die Flanken der Taktimpulse gesteuert und unterscheidet sich bei den einzelnen Befehlen. Der Datenteil des Buchs (Abschnitt 5.4.)

enthält auch die Zeitdiagramme für Speichertzugriff, IN/OUT-Verkehr usw.

Um Wiederholungen zu vermeiden, soll hier nicht weiter darauf eingegangen werden. Dem Leser ist jedoch zu empfehlen, sich die Zeitdiagramme gründlich anzusehen und bei der Fehlersuche an Schaltungen zu Rate zu ziehen.

1.2.3. Anschluß von Tastatur, Kassettenmagnetbandgerät und Fernschreiber (Drucker)

Die wichtigsten Peripheriegeräte sind Tastatur und Bildschirm. Aber es ist auch notwendig, Programme auf externen Datenträger, wie Lochstreifen oder Magnetbandkassette, zu speichern. Wertvoll ist die Möglich-

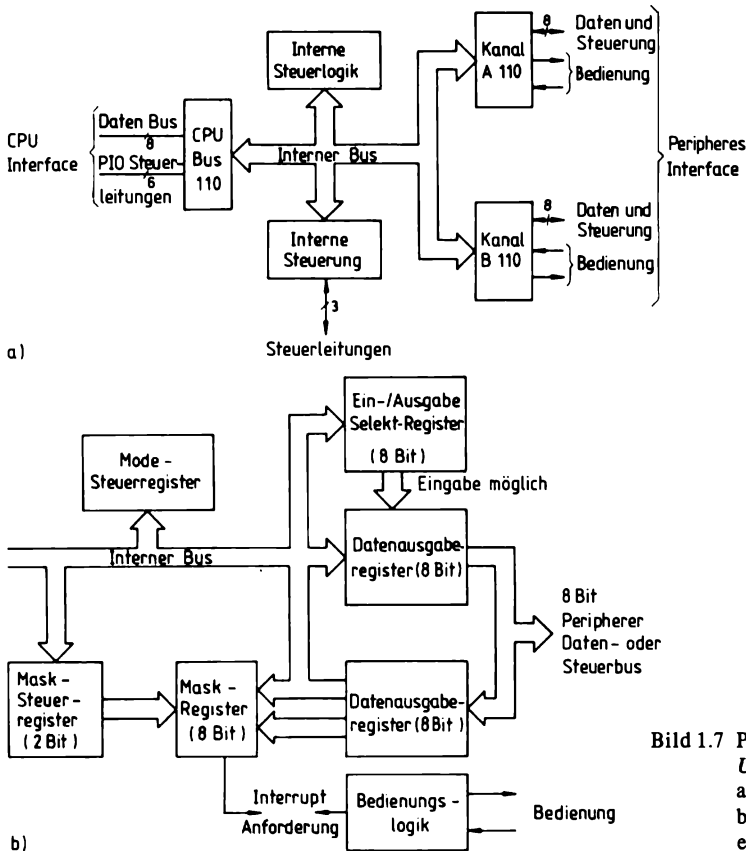


Bild 1.7 PIO-Baustein

UB 855 D;

a - Übersichtsschaltplan,

b - Übersichtsschaltplan eines Kanals

keit, Programme und Daten auszudrucken. Beide Forderungen erfüllt ein Fernschreiber mit Lochstreifensender.

In diesem Fall wird die Leiterplatte des Grundgerätes um Netzteil, Taktoszillator, Resetschaltung und Fernschreiberkopplung erweitert und ergibt mit dem Fernschreiber einen eigenständigen Mikrocomputer.

Die Kopplung der genannten Peripheriegeräte mit dem Computer erfolgt über den Schaltkreis *UB 855 D* (PIO). Die Bezeichnung «PIO» entstand aus der Abkürzung von «Parallel-Input/Output», d. h., der Schaltkreis kann Signale gleichzeitig über

mehrere Leitungen ausgeben oder empfangen. Um die Funktion der nachfolgenden Schaltungen besser verständlich darzustellen, sollen zunächst einige Informationen zum *UB 855 D* folgen.

Bild 1.7 zeigt die interne Struktur des Bauelementes, Tabelle 1.4. enthält die Bezeichnung der Anschlüsse und Erläuterungen dazu.

Die nach außen führenden Anschlüsse sind in die «Kanäle» A und B zu je 8 Leitungen zusammengefaßt und belegen je eine Adresse. Bei manchen Anwendungsfällen soll das angeschlossene Gerät die Daten-

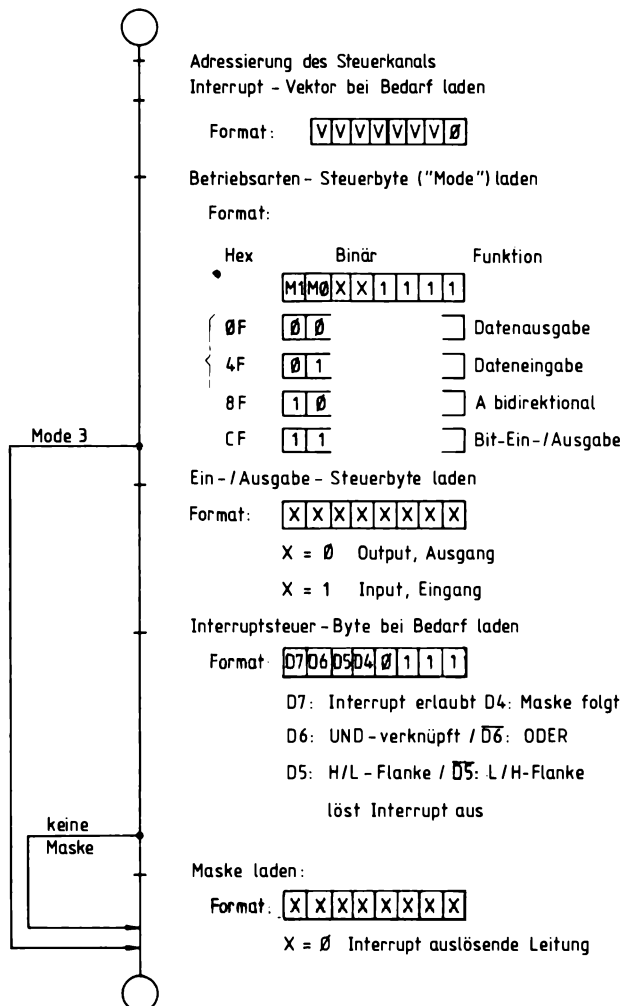


Bild 1.8
Programmierung des *UB 855 D*

Tabelle 1.4. Anschlußbezeichnungen U 855 PIO

Anschlüsse	Bedeutung
D0...D7 (Data Bus)	Datenbus (bidirektional, tri-state) Dieser Bus wird für den Transfer aller Daten und Steuerworte zwischen der CPU und der PIO verwendet. D0 ist das niederwertigste bit des Busses.
B/ $\overline{A}$ SEL (Port B or A Select)	Auswahl Kanal B oder A (Eingang) Dieser Anschluß definiert, welcher der beiden PIO-Kanäle angesprochen ist. Low $\hat{=}$ Kanal A High $\hat{=}$ Kanal B Im praktischen Einsatz wird häufig die Adresse A0 der CPU für diese Auswahl-funktion verwendet.
C/ $\overline{D}$ SEL (Control or Data Select)	Auswahl Steuerwort/Datenwort (Eingang) Dieser Anschluß definiert die Art der Information bei einem Transfer zwischen CPU und PIO. High-Pegel an C/ $\overline{D}$ SEL während einer CPU-Schreiboperation an die PIO hat zur Folge, daß die PIO den Inhalt des Datenbusses während dieser Zeit als Steuerwort interpretiert. Low-Pegel an diesem Anschluß bedeutet, daß der Datenbus zum Transfer von Daten zwischen CPU und PIO benutzt wird. Im praktischen Einsatz wird häufig die Adresse A1 für die angegebene Auswahlfunktion verwendet.
$\overline{CS}$ (Chip Select)	Bausteinauswahl (Eingang, Low-aktiv) Low-Pegel an diesem Anschluß versetzt die PIO in die Lage, Steuerworte oder Daten von der CPU während einer CPU-Schreiboperation zu empfangen bzw. während einer CPU-Leseoperation Daten an die CPU zu senden. Im praktischen Einsatzfall wird dieses Signal meist durch Dekodierung der Adressen A2...A7 gewonnen, wenn A0 und A1 bereits für B/ $\overline{A}$ SEL und C/ $\overline{D}$ SEL verwendet wurden.
$\overline{MI}$ (Machine Cycle one Signal)	Maschinenzyklus 1-Signal (Eingang, Low-aktiv) Dieses von der CPU ausgesandte Signal dient der Synchronisierung innerer Operationen in der PIO. Sind die Signale $\overline{MI}$ und $\overline{RD}$ aktiv, dann befindet sich die CPU im Befehlshole-zyklus (dieser Fall ist für die RETI-Erkennung durch die PIO von Bedeutung). Sind $\overline{MI}$ und $\overline{IORQ}$ aktiv, quittiert die CPU eine Interruptanforderung. $\overline{MI}$ hat in der PIO zwei weitere Funktionen: 1. Das Signal $\overline{MI}$ synchronisiert die Interruptlogik der PIO. 2. Ist $\overline{MI}$ aktiv ohne aktives $\overline{RD}$ - oder $\overline{IORQ}$ -Signal, geht die PIO in den RESET-Zustand.
CP (Clockpulse)	Systemtakt (Eingang) Die U 855 benutzt den Standardsystemtakt (Einphasentakt) des U-880-Systems. Dieser sichert die Synchronisation der Signale untereinander ab.
$\overline{IORQ}$ (Input/Output Request)	Ein-/Ausgabeanforderung (Eingang, Low-aktiv) Das Signal $\overline{IORQ}$ wird in Verbindung mit den Signalen B/ $\overline{A}$ SEL, C/ $\overline{D}$ SEL, $\overline{CS}$, $\overline{RD}$ und $\overline{MI}$ zum Transport von Befehlen und Daten zwischen CPU und PIO verwendet. Sind $\overline{CS}$, $\overline{RD}$ und $\overline{IORQ}$ aktiv, dann überträgt der mit B/ $\overline{A}$ selektierte Kanal Daten zur CPU (Leseoperation). Sind $\overline{CS}$ und $\overline{IORQ}$ aktiv, $\overline{RD}$ und $\overline{MI}$ jedoch nicht aktiv, dann schreibt die CPU Daten- oder Befehlsworte (je nach Signal-Vorgabe am C/ $\overline{D}$ -Eingang) in den selektierten Kanal ein (Schreiboperation). Sind $\overline{MI}$ und $\overline{IORQ}$ aktiv, bedeutet das, daß die CPU eine Interruptanforderung quittiert, und derjenige interruptanfordernde Kanal, der die höchste Priorität hat, legt seinen Interruptvektor auf den Datenbus.

Anschlüsse	Bedeutung
$\overline{RD}$ (Read)	Lesezyklus-Signal (Eingang, Low-aktiv) Mit aktivem $\overline{RD}$-Signal wird seitens der CPU eine Speicher- oder I/O-Leseoperation gekennzeichnet. Um Daten von der PIO zur CPU zu übertragen, wird das $\overline{RD}$-Signal in Verbindung mit den Signalen B/A SEL, C/D SEL, $\overline{CS}$ und $\overline{IORQ}$ verwendet.
IEI (Interrupt Enable In)	Interrupt-Freigabe-Eingang (Eingang, High-aktiv) Dieses Signal wird benötigt, um eine Interruptprioritätskette zu bilden (IEI ist mit IEO des nächsthöherpriorisierten peripheren Schaltkreises zu verbinden). High-Pegel bedeutet, daß momentan kein Interrupt höherer Priorität abgearbeitet oder angemeldet wird. (Ausnahme bei noch nicht bestätigtem Interrupt des aktuell höchstpriorisierten peripheren Schaltkreises und RETI-Dekodierung).
IEO (Interrupt Enable Out)	Interrupt-Freigabe-Ausgang (Ausgang, High-aktiv) Das IEO-Signal ist ein weiteres für die Funktion der Interruptprioritätskette erforderliches Signal. IEO führt nur dann High-Pegel, wenn der Eingang IEI desselben Schaltkreises High-Pegel erhält und kein eigener Interrupt abgearbeitet oder angemeldet wird (Ausnahme bei noch nicht bestätigtem Interrupt des aktuell höchstpriorisierten peripheren Schaltkreises und RETI-Dekodierung). Auf diese Weise kann verhindert werden, daß Funktionseinheiten mit niedriger Priorität einen Interrupt auslösen können, solange von einer Funktionseinheit mit höherer Priorität bei der CPU eine Interruptbehandlung angefordert wird.
$\overline{INT}$ (Interrupt Request)	Interruptanforderung (Ausgang, Open-drain, Low-aktiv) Aktivierung des Ausganges signalisiert der CPU die Anmeldung eines Interrupts.
A0...A7 (Port A Bus)	Kanal-A-Bus (bidirektional, Tristate) Dieser 8-bit-Bus wird zum Transfer von Daten und/oder Status- oder Steuerinformationen zwischen Kanal A der PIO und externen angeschlossenen Geräten benutzt. A0 ist das niederwertigste bit des Busses.
$\overline{ASTB}$ (Port A Strobe Pulse)	Kanal A-Strobe (Eingang, Low-aktiv) Die Bedeutung dieses Signals hängt von der Betriebsart ab, die für Kanal A gewählt wurde: 1. MODE 0: Der Strobeimpuls wird von der Peripherie abgegeben, um die Daten aus dem Ausgaberegister zu übernehmen. Das Ende des Strobeimpulses (positive Flanke) gilt als Quittierung der erfolgten Übernahme. 2. MODE 1: Der Strobeimpuls wird von der Peripherie abgegeben, um Daten von der Peripherie in das Eingaberegister des Kanals zu laden. Die Daten werden in die U855 geladen, wenn das Signal aktiv ist. 3. MODE 2: Wenn das Signal aktiv ist, werden Daten vom Ausgaberegister des Kanals A auf den bidirektionalen Bus des Kanals A gelegt. Das Ende des Strobeimpulses (positive Flanke) gilt als Bestätigung für den Empfang der Daten. 4. MODE 3: Der Strobeimpuls ist intern verboten.
ARDY (Register A Ready)	Quittung Kanal A (Ausgang, High-aktiv) Bedeutung ist abhängig von Betriebsart: 1. MODE 0: Das Signal wird aktiv, um anzuzeigen, daß das Ausgaberegister des Kanals geladen ist und daß die Daten abgerufen werden können. Nach Übernahmequittierung (steigende Flanke des Strobeimpulses) durch die periphere Schaltung wird das Signal inaktiv. 2. MODE 1: Das Signal ist aktiv, wenn das Eingaberegister des Kanals gelesen und es bereit ist, Daten vom peripheren Gerät zu übernehmen.

1. Schaltungstechnik

Anschlüsse	Bedeutung
	3. MODE 2: Das Signal ist aktiv, wenn Daten ins Ausgaberegister vom Kanal A geladen und für einen Transfer zum peripheren Gerät verfügbar sind. In dieser Mode liegen die Daten am Kanal-A-Datenbus nicht an, sofern nicht $\overline{\text{ASTB}}$ aktiv ist.
	4. MODE 3: Das Signal ist nicht verwendbar und liegt auf Potential «Low».
B0...B7 (Port B Bus)	Kanal-B-Bus (bidirektional, Tri-state) Dieser 8-bit-Bus wird zum Transfer von Daten und/oder Status- oder Steuerinformationen zwischen Kanal B der PIO und extern angeschlossenen Geräten benutzt. Der Kanal-B-Bus ist zur Ansteuerung von Darlingtonstufen ausgelegt (minimal 1,5 mA bei $U_{OH} = 1,5 \text{ V}$). B0 ist das niederwertigste bit des Busses.
$\overline{\text{BSTB}}$ (Port B Strobe Pulse)	Kanal B-Strobe (Eingang, Low-aktiv) Bedeutung entsprechend $\overline{\text{ASTB}}$ mit der folgenden Ausnahme: In der bidirektionalen Betriebsart des Kanals A werden mit diesem Signal Daten vom peripheren Gerät in das Eingaberegister des Kanals A eingeschrieben.
BRDY (Register B Ready)	Quittung Kanal B (Ausgang, High-aktiv) Bedeutung entspricht ARDY mit der folgenden Ausnahme: In der bidirektionalen Betriebsart des Kanals A ist das Signal «High», wenn das Eingaberegister des Kanals A gelesen und bereit ist, Daten vom peripheren Gerät zu übernehmen.

übergabe oder -übernahme quittieren. Dies kann z. B. sinnvoll sein, um die Übertragungsgeschwindigkeiten einander anzupassen oder um festzustellen, ob das Gerät eingeschaltet wurde. Die Leitungen ARDY und BRDY («READY» = fertig) können dem Gerät mitteilen, daß Daten bereitstehen bzw. übernommen werden können, die Leitungen $\overline{\text{ASTB}}$ und $\overline{\text{BSTB}}$ («STROBE») haben diese Funktion auf seiten des Gerätes. Im Wechselspiel dieser Signale ist ein optimaler Informationsaustausch möglich.

Wie bereits erwähnt, ist das Besondere an den modernen, höher integrierten Schaltkreisen die Programmierbarkeit ihrer Arbeitsweise. Dazu dienen die den Kanälen zugeordneten Steuerports, die über je eine eigene Adresse verfügen. Sie werden je nach gewünschter Betriebsart mit einer Folge von 8-bit-Daten geladen. Da dies im Regelfall beim Start des Programms erfolgt, spricht man von «Initialisierung».

Bild 1.8 zeigt einen Programmablaufplan zur Programmierung des Bauelementes. Für den Anschluß von Tastatur und Fernschreiber sind die Leitungen 0 bis 6 von Kanal A für Ausgabe programmiert, Leitung A7 und Kanal B vollständig arbeiten als Eingabe.

Zur Tastaturschaltung sollen den Erläuterungen einige allgemeine Gedanken vorangehen, die auch für andere Baugruppen zutreffen. Typisch für die Mikrocomputertechnik ist, daß ein Problem häufig sowohl überwiegend mittels Programm («Software») oder überwiegend mit Hilfe einer Schaltung («Hardware») lösbar ist, wie dies am Beispiel der Zeitverzögerung bereits sichtbar wurde.

Die erste Variante ist aus mehreren Gründen vorteilhafter:

a) Der materielle Aufwand ist geringer. Daraus ergeben sich geringere Kosten, eine höhere Zuverlässigkeit durch den einfacheren Aufbau und geringere Mühe bei der Fehlersuche im Störfalle.

b) Programmierbare Eigenschaften sind flexibler nutzbar, mit unterschiedlichen Programmen kann das gleiche Gerät unterschiedliche Aufgaben lösen. So könnte der Tastatur z. B. durch Einlesen eines anderen Programmes vom Kassettenrecorder ein Zeichensatz mit BASIC-Schlüsselwörtern zugeordnet werden. Nur in wenigen Fällen, wenn Zeitforderungen es nicht zulassen, bringt eine Hardware-Lösung Vorteile. Bei der Tastaturabfrage ist dies nicht der Fall. Allerdings ist vielen Amateuren die Programmie-

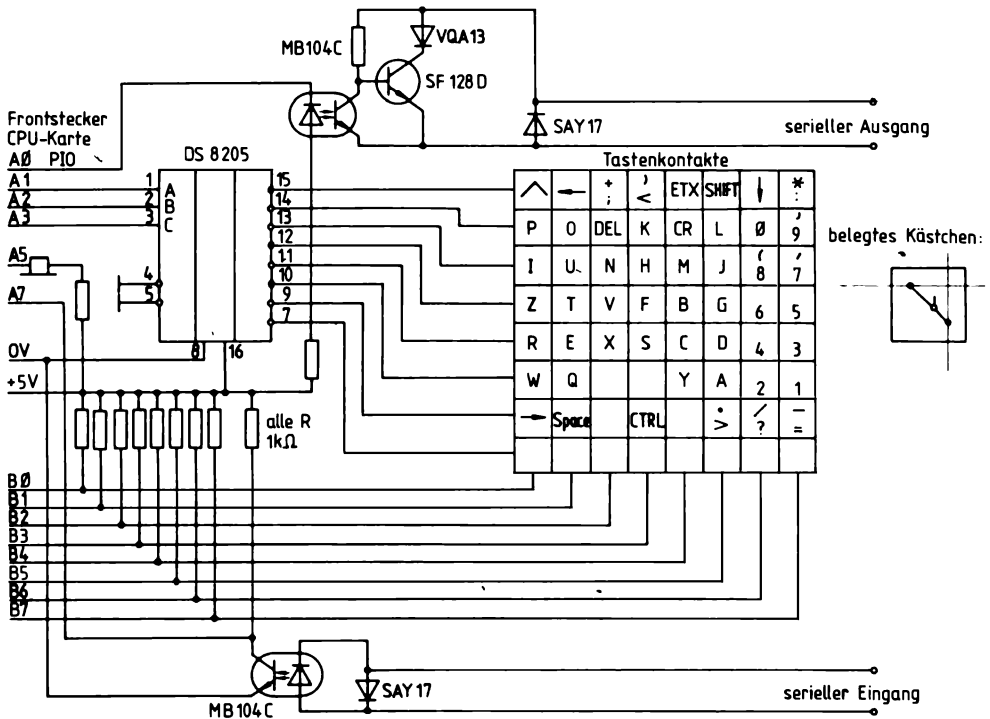


Bild 1.9 Tastaturschaltung und serielle Schnittstelle

rung noch nicht ganz «geheuer», und sie lösen Probleme deshalb lieber mittels Hardware.

Die Tastaturschaltung nach Bild 1.9 arbeitet mit nur 3 Leitungen von Kanal A des PIO, die ein Dekoder aufspaltet, so daß eine Matrixschaltung entsteht. Die Tasten verbinden Spaltenleitungen mit Zeilenleitungen, wenn sie gedrückt sind. Der Dekoder sorgt dafür, daß mit Sicherheit nur eine Zeile Signal führen kann. Das Tastaturprogramm ist in Abschnitt 2.4. beschrieben, so daß hier nähere Erläuterungen nicht nötig sind. Bild 1.9 zeigt auch den Anschluß der Leitung für Fernschreiber oder andere mit 2- bzw. 4-Draht-Leitung arbeitende Geräte. Bekanntlich arbeiten Fernschreiber und einige andere Geräte mit serieller Informationsübertragung, d. h., die Datenbits liegen nicht auf mehreren Leitungen gleichzeitig an, sondern folgen einander auf einer Leitung in einem bestimmten Zeitraster. Bei langsamen

Geräten arbeitet man dabei mit der sogenannten asynchronen Datenübertragung, wobei meist ein Strom geschaltet wird. Jedes Zeichen kündigt sich durch eine Unterbrechung des im Ruhezustand fließenden Stromes (meist 20 bis 40 mA) an. Nach der durch die Übertragungsgeschwindigkeit festgelegten Zeit für 1 bit folgen auf dieses Startbit 5 bis 8 Informationsbits, bei denen die Pegel entsprechend den Daten liegen. Danach folgen 1 bis 2 Stopp-bits, bei denen der Strom immer eingeschaltet ist. Dadurch kann das Gerät das Startbit eines nachfolgenden Zeichens mit Sicherheit erkennen.

Bild 1.10 zeigt den zeitlichen Verlauf eines asynchronen, seriellen Zeichens.

Die Fernschreibtechnik wendet seit Jahren 5 Informationsbits mit historisch entstandener, nicht systematischer Kodierung an. Bei Mikrocomputern hat sich der systematische 7-bit-ASCII-Code durchgesetzt, der in Tabelle 1.5. gezeigt ist. Im Sortiment

Tabelle 1.5. 7-bit-ASCII-Kode

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0 0000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1 .0001	DLE	DC 1	DC 2	DC 3	DC 4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	VS
2 0010	Space	!		#	\$	%	&	'	(	)	*	+		-		/
3 0011	0	1	2	3	4	5	6	7	8	9			<	=	>	?
4 0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5 0101	P	Q	R	S	T	U	V	W	X	Y	Z	[		]	^	_
6 0110		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7 0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

der programmierbaren Peripherieschaltkreise gibt es auch einige, die speziell für alle Arten der seriellen Datenübertragung entworfen sind. In Abschnitt 3.4. folgen dazu einige Ausführungen.

Da der PIO jedoch wesentlich schneller arbeitet als z. B. herkömmliche Fernschreiber, kann er auch eine langsame serielle Datenausgabe steuern. Um auch andere Geräte (z. B. Amateurfunkstationen) problemlos anschließen zu können, trennen optoelektronische Koppler die Leitung galvanisch vom Mikrocomputer. Die Einspeisung des Linienstromes erfolgt wie bei einem Fernschreiber durch das angeschlossene externe Gerät. Die Schaltung verwendet je eine PIO-Leitung, um auch Duplex-Verkehr zu erlauben, bei dem gleichzeitig gesendet und empfangen werden kann. Leitung A0 ist die Ausgabe. Mit L-Pegel schaltet der Strom durch die Leuchtdiode des Kopplers, der

Transistor SF 127 schaltet den Linienstrom aus. H-Pegel an diesem Ausgang schaltet den Strom ein, was vereinbarungsgemäß als L-Signal auf der Leitung gilt.

Die Leuchtdiode zeigt den Linienstrom an und verbessert das Schaltverhalten des Transistors. Leitung 7 von Kanal A ist als Eingang programmiert. Ohne Linienstrom liegt der Pegel über dem Widerstand an +5 V. (H-Pegel der Leitung entspricht H-Pegel am PIO). Die Programme für Senden und Empfang von Zeichen folgen in Abschnitt 3.4., hier sei nur das grundsätzliche Prinzip erläutert. Bei einer Ausgabe auf den PIO bleiben die Daten an den Ausgängen bis zur nächsten unverändert, auch wenn der Prozessor inzwischen die PIO-Adressen abschaltet und ein anderes Programm ausführt. Bei der seriellen Zeichenausgabe ist dieses andere Programm eine Zeitschleife (wie in Abschnitt 1.1. erläutert) mit der Länge eines

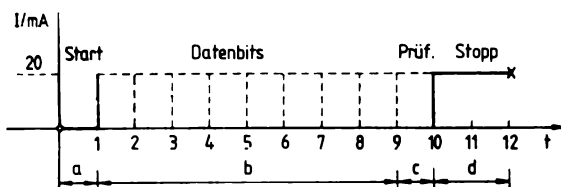


Bild 1.10
Zeichenform bei asynchroner serieller Datenübertragung

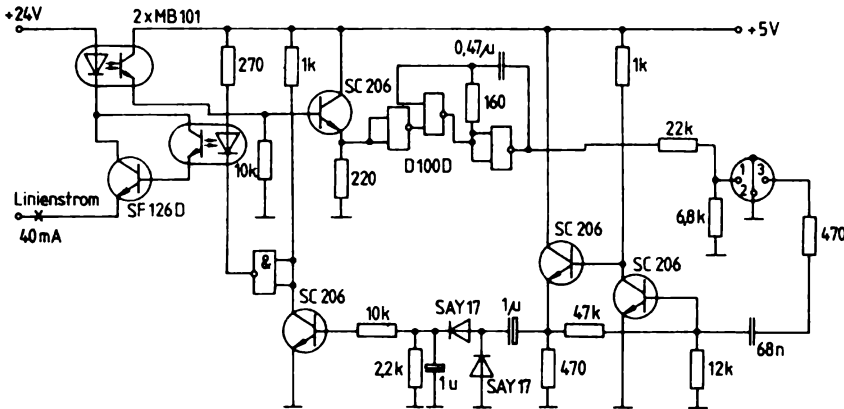


Bild 1.11 Tonmodulator und Demodulator für Kassettenrecorder

bits der entsprechenden Übertragungsgeschwindigkeit. Beim Empfang eines Zeichens erfolgt die Abfrage der Leitung 7 jeweils um die Zeit eines bits verzögert. Da die Zeitschleife mit der quarzstabilisierten Taktfrequenz des Mikrocomputers läuft, arbeitet die Schaltung sehr sicher und ist leicht auf unterschiedliche Geschwindigkeiten einstellbar.

Neben der Möglichkeit, den Computer mit Hilfe einer Fernschreibmaschine oder als Ersatz dafür zu betreiben, gestattet der serielle Anschluß auch auf einfache Weise die Speicherung von Daten und Programmen auf üblichen Kassettenmagnetbandgeräten. Dabei schaltet der Linienstrom einen Tongenerator ein und aus, dessen Ausgangssignal der Recorder aufnimmt. Der Demodulator richtet beim Abspielen die NF gleich und schaltet seinerseits den Linienstrom. Bild 1.11 zeigt den Stromlaufplan.

1.3. Bildschirmsteuerung

Die Bildschirmsteuerung ist der komplizierteste Teil des Eigenbau-Computers, da vorwiegend niedrig integrierte TTL-Schaltkreise verwendet werden. Bild 1.12 zeigt den Übersichtsschaltplan.

Die Baugruppe besteht aus einem Speicher, den der Computer beschreibt oder den

eine Zähschaltung (Zeitbasis) zyklisch ausliest und den Dateninhalt aufbereitet an einen üblichen Fernsehapparat liefert.

Die Schaltung erzeugt mit einem Bildwiederholtspeicher von 1 kByte 16 Zeilen mit je 64 Zeichenplätzen ($16 \times 64 = 1024$). Die Adresse des Zeichens im Bildwiederholtspeicher bestimmt seinen Platz auf dem Bildschirm. In diesem Speicher stehen die darzustellenden Zeichen im ASCII-Code auf je einem Speicherplatz. Der Adressenmultiplexer besorgt die Umschaltung des Bildwiederholtspeichers zwischen Computer und Zeitbasis, wobei die Zugriffsteuerung dem Computer den Vorrang einräumt. Bekanntlich erfolgt die Darstellung eines Fernsehbildes mit einem Elektronenstrahl, der den Bildschirm in Zeilen von links nach rechts und von oben nach unten beschreibt, wobei er in seiner Helligkeit moduliert wird.

Die Zeichen der Schriftzeilen «zerfallen» dadurch in 8 Bildschirmzeilen, so daß der Inhalt des Bildwiederholtspeichers mit Hilfe eines sogenannten Zeichengenerators aufbereitet werden muß. Dieser Zeichengenerator ist ein EPROM, das die Zeichen auf 6×8 Speicherplätzen enthält. An seinen Adreßeingängen liegen die aktuelle Bildschirmzeile und der aktuelle Inhalt des Bildwiederholtspeichers. Die Datenausgänge liefern das Modulationssignal für die Bildpunkte der Zeile.

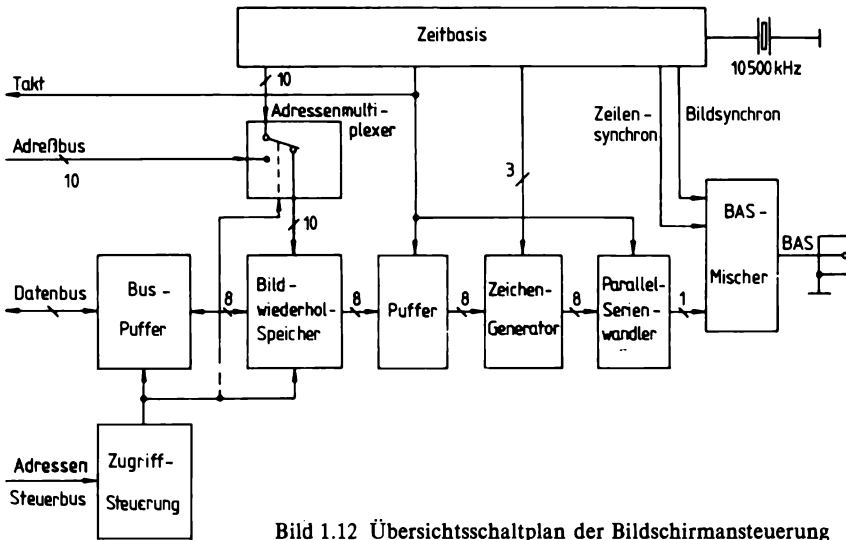


Bild 1.12 Übersichtsschaltplan der Bildschirmansteuerung

Bild 1.13 gibt ein Beispiel für den Zeichenaufbau. Abschnitt 5. enthält das HEX-Listing des Zeichengenerators. Der Parallel-Serienwandler, ein Schieberegister, übernimmt die 8 bit vom Zeichengenerator und gibt sie mit hoher Frequenz nacheinander weiter.

Neben der Helligkeitsinformation benötigt der Fernseher noch Synchronsignale für Zeilen- und Bildwechsel, die ebenfalls von der quarzstabilisierten Zeitbasis kommen.

Im BAS-Mischer erfolgt die phasen- und pegelrichtige Verknüpfung aller für den Aufbau des Bildes nötigen Signale. Der Aufbau der Bildschirmansteuerung erfolgte zunächst zusammen mit der Schaltung nach Bild 1.4 («Grundgerät») auf einer handelsüblichen Universal-Leiterplatte mit den Maßen 215 mm × 170 nm. Diese Lösung erfordert weniger Steckverbinder, als bei kleineren Karten nötig sind.

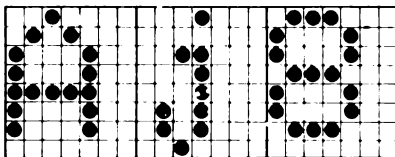


Bild 1.13 Zeichenaufbau auf dem Bildschirm

Es wurde auch eine doppelseitige Leiterplatte dafür entwickelt. Beim Nachbau durch andere Amateure zeigte sich jedoch, daß die Variante mit kleinen Karten leichter zu beherrschen ist.

1.3.1. Bildwiederhol-speicher

Bild 1.14 zeigt den Bildwiederhol-speicher und einen Teil der übrigen Schaltungen. Die im Bild 1.12 als Zugriffsteuerung bezeichnete Schaltung erzeugt das Chip-Select-Signal (Speicheranwahl). Sie dekodiert die Signale der Adreßleitungen A10 bis A15 und das Speicheranforderungssignal vollständig, wie dies beim Grundgerät auch geschieht. Als Dekoder dient hier das Gatter 20, da nur eine Ausgangsleitung nötig ist. Die Anfangsadresse des Bildwiederhol-speichers ist damit auf 0FC00H festgelegt. Natürlich können auch andere Dekoder an dieser Stelle arbeiten, wenn man eine andere Adresse wünscht. Beim Bildwiederhol-speicher müssen bei Zugriff des Prozessors mit dem $\overline{CS}$ -Signal auch die Adreßleitungen der RAM-Schaltkreise umgeschaltet werden; sie liegen deshalb nicht direkt am Adreßbus wie bei den übrigen Speicherschaltungen.

Im Normalfall, wenn der Speicherinhalt

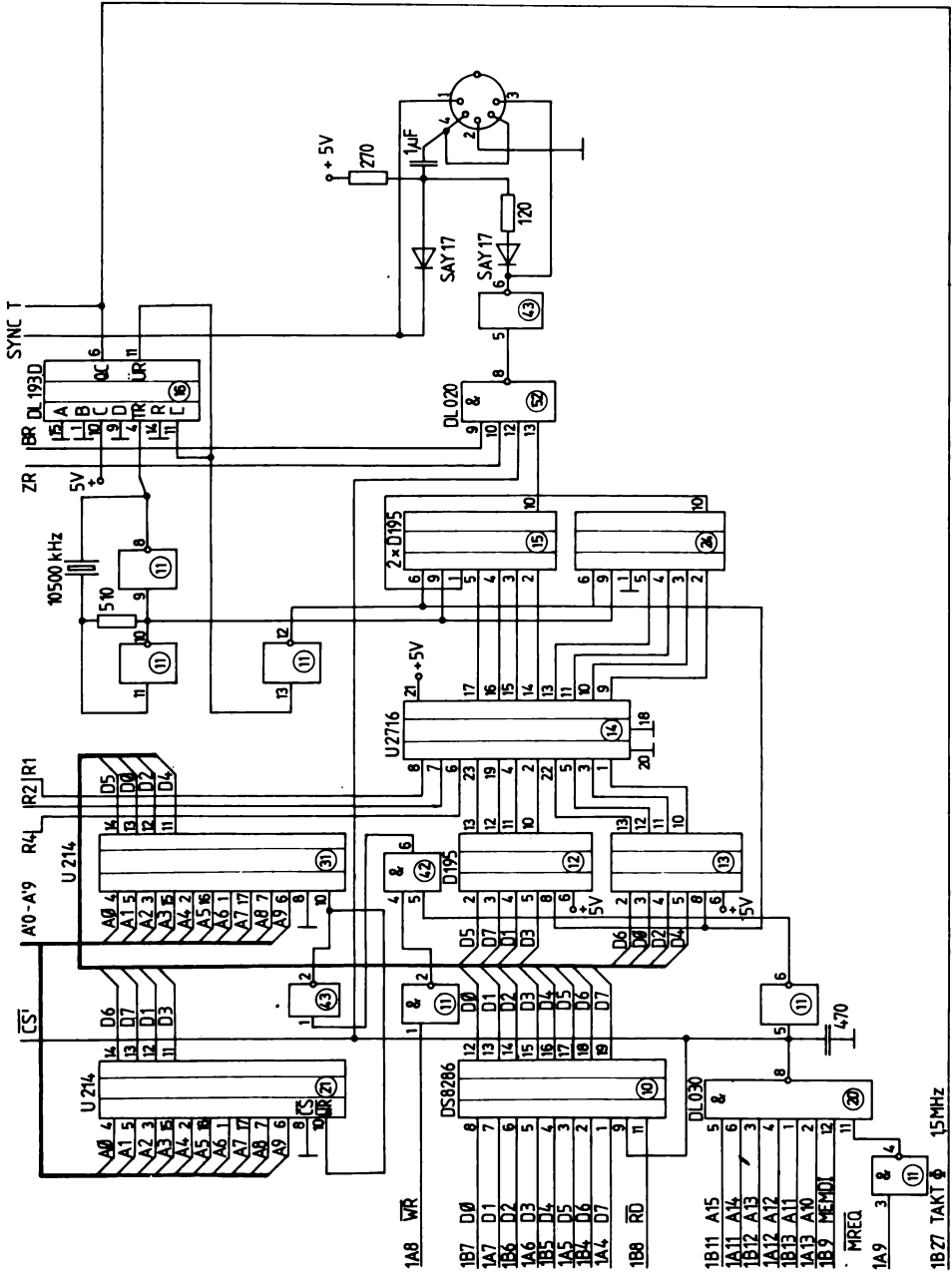


Bild 1.14 Stromlaufplan des Bildwiederholerspeichers (siehe Hinweis auf Seite 272)

auf dem Bildschirm steht, erzeugt die Zeitbasis die Adressen. Da in jedem Fall ein $\overline{\text{CS}}$ -Signal gebraucht wird, liegt Stift 8 der RAM-Schaltkreise (21 und 31) bei Einsatz des *U 214* an Masse. Der pinkompatible Schaltkreis *U 224* dagegen übernimmt die Adressen mit der Flanke des $\overline{\text{CS}}$ in einen Zwischenspeicher, so daß Taktimpulse an $\overline{\text{CS}}$ bei wechselnden Adressen notwendig sind. Er ist damit in dieser Schaltung nicht einsetzbar. Im Gegensatz zu den anderen Speicherschaltungen ist hier auch der Buspuffer *DS 8286* (oder ein vergleichbarer Schaltkreis) unbedingt erforderlich. Da die RAM-Schaltkreise bei der Bilddarstellung ständig gelesen werden (denn die Zeitbasis wählt sie an), geben ihre Datenleitungen ständig Signale ab, die natürlich nicht auf den Datenbus gelangen dürfen. Der Buspuffer muß dem Prozessor sowohl das Beschreiben als auch das Lesen des BildwiederholSpeichers gestatten, denn beim «Rollen» des Bildschirminhalts nach oben gelangen die Zeichen über den Prozessor in die nächst höhere Zeile. Zur Ausgabe auf den Bildschirm gelangen die Zeichen zunächst in einen Puffer aus den Registern 12 und 13, um Zeit für den etwas langsameren Zeichengenerator zu gewinnen. Je nach aktueller Zeile des Fernsehbildes muß aus dem gleichen Zeichen eine andere Zeile an das nachfolgende Schieberegister mit den Schaltkreisen 15 und 24 gelangen. Die Auswahl erfolgt über die Leitungen R1, R2 und R4 von der Zeitbasis. Oberhalb des Schieberegisters zeigt Bild 1.14 den Taktoszillator und eine erste Teilerstufe mit dem Schaltkreis *P 193* (16). Die relativ hohe Frequenz von 10,5 MHz, mit der die Schieberegister arbeiten, hat sowohl für den Aufbau der Schaltung als auch für die Auswahl des Bildschirmteils Bedeutung. Beim Aufbau ist auf kurze Leitungsführung zu achten, die Anforderungen an das Bildschirmgerät folgen weiter unten. Nach sieben Bildpunkten einer Zeile beginnt ein neues Zeichen. Die neue Adresse kommt über die Leitungen A0 bis A9 von der Zeitbasis, an die der Takt T über das Gatter 7 ebenfalls gelangt. Beim Zugriff des Prozessors auf den BildwiederholSpeicher sowie an den Rändern des Bildschirms wird

das Bild über Gatter 52 dunkel getastet. Die nachfolgende UND-Schaltung verknüpft das aufbereitete Helligkeitssignal mit den Synchronimpulsen von der Zeitbasis.

1.3.2. Zeitbasis

Bild 1.15 zeigt die Multiplexer für die Umschaltung der Adressen des BildwiederholSpeichers auf den Prozessorbus und eine Zählerschaltung für das zyklische Auslesen sowie die Erzeugung der Zeilen- und Bildsynchronimpulse. Die Multiplexer schalten mit dem Chip-Select-Signal. Falls die angegebenen Typen nicht zur Verfügung stehen, können auch einfache Gatter als «verdrahtete ODER» deren Zweck erfüllen. Die Zählerkette bietet keine Besonderheiten, die Teilverhältnisse und Frequenzen sind angegeben, so daß Fehler in der Verdrahtung leichter zu finden sind. Der Schaltkreis 41 besorgt die Ansteuerung des Zeichengenerators und tastet abwechselnd 8 Zeilen des Bildes hell und 8 dunkel. Wenn man den Bildschirmspeicher auf 2 kByte erweitert, kann Leitung 7 dieses Schaltkreises als Adresse 6 arbeiten, und der Bildschirm ist mit 32 Zeilen nutzbar. Am Multiplexer verschieben sich die anderen Adreßleitungen, und das CS-Signal muß entsprechend mit A10 verknüpft werden.

1.3.3. Bildschirmgerät

Als Datensichtgerät kann ein handelsübliches Fernsehgerät dienen. Dabei ist auf galvanische Trennung der Masseleitung des Gerätes vom Stromnetz zu achten. In vielen Geräten werden die Versorgungsspannungen direkt mit Gleichrichtern und Thyristoren aus der Netzspannung gewonnen. Diese Fernsehgeräte darf man nur mit Trenntransformator am Computer betreiben. Es könnte sonst die Hochspannung aus dem Netz direkt auf die Computerschaltung gelangen und kostspielige Zerstörungen anrichten. Transportable Fernseher verfügen meist über einen Netztransformator, so daß sie für die Anwendung als Datensichtgerät besser geeignet sind.

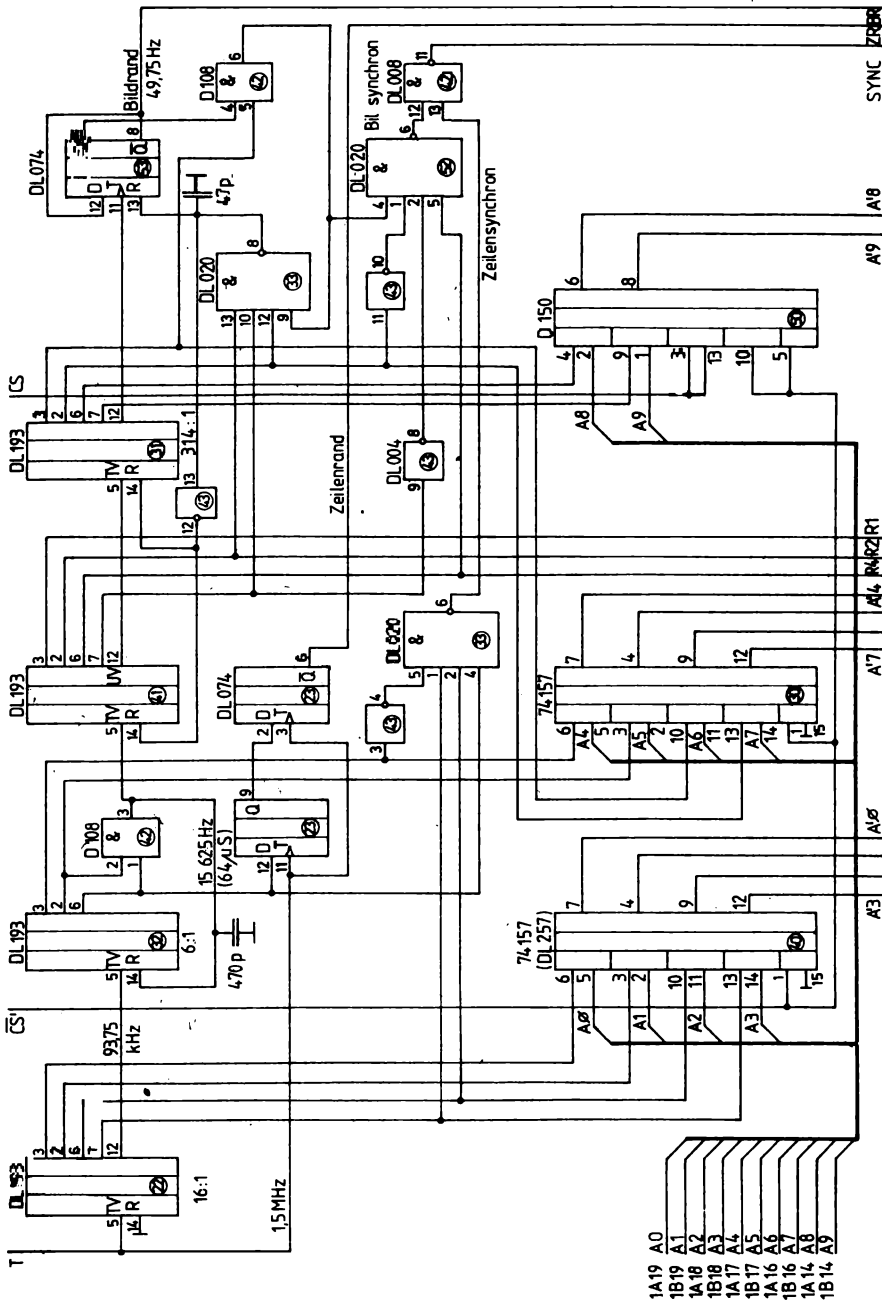


Bild 1.15 Stromlaufplan der Zeitbasis (siehe Hinweis auf Seite 272)

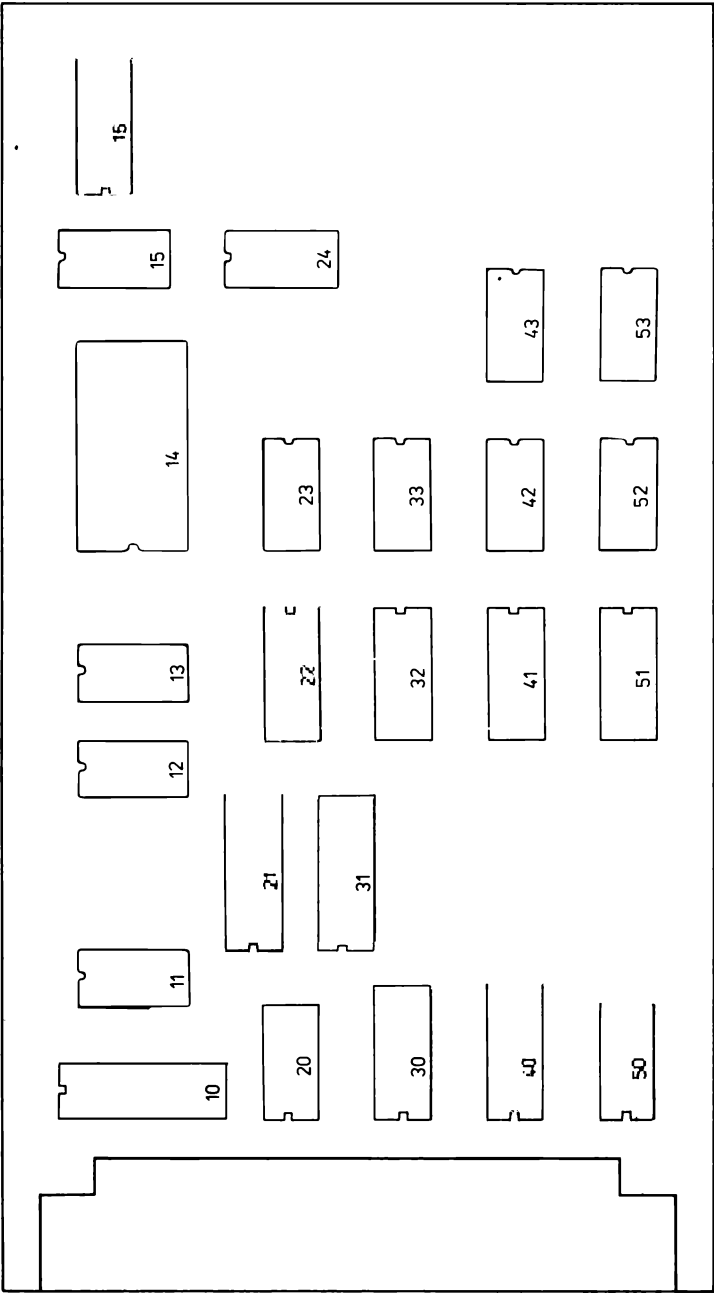


Bild 1.16 Leiterplatte Bildschirmsteuerung;
a – Layout (siehe Anhang, S.249), b – Bestückung, c – Bauelementeanordnung

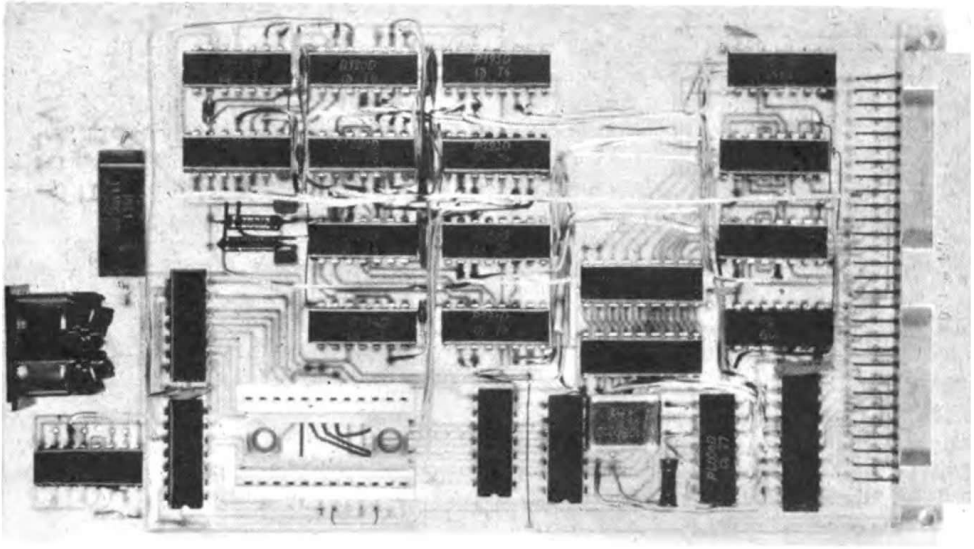
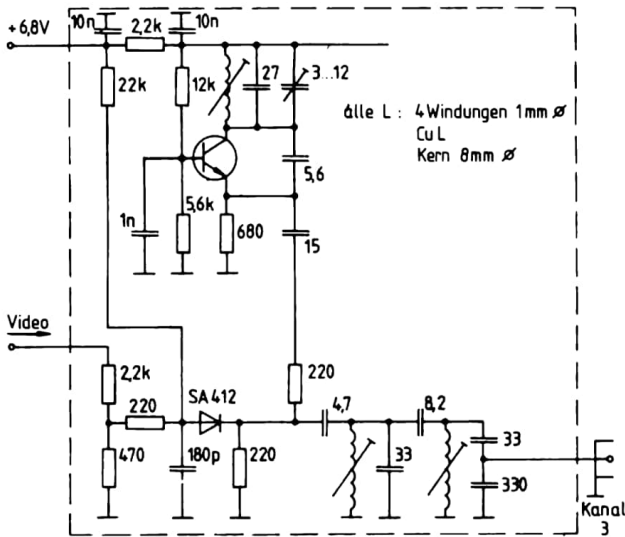
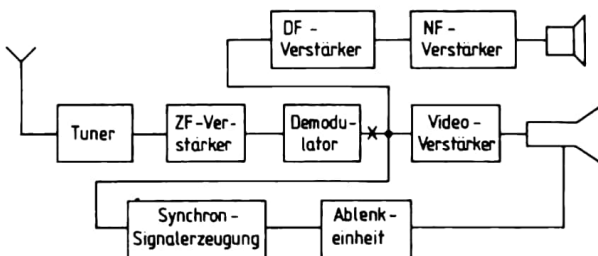


Bild 1.16 b


Bild 1.17
HF-Modulator

Bild 1.18
Übersichtsschaltplan eines Fernsehempfängers

Eine Möglichkeit, ohne Trenntransformator auszukommen, besteht darin, einen HF-Oszillator mit dem BAS-Signal zu modulieren und am Antenneneingang des Fernsehers galvanisch getrennt einzuspeisen. Bild 1.17 zeigt eine bewährte Schaltung. Der Aufbau muß im metallisch geschirmten Gehäuse (z. B. aus Leiterplattenmaterial) erfolgen, damit der Oszillator nicht zum Sender wird und Funkstörungen verursacht. Ein großer Nachteil dieser Lösung ist die etwas schlechtere Bildqualität.

Wie schon erwähnt, erfolgt die Zeichenausgabe mit einer Taktfrequenz von 10,5 MHz. Durch die Modulation, vor allem aber bei der Übertragung im ZF-Teil des Fernsehers, geht Information verloren, und die Darstellung verliert an Schärfe. Deshalb ist es günstiger, das BAS-Signal direkt einzuspeisen. Einige Fernseher verfügen über einen Video-Eingang, den man dafür nutzen kann, jedoch läßt sich die erforderliche Schaltung auch meist leicht nachrüsten, denn der Aufwand beschränkt sich auf die Buchse und einen Elektrolytkondensator von 100 µF. Die Einkopplung erfolgt nach

dem Bildmodulator der Fernseherschaltung.

In einigen moderneren Fernsehern liegt diese Stelle (Bild 1.18) innerhalb von Schaltkreisen, so daß Bild- und Synchronsignale getrennt zugeführt werden müssen. In diesem Fall sind in der Schaltung nach Bild 1.14 die Signale der Klemmen 1 und 3 des Diodensteckers zu verwenden. Da ein Teil der Fernseherschaltung für den Betrieb als Datensichtgerät nicht erforderlich ist, kann man durch Eigenbau beträchtlich billiger kommen. Bild 1.19 zeigt den Übersichtsschaltplan eines Datensichtgerätes, das aus Ersatzbaugruppen eines Fernsehers aufgebaut wurde.

Transportable Farbfernseher sind wegen der geringeren Auflösung der kleineren Schlitzmasken-Bildröhren für den Betrieb am Computer ungeeignet. Die Zeichen zerfallen bei der Darstellung in mehrere Farben und sind dadurch nicht lesbar. Farbfernseher mit großer Bildröhre sind geeignet, können aber mit der einfachen Schaltung nach Bild 1.14 keine unterschiedlichen Farben darstellen.

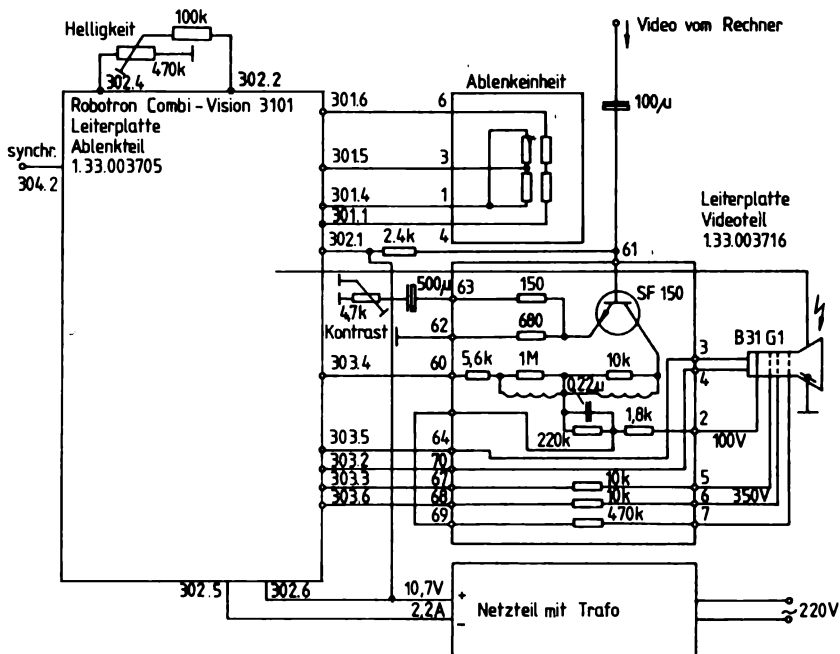


Bild 1.19 Übersichtsschaltplan eines Datensichtgerätes

1.4. Stromversorgung

Die Anforderungen an die Stromversorgung eines Mikrocomputers entsprechen bezüglich der Toleranzen denen bei herkömmlichen Digitalschaltungen mit TTL-Schaltkreisen. Die Stromlaufpläne der Netzteile unterscheiden sich deshalb nicht von den bei dieser Technik üblichen; neben modernen Schaltnetzteilen für größere Ströme sind vor allem integrierte Festspannungsregler üblich. Einige Speicherschaltkreise (U 555, U 256) erfordern neben +5 V noch +12 V und -5 V. Die Stromversorgung muß diese Spannungen in festgelegter Reihenfolge zu- und abschalten. Die Spannung -5 V soll als erste einschalten und als letzte abschalten, sonst könnten die Bauelemente zerstört werden. Meist reicht es aus, wenn dazu das Netzteil für -5 V mit einem für den Strombedarf überdimensionierten Ladekondensator versehen wird. Die Kondensatoren der Netzteile für die anderen Spannungen entladen sich nach dem Abschalten schneller. Wer ganz sichergehen will, kann auch mit einer Relaischaltung die Reihenfolge steuern.

Bild 1.20 zeigt ein Schaltnetzteil mit dem universellen Spannungsregler-Schaltkreis MAA 723.

Schaltregler zeichnen sich bei vergleichbarer Ausgangsleistung gegenüber kontinuierlichen Reglern durch geringere Erwärmung aus. Letztere sind jedoch in Entwurf und Aufbau weniger kritisch. Bild 1.21 zeigt eine Schaltung, bei der die Spannungen +5 V und +12 V mit gemeinsamen Gleichrichtern erzeugt werden. Sie belastet den Leistungstransistor beim 5-V-Teil jedoch nicht mit den für den Schaltkreis erforderlichen 12 V, wodurch die Verlustleistung etwas geringer ist als bei der Standardschaltung.

Bild 1.22 zeigt eine Schaltung mit dem Schaltkreis B 3170 (LM 317), bei dem die Ausgangsspannung durch externe Widerstände einstellbar ist. Sie kann die Programmierungsspannung für verschiedene EPROM liefern, ist jedoch mit entsprechend dimensionierten Widerständen auch für die übrigen Netzteile geeignet. Ebenso wie bei anderen elektronischen Geräten sollte die Netzleitung Störschutzglieder aufweisen.

1.5. Mechanischer Aufbau

Der Aufbau eines Mikrocomputers unterscheidet sich kaum von dem üblicher elektronischer Geräte. Ein wichtiger Unterschied

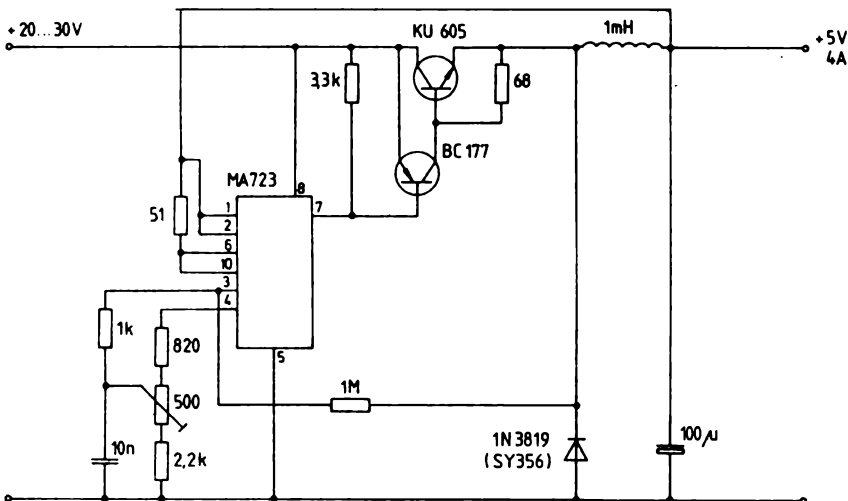
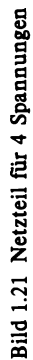


Bild 1.20 Schaltnetzteil 5 V/4 A



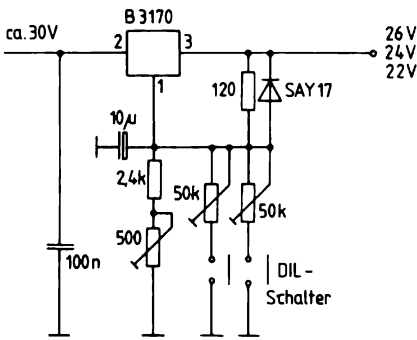


Bild 1.22 Netzteil mit einstellbarer Ausgangsspannung

besteht darin, daß er nicht «fertig» wird, weil er ein ständiges Experimentierfeld bildet, auf- und abrüstbar sein soll und das Grundgerät ständig zur Inbetriebnahme und Erprobung neuer Baugruppen dient. Beim Aufbau muß dem Rechnung getragen werden. Die Rückverdrahtung sollte leicht zugänglich sein. Man muß jederzeit Leiterplatten stecken oder ziehen können. Selbstverständlich müssen die Sicherheitsvorschriften für elektronische Geräte beachtet werden, z. B. Schutz vor Berührung von Hochspannung führenden Teilen usw. Die Leiterplatten sollten senkrecht stehen, so daß die Bauelemente durch die natürliche Belüftung gekühlt werden. Zu den Elektronikleiterplatten sind einige Vorbemerkungen angebracht.

Viele Amateure haben das durchaus verständliche Bestreben, ihren Leiterplatten ein professionelles Aussehen zu geben und die Qualitätsnormen der industriellen Produkte zu erreichen. Jedoch befindet sich der Amateur eher in der Situation des Entwicklers im Labor: Bei der Produktion einer größeren Stückzahl von gleichen Leiterplatten lohnt sich der Aufwand für den Entwurf einer gedruckten Verdrahtung. Bei einzelnen Mustern ist dies nicht der Fall, vor allem wenn auch die Schaltung noch Veränderungen unterworfen ist, sich in Entwicklung befindet. Dann ist der Aufbau auf universellen Lochraster-Leiterplatten zweckmäßiger. Die Bestückung ist zwar zeitaufwendiger als bei einer fertigen, geätzten Leiterplatte, jedoch bringt dies auch einen Vorteil mit sich: Man kann für eine bestimmte Aufgabe Bauele-

mente mit unterschiedlicher Stiftbelegung einsetzen, wichtig ist dann nur die logische Funktion, z. B. kann ein AND-Gatter *D 108* zwei NAND-Gatter *D 100* ersetzen. Die bei höher integrierten Digitalschaltungen üblichen doppelseitigen Leiterplatten mit durchkontaktierten Leiterzügen kann der Amateur nicht selbst herstellen. Als Notbehelf kann man Leiterplatten verwenden, bei denen Leiterzüge zwar auf beiden Seiten vorhanden sind, die Durchkontaktierungen jedoch von Hand mittels Drahtbrücken gelötet werden. Bei dieser Technik kann es passieren, daß für eine Verbindung zwischen Bauelementen an entgegengesetzten Rändern der Leiterplatte mehrere Lötstellen erforderlich sind! Auf Universal-Leiterplatten kann ein Draht mit 2 Lötstellen die gleiche Aufgabe lösen. Optimal für die Konstruktion von Leiterplatten erschienen deshalb für den Amateur einseitige Leiterplatten, die zwar die meisten Leiterzüge schon tragen, jedoch noch einige Drahtbrücken aufweisen.

Bei den Speicherschaltungen liegen z. B. mehrere gleiche Bauelemente mit ihren meisten Anschlüssen parallel am BUS, nur die Chip-Select-Signale kommen von anderen Ausgängen des Dekoders. Dieses Prinzip der Konstruktion von Leiterplatten bringt dem Amateur folgende Vorteile:

- Leiterplatten können selbst gefertigt werden.
- Bei der Bestückung ist man nicht immer streng an bestimmte Bauelemente gebunden.
- Wenn ein Teil der Leiterplatten als Universal-Lochraster ausgeführt ist, kann man problemlos experimentieren.

Ein Nachteil dieses Konzeptes sei nicht verschwiegen: Die Möglichkeit, fehlerhaft zu bestücken, ist größer. Hier hilft nur äußerst exakte Arbeit und gewissenhafte Kontrolle. Bild 1.23 zeigt die Leiterplatte für die Prozessorbaugruppe mit der Schaltung von Bild 1.4. (siehe Anhang 5.5. Leiterplattenlayouts).

Neben der angegebenen Schaltung können auf dem Universalteil beliebige andere Schaltungen aufgebaut werden, z. B. die bereits erwähnten für einen Fernschreiberanschluß oder ein weiterer Speicherschaltkreis.

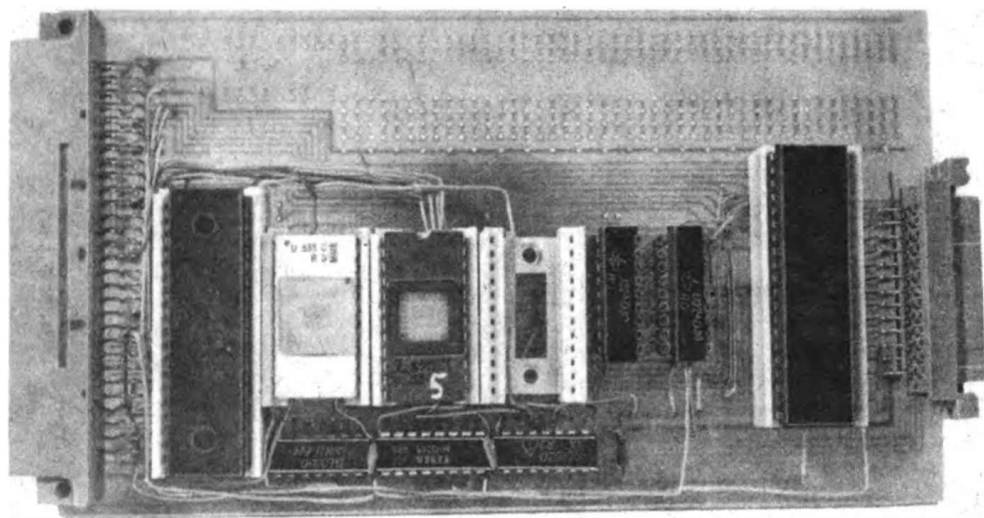


Bild 1.24 Bestückte CPU-Leiterplatte

Da sich die höher integrierten Speicherschaltkreise in ihrer Stiftbelegung nur geringfügig unterscheiden, besteht die Möglichkeit, anstelle der EPROM U 555 Schaltkreise der Typen 2716, 2732, 2764 sowie die pinkompatiblen RAM 6116 u. a. einzusetzen. Aus diesem Grunde soll ein Bestückungsplan auch nicht angegeben werden.

Im Gegensatz zu den im Abschnitt 3.1. vorgestellten Speicherleiterplatten muß hier bei der Bestückung noch viel Handarbeit geleistet werden, wozu lötfähiger Kupferlackdraht von 0,2 bis 0,5 mm Durchmesser oder plastisolierter Schaltdraht geeignet sind.

Bild 1.25 zeigt die Leiterplatte mit der gedruckten Rückverdrahtung der Steckverbinder für das BUS-System.

1.6. Inbetriebnahme

Es wurde bereits betont, daß nur gewissenhafte Arbeit und Kontrolle des Aufbaus vor der Inbetriebnahme zum Ziele führen. Trotzdem sind Fehler bei der Bestückung nicht ausgeschlossen. Im Gegensatz zu anderen elektronischen Geräten arbeitet ein

Mikrocomputer auch bei vielen kleineren Fehlern (z. B. Kurzschluß einer einzigen Datenleitung mit Masse) nicht einmal teilweise. Die Fehlersuche kann deshalb sehr zeitaufwendig sein, und Anfängern ist zu raten, sich von einem erfahrenen Amateur helfen zu lassen.

Neben allgemein üblichen Laborgeräten wie Oszilloscope, Logikpegelanzeige, Spannungsmesser und Durchgangsprüfer können bei der Fehlersuche einige spezielle Hilfsmittel entscheidend zum Erfolg beitragen. Dabei kann der Aufwand erstaunlich gering bleiben. Bild 1.26 zeigt eine Schaltung, die es gestattet, ein Programm auf Tastendruck in Einzelschritten abzuarbeiten. Wenn dabei die Pegel der Adressen und Daten kontrolliert werden, kann man viele Fehler, u. a. den oben als Beispiel genannten, schnell finden. Die Schaltung arbeitet wie folgt:

Ein Low-Pegel am Eingang «WAIT» stoppt die Arbeit des Prozessors, die Pegel der BUS-Leitungen bleiben konstant.

Mit einem Tastendruck erzeugt das Flip-Flop 2 über das nachfolgende Gatter einen Impuls am Rücksetzeingang von Flip-Flop 1. Dessen Ausgang schaltet auf L-Pegel, WAIT erhält H-Pegel, und der Prozessor

Tabelle 1.6. Zuordnung der Tasten

!	"	#	\$	%	&	'	(	)	0
1	2	3	4	5	6	7	8	9	
Q	W	E	R	T	Z	U	I	O	P
A	S	D	F	G	H	J	K	L	SHIFT
CTRL	Y	X	C	V	B	N	M	CR	DEL
ETX									

:		
<	+	↑
-	-	-
/		>

Leertaste (Space)

RESET

fährt in der Bearbeitung des aktuellen Befehls fort. Mit dem Lesen des nächsten Bytes vom Speicher oder bei Zugriff auf eine Ein-/Ausgabeadresse erzeugt er das Signal $\overline{MREQ}$ oder $\overline{IORQ}$, die das Flip-Flop 1 wieder setzen. Damit liegt $\overline{WAIT}$ wieder aktiv an.

Die Kontrolle der Logikpegel mit einem üblichen TTL-Prüfstift ist zwar möglich, wegen der Vielzahl der Leitungen jedoch recht mühsam. Effektiver arbeiten hier Schaltungen, die alle Pegel gleichzeitig anzeigen, z.B. über von TTL-Schaltkreisen mit offenem Kollektor (7406, DL 003) angesteuerten Lichtemitterdioden. Besonders komfortabel ist die Schaltung nach Bild 1.25, die alle wichtigen Pegel in hexadezimaler Schreibweise darstellt. Dieser Aufwand erscheint jedoch für den einzelnen Amateur weniger sinnvoll.

Bei der Inbetriebnahme von Schaltungen kommt es häufig vor, daß nur einige Teile nicht wie gewünscht arbeiten. Wenn der Computer bereits betriebsfähig ist, kann in solchen Fällen ein kurzes Testprogramm sehr nützlich sein. Mit einer Programmschleife, die immer wieder ein bestimmtes Byte auf eine konstante Adresse schreibt, kann man sehr gut oszilloskopisch die Signale verfolgen und Fehler in der Dekodierung der Adresse oder ähnliches finden.

Mit Hilfe des $\overline{NMI}$ (Nicht Maskierbarer Interrupt) kann solch ein Testprogramm auch von Anfang an im Schrittbetrieb ablaufen. In die $\overline{NMI}$ -Sprunganweisung des Monitorprogrammes (Adresse FF1) ist dazu die Startadresse des Testprogrammes einzutragen. Bei Bedarf kann nun auf Einzelschritt geschaltet, danach mit $\overline{RESET}$ und $\overline{NMI}$ eine definierte Adresse im Testprogramm

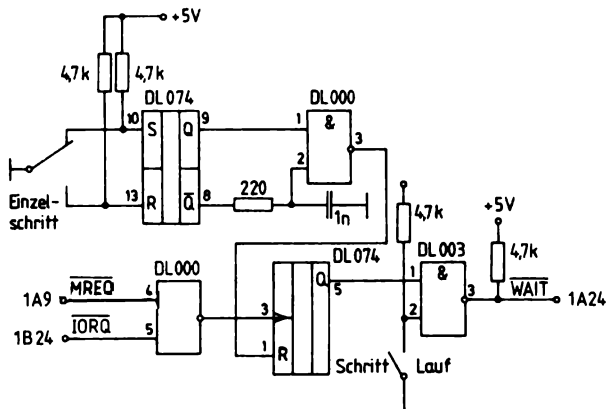


Bild 1.26
Stromlaufplan für die Einzelschrittsteuerung

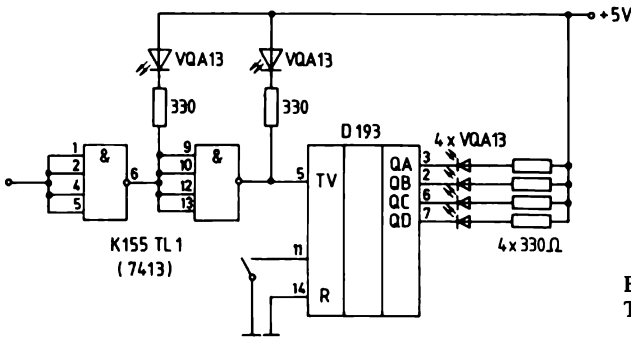


Bild 1.27
TTL-Impulsprüfstift

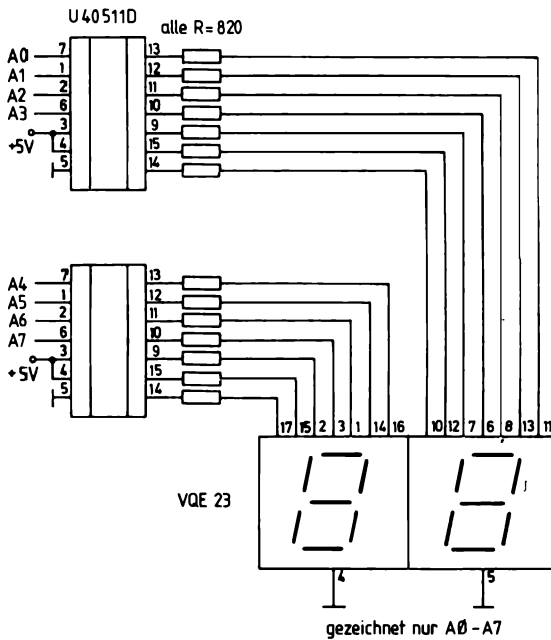


Bild 1.28
Hexadezimale Anzeige für Logik-Pegel

angesprochen und dieses schrittweise abgearbeitet werden. Der Prozessor macht dabei den «Umweg» über die Adresse 66, die er bei Auftreten des $\overline{\text{NMI}}$ zunächst aufruft. Wer die Mühe des Abtippens der Programme sparen möchte, kann bei der Firma J.-D. Baer Ernst-Thälmann-Str. 47 Frankfurt/O. 1200

seine EPROMs programmieren lassen oder entsprechende ROMs beziehen. Auch die im Anhang aufgeführten Leiterplatten können bei dieser Firma bezogen werden.

2. Programmierung

Das Programm eines Computers ist die in seinem Speicher untergebrachte Folge von Anweisungen an den Prozessor. Diese allgemeine Formulierung deutet bereits an, daß dabei unterschiedlichste Formen der Darstellung, der Art und Weise ihrer Erstellung und des Betriebes von Programmen möglich sind. Die Möglichkeiten sind so vielfältig und interessant, daß über kurz oder lang viele Amateure den Lötkolben mit dem Bleistift vertauschen werden. Ist doch das Programmieren als höchste und ideale Form des Bastelns zu betrachten, denn man benötigt dazu nicht ständig neue und teure Bauelemente! Die Grundlagen des Programmierens lassen sich besonders effektiv mit Hilfe der Programmiersprache BASIC erlernen, die speziell für die Ausbildung von Studenten entwickelt wurde und in der Mikrocomputertechnik am stärksten verbreitet ist. Zunächst folgen jedoch noch einige allgemeingültige Regeln der Programmentwicklung. Ebenso wie in der Schaltungstechnik muß vor Beginn der Entwicklung eines Programms die Aufgabenstellung konkret vorliegen und eine Konzeption mit ihren Entscheidungen den Lösungsweg klar aufzeichnen. Der Formulierung von Aufgabenstellungen geht in Wirtschaft und Industrie häufig eine Problemanalyse voraus.

Die konkrete Lösung des Programmierproblems findet ihren Abschluß im Test (Ausprobieren), denn jedes längere Programm enthält zunächst Fehler. Manchmal sind Programme so komplex und unübersichtlich, daß nicht alle Möglichkeiten, z. B. bei Fehlbefehlsbedienung, getestet werden und Fehler sich erst nach Jahren herausstellen.

2.1. Einführung in die Programmiersprache BASIC

2.1.1. BASIC-Sprachelemente

BASIC ist die Abkürzung von «Beginners All Purpose Symbolic Instruction Code», was man etwas frei mit «Universelle Programmiersprache für Anfänger» übersetzen kann. Ihre Sprachelemente bestehen zu einem großen Teil aus Wörtern der englischen Sprache. Sie sind leicht zu verstehen und zu behalten. BASIC hat somit den Vorteil, daß es unabhängig vom eingesetzten Mikroprozessor, von der Größe des verfügbaren Speichers und von der Art der Peripherie ist, wenn Mindestanforderungen erfüllt sind. Im Regelfall muß der Programmierer deshalb keine Elektronikkenntnisse haben. Der Computer arbeitet bei BASIC mit zwei Programmen:

Das eigentliche BASIC-Programm ist eine Folge der oben erwähnten Sprachelemente. Es wird mit Hilfe des BASIC-Interpreters zeilenweise abgearbeitet, indem es in eine Folge von Befehlen in der Maschinensprache des Prozessors umgewandelt wird. Der BASIC-Interpreter ist ein Programm, das sich außer dem BASIC-Programm ebenfalls im Speicher des Computers befinden muß.

Jede Zeile des BASIC-Programms hat eine andere, maximal 5stellige dezimale Zeilennummer. Damit ist sie eindeutig wie mit einer Adresse gekennzeichnet. Die Nummern müssen nicht fortlaufend belegt sein, im Gegenteil, bei der Programmerstellung läßt man absichtlich Lücken, um später beim Test noch etwas einfügen zu können. Nach der Zeilennummer folgt ein Schlüsselwort, das den eigentlichen Befehl darstellt. Danach können noch Glieder folgen, mit denen eine Operation ausgeführt werden soll,

2. Programmierung

z. B. eine mathematische Funktion und deren Argumente oder Gleichungen. Leider hat sich, bedingt durch die stürmische Entwicklung der Computertechnik, eine Vielzahl von BASIC-«Dialekten» herausgebildet, so daß man doch eine geringe Computerabhängigkeit vorfindet. Die größte Variation bei den verschiedenen Computern bzw. BASIC-Interpretern liegt in der Anzahl der jeweils zulässigen Befehle. In der umfangreichen Literatur (z. B. [2], [4], [10]) sind Arbeitsweise und Anwendung von verschiedenen BASIC-Interpretern ausführlich beschrieben.

Tabelle 2.1. enthält die Befehle des im Anhang, Abschnitt 5.2.5. aufgelisteten Interpreters. Er entstammt einem CP/M-Gerät und wurde mit Hilfe mehrerer Amateure an die in Abschnitt 1. beschriebenen Schaltungen und unter 2.3. aufgeführten Programme angepaßt. Zum Betrieb des Interpreters ist im Grundgerät nur das Programm IN/OUT-Monitor (1 kByte) erforderlich.

Tabelle 2.1. BASIC-Schlüsselwörter

Systembefehle	
AUTO	Automatische Bildung von Zeilen-Nummern
CLEAR	Variablen auf 0 setzen
CONTINUE	Fortsetzung des Programmablaufes nach BREAK
COPY	Kopieren von Programmzeilen
DELETE	Löschen von Programmzeilen
EDIT	Programmzeilen ändern
LIST, LLIST	Ausgabe von Programmzeilen
ALOAD	Laden von Programmen vom externen Speicher
ASAVE	Laden von Programmen auf externe Speicher
AMERGE	Laden von mehr als einem Programm («Mischen»)
RUN	Start eines Programms
NEW	Löschen des BASIC-Programms
RENUMBER	Zeilen neu numerieren
TRACE, LTRACE	Testlauf mit Ausgabe der Zeilennummer
PRECISION	Abrunden auf weniger als 11 Stellen

Programmbefehle	
CALL	Aufruf von Assemblerunterprogrammen
DATA	Daten (Konstanten)
DIM	Feldvereinbarungen
END	Programmende
FOR...STEP...NEXT	Schleifenanweisung
GOSUB	Unterprogrammaufruf
GOTO	Sprungbefehl
IF...THEN...ELSE	Bedingter Sprung
INP	Eingabe von einem E/A-Port
INPUT	Eingabe von Daten über ein Gerät
LET	Arithmetische Zuweisung
ON...GOTO	Arithmetische Sprunganweisung
ON...GOSUB	Arithmetischer Unterprogrammaufruf
OUT	Ausgabe auf ein E/A-Port
PRINT, LPRINT, ?	Ausgabebefehle
PRINT USING	Formatierte Ausgabe
READ, RESTORE	Zuweisung von Konstanten
REM, '	Kommentar-Kennzeichnung
RETURN	Rücksprung vom Unterprogramm
STOP	Unterbrechung des Programmablaufes
SPC	Leerzeichenausgabe
TAB	Ausgabe in Tabellenform
USR	Aufruf von Maschinenprogrammen mit Parameter
WIDTH, LWIDTH	Zeilenlänge der Ausgabe festlegen

Mathematische Operationen

Potenz (^)	Division (/)
Multiplikation (×)	Addition (+)
Subtraktion (-)	

Vergleichsoperationen

gleich (=)	ungleich (> < oder <>)
größer als (>)	größer gleich (> = oder >=)
kleiner als (<)	kleiner gleich (< = oder <=)

Logische Operationen

AND	OR	XOR
-----	----	-----

Funktionen

ABS	Absoluter Wert
ATN	Arcus Tangens
COS	Cosinus
EXP	e hoch x
INT	Ganzzahliger Teil
SGN	Vorzeichen
SQR	Quadratwurzel
SIN	Sinus
DEF FN	Anwenderfunktion definieren
TAN	Tangens
LOG	Natürlicher Logarithmus
LVAR	Ausgabe von Variablen
RND	Pseudo-Zufallszahlengenerator
RANDOMIZE	Startpunkt für RND neu festlegen

Stringoperationen

ASC	Dezimalwert eines ASCII-Zeichens
CHR \$	ASCII-Zeichen eines Dezimalwertes
LEFT \$	Linker Teil einer Zeichenkette
LEN	Länge einer Zeichenkette
MID \$	Teil aus einer Zeichenkette
RIGHT \$	Rechter Teil aus einer Zeichenkette
STR \$	Zahlenwert-Zuweisung für Zeichenkette
INSTR	Suche einer Zeichenkette

Abschnitt 2.3. erläutert die Form der Darstellung und die Eingabe in den Computer. Startadresse des Programms ist 8 000 H.

Die Zeichen der BASIC-Schlüsselwörter und ihrer Operanden können große Buchstaben des lateinischen Alphabets, mathematische Zeichen, Satzzeichen oder arabische Ziffern sein. Als Konstanten im mathematischen Sinne sind ganze und dezimalgebrochene Zahlen, sogenannte Gleitkommazahlen, zugelassen. Sie werden anstelle des Kommas mit einem Punkt geschrieben, z.B.:
 $.3 - 3.145926 \quad .16E-20$

Im letzten Beispiel steht das E für «Exponent zur Basis 10», die Zahl würde also in üblicher Schreibweise $0,16 \cdot 10^{-20}$ lauten.

Variable Zahlen bezeichnet man mit

einem Buchstaben oder mit einem Buchstaben gefolgt von einer Ziffer. Bei der Bezeichnung von Variablen sind zwar bis zu 7 Zeichen zulässig, der Interpreter bewertet jedoch nur die ersten beiden, zulässig sind z.B. M1 und M2. Bezeichnet man aber eine Variable mit MESS1 und eine weitere mit MESS2, so werden beide nicht unterschieden, d. h., eine Wertzuweisung auf MESS1 gilt auch für MESS2. Variable können einen Index besitzen, den man in Klammern an den Namen der Variablen fügt. Beispiel: Meßwert 13 der Meßreihe 5 kann lauten:

$$M5(13) = -7.$$

Der Index kann auch eine Variable sein oder durch Berechnungen eines mathematischen Ausdrucks entstehen. Für die Berechnung von Ausdrücken sind neben den Grundrechenarten auch die Winkelfunktionen, die Potenz- einschließlich der Logarithmenrechnung und logische Funktionen zugelassen. Tabelle 2.1. zeigt auch die mathematischen Operationen und Funktionen sowie die Symbole für deren Aufruf. Diese leistungsfähige und vergleichsweise einfach zu handhabende Arithmetik trug wesentlich zur Verbreitung der Programmiersprache BASIC bei. Häufig kommen in Programmen Texte vor, die entweder dem Dialog mit dem Programmierer dienen oder Arbeitsergebnisse des Programms sind. Sie können dazu den sogenannten Stringvariablen zugeordnet werden. Diese unterscheiden sich von den zuvor erwähnten durch ein angehängtes \$-Zeichen, z.B.:

$$A\$ = \text{«JA»}.$$

Mit diesen Variablen kann man zwar keine mathematischen Operationen durchführen, logische Funktionen wie Vergleich mit einer anderen Zeichenkette oder andere, weiter hinten beschriebene Operationen eröffnen jedoch interessante Möglichkeiten. Aufgabe und Sinn der meisten BASIC-Befehle geht bereits aus ihrer Bezeichnung in Tabelle 2.1. hervor. Die Systembefehle steuern die Bedienung und den Dialog mit dem Computer. Sie werden ohne Zeilennummer eingegeben und sofort ausgeführt. Auch viele weitere Befehle kann man im Direktmodus wie bei einem Taschenrechner ohne Zeilennummer sofort ausführen lassen, jedoch ist diese Be-

2. Programmierung

triebsart nur bei kleinen Problemen zweckmäßig. Nachstehend folgt die Erläuterung einiger BASIC-Befehle an Programmbeispielen. Mit dem Befehl PRINT können Texte und Ergebnisse von Berechnungen auf den Bildschirm ausgegeben werden. Mit dem Befehl LPRINT arbeitet der Drucker. Texte stehen in Anführungszeichen, der Befehl PRINT «HALLO» (1)

ergibt im Direktmodus die Ausschrift HALLO auf dem Bildschirm. In der Befehlszeile bildet die Zeichenkette HALLO den sogenannten Operanden des Befehls PRINT. Bei PRINT und den meisten anderen Befehlen sind als Operanden noch Konstanten, Variable sowie Funktionen und arithmetische Ausdrücke zugelassen.

Als Kurzform für PRINT ist auch das Fragezeichen zugelassen. Mehrere Operanden können in einer Zeile ausgegeben werden, wobei sie in der Anweisung durch Komma oder Semikolon getrennt stehen. Das Komma bewirkt die Ausgabe mit automatischer Tabellierung nach je 15 Druckpositionen, mit der Funktion TAB kann man die Druckposition wählen. Die Zeilen
N = 77 (2)

PRINT «DURCHSCHNITTliches ALTER:» ,N TAB(50); «Jahre» bewirken die Ausschrift des berechneten Wertes für N auf der nächsten durch 15 teilbaren Druckposition nach dem Text; das Wort «Jahre» erscheint ab Spalte 50. Ein Semikolon veranlaßt die Zeichenausgabe auf der nächsten freien Druckposition. Häufig soll die Ausgabe von Rechenwerten bestimmten Formen genügen, z. B. mit 2 Stellen vor und nach dem Dezimalpunkt. Dies ist mit der PRINT-Anweisung möglich, wenn man für die Festlegung der Ausgabe eine sogenannte Maske benutzt:

```
5  A$ = ##.## (3)
50  V = 5
75  PRINT USING A$; V
100 PRINT USING 120; SQR(V)
120 !##.##
130 PRINT SPC (2);V
```

Im Beispiel soll V der berechnete Wert einer Variablen sein. Argumente von Funktionen (Quadratwurzel, Zeile 100) stehen in Klammern.

Die Ausgabe einer Anzahl Leerzeichen kann man mit dem Befehl SPC programmieren (Zeile 130). Vor der Ausgabe der Variablen V fügt der Computer noch 2 Leerzeichen (Space) ein. Mit den Befehlen WIDTH n bzw. LWIDTH n kann man die Anzahl n der Zeichen pro Zeile bei der Ausgabe wählen. Der Operand n darf zwischen 15 und 255 liegen. Damit ist es möglich, auf den Bildschirm mit 63 und auf den Fernschreiber mit 75 Zeichen je Zeile auszugeben.

Enorme Bedeutung haben Programmschleifen zur mehrfachen Wiederholung eines Befehls durch den Computer. Eine einfache Zeitschleife, bei der nur ein Wert gezählt wird, kann man mit der Schleifenanweisung programmieren. Dazu ist eine Variable erforderlich, deren Anfangswert die Länge der Zeit bestimmt:

```
10  FOR V = 1 TO 10 000 (4)
200 NEXT V
```

Nach dem Start mit RUN dauert es eine Weile, bis der BASIC-Interpreter sich wieder meldet. Häufiger stehen zwischen dem Beginn der Schleife (FOR....) und dem Rücksprung (NEXT....) andere Anweisungen. So können z. B. in das Programm (4) die Zeilen 100 und 120 des Beispiels (3) eingefügt werden. Ohne weitere Festlegung erhöht sich die Variable der Schleife um den Betrag 1 bei jedem Durchlauf. Mit der Anweisung STEP kann man eine andere Schrittweite festlegen:

```
10  FOR V = 1 TO 10 STEP 0.3. (5)
Sollen Schleifen ineinander geschachtelt auftreten, so muß man die Schleifenvariablen unterschiedlich bezeichnen:
10  FOR V1 = 1 TO 10 (6)
20  FOR V2 = -7 TO 7
30  PRINT TAB(V2*V2); «*»
40  NEXT V2
50  NEXT V1
```

Der Computer führt die Befehle entsprechend den aufsteigenden Zeilennummern aus, bis ein Sprungbefehl auf eine Zeile außerhalb dieser Reihe weist. Diese Sprungbefehle erlauben es, dem Programm die erforderliche Struktur zu geben.

Neben dem Befehl GOTO 1234, der immer die weitere Abarbeitung des Programms ab der Zeilennummer 1234 zur Folge hat,

besitzen besonders Sprungbefehle große Bedeutung, deren Ausführung von einer Bedingung abhängig ist.

```
100 IF V = 0 THEN 10 ELSE 400 (7)
```

Wenn die Variable V den Wert 0 erreicht hat, läuft das Programm wieder ab Zeile 10, anderenfalls ab Zeile 400. Mit dieser Anweisung kann man Schleifen aufbauen, deren Länge von programmierbaren Bedingungen abhängt. Mit der arithmetischen Sprunganweisung.

```
ON V GOTO 1234, 5678, 9012 (8)
```

kann das Programm entsprechend dem ganzzahligen Teil der berechneten oder eingegebenen Variablen V auf mehrere Sprungziele (im Beispiel 3) verzweigen. Liegt der Zahlenwert von V unter 1 oder über 3 (im Beispiel 8), so läuft das Programm mit der folgenden Anweisung weiter.

Soll eine Folge von Befehlen an mehreren Stellen des Programms laufen, so kann man sie zu einem Unterprogramm zusammenfassen, das vom Hauptprogramm mehrfach aufgerufen wird. Beim Auftreten des Befehls

```
100 GOSUB 1234
```

springt der Computer auf Zeilennummer 1234, speichert aber die aktuelle Zeilennummer. Nach Abarbeitung des Unterprogramms setzt er das Programm mit der auf Zeile 100 folgenden fort. Das Ende des Unterprogramms ist durch das Schlüsselwort RETURN gekennzeichnet. In der Praxis folgen die Unterprogramme mit ihren Zeilennummern meist dem Hauptprogramm. Mit dem Befehl END kann man verhindern, daß der Computer nach der Abarbeitung des Hauptprogramms die Zeilennummer eines dahinter stehenden Unterprogramms erreicht. Er beendet den Programmablauf. Ebenso wie beim Sprung kann ein Unterprogrammaufruf auch von einer Bedingung abhängen. Der arithmetische Unterprogrammaufruf

```
ON V GOSUB 1234, 5678,.....
```

entspricht sinngemäß dem arithmetischen Sprung.

Mit dem Schlüsselwort REM kann man Bemerkungen und Erläuterungen in den Programmtext einfügen. Das erleichtert dem Programmierer oder einem anderen das Verständnis des Programms.

Ein Apostroph kündigt die Bemerkung unmittelbar nach einer Anweisung in der gleichen Zeile an.

Oft wird eine Folge von Konstanten für die Berechnung von Werten benötigt (z. B. π , e). Im Programm sind diese durch das Schlüsselwort DATA gekennzeichnet, dem die Konstanten jeweils durch ein Komma getrennt folgen. Mit dem Befehl READ kann man sie Variablen zuordnen. Beispiel:

```
10 DATA 3.14, 2.7, 1 (9)
```

```
100 READ P, E, E1
```

Nach der Abarbeitung enthält die Variable P den Wert 3.14, die Variablen E und E1 arbeiten mit den Zahlen 2.7 und 1 weiter. Nach dem Befehl RESTORE erfolgt die Zuordnung des nächsten zu lesenden DATA-Wertes wieder von vorn, d. h., im Beispiel (9) kann einer anderen Variablen ebenfalls der Wert 3.14 zugewiesen werden.

Die Anweisung DIM V(10) reserviert einen Teil des Speichers für Zahlenwerte und ordnet ihn der indizierten Variablen V(10) zu. Mit dem Befehl INPUT V erhält die Variable V ihren Wert über ein Eingabegerät (Tastatur, Kassettenrecorder). Die Befehle CALL, PEEK, POKE, INP und OUT «greifen» auf die Computerschaltung zu. Sie sind damit nicht geräteunabhängig wie die anderen Befehle. Mit CALL 822 erfolgt der Aufruf eines Unterprogramms, das nicht in der Programmiersprache BASIC geschrieben ist, sondern sich als Maschinencode im Speicher befindet (Piepton des Monitors). Die Adresse 822 kann dezimal angegeben werden. Soll eine Adresse hexadezimal verarbeitet werden, muß ein &-Zeichen sie kennzeichnen ($822 \text{ dez} \cong 336 \text{ Hex} \cong \&336$). Das Unterprogramm endet mit einem Rücksprungbefehl in Maschinencode. Beim Aufruf von Unterprogrammen in Maschinensprache kann es wünschenswert sein, Parameter (ganze, positive Zahlen bis 65535) an diese zu übergeben, um z. B. die Länge einer Ausgabe festzulegen. Im BASIC-Programm folgen sie der CALL-Anweisung jeweils durch ein Komma getrennt. Nach Ausführung des Befehls steht die Anzahl der Parameter im Registerpaar BC des Prozessors, das Registerpaar HL zeigt mit seinem Inhalt auf die Rückkehradresse, und der Stack

2. Programmierung

(siehe Abschnitt 2.2.) enthält die Parameter. Vor Anwendung dieser Möglichkeit des Befehls sollte man sich die Wirkungsweise mit einem CALL 56,1,2,... ansehen (siehe Abschnitt 2.4.1.). Der Befehl USR verfolgt ein ähnliches Ziel wie CALL (Aufruf eines Unterprogramms in Maschinensprache). Allerdings übergibt der BASIC-Interpreter nicht nur einen Parameter, er erwartet auch einen solchen zurück, den er in die interne Zahlendarstellung umwandeln und einer Variablen zuordnen kann. Sinnvoll läßt sich dies bei einem A/D-Wandler anwenden, dem man die Nummer der Meßstelle übergibt und den zugehörigen Meßwert übernimmt. Obwohl die Programmierung in Maschinensprache erst im Abschnitt 2.2. folgt, soll der Systematik und Vollständigkeit wegen doch hier ein Beispiel die dargestellte Problematik verdeutlichen.

Dem Leser ist zu empfehlen, das Beispiel später nochmals durchzuarbeiten. Das BASIC-Programm (10) übergibt an das einfache Unterprogramm in Maschinensprache (11) eine Zahl, dieses gibt sie um 1 erhöht an den Interpreter zurück.

```
10 INPUT N (10)
20 A = USR(N)
30 PRINT A
```

Das Unterprogramm (11) erhält den Wert für N im Registerpaar DE und übergibt sein Ergebnis in den Registern A (hochwertiger) und B (niederwertiger Teil). Die beiden CALL-Befehle besorgen die nötige Umwandlung (Konvertierung) der Zahlendarstellung. In die Systemtabelle des Monitors ist auf den Adressen FFAH und FFBH die Anfangsadresse des Anwenderprogramms (im Beispiel 0E00) einzutragen.

```
0E00 CD 27 80 CALL GKBIN (11)
0E03 7A LD A,D
0E04 43 LD B,E
0E05 04 INC B
0E06 CD 2A 80 CALL BINGK
0E09 C9 RET
```

Die Befehle V = INP(123) bzw. OUT(123), V ordnen der Variablen V ein Datenbyte als Zahl zu, das sie von der IN/OUT-Adresse 123 erhalten oder an diese ausgeben. Die Adresse kann auch als ganzzahliger Teil des Ergebnisses einer Berechnung entstehen.

Ein Beispiel diene wieder zur näheren Erläuterung:

```
10 A = 1 (12)
20 OUT 252,A
30 B = INP(253)
40 IF B > 254 THEN 30
50 PRINT B
```

Der Befehl B = INP(253) liest ein Datenbyte von der Adresse 253 (dezimal) und ordnet es der Variablen B zu. Adresse 253 gehört zum Kanal B des Schaltkreises PIO im Grundgerät und dient als Eingabekanal für die Tastatur. In Verbindung mit dem Befehl OUT 252, A, der an Kanal A des PIO den Wert der Variablen A ausgibt, könnte man die Tastaturabfrage auch in BASIC programmieren.

Die arithmetische Zuweisung LET hat als Schlüsselwort keine praktische Bedeutung mehr, man kann sie bei der Eingabe von Gleichungen auch fortlassen.

Der Befehl LVAR bzw. LLVAR veranlaßt die Ausgabe aller im Programmlauf mit Werten belegten Variablen.

Neben den bekannten Funktionen wie Sinus, Cosinus usw. besteht die Möglichkeit, mit dem Befehl DEF FN... eigene Funktionen zu definieren, die dann mit dem festgelegten Namen aufgerufen werden können. Das Beispiel (13) zeigt zwei Möglichkeiten. In Zeile 30 steht die Definition der Funktion PYTAGO, die mit zwei Parametern arbeitet. Mit Zeile 40 erfolgt der Aufruf. Man beachte, daß die Parameter andere Variablen sein können. In den Zeilen 80 bis 140 steht die Definition einer rekursiven Anwenderfunktion zur Berechnung der Zahl π . Je größer die Anzahl der Schleifen ist, desto genauer erscheint das Ergebnis. Die Berechnung einer rekursiven Anwenderfunktion kann man beim Erreichen einer festzulegenden Bedingung mit dem Befehl FNEND abbrechen.

```
10 INPUT «KATHETE 1»;X (13)
20 INPUT «KATHETE 2»;Y
30 DEF FN PYTAGO(A,B)
   = SQR(A2 + B2)
40 C = FN PYTAGO(X,Y)
50 PRINT «HYPOTENUSE = »;C
60 REM
70 PRINT:PRINT:PRINT
```

```

80 DEF FN (N)
90 P1 = 0
100 FOR I = 1 TO N
110 P1 = P1 + 1/(I↑2)
120 NEXT I
130 P2 = SQR(P1*.6)
140 FNEND P2
150 INPUT «SCHLEIFENANZAHL» N
160 A = FNP(N)
170 PRINT «PI =»; A

```

Zur Zeile 110 ist noch eine Erklärung notwendig:

Diese Zeile stellt keine Gleichung dar, sondern eine Anweisung. (Als Gleichung wäre sie falsch!) Mit ihr erhält man den neuen Wert für P1 aus dem alten P1 plus 1 geteilt durch I hoch 2.

Besonders attraktiv sind die Stringbefehle, mit denen man Zeichenkettenvariable bearbeiten kann. Die folgenden Beispiele ergeben zusammen ein Programm, mit dem im Amateurfunkwettkampf (Contest) die Rufzeichen der Gegenstellen registriert, mit den schon gearbeiteten verglichen und zum Abschluß geordnet nach Bezirken der DDR ausgegeben werden.

Im Beispiel (14) enthalten die Zeilen 50 und 60 die Eingabe des Rufzeichens und seinen Vergleich mit einer Zeichenkette als Abbruchkriterium für die Eingabeschleife. Der Befehl LEN in Zeile 70 ermittelt die Länge einer Zeichenkette. Hier dient er zur Prüfung auf Eingabefehler bei den Rufzeichen. In den Zeilen 80 bis 100 erfolgt der Vergleich des neu eingegebenen Rufzeichens mit den bereits gespeicherten indizierten Stringvariablen. (Außer dem Operator «=» ist bei Stringvariablen noch «><» und «+» zulässig). Wenn das neue CALL mit einem bereits erfaßten übereinstimmt, wird es nicht gespeichert (Zeilen 170–190).

```

20 DIM A$(4400)
30 N = 0
40 N = N + 1
50 INPUT «CALL», A$
60 IF A$ = «ENDE» THEN 200
70 IF LEN (A$) ≥ 5 THEN 50
80 FOR I = 1 TO N - 1
90 IF A$ = A$ (I) THEN 170
100 NEXT I
110 A$ (N) = A$

```

```

120 INPUT «ZEIT»; Z
130 INPUT «RAPPORT»; R
140 IF N = 1 THEN LPRINT «CALL»,
    «UTC», «RST»
150 LPRINT A$, Z, R
160 GOTO 40
170 PRINT «DUPLIKAT»
180 N = N - 2
190 GOTO 40

```

Im Beispiel (15) erfolgt mit dem Befehl RIGHT\$ von rechts beginnend die Abspaltung eines Zeichens aus der Stringvariablen, was im Falle der Amateurfunkrufzeichen der DDR den Bezirk kennzeichnet. (In gleicher Weise arbeitet der Befehl LEFT\$, nur zählt er die Zeichen von links.) Beide Befehle erlauben es auch, mehrere Zeichen abzutrennen. In Zeile 270 des Beispiels wird der Buchstabe des Bezirkskenners mit dem Befehl ASC in die entsprechende Dezimalzahl umgewandelt, wobei man ASCII-Kodierung voraussetzt (d.h. A = 41H = 65 dez). Die Variable V entsteht dann als Zahl von 1 bis 15. Will man den Befehl ASC auf eine beliebige Stringvariable anwenden, wandelt er das erste Zeichen um.

```

200 N = N - 1
210 LPRINT «DUPLIKAT-
    KONTROLLE»
220 FOR K = 1 TO 15
230 LPRINT
240 READ B$
250 LPRINT «BEZIRK»; B$
260 FOR I = 1 TO N
270 V = ASC(RIGHT$(A$(I), 1)) - 64
280 IF V = K THEN LPRINT A$(I) « »
290 NEXT I
300 LPRINT
310 NEXT K
320 LPRINT:LPRINT«ENDE»
330 RESTORE
340 DATA «ROSTOCK», «SCHWERIN»,
    «NEUBRANDENBURG»
350 DATA «POTSDAM», «FRANK-
    FURT», «COTTBUS», «MAGDE-
    BURG», «HALLE»
360 DATA «ERFURT», «GERA»,
    «SUHL», «DRESDEN», «LEIPZIG»
370 DATA «KARL-MARX-STADT»,
    «BERLIN»

```

Das Gegenstück zu ASC ist der Befehl

CHR\$, mit dem man aus einer Dezimalzahl das ASCII-Zeichen bilden kann. Das Beispiel (16) gibt den Inhalt des Zeichengenerators auf den Bildschirm. Durch das Zeichen «Wagenrücklauf», das die Bildschirmsteuerung auswertet, werden allerdings einige quasi-Grafikzeichen überschrieben.

```
10 WIDTH 63 (16)
20 FOR I = 1 TO 127
30 Z = CHR$(I)
40 PRINT Z
50 NEXT I
```

Der Befehl MID\$ trennt Zeichen «mitten aus einem String». Es müssen als Parameter die Bezeichnung des Strings, die Startposition von links sowie die Anzahl der Zeichen angegeben werden.

Wie schon bemerkt, sind mit Stringvariablen mathematische Operationen nicht möglich. Soll die berechnete Zahl einer Zeichenkettenvariablen zugewiesen werden, so kann das mit dem Befehl STR\$ geschehen.

```
A = STR$(3.14) (17)
```

Mit diesem Befehl korrespondiert die Anweisung VAL, die aus einer Zeichenkettenvariablen einen Zahlenwert herauslöst. Der Wert muß am Beginn des Strings stehen.

```
10 A$ = «45E» (18)
20 A = VAL(A$)
30 PRINT A
40 PRINT VAL(A$ + «3»)
```

In Zeile 40 bildet der Computer die Exponentialdarstellung der Zahl.

Mit der Anweisung INSTR erhält man eine Zahl, die von links gezählt den Beginn der Übereinstimmung zweier Zeichenkettenvariablen kennzeichnet. Voraussetzung ist, daß die 2. Stringvariable in der ersten enthalten ist, sonst entsteht der Wert 0.

```
100 A = INSTR(«12ABC34», «C3») (19)
110 PRINT A
```

Dieser Befehl erscheint in anderen BASIC-Interpretern auch mit der Bezeichnung POS.

Der Befehl LINEINPUT entspricht dem INPUT bei numerischen Variablen. Er erlaubt es, eine komplette Zeile bis zum Zeichen «Wagenrücklauf» einer Stringvariablen zuzuweisen.

```
10 LINEINPUT B$ (20)
20 PRINT B$
```

2.1.2. Bedienung des BASIC-Interpreters

Mit dem Kommando B des Monitorprogramms erfolgt der Sprung auf Adresse 8000H und damit der Start des Interpreters. Zunächst meldet er sich mit der Frage nach dem Ende des Schreib-/Lesespeichers und erwartet eine Eingabe. Danach kann man ein Programm eingeben oder einzelne Zeilen ohne Zeilennummer sofort wie bei einem Taschenrechner abarbeiten lassen. Neben den im letzten Abschnitt vorgestellten Programmbefehlen enthält Tabelle 2.1. noch die Systembefehle, die für Erstellung und Betrieb von BASIC-Programmen gedacht sind. Nach der Eingabe von Programmzeichen kann man mit dem Befehl LIST Zeilen auf den Bildschirm bringen, um sie zu kontrollieren, oder mit LLIST auf einem Drucker dokumentieren. Mit LIST 1-9 erscheinen nur die mit diesen Nummern bezeichneten Zeilen. Der Befehl DELETE 1234-5678 bewirkt die Löschung der Zeilen 1234 bis 5678 im Programmspeicher. Soll eine fehlerhafte Zeile durch eine andere ersetzt werden, kann man sie durch Eingabe einer neuen Zeile mit der gleichen Zeilennummer überschreiben.

Der Interpreter enthält einige Befehle zur komfortableren Programmerstellung, die vorwiegend im Direktmodus arbeiten. So können mit dem Befehl AUTO die nötigen Zeilennummern «automatisch» erzeugt werden. Ohne Parameter entstehen Zeilen im Zehnerabstand, beginnend mit Zeile 10. Steht bereits etwas im Speicher, so kann man eine Startzeile und den gewünschten Zeilenabstand festlegen:

```
AUTO 220,10 (21)
```

Bei der Angabe nur eines Parameters bewertet der Interpreter dies als Startzeile. Mit der Taste ETX (CONTROL C) kann man diese Betriebsart und auch den Programmablauf abbrechen. Der Befehl CONTINUE bewirkt die Fortsetzung des Programmlaufs. Der Befehl COPY erlaubt es, eine Anzahl von Programmzeilen auf andere Zeilennummern zu kopieren, sie müssen dann nicht nochmals eingetippt werden:

```
COPY 1000,10 = 50-100 (22)
```

Im Beispiel (22) haben die Operanden folgende Bedeutung:

- 1000 – Beginn der neuen Zeilen
- 10 – Zeilenabstand
- 50–100 – zu kopierende Zeilen

Mit dem Befehl EDIT 1234 kann man die Zeile 1234 verändern. Bei längeren Zeilen mit kompliziertem Inhalt (z. B. Formeln) bringt das gegenüber dem Neuschreiben Vorteile. Nach dem Aufruf des EDIT-Befehls gelangt zunächst die aktuelle Zeilennummer in die Anzeige. Mit weiteren Kommandos, die aus n Zahlen ($1 < n \leq 255$) sowie Buchstaben und Steuerzeichen bestehen, ist nun eine Bearbeitung des Textes der Zeile möglich. Gibt man keine Zahl im Kommando an, so bewertet der Interpreter dies als «1». Tabelle 2.2. zeigt die Kommandos und ihre Wirkung.

