

## 2. BASIC-Kleincomputer mit Grafik

*Bernd Hübler*

Dieser Abschnitt wendet sich vor allem an den experimentierfreudigen Amateur, der sich mit möglichst geringem Aufwand einen Kleincomputer aufbauen möchte. Das nachfolgend beschriebene Mikrocomputersystem entspricht in seiner Leistungsfähigkeit weitestgehend den zum Zeitpunkt der Manuskripterstellung bekannten Kleincomputern. Die angedeuteten Erweiterungsvarianten erlauben dem Leser, sich aus den gegebenen Schaltungsvorschlägen und der gebotenen Software »seinen« Computer aufzubauen.

### 2.1. Systembeschreibung

Bild 2.1 zeigt den prinzipiellen Aufbau des BASIC-Kleincomputers. Die einzelnen Funktionsgruppen werden, obwohl eine enge funktionelle Kopplung vorhanden ist, in einzelnen Abschnitten beschrieben. Die CPU, ein Mikroprozessor vom Typ U880, bestimmt im wesentlichen die Leistungsfähigkeit des Systems. Die CPU hat Zugriff auf volle 64-Kbyte-RAM sowie auf 4-Kbyte-ROM (erweiterbar auf 16 Kbyte). Der RAM-Speicherbereich besteht, wie in Bild 2.2 dargestellt, aus frei verfügbarem Arbeitsspeicher, Bildwiederholspeicher und dem Speicherbereich, der für das Betriebsprogramm reserviert ist. Das Betriebsprogramm belegt den Speicherbereich ab Adresse FOOOH. Dieses Betriebsprogramm, im weiteren auch Monitorprogramm genannt, hat vor allen Dingen die Aufgabe, die Kommunikation zwischen Mensch und Rechner zu realisieren. Weitere Hilfsmittel dafür sind eine grafische Bildschirmsteuerung und eine Tastatur. Die Tastatur ist vom eigentlichen Rechner trennt und wird über einfa-

ches Interface (parallel, auf Wunsch auch seriell) mit dem Rechnerbus verbunden. Die Tastatur benötigt nur wenig Schaltkreise und ermöglicht dennoch einen recht hohen Bedienungskomfort, z. B. automatisches Wiederholen eines Zeichens bei langem Tastendruck.

Die Bildschirmdarstellung ist voll grafisch orientiert und gestattet eine visuell attraktive Darstellung von Daten. Die Bildschirmauflösung beträgt 256 x 256 Bildpunkte, lässt sich aber nach einigen Schaltungsänderungen sowie der Softwareanpassung bis auf 512 x 256 Pixel steigern. Die Textdarstellung arbeitet in 32 Zeichen auf 24 Zeilen und kann ohne Hardwareänderung bis 42 Zeichen auf 32 Zeilen verändert werden. Die grafische Darstellung wird durch Unterprogramme unterstützt, die auch in der Lage sind, Punkte auf dem Bildschirm zu setzen oder zu löschen sowie Vektoren zu zeichnen. Eine weitere Möglichkeit der Grafikerzeugung stellt der programmierbare Zeichengenerator dar.

Der Bildwiederholspeicher ist Bestandteil des 64 Kbyte großen Arbeitsspeichers. Auf diese Art und Weise wird ein separater Bildwiederholspeicher eingespart, was für den Amateur zweifellos eine ökonomische Lösung darstellt. Da sowohl die CPU als auch die Videosteuerung sozusagen gleichzeitig auf den Arbeitsspeicher zugreifen, ist eine Synchronisationsschaltung notwendig, die die beiden Zugriffsmöglichkeiten vereinigt.

Weitere Bestandteile des nachfolgend beschriebenen Computersystems sind die Eing- und Ausgabebaugruppen. Zur freien Verfügung stehen die zwei 8-bit-Ports einer PIO, 3 CTC-Kanäle und ein SIO-Kanal. Ein SIO-Kanal wird für das eingebaute Kassettenin-

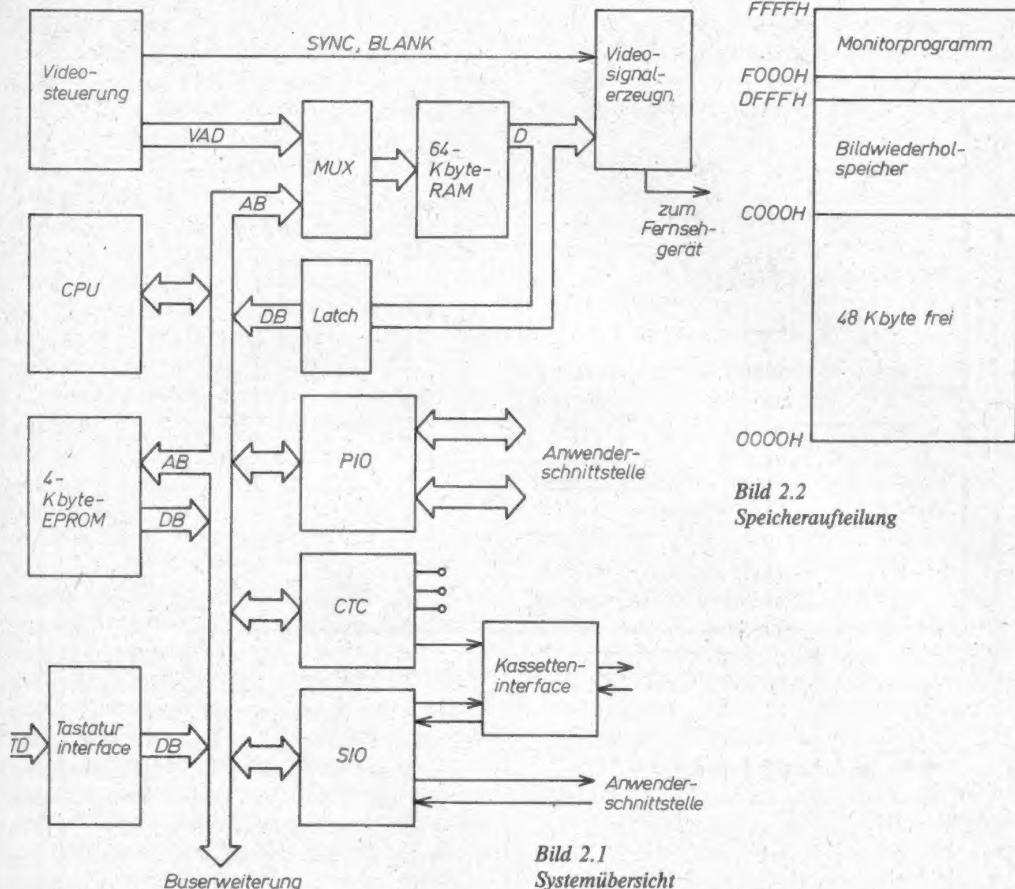


Bild 2.1  
Systemübersicht

terface genutzt, mit dem sich Daten oder Programme auf einen handelsüblichen Kassettenrecorder aufzeichnen lassen.

Zu jedem Computer gehört natürlich Software. Geboten wird hier ein einfaches Monitorprogramm und ein komfortabler BASIC-Interpreter, der den meisten Ansprüchen gerecht werden dürfte.

## 2.2. CPU-Baugruppe

Die CPU-Baugruppe stellt, wie bereits der Name sagt (Central-Processing-Unit = zentrale Verarbeitungseinheit), das Herzstück des Mikrocomputers dar. Basierend auf dem Mikroprozessor U880, bestimmt sie im wesentlichen die Leistungsfähigkeit des Systems.

### 2.2.1. Schaltungsbeschreibung

Bestimmte elementare Details des Stromlaufplans nach Bild 2.3 werden aus Gründen der Zusammengehörigkeit in anderen Abschnitten dargestellt und beschrieben (Taktversorgung).

Der Aufbau und die Wirkungsweise des U880 wird als bekannt vorausgesetzt und ist z.B. in [1] nachzulesen. Die Steuerleitungen NMI und BUSRQ werden vom System nicht benutzt und deshalb über Widerstände auf H-Potential gelegt. Die INT-Leitung hat Verbindung zu den einzelnen Peripherieschaltkreisen (PIO, SIO und CTC). Die WAIT-Leitung ist mit der RAM- bzw. Video-steuerung verbunden. Auch INT und WAIT werden, da sie L-aktiv sind, im nicht ansteuerten Zustand mit entsprechenden Wi-

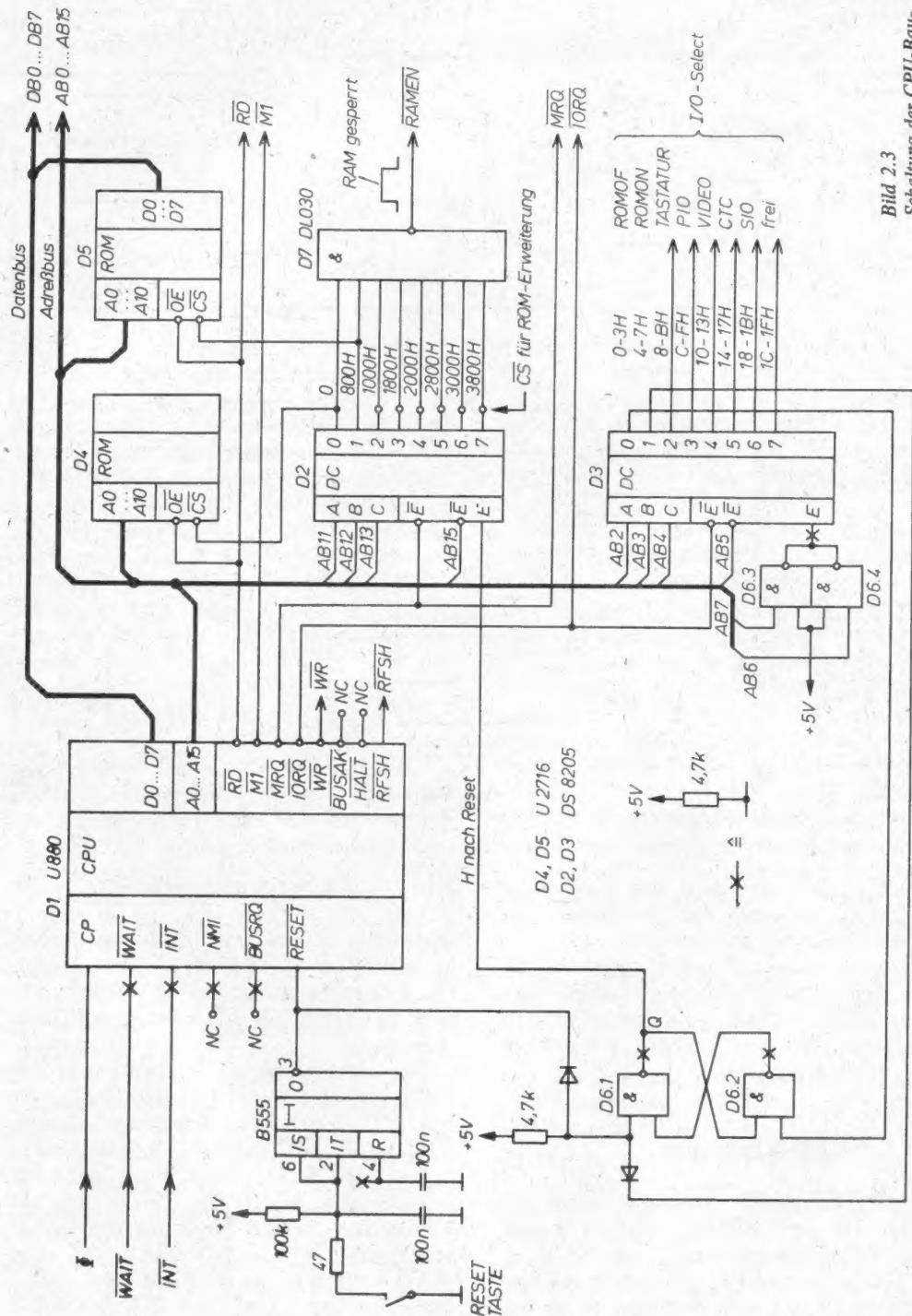


Bild 2.3  
Schaltung der CPU-Bau-  
gruppe

derständen auf ein sicheres H-Potential gebracht. Die Daten- und Adreßausgänge der CPU *U 880* sind als Tri-State-Ausgänge ausgeführt. Sie können ebenso wie die L-aktiven Steuerausgänge, von denen nur die Signale RD, WR, MREQ, IORQ über Tri-State-Ausgänge verfügen, eine TTL-Lasteinheit treiben. Auf Grund eines möglichst kompakten Aufbaus des Computers, wurde auf eine Pufferung der Busleitungen verzichtet. Es versteht sich von selbst, daß die statische und kapazitive Busbelastung in den vorgeschriebenen Grenzen bleibt. Bei Bedarf können natürlich Bustreiber, vorzugsweise *DS 8286* und/oder *DS 8282*, eingesetzt werden.

### 2.2.1.1. Taktversorgung

Die Taktfrequenz beträgt in der vorgestellten Variante 1,5 MHz, kann aber bei Beachtung der in Abschnitt 2.3. und Abschnitt 2.4. gegebenen Hinweise auch auf 3 MHz erhöht werden. Die Taktversorgung ist in Bild 2.3 nicht dargestellt. Die CPU erhält ihren Takt nämlich von der Video- bzw. RAM-Steuerung. Zentraler Takt in diesem Rechnersystem ist der Videotakt. Aus ihm werden alle anderen Takte synchron abgeleitet. Das ist nötig, da sowohl die CPU als auch die Bildschirmsteuerung auf denselben RAM zugreifen. Da der Zugriff der Bildschirmsteuerung auf den im RAM liegenden Bildwiederholspeicher absolute Priorität hat, muß sich die CPU dem Zeitregime der Video- und RAM-Steuerung unterordnen.

### 2.2.1.2. Startlogik

Der Mikroprozessor *U 880* kann grundsätzlich über RESET oder über Interrupt gestartet werden (siehe [1]). In der Regel wird der Prozessor mit einem RESET-Impuls gestartet. Die an den RESET-Eingang des Prozessors angeschlossene Kombination aus RC-Glied und dem Timerschaltkreis *B 555* bewirkt nach Zuschalten der Betriebsspannung das Einschalt-RESET (Power-On-Reset) und damit den Start des Systems. Die RESET-Schaltung wird häufig auch mit ein-

fachem oder sogar ohne Gatter ausgeführt. Empfehlenswert sind aber Schaltkreise mit Triggereigenschaften. Mit der Taste S1 kann das System zurückgesetzt werden. Eine zeitliche Begrenzung des RESET-Impulses ist nicht notwendig, da die Videosteuerung weiterhin die dynamischen Speicher auffrischt. Auf Grund der Gesamtwirkungsweise des Computers kann man auch auf die sonst beim Einsatz dynamischer Speicherbausteine notwendige Verknüpfung mit M1-Signal verzichten, ohne Datenverluste befürchten zu müssen (siehe [1]). Allerdings sollte eine prellarme Taste eingesetzt werden.

Nach dem Start durch einen RESET-Impuls beginnt die CPU die Befehlsabarbeitung bei der Speicherzelle 0000, wo selbstverständlich ein Programm (im EPROM) gespeichert sein muß. Aus diesem Grund legt man den Beginn des Betriebssystems des Mikrocomputers häufig auf die Speicherzelle 0000. Diese Methode ist mit einigen Nachteilen verbunden, die den Ausschlag für die Wahl einer anderen Lösung gaben. Die genannten Nachteile sind vor allem softwarebedingt und ergeben sich daraus, daß im oben genannten Fall (d. h. ROM ab Adresse 0000) ab Adresse 0000 kein frei verfügbarer Speicherplatz vorhanden ist. Dies soll im folgenden kurz erläutert werden.

Für Mikrocomputer auf der Basis des Prozessors *U 880* existiert heute ein breites Softwareangebot, das von einfachen Monitorprogrammen über Assembler bis hin zu Interpretern und Compilern höherer Programmiersprachen (z. B. BASIC, PASCAL usw.) und Betriebssystemen (SCP, BCU, CP/M usw.) reicht. In vielen Fällen läuft die Standardsoftware ab Adresse 0000. Ein Umschreiben bewährter Programme für andere Speicherbereiche bedeutet großen Aufwand und entsprechend viele Fehlermöglichkeiten. Außerdem sind einige der genannten Programme nur im RAM-Speicher lauffähig.

Nicht zu vergessen ist, daß sich im Anfangsbereich des Adreßraums des *U 880* einige Adressen mit besonderer Bedeutung (00H, 03H, 10H, 18H...38H, 66H: Restart, Interruptmode 1 und NMI) befinden, die möglichst allgemein verfügbar sein sollen.

Aus den genannten Gründen wurde die

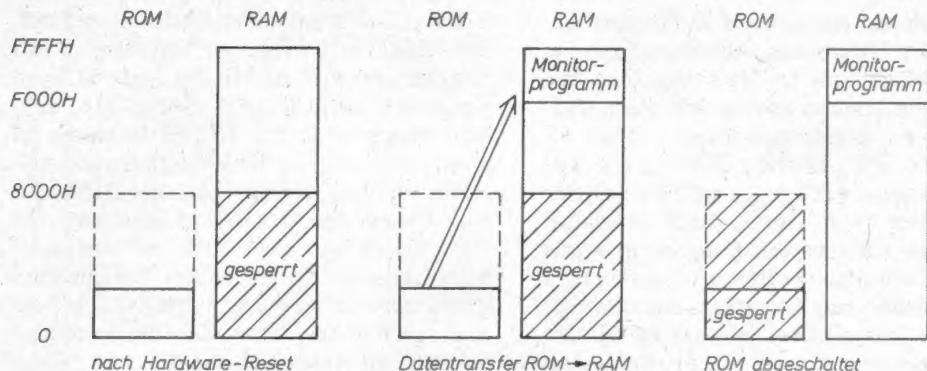


Bild 2.4  
RAM und EPROM, Speicheraufteilung

Voraussetzung dafür geschaffen, daß – beginnend bei der Adresse 0000 – ein großer zusammenhängender RAM-Bereich zur Verfügung steht. Die beiden widersprüchlichen Forderungen, nämlich RAM und ROM ab Speicheradresse 0000, werden von der in Bild 2.3 gezeigten Startlogik erfüllt, indem je nach Betriebszustand entweder die unteren 32 Kbyte des RAM-Speichers oder die EPROMs aus dem 64-Kbyte-Speicherbereich ausgeblendet werden. Bild 2.4 zeigt den prinzipiellen Ablauf und die Speicheraufteilung nach RESET. Nach dem System-RESET wird das aus den beiden Gattern D6.1 und D6.2 zusammengesetzte Flip-Flop gesetzt. Der Ausgang Q (D6.1) nimmt H-Pegel an, wodurch der Chipselektdecoder D2 (8205) freigegeben wird. Bei einem Speicherzugriff auf eine Adresse kleiner als 8000H wird ein Decoderausgang aktiv und kann ein EPROM auswählen. Gleichzeitig nimmt der Ausgang des 8fach-NAND-Gatters D7 H-Pegel an (Signal RAMEN). Dieser Ausgang ist mit der noch zu beschreibenden RAM-Ansteuerung verbunden und verhindert in diesem Fall den RAM-Zugriff. Im EPROM ist ein Ladeprogramm gespeichert, das die Betriebssoftware, die z. B. auf Kassette oder, wie in der vorliegenden Variante, ebenfalls im EPROM gespeichert ist, in den RAM-Arbeitsspeicher kopiert. Nachdem der Kopiervorgang beendet ist, wird das Betriebssprogramm (hier das Monitorprogramm) angesprungen. Vom Monitorprogramm aus wird nun durch einen OUT-Befehl auf das

ROMOF-Port das Flip-Flop (D6.1, D6.2) zurückgesetzt. Wie das praktisch aussieht, kann der in Abschnitt 2.8. beschriebenen Betriebssoftware entnommen werden. Der Meßpunkt Q nimmt L-Pegel an. Damit ist der Decoder D2 gesperrt, die EPROMs können nicht mehr selektiert werden. Das Signal RAMEN (Ausgang D7) bleibt jetzt auch bei Speicherzugriffen im Adressbereich 0000H bis 8000H aktiv. Damit steht auch dieser Speicherbereich als RAM zur Verfügung (Bild 2.4).

### 2.2.1.3. Speicher

Bild 2.3 zeigt, dass die CPU-Baugruppe mit 4 Kbyte Programmspeicher ausgerüstet ist. Diese 4 Kbyte teilen sich auf in 2 EPROMs U2716. Der Decoder D2 (8205) übernimmt die Chipselektierung. D2 decodiert den Speicherbereich von 0000H bis 3FFFH in acht 2-Kbyte-Blöcken. Da die Decodierung unvollständig ist, werden auch Chipselektionssignale im Bereich von 4000H bis 7FFF erzeugt. Diese im allgemeinen nicht nachteilige Einschränkung muß man bei individuellen Erweiterungen berücksichtigen.

Die freien Decoderausgänge ermöglichen eine EPROM-Speichererweiterung bis auf 16 Kbyte. Bei der Nutzung dieses Bereichs sind verschiedene Varianten denkbar. Wer nur in BASIC programmieren möchte, kann den BASIC-Interpreter (Abschnitt 2.9.) mit den zugehörigen Ein- und Ausgaberoutinen (Abschnitt 2.8.) kombinieren und in 8 EPROMs «brennen». Auf diese Weise läßt sich ein reiner BASIC-Rechner aufbauen.

Die Schaltung gestattet aber auch das Abspeichern häufig benötigter Programme auf EPROM. Dadurch, daß sich das Flip-Flop (D6.1, D6.2) softwaregesteuert setzen läßt, ist auch ohne Hardware-RESET das Einblenden des EPROM-Bereichs problemlos möglich. Ein OUT-Befehl auf die Adresse 04H (ROMON) bewirkt die Umschaltung der Speicher. Die Speicherkonfiguration entspricht dann der, die sich nach einem Hardware-RESET einstellt (Bild 2.4).

Tabelle 2.1. I/O-Portselektionierung

| Ausgang<br>DS 8205 | Decodierte I/O-Ports |          |  |  |
|--------------------|----------------------|----------|--|--|
|                    | Adressen             | Funktion |  |  |
| 0                  | 00, 01, 02, 03       | ROMOF    |  |  |
| 1                  | 04, 05, 06, 07       | ROMON    |  |  |
| 2                  | 08, 09, 0A, 0B       | Tastatur |  |  |
| 3                  | 0C, 0D, 0E, 0F       | PIO      |  |  |
| 4                  | 10, 11, 12, 13       | VIDEO    |  |  |
| 5                  | 14, 15, 16, 17       | CTC      |  |  |
| 6                  | 18, 19, 1A, 1B       | SIO      |  |  |
| 7                  | 1C, 1D, 1E, 1F       | frei     |  |  |

### 2.2.1.4. I/O-Portselektionierung

Der Datenverkehr der CPU mit peripheren Geräten wird, sofern diese nicht wie gewöhnliche Speicherbereiche ansprechbar sind, über I/O-Ports (Ein-/Ausgabe-Tore) abgewickelt. Die CPU U 880 liefert während eines I/O-Zyklus (IORQ = L) die niederwertigen Adreßbits AB0 bis AB7 als I/O-Portadresse. Ob es sich um einen Eingabe- oder Ausgabezyklus handelt, richtet sich nach der gleichzeitigen Aktivität des entsprechenden CPU-Signals RD oder WR (siehe [1]).

Im vorliegenden System übernimmt der Decoder D3 (8205) die zentrale I/O-Portselektionierung. Die Adreßleitungen AB0 und AB1 werden im allgemeinen für spezielle Selektionsfunktionen im Zusammenhang mit den I/O-Schaltkreisen PIO, SIO und

CTC benötigt und sind damit bereits belegt. Aus den in der Wertigkeit nächstfolgenden Adressen AB2 bis AB7 wird das I/O-Portselektionsignal mit dem Decoder D2 (8205), wie in Tabelle 2.1. angegeben, decodiert.

Damit verfügt man über 8 (x4) Portadressen. Im Bedarfsfall ist die Unterscheidung der an jedem Ausgang vorliegenden 4 Einzeladressen mit zusätzlichem Decoderaufwand (jeweils 1 aus 4) möglich.

Die verschiedenen Ein- und Ausgabebausteine sind mit den entsprechenden Decoderausgängen verbunden. Ein Decoderausgang ist noch frei, so daß man beispielsweise noch eine PIO anschließen kann. Für zusätzliche Ports muß der Decodierumfang erweitert werden (z. B. durch einen Decoder MH 74154).

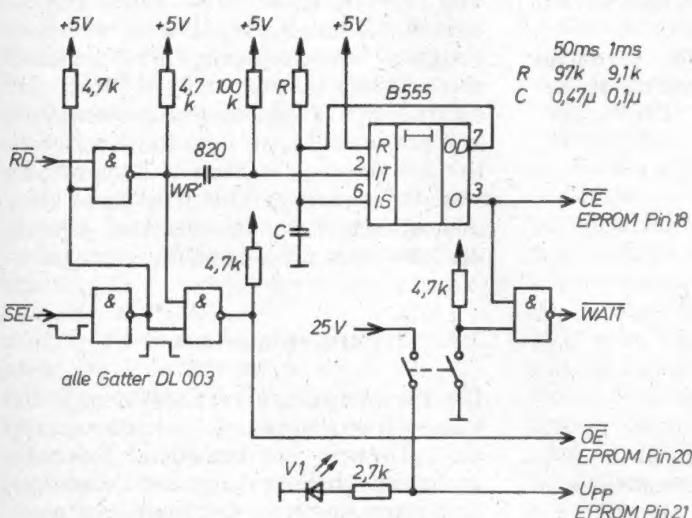


Bild 2.5  
EPROM-Programmierzusatz

### 2.2.1.5. EPROM-Programmierzusatz

Oft besteht der Wunsch, EPROMs zu programmieren. Eine sehr einfache Lösung zeigt Bild 2.5. Mit dieser Schaltung können EPROMs vom Typ 2716 programmiert werden. Die Anschlußbelegung dieser Typen ist in Bild 2.6 zu sehen. Die Funktionsweise der Schaltung basiert darauf, daß nach Anlegen der Adressen und Daten an den zu programmierenden Schaltkreis der Prozessor für die Dauer des Programmierimpulses in den WAIT-Zustand gebracht wird und damit die angelegten Adressen und Daten konstant bleiben. Auf diese Weise speist man ein entsprechend umfangreiches Ausgabeport ein. Der Programmierimpuls sowie der Waitimpuls werden mit einem Timer B 555 aus den CPU-Signalen erzeugt, d. h., ein Schreibbefehl löst automatisch einen Programmierimpuls aus. Die SEL-Leitung selektiert den EPROM und wird mit dem Speicherdecoder verbunden. Die Funktionsweise der Programmierzvorrichtung ist als Impulsschema in Bild 2.7 dargestellt.

Die Programmierspannung von 25 V läßt sich einfach mit einem Schalter zuschalten und durch eine Leuchtdiode signalisieren.

Es erscheint wichtig, darauf hinzuweisen, daß diese einfache Schaltungslösung die von den Herstellern vorgeschriebenen Programmervorschriften bezüglich der Vor- und Nachhaltezeiten der Daten und Adressen nicht ganz einhält (in Datenblättern werden typisch 2 µs gefordert!).

Der Programmierzvorgang ist unkompliziert. Nachdem durch entsprechende Be-

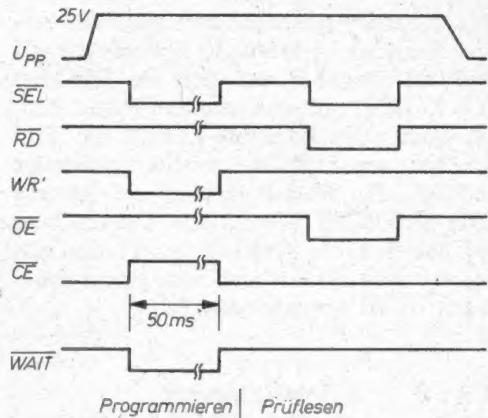


Bild 2.7  
Impulsdigramm der Programmierschaltung

fehle der ROM-Bereich zugänglich gemacht wurde (OUT ROMON), wird mit dem Schalter die Programmierspannung zugeschaltet. Das Programmieren erfolgt durch einfache Schreibbefehle [z. B. LD (HL), A], das Lesen durch einfache Lesebefehle [z. B. LD A, (HL)]. Ein EPROM kann demzufolge durch einen LDIR-Befehl beschrieben oder gelesen werden. Der Programmierzvorgang dauert knapp 2 Minuten. Ist der Speicher fertig programmiert, wird die 25-V-Programmierspannung wieder abgeschaltet.

Der schnelle Programmieralgorithmus nach Bild 2.8 arbeitet mit Impulsen von nur 1 ms Dauer. Nach jedem Schreibversuch wird die programmierte Zelle gelesen. Stimmt das programmierte Byte, wird nach einigen Sicherheitsprogrammierimpulsen zur nächsten Adresse fortgeschritten, andernfalls die Programmierung wiederholt. Wer es ganz eilig hat, kann vor dem Schreiben das zu programmierende Datenbyte testen: Ist es nämlich FFH, also gleich einer unprogrammierten Speicherzelle, braucht die Zelle nicht programmiert zu werden.

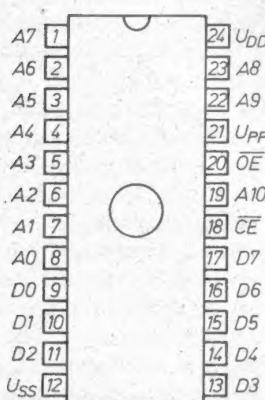


Bild 2.6  
Anschlußbelegung  
des U 2716

### 2.3. Fernsehhinterface

Die Kommunikation zwischen Mensch und Computer setzt geeignete Ein- und Ausgabemedien voraus. Von besonderer Bedeutung in diesem Zusammenhang sind Geräte bzw. Systemkomponenten, die den visuell unter-

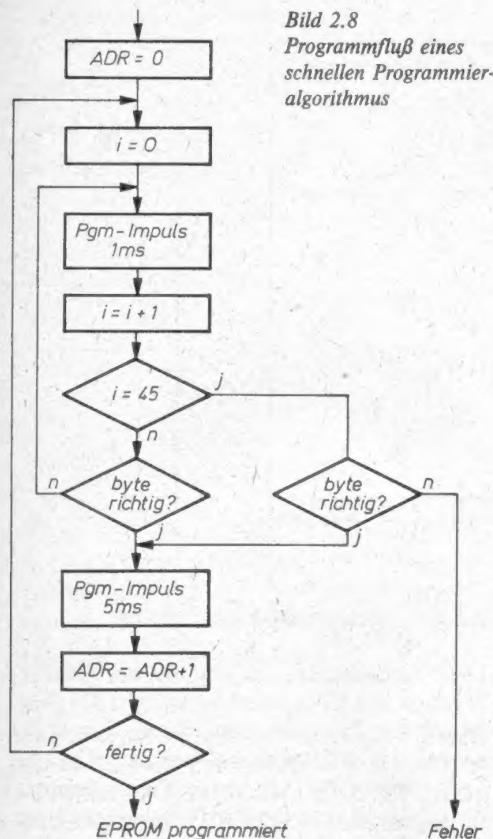


Bild 2.8  
Programmfluß eines schnellen Programmieralgorithmus

### 2.3.1. Bildschirmdarstellung

Das vom Fernsehen bekannte Darstellungsprinzip beruht auf der Helligkeitssteuerung des Elektronenstrahls in der Bildröhre. Der abhängig von der Videoinformation getaste Strahl überstreicht dabei in einem Zeilenraster den Bildschirm von oben nach unten. Nach der CCIR-Norm beträgt die Zeilenfrequenz 15625 Hz (d. h. Zeilendauer 64 µs) und 625 Zeilen je Bild. Die 625 Zeilen werden mit einem Zeilensprungverfahren und einer Bildfrequenz von 25 Hz dargestellt.

Das Zeilensprungverfahren ist zur Zeichendarstellung allerdings nicht besonders gut geeignet, da an horizontalen Kanten ein störendes Flimmern auftritt. Deshalb erzeugt das Fernsehinterface 2 identische Halbbilder mit etwa der Hälfte der 625 Fernsehzeilen und einer Bildfrequenz von etwa 50 Hz. Damit ist eine ausreichende Flimmerfreiheit des Bildes garantiert.

Bild 2.9 zeigt das verwendete Bildschirmformat. Das Interface erlaubt eine Darstellung von  $256 \times 256$  Bildpunkten. Jeder dieser 65 255 Bildpunkte ist einzeln setz- oder löscharbar. Die alphanumerischen Zeichen werden softwaregesteuert umgesetzt, die zugehörige Software wird in Abschnitt 2.8. beschrieben. Bei der Wahl der Bildschirmauflösung wurden folgende Punkte berücksichtigt:

- Der Aufbau des Rechners soll ohne speziell ausgesuchte Bauelemente möglich sein.
- Die Bildpunktfrquenz muß wenigstens mit ihrer Grundwelle noch vom HF- und Videoteil eines normalen handelsüblichen Fernsehempfängers verarbeitet werden können.

Bild 2.10 zeigt das BAS-Signal (Bild-, Austast- und Synchronsignal), das von der Bildschirmsteuerung erzeugt werden muß, um ein handelsübliches Fernsehgerät ansteuern zu können. Wie das Bild verdeutlicht, ist neben der eigentlichen Videoinformation, die die 256 Bildpunkte je Zeile abhängig von der jeweils darzustellenden Information hell- oder dunkeltastet, auch die Bereitstellung der Zeilen- und Bildsynchrosignalen sowie der Austastsignale notwendig. Durch

stützten Dialog ermöglichen. Aus der Vielzahl der Displayvarianten hat sich als universelles Anzeigemedium für Ziffern- und Zeichendarstellung bis hin zur Grafik der Bildschirm durchgesetzt. Das gilt in besonderem Maße für den Bereich der Mikrocomputer. Die Industrie bietet hierfür spezielle Bildschirmgeräte (vom Schwarzweiß- bzw. einfärbigen Monitor bis hin zum Farbmonitor) an.

Für viele Belange, vor allem natürlich aus der Sicht des Amateurs, erweist sich für den genannten Zweck der Einsatz eines handelsüblichen Fernsehgeräts als technisch und ökonomisch sinnvoll.

In [2] wurde ein für die Textdarstellung geeignetes Fernsehinterface vorgestellt. Inzwischen besteht vielfach der Wunsch, neben der einfachen Textdarstellung auch die grafische Ausgabe zu ermöglichen. Im folgenden wird ein einfaches Fernsehinterface beschrieben, das voll grafisch orientiert ist.

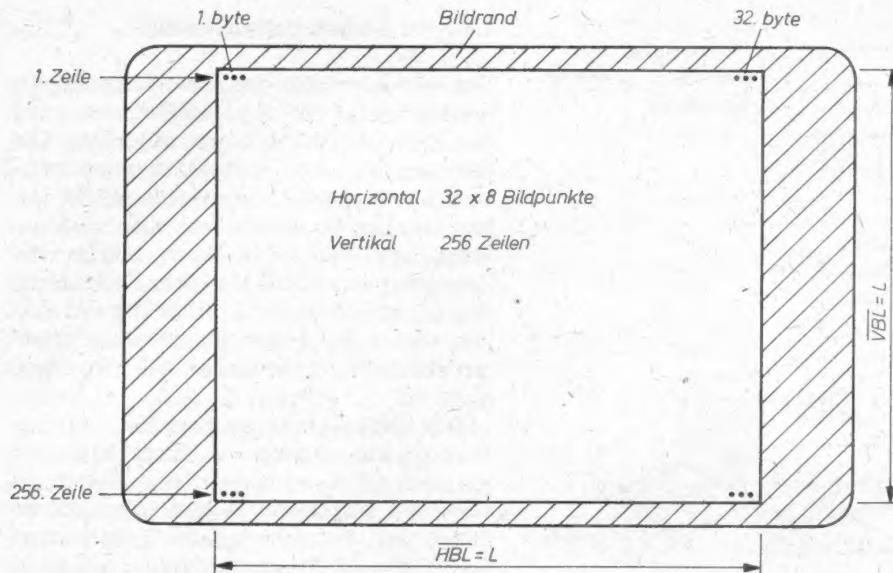
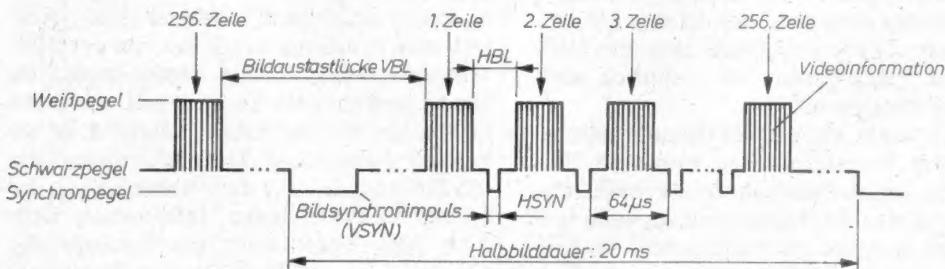


Bild 2.9  
Bildschirmaufbau

das Zeilenaustasten entsteht ein linker und ein rechter schwarzer Bildrand. In die Zeilenaustastphase fällt auch der Strahlrücklauf, der somit nicht sichtbar ist. Bei der Bildaustastung entsteht ein oberer und ein unterer schwarzer Bildrand. Durch die Festlegung von Bildrändern wird abgesichert, daß keine Bildinformationen von den Bildschirmrändern verschluckt werden. Die besonders in den Bildschirmmecken häufig auftretenden Unschärfen und Verzeichnungen haben keine Auswirkung.

Bild 2.10  
BAS-Signal, wie es vom Videointerface erzeugt wird



Darstellung vereinfacht, nicht maßstäblich

### 2.3.2. Schaltungsbeschreibung

Die Videoschaltung ist eng mit der Gesamt-funktion des Computers verknüpft. Sie übernimmt die Taktversorgung des gesamten Systems. Als Bildwiederholspeicher wird ein Teil des CPU-Arbeitsspeichers genutzt. Nachfolgend sind die zur Bilderzeugung verantwortlichen Baugruppen beschrieben.

#### 2.3.2.1. Takterzeugung

Der Taktgenerator muß eine Frequenz von 6 MHz liefern. Bild 2.11 zeigt die einfachste und günstigste Form der Takterzeugung. Der Quarzgenerator stellt den sehr genauen und stabilen Bildpunktztakt bereit. Oft ist es schwierig, einen solchen speziellen Quarz zu

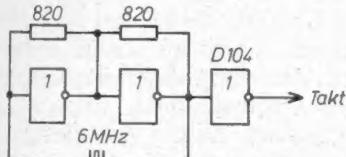


Bild 2.11  
Taktgenerator mit 6-MHz-Quarz

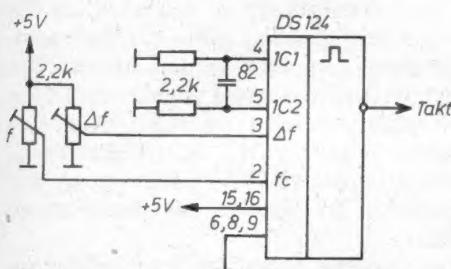


Bild 2.12  
Taktgenerator mit VCO-Schaltkreis

beschaffen. Dann bietet sich der Einsatz eines TTL-VCO-Schaltkreises an. Der VCO-Schaltkreis 74 LS 124/DS 124 beinhaltet 2 spannungsgesteuerte RC-Oszillatoren. Diese

VCO-Schaltkreise schwingen mit hoher Genauigkeit, wenn bestimmte Randbedingungen (günstige Spannungsversorgung, temperaturstabile RC-Bauelemente) eingehalten werden. Bild 2.12 zeigt eine Anwenderschaltung. Es wird nur ein Oszillatorsystem verwendet. Mit den beiden Einstellreglern lassen sich die Frequenz und der mögliche Frequenzhub einstellen. Beide Regler werden wechselseitig abgeglichen. Ein unerwünschter Effekt ist das oft auftretende «Schwimmen» des Bildes. Diese Erscheinung kann man verringern, wenn das Bild synchron zur 50-Hz-Netzfrequenz aufgebaut wird. Dazu eignet sich eine PLL-Schaltung. Bei Verwendung des genannten VCO-Schaltkreises ist die notwendige Schaltung nicht kompliziert. Wie Bild 2.13 zeigt, wird die Netzfrequenz (über einen Trigger) und das Vertikalaustastsignal einem Phasendetektor zugeführt. Ein aus 2 Transistoren aufgebauter Integrator erzeugt die Steuerspannung des VCO. Beim Abgleich ist wieder darauf zu achten, daß der maximale Frequenzhub einerseits möglichst klein, aber andererseits so groß ist, daß der Phasenregelkreis sicher einrastet und dieser Zustand gehalten wird. Gegebenenfalls ist der Kondensator C etwas zu ändern.

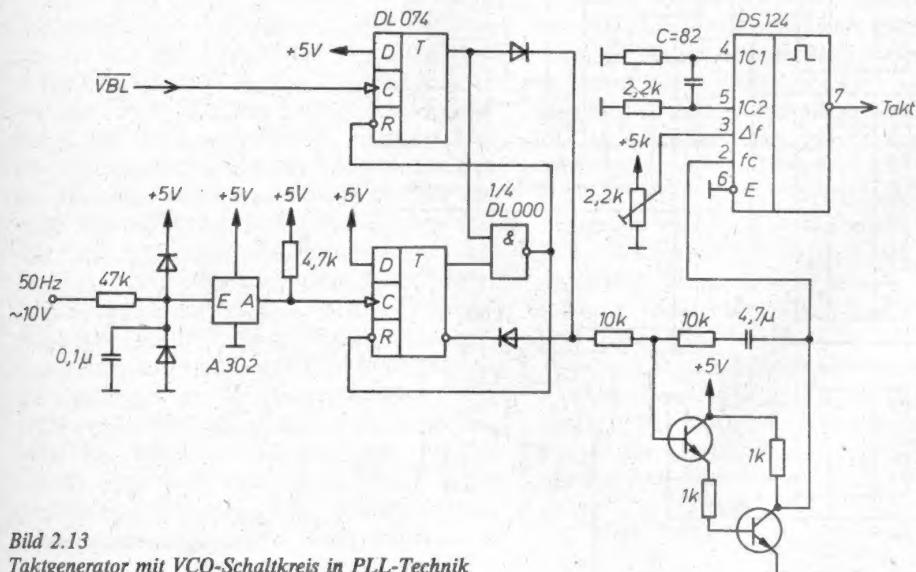


Bild 2.13  
Taktgenerator mit VCO-Schaltkreis in PLL-Technik

### 2.3.2.2. Horizontal- und Vertikalablenkung

Die Horizontal- und Vertikalablenkung hat die Aufgabe, die für die Bilderzeugung notwendigen Synchron- und Austastsignale sowie die Adressen für das Bildauslesen aus dem Bildwiederholspeicher bereitzustellen.

Bild 2.14 zeigt den Teil der Videosteuerung, der für die Horizontalablenkung verantwortlich ist. Dem Zähler D1 wird der 6-MHz-Bildpunkttaakt zugeführt. An seinen Ausgängen QA, QB, QC erscheinen die untersetzten Frequenzen (3,0, 1,5 und 0,75 MHz) und werden den Anschlußpunkten 1, 2 und 3 zugeführt, deren weitere Verwendung später beschrieben wird. Dieser erste Zähler zählt die 8 horizontalen Bildpunkte, die zu einem aus dem Bildwiederholspeicher ausgelesenen byte gehören, und gibt anschließend einen Ladeimpuls an das Schieberegister der Videosignalzeugung (siehe Abschnitt 2.3.2.3.). An QD des Zäh-

lers D1 und QA bis QD des Zählers D2 werden die Adressen VAD0 bis VAD4 abgenommen und dem Bildwiederholspeicher (hier Teil des Arbeitsspeichers, siehe Abschnitt 2.4.) zugeführt. An den Adreßleitungen VAD0 bis VAD4 treten die Adressen 0 bis 32 auf. Es können also 32 byte je Zeile adressiert werden. Da jedes byte aus 8 bits besteht, ist eine horizontale Auflösung von  $8 \times 32 = 256$  bits (Bildpunkte) gegeben. Das Flip-Flop D3.1 erweitert den Zählbereich des Horizontalzählers so, daß noch ein linker und rechter Rand sowie das Zeilensynchronsignal gewonnen werden können. Das Synchron- und das Austastsignal werden aus den Zählerzuständen decodiert. Am Ausgang des Flip-Flop D8.2 ist der Horizontal-synchronimpuls (HSYN) nachweisbar. Der Schaltkreis D7 erzeugt das Austastsignal (HBL).

Der zeitliche Ablauf der Horizontaladreß-erzeugung ist im Impulsdiagramm (Bild 2.15) dargestellt. Der Gesamtzählum-

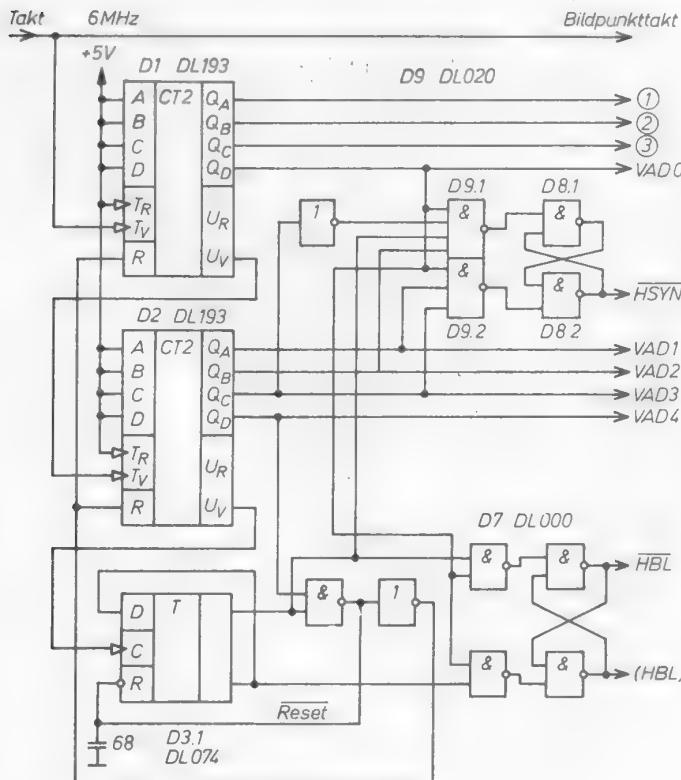


Bild 2.14  
Schaltung der Horizontalablenkung

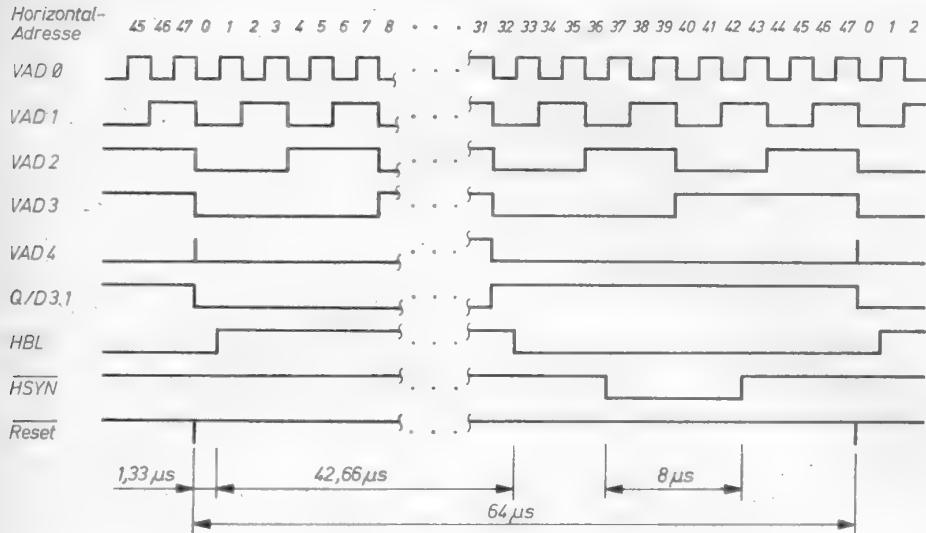


Bild 2.15  
Impulsdiagramm der Horizontalablenkung

fang beträgt 384. Beim 6-MHz-Takt hat jede Zeile eine exakte Länge von 64 µs.

Die Vertikalablenkung (Bild 2.16) hat die Aufgabe, die sichtbaren und unsichtbaren Zeilen des Bilds zu erzeugen. Der sichtbare Bereich sind genau 256 Zeilen. Diese werden von den beiden Binärzählern D4 und D5 gezählt. Von den Zählerausgängen gelangen die Adressen VAD5 bis VAD12 in den Bildwiederholspeicher. Das Flip-Flop D3.2 erzeugt das Vertikal- oder Bildauftastsignal. Nach 256 hellgetasteten Zeilen kippt das Flip-Flop und aktiviert das VBL-Signal. Der Gesamtzählumfang des Vertikalzählers beträgt 320. Bei einer Bildpunktfréquenz von 6 MHz liegt damit die Bildfréquenz mit 48,83 Hz etwas unter den geforderten 50 Hz. Die Synchronisation wird durch diese geringe Abweichung aber nicht beeinträchtigt. Das Bildsynchrosignal **VSYN** erzeugt man im Gegensatz zur Horizontalablenkung mit Hilfe zweier Monoflops. Durch diese Lösung kann die vertikale Bildlage etwas justiert werden. Das ist in bestimmten Fällen erforderlich, weil durch die hohe Anzahl der beschreibbaren Zeilen der obere und untere schwarze Bildrand recht schmal sind. Die

Lage des Bildes wird mit R1 justiert, während man mit R2 die beste Synchronisation einstellt.

Die beschriebenen beiden Schaltungsteile erzeugen die Videoadressen VAD0 bis VAD12. Das ist ein Adressierumfang von 8 Kbyte. Wie bereits erwähnt, ist der Bildwiederholspeicher Bestandteil des 64 Kbyte großen Arbeitsspeichers. Um eine vollständige 16-bit-Adresse bereitzustellen, fehlen noch die Adressen VAD13 bis VAD15. Im vorgestellten Computersystem bestimmt die CPU diesen Adreßteil. Dazu ist das Ausgabeport VIDEO (Adresse 10H) vorgesehen, das mit einem D-Latch **DL 075** (D6) aufgebaut ist. Die Schaltung erlaubt das Anlegen eines Bildwiederholspeichers auf Adressen, die ein Vielfaches der Bildwiederholspeichergröße sind (8 Kbyte). Bei Bedarf kann jeder 4-Kbyte-Block Beginn des Bildspeichers sein. Dann muß ein 4-bit-Addierer die Adresse VAD12 und VAD12a addieren und den dabei auftretenden Überlauf mit dem Adreßteil VAD13 bis VAD15 addieren. Die 4 Addiererausgänge sind dann die Adressen VAD12 bis VAD15, die gemeinsam mit VAD0 bis VAD11 dem Speicher zugeführt werden.

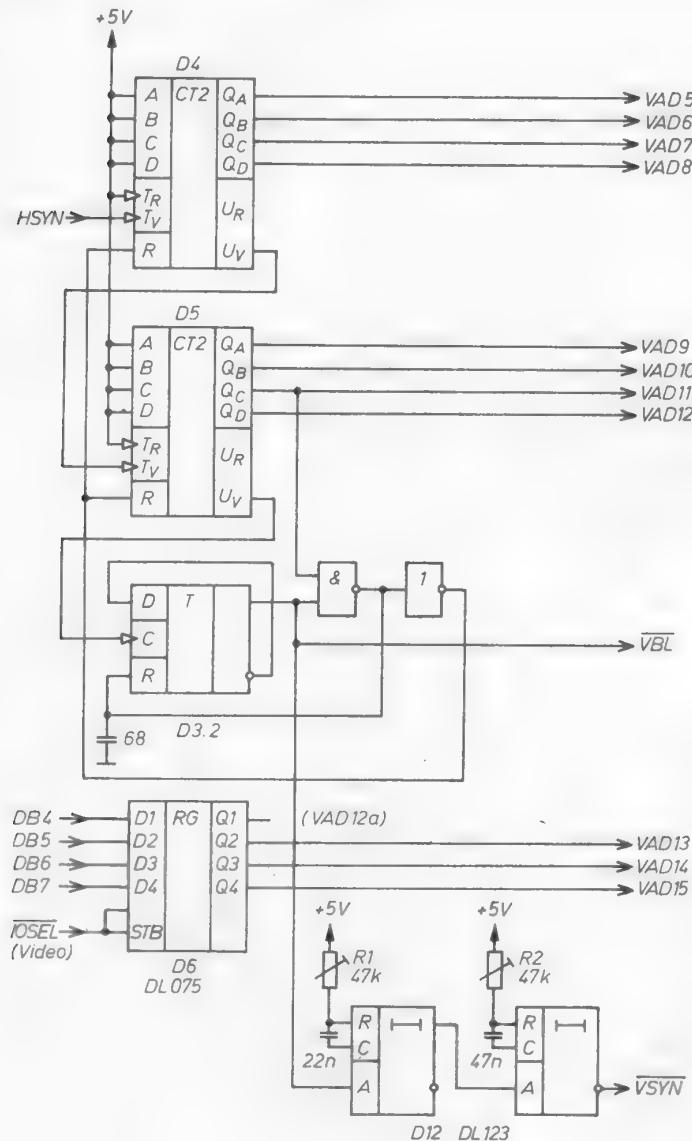


Bild 2.16  
Schaltung der Vertikalablenkung

### 2.3.2.3. Videosignal erzeugung

Die Baugruppe ist in Bild 2.17 dargestellt. Die Datenleitungen D0 und D7 sind mit den D0-Ausgängen der Speicherbausteine (RAM-Baugruppe, Abschnitt 2.4.) verbunden. Die fallende Flanke am Anschlußpunkt 3 (QC des Zählers D1, Bild 2.14.) lädt die Daten D0 bis D7 parallel in das Schieberegister D10. Mit jedem Taktimpuls (BP)

wird die Information als serielle Bildinformation aus dem Schieberegister herausgeschoben und dem BAS-Mischer zugeführt. Im BAS-Mischer werden außerdem die Synchron- und Austastsignale gemischt, anschließend mit einem Transistor entkoppelt und der Ausgangsbuchse mit einer Impedanz von  $75\ \Omega$  zugeführt.

Dieser Ausgang wird mit dem Fernsehgerät über den erforderlichen Videoeingang

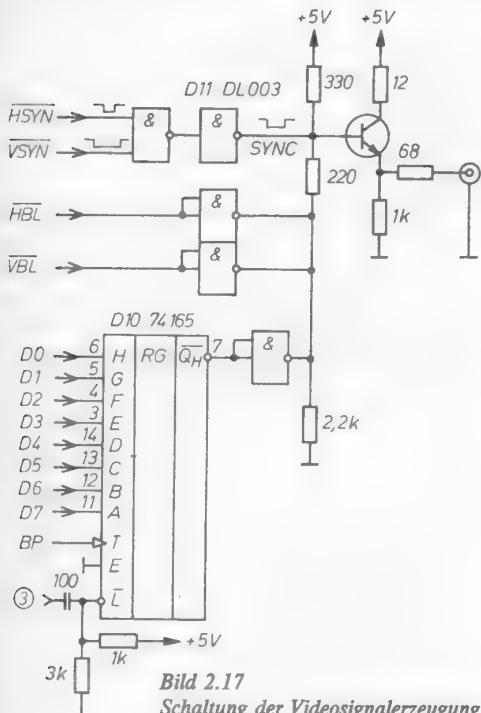


Bild 2.17  
Schaltung der Videosignal erzeugung

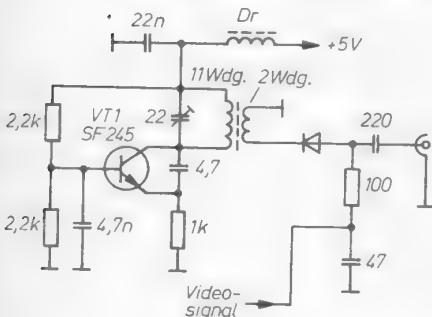


Bild 2.18  
HF-Modulator

verbunden. Zu beachten ist, daß das Fernsehgerät galvanisch vom Netz getrennt sein muß. Ältere Röhrengeräte sind also nicht geeignet.

Am BAS-Ausgang kann aber auch ein HF-Modulator (Bild 2.18) angeschlossen werden. Er ermöglicht das Einspeisen des Videosignals direkt in den Antenneneingang eines Fernsehgeräts. Beim Aufbau sind unbedingt die Störstrahlungsbestimmungen der Deutschen Post zu beachten!

### 2.3.3. Inbetriebnahme

Die Inbetriebnahme ist nicht schwierig. Bei der Taktgebervariante mit dem VCO-Schaltkreis sollte man die Frequenz mit einem Frequenzzähler überprüfen. Ist der 6-MHz-Takt nachweisbar, werden alle Adressausgänge sowie die Verbindungspunkte 1, 2 und 3 überprüft. Dabei müssen alle Zählerzustände im möglichen Zählumfang durchlaufen werden. Bei offenen Datenleitungen D0 bis D7 am Schieberegister D10 muß auf dem Bildschirm ein weißes Bildfenster zu sehen sein, das von einem schwarzen Rand umgeben ist. Gegebenenfalls läßt sich die Bildlage durch Verändern der Einstellregler der Vertikalablenkung optimieren.

Die richtige Funktion der Bildschirmsteuerung muß abgesichert sein, ehe die anderen Bestandteile des Computers aufgebaut werden. Die Zählerketten der Horizontal- und Vertikalablenkung haben vor allem die Aufgabe, den Rechner und die Ansteuerung der dynamischen RAMs mit Takten zu versorgen. Außerdem ersetzt die Videosteuerung die Refresh-Automatik des Prozessors U 880.

### 2.3.4. Erweiterungen

Selbstverständlich könnte das Bildschirminterface auch für eine Grafikauflösung von  $512 \times 256$  Pixel (o. ä.) ausgelegt werden. Die Änderungen sind minimal: Lediglich den Horizontalzähler müßte man dazu um eine Stufe erweitern und die Taktfrequenz auf 12 MHz erhöhen. Die hauptsächlichen Änderungen betreffen allerdings die RAM-Ansteuerung (siehe Abschnitt 2.4.).

Eine Erhöhung der Horizontalauflösung wird meist gewünscht, wenn die Anzahl der darstellbaren alphanumerischen Zeichen nicht mehr ausreicht. Es sind ja bei einer Grafikauflösung von  $256 \times 256$  Pixel bestensfalls 42 Zeichen auf 25 Zeilen darstellbar (theoretisch wäre auch die recht unübersichtliche Darstellung auf 32 Textzeilen denkbar). Ein Vorschlag für den experimentierfreudigen Amateur ist die gleichzeitige Verknüpfung einer alphanumerischen und einer grafischen Bildschirmsteuerung, also

eine Schaltung, die zusätzlich einen 2 oder 4 Kbyte großen Speicher und einen Zeichen-generator (EPROM) einschließlich Schieberegister enthält. Der zusätzliche Bildwiederholspeicher wird parallel zum Grafikspeicher ausgelesen. Für die CPU kann dieser RAM-Bereich im beschriebenen Computersystem im Adreßraum EOOOH bis EFFFH angeordnet werden.

Auch eine Umrüstung auf Farbbetrieb ist denkbar, wenn man für die Farben Rot, Grün und Blau einen eigenen Bildwiederholspeicher vorsieht. Jedem dieser 3 Bildspeicher ist ein eigenes Schieberegister zugeordnet, um die 3 Videosignale (Rot, Grün und Blau) zu erzeugen. Die 3 Bildwiederholspeicher müssen (von der Videoseite aus gesehen) parallel ausgelesen werden, von der CPU aber einzeln zugänglich sein.

## 2.4. Speicherbaugruppe

### 2.4.1. Dynamische RAM-Bausteine

Dynamische Speicherschaltkreise haben gegenüber statischen Speichern viele Vorteile: So werden in einem Schaltkreis weitaus mehr Speicherzellen vereinigt, die Leistungsaufnahme ist bedeutend geringer, und der Preis je bit ist kleiner. Es gibt dynamische Speicherschaltkreise von 4 Kbit bis 1 Mbit. Dabei bringen dynamische 4-Kbit-RAMs kaum noch Vorteile. Sehr «große» RAMs haben dagegen erst bei 16-bit-Rechnern eine Bedeutung. Für das vorgestellte Computerprojekt kommen vor allen Dingen 64-Kbit-RAMs vom Typ U2164 in Frage. Die Anschlußbelegung des U2164 (4164) zeigt Bild 2.19.

Durch geringfügige Schaltungserweiterungen können auch 16-Kbit-RAMs eingesetzt werden.

Im Gegensatz zu statischen RAMs (U202), in denen bekanntlich Flip-Flops die Informationen speichern, enthalten die modernen dynamischen RAMs zur Informationsspeicherung Eintransistor-Speicherzellen, in denen Kapazitäten die Informationen tragen. Da die Kapazitäten nur sehr klein sind, müssen die Ladungen mit Leseverstärkern wieder auf die erforderlichen Pegel ge-

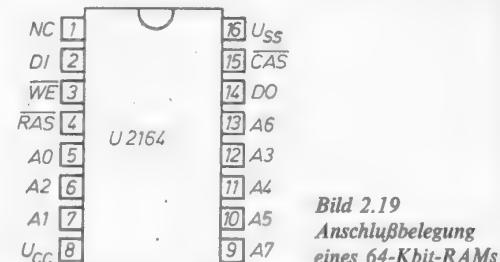


Bild 2.19  
Anschlußbelegung  
eines 64-Kbit-RAMs

bracht werden. Weiterhin enthält ein dynamisches RAM zusätzlich zur Funktion wichtige Schaltungskomplexe wie Decoder und Taktsteuerungen.

