

- D1, D19 : D174
- D2 : D104
- D3 : UB 880
- D4 : DS 8205
- D5 : U 40098 D
- D6 bis D9, D21 : U 555 C
- D10 bis D17 : U 202
- D18 : DS 8212
- D20 : UB 857
- D22, D23 : D 121
- D24, D25 : D 195
- D26 : D 100

Bild 1: Stromlaufplan des Heimcomputers BCS 3

Wenn ja, wird aus der Zeilenanfangstabelle im EPROM die Adresse der Zeile im Bildspeicher entnommen. Der Bildspeicher ist Bestandteil des RAM, wird aber mit den Adressen 3800H bis 3BFFH angesteuert. Das Programm liest dann die Zeile 1400H im Speicher. Über den Ausgang A₅ von D₄ wird ein WAIT-Impuls bis zum nächsten Zeilenimpuls ausgelöst. Dadurch beginnt die Anzeige immer an der gleichen Stelle. Nach drei Füllbefehlen springt das Programm in den Bildspeicher. Dazu liefern der RAM den Zeichenkode und der Zeichengenerator D₂₃ das Bitmuster der Zeichenlinie. Letzteres wird mit der Rückflanke von MREQ in die Schieberegister D₂₄ und D₂₅ übernommen und mit dem doppelten Takt zum Modulator herausgeschoben. Mit R₃₄ wird die Länge des Impulses so eingestellt, daß ein sauberes Bild entsteht. Das Bitmuster des Zeichengenerators entspricht dem des U 402. Zusätzlich wurden die Kleinbuchstaben programmiert. Wenn Bit 7 des Zeichenkodes 0 ist, wird an D₁₈ Clear ausgelöst, und der Prozessor liest 00=NOP. Der Befehlszähler wird erhöht und das nächste Zeichen gelesen, bis am Zeilenende Bit 7 gleich 1 ist. Dann entfällt das Clear, der Prozessor liest einen RET-Befehl, und das Programm kehrt in die Interruptroutine zurück. Dieser Ablauf wiederholt sich für jede Zeichenlinie einmal. Der Zyklus dauert genau 160 Takte, so daß die Anzeige immer an der gleichen Stelle auf dem Bildschirm beginnt. Programmiert sind zwölf Zeilen zu je 27 Zeichen. Der Prozessor ist damit zu 40% ausgelastet. Mit einer Taktfrequenz von 3,25 MHz lassen sich 40 Zeichen je Zeile darstellen. Das bedingt aber die Verwendung schnellerer Bauelemente (UA 880). Außerdem muß das Programm entsprechend geändert werden (s. Bildschirm-Interruptroutine).

1	"	#	x	%	&	'	()	
@	Q	W	*	E	R	T	Z	U	i
A	S	D	+	G	-	H	/	J	K
SHIFT	Y	X	C	V	B	=	<	SP	ENTER

Bild 2: Leiterseite der Platine

Tastatur

Die Tastatur ist sehr einfach aufgebaut (Bild 2). 40 Schalter bilden eine 4x10-Matrix. Zur Auswahl einer Matrixzeile wird genau eine der Adreßleitungen A0 bis A9 auf L geschaltet. Dann werden über das CMOS-Tristategatter U 40098 die Spalten abgefragt. Aus Zeilen- und Spaltennummer wird der Zeichenkode berechnet. Bei gleichzeitigem Drücken von SHIFT lassen sich Sonderzeichen eingeben. Als Tasten lassen sich Folietasten von Taschenrechnern, Mikrotaster oder auch Halltasten verwenden. Bei Letzteren müssen die Adressen negiert werden, da diese mit H selektiert sind. Für die Tastenköpfe wurden quadratische Plasteplatten, die aus einem Plastetabrett herausgesägt wurden, verwendet.

Hinweise zum Bau

Für alle IS außer den RAMs und dem Zeichengenerator können Bastlertypen verwendet werden. Der Heimcomputer wurde auf einer Möbelspanplatte montiert. Die genaue Größe richtet sich nach dem Netzteil und der Tastatur. Der Netzteil nutzt für die Spannungen von 5 V und 12 V die Festspannungsregler MA 7805 und MA 7812. Für -5 V wurde eine Z-Diode SZ 600/5,1 verwendet. Der Verbrauch beträgt 5 V/800 mA, 12 V/400 mA und -5 V/200 mA (mit fünf EPROMs). Wenn der Rechner später er-

weitert werden sollte, müssen entsprechende Reserven vorgesehen werden. Die Leiterplatte (Bilder 3 bis 5) wird zunächst gebohrt. Dabei leistet eine Lochrasterplatte als Schablone gute Dienste. Dann werden die Leiterzüge mit einer 0,6-mm-Röhrchenfeder gezeichnet. Der Abdecklack aus dem Atzsatz kann mit Kopierstiftmine angefärbt werden. Das Ätzsatz ist in destilliertem Wasser zu lösen. Geätzt wird bei 80°C. Nach der Sichtkontrolle wird die Leiterplatte mit einer Kolophonium-Spiritus-Lösung lackiert. Dann wird die gesamte Leiterplatte bestückt. Die engen Leitungen für die Speicher werden auf Kurzschluß und Durchgang geprüft. Für die EPROMs empfehlen sich Fassungen. Da diese auf der Bestückungsseite nicht gelötet werden können, müssen die Anschlüsse vorher mit dünnen Drähten durchkontaktiert werden. Nun kann die Stromversorgung angeschlossen werden. Ohne EPROMs wird der Verbrauch gemessen (bei 5 V etwa 700 mA). Mit einem Oszilloskop können Taktgenerator und CPU kontrolliert werden. Nun wird die Leiterplatte mit dem ersten EPROM bestückt. Nach dem RESET müssen Bild- und Zeilenimpulse am Modulator anliegen. Jetzt wird der Fernsehempfänger über ein geschirmtes Kabel angeschlossen und im Band I der Sender gesucht (Abgleich mit C₄). Die Spule des Taktgenerators wird so eingestellt, daß ein stehendes Bild entsteht (zwölf waagerechte schwarze Balken auf weißem Grund). Nun werden die anderen EPROMs in die Fassungen gesteckt. Auf dem Bildschirm erscheint nach RESET die Ausschrift

BASIC-SE 2.4.

>

Alle anderen Zeichenstellen enthalten einen Punkt. Wenn die Buchstaben flak-

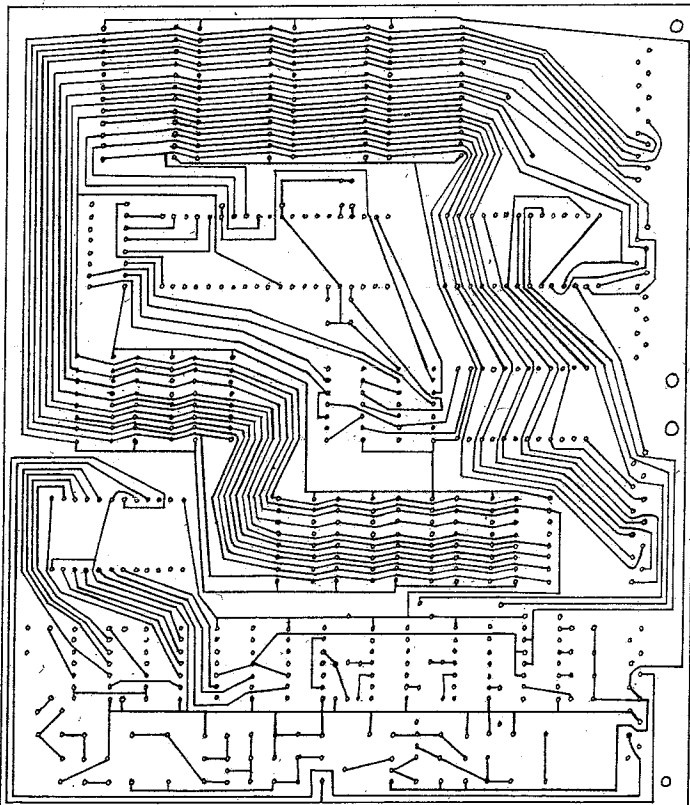


Bild 3: Bestückungsseite der Platine

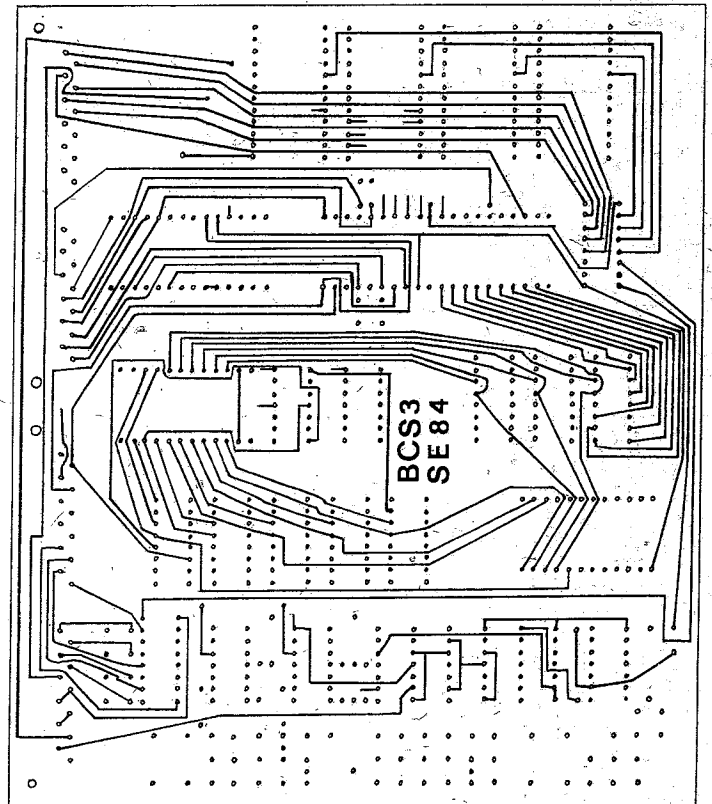


Bild 4: Bestückungsplan

kern, wird R₃₄ durch einen Einsteller ersetzt. Von 0 beginnend, wird der Widerstand so lange vergrößert, bis das Flakern verschwindet. Nun wird die Tastatur angeschlossen, und der Rechner ist betriebsbereit.

Betriebssystem des BCS 3

Das Betriebssystem des Rechners besteht aus einer Gruppe von Unterprogrammen, die die Tastatur und den Bildschirm steuern.

- **Initialisierung:** Dieses Programm wird nach jedem RESET durchlaufen. Der CTC wird programmiert, und anschließend wird zur Adresse 15AH gesprungen (Beginn des Basic-Interpreters).
- **LOBS (Adresse 02CH)** Löschen des Bildschirms, Cursor auf 0: Der Bildschirm wird mit Punkten gefüllt. Als Zeilenende werden die RET-Befehle (0E0H) abgespeichert.
- **BSE (088H)** Bildschirmausgabe eines Zeichens: Das Zeichen steht im A-Register. Es wird an der Stelle des Cursors abgespeichert. Der Cursor wird um eine Stelle nach rechts verschoben. Das Zeichen 1EH bedeutet Zeilenschaltung.
- **TSS (131H)** Tastatureingabe einer Zeichenkette: Die Eingabe erfolgt in den Bildspeicherbereich. Sie wird mit ENTER abgeschlossen. Die Adresse des ersten eingegebenen Zeichens steht dann in DE, die des Cursors in HL.