Tabelle 2.2. Steuerzeichen EDIT-Befehl

n D	n Zeichen löschen
E	Ende des Editierens mit Veränderung
nFx	Stellt Cursor auf das n-te Zeichen x
H	Löscht alle Zeichen rechts vom Cursor und schaltet auf Eingabe
I	Fügt die folgenden Zeichen ein, bis CR oder ESC
nKx	Löscht alle Zeichen vom Cursor bis zum n-ten Zeichen x
L	Ausgabe der Zeile
Q	Ende des Editierens ohne Veränderung
nR	Ersetzt die folgenden n Zeichen durch n vorhandene
X	Setzt Cursor ans Zeilenende und schaltet auf Eingabe
SPACE	Cursor nach rechts
BS	Cursor nach links

Auch der Befehl RENUMBER ist vor allem bei der Programmerstellung wertvoll. Ohne Argument beginnt er die neuen Zeilennummern ab 10 mit einer Schrittweite von ebenfalls 10.

- a) RENUMBER 1111,111,11 (23)
- b) RENUMBER 1111,111

Im Beispiel (23) bedeuten die Argumente:

- 1111 – neue Startzeile,
- 111 – Zeilenabstand, beginnend bei neuer Startzeile

- 11 – 1. Zeile des neu zu nummerierenden Programmteils, darunterliegende Zeilennummern bleiben unverändert.

Dabei muß die neue Startzeile oberhalb der alten liegen, d. h., man kann Lücken innerhalb der Zeilennummern eines Programms erzeugen. Das ist sinnvoll, wenn verschiedene BASIC-Programme zusammen geladen werden sollen.

Mit dem Schlüsselwort RUN beginnt der Ablauf des BASIC-Programms bei der niedrigsten Zeilennummer. Werden beim Lauf des Programms Werte für Variable berechnet, die man bei einem weiteren «RUN» benötigt, startet man das Programm mit der ohne Zeilennummer einzugebenden Anweisung GOTO 1234 auf der Zeile 1234. Auf diese Weise können auch mehrere, durch den Befehl STOP abgetrennte BASIC-Programme mit den gleichen Variablen arbeiten.

Beim Testlauf von Programmen treten oft Fehler auf, die wegen der relativ schnellen Abarbeitung schwierig zu finden sind. Bei der Anweisung TRACE 1 bzw. LTRACE 1 schaltet der Interpreter zwischen die Abarbeitung der Zeilen die Ausgabe der Zeilennummern auf den Bildschirm oder auf den Drucker, so daß man den Ablauf verfolgen kann. Mit TRACE 0 kann man diese Betriebsart wieder ausschalten.

Nach einem Programmabsturz kann der Interpreter auf Adresse 8003H mit RESET wieder gestartet werden, ohne das BASIC-Programm im Speicher zu löschen.

Die Befehle ALOAD und ASAVE erlauben das Einlesen oder Abspeichern umfangreicherer BASIC-Programme auf externen Datenträgern, wie z. B. Lochstreifen oder Kassettenmagnetband im ASCII-Code. Während der Computer beim Laden eines Programms mit ALOAD ein möglicherweise schon vorhandenes löscht, können mit dem Befehl AMERGE mehrere Programme «gemischt» werden. Sofern sich dabei Zeilennummern gleichen, überschreibt sie das zuletzt gelesene Programm. Für die Befehle ALOAD und AMERGE ist mit CALL 892 der Kassettenrecorder als Eingabegerät zuzuweisen.

2. Programmierung

```
10 PRINT "EWIGER KALENDER"
20 FOR S=1 TO 14
30 PRINT
40 NEXT S
50 PRINT " TAG :";
60 INPUT T
70 IF T < 1 THEN 770
80 IF T > 31 THEN 770
90 PRINT " MONAT :";
100 INPUT M
110 IF M < 1 THEN 770
120 IF M > 12 THEN 770
130 PRINT " JAHR :";
140 INPUT J
150 O=M
160 X=0
170 J=J-100
180 X=X+1
190 IF J>100 THEN 170
200 IF M<= 2 THEN 360
210 P=J
220 Y= INT ((O+1)*26/10)
230 V= INT ( X/4)
240 Z= INT (P/4)
250 G= T+P+Y+Z+V-2*X
260 G=G-7
270 IF G>7 THEN 260
280 B= INT (G)
290 PRINT " SIE SIND AM ";
300 ON B GOSUB 510,390,410,430,450,470,490
310 PRINT ",DEM ";T;". ";
320 ON M GOSUB 530,550,570,590,610,630,650,670,690,710,730,750
330 J=J+X*100
340 PRINT ;J;" GEBOREN !";
350 STOP
360 O=M + 12
370 P=J-1
380 GOTO 220
390 PRINT " MONTAG ";
400 RETURN
410 PRINT " DIENSTAG ";
420 RETURN
430 PRINT " MITTWOCH ";
440 RETURN
450 PRINT " DONNERSTAG ";
460 RETURN
470 PRINT " FREITAG ";
480 RETURN
490 PRINT " SONNABEND ";
500 RETURN
510 PRINT " SONNTAG ";
520 RETURN
530 PRINT " JANUAR ";
540 RETURN
550 PRINT " FEBRUAR ";
560 RETURN
570 PRINT " MÄRZ ";
580 RETURN
590 PRINT " APRIL ";
600 RETURN
610 PRINT " MAI ";
620 RETURN
630 PRINT " JUNI ";
640 RETURN
650 PRINT " JULI ";
660 RETURN
670 PRINT " AUGUST ";
680 RETURN
690 PRINT " SEPTEMBER ";
700 RETURN
710 PRINT " OKTOBER ";
720 RETURN
730 PRINT " NOVEMBER ";
740 RETURN
750 PRINT " DEZEMBER ";
760 RETURN
770 PRINT " FEHLERHAFTE EINGABE !"
780 STOP
```

Bild 2.1 BASIC-Programm «Ewiger Kalender»

```

10 W=0'SOLLWERT
20 Y=0'ZUGEFUEHRTE ENERGIE
30 X1=0'ANFANZUSTAND
40 X2=0
50 Z=0'STOBERGROESSE
60 PRINT "REGLEERTYP"
70 PRINT:PRINT"1 = P-REGLER":PRINT
80 PRINT "2 = ZWEIPUNKTREGLER":PRINT
90 PRINT "3 = PI-REGLER":PRINT
100 INPUT V
110 PRINT "ABTASTZEIT"
120 INPUT D
130 IF D=0 THEN 240
140 FOR N=0 TO 499
150 U=W-X2
170 IF N=0 THEN 190
180 W=1'EINHEITSSPRUNG
190 GOSUB 400'REGLERAUSWAHL
200 GOSUB 290'TOTZEIT
210 GOSUB 250'VERZOEBERUNG
220 PRINT X2;TAB(X2*30);"X"
230 NEXT N
240 STOP
250 REM V E R Z O E B E R U N G
260 T1=800
270 X2=X2*(1-D/T1)+D*X1/T1
280 RETURN
290 REM T O T Z E I T G L I E D
300 H2=ABS(H1-Y)
310 IF H2<=0 THEN 370
320 T2=T2-D
330 IF T2>0 THEN 350
340 X1=Y
350 H1=Y
360 RETURN
370 T2=240
380 GOTO 350
390 '
400 ON V GOTO 610,490,420
410 RETURN
420 REM P I - R E G L E R
430 K1=1
440 T3=2400
450 Y=Y+K1*U+U*D/T3
470 X2=Y
480 RETURN
490 REM Z W E I P U N K T R E G L E R
500 U1=0.08'OBERER SCHALTWERT
510 U2=0.07
520 IF U>U1 THEN 560
530 IF U<U2 THEN 580
540 RETURN
560 Y=1
570 GOTO 540
580 Y=0
590 GOTO 540
600 '
610 REM P - R E G L E R
620 K1=1'VERSTAEKERUNG
630 Y=K1*U
650 RETURN

```

Bild 2.2 Simulationsprogramm eines Regelkreises

2.1.3. Programmbeispiele in BASIC

Einige konkrete Programmbeispiele sollen nun die Erläuterungen zur Programmiersprache BASIC untermauern. Das Bild 2.1 zeigt ein Programm, das von einem 15jährigen Schüler unter geringfügiger Anleitung erstellt wurde. Es arbeitet vor allem mit dem PRINT-Befehl sowie wenigen mathematischen Operationen und erlaubt die Bestimmung des Wochentages aus dem Datum für beliebige Tage des Gregorianischen Kalenders. Es läßt sich beispielsweise ermitteln, ob man ein «Sonntagskind» ist. Die Berechnung erfolgt nach der Formel von Zeller [12]. Man kann also auch bei der Programmierung eines Spieles seine Mathematikkenntnisse auffrischen. Viele Amateure nutzen ihre Computer vorwiegend zur Weiterbildung.

Ebenfalls unter Nutzung der komfortablen Arithmetik des BASIC-Interpreters entstand das in Bild 2.2 gezeigte Simulationsprogramm für einen einschleifigen Regelkreis, wie er in Bild 2.3 als Signalflußbild dargestellt ist. Dieses Programm erlaubt die Bestimmung der einzustellenden Reglerparameter wesentlich schneller, als es mit einem experimentellen Verfahren möglich wäre, da z. B. Temperaturprozesse recht langsam ablaufen. Dieses Programmbeispiel gibt einen Einblick in eine Form der Lösung von Problemen, die in der Volkswirtschaft zunehmend an Bedeutung gewinnt.

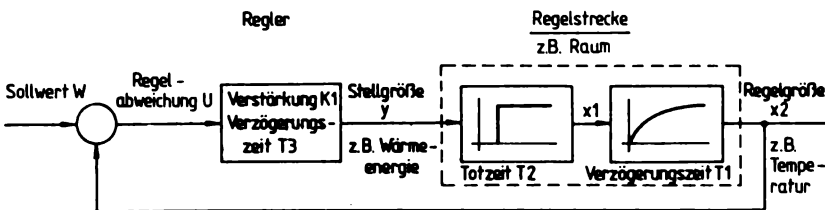


Bild 2.3 Signalflußbild des Regelkreises

2.2. Maschinenbefehle des Mikroprozessors *UB 880 D*

Die Programmiersprache BASIC hat zwar viele Vorteile, jedoch auch zwei beträchtliche Nachteile:

- Bedingt durch die zeilenweise Interpretation, erfolgt die Abarbeitung von Programmen nicht mit der technisch möglichen Geschwindigkeit.
- BASIC-Interpreter und BASIC-Programm belegen viel Speicherplatz.

Bei numerischen Berechnungen stört der erstgenannte Mangel meist nicht weiter, sollen jedoch Geräte gesteuert werden, z. B. ein EPROM-Programmierzusatz, so lassen sich die Zeitbedingungen oft nicht erfüllen. Alle Möglichkeiten, die der eingesetzte Mikroprozessor bezüglich Arbeitsgeschwindigkeit und Ausnutzung des Speicherbereiches bietet, lassen sich nur mit Programmen nutzen, die als sogenannter Maschinencode im Speicher stehen und die nicht zuvor vom Prozessor aufbereitet werden müssen. Die Kenntnis der Maschinenbefehle des Prozessors ergänzt deshalb die Möglichkeiten bei der Programmierung ganz wesentlich.

Jedes Programm muß als eine Folge von H- und L-Signalen in den Prozessor gelangen. Diese unmittelbar auf den Prozessor bezogenen Zeichen werden als Maschinensprache bezeichnet. Die byteweise Anordnung der Zeichen im Speicher kann man zwar durch die hexadezimale Schreibweise effektiv und exakt darstellen, jedoch ist diese Form weniger anschaulich. Für den Programmierer entstand deshalb eine aus Abkürzungen der Befehlswirkung gebildete Form der Darstellung, die sogenannte Mnemonik. Damit erhalten Befehle unterschiedlicher hexadezimaler Kodierung, aber ähnlicher logischer Funktion auch ähnliche Bezeichnungen, die sich besser im Gedächtnis behalten lassen. Die Zuordnung kann man in Tabellen übersichtlich erfassen. Einen wichtigen Bezugspunkt bei der Programmierung in Maschinensprache bildet dabei die interne Struktur des Mikroprozessors sowie der angeschlossenen Schaltungen und programmierbaren Bauelemente. Mit

dem Ausdruck Computerkonfiguration bezeichnet man dabei die Anordnung der Speicher (RAM und ROM) im möglichen Adreszenbereich, die Zahl und Art von Eingabe-/Ausgabeschaltungen und deren Adressen. Im Regelfall berücksichtigen Maschinenprogramme die im konkreten Fall vorliegende Schaltung mit ihren Adressen und laufen oft nur mit dieser Konfiguration. Der Befehlssatz, aus dessen Befehlen das Programm zusammengefügt ist, bezieht sich jedoch nur auf die innere Struktur des Prozessors und den grundsätzlichen Aufbau eines Computers.

Bild 2.4 zeigt die interne Struktur des Mikroprozessors *UB 880 D*. Neben der Arithmetik-Logik-Einheit (ALU), in der die Daten entsprechend einem Befehl verknüpft werden, ist vor allem der Registerblock rechts im Bild wichtig für den Programmierer. Er enthält 8-bit- und 16-bit-Speicherplätze, die dem Programmierer über Befehle direkt zugänglich sind und deren Inhalt er über die ALU verknüpfen oder von außen beeinflussen kann. Die 8-bit-Register sind schmal, die 16-bit-Register breit gezeichnet. Neben den allgemein anwendbaren Registern sind einige bevorzugt oder ausschließlich für einen besonderen Zweck bestimmt. Adreßregister und Programmzähler (PC, program counter) enthalten die aktuelle Speicheradresse des laufenden Programms. Der Kellerzeiger (SP, stack pointer) dient zur Abspeicherung der Rückkehradresse bei einem Unterprogrammaufruf. Er zeigt mit seinem Inhalt, einer Adresse, auf den nächsten freien Speicherplatz im «Keller». Der «Keller» (stack) als Teil des Speichers dient ausschließlich der Zwischenspeicherung von Rückkehradressen, denn man kann selbstverständlich Unterprogramme auch geschachtelt aufrufen. Bei Dekodierung eines Returnbefehls holt der Prozessor die Adreßteile byteweise von den Adressen, die oberhalb der im SP befindlichen im Speicher stehen und stellt den SP auf die nun freie, um 2 Positionen höhere Adresse ein. Die Aufgabe von Vektor- und Auffrischregister wird im Abschnitt 3. erläutert. Die Index-Register IX und IY sowie die Register H und L dienen vorwiegend der Speicheradressierung, wobei in H der höher-

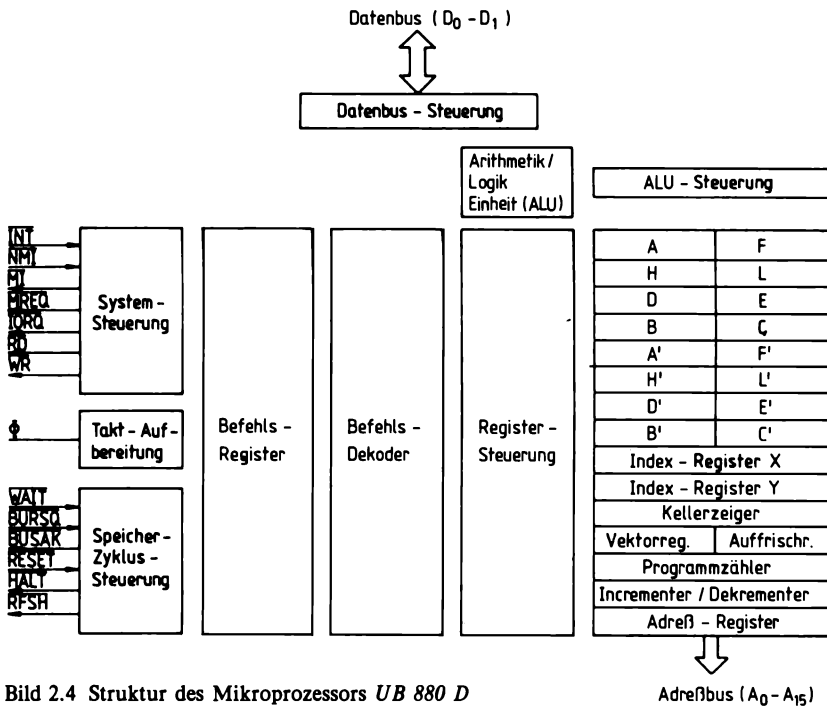
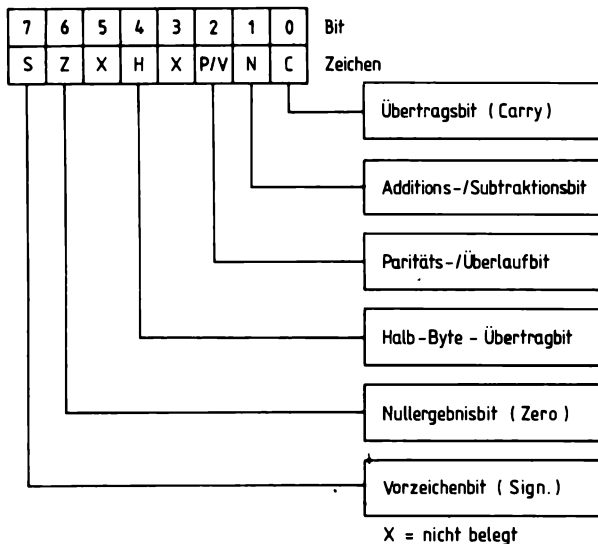


Bild 2.4 Struktur des Mikroprozessors UB 880 D


Bild 2.5
Flag-Register

2. Programmierung

wertige Adreßteil steht. Die 8-bit-Register A, F, C, D, E, H und L sind als «Schattenregister» noch einmal vorhanden, es kann jedoch immer nur eine Gruppe mit der jeweiligen Bezeichnung aktiv sein, die Umschaltung erfolgt mittels Befehl. Das Register A (Akkumulator) kann am vielseitigsten eingesetzt werden. Viele Befehle arbeiten nur mit ihm oder in Verbindung mit A und einem anderen Register. Das Register F (Flags) dient als Hilfsspeicher für arithmetische und logische Operationen. Werden z. B. zwei 8-bit-Werte addiert, so kann ein Übertrag von 1 bit auftreten, für den im Flag-Register ein Platz vorgesehen ist. Nach der Addition

kann der Programmierer das F-Register abfragen, um einen möglichen Übertrag zu erfahren. Bild 2.5 zeigt die bit-Anordnung im Flag-Register und ihre Bedeutung.

Es gibt wesentlich mehr unterschiedliche Befehle in Maschinensprache als in BASIC. Ihre ausführliche Beschreibung kann deshalb wegen des Platzbedarfs hier nicht erfolgen. Sie sind jedoch in den Tabellen 2.1. bis 2.9. vollständig enthalten. Die komplizierteren Befehle werden im Abschnitt 5.1. am Beispiel erläutert. Die Wirkung fast aller Befehle kann man auch bei den Programmbeispielen sowie den Programmen im Anhang erkennen. Die nachfolgende Übersicht ord-

Tabelle 2.3. 8-bit-Ladebefehle

Quelle

		implizit		Register								Register indirekt			indiziert		erweit. Adr.	Direkt
		I	R	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(IX+d)	(IX+d)	(nn)	n	
Ziel	Register	A	ED 57	ED 5F	7F	78	79	7A	7B	7C	7D	7E	0A	1A	DD 7E d	FD 7E d	3A n	3E n
		B			47	40	41	42	43	44	45	46			DD 46 d	FD 46 d		06 n
		C			4F	48	49	4A	4B	4C	4D	4E			DD 4E d	FD 4E d		0E n
		D			57	50	51	52	53	54	55	56			DD 56 d	FD 56 d		16 n
		E			5F	58	59	5A	5B	5C	5D	5E			DD 5E d	FD 5E d		1E n
		H			67	60	61	62	63	64	65	66			DD 66 d	FD 66 d		26 n
		L			6F	68	69	6A	6B	6C	6D	6E			DD 6E d	FD 6E d		2E n
Register indirekt	(HL)			77	70	71	72	73	74	75							36 n	
	(BC)			02														
	(DE)			72														
indiziert	(IX+d)			DD 77 d	DD 70 d	DD 71 d	DD 72 d	DD 73 d	DD 74 d	DD 75 d								DD 36 d
	(IX+d)			FD 77 d	FD 70 d	FD 71 d	FD 72 d	FD 73 d	FD 74 d	FD 75 d								FD 36 d
erweitert Adr.	(nn)			32 n														
implizit	I			ED 47														
	R			ED 4F														

Tabelle 2.4. 16-bit-Ladebefehle

		Quelle								erweit. Direkt- wert	erweit. Adr. (nn)	indir. Reg.- Adr. (SP)
		Register										
		AF	BC	DE	HL	SP	IX	IY				
Ziel	R E G I S T E R	AF										F1
		BC							01 n n	ED 48 n		C1
		DE							11 n n	ED 58 n		D1
		HL							21 n n	2A n n		E1
		SP				F9		DD F9	FD F9	31 n n	ED 78 n	
		IX							DD 21 n	DD 2A n		DD E1
		IY							FD 21 n	FD 2A n		FD E1
PUSH Befehle	erweit. Adr.	(nn)		ED 43 n	ED 53 n	22 n n	ED 73 n	DD 22 n	FD 22 n	\		
	indir. Reg.- Adr.	(SP)	F5	C5	D5	E5		DD E5	FD E5			

Achtung: Die PUSH und POP-Befehle verändern den SP-Inhalt nach jeder Ausführung

POP Befehle

net die Befehle systematisch in Funktionsgruppen ein und gibt Beispiele zur Anwendung der Befehlstabelle.

2.2.1. Lade- und Austauschbefehle

Vor ihrer Verarbeitung müssen die Datenbytes auf vielfältige Weise transportiert werden, d. h. von einem Speicherplatz oder Register in andere Speicherplätze oder Register fließen. Man unterscheidet 8-bit- und 16-bit-Ladebefehle (Tabellen 2.3., 2.4. und 2.5.). Aus der allgemeinen Form «LD Ziel, Quelle» ermittelt man die konkreten Befehle, z. B.: LD HL, 1234 mit der hexadezimalen Kodierung 21 34 12 oder LD A, (HL) ergibt den hexadezimalen Kode 7E. Ähnlich wie bei BASIC folgen mögliche Operanden dem Code für die Operation. Bei Adressen folgt zunächst das niederwertige Byte, wie das erste Beispiel zeigt. Der 2. Befehl zeigt

die sogenannte indirekte Adressierung, dies ist aus den Klammern um den Operanden ersichtlich. Bei diesem Befehl wird der Akkumulator A mit dem Dateninhalt desjenigen Speicherplatzes geladen, dessen Adresse im Registerpaar HL steht. Enthält z. B. der Speicherplatz mit der binären Adresse 1234 als Dateninhalt das ASCII-Zeichen «B» (entsprechend 42H), so steht dieses Zeichen nach der Ausführung beider Befehle im Ak-

Tabelle 2.5. Austauschbefehle

		implizite Adressierung				
		AF	BC, DE u. HL	HL	IX	IY
implizit	AF	08				
	BC DE u. HL		D9			
	DE			EB		
	indir. Reg.- Adr.	(SP)		E3	DD E3	FD E3

Tabelle 2.6. Block-Ladebefehle

		Quelle	
		indir. Reg. Adr.	(HL)
Ziel	indir. Reg. Adr.	(DE)	ED A0 "LDI"-Laden (DE) ← (HL) Erhöhen HL u. DE, Erniedrigen BC
			ED B0 "LDIR"-Laden (DE) ← (HL) Erhöhen HL u. DE, Erniedrigen BC, wdh. bis BC=0
			ED A8 "LDD"-Laden (DE) ← (HL) Erniedrigen HL u. DE, Erniedrigen BC
			ED B8 "LDDR"-Laden (DE) ← (HL) Erniedrigen HL u. DE, Erniedrigen BC, wdh. bis BC=0

Reg. HL bezeichnet die Quelle
Reg. DE bez. das Ziel
Reg. BC ist der Bytezähler

kumulator (A) des Prozessors. Der vorherige Inhalt des Akkumulators ist damit überschrieben. Der Speicherplatz 1234 ist nun jedoch keinesfalls «leer», denn sein Inhalt wurde mit dem Ladebefehl nur kopiert, nicht «entladen». Diese Arbeitsweise gilt für alle Ladebefehle. Nach den Befehlen LD A, B oder PUSH DE befinden sich im Register B bzw. im Registerpaar DE weiterhin die gleichen Inhalte wie vor der Ausführung des Befehls.

Die Möglichkeiten der indirekten Adressierung beschränken sich nicht auf die Ladebefehle, sondern existieren auch bei den meisten anderen Befehlsgruppen.

2.2.2. Speicherbereichsbefehle

Während sich die Befehle der Tabellen 2.3. bis 2.5. auf 1 oder 2 Byte beziehen, erlauben es die der Tabellen 2.6. und 2.7., ganze Bereiche des Speichers zu transportieren oder mit dem Inhalt des Akkumulators zu vergleichen.

2.2.3. Arithmetische und Logikbefehle

Die Tabellen 2.8., 2.9. und 2.10. enthalten die Befehle der 8-bit- und 16-bit-Arithmetik, Logikbefehle und Operationen mit den Registern A und F.

Tabelle 2.7. Block-Vergleichsbefehle

		SUCH - Adresse
indir. Reg. Adr.	(HL)	
ED A1	"CPI"	Erhöhen HL, Erniedrigen BC
ED B1	"CPIR"	Erhöhen HL, Erniedrigen BC, wdh. bis BC=0 oder bis Übereinstimmung gefunden wird
ED A9	"CPD"	Erniedrigen von HL u. BC
AD B9	"CPDR"	Erniedrigen von HL u. BC, wdh. bis BC=0 oder bis Übereinstimmung gefunden wird

HL bezeichnet den Speicherplatz, der mit dem AC-Inhalt verglichen werden soll.
BC ist der Bytezähler

Da die theoretischen Grundlagen der Digitaltechnik bei den Amateuren oft schnell in Vergessenheit geraten, soll auf diese Befehle trotz der Beispiele im Abschnitt 5. näher eingegangen werden. Eine Addition setzt 2 zu verknüpfende Summanden voraus. Bei den 8-bit-Befehlen muß sich einer davon im Akkumulator befinden, der andere wird durch die «Quelle» adressiert. Bild 2.6a gibt dazu ein Beispiel: Der Akkuinhalt beträgt 0C0H, im Register B steht der Wert 7EH. Nach Ausführung des Befehls ADD A,B (Befehlscode 80H) in der Arithmetik-Logik-Einheit (ALU in Bild 2.4) enthält der Akku das Ergebnis. Die Addition erfolgt mit 8 bit parallel. Um handschriftlich solche Rechnungen auszuführen, schreibt man die Binärstellen wie im Bild 2.6a gezeigt und rechnet in

Tabelle 2.8. 8-bit-Arithmetikbefehle

Quelle											
	Registrieradressierung								indir. Reg. adr.	indizierte Adressierung	Direkt- wert
	A	B	C	D	E	H	L	(HL)	(R-d)	(Y-d)	n
Addition ADD	87	80	81	82	83	84	85	86	DD 86 d	FD 86 d	C6 n
Addition mit Übertrag ADC	8F	88	89	8A	8B	8C	8D	8E	DD 8E d	FD 8E d	CE n
Subtraktion SUB	97	90	91	92	93	94	95	96	DD 96 d	FD 96 d	D6 n
Subtr. m. Übertr. SBC	9F	98	99	9A	9B	9C	9D	9E	DD 9E d	FD 9E d	DE n
"UND" AND	A7	A0	A1	A2	A3	A4	A5	A6	DD A6 d	FD A6 d	E6 n
exklusives oder XOR	AF	A8	A9	AA	AB	AC	AD	AE	DD AE d	FD AE d	EE n
oder OR	B7	B0	B1	B2	B3	B4	B5	B6	DD B6 d	FD B6 d	F6 n
Vergleich CP	BF	B8	B9	BA	BB	BC	BD	BE	DD BE d	FD BE d	FE n
Erhö. um 1 INC	3C	04	0C	14	1C	24	2C	34	DD 34 d	FD 34 d	
Ernied. um 1 DEC	3D	05	0D	15	1D	25	2D	35	DD 35 d	FD 35 d	

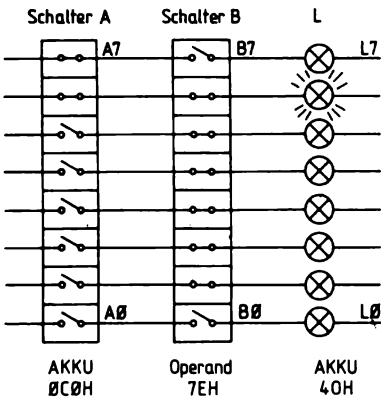
011111110 7EH Register B

111000000 0C0H AKKU vor Addition

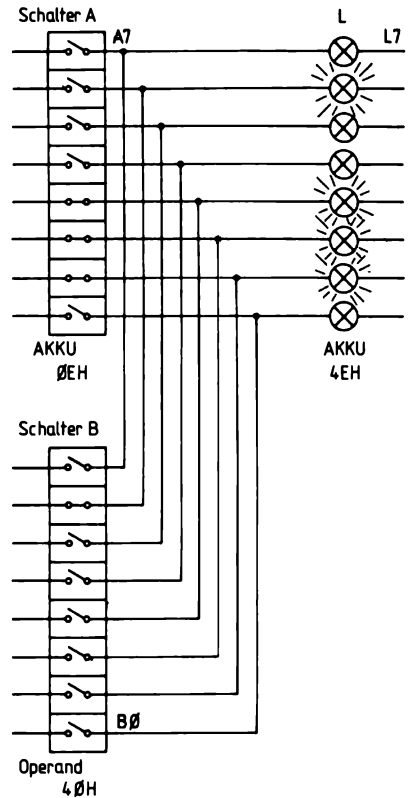
C-Flag: 1 001111110 3EH AKKU nach Addition

Bild 2.6 Erläuterung der arithmetischen und Logikbefehle;

a - binäre Addition,



b - AND-Befehl,



c - OR-Befehl

2. Programmierung

Tabelle 2.9. 16-bit-Arithmetikbefehle

		Quelle					
		BC	DE	HL	SP	IX	IY
Ziel	ADD	HL	09	79	29	39	
		IX	DD	DD		DD	DD
		IY	FD	FD		FD	FD
	Addition mit Übertrag Flagsetzer ADC	HL	ED	ED	ED	ED	
	Subtr. mit Übertrag u. Flagsetzer SBC	HL	ED	ED	ED	ED	
	Erhöhen um 1 INC		03	13	23	33	DD
	Erniedrigen um 1 DEC		0B	1B	2B	3B	DD

der üblichen Form von rechts beginnend die einzelnen Positionen aus. Dabei gilt: $0 + 0 = 0$, $0 + 1 = 1$, $1 + 1 = 10$. Die letzte binäre Gleichung enthält in der führenden 1 des Ergebnisses den Übertrag, der im Gegensatz zur herkömmlichen dezimalen Arithmetik hier nur den Wert 1 annehmen kann. Im Mikroprozessor hat dieses bit, wie bereits erwähnt, im Flag-Register seinen Platz. Bei der Addition mit Übertrag geht der Inhalt dieses bits in die Addition ein, d. h., wenn es von einer vorherigen Operation her gesetzt ist, wird ein dritter Operand mit dem Wert 01H nach obiger Vorschrift zusätzlich zum

Akkuinhalt addiert. Die Befehle SUB und SBC arbeiten entsprechend, wobei die Rechenvorschriften lauten: $0 - 0 = 0$, $1 - 0 = 1$, $1 - 1 = 0$, $0 - 1 = 11$. Die Logikbefehle AND, OR und XOR gibt es auch im BASIC-Interpreter, jedoch kann man sie mit Beispielen von Prozessorregistern anschaulicher erläutern. Bild 2.6b zeigt eine Kontaktschaltung, bei der die Lampen L0 bis L7 nur leuchten, wenn die jeweiligen Schalter A0 bis A7 UND B0 bis B7 beide geschlossen sind. Wie die untersten Zeilen zeigen, läßt sich die Funktion dieser UND-Glieder auf den AND-Befehl leicht übertragen (and = engl. und). In der Praxis dient dieser Befehl häufig zum Ausblenden von bits. Beispielsweise kann man mit dem Operanden 0FH die untere Tetrade aus einem Byte erhalten (siehe Monitorprogramm im Anhang).

Der OR-Befehl (or engl. oder) entspricht sinngemäß der Schaltung von Bild 2.6c. Wenn mindestens ein Schalter A ODER B geschlossen ist, leuchtet die zugehörige Lampe. Wie das Beispiel zeigt, nutzt man diesen Befehl häufig, um in einem 8-bit-Binarwert eine Stelle auf High-Pegel zu setzen. Der Befehl XOR läßt sich schlechter an einem Bild erläutern, weil die zugehörige Schaltung unübersichtlich ist. Nach seiner Abarbeitung sind die bits bei den Operanden, die sich im Pegel unterscheiden, im Ak-

Tabelle 2.10. AF-Operationen und Steuerbefehle

Einstellen d. Dezimale des Akkumulators	DAA	27
Komplement des Akkumulators	CPL	2F
Vorzeichenumkehr des Akkumulators (Zweierkomplement)	NEG	ED 44
Komplement des Übertrags - Flags	CCF	3F
Setzen des Übertrags - Flags	SCF	37

NOP	00
HALT	76
INT abweisen DI	F3
INT annehmen EI	FB
Setzen Int. mode 0 IM0	ED 46
Setzen Int. mode 1 IM1	ED 56
Setzen Int. mode 2 IM2	ED 5E

8080A - Mode

CALL zu Adresse 003BH

Indirektes CALL unter Verwendung von Reg. I u. 8 Bit vom Int. auslösenden Gerät als Zeiger.

Tabelle 2.11. Beeinflussung des Flag-Registers

Befehl	Flags				Bemerkungen			
	C	Z	P/V	S	N	H		
ADD s, ADC s	↑	↑	V	↑	0	↑	8-bit-Addition oder Addition mit Übertrag	
SUB s, SBC s, CMP s, NEG	↑	↑	V	↑	1	↑	8-bit-Subtraktion, Subtraktion mit Übertrag, Vergleich und Negation des Akkumulators	
AND s	0	↑	P	↑	0	1	Logische Operationen	
OR s, XOR s	0	↑	P	↑	0	0	8-bit-Erhöhung	
INC s		↑	V	↑	0	↑	8-bit-Erniedrigung	
DEC s		↑	V	↑	1	↑	16-bit-Addition	
ADD HL, dd	↑	.	.	.	0	X	16-bit-Subtraktion mit Übertrag	
ADC HL, dd	↑	↑	V	↑	0	X	16-bit-Subtraktion mit Übertrag	
SBC HL, dd	↑	↑	V	↑	1	X	Zyklische Verschiebung – Akkumulator	
RLA, RLCA, RRA, RRCA	↑	.	.	.	0	0	Zyklische Verschiebung – Speicherplatz s	
RL s, RLC s, RR s, RRC s	↑	↑	P	↑	0	0	Verschiebung – Speicherplatz s	
SLA s, SRA s, SRL s	↑	↑	P	↑	0	0	Zyklische Verschiebung – Zahl links und rechts	
RLD, RRD	.	↑	P	↑	0	0	Dezimale Einrichtung – Akkumulator	
DAA	↑	↑	P	↑	1	↑	Komplement des Akkumulators	
CPL	1				0	0	Setzen des Übertrags	
SCF	↑				0	0	Komplement des Übertrags	
CCF	↑	.		.	0	X	Eingabe, indirekte Registeradresse	
IN r, INF		↑	P	↑	0	0	Block-Ein- und Ausgabe, Z = 0, wenn B ≠ 0, sonst Z = 1	
INI, IND, OUTI, OUTD		↑	X	X	1	X	Z = 0, wenn B ≠ 0, sonst Z = 1	
INIR, INDR, OTIR, OTDR		1	X	X	1	X	Block-Transfer-Befehle	
LDI, LDD		X	↑	X	0	0	P/V = 1, wenn BC ≠ 0, sonst P/V = 0	
LDIR, LDDR		X	0	X	0	0	Block-Such-Befehle, Z = 1, wenn A = (HL), sonst Z = 0;	
CPI, CPIR, CPD, CPDR		↑	↑	X	1	X	P/V = 1, wenn BC ≠ 0, sonst P/V = 0	
LD A, I; LD A, R		↑	IFF2		0	0	Inhalt des Interrupt-Annahme-Flip-Flop 2 (IFF2) ins P/V-Flag überführt	
BIT b, s		↑	X	X	0	1	Zustand des bits b im Speicherplatz s ins Z-Flag überführt	

Symbol	Bedeutung
C	Übertragsflag. C = 1, wenn die Operation einen Übertrag vom MSB des Operanden oder des Ergebnisses erzeugt.
Z	Null-Flag. Z = 1, wenn das Ergebnis der Operation Null ist.
S	Vorzeichen-Flag. S = 1, wenn das MSB des Ergebnisses Eins ist.
P/V	Paritäts- oder Überlauf-Flag. Parität (P) und Überlauf (V) benutzen das gleiche Flag. Logische Operationen beeinflussen das Flag entsprechend der Parität des Ergebnisses, arithmetische Operationen stellen dieses Flag entsprechend dem Überlauf des Ergebnisses. P/V = 1, wenn das Ergebnis paarig ist, P/V = 0, wenn das Ergebnis unpaarig ist. P/V = 1, wenn das Ergebnis einen Überlauf enthält.
H	Halbbyte-Übertragsflag. H = 1, wenn Addition oder Subtraktion einen Übertrag innerhalb von 4 Akkumulatorbits erzeugen.
N	Additions-/Subtraktionsflag. N = 1, wenn vorangegangene Operation eine Subtraktion war.
	H- und N-Flags werden für die Dezimalkorrektur (DAA) benutzt, um das Ergebnis einer Addition oder Subtraktion von gepackten BCD-Zahlen in das Format gepackter BCD-Zahlen zu wandeln.
↑	Flag wird entsprechend dem Ergebnis der Operation gestellt.
0	Flag wird durch die Operation nicht beeinflusst.
1	Flag wird durch die Operation gelöscht.
X	Flag wird durch die Operation gesetzt.
	Flag unbestimmt.
V	P/V-Flag entspricht dem Ergebnis-Überlauf der Operation.
P	P/V-Flag entspricht der Parität des Ergebnisses der Operation.
r	Eines der U-880-D-Register A, B, C, D, E, H, L.
s	Ein 8-bit-Speicherplatz, der durch eine der für den jeweiligen Befehl zulässigen Adressierungsarten definiert ist.
dd	Ein 16-bit-Speicherplatz, der durch eine der für diesen Befehl zulässigen Adressierungsarten definiert ist.
ii	Eines der zwei Indexregister IX oder IY
R	Auffrischzähler
n	8 bit im Bereich 0–255
nn	16 bit im Bereich 0–65 535

kumulator auf H-Pegel gesetzt; bits mit gleichem Pegel stehen auf Null. Dabei ist es gleichgültig, ob die Übereinstimmung bei H-Pegel oder L-Pegel vorliegt.

Auch der Vergleichsbefehl bezieht sich immer auf den Inhalt des Akkumulators, jedoch wird dessen Inhalt bei der Ausführung des Befehls im Gegensatz zu den zuvor genannten Befehlen nicht verändert. Dadurch sind mehrere Vergleiche mit einem Akkuinhalt und verschiedenen Operanden möglich. Die Beeinflussung des Flag-Registers erfolgt wie bei einer Subtraktion (siehe Anhang).

Die 8-bit-Inkrement-Befehle bedürfen keiner weiteren Erläuterung. Zu beachten ist, daß sie im Gegensatz zu den 16-bit-Inkrement-Befehlen der Tabelle 2.9. das Flag-Register beeinflussen.

Tabelle 2.11. zeigt, welche Flags schalten. Befehle, die in dieser Tabelle fehlen, beeinflussen das Flag-Register nicht.

2.2.4. Verschiebepfehle

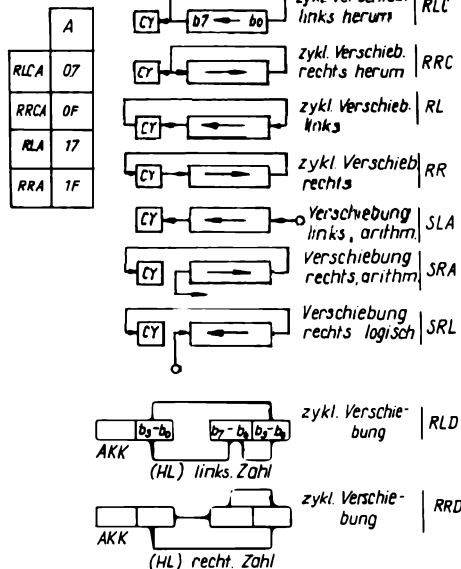
Auch das Verschieben eines Bytes in einem Register ist im Prinzip eine arithmetische Operation, verändert sich doch dabei der Wert des Bytes um eine Binärstelle. Tabelle 2.12. enthält die vielfältigen Möglichkeiten. Die nebenstehenden Darstellungen zeigen im Kästchen CY das Übertragflag (Carry).

2.2.5. Bitbefehle

Mit der Vielzahl der Bitbefehle kann man in den ansonsten nur byteweise organisierten Registern und Speicherplätzen einzelne Datenbits auf ihren Pegel prüfen und auf L- oder H-Pegel schalten. Tabelle 2.13. zeigt die Aufstellung.

Tabelle 2.12. Verschiebepfehle

Quelle und Ziel															
	A	B	C	D	E	H	L	(HL)	(X+d)	(Y+d)					
RLC	CB 07	CB 00	CB 01	CB 02	CB 03	CB 04	CB 05	CB 06	DD CB 07	FD CB 08					
RRC	CB 0F	CB 08	CB 09	CB 0A	CB 0B	CB 0C	CB 0D	CB 0E	DD CB 0F	FD CB 10					
RL	CB 17	CB 10	CB 11	CB 12	CB 13	CB 14	CB 15	CB 16	DD CB 17	FD CB 18					
RR	CB 1F	CB 18	CB 19	CB 1A	CB 1B	CB 1C	CB 1D	CB 1E	DD CB 1F	FD CB 20					
SLA	CB 27	CB 20	CB 21	CB 22	CB 23	CB 24	CB 25	CB 26	DD CB 27	FD CB 28					
SRA	CB 2F	CB 28	CB 29	CB 2A	CB 2B	CB 2C	CB 2D	CB 2E	DD CB 2F	FD CB 30					
JRL	CB 3F	CB 38	CB 39	CB 3A	CB 3B	CB 3C	CB 3D	CB 3E	DD CB 3F	FD CB 40					
RLD								ED 6F							
RRD								ED 67							



2. Programmierung

Tabelle 2.13. Bitbefehle

		Registeradressierung							indi. Reg.	indiziert	
		A	B	C	D	E	H	L	(HL)	(R+d)	(R+d)
TEST "BIT"	0	CB 47	CB 48	CB 49	CB 4A	CB 4B	CB 4C	CB 4D	CB 4E	CB 4F	CB 4F
	1	CB 4F	CB 48	CB 49	CB 4A	CB 4B	CB 4C	CB 4D	CB 4E	CB 4F	CB 4F
	2	CB 57	CB 50	CB 51	CB 52	CB 53	CB 54	CB 55	CB 56	CB 57	CB 57
	3	CB 5F	CB 58	CB 59	CB 5A	CB 5B	CB 5C	CB 5D	CB 5E	CB 5F	CB 5F
	4	CB 67	CB 60	CB 61	CB 62	CB 63	CB 64	CB 65	CB 66	CB 67	CB 67
	5	CB 6F	CB 68	CB 69	CB 6A	CB 6B	CB 6C	CB 6D	CB 6E	CB 6F	CB 6F
	6	CB 77	CB 70	CB 71	CB 72	CB 73	CB 74	CB 75	CB 76	CB 77	CB 77
	7	CB 7F	CB 78	CB 79	CB 7A	CB 7B	CB 7C	CB 7D	CB 7E	CB 7F	CB 7F
LÖSCHEN "RES"	0	CB 87	CB 80	CB 81	CB 82	CB 83	CB 84	CB 85	CB 86	CB 87	CB 87
	1	CB 8F	CB 88	CB 89	CB 8A	CB 8B	CB 8C	CB 8D	CB 8E	CB 8F	CB 8F
	2	CB 97	CB 90	CB 91	CB 92	CB 93	CB 94	CB 95	CB 96	CB 97	CB 97
	3	CB 9F	CB 98	CB 99	CB 9A	CB 9B	CB 9C	CB 9D	CB 9E	CB 9F	CB 9F
	4	CB A7	CB A0	CB A1	CB A2	CB A3	CB A4	CB A5	CB A6	CB A7	CB A7
	5	CB AF	CB A8	CB A9	CB AA	CB AB	CB AC	CB AD	CB AE	CB AF	CB AF
	6	CB B7	CB B0	CB B1	CB B2	CB B3	CB B4	CB B5	CB B6	CB B7	CB B7
	7	CB BF	CB B8	CB B9	CB BA	CB BB	CB BC	CB BD	CB BE	CB BF	CB BF
BITSETZEN "SET"	0	CB C7	CB C0	CB C1	CB C2	CB C3	CB C4	CB C5	CB C6	CB C7	CB C7
	1	CB CF	CB C8	CB C9	CB CA	CB CB	CB CC	CB CD	CB CE	CB CF	CB CF
	2	CB D7	CB D0	CB D1	CB D2	CB D3	CB D4	CB D5	CB D6	CB D7	CB D7
	3	CB DF	CB D8	CB D9	CB DA	CB DB	CB DC	CB DD	CB DE	CB DF	CB DF
	4	CB E7	CB E0	CB E1	CB E2	CB E3	CB E4	CB E5	CB E6	CB E7	CB E7
	5	CB EF	CB E8	CB E9	CB EA	CB EB	CB EC	CB ED	CB EE	CB EF	CB EF
	6	CB F7	CB F0	CB F1	CB F2	CB F3	CB F4	CB F5	CB F6	CB F7	CB F7
	7	CB FF	CB F8	CB F9	CB FA	CB FB	CB FC	CB FD	CB FE	CB FF	CB FF

2.2.6. Eingabe-/Ausgabebefehle

Die Tabellen 2.14. und 2.15. enthalten Befehle, die mit den je 256 Adressen für Eingabe- und Ausgabekanäle arbeiten. In ähnlicher Form wie bei den Speicherbefehlen gibt es auch hier Befehle für Datenblöcke. Wegen der geringeren Anzahl der Adressen reicht der Inhalt von Register C zur Adressierung aller möglichen Kanäle. Register B arbeitet als Bytezähler.

Tabelle 2.14. Eingabebefehle

		Kanaladresse			
			Direktwert	indir. Reg.	
			n	c	
Eingabe Ziel	Eingabe IN	REGISTERSTELLE	A	DB n	ED 78
			B		ED 40
			C		ED 48
			D		ED 50
			E		ED 58
			H		ED 60
			L		ED 68
Eingabe + Erh. HL Ern. B INI		indirekt Register (HL)		ED A2	
Eingabe Erh. HL, Ern. B. wdh. - B = 0 INIR				ED B2	
Eingabe + Ern. HL + Ern. B IND				ED AA	
Eingabe + Ern. HL + Ern. B wdh. - B = 0 INDR				ED BA	

Block - Eingabe - Befehle

Block - Eingabe - Befehle

2.2.7. Befehle für Sprünge,
Unterprogrammaufruf
und Rücksprung

In Tabelle 2.16. sind Befehle mit den oben genannten Aufgaben enthalten. Die meisten arbeiten mit Bedingungen, ähnlich wie die Kombinationen «IF ... THEN» und «ON ... GOSUB» in BASIC. Dabei erfolgt die Auswertung der im Flag-Register von den vorangegangenen Befehlen hinterlassenen Informationen. Die Vielzahl dieser Befehle verweist auf die verschiedenartigen Möglichkeiten, ein Programm zu formulieren.

Eine Sonderstellung nehmen die Restartbefehle (Tabelle 2.17.) ein. Es handelt sich dabei auch um den Aufruf eines Unterprogramms, d. h., es wird eine RET-Adresse im Stack gespeichert. Der Aufruf benötigt aber keine Adressenangabe als Operanden und ist deshalb um 2 Byte (das sind 60 %) kürzer als die anderen CALL-Befehle. Die Adressen sind den RST-Befehlen fest zugeordnet (siehe Tabelle) und befinden sich alle am Anfang des Adressenbereiches. Soll Speicherplatz gespart werden, so kann man oft benötigte Unterprogramme auf diese Adressen legen.

Tabelle 2.15.
Ausgabebefehle

		Quelle							indir. Reg.
		A	B	C	D	E	H	L	(HL)
OUT	Direktwert	n	D3 n						
	indir. Reg.	(C)	ED 79	ED 41	ED 49	ED 51	ED 59	ED 61	ED 69
Ausgabe + Erh. HL + Ern. B OUTI		indir. Reg.	(C)						ED A3
Ausgabe + Erh. HL + Ern. B wdh. - B = 0 OUTIR		indir. Reg.	(C)						ED B3
Ausgabe + Ern. HL + B OUTD		indir. Reg.	(C)						ED AB
Ausgabe + Ern. HL + B wdh. - B = 0 OUTDR		indir. Reg.	(C)						ED BB

Block - Ausgabe - Befehle

Ziel - Kanaladresse

Tabelle 2.16. Sprung-/Rücksprungbefehle und Unterprogrammaufruf

		Bedingung											
			unbedingte	Übertrag C	kein Übertr. NC	= 0 Z	≠ 0 NZ	paritätsang. PE	paritätsang. PO	< 0 M	> 0 P	Reg. B ≠ 0	
Sprung *JP*	erweit. Direktiv.	nn	C3 n n	DA n n	D2 n n	CA n n	C2 n n	EA n n	E2 n n	FA n n	F2 n n		
Sprung JR	relativ	PC+e	18 e-2	38 e-2	30 e-2	28 e-2	20 e-2						
Sprung JP	indirektes Register	(HL)	E9										
Sprung JP		(IX)	DD E9										
JP		(IY)	FD E9										
CALL	erweit. Direktiv.	nn	CD n n	DC n n	D4 n n	CC n n	C4 n n	EC n n	E4 n n	FC n n	F4 n n		
Ernie. B., Sp. bei ≠ 0 DJNZ	relativ	PC+e											10 e-2
Rücksprung RET	indirekt Register	(SP) (SP+1)	C9	D8	D0	C8	C0	E8	E0	F8	F0		
Rückspr. v. Interrupt RETI	indirekt Register	(SP) (SP+1)	ED 4D										
Rückspr. nicht mask. Interr. RETN	indirekt Register	(SP) (SP+1)	ED 45										

Achtung: Gewisse Flags haben mehr als eine Bedeutung

Tabelle 2.17. Restart-Befehle

		OP CODE	
C A L L	0000 _N	C7	RST 0
	0008 _N	CF	RST 8
	0010 _N	D7	RST 16
A D R E S S E	0018 _N	DF	RST 24
	0020 _N	E7	RST 32
	0028 _N	EF	RST 40
	0030 _N	F7	RST 48
	0038 _N	FF	RST 56

2.3. Monitorprogramm

Nach der Vorstellung der Befehle des Mikroprozessors soll nun die Programmierung in Maschinensprache folgen.

Bei der Schaltungs- oder Programmentwicklung muß der Bediener mit dem Mikrocomputer über die Peripherie (Tastatur, Bildschirm usw.) in Verbindung treten können, muß Programme eingeben, im Speicher kontrollieren, starten und verändern. Dazu ist ein sogenanntes Monitorprogramm erforderlich, welches meist fest im ROM des Mikrocomputers untergebracht ist und beim Einschalten als erstes arbeitet. Dieses Monitorprogramm, oft auch als «der Monitor» bezeichnet, besteht im wesentlichen aus zwei Teilen:

- Programme für den jeweils vorhandenen Computer bezüglich Speichergröße und -ausführung sowie Peripherie (z. B. Anzahl der Tasten oder Bildschirmzeilen).
- Programme, die sich auf den eingesetzten Prozessor beziehen und von der Konfiguration unabhängig sind.

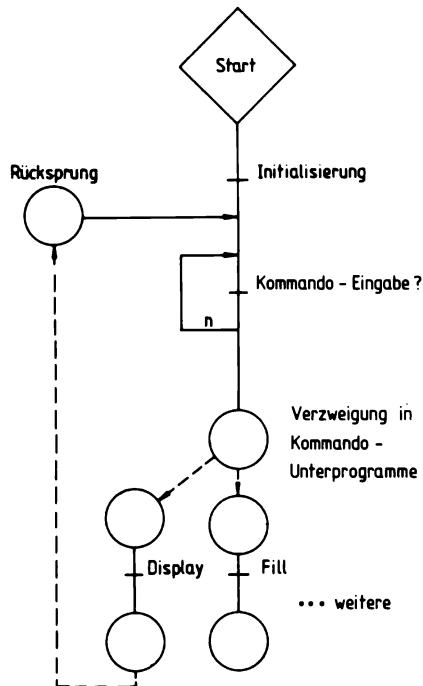
Für beides hat sich inzwischen eine Art unverbindlicher Standard herausgebildet, der in der Literatur auch ausführlich beschrieben ist und vor allem eine optimale Ausnutzung der Speicherkapazität des Monitorprogramms erlaubt.

2.3.1. Struktur

Bild 2.7 zeigt die Struktur des Monitorprogramms. Nach dem Einschalten bzw. nach dem RESET-Signal erfolgt zunächst die Initialisierung. Die entsprechend der jeweiligen Hardware-Konfiguration vorhandenen programmierbaren Peripheralschaltkreise PIO, CTC, SIQ usw. programmiert man je nach

ihrer Verwendung, z. B. ein PIO für eine Tastatur bei Port A auf Ausgabe und bei Port B auf Eingabe. Der Speicher für den Bildschirm sowie die unter 2.5. näher erläuterte RAM-Tabelle des Monitorprogramms, die nach dem Einschalten einen zufälligen Inhalt aufweisen, müssen einen definierten Anfangszustand erhalten. Der Bildschirm wird «gelöscht», indem der Prozessor den Bildwiederholpeicher mit Leerzeichen beschreibt. Nach der Initialisierung gelangt das Programm in die Aufrufschleife und erwartet die Eingabe eines Monitorkommandos. Die Kommandos werden als Unterprogramme abgearbeitet, anschließend erfolgt die Rückkehr in die Aufrufschleife, und es kann eine neue Kommandoingabe erfolgen.

Die Programmteile für die Kommandos rufen weitere Unterprogramme auf, z. B. für eine Konvertierung in eine andere Form der Darstellung oder für die Ausgabe eines Zeichens auf dem Bildschirm.



Gerätebezogene Unterprogramme: z.B. Eingabe eines Zeichens von der Tastatur, Ausgabe auf Bildschirm usw.

allgemeine Unterprogramme: z.B. Konvertierung eines ASCII-Zeichens in ein hexadezimalen Halbbyte u.s.w.

2.3.2. IN/OUT-Routinen

Das zuletzt genannte Unterprogramm wird außer vom Monitor auch von anderen Programmen benötigt. Da alle Programme irgendwelche Daten über Eingabekanäle erhalten und nach der Bearbeitung wieder ausgeben, sind die Unterprogramme für Peripheriegeräte wie Tastatur, Bildschirm und externe Speicher (Kassettenrecorder, Lochband usw.) bei jedem Mikrocomputer von allgemeinem Interesse.

Im vorliegenden Monitor erfolgt die Verarbeitung in Form von Einzelzeichen, üblich ist auch die Verarbeitung von Datenblöcken.

2.3.3. Einsprungtabelle

Es besteht natürlich die Möglichkeit, aus anderen Programmen die IN/OUT-Routinen direkt aufzurufen (CALL-Befehl). Bei einer Änderung des Monitorprogramms müßten jedoch auch Adressen in allen anderen Programmen (z. B. BASIC-Interpreter, Editor, Assembler) geändert werden. Deshalb fügt

Bild 2.7 Struktur des Monitors

2. Programmierung

Tabelle 2.18. Einsprungtabelle

Adresse	Befehl.	Bemerkung
0E0H	MCI: JMP CI	Einzelzeicheneingabe Tastatur, Zeichen in Register A
0E3H	MRI: JMP RI	Einzelzeicheneingabe INPUT-Kanal, Zeichen in Register A
0E6H	MCO: JMP CO	Einzelzeichenausgabe Bildschirm, Zeichen in Register C
0E9H	MWO: JMP WO	Einzelzeichenausgabe OUTPUT-Kanal, Zeichen in Register C
0ECH	MLO: JMP LO	Einzelzeichenausgabe LIST-Kanal, Zeichen in Register C
0EFH	MCSTS: JMP CSTS	Tastaturstatus 0 = kein Zeichen -1 = Zeichen
0F2H	MIOC: JMP IOCHK	IN/OUT-Zuweisung abfragen, Byte in Register A
0F5H	MIOS: JMP IOSET	IN/OUT-Gerät zuweisen, Byte in Register A

man eine Tabelle mit Sprungbefehlen zu den IN/OUT-Routinen an einer Stelle ein, wo Änderungen nicht zu erwarten sind, z. B. hinter den Restart-Adressen am Anfang des Monitors. Im vorliegenden Monitor ist die Tabelle 2.18. vorhanden.

Anstelle der wirklichen IN/OUT-Routine kann nun eine Adresse dieser Sprungtabelle, z. B. Adresse 00E0H für eine Zeicheneingabe von der Tastatur, aufgerufen werden. Änderungen der Routine CI selbst wirken sich so auf übergeordnete Programme nicht aus.

2.3.4. IN/OUT-Zuweisung

Die Ausgabe und Eingabe von Zeichen erfolgt meist über verschiedene Peripheriegeräte. So kann ein Programm durch den INPUT-Kanal mit der Routine MRI sowohl über die Tastatur, über ein Kassettenmagnetbandgerät oder auch von einem anderen externen Speicher in den Mikrocomputer gelangen. Gleiches gilt für die Ausgabe: Mit der Routine MLO kann z. B. auf den Bildschirm, auf einen externen Speicher, auf verschiedene Drucker oder andere Geräte ausgegeben werden.

Die Auswahl des physisch vorhandenen Geräts «Tastatur» zur logischen Funktion «INPUT-Kanal» (Zeicheneingabe) speichert der Mikrocomputer in einem Byte. Beim Aufruf der Routine MRI wird dieses IN/OUT-Byte bei jedem Zeichen abgefragt und ausgewertet. Die Ausgabekanäle OUTPUT und LIST arbeiten in gleicher Weise. Die Änderung des IN/OUT-Bytes und der damit

verbundene Wechsel der physischen Geräte erfolgt mit Monitorkommandos.

2.3.5. Debugger

Der auf den Mikroprozessor und nicht auf das Gerätekonzept zugeschnittene Teil des Monitors erlaubt vor allem die Arbeit mit dem Speicher, z. B. lesen, schreiben, ändern, Programme starten. Da diese Aktivitäten zu meist beim Test und bei der Fehlersuche an Programmen gebraucht werden, hat sich hierfür die Bezeichnung DEBUG-Programm oder DEBUGGER eingebürgert. Die Tabelle 2.19. gibt eine Übersicht der Monitorkommandos, ihren Anwendungszweck und die Form, in der die Kommandos eingegeben werden müssen. Einige bestehen nur aus einem Sprung in ein anderes Programm, das dann nur die IN/OUT-Routinen benutzt (z. B. BASIC). Das erste Zeichen, ein Buchstabe, ist die eigentliche Auswahl des Kommandos. Meist sind dann noch weitere «Parameter», wie Speicheradressen oder Datenbytes, erforderlich.

Die Eingabe von Adressen und Daten erfolgt im Hexadezimalcode, wobei sowohl große als auch kleine Zeichen zulässig sind. Ein Beispiel soll die Bedienung des Monitors näher erläutern: Die Eingabe D0,F (CR) ergibt die Bildschirmausschrift des Inhalts der Speicherplätze 1 bis 15.

Das Komma dient als Trennzeichen, es schließt die Eingabe der ersten Adressen ab. Als weiteres mögliches Trennzeichen dient auch das Leerzeichen (Space-Taste).

Tabelle 2.19. Monitorkommandos

Kommando	Funktion	Syntax, Parameter
A	Assembler-Aufruf	A
B	BASIC-Aufruf	B
C	COPY WO: = RI	C
D	DUMP	DAADR, EADR
E	End-File	EAADR
F	Fill	FAADR, EADR, Byte
G	GO	GAADR, EADR1, EADR2 oder GAADR oder G
H	HEX-Arithmetik	HADR, AD
I	INPUT-Kanal zuweisen	IT oder IL oder IU
J	JUMP RAM-Tabelle	J
K	KNOW RAM-Test	KAADR, EADR
L	LIST-Zuweisung	LT oder LL oder LU
M	MOVE	MAADR, EADR, ZADR
O	OUTPUT-Kanal zuweisen	OT oder OL oder OU
P	PRINT Disassembler	PAADR, EADR
R	READ ext. Speicher	RADR
S	SUBSTITUTE	SADR,
T	TEXT Editor	T
V	VÉRIFY	VAADR, EADR, ZADR
W	WRITE	WAADR, EADR
X	Register anzeigen	X oder Xr
Z	Zweitregistersatz	Z oder Zr

Erläuterungen:

ADR	Adresse	AADR	Anfangsadresse
EADR	Endeadresse	ZADR	Zieladresse
r	Register A, B, ...		

Bei Eingabe von mehr als vier hexadezimalen Zeichen gelten nur die letzten vier. Man kann somit Eingabefehler durch Überschreiben berichtigen. Dieses Prinzip gilt bei allen Parametereingaben. Es sind nur die letzten HEXA-Zeichen gültig. Der Abschluß der Kommandos erfolgt mit der Taste Wagenrücklauf (CR).

Erst mit dem Druck auf diese Taste wird das Kommando ausgeführt. Bei Eingabe

nicht mit einem Kommando belegter Buchstaben oder anderer fehlerhafter Eingaben antwortet der Mikrocomputer als Fehlermeldung mit einem Fragezeichen und gelangt anschließend wieder in die Aufrufschleife.

2.3.6. RAM-Tabelle

Der Monitor benötigt für seine Aufgaben auch einige RAM-Speicherplätze. Beim Testlauf eines Programms unter Verwendung von Haltepunkten müssen die Adressen der Haltepunkte sowie deren Speicherinhalt und der Inhalt der Register des Mikroprozessors im RAM gespeichert werden. Durch Anzeige dieser Speicherplätze mit den Monitorkommandos X und Z kann man die richtige Arbeitsweise des zu testenden Programms kontrollieren.

Weitere Speicherplätze der RAM-Tabelle sind die Adressen für den jeweils aktuellen Ausgabeplatz auf den Bildschirm sowie für Peripheriegeräte des Nutzers, die den IN/OUT-Routinen zugewiesen werden können. Die aktuelle Zuweisung der Peripheriegeräte selbst steht ebenfalls in dieser Tabelle. Auch Steuerzeichen bzw. Betriebsweisen von Peripheriegeräten, Zeitkonstanten usw., die veränderlich sind, bringt man dort unter. Beim Start des Monitors wird die RAM-Tabelle in einen definierten Zustand versetzt, indem sie aus dem ROM übertragen wird. So ist der Platz des Registers A mit AA, B mit BB usw. belegt, so daß die im zu testenden Programm festgelegten Veränderungen (z. B. LD A,B) beim Haltepunkt erkennbar sind.

2.4. Programmbeispiele in Maschinensprache

2.4.1. Bedienung des Monitors

Bevor konkrete Beispiele folgen, soll zunächst die Bedienung des Monitors näher erläutert werden. Als besonders gut verständliche Zielstellung mögen wieder die Erstellung und der Testlauf eines Programms für eine Zeitverzögerung dienen. Entsprechend dem Programmablaufplan von Bild 1.1 wol-

2. Programmierung

len wir zunächst die innere Schleife aufbauen. Wir laden das Register B mit der maximal möglichen Zahl und vermindern den Inhalt um den Wert 1, bis das Register B leer ist (Meldung durch Z-Flag). In der Mnemonik geschrieben, ergibt sich:

```
LD      B,0FFH
M1:    DEC      B
      JPNZ     M1
```

Dabei stellt die Marke M1 die symbolische Form der Adresse des Decrementbefehls dar. Da wir Startadresse und Länge der Befehle bisher nicht wissen, können wir die Sprungadresse noch nicht anders angeben. Das vollständige Unterprogramm sieht dann wie folgt aus:

```
LD      A,0FFH
M2:    LD      B,0FFH
M1:    DEC      B
      JPNZ     M1
      DEC      A
      JPNZ     M2
      RET
```

Wir legen die Startadresse an den Beginn des RAM-Bereiches im Grundgerät, entnehmen die hexadezimale Codierung den Tabellen und stellen alles noch einmal übersichtlich dar:

Adressen	Daten	Mnemonik
0C00	3E	LD A,0FFH
0C01	FF	
0C02	06	LD B,0FFH
0C03	FF	
0C04	05	DEC B
0C05	C2	JPNZ 0C04
0C06	04	
0C07	0C	
0C08	3D	DEC A
0C09	C2	JPNZ 0C02
0C0A	02	
0C0B	0C	
0C0C	C9	RET

Die Eingabe des Programms in den Computer erfolgt mit dem Kommando S (Substitution). Nach der Eingabe der Zeichenfolge S0C00 und Leertaste gibt der Computer den augenblicklichen Speicherinhalt und danach einen Bindestrich aus. Nach der Eingabe von 3E und Leertaste überschreibt er damit den alten Speicherinhalt der Adresse 0C00H und zeigt den Inhalt der nächsten auf dem

Bildschirm an. Nach Eintippen des ganzen Programms wird mit CR abgeschlossen. Mit dem P-Kommando kontrollieren wir es noch einmal auf Fehler. Der Disassembler wandelt den Speicherinhalt wieder in die Mnemonik um. Die Eingabe lautet P0C00, 0C0C (CR). Wenn alles richtig eingetippt wurde, kann mit dem GO-Kommando der Testlauf beginnen. Am Ende des Programms befindet sich ein RET-Befehl. Es steht jedoch keine Rückkehradresse im Keller. Deshalb setzen wir an die Stelle des RET einen Haltepunkt im Testlauf: G0C00, 0C0C (CR). Nach der Verzögerungszeit unserer Zeitschleife meldet der Computer die Adresse des Haltepunkts. Mit dem Kommando X (CR) sehen wir uns die Register an und stellen fest, daß die Register A und B nicht mehr den bei der Initialisierung eingetragenen Wert besitzen, sondern auf 00 stehen. Somit ist die ordnungsgemäße Funktion des Programms erwiesen. Es ist jedoch noch verbesserungswürdig, denn die «Speicherplatzverschwendung» bei den Sprungbefehlen stört. Mit relativen Sprüngen, bei denen nur der Abstand des Ziels vom Absprungort als Argument und keine vollständige Adresse erforderlich ist, kann man jeweils 1 Byte einsparen. Außerdem läuft das Programm dann adressenunabhängig. Jetzt enthält es beim 1. Sprungbefehl die konkrete Adresse 0C04. Würde das Programm auf der Adresse 0D00 im Speicher stehen, erfolgte der Sprung trotzdem auf 0C04, wo aber dann nichts Sinnvolles steht. Der relative Sprung 20FD bedeutet nur «3 Speicherplätze zurückspringen, wenn das Z-Flag nicht gesetzt ist», und das Programm läuft damit auf beliebigen Adressen.

Für die innere Schleife bietet sich der Befehl DJNZ an, der den Inhalt von Register B um 1 vermindert («dekrementiert») und einen Sprung um die Distanz e-2 von der Adresse des Befehls ausführt. Der Operand des Befehls ist also keine konkrete Adresse, sondern wird als Zahl bewertet. Um sowohl rückwärts als auch vorwärts im Programm springen zu können, dient das Bit 7 dieses Bytes als Vorzeichen. Nicht gesetzt, bedeutet positive Zahl und damit Sprung vorwärts. Daraus ergibt sich eine maximale Sprungdistanz von +127 bis -128 Speicherplätzen.

Im Beispiel soll der Sprung zum DJNZ-Befehl erfolgen, also um -2 . Damit ergibt sich der hexadezimale Wert FE als Operand. Das Argument des 2. relativen Sprunges berechnen wir mit Hilfe der hexadezimalen Arithmetik des Monitors. Entsprechend dem allgemeinen Ausdruck

H Ziel, Quelle + 2

erhalten wir im Beispiel den Wert F9H. Die Summe der Adressen, die das H-Kommando liefert, benötigen wir bei Vorwärtssprüngen. Es ergibt sich nun:

Adresse	Daten	Mnemonik
0C00	3E	LD A, 0FFH
0C01	FF	
0C02	06	LD B, 0FFH
0C03	FF	
0C04	10	DJNZ 0C04
0C05	FE	
0C06	3D	DEC A
0C07	20	JR NZ 0C02
0C08	F9	
0C09	C9	RET

Mit der Anweisung M0C00, C09, D00 verschiebt man das Programm auf die Adressen D00 bis D0A und kann es, wie bereits beschrieben, mit dem GO-Kommando prüfen. Mit der Eingabe OT weist man als Ausgabekanal den Bildschirm zu und kann sich nach Eingabe der Folge WD00,D09 (CR) ansehen, wie der Computer dieses Programm auf einen externen Speicher (z. B. Kassettenrecorder) ausgibt. Diese Form ist für langsame externe Speicher international üblich und soll deshalb ausführlicher erläutert werden. Die Zeile

```
:0A0D00003EFF06FF10FE3D20F9C97A
```

läßt nur schwer unser Programm erkennen. Der Doppelpunkt ist das Kennzeichen für den Beginn eines Datenblocks. Falls man beim Einlesen mitten in einen Block schaltet oder wenn ein anderer Fehler auftritt, kann der nächste Datenblock daran erkannt werden. Sonst treten ja nur hexadezimale Zahlen auf. Die nächsten beiden Ziffern geben die Blocklänge an. Bei Ausgabe eines größeren Speicherbereichs (siehe Anhang) haben die Blöcke eine «Nutzlänge» von 10H (16dezimal) Bytes. Üblich sind auch Blocklängen von 80H, jedoch lassen sie sich schlechter auf Bildschirm oder Drucker dar-

stellen. Die folgenden 4 hexadezimalen Ziffern bezeichnen die Anfangsadresse des Datenblocks, im Beispiel 0D00. Es folgen 2 Bytes, die das vorliegende Monitorprogramm nicht auswertet. Bei größeren Programmsystemen dienen sie mitunter zur Bezeichnung von Plätzen im Datenblock, auf denen Adressen stehen.

Nach den Datenbytes unseres Programms folgen am Schluß 2 hexadezimale Ziffern zur Kontrolle. Sie errechnen sich durch Addition aller Daten des Blocks ohne Übertrag. Beim Einlesen berechnet der Computer diese Prüfsumme erneut. Nur wenn sie mit der gelesenen Summe übereinstimmt, erkennt er den Datenblock als fehlerfrei an und überträgt ihn aus dem Zwischenspeicher des Einleseprogramms auf die zugehörige Adresse. Ist als Eingabekanal kein externes Gerät zugewiesen, so können in dieser Form mit dem R-Kommando des Monitors auch Programme (z. B. aus dem Anhang des Buches) über die Tastatur eingegeben werden. Gegenüber dem S-Kommando besteht eine höhere Sicherheit gegen Tippfehler. Das R-Kommando erwartet als Abschluß einen Datenblock, der die Startadresse des Programms angibt. In unserem Beispiel wird er im Anschluß an das W-Kommando mit E0D00(CR) erzeugt. Das Einleseprogramm erlaubt es auch, Datenblöcke mit einem Adressenversatz einzulesen. Das Programm läuft zwar in der Regel nur auf den Adressen, für die es geschrieben ist, jedoch kann man die Verschiebung für die Programmierung von EPROMs nutzen. Soll ein EPROM später auf Adresse 8000H laufen, wie z. B. der BASIC-Interpreter, so kann dieses Programm zunächst mit R9000(CR) in einen RAM-Bereich ab Adresse 1000H eingegeben werden, von wo man es auf das EPROM programmiert. Nach dem gerade beschriebenen Muster kann man mit dem Grundgerät Programme erstellen und betreiben. Trotz der Ausführlichkeit der letzten Darstellung wurden längst nicht alle Möglichkeiten der Arbeit mit dem Monitor und dem Grundgerät gezeigt. Der Autor empfiehlt jedem Amateur, zunächst alle Monitorkommandos und danach alle Befehle des Mikroprozessors in Maschinensprache «durchzuspielen».

2.4.2. Unterprogrammtechnik

Die Zeitschleife ist ein universelles Unterprogramm. Nachdem an ihrem Beispiel die Arbeit mit dem Monitor erläutert wurde, sollen noch einige allgemeine Erläuterungen zu Unterprogrammen folgen. Dabei erscheint es sinnvoll, zunächst nochmals auf die Aufgabe des «Kellerzeigers» einzugehen. Bei der Konzeption eines Programms in Maschinensprache muß man im Gegensatz zu BASIC auch die Aufteilung des Speichers festlegen. Die Startadresse des Programms wird festgelegt, sein Ende ergibt sich nach der Umwandlung der Mnemonik in Maschinencode. Ein Teil des RAM-Bereiches im Speicher muß für die Aufnahme von Rückkehradressen reserviert bleiben, sonst ist kein Aufruf von Unterprogrammen möglich. Deshalb erhält der Prozessor bereits mit den ersten Befehlen des Monitorprogramms eine Information darüber, wo sich dieser Speicher im Adreßraum befindet. Dies geschieht durch Laden des Stackpointers (Kellerzeiger) auf die oberste Adresse dieses Speicherbereiches, denn im Gegensatz zur sonstigen Arbeit des Prozessors benutzt er den Stack von oben nach unten. Der Keller des Monitors beginnt auf Adresse 0FCEH (Zeilen 9, 88 in Programm 5.2.1.).

Nach dem Aufruf des ersten Unterprogramms CALL BLOE (Zeile 75) steht auf dem obersten Kellerplatz (0FCEH) der H-Teil des danach folgenden Befehls LD HL,RUFZ mit «00», darunter der L-Teil «23» auf Adresse 0FCDH. Der Stackpointer «zeigt» auf die darunter liegende freie Adresse 0FCCH, und der Prozessor arbeitet mit der durch den CALL in den Programmzähler geladenen Adresse 027DH weiter, wie oben geschildert. Tritt ein RET-Befehl auf, holt der Prozessor die oberhalb des Stackinhaltes stehende Adresse (zunächst L-Teil und danach H-Teil) und lädt damit den Programmzähler PC. Der Stackpointer zeigt danach wieder auf die nun um 2 Plätze erhöhte Adresse. Der Stack ist also ein Speicherbereich, bei dem der Zugriff nach dem erstmaligen Einstellen des Zeigers in Abhängigkeit von der «Vorgeschichte» erfolgt. Was beim Schreiben zuletzt in den Speicher kam, wird

beim Lesen zuerst wieder entnommen. Besonders gut kann man diese Vorgänge mit Hilfe der im Bild 1.14 angegebenen Schaltung im Schrittbetrieb verfolgen. Nach einem Rückkehrbefehl setzt der Prozessor die Abarbeitung des Programms mit der aus dem Stack in den Programmzähler geholten Adresse fort.

Die PUSH- und POP-Befehle beziehen sich ebenfalls auf den Stack. Sie laden den Inhalt eines Registerpaares (siehe Befehlsliste) und verstellen den Stackpointer wie ein Unterprogrammaufruf. Im Gegensatz dazu läuft das Programm jedoch mit der auf den PUSH-Befehl folgenden Adresse weiter, der Inhalt des Programmzählers wird also um 1 oder 2 erhöht wie bei anderen 1- oder 2-Byte-Befehlen auch.

Der Inhalt des in den Stack gebrachten Registerpaares bleibt zwar erhalten, kann aber nun durch ein Programm überschrieben werden, denn im Stack ist er ja «gerettet».

Mit dem POP-Befehl kann man ein Registerpaar wieder mit dem Inhalt der letzten belegten Stackadresse laden, der Stackpointer erhöht sich um 2 Plätze und zeigt auf die jetzt wieder freien Plätze.

In universellen Unterprogrammen sollte der Inhalt aller benutzten und damit Veränderungen unterworfenen Register zu Beginn in den Stack gerettet und vor dem Rücksprungbefehl wiederhergestellt werden, die Programme sind sonst wesentlich schwieriger zu beherrschen. Selbstverständlich müssen die den Stack beeinflussenden Befehle immer paarig auftreten, um einen Überlauf zu vermeiden. Insbesondere beim Test kann es vorkommen, daß der durch einen Fehler anwachsende Stack im RAM das zu testende Programm überschreibt.

Im Gegensatz zu BASIC erlaubt die Maschinensprache auch Rücksprungbefehle, deren Ausführung von verschiedenen Bedingungen abhängt. Ein Unterprogramm kann daher mehrere Rücksprungstellen aufweisen. In solchen Fällen ist die Rettung der Registerinhalte nur im übergeordneten Programm möglich. Das Kommando G des Monitorprogramms ordnet den zu testenden Programmen einen eigenen Speicherbereich als Stack zu (Zeilen 72–74), der ab Adresse

OECEH beginnt. Zur Kontrolle kann man sich die Speicherplätze des «Kellers» nach der Ausführung von Befehlen mit dem Kommando D (Dump, Display) ansehen.

2.4.3. Unterprogramme aus dem Monitor

Der Monitor ist ein Lehrbeispiel für die Programmierung in Maschinensprache. Nicht nur die Unterprogramme für die Peripherie, auch viele weitere Routinen enthalten grundsätzliche Lösungen für andere Programme. Deshalb sollen einige Routinen näher erläutert werden.

Tabellen

Eine wichtige, oft benötigte Aufgabe ist die Arbeit mit Tabellen. Ein Beispiel bietet die Verzweigung in Abhängigkeit vom eingegebenen Kommando (Zeilen 134 bis 216 des Monitorprogramms in Abschnitt 5.2.). Das Programm besteht aus der Routine FCGA und der zugehörigen Tabelle. In der Aufrufschleife erfolgt zunächst das Laden der Tabellenadresse in das Registerpaar HL und des 1. Zeichens des Kommandos in Register A. Die Routine FCGA vergleicht das Kommandozeichen mit dem 1. in der Tabelle gespeicherten Zeichen, übernimmt die zugehörige Adresse in das Registerpaar HL, wenn Übereinstimmung vorliegt, und verläßt die Tabelle. Anderenfalls überspringt das Programm die nachfolgenden Adreßbytes und prüft das nächste Zeichen. Das Ende der Tabelle ist gekennzeichnet. Sofern kein Zeichen der Tabelle mit dem eingegebenen übereinstimmt, wird als Kennzeichen das Carryflag gesetzt und die Routine verlassen. Die Routine FCGA kann man in Verbindung mit einer anderen Tabelle auch für andere Verzweigungen in Abhängigkeit eines Zeichens im Register A anwenden.

Bildschirmausgabe

Die Zeilen 477 bis 583 enthalten das Programm der Bildschirmausgabe CO, das sich im wesentlichen aus 2 Teilen zusammensetzt:

- Die Ausgabe von Zeichen auf den Bildschirmspeicher und dessen Verwaltung, d. h. das Abspeichern des nächsten zu beschreibenden Platzes; Prüfung auf Füllstand; ggf. Freimachen einer Zeile durch Bildschirmrollen.

- Die Auswertung von Steuerzeichen.

Das Programm erwartet die auszugebenden Zeichen im ASCII-Format, der Zeichenvorrat basiert auf sieben Bits. Von den dadurch verfügbaren 128 Zeichen dienen 32 als Steuerzeichen. Die Bildschirmausgabe erkennt einige dieser Zeichen und wertet sie aus, ohne sie auf den Bildschirm zu bringen. Diese Auswertung erfolgt unter Verwendung der Routine FCGA und der Tabelle COTAB, die einige der wichtigsten Steuerzeichen enthält. Alle anderen Zeichen werden nicht ausgewertet. Die Verwaltung des Bildschirms bezieht sich zunächst auf den aktuellen Speicherplatz, den der sogenannte Cursor anzeigt und auf den das nächste Zeichen bei der Ausgabe gelangt. Die Hardware der Bildschirmsteuerung bildet ihn aus dem gesetzten Bit 7. Eine normale Zeichenausgabe verschiebt den Cursor um einen Platz, der Zähler in der RAM-Tabelle wird um 1 erhöht. Hat der Zähler das Ende des Bildschirm-RAM erreicht, so wird der Inhalt des Bildschirms zurückgelesen, um eine Zeile (hier 64 Bildschirmplätze) verschoben, die letzte Bildschirmzeile gelöscht, d. h. mit Leerzeichen beschrieben, und der Cursorzähler auf den Anfang der letzten Zeile eingestellt.

Die Auswertung der implementierten Steuerzeichen beeinflusst den Cursor, ohne das auf dem Cursorplatz stehende Zeichen zu verändern.

Backstep	Zähler – 1
Linefeed	Zähler + 64
Horizontal-tabulator	Zähler + 1
Vertikal-tabulator	Zähler – 64
Wagenrücklauf	Zähler auf Zeilenanfang
New line	Zähler auf Zeilenanfang nächste Zeile

Tastatureingabe

Die Zeilen 305 bis 472 enthalten das Programm für die Eingabe von Zeichen über die Tastatur des Grundgeräts. In der Schaltung von Bild 1.6 arbeiten die Bits 1, 2 und 3 des PIO-Port A auf einen Dekoder, um noch einige Leitungen dieses Ports für die serielle Schnittstelle und den Tonausgang zu gewinnen. Port B ist in Betriebsart 3 auf Eingabe programmiert. Die Tasten liegen in Matrix-Form (Spalten und Zeilen) zwischen den 8 Dekoderausgängen und den Eingängen des Ports B. Bei zyklischer Ansteuerung schaltet jeweils ein Dekoderausgang auf 0 V, während die Eingänge B0 bis B7 über Widerstände an +5 V liegen. Ist eine Taste geschlossen, liegt an einem Eingang 0 V, und der Prozessor kann die zugehörige Taste durch eine Konvertierungstabelle bestimmen. Das eigentliche Tastaturprogramm belegt die Zeilen 305 bis 342. Die Zeilen 344 bis 369 enthalten die zyklische Abfrage. Die Umwandlung der Zeilen- und Spaltenzähler in die Form der ASCII-Zeichen steht in den Zeilen 371 bis 415, die restlichen Speicherplätze enthalten die Tabelle der Zeichen.

Die Anordnung der Tasten in der Matrixschaltung, d. h. die Zuordnung eines bestimmten Zeichens zu einer definierten Taste, bestimmt den Tabellenplatz des Zeichens. Daraus ergibt sich, daß der Anschluß der Tasten auch völlig anders erfolgen kann. Man muß bei einer anderen Schaltung als in Bild 1.6 lediglich die Tabelle ändern. Durch Zeitschleifen (Zeilen 121 bis 131) und die Ausgabe eines Pieptones (Zeilen 717 bis 738) geschieht die Entprellung der Tasten. Mit den Tasten Shift und Control kann man den Tasten je eine weitere Bedeutung zuordnen, was natürlich auch mit dem Einlesen eines anderen Tastaturprogramms denkbar ist.

2.5. Programmierung mit Assembler

Die in Abschnitt 2.4. ausführlich beschriebene Programmierung in Maschinensprache ist im Vergleich zu BASIC recht mühsam und fehlerträchtig. Die schematische Über-

setzung der Mnemonik in den Maschinencode, insbesondere die Berechnung von Argumenten der relativen Sprungbefehle, kann jedoch auch der Computer mit einem speziellen Programm, dem Assembler, ausführen. Der Assembler erwartet die Eingabe eines Programms als sogenannten Quelltext in der Mnemonik, wobei Sprungziele und Operanden symbolisch mit Marken oder Namen bezeichnet werden können. Er liefert daraus den Maschinencode für den Computer und für den Programmierer eine Liste, in der Quelle und Maschinencode übersichtlich zusammen enthalten sind.

Da der Assembler das Quellprogramm schematisch übersetzt, muß man bei der Schreibweise der Mnemonik sehr korrekt vorgehen. Bedingt durch die stürmische Entwicklung der Mikrocomputertechnik und andere historische Voraussetzungen, gibt es keine einheitliche Festlegung für die Mnemonik der Befehle.

Heute sind in der DDR drei unterschiedliche Typen für den Prozessor *UB 880 D* üblich, die sich jedoch nur bei einigen Befehlen unterscheiden. International am weitesten verbreitet ist die Schreibweise, die in den Tabellen für die Programmierung in Maschinensprache enthalten ist. In der Literatur findet man jedoch auch häufig eine Mnemonik, die vom Prozessor *8080* übernommen wurde. Steht in einem Quellprogramm anstelle von LD A,B der entsprechende Befehl MOV A,B, so «verstehen» der Assembler nur eine Schreibweise, die andere bewertet er als Fehler. Der im Anhang aufgeführte Assembler arbeitet mit der ROBOTRON-Mnemonik. Die korrekte Schreibweise der Befehle kann man in den Programmlisten im Anhang nachlesen.

2.5.1. Editor

Die Erstellung des Quellprogramms erfolgt ebenfalls mit einem Hilfsprogramm. Mit dem Editor können Texte über die Tastatur oder von einem externen Gerät in den Speicher des Computers eingegeben und komfortabel bearbeitet werden. Man kann Zeichen und sogar ganze Zeilen einfügen oder strei-

chen, und der übrige Text verschiebt sich entsprechend im Speicher. In Verbindung mit einem Drucker macht der Editor den Computer zur komfortablen Schreibmaschine. Der im Abschnitt 5.2.6. aufgeführte Editor entstand nicht nur, um damit Programme zu erstellen, sondern diente auch zum Schreiben dieses Buches. Der relativ geringe Umfang des Programms bedingt natürlich auch Grenzen der Leistungsfähigkeit und konnte nur durch direkten Zugriff auf Unterprogramme aus dem Monitor realisiert werden. Trotzdem ist der Effekt erstaunlich. Einige Einzelheiten dienen auch der Zusammenarbeit mit anderen vorhandenen Programmen. Die meisten Kommandos zur Bedienung des Editors (Tabelle 2.20) entsprechen den vergleichbaren des Monitors; nur sind die Argumente hexadezimale Zeilennummern und lediglich bei der Textpufferzuweisung Adressen im Speicher. Erfolgt keine ausdrückliche Pufferzuweisung, so legt der Editor den Text im Adreßbereich 4000H bis 7FFFFH ab. Mit dem Kommando Q kann man den Editor verlassen und mit dem Monitor oder über diesen mit dem Assembler weiterarbeiten. Stellt sich beim Testlauf des Programms ein Fehler heraus, geht man in den Editor zurück und bearbeitet den Quelltext erneut. Beim Aufruf des Editors erfolgt deshalb zunächst eine Abfrage, ob ein neuer Text bearbeitet werden soll oder ob sich bereits etwas im Textpuffer befindet. Der Buchstabe Y bewirkt die Neuzuweisung des Textpuffers, alle anderen Zeichen stellen den Zeilenzähler nur auf Null und erlauben weiterhin den Zugriff auf den Text im Puffer.

Das Kommando E ist das Kernstück des Editors, es erlaubt Eingabe und Bearbeitung von Text über die Tastatur. Um den Computer auch für Schreibarbeiten nutzen zu können, erfolgt dabei eine Umschaltung des Zeichenvorrates. Die großen Buchstaben erscheinen nur noch mit der Taste Shift. Die letzte Bildschirmzeile dient der Bearbeitung. Alle Zeichen, die sich in dieser Zeile links vom Cursor befinden, gelangen nach Betätigung der Tasten CR, LF oder Control Z in den Speicher des Textpuffers.

Tabelle 2.20. Editor-Kommandos

Kommando	Funktion	Syntax, Parameter
A(sign)	Textspeicher-Zuweisung	A nnnn (CR) nnnn (CR)
D(isplay)	Darstellen einer Anzahl von n Zeilen mit Weiterstellen des Zeilenzählers	Dn (CR) oder D(Leerz.)...(CR)
E(dit)	Texteingabe-Modus	En (CR)
G(o)	Weiterstellen des Zeilenzählers vorwärts um n Zeilen	Gn (CR)
I(nput)	Zuweisung Eingabekanal	IT, IL oder IU
L(ist)	Zuweisung Druckerkanal	LT, LL oder LU
O(utput)	Zuweisung Ausgabekanal	OT, OL oder OU
P(rint)	Ausgabe aller Zeilen auf Listkanal	P
Q(uit)	Rücksprung in den Monitor	Q
R(ead)	Lesen des Eingabekanals bis zum ETX-Zeichen	R
T(op)	Rückstellen des Zeilenzählers an den Anfang	T
U(p)	Rückstellen des Zeilenzählers um n Zeilen	Un (CR)
W(rite)	Ausgabe einer Anzahl oder aller Zeilen	Wn (CR) oder W (CR)

Erläuterungen:

n: Hexa-Zahl, max. vierstellig

(CR): Wagenrücklauf-Taste

Tabelle 2.22. Editor-Fehlermeldungen

E1	Kommandofehler
E2	Textpuffer voll
E3	Parameterfehler

2. Programmierung

Tabelle 2.21. Editor-Steuerzeichen

Taste	Funktion
BS	Cursor nach links, Korrektur durch Überschreiben möglich
DEL	Inhalt der Anzeigezeile löschen
LF	Zeile in Textpuffer übernehmen, nächste Zeile aus dem Puffer anzeigen
CR	Zeile in Puffer übernehmen, E-Kommando verlassen
Control D	vorletzte Zeile in Arbeitszeile dupeln
Control Z	Zeile in Puffer übernehmen, neue Zeile beginnen
Control X	Zeichen über dem Cursor löschen, Rest nach links verschieben
Control S	Zeichen über dem Cursor einfügen, Rest nach rechts verschieben

Tabelle 2.21. zeigt die Steuerzeichen des Editors, Bild 2.8 die Arbeitsweise als Programmablaufplan. Tabelle 2.22. erläutert die Fehlermeldungen.

2.5.2. Assemblerbeschreibung

Gegenüber der manuellen Übersetzung eines Programms mit Tabellen der hexadezimalen Kodierung von Befehlen hat ein Assemblerprogramm zwei wesentliche Vorteile:
Die Übersetzung längerer Programme verläuft um mehrere Größenordnungen schneller und ist syntaktisch fehlerfrei. Das bedeutet aber nicht die Beseitigung logischer Fehler. Ist z. B. bei einem Sprung das Ziel im Quellprogramm falsch bezeichnet oder ein Flag falsch ausgewertet, dann erfolgt die

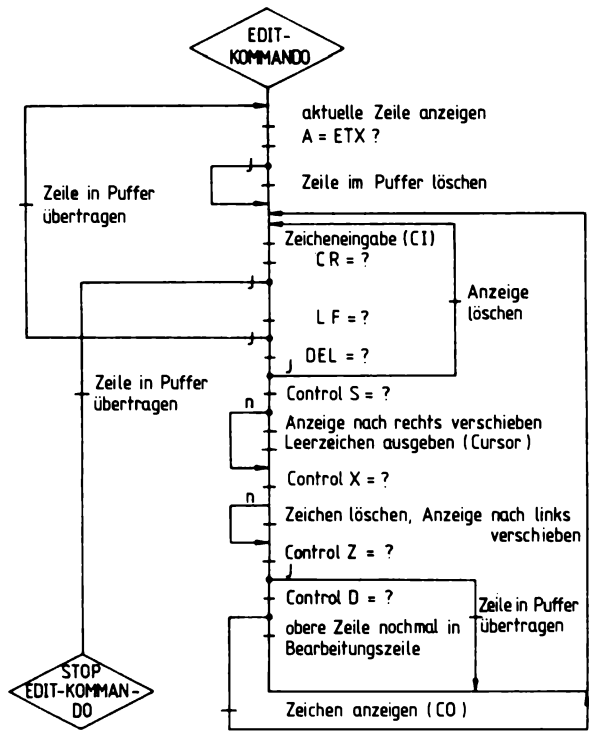


Bild 2.8
Arbeitsweise des Editors

Übersetzung zwar syntaktisch richtig als Sprung, aber logisch falsch beim Operanden. Das Programm kann nicht richtig arbeiten. Trotzdem lohnt sich der Aufwand für die Nutzung eines Assemblers. Sehr wertvoll ist auch die übersichtliche Darstellung der Programme, wie das Beispiel des Monitorprogramms im Anhang zeigt. Der Ausdruck ist in mehrere, unterschiedlich breite Spalten formatiert. Links befindet sich die Adresse, für die das Maschinenprogramm übersetzt wurde. Neben der üblichen hexadezimalen Angabe bieten einige Assembler sie noch in dezimaler Schreibweise. Damit kann der Programmierer besser Berechnungen ausführen oder für den Aufruf durch den BASIC-Interpreter ablesen, denn in dieser Programmiersprache muß man Adressen meist dezimal angeben.

Die nächste Spalte des Assemblerausdrucks enthält das Maschinenprogramm (Objektcode) in hexadezimaler Kodierung, wobei nicht mehr als ein Befehl pro Zeile steht. Entsprechend den unterschiedlichen Befehlslängen kann diese Spalte bis zu 4 zweistellige hexadezimale Zahlen enthalten.

Die immer besetzte Spalte mit der dezimalen Zeilennummer trennt den Maschinencode vom Quellprogramm. Nicht immer muß das Quellprogramm Befehle enthalten, auch Erläuterungen, die den Sinn von Programmteilen oder für den Eintritt in ein Programm erforderliche Parameter für eine spätere Bearbeitung festhalten, sind äußerst nützlich. Diese Kommentare können entweder hinter einem Befehl stehen oder schon unmittelbar in der Spalte nach der Zeilennummer beginnen. Sie sind durch ein vorangestelltes Semikolon gekennzeichnet. Unmittelbar auf die Zeilennummer folgt das Markenfeld. Wie bereits erwähnt, symbolisiert eine Marke den Wert einer Adresse. Meist kennzeichnet sie ein Sprungziel, manchmal auch den Beginn oder das Ende von Tabellen. Eine Marke besteht aus maximal 5 Zeichen, von denen das erste ein Buchstabe sein muß, und wird von einem nachfolgenden Doppelpunkt gekennzeichnet. Den konkreten Wert, den eine Marke symbolisiert, berechnet der Assembler erst bei der Übersetzung. Hier zeigt sich der Sinn

Tabelle 2.23. Pseudobefehle des Assemblers

Pseudo-befehl	Bedeutung
EQU	Wertzuweisung einmalig
DEF	Wertzuweisung mehrmalig
DB	Wertzuweisung für Bytes
DA	Wertzuweisung für Adressen
BER	Reservierung eines Speicherbereiches
ORG	Wertzuweisung für die Adresse des folgenden Programms
#	aktueller Adressenwert
PN	Programmname, Anfang eines Programms
END	Ende eines Programms

symbolischer Adressen und Operanden. Hätte man das Sprungziel gleich als Adresse angegeben, müßte bei Änderung möglicherweise an vielen Stellen ein neuer Wert stehen. So kann man etwas dazuschreiben oder wegstreichen, und der Assembler setzt an allen Stellen des Programms die neuen Sprungadressen ein. Nach dem Markenfeld folgt im Assemblerlisting die Mnemonik eines Befehls. Neben den auf den Prozessor bezogenen gibt es auch sogenannte Pseudobefehle, die nur die Arbeitsweise des Assemblers steuern und natürlich nicht in den Maschinencode zu übersetzen sind. Tabelle 2.23. zeigt die Pseudobefehle des Assemblers, im Monitorlisting kann man bei Unklarheiten den Gebrauch nachlesen.

Die auf den Befehl häufig folgenden Operanden sind trotz der symbolischen Bezeichnung oft konkrete Adressen oder andere Zahlen. Sie können sowohl in hexadezimaler Kodierung als auch dezimal dargestellt werden. Im Gegensatz zu den Marken und anderen symbolischen Werten muß ihr erstes Zeichen eine Ziffer sein. Bei hexadezimalen Zahlen, die mit einem Buchstaben beginnen, stellt man die Ziffer 0 davor.

2. Programmierung

2.5.3. Bedienung des Assemblers

Das vom Assembler zu bearbeitende Quellprogramm kann sowohl im RAM stehen als auch von einem externen Gerät über den Inputkanal in den Computer gelangen. Ist kein externes Gerät zugewiesen (IT), kann das Programm nur mit Hilfe des Editors in den RAM gelangt sein, und der Assembler erwartet es auf den damit festgelegten Textpuffer-Adressen. Da der Assembler etwa 8k RAM benötigt, können dann nur kurze Programme bei der in Bild 3.1 gezeigten Aufteilung des Speichers bearbeitet werden (maximal ca. 1 k Objektcode). Die Übersetzung längerer Programme ist zwar möglich, jedoch müssen dann 2 externe Geräte am Computer arbeiten: Der Inputkanal muß das Quellprogramm liefern, Out- und Listkanal müssen den Objektcode und die Liste aufnehmen und speichern bzw. drucken. Ist als Listgerät der Bildschirm zugewiesen (LT), erscheint die Liste mit je 16 Zeilen. Nach Eingabe eines Leerzeichens folgen die nächsten 16. Der Maschinencode gelangt nach der Übersetzung auf die im Quellprogramm durch ORG-Befehl festgelegte Adresse, wenn kein externes Gerät dem Outputkanal zugewiesen ist. Dabei muß man darauf achten, daß nicht der Textpuffer oder der Arbeitsspeicher des Assemblers (1000H bis 2FFFFH) überschrieben werden. Um sicherzugehen und auch für diese Adressbereiche Programme erstellen zu

Tabelle 2.24. Assembler-Fehlermeldungen

20	Fehlerhafter Programmname
21	Markenfehler (zu lang, Syntax)
22	Marke doppelt
24	Fehlerhafte Mnemonik
25	Operandenfehler
27	Zeichenkette zu lang
28	Symboltabelle voll
29	Relativsprung zu weit

können, sollte die Ausgabe des Objektcodes bei längeren Programmen immer auf Kassetten erfolgen.

Die Übersetzung des Quellprogramms erfolgt in mindestens 2 Schritten. Zunächst muß der Assembler alle Befehle auf ihre Länge prüfen und damit für symbolische Adressen konkrete Werte berechnen. Erst in einem zweiten Durchlauf können diese Werte dann als Operanden bei den Befehlen eingesetzt werden. Nach dem Aufruf des Assemblers muß deshalb zunächst Pass 1 ablaufen. Erst danach kann mit Pass O das Objektprogramm oder mit Pass P die Liste entstehen. Bei Eingabe von Kassette oder einem anderen externen Gerät muß man das Quellprogramm also mindestens zweimal «abspielen». Selbstverständlich entstehen bei der Programmierung mit Assembler trotz aller Sorgfalt oft Fehler.

Tabelle 2.24. erläutert die Bedeutung der Fehlermeldungen des Assemblers.

3. Erweiterungen

Wenn das Grundgerät mit Tastatur und Bildschirm korrekt arbeitet, kann man die Leistungsfähigkeit durch zusätzliche Elektronikleiterplatten und Programme beträchtlich steigern. Dabei bilden vor allem Speichererweiterungen die Voraussetzung für den Übergang zu höheren Programmiersprachen. Aber auch der Anschluß weiterer Peripheriebaugruppen kann interessante Möglichkeiten eröffnen. Hier hat der Amateur viele Mittel, um individuelle Anwendungsfälle für Modellsteuerung, Fotoarbeiten, Musikelektronik oder ähnliches mit modernster Technik anzugehen.

Jede Erweiterung der Elektronik stellt eine Belastung der Ausgangsleitungen des Prozessors dar. Sowohl der aus den angeschlossenen TTL-Schaltkreisen bei L-Pegel herausfließende Strom als auch die Kapazitäten von Leitungen und Eingängen anderer Schaltkreise dürfen nicht unbegrenzt anwachsen. Während der Ausgangsstrom auf eine Standard-TTL-Last (1,6 mA) festgelegt ist, hängt die Größe der zulässigen Kapazitäten an den Ausgängen vor allem von der Arbeitsgeschwindigkeit ab. Bei umfangreichen Mikrocomputer-Systemen verstärkt man deshalb die Signale auf den einzelnen Leiterplatten mit Pufferschaltkreisen. Diese Bauelemente arbeiten mit der besonders schnellen Schottky-Technologie, um möglichst geringe Signalverzögerungen zu verursachen. Neben dem Vorteil der höheren Funktionssicherheit bei größeren Systemen bringt ihr Einsatz folgende Nachteile:

- Der Stromverbrauch ist relativ hoch.
- Zur Steuerung der Übertragungsrichtung ist für Interruptbetrieb eine Logikschaltung erforderlich.

Aus diesen Gründen sollte der Amateur seine Schaltungen so dimensionieren, daß Buspuffer nur in Ausnahmefällen vorkommen.

3.1. Speicher

3.1.1. Schaltungen

Die wichtigste Erweiterung ist ein möglichst großer verfügbarer Speicher. Alle komfortableren Programme, wie z. B. BASIC-Interpreter, Editor und Assembler, benötigen wesentlich mehr Speicherplatz, als das Grundgerät mit der Bestückung nach Bild 1.1 aufweist. Dabei wird die Vergrößerung des RAM-Bereiches bei Einsatz dynamischer Speicher zwar billiger, aber das wiederholte Laden umfangreicher Programme von Kassette ist

Bildwiederholsp	F 000
	F C00
z. Zt. frei	
	E 000
ASSEMBLER (7K)	
EDITOR z. Zt. frei	C 400
	C 000
	B 000
BASIC (12K)	8000
dyn. RAM	
(16K)	
	4000
stat. RAM	
(12K)	
System-RAM	1000
Reassembler	C00
Debugger	800
In-/ Out-Routinen	400
	0

Bild 3.1
Speicheraufteilung

sehr umständlich. Deshalb erscheint es vorteilhaft, den Speicher nur zur Hälfte als RAM auszulegen und die oben genannten wichtigen Programme im EPROM zur schnelleren Verfügung zu speichern. Bild 3.1 zeigt die Speicheraufteilung. Es sei noch einmal daran erinnert, daß der Prozessor zwar maximal «nur» 64 kByte-Speicher direkt adressieren, jedoch mit den MEMDI-Signalen auch problemlos größere Speicherbaugruppen umschalten kann.

Der Aufbau von zusätzlichen Speicherleiterplatten ist entsprechend der im Abschnitt 1.5. erläuterten Konzeption besonders effektiv, weil hier fast alle Anschlüsse mit denen des Nachbarschaltkreises an gemeinsamen Leitungen liegen. Bild 3.2 zeigt ein Layout für Byte-organisierte Speicher (z. B. EPROM *U 555*). Ebenso wie bei der CPU-Leiterplatte können jedoch auch andere Schaltkreise mit ähnlicher Anschlußbelegung bestückt werden (Bild 3.3). Bei höherem Integrationsgrad der eingesetzten Schaltkreise muß der Dekoder für die CS-Signale nur eine geringe Zahl von Adressen auswerten. Jedoch sollte man immer auf vollständige Dekodierung aller Adreßleitungen des Prozessors achten, wie dies bei der CPU-Baugruppe bereits erläutert wurde.