Die einzelnen Speicherzellen sind in einer Matrix aus Zeilen (row) und Spalten (column) angeordnet. Die Adresse wird dem RAM zu je 8 bit (7 bit bei 16-Kbit-RAMs) multiplex als Zeilen- und Spaltenadresse zugeführt. Die beiden Adreßteile werden in 2 Latches übernommen. Die ZeilenAdresse durch einen RAS-Impuls (row address strobe), die SpaltenAdresse durch einen CAS-Impuls (column address strobe). Wie bereits erwähnt, wird die Information in Kapazitäten gespeichert, die sich durch Leckströme langsam entladen. Um einen Informationsverlust zu vermeiden, muß jede Zeile mindestens einmal in 2 ms aufgefrischt werden (refresh). Das ist übrigens eine beachtlich lange Zeit, wenn man bedenkt, daß die Kapazität einer Speicherzelle nur etwa 0,04 pF beträgt. Beim Lesen und Auffrischen verstärken die Leseverstärker die Ladung der Kapazitäten einer gesamten Zeile und geben beim Lesen die mit der SpaltenAdresse ausgewählten Zelleninformationen an die Ausgangsstufe des RAMs weiter. Die Leseverstärker sorgen weiterhin für das Nachladen der Zellenkapazitäten und beschreiben sie während eines Schreibvorgangs (WE=L) neu.

Dynamische RAMs ermöglichen verschiedene Betriebsarten. Die im vorliegenden Computersystem verwendeten Formen sind in Bild 2.20 und Bild 2.21 dargestellt. Die dazugehörigen typischen Zeiten der 64-Kbit-Speichertypen enthält Tabelle 2.2.

Ein spezieller Refreshzyklus wird nicht verwendet, da das ständige Auslesen des Bildwiederholspeichers für ein ununterbrochenes Auffrischen der Speicherzellen sorgt.

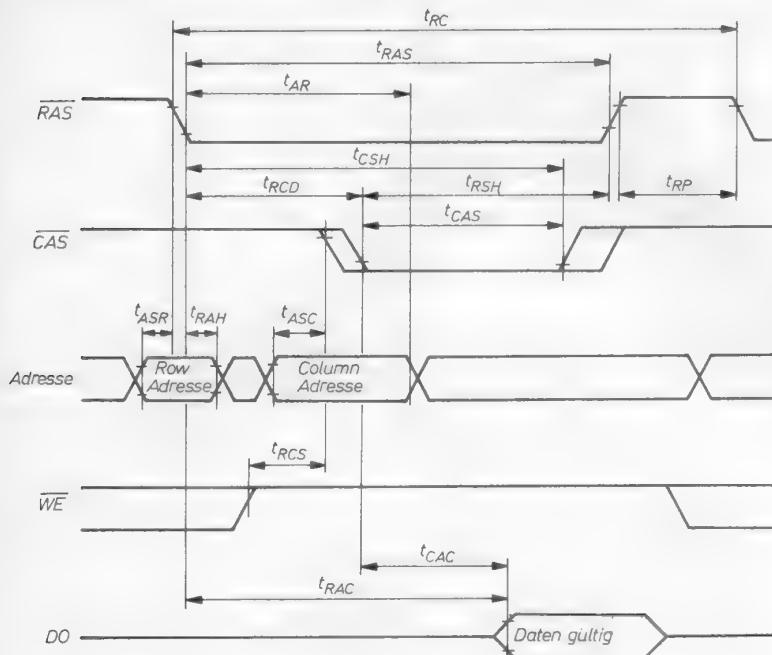


Bild 2.20  
Leszyklus eines  
DRAMs (Read)

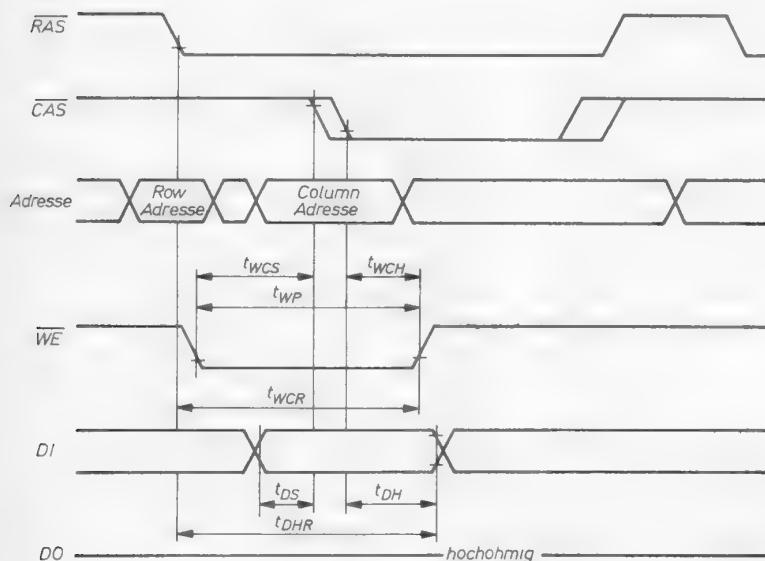


Bild 2.21  
Schreibzyklus eines  
DRAMs (Early Write)

| Symbol    | Parameter             | 4164-15 |       | 4164-20 |       | Einheit |
|-----------|-----------------------|---------|-------|---------|-------|---------|
|           |                       | min.    | max.  | min.    | max.  |         |
| $t_{REF}$ | Refresh-Zyklus-Zeit   |         |       | 2       | 2     | ms      |
| $t_{RP}$  | RAS High Impulsweite  | 100     |       | 120     |       | ns      |
| $t_{RAS}$ | RAS Impulsbreite      | 150     | 10000 | 200     | 10000 | ns      |
| $t_{CAS}$ | CAS Impulsbreite      | 75      |       | 100     |       | ns      |
| $t_{CSH}$ | CAS Haltezeit         | 150     |       | 200     |       | ns      |
| $t_{RSH}$ | RAS Haltezeit         | 75      |       | 100     |       | ns      |
| $t_{RCD}$ | Verzögerung RAS/CAS   | 25      | 75    | 30      | 100   | ns      |
| $t_{ASR}$ | Row Adresse vor RAS   | 0       |       | 0       |       | ns      |
| $t_{ASC}$ | Col. Adresse vor CAS  | -5      |       | -5      |       | ns      |
| $t_{RAH}$ | Row Adresse nach RAS  | 20      |       | 25      |       | ns      |
| $t_{CAH}$ | Col. Adresse nach CAS | 25      |       | 25      |       | ns      |
| $t_{AR}$  | Col. Adresse nach RAS | 95      |       | 120     |       | ns      |
| $t_{RC}$  | Leszyklus             | 260     |       | 330     |       | ns      |
| $t_{RAC}$ | RAS Zugriffszeit      |         | 150   |         | 200   | ns      |
| $t_{CAC}$ | CAS Zugriffszeit      |         | 75    |         | 100   | ns      |
| $t_{WCS}$ | WE vor CAS            | -10     |       | -10     |       | ns      |
| $t_{WCH}$ | WE nach CAS           | 45      |       | 55      |       | ns      |
| $t_{WCR}$ | WE nach RAS           | 95      |       | 120     |       | ns      |
| $t_{WP}$  | WE Impulsdauer        | 45      |       | 55      |       | ns      |
| $t_{DS}$  | Daten vor CAS         | 0       |       | 0       |       | ns      |
| $t_{DH}$  | Daten nach CAS        | 45      |       | 55      |       | ns      |

Tabelle 2.2.  
Zeitverhalten verschiedener  
dynamischer RAMs

#### 2.4.2. Schaltungsbeschreibung

Bild 2.22 zeigt die eigentliche RAM-Baugruppe. Sie besteht aus 8 Schaltkreisen U2164. Die Dateneingänge der RAMs sind mit dem Systemdatenbus gekoppelt. Die Datenausgänge sind zum einen mit dem Videoschieberregister (Bild 2.17), zum anderen mit einem Latch vom Typ DS 8282 verbunden. Die 4 Multiplexer leiten, je nach Betriebszustand, den niederwertigen oder den höherwertigen Adreßanteil den Speicherschaltkreisen zu. Ob den Speicherschaltkreisen die Adreßinformation der CPU (AB0 bis AB15) oder die Videoadresse (VAD0 bis VAD15) zugeleitet wird, entscheidet das Signal MUXB bzw. C/V.

Wie bereits mehrfach erwähnt, teilen sich die CPU und die Bildschirmsteuerung den gemeinsamen 64-Kbyte-Speicher des Computers. Das Impulsschema nach Bild 2.23 verdeutlicht die Speichersteuerung. Während der Darstellung eines bytes auf dem Bildschirm muß sowohl das nächste anzuzeilende byte aus dem Bildwiederholspeicher gelesen werden können als auch ein eventueller CPU-Zugriff auf denselben Speicher möglich sein. Zu diesem Zweck teilt man die Zeit, die die Videorealisierung eines by-

tes dauert (etwa 1,3 µs) in 2 gleiche Abschnitte. In der einen Phase wird ein Zugriff auf den Bildwiederholspeicher durchgeführt, in der anderen Phase der CPU-Zugriff. Das Signal C/V (CPU/VIDEO), es handelt sich um den durch 8 geteilten Bildpunktakt, entscheidet über die Richtung, aus der die Adressen in den RAM-Block gelangen, und wird deshalb als MUXB den Multiplexern in Bild 2.22 zugeführt. Am Ende eines Videozyklusses übernimmt das Videoschieberregister das aus dem Speicher gelesene byte, während das gelesene byte am Ende eines CPU-Zyklus durch das Signal STB in das Latch DS 8282 zwischengespeichert wird. Um die Schaltung nicht zu komplizieren, werden beide Speicherzugriffe immer ausgeführt, d.h., ein CPU-Zyklus wird auch durchlaufen, wenn die CPU gar keine Daten aus dem Speicher lesen will. Der Grund dafür ist, daß den CPU-Zyklus nicht, wie normalerweise üblich, das MRQ-Signal auslöst, sondern das strenge Zeitraster der Videosteuerung. Um ein störungsfreies Bild zu erzeugen, hat der Bildschirmteil beim Speicherzugriff absoluten Vorrang.

Es wird also in jeder Phase (CPU oder Video) ein vollständiger Speicherzyklus abgearbeitet. Die Signale RAS, MUXA und CAS

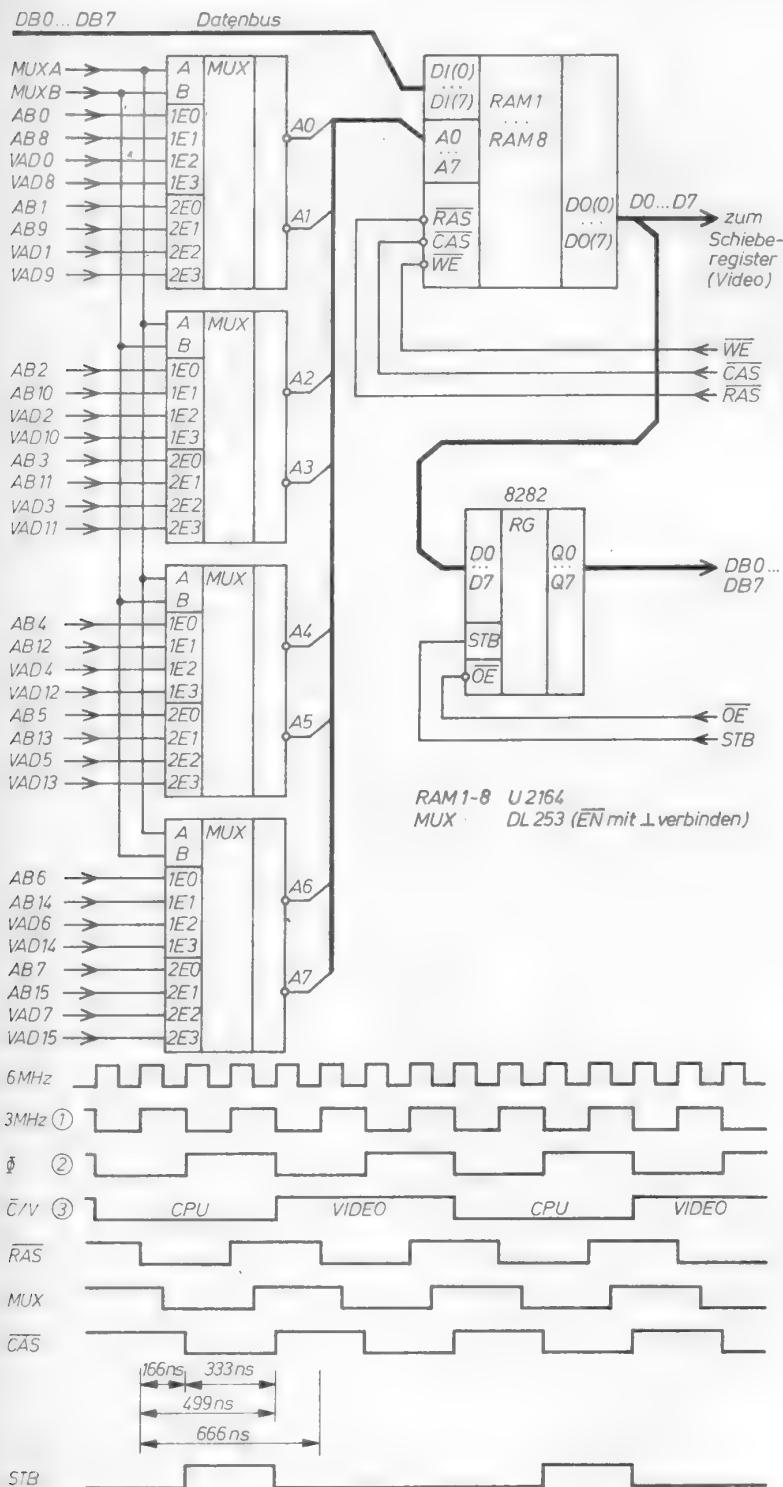


Bild 2.22  
Schaltung der RAM-Ansteuerung (Multiplexer, RAM)

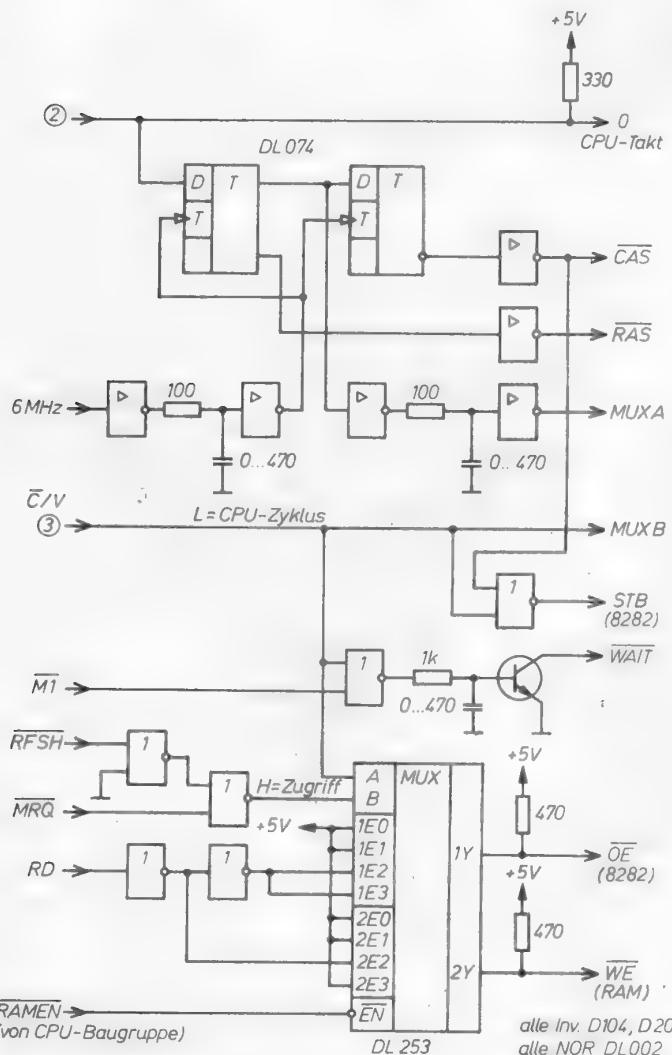
Bild 2.23  
Impulsschema der Speichersteuerung

erzeugt die Schaltung nach Bild 2.24. 2 D-Flip-Flops bilden ein Schieberegister, an dessen Ausgängen sich die schon gezeigten Impulsfolgen nachweisen lassen. Das MUXA-Signal wird durch Gatterverzögerungen gewonnen. Die gezeichneten RC-Glieder müssen gegebenenfalls, vor allen Dingen in zeitkritischen Fällen (langsame Speicher u. ä.), optimiert werden.

Ein Multiplexer *DL 253* und einige Gatter steuern den Zugriff auf den Speicher aus der CPU-Richtung. Die folgenden Fälle werden berücksichtigt:

- a- Die CPU will lesen. Unabhängig davon, welche Art von Speicherzyklus gerade läuft (C/V), liest die CPU die zuletzt im Latch gespeicherten Daten.
- b- Die CPU will schreiben. Das eigentliche WR-Signal der CPU ist ungeeignet. Der Schreibzyklus wird durch ein nicht aktives RD bei gleichzeitig aktivem MRQ-Signal erkannt. Geschrieben wird aber erst, wenn C/V einen CPU-Zyklus erlaubt.
- c- Refresh-Zugriffe der CPU werden ignoriert.

Wenn die CPU mit dem doppelten Takt des



alle Inv. D104, D204  
alle NOR DL002

**Bild 2.24**  
Schaltung der Impulserzeugung der Speichersteuerung

Signals C/V betrieben wird, ist die Synchronisation des CPU- und Videozugriffs auf ein und denselben Speicher relativ einfach. Das sollen Bild 2.25 bis Bild 2.28 demonstrieren. Der erste Fall zeigt die Abläufe, wenn die CPU zugreifen will, aber zur Zeit ein Video-

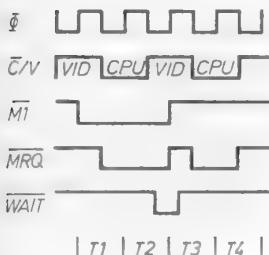


Bild 2.25  
Mi-Zyklus ohne Wait

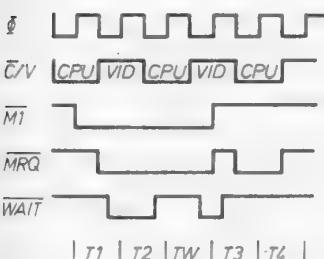


Bild 2.26  
Mi-Zyklus mit Wait

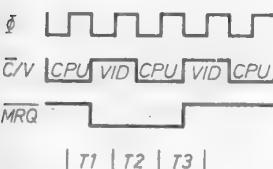


Bild 2.27  
Speicherzugriff erfordert keinen Waitzyklus

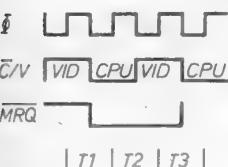


Bild 2.28  
Speicherzugriff erfordert auch in diesem Fall keinen Waitzyklus

zugriff läuft. Da der M1-Zyklus der CPU nur sehr kurz ist, wird ein Waittakt eingefügt. Es muß sichergestellt werden, daß wenigstens ein vollständiger CPU-Speicherzugriff mit dem aktiven MRQ-Signal zusammenfällt. Bild 2.26 zeigt ebenfalls einen M1-Zyklus. Hier ist das Einfügen eines Waittakts nicht erforderlich, weil die Zugriffe vorteilhaft zusammenfallen. Ebenso günstig sehen die Zugriffsverhältnisse in den Fällen aus, die Bild 2.27 und Bild 2.28 verdeutlichen.

#### 2.4.3. Schaltungserweiterungen

Bei der Wahl der Bildschirmauflösung und der CPU-Taktfrequenz wurden (neben den schon früher genannten Kriterien) leicht beschaffbare Bauelemente zugrunde gelegt. Natürlich läßt sich beim Einsatz einer CPU U 880 der CPU-Takt auf 3 MHz erhöhen. Unter Beibehaltung der Bildschirmauflösung würde sich jedoch die Synchronisation zwischen CPU- und Videospeicherzugriff wesentlich komplizierter gestalten. Wer allerdings über sehr schnelle 64-Kbyte-DRAMs (Zugriffszeit unter 150 ns, Zykluszeit unter 330 ns) verfügt, kann einen Computer mit  $512 \times 256$  Grafikpunkten und 3 MHz CPU-Taktfrequenz aufbauen. Neben Änderungen in der Horizontalablenkung der Bildschirmsteuerung ist vor allen Dingen die RAS/CAS-Erzeugung den neuen Bedingungen anzupassen.

Sinnvoll können auch mehrere RAM-Bänke sein. Beispielsweise kann der Speicherbereich ab COOOH mit zwei 16-Kbyte-Speicherbänken hinterlegt werden, die mit zusätzlichen Lese-/Schreibauswahllogiken und Schieberegistern ergänzt, eine Farbgrafik erlauben.

#### 2.4.4. Aufbauhinweise

Die Beachtung der folgenden Hinweise ist Voraussetzung für eine richtige Ansteuerung. RAM-Schaltkreise werden meist in großer Stückzahl eingesetzt. Die notwendigen Treiber und Multiplexer müssen alle Eingänge der Speicher-IS treiben können.

Statisch stellen RAMs keine nennenswerte Last dar. Die Eingangsströme liegen in der Größenordnung von etwa  $10\text{ }\mu\text{A}$ . Problematischer sind schon die kapazitiven Lasten, die von den Treibern angesteuert werden müssen. Die Eingangskapazitäten der Adreß- und Steuerleitungen betragen höchstens  $10\text{ pF}$ . In der Praxis liegt die durchschnittliche Eingangskapazität bei etwa  $3,5\text{ pF}$ . Rechnet man aber mit  $10\text{ pF}$ , so stellt eine 64-Kbyte-Speicherbaugruppe, bestehend aus 32 Stück 16-Kbit-RAMs, eine kapazitive Last von  $320\text{ pF}$  je Adreßpin dar, da ja alle Adreßleitungen zusammengeschaltet sind (gleiches gilt natürlich dementsprechend für die Steuerleitungen WR, RAS, CAS). Hinzu kommt noch die Verdrahtungskapazität der Leiterbahnen. Auch wenn die durchschnittlichen Eingangskapazitäten der RAMs kleiner als  $10\text{ pF}$  sind, so muß man trotzdem mit einer beachtlichen kapazitiven Belastung der Treiber rechnen. Diese Belastungen wirken sich negativ auf die Schaltzeiten aus, da die Treiberausgänge bei einem Pegelwechsel diese Kapazitäten umladen müssen. Besonders beim Einsatz von Low-Power-Schottky-Schaltkreisen kann es zu Problemen führen, denn sie weisen einen höheren Innenwiderstand als Standard-TTL-Schaltkreise auf. Zum Treiben von mehr als 16 RAM-Bausteinen sollte man deshalb Standard-TTL-Schaltkreise verwenden. Insbesondere gilt das für die Multiplexer.

Bei einem Pegelwechsel an den Treiberausgängen treten an den RAM-Eingängen positive und negative Spannungsspitzen auf. Dieses Überschwingen an den Impulsflanken wird hervorgerufen, weil die Treiberausgänge die Lastkapazitäten nur in endlicher Zeit umladen können. Die Spannungsspitzen verursachen eventuell Funktionsstörungen und in seltenen Fällen sogar Bauelementzerstörungen. Besonders kritisch sind Überschwinger auf den Steuerleitungen (RAS, CAS, WR), da sie zu Veränderungen des Speicherinhaltes führen können. Fehler dieser Art sind schwer zu lokalisieren. Wichtig ist es deshalb, schon beim Schaltungsentwurf Vorsorge zu tragen. Das Überschwingen wird durch entsprechende Reihenwiderstände zumindest stark gedämpft. Die Widerstände sollen einen Wert zwischen 22 und  $47\text{ }\Omega$  haben (Extremwerte sind 10 und  $100\text{ }\Omega$ ). In kritischen Fällen, bei Versuchsschaltungen und/oder Verdrahtungen in Wickeltechnik, muß man den Wert experimentell ermitteln. Zu kleine Werte bewirken eine möglicherweise zu geringe Dämpfung, zu große Werte hingegen verschleifen die Impulsflanken und verlangsamen damit die Speicherzyklen. Die Dämpfungswiderstände sollten möglichst nah an den Treiber- und Multiplexerausgängen angeordnet werden.

Ein weiteres Problem stellt die Spannungsversorgung der Treiber- und RAM-Schaltkreise dar. Leitungsinnenwiderstände und -induktivitäten der Versorgungsleitungen verursachen während der Speicherzyklen starke Einbrüche der Betriebsspannung. Deshalb müssen alle RAM-Bausteine, Multiplexer und Treiber ausreichend abgeblockt werden. Bild 2.29 zeigt die Verhältnisse auf den Versorgungsleitungen eines 64-Kbit-DRAMs. Zu erkennen sind Stromspitzen von bis zu  $100\text{ mA}$ . Das bedeutet, daß bei byteweiser Ansteuerung Stromspitzen von bis zu  $800\text{ mA}$  auftreten. Die Abblock kondensatoren müssen diese Spitzenströme liefern können. Am besten sind induktivitätsarme keramische Kondensatoren in der Größenordnung von 47 bis  $100\text{ nF}$  geeignet. Da die Stromspitzen nur wenige Nanosekunden dauern, müssen die Kondensatoren eine hohe Entladegeschwindigkeit aufweisen.

Ein weiteres Problem stellt die Spannungsversorgung der Treiber- und RAM-Schaltkreise dar. Leitungsinnenwiderstände und -induktivitäten der Versorgungsleitungen verursachen während der Speicherzyklen starke Einbrüche der Betriebsspannung. Deshalb müssen alle RAM-Bausteine, Multiplexer und Treiber ausreichend abgeblockt werden. Bild 2.29 zeigt die Verhältnisse auf den Versorgungsleitungen eines 64-Kbit-DRAMs. Zu erkennen sind Stromspitzen von bis zu  $100\text{ mA}$ . Das bedeutet, daß bei byteweiser Ansteuerung Stromspitzen von bis zu  $800\text{ mA}$  auftreten. Die Abblock kondensatoren müssen diese Spitzenströme liefern können. Am besten sind induktivitätsarme keramische Kondensatoren in der Größenordnung von 47 bis  $100\text{ nF}$  geeignet. Da die Stromspitzen nur wenige Nanosekunden dauern, müssen die Kondensatoren eine hohe Entladegeschwindigkeit aufweisen.

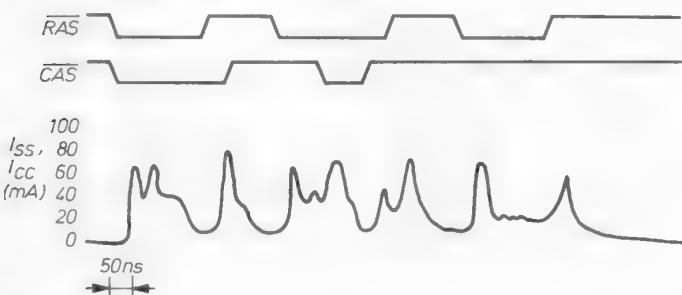


Bild 2.29  
Stromspitzen auf den Versorgungsspannungsleitungen

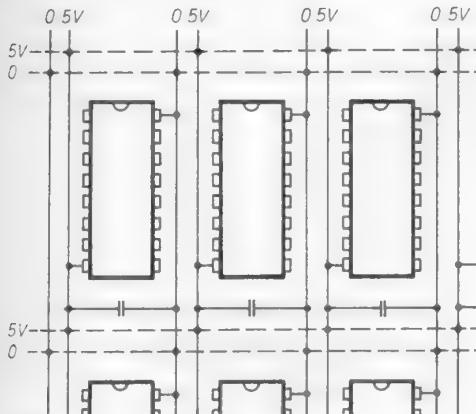


Bild 2.30  
Schema der Zuführung der Versorgungsspannungen zu den RAM-Bausteinen

den dauern, bewirken Tantal- oder Elektrolytkondensatoren keine wirksame Verringerung der Störungen auf den Versorgungsleitungen.

Weiterhin trägt ein gutes Layout der Leiterplatte nicht unwe sentlich zur Zuverlässigkeit der Speicherbaugruppe bei. Alle Versorgungsleitungen sollen kurz und dick sein. Bild 2.30 zeigt einen Layoutvorschlag für die Versorgungsspannungsführung zum 64-Kbit-RAM U2164.

Wenn die bisher beschriebenen Baugruppen aufgebaut sind, ist die Funktionsprüfung möglich. Obwohl der Rechner mit dynamischen Speicherbausteinen arbeitet, läßt sich die Funktionsweise am besten mit einer einfachen Einzelschrittlogik testen (siehe auch [2]). Ebenfalls in [2] wurde ein unabdingt erforderlicher RAM-Test veröffentlicht. Am besten, man «brennt» ihn, entsprechend angepaßt, in den EPROM ein und startet ihn nach RESET.

## 2.5. Alphanumerische Tastatur

Die alphanumerische Tastatur ermöglicht den Dialog mit dem beschriebenen Mikrocomputer. Sie kann in 2 Varianten aufgebaut werden. Die einfachere Tastatur hat bis zu 64 Tasten, die komfortablere ist zusätz-

lich noch mit einem getrennten Ziffern- und Cursorblock ausgestattet. Im Gegensatz zu anderen denkbaren und üblichen Lösungen wird der jeweilige Tastaturcode hardwaremäßig erzeugt.

### 2.5.1. Schaltungsbeschreibung

Die Funktionsweise und der Aufbau der Schaltung entsprechen in ihren Grundzügen der schon in [2] veröffentlichten alphanumerischen Tastatur. Inzwischen wurden jedoch einige Verbesserungen eingearbeitet.

Die Tastatur besteht aus 2 Schaltungskomplexen: der eigentlichen Tastaturschaltung und dem Interface zum Systembus. Sie ist so konzipiert, daß beim Drücken einer Taste der entsprechende Tastaturcode, bei Anforderung durch die CPU, auf den Datenbus zur Verfügung gestellt wird. Dabei muß natürlich der Tastaturcode identisch mit dem im Computer verwendeten Zeichencode gemäß ISO-7-bit bzw. ASCII sein. Alle Tastaturcodes sind auf einem EPROM gespeichert. Die Tastaturschaltung sorgt nun dafür, daß beim Drücken einer Taste dem EPROM gerade die Adresse bereitgestellt wird, auf der der dazugehörige Zeichencode abgespeichert ist. Dieser Code wird über ein Kabel an die Interfaceschaltung geführt. Für die Meldung an das System über das Vorliegen eines gültigen Zeichens steht das Konsolstussignal zur Verfügung.

#### 2.5.1.1. Tastaturschaltung

Bild 2.31 verdeutlicht die Funktionsweise. Ein Taktgenerator ( $f \approx 1,5 \text{ kHz}$ ), bestehend aus dem Schmitt-Trigger D1.1 und einem RC-Glied, steuert den 4-bit-Binärzähler D2 an. Dessen Ausgänge werden zum einen dem Adreßlatch D5 zugeführt. Die Zählerzustände bilden damit nach Übernahme in den Latch die 4 niedrigerwertigen Adreßbits A0 bis A3 für den Tastatur-EPROM. Zum anderen steuert der Zähler den 1-aus-16-Decoder D3 an. Aus den Zählerzuständen wird mit Hilfe des Decoders das Spaltenauswahlsignal für die Kontaktmatrix der Tastatur gewonnen. Die Decoderausgänge werden durch die

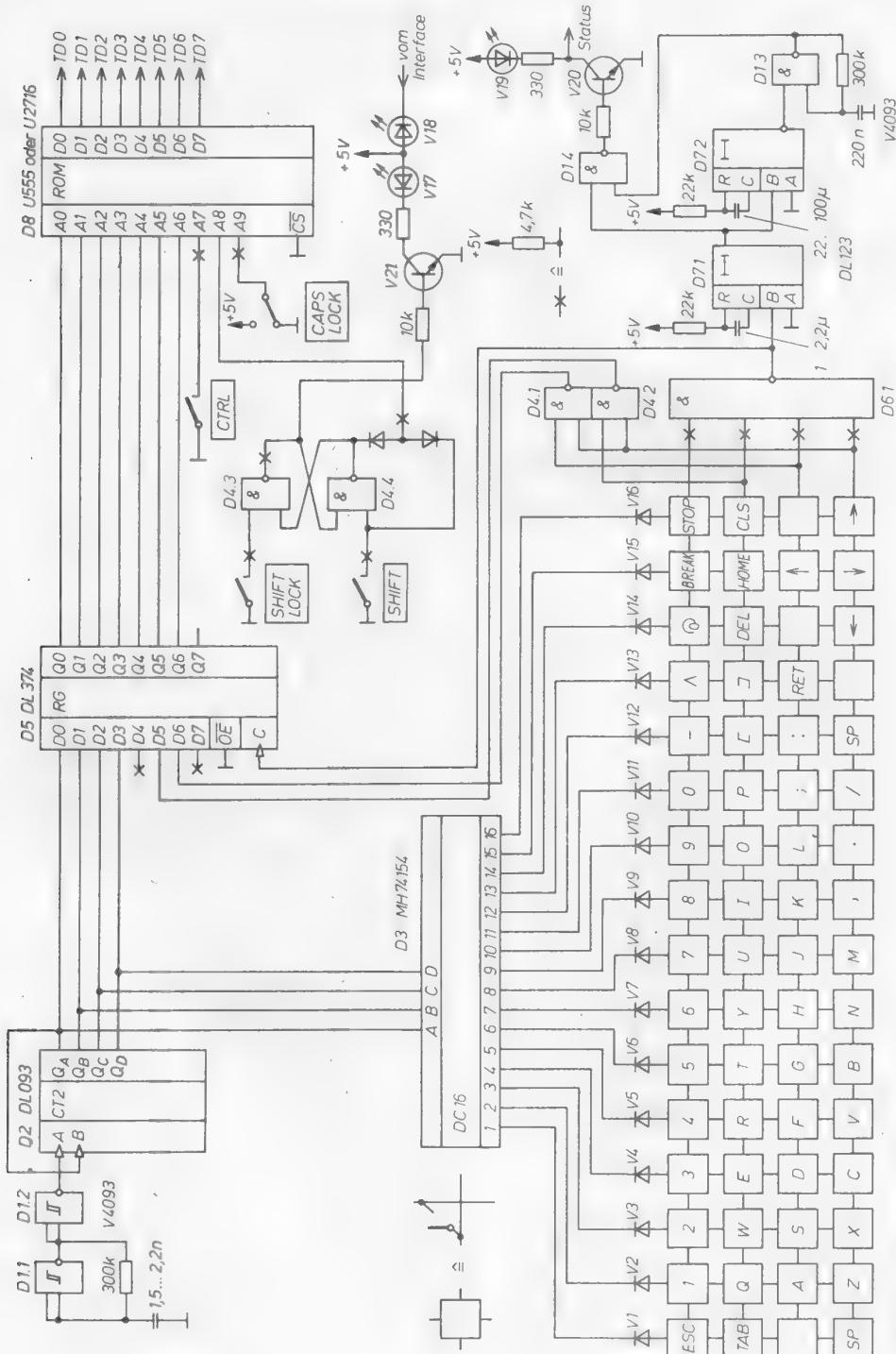


Bild 2.31 Schaltung der Tastatur

Dioden V1 bis V16 geschützt. Die Dioden verhindern das Zerstören von Ausgängen, wenn man mehrere Tasten gleichzeitig drückt.

Die Kontaktmatrix besteht aus 16 Spalten und 4 Zeilen. Daraus ergibt sich die maximale Tastenanzahl von 64. Die Matrix läßt sich leicht auf bis zu 128 Tasten erweitern. Dazu muß lediglich der Zählumfang des Spaltenzählers erhöht werden. Bild 2.32 zeigt die Erweiterung für zusätzliche 32 Tasten, was im allgemeinen ausreichen dürfte. Das Flip-Flop D9 erzeugt das neue Adressbit A4 und schaltet zwischen den Decodern D3 und D10 um, während der Decoder D10 die Spaltensignale der jetzt größeren Matrix zur Verfügung stellt. Die Schaltung wurde für den Einsatz von kontaktenschließenden Tastern (Mikroschalter oder Tasten mit Elastomerkontakten) ausgelegt. Wird eine Tastatur mit kontaktlosen Schaltern auf Basis von Hallelementen bevorzugt, dann muß man die Schaltungsänderungen nach Bild 2.33 und Bild 2.34 berücksichtigen. Sie machen sich erforderlich, weil nun die Spaltenauswahlsignale mit den Freigabeeingängen der Hallelemente verbunden werden müssen. Die Hallelemente B 461 G und B 462 G liefern an ihren Open-Collector-Ausgängen ein L-Signal, wenn auf sie ein Magnetfeld wirkt (also wenn die Taste gedrückt wird) und wenn der Freigabeeingang auf H-Potential liegt. Die Spaltensignale müssen demzufolge H-aktiv sein. Dazu gibt es 2 Möglichkeiten.

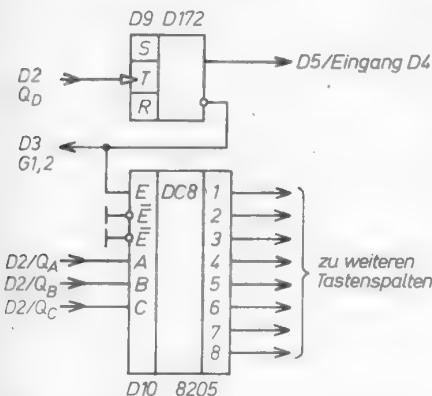
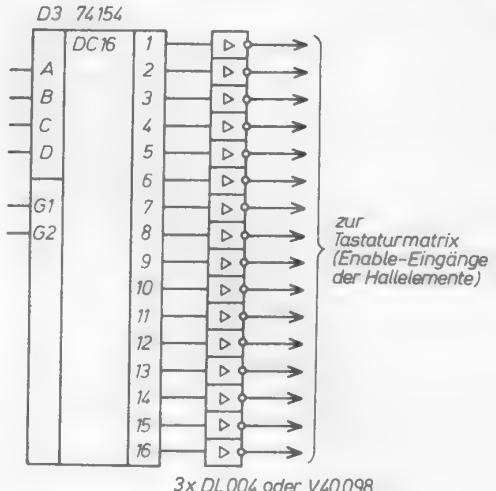
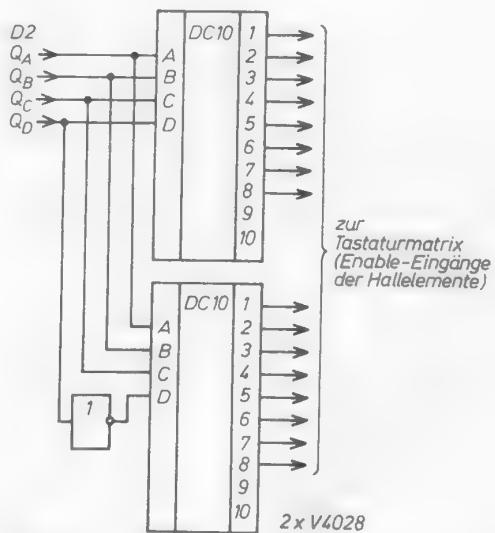


Bild 2.32  
Zusätzlicher Decoder für vergrößerte Tastaturnmatrix



3x DL 004 oder V40098

Bild 2.33  
Schaltungserweiterung für den Einsatz von Hall-Tasten



2x V4028

Bild 2.34  
Schaltungsänderung für den Einsatz von Hall-Tasten

Entweder man schaltet hinter die Decoder D3 und D10 (im Bedarfsfall) Inverter (Bild 2.33), oder die TTL-Decodert werden durch die CMOS-Schaltkreise V 4028 ersetzt (Bild 2.34). Halltasten mit den TESLA-IS MH 3 SD 2 lassen sich nicht ohne weiteres einsetzen. Zusammengefaßt ist einzuschätzen,

zen, daß Halltasten in dieser Schaltung elektrisch keine bedeutenden Vorteile bringen. Doch zurück zur Schaltung nach Bild 2.31. Für den Fall, daß keine Taste gedrückt ist, liegen alle Zeilenpotentiale über den Widerständen  $R_2$  bis  $R_5$  auf H-Pegel. Beim Schließen eines Tastenkontakts wird (wenn der Zähler diese Spalte anwählt) über den jeweiligen Decoderausgang das Zeilenpotential kurzzeitig auf L-Pegel gezogen. Am Ausgang des Gatters D6.1 entsteht ein kurzer H-Impuls, der das Monoflop D7.1 triggert. Es handelt sich bei D7 um ein retriggerbares Monoflop vom Typ *DL 123*. Dieses Monoflop bewirkt das Entprellen. Bild 2.35 veranschaulicht die Funktionsweise der Entprellschaltung. Wenn das Monoflop durch den ersten Zeilenimpuls in den aktiven Zustand kippt (Ausgang Q wird H), wird die Zeilen- und Spaltenadresse in den Latchschaltkreis D5 eingespeichert. Die Spaltenadresse ist der Wert des Zählers, die Zeilenadresse wird durch den Dezimal-Binärwandler, gebildet aus den Gattern D4.1 und D4.2, an Latch D5 gelegt. Solange das Monoflop Impulse innerhalb der Kippzeit erhält, bleibt sein Ausgang aktiv. Starke Kontaktprellungen machen sich durch den Ausfall eines oder mehrerer Impulse am Ausgang des Gatters D6.1 bemerkbar. Wie aus dem Impulsvorlauf in Bild 2.35 zu ersehen ist, wird die Kipp- und damit Entprellzeit durch die RC-Beschaltung an D7.1 eingestellt. Dabei dürfen sich einerseits fehlende Zeilenimpulse nicht am Ausgang Q bemerkbar machen, und andererseits muß die maximal mögliche Eingabegeschwindigkeit der Ta-

statur auf einem bestimmten Wert gehalten werden. Eine Kippzeit von etwa 30 ms stellt einen guten Kompromiß dar. Das Signal am Ausgang Q des Schaltkreises D7.1 eignet sich nun schon als Statussignal der Tastatur. Denn es führt H-Pegel, wenn ein Tastendruck erkannt wurde und ein gültiger Code abgenommen werden kann.

Die vorgestellte Tastatur hat noch eine automatische Repeat-Einrichtung. Durch sie läßt sich eine sehr angenehme Bedienung der Tastatur erreichen, wenn man beispielsweise eine lange Zeile unterstreichen will. Realisiert wird die Repeat-Funktion durch die zweite Hälfte des Schaltkreises *DL 123* (D7.2) und die Gatter D1.3 und D1.4. Das Statussignal am Ausgang Q von D7.1 gelangt an den Monoflop D7.2. D7.2 wird mit der steigenden Flanke des Statussignals getriggert. Die Kippzeit dieses Monoflops beträgt etwa 0,5 s. Wenn eine Taste, länger als bei normaler Betätigung üblich, gedrückt wird (einstellbar am RC-Glied an D7.2), kippt das Monoflop zurück und gibt den Repeat-Generator frei. Der Generator, gebildet aus dem Schmitt-Trigger D1.3 und einer RC-Beschaltung, schwingt mit etwa 15 Hz. Diese Schwingung unterbricht nun das Statussignal mit Hilfe des Gatters D1.4. Nach Invertieren durch den Transistor V20 steht das fertige Konsolstatussignal an. Im Adreßlatch D5 werden die Spalte und die Zeile der erkannten Taste zwischengespeichert. Die Ausgänge von D5 sind mit den Eingängen des EPROMs D8 verbunden. D8 übernimmt die notwendigen Umcodierungen unter Berücksichtigung des Zustands einiger wichti-

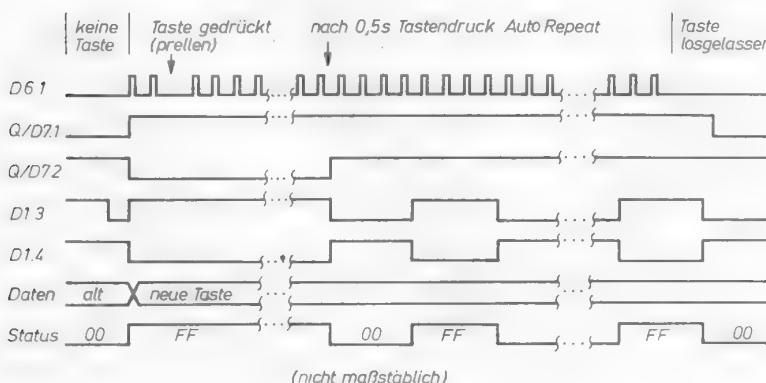


Bild 2.35  
Impulsdigramm der  
Tastatur

ger Sondertasten. Dazu gehören:

### CTRL

Durch gleichzeitiges Drücken dieser Controllaste wird ein Steuerzeichen erzeugt. Beispiel: C und CTRL erzeugt den Code 03H.

### SHIFT

Großbuchstabenschaltung.

### SHIFT LOCK

Dauer-Großbuchstabenschaltung; Zurückschaltung durch SHIFT.

### CAPS

Dauer-Großbuchstabenschaltung; die Ziffernreihe auf der Tastatur bleibt im Gegensatz zu SHIFT von der Umschaltung unberührt.

Die Zuordnung der EPROM-Adressebereiche zu den Spalten und Zeilen der Matrix ist dem Stromlaufplan nach Bild 2.31 sowie der Tabelle 2.3 zu entnehmen. Die Tabelle 2.4 zeigt die Codes aller Tasten bzw. Zeichen in den verschiedenen Ebenen (SHIFT, CTRL usw.) entsprechend dem 7-bit bzw. ASCII-Code. Mit diesen Hilfsmitteln dürfte die Programmierung eines Tastatur-EPROMs keine Schwierigkeiten bereiten (siehe auch [2]). Prinzipiell lassen sich die Tasten in der Matrix beliebig anordnen. Es wird jedoch gefordert, daß auf der jeweiligen EPROM-Adresse auch der richtige Zeichencode abgespeichert ist. Als EPROM eignen sich die Schaltkreise U555 oder U2716. Der U2716 wird zwar nur zur Hälfte genutzt, benötigt aber zum Betrieb nur eine Spannung (5 V). Einige Schaltungsdetails wurden bisher noch nicht erwähnt. Ein RS-Flip-Flop, bestehend aus den beiden Gattern D4.3 und D4.4, realisiert die bei Tastaturen übliche Großbuchstabenumschaltung. Der Zustand SHIFT LOCK wird mit der Leuchtdiode V17 angezeigt. 2 weitere Leuchtdioden erleichtern das Arbeiten mit der Tastatur. Die LED V19 zeigt

das Konsolstatussignal an. Das Signal re皮tiert, wenn die Diode V19 blinks. Eine besonders wichtige Kontrollanzeige stellt V18 dar. Diese Diode leuchtet immer dann, wenn die CPU die Tastatur abfragt. Anders ausgedrückt ist der Computer eingabebereit, wenn die LED V18 leuchtet (meistens ist es ein schlechtes Zeichen, wenn diese Diode längere Zeit dunkel bleibt). Über ein Kabel wird die Tastaturelektronik mit dem Interface verbunden. Die folgenden Signale müssen übertragen werden:

- DA0...7 (die 8 Datenleitungen des EPROMs),
- STATUS (das Konsolstatussignal),
- Steuerleitung für V18,
- Betriebsspannungen +5 V, +12 V, -5 V (die letzten beiden nur bei Verwendung des EPROMs U555).

#### 2.5.1.2. Interface

Die Interfaceschaltung verbindet die Tastatur mit dem Systembus des Computers. In [2] übernahm der Tastatur-EPROM zusätzlich diese Aufgabe. Um die Gebrauchswertigkeiten der Tastatur zu erhöhen, wurde eine neue Schaltungsvariante geschaffen (Bild 2.36). Sie ist denkbar einfach. 2 Multiplexer mit Tri-State-Verhalten sind der Kern der Schaltung. Bei einem Zugriff durch die CPU werden die Ausgangstreiber der Multiplexer aktiviert. Die Enable-Eingänge der Schaltkreise sind mit dem zentralen I/O-Decoder verbunden. Dieses I/O-Select-Signal wird zusätzlich einem Monoflop (DL 000) zugeführt. Dessen Ausgangsimpulse lassen die Diode V18 der Tastatur aufleuchten. Das Monoflop soll die Zugriffsimpulse verlängern, da sonst die Leuchtdiode nur dunkel glimmen würde. Die CPU kann, gesteuert über die Adresseleitung AB0 des Systembusses, sowohl den Tastencode als auch den Konsolstatus der Tastatur lesen. AB0 ist mit den Select-Eingängen der Multiplexer verbunden. Wie die Daten und das Konsolstatussignal an die Multiplexer gelegt werden, verdeutlicht Bild 2.36. Als Besonderheit fällt auf, daß das Konsolstatussignal auf alle B-Eingänge gelegt wurde. Diese Schaltungsvariante vereinfacht die softwarenmäßige

Tabelle 2.3.  
Zuordnung der Tastenzeilen zur ROM-Adresse

| Zeile der Kontaktmatrix | Adressebereich des EPROM (A0 bis A6) |
|-------------------------|--------------------------------------|
| 0                       | 00 bis 1F                            |
| 1                       | 20 bis 3F                            |
| 3                       | 40 bis 5F                            |
| 4                       | 60 bis 7F                            |

Abfrage der Tastatur.

Die Adressen des Interface sind:

Tastaturdaten 08H und

Tastaturstatus 09H.

Wird das Statusport angesprochen, so liest

man das byte 00 (keine Taste gedrückt) oder das byte FFH (Taste gedrückt).

Die verwendete vereinfachte Portselektion macht es notwendig, besonders darauf hinzuweisen, daß CPU-Ausgabebefehle (also

Tabelle 2.4. Codes aller Tasten bzw. Zeichen in den verschiedenen Ebenen

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

Die Codes 40, 7B bis 7E (5B bis 5D) sind für deutsche und ASCII-Belegung angegeben.

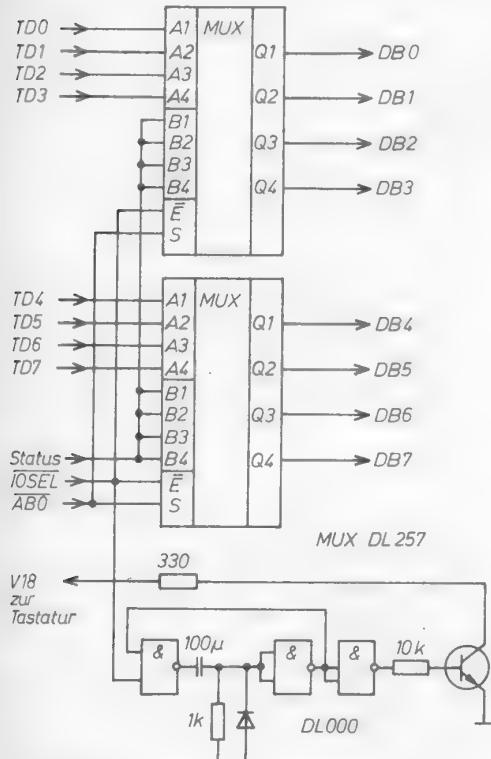


Bild 2.36  
Anschluß der Tastatur an den Systembus

OUT-Befehle), die die Tastaturadressen ansprechen, verboten sind. Sie können den CPU-Baustein zerstören. Das betrifft aufgrund der unvollständigen Decodierung auch die Adressen 0AH und 0BH. Wenn dieser Hinweis beachtet wird, macht sich die Schaltungsvereinfachung in keiner Weise nachteilig bemerkbar.

## 2.5.2. Aufbau und Inbetriebnahme

Das größte Problem für den Amateur dürfte der Aufbau der eigentlichen Tastatormatrix darstellen. Am besten ist natürlich eine industriell hergestellte Tastatur geeignet. Gelegentlich werden Tastaturen (z. B. von elektronischen Schreibmaschinen) im Amateurbedarfshandel angeboten. Sie haben vor allen Dingen den Vorteil, daß sie ergonomisch günstig aufgebaut und außerdem mechanisch stabil sind. 2 gewichtige Gründe,

die man auch beim Selbstbau beachten muß. Industrielle Tastaturen (speziell Schreibmaschinentastaturen) verfügen jedoch oft nicht über die vielen notwendigen Sondertasten und zwingen deshalb meist zum «Anstricken». Speziell betrifft das die Cursor-Steuertasten und die RETURN-Taste. Gerade die RETURN-Taste ist besonders wichtig, da sie alle Eingaben abschließt. Weil diese Taste zentrale Bedeutung hat, muß sie etwas größer und günstig angeordnet sein. Auch auf die nachfolgenden weiteren wichtigen Tasten sollte man keinesfalls verzichten.

**Cursortasten**, ohne sie kann der Computer kaum bedient werden.

**ESC**, Escape ermöglicht das Umschalten in vielen Programmen (Monitor, Basic usw.).

**CTRL**, Control gestattet die Eingabe von speziellen Steuerzeichen (z. B. in Basic zur Programmunterbrechung).

**DEL**, Delete löscht das zuletzt eingegebene Zeichen.

Diese Hinweise gelten insbesondere beim Einsatz von industriellen Tastaturen. Sie sollten aber auch beim völligen Selbstbau berücksichtigt werden. Steht keine Fertigtautatur zur Verfügung, so fällt der Auswahl geeigneter Kontaktlemente eine besondere Bedeutung zu. Als Taster eignen sich beispielsweise Mikrotaster mit Bedienkappe des VEB Elektroschaltgerätewerk Auerbach. Diese Tasten lassen sich gut aneinanderreihen. Bei einigen mit diesen Schaltern aufgebauten Tastaturen wurden die Plasteinsätze aus den Bedienkappen entfernt und lackierte Schilder in die entstehenden Vertiefungen geklebt. Einige so realisierte Tastaturen arbeiten schon seit einigen Jahren zuverlässig. Nachteilig an dieser Variante ist der zu geringe Hub der Tasten und das etwas laute Schreibgeräusch. Halltasten sind zwar recht hoch, so daß keine ausgesprochen flache Tastatur aufgebaut werden kann, haben aber sehr angenehme Drückeigenschaften. Hinweise zum Einsatz wurden bereits weiter oben gegeben. Eine flache Tastatur mit guten Schreibegenschaften läßt sich mit Kontaktbauelementen auf Elastomerbasis (TSE) realisieren. Unabhängig davon, welche Taster der Amateur verwendet, sollte auf eine bediengerechte Anordnung, wie sie Bild 2.37



Bild 2.37  
Beispiel einer Tastenanordnung

|       |      |       |
|-------|------|-------|
| Break | Stop | Erase |
| Home  | ↑    | CLS   |
| ←     | ↓    | →     |

Bild 2.38

Beispiel einer günstigen Anordnung der Cursor-tasten

zeigt, geachtet werden. Eine vorteilhafte Platzierung der Cursor-tasten ist in Bild 2.38 zu sehen. Voraussetzung für gute Schreibeigenschaften und deshalb wichtig ist der Versatz der einzelnen Tastenreihen und der Abstand zwischen den einzelnen Tasten. Besonders große Sorgfalt erfordert die mechanische Konstruktion und der Bau der Leertaste. Sie sollte möglichst groß sein. Damit die Leertaste sich nicht beim Drücken verkantet und/oder verklemmt, ist eine geeignete Führung notwendig.

Um die Montage der vielen Einzeltasten zu erleichtern, ist es günstig, Streifen aus dünnem Blech oder Hartpapier zwischen den einzelnen Tastenreihen einzubauen. Diese Streifen verbessern die horizontale Ausrichtung der Tastenreihen.

Die vielen Forderungen an eine gute Tastatur schließen den Einsatz von Elastomermatten, wie sie in Taschenrechnern verwendet werden, von vornherein aus. Ebenso, wie bei den gleichfalls bedenklichen Sensor-tasten, bereitet das Herstellen geeigneter und zuverlässiger Kontaktflächen (Veredelung) dem Amateur meist große Schwierigkeiten.

Der elektrische Aufbau ist weit unproblematischer. Zur Inbetriebnahme – vorerst ohne EPROM-Bestückung – wird zunächst die Taktfrequenz am Ausgang des Gatters D1.2 mit einem Oszilloskop kontrolliert. An-

schließend führt man eine oszillographische Signalkontrolle an folgenden Stellen der Schaltung durch:

- Zählerausgänge *DL 093* (D2) einschließlich Eingänge *DL 374* (D5) und *MH 74154* (D3).
- Decoderausgänge *MH 74154* (D3). Das Decoderausgangssignal muß bei gedrückter Taste auch auf der jeweiligen Zeile erscheinen.
- Ein- und Ausgänge der Monoflops. An diesen Punkten erscheint nur bei gedrückter Taste ein Signal.

Sind alle Signale ordnungsgemäß vorhanden, steckt man den Tastatur-EPROM auf und überprüft, ob nach dem Drücken einer Taste auch der richtige Code an den Datenausgängen des EPROMs anliegt. Gemäß ISO-7-bit bzw. ASCII (siehe Tabelle 2.4.) muß sich z. B. für den Buchstaben a der Code 61H ergeben, wenn nicht gleichzeitig die Taste CTRL oder SHIFT gedrückt ist (bzw. SHIFT LOCK und/oder 'CAPS nicht vorher eingeschaltet wurde). Als praktisches Hilfsmittel für diese Kontrolle erweisen sich 8 Leuchtdioden mit entsprechenden Treibertransistoren. Notfalls erfüllt auch ein einfacher Vielfachmesser den gewünschten Zweck. In dieser Art lassen sich alle Tastenfunktionen und -codes kontrollieren. Das Zusammenspiel zwischen den einzelnen Schaltungskomponenten Tastatur und Interface läßt sich dann nach Inbetriebnahme der Betriebsssoftware prüfen.

### 2.5.3. Tastatur mit serieller Schnittstelle

Amateure, denen die Anzahl der Verbindungsleitungen von der eigentlichen Tastatur zum Businterface zu groß ist, können

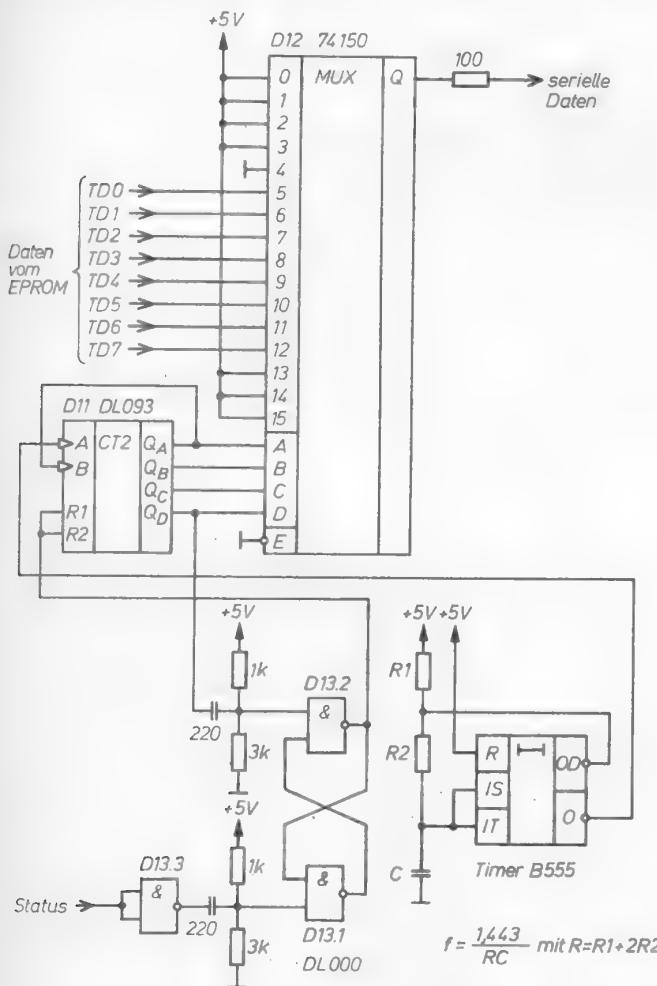


Bild 2.39  
Anschlußsteuerung für die serielle Verbindung zum Computer

die Tastatur auch mit einer seriellen Schnittstelle ausrüsten. Die dafür notwendige Schaltung verursacht nur geringen Aufwand (Bild 2.39). Ein Flip-Flop, aufgebaut aus 2 Gattern, steuert den Serialisierungsvorgang. Wenn das Statussignal aktiv wird, kippt das Flip-Flop und gibt den Zähler D11 frei. Die Zählerausgänge sind mit den Adreßeingängen des Multiplexers D12 verbunden. Es handelt sich um den 16-auf-1-Multiplexer MH 74150 (es können auch 2 Schaltkreise von Typ DL 251 verwendet werden). An den Eingängen des Multiplexers liegen die vom EPROM kommenden Daten sowie die Start- und Stopbits. Nachdem alle bits gesendet wurden, kippt das

Flip-Flop wieder in seine Ausgangslage zurück und sperrt den Zähler durch Aktivieren der Resetleitung. Die Serialisierung ist als Impulsdiagramm in Bild 2.40 dargestellt.

Der Timer B 555 erzeugt den Sendetakt. Als Empfänger wirkt auf der Computerseite ein Kanal der SIO. Wie das konkret aussehen kann, wird in Abschnitt 2.7.2. noch erläutert.

Zur Verbindung der Tastatur mit dem Rechner werden nun nur noch sehr wenige Leitungen benötigt:

- eine Datenleitung,
- Betriebsspannung,
- gemeinsame Masse,
- eventuell Steuerleitung für V18.