Die Unterprogramme belegen zusammen mit dem Bildschirm-Interruptprogramm den Bereich von 0 bis 14DH. Mit diesen Angaben wird es möglich, den BCS 3 mit anderen Programmen zu betreiben oder den Basic-Interpreter auf anderen Rechnern zu implementieren.

Erweiterungen

Der vorgestellte Rechner ist eine Minimalvariante. Um ihn später erweitern zu können, wurden einige Voraussetzungen geschaffen: Auf der Leiterplatte sind alle wichtigen Anschlüsse des Prozessors auf die beiden Steckerleisten herausgeführt. Vor allem können zusätzliche RAMs (U 202 oder U 256) angeschlossen werden. Im Basic-Interpreter wird festgestellt, wie lang der Speicherbereich ist. Von 3DA1H beginnend wird getestet, ob eine RAM-Zelle vorliegt. Bei der LIST-Ausgabe wird anschließend angezeigt, wieviele Speicherplätze zur Verfügung stehen. Bei Speichererweiterungen ab 4000H muß das Programm also nicht geändert werden. Von den EPROMs benötigt der Interpreter nur D₆ und D₇. Damit bleiben D₈ und D₉ für eigene Programme frei.

Um die Programme nicht jedesmal neu eingeben zu müssen, ist es günstig, sie auf Magnetkassette auszulagern. Dafür wurden die Basic-Kommandos SAVE und LOAD vorgesehen. Die LOAD-Routine beginnt bei 800H, SAVE bei 803H. Das Programm ist also nicht in der Grundversion enthalten und muß vom Anwender selbst erstellt werden. Dafür wurde der Q-Ausgang von D_{1,2} vorgesehen. Als Eingang kann TU genutzt werden. Von 3DA1H ab muß ausgelagert werden. Die Programm länge steht in 3C06/7H. Die Rückkehradresse ist 193H (Kommandoingabe) oder

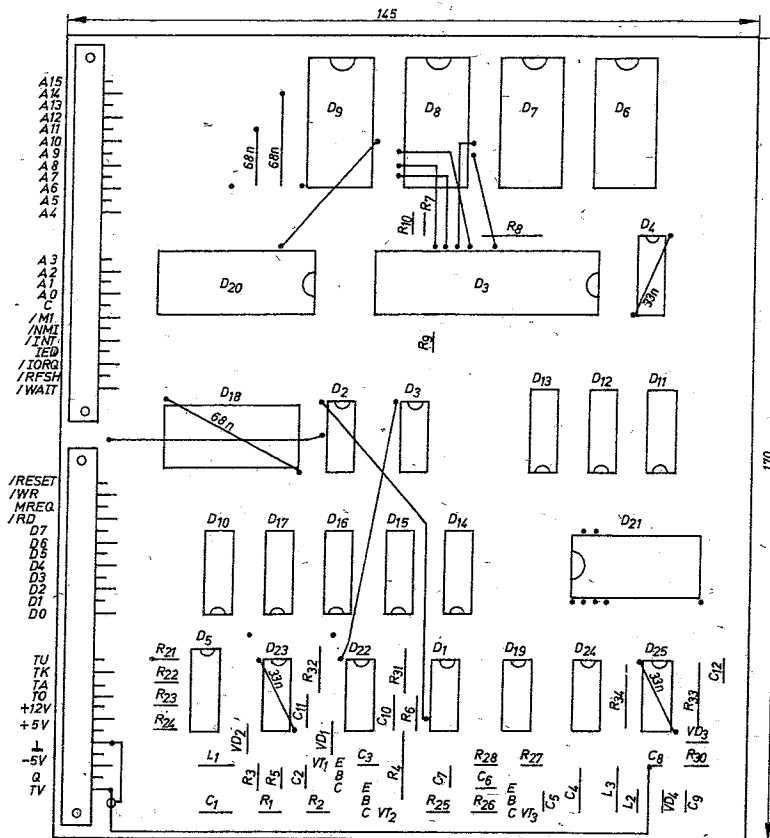


Bild 5: Günstige Anordnung der Tasten

190H (Initialisierung). Auch weitere PIO-, CTC- und SIO-Schaltkreise lassen sich anschließen und über IN und OUT steuern. Tafel 4 gibt wichtige Adressen, Tafel 5 die Bildschirminterruptroutine und Tafel 6 die Aufstellung des Zeichenkodes an.

Beschreibung des Basic-SE 2.4

Allgemeines

Basic-SE ist eine problemorientierte Programmiersprache. Sie wurde speziell für kleine Mikrorechnersysteme wie den BCS 3 entwickelt.

Basic wird meist interpretativ abgearbeitet, d. h., die Befehle stehen als lesbarer Text im Rechner und erst während der Verarbeitung wird der Text in Maschinenbefehle umgesetzt. Das hat den Vorteil, daß das Programm vor der Abarbeitung nicht überprüft werden muß und jederzeit geändert werden kann. Der Nachteil liegt in der längeren Rechenzeit. Da eine Anweisung in 1...2 ms ausgeführt wird, spielt das aber keine große Rolle. Um Speicherplatz zu

Tafel 4: Wichtige Adressen

EPROM D ₆ :	0H bis 3FFH
D ₇ :	400H bis 7FFH
D ₈ :	800H bis 0BFFH
D ₉ :	0C00H bis 0FFFH
CTC D ₂₀ :	0F8H - Kanal 0
	0FAH - Kanal 2
	0FBH - Kanal 3
	0F9H - Kanal 1
Aufteilung des RAM	
3C00/1H	Adresse des RAM-Endes
3C06/7H	Programmlänge in Byte
3C08/9H	Adresse des Cursors
3C50H	oberes Kellerende
3C50H bis 3D9FH	Bildspeicher
3DA1H	Beginn des Programms

sparen, wird jedes Basic-Schlüsselwort intern durch ein Kodezeichen dargestellt. Basic ist zeilenorientiert. Jede Zeile beginnt mit einer laufenden Nummer zwischen 1 und 9999. 9999 ist für die END-Zeile reserviert. Die Zeilen werden in Reihenfolge der Numerierung abgearbeitet. Die Reihenfolge der Eingabe ist ohne Bedeutung.

Tafel 5: Bildschirm-Interruptroutine

ORG	46H
BINT: EXAF	
IN	0FAH; Kanal 2 CTC
CPL	;Negieren
ADD	16
ADD	A ;Verdoppeln
CMP	24 ; Sprung zum Ende, wenn
JRNC	E2 ;Zeilennummer > 2x12
EXX	
LD	HL,AL; Berechnung der Adresse
ADD	L ;in der Liste
LD	L,A
LD	B,M
INC	L
LD	H,M; Laden der Zeilenadresse,
LD	L,B
LD	DE,RADR; Rückkehradresse
LD	B,8; Anzahl der Zeichenlinien
LD	A,(1400H); WAIT-Auslösung
NOF	; 3 Füllbefehle
ZZ: LD,	A,I
NOF	
PUSH	DE
JMP	M; Sprung in Bildspeicher
RADR: DJNZ	ZZ; Linienzählung
EXX	
E2: EXAF	
EI	
RET	; Rückkehr vom Interrupt
ALI: DA	3850H; 1. Zeile
DA	386CH; 2. Zeile
:	
DA	3984H; 12. Zeile

Tafel 6: Aufstellung der Zeichenkodes

Ta	Ko	T+S	Ko	Ta	Ko	T+S	Ko	Ta	Ko	T+S	Ko	Ta	Ko
0	30	SP	20	A	41	&	26	K	4A	:	3A	U	55
1	31	!	21	B	42	'	27	L	4C	;	3B	V	56
2	32	"	22	C	43	(28	M	4D	<	3C	W	57
3	33	#	23	D	44)	29	N	4E	=	3D	X	58
4	34	*	24	E	45	*	2A	O	4F	>	3E	Y	59
5	35	%	25	F	46	+	2B	P	50	?	3F	Z	5A
6	36	&	26	G	47	,	2C	Q	51	@	40	SP	20
7	37	'	27	H	48	-	2D	R	52	A	41	←	5C
8	38	(28	I	49	.	2E	S	53	B	42	ENTER	5D
9	39)	29	J	4A	/	2F	T	54	C	43	SHIFT	5E

Erläuterung: Ta = Taste, Ko = Kode (hexadezimal), T+S = Taste + SHIFT, SP = SPACE (Leerzeichen), SHIFT = Umschaltung Buchstaben - Zeichen, ENTER = Ende der Eingabe, ← = Rückschritt, Löschen des letzten Zeichens

Grundelemente von Basic

Zeichen

Buchstaben: A, B, C, ..., X, Y, Z

Ziffern: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

arithmetische Operatoren: +, -, *, /

Vergleichoperatoren: <, =, >

logische Operatoren: AND, OR

Sonderzeichen: ,, ;, ', #, /, ENTER

Konstanten

In Basic-SE können ganze Zahlen im Bereich von -32627 bis 32627 und Hexadezimalzahlen von 0 bis 0FFFFH dargestellt werden. Hexadezimalzahlen müssen mit einer Ziffer beginnen. Die ganzen Zahlen werden intern mit 15 bit Betrag und 1 bit Vorzeichen dargestellt, also nicht im Zweierkomplement. Daneben gibt es noch logische Konstanten. Sie werden auch mit 16 bit dargestellt. Ungleich 0 ist WAHR und gleich 0 ist FALSCH.

Namen

Namen bestehen aus ein oder zwei Buchstaben. Die Namen IF, OR und alle mit H beginnenden sind verboten.

Variablen

Variable sind durch ihren Namen gekennzeichnete Größen, denen im Verlauf der Programmabarbeitung unterschiedliche Werte zugeordnet werden können. Für jede Variable wird bei ihrem ersten Auftreten im Programm ein Bereich von vier Byte reserviert. Zwei Byte enthalten den Namen und weitere zwei den Wert. Der Anfangswert ist immer 0 (nach dem Speicherlösch). Die Bereiche die Variablen werden vom Speicherende abwärts geführt.

Arithmetische Ausdrücke

Ein Ausdruck ist eine Folge von Operanden und Operatoren. Als Operanden können Konstanten, Variablen, Funktionen oder Ausdrücke in Klammern auftreten. Die Operationen werden in folgender Reihenfolge ausgeführt, wenn nicht durch Klammerung eine andere Reihenfolge erzwungen wird: *, /, -, +, OR, AND, >, <, =. Gleiche Operationen werden von links nach rechts ausgeführt. Die Division wird immer abgerundet.

Beispiel:

mathematisch	Basic
x-b	X-B
2c	2*C
a ²	A*A
x/yz	X/Y/Z
a+bd	A+B*D
(a+b)c	(A+B)*C

AND und OR werden bitweise durchgeführt. Ihre Anwendung ist nur bei Hexadezimalzahlen oder logischen Ausdrücken sinnvoll:

Beispiel: 0FF0H AND 0333H ergibt 0320H
0FF0H OR 0333H ergibt 0FF3H

<, > und = liefern die logischen Werte 0 und 1.

Beispiel: 1=2 ergibt 0
1>2 ergibt 0
1<2 ergibt 1

Zufallsgenerator RND

RND (Ausdruck) ist Bestandteil von Ausdrücken. Er liefert eine Zahl zwischen 1 und Ausdruck (<256).