Der Schaltkreis *U 224 (6514)* entspricht mit seiner Anschlußbelegung dem im Grundgerät eingesetzten *U 214*. Er wird jedoch in der stromsparenden CMOS-Technologie gefertigt. Der Leistungsbedarf ist so gering, daß eine bestückte Leiterplatte von einer kleinen Akkumulatorenbatterie gespeist werden kann. Die in diesen RAM gespeicherten Daten gehen dann nicht beim Abschalten der Spannung verloren. Man erhält so die Möglichkeit, Programme sicher gegen Netzausfälle zu erstellen, braucht sie nicht nach jedem Abschalten erneut einzulesen und kann sie zu einem anderen Amateur «tragen», mit dem man aufgrund unterschiedlicher Hardware keine Programme über Magnetbandkassetten tauschen kann. Bild 3.4 zeigt eine für die Bestückung mit *U 224* oder *U 214* (kein CMOS-RAM) geeignete Leiterplatte. Bild 3.5 enthält die Dekodierschaltung der Chip-select-Signale sowie die Stromversorgung mit Batterie. Die Widerstände an den Dekoderausgängen verhindern undefinierte Zustände bei der Umschaltung auf Batteriebetrieb. Mit dem 4fach-DIL-Schalter kann man die Anfangsadresse der Leiterplatte in 1-k-Bereichen umschalten. Die Schaltung dekodiert zwar Adressen vollständig, jedoch liegen die 1-k-

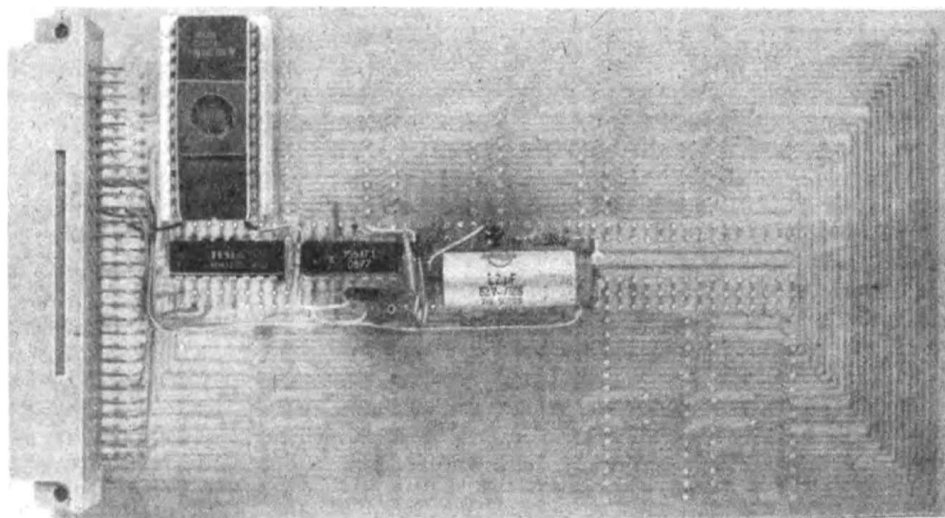


Bild 3.3a EPROM-Karte mit 64-k-Speicher

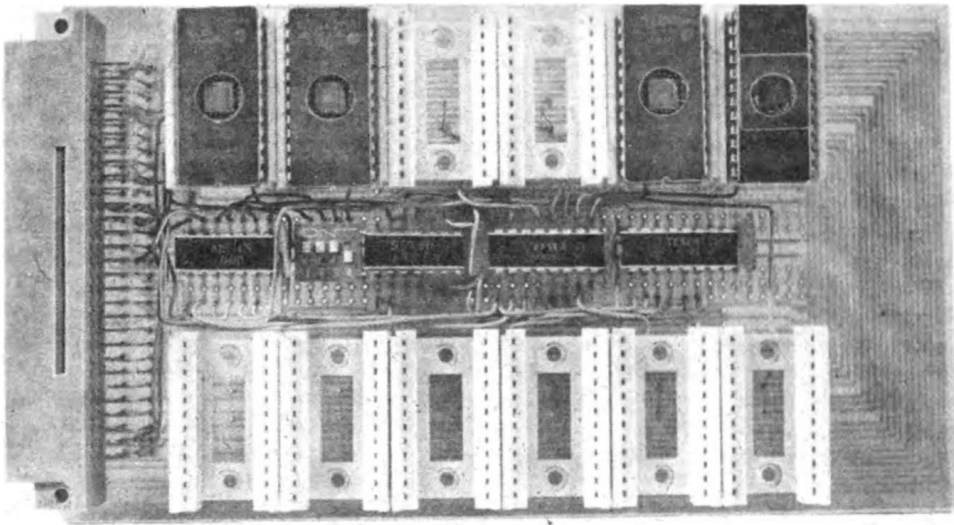


Bild 3.3b EPROM-Karte mit 32-k-Speichern

Bereiche auf der Leiterplatte nicht aneinander. Dadurch kann es bei einer Teilbestückung zu Lücken im Speicherbereich kommen, wenn ungerade Anfangsadressen eingestellt sind. Auch die Wertigkeit der beiden unteren Schalter ist gegenüber den anderen beiden negiert. Trotzdem kann die Schal-

tung auch für andere Speicherkarten bei 1-k-Bereichen empfohlen werden (z. B. MEM1 bei Bestückung mit U 555 C). Große RAM-Bereiche lassen sich günstig mit dynamisch arbeitenden Speichern (DRAM) aufbauen. Bei DRAM müssen ständig innerhalb kurzer Zeiten Speicherzugriffe erfolgen, andernfalls

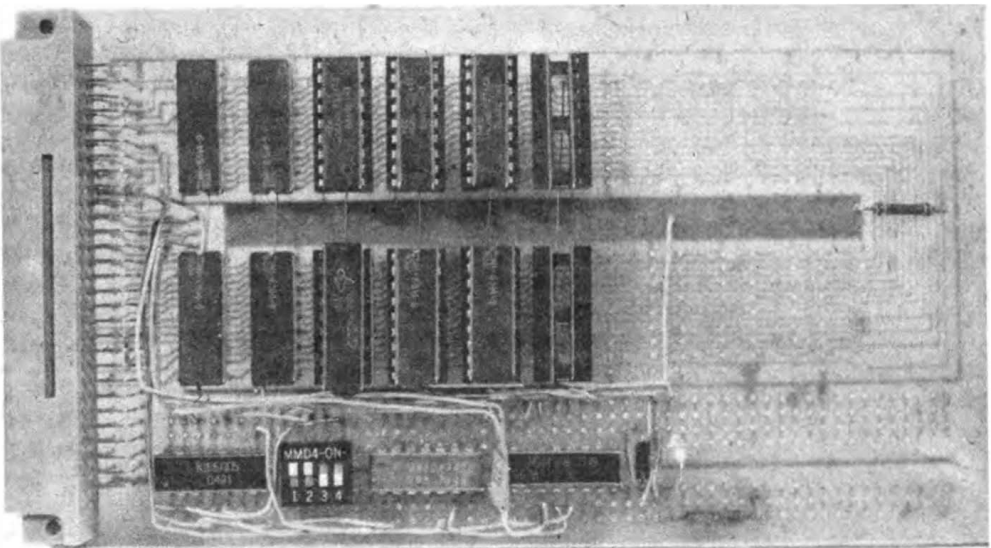


Bild 3.4b Speicherkarte MEM 2; bestückte Leiterplatte

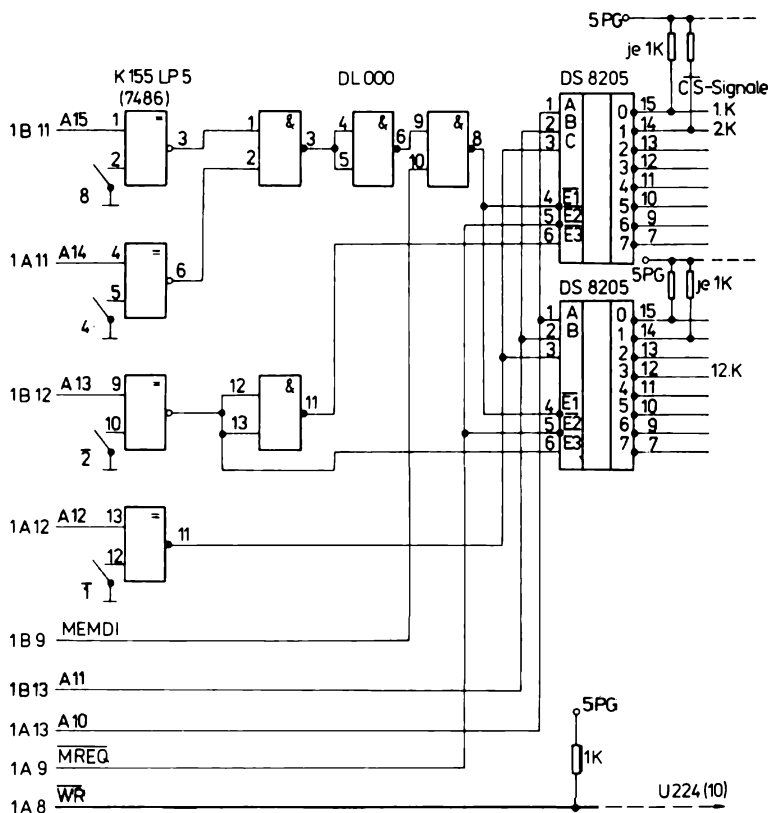


Bild 3.5
Schaltung und
Adreßdekodierung in
1-kByte-Bereichen
für MEM 1 und
MEM 2

geht der Dateninhalt verloren. Während die statischen RAM ihre Daten in Flip-Flop-Schaltungen solange halten, wie die Stromversorgung anliegt, benutzen dynamische RAM einen Kondensator, auf dem sie eine Ladung speichern. Diese Kondensatoren sind so klein, daß sie sich sowohl nach einigen Millisekunden über die Verlustwiderstände als auch beim Lesen entladen. Nach dem Lesen kann man jedoch die gespeicherte Information zurückladen, da sie nun bekannt ist. Dieses Auffrischen der Ladung heißt «Refresh». Die Steuerung dieses Vorganges kann sowohl im Speicherschaltkreis als auch mit einer externen Schaltung oder durch den Prozessor erfolgen. Der *UB 880 D* hat hierzu einen besonderen Ausgang. In den Zeitdiagrammen erkennt man das Prinzip des Refresh-Zyklus: Nach jedem Speicherzugriff, wenn der Prozessor den gelesenen Befehl dekodiert und ausführt, gibt er

aus dem Refresh-Register eine Adresse, die von den aktuellen Programmadressen unabhängig ist und die er nach jedem Refresh-Zyklus erhöht. Da hierbei die Speicherdaten nicht nach außen gelangen, können gleichzeitig viele Zellen aufgefrischt werden. Deshalb ist keine vollständige Adresse erforderlich, sondern man kommt mit den 7 Adreßleitungen aus. Auch ein mögliches Problem ist im Zeitdiagramm erkennbar: Bei aktivem WAIT verzögert sich die Bildung des Refresh-Signals entsprechend. Die Steuerung des Auffrischens dynamischer Speicher durch die CPU und der Einsatz von Baugruppen, die WAIT-Signale erzeugen, sind kaum «unter einen Hut» zu bringen, wenn auch in der Praxis die Daten meist noch nicht nach der vom Schaltkreis-Hersteller als Maximum genannten Zeit von 2 ms verlorengelassen sind.

Der Schaltkreis *U 256 (4116)* arbeitet mit

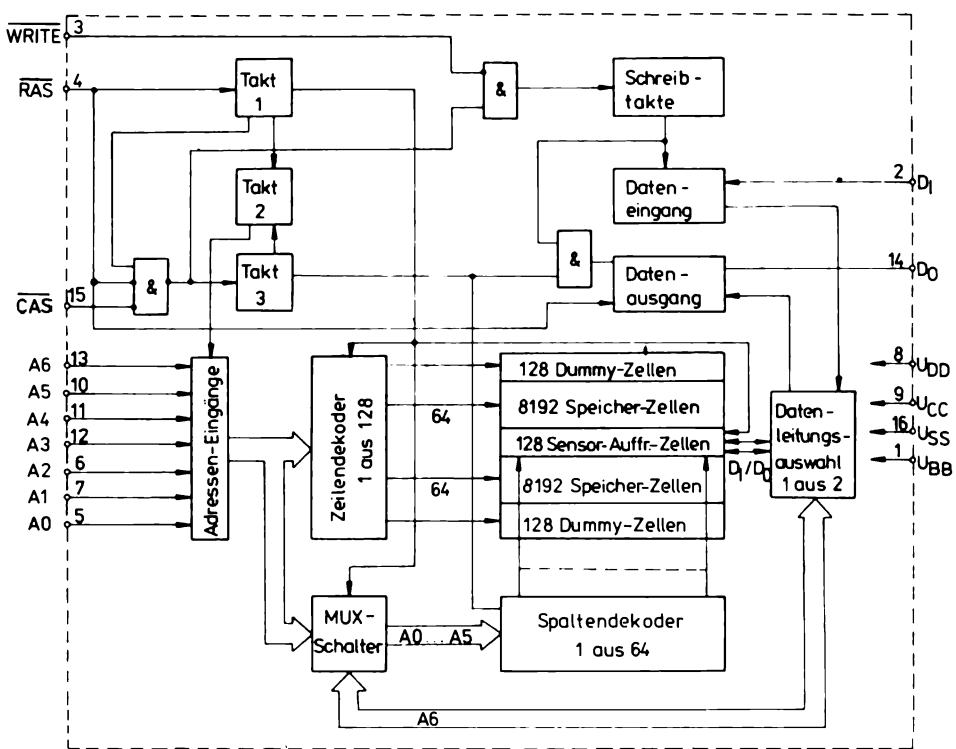


Bild 3.6 Übersichtsschaltplan U 256 D (4116)

der Organisation $16\,384 \times 1$ bit. Bild 3.6 zeigt das Übersichtsschaltbild des internen Aufbaus. Die Speicherplätze sind wieder in einer Matrix aus 2×64 Zeilen und 128 Spalten angeordnet. Die Zahl der Adressenleitungen des Schaltkreises beträgt jedoch nur 7, obwohl bei dieser Anzahl von Speicherzellen 14 erforderlich sind. Die Adressen gelangen in 2 Gruppen nacheinander in den Schaltkreis. Die Bauelemente haben dadurch trotz hoher Speicherkapazität kleine Abmessungen und einen günstigen Preis.

Die Umschaltung der Adressen erfolgt innerhalb der Schaltkreise und extern auf der Leiterplatte durch Multiplexer. Deren Steuerung geschieht mit den Signalen $\overline{\text{RAS}}$ (Row Address Strobe, Zeilenadressenabtastung) und $\overline{\text{CAS}}$ (Column Address Strobe, Spaltenadressenabtastung), die sich in einer Logikschaltung auf der Speicherleiterplatte aus den Signalen des Prozessors gewinnen las-

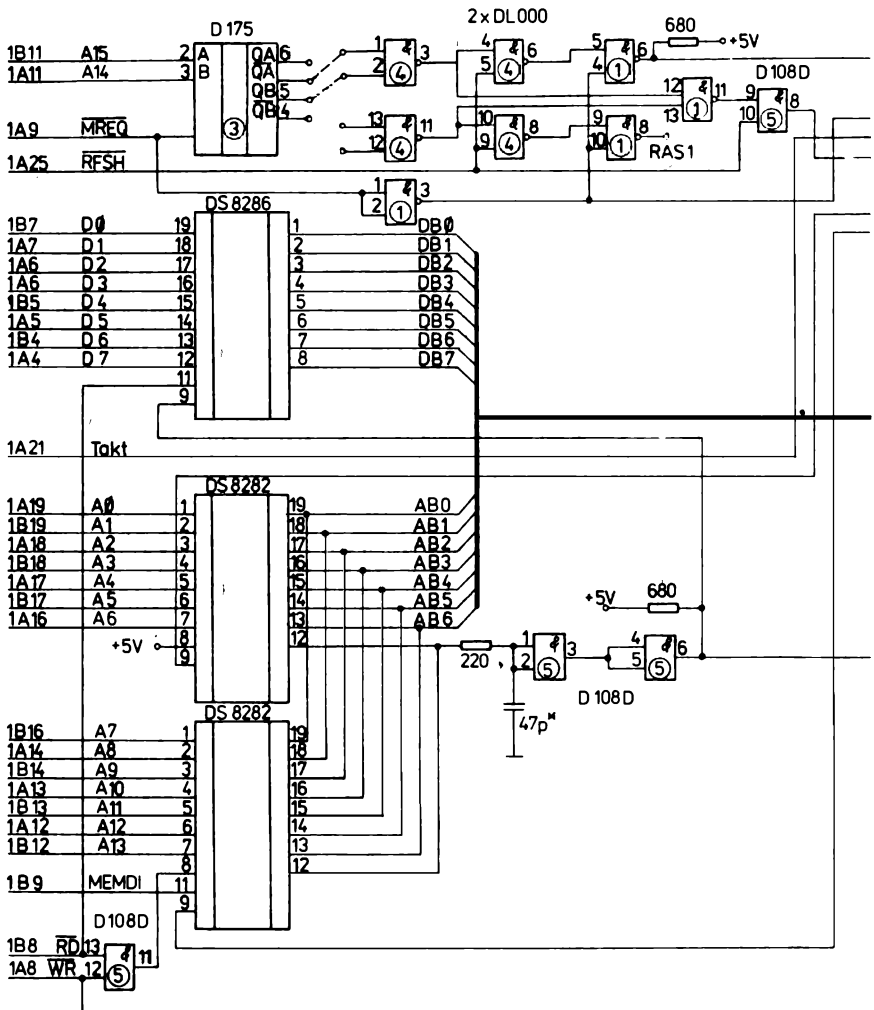
sen. Die Bilder 3.7 und 3.8 zeigen Layout und Schaltung einer 16-kByte-Speicherkarte mit dynamischen RAM U 256.

Wie man aus den Zeitdiagrammen für Lese- und Schreibzyklen (siehe Anhang) entnehmen kann, ist die Umschaltung der Multiplexer recht zeitkritisch. Da sich die Laufzeiten innerhalb der Schaltkreise auch auswirken, muß die Verzögerungsschaltung abgeglichen werden. Andere als die angegebenen Bauelemente (z. B. Low-Power-TTL) können in dieser Schaltung die Funktion in Frage stellen. Der Aufbau dieser Leiterplatte ist nur dem erfahrenen Amateur zu empfehlen.

Für den Amateur sind auch ältere Schaltkreise interessant, die häufig preisgünstig angeboten werden oder ausgesonderten Industrieerzeugnissen entstammen.

Der Schaltkreis U 202 D ist ein statischer RAM mit einer Kapazität von $1\,024 \times 1$ bit.

3. Erweiterungen



Im Gegensatz zu den bisher besprochenen verfügt dieser Typ über getrennte Anschlüsse für die Daten. Um ihn am bidirektionalen Datenbus des Mikrocomputers zu betreiben, muß man die Leitung DI (Stift 11, Dateneingang) mit Leitung D0 (Stift 12, Datenausgang) über einen Buspuffer DS 8216 entkoppelt an den Datenbus schalten.

3.1.2. Speicherumschaltung

Bei großen Speicheranordnungen oder zur mehrfachen Nutzung eines bestimmten Adressbereiches kann eine Speicherumschaltung nützlich sein. Aus anderen Geräten entnommene Programme (z. B. Schachprogramme) laufen ohne aufwendige Änderung nur auf den Originaladressen (oft ab Adresse 0) und nutzen häufig die Restartbefehle. Auch das weit verbreitete Diskettenbetriebssystem CP/M erfordert ab Adresse 0 einen Schreib-/Lesespeicher. Für dieses Be-

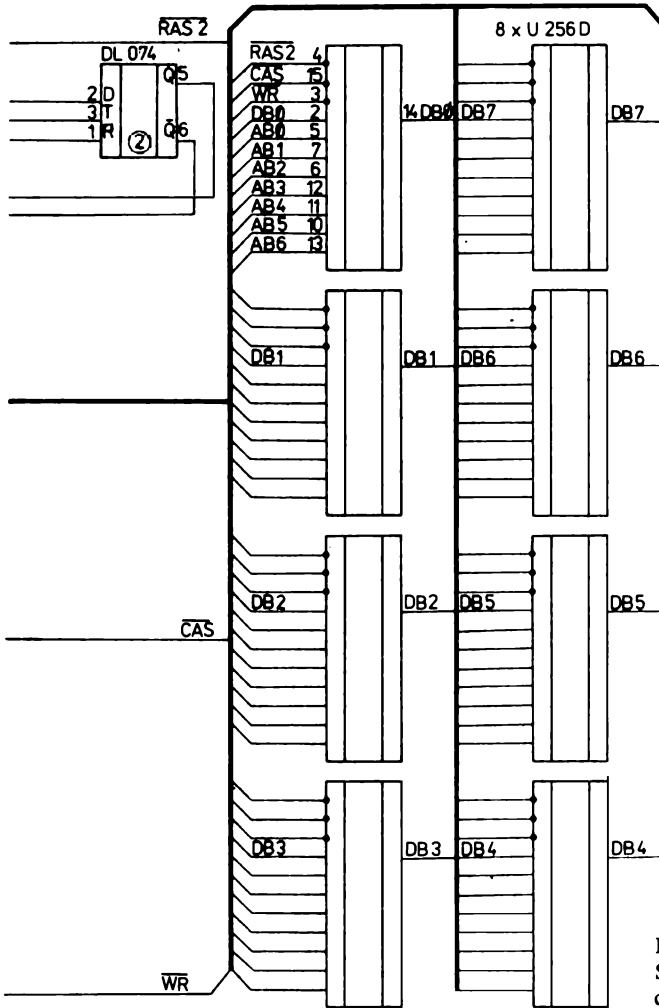


Bild 3.8
Schaltung eines dynamischen Speichers mit 16 kByte

triebssystem existiert ebenfalls eine Einsprungtabelle für die IN/OUT-Routinen. Anpassung und Nutzung von einfachen CP/M-Programmen sind im allgemeinen mit relativ geringem Aufwand möglich.

Der Monitor wird bei Speicherumschaltung auf der CPU-Baugruppe für die Arbeitsadressen übersetzt, die im oberen Speicherbereich liegen. Der Prozessor beginnt jedoch nach dem Einschalten oder nach dem Resetsignal sein Programm mit der Adresse 0. Man muß nun mit einer Schaltung (Bild 3.9) dafür sorgen, daß er zur Startadresse des Mo-

nitores findet. Sie schaltet den Dekoder für die Speicher der CPU-Baugruppe von Adresse 0 auf 0C000H um. Nach dem Einschalten entsteht zunächst ein Resetsignal, welches das RS-Flip-Flop aus den Gattern des Schaltkreises DL 003 rücksetzt. Dabei schaltet MEMDI auf L-Pegel und sperrt ein ebenfalls auf Adresse 0 beginnendes RAM. Die Adreßleitungen 14 und 15 gelangen nicht auf den Dekoder DS 8205.

Unabhängig von ihrem Signalpegel ist Anschluß 6 des Dekoders jedoch über das gesperrte, zweite NAND-Gatter DL 000 auf

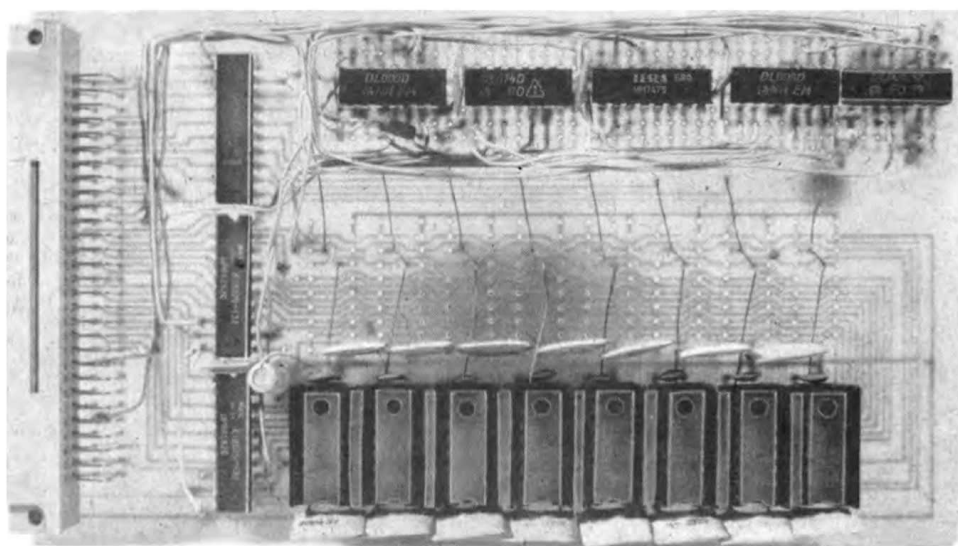


Bild 3.7b Speicherkarte MEM 3; bestückte Leiterplatte

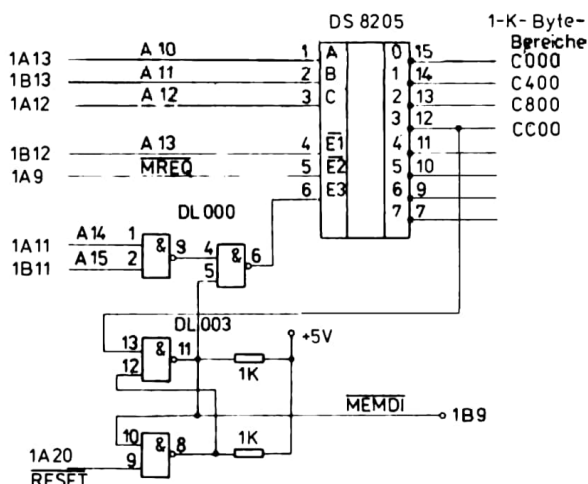


Bild 3.9
Anordnung zur Speicherumschaltung

Freigabe geschaltet. Als erster Befehl muß im Speicher ein Sprung auf Adresse 0C003H stehen. Die Adreßbits 14 und 15 schalten damit auf H-Pegel und würden den Dekoder nun auch freigeben, was aber über das Flip-Flop schon erfolgt. Die nächsten Befehle des Monitors enthalten die Übertragung der Systemtabellen in den RAM, der hier auf Adresse 0CC00H beginnt. Mit dem CS-Signal für diesen Speicher am Stift 12 des Dekoders schaltet das Flip-Flop um, gibt das

RAM ab Adresse 0 frei und schaltet die obersten Adressen auch auf den Dekoder des Grundgerätes durch. Die Schaltung kann auch auf eine noch höhere Adresse umgestellt werden, wenn man ein Gatter mit 3 oder 4 Eingängen benutzt, jedoch ist dann die Adresse des Bildschirmspeichers neu festzulegen. Mit Hilfe eines Ausgangskanals ist natürlich eine komfortable, programmgesteuerte Speicherumschaltung über die MEMDI-Signale jederzeit auch möglich.

3.2. EPROM-Programmierzusatz

Die Programmierung von EPROMs erfolgt je nach Typ mit einer Spannung von 22 bis 48 Volt. Auch die Signalfolge des Zeitdiagramms ist bei den einzelnen EPROMs unterschiedlich. Bild 3.10 zeigt den Signalablauf beim U 555 C. Der Anschluß «Chip select» (20), der im normalen Lesebetrieb von +5 V auf 0 V schaltet, erhält beim Programmieren die Spannung +12 V. Nach dem Anschalten der Adressen und Daten erscheint am Programmierereingang (18) ein Impuls mit der Amplitude +26 V und einer maximalen Länge von 1 ms. Zum Beschreiben eines Speicherplatzes sind mindestens 50 ms Programmierzeit erforderlich. Dies geschieht in einem Zyklus mit mindestens 50 Schleifen, bei dem nach 1 ms eine andere Speicherzelle einschaltet. Ein Speicherzugriff des Prozessors dauert je nach Taktfrequenz eine bedeutend kürzere Zeit als 1 ms. Die Schaltung des Programmierzusatzes muß dafür sorgen, daß trotzdem die Daten- und Adreß-Signale am EPROM solange anliegen. Im einfachsten Fall bewirkt das ein monostabiler Multivibrator, der ein WAIT-Signal von 1 ms an den Prozessor gibt. Er

nimmt seine aktive Lage ein, wenn die Speicheranwahl ($\overline{CS}$) des Programmierplatzes anliegt. Das zugehörige Programmierprogramm (Bild 3.12) muß die Erholzeit des monostabilen Multivibrators mit einer Zeitschleife überbrücken. In der Schaltung nach Bild 3.11 bewirkt der Programmierschalter über das Exklusiv-OR-Glied auch die Umschaltung der Phase des $\overline{CS}$ -Signals. Die höher integrierten EPROMs sind einfacher zu programmieren, weil die Programmierzeit von 50 ms mit einem Impuls (nicht in Zyklen) anliegen darf. Damit ist es möglich, das Movekommando des Monitors auch zur Programmierung zu benutzen. Es enthält deshalb eine Zeitschleife von 1 ms für die Erholzeit der Programmierschaltung. Bei den höher integrierten Speichern hat der Programmierimpuls 5-V-Pegel, und die Programmierspannung liegt über einen Schalter oder Stecker an. Die Anschlußbelegungen und die Zeitdiagramme der Typen 2716, 2732 und 2764 unterscheiden sich nur geringfügig. Die Bilder 3.13 bis 3.15 zeigen die Zusatzschaltung für die Programmierung. Der Aufwand ist recht gering, so daß jede normale Speicherkarte auch zur Programmierung der EPROMs dienen kann.

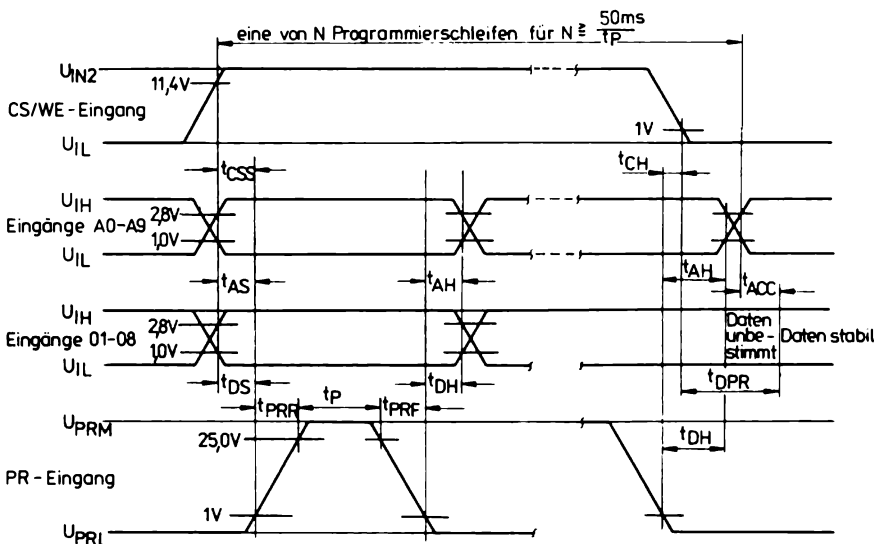


Bild 3.10 Zeitdiagramm der Programmierung des U 555 C

3. Erweiterungen

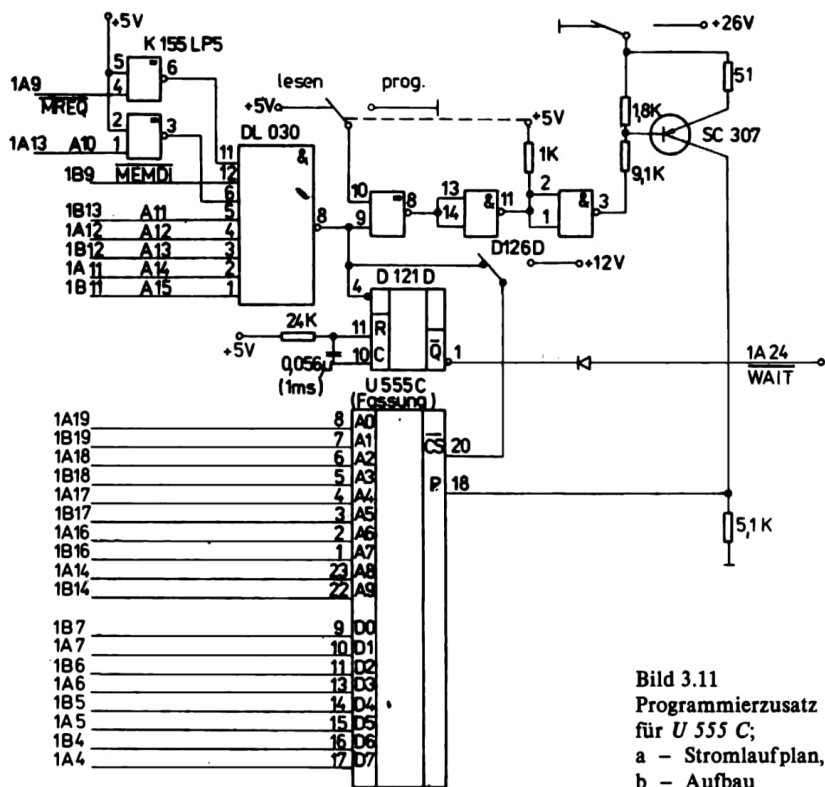
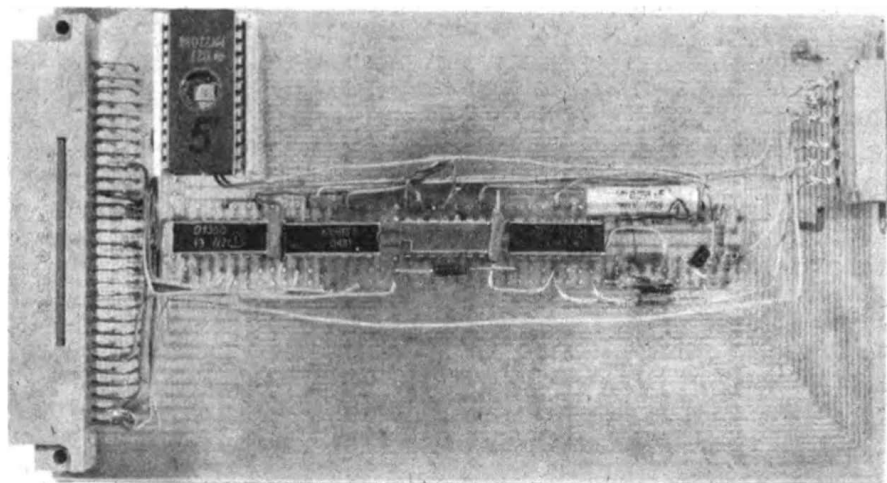


Bild 3.11
 Programmierzusatz
 für U 555 C;
 a - Stromlaufplan,
 b - Aufbau



BBAS-88

LOC	CODE	LINE	SOURCE
		00001	PM PR
		00002	TITL 'PROGRAMMIERPROGRAMM 2708'
		00003	
		00004	ORG OC50H
		00005	
OC50	3EC8	00006	LD A,0C8H ;ANZAHL ZYKLEN
OC52	210010	00007	LD HL,1000H ;QUELLE ,ZU PROGRAMMIERENDES PROG.
OC55	1100F8	00008	LD DE,0F800H ;ZIEL-ADRESSE PROGRAMMIERPLATZ
OC58	010004	00009	LD BC,400H ;ANZAHL SPEICHERPLAETZE
OC5B	F5	00010	P2: PUSH AF ;ERHOLZEIT MONO
OC5C	3EFF	00011	LD A,OFFH
OC5E	3D	00012	P1: DEC A
OC5F	20FD	00013	JRNZ P1-#
OC61	F1	00014	POP AF
OC62	EDA0	00015	LDI
OC64	EA5B0C	00016	JPPE P2 ;NACHSTE ZELLE
OC67	3D	00017	DEC A ;NACHSTER ZYKLUS
OC68	C2520C	00018	JPNZ P3
OC6B	C9	00019	RET
OC6C		00020	END

Bild 3.12 Programmierprogramm für *U 555 C*

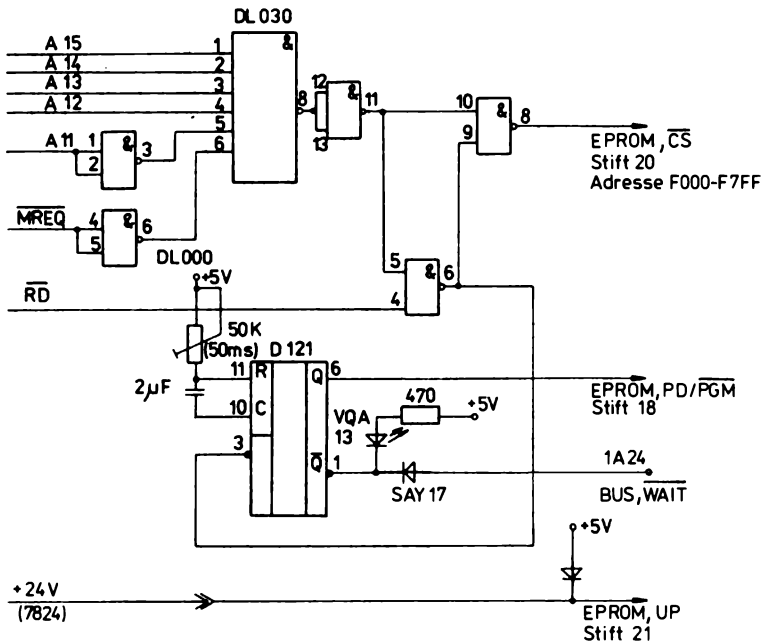


Bild 3.13
· Programmierzusatz
für EPROM
U 2716 C

3. Erweiterungen

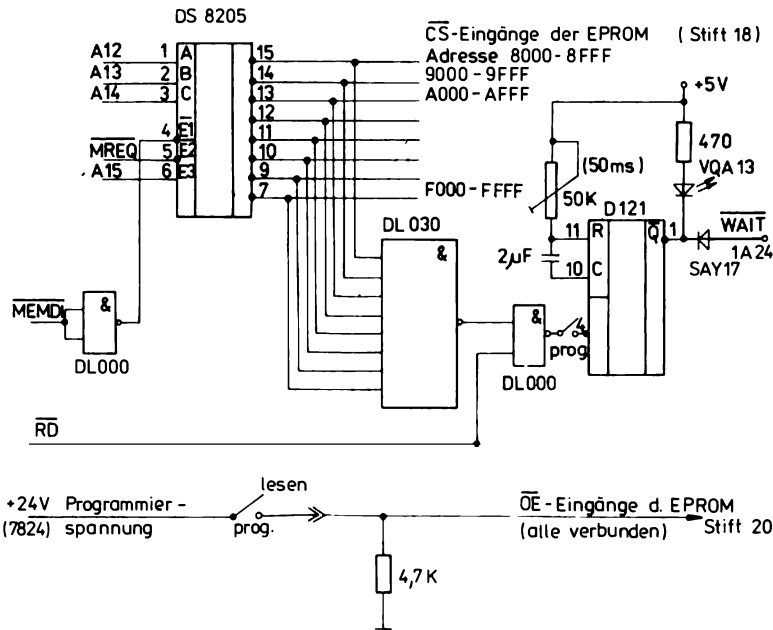


Bild 3.14
Programmierzusatz
für EPROM
U 2732 C

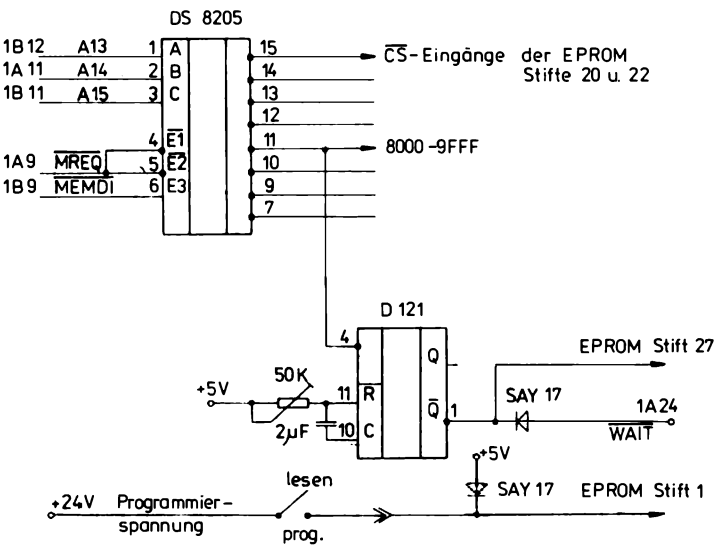


Bild 3.15
Programmierzusatz
für EPROM 2764

Eine andere Möglichkeit, Adressen und Daten am EPROM während des Programmierens stabil zu halten, ist deren Pufferung in Registern, z. B. 8212, 8282 oder Peripherieschaltkreisen. Dies bedingt zwar mehr Aufwand, hat jedoch auch Vorteile:

- Bei einem defekten EPROM ist es nicht ausgeschlossen, daß die Programmspannung auf den Bus gelangt und bei den angeschlossenen Schaltkreisen größeren Schaden anrichtet. Für zwischengeschaltete Puffer ist diese Gefahr geringer.
- Will man den Inhalt dynamischer RAMs in ein EPROM programmieren, muß man längere WAIT-Signale vermeiden. Beim Einsatz von Puffern treten keine WAIT-Signale auf.
- Liegen die Adressen des Programmiergerätes im Speicherbereich, so sind sie für den allgemeinen Gebrauch verloren. Beim Einsatz von Peripherieschaltkreisen oder mit IN/OUT-Adressen gesteuerten Buspuffern ist dies nicht der Fall.

Ein universelles Programmiergerät für alle üblichen EPROM-Typen kann man bei Bedarf unter Verwendung der Schaltungen von Bild 3.11 sowie der 2-PIO-Karte nach Abschnitt 3:5. aufbauen.

3.3. Zeitgeberanschluß

In Elektronikschaltungen arbeiten die unterschiedlichsten Baugruppen frequenz- oder zeitgesteuert, sei es als astabile oder monostabile Multivibratoren oder als Zähler und Teiler. Für diese Anwendungsfälle enthält

das Sortiment der programmierbaren Peripherieschaltkreise spezielle Bauelemente. Der *U 857 CTC* (Counter/Timer Circuit) kann als Teiler für die Taktfrequenz oder als Zähler mit programmierbaren Kennwerten dienen. Mit Erreichen des Zählerinhalts 0 kann er beim Prozessor eine Programmunterbrechung auslösen. Die Bilder 3.16 und 3.17 zeigen die Übersichtsschaltbilder. Jeder der Kanäle kann als unabhängiger Zähler oder Zeitgeber dienen, aber auch die Reihenschaltung von Kanälen ist durch Brücken der Aus- und Eingänge möglich.

In der Betriebsart Zeitgeber gibt der Kanalausgang eine aus der Taktfrequenz gewonnene Frequenz ab. Da Vorteiler und Rückwärtszähler selbstverständlich programmierbar sind, kann sich die Ausgangsfrequenz in weiten Grenzen ändern. Der Vorteiler arbeitet nur mit den Teilerfaktoren 16 und 256, der Rückwärtszähler wird bei Erreichen des Zählerstandes Null mit dem Inhalt des Zeitkonstantenregisters geladen und zählt dann weiter. Der Inhalt des Registers wird bei der Initialisierung des Schaltkreises zwischen 0 und 0FFH gewählt. Den Inhalt des Rückwärtszählers kann die CPU lesen, was aber vor allem bei der Betriebsart «Zähler» eine Rolle spielt. Beim Nulldurchgang des Rückwärtszählers erscheint der Übertragimpuls am Kanalausgang *ZC/T0*, und der Kanal kann einen Interrupt auslösen. Eine übersichtliche Darstellung der Programmierung zeigt Bild 3.18.

Die etwas verzwickte Interrupttechnik soll anhand des Programms für eine Digitaluhr näher betrachtet werden. Ein CTC-Schalt-

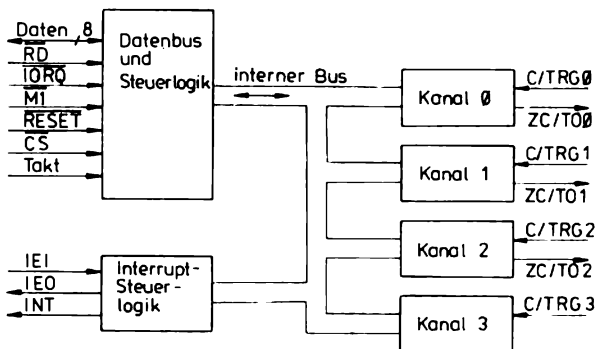


Bild 3.16
Übersichtsschaltplan UB 857 D
(CTC)

3. Erweiterungen

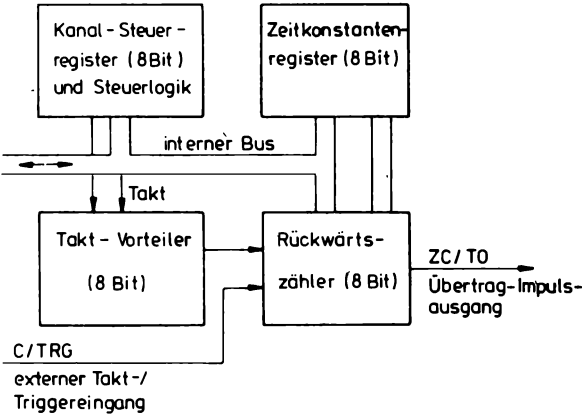


Bild 3.17
Übersichtsschaltplan eines CTC-Kanals

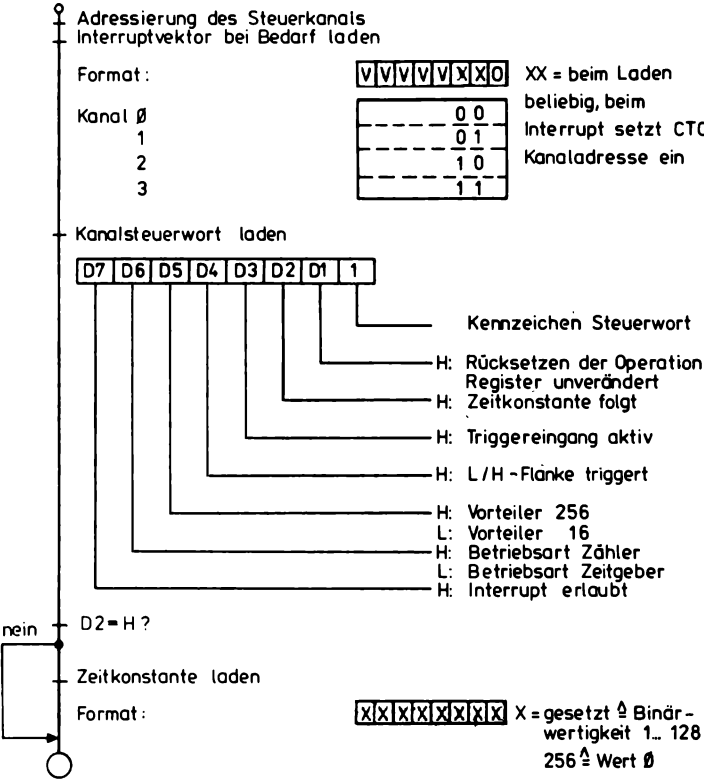


Bild 3.18
Programmierung des CTC

kreis erzeugt dabei alle 20 ms eine Programmunterbrechung. Der Prozessor stellt einige Speicherplätze so, daß darauf die Uhrzeit steht, und gibt sie auf den Bildschirm aus. Zunächst sollen jedoch einige allgemeine Bemerkungen zum Interrupt folgen. Der Prozessor erlaubt 3 Interruptbetriebsarten, Mode 0, Mode 1 und die besonders interessante und leistungsfähige Mode 2. Ein Interrupt bewirkt immer das Zwischenschieben eines Programmstückes in das laufende Programm. Dieses Programmstück ist ein Unterprogramm, d. h., es muß mit einem Rücksprungbefehl RETI abgeschlossen sein, und beim Aufruf wird die aktuelle Adresse im Stack abgelegt. Das Interruptprogramm steht meist auf anderen Adressen als das gerade aktuelle. Wenn in der Interruptroutine Register der CPU benötigt werden, die auch das unterbrochene Programm benutzt, so müssen diese zu Anfang der Interruptroutine in den Speicher gerettet und am Ende wieder zurück in den Prozessor transportiert werden. Natürlich sind auch ineinander geschachtelte Unterbrechungen möglich, die Priorität wird durch die Schaltung festgelegt. Wie bereits erwähnt, ist Mode 2 die leistungsfähigste Interruptbetriebsart. Die anderen beiden können als Überbleibsel der stürmischen Entwicklung dieser Technik betrachtet werden und existieren nur aus Gründen der Austauschbarkeit von Programmen.

Eine Peripherieschaltung löst Interrupt aus, indem sie die Leitung INT auf L-Pegel schaltet. Die CPU reagiert je nach programmierter Interruptbetriebsart. Nach einem RESET ist Mode 0 eingestellt. Dabei erwartet die CPU einen 1-Byte-Befehl, den die Peripherieschaltung abgeben muß, z. B. einen Restart-Befehl, auf dessen Startadresse das Interruptprogramm beginnt. Mode 1 und 2 müssen vorher mit einem Programmbefehl eingestellt werden, eine Rückkehr in Mode 0 ist ebenfalls programmierbar. Bei Mode 1 erfolgt immer ein Sprung auf Adresse 38H, wo dann die Interruptroutine beginnen muß.

In Betriebsart 2 schließlich kann eine ganz beliebige Adresse angesprungen werden. Die Schaltung gibt dazu einen sogenannten Vektor ab. Bei der Initialisierung

wurde er in einem Register des Peripherieschaltkreises gespeichert. Die CPU bildet daraus in Verbindung mit den in ihrem Interruptregister gespeicherten 8 bit eine Adresse, auf der die Startadresse des Interruptprogramms steht. Die CPU führt damit einen indirekten Unterprogrammaufruf aus, der Vektor zeigt auf die Adresse, in der als Dateninhalt die Startadresse der Interruptroutine steht. Im Programmbeispiel Bild 3.19 erscheint dieser Sachverhalt etwas weniger deutlich, weil der Kanal 3 des CTC als interruptfähiger Zeitgeber arbeitet. Im Gegensatz zu den anderen Peripherieschaltkreisen hat der CTC für alle 4 Kanäle nur 1 Vektorregister.

Um die Kanäle unterscheiden zu können, setzt jeder seine Kanalnummer in den Vektor ein (siehe Bild 3.18). Als Vektor für Kanal 0 wurde in Zeile 6 die Adresse 0C08H festgelegt, für Kanal 3 ergibt sich damit als Low-Teil des Vektors 0C0EH (Zeile 79). Dort steht die Adresse der Interruptroutine, im Beispiel die Adresse 0CBBH (Zeilen 78/79 bzw. 88).

Die Inbetriebnahme des Programms erfolgt mit Hilfe des Sprunges ins System-RAM (J-Kommando), da im GO-Kommando des Monitors beim Erreichen eines Haltepunktes die Interruptfähigkeit des Prozessors abschaltet. Dazu ist der Programm-anfang (0C50H) mit dem S-Kommando auf Adresse 0FF9H einzutragen. Nach Aufruf des Uhrenprogramms (Taste J) ist die Uhr zunächst mit S 1234 zu stellen, die Zahl 1234 stellt die aktuelle Uhrzeit in Stunden und Minuten dar.

Die Ankopplung des CTC an den Bus des Mikrocomputers zeigt Bild 3.20, wobei noch ein Schaltkreis *U 856 SIO* für den Anschluß eines Fernschreibers und eines Kassettens magnetbandgerätes gemeinsam mit dem CTC arbeitet. Die Erläuterung dieses Schaltkreises folgt im nächsten Abschnitt. Tabelle 3.1. enthält die Erläuterung der Anschlüsse des CTC. Die CTC-Schaltung weist keine Besonderheiten auf, man muß nur auf ordnungsgemäßen Anschluß der Leitungen INT, IEI und IEO achten, weil sonst der Interruptbetrieb nicht funktioniert. Ein Schaltkreis kann Interrupt nur auslösen, wenn sein

3. Erweiterungen

EBAS-88			
LOC	CODE	LINE	SOURCE
		00001	PN UR
		00002	CTCA: EQU 6CH ;ZEITGEBER SIO
		00003	CTCB: EQU 6DH ;ZEITGEBER SIO
		00004	CTCC: EQU 6EH ;ZAEHLKANAL
		00005	CTCD: EQU 6FH ;ZEITGEBER UHR
		00006	UVZ: EQU 0C08H ;VEKTOR
		00007	UVEK: EQU 1(UVZ) ;VEKTOR LOW-TEIL
		00008	BEART: EQU 0B5H ;BETRIEBSART CTCD
		00009	ZKON: EQU 115 ;ZEITKONSTANTE CTCD
		00010	UPU: EQU 0B00H ;MILLISEK.-PUFFER
		00011	ASTD: EQU UPU+3 ;STUNDENPUFFER
		00012	AMIN: EQU UPU+2
		00013	ASEK: EQU UPU+1
		00014	CRLF: EQU 032DH ;ROUTINEN AUS DEM MONITOR
		00015	PCHK: EQU 518H
		00016	PARAM: EQU 4D7H
		00017	CONV: EQU 495H
		00018	BPLA: EQU 0FC39H ;BILDSCHIRMPLATZ F. ANZEIGE
		00019	TITL 'UHR'
		00020	ORG 0C50H
		00021	MUHR: EQU #
OC50	0E02	00022	LD C,2
OC52	CD2D03	00023	CALL CRLF
OC55	CD1805	00024	CALL PCHK
OC58	3818	00025	JRC UANZ-# ;UHRZEIT AUSGEBEN
OC5A	FE53	00026	CMP 'S'
OC5C	C0	00027	RNZ
		00028	;UHRZEIT STELLEN
OC5D	CDD704	00029	CALL PARAM
OC60	7C	00030	LD A,H
OC61	CDF20C	00031	CALL BCDBI
OC64	32030E	00032	LD (ASTD),A
OC67	7D	00033	LD A,L
OC68	CDF20C	00034	CALL BCDBI
OC6B	32020E	00035	LD (AMIN),A
OC6E	CD9B0C	00036	CALL ICTCD
OC71	C9	00037	RET
		00038	;UHRZEIT AUSGEBEN
		00039	UANZ: EQU #
OC72	F5	00040	PUSH AF
OC73	C5	00041	PUSH BC
OC74	D5	00042	PUSH DE
OC75	E5	00043	PUSH HL
OC76	21010E	00044	LD HL,ASEK
OC79	11040E	00045	LD DE,UPU+4
OC7C	010300	00046	LD BC,3
OC7F	F3	00047	DI
OC80	EDB0	00048	LDIR
OC82	FB	00049	EI
OC83	0603	00050	LD B,3
OC85	21060E	00051	LD HL,UPU+6
OC88	1139FC	00052	LD DE,BPLA
OC8B	7E	00053	UANZ1: LD A,(HL)
OC8C	CD040D	00054	CALL BICBD
OC8F	CD190D	00055	CALL BBYTE
OC92	2B	00056	DEC HL
OC93	13	00057	INC DE
OC94	10F5	00058	DJNZ UANZ1-#
OC96	E1	00059	POP HL
OC97	D1	00060	POP DE
OC98	C1	00061	POP BC
OC99	F1	00062	POP AF
OC9A	C9	00063	RET
		00064	;INITIALISIERUNG CTCD ALS ZEITGEBER 20 MS
OC9B	E5	00065	ICTCD: PUSH HL
OC9C	C5	00066	PUSH BC
OC9D	F5	00067	PUSH AF
OC9E	ED5E	00068	IM2 ;INTERRUPT-MODE 2 (MIT VEKTOR)
OCA0	3E0C	00069	LD A,0CH ;HIGH VEKTOR
OCA2	ED47	00070	LD I,A
OCA4	FB	00071	EI ;ERLAUBE INTERRUPT
OCA5	3EB5	00072	LD A,BEART ;BETRIEBSART
OCA7	D36F	00073	OUT CTCD
OCA9	3E73	00074	LD A,ZKON ;ZEITKONSTANTE
OCAB	D36F	00075	OUT CTCD
OCAD	3E08	00076	LD A,UVEK ;UHRVEKTOR
OCAF	D36C	00077	OUT CTCA ;UHRVEKTOR LOW-TEIL
OCB1	21BB0C	00078	LD HL,SE ;INTERRUPTROUTINE


```

OCB4 220B0C 00079 LD (UVZ+6),HL ;VERSATZ P.CTC-KANAL D
OCB7 F1 00080 POP AF
OCB8 C1 00081 POP BC
OCB9 E1 00082 POP HL
OCBA C9 00083 RET
00084 ;ISR SEKUNDENTAKT
00085 ;IN: -
00086 ;OUT:UHRZEIT + 20 MS
00087 SE: EQU #
OCBB E5 00088 PUSH HL
OCBC F5 00089 PUSH AF
OCBD 21000E 00090 LD HL,UPU ;MILLISEKUNDEN -ZAEHLER
OCC0 35 00091 DEC (HL)
OCC1 2008 00092 JRNZ MSAUS-#
OCC3 3632 00093 LD (HL),32H ;= 1 SEK
OCC5 CDD00C 00094 CALL UHR
OCC8 CD720C 00095 CALL UANZ
OCCB F1 00096 MSAUS: POP AF
OCCC E1 00097 POP HL
OCCD FB 00098 EI
OCEE ED4D 00099 RETI
00100 ;SUBROUTINE UHRZEIT
00101 ;AUFRUF ALLE SEKUNDE
00102 ;WEITERSTELLEN DER UHR IM SPEICHER
00103 UHR: EQU #
OCD0 E5 00104 PUSH HL
OCD1 D5 00105 PUSH DE
OCD2 C5 00106 PUSH BC
OCD3 F5 00107 PUSH AF
OCD4 0603 00108 LD B,3
OCD6 11010E 00109 LD DE,UPU+1
OCD9 21EFOC 00110 LD HL,SEK
OCD C 1A 00111 UHR1: LD A,(DE)
OCD D 3C 00112 INC A
OCD E BE 00113 CMP (HL)
OCD F 2008 00114 JRNZ UHR2-#
OCE1 AF 00115 XOR A
OCE2 12 00116 LD (DE),A
OCE3 23 00117 INC HL
OCE4 13 00118 INC DE
OCE5 40F5 00119 DJNZ UHR1-#
OCE7 1801 00120 JR UHR3-#
OCE9 12 00121 UHR2: LD (DE),A
OCEA F1 00122 UHR3: POP AF
OCEB C1 00123 POP BC
OCEC D1 00124 POP DE
OCE D E1 00125 POP HL
OCEE C9 00126 RET
00127 SEK: EQU #
OCEF 3C 00128 DB 60 ;SEK
OCF0 3C 00129 DB 60 ;MIN
OCF1 18 00130 DB 24 ;STD
00131 ;*-----*
00132 ;SUBR. ZUR KONVERTIERUNG BCD --> BINAER
00133 ; EING.PARAM.: A...DEZIMAL (BCD-FORMAT)
00134 ; AUSG.PARAM.: A...BINAERWERT
00135 ; VERAEND.REG.: A,F
OCF2 C5 00136 BCDBI: PUSH BC
OCF3 4F 00137 LD C,A
OCF4 0F 00138 RRCA
OCF5 0F 00139 RRCA
OCF6 0F 00140 RRCA
OCF7 0F 00141 RRCA
OCF8 E60F 00142 AND OFH
OCFA 47 00143 LD B,A
OCFB 79 00144 LD A,C
OCFC 2804 00145 JRZ BC2-#
OCFE D606 00146 BC1: SUB 6
OD00 10FC 00147 DJNZ BC1-#
OD02 C1 00148 BC2: POP BC
OD03 C9 00149 RET
00150
00151 ;;;
00152
00153 ;SUBR. ZUR KONVERTIERUNG BINAER --> BCD
00154 ; EING.PARAM.: A...BINAERWERT
00155 ; AUSG.PARAM.: A...DEZIMAL (BCD-FORMAT)
00156 ; VERAEND.REG.: A,F
OD04 C5 00157 BIBCD: PUSH BC
OD05 0600 00158 LD B,0
OD07 B7 00159 OR A
OD08 D60A 00160 BI1: SUB 10
OD0A 04 00161 INC B

```

3. Erweiterungen

OD0B	30FB	00162	JRNC BI1-#
OD0D	C60A	00163	ADD 10
OD0F	05	00164	DEC B
OD10	4F	00165	LD C,A
OD11	78	00166	LD A,B
OD12	07	00167	RLCA
OD13	07	00168	RLCA
OD14	07	00169	RLCA
OD15	07	00170	RLCA
OD16	81	00171	ADD C
OD17	C1	00172	POP BC
OD18	C9	00173	RET
		00174	
		00175	
		00176	;SUBROUTINE ZUR AUSGABE VON 8 BIT
		00177	;BINÄRINFORMATION AUF BILDSCHIRM
		00178	;IN:A BYTE
		00179	; DE BILDSCHIRMPLATZ
		00180	;OUT:DE-DE+1
		00181	;KILL:AF
		00182	B BYTE: EQU #
OD19	F5	00183	PUSH AF
OD1A	D5	00184	PUSH DE
OD1B	0F	00185	RRCA
OD1C	0F	00186	RRCA
OD1D	0F	00187	RRCA
OD1E	0F	00188	RRCA
OD1F	CD9504	00189	CALL CONV
OD22	D1	00190	POP DE
OD23	12	00191	LD (DE),A
OD24	F1	00192	POP AF
OD25	13	00193	INC DE
OD26	D5	00194	PUSH DE
OD27	CD9504	00195	CALL CONV
OD2A	D1	00196	POP DE
OD2B	12	00197	LD (DE),A
OD2C	C9	00198	RET
OD2D		00199	BND

Bild 3.19 Uhrprogramm

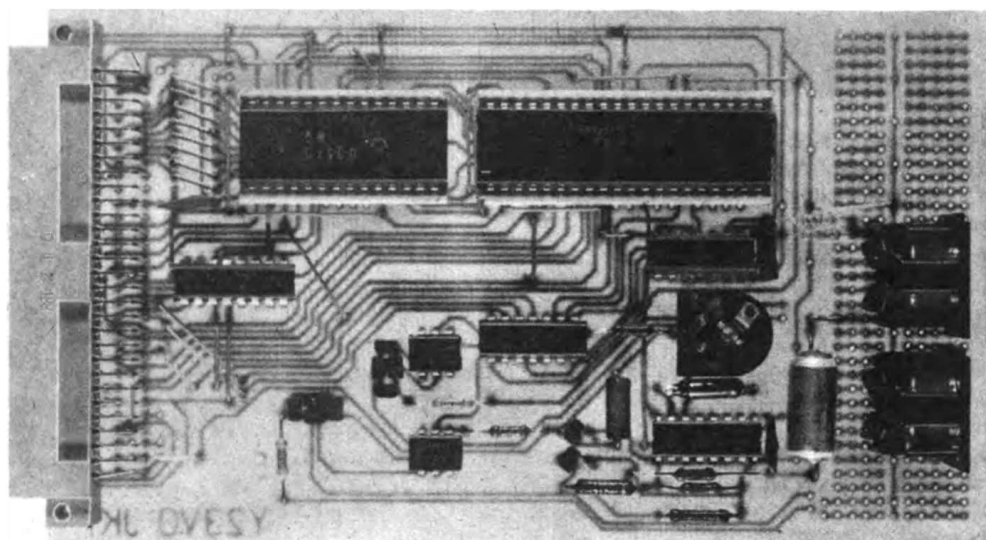


Bild 3.20 CTC/SIO-Karte; b – bestückte Leiterplatte

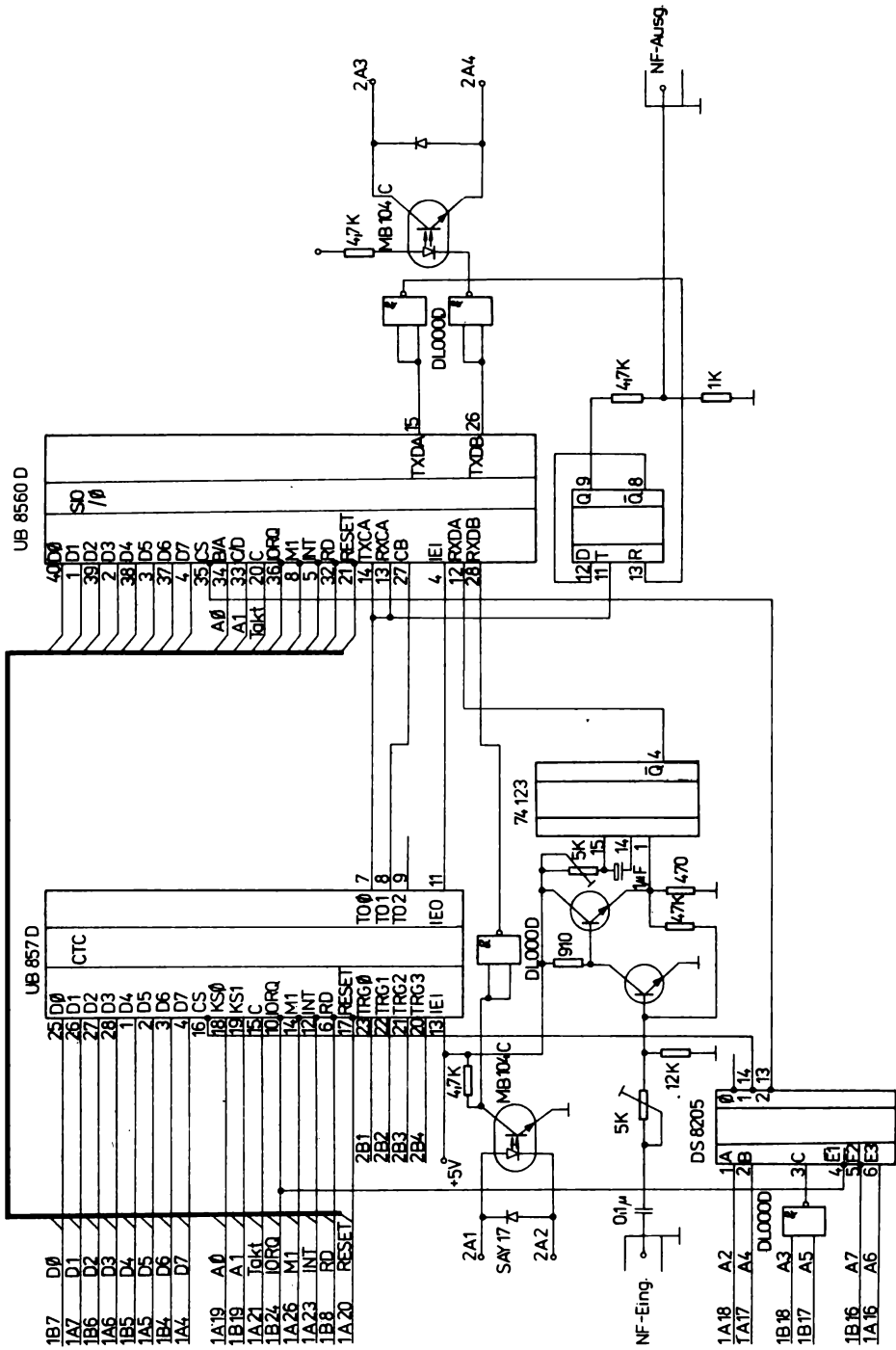


Bild 3.20 CTC/SIO-Karte; a – Stromlaufplan

3. Erweiterungen

Tabelle 3.1. Anschlußbezeichnungen U 857 CTC

Signalname	Signalbedeutung
D0 bis D7	8-bit-bidirektionaler Datenbus: Tri-state
CE	Chipauswahl: Eingang, Low-aktiv
KS0; KS1	Kanalauswahl: Eingabe einer 2-bit-Adresse des vom Mikroprozessor angesprochenen Kanals
$\overline{M\bar{I}}$	CPU-Maschinenzyklus M1: Eingang, Low-aktiv
$\overline{I\bar{O}R\bar{Q}}$	Ein-/Ausgabeanforderung: Eingang, Low-aktiv
RD	CPU-Leseanforderung: Eingang, Low-aktiv
IEI	Interruptfreigabe: Eingang
C/TRG0	Takt-/Triggereingänge
C/TRG1	(Kanal 0 bis 3)
C/TRG2	Takteingänge für Zähler bzw. Zeitgebertriggerung
C/TRG3	Programmierbar: High- oder Low-aktiv
$\overline{R\bar{E}S\bar{E}T}$	Rücksetzeingang: Low-aktiv
	Der Zählvorgang aller Kanäle wird unterbrochen, und die Interrupt-Freigabebits der Steuerregister aller 4 Kanäle werden zurückgesetzt.
	Die Ausgänge ZC/TO 0...2 und $\overline{I\bar{N}T}$ werden in den inaktiven Zustand gebracht. Der Ausgang IEO wird gleich dem Wert am Eingang IEI gesetzt. Das Interrupt-Vektorregister wird nicht beeinflusst. Die Datenein-/Ausgänge werden in den hochohmigen Zustand gebracht.
C	Systemtakt
IEO	Interruptfreigabe: Ausgang;
	High-aktiv für Interruptprioritätskette
ZC/TO0	Nulldurchgang/Zeitgebermeldung-Ausgänge
ZC/TO1	(Kanal 0 bis 2)
ZC/TO2	Nullsignal des Rückwärtszählers bzw. Meldung des Zeitgebers
$\overline{I\bar{N}T}$	Interruptanforderung, Ausgang, Open-Drain-Ausgang, Low-aktiv

IEI auf H-Pegel liegt. Beim CTC ist das in dieser Schaltung immer der Fall. Sein Kanal 0 hat deshalb die höchste Priorität (Vorrang). Die Kanäle 0 und 1 erzeugen jedoch nur Frequenzen für den SIO, Kanal 2 ist außer Betrieb. Wenn Kanal 3 einen Interrupt anmeldet, schalten die Leitungen INT und IEO auf L-Pegel. Damit liegt an IEI des SIO ebenfalls L-Pegel, und er kann keinen Interrupt mehr auslösen, auch wenn er entsprechend programmiert worden wäre. Damit ist durch die Schaltung die Priorität festgelegt: CTC vor SIO. Der Prozessor wertet den L-Pegel auf der Leitung INT aus, indem er gleichzeitig die Signale $\overline{I\bar{O}R\bar{Q}}$ und $\overline{M\bar{I}}$ ausgibt. Sonst kommt $\overline{M\bar{I}}$ nur in Verbindung mit $\overline{M\bar{R}\bar{E}\bar{Q}}$ vor.

Dies veranlaßt den Peripherieschaltkreis, seinen Vektor abzugeben. Der Prozessor springt in der bereits geschilderten Weise in die Interruptroutine, an deren Ende der Befehl RETI stehen muß. Dieser wird nicht nur im Prozessor durch den Rücksprung auf die aktuelle Adresse des unterbrochenen Programms ausgewertet, sondern auch im Peripherieschaltkreis als Ende der Unterbrechung erkannt, und die Leitungen INT und IEO schalten wieder auf H-Pegel.

3.4. Datenspeicherung auf Magnetband sowie Aus- und Eingabe serieller Daten

Bereits mit dem Grundgerät sind Aufzeichnungen und Einlesen von Programmen mit Magnetbandkassetten sowie das Ausdrucken von Programmen über den seriellen Ausgang möglich. Die IN/OUT-Routinen des Monitorprogramms arbeiten mit 7-bit-ASCII-Zei-

chen, je einem Start- und Stopbit und einer Übertragungsgeschwindigkeit von 100 bit je Sekunde. Die Geschwindigkeit kann man in den RAM-Speicherplätzen FFC und FFD einstellen. Das Lese-Programm (Programm 5.2.1.) arbeitet um die Länge eines bits schneller (Zeile 610). Bei Eingabe eines Zeichens erfolgt in der Zeit des Stopbits die Verarbeitung im Monitor, z. B. im Readkommando. Man kann deshalb diese Zeit

EBAS-88

LOC	CODE	LINE	SOURCE
		00001	PN FS
		00002	TITL 'FERNSCHREIBEN'
		00003	FLGIO: EQU OFFEH ;SPEICHERPLATZ F.STEUERUNG I/O-GERAETE
		00004	IOBYT: EQU OFFFH ;SPEICHERPLATZ F.ZUORDNUNG I/O-GERAETE
		00005	
		00006	PIOAD: EQU OFCH ;BIT 7 EINGABE,BIT 0 AUSGABE
		00007	CTCA: EQU 06CH
		00008	CTCB: EQU 06DH
		00009	SIOAD: EQU 078H
		00010	SIOBD: EQU 079H
		00011	SIOAC: EQU 07AH
		00012	SIOBC: EQU 07BH
		00013	USRTC: EQU 7FH
		00014	USRTD: EQU 7EH
		00015	
		00016	USBZ: EQU 1BH ;UMSCHALTUNG BUCHST.-ZIFFERN-
		00017	USZB: EQU 1FH ;UMSCHALTUNG ZIFFERN-BUCHST.
		00018	CR: EQU 00DH ;WAGEN-RUECKLAUF
		00019	LF: EQU 00AH ;ZEILENVORSCHUB
		00020	ETX: EQU 003H ;CONTROL/C (END OF TEXT)
		00021	APST: EQU 27H ;APOSTROPH
		00022	ZS: EQU 6DH ;ZEITSCHLEIFE MONITOR
		00023	;PARAMETER BEI AUSGABE
		00024	;***IN: C...ZEICHEN
		00025	;**OUT: -
		00026	;*KILL: A,F
		00027	;PARAMETER BEI EINGABE
		00028	;***IN: -
		00029	;**OUT: A...ZEICHEN
		00030	;*KILL: A,F
		00031	ORG 0C50H
		00032	;FERNSCHREIBEINGABE UEBER PIO
0C50	DBFC	00033	FEP: IN PLOAD
0C52	CB7F	00034	BIT 7,A ;WARTEN AUF STARTBIT
0C54	28FA	00035	JRZ FEP-#
0C56	CD890D	00036	CALL FSE
0C59	1F	00037	RRA
0C5A	E61F	00038	AND 1FH ;STARTBIT RAUSSCHIEBEN
0C5C	CD120D	00039	CALL CVCA ;STOPBITS AUSBLENDEN
0C5F	28EF	00040	JRZ FEP-#
0C61	C9	00041	RET
		00042	;*****
		00043	;FERNSCHREIBEAUSGABE UEBER PIO
0C62	F5	00044	PAP: PUSH AF
0C63	CDC30C	00045	CALL CVAC
0C66	B7	00046	OR A
0C67	C4760D	00047	CANZ FSA ;ERST STEUERZEICHEN
0C6A	79	00048	LD A,C
0C6B	CD760D	00049	CALL FSA
0C6E	F1	00050	POP AF
0C6F	C9	00051	RET
		00052	;*****
		00053	;FERNSCHREIBEINGABE UEBER SIO
0C70	DB7B	00054	FES: IN SIOBC
0C72	0F	00055	RRCA
0C73	30FB	00056	JRNC FES-#
0C75	DB79	00057	IN SIOBD
0C77	E61F	00058	AND 1FH

3. Erweiterungen

```

0C79 CD120D 00059 CALL CVCA
0C7C 28F2 00060 JRZ FES-#
0C7E C9 00061 RET
00062 ;*-----*
00063 ;FERNSCHREIBAUSGABE UEBER SIO
00064 FAS: EQU #
00065 PUSH AF
00066 CALL CVAC
00067 OR A
00068 CANZ FAS1 ;ERST STEUERZEICHEN
00069 LD A,C
00070 CALL FAS1
00071 POP AF
00072 RET
00073 FAS1: PUSH AF
00074 FAS2: IN SIOBC
00075 AND 4 ;PUFFER LEER?
00076 JRZ FAS2-#
00077 POP AF
00078 OUT SIOBD
00079 RET
00080 ;*-----*
00081 ;ASCII-AUSGABE UEBER SIO
00082 AAS: RES 7,C
00083 AAS1: IN SIOAC
00084 AND 4
00085 JRZ AAS1-#
00086 LD A,C
00087 OUT SIOAD
00088 RET
00089 ;*-----*
00090 ;ASCII-BINGABE UEBER SIO
00091 AAS: IN SIOAC
00092 RRCA
00093 JRNC AAS-#
00094 IN SIOAD
00095 AND 7FH
00096 RET
00097 ;*-----*
00098 ;ASCII-AUSGABE UEBER USART 8251
00099 AAU: IN USRTC
00100 BIT 0,A ;SENDER BEREIT ?
00101 JRZ AAU-#
00102 LD A,C ;ZEICHENAUSGABE
00103 OUT USRTD
00104 RET
00105 ;*-----*
00106 ;ASCII-BINGABE UEBER USART 8251
00107 AEU: IN USRTC
00108 BIT 1,A ;EMPFAENGER BEREIT?
00109 JRZ AEU-#
00110 IN USRTD ;ZEICHENBINGABE
00111 AND 7FH
00112 RET
00113 ;*-----*
00114 ; SUBROUTINEN
00115 ;*-----*
00116 ;SUBR. ZUR KONVERTIERUNG ASCII-CCITT
00117 ;**IN: C...ZEICHEN IM ASCII-KODE
00118 ;**OUT: C...ZEICHEN IM CCITT-KODE
00119 ; A=0...KEINE UMSCHALTUNG BUCHST.-ZEICHEN
00120 ; A'0...KODE FUER UMSCH. BUCHST.-ZEICHEN
00121 ;*KILL: A,P,C
00122 CVAC: PUSH HL
00123 PUSH BC
00124 LD A,C
00125 AND 7FH
00126 BIT 6,A ;GROSS=-KLEIN
00127 CANZ GK
00128 CVACO: LD HL,TBCVZ+LTB-1
00129 LD BC,LTB
00130 CPDR
00131 JRZ CVAC1-#
00132 LD A,'$'
00133 JR CVACO-#
00134 CVAC1: CMP CR
00135 JRZ CVAC6-#
00136 CMP LF
00137 JRZ CVAC6-#
00138 CMP ' '
00139 JRZ CVAC6-#
00140
0C79 E5 00059
0C7C C5 00060
0C7E 79 00061
0C7E E67F 00062
0C80 CDC30C 00063
0C83 B7 00064
0C84 C48DOC 00065
0C87 79 00066
0C88 CD8DOC 00067
0C8B F1 00068
0C8C C9 00069
0C8D F5 00070
0C8E DB7B 00071
0C90 E604 00072
0C92 28FA 00073
0C94 F1 00074
0C95 D379 00075
0C97 C9 00076
0C98 CBB9 00077
0C9A DB7A 00078
0C9C E604 00079
0C9E 28FA 00080
0CA0 79 00081
0CA1 D378 00082
0CA3 C9 00083
0CA4 DB7A 00084
0CA6 0F 00085
0CA7 30FB 00086
0CA9 DB78 00087
0CAB E67F 00088
0CAD C9 00089
0CAE DB7F 00090
0CB0 CB47 00091
0CB2 28FA 00092
0CB4 79 00093
0CB5 D37E 00094
0CB7 C9 00095
0CB8 DB7F 00096
0CBA CB4F 00097
0CBC 28FA 00098
0CBE DB7E 00099
0CC0 E67F 00100
0CC2 C9 00101
0CC3 E5 00102
0CC4 C5 00103
0CC5 79 00104
0CC6 E67F 00105
0CC8 CB77 00106
0CCA C40F0D 00107
0CCD 21750D 00108
0CD0 014000 00109
0CD3 EDB9 00110
0CD5 2804 00111
0CD7 3E40 00112
0CD9 18F2 00113
0CDB FE0D 00114
0CDD 281C 00115
0CDF FE0A 00116
0CE1 2818 00117
0CE3 FE20 00118
0CE5 2814 00119

```

```

OCB7 79      00141  CVAC2: LD  A,C
OCB8 FB1F    00142  CMP  USZB
OCBA 21F80F  00143  LD   HL,FLGIO
OCED 3012    00144  JRNZ CVAC5-#
OCBF CB4E    00145  BIT  1,(HL)          ;0Z
OCF1 2009    00146  JRNZ CVAC4-#
OCF3 CBCE    00147  SET  1,(HL)          ;0Z
OCF5 C1      00148  CVAC3: POP  BC
OCF6 4F      00149  LD   C,A
OCF7 3E1B    00150  LD   A,USBZ
OCF9 E1      00151  POP  HL
OCFA C9      00152  RET
OCFB 79      00153  CVAC6: LD  A,C
OCFC C1      00154  CVAC4: POP  BC
OCFD 4F      00155  LD   C,A
OCFE AF      00156  XOR  A
OCFF E1      00157  POP  HL
OD00 C9      00158  RET
OD01 D620    00159  CVAC5: SUB  20H
OD03 CB4E    00160  BIT  1,(HL)          ;0Z
OD05 2B55    00161  JRZ  CVAC4-#
OD07 CB8E    00162  RES  1,(HL)          ;0Z
OD09 C1      00163  POP  BC
OD0A 4F      00164  LD   C,A
OD0B 3E1F    00165  LD   A,USBZ
OD0D E1      00166  POP  HL
OD0E C9      00167  RET
OD0F CBAF    00168  GK:  RES  5,A
OD11 C9      00169  RET
          00170
          00171  ;SUBR.ZUR KONVERTIERUNG CCITT-ASCII
          00172  ;***IN: A...CCITT-ZEICHEN
          00173  ;**OUT: A...ASCII-ZEICHEN
          00174  ;*KILL A,F
OD12 E5      00175  CVCA:  PUSH HL
OD13 21F80F  00176  LD   HL,FLGIO
OD16 FB1F    00177  CMP  USZB
OD18 2002    00178  JRNZ CVCA1-#
OD1A CB86    00179  RES  0,(HL)
OD1C FB1B    00180  CVCA1: CMP  USBZ
OD1E 2002    00181  JRNZ CVCA2-#
OD20 CBC6    00182  SET  0,(HL)
OD22 CB46    00183  CVCA2: BIT  0,(HL)
OD24 21360D  00184  LD   HL,TBCVZ
OD27 2003    00185  JRNZ CVCA0-#
OD29 21560D  00186  LD   HL,TBCVB
OD2C 85      00187  CVCA0: ADD  L
OD2D 6F      00188  LD   L,A
OD2E 7C      00189  LD   A,H
OD2F CE00    00190  ADC  0
OD31 67      00191  LD   H,A
OD32 7E      00192  LD   A,(HL)
OD33 E1      00193  POP  HL
OD34 B7      00194  OR   A
OD35 C9      00195  RET
          00196  ;TABELLE ZUR KONVERTIERUNG CCITT '---' ASCII
OD36          00197  TBCVZ: DB  '#3'
OD38 0A      00198  DB  'LF'
OD39          00199  DB  ' '
OD3B 27      00200  DB  'APST'
OD3C          00201  DB  '87'
OD3E 0D      00202  DB  'CR'
OD3F 03      00203  DB  'RTX'
OD40          00204  DB  '4;1'
OD44          00205  DB  ':(5+'
OD48          00206  DB  '256'
OD4C          00207  DB  '019'
OD4F          00208  DB  '7%'
OD51 00      00209  DB  0
OD52          00210  DB  '._/='
OD55 00      00211  DB  0
OD56          00212  TBCVB: DB  '*B'
OD58 0A      00213  DB  'LF'
OD59          00214  DB  'A '
OD5B          00215  DB  'SIU'
OD5E 0D      00216  DB  'CR'
OD5F          00217  DB  'DRJN'
OD63          00218  DB  'FCKT'
OD67          00219  DB  'ZLWH'
OD6B          00220  DB  'YPQ0'
OD6F          00221  DB  'BG'
OD71 00      00222  DB  0

```

3. Erweiterungen

```

0D72      00      00223      DB      'MXV'
0D75      00      00224      DB      0
00225      LTB:    EQU      #-TBCVZ
00226      ;*-----*
00227      ;FERNSCHREIBZEICHENAUSGABE(PIO)
00228      ;IN:A-ZEICHEN IN CCITT OHNE START- UND STOP-BITS
00229      ;OUT:A-ZEICHEN KOMPLETT
00230      ;KILL:PIOAD
00231      FSA:     EQU      #
0D76      C5      00232      PUSH   BC
0D77      0608    00233      LD      B,8
0D79      F660    00234      OR      60H      ;CY=0,2 STOP-BITS
0D7B      17      00235      RLA          ;CY=STARTBIT
0D7C      D3FC    00236      FSA1:   OUT    PIOAD
0D7E      CD6D00  00237      CALL   ZS
0D81      CD6D00  00238      CALL   ZS
0D84      1F      00239      RRA
0D85      10F5    00240      DJNZ   FSA1-#
0D87      C1      00241      POP    BC
0D88      C9      00242      RET
00243      ;*-----*
00244      ;FERNSCHREIBZEICHENEINGABE(PIO)
00245      ;IN:-
00246      ;OUT:A...FS-ZEICHEN
00247      ;KILL:AF
00248      FSE:     EQU      #
0D89      C5      00249      PUSH   BC
0D8A      0607    00250      LD      B,07      ;BITZAEHLER
0D8C      CD6D00  00251      CALL   ZS
0D8F      DBFC    00252      FSE1:   IN     PIOAD      ;BIT EINLESEN
0D91      CD6D00  00253      CALL   ZS
0D94      CD6D00  00254      CALL   ZS
0D97      17      00255      RLA
0D98      CB19    00256      RR      C
0D9A      10F3    00257      DJNZ   FSE1-#
0D9C      CB19    00258      RR      C
0D9E      79      00259      LD      A,C
0D9F      2F      00260      CPL
0DA0      F6C0    00261      OR      0COH
0DA2      C1      00262      POP    BC
0DA3      C9      00263      RET
00264      ;*-----*
00265      ;INITIALISIERUNG CTC UND SIO
0DA4      0602    00266      BEGIN:  LD      B,LCTCA
0DA6      0B6C    00267      LD      C,CTCA
0DA8      21CD0D  00268      LD      HL,TBCTC
0DAB      EDB3    00269      OTIR
0DAD      0602    00270      LD      B,LCTCB
0DAF      0E6D    00271      LD      C,CTCB
0DB1      EDB3    00272      OTIR
0DB3      0608    00273      LD      B,LSIOA
0DB5      0E7A    00274      LD      C,SIOAC
0DB7      EDB3    00275      OTIR
0DB9      0608    00276      LD      B,LSIOB
0DBB      0E7B    00277      LD      C,SIOBC
0DBD      EDB3    00278      OTIR
0DBF      C9      00279      RET
00280      ;*-----*
00281      ;INITIALISIERUNG USART F.ASCII
0DC0      3E40    00282      IUSRT:  LD      A,40H
0DC2      D37F    00283      OUT    USRTC
0DC4      3E4A    00284      LD      A,4AH      ;2STOP,OHNE PAR.,7BIT,CX16
0DC6      D37F    00285      OUT    USRTC
0DC8      3E27    00286      LD      A,27H
0DCA      D37F    00287      OUT    USRTC
0DCC      C9      00288      RET
00289      ;*-----*
00290      ;TABELLEN ZUR PROGRAMMIERUNG VON CTC UND SIO
00291      TBCTC:    EQU      #
00292      ;PROGRAMMIERUNG CTC FUER SIO KANAL A (4800HZ)
00293      TCTCA:    EQU      #
0DCD      05      00294      DB      5
0DCB      14      00295      DB      20      ;3CH F.100 BAUD
00296      LCTCA:    EQU      #-TCTCA
00297      ;PROGRAMMIERUNG CTC SIO KANAL B (3200HZ)
00298      TCTCB:    EQU      #
0DCF      05      00299      DB      5
0DD0      1E      00300      DB      30
00301      LCTCB:    EQU      #-TCTCB
00302      ;PROGRAMMIERUNG SIO FUER ASCII
00303      TSIOA:    EQU      #
0DD1      04      00304      DB      4

```



```

ODD2 48      00305      DB 48H      ;TEILER 16,1STOP-BIT
ODD3 01      00306      DB 1
ODD4 00      00307      DB 0          ;INT-SPERRE
ODD5 03      00308      DB 3
ODD6 41      00309      DB 041H      ;7BIT,RX-FREIGABE
ODD7 05      00310      DB 5
ODD8 28      00311      DB 28H       ;7BIT,TX-FREIGABE
          00312      LSIOA: EQU #-TSIOA
          00313      ;PROGRAMMIERUNG SIO FUER FERNSCHREIBER
          00314      TSIOB: EQU #
ODD9 04      00315      DB 4
ODDA CC      00316      DB 0CCH      ;TEILER 64,2STOP-BITS
ODDB 01      00317      DB 1
ODDC 00      00318      DB 0          ;INT-SPERRE
ODDD 03      00319      DB 3
ODDE 01      00320      DB 1          ;5BIT,RX-FREIGABE
ODDF 05      00321      DB 5
ODE0 08      00322      DB 8          ;8BIT,TX-FREIGABE
          00323      LSIOB: EQU #-TSIOB
ODE1          00324      END

```

Bild 3.21 Programme für serielle Datenübertragung

nicht wesentlich verkürzen, ohne von der Einzelzeichenverarbeitung im Monitor abzugehen. Nur durch blockweises Einlesen in einen Pufferspeicher und nachträgliche Auswertung kann die Geschwindigkeit ohne Änderung der Schaltung erhöht werden. Zunächst sollen jedoch noch Hinweise für die Möglichkeiten des Fernschreiberanschlusses beim Grundgerät folgen. Die Schaltung ist dazu zwar geeignet, jedoch fehlt im Monitor eine entsprechende Routine.

Die Fernschreibtechnik verwendet eine andere Kodierung der Zeichen, mit den Unterprogrammen CVAC und CVCA erfolgt die Umwandlung (Zeilen 110 bis 218 Bild 3.21). Sie fanden im Monitorprogramm keinen Platz mehr.

Soll am seriellen Ausgang des Grundgerätes Fernschreibtechnik arbeiten, so müssen die Startadressen der Routinen für Eingabe (0C50) und Ausgabe (0C62) als Anwendergerät in die Sprungadressen (siehe I-0-Monitor, Zeilen 835 bis 840) im RAM (!) eingetragen werden. Die Eingabe-Routine ist natürlich nicht nötig, wenn nur gedruckt oder gestanzt werden soll.

Mit den Kommandos IU, OU bzw. LU erfolgt die Zuweisung. Die Schaltung nach Bild 3.18 verwendet einen Schaltkreis SIO, dessen einer Kanal entsprechend Bild 3.19 für Fernschreiber, der andere für Kassettenmagnetband programmiert ist. Der SIO ist einer der interessantesten Schaltkreise und erlaubt die unterschiedlichsten Formen der seriellen Datenübertragung. Er führt die

Umwandlung eines mit dem Prozessor auszutauschenden Zeichens in die serielle Form einschließlich Erzeugung bzw. Verarbeitung von Start- und Stopbits vollkommen selbstständig durch, unabhängig von der programmierten Übertragungsgeschwindigkeit benötigt der Computer für die Ausgabe oder Eingabe eines Zeichens nur die Zeit für den Ablauf weniger Befehle (Zeilen 82 bis 98 von Bild 3.21). Aus der Sicht des Anwenders ist damit die Übertragung von ASCII-Zeichen mehr als zehnmal so schnell möglich wie mit dem PIO, ohne daß es zu Zeitproblemen kommt. Für die Sicherheit der Abspeicherung auf handelsüblichen Kassettenrecorder spielt allerdings das Modulationsverfahren, d. h. die Umwandlung der Datenbits in hörbare Signale, eine Rolle. Je höher die Übertragungsgeschwindigkeit, um so geringer wird bei Tastung eines Tongenerators die Anzahl der NF-Schwingungen je Bit, und desto höhere Anforderungen ergeben sich für Recorder und Bandmaterial. Dies gilt natürlich auch bei anderen Modulationsverfahren ganz allgemein.

Tabelle 3.2. erläutert die Anschlüsse des Schaltkreises *U 856 SIO*. Bild 3.22 zeigt ein Übersichtsschaltbild der inneren Struktur. Der Schaltkreis enthält eine größere Anzahl von Registern, mit deren Hilfe seine Arbeitsweise programmierbar ist, als die bisher vorgestellten Peripheralschaltkreise. Tabelle 3.3. gibt eine Übersicht.

Die Schaltung nach Bild 3.18 arbeitet auf Kanal A mit 300, auf Kanal B mit 50 Baud

3. Erweiterungen

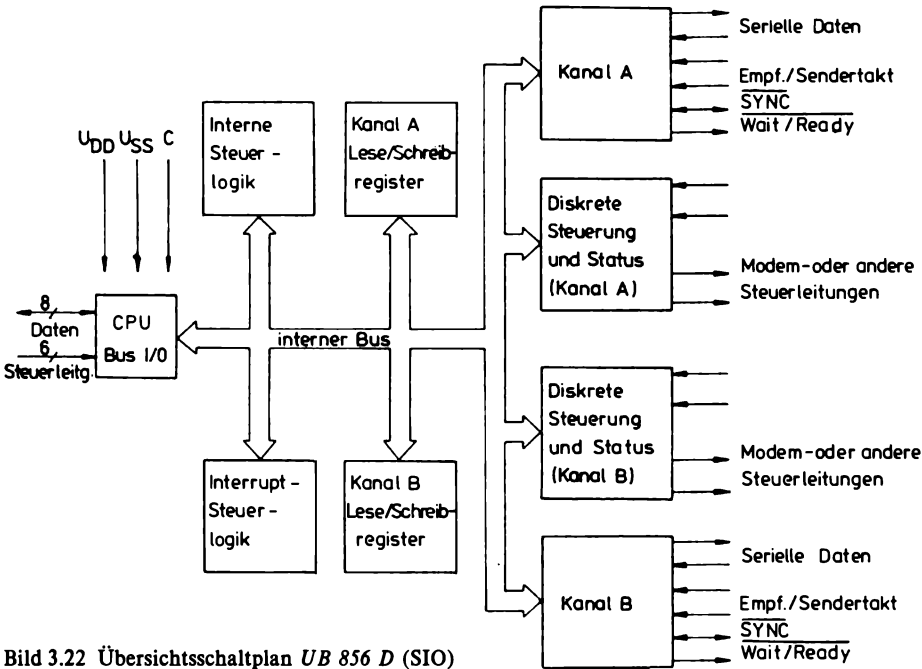


Bild 3.22 Übersichtsschaltplan UB 856 D (SIO)

Tabelle 3.2. Anschlußbezeichnungen U 856 SIO

Signalname	Signalbedeutung
D0 bis D7	Datenbus: bidirektionale Tri-State-Datenleitungen zum Anschluß an den Systemdatenbus
B/ $\overline{A}$	Kanal «B/A-Select»: Eingang zur Kanalauswahl (High bedeutet «Kanal B selektiert»)
C/ $\overline{D}$	«Control/Data-Select»: Eingang Steuerwort/Daten-Übertragung (High bedeutet Steuerwort übertragen)
$\overline{CE}$	«Chip enable»: Eingang, Low-aktiv: Chip-Freigabe
M1	«Machine Cycle 1»: Eingang, Low-aktiv: Maschinenzyklus M1 oder CPU-Steuerbus-signal
$\overline{IORQ}$	«I/O-Request»: Eingang, Low-aktiv: Ein-/Ausgabeeanforderung von der CPU
RD	«Read»: Eingang, Low-aktiv: Lesezyklus der CPU (Steuerbussignal)
C	«Clock»: Eingang: Systemtakt
INT	«Interruptrequest»: Ausgang, Low-aktiv: Interruptanforderung an CPU
IEI	«Interrupt enable In»: Eingang, High-aktiv: Verbindung von IEI mit IEO des nächst höher priorisierten E/A-Schaltkreises ermöglicht Interruptprioritäts-Kaskadierung. High-Pegel an IEI bedeutet, daß momentan kein Interrupt höherer Priorität abgearbeitet wird.
IEO	«Interrupt enable Out»: Ausgang, High-aktiv: IEO führt nur High-Pegel, wenn vom betreffenden und allen höher priorisierten Schaltkreisen der Interruptprioritätskette kein Interrupt in Abarbeitung befindlich oder angemeldet ist. (Ausnahme: noch nicht bestätigte Interruptmeldung eines höher priorisierten E/A-Schaltkreises während ED-Dekodierung)

Signalname	Signalbedeutung
RESET	«Reset»: Eingang, Low-aktiv: Sperrt sowohl Sender als Empfänger. TxDA und TxDB werden in den aktiven Zustand gebracht, Modem Control in den High-Zustand. Nach dem Rücksetzen müssen die Steuerregisterinformationen neu eingeschrieben werden, bevor irgendeine Datenübertragung stattfindet. Sämtliche Interrupts werden gesperrt.
W/RDYA W/RDYE	«Wait/Ready»: Ausgang programmierbar, Open-Drain oder Gegentaktausgang 1 Anschluß pro Kanal, der als «Ready-Leitung» für den Anschluß von DMA-Controllern oder als «Wait-Leitung» zur Synchronisation der CPU mit der Datenübertragungsrate programmiert werden kann.
CTSA CTSB	«Clear to Send»: 1 Anschluß pro Kanal, Schmitt-Trigger-Eingänge, Low-aktiv: In der Betriebsart «Auto-Enable» sperrt dieses Signal den Sender seines Kanals. Wird der Anschluß nicht zur Senderfreigabe verwendet, steht er als allgemeiner Eingang zur freien Verfügung. Die beiden Leitungen haben Schmitt-Trigger-Eingänge, so daß auch Eingangssignale geringer Flankensteilheit einwandfrei verarbeitet werden.
DCDA DCDB	«Data Carrier Detect»: 1 Anschluß pro Kanal, Schmitt-Trigger-Eingänge, Low-aktiv: Im «Auto-Enable»-Mode Eingang zur Empfängerfreigabe, sonst frei verfügbar
RxDA RxDB	«Receive Data»: 1 Anschluß pro Kanal, Eingänge, High-aktiv: serielle Daten
TxDA TxDB	«Transmit Data»: 1 Anschluß pro Kanal, Ausgänge, High-aktiv: serielle Daten
RxCA RxCB	«Receiver Clock»: 1 Anschluß pro Kanal, Schmitt-Trigger-Eingänge, Low-aktiv: Empfängertakteingang, als Taktgeschwindigkeit in der asynchronen Betriebsart können x1, x16, x32 oder x64 programmiert werden.
TxCA TxCB	«Transmitter Clock»: 1 Anschluß pro Kanal, Schmitt-Trigger-Eingänge Sendertakteingang; als Taktgeschwindigkeit sind hier ebenfalls x1, x16, x32 oder x64 möglich.
RTSA RTSB	«Request to Send»: 1 Anschluß pro Kanal, Ausgänge, Low-aktiv: Sobald das RTS-bit eines Kanals gesetzt ist, geht die zugehörige RTS-Leitung in den Low-Zustand über. Wird das RTS-bit in der asynchronen Betriebsart rückgesetzt, geht die zugehörige RTS-Leitung in den High-Zustand, sobald das Senderregister leer ist. In der synchronen Betriebsart fungiert der Anschluß einfach als Ausgang, an dem dauernd der Wert des RTS-bits liegt.
DTRA DTRB	«Data Terminal Ready»: 1 Kontakt pro Kanal, Ausgänge Low-aktiv: Ausgang, an dem der Wert des DTR-bits liegt.
SYNCA SYNCB	«External Character Synchronisation»: 2 Ein-/Ausgabeanschlüsse, Low-aktiv: Funktion des Anschlusses ist abhängig vom programmierten Übertragungsmodus: 1. Externe Synchronisation: Eingang: Aufbauen des Zeichens beginnt mit der steigenden Flanke von RxC, die auf die fallende Flanke des SYNC-Signals folgt. Eingang darf frühestens 2 Taktzyklen nach der steigenden Flanke von RxC aktiviert werden. Empfehlung: SYNC-Aktivierung nur mit der fallenden Flanke von RxC erfüllt diese Forderung.

(Bits pro Sekunde). Die Übertragungsgeschwindigkeiten liegen jedoch nicht durch die Schaltung fest, sondern durch die programmierten Ausgangsfrequenzen, die der CTC an den SIO gibt, und die programmierten Teilerfaktoren des SIO. Sie sind damit in weiten Grenzen wählbar. Während der Fernschreibkanal das Signal unverändert – lediglich galvanisch getrennt auf Eingang und Ausgang – führt, erfolgt bei Kanal B eine Modulation mit der NF vom CTC. Da dieser die Frequenzen als Übertragimpulse synchron zum Prozessortakt abgibt, bildet ein D-Flip-Flop daraus eine Mäanderwelle, die sich besser aufzeichnen läßt. Die Demodulation besorgt ein nachtriggerbares Monoflop, dessen aktive Zeit etwas länger als eine NF-Periode eingestellt ist.

Bei dem bisher beschriebenen Modulationsverfahren besteht ein Datenbit des seriellen Zeichens aus mehreren Perioden einer NF-Schwingung. Für die Datenaufzeichnung sind jedoch noch einige weitere Verfahren üblich. Häufig arbeiten die Modulatorschaltungen mit zwei Frequenzen, die sie in Abhängigkeit vom Signalpegel erzeugen (Frequenzmodulation), z. B. 2 400 Hz und 1 200 Hz. Da hierbei der Kassettenrecorder ständig ein Signal erhält, kommt es nicht zu Einschwingvorgängen wie beim Verfahren Ton/kein Ton. Die Aufzeichnung gelingt damit auch sicher bei Recordern geringerer Qualität. Manchmal findet man auch Schaltungen vor, bei denen jedes Datenbit durch zwei Frequenzen gekennzeichnet wird und nur die Zeitdauer in Abhängigkeit vom Pegel veränderlich ist. Beispiel: $\frac{1}{2}$ der Bitzeit 1 kHz, $\frac{2}{3}$ der Zeit 2 kHz entspricht L-Pegel, bei H-Pegel umgekehrt [16].

Stark verbreitet sind auch Schaltungen, die nicht mehrere Schwingungen für ein Datenbit aufzeichnen, sondern nur eine Flanke der Modulationsfrequenz als Signal verwenden (Phasenmodulation). Mit diesem Verfahren ist eine hohe Aufzeichnungsgeschwindigkeit möglich, ohne daß an den Kassettenrecorder höhere Anforderungen bestehen. Bei einer Aufzeichnungsfrequenz von 1 200 Hz, die mit Sicherheit jeder Recorder bringt, kann man damit 1 200 Bit/s aufzeichnen. Allerdings steigen die Anforderun-

gen an das Bandmaterial und den Recorder. Fehler in der Magnetisierungsschicht bewirken anders als bei den zuvor genannten Verfahren sofort Datenfehler. Bei blockweiser Übertragung der Daten ist dieses Verfahren trotz der hohen Geschwindigkeit auch ohne SIO oder andere aufwendige Hardware realisierbar [9], [14], [15].

3.5. Handleser für Lochstreifen

Obwohl sich Kassettenmagnetbandgeräte als externe Speicher für Programme und Daten allgemein durchgesetzt haben, befindet sich auch noch Lochbandtechnik im Einsatz. Vor allem bei kurzen Programmen ist der schnellere Zugriff wertvoll, ein Lochstreifen ist schneller zu finden als ein Programm hinten auf der Kassette. Die Unterschiede zwischen den einzelnen Computersystemen lassen sich leichter überbrücken, denn auf dem Lochband kann man den Unterschied zum eigenen System ohne Hilfsmittel sehen und, wenn nötig, ein kurzes Anpassungsprogramm schreiben. Die Abspeicherung ist sicherer als auf magnetischen Informationsträgern, denn selbst ein gerissener Lochstreifen kann so geklebt werden, daß man ihn fehlerfrei einlesen kann. Aus all diesen Gründen erscheint der Aufbau einer einfachen Leseeinrichtung für Lochstreifen sinnvoll.

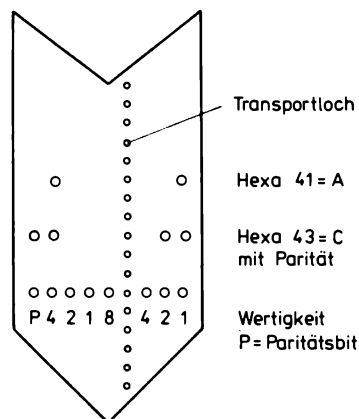


Bild 3.23 Kodierung auf 8-Kanal-Lochstreifen

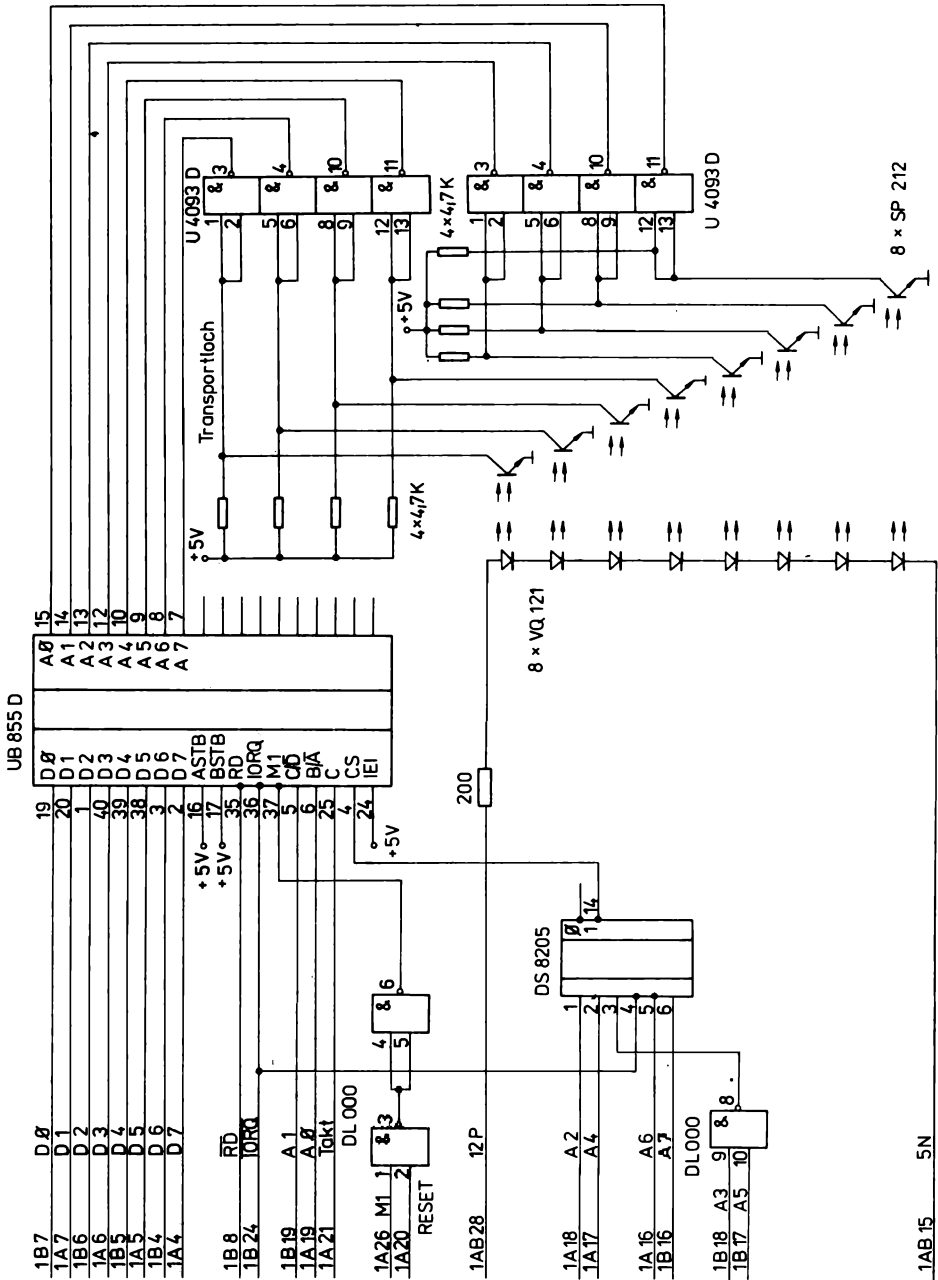


Bild 3.25 Handleser für Lochstreifen

Bild 3.24 zeigt das Layout einer Leiterplatte für 2 Schaltkreise *U 855 PIO*. Die Leiterplatte ist universell anwendbar, für den Lochstreifenleser wird nur ein Port benötigt, so daß sie nicht mit 2 PIO-Schaltkreisen bestückt werden muß. Die Signale RESET und $\overline{M1}$ sind ebenso wie bei der CPU-Karte zu verknüpfen. Um die Karte universell einsetzen zu können, sollte man die Schaltkreisadressen vollständig dekodieren. Bild 3.23 zeigt die Anordnung der ASCII-Zeichen auf Lochstreifen, Bild 3.25 zeigt die Abtastschaltung. Sie arbeitet mit Infrarotemitterdioden und passenden Fototransistoren mit nachgeschalteten Schmitt-Trigger. Die Synchronisation im Leseprogramm erfolgt mit dem Transportloch, welches auf Bit 7 geschaltet ist. Dieses Loch ist kleiner als die Datenlöcher, so daß jene mit Sicherheit alle erfaßt werden. Es sei nicht verschwiegen, daß man die Mechanik eines solchen Lesers recht genau ausführen muß.

3.6. Anschluß des D/A-Wandlers *C 5658 D*

Eine sehr interessante und zukunftssträchtige Angelegenheit ist die Computergrafik. Zur Ausgabe von Kurven und ähnlichem auf übliche x-y-Schreiber müssen die digitalen Ergebnisse des Computers mittels Digital-

/Analogwandler umgeformt werden. Die Ankopplung kann ebenfalls mit der 2-PIO-Karte erfolgen. Bild 3.26 zeigt die Schaltung. Sie ist so ausgelegt, daß auch die 12-bit-Typen dieses D/A-Wandlers eingesetzt werden können. Die Ausgabe eines Digitalwertes erfolgt dann auf 2 PIO-Ports.

3.7. Anschluß des A/D-Wandlers *C 520 D*

Für viele Amateure ist sicherlich der Anschluß eines A/D-Wandlers an den Computer interessant. Bild 3.27 zeigt eine Schaltung mit PIO. Der A/D-Wandler arbeitet im Multiplexbetrieb mit BCD-Code an den Ausgängen und wurde für die Zusammenschaltung mit einem Decoder entworfen. Die Anschaltung an einen Computer erfordert deshalb ein kleines Programm, zu dem einige Erläuterungen sinnvoll erscheinen.

Das Zeitdiagramm (Bild 3.28) zeigt die Multiplexsteuerung der Signale an den BCD-Ausgängen.

Wenn ein Multiplexausgang auf L-Pegel schaltet, steht an den BCD-Ausgängen die entsprechende Stelle des Meßwertes (MSD = höchstwertige, LSD = geringwertigste Stelle). Der Computer fragt in einer Programmschleife die Multiplexerausgänge ab und übernimmt nacheinander die einzelnen Stellen eines Meßwertes.