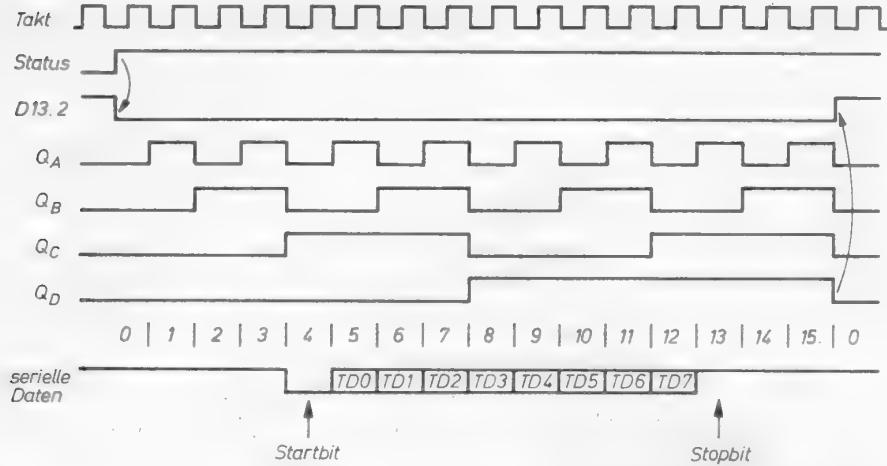


Bild 2.40  
Impulsdiagramm zur Anschlußsteuerung nach  
Bild 2.39

## 2.6. Kassetteninterface

Als externen Daten- und Programmspeicher setzt der Amateur vorwiegend Magnetbänder ein. Im folgenden wird ein einfaches Kassetteninterface beschrieben, welches die Aufzeichnung und die Wiedergabe von Daten und Programmen mit einem handelsüblichen Kassettenrecorder ermöglicht. Mit diesem Interface verfügt der Computer über einen zuverlässigen und billigen externen Massenspeicher.

### 2.6.1. Aufzeichnungsverfahren

Ein Magnetbandgerät hat, über Band gemessen, näherungsweise eine Bandpaßcharakteristik. Gleichspannungen, die bei der Aufzeichnung langer Folgen von L- oder H-bits auftreten würden, lassen sich nicht aufzeich-

nen. Digitale Daten können mit einem Audio-Kassettengerät nur dann aufgezeichnet werden, wenn sie mediengemäß codiert sind. Das Codierungsverfahren sollte dabei den Übertragungseigenschaften des Speichermediums, also dem Kassettenrecorder, möglichst gut angepaßt sein. Einige Möglichkeiten der Codierung werden nachfolgend kurz aufgezeigt.

Ein sehr einfaches Verfahren ist die Amplitudenmodulation. Die Datenbits tasten einen NF-Träger von meist etwa 2 kHz. Bild 2.41 zeigt das modulierte Signal. Die Information gewinnt man durch einfaches Gleichrichten und Sieben des wiedergegebenen Signals zurück. Eine praktische Realisierung wurde in [1] veröffentlicht. Dieses Verfahren hat 2 entscheidende Nachteile. Besonders ungünstig ist die geringe Datenübertragungsrate von (meist) nur 110 baud (110 bit/s). Die Aufzeichnung eines Blocks von 1 Kbyte Länge dauert etwa 1 Minute und 42 Sekunden. Es ist leicht einzusehen, daß die Aufzeichnung und Wiedergabe des weiter hinten beschriebenen BASIC-Interpreters unakzeptabel lange dauern würde.

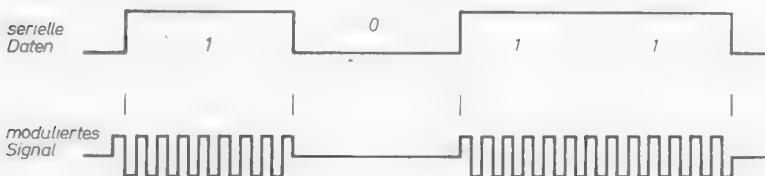


Bild 2.41  
Aufzeichnung mit  
Amplitudenmodulation

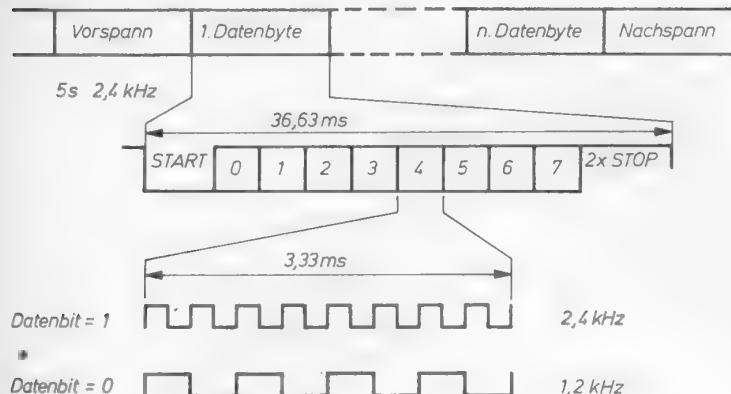


Bild 2.42  
Aufzeichnung nach  
dem Kansas-City-Ver-  
fahren

Weiterhin können Amplitudenschwankungen oder ein falsch eingestellter Pegel bei der Wiedergabe zu Lesefehlern führen. Diese Nachteile lassen das Laden von langen Programmen oder Daten mitunter zur Qual werden.

Auch die Frequenzmodulation ist zum Codieren geeignet. Ein Beispiel dafür ist das vielfach verwendete Modulationsverfahren nach dem Kansas-City-Standard (Bild 2.42). Ein L-bit besteht aus 4 Schwingungen (1,2 kHz), ein H-bit aus 8 (2,4 kHz). Moduliert wird mit einem Multiplexer, dem die beiden Frequenzen (1,2 und 2,4 kHz) zugeführt werden. Das zu übertragende Datenbit wählt die entsprechende Frequenz aus, indem es die Auswahlleitung des Multiplexers steuert. Zur Demodulation benutzt man einen Impulsbreitendifferenzkennung. Der Kansas-City-Standard ermöglicht eine recht sichere Datenaufzeichnung. Allerdings beträgt die Übertragungsgeschwindigkeit nur 300 baud, so daß auch dieses Modulationsverfahren für die Aufzeichnung großer Datenmengen ebenfalls nicht geeignet ist. Das Speichern eines 1-Kbyte-Blocks auf Magnetband dauert etwa 38 s.

Weitaus höhere Datendichten erreicht man mit den verschiedenen PCM-Verfahren, die auch in der EDV-Technik verwendet werden. Das benutzte Kassetteninterface arbeitet mit Phase-Encoding (Richtungstaktschritt). Bild 2.43 zeigt den Code sowie die Modulation und Demodulation. Eine Exklusiv-Oder-Verknüpfung von Takt (1) und Daten (2) bewirkt die Modulation. Die Impuls-

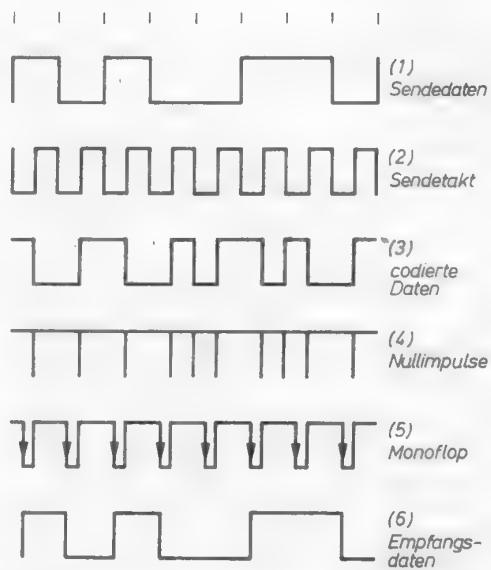


Bild 2.43  
Aufzeichnung nach dem Phase-Encoding-Verfahren

folge (3) enthält die codierten Daten: Ein H-bit wird durch einen H/L-Sprung und ein L-bit durch einen L/H-Sprung in der Mitte des bits gekennzeichnet. Bei aufeinanderfolgenden gleichen bits entstehen redundante Flußwechsel, die bei der Rückgewinnung der Information von den eigentlichen Flußwechseln unterschieden werden müssen. Das wird durch die Aufzeichnung von Synchronzeichen am Beginn eines Datenblocks erreicht. Bei der Wiedergabe gelangt das vom Kasset-

tenrecorder kommende Signal an einen Nulldurchgangsdetektor. Die Nulldurchgänge (4) triggern einen monostabilen Multivibrator, der nach 3/4 der Bitlänge wieder zurückkippt (5). Mit der entstehenden H/L-Flanke wird die Information in das Empfangsschieberegister geschoben.

Mit einem Interface nach diesem Verfahren kann man recht beachtliche Übertragungsgeschwindigkeiten erreichen. Im Mustergerät sind es 2880 baud, d. h., 1 Kbyte werden in nur 3,5 s aufgezeichnet.

## 2.6.2. Schaltungbeschreibung

Mit der Schaltung lassen sich unterschiedliche Übertragungsgeschwindigkeiten (Baudraten) realisieren. Damit kann das Interface an die Eigenschaften des verwendeten Kassettenrecorders optimal angepaßt werden. Der etwas höhere Bauelementeaufwand, im Vergleich zu anderen denkbaren Schaltungslösungen, bringt neben der einstellbaren Baudrate noch den zusätzlichen Vorteil, daß man das Interface ohne komplizierten Abgleich in Betrieb nehmen kann. Bild 2.44 zeigt die Schaltung. Zentrale Bausteine sind die SIO U 856 und der CTC U 857 (siehe auch Abschnitt 2.7.). Das Kassetteninterface benutzt das Port A der SIO. Port A hat getrennte Eingänge für den Sendetakt (Tx C) und den Empfangstakt (Rx C), was für die Funktion der Schaltung erforderlich ist. Kanal 2 des CTC-Bausteins erzeugt den Takt. Das Gatter D1.1 arbeitet als Takttreiber für den Takt T.

Der Modulator ist sehr einfach aufgebaut. Er besteht aus dem Zähler D2 und einem Exklusiv-Oder D1.2. Der Zähler D2 teilt den Takt T durch 8. Er erzeugt den notwendigen symmetrischen Modulationstakt, der entsprechend Bild 2.43 erforderlich ist (man muß bedenken, daß der CTC-Baustein an seinen Ausgängen nur stark unsymmetrische Nulldurchgangsimpulse liefert). Dazu würde an sich ein einfaches Flip-Flop genügen. Die Teilung durch 8 bewirkt lediglich die Angleichung von Empfangs- und Sendetakt. Damit spart man einen CTC-Kanal ein. An QC des Zählers wird der Sendetakt TxCA abgenommen und dem SIO-Baustein zugeführt. Die

Baudrate entspricht der Frequenz des Sendetakts TxCA. Jede fallende Flanke des Sendetakts schiebt ein bit aus dem internen Senderegister der SIO. Dieses bit wird nun mit dem Takt TxCA exklusiv-oder-verknüpft. Am Ausgang von Gatter D1.2 steht das codierte Signal. Über R1 und R2 gelangt es an den Kassettenrecorder. Der Demodulator ist etwas aufwendiger. Bild 2.45 verdeutlicht die Funktionsweise der Empfangsschaltung. Das Eingangssignal gelangt zum Operationsverstärker N1. An seinem Ausgang liegt das begrenzte Wiedergabesignal, welches durch das D-Flip-Flop D4.1 mit dem Takt T synchronisiert wird. Diese Synchronisationsschaltung verringert den Einfluß von Bandjitter und Störimpulsen auf die Demodulation und erhöht damit die Übertragungssicherheit. Die synchronisierten Daten werden über das Exklusiv-Oder D1.4 dem Eingang RxDA der SIO zugeführt. Mit dem Schalter S1 wird die Phasenlage der Daten an RxDA festgelegt. Die Stellung des Schalters ist abhängig vom verwendeten Kassettenrecorder. Bei falscher Phasenlage werden invertierte Daten gelesen. Die Stellung des Schalters muß man bei der Inbetriebnahme durch Versuch ermitteln. Den Empfangstakt RxCA erzeugen die Schaltungselemente D1.3, D4.2 und D3. Der Zähler D3 und das D-Flip D4.1 bilden einen monostabilen Multivibrator. Das Exklusiv-Oder-Gatter D1.3 erzeugt bei jedem Nulldurchgang des Eingangssignals einen kurzen L-Impuls, welcher das Monoflop triggert. An Hand von Bild 2.46 soll die Funktionsweise erläutert werden. Ausgehend von der Ruhestellung des Monoflop sei am Ausgang Q des Flip-Flop D4.2 L-Pegel. Dieser Ausgang ist mit dem Ladeeingang des Zählers D3 verbunden. Der an den Dateneingängen (mit den DIL-Schaltern S2 bis S5) vorgewählte Wert wird in den Zähler übernommen. Der von D1.3 erzeugte Nulldurchgangsimpuls setzt das Flip-Flop D4.2, der Ausgang Q wird H. Damit ist der Zähler freigegeben. Er beginnt mit dem nächsten Taktimpuls T rückwärts zu zählen. Bei Erreichen des Zählerstands «Q» entsteht ein Übertragsimpuls, der das D-Flip-Flop triggert, das nun zurückkippt. Somit ist die Ausgangsstellung (Q = L) wieder erreicht. Am Ausgang Q des Flip-Flop liegt die schon in Bild 2.43 darge-

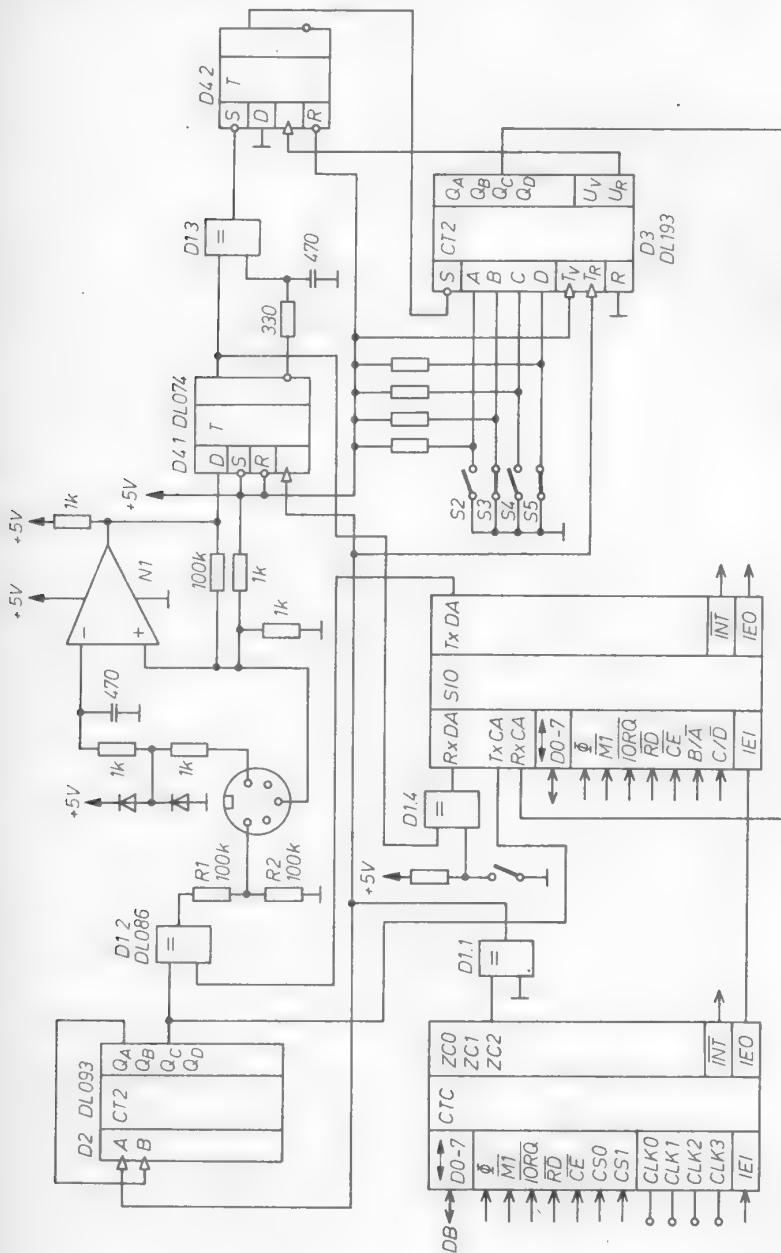


Bild 2.44 Schaltung des Kassetteninterface

stellte Impulsfolge (5). Als Empfangstakt RxCA für die SIO wird das Ausgangssignal QC des Zählers D3 verwendet. Mit jeder steigenden Flanke des Empfangstakts übernimmt die SIO die in RxDA liegende Infor-

mation in das Empfangsschieberegister. Die Lage dieser Flanke und damit die Kippzeit des Monoflop wird mit den DIL-Schaltern S2 bis S5 auf  $\frac{1}{4}$  der Bitlänge eingestellt. Im Mustergerät beträgt der Einstellwert 5.

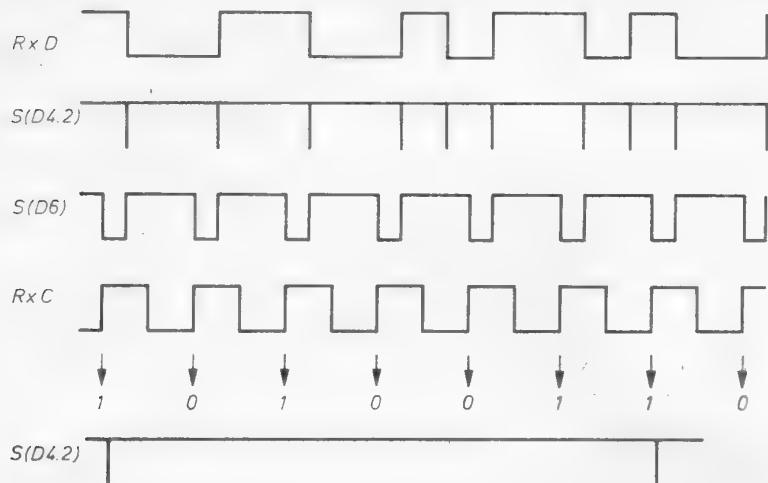


Bild 2.45  
Impulsdiagramm des  
Demodulators

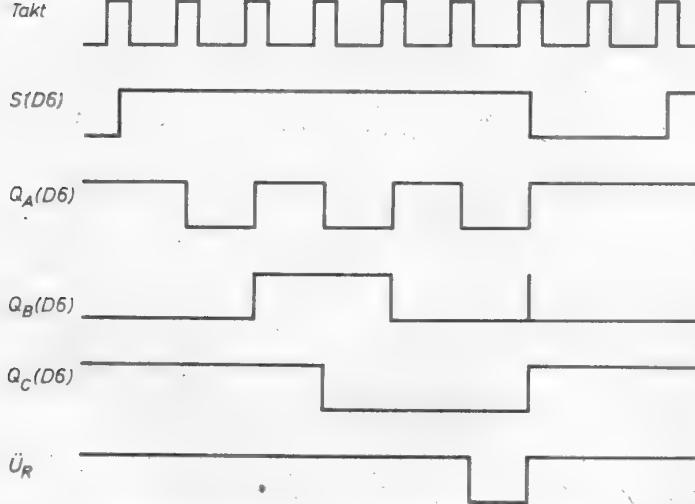


Bild 2.46  
Impulsdiagramm des Mono-  
flop

### 2.6.3. Inbetriebnahme und Anwendung

Die Inbetriebnahme ist mit der später beschriebenen Betriebssoftware nicht schwierig. Es wird am besten mit SAVE ein kurzer Datenblock mit Vollaussteuerung (selbst eine leichte Übersteuerung schadet nicht) aufgezeichnet. Eine eventuelle Aussteuerungsautomatik schaltet man vorher ab. Mit LOAD erfolgt die Wiedergabe. Werden die Daten nicht gelesen, so ist die Phasenlage des Eingangssignals mit dem Schalter S1 zu ändern. Dann müßte der aufgezeichnete Block wieder geladen werden. Lesefehler erfordern ein Überprüfen der Pegel bei Auf-

nahme und Wiedergabe. Der optimale Abtastzeitpunkt läßt sich mit den Schaltern S2 bis S5 einstellen. Wenn erforderlich, kann man den Abtastzeitpunkt korrigieren. Sollwert ist 5. Notfalls muß die Übertragungsgeschwindigkeit reduziert werden, indem die Baudrate durch Verändern der Zeitkonstante des CTC-Kanals 2 reduziert wird. Siehe dazu Abschnitt 2.7. und Abschnitt 2.8.

Die maximal erzielbare Übertragungsgeschwindigkeit hängt vom verwendeten Magnetbandgerät ab. Einfache Mono-Kassettenrecorder eignen sich oft besser, als gute HiFi-Geräte. HiFi-Tapedecks erreichen ihre guten Übertragungseigenschaften unter an-

derem durch stark gegengekoppelte Verstärkerstufen. Gerade dieser Umstand ist beim gewählten Phase-Encoding nicht unbedingt vorteilhaft. Das vom Autor realisierte Interface wird mit dem Kassettenrecorder *Geracord GC-6020 portable* betrieben. Er verfügt über eine relativ genau arbeitende Bandzählühr, mit der sich Datenaufzeichnungen schnell auffinden lassen. Auch andere Geräte wurden mit Erfolg mit diesem Interface betrieben.

Die Übertragungsrate sollte nicht wesentlich kleiner als 1200 baud sein. Die im Mustergerät gewählte Baudrate von 2880 baud stellt einen guten Kompromiß zwischen möglichst geringer Fehlerzahl und hoher Geschwindigkeit dar (1 Kbyte wird in 3,5 s aufgezeichnet). Allerdings ist hochwertiges Bandmaterial die Voraussetzung für die erreichten Werte. Gute Eigenschaften haben die Chromdioxid-Kassetten K60. 90-Minuten-Kassetten sind wenig geeignet. Sie sind auf Grund ihrer Länge recht unpraktisch (lange Suchzeiten). Die Anzahl der Lesefehler war beim Einsatz von hochwertigem Bandmaterial immer sehr gering. Treten bei der Wiedergabe dennoch Lesefehler auf, so sorgt die vorhandene Software für eine selbständige Fehlerkorrektur. Somit steht ein zuverlässiger, schneller und billiger Massenspeicher zur Verfügung.

#### 2.6.4. Aufzeichnungsformate

Bisher wurden vor allen Dingen die Hardwareaspekte der Aufzeichnung von Daten auf Audiokassetten erörtert. Einen wesentlichen Beitrag zur sicheren Datenspeicherung auf Magnetband leistet die notwendige Betriebssoftware. Die Software wird in Abschnitt 2.8. beschrieben. Vorausgreifend sollen hier jedoch schon einige Ausführungen zu den verwendeten Aufzeichnungsverfahren gemacht werden.

Es wird grundsätzlich asynchron aufgezeichnet. Das verwendete asynchrone Datenformat zeigt Bild 2.47. Das Datenbyte wird von einem Start- und einem Stopbit eingeschlossen. Auf das Paritätsbit wurde verzichtet. Selbstverständlich sind auch synchrone Übertragungen möglich. Sie bringen hier

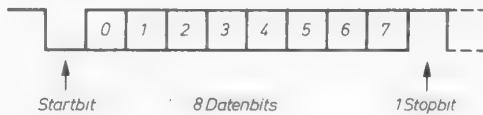


Bild 2.47  
Verwendetes asynchrones Datenformat

aber keine bedeutenden Vorteile. Mehr über andere Übertragungsarten kann man in Abschnitt 2.2. nachlesen. Das selbständige Korrigieren eventuell auftretender Lesefehler wird durch eine geeignete Formatierung der Daten ermöglicht. Bild 2.48 verdeutlicht, daß die aufzuzeichnenden Daten in einzelne Blöcke zu je 128 byte (oder weniger, falls die Anzahl der zu übertragenden bytes kleiner 128 ist) aufgeteilt werden. Ein Datenblock beginnt mit 6 Synchronzeichen (FFH). Es folgt das ASCII-Steuerzeichen GS (group separator = Gruppentrennung, Code 1DH), das den Beginn eines Datenblocks kennzeichnet. Das nächste byte enthält die Anzahl n der nun folgenden Datenbytes. Nach der Übertragung der n Datenbytes wird ein Prüfsummenbyte gesendet. Dieses byte ist das Zweierkomplement der Summe aller n Datenbytes einschließlich n. Ein Prüfsummenbyte hat sich als ausreichend erwiesen.

Zu Beginn jeder Datei wird ein Block aufgezeichnet, der alle wichtigen Datenparameter enthält. Diesen Block nennt man Kopf. Er beginnt mit 20 Synchronzeichen (FFH). NL (1EH) kennzeichnet den Beginn der Aufzeichnung, wird also nur vor Beginn der Datei 1 (Bild 2.48) gesendet. Die Kennung des Kopfs ist das ASCII-Zeichen SOH (start of heading = Anfang des Kopfs, Code 01H). Die nächsten 8 Zeichen enthalten den Dateinamen. Die weiteren Dateiparameter sind:

- TT Dateityp (maximal 2 Zeichen), z. B.
  - P Maschinenprogramme mit Autostart
  - BA BASIC-Programm im ASCII-Format
  - BI BASIC-Programm im interpreterinternen Format
  - A Textdatei
  - AA Anfangsadresse (2 byte)
  - EE Endadresse (2 byte)
  - SS Startadresse für Autostart (2 byte)
  - TF Textfeld (16 Zeichen)

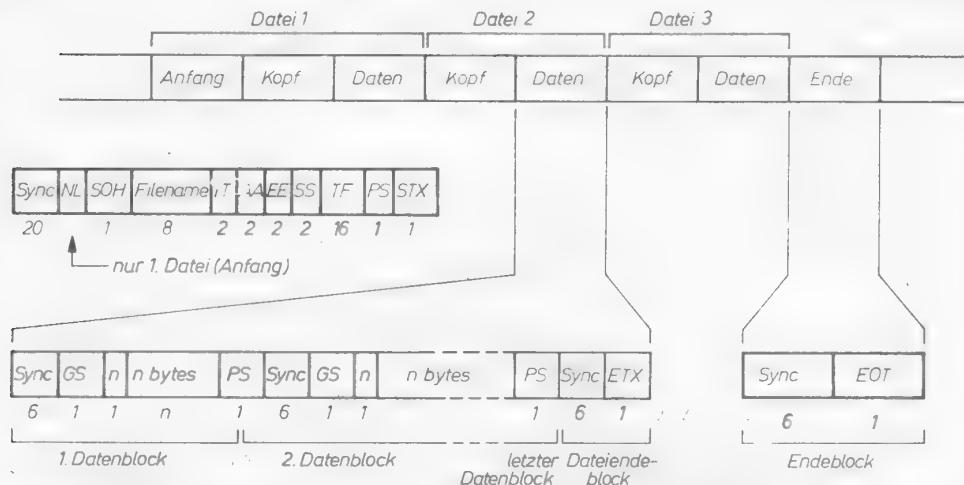


Bild 2.48  
Verwendetes Datenformat für Kassettenaufzeichnung

Im Textfeld können beispielsweise das Erstellungsdatum oder der Bearbeitungszustand der Datei eingetragen werden. Anschließend folgt die Prüfsumme des Kopfes. Die Prüfsumme ist das Zweierkomplement der Summe der 32 bytes des Kopfes (beginnend mit dem Filenamen). Schließlich beendet das Steuerzeichen STX (start of text = Anfang des Textes, Code 02H) den Kopfblock, zugleich zeigt STX (wie der Name schon sagt) den Beginn der eigentlichen Datenblöcke an. Sind alle Datenblöcke übertragen, beendet ein kurzer Block die Datei. Dieser Dateiendeblock besteht aus 6 Synchronzeichen (FFH) und ETX (end of text = Ende des Textes, Code 03H). Die gesamte Datei wird noch zweimal übertragen. Die Aufzeichnung endet mit dem Endeblck, der aus 6 Synchronzeichen (FFH) und dem ASCII-Zeichen EOT (end of transmission = Ende der Übertragung, Code 04H) besteht.

Das eben beschriebene Dateiformat wird vom Monitorprogramm erzeugt.

Reine ASCII-Dateien werden anders aufgezeichnet. Der Kopf ist wie oben beschrieben aufgebaut. Allerdings haben die Fileparameter AA, EE und SS keine Bedeutung und sind zu 0 gesetzt. Jeder Datenblock beginnt mit Synchronzeichen (0FFH). Es fol-

gen beliebige ASCII-Zeichen. Das Ende der Zeichenkette ist CR (carriage return, Code 0DH) und/oder LF (line feed, Code 0AH). CR/LF schließt immer eine Zeile und damit einen Datenblock ab. Die ASCII-Aufzeichnung wird also immer zeilenweise durchgeführt. Es liegt in der Natur der Dinge, daß eine 3malige Speicherung der Datei dabei oft nicht möglich und deshalb nicht vorgesehen ist.

Nicht nur die Hardware, sondern auch die Software wird für die Wiedergabe umfangreicher. Sie und die Aufnahmesoftware wurden als Assemblerlisting bereits in [2] veröffentlicht. Die Programme für SAVE und LOAD dieses Computers sind im Detail etwas modifiziert worden, trotzdem kann aber am Listing in [2] die Funktionsweise studiert werden.

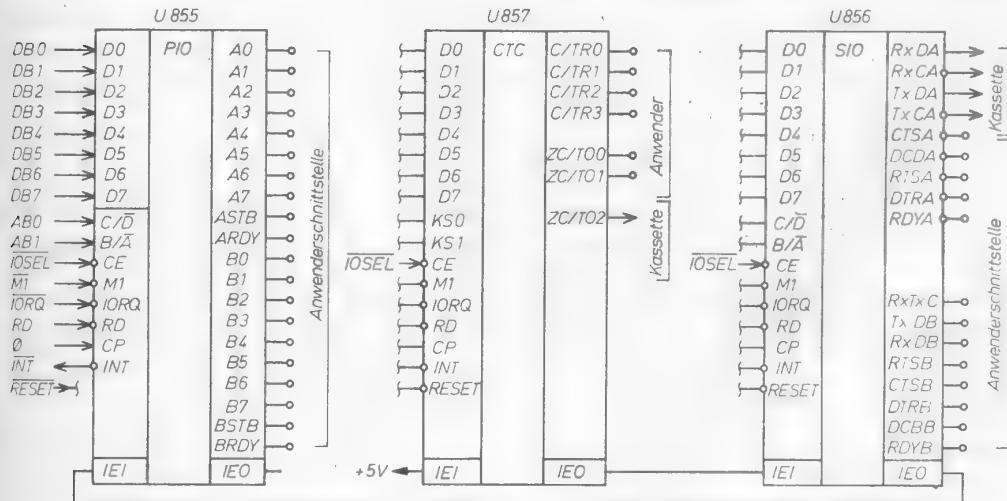
Eine Datei wird eingelesen, wenn der auf Band abgespeicherte Dateiname mit dem eingegebenen Filenamen übereinstimmt. Dabei sind alle 8 Zeichen des Filennamens signifikant. Bei der Wiedergabe werden die einzelnen Datenblöcke mitgezählt. Traten keine Lesefehler auf, so ist nach Erreichen des Dateiendeblocks (ETX) die Wiedergabe beendet. Lagen jedoch Lesefehler vor, so wird die nächste Datei gelesen, und nur die beim ersten Laden fehlerhaften und «germekten» Datenblöcke werden neu geladen. Das Einlesen steuert man über einige vom LOAD-Programm verwaltete Flags. Bei mangelhaftem Bandmaterial oder mechanischen

Fehlern des Kassettenrecorders kann die Anzahl der Lesefehler so groß werden, daß eine Korrektur durch Nachladen der fehlerhaften Datenblöcke nicht mehr möglich ist. In diesem Fall wird keine Korrektur versucht, sondern die Datei neu geladen. Da jede Datei automatisch 3mal aufgezeichnet wurde, ist in der Regel (auch bei weiteren Lesefehlern) das Laden (und Korrigieren) des Files möglich. Nur in ganz seltenen Fällen wird beim Lesen der EOT-Block erreicht, dann war keine fehlerfreie Wiedergabe möglich. Oft kann eine solche Aufzeichnung nach mehreren Versuchen doch noch geladen werden. Dann sollte man nicht vergessen, diese Datei noch einmal neu aufzuzeichnen.

## 2.7. Ein- und Ausgabebaugruppe

Bisher wurden schon verschiedene I/O-Einheiten beschrieben: die Tastatur und das Kassetteninterface. Dieser Abschnitt behandelt nun die Schnittstellen des Computers, die vom System selbst nicht verwendet werden und deshalb dem Anwender frei zur Verfügung stehen. Die Verschaltung der I/O-Bausteine mit dem Bus zeigt Bild 2.49.

Bild 2.49  
Schaltung der I/O-Baugruppe



### 2.7.1. Parallele Schnittstelle

Als parallele Schnittstelle ist ein PIO-Baustein U855 vorgesehen. Mit ihm stehen zwei 8-bit-bidirektionale Ports zur Verfügung. Zusätzlich verfügt jedes Port über die beiden Handshake-Leitungen READY und STROBE, die zur Ablaufsteuerung eingesetzt werden können. Alle Ein-/Ausgabe-Leitungen einschließlich der Handshake-Leitungen des PIO-Bausteins werden auf die I/O-Steckerleiste geführt. Um eine möglichst universelle Schnittstelle zu schaffen, die sowohl für Eingabe als auch für Ausgabe geeignet ist, wurde auf den Einsatz von Treiberschaltkreisen oder anderen Entkoppelungsstufen auf der Peripherieseite der PIO verzichtet. Beim Anschluß von eigenen Baugruppen an die PIO-Schnittstelle muß man deshalb unbedingt die elektrischen Kennwerte des PIO-Bausteins einhalten. Diese können beispielsweise [1] entnommen werden.

Der PIO-Baustein hat die Kanäle A und B. Der Kanal A kann in 4, der Kanal B in 3 verschiedenen Betriebsarten arbeiten:  
 Mode 0 Byteausgabe,  
 Mode 1 Byteeingabe,  
 Mode 2 Bidirekionaler Betrieb,  
 Mode 3 Bitemp-/ausgabe.

Die gewünschte Betriebsart wird durch Ausgabe des entsprechenden Steuerworts eingestellt. Eine Zusammenfassung der Program-

Tabelle 2.5. Programmertabelle der PIO

| Steuerwort      | D7  | D6  | D5   | D4  | D3  | D2  | D1  | D0  | Bemerkungen  |
|-----------------|-----|-----|------|-----|-----|-----|-----|-----|--|
| Interruptvektor | V7  | V6  | ' V5 | V4  | V3  | V2  | V1  | 0   |  |
| Betriebsart     | M1  | M0  | X    | X   | 1   | 1   | 1   | 1   | OFH Byteausgabe, 4FH Byteeingabe, 8FH bidirektional, CFH Bitbetrieb                                  |
| I/O-Auswahl     | IO7 | IO6 | IO5  | IO4 | IO3 | IO2 | IO1 | IO0 | bei Bitbetrieb: 1 = Eingang, 0 = Ausgang   |
| Interruptart    | EI  | A/O | H/L  | MF  | 0   | 1   | 1   | 1   | bei Bitbetrieb: EI: Interruptfreigabe, A/O: AND(1)/OR(0), H/L: Flanke (L/H = 1), MF: Maske folgt (1) |
| Maske           | MB7 | MB6 | MB5  | MB4 | MB3 | MB2 | MB1 | MB0 | bei Bitbetrieb: 0 = Interrupt möglich  |
| Interruptenable | EI  | X   | X    | X   | 0   | 0   | 1   | 1   | 83H Interrupt erlaubt, 03H Interrupt gesperrt  |

miermöglichkeiten enthält Tabelle 2.5. Weitergehende Informationen zur PIO, z. B. Zeitabläufe und Programmierung sind in [1] und [4] ausführlich beschrieben. Programmbeispiele findet man in [3]. Hier nur eine Kurzbeschreibung der Betriebsarten.

### Mode 0

Die CPU kann ein Datenbyte auf das Kanal-Ausgaberegister und damit auf die Port-Datenleitungen ausgeben (dieses Datenbyte wird bei Bedarf zu Kontrollzwecken auch von der CPU wieder gelesen). Unabhängig davon, ob die Peripherie das Datenbyte auch wirklich übernommen hat, besteht die Möglichkeit, daß die CPU jederzeit ein neues byte ausgibt. Oft ist es wünschenswert, erst dann ein neues byte auszugeben, wenn das externe Gerät das alte byte sicher übernommen hat. Hierzu benutzt man die beiden Handshake-(«Hände-schüttel»-) Leitungen READY und STROBE. Die READY-Leitung wird nach Ausgabe eines Datenbytes auf das Datenausgaberegister aktiv, d.h., READY bekommt H. Das externe Gerät übernimmt nun nach Erkennen der Aktivierung des READY-Signals das anliegende byte. Daraufhin wird die Übernahme durch Aktivierung von STROBE (L) quittiert, worauf READY wieder in den nichtaktiven Zustand (L) geht. Leider hat die PIO kein Statusregister, so daß die CPU nicht den Zustand der READY-Leitung abfragen kann. Die einzige Möglichkeit, das Quittieren der Peripherie der CPU mitzuteilen, ist der Interrupt. Dieser wird mit der steigenden Flanke des STROBE-Signals ausgelöst, natürlich nur, wenn das Interrupt-Freigabe-Flip-Flop gesetzt ist.

### Mode 1

Die CPU kann mit einem IN-Befehl ein byte aus dem Dateneingaberegister lesen. Die anliegende Information wird mit dem aktiven STROBE-Signal übernommen und zwischengespeichert. Dabei gibt es die einfache Möglichkeit, STROBE ständig aktiv zu schalten (also einfach mit Masse verbinden, allerdings ohne Zwischenspeicherung). Vielfach ist die Eingabe mit den Handshakesignalen günstiger. Der Ablauf ist einfach: Das aktive READY-Signal zeigt dem externen Gerät, daß es ein neues Datenbyte in das Eingaberegister schreiben kann. Zum Einschreiben aktiviert man die STROBE-Leitung (durch die Peripherie). READY wird entaktiviert, bis die CPU das PIO-Port gelesen hat. Wie schon bei der Byteausgabe gibt es nur die Möglichkeit der Interruptauslösung, um der CPU den Status der READY-Leitung mitzuteilen.

Bei der Anwendung der Handshakesignale muß man beachten, daß nach der Initialisierung des Kanals die READY-Leitung inaktiv ist. Durch einen IN-Befehl wird READY aktiv geschaltet, und das Port ist für die Eingabe mit «Hände schütteln» bereit.

### Mode 2

Im Mode 2 wird der Kanal A gleichzeitig zur Ein- und Ausgabe benutzt. Die bidirektionale Betriebsart ist nur mit dem Kanal A möglich, weil die Handshakesignale beider Kanäle für die Ein-/Ausgabesteuerung benötigt werden. Anders als bei der Byteausgabe (Mode 0) liegt das Ausgabebyte nicht ständig an den Portleitungen an, da es zu Konflikten zwischen ein- und ausgehenden Daten kommen könnte. Die Portleitungen werden zur Ausgabe freigegeben, wenn

STROBE des Kanals A aktiv (L) ist. Die Daten lassen sich wie im Mode 1 eingeben. Zur Steuerung werden die Leitungen des Kanals B verwendet.

Diese Betriebsart hat vor allen Dingen für die Rechnerkopplung Bedeutung.

### Mode 3

Jede der 8 Datenleitungen kann man beliebig als Ein- oder Ausgänge definieren. Dementsprechend komplizierter ist die Programmierung. Unmittelbar nach der Ausgabe des Steuerworts für bit E/A (CFH) ist die Ein-/Ausgangsdefinition erforderlich. Tabelle 2.5. zeigt das Steuerwort. I steht für Eingabe. Soll die Leitung zur Eingabe definiert werden, so ist das entsprechende bit zu 1 zu setzen. O steht für Ausgabe, das entsprechende bit ist zu 0 zu setzen.

Im Mode 3 werden die Quittierungssignale READY und STROBE nicht verwendet. Die Eingabedaten lassen sich also nicht zwischenspeichern.

## 2.7.2. Serielle Schnittstelle

Zur seriellen Ein- und Ausgabe ist eine SIO vorgesehen. Sie wandelt 8-bit-parallele in 1-bit-serielle Informationen um. Somit sind nur sehr wenige Leitungen zur Datenübertragung erforderlich. Weiterhin arbeiten viele externe Geräte von Natur aus seriell, beispielsweise Fernschreiber, Magnetbandspeicher oder Floppy Disk. Ein Interface für die Datenaufzeichnung auf Magnetband wurde bereits in Abschnitt 2.6. beschrieben. Für das Kassetteninterface wurde der Kanal A der SIO verwendet. Der Kanal B steht frei zur Verfügung. Einige Anwendungen, die für den Amateur interessant sind, werden im folgenden vorgestellt.

Man verwendet dazu die einfachere asynchrone Datenübertragung. Bild 2.50 zeigt das Übertragungsformat. Die zu einem Zeichen gehörenden Daten werden nacheinan-

der bit für bit gesendet oder empfangen. Jedes Zeichen erhält ein Startbit. Es folgen die einzelnen bits des Datenworts. Zur Fehlerkontrolle kann nun das Paritätsbit folgen. Es gibt verschiedene Möglichkeiten, ein Paritätsbit zu senden. Bei gerader Parität (even) wird das bit auf logisch 1 gesetzt, wenn das zu sendende Zeichen eine ungerade Anzahl von gesetzten bits enthält, d. h., bei einer geraden Anzahl von «Einsen» ist das Paritätsbit nicht gesetzt. Im Falle der ungeraden Paritätsprüfung (odd) wird das bit entgegengesetzt erzeugt. Der Empfänger prüft das empfangene Zeichen nach der gleichen Vorschrift. Er kann auf diese Art und Weise Übertragungsfehler erkennen (das Paritätsbit ermöglicht allerdings keine vollständige Übertragungssicherheit). Die anschließende Übertragung eines oder zweier Stopbits beendet das Zeichen. Die Start- und Stopbits gestatten die Synchronisation. Entscheidend für einen reibungslosen Datentransfer ist die Übereinstimmung der Datenformate und Übertragungsgeschwindigkeiten auf der Sender- und Empfängerseite. Übertragen wird normalerweise mit genormten Geschwindigkeiten. Dabei sind z. B. folgende Baudraten üblich (1 baud = 1 bit/s):

45,45, 50, 110, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200 baud.

Die Sende- und Empfangstakte können auf unterschiedliche Weisen erzeugt werden. Es bietet sich natürlich an, die Takte mit dem CTC-Baustein aus dem Systemtakt zu bilden. Durch den gewählten Systemtakt von 1,5 MHz ist die genaue Erzeugung der genormten Takte allerdings nicht möglich. Die Tabelle 2.6. zeigt einige CTC-Teilverhältnisse mit den sich daraus ergebenden Baudraten. Die Angaben gehen davon aus, daß sowohl der entsprechende CTC-Kanal als auch die SIO mit einem Vorteilverhältnis von 16 programmiert wird.

Zum seriellen Informationsaustausch zwischen dem Computer und anderen Geräten kann eine V24-Schnittstelle vorgesehen wer-

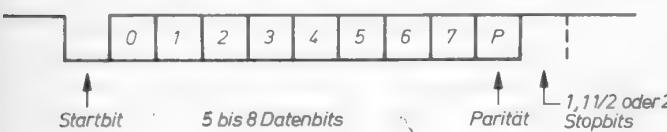
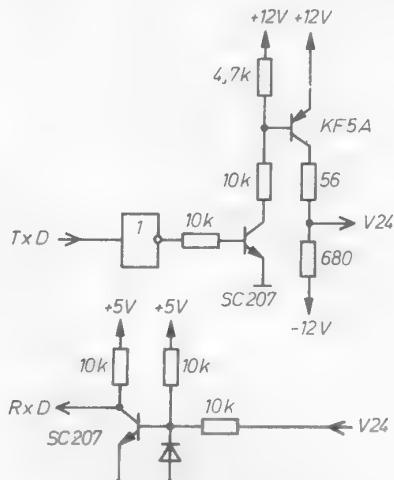


Bild 2.50  
Asynchrones Datenformat

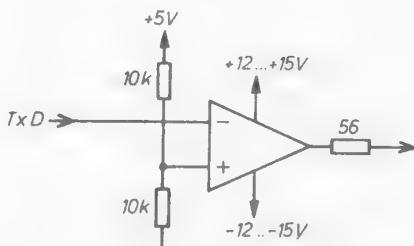
**Tabelle 2.6.**  
Der Zusammenhang zwischen CTC-Zeitkonstante und Baudrate

| CTC-Zeitkonstante | Baudrate | Baudrate des Kassetteninterface |
|-------------------|----------|---------------------------------|
| 1                 | 5 859.4  | 11 718.8                        |
| 2                 | 2 929.7  | 5 859.4                         |
| 3                 | 1 953.1  | 3 906.3                         |
| 4                 | 1 464.8  | 2 929.7                         |
| 5                 | 1 171.9  | 2 343.8                         |
| 6                 | 976.6    | 1 953.1                         |
| 7                 | 837.1    | 1 674.1                         |
| 8                 | 732.4    | 1 464.8                         |
| 9                 | 651.0    | 1 302.1                         |
| 10                | 585.9    | 1 171.9                         |
| 11                | 532.7    | 1 065.3                         |
| 12                | 488.3    | 976.6                           |
| 13                | 450.7    | 901.4                           |
| 14                | 418.5    | 837.1                           |
| 15                | 390.6    | 781.3                           |
| 16                | 366.2    | 732.4                           |
| 17                | 344.7    | 689.3                           |
| 18                | 325.5    | 651.0                           |
| 19                | 308.4    | 616.8                           |
| 20                | 293.0    | 585.9                           |
| 117               | 50.1     |                                 |

den. Ursprünglich wurde diese Schnittstelle (RS 232C) in der Großrechnertechnik verwendet, um den Rechner an ein Modem anzuschließen. Nun hat sich diese Schnittstelle aber auch in der Mikrocomputertechnik zur Verbindung von Rechnern und Peripherie durchgesetzt. Das periphere Gerät kann ein anderer Computer oder ein Drucker sein. Viele Drucker arbeiten mit dieser Schnittstelle. Das Kombinat VEB *robotron* entwickelte einige Geräte, die auch für die Ergänzung von Kleincomputern vorgesehen sind. Dazu gehören die beiden Thermodrucker *K 6303* (40 Zeichen je Zeile) und *K 6304* (80 Zeichen je Zeile) sowie die elektronische Reiseschreibmaschine *Erica electronic portable S 6005*. All diese Geräte verfügen über eine V24-Schnittstelle. Um den Anschluß der Drucker zu ermöglichen, muß der SIO-Kanal B mit Anpaßstufen, die die notwendigen Übertragungspegel erzeugen, ergänzt werden. Die V24-Schnittstelle arbeitet mit den Pegeln +12 V (0-bit) und -12 V (1-bit). Durch die Wahl dieser Pegel wird eine sichere Datenübertragung auch über große Entfernungen garantiert. Bild 2.51 zeigt die notwendigen Anpaßstufen. Bild 2.52 zeigt



**Bild 2.51**  
V24-Anpaßstufen



**Bild 2.52**  
Erzeugung der V24-Pegel mit einem OPV

einen V24-Treiber mit Operationsverstärker, der für kurze Leitungen gut geeignet ist.

Außer den seriellen Sende- und Empfangsdaten (Tx D und Rx D) werden auch noch die folgenden Steuersignale auf die Schnittstelle geführt:

RTSB Sendaufforderung (Ausgang),  
DTRB Datenstation bereit (Ausgang),  
DCDB Empfängerfreigabe (Eingang),  
CTSB Senderfreigabe (Eingang).

Mit diesen Steuersignalen können verschiedenen schnellen Geräte miteinander synchronisiert werden. Beispielsweise läßt sich mit Hilfe des CTS-Eingangs das Senden von Daten ausschließen, wenn das periphere Gerät die eingetroffenen Daten noch nicht verarbeitet hat. Umgekehrt kann das Signal RTS verhindern, daß das periphere Gerät weitere

Daten sendet, solange der Computer die empfangenen Informationen noch nicht verarbeitet hat. Dazu muß man die Anschlußpunkte CTS und RTS des Computers und des peripheren Geräts kreuzweise miteinander verbinden.

Eine weitere Ausführungsform eines seriellen Interface kann die Stromschnittstelle sein. Es wird in diesem Fall meist mit 20-mA-Strömen gearbeitet. Bekannte Beispiele sind IFSS, Fernschreiber und MIDI (MIDI ist eine Norm, die die Kopplung verschiedener Musikinstrumente untereinander und mit dem Computer erlaubt).

Bereits in Abschnitt 2.5. wurde der Aufbau einer seriellen Tastatur vorgeschlagen. Tastatur und Rechner können im einfachsten Fall mit TTL-Pegeln verbunden sein. An diesem Anwendungsbeispiel soll die Programmierung der SIO gezeigt werden (die Einstellung der Baudrate wird nicht berücksichtigt):

INIT: LD C,CSIOB OB)

LD B,6

LD HL,TAB

OTIR

RET

|             |   |
|-------------|---|
| TAB: DEFB 4 | ;Register WR4                           |
| DEFB 48H    | ;1 Stopbit, keine Parität, Taktmode x16 |
| DEFB 1      | ;Register WR1                           |
| DEFB 08H    | ;keine Interrupts                       |
| DEFB 3      | ;Register WR3                           |
| DEFB OC1H   | ;8 bits, Empfängerfreigabe              |

|                    |                             |
|--------------------|-----------------------------|
| CSTS: IN A,(CSIOB) | ;Status lesen               |
| AND 1              |                             |
| RET Z              | ;Taste gedrückt             |
| OR 257             |                             |
| RET                | ;Taste gedrückt             |
|                    | ;Zeichen von Tastatur lesen |

|               |                             |
|---------------|-----------------------------|
| CI: CALL CSTS |                             |
| JR Z,CI       | ;warten, bis Taste gedrückt |
| IN A,(DSIOB)  | ;Tastaturcode lesen         |
| RET           |                             |

## 2.8. Monitorprogramm

Bisher wurde vor allen Dingen die Computerhardware behandelt. Das im folgenden beschriebene Monitorprogramm stellt die wesentlichen zur Funktion des Computers notwendigen Programme und Bedienerkommandos zur Verfügung. Es basiert auf dem schon in [2] als Assemblerlisting vorgestellten Monitor. Aufgrund der umfangreicher Systemroutinen (Grafik) mußte jedoch auf einige Kommandos verzichtet werden.

Das Monitorprogramm wurde unter folgenden Gesichtspunkten geschrieben:

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

### 2.8.1. Beschreibung der Monitorkommandos

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

Im folgenden sind die einzelnen Kommandos beschrieben. Dabei ist zu beachten, daß die Kommandonamen nicht ausgeschrieben werden müssen. So kann man beispielsweise MEMORY auch durch MEMOR, MEMO, MEM, ME oder M aufrufen. Alle eingegebenen Zeichen sind signifikant. Beginnen verschiedene Kommandonamen mit den gleichen Buchstaben und sind nur diese angegeben, so wird das zuerst gefundene Kommando ausgeführt. Ein Beispiel: M entspricht MEMORY, aber nicht MOVE, für MOVE muß mindestens MO eingegeben werden. In Zweifelsfällen schafft das Kom-

mando HELP Klarheit über die Reihenfolge der Kommandos.

Bei der Beschreibung der Software, also des Monitorprogramms und des BASIC-Interpreters, werden die folgenden Symbole zur einheitlichen Syntax- und Kommandobeschreibung verwendet:

**CR** Abschluß der Zeile (durch Drücken der RETURN-Taste).

[**I**] Die in eckigen Klammern eingeschlossenen Angaben sind optional und können wahlweise verwendet werden.

(**C**) Die in spitzen Klammern eingeschlossenen Angaben müssen verwendet werden.

**HELP** Anzeige aller im Speicher vorhandenen Kommandos HELP CR

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

**MEMORY** Anzeigen und Modifizieren von Speicherbereichen **MEMORY** <adr> CR

Es wird der Speicherbereich ab Adresse <adr> mit einer Länge von 256 bytes in hexadezimaler Form und in Textdarstellung ausgegeben. Bei der Textdarstellung werden nicht geschriebene Codes (00H bis 1 FH) als Punkt (.) ausgegeben.

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

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

**MOVE** Umladen von Speicherbereichen

**MOVE** <startadr> <endadr> <zieladr> CR

Es wird der Bereich von <startadr> bis einschließlich <endadr> nach <zieladr> kopiert.

Dabei können sich die Speicherbereiche auch überlappen.

**FILL** Füllen eines Speicherbereichs  
**FILL** <startadr> <endadr> [<byte>]  
CR

Der Speicherbereich von <startadr> bis einschließlich <endadr> wird mit dem angegebenen <byte> gefüllt. Fehlt das Füllbyte, so wird automatisch <byte> als 00 angenommen.

**IN** Porteingabe  
**OUT** Portausgabe  
    IN <adr> CR  
    OUT <adr> <byte> CR

Mit dem **IN**-Befehl lassen sich Ports direkt lesen. Das gelesene byte wird auf den Consoikanal ausgegeben (Bildschirm). Mit dem **OUT**-Befehl können z. B. Ports initialisiert werden. Dabei wird auf das Port <adr> der Wert <byte> ausgegeben.

**SAVE** Abspeichern auf Magnetband  
**SAVE** <filename> [<typ>] <startadr>  
          <endadr>  
          [<entry>] [<textfeld>]

**SAVE** zeichnet den Speicherbereich von <startadr> bis <endadr> unter dem angegebenen Filenamen auf Magnetband auf. Der Name kann dabei beliebig lang sein, es werden aber nur die ersten 8 Zeichen als <filename> übernommen. Zu beachten ist, daß der Name mit einem Buchstaben beginnen muß und kein Leerzeichen oder Komma enthält, da diese ja Trennzeichen bei der Kommandoeingabe sind. Durch Angabe eines Filetyps <typ> läßt sich die Datei besonders kennzeichnen. Die beschriebenen Programme erzeugen folgende Filetypen:

- **P** lauffähiges Maschinenprogramm, das nach dem Laden automatisch gestartet wird,
- **BA** aufgezeichnetes BASIC-Programm,
- **A** Aufzeichnung von im ASCII-Code abgespeicherten Daten.

Bei der Angabe eines Filetyps sind 2 Zeichen signifikant. Die Angabe eines Typs wird durch das Zeichen @ eingeleitet. Fehlt die Typangabe, so setzt das **SAVE**-Programm automatisch den Typ P ein (entspricht der Eingabe @P). Dateien vom Typ P erfordern noch einen weiteren Parameter, nämlich den Eintrittspunkt, bei dem das Programm nach dem Laden gestartet wird.

Fehlt `<entry>`, so wird als Eintrittspunkt automatisch die Restartadresse des Monitors eingetragen. Schließlich können im Textfeld noch zusätzliche Informationen, wie das Erstellungsdatum, folgen. Es werden 16 Textzeichen (eingeleitet durch ein Semikolon) aufgezeichnet.

#### **LOAD Laden vom Magnetband**

`LOAD <filename> [<@Q>] [<offset>]`

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

#### **GO Start von Maschinenprogrammen**

`GO <startadr> CR`

Mit `GO` läßt sich ein Programm starten. Die `<startadr>` muß angegeben werden. Das Anwenderprogramm kann mit `Ret (0C9H)` zum Monitorprogramm zurückkehren.

**SIZE** Anzeige des belegten Speichers  
`SIZE` gibt zunächst den für Anwenderprogramme (z. B. BASIC-Interpreter) zur Verfügung stehenden Speicherplatz aus. Diese Adresse kann durch Neueingabe verändert werden. Der BASIC-Interpreter (Abschnitt 2.9.) verwendet den `RAMTOP`-Wert in seiner Initialisierungsphase zur Speicherbereichsaufteilung. Nach Drücken der `RETURN`-Taste gibt `SIZE` die Adresse des derzeitigen Bildwiederholspeichers aus. 8 Kbyte ab dieser Adresse werden durch den Bildspeicher belegt. Auch diese Adresse läßt sich durch Neueingabe verändern. Zu beachten

ist, daß diese Adressen Vielfache von 8 Kbyte sein müssen. Bei Leereingaben wird der angezeigte Wert wieder abgespeichert. **IOBYTE** Veränderung der IO-Zuweisung Manchmal wird die Ausgabe nicht auf dem Bildschirm, sondern beispielsweise auf einem Drucker gewünscht. Der vorgestellte Monitor ermöglicht schon derartige Umleitungen. In dieser Monitorversion wurde das Konzept geändert. Es wird nun ein IO-byte verwendet, das die Zuordnung zwischen logischen und physischen Geräten übernimmt.

Die logischen Geräte sind:

#### **CON**

Die Konsole, d. h., das Standardein- und Ausgabegerät, welches die Kommunikation mit dem Rechner ermöglicht. Im vorliegenden Rechner ist der Konsole CON die Tastatur und der Bildschirm zugeordnet.

#### **RDR**

Vom Leserkanal (Reader) können Daten empfangen werden. Voreingestellt ist die Verbindung zum Kassetteninterface (SIO Port A).

#### **PUN**

Der Datenausgabekanal PUN steht für Punch (Stanzer). Über diesen Kanal werden Daten oder Programme ausgegeben. Auch hier ist die Verbindung zur Kassettenaufzeichnung (SIO Port A) voreingestellt.

#### **LST**

Listeinheit oder Druckerschnittstelle. Beispielsweise wird ein BASIC-Programm bei Verwendung des Befehls LIST über diesen Kanal als Liste ausgegeben. Da dem Amateur zum Zeitpunkt der Manuskripterstellung in der Regel noch kein Drucker zur Verfügung steht, wurde der Kanal in der Initialisierungsphase mit der Bildschirmausgabe verbunden.

Im IO-byte sind diese 4 logischen Geräte wie folgt angeordnet

bit 0,1 CON

bit 2,3 RDR

bit 4,5 PUN

bit 6,7 LST

und lassen sich den physischen Geräten TTY, CAS, PRN und USR zuordnen. Dabei bedeuten:

TTY Bildschirm bzw. Tastatur,

CAS Kassetteninterface (SID Port A),

PRN Druckerinterface (SIO Port B),

USR Anwenderschnittstelle. Es können in die entsprechenden RAM-Zellen Sprünge zu eigenen Gerätetreibern eingetragen werden.

Die Codierung der einzelnen physischen Geräte ist:

TTY 00B

CAS 01B

PRN 10B

USR 11B

Beispiel: Es soll ein Bedienerprotokoll erstellt werden, d. h., alle Ausgaben müssen auf dem Bildschirm und zusätzlich auf dem Drucker erscheinen. Sämtliche Dateieingaben übernimmt ein selbst realisierter Gerätetreiber. Die Datei- und Listausgabekanäle sollen ihre Standardzuordnung behalten. Daraus ergibt sich folgende Konfiguration: LST=PRN, PUN=CAS, RDR=USR und CON=PRN

Das IO-byte heißt im binären Format 10 01 11 10 und entspricht dem Hexwert 9EH.

Der Befehl IOBYTE zeigt das gerade aktu-

elle IO-byte in hexadezimaler Form an und wartet anschließend auf eine Eingabe. Bei einer Leereingabe, also wenn man nur RETURN drückt, wird das IO-byte unverändert wieder abgespeichert. Eine hexadezimale Eingabe verändert das IO-byte. Achtung: Die Konsoleingabe, d. h., die Tastatur kann nicht umgeleitet werden. Prinzipiell ist das zwar ohne weiteres möglich, aber in der Regel erstens nicht sinnvoll, und zweitens kann durch ein falsches IO-byte eventuell der Computer unbedienbar werden.

## 2.8.2. Das Programm

Tabelle 2.7. zeigt, wie das Monitorprogramm als Hex-Dump in die 2 notwendigen EPROMs U2716 gebrannt werden muß. Die Liste besteht aus Urladeprogramm, Monitorkommandos, Ein- und Ausgabeprogramm, Zeichengenerator sowie vorinitialisierten RAM-Zellen.

Das Monitorprogramm läuft nur im RAM!