Beispiel:
RND(6) - Würfel
10 IF RND(6)-6=0 PRINT 'SECHS'

Programmzeile

Ein Basicprogramm besteht aus einer Folge von Programmzeilen, die jeweils eine Zeilennummer, Basicanweisungen und Kommentare (durch ; getrennt) enthalten. Um Korrekturen einfügen zu können, empfiehlt sich ein Zeilennummernabstand von 10. Leerzeichen können eingegeben werden, sie werden aber nicht abgespeichert. Bei der LIST-Ausgabe wird automatisch hinter jedes Schlüsselwort ein Leerzeichen eingefügt. Eine Zeile hat eine Länge von maximal 27 Zeichen. Zeile 9999 END wird automatisch angefügt.

Basicanweisungen

Ergibtanweisung

Durch die Ergibtanweisung
LET Variable = Ausdruck

wird der links stehenden Variable der rechts stehende Wert zugeordnet. Dieser kann auch ein Vergleichsausdruck sein. Das LET kann weggelassen werden.

Beispiel: A=1;B=A+2;C=A OR B;
LET D=I>10

Sprunganweisung

Mit GOTO wird der Ablauf des Programms unterbrochen und an der angegebenen Zeilennummer fortgesetzt. In Unterprogramme darf nicht hinein- oder herausgesprungen werden. Die Zeilennummer kann auch als Ausdruck angegeben werden.

Beispiel: 50 GOTO 100
60 N=M+L
:
80 I=200
90 GOTO I-140
100 A=B
:

Kommentare

Der der REM-Anweisung folgende Text wird bis zum nächsten ; oder ENTER als Kommentar gewertet.

Bedingte Anweisung

Wenn der Ausdruck einen Wert ungleich 0 ergibt (logisch WAHR=1) wird die folgende Anweisung ausgeführt, sonst nicht. Das THEN kann weggelassen werden.

Beispiel: Summation der Zahlen von 1 bis 100

```
10 REM 'SUMME 1 - 100'
20 S=0;I=1
40 S=S+I;I=I+1
60 IF I<101 GOTO 40
70 PRINT S
```

Unterprogramm anweisung

Programmabschnitte, die häufig vorkommen, können als Unterprogramme geschrieben werden. Der Aufruf erfolgt über die Anweisung GOSUB mit der entsprechenden Zeilennummer. Unterprogramme dürfen mehrere Ein- und Ausgänge haben, dürfen aber nicht mit GOTO aufgerufen werden. Ein RETURN ohne vorheriges GOSUB führt zum Fehler (ohne Zeilenangabe).

Beispiel: Das Programm berechnet die Anzahl der Tage, die zwischen zwei Daten liegen (JAHR<89).

```
10 REM'DIFFERENZ'
20 PRINT'ERSTER TAG'
30 GOSUB100;E=A
40 PRINT'LETZTER TAG'
50 GOSUB100
60 PRINT A-E,'TAGE';END
100 INPUT/,,'TAG',A
110 INPUT/,,'MONAT',B
120 INPUT/,,'JAHR',C
130 A=A+C*365
140 IF B<3 GOTO170
150 A=A-(B*4+23)/10
160 C=C+1
170 A=A+B*31+(C-1)/4
180 RETURN
```

Eingabe

INPUT ist eine Eingabe über Tastatur und Ausgabe auf den Bildschirm. Mit # werden so viele Leerzeichen ausgegeben, wie der Ausdruck ergibt. Der Text zwischen den Hochkommas wird ebenfalls ausgegeben. Er darf auch Leerzeichen enthalten.

Bei einem Namen wird eine Eingabe erwartet. Mit dem Zeichen ENTER wird die Eingabe abgeschlossen. Nur die letzten fünf eingegebenen Ziffern werden berücksichtigt. Für das Zeichen / wird eine Zeilenschaltung ausgeführt. Ohne das Zeichen / wird am Zeilenende keine neue Zeile begonnen. Die Reihenfolge der Anweisungen ist beliebig.

Beispiel: 10 INPUT 'A=#,1,A,'DANKE'
Bildschirm: A= 12345 DANKE

Bildschirmausgabe

#,/, 'Text' arbeitet wie bei INPUT. Ausdrücke werden berechnet und ausgegeben.

Beispiel: 10 A=1234
20 I=4
30 PRINT 'A=#,I,A-1
:
Bildschirm: A= 1233

Steht vor einem Ausdruck das Wort BYTE, so wird der L-Teil als Hexadezimalzahl ausgegeben.

PRINT BYTE PEEK(1000H)
—Anzeige von Zelle 1000H

Bildschirmlöschen
Mit CLEAR wird der Bildschirm gelöscht,
der Cursor steht oben links.

Programmende
Mit END wird das Programm beendet und
zur Kommandoingabe zurückgekehrt.

Hardwarezugriff

Basic-SE bietet die Möglichkeit, alle Ressourcen des Rechners zu nutzen. Dafür existieren einige Anweisungen, die direkt auf die Hardware zugreifen.

Speicherschreiben
Mit POKE kann der Speicherinhalt geändert werden. Der erste Ausdruck gibt die Adresse an und die folgenden Ausdrücke den neuen Inhalt dieser und der folgenden Speicherzellen. Nur die niederen acht Bit jedes Ausdrucks werden abgespeichert.

Beispiel: Ausgabe des Zeichenvorrats auf den Bildschirm
10 REM ZEICHENVORRAT
20 I=20H;Z=3C88H;K=0
30 POKE Z,I
40 I=I+1;Z=Z+1;K=K+1
50 IF I=80H END
60 IF K<16 GOTO 30
70 Z=Z+12;K=0;GOTO 30

Speicherlesen
PEEK ist keine Anweisung, sondern Bestandteil von Ausdrücken. Es liefert den Inhalt der Speicherzelle, deren Adresse sich aus dem Ausdruck ergibt.

Beispiel:
10 D=PEEK(1000H)
20 POKE 1000H,D+1
(Inkrement der Zelle 1000H)

Eingabe
IN ist wie PEEK Bestandteil von Ausdrücken. Es wird ein U-880-Eingabebefehl ausgeführt mit der Adresse der niederen 8 Bit des Ausdrucks. Das Ergebnis ist ein 8-bit-Wort. Die oberen 8 Bit sind 0.
30 K=IN(24H)AND 80H

Ausgabe
Mit OUT werden ein oder mehrere Ausgabebefehle ausgelöst. Der erste Ausdruck gibt die Adresse, die folgenden die auszugebenden Daten (8 Bit) an.

Beispiel: Drei Ausgaben zu Adresse 43H
40 OUT 43H,OFFH,0F7H,0B3H

Kommandos
Nach dem RESET des Rechners meldet sich das Programm mit der Ausschrift Basic-SE. Dann wird in der nächsten Zeile ein > ausgegeben, d. h., ein Kommando wird erwartet. Das Endezeichen jeder Kommandoingabe ist ENTER.

Jede Anweisung kann auch als Kommando sofort ausgeführt werden, wenn sie ohne Zeilennummer hinter dem > eingegeben wird. Andererseits können die folgenden Kommandos auch im Programm stehen. Allerdings wird das Programm nach der Ausführung eines Kommandos abgebrochen.

- NEW: Löschen des alten Programms, erfolgt nach dem Einschalten automatisch
- Zahl der Anweisung;Anweisung;... EN-

TER: Eingabe einer Programmzeile (maximal 27 Zeichen).

Die Zeilennummer wird auf sechs Stellen erweitert. Wenn dabei Zeichen verschwinden, stört das nicht weiter. Wird eine vorhandene Zeilennummer eingegeben, so wird diese Zeile gelöscht (Kennzeichen:S). Alle erkannten Schlüsselwörter werden in Kleinbuchstaben umgewandelt (auch bei LIST).

- RUN: Start des Programms, als erstes werden der Bildschirm und die Variablen gelöscht.
- LIST: Ausgabe des Programms auf den Bildschirm.
Nach jeweils elf Zeilen muß eine Taste zur Fortsetzung der Ausgabe gedrückt werden. Als letztes wird die Zahl der freien Stellen ausgegeben.

Wenn sich das Programm in einer endlosen Schleife befindet, kann mit RESET zur Kommandoingabe zurückgekehrt werden. Dabei bleibt das Programm erhalten.

Fehler
Bei syntaktischen Fehlern wird die betreffende Zeile mit einem F ausgegeben und das Programm abgebrochen.

Beispielprogramme
Mondlandung
Der Bediener soll die Fähre weich auf dem Mond landen. Dazu werden die Zeit S in s, der Treibstoff T in l, die Entfernung E zum Mond in m und die Sinkgeschwindigkeit G in m/s angezeigt. Der Treibstoffverbrauch V muß in l/s (max. 100 l/s) eingegeben werden. Das Ziel besteht in der Geschwindigkeit 0 bei E=0. Das Programm zeigt Tafel 7.

Tafel 7: Programm Mondlandung

```
10 REM MONDLANDUNG
20 S=0;T=500
40 E=1000;G=100
60 PRINT 'S',S,'T',T,'L G'
70 PRINT G,'M/S',/
72 PRINT 'E',E,'M V',/
75 INPUT V,/
80 IF V>100 GOTO 1000
90 IF V>T GOTO 1000
100 A=3-10*V/(20+T/100)
110 IF G>-1*A GOTO 200
150 X=E+G*G/(2*A)
160 IF X<0 GOTO 1100
200 E=E-G-A/2
210 G=G+A;S=S+1;T=T-V
240 IF E>0 GOTO 60
250 GOTO 1110
1000 PRINT 'SCHUMMLERK',/
1010 GOTO 60
1100 E=X
1110 E=E/-2
1120 IF E>0 GOTO 1200
1130 PRINT 'SAUBER!',;END
1200 IF E>1 GOTO 1300
1210 PRINT 'ETWAS HART',;END
1300 PRINT 'BRUCH,KRATER'
1310 PRINT E,'M TIEF'
```

Zahlenraten
Es ist eine vierstellige Zahl zu raten, die der Rechner vorgibt. Wenn eine richtige Ziffer geraten wurde, wird ein — ausgegeben, wenn sie auch noch an der richtigen Stelle steht, ein +. Die Ziffern können auch doppelt auftreten. Das Programm zeigt Tafel 8.