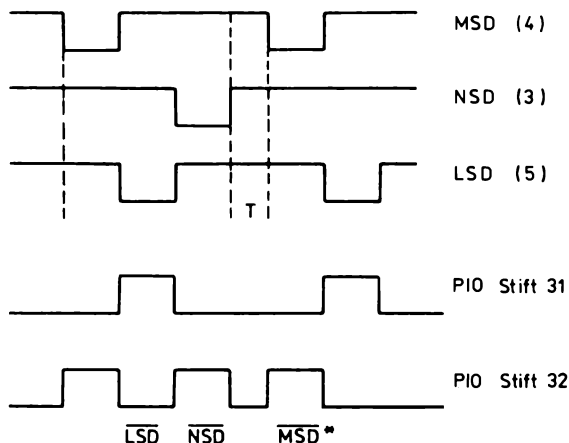


Bild 3.28
Zeitdiagramm des Schaltkreises
C 520 D

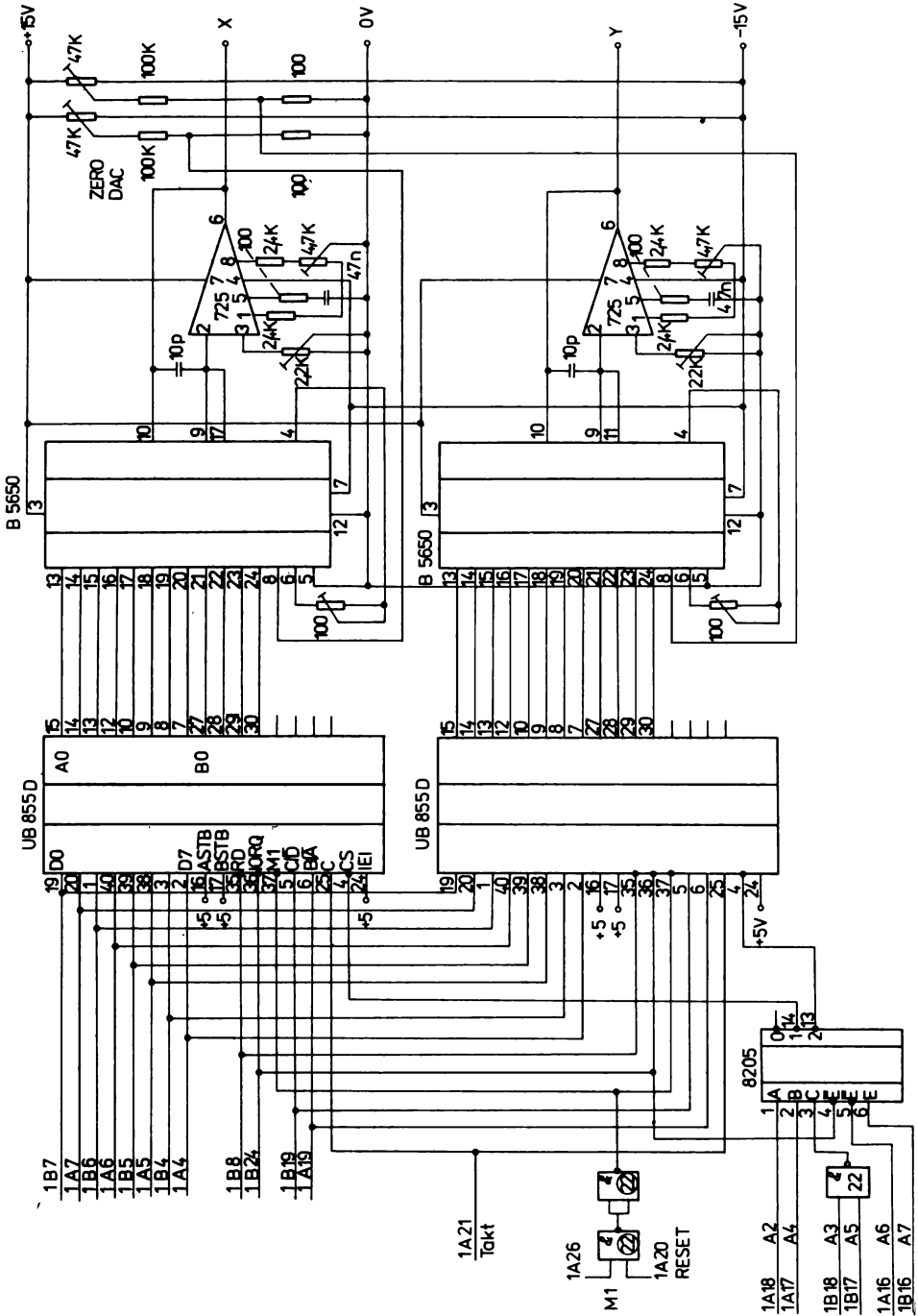


Bild 3.26 Anschlußsteuerung für D/A-Wandler

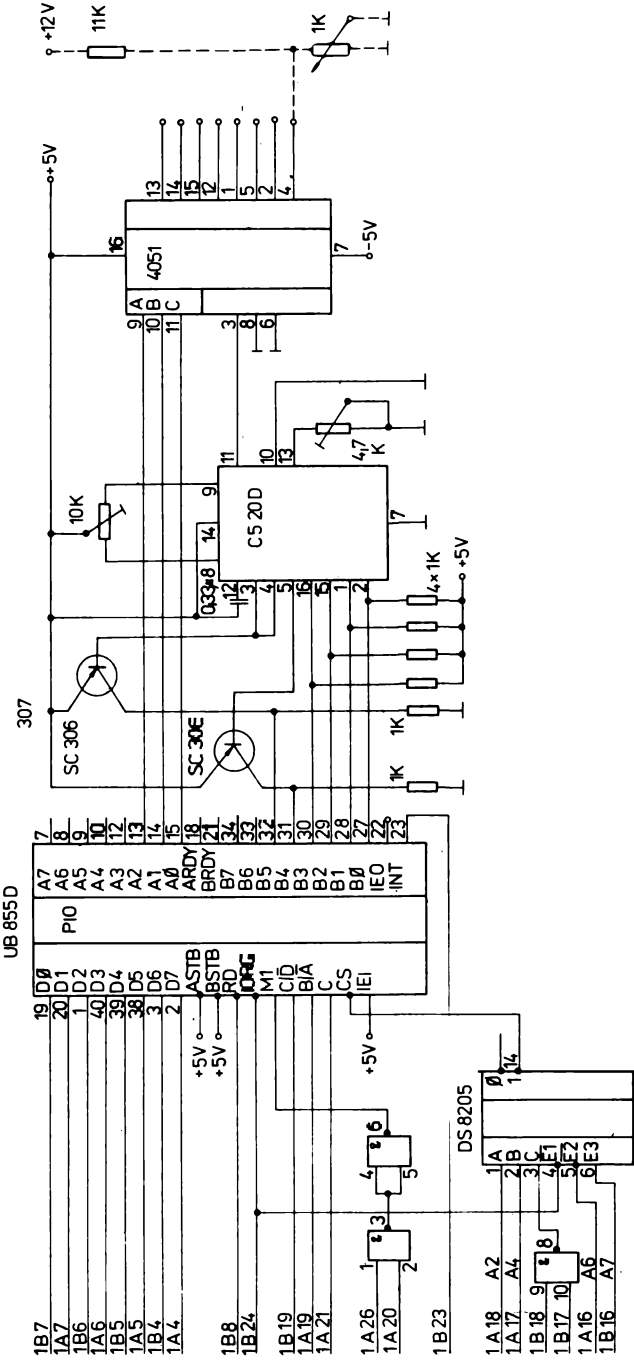


Bild 3.27 Anschluß eines A/D-Wandlers C 520 D

Da die Reihenfolge der Multipleximpulse festliegt, können zwei der Ausgänge zusammen geschaltet werden. Diese Ausgänge weisen zwar offene Kollektoren auf, liefern jedoch wegen interner Kollektorwiderstände kein «vorschriftsmäßiges» Logiksignal, wenn man sie nur mit einem Widerstand gegen +5 V schaltet. Deshalb folgen noch pnp-Transistoren, bevor die Impulse auf das PIO gelangen. Das Programm (Bild 3.29) wandelt die BCD-Werte durch Addition von 30H in ASCII-Zeichen um. Sollen mehrere Meß-

stellen wahlweise am A/D-Wandler arbeiten, so kann das verbleibende PIO-Port einen einfachen Meßstellenumschalter steuern. Bild 3.30 zeigt die Umschaltung einiger Temperaturmeßstellen mit dem CMOS-Multiplexer 4051. Da die Multiplexer interne Dekoder enthalten, beschränkt sich die Umschaltung auf einen Ausgabebefehl je Meßstelle. Nach der Umschaltung müssen Verstärker und A/D-Wandler erst auf den neuen Wert einschwingen, die Messung darf deshalb erst einige Millisekunden später stattfinden.

```

EBAS-88
LOC   CODE   LINE   SOURCE
                                00001      PN   AD
                                00002      ;
                                00003      ;PROGRAMM FUER A/D-WANDLER C520
                                00004
                                00005      ORG   OC50H
                                00006      PIOBC: EQU   OAPH           ;STEUERPORT
                                00007      PIOBD: EQU   OADH           ;DATENPORT
                                00008      BRAM:  EQU   OFCOOH        ;ANFANG BILDWIEDERHOLSPESICHER
                                00009
                                00010      AN:   LD    HL,BRAM+03DH      ;ANZEIGEPLATZ AUF BILDSCHIRM
                                00011      IN    PIOBD
                                00012      AND   10H
                                00013      JRZ   AN-#
                                00014      CALL  MW
                                00015      ADU2:  IN    PIOBD
                                00016      AND   20H
                                00017      JRZ   ADU2-#
                                00018      CALL  MW
                                00019      ADU3:  IN    PIOBD
                                00020      AND   20H
                                00021      JRNZ  ADU3-#
                                00022      ADU4:  IN    PIOBD
                                00023      AND   20H
                                00024      JRZ   ADU4-#
                                00025      CALL  MW
                                00026      RET
                                00027      ;-----
                                00028      ;INITIALISIERUNG PIO
                                00029      ADINI: LD    A,OCFH           ;BIT-MODE
                                00030      OUT   PIOBC
                                00031      LD    A,3FH
                                00032      OUT   PIOBC
                                00033      LD    A,7
                                00034      OUT   PIOBC
                                00035      RET
                                00036      ;-----
                                00037      ;MESSWERTERFASSUNG
                                00038      ;AUSGELOST DURCH MULTIPLEXER-IMPULSE DES C520
                                00039      ;IN:  HL-AKT.BILDSCHIRMPLATZ
                                00040      ;OUT: MESSWERT ALS ASCII-ZEICHEN IN A UND AUF BILDSCHIRM
                                00041      ;      HL=HL-1
                                00042      ;KILL: AF,
                                00043      MW:   CALL  ZS
                                00044      IN    PIOBD
                                00045      AND   0FH
                                00046      OR    30H
                                00047      LD    (HL),A
                                00048      DEC   HL
                                00049      RET
                                00050      ;-----
                                00051      ;ZEITSCHLEIFE ZUR VERZOEGERTEN ABFRAGE MITTE SYNCHRONIMPULS
                                00052      ZS:   LD    A,08H
                                00053      ZS1:  DEC   A
                                00054      JRNZ  ZS1-#
                                00055      RET
                                00056      END
0C50   213DFC   00010      AN:   LD    HL,BRAM+03DH      ;ANZEIGEPLATZ AUF BILDSCHIRM
0C53   DBAD    00011      IN    PIOBD
0C55   E610    00012      AND   10H
0C57   28F7    00013      JRZ   AN-#
0C59   CD820C  00014      CALL  MW
0C5C   DBAD    00015      ADU2:  IN    PIOBD
0C5E   E620    00016      AND   20H
0C60   28FA    00017      JRZ   ADU2-#
0C62   CD820C  00018      CALL  MW
0C65   DBAD    00019      ADU3:  IN    PIOBD
0C67   E620    00020      AND   20H
0C69   20FA    00021      JRNZ  ADU3-#
0C6B   DBAD    00022      ADU4:  IN    PIOBD
0C6D   E620    00023      AND   20H
0C6F   28FA    00024      JRZ   ADU4-#
0C71   CD820C  00025      CALL  MW
0C74   C9      00026      RET
                                00027      ;-----
0C75   3ECF    00028      ;INITIALISIERUNG PIO
0C77   D3AF    00029      ADINI: LD    A,OCFH           ;BIT-MODE
0C79   3E3F    00030      OUT   PIOBC
0C7B   D3AF    00031      LD    A,3FH
0C7D   3E07    00032      OUT   PIOBC
0C7F   D3AF    00033      LD    A,7
0C81   C9      00034      OUT   PIOBC
                                00035      RET
                                00036      ;-----
                                00037      ;MESSWERTERFASSUNG
                                00038      ;AUSGELOST DURCH MULTIPLEXER-IMPULSE DES C520
                                00039      ;IN:  HL-AKT.BILDSCHIRMPLATZ
                                00040      ;OUT: MESSWERT ALS ASCII-ZEICHEN IN A UND AUF BILDSCHIRM
                                00041      ;      HL=HL-1
                                00042      ;KILL: AF,
0C82   CD8E0C  00043      MW:   CALL  ZS
0C85   DBAD    00044      IN    PIOBD
0C87   E60F    00045      AND   0FH
0C89   F630    00046      OR    30H
0C8B   77      00047      LD    (HL),A
0C8C   2B      00048      DEC   HL
0C8D   C9      00049      RET
                                00050      ;-----
                                00051      ;ZEITSCHLEIFE ZUR VERZOEGERTEN ABFRAGE MITTE SYNCHRONIMPULS
0C8E   3E08    00052      ZS:   LD    A,08H
0C90   3D      00053      ZS1:  DEC   A
0C91   20FD    00054      JRNZ  ZS1-#
0C93   C9      00055      RET
0C94                                00056      END

```

Bild 3.29 Programm für A/D-Wandler

4. Weitere Mikrocomputerbauelemente und Geräte

Neben den bisher vorgestellten Bauelementen gibt es eine Vielzahl interessanter Schaltkreise, die der Amateur zum Aufbau von Mikrocomputern heranziehen kann.

Vor allem sind dies weitere programmierbare Peripherieschaltkreise, die aus anderen Mikroprozessorsystemen stammen. Aber auch Interfaceschaltkreise und die Vielzahl von Speicher-IS können oft vorteilhaft zur Lösung von schaltungstechnischen Problemen beitragen.

Völlig andere Möglichkeiten ergeben sich bei Einsatz anderer Prozessoren. Mit dem Sortiment der Einchip-Mikrocomputer kann man Minimalsysteme aufbauen, die ein Gerät steuern (z. B. eine Funkuhr, ein Meßgerät) und nicht erweiterungsfähig sein müssen. Die 16-bit-Prozessoren sind zwar im Amateurbereich nicht «ausgelastet», das trifft jedoch auch bei anderen hochintegrierten Schaltkreisen immer öfter zu. Einen Zweck kann man als Amateur mit Sicherheit durch die Beschäftigung mit diesen Bauelementen erreichen: Man wird nicht dümmer.

4.1. Peripherieschaltkreise

Für andere 8-bit-Mikroprozessorsysteme gibt es programmierbare Peripherieschaltkreise, die auch in Verbindung mit dem *UB 880 D* einsetzbar sind. Sie werden zwar in der DDR nicht gefertigt, stehen der Industrie jedoch durch Importe aus der UdSSR und den anderen sozialistischen Ländern zur Verfügung und sind auch manchmal im Amateurbedarfshandel im Angebot. Weit verbreitet sind Bauelemente, die zum Vorgänger des *UB 880*, dem Mikroprozessor *8080* und dessen Weiterentwicklung *8085* gehören.

Beim Einsatz dieser Schaltkreise ist die Begrenzung der Arbeitsgeschwindigkeit zu

beachten. Zwar läuft die Schaltung nach Bild 1.1 mit einer Taktfrequenz von 1,5 MHz, aber der *UB 880 D* kann mit maximal 2,5 MHz arbeiten. Bei den Schaltkreisen des Mikroprozessors *8080* sind nur 2 MHz zulässig. Der Schaltkreis *8255* entspricht anwendungsseitig dem *UB 855 D* (PIO). Er enthält jedoch 3 programmierbare 8-bit-Kanäle für parallele Ein- und Ausgabe. Bild 5.53 zeigt den Übersichtsschaltplan.

Der *8255* verfügt über 6 Steuerleitungen. H-Pegel (!) am Anschluß RESET (35) bringt den Schaltkreis in einen definierten Anfangszustand, bei dem alle internen Register rückgesetzt sind und die Kanäle A, B und C in die Betriebsart Eingabe schalten. Mit L-Pegel am Eingang $\overline{CS}$ (6) kann man den Schaltkreis anwählen. Soll der Prozessor Daten vom *8255* lesen, muß der Anschluß RD (5), beim Schreiben der Anschluß WR (36), auf L-Pegel liegen. Mit Hilfe der Leitungen A0 und A1 erfolgt die Auswahl der Kanäle und des Steuerregisters. Tabelle 4.1. erläutert die möglichen Betriebsweisen. Die letzte Spalte zeigt, daß man das Steuerregister nur beschreiben kann.

Tabelle 4.1. Betriebsarten des 8255

Pegel					Funktion
A0	A1	RD	WR	$\overline{CS}$	
0	0	0	1	0	Eingabe von Kanal A
0	1	0	1	0	Eingabe von Kanal B
1	0	0	1	0	Eingabe von Kanal C
0	0	1	0	0	Ausgabe auf Kanal A
0	1	1	0	0	Ausgabe auf Kanal B
1	0	1	0	0	Ausgabe auf Kanal C
1	1	1	0	0	Ausgabe auf Steuerport
X	X	X	X	1	Schaltkreis nicht ausgewählt
1	1	0	1	0	nicht zulässig

Bei der Programmierung der Kanalleitungen unterscheidet man 3 Betriebsarten, wobei der Kanal C je zur Hälfte den anderen beiden zugeordnet wird.

Betriebsart 0 bedeutet einfache Ein-/Ausgabe. Dabei sind alle Leitungen als Datenleitungen für Ausgabe oder Eingabe programmierbar.

Bei Betriebsart 1 arbeiten nur die Kanäle A und B als Datenports, wobei Eingabe oder Ausgabe kanalweise programmierbar ist. Der Kanal C liefert bei dieser Betriebsart Steuerleitungen für Quittungsbetrieb mit

dem angeschlossenen Gerät. In Betriebsart 2 kann Kanal A bidirektional, wie ein Datenbus arbeiten. Von Kanal C dienen 5 Leitungen für Steuerzwecke. Die Bilder 5.50 und 5.51 zeigen die Zuordnung der Steuerworte für die Programmierung des 8255.

Ein anderer interessanter Schaltkreis aus diesem System ist der 8251, der in vielen Fällen die Aufgaben des UB 856 D (SIO) erfüllen kann. Er enthält nur einen seriellen Kanal, was aber in vielen Fällen ausreicht (Bild 5.52). Die Leistungsfähigkeit ist auch in anderen Punkten geringer. So entfällt z. B.

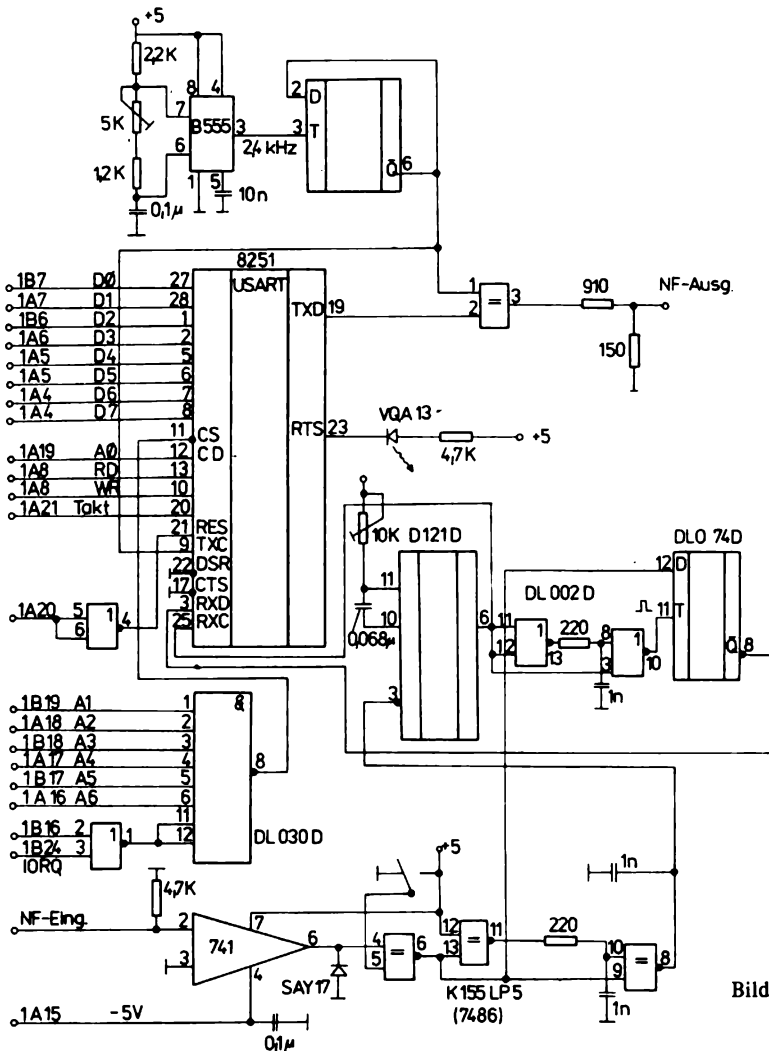
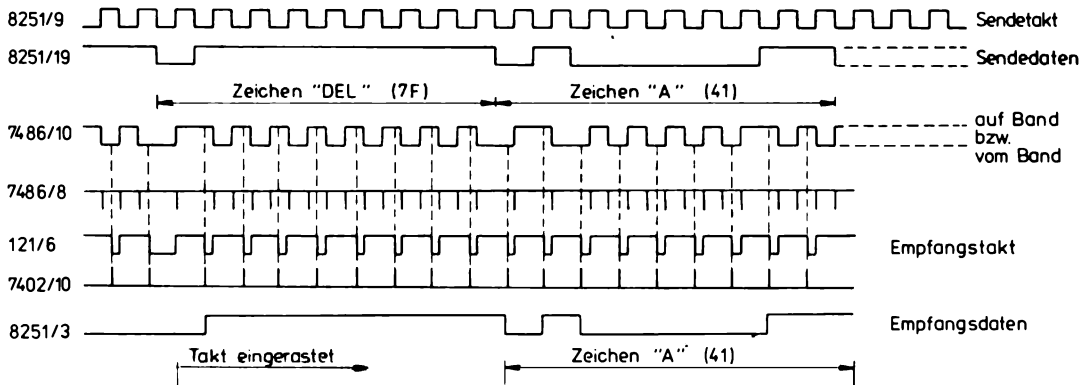


Bild 4.1 Kassetteninterface mit USART 8251; a - Stromlaufplan,

4. Weitere Mikrocomputerbauelemente und Geräte



b – Zeitdiagramme der Phasenmodulation/-demodulation

die Möglichkeit, die synchrone Datenübertragung mit automatischer CRC-Prüfung zu betreiben. Von geringer Bedeutung für den Amateur ist auch die Beschränkung beim Interrupt, denn alle programmierbaren Peripherieschaltkreise des 8080 und 8085 können keinen Interruptvektor abgeben. Für die Praxis des Amateurs ist der 8251 oft besser als der UB 856 D (SIO) geeignet, weil er weniger Aufwand erfordert.

Bild 4.1 zeigt Schaltung und Zeitdiagramm für ein schnelles Kassetteninterface mit diesem Schaltkreis. Das Programm ist in Bild 3.21 mit enthalten. Den Sendetakt liefert in dieser Schaltung der Timer B 555 D, er ist deshalb nicht von der Taktfrequenz des Computers abhängig wie beim UB 857 D (CTC). Die Ausgabe der seriellen Daten geschieht mit der bereits erwähnten Phasenmodulation und einer Geschwindigkeit von ca. 1200 bit/s. Der USART ist dabei auf ein Teilverhältnis von 1:1 für den Datentakt programmiert.

Bei der Ausgabe erfolgt die Phasenmodulation mit Hilfe des Exklusiv-Oder-Gatters, indem die Flanken des Taktsignals in Abhängigkeit vom Datenausgang des USART verschoben werden (siehe Zeitdiagramm). Die Exklusiv-Oder-Funktion kann auch mit 4 NAND-Gattern realisiert werden. Der Demodulator ist etwas aufwendiger. Den Empfangertakt gewinnt man aus den Flanken des abgespielten Signals durch Differenzieren

(Gatterschaltung) und Triggern des monostabilen Multivibrators zurück. Gleichlaufschwankungen des Recorders wirken sich deshalb in gewissen Grenzen nicht aus. Der D 121 D ist auf $\frac{3}{4}$ der Periodendauer der Takt- bzw. Aufzeichnungsfrequenz eingestellt. Mit seinem Ausschalten übernimmt das D-Flip-Flop den Signalpegel des D-Eingangs und gibt ihn an den USART weiter. Manche Kassettenrecorder invertieren das Signal, so daß man mit einem Gatter nach dem Vorverstärker den nötigen Pegel wiederherstellen kann. Den Abgleich der Schaltung nimmt man vorteilhaft mit einem Oszilloskop und einer Brücke zwischen Modulatorausgang und Demodulatoreingang (ohne Kassettenrecorder) vor. Zum Betrieb der Schaltung müssen nach der Initialisierung des USART-Schaltkreises die Anfangsadressen der Programmteile für Ausgabe und Eingabe wie bereits früher beschrieben in die Systemtabelle eingetragen werden. Beim Einlesen von Daten ist das Einlesekommando erst bei Ertönen des Kenn tones (unmoduliertes Signal) vom Band gültig zu machen. Damit der D 121 D nicht mit der falschen Flanke einrastet, wird bei der Ausgabe noch ein kurzer Text vor den eigentlichen Daten ausgegeben, z. B. E0 (CR).

Natürlich arbeitet die gezeigte Modulatorschaltung auch mit anderen Peripherieschaltkreisen für serielle Ausgabe, wie z. B. dem UB 856 D.

Ein weiterer Peripherieschaltkreis aus diesem Mikroprozessor-System ist der 8253, der als programmierbarer Zähler- und Zeitgeberbaustein einige Aufgaben des *UB 857 D* (CTC) übernehmen kann. Wegen der eingeschränkten Möglichkeiten beim Interrupt ist er in Verbindung mit dem *UB 880 D* nicht von Interesse. Gleiches gilt für die Schaltkreise 8257 (Schaltkreis für direkten Speicherzugriff) und 8259 (Prioritätssteuerung). Wichtige Bauelemente sind sogenannte CRT-Controller, mit denen der Elektronikaufwand für eine Bildschirmsteuerung drastisch verringert werden kann. Zum Beispiel ersetzt der Schaltkreis 9364 die Schaltung nach Bild 1.12 komplett. Dabei entfallen die Anschlüsse zum Adreßbus. In der VR Bulgarien wird der CRT-Controller 6845 produziert, bei dem u.a. die Anzahl der Zeichen und Zeilen programmierbar ist. Zum Mikroprozessorsystem 8080 gehört der CRT-Controller 8275, der zusätzlich noch einen DMA-Schaltkreis 8257 in der Anwenderschaltung benötigt.

Die Halbleiterindustrie der DDR produziert leider keinen dieser Typen, obwohl die Stückzahl sicherlich höher liegen würde als bei SIO und DMA.

4.2. Speicher

Auf dem Gebiet der Speicher ist die Entwicklung der Bauelemente besonders beeindruckend. Ein weitverbreiteter Mikroprozessor kann volkswirtschaftlich gesehen nicht so schnell durch einen anderen ersetzt werden, weil die nach einiger Zeit erarbeiteten Programme sehr viel Aufwand erfordert haben und bei einem neuen Typ meist nicht ohne umfangreiche Änderungen nutzbar sind. Neue Typen setzen sich deshalb trotz höherer Leistungsfähigkeit manchmal nur langsam durch.

Bei den Speichern ist dies nicht so, im Gegenteil, bei Ersatz mehrerer Leiterplatten durch eine mit Speichern höheren Integrationsgrades lassen sich beträchtliche wirtschaftliche Effekte erzielen. Deshalb hält die Entwicklung von Speicherschaltkreisen größerer Kapazität unvermindert an. Im Ver-

gleich zu den bisher vorgestellten Typen ist jedoch «nur» die Leistungsfähigkeit, nicht das Arbeitsprinzip anders, so daß auf ausführliche Erläuterungen verzichtet werden kann. Im Abschnitt 5. sind die Anschlüsse einiger neuerer Speicherschaltkreise aufgeführt.

4.3. Interface-Schaltkreise

Das in der Computertechnik übliche Wort «Interface» kann man etwas frei mit «Anpassung» übersetzen. Die Interface-Schaltkreise *DS 8216*, *DS 8212*, *DS 8282*, *DS 8283*, *DS 8286* und *DS 8287* sollen die leistungsschwachen Leitungen der MOS-Schaltkreise an das Bus-System und an äußere Schaltungen anpassen. Um auf dem Bus geringe Signalverzögerungen zu erreichen, werden die Bauelemente nach der Schottky-Technologie gefertigt. Die Interface-Schaltkreise erlauben entsprechend ihrer Aufgabenstellung höhere Ausgangslastfaktoren als übliche TTL-Gatter. Man unterscheidet dabei 2 Typen, die bidirektional arbeitenden und die nur in eine Richtung wirksamen. Erstere sind für den Datenbus nötig, die letzteren für Adreß- und Steuerleitungen. Die Schaltkreise *8216*, *8286* und *8287* arbeiten in beiden Richtungen und werden deshalb oft als Transceiver (Sendeempfänger) bezeichnet. Die Schaltkreise *8212*, *8282* und *8283* enthalten interne D-Flip-Flops, mit denen die Eingangssignale gepuffert werden können. Sie werden häufig als Latch (Zwischenspeicher) bezeichnet. Alle erwähnten Interface-Schaltkreise verfügen über Tristate-Ausgänge. Außer für den eigentlichen Verwendungszweck am Bus-System arbeiten diese Bauelemente auch vorteilhaft in Eingabe- und Ausgabegruppen, wenn man ihnen entsprechende Adressen zuordnet. So kann man z. B. eine Tastaturabfrage auch mit 2 Schaltkreisen *8282* aufbauen. Da die Interface-Schaltkreise nicht programmierbar sind, unterscheiden sie sich in der Anwendung nicht wesentlich von üblichen TTL-Schaltkreisen.

4.4. Einchip-Mikrocomputer

Die Einchip-Mikrocomputer enthalten bereits einen bescheidenen Speicher, sowohl ROM als auch RAM, und verfügen über programmierbare Peripherieleitungen. Im Normalfall programmiert der Hersteller das ROM. Diese Bauelemente sind dadurch relativ preiswert.

Zwar wäre zu vermuten, daß sie nur für den geplanten Einsatz, z. B. in einer Waschmaschine, brauchbar sind, da der Amateur das im ROM untergebrachte Programm und die darauf bezogene Anwenderschaltung nicht kennt. Dem ist aber nicht so. Für die Prüfung des Schaltkreises beim Hersteller gibt es eine Möglichkeit, ihn trotzdem zu betreiben. Bevor jedoch darauf eingegangen wird, soll zunächst ein kurzer Überblick über das Sortiment der Einchip-Mikrocomputer aus DDR-Produktion folgen.

Für die Schaltungsentwicklung gibt es den 882, bei dem zum Anschluß von Speicherschaltkreisen Adreß- und Steuerleitungen zusätzlich herausgeführt sind. Diese Bauelemente sind auch für den Amateur sehr interessant, erlauben sie es doch, mit geringem Aufwand funktionsfähige Mikrocomputer aufzubauen. Auf zwei Schwierigkeiten sei jedoch hingewiesen:

- Die Einchip-Mikrocomputer arbeiten dynamisch wie die Speicher U 256, d. h., der Takt muß stets anliegen, sonst gehen die

Informationen verloren. Da außerdem ein WAIT-Anschluß fehlt, ist die Fehlersuche im Schrittbetrieb nicht so einfach zu realisieren wie beim Mikroprozessor UB 880 D.

- Hilfsmittel für die Programmentwicklung (Monitor, Assembler, Disassembler) sind entweder nur mit hohem Aufwand nutzbar (64-k-Computer mit Floppy-Disk-Speicher) oder fehlen zur Zeit völlig.

Bild 4.2 enthält die Bezeichnungssystematik der Einchip-Mikrocomputer, die in der DDR gefertigt werden. Bild 4.3 zeigt die interne Struktur des UB 8810 D. Die 32 programmierbaren Ein-/Ausgabeleitungen sind in 4 Gruppen zu je 8 organisiert (Port 0 bis 3). Die Ports 0 und 1 können auch zur Erweiterung des bescheidenen internen Speichers von 2 kByte ROM und 124 Byte RAM dienen. Da der Befehlszähler mit 16 bit arbeitet, sind maximal 64-k-Byte-Programmspeicher und zusätzlich 62-k-Datenspeicher direkt adressierbar. Allerdings arbeiten die Ports 0 und 1 dann im Multiplexbetrieb, d. h., Daten und Adressen folgen nacheinander auf den gleichen Leitungen. Dafür sind äußere Zwischenspeicher erforderlich. Port 2 kann man als übliche programmierbare Parallelschnittstelle nutzen, d. h., die Leitungen können bitweise zur Eingabe oder Ausgabe von Signalen dienen. Die Leitungen von Port 3 arbeiten als eigenständige Ein-/Ausgabe oder als Steuerleitungen (Quittungsbe-

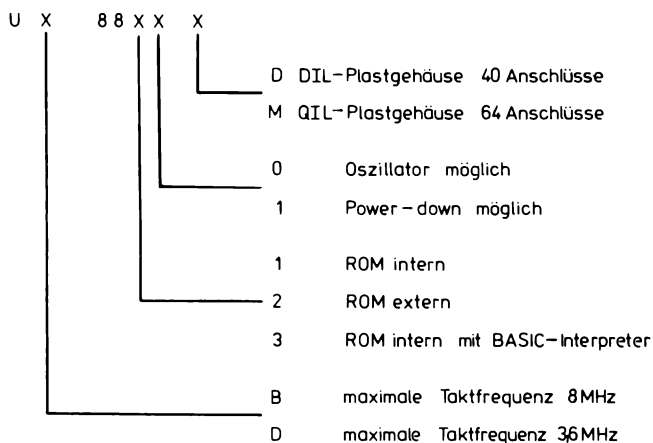


Bild 4.2
Bezeichnungssystematik der
Einchip-Mikrocomputer

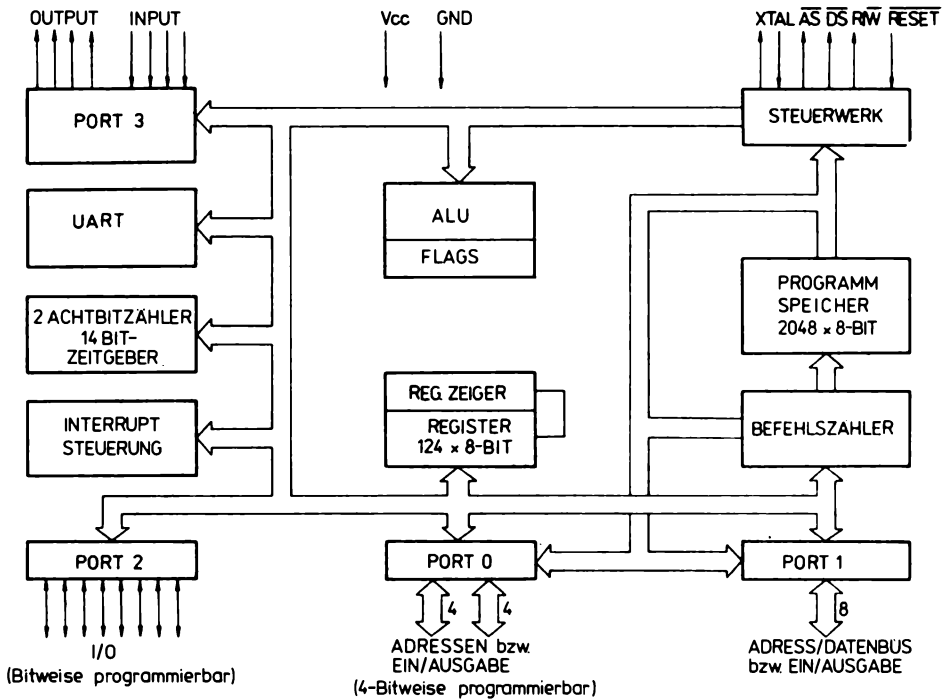


Bild 4.3 Struktur des UB 8810 D

trieb für Port 2). Das Bauelement enthält eine Schaltung zum Senden und Empfangen asynchroner, serieller Signale. Die Datenübertragungsgeschwindigkeit wird mit der ebenfalls integrierten Zähler-/Zeitgeber-schaltung gesteuert.

Für Programmunterbrechungen ist eine Steuerung vorhanden. Der Arbeitsspeicher wurde im Bild nicht als RAM, sondern als Registerblock dargestellt, um anzudeuten, daß jeder dieser Speicherplätze als Akkumulator dienen kann. Der Registerzeiger adressiert die Registergruppen. Trotz der bescheidenen Kapazität ist der Speicher damit sehr effektiv und flexibel nutzbar. Auch die Pufferspeicher für die Ports liegen mit ihren Adressen im Registerblock. Neben dem Flag-Register existieren natürlich noch weitere Register, die für die Steuerung des Programms nötig sind. Bild 5.59 enthält dazu eine Übersicht. Die Arithmetik-Logik-Einheit (ALU) arbeitet mit 8 bit parallel, wie dies auch beim UB 880 D der Fall ist. Eine

Besonderheit ist die Abarbeitung der Befehle im sogenannten Pipeline-Verfahren, wobei während der Ausführung eines Befehls im Prozessor bereits der nächste vom Speicher geholt wird. Die Ausführungszeit verkürzt sich dadurch etwas. Für den Test beim Hersteller enthält jeder Schaltkreis außer dem 2-k-Byte-Anwenderprogramm noch ein zusätzliches ROM, das den Betrieb mit externem Speicher gestattet. Dabei werden die Ports 0 und 1 als Daten- und Adreßleitungen programmiert, und der Computer beginnt sein Programm auf Adresse 812H. Diese Testbetriebsart ist für den 886 die einzig vorgeschriebene und damit für den Amateur interessant. Sie schaltet ein, wenn man an den RESET-Eingang des Schaltkreises eine Spannung von ca. 6,5 V legt.

4.5. 16-bit-Mikroprozessoren

Die Mikroprozessoren *U 8001* und *U 8002* arbeiten mit 16 bit breitem Datenbus. Während der *UB 880 D* ungefähr 8 200 Transistoren enthält, haben die Typen *8001/8002* etwa 17 500 integriert. Entsprechend leistungsfähig ist auch der Befehlssatz. Er besteht aus 414 Grundbefehlen, u. a. für Multiplikation, Division, relativen CALL, Vergleich von Strings und statische oder dynamische bit-Manipulation.

Bei den Typen *8001/8002* befindet sich der 16-bit-Prozessor auf einem Siliziumscheibchen (Chip). Der Schaltkreis *U 830 D* stellt eine 8-bit-CPU dar, die durch Parallelschaltung von 2 Schaltkreisen ebenfalls mit 16 bit arbeiten kann.

Die Anwendung von 16-bit-Prozessoren ist z. Z. sicherlich nur für wenige Amateure interessant. Deshalb soll darauf vorläufig nicht näher eingegangen werden. Die weitere Entwicklung bei diesen Bauelementen ist allerdings mit großem Interesse zu verfolgen. In Zukunft wird mit einem verstärkten Einsatz dieser Bauelemente vor allem in der industriellen Steuerungstechnik gerechnet.

4.6. Folienspeicher (Floppy-Disk)

Je umfangreicher die Programme und je größer ihre Anzahl, desto wichtiger ist es, sie übersichtlich auf externen Datenträgern zu speichern und schnell in den Computer laden zu können. Obwohl sicherlich für die meisten Amateure aus Kostengründen das Kassettenmagnetband auch in Zukunft das

optimale Speichermedium bleiben wird, soll eine kurze Übersicht die Entscheidung für oder gegen die Folienspeichertechnik erleichtern helfen. Der Folienspeicher ist eine runde, flexible Scheibe mit magnetisierbarer Schicht, die in einer ausgekleideten Papiertasche rotiert. Übliche Bezeichnung sind auch «Floppy-Disk» oder «Diskette». Standard-Disketten haben 8 Zoll, Minidisketten 5,25 Zoll und die in letzter Zeit aufgekommenen Mikrodisketten 3,5 Zoll Durchmesser (Bild 4.4). Die Speicherkapazität hängt von der Größe der Diskette, aber auch vom Aufzeichnungsverfahren ab. Das Minidiskettenlaufwerk K 5600.10 von *robotron* erlaubt z. B. die Speicherung in 40 konzentrischen Spuren mit je 16 Sektoren zu 256 Byte. Die Übertragungsrate beträgt 250 kbit/s, die Drehzahl der Diskette 300 U/min, die Schrittzeit von Spur zu Spur 8 ms. Bei den Disketten kann die Bestimmung der Sektoren mit optoelektronisch abgetasteten Löchern in der Diskette für jeden Sektor erfolgen, man spricht dann von Hard-Sektor-Disketten. Bei den Soft-Sektor-Disketten übernimmt ein Programm die Zuordnung, indem zwischen die Sektoren mit den Datenbytes «Lücken» geschrieben werden. Zur Synchronisation ist dann nur ein Index-Loch nötig. Die Bewegung des Magnetkopfes erfolgt mit einem Schrittmotor. Zur Steuerung des Laufwerks sowie des Speichervorgangs ist eine umfangreiche Elektronik erforderlich, die etwa einer Computerminimalkonfiguration entspricht. Für den Anwender bringen jedoch die große Speicherkapazität und der schnelle Zugriff eine beträchtliche Steigerung der Leistungsfähigkeit und des Komforts mit sich. Um möglichst große Flexibili-



Bild 4.4
Folienspeicher

Spurenaufbau:

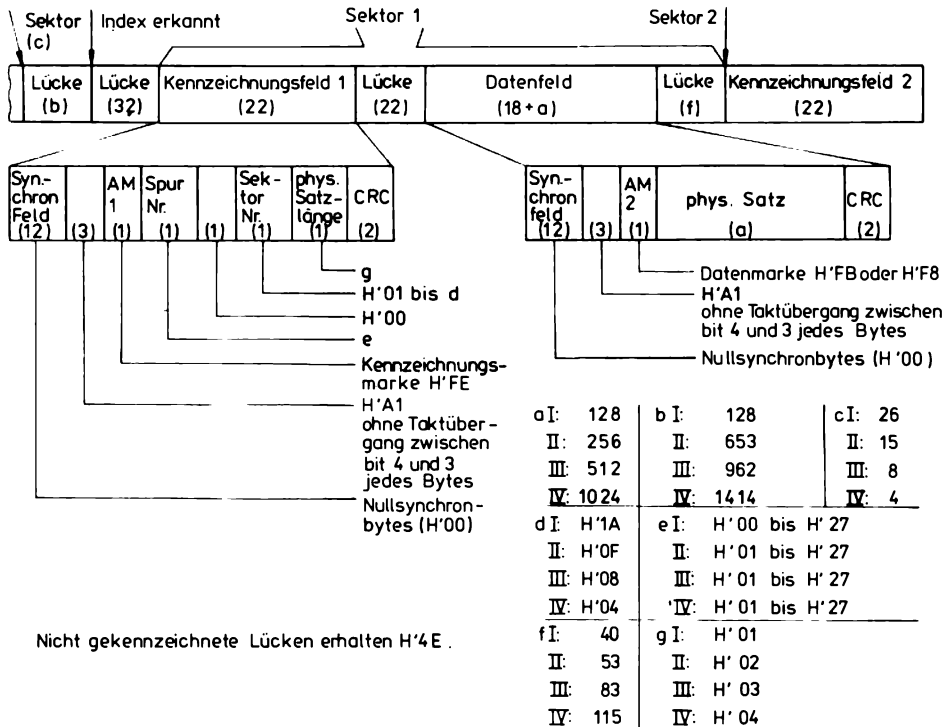


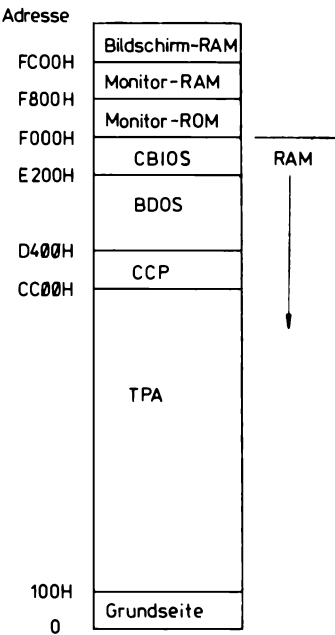
Bild 4.5 Aufteilung eines Sektors in einer Diskettenspur

tät zu erreichen, legt man meist den gesamten Speicher des Computers als RAM aus. Anstelle des Monitors tritt ein sogenannter Urlader, der ein Betriebssystem genanntes Programm mit 4 bis 10 kByte Speicherbedarf gleich nach dem Einschalten von der Diskette in den Speicher des Computers lädt. Dieses Betriebssystem unterstützt oder steuert dann die weitere Arbeit. Programme können mit Namen versehen und unter diesem zum Laden in den Speicher aufgerufen werden. Wenn Programme oder Daten den Umfang des Speichers überschreiten, kann man nicht benötigte Teile auf die Diskette «auslagern». Diese Möglichkeiten machen ein Diskettenlaufwerk für jene Amateure unentbehrlich, die sich vorwiegend mit Software beschäftigen wollen und bereits über Erfahrungen in der Computertechnik und in der Programmierung verfügen.

4.6.1. Betriebssystem für Floppy-Disk

Kurz vor Redaktionsschluß dieses Buches war es möglich, ein Floppy-Disk-Laufwerk zu erwerben. Natürlich wurde alles daran gesetzt, es so schnell wie möglich zu nutzen. Die im folgenden beschriebene Lösung arbeitet zwar schon zufriedenstellend, ist jedoch als vorläufig zu betrachten, denn besonders bei der Behandlung von Lesefehlern sind noch Verbesserungen in Hard- und Software möglich. Die Speicherung der Daten auf der Diskette erfolgt in strenger Ordnung, alle zu einem Programm gehörenden Bytes bilden eine Datei, die durch einen Namen und Sektornummern eindeutig festgelegt ist. Da ständig gelöscht und geschrieben wird, muß auf jeder Diskette ein Inhaltsverzeichnis die Kontrolle über möglicherweise

4. Weitere Mikrocomputerbauelemente und Geräte



völlig verteilt liegende Sektoren eines Programms bewahren helfen. Für die Dateiverwaltung und die Steuerung des Zugriffs für Schreiben und Lesen ist das erwähnte umfangreiche Betriebssystem erforderlich. Natürlich gibt es für diese Aufgabe recht unterschiedliche Lösungen. Eines der am weitesten verbreiteten Betriebssysteme ist CP/M. Es zeichnet sich durch folgende Vorteile aus:

- Umfangreiche Literatur.
- Viele Programme vorhanden.
- Es ist für die Anpassung an unterschiedliche Computer-Konfigurationen besonders gut geeignet, weil mehrere saubere Schnittstellen festgelegt sind.

Bild 4.6 zeigt die Aufteilung des Speichers. In seiner Minimalausführung arbeitet das Betriebssystem CP/M schon mit einem Speicher von 20 kByte, viele neuere Programme benötigen jedoch mehr Platz, so daß der Ausbau auf 64 kByte sinnvoll ist. In der sogenannten Grundseite (das ist der Speicherbereich von 0 bis 0FFH) liegen einige Speicherplätze, die sowohl vom Betriebsy-

Bild 4.6 CP/M-Speicheraufteilung

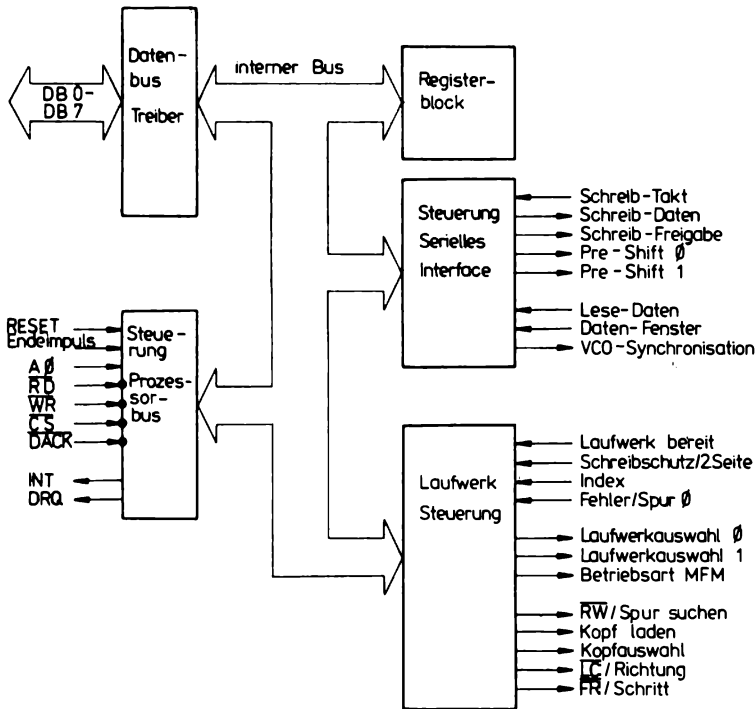


Bild 4.7
Übersichtsschaltplan
U 8272

stem als auch von Anwenderprogrammen genutzt werden. Auf Adresse 0 beginnt ein Sprungbefehl zur Warmstart-Routine, mit der die Betriebssystemteile CCP und BDOS von der Diskette in den Speicher geladen werden. Aus dem Operanden dieses Sprungs, der auf den zweiten Befehl in der Einsprungstabelle am CBIOS-Anfang zeigt, können Anwenderprogramme die Größe des Speichers berechnen. Die Startadressen von CCP und BDOS können zwar je nach verfügbarem Speicher (z. B. 20 kByte oder 48 k) auf unterschiedlichen Adressen liegen, der Umfang dieser Programmteile steht aber fest. Adresse 3 nimmt das I/O-Byte ein. In fast gleicher Weise wie beim IN/OUT-Monitor des Grundgerätes beschrieben (Zeilen 22 bzw. 844 des Programms), dient es der Zuordnung von Peripheriegeräten. Das Byte auf Adresse 4 enthält in den unteren 4 bit den Code für das aktuelle Diskettenlaufwerk, denn CP/M kann bis zu 16 Geräte ansteuern. Die oberen 4 bit sind für den Amateur ohne Bedeutung, sie erlauben die Einteilung der Disketten in verschiedene Nutzerbereiche.

Auf Adresse 5 steht ein Sprung zum Diskettenbetriebssystem BDOS. Dies ist die eigentliche, geräteunabhängige Einsprungstelle in das System. Durch Parameter, die in den Registern C, D und E übergeben werden, kann man z. B. bei der Version 2.2 von CP/M 36 Funktionen auswählen.

TPA ist der Speicherbereich, in dem die Anwenderprogramme laufen. Sie werden durch Eingabe des Programmnamens über die Tastatur von der Diskette in diesen Teil des Speichers geladen und gestartet.

Die Abkürzung CCP steht für Console Command Processor. Dieses Programm erlaubt die Abarbeitung der fest eingebauten Kommandos:

DIR	Anzeige des Inhalts von Disketten,
ERA	Löschen von Dateien auf Disketten,
REN	Umbenennen von Dateien auf Disketten,
SAVE	Abspeichern von n «Seiten» des TPA auf Disketten,
USER	Nutzerbereich festlegen,

TYPE ASCII-Dateien auf Bildschirm ausgeben.

In Ausnahmefällen können Anwenderprogramme den CCP überschreiben, wenn der Speicherplatz im TPA sonst nicht ausreicht. Sie müssen dann mit einem «Warmstart» (Sprung auf Adresse 0 der Grundseite) enden, um das Betriebssystem nach dem Abschluß wieder in den Speicher zu laden.

Mit dem Programmteil CBIOS erfolgt die Anpassung des Betriebssystems an unterschiedliche Hardware-Konfigurationen ähnlich wie im Abschnitt 2.3. für die IN/OUT-Routinen des Monitors erläutert. Zu den bereits dort vorgestellten Routinen für Tastatur, Bildschirm, Kassetteninterface und Drucker kommen im CBIOS noch Routinen für die Ansteuerung des bzw. der Floppy-Laufwerke. In industriellen Geräten fehlt fast immer ein Monitorprogramm, an denen darf aber auch meist nicht «gebastelt» werden. Für den Amateur ist der Monitor schon deshalb sinnvoll, weil mit seiner Hilfe die Schaltung und das Betriebssystem leichter in Gang zu bringen sind. Im Abschnitt 5.2. sind sowohl HEX-Listing des auf Adresse F000 verschobenen Monitors als auch ein CBIOS dargestellt. Beim Monitor fehlt hier der Disassembler, denn der Platz oberhalb des Debuggers wird für den RAM des Monitors benötigt. Das Grundgerät nach Abschnitt 1.2. wurde wie folgt verändert:

- Speicherumschaltung ähnlich Bild 3.9, jedoch unter Verwendung eines Erweiterungsgatters (Adresse F000H).
- Einbeziehung des Kassetteninterface nach Bild 4.1 in die Monitor-Routinen RI und WO.
- Erhöhung der Prozessor-Taktfrequenz auf 2,625 MHz (10,5 MHz : 4). Daraus folgt die Zeitkonstante 0D4H im System-RAM für das Druckprogramm im CBIOS.

Mit diesen Änderungen ist der Computer ohne den restlichen Speicher betriebsfähig, man kann also auch so mit dem Aufbau beginnen.

Das Programm CBIOS wird z. Z. mit dem Monitor von Kassette geladen. Es enthält vor allem Programme für die Arbeit mit Disketten. Viel Platz nimmt darin der sogenannte Physische Treiber für die Anschlußsteuerung

ein, mit der die Elektronik des Laufwerkes an den Computer-Bus gekoppelt wird. Weitere Routinen sind für die Arbeit mit BDOS nötig. Denn da dieses geräteunabhängig arbeiten soll, bei den Laufwerken aber große und kleine Disketten mit unterschiedlichen Speicherkapazitäten möglich sind, gibt es eine Reihe von genormten Schnittpunkten. Das BDOS verwendet grundsätzlich Datenblöcke von 128 Byte. Üblich sind jedoch bei doppelter Schreibdichte Blöcke mit 256 Byte. Im CBIOS muß man deshalb beim Lesen die Sektoren «spalten» und beim Schreiben «zusammensetzen». In der vorläufigen Version ist dies in den Unterprogrammen RFLOP, WFLOP und TRANS noch recht primitiv ausgeführt. Durch größere Blöcke läßt sich auf der Diskette Speicherplatz gewinnen, denn es fallen Kennzeichnungsfelder für Sektoren weg.

Um Programme von einer Diskette auf eine andere kopieren zu können, benötigt man entweder ein Programm, bei dem durch mehrmaliges Tauschen der Quell- und Zieldiskette größere Programme «stückweise» übertragen werden können, oder ein zweites Laufwerk. Letzteres kann man auf einfache Weise durch ein Programm im Speicher simulieren (Unterprogramm RAM-DISK im CBIOS).

So richtig lohnt sich dies sicher erst, wenn der dafür benötigte Speicherplatz zusätzlich zur Verfügung steht und nicht vom TPA abgeht, aber die vorgestellte Lösung ist erst einmal besser als gar keine Kopiermöglichkeit.

Dieser simulierte Floppy-Speicher erlaubt es auch, einige CP/M-Programme völlig ohne Diskettenstation zu betreiben. Die Programme und Daten muß man dann allerdings von der Kassette laden.

Den Abschluß des CBIOS bildet ein Druckprogramm für große und kleine Buchstaben mit dem Empfangsfernreiber F 1219. Auch dieses Programm entstand in kurzer Zeit und läßt sich bezüglich des benötigten Speicherplatzes noch optimieren.

4.6.2. Floppy-Disk-Schaltkreis U 8272

Der U 8272 ist ein hochintegrierter Peripherieschaltkreis, der den größten Teil der Elektronik zum Anschluß von Diskettenlaufwerken an einen Computer-Bus enthält. Er erlaubt die Ansteuerung von bis zu 4 Laufwerken, die man mit unterschiedlichen Aufzeichnungsverfahren betreiben kann. Der Schaltkreis liefert Signale sowohl für den Prozessor, für einen DMA-Schaltkreis (bei Bedarf) und für das Laufwerk.

Tabelle 4.2. zeigt die Anschlußbezeichnungen. Er kann 15 verschiedene Kommandos ausführen, die z. T. mit bis zu 9 Bytes programmiert werden. Die Adreßmarken kann der Schaltkreis völlig selbständig feststellen, die Steuerung einer PLL-Schaltung vereinfacht sich. Die vom Laufwerk abhängigen Zeiten für Schritt, Kopf laden und halten sind programmierbar. Der Schaltkreis erlaubt es sogar, mit einem Kommando Daten von beiden Seiten der Diskette zu lesen bzw. beidseitig zu schreiben. Der U 8272 enthält zwei Register, die durch den Prozessor kontrolliert werden können, ein Status- und ein Datenregister. Das 8-bit-Hauptstatusregister enthält den aktuellen Zustand des Floppy-Disk-Controllers (FDC) und kann zu beliebigen Zeiten von der CPU abgefragt werden.

Das 8-bit-Datenregister besteht aus mehreren Registern, von denen immer nur eines zu einer bestimmten Zeit mit dem Datenbus verbunden ist. Dort sind Daten, Kommandobytes, Parameter und Informationen über den aktuellen Zustand des Laufwerkes gespeichert. Tabelle 4.3. enthält die bits des Hauptstatusregisters und ihre Bedeutung.

Die Programmierung des U 8272 unterscheidet sich etwas von der bei anderen Peripherieschaltkreisen. Man unterscheidet bei den Kommandos drei Phasen:

- Kommandophase: Der Schaltkreis erhält alle Daten für ein bestimmtes Kommando.
- Ausführungsphase: Der FDC führt das Kommando aus.
- Resultatphase: Nach dem Abschluß der Operation sind Status- und andere Informationen (z. B. Spur-Nr. usw.) verfügbar.

Tabelle 4.2. Anschlüsse des U 8272 D

Symbol	Stift	Ein/Ausg.	Erläuterung
RST	1	E	Reset
RD	2	E	0 = Lesen
WR	3	E	0 = Schreiben
CS	4	E	0 = Auswahl
AO	5	E	0 = Status-, 1 = Datenregister
DB0-7	6-13	E/A	Datenbus
DRQ	14	A	DMA-Anforderung
DACK	15	E	DMA-Quittung
TC	16	E	Ende-Impuls
IDX	17	E	Index-Impuls
INT	18	A	Interrupt
CLK	19	E	Takt
GND	20		0 Volt
WRCLK	21	E	Schreibtakt
DW	22	E	Daten-Fenster
RDDATA	23	E	Lesedaten
VCO	24	A	VCO-Freigabe
WE	25	A	Schreibfreigabe
MFM	26	A	1 = MFM (doppelte Dichte), 0 = FM
HDSEL	27	A	Kopfauswahl
DS0, DS1	28,29	A	Laufwerkauswahl
WRDATA	30	A	Schreibdaten
PS1, PS0	31,32	A	Prekompensation
FLT/TO	33	E	Fehler/Spur B
WP/TS	34	E	Schreibschutz/zweiseitig
RDY	35	E	Laufwerk bereit
HDL	36	A	Kopf an Diskette anlegen (laden)
FR/STP	37	A	Fehler löschen/Schritt
LCT/DIR	38	A	Sparschaltung/Schrittrichtung
RW/SEEK	39	A	Lesen, Schreiben/Spur suchen
+5 Volt	40		

Tabelle 4.3. Statusregister des U 8272 D

Bit	Symbol	Beschreibung
0	D0B	Floppy 0 aktiv
1	D1B	1 aktiv
2	D2B	2 aktiv
3	D3B	3 aktiv
4	CB	Ein Lese- oder Schreibkommando läuft ab.
5	NDM	Der Schaltkreis arbeitet im Nicht-DMA-Betrieb. bit ist nur in der Ausführungsphase = 1. Der Übergang nach 0 zeigt deren Ende an.
6	DIO	Richtung der Datenübertragung, 1 $\hat{=}$ zur CPU
7	RQM	Fertigmeldung des U 8272 D

Während der Kommandophase – Programmierung des Schaltkreises – darf das nächste Byte erst in den Schaltkreis geschrieben werden, wenn das Hauptstatusregister die Bereitschaft des Schaltkreises dazu meldet.

Die bits 6 und 7 erlauben es, den Datenverkehr mit Quittungssignalen für «Bereitschaft» und «Richtung» abzuwickeln.

In der Ausführungsphase ist das Lesen des Hauptstatusregisters nur in der Betriebsart ohne DMA-Schaltkreis nötig und außerdem vereinfacht. Dabei kann die Bereitschaft des Schaltkreises zur Datenübertragung auch durch die Interruptleitung signalisiert werden. Die Interruptroutine oder die Abfrage muß bei doppelter Schreibdichte (Aufzeich-

4. Weitere Mikrocomputerbauelemente und Geräte

Tabelle 4.4. Befehlssatz des U 8272 D (Auszug)

Phase	Datenbit								Bemerkung
	7	6	5	4	3	2	1	0	
Kommando	mt	mfm	sk	0	0	1	x	y	lesen: x = 1, schr.: y = 1
	0	0	0	0	0	hds	ds1	ds2	
	–			C				–	Sektorinformation
	–			H				–	vor Ausführung des
	–			R				–	Kommandos
	–			EOT				–	
	–			GPL				–	
	–			DTL				–	
Ausführung									Datenübertragung zwisch-
									en Laufwerk und
									Prozessor
Resultat				ST0					Statusinformation
				ST1					nach Ausführung des
				ST2					Kommandos
				C					
				H					
				R					
				N					
Kommando				07H					Kopf auf Spur 0
	0	0	0	0	0	0	ds1	ds0	fahren
Kommando				04H					Laufwerkstatus
	0	0	0	0	0	hds	ds1	ds0	prüfen
Resultat	–			ST3				–	Status des Laufwerks
Kommando				08H					Prüfe Interruptstatus
Resultat	–			ST0				–	Status nach Spur-
				PCN				–	suche
Kommando				03H					Initialisierung
	–		srt	–I–	hut			–	
	–	hlt				–I	ND		
Kommando				0FH					Spur suchen
	0	0	0	0	0	hds	ds1	ds2	
				NCN					

nungsverfahren MFM) recht schnell erfolgen. Alle 13 µs liegt ein Byte von der Diskette an.

Den Abschluß der Ausführungsphase bildet der Ende-Impuls, den der Floppy-Controller vom Prozessor erhält.

Danach beginnt die Resultatphase. Dabei müssen alle Bytes, die in Tabelle 4.4. beim jeweiligen Kommando in dieser Phase angegeben sind, auch gelesen werden, sonst ist das Kommando nicht richtig beendet und

der Schaltkreis nimmt ein neues nicht an.

Wie aus der Tabelle ersichtlich ist, enthält der U 8272 noch 4 weitere Statusregister, die nur in der Resultatphase gelesen werden können. Nach der Initialisierung des Schaltkreises mit dem Kommando «Specify» erfolgt ohne weitere Programmierung die Abfrage der Floppy-Laufwerke auf ihren Bereitschaftsstatus. Die Auswahlleitungen werden kurzzeitig aktiviert und die Ready-Leitung geprüft. Ist eine Klappe offen oder ein Lauf-

Tabelle 4.5. Mnemonik der Befehle des U 8272 D

A0	Adresse 0
C	Spur
ds0, ds1	Laufwerksauswahl
dtl	Blocklänge der Daten
eot	letzte Sektornummer
gpl	Länge der Lücke 3 der Aufzeichnung
H, hds	Kopf (0 oder 1 bei zweiseitiger Aufzeichnung)
hlt	Kopfladezeit (programmierbar je nach Laufwerk)
hut	Kopfhaltzeit (s.o.)
mfm	Aufzeichnungsverfahren mit doppelter Dichte
mt	Aufzeichnung mit zwei Köpfen
N	Byteanzahl je Datenblock
NCN	neue Spurnummer für Suche
ND	Nicht-DMA-Betriebsart
PCN	aktuelle Spurnummer
R	Aufzeichnungsnummer, Blocknummer
sk	Sprungbefehl für gelöschte Datenfelder
srt	Schrittzeit (programmierbar je nach Laufwerk)
St0-3	Statusregister

werk nicht vorhanden, kann das der Prozessor leicht feststellen. Diese Abfrage erfolgt auch zwischen den Kommandos.

4.6.3. Anschlußsteuerung für Floppy-Disk-Laufwerk und Inbetriebnahme

Bild 4.8 zeigt den Stromlaufplan, der für die Aufzeichnung mit doppelter Dichte (Verfahren MFM) ausgelegt ist. Neben dem Quarzoszillator und den Puffern sind insbesondere für die Trennung der Daten- und Taktimpulse noch einige weitere Bauelemente nötig (Schaltkreise 15, 16, 17, 21). Beim Schreiben auf die Diskette besorgen die Schaltkreise 19 und 20 die Aufbereitung der Takt- und Datensignale.

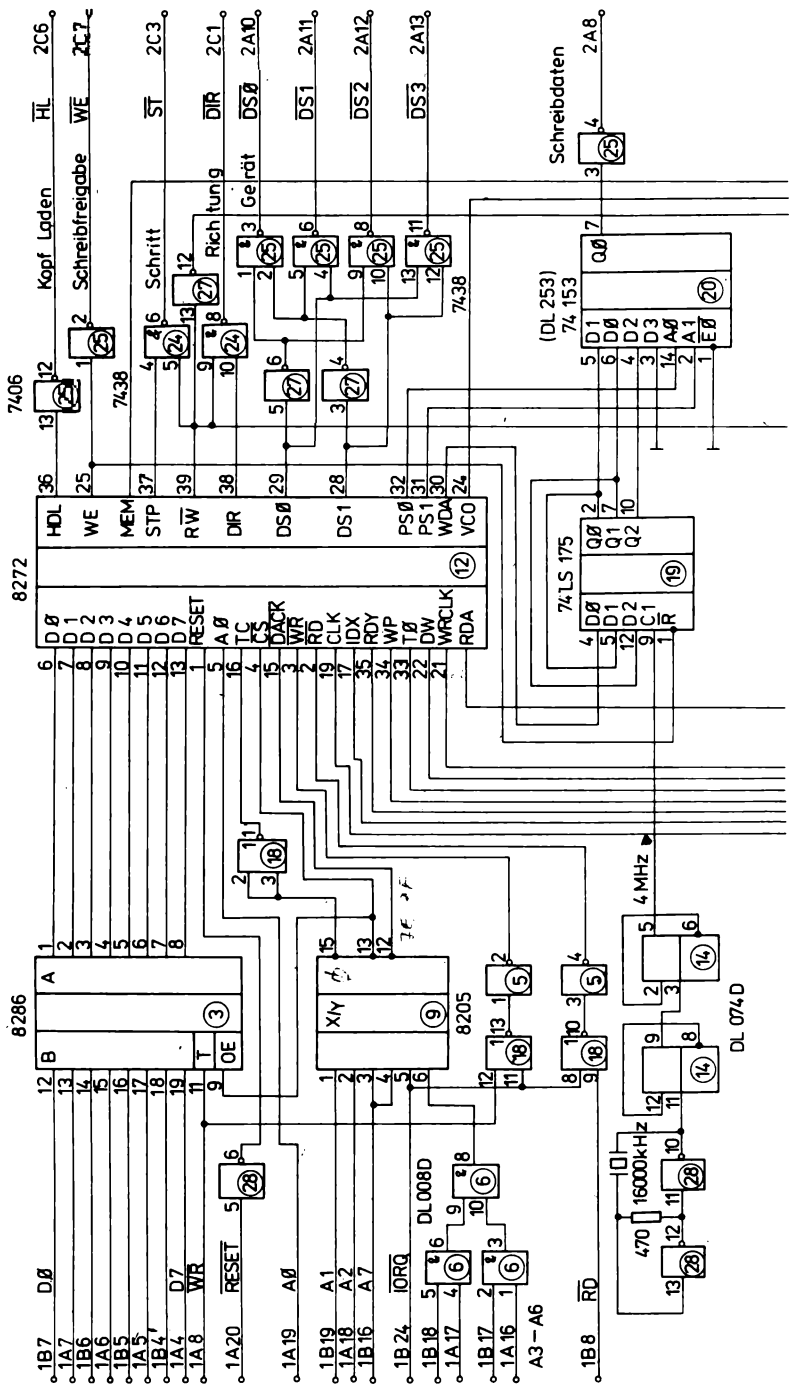
Da der U 8272 D ständig, auch wenn nicht geschrieben oder gelesen wird, den Status des Laufwerks abfragt, muß man dessen Elektronik so verändern, daß ein Ready-Signal unabhängig vom Signal «Kopf laden» entsteht. Dazu ist die Verbindung des Aus-

gangs 10 mit einer Diode auf den Anschluß 11 von IC 22 zu führen.

Die Inbetriebnahme erfolgt nach Aufbau der Hardware, indem man die Unterprogramme des CBIOS mit dem Monitor von der Kassette lädt und mit der Initialisierung des U 8272 beginnend (Kommando INIFD) einzeln mit Haltepunkten auf jedem RET testet (für das Kommando SEEK muß auf dem Speicherplatz Track ein Wert ungleich Null stehen).

Der nächste Schritt ist das Lesen eines Sektors (Unterprogramm R 8272). Hierzu ist eine initialisierte Diskette notwendig. Den Dateninhalt des gelesenen Sektors kann man sich im Puffer FDMA mit dem D-Kommando des Monitors ansehen. Bei fehlerhafter Arbeit des Programms steht im Register A der Wert 1. Die genaue Art des Fehlers kann mit Hilfe der Tabelle 5.10. und den Werten auf den Speicherplätzen RESLT ermittelt werden. Die nun erforderlichen Programme des Betriebssystems CCP und BDOS werden auf einem anderen CP/M-Rechner erstellt und auf EPROM gespeichert. Danach kann man sie im eigenen Rechner auf Kassette mit CBIOS übertragen. Nach dem Einlesen aller Teile auf die richtigen Adressen müßte CP/M funktionieren (Sprung auf GOCPM des CBIOS). Nun muß das Betriebssystem auf eine Diskette übertragen werden.

Am Mustergerät traten anfangs Probleme beim Lesen der inneren Spuren der Disketten auf. Ursache ist vorwiegend die einfache Schaltung des Datenseparators. Industrielle Schaltungen verwenden ausschließlich PLL-Schaltungen, die Schwankungen der Drehzahl des Laufwerks ausgleichen können. Die zweite mögliche Ursache sind zu große Toleranzen in der Einstellung der Laufwerke K 5100 durch den Service. Aus den Erfahrungen mit «schnellen» Kassetteninterface lag die Vermutung nahe, daß die Toleranzen des eigenen Laufwerks zu beherrschen sind, wenn die Disketten mit diesem initialisiert wurden. Wenn sie jetzt von anderen Laufwerken beschrieben werden, kann sie das Gerät trotzdem ohne Probleme lesen. Das Formatierprogramm 5.2.10. ist im Anhang zu finden.



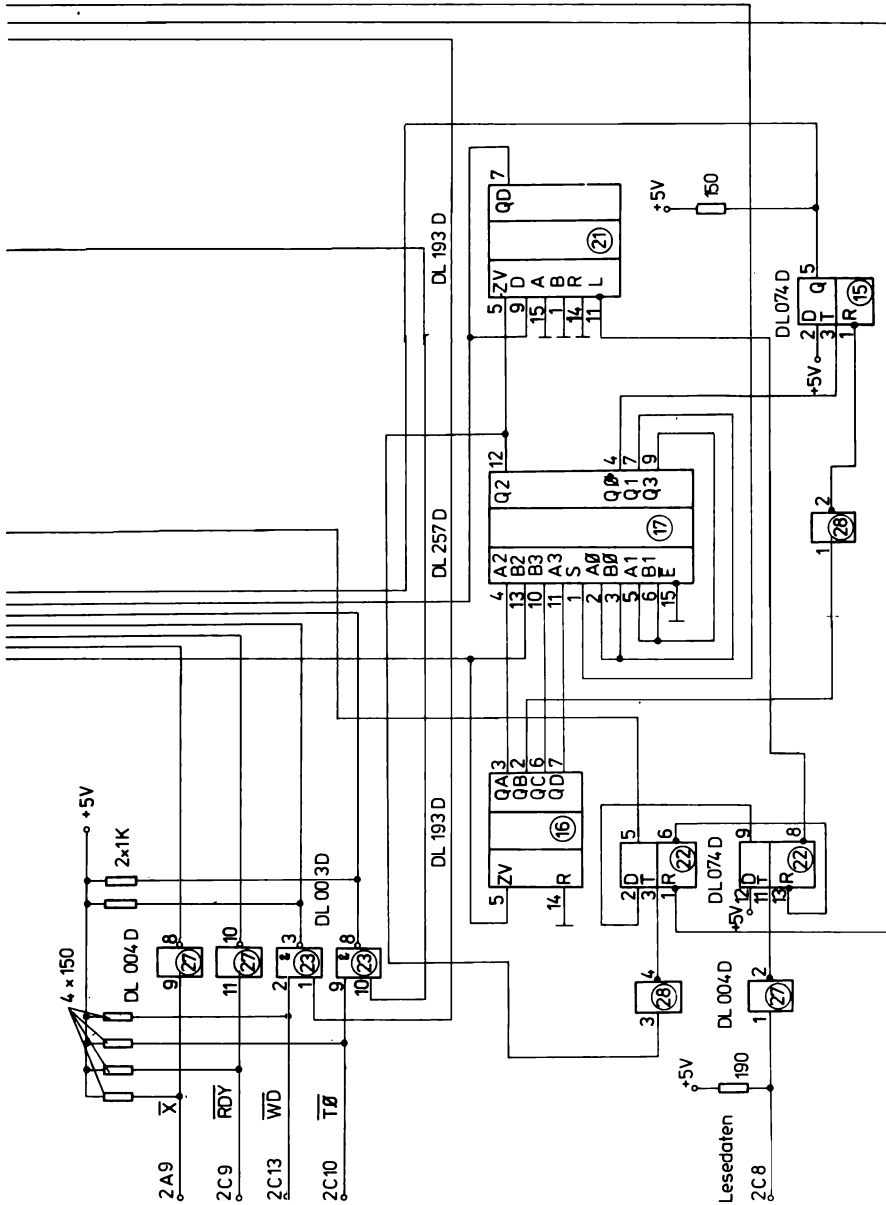


Bild 4.8 Stromlaufplan der Anschlußsteuerung

5. Anhang

5.1. Funktionserläuterung ausgewählter Maschinenbefehle des Mikroprozessors *UB 880 D*

Einige Befehle des Mikroprozessors *UB 880 D* sind einerseits sehr komplex, kommen aber andererseits in Programmen seltener vor. Deshalb ist es angebracht, neben den Programmier tabellen in Abschnitt 2.

einen «Wissensspeicher» zu nutzen, um die Funktion dieser Befehle am Beispiel vor dem Einsatz nochmals zu klären. Tabelle 5.1. enthält einige Erläuterungen zu den verwendeten Abkürzungen.

Die nachfolgenden Beispiele bilden eine Auswahl aus [20] und [21], wo der interessierte Leser alle Befehle des Prozessors erläutert findet.

Tabelle 5.1. Erläuterungen der Befehlsbeschreibung des *UB 880 D*

Register r, r'	Registerpaar tt	Bedingung cc	Bedingung c
A = 111	BC = 0	NZ (nicht Null) = 000	NZ = 00
B = 000	DE = 1	Z (Null) = 001	Z = 01
C = 001		NC (kein Übertrag)	NC = 10
D = 010	<u>Indexregister ii</u>		
E = 011	IX = 0		
H = 100	IY = 1	C (Übertrag) = 011	C = 11
L = 101		PO (Parität ungerade)	
<u>Registerpaar dd</u>	<u>Registerpaar qq</u>	PE (Parität gerade)	
BC = 00	BC = 00		
DE = 01	DE = 01		
HL = 10	HL = 10	P (Positives Vorzeichen) = 110	
SP = 11	AF = 11	M (Negatives Vorzeichen) = 111	
<u>Registerpaar pp</u>	<u>bit b</u>		
BC = 00	0 000		
DE = 01	1 001		
IX, IY = 10	2 010		
SP = 11	3 011		
	4 100		
	5 101		
	6 110		
	7 111		

Das für die indizierte Adressierung benutzte Symbol d kann die Werte von -128 bis +127 (Darstellung im Zweierkomplement) annehmen.

Das für die relativen Sprünge benutzte Symbol e kann die Werte von -126 bis +129 annehmen.

EXAFOperation $AF \leftrightarrow AF'$

Maschinencode:

00001000 08

Beschreibung: Die 16-bit-Inhalte der Registerpaare AF und AF' werden ausgetauscht.

M-Zyklen: 1

T-Zyklen: 4

Flag-Register: F und F' werden ausgetauscht

Beispiel: Das Registerpaar AF enthalte 7981H und das Registerpaar AF' enthalte C040H. Nach dem Befehl EXAF ist das Registerpaar AF mit C040H und AF' mit 7981H geladen.

EX (SP), HLOperation $H \leftrightarrow (SP + 1)$, $L \leftrightarrow (SP)$

Maschinencode:

11100011 E3

Beschreibung: Der Inhalt des Registers L wird mit dem Inhalt der Speicherstelle ausgetauscht, die durch SP adressiert wird. Der Inhalt des Registers H wird mit dem Inhalt der Speicherstelle ausgetauscht, die durch SP + 1 adressiert wird. Der Stack-Pointer SP wird nicht verändert.

M-Zyklen: 5

T-Zyklen: 19

Flag-Register: wird nicht beeinflusst.

Beispiel: Der Inhalt von HL sei 2751H, der von SP sei 8586H. Die Speicherstelle 8586H enthalte 34H, die Speicherstelle 8587H enthalte 47H. Nach dem Befehl EX (SP), HL enthält das Registerpaar HL 4734H, und der Speicherplatz 8586H ist mit 51H und der Speicherplatz 8587H ist mit 27H geladen.

EX DE, HLOperation $DE \leftrightarrow HL$

Maschinencode:

11101011 EB

Beschreibung: Die 16-bit-Inhalte der Registerpaare DE und HL werden ausgetauscht.

M-Zyklen: 1

T-Zyklen: 4

Flag-Register: wird nicht beeinflusst

Beispiel: Das Registerpaar DE enthalte 2EC7H und das Registerpaar HL enthalte 7827H. Nach dem Befehl EX DE, HL ist das Registerpaar DE mit 7827H und HL mit 2EC7H geladen.

EXXOperation $BC \leftrightarrow BC'$, $DE \leftrightarrow DE'$, $HL \leftrightarrow HL'$

Maschinencode:

11011001 D9

Beschreibung: Die 16-bit-Inhalte der Registerpaare BC und BC', DE und DE', HL und HL' werden ausgetauscht.

M-Zyklen: 1

T-Zyklen: 4

Flag-Register: wird nicht beeinflusst

Beispiel: Der Inhalt der Registerpaare sei

 $BC = 8895H$, $DE = A2D3H$, $HL = 0978H$, $BC' = 5A44H$, $DE' = 0093H$, $HL' = B004H$. Nach dem Befehl EXX sind die Registerpaare wie folgt geladen: $BC = 5A44H$, $DE = 0093H$, $HL = B004H$, $BC' = 8895H$, $DE' = A2D3H$, $HL' = 0978H$.**EX (SP), ii**Operation $IX_H \leftarrow (SP + 1)$, $IX_L \leftarrow (SP)$

Maschinencode:

11ii11101 IX oder IY

DD FD

11100011 E3

Beschreibung: Der Inhalt des niederwertigen Teils des Indexregisters ii wird mit dem Inhalt der Speicherstelle, die durch SP adressiert wird, ausgetauscht. Der Inhalt des höherwertigen Teils des Indexregisters ii wird mit dem Inhalt der Speicherstelle, die durch SP + 1 adressiert wird, ausgetauscht.

M-Zyklen: 6

T-Zyklen: 23

Flag-Register: wird nicht beeinflusst

Beispiel: Der Inhalt von IX sei 1398H, der von SP sei FF80H. Die Speicherstelle FF80H enthalte 02H, die Speicherstelle FF81H enthalte 66H. Nach dem Befehl EX (SP), IX enthält das Indexregister IX 6602H, und der Speicherplatz FF80H ist mit 98H und der Speicherplatz FF81H mit 13H geladen. Der Stack-Pointer enthält dann FF80H.

LDI

Operation (DE) $\leftarrow$ (HL), DE $\leftarrow$ DE + 1,
HL $\leftarrow$ HL + 1, BC $\leftarrow$ BC - 1

Maschinencode:

11101101 ED

10100000 A0

Beschreibung: Der Inhalt der durch HL adressierten Speicherstelle wird in die durch DE adressierte Speicherstelle geladen. Danach werden die Registerpaare DE und HL um 1 erhöht und das Registerpaar BC um 1 vermindert.

M-Zyklen: 4

T-Zyklen: 16

Flag-Register:

S, Z - nicht definiert

H, N - 0 gesetzt

P/V - 1 falls BC - 1 $\neq$ 0, sonst 0

C - wird nicht beeinflusst

Beispiel: Es sei HL = 2178H, DE = 3455H, BC = 0100H. Der Inhalt der Speicherzelle 2178H sei A7H. Durch die Ausführung des Befehls LDI ergeben sich folgende Inhalte: HL = 2179H, DE = 3456H, BC = 00FFH, (3455H) = A7H

LDIR

Operation (DE) $\leftarrow$ (HL), DE $\leftarrow$ DE + 1,
HL $\leftarrow$ HL + 1, BC $\leftarrow$ BC - 1

Maschinencode:

11101101 ED

10110000 B0

Beschreibung: Der Inhalt der durch HL adressierten Speicherstelle wird in die Speicherstelle, die DE adressiert, geladen. Danach werden die Registerpaare HL und DE um 1 erhöht und das Registerpaar BC um 1 vermindert. Dieser Vorgang wird so lange wiederholt, bis BC = 0 erreicht ist.

M-Zyklen: 5; T-Zyklen: 21 für BC $\neq$ 0

M-Zyklen: 4; T-Zyklen: 16 für BC = 0

Flag-Register:

S, Z - nicht definiert

C - wird nicht beeinflusst

H, P/V, N - werden 0 gesetzt

Beispiel: Die Registerpaare und Speicherplätze sind wie folgt geladen:

HL = 1000H, DE = 2000H, BC = 0003H.

(2000H) = XX (1000H) = 45H

(2001H) = XX (1001H) = 37H

(2002H) = XX (1002H) = 21H

Nach der Ausführung des Befehls LDIR ergeben sich folgende Inhalte:

HL = 1003H, DE = 2003H, BC = 0000H

(2000H) = 45H (1000H) = 45H

(2001H) = 37H (1001H) = 37H

(2002H) = 21H (1002H) = 21H

LDDR

Operation (DE) $\leftarrow$ (HL), DE $\leftarrow$ DE - 1,
HL $\leftarrow$ HL - 1, BC $\leftarrow$ BC - 1

Maschinencode:

11101101 ED

10111000 B8

Beschreibung: Der Inhalt der durch HL adressierten Speicherstelle wird in die Speicherstelle, die durch DE adressiert wird, geladen. Danach werden die Registerpaare HL, DE und BC um 1 vermindert. Dieser Vorgang wird so lange wiederholt, bis BC = 0 erreicht ist.

M-Zyklen: 5; T-Zyklen: 21 für BC $\neq$ 0

M-Zyklen: 4; T-Zyklen: 16 für BC = 0

Flag-Register:

S, Z - nicht definiert

H, P/V, N - werden 0 gesetzt

C - wird nicht beeinflusst

Beispiel: Folgende Registerpaare und Speicherplätze sind geladen:

HL = 3004H, DE = 4004H, BC = 0003H

(3004H) = 2AH, (4004H) = XX

(3003H) = C7H, (4003H) = XX

(3002H) = FOH, (4002H) = XX

Nach der Ausführung des Befehls LDDR ergeben sich folgende Inhalte:

HL = 3001H, DE = 4001H, BC = 0000H

(3004H) = 2AH, (4004H) = 2AH

(3003H) = C7H, (4003H) = C7H

(3002H) = FOH, (4002H) = FOH

LDD

Operation (DE) $\leftarrow$ (HL), DE $\leftarrow$ DE - 1,
HL $\leftarrow$ HL - 1, BC $\leftarrow$ BC - 1

Maschinencode:

11101101 ED

10101000 A8

Beschreibung: Der Inhalt der durch HL adressierten Speicherstelle wird in die durch DE adressierte Speicherstelle geladen. Danach werden die Registerpaare DE, HL und BC um 1 vermindert.

M-Zyklen: 4

T-Zyklen: 16

Flag-Register:

S, Z – nicht definiert

H, N – werden 0 gesetzt

P/V – 1 falls $BC - 1 \neq 0$ sonst 0

C – wird nicht beeinflusst

Beispiel: HL = 2106H, DE = 5794H,

BC = 0050H (2106H) = 27H, (5794H) = XX

Nach dem Befehl LDD ergeben sich folgende Inhalte:

HL = 2105H, DE = 5793H, BC = 004FH

(2106H) = 27H, (5794H) = 27H

CPI

Operation A – (HL), $HL \leftarrow HL + 1$,
BC ← BC – 1

Maschinencode:

11101101 ED

10100001 A1

Beschreibung: Der Inhalt des Akkumulators wird mit dem Inhalt der durch HL adressierten Speicherstelle verglichen, und entsprechend dem Ergebnis wird das Flag-Register geladen. Danach wird das Registerpaar HL um 1 erhöht und das Registerpaar BC um 1 vermindert.

M-Zyklen: 4

T-Zyklen: 16

Flag-Register:

S – 1, falls das Ergebnis negativ, sonst 0

Z – 1, falls A = (HL), sonst 0

H – 1, falls von bit 4 «geborgt» wurde,
sonst 0

P/V – 1, falls $BC - 1 \neq 0$, sonst 0

N – 1 gesetzt

C – wird nicht beeinflusst

Beispiel: Folgende Register und Speicherplätze sind geladen:

HL = 3700H, A = 07H, (3700H) = A0H,

F = 0, BC = 0005H

Nach der Ausführung des Befehls CPI ergeben sich folgende Inhalte:

HL = 3701H, A = 07H, F = 26H

CPD

Operation: A – (HL), $HL \leftarrow HL - 1$,
BC ← BC – 1

Maschinencode:

11101101 ED

10101001 A9

Beschreibung: Der Inhalt des Akkumulators wird mit dem Inhalt der durch HL adressierten Speicherstelle verglichen, und entsprechend dem Ergebnis wird das Flag-Register geladen. Danach werden die Registerpaare HL und BC um 1 vermindert.

M-Zyklen: 4

T-Zyklen: 16

Flag-Register:

S – 1, falls das Ergebnis negativ, sonst 0

Z – 1, falls A = (HL), sonst 0

H – 1, falls von bit 4 «geborgt» wurde,
sonst 0

P/V – 1, falls BC = 0, sonst 0

N – 1 gesetzt

C – wird nicht beeinflusst.

Beispiel: Folgende Register und Speicherplätze sind geladen:

HL = 1000H, A = 10H, BC = 0001H,

F = 00H, (1000H) = 10H

Nach der Ausführung des Befehls CPD ergeben sich folgende Inhalte:

HL = 0FFFH, A = 10H, BC = 0000H,

F = 42H

CPIR

Operation A – (HL), $HL \leftarrow HL + 1$,
BC ← BC – 1

Maschinencode:

11101101 ED

10110001 B1

Beschreibung: Der Inhalt des Akkumulators wird mit dem Inhalt der durch HL adressierten Speicherstelle verglichen. Entsprechend dem Ergebnis wird das Flag-Register geladen. Danach wird das Registerpaar HL um 1 erhöht und das Registerpaar BC um 1 vermindert. Dieser Vorgang wird so lange wiederholt, bis BC = 0 oder der Inhalt des Akkumulators gleich dem Inhalt der durch HL adressierten Speicherstelle ist.

M-Zyklen (für $BC \neq 0$ und $A \neq (HL)$): 5

T-Zyklen: 21

M-Zyklen (für $BC = 0$ oder $A = (HL)$): 4

T-Zyklen: 16

5. Anhang

Flag-Register:

S - 1, falls das Ergebnis negativ, sonst 0

Z - 1, falls A = (HL), sonst 0

H - 1, falls von bit 4 «geborgt» wurde,
sonst 0

P/V = 1, falls BC - 1 ≠ 0, sonst 0

N - 1 gesetzt

C - wird nicht beeinflusst

Beispiel: Folgende Register und Speicher-
plätze sind geladen:

HL = 3107H, A = 7CH, BC = 0004H,

F = 00H

(3107H) = 02H, (3108H) = 3FH,

(3109H) = 7CH, (310AH) = B0H.

Nach der Ausführung des Befehls CPIR er-
geben sich folgende Inhalte:

HL = 310AH, A = 7CH, BC = 0001H,

F = 46H

CPDR

Operation: A - (HL), HL ← HL - 1,
BC ← BC - 1

Maschinencode:

11101101 ED

10111001 B9

Beschreibung: Der Inhalt des Akkumulators
wird mit dem Inhalt der durch HL adressier-
ten Speicherstelle verglichen, und entspre-
chend dem Ergebnis wird das Flag-Register
geladen. Danach werden die Registerpaare
HL und BC um 1 vermindert. Dieser Vor-
gang wird so lange wiederholt, bis BC = 0
oder der Inhalt des Akkumulators gleich
dem Inhalt der durch HL adressierten Spei-
cherstelle ist.

M-Zyklen (für BC ≠ 0 und A ≠ (HL)): 5

T-Zyklen: 21

M-Zyklen (für BC = 0 oder A = (HL)): 4

T-Zyklen: 16

Flag-Register:

S - 1, falls das Ergebnis negativ, sonst 0

Z - 1, falls A = (HL), sonst 0

H - 1, falls von bit 4 «geborgt» wurde,
sonst 0

P/V - 1, falls BC ≠ 0, sonst 0

N - 1 gesetzt

C - wird nicht beeinflusst

Beispiel: Folgende Register und Speicher-
plätze sind geladen:

HL = 1005H, A = 22H, F = 00H, BC = 003H
(1005H) = 22H, (1004H) = 6FH,
(1003H) = 60H.

Nach der Ausführung des Befehls CPDR er-
geben sich folgende Inhalte:

HL = 1004H, A = 22H, F = 46H,
BC = 0002H

XOR r

Operation: $A \leftarrow A \oplus r$

Maschinencode:

10101 ← r → A8 - AF

Beschreibung: Der Inhalt des Registers r
wird mit dem Inhalt des Akkumulators A ex-
klusiv geodert.

M-Zyklen: 1

T-Zyklen: 4

Flag-Register:

S - 1, falls das Ergebnis negativ, sonst 0

Z - 1, falls das Ergebnis 0, sonst 0

P/V - 1, falls Parität gerade, sonst 0

H, N, C - 0 gesetzt

Beispiel: Der Inhalt des Registers A sei 4CH,
D sei 75H. Nach der Ausführung des Befehls
XOR D enthält der Akkumulator A = 39H.

XOR n

Operation: $A \leftarrow A \oplus n$

Maschinencode:

11101110 EE

← n → xx

Beschreibung: Der Inhalt des Akkumulators
A wird mit der Konstanten n exklusiv ge-
odert.

M-Zyklen: 2

T-Zyklen: 7

Flag-Register: wie XOR r

Beispiel: Der Inhalt des Akkumulators A sei
73H, n sei 8CH. Nach der Ausführung des
Befehls XOR n enthält der Akkumulator
A = FFH.

XOR (ii + d)

Operation: $A \leftarrow A \oplus (ii + d)$

Maschinencode:

11 ii 11101

10101110

← d →

IX oder IY

DD FD

AE

xx

Beschreibung: Der Inhalt des Akkumulators A wird mit dem Inhalt der durch $ii + d$ adressierten Speicherstelle exklusiv geodert.

M-Zyklen: 5

T-Zyklen: 19

Flag-Register: wie XORr

Beispiel: Der Inhalt von A sei 90H, IX sei 3750H, $d = 27H$ und $(3777H) = 82H$. Nach der Ausführung des Befehls enthält der Akkumulator A = 12H.

XORM

Operation: $A \leftarrow A \oplus (HL)$

Maschinencode:

10101110 AE

Beschreibung: Der Inhalt des Akkumulators A wird mit dem Inhalt der durch HL adressierten Speicherstelle exklusiv geodert.

M-Zyklen: 2

T-Zyklen: 7

Flag-Register: wie XORr

Beispiel: Der Inhalt von A sei 37H, HL sei 87C0H, $(87C0H) = 37H$. Nach Ausführung des Befehls XORM enthält der Akkumulator A = 00H.

CMP r

Operation: $A - r$

Maschinencode:

10111 $\leftarrow r \rightarrow$ B8 - BF

Beschreibung: Der Inhalt des Registers r wird mit dem Inhalt des Akkumulators verglichen. Der Inhalt von A bleibt erhalten. Entsprechend dem Ergebnis wird das Flag-Register gesetzt.

M-Zyklen: 1

T-Zyklen: 4

Flag-Register:

S - 1, falls das Ergebnis negativ, sonst 0

Z - 1, falls Ergebnis 0, sonst 0

H - 0, falls von bit 4 «geborgt» wurde, sonst 1

P/V - 1, falls Überlauf, sonst 0

N - 1 gesetzt

C - 0, falls geborgt wurde, sonst 1

Beispiel: Der Inhalt von A sei 47H, H sei 22H, F = 00H. Nach der Ausführung des Befehls CMP H enthält F = 22H.

CMP n

Operation: $A - n$

Maschinencode:

11111110 FE

$\leftarrow n \rightarrow$ xx

Beschreibung: Der Inhalt des Akkumulators A wird mit der Konstanten n verglichen. Der Inhalt von A bleibt erhalten. Entsprechend dem Ergebnis wird das Flag-Register gesetzt.

M-Zyklen: 2

T-Zyklen: 7

Flag-Register: wie CMP r

Beispiel: Der Inhalt von A sei 75H, n sei 75H. Nach der Ausführung des Befehls CMP n enthält das Flag-Register F = 62H.

CMPM

Operation: $A - (HL)$

Maschinencode:

10111110 BE

Beschreibung: Der Inhalt des Akkumulators A wird mit der durch HL adressierten Speicherstelle verglichen. Der Inhalt von A bleibt erhalten. Entsprechend dem Ergebnis wird das Flag-Register gesetzt.

M-Zyklen: 2

T-Zyklen: 7

Flag-Register: wie CMP r

Beispiel: Es sei A = 20H, HL = E000H, $(E000H) = 30H$. Nach Ausführung des Befehls CMPM enthält F = A3H.

CMP (ii + d)

Operation: $A - (ii + d)$ IX oder IY

Maschinencode:

11ii11101 DD FD

10111110 BE

$\leftarrow d \rightarrow$ xx

Beschreibung: Der Inhalt des Akkumulators A wird mit der durch $ii + d$ adressierten Speicherstelle verglichen. Der Inhalt von A bleibt erhalten. Entsprechend dem Ergebnis wird das Flag-Register gesetzt.

M-Zyklen: 5

T-Zyklen: 19

Flag-Register: wie CMP r

Beispiel: Der Inhalt des Akkumulators sei 57H, IX sei 7420H, $d = 13H$ und $(7433H) = 31H$. Nach Ausführung des Befehls CMP (IX + d) enthält F = 22H.

INC r

Operation: $r \leftarrow r + 1$

Maschinencode:

00 $\leftarrow r \rightarrow$ 100

Beschreibung: Der Inhalt des Registers r wird um 1 erhöht.

M-Zyklen: 1

T-Zyklen: 4

Flag-Register:

S - 1, falls das Ergebnis negativ, sonst 0

Z - 1, falls das Ergebnis 0, sonst 0

H - 1, falls Übertrag von bit 3, sonst 0

P/V - 1, falls r vor der Operation 7FH war, sonst 0

N - 0 gesetzt

C - wird nicht beeinflusst.

Beispiel: Der Inhalt von D sei 3EH. Nach der Ausführung des Befehls INCD enthält D = 3FH.

INCM

Operation: $(HL) \leftarrow (HL) + 1$

Maschinencode:

00110100 34

Beschreibung: Der Inhalt der durch HL adressierten Speicherstelle wird um 1 erhöht.

M-Zyklen: 3

T-Zyklen: 11

Flag-Register: wie INC r

Beispiel: Der Inhalt von HL sei A027H und (A027H) = 35H. Nach der Ausführung des Befehls ergibt sich (A027H) = 36H.

INC (ii + d)

Operation: $(ii + d) \leftarrow (ii + d) + 1$

Maschinencode:

11ii11101 IX oder IY

00110100 DD FD

$\leftarrow d \rightarrow$ 34

xx

Beschreibung: Der Inhalt der durch ii + d adressierten Speicherstelle wird um 1 erhöht.

M-Zyklen: 6

T-Zyklen: 23

Flag-Register: wie INC r

Beispiel: Es sei IX = 3434H, d = 21H und (3455H) = 1FH. Nach der Ausführung des Befehls INC (IX + d) ergibt sich (3455H) = 20H.

DEC r

Operation: $r \leftarrow r - 1$

Maschinencode:

00 $\leftarrow r \rightarrow$ 101

Beschreibung: Der Inhalt des Registers r wird um 1 vermindert.

M-Zyklen: 1

T-Zyklen: 4

Flag-Register:

S - 1, falls das Ergebnis negativ, sonst 0

Z - 1, falls das Ergebnis 0, sonst 0

H - 0, falls von bit 4 «geborgt» wurde, sonst 1

P/V - 1, falls r vor der Operation 80H war, sonst 0

N - 1 gesetzt

C - wird nicht beeinflusst

Beispiel: Das Register B enthält 2CH. Nach der Ausführung des Befehls DECB enthält B = 2BH.

DECM

Operation: $(HL) \leftarrow (HL) - 1$

Maschinencode:

00110101 35

Beschreibung: Der Inhalt der durch HL adressierten Speicherstelle wird um 1 vermindert.

M-Zyklen: 3

T-Zyklen: 11

Flag-Register: wie DEC r

Beispiel: HL sei 2100H und (2100H) = 20H. Nach der Ausführung des Befehls DECM ergibt sich (2100H) = 1FH.

DEC (ii + d)

Operation: $(ii + d) \leftarrow (ii + d) - 1$

Maschinencode:

11ii11101 IX oder IY

00110101 DD FD

$\leftarrow d \rightarrow$ 35

Beschreibung: Der Inhalt der durch ii + d adressierten Speicherstelle wird um 1 vermindert.

M-Zyklen: 6

T-Zyklen: 23

Flag-Register: wie DEC r

Beispiel: IX sei 7BC0H, d = 10H und (7BD0H) = 33H. Nach der Ausführung des Befehls DEC (IX + d) ergibt sich (7BD0H) = 32H.

ADDHL, ddOperation: $HL \leftarrow HL + dd$

Maschinencode:

00dd1001 09 bis 39

Beschreibung: Der Inhalt des 16-bit-Registers dd wird zum Inhalt des Registerpaares HL addiert.

M-Zyklen: 3

T-Zyklen: 11

Flag-Register:

S, Z, P/V – werden nicht beeinflusst

H – nicht definiert

N – 0 gesetzt

c – 1, falls Übertrag von bit 15, sonst 0

Beispiel: HL sei 21C0H und DE 4781H.

Nach der Ausführung des Befehls ADDHL, DE ergibt sich HL zu 6941H.

ADDii, ppOperation: $ii \leftarrow ii + pp$

Maschinencode:

IX oder IY

11ii11101

DD FD

00pp1001

09 – 39

Beschreibung: Der Inhalt des 16-bit-Registers wird zum Inhalt des Indexregisters ii addiert (beachte: ADD IX, IY bzw. ADD IY, IX nicht möglich).

M-Zyklen: 4

T-Zyklen: 15

Flag-Register: wie ADDHL, dd

Beispiel: Es sei IX = 00010H. Nach der Ausführung des Befehls ADDIX, IX ergibt sich IX zu 0020H. (Entspricht Linksverschiebung der 16 bit um 1 Bitposition.)

ADCHL, ddOperation: $HL \leftarrow HL + dd + CY$

Maschinencode:

11101101 ED

01dd1010 4A – 7A

Beschreibung: Der Inhalt des 16-bit-Registers dd plus das Carrybit CY wird zum Inhalt des Registerpaares HL addiert.

M-Zyklen: 4

T-Zyklen: 15

Flag-Register:

S – 1, falls das Ergebnis negativ, sonst 0

Z – 1, falls das Ergebnis 0, sonst 0

H – nicht definiert

P/V – 1, falls Überlauf, sonst 0

N – 0 gesetzt

C – 1, falls Übertrag von bit 15, sonst 0

Beispiel: HL sei A000H, SP = 1110H und F = 41H. Nach der Ausführung des Befehls ADCHL, SP ergibt sich HL zu B111H.

SBCHL, ddOperation: $HL \leftarrow HL - dd - CY$

Maschinencode:

11101101 ED

01dd0010 42 – 72

Beschreibung: Der Inhalt des 16-bit-Registers dd plus Carrybit CY wird vom Inhalt des Registerpaares HL subtrahiert.

M-Zyklen: 4

T-Zyklen: 15

Flag-Register:

S – 1, falls Ergebnis negativ, sonst 0

Z – 1, falls Ergebnis 0, sonst 0

H – nicht definiert

P/V – 1, falls Überlauf, sonst 0

N – 1 gesetzt

C – 1, falls geborgt wurde, sonst 0

Beispiel: HL sei BD20H, DE sei 0200H und F = 80H. Nach Ausführung des Befehls ergibt sich HL zu BB20H.

INCddOperation: $dd \leftarrow dd + 1$

Maschinencode:

00dd0011 03 – 33

Beschreibung: Der Inhalt des 16-bit-Registers dd wird um 1 erhöht.

M-Zyklen: 1

T-Zyklen: 6

Flag-Register: wird nicht beeinflusst

Beispiel: Der Inhalt von BC sei 0095H. Nach der Ausführung des Befehls INC BC enthält BC 0096H.

INCiiOperation: $ii \leftarrow ii + 1$

Maschinencode:

IX oder IY

11ii11101

DD FD

00100011

23

Beschreibung: Der Inhalt des Indexregisters ii wird um 1 erhöht.

M-Zyklen: 2 T-Zyklen: 10

Flag-Register: wird nicht beeinflusst

Beispiel: Das Indexregister IY enthält 27CFH. Nach der Ausführung des Befehls INC IY ergibt sich IY zu 27D0H.

5. Anhang

DEC ii

Operation: $ii \leftarrow ii - 1$

Maschinencode: IX oder IY

11ii11101 DD FD

00101011 2B

Beschreibung: Der Inhalt des Indexregisters ii wird um 1 vermindert.

M-Zyklen: 2

T-Zyklen: 10

Flag-Register: wird nicht beeinflusst

Beispiel: Der Inhalt von IY sei 30C0H. Nach der Ausführung des Befehls DEC IY enthält IY 30BFH.

DEC dd

Operation: $dd \leftarrow dd - 1$

Maschinencode:

00dd1011 0B - 3B

Beschreibung: Der Inhalt des 16-bit-Registers dd wird um 1 vermindert.

M-Zyklen: 1

T-Zyklen: 6

Flag-Register: wird nicht beeinflusst

Beispiel: Der Inhalt von SP sei F845H. Nach der Ausführung des Befehls DEC SP ergibt sich F844H.

DAA

Maschinencode:

00100111 27

Beschreibung: Der DAA-Befehl korrigiert nach einer Addition oder Subtraktion zweier gepackter BCD-Zahlen den Inhalt des Akkumulators so, daß wieder gepacktes BCD-Format entsteht.

M-Zyklen: 1

T-Zyklen: 4

Flag-Register:

S - 1, falls bit 7 1, sonst 0

Z - 1, falls Ergebnis 0, sonst 0

H, C - siehe Tabelle 5.2.

P/V - 1, falls Parität des Akkumulators gerade, sonst 0

N - wird nicht beeinflusst

CPL

Operation: $A \leftarrow \bar{A}$

Maschinencode:

00101111 2F

Beschreibung: Der Inhalt des Akkumulators ergibt sich aus der Negation (bitweises Negieren) des Akkumulatorinhaltes

M-Zyklen: 1

T-Zyklen: 4

Flag-Register:

S, Z, P/V, C - werden nicht beeinflusst

H, N - 1 gesetzt

Beispiel: Der Inhalt des Akkumulators sei 17H. Nach der Ausführung des Befehls CPL ergibt sich A zu E8H.

Tabelle 5.2. Durch den DAA-Befehl ausgelöste Operationen

Befehl	CY vor DAA	Wert der bits 7...4	H vor DAA	Wert der bits (3...0)	Durch DAA addierter Wert	CY nach DAA
ADD ADC INC	0	0-9	0	0-9	00	0
	0	0-8	0	A-F	06	0
	0	0-9	1	0-3	06	0
	0	A-F	0	0-9	60	1
	0	9-F	0	A-F	66	1
	0	A-F	1	0-3	66	1
	1	0-2	0	0-9	60	1
	1	0-2	0	A-F	66	1
	1	0-3	1	0-3	66	1
SUB	0	0-9	0	0-9	00	0
SBC	0	0-8	1	6-F	FA	0
DEC	1	7-F	0	0-9	A0	1
NEG	1	6-F	1	6-F	9A	1

NEGOperation: $A \leftarrow 0 - A$

Maschinencode:

11101101 ED

01000100 44

Beschreibung: Der Inhalt des Akkumulators A ergibt sich aus der Subtraktion des Akkumulatorinhaltes von 0 (Zweierkomplement).

M-Zyklen: 2

T-Zyklen: 8

Flag-Register:

S – 1, falls das Ergebnis negativ, sonst 0

Z – 1, falls das Ergebnis 0, sonst 0

H – 1, falls von bit 4 «geborgt» wurde, sonst 0

P/V – 1, falls A vor dem Befehl 80H war, sonst 0

N – 1 gesetzt

C – 0, falls A vor dem Befehl 00H war, sonst 1

Beispiel: A sei 36H. Nach der Ausführung des Befehls NEG ergibt sich A zu CAH.

NOP

Operation: –

Maschinencode:

00000000 00

Beschreibung: Die CPU führt keine Operation aus.

M-Zyklen: 1

T-Zyklen: 4

Flag-Register: werden nicht beeinflusst.

CCFOperation: $CY \leftarrow \overline{CY}$

Maschinencode:

00111111 3F

Beschreibung: Das Carrybit CY wird komplementiert.

M-Zyklen: 1

T-Zyklen: 4

Flag-Register:

S, Z, P/V – werden nicht beeinflusst

H – nicht definiert

N – 0 gesetzt

C – 1, falls CY vor dem Befehl = 0, sonst 0

HALT

Operation: –

Maschinencode:

01110110 76

Beschreibung: Die CPU führt so lange NOP-Befehle aus, bis ein Interrupt- oder der RESET-Eingang aktiv wird.

Refreshzyklen werden erzeugt.

M-Zyklen: 1

T-Zyklen: 4

Flag-Register: werden nicht beeinflusst

SCFOperation: $CY \leftarrow 1$

Maschinencode:

00110111 37

Beschreibung: Das Carrybit CY wird 1 gesetzt

M-Zyklen: 1

T-Zyklen: 4

Flag-Register:

S, Z, P/V – werden nicht beeinflusst

H, N – 0 gesetzt

C – 1 gesetzt

IN nOperation: $A \leftarrow (n)$

Maschinencode:

11011011 DB

 $\rightarrow n \rightarrow$ xx

Beschreibung: Der Inhalt des Akkumulators A wird mit dem Inhalt des durch n adressierten Eingabekanals geladen.

M-Zyklen: 3

T-Zyklen: 11

Flag-Register: wird nicht beeinflusst

Beispiel: $n = 3CH$ und $(n) = 27H$. Nach der Ausführung des Befehls IN n ergibt sich A zu 27H.

IN rOperation: $r \leftarrow (C)$

Maschinencode:

11 101 101 ED

 $01 \leftarrow r \rightarrow 000$ 40 – 78

Beschreibung: Das Register r wird mit dem Inhalt des durch C adressierten Eingabekanals geladen.

M-Zyklen: 3

T-Zyklen: 12

5. Anhang

Flag-Register:

S – 1, falls Eingabedaten negativ, sonst 0

Z – 1, falls Eingabedaten 0, sonst 0

H, N – 0 gesetzt

P/V – 1, falls Eingabedaten gerade Parität,
sonst 0

C – wird nicht beeinflusst

Beispiel: Das Register C enthalte 70H und die Kanaladresse 70H enthalte als Eingabedaten 25H. Nach der Befehlsausführung INL ergibt sich L zu 25H.

OUTn

Operation: (n) ← A

Maschinencode:

11010011 D3
 n xx

Beschreibung: Der Akkumulatorinhalt wird an den Ausgabekanal mit der Adresse n gegeben.

M-Zyklen: 3

T-Zyklen: 11

Flag-Register: wird nicht beeinflusst

Beispiel: n sei 7CH und A enthalte 12H. Nach der Ausführung des Befehls OUTn enthält der Ausgabekanal mit der Adresse 7CH den Wert 12H.

INIR

Operation: (HL) ← (C), B ← B – 1,
HL ← HL + 1

Maschinencode:

11101101 ED
10110010 B2

Beschreibung: Die durch HL adressierte Speicherstelle wird mit dem Inhalt des durch C adressierten Eingabekanals geladen. Anschließend wird B um 1 vermindert und HL um 1 erhöht. Dieser Vorgang wird so lange wiederholt, bis B = 0 erreicht ist.

M-Zyklen: 5 T-Zyklen: 21 für B ≠ 0

M-Zyklen: 4 T-Zyklen: 16 für B = 0

Flag-Register:

S, H, P/V – nicht definiert

Z, N – 1 gesetzt

C – wird nicht beeinflusst

Beispiel: C enthalte 09H, B = 03H, HL = 3000H. Die Bytes 52H, 67H und 42H werden über das Eingabeter mit der Adresse 09H zugeführt. Nach dem Befehl INIR ergibt sich

(3000H) = 52H

(3001H) = 67H

(3002H) = 42H

HL = 3003H, B = 0

IND

Operation: (HL) ← (C), B ← B – 1,
HL ← HL – 1

Maschinencode:

11101101 ED
10101010 AA

Beschreibung: Die durch HL adressierte Speicherstelle wird mit dem Inhalt des durch C adressierten Eingabekanals geladen. Anschließend werden B und HL um 1 vermindert.

M-Zyklen: 4

T-Zyklen: 16

Flag-Register:

S, H, P/V – nicht definiert

Z – 1, falls B – 1 = 0, sonst 0

N – 1 gesetzt

C – wird nicht beeinflusst

Beispiel: Der Inhalt von C sei 96H, HL sei 3C40H, B = 25H und der Eingabekanal mit der Adresse 96H enthalte 11H. Nach der Ausführung des Befehls ergibt sich (3C40H) = 11H, HL = 3C3FH und B = 24H.

INDR

Operation: (HL) ← (C), B ← B – 1,
HL ← HL – 1

Maschinencode:

11101101 ED
10111010 BA

Beschreibung: Die durch HL adressierte Speicherstelle wird mit dem Inhalt des durch C adressierten Eingabekanals geladen. Anschließend werden B und HL um 1 vermindert. Dieser Vorgang wird so lange wiederholt, bis B = 0 erreicht ist.

M-Zyklen: 5 T-Zyklen: 21 für B ≠ 0

M-Zyklen: 4 T-Zyklen: 16 für B = 0

Flag-Register:

S, H, P/V – nicht definiert

Z, N – 1 gesetzt

C – wird nicht beeinflusst

Beispiel: HL sei 1106H, B = 03H, C = 17H.

Die Daten 22H, 31H, 37H werden über das Eingabeter mit der Adresse 17H zur Verfügung gestellt. Nach Ausführung des Befehls ergibt sich:

(1106H) = 22H, (1105H) = 31H,

(1104H) = 37H, HL = 1103H und B = 00H.

INI

Operation: (HL) ← (C), B ← B - 1,
HL ← HL + 1

Maschinencode:

11101101 ED

10100010 A2

Beschreibung: Die durch HL adressierte Speicherstelle wird mit dem Inhalt des durch C adressierten Eingabekanals geladen. Anschließend wird B um 1 vermindert und HL um 1 erhöht.

M-Zyklen: 4

T-Zyklen: 16

Flag-Register:

S, P/V, H – nicht definiert

Z – 1, falls B - 1 = 0, sonst 0

N – 1 gesetzt

C – wird nicht beeinflusst

Beispiel: HL enthalte 7632H, C = 44H, B = 05H, und der Eingabekanal mit der Adresse 44H enthalte CBH. Nach der Ausführung des Befehls INI ergibt sich HL = 7633H, B = 04H, (7632H) = CBH.