Tabelle 2.7. Hexdump des Monitorprogramms

|       |   |
|-------|---|
| 0000: | 00 00 00 00 06 00 10 FE 21 16 00 11 00 F0 01 00 . . . . . ! . . . p.. =024D |
| 0010: | 10 ED B0 C3 00 F0 C3 21 F0 C3 5E F7 C3 4E F7 C3 .mOC.pC!pC^wCkWC =0A34      |
| 0020: | BE F7 C3 9B F7 C3 7F F7 C3 4F F7 C3 FE F6 C3 16 >wC.wC.wCwC^vC. =0BDC       |
| 0030: | F7 C3 A8 F6 C3 67 F0 31 6B FF D3 00 0E 84 CD 16 wC(vCgpk.S...M. =0955       |
| 0040: | F7 21 FF BF 22 A1 FE 23 54 5D CD 09 F0 CD 79 F1 w!.?!"!~WTJM.pMyq =0968     |
| 0050: | 0C 20 48 2E 4D 4F 4E 20 56 33 2E 30 20 31 30 2F . H.MON V3.0 10/ =0343      |
| 0060: | 38 35 0D 0A 00 CD 35 F7 18 13 CD 79 F1 49 4C 4C 85...M5w..MyqILL =05C0      |
| 0070: | 45 47 41 4C 20 43 4F 4D 40 41 4E 44 00 31 6B FF EGAL COMMAND.1k. =04D3      |
| 0080: | 0E 1B CD 09 F0 OE OC CD 09 F0 CD 86 F1 CD 5A F1 ..M.p..M.pM.qMZq =082B      |
| 0090: | 38 EB ED 53 4E FE 21 00 F0 01 FF FF 3E ED ED B1 8kmSN~!..p...>mml =0988     |
| 00A0: | 20 CB 3E FF BE 20 F5 ED 5B 4E FE 1A 23 13 0B BE H>.> um[N~.M..> =07A5       |
| 00B0: | 28 F9 FE 20 28 08 FE 2C 28 04 FE 0D 20 DE AF 2B (y~ (.~, (.~. ^/+ =06A8     |
| 00C0: | C5 01 08 00 ED B1 C1 20 D3 01 67 F0 C5 E9 CD 5A E...m1A S.gpeIMZ =084D      |
| 00D0: | F1 21 F7 FD 06 08 CD 5A F1 28 0C 77 23 13 10 F6 q!..).MZq(.w#..v =0713      |
| 00E0: | CD 5A F1 CB 13 18 F9 36 20 23 10 FB C9 CD 5A F1 MZqh..y6 M.(IMZq =0869      |
| 00F0: | 21 FF FD 06 02 FE 40 20 EE 13 18 DA CD 5A F1 21 !..~@ n..ZMZq! =07AF        |
| 0100: | 07 FE 06 10 FE 3B 20 DF 13 1A FE 0D 28 D9 77 23 .~..~! ..~..(Yw# =0626      |
| 0110: | 10 F6 C9 3E 20 CD B0 F7 CD 8D F1 CD 5A F1 CD 25 .vi) M0wM.qMZqM% =09F6      |
| 0120: | F1 1A D8 C5 21 00 00 1A CD 25 F1 38 0B 29 29 29 q.XE!...M%qB.)) =0584       |
| 0130: | 29 06 00 4F 09 13 18 EF C1 AF C9 D6 30 DB FE OA )..0..oA/IVOX. =06C0        |
| 0140: | 3F DO FE 11 D8 FE 17 3F DB D6 07 C9 F5 1F 1F ?P~.X~.?XV.Iu.. =081A          |
| 0150: | 1F CD 3F F1 F1 F5 E6 0F C6 30 FE 3A 38 02 C6 07 .M?qquf.FO~:8.F. =082C      |
| 0160: | CD B0 F7 F1 C9 F5 7C CD 36 F1 7D CD 36 F1 F1 C9 M0wqIu;M6q)M6qqI =0B8E      |
| 0170: | 1A FE 20 20 0D 13 1A FE 20 28 FA FE OD 4E 05 AF .~ .~.~ (z~.!.~ =05B9       |
| 0180: | C9 FE QD 37 C8 3F FE 2C 13 28 F3 B7 1B C9 E3 .I~.7H?~,.(s7..IC =0802        |
| 0190: | 7E 23 B7 28 05 CD B0 F7 18 F6 E3 C9 CD 79 F1 OD ~#7(.M0w.vcIMyq. =08F7      |
| 01A0: | 0A 26 00 11 50 FE 06 51 CD 03 F0 FE 06 28 ED FE .L..P~.QM.p~.(m~ =06BD      |
| 01B0: | 08 28 0F FE 0D 28 1B FE 20 38 ED CD 4E F7 12 13 .(.~.~.BmM0w.. =0669        |
| 01C0: | 10 E6 7B FE 51 28 E1 1B 04 CD 79 F1 08 20 08 00 .fx~Q(a..Myq. .. =064C      |
| 01D0: | 18 D6 12 11 50 FE 3E 0D CD B0 F7 3E 0A C3 B0 F7 .V..P~).M0w).C0W =07D0      |

01E0: ED FF 4D 45 4D 4F 52 59 00 CD 05 F1 38 03 22 01 m.MEMORY.M.q8.". =05E6  
 01F0: FE FE 51 C8 3E 0C CD B0 F7 3E 0A CD B0 F7 06 08 ~~@H>.M0w>.M0w.. =089D  
 0200: OE 20 AF CD 36 F1 3C F5 CD 09 F0 F1 10 F5 CD 79 . /M6q<uM.pq.uMy =0904  
 0210: F1 30 31 32 33 34 35 36 37 00 06 18 3E 2D CD B0 q01234567...>-MO =0493  
 0220: F7 10 FB 06 08 3E 08 CD B0 F7 3E 20 CD B0 F7 3E w.{..>.M0w> M0w> =07DA  
 0230: 2D CD B0 F7 10 FB CD CO F1 3E 48 32 8D F3 2A 01 -M0w.(M&q>H2.b\*. =08BD  
 0240: FE 22 03 FE 2A 03 FE 7D E6 F8 6F 22 03 FE EB CD ~".~\*.\*~}fxo~.^~km =08F1  
 0250: 79 F1 01 OA OA OA 00 06 10 C5 D5 06 08 1A 13 CD yq.....EU..M =0441  
 0260: 36 F1 3E 20 CD B0 F7 10 F4 06 08 D1 1A 13 E6 7F 6q> M0w.t..G..f. =076E  
 0270: FE 20 30 02 3E 2E CD B0 F7 10 F1 C1 10 DB CD 89 ~ O.>.M0w.qA.[M. =0833  
 0280: F3 CD 4C F3 3A 8D F3 67 FE 51 20 0D CD 79 F1 1B sMLs:.sg~Q .Myq. =08EE  
 0290: 93 81 00 CD FD F0 C3 D6 F1 5C 16 02 CD 03 F0 FE ...M>pCVq\..M.p~ =098A  
 02A0: 1B 20 0D 3E 48 BB 20 02 3E 41 32 8D F3 C3 68 F2 . >H1 ..>A2.sChr =05F9  
 02B0: 01 F8 FF FE 1A 28 69 0E FF FE 08 28 63 03 F3 FE .x.~.(1..~.(c..~ =0743  
 02C0: 15 28 5D 0E 08 FE 0A 28 57 FE 51 28 DD FE 20 D4 .{1..~.(W~Q[J]~ T =067D  
 02D0: B0 F7 6F 7B FE 41 7D 28 15 CD 25 F1 F5 DC 4C F3 Owó(~A)(.M%qu\Ls =097D  
 02E0: F1 38 9B 21 05 FE ED 6F 15 20 B1 3A 05 FE 2A 01 q8.!..~mo. 1:~\*. =0692  
 02F0: FE 77 BE 3A 8D F3 F5 3E 48 32 8D F3 CD 4C F3 7E ~w>:su>H2.sMLs~ =09A4  
 0300: CD 36 F1 3E 41 32 8D F3 CD 4C F3 7E E6 7F 2E 20 M6q>A2.sMLs~f.~ =0932  
 0310: 30 02 3E 2E CD B0 F7 F1 32 8D F3 20 33 01 01 00 O.>.M0wq2.s 3... =060A  
 0320: 2A 03 FE 11 80 00 19 EB 2A 01 FE 09 22 01 FE CB \*.~....k\*.~.".~K =05DE  
 0330: 78 28 05 EB 2A 03 FE 13 B7 ED 52 28 03 FA 68 F2 x(.k\*.~.^mR(.zhr =0743  
 0340: OD 20 02 0E 07 OC 2A 03 FE 09 22 03 FE C3 2E F2 . ....\*.~.^..~C.r =04BA  
 0350: CD 79 F1 OC 52 41 4D 20 45 52 52 4F 52 20 00 C3 Myq.RAM ERROR .C =05B0  
 0360: 2E F2 D9 2A 01 FE ED 4B 03 FE 87 ED 42 3E 1B CD .rY\*.~mk.~^mB).M =0867  
 0370: B0 F7 7D B7 1F 1F 37 1F C6 03 E6 9F CD B0 F7 7D Ow>J..7.F.f.M0w> =08AE  
 0380: E6 07 47 3A 8D F3 FE 41 28 10 AF 05 04 28 04 C6 f.G.:s~A(/.//(.F =060F  
 0390: 03. 10 FC F6 80 CD B0 F7 D9 C9 3E 18 80 18 F4 CD ..iv.M0wYI>..tM =094A  
 03A0: 79 F1 01 48 20 20 43 55 52 53 4F 52 3A 00 2A 01 yq.H CURSOR:.\*. =0436  
 03B0: FE C3 4F F1 ED FF 4D 4F 56 45 00 CD 05 F1 DA 54 ~C0qm.MOVE.M.qZT =0915  
 03C0: F0 22 01 FE CD 05 F1 DA 54 F0 22 03 FE CD 05 F1 p~.^~M.qZTp~.^~M.q =08D8  
 03D0: DA 54 F0 44 4D 2A 01 FE ED 5B 03 FE E5 AF IED ZTpDM~.^~m[.^~e/km =098D  
 03E0: 52 23 5D 59 44 4D E1 E5 B7 ED 52 E1 38 03 ED BO R#PYDMae7mRa8.m0 =0824  
 03F0: C9 0B 09 EB 09 EB 03 ED BB C9 ED FF 4C 4F 41 44 I..k.k.m8l.m.LOAD =0839  
 0400: 00 CD DB F0 CD D7 F0 CD 05 F1 30 03 21 00 00 E5 .MBpMWpM.q0!...e =0805  
 0410: FD E1 CD 35 F7 2A FF FD 1E 00 D9 16 BD CB 7A CA 3aM5W~.).Y..KzJ =0999  
 0420: OA F5 D9 16 00 D9 CD 2B F5 FE 1E 20 03 CD 6B F7 .uY..Ym+u~. .Mkw =0822  
 0430: FE 01 20 F2 AF 4F 5F 21 F7 FD 06 08 CD 6B F7 BE ~. r/O\_!w)..Mkw> =087E  
 0440: 20 DB 23 81 4F 10 F5 06 18 CB 7A 20 02 CB F2 CD [#.O.u..Kz .KrM =0702  
 0450: 39 F5 28 04 CB 7A 20 C3 CB 7A 2B 2A CD 79 F1 0D 9u(.Kz CKz(\*Myq. =075D  
 0460: 0A 46 49 4C 45 20 46 4F 55 4E 44 20 00 FD E5 C1 .FILE FOUND .)~e =0589  
 0470: 2A 01 FE 09 22 01 FE 2A 03 FE 09 22 03 FE 78 B1 \*.~.".~\*~.".~x1 =05D3  
 0480: 28 04 AF 32 FF FD DU 21 07 FE 2A 01 FE CD 2B F5 (.//.3)1..~\*.~M+u =0822  
 0490: FE 04 CA OA F5 FE 03 28 SE FE 1D 20 F0 OE 00 CD ~.J.u~.(^~. p..M =0758  
 04A0: 6B F7 47 4F CB AA CB 7A 20 26 CB F2 CB 62 20 20 kwGOK\*Kz &KrKb =0822  
 04B0: D5 7A E6 0F DD 56 01 BA D1 20 15 DD 7E 00 III 20 Uzf.JV.:G .J~.; =076E  
 04C0: OF DD 23 DD 23 D9 15 D9 20 02 CB E2 CB B2 CB EA .JWJWY.Y .Kb2Kj =08D7  
 04D0: CD 39 F5 28 1F CB 6A 20 47 CB 7A 28 17 D9 14 7A M9u(.Kj GKz(.Y.z =06C9  
 04E0: D9 FE 08 CA 05 F4 IIU 72 01 DD CB 01 IED 73 00 Y~.J.tJr.JK.>Js. =08A9  
 04F0: DD 23 DD 23 13 18 96 7A E6 80 57 IED 4B 03 FE 1#J#.~zf.W+mK.~ =075C  
 0500: ED 42 C2 07 F4 D9 7A B7 28 05 D9 55 C3 10 F4 CD mBB.tYz7(.YUC.tM =08E5  
 0510: 19 F5 3A FF FD FE 50 CO 7D FE 51 C8 2A 05 FE E9 .u.:.)~PQ~)~QH~.~i =09FC  
 0520: D9 CD 79 F1 20 42 41 44 20 46 49 4C 45 20 00 7B YMqg BAD FILE .{ =05D2  
 0530: B7 CB CD 36 F1 CD 79 F1 20 45 52 52 4F 52 53 00 7HM6qMyq ERRORS. =07A7  
 0540: C9 CD 6B F7 3C 20 FA CD 6B F7 FE FF 2B F9 C9 CD IMku< zMkw~.(yIM =0B31  
 0550: 6B F7 CB 72 20 01 77 81 4F 23 10 F3 CD 6B F7 81 kwKr .w.O..sMkw. =07DD  
 0560: CB B2 CB D9 1C D9 C9 ED FD 53 41 56 45 00 CD BB K2HY.Yim.SAVE.M8 =097C  
 0570: FO CD D7 FO CD 05 F1 DA 54 F0 22 01 FE CD 05 F1 pMWpM.qZTp~.^~M.q =0A49  
 0580: DA 54 F0 22 03 FE CD 05 F1 30 03 21 1E FO 22 05 ZTp~.^~M.q0!.p~. =068D  
 0590: FE CD E6 F0 CD 35 F7 21 07 F6 E5 21 98 F5 E5 E5 ~MfpM5W!.ve!.uee =0B15  
 05A0: OE 1E 3A FF FD FE 20 20 05 3E 50 32 FF FD 06 14 ...J~. >P2.).~. =067B  
 05B0: 3E FF CD 94 F7 10 F9 79 CD 94 F7 3E 01 CD 94 F7 >.M.w.yyM.w>.M.w =0A06  
 05C0: 21 F7 FD 06 20 0E 00 7E CD 94 F7 7E 81 4F 23 10 !w). ..~M.w~.O.. =06A0  
 05D0: F6 ED 44 CD 94 F7 3E 02 CD 94 F7 ED 5B 01 FE 06 vmdM.w>.M.wm[.~. =0964

05E0: 06 3E FF CD 94 F7 10 F9 2A 03 FE 23 ED 52 28 26 .>.M.w.y.~#mR (& =077F  
 05F0: AF B4 20 04 85 F2 E4 F5 2E 80 45 3E 1D CD 94 F7 /4 ..rdv.~.E>.M.w =087D  
 0600: 78 4F CD 94 F7 1A CD 94 F7 1A 13 81 4F 10 F6 ED xDM.w.M.w...D.vm =0881  
 0610: 44 CD 94 F7 18 C9 3E 03 0E FF C3 94 F7 06 20 3E DM.w.I>...C.w. > =077D  
 0620: FF CD 94 F7 10 F9 3E 04 C3 94 F7 ED FF 49 4E 00 .M.w.y>.C.wm.IN. =0973  
 0630: CD 05 F1 4D ED 78 C3 36 F1 ED FF 4F 55 54 00 CD M.qMMxC6qm.QUT.M =0910  
 0640: 05 F1 4D CD 05 F1 ED 69 C9 ED FF 46 49 4C 4C 00 .qMM.qmiIm.FILL. =0838  
 0650: CD 05 F1 DA 54 FO 22 01 FE CD 05 F1 DA 54 FO 22 M.qzTp".~M.qzTp" =0905  
 0660: 03 FE CD 05 F1 4D 2A 01 FE ED 5B 03 FE 30 02 0E .~M.qM\*.~mf.~O.. =06C3  
 0670: 00 71 E5 AF EB ED 52 44 4D E1 54 5D 13 ED B0 C9 .qe/kmRDMAIJ.mOI =08CB  
 0680: ED FF 53 49 5A 45 00 CD 79 F1 52 41 4D 54 4F 50 m.SIZE.MyqRAMTOP =0731  
 0690: 3A 00 2A A1 FE CD 4F F1 CD FD FO 38 03 22 A1 FE :.\*~MOqM)p8."!~ =0BC6  
 06A0: CD 79 F1 43 52 54 42 47 4E 3A 00 2A 37 FE CD 4F FO MyqCRTBGN:.\*~MO =06AC  
 06B0: F1 CD FD FO D8 22 37 FE 54 5D CD 43 F9 C9 3A A2 qMjpx"7~TlMCyI:" =0A39  
 06C0: FE 47 3A A1 FE C9 ED FF 47 4F 00 1B CD 05 F1 DA ~G!:~Im.G0..M.qz =0921  
 06D0: 54 FO 01 67 FO C5 E9 ED FF 48 45 4C 50 00 21 00 Tp.gpEim.HELP.! =0780  
 06E0: 00 01 00 00 CD CO F1 CD 79 F1 20 20 20 00 3E ED ...M@qMyq .~m =0641  
 06F0: ED B1 CO 3E FF BE 20 F6 E5 C5 06 10 7E 23 B7 28 m1@>.> veE.~#2! =08AF  
 0700: 06 10 F9 C1 E1 18 E7 C1 E1 23 08 B7 28 D8 CD ...yAa.gAa#.~?!(XM =0882  
 0710: B0 F7 18 F5 3A 3D FE C9 ED FF 49 4F 42 59 54 45 Ow.u:=?Im.I0BYTE =08AA  
 0720: 00 CD FE F6 CD 36 F1 CD FD FO D8 4D F5 79 32 3D .M~vM6qMjpxMuyZ= =0A71  
 0730: FE F1 C9 DE 08 FE 1A C8 DB 19 E6 01 28 F5 DB 18 ~qlf.~.Hl.f.(ul. =0966  
 0740: C9 79 D3 18 DB 19 E6 04 28 FA C9 0E 19 06 08 21 IyS.[.f.(zI....! =064C  
 0750: 47 F7 ED B3 3E 05 D3 16 3E 05 D3 16 C9 04 04 01 Gwm3>.S.,.S.I... =0608  
 0760: 40 03 C1 05 EA DE 09 B7 20 04 32 5D F7 C9 3A 5D @.A.j{.7 .2JwI:j =0698  
 0770: F7 3D C9 00 CD 4F F7 28 FB 3E 01 32 5D F7 DB 08 w=I.MUW({).2Jwl. =0JDB  
 0780: C9 3A 3D FE E6 0C 28 EC FE 04 CA 1D F7 FE OC CA I:=~t. (1~.J.w~.J =0SF6  
 0790: 44 FE 3E 1A C9 3A 3D FE E6 CO CA BE F7 FE 40 CA D~).I:=~t@J>w~@J =0A05  
 07A0: F1 F7 FE DE CA 3E FE C3 4A FE C5 4F CD 9B F7 C1 .w~.J>~CJ>EOM.WA =0B55  
 07B0: C9 3A 3D FE E6 30 CA BE F7 FE 10 CA 2B F7 FE 20 I:=~foJ>w~.J+~w~ =09EB  
 07C0: CA 7F F7 C3 47 FE C5 4F 32 BD F7 CD BE F7 3A DE J.wCG>E02=wMw:= =0ABB  
 07D0: F7 C1 C9 00 3A 3D FE E6 03 CA D3 F7 FE 01 CA 9B wAI.:=~f.JSw~.J. =09D7  
 07E0: F7 FE 02 CA 7F F7 C3 41 FE D5 C5 32 3C FE DD E5 w~.J.wCA>UE2<~le =0B01  
 07F0: E5 C3 OB F9 36 00 79 21 3B FE CB 46 20 08 FE 1B eC.y6.y!~KF .~. =070A  
 0800: C2 76 F8 CB C6 CB 8E 18 31 CB 7F 28 40 CB 4E CB BvxKFK..1K. (GKNK =0BF9  
 0810: CE CB BF 28 OC CB 86 06 00 CB B9 2A 39 FE 09 18 NK?(.K..K9\*~.~. =06E9  
 0820: 0D 21 00 00 B7 28 07 47 11 40 01 19 10 FD 22 39 .!.7(G.~.~.)"9 =032E  
 0830: FE 11 20 01 19 36 F8 22 DC F7 E1 DE E1 C1 D1 3A ~. .6x"\walaAQ: =08D7  
 0840: 3C FE C9 D9 08 F1 C1 D1 E1 D9 08 18 ED CB DE E1 ~.qAgAaY..mK.a =0A60  
 0850: E5 D9 08 E5 D5 C5 F5 21 2D F8 E5 D9 E5 21 55 F8 eY.eUEu!-xeYe!Ux =0A91  
 0860: 06 00 CB 21 09 FE 23 66 6F E3 C9 19 F9 19 F9 5F ..K!.~#focI.y.y\_ =069E  
 0870: F9 7D F9 D4 F9 EE F9 F1 F9 E5 F9 E8 F9 43 F9 75 y>yTyuqyeyhyCyu =0D7D  
 0880: F8 39 F9 2C F9 75 F8 75 F8 75 F8 C9 21 18 F8 E5 x9,y,yuxuxuxI!.xe =0A75  
 0890: E6 7F FE 20 38 2D 6F 26 00 29 29 11 F7 F9 19 f.~ B-o&.).)).wy. =0612  
 08A0: DD 2A 39 FE DD E5 11 20 00 06 08 7E DD 77 00 23 J\*9~Je. ...~Jw.# =0634  
 08B0: DD 19 10 F7 E1 23 7D E6 1F CO 2B CD DO FB 3E OA J..wA#f.@+MPx. =084B  
 08C0: 22 39 FE 2A 39 FE 11 40 01 FE OD 28 19 FE 04 28 "9~\*9~.Q.~.~.~. =0588  
 08D0: 1A FE 08 DE 37 FE OC 28 54 FE 01 28 61 FE 1A 28 .~. (7~.(T~.(a~.~. =05CD  
 08E0: 35 FE 15 DE DO C9 7D E6 EO 6F C9 19 7C E6 1F FE 5~.(PI)f'oi.lf.~ =091C  
 08F0: 1F DE ED 52 E5 21 00 00 E5 19 D1 01 00 1E EO BO .XmRe!..e.Q...m0 =06C7  
 0900: 62 6B 13 01 00 02 36 00 ED B0 E1 C9 7D E6 1F 2B bK... .6.m0aIjf.+ =060D  
 0910: CO 23 7D F6 1F 6F B7 ED 52 7C E6 EO FE 00 DO 18 #M)v.o7mRif!"P. =0902  
 0920: D1 01 18 F8 C5 CD 43 F9 3E 21 32 DE F7 AF 01 3E ...xEMCy.~?I2[w/.> =074D  
 0930: FF 47 2A 37 FE 70 54 5D 13 01 00 20 ED B0 2A OC .G#7~pTJ... mOK. =05CD  
 0940: F8 C9 21 36 00 22 DE F7 21 36 F8 22 1F F8 C9 21 xI!6."~w!6x".xI! =0781  
 0950: 00 00 22 DE F7 22 1F F8 C9 7C E6 EO 67 2E 00 22 ..~w~.xI!f'g.. =06F2  
 0960: 37 FE 22 39 FE 22 OC F8 22 EO F8 32 77 F9 32 07 7~"9~".x"~x2wy2. =0789  
 0970: F9 7A D3 10 C9 7D CB 3C CB 1D CB 3C CB 1D CB 3C yz.S.I)K<K.K<K.K< =0881  
 0980: CB 1D E6 07 07 07 F6 C6 32 7B F9 3E 80 B4 67 K.f.~.vF2(y).4g =0725  
 0990: CB C6 C9 EB 7B 95 06 2C 30 04 06 2D ED 44 F5 78 KFIk{.,.O...-mDux =078C  
 09A0: 32 C4 F9 7A 94 06 24 0E FF 28 08 OC D2 9D F9 06 2Dyz...~.~.~.R.y. =06DE  
 09B0: 25 ED 44 F5 78 32 CC F9 79 C6 00 D9 67 6F 08 F1 %mDux2LyyF.Ygo.q =08A1  
 09C0: 5F F1 4F 16 00 42 08 08 D9 E5 CD 5F F9 E1 7D BB \_q0..B..YeM.ya); =0803  
 09D0: C2 CO F9 7C BA C8 08 F2 CC F9 2C D9 B7 ED 5A C3 B@y:H.rLy,Y7mZC =0AFAE

09E0: B1 F9 24 D9 B7 ED 42 C3 B1 F9 3A 72 F9 32 E3 F9 1y\$Y7mBC1y!ry2cy =0AAD  
 09F0: 3E 46 32 72 F9 CD 5F F9 3E 00 01 3E 86 01 3E C6 >F2ryM\_y>...>F =064E  
 0A00: 32 72 F9 C9 3E C9 01 3E 7D 32 5F F9 C9 00 00 00 2ryI>I.>32\_yI... =067C  
 0A10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0A20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0A30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0A40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0A50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0A60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0A70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0A80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0A90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0AA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0AB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0AC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0AD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0AE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0AF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0B00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000  
 0B10: 00 00 00 00 00 20 20 20 20 20 00 20 00 50 50 50 ..... .PPP =01B0  
 0B20: 00 00 00 00 00 50 50 F8 50 F8 50 50 00 20 F0 28 ....PPxPxPP. p( =04B8  
 0B30: 70 A0 78 20 00 18 98 40 20 10 C8 C0 00 30 48 28 p x ...@ .H@.OH( =04F0  
 0B40: 10 A8 48 B0 00 60 60 40 20 00 00 00 40 20 10 .(HO.'@ ...@. =0340  
 0B50: 10 10 20 40 00 10 20 40 40 20 10 00 20 AB 70 .. @.. @@@ .. (p =02D8  
 0B60: F8 70 A8 20 00 00 20 20 F8 20 20 00 00 00 00 00 xp{ .. x .. =03A8  
 0B70: 00 60 60 40 20 00 00 00 F8 00 00 00 00 00 00 00 00 .''@ ..x..... =0218  
 0B80: 00 00 30 30 00 00 80 40 20 10 08 00 00 70 88 88 ..00...@ ..p.. =02D8  
 0B90: A8 88 88 70 00 20 30 20 20 20 70 00 70 88 80 (...p. O p.p.. =04E0  
 0BA0: 60 10 08 F8 00 70 88 80 70 88 70 00 40 60 50 '...x.p..p..p.@'P =05C0  
 0BB0: 48 FC 40 40 00 F8 08 08 78 80 88 70 00 60 10 08 H;@.x..x..p. ' =0534  
 0BC0: 78 88 88 70 00 F8 80 40 20 10 08 08 00 70 88 88 x..p..x.@ ..p.. =0570  
 0BD0: 70 88 88 70 00 70 88 88 F0 80 40 30 00 00 30 30 p..p..p..p.@@.00 =05B0  
 0BE0: 00 30 30 00 00 00 30 30 00 30 30 20 10 40 20 10 .00...00.00 .@. =01C0  
 0BF0: 08 10 20 40 00 00 00 F8 00 F8 00 00 00 10 20 40 .. @...x.x.... @ =02D8  
 0C00: 80 40 20 10 00 70 88 80 40 20 00 20 00 70 88 80 ..@..p..@ ..p.. =0460  
 0C10: B0 A8 A8 70 00 20 50 88 88 F8 88 88 00 78 88 88 0((p. M..x..x.. =0780  
 0C20: 78 88 88 78 00 70 88 08 08 08 88 70 00 78 88 88 x..x..p.....p..x.. =0590  
 0C30: 88 88 88 78 00 F8 08 08 78 08 08 F8 00 F8 08 08 ...x..x..x..x..x..x.. =05A0  
 0C40: 78 08 08 08 00 70 88 08 E8 88 88 F0 00 88 88 88 x....p..h..p.... =0610  
 0C50: F8 88 88 88 00 70 20 20 20 20 70 00 80 80 80 x....P P.... =0590  
 0C60: 80 80 88 70 00 88 48 28 18 28 48 88 00 08 08 08 08 ..p..H(.H.... =0418  
 0C70: 08 08 08 F8 00 88 D8 A8 A8 88 88 88 00 88 98 A8 ...x..X((.....( =0720  
 0C80: C8 88 88 88 00 70 88 88 88 88 88 70 00 78 88 88 H....p.....p..x.. =0770  
 0C90: 78 08 08 08 00 70 88 88 88 A8 48 B0 00 78 88 88 x....p..(HO.x.. =05C0  
 0CA0: 78 28 48 88 00 70 88 08 70 80 88 70 00 F8 20 20 x(H..p..p..p..x =0590  
 0CB0: 20 20 20 20 00 88 88 88 88 88 88 70 00 88 88 88 .....p.... =05B8  
 0CC0: 88 88 50 20 00 88 88 88 A8 A8 D8 88 00 88 88 50 ..P ....((X..P =0728  
 0CD0: 20 50 88 88 00 88 88 88 88 88 50 20 20 00 F8 80 40 P.....P ..x.@ =0580  
 0CE0: 20 10 08 F8 00 70 10 10 10 10 70 00 00 08 10 ..x.p.....p.... =0278  
 0CF0: 20 40 80 00 00 70 40 40 40 40 40 70 00 20 70 A8 @...p@eee@p. p( =0438  
 0D00: 20 20 20 20 00 00 00 00 00 00 00 F8 00 30 30 10 .....x.00. =01E8  
 0D10: 20 00 00 00 00 00 30 40 78 48 F8 00 10 10 70 .....O@xHx..p =02D8  
 0D20: 90 90 90 78 00 00 60 90 10 10 E0 00 40 40 70 .....x...'.@@p =0508  
 0D30: 48 48 48 F0 00 00 00 60 90 70 10 E0 00 00 C0 20 HHhp...'.p..@ =04F8  
 0D40: 70 20 20 20 00 00 00 E0 90 90 E0 80 F0 10 10 70 p .....'.p..p =05B0  
 0D50: 90 90 90 90 00 00 20 00 20 20 20 20 00 00 40 00 ..... . .@. =0320  
 0D60: 40 40 40 50 20 00 10 90 50 30 50 90 00 00 10 20 @@@P ..POP.... =0360  
 0D70: 20 20 20 60 00 00 00 58 A8 A8 A8 88 00 00 70 '...X(((.p.... =0408  
 0D80: 90 90 90 90 00 00 60 90 90 90 60 00 00 00 70 .....'.p..p =0520  
 0D90: 90 90 70 10 10 00 00 70 48 48 70 40 C0 00 00 D0 ..p....pHHhp@..P =04F0  
 0DAO: 30 10 10 10 00 00 00 E0 10 60 80 70 00 00 20 70 0.....'.p.. p =0330  
 0DB0: 20 20 A0 40 00 00 00 48 48 48 48 F0 00 00 00 88 @...HHHhp.... =03B8  
 0DC0: 88 50 50 20 00 00 00 88 88 A8 A8 50 00 00 00 88 .PP .....((P.... =0480  
 0DD0: 50 20 50 88 00 00 00 90 90 A0 C0 80 E0 00 00 F8 P P..... @.'..x =0620

### **2.8.3. Programmtechnische Details**

Das Monitorprogramm belegt den Speicherbereich von 0F000H bis 0FFFFH. Nach Reset wird das Programm aus den beiden EPROMs in den RAM dupliziert. Dazu benutzt man eine kleine Befehlsfolge, die dem Listing in Bild 2.53 zu entnehmen ist.

Eine Zeitschleife soll nach dem Einschalten des Computers vor allen Dingen den dynamischen RAMs genügend Zeit zum Formatieren geben. Nach Anlegen der Speisespannungen brauchen dynamische RAM-Bausteine einige Zyklen zum Aufladen der internen Kapazitäten. Die Zeitschleife sichert ab, daß die Speicher voll funktionsfähig sind.

Das Monitorprogramm ist ab Adresse 16H in den beiden EPROMs abgelegt. Es wird komplett mit dem LDIR-Befehl auf die Adresse 0F000H umgeladen und anschließend angesprungen. Nun kann der ROM-Bereich ausgeblendet werden, denn man benötigt ihn ja nicht mehr. Das Umschalten vom ROM auf den vollen 64-Kbyte-RAM-Bereich bewirkt ein beliebiger Ein- oder Ausgabebebefehl, der die Portadresse 00H (ROMOF) anspricht. Den ROM kann man auch wieder in den Speicherbereich legen. Dazu muß die Portadresse 04H angesprochen werden. Im wesentlichen entspricht der Aufbau des Monitorprogramms dem schon in [2] veröffentlichten Programm. Speziell für den hier beschriebenen Computer wurden jedoch einige Programmteile ganz gestrichen, andere optimiert. Völlig verändert sind die Routinen zur Tastaturabfrage und die Steuerung der Bildschirmausgabe. Da die Bildschirmausgabe grafisch orientiert ist, mußte der Zeichengenerator als Teil des Monitorprogramms aufgenommen werden. Dieser Zeichengenerator benötigt etwa 1 Kbyte Platz,

was zwangsläufig auf Kosten des Kommandoumfangs des Monitors gehen mußte.

Leider findet man das gesamte Programm aus Platzgründen nur als Hex-Listing in diesem Beitrag. Die für den Anwender wichtigsten Programmteile sollen aber ausschnittsweise als Assemblerlisting dargestellt und beschrieben werden.

Das Programm beginnt, wie schon der Monitor in [2], mit einer Sprungtabelle (Bild 2.53). Sie wurde gegenüber [2] nur geringfügig verändert. Diese Tabelle enthält die wichtigsten Systemeinsprünge. Da sie sich vom Anwender uneingeschränkt benutzen lassen, sind sie im folgenden detailliert beschrieben:

JP BEGIN

Das ist die Kaltstartadresse des Monitors. Durch sie wird das gesamte System initialisiert, einschließlich aller Ports. Der Zeichengenerator bleibt jedoch unverändert, so daß selbstdefinierte Zeichen erhalten bleiben. Nur ein Reset überschreibt den Zeichensatz mit den fest im EPROM abgespeicherten Mustern. Nach einem Kaltstart wird der Bildschirm gelöscht und eine Meldung ausgegeben. Wenn dann das Promptzeichen (&) erscheint, ist der Computer eingabebereit.

JP CI

CI liest ein Zeichen von der Tastatur. Das Zeichen steht nach Rückkehr aus CI im Register A zur Verfügung. Alle anderen Register bleiben unverändert. War bei Aufruf von CI keine Taste gedrückt, was CSTS durch Aufruf erkennt, verharrt das Programm in einer Abfrageschleife. Diese Schleife und damit CI wird erst dann verlassen, wenn ein Tastendruck erkannt worden ist. Das IO-byte wirkt nicht auf die Tastatur.

JP RI

RI steht eigentlich für Reader-Input, also Lochbandleser. Mit diesem Sprung wird im

|      |         |       |             |
|------|---------|-------|-------------|
| 0000 | 00      | MONI: | NOP         |
| 0001 | 00      |       | NOP         |
| 0002 | 00      |       | NOP         |
| 0003 | 00      |       | NOP         |
| 0004 | 06 00   |       | LD B,0      |
| 0006 | 10 FE   |       | DJNZ *      |
| 0008 | 21 0016 |       | LD HL,PGM   |
| 000B | 11 F000 |       | LD DE,START |
| 000E | 01 1000 |       | LD BC,4096  |
| 0011 | ED B0   |       | LDIR        |
| 0013 | C3 F000 |       | JP START    |
| 0016 |         | PGM:  |             |

.PHASE 0F000H

I SPRUNGTABLELLE:

|      |         |        |           |                      |
|------|---------|--------|-----------|----------------------|
| F000 | C3 F021 | START: | JP BEGIN  |                      |
| F003 | C3 F75E | CI:    | JP CIX    | I TASTATUR           |
| F006 | C3 F76B | RI:    | JP RIX    | I CASSETTE LESEN     |
| F009 | C3 F7BE | CO:    | JP COX    | I BILDSCHIRM         |
| F00C | C3 F79B | POO:   | JP POX    | I CASSETTE SCHREIBEN |
| F00F | C3 F77F | LO:    | JP LOX    | I DRUCKER            |
| F012 | C3 F74F |        | JP CSTS   | I KONSOLSTATUS       |
| F015 | C3 F6FE |        | JP IOBYTE |                      |
| F018 | C3 F716 |        | JP IOSET  |                      |
| F01B | C3 F6A8 | MEMCK: | JP MEMSI  | I SPEICHERBEREICH    |
| F01E | C3 F067 | RSTAR: | JP MAIN   | I RESTART MONITOR    |

|      |             |        |                                      |                             |
|------|-------------|--------|--------------------------------------|-----------------------------|
| F021 | 31 FF6B     | BEGIN: | LD SP,STACK                          |                             |
| F024 | D3 00       |        | OUT (ROMOF),A                        |                             |
| F026 | 0E 84       |        | LD C,IOBYT                           |                             |
| F028 | CD F716     |        | CALL IOSET                           |                             |
| F02B | 21 BFFF     |        | LD HL,CRTBG-1                        |                             |
| F02E | 22 FEA1     |        | LD (MSIZE),HL                        |                             |
| F031 | 23          |        | INC HL                               | I BILDSCHIRM INITIALISIEREN |
| F032 | 54          |        | LD D,H                               |                             |
| F033 | 5D          |        | LD E,L                               |                             |
| F034 | CD F009     |        | CALL CO                              |                             |
| F037 | CD F179     |        | CALL PRINT                           | I MELDUNG AUSGEBEN          |
| F03A | 0C 20 48 2E |        | DEFB CLS,' H.MON V3.0 10/85',CR,LF,0 |                             |
| F03E | 4D 4F 4E 20 |        |                                      |                             |
| F042 | 56 33 2E 30 |        |                                      |                             |
| F046 | 20 31 30 2F |        |                                      |                             |
| F04A | 38 35 0D 0A |        |                                      |                             |
| F04E | 00          |        |                                      |                             |
| F04F | CD F735     |        | CALL CINIT                           | I CASSETTE INITIALISIEREN   |
| F052 | 18 13       |        | JR MAIN                              |                             |

Bild 2.53  
Anfang des Monitorprogramms

System ein Zeichen vom Kassettenport (SIO Port A) geholt. Über das IO-byte gesteuert, ist das Zeichen auch an einer anderen Eingabeschnittstelle greifbar. Das Zeichen wird, wie es schon in CI der Fall war, im Regi-

ster A übergeben. Alle anderen Register bleiben auch hier unverändert.

JP CO

CO ist die Consolausgabe, d. h. die Ausgabe auf den Bildschirm. Über diese Schnittstelle wird die gesamte alphanumerische und grafische Bildschirmsteuerung realisiert. Das Übergaberegister für das auszugebende Zeichen ist das Register C. In diesem Register erwartet CO das Zeichen. Alle Register blei-

ben erhalten. Auch die Consolausgabe läßt sich über das IO-byte umlenken. Vorsicht: durch eine fehlerhafte Veränderung des IO-byte könnte eine Bedienung des Computers in Frage gestellt werden. Beispielsweise, wenn die Bildschirmausgaben auf einem nicht vorhandenen Drucker erfolgen.

Zum Bildaufbau sind eine Reihe von Steuerzeichen vorhanden, sie werden detailliert im Abschnitt 2.8.4. beschrieben. Hier nur die Übersicht:

|             |   |
|-------------|---|
| <b>CR</b>   | 0DH Carriage Return, Wagenrücklauf.                       |
| <b>LF</b>   | 0AH Line Feed, Zeilenvorschub.                            |
| <b>BS</b>   | 0SH Backspace.  |
| <b>HOME</b> | 01H Der Cursor wird an den Bildschirmanfang positioniert. |
| <b>RIGH</b> | 15H Der Cursor bewegt sich nach rechts.                   |
| <b>UP</b>   | 1AH Der Cursor bewegt sich um eine Zeile nach oben.       |
| <b>CLS</b>  | 0CH Löscht den Bildschirm, anschließend HOME.             |
| <b>ESC</b>  | 1EH Escape, Umschaltung auf diverse Grafikfunktionen.     |

Alle anderen Codes von 20H bis 7FH sind darstellbare Zeichen und werden entsprechend angezeigt.

#### **JP PO**

PO bedeutet eigentlich Punch Out. Gemeint ist damit der Lochbandstanzer. Dieser Sprung ruft normalerweise die Kassettenausgabe (SIO Port A) auf. Wieder wird das auszugebende Zeichen im Register C erwartet und steht nach der Ausgabe auch im Akkumulator. Alle anderen Register bleiben unverändert. Das IO-byte kann zur Umleitung auf ein anderes Ausgabegerät verändert werden.

#### **JP LO**

Die Druckerschnittstelle wird im beschriebenen Computer durch eine genormte V24-Schnittstelle realisiert. Damit ist ein problemloser Druckeranschluß möglich. Wenn kein Drucker angeschlossen werden kann oder soll, empfiehlt es sich, das IO-byte so zu modifizieren, daß die Listausgabe auf den Bildschirm geführt wird. Wie auch schon bei den anderen Ausgabefunktionen wird das Zeichen im Register C erwartet und zusätzlich im Akkumulator wieder zurückgegeben.

#### **JP CSTS**

CSTS steht für Consol Status. Durch Aufruf dieser Routine läßt sich feststellen, ob eine Taste gedrückt wurde. CSTS liefert im Register A den Wert 0, wenn kein Zeichen von der Tastatur geliefert werden kann. Andererseits enthält der Akkumulator das byte OFFH, wenn ein Zeichen zur Verfügung steht. In beiden Fällen sind die entsprechenden Flags gesetzt. Viele Programme, wie auch der noch zu beschreibende BASIC-Interpreter, rufen während ihrer Programmausführung diese Routine auf, um z. B. festzustellen, ob eine Programmunterbrechung vorliegt. Das folgende kleine Beispiel soll dies verdeutlichen:

**SCHLEIFE:** hier steht das Anwendungsprogramm

|                 |                                  |
|-----------------|----------------------------------|
| CALL CSTS       | ;Taste gedrückt?                 |
| JP Z, SCHLEIFE  | ;nein, Programm weiter ausführen |
| CALL CI         | ;welche Taste?                   |
| CP 'Q'          | ;wurde 'Q' gedrückt?             |
| JP NZ, SCHLEIFE | ;nein, weiter                    |
| JP END          | ja, abbrechen                    |

In diesem Beispiel wird das ab Marke SCHLEIFE stehende Anwendungsprogramm so lange durchlaufen, bis die Taste Q erkannt wird.

#### **JP IOBYTE**

Dieses Unterprogramm liefert im Akkumulator das derzeit gültige IO-byte zurück. Das IO-byte kann mit dem folgenden Aufruf modifiziert werden und wurde bereits weiter oben beschrieben (Monitorkommando IO-BYTE).

#### **JP IOSET**

Über diesen Einsprung läßt sich die Zuordnung von logischen zu physischen Geräten einstellen. Das IO-byte muß im Register C übergeben werden. Natürlich kann man das IO-byte auch über das entsprechende Monitorkommando verändern. Mit Hilfe der Einsprünge JP IOBYTE und JP IOSET können auch Anwendungsprogramme Veränderungen in der Zuordnung von logischen zu physischen Geräten vornehmen.

#### **JP MEMCK**

Durch Aufruf der Routine MEMCK läßt sich mit einem Anwendungsprogramm die höchste zur Verfügung stehende freie RAM-Zelle erfragen. Dabei wird der Low-Teil die-

ser Adresse im Register A, der High-Teil im Register B zurückgegeben. Beispielsweise ruft der BASIC-Interpreter in seiner Initialisierungsphase dieses Unterprogramm.

#### JP RESTART

Das ist die Warmstartadresse des Monitors. Über diesen Einsprung kann zum Monitor zurückgekehrt werden. Selbstverständlich könnte man auch über den Kaltstarteinsprung auf der Adresse 0F00H zum Monitor zurückkehren, aber im Gegensatz zum Kaltstart bleibt der Systemzustand erhalten, d.h., daß z. B. eine für ein anderes Übertragungsformat umprogrammierte SIO nicht neu initialisiert wird. Erst beim Aufruf der Monitorkommandos SAVE oder LOAD wird die SIO entsprechend programmiert. Damit soll verhindert werden, daß der Rechner bei dem Versuch, Daten zu lesen, eventuell «hängen»

bleibt oder Daten in einem falschen Format aufzeichnet (was zweifelslos unangenehmer ist). Die so überflüssig erscheinende Initialisierung der Ports in der Kaltstartphase sichert eine sofortige Betriebsbereitschaft der Ein- und Ausgabebaugruppen ab, wenn beispielsweise Ein- oder Ausgaben durchgeführt werden, ohne daß vorher SAVE oder LOAD benutzt wurden.

Wie bereits mehrfach erwähnt, basiert das Programm auf dem schon in [2] vorgestellten Monitor. Stark verändert wurde vor allen Dingen der MEMORY-Befehl. Da die Bildschirmroutinen grafisch arbeiten, mußte eine andere Form der Dumpausgabe gefunden werden. Sie arbeitet jetzt mit direkter Cursoradressierung, um die durch die grafische Ausgabe verminderte Schreibgeschwindigkeit auszugleichen. Unverändert blieben

Tabelle 2.8. Markentabelle

#### Symbols:

|       |         |       |          |       |        |
|-------|---------|-------|----------|-------|--------|
| FE01  | ARG1    | FE03  | ARG2     | FE05  | ARG3   |
| F105  | ASHEX   | F021  | BEGIN    | 0008  | BS     |
| F9F7I | CHARGEN | F735  | CINIT    | F75E  | CIX    |
| 000C  | CLS     | F917I | CLSCREEN | F919I | CLSW   |
| F125  | CNVBN   | 000D  | CP1A     | 000F  | CP1B   |
| F794  | CPO     | F72B  | CP00     | F72C  | CPO1   |
| F72E  | CPO11   | 000D  | CR       | F1C0  | CRALF  |
| F76B  | CRI     | F71D  | CRI1     | C000  | CRTBG  |
| FE37  | CRTRAM  | 0019  | CSIOA    | 001B  | CSIOB  |
| F74F  | CSTS    | F75D  | CSTSR    | F747  | CTAB   |
| 0014  | CTCO    | 0015  | CTC1     | 0016  | CTC2   |
| 0017  | CTC3    | 000A  | CURD     | 0008  | CURL   |
| 0015  | CURR    | FE39  | CURXY    | FE07  | DATA   |
| 000C  | DP1A    | 000E  | DP1B     | 0018  | DSIOA  |
| 001A  | DSIOB   | 0003  | EOF      | 0004  | EOT    |
| 0006  | ERA     | F054  | ERROR    | 001B  | ESC    |
| FE3B  | ESCSEQ  | 0003  | ETX      | F65B  | FILL1  |
| FDF7  | FINA    | FDF7  | FITY     | F15A  | FNEXT  |
| F0E6  | GCOM    | F0B8  | GFINA    | F0D7  | GFITY  |
| F943  | GPAGE   | 001D  | GS       | FFED  | HEAD   |
| F6D1  | HELP1   | 0001  | HOME     | F0FD  | INHEX  |
| F186  | INL     | F18D  | INLOO    | FE17  | INTV   |
| FE3D  | IOB     | 0084  | IOBYT    | F6FE  | IOBYTE |
| FE4D  | IORET   | F716  | IOSET    | 0009  | KEYBC  |
| 0008  | KEYBD   | F3EB  | LDEXT    | 000A  | LF     |
| F3F9  | LOAD    | F067  | MAIN     | F1DB  | MEM    |
| F38D  | MEMMO   | F6A8  | MEMSI    | F3C6  | MOVE   |
| 001E  | NL      | F136  | OUTH     | F13F  | OUTH1  |
| F14A  | DUTH2   | F14F  | OUTHL    | F179  | PRINT  |
| FE3E  | PRINTER | 0000  | ROMOF    | 0004  | ROMON  |
| F558  | SAEXT   | F578  | SAVE     | 0001  | SCH    |
| FF6B  | STACK   | 0002  | STX      | F000  | SUCHBG |
| FE50  | TXTBU   | FE4E  | TXTPT    | FE41  | USERCO |
| FE4A  | USERLO  | FE47  | USERPO   | FE44  | USERRI |
| 0010  | VIDEO   |       |          |       |        |

SAVE, LOAD, HELP und viele andere Unterprogramme.

Das Monitorprogramm modifiziert sich während seines Laufs selbst. Das Programm ist nur im RAM lauffähig! Der Speicherbereich ab 0F000H muß also unbedingt RAM-Bereich sein.

Als Referenz zu [2] wirkt die Markentabelle 2.8., die viele wichtige Adressen enthält. Obwohl das Monitorprogramm hier nur als Hex-Dump abgedruckt werden kann, sind so doch einige Eingriffe möglich. Eine Ausnahme bilden die Bildschirmroutinen, die der folgende Abschnitt erläutert.

#### 2.8.4. Grafiksoftware

Die Hardwaregestaltung des Computers erlaubt eine Grafikauflösung von 256 mal 256 Bildpunkten. Mit den in früheren Abschnitten gegebenen Hinweisen ist eine Erhöhung der Bildauflösung bzw. die Aufrüstung auf Farbbetrieb denkbar und möglich. Um nach einer Hardwareerweiterung auch die Software den neuen Gegebenheiten anzupassen, soll dem Amateur als Grundstock die Grafiksoftware des Monitorprogramms als detailliertes Assemblerlisting zur Verfügung gestellt werden. Weiterhin kann der Anwender so die Eigenschaften der Grafiksoftware den eigenen Wünschen anpassen.

Die Grafikgrundsoftware unterstützt in der vorgestellten Form die Darstellung von Texten auf 24 Zeilen zu je 32 Zeichen sowie das Setzen und Löschen von Punkten oder Vektoren auf der 256 mal 256 großen Bildpunktmatrix.

Die Kommandos werden an der CO-Schnittstelle des Monitorprogramms übergeben. Dabei befindet sich das auszugebende Zeichen im Register C des Prozessors. Handelt es sich um einen Grafikbefehl, so müssen eventuell notwendige Koordinaten wie folgt übergeben werden:

Register L Zielkoordinate x,

Register H Zielkoordinate y,

Register E Anfangskoordinate x,

Register D Anfangskoordinate y.

Die Bildschirmsteuersoftware versteht viele Befehle:

#### CR ODH

Carriage, Return, Wagenrücklauf

Der Cursor kehrt an den Zeilenanfang zurück, CR wird meist gemeinsam mit LF angewendet. CR ist auch das Abschlußzeichen bei einer Eingabe über die Tastatur (RETURN-Taste).

#### LF OAH

Line Feed, Zeilenvorschub

Der Cursor bewegt sich eine Zeile tiefer. Dabei wird gegebenenfalls der Bildschirminhalt nach oben gerollt.

#### BS 08H

Backspace

Der Cursor bewegt sich um eine Position nach links.

#### HOME 01H

Der Cursor wird am Bildschirmanfang positioniert.

#### RIGH 15H

Der Cursor bewegt sich um eine Position nach rechts. Dabei wird gegebenenfalls ein LF und eventuell ein Rollen ausgeführt.

#### UP 1AH

Der Cursor bewegt sich um eine Zeile nach oben.

#### CLS 0CH

Löscht den Bildschirm. Anschließend wird ein HOME ausgeführt, so daß der Cursor nach Ausführung des Kommandos CLS am Bildschirmanfang steht.

#### ESC 1EH

Escape, Umschaltung

Die nach ESC folgenden Zeichen werden nicht nach dem ASCII-Code interpretiert. Sie bilden Steuersequenzen, die ein Befehlsdecoder entschlüsselt und zur Ausführung bringt. Mehrere Escapesequenzen versteht die Software, die nachfolgend beschrieben werden.

#### ESC rc

Der Cursor wird auf die Zeile r-80H und die Spalte c-80H positioniert. Um den Cursor zu positionieren, müssen also 3 Zeichen für eine komplette Escapesequenz ausgegeben werden, wobei das gesetzte 8.bit bei der Ausgabe der Koordinaten die Cursorpositionierung kennzeichnet.

#### ESC 00H

CLSREEN: Löscht den Bildschirm. Entspricht dem Befehl CLS.

#### ESC 01H

**CLSW:** Löscht den Bildschirm weiß.

**ESC 02H**

**SETPOINT:** Setzt oder löscht einen Punkt auf den Koordinaten, die in den Registern H und L übergeben wurden.

**ESC 03H**

**DRAW:** Zeichnet oder löscht einen Vektor von den Koordinaten, die in den Registern D und E übergeben wurden, nach einem Punkt mit den Koordinaten, die in den Registern H und L übergeben wurden.

**ESC 04H**

**POINT:** Testet, ob der Bildpunkt mit den Koordinaten in den Registern H und L gesetzt ist: Zero-Flag gesetzt  $\hat{=}$  Bildpunkt war dunkel. Zero-Flag nicht gesetzt  $\hat{=}$  Bildpunkt war hell.

**ESC 05H**

**PENUP:** Der Schreibstift wird angehoben. Dieser Befehl stammt aus der Plottertechnik. Ist der Schreibstift angehoben, sind alle Zeichenbefehle ohne Wirkung.

**ESC 06H**

**PENDOWN:** Der Schreibstift wird abgesenkt. Umkehrung des Befehls PENUP. Zeichnen ist erst nach PENDOWN möglich. Die Software wird mit PENDOWN initialisiert.

**ESC 07H**

**ERASER:** Schaltet den Schreibmodus auf «Radiergummi» um. Es können bereits gezeichnete Linien oder Punkte wieder gelöscht werden.

**ESC 08H**

**PEN:** Schaltet den normalen Schreibmodus ein (Initialisierungszustand).

**ESC 09H**

**PAGE:** Diese Funktion erlaubt die Erzeugung mehrerer Bildschirmseiten. Das Doppelregister enthält beim Aufruf die Adresse des Bildwiederholspeichers, auf den die Grafikbefehle wirken. Das Doppelregister DE enthält die Adresse des Bildwiederholspeichers, der durch die Hardware zur Anzeige gebracht wird. Zu beachten ist, daß bei der vorgestellten Hard- und Software diese Adressen Vielfache von 8 Kbyte sein müssen.

**ESC 0BH**

**CURSOR OFF:** Cursordarstellung wird unterdrückt. Diese Betriebsart ist sinnvoll, wenn Grafiken erstellt werden sollen und ein

irgendwo auf dem Bildschirm stehender Cursor stört.

**ESC 0CH**

**CURSOR ON:** Cursor wird angezeigt. (Initialisierungswert)

Selbstverständlich sind diese Befehle nur Grundbefehle. Beispielsweise wären folgende Erweiterungen denkbar:

- Erhöhung der Anzahl der Zeichen je Zeile. Bei einer Zeichenmatrix von 6 mal 10 Bildpunkten ist eine Textdarstellung von 40 Zeichen auf 24 Zeilen denkbar.
- Erweiterung der Grafikgrundbefehle: Quadrate oder Kreise.
- Bildschirmattribute (Blinken, Invertieren usw.).
- Verlegen des Grafiknullpunktes koordinatengemäß auf unten links.
- Maßstabsänderung und Kursivdarstellung des Zeichensatzes.

Mit Hilfe des Assemblerlistings kann der Programmfluß verfolgt werden. Nach Auswerten des IO-bytes gelangt das Zeichen zur Conout-Routine. Nach dem Retten aller Register wird der Cursor gelöscht. Je nachdem ob eine Escapesequenz läuft oder nicht, wird das Programm verzweigt:

1. Es läuft keine Escapesequenz. Test, ob eine beginnt, wenn nicht Sprung zum Textausgabeteil des Programms (con2).
2. Es läuft eine Escapesequenz. Auswertung der Steuerfolge. Entweder Cursor positionieren oder Grafikbefehl ausführen.

Der Cursor wird im Programmteil ab con1 positioniert. Das Programm weist keine Besonderheiten auf, wertet ab Marke con12 den Grafikbefehl aus und steuert die Sprungverteilung zu den entsprechenden Unterprogrammen. Die Tabelle contbl enthält die Sprungadressen.

Ein wichtiges Unterprogramm ist setpoint. Hier wird aus den in den Registern H und L übergebenen Koordinaten eine Adresse im Bildwiederholspeicher berechnet. Wie schon früher beschrieben, ist der Bildschirm horizontal in 8 bit breite Zeichenfelder aufgeteilt. Dies ergibt sich aus der Verarbeitungsbreite von einem byte des Mikroprozessors U 880. Der Bildwiederholspeicher kann demzufolge auch nur byteweise beschrieben oder gelesen werden. Da bei einer Grafikausgabe

aber jeder beliebige Punkt auf dem Bildschirm zugänglich sein soll, muß die Routine setpoint zusätzlich auf einzelne bits im byte zugreifen können. Das Programm setpoint dividiert zunächst die Koordinaten durch 8, um die relative Byteadresse im Bildwiederholspeicher zu bestimmen. Die zweckmäßige Zuordnung der Koordinaten Y und x zu den Registern H und L erlaubt diese einfache Lösung. Nun muß man die Bitposition bestimmen. Sie ergibt sich einfach aus den niedrigerwertigen 3 bits der x-Koordinate. Mit Hilfe mehrerer Schiebebefehle und einem OR-Befehl wird der U 880-Befehl SET X, (HL) sozusagen zusammengebaut (hier steht X für das entsprechend zu beeinflussende bit). Nachdem die Adresse des Bildpunkts durch die Bildwiederholspeicheranfangsadresse vervollständigt wurde, wird der erzeugte Bitbefehl ausgeführt.

Die Unterprogramme pen, erase und point modifizieren den genannten OR-Befehl (siehe Marke setp1) derart, daß je nach Betriebsart die Befehle SET x, (HL), RES x, (HL) oder BIT x, (HL) generiert werden (x steht immer für das entsprechende bit). Wie zu erkennen ist, modifiziert sich das Programm entsprechend der auszuführenden Funktion und deren Parameter. Diese Verfahrensweise ermöglicht schnell laufende und kurze Programme. Gerade bei Grafikbefehlen muß die Laufzeit besonders gering sein. Insbesondere gilt das für setpoint, denn die noch zu beschreibende Routine draw zeichnet ja eine Linie aus vielen Einzelpunkten, wozu sie sehr oft setpoint aufruft. Selbstmodifizierende Programme eignen sich nur für den RAM-Speicher. Weiterhin stehen sie nur sehr selten so im Speicher, wie sie beispielsweise im Assemblerlisting abgedruckt sind. Vor allem erschwert dieser Umstand die Fehlersuche in der Entwicklungsphase. Doch wegen der genannten Vorteile wurden die weniger schwerwiegenden Nachteile in Kauf genommen.