Tafel 8: Programm: Zahlenraten

```
10 REM ZAHLENRATEN
20 A=RND(9)
30 B=RND(10)-1
40 C=RND(10)-1
50 D=RND(10)-1
60 INPUT 'Z:',Z
70 IF Z>9999 GOTO 60
80 IF Z<1000 GOTO 60
90 V=1000*A+100*B+10*C+D
100 IF V=Z GOTO 2000
110 U=V-(V/10)*10
120 T=Z-(Z/10)*10
130 IF U=T GOTO 1000
140 IF (T=A) OR (T=B) GOTO 500
150 IF (T=C) OR (T=D) GOTO 500
160 V=V/10;Z=Z/10
180 IF Z>0 GOTO 110
190 X=X+1
200 PRINT X;GOTO 60
500 PRINT '-';GOTO 160
1000 PRINT '+';GOTO 160
2000 PRINT 'RICHTIG'
```

Literatur

- [1] Hopper, R.: Experimentiermikrorechner. Funkamateur, Berlin 32 (1983) 8, S. 378 bis 33 (1984) 3, S. 120 und 121
- [2] Schiller, E.: U-880-System mit minimalem Aufwand. radio fernsehen elektronik, Berlin 32 (1983) 3, S. 154-156
- [3] Schindler, S.: Heimcomputer Z 9001. radio fernsehen elektronik, Berlin 33 (1984) 3, S. 148 und 149
- [4] Kieser, H.; Meder, M.: Mikroprozessortechnik. Berlin: VEB Verlag Technik 1982
- [5] Smutný, T.: Programování mikro počítače 3 PR 1. Amatérské Radio řada B, Prag 32 (1983) 2, S. 69-75
- [6] Strelöcke, K.; Hoffmann, P.: Dialogprogrammiersprache BASIC. Berlin: Verlag Die Wirtschaft 1982

Fortsetzung von Seite 12

- [4] Schlegel, W. E.; Blodszun, A.: Leipziger Frühjahrsmesse 1983. Bauelemente. radio fernsehen elektronik, Berlin 32 (1983) 6, S. 347-349
- [5] Schlegel, W. E.: Geschafft!? radio fernsehen elektronik, Berlin 32 (1983) 6, S. 343 und 344
- [6] Neues Bild vom Mikro-Markt. elektronik-zeitung, Lehfelden 20 (1982) 12, S. 1
- [7] Baker, S.: Super Z 80: Fünffmal schneller als Vorgänger. Elektronik, München 32 (1983) 13, S. 11 und 12
- [8] Pelka, H.: Der Einchip-Mikrocomputer in Feingerätebau und Meßtechnik (Teile 1 bis 7). Feinwerktechnik & Meßtechnik, München 87 (1979) 3 bis 88 (1980) 2
- [9] Franke, K.; Leichsenring, A.: Entwicklungsmodul für Einchip-Mikrorechner. radio fernsehen elektronik, Berlin 33 (1984) 1, S. 8-10
- [10] Roth, M.: Mikroprozessoren. Ilmenau: Wissenschaftliche Zeitschrift der TH Ilmenau 1982
- [11] Kieser, H.; Meder, M.: Mikroprozessortechnik. Berlin: VEB Verlag Technik 1982
- [12] Seifart, M.: Digitale Schaltungen und Schaltkreise. Berlin: VEB Verlag Technik 1982
- [13] Barthold, H.; Büberich, H.: Mikroprozessoren. Amateurreihe Electronica, Bd. 186 bis 188. Berlin: Militärverlag der DDR 1980

getragen werden, das schließt aber eine herkömmliche UV-Härtung aus. Das Absorptionsspektrum des Fotoinitiators im Klebstoff muß mit dem Emissionsspektrum des Strahles übereinstimmen, damit möglichst viel Energie zur Radikalbildung und damit zur Polymerisation verwendet werden kann. Erfolge erzielt man hier mit Klebstoffen, die durch einen getrennt aufgetragenen Aktivator härten. Unabhängig vom Auftragverfahren bleibt unter dem Chip im Rest unbestrahlter bzw. nicht ausgehärteter Klebstoff, der in einer nachfolgenden IR-Strecke nachgehärtet werden kann.

▼ **Unter Antisitedefekt** versteht man in der Halbleitertechnik einen für binäre Halbleiter fundamentalen Gitterstrukturdefekt, bei dem sich ein Anion auf dem Platz eines Kations befindet. So ist im GaAs-Kristall hier im Kristallzentrum Arsen anstelle von Gallium vorhanden. Zwischen den richtig an den Kubusecken platzierten Arsenatomen (AS_{As}) sitzt fälschlicherweise ein Arsenatom auf dem Platz eines Galliumatoms (AS_{Ga}). Man nimmt an, daß der Antisitedefekt die Ursache für die halbisolierenden Eigenschaften von undotiertem GaAs ist. Die Verteilung tiefer Störstellen in GaAs-Scheiben läßt sich mit Hilfe der IR-Absorption ermitteln.

▼ Nach der Dick- und Dünnschichttechnik haben sich auf dem Gebiet der Mikrowellentechnik nunmehr als ein weiterer großer Innovationsschub auch **monolithisch integrierte Mikrowellschaltungen** (MMIC) durchgesetzt, bei denen aktive, passive und Streifenleitungsstrukturen auf einem Halbleiterchip vereint werden. Dabei wird als Halbleitermaterial wegen seines höheren spezifischen Widerstandes und der größeren Elektronenbeweglichkeit vorzugsweise Galliumarsenid verwendet. Die Herstellung der aktiven Inseln auf dem halb-isolierenden GaAs-Substrat erfolgt mit Epitaxieverfahren (Flüssigphasen- oder Molekularstrahl epitaxie). Zur Herstellung der Halbleiterkomponenten müssen die Dotierprofile der aktiven Inseln definiert mit hochgenauer Fotolithografie oder Elektronenstrahlolithografie und mit der Ionenimplantation verändert werden. Die Leitungsstrukturen und passiven Elemente werden durch Aufdampfen oder Aufstäuben aufgebaut. Anwendungsgebiete sind neben der analogen Mikrowellentechnik auch digitale Schaltungen mit hoher Taktfrequenz und hoher Übertragungsrate (optische Nachrichtentechnik).

▼ **Die laserunterstützte Elektrolyse** arbeitet mit einem Laserstrahl, der auf eine im Elektrolyten befindliche Elektrodenoberfläche (Substrat) auftrifft, wobei die Strahlungsenergie im Bereich der Auftreffstelle zum Teil absorbiert wird, so daß sich eine lokale Erwärmung des Phasengrenzbereichs Elektrode-Elektrolyt ergibt. Diese lokale Erwärmung kann unterschiedliche Effekte bewirken, wie eine Verschiebung des Gleichgewichtspotentials Elektrode-Elektrolyt, eine Intensivierung des Stofftransports bei der Elektrolyse, eine höhere lokale Abscheidgeschwindigkeit, eine Induzierung der autokatalytischen Abscheidung bei der außenstromlosen Elektrolyse. Die Anwendung der laserunterstützten Elektrolyse zur selektiven Beschichtung dünner Grundwerkstoffe mit verschiedenen Metallen (Cu, Edelmetalle) wird technisch beherrscht, bei dickeren Substraten jedoch nur bedingt, da hier hohe Laserleistungsdichten mit starker Strahlfokussierung notwendig werden.

▼ **Die Mikromechanik**, die im Strukturbereich von 10...1 µm angesiedelt ist und zum Aufbau ihrer Funktionskomponenten Methoden und Verfahren der Mikroelektronik benutzt, erfordert zu ihrer Realisierung die dritte Dimension und damit Tiefenstrukturen. Solche Tiefenstrukturen lassen sich bei Silizium durch das anisotrope Ätzverhalten und den Einbau von hochdotierten p⁺-Schichten als Ätzstopp herstellen. Für die Tiefenstrukturierung anderer Materialien ist ein Lithografieverfahren für große Dicken von

Resistschichten erforderlich. Die Resistdicke bestimmt die Strukturtiefe sowohl bei der Abscheidung von Schichten im Plasma oder in der Galvanik als auch bei der substraktiven Ätztechnik.

▼ **Ein neues staatliches Längenmaß** wurde von Mitarbeitern des Leningrader Instituts für Metrologie entwickelt. Als Normal diente die Strahlung eines Helium-Neon-Lasers. Das Meter ist nun als die Länge eines Weges definiert, den der Laserstrahl im Vakuum im 299 792 458. Bruchteil einer Sekunde zurücklegt.

Ein neues sowjetisches Lichtnormal, das die Präzision von Lichtmeßgeräten nach Meinung der Experten wesentlich erhöhen wird, wurde ebenfalls entwickelt. Die erreichte Schwärzung des schwarzen Körpers und seine Wärmebeständigkeit erreichten, daß Fehler bei der Bestimmung der Strahlungsleistung auf ein Minimum begrenzt sind.

▼ **Für die Herstellung von 256-Kbit-RAMs** ist höchste Reinheit erforderlich: Je 27 l Luft sind max. zehn Staubpartikel eines Durchmessers von max. 100 nm zugelassen. Die für den Chip verwendeten Strukturbreiten betragen 1,5 µm.

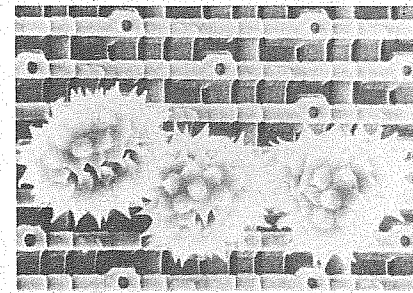


Foto: Siemens

Um diese hohen Forderungen an die Reinheit zu erfüllen, werden je Stunde $2 \cdot 10^6$ m³ Luft durch Filter und Waschanlagen gepumpt. Im Bild sind Gänseblümchenpollen (20 µm Ø) auf Leiterbahnen eines 256-Kbit-RAM dargestellt, der 10 000 Transistoren/mm² enthält.

▼ **Ein optisches Lotungssystem** dient zum Aufnehmen von dreidimensionalen Szenen im Fertigungsprozeß als Höhenrasterbild. Dabei werden die Bilddaten als digitale Information einem Fertigungsautomaten oder Roboter zugeführt. Aus dem Höhenrasterbild werden dabei die für die jeweilige Aufgabenstellung wichtigen Parameter ausgewählt und zur Steuerung verwendet. Das Lotungssystem besteht aus einem IR-Sender, der einen modulierten Lichtstrahl aussendet. Ein Detektor empfängt als Echo das vom beleuchteten Oberflächenelement diffus reflektierte Licht. Die Phasendifferenz zwischen Empfangs- und Modulationssignal ist ein Maß für den Laufweg des Lichtes und damit auch für die Höhe des reflektierenden Oberflächenelements gegenüber einer Referenzebene.

▼ **Der Prototyp eines mobilen Roboters** (Amooty), der verschiedene Kontroll- und Wartungsarbeiten in Gefahrenzonen übernehmen kann, ist von Toshiba und der Universität Tokio gemeinsam entwickelt worden.

Er bewegt sich automatisch mit einer Geschwindigkeit von 15 m/min, orientiert sich selbständig an seinem Einsatzort, erkennt unerwartete Hindernisse, umgeht sie oder wählt selbst eine andere Route zum vorgesehenen Ziel, wobei Sensoren Position und Richtung überwachen. Amooty setzt sich aus drei Grundeinheiten zusammen, die Füßen, Händen und Augen eines Menschen entsprechen. Ein integriertes Kontrollsystem steuert seine Bewegungen. Vier freibewegliche, einzeln gesteuerte Rollsysteme, jedes mit drei kleinen Rädern, bilden die „Füße“. Mit Hilfe von „Radarmen“ und den kleinen Rädern kann sich Amooty nicht nur vor- und rückwärts, sondern auch auf- und abwärts bewegen. So bewältigt er mühelos 22 cm hohe Treppenstufen. Neben sechs Grundbewegungen des Mani-

pulators sind zusätzliche Einstellungen möglich, so daß er insgesamt 84 verschiedene Bewegungen ausführen kann.

Ausgerüstet mit Greifern oder anderen Werkzeugen, ergibt sich für den Roboter ein breites Anwendungsgebiet. Visuelle Sensoren, kombiniert mit Fernsehkamera und einem Laserstrahl, ermöglichen ein Erkennen der Umgebung.

Das integrierte Kontrollsystem, das mit einem hochentwickelten Computer arbeitet, koordiniert die Arbeit des Rollsystems, des Manipulators und der visuellen Sensoren. Jedes einzelne System kann aber auch unabhängig von den anderen durch Mikrocomputer gesteuert werden.