INF

Operation: F ← (C)

Maschinencode:

11101101 ED

01110000 70

Beschreibung: Das Flag-Register wird mit dem Inhalt des durch C adressierten Eingabekanals geladen.

M-Zyklen: 9

T-Zyklen: 12

Flag-Register: siehe Beschreibung

Beispiel: C enthalte 24H, und der Eingabekanal mit der Adresse 24H enthalte 81H. Nach der Ausführung des Befehls INF ergibt sich F = 81H.

OUTr

Operation: (C) ← r

Maschinencode:

11 101 101 ED

01 ← r → 001 41 - 79

Beschreibung: An den durch Register C adressierten Ausgabekanal wird der Inhalt des Registers r gegeben.

M-Zyklen: 3

T-Zyklen: 12

Flag-Register: werden nicht beeinflusst

Beispiel: Der Inhalt von C sei 02H und Register D enthalte A5H. Nach der Ausführung des Befehls OUTr enthält der Ausgabekanal mit der Adresse 02H die Daten A5H.

OUTI

Operation: (C) ← (HL), B ← B - 1,
HL ← HL + 1

Maschinencode:

11101101 ED

10100011 A3

Beschreibung: Der Inhalt der durch HL adressierten Speicherstelle wird an den durch C adressierten Ausgabekanal gegeben. Anschließend wird B um 1 vermindert und HL um 1 erhöht.

M-Zyklen: 4

T-Zyklen: 16

Flag-Register:

S, H, P/V – nicht definiert

Z – 1, falls B - 1 = 0, sonst 0

N – 1 gesetzt

C – wird nicht beeinflusst

Beispiel: HL sei 7185H, (7185H) = 77H, C = 19H und B = 06H. Nach der Ausführung des Befehls ergibt sich: HL = 7186H, B = 05H, und an den Ausgabekanal mit der Adresse 19H sind die Daten 77H gegeben worden.

OUTD

Operation: (C) ← (HL), B ← B - 1,
HL ← HL - 1

Maschinencode:

11101101 ED

10101011 AB

Beschreibung: Der Inhalt der durch HL adressierten Speicherstelle wird an den durch C adressierten Ausgabekanal gegeben. Anschließend werden B und HL um 1 vermindert.

M-Zyklen: 4

T-Zyklen: 16

Flag-Register:

S, H, P/V – nicht definiert

Z – 1, falls B – 1 = 0, sonst 0

N – 1 gesetzt

C – 1 wird nicht beeinflusst

Beispiel: C sei F1H, HL = 7100H, (7100H) = 37H und B = 0CH. Nach Ausführung des Befehls OUTD ergibt sich HL = 70FFH, B = 08H, und an das Ausgabeter mit der Adresse F1H sind die Daten 37H gegeben worden.

OTIR

Operation: $(C) \leftarrow (HL), \quad B \leftarrow B - 1, \quad HL \leftarrow HL + 1$

Maschinencode:

11101101 ED

10110011 B3

Beschreibung: Der Inhalt der durch HL adressierten Speicherstelle wird an den durch C adressierten Ausgabekanal gegeben. Anschließend wird B um 1 vermindert und HL um 1 erhöht. Dieser Vorgang wird so lange wiederholt, bis B = 0 erreicht ist.

M-Zyklen: 5 T-Zyklen: 21

M-Zyklen: 4 T-Zyklen: 16

Flag-Register:

S, H, P/V – nicht definiert

Z, N – 1 gesetzt

C – wird nicht beeinflusst

Beispiel: C enthalte 08H, B = 03H, HL = 1100H und (1100H) = 20H, (1101H) = 27, (1102H) = 25H. Nach der Ausführung des Befehls OTIR sind die Daten 20H, 27H und 25H an das Ausgabeter mit der Adresse 0BH gegeben worden und HL = 1103H und B = 00H.

OTDR

Operation: $(C) \leftarrow (HL), \quad B \leftarrow B - 1, \quad HL \leftarrow HL - 1$

Maschinencode:

11101101 ED

10111011 BB

Beschreibung: Der Inhalt der durch HL adressierten Speicherstelle wird an den durch C adressierten Ausgabekanal gegeben. Anschließend werden B und HL um 1 vermindert. Dieser Vorgang wird so lange wiederholt, bis B = 00H erreicht ist.

M-Zyklen: 5 T-Zyklen: 21 für B ≠ 0

M-Zyklen: 4 T-Zyklen: 16 für B = 0

Flag-Register:

S, H, P/V – nicht definiert

Z, N – 1 gesetzt

C – wird nicht beeinflusst

Beispiel: Der Inhalt von C sei 12H, B = 03H, HL = 0230H und (0230H) = 87H, (022FH) = 42H, (022EH) = C0H. Nach der Ausführung des Befehls OTDR ergibt sich HL = 022DH, B = 00H, und an das Ausgabeter mit der Adresse 12H sind die Daten 87H, 42H und C0H gegeben worden.

Bemerkung: Bei den Eingabe- und Ausgabebefehlen liegt die Kanaladresse auf dem unteren Adressenbus A0...A7. Auf der oberen Hälfte des Adressenbusses A8...A15 liegt bei IN_n und OUT_n der Akkumulatorinhalt A, bei den Befehlen IN_r, INI, INIR, IND, INDR der nicht dekrementierte Wert B und bei den Befehlen OUT_r, OUTI, OTIR, OUTD, OTDR der dekrementierte Wert des Registers B.

5.2. Programme

5.2.1. Programm IN/OUT-Monitor

Assemblerausdruck

EBAS-88

LOC	CODE	LINE	SOURCE
		00001	PN IO
		00002	
		00003	
		00004	
		00005	
		00006	
		00007	
		00008	TITL 'I-O MONITOR Y23V0'
		00009	
		00010	
		00011	;DEFINITION VON KONSTANTEN
		00012	
		00013	ROM: EQU 00000H
		00014	RAM: EQU ROM+0C00H
		00015	RESTR: EQU ROM+400H
		00016	BRAM: EQU 0FC00H ;BILDSCHIRM-RAM
		00017	BOM: EQU ROM+0F00H ;SYSTEMRAM
		00018	STACK: EQU ROM+0FCEH
		00019	RSLOC: EQU ROM+0FE3H ;USER-STACK
		00020	CURS: EQU ROM+0FFF1H ;AKT.BILDSCHIRMPLATZ
		00021	FLGIO: EQU ROM+0FFFH ;SPEICHERPLATZ F.STEUERUNG I/O-GERAETE
		00022	IOBYT: EQU ROM+0FFFH ;SPEICHERPLATZ F.ZUORDNUNG I/O-GERAETE
		00023	RANZK: EQU ROM+0FFCH
		00024	
		00025	PIOAD: EQU 0FCH ;DATENAUSGABE KANAL A
		00026	PIOBD: EQU 0FDH ;EINGABEDATEN KANAL B
		00027	PIOAC: EQU 0F8H ;STEUERADRESSE KANAL A
		00028	PIOBC: EQU 0FFH ;STEUERADRESSE KANAL B
		00029	
		00030	IMSK: EQU 0F3H ;MASKE FUER IN-KANAL (TERMINAL/EXT.SPEICHER)
		00031	OMSK: EQU 0CFH ;MASKE FUER OUT-GERAET (TERMINAL/EXT.SPEICHER)
		00032	LMSK: EQU 03FH ;MASKE FUER LIST-GERAET (TERMINAL/DRUCKER)
		00033	IT: EQU 0 ;"IN"-GERAET I = TERMINAL (TASTATUR)
		00034	IL: EQU 0 ;"IN"-GERAET I = KASSETTENREKORDER
		00035	IU: EQU 0CH ;"IN"-GERAET I = VOM ANWENDER DEF. GERAET
		00036	OT: EQU 0 ;"OUT"-GERAET O = TERMINAL (BILDSCHIRM)
		00037	OL: EQU 020H ;"OUT"-GERAET O = KASSETTENREKORDER
		00038	OU: EQU 030H ;"OUT"-GERAET O = VOM ANWENDER DEF. GERAET
		00039	LT: EQU 0 ;LIST-GERAET L = TERMINAL (BILDSCHIRM)
		00040	LL: EQU 080H ;LIST-GERAET L = FERNSCHREIBER
		00041	LU: EQU 0C0H ;LIST-GERAET L = VOM ANWENDER DEF. GERAET
		00042	CR: EQU 00DH ;WAGEN-RUECKLAUF
		00043	LF: EQU 00AH ;ZEILENVORSCHUB
		00044	BTX: EQU 003H ;CONTROL/C (END OF TEXT)
		00045	APST: EQU 27H ;APOSTROPH
		00046	ULOCR: EQU ROM+0FF9H
		00047	ASS: EQU 0C400H ;ASSEMBLER
		00048	BASIC: EQU 8000H
		00049	PRINT: EQU 00800H ;REASSEMBLER
		00050	TEXT: EQU 0C000H ;TEXTEDITOR
		00051	TRAM: EQU 0EFA0H ;SPEICHERUMSCHALTUNG
		00052	BOT: EQU 0FFH ;END OF TABL
		00053	
		00054	EJEC
		00055	ORG ROM
		00056	BEGIN: EQU #
		00057	
0000	21CE0F	00058	LD HL,STACK ;UEBERTRAGEN DER SYSTEMTABELLEN IN DEN RAM
0003	11CE03	00059	LD DE,TOS ;ZIEL
0006	013200	00060	LD BC,LTO5 ;QUELLE
0009	F9	00061	LD SP,HL ;LAENGE
000A	EB	00062	EX DE,HL
000B	EDB0	00063	LDIR
		00064	;PROGRAMMIERUNG PIO FUER TASTATUR
000D	0603	00065	LD B,LPIOA ;LAENGE
000F	0E8E	00066	LD C,PIOAC ;ADRESSE PORT A COMMANDO
0011	210101	00067	LD HL,TPIOA ;TAB.-ANFANG
0014	EDB3	00068	OTIR
0016	0603	00069	LD B,LPIOB
0018	0C	00070	INC C

5. Anhang

```

0019 EDB3      00071      OTIR
001B 21B30F    00072      LD HL,RSLOC          ;USER-STACK
001E 74        00073      LD (HL),H
001F 35        00074      DEC (HL)
0020 CD7D02     00075      CALL BLOE
0023 213B00     00076      LD HL,RUFZ          ;CALL-AUSGABE
0026 CD2801     00077      CALL STR
0029 1B1D      00078      JR START-#
00079          00079      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00080          00080      ORG 38H+ROM
0038 C30004     00081      JMP RESTR
00082          00082      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00083          00083      RUFZ: EQU #
003B          00084      DB 'Y23'
003E          00085      DB 'V0'
0040 00        00086      DB 00H
00087          00087      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
0041 31CE0F     00088      ERROR: LD SP,STACK          ;STACK RESET
0044 CD2101     00089      CALL COMC
0047          00090      DB '?'          ;FEHLERMELDUNG
00091          00091      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00092          00092      ;ANFORDERUNG EINES KOMMANDOS
0048 CD2D03     00093      START: CALL CRLF          ;LEERZEILE
004B CD3603     00094      CALL TON
004E CD2101     00095      CALL COMC
0051          00096      DB '!'
00097          00097      ;RETZEN DER STARTADRESSE
0052 114800     00098      LD DE,START
0055 D5        00099      PUSH DE
00100          00100      ;EINGABE PROGRAMMNAME
0056 CD0C01     00101      CALL TI          ;ZEICHENEINGABE+ECHO
0059 219600     00102      LD HL,JPTBL          ;SPRUNGTABELLE
005C CD7A00     00103      CALL FCQA          ;SUCHE
005F 3B80      00104      JRC ERROR-#          ;WENN NICHT GEFUNDEN
0061 0E02      00105      LD C,2          ;PARAMETERANZAHL
0063 E9        00106      JMP (HL)
00107          00107      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00108          00108
00109          00109
00110          00110      SUBROUTINEN
00111          00111
00112          00112
00113          00113      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00114          00114      ORG 66H+ROM
0066 21F90F     00115      ;NMI-EINSPRUNG
0069 E3        00116      LD HL,UOCR
006A FB        00117      EX (SP),HL
006B ED45      00118      EI
00119          00119      RETN
00120          00120      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00121          00121      ;ZEITSCHLEIFE
006D B5        00122      ZS: EQU #
006E F5        00123      PUSH HL
006F 2AF00F     00124      PUSH AF
0072 2B        00125      LD HL,(RAMZK)
0073 7C        00126      ZS1: DEC HL
0074 B5        00127      LD A,H
0075 20FB       00128      OR L
0077 F1        00129      JRNZ ZS1-#
0078 B1        00130      POP AF
0079 C9        00131      POP HL
00132          00132      RET
00133          00133      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00134          00134      ;SUBROUTINE VERZWEIGUNG
00135          00135      ;IN: ZEICHEN IN A
00136          00136      ; TABELLENADRESSE IN HL
00137          00137      ;OUT: CY=0:HL=ADR,CY=1:NOT FOUND
007A D5        00138      FCQA: EQU #
007B 5F        00139      PUSH DE
00140          00140      LD E,A          ;ZEICHEN
00141          00141      FCQA1: EQU #
007C 7E        00142      LD A,(HL)
007D FEFF      00143      CMP EOT          ;END OF TAB
007F 2B11      00144      JRZ FCQA3-#
0081 BB        00145      CMP E
0082 2805      00146      JRZ FCQA2-#          ;FOUND
0084 23        00147      INC HL
0085 23        00148      INC HL
0086 23        00149      INC HL
0087 1BF3      00150      JR FCQA1-#
00151          00151      FCQA2: EQU #
0089 7B        00152      LD A,E

```



```

008A 23      00153      INC  HL
008B 5E      00154      LD   E,(HL)
008C 23      00155      INC  HL
008D 56      00156      LD   D,(HL)
008E 23      00157      INC  HL
008F EB      00158      EX   DE,HL
0090 1802    00159      JR    PCGA4-#
                                ;NOT FOUND
0092 37      00160      FCGA3: EQU #
0093 7B      00161      SCF
                                ;
0094 D1      00162      LD   A,E
0095 C9      00163      FCGA4: EQU #
                                ;
                                POP DE
                                RET
                                00165
                                00166
                                00167
                                ;ADRESSTABELLE KOMMANDOS
                                JPTBL: EQU #
0096         00169      DB   'A'
0097 00C4    00170      DA   ASS
0099         00171      DB   'B'
009A 0080    00172      DA   BASIC
009C         00173      DB   'C'
009D A503    00174      DA   COPY
009F         00175      DB   'D'
00A0 6405    00176      DA   DE.DISP
00A2         00177      DB   'E'
00A3 8505    00178      DA   DE.EOF
00A5         00179      DB   'F'
00A6 A905    00180      DA   DE.FILL
00A8         00181      DB   'G'
00A9 B705    00182      DA   DE.GOTO
00AB         00183      DB   'H'
00AC F405    00184      DA   DE.HEXN
00AE         00185      DB   'I'
00AF 6B03    00186      DA   INK
00B1         00187      DB   'J'
00B2 F90F    00188      DA   ULOC
00B4         00189      DB   'K'
00B5 8C06    00190      DA   DE.KNOW
00B7         00191      DB   'L'
00B8 5203    00192      DA   LKOM
                                ;LIST-ZUWEISUNG
00BA         00193      DB   'M'
00BB 0B06    00194      DA   DE.MOVE
00BD         00195      DB   'O'
00BE 8C03    00196      DA   OUTK
00C0         00197      DB   'P'
00C1 0008    00198      DA   PRINT
                                ;DISASSEMBLER
00C3         00199      DB   'R'
00C4 1E06    00200      DA   DE.READ
00C6         00201      DB   'S'
00C7 7006    00202      DA   DE.SUBS
00C9         00203      DB   'T'
00CA 00C0    00204      DA   TEXT
00CC         00205      DB   'V'
00CD E906    00206      DA   DE.VER
00CF         00207      DB   'W'
00D0 AA06    00208      DA   DE.WRITE
00D2         00209      DB   'X'
00D3 0207    00210      DA   DE.X
00D5         00211      DB   'Y'
00D6 A0EF    00212      DA   TRAM
00D8         00213      DB   'Z'
00D9 7907    00214      DA   DE.ZR
00DB FF      00215      DB   BOT
                                ;-----*
00216      ;SPRUNGTABELLE IN/OUT-ROUTINEN
00217      ORG 050H
00218      MCI: JMP CI ;CONSOLE INPUT
00219      MRI: JMP RI ;READER INPUT
00220      MCO: JMP CO ;CONSOLE OUTPUT
00221      MWO: JMP WO ;WRITE OUTPUT
00222      MLO: JMP LO ;LISTER OUTPUT
00223      MCSTS: JMP CSTS
00224      MIOC: JMP IOCHK
00225      MIOS: JMP IOSET
                                ;-----*
00226      ;SUBR. ZUR ABFRAGE DER AKTUELLEN GERÄTEZUWEISUNG
00227      ;***IN: -
00228      ;**OUT: A...AKTUELLE ZUWEISUNG
00229      ;*KILL: A
00230      IOCHK: LD A,(IOBYT)
00231      RET
00232
00F8 3AFF0F  00233
00FB C9      00234

```

5. Anhang

```

00235 ;*-----*-----*-----*-----*-----*
00236 ;SUBR. ZUR VERÄNDERUNG DER AKT. GERÄTEZUWEISUNG
00237 ;***IN: C...NEUE GERÄTEZUWEISUNG
00238 ;**OUT: -
00239 ;*KILL: A
00FC 79 00240 IOSET: LD A,C
00FD 32FF0F 00241 LD (IOBYT),A
0100 C9 00242 RET
00243 ;*-----*-----*-----*-----*-----*
00244 ;PIO TABELLEN
00245 TPIOA: EQU #
0101 CF 00246 DB 0CFH ;BITWEISE E/A
0102 C0 00247 DB 0COH ;1-IN,0-OUT
0103 07 00248 DB 007H ;INT-STEUERWORT
00249 LPIOA: EQU #-TPIOA
00250 TPIOB: EQU #
0104 CF 00251 DB 0CFH
0105 FF 00252 DB 0FFH
0106 07 00253 DB 07H
00254 LPIOB: EQU #-TPIOB
00255 ;*-----*-----*-----*-----*-----*
00256 ;SUBR. ZUM TEST, OB UNTERBRECHUNG IETIO EINGEGEBEN WURDE
00257 ;***IN: -
00258 ;**OUT: A=0...KEIN ZEICHEN
00259 ; A SONST...ZEICHEN
0107 CD1C03 00260 ;*KILL: A,F
010A B7 00261 BREAK: CALL CSTS
010B C8 00262 OR A
00263 OR
00264 ;*-----*-----*-----*-----*-----*
00265 ;SUBR. ZUR EINGABE EINES ASCII-ZEICHENS ÜBER DAS TERMINAL
00266 ;UND AUSGABE DES ECHOS
00267 ;***IN: -
00268 ;**OUT: A...ZEICHEN
010C C5 00269 ;*KILL: A,F
010D CDE000 00270 TI: PUSH BC
0110 FE03 00271 CALL MCI
0112 CA4800 00272 CMP BTX
0115 4F 00273 JPZ START
0116 CDE600 00274 LD C,A
0119 79 00275 CALL MCO
011A CB77 00276 LD A,C
011C C4BC01 00277 BIT 6,A ;KLEINBUCHSTABE ?
011F C1 00278 CANZ SH2
0120 C9 00279 POP BC
00280 RET
00281 ;*-----*-----*-----*-----*-----*
00282 ;SUBR. Z. AUSG. E. ZEICHENS ÜB. DAS TERMINAL
00283 ;***IN: ZEICHEN AUF SPEICHERPLATZ HINTER CALL
00284 ;**OUT: C...ZEICHEN
00285 ; (SP)=(SP)+1
0121 E3 00286 ;*KILL: A,F,C
0122 4E 00287 COMC: EX (SP),HL
0123 23 00288 LD C,(HL)
0124 E3 00289 INC HL
0125 C3E600 00290 EX (SP),HL
00291 JMP MCO
00292 ;*-----*-----*-----*-----*-----*
00293 ;SUBROUTINE ZUR AUSGABE EINER ZEICHENFOLGE,ABSCHLUSS DER FOLGE MIT 00
00294 ;***IN: ANFANGSADR. IN HL
00295 ;**OUT: -
00296 ;*KILL: A,F,HL
0128 7E 00297 STR: LD A,(HL)
0129 4F 00298 LD C,A
012A 23 00299 INC HL
012B B7 00300 OR A
012C C8 00301 RZ
012D CDEC00 00302 CALL MLO
0130 18F6 00303 JR STR-#
00304 ;*-----*-----*-----*-----*-----*
00305 ;SUBR. ZUR EINGABE EINES ZEICHENS ÜBER DAS TERMINAL
00306 ;***IN: -
00307 ;**OUT: A...ZEICHEN IN ASCII
00308 ;*KILL: A,F
00309 CI: EQU #
0132 CD3C01 00310 CALL CIO
0135 CB77 00311 BIT 6,A ;UMWANDLUNG KLEIN/GROSS
0137 C4BC01 00312 CANZ SH2
013A B7 00313 OR A
013B C9 00314 RET
013C C5 00315 CIO: PUSH BC
013D D5 00316 PUSH DE

```

```

013E B5      00317      PUSH HL
013F 21FE0F  00318      LD HL,FLGIO
0142 CD6901  00319      CI2: CALL ZYKL
0145 28FB    00320      JRZ CI2-#
          00321      ;TASTE GEDRUECKT
0147 57      00322      LD D,A
0148 CD8B01  00323      CALL CVTA          ;:- ASCII
014B F5      00324      PUSH AF
014C CD3603  00325      CALL TON
014F CD6901  00326      CI1: CALL ZYKL
0152 20FB    00327      JRNZ CI1-#
          00328      ;TASTE FREI
0154 F1      00329      POP AF
          00330      ;UMSCHALTASTE ?
0155 FE11    00331      CMP 11H          ;DC1=SHIFT
0157 2808    00332      JRZ CI3-#
0159 FE12    00333      CMP 12H          ;DC2=CONTROL
015B 2808    00334      JRZ CI4-#
015D B1      00335      POP HL
015E D1      00336      POP DE
015F C1      00337      POP BC
0160 C9      00338      RET
0161 CBDE    00339      CI3: SET 3,(HL)
0163 18DD    00340      JR CI2-#
0165 CBEE    00341      CI4: SET 4,(HL)
0167 18D9    00342      JR CI2-#
          00343      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
0169 0608    00344      ;SUBR. FUER EINEN ZYKLUS DER TASTENEINGABE
016B 0EFC    00345      ;**IN: -
016D ED78    00346      ;**OUT: A=0...KEINE EINGABE
016F B6F1    00347      ;      A...SONST SPALTENBYTE
0171 5F      00348      ;      B...ZEILENBYTE
0172 ED59    00349      ;*KILL: A,B,C,E,F
0174 CD6D00  00350      ZYKL: EQU #
0177 DBFD    00351      LD B,08H
0179 FEFF    00352      LD C,PIOD
017B 2007    00353      IN A
017D 7B      00354      AND OP1H
017E C602    00355      ZYK1: LD E,A
0180 10EF    00356      OUT E
0182 AF      00357      CALL ZS
0183 C9      00358      IN PIOBD
0184 1EFE    00359      CMP OFPH
0186 CB0B    00360      JRNZ ZYK2-#          ;TASTE GEDRUECKT
0188 10FC    00361      LD A,E
018A C9      00362      ADD 02H
018B 4F      00363      DJNZ ZYK1-#          ;KEINE TASTE GEDRUECKT
018C AF      00364      XOR A
018D 47      00365      RET
018E 1EFE    00366      ZYK2: LD E,OFBH
018F CB0B    00367      ZYK3: RRC E
0190 10FC    00368      DJNZ ZYK3-#
0191 C9      00369      RET
          00370      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
0192 0608    00371      ;SUBR. ZUR KONVERTIERUNG TASTENKODE -> ASCII
0193 0EFC    00372      ;**IN: DE...ZEICHEN IM TASTENKODE
0194 ED78    00373      ;**OUT: A...ZEICHEN IM ASCII-KODE
0195 B6F1    00374      ;*KILL: A,B,C,D,E,F
0196 5F      00375      CVTA: PUSH HL
0197 21FE0F  00376      XOR A
0198 CD8B01  00377      LD B,A          ;B=0 F.ADD HL,BC
0199 2808    00378      ;REL. TAB.PLATZ FUER SPALTE
019A CD6901  00379      CVTA0: ADD 7
019B 2808    00380      RRC D
019C 38FA    00381      JRC CVTA0-#
019D D607    00382      SUB 7
019E 4F      00383      LD C,A
019F 38FA    00384      ;REL. TAB.PLATZ FUER ZEILE
0200 21FE0F  00385      XOR A
0201 3C      00386      CVTA1: INC A
0202 CB0B    00387      RRC E
0203 38FB    00388      JRC CVTA1-#
0204 3D      00389      DEC A
0205 21FE0F  00390      ;REL. TAB.PLATZ FUER ZEICHEN IN C
0206 81      00391      ADD C
0207 4F      00392      LD C,A
0208 21C501  00393      LD HL,TTAB
0209 09      00394      ADD HL,BC
020A 7E      00395      LD A,(HL)
020B 21FE0F  00396      ;TEST AUF UMSCHALTUNG
020C CB5E    00397      LD HL,FLGIO
020D 09      00398      BIT 3,(HL)

```

5. Anhang

144

```

00481 ;*KILL: A,F
00482 CO: EQU #
01FF F5 00483 PUSH AF
0200 C5 00484 PUSH BC
0201 D5 00485 PUSH DE
0202 E5 00486 PUSH HL
0203 DDE5 00487 PUSH IX
0205 114000 00488 LD DE,64 ;1ZEILE
0208 2AF10F 00489 LD HL,(CURS) ;CURSOR POSITION LADEN
020B E5 00490 PUSH HL ;CURSORPOSITION IN STACK
020C 7E 00491 LD A,(HL) ;ZEICHEN HOLEN
020D E67F 00492 AND 7FH ;CURSOR LOESCHEN
020F 77 00493 LD (HL),A
0210 79 00494 LD A,C ;NEUES ZEICHEN
0211 E67F 00495 AND 7FH
0213 214E02 00496 LD HL,COTAB ;STEUERZEICHEN ?
0216 CD7A00 00497 CALL PCGA
0219 3806 00498 JRC C00-# ;NEIN
00499 ;STEUERZEICHEN AUSWERTEN
021B E5 00500 PUSH HL
021C DDE1 00501 POP IX
021E E1 00502 POP HL
021F DDE9 00503 JMP (IX)
0221 79 00504 C00: LD A,C
0222 E1 00505 POP HL ;CURSORPOSITION LADEN
0223 77 00506 LD (HL),A ;ZEICHEN AUSGEBEN
0224 23 00507 HTAB: INC HL ;NAECHSTER BILDSCHIRMPLATZ
0225 CD6402 00508 C03: CALL VOLL
0228 7E 00509 C01: LD A,(HL) ;ZEICHEN RUECKLESEN
0229 C680 00510 ADD 80H ;CURSOR SETZEN
022B 77 00511 LD (HL),A ;ZEICHEN AUSGEBEN
022C 22F10F 00512 C02: LD (CURS),HL
022F DDE1 00513 POP IX
0231 E1 00514 POP HL
0232 D1 00515 POP DE
0233 C1 00516 POP BC
0234 F1 00517 POP AF
0235 C9 00518 RET
00519 ;STEUERZEICHEN
0236 2B 00520 BS: DEC HL ;BACKSTEP
0237 18EF 00521 JR C01-#
0239 7D 00522 WAG: LD A,L ;WAGENRUECKLAUF
023A B6C0 00523 AND 0COH ;ZEILENANFANG EINSTELLEN
023C 6F 00524 LD L,A
023D 18E6 00525 JR C03-#
023F 19 00526 ZVOR: ADD HL,DE
0240 18E3 00527 JR C03-#
0242 ED52 00528 CUP: SBC HL,DE
0244 18E2 00529 JR C01-#
0246 19 00530 NL: ADD HL,DE
0247 18F0 00531 JR WAG-#
0249 CD3603 00532 BELL: CALL TON
024C 18DA 00533 JR C01-#
00534 ;STEUERZEICHENTABELLE
00535 COTAB: EQU #
024E 0D 00536 DB CR ;CARRIAGE RETURN
024F 3902 00537 DA WAG ;WAGENRUECKLAUF
0251 0A 00538 DB LF ;LINE FEED
0252 3F02 00539 DA ZVOR ;ZEILENVORSCHUB
0254 0B 00540 DB OBH ;VERT.TABULATOR
0255 4202 00541 DA CUP ;CURSOR UP
0257 09 00542 DB 09 ;HOR. TABULATOR
0258 2402 00543 DA HTAB
025A 0B 00544 DB 0B ;BACKSTEP
025B 3602 00545 DA BS
025D 1E 00546 DB 1EH ;NEW LINE
025E 4602 00547 DA NL
0260 07 00548 DB 7 ;BELL
0261 4902 00549 DA BELL
0263 FF 00550 DB OFFH
00551 ;TEST AUF BILDSCHIRNUMUEBERLAUF
0264 7C 00552 VOLL: LD A,H
0265 FE00 00553 CMP H(BRAM+1024)
0267 C0 00554 RNZ ;NOCH KEIN UEBERLAUF
00555 ;ROLL UP
00556 ROLA: EQU #
0268 C5 00557 PUSH BC
0269 2140FC 00558 LD HL,BRAM+64
026C 1100FC 00559 LD DE,BRAM
026F 01C003 00560 LD BC,1024-64
0272 EDB0 00561 LDIR
00562 ;LETZTE ZEILE LOESCHEN

```

5. Anhang

```

00563 ;IN: HL ZEILENENDE
00564 ;OUT: HL - 64
0274 012040 00565 LD BC,4020H ;64 BYTES, LEBERZEICHEN
0277 2B 00566 ZLOB: DEC HL
0278 71 00567 LD (HL),C
0279 10FC 00568 DJNZ ZLOB-#
027B C1 00569 POP BC
027C C9 00570 RET
027D E5 00571 BLOB: PUSH HL
027E D5 00572 PUSH DE
027F C5 00573 PUSH BC
0280 21FFFF 00574 LD HL,BRAM+1023 ;BILDSCHIRM LOESCHEN
0283 11FEFF 00575 LD DE,BRAM+1022
0286 01FF03 00576 LD BC,1023 ;ANZAHL
0289 3620 00577 LD (HL),' ' ;LEBRZEICHEN
028B EDB8 00578 LDDR
028D 22F10F 00579 LD (CURS),HL
0290 C1 00580 POP BC
0291 D1 00581 POP DE
0292 E1 00582 POP HL
0293 C9 00583 RET
00584 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00585 ;SUBR. ZUR EINGABE EINES ZEICHENS UEBER DAS GERAET "I"
00586 ;***IN: -
00587 ;**OUT: A...ZEICHEN
00588 ; CARRY=1...TIME OUT
00589 ;*KILL: A,F
0294 3AFF0F 00590 RI: LD A,(IOBYT)
0297 E60C 00591 AND OCH ;NEG. IMSK
00592 ;EINGABE UEBER TERMINAL
0299 CAE000 00593 JPZ MCI
029C FE08 00594 CMP IL
029E 2023 00595 JRNZ R18-#
00596 ;EINGABE UEBER SERIELLE SCHNITTSTELLE
00597 LIN: EQU #
00598 IN P10AD
00599 BIT 7,A
02A2 CB7F 00600 JRZ LIN-# ;WARTEN AUF START-BIT
02A4 28FA 00601 PUSH BC
02A6 C5 00602 LD B,8
02A7 0608 00603 CALL ZS ;BIT-ANZAHL
02A9 CD6D00 00604 LIN1: IN P10AD ;ABTASTUNG MITTE BIT
02AC DBFC 00605 CALL ZS
02AE CD6D00 00606 CALL ZS
02B1 CD6D00 00607 RLA
02B4 17 00608 RR C ;AUS CY-FLAG IN C
02B5 CB19 00609 DJNZ LIN1-#
02B7 10F3 00610 RR C ;STOP-BIT
02B9 CB19 00611 LD A,C
02BB 79 00612 POP BC
02BC C1 00613 CPL
02BD 2F 00614 AND 7FH
02BE E67F 00615 JRZ LIN-#
02C0 28DE 00616 RET
02C2 C9 00617 ;EINGABE UEBER VOM ANWENDER DEF. GER.
02C3 3EF3 00618 R18: LD A,L(RULOC)
02C5 182D 00619 JR HUSER-#
00620 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00621 ;SUBR. ZUR AUSGABE EINES ZEICHENS AUF SERIELLE SCHNITTSTELLE
00622 ;***IN: C...ZEICHEN
00623 ;**OUT: -
00624 ;*KILL: A,F
00625 LDRU: EQU #
00626 PUSH AF
00627 PUSH BC
00628 LD A,C
00629 OR 80H ;STOP-BIT HINZU
02C7 F5 00630 LD B,9 ;BIT-ANZAHL MIT START-BIT
02C8 C5 00631 RLA ;CY-FLAG-STARTBIT
02C9 79 00632 FSA1: OUT P10AD
02CA F680 00633 CALL ZS
02CC 0609 00634 CALL ZS
02CE 17 00635 RRA
02CF D3FC 00636 DJNZ FSA1-#
02D1 CD6D00 00637 POP BC
02D4 CD6D00 00638 POP AF
02D7 1F 00639 RET
02D8 10F5 00640 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
02DA C1 00641 ;SUBR. Z. AUSG. E. ZEICH. UEBER DAS GERAET "O"
02DB F1 00642 ;***IN: C...ZEICHEN
02DC C9 00643 ;**OUT: -
00644 ;*KILL: A,F
02DD CBB9 00645 W0: RES 7,C

```

```

02DF 79      00646      LD    A,C
02E0 B7      00647      OR     A
02E1 EAE602  00648      JFPE W00
02E4 CBF9    00649      SET   7,C
02E6 3AFFOF  00650      W00:   LD    A,(IOBYT)
02E9 E630    00651      AND    30H      ;NEG. OMSK
02EB CAE600  00652      W01:   JPZ   MCO      ;AUSGABE UEBER TERMINAL
02EE FE20    00653      CMP    OL
02F0 28D5    00654      JRZ   LDRU-#
00655      ;AUSGABE UEBER VOM ANWENDER DEF. GER.
02F2 3EF6    00656      LD     A,L(PULOC)
00657      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00658      ;SUBR. ZUM AUFRUF EINES VOM ANWENDER DEF. HANDLERS
02F4 E5      00659      HUSER:  PUSH  HL
02F5 21FFOF  00660      LD     HL,IOBYT
02F8 6F      00661      LD     L,A
02F9 E3      00662      EX     (SP),HL
02FA C9      00663      RET
00664      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00665      ;SUBR. ZUR AUSGABE VON CR UND LF AUF DEM GERÄT "L"
00666      ;**IN: -
00667      ;**OUT: -
00668      ;*KILL: A,F,C
02FB 0E0D    00669      LCRLP: LD    C,CR
02FD CDECOO  00670      CALL  MLO
0300 0E0A    00671      LD     C,LF
00672      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00673      ;SUBR. ZUR AUSGABE UEB. D. GER. "L" MIT TEST AUF EINGABE VON CTRL/C
00674      ;**IN: C...ZEICHEN
00675      ;**OUT: -
00676      ;*KILL: A,F
0302 CD0701  00677      LO:    CALL  BREAK
0305 3E1E    00678      LD     A,1EH      ;NEWLINE
0307 B9      00679      CMP    C
0308 28F1    00680      JRZ   LCRLP-#
030A 3AFFOF  00681      LD     A,(IOBYT)
030D E6C0    00682      AND    OCOH      ;NEG. LMSK
030F 28DA    00683      JRZ   W01-#      ;AUSGABE UEBER TERMINAL
0311 FE80    00684      CMP    LL
0313 28B2    00685      JRZ   LDRU-#      ;AUSGABE UEBER DRUCKER
0315 FE00    00686      CMP    LU
0317 3EF9    00687      LD     A,L(ULOC)
0319 28D9    00688      JRZ   HUSER-#      ;AUSGABE UEBER ANWENDERGERÄT
031B C9      00689      RET
00690      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00691      ;SUBR. ZUM TEST AUF ETX-EINGABE UEBER DAS TERMINAL
00692      ;**IN: -
00693      ;**OUT: A=0... KEINE EINGABE
00694      ;      A=-1... EINGABE
00695      ;*KILL: A,F
031C DBFC    00696      CSTS:  IN     PLOAD
031E CBC7    00697      SET   0,A      ;LINIENSTROM EIN
0320 E6F1    00698      AND    0F1H      ;1.ZEILE
0322 D3FC    00699      OUT    PLOAD
0324 DBFD    00700      IN     PIOBD
0326 FEEF    00701      CMP    0E0H      ;5.SPALTE
0328 3EFF    00702      LD     A,0FFH
032A C8      00703      RZ
032B AF      00704      XOR    A
032C C9      00705      RET
00706      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00707      ;SUBR. ZUR AUSGABE VON CR UND LF AUF DEM TERMINAL
00708      ;**IN: -
00709      ;**OUT: -
00710      ;*KILL: A,F,C
032D CD2101  00711      CRLP:  CALL  COMC
0330 0D      00712      DB     CR
0331 CD2101  00713      CALL  COMC
0334 0A      00714      DB     LF
0335 C9      00715      RET
00716      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00717      ;SUBROUTINE TON
00718      ;IN:-
00719      ;OUT:TON AN PIO AD
00720      ;KILL:PIOAD
00721      TON:   EQU    #
00722      PUSH  AF
00723      PUSH  BC
00724      LD     C,0FFH      ;PERIODENANZAHL
00725      LD     B,30H      ;PERIODENLAENGE
00726      TON1:  LD     B,30H
00727      TON2:  DJNZ  TON2-#      ;1/2 PERIODE
0336 F5      00722      IN     PLOAD
0337 C5      00723      IN     PLOAD
0338 0E0F    00724      LD     C,0FFH
033A 0630    00725      LD     B,30H
033C 10FE    00726      DJNZ  TON2-#
033E DBFC    00727      IN     PLOAD

```

5. Anhang

```

0340 CB6F 00728 BIT 5,A
0342 280A 00729 JRZ TON3-#
0344 CBAF 00730 RES 5,A
0346 D3FC 00731 TON4: OUT PLOAD
0348 0D 00732 DEC C
0349 20EF 00733 JRNZ TON1-#
034B C1 00734 POP BC
034C F1 00735 POP AF
034D C9 00736 RET
034E CBEF 00737 TON3: SET 5,A
0350 18F4 00738 JR TON4-#
00739 ;*-----*-----*-----*-----*-----*-----*
00740 ;ZUWEISUNG PERIPHERIEGERÄTE
00741 LKOM: EQU # ;LISTER
00742 LD B,LMSK ;LISTER-MASKE
00743 CALL TI
00744 CMP 'L'
00745 JRZ LKL-#
00746 CMP 'U'
00747 JRZ LKU-#
00748 LD C,LT ;TERMINAL
00749 JR ASGN-#
00750 LKL: LD C,LL
00751 JR ASGN-#
00752 LKU: LD C,LU
00753 JR ASGN-#
00754 INK: EQU # ;INPUT-KANAL
00755 LD B,IMSK
00756 CALL TI
00757 CMP 'U'
00758 JRZ IKU-#
00759 CMP 'L'
00760 JRZ IKL-#
00761 LD C,IT ;TERMINAL
00762 JR ASGN-#
00763 IKL: LD C,IL LINE
00764 JR ASGN-#
00765 IKU: LD C,IU USER
00766 ;GERÄT IM I/O-BYTE EINTRAGEN
00767 ASGN: CALL IOCHK
00768 AND B
00769 OR C
00770 LD C,A
00771 CALL IOSET
00772 RET
00773 OUTK: EQU # ;OUT-KOMMANDO
00774 LD B,OMSK
00775 CALL TI
00776 CMP 'L'
00777 JRZ OKL-#
00778 CMP 'U'
00779 JRZ OKU-#
00780 LD C,OT
00781 JR ASGN-#
00782 OKL: LD C,OL
00783 JR ASGN-#
00784 OKU: LD C,OU
00785 JR ASGN-#
00786 ;*-----*-----*-----*-----*-----*-----*
00787 ;KOMMANDO COPY
00788 COPY: EQU #
00789 CALL MRI
00790 CMP EFX
00791 RZ
00792 LD C,A
00793 CALL MLO
00794 JR COPY-#
00795 ;*-----*-----*-----*-----*-----*-----*
00796 ORG 3CEH+ROM
00797 ;SYSTEMTABELLEN
00798 TOS: EQU #
00799 USER: EQU TOS-8
00800 ;TABELLE DER REGISTERINHALTE DES ANWENDERPROGRAMMS
00801 BLOC: DB 0EEH
00802 DLOC: DB 0DDH
00803 CLOC: DB 0CCH
00804 BLOC: DB 0BBH
00805 FLOC: DB 0FFH
00806 ALOC: DB 0AAH
00807 XLOC: DA 1234H
00808 YLOC: DA 5678H
00809 DB 0

0352 063F 00742 LD B,LMSK
0354 CD0C01 00743 CALL TI
0357 FE4C 00744 CMP 'L'
0359 2808 00745 JRZ LKL-#
035B FE55 00746 CMP 'U'
035D 2808 00747 JRZ LKU-#
035F 0800 00748 LD C,LT ;TERMINAL
0361 181F 00749 JR ASGN-#
0363 0880 00750 LKL: LD C,LL
0365 181B 00751 JR ASGN-#
0367 08C0 00752 LKU: LD C,LU
0369 1817 00753 JR ASGN-#
00754 INK: EQU # ;INPUT-KANAL
00755 LD B,IMSK
036B 06F3 00756 CALL TI
036D CD0C01 00757 CMP 'U'
0370 FE55 00758 JRZ IKU-#
0372 280C 00759 CMP 'L'
0374 FE4C 00760 JRZ IKL-#
0376 2804 00761 LD C,IT ;TERMINAL
0378 0800 00762 JR ASGN-#
037A 1806 00763 IKL: LD C,IL LINE
037C 0808 00764 JR ASGN-#
037E 1802 00765 IKU: LD C,IU USER
0380 080C 00766 ;GERÄT IM I/O-BYTE EINTRAGEN
00767 ASGN: CALL IOCHK
0382 CDF800 00768 AND B
0385 A0 00769 OR C
0386 B1 00770 LD C,A
0387 4F 00771 CALL IOSET
0388 CDFC00 00772 RET
038B C9 00773 OUTK: EQU # ;OUT-KOMMANDO
00774 LD B,OMSK
038C 06CF 00775 CALL TI
038E CD0C01 00776 CMP 'L'
0391 FE4C 00777 JRZ OKL-#
0393 2808 00778 CMP 'U'
0395 FE55 00779 JRZ OKU-#
0397 2808 00780 LD C,OT
0399 0800 00781 JR ASGN-#
039B 18E5 00782 OKL: LD C,OL
039D 0E20 00783 JR ASGN-#
039F 18E1 00784 OKU: LD C,OU
03A1 0E30 00785 JR ASGN-#
03A3 18DD 00786 ;*-----*-----*-----*-----*-----*-----*
00787 ;KOMMANDO COPY
00788 COPY: EQU #
03A5 CDE300 00789 CALL MRI
03A8 FE03 00790 CMP EFX
03AA C8 00791 RZ
03AB 4F 00792 LD C,A
03AC CDEC00 00793 CALL MLO
03AF 18F4 00794 JR COPY-#
00795 ;*-----*-----*-----*-----*-----*-----*
00796 ORG 3CEH+ROM
00797 ;SYSTEMTABELLEN
00798 TOS: EQU #
00799 USER: EQU TOS-8
00800 ;TABELLE DER REGISTERINHALTE DES ANWENDERPROGRAMMS
03CE E8 00801 BLOC: DB 0EEH
03CF DD 00802 DLOC: DB 0DDH
03D0 CC 00803 CLOC: DB 0CCH
03D1 BB 00804 BLOC: DB 0BBH
03D2 FF 00805 FLOC: DB 0FFH
03D3 AA 00806 ALOC: DB 0AAH
03D4 3412 00807 XLOC: DA 1234H
03D6 7856 00808 YLOC: DA 5678H
03D8 00 00809 DB 0

```



```

03D9 0C      00810 ILOC: DB   0CH
03DA 00      00811 LSLC: DB   00H
03DB 00      00812 HSLC: DB   00H
03DC 0E      00813 ESLC: DB   0BH
03DD 0D      00814 DSLC: DB   0DH
03DE 0C      00815 CSLC: DB   0CH
03DF 0B      00816 BSLC: DB   0BH
03E0 0F      00817 FSLC: DB   0FH
03E1 0A      00818 ASLC: DB   0AH
03E2 C6      00819      DB   L(USER)
03E3 00      00820 SLOC: DB   0
03E4 213412  00821 EXI1: LD   HL,1234H
          00822 LLOC: EQU  #-2
          00823 HLOC: EQU  #-1
03E7 FB      00824      EI
03E8 C38967  00825 JMP  6789H
          00826 PLOC: EQU  #-1
          00827 ;*-----*
          00828 ;TABELLE DER HALTEPUNKTE
03EB 0000    00829 TLOC: DA   0
03ED 00      00830 DB   0
03EE 0000    00831 DA   0
03F0 00      00832 DB   0
          00833 ;*-----*
03F1 00FC    00834 PLA:  DA   BRAM
          00835 RULOC: EQU  #
03F3 C30000  00836 JMP  #
          00837 PULOC: EQU  #
03F6 C30000  00838 JMP  #
          00839 ULLOC: EQU  #
03F9 C34800  00840 JMP  START
03FC 18      00841 ZK:   DB   16H
03FD 01      00842 DB   01
03FE 02      00843 FLG:  DB   2
03FF 20      00844 IOB:  DB   20H
          00845 LTOS: EQU  #-TOS
          00846 END
0400
          00001      PN   DE

```

```

;30 F.50 BAUD
;2 F. 50 BAUD
;FLGIO
;IOBYT

```

Maschinencode

```

:1010000021CE0F11CE03013200F9EBEDB006030E35
:10101000FE210101EDB306030CEDB321E30F74359E
:10102000CD7D02213B00CD2801181DFFFFFFFFFFF2
:10103000FFFFFFFFFFFFFFFFF30004593233564F8E
:101040000031CEQFCD21013FCD2D03CD3603CD2173
:10105000013E114800D5CD0C01219600CD7A003813
:10106000E00E02B9FF21F90F3FBED45E752A60
:10107000FC0F2B7CB520FBF1E1C9D55F7EFEEFF287C
:1010800011BB28052323218F37B235E235623EB70
:101090001802377BD1C94100C442008043A50344F4
:1010A000640545850546A90547B70548F405496B1C
:1010B000034AF90F4B8C064C52034D0B064F8C0321
:1010C000500008521E065370065400C056E90657D9
:1010D000AA0658020759A0EF5A7907FFFFFFFFF42
:1010E000C33201C39402C3FF01C3DD02C30203C3C1
:1010F0001C03C3F800C3FC003AFF0FC97932F0F8D
:10110000C9CF0007CFF07CD1C03B7C8C5DEC000CE
:10111000FE03CA48004FCD860079C8B7F04BC01C1BD
:10112000C9E34E23E3C3E6007E4F23B7C8CDECO0BE
:1011300018F6CD3C01CB77C4BC01B7C9C5D5E5214B4
:10114000F80FCD690128FB57CD8B01F5CD3603CD0C
:10115000690120FBF1FE112808FE122808E1D1C127
:10116000C9CBDE18DDCB618D906080EFCED78E613
:10117000F15FD59CD6D00DBDFEFF20077FC60260
:1011800010EFAFC91EFECEB0B10FC9C9547C60779
:10119000C80A38FAD6074FAF3CC80B38FB3D814F1B
:1011A00021C501097E21F80FCB58C48B01CB9ECBC9
:1011B00066C4C201CBA6B1C9C772803CBAAF9C9CBAC
:1011C000A7C9861FC90B706974727709086757431
:1011D0006571203B7F6E76780003C6B86673001B
:1011E00000030D6D2637912116C6A5764613E0AD7
:1011F0003038363432F3A3937353313D0E20F509
:10120000C5D5E5DDE51140002AF10FE57E67F77E3
:1012100079E67F214E02CD7A003806E5DDE1E1DD99
:10122000E979E17723CD64027EC6807722F10FDD74
:10123000E1E1D1C1F1C92B18EF7DE6C06F18E619C5
:1012400018E3ED5218E21918F0CD360318DA0D390B
:10125000020A3F020B42020924020836021E4602D1
:10126000074902FF7CF800C0C52140FC1100FC01C3
:10127000C003EDB00120402B7110FCC1C9E5D5C5FC
:1012800021FFFF11FEFF01FF033620EDB822F10F11
:10129000C1D1E1C93AFF0FE60CCAB000FE082023E5
:1012A000DBFCCB7F28FAC50608CD6D00DBFCCD6DDD
:1012B00000CD6D0017CB1910F3CB1979C12FE67F44
:1012C00028DEC93EF3182DF5C579F680060917D337
:1012D000FCCD6D00CD6D001F10F5C1F1C9CBB97902
:1012E000B7EAB602CBF93AFF0FE630CAE600FE2085
:1012F00028D53BF6E521FF0F6FE3C90E0DCDECO0BA
:101300000E0ACD07013E1EB928F13AFF0FE6C028AC
:10131000DAFE8028B2FEC03EF928D9C9DBFCBCB773
:10132000E6F1D3FCDBFDFEEF3FEFFC8AFC9CD2101E6
:101330000DCD21010AC9F5C50EFF063010FEDBFCFC
:10134000CB6F280ACBAFD3FC0CD20EFC1F1C9CEBF97
:1013500018F4063FCD0C01FE4C2808FE5528080B57
:1013600000181F0E80181B0EC0181706F3CD0C01B5
:10137000FE55280CFE428040E0018060E08180214
:101380000B0CCDF800A0B14FCDFC00C906CFCD0C9E
:1013900001FE4C2808FE5528080E0018E5E2018FE
:1013A000E10E3018DDCDE300F803C84FCBDC001890
:1013B000F4FFFFFFFFFFF0000000000000000000
:1013C000FFFFFFFFFFF00000000000000000000
:1013D000CCBBFAA34127856000C00000E0DC0BBB
:1013E0000F0AC600213412FBC3896700000000000
:1013F0000000FCC30000C30000C348001001022025

```

5. Anhang

5.2.2. Programm Debugger Assemblerausdruck

```
00002
00003          TITL 'Y23VO  DEBUGGER'
00004
00005 CR:      EQU 00DH                ;WAGEN-RUECKLAUF
00006 LF:      EQU 00AH                ;ZEILENVORSCHUB
00007 ETX:     EQU 003H                ;CONTROL/C (END OF TEXT)
00008 BOM:     EQU 00F00H              ;SYSTEMRAM-BEGINN
00009 TLOC:    EQU 00FEBH
00010 LLOC:    EQU 00FE5H
00011 PLOC:   EQU 00FEAH
00012 EXI1:    EQU 00FE4H
00013
00014 OMSK:     EQU 0CFH
00015 IMSK:     EQU 0F3H
00016
00017
00018 ;RUECKSPRUNG IN DEN MONITOR
00019 ;RETEN DER REGISTERINHALTE DES ANWENDERPROGRAMMS
00020 RESTR:    EQU #
0400  F3     00021 DI
0401  B5     00022 PUSH HL
0402  D5     00023 PUSH DE
0403  C5     00024 PUSH BC
0404  F5     00025 PUSH AF
0405  DDE5   00026 PUSH IX
0407  FDE5   00027 PUSH IY
0409  ED57   00028 LD  A,I
040B  F5     00029 PUSH AF
040C  08     00030 EXAF
040D  D9     00031 EXX
040E  E5     00032 PUSH HL
040F  D5     00033 PUSH DE
0410  C5     00034 PUSH BC
0411  F5     00035 PUSH AF
0412  D9     00036 EXX
0413  08     00037 EXAF
00038 ;REGISTER AUF TAB.IM SYST.-RAM UMSPEICHERN
0414  21800  00039 LD  HL,24
0417  39     00040 ADD HL,SP
0418  060B   00041 LD  B,11
041A  EB     00042 EX  DE,HL
041B  21E40F 00043 LD  HL,EXI1
041E  2B     00044 RST0: DEC HL
041F  72     00045 LD  (HL),D
0420  2B     00046 DEC HL
0421  73     00047 LD  (HL),E
0422  D1     00048 POP DE
0423  10F9   00049 DJNZ RST0-#
00050 ;TEST OB HALTEPUNKT ERREICHT IST
0425  C1     00051 POP BC
0426  0B     00052 DEC BC
0427  F9     00053 LD  SP,HL
0428  21EBOF  00054 LD  HL,TLOC
042B  7E     00055 LD  A,(HL)
042C  91     00056 SUB C
042D  23     00057 INC HL
042E  2004   00058 JRNZ RSTA-#
0430  7E     00059 LD  A,(HL)
0431  98     00060 SBC B
0432  280C   00061 JRZ RST1-# ;HALTEPUNKT 1 ERREICHT
00062 ;TEST, OB HALTEPUNKT 2 ERREICHT WURDE
0434  23     00063 RSTA: INC HL
0435  23     00064 INC HL
0436  7E     00065 LD  A,(HL)
0437  91     00066 SUB C
0438  23     00067 INC HL
0439  2004   00068 JRNZ RSTB-#
043B  7E     00069 LD  A,(HL)
043C  98     00070 SBC B
043D  2801   00071 JRZ RST1-# ;HALTEPUNKT 2 ERREICHT
00072 ;KIN HALTEPUNKT ERREICHT
043F  03     00073 RSTB: INC BC
00074 ;RETEN VON HL
0440  21E50F  00075 RST1: LD  HL,LLOC
0443  73     00076 LD  (HL),E
0444  23     00077 INC HL
0445  72     00078 LD  (HL),D
```

```

00079 ; RETTEN VON PC
0446 2EE9 00080 LD L,L(PLOC-1)
0448 71 00081 LD (HL),C
0449 23 00082 INC HL
044A 70 00083 LD (HL),B
00084 ;AUSGABE DER STOP-ADRESSE
044B C5 00085 PUSH BC
044C CD2101 00086 CALL IO.COMC
044F 00087 DB '$'
0450 E1 00088 POP HL
0451 CDB004 00089 CALL LADR
00090 ;RESTAURIEREN DES SPEICHERINHALTES AN DEN HALTEPUNKTEN
00091 ;UND LOESCHEN DER TABELLE FUER DIE HALTEPUNKTE
0454 21EB0F 00092 LD HL,TLOC
0457 1602 00093 LD D,2
0459 4E 00094 RST2: LD C,(HL)
045A AF 00095 XOR A
045B 77 00096 LD (HL),A
045C 23 00097 INC HL
045D 46 00098 LD B,(HL)
045E 77 00099 LD (HL),A
045F 23 00100 INC HL
0460 79 00101 LD A,C
0461 B0 00102 OR B
0462 2802 00103 JRZ RST3-#
0464 7E 00104 LD A,(HL)
0465 02 00105 LD (BC),A
0466 23 00106 RST3: INC HL
0467 15 00107 DEC D
0468 20EF 00108 JRNZ RST2-#
046A C34800 00109 JMP IO.START
00110 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00111 ;RUECKSPRUNGRoutine ZUM STARTEN EINES ANWENDERPROGRAMMS
046D D1 00112 EXIT: POP DE
046E C1 00113 POP BC
046F F1 00114 POP AF
0470 DDE1 00115 POP IX
0472 FDE1 00116 POP IY
0474 08 00117 EXAF
0475 D9 00118 EXX
0476 F1 00119 POP AF
0477 BD47 00120 LD I,A
0479 E1 00121 POP HL
047A D1 00122 POP DE
047B C1 00123 POP BC
047C F1 00124 POP AF
047D D9 00125 EXX
047E 08 00126 EXAF
047F E1 00127 POP HL
0480 F9 00128 LD SP,HL
0481 C3E40F 00129 JMP EX11
00130 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00131 ;SUBR. ZUM INKREMENTIEREN VON HL UND VERGLEICHEN MIT DE
00132 ;**IN: HL...1. OPERAND
00133 ; DE...2. OPERAND
00134 ;**OUT: HL=HL+1
00135 ; CARRY=0: HL=DE
00136 ; CARRY=1: HL=DE,HL=0
00137 ;*KILL: A,F,H,L
0484 23 00138 HILO: INC HL
0485 7C 00139 LD A,H
0486 B5 00140 OR L
0487 37 00141 SCF
0488 C8 00142 RZ
0489 7B 00143 LD A,E
048A 95 00144 SUB L
048B 7A 00145 LD A,D
048C 9C 00146 SBC H
048D C9 00147 RET
00148 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00149 ;SUBR. ZUR AUSGABE EINES ZEICHENS UEBER DAS GERAET "0"
00150 ;**IN: ZEICHEN AUF SPEICHERPLATZ HINTER CALL
00151 ;**OUT: C...ZEICHEN
00152 ; (SP)=(SP)+1
00153 ;*KILL: A,F,C
048E E3 00154 POC: EX (SP),HL
048F 4E 00155 LD C,(HL)
0490 23 00156 INC HL
0491 E3 00157 EX (SP),HL
0492 C3E900 00158 JMP IO.MWO
00159 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00160 ;SUBR. ZUR KONVERTIERUNG EINES HALBBYTES IN EIN HEXADEZ. ASCII-ZEICHEN

```

5. Anhang

```

00161 ;***IN: A...HALBBYTE
00162 ;**OUT: A,C...HEXADEZ. ASCII-ZEICHEN
00163 ;*KILL: A,F,C
0495 B60F 00164 CONV: AND OFH
0497 C690 00165 ADD 90H
0499 27 00166 DAA
049A CE40 00167 ADC 40H
049C 27 00168 DAA
049D 4F 00169 LD C,A
049E C9 00170 RET
00171 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00172 ;SUBR. ZUR AUSGABE EINES REGISTERINHALTES UEBER DAS GERAET "L"
00173 ;***IN: HL...TABELLENPLATZ DES REGISTERS
00174 ;**OUT: HL=HL+3
00175 ;*KILL: A,F,B,C,D,E,H,L
049F 23 00176 LREG: INC HL
04A0 5E 00177 LD E,(HL)
04A1 160F 00178 LD D,H(BOM) BOM/256
04A3 23 00179 INC HL
04A4 46 00180 LD B,(HL)
04A5 23 00181 INC HL
04A6 1A 00182 LD A,(DE)
04A7 CDB504 00183 CALL LBYTE
04AA 05 00184 DEC B
04AB F8 00185 RM
04AC 1B 00186 DEC DE
04AD 1A 00187 LD A,(DE)
04AE 1805 00188 JR LBYTE-#
00189 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00190 ;SUBR. ZUR AUSGABE VON 16 BIT BIN. INF. ALS ASCII-ZEICHEN
00191 ;UEBER DAS GERAET "L"
00192 ;***IN: HL...BIN. INF.
00193 ;**OUT: -
00194 ;*KILL: A,F,C
04B0 7C 00195 LADR: LD A,H
04B1 CDB504 00196 CALL LBYTE
04B4 7D 00197 LD A,L
00198 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00199 ;SUBR. ZUR AUSGABE VON 8 BIT BIN. INF. ALS ASCII-ZEICHEN
00200 ;UEBER DAS GERAET "L"
00201 ;***IN: A...BIN. INF.
00202 ;**OUT: -
00203 ;*KILL: A,F,C
04B5 F5 00204 LBYTE: PUSH AF
04B6 0F 00205 RRCA
04B7 0F 00206 RRCA
04B8 0F 00207 RRCA
04B9 0F 00208 RRCA
04BA CD9504 00209 CALL CONV
04BD CDEC00 00210 CALL IO.MLO
04C0 F1 00211 POP AF
04C1 CD9504 00212 CALL CONV
04C4 C3EC00 00213 JMP IO.MLO
00214 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00215 ;SUBR. ZUR EINGABE VON HEXADEZ. PARAMETERN UEBER DAS TERMINAL
00216 ;***IN: C...ZAHL DER PARAMETER
00217 ;**OUT: PARAMETER IM STACK
00218 ; SP=SP-2*C
00219 ;*KILL: A,F,B,C,H,L,SP
04C7 CDD704 00220 EXPR: CALL PARAM
04CA E3 00221 EX (SP),HL
04CB E5 00222 PUSH HL
04CC 0D 00223 DEC C
04CD 3003 00224 JRNC EXO-#
04CF 2003 00225 JRNZ ERR1-#
04D1 C9 00226 RET
04D2 20F3 00227 EXO: JRNZ EXPR-#
04D4 C34100 00228 ERR1: JMP IO.ERROR
00229 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00230 ;SUBR. ZUR EINGABE VON 24 BIT BIN. INF. ALS HEXADEZ. ASCII-ZEICHEN
00231 ;UEBER DAS TERMINAL
00232 ;***IN: -
00233 ;**OUT: A,H,L...BIN. INF.
00234 ; B...TRENNZEICHEN
00235 ;*KILL: A,F,B,H,L
04D7 CD1805 00236 PARAM: CALL PCHK
04DA 28F8 00237 JRZ ERR1-#
04DC C5 00238 PAO: PUSH BC
04DD 210000 00239 LD HL,0
04E0 4C 00240 LD C,H
04E1 47 00241 PA1: LD B,A
04E2 CD0605 00242 CALL NIBLE

```

```

04E5 3813      00243      JRC PA2-#
04E7 47        00244      LD B,A
04E8 79        00245      LD A,C
04E9 29        00246      ADD HL,HL
04EA 17        00247      RLA
04EB 29        00248      ADD HL,HL
04EC 17        00249      RLA
04ED 29        00250      ADD HL,HL
04EE 17        00251      RLA
04EF 29        00252      ADD HL,HL
04F0 17        00253      RLA
04F1 4F        00254      LD C,A
04F2 78        00255      LD A,B
04F3 B5        00256      OR L
04F4 6F        00257      LD L,A
04F5 CD0C01    00258      CALL IO.TI
04F8 18E7      00259      JR PA1-#
04FA 78        00260      LD A,B
04FB CD1B05    00261      CALL P2C
04FE 20D4      00262      JRNZ ERR1-#
0500 E3        00263      EX (SP),HL
0501 60        00264      LD H,B
0502 E3        00265      EX (SP),HL
0503 79        00266      LD A,C
0504 C1        00267      POP BC
0505 C9        00268      RET
0506 D630      00269      ;*-----*
0508 D8        00270      ;SUBR. ZUR KONVERTIERUNG VON 1 HEXAD. ASCII-Z. IN 1 HALBBYTE BIN. INF.
0509 C6E9      00271      ;**IN: A...HEXADEZ. ASCII-Z.
050B D8        00272      ;**OUT: A...HALBBYTE
050C C606      00273      ; CARRY=1: FALSCHES ASCII-Z.
050E F21405    00274      ;*KILL: A,F
0511 C607      00275      NIBBLE: SUB '0'
0513 D8        00276      RC
0514 C60A      00277      ADD 0B9H
0516 B7        00278      RC
0517 C9        00279      ADD 6
0518 CD0C01    00280      JPP N10
0519 C607      00281      ADD 7
051A D8        00282      RC
051B C60A      00283      N10: ADD 10
051C B7        00284      OR A
051D C9        00285      RET
051E D8        00286      ;*-----*
051F C60A      00287      ;SUBR. ZUR EINGABE EINES ZEICHENS UEBER DAS TERMINAL
0520 C60A      00288      ;UND TEST AUF ERLAUBTE TRENNZEICHEN
0521 C60A      00289      ;**IN: -
0522 C60A      00290      ;**OUT: A...EINGEGEBENES ZEICHEN
0523 C60A      00291      ; CARRY=1: EINGEG. Z. IST CR
0524 C60A      00292      ; ZERO=1: EINGEG. Z. IST LEEER., KOMMA ODER CR
0525 C60A      00293      ;*KILL: A,F,C
0526 C9        00294      PCHK: CALL IO.TI
0527 C5        00295      ;*-----*
0528 CDE300    00296      ;SUBR. ZUM TEST AUF ERLAUBTE TRENNZEICHEN
0529 CDE300    00297      ;**IN: A...ZEICHEN
052A CDE300    00298      ;**OUT: -
052B CDE300    00299      ; CARRY=1: EINGEG. Z. IST CR
052C CDE300    00300      ; ZERO=1: EINGEG. Z. IST LEEER., KOMMA ODER CR
052D CDE300    00301      ;*KILL: F
052E CDE300    00302      P2C: CMP
052F CDE300    00303      RZ
0530 CDE300    00304      CMP ','
0531 CDE300    00305      RZ
0532 CDE300    00306      CMP CR
0533 CDE300    00307      SCF
0534 CDE300    00308      RZ
0535 CDE300    00309      CCF
0536 CDE300    00310      RET
0537 CDE300    00311      ;*-----*
0538 CDE300    00312      ;SUBR. ZUR EINGABE EINES BIN. BYTES UEBER DAS LOG. GERAET "I"
0539 CDE300    00313      ;UND ADDIEREN DES BYTES ZUM INHALT VON D
053A CDE300    00314      ;**IN: -
053B CDE300    00315      ;**OUT: A...EINGEGEBENES BYTE
053C CDE300    00316      ; D=D+A
053D CDE300    00317      ; FLAGS...ENTSPR. DEM ERGEBNIS DER ADD.
053E CDE300    00318      ;*KILL: A,F,D
053F CDE300    00319      IBYTE: PUSH BC
0540 CDE300    00320      CALL IO.MRI
0541 CDE300    00321      CALL NIBBLE
0542 CDE300    00322      RLCA
0543 CDE300    00323      RLCA
0544 CDE300    00324      RLCA

```

5. Anhang

```

0531 07      00325      RLCA
0532 4F      00326      LD      C,A
0533 CDE300  00327      CALL IO.MRI
0536 CD0605  00328      CALL NIBL
0539 B1      00329      OR      C
053A 4F      00330      LD      C,A
053B 82      00331      ADD     D
053C 57      00332      LD      D,A
053D 79      00333      LD      A,C
053E C1      00334      POP     BC
053F C9      00335      RET

00336 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00337 ;SUBR. ZUR AUSGABE VON 16 BIT BIN. INF. ALS ASCII-Z. UEBER GERAET "O"
00338 ;**IN: HL...BIN. INF.
00339 ;**OUT: -
00340 ;*KILL: A,F,C
0540 7C      00341      OADR: LD      A,H
0541 CD4505  00342      CALL OBYTE
0544 7D      00343      LD      A,L
00344 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00345 ;SUBR. ZUR AUSGABE VON 8 BIT BIN. INF. ALS ASCII-Z. UEBER GERAET "O"
00346 ;**IN: A...BIN. INF.
00347 ;**OUT: -
00348 ;*KILL: A,F,C
0545 5F      00349      OBYTE: LD     E,A
0546 0F      00350      RRCA
0547 0F      00351      RRCA
0548 0F      00352      RRCA
0549 0F      00353      RRCA
054A CD9504  00354      CALL CONV
054D CDE900  00355      CALL IO.MWO
0550 7B      00356      LD      A,E
0551 CD9504  00357      CALL CONV
0554 CDE900  00358      CALL IO.MWO
0557 7B      00359      LD      A,E
0558 82      00360      ADD     D
0559 57      00361      LD      D,A
055A C9      00362      RET

00363 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00364 ;SUBR. ZUR AUSGABE VON CR UND LF AUF DEM GERAET "O"
00365 ;**IN: -
00366 ;**OUT: -
00367 ;*KILL: A,F,C
055B CD8E04  00368      OCRLF: CALL POC
055E 0D      00369      DB      CR
055F CD8E04  00370      CALL POC
0562 0A      00371      DB      LF
0563 C9      00372      RET
00373 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00374 ;DISPLAY-KOMMANDO
00375 DISP: EQU #
0564 CDC704  00376      CALL EXPR
0567 D1      00377      POP     DE
0568 E1      00378      POP     HL
00379 ;AUSGABE DER ADRESSE
00380 DIO: CALL IO.LCRLF
0569 CDFB02  00381      CALL LADR
056C CDB004  00382      ;AUSGABE EINES LEERZEICHENS
00383 DI1: LD      C,' '
056F 0E20    00384      CALL IO.LO
0571 CD0203  00385      ;AUSGABE EINES BYTES
00386 LD      A,(HL)
0574 7E      00387      CALL LBYTE
0575 CDB504  00388      ;KOMMANDO-ENDE ?
00389 CALL HIO
0578 CD8404  00390      JPC     IO.LCRLF
057B DAFB02  00391      ;ZEILENENDE ?
00392 LD      A,L
057E 7D      00393      AND     OFH
057F E60F    00394      JRNZ   DI1-#
0581 20EC    00395      JR      DIO-#
0583 18E4    00396      ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00397 ;END-FILE KOMMANDO
00398 EOF: EQU #
00399 DEC     C
0585 0D      00400      CALL EXPR
0586 CDC704  00401      ;AUSGABE VON ": "
00402 CALL POC
0589 CD8E04  00403      DB      ':'
058C 00404      ;AUSGABE DER REKORDLAENGE 0
00405 XOR     A
058D AF      00406      LD      D,A

```

```

058F CD4505 00407 CALL OBYTE
00408 ;AUSGABE DER STARTADRESSE
0592 E1 00409 POP HL
0593 CD4005 00410 CALL OADR
00411 ;AUSGABE DES DATENTYPS
0596 3E01 00412 LD A,1
0598 CD4505 00413 CALL OBYTE
00414 ;BERECHNUNG UND AUSGABE DER CHECKSUMME
059B AF 00415 XOR A
059C 92 00416 SUB D
059D CD4505 00417 CALL OBYTE
00418 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00419 ;SUBR. ZUR AUSGABE VON 60 LEEBSTELLEN AUF DEM GERAET "0"
00420 ;**IN: -
00421 ;**OUT: -
00422 ;*KILL: A,F,B,C
05A0 063C 00423 LEAD: LD B,60
05A2 CD8E04 00424 LEO: CALL POC
05A5 00 00425 DB 0
05A6 10FA 00426 DJNZ LEO-#
05A8 C9 00427 RET
00428 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00429 ;FILL-KOMMANDO
00430 FILL: EQU #
00431 INC C
05A9 .OC 00432 CALL EXPR
05AA CDC704 00433 POP BC ;KONSTANTE IN C
05AD C1 00434 POP DE ;ENDADRESSE IN DE
05AE D1 00435 POP HL ;ANFANGSADRESSE IN HL
05AF E1 00436 FIO: LD (HL),C
05B0 71 00437 CALL HILO
05B1 CD8404 00438 JRNC FIO-#
05B4 30FA 00439 RET
05B6 C9 00440 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00441 ;GO-KOMMANDO
00442 GOTO: EQU #
00443 ;VORBEREITEN DES SPRUNGES IN DAS ANWENDERPROGRAMM
00444 LD HL,EXIT
00445 EX (SP),HL
00446 ;STARTADRESSE ?
05B7 216D04 00447 CALL PCHK
05BA E3 00448 JRZ GO0-# ;KEINE STARTADRESSE ANGEZEIGT
00449 ;EINGABE DER STARTADRESSE
05C0 CDC0C4 00450 CALL PAO
05C3 EB 00451 EX DE,HL
05C4 21EA0F 00452 LD HL,FLOC
05C7 72 00453 LD (HL),D
05C8 2B 00454 DEC HL
05C9 73 00455 LD (HL),E
05CA 3B25 00456 GO0: JRC GO4-# ;KEINE HALTEPUNKTE ANGEZEIGT
05CC 110200 00457 LD DE,2
00458 ;EINGABE DES HALTEPUNKTES
05CF CD2101 00459 GO1: CALL IO.COMC
05D2 00460 DB '-'
05D3 CDD704 00461 CALL PARAM
05D6 E5 00462 PUSH HL
05D7 14 00463 INC D
05D8 3806 00464 JRC GO2-# ;NUR EIN HALTEPUNKT
05DA 1D 00465 DEC E
05DB 20F2 00466 JRNZ GO1-#
05DD D24100 00467 JPNC IO.ERROR
05E0 21B0F 00468 GO2: LD HL,TLOC
00469 ;ADRESSE DER HALTEPUNKTE IN TABELLE
00470 GO3: POP BC
05E3 C1 00471 LD (HL),C
05E4 71 00472 INC HL
05E5 23 00473 LD (HL),B
05E6 70 00474 INC HL
05E7 23 00475 ;RETEN DES SPEICHERINHALTES AM HALTEPUNKT
00476 LD A,(BC)
05E8 0A 00477 LD (HL),A
05E9 77 00478 INC HL
05EA 23 00479 ;SETZEN DES HALTEPUNKTES
00480 LD A,OPPH ;RST 56
05EB 3E0F 00481 LD (BC),A
05ED 02 00482 DEC D
05EE 15 00483 DJNZ GO3-#
05EF 20F2 00484 GO4: JMP IO.CRLF ;SPRUNG IN ANWENDERPROGRAMM
05F1 C32D03 00485 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00486 ;ADDITION UND SUBTRAKTION VON HEXADEZIMALZAHLEN
00487 HEXN: EQU #
00488 CALL EXPR
05F4 CDC704

```

5. Anhang

05F7	CD2D03	00489	CALL IO.CRLF	
05FA	D1	00490	POP DE	;2. OPERAND IN DE
05FB	E1	00491	POP HL	;1. OPERAND IN HL
05FC	E5	00492	PUSH HL	
05FD	19	00493	ADD HL,DE	;ADDITION
05FE	CDB004	00494	CALL LADR	;AUSGABE DER SUMME
0601	CDFD01	00495	CALL IO.BLK	
0604	E1	00496	POP HL	
0605	B7	00497	OR A	
0606	ED52	00498	SBC HL,DE	;SUBTRAKTION
0608	C3B004	00499	JMP LADR	;AUSGABE DER DIFFERENZ
		00500	;-----*	
		00501	;MOVE-KOMMANDO	
		00502	MOVE: EQU #	
		00503	INC C	
060B	0C	00504	CALL EXPR	
060C	CDC704	00505	POP BC	;ADRESSE DES ZIELGEBIETES IN BC
060F	C1	00505	POP DE	;ENDADRESSE DES QUELLGEB. IN DE
0610	D1	00506	POP HL	;ANFANGSADRESSE DES QUELLGEB. IN HL
0611	E1	00507	POP HL	
0612	7E	00508	MVO: LD A,(HL)	
0613	02	00509	LD (BC),A	
0614	03	00510	INC BC	
0615	CD6D00	00511	CALL IO.ZS	;ERHOLZEIT PROG.->MONO
0618	CD8404	00512	CALL HILO	
061B	30F5	00513	JRNC MVO-#	
061D	C9	00514	RET	
		00515	;-----*	
		00516	;READ-KOMMANDO	
		00517	READ: EQU #	
		00518	DEC C	
061E	0D	00519	CALL EXPR	;EINGABE DES OFFSET
061F	CDC704	00520	CALL IO.CRLF	
0622	CD2D03	00521	;WARTEN AUF EINGABE VON ":"	
0625	CDE300	00522	REDO: CALL IO.MRI	
0628	FE3A	00523	CMF ':'	
062A	20F9	00524	JRNZ REDO-#	
062C	AF	00525	XOR A	
062D	57	00526	LD D,A	
062E	CD2705	00527	CALL IBYTE	;EINGABE DER REKORDLÄNGE
0631	282F	00528	JRZ RED3-#	;REKORDLÄNGE = 0
0633	5F	00529	LD E,A	
		00530	;EINGABE DER REKORDADRESSE	
0634	CD2705	00531	CALL IBYTE	
0637	67	00532	LD H,A	
0638	CD2705	00533	CALL IBYTE	
063B	6F	00534	LD L,A	
		00535	;EINGABE DES DATENTYPS	
063C	CD2705	00536	CALL IBYTE	
063F	4B	00537	LD C,E	
		00538	;EINGABE DER REKORDINFORMATION	
0640	E5	00539	PUSH HL	
0641	2100FF	00540	LD HL,QFF00H	
0644	39	00541	ADD HL,SP	
0645	CD2705	00542	RED1: CALL IBYTE	
0648	77	00543	LD (HL),A	
0649	23	00544	INC HL	
064A	1D	00545	DEC E	
064B	20F8	00546	JRNZ RED1-#	
064D	CD2705	00547	CALL IBYTE	;EINGABE DER CHECKSUMME
0650	C24100	00548	JPNZ IO.ERROR	;FALSCHER CHECKSUMME
		00549	;BERECHNUNG DER LADEADRESSE	
0653	D1	00550	POP DE	
0654	E3	00551	EX (SP),HL	
0655	EB	00552	EX DE,HL	
0656	19	00553	ADD HL,DE	
0657	0600	00554	LD B,0	
0659	09	00555	ADD HL,BC	
065A	EB	00556	EX DE,HL	
065B	E3	00557	EX (SP),HL	
		00558	;LADEN DES GELESSENEN REKORDS	
065C	2B	00559	RED2: DEC HL	
065D	1B	00560	DEC DE	
065E	EDB8	00561	LDDR	
0660	18C3	00562	JR REDO-#	
		00563	;LADEN DER STARTADRESSE	
0662	21EA0F	00564	RED3: LD HL,PLOC	
0665	CD2705	00565	CALL IBYTE	
0668	77	00566	LD (HL),A	
0669	2B	00567	DEC HL	
066A	CD2705	00568	CALL IBYTE	
066D	77	00569	LD (HL),A	
066E	E1	00570	POP HL	
066F	C9	00571	RET	


```

00572 ;*-----*-----*-----*-----*-----*-----*-----*-----*
00573 ;SUBSTITUTE-KOMMANDO
00574 SUBS: EQU #
0670 CDD704 00575 CALL PARAM ;EINGABE DER 1. ADRESSE (HL)
0673 DB 00576 RC
00577 ;ANZEIGE DES BYTES
0674 7E 00578 SUO: LD A,(HL)
0675 CDB504 00579 CALL LBYTE
0678 CD2101 00580 CALL IO.COMC
067B 00581 DB '-'
067C CD1805 00582 CALL PCHK
067F D8 00583 RC ;ENDE DES KOMMANDOS (CR)
0680 2807 00584 JRZ SU1-# ;KEINE AENDERUNG DES INHALTES (LEERZ.)
00585 ;EINGABE DES NEUEN SPEICHERINHALTES
0682 EB 00586 EX DE,HL
0683 CDDC04 00587 CALL PA0
0686 EB 00588 EX DE,HL
0687 73 00589 LD (HL),E
0688 D8 00590 RC ;ENDE DES KOMMANDOS (CR)
0689 23 00591 SU1: INC HL
068A 18E8 00592 JR SU0-#
00593 ;*-----*-----*-----*-----*-----*-----*-----*-----*
00594 ;SPEICHERTEST
00595 KNOW: EQU #
068C CDC704 00596 CALL EXPR
068F 0600 00597 LD B,0
0691 D1 00598 POP DE ;ENDEADR
0692 E1 00599 ST1: POP HL ;ANF-ADR
0693 E5 00600 PUSH HL
0694 7D 00601 ST2: LD A,L ;FILL
0695 AC 00602 XOR H
0696 A8 00603 XOR B
0697 77 00604 LD (HL),A
0698 CD8404 00605 CALL HILO
069B 30F7 00606 JRNC ST2-#
069D E1 00607 POP HL
069E 7D 00608 ST3: LD A,L
069F AC 00609 XOR H
06A0 A8 00610 XOR B
06A1 BE 00611 CMP (HL)
06A2 2057 00612 JRNZ V1-#
06A4 CD8404 00613 CALL HILO
06A7 30F5 00614 JRNC ST3-#
06A9 C9 00615 RET
00616 ;*-----*-----*-----*-----*-----*-----*-----*-----*
00617 ;WRITE-KOMMANDO
00618 WRITE: EQU #
06AA CDC704 00619 CALL EXPR
06AD CD2D03 00620 CALL IO.CRLF
06B0 D1 00621 POP DE ;ENDADRESSE IN DE
06B1 E1 00622 POP HL ;ANFANGSADRESSE IN HL
00623 ;AUSGABE VON ":"
06B2 CD8B04 00624 WRO: CALL POC
06B5 00625 DB ':'
00626 ;BESTIMMEN DER REKORDLAENGE
06B6 011000 00627 LD BC,16
06B9 E5 00628 PUSH HL
06BA 04 00629 WR1: INC B
06BB 0D 00630 DEC C
06BC 2805 00631 JRZ WR2-#
06BE CD8404 00632 CALL HILO
06C1 30F7 00633 JRNC WR1-#
06C3 E1 00634 WR2: POP HL
06C4 D5 00635 PUSH DE
06C5 1600 00636 LD D,0
06C7 78 00637 LD A,B
06C8 CD4505 00638 CALL OBYTE ;AUSGABE DER REKORDLAENGE
06CB CD4005 00639 CALL OADR ;AUSGABE DER REKORDADRESSE
00640 ;AUSGABE DES DATENTYPS
06CE AF 00641 XOR A
06CF CD4505 00642 CALL OBYTE
00643 ;AUSGABE DER REKORDINFORMATION
06D2 7E 00644 WR3: LD A,(HL)
06D3 CD4505 00645 CALL OBYTE
06D6 23 00646 INC HL
06D7 10F9 00647 DJNZ WR3-#
00648 ;BERECHNUNG UND AUSGABE DER CHECKSUMME
06D9 AF 00649 XOR A
06DA 92 00650 SUB D
06DB CD4505 00651 CALL OBYTE
06DE D1 00652 POP DE
06DF 2B 00653 DEC HL

```

5. Anhang

```

00654 ;AUSGABE VON CR,LF
00655 CALL OCRLF
00656 CALL HILO
00657 JRNC WRO-#
00658 RET
00659 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00660 ;VERGLEICH VON SPEICHERINHALTEN
00661 VER: EQU #
00662 INC C
00663 CALL EXPR
00664 POP BC ;ZIEL
00665 POP DE ;ENDE
00666 POP HL ;START
00667 V2: LD A,(BC)
00668 CMP (HL)
00669 JRNZ V1-# ;BEI FEHLER
00670 INC BC
00671 CALL HILO
00672 JRNC V2-#
00673 RET
00674 V1: CALL IO.CRLF
00675 CALL LADR
00676 RET
00677 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
00678 ;X-KOMMANDO
00679 X: EQU #
00680 LD HL,ACTBL
00681 CALL PCHK
00682 JRC X5-# ;AUSGABE ALLER REGISTER
00683 ;BESTIMMUNG DER AUSZUGEBENDEN REGISTER
00684 LD C,NREGS
00685 X0: CMP (HL)
00686 JRZ X1-#
00687 INC HL
00688 INC HL
00689 INC HL
00690 DEC C
00691 JRNZ X0-#
00692 JMP IO.ERROR
00693 X1: CALL IO.BLK
00694 ;AUSGABE DES AKTUELLEN REGISTERINHALTES
00695 X2: CALL LRBG
00696 CALL IO.COMC
00697 DB '-'
00698 CALL PCHK
00699 RC ;ENDE DES KOMMANDOS (CR)
00700 JRZ X4-# ;ANZEIGE DES NAECHSTEN REG. (LEERZ.)
00701 PUSH HL
00702 PUSH BC
00703 ;EINGABE EINES NEUEN REGISTERINHALTES
00704 CALL PAO
00705 LD A,L
00706 LD (DE),A
00707 POP AF
00708 OR A
00709 JPM X3 ; 8 BIT-REGISTER
00710 ;16 BIT-REGISTER
00711 INC DE
00712 LD A,H
00713 LD (DE),A
00714 X3: POP HL
00715 X4: XOR A
00716 OR (HL)
00717 RM ;ALLE REGISTER ANGEZEIGT
00718 LD A,B
00719 CMP CR
00720 RZ ;ENDE DES KOMMANDOS
00721 JR X2-#
00722 ;ANZEIGE ALLER REGISTER
00723 X5: CALL IO.CRLF
00724 X6: CALL IO.BLK
00725 XOR A
00726 OR (HL)
00727 RM ;TABELLENENDE MINUS-FF
00728 LD C,(HL)
00729 CALL IO.MCO ;AUSGABE DES REGISTER-NAMENS
00730 CALL IO.COMC
00731 DB '-'
00732 CALL LRBG ;AUSGABE DES REGISTER-INHALTES
00733 JR X6-#
00734 ;TABELLE DER REGISTER-ADRESSEN
00735 ACTBL: DB 'A'
00736
00737
00738
00739
00740
00741
00742
00743
00744
00745
00746
00747
00748
00749
00750
00751
00752
00753
00754
00755
00756
00757

```

```

0758 D3      00736      DB L(10.ALOC)
0759 00      00737      DB 0
075A      00738      DB 'B'
075B D4      00739      DB L(10.BLOC)
075C 00      00740      DB 0
075D      00741      DB 'C'
075E D0      00742      DB L(10.CLOC)
075F 00      00743      DB 0
0760      00744      DB 'D'
0761 CF      00745      DB L(10.DLOC)
0762 00      00746      DB 0
0763      00747      DB 'E'
0764 CE      00748      DB L(10.ELOC)
0765 00      00749      DB 0
0766      00750      DB 'F'
0767 D2      00751      DB L(10.FLOC)
0768 00      00752      DB 0
0769      00753      DB 'H'
076A E6      00754      DB L(10.HLOC)
076B 00      00755      DB 0
076C      00756      DB 'L'
076D E5      00757      DB L(10.LLOC)
076E 00      00758      DB 0
076F      00759      DB 'M'
0770 E6      00760      DB L(10.HLOC)
0771 01      00761      DB 1
0772      00762      DB 'P'
0773 EA      00763      DB L(10.PLOC)
0774 01      00764      DB 1
0775      00765      DB 'S'
0776 E3      00766      DB L(10.SLOC)
0777 01      00767      DB 1
0778 FF      00768      DB OFFH
0779      00769      NREGS: EQU 11 ;(#-ACTBL)/3
077A      00770      ;*-----*-----*-----*-----*-----*-----*
077B      00771      ;Z-KOMMANDO
077C      00772      ZR: EQU #
077D      00773      LD HL,ZACTL
077E      00774      CALL PCHK
077F      00775      JRC I5-# ;ANZEIGE ALLER REGISTER
0780      00776      LD C,ZREGS
0781      00777      JMP IO
0782      00778      ;TABELLE DER REGISTERADRESSEN
0783      00779      ZACTL: DB 'A'
0784      00780      DB L(10.ASLC)
0785      00781      DB 0
0786      00782      DB 'B'
0787      00783      DB L(10.BSLC)
0788      00784      DB 0
0789      00785      DB 'C'
078A      00786      DB L(10.CSLC)
078B      00787      DB 0
078C      00788      DB 'D'
078D      00789      DB L(10.DSLC)
078E      00790      DB 0
078F      00791      DB 'E'
0790      00792      DB L(10.ESLC)
0791      00793      DB 0
0792      00794      DB 'F'
0793      00795      DB L(10.FSLC)
0794      00796      DB 0
0795      00797      DB 'H'
0796      00798      DB L(10.HSLC)
0797      00799      DB 0
0798      00800      DB 'L'
0799      00801      DB L(10.LSLC)
079A      00802      DB 0
079B      00803      DB 'L'
079C      00804      DB L(10.LLOC)
079D      00805      DB 0
079E      00806      DB 'X'
079F      00807      DB L(10.XLOC+1)
07A0      00808      DB 1
07A1      00809      DB 'Y'
07A2      00810      DB L(10.YLOC+1)
07A3      00811      DB 1
07A4      00812      DB OFFH
07A5      00813      ZREGS: EQU 11 ;(#-ZACTL)/3
07A6      00814      END
07A7
07A8

```

5. Anhang

EBAS-88		CROSS-REFERENCE							PN: IO
ROM	0000	0000	0000	0000	0000	0000	0000	0000	0000
		0000	0000	0000	0000	002B	0064	03B1	
RAM	0C00								
RESTR	0400	0038							
BRAM	FC00	0265	0269	026C	0280	0283	03F1		
EOM	0F00								
STACK	0FCE	0000	0041						
RSLOC	0FE3	001B							
CURS	0FF1	0208		022C	028D				
FLGIO	0FFE	013F		01A5					
IOBYT	0FFF	00F8		00FD	0294	02B6	02F5	030A	
RAMZK	0FFC	006F							
PIOAD	00FC	016B	02A0	02AC	02CF	031C	0322	033E	
		0346							
PIOBD	00FD	0177	0324						
PIOAC	00FE	000F							
PIOBC	00FF								
IMSK	00F3	036B							
OMSK	00CF	038C							
LMSK	003F	0352							
IT	0000	0378							
IL	0008	029C	037C						
IU	000C	0380							
OT	0000	0399							
OL	0020	02EE	039D						
OU	0030	03A1							
LT	0000	035F							
LL	0080	0311	0363						
LU	00C0	0315	0367						
CR	000D	024E	02FB	0330					
LF	000A	0251	0300	0334					
BTX	0003	0110	03A8						
AFST	0027								
ULOCR	0FF9	0066	00B2						
ASS	C400	0097							
BASIC	8000	009A							
PRINT	0800	00C1							
TEXT	C000	00CA							
TRAM	EFA0	00D6							
EOT	00FF	007D	00DB						
BEGIN	0000								
RUFZ	003B	0023							
ERROR	0041	005F	04D4	05DD	0650	0715			
START	00A8	0029	0052	0112	03F9	046A			
ZS	006D	0174	02A9	02AE	02B1	02D1	02D4	0615	
ZS1	0072	0075							
PCGA	007A	005C	0216						
PCGA1	007C	0087							
PCGA2	0089	0082							
PCGA3	0092	007F							
PCGA4	0094	0090							
JPTBL	0096	0059							
MCI	00E0	010D	0299						
MRI	00E3	03A5	0528	0533	0625				
MCO	00E6	0116	0125	02EB	074B				
MWO	00E9	0492	054D	0554					
MLO	00EC	012D	02FD	03AC	04BD	04C4			
MCSTS	00EF								
MIOC	00F2								
MIOS	00F5								
IOCHK	00F8	00F2	0382						
IOSET	00FC	00F5	0388						
TPIOA	0101	0011	0104						
LPIOA	0003	000D							
TPIOB	0104	0107							
LPIOB	0003	0016							
BREAK	0107	0302							
TI	010C	0056	0354	036D	038E	04F5	0518		
CMC	0121	0044	004E	032D	0331	044C	05CF	0678	
		071E	074E						
		0130							
STR	0128	0026							
CI	0132	00E0							
CIO	013C	0132							
CI2	0142	0145	0163	0167					
CI1	014F	0152							
CI3	0161	0157							
CI4	0165	015B							
ZYKL	0169	0142	014F						
ZYK1	0171	0180							
ZYK2	0184	017B							

ZYK3	0186	0188							
CVTA	018B	0148							
CVTA0	018E	0192							
CVTA1	0198	019B							
SHIFT	01B8	01AA							
SH2	01BC	011C	0137						
SH1	01BF	01BA							
CONT	01C2	01B1							
TTAB	01C5	01A0	01FD						
LTAB	0038								
BLK	01FD	0601	0718	0744					
CO	01FF	00B6							
COO	0221	0219							
HTAB	0224	0258							
CO3	0225	023D	0240						
CO1	0228	0237	0244	024C					
CO2	022C								
BS	0236	025B							
WAG	0239	0247	024F						
ZVOR	023F	0252							
CUP	0242	0255							
NL	0246	025E							
BELL	0249	0261							
COTAB	024E	0213							
VOLL	0264	0225							
ROLLA	0268								
ZLOE	0277	0279							
BLOE	027D	0020							
RI	0294	00E3							
LIN	02A0	02A4	02C0						
LIN1	02AC	02B7							
RI8	02C3	029E							
LDRU	02C7	02F0	0313						
FSA1	02CF	02D8							
WO	02DD	00E9							
W00	02E6	02E1							
W01	02EB	030F							
HUSER	02F4	02C5	0319						
LCRLP	02FB	0308	0569	057B					
LO	0302	00EC	0571						
CSTS	031C	00EF	0107						
CRLF	032D	0048	05F7	0622	06AD	06FB	0741		
TON	0336	004B	05F1	0249					
TON1	033A	0349							
TON2	033C	033C							
TON4	0346	0350							
TON3	034E	0342							
LKOM	0352	00B8							
LKL	0363	0359							
LKU	0367	035D							
INK	036B	00AF							
IKL	037C	0376							
IKU	0380	0372							
ASGN	0382	0361	0365	0369	037A	037E	039B	039F	
		03A3							
OUTK	038C	00BE							
OKL	039D	0393							
OKU	03A1	0397							
COPY	03A5	009D	03AF						
TOS	03CE	0003	03CE	0400					
USER	03C6	03E2							
ELOC	03CE	0764							
DLOC	03CF	0761							
CLOC	03D0	075E							
BLOC	03D1	075B							
FLOC	03D2	0767							
ALOC	03D3	0758							
XLOC	03D4	07A2							
YLOC	03D6	07A5							
ILOC	03D9	079F							
LSLC	03DA	079C							
HSLC	03DB	0799							
ESLC	03DC	0793							
DSLC	03DD	0790							
CSLC	03DE	078D							
BSLC	03DF	078A							
FSLC	03E0	0796							
ASLC	03E1	0787							
SLOC	03E3	0776							
EXI1	03E4								
LLOC	03E5	076D							
HLOC	03E6	076A	0770						

5. Anhang

PLOC 03EA 0773
 TLOC 03EB
 PLA 03F1
 RULOC 03F3 02C3
 PULOC 03F6 02F2
 ULOC 03F9 0317
 ZK 03FC
 FLG 03FE
 IOB 03FF
 LTOS 0032 0006

EBAS-88

CROSS-REFERENCE PN: DE

CR	000D	0521	055E	073C				
LF	000A	0562						
ETX	0003							
ROM	0F00	04A1						
TLOC	0FEB	0428	0454	05E0				
LLOC	0FE5	0440						
PLOC	0FEA	0446	05C4	0662				
EXI1	0FE4	041B	0481					
OMSK	00CF							
INSK	00F3							
RESTR	0400							
RSTO	041E	0423						
RSTA	0434	042E						
RSTB	043F	0439						
RST1	0440	0432	043D					
RST2	0459	0468						
RST3	0466	0462						
EXIT	046D	05B7						
HIL0	0484	0578	05B1	0618	0698	06A4	06BE	06E3
		06F5						
POC	048E	055B	055F	0589	05A2	06B2		
CONV	0495	04BA	04C1	054A	0551			
LREG	049F	071B	0752					
LADR	04B0	0451	056C	05FE	0608	06FE		
LBYTE	04B5	04A7	04AE	04B1	0575	0675		
EXPR	04C7	04D2	0564	0586	05AA	05F4	060C	061F
		068C	06AA	06BA				
EX0	04D2	04CD						
ERR1	04D4	04CF	04DA	04FE				
PARAM	04D7	04C7	05D3	0670				
PA0	04DC	05C0	0683	072A				
PA1	04E1	04F8						
PA2	04FA	04E5						
NIBLB	0506	04E2	052B	0536				
NIO	0514	050E						
PCHK	0518	04D7	05BB	067C	0705	0722	077C	
P2C	051B	04FB						
IBYTE	0527	062E	0634	0638	063C	0645	064D	0665
		066A						
OADR	0540	0593	06CB					
OBYTE	0545	0541	058F	0598	059D	06C8	06CF	06D3
		06DB						
OCRLF	055B	06E0						
DISP	0564	00A0						
DIO	0569	0583						
DI1	056F	0581						
EOF	0585	00A3						
LEAD	05A0							
LE0	05A2	05A6						
FILL	05A9	00A6						
FIO	05B0	05B4						
GOTO	05B7	00A9						
G00	05CA	05BE						
G01	05CF	05DB						
G02	05E0	05D8						
G03	05E3	05EF						
G04	05F1	05CA						
HEXN	05F4	00AC						
MOVE	060B	00BB						
MVO	0612	061B						
READ	061E	00C4						
RED0	0625	062A	0660					
RED1	0645	064B						
RED2	065C							
RED3	0662	0631						
SUBS	0670	00C7						
SU0	0674	068A						
SU1	0689	0680						
KNOW	068C	00B5						
ST1	0692							

```

ST2 0694 069B
ST3 069E 06A7
WRITE 06AA 00D0
WRO 06B2 06B6
WR1 06BA 06C1
WR2 06C3 06BC
WR3 06D2 06D7
VER 06E9 00CD
V2 06F0 06F8
V1 06FB 06A2 06F2
X 0702 00D3
X0 070C 0713 07B3
X1 0718 07D0
X2 071B 073F
X3 0737 0731
X4 0738 0726
X5 0741 0708 077F
X6 0744 0755
ACTBL 0757 0702
NREGS 000B 070A
ZR 0779 00D9
ZACTL 0786 0779
ZREGS 000B 0781

```

Maschinencode

```

:10140000F3E5D5C5F5DD5E5FDE5ED57F508D9B5D5FD
:10141000C5F5D90821180039060BEB21E40F2B7212
:101420002B73D110F9C10BF921E80F7E912320040E
:101430007E98280C23237E91232004F7E9828010384
:1014400021E50F7323722E9712370C5CD21014070
:10145000E1CDB00421E80F16024EAF772346772380
:1014600079B028027E02231520BFC34800D1C1F1D4
:10147000DD1FDE108D9F1ED47E1D1C1F1D908E1A4
:10148000F9C3E40F237CB537C8B957A9CC9E34E3A
:1014900023E3C3E900E60FC69027CE840274FC923B8
:1014A0005E160F2346231ACDB50405F81B1A18053E
:1014B0007CCDB5047DDF50F0F0F0F0CD9504CDECC005D
:1014C000F1CD9504C3EC00CD704E3E50D30032046
:1014D00003C920F3C34100CD180528F8C521000039
:1014E0004C7CD060538134779291729172917299D
:1014F000174F78B56FCD0C0118E778CD1B0520D4B8
:10150000E360E379C1C9D630D8C6E9D8C606F2147B
:101510005C607D8C60AB7C9CD0C01FE20C8FE2CE7
:10152000C8FED037C83FC9C5DE300CD0605070786
:101530007074FCDE300CD0605B14FB25779C1C9EA
:101540007CCD45057D5F0F0F0F0F0CD9504CDE900D4
:101550007BCD9504CDE9007B8257C9CD8E040DCD9E
:101560008E040AC9CD704D1E1CDFB02CDB0040E73
:1015700020CD02037E8DB504CD8404DAFB027DE8E6
:101580000F208C18E40DCDC704CD8E043AAAF57CD33
:101590004505E1CD40053E01CD4505AF92CD450560
:1015A000063CCD8E040010FAC90CCDC704C1D1E1B0
:1015B00071CD840430FAC9216D04E3CD1805280AE1
:1015C000CDDC04EB21EAF0722B733825110200CD1C
:1015D00021012CDD704E51438061D20F2D241009B
:1015E00021E80FC1712370230A77233EFP021520E0
:1015F000F2C32D03CD704CD2D03D1E1E519CDB044

```

```

:1016000004CDFD01E1B7ED52C3B0040CCDC704C158
:10161000D1E17E0203CD6D00CD840430F5C90DCD3E
:10162000C704CD2D03CDE300FE3A20F9AF57CD27F7
:1016300005282F5FCD270567CD27056FCD27054BE3
:10164000E52100FF39CD270577231D20F8CD27059B
:10165000C24100D1E1EB19060009EBE32B1EBDB807
:1016600018C321EAFCD2705772BCD270577E1C9D0
:10167000CD704D87ECD504CD21012DCD1805D808
:101680002807EBCDDC04EB73D8231E88CD704069C
:1016900000D1E1E57DACA877CD840430F7E17DACB5
:1016A000A8BE2057CD840430F5C9CD704CD2D03B5
:1016B000D1E1CDBE043A011000E5040D2805CD845A
:1016C0000430F7E1D5160078CD4505CD4005AFCDD6
:1016D00045057E8D45052310F9AF92CD4505D12BAB
:1016E000CD5B05CD840430CAC90CCDC704C1D1E19B
:1016F0000ABE200703CD840430F6C9CD2D03CDB03A
:1017000004C9215707CD180538370E0BBE28092309
:101710002323D020F7C34100CDFD01CD9F04CD2132
:10172000012DCD1805D8281E85C5CDDC047D12F1BA
:10173000B7FA3707137C12E1A1FB6F878F80C81878
:10174000DACD2D03CDFD01AFB6F84E8D8600CD21AB
:10175000013DCD9F0418ED41D30042D10043D0009C
:1017600044CF0045C80046D20048E6004CE5004D8F
:10177000E60150EA0153E301FF218607CD18053841
:10178000C00E0BC30C0741E10042DF0043DE004402
:10179000DD0045DC0046E800048DB004CDA0049D9BA
:1017A0000058D50159D701FFFFFFFFFFFFFFFFFFFFE3
:1017B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF39
:1017C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF29
:1017D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF19
:1017E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF09
:1017F000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9

```

5.2.3. Programm Disassembler in Maschinencode

```
:10180000CD704C103DDE1110000DDE5E17D917C80
:1018100098D0193AFF0FFEC0C4900AD93E03CD46B6
:101820000A1E00DD7E00FEDD20041E801806FEFD7F
:10183000200A1ECDD7E0208DD2318E721ED08FE28
:10184000ED2813FECB2018CBC32E7ECB7B280BDDDF
:1018500023DD7E0118092B9E1E20DD23DD7E00572C
:101860000CB7E2328FB7AAE23A62320F4CDEA09CD34
:10187000420ADD23D918937A0F0F0FA1C9FDE900A1
:101880000C7B02F440C042495402682CF480C0522C
:10189000455302682CF4C0C053455402682CF440F0
:1018A000C71A03F641C71B02F642CF1F02122CBBE8
:1018B0004ACF1D02122CEB4BCF016B0A70A943CF0C
:1018C0000128700BEB44C74E45C74DC0F08C945CF23
:1018D00008CE46FF0DB056FF0DB15EFF0DB247E7D3
:1018E00001F867F777C4A0E46AFA00002AF276FFED
:1018F00048414CD440C001762CF406C701762CEFF49
:1019000002E701F93AFF01410A70A932FF0128708C
:101910000BC101C0F16B2CF02AFF01750A70A922BF
:10192000FF0128700BF5F9FF01132CF5F5FF058475
:10193000C5CF05EBF1FF0684C1CF06EB20E74A5285
:101940007E20F118FF4A5203F110FF444A4E5A20FC
:10195000F1CDF07F0C4C743417D20F0C9FFB88C027
:10196000C752FDE9FF4A4D5020ED04C669F609CF84
:101970001C02752CEB03CF16EB0BCF17EB80C07F4F
:1019800002F4C6C77F02EFC3FF4A4D5002F0C2C740
:101990000C7D20F0EBFF09112C92D9FF4558D80897
:1019A000FF455884E3FF0928130BF507E77CC1DBEB
:1019B000FF1A03306FC8D3FF1B02306FC827FF44E4
:1019C00041C12FFF19CC3FFF4343C6F3FF44C9FB7E
:1019D000FF45C937FF5343C6C7C7525354026CC8AB
:1019E0000FF4E4F0DB1CB7E23C07EB5E67F21E5B0
:1019F00009E5FE203850FE60384A0E07C6893807D0
:101A0000212A0A856FE8E9213B0B856FE8CD770821
:101A10003CCB7E2328FB3D20F818CF2AA1A6B14B52
:101A20006567546A793342CBADC6CD7708C5CD958D
:101A3000041865CB7B2806DD7EFECDAE0A6A26EDB6
:101A4000184E3E1E185021430B18C57AE638C5CDF6
:101A5000B504C1C9DD233E30CD960ADD7E00CD4EF2
:101A60000A3E481831AF08791864DD23DD23CD9D87
:101A70000ADD6600DD6E9F1817DD23D9D5D9E1DD5B
:101A8000B5C10923CD9D0A0A4F1706Q0300105095B
:101A9000C5CDB004C1C9C54FCEDEC00C1C93E4D187C
:101AA000F521720B180321800B7AA1189C3E0218B5
:101AB00004CD77081F216B0BFE02208DCB7B28EB17
:101AC000C602CB7328E5CD770818017A3CA1CB7B01
:101AD0002827CB43204FFB05381F477AEE70E67863
:101AE000283D7AEE46B6C7283678B92004CBCDD10
:101AF00023C603CB732802C60321E80BCD100ACB03
:101B00004BC8CB8B08B72813473E2BF2140B78ED4C
:101B100044473E2DCD960A78CD4E0A3E291880784E
:101B2000B928CA18D4F579CDEDOAF1B9C8477AEECB
:101B300040E640C8AFCD460A7818BE9ADCCCA3CD
:101B40009FBCB6892681188320A002A041C65055AB
:101B50005348A0504F5082434144CA05245D4456D
:101B600058832CA829AC4AD0A94D8352CC52D2423A
:101B7000C344C548CC53D049D849D91A43821143EC
:101B800082CC443D049CE4F55D44144CA44144C390
:101B9000535C25390414EC4584FD24F52A019A032
:101BA000FBFB8E8F0EC30FC30EA00FA0534CC1536F
:101BB00052C1534CD3538EC9C449D244D2EDA5A7F
:101BC000A04EC34A050CF50C550A04DA028100B2D
:101BD000C1410A10A928110BC1410A11A9492CC100
:101BE000522CC1412CC9412CD2C1C2C3C4C5C8CCDE
:101BF0002812A948D84CD8289448D94CD92895FF00
```


5.2.4. Zeichengeneratorinhalt

1-k-Speicherbereich

```

:1010000000000000FFFFFFFFFF000000000000000000B4
:10101000101010101010101010101010F8F8101000
:1010200010101010101010101000000000F0F810101A
:101030001010101010F0F0101010101000000000030
:1010400000000000001F1F101000000000F0F0101042
:10105000101010105474101010101010F0F00000048
:10106000101010101010101000000007C7C20200000CC
:101070007E7E0000C2C2000000007E7E2C20000070
:101080008181A5A59898FFFF8181A0A08484FEFA0
:1010900010101038387070000000000000F8F8545448
:1010A000101012121E1E121210105050505090901C
:1010B00030301212C2C20404000004040F82020F0
:1010C0000000000000C0C0202010108080000000040
:1010D0000000000000000040400005C5C5C5C000098
:1010E0000808020200000008080202000004040D0
:1010F00000001E1E547C101010108080402020E4
:1011000000000000000000000101010000010107F
:10111000282828280000000028287C7C2C6828B03
:10112000101010787878101064640808020204C6F
:1011300010101028285C5C343420204040000000005F
:10114000080802020202008080202008080808020F5
:10115000000007C7C5454000000007C7C10100000D7
:101160000000000002020404000007C7C00000000C7
:1011700000000000000040400404080802020404017
:10118000383854544444383810101010101038387F
:101190003838040420207C7C7C0808040438381F
:1011A000080848487C7C08080407C7C04043838A7
:1011B0001C1C7878444438387C7C0808020202087
:1011C00038387C7C4444383838387C7C040470706F
:1011D000000000000000000000001010101020208F
:1011E0000808040402020080800007C7C7C040000A7
:1011F00020200808080802020383804041010101097
:10120000383854545C4438381010444444444444FE
:10121000787824242424787838380404038387E
:1012200078782424242478784040787840407C7C66
:10123000404078784040404038385C44444438381E
:1012400044447C7C444444443838101010103838E
:101250001C1C0808484830304444606058584444D6
:101260004040404040407C7C6C6C5454444444476
:10127000444454544C444438384444444438382E
:101280007C7C787840404040383844445C34345E
:101290004C447878585844443838787804043838BE
:1012A0007C7C10101010104444444444443838DE
:1012B00044446C6C28281010444454545454282836
:1012C00044442828282844444447C7C10101010AE
:1012D0007C7C0808020207C7C545444444444448E
:1012E0003838444444443838545444444443838E6
:1012F0003838000000000000000000007C7C86
:10130000FFFFFFFFFFFFFFFF000038383C3C3C85
:1013100040404444444478780000404040403C3CD5
:101320000404444444443C3C00004444040403838B5
:10133000080801010101010000044447C7C0404B5
:10134000404064644444444101030301010383835
:10135000101030301010707060602424303024245D
:1013600030301010101038380005454545454575
:1013700000006464444444400044444444383815
:1013800000000444787840400000444444440404AD
:10139000000064644040404000004040383878A5
:1013A00010103838101018180000444444443838DD
:1013B00000004444444410100005454545428285D
:1013C00000006C6C1010444000044428281010A5
:1013D00000007C7C10107C7C242438383C3C3C55
:1013E000444444444444383844444444443838ED
:1013F0000000181838382C2C040414747070404065