Ein weiteres Unterprogramm, das in dieser Art programmiert wurde, ist draw. Draw zeichnet Vektoren von den Anfangskoordinaten x0 und y0 (Register DE) zu den Endkoordinaten x1 und y1 (Register HL). Nach Tausch der Koordinaten werden die Geradensteigungen dx und dy berechnet. Mit de-

ren Hilfe wird der Vektor gezeichnet. Man benutzt dazu einen relativ einfachen Algorithmus, der auf *Bresenham* zurückgeht. An einer gedachten Linie von (x0,y0) nach (x1,y1) sind die einzelnen Bildpunkte so zu setzen, daß sie möglichst nahe an dieser Linie liegen. Geht die Linie beispielsweise von unten links nach rechts oben, so muß man, daraus ableitend, einen Schritt nach oben gehen, wenn man sich unterhalb dieser gedachten Linie befindet. Ist man aber gerade oberhalb dieser Linie, so muß man die momentane Schreibposition nach rechts bewegen. Das wird erreicht, wenn man die Steigung in x- und y-Richtung mit der gewünschten Geradensteigung vergleicht. Im Grunde läßt sich das Verfahren so weit vereinfachen, daß lediglich das Vorzeichen der momentanen Steigung zur Entscheidung, ob ein Schritt in x- oder y-Richtung gemacht werden soll, herangezogen wird. Nun müssen noch weitere Fälle berücksichtigt werden, damit das Ganze auch funktioniert. Erstens soll der Algorithmus bei Zeichenbefehlen aus allen 4 Quadranten nach allen 4 Quadranten arbeiten. Erreicht wird das durch entsprechende Negationen und Programmodifikationen. Zweitens müssen Sonderfälle berücksichtigt werden (z. B. das Zeichnen einer Geraden), was eine sinnvolle Vorbelegung erfordert. Das Unterprogramm draw erfüllt all diese Bedingungen und ermöglicht in Verbindung mit setpoint ein recht schnelles Zeichnen von Vektoren auf dem Bildschirm. Das Unterprogramm char kopiert ein Zeichen aus dem Zeichengenerator (charge) in den Bildwiederholspeicher. Anschließend wird der Cursor auf die nächste Schreibposition bewegt. Der Zeichengenerator selbst ist sozusagen Bestandteil des Programms. Demzufolge steht er ebenfalls im RAM-Speicher und kann beliebig verändert werden. Bild 2.54 zeigt den Zeichensatz mit den dazugehörigen Hexcodes. Mit der Monitorfunktion MEMORY können bequem einzelne Zeichen verändert werden. Es ist sogar denkbar, einen gänzlich anderen Zeichensatz vom Magnetband nachzuladen, z. B. einen ganz speziellen Zeichensatz aus Buchstaben, Zahlen und Impulssymbolen, um mit dem Computer einen einfachen Logikanalysator zu realisieren (Bild 2.55).

|     |     |     |     |
|-----|-----|-----|-----|
| 00H | 20H | 50H | 50H |
| 00H | 20H | 50H | 50H |
| 00H | 20H | 50H | F8H |
| 00H | 20H | 00H | 50H |
| 00H | 20H | 00H | F8H |
| 00H | 00H | 00H | 50H |
| 00H | 20H | 00H | 50H |
| 00H | 00H | 00H | 00H |
| 20H | 18H | 30H | 60H |
| F0H | 98H | 48H | 60H |
| 28H | 40H | 28H | 40H |
| 70H | 20H | 10H | 20H |
| A0H | 10H | A8H | 00H |
| 78H | C8H | 48H | 00H |
| 20H | C0H | B0H | 00H |
| 00H | 00H | 00H | 00H |
| 40H | 10H | 20H | 00H |
| 20H | 20H | A8H | 20H |
| 10H | 40H | 70H | 20H |
| 10H | 40H | F8H | F8H |
| 10H | 40H | 70H | 20H |
| 20H | 20H | A8H | 20H |
| 40H | 10H | 20H | 00H |
| 00H | 00H | 00H | 00H |
| 00H | 00H | 00H | 00H |
| 00H | 00H | 00H | 80H |
| 00H | F8H | 00H | 40H |
| 60H | 00H | 00H | 10H |
| 60H | 00H | 30H | 08H |
| 40H | 00H | 30H | 00H |
| 20H | 00H | 00H | 00H |
| 70H | 20H | 70H | 70H |
| 88H | 30H | 88H | 88H |
| 88H | 20H | 80H | 80H |
| A8H | 20H | 60H | 70H |
| 88H | 20H | 10H | 80H |
| 88H | 20H | 08H | 88H |
| 70H | 70H | F8H | 70H |
| 00H | 00H | 00H | 00H |
| 40H | F8H | 60H | F8H |
| 60H | 08H | 10H | 80H |
| 50H | 08H | 08H | 40H |
| 48H | 78H | 78H | 20H |
| FCH | 80H | 88H | 10H |
| 40H | 88H | 88H | 08H |
| 40H | 70H | 70H | 08H |
| 00H | 00H | 00H | 00H |
| 70H | 70H | 00H | 00H |
| 88H | 88H | 30H | 30H |
| 88H | 88H | 30H | 30H |
| 70H | F0H | 00H | 00H |
| 88H | 80H | 30H | 30H |
| 88H | 40H | 30H | 30H |
| 70H | 30H | 00H | 20H |
| 00H | 00H | 00H | 10H |

|     |     |     |     |
|-----|-----|-----|-----|
| 40H | 00H | 10H | 70H |
| 20H | 00H | 20H | 88H |
| 10H | F8H | 40H | 80H |
| 08H | 00H | 80H | 40H |
| 10H | F8H | 40H | 20H |
| 20H | 00H | 20H | 00H |
| 40H | 00H | 10H | 20H |
| 00H | 00H | 00H | 00H |
| 70H | 20H | 78H | 70H |
| 88H | 50H | 88H | 88H |
| 80H | 88H | 88H | 08H |
| B0H | 88H | 78H | 08H |
| A8H | F8H | 88H | 08H |
| A8H | 88H | 88H | 88H |
| 70H | 88H | 78H | 70H |
| 00H | 00H | 00H | 00H |
| 78H | F8H | F8H | 70H |
| 88H | 08H | 08H | 88H |
| 88H | 08H | 08H | 08H |
| 88H | 78H | 78H | E8H |
| 88H | 08H | 08H | 88H |
| 88H | 08H | 08H | 88H |
| 78H | F8H | 08H | F0H |
| 00H | 00H | 00H | 00H |
| 88H | 70H | 80H | 88H |
| 88H | 20H | 80H | 48H |
| 88H | 20H | 80H | 28H |
| F8H | 20H | 80H | 18H |
| 88H | 20H | 80H | 28H |
| 88H | 20H | 88H | 48H |
| 88H | 70H | 70H | 88H |
| 00H | 00H | 00H | 00H |
| 08H | 88H | 88H | 70H |
| 08H | D8H | 98H | 88H |
| 08H | A8H | A8H | 88H |
| 08H | A8H | C8H | 88H |
| 08H | 88H | 88H | 88H |
| 08H | 88H | 88H | 88H |
| F8H | 88H | 88H | 70H |
| 00H | 00H | 00H | 00H |
| 78H | 70H | 78H | 70H |
| 88H | 88H | 88H | 88H |
| 88H | 88H | 88H | 08H |
| 78H | 88H | 78H | 70H |
| 08H | A8H | 28H | 80H |
| 08H | 48H | 48H | 88H |
| 08H | B0H | 88H | 70H |
| 00H | 00H | 00H | 00H |
| F8H | 88H | 88H | 88H |
| 20H | 88H | 88H | 88H |
| 20H | 88H | 88H | 88H |
| 20H | 88H | 88H | A8H |
| 20H | 88H | 50H | D8H |
| 20H | 70H | 20H | 88H |
| 00H | 00H | 00H | 00H |

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| 88H | 88H | 88H | F8H | 70H |
| 88H | 88H | 88H | 80H | 10H |
| 50H | 50H | 50H | 40H | 10H |
| 20H | 20H | 20H | 20H | 10H |
| 50H | 20H | 20H | 10H | 10H |
| 88H | 20H | 20H | 08H | 10H |
| 88H | 20H | 20H | F8H | 70H |
| 00H | 00H | 00H | 00H | 00H |
| 00H | 70H | 20H | 20H | 00H |
| 08H | 40H | 70H | 70H | 00H |
| 10H | 40H | 40H | A8H | 00H |
| 20H | 40H | 40H | 20H | 00H |
| 40H | 40H | 40H | 20H | 00H |
| 80H | 40H | 20H | 20H | 00H |
| 00H | 70H | 20H | 20H | F8H |
| 00H | 00H | 00H | 00H | 00H |
| 30H | 00H | 10H | 10H | 00H |
| 30H | 00H | 10H | 10H | 00H |
| 10H | 30H | 70H | 70H | 60H |
| 20H | 40H | 90H | 90H | 90H |
| 00H | 78H | 90H | 90H | 10H |
| 00H | 48H | 90H | 90H | 10H |
| 00H | F8H | 78H | 78H | E0H |
| 00H | 00H | 00H | 00H | 00H |
| 40H | 00H | 00H | 00H | 00H |
| 40H | 00H | C0H | C0H | 00H |
| 70H | 60H | 20H | 20H | E0H |
| 48H | 90H | 70H | 70H | 90H |
| 48H | 70H | 20H | 20H | 90H |
| 48H | 10H | 20H | 20H | E0H |
| F0H | E0H | 20H | 20H | 80H |
| 00H | 00H | 00H | 00H | F0H |
| 10H | 00H | 00H | 00H | 00H |
| 10H | 20H | 40H | 40H | 10H |
| 70H | 00H | 00H | 00H | 90H |
| 90H | 20H | 40H | 40H | 50H |
| 90H | 20H | 40H | 40H | 30H |
| 90H | 20H | 40H | 40H | 50H |
| 90H | 20H | 50H | 50H | 90H |
| 00H | 00H | 20H | 20H | 00H |
| 00H | 00H | 00H | 00H | 00H |
| 10H | 00H | 00H | 00H | 00H |
| 10H | 00H | 00H | 00H | 00H |
| 20H | 58H | 70H | 70H | 60H |
| 20H | A8H | 90H | 90H | 90H |
| 20H | A8H | 90H | 90H | 90H |
| 20H | A8H | 90H | 90H | 90H |
| 60H | 88H | 90H | 90H | 60H |
| 00H | 00H | 00H | 00H | 00H |
| 00H | 00H | 00H | 00H | 00H |
| 70H | 70H | D0H | D0H | E0H |
| 90H | 48H | 30H | 30H | 10H |
| 90H | 48H | 10H | 10H | 60H |
| 70H | 70H | 10H | 10H | 80H |
| 10H | 40H | 10H | 10H | 70H |
| 10H | C0H | 00H | 00H | 00H |

|       |     |       |     |       |     |       |     |
|-------|-----|-------|-----|-------|-----|-------|-----|
| ..... | 00H | ..... | 00H | ..... | 00H | ..... | 00H |
| ..... | 20H | ..... | 00H | ..... | 00H | ..... | 00H |
| ..... | 70H | ..... | 48H | ..... | 88H | ..... | 88H |
| ..... | 20H | ..... | 48H | ..... | 88H | ..... | 88H |
| ..... | 20H | ..... | 48H | ..... | 50H | ..... | A8H |
| ..... | A0H | ..... | 48H | ..... | 50H | ..... | A8H |
| ..... | 40H | ..... | F0H | ..... | 20H | ..... | 50H |
| ..... | 00H | ..... | 00H | ..... | 00H | ..... | 00H |
| ..... | 00H | ..... | 00H | ..... | 00H | ..... | 40H |
| ..... | 00H | ..... | 00H | ..... | 00H | ..... | 20H |
| ..... | 88H | ..... | 90H | ..... | F8H | ..... | 20H |
| ..... | 50H | ..... | 90H | ..... | 40H | ..... | 10H |
| ..... | 20H | ..... | A0H | ..... | 20H | ..... | 20H |
| ..... | 50H | ..... | C0H | ..... | 10H | ..... | 20H |
| ..... | 88H | ..... | 80H | ..... | F8H | ..... | 40H |
| ..... | 00H | ..... | E0H | ..... | 00H | ..... | 00H |
| ..... | 20H | ..... | 10H | ..... | 00H | ..... | A8H |
| ..... | 20H | ..... | 20H | ..... | 10H | ..... | 50H |
| ..... | 20H | ..... | 20H | ..... | A8H | ..... | A8H |
| ..... | 20H | ..... | 40H | ..... | 40H | ..... | 50H |
| ..... | 20H | ..... | 20H | ..... | 00H | ..... | A8H |
| ..... | 20H | ..... | 20H | ..... | 00H | ..... | 50H |
| ..... | 20H | ..... | 10H | ..... | 00H | ..... | A8H |
| ..... | 20H | ..... | 00H | ..... | 00H | ..... | 00H |

Bild 2.54

Zeichengenerator in grafischer und hexadezimaler Form. (Achtung: das höchstwertigste Bit steht jeweils rechts!)

Bild 2.55 Grafik-Unterprogramme

```

        .comment;

*****
*      Grafik-Routinen fuer 256*256 pixel          *
*      Alpha-Routinen fuer 32x24 zeichen          *
*****
Parameteruebergabe
C: Zeichen oder Steuercode
H: y zielkoordinate
L: x zielkoordinate
D: y
E: x
F: Z-Flag (POINT)

F7D3      conout::           push de      ;register retten
F7D3      D5                 push bc
F7D4      C5
F7D5      32 FE3C           ld   (akk),a
F7D8      DD E5             push ix
F7DA      E5
F7DB      C3      coinit:  defb 0c3h  ;jp vor init, ld hl,... nach init
F7DC      F90B      cursor:  defw coini ;nach init cursoradresse
F7DE      36 00      cursi:  ld   (hl),0  ;cursor loeschen
F7E0      79
F7E1      21 FE3B           ld   a,c      ;zeichen
F7E2      1d   hl,escseq
F7E4      CB 46             bit  0,(hl)  ;lauft esc-sequenz?
F7E6      20 0B             jr   nz,coni  ;ja
F7E8      FE 1B             cp   esc
F7EA      C2 F876           jp   nz,con2  ;normales zeichen
F7ED      CB C6             set  0,(hl)  ;esc-flag
F7EF      CB 8E             res  1,(hl)  ;rc-flag
F7F1      18 31             jr   coret

F7F3      CB 7F      coni:   bit  7,a      ;grafikbefehl?
F7F5      28 40             jr   z,coni2  ;ja
F7F7      CB 4E             bit  1,(hl)  ;ir?
F7F9      CB CE             set  1,(hl)
F7FB      CB BF             res  7,a
F7FD      28 0C             jr   z,coni0  ;row
F7FF      CB 86             res  0,(hl)
F801      06 00             ld   b,0
F803      CB B9             res  7,c      ;col
F805      2A FE39           ld   hl,(curxy)
F808      09
F809      18 0D             add  hl,bc
                           jr   conret

F80B      21 0000           con10:  ld   hl,0      ;adresse bildwiederholspeicher
F80E      B7
F80F      28 07             jr   z,conret
F811      47
F812      11 0140           ld   b,a      ;multiplikationszaehler
                           ld   de,10*20h ;abstand zwischen zwei zeilen
F815      19      con100:  add  hl,de
F816      10 FD             djnz con100
F818      22 FE39           conret: ld   (curxy),hl
F81B      11 0120           ld   de,9*32      ;cursorzeile
F81E      19
F81F      36 F8             add  hl,de      ;berechnen
                           ld   (hl),11111000b ;cursor

```

```

F821    22 F7DC      ld    (cursor),hl
F824    E1           coret:   pop   hl
F825    DD E1       coret:   pop   ix
F827    C1           coret:   pop   bc
F828    D1           coret:   pop   de
F829    3A FE3C      coret:   ld    a,(akku)
F82C    C9           coret:   ret
F82D          coreti:
F82D    D9           coreti:  exx
F82E    08           coreti:  ex   af,af'
F82F    F1           coreti:  pop   af
F830    C1           coreti:  pop   bc
F831    D1           coreti:  pop   de
F832    E1           coreti:  pop   hl
F833    D9           coreti:  exx
F834    08           coreti:  ex   af,af'
F835    1B ED         coreti:  jr    coret

;grafik-befehle
F837    CB 86         con12:  res   0,(hl)      ;esc-flag ruecksetzen
F839    E1
F83A    E5           con12:  push  hl
F83B    D9           con12:  push  hl
F83C    08           con12:  exx
F83D    E5           con12:  ex   af,af'
F83E    D5           con12:  push  hl
F83F    C5           con12:  push  de
F840    F5           con12:  push  bc
F841    21 F82D      con12:  push  af
F842    E5           con12:  ld    hl,coreti
F844    E5           con12:  push  hl
F845    D9           con12:  exx
F846    E5           con12:  push  hl
F847    21 F855      con12:  ld    hl,contbl
F84A    06 00         con12:  ld    b,0
F84C    CB 21         con12:  sla   c          ;#2
F84E    09           con12:  add   hl,bc
F84F    7E           con12:  ld    a,(hl)      ;adresse der grafikroutine
F850    23           con12:  inc   hl
F851    66           con12:  ld    h,(hl)
F852    6F           con12:  ld    l,a
F853    E3           con12:  ex   (sp),hl      ;hl wieder koordinate
F854    C9           con12:  ret

;tabelle der grafikbefehle
F855    F917         contbl: defw  clscreen   ;0
F857    F919         contbl: defw  clsw      ;1
F859    F95F         contbl: defw  setpoint   ;2
F85B    F97D         contbl: defw  draw       ;3
F85D    F9D4         contbl: defw  point      ;4
F85F    F9EE         contbl: defw  penup      ;5
F861    F9F1         contbl: defw  pendn      ;6
F863    F9E5         contbl: defw  erase      ;7
F865    F9E8         contbl: defw  pen        ;8
F867    F943         contbl: defw  gpage     ;9
F869    F875         contbl: defw  reserve    ;10
F86B    F939         contbl: defw  cursof    ;11
F86D    F92C         contbl: defw  cursor     ;12
F86F    F875         contbl: defw  reserve    ;13
F871    F875         contbl: defw  reserve    ;14
F873    F875         contbl: defw  reserve    ;15

```

|                |            |          |                                   |  |
|----------------|------------|----------|-----------------------------------|--|
| F875           | C9         | reserve: | ret                               | nicht benutzt, reserve dient der Verweiterung. |
| <hr/>          |            |          |                                   |  |
| F876           | 21 F818    | con2:    | ;text ausgabe                     |  |
| F879           | E5         |          | ld hl,conret                      | fruecksprung (cursor an)                       |
| F87A           | E6 7F      |          | push hl                           |  |
| F87C           | FE 20      |          | and 7fh                           |  |
| F87E           | 38 2D      |          | cp "                              | ;alpha oder steuerzeichen?                     |
| F880           | 6F         | char1:   | jr c,con3                         | ;steuerzeichen                                 |
| F881           | 26 00      |          | ;zeichen auf bildschirm schreiben |  |
|                |            |          | ld l,a                            | ;zeichencode                                   |
|                |            |          | ld h,0                            |  |
|                |            |          | rept 3                            | ;#8  |
|                |            |          | add hl,hl                         |  |
|                |            |          | endm                              |  |
| F883           | 29         | +        | add hl,hl                         |  |
| F884           | 29         | +        | add hl,hl                         |  |
| F885           | 29         | +        | add hl,hl                         |  |
| F886           | 11 F9F7    |          | ld de,chargen                     |  |
| F889           | 19         |          | add hl,de                         | ;adresse im zeichengenerator                   |
| F88A           | DD 2A FE39 |          | ld ix,(curxy)                     | ;cursor  |
| F88E           | DD E5      |          | push ix                           |  |
| F890           | 11 0020    |          | ld de,20h                         | ;offset zur naechsten zeile                    |
| F893           | 06 08      |          | ld b,8                            |  |
| F895           | 7E         | char1:   | ld a,(hl)                         | ;zeile im zeichen                              |
| F896           | DD 77 00   |          | ld (ix),a                         | ;in den bildwiederholmspeicher                 |
| F899           | 23         |          | inc hl                            | ;naechste zeile im zeichen                     |
| F89A           | DD 19      |          | add ix,de                         | ;naechste zeile auf dem bildschirm             |
| F89C           | 10 F7      |          | djnz char1                        | ;weiter  |
| F89E           | E1         |          | pop hl                            |  |
| <hr/>          |            |          |                                   |  |
| ;cursor rechts |            |          |                                   |  |
| F89F           | 23         | cocur:   | inc hl                            | ;naechste zeichenposition                      |
| F8A0           | 7D         |          | ld a,l                            | ;test ob noch in zeile                         |
| F8A1           | E6 1F      |          | and 00011111b                     |  |
| F8A3           | CO         |          | ret nz                            | ;keine neue zeile                              |
| F8A4           | 2B         |          | dec hl                            | ;zur korrektur                                 |
| F8A5           | CD F8D0    |          | call cocr                         | ;cr ausfuehren                                 |
| F8A8           | 3E 0A      |          | ld a,1f                           | ;anschliessend lf                              |
| F8AA           | 22 FE39    |          | ld (curxy),hl                     |  |
| <hr/>          |            |          |                                   |  |
| F8AD           | 2A FE39    | con3:    | ld hl,(curxy)                     |  |
| F8B0           | 11 0140    |          | ld de,10*20h                      | ;abstand zwischen zwei zeilen                  |
| F8B3           | FE 0D      |          | cp cr                             |  |
| F8B5           | 28 19      |          | jr z,cocr                         | ;carriage return                               |
| F8B7           | FE 0A      |          | cp lf                             |  |
| F8B9           | 28 1A      |          | jr z,colf                         | ;line feed                                     |
| F8BB           | FE 08      |          | cp bs                             |  |
| F8BD           | 28 37      |          | jr z,cobs                         | ;bs  |
| F8BF           | FE 0C      |          | cp cls                            |  |
| F8C1           | 28 54      |          | jr z,clscreen                     | ;cls   |
| F8C3           | FE 01      |          | cp home                           |  |
| F8C5           | 28 61      |          | jr z,cohome                       | ;home  |
| F8C7           | FE 1A      |          | cp curu                           |  |
| F8C9           | 28 35      |          | jr z,cooup                        | ;cursor hoch                                   |
| F8CB           | FE 15      |          | cp curr                           |  |
| F8CD           | 28 D0      |          | jr z,cocur                        | ;cursor rechts                                 |
| F8CF           | C9         |          | ret                               |  |

```

;carriage return
F8D0 7D      cocr:    ld a,1           ;1. position herstellen
F8D1 E6 E0
F8D3 6F
F8D4 C9      ret

;line feed
F8D5 19      colf:    add hl,de        ;naechste zeile
F8D6 7C
F8D7 E6 1F
F8D9 FE 1F
F8DB D8      corol:   and 0001111b    ;relative position
F8DC ED 52
F8DE E5      coroll:  cp 24*10/8+1  ;im bildschirmbereich?
F8DF 21 0000
F8E2 E5
F8E3 19      add hl,de        ;1. zeile
F8E4 D1      pop de
F8E5 01 1E00
F8E8 ED B0
F8EA 62      ldir
F8EB 6B      ld h,d          ;nun letzte zeile loeschen
F8EC 13      ld l,e
F8ED 01 0200
F8F0 36 00
F8F2 ED B0
F8F4 E1      pop hl          ;cursoradresse
F8F5 C9      ret

;backstep
F8F6 7D      cobs:    ld a,l
F8F7 E6 1F
F8F9 2B
F8FA C0      dec hl
F8FB 23
F8FC 7D
F8FD F6 1F
F8FF 6F      ret nz          ;normaler bs
                           ;ans zeilenende
                           ;anschliessend cursor hoch

;cursor hoch
F900 B7      cocuup:  or a
F901 ED 52
F903 7C
F904 E6 E0
F906 FE 00      cocupi:  cp 0           ;ist cursor noch im bildschirm?
F908 D0      ret nc          ;ja
F909 18 1D      jr cohomed

;bildschirm loeschen
F90B 01 F818
F90E C5      coini:   push bc
F90F CD F943
F912 3E 21      call gpage
F914 32 F7DB
                           ld a,21h    ;programm patchen
                           ld (coinit),a

;clsreen::
F917 AF      xor a          ;dunkel loeschen
F918 01
F919 3E FF      clsw:::  defb 1
F91B 47      ld a,255      ;weiss loeschen
                           ld b,a

```

```

F91C 2A FE37      ld   h1,(crtram)
F91F 70           ld   (h1),b      $loeschen
F920 54           ld   d,h
F921 50           ld   e,l
F922 13           inc  de
F923 01 2000      ld   bc,256/8*256 ;anzahl
F926 ED B0         ldir

;home
F928 2A F80C      cohomed: ld   h1,(con10+1)
F92B C9           ret

F92C 21 0036      curson: ld   h1,0036h    ;(ld(h1),0)
F92F 22 F7DE      ld   (curs1),h1
F932 21 F836      ld   h1,0f836h    ;(ld(h1),11111000b)
F935 22 F81F      ld   (curs2),h1
F938 C9           ret

F939 21 0000      cursof: ld   h1,0      ;programm so veraendern, dass
F93C 22 F7DE      ld   (curs1),h1  alle cursor zeichnenden
F93F 22 F81F      ld   (curs2),h1  ibefehle durch nops ersetzt
F942 C9           ret ;werden.

F943 7C           gpage: ld   a,h      ;hi-adresse des bildwiederhol-
F944 E6 E0         and  11100000b ;speichers
F946 67           ld   h,a
F947 2E 00         ld   1,0
F949 22 FE37      ld   (crtram),h1
F94C 22 FE39      ld   (curxy),h1 ;programm patchen
F94F 22 F80C      ld   (con10+1),h1
F952 22 F8E0      ld   (coroli+1),h1
F955 32 F977      ld   (setp3+1),a
F958 32 F907      ld   (cocup1+1),a
F95B 7A           ld   a,d
F95C D3 10         out  (video),a
F95E C9           ret

;setpoint::
F95F 7D           setpoint: ld   a,l      ;x-adresse
                           ;adresse umrechnen
                           rept 3  ;/8 weil 8 pixel pro zeichenfeld
                           srl  h
                           rr   1
                           endm

F960 CB 3C         +
F962 CB 1D         +
F964 CB 3C         +
F966 CB 1D         +
F968 CB 3C         +
F96A CB 1D         +
F96C E6 07         and  111b      ;vorderen bits zu 0
                           rept 3  ;*8 wegen code der bitbefehle
                           rlca
                           endm

F96E 07           +
F96F 07           +
F970 07           +
F971 F6 C6         setpi: or   11000110b ; op-code des bit-befehls
F973 32 F97B      ld   (setp2+1),a
F976 3E 00         setp3: ld   a,0      ;hi-teil der bildschirmadresse
F978 B4           or   h      ;statische Adresse bestimmen

```

```

F979 67
F97A CB C6      setp2:   ld   h,a
F97C C9          set   0,(hl)
                         ret

F97D E8          draw::    ex   de,h1
F97E 7B          ld   a,e      ;dx berechnen
F97F 95          sub  1      ;dx=xend-x0
F980 06          defb 06h     ;ld b, ...
F981 2C          inc   1
F982 30 04        jr   nc,draw2
F984 06          defb 06h     ;ld b, ...
F985 2D          dec   1      ;dx=-dx
F986 ED 44        neg
F988 F5          draw2:   push af
F989 78          ld   a,b
F98A 32 F9C4      draw3:   ld   (xstep),a
F98D 7A          ld   a,d      ;dy berechnen
F98E 94          sub  h      ;dy=yend-y0
F98F 06          defb 06h     ;ld b, ...
F990 24          inc   h
F991 0E FF        ld   c,-1     ;d=-1 (<0)
F993 08          jr   z,draw4  ;wenn dy=0
F995 0C          inc   c      ;d=0
F996 D2 F99D      jp   nc,draw4
F999 06          defb 06h     ;ld b, ...
F99A 25          dec   h
F99B ED 44        neg
F99D F5          draw4:   push af
F99E 78          ld   a,b
F99F 32 F9CC      draw4:   ld   (ystep),a
F9A2 79          ld   a,c      ;d
F9A3 C6 00        add  a,0
F9A5 D9          exx
F9A6 67          ld   h,a      ;hl=d setzen
F9A7 6F          ld   l,a
F9A8 08          ex   af,af'   ;vorzeichen d
F9A9 F1          pop  af      ;dy
F9AA 5F          ld   e,a
F9AB F1          pop  af      ;dx
F9AC 4F          ld   c,a
F9AD 16 00        ld   d,0
F9AF 42          ld   b,d
F9B0 08          ex   af,af'
F9B1 08          next:    ex   af,af'
F9B2 D9          exx
F9B3 00          push hl
F9B4 CD F95F      call setpoint
F9B7 E1          pop  hl
F9B8 7D          ld   a,l
F9B9 BB          cp   =
F9BA C2 F9CO      jp   nz,noend  ;nein
F9BD 7C          ld   a,h
F9BE BB          cp   =
F9BF C8          ret  z      ;ja
F9C0 08          noend:  ex   af,af'   ;vorzeichen d
F9C1 F2 F9CC      jp   p,ystep  ;wenn d>=0
F9C4 2C          xstep:  inc  l      ;bei dx>=0, sonst dec 1
F9C5 D9          exx
F9C6 B7          or   a
F9C7 ED 0A          adc  hl,de     ;d=d+dy
F9C9 C3 F9B1      jp   next

```

```

F9CC  24      ystep:   inc  h          ;bei dy>=0, sonst dec h
F9CD  D9
F9CE  B7
F9CF  ED 42
F9D1  C3 F9B1

F9D4  3A F972  point:::    ld   a,(setp1+1) ;curr code retten
F9D7  32 F9E3
F9DA  3E 46
F9DC  32 F972
F9DF  CD F95F
F9E2  3E 00      point1:::   ld   (point1+1),a
F9E4  01
F9E5  3E 86      erase:::   ld   a,10000110b ;res x,(h1)
F9E7  01
F9E8  3E C6      pen:::    ld   a,11000110b ;set x,(h1)
F9EA  32 F972
F9ED  C9          ret

F9EE  3E C9      penup:::   ld   a,0c9h     ;ret
F9F0  01
F9F1  3E 7D      pendn:::   ld   a,07dh     ;ld a,1
F9F3  32 F95F
F9F6  C9          ret

C      INCLUDE CHARGEN.MAC
F9F7  .           C      chargen:::

F9F7  00 00 00 00  C
F9FB  00 00 00 00  C      defb 0,0,0,0,0,0,0,0 ;0H
F9FF  00 00 00 00  C      defb 0,0,0,0,0,0,0,0 ;1H

;listing ist hier aus platzgruenden unterbrochen

FDE3  20 20 10 00  C
FDE7  00 10 A8 40  C      defb 0,16,168,64,0,0,0,0 ;7EH  B
FDEB  00 00 00 00  C
FDEF  A8 50 A8 50  C      defb 168,80,168,80,168,80,168,0 ;7FH
FDF3  A8 50 A8 00  C

```

## 2.9. BASIC-Interpreter

Wenn das Monitorprogramm läuft, kann der Computer durch diesen BASIC-Interpreter erweitert werden. Er ist weitgehend auf den Syntax des am weitesten verbreiteten Microsoft-BASIC zugeschnitten. Viele Programme anderer Kleincomputer (*KC 85-1*, *KC 85-2*) laufen auch mit diesem Interpreter. Anpassungen müssen vorgenommen werden, wenn die Programme spezielle Ausgabebefehle (WINDOW u. ä.), Ausgabeformate oder maschinennahe Befehle und Funktionen (PEEK, POKE) verwenden. Der im folgenden beschriebene Interpreter verfügt über einen erweiterten Befehlssatz, der teilweise speziell auf diesen Computer zugeschnitten

ist. Das betrifft beispielsweise die Grafikbefehle. Aber der Interpreter lässt sich auch auf andere Kleincomputer installieren, soweit die notwendigen Randbedingungen, wie Speicherbedarf, erfüllt sind. Auf Anpaßmöglichkeiten wird speziell in Abschnitt 2.9.4. hingewiesen.

### 2.9.1. Allgemeine Informationen

Der BASIC-Interpreter kann in 2 Betriebsarten betrieben werden. Im Direktmodus (oder Tischrechnermodus) wird jede Anweisung direkt nach der Eingabe ausgeführt:

```

PRINT 1 + 2 <CR>
3
READY

```

Die zweite Betriebsart, der Programmiermodus unterscheidet sich vom Direktmodus dadurch, daß vor jeder Zeile eine Zeilennummer steht:

10 PRINT 1 + 2 <CR>

RUN

3

READY

Die Anweisungen werden erst nach Start des Programms mit RUN ausgeführt.

Das Zeilenformat des Interpreters ist:

NNNN Anweisung [:Anweisung ...] <CR>

Jede Zeile beginnt mit einer Zeilennummer und endet mit <CR>, also durch Drücken der RETURN-Taste. Die Zeilennummer NNNN kann zwischen 0 und 65 529 liegen. In einem BASIC-Programm werden die Zeilen in beliebiger Schrittweite aufsteigend numeriert. Dabei ist eine Schrittweite von 10 als günstiger Wert anzusehen, da sich so in der Erprobungsphase problemlos noch Zeilen einfügen lassen. In jeder Zeile können auch mehrere Anweisungen stehen. Die Anweisungen müssen dann durch einen Doppelpunkt getrennt werden. Mit Drücken der RETURN-Taste wird die eingegebene Zeile in ein internes Format übersetzt und in das Programm eingefügt.

Der Interpreter verarbeitet den gesamten ASCII-Zeichensatz. Dabei haben einige Zeichen eine besondere Bedeutung:

- = Ergibtzeichen, Gleichheitszeichen
- <> kleiner/größer als
- +
- Subtraktion
- \* Multiplikation
- / Division
- <sup>^</sup> (Dach) Potenzierungszeichen
- ( ) Klammer auf/zu
- \$ Dollar, kennzeichnet Zeichenketten
- ! kennzeichnet Formatzeile für PRINT USING
- ' kennzeichnet Kommentar, ähnlich REM
- ;
- : trennt Anweisungen in einer Zeile
- & kennzeichnet nachfolgende Zahlen als Hex- oder Bin-Zahlen
- ? Kurzzeichen für PRINT

Weiterhin werden die folgenden Steuerzeichen akzeptiert:

CR RETURN, Ende der Zeile

BS Löschen des zuletzt eingegebenen Zeichens

LF Zeilenschaltung, kann zur übersichtlicheren Programmerstellung genutzt werden (ohne Wirkung auf den Programmfluß)

^C Abbruch des laufenden Programms und Rückkehr in die Befehlsebene

^O Unterdrücken der Consoalausgabe

^S Ausführung anhalten (zum Beispiel beim Listen langer Programme)

^Q Fortsetzen nach ^S

^T Anzeige der gerade bearbeiteten Zeile

^U Löschen der eingegebenen Zeile (vor RETURN!)

Der BASIC-Interpreter verarbeitet Gleitkommazahlen im Bereich von etwa 1E - 38 bis 1E + 38. Alle Ausgaben können mit bis zu 11 Stellen genau sein. Intern wird noch genauer gerechnet. Die Gleitkommazahlen werden intern im 6-byte-Hidden-bit-Format dargestellt:

byte 1 Exponent mit Vorzeichen  
(7 + 1 bit)

byte 2-6 Mantisse mit Vorzeichen (40 bit)

Ganzzahlige Hexadezimalkonstanten werden durch Vorsetzen eines Et-Zeichens gekennzeichnet, z. B. &3FA. Der Zahlenbereich beträgt 0 bis 65 535.

Auch Binärkonstanten lassen sich darstellen. Man kennzeichnet dies durch das Voranstellen von zwei Et-Zeichen, z. B. &&1001010101. Wie bei den Hexadezimalkonstanten ist die Darstellung im Zahlenbereich von 0 bis 65 535 möglich.

Um mit dem BASIC-Interpreter neben numerischen Rechnungen auch die Textverarbeitung zu ermöglichen, gibt es die Strings (Zeichenketten). Ein String ist eine Kette aus 0 bis 255 beliebigen Zeichen und wird immer in Anführungszeichen eingeschlossen (Beispiel: "abc").

Variablen sind Bezeichner für Größen, deren Werte erst während des Programmlaufs festgelegt werden (im Gegensatz zu Konstanten, deren Wert schon bei der Programmerstellung festgelegt wird). Stringvariablen erkennt man am Nachsatz eines Dollarzeichens. Einige Beispiele:

```
A=1
A$="text"
TEXT$=A$
```

Ein Variablenname kann beliebig lang sein, jedoch sind nur die ersten 4 Zeichen signifikant. Wie im Beispiel ersichtlich, haben numerische und Stringvariablen manchmal den gleichen Namen. Die Unterscheidung ist durch das Dollarzeichen abgesichert.

Eine weitere Form der Variablen ist das Feld. Ein Feld ist eine Zusammenfassung vieler numerischer oder Zeichenkettendaten unter gleichem Namen. Auf jedes einzelne Feldelement kann man durch Indizierung Bezug nehmen. Ein Feld muß in der Regel mit DIM dimensioniert werden. Dabei sind so viele Indizes möglich, wie in der Dimensionierungsvereinbarung vorgegeben wurden. Die Anzahl der Dimensionen beträgt maximal 255. Das ist allerdings ein rein theoretischer Wert, denn die Größe der Dimensionen ist wie die Anzahl der Indizes von der verfügbaren Speichergröße abhängig. Einige Beispiele zur Verwendung von Feldern:

```
DIM A(10)
DIM B(3,3,3)
DIM TEXT$(200,2)
TEXT$(22,1)="abc"
```

Wird eine Variable oder eine Feldvariable das erste Mal benutzt, ohne vorher eingerichtet worden zu sein, übernimmt der Interpreter automatisch das Einrichten, d. h., er reserviert in seinem Variablerspeicher den notwendigen Speicherplatz. Felder werden dabei mit 11 Feldelementen (0 bis 10) je Dimension eingerichtet. Alle Werte der Variablen werden mit 0 bzw. mit einem Leerstring initialisiert.

## 2.9.2. BASIC-Sprachumfang

In diesem Abschnitt werden die Anweisungen, Befehle und Funktionen des BASIC-Interpreters beschrieben. Die meisten Befehle werden nur kurz erläutert, da diese Sprachbeschreibung in erster Linie zum Nachschlagen dienen soll.

Alle Angaben entsprechen dem schon in der Monitorbeschreibung verwendeten Muster. Zusätzlich soll vereinbart werden:

- | Der senkrechte Strich zwischen mehreren Angaben deutet an, daß nur eine Angabe zu verwenden ist.
- X und Y stellen einen beliebigen numerischen Ausdruck dar.
- I und J stellen einen beliebigen numerischen Ausdruck dar, der vor Ausführung der Funktion in einen Integer-Wert konvertiert wird.
- X\$ und Y\$ stellen einen beliebigen Stringausdruck dar.

### 2.9.2.1. Editbefehle

Mit den in diesem Abschnitt beschriebenen Befehlen lassen sich komfortable BASIC-Programme erstellen.

**AUTO** [<zeilennummer>[, <increment>]] stellt die automatische Zeilennumerierung ein. Die Numerierung beginnt bei <zeilennummer> mit der Schrittweite <increment>. Für fehlende Argumente wird der Wert 10 angenommen. Eine schon vorhandene Zeilennummer wird durch ein Sternchen \* gekennzeichnet. Mit RETURN ohne Eingabe einer Zeile wird der AUTO wieder abgeschaltet.

**COPY** <neue zeile>[,<inc>]=<1.zeile>[-<n.zeile>]

Der angegebene Zeilenbereich wird auf den neuen Zeilenbereich <neue zeile> mit der Schrittweite <inc> kopiert.

**DELETE** <1.zeile> [-<n.zeile>]

Es wird die angegebene Zeile oder der Zeilenbereich gelöscht.

**EDIT** <zeilennummer>

editiert die angegebene Zeile mit Hilfe einer Anzahl von Subkommandos (n ist eine Zahl zwischen 1 und 255 und kann entfallen):

- A Die unveränderte Zeile wird in den Edit-Puffer geladen.
- nD n Zeichen werden gelöscht und angezeigt.
- E beendet EDIT und ersetzt die Zeile.
- nFx findet das n-te Zeichen x und hält direkt davor an.
- H löscht alles rechts vom Cursor und geht in den I-Modus.
- I Insert ermöglicht das Einfügen von Text, bis ESC oder RETURN gedrückt wird.

nKx löscht vom Cursor bis zum n-ten Auftreten des Zeichens x, dieses aber nicht.

L listet die Zeile.

Q beendet EDIT, ohne die Zeile zu ersetzen.

nR n Zeichen werden durch n neue Zeichen ersetzt.

Drücken der 'Leertaste bewegt den Cursor nach rechts, die BS-Taste bewegt den Cursor nach links. RETURN beendet den EDIT und ersetzt die Zeile (wirkt wie E). Der Zeileneditor ermöglicht eine schnelle und bequeme Korrektur. Allerdings ist ein wenig Übung notwendig.

**LIST [<1.zeile>-[<n.zeile>]]**

**LLIST [<1.zeile>-[<n.zeile>]]**

listet ein Programm komplett (wenn die Angaben <1.zeile> und <2.zeile> fehlt) oder von <1.zeile> bis <n.zeile> (oder, wenn <n.zeile> fehlt, bis zum Ende) auf den Bildschirm (LIST) bzw. auf den Drucker (LLIST).

**LIST 10-100** listet die Zeilen 10 bis 100

**LIST 10** listet alle Zeilen ab 10.

**LIST -100** listet alle Zeilen bis 100

**NEW**

Das BASIC-Programm und alle Variablen werden gelöscht. CLOSE wird automatisch ausgeführt.

**RENUMBER [<neue nr>][,<inc>][,<alte nr>]]**

numeriert das Programm beginnend mit 10 in der Schrittweite 10 neu. Die Neunumerierung lässt sich durch Angabe von Parametern steuern. <neue nr> ist die erste neue Zeilennummer, <inc> die Schrittweite und <alte nr> die Zeile, ab der die Neunumerierung beginnen soll. Alle Referenzen (GOTO, GOSUB usw.) werden korrigiert.

### 2.9.2.2. Programmanweisungen

In diesem Abschnitt sind alle Befehle und Anweisungen zusammengefaßt, die den Programmfluß steuern. Einige weitere Befehle und Funktionen unterstützen den Datentransfer und die Einbindung von Maschinenprogrammen.

**CALL <I>[,<J>,...]**

Ein Maschinenprogramm, beginnend bei der Adresse I, wird aufgerufen. Es können Argu-

mente nach folgendem Schema übergeben werden:

<SP> → Adresse (I)  
Argument n

...

Argument 1 (J)

<HL> enthält die Rückkehradresse zum Interpreter

<BC> enthält die Anzahl der Argumente auf dem Stack

**CLEAR [<I>]**

Alle Variablen, Felder und Strings werden gelöscht. Ein Speicherbereich der Größe I für Strings lässt sich reservieren. Beim Start des Interpreters sind 100 bytes für Zeichenketten reserviert.

CLEAR 1000 reserviert 1000 bytes für Zeichen

**CONT**

Die Programmausführung wird nach Unterbrechung durch ^C oder STOP fortgesetzt.

**DATA <konstantenliste>**

definiert Datenfelder für die READ-Anweisung. Die DATA-Anweisungen sind nicht ausführbar und lassen sich an beliebiger Stelle im Programm plazieren. Die <konstantenliste> kann eine frei wählbare Anzahl von Konstanten (beliebigen Typs unter Beachtung der maximalen Zeilenlänge) enthalten, die jeweils durch ein Komma getrennt werden. Siehe auch READ und RESTORE.

DATA 10,20,30,"abc"

**DIM <liste>**

legt für ein- oder mehrdimensionale Felder die Dimension und die Indexbereiche fest. Die Felder können 1 bis 255 Dimensionen haben. Die <liste> kann wie folgt aussehen: DIM A(10,20),TEXT\$(100)

Einfache Variablen lassen sich ebenfalls vor ihrem Gebrauch einrichten:

DIM A,B,C,D

**END**

beendet die Programmausführung und schließt alle geöffneten Dateien. Die END-Anweisung steht normalerweise am Ende eines Programms. END kann auch entfallen, da das Programm nach Ausführung der letzten Programmzeile automatisch beendet wird.

**ERASE <liste>**

Felder können neu dimensioniert werden,

wenn sie vorher mit ERASE gelöscht worden sind.

**ERASE A,TEXT\$**

**FOR ... NEXT**

**FOR <var> = X TO Y [STEP Z]**

...

**NEXT [<var>]**

bewirkt, daß der von FOR und NEXT eingeschlossene Programmteil in Abhängigkeit der Ausdrücke X, Y und Z wiederholt ausgeführt wird. Die Variable <var> verwendet man als Zähler. X ist der Anfangswert des Zählers. Der Programmteil wird bis zum Endwert Y durchlaufen. Der Zähler erhöht sich oder sinkt nach jedem Schleifendurchlauf um den Wert Z. Fehlt der Zusatz STEP Z, ist die Schrittweite automatisch 1. Nach Erreichen des Endwerts wird die Programmausführung nach der NEXT-Anweisung fortgesetzt.

FOR-Schleifen können beliebig oft ineinander verschachtelt werden.

**FOR I=1 TO 10**

**FOR J=1 TO 10**

...

**NEXT J**

**NEXT I**

An Stelle von NEXT J und NEXT I kann auch NEXT J,I geschrieben werden. Es ist auch möglich die Namen der Schleifenvariablen ganz wegzulassen. Das sollte man aber nur in zeitkritischen Programmen tun, da die Übersichtlichkeit des Programms leidet.

**GOSUB ... RETURN**

**GOSUB <zeilennummer>**

Verzweigung zum Unterprogramm <zeilennummer>. RETURN schließt das Unterprogramm ab; der Programmablauf wird nach der GOSUB-Anweisung fortgesetzt.

**GOTO <zeilennummer>**

Verzweigung zur Zeile, die durch Zeilennummer gekennzeichnet ist.

**IF..THEN..ELSE**

**IF <ausdruck> THEN <anweisung:zeilennummer>**

[**ELSE <anweisung:zeilennummer>**]

Der Programmablauf verzweigt sich je nach Wert des logischen Ausdrucks. Ergibt der Ausdruck den Wert wahr (d. h. ungleich 0), so wird der THEN-Zweig der IF-Anweisung ausgeführt, andernfalls der ELSE-Zweig. Der ELSE-Zweig kann auch entfallen, dann wird,

falls der Ausdruck unwahr ist, die Programmausführung in der nächsten Zeile fortgesetzt. Als <ausdruck> sind numerische oder Vergleichsausdrücke erlaubt.

Einige Beispiele:

**IF A>=10 THEN A=B ELSE B=A**

**IF A>=10 AND A<100 THEN A=B:C=D**

**IF NOT(A=10) THEN 1000**

**IF A THEN A=0 ELSE 1000**

Vergleichsausdrücke sind auch in numerischen Ausdrücken verwendbar. Bei geschickter Programmierung kann man sogar auf IF-Anweisungen verzichten. Solche kompakten Programme sind in zeitkritischen Fällen vorteilhaft.

**PRINT A=B** gedruckt wird: 0 (unwahr) oder -1 (wahr)

**A=100-B>A**

**LET <variable>=<ausdruck>**

Der Wert eines Ausdrucks wird einer Variablen zugewiesen. LET muß bei diesem BASIC-Interpreter nicht verwendet werden und ist nur der Vollständigkeit wegen im Sprachumfang enthalten.

**LVAR**

**LLVAR**

listet alle verwendeten Variablen auf dem Bildschirm (LVAR) oder Drucker (LLVAR).

**ON..GOTO**

**ON <ausdruck> GOT <zeilenliste>**

**ON..GOSUB**

**ON <ausdruck> GOSUB <zeilenliste>**

Die Programmausführung wird in Abhängigkeit vom Wert des numerischen Ausdrucks verzweigt. Ergibt der Ausdruck den Wert 1, verzweigt das Programm zum ersten Element der Sprungzielliste, beim Wert 2 zum zweiten Element usw., beim Wert i zum i-ten Element. Ist kein Sprungziel vorhanden (z.B. wenn der Ausdruck den Wert 0 ergibt), verzweigt das Programm nicht, sondern die Programmausführung wird mit der der ON folgenden Anweisung fortgesetzt.

**10 ON A GOTO 100,200,300**

**20 PRINT "KEINE VERZWEIGUNG"**

**100 PRINT "A WAR 1"**

**200 PRINT "A WAR 2"**

**300 PRINT "A WAR 3"**

**READ <variablenliste>**

Wie INPUT, nur werden die Daten nicht von der Tastatur geholt, sondern aus einem durch DATA definierten Datenfeld. Mit der

ersten READ-Anweisung wird der erste Wert aus den DATA-Zeilen der ersten Variablen der <variablenliste> zugewiesen. Weitere READ-Anweisungen überführen das jeweils nächste Element der Konstantenliste zur entsprechenden Variablen. Siehe auch DATA und RESTORE.

10 DATA 1,2,3, "Ende"

20 READ A,B            A wird 1, B wird 2  
30 READ C            C wird 3  
40 READ D\$            D\$ = "ENDE"

**REM [<kommentar>]**

definiert die gesamte restliche Zeile zum Kommentar. Die durch REM eingeleiteten Kommentare werden bei der Programmausführung übersprungen. Wird anstelle REM das Zeichen ' geschrieben, so gilt der Kommentar nur bis zur nächsten Anweisung.

10 A=1: REM A WIRD 1 : B=1 WIRD  
NICHT AUSGEFUEHRT

20 A=1 'A WIRD 1 : B=1 'B WIRD 2

**REPEAT...UNTIL**

**REPEAT**

...  
**UNTIL <ausdruck>**

Die Anweisungen, die zwischen REPEAT und UNTIL stehen, werden so lange ausgeführt, bis der Ausdruck wahr (also -1 oder ungleich 0) wird. Verschachtelungen sind wie auch bei der FOR-Anweisung beliebig möglich.

**RESTORE [<zeilennummer>]**

Soll ein Datenfeld (DATA) erneut oder ab einer bestimmten DATA-Zeile gelesen werden, so muß man mit RESTORE den (internen) DATA-Zeiger zurück oder auf die entsprechende Zeilennummer setzen.

**RUN [[name,][zeilennummer]]**

startet ein Programm ab Anfang oder ab <zeilennummer>. Bei Angabe eines Namens wird erst ein LOAD <name> ausgeführt.

**STOP**

unterbricht die Programmausführung. Fortsetzung mit CONT.

**SWAP <variable>,<variable>**

vertauscht die Werte oder Zeichenketten der beiden angegebenen Variablen. Sortierprogramme können mit diesem Befehl besonders schnell arbeiten.

10 SWAP A,B ersetzt die folgenden Zeilen:

10 X=B

**11 B=A**

**12 A=X (X ist eine Hilfsvariable)**

**SYSTEM**

Der BASIC-Interpreter wird verlassen.

**TRACE n**

**LTRACE n**

TRACE 1 schaltet die Protokollierung der Programmausführung auf dem Bildschirm oder dem Drucker (LTRACE) an, TRACE 0 beendet den Vorgang.

**USR (I)**

ruft ein Maschinenprogramm auf. Dabei kann der Wert I übergeben werden. Die Adresse des Maschinenprogramms muß man wie folgt eintragen:

**POKE &131,LO** Low-Teil der Adresse

**POKE &132,HI** High-Teil der Adresse

Durch USR (I) wird das Unterprogramm angesprungen. Nun kann sich das Unterprogramm durch den Aufruf CALL 133H den Wert I holen.

Das Doppelregister DE enthält jetzt den Wert I. Ein Ergebnis kann an den Interpreter zurückgegeben werden:

**LD A,HI** High-Teil des Ergebnisses

**LD B,LO** Low-Teil des Ergebnisses

**CALL 136H** Übergabe an den Interpreter  
Mit RET gelangt man wieder zum Interpreter. Im Maschinenunterprogramm dürfen die CPU-Register HL und IY nicht zerstört werden (gegebenenfalls retten).

Ein kleines Programmbeispiel (es soll auf Adresse 8000H stehen) verdoppelt den Funktionswert (I):

8000 CALL 133 H ;Argument holen

8003 PUSH HL ;HL retten

8004 LD L,E ;HL:=DE

8005 LD H,D

8006 ADD HL,DE

8007 LD A,H ;zur Übergabe umladen

8008 LD B,L

8009 POP HL ;HL zurückholen

800A JP 136H ;Ergebnis übergeben und ;zurück zum Interpreter

Das BASIC-Programm:

10 POKE &131,0:POKE &132,&80

20 PRINT USR (10)

Nach RUN wird 20 ausgedruckt.

**WHILE...WEND**

**WHILE <ausdruck>**

...

**WEND**

Die Anweisungen zwischen WHILE und WEND werden so lange ausgeführt, bis der <ausdruck> unwahr wird. FOR- und REPEAT-Schleifen garantieren, daß die Anweisungen wenigstens einmal ausgeführt werden, während es bei WHILE-Schleifen dazu kommen kann, daß die Anweisungen gar nicht abgearbeitet werden.

### 2.9.2.3. Numerische Funktionen

Gegenüber anderen BASIC-Interpretern wurde der Befehlssatz um einige Funktionen und Befehle erweitert. Neue Funktionen sind z. B. LGT, RAD, DEG oder die Konstante PI.

#### ABS (X)

ermittelt den Absolutbetrag von X.

A=ABS(-2) ergibt 2

A=ABS(2) ergibt 2

#### ATN (X)

Der Arcustangens des numerischen Ausdrucks X wird berechnet. Die Berechnung ist in den Winkeleinheiten Altgrad oder Radian möglich. Siehe dazu auch DEG und RAD.

#### COS (X)

berechnet den Cosinus von X in den Winkel единиц Altgrad oder Radian.

#### DEG

berechnet trigonometrische Funktionen in Altgrad.

#### EXP (X)

berechnet die Exponentialfunktion  $e^x$ . X muß kleiner 87 bleiben.

#### FIX (X)

wandelt X in eine ganze Zahl ohne Runden um. Im Gegensatz zu INT liefert FIX nicht die nächstniedrige Zahl, wenn X negativ ist.

A=FIX(2.75) das Ergebnis ist 2

A=FIX(-2,75) das Ergebnis ist -2

#### FRAC (X)

entfernt den ganzzahligen Teil des Ausdrucks X.

PRINT FRAC (3.14) ergibt die Bildschirmausschrift 0.14

#### INT (X)

Die Integer-Funktion ermittelt den ganzzahligen Anteil von X. Siehe auch FIX.

A=INT(PI) ergibt 3

A=INT(-3,7) ergibt -4

#### LGT (X)

berechnet den dekadischen Logarithmus von X.

#### LOG (X)

berechnet den natürlichen Logarithmus des numerischen Ausdrucks X.

#### PI

Die reservierte Variable PI enthält die Konstante 3.1415926536.

#### RAD

berechnet trigonometrische Funktionen im Bogenmaß (Radian). Das ist auch die Standardeinstellung nach Start des Interpreters, bzw. nach Ausführung der Befehle NEW, RUN und CLEAR.

#### RANDMIZE

sorgt für einen wirklich zufälligen Beginn der Zufallszahlen, die durch RND erzeugt werden.

#### RND (X)

erzeugt eine Zufallszahl zwischen 0 und 1. Mit X wird gesteuert: Bei X < 0 wird der Zufallszahlengenerator neu initialisiert; bei X = 0 wird keine neue Zahl erzeugt. Beide Bedingungen nutzt man normalerweise als Programmtest. Wenn X positiv ist, arbeitet der Zufallszahlengenerator normal. Siehe auch RANDMIZE.

#### SGN (X)

ermittelt das Vorzeichen von X:

SGN(X) = 1 wenn X > 0

SGN(X) = 0 wenn X = 0

SGN(X) = -1 wenn X < 0

#### SIN (X)

berechnet den Sinus von X in der ausgewählten Winkeleinheit.

#### SQR (X)

berechnet die Quadratwurzel von X.

#### SQU (X)

berechnet das Quadrat von X.

A=SQR (SQU(X)+SQU(Y))

#### TAN (X)

berechnet den Tangens von X in der gewählten Winkeleinheit.

### 2.9.2.4. Zeichenkettenfunktionen

Der Interpreter hat neben den Standard-Zeichenkettenfunktionen auch einige sinnvolle Erweiterungen. In diesem Zusammenhang sei auf die Konvertierungsfunktionen BIN\$ und HEX\$ hingewiesen.

**ASC (X\$)**

Die Funktion gibt den ASCII-Code eines Zeichens bzw. des ersten Zeichens von X\$ zurück.

A=ASC ("BASIC") ergibt 66

**BIN\$ (I)**

erzeugt eine Zeichenkette, die den binären Wert des Ausdrucks I darstellt. Die Kette ist 16 Zeichen lang.

A\$=BIN\$(255) ergibt den String  
0000000011111111

**CHR\$ (I)**

wandelt den Wert I in ein Textzeichen entsprechend dem ASCII-Code um.

A\$=CHR\$(65) ergibt das Zeichen A

**HEX\$ (I)**

erzeugt eine Zeichenkette, die den Hex-Wert des Ausdrucks I darstellt. Die Kette ist 4 Zeichen lang.

A\$=HEX\$(2222) ergibt den String 56CE

**INKEY\$**

fragt die Tastatur ab. Ist eine Taste gedrückt, liefert die Funktion einen String der Länge 1 mit dem Zeichen, andernfalls ist die Kette leer.

10 A\$=INKEY\$ : IF A\$="" THEN 10

Im Beispiel wird so lange gewartet, bis man eine Taste drückt. Die INKEY\$-Funktionen wendet man an, wenn im Programm eine Tastaturabfrage realisiert werden muß oder wenn sonst nicht eingebbare Zeichen (z. B. CR) verarbeitet werden sollen.

**INSTR (X\$, Y\$, [I [,J]])**

Instring sucht im X\$ das erste Erscheinen des Strings Y\$. Dabei kann die Suche ab Position I beginnen und nach J Zeichen abgebrochen werden. Zurückgegeben wird die Position. Wird der String nicht gefunden, dann erscheint eine Null.

PRINT INSTR("ABCDE","CD") druckt 3

PRINT INSTR("ABCDE","CD",4) druckt 0

**LEFT\$ (X\$,I)**

Der Wert des numerischen Ausdrucks I bestimmt die Anzahl der Zeichen, die dem X\$ von links aus entnommen werden.

A\$=LEFT\$("AUTOMAT",4) ergibt AUTO

**LEN (X\$)**

bestimmt die Anzahl der Zeichen eines Textausdrucks.

A\$="AUTOMAT"

PRINT LEN(A\$) es wird 7 ausgedruckt.

**MID\$ (X\$,I,J)**

entnimmt dem String X\$ beginnend mit dem I-ten Zeichen genau J Zeichen.

A\$=MID\$("BEGINN", 4,2)

MID\$(X\$,I) wie RIGHT\$

**RIGHT\$ (X\$,I)**

Es werden die letzten I Zeichen des Textausdrucks X\$ zurückgegeben.

PRINT RIGHT\$("TEXTAUSDRUCK",8)  
druckt AUSDRUCK

**SPACE\$ (I)**

erzeugt eine Zeichenkette, bestehend aus I Leerzeichen.

**STR\$ (X)**

wandelt den Wert des numerischen Ausdrucks X in einen entsprechenden String. Die entstehende Zeichenkette entspricht der, die bei einer PRINT-Anweisung ausgegeben werden würde. Deshalb wirkt die Anweisung PRECISION auch auf die STR\$-Funktion.

A\$="PI BETRAEGT "+STR\$(PI)

STRING\$ STRING\$(I,J)

STRING\$(I,X\$)

erzeugt einen String der Länge I, deren Zeichen alle dem ASCII-Code von J oder dem ersten Zeichen von X\$ entsprechen. STRING\$(30,&20) und STRING(30, " ") sind gleichwertig und haben dieselbe Wirkung wie SPACE\$(30).

**VAL (X\$)**

konvertiert die als Zeichenkette (X\$) vorhandene Zahl in ihr numerisches Äquivalent.

A\$="2.2 cm"

A=VAL(A\$) A wird 2.2

A=VAL("&" + H\$) H\$ sei 1BCD, dann wird A zu 7117

### 2.9.2.5. Sonstige Funktionen und Anweisungen

**DEF FN**

DEF FN<name> (<parameterliste>)  
[=<ausdruck>]

benennt und definiert eine Anwenderfunktion mit <name>. <parameterliste> besteht aus Variablen, die lokal auf die Funktion beschränkt sind und die Parameterübergabe bewirken. <ausdruck> stellt die eigentliche Funktion dar. Beispiel:

10 DEF FNASN(X)=ATN(X/SQR(-X\*X+1))  
 In der Zeile 10 wurde eine Funktion zur Berechnung des Arcussinus definiert. Bevor die Funktion das erste Mal benutzt werden kann, muß die Zeile 10 einmal durchlaufen werden sein. Der Aufruf der Funktion sieht beispielsweise so aus:

PRINT FNASN(.5) oder A=FNASN(.1)

Neben rein numerischen Funktionen sind auch String- oder gemischte Funktionen möglich:

10 DEF FNKETT\$(X\$,Y\$)=X\$+" "+Y\$  
 20 DEF FNBYTE\$(I)=RIGHT\$(HEX\$(I),2)

In diesem Interpreter sind, obwohl in BASIC normalerweise nicht üblich, auch mehrzeilige Funktionsdefinitionen erlaubt. Mehrzeilige Funktionen können überdies auch rekursiv benutzt werden, d. h., daß sich die Funktion selbst aufrufen kann. Ein bekanntes (wenn auch in der Praxis nicht besonders sinnvolles) Beispiel zur rekursiven Berechnung der Fakultät soll dies belegen:

10 DEF FNFAC(N)  
 20 IF N<=1 THEN FNRETURN 1  
 30 FNEND N\*FNFAC(N-1)

Wie man sieht, fehlt in der Zeile 10 der <ausdruck>, was eine mehrzeilige Funktion kennzeichnet. Eine mehrzeilige Funktion muß durch FNEND abgeschlossen werden. Ein vorzeitiges Verlassen ist durch FNRETURN möglich.

**FNEND [<funktionswert>]**

stellt sowohl das logische als auch physische Ende einer Funktionsdefinition dar. Dabei wird der <funktionswert> übergeben. Siehe auch DEF FN.

**FNRETURN [<funktionswert>]**

Damit läßt sich eine definierte mehrzeilige Funktion vorzeitig, z. B. in Abhängigkeit eines Vergleichs, am einfachsten verlassen. FNRETURN wirkt wie FNEND, kann aber an beliebiger Stelle innerhalb der Funktionsdefinition stehen. Siehe auch DEF FN.

**FRE FRE(X)**

**FRE(X\$)**

Die erste Form gibt die Anzahl der freien Speicherplätze zurück. Die zweite Form gibt die Anzahl der freien Speicherplätze im Stringspace (Zeichenkettenspeicher) zurück. Bei umfangreichem Stringgebrauch kann die Funktion FRE (X\$) einige Sekunden dauern.

**INP (I)**

Mit INP kann ein byte vom Port I gelesen werden.

**OUT I,J**

gibt das byte J auf Port I aus.

**PEEK (I)**

liest die Speicherzelle I.

PRINT PEEK(&100) gedruckt wird 195

**POKE I,J**

Gegenteil von PEEK. Das byte J wird auf der Adresse I gespeichert.

**FOR I=&D000 TO &D7FF : POKE I,I AND &FF : NEXT**

wird einen Teil des Bildwiederholspeichers beschreiben.

**PRECISION n**

Damit läßt sich die Anzahl der Ausgabestellen einstellen. Wirkt auf PRINT und STR\$.

**WAIT I,J,K**

Warteschleife für Ports. Das Programm wird dann fortgesetzt, wenn der Wert am Port I mit K OR-verknüpft und mit J AND-verknüpft nicht mehr Null ergibt.

## 2.9.2.6. Ein-/Ausgabeanweisungen

Es gibt in dieser Interpreterversion mehrere Möglichkeiten Daten ein- oder auszugeben. Eine davon ist die Ein- oder Ausgabe über die Konsole, also Tastatur und Bildschirm. Eine weitere Möglichkeit stellt die Ein- bzw. Ausgabe mit einem externen Speichermedium dar. Da dies ein Kassettenrecorder ist, müssen einige Einschränkungen in Kauf genommen werden:

- Dateinummern haben hier keine Bedeutung. Man muß sie aber programmieren, damit der Syntax eingehalten wird.
- Die Funktion EOF ist ohne Wirkung und bringt immer -1, also wahr zurück.
- Eingabeschleifen sollen möglichst schnell laufen. Bei zu großen Programmlaufzeiten kann es sonst passieren, daß bestimmte Werte überlesen werden.

Diese Einschränkungen kann man teilweise beseitigen, wenn elektrisch steuerbare Kassettenlaufwerke zur Verfügung stehen und diese durch entsprechende Softwareänderungen an den Interpreter angepaßt werden. Der Befehlsumfang ergab sich aus dem Umstand, daß der Interpreter auch mit Betriebssyste-

men wie SCPX, BCU, CP/M, UDOS und MEOS arbeitet. Diese Betriebssysteme arbeiten mit Floppy-Disk als externes Speichermedium und ermöglichen einen Datenzugriff, der die oben genannten Einschränkungen von vorn herein ausschließt. Um entsprechende Erweiterungen an ein beliebiges Betriebssystem für den Amateur offen zu halten, wurden die Befehle sozusagen aufwärtskompatibel gestaltet. Im Abschnitt 2.9.4. werden diesbezügliche Hinweise gegeben.

**CLOSE [#]<zahl>[, [#]<zahl>...]**

Die angegebenen Files werden geschlossen. CLOSE ohne Spezifikation schließt alle Files. END und NEW schließen ebenfalls alle Files. (Wenn nicht anders installiert, schreibt CLOSE einen Dateiendeblock auf Magnetband.)

**CURSOR <x,y>**

Der Cursor wird auf die Bildschirmposition (x,y) bewegt.

**EOF (I)**

Test, ob das Ende der Datei I erreicht worden ist. Gibt wahr (-1) oder unwahr (0) zurück. (Ohne Installation durch den Anwender liefert die Funktion immer wahr.)

**INPUT [<"text">]; <variablenliste>**

weist Variablen Werte zu, die sich über die Tastatur eingeben lassen. Zur Unterstützung kann eine Nachricht für den Bediener ausgegeben werden (<"text">). In der Variablenliste stehen die Variablen, denen man Werte zuweisen soll. INPUT wird beendet, wenn die Variablenliste abgearbeitet ist.

10 INPUT "GIB A,B,C EIN ";A,B,C

20 INPUT D\$

Nach Ausgabe des Textes wird die Eingabe erwartet. Die Werte können entweder einzeln (jeden mit CR bestätigt) oder durch Komma getrennt eingegeben werden. Bei Strings ist zu beachten, daß bestimmte Zeichen Sonderfunktionen haben und somit nicht eingebbar sind (z. B.: ",). Stört das, so muß man LINE INPUT benutzen.

**INPUT# <I>,<variablenliste>**

Wie INPUT, nur werden die Daten nicht von der Tastatur, sondern von einem externen Speichermedium gelesen.