▼ **Mit sRAM** (Static Random Access Memory) wird ein **RAM-Speicher** bezeichnet, der aus Transistoren mit hoher Elektronenbeweglichkeit (HEMT) besteht. Die Speicherkapazität beträgt 1 Kbit und die Zugriffszeit 0,9 ns bei 77 K. Auf einem Chip von 3,0 mm × 2,9 mm sind 7244 Elemente vereint, die insgesamt 1 024 Speicherzellen ergeben. Die Speicher verwenden die direktgekoppelte FET-Logik (DCFL). Dabei müssen die einzelnen Transistoren eine hohe Gleichheit der Schaltschwellenspannungen aufweisen. Diese Anforderung läßt sich mit einem präzisen Molekularstrahl-Kristallzielverfahren und durch selektive Trockenätztechnik erfüllen, die die genaue Streuung der GaAs- und AlGaAs-Materialien ermöglichen.

▼ **Berichtigung:** Im Beitrag „Basic-Heimcomputer“ im Heft 1 (1985) S. 13–18, sind bedauerlicherweise mehrere Bildunterschriften miteinander vertauscht. Die unter Bild 2 abgebildete Zeichnung zeigt die günstige Anordnung der Tasten, Bild 3 die Leiterseite der Platine, Bild 4 die Bestückungsseite der Platine und Bild 5 den Bestückungsplan. Wir bitten, diesen Fehler zu entschuldigen.

Es haben sich auch noch einige Fehler in den Stromlaufplan, die Leiterzuggbilder und in den Bestückungsplan eingeschlichen.

Stromlaufplan

- Pin 29 und Pin 11 von D₃ sind vertauscht
- Pin 26 (RESET) von D₃ muß über Leitung 11 mit Pin 17 von D₂₀ verbunden sein
- Pin 10 von D_{26.2} muß mit Pin 9 von D₄ statt mit Pin 11 von D₄ verbunden sein
- Pin 11 von D_{1.2} muß mit Pin 1 von D₂₂ statt mit Pin 1 von D₂₃ verbunden sein

Bestückungsseite

- Leiterzug von D₀ am Steckverbinder muß statt mit Pin 2 von D₅ mit Pin 3 von D₅ verbunden sein
- Leiterzug von Pin 8 von D₅ (L) nach Pin 15 von D₃ fehlt
- Leiterzug von Pin 13 von D₂₀ nach Pin 11 von D₃ fehlt (+5 V)
- fehlende Leiterzüge für VD₁ und VD₂

Bestückungsplan

- die IS zwischen D₂ und D₁₃ ist D₂₆

Berichtigungen zur Beitragsfolge „Multimikrorechnersysteme“ Hefte 4 bis 12 (1984):

1. Heft 5, S. 300, linke Spalte, 38. Zeile: statt „Speicherzeit“ lies „Speicherplatz“.
2. Heft 5, S. 300, Bild 10: Die Bezeichnungen ARDY, ASTB sind zu vertauschen (ASTB einschließlich des Negationspunktes im PIO-Schaltssymbol).
3. Heft 6, S. 367, rechte Spalte, 36. Zeile: statt „LSI-“ lies „SSI-“.
4. Heft 6, S. 370, Bild 26: statt „Adressenhalterregister“ lies „Datenhalterregister“.
5. Heft 7, S. 438: Die Bilder 32 und 33 sind miteinander zu tauschen.
6. Heft 11, S. 709, Tafeln 3 und 4, linke Spalte, 11. Zeile: Bei allen Anmerkungen zu BUS REQUEST lies statt „Befehl“ „Maschinenzyklus“; des weiteren in Tafel 4 bei „Dominanzebene 3“ „BUS REQUEST“ statt „REQUEST BUS“.

Erweiterungen für Basic-Heimcomputer

Dipl.-Math. ECKHARD SCHILLER

Seit dem Erscheinen des ersten Beitrages [1] über den Basic-Heimcomputer wurde er von vielen Amateuren nachgebaut. Im folgenden Beitrag wird gezeigt, wie die Einschränkungen des Grundgeräts (geringer Speicherplatz, fehlender Kassettenanschluß und unvollständiger Basicbefehlsvorrat) beseitigt werden können.

Anschluß eines Kassettengerätes

Um Programme und Daten archivieren zu können, werden bei Heimcomputern fast ausschließlich Kassettenbandgeräte benutzt. Sie sind preiswert und meistens vorhanden. Die Aufzeichnungsgeschwindigkeit liegt zwischen 100 Bd und 4 800 Bd, das sind 10...500 byte/s. Wenn die Geschwindigkeit 1 200 Bd überschreitet, steigen die Anforderungen an Bandmaterial und Kassettenrecorder stark an. Deshalb wurde der Kassettenanschluß des Heimcomputers mit diesem Wert realisiert. Als Aufzeichnungsverfahren wird die Richtungstaktschrift verwendet (Phase Encoding).

Datenaufbau

Die Daten werden in Blöcken zu maximal 256 byte ausgegeben. Die Blöcke entsprechen dem Intel-Hex-Format für Lochbänder. Vor dem ersten Block wird zur Einregelung des Verstärkers einige Sekunden lang eine 0101-Folge ausgegeben. Dann folgt das Synchronwort 0E6H. Anschließend beginnen die Daten mit der Blocklänge (1 bis 256), der Speicheradresse (2 byte) und dem Kontrollwort 00H. Danach erscheinen die eigentlichen Daten. Von allen Bytes wird die Summe gebildet und als letztes Byte des Blocks ausgegeben. Zwischen zwei Blöcken befindet sich eine kurze 0101-Folge. Der letzte Block hat als Kennzeichen die Speicheradresse 0. Bei der Eingabe wird das Synchronwort 0E6H gesucht. Da die Daten auch negiert erscheinen können, wird ebenfalls nach dem Komplement von 0E6H gesucht. Bei der Erkennung von 19H müssen alle Bits negiert werden. Wird das Kontrollwort nicht richtig erkannt, kommt eine Fehlerausdrift. Falls die Summe nicht mit der aufgezeichneten übereinstimmt, erscheint bei der Eingabe die Ausdrift SF.

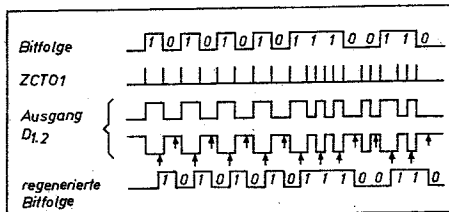


Bild 1: Kodierung der Daten bei SAVE/LOAD

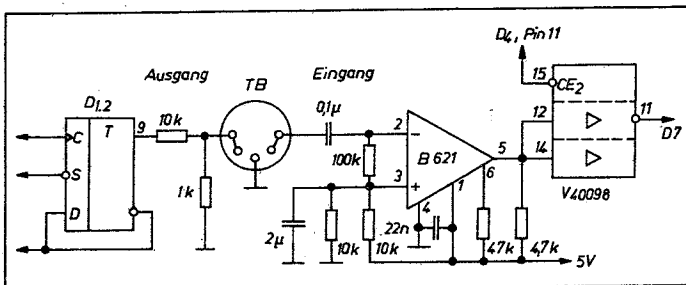


Bild 2: Tonbandanschluß für den Basic-Heimcomputer

Bild 4: Speichererweiterung mit der IS U 214 bzw. U 224 (1 bis 8 Kbyte)

Schaltung

Da auf der Grundleiterplatte des Basic-Heimcomputers keine PIO vorhanden ist, wird der Ausgang ZCT01 der CTC D₂₀ zur Datenausgabe genutzt. Die Umwandlung der Datenbytes in eine serielle Bitfolge wird durch die Software realisiert. Dabei werden immer zwei aufeinanderfolgende Bits verglichen. Wenn sie den gleichen Wert besitzen, gibt die CTC zwei Impulse ab, sonst nur einen. Das Flip-Flop D_{2.1} setzt die Impulse in Rechteckschwingungen um. Da die Stellung von D_{2.1} nicht bekannt war, gibt es dabei zwei Möglichkeiten (Bild 1).

Die Rechteckimpulse gelangen über einen Spannungsteiler an die Diodenbuchse. Die Signale vom Kassettenrecorder werden vom Operationsverstärker B 621 verstärkt und wieder in Rechteckform gewandelt (Bild 2). Als Eingangsschaltung kann auch die anderer Kleincomputer (AC 1, LC 80) genutzt werden. Über ein Gatter der IS V 40098 gelangen die Signale an den Datenbus (Leitung D7). Soll keine neue IS verwendet werden, kann auch D_{5.2} benutzt werden. Ihre Funktion übernimmt dann das freie Gatter von D₂. Leider wurde auf der Grundleiterplatte kein Kassetteneingang vorgesehen. Deshalb muß ein schon belegter Stift der Steckerleiste geändert werden (5 V, WAIT, NMI).

Speichererweiterungen

Auf der Grundleiterplatte stehen für die Abspeicherung von Programm und Daten nur etwa 500 byte zur Verfügung. Das ist für viele Anwendungen zu wenig. Deshalb werden im folgenden verschiedene Möglichkeiten zur Speichererweiterung vorgestellt.

Erweiterungen mit den IS U 202, U 215 und U 225

Diese Schaltkreise haben eine Organisation von 1K × 1 bit. Die Typen U 215 und U 225 sind nicht pinkompatibel zum U 202. Deshalb muß für sie die Leiterplatte geändert werden. Welche Adreßleitung der CPU an welchen Adreßeingang eines Speicherschaltkreises kommt, ist bei statischen RAMs völlig ohne Bedeutung. Wenn der Speicher erweitert werden soll, können auf die RAMs, die schon auf der Leiterplatte angeordnet sind, weitere acht RAMs im Huckepackverfahren aufgesetzt werden

(gleicher Typ). Alle Anschlüsse außer /CS werden verlötet. Die /CS-Anschlüsse werden miteinander verbunden und an eine zusätzliche IS D 110 angeschlossen (Bild 3), die auf die IS D₂₆ gesetzt wird. Damit stehen fast 1,5 Kbyte zur Verfügung.

Erweiterung mit der IS U 214

Die IS U 214 ist ein (1K × 4)-bit-Speicher. Sie ist wesentlich billiger als vier IS U 202 und benötigt weniger Strom. Auf der Grundleiterplatte kann sie nach einer geringen Änderung eingesetzt werden. Da sie keine getrennten Datenein- und Datenausgänge wie die U 202 besitzt, werden die Datenleitungen der beiden U 214 an die Eingänge des Registers D₁₈ angeschlossen. Die Eingänge und die zugehörigen Ausgänge des D₁₈ müssen dann jeweils durch einen 1-kΩ-Widerstand verbunden werden. Über diesen gelangen die Daten beim Schreiben vom Bus an die Datenleitungen der U 214. Dabei ist D₁₈ hochohmig. Beim Lesen und Bildschirmzugriff verhindert der relativ niederohmige Ausgang der U 214 eine Störung durch den Widerstand. Diese Funktion wird allerdings durch den Hersteller nicht garantiert! Probleme dürfte es aber höchstens mit der Anzeige geben, falls der Zeichengenerator D₂₆ den H-Pegel nicht mehr richtig erkennt.