```

2-k-Speicherbereich

```

:1014000000000000FFFFFFFFFF000000000000FFFFE2
:10141000101010105070FFFF10101010FBFFFFF96
:101420001010101010101F1FFFFF00000008F8FFFFF4C
:1014300010101010F0F0FFFF101010100000FFFFF50
:101440000000000001F1FFFFF00000000F0F0FFFFF82
:10145000101010107C7CFFFF10101010F0F0FFFFF38
:101460001010101010101E1FFFFF00007C7C2020FFFFC0
:101470007E7E0000C2C2FFFFF00007E7E2C2FFFFF70
:101480008181A5A59898FFFF8181A0A08484FFFFF9A
:101490001010103838F0F8FFFFF0000000FAFFFFF00
:1014A000101012121E1E1FFFF101050505050FFFFF60
:1014B00030301212C2C2FFFFF00000404FAFFFFF28
:1014C0000000000004040FFFFF101080800000FFFFF80
:1014D0000000000000000FFFFF00005C5C544FFFFB8
:1014E0000808020200000FFFFF080802020000FFFFFD8
:1014F00000001E1E7C3CFFFF10108080404FFFFF5C
:1015000000000000000000FFFFF101010100000FFFFF9F
:1015100028282828000000FFFFF28287C7C787CFFFFF3
:10152000101050507878FFFFF6464080802020FFFF7
:10153000101028284848FFFFF202040400000FFFFF9F
:101540000808020202020FFFFF202008080808FFFFFAF
:10155000000007C7C5454FFFFF00007C7C1010FFFFFD7
:101560000000000002020FFFFF00007C7C0000FFFFF47
:1015700000000000000000FFFFF0404080802020FFFF17
:10158000383854544444FFFFF101010101010FFFFF1F
:10159000383804042020FFFFF7C7C08080404FFFFF87
:1015A000080848480878FFFFF70707C7C0404FFFFF37
:1015B0001C1C78784444FFFFF7C7C080802020FFFF37
:1015C00038387C7C4444FFFFF38387C7C0404FFFFFBF
:1015D000000000000000000FFFFF0000101010FFFFFCF
:1015E0000808040402020FFFFF00007C7C7070FFFFF57
:1015F00020200808080808FFFFF383804041010FFFFF7
:10160000383854544040FFFFF101044444444FFFFFB6
:101610007C7C24242424FFFFF383804040444FFFFFCE
:101620007C7C24242424FFFFF704078784040FFFFFB6
:10163000707078784040FFFFF38385C4444FFFFFAE
:1016400044447C7C4444FFFFF383810101010FFFFFB6
:101650001C1C08084848FFFFF444460604848FFFFFDE
:1016600040404040404040FFFFF6C6C54544444FFFFF6
:10167000444454544444FFFFF383844444444FFFFF36
:10168000444478784040FFFFF383844444848FFFFFDE
:10169000444478784848FFFFF383878580404FFFFFBE
:1016A0007C7C10101010FFFFF444444444444FFFFF6E
:1016B00044446C6C2828FFFFF444454545454FFFFFA6
:1016C000444428282828FFFFF44447C7C1010FFFFF56
:1016D0007C7C080802020FFFFF3838202020FFFFFD6
:1016E0000000020200808FFFFF383808080808FFFFF1E
:1016F000383800000000FFFFF00000000000FFFFF7E
:10170000FFFFFFFFFFFFFFFF000038383C3CFFFFFBF
:10171000404044444444FFFFF000040404040FFFFF3D
:10172000040444444444FFFFF000044440404FFFFF9D
:101730000808010101010FFFFF00004447C7C3CFFFFF1D
:10174000404064644444FFFFF101030301010FFFFF2D
:10175000101030301010FFFFF606024243030FFFFF85
:10176000303010101010FFFFF000054545454FFFFF8D
:1017700000006464444444FFFFF000044444444FFFFF0D
:10178000000004447878FFFFF000044447C7CFFFFF65
:1017900000006464404040FFFFF000040403838FFFF15
:1017A000101038381010FFFFF000044444444FFFFF7D
:1017B000000044446C6CFFFFF000054545454FFFFF7D
:1017C00000006C6C101044400004442828FFFFF4D
:1017D00000007C7C1010FFFFF080810101010FFFFFA5
:1017E00010101010101010FFFFF202010101010FFFFFD1
:1017F00000000005454FFFFF04041442020FFFFFD5

```

5.2.5. Programm BASIC-Interpreter in Maschinencode

```

:10800000C3B7ACC37381C3F90FC3B000C3E300C3BC
:10801000E600C3E900C3E000C3E000C3F200C3F500
:1080200000C300AEC33E00C32DE7C31E8FA39FAD0E
:10803000A00C9F0680F68ED4932F8F0EA70EAE8EBAE
:1080400009D647A4A850A3D3A8EAA86A9C9992BA6
:10805000092B94A892E8B6926C92F6922A8F2EF7
:108060009379A7A1798D9C7C3B9E7CBF9E7F17A7AF
:1080700050BD8C46BC8CC3B66A853B8B0B88558A69
:10808000658DA68A2788C5871D9023896A86A8C875B
:10809000E687D088C186DD930689FC86E393368FDB
:1080A000719C5C895C89550A72871D90C5940D8886
:1080B0000D885989DB97D897C49946946294F986FC
:1080C000659498999B995C896D81599C269CF9982E
:1080D000B69B4F0AB09A9D87FE94D3822A8AF29A07
:1080E0006B9A69A9F299F5994195FE84EC84B39569
:1080F000A397FE97F786210400390111007E23FE35
:1081000081C07E23E56667FAB3EB2807EB7C922073
:10811000027D93E123C80918BA8CD3EB1C5E3C1B7D0
:10812000E58D52C5E3C1A00C39DD882313424BD180
:10813000C9852A620C06000909CD3EB1E1C9D5EBEB
:1081400021D8FF397C922007D93EBD1D018071BF5
:108150002DBEC478618163A4F0CB7200AC1248493
:108160000ACDD8B84C3F4822A4C0C22540C1E0201F7
:1081700010B01181E0118A0A11E2041E01CDD87F
:1081800082CDF882CDE889212A2B1D2807CB7E234C
:1081900028BF18F6CDBD842A540C7CA53C0C4A1193
:1081A000CDFB8232A8B0CCDD88921FFFF22540C3A93
:1081B000AB0C8728252A9A0CF45A28D59B0CC019C2
:1081C000DAB688229A0CC5CD99A1CD6A84CD0784BB
:1081D000CDA786D1B728C937180ACDD784CD478641
:1081E0003C3D28C5F5FE803CA9086D921AB0C343539
:1081F000D9C4987FE20200123D5CD098347D1F171
:10820000D21486D5C5237EB7280CFE2028F7FE0998
:1082100008F3F80A28E8F5CD9282C53014BB2A5ED2
:108220000C1A0203137C9220027D9320F4ED435E2E
:108230000CD1F12821ED4B5E0C8109583CD1981B16E
:10824000225B0CB87423D71323722311B10C1A19
:10825000772313B720F9CD28223EB21A981B562P0
:108260006B7E23B6C8232323AFBE2320FCBB7323EE
:1082700072418BC110000D5280CD1CD8586D5280EBA
:108280003EABCD518111FFFC48586C26D81EBD121
:108290003E52A5C0C44D7E23B62BC823237E23C2
:1082A000666F82560697E23666F3FC83FD0D91973
:1082B000D918E2C02A5C0CAF32A30C772377232B3
:1082C0005B0C2A5C0C2B22500C2A40C22480CCDB8C
:1082D0007B86E2A5B0C22600C22620CC1B7B5A0C5C
:1082E00021080C22060CAF767F22580C324B0C325C
:1082F000A70C85C52A5B0C9A9F32000CFD210F8038
:10830000FD228B0CFD21900CC9A9F32030C805111D
:10831000B10C7E47B7CAD83FE21384F822287BC1
:108320003A030CB7477E20A3FE30380AFE3A383B10
:10833000D51163A9B5180223131AB72825FE2020BA
:1083400009137FE2020032318F8EB1AFE60380E282
:10835000E85FAEB67FE20491A1730DBF178F68056
:108360001808B17FE603802B65FD1EBFEC9363ABE
:10837000200202C3EB2312130CD63A280AF9E8068C
:108380003A2811FB49200332030C30C6542805D6B94
:10839000C21283477EB72815B828D023120C1318B1
:1083A000F381E504EB87E2328FBB188C21A00C2A
:1083B0001213121312C9783DFB2600284CD7084FD
:1083C0001008CD7084CD7084183A28BC668ACD94
:1083D000C984FE7F28EAF5CD7084F118363E3FCD82
:1083E00072843E3FCD7284CD6484CD078427E2B7FE
:1083F000FE0309C9DC82F53E58CD7284F1F64018D5

```

```

:1084000071CDF683CD0E8A21B10C06017832A70C0E
:10841000CD984FE07283CFE0DCAE989FE03CAE1B6
:1084200089FE1528DCF0E8288DFE122014CDF68367
:10843000CD0E8A21B10C480D28D67E23CD66841836
:10844000F6FE09280FE0A20060528BB0A1804FEC5
:108450002038BD4F78FEFD3E07300479772304CDE8
:10846000668418AC3E20FE0A2008CDE893E0AC97B
:108470003E5CC54F3A000CB7203F79FE092027FD2E
:108480007B0186F83DFDBE003817FD7E00B6072FB1
:10849000C609470E20CD8D0CFD3A0010F80B0918CA
:1084A00018CDE8918F7FE20380CFD7E00FDBE01C8
:1084B0000CCE8B9FD3400CD8D0C79C1B7C97ECBEP20
:1084C000CD6684CB7E0C2318F4CD0980E67FFE18EC
:1084D000280FFB0FC0CDF3833A000C2F32000CAFF3
:1084E000C9CDF383CDE89C2D480AFC9CDA695C18A
:1084F000CD7382C5CDE89E1D14E23462378B128D4
:1085000033CD9386C5A2E34623C5E3B87C922002F0
:108510007D93C1381EB35C5EB0CD9A1E1CD6484EF
:10852000CD378521B10C01F484C52B0600CDF09028
:10853000C33491C43A08101B00C3E17803B723D7
:108540000C8E2F3C85FEC920010BD680E5163A953
:108550002808CB7E2328F3CD20F7FE202808B784
:108560000CBFF02FA3B85032318F03E29324E0CCDD7
:108570002788E3CDF680D1200209F98B080ACD3130
:10858000081E5CD08883E52A540C3E39FC5D1874
:10859000C888B88B85B5E5CD0A0E1C5DDE5D5018C
:1085A0000081D21000051597FE8A43E012008CD48
:1085B0004786CD858B5C0D0A0CDBF9F81C5DD852A
:1085C000D5F533E52A500CE30681C533CD18803C40
:1085D000CDE8622500C7FE3A2839B7C26D812392
:1085E0007E23B623CAC8865E2356ED53540C3AA3A5
:1085F0000C87E2820F5FCA659D5E53E3CDD7284EB62
:10860000CD9C91383E8CD7284E1D1F1873006CDFCBB
:10861000828738E0CD47861CC85D5C8D680DA2B45
:1086200088FE46CA8B88F81D380FD649DA6D81FE5A
:1086300023D26D8111B080180317680074F060098
:10864000EB094E2346E5E237FEFD52008237FE94
:108650003CAC8B720F8FE3AD0FE2028BFF0928E7FA
:10866000F0A28E3FE303F3C3DC9280FCD8586E54
:10867000CD9282D1D2818760691804EB2A5C0C2B81
:1086800022640CEBC92BCDA786CD98718BAC1C1E8
:10869000B71834CD18803CC0CD984FE14200CE539
:1086A0002A540C7CA53CC49D97E1C9FE13200BCD38
:1086B000C984FE3032804FE1120F5FE030CDDF38318
:1086C0003E0CF6C022500CC1F52A540C7DA43C2BB3
:1086D0000922560C2A500C22580CCDF882F121AFF9
:1086E000ACC29481C3A0811E112A580C7CB52856B7
:1086F000EB2A560C22540CEBC9CD9F3C93181B94FE52
:1087000032304FD77033E2CEBCCD4786CD1B9451
:10871000FD7704C97FE41D8FE5B3FC9CD888B182A
:108720000CCD4786CD858BCDBF9FFA44873A6B0C25
:10873000FE91DA75A0018091DD210000110000CCDD
:1087400036A051C81E05C37E812B110000237EB7C1
:10875000C8FE303FD0FE3AD0E852198197B7ED52DA85
:108760006D81626B19291929D6305F160019B8E16A
:1087700018DB2826CD2487CD4886C0E52A4040CEDD9
:108780007D932D4B81EB3225A0CE1C3C682CAC282BC
:10879000CD682010C85181C0E83093181C1E5E0B
:1087A000D5540CD57AB33C3A70C572004F3A27D0
:1087B000018CDD5C5CD987CD0D885E2A540C7C6F
:1087C0009220027D93E123DC9582D4928260692B02
:1087D000D81E08C37E81C016FFC9F6805623F9FE41
:1087E0008C180320EE8122540C7CA53C21A70C7EAC

```

167

5. Anhang

:1090000081CD9A8BCD4886C26D81183A0E02CD3142
:1090100081ED5B540CD516A0D533C3CC85200CCD87
:109020000E9D32440C2F32020C1809CDB9ABBCD487C
:1090300086C26D8116FFCDP680F9FEA01E13A027B96
:1090400081D1ED53540C3A020C3C28033D202ED123
:10905000F130132037E1C1702B7128C1702B712BE4
:109060000P1702B7118EAF5D521020CCB7E2B01774F
:10907000B711440CC41091B1F11FC38ABBD5B66CF
:109080000CCD8D9221440CCD19A018C3D83927EC0
:1090900022060CE177232371237018BA5E2A540CBF
:1090A000237CB5E1C01E0CC37E813EA0C5D181F66C
:1090B00080473E29324B0CC36F8DCD880BCDDDA10C
:1090C0000CD8D90CD639201C292C57E82323E5CD498B
:1090D00091E142E346CDE190E5CD5992D1C9CD49DC
:1090E0009121440CE5772323732372E1C92B0622D7
:1090F00050E5B5E7F237E0CB72806BA803B820F4EB
:10910000FE220C4786E323B79CDD19011440C3E5F
:10911000D5A2060C22660C3E0132020CCD19A07C29
:10912000220027D9322060CE17E0C1810C37B8138
:1091300023CD8D90CD6392CD6CA01C1DC0ACD6639
:1091400084FE0DCCFB890318F2B70E1F52A5A0CF8
:10915000EBA2480C4FAF478D427C922027D9338BA
:109160000272480C23EB1C9F11B0E28C0BF550100
:109170004B91C52A040C22480CAF08ED5B5A0CD960
:1091800021080CED5B060C7C9220027D930118391FB
:10919000203F2A5E0CEDE5B600C7C9220027D93280C
:1091A0000ACB7E2323CD929118EBC1EB2A620CE0D2
:1091B00052EB2844CD6CA0CB7BE50928EDB34806AD
:1091C0000090923D17C9220027D9328EDB501C4B9
:1091D00091C57E2323E2356232323CB877C8D5D940
:1091E0000C12A480CED42D9D8D9626B8D42D9D0909
:1091F0005059D908E5DDE1C908D0D92A480CB84F10
:10920000060092EDBD8B626B13DD7FD4DD73CC344
:109210007691C5E52A660CE3CD148CE3CD898B7BEF
:10922000E52A660CE5861E0FDA7E81CDD9E0D1CD73
:109230006792E3CD6692E52A460CB8CD4992CD49B3
:109240009221AC8BE3E5C30C91E1E34E2346237EFO
:10925000023666F78B1C8EDB0C960694F060018F396
:109260000CD898B2A660CEBCD8392EB0D505091B70
:109270004E2A480C7C9220027D9320054709224803
:109280000CE1C92A060C2B62B2B462B4E2B2B7C9248
:1092900020027D93C022060CC901328FC5CD609299
:1092A000AF5732020C7EB7C9CD9D92285823235E5A
:1092B00023561AC3328F3E01CDD890CD1E942A462E
:1092C0000C73C1C30C91CDF93AF3E34F3E5E57E338
:1092D000B8380278110E0005CD4991C1E1E52323C3C
:1092E00046236668060009444DCDE190CD5992D1E0
:1092F0000CD6792C30C91CDF93D1D51A9018CBEBCB
:109300007ED1430405CA4487C51EFFF2928D833EB6
:109310002CCD5181CD1B943E29CD5181F1E33DBE31
:10932000060030AA4F7E91BB4738A34318A0E13E08
:1093300028CD5181CD958BED5B660CD53E2CCD5162
:1093400081CD958BED5B660CD501FF007EFE2C2058
:10935000155C5D1894C11D431C28AA7FE2E2C2006DD
:109360000CD51894C14B3E29CD5181E3C5CD669240
:10937000C1EBE1E3D5C5CD6692C1D17896D2D19348
:10938000ED44B930014F23237E23666FE5C54806BF
:109390000009C1EB799638383C4F4623237E23667B
:1093A0006FE87B7281DC506001AEDB179C12020F2
:1093B0004FC5D5E51806131ABE20032310F8E1D1D6
:1093C0000C12009D1ED527DCD328FE1C979B720D6C8
:1093D000D1AF1E3C3D1E944FED78C328FCD0A9480
:1093E000ED79C9CD0A9447C51E00CD488628083E80
:1093F0002CCD5181CD1B84C1ED78ABA028FAC9EBEF

:109400003E29CD5181C1D1C543C9CD1B94D53E2C38
:109410000C5181CD1B94C1C9CD4786CD858BCD273C
:10942000877AB7C24487CD48867BC9CD9D92CA0E844
:109430009D5F23237E23666F85194672E3C57ECDCB
:10944000DEA0C1E170C93809CD1B80E8034FC31EF9
:1094500008CD1B94FE04D26D8147CD1180E86FCB00D
:109460001E8EBCD9F95CD1B94FE80DA44871B30270156
:109470004FD60E30FC6C1CED4481FD7702C9CD14D9
:1094800087DA6D814F06033CD0C80381B3C20F61027
:10949000F6CDD0C8038113C28F83DB98205CDBA94A9
:1094A0001E83D0E07C30F801E1318110603CDB694A0
:1094B000B720F810F8C9CD0C80D01E14C37E8114DB
:1094C0000C4B48218F5CD1487DA6D814FCD4786C2BA
:1094D0006D81E5C501FF08CD128010FBC1CD128062
:1094E0002A5E0CEB2A5C0C4E23CD12807C9220026B
:1094F0007D9320F301FF08CD128010FBC1C9FE9798
:1095000016FF280B32660CCDB482160021650CCDF7
:109510004786D7E942A5C0C0603CD0C8038A05F74
:1095200096A2201873CD38E17EB7320E810EB224C
:109530005830AAB0C87CCD8B9C356821E15C37EDA
:1095400081C08101FF08CD128010FBA2A5C0C7E2354
:10955000562328135E23562385CD188FCDDA12318
:109560000CD9595E17E9F0828050820CD12800CD37E0
:109570008E5B710CDD9595810CDD12800E0A39
:10958000CD128018C90E1ACD128001FF08CD1280AD
:1095900010FBC3A0817EB7C84FCD12802318FEBE15
:1095A00021E3E5BF0211580FDD22880CFD21DF
:1095B000950CC908EBE2A540C7CA53CC244872A5C54
:1095C0000C7E23B6EBCA0D88110A00D508DC85860F
:1095D000ED53A10CD1AFC2C2006CD4786DC8586E0F
:1095E000539F0C8E52A5C0C23235E35C237E2C20BE
:1095F0000EBC4786DC8586E2AA10CEDE52AD681ED11
:1096000053A0C0B7C26D81CD928C2D81876069D92B
:1096100010100021000D911FFFCDD9582D92EB5C
:10962000ED4B9F0C78AC6AD81CDC64E5BA10C4E
:1096300019DAB68E2A560C23234E36EB2AAC0CE2
:10964000ED423806280E80669180DEB8B9FC02A9AE2
:109650000C091823AA10C229A0CEBCD4786B7CA35
:109660006897FE882814FE8C2810FEA2280CFEC9DC
:109670002808FE9D2804F8B20E1CD478630DF229E
:109680009C0C2B1100000800CD4786302EB5B72136
:109690009896ED5230112189ABCD8D84CD9A97E1DA
:1096A000CD478638FB18B7626B2929129D6305F58
:1096B000160019EBE10C18D079329E0CE52AAC0CF
:1096C000EBE5ED52382ACD92826069D1D92AA10CFE
:1096D000ED5B9F0CD9CD958238132175ABD5CDBDEF
:1096E00084E1CD9A1CD9A97E17EC35E96D9E5D933
:1096F000D1E1AF0698CDAB9FCDDDA1060023E57E7D
:10970000B7280442318F83A9E0C9B283C381C4FC4
:1097100006002A9C0C545D09E52A5E0CEDE52ED42B0
:1097200044E1EDB0ED535E0C181EED444F0600C4
:109730002A5E0C545D09CD3E81225E0CEBE584BBB
:1097400009C0CD42444DE1EDB8D12A9C0C1EB7288F
:109750000577231318F7E52A5C0CEB5E82E17ED9
:10976000FE2CCA7A96C35E9623B623B62C23796D2
:10977000ED5BAC0CCD92826069ED4B9F0CEDE5BA173
:109780000C2323732372237E20F820F80EB23B654
:1097900023B620EEDC682C3A0812E9A00CCD1A1EA
:1097A000C3E89CD7382D1C5CD9282D24487545DF8
:1097B000C3E57C9220027D93D24487CDD889C121F1
:1097C0005682E3EB2A5E0C1A0203137C9220027D80
:1097D000494ED435E0CC93E7F013EBFF5CD9A68
:1097E0008BCD48862014CDBF9F3D2F47F14F2FA032
:1097F000473AA30CA1B032A30CC9F1C36D81CD854A

```

:10980000086C0E1CD9282D24487606923234E2346ED
:1098100023C5CD3785E1E5CD9A1CD648421B10C47
:10982000051E1FFC17E867F772320F8E1570600CD7A
:1098300000984FE3A3010FE30380CD6304780070716
:1098400008007814718E9E5212D9E83050A200104EC
:1098500009217198BE234E234623C5D9D8CD9C3F850
:109860000AD20F1F6B03804FE30380CD6304780070716
:1098700008420A49851C7864CD9846B5894916991F
:10988000044E6980D659952F984568995811994B8F
:109890000AF98A80E997F5999419C9800C1D1CDEE5F
:1098A000089C303987EB7C814CD66842310F9C9E532
:1098B00021FA98E337F5CD9844FF13534C8F5DC8A
:1098C0000FA987ECD6684F1F53005CD75991802239E
:1098D000147EB72805B920EB10E9F1C9CD2A85CD52
:1098E0000EB89C1C315987EB7C83E5CCD72847EB741
:1098F0002808CD6684CD759910F43E5CC372847ED1
:1099000007BC8CD984CD668477231410F2C9360058F
:109910005A06FFCDA498CD984FE0D2848F1BC869
:109920000F87F200F15142812321B57EC6684CD7571
:109930000991E837F5BF8FF3808F13E07CD72848D5
:109940000D5921C4D5E6B6260019444D03CD1F8111
:10995000D1F1CD6684772318ED7AB7C8152B7ECD9B
:109960000668410F5C9CD2A85CDEB89C1D137F521A0
:10997000B10C1E81E51D7EB728072372B27321
:1099800018F5E1C9C0EB21870CED5F7723772377CA
:10999000237723E67F77233680EB9C9DF95C0E5FB
:1099A0002A5E0CCCEB9CD9386ED5B600C7C920418
:1099B00002793283B4E237E23C7F202BCD7284C8
:1099C00079E67FC472843E24CB79C47D83A3E3DCD57
:1099D000728497179F22660C852009CD8C9CFCDDBE
:1099E000A1CDBD90CD3491E1232323232318B37C
:1099F000E1C9CD84E22A580C2BCD649A38278FB9B0
:109A00000F87F28F5237CD3E2178D60D201377236C
:109A1000723772377CD649A380BF8E0A28DBB718B3
:109A2000DDFE0D20D4360023361A2A5E0CE5FED684
:109A30001A200932AB0CCDC282C3A0817E23B7208D
:109A4000FB7E2B728FB2B29A0CE13E3FF32AB0CA6
:109A5000CD348663C3DCA9813005F5AFC3F1811ED2
:109A600018C37F81CD0C0D8E67FC9CDBA8221B0E9
:109A70000C0600CD649A383228F9FE7F28F5F81ACC
:109A80002828FE0A2004040528E94F78FEFF7928DB
:109A900002230477D60D20DB773EFE32AB0C21B0DB
:109AA0000CD47863C3D28C618A8CD282C3A081EE
:109AB00032660CCD478611FFFF280B3E2CCD51811A
:109AC000D26D81CD8586C26D81ED539A0C3FBD32FB
:109AD0000AB0CC307953CA2D9A3C2892CD282AFED
:109AE000232AB0CED5B9A0CD7AA33C0C1C85C5C8C3A4
:109AF000DA87C8D2487CD8586ED539A0C110A00C7
:109B0000FDC20066CD4786CD8586ED539C0CF8EADB1
:109B1000CD5181CD73E2D4C56069D9110100210079
:109B20000D9CD9582D24487E5D9EBED4B9C0C78DA
:109B3000B1CA6D81CD6A0ED5B9A0C19DA6B8EE5CA
:109B4000ED5B9A0CCD9282DA4487D1C560697E23A1
:109B5000B6280A237E23666FED52AD6A8B8C1D1D1FF
:109B6000E5ED5E23BED42E8F380AED42DA4487098A
:109B7000EBE1E519E3E55059ED4B5E0C9E5CD1934
:109B800081E122580C6069C1E3D5EDB0E1ED4B9A55
:109B90000C2323712370237EB720FB23D17C9220DA
:109BA000027D93280BD5EB2A9C0C09444DEB18E160
:109BB000CDD8B9C35682CD6A2D1E5E216D0CCD19DB
:109BC000A02A600CE33A020CF53E2CD5181CD6AFF
:109BD0008BCD1A020CA81FDA908BE3EBE52A660CEA
:109BE0007C9220027D93C24487D1E1E3D5CD19A0B8
:109BF000E1116D0CCD19A0E1C93801324E0CC36AD2

```

```

:109C000008DE519EB2A620CB7ED5E3C1EB1B1B1D70
:109C10001B2802EDB0DD53620C324E0CE17FE2C9F
:109C20000CCD478618D3C8ED739A0CE511579CD563
:109C3000CD24870800D5C488628113E2CCD5101CE
:109C40000CC5CD2A87C1EBE3EB1EAD5ED2A9A0CDF
:109C50002B722B732BD9C91C93800CA1E94C0FE91
:109C600008286FD248732AF0CC9CD27871AC332EE
:109C70008FC2487D532ECCD5181CD1B94D112C9D7
:109C8000210847CD05A180CCD05A01804CD1DD8161
:109C9000D1CD2D97F8B7C83A6B0CB7CA75F9903038
:109CA00011ED44D9DD5CD02A09DDE83CDF59FD995
:109CB000DDE1F829D0F5CD21A067F1CD8F9DB42146
:109CC000660CF2D69CCD5A9D30692334CA559D2E20
:109CD00001CDAC9D185DAF904778B9F5237E9A5768
:109CE000237EDD9DD6F237EDD9CD6F237E994F262
:109CF000DC729D6863AF4779B72027DD4CDD7DDDE1
:109D000067DD6AA54656F7BD608FDE020E6AF32C1
:109D10006B0C90529CB1208DD29083002DD2308A8
:109D2000CC1F2139D785C45B72808216B0C867720
:109D30003D0CC878216B0C87FPC459D462737E86805D
:109D4000A94FC3F59F1CC014C0DD2CC0DD24C00C7E
:109D5000C00E80B8034C018E06C37E817E33F237E8A50
:109D6000057237EDD8DD6F237EDD8DD6F237E89CD
:109D70004FC9216C0C7E277AF67904772ED52F6
:109D8000EB6FD9DD6F7DD9CDD677D994FC90645
:109D900000D6083810435ADD508DD7CDD6F08DD3C
:109DA00061E80018ECC6096AF2DC8791F4FDD7C1E
:109DB0001FDD67DD7D1FDD6FCB1ACB1BCE1818E8CD
:109DC0000000000000081062385AC3117F53CB9EA9
:109DD000B7237CCFEA60D5737FCB5C60BB1380DD29
:109DE000E34C9876805C293BAA3882CD8BF9BF7EA24
:109DF0000487216B0C7E013580DD21F30411FA3399
:109E000090F570D5DD85C5D9C949CC1DD1D104CDE3
:109E1000C39E21C09CD889C21C69D0CDBA701801E
:109E20000DD210000110000CD949CF1CDA1A101A5
:109E30003180DD21177211D2F71804C1DDE1D1CDD7
:109E4000BF9F0C82800CD639F79D5D94F41DD5E8105
:109E5000D9910005058DD2100021F39CE5216864
:109E60009E8E5E5E5E21660CFE23B7200E435ADD2D
:109E70005508DD7CDD6F8DD614FC995E81E081F6D
:109E800057793012E5D9E319E3EBDD5E3ED5AE369
:109E9000DDE1EB89D9E11F4FDD7C1FDD67DD7D1F33
:109EA000DD6FCB1CCB1DCB181D7A2003EBE1C9CDD8
:109EB000DA9F012084DD210000110000CD759FC153
:109EC000DDE1D1CDBF9FCA70812E8FCD639FDE53F
:109ED0003432BE5D9814E2B562B5E2EDB6E674F
:109EE000EBD941EBDD5FDE1AF4F575FDD2100030
:109EF00032AE0CE5FDE5C5E578D9E3B7ED52E3EB0D
:109F0000FDC3E3D52E3FDE1EB99D9E1473AAE0C13
:109F1000D8003F300932AE0CF1FC1F13718014F1D1B
:109F2000E1E10C0D1FFA5B9F17C1B3C1208DD2960
:109F3000083002DD2308CB112908FD29083002FD75
:109F40002308CD103AAE0C1732AE0C79B23DDBA45
:109F5000DD5209FE5216B0C35E12097182DFD8143
:109F6000C3349D78B72807D216B0CAE80471FA895
:109F70007F82B69FC68077CAAD9ECD21A0772BC987
:109F8000CDBF9F2F7E721B7E1F209D3C359DC02E8
:109F9000A078B7C8C60238F347CD949C216B0C3427
:109FA000C018E8CDBF9F0688110000216B0CAFD56B
:109FB000DDE11100007060023368017C3F99C3AE3
:109FC0006B0C87C83A6A0CFE2F179FC03CC9CDBFB7
:109FD0009F0216A0C7E8E8077C9BE2A660CE3E5E0
:109FE0002A680CE3E52A6A0C83E5EBC911660C016B
:109FF0000600EDB0C9ED53660CD22680CED436A36

```

5. Anhang

:10A0000000CC921660C5E2356234EDD69234EDD61AB
:10A010002342364623C911660C010600B08EB0B7D
:10A020000C9216A0C7E07371F773F1F2323777907E3
:10A030000371F4F1FAC978B72885CDBF9F792888B5
:10A04000216A0CAEF79FAC89CD50A01FA9C3C89F42
:10A050000237BEC02B79BEC02BDD7CBEC02BDD7D3E
:10A060000BEC02B7ABEC02B7B96C0E1C958235623AF
:10A0700004E234623C947F757FDD67D66FB7C85FD
:10A080000CD02A0CD21A0AE67F29BA01B7AA33C20FD
:10A090000ADD2BDD7CDDA53C20010103EA890CD6F97
:10A0A0009D717DC459D0600D7D9D81C9216B0C8F
:10A0B0007FEFA83A660CD07ECD75A036A87BF579D9
:10A0C00017CDF09CF1C921000078B4C83E113DC800
:10A0D000293800EB29EB30F60930F3C36B8BF26E6
:10A0E000CA71A1FE2D52805FE2B28012BCD0E9D52
:10A0F00047757F2F4CD47863056FE2B2826FE4500
:10A100002025CD478615FEA6280E2B2D00A14FE12
:10A110002B2805FEA528012BCD478638491420079A
:10A12000AF935F0C0C28BCE7B90F239A1CDDAF97E
:10A13000ED371C2C09DC8C9737F5CDAEA120FAD78
:10A1400003007C17DDE1D1C3DC39ED1F1CC29FBE9A7
:10A15000D571788947C55F5D5CDB8E9FF1D620CA1AD
:10A16000A1E1C1D1188F788787838706D6305F189B
:10A17000A7110000CD47862822380CD641381CFE82
:10A18000063018C60A1802D630087AFE10D2559D3D
:10A1900008BE29292929B56FEB18D95E3CD188FE1E
:10A1A00009CDDA9FCD69FC1DDB1D1C3949C08F58E
:10A1B000CD8E9FF13DC09D5E5F5CDA9F9F1B1D13C06
:10A1C0000C9B52193ACDDBD84E1CD178F21AFOC7E05
:10A1D0003600F5CDDDA1F132AF80C33091AF32AA1C
:10A1E0000C216E0C3620B6082802362D0CDB9F77DC
:10A1F000FAA1362D5E3CD29FE1B42376303AA0C330
:10A200005717DAB2A2CAA4A285D1F4A21AFOC3417
:10A2100035285157C60BFA55A2B28023037477E63
:10A220000903C4F047A160BE123CD83A35FA0100E8
:10A2300000EDB12B013CA2C55AFF51847817FE2ED24
:10A2400008FE20C8FE30280AFE252005237FE2DEDC
:10A250000C82B3620C94E0D28010C0602E1237A16C0
:10A26000000C3D7A3010003C60CFA74A2FE0D30048C
:10A2700003D73E02D602E1F5CDB5AA43630200123F8
:10A280000CD6DA42D7FE3028FAFE2E280123F12866
:10A290001A364523362BF29DA23306DEDA4062FC4A7
:10A2A000060A30FB0C63A23702377233060DEE216EA3
:10A2B00000CC923C5E57A1FADCEA3103B6DD21C98C
:10A2C0001B1104BFCD36A0FAD3A2E1C1CDDDA12E75
:10A2D0003625C9160BCDF9FC41FAA81C1FA83A3C5
:10A2E0000C55F789293F4F9A4CD01A5C6DCDA4B3C454
:10A2F0001BA5D3C45AA4D17BB720012B3DF4F9A40C
:10A30000E5216E0C460E203AA00C5FE62028077C5D
:10A31000B90E2A20014171CD47862810FE45280C30
:10A32000FE3028F2FE2C28EEFB2E20032B3630CBFA
:10A330006323032B3624CB53200227B0B1280270B4
:10A34000236300216D00C233AA80C9592C87FE207E
:10A3500028F4FE2A28F02BE5F5CD4786FE2D28FB07
:10A36000F82B28FAFE2428F0FE30201023CD478653
:10A37000300A2B012B77F128FB118CB128FD8126
:10A380003625C95F79B728013D33FA8EA3AFC5F59D
:10A390000FCB6A120FBC17B90C15F8278FAAAA39290
:10A3A00093F4F9A4C5CD01A51811CDF9A479CDS1B
:10A3B000A44FAF9293CDF9A4C5474PCDD6DA4C1B1C
:10A3C00020032A80C833DF4F9A450C300A3CDBFF9
:10A3D0009F37C41FA4B1C1F579B7F528013D804F2F
:10A3E0007AE604FE019F57814FD60BF5C5FCB6A156
:10A3F000FABDA3C1F1C5F5FAFBA3AFED44803C82B1

:10A40000470800CD6DA4F1F415A5CF120012BF18B
:10A41000380AC60B909205CD91A2EBD1C300A3D551
:10A42000AFF5CDA8A40115A2DD21F80211FDFCD45
:10A4300036A0F245A4F1CDAFA1F5C325A4F1CDB668
:10A44000A1F5CD48A4F1D1C9013AA5DD21B7431149
:10A450000F3CFCD36A0E1F23DA48905008362E22CE
:10A46000A80C2B48C90CD0362C230B03C20D5C58559
:10A470000CD809C3CCD75A0CDF59FB1C111C2A43E1D
:10A480000BCD5AA4C5F5E5D5CDD0A0E1062F047B7F
:10A49000965F237A95E723DD7D9EDD6F23DD7C9EB4
:10A4A000DD6723799E4F2B2B2B2B30E2D5A9D233A
:10A4B0000DF59FBBE17023F1C13D0C0D5AA477C6
:10A4C0001C900E40B540200CA9A3B0000B1F50533
:10A4D0000809698000040420F0000A08601000016
:10A4E0001027000000E8030000006400000000AD
:10A4F000000000000000000000067C83D36302318FE
:10A500007FB7823C473CD60330FCC6054F3AA00C88
:10A5100086400CF49C92004AC8CD5AA4363023D18A8
:10A52000F6CD4786301ECD858685CD9282D2E18775
:10A530006069232323CD4786D69CC2448747CDP04C
:10A5400090B11803C9295B8CD488637280CFE2C283A
:10A5500005F83BC26D81CD4786BB2A6600D11EB2F
:10A560008F5D5A6B0CA448723234E236669C377F1
:10A57000A5CDF9A6CD7284AF5F57CDF9A657E233B
:10A58000FE23CA8E5F5E27CA81A605CA6DA6FE2B5C
:10A590003E0828E62B7E23FE2ECAD6A5B8E20D2FE7C
:10A5A000242814FE2A20CA78FB0E22338037FE24C3
:10A5B0003E202007051CFEAFCE640231C82571C0E30
:10A5C000000528467E23FE22817FE2328F0FE2CA9
:10A5D0002019C6218E87FE233E2BC271A50E0193
:10A5E0002030C0528257E2328F6D554F5D5E28
:10A5F0002016BE201323BE200F23BE200B237D86A7
:10A60000043805D1471423CABBD120C85A201395
:10A610001D78B7280B7ED62D2806FEFE2005CDBA43
:10A620000CD205E1F42851C5D3CD9A8BD1C1C5E575
:10A630004378881FE19D244877AF800CD8E1CD31F0
:10A6400091E1CD488637280CFE3B2805FE2CC26DD3
:10A6500081CD4786C1BE1E5F5D57E9023234E23DE
:10A66000666916005F1978B7C277A51806CD9FA6F6
:10A67000CD7284B1F1C25EA5DCE8E893C6D6692E1A4
:10A680000C908011E4C05281C7E23FE45280CFE52D7
:10A690002808FE4C2804FE43200A5F0C0528057E8E
:10A6A00023BB287CFD9A6E1FE128CD5C50958BF3
:10A6B000D1C1C5E52A660C410B007BFBA5282BD58D
:10A6C0000C5CDDC92C1D17896477BFBA4C280E8FE5267
:10A6D0000281578CB389008CDF1A6084705CD349120
:10A6E0000C1CDF1A6C341A6AF18E8C789630EDA1F8F6
:10A6F0000EA0405C82D6648418F9F57AB73E2BC47214
:10A7000084F1C921CD9C6480000000000080CDDA86
:10A710009F2108A7CDECF9C1DDB1D1CDBFF9F28448B
:10A7200078B7CA0F9DD5DDE5C579F67FCD02A0F2D9
:10A7300044A7D5DD85C5CDA0C1DDBE1D1F5CD3670
:10A74000A0E17C1FE126A0CE122680C122660C88
:10A750000DC03A7CCD29F5DDE5C5CDB89DC1DDBE106
:10A76000D1C3F9E013881D213BA115C29C3D3F2F
:10A770000E3A6B0CFE88D2809FCD9A9FCDADA0C1F2
:10A78000DD1D1F5CD919C219EA7CDEAA7216B0CEF
:10A79000F1B7FA97A78601863F77D0C3809F0ACC8E
:10A7A000054556156ACF37A092276DF595E89300E3
:10A7B000071D0FCF7782174B12182C42E77825858B9
:10A7C000951D7A6DCB4658637C9FBEFF757ED213
:10A7D000F7172731800000000000081CDDA9F113B35
:10A7E0009E5D5CDD0A0CD3F9E81CDDA9F7E23CD63
:10A7F0000C9FFEF1C1DDB1D13DC8D5DBB5C5F5B554

```

:10A800000CD3F9E1CD05A0E5CD949CE118E5CDBFF
:10A810009FFA35A821870CDD8C9FC8013598DD2122
:10A820007A44110000CD3F9E012B868DD2146B11118
:10A830000000CD949CD02A07B594F36802BA6362C
:10A84000080CDF39C21870CC316A0219CA8CDB39CAE
:10A850000CDDA9F014983DD21DA0F1121A2CDF59FC9
:10A860000C1DDE1D1CD398CDDA9FCDDA0C1DDE18B
:10A87000D1CD919C21A2A8CDB89CCDBF9F37288D5
:10A88000A8CDB809CCDBF9F7F5F4D29F21A2A8CDB
:10A890000839CF1D4D29F21A2A8CDBA721A2DA0F01
:10A8A000498100000000007F0790BA34766A82E494
:10A8B000E9E74BF184B14F7F3B288631B664699953
:10A8C000087E436E33523872431E75DA58621A2DAC4
:10A8D0000F4983CDDA9FCD50A8C1DDE1D1CDD89FFB
:10A8E000EBCDF59PCDA1A8C3BF9ECDBF9FFC03A76C
:10A8F000CDF29F3A6B0CF813810010081DD2100F3
:10A90000005159CCD39E21889CE5211A49CDDBA718
:10A91000219CA8C90D1407BAFE6275116CB8D66F
:10A92000784CBD7DD13E7A01CB23C4D77BDC3A0A7B
:10A9300017347C36C1A381F77CEB1661A8197D5DBF
:10A94000788F60B97DA2441272637D1662FBA792D4
:10A9500078C0F0BFC0C47E7B8BA1AAA7FF6FFFFF7
:10A96000FF7F80454E5C446F2D4B458DA44415493
:10A970000C1E45E5D44449CD524541D24645D4AB
:10A98000474F2054CF464B45EBC4A9C652455354B6
:10A990004F52C5474F205355C2524545552CE527F
:10AA0000C5D53544FD04F5D4FCE4B554CC0C5728
:10AA10004149D4445C6504F4BC55052494ED4BF6F
:10AA20004C49535445CE434C4541D24645E254D2
:10AA3000552CE534156C5A15553494EC7544142D5
:10AA4000A854CF46C8E535043A8544845CBA4F4D4DA
:10AA5000535445D0ABADAAAFDE414BC44FD2B8ED1D
:10AA6000BC5347CE8494ED44142D3553D24652C58A
:10AA7000494ED0504FD3351D2524CE44CF745D0C
:10AA800058D0434FD335349CE5441CBA154CE8504D4
:10AA900045CB4C45CE535452A45641CC4153C3A30D
:10AAB0004852A44C454654A45249474854A44D4941
:10AAC00044A4C504FD3494E5354D2454535C54CB
:10AAD0005052494ED45424143C54C54524143C5AF
:10AAE00052414E444F4D495AC55357495443C84C0F
:10AAB00057494454C84C4E554CC57494454C84C73
:10AAB0005641D24C4C5641D253504541CBA750520F
:10AAA00045434953494FCB43414CC0CB494CC0458F
:10AAB0005B4348414E47C54C494BC54C4F41444709
:10AAC000CF5255CB4C4F41C4CAE45D7415554CF433C
:10AAD0004F50D9414C4F4144C3414D45524745C366
:10AAB000414C4F41C4414D455247C541534156C564
:10AAE0004C4953D44C4C4953D452454E554D424584
:10AB0000D244454C4554C5454449D4434F4ED400E6
:10AB10007C534E45584520572F4F20464FD25359FF
:10AB20004E544158204552524FD252455455524E80
:10AB300020572F4F20474F5355C24F5554204F4653
:10AB400020444154C1494C4C4547414C2046554E4E
:10AB50004354494FCB41524954484D4554494320E8
:10AB60004F564552464C4FD74F5554204F46204DD7
:10AB7000454D4F52D9554B44546494E45442053C4
:10AB8000544154454D454E54A053545253435249A8
:10AB90005054204F5554204F462052414E47C552E5
:10ABA000452D44494D454E53494F4E454420415251
:10ABB0005241D943414E2754202FB0494C4C454770
:10ABC000414C204449524543D454595045204D49A5
:10ABD000532D4D415443C84E4F20535452494E4774
:10ABE0002053504143C5535452494E4720544F4F70
:10ABF000204CF48C7544F4F20434F4D504C45D8DB

```

```

:10AC0000043414E275420434F4E54494E55C5554E4F
:10AC1000444546494E454420555345522043414CF6
:10AC2000CC46494C45204E4F5420464F554E4C449C2
:10AC30004C4C4547414C20454F6G46494C45532056
:10AC4000444946464552454ED45245434F564552D7
:10AC500045C4464E52455455524E20572F4F20461C
:10AC6000554E4354494F4E2043414CC0CD495353CC
:10AC7000494E472053544154454D454E454204E5558
:10AC80004D4245D22A494E56414C4C4420494E5086
:10AC900055548A2040204C494E45A00A5245414413
:10ACA000593A8A2A4558545241204C4F53548A0AE3
:10ACB0002A42524541CB00AF21000C3F772C20FCAB
:10ACC000243BF82BF9225A0C21FF7F22540C3E2C79
:10ACD00032B00C3BC3328D0C32A40C21B7AC22A58D
:10ACE0000CFD21900C210F80228E0C3E3F32910CR6
:10ACF00032960C3E3832920C32970C21080C220608
:10AD000000C21008022870C22890C228B0C2192AD11
:10AD1000CDBD84CDBE283CD47862005CD218018149A
:10AD200021B10CCD4886CDD8A07E87C26D81CD2786
:10AD300087EB2B2B2040C22480CB5110003B7ED06
:10AD400052E1DA4D81119CFP191100037C9220021F
:10AD50007D9311011030031100037C9220027D933A
:10AD600038E0F9225A0CBE225C0CCD3E81B7EBEDBA
:10AD70005201F0F09CDEB89CD09A121A1ADCBBD14
:10AD80008421BD8422A50C3B00320010CDBA82C3C4
:10AD9000A0810A4849478455354204D454D4F52DC
:10ADA000D92042595445532046524A0A0A8A2023
:10ADB00002020202020202020202020202020202093
:10ADC00002020202020202020202020202020202083
:10ADD00002020202020202020202020202020202073
:10ADE00002020202020202020202020202020202063
:10ADF0000202020208A000000004F4FBE79C1C36198B7
:10AEB00002100103E80BC2809237E2F77BE2F772893
:10AEB1000F22B7D44C9FFFFFFFFFFF77777777777796

```

5. Anhang

5.2.6. Programm Editor Assemblerausdruck

```

EBAS-88
LOC   CODE   LINE   SOURCE
                                00001      PN      ED
                                00002
                                00003
                                00004
                                00005      TITL 'EDITOR Y23VO'
                                00006
                                00007
                                00008 ;VEREINBARUNGEN
                                00009 CR: EQU 0DH ;WAGENRUECKLAUF
                                00010 LP: EQU 0AH ;ZEILENVORSCHUB
                                00011 ETX: EQU 3 ;END OF TEXT
                                00012 CTRLD: EQU 4 ;ZEILE DOPPELN
                                00013 DEL: EQU 7FH ;ZEILE LOESCHEN
                                00014 CAN: EQU 18H ;ZEICHEN LOESCHEN
                                00015 INS: EQU 13H ;ZEICHEN EINFUEGEN
                                00016 NUL: EQU OFFH ;TABELLENENDE
                                00017 NL: EQU 1EH ;NEUE ZEILE
                                00018 SUD: EQU 1AH ;ZEILE EINFUEGEN
                                00019 RAME: EQU 7FFFH ;STANDARDZUWEISUNG
                                00020 RAMA: EQU 4000H
                                00021 SRAM: EQU 0D39H
                                00022 CIO: EQU 13CH
                                00023 RI: EQU 0E3H
                                00024 CO: EQU 0E6H
                                00025 WO: EQU 0E9H
                                00026 LO: EQU 0ECH
                                00027 ERROR: EQU 0038H
                                00028 CURS: EQU OFF1H
                                00029 TI: EQU 10CH
                                00030 PAO: EQU 4DCH
                                00031 PCGA: EQU 7AH
                                00032 PCHK: EQU 518H
                                00033 STR: EQU 128H
                                00034 INK: EQU 36BH
                                00035 LKOM: EQU 352H
                                00036 OUTK: EQU 38CH
                                00037 BRAM: EQU OFCOOH ;BILDSCHIRM-RAM
                                00038
                                00039 ORG 0C000H
C000 F5 00040 PUSH AF
C001 C5 00041 PUSH BC
C002 D5 00042 PUSH DE
C003 E5 00043 PUSH HL
C004 2157C1 00044 LD HL,ATEXT
C007 CD2801 00045 CALL STR
C00A CD0C01 00046 CALL TI ;TI
C00D FE59 00047 CMP 'Y'
C00F 2011 00048 JRNZ RESTR-#
                                00049 ;UEBERTRAGEN SYSTEMABELLEN
C011 2154C3 00050 LD HL,STAB ;QUELLE
C014 11390D 00051 LD DE,RAMEZ ;ZIEL
C017 010600 00052 LD BC,LSTAB
C01A EDB0 00053 LDIR
C01C 2A390D 00054 LD HL,(RAMEZ) ;ENDE SETZEN
C01F 3E03 00055 LD A,ETX
C021 77 00056 LD (HL),A
                                00057 ;RUECKSETZEN ZEICHENZAehler
C022 2A390D 00058 RESTR: LD HL,(RAMEZ)
C025 223D0D 00059 LD (ZEZL1),HL
                                00060 START: EQU #
                                00061 ;STARTADRESSE RETTEN
C028 2128C0 00062 LD HL,START
C02B E5 00063 PUSH HL
C02C 2190C1 00064 LD HL,PIONT
C02F CD2801 00065 CALL STR
                                00066 ;EINGABE KOMMANDO
C032 213FC0 00067 LD HL,JPTBL
C035 CD0C01 00068 CALL TI ;TI
C038 CD7A00 00069 CALL PCGA
C03B DA3AC3 00070 JPC E1
C03E E9 00071 JMP (HL)
                                00072
                                00073

```



```

00074
00075 JPTBL: EQU #
C03F 00076 DB 'A'
C040 6DC0 00077 DA ASGN
C042 00078 DB 'D'
C043 12C1 00079 DA DISPL
C045 00080 DB 'E'
C046 ADC1 00081 DA EDIT
C048 00082 DB 'G'
C049 6CC2 00083 DA GO
C04B 00084 DB 'I'
C04C 6B03 00085 DA INK
C04E 00086 DB 'L'
C04F 5203 00087 DA LKOM
C051 00088 DB 'O'
C052 8C03 00089 DA OUTK
C054 00090 DB 'P'
C055 94C0 00091 DA PRINT
C057 00092 DB 'Q'
C058 67C0 00093 DA ENDE
C05A 00094 DB 'R'
C05B B2C0 00095 DA READ
C05D 00096 DB 'U'
C05E 87C2 00097 DA UP
C060 00098 DB 'W'
C061 DBC0 00099 DA WRITE
C063 00100 DB 'T'
C064 96C1 00101 DA TOP
C066 FF 00102 DB NUL
00103
00104
00105
00106 ENDE: EQU #
C067 E1 00107 POP HL ;STARTADR. KOMMANDO
C068 E1 00108 POP HL
C069 D1 00109 POP DE
C06A C1 00110 POP BC
C06B F1 00111 POP AF
C06C C9 00112 RET
00113
00114 ;KOMMANDO SPEICHERZUWEISUNG
00115 ASGN: EQU #
C06D 216AC1 00116 LD HL,BOM
C070 CD2801 00117 CALL STR
C073 CD1805 00118 CALL PCHK
C076 C8 00119 RZ
C077 CDDC04 00120 CALL PAO
C07A 22390D 00121 LD (RAMEZ),HL
C07D 223D0D 00122 LD (ZEZL1),HL
C080 3E03 00123 LD A,ETX ;ENDE SETZEN
C082 77 00124 LD (HL),A
C083 217DC1 00125 LD HL,SOM
C086 CD2801 00126 CALL STR
C089 CD1805 00127 CALL PCHK
C08C C8 00128 RZ
C08D CDDC04 00129 CALL PAO
C090 223D0D 00130 LD (RAMAZ),HJ.
C093 C9 00131 RET
00132
00133 ;AUSGABE ALLER ZEILEN AUF KANAL LO
00134 PRINT: EQU #
C094 CDBEC2 00135 CALL CRLF
C097 2A390D 00136 LD HL,(RAMEZ),
C09A 7E 00137 LD A,(HL)
C09B FE03 00138 CHP ETX
C09D C8 00139 RZ ;END OF FILE
C09E FE1E 00140 CMP NL ;NEW LINE
C0A0 CCAACO 00141 CAZ P3 ;UHWANDELN
C0A3 2B 00142 DEC HL
C0A4 4F 00143 LD C,A
C0A5 CDECO0 00144 CALL LO
C0A8 18F0 00145 JR P1<#
C0AA OE0D 00146 LD C,CR
C0AC CDECO0 00147 CALL LO
C0AF 3E0A 00148 LD A,LF
C0B1 C9 00149 RET
00150
00151
00152 ;LESEN BIS ENDE VON KANAL RI
00153
00154 READ: EQU #
C0B2 2A3D0D 00155 LD HL,(ZEZL1)

```

5. Anhang

```

COB5 23      00156      INC HL
COB6 2B      00157      R1: DEC HL
COB7 CD9DC1  00158      CALL KOPU
COBA CDE300  00159      R2: CALL RI
COBD B67F    00160      AND 7FH
COBF FE0D    00161      CMP CR
COC1 280D    00162      JRZ LZNL-#
COC3 FE0A    00163      CMP LF
COC5 2809    00164      JRZ LZNL-#
COC7 77      00165      R3: LD (HL),A
COC8 FE03    00166      CMP BTX ;TEXTENDE
COCA 20BA    00167      JRNZ R1-#
COCB 223D0D  00168      LD (ZEZL1),HL
COCF C9      00169      RET
              00170
              00171
              00172
              00173 ;PRUEFEN LETZTES ZEICHEN NL=?
              00174 ;J:CR,LF UEBERLESEN
              00175 ;N:CR,LF UMWANDELN
              00176 LZNL: INC HL
COD0 23      00177      LD A,(HL)
COD1 7E      00178      DEC HL
COD2 2B      00179      CMP 1EH
COD3 FE1E    00180      JRZ R2-#
COD5 28E3    00181      LD A,1EH ;CR O. LF ERSETZEN
COD7 3E1E    00182      JR R3-#
COD9 18EC    00183
              00184
              00185
              00186 ;AUSGABE EINER ANZAHL ZEILEN AUF KANAL WO
              00187 ;MIT AUSGABE ENDE-ZEICHEN
              00188 WRITE: EQU #
              00189 CALL PCHK
              00190 JRC W1-# ;ALLE
              00191 CALL PAO
              00192 W2: EX DE,HL
              00193 W3: CALL W1Z
              00194 CMP BTX ;TEXTENDE
              00195 RZ
              00196 DEC DE
              00197 LD A,B
              00198 OR D
              00199 JRNZ W3-#
              00200 LD C,BTX ;ENDE SETZEN
              00201 CALL W0
              00202 RET
              00203 W1: LD HL,OFFFPH
              00204 JR W2-#
              00205
              00206
              00207 ;AUSGABE EINER ZEILE AUF KANAL WO
              00208 ;IN: (ZEZL1)
              00209 ;OUT: (ZEZL1)-ZEILENLAENGE
              00210
              00211 W1Z: EQU #
              00212 LD HL,(ZEZL1)
              00213 INC HL
              00214 W1Z1: DEC HL
              00215 LD A,(HL)
              00216 PUSH AP
              00217 LD C,A
              00218 CALL W0
              00219 POP AP
              00220 CMP BTX ;TEXTENDE
              00221 RZ
              00222 CMP NL ;ZEILENENDE
              00223 JRNZ W1Z1-#
              00224 DEC HL
              00225 LD (ZEZL1),HL
              00226 RET
              00227
              00228 ;AUSGABE EINER ANZAHL ZEILEN AUF BILDSCHIRM
              00229 ;MIT EINSTELLUNG (ZEZL1)
              00230
              00231 DISPL: EQU #
              00232 D4: CALL PCHK
              00233 RC ;ENDE BEI CR
              00234 JRZ DSPL1-# ;EINZELN BEI LEEERZEICHEN
              00235 CALL PAO ;ANZAHL IN HL
              00236 DSPL2: CALL DISPL1
              00237 CMP BTX ;ENDE ?

```

```

C120 C8      00238      RZ
C121 2B      00239      DEC HL
C122 7D      00240      LD A,L
C123 B4      00241      OR H
C124 20F5    00242      JRNZ DSPL2-#
C126 C9      00243      RET
              00244
              00245 ;AUSGABE ZEILENWEISE BEI LEEERZEICHEN
C127 CD2FC1  00246      DSPL1: CALL DISP1
C12A FE03    00247      CMP BTX
C12C 20E4    00248      JRNZ D4-#
C12E C9      00249      RET
              00250
              00251
              00252
              00253 ;DISPLAY FUER EINE ZEILE
              00254 ;IN: (ZEZL1)-AKT. ZEICHEN
              00255 ;OUT: (ZEZL1+1ZEILE),A=1EH ODER 4
              00256 DISP1: EQU #
C12F C5      00257      PUSH BC
C130 E5      00258      PUSH HL
C131 CDBEC2  00259      CALL CRLP
C134 21COFF  00260      LD HL,BRAM+960 ;UNTERSTE BILDSCH.-ZEILE
C137 22F10F  00261      LD (CURS),HL
C13A 2A3D0D  00262      LD HL,(ZEZL1)
C13D 7E      00263      D1: LD A,(HL)
C13E FE03    00264      CMP BTX
C140 2812    00265      JRZ D2-# ;END OF FILE
C142 2B      00266      DEC HL
C143 CD9DC1  00267      CALL KOPU
C146 FE1E    00268      CMP NL
C148 F5      00269      PUSH AF
C149 4F      00270      LD C,A
C14A C4E600  00271      CANZ CO
C14D 79      00272      LD A,C
C14E F1      00273      POP AF
C14F 20EC    00274      JRNZ D1-#
C151 223D0D  00275      LD (ZEZL1),HL
C154 E1      00276      D2: POP HL
C155 C1      00277      POP BC
C156 C9      00278      RET
              00279
              00280
              00281 ;TEXTE
C157 OD      00282      ATEXT: DB CR
C158 OA      00283      DB LF
C159          00284      DB 'TE 01'
C15E OD      00285      DB CR
C15F OA      00286      DB LF
C160          00287      DB 'NEW? (N)'
C169 00      00288      DB 0
C16A OD      00289      BOM: DB CR
C16B OA      00290      DB LF
C16C          00291      DB 'END OF MEMORY?'
C17C 00      00292      DB 0
C17D OD      00293      SOM: DB CR
C17E OA      00294      DB LF
C17F          00295      DB 'START OF MEMORY?'
C18F 00      00296      DB 0
C190 OD      00297      PIONT: DB CR
C191 OA      00298      DB LF
C192          00299      DB '33 '
C195 00      00300      DB 0
              00301
              00302
              00303
              00304 ;ZEICHENZAehler RUECKSTELLEN
              00305 ;IN: (ZEZL1)-AKT.
              00306 ;OUT: (ZEZL1)-(RAMEZ)
              00307 TOP: EQU #
C196 2A390D  00308      LD HL,(RAMEZ)
C199 223D0D  00309      LD (ZEZL1),HL
C19C C9      00310      RET
              00311
              00312
              00313
              00314 ;KONTROLLRECHNUNG PUPPER
              00315 ;IN: 'HL'-ZU PRUEFENDE ZELLE
              00316 ; 'HL'-'RAMEZ'='0 SONST FEHLER
              00317 ;KILL: AF
              00318 KOPU: EQU #
C19D E5      00319      PUSH HL

```

5. Anhang

```

C19E D5      00320      PUSH DE
C19F EB      00321      EX DE,HL
C1A0 2A3B0D  00322      LD HL,(RAMAZ)
C1A3 EB      00323      EX DE,HL
C1A4 B7      00324      OR A
C1A5 ED52    00325      SBC HL,DE
C1A7 FC41C3  00326      CAM E2      ;PUFFER VOLL
C1AA D1      00327      POP DE
C1AB E1      00328      POP HL
C1AC C9      00329      RET
              00330
              00331
              00332
              00333 ;EDIT-KOMMANDO
              00334 EDIT: EQU #
              00335 ;AKTUELLE ZEILE IN ANZEIGE
C1AD CD2FC1  00336 ED1: CALL DISP1
C1B0 FE03    00337      CMP ETX
C1B2 C4F8C2  00338      CANZ ZLOE
C1B5 CD3C01  00339 ED2: CALL CIO
C1B8 FE0D    00340      CMP CR
C1BA 2825    00341      JRZ ED4-#
C1BC FE1A    00342      CMP SUB
C1BE 282B    00343      JRZ ED5-#
C1C0 FE0A    00344      CMP LF
C1C2 2818    00345      JRZ ED3-#      ;NAECHSTE
C1C4 FE7F    00346      CMP DEL
C1C6 CCC8C2  00347      CAZ PULOE      ;ANZEIGE LOESCHEN
C1C9 FE04    00348      CNP CTRLD      ;ZEILE DOPPELN
C1CB CC1FC2  00349      CAZ DUPL
C1CE FE18    00350      CMP CAN
C1D0 281E    00351      JRZ CHDEL-#
C1D2 FE13    00352      CNP INS
C1D4 282D    00353      JRZ CHINS-#
C1D6 4F      00354      LD C,A
C1D7 CDE600  00355      CALL CO
C1DA 18D9    00356      JR ED2-#
              00357
              00358
              00359 ;AUSWERTUNG LF
C1DC CDE1C1  00360 ED3: EQU #      ;NEUE ZEILE
C1DF 18CC    00361      CALL ED4
              00362      JR ED1-#
              00363
              00364 ;AUSWERTUNG CR,ENDE EDITKOMMANDO
C1E1 CD37C2  00365 ED4: CALL LANG
C1E4 CD42C2  00366      CALL MOVE
C1E7 CDDBC2  00367      CALL APZP
C1EA C9      00368      RET
              00369
              00370 ;AUSWERTUNG CONTROL Z-SUB
              00371 ;ZEILEN EINFUEGEN
C1EB CDE1C1  00372 ED5: EQU #
C1EE 18C5    00373      CALL ED4
              00374      JR ED2-#
              00375
              00376
              00377 ;AUSWERTUNG CONTROL X-CANCEL
              00378 ;ZEICHEN AUF BILDSCHIRM LOESCHEN
              00379 CHDEL: EQU #
C1F0 CD37C2  00380      CALL LANG
C1F3 CD30C3  00381      CALL REST
C1F6 2AF10F  00382      LD HL,(CURS)
C1F9 E5      00383      PUSH HL
C1FA D1      00384      POP DE
C1FB 23      00385      INC HL
C1FC EDB0    00386      LDIR
C1FE 3E20    00387      LD A,' '      ;ENDE LOESCHEN
C200 77      00388      LD (HL),A
C201 18B2    00389      JR ED2-#
              00390
              00391 ;AUSWERTUNG CONTROL S
              00392 ;ZEICHEN AUF BILDSCHIRM EINFUEGEN
              00393 CHINS: EQU #
C203 2AF10F  00394      LD HL,(CURS)
C206 E5      00395      PUSH HL
C207 7E      00396      LD A,(HL)
C208 B67F    00397      AND 7FH      ;CURSOR LOESCHEN
C20A 77      00398      LD (HL),A
C20B CD37C2  00399      CALL LANG
C20E CD30C3  00400      CALL REST
C211 21FEFF  00401      LD HL,BRAM+1022

```

```

C214 11FFFF 00402 LD DE,BRAM+1023
C217 EDB8 00403 LDDR
C219 E1 00404 POP HL
C21A 3EAO 00405 LD A,0A0H ;CURSOR SETZEN
C21C 77 00406 LD (HL),A
C21D 1896 00407 JR ED2-#
00408
00409
00410
00411 ;AUSWERTUNG CONTROL D
00412 ;ZEILE DOPPELN
00413 DUPL: EQU #
C21F C5 00414 PUSH BC
C220 D5 00415 PUSH DE
C221 E5 00416 PUSH HL
C222 2180FF 00417 LD HL,BRAM+380H ;VORLETZTE ZEILE
C225 11C0FF 00418 LD DE,BRAM+3C0H ;LETZTE ZEILE
C228 013E00 00419 LD BC,3EH ;LAENGE
C22B EDB0 00420 LDIR
C22D ED53F10F 00421 LD (CURS),DE
C231 3E20 00422 LD A,020H ;LEERZ.
C233 E1 00423 POP HL
C234 D1 00424 POP DE
C235 C1 00425 POP BC
C236 C9 00426 RET
00427
00428
00429
00430 ;ERMITTLUNG DER LAENGE ANZEIGEPUFFER
00431 ;IN: (CURS)=AKT.ZEILE
00432 ;OUT: A=LAENGE
00433 LANG: PUSH HL
C237 E5 00434 LD HL,(CURS)
C238 2AF10F 00435 LD A,L
C23B 7D 00436 LD HL,BRAM+960 ;ANFANG LETZTE BILDSCH.-ZEILE
C23C 21C0FF 00437 SUB L
C23F 95 00438 POP HL
C240 E1 00439 RET
C241 C9 00440
00441
00442 ;VERSCHIEBUNG PUFFERINHALT NACH UNTEN UM 1 ZEILE
00443 ;IN: A ZEILENLAENGE
00444 ;'BC' = ANZAHL BIS ENDE
00445 ;'DE' = ZIELADR.=ZEZL1-ANZAHL-ZEILENLAENGE
00446 ;'HL' = QUELLADRESSE=ZEZL1-ANZAHL
00447 MOVE: EQU #
C242 B7 00448 OR A ;SONDERFALL A=0
C243 C8 00449 RZ
C244 E5 00450 PUSH HL
C245 D5 00451 PUSH DE
C246 C5 00452 PUSH BC
C247 F5 00453 PUSH AF
C248 2A3D0D 00454 LD HL,(ZEZL1)
C24B 010000 00455 LD BC,0
C24E 7E 00456 MOV1: LD A,(HL)
C24F 2B 00457 DEC HL
C250 FE03 00458 CMP ETX
C252 03 00459 INC BC
C253 20F9 00460 JRNZ MOV1-#
C255 23 00461 INC HL
C256 E5 00462 PUSH HL ;'HL'=TEXTENDE
C257 D1 00463 POP DE
00464 ;BERECHNUNG ZIELADRESSE
C258 F1 00465 POP AF
C259 C5 00466 PUSH BC
C25A 0600 00467 LD B,0
C25C 4F 00468 LD C,A
C25D 0C 00469 INC C ;PLATZ F.1EH
C25E B7 00470 OR A
C25F ED42 00471 SBC HL,BC
C261 CD9DC1 00472 CALL KOPU
C264 C1 00473 POP BC
C265 EB 00474 EX DE,HL ;ZIEL/QUELLE
C266 EDB0 00475 LDIR
C268 C1 00476 POP BC
C269 D1 00477 POP DE
C26A E1 00478 POP HL
C26B C9 00479 RET
00480
00481
00482
00483

```

5. Anhang

```

00484
00485
00486 ;GO ZEILENNUMMER
00487 GO: EQU #
C26C CD1805 00488 CALL PCHK
C26F 382C 00489 JRC G8-# ;ALLE
C271 C8 00490 RZ
C272 CDDC04 00491 CALL PAO ;ANZAHL IN HL
C275 ED5B3D0D 00492 G7: LD DE,(ZEZL1)
C279 EB 00493 EX DE,HL
C27A CDAEC2 00494 G1: CALL G1Z ;GO 1 ZEILE
C27D D8 00495 RC ;ENDE
C27E 1B 00496 DEC DE
C27F 7B 00497 LD A,E
C280 B2 00498 OR D
C281 20F7 00499 JRNZ G1-#
C283 223D0D 00500 G5: LD (ZEZL1),HL
C286 C9 00501 RET
00502
00503 UP: EQU #
C287 CD1805 00504 CALL PCHK
C28A C8 00505 RZ
C28B CDDC04 00506 CALL PAO
C28E ED5B3D0D 00507 LD DE,(ZEZL1)
C292 CDA2C2 00508 G4: CALL G1ZZ
C295 2B 00509 DEC HL
C296 7D 00510 LD A,L
C297 B4 00511 OR H
C298 20F8 00512 JRNZ G4-#
C29A EB 00513 EX DE,HL
C29B 18E6 00514 JR G5-#
00515
C29D 21FFFF 00516 G8: LD HL,OFFFHH ;ALLE
C2A0 18D3 00517 JR G7-#
00518
00519 ;GO 1 ZEILE ZURUECK
00520 ;IN: HL= AKT. PUFFERZELLE
00521 ;OUT: A=1EH AM PUFFERENDE
C2A2 13 00522 G1ZZ: INC DE
C2A3 13 00523 G6: INC DE
C2A4 1A 00524 LD A,(DE)
C2A5 CD1CC3 00525 CALL TEPUE
C2A8 FE1E 00526 CMP NL
C2AA 20F7 00527 JRNZ G6-#
C2AC 1B 00528 DEC DE
C2AD C9 00529 RET
00530
00531 ;GO 1 ZEILE
00532 ;OUT: CY=1 BEI EOF
C2AE 7E 00533 G1Z: LD A,(HL)
C2AF FE03 00535 CMP BTX
C2B1 2809 00536 JRZ G1Z1-#
C2B3 2B 00537 DEC HL
C2B4 CD9DC1 00538 CALL KOPU
C2B7 FE1E 00539 CMP NL
C2B9 20F3 00540 JRNZ G1Z-#
C2BB C9 00541 RET
00542
C2BC 37 00543 G1Z1: SCF
C2BD C9 00544 RET
00545
00546
C2BE F5 00547 CRLF: PUSH AF ;AUSGABE NEUE ZEILE BILDSCHIRM
C2BF C5 00548 PUSH BC
C2C0 0E1E 00549 LD C,NL
C2C2 CDE600 00550 CALL CO
C2C5 C1 00551 POP BC
C2C6 F1 00552 POP AF
C2C7 C9 00553 RET
00554
00555
00556
00557 ;ANZEIGE LOESCHEN
00558 PULOE: EQU #
C2C8 C5 00559 PUSH BC
C2C9 E5 00560 PUSH HL
C2CA 0641 00561 LD B,41H
C2CC 3E20 00562 LD A,1
C2CE 21BFFF 00563 LD HL,BRAM+959 ;LETZTE BILDZEILE-1ZEICH.
C2D1 22F10F 00564 LD (CURS),HL
C2D4 77 00565 PUL1: LD (HL),A

```

```

C2D5 23      00566      INC HL
C2D6 10FC    00567      DJNZ PUL1-#
C2D8 E1      00568      POP HL
C2D9 C1      00569      POP BC
C2DA C9      00570      RET
                00571
                00572
                00573      ;BILDSCH.IN ZEICHENPUFFER UEBERTRAGEN
                00574      ;IN: A=LAENGE
                00575      APZF: EQU #
C2DB B7      00576      OR A
C2DC C8      00577      RZ
C2DD C5      00578      PUSH BC
C2DE 47      00579      LD B,A
C2DF 11C0FF  00580      LD DE,BRAM+960      ;LETZTE BILDSCH.-ZEILE
C2E2 2A3D0D  00581      LD HL,(ZEZL1)      ;ZIEL
C2E5 1A      00582      APZF1: LD A,(DE)
C2E6 E67F    00583      AND 7FH      ;CURSOR AUSBLENDEN
C2E8 77      00584      LD (HL),A
C2E9 13      00585      INC DE
C2EA 2B      00586      DEC HL
C2EB 10F8    00587      DJNZ APZF1-#
C2ED 361E    00588      LD (HL),NL
C2EF 2B      00589      DEC HL
C2F0 223D0D  00590      LD (ZEZL1),HL
C2F3 CDBEC2  00591      CALL CRLF
C2F6 C1      00592      POP BC
C2F7 C9      00593      RET
                00594
                00595      ;1 ZEILE IM PUFFER LOESCHEN
                00596      ;IN: 'ZEZL1'-ANFANG NAECHSTE ZEILE
                00597      ;NAECHSTER SPEICHERPLATZ 1EH
                00598      ZLOE: EQU #
C2F8 F5      00599      PUSH AF
C2F9 C5      00600      PUSH BC
C2FA D5      00601      PUSH DE
C2FB E5      00602      PUSH HL
C2FC 2A3D0D  00603      LD HL,(ZEZL1)
C2FF E5      00604      PUSH HL
                00605      ;BERECHNUNG ANZAHL,-' BC
C300 010000  00606      LD BC,0
C303 3E03    00607      LD A,ETX
C305 56      00608      ZL1: LD D,(HL)
C306 03      00609      INC BC
C307 2B      00610      DEC HL
C308 BA      00611      CHP D
C309 20FA    00612      JRNZ ZL1-#
C30B E1      00613      POP HL
C30C E5      00614      PUSH HL
C30D D1      00615      POP DE      ;'HL'='ZEZL1'
                00616      ;EINSTELLUNG DE AUF ZIELADRESSE
C30E CDA2C2  00617      CALL G1ZZ
C311 ED533D0D 00618      LD (ZEZL1),DE
C315 EDB8    00619      LDDR
C317 E1      00620      POP HL
C318 D1      00621      POP DE
C319 C1      00622      POP BC
C31A F1      00623      POP AF
C31B C9      00624      RET
                00625
                00626      ;TEST PUFFERENDE
C31C E5      00627      TEPUE: PUSH HL
C31D 2A390D  00628      LD HL,(RAMEZ)
C320 B7      00629      OR A
C321 ED52    00630      SBC HL,DE
C323 FC28C3  00631      CAM TEP1
C326 E1      00632      POP HL
C327 C9      00633      RET
                00634
                00635      TEP1: EQU #
C328 ED5B390D 00636      LD DE,(RAMEZ)
C32C 13      00637      INC DE
C32D 3E1E    00638      LD A,NL      ;PUFFERENDE
C32F C9      00639      RET
                00640
                00641
                00642      REST: EQU #
                00643      ;IN: A - LAENGE
                00644      ;OUT: B=0,C=ANZAHL BIS ZEILENENDE
                00645
C330 F5      00646      PUSH AF
C331 47      00647      LD B,A

```

5. Anhang

```

C332 3E3F      00648      LD  A,3FH
C334 90        00649      SUB B
C335 0600      00650      LD  B,0
C337 4F        00651      LD  C,A
C338 F1        00652      POP AF
C339 C9        00653      RET
              00654
              00655
              00656      ;FEHLERMELDUNGEN
              00657
C33A 214531    00658      E1: LD  HL,3145H      ;KOMMANDOFEHLER
C33D 22FEFF    00659      ERA: LD  (BRAM+1022),HL ;PLATZ FEHLERANZEIGE
C340 C9        00660      RET
              00661
C341 214532    00662      E2: LD  HL,3245H      ;PUFFER VOLL
C344 CD3DC3    00663      CALL ERA
C347 2A3B0D    00664      LD  HL,(RAMAZ)
C34A 3603      00665      LD  (HL),ETX
C34C C33800    00666      JMP  ERROR
              00667
C34F 214533    00668      E3: LD  HL,3345H      ;ADRESSE O.PARAMETER FALSCH
C352 18E9      00669      JR  ERA-#
              00670
              00671
              00672      ;SYSTEMTABELLE
              00673
              00674      STAB: EQU #
C354 FF7F      00675      DA  RAME      ;RAHENDE
C356 0040      00676      DA  RAMA      ;RAMANFANG
C358 FF7F      00677      DA  RAME      ;RAME-ZUWEISUNG
              00678      LSTAB: EQU #-STAB
              00679
              00680
              00681
              00682
              00683      ORG  SRAM
              00684
OD39          00685      RAMEZ: BER 2      ;RAHENDEZELE
OD3B          00686      RAMAZ: BER 2      ;RAMANFANG
OD3D          00687      ZE2L1: BER 2      ; RAME ;ZEICHENZAehler1
OD3F          00688      END

```

EBAS-88		CROSS-REFERENCE		PN: ED				
CR	000D	COAA	COBF	C157	C15E	C16A	C17D	C190
		C1B8						
LF	000A	COAF	COC3	C158	C15F	C16B	C17E	C191
		C1C0						
ETX	0003	CO1F	C080	C09B	C0C8	C0E7	C0EF	C106
		C11E	C12A	C13E	C1B0	C250	C2AF	C303
		C34A						
CTRLD	0004	C1C9						
DEL	007F	C1C4						
CAN	0018	C1CE						
INS	0013	C1D2						
NUL	00FF	C066						
NL	001E	C09E	C109	C146	C2A8	C2B7	C2C0	C2ED
		C32D						
SUB	001A	C1BC						
RAME	7FFF	C354	C358					
RAMA	4000	C356						
SRAM	0D39	C35A						
CIO	013C	C1B5						
RI	00E3	C0BA						
CO	00E6	C14A	C1D7	C2C2				
WO	00E9	C0F1	C102					
LO	00EC	C0A5	C0AC					
ERROR	003E	C34C						
CURS	0FF1	C137	C1F6	C203	C22D	C238	C2D1	
TI	010C	C00A	C035					
PAO	04DC	C077	C08D	C0E0	C118	C272	C28B	
FCGA	007A	C038						
PCHK	0518	C073	C089	C0DB	C112	C26C	C287	
STR	0128	C007	C02F	C070	C086			
INK	036B	C04C						
LKOH	0352	C04F						
OUTK	038C	C052						
BRAM	FC00	C134	C211	C214	C222	C225	C23C	C2CE
		C2DF	C33D					
RESTR	C022	C00F						

START	C028	C028						
JPTBL	C03F	C032						
BNDE	C067	C058						
ASGN	C06D	C040						
PRINT	C094	C055						
P1	C09A	C0A8						
P3	C0AA	C0A0						
READ	COB2	C05B						
R1	COB6	COCA						
R2	COBA	C0D5						
R3	C0C7	C0D9						
LZNL	C0D0	C0C1	C0C5					
WRITE	C0DB	C061						
W2	C0E3	C0F8						
W3	C0E4	C0ED						
W1	C0F5	C0DE						
W1Z	COFA	C0E4						
W1Z1	C0FE	C10B						
DISPL	C112	C043						
D4	C112	C12C						
DSPL2	C11B	C124						
DSPL1	C127	C116						
DISP1	C12F	C11B	C127	C1AD				
D1	C13D	C14F						
D2	C154	C140						
ATEXT	C157	C004						
EOM	C16A	C06D						
SOM	C17D	C083						
PIONT	C190	C02C						
TOP	C196	C064						
KOPU	C19D	C0B7	C143	C261	C2B4			
EDIT	C1AD	C046						
ED1	C1AD	C1DF						
ED2	C1B5	C1DA	C1EE	C201	C21D			
ED3	C1DC	C1C2						
ED4	C1E1	C1BA	C1DC	C1EB				
ED5	C1EB	C1BE						
CHDEL	C1F0	C1D0						
CHINS	C203	C1D4						
DUPL	C21F	C1CB						
LANG	C237	C1E1	C1F0	C20B				
MOVE	C242	C1E4						
MOV1	C24E	C253						
G0	C26C	C049						
G7	C275	C2A0						
G1	C27A	C2B1						
G5	C283	C29B						
UP	C287	C05E						
G4	C292	C298						
G8	C29D	C26F						
G1ZZ	C2A2	C292	C30E					
G6	C2A3	C2AA						
G1Z	C2AE	C27A	C2B9					
G1Z1	C2BC	C2B1						
CRLF	C2BE	C094	C131	C2F3				
PULOE	C2C8	C1C6						
PUL1	C2D4	C2D6						
APZP	C2DB	C1E7						
APZP1	C2E5	C2EB						
ZLOE	C2F8	C1B2						
ZL1	C305	C309						
TEPUE	C31C	C2A5						
TEP1	C328	C323						
REBT	C330	C1F3	C20E					
E1	C33A	C03E						
BRA	C33D	C344	C352					
E2	C341	C1A7						
E3	C34F							
STAB	C354	C011	C35A					
LSTAB	0006	C017						
RAHEZ	0D39	C014	C01C	C022	C07A	C097	C196	C31D
		C328						
RAHAZ	0D3B	C090	C1A0	C347				
ZEZL1	0D3D	C025	C07D	C0B2	C0CC	C0FA	C10E	C13A
		C151	C199	C248	C275	C283	C28E	C2E2
		C2F0	C2FC	C311				

Maschinencode

```

:10C00000F5C5D5E52157C1CD2801CD0C01FE59203C
:10C01000112154C311390D040600EDB02A390D3E2E
:10C0200003772A390D223D0D2128C0E52190C1CD8D
:10C030002801213FC0C0D001CD7A00DA3AC3E9A195
:10C040006DC04412C145ADC1476CC2496B034C522F
:10C05000034F8C035094C05167C052B2C05587C281
:10C06000057DBC05496C1FFE1E1D1C1F1C9216AC1DA
:10C07000CD2801CD1805C8CDDC0422390D223D0D97
:10C080000380377217DC1CD2801CD1805C8CDDC0444
:10C090000223B0DC9CDBEC22A390D7F8E03C8FE1E4D
:10C0A0000CAAC02B4PCDE0018F00B0DCDE0003E0D
:10C0B0000AC92A3D0D232BCD9DC1CDE300E67FFEAD
:10C0C0000D280DFE0A280977F8E0320E3A223D0DC93E
:10C0D0000237E2BFE1E28E33B1E18ECCD18053815D6
:10C0E0000CDD04EBCDFAC0F803C81B7BB220F50FE8
:10C0F00003CD9E00C921FFFF1E892A3D0D232B7E5E
:10C10000F54FCD8900F1FE03C8FE1E20F12B223D0C4
:10C110000D9C9D1805D8280FCD0C04C2D2CF1C803E5
:10C12000082B7DB420F5C9CDD2FC01F80320E4C9C5D
:10C130000E5CDBEC221C0FF22F10F2A3D0D7F8E03D8
:10C1400028122BCD9DC1FE18F54FC4E60079F120CB
:10C150008C223D0DDE1C1C9D0DA54452030310D0AD4
:10C160004E845573F2020284E29000D0A45428420B9
:10C170004F46204D454D4F52593F202000D0A5348
:10C1800054415254204F46204D454D4F52593F0087
:10C19000D0A3E3E20002A390D223D0DC9E5D5EBA2
:10C1A0002A3B0DEBB7ED52FC41C3D1E1C9D2FC104
:10C1B000F803C4F8C2CD3C01F80D2825F1A1A282B33
:10C1C000F80A2818F87FCC8C2FE04CC1FC2F8188F
:10C1D0002818FE143282D4FCDB60018D9CDE1C11839
:10C1E0000CCC37C2CD42AC2CDDBC2C9CDE1C118C56D
:10C1F000CD37C2CD30C32AF10F85D123E8D03E20BB
:10C200007718B22AF10F857E86777CD37C2CD30C1
:10C210000321F8FF11FF8FEDB88138A2771896C5E0
:10C2200005E52180FF11C0FF013800E8D80E53F1D7
:10C230000F3E208D1D1C1C9E52AF10F7D21C0FF9554
:10C24000E1C9B7C8E5D5C5F52A3D0D0100007E2B33
:10C25000F803C320F923E5D1F1C056004F0C87ED2D
:10C2600042CD9DC1C1E8EDB0C1D1E1C9CD180538BA
:10C270002CC8CDDC04ED5B3D0DEBDAEC2D81B7BP5
:10C28000B220F7223D0DC9C21805C8CDDC04ED5B09
:10C290003D0D0CDA2C22B7DB420F8EB18E621FFFFA7
:10C2A00018D313131ACD1C33FE1E20F71BC97FE824
:10C2B0000328092BCD9DC1FE1E20F3C937CF9F5C42
:10C2C0000E18CDB600C1F1C9C5E06413E2021BF85
:10C2D000FF22F10F772310FC8E1C1C9B7C8C5471190
:10C2E0000CFF2A3D0D1A867F77132B40F8361E2B60
:10C2F0000223D0DCDBE0C1C9F5C5D5E52A3D0D5E2E
:10C300000100003E0356032BBA20FA8E1E5D1CDA28D
:10C31000C2E533D0DE8E1D1C1F1C9E52A390DA4
:10C32000B7ED52FC28C3E1C9ED5B390D133E1EC9C0
:10C33000F5473E3F90060004FF1C921453122FEFFEF
:10C34000C9214532CD3DC32A3B0D3603C3380021F8
:10C35000453318E9FF7F0040FF7FFFFF9FF9FF2E
:10C36000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FFD
:10C37000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FFC
:10C38000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FFB
:10C39000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FFA
:10C3A000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF9D
:10C3B000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF8D
:10C3C000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF7B
:10C3D000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF6D
:10C3E000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF5D
:10C3F000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF4D

```

5.2.7. Programm Assembler

```

:10C40000C3B0C4C345220040000F06000BDD461B2D
:10C41000DD46B0DD70B1DD70BFFD365F01006F114C
:10C42000007F2100DF31001E3A00F782A0081320099
:10C43000E722006E8ED5B9FDD219EDD2AAEFD2A8E98
:10C44000ED7BAFFD21E5ED438E6D535ED4ABE9DD20
:10C4500022BAFDD22E8ED731C0A001D1A00C1020049
:10C46000D1120087F90089DDF98AFDF9138D57141F
:10C47000ED5F31ED4741ED4F00000000000000008E
:10C480004100420043004400450048004C00460083
:10C49000490052004D0042434445414653505043E9
:10C4A0004958495849584C0000000000000000000B5
:10C4B0002100C411001001AF00E8D02180C5118032
:10C4C0001501AF18EDB0C30310FFFFF9FF9FF9FF9FF23
:10C4D000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF6C
:10C4E000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF5C
:10C4F000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF4C
:10C50000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF3B
:10C51000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF2B
:10C52000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF1B
:10C53000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FF0B
:10C54000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FFFB
:10C55000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FFEB
:10C56000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FFDB
:10C57000FFFFF9FF9FF9FF9FF9FF9FF9FF9FF9FF9FFCB
:10C580000000000000414443205B414444205A414E96
:10C5900044205D424552206642495420543414361
:10C5A000203943414C4C3A43414D203B43414E439B
:10C5B0003C43414E5A3D434150203E434150453F4C
:10C5C0004341504F4043415A20414343462000439A
:10C5D000442020614350442018435044219435059
:10C5E000492014435049521B43504C20014444120DA
:10C5F000206544414120024442202064454320B8
:10C600005944454620684449202030444A5E3A1343
:10C610004549202044544A5436945484206B64521
:10C6200051552067455820204A45584146054558F0
:10C6300058200644414C5407494D30201C494D3183
:10C64000201D494D32201E494E202037494E43209F
:10C6500058494E44201F494E445220494E462023FB
:10C66000494E492021494E4952224A4D5020574AAD
:10C6700050432042A4504D20434A504E43444A5072
:10C680004E5A454505020464A505045474A50500D
:10C690004F484A505A20494A522020324A52432099
:10C6A000334A524E4A3344A524E5A354A525A203631
:10C6B0004C442020624C444420244C444452254C99
:10C6C000444920264C444952274E454720284E4F86
:10C6D000502008F4F5220205F4F524720634F544450
:10C6E000522A4F5449522C4F555420384F555444DE
:10C6F000294F5554492B504E20206C504F50204B01
:10C7000050553484C524320200C5245332055520E
:10C710004552409524554492F5245544E30524CE8
:10C7200020204D524C41200A524C43204E524C4342
:10C73000410B5244C4402D524D20200D524E43208F
:10C74000E8524E5A200F52502020105250452011AE
:10C7500052504F2012525220204F5252412014521E
:10C76000524320505252434115525244202E5253AC
:10C77000542017525A20201354243205C534346F7
:10C7800020165345542056534C41205153524120BA
:10C790005253524C20535422058449544C6AD4
:10C7A000584F5220603F2F27F3FB08D97600C91756
:10C7B000007D8F8D0C0F088E0C81F0F37F49B9A163
:10C7C0001B4565E5AABA2657A08B8A0B04ABBB3C
:10C7D00043B36F674D51018B38302028D3D3CDDC6D
:10C7E000PC4C4F4EC84CDDAFD2C2F2EAE2CAE352
:10C7F000C1C51000180820283B4080C8904058011

```


5. Anhang

:10D00000150D2E535E2C7E69732C77C35B20FE6B4
:10D01000CA192015F802DA2E202E97FE04DA220ED
:10D020002E717E2E35FE58280236FD2C180616026B
:10D03000B720011545217D107E261368B6772C791F
:10D04000E60FF0F2015B1EA4B20204D2E6FC7E87CB
:10D0500028043C4571F2E7F5E69732E39AFB62ED1
:10D0600034722003C3E818AF213613772C72775E
:10D07000C39F1CCD762D2806CD1822C35E214158D2
:10D08000137E7F0830F2E2E5B7EB728E0CD41AC222
:10D090005B212E581E4954010800EBD0C2E5B2C217
:10D0A0005B212419313E854010800EBD0C21F813E8
:10D0B0001E2E54010200EBD0C6DE28201BCD891A22
:10D0C0002006CD2D22C35E21216013541E2E80E0298
:10D0D000C6F2420EDC35E21CDD828C35E2121640D
:10D0E000137E84C1822CD441AC43922CD911A2018
:10D0F0002E21F9133604CD512921F82411201501D0
:10D100001800EBD0A2C13222813B432C13CD6200
:10D1100026211713113215010500EBD0CDE32A21A8
:10D12000B722CD7F2D000000004421CDB0DFF7E9C20
:10D13000F54FCD66DF1CA5419217B107F87B5A970
:10D1400027C35C22215313224913224B13000000F2
:10D150000C90000000CD441ACA6321CD9221E0839
:10D16000C3F718CD5C2BC25B2121F8134E2C461E4E
:10D1700008213013712C702E6A7FE6020F2E7F878C
:10D18000C3F178CDD81CCD441A20442E6F78FE2737
:10D19000202C2C7E541E35120622132C7EB7282FED
:10D1A000FE27280B1205F29A21CD412204180523EF
:10D1B0007EB7201B3E23902E34774FAF1843CD5CB3
:10D1C0002B2000CD1022200721351371C3E81C384
:10D1D000781ACDE11CCD441AC29C1BCD5C2BC29C91
:10D1E0001B21F13462C7E2B3B12CD441A1283E
:10D1F00008CD39221B04C3F718CD5C2BC70F3CD10C7
:10D200002247E2307E2C8147FE88471E04C3712119
:10D2100021F8134E2C7EB7C9162121FA13723A7BDB
:10D2200010B720062A3213221B15C35129162218C3
:10D23000B916231E516241E81162518D16261818
:10D24000D9162718D52100312E00247E47A777B8E8
:10D250007028F7B7010D00ED42227E10CD1D2E2162
:10D2600007130E103E30CD9F2421C322CD72DCD0A
:10D270004421CDE0DFF8E51C84FCD6E6DF7217B10A0
:10D28000FE313601CA2619F8A7F3600CA3619FE5045
:10D290003602CACB1818D2000000000000000000BF
:10D2A00000000000000000000000000000000007E
:10D2B0000000000000000000000000000000000A4
:10D2C0005929000D0A415353454D24C45520D0A10
:10D2D000504153533A20200000000000000000009D
:10D2E00000000000000000000000000000000003F5
:10D2F0000010FB0663040D2823CD012E8E67F00FEFF
:10D30000C28F6E0D2815FE0A2811F81E280DFE1B
:10D31000092002C3E20770528DC2318DD361E21FA7D
:10D320002362C2C21F1233620211E24363B21422492
:10D33000362716640B47219C1406007FEF3BCA3138
:10D3400024F81E200A78B720AE216413361EC9FE23
:10D350002020067E8B72040181CFE3A20130E0678CD
:10D36000B7C4BF23444DC602215813776069181A09
:10D37000FE3B280504237E18C878B7201A8B2164E9
:10D380001336202C363BBEC32D2406F04237FEF0F
:10D3900020202278B728FE6E0E04216E132D3620BC
:10D3A0000D20FAEB78B728060E0553CDBF237FEF7E
:10D3B0001EC82B1832FE1E28DDEF3B20CF18D8CD0B
:10D3C000A5247D906F30012578B930014806005A8B
:10D3D00016138DB07BCDB1245F78B9C9EB3600CDB5
:10D3E0007FE2028F8BEC9166F0E2606FF04237FE6A
:10D3F000FE20200478B728F67FE2C2020545D21E4

:10D40000FA23360021F12336002195133620626B72
:10D41000166F78B7C4BF2316950E2718CEFE3B2093
:10D420001C78B7C4BF23DCDC237FE3BC016BC0ED9
:10D430003C2B06FF04237FE1E20F918E4FE1E2866
:10D4400080FE27200CAF32FA23321E2432242418D9
:10D450009CF82720903E3B321E241891FE40D0C6E81
:10D46000F9C9E60F07070777C9B60FE677C94678
:10D47000EB7BBEC023EB230D20F5C97E0F0F0FF5
:10D480008B6FF630FE3AD8C607C97E18F3FE3AD04A
:10D49000573B2FA7C9FE5BD0573E40BA7AC97759
:10D4A000203D20B7C9D57C21BD2470237123732D0
:10D4B0007721BD2446234E235E23666BC9000000FE
:10D4C00000415353454D24C45520524F544FADF
:10D4D0004F4C4C42465A202020202020204F42F2
:10D4E0002E434F4445202020205A5494C52020BA
:10D4F0002020205155454C450524F752414D4B4
:10D5000020454E544841454C542020202A2A20A8
:10D510004645484C4552204845583D3D4455A21D2
:10D5200028132D7FE2028FAFE48CA8E25000000F8
:10D5300000000000000000000FE42C3A3726FE4200121
:10D540002D7DF81C3054CD9D25C847C9D5C5C8B7E8
:10D55000280778C60A471D18F7C9D2C5C8B71600BE
:10D56000280B78C664477ACE00571D18F3CD9C254A
:10D57000C8B7280B78C6E8477ACE03571D18F3CDF5
:10D580009C25C8B77A280D78C6C10477ACE27573819
:10D59000091D18F12E28702C7F709AFC9D27ED63001
:10D5A00038F8F80A5F9D7F81C30B2DCDDBF27
:10D5B00025C8432DCD7F25C8E28702C731810CD3B
:10D5C000325C85F2DCDD325C807070707835F3B46
:10D5D00001B7C97ED63038C2FE0A8D607FE0A384F
:10D5E00089F810D8AFC90000000000000000000024
:10D5F000000000000000000000000000000000002B
:10D6000000000000000000000000000000000000A
:10D61000000000000000000000000000000000000A
:10D62000000000000000000000000000000000000F
:10D63000000000000000000000000000000000000F
:10D640002E28702C77AC9C90808AF572D7ED630DAF5
:10D650009A25F8D00ED29C26CF7AC800F0D20EA47
:10D660000CC9212913562D462116130E0078D61009
:10D67000477ADE275738030C18F378C610477ACE5E
:10D6800027DC52678D6E8477ADE035738030C182D
:10D69000F378C6E8477ACE03CD52678D664477AB4
:10D6A000DE005738030C18F378C664477ACE800CDF5
:10D6B000C526DC8E26C60A477AC8C5263E30802C5B
:10D6C000772C3644C95779C6302C770E00C978D6E6
:10D6D0000A47D80CC3CE2621291352E46211B13E1
:10D6E00036482DCDE262D43CDE8C26C978E60FCD5E
:10D6F0000027778E6F007070707070707070707C9C1
:10D700000C630FE3AD8C607C9E5210013732C722C27
:10D71000712C702CC1702C712C7272106137E2D4E2C
:10D720002D46C52D462D4E2D562D5E1C9210614E0
:10D730007EB7C0442821F8137EB7C27C27E2B347ED6
:10D74000B7C82107143635210514581C1C732E063C
:10D7500036022E076E542613CD0328212E13CD77C3
:10D7600027210814347FE40CC9027210714342151
:10D77000341335C24727C93AC2334C92130135468
:10D780001E28010200EBD02106147FE02C2C1274A
:10D79000210814541E12CD03282E0836002E0636FA
:10D7A00001CD14282B046E54ECD9D9C15D2E05D6
:10D7B0006E2C2C7DBBCABE276B2CC3A727CD4457
:10D7C00028212F1311414C309282106147FE02E8
:10D7D0000C90270E3ACDE9DF0E18061500QDCFPADB
:10D7E00027CDF627CDF627CDF62700000000000054
:10D7F000000000C35C22063216003B304ACDE9DF4D

```

:10D8000010F8C946EB2CC3EC264BCDDA265901049F
:10D8100000EDB0C921121406007EDC5C24E60F078E
:10D820000707074F2C7ED5C24E60FB180472D5DA6
:10D830002E057EBB28056B2C2C18DEAF90475413A9
:10D8400013C30428CDF627210814AFOE07CD9F245B
:10D85000361E2C36202C363A2E0636012E140E0695
:10D860003E30CD9F24E0536182E043611C921597D
:10D87000131805462C560E087A175778173002EEFD
:10D8800031470D20F32C561D20BC21FF1370682624
:10D89000127EB7281F256E6711581346EB7EB828F5
:10D8A0001678C603300114835F300114EB7E2366C3
:10D8B0006FB420E4F601C9234EBB237EB9EB200ED2
:10D8C0000520F4EB23116013010200EDB0C904043C
:10D8D0000EB230520FCC3AD28212E13541E6001024A
:10D8E00000EDB021FF136E26127EB72856256E6715
:10D8F0007EC604300124856F3001247E2BB62806B5
:10D900007E23666F18EAB210014010200EDB021BE
:10D9100000145E2C562158134E0C0600EDB0216009
:10D92000130140400EDB0210014732C722A7B107CC8
:10D930000A280AD01628CD1A22C9C300007DEBDA46
:10D940003429C9EB2100147E2C666FEB7225731805
:10D950000C421F9137FE01CABF2AF802CA72AFED0
:10D9600004CADD29FE08CA022AFEB10CAA729FE2023
:10D970000CA9929FE40CA8829FE80CABF29C9C3D4D2
:10D980002D00000000000000CD7E292C1C24114FC0
:10D9900015011200EDB0C3E3A2A21D324110A1501A9
:10D9A0002D00EDB0C3E3A2A2104153E3D0E4BCD9F63
:10D9B000024211725110815010800EDB0C3E3A2A2121
:10D9C0003413111315462CCD7B244FEB712CEBDC6A
:10D9D0008A244FEB712C2CEB0520EB212E13541EC7
:10D9E00028010200EDB0CDD726211731107150102C
:10D9F0000400EDB0CDD622621171311041501000AE
:10DA0000EDB0212A13541E28010200EDB0CDD62268C
:10DA1000211713112015010500EDB02164137FEFE
:10DA20003B2862FE202866B72811127150606CD6F
:10DA30009B2A28030520F8626B36A1261A13112EBF
:10DA4000150605CD9B2A28030520F8216F131133F5
:10DA500015CD9B2A20FB2195137EB780AEB362C87
:10DA60002CEBBD9B2A20FB21BC13EB3E47BD38029B
:10DA70002E47EB063CCD9B2A28697BFE703064055F
:10DA8000C2752A185E1127150648C3752A2C7EFE1A
:10DA90003B20A821471511BC1318D77EB7C84F2CFB
:10DAA000EB712CBBF601C9212A131E285401020048
:10DAB0000EDB0CDD62262E17112015010500EDB02125
:10DAC0002C133420022C34210C25110415010B00D9
:10DAD000EDB021FA13CD7B244FEB712CEBBD8A24D2
:10DAE0004FEB71C3A22D000000000000000000021D8
:10DAF0006F13CDA5240625CD372CC0CDB124CD691B
:10DB00002CCEAF2BCDB1247EFE2820012CCDA5241C
:10DB1000CD532C0405CAAF2B05281A05282D05283E
:10DB20002ACDB12401020011FD13EDB07EB7C8FE6D
:10DB300029C02C18F7CDB1242C2C7EFE2920E2EB35
:10DB400021FD13364D2C3600EB18E121FB13360076
:10DB50002C3600CD212BC8CDA5241819216F13CD4B
:10DB6000A52406257EFE27285ACD372CC0CDB1240A
:10DB7000CD692C283D21FB1336002C3600CDB12475
:10DB8000CDA5247EFE28282E0600FE2D200104218E
:10DB9000021470CDB1247EFE282D2821FE2B281DFEFF
:10DBA000232835CD2B2CDA72FCDD312CDA0B2DF61A
:10DBB00001C9CDB1242C2CCDA52418B92CCDA52478
:10DBC0007E18DC2C11FB13010100EDB0EB3600EBED
:10DBD0007EFE27C02C7EB7C92CCDA524212E134E46
:10DBE0002C462102147EB72807AF914F3E0098477C
:10DBF00021FB137EB81772C7E8877CDB1247EFE2B88
:10DC00000CA802BFE2DCA802BFE2920022C7EB7C095
:10DC10002103147EB7C8FE48280721FC133600AF45
:10DC20000C921FC137E36002D77AF94ACD96245109
:10DC30000C94ACD8D2451C90E007EFE28280BFE292D
:10DC4000280D2C0520F3797B7C90C0DC00C18F30D65
:10DC50000C018EF562C5E218010061405C87E234E96
:10DC600023BA20F779B20F3C97EFE482804FE4C76
:10DC700020054F2C7EFE282103143600C071C92CCC
:10DC80002C7EFE2EBCAED2C2132131E5954010200A7
:10DC9000EDB0CDB1241E5B5478CD2B2CD2AF2B0129
:10DCA0000100EBD0CDDC2C2034CDDC2C2030CDDCC0F
:10DCB0002C202CDDC2C2028CDA5243E07902158FB
:10DCC0001377CD6E28C0216013C3DF2B7ECD2B2CA4
:10DCD000D4312CD2AF2B010100EDB0AF9C04040444
:10DCE00004EB436002CDD20FAEB03B82CCDB12440
:10DCF0001E5954010100EDB07ECD2B2CD4312CD215
:10DD0000AF2B010100EDB02CC3952C1E175401114F
:10DD100000EDB0CDB1241E1754061178CD2B2CD4AE
:10DD2000312C302AC2C10520F2CDDA5242117137E7E
:10DD3000FE30200E1E1E54EB011000EDB0EB362023
:10DD400018EACD1F25CAAF2B212813C3DF2EBB36D2
:10DD5000202C0520FAEB0CA52418D1219513CDA5B3
:10DD6000240626C3F72A3ABB13B7C0219513CDA5C5
:10DD7000240626C3642B216413060478CD2B2CD2EB
:10DD8000AF2B2C7EB7280CCD2B2CD4312CD2AF2B23
:10DD90000520FE78FE0338082E64CD532C0405C906
:10DDA000AF2B2C7EB7280CCD2B2CD4312CD2AF2B23
:10DDB00028FAE5D1E1E5EBED5245040E14ECDECE66
:10DDC000DF2310F92104153E20770180001105158D
:10DDD000EDB0F1C1C50800EDCDDCF0E0ACD8CDDFC10B
:10DDE000CDFE2FDFE6C00D90D96C00D90D96C00D90B
:10DDF000DFE20C8C35C227EB7C84FCD6E6DF231804
:10DE0000F6CDEDFE60FC23DFD97DB4200330000DB
:10DE100007E2BCB77C41A2E9D9C9CAF9C9092A30F3
:10DE20000D0E0ED9C9000000000000000000000027
:10DE3000000000000000000000000000000000000E2

```

5. Anhang

5.2.8. Programm CBIOS

Assemblerausdruck

EBAS-88

LOC	CODE	LINE	SOURCE
		00001	PN CB
		00002	
		00003	
		00004	
		00005	TITL 'CBIOS Y23V0'
		00006	
		00007	;DEFINITION VON KONSTANTEN
		00008	
		00009	CFDC: EQU 7CH ;STEUERUNG FDC
		00010	DFDC: EQU 7DH ;DATEN FDC
		00011	TC: EQU 78H ;TERMINAL COUNT (ENDE-IMPULS)
		00012	PIOAD: EQU 0FCH
		00013	NDISK: EQU 2 ;ANZAHL LAUFWERKE,1 IM RAM SIMULIBRT
		00014	CR: EQU 0DH
		00015	LF: EQU 0AH
		00016	ROM: EQU 0E200H ;60K RAM
		00017	BDOS: EQU ROM-0E00H
		00018	CCP: EQU ROM-1600H
		00019	RDSK: EQU 7000H ;IM RAM SIMULIERTE FLOPPY
		00020	MCSTS: EQU 0F003H
		00021	CIO: EQU 0F13CH
		00022	MCO: EQU 0F009H
		00023	MWO: EQU 0F00FH
		00024	MRI: EQU 0F012H
		00025	RESTR: EQU 0F400H ;RESTART MONITOR
		00026	ZS: EQU 0F089H
		00027	STR: EQU 0F128H ;STRING-AUSGABE
		00028	
		00029	ORG ROM
		00030	
E200	C39BE2	00031	JMP BOOT
E203	C3CDE2	00032	WBOTE: JMP WBOOT
E206	C303F0	00033	JMP MCSTS
E209	C33CF1	00034	JMP CIO
E20C	C309F0	00035	JMP MCO
E20F	C353E5	00036	JMP DRUCK
E212	C30FF0	00037	JMP MWO
E215	C312F0	00038	JMP MRI
E218	C333E2	00039	JMP HOME
E21B	C338E2	00040	JMP SDISK
E21E	C36BE2	00041	JMP STRCK
E221	C366E2	00042	JMP SSEC
E224	C370E2	00043	JMP SDMA
E227	C353E3	00044	JMP READ
E22A	C349E3	00045	JMP WRITE
E22D	C353E2	00046	JMP LSTS ;LISTER STATUS
E230	C354E2	00047	JMP STRAN ;SECTOR TRANSFORMATION
		00048	
		00049	;KOPF AUF SPUR NULL STELLEN
E233	010000	00050	HOME: LD BC,0
E236	1833	00051	JR STRCK-#
		00052	;LAUFWERK AUSWAEHLEN
		00053	;IN:C LAUFWERK (A ODER B)
		00054	;OUT:HL DPH-VEKTOR
E238	210000	00055	SDISK: LD HL,0 ;FEHLER
E23B	05	00056	PUSH BC
E23C	0600	00057	LD B,0
E23E	79	00058	LD A,C
E23F	FE01	00059	CMP NDISK-1
		00060	DISKS: EQU #-1
E241	D200F4	00061	JPNC RESTR ;FEHLER
E244	69	00062	LD L,C
E245	60	00063	LD H,B
E246	29	00064	ADD HL,HL
E247	29	00065	ADD HL,HL
E248	29	00066	ADD HL,HL
E249	29	00067	ADD HL,HL
E24A	1115E5	00068	LD DE,DPBAS ;*16
E24D	19	00069	ADD HL,DE ;DPH-VEKTOR 1.LW
E24E	320400	00070	LD (4),A
E251	C1	00071	POP BC
E252	C9	00072	RET
E253	C9	00073	LSTS: RET ;LISTER STATUS

```

00074 STRAN: EQU # 0 ;SECTOR TRANSFORMATION
00075 ;IN: BC LOG.SECTOR
00076 ; DE TABELLE
E254 0600 00077 LD B,0
E256 7A 00078 LD A,D ;DE=0-3KEINE TRANSP.
E257 B7 00079 OR A
E258 2004 00080 JRNZ STRN1-#
E25A 60 00081 LD H,B
E25B 69 00082 LD L,C
E25C 23 00083 INC HL
E25D C9 00084 RET
E25E EB 00085 STRN1: EX DE,HL
E25F 09 00086 ADD HL,BC
E260 7E 00087 LD A,(HL) ;NEUE NR
E261 3212F8 00088 LD (SEC1),A
E264 6F 00089 LD L,A
E265 C9 00090 RET
00091 ;SEKTOR EINSTELLEN
00092 ;IN: C SEKTOR
00093 ;OUT: (SEC1) =C
00094 SSEO: EQU 0
E266 2112F8 00095 LD HL,SEC1
E269 71 00096 LD (HL),C
E26A C9 00097 RET
00098 ;SPUR ANWAEHLEN
00099 ;IN: C SPUR
00100 ;OUT (TRACK)=C
00101 STRCK: EQU #
E26B 2102F8 00102 LD HL,TRACK
E26E 71 00103 LD (HL),C
E26F C9 00104 RET
00105 ;DMA SETZEN
00106 ;IN: BC DMA
E270 ED4313F8 00107 SDMA: LD (DMA),BC
E274 C9 00108 RET
00109 ;-----
00110 ;CCP UND BDOS VON DEN ERSTEN SPUREN DER DISKETTE LADEN
E275 0A 00111 MSGS: DB 0AH
E276 0D 00112 DB 0DH
E277 0D 00113 DB 'Y23V0 56K CP/M 2.2'
E289 0D 00114 DB 0DH
E28A 0A 00115 DB 0AH
E28B 0D 00116 DB 'RAMDISK ? (Y)'
E298 0D 00117 DB 0DH
E299 0A 00118 DB 0AH
E29A 00 00119 DB 0
00120 BOOT: EQU #
E29B 3EC3 00121 LD A,0C3H
E29D 320000 00122 LD (0),A
E2A0 2103E2 00123 LD HL,WDOTB
E2A3 220100 00124 LD (4),HL
E2A6 320500 00125 LD (5),A
E2A9 2106D4 00126 LD HL,BDOS+6
E2AC 220600 00127 LD (6),HL
E2AF 323800 00128 LD (38H),A ;HALTEPUNKT VORBEREITEN
E2B2 2100F4 00129 LD HL,RESTR
E2B5 223900 00130 LD (39H),HL
E2B8 2175E2 00131 LD HL,HSGS
E2BB CD28F1 00132 CALL STR
E2BE CD3CF1 00133 CALL CIO
E2C1 FE59 00134 CMP 'Y'
E2C3 CC16E3 00135 CAZ IRDSK ;RAMDISK ERLAUBEN
E2C6 AF 00136 XOR A ;IOBYTE
E2C7 320300 00137 LD (3),A
E2CA 320400 00138 LD (4),A
00139 WBOOT: EQU # ;LESEN DER SYSTEMSEKTOREN
E2CD 318000 00140 LD SP,80H
E2D0 CDF2E3 00141 CALL INIFD ;INITIALISIERUNG FLOPPY-CONTROLLER
E2D3 0E00 00142 LD C,0 ;LAUFWERK
E2D5 CD38E2 00143 CALL SDISK
E2D8 062C 00144 LD B,2CH ;SEKTORZAEBLER
E2DA 0E00 00145 LD C,0 ;SPUR-NR.
E2DC CD6BE2 00146 CALL STRCK
E2DF 1601 00147 LD D,1 ;SEKTOR
E2E1 2100CC 00148 LD HL,CCP ;LADEZIEL
E2E4 C5 00149 LOAD1: PUSH BC ;NAECHSTEN SEKTOR
E2E5 D5 00150 PUSH DE
E2E6 E5 00151 PUSH HL
E2E7 4A 00152 LD C,D
E2E8 CD66E2 00153 CALL SSEC ;NEUEN SECT.ANWAEHLEN
E2EB C1 00154 POP BC ;BC=LADEADRESSE
E2EC C5 00155 PUSH BC ;LADEADR RETTEN