**LINE INPUT [<"text">]; <stringvariable>**

Es wird eine Zeile von der Tastatur geholt und der Stringvariablen zugewiesen. Im Ge-

gensatz zu INPUT lassen sich alle Zeichen eingeben.

**LINE INPUT# <I>,<stringvariable>**

Wie LINE INPUT, nur werden die Daten nicht von der Tastatur, sondern von einem externen Speichermedium gelesen.

**NULL I[,J]**

**LNULL I[,J]**

stellt die Anzahl der Nullzeichen ein, die nach jedem CR/LF ausgegeben werden. I ist die Anzahl und J der Code des Zeichens. Fehlt J, so wird der Code 0 angenommen. LNULL wirkt auf den Drucker. Im Beispiel soll auf dem Drucker ein linker Rand von 6 Zeichen eingestellt werden:

LNULL 6,&20

**OPEN <"modus">, [#]<dateinummer>,<name>**

Eine Datei, die unter der <dateinummer> verwaltet wird, wird mit dem Filenamen <name> für Eingabe (Modus "I") oder ausgabe ("O") eröffnet. (Wenn nicht anders durch den Anwender installiert, erzeugt OPEN einen Dateikopf auf Magnetband.)

**POS (I)**

**LPOS (I)**

Diese beiden Funktionen liefern die aktuelle Druckposition des Bildschirms (POS) oder des Druckers (LPOS). I ist ein Dummywert und sichert ab, daß der Syntax bei Funktionen eingehalten wird.

**PRINT [<ausgabeliste>]**

**LPRINT [<ausgabeliste>]**

Ausgabe von numerischen Werten und Texten auf dem Bildschirm (PRINT) oder Drucker (LPRINT). Die <ausgabeliste> enthält alle zu druckenden Werte, durch Komma oder Semikolon getrennt. Das Komma teilt die Ausgabezeile in 14 Spalten große Zonen auf, in denen numerisch Werte rechtsbündig und Texte linksbündig angezeigt oder gedruckt werden, während das Semikolon nur die Elemente der Liste trennt und keine Positionierung vornimmt. Neben Komma und Semikolon lassen sich die Befehle TAB und SPC zusätzlich zum Positionieren nutzen. Am besten eignet sich jedoch PRINT USING zur Ausgabeformatierung. Beendet man die <ausgabeliste> mit einem Komma oder Semikolon, so wird kein CR/LF ausgegeben, und es kann in der gleichen Zeile weitergedruckt werden. Eine PRINT-Anwei-

sung ohne Ausgabeliste bewirkt die Ausgabe einer Leerzeile.

**PRINT USING <zeilennummer>; <ausgabeliste>**

**LPRINT USING X\$; <ausgabeliste>**

Es werden Zahlen oder Zeichenketten in einem angegebenen Format gedruckt. Das Format wird in einer Zeile (und dort durch ! gekennzeichnet) oder in einem String spezifiziert. Zur Formatsteuerung benutzt man folgende Zeichen:

# Ziffernfeld

+ das Vorzeichen wird an der Stelle gedruckt, wo + steht

- wie +, nur werden positive Zahlen ohne Vorzeichen gedruckt

\*\* leere Positionen werden mit Sternchen gefüllt

\$\$ Ein \$ wird unmittelbar vor die erste Zahl gedruckt

\*\*\*\$ Kombination aus \*\* und \$\$

, an alle 3 Stellen wird ein Komma eingefügt (muß links vom Dezimalpunkt stehen).

~~~ Zahlen werden im Exponentialformat ausgegeben

leitet ein Stringformat ein. Das Format wird durch die folgenden Zeichen festgelegt:

L Linksangleich,

R Rechtsangleich,

C Zentrierung,

E Linksangleich mit Erweiterung, falls der String zu lang ist.

### Einige Beispiele:

**PRINT USING "#.#.#";PI** druckt 3.14

A\$="+.#+#~~~"

**PRINT USING A\$;PI** druckt +3.14E+0

10 !LLLLL'LLLLLL

20 PRINT USING 10;"abc";"cde"

**PRINT# <I>, [<ausgabeliste>]**

Wie PRINT, nur werden die Daten auf ein durch I gekennzeichnetes File (bzw. Gerät) ausgegeben. Die Ausgabeformate lassen sich auch hier durch USING einstellen.

**SPC (I)**

gibt, in PRINT angewendet, I Leerzeichen aus.

**SWITCH <I>**

verändert den Teil des JO-bytes, der für die Bildschirmausgabe verantwortlich ist. (Siehe

auch Beschreibung des Monitorprogramms.)

**TAB (I)**

erzeugt (in PRINT angewendet) bis zur Position I Leerzeichen.

10 FOR I=-6 TO 6

20 PRINT TAB(SQU (I));"\*\*"

30 NEXT I

**WIDTH <I>[.]**

**LWIDTH <I>[.]**

stellt die Zeilenbreite am Drucker (LWIDTH) oder Bildschirm (WIDTH) auf I ein. Nach der Ausgabe von I Zeichen wird automatisch ein CR/LF eingefügt. Ein nachgestelltes Komma unterdrückt die automatische CR/LF-Ausgabe. Die Zahlenausgabe in PRINT-Anweisungen wird in Abhängigkeit von WIDTH so gesteuert, daß keine Zahlen beim Zeilenwechsel «zerhackt» werden.

### 2.9.2.7. Grafikbefehle

Die Grafikbefehle ermöglichen in Verbindung mit den bereits beschriebenen Grafikunterprogrammen des Monitorprogramms (über Conout) das Erstellen von Grafiken.

**CLS [n]**

löscht den Bildschirm. CLS 0 wirkt wie CLS ohne Argument. CLS 1 löscht den Schirm weiß. (Achtung: Textausgaben sind nicht mehr sichtbar!)

**DRAW <x0,y0,x1,x2>**

Es wird eine Linie von den Koordinaten (x0,y0) nach (x1,y1) gezeichnet.

**DRAW 0,0,255,255** zeichnet eine Diagonale über den Schirm

**DRAWTO <x,y>**

zeichnet eine Linie nach (x,y).

**MOVE 20,20**

**DRAWTO 20,40**

**DRAWTO 40,40**

**DRAWTO 40,20**

**DRAWTO 20,20**

Mit diesem kurzen Programm läßt sich ein Viereck zeichnen.

**MOVE <x,y>**

Die Grafikkoordinaten werden auf (x,y) gesetzt.

**PAGE PAGE <adr1>, <adr2>**

Mit Page können mehrere Bildschirmzeiten erzeugt werden. Dieser Befehl ist sehr sy-

stemspezifisch. Vor seiner Anwendung sollte man sich über die jeweilige Speicheraufteilung im klaren sein. <adr1> ist die Adresse, ab der der Bildwiederholspeicher von der Hardware ausgelesen werden soll. <adr2> ist die Adresse des Bildwiederholspeichers, auf welchen die Grafikbefehle wirken. Normalerweise sind beide Adressen gleich. Das System initialisiert &C000, &C000.

#### *Ein Beispiel:*

Der Interpreter wurde mit einem maximalen Speicherplatz &7FFF gestartet.

```
PAGE &C000,&8000 'ab Adresse &8000
                  wird ein Bildspeicher
CLS              'erzeugt und (unsicht-
                  bar!) gelöscht
FOR I=1 TO 100   'es werden 100 zufäl-
                  lige Vektoren
DRAWTO RND(1)*255,RND(1)*255
                  '(unsichtbar) gezeichnet
```

NEXT

PAGE &8000,&8000 'und jetzt angezeigt  
Mit PAGE &C000, &C000 lässt sich dann wieder auf den alten Bildspeicher zurückschalten.

PEN n

Mit PEN wird der Schreibmodus ausgewählt:

- n=0 Penup; der «Schreibstift» ist inaktiv, alle Grafikbefehle wirken wie MOVE, d. h., sie zeichnen, ohne eine Spur auf dem Schirm zu hinterlassen.
- n=1 Pendown; der «Schreibstift» wird aktiv.
- n=2 Erase; die Grafikbefehle wirken löschend.
- n=3 Pen; die Grafikbefehle - wirken schreibend.

#### *Ein Beispiel:*

CLS 1 'der Schirm wird weiß gelöscht

PEN 2 'löschen schreiben

DRAW 0,0,255,255

PLOT 77,3

PLOT <x,y>

setzt oder löscht einen Punkt auf der Koordinate (x,y) (in Abhängigkeit vorhergegangener PEN-Befehle).

POINT (<x,y>)

Abfrage, ob ein Punkt mit den Koordinaten (x,y) auf dem Bildschirm gesetzt ist.

10 x=100: y=100