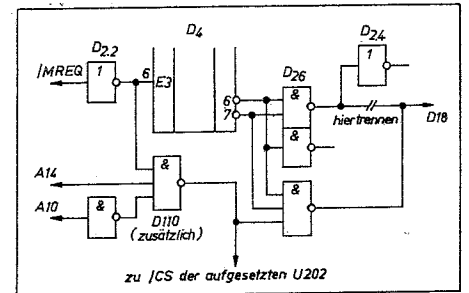


Bild 3: Speichererweiterung mit der IS U 202 nach der Huckepackmethode, Änderung an der Grundleiterplatte

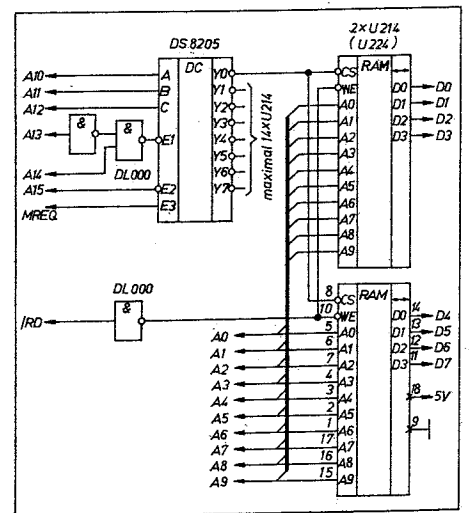


Bild 4 zeigt, wie mit Hilfe von IS U 214 als Zusatzspeicher die Kapazität auf insgesamt 9 Kbyte erhöht werden kann. Als Schreibsignal wird nicht \overline{WR} , sondern RD verwendet. Damit können Unklarheiten zu Beginn eines Schreibzyklus vermieden werden. Anstelle der U 214 kann auch der CMOS-Typ U 224 verwendet werden.

Erweiterung mit der IS U 256

Die IS U 256 ist ein dynamischer (16 K×1)-bit-RAM. Sie benötigt einen periodischen Refreshzyklus, der über die CPU ausgeführt wird. Deshalb darf der Reset nur einige hundert Mikrosekunden lang zur Vermeidung eines Datenverlustes betätigt werden. Aus diesem Grund wurde im Interpreter die Möglichkeit geschaffen, ein Basicprogramm außer durch den Reset auch durch NMI abzubrechen. Wie beim Reset wird bei NMI ein Sprung nach 0 ausgeführt. Eine sehr einfache Schaltung für die U 256 zeigt Bild 5.

Das Signal MREQ = H löst einen Speicherzyklus aus. \overline{RAS} wird eingeschaltet, und nach einer Verzögerungszeit von mindestens 25 ns schalten die Multiplexer DL 257 auf die CAS-Adresse um. Bei $A15 = L$ und $A14 = H$ wird nach weiteren 10 ns \overline{CAS} eingeschaltet (Bild 6). Alle anderen Speicherzugriffe dienen zum Auffrischen des Speicherinhalts. Bei einem Schreibzyklus sind $\overline{RD} = \overline{RFSH} = H$ und $\overline{WE} = L$. Damit bleiben die Datenausgänge gesperrt, so daß sie mit den Eingängen verbunden werden können. Die Stromversorgungsanschlüsse der U 256 müssen durch Keramikkondensatoren abgestützt werden, besonders die +12 V. Wenn es bei der Inbetriebnahme Probleme geben sollte, können mit Hilfe von 100-pF-Kondensatoren die Verzögerungszeiten verlängert werden.

Hinweise zum Aufbau

Da sich der Basic-Heimcomputer nur als Ganzes in Betrieb nehmen läßt, werden noch einmal einige wichtige Fehlerbilder erläutert:

- Keine Impulse an D_{22}/D_{23} : Fehler in der CPU, dem EPROM, dem CTC oder in ihren Verbindungen
- Impulse an D_{22}/D_{23} , aber kein Bild vorhanden: Fehler im RAM oder in D_{18} . Zum Testen kann eine Adrebleitung zum

Tafel 1: Hexadezimalausdruck des EPROM D₆ (Basicinterpreter, Bereich 0 bis 3FFF)

0000	01 F8 03 21 1F 00 ED B3 01 F9 02 ED B3 01 FA 02
0010	ED B3 01 FB 02 ED B3 34 00 3C ED 5E C3 B9 01 05
0020	0A 2B FF 40 7F 05 7F 32 13 49 43 06 D1 13 59 FF
0030	D1 D9 08 FB ED 4D F3 C9 10 F2 47 0E 0E 1E 5F FF
0040	FF FF FF 08 DB FA FE 04 20 E8 32 10 A 02 07 47
0050	0030 00 21 80 38 E9 ED 4B 06 3C 21 80 3C DD 8C 3C
0060	06 10 36 20 18 01 C7 23 10 F8 36 FF 23 00 20 F8
0070	36 F7 C9 E5 2A 08 3C FE 1E 28 42 E6 7F 27 23 7E
0080	FE F7 20 D0 D5 C5 2A 24 3C 11 1E 00 97 ED 52 22
0090	24 3C ED 4B 05 3C 21 00 00 05 19 10 FD 44 4D 0B
00A0	2A 9F C1 11 00 3C ED B6 06 1D 2B 36 20 14 FB CA
00B0	D1 CB 7E 28 02 23 18 C7 22 08 C1 E1 C9 CB 7E 23
00C0	2C FB 18 2B 05 05 06 0A CD E0 00 20 F9 10 F9 CD
00D0	E0 08 28 FB 4F CD E0 00 B9 28 F4 EF 7F 91 E1 C9
00E0	03 21 FE 13 0E 00 7E BE 20 FC 06 00 CB 5F 20 16
00F0	04 87 C9 2A 08 3C FB CB 15 CB 14 7C FE 23 28 2E E6
0100	03 06 10 F8 18 E0 3A FF 11 CB 47 28 02 CB D8 78
0110	21 30 A1 85 6F 7E 81 FE 0A 38 12 FE 5E 28 0E FE
0120	05 9E 31 00 3C ED B6 06 1D 2B 36 20 14 FB CA
0130	30 41 4E 55 00 26 3A 00 0A 08 3C 22 24 3C 36 5F
0140	CD C4 00 2A 05 3C FE 5C 20 11 2B CB 7E 28 01 2B
0150	36 20 2B 07 7E 28 01 2B 22 06 3C FE 7F 20 DF FD
0160	2A 24 3C 00 24 08 7F FB FB 3A D2 69 0D 5F CD 4E
0170	06 CD 24 08 3A 06 3C 87 3D BD DA 69 0D 26 00
0180	CB 30 44 40 29 09 29 09 29 09 29 09 29 09 29 09
0190	05 9E 31 00 3C ED B6 06 1D 2B 36 20 14 FB CA
01A0	20 CB 43 C8 CB 28 C9 42 43 2D 05 45 20
01B0	33 2E 31 3A 1E 0F 27 CE 1E 2A 00 3C 2B 7E FE 1E
01C0	28 58 0E 08 CD 5A 09 FB 21 87 01 06 06 7E EF 23
01D0	10 FB 21 00 00 22 0C 3D 08 01 CD F7 0E 22 06
01E0	3C CD 56 00 23 54 5D 23 22 06 3C 01 04 00 ED 43
01F0	02 3C 83 21 04 01 ED B0 2A 00 CD ED 58 02 3C 19
0200	3E FF 77 BE 20 06 97 77 BE 23 28 FA 22 04 3C CD
0210	02 3C 83 21 04 01 ED B0 2A 00 CD ED 58 02 3C 19
0220	00 3E 3E EF CD 38 24 FD 7E 06 FE 3A 38 15 CD 54
0230	00 FD 36 00 1E FD 36 03 CE 1E EF FD 21 2B 3C
0240	C3 E2 03 FE 30 DA 65 0D 2F 32 00 8C CD 54 02 CD
0250	02 E2 18 BE CD F7 0C 22 29 3C FD E5 D1 FD 21 2B
0260	3C 01 03 00 1A 87 30 01 13 1A FE 7F C8 FE 3F 20
0270	02 3E 9F FE 27 20 18 FD 77 00 13 FD 23 03 CB 74
0280	C2 65 0D 1R 87 30 01 13 5F FE 27 26 FA 18 34 C3
0290	E2 12 0F 3E 8C 06 1A FD ED 41 C3 CA ED D8 E1 28
02A0	28 CB 7E 23 28 EF CB 7E 28 0D 23 DD 7E 00
02B0	CB 7F 20 F7 ED 1A 20 DE 18 ED 0E D5 D1 7E 87 18
02C0	01 1A C1 FD 77 00 13 FE 20 CA 64 02 0C FD 23 C3
02D0	64 02 FD 36 00 1E ED 43 0A 3C ED 48 02 3C 2A 00
02E0	3C 18 09 3E 1E 23 ED 84 E2 14 03 83 5E 23 56
02F0	2R 2D 3C 97 ED 52 1E 30 3B 28 0E 20 09 5A 0D
0300	E2 12 0F 3E 8C 06 1A FD ED 41 C3 CA ED D8 E1 28
0310	C1 ED B8 23 54 50 21 29 3C ED 4B 08 3C ED B0 2A
0320	02 3C ED 4B 0A 0C 09 22 02 3C 11 29 3C 19 20 E5
0330	ED 58 02 3C 18 23 06 7E FE 1E 08 1E 23 28 F8 ED
0340	53 02 3C D1 ED B0 2A 3C FE 1E C2 DA ED 0C C9 1A
0350	0F 13 1R 67 13 CD 0C 06 1A 87 FA 03 0A FE 40 FA
0360	64 83 F6 28 EF 1A FE 1E 13 20 ED C9 21 12 0F 01
0370	01 ED 5A 06 2B 28 04 CB 7E 28 FA 23 7E EF
0380	10 FB 3E 28 18 DE ED 58 0C 3C 06 13 1A FE 27
0390	18 28 13 C5 CD 4F 03 C1 16 F2 CD C4 00 2A 08 3C
03A0	28 2D 08 3C 1E 04 CD 4F 03 ED 58 04 3C 21 00 00
03B0	23 1A 1R 87 28 FA CD 5C 06 3E 42 EF 03 12 02 CD
03C0	56 00 ED 58 0C 3A 2A 02 3C 19 36 00 EB 2A 04 3C
03D0	37 ED 52 44 4D 02 0E 13 ED B0 FD 2A 00 3C FD 23
03E0	FD 23 FD 7E 0E FE 80 FA 0C 84 FE CE FR 64 0D FD
03F0	23 21 12 0F 67 FE 23 66 6F E3 CD 24 08 2A 00 3C

Tafel 2: Hexadezimalausdruck des EPROM D₇ (Bereich 400H bis 7FFF)