```

5. Anhang

E2ED	CD70E2	00156	CALL SDMA	
E2F0	CD5DE3	00157	CALL RFLOP	;SYSTEM RAUFSCHREIBEN: WFLOP EINSETZEN
E2F3	FE00	00158	CMP 0	;FEHLER?
E2F5	C400F4	00159	CANZ RESTR	;JA
E2F8	E1	00160	POP HL	
E2F9	118000	00161	LD DE,128	
E2FC	19	00162	ADD HL,DE	;LADEADR. ERHOEHEN
E2FD	D1	00163	POP DE	
E2FE	C1	00164	POP BC	
E2FF	05	00165	DEC B	;SEKTORZAHL - 1
E300	283A	00166	JRZ GOCPM-#	
E302	14	00167	INC D	;NAECHSTEN SECT.
E303	7A	00168	LD A,D	
E304	FE21	00169	CMP 21H	;LETZTER?
E306	38DC	00170	JRC LOAD1-#	
E308	1601	00171	LD D,1	;1.SECT.NAECHST.SPUR
E30A	0C	00172	INC C	
E30B	C5	00173	PUSH BC	
E30C	D5	00174	PUSH DE	
E30D	E5	00175	PUSH HL	
E30E	CD6BE2	00176	CALL STRCK	
E311	E1	00177	POP HL	
E312	D1	00178	POP DE	
E313	C1	00179	POP BC	
E314	18CB	00180	JR LOAD1-#	
		00181		
E316	2100E2	00182	IRDSK: LD HL,0E200H	;RAMDISK INITIALISIEREN
E319	110070	00183	LD DE,RDSK	;RAM-DISK
E31C	013300	00184	LD BC,51	
E31F	EDB0	00185	LDIR	
E321	210370	00186	LD HL,7003H	;SPRUNG ZUM CCP
E324	220100	00187	LD (1),HL	
E327	3E02	00188	LD A,2	
E329	3240E2	00189	LD (DISKS),A	
E32C	210662	00190	LD HL,6206H	;BDOS-SPRUNG
E32F	220600	00191	LD (6),HL	
E332	3EC3	00192	LD A,0C3H	
E334	77	00193	LD (HL),A	
E335	2106D4	00194	LD HL,BDOS+6	
E338	220762	00195	LD (6207H),HL	
E33B	C9	00196	RET	
		00197		
E33C	018000	00198	GOCPM: LD BC,80H	
E33F	CD70E2	00199	CALL SDMA	
E342	3A0400	00200	LD A,(4)	;AKTUELLES LAUFWERK HOLEN
E345	4F	00201	LD C,A	
E346	C300CC	00202	JMP CCP	
		00203		
E349	3A0400	00204	WRITE: LD A,(4)	
E34C	B7	00205	OR A	
E34D	2868	00206	JRZ WFLOP-#	
E34F	CDD5E4	00207	CALL WRDSK	;RAM-DISK
E352	C9	00208	RET	
E353	3A0400	00209	READ: LD A,(4)	
E356	B7	00210	OR A	
E357	2804	00211	JRZ RFLOP-#	
E359	CDE5E4	00212	CALL RRDSK	;RAM-DISK
E35C	C9	00213	RET	
		00214		
		00215	;FLOPPY LESEN MIT SEKTORAUFLÖSUNG	
		00216	;IN: SEC1,TRACK,DMA V. BDOS	
		00217	;OUT:DMA/BDOS=1/2ZDMA/FLOPPY	
E35D	E5	00218	RFLOP: PUSH HL	
E35E	D5	00219	PUSH DE	
E35F	C5	00220	PUSH BC	
		00221	;VERGLEICH OB AUS PUFFER GELESEN WERDEN KANN	
E360	3E46	00222	LD A,46H	;LETZTE OPERATION LESEN ?
E362	2100F8	00223	LD HL,CTAB	
E365	BE	00224	CMP (HL)	
E366	2033	00225	JRNZ RFLP-#	
E368	23	00226	INC HL	
E369	1109F8	00227	LD DE,RESLT	;GLEICHES LAUFWERK ?
E36C	1A	00228	LD A,(DE)	
E36D	E603	00229	AND 3	
E36F	BE	00230	CMP (HL)	
E370	2029	00231	JRNZ RFLP-#	
E372	23	00232	INC HL	;GLEICHE SPUR ?
E373	13	00233	INC DE	
E374	13	00234	INC DE	
E375	13	00235	INC DE	
E376	1A	00236	LD A,(DE)	
E377	BE	00237	CMP (HL)	


```

E378 2021 00238 JRNZ RFLP-#
E37A 23 00239 INC HL ;KOPF ?
E37B 13 00240 INC DE
E37C 1A 00241 LD A,(DE)
E37D BE 00242 CMP (HL)
E37E 201B 00243 JRNZ RFLP-#
E380 23 00244 INC HL
E381 3A12F8 00245 LD A,(SEC1) ;HOSTSEKTOR=BDOS/2
E384 1F 00246 RRA
E385 3814 00247 JRC RFLP-# ;UNGRADE
E387 BE 00248 CMP (HL)
E388 2011 00249 JRNZ RFLP-#
00250 ;GLEICH, ALSO PUFFER LESEN, NICHT LAUFWERK
E38A 2173F9 00251 LD HL,FDMA+80H ;QUELLE
E38D ED5B13F8 00252 LD DE,(DMA) ;ZIEL
E391 018000 00253 LD BC,80H ;ANZAHL BYTES
E394 EDB0 00254 LDIR ;UEBERTRAGEN
E396 C1 00255 POP BC
E397 D1 00256 POP DE
E398 E1 00257 POP HL
E399 AF 00258 XOR A
E39A C9 00259 RET
E39B CDD9E3 00260 RFLP: CALL TRANS ;SEKTOR LESEN + UMRECHNEN
E39E F5 00261 PUSH AF ;Fehler R8272
E39F 21F3F8 00262 LD HL,FDMA
E3A2 CB48 00263 BIT 1,B
E3A4 2003 00264 JRNZ B2-#
E3A6 2173F9 00265 LD HL,FDMA+80H ;QUELLE
E3A9 ED5B13F8 00266 LD DE,(DMA) ;ZIEL
E3AD 018000 00267 LD BC,80H
E3B0 EDB0 00268 LDIR
E3B2 F1 00269 POP AF
E3B3 C1 00270 POP BC
E3B4 D1 00271 POP DE
E3B5 E1 00272 POP HL
E3B6 C9 00273 RET
00274 ;-----
00275 ;FLOPPY SCHREIBEN MIT SEKTOR EINFUEGEN
E3B7 E5 00276 WFLOP: PUSH HL
E3B8 D5 00277 PUSH DE
E3B9 C5 00278 PUSH BC
E3BA CDD9E3 00279 CALL TRANS
E3BD B7 00280 OR A ;LESEFEHLER ?
E3BE 2015 00281 JRNZ B4-#
E3C0 11F3F8 00282 LD DE,FDMA
E3C3 CB48 00283 BIT 1,B
E3C5 2003 00284 JRNZ B3-#
E3C7 1173F9 00285 LD DE,FDMA+80H
E3CA 2A13F8 00286 LD HL,(DMA)
E3CD 018000 00287 LD BC,80H
E3D0 EDB0 00288 LDIR
E3D2 CD25E4 00289 CALL W8272
E3D5 C1 00290 B4: POP BC
E3D6 D1 00291 POP DE
E3D7 E1 00292 POP HL
E3D8 C9 00293 RET
00294 ;-----
00295 ;SEKTORTUMRECHNUNG
00296 ;IN: BDOS-SEKTOR-NR IN SEC1
00297 ;OUT:FLOPPY-SEKTOR-NR IN SECTR
00298 ; B=0 WENN VORDERER TEIL SONST HINTEREN AUSWERTEN
00299 ; GLESENER FLOPPY-SEKTOR IN BIOS-PUFFER FDMA
00300 TRANS: EQU #
E3D9 0600 00301 LD B,0
E3DB 3A12F8 00302 LD A,(SEC1) ;BDOS-SEKTOR
E3DE B7 00303 OR A
E3DF 1F 00304 RRA ;UNGRADE?, /2
E3E0 3007 00305 JRNC B1-# ;NEIN
E3E2 C601 00306 ADD 1
E3E4 DC00F4 00307 CAC RESTR ;Fehler
E3E7 06FF 00308 LD B,OFFH ;KENNZEICHEN
E3E9 C5 00309 B1: PUSH BC
E3EA 3204F8 00310 LD (SECTR),A
E3ED CD2CE4 00311 CALL R8272
E3F0 C1 00312 POP BC ;FLAG F.UNGRADE
E3F1 C9 00313 RET
00314 ;-----
00315 ;TREIBER MIT U8272
00316 ;-----
E3F2 0600 00317 INIFD: LD B,0 ;INITIALISIERUNG P8272
E3F4 10FE 00318 DJNZ INIFD+2-#
E3F6 DB7C 00319 IN CFDC
E3F8 FE80 00320 CMP 80H

```

5. Anhang

```

E3FA 2804 00321 JRZ SPEZI-#
E3FC DB7D 00322 IN DFDC
E3FE 18F2 00323 JR INIFD-#
E400 2122E4 00324 SPEZI: LD HL,STAB-1 ;PARAMETER LADEN
E403 010303 00325 LD BC,0303H ;SPECIFY-COMM 3BYTES
E406 CD92E4 00326 CALL WCOM1 ;SCHREIBEN COMM
E409 010702 00327 RECAL: LD BC,0207H ;SPUR 0 EINSTELLEN
E40C CD85E4 00328 CALL RDY ;DRIVE READY?
;-----
E40F 010801 00329 SENSE: LD BC,0108H ;PRUEFE INTERRUPT STATUS
E412 CD8FE4 00331 CALL WCOM
E415 CDACE4 00332 CALL RBYTE ;RESULT REG 0:IC1,IC2,SE,EC,NR,HD,US1,US2
E418 47 00333 LD B,A
E419 FE80 00334 CMP 80H
E41B C4ACE4 00335 CANZ RBYTE ;PCN HOLEN
E41E CB68 00336 BIT 5,B ;SEEK ENDE?
E420 28ED 00337 JRZ SENSE-#
E422 C9 00338 RET
E423 1F 00339 STAB: DB 1FH ;XXXX-SRT,XXXX-HUT
E424 33 00340 DB 33H ;XXXXXXX-HLT,X=ND
;-----
00341 ;SEKTOR SCHREIBEN/LESEN IM BETRIEBSART OHNE DMA-SCHALTKEIS
00342 ;SECTOR IN (SECTR)
00343 ;SPUR IN (TRACK)
00344 ;AKT.LAUFWERK IN (UNIT)
00345 ;ZIEL-/QUELLADRESSE = FDMA
00346 ;KILL: A B DE HL
E425 11 00347 W8272: DB 11H ;CODE LD DE,...
E426 EDA3 00349 OUTI ;WRITE DATA
E428 3E45 00350 LD A,45H ;SCHREIBKOMMANDO
E42A 1805 00351 JR RWIT-#
;-----
E42C 11 00352 R8272: DB 11H ;CODE LD DE,...
E42D EDA2 00354 INI
E42F 3E46 00355 LD A,46H ;LESEKOMMANDO
E431 3200F8 00356 RWIT: LD (CTAB),A
E434 ED5366E4 00357 LD (MODE),DE ;EINTRAGEN INI/OUTI-OP-CODE
E438 CD75E4 00358 CALL SEEK
E43B 060A 00359 RWLOP: LD B,10 ;ANZAHL VERSUCHE
E43D C5 00360 RWOP: PUSH BC
E43E F3 00361 DI
E43F 0609 00362 LD B,9 ;9COMMANDOBYES
E441 3A00F8 00363 LD A,(CTAB)
E444 4F 00364 LD C,A
E445 CD85E4 00365 CALL RDY ;AUSGABE KOMMANDO
E448 21F3F8 00366 LD HL,FDMA
E44B 0E7D 00367 LD C,DFDC
E44D CD5CE4 00368 CALL RW
E450 D378 00369 OUT TC ;ENDE-IMPULS
E452 CDB5E4 00370 CALL RRSALT
E455 C1 00371 POP BC
E456 C8 00372 RZ ;STO=0-KEINE FEHLER
E457 10E4 00373 ERR: DJNZ RWOP-#
E459 3E01 00374 LD A,1
E45B C9 00375 RET ;ERROR
;-----
00376 ;SCHREIBEN ODER LESEN 256 BYTES
00377 ;IN: HL QUELLE ODER ZIEL
00378 ;C: DATENPORT
00379 RW: EQU #
E45C 0600 00381 LD B,0 ;256 BYTES
E45E DB7C 00382 RW1: IN CFDC
E460 07 00383 RLCA ;RQM-TEST
E461 30FB 00384 JRNC RW1-#
E463 07 00385 RLCA ;DIO
E464 D7 00386 RLCA ;NON-DMA
E465 00 00387 RNC ;FERTIG?
E466 EDA2 00388 MODE: INI ;INI BEI READ DATA SONST OUTI
E468 20F4 00389 JRNZ RW1-#
E46A C9 00390 RET
;-----
00391 SDS: LD BC,0204H ;PRUEFE LAUFWERK STATUS
E46B 010402 00392 CALL WCOM
E46E CDBFE4 00393 CALL RBYTE ;STATUS REG 3:FAULT,WP,RDY,T0,TS,HD,US1,U
E471 CDACE4 00394 RET
E474 C9 00395 ;-----
E475 010F03 00397 SEEK: LD BC,030FH ;COMM SPUR EINSTELLEN
E478 CD85E4 00398 CALL RDY
E47B CD0FE4 00399 CALL SENSE
E47E DB7C 00400 SKBSY: IN CFDC
E480 E60F 00401 AND OFH
E482 20FA 00402 JRNZ SKBSY-#

```

```

E484 C9      00403      RET
              00404      ;
E485 C5      00405      RDY:  PUSH BC                      ;LAUFWERK BETRIEBSFABHIG ?
E486 CD6BE4  00406      CALL SDS
E489 C1      00407      POP BC
E48A CB6F    00408      BIT 5,A                      ;READY-BIT IN STATUSREG.3
E48C CC00F4  00409      CAZ  RSTR                      ;FEHLER
              00410      ;
E48F 2100F8  00411      WCOM: LD  HL,CTAB                      ;COMM IN FDC SCHREIBEN
              00412      ;IN: B ANZAHL D. BYTES,C COMM
E492 CDA5E4  00413      WCOM1: CALL DELAY
E495 DB7C    00414      IN  CFDC
E497 B6C0    00415      AND  OCOH
E499 FB80    00416      CMP  80H                      ;RQM,DIO=OUT
E49B 20F5    00417      JRNZ WCOM1-#
E49D 79      00418      LD   A,C
E49E D37D    00419      OUT  DFDC
E4A0 23      00420      INC  HL
E4A1 4E      00421      LD   C,M
E4A2 10EE    00422      DJNZ WCOM1-#
E4A4 C9      00423      RET
              00424      ;
E4A5 C5      00425      DELAY: PUSH BC                      ;VERZOEBERUNG P.STATUSFLAG 8272
E4A6 060F    00426      LD   B,OFH
E4A8 10FE    00427      DEL1: DJNZ DEL1-#
E4AA C1      00428      POP  BC
E4AB C9      00429      RET
              00430      ;
E4AC CDA5E4  00431      RBYTE: CALL DELAY                      ;1 BYTE LESEN
E4AF CDCBE4  00432      CALL IRDY
E4B2 DB7D    00433      IN   DFDC
E4B4 C9      00434      RET
              00435      ;
E4B5 0606    00436      RRS1T: LD  B,6                      ;LESE 7 RESULT BYTES
E4B7 CDACE4  00437      CALL RBYTE
E4BA 2109F8  00438      LD   HL,RESLT
E4BD 77      00439      LD   M,A
E4BE B6C0    00440      AND  OCOH                      ;ERROR?
E4C0 4F      00441      LD   C,A
E4C1 CDACE4  00442      RESL1: CALL RBYTE
E4C4 23      00443      INC  HL
E4C5 77      00444      LD   M,A
E4C6 10F9    00445      DJNZ RESL1-#
E4C8 79      00446      LD   A,C                      ;FEHLERMELDUNG STATUS REG 0
E4C9 B7      00447      OR   A
E4CA C9      00448      RET
              00449      ;
E4CB DB7C    00450      IRDY: IN  CFDC                      ;BEREIT F.DATENEINGABE ?
E4CD 07      00451      RLCA
E4CE 30FB    00452      JRNC IRDY-#
E4D0 E680    00453      AND  80H
E4D2 07      00454      RLCA
E4D3 DB      00455      RC
              00456      ;
E4D4 FF      00457      ;FEHLERBEHANDLUNG
              00458      ERM: DB  OFFH                      ;SPRUNG IN MONITOR
              00459      ;
              00460      ;RAM-DISK
              00461      ;SCHREIBEN
E4D5 CDF3E4  00462      WRDSK: CALL RADR                      ;SEKTORADR BERECHNEN
E4D8 3816    00463      JRC  BDERR-#                      ;BEREICH UEBERSCHRITTEN
E4DA EB      00464      EX   DE,HL                      ;ZIEL-/QUELLADR. RAM-DISK
E4DB 2A13F8  00465      LD   HL,(DMA)                      ;ZIEL-/QUELLADR.BDOS
E4DE 018000  00466      RDSK1: LD  BC,80H                      ;=1 SEKTOR
E4E1 EDB0    00467      LDIR
E4E3 AF      00468      XOR  A
E4E4 C9      00469      RET
              00470      ;LESEN
E4E5 CDF3E4  00471      RRSK: CALL RADR                      ;FEHLER
E4E8 3806    00472      JRC  BDERR-#
E4EA ED5B13F8 00473      LD   DE,(DMA)
E4EE 18EE    00474      JR   RDSK1-#
E4F0 3E01    00475      BDERR: LD  A,1                      ;BDOS-ERROR
E4F2 C9      00476      RET
              00477      ;SEKTORADR. IM RAM BERECHNEN
              00478      ;DE AKT. SPUR
              00479      ;C AKT.SSEKTOR
              00480      ;B MAX.SSEKTORANZAHL/DISK
E4F3 1600    00481      RADR: LD  D,0
E4F5 3A02F8  00482      LD   A,(TRACK)
E4F8 5F      00483      LD   E,A
E4F9 3A12F8  00484      LD   A,(SEC1)

```

5. Anhang

```

E4FC 4F      00485      LD      C,A
E4FD 210000   00486      LD      HL,0
E500 061A     00487      LD      B,26
E502 19       00488      RADR1:  ADD  HL,DE          ;UMWANDL.SPUREN IN SEKTOREN
E503 10FD     00489      DJNZ   RADR1-#
E505 09       00490      ADD    HL,BC          ;AKT. SEKT.DAZU
E506 29       00491      ADD    HL,HL          ;UMWANDLUNG SEKTOREN IN BYTES
E507 29       00492      ADD    HL,HL
E508 29       00493      ADD    HL,HL
E509 29       00494      ADD    HL,HL
E50A 29       00495      ADD    HL,HL
E50B 29       00496      ADD    HL,HL
E50C 29       00497      ADD    HL,HL
E50D 010070   00498      LD      BC,RDSK          ;RAM-DISK-ANFANG
E510 09       00499      ADD    HL,BC
E511 3EBF     00500      LD      A,0BFH          ;HIGH-ADR RAM-DISK
E513 BC       00501      CMP    H
E514 C9       00502      RET
;-----
E515 0000     00503      ;
E517 0000     00504      DPBAS: EQU #
E519 0000     00505      DPHA:  DA  0
E51B 0000     00506      DA  0
E51D 15F8     00507      DA  0
E51F 44E5     00508      DA  0
E521 B4F8     00509      DA  DIRBF
E523 95F8     00510      DA  DPBF
E525 0000     00511      DA  CHKO
E527 0000     00512      DA  ALLO
E529 0000     00513      DPHR:  DA  0
E52B 0000     00514      DA  0
E52D 15F8     00515      DA  0
E52F 35E5     00516      DA  0
E531 E3F8     00517      DA  DIRBF
E533 C4F8     00518      DA  DPBR          ;GERÄTEBESCHREIBUNG
;-----
E535 1A00     00519      DA  CHK1          ;DIRECTORY-KONTROLLE
E537 03       00520      DA  ALL1          ;BELEGUNG DISKETTE
E539 07       00521      ;DISK-PARAMETER-BLOCK RAM-DISK
E53B 00       00522      DPBR:  DA  26          ;SEKTOREN
E53D 1400     00523      DB  3          ;BLOCKGRÖSSE 1K
E53F 00       00524      DB  7          ;(SEKTOREN/BLOCK)-1
E541 00       00525      DB  00         ;NICHT BENUTZT
E543 1F00     00526      DA  20          ;(BLOECKE/DISK)-1
E545 80       00527      DA  31          ;32 DIRECTORY-ENTR.
E547 00       00528      DB  080H        ;1BELEGTER BLOCK DURCH DIR.
E549 1000     00529      DB  0
E54B 0000     00530      DA  16          ;DIR/4
E54D 0000     00531      DA  0          ;0SPUREN DURCH SYSTEM BELEGT
E54F 2000     00532      ;DISK-PARAMETER-BLOCK FLOPPY 5,25 ZOLL
E551 04       00533      DPBF:  DA  20H        ;20LOGISCHE=10H=16 PHYSISCHE SEKTOREN/SPUR
E553 0F       00534      DB  4          ;2K-BLOECKE
E555 01       00535      DB  15
E557 4900     00536      DB  1
E559 3F00     00537      DA  73
E55B 80       00538      DA  63
E55D 00       00539      DB  80H
E55F 1000     00540      DB  0
E561 0300     00541      DA  16
E563 0000     00542      DA  3          ;SYSTEMSPUREN
;-----
E565 0000     00543      ;
E567 0000     00544      DRUCK: EQU #          ;FOLGT
E569 0000     00545      ;
E56B 0000     00546      ;
E56D 0000     00547      ;RAM-ADRESSEN
E56F 0000     00548      ;
E571 0000     00549      ORG    0F800H
E573 46       00550      CTAB:  DB  46H          ;MFM LESEN
E575 00       00551      UNIT:  DB  0          ;LAUPWERK A VOREINGESTELLT
E577 00       00552      TRACK: DB  0
E579 00       00553      HEAD:  DB  0
E57B 01       00554      SECTR: DB  1
E57D 01       00555      N:     DB  1          ;N=1=256BYTES
E57F 10       00556      BOT:   DB  16         ;16 SEKTOREN JE SPUR
E581 1B       00557      GPL:   DB  1BH        ;LUECKE
E583 FF       00558      DTL:   DB  0FFH
E585 09       00559      RESLT: BER 7          ;RESULT TAB F.FDC
E587 1E       00560      DZEIL: DB  30         ;DRUCKER 30 ZEILEN/SEITE
E589 00       00561      MERK:  DB  0          ;UMSCHALTZ. FERNSCHR.
E58B 01       00562      SEC1:  DB  1          ;BDOS-SEKTOR
E58D 8000     00563      DMA:   DA  80H        ;BDOS-DMA
E58F 00       00564      DIRBF: BER 128        ;INHALTSVERZ.DISKETTE
E591 00       00565      ALLO:  BER 31
E593 00       00566      CHKO:  BER 16

```

```

P8C4      00567 ALL1: BBR 31
P8E3      00568 CHK1: BBR 16
P8F3      00569 FDMA: BBR 256                ;FLOPPY-ZWISCHENSPEICHER IM BIOS
          00570
          00571 ORG DRUCK
          00572
          00573
          00574 TITL 'DRUCK'
          00575
          00576
          00577 ;DEFINITION VON KONSTANTEN
          00578
          00579 PIOAD: EQU OFCH
          00580 CR: EQU ODH
          00581 LP: EQU OAH
          00582 ZS: EQU OF089H
          00583 ;-----
          00584 ;DRUCK MIT FERNSCHREIBER F12XX
          00585 ;-----
          00586 ;DRUCKPROGRAMM FUER GROSSE UND KLEINE BUCHSTABEN
          00587 ;IN :C ZEICHEN IN ASCII
          00588 ;OUT :A ZEICHEN IN CCITT
          00589 ;KILL:A,F
          00590 PUSH BC
          00591 PUSH DE
          00592 PUSH HL
          00593 LD A,LF
          00594 CMP C
          00595 CAZ ZEPR
          00596 CALL KODE                ;ZEICHEN IN A CCITT,C ASCII
          00597 CALL ART
          00598 LD HL,MERK
          00599 CMP (HL)
          00600 JRZ ZAUS-#                ;KEIN STEUERZ.AUSGEBEN
          00601 LD (HL),A                ;NEUE ART LADEN
          00602 CALL FSA
          00603 ZAUS: LD A,C
          00604 CALL FSA
          00605 POP HL
          00606 POP DE
          00607 POP BC
          00608 RET
          00609 ;KODEWANDLUNG,ASCII/CCITT
          00610 ;IN C ZEICHEN
          00611 ;OUT A CCITT C ASCII
          00612 ;T7S STEUER ASCII
          00613 ;T5S
          00614 ;T7K KLEINE ASCII
          00615 ;T51
          00616 ;T7G GROSSE ASCII
          00617 ;T53
          00618 ;T7Z ZEICHEN ASCII
          00619 ;T52
          00620 KODE: EQU #
          00621 PUSH BC
          00622 LD A,C
          00623 LD HL,T7S
          00624 LD BC,LT7S
          00625 PUSH BC
          00626 POP DE
          00627 CPIR
          00628 JRZ GF-#
          00629 LD HL,T7K
          00630 LD BC,LT7K
          00631 PUSH BC
          00632 POP DE
          00633 CPIR
          00634 JRZ GF-#
          00635 LD HL,T7G
          00636 LD BC,LT7G
          00637 PUSH BC
          00638 POP DE
          00639 CPIR
          00640 JRZ GF-#
          00641 LD HL,T7Z
          00642 LD BC,LT7Z
          00643 PUSH BC
          00644 POP DE
          00645 CPIR
          00646 JRZ GF-#
          00647 LD A,4
          00648 POP BC

```

5. Anhang

```

E5A9 C9      00649      RET
E5AA 19      00650      GF:  ADD HL,DE
E5AB 2B      00651      DEC HL
E5AC 7E      00652      LD A,(HL)
E5AD C1      00653      POP BC
E5AE C9      00654      RET
E5AF 2110F8  00655      ZEPR: LD HL,DZEIL      ;ZEILENZAHL /SEITE PRUEFEN
E5B2 C5      00656      PUSH BC      ;PAUSE NACH CRLF BEI 100 BAUD
E5B3 0640    00657      LD B,40H
E5B5 CD89F0  00658      Z2:  CALL ZS
E5B8 10FB    00659      DJNZ Z2-#
E5BA C1      00660      POP BC
E5BF 35      00661      DEC (HL)
E5BC C0      00662      RNZ
E5BD F5      00663      LCRLF: PUSH AF      ;AUSGABE 4 LEERZEILEN
E5BE C5      00664      PUSH BC
E5BF 3E1E    00665      LD A,30
E5C1 77      00666      LD (HL),A
E5C2 0604    00667      LD B,4
E5C4 3B02    00668      L1:  LD A,2      ;LF IN CCITT
E5C6 CDEAE5  00669      CALL FSA
E5C9 10F9    00670      DJNZ L1-#
E5CB C1      00671      POP BC
E5CC F1      00672      POP AF
E5CD C9      00673      RET
00674      ;SUBROUTINE ART ERMITTELN
00675      ;IN: C ASCII
00676      ;(HL) LETZTE ART
00677      ;A CCITT
00678      ;OUT: A STEUERZEICHEN
00679      ;C CCITT
00680      ;*40H=ZIFFER=DO=1,*60H=KLEINE BUCHST.=A=0,SONST D1=1
E5CE F5      00681      ART:  PUSH AF
E5CF C5      00682      PUSH BC
E5D0 79      00683      LD A,C
E5D1 FE60    00684      CMP 60H
E5D3 300A    00685      JRNC KB-#
E5D5 FE40    00686      CMP 40H
E5D7 300C    00687      JRNC GB-#
E5D9 C1      00688      POP BC
E5DA F1      00689      POP AF
E5DB 4F      00690      LD C,A
E5DC 3E1B    00691      LD A,1BH      ;ZI/ZEI
E5DE C9      00692      RET
E5DF C1      00693      KB:  POP BC
E5E0 F1      00694      POP AF
E5E1 4F      00695      LD C,A
E5E2 3E1F    00696      LD A,1FH      ;KLEINE
E5E4 C9      00697      RET
E5E5 C1      00698      GB:  POP BC
E5E6 F1      00699      POP AF
E5E7 4F      00700      LD C,A
E5E8 AF      00701      XOR A
E5E9 C9      00702      RET
00703      ;FERNSCHREIBDAUSGABE UEBER PIO
E5EA C5      00704      FSA:  PUSH BC
E5EB 0608    00705      LD B,8
E5ED F660    00706      OR 60H
E5EF 17      00707      RLA
E5F0 D3FC    00708      FSA1: OUT P1OAD
E5F2 CD89F0  00709      CALL ZS
E5F5 CD89F0  00710      CALL ZS
E5F8 1F      00711      RRA
E5F9 10F5    00712      DJNZ FSA1-#
E5FB C1      00713      POP BC
E5FC C9      00714      RET
00715      ;TABELLEN
E5FD 0A      00716      T7S:  DB LF
E5FE 0D      00717      DB CR
E5FF 20      00718      DB 20H
00719      LT7S: EQU #-T7S
00720      T5S:  DB 2
00721      DB 8
00722      DB 4
00723      T7K:  DB 61H
00724      DB 62H
00725      DB 63H
00726      DB 64H
00727      DB 65H
00728      DB 66H
00729      DB 67H
00730      DB 68H

```

B60B	69	00731	DB	69H
B60C	6A	00732	DB	6AH
B60D	6B	00733	DB	6BH
B60E	6C	00734	DB	6CH
B60F	6D	00735	DB	6DH
B610	6E	00736	DB	6EH
B611	6F	00737	DB	6FH
B612	70	00738	DB	70H
B613	71	00739	DB	71H
B614	72	00740	DB	72H
B615	73	00741	DB	73H
B616	74	00742	DB	74H
B617	75	00743	DB	75H
B618	76	00744	DB	76H
B619	77	00745	DB	77H
B61A	78	00746	DB	78H
B61B	79	00747	DB	79H
B61C	7A	00748	DB	7AH
		00749	LT7K: EQU	#-T7K
B61D	03	00750	DB	3
B61E	19	00751	DB	19H
B61F	0E	00752	DB	0EH
B620	09	00753	DB	9
B621	01	00754	DB	1
B622	0D	00755	DB	0DH
B623	1A	00756	DB	1AH
B624	14	00757	DB	14H
B625	06	00758	DB	6
B626	0B	00759	DB	0BH
B627	0F	00760	DB	0FH
B628	12	00761	DB	12H
B629	1C	00762	DB	1CH
B62A	0C	00763	DB	0CH
B62B	18	00764	DB	18H
B62C	16	00765	DB	16H
B62D	17	00766	DB	17H
B62E	0A	00767	DB	0AH
B62F	05	00768	DB	5
B630	10	00769	DB	10H
B631	07	00770	DB	07H
B632	1E	00771	DB	1EH
B633	13	00772	DB	13H
B634	1D	00773	DB	1DH
B635	15	00774	DB	15H
B636	11	00775	DB	11H
B637		00776	T7G: DB	'ABCDEFGHJKLMNPOQRSTUVWXYZ'
		00777	LT7G: EQU	#-T7G
B651	03	00778	DB	3
B652	19	00779	DB	19H
B653	0E	00780	DB	0EH
B654	09	00781	DB	09H
B655	01	00782	DB	01H
B656	0D	00783	DB	0DH
B657	1A	00784	DB	1AH
B658	14	00785	DB	14H
B659	06	00786	DB	06
B65A	0B	00787	DB	0BH
B65B	0F	00788	DB	0FH
B65C	12	00789	DB	12H
B65D	1C	00790	DB	1CH
B65E	0C	00791	DB	0CH
B65F	18	00792	DB	18H
B660	16	00793	DB	16H
B661	17	00794	DB	17H
B662	0A	00795	DB	0AH
B663	05	00796	DB	5
B664	10	00797	DB	10H
B665	07	00798	DB	7
B666	1E	00799	DB	1EH
B667	13	00800	DB	13H
B668	1D	00801	DB	1DH
B669	15	00802	DB	15H
B66A	11	00803	DB	11H
B66B		00804	T7Z: DB	' !"()+,-./0123456789:=-?'
		00805	LT7Z: EQU	#-T7Z
B682	04	00806	DB	4
B683	1D	00807	DB	1DH
B684	05	00808	DB	05H
B685	0F	00809	DB	0FH
B686	12	00810	DB	12H
B687	11	00811	DB	11H
B688	0C	00812	DB	0CH

5. Anhang

E689	03	00813	DB	03H	
E68A	1C	00814	DB	1CH	
E68B	1D	00815	DB	1DH	
E68C	16	00816	DB	16H	
E68D	17	00817	DB	17H	
E68E	13	00818	DB	13H	
E68F	01	00819	DB	01H	
E690	0A	00820	DB	0AH	
E691	10	00821	DB	10H	
E692	15	00822	DB	15H	
E693	07	00823	DB	07	
E694	06	00824	DB	6	
E695	18	00825	DB	18H	
E696	0E	00826	DB	0EH	
E697	1E	00827	DB	1EH	
E698	19	00828	DB	19H	
		00829	ORG	0F800H	
		00830	;-----		
F800	46	00831	CTAB: DB	46H	;MPM LESEN
F801	01	00832	UNIT: DB	1	;LAUFWERK B VOREINGEST.VORL.
F802	00	00833	TRACK: DB	0	
F803	00	00834	HEAD: DB	0	
F804	01	00835	SECTR: DB	1	
F805	01	00836	N: DB	1	;N=1=256BYTES
F806	0F	00837	EOT: DB	15	;15 SECTOREN JE SPUR
F807	1B	00838	GPL: DB	1BH	;LUECKE
F808	FF	00839	DTL: DB	0FFH	
F809		00840	RESLT: BER	7	;RESULT TAB P.FDC
F810	1E	00841	DZEIL: DB	30	;DRUCKER 30 ZEILEN/SEITE
F811	00	00842	MERK: DB	0	;UMSCHALTZ. FERNSCHR.
F812	01	00843	SEC1: DB	1	;BDOS-SEKTOR
F813	8000	00844	DMA: DA	80H	;BDOS-DMA
F815		00845	DIREF: BER	128	;INHALTSVERZ.DISKETTE
F895		00846	ALLO: BER	31	
F8B4		00847	CHK0: BER	16	
F8C4		00848	ALL1: BER	31	
F8E3		00849	CHK1: BER	16	
F8F3		00850	BND		

EBAS-88		CROSS-REFERENCE			PN: CB
CFDC	007C	E3F6	E45E	E47E	E495 E4CB
DFDC	007D	E3FC	E44B	E49E	E4B2
TC	0078	E450			
PIOAD	00FC	E5F0			
NDISK	0002	E23F			
CR	000D	E5FE			
LF	000A	E556	E5FD		
ROM	E200	0000	0000	0000	
BDOS	D400	E2A9	E335		
CCP	CC00	E2E1	E346		
RDSK	7000	E319	E50D		
MCSTS	F003	E206			
CIO	F13C	E209	E2BE		
MCO	F009	E20C			
MWO	F00F	E212			
MRI	F012	E215			
RESTR	F400	E241	E2B2	E2F5	E3E4 E48C
ZS	F089	E5B5	E5F2	E5F5	
STR	F128	E2BB			
WBOTE	E203	E2A0			
HOME	E233	E218			
SDISK	E238	E21B	E2D5		
DISKS	E240	E329			
LSTS	E253	E22D			
STRAN	E254	E230			
STRN1	E25E	E258			
SSEC	E266	E221	E2E8		
STRCK	E26B	E21E	E236	E2DC	E30E
*SDMA	E270	E224	E2ED	E33F	
MSG5	E275	E2B8			
BOOT	E29B	E200			
WBOOT	E2CD	E203			
LOAD1	E2E4	E306	E314		
IRDSK	E316	E2C3			
GOCFM	E33C	E300			
WRITE	E349	E22A			
READ	E353	E227			
RFLOP	E35D	E2F0	E357		
RFLP	E39B	E366	E370	E378	E37E E385 E388
B2	E3A9	E3A4			

WFLOP	E3B7	E34D					
B3	E3CA	E3C5					
B4	E3D5	E3BE					
TRANS	E3D9	E39B	E3BA				
B1	E3E9	E3E0					
INIFD	E3F2	E2D0	E3F4	E3FE			
SPEZI	E400	E3FA					
RECAL	E409						
SENSE	E40F	E420	E47B				
STAB	E423	E400					
W8272	E425	E3D2					
R8272	E42C	E3ED					
RWIT	E431	E42A					
RWLOP	E43B						
RWOP	E43D	E457					
ERR	E459						
RW	E45C	E44D					
RW1	E45E	E461	E468				
MODE	E466	E434					
SDS	E46B	E486					
SEEK	E475	E438					
SKBSY	E47E	E482					
RDY	E485	E40C	E445	E478			
WCOM	E48F	E412	E46E				
WCOM1	E492	E406	E49B	E4A2			
DELAY	E4A5	E492	E4AC				
DEL1	E4A8	E4A8					
RBYTE	E4AC	E415	E41B	E471	E4B7	E4C1	
RRSLT	E4B5	E452					
RESL1	E4C1	E4C6					
IRDY	E4CB	E4AF	E4CE				
ERM	E4D4						
WRDSK	E4D5	E34F					
RDSK1	E4DE	E4EE					
RRDSK	E4E5	E359					
BDERR	E4F0	E4D8	E4B8				
RADR	E4F3	E4D5	E4E5				
RADR1	E502	E503					
DPBAS	E515	E24A					
DPHA	E515						
DPHR	E525						
DPBR	E535	E52F					
DPBF	E544	E51F					
DRUCK	E553	E20F	F9F3				
CTAB	F800	E362	E431	E441	E48F		
UNIT	F801						
TRACK	F802	E26B	E4F5				
HEAD	F803						
SECTR	F804	E3EA					
N	F805						
EOT	F806						
GPL	F807						
DTL	F808						
RESLT	F809	E369	E4BA				
DZEL	F810	E5AF					
MERK	F811	E562					
SEC1	F812	E261	E266	E381	E3DB	E4F9	
DMA	F813	E270	E38D	E3A9	E3CA	E4DB	E4EA
DIREF	F815	E51D	E52D				
ALLO	F895	E523					
CHK0	F8B4	E521					
ALL1	F8C4	E533					
CHK1	F8E3	E531					
FDMA	F8F3	E38A	E39F	E3A6	E3C0	E3C7	E448
ZAUS	E56C	E566					
KODE	E574	E55C					
GF	E5AA	E580	E58C	E598	E5A4		
ZEPR	E5AF	E559					
Z2	E5B5	E5B8					
LCRLF	E5BD						
L1	E5C4	E5C9					
ART	E5CE	E55F					
KB	E5DF	E5D3					
GB	E5E5	E5D7					
FSA	E5EA	E569	E56D	E5C6			
FSA1	E5F0	E5F9					
T7S	E5FD	E576	E600				
LT7S	0003	E579					
T5S	E600						
T7K	E603	E582	E61D				
LT7K	001A	E585					
T51	E61D						

5. Anhang

T7G	B637	B58E	B651
LT7G	001A	B591	
T53	B651		
T7Z	B66B	B59A	B682
LT7Z	0017	B59D	
T52	B682		

Maschinencode

:10120000C39BE2C3DE2C303F0C33CF1C309F0C307
:1012100053E5C30FF0C312F0C333E2C338E2C36B2C
:10122000E2C366E2C370E2C353E3C349E3C353E2DC
:10123000C354E20100001833210000C5060079FB06
:1012400001D200F46960292929291115E519320410
:1012500000C1C9C906007AB72004606923C9EB0937
:101260007E3212F86FC92112F871C92102F871C9D2
:10127000ED4313F8C90A0D593233564F2035364B41A
:1012800002043502F4D20322E320D0A52414D4449F9
:10129000534B203F202859290D0A003EC33200003D
:1012A0002103E22201003205002106D42206003289
:1012B00038002100F4223900217E52C2D2EF1CD3C1F
:1012C000F1FE59CC1E3AF32030032040031800046
:1012D000CDFE2E30E00CD38E2062C0B00CD6BE21607
:1012E000012100CC5D5E54ACD66E2C1C5CD70E28D
:1012F000CD5DE3FE00C400F4E111800019D1C10509
:10130000283A147AFE2138CD16010CC5D5E5CD6BB0
:10131000E2E1D1C118CE2100E2110070013300EDED
:10132000B02103702201003B023240E22106622217
:1013300006003EC3772106D4220762C9018000CD92
:1013400070E23A04004FC300CC3A0400B72868CDDD
:10135000D5E4C93A0400B72804CDE5E4C9E5D5C50C
:101360003E462100F8BE2033231109F81AE603BED9
:101370002029231313131ABE202123131ABE201B66
:10138000233A12F81F3014BE20112173F9ED5B13B4
:10139000F8018000EDB0C1D1E1AF9C9CD983F521AD
:1013A000F3F8CB4820032173F9ED5B13F8018000BB
:1013B000EDB0F1C1D1E1C9E5D5C5CD983B720156F
:1013C00011F3F8CB4820031173F92A13F8018000B8
:1013D000EDB0CD25E4C1D1E1C906003A12F8B71F3E
:1013E0003007C601DC00F406FFC53204F8CD2CE45A
:1013F000C1C9060010FEDB7CFE802B04DB7D18F2EC
:101400002122E4010303CD92E4010702CD85E4012A
:101410000801CD8FE4CDACE447FE80C4ACE4CB68DA
:1014200028EDC91F3311EDA33E45180511EDA23B6D
:10143000463200F8ED5366E4CD75E4060AC5F306BE
:10144000093A00F84FCD85E421F3F80E7DCD5CB438
:10145000D378CDB5E4C1C810E43E01C90600DB7CF9
:101460000730FB0707D0EDA220FAC9010402CD8F9D
:10147000E4CDAE4C9010F03CD85E4CDOFE4DB7C02
:10148000E60F20FAC9C5CD6BE4C1CB6FCC00F421C7
:1014900000F8CDA5E4DB7CE6C0FE8020F579D37DA5
:1014A000234E10EEC9C5060F10FEC1C9CDA5E4CD6F
:1014B000CBE4DB7DC90606CDACE42109F877E6C0B4
:1014C0004FCDAE4237710F979B7C9DB7C0730FB4B
:1014D000E68007D8FFCDF3E43816EB2A13F8018035
:1014E00000EDB0AFC9CDF3E43806ED5B13F818EEAC
:1014F0003E01C916003A02F85F3A12F84F21000087
:10150000061A1910FD0929292929292929010070FC
:10151000093EBFBCC90000000000000000015F844EF
:10152000E5B4F895F800000000000000000015F8355B
:10153000E5E3F8C4F81A0003070014001F00800058
:10154000100000002000040F0149003F008000103F
:10155000000300C5D5E53E0AB9CCAPE5CD74E5CDB5
:10156000CEB52111F8BE280477CDEAE579CDEAE58C
:10157000E1D1C1C9C57921FDE5010300C5D1EDB1B6
:10158000282B2103E6011A00C5D1EDB1281C213716
:10159000B6011A00C5D1EDB12810216BE601170054
:1015A000C5D1EDB128043E04C1C9192B7EC1C921A2
:1015B00010FB8C50640CD89F010FBC135C0F5C53E19
:1015C0001E7706043E02CDEAE510F9C1F1C9F5C562
:1015D00079FE60300AFE40300CC1F14F3E1BC9C19C
:1015E000F14F3E1FC9C1F14FAFC9C50608F66017DC
:1015F000D3FCCD89F0CD89F01F10F5C1C90A0U20AB

:101600000208046162636465666768696A6B6C6D91
:101610006B6F707172737475767778797A03190EBC
:1016200009010D1A14060B0F121C0C1816170A05C7
:1016300010071E131D1511414243444546474849B2
:101640004A4B4C4D4E4F50515253545556575859B2
:101650005A03190E09010D1A14060B0F121C0C184F
:1016600016170A0510071E131D15112021222829FF
:101670002B2C2D2E2F303132333435363738393A42
:101680003D3F041D050F12110C031C1D16171301FD
:101690000A10150706180E1E19FFFFFFFFFFF8B8

5.2.9. Programm Monitor für Adresse 0F000H (Ausdruck von Adresse 7000H)

[illegible]

5.2.10. Programm zum Formatieren von Disketten

```

'DISKETTEN-INIT'      MACRO-80 3.36  17-Mar-80-  PAGE 1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

.Z80
TITLE 'DISKETTEN-INIT'

;PROGRAMM ARBEITET IN VERBINDUNG MIT MONITOR

;VEREINBARUNGEN
CFDC EQU 7CH          ;FLOPPY-CONTROLLER
TC EQU 7DH
FMTAB EQU 300H        ;ID-TABELLE
CTAB EQU OF800H
RESLT EQU CTAB+9
BDOS EQU 5
CI EQU OE209H         ;MONITORADRESSEN
RESTR EQU OF400H

ASEG ORG 100H
;-----
;HAUPTPROGRAMM
;-----
FORM: LD DE,TEXT      ;TEXTAUSGABE
      LD C,9
      CALL BDOS
      CP CI
      JZ 79H
      NZ,0
      CALL RECAL
      C,0
      LD B,1
      LD E,0
      LD A,C
      LD (CTAB+2),A
      PUSH BC
      CALL SEEK
      BC
      LD HL,FMTAB+10H*-1
      LD A,10H
      LD (HL),1
      LD HL
      DEC (HL),A
      LD HL
      DEC (HL),E
      LD HL
      DEC (HL),C
      LD HL
      DEC A
      JR NZ,FORM2
      LD (HL),OESH

FORM2: LD HL,FMTAB+10H*-1
      LD A,10H
      LD (HL),1
      LD HL
      DEC (HL),A
      LD HL
      DEC (HL),E
      LD HL
      DEC (HL),C
      LD HL
      DEC A
      JR NZ,FORM2
      LD (HL),OESH

;HARTEN AUF CONSOLEINGABE
;KLEINES Y
;ENDE DES FORMATIERENS
;KOPF AUF SPUR 0
;SPUR
;SEKTOR
;KOPFAUSWAHL F. BEIDSEITIGE LAUFNERKE
;SPUR EINSTELLEN
;ANZAHL SEKTOREN
;SETZEN N
;SET R
;SET H
;SET C
;AUFSCHREIBENDES BYTE

```

```

51 0135 28 36 35 HL (HL),53 ;GAP (LUECKENLAENGE)
52 0136 28 36 35 HL (HL),53 ;GAP (LUECKENLAENGE)
53 0138 28 36 10 HL (HL),10H ;EOT (LETZTE SEKTOR-NR.)
54 0139 28 36 10 HL (HL),10H ;EOT (LETZTE SEKTOR-NR.)
55 013B 28 36 01 HL (HL),1 ;N, 1 = 256 BYTES/SEKTOR
56 013C 28 36 01 HL (HL),1 ;N, 1 = 256 BYTES/SEKTOR
57 013E 28 36 00 HL (HL),0
58 013F 28 36 00 HL (HL),0
59 0141 28 36 00 HL (HL),0
60 0142 C5 PUSH BC ;ANZAHL KOMMANDOBYTES F. FDC
61 0143 0E 06 LD B,6 ;KOMMANDO FORMAT A TRACK
62 0145 0E 4D LD C,4DH ;IN FDC SCHREIBEN
63 0147 CD 01F2 CALL MCON1 ;AUSFUEHRUNG
64 014A CD 01A0 CD 01A0 OUT (TC),A ;ENDE-IMPULS
65 014D D3 7B CALL RRS1T
66 014F CD 0215 POP BC
67 0152 C1 JP NZ,RESTR ;FEHLER
68 0153 C2 F400 INC C
69 0154 0C LD A,C
70 0157 79 LD C,40
71 0158 FE 28 CP NZ,FORH0
72 015A 20 B9 JR FORH
73 015C 1B A2 JR
74
75 015E 46 6F 72 6D TEXT: DEFM 'Format Y23V0',0DH,0AH
76 0162 61 74 20 59
77 0166 32 33 56 4F
78 016A 0D 0A
79 016C 34 30 20 54 DEFM '40 TRACKS, 16 SECTORS, 256 BYTES',0DH,0AH
80 0170 52 41 43 4B
81 0174 53 2C 20 31
82 0178 36 20 53 45
83 017C 43 54 4F 52
84 0180 53 2C 20 32
85 0184 35 36 20 42
86 0188 59 54 45 53
87 018C 0D 0A DEFM 'Diskette ? (y)',0DH,0AH,'s'
88 018E 44 69 73 6B
89 0192 45 74 74 65
90 0196 20 3F 20 20
91 019A 28 79 29 0D
92 019E 0A 24
93
94
95
96 01A0 0E 7D ;FORMATIERSCHLEIFE FUER EINE SPUR
97 01A2 06 40 LD C,DFDC
98 01A4 DE 7C LD B,10H*4 ;ANZAHL ID
99 01A6 07 RLCA IN A,(CFDC)
100 01A7 30 FB JR NC,F0RM3 ;RQM-TEST
101 01A9 07 RLCA ;DIO
102 01AA 07 RET ;NON-DMA
103 01AB D0 RET ;FERTIG ?
104 01AC ED A3 OUTI NC
105 01AE 20 F4 JR NZ,F0RM3
106 01B0 C9 RET
107

```

108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

```

162      020C      CD 0205      RBYTE: CALL DELAY      ;1 BYTE LESEN
163      020F      CD 0228      CALL IRDY
164      0210      DB 7D      IN A,(DFDC)
165      0212      C9      RET
166
167      0215      06 06      RSLT: LD B,6      ;LESE 7 RESULT BYTES
168      0217      CD 020C      CALL RBYTE
169      021A      21 F809      LD HL,RSLT
170      021D      77      LD M,A
171      021E      E6 C0      AND OCOH      ;ERROR?
172      0220      4F      LD C,A
173      0221      CD 020C      RSL1: CALL RBYTE
174      0224      23      LD HL
175      0225      77      LD M,A
176      0226      10 F9      DJNZ RSL1
177      0228      79      LD A,C
178      0229      B7      OR A
179      022A      C9      RET
180
181      022B      DB 7C      IRDY: IN A,(DFDC)      ;BEREIT F.DATENEINGABE ?
182      022D      07      RLCA
183      022E      30 FB      JR NC,IRDY
184      0230      E6 80      AND 80H
185      0232      07      RLCA
186      0233      DB      RET C
187
188      0234      C3 F400      ;FEHLERBEHANDLUNG: SPRUNG IN MONITOR
189      JP RSTR
190
191      ;
192
193
194
195

```

END

'DISKETTEN-INIT' MACRO-80 3.36 17-Mar-80 PAGE 5

Macros:

```

Symbols:
BDOS , 0005      CFDC      007C      E209      CTAB      F800
DEL1  0208      DELAY      007D      F4      01A0
FMTAB 0300      FORM      0100      FORM2      0127
FORM3 01A4      IRDY      0115      RDY      01E5
RECAL 01B1      RSL1      020C      RSTR      F400
RSLT  0215      SDS      0221      SEEK      01B7
SKBSY 01DE      TC      007B      TEXT      015E
WCOM1 01F2      0078      WCOM      01EF

```

No Fatal error(s)

5.3. Abkürzungen

CPU	– central processing unit, Zentrale Verarbeitungseinheit
CRT	– cathode ray tube, Kathodenstrahlröhre, Bildröhre
RAM	– random access memory, Schreib-/Lesespeicher
ROM	– read only memory, Lesespeicher
PROM	– programmable ROM, programmierbarer Lesespeicher
EPROM	– erasable PROM, löschbarer PROM
SRAM	– statischer RAM
DRAM	– dynamischer RAM
Byte	– 8 bit
Chip	– Halbleiterscheibe, Schaltkreis
PIO	– parallel in/out, Ein-/Ausgabeschaltkreis
UART	– universal asynchron receiver/transmitter, universeller Ein-/Ausgabeschaltkreis für asynchrone, serielle Datenübertragung
USART	– universal synchron/asynchron receiver/transmitter, universeller Ein-/Ausgabeschaltkreis für synchrone und asynchrone serielle Datenübertragung
RX	– Empfänger
TX	– Sender
CRC	– cycling redundancy code, Datensicherungsverfahren
ZVE	– Zentrale Verarbeitungseinheit

Anschlußbezeichnungen

A0 bis A15	Adreßleitung
A0 bis A7	Portleitung
AD0 bis AD15	Adreß-/Datenleitung
B0 bis B7	Portleitungen
C, CP	Takt
CE	Chipenable, Schaltkreis-Freigabe
CS	Chipselekt, Schaltkreis-Auswahl
D0 bis D12	Datenleitungen
DI0 bis DI8	Dateneingänge
DO0 bis DO8	Datenausgänge
O0 bis O8	Ausgänge
PA0 bis PC7	Port A bis C Leitungen
PO0 bis P37	Port 1 bis 3 Leitungen

Steuerleitungen

R/W	lesen/schreiben
RD	lesen
WR	schreiben
IORQ	Ein-/Ausgabeanforderung
MREQ	Speicheranforderung
RTS	request to send, Sende-anforderung
DTR	data terminal ready, Fertigmeldung
CTS	clear to send, Senderfreigabe
B/ $\bar{A}$	Kanalauswahl
C/ $\bar{D}$	Auswahl Steuerung/Daten
DCD	data carrier detect, Empfänger-Freigabe-Eingang (SIO)
TxC	transmitter clock, Sendertakt
RxC	receiver clock, Empfängertakt
NC	no connection, nicht angeschlossen
RAS	row adress strobe, Zeilenadressenabtastung
CAS	column adress strobe, Spaltenadressenabtastung
RFSH	Auffrischen dynamischer Speicher
BUSRQ	Busanforderung
BUSAK	Busfreigabe durch CPU
M1	Kennzeichen Befehlshole-Zyklus
INT	Interruptanforderung
OE	Output enable, Ausgang durchschalten
U_{cc}	+5 V
U_{ss}	0 V
U_{pp}	22 bis 26 V (Programmierspannung)

5.4. Schaltkreisdaten, Anschlüsse, Vergleichsliste

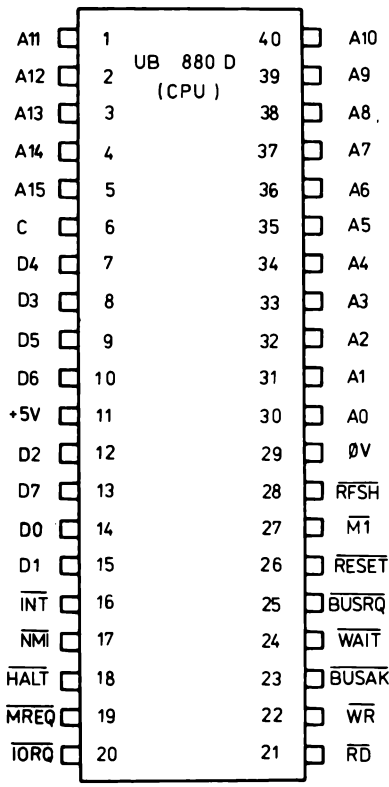
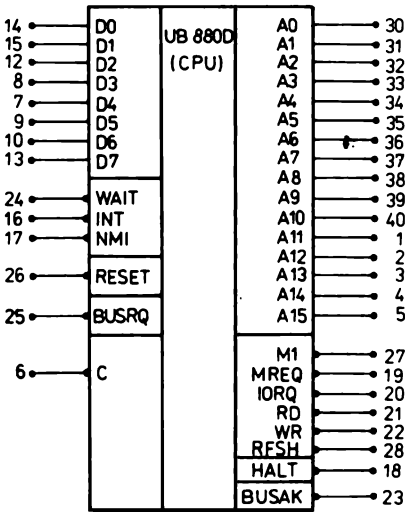


Bild 5.1 Anschlußbelegung und Schaltzeichen UB 880 D (CPU)

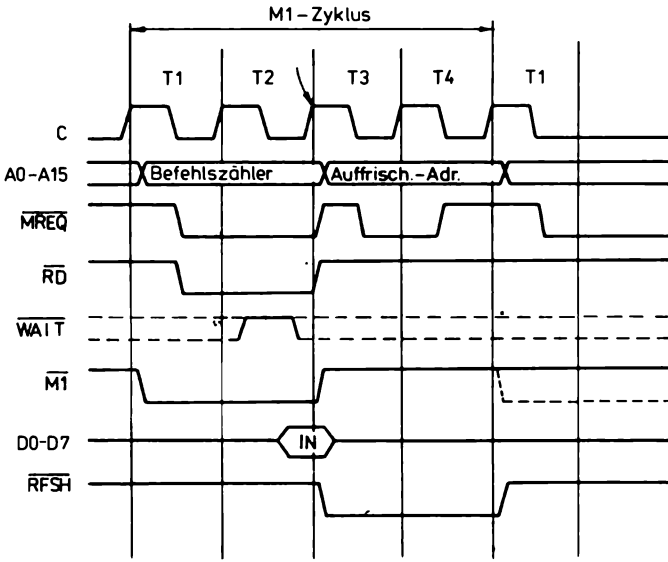


Bild 5.2 Aufruf Operationscode

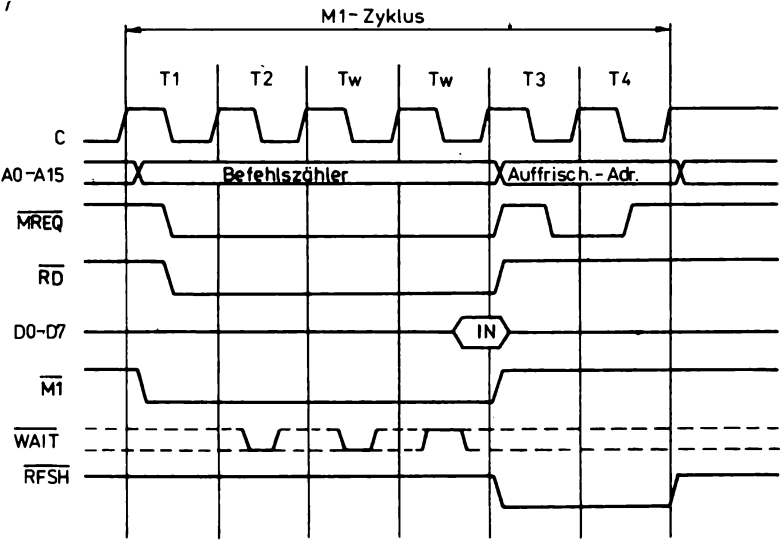


Bild 5.3 Aufruf Operationscode mit WAIT

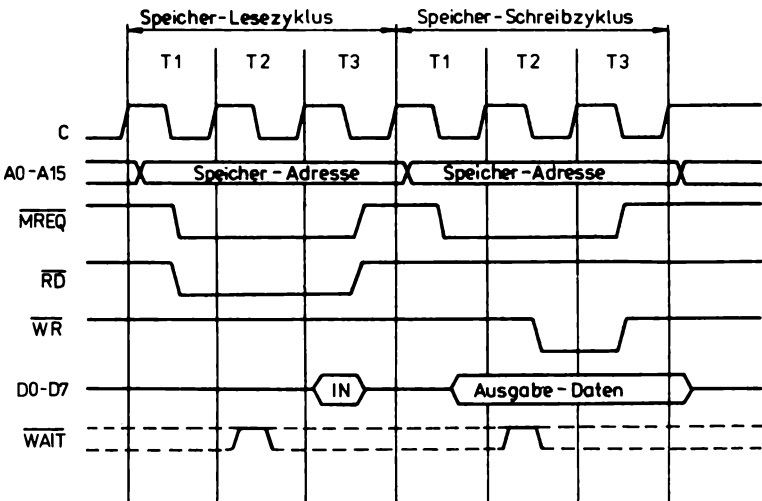


Bild 5.4 Speicher-Lese- oder Schreibzyklus

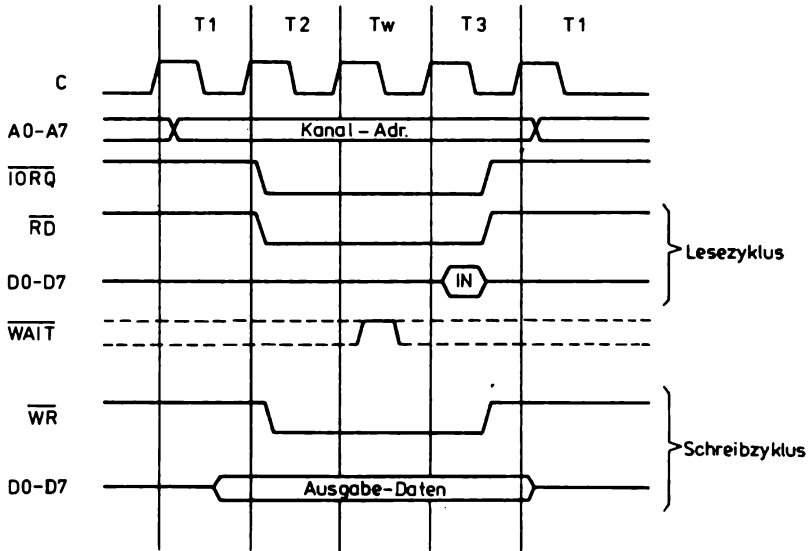


Bild 5.5 Eingabe- oder Ausgabezyklus

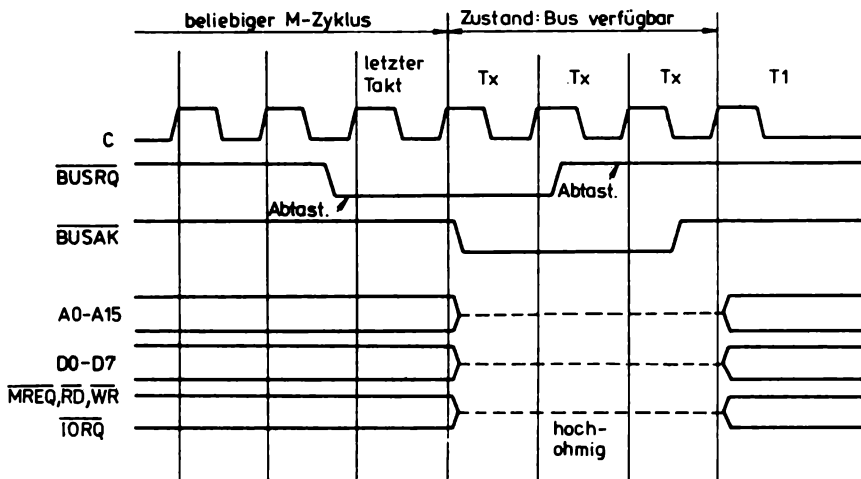


Bild 5.6 Bus-Anforderungs-/Bus-Bestätigungszyklus

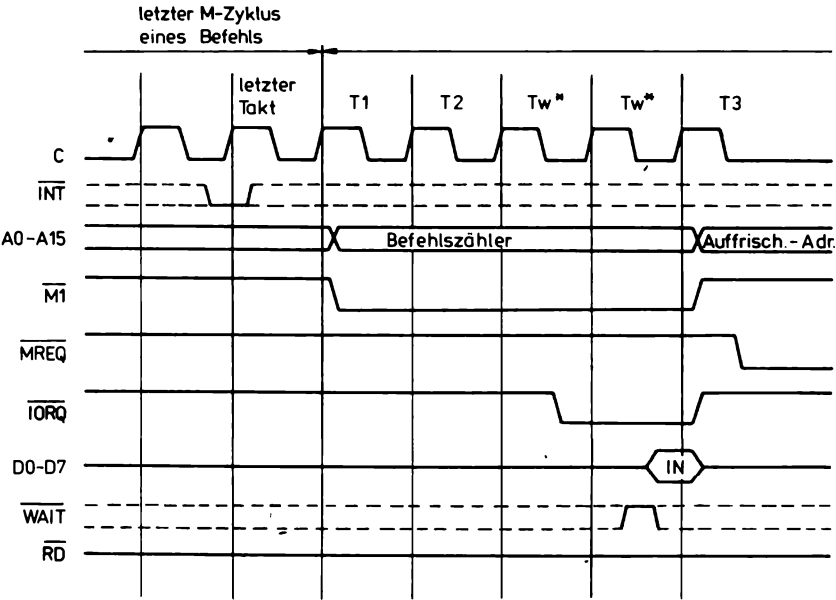


Bild 5.7 Interruptzyklus

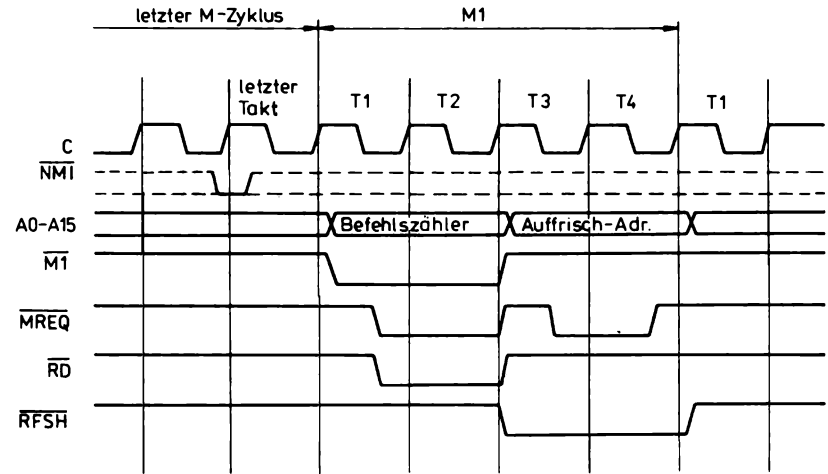


Bild 5.8 Nichtmaskierbarer Interrupt

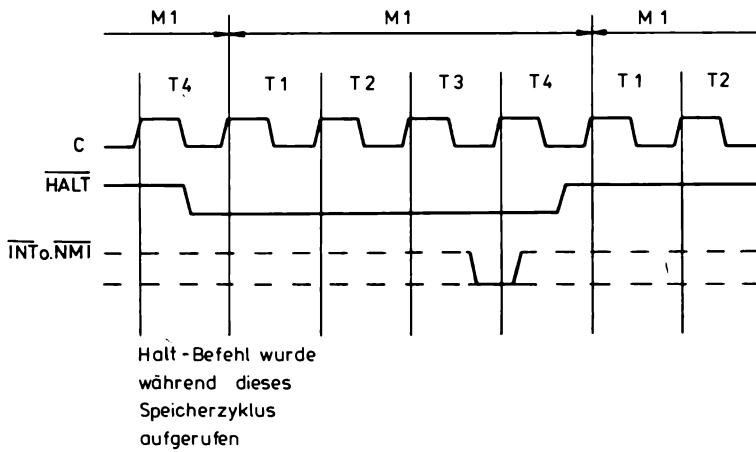


Bild 5.9 Halt-Ablauf

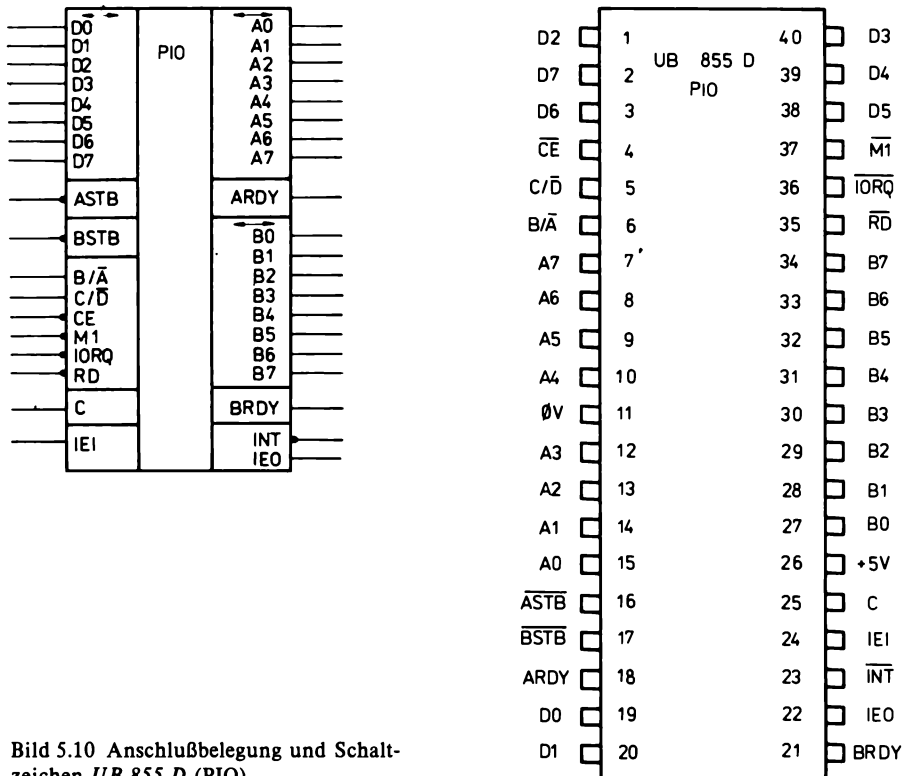
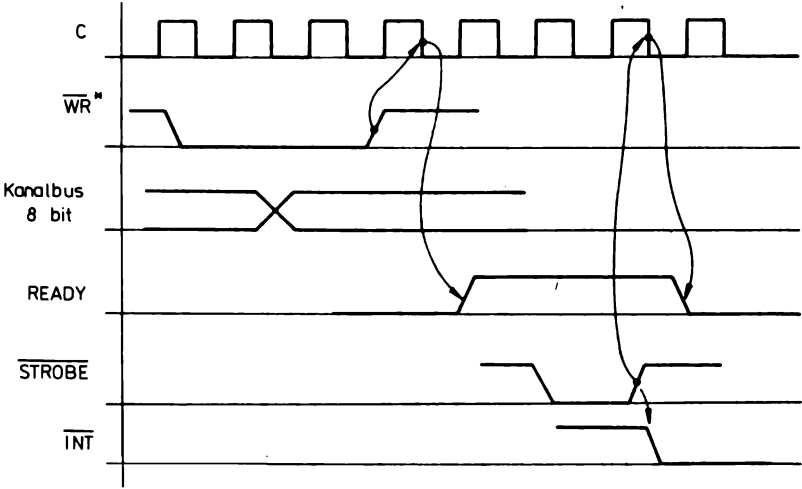
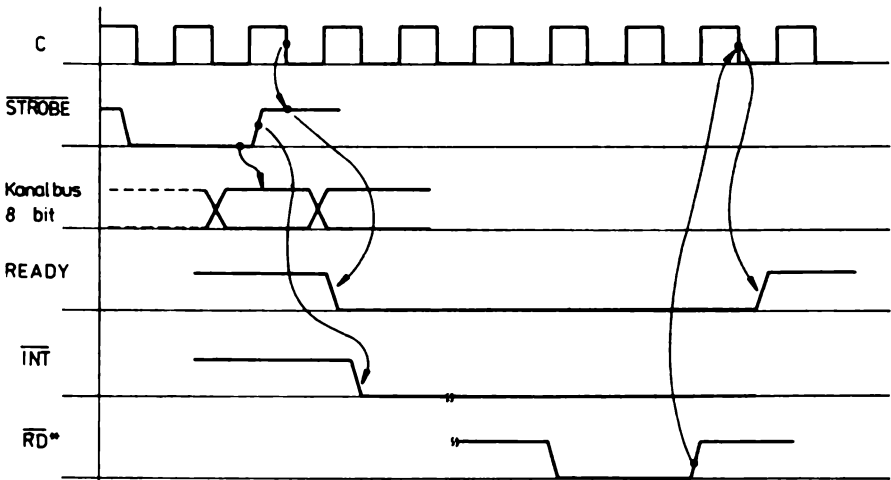


Bild 5.10 Anschlußbelegung und Schaltzeichen UB 855 D (PIO)



$$\overline{WR}^* = RD \cdot \overline{CE} \cdot \overline{C/D} \cdot \overline{IORQ} \cdot M1$$

Bild 5.11 Betriebsart 0 mit Quittung



$$\overline{RD}^* = \overline{RD} \cdot \overline{CE} \cdot \overline{C/D} \cdot \overline{IORQ}$$

Bild 5.12 Betriebsart 1 mit Quittung

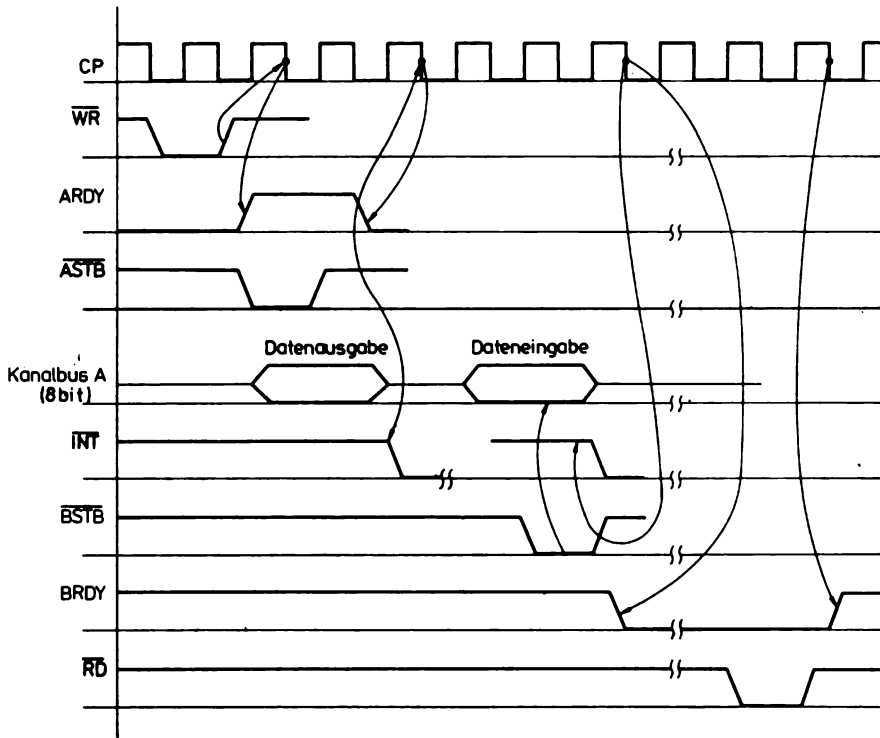


Bild 5.13 Betriebsart 2

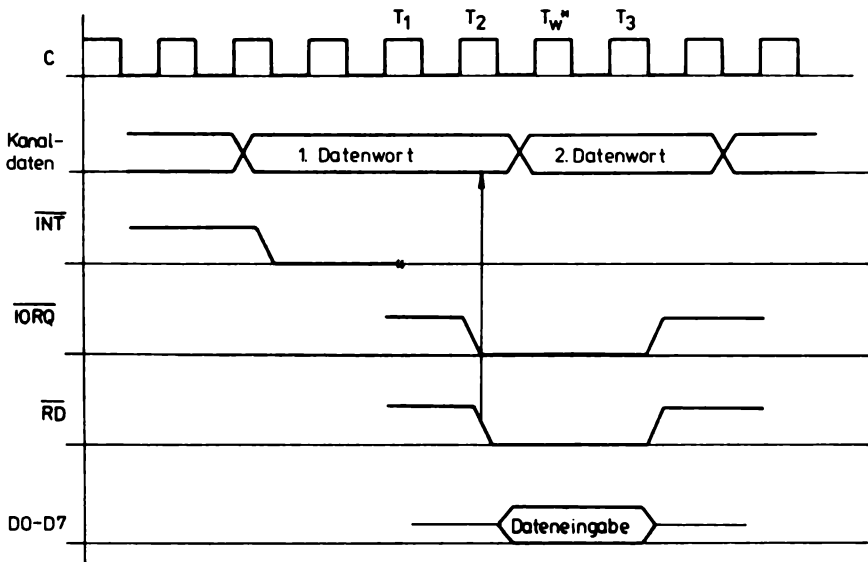


Bild 5.14 Betriebsart 3

1. Datenwort wird auf
CPU - Datenbus gelegt

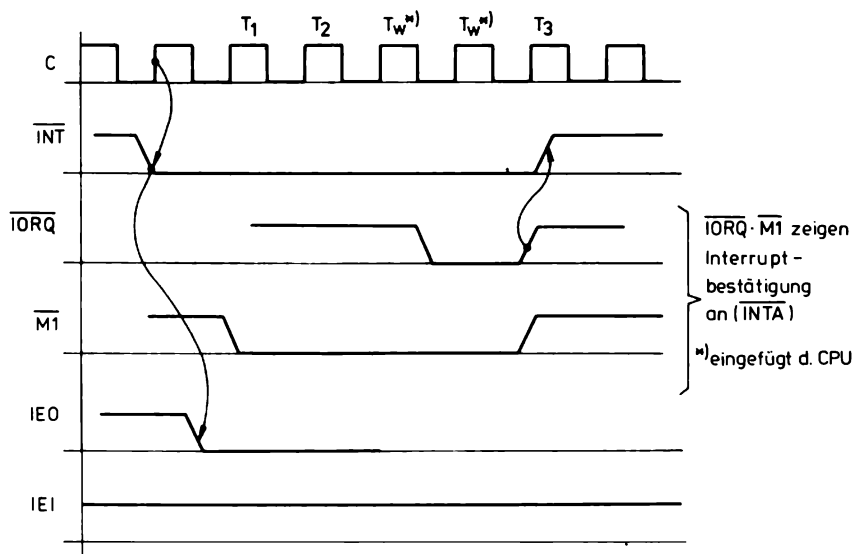


Bild 5.15 Interrupt-Quittungs-Zyklus

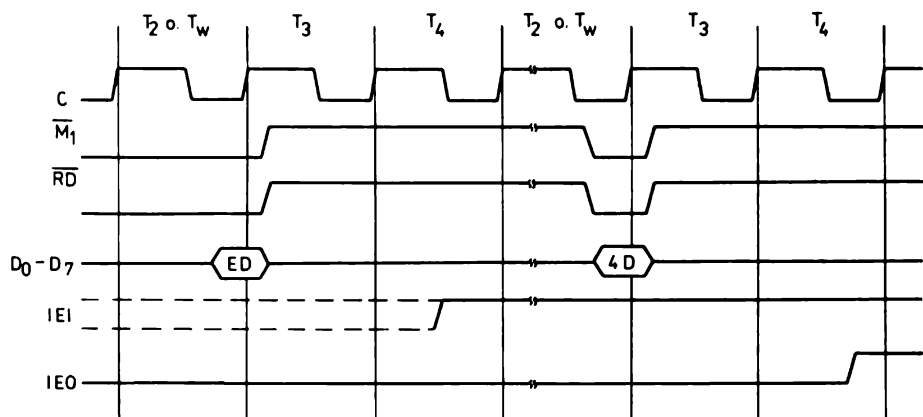


Bild 5.16 RETI-Zyklus

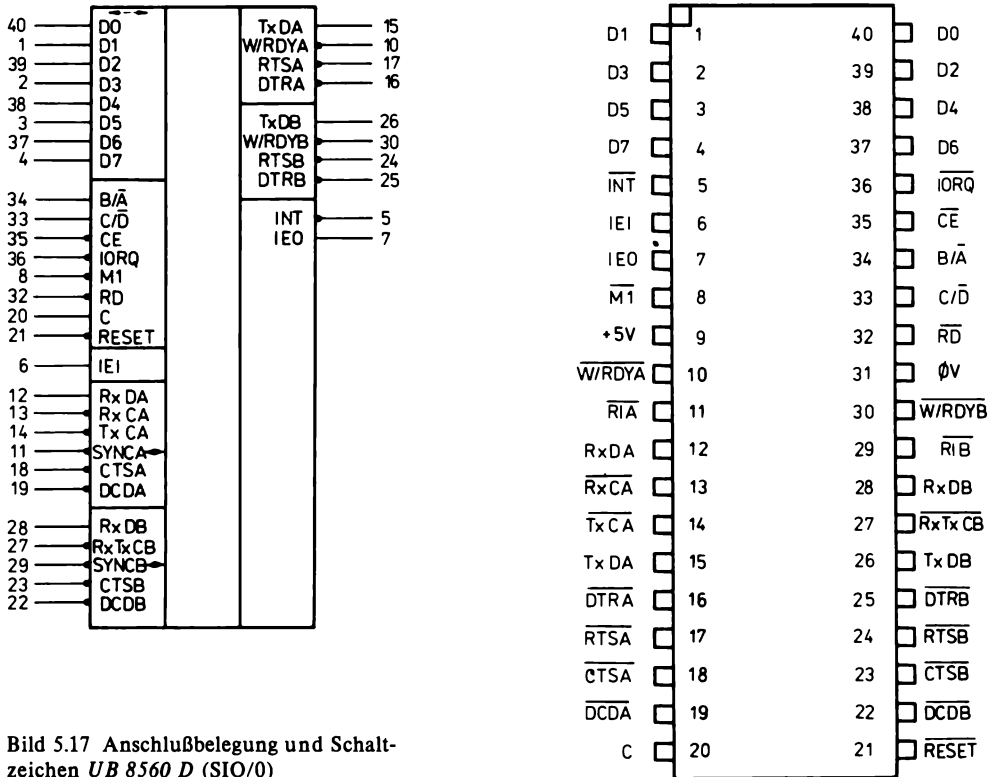


Bild 5.17 Anschlußbelegung und Schaltzeichen *UB 8560 D* (SIO/0)

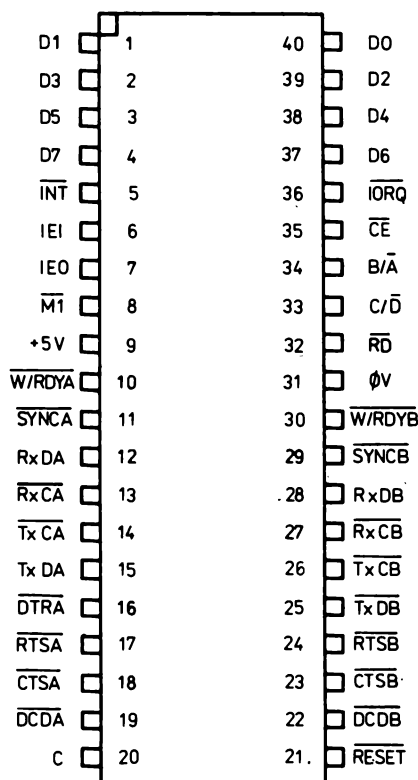


Bild 5.18 Anschlußbelegung UB 8561 D (SIO/1)

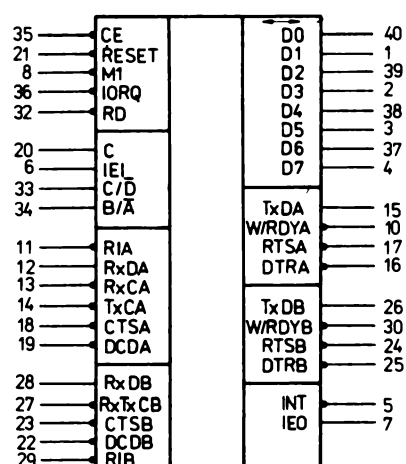
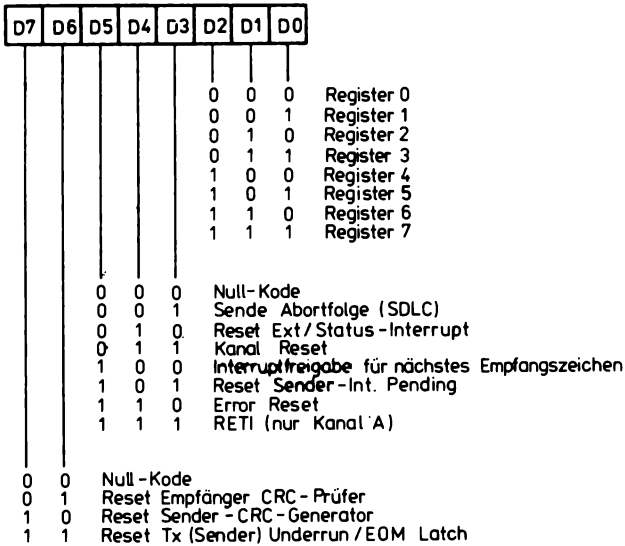


Bild 5.19 Anschlußbelegung und Schaltzeichen UB 8563 D

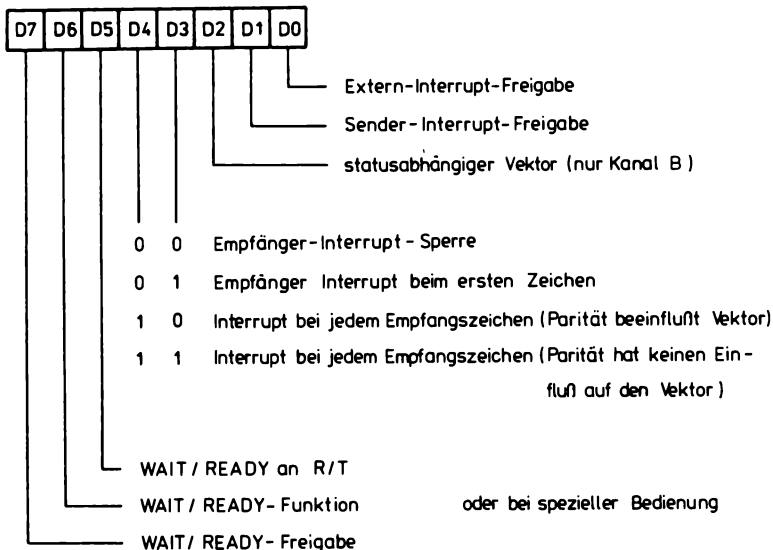
Bild 5.20

Darstellung der Schreibregister

Schreibregister 0

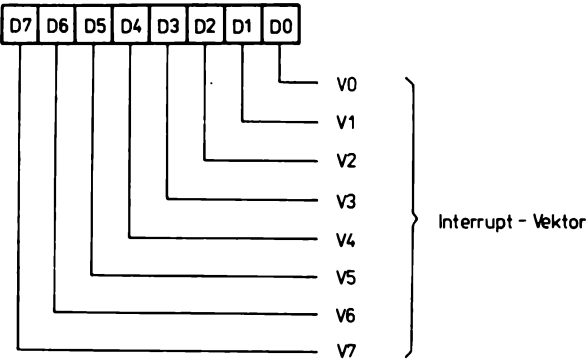


Schreibregister 1 (Write-Register 1)

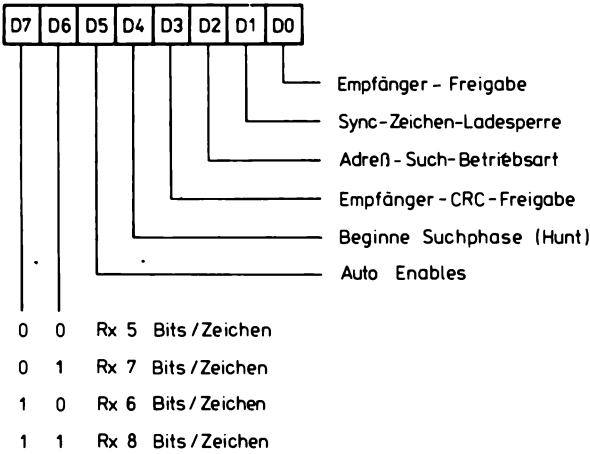


5. Anhang

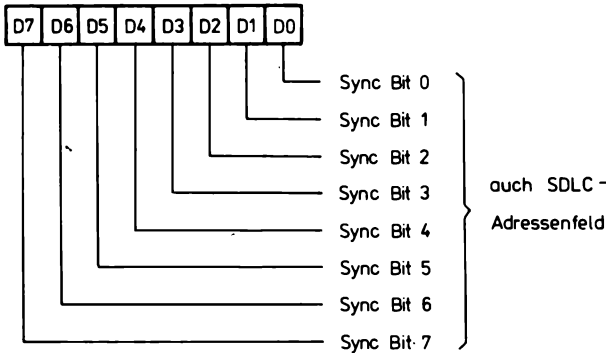
Schreibregister 2 (nur Kanal B) (Write-Register 2)



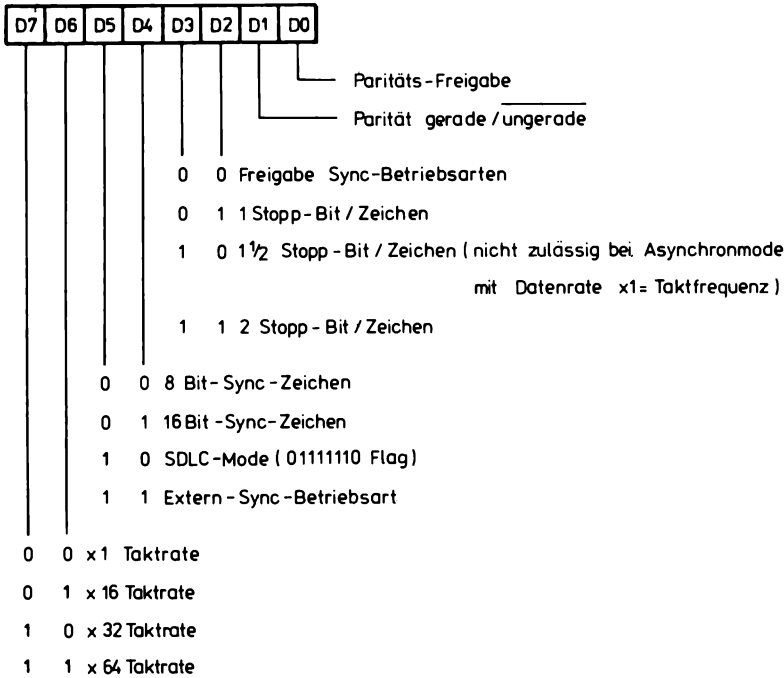
Schreibregister 3 (Write-Register 3)



Schreibregister 6 (Write-Register 6)



Schreibregister 4 (Write-Register 4)



Schreibregister 5 (Write-Register 5)

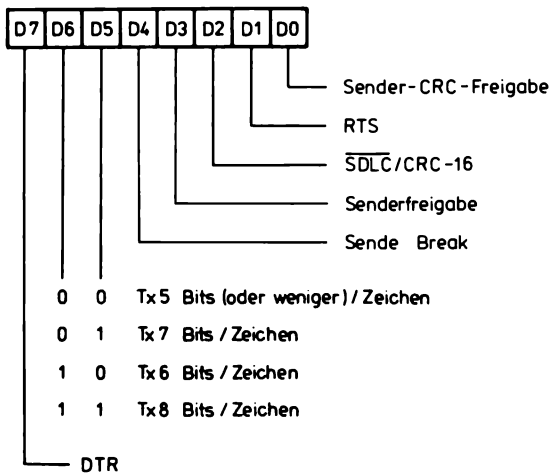
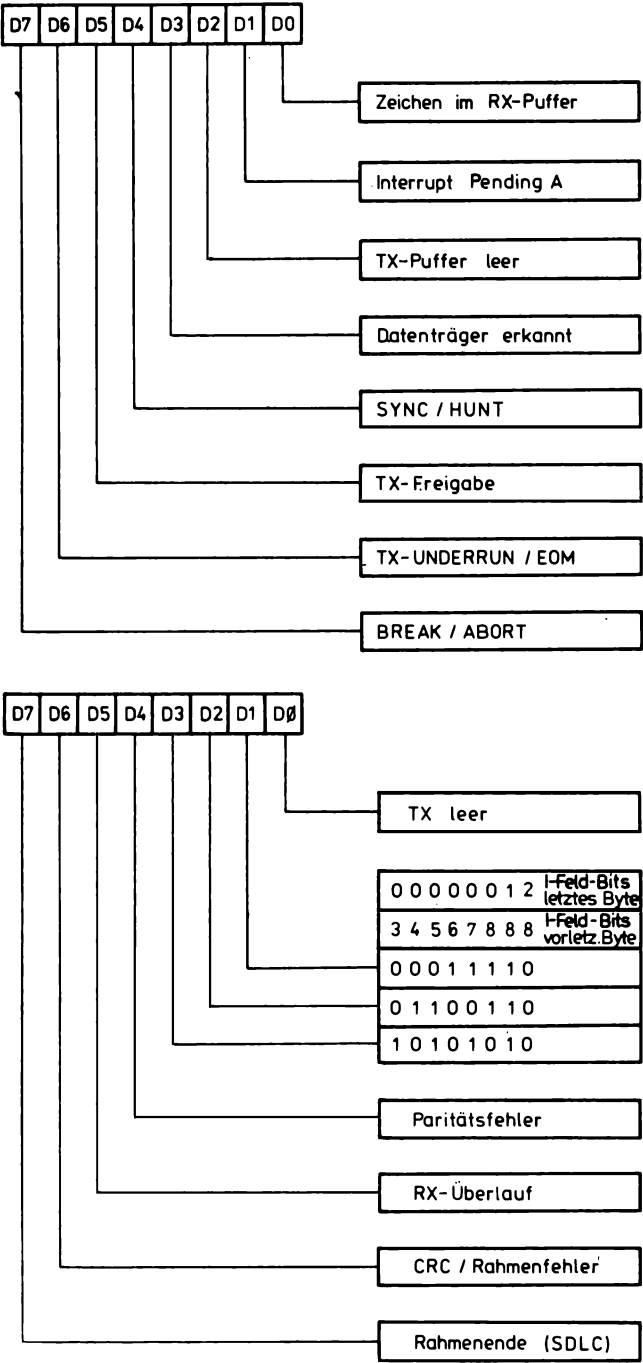


Bild 5.21
Leseregister



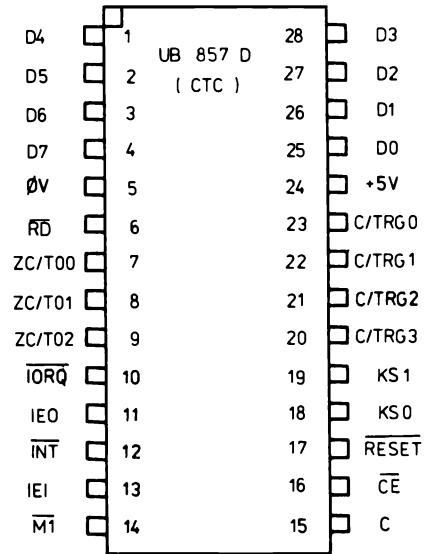
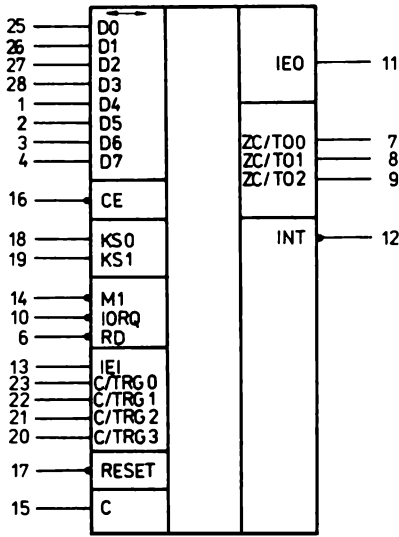


Bild 5.22 Anschlußbelegung und Schaltbild UB 857 D (CTC)

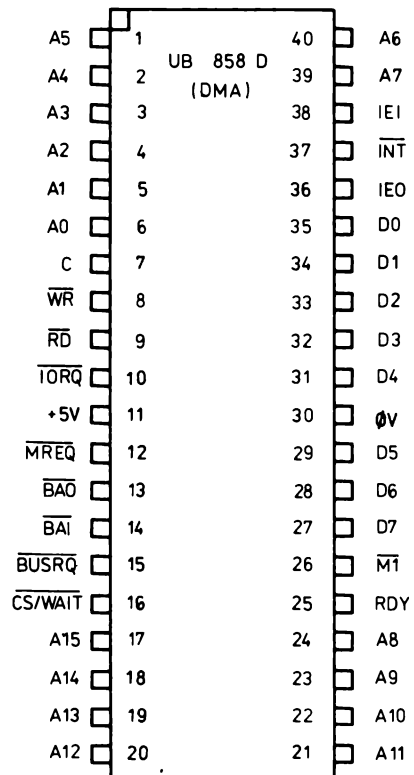
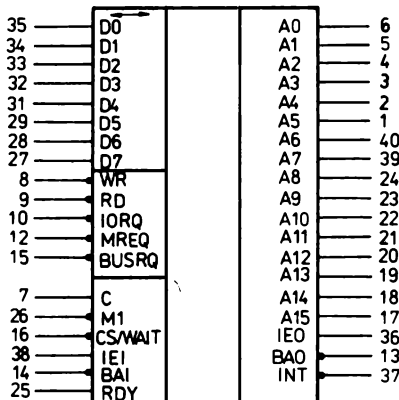


Bild 5.23 Anschlußbelegung und Schaltbild UB 858 D (DMA)

5. Anhang

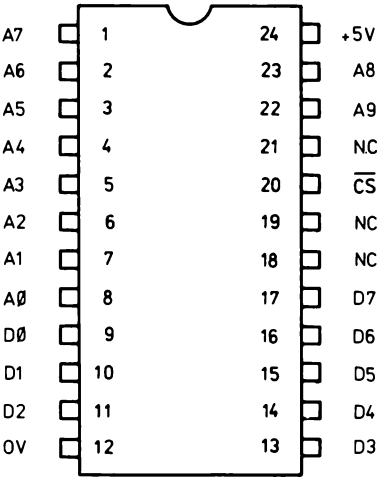


Bild 5.24 U 505 D

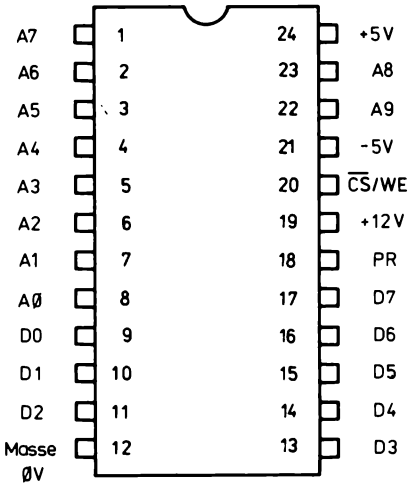


Bild 5.25 U 555 C

Tabelle 5.3. Technische Daten U 555 C

Grenzwerte

Kennwert	Symbol	Wert		Einheit
		min.	max.	
Betriebsspannungen	U_{DD}		20	V
	U_{CC}		15	V
	U_{SS}		15	V
	U_{PR}	-0,3	32	V
Betriebsspannung für Programmierbetrieb				
Eingangsspannungen	U_{I1}		15	V
$A_r O_1$				
Eingangsspannung CS/WE	U_{I2}		15	V
Umgebungstemperatur	ϑ_a	0	70	°C
Lagerungstemperatur	ϑ_{stg}	-55	125	°C
Gesamtverlustleistung bei $\vartheta_a = 45^\circ\text{C}$	P_V		1,5	W

Statische Kennwerte (auf $U_{SS} = 0 \text{ V}$ bezogen)

Kennwert	Symbol	Meßbedingung	Wert			Einheit
			min.	typ.	max.	
Betriebsspannungen	$-U_{BB}$		4,75	5,0	5,25	V
	U_{CC}		4,75	5,0	5,25	V
	U_{DD}		11,4	12,0	12,6	V
Eingangs-High-Spannungen für Chip-Select, Adressen und Daten	U_{IH1}		3,0	–	$U_{CC}-0,5$	V
Eingangs-High-Spannungen für Programmierung an $\overline{CS}/WE$	U_{IH2}		11,4	–	12,6	V
Eingangs-Low-Spannung	U_{IL}		–0,3	–	0,8	V
Umgebungstemperatur	θ_a		0	25	70	°C
Eingangsreststrom	I_1	$U_{IH} = 5,5 \text{ V}$ $U_{\overline{CS}/WE} = 5,5 \text{ V}$			7	µA
Ausgangsreststrom	I_0	$U_{\overline{CS}/WE} = 3 \text{ V}$ $U_{\overline{CS}/WE} = 5 \text{ V}$	–7		7	µA
Ausgangsspannung Low	U_{OL}	$I_{OL} = 1,6 \text{ mA}$			0,4	V
Ausgangsspannung High	U_{OH}	$U_{IL} = 0,8 \text{ V}$ $U_{IH} = 3 \text{ V}$	2,4			V
Stromaufnahme	I_{BB}		–45			mA
	I_{CC}				10	mA
	I_{DD}				65	mA
Programmierstromaufnahme	I_{PR1}				20	mA
	I_{PR2}		–3			mA
Eingangskapazität	C_1				6	pF
Ausgangskapazität	C_0				12	pF

Dynamische Kennwerte (auf $U_{SS} = 0 \text{ V}$ bezogen)

Kennwert	Symbol	Meßbedingung	Max. Wert	Einheit
Chip-Selectionszeit	t_{OC}		120	ns
Chip-Deselectionszeit	t_{OD}		120	ns
Zugriffszeit	t_{ACC}	$U_{\overline{CS}/WE} = 0,8 \text{ V}$	450	ns

5. Anhang

Programmierbedingungen

Kennwert	Symbol	Wert			Einheit
		min.	typ.	max.	
Betriebsspannungen bezogen auf U_{SS}	U_{BB}	-5,25	-5,0	-4,75	V
	U_{CC}	4,75	5,0	5,25	V
	U_{DD}	11,4	12,0	12,6	V
Programmierimpuls H-Pegel	U_{PRH}	25	26	27	V
Programmierimpuls L-Pegel	U_{PRL}	0	-	1	V
Betriebstemperatur (Programmierbetrieb)	θ_A	20	25	30	°C
Programmierimpulsbreite	t_p	0,1	-	1	ms
Programmierimpulsanstiegszeit	t_{PRR}	0,5	-	2	µs
Programmierimpulsabfallzeit	t_{PRF}	0,5	-	-	µs
Programmierzeit	t_R	50	-	-	ms
Adressenbereitstellzeit	t_{RS}	10	-	-	µs
CS/WE-Bereitstellzeit	t_C	10	-	-	µs
Daten-Bereitstellzeit	t_{DS}	10	-	-	µs
Adressen-Haltezeit	t_{AH}	1	-	-	µs
CS/WE-Haltezeit	t_{CH}	0,5	-	-	µs
Daten-Haltezeit	t_{DH}	1	-	-	µs
Datenverzögerung nach Programmier-/Leseumschaltung	t_{DPR}	-	-	10	µs

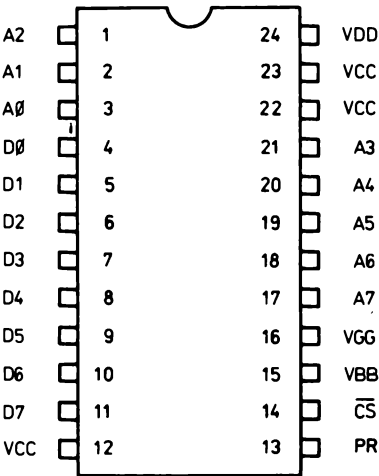


Bild 5.26 U 552 C; Anschlußbelegung

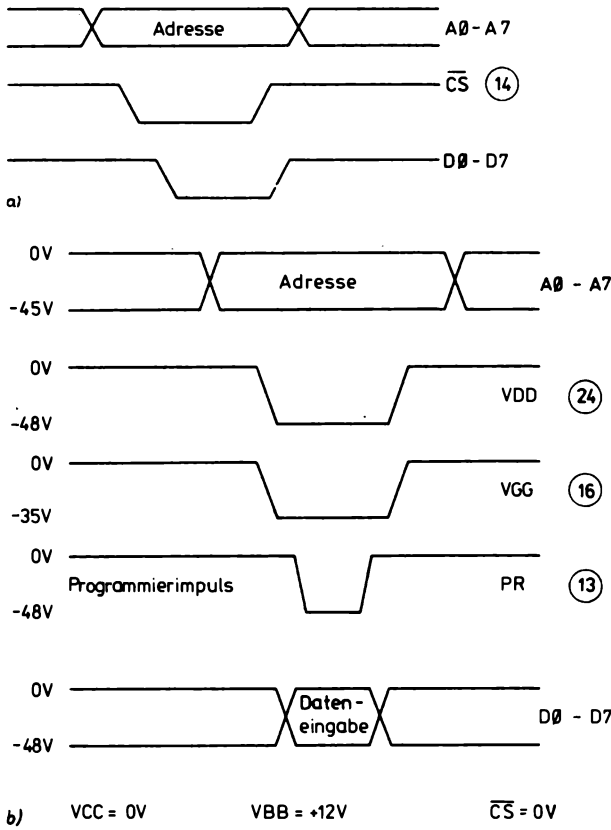
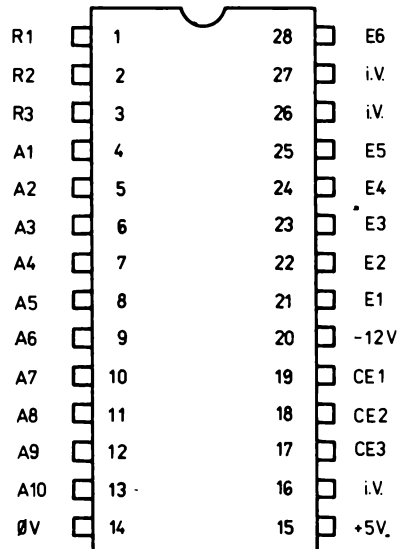


Bild 5.27
U 552 C; a – Daten lesen,
b – Programmieren

Bild 5.28 U'401 D



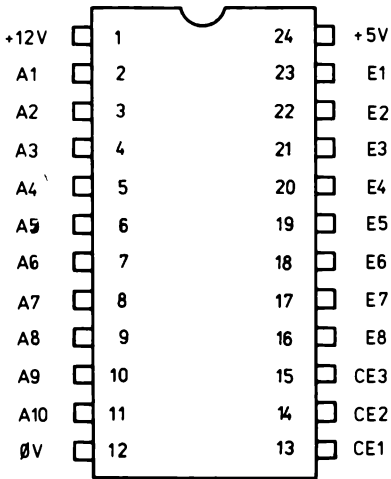


Bild 5.29 U 402 D

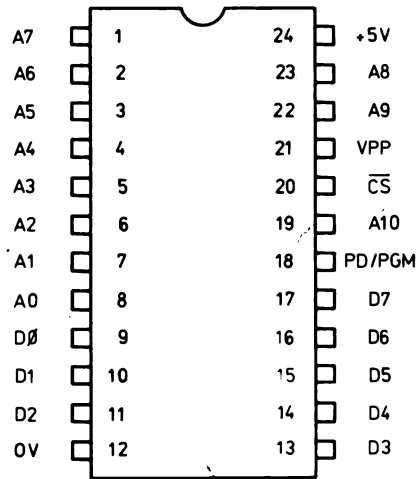


Bild 5.30 U 2716 C Anschlußbelegung

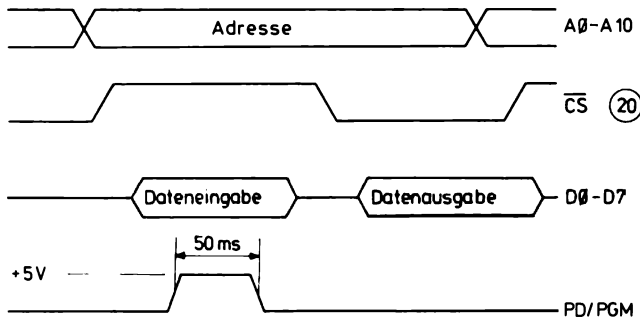
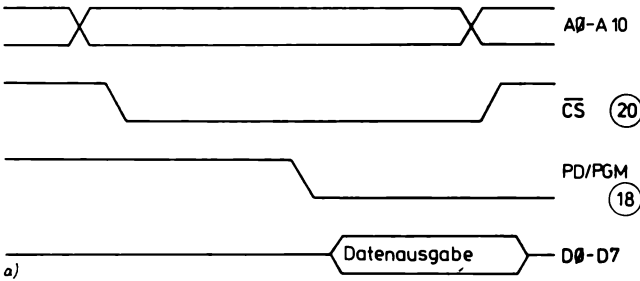


Bild 5.31
U 2716 C; a – Daten lesen,
b – Programmieren

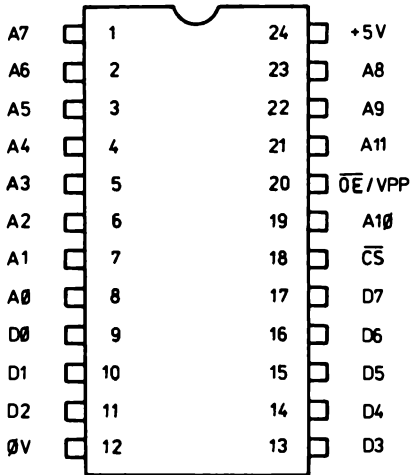


Bild 5.32 U 2732 C Anschlußbelegung

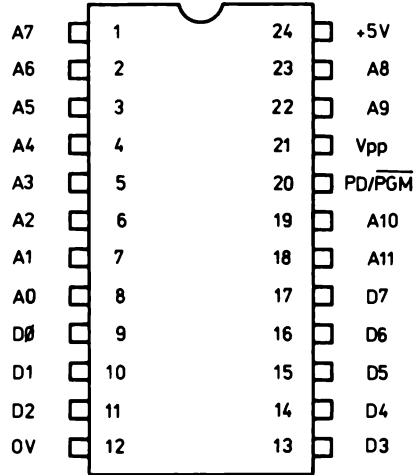


Bild 5.34 2532 Anschlußbelegung

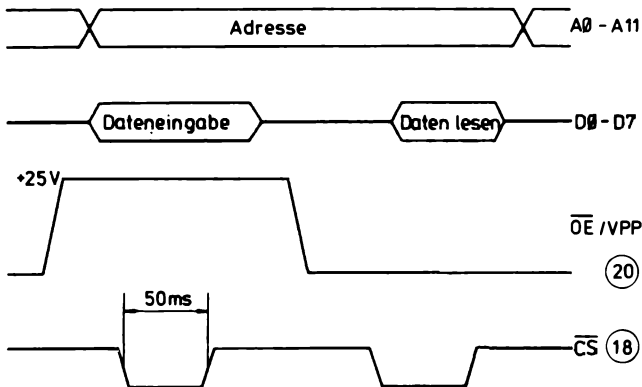


Bild 5.33
U 2732 C Daten lesen und Programmieren

5. Anhang

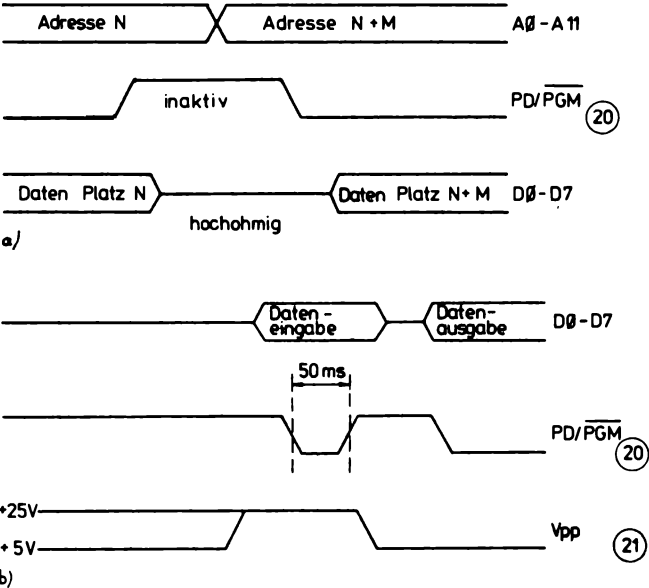
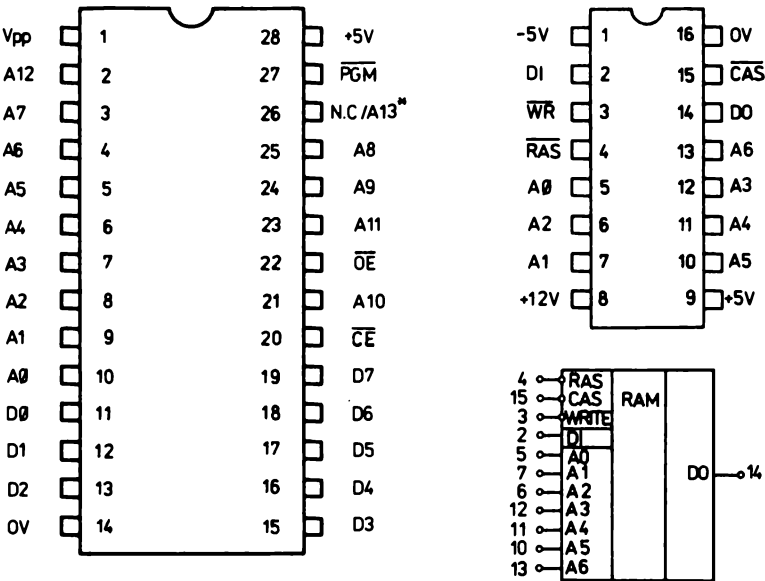


Bild 5.35
2532; a – Daten lesen,
b – Programmier n



* bei 27128

Bild 5.36 2764/27128 Anschlußbelegung

Bild 5.37 U 256 D Anschlußbelegung
und Schaltbild

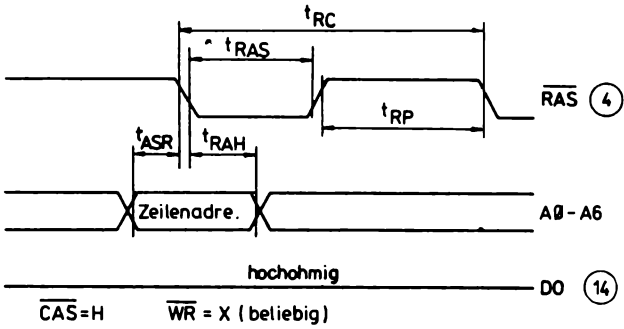


Bild 5.38
RAS-Refresh-Zyklus

Tabelle 5.4. Technische Daten U 256 D

Grenzwerte

Kenngröße	Kurzzeichen	Einheit	Min.	Max.
Betriebsspannungen, bezogen auf U_{BB}	U_{DD}	V	-1,0	20
	U_{CC}	V	-1,0	20
Betriebsspannungen, bezogen auf U_{BB} ($U_{DD} - U_{SS} > 0$)	U_{SS}	V	0	20
Betriebsspannungen, bezogen auf U_{SS}	U_{DD}	V	-1,0	15
	U_{CC}	V	1,0	15
Spannungen an den restlichen Anschlüssen, bezogen auf U_{BB}	U_1	V	-0,5	20
Umgebungstemperatur	θ_a	°C	0	70
Lagerungs- U 256 C	θ_i	°C	-55	155
temperatur U 256 D			-55	125
Verlustleistung	P_v	W	-	1,0
Ausgangskurzschlußstrom	I_{OS}	mA	-	50

Betriebsbedingungen

Alle Spannungen sind auf $U_{SS} = 0$ V bezogen

Kenngröße	Kurzzeichen	Einheit	Wert		
			min.	typ.	max.
Betriebsspannungen	$-U_{BB}$	V	4,5	5,0	5,5
	U_{CC}	V	4,5	5,0	5,5
	U_{DD}	V	10,8	12,0	13,2
H-Eingangsspannungen für $\overline{RAS}$, $\overline{CAS}$, $\overline{Write}$	U_{IHC}	V	2,7		5,5
H-Eingangsspannung für alle anderen Eingänge	U_{IH}	V	2,4		5,5
L-Eingangsspannung für alle Eingänge	U_{IL}	V	-1,0		0,8

5. Anhang

Kenngröße	Kurzzeichen	Einheit	Wert		
			min.	typ.	max.
Lese- oder Schreib-Zykluszeit ¹⁾	t_{RC}	ns	375		
Lese-Schreib-Zykluszeit ¹⁾	t_{RWC}	ns	375		
	t_{RMW}	ns	405		
Page-Mode-Zykluszeit	t_{PC}	ns	225		
Übergangszeit (Anstieg, Abfall)	t_1	ns	3	50	
$\overline{RAS}$ -Vorladezeit	t_{RP}	ns	120	–	
$\overline{RAS}$ -Impulsbreite	t_{RAS}	ns	200	10 000	
$\overline{RAS}$ -Haltezeit	t_{RSH}	ns	135	–	
$\overline{CAS}$ -Haltezeit	t_{CSH}	ns	200		
$\overline{RAS}$ - $\overline{CAS}$ -Verzögerungszeit ²⁾	t_{RCD}	ns	25	65	
$\overline{CAS}$ -Impulsbreite	t_{CAS}	ns	135		
$\overline{CAS}$ - $\overline{RAS}$ -Vorladezeit	t_{CRP}	ns	20		
Zeilenadressenvorhaltezeit	t_{RAH}	ns	25		
Zeilenadressenvorhaltezeit	t_{ASR}	ns	0		
Spaltenadressenvorhaltezeit	t_{ASC}	ns	–10	–	–
Spaltenadressenhaltezeit	t_{CA}	ns	55		
Spaltenadressenhaltezeit von $\overline{RAS}$ an	t_{AR}	ns	120		
Lesekommandovorhaltezeit	t_{RCS}	ns	0		
Lesekommandohaltezeit	t_{RCH}	ns	0		
Schreibkommandohaltezeit	t_{WCH}	ns	55		
Schreibkommandohaltezeit von $\overline{RAS}$ an	t_{WCR}	ns	120		
Schreibkommandopulsbreite	t_{WP}	ns	55		
Schreibkommando- $\overline{RAS}$ -Vorhaltezeit	t_{RWL}	ns	70		
Schreibkommando- $\overline{CAS}$ -Vorhaltezeit	t_{CWL}	ns	70		
Dateneingangsvorhaltezeit ³⁾	t_{DS}	ns	0		
Dateneingangshaltezeit ³⁾	t_{DH}	ns	55		
Dateneingangshaltezeit von $\overline{RAS}$ an	t_{DHR}	ns	120		
$\overline{CAS}$ -Vorladezeit (nur für Page-Mode)	t_{CP}	ns	80		
Refresh-Periode	t_{REF}	ms	–		2
Write-Kommando-Vorhaltezeit	t_{WCS}		–20		
$\overline{CAS}$ -Write-Verzögerungszeit	t_{CWD}	ns	80		–
$\overline{RAS}$ -Write-Verzögerungszeit	t_{RWD}		145		
Umgebungstemperatur	θ_a	°C	0	25	70

- 1) Die Angaben für $t_{RC(\min)}$, $t_{RWC(\min)}$ und $t_{RMW(\min)}$ werden benutzt, um die Zykluszeit für den vollen Temperaturbereich ($0^\circ\text{C} \leq \theta_a \leq 70^\circ\text{C}$) anzugeben.
- 2) Betrieb innerhalb $t_{RCD(\max)}$ sichert, daß $t_{RAC(\max)}$ eingehalten wird. $t_{RCD(\max)}$ ist Bezugspunkt. Wenn $t_{RCD} > t_{RCD(\max)}$, dann wird die Zugriffszeit ausschließlich von t_{CAC} bestimmt.
- 3) Diese Parameter sind auf die $\overline{CAS}$ -Flanke in Early-Write-Zyklen und auf die $\overline{WRITE}$ -Flanke in Delayed-Write- oder Read-Modify-Zyklen bezogen.