```
20 IF POINT(x,y) THEN PEN 2: PLOT x,y
                           ELSE PEN 3: PLOT x,y
```

#### 2.9.2.8. Befehle zur Programmspeicherung

Mit den folgenden Befehlen können Programme auf ein externes Speichermedium ausgelagert oder wieder eingelesen werden. Wenn, wie hier, ein Kassettenrecorder genutzt wird, gibt es auch bei diesen Befehlen gewisse Einschränkungen (siehe in diesem Zusammenhang auch Abschnitt 2.9.2.5.). Die ASCII-Befehle ALOAD, AMERGE usw. kann man nur bei geringen Datenübertragungsraten des Kassettenrecorders verwenden! Anmerkung: <name> ist eine Zeichenkette.

**ALOAD <name>**

lädt ein unter dem angegebenen Filenamen im ASCII-Code abgelegtes BASIC-Programm in den Rechner.

**AMERGE <name>**

mischt ein im ASCII-Code abgelegtes BASIC-Programm mit einem schon im Speicher vorhandenen Programm.

**LOAD <name>**

Das Programm, das mit SAVE und dem Filenamen <name> im interpreterinternen Format aufgezeichnet wurde, kann mit LOAD wieder geladen werden. Siehe auch RUN <name>.

**SAVE <name>**

Ein im Speicher vorhandenes Programm wird unter dem angegebenen Namen auf ein externes Speichermedium (Kassette) ausgelagert.

#### 2.9.2.9. Fehlerbehandlung des BASIC-Interpreters

Der verwendete BASIC-Interpreter ermöglicht eine komfortable Fehlerbehandlung, wie sie bei Heimcomputern im allgemeinen nicht üblich ist. Deshalb soll sie im folgenden näher beschrieben werden.

**ERR**

**ERL**

Beide sind reservierte Variablen. Sie werden

in Fehlerbehandlungsprogrammen verwendet.

- ERR enthält den Fehlercode.
- ERL enthält die Zeilennummer, in der der Fehler auftrat.

### **ERROR <I>**

simuliert das Auftreten eines Fehlers. I ist die Fehlernummer.

### **ON ERROR GOTO <zeilennummer>**

Bei Auftreten eines Fehlers wird in ein Fehlermaßnahmeprogramm ab <zeilennummer> abgearbeitet. Ohne Wirkung, wenn <zeilennummer> = 0. Siehe auch ERROR, ERR, ERL, RESUME.

### **RESUME RESUME**

RESUME NEXT

RESUME <zeilennummer>

Die Programmausführung wird nach einem Fehlerbehandlungsprogramm wie folgt fortgesetzt:

RESUME bei der Anweisung, die den Fehler verursacht hat.

RESUME NEXT bei der Anweisung, die der fehlerverursachenden unmittelbar folgt.

RESUME <zeile> bei der angegebenen Zeile.

Im Programmablauf können die verschiedensten Fehler auftreten, die einen Programmabbruch verursachen würden. Das wäre besonders ärgerlich, wenn eine mathematische Funktion punktweise berechnet und grafisch dargestellt werden soll und unerwarteterweise ein Fehler auftritt (Division durch 0 oder Überlauf). Die bisher berechneten Werte sind somit oft verloren, da sich das Programm nicht mehr fortsetzen lässt.

Auch Eingabefehler können zum Programmabbruch führen, wie das folgende einfache Beispiel zeigt:

10 INPUT «Gib zwei Zahlen ein»;A,B

20 C=A/B

30 PRINT C

40 GOTO 10

Normalerweise wird das Programm in der Schleife bleiben, nach jeder Eingabe die einfache Rechnung ausführen und das Ergebnis anzeigen. Gibt man B=0 oder gar nichts ein, kommt es in der Zeile 20 zum Fehler CAN'T /0, und das Programm wird abgebrochen. Um das zu verhindern, werden die folgenden Zeilen in das Programm eingefügt:

5 ON ERROR GOTO 100

### **100 REM FEHLERBEHANDLUNGSPROGRAMM**

### **150 RESUME 10**

Nach RUN wird in der Zeile 5 die Fehlerbehandlung «eingeschaltet». Tritt jetzt der oben beschriebene Fehler auf, so wird das Programm nicht abgebrochen, sondern zur Zeile 100 verzweigt. Die in der Zeile 150 stehende Anweisung RESUME 10 bewirkt den Rücksprung in die Zeile 10, und man kann die Eingabe wiederholen. Nun soll eine weitere Zeile in das Programm aufgenommen werden:

25 D=B\*B/A

Konnte bisher nur ein Fehler auftreten, wenn B=0 war, so führt jetzt auch eine falsche Eingabe von A zum Fehler. Welche der beiden Variablen war bei der Eingabe 0? Durch eine Erweiterung der Fehlerbehandlungsroutine ist das einfach feststellbar.

110 IF ERL=20 THEN PRINT "DIE VARIABLE B WAR 0"

120 IF ERL=25 THEN PRINT "DIE VARIABLE A WAR 0"

Zur Feststellung wird die reservierte Systemvariable ERL herangezogen, die die Fehlerzeile enthält. Eine weitere Variable ERR enthält den Code des Fehlers. Im Beispiel könnte das Behandlungsprogramm auch wie folgt aussehen:

110 IF ERR=11 AND ERL=20 THEN B=1  
E-38

120 IF ERR=11 AND ERL=25 THEN A=1  
E-38

### **150 RESUME**

Tritt wieder ein Fehler CAN'T /0 auf, lädt man die entsprechende Variable bei der Fehlerbehandlung mit einer sehr kleinen Zahl. Danach wird die entsprechende Division noch einmal ausgeführt (RESUME ohne Zeilennummer). Welche der beiden Variablen neu geladen wird, entscheiden die Zeilen 110 und 120, auch wenn es sich tatsächlich um eine Division durch Null (ERR=11) handelt. Trat beispielsweise ein arithmetischer Überlauf auf (in diesem Programm in der Zeile 25 bei der Multiplikation B\*B), so ist es wahrscheinlich günstiger, die Eingabe zu wiederholen. In Zeile 150 wird diese Entscheidung gefällt:

150 IF ERR=6 THEN RESUME 10 ELSE RESUME

Andere Fehler sollen zum gewohnten Programmabbruch führen:

140 IF ERR <> 6 AND ERR <> 11 THEN  
END

Die Fehlercodes sind in der Fehlerliste enthalten. Folgende Fehler kennt der BASIC-Interpreter:

- 1 NEXT W/O FOR
- 2 SYNTAX ERROR
- 3 RETURN W/O GOSUB
- 4 OUT OF DATA
- 5 ILLEGAL FUNCTION
- 6 ARITHMETIC OVERFLOW
- 7 OUT OF MEMORY
- 8 UNDEFINED STATEMENT
- 9 SUBSCHRIFT OUT OF RANGE
- 10 RE-DIMENSIONED ARRAY
- 11 CAN'T /0
- 12 ILLEGAL DIRECT
- 13 TYPE MIS-MATCH
- 14 NO STRINGSPACE
- 15 STRING TOO LONG
- 16 TOO COMPLEX
- 17 CAN'T CONTINUE
- 18 UNDEFINED USER CALL
- 19 FILE NOT FOUND
- 20 ILLEGAL EOF
- 21 FILES DIFFERENT
- 22 RECOVERED
- 23 FNRETURN W/O FUNCTION CALL
- 24 MISSING STATEMENT NUMBER
- 25 WEND W/O WHILE
- 26 UNTIL W/O REPEAT
- 27 RESUME W/O ERROR

19, 20, 28–31 sind für Erweiterungen reservierte Fehlercodes. Die Fehlermeldungen sind selbsterklärend, so daß sich eine Erläuterung erübrigkt.

Der BASIC-Interpreter ermöglicht auch die Selbstdefinition von Fehlern. Im oben benutzten Beispiel soll zusätzlich verhindert werden, daß die Variable A nie größer 100 wird. Dazu erweitert man das Programm noch einmal:

15 IF A>100 THEN ERROR 50  
130 IF ERR=50 THEN A=100: RESUME  
NEXT

Der Fehlercode 50 soll diesen Fehler ( $A>100$ ) kennzeichnen. Zeile 15 bewirkt durch den Befehl ERROR 50 eine Unterbrechung.

Selbstdefinierte Fehlercodes ermöglichen

sehr übersichtliche Programme, da ja die gesamte Fehlerbehandlung zentral in einem Programmteil durchgeführt werden kann. Das vermeidet vor allen Dingen bei umfangreichen Projekten verworrene Programmstrukturen. Der ERROR-Befehl steht aber nicht nur zur Fehlerbehandlung zur Verfügung. Er gestattet als «Software-Interrupt»-Quelle einige ganz neue Programmiertechniken, die bisher bei Kleincomputern ausgeschlossen waren. Natürlich lassen sich hier nicht alle Möglichkeiten beschreiben.

### 2.9.3. Das Programm

Auf den folgenden Seiten ist das Programm als Hex-Dump abgedruckt. Leider ist es bei einem so umfangreichen Programm, es ist etwa 14 Kbyte lang, nicht einfach eine zweckmäßige Form zur Veröffentlichung zu finden. Der Hex-Dump (Tabelle 2.9) ist eine Kompromißlösung. Aber diese Form ermöglicht es, den BASIC-Interpreter einem großen Interessentenkreis zugänglich zu machen. Um das «Eintippen» zu erleichtern, wird jede Zeile der Dumps mit einer Prüfsumme ergänzt. Diese Prüfsumme ist einfach die 16-bit-Summe der bytes dieser Zeile.

**Tabelle 2.9.** Hexdump des BASIC-Interpreters

|       |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |                          |                     |       |
|-------|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------------------|---------------------|-------|
| 0100: | C3 | 70  | 37 | C3 | 03 | F0 | C3 | 06 | F0 | C3 | 09 | F0 | C3 | 0C | F0 | C3                       | Cp7C.pC.pC.pC.pC.pC | =0917 |
| 0110: | 0F | F0  | C3 | 12 | F0 | C3 | 15 | F0 | C3 | 18 | F0 | C3 | 1B | F0 | C3 | 1E                       | .pC.pC.pC.pC.pC.    | =0906 |
| 0120: | F0 | C3  | 75 | 34 | C3 | 78 | 34 | C3 | 8F | 34 | C3 | 7B | 34 | C3 | 72 | 34                       | pCu4Cx4C.4C{4Cr4    | =082C |
| 0130: | C3 | 49  | 10 | C3 | 32 | 10 | C3 | C8 | 18 | 70 | 37 | FF | EF | 00 | 00 | 00                       | CI.C2.CH.p7.o...    | =0659 |
| 0140: | 00 | 43  | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .C.....             | =0044 |
| 0150: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0160: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0170: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0180: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | .....                    | =0OFF               |       |
| 0190: | FF | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0OFF |
| 01A0: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 01B0: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 01C0: | 00 | 00  | 00 | 80 | 00 | 80 | 00 | 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | ..0..               | =0180 |
| 01D0: | 00 | 00  | 00 | C3 | 09 | 01 | C3 | 03 | 01 | 00 | 20 | 0E | 00 | 00 | 01 | 00                       | ..C..C..            | =01C3 |
| 01E0: | 50 | 38  | 00 | 00 | 00 | 00 | FF | FF | FF | FF | 00 | 00 | 75 | 00 | 00 | PB.....U.                | =04F9               |       |
| 01F0: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0200: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0210: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | C3 | 70 | 37 | 00 | 00 | 00                       | Cp7..               | =016A |
| 0220: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 2C | 72 | 00 | 00 | 00 | 00 | 00                       | .,r..               | =009E |
| 0230: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0240: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0250: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0260: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0270: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0280: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0290: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 02A0: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 02B0: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 02C0: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 02D0: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 02E0: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 02F0: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0300: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0310: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0320: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0330: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00                       | .....               | =0000 |
| 0340: | 00 | 00  | 00 | 00 | 00 | 00 | 00 | 00 | AC | 27 | B6 | 28 | 1A | 32 | D7 | 27                       | ,6(.2W              | =02FB |
| 0350: | 30 | 01  | A0 | 18 | C7 | 1D | D9 | 18 | D4 | 18 | 0F | 35 | 34 | 2F | 34 | 30                       | 0..G.Y.T..54/40     | =0485 |
| 0360: | F4 | 25  | 8A | 2F | 70 | 30 | 7D | 30 | 00 | 31 | 17 | 31 | 73 | 24 | 38 | 32                       | t%/.pO)0.1.15\$82   | =0499 |
| 0370: | EB | 31  | E5 | 31 | CA | 31 | 9C | 31 | 45 | 1C | 62 | 1A | 1E | 54 | 1C | k1eIJ1.1E.b..T.          | =0583               |       |
| 0380: | 90 | 1C  | A9 | 1C | B9 | 1C | E9 | 1C | F2 | 1C | A5 | 32 | 65 | 1C | 45 | 33 ..).9.i.r.%2e.E3      | =0629               |       |
| 0390: | 4B | 33  | 1E | 34 | 21 | 1D | 79 | CC | 29 | 79 | 64 | 24 | 7C | 44 | 26 | 7C K3.4!.yL)y.\$!D&!     | =0511               |       |
| 03A0: | C8 | 26  | 7C | 13 | 32 | 7C | FB | 31 | 7F | 3D | 2F | 50 | 0E | 16 | 46 | 0D H&!.2!{1..P..F.       | =0509               |       |
| 03B0: | 16 | 46  | OB | 16 | C5 | 0F | 6C | OE | 74 | 14 | 13 | 11 | 71 | 12 | 77 | 13 ..F..E.l.t..q.w.      | =0384               |       |
| 03C0: | 8E | -13 | BD | 16 | DF | 13 | 2F | 11 | CF | 10 | C7 | 19 | 40 | 12 | 67 | 0F ..=..0.G.Q.g.         | =052D               |       |
| 03D0: | B2 | 10  | FO | 10 | 15 | 11 | C3 | 0F | DO | 1D | 1E | 12 | 01 | 10 | D6 | 1D 2.p...C.P....V.       | =04DB               |       |
| 03E0: | E0 | 18  | 7A | 24 | 79 | 12 | 77 | 10 | C7 | 19 | 49 | 36 | 15 | 11 | 11 | .z\$y.w.G.I6....         | =0453               |       |
| 03F0: | 76 | 12  | C5 | 20 | C2 | 20 | 7A | 22 | 39 | 1E | 55 | 1E | FE | 0F | 58 | 1E v..E B z"9.U.^..X.    | =0538               |       |
| 0400: | 94 | 22  | 91 | 22 | 11 | 09 | 62 | 24 | 2F | 24 | 00 | 24 | BD | 23 | 7C | 13 ..".b\$/\$..\$.=\$#!! | =03EF               |       |
| 0410: | A2 | 10  | 92 | 36 | D9 | 0A | 4C | 13 | F9 | 22 | 9C | 35 | A1 | 35 | E8 | 36 .."6Y.L.y".5!5h6      | =069C               |       |
| 0420: | 85 | 0D  | 82 | 0D | 80 | 1E | 8D | 20 | E8 | 20 | 4F | 32 | 65 | 32 | B8 | 32 .....                 | h 02e282            | =0576 |
| 0430: | 09 | 33  | 1A | 33 | 23 | 33 | 52 | 33 | 84 | 33 | 49 | 35 | 1B | 35 | C5 | 33 .3.3#3R3.3I5.5E3      | =03E1               |       |
| 0440: | D3 | 33  | E9 | 33 | 04 | 34 | 0C | 34 | 14 | 34 | 3E | 34 | 1E | 01 | EC | OF S313.4.4.4>4..1.      | =046E               |       |
| 0450: | 45 | 4E  | C4 | 46 | 4F | D2 | 4E | 45 | 58 | D4 | 44 | 41 | 54 | C1 | 50 | 52 ENDFORNEXTDATAPR      | =0689               |       |
| 0460: | 49 | 4E  | 54 | 20 | A3 | 49 | 4E | 50 | 55 | 54 | 20 | A3 | 49 | 4E | 50 | 55 INT #INPUT #INPU      | =053D               |       |
| 0470: | D4 | 44  | 49 | CD | 52 | 45 | 41 | C4 | 4C | 45 | D4 | 47 | 4F | 20 | 54 | CF TDIMREADLETGO TO      | =0708               |       |
| 0480: | 46 | 4E  | 45 | 4E | C4 | 49 | C6 | 52 | 45 | 53 | 54 | 4F | 52 | C5 | 47 | 4F FNENDIFRESTOREGO      | =0634               |       |
| 0490: | 20 | 53  | 55 | C2 | 52 | 45 | 54 | 55 | 52 | CE | 52 | 45 | CD | 53 | 54 | 4F SUBRETURNREMSTO       | =0644               |       |
| 04A0: | DO | 4F  | 55 | D4 | 4F | CE | 4E | 55 | 4C | 57 | 41 | 49 | D4 | 44 | 45 | POUTONNULLWAITDE         | =075E               |       |
| 04B0: | C6 | 50  | 4F | 4B | C5 | 50 | 52 | 49 | 4E | D4 | 43 | 4C | 45 | 41 | D2 | 46 FPOKEPRINTCLEARF      | =06AF               |       |
| 04C0: | 4E | 52  | 45 | 54 | 55 | 52 | CE | 53 | 41 | 56 | C5 | A1 | 45 | 4C | 53 | C5 NRETURNSAVE!ELSE      | =06A7               |       |

04D0: 4C 50 52 49 4E D4 54 52 41 43 C5 4C 54 52 41 43 LPRINTTRACELTRAC =058E  
 04E0: C5 E2 41 4E 44 4F 4D 49 5A C5 53 57 49 54 43 C8 ERANDOMIZESWITCH =0640  
 04F0: 4C 57 49 44 54 C8 4C 4E 55 4C CC 57 49 44 54 C8 LWIDTHLNNULLWIDTH =0653  
 0500: 4C 56 41 D2 4C 4C 56 41 D2 A7 50 52 45 43 49 53 LVARLLVAR'PRECIS =0623  
 0510: 49 4F CE 43 41 4C CC 45 52 41 53 C5 53 57 41 DO IONCALLERASESWAP =06AD  
 0520: 4C 49 4E C5 52 E5 CE 4C 4F 41 C4 4E 45 D7 41 55 LINERUNLOADNEWAU =06BD  
 0530: 54 CF 43 4F 50 D9 41 4C 4F 41 C4 41 4D 45 52 47 TOCOPYALOADAMERG =062B  
 0540: C5 41 E2 41 56 C5 4C 49 53 D4 4C 4C 49 53 D4 52 EASAVELISTLILSTR =06CB  
 0550: 45 4E 55 4D 42 45 D2 44 45 4C 45 54 C5 45 44 49 ENUMBERDELETEEDI =0593  
 0560: D4 44 45 C7 52 41 C4 57 48 49 4C C5 57 45 4E C4 TDEGRADWHILEWEND =0722  
 0570: 52 45 50 4B 41 D4 55 4E 54 49 CC 45 52 52 4F D2 REPEATUNTILERRO =0657  
 0580: 52 45 53 55 4D C5 4F 50 45 CE 43 4C F4 53 C5 43 RESUMEOPENCLOSEC =063C  
 0590: 4C D3 43 55 52 53 4F D2 44 52 41 D7 4D 4F 56 C5 LSCURSORDRAWMOVE =06E2  
 05A0: 50 4C 4F D4 50 45 CE E0 41 47 C5 53 59 53 54 45 PLOTPENPAGESYSTE =0657  
 05B0: CD 43 4F 4E D4 E5 53 49 4E C7 50 C9 54 41 42 AB MCONTUSINGPITAB( =071F  
 05C0: 54 CF 46 CE 53 50 43 E5 54 48 45 CE 4E 4F D4 53 TOFNNSPC(THENNOTS =0738  
 05D0: 54 45 DO AB AD AA AF 44 49 D6 4D 4F C4 DE 41 4E TEP+-\*/DIVMOD^AN =084A  
 05E0: C4 F4 D2 58 4F D2 BE E1 BC 53 47 CE 49 4E D4 46 DORXOR>=SGNINTF =08AE  
 05F0: 49 D8 41 42 D3 55 53 D2 46 52 C5 49 4E DO 50 4F IXABSUSRFRINPPO =0754  
 0600: D3 4C 50 4F D3 45 4F C6 53 51 D2 52 4E C4 4C 4F SLPOSEOFSQRNRNDLO =0760  
 0610: C7 45 E8 DO 43 4F D3 53 49 CE 54 41 CE 41 54 CE GEXPCCOSSINTANATN =07C9  
 0620: 50 45 45 CB 46 52 41 C3 4C 47 D4 53 51 D5 42 49 PEEKFRACLGTSQUBI =06AC  
 0630: 4E A4 48 45 58 A4 4C 45 CE 53 54 52 A4 56 41 CC N\$HEX\$LENSTR\$VAL =06DA  
 0640: 41 53 C3 53 50 41 43 45 A4 43 48 52 A4 4C 45 46 ASCSPACE\$CHR\$LEF =05BF  
 0650: 54 A4 E2 49 47 48 54 A4 4D 49 44 A4 49 4E 4B 45 T\$RIGHT\$MID\$INKE =05BF  
 0660: 59 A4 53 54 52 49 4E 47 A4 45 52 D2 45 52 CC 50 Y\$STRING\$ERRERLP =0694  
 0670: 4F 49 4E D4 49 4E 53 54 D2 00 00 00 00 00 00 4E DINTINSTR.....N =0418  
 0680: 45 58 54 20 57 2F 4F 20 46 4F D2 53 59 4E 54 41 EXT W/O FORSYNTA =04FC  
 0690: 5B 20 45 52 52 4F D2 52 45 54 E5 52 4E 20 57 2F X ERRORRETURN W/ =0508  
 06A0: 4F 20 47 4F 53 55 C2 4F 55 54 20 4F 46 20 44 41 0 GOSUBOUT OF DA =04C1  
 06B0: 54 C1 49 4C 4C 45 47 41 4C 20 46 55 4E 43 54 49 TAILLEGAL FUNCTI =04F8  
 06C0: 4F CE 41 52 49 54 48 4D 45 54 49 43 20 4F 56 45 ONARITHMETIC OVE =0511  
 06D0: 52 46 4C 4F D7 4F 55 54 20 4F 46 20 4D 45 4D 4F RFLOWOUT OF MEMO =0505  
 06E0: 52 D9 55 4E 44 45 46 49 4E 45 44 20 53 54 41 54 RYUNDEFINED STAT =0519  
 06F0: 45 4D 45 4E 54 A0 53 55 42 53 43 52 49 50 54 20 EMENT SUBSCRIPT =04F8  
 0700: F4 55 54 20 4F 46 20 52 41 4E 47 C5 52 45 2D 44 OUT OF RANGERE-D =04C2  
 0710: 49 4D 45 4E 53 49 4F 4E 45 44 20 41 52 52 41 D9 IMENSIONED ARRAY =050A  
 0720: 43 41 4E 27 54 20 2F B0 49 4C 4C 45 47 41 4C 20 CAN'T /OILLEGAL =0466  
 0730: 44 49 52 45 43 D4 54 59 50 45 20 4D 49 E5 2D 4D DIRECTTYPE MIS-M =0500  
 0740: 41 54 43 C8 4E 4F 20 53 54 52 49 4E 47 20 53 50 ATCHNO STRING SP =04F7  
 0750: 41 43 C5 53 54 52 49 4E 47 20 54 4F 4F 20 4C 4F ACESTRING TOO LO =04ED  
 0760: E4 C7 54 4F 4F 20 43 4F 4D 50 4C 45 D8 43 41 4E NGTOO COMPLEXCAN =0591  
 0770: 27 54 09 43 4F 4E 54 49 4E 55 C5 55 4E 44 45 46 'T.CONTINUEUNDEF =04DB  
 0780: 49 4E 45 44 20 55 53 45 52 20 43 41 4C CC 46 49 INED USER CALLFI =04CA  
 0790: 4C 45 20 4E 4F 54 20 46 4F 55 4E C4 49 4C 4C 45 LE NOT FOUNDILLE =04E4  
 07A0: 47 41 4C 20 45 4F C6 46 49 4C 45 53 20 44 49 46 GAL EOFILES DIF =04B4  
 07B0: 46 45 52 45 45 4E D4 52 45 43 4F 56 45 52 45 C4 46 FERENTRECOVEREDF =05A9  
 07C0: 4E 52 45 54 55 52 4E 20 57 2F 4F 20 46 55 4E 43 NRETURN W/O FUNC =046F  
 07D0: 54 49 4F 4E 20 43 41 4C CC 4D 49 53 53 49 4E 47 TION CALLMISSING =0510  
 07E0: 20 53 54 41 54 45 4D 45 4E 54 20 4E 55 4D 42 45 STATEMENT NUMBE =046C  
 07F0: D2 57 45 4E 44 20 57 2F 4F 20 57 48 49 4C C5 E5 RWEND W/O WHILEU =0563  
 0800: 4E 54 49 4C 20 57 2F 4F 20 52 45 50 45 41 D4 52 NTIL W/O REPEATR =04DF  
 0810: 45 53 55 4D 45 20 57 2F 4F 20 45 52 52 4F D2 44 ESUME W/O ERRORD =04E2  
 0820: 49 53 48 20 49 2F 4F 20 45 52 52 4F D2 46 49 4C ISK I/O ERRORFIL =04D3  
 0830: 45 20 4E 4F 54 20 4F 50 45 CE 46 49 4C 45 20 41 E NOT OPENFILE A =04A9  
 0840: 4C 52 45 41 44 59 20 4F 50 45 CE 46 49 4C 45 20 LREADY OPENFILE =04D3  
 0850: 41 4C 52 45 41 44 59 20 45 58 49 53 54 D3 45 52 ALREADY EXISTSER =0519  
 0860: 52 4F 52 A0 2A 49 4E 56 41 4C 49 44 20 49 4E 50 ROR \*INVALID INP =04CB  
 0870: 55 54 8A 20 40 20 4C 49 4E 45 A0 0A 52 45 41 44 UT. @ LINE .READ =04A1  
 0880: 59 8A 2A 45 58 54 52 41 20 4C 4F 53 54 8A 0A 2A Y.\*EXTRA LOST..\* =04B1  
 0890: 42 52 45 41 CB 00 16 FF 21 04 00 39 01 11 00 7E BREAK...!..9...~ =03EB

08A0: 23 FE 81 C0 7E 23 E5 66 6F 7A B3 EB 28 07 EB 7C #~. @~#efoz3k(.ki =086B  
 08B0: 92 20 02 7D 93 E1 23 C8 09 18 E4 CD E2 08 C5 E3 . .).a#H..dMb.Ec =07F4  
 08C0: C1 B7 E5 ED 52 C5 E3 C1 EB E3 03 ED BB 23 13 42 A7emREcAkC.m8#.B =09F3  
 08D0: 4B D1 C9 0E 03 E5 2A 9D 01 06 00 09 09 CD E2 08 KQI..e\*....Mb. =0572  
 08E0: E1 C9 D5 EB 21 DB FF 39 7C 92 20 02 7D 93 EB D1 aIUk!X.9!. .).kQ =0997  
 08F0: D0 1E 07 18 33 BE CA 48 OF 18 16 3A 8A 01 B7 20 P...3>JH...:..7 =04E9  
 0900: 0A C1 21 64 08 CD 5B 0D C3 29 08 2A 87 01 22 8F .A!d.M[.C).\*.. =04E7  
 0910: 01 1E 02 01 1E 0B 01 1E 0A 01 1E 12 01 1E 01 01 .....:..7D.5\* =02EC  
 0920: 1E 19 01 1E 1A 01 1E 1B 3A 0F 02 B7 C4 1D 35 2A .....:..7D.5\* =02EC  
 0930: C8 01 7C B5 28 2D 3A CA 01 B7 20 27 7B 32 CA 01 H..!5(-:J.7 '2J. =05CA  
 0940: D9 2A 8F 01 22 CB 01 23 7C B5 D9 28 16 EB 2A D1 Y\*..K.M!5Y!.k\*Q =06D2  
 0950: 01 22 CF 01 2A BB 01 22 CD 01 01 CE 0E C5 C3 D2 ."O.\*..M..N.ECR =05D0  
 0960: 10 1E 16 CD 07 0B CD 2D OB CD OB 13 D5 AF BB 20 ...M..M..M..U/; =0572  
 0970: 02 1E 05 3E 1F BB 30 0E 21 5E 08 CD 5B 0D 26 00 ...>;O.!^..M[.&. =035D  
 0980: 6B CD EE 29 18 13 21 7F 06 1D 28 0A CB 7E 23 28 kMn)...!...(.K~#( =0503  
 0990: FB 18 F6 1E 00 D5 CD 5B 0D 2A 8F 01 7C A5 3C F5 (.v..UMI.\*..%;<u =073D  
 09A0: C4 E6 29 E1 E3 3E 02 BD 20 OD F1 28 OB CD 0B 13 Df)ac...=q(.M.. =06D0  
 09B0: ED 5B 8F 01 C3 ED 20 D1 CD 2D OB CD 9C 0B 32 21 m[.Cm QM-.M..2! =0745  
 09C0: 02 CD FB 12 21 FF FF 22 8F 01 3A 21 02 B7 28 30 .Mx.!..".!..7(0 =0616  
 09D0: 2A 10 02 FA 2E 36 ED 5B 12 02 19 DA OD 18 22 10 \*..z.6ml...Z..". =0440  
 09E0: 02 E5 CD EE 29 D1 D5 CD BB 0A 3E 2A 38 02 3E 20 .eMn)QUMB.\*>8. > =0700  
 09F0: CD F4 OC CD B1 OC CD 4B OF D1 B7 28 BB 37 18 OA Mt.M1.MH.Q7(;7.. =073F  
 0AA0: CD B1 OC CD 4B OF 3C 3D 28 BA F5 FE 03 CA BD OF M1.MH.<(:u^..J.. =0765  
 0A10: D9 21 21 02 34 35 D9 CC 4E 10 FE 20 20 01 23 D5 Y!!45YLN..~ ..HU =05C0  
 0A20: CD AC OB 47 D1 F1 D2 1A OF D5 C5 23 7E B7 28 0C M,.GQqr.UE#~7(. =07AE  
 0A30: FE 20 28 F7 FE 09 28 F3 FE 0A 28 EF F5 CD B8 0A ~ (w~. (s~. (ouMB. =0902  
 0A40: C5 30 14 EB 2A 99 01 1A 02 03 13 7C 92 20 02 7D E0.k\*....;. .) =0497  
 0A50: 93 20 F4 ED 43 99 01 D1 F1 28 21 ED 4B 99 01 E1 . tmC..Qq(!mK..a =082F  
 0A60: 09 E5 CD BB 08 E1 22 99 01 EB 74 23 23 D1 73 23 .eM;.a"..kt#HQsM =0727  
 0A70: 72 23 11 29 02 1A 77 23 13 B7 20 F9 CD EB 0A 23 r#)..w#T. yMk.# =054D  
 0A80: EB 21 C4 09 E5 62 6B 7E 23 B6 C8 23 23 AF BE k!D.ebk~#6H#H#H/ > =0780  
 0A90: 23 20 FC EB 73 23 72 18 EC 11 00 00 D5 28 0C D1 # !ksMr.1...U(.Q =0621  
 0AA0: CD 82 OF D5 28 0E 3E D8 CD F5 08 11 FF FF C4 82 M..U(.>XMU...D. =089E  
 0AB0: OF C2 11 09 EB D1 E3 E5 2A 97 01 44 4D 7E 23 B6 .B..kQce\*..DM~#6 =0719  
 0AC0: 2B C8 23 23 7E 23 66 6F ED 52 60 69 7E 23 66 +HM~#fomR' i~#f0 =062D  
 0AD0: 3F C8 3F DO D9 19 D9 18 E2 C0 CD 1D 35 2A 97 01 ?H?PY.Y.bEM.5\*. =07JC  
 0AE0: AF 32 19 02 77 23 77 23 22 99 01 2A 97 01 2B 22 /2..wMwH..\*.+ =03FB  
 0AF0: 8B 01 2A 3B 01 22 83 01 CD 78 OF CD 65 32 2A 99 ..\*;..Mx.Me2\*. =0513  
 0B00: 01 22 9B 01 22 9D 01 C1 ED 7B 95 01 21 43 01 22 ..".."Am!..!C.. =04C5  
 0B10: 41 01 AF 67 6F 22 93 01 22 C8 01 22 CB 01 32 CA A./g0 ..H..K.2J =0552  
 0B20: 01 32 89 01 32 1D 02 E5 C5 2A 8B 01 C9 AF 32 3D .2..2..eE\*..I/2 =0555  
 0B30: 01 FD 21 09 01 FD 22 D4 01 FD 21 D9 01 C9 EB 21 .)!!..)"T.)!Y.Ik! =06EA  
 0B40: 31 0B E3 E5 EB FD 21 0F 01 FD 22 D4 01 FD 21 DF 1.cek!)!!."T.)!\_ =080E  
 0B50: 01 C9 EB 21 31 0B E3 E5 3E 4F 32 EC 01 CD 87 0B .Ik!1.ce)021.M.. =06E5  
 0B60: 11 24 01 ED 53 D4 01 FD 21 E5 01 C9 EB 21 9C 0B \$.mST.)!e.Ik!.. =06CB  
 0B70: E3 E5 3E 49 32 EC 01 3E 01 32 94 13 CD 87 0B 11 ce)I21..>.2..M.. =05F6  
 0B80: 21 01 ED 53 D7 01 C9 EB CA 11 09 3E FF 32 0E 02 !.msW.IkJ..2.. =0651  
 0B90: CD OE 1E 32 EB 01 3E 2C CD F5 08 C9 DD 21 03 01 M..2k.,Mu.I!).. =0616  
 0BA0: DD 22 D7 01 AF 32 3D 01 32 94 13 C9 AF 32 40 01 J" W./2=..2..I/2@. =05BA  
 0BB0: 0E 05 11 29 02 7E 47 B7 CA 6D 0C FE 21 38 6B FE ...).~G7Jm.~!8k~ =05CE  
 0BC0: 22 CA 5A OC 3A 40 01 B7 47 7E 20 5E FE 3F 3E 98 "JZ..@.76~ ^~?). =05DA  
 0BD0: 28 58 7E FE 30 38 04 FE 3A 38 4F D5 11 50 04 E5 (X~~~0B.~:80U.P.e =0646  
 0BE0: 18 02 23 13 1A B7 28 39 FE 20 20 09 13 7E FE 20 ..#..7(9~ ..~~ =0478  
 0BF0: 20 03 23 18 F8 EB 1A FE 60 38 02 E6 5F AE E6 7F ..#.xk.~'B.f\_.f. =074B  
 0CO0: EB 20 5D 1A 17 30 DB F1 78 FE 64 CB FF 38 1A D1 k J..0[qx~dK.8.Q =085C  
 0C10: EB 36 FF 78 D6 64 F6 80 OC 23 EB 23 12 13 OC 18 K6.xVdv..#k#.... =06CE  
 0C20: 94 E1 7E FE 60 38 02 E6 5F D1 EB FE 9D 36 3A 20 .a~~~'B.f\_Qk~.6! =08B7  
 0C30: 02 OC 23 EB 23 12 13 OC D6 3A 28 OA FE 6E 06 3A ..#k#...V:(.~n.: =045E  
 0C40: 28 11 FE 49 20 03 32 40 01 D6 56 28 05 D6 OC C2 (.~I .2@.VV(.V.B =0513  
 0C50: B5 0B 47 7E B7 28 16 B8 28 DO 23 12 OC 13 18 F3 5.G~7(.8(PH....s =0589  
 0C60: E1 E5 04 EB CB 7E 23 28 FB EB C3 E4 OB 21 28 02 ae.KK~#((kCd.!(. =082C

OC70: 12 13 12 13 12 C9 78 3D FD B6 00 28 34 10 02 18 .... Ix=)6.(4... =0413  
 OC80: 2D 2B CD FE OC 18 33 3E 3F CD 06 0D 3E 3F CD 06 -+M~..3>?P..>?M. =0527  
 OC90: 0D CD F2 OC CD B1 OC 23 7E 2B FE 03 C9 CD 31 00 .Mr.M1.#~+~.IM1. =0701  
 OCA0: F5 3E 5E CD 06 0D F1 F6 40 18 5B CD A0 OC CD 2B u>^M..qv@.IM .M+ =072C  
 OCBO: 13 21 29 02 06 01 78 32 1D 02 CD 67 0D FE 07 28 .!...x2..Mg.~. (=039D  
 OCC0: 20 FE 0D CA 06 13 FE 03 CA FE 12 FE 15 28 DC FE ~.J..~.J~.~.(\~ =07FE  
 OCD0: 08 28 A3 FE 0A 20 06 05 28 D7 04 18 04 FE 20 38 .(W~. ..(W...~ 8 =047B  
 OCEO: D9 4F 78 FE FD 3E 07 30 04 79 77 23 04 CD F4 OC YOx~)>.O.yw#.Mt. =06F8  
 OCFO: 18 C8 3E 20 FE 0A 20 0E CD OB 13 3E 0A C9 CD 04 .H>~. .M..>.IM. =0541  
 ODO0: 0D CD F2 OC 3E 08 C5 4F 3A 3D 01 B7 20 49 79 FE .Mr.>.EO:=.7 Iy~ =0641  
 OD10: 09 20 27 FD 7E 01 E6 F8 3D FD BE 00 38 17 FD 7E . .~.fx=)7.8.)~ =076C  
 OD20: 00 E6 07 2F CD 09 47 0E 20 CD D3 01 FD 34 00 10 .f..G.F. MS..)4.. =0542  
 OD30: F8 0E 09 18 22 CD OB 13 18 F7 FE 08 20 03 FD 35 x... "M...W~. .)5 =059E  
 OD40: 00 FE 20 38 OF FD 7E 00 FD 96 01 FD B6 05 CC OB ~.B.3~.)).)6.L. =0703  
 OD50: 13 FD 34 00 CD D3 01 79 C1 B7 C9 7E CB BF CD F4 .)4.MS.yA7I^K?Mt =0968  
 OD60: OC CB 7E CO 23 18 F4 CD D6 01 E6 7F FE 7F 28 F7 .K~@#.tMV.f.~.(W =08E9  
 OD70: 32 27 02 FE OF CO CD 9D 0C 3A 3D 01 2F 32 3D 01 2'..~.Q.M..:=./2=. =0485  
 OD80: AF C9 CD 45 OB C1 CD 99 0A C5 21 00 00 22 15 02 /IME.AM..E!..". =05E5  
 OD90: CD OB 13 E1 D1 4E 23 46 23 78 B1 28 51 CD 90 OF M..aQN#F#W1(QM.. =0685  
 ODAO: C5 4E 23 46 23 C5 E3 EB 7C 92 20 02 7D 93 C1 38 EN#F#Eck!. .).A8 =076B  
 ODB0: 3C E3 E5 C5 EE CD EE 29 E1 CD F2 OC CD F1 0D DD <ceEkMn)aMr.Mq. ] =0AEC  
 ODC0: 7E 00 B7 28 12 47 17 30 09 AF DD 77 00 DD 77 01 ~.7(.G.O./Jw.Jw. =055E  
 ODD0: 18 05 CD F2 OC 10 FB DD 7E 01 DD 77 00 01 90 0D ..Mr..(1~.Jw.... =0641  
 ODE0: C5 21 29 02 2B 06 00 CD 98 1A C3 DC 1A C1 C3 B8 E!).+.M..C\ AC8 =06B6  
 ODF0: 09 01 28 02 DD 21 15 02 3E E1 7E 03 B7 23 02 C8 ..(.J!..>a~.7W.H =048D  
 OE00: F2 FA 0D FE FF 20 06 7E 23 C6 64 18 3F FE 81 28 rz.~. ~#Fd.?~. (=07E5  
 OE10: 08 FE BD 28 04 FE BF 20 05 DD 34 01 18 29 FE 82 .~=.(.?.14..)~. =06A4  
 OE20: 20 17 E5 F5 CD 48 OF 28 OC FE 2C 20 F7 DD 35 00 .euMH. (.~, w15. =06BC  
 OE30: DD 35 01 18 EF F1 E1 18 08 FE BE 28 04 FE CO 20 15..oqa..~)>(.~@ =07D2  
 OE40: 06 DD 35 00 DD 35 01 FE 9D 20 01 0B D6 80 E5 21 .15.15.~. .V.e! =064E  
 OE50: 50 04 28 08 CB 7E 23 28 FB 3D 20 F8 7E FE 20 28 P. (.K~#((= x~~ (=062C  
 OE60: 08 B7 CB BF 02 FA F9 0D 03 23 18 F0 3E 29 32 89 .7K?,zy..#.p)>2. =069B  
 OE70: 01 CD 2F 11 E3 CD 98 08 D1 20 02 09 F9 EB 0E OA .M/.cM..Q ..yk.. =0656  
 OE80: CD D5 08 E5 CD 13 11 E3 E5 2A 8F 01 E3 3E D1 CD MU.eM..ce\*.c>QM =08C1  
 OE90: F5 08 CD C1 14 CD BE 14 E5 CD 0B 28 E1 C5 DD E5 05 u..MA.M) .eM.(aEje =09B8  
 OEA0: D5 01 00 81 DD 21 00 00 51 59 7E FE D6 3E 01 20 U...J!..QY~~V. =05B0  
 OEB0: OE CD 48 OF CD BE 14 E5 CD 0B 28 CD C8 27 E1 C5 .MH.M) .eM.(MH'aE =0818  
 OEC0: DD E5 D5 F5 33 E5 2A 8B 01 E3 06 81 C5 33 CD 12 JeUu3e\*..c..E3M. =089B  
 OED0: 01 3C CC 95 OF ED 73 D1 01 22 BB 01 7E FE 3A 28 .<L..msQ."..~~:( =066B  
 OEE0: 39 B7 C2 11 09 23 7E 23 B6 23 CA CD OF 5E 23 56 97B..#~#6#JM.~#V =05E6  
 OEF0: ED 53 8F 01 3A 19 02 B7 28 20 F5 FC 45 OB D5 E5 M\$...7( u!E.Ue =071F  
 OF00: 3E 3C CD 06 0D EB CD EE 29 3E 3E CD 06 0D E1 D1 ><M..kMn)>M..aQ =0737  
 OF10: F1 87 30 06 CD 31 OB 87 38 EO CD 48 OF 11 CE OE q.O.M1..B'MH..N. =0667  
 OF20: D5 C8 D6 80 DA 33 11 FE 7F 20 0A 23 7E FE AO CA UHV.Z3.~. .W~ J =08C1  
 OF30: 93 11 C3 11 09 FE 4E D2 11 09 11 B4 03 07 4F 06 ..C..~NR...4..0. =04DD  
 OF40: 00 E8 09 4E 23 46 C5 EB 23 7E FE AB 20 08 23 7E .k.N#FEKh~^( .~. =066B  
 OF50: FE 3A C8 B7 20 F8 FE 3A DO FE 20 28 E8 FE 0A 28 ~:H7 x~:P~( k~. (=0938  
 OF60: E7 FE 30 3F 3C 3D C9 28 OF CD 82 OF E5 CD B8 0A 9~?<=I(.M..eMB. =079F  
 OF70: D1 D2 EB 10 60 69 18 04 EB 2A 97 01 2B 22 9F 01 QRk.'i..k\*..+.. =061D  
 OF80: EB C9 2B CD 48 OF CD 4E 10 18 BE C1 C1 B7 18 3C kI+MH.MN..>AA7.< =0791  
 OF90: CD 12 01 3C CO CD 03 01 E6 7F 32 27 02 FE 14 20 20 M..<@M..f.2'.~. =059F  
 OFA0: OC E5 2A 8F 01 7C A5 3C C4 87 20 E1 C9 FE 13 20 .e\*..!%<D. ai~. =074E  
 OFB0: 00 CD 67 0D FE 03 28 04 FE 11 20 F5 FE 03 C0 CD .Mg.~. (~. u~. @M =072B  
 OFC0: 9D OC 3E CO F6 CO CC 1D 35 22 BB 01 C1 F5 2A 8F ..>QvGL.5" ..Au\*. =0798  
 OFD0: 01 7D A4 3C 28 09 22 91 01 2A 8B 01 22 93 01 CD .)\$.<(.~. \*..".M =047C  
 OFE0: 2D OB F1 21 8E 08 C2 93 09 C3 00 09 1E 11 2A 93 -.q!..B..CB..\*. =05AE  
 OFF0: 01 7C B5 28 56 EB 2A 91 01 22 8F 01 EB C9 CD 3E .15(Vk\*..".kIM) =06C8  
 1000: OB CD 0E 1E FE 32 30 41 FD 77 03 3E 2C BE CO CD .M..~?20A)w.,>@M =06D1  
 1010: 48 OF CD 0E 1E FD 77 04 C9 7E FE 41 D8 FE 5B 3F H.M..)w.I~?AX~[? =07BE  
 1020: C9 CD C1 14 18 OC CD 48 OF CD BE 14 CD C8 27 FA IMA...MH.M>.MH'z =0808  
 1030: 49 10 3A A6 01 FE 91 DA 7E 28 01 80 91 DD 21 00 I.:&..~.Z~(....)!!. =0659

1040: 00 11 00 00 CD 3F 28 51 C8 1E 05 C3 28 09 2B 11 ...M?(QH..C(.+. =03B1  
 1050: 00 00 23 7E B7 C8 FE 30 3F D0 FE 3A D0 E1 21 98 ..#~7H~0?P~!Pe!. =0803  
 1060: 19 B7 ED 52 DA 11 09 62 A1 19 29 19 29 D6 30 5F .7mRZ..bk.),)VO\_ =0589  
 1070: 16 00 19 EB E1 18 D1 28 26 CD 29 10 CD 49 OF CO ...ka.[(&M).MI.Q =0627  
 1080: E5 2A 3B 01 ED 52 E1 DA 11 09 2A 99 01 01 28 00 e\*;.mRkZ..\*...(. =0556  
 1090: 09 7C 92 20 02 7D 93 D2 F1 08 EB 22 95 01 E1 C3 .i. .).Rq,k..,aC =0758  
 10A0: EF OA CA EB OA FE 22 CA 0B 36 CD EF OA 01 CE OE o.Jk.^~J.6Mo..N. =0783  
 10B0: 18 1C OE 03 CD D5 08 C1 E5 ED 5B 8F 01 D5 7A B3 ...MU.AemL..Uz3 =076F  
 10C0: 3C 3A 1D 02 57 20 04 AF 32 1D 02 1E 8E D5 C5 CD <:.W ./2...UEM =0523  
 10D0: 4E 10 CD 15 11 E5 2A BF 01 7C 92 20 02 7D 93 E1 N.M..e\*..!. .).a =0611  
 10E0: 23 DC EB OA D4 BB OA 60 69 2B DE 1E 08 C3 28 09 #\\$.TB.'i+x..C(. =0640  
 10F0: CO CD 96 08 56 23 F9 FE BE 1E 03 20 F0 E1 22 8F @M..VH~y... pa\*. =07EC  
 1100: 01 7C A5 3C 21 1D 02 7E 72 20 04 B7 C2 4D 1A 21 .1%<!.~r~ .7B..! =04B3  
 1110: CE OE E3 01 3A OE 00 06 00 79 48 47 7E B7 C8 BB N.c.:...yHG~7HB =05CB  
 1120: C8 23 FE 22 28 F3 D6 8C 20 F2 BB 8A 57 18 ED FE HW~"(sV. r8.W.m~ =0936  
 1130: A0 28 60 CD C2 16 3E E2 CD F5 08 D5 3A 3F 01 F5 ('MB.)>bMu.U!?:u =07FB  
 1140: CD D3 14 F1 E3 22 8B 01 1F CD C3 14 20 07 E5 CD MS.qc...MC. .@M =07D2  
 1150: 1F 28 D1 E1 C9 E5 2A A1 01 E5 23 23 5E 23 56 21 .(QaIe\*!.eWW^WV! =0696  
 1160: 47 03 7C 92 20 02 7D 93 30 18 2A 95 01 7C 92 20 G.!. .).O.\*..!. =04C0  
 1170: 02 7D 93 D1 30 14 2A 99 01 7C 92 20 02 7D 93 30 .).QO.\*..!. .).O =055B  
 1180: 09 3E D1 CD 2F 1C EB CD 72 1A CD 2F 1C E1 CD 22 .>QM/.kMr.M..aM" =075C  
 1190: 28 E1 C9 CD 48 OF 3E 28 CD F5 08 CD C2 16 CD C2 (aIMH.)(Mu.MB.MB =085A  
 11A0: 14 E5 D5 D5 EB CD 72 1A CD 2F 1C E1 CD 22 28 D1 .eUUKMr.M..am" (Q =08C8  
 11B0: E1 3E 2C CD F5 08 D5 CD OE 1E B7 C4 49 10 F5 1E a>,Mu.UM..7J1.u. =07D0  
 11C0: FF 7E FE 29 28 08 3E 2C CD F5 08 CD OE 1E 3E 29 .~)(.,Mu.M..>) =0668  
 11D0: CD F5 08 3E E2 CD F5 08 F1 E3 3D BE 06 00 30 08 Mu.>bMu.qc=>..O. =07C1  
 11E0: 4F 7E 91 B1 47 38 01 43 C5 23 23 46 23 66 68 06 0~.jG8.CE#MFfh. =0524  
 11F0: 00 09 C1 E3 C5 CD CE 14 C1 D1 E5 2A A1 01 78 96 ..AcEMN.Aqe\*!x..x. =0872  
 1200: F5 78 38 01 7E 23 23 4E 23 46 CD 05 1C F1 38 09 ux8.~##NF.FM.~.qB. =0541  
 1210: 28 07 EB 36 20 23 3D 20 FA CD OF 1C E1 C9 FE C1 (.k6 #= zM..aI~A =074B  
 1220: CA 58 33 CD 0E 1E 7E 47 FE BE 28 06 3E EA CD F5 JX3M..~G~.().Mu =0757  
 1230: 08 2B 4B OD 78 CA 22 OF CD 83 OF FE 2C CO 18 F3 .+K.xJ".M..~,@.s =0652  
 1240: CD D3 14 7E FE 8A 28 06 3E D4 CD F5 08 2B E1 CD MS.~.().TMu.+eM =08A1  
 1250: C8 27 E1 28 09 CD 48 OF DA CF 10 C3 21 OF 16 01 H'a(.MH.ZO.C!... =05E8  
 1260: CD 13 11 B7 C8 CD 48 OF FE 9D 20 F4 15 20 F1 18 M..7HMMH.~.t. q. =0781  
 1270: E4 CD 52 0B 18 03 CD 3E 0B CA 0B 13 FE CE CA 46 dMR...M>.J..~NMF =0703  
 1280: 2D FE DO 28 4E FE D3 28 49 E5 FE 2C 2B 32 FE 3B ~-P(N~S(Ie~, (2~) =0855  
 1290: 28 5F C1 CD D3 14 E5 3A 3F 01 B7 20 1A CD 02 2A (\_AMS.e??.7..M.\* =0645  
 12A0: CD 95 1A 2A A1 01 FD 7E 00 86 FD BE 01 D4 0B 13 M..\*!.)~..).T.. =06F7  
 12B0: CD DC 1A CD F2 0C AF C4 DC 1A E1 CD 49 OF 18 B9 M..Mr./D..aMI..9 =08CE  
 12C0: FD 7E 00 FD B1 02 D4 OB 13 30 26 D6 0E 30 FC 2F )~.).O&V.0!! =06BF  
 12D0: 18 18 37 F5 CD OB 1E 3E 29 CD F5 08 2B F1 E5 3E ..7uM..>)Mu.+qe> =06C2  
 12E0: FF 38 04 FD 7E 00 2F 83 30 07 3C 47 CD F2 OC 10 B.~.~.O.<GMr.. =05FD  
 12F0: FB E1 CD 48 OF C8 18 81 21 7B 08 C3 1A 02 21 29 (aMH.H..!(&C..!) =062E  
 1300: 02 77 23 CD A0 0C 36 00 21 28 02 3E OD FD 77 00 .w#M .6!.().Jw. =0455  
 1310: CD 06 0D 3E 0A CD 06 0D FD 7E 03 3C 3D FD 77 00 M..>.M..~.(>)w. =0573  
 1320: C8 F5 FD 7E 04 CD 06 FD F1 18 F1 CD OB 13 3A 21 Hu~.M..q.QM..!: =075C  
 1330: 02 B7 C8 EB C5 2A 10 02 E5 CD EE 29 D1 CD BB OA .7He\*..eMn)QMB. =0890  
 1340: 3E 2A 38 02 3E 20 CD F4 OC C1 E1 C9 D1 11 0A 00 >\*8.> Mt.AaIQ... =0624  
 1350: D5 DC 82 OF EB E3 EB FE 2C 20 06 CD 48 OF DC 82 U\..kck~, .MH.\. =08CD  
 1360: 0F B7 C2 11 09 ED 53 12 02 7A B3 CA 11 09 3E 01 .7B..mS..z3J..>. =0546  
 1370: 32 21 02 E1 C3 DE 09 CD 6C OB 18 12 FE 85 20 08 2!.aC^.M1...~. . =05F9  
 1380: CD 48 OF CD 6C OB 18 05 3E 86 CD F5 08 F6 AF F5 MH.M1...>.Mu.v/u =07AD  
 1390: 7E FE 22 3E 00 32 3D 01 20 OD CD 96 1A 3E 3B CD ~~~>.2=. .M..;M =053C  
 13A0: F5 08 E5 CD DC 1A E1 E3 E5 CD 44 1A CD 8C 0C CA u.eM\..aceMD.M..J =09AB  
 13B0: 8B OF F1 28 2F E3 CD C2 16 CD C2 14 E3 D5 06 00 ..q/cMB.MB.CU.. =07CB  
 13C0: CD 98 1A EB 21 CC 13 E3 D5 C3 55 11 E1 CD 49 OF M..k!L..cUCU.aMI. =0851  
 13D0: C8 3E 2C CD F5 08 E5 CD 87 OC CA BC OF 18 D6 E5 H>,Mu.eM..J..>Ve =0879  
 13E0: 2A 9F 01 F6 AF 32 8A 01 E3 18 05 3E 2C CD F5 08 \*..v/2..c..>,Mu. =0660  
 13F0: CD C2 16 E3 D5 7E FE 2C 28 OC 3A 8A 01 B7 20 54 MB.cU~, (.:..7 T =0729  
 1400: CD 87 OC CA EB OF 3A 3F 01 B7 28 1A CD 48 OF 57 M..J..?..7(.MH.W =05B2

1410: 47 FE '22 28 05 16 3A 06 2C 2B CD 99 1A E█ 21 31 G~^(...,.,+M..,k!1 =04FE  
 1420: 14 E3 D5 C3 55 11 CD 48 OF CD E7 28 E3 CD 1F 28 .cUCU.MH.Mg(cM.( =07EC  
 1430: E1 CD 49 OF 28 05 FE 2C C2 FB 08 E3 CD 49 OF 20 aMI.(.~,B(.cMI. =074A  
 1440: AA D1 3A 8A 01 B7 EB C2 7D OF E6 21 B2 08 D5 C4 \*Q:..7kB).6!..UD =082A  
 1450: 5B OD E1 C9 CD 13 11 B7 20 10 23 7E 23 B6 1E 04 [.AIM..7 .W~#6.. =0586  
 1460: 2B 69 23 5E 23 56 ED 53 87 01 CD 48 OF FE 83 20 (.i#^WVmS..MH.~. =0618  
 1470: E3 C3 06 14 11 00 00 C4 C2 16 22 E█ 01 CD 98 08 cC....DB..".M.. =0588  
 1480: C2 1D 09 F9 D5 7E 23 F5 D5 CD F5 27 E3 E5 CD 8C B..yU~WuUMu'cEM. =0A2B  
 1490: 24 E1 CD 1F 28 E1 CD 0E 28 E5 CD 3F 28 E1 C1 90 \$aM. (aM. (eM? (aA. =0848  
 14A0: CD 75 28 ZB 09 EB 22 8F 01 69 60 C3 CA OE F9 2A Mu(.k..i'CJ.y.\* =06BF  
 14B0: BB 01 7E FE 2C C2 CE OE CD 48 OF CD 77 14 CD D3 ..~.,BN.MH.MW.MS =07EE  
 14C0: 14 F6 37 3A 3F 01 BF B7 E8 1E OD C3 28 09 CD D3 .v7:?.7h..C(.MS =0648  
 14D0: 14 18 EF 2B 16 00 D5 0E 01 CD D5 08 CD 4D 15 22 ..o+.+U..MU.MM." =053B  
 14E0: 8D 01 2A E█ 01 C1 78 FE 78 D4 C1 14 7E 16 00 D6 ..\*.Ax~xTA.~..V =0708  
 14F0: E1 38 15 FE 03 30 11 FE 01 17 AA BA 57 DA 11 09 a8.~.0.~..\*:WZ.. =0635  
 1500: 22 85 01 CD 48 OF 18 E7 7A B7 C2 3A 16 7E 22 B█ ..MH..9z7B:~.~. =0633  
 1510: 01 D6 D7 D█ FE OA DO 5F 3A 3F 01 3D B3 7B CA E█ .VWX~.P\_.?=.3(J) =082A  
 1520: 1B 07 83 5F 21 96 03 19 78 56 BA DO 23 CD C1 14 ...!...xV:PwMA. =05F4  
 1530: C5 01 E2 14 C5 ED 4B A1 01 C5 ED 4B A3 01 C5 ED E.b.EmK!.EmKw.Em =08AE  
 1540: 4B A5 01 C5 4E 23 46 C5 2A 85 01 18 89 AF 32 3F K%.EN#FE\*.../2? =05A3  
 1550: 01 CD 48 OF DA EC 28 CD 19 10 30 48 FE D7 28 ED .MH.Z1(M..OH~W(m =076B  
 1560: FE 2E CA EC 28 FE D█ 28 2A FE 22 CA 96 1A FE D5 ~.J1 (X~\*~J..~U =099F  
 1570: CA 9D 16 FE D2 D2 CA 16 19 FE 26 CA 7A 29 FE CF CA J..~RJ..~&Jz)~OJ =096E  
 1580: 41 32 3C 28 30 3E 28 CD F5 08 CD D3 14 3E 29 CD A2<(O>(Mu.MS.)M =061F  
 1590: F5 08 C9 16 7D CD D6 14 2A BD 01 E5 CD D█ 27 CD u.I.)MV. \*. .eM!`M =0849  
 15A0: C1 14 E1 C9 CD C2 16 E5 EB 22 A1 01 3A 3F 01 B7 A.aIMB.eK"!.:?..7 =07E9  
 15B0: CC F5 27 E1 C9 23 7E 06 00 17 4F C5 CD 48 OF 79 Lu'aIM~...DEMH.y =0701  
 15C0: FE 3B 38 36 FE 46 28 35 FE 48 28 31 FE 42 28 27 ~;86~F(5~H(1~B(1 =0676  
 15D0: FE 44 28 23 FE 4A 28 1F FE 4C 28 1B D2 49 10 3E ~D(W~J(.~L(.RI.) =0612  
 15E0: 2B CD F5 08 CD CE 14 3E 2C CD F5 08 E█ 2A A1 01 (Mu.MN.),Mu.K!\*. =07BC  
 15F0: E3 E5 E█ CD 0E 1E EB E3 18 08 CD 85 15 E3 11 9F cekM..kc..M..c.. =0894  
 1600: 15 D5 01 48 03 09 4E 23 66 69 E9 37 D2 F6 AF F5 .U.H..NWfii7Rv/u =070B  
 1610: CD 21 10 F1 E█ C1 DD E1 E3 EB CD FE 27 F5 CD 32 M!.qkAJackM~'uM2 =080D  
 1620: 10 F1 C1 79 21 C7 18 38 OC 20 05 A3 4F 78 A2 E9 .qAy!G.B..W0x!i =0699  
 1630: B3 4F 78 B2 E9 AB 4F 78 AA E9 21 4C 16 3A 3F 01 30x2i+0x\*!L.:? =0717  
 1640: 1F 7A 17 5F 16 64 78 BA DO C3 30 15 4E 16 79 B7 .z.\_.dx:PCO.N.y? =0627  
 1650: 1F C1 DD E1 D1 F5 CD C3 14 21 93 16 E5 CA 3F 28 .AjaQuMC.!..ej? =08E8  
 1660: AF 32 3F 01 D5 CD OF 1C D1 4E 23 46 23 C5 4E 23 /2?.UM..QN#F#EN# =05CF  
 1670: 46 C5 CD 13 1C CD 75 28 E1 E3 55 E1 7B B2 CB 7A FEM..Mu(acua(2Hz =08DA  
 1680: D6 01 D8 AF BB 3C DO 15 1D 0A BE 23 03 28 ED 3F V.X!/(<P...)>W..(m? =0699  
 1690: C3 D3 27 3C 8F C1 A0 C6 FF 9F C3 AF 27 16 5A CD CS'<.A.F..C/.Z.M =0923  
 16A0: D6 14 CD C1 14 CD 32 10 7B 2F 4F 7A 2F CD C7 18 V.MA.M2./Oz/MG. =06E9  
 16B0: C1 C3 E2 14 CD 49 OF C8 3E 2C CD F5 08 01 B4 16 ACb.MI.H>,Mu..4. =0766  
 16C0: C5 F6 AF 32 3E 01 46 CD 19 10 DA 11 09 AF 32 3F Ev/2?.FM..Z.../2? =062B  
 16D0: 01 4F DD 21 00 00 CD 48 OF 38 05 CD 19 10 38 23 .0!..MH.B..8..# =0400  
 16E0: 4F CD 48 OF 38 05 CD 19 10 38 18 DD 6F CD 48 OF 38 FB CD 19 B..M..B.JgMH.B(C. =0566  
 16F0: 38 05 CD 19 10 38 OC DD 67 CD 48 OF 38 FB CD 19 B..M..B.JgMH.B(C. =05F8  
 1700: 10 30 F6 D6 24 20 09 3C 32 3F 01 CB F9 CD 48 OF .0VV\$.<2?.KyMH. =05EF  
 1710: 3A 89 01 3D CA CE 17 86 D6 27 CA A2 17 AF 32 E█ :..=JN..V'J/.2. =0720  
 1720: 01 E5 ED 5B F█ 01 2A 99 01 18 09 23 23 23 23 23 .em!..\*...#### =045E  
 1730: 23 23 23 23 7C 92 20 02 7D 93 28 18 79 96 23 20 #####. ).(y..H =045E  
 1740: EA 78 96 23 20 E6 DD 7D 96 23 20 E1 DD 7C 96 23 jx..W fJ3..W aJ1..W =0847  
 1750: 20 DC 18 40 E1 E3 D5 11 A7 15 7C 92 20 02 7D 93 \.QacU..!..). =06FA  
 1760: D1 28 34 E3 E5 C5 01 OA 00 2A 9D 01 E5 09 C1 E5 Q(4ceE...\*..e.Ae =0721  
 1770: CD BB 08 E1 22 9D 01 60 69 22 9B 01 2B 36 00 7C M;.a..i..+6..! =0595  
 1780: 92 20 02 7D 93 20 F5 D1 73 23 72 23 DD 7D 77 23 . .). uQsHrN3)w# =06C9  
 1790: DD 7C 77 23 E█ E1 C9 32 A6 01 21 79 06 22 A1 01 !:w#kai2&.!y..!. =06C5  
 17A0: E1 C9 E5 2A 3E 01 E3 57 D5 C5 DD E5 CD 26 10 DD ale\*>.cWUE!eM&.J =096E  
 17B0: E1 C1 F1 E█ E3 E5 E█ 3C 57 7E FE 2C 28 EA 3E 29 aAqkcek<W~~,(j) =09E5  
 17C0: CD F5 08 22 8D 01 E1 22 3E 01 1E 00 D5 11 E5 F5 Mu."..a">...U.eu =069A  
 17D0: 2A 9B 01 3E 19 ED 5B 9D 01 7C 92 20 02 7D 93 28 \*...m{...}.( =056B

17EO: 31 7E B9 23 20 10 7E BB 23 20 0C 7E DD BD 23 20 1~9H .~8H .~J=H =059B  
 17FO: 07 7E DD BC 18 02 23 23 5E 23 56 23 20 D5 3A .~J<sub>C</sub>..WWH^HWU U: =04CA  
 1800: 3E 01 B7 C2 17 09 F1 CA 08 24 96 28 69 1E 09 C3 >.7B. qJ.%. (I..C =0500  
 1810: 28 09 11 06 00 F1 CA 22 24 71 23 70 23 4F DD 7D (... qJ \$q#p#01) =0519  
 1820: 77 23 DD 7C 77 23 CD D5 08 23 23 22 85 01 71 23 w#Jiw#MU.WH..WH =0589  
 1830: 3A 3E 01 17 79 01 0B 00 30 02 C1 03 71 23 70 23 :>..y...O.A.q#pH =0332  
 1840: F5 E5 CD CF 28 EB E1 F1 3D 20 EA F5 42 4B EB 19 ueMO(kaq= juBKk. =0A28  
 1850: 38 BB CD E2 08 22 9D 01 2B 36 00 7C 92 20 02 7D 8!Mb."..+6.!. .) =0578  
 1860: 93 20 F5 03 57 2A 85 01 5E EB 29 09 EB 2B 2B 73 . u.WK..^k). k++t =05E1  
 1870: 23 72 23 F1 38 26 47 4F 7E 23 16 E1 5E 23 56 23 #r#q8&GU~.a~WVH =052F  
 1880: E3 F5 7C 92 20 02 7D 93 30 83 E5 CD CF 28 D1 19 cui. . .o.MO(Q. =085E  
 1890: F1 3D 44 4D 20 E5 29 09 29 C1 09 EB 2A 8D 01 C9 q=DM e). I.A.K\*.I =0655  
 18AO: 2A 9D 01 EB 21 00 00 39 3A 3F 01 B7 28 OD CD OF \*. .k!..9?:?..7(.M. =044F  
 18BO: 1C CD 1B 1B 2A 95 01 EB 2A B3 01 AF 32 3F 01 ED .M..\*.k\*..//?..m =0586  
 18CO: 52 EB AF 06 98 18 OA 41 50 1E 00 21 3F 01 73 06 Rk/....AP..!?.s. =0435  
 18DO: 90 C3 B4 27 3A DF 01 18 03 3A D9 01 47 AF 18 EB .C4!~...Y.G/.h =066D  
 18EO: CD 52 1A CD 44 1A EB 73 23 72 EB 2B CD 48 OF 28 MR.MD.ks#Rk+MH. (=06B9  
 18FO: 07 FE E2 20 F7 C3 13 11 B7 20 OF 23 7E 23 B6 CA .~b wC..7 .WH#6J =070F  
 1900: 11 09 23 5E 23 56 ED 53 8F 01 CD 48 OF 28 E9 FE ..WH#VmS..MH. (i~ =0617  
 1910: 8B 28 E2 C3 OA 19 CD 52 1A 3A 3F 01 B7 F5 22 8D .(bc..MR.:?.7u". =0689  
 1920: 01 EB 7E 23 66 6F B4 CA 1A 09 7E FE 28 20 71 CD .k~Nfo4J..~~( qM =0705  
 1930: 48 OF 22 85 01 18 05 3E 2C CD F5 08 0E 05 CD D5 H. ....>,Mu..MU =0505  
 1940: 08 3E 29 32 89 01 CD C2 16 EB 3A 3F 01 B7 37 20 .>2..MB.k?:.77 =0543  
 1950: 10 4E 23 46 C5 23 4E 23 46 C5 23 4E 23 46 C5 18 .N#FE#N#FE#N#FE. =04E2  
 1960: 08 F5 D5 EB CD B8 1A D1 F1 E5 F5 EB 7E FE 29 20 .uUKMB.Qqeuk~~) =0AA8  
 1970: C6 2A 8D 01 3E 28 CD F5 08 E5 2A 85 01 CD C2 16 F\*..>(Mu.e\*.MB. =06E8  
 1980: E3 CD 3B 11 7E FE 29 28 OD 3E 2C CD F5 08 E3 3E cM;..~~( .>,Mu.c> =072B  
 1990: 2C CD F5 08 18 E7 CD 48 OF E3 3E 29 CD F5 08 3E ,Mu..gMH.c>)Mu. > =076B  
 19AO: D5 CD 49 OF 28 10 3E E2 CD F5 08 CD D3 14 CD 49 UMI. (.>bMu..MS.MI =07E6  
 19BO: 0F C2 11 09 18 38 0E 02 CD D5 08 ED 5B 8F 01 D5 .B..8..MU.m[.U =05A2  
 19CO: 16 D2 D5 33 C3 CE 0E 20 OC CD 17 25 32 7F 01 2F .RU3CN. .M.%2..// =05A5  
 19DO: 32 3F 01 18 09 CD D3 14 CD 49 OF C2 11 09 CD 96 2?..MS.M1.B..M. =05AB  
 19EO: 08 F9 FE D2 1E 17 C2 28 09 D1 ED 53 8F 01 3A 3F .y~R..B..QmS..:? =0713  
 19FO: 01 3C 28 03 3D 20 2E D1 F1 30 13 20 37 E1 C1 70 .<.= .QqO. 7aAp =0561  
 1A00: 2B 71 2B C1 70 2B 71 2B C1 70 2B 71 18 EA F5 D5 +q+Apt+q+Apt+q.juU =0758  
 1A10: 21 3F 01 CB 7E 28 01 77 B7 11 7F 01 C4 B8 1A E1 !?.K~(.w7...D8.a =0609  
 1A20: F1 1F C3 C3 14 ED 5B A1 01 CD 2F 1C 21 7F 01 CD q.CC.m[!..M/.!.M =071A  
 1A30: 22 28 18 C3 CD 2F 1C 7E 22 41 01 E1 77 23 23 71 "(.CM/.~"A.aw#Hq =052E  
 1A40: 23 70 18 B4 E5 2A 8F 01 23 7C B5 E1 CO 1E OC C3 Wp.4e\*..#;5a@..C =06E0  
 1A50: 28 09 3E D2 CD F5 08 F6 80 47 3E 29 32 89 01 C3 (.>RMu.v.G)2..C =06AE  
 1A60: C7 16 CD C1 14 CD 02 2A CD 95 1A CD OF 1C 01 B5 G.MA.#M..M..5 =06A2  
 1A70: 1C C5 7E 23 23 E5 CD F1 1A E1 4E 23 46 CD 89 1A .E~#H#Eq.a#HFM.. =076A  
 1A80: E5 CD 05 1C D1 C9 CD F1 1A 21 7F 01 E5 77 23 23 eM..QIMq.!..ew## =0788  
 1A90: 73 23 72 E1 C9 2B 06 22 50 0E FF 23 7E OC B7 s#rai+. "Pe..#~.7 =06AB  
 1AA0: 28 06 BA 28 03 B8 20 F4 FE 22 CC 48 OF E3 23 EB (.:(.B t~"LH.c#k =0713  
 1AB0: 79 CD 89 1A 11 7F 01 3E D5 2A 41 01 22 A1 01 3E yM..>UxA.!..> =04FB  
 1AC0: 01 32 3F 01 CD 22 28 7C 92 20 02 7D 93 22 41 01 .2?..M"(i..).A. =042E  
 1AD0: E1 7E CO 1E 10 C3 28 09 23 CD 95 1A CD OF 1C CD a~@..C(.#M..M..M =06A5  
 1AE0: 75 28 1C 1D C8 OA CD F4 0C FE OD CC 18 13 03 18 u(..H.Mt.^L.... =0592  
 1AF0: F2 B7 0E F1 F5 2A 95 01 EB 2A 83 01 4F AF 47 ED r7.qu\*..k\*..0/Gm =0828  
 1B00: 42 7C 92 20 02 7D 93 38 07 22 83 01 23 EB F1 C9 B1. .).B..#.WkqI =062F  
 1B10: F1 1E 0E 28 CO BF F5 01 F3 1A C5 2A 3B 01 22 83 q..(B?u.s.E!;. =0697  
 1B20: 01 AF 08 ED 5B 95 01 D9 21 43 01 ED 5B 41 01 7C ./.m[..Y!C.m[.A. =05DA  
 1B30: 92 20 02 7D 93 01 2B 1B 20 43 2A 99 01 ED 5B 9B . .).+. C\*.m[. =0515  
 1B40: 01 7C 92 20 02 7D 93 28 0C CB 7E 23 23 23 23 CD .!. .).(.K~WH#MM =0517  
 1B50: 7E 1B 18 E9 C1 EB 2A 9D 01 ED 52 EB 28 46 CB 7E ~..iAk\*.mRk(FK~ =07EF  
 1B60: 23 23 CD 75 28 E5 09 28 EB E3 4E 06 00 09 09 23 #HMu(e.(kcN..# =051D  
 1B70: D1 7C 92 20 02 7D 93 28 DC D5 01 70 1B C5 7E 23 Q1. .).(.U.p.E~# =06DC  
 1B80: 23 5E 23 56 23 23 C8 B7 C8 D5 D9 C1 2A B3 01 #HVV#H7HUYA\*. =06C7  
 1B90: ED 42 D9 D8 62 6B ED 42 D9 DO D9 50 59 D9 08 mBYXbkmBYPPYY. =09C1  
 1BA0: E5 DD E1 C9 08 DO D9 2A 83 01 EB 4F 06 00 09 2B eJai.PY\*..k0...+ =073F

1BBO: ED BB 62 EB 13 DD 72 FD DD 73 FC C3 1E 1B C5 E5 mBbk.Jr)Js|C..Ee =09C3  
 1BCO: 2A A1 01 E3 CD 4D 15 E3 CD C2 14 7E E5 2A A1 01 \*!.cMM..CMB..~\*!.. =0793  
 1BDO: E5 86 1E OF DA 28 09 CD 86 1A D1 CD 13 1C E3 CD e...Z(.M..QM..cM =078D  
 1BEO: 12 1C E5 2A 81 01 ■■ CD F5 1B CD F5 1B 21 E5 14 ..@\*.kMu..Mu.!@.. =077E  
 1BFO: E3 E5 C3 #4 1A E1 E3 4E 23 46 23 7E 23 66 6F 78 ceC4.acNWF#~Wfox =07E5  
 1COO: B1 C8 EU BO C9 60 69 4F 06 00 18 F3 CD C2 14 2A 1HmoI'10...SMB.\* =07D5  
 1C1O: A1 01 EB CD 2F 1C EB CO D5 50 59 1B 4E 2A B3 01 !.KM/.k@UPY.N\*.. =06E5  
 1C2O: 7C 92 20 02 7D 93 20 05 47 09 22 B3 01 E1 C9 2A 1. .). .G..~.aI\* =052F  
 1C3O: 41 01 2B ZB 2B 46 ■■ 4E 2B 2B 7C 92 20 02 7D 93 A.++F+N++!. .). =0418  
 1C4O: CO 22 41 01 C9 01 DC 18 C5 CD OC 1C AF 57 32 3F @".A.I.\.EM..W?2 =0613  
 1C5O: 01 7E B7 C9 01 DC 18 C5 CD 49 1C CA F8 1C 23 23 ..~?I.\.EMI.Jx..WW =070F  
 1C6O: 5E 23 56 1A C9 E1 3E 28 CD F5 08 CD OE 1E D5 3E ^~W.U.Ia>(Mu..Mu..U) =06D7  
 1C7O: 2C CD F5 08 CD D3 14 3E 29 CD F5 08 E3 E5 E5 3A ,Mu.MS.)>Mu.cee: =08C2  
 1C8O: 3F 01 B7 28 05 CD 58 1C 18 03 CD 11 1E D1 18 05 ?.? (.MX...M..Q.. =046A  
 1C9O: CD 11 1E 3E 20 F5 7B CD 86 1A 47 F1 04 05 28 15 M..> u(M..Gq..(. =0585  
 1CAO: 2A B1 01 77 23 10 FC 18 0C 3E 01 CD 86 1A CD 11 \*.~.W.H!..).M..M.. =0500  
 1CBO: 1E 2A 81 01 73 C1 C3 B4 1A CD F2 1D AF E3 4F 3E ..\*.sAC4.Mr./cO> =078A  
 1CCO: E5 E5 7E B8 38 02 78 11 0E 00 C5 CD F1 1A C1 E1 ee~BB.x..~EMq.Aa =0810  
 1CDO: E5 23 23 46 23 66 68 06 00 09 44 4D CD 89 1A CD eHHFHfh..DMM..M =053F  
 1CEO: 05 1C D1 CD 13 1C C3 B4 1A CD F2 1D D1 D5 1A 90 ..GM..C4.Mr.QU.. =07AB  
 1CFO: 18 CB EB 7E D1 43 04 05 CA 49 10 C5 1E FF FE 29 .Kk~QC..JI.E..~) =0795  
 1D0O: 28 08 3E 2C CD F5 08 CD OE 1E 3E 29 CD F5 08 F1 (.>,Mu..M..)>Mu.q =067F  
 1D1O: E3 3D HE 06 00 30 AA 4F 7E 91 EH 47 38 A3 43 18 c==..~.0\*~. ;G8#C. =0654  
 1D2O: AO E1 3E 28 CD F5 08 CD CE 14 ED 5B A1 01 D5 3E a>(Mu..MN.m!..U..~ =085D  
 1D3O: 2C CD F5 08 CD CE 14 ED 5B A1 01 D5 01 FF 00 7E ,Mu..MN.m!..U..~ =07E2  
 1D4O: FE 2C 20 15 C5 CD OB 1E C1 1D 43 1C 28 AA 7E FE ~, .EM..A.C. (\*~ =06A5  
 1D5O: 2C 20 06 C5 CD OB 1E C1 4B 3E 29 CD F5 08 E3 C5 , .EM..AK)>Mu..cE =06F2  
 1D6O: CD 12 1C C1 EB E1 E3 D5 C5 CD 12 1C C1 D1 78 96 M..AkacUEM..AQx. =09AO  
 1D7O: D2 C4 1D ED 44 B9 30 01 4F 23 23 7E 23 66 6F E5 RD.mD90.0#~Wfoe =06BE  
 1D8O: C5 48 06 00 09 C1 EB 79 96 38 38 3C 4F 46 23 23 EH...Aky.B8<OF# =055E  
 1D9O: 7E 23 66 6F EB 78 B7 28 1D C5 06 00 1A ED B1 79 ~#folkx7(.E...mly =06D1  
 1DAO: C1 20 20 4F C5 D5 E5 18 06 13 1A BE 20 03 23 10 A QEUe....> .W.. =052E  
 1DBO: F8 E1 D1 C1 20 09 D1 ED 52 7D CD DC 18 E1 C9 79 xaQA .QmR)M\..aIy =0A05  
 1DCO: B7 20 D6 D1 18 F3 CD 11 1E 4F ED 78 C3 DC 18 7 VQ/.sM..OmxC\.. =089F  
 1DDO: CD FD 1D ED 79 C9 CD FD 1D 47 C5 1E 00 CD 49 OF M)..myIM).GE..MI.. =084C  
 1DEO: 28 08 3E 2C CD F5 08 CD OE 1E C1 ED 78 AB AO 28 (.>,Mu..M..Amx+ { =06F6  
 1DFO: FA C9 ■■ 3E 29 CD F5 08 C1 D1 C5 43 C9 CD OE 1E zIK>Mu.AQECIM.. =093B  
 1E0O: D5 3E 2C CD F5 08 CD OE 1E C1 C9 CD 48 OF CD BE U>,Mu..M..AIMH.M> =083B  
 1E1O: 14 CD 2C 10 7A B7 C2 49 10 CD 49 OF 7B C9 CD 49 19 .M..z7B1.MI..(IMI =06E8  
 1E2O: 1C CA 17 25 5F 23 23 7E 23 66 6F E5 19 46 72 E3 .J..~.WW~Wfoe.Frc =05D6  
 1E3O: C5 7E CD E7 28 C1 E1 70 C9 38 09 CD 15 01 EE 03 E~Mg(AapIB..M..n.. =080F  
 1E4O: 4F C3 18 01 CD OE 1E FE 04 D2 11 09 47 CD 15 01 DC..M..~.R..GM.. =053C  
 1E5O: E6 FC B0 18 EB CD 3E OB CD OE 1E FE OE DA 49 10 f10.KM>.M..~.ZI.. =07E3  
 1E6O: FD 77 01 4F D6 OE 30 FC C6 1C ED 44 81 FD 77 02 )w.OV.O!F.mD.)w.. =07DE  
 1E7O: 7E FE 2C 3E 00 20 05 CD 48 OF 3E 01 FD 77 05 C9 ~~, > ..MH.>.)w.I =05B0  
 1E8O: 08 EB 2A 8F 01 7C A5 3C C2 49 10 2A 97 01 7E 23 .k\*..!~BI..\*.~# =0588  
 1E9O: B6 EB CA 15 11 11 0A 00 D5 08 DC 82 OF ED 53 17 6kJ...U..~.MS.. =064D  
 1EAO: 02 D1 FE 2C 20 06 CD 48 OF DC 82 OF ED 53 15 02 .Q~, .MH..~.mS.. =060B  
 1EB0: E5 2A 97 01 23 23 5E 23 56 E1 FE 2C 20 0E CD 48 e\*..~.WW~WVa~, .MH =0612  
 1ECO: OF DC 82 OF 2A 17 02 ED 52 DA 11 09 ED 53 22 02 .~..\*.mRZ..mS.. =0556  
 1EDO: B7 C2 11 09 CD ■■ OA D2 EB 10 60 69 D9 11 01 00 7B..M8.Rk.'iY... =06A3  
 1EEO: 21 00 00 D9 11 FF FF CD ■■ OA D9 28 E3 ED 4B 15 !..Y...M..Y+kmK.. =07D7  
 1EFO: 02 78 B1 CA 11 09 CD CF 28 ED 5B 17 02 19 DA OD .x1J..MO(m!..Z.. =0634  
 1FOO: 18 2A 97 01 23 23 4E 23 46 EB 2A 22 02 ED 42 38 .~..##NNFk\*.mB8 =0477  
 1F1O: 06 28 0E 60 69 18 OD ED 4B 15 02 2A 10 02 09 18 .('..mK..\*.... =02D6  
 1F2O: 03 2A 17 02 22 10 02 EB CD 48 OF B7 CA 52 20 FE \*.~..~.kMH.7JR ~ =057A  
 1F3O: 8A 28 31 FE 8E 28 2D FE D4 28 29 FE 9D 28 25 FE .(1~.(-~T()~.(%~ =07CD  
 1F4O: CE 28 21 FE 8D 28 1D FE C2 28 19 3C 20 DA 23 7E N(!~. (.~B.(.Z~ =06BF  
 1F5O: FE A4 20 D4 FE 2C 28 DO FE 3B 28 CC CD 48 OF 28 ~\* T~, (P~; (LMH..( =0831  
 1F6O: G7 30 F1 2B CD 48 OF 30 C2 22 12 02 2B 11 00 00 Goq+MH.0B..+... =049B  
 1F7O: OE 00 CD 48 OF 30 2B E5 B7 21 98 19 ED 52 30 11 ..MH.0+e7!..mRO. =057B

1F80: 21 F6 06 CD 5B 0D CD 84 20 E1 CD 48 OF 38 FB 18 !v.MI.M. aMH.B(. =0713  
 1F90: 9A 62 6B 29 29 19 29 D6 30 5F 16 00 19 EB E1 OC .bk)).)VO\_\_\_.ka. =0567  
 1FA0: 18 D0 79 32 14 02 E5 2A 22 02 EB E5 ED 52 38 2A .Py2..e\*\*.kemRB\* =064D  
 1FB0: CD B8 OA 60 69 D1 D9 2A 17 02 ED 5B 15 02 D9 CD MB.'iQY\*.mL..YM =074A  
 1FC0: BB OA 38 13 21 E2 06 D5 CD 5B 0D E1 CD EE 29 CD ;.8.!b.UM(.aMn)M =07B5  
 1FD0: 84 20 E1 7E C3 2B 1F D9 E5 D9 D1 E1 AF 06 98 CD . a~C+.YeYQa//..M =0973  
 1FE0: B4 27 CD 02 2A 06 00 23 E5 7E B7 28 04 04 23 18 4'M.\*..#e~7(..M. =0482  
 1FF0: F8 3A 14 02 90 28 3C 38 1C 4F 06 00 2A 12 02 54 x:...(<8.D..\*.T =0377  
 2000: 5D 09 E5 2A 99 01 ED 52 ED 42 44 4D E1 ED B0 ED 1.e\*\*.mRmBDMamOm =0879  
 2010: 53 99 01 18 1E ED 44 4F 06 00 2A 99 01 54 5D 09 S....mDO..\*.TJ. =0427  
 2020: CD E2 08 22 99 01 EB E5 ED 4B 12 02 ED 42 44 4D Mb.. kemK..mBMD =074F  
 2030: E1 ED B8 D1 2A 12 02 1A B7 28 05 77 23 13 18 F7 amBQ%..7(.w..w =064F  
 2040: E5 2A 97 01 EB CD 85 OA E1 7E FE 2C CA 64 1F C3 e\*\*.kM..a~^,Jd.C =0887  
 2050: 2B 1F 23 B6 23 B6 2B C2 04 1F ED 5B 22 02 CD B8 +.H6H6+B..mL..MB =05FD  
 2060: OA 60 69 ED 4B 15 02 ED 5B 17 02 23 23 73 23 72 .'imK..m[..#W\$Wr =04D1  
 2070: 23 7E B7 20 FB EB 09 EB 23 B6 23 B6 20 EE CD EF #~7 (k.k#6H6 nMo =08CE  
 2080: OA C3 B8 09 2A 10 02 CD E6 29 C3 OB 13 CD 99 OA .CB.\*..MF)C..M.. =05F7  
 2090: D1 C5 CD B8 OA D2 49 10 54 5D E3 E5 7C 92 20 02 QEMB.RI.Tjce. . =07F9  
 20A0: 7D 93 D2 49 10 CD F8 12 C1 21 7C OA E3 EB 2A 99 J.RI.Mx.A!..ck\*. =080B  
 20B0: 01 1A 02 03 13 7C 92 20 02 7D 93 20 F4 ED 43 99 .....i. .). tmC. =0550  
 20C0: 01 C9 3E 7F 01 3E BF F5 CD D3 14 CD 49 OF 20 14 .I>..>?uMS.MI. . =0687  
 20D0: CD C8 27 3D 2F 47 F1 4F 2F AO 47 3A 19 02 A1 BO MH'=/Gq0/ G...!O =066B  
 20E0: 32 19 02 C9 F1 C3 11 09 CD 82 OF CO E1 CD B8 OA 2..IqC..M..@aM. =0772  
 20F0: D2 49 10 60 69 23 23 4E 23 46 23 C5 CD F1 OD E1 RI .'i#NNMF#EMq.a =0685  
 2100: E5 CD EE 29 CD F2 OC CD E1 OD CD OB 13 E1 E5 CD @Mn)Mr.Ma.M..aeM =09CD  
 2110: EE 29 CD F2 OC 21 29 02 E5 1E FF 1C 7E E6 7F 77 h)Mr.!).e...~f.w =07A6  
 2120: 23 20 F8 E1 57 06 00 CD 67 OD FE 3A 30 10 FE 30 # xaW..Mg.~:0.~0 =0660  
 2130: 38 0C DC 30 4F 78 07 07 80 07 81 47 18 E9 E5 21 8.VOOx....G.ie. =0575  
 2140: 25 21 E3 05 04 20 01 04 D9 21 69 21 BE 23 4E 23 %!c.. ..Y!i!>NM =042D  
 2150: 46 23 C5 D9 CB D9 C1 34 35 20 F1 FE 60 38 04 E6 F#EHYHA45 q~^B.t =0863  
 2160: 5F 18 E6 D9 3E 07 C3 06 0D 20 9C 21 51 CC OF 4C \_..fy>..C. .!QL.L =05A6  
 2170: D4 21 46 AD 21 49 0E 22 44 DE 21 0D 5B 22 52 F7 T!F-!I."D^!.I."RW =0598  
 2180: 21 45 SE 22 58 09 22 4B A7 21 48 06 22 08 50 22 !E^"X."K'!H..P. =0366  
 2190: 41 94 21 00 C1 D1 CD OB 13 C3 ED 20 7E B7 C8 14 A..!AQm..Cm ~Z.H. =0754  
 21A0: CD F4 OC 23 10 F6 C9 E5 21 F2 21 E3 37 F5 CD 67 Mt..W.vie!r!c7uMg =091B  
 21B0: OD 4F F1 35 34 C8 F5 DC F2 21 7E CD F4 OC F1 F5 .0q54Hu!r!~Mt.qu =0993  
 21C0: 30 05 CD 6B 22 18 02 23 14 7E B7 28 05 B9 20 EB O.Mk..#.~7(.9 k =0506  
 21D0: 10 E9 F1 C9 CD E4 OD CD OB 13 C1 C3 OD 21 7E B7 .iQiMd.M..AC.!~7 =0843  
 21E0: C8 3E 5C CD 06 0D 7E B7 28 08 CD F4 OC CD 6B 22 H>M..~7(.Mt.Mk" =06CE  
 21F0: 10 F4 3E 5C C3 06 0D 7E B7 C8 CD 67 OD CD F4 OC .t>C..~7Hmg.Mt. =077F  
 2200: 77 23 14 10 F2 C9 36 00 5A 06 FF CD 9C 21 CD 67 w...ri6.Z..M..!Mg =06CC  
 2210: OD FE OD 28 46 FE 1B C8 FE 08 20 0E 15 14 28 11 ~. (F7.H~. . .( =04FD  
 2220: 2B 15 CD FE OC CD 6B 22 18 E4 F5 7B FE FF 38 08 +.M~.Mk".du{~.B. =081A  
 2230: F1 3E 07 CD 06 0D 18 D6 92 1C 14 D5 EB 6F 26 00 q>.M..V...Uko&. =061B  
 2240: 19 44 4D 03 CD C1 08 D1 F1 CD F4 OC 77 23 18 BE .DM.MA.QqMt.wH.> =0742  
 2250: 7A B7 C8 15 2B CD 04 0D 10 F6 C9 CD E4 OD CD OB z7H.+M...v1Md.M. =077C  
 2260: 13 C1 D1 37 F5 21 29 02 C3 1F OA E5 1D 7E B7 28 .Aq7u!..C.e.^7( =0668  
 2270: 07 23 7E 2B 77 23 18 F5 E1 C9 CO EB 21 C2 01 ED .W~+wW.uai@k!B.m =07AO  
 2280: 5F 77 23 77 23 77 23 E6 7F 77 23 36 80 EB \_wWwWwWwWf.w#6.k =0667  
 2290: C9 CD 3E OB CO E5 2A 99 01 CD OB 13 CD 90 OF ED IM>.@e\*\*.M..M..m =078C  
 22A0: 5B 9B 01 7C 92 20 02 7D 93 28 48 4E 23 7E 23 CB {..!.. .). (HN#~HK =0584  
 22B0: 7F 20 42 CD 06 0D 79 E6 7F C4 06 0D 7E E6 7F C4 . BM..yf.D..~f.D =071D  
 22C0: 06 OD 23 7E B7 C4 06 OD 23 3E 24 CB 79 C4 06 OD ..#~7D..#>KyD.. =04E2  
 22D0: 3E 3D CD 06 0D 79 17 9F 22 A1 01 E5 20 09 CD F5 >=M..y...!..e .Mu =061E  
 22E0: 27 CD 02 2A CD 95 1A CD DC 1A E1 23 23 23 23 23 'M.\*M..M\..a#\*\* =05EF  
 22F0: 23 18 A6 E1 C9 23 23 18 F2 C8 D2 49 10 CD 82 OF #.kai##.rHR1.M.. =072C  
 2300: ED 53 10 02 11 OA 00 FE 2C 20 06 CD 48 OF DC 82 mS....~, .MH.\. =053F  
 2310: OF ED 53 12 02 3E E2 CD F5 08 CD 99 OA D1 C5 60 .m.S...>bMu.M..QE' =07B3  
 2320: 69 D9 11 01 00 21 00 00 D9 CD BB OA D2 49 10 E5 iY...!..YM;.RI.e =05F0  
 2330: D9 EB ED 4B 12 02 78 B1 CA 11 09 CD CF 28 ED 5B Ykmk..x1J..MO(mI =0849  
 2340: 10 02 19 DA OD 18 E5 ED 5B 10 02 CD B8 OA DA 49 ...Z..em[..MB.ZI =061B

2350: 10 D1 C5 60 69 7E 23 B6 28 0A 23 7E 23 66 6F ED .OE'!~#6(.#~#fom =067E  
 2360: 52 DA OD 18 C1 E1 D1 E5 ED 52 E3 2B ED 42 E3 38 RZ..AaQemRcmBk8 =0948  
 2370: 0A ED 42 DA 49 10 09 EB E1 E5 19 E3 E5 50 59 ED .mBZI..kae.cePYm =089D  
 2380: 4B 99 01 09 E5 CD BB 08 E1 22 99 01 60 69 C1 E3 K...eM;.a...iAc =076D  
 2390: D5 ED BO E1 ED 4B 10 02 23 23 71 23 70 23 7E B7 UmQamK..#Wq#p#~7 =073F  
 23A0: 20 FB 23 D1 7C 92 20 02 7D 93 28 0B D5 EB 2A 12 {#Q1. .}.Uk\*. =067E  
 23B0: 02 09 44 4D E3 18 E1 CD FB 12 C3 7C 0A CD C2 16 ..DMk.aMx.C1.MB. =0745  
 23C0: D5 E5 21 AB 01 CD 22 28 2A 9B 01 E3 3A 3F 01 F5 Ue!{.M"(\*.c?..u =0683  
 23D0: 3E 2C CD F5 08 CD C2 16 C1 3A 3F 01 A8 1F DA C9 >,Mu.MB.A!?.(ZI =077E  
 23E0: 14 E3 EB E5 2A 9B 01 7C 92 20 02 7D 93 C2 49 10 .cke\*..!.).BI. =06E8  
 23F0: D1 E1 E3 D5 CD 22 28 E1 11 A8 01 CD 22 28 E1 C9 GacUM"(a..M"(ai =08DD  
 2400: 3E 01 32 89 01 C3 C2 16 E5 19 EB 2A 9D 01 B7 ED >..Ck.E.k..7m =06EB  
 2410: 52 E3 C1 EB 1B 1B 1B 1B 1B 28 02 ED BO ED 53 RckA.....(m0mS =068A  
 2420: 9D 01 32 89 01 E1 7E FE 2C CO CD 48 OF 18 D1 C8 ..2..a~~, @MH..@H =0778  
 2430: ED 73 10 02 E5 11 60 24 D5 CD 29 10 0E 00 D5 CD ms..e.'\$UM)...UM =0677  
 2440: 49 OF 28 10 3E 2C CD F5 08 OC C5 CD 29 10 C1 EB I.(.),Mu..EM).Ak =0647  
 2450: E3 EB 18 EA 1B EB 2A 10 02 2B 72 2B 73 2B 2B C9 ck.jUk\*..+r+b+i =0726  
 2460: E1 C9 3E 00 C4 0E 1E CO FE OB 28 F6 D2 49 10 32 AI>..D..@~.({vR1.2 =071C  
 2470: 25 02 C9 CD 2C 10 1A C3 DC 18 CD 29 10 D5 3E 2C %IM,..C1.M).U), =060F  
 2480: CD F5 08 CD OE 1E D1 12 C9 21 2E 2F CD OE 28 18 Mu.M..Q.I!./M.!. =0608  
 2490: OC CD OE 28 18 04 C1 DD,E1 D1 CD DB 27 78 B7 C8 .M.(..A1aQMI'x7H =0841  
 24A0: 3A A6 01 B7 CA FE 27 90 30 11 ED 44 D9 DD E5 CD :&.7J~..O..mDYleM =08F1  
 24B0: OB 28 D9 DD E3 CD FE 27 D9 DD E1 FE 29 DO F5 CD .(Yjcm'Yja~)PuM =080E  
 24C0: 2A 28 67 F1 CD 98 25 B4 21 A1 O1 F2 DF 24 CD 63 \*(gqM.%4!!..r\_\$.Mc =07D0  
 24D0: 25 30 69 23 34 CA 5E 25 2E 01 CD BS 25 18 5D AF %0i#4J%..MS%.J/ =055C  
 24E0: 90 47 7E 9B 5F 23 7E 9A 57 23 7E DD 9D DD 6F 23 .G~.\_W~.WH~!J.Jo# =076B  
 24F0: 7E DD 9C DD 67 23 7E 99 4F DC 7B 25 68 63 AF 47 ~!J.Jg~.O!(%hc/G =0801  
 2500: 79 B7 20 27 DD 4C DD 7D DD 67 DD 6A AF 54 65 6F y7 'JLJj!gjjj/Teo =085C  
 2510: 78 D6 08 FE DO 20 E8 AF 32 A6 01 C9 05 29 CB 12 xV.^P h/2&..I.K. =0788  
 2520: 08 DD 29 08 30 02 DD 23 08 CB 11 F2 1C 25 78 5C .1).O.JM.K.r..%x\ =0533  
 2530: 45 B7 28 08 21 A6 01 86 77 30 DC C8 78 21 A6 01 E7(.!&.w0Nx!&. =0605  
 2540: B7 FC 4E 25 46 23 7E E6 80 A9 4F C3 FE 27 1C CO 7!N%FM~!..)OC~..@ =082F  
 2550: 14 CO DD 2C CO DD 24 CO OC CO OE 80 34 CO 1E 06 .@J, @J\$@. @..@Q.. =06D0  
 2560: C3 28 09 7E 83 5F 23 7E 8A 57 23 7E DD 8D DD 6F C(.~.\_W~.WH~!J.Jo =072D  
 2570: 23 7E DD 8C DD 67 23 7E 89 4F C9 21 A7 01 7E 2F #~!J.Jg~.OI!..~/. =0706  
 2580: 77 AF 6F 67 90 47 7D ED 52 EB 6F DD 9D DD 6F 7D w/oq.G)mRkoJ.Jo =092C  
 2590: DD 9C DD 67 7D 99 4F C9 06 00 D6 08 38 10 43 5A 1.Jg}.OI..V.B.CZ =0684  
 25A0: DD 55 08 DD 7C DD 6F 08 DD 61 0E 00 18 EC C6 09 JU.J!J.Jo..1F. =0706  
 25B0: 6F AF 2D C8 79 1F 4F DD 7C 1F DD 67 DD 7D 1F DD o/-Hy.OJ!..Jgjj.J =080C  
 25C0: 6F CB 1A CB 1B CB 18 18 EB 00 00 00 00 81 06 OK.K.K.h..... =04A4  
 25D0: 23 85 AC C3 11 7F 53 CB 9E B7 23 7C CC FE A6 OD .,,C..SK.7W..L~.l. =0839  
 25E0: 53 7F CB 5C 60 BB 13 80 DD E3 4E 38 76 80 5C 29 S.K!'.!..JcNBv.\) =0768  
 25F0: 3B AA 38 82 CD C8 27 B7 EA 49 10 21 A6 01 7E 01 ;#8.MH'7jI.!&.. =069C  
 2600: 35 80 DD 21 F3 04 11 FA 33 90 F5 70 D5 DD E5 C5 5.J!s..z3.upuJeE =0939  
 2610: CD 9D 24 C1 DD E1 D1 04 CD CC 26 21 C9 25 CD 91 M.\$A1aQ.ML!&IM. =090E  
 2620: 24 21 CF 25 CD 01 30 01 80 80 DD 21 00 00 11 00 !\$!0%M.O...!J!.... =0447  
 2630: 00 CD 9D 24 F1 CD C6 29 01 31 80 DD 21 17 72 11 .M.\$qMF).J..!..r. =0685  
 2640: D2 F7 1B 04 C1 DD E1 D1 CD C8 27 C8 2E 00 CD 6C Rw..A1aQmH'H..M1 =0920  
 2650: 27 79 D5 D9 4F D1 DD E5 E1 D9 01 00 00 50 58 DD 'yUYOQJeaY...PXJ =0870  
 2660: 21 00 00 21 FC 24 E5 21 71 26 E5 E5 E5 21 A1 !..!\$!e!qkeeee!! =0755  
 2670: 01 7E 23 B7 20 0E 43 5A DD 55 08 DD 7C DD 6F 08 .~#7..CZIU.J!J.Jo =060B  
 2680: DD 61 4F C9 E5 EB 1E 08 1F 57 79 30 12 E5 D9 E3 JaOiek...WyO.eYc =081E  
 2690: 19 E3 EB DD E5 E3 ED 5A E3 DD E1 EH 89 D9 E1 1F .ck1ecmZcJak.Ya. =0BC1  
 26A0: 4F DD 7C 1F DD 67 DD 7D 1F DD 6F CB 1C CB 1D CB 0J!..Jgjj.JoK.K.K =086A  
 26B0: 18 1D 7A 20 D3 E3 E1 C9 CD E3 27 01 20 E4 DD 21 ..z SkalImc'. .J! =07B1  
 26C0: 00 00 11 00 00 CD FE 27 C1 DD E1 D1 CD C8 27 CA ....M~'!A1aQmH'J =07D9  
 26D0: 14 09 2E FF CD 6C 27 FD E3 34 34 2B E5 D9 E1 4E ...M1!)@44+eYaN =080C  
 26E0: 2B 56 2B 5E 2B 7E 2B 6E 67 EB D9 41 E3 DD E5 FD +V+^+~+ngkYAkJ!e =0862  
 26F0: E1 AF 4F 57 5F DD 21 00 00 32 24 02 E5 FD E5 C5 a/OW\_!..2\$.e)eE =0777  
 2700: E5 78 D9 E3 B7 ED 52 E3 EB FD E5 E3 ED 52 E3 FD exYc7mRck)ecmRc) =0CC1  
 2710: E1 EB 99 D9 E1 47 3A 24 02 DE 00 3F 30 09 32 24 ak.YaG:\$.^..?0.2\$ =0672

2720: 02 F1 F1 F1 37 18 04 C1 FD E1 E1 OC OD 1F FA 67 .qqq7..A)aa...zg =0841  
 2730: 27 17 CB 13 CB 12 08 DD 29 DB 30 02 DD 23 DB CB '.,K.K.,J).O.JW.K =0514  
 2740: 11 29 DB FD 29 08 30 02 FD 23 QB CB 10 3A 24 02 .).).O.).W.K.:B. =0405  
 2750: 17 32 24 02 79 B2 B3 DD B4 DD B5 20 9F E5 21 A6 .2\$.y231415 .e!k =07DB  
 2760: 01 35 E1 20 97 18 2D FD E1 C3 3D 25 78 B7 28 20 .5a ..-)aC=%x7( =068D  
 2770: 7D 21 A6 01 AE 80 47 1F AB 78 F2 BF 27 C6 B0 77 )!k...G.(xr.F.w =075E  
 2780: CA B6 26 CD 2A 28 77 2B C9 CD C8 27 2F B7 21 B7 J6&M\*(w+IMH'7!7 =07AA  
 2790: E1 F2 17 25 C3 5E 25 CD 0B 2B 78 B7 C8 C6 02 38 ar.%C^%M.(x7HF.B =074C  
 27A0: F3 47 CD 9D 24 21 A6 01 34 CO 18 EE CD C8 27 06 eGM.\$!L.48.hMH'. =0746  
 27B0: BE 11 00 00 21 A6 01 4F D5 DD E1 11 00 00 70 06 ....!k.0U1ja...p. =04CA  
 27C0: 00 23 36 80 17 C3 F9 24 3A A6 01 B7 C8 3A AE 01 .W6..Cy\$!z.7H%: =0610  
 27D0: FE 2F 17 9F CO 3C C9 CD C8 27 2F 20 21 A5 01 7E EE ~/.@  
 27E0: EO 77 C9 EB 2A A1 01 E3 E5 2A A3 01 E3 E5 2A A5 .w!k!.ce\*H.c\*% =08A4  
 27F0: 01 E3 E5 EB C9 11 A1 01 01 06 00 ED B0 C9 ED 53 .cekI.!...mOImS =07DD  
 2800: A1 01 DD 22 A3 01 ED 43 A5 01 C9 21 A1 01 9E 23 !.J%H.mC%.I!!!.~H =0628  
 2810: 5E 23 4E DD 69 23 4E DD 61 23 4E 23 46 23 C9 11 VWNjI#NjI#NMFHI. =0593  
 2820: A1 01 01 06 00 EB ED B0 EB C9 21 A5 01 7E 07 37 !...kmOkI!%~.7 =0668  
 2830: 1F 77 3F 1F 23 23 77 79 07 37 1F 4F 1F AE C9 78 .w?..WWwy.7.O..Ix =04E4  
 2840: B7 28 85 CD C8 27 79 28 88 21 A5 01 AE 79 FA D1 7(.MH'y(.!.%..yz@ =0802  
 2850: 27 CD 59 2B 1F A9 C3 D1 27 23 78 BE CO 2B 79 BE 'MY(.)CG'Wx>y+y =0773  
 2860: C0 2B DD 7C BE CO ZB DD 7D BE CO ZB 7A BE CO 2B @+J>B+]>B+z>@+ =0913  
 2870: 7B 96 CO E1 C9 5E 23 56 23 4E 23 46 23 C9 47 4F (.GaI#WVNMFHIGO =06AE  
 2880: 57 5F DD 67 DD 6F B7 C8 E5 CD OB 28 CD 2A 28 AE W\_1gJb7HeM.(M%). =0877  
 2890: 67 F2 A4 28 1B 7A A3 3C 20 0A 2B DD 7C DD A5 gr\$!(.z#< .J+J!J% =07A6  
 28A0: 3C 20 01 0D 3E A8 90 CD 98 25 7C 17 DC 4E 25 06 < ..>(.M.%).\\N%. =0552  
 28B0: 00 DC 7B 25 E1 C9 21 A6 01 7E FE A# 3A A1 01 DO .\(\%aI!k.~\:(!!.P =07BE  
 28C0: 7E CD 7E 2# 36 A8 F# F5 79 17 CD F9 24 F1 C9 21 ~M~(6((uy.My\$QI! =0894  
 28D0: 00 00 78 B1 C8 3E 11 3D C8 29 38 08 # 29 EB 30 ..x1H>.=H(B,k)ko =05DD  
 28E0: F6 09 30 F3 C3 OD 18 FE 26 CA 7A 29 FE 2D F5 28 v.0sC..~^&Jz~u- =07E3  
 28F0: 05 FE 2B 28 01 2B CD 17 25 47 57 5F 2F 4F CD 48 .~+(.M.+KzW-/OMH =051B  
 2900: OF 3B 56 FE 2E 28 26 FE 45 20 25 CD 48 OF 15 FE .8V~.(&^E %MH..~ =05D6  
 2910: #B 28 0E FE 2D 28 0A 14 FE # 28 05 FE D7 28 01 X.(.-~(..~+(.~W(. =05D3  
 2920: 2B CD 48 OF 38 49 14 20 07 AF 93 5F OC OC 28 CE +MH.BI. ./.~..(N =04BA  
 2930: E5 7B 90 F2 42 29 CD E3 27 ED 44 21 C9 25 CD F5 e(.rB)Mc'mD!I%Mu =0926  
 2940: 27 37 F5 CD D3 29 20 FB F1 30 07 C1 DD E1 D1 CD '7uMS) (q0.A)zQM =097C  
 2950: CC 26 D1 F1 CC DB 27 # 29 D5 57 78 # 47 C3 # L&qQL'kIUWx.GE@ =0A54  
 2960: #5 CD 97 27 F1 D6 30 CD C6 29 E1 C1 D1 18 8F 7B UM.'qVOMF)aQ..( =09A8  
 2970: 87 87 83 87 86 D6 30 5F 18 A7 11 00 00 23 96 28 ....VO\_.'.~..N.( =05B4  
 2980: 2E 2B CD 48 OF 28 22 38 OC D6 41 38 1C FE 06 30 .+MH.(^B.VAB.^~.0 =04AA  
 2990: 18 C6 0A 18 02 D6 30 08 7A FE 10 D2 5E 25 08 # F...VO.z~.R~%.k =05E0  
 29A0: 29 29 29 29 # 6F # 18 D9 E5 CD C2 18 E1 C9 CD ))))5ok.YeMB.AIM =08A7  
 29B0: 48 OF 28 F5 FE 31 37 28 04 FE 30 20 EC CB 13 CB H.(u~17(.~0 1K.K =06E9  
 29C0: 12 DA 5E 25 18 E9 CD E3 27 CD AF 27 C1 DD E1 D1 .Z^%.!M'/'A)z@ =093A  
 29D0: C3 9D 24 C8 F5 CD 97 27 F1 3D C9 D5 E5 F5 CD #B C.~H~M.'q=IueuMB =0AF7  
 29E0: 26 F1 E1 D1 3C C9 E5 21 73 08 CD # 0D E1 CD C1 &qab@Ie!s.M..AMA =0BF3  
 29F0: 18 21 25 02 7E 36 00 F5 CD 02 2A F1 32 25 02 C3 .!%~.6.M.\*q2%.C =050F  
 2A00: D8 1A AF 32 20 02 21 A9 01 36 20 E6 08 28 02 36 X./2 .!).6 f. (.6 =0464  
 2A10: 2B CD C8 27 F2 1F 2A 36 2D E5 CD # 27 E1 B4 23 +MH'r.\*#6-eM!a4# =07F1  
 2A20: 36 30 3A 20 02 57 17 DA D7 2A CA CF 2A E5 CD 44 60! .W.ZW\*JO!eMD =06C4  
 2A30: 2C 21 25 02 34 35 28 51 57 C6 0B FA 7A 2A BE 28 ,!%.45(QWF.zz\*)( =0502  
 2A40: 02 30 37 47 7E 90 3C 4F 04 7A 16 0B E1 23 CD A8 .07G~.  
 2A50: 2B E5 AF 01 00 00 ED B1 2B 01 61 2A C5 AF F5 18 +e/.m1+.aE/E/u. =0696  
 2A60: 47 E1 7E FE 2D C8 FE 20 C8 FE 30 28 0A FE 25 20 Ga~~~H~ H~0(.~% =0822  
 2A70: 05 23 7E FE 2D C8 2B 36 20 C9 4E OD 28 01 0C 06 .W~~~-H+6 IN.(!... =0479  
 2A80: 02 E1 23 7A 16 00 C3 FC 2B 01 00 03 C6 0C FA 99 .a#z..CI+...F.z. =05E9  
 2A90: 2A FE OD 30 04 3C 47 3E 02 D6 02 E1 F5 CD 7F 2C \*~.O.<G>.V.auM., =0652  
 2AA0: 36 30 20 01 23 CD 92 2C 2B 7E FE 30 28 FA FE 2E 60 .WM.,+~^O(z~. =065A  
 2AB0: 28 01 23 F1 28 1A 36 45 23 36 2B F2 C2 2A 36 2D (.Wq(.6E#6+rB#6- =04BF  
 2AC0: # 44 06 2F 04 D6 0A 30 FB C6 3A 23 70 23 77 23 mD/.V.O(F:#p#W# =05C5  
 2AD0: 36 00 EB 21 A9 01 C9 23 C5 E5 7A 1F DA F3 2B 01 6.k!).IWEez.Zs+. =0714  
 2AE0: 0E B6 DD 21 C9 1B 11 04 BF CD 3F 28 FA F8 2A E1 .6!I...?M?(zx#a =07AB

2AFO: C1 CD 02 2A 3U 36 25 C9 16 0B CD C8 27 C4 44 2C AM.\*+6%1..MH'DD, =061A  
 2B00: E1 C1 FA 4B 2B C5 5F 78 92 93 F4 1E 2D CD 26 ZD aAz(+E\_x..t.-M-&= =08BF  
 2B10: CD 92 2C B3 C4 40 2D B3 C4 7F 2C D1 7B B7 20 01 M.,3D@-3D.,9(7 . =07B5  
 2B20: 2D 3D F4 1E 2D E5 21 A9 01 46 0E 20 3A 20 02 5F +=t.-e!).F. : \_ =0486  
 2B30: E6 20 28 07 78 B9 0E 2A 20 01 41 71 CD 48 OF 28 f (.x9.\* .AqMH.( =04BD  
 2B40: 10 FE 45 28 OC FE 30 28 F2 FE 2C 28 EE FE 2E 20 .~E(.~O(r~, (n~. =0758  
 2B50: 03 2D 36 30 CB 63 28 03 2B 36 24 CB 53 20 02 2B .+60Kc(.+6sKS .+ =03DD  
 2B60: 70 E1 2B 02 70 23 36 00 21 A8 01 23 3A 1E 02 95 pa!.p#6.!(.#:. =0420  
 2B70: 92 C8 7E FE 20 28 F4 FE 2A 28 F0 2B E5 F5 CD 48 .H~~ (t~\*(p+euMH =096C  
 2B80: OF FE 2D 2B 28 F8 FE 2B F4 FE 24 28 F0 FE 30 20 .~-(x~+(t~\*(p~0 =0827  
 2B90: 10 23 CD 4B OF 30 0A 2B 01 2B 77 F1 28 FB C1 18 .MH.M.0.4.+wq((A. =054C  
 2BA0: CB F1 2B FD E1 36 25 C9 5F 79 B7 28 01 3D 83 FA Kq({a6%I\_y7(.=.z =0858  
 2BB0: B3 2B AF C5 F5 FC 3B 29 20 FB C1 7B 90 C1 5F 82 3+/Eu;[] {A(.A.\_ =09D0  
 2BC0: 78 FA CF 2B 92 93 F4 1E 2D C5 CD 26 2D 18 11 CD xz0+..t.-EM&..M =07AB  
 2BD0: 1E 2D 79 CD 2B 2C 4F AF 92 93 CD 1E 2D C5 47 4F .-yM.,0//..M.-EG0 =06D5  
 2BE0: CD 92 2C C1 B1 20 03 2A 1E 02 83 3D F4 1E 2D 50 M.,A1 .\*.=t.-P =05B9  
 2BF0: C3 25 2B CD C8 27 37 C4 44 2C E1 C1 F5 79 B7 F5 C%+MH'7DD,aAuYzu =08F6  
 2CO0: 28 01 3D 80 4F 7A E6 04 FE 01 9F 57 81 4F D6 OB (.=.0zf..W.W.OV. =063F  
 2C10: F5 C5 FC 3B 29 FA 12 2C C1 F1 C5 F5 FA 20 2C AF uE{[]z.,AqEuz, / =0A53  
 2C20: 1D 44 80 3C 82 47 0E 00 CD 92 2C F1 F4 3A 2D C1 mD.<.G..M.,qt:-A =075C  
 2C30: F1 20 01 2B F1 3B 04 C6 0B 90 92 C5 CD B6 2A EB q .+q8.F...EM6\*k =07BA  
 2C40: D1 C3 25 2B D5 AF F5 CD 6D 2C 01 15 A2 DD 21 F8 QC%+U/uMm,,!1\*x =0871  
 2C50: 02 11 FD FF CD 3F 2B F2 6A 2C F1 CD D4 29 F5 C3 ..).M?(rj,qMT)uC =093E  
 2C60: 4A 2C F1 CD 3B 29 F5 CD 6D 2C F1 D1 C9 01 3A 05 J,qM{]uMm,qQI.:% =08FE  
 2C70: 1D 21 B7 43 11 FC 3F CD 3F 28 E1 F2 62 2C E9 05 !17C.!?M?(arb,i. =07C7  
 2C80: 20 08 36 2E 22 1E 02 23 48 C9 0D CO 36 2C 23 0E .6..#HI.@6,#. =0362  
 2C90: 03 C9 D5 C5 E5 CD 89 24 3C CD 7E 28 CD FE 27 E1 .IUEeM.\$<M~(M~'a =0947  
 2CA0: C1 11 E7 2C 3E 0B CD 7F 2C C5 F5 E5 D5 CD 0B 28 A.g,>M.,EueUM.{ =081A  
 2CB0: E1 06 02 F4 7B 96 5F 23 7A 9E 57 23 DD 7D 9E DD a/.(.\_.#z.WH)].] =0714  
 2CC0: 6F 23 DD 7C 9E DD 67 23 79 9E 4F 2B 2B 28 30 o#1].jgHy.0+++\*0 =0632  
 2CD0: E2 CD 63 25 23 CD FE 27 EB E1 70 23 F1 C1 3D 20 bMc%#MH'kap#qA= =08BA  
 2CE0: C5 CD 7F 2C 77 D1 C9 00 E4 0B 54 02 00 CA 9A 3B EM.,wQI.d.T..J.; =0732  
 2CF0: 00 00 E1 F5 05 00 80 96 98 00 00 40 42 OF 00 00 ..au.....@B... =041A  
 2DO0: A0 B5 01 00 00 10 27 00 00 00 E8 03 00 00 00 64 .....'..h....d =02AD  
 2D10: 00 00 00 00 00 00 00 00 00 01 00 00 00 00 B7 C8 .....7H =018A  
 2D20: 3D 36 30 23 18 F8 7B 82 3C 47 3C D6 03 30 FC C6 =60%.x<.G<V.0!F =065D  
 2D30: 05 4F 3A 20 02 E6 40 CO 4F C9 20 04 C8 CD 7F 2C .0:. f@#0I .HM., =0612  
 2D40: 36 30 23 3D 18 F6 CD 48 OF 30 1E CD 82 OF E5 CD 60%=.vMH.O.M..eM =0656  
 2D50: BB OA D2 EB 10 60 69 23 23 23 CD 48 OF D6 9C C2 8.RK.i##MH.V.B =0719  
 2D60: 49 10 47 CD 98 1A E1 18 03 CD CE 14 CD 49 OF 37 I.GM..a..MN.MI.7 =0626  
 2D70: 2B OC FE 2C 28 05 FE 3B C2 11 09 CD 48 OF EB 2A (.~,(.~;B..MH.K\*= =05D9  
 2D80: A1 01 01 D1 EB E5 F5 D5 46 AF BO CA 49 10 23 23 ...QkeuUF/OJI.## =081C  
 2D90: 4E 23 66 69 C3 9D 2D CD 1F 2F CD 06 0D AF 5F 57 N#fIC.-M./M../\_W =062D  
 2DA0: CD 1F 2F 57 7E 23 FE 23 CA E4 2D FE 27 CA A7 2E M./W~W~WJd-~'J'. =07D3  
 2DB0: 05 CA 93 2E FE 2B 3E 08 2B E6 2B 7E 23 FE 2E CA J..~+. (f+~#~.J =06CF  
 2DC0: FC 2D BE 20 D2 FE 24 28 14 FE 2A 20 CA 78 FE 02 !-> R^\$(\*.~\*J x~. =07C1  
 2DD0: 23 3B 03 7E FE 24 3E 20 20 07 05 1C FE AF C6 10 #8.~\$> ...~/.F. =0527  
 2DE0: 23 1C B2 57 1C 0E 00 05 28 46 7E 23 FE 2E 28 17 #..W....(F~W~.(. =03C1  
 2DF0: FE 23 2B F0 FE 2C 20 19 CB F2 18 E8 7E FE 23 3E ~#(p~, .Kr.h~~#) =0836  
 2E00: 2E C2 97 2D 0E 01 23 0C 05 28 25 7E 23 FE 23 2B .B.-..#.(%~#~#( =042E  
 2E10: F6 D5 54 5D FE 5E 20 16 BE 20 13 23 BE 20 OF 23 vUT1^~ .> .#> .# =0632  
 2E20: BE 20 DB 23 78 D6 04 38 05 D1 47 14 23 CA EB D1 > .#xV.B.QG.#JkQ =0670  
 2E30: 2B 1C CB 5A 20 13 1D 7B B7 28 0E 7E D6 2D 28 06 +.KZ ..x7(.~V-(. =04D0  
 2E40: FE FE 20 05 CB DA CB D2 05 E1 F1 28 51 C5 D5 CD ~~ .K2KR.aq(GEUM =0A1A  
 2E50: D3 14 D1 C1 C5 E3 43 78 81 FE 19 D2 49 10 7A F6 S.QAEeCx.~.RI.zv =0911  
 2E60: 80 CD 03 2A CD D9 1A E1 CD 49 OF 37 28 0C FE 3B .M.\*MY.aMI.7(.~; =06E4  
 2E70: 28 05 FE 2C C2 11 09 CD 48 OF C1 EB E1 E5 F5 D5 (.~,.B..MH.AkaeuU =0893  
 2E80: 7E 90 23 23 4E 23 66 69 16 00 5F 19 78 B7 C2 9D ~.##N#fI..\_.x7B. =05B0  
 2E90: 2D 18 06 CD 1F 2F CD 06 OD E1 F1 C2 83 2D DC 0B --.M./M..aqB.-\,. =0671  
 2EA0: 13 E3 CD 12 1C E1 C9 0E 01 1E 4C 05 28 1C 7E 23 .cM..a1...L.(.~# =04FE  
 2EB0: FE 45 28 OC FE 52 28 08 FE 4C 28 04 FE 43 20 OA ~E(.~R(.~L.(.~C . =05D8

2EC0: 5F 0C 05 28 05 7E 23 BB 28 F7 CD 1F 2F E1 F1 28 ... .~#; (WM./aq( =062D  
 2ED0: CD C5 D5 CD CE 14 D1 C1 C5 E5 2A A1 01 41 0E 00 MEUMN.QAEe\*!.A.. =086D  
 2EE0: 7B FE 45 28 2B D5 C5 CD C0 1C C1 D1 78 96 47 7B {~E(+UEM@.AQx.G( =0886  
 2EF0: FE 4C 28 0E FE 52 28 15 78 CB 38 90 08 CD 17 2F C3 67 2E AF 18 EC .GEM\AM./Cg./.1 =0633  
 2FO0: 08 47 C5 CD DC 1A C1 CD 17 2F C3 67 2E AF 18 EC .GEM\AM./Cg./.1 =07B6  
 2FI0: 7B 96 30 ED AF 18 EA 04 05 C8 CD F2 OC 18 F9 F5 x.0m/.j..HMr..y.. =087E  
 2F20: 7A B7 3E 2B C4 06 0D F1 C9 21 DB 27 E3 E9 00 00 z7>+D..q!l'ci.. =071A  
 2F30: 00 00 00 80 CD E3 27 21 2E 2F CD F5 27 C1 DD E1 ....Mc'!.Mu'Aja =073D  
 2F40: D1 CD C8 27 28 44 78 B7 CA 18 25 D5 DD E5 C5 79 QMH'(Dx7J.%U1eEy =0904  
 2F50: F6 7F CD OB 28 F2 6A 2F D5 DD E5 C5 CD B6 28 C1 v.M. (rJ/U)eEM6(A =09C8  
 2F60: DD E1 D1 F5 CD 3F 28 E1 7C 1F E1 22 A5 01 E1 22 JaQuM?(al.a.%a." =08E0  
 2F70: A3 01 E1 22 A1 01 DC 29 2F CC DB 27 D5 DD E5 C5 #.a!..).L[U]jeE =08A7  
 2F80: CD F4 25 C1 DD E1 D1 CD 48 26 01 38 81 DD 21 3B Mt%AJaQMH&.8.); =0864  
 2F90: AA 11 5C 29 CD 48 26 3A A6 01 FE 88 D2 89 27 CD \*.).)MH&:&.^R.'M =0731  
 2FA0: E3 27 CD B6 28 C1 DD E1 D1 F5 CD 9A 24 21 C4 2F c'M6(AJaQuM.\$!D/ =0999  
 2FB0: CD 10 30 21 A6 01 F1 B7 FA BD 2F 86 01 86 3F 77 M.O!&.q7z=...?w =0726  
 2FC0: D0 C3 89 27 0A CC D5 45 56 15 6A CF 37 A0 92 27 PC..LUeV.j07 .' =0767  
 2FD0: 6D F5 95 EE 93 00 71 DO FC A7 78 21 74 B1 21 82 mu.n..qP!x!t1!. =08BD  
 2FEO: C4 2E 77 82 58 58 95 1D 7A 6D CB 46 58 63 7C E9 D.w.XX..zmKFXc:i =0765  
 2FF0: FB EF FD 75 7E D2 F7 17 72 31 80 00 00 00 00 00 (o)u~Rw.r1..... =06DD  
 3000: 81 CD E3 27 11 44 26 D5 E5 CD 0B 28 CD 48 26 E1 .Mc'.D&UeM.(MH&a =07A9  
 3010: CD E3 27 7E 23 CD F5 27 FE F1 C1 DD E1 D1 3D C8 Mc'~#M'~qAJaQ=G =0AA5  
 3020: D5 DD E5 C5 F5 E5 CD 48 26 E1 CD 0E 28 E5 CD 9D U1eEueMH&aM. (e.M. =0AA4  
 3030: 24 E1 18 E5 CD C8 27 FA 5B 30 21 C2 01 CD F5 27 \$a.eMH'z!O!B.Mu' =0810  
 3040: C8 01 35 98 DD 21 7A 44 11 00 00 CD 48 26 01 28 H.5. J!zD..MH&.( =04C7  
 3050: 68 DD 21 46 B1 11 00 00 CD 9D 24 CD 0B 28 7B 59 hJ!F1..M.\$M. ((Y =05D0  
 3060: 4F 36 80 2B 46 36 80 CD FC 24 21 C2 01 C3 1F 28 06.+F6.M!\$!B.C.( =0607  
 3070: 01 49 81 DD 21 DA OF 11 21 A2 CD 9D 24 CD E3 27 .I.J!Z..!M.\$Mc' =06EB  
 3080: 01 49 83 DD 21 DA OF 11 21 A2 CD FE 27 C1 DD E1 .I.J!Z..!M'~AJa =07F9  
 3090: D1 CD CC 26 CD E3 27 CD B6 28 C1 DD E1 Di CD 9A QML&Mc'M6(AJaQM. =0AC9  
 30A0: 24 21 CF 30 CD 91 24 CD C8 27 37 F2 B5 30 CD 89 \$!0OM.\$MH'7r50M. =07E6  
 30B0: 24 CD C8 27 B7 F5 F4 DB 27 21 CF 30 CD BC 24 F1 \$MH'7ut[!0OM.\$q =0910  
 30C0: D4 DB 27 21 D5 30 C3 01 30 21 A2 DA OF 49 81 00 T[!UOC.O!"Z..I. =0666  
 30D0: 00 00 00 07 90 BA 34 76 6A 82 E4 E9 E7 4B .....:4vj.digK =0665  
 30E0: F1 84 B1 4F 7F 3B 28 86 31 B6 64 99 97 87 E4 36 q.10.;(.16di..d6 =07CB  
 30F0: E3 35 23 87 24 31 E7 5D A5 86 21 A2 DA OF 49 83 c5H..\$ig1%.!"Z.I. =06FE  
 3100: CD E3 27 CD 7D 30 C1 DD E1 D1 CD E4 27 EB CD FE Mc'M)OAJaQMd'kM~ =0B2F  
 3110: 27 CD 70 30 C3 C8 26 CD C8 27 FC 29 2F FC DB 27 'MpOCH&MH'!)/![' =0853  
 3120: 3A A6 01 FE 81 38 10 01 00 81 DD 21 00 00 51 59 :&.^8....!..QY =04D2  
 3130: CD CC 26 21 91 24 E5 21 4D 31 CD 01 30 21 C9 30 ML&..\$.e!M1M.O!IO =0631  
 3140: C9 65 86 DD 21 E0 2E 11 29 D3 C3 48 26 OD 14 07 Ie.J!..)SCH&... =0626  
 3150: BA FE 62 75 51 16 CE DB D6 78 4C BD 7D D1 3E 7A :~buQ.NXVxL=)Q)z =08F9  
 3160: 01 CB 23 C4 D7 7B DC 3A OA 17 34 7C 36 C1 A3 81 .K#DW(\!:..4;6A#. =0707  
 3170: F7 7C EB 16 61 AE 19 7D 5D 78 8F 60 B9 7D A2 44 wlk.a..)x.'9)'D =07F9  
 3180: 12 72 63 7D 16 62 FB 47 92 7E CO FO BF CC 4C 7E ..rcj.b(G.~@pPLL~ =0833  
 3190: 7E 8E AA AA AA 7F FF FF 7F 80 3E 04 CD 86 ~.\*\*\*.v.....).M. =0A10  
 31A0: 1A CD 32 10 2A 81 01 7A CD B2 31 7B CD B2 31 C3 .M2.\*..zM21(M21C =06ED  
 31B0: B5 1C F5 1F 1F 1F CD BB 31 F1 F5 E6 OF C6 30 5.u....M;1quf.FO =07CC  
 31C0: FE 3A 38 02 C6 07 77 23 F1 C9 3E 10 CD 86 1A CD :~B.F.w#qI).M..M =071B  
 31D0: 32 10 2A 81 01 06 10 36 30 CB 23 CB 12 30 01 34 2.\*....60KHK.0.4 =039A  
 31E0: 23 10 F4 18 CA CD OB 28 C3 48 26 CD F4 25 01 5E #.t.JM. (CH&Mt%.^ =067F  
 31F0: 7F DD 21 D8 5B 11 38 A9 C3 48 26 C1 DD E1 D1 D5 .J!X[.8)CH&AJaQU =08F8  
 3200: DD E5 C5 CD E3 27 CB 17 32 C1 DD E1 D1 CD 48 26 JeEMC'M.2AjaQMH& =09FF  
 3210: C3 96 24 C1 DD E1 D1 CD CC 26 CD E3 27 CD B6 28 C.\$AJaQML&Mc'M6( =0AOE  
 3220: C1 DD E1 D1 CD 3F 28 C8 CD C8 27 F0 01 00 81 DD AJaQMN?(HMH'p...J =0957  
 3230: 21 00 00 51 59 C3 9D 24 CD E3 27 CD 1A 32 C3 96 !..QYC.\$Mc'M.2C. =0698  
 3240: 24 01 49 82 DD 21 DA OF 11 21 A2 23 C3 FE 27 E5 \$.I.J!Z..!WC~'e =069B  
 3250: 3E 01 32 40 31 21 71 32 11 70 30 01 1A 00 32 26 >.20!q2.p0...2& =02CA  
 3260: 02 ED B0 E1 C9 E5 3E C9 32 40 31 AF 21 8B 32 18 .m0aIe>I2@1!.2. =077D  
 3270: E7 01 34 87 DD 21 00 00 11 00 00 CD 9D 24 CD E3 9.4.J!....M.\$Mc =05F0  
 3280: 27 01 34 89 DD 21 00 00 11 00 00 01 49 81 DD 21 '4.J!....I.J! =03BD

3290: DA OF 11 21 A2 CD 9D 24 CD E3 27 01 49 83 DD 21 Z..!"M.SMc'.I.! =06ED  
 32A0: DA OF 11 21 A2 3A 27 02 B7 5F 3E 00 32 27 02 EB Z..!".'7\_>2'.( =03F7  
 32B0: 03 7B 1E 01 E5 C3 95 1C ED 5B BB 01 D5 CD D3 14 .(.eC..ml..UMS. =0753  
 32C0: E5 CD C8 27 E1 D1 20 30 47 0E BB 2B 18 12 B7 20 eMH'a@ OG.=+.7 =06E1  
 32D0: 11 23 7E 23 B6 CA 11 09 23 5E 23 56 ED 53 8F 01 .#W6J..W^VmS.. =0539  
 32E0: 16 BB CD 48 OF 28 E7 BA 28 06 BB 20 F5 04 18 F2 .>MH.(g:(.9 u..r =06CB  
 32F0: 04 05 CA 13 11 05 18 EA CD D3 DB 06 BD DD E1 D5 ..J...jMS..=j@U =06FC  
 3300: ED 5B 8F 01 D5 C5 33 DD E9 CO CD 96 08 F9 FE BU m[.UE311@M..y^= =0A4A  
 3310: C2 20 09 E1 22 BB 01 C3 OF 11 CD D3 08 54 BB 06 B .a"-.C..MS.T]. =05C0  
 3320: BF 18 DA CD D3 14 D9 CD C8 27 BB CD 96 08 F9 FE ?ZMS.YMH'.M..y^ =0964  
 3330: BF C2 23 09 D1 21 CE 0E E3 08 D9 C2 13 11 D9 ED ?BW.B!N.c.YB..Ym =07EB  
 3340: 53 8F 01 18 D8 3A CA 01 C3 DC 18 ED 5B CB 01 C3 S...X:J.C..nIK.C =0766  
 3350: C2 18 CD 0E 1E C3 28 09 CD 48 OF CA 11 09 3E 8A B.M..C(.MH.J..). =0597  
 3360: CD F5 08 D2 11 09 CD 4E 10 7B B2 28 08 E5 CD BB Mu.R..MN.(2(.eMB =07AB  
 3370: OA E1 D2 BB 10 ED 53 C8 01 AF 32 CA 01 32 CB 01 .aRk.mSH./2J.2K. =076B  
 3380: 32 CC 01 C9 F5 3A CA 01 B7 CA 26 09 ED 5B BB 01 2L.Iu:J.7Jk.mIK. =0786  
 3390: F1 ED 7B CF 01 01 CE 0E C5 01 79 33 C5 DA CB 10 qm(0..N.E.y3EZ0. =07F6  
 33A0: 28 06 3E 82 CD F5 08 37 ED 53 8F 01 2A CD 01 DO (.).Mu.7mS..\*MP =0687  
 33B0: B7 23 C2 13 11 7E 23 B6 C8 23 BB 23 56 ED 53 8F 7MB..~#6HM^VmS. =06AB  
 33C0: 01 23 C3 13 11 0E OC CA 09 01 CD 0E 1E E6 01 4F .WC....J..M..f.O =0428  
 33D0: C3 55 34 CD 62 34 0E 1B CD 09 01 4A CB F9 CD 09 CU4Mb4..M..JkyM. =0693  
 33E0: 01 4B CB F9 C3 09 01 00 00 FE D1 ED 5B E7 33 28 .KKyC...~#mfg3( =0736  
 33F0: 09 CD 62 34 3E 2C CD F5 08 3E 23 BB CD 62 34 E3 .Mb4),Mu.>#UMB4c =071C  
 3400: 0E 03 18 4C CD 62 34 ED 53 E7 33 C9 CD 62 34 E5 ...LMb4mSg3Mb4e =0743  
 3410: 0E 02 18 3C CD 0E 1E E6 03 C6 05 4F 18 37 E1 3E ...<M..f.F.0.7a> =04CE  
 3420: 28 CD F5 08 CD 62 34 3E 29 CD F5 DB E5 0E 04 EB (Mu.Mb4)Mu.e..k =0768  
 3430: CD 55 34 3E 00 CA 39 34 3D CD AF 27 E1 C9 CD 29 MU4).J94=M'/AIM) =074B  
 3440: 10 D5 3E 2C CD F5 08 CD 29 10 E3 EB OE 09 18 06 .U.,Mu.M.).CK... =0622  
 3450: BB 22 E7 33 3E E5 C5 06 1B CD 09 01 C1 CD 09 01 k."g3>eE..M..AM.. =06A7  
 3460: E1 C9 CD 0E 1E F5 3E 2C CD F5 DB CD 0E 1E 57 F1 aIM..u>,Mu.M..Wq =080D  
 3470: 5F C9 3E FF C9 C3 06 01 C3 OC 01 3A EC 01 FE 49 \_I>.IC..C..:1.^I =0736  
 3480: C8 06 20 3E FF CD OC 01 10 F9 3E 1A C3 OC 01 BB H. >M...y>C..e =061B  
 3490: 3A 02 02 B7 3E 41 28 04 32 F6 01 3C 32 F5 01 21 ..>A(.2v.<2u.. =045B  
 34A0: 00 00 22 F7 01 22 F9 01 22 FB 01 3A EC 01 FE 49 .."w."y."{.:1.^I =05C2  
 34B0: 28 31 06 14 0E FF CD OC 01 10 F9 01 1E CD OC 01 (1...M..y..M.. =0469  
 34C0: 0E 01 CD OC 01 21 ED 01 06 20 16 00 4E CD OC 01 ..M..!m.. .Nm.. =035C  
 34D0: 7E 82 57 23 10 F6 ED 44 4F CD OC 01 0E 02 CD OC ~.WW.vmDOM...M. =05C3  
 34E0: 01 18 2A CD 06 01 FE 1E 20 F9 CD 06 01 FE 01 20 ..\*M..~. yM..~. =053F  
 34F0: F2 AF 4F 21 ED 01 06 20 CD 06 01 BE 20 E5 23 81 r/0!m.. M..> eH. =0660  
 3500: 4F 10 F5 CD 06 01 81 C2 E3 36 CD 06 01 E1 C9 CD 0.uM..Be6M..AIM =07CF  
 3510: 11 1E 32 BB 01 CD 2D 01 C3 AF 27 20 OC 3E 2A 32 ..2k.M.-C/.>\*2 =04A7  
 3520: EB 01 AF 32 OF 02 C3 2A 01 38 BB 3E 23 CD F5 08 k./2..Ck.B.>#Mu. =0537  
 3530: D2 11 09 CD 0E 1E 32 EB 01 E5 CD 2A 01 E1 CD 49 R..M..2k.eM\*.aMI =06D7  
 3540: OF C8 3E 2C CD F5 08 18 D2 CD CE 14 E5 CD 58 1C .H>,Mu..RMN.eMX. =07CA  
 3550: 32 EC 01 E1 3E 2C CD F5 08 38 08 3E 23 CD F5 08 21.m>,Mu.B.>#Mu. =069F  
 3560: D2 11 09 CD 0E 1E 32 EB 01 3E 2C CD F5 08 CD 75 R..M..2k,>,Mu.Mu =0679  
 3570: 35 CD 27 01 C9 CD CE 14 E5 CD 58 1C 2B 2B 2B 4E 5M'.IMN.eMX.++N =0697  
 3580: 3E 08 BB 30 01 4F 21 ED 01 06 20 36 20 23 10 FB >.90.0!m.. 6 #.( =0438  
 3590: 21 ED 01 BB 06 00 ED BO AF 12 E1 C9 D9 CD DA OA !m.k..m0/.aIYMZ. =0892  
 35A0: D9 CD 75 35 26 49 2E 04 22 EB 01 21 FF FF 22 0E YM45&I..".k.!..". =064E  
 35B0: 02 CD 27 01 21 28 02 06 00 CD 2D 01 14 28 40 CD .M..!(...M-..(BM =038C  
 35C0: 21 01 E6 7F 28 F3 FE 7F 28 EF FE 1A 28 31 FE OA !..f.(>.(o^..(1^.. =07AF  
 35D0: 20 04 04 05 28 E3 4F 7B 3C 79 28 02 23 04 77 D6 ...(>0x<)!.#.WV =0452  
 35E0: OD 20 D6 77 3E FE 32 21 02 21 28 02 CD 48 OF 3C . Vw>2!.!(.MH.< =04B6  
 35F0: 3D 28 C1 30 05 F5 AF C3 17 OA 1E 18 C3 28 09 CD =(AO.u/C...C..M =05DA  
 3600: 1D 35 CD EB OA C3 BB 09 CD 75 35 CD 49 OF 00 11 .5Mk.C8.Mu5MI... =0645  
 3610: FF FF 28 0E 3E 2C CD F5 08 D2 11 09 CD 82 OF C2 ..(.),Mu.R..M..B =0724  
 3620: 11 09 ED 53 10 02 3E FD 32 21 02 C3 95 36 3C 3C ..ms..>)2!.G.6<< =0502  
 3630: CA B4 35 CD EB OA AF 32 21 02 ED BB 10 02 7A A3 J45Mk./2!.m..zW =06F0  
 3640: 3C 01 CE 0E C5 C8 C3 E4 10 CD 75 35 32 0E 02 CD <.N.EHCd.Mu52..M =06E3  
 3650: 1D 35 3E 04 32 EB 01 32 OF 02 CD 48 OF 2B 3E 4F .5>.2k.2..MH.+>O =03D1