0400	ED 48 02 3C 3A 14 3C ED 41 20 0D 3A 15 3C ED 41
0410	20 06 E5 FD E1 C3 E8 3E 1E ED B4 FA 04 04 3E
0420	CD 05 06 28 00 CD C7 06 28 1F CD 11 08 CD 4E 06
0430	7C 85 CD E1 03 06 3C 18 02 06 01 FD 7E 00 FE 3A
0440	FD 23 CA ED 03 FE 1E CA ED 03 10 EF C3 65 0D CD
0450	24 08 FD E5 18 87 FD E1 18 DF CD 06 06 28 DA CD
0460	C7 06 28 F6 CD 22 07 DD E5 FD E5 CD 38 01 7A FE
0470	24 20 05 CD 4B 09 18 0F 21 00 00 0E FF FD 7E 00
0480	FE 7F 28 03 CD 2F 08 FD E1 DD E1 CD F6 06 19 CH
0490	CD 05 06 28 00 CD C7 06 28 1F CD 11 08 CD 4E 06
04A0	0E 3E 1E EF 18 9D CD 58 00 18 CD CD 8C BA FD
04B0	7E FF FE 2B CD C9 04 CD 4E 06 18 FD CD 22 07 C9
04C0	3D FD BE 0C C7 05 FD FD 23 CD 2F 06 CD F6 06 3C
04D0	2D 27 3E 2E FD BE 0C C7 05 FD FD 23 CD 24 08
04E0	DD CB 0E FE 54 5D FD 7E 0E FE 2C 28 FD 23 2B E8
04F0	CD 24 88 EB 42 48 CD CE 07 DD CB 01 FE CD 93 09
0500	DD 7E 01 E6 7F FE 24 28 02 29 2D 73 04 DD 72
0510	05 DD 73 92 DD 14 93 0E DD 19 CD 11 08 CD 4E 06
0520	18 ED CD 64 81 7E FE 20 39 01 97 80 77 C3 39 04
0530	CD 64 91 7E 2F 80 27 77 C3 39 04 CD 8C BA DD E5
0540	FD E5 C3 75 04 FD 22 0A 3C FD E1 DD E1 DD E5 FD
0550	E3 FD 0E FE BC C2 65 0D FD 23 CD 2F 06 CD C1
0560	09 CD 94 0A 20 09 D1 D1 FD 2A 08 C3 C3 39 04 C5
0570	E5 FD 7E 0E BF C2 93 05 FD 23 CD 2F 06 CD 7E
0580	0E E1 C1 CD 79 08 DD E5 CD C3 08 DD E1 CD F6 06
0590	0E 3E 1E EF 18 9D CD 58 00 18 CD CD 8C BA FD
05A0	E5 CD 2A 08 4D E1 71 23 CD 4E 06 18 F3 CD 2A 08
05B0	4D 4E 06 CD 2A 08 ED 69 18 F6 2A 00 3C ED 1B
05C0	02 3C FE FE ED 81 3E 44 C2 75 0D FD 23 FD 18 1B
05D0	18 1A FE 1E 20 EC ED 53 26 3C C3 39 04 CD 22 07
05E0	FD E5 FD 2A 26 3C 3E 2D FD BE 0C FD 23 28 06 FD
05F0	23 FD 23 FD 23 CD 2F 08 FD 22 26 3C FD E1 CD F6
0600	06 CD 4E 06 18 FE 7E 00 FE 3B 28 2D FE CD
0610	C2 78 06 FD 7E 06 FD 7E 06 FD 28 0E CD 5F 60 C5
0620	06 20 08 3C 23 04 CB 7E 28 FA 76 06 27 2B 03
0630	F2 0C 06 C6 07 47 2E 0E 10 FB FD 23 FD 7E 00
0640	FE 3A 28 11 FE 2C 28 08 FE 1E CD C3 55 06 FD 7E
0650	00 FE 2C 28 04 D1 C3 39 04 FD 23 C9 05 0E FF CD
0660	7E 08 CD 67 06 D1 C9 79 FE 2E 08 7E 87 C8 FE
0670	23 10 F9 32 11 2C CD 6B 0C 21 16 3C CD 7E 28
0680	03 3E 2D EF 7E E5 07 DD 7E 06 47 0E 2A 12 3A ED
0690	6F CD AB 06 CD 87 06 CD 87 06 CD 87 06 CD 87 06
06A0	73 00 3E 20 C3 73 00 23 CD 8B 06 04 20 8A 3E 2E
06B0	EF 0D 97 ED 6F 87 28 01 8D F6 39 FE 3A 38 02 C6
06C0	07 CB 79 C4 73 00 C9 7E 00 FE 27 20 18 FD 23
06D0	FD 7E 0E 1E C8 FD 23 FE 2F C8 EF 18 F2 FE 25
06E0	CD FD 23 CD 2A 08 21 14 C3 01 00 00 CD 87 06 28
06F0	2D CD 96 0E 3C 79 FE 24 20 1A 79 FE FE 3E 24
0700	C2 78 06 FD 7E 06 FD 7E 06 FD 28 0E CD 5F 60 C5
0710	DD 36 06 00 C9 DD 71 02 DD 70 03 DD 75 84 DD 74
0720	05 0D 2A 00 2A 00 ED 48 02 3C DD 0D 09 FD 7E 00 FE
0730	4B 47 3E 40 FA 75 0D 0E 00 51 FD 23 FD 7E 00 FE
0740	FA 89 07 FE 2A 28 0E FE 30 FA 59 07 FE 40 F2 C6
0750	07 FE 3A F2 59 07 4F FD 23 78 DD 96 00 E6 7F CD
0760	E9 07 DD 7E 01 E6 7F BF C2 E9 07 DD 5E 04 DD CB
0770	0E 7E 0C 37 3E 28 FD BE 08 09 23 CD 24 08 2B
0780	DD 46 05 4B CD 01 7E 28 14 CD CE 07 FD 7E 00
0790	FE 2C 20 0A FD 23 CD 24 88 2A 08 DA 69 0D 09 3E 24
07A0	8A 28 29 29 01 04 00 09 18 0F FD 7E 00 FE 2C
07B0	20 08 FD 23 E5 CD 2A 08 5D E1 CD 93 09 44 DD
07C0	6E 0D DD 66 03 97 ED 42 FA 69 0D 09 C9 D5 EB
07D0	21 00 00 3E 11 CB 18 CD 19 3D 28 09 38 01 1C CB
07E0	1C CB 1D 18 FD 61 69 C9 3D 28 09 38 01 1C CB
07F0	23 DD 23 DD 23 DD 23 DD 23 DD 23 DD 23 DD 23 DD

RAM aufgetrennt und ein Negator dazwischen gesetzt werden

- Wenn Bild und Tastatur in Ordnung sind, ist die Schaltung fehlerfrei. Sollten dann an bestimmten Stellen das Bild verschwinden und der Rechner nur noch auf Reset reagieren, kann der Fehler im Basicprogramm liegen. Nicht alle falschen Befehle werden vom Interpreter erkannt. Aber auch die RAMs und EPROMs können Fehler hervorrufen.

Und noch ein Hinweis: In der Version 2.4 nach [1] ist ein Programmfehler bei OUT. Die Zellen ab 0404H müssen geändert werden in CD 10 04 C5.

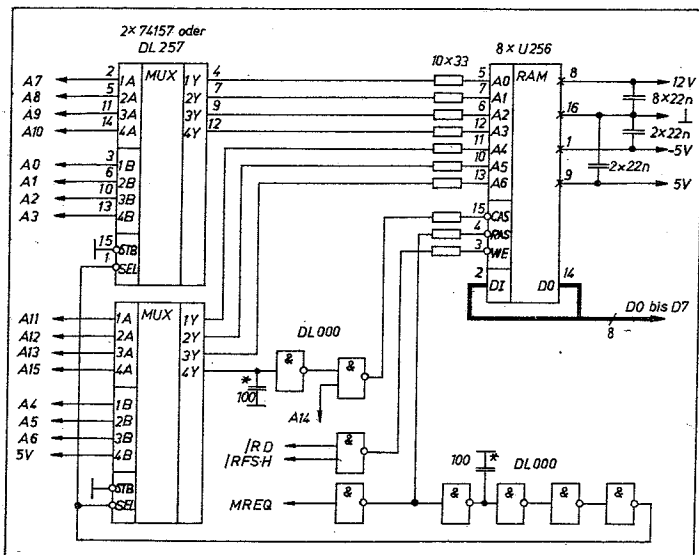


Bild 5: Speichererweiterung mit der IS U 256 (16 Kbyte)

Neuer Basicinterpreter (Version 3.1)

Der Interpreter nach [1] wurde auf volle 4 Kbyte erweitert. Damit sind nun alle vier EPROMs programmiert (Tafeln 1 bis 4). Durch eine platzsparende Programmierung gelang es, fast alle Basicbefehle zu realisieren. Nur auf mathematische Funktionen mußte aus Platzgründen verzichtet werden. Die Beschreibung kann natürlich kein Basiclehrbuch ersetzen; Im wesentlichen wird nur auf die Unterschiede zu anderen Basicversionen von Büro- oder Kleincomputern eingegangen. Für die Grafikbefehle muß der Zeichengenerator aus [1] um die Zeichen 0 bis 15 erweitert werden (Tafel 5). Zur Kontrolle der EPROM-Listen wurde von jeweils 256 Zellen die Summe gebildet (Tafel 6). Damit kann jederzeit eine EPROM-Kontrolle durchgeführt werden. Besonders die IS 555 neigen dazu, nach einigen Tagen oder auch erst nach Monaten einzelne Bits zu „vergessen“, so daß plötzlich unerklärliche Fehler auftreten können.

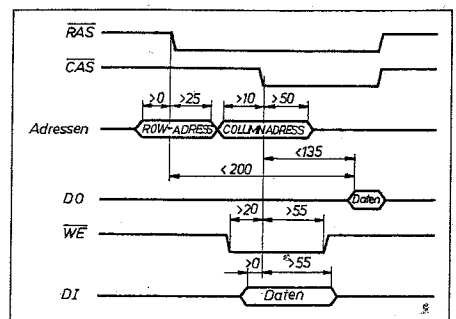


Bild 6: Die wichtigsten Zeiten in ns für die U 256 (S 256) im Lese- bzw. frühen Schreibzyklus

Tafel 3: Hexadezimalausdruck des EPROM D₃ (Bereich 800H bis 0BFFF)

Table with 48 columns and 100 rows of hexadecimal data representing EPROM D3.

Tafel 4: Hexadezimalausdruck des EPROM D₄ (Bereich 0C00H bis 0FFFF)

Table with 48 columns and 100 rows of hexadecimal data representing EPROM D4.

Tafel 5: Ergänzung des Zeichengenerators D₂₁ (Bereich 0 bis 7FH)

Table with 48 columns and 10 rows of hexadecimal data representing the character generator D21.

Tafel 6: Bildschirm Ausgabe der Kontrollsummen und ihres Berechnungsprogramms. Bei 40 Zeichen je Zeile sind die ersten beiden Summen 6CCBH und 529DH

```
6CED 5267 67C9 618F
751E 771D 710A 6EB4
77EC 73F3 6690 5BB3
5E09 5AB3 6682 664A
>LIST
10 j=0
20 s=0;FOR i=1TO 256
30 s=s+PEEK(j):j=j+1
40 NEXT i:PRINT %0,s
50 IF j<1000GOTO 20
9999 END
```

Tafel 7: Programmänderungen für 40 Zeichen je Zeile

Table with 2 columns: Adresse and neuer Inhalt (alles hexadezimal). Lists memory addresses and their corresponding new hex values.

Bildschirmformate

Eine Bildschirmzeile enthält 29 sichtbare und ein unsichtbares Zeichen. Durch die Erhöhung der Taktfrequenz von 5 MHz auf 7 MHz kann eine Zeilenlänge von 40 + 1 Zeichen erreicht werden...

Kommandos

Wenn das Netz eingeschaltet wird, erscheint nach dem Reset die Ausschrift: BASIC-SE 3.1:

Nun muß die gewünschte Anzahl der Zeilen eingegeben werden (4 bis 29). Eine hohe Zeilenzahl bedeutet einen großen Speicherverbrauch...

LOAD

Tonbandeingabe: LOAD eingeben; Gerät auf Wiedergabe schalten; wenn ein Hupton zu hören ist, Taste ENTER drücken...

SAVE

Tonbandausgabe: SAVE eingeben; Ge-

rät auf Aufnahme schalten; Taste ENTER drücken.

SAVED

Außer dem Programm wird auch der Speicher bis RAMTOP ausgegeben. Damit bleibt der Stand der Variablen erhalten...