Elektrische Hauptkenngrößen

Kenngröße	Kurzzeichen	Einheit	Min.	Max.
Stromaufnahme, mittlerer Wert bei $\overline{\text{RAS}}$ - $\overline{\text{CAS}}$ -Zyklen	I_{DD1}	mA		35
Ruhestrom	I_{BB1}	μA		200
	I_{DD2}	mA		1,5
	I_{BB2}	μA		100
Eingangsstrom	I_1	μA	-10	10
			-20	20
Ausgangsreststrom	I_0	μA	-10	10
			-20	20
H-Ausgangsspannung ⁴⁾	U_{OH}	V	2,4	
			2,0	
L-Ausgangsspannung ⁴⁾	U_{OL}	V		0,4
				0,8
Zugriffszeit von $\overline{\text{RAS}}$ aus	t_{RAC}	ns		200
Zugriffszeit von $\overline{\text{CAS}}$ aus	t_{CAC}	ns		135

- 4) U_0 schaltet zwischen U_{SS} und U_{CC} , wenn der Ausgang aktiviert wird und keine Strombelastung auftritt. Es ist erlaubt, U_{CC} auf U_{SS} zu reduzieren, ohne die Datenspeicherung und Refresh zu beeinträchtigen; allerdings wird dabei der Wert $U_{\text{OH (min)}}$ nicht garantiert.

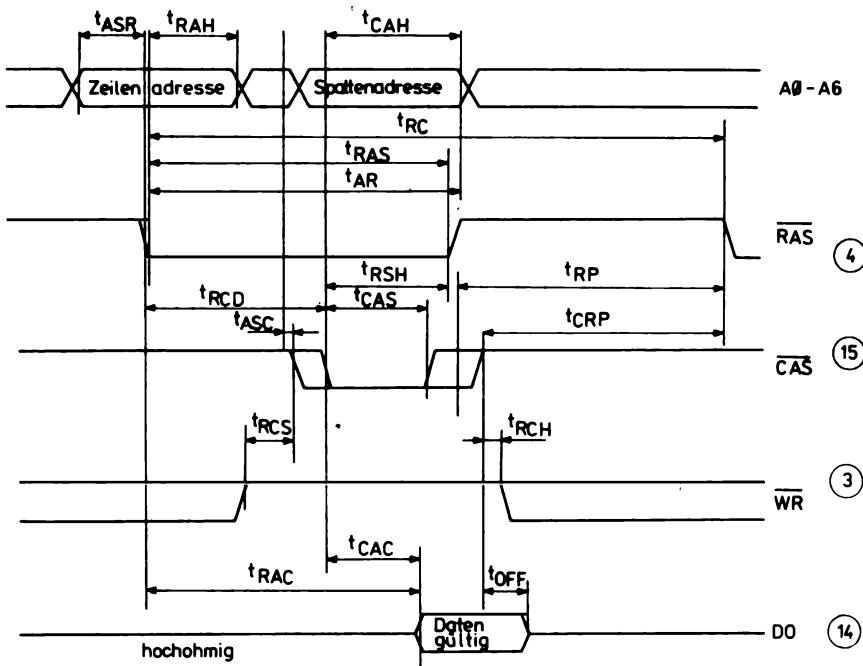


Bild 5.39 Lese-Zyklus

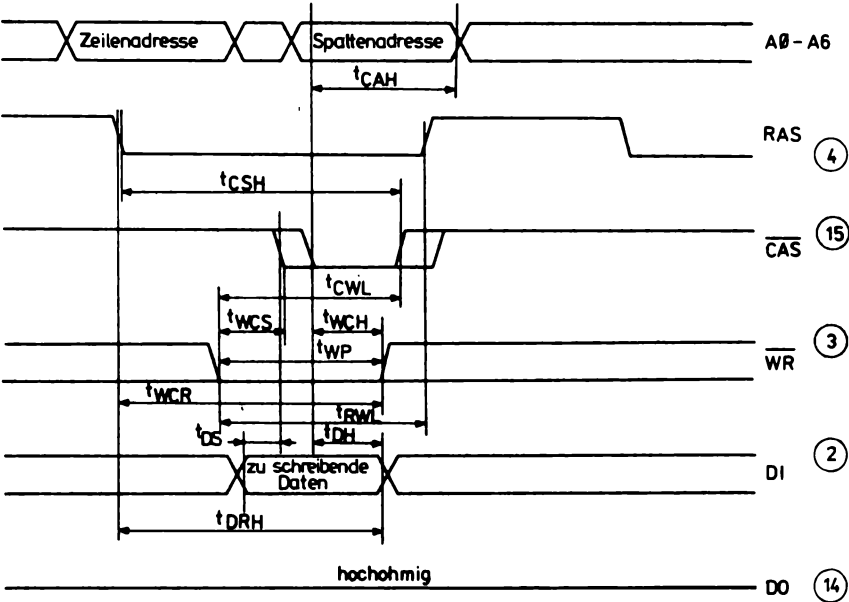


Bild 5.40 Schreib-Zyklus

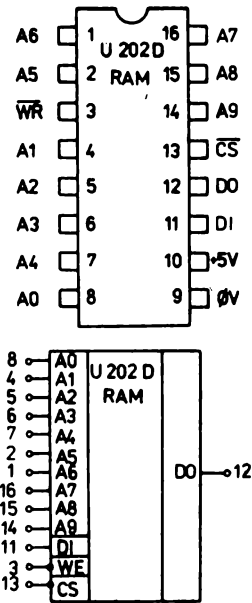


Bild 5.41
U 202 D

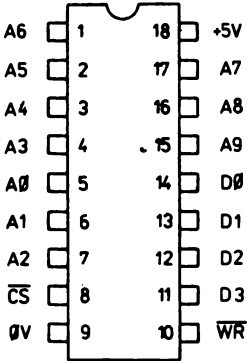


Bild 5.42
U 214 D/U 224 D

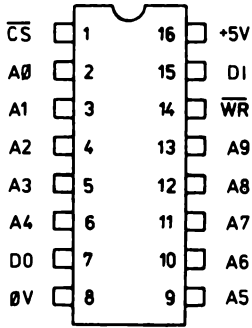


Bild 5.43
U 215 D/U 225 D

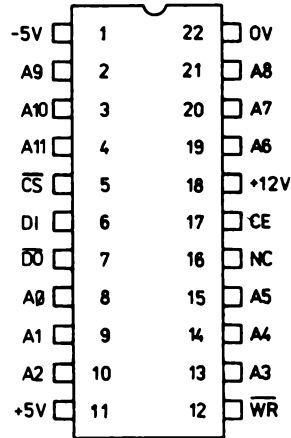


Bild 5.46
2107

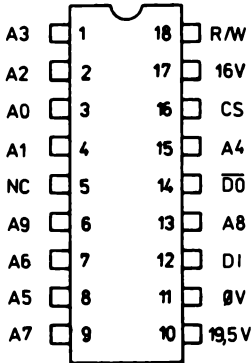


Bild 5.44
U 253 D

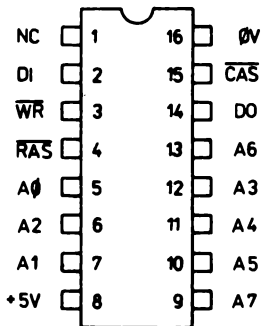


Bild 5.45 U 2164 (4164)

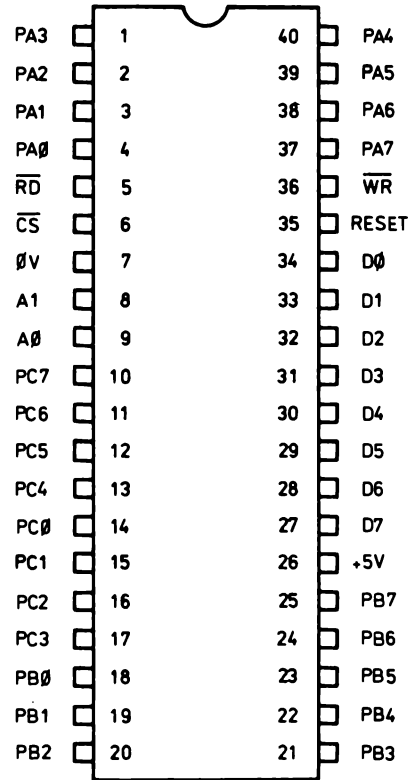


Bild 5.47 8255 Anschlußbelegung

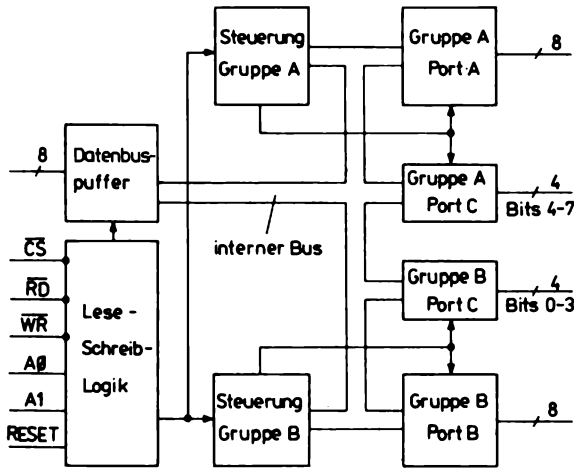


Bild 5.48
8255 Übersichtsschaltplan

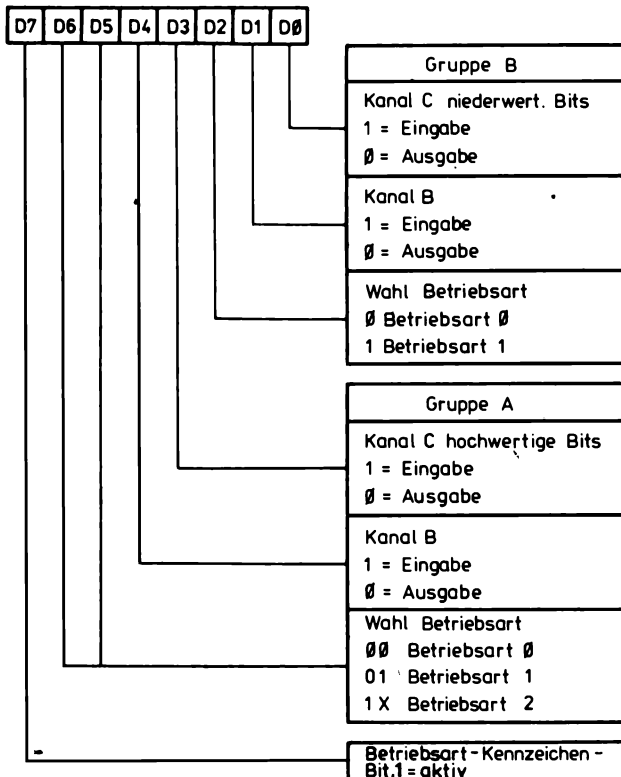


Bild 5.49
Betriebsart-Auswahl

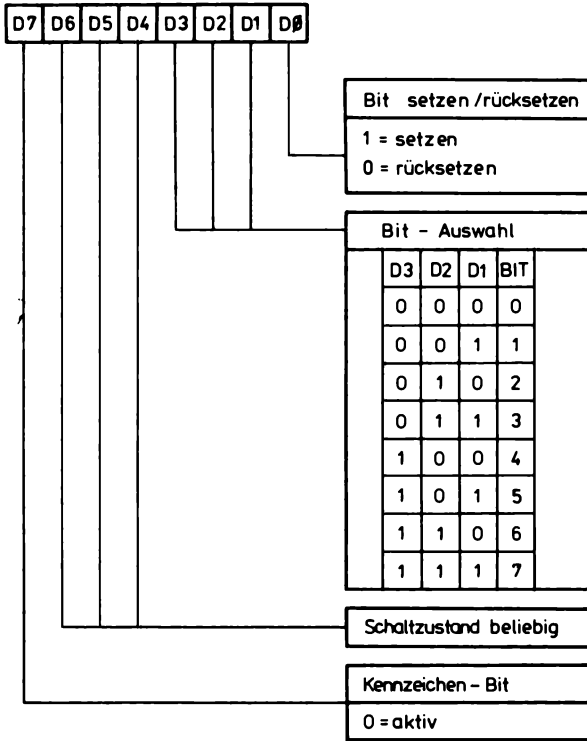


Bild 5.50
Steuerwort Bitsetzen

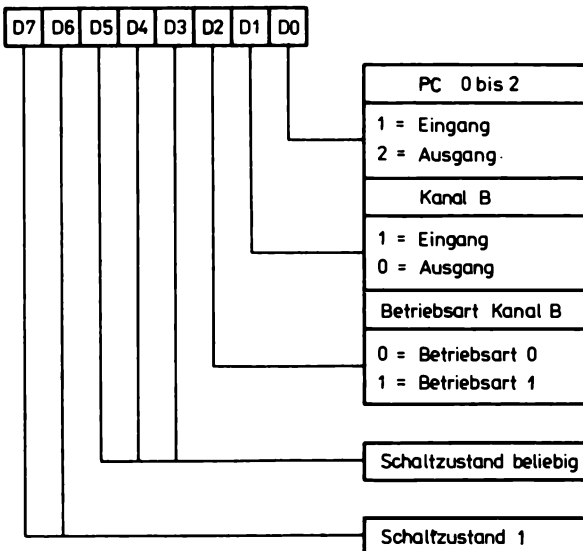


Bild 5.51
Steuerwort Bidirektionale Betriebsa

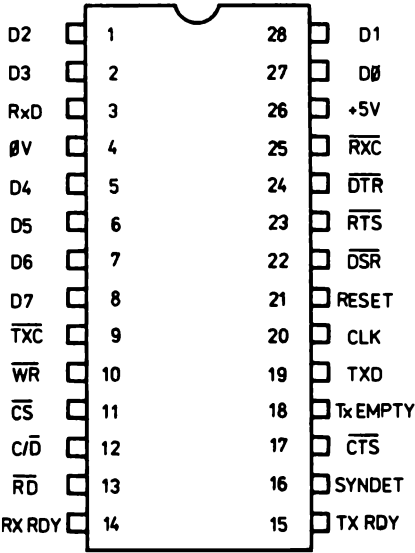


Bild 5.52 8251 Anschlußbelegung

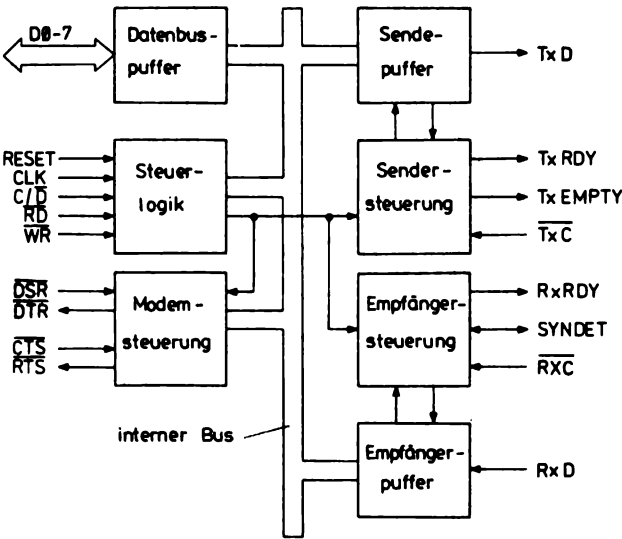


Bild 5.53
8251 Übersichtsschaltplan

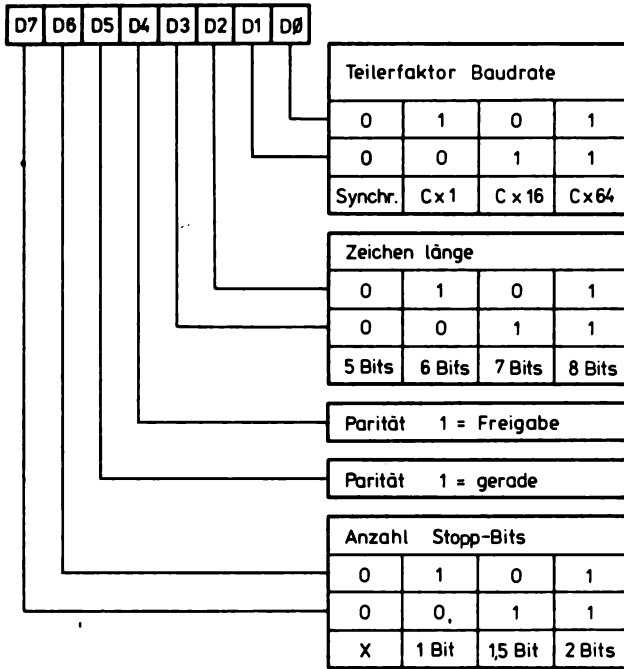


Bild 5.54
Steuerwort Asynchronbetrieb

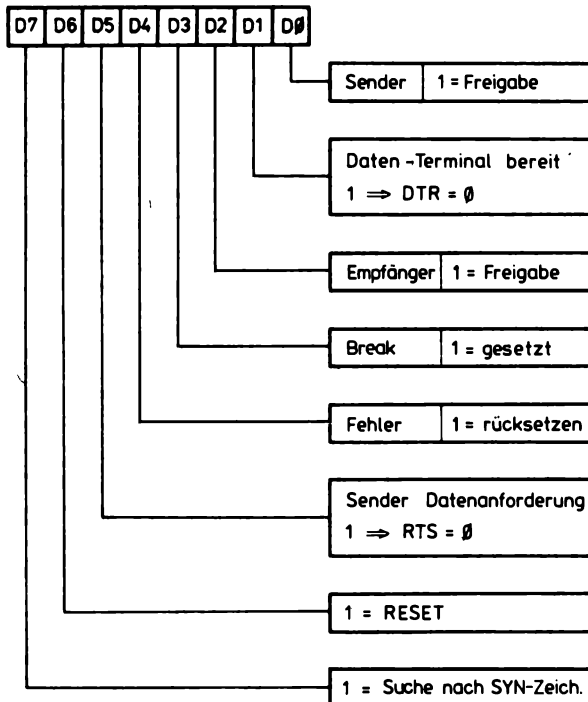


Bild 5.55
Kommandowort

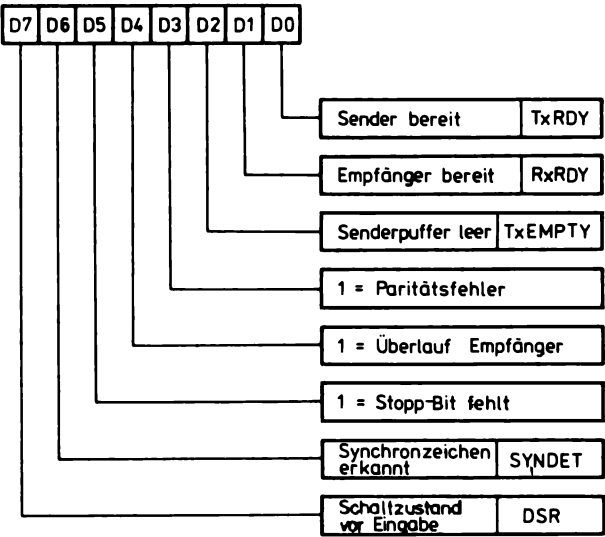


Bild 5.56
Lese-Register

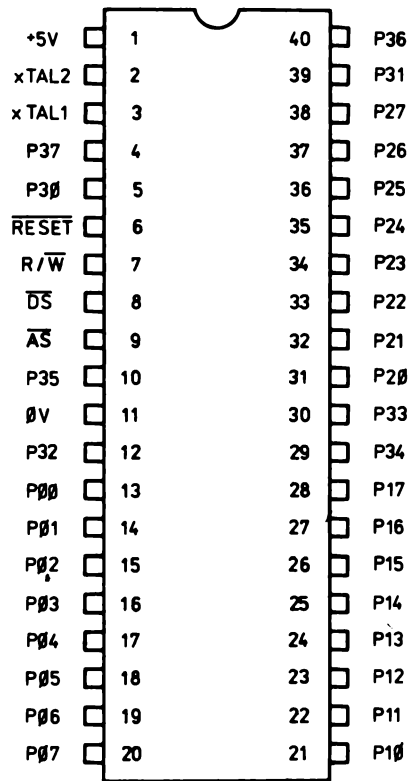


Bild 5.58 UB 8810 D

				Adresse	Funktion	Kurz- zeichen
P36	1	64	+5V	FF	Kellerzeiger Bit 7-0	SPL
P31	2	63	XTAL2	FE	Kellerzeiger Bit 15-8	SPH
P27	3	62	XTAL1	FD	Registerzeiger	RP
P26	4	61	P37	FC	Flags	Flags
P25	5	60	P30	FB	Interrupt - Maske	IMR
P24	6	59	RESET	FA	Interrupt - Anforderung	IRQ
P23	7	58	R/W	F9	Interrupt - Priorität	IPR
P22	8	57	DS	F8	Betriebsart Kanäle 0 und 1	P01M
P21	9	56	AS	F7	Betriebsart Kanal 3	P3M
P20	10	55	P35	F6	Betriebsart Kanal 2	P2M
P33	11	54	P32	F5	Vorteiler T0	PRE0
P34	12	53	P00	F4	Voreinstellung Zähler 0	T0
P17	13	52	P01	F3	Vorteiler T1	PRE1
P16	14	51	P02	F2	Voreinstellung Zähler 1	T1
P15	15	50	P03	F1	Betriebsart Zeitgeber	TMR
P14	16	49	P04	F0	Serielle Ein-/Ausgabe- Daten	SIO
P13	17	48	0V			
P12	18	47	P05		nicht belegt	
P11	19	46	P06			
P10	20	45	P07	7F		
D7	21	44	IACK		Register - Block für Anwender.	
D6	22	43	SYNC			
D5	23	42	SCLK			
D4	24	41	MDS	4		
A0	25	40	D0	3	In - Out - Kanal 3	
A1	26	39	D1	2	Kanal 2	
A2	27	38	D2	1	Kanal 1	
A3	28	37	D3	0	Kanal 0	
A4	29	36	A11			
A5	30	35	A10			
A6	31	34	A9			
A7	32	33	A8			

Bild 5.57 UB 8820 D

Bild 5.59 Registerblock der Einchip-Mikrocomputer

Tabelle 5.5. Befehlssatz UB 8820 M

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	DEC R1	DEC R1	ADD r1,r2	ADD r1,r2	ADD R2,R1	ADD R2,R1	ADD R1,IM	ADD R1,IM	LD r1,R2	LD r2,R1	DJNZ r1,RA	JR R1	LD r1,IM	JP R1	INC r1	
1	RLC R1	RLC R1	ADC r1,r2	ADC r1,r2	ADC R2,R1	ADC R2,R1	ADC R1,IM	ADC R1,IM								
2	INC R1	INC R1	SUB r1,r2	SUB r1,r2	SUB R2,R1	SUB R2,R1	SUB R1,IM	SUB R1,IM								
3	JP R1	SRP R1	SBC r1,r2	SBC r1,r2	SBC R2,R1	SBC R2,R1	SBC R1,IM	SBC R1,IM								
4	DA R1	DA R1	OR r1,r2	OR r1,r2	OR R2,R1	OR R2,R1	OR R1,IM	OR R1,IM								
5	POP R1	POP R1	AND r1,r2	AND r1,r2	AND R2,R1	AND R2,R1	AND R1,IM	AND R1,IM								
6	COM R1	COM R1	TCM r1,r2	TCM r1,r2	TCM R2,R1	TCM R2,R1	TCM R1,IM	TCM R1,IM								
7	PUSH R1	PUSH R1	TM r1,r2	TM r1,r2	TM R2,R1	TM R2,R1	TM R1,IM	TM R1,IM								
8	DECW R1	DECW R1	LDE r1,r2	LDE r1,r2	LDE R2,R1	LDE R2,R1	LDE R1,IM	LDE R1,IM								DI
9	RL R1	RL R1	LDE r1,r2	LDE r1,r2	LDE R2,R1	LDE R2,R1	LDE R1,IM	LDE R1,IM								EI
A	INCW R1	INCW R1	CP r1,r2	CP r1,r2	CP R2,R1	CP R2,R1	CP R1,IM	CP R1,IM								RET
B	CLR R1	CLR R1	XOR r1,r2	XOR r1,r2	XOR R2,R1	XOR R2,R1	XOR R1,IM	XOR R1,IM								IRET
C	RRC R1	RRC R1	LDC r1,r2	LDC r1,r2	LDC R2,R1	LDC R2,R1	LDC R1,IM	LDC R1,IM								RCF
D	SRA R1	SRA R1	LDC r2,lr1	LDC r2,lr1	LDC R1,IM	LDC R1,IM	LDC R1,IM	LDC R1,IM								SCF
E	RR R1	RR R1	LD r1,r2	LD r1,r2	LD R2,R1	LD R2,R1	LD R1,IM	LD R1,IM								CCF
F	SWAB R1	SWAB R1	LD r1,r2	LD r1,r2	LD R2,R1	LD R2,R1	LD R1,IM	LD R1,IM								NOP

Erläuterungen: R – Register, 8-Bit – Adresse
r – Register, 4-Bit – Adresse
Nutzung Registerzeiger

R1, r1 – Zieladresse
R2, r2 – Quelladresse

F1	1	48	$\overline{CS}$
$\overline{F2}$	2	47	IV
F3	3	46	Z
$\overline{RQ}$	4	45	V
$\overline{WR}$	5	44	C
S2	6	43	S4
S1	7	42	S3
K1-15	8	41	K2-15
K1-14	9	40	K2-14
K1-13	10	39	K2-13
K1-12	11	38	K2-12
K1-11	12	37	+5V
$\emptyset V$	13	36	K2-11
K1-10	14	35	K2-10
K1-9	15	34	K2-9
K1-8	16	33	K2-8
K1-7	17	32	K2-7
K1-6	18	31	K2-6
K1-5	19	30	K2-5
K1-4	20	29	K2-4
K1-3	21	28	K2-3
K1-2	22	27	K2-2
K1-1	23	26	K2-1
K1-0	24	25	K2-0

Bild 5.60 U 8032 Anschlußbelegung
und Schaltbild

AD0	1	48	AD8
AD9	2	47	SN6
AD10	3	46	SN5
AD11	4	45	AD7
AD12	5	44	AD6
AD13	6	43	AD4
STOP	7	42	SN4
μI	8	41	AD5
AD15	9	40	AD3
AD14	10	39	AD2
+5V	11	38	AD1
$\overline{VI}$	12	37	SN2
NVI	13	36	$\emptyset V$
SE6T	14	35	TAKT
NMI	15	34	$\overline{AS}$
RESET	16	33	Substrat
$\overline{IO}$	17	32	B/W
MREQ	18	31	N/S
$\overline{DS}$	19	30	R/W
ST3	20	29	BUSAK
ST2	21	28	$\overline{WAIT}$
ST1	22	27	$\overline{BUSRQ}$
ST0	23	26	SN0
SN3	24	25	SN1

Bild 5.61 U 8001

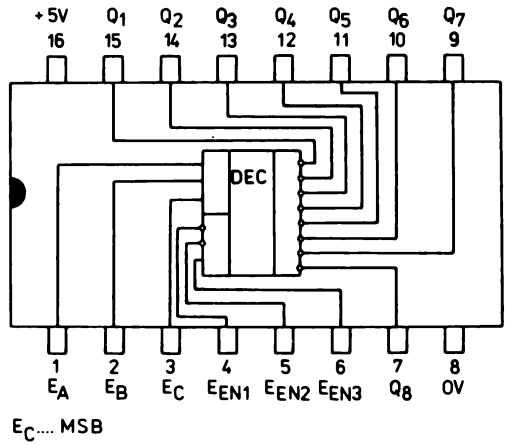
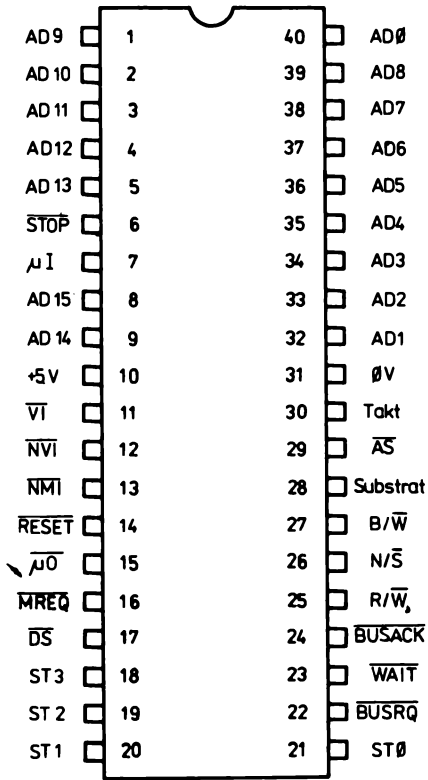


Bild 5.63 DS 8205

Bild 5.62 U 8002

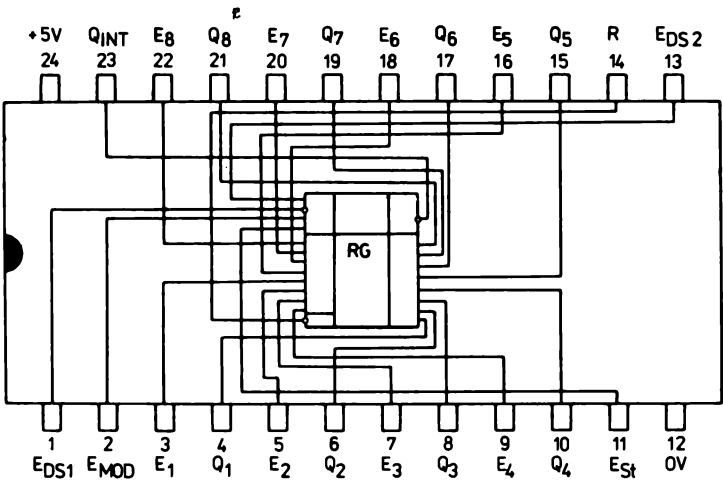


Bild 5.64
DS 8212 Anschluß-
belegung

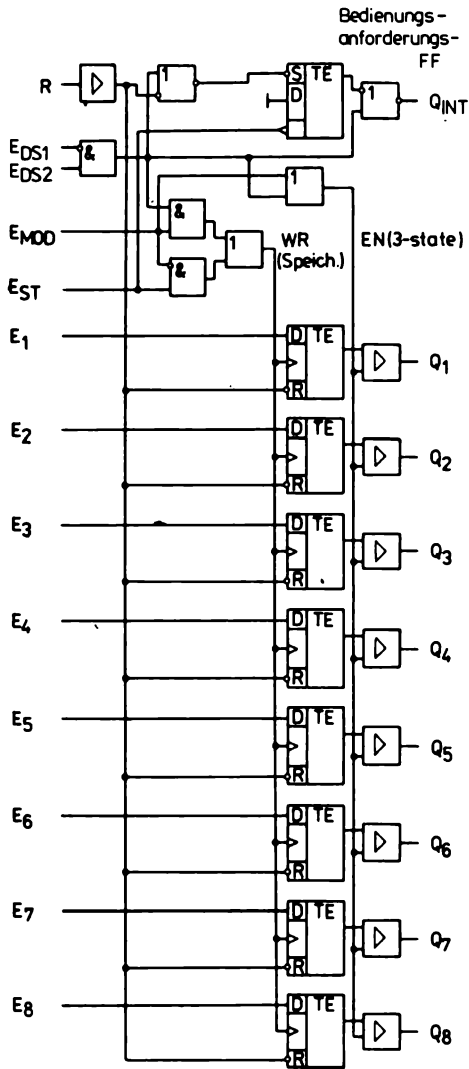


Bild 5.65 DS 8212 Übersichtsschaltplan

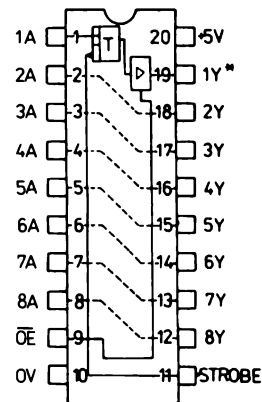

DS 8283 invertierende
Ausgänge

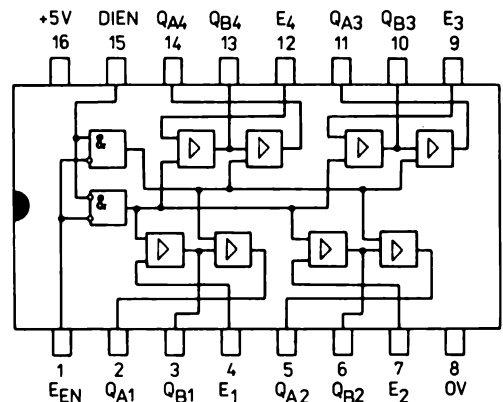
Bild 5.67
DS 8282/DS 8283


Bild 5.66 DS 8216

5. Anhang

Tabelle 5.6. Technische Daten Interface-Schaltkreise

Statische Kennwerte ($U_s = 5 \text{ V}$, $\theta_s = 0$ bis $+70^\circ\text{C}$)

		min.	max.
H-Ausgangsspannung	U_{OH}		
$U_s = 4,75 \text{ V}$	in V	2,4	
<i>DS 8205 D, DS 8282 D, DS 8283 D</i>		2,4	
<i>DS 8286 D, DS 8287 D</i>			
$U_s = 4,75 \text{ V}$, $-I_{OH} = 1 \text{ mA}$			
<i>DS 8216 D (A-Ausgänge), DS 8212 D</i>		3,65	
$U_s = 4,75 \text{ V}$, $-I_{OH} = 10 \text{ mA}$			
<i>DS 8216 D (B-Ausgänge)</i>		2,4	
L-Ausgangsspannung	U_{OL}		
$U_s = 4,75 \text{ V}$	in V		
<i>DS 8205 D</i>			0,45
<i>DS 8282 D, DS 8283 D, DS 8286 D, DS 8287 D</i>			0,5
$U_s = 4,75 \text{ V}$, $I_{OL} = 15 \text{ mA}$			
<i>DS 8216 D (A-Ausgänge), DS 8212 D</i>			0,45
$U_s = 4,75 \text{ V}$, $I_{OL} = 5,5 \text{ mA}$ <i>DS 8216 D (B-Ausgänge)</i>			0,6
L-Eingangsstrom	$-I_{IL}$		
$U_s = 5,25 \text{ V}$, $U_{IL} = 0,45 \text{ V}$	in μA		
<i>DS 8205 D, DS 8212 D (außer MD, $\overline{S_1}$)</i>			250
<i>DS 8283 D, DS 8282 D, DS 8286 D, DS 8287 D</i>			200
<i>DS 8216 D (Dateneingänge)</i>			250
<i>DS 8216 D ($\overline{OE}$, $\overline{DIR}$)</i>			500
<i>DS 8212 D (MD)</i>			750
<i>DS 8212 D (S_1)</i>			1000
H-Eingangsstrom	I_{IH}		
$U_s = 5,25 \text{ V}$, $U_{IH} = 5,25 \text{ V}$	in μA		
<i>DS 8205 D, DS 8212 D (außer MD, $\overline{S_1}$)</i>			10
<i>DS 8216 D (Dateneingänge)</i>			
<i>DS 8212 D (MD)</i>			30
<i>DS 8212 D ($\overline{S_1}$)</i>			40
<i>DS 8216 D ($\overline{OE}$ und $\overline{DIR}$)</i>			20
<i>DS 8282 D, DS 8283 D</i>			50
<i>DS 8286 D, DS 8287 D</i>			
Ausgangskurzschlußstrom¹⁾	$-I_{OS}$		
$U_s = 5 \text{ V}$	in mA		
<i>DS 8205 D</i>		40	120
<i>DS 8212 D</i>		15	75
<i>DS 8216 D (A-Ausgänge)</i>		15	65
<i>DS 8216 D (B-Ausgänge)</i>		30	120
Ausgangsstrom bei tri-state	$-I_{OZ}$		
$U_s = 5,25 \text{ V}$	in μA		
<i>DS 8282 D, DS 8283 D, DS 8286 D</i>			50
<i>DS 8287 D</i>			

Statische Kennwerte ($U_s = 5 \text{ V}$, $\theta_a = 0$ bis $+70^\circ \text{C}$)

		min.	max.
Flußspannung der Eingangsdiode	$-U_1$		
$-I_1 = 18 \text{ mA}$, $U_s = 4,75 \text{ V}$	in V		
DS 8205 D			1,5
$-I_1 = 5 \text{ mA}$, $U_s = 4,75 \text{ V}$			
DS 8282 D, DS 8283 D			1
DS 8286 D, DS 8287 D			
Stromaufnahme	I_s		
$U_s = 5,25 \text{ V}$	in mA		
DS 8205 D			70
DS 8212 D, DS 8216 D			130
DS 8282 D, DS 8283 D, DS 8286 D, DS 8287 D			160
H-Eingangsspannung	U_{IH}		
DS 8212 D, DS 8216 D	in V	2	
Einstellwerte wie Betriebsbedingung			
L-Eingangsspannung	U_{IL}		
DS 8212 D, DS 8216 D	in V		0,8
Einstellwerte wie Betriebsbedingung			
Eingangsclampingspannung	U_{IK}		
$U_s = 4,75 \text{ V} - I_1 = 5 \text{ mA}$	in V		
DS 8212 D, DS 8216 D			1
Ausgangsstrom im hochohmigen Zustand	I_{OZ}		
$U_s = 5,25 \text{ V}$, $U_O = 0,45 \text{ V}$	in μA		
bzw. $U_O = 5,25 \text{ V}$			
DS 8212 D, DS 8216 D (Ausgang A)			20
DS 8216 D (Ausgang B)			100

1) Kurzschluß nur an einem Ausgang und nicht länger als 1 s.

Dynamische Kennwerte ($U_s = 5 \text{ V}$, $\theta_a = 25^\circ \text{C} \pm 5 \text{ K}$).

		min.	max.
Signalverzögerungszeit	t_{PLH}		
$U_s = 5 \text{ V}$	in ns		
A $\rightarrow$ Y DS 8205 D			18
E $\rightarrow$ Y DS 8205 D			18
A $\rightarrow$ Y DS 8212 D			30
$\overline{S}_1$ oder $S_2 \rightarrow \text{INT}$ DS 8212 D			30
STB, $\overline{S}_1$ oder $S_2 \rightarrow$ Y DS 8212 D			40
B $\rightarrow$ A DS 8216 D			25
DI $\rightarrow$ B DS 8216 D			30
ST $\rightarrow$ B DS 8282 D			55
ST $\rightarrow$ B DS 8283 D			45
A $\rightarrow$ B DS 8286 D			35
DS 8287 D			35

5. Anhang

Dynamische Kennwerte ($U_s = 5 \text{ V}$, $\theta_a = 25^\circ\text{C} \pm 5 \text{ K}$).

		min.	max.
$U_s = 5 \text{ V}$			
$A \rightarrow Y \text{ DS } 8205 \text{ D}$	t_{PHL} in ns		18
$E \rightarrow Y \text{ DS } 8205 \text{ D}$			18
$A \rightarrow Y \text{ DS } 8212 \text{ D}$			30
$\text{CLR} \rightarrow Y \text{ DS } 8212 \text{ D}$			55
$\text{STB} \rightarrow \text{INT DS } 8212 \text{ D}$			40
$\text{STB}, \overline{S_1} \text{ oder } S_2 \rightarrow Y \text{ DS } 8212 \text{ D}$			40
$\text{DI} \rightarrow B \text{ DS } 8216 \text{ D}$			30
$B \rightarrow A \text{ DS } 8216 \text{ D}$			25
$A \rightarrow B \text{ DS } 8282 \text{ D}$			35
$A \rightarrow B \text{ DS } 8283 \text{ D}$			25
$A \rightarrow B \text{ DS } 8286 \text{ D}$			35
$\text{DS } 8287 \text{ D}$			25

$\overline{S_1}, S_2 \text{ oder MD} \rightarrow Y \text{ DS } 8212 \text{ D}$	t_{PZX} in ns		45
$\text{DS } 8216 \text{ D}$			65
$\overline{\text{OE}} \rightarrow B \text{ DS } 8282 \text{ D}$		10	50
$\text{DS } 8283 \text{ D}$		10	50
$\overline{\text{OE}} \rightarrow \overline{B} \text{ DS } 8286 \text{ D}$		10	50
$\text{DS } 8287 \text{ D}$		10	50

$\overline{S_1}, S_2 \text{ oder MD} \rightarrow Y \text{ DS } 8212 \text{ D}$	t_{PXZ} in ns		45
$\text{DS } 8216 \text{ D}$			35
$\overline{\text{OE}} \rightarrow B \text{ DS } 8282 \text{ D}$			25
$\text{DS } 8283 \text{ D}$			25
$\overline{\text{OE}} \rightarrow B \text{ DS } 8286 \text{ D}$			25
$\text{DS } 8287 \text{ D}$			25

Betriebsbedingungen

Betriebsspannung	$-U_s$ in V	min. 4,75	max. 5,25
L-Ausgangsstrom	I_{OL} in mA		
$\text{DS } 8205 \text{ D}$			1,5
$\text{DS } 8212 \text{ D}, \text{ DS } 8216 \text{ D (A-Ausgänge)}$			15
$\text{DS } 8216 \text{ D (B-Ausgänge)}$			55
$\text{DS } 8282 \text{ D}, \text{ DS } 8283 \text{ D}, \text{ DS } 8286 \text{ D/DS } 8287 \text{ D}$			
(Anschluß B)			32
$\text{DS } 8286 \text{ D/DS } 8287 \text{ D (Anschluß A)}$			16
H-Ausgangsstrom	$-I_{\text{OH}}$ in mA		
$\text{DS } 8205 \text{ D}$			10
$\text{DS } 8212 \text{ D}, \text{ DS } 8216 \text{ D (A-Ausgänge)}$			
$\text{DS } 8286 \text{ D/DS } 8287 \text{ D (Anschluß A)}$			1
$\text{DS } 8216 \text{ D (B-Ausgänge)}$			10
$\text{DS } 8282 \text{ D}, \text{ DS } 8283 \text{ D}, \text{ DS } 8286 \text{ D/DS } 8287 \text{ D (Anschluß B)}$			5

Impulswerte DS 8212 D DS.8282 D, DS 8283 D	t_w in ns	30 15	
Hold-Zeit DS 8212 D DS 8282 D, DS 8283 D	t_H in ns	20 25	
Voreinstellzeit, DS 8282 D, DS 8283 D DS 8212 D	t_v in ns	0 15	
Umgebungstemperatur	ϑ_a in °C	0	+70
Grenzwerte, gültig für den Betriebstemperaturbereich			
Betriebsspannung Eingangsspannung	U_s U_1 in V	min. 0	max. 7 5,5

Tabelle 5.7. Logiktablelle DS 8212

Funktion	CLR	MD	S ₁	S ₂	STB	Daten- eingang	Daten- ausgang
CLEAR	L	H	H	X	X	X	L
	L	L	L	H	L	X	L
Hochohmiger Zustand	X	L	X	L	X	X	Z
Speichern	X	L	H	X	X	X	Z
	H	H	H	L	X	X	Q ₀
	H	L	L	H	L	X	Q ₀
Datenbus	H	H	L	H	X	L	L
	H	H	L	H	X	H	H
Datenbus	H	L	L	H	H	L	L
	H	L	L	H	H	H	H

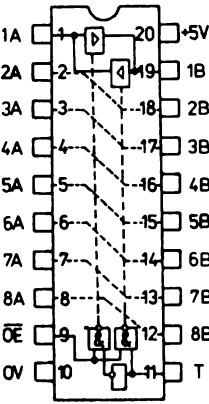
Status-Flip-Flop	CLR	S ₁	S ₂	STB	INT
	L	H	X	X	H
	L	X	L	X	H
	H	X	X	↓	L
	H	L	H	X	L

X = Pegel beliebig

5. Anhang

Tabelle 5.8. Logiktablelle
DS 8216/8286

DIR	OE	
L	L	DI → B
H	L	B → A
L	H	
H	H	hochohmig



*DS 8287 invertierende
Ausgänge

Bild 5.68
DS 8286/DS 8287

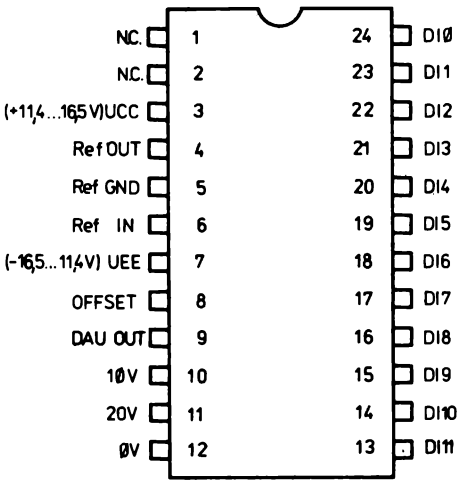


Bild 5.69 C 5658 D

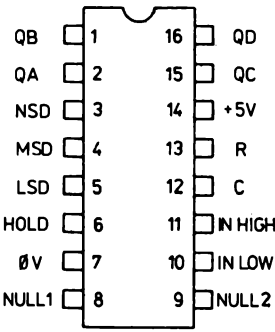


Bild 5.70 C 520 D

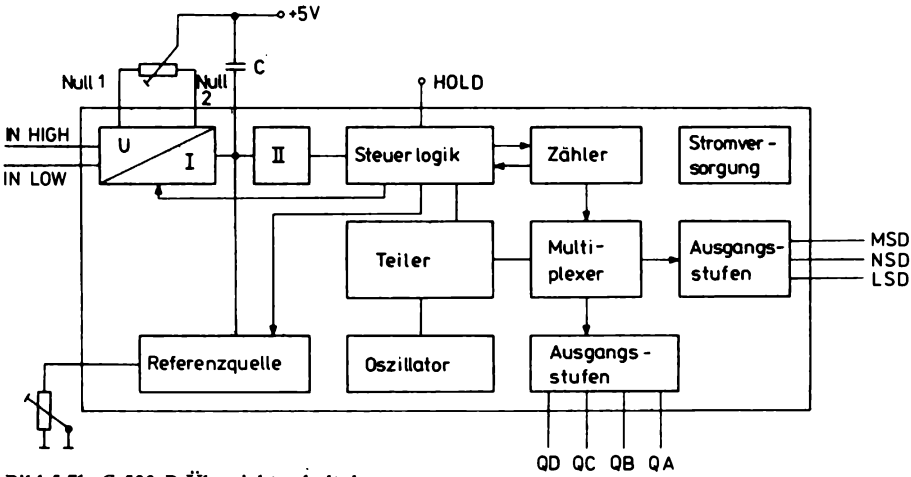
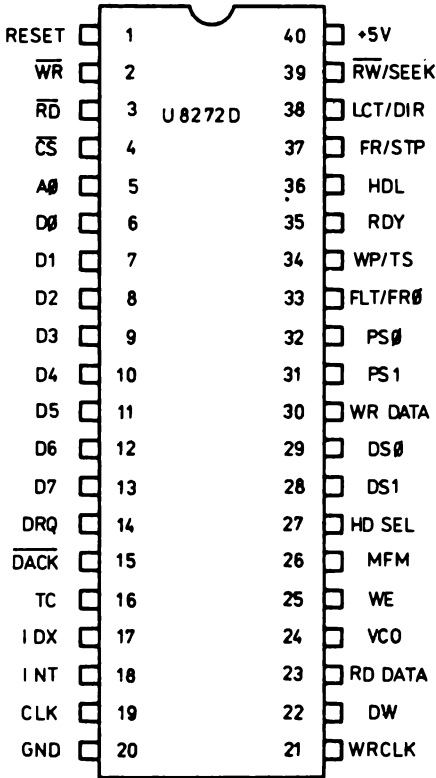


Bild 5.71 C 520 D-Übersichtsschaltplan



UB 857 D	DDR	Bild 5.22/
Z 80 CTC	NSW	Seite 219
MK 3882	NSW	
UB 858 D	DDR	Bild 5.23/
Z 80 DMA	NSW	Seite 219
MK 3883	NSW	
U 555 C	DDR	Bild 5.25/
2708	NSW	Seite 220
K 573 RF 1	UdSSR	
U 552 C	DDR	Bild 5.26/
1702 A	NSW	Seite 222
8702	NSW	
U 402 D	DDR	Bild 5.29/
TMS 2500	NSW	Seite 224
U 2716	DDR	Bild 5.30/
K 573 RF 5	UdSSR	Seite 224
U 256 D	DDR	Bild 5.37/
4116	NSW	Seite 226
K 565 RU 3A	UdSSR	
U 202 D	DDR	Bild 5.41/
2102	NSW	Seite 230
K 565 RU 2	UdSSR	
U 224 D	DDR	Bild 5.42/
6514	NSW	Seite 230
444	NSW	

Bild 5.72 U 8272 D

Tabelle 5.9. Schaltkreisvergleichsliste

Typ	Hersteller (Land)			
UB 880 D	DDR	Bild 5.1/		
Z 80 CPU	NSW	Seite 205		
MK 3880	NSW			
UB 855 D	DDR	Bild 5.10/		
Z 80 PIO	NSW	Seite 209		
MK 3881	NSW			
UB 8560 D	DDR	Bild 5.17/		
Z 80 SIO/0	NSW	Seite 213		
MK 3884	NSW			
UB 8561 D	DDR	Bild 5.18/		
Z 80 SIO/1	NSW	Seite 214		
UB 8563 D	DDR	Bild 5.19/		
Z 80 DART	NSW	Seite 214		
8255	ČSSR, VR Polen	Bild 5.47/		
K 580 IK 55	UdSSR	Seite 231		
8251	NSW, ČSSR, VR Polen	Bild 5.52/		
K 580 IK 51	UdSSR	Seite 234		
UB 8820 M	DDR	Bild 5.57/		
Z 8/64	NSW	Seite 237		
DS 8212	DDR	Bild 5.64/		
K 589 IK 12	UdSSR	Seite 240		
DS 8216	DDR	Bild 5.66/		
K 589 AP 16	UdSSR	Seite 241		
C 520 D	DDR	Bild 5.70/		
AD 2020	NSW	Seite 246		

Tabelle 5.10. Fehlermeldungen des U8272

reg/bit	Name	Symbol	Beschreibung
0 7 0 6	Interrupt Code	IC	bit 7 = bit 6 = 0 (normaler Abschluß des Kommandos, keine Fehler) bit 7 = 0, bit 6 = 1 (fehlerhafter Abschluß) bit 7 = 1, bit 6 = 0 (fehlerhaftes Kommando) bit 7 = bit 6 = 1 (anormaler Abschluß wegen Statusänderung des Laufwerks)
0 5	Seek End	SE	bit schaltet am Ende des Kommandos <i>Spur suchen</i> auf H
0 4	Equipment Check	EC	bit schaltet bei Fehlersignal vom Laufwerk auf H
0 3	Not Ready	NR	bit schaltet, wenn das Laufwerk im Nicht-Bereit-Status ist
0 2	Head Address	HD	bit zeigt den Status des Kopfes bei Interrupt
0 1 0 0	Unit Select	USA US0	bits zeigen die Gerätenummer beim Interrupt an
1 7 1 6	End of Cylinder	EN	bit schaltet bei Zugriff auf einen Sektor größer als der letzte der Spur nicht benutzt, immer L
1 5	Data Error	DE	bit schaltet bei CRC-Fehler im Daten- oder ID-Feld auf H
1 4	Over Run	OR	Datenverkehr mit CPU zu langsam
1 3			nicht benutzt, immer L
1 2	No Data	ND	bit wird gesetzt: a – bei Ausführung der Kommandos <i>Daten lesen</i> , <i>gelöschte Daten schreiben</i> und <i>Daten suchen</i> , wenn der Controller den im ID-Register gespeicherten Wert nicht findet, b – bei fehlerhafter Ausführung des Kommandos <i>ID lesen</i> , c – wenn beim Kommando <i>Spur lesen</i> der erste Sektor nicht gefunden wird
1 1	Not Writeable	NW	bit schaltet, wenn bei den Kommandos <i>Daten schreiben</i> , <i>gelöschte Daten schreiben</i> und <i>Spur formatieren</i> das Laufwerk ein Schreibschutzsignal abgibt
1 0	Missing Address Mark	MA	bit wird auf H gesetzt: a – wenn der Controller die ID-Adreßmarke bis zum erneuten Auftreten des Indexloches nicht findet, b – wenn der Controller die Datenadreßmarke nicht findet
2 7 2 6	Control Mark	CM	nicht benutzt bit schaltet, wenn bei den Kommandos <i>Daten lesen</i> und <i>Daten suchen</i> ein Sektor mit gelöschter Datenadreßmarke auftritt
2 5	Data Error	DD	bit schaltet bei CRC-Fehler im Datenfeld
2 4	Wrong Cylinder	WC	bit schaltet, wenn die Werte von C auf der Diskette und der im IDR gespeicherte sich unterscheiden
2 3	Scan Equal Hit	SH	beim Kommando <i>Daten suchen</i> schaltet bit bei Übereinstimmung auf H
2 2	Scan Not Satisfied	SN	bit schaltet beim Kommando <i>Daten suchen</i> , wenn auf der Spur kein Sektor die Bedingung erfüllt
2 1	Bad Cylinder Missing	BC	wenn der Wert von C auf der Diskette, mit dem im IDR gespeicherten nicht übereinstimmt und FF ist, schaltet bit auf H
2 0	Address Mark	MD	bit schaltet, wenn der Controller beim Datenlesen von der Diskette keine Datenadreßmarke und keine gelöschte Adreßmarke findet
3 7	Fault	FT	bit zeigt Fehlersignal des Laufwerks an
3 6	Write Protected	WP	bit zeigt Schreibschutz-Statussignal des Laufwerks an
3 5	Ready	RDY	bit zeigt Bereitschaftsstatus des Laufwerks an
3 4	Track0	T0	bit zeigt Statussignal Spur 0 des Laufwerks an
3 3	Two Side	TS	bit zeigt Statussignal des Laufwerks an
3 2	Head	HD	bit zeigt Seitenauswahl des Laufwerks an
3 1	Unit1	US1	bits zeigen Geräteauswahl an
3 0	Unit0	US0	

5.5. Leiterplattenlayouts

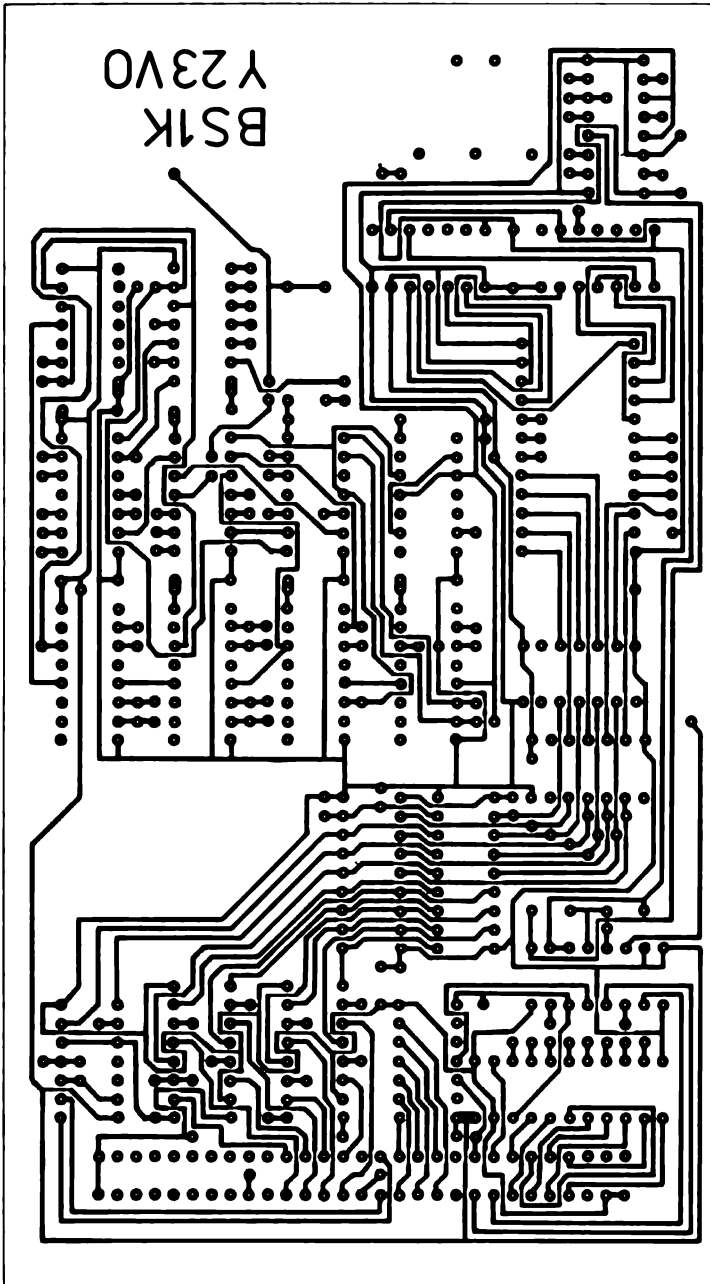


Bild 1.16a Bildschirmansteuerung

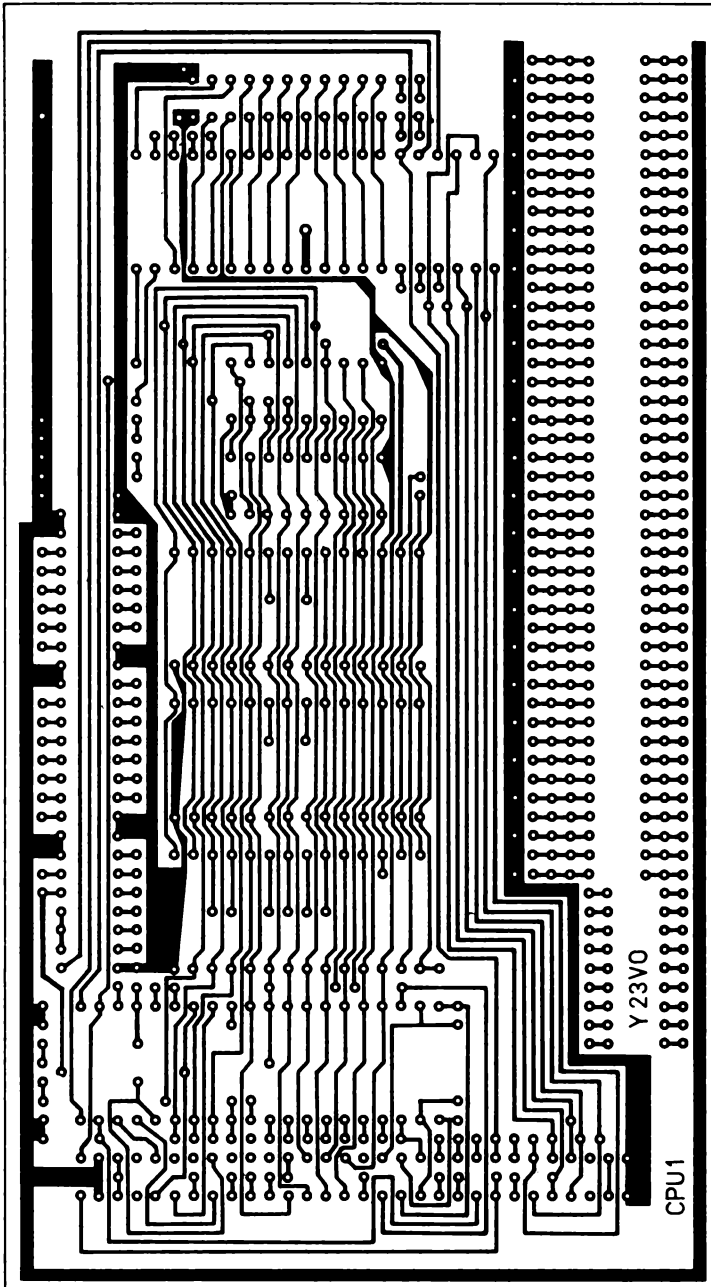


Bild 1.23 Leiterplatte CPU 1

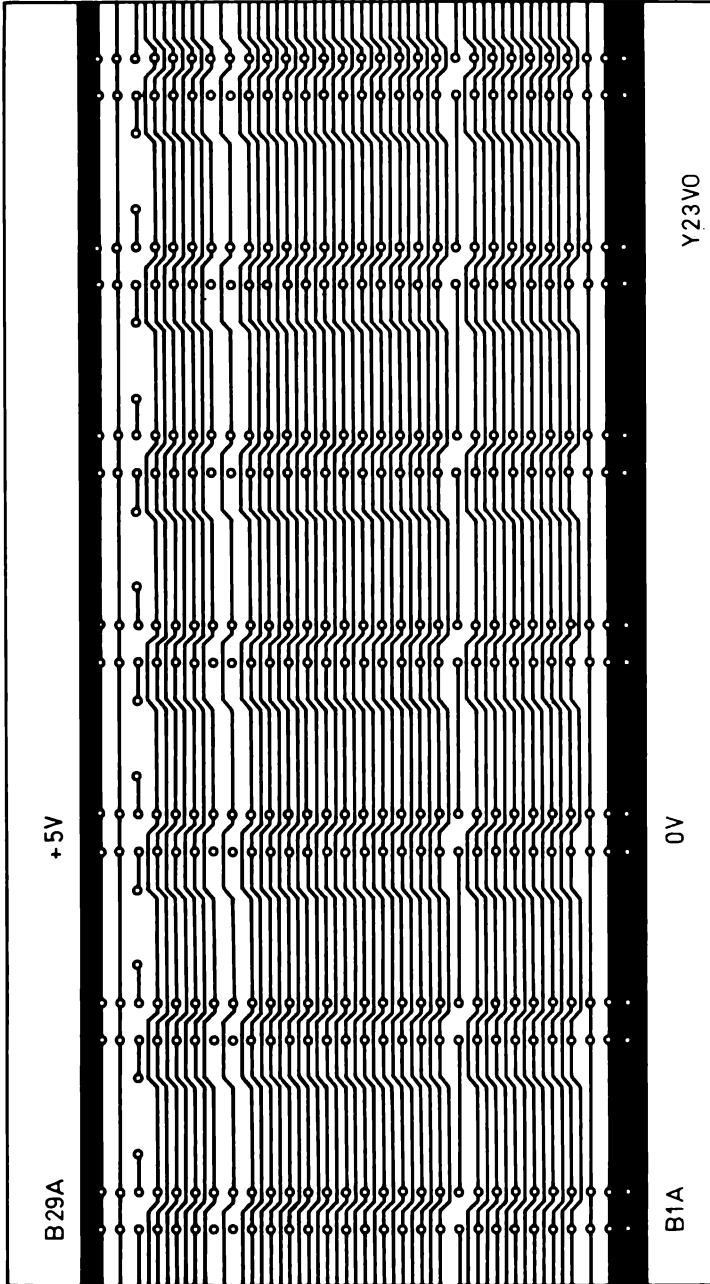


Bild 1.25 Leiterplatte für die Rückverdrahtung

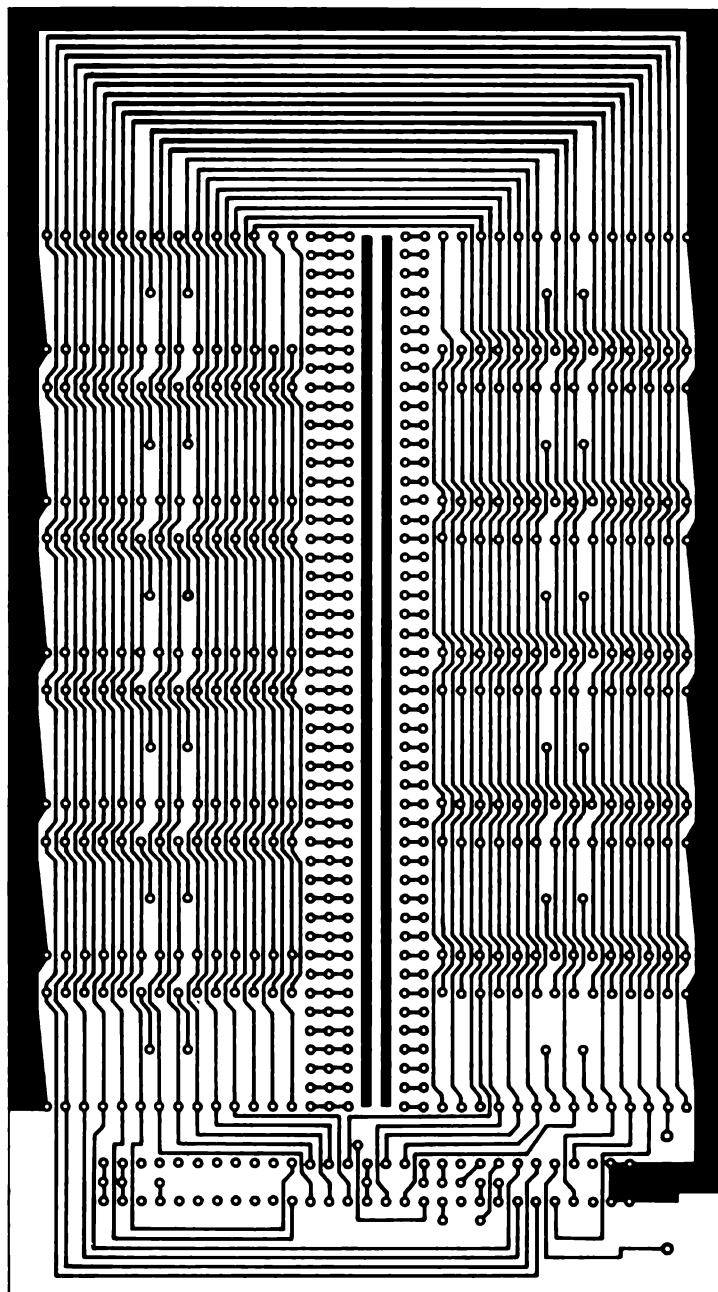


Bild 3.2 Speicherkarte MEM 1

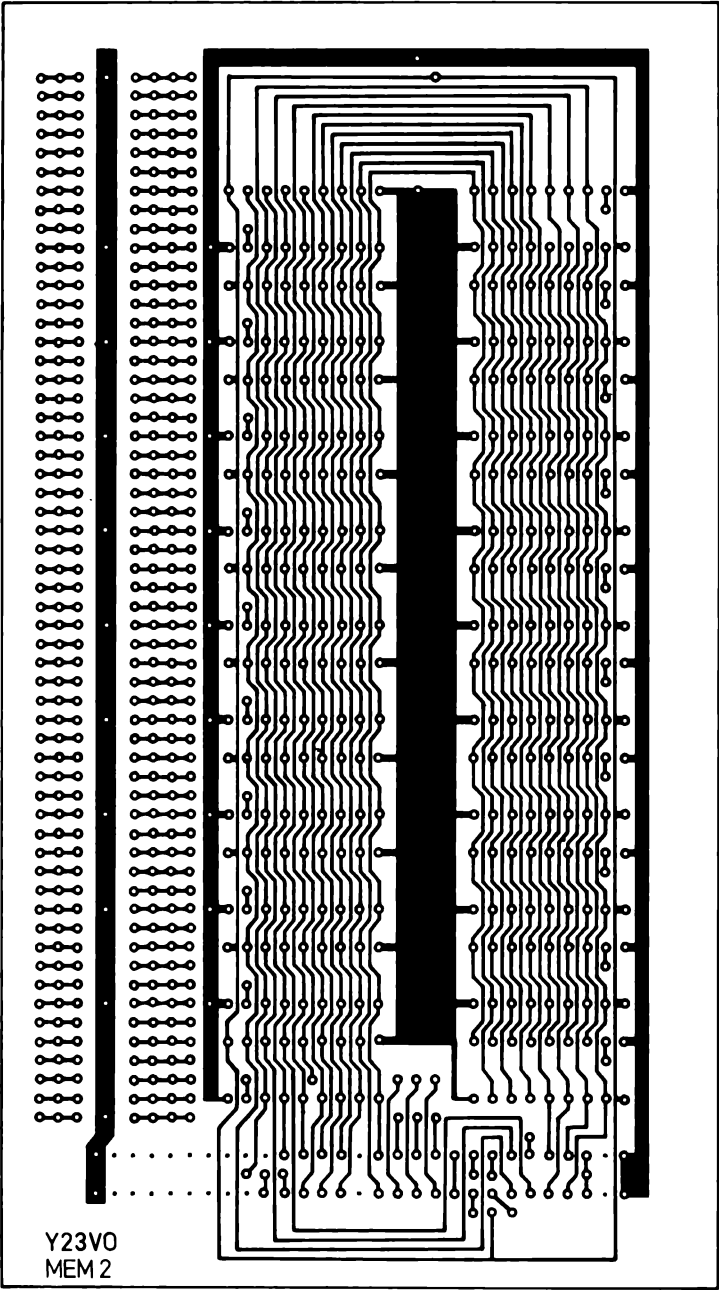


Bild 3.4a
Speicherkarte
MEM 2

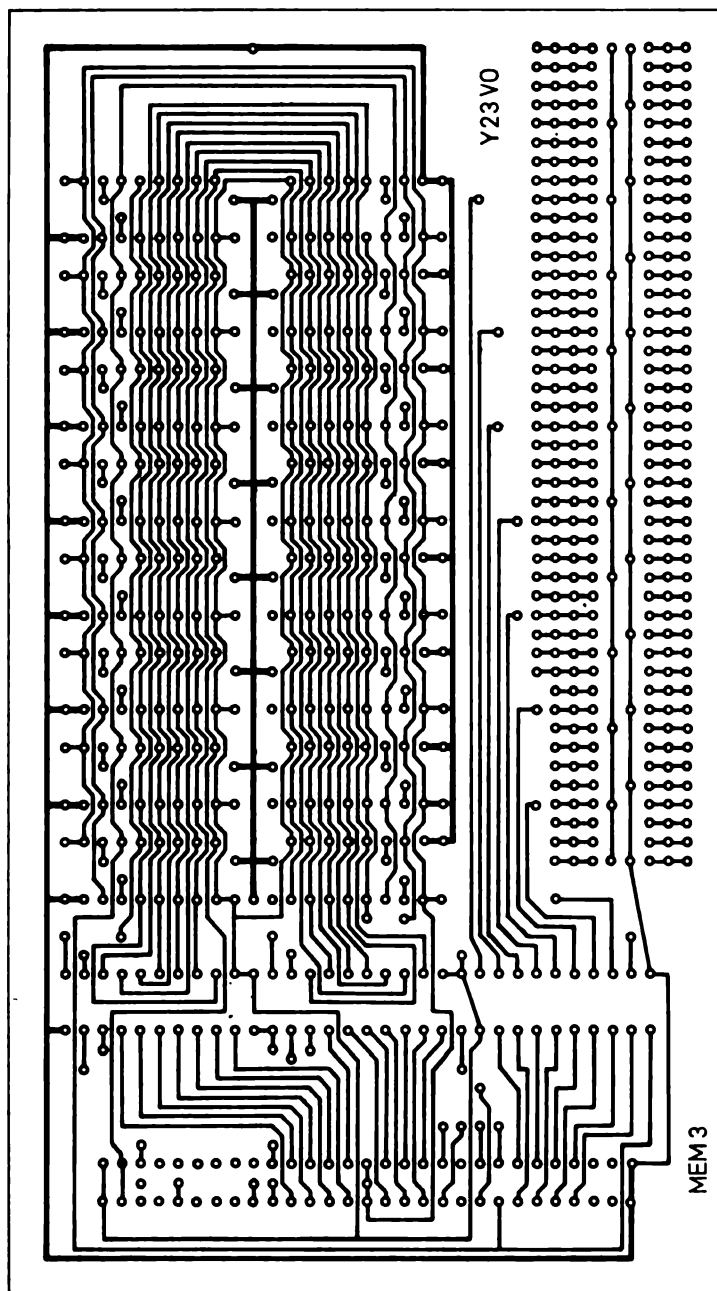


Bild 3.7 a
Speicherkarte
MEM 3

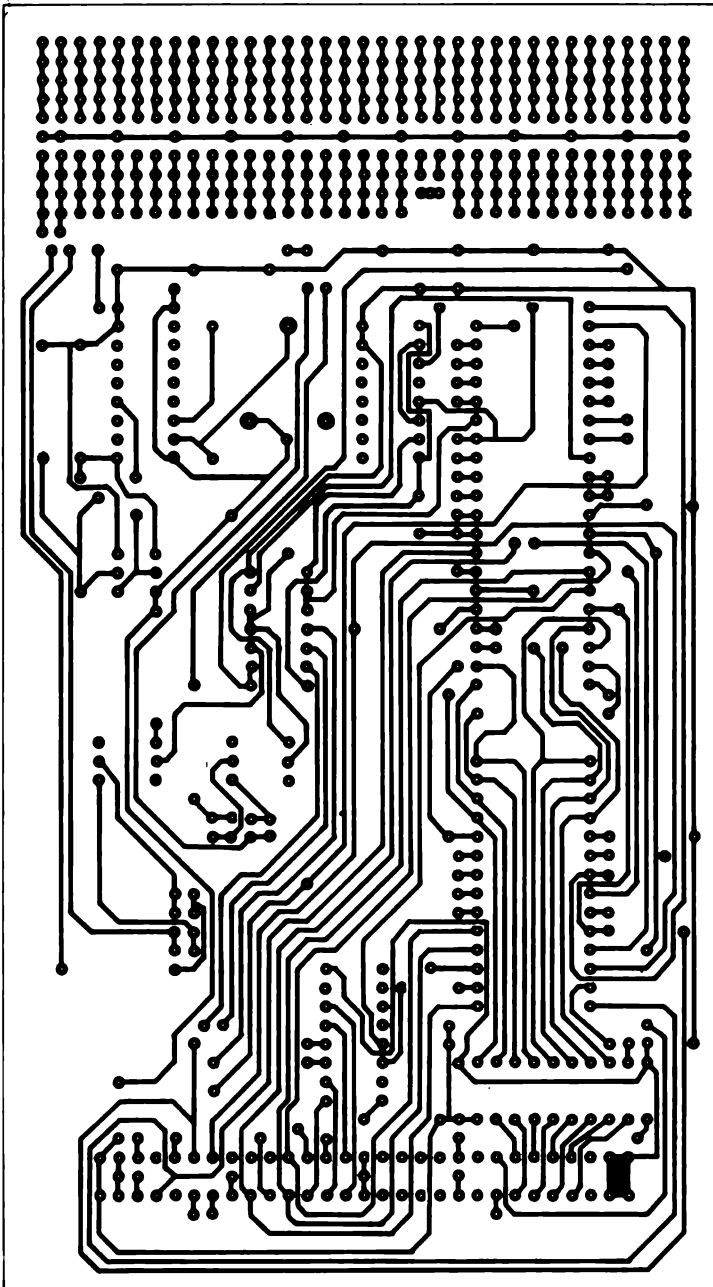


Bild 3.20 c CTC/SIO-Karte

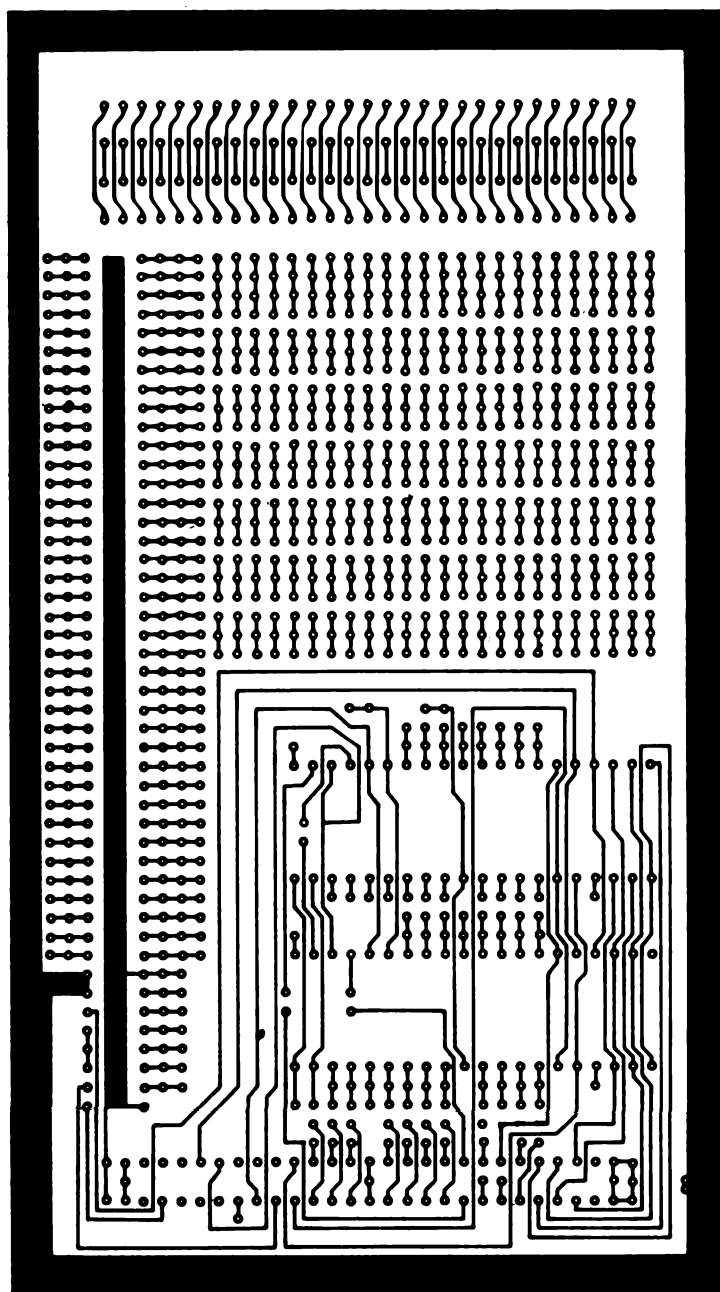


Bild 3.24 Leiterplatte 2P10

6. Literaturverzeichnis

- [1] Klein, R. D.: Datensichtgerät zum Selbstbau. Elektronik 1977, Heft 1, S. 59–61
- [2] Klein, R. D.: BASIC-Interpreter. München 1982
- [3] Duncan, F. G.: Mikroprozessor-Software. München 1980
- [4] Gottfried, B. S.: Programmieren in BASIC. Düsseldorf 1978
- [5] TGL 37787: Festwertspeicher U 555 C
- [6] TGL 37271: Linieninterface BUS K 1520
- [7] Kramer, M.: Mikrorechnersteuerung mit serieller Schnittstelle. rfe 32 (1982), Heft 8
- [8] Dynamische Speicher. mc 1982
- [9] VEB Polytechnik: Polycomputer 880, Beschreibung. Karl-Marx-Stadt 1983
- [10] Strelocke, K.; Hoffmann, P.: Die Dialogsprache BASIC. Berlin 1981
- [11] Stengelmeier, W.: Disassembler für 8080. Hobby-Computer 2 (1983), S. 67–69
- [12] Butkewitsch, A. W.; Selikson, M. S.: Ewige Kalender, Leipzig 1982
- [13] Stuhlmüller, P.: 16-Bit-Generation Z 8000. München 1982.
- [14] –: Kassetteninterface. Radio 59 (1983), Heft 9
- [15] Heyder, F.: Amateurcomputer 1. Funkamateure 33 (1984), Artikelserie
- [16] –: LC 80, Beschreibung. rfe 34 (1984), Heft 10
- [17] Siemens AG: Mikroprozessor-Bausteine, München 1977
- [18] VEB Robotron-Elektronik Radeberg: Service-Mitteilung Combi-Vision 3101. Dresden 1977
- [19] Kruse, R.: Schwarzweiß-Fernseher mit Halbleiterbauelementen. In: Schaltungssammlung für den Amateur. Zweite Lieferung. Herausg. v. K. Schlenzig und W. Stämmler, Berlin 1979
- [20] VEB Funkwerk Erfurt: Befehlsbeschreibung U 880 D. Erfurt 1981
- [21] ZILOG Inc.: Z80-Assembly Language Programming Manual. Cupertino (USA) 1977
- [22] VEB Robotron Buchungsmaschinenwerk: BASIC-Interpreter für SIOS 1526. Karl-Marx-Stadt 1983
- [23] Daulbach, G.: Microsoft-BASIC 80. Vaterstetten 1983
- [24] Lesea; Zaks, R.: Mikroprozessor-Interface-Techniken. Vaterstetten 1981
- [25] VEB Kombinat Mikroelektronik: Aktive elektronische Bauelemente. Erfurt 1984
- [26] VEB Kombinat Mikroelektronik: Halbleiter-Bauelemente, Kurzinformation (Vergleichsliste). Erfurt 1983
- [27] Zaks, R.: CP/M-Handbuch. Düsseldorf 1981

7. Sachwortverzeichnis

- Abfrage eines Tastenkontaktes 9
- Adresse 10
- Adressenmultiplexer 27
- adressierbarer Speicherbereich 11
- Adreßregister 52
- A/D-Wandler 106
- , Programm für 109
- AF-Operationen 58
- Akkumulator 54
- Akkumulatorenbatterie 78
- ALOAD 49
- Amateurfunkstation 26
- Amateurfunkwettkampf 47
- AMERGE 49
- AND-Befehl 57
- Anschlußsteuerung 119
- Anwenderprogramm 115
- Apostroph 45
- Arbeitsspeicher 115
- Arithmetik-Logik-Einheit 52
- Arithmetische Befehle 56
- ASAVE 49
- ASC 47
- Assembler 72
- , Programmausdruck 182
- asynchrone Datenübertragung 25
- asynchrones, seriellcs Zeichen 25
- Aufbau eines Computers 10
- Auffrischregister 52
- Aufrufschleife 65
- Aufzeichnungsfrequenz 112
- Aufzeichnungsgeschwindigkeit 104
- Ausgabebefehle 63
- Ausführungsphase 120
- Ausgangsfrequenz 89
- Ausgangslastfaktor 113
- Ausgangsstrom bei TTL-Last 77
- Austauschbefehle 35
- AUTO 48

- Backstep 71
- BASIC 41
- BASIC-Dialekte 42
- BASIC-Interpreter 41
- , Bedienung 48
- , Programmausdruck 166
- , Startadresse 43

- BAS-Mischer 28
- BDOS 119
- Befehlssatz 10, 52
- Befehlszähler 10
- Befehlszeile 44
- Betriebssystem CP/M 82
- Betriebssystem für Floppy-Disk 117
- bidirektional 16
- Bilddemodulator 34
- Bildpunkte 27
- Bildqualität 34
- Bildschirm 20
- Bildschirmsteuerung 27
- Bildschirm Ausgabe 71
- Bildschirmgerät 30
- Bildschirm löschen 65
- Bildschirmzeile 27
- Bildsynchronimpulse 30
- Bildwiederholtspeicher 27, 28
- binäre Addition 57
- BIT 62
- Bitbefehle 62
- bit-organisierter Speicher 81
- Blockladebefehle 56
- Blocklänge 69
- Blockvergleichsbefehle 56
- Bus-Anforderungs-/ Bus-Bestätigungszyklus 207
- Buspuffer 77
- BUS-System 10, 16
- Byte-organisierte Speicher 78
- B 3170 35
- B 555 D 112

- CALL 45, 64
- CAS 81
- CBIOS 119
- , Programmausdruck 186
- CCF 135, 58
- CCP 119
- Chip-Select-Signal 15
- CHR\$ 48
- CMOS-Multiplexer 109
- CMOS-Technologie 78
- CMP (ii + d) 131, 57
- CMP M 131, 57
- CMP n 131, 57
- CMP r 131, 57

Computergrafik 106
 Contest 47
 CONTINUE 48
 Control 72
 COPY 48
 CPD 129, 56
 CPDR 130, 57
 CPI 129, 56
 CPIR 129, 56
 CPL 134, 58
 CP/M-Speicheraufteilung 118
 CRT-Controller 113
 Cursor 71
 C 520 D 106
 –, Anschlußbelegung 246
 C 5658 D 106
 –, Anschlußbelegung 246

 DAA 134, 58
 DATA 45
 Datei 117
 Dateiverwaltung 118
 Daten 10
 Datenblock 69
 Datensichtgerät 30
 Datenübertragung, asynchron 25
 Datenübertragung, seriell 101
 D/A-Wandler 106
 Debugger 66
 –, Programmausdruck 150
 DEC dd 134, 57
 DEC ii 134, 57
 DEC (ii + d) 132, 57
 DEC M 132, 57
 Decoder 106
 DEC r 132, 57
 DEF FN 46
 Dekodierung der Adreßleitungen 78
 Dekodierung der CS-Signale 78
 DELETE 48
 D-Flip-Flop 104
 DI 58
 DIL-Schalter 78
 DIR 119
 Direktmodus 43
 Disassembler, Programmausdruck 164
 Disketten 116
 Doppelpunkt 69
 doppelte Schreibdichte 120
 DRAM 79
 Drucker 20
 DS 8205 D 14, 240
 –, Technische Daten 242f.
 DS 8212 D 113
 –, Anschlußbelegung 240
 –, Logiktablelle 245
 –, Technische Daten 242f.
 –, Übersichtsschaltplan 241
 DS 8216 D 113
 –, Anschlußbelegung 241
 –, Logiktablelle 246
 –, Technische Daten 242f.
 DS 8282/83 D 113
 –, Anschlußbelegung 241
 –, Technische Daten 242f.
 DS 8286/87 D 30, 113
 –, Anschlußbelegung 246
 –, Logiktablelle 246
 –, Technische Daten 242f.
 Duplex-Verkehr 26
 Durchgangsprüfer 38
 dynamische RAM 79
 D 121 D 112

 EDIT 49
 Editor 72
 –, Programmausdruck 172
 EI 58
 Einchip-Mikrocomputer 110, 114
 –, Befehlstablelle 238
 –, Registerblock 237
 Eingabebefehle 63
 Eingabe- oder Ausgabezyklus 207
 Eingabe-Routine 101
 Einleseprogramm 69
 Einsprungtablelle 65, 66
 Einzelschrittsteuerung 39
 Elektronikleiterplatten 37
 ELSE 45
 Empfangsfernseher *F 1219* 120
 Ende-Impuls 122
 Entprellschaltung 9
 EPROM 11
 – Programmierzusatz 85
 ERA 119
 Ewiger Kalender 50
 EXAF 127
 EX DE, HL 127
 Exklusiv-Oder-Gatter 112
 EX (SP), HL 127
 EX (SP), ii 127
 EXX 127

 Farbfernsehergeräte 34
 Fehlersuche 39
 Fernschreiber 20
 Fernschreibanschluß 101
 Festspannungsregler 35
 Flag-Register 53
 –, Beeinflussung der 59
 Flip-Flop-Schaltungen 80
 Floppy-Disk 116

7. Sachwortverzeichnis

Floppy-Disk-Controller 120
FNEND 46
Folienspeicher 116
Fototransistoren 106
Frequenzmodulation 104
Funkuhr 110

Gleichlaufschwankungen 112
Gleitkommazahlen 43
GOSUB 45
GOTO 44, 49
Gregorianischer Kalender 51
Grundgerät 11

HALT 135, 58
-, Ablauf 209
Handleser für Lochstreifen 104
Hardware 24
Hauptstatusregister 122
hexadezimale Schreibweise 9
HF-Oszillator 34
hochohmiger Zustand 14
Horizontaltabulator 71

IEI 16
IEO 16
IM0 58
IM1 58
IM2 58
Inbetriebnahme 38
INC dd 133
INC ii 133
INC (ii + d) 132, 57
INC M 132, 57
INC r 132, 57
IND 136, 63
INDR 136, 63
Index 43
Index-Loch 116
Index-Register 52
indirekte Adressierung 56
Industriestandard 11
INF 137
Informationsaustausch 24
Infrarotemitterdioden 106
INI 137, 63
INIR 136, 63
Initialisierung 24, 65
IN n 135
IN/OUT-Adressen 15
IN/OUT-Anschlüsse 15
IN/OUT-Zuweisung 66
INP 45
IN r 135
INSTR 48
INT (Interrupt) 16, 91

Integrationsgrad 113
Interface-Schaltkreise 113
Interruptprogramm 91
Interrupt-Quittungs-Zyklus 212
Interruptzyklus 208
IORQ 14

JP 64

Kassetteninterface mit 8251 111
Kassettenrecorder 20, 27, 112
Keller 52
Kellerzeiger 52, 70
Kodierung von Signalen 9
Komma 44, 66
Kommandophase 120
Kontaktprellen 9
kontinuierlicher Regler 35
Konvertierung 65
Konvertierungstabelle 72
Koppel-BUS 17

Ladebefehle 55
Ladungsspeicherung 80
Latch 113
Laufwerksteuerung 116, 123
LDD 128, 56
LDDR 128, 56
LDI 128, 56
LDIR 128, 56
Leerzeichen 44
LEN 47
Lese-Programm 97
Leseregister 218
LET 46
Linefeed 71
LINEINPUT 48
LIST 48
LLIST 48
LLVAR 46
Lochstreifen 104
Lochstreifensender 21
Logarithmenrechnung 43
Logikbefehle 56
Logikpegelanzeige 38
logische Funktion 43
LPRINT 44
LTRACE1 49
LVAR 46
LWIDTH 44

MAA 723 35
Mäanderwelle 104
Marke 75
Maschinencode 52
Maschinensprache 52

Matrixschaltung 25
 MEMDI 16
 Meßgerät 110
 Meßstellenumschalter 109
 MID\$ 48
 Mikrodisketten 116
 Mikroprozessor 8080 110
 Mikroprozessor 8085 110
 Mikroprozessorsysteme 110
 Minidisketten 116
 Minidiskettenlaufwerk 116
 Mnemonik 52
 Modulationssignal 27
 Modulationsverfahren 101
 Monitorprogramm 64
 –, Assemblerausdruck 139
 –, Maschinencode 149
 –, Maschinencode für Adresse 0F000H 199
 Monitortabellen 71
 Monoflop, nachtriggerbar 104
 monostabiler Multivibrator 9, 112
 MREQ 14
 Multiplexer 81
 Multiplexbetrieb 106
 Multiplexsteuerung 106

 NEG 135, 58
 New line 71
 NEXT 44
 NF-Modulation 104
 NF-Schwingungen 101
 nichtmaskierbarer Interrupt 208
 NOP 135, 58
 Nutzerbereiche 119

 offener Kollektor 20
 Operanden des Befehls 44
 Operationscode 205
 –, mit WAIT 206
 OR-Befehl 57
 Originaladresse 82
 Oszilloscop 38
 OTDR 138, 63
 OTIR 138, 63
 OUT 45
 OUT D 137, 63
 OUTI 137, 63
 OUT r 137

 Parallel-Serien-Wandler 28
 PEEK 45
 Peripherie 10, 64
 Peripherieschaltkreise 110
 Phasenmodulation 104, 112
 physischer Treiber 119
 Pipeline-Verfahren 115

 POKE 45
 POP 55
 POS 48
 Potenzrechnung 43
 PRINT 44
 Priorität 91
 Programm 10
 Programmablaufplan 8
 Programmbefehle 42
 Programm Ewiger Kalender 50
 Programme für serielle Datenübertragung 97
 Programmieren von EPROMs 69
 Programmierphase 120
 Programmiersprache 41
 Programmierung 41
 Programm Regelkreis 51
 Programmschleife 44
 Programmtest 41
 Programmunterbrechung 115
 Programmzähler 52
 Prüfsumme 69
 Pufferschaltkreise 77
 PUSH 55

 quasi-Grafikzeichen 48
 Quelltext 72
 Quittieren der Datenübergabe 21, 24
 Quittungssignale 122

 RAM 15
 RAM-DISK 120
 RAM-Tabelle 67
 RAS 81
 RD (READ) 16, 45
 Readkommando 97
 Regelkreis 51
 Registerblock 115
 Registerzeiger 115
 relative Sprünge 68
 REM 45
 REN 119
 RENUMBER 49
 RES 62
 Restartbefehle 64
 RESTORE 45
 Resultatphase 120
 RET 64
 RETI 64
 RETI-Zyklus 212
 RETN 64
 RETURN 45
 RFLOP 120
 RIGHTS\$ 47
 RL 61
 RLC 61
 RLD 61

7. Sachwortverzeichnis

Rollen des Bildschirminhalts 30, 71
RR 61
RRC 61
RRD 61
RST 64
Rückkehradresse 52
Rücksetzen 16
Rücksprungbefehle 63, 64
Rückwärtszähler 89
RUN 49

SAVE 119
SCF 135, 58
Schaltnetzteil 35
Schattenregister 54
Schlüsselwort 41
Schmitt-Trigger 106
Schottky-Technologie 77
Schreib-/Lesespeicher 15
Schreibregister 215f.
Schrittbetrieb 39
Schrittmotor 116
Sektoren 116
–, Aufteilung der 117
Sektornummer 117
serielle Informationsübertragung 25
serielle Schnittstelle 25
SET 62
Shift 72
Signalbezeichnung 16
Signalverknüpfung im Prozessor 10
Signalverzögerung 113
Simulationsprogramm 51
SLA 61
Soft-Sektor-Disketten 116
Software 24
Space 44
Spaltenadressenabtastrung 81
Spaltendecoder 11
Spaltenleitungen 25
Spaltenschalter 11
Spannungsmesser 38
SPC 44
Speicher 113
Speicher abschalten 17
Speicheradressierung 54
Speicheranforderung 14
Speicheraufteilung 77
Speicherbereich, adressierbar 11
Speicherbereichsbefehle 56
Speichererweiterungen 77
Speicher lesen 10
Speicher-Lese- oder Schreibzyklus 206
Speicherumschaltung 82
Speichervorgang 116
Speicherzugriff 10

Sprachelemente 41
Sprungbefehle 63, 64
SRA 61
SRL 61
Standard-Disketten 116
Startadresse 68, 70
Steckerbelegung 16
Steckverbinder 11
STEP 44
Steuerbefehle 58
Steuerleitungen 10
STOP 49
Stopbit 97
Stringbefehle 47
Stringvariable 43
STROBE 24
Stromversorgung 35
STR\$ 48
Substitution 68
Synchronisation 116
Synchronsignale 28
syntaktische Fehler 74
Systembefehle 42
System-BUS 17

Takt 16
Taktfrequenz 27
Tastatur 20
Tastatureingabe 72
Tastaturprogramm 25, 72
Tastaturschaltung 25
Teiler 89
Teilerfaktor, programmierbar 104
Temperaturprozesse 51
Testbetriebsart 115
Tongenerator 27, 101
TRACE1 49
TRANS 120
Transceiver 113
Transportloch 106
Trennzeichen 66
Tristate-Ausgänge 14
TTL-Prüfstift 39
TYPE 119

Übersetzung 74
Übertragimpuls 89
Übertragungsgeschwindigkeit 24, 104
Übertragungsrage 116
UB 855 D 15
–, Anschlußbezeichnungen 22
–, Programmierung 21
UB 856 D 213
–, Anschlußbezeichnungen 102
–, Übersichtsschaltplan 102
UB 857 D 219

- , Anschlußbezeichnungen 96
- , Übersichtsschaltplan 89
- UB 858 D – DMA 219
- UB 880 D 205
- , Maschinenbefehle 52, 126 ff.
- , Struktur 53
- Uhrprogramm 92
- Umwandlung von BCD in ASCII-Code 109
- Universal-Leiterplatten 37
- Unterprogramm 91
- Unterprogrammaufruf 52, 63
- Unterprogrammtechnik 70
- Urlader 117
- USER 119
- USR 46
- U 202 D 81, 230
- U 214 D 15, 30
- U 215 D 231
- U 224 D 30, 230
- U 2255 D 231
- U 253 D 231
- U 256 D 81, 226
- , Lesezyklus 229
- , Schreibzyklus 230
- , Technische Daten 227
- U 401 D 223
- U 402 D 224
- U 505 D 220
- U 555 D 11, 220
- , Programmierprogramm 87
- , Programmierung 85
- , Programmierzusatz 86
- , Zeitdiagramm 87
- U 830 D 116
- U 2164 231
- U 2716 C 224
- , Programmierzusatz 87
- U 2732 C 225
- , Programmierzusatz 88
- U 8001 D 239
- U 8002 D 240
- U 8272 D 120
- , Anschlußbelegung 247
- , Anschlußbezeichnung 121
- , Befehlssatz 122
- , Mnemonik der Befehle 123
- , Statusregister 121
- , Übersichtsschaltplan 118
- U 8810 D 114
- , Anschlußbild 236
- , Struktur 115
- U 8820 D 114
- , Anschlußbild 237
- VAL 48
- Variable 43
- Vektorabgabe 91
- Vektorregister 52, 91
- Vergleichsoperationen 42
- Verschiebepfeile 61
- Vertikaltabulator 71
- Verzögerungszeit 9
- Vorteiler 89
- Vorwärtssprünge 69
- V 4051 D 109
- Wagenrücklauf 48, 71
- WAIT 38
- Warmstart-Routine 119
- WFLOP 120
- WIDTH 44
- Winkelfunktionen 43
- XOR-Befehl 58
- XOR (ii + d) 130, 57
- XOR M 131, 57
- XOR n 130, 57
- XOR r 130, 57
- x-y-Schreiber 106
- Zähler 89
- Zählschaltung 27
- Zählschleife 9
- Zeichenaufbau 28
- Zeichengenerator 27
- Zeichengeneratorinhalt 165
- Zeichenkette 44
- Zeichenkettenvergleich 43
- Zeilenadressenabtastung 81
- Zeilendecoder 11
- Zeilenleitungen 25
- Zeilennummer 41
- Zeilensynchronimpulse 30
- Zeitbasis 30
- Zeitgeber 89
- Zeitkonstantenregister 89
- Zeitschleife 44
- Zeitverzögerung 67
- Zugriffsteuerung 27
- Zwischenspeicherung 11
- 1-k-Bereiche 78
- 3-Zustandsausgänge 14
- 7-bit-ASCII-Code 26
- 8-bit-Arithmetikbefehle 57
- 8-bit-Ladebefehle 54
- 16-bit-Arithmetikbefehle 58
- 16-bit-Ladebefehle 55
- 16-bit-Mikroprozessoren 116
- 886 115
- 2107 231
- 2532 225, 226

7. Sachwortverzeichnis

2764 226	–, Anschlußbild 231
–, Programmierzusatz 88	–, Betriebsarten 110
27128 226	–, Übersichtsschaltplan 234
6845 113	8257 113
8251 111, 234	8259 113
8253 113	8275 113
8255 110	9364 113

Hinweis:

Nach Redaktionsschluß wurden in den Bildern 1.14 und 1.15 noch einige Fehler bemerkt.

Bitte entsprechend korrigieren!

Bild 1.14

- Gatter 42: Anschlüsse 10, 9, 8
- IS 16: T-Ausgang nach Bild 1.15 von ÜR; ÜR-Anschluß 13; Stift 1 nur bei 9-MHz-Quarz an Masse
- Gatter 52 (Anschlüsse von oben nach unten): 12, 9, 10, 13; Anschluß 10 (neu) von CS ab, an 2/52

Bild 1.15

- IS 31 lies 51 (*DL 193 D*)
- Kondensatoren abgleichen: 0...1 nF