```

3660: 28 0C 3E 2C CD F5 08 3E 44 CD F5 08 3E 44 32 EC {.,Mu.>DMu.>D21 =0654
3670: 01 CD 27 01 E5 2A 99 01 E8 2A 97 01 OE FE CD 24 .M!.e...k!..~M$ =0649
3680: 01 4E 23 CD 24 01 7C 92 20 02 7D 93 20 F3 E1 C3 .N#M$.!.!.saC =065B
3690: 1D 35 CD 75 35 CD DA 0A 2E 04 26 49 22 EB 01 2E .5Mu5MZ...&I"k.. =0557
36A0: 00 22 0E 02 CD 27 01 2A 97 01 CD 2D 01 14 28 33 .".M'..K..(3 =0353
36B0: CD 21 01 3C 3C 20 2C 06 03 CD 2D 01 14 28 1E CD M!.<,..M-.!.M =03DE
36C0: 21 01 77 CD E2 08 7E B7 23 20 EC 10 EC CD 1D 35 !.wMb.^# 1.M.5 =06CF
36D0: 22 99 01 3A 21 02 B7 CC F8 12 C3 7C 0A CD DA 0A "...!..7Lx.C1.MZ. =06A0
36E0: 1E 14 01 1E 15 C3 28 09 CD 75 35 CD 1D 35 3E 04 ....C(.Mu5M.5). =0432
36F0: 32 E8 01 CD 49 0F 00 3E 4F 28 OC 3E 2C CD F5 08 2k.MI..>O!,Mu. =0538
3700: 3E 44 CD F5 08 3E 44 32 EC 01 21 FF FF 22 0E 02 >DMu.>D21.!. =063E
3710: CD 27 01 E1 2A 97 01 7E 23 E6 23 2B 31 5E 23 56 5'.a!.~#H#(1^H =0542
3720: 23 E5 CD C2 18 CD 02 2A 23 CD 59 37 E1 7E FE 09 #eMB.M.*#MY7a~~. =078E
3730: 28 05 0E 20 CD 24 01 CD F1 OD E5 21 29 02 CD 59 (.. M$.Mq.e!).MY =056F
3740: 37 E1 0E 0D CD 24 01 0E 0A CD 24 01 18 C9 0E 1A 7a..M$...M$.I.. =0438
3750: CD 24 01 CD 1D 35 C3 E8 09 7E B7 CB 4F CD 24 01 M$.M.5CB.^7HDM$. =06D3
3760: 23 18 F6 ED FF 42 41 53 49 43 00 C3 00 01 00 00 #.vm.BASIC.C.... =0543
3770: 21 28 03 F9 22 95 01 FD 21 D9 01 21 61 09 22 01 !(y)..}!Y.!a.. =04A3
3780: 01 21 0D 38 CD 5B OD CD 8C OC CD 48 OF FE 51 28 ...!BM!.M..MH.^Q( =059C
3790: 22 2F 3C 20 07 CD 1B 01 60 6F 18 14 21 29 02 CD "/<.M..`o..!.M =03B1
37A0: 49 OF CD E7 28 7E B7 C2 11 09 CD 2C 10 E8 2B 2B I.Mg(^7B..M..k++ =068F
37B0: 22 3B 01 2A 3B 01 22 83 01 E5 11 00 03 B7 ED 52 ";.*..e...7mR =0459
37C0: E1 DA F1 08 11 9C FF 19 11 00 03 7C 92 20 02 7D aZq.....!. =063A
37D0: 93 ED 5B 39 01 30 03 11 00 03 7C 92 20 02 7D 93 .m!9.0.....!. =049C
37E0: 38 DF F9 22 95 01 EB 22 97 01 CD E2 08 B7 ED B_y..k..Mb.7km =08B3
37F0: 52 01 F0 FF 09 CD 0B 13 CD EE 29 21 1C 38 CD 5B R.p..M..Mn)!..BM! =06B7
3800: 0D 21 5B 0D 22 1B 02 CD DA 0A C3 E8 09 0A 48 69 .!L..MZ.CB..Hi =04C5
3810: 67 5B 65 73 74 20 4D 65 6D 6F 72 F9 20 42 79 74 ghest Memory Byt =0683
3820: 65 73 20 46 72 65 65 6A 0A 57 65 6C 63 6F 6D 65 es Free..Welcom =055A
3830: 20 74 6F 20 48 42 41 53 49 43 0A 56 65 72 73 69 to HBASIC.Versi =04E0
3840: 6F 6E 20 33 2E 33 20 20 32 34 2E 20 4F 63 74 6F on 3.3 24. Octo =041A
3850: 62 65 72 20 31 39 38 35 8A 00 00 00 00 00 00 00 00 00 00 ber 1985..... =02BA
3860: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000
3870: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... =0000

```

## 2.9.4. Programmtechnische Details

Weil das etwa 140seitige Assemblerlisting nicht abgedruckt werden kann, sind nachfolgend einige programmtechnische Details in Kurzform erläutert.

Eine eingegebene Zeile wird sofort nach Bestätigung durch die RETURN-Taste in das interne Format übersetzt. Dabei wird jeder BASIC-Befehl durch ein sogenanntes Token ersetzt, d. h., durch ein byte größer/gleich 80H (größer/gleich 80H, damit Tokens von normalen Textzeichen unterscheiden werden können). Funktionen erhalten zusätzlich noch das Steuerbyte FFH. Zum Beispiel wird das Schlüsselwort END durch 80H ersetzt. Aufgrund dieser Vorübersetzung muß der Interpreter in der Ausführungsphase nicht jedesmal das Schlüsselwort «entziffern», sondern kann gleich aus dem Token eine Adresse bestimmen, auf der die Einsprungadresse der jeweiligen BASIC-

Funktion steht und das entsprechende Unterprogramm anspringen.

BASIC-Zeilen mit Zeilennummer werden ab TXTBG wie folgt abgelegt:

(TXTBG) → 00

|      |                              |
|------|------------------------------|
| AAAA | Adresse der nächsten Zeile   |
| NNNN | Zeilennummer (binär)         |
| ...  | Anweisungen                  |
| 00   | Zeilenende                   |
| AAAA | Adresse der nächsten Zeile   |
| usw. |                              |
| ...  | Letzte Anweisung             |
| 00   | Kennzeichen für Programmende |
| 00   |                              |

Direkt hinter dem Programm werden alle Variablen, Feldvariablen und Funktionen gespeichert.

| *****<br>  START BASIC<br>***** |         |        |     |             |                            |
|---------------------------------|---------|--------|-----|-------------|----------------------------|
| 0100                            | C3 3770 | START: | ORG | BASIC       |                            |
| 0103                            | C3 F003 | CI:    | JP  | INIT        | ;A=ZEICHEN TASTATUR        |
| 0106                            | C3 F006 | RI:    | JP  | MONITOR+3H  | ;A=ZEICHEN LESER           |
| 0109                            | C3 F009 | CO:    | JP  | MONITOR+6H  | ;C=ZEICHEN BILDSCHIRM      |
| 010C                            | C3 F00C | PDO:   | JP  | MONITOR+9H  | ;C=ZEICHEN STANZER         |
| 010F                            | C3 F00F | LO:    | JP  | MONITOR+OCH | ;C=ZEICHEN DRUCKER         |
| 0112                            | C3 F012 | CSTS:  | JP  | MONITOR+OFH | ;IO-TEST                   |
| 0115                            | C3 F015 | IOCHK: | JP  | MONITOR+12H | ;CONSOLSTATUS              |
| 0118                            | C3 F018 | IOSET: | JP  | MONITOR+15H | ;IO-SET                    |
| 011B                            | C3 F01B | MEMSI: | JP  | MONITOR+18H | ;HIMEM A=LO B=HI           |
| 011E                            | C3 F01E | TRAPX: | JP  | MONITOR+1BH | ;MONITOR                   |
| 0121                            | C3 3475 | FILIN: | JP  | FILINX      | ;DATEI LESEN               |
| 0124                            | C3 3478 | FIOUT: | JP  | FIOUTX      | ;DATEI SCHREIBEN           |
| 0127                            | C3 348F | FILOP: | JP  | FILOPX      | ;DATEI EROEFFNEN           |
| 012A                            | C3 347B | FILCL: | JP  | FILCLX      | ;DATEI SCHLIESSEN          |
| 012D                            | C3 3472 | FIEOF: | JP  | FIEOFX      | ;DATEI-ENDE TESTEN         |
| 0130                            | C3 1049 | USR:   | JP  | IFERR       | ;HIER USR-SPRUNG EINTRAGEN |
| 0133                            | C3 1032 |        | JP  | GETW        | ;HOLT USR-ARGUMENT         |
| 0136                            | C3 18C8 |        | JP  | FRE4        | ;UEBERGIBT FUNKTIONSWERT   |

## ;Eine Auswahl der wichtigsten RAM-Zellen

|      |         |        |      |        |                            |
|------|---------|--------|------|--------|----------------------------|
| 0139 | 3770    | LOMEM: | DEFW | INIT   | ;BEGINN BASIC-PGM          |
| 013B | EFFF    | HIMEM: | DEFW | OFFFFH | ;HIGHEST MEMORY            |
| 018B | 0000    | RUNAD: | DEFW | 0      | ;AKTUELLE PGM-ADRESSE      |
| 018D | 0000    | TXTPT: | DEFW | 0      | ;TEXTPOINTER               |
| 018F | FFFF    | RUNLN: | DEFW | OFFFH  | ;MOMENTANE ZEILENNUMMER    |
| 0195 | 0000    | STACK: | DEFW | 0      | ;SP                        |
| 0197 | 0000    | TXTBG: | DEFW | 0      | ;BEGINN DES PROGRAMMS      |
| 0199 | 0000    | VARBG: | DEFW | 0      | ;BEGINN DER VARIABLEN      |
| 019B | 0000    | ARRBG: | DEFW | 0      | ;BEGINN DER FELDER         |
| 019D | 0000    | ARREN: | DEFW | 0      | ;ENDE DER FELDER           |
| 01A1 | 0000    | XLSB:  | DEFW | 0      | ;X-REGISTER, STRINGPOINTER |
| 01A3 | 0000    | XNSB:  | DEFW | 0      |                            |
| 01A5 | 00      | XMSB:  | DEFB | 0      |                            |
| 01A6 | 00      | XEXP:  | DEFB | 0      |                            |
| 01D3 | C3 0109 | JPC0:  | JP   | CO     | ;SPRUNG ZUM AUSGABEGERAET  |
| 01D6 | C3 0103 | JPCI:  | JP   | CI     | ;SPRUNG ZUM EINGABEGERAET  |
| 01E8 | FF      | DNUCT: | DEFB | 255    | ;ANZAHL DER SYNC-ZEICHEN   |
| 01E9 | FF      | DNUCH: | DEFB | 255    | ;SYNCH-ZEICHEN             |
| 01EB | 00      | FILNR: | DEFB | 0      | ;DATEI-NUMMER              |
| 01EC | 00      | FILMD: | DEFB | 0      | ;IN/OUT                    |
| 01ED |         | FILNA: | DEFS | 33     | ;FILE-NAME USW             |
| 020E | 00      | FIBIN: | DEFB | 0      | ;BIN=0, ASCII=-1           |
| 020F | 00      | FIFLG: | DEFB | 0      | ;SAVE/LOAD FLAG            |
| 021A | C3 3770 | JPRET: | JP   | INIT   | ;RETURNSPRUNG BEI READY    |
| 0229 |         | BUFER: | DEFS | 255    | ;TEXTPUFFER                |
| 0328 |         |        | DEFS | 32     | ;FREI, PUFFERWEITERUNG     |

Bild 2.56 Anfang des BASIC-Interpreters und einige wichtige RAM-Zellen

## (VARBG)

1. Zeichen des Variablenamens, oder 00  
 Bit 7 = 0 bei numerischen Variablen  
   =1 bei Stringvariablen  
   =-1 bei Funktionsnamen

2. Zeichen des Variablenamens  
 Bit 7 = 0 bei numerischen Variablen  
   =0 bei Stringvariablen  
   =-1 bei Funktionsnamen

### 3. Zeichen des Variablenamens, oder 00

### 4. Zeichen des Variablenamens, oder 00

Jede einfache Variable benötigt einen Speicherplatz von 10 byte. Davon entfallen 4 byte auf den Namen und die restlichen 6 byte auf den Wert. Numerische Werte werden im Hidden-bit-Format gespeichert. Bei Stringvariablen wird die Länge und die Adresse der Zeichenkette gespeichert, bei selbstdefinierten Funktionen (DEF FN) ein Pointer, der auf die Übergabeparameter zeigt. Ab ARRBG sind die Feldvariablen in ähnlicher Weise abgelegt. Zusätzliche Angaben hinter dem Variablenamen geben die Größe des Feldes, die Anzahl der Dimensionen und deren Größe an.

Der Speicher für Zeichenketten (String-space) liegt direkt unter dem maximalen Speicherplatz. Darunter wird der Stack angelegt.

Der Anfang des BASIC-Interpreters und einige wichtigen RAM-Zellen sind in Bild 2.56 zu sehen.

Die Ein- und Ausgabebefehle sind komfortabler ausgelegt, als sie in Verbindung mit einem einfachen Kassettenrecorder genutzt werden können. Verfügt man z. B. über ein steuerbares Laufwerk, dann lässt sich der Interpreter mit neuen Ein-/Ausgaberoutinen ergänzen. Um diese Eingriffe möglichst einfach zu gestalten, wurden die entsprechenden Sprünge in die Sprungleiste am Anfang des Interpreters aufgenommen. Hier können andere Programme aufgerufen werden; die Dateipuffer für Ein- und Augabe verwalten, Dateien öffnen und schließen sowie auf Dateiende testen. Die dafür notwendigen Argumente und Angaben sind in den entsprechenden RAM-Zellen abgelegt.

Die Zelle FILMD gibt an, ob es sich um eine Eingabe oder Ausgabe handelt. Bei Programmspeicherbefehlen (SAVE, LOAD) ist die Dateinummer immer #4. Eine Unterscheidung, ob ASCII-Daten oder binäre Daten (z. B. bei SAVE, LOAD) übertragen werden, ist in der Flagzelle FIBIN festgehalten. Bei Bedarf können die vor jedem PRINT#-Befehl gesendeten Synchronzeichen in Anzahl und Code den eigenen Bedürfnissen angepasst werden. Initialisiert sind 255 Synchronzeichen mit dem Code FFH (Vergleiche NULL-Befehl).

## 2.10. Einige Leiterplattenentwürfe

Aus Platzgründen ist es leider nicht möglich, das Layout für einen Einkartencomputer widerzugeben. Um dennoch die Entwicklung und den Bau eines Computers zu unterstützen, werden einzelne Baugruppen aus den vorhergehenden Abschnitten, sozusagen formfrei, dargestellt. Bei entsprechender Zusammenstellung ist der Aufbau eines Einkartencomputers oder die modulare Bauweise möglich.

Die Leiterplattenentwürfe sind grundsätzlich 2seitig, aber dennoch so ausgelegt, daß man keine galvanische Durchkontaktierung benötigt.

Bild 2.57 zeigt das Layout der CPU-Baugruppe, die in Abschnitt 2.2. beschrieben wurde.

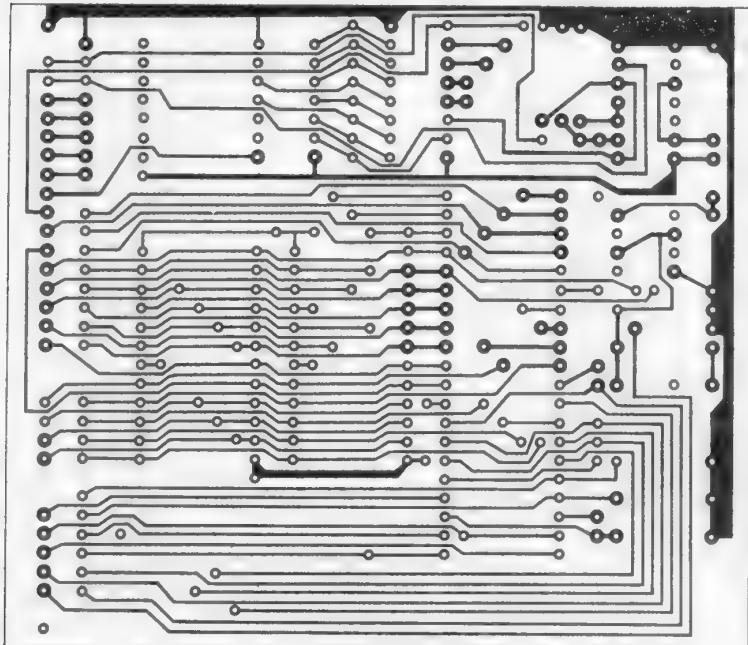
Auf der Leiterplatte nach Bild 2.58 wurden die Horizontalablenkung (Bild 2.14) und die Vertikalablenkung (Bild 2.16) vereinigt.

Die Videosignalzeugung (Bild 2.17) läßt sich auf der RAM-Platine nach Bild 2.59 unterbringen.

Mit den beiden Drahtbrücken Br1 und Br2 kann die normale oder inverse Darstellungsweise auf dem Bildschirm gewählt werden. (Achtung: Nur eine Brücke schließen!) Weiterhin enthält die Leiterplatte die in Bild 2.22 dargestellten Schaltungsdetails.

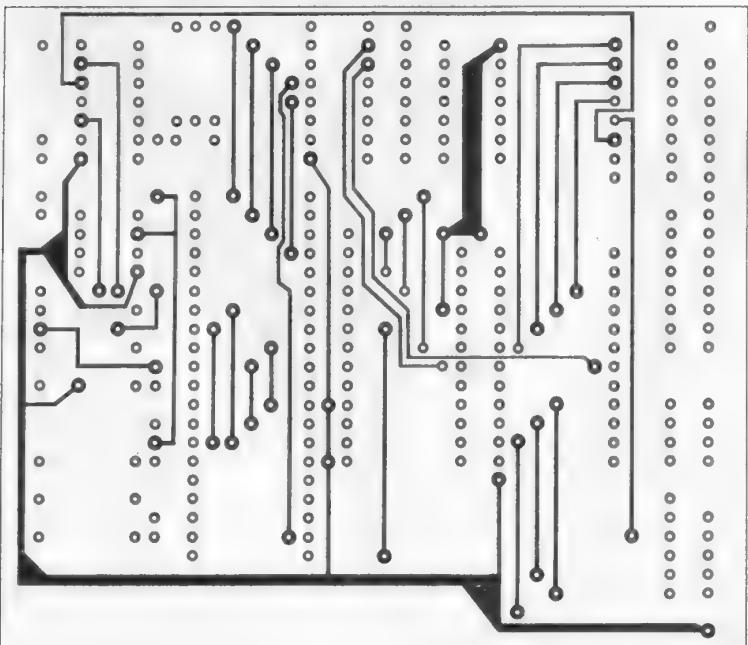
Die Speicherbaugruppe benötigt zum Betrieb die Speichersteuerung nach Bild 2.24. Das Leiterplatten-Layout dazu zeigt Bild 2.60. Im Gegensatz zur Schaltung (Bild 2.24) werden als Inverter D 100- oder DL 000-Gatter verwendet. Die mit dem Sternchen gekennzeichneten Kondensatoren sind optimal und müssen bei Bedarf experimentell ausgewählt werden.

Der letzte Leiterplattenvorschlag nach Bild 2.61 beinhaltet das Layout des Kassetteninterface aus Abschnitt 2.6.2. Die dargestellten Schalter sind DIL-Schalter und können selbstverständlich durch Wickelstifte ersetzt werden.

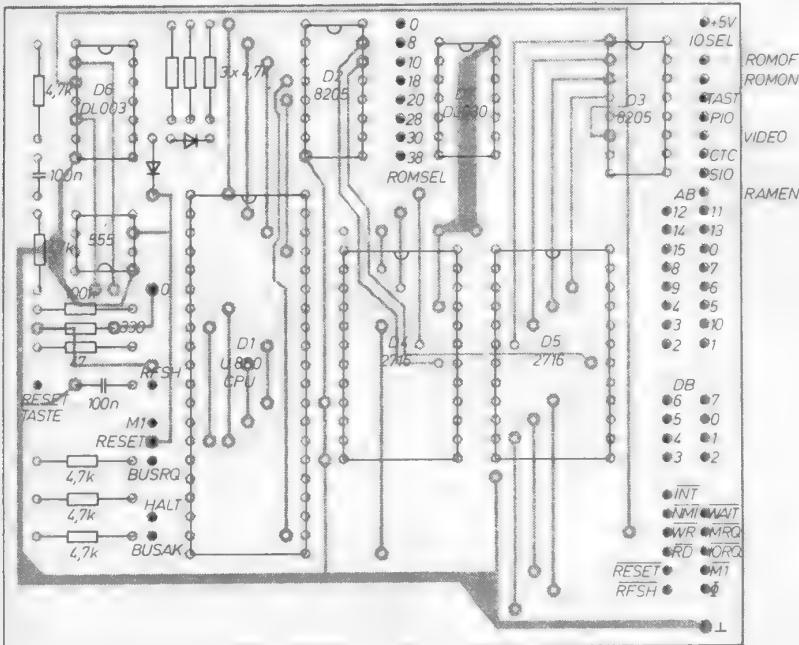


a)

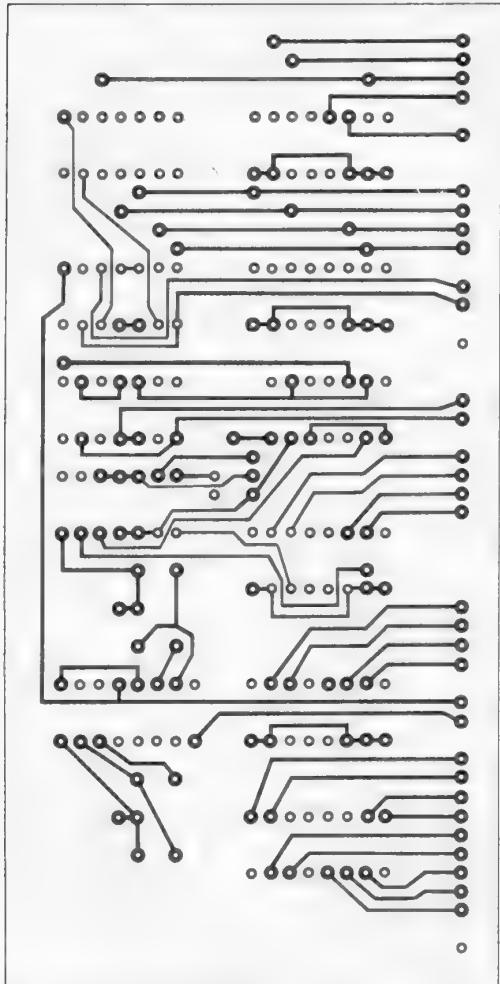
Bild 2.57  
CPU-Baugruppe nach  
Bild 2.3; a – Lötseite,  
b – Bestückungsseite,  
c – Bestückungsplan



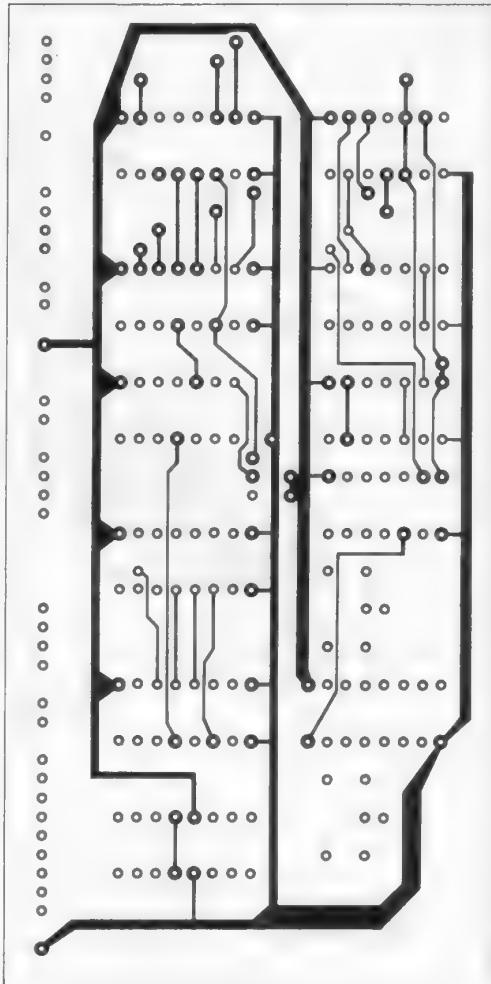
b)



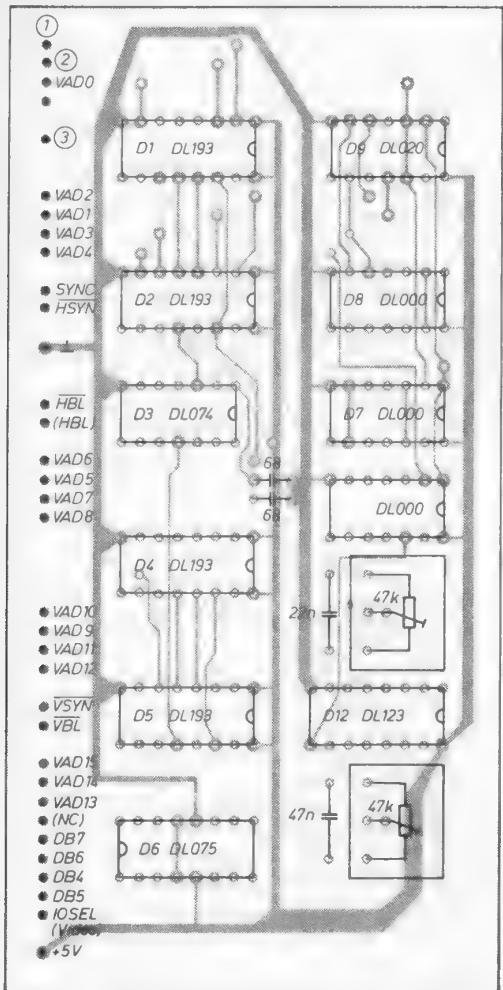
c)



a)

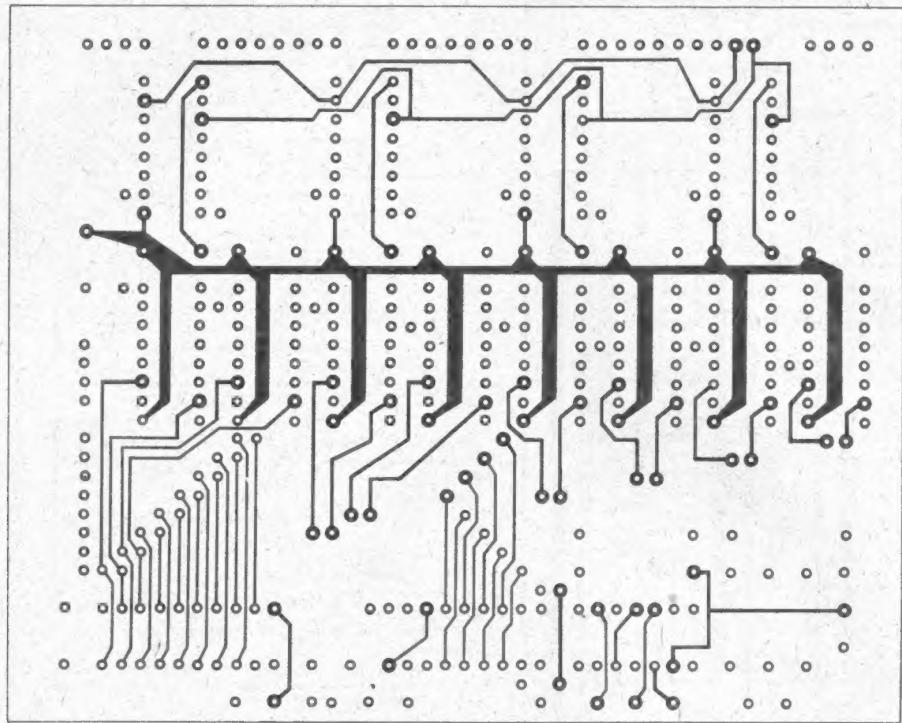


b)

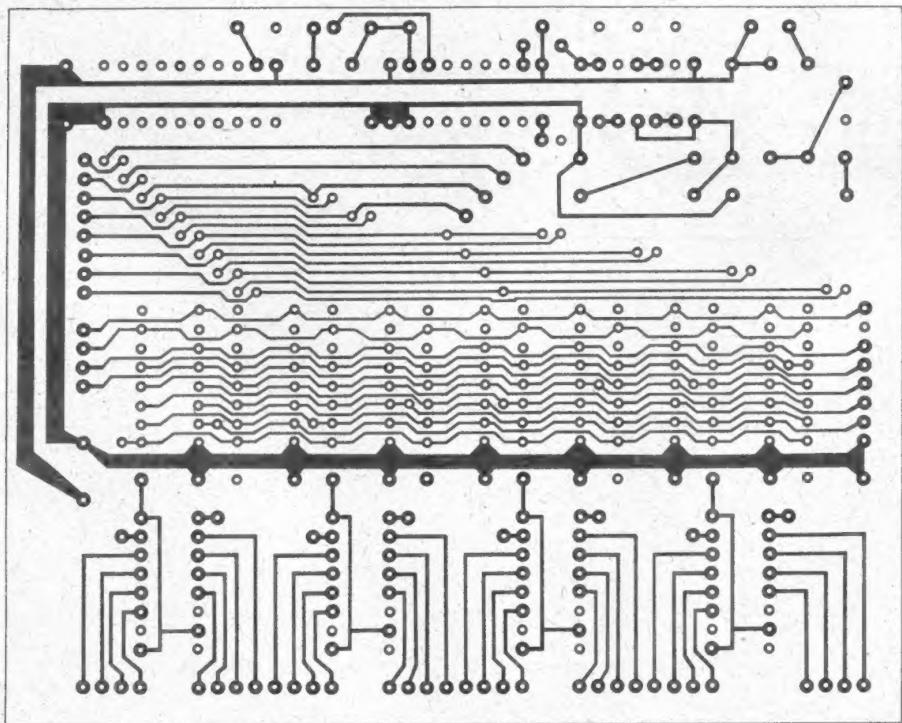


c)

**Bild 2.58**  
Videosteuerung nach Bild 2.14 und Bild 2.16; a – Lötseite, b – Bestückungsseite, c – Bestückungsplan



a)



b)

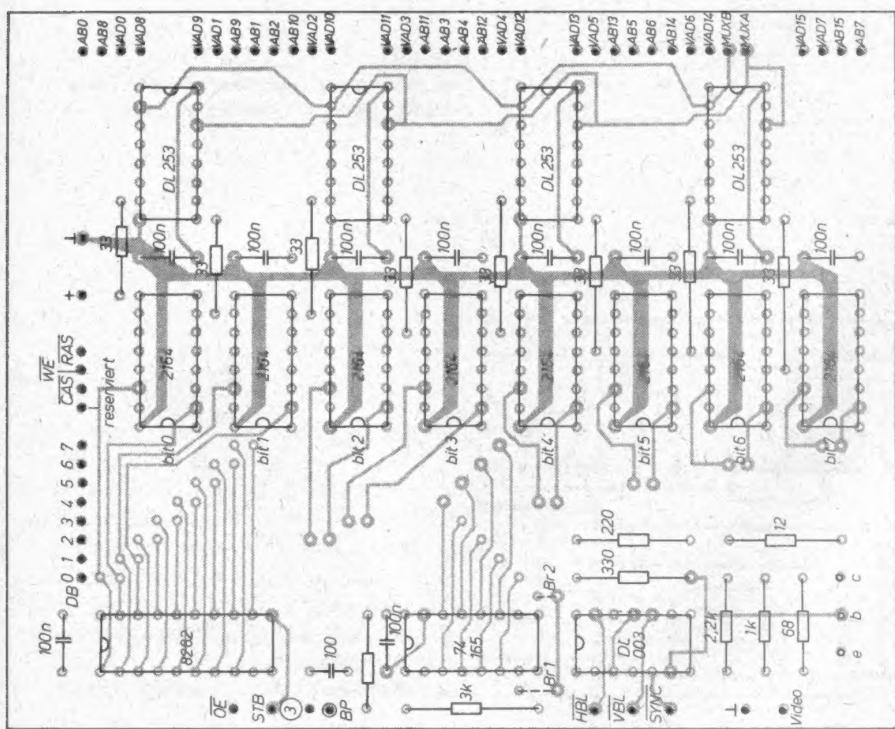
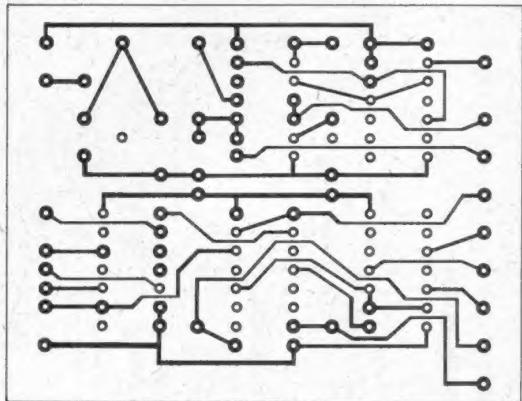
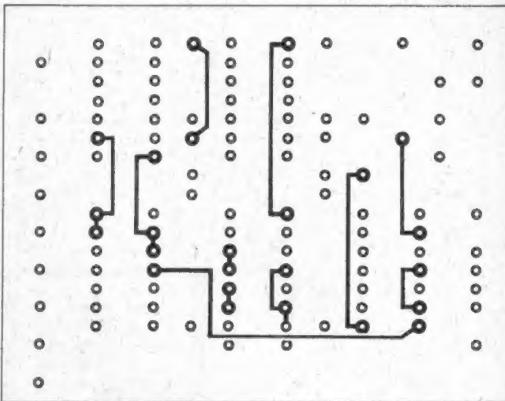


Bild 2.59  
Speicherbaugruppe nach Bild 2.17 und 2.22; a –  
Lötseite, b – Bestückungssseite, c – Bestückungsplan

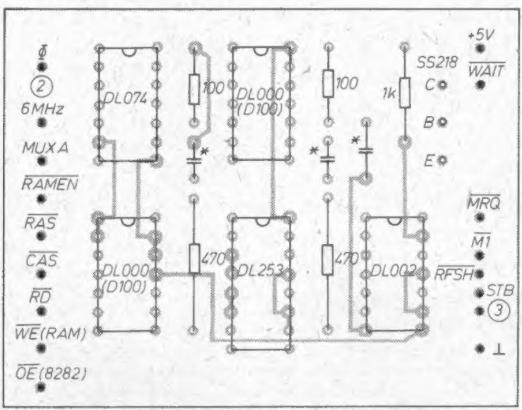
c)



a)



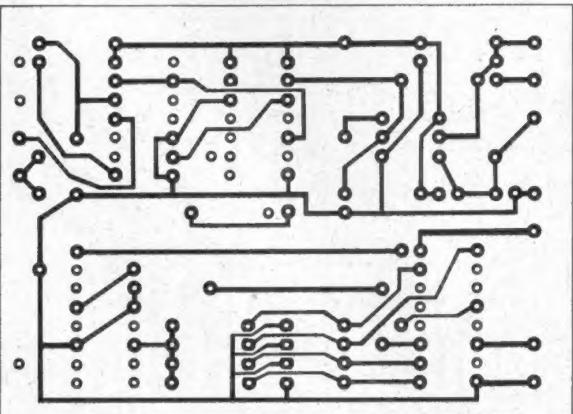
b)



c)

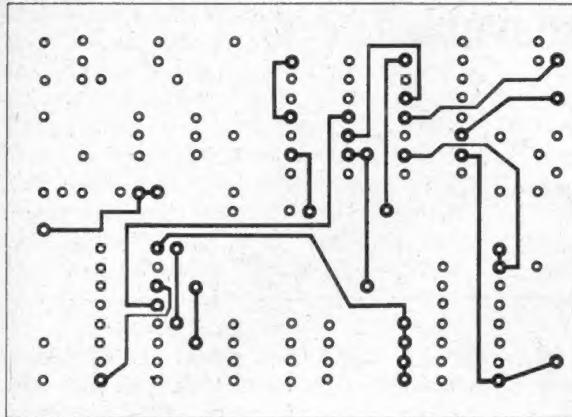
Bild 2.60

Speichersteuerung nach Bild 2.24; a – Lötseite, b – Bestückungsseite, c – Bestückungsplan

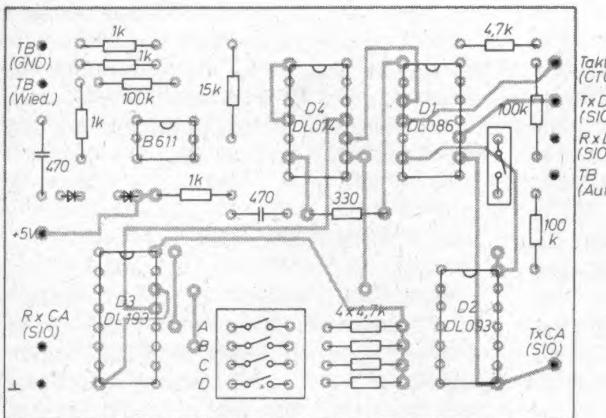


a)

Bild 2.61  
Kassetteninterface nach Bild 2.44;  
a – Lötseite, b –  
Bestückungsseite, c – Bestückungsplan



b)



c)

## 2.11. Literatur

- [1] H. Kieser/M. Meder, Mikroprozessortechnik, Berlin 1985
- [2] B. Hübner/K.-P. Evert, Ausbaufähiger Mikrocomputer mit dem U 880, Berlin 1985
- [3] B. Lampe/G. Jorke/N. Wengel, Algorithmen der Mikrorechentechnik, Berlin 1983
- [4] G. Wärme/V. Otto/B. Graffunder, Mikrorechneranwendung-Gerätetechnik U 880, Berichte zur Nachrichtentechnik (Band 13), Institut für Nachrichtentechnik, Berlin 1982