LIST

Das gespeicherte Programm wird angezeigt. Nach jeweils zwölf Zeilen muß eine Taste gedrückt werden, damit die Ausgabe fortgesetzt wird...

RUN

Der Bildschirm, alle Variablen und der Speicher bis RAMTOP werden gelöscht. Dann wird das Programm gestartet...

NEW

Besitzt dieselbe Wirkung wie das Einschalten des Netzes.

Auch alle anderen Anweisungen können als Kommando verwendet werden, wenn sie ohne Zeilennummer eingegeben werden.

Programmeingabe

Jede Programmzeile beginnt mit einer Zeilennummer (< 9999). Zweckmäßig sind Zehnerschritte. In einer Zeile können mehrere Anweisungen stehen...

Doppelpunkt getrennt werden müssen (insgesamt maximal 60 Zeichen). Wenn eine Zeile gestrichen werden soll, müssen die Zeilennummer und ENTER eingegeben werden...

Variablen

Zahlenvariablen können mit zwei Buchstaben oder einem Buchstaben und einer Zahl bezeichnet werden, Zeichenkettenvariablen mit einem Buchstaben und dem Zeichen (24H). Um die Laufzeit zu verringern, werden die Zahlen intern in Integer- und Realwerte geteilt...

Beispiel: 10H = 16, 0FFF = 255

Einfache Zahlenvariablen müssen nicht vereinbart werden. Ein- und zweidimensionale Zahlenfelder, einfache Zeichenkettenvariablen und eindimensionale Zeichenkettenfelder müssen vor der Benutzung mit einer DIM-Anweisung definiert werden.

Beispiel: 10 DIM A(10), B(5,15)
20 DIM C(10), D (10,3)

Dabei ist C() eine Variable mit maximal zehn Zeichen, D () besteht aus zehn Elementen zu je maximal drei Zeichen.

Intern werden Zahlen in 6 byte speicher- aufwärts hinter dem Programm gespeichert. Die ersten zwei Bytes enthalten den Namen. Das dritte Byte enthält OFFH bei Integerwerten oder im linken Halbbyte Vorzeichen und Komma und rechts die höchste Ziffer. Das vierte Byte beinhaltet die zweite und dritte Ziffer. Bei Integer- werten wird es nicht genutzt. Im fünften und sechsten Byte stehen die letzten vier Ziffern (dezimal) bzw. bei Integerzahlen der Wert in der Form LH (binär).

Beispiel: 10 = 0FF00A00H
1000 = 0FF00E803H
100000 = 000100000H
-123.4567 = 0C1234567H

Ausdrücke

Ausdrücke bestehen aus Konstanten, Variablen oder Funktionen, die durch Operatoren verknüpft werden.

Funktionen

Außer bei INT müssen die Argumente x immer Integerzahlen sein. Gegebenenfalls müssen sie vorher durch INT behandelt werden. Wenn das Argument keine Integerzahl ist, erscheint IF. Falls die Funktion mit \square endet, entsteht als Ergebnis eine Zeichenkette.

- IN(x)
Es wird ein U-880-IN-Befehl mit der Adresse x (< 256) ausgeführt. Das Ergebnis ist eine ganze Zahl zwischen 0 und 255.
- PEEK(x)
x wird als Speicheradresse gewertet (0 bis 0FFFFH). Das Ergebnis ist der Inhalt dieser Speicherzelle.
- RND
Es wird eine Zufallszahl zwischen 0 und 0.9999 erzeugt. Sie entsteht aus der letzten Zufallszahl und dem U-880-R-Register.
- CHR(x)
Der L-Teil von x wird in ein ASCII-Zeichen umgewandelt.
- INKEY
Die Tastatur wird abgefragt. Wenn eine Taste gedrückt ist, wird das Zeichen, sonst eine leere Zeichenkette geliefert.
- USR(x)
Es wird ein U-880-Unterprogramm mit der Adresse x gestartet. Der Inhalt der Register CBLH wird als Ergebnis übernommen.
- INT(x)
x wird in die nächstkleinere ganze Zahl umgewandelt. Wenn der Betrag < 32628 ist, ergibt sich eine Integerzahl.
- LEN(a)
Die Länge der Zeichenkettenvariablen a wird als Integerzahl geliefert.

Operatoren

Ausdrücke werden von links nach rechts unter Berücksichtigung der Vorrangregeln berechnet. Die Operationen haben folgende Rangfolge:

* /, -, +, OR, AND, >, <, #, =

Durch Klammerung kann die Reihenfolge geändert werden.

>, <, =, # (ungleich) liefern die logischen Werte 0 (falsch) und 1 (richtig). AND und OR wirken bitweise auf Integerzahlen oder logische Werte.

Zeichenkettenverarbeitung

Zeichenkettenvariablen und Konstanten können verglichen (<, >, =, #) und addiert werden (nur bei LET).

Auch eine Verarbeitung von Teilen von Zeichenkettenvariablen ist möglich. Dazu muß das erste Byte und die Länge in Klammern angegeben werden.

Beispiel: -10 DIM A(10), B(5,10)
20 A="ABCDEFGHIJKL", B(2)
=A(5,5)
30 PRINT A, A(3,4)
40 PRINT B(2)
, B(2, LEN(B(2)) - 2, 3)
Ergebnis: ABCDEFGHIJ CDEF
EFGHI GHI

Anweisungen

- DIM x(a), y(b,c), z(d), u(e,f), ...
Mit DIM wird der Speicherplatz für die Felder reserviert. Bei Speicherüberlauf kommt der Fehler MF.
- LET x=a, y(b)="text"
Den Variablen wird ein Wert zugewiesen. Das LET kann weggelassen werden.
- REM text
Kommentare, die Schlüsselworte enthalten, müssen in Hochkommata eingeschlossen werden.
- DATA m,n,l, ...
Mit DATA kann eine Liste von Zahlen, Variablen oder Zeichenketten in einer oder mehreren zusammenhängenden Zeilen vereinbart werden. Die DATA-Anweisungen sollten am Ende des Programms stehen, um Laufzeit zu sparen.
- READ a,b, ...
READ weist den Variablen, die nach READ stehen, die Werte der DATA-Anweisungen zu. Abweichend von anderen Basicdialekten muß vor der ersten READ-Anweisung immer ein RESTORE stehen.
- RESTORE
Mit RESTORE wird ein Zeiger auf die erste DATA-Anweisung gesetzt.
- PRINT a+b, "text"; x
PRINT realisiert die Bildschirmausgabe. Die Ausdrücke (Variablen, Konstanten) werden berechnet und ausgegeben. Auch "text" wird ausgegeben. Für PRINT kann auch ein Fragezeichen geschrieben werden. Falls vor einem Ausdruck ein % steht, wird er hexadezimal ausgegeben.
Trennzeichen:
, \triangle Vorrücken zur nächsten Spalte
; \triangle Ausgabe auf der nächsten Position
Wenn am Ende der PRINT-Anweisung kein Trennzeichen steht, wird eine Zeilenschaltung ausgeführt.
- INPUT "text"; x, ...
Mit INPUT können Variablen während des Programmablaufs Werte über die Tastatur zugewiesen werden. Die Zeichen , und ; und "text" wirken wie bei PRINT.
- GOTO n
Damit wird ein Sprung nach n ausgeführt, n ist eine Zeilennummer und kann auch als Variable angegeben werden.
- IF a THEN Anweisung
Wenn der Ausdruck a ungleich 0 ist, wird die Anweisung hinter THEN ausgeführt, sonst nicht. THEN kann entfallen. Für IF a THEN zeilennummer muß geschrieben werden:
IF a GOTO zeilennummer.

- FOR x=m TO n STEP l ... NEXT
Die Zählvariable x wird jeweils um den STEP-Wert l (bzw. 1, wenn STEP fehlt) erhöht. Bei NEXT wird abgefragt, ob die Zählvariable gleich dem Endwert n ist. In diesem Fall wird das Programm hinter NEXT fortgesetzt. Sonst wird die Schleife wiederholt. Sie darf nicht mit GOTO verlassen werden.
- GOSUB ... RETURN
Mit GOSUB wird in ein Unterprogramm verzweigt. Dieses Unterprogramm darf nur mit RETURN verlassen werden, nicht mit GOTO.
- END
Zeile 9999 END wird automatisch generiert. Aber auch in jeder anderen Zeile kann das Programm mit END verlassen werden.
- CLS
Der Bildschirm wird gelöscht und der Cursor auf die oberste Zeile gesetzt.
- OUT a,x,y,z, ...
a wird als Adresse eines U-880-OUT-Befehls gewertet, zu der die folgenden Werte (jeweils der L-Teil) ausgegeben werden.
- POKE a,x,y, ...
Der L-Teil von x wird auf die Adresse a abgespeichert, y auf a+1 usw.
- PLOT x,y
Mit PLOT kann eine Pseudografik erzeugt werden. Dazu wird eine Zeichenstelle in vier Rasterpunkte unterteilt. Der Rasterpunkt mit den Koordinaten x (0 bis 57, Spaltennummer) und y (0 bis Zeilenzahl * 2 - 1) wird mit PLOT auf schwarz gesetzt. Gleichzeitig wird der Cursor auf diese Zeichenstelle eingestellt, so daß PLOT 2*x, 2*y auch an Stelle von PRINT AT x,y genutzt werden kann.
- UNPLOT x,y
Der mit PLOT gesetzte Punkt wird wieder gelöscht.

Fehler

Wenn ein Fehler erkannt wird, erscheinen ein Kennzeichen und die betreffende Zeile (Tafel 8).

Beispielprogramme

Wurzel

Da die Quadratwurzel nicht im Interpreter enthalten ist, muß sie durch ein Unterprogramm bestimmt werden.

```
10 REM 'WURZEL'
20 INPUT 'ZAHL', Z
30 GOSUB 100
40 PRINT
50 PRINT 'WURZEL'; X, X*X
60 END
100 X=1:FOR I=1 TO 13
110 X=(Z/X+X)/2
120 NEXT:RETURN
```

Tafel 8: Kennzeichen und Erläuterung der möglichen Fehler

(@)	weder Zahl noch Schlüsselwort
A	Arithmetikfehler
B	Syntaxfehler
D	DATA fehlt
F	Funktion und Anweisung verwechselt
I	keine Integerzahl
M	Speicher voll
U	Überlauf bei PLOT oder Feldern
Z	Zeile nicht vorhanden
\square	keine Zeichenkette
= ()	dieses Zeichen fehlt

Tafel 9: Basicprogramm für das Bildschirmspiel Labyrinth

```

10 REM 'LABYRINTH'
20 GOTO 260
30 FOR I=1 TO 3:POKE P(I),C(I)
40 S=P(I)+R(I)
50 IF PEEK(S)<20H GOTO 220
60 P(I)=S:S=INT(2*RND)
70 IF S=0 THEN S=-1
80 IF (R(I)=1)OR(R(I)=-1)THEN S=S*X30
90 D(I)=R(I):R(I)=S
100 S=PEEK(P(I))
110 IF S=58H THEN S=2EH
120 C(I)=S:POKE P(I),58H
130 IF S=43H END:NEXT
140 IF INKEY="" THEN Z=M-30
150 IF INKEY="F" THEN Z=M-1
160 IF INKEY="G" THEN Z=M+1
170 IF INKEY="V" THEN Z=M+30
180 IF PEEK(Z)<20H THEN Z=M
190 IF PEEK(Z)=2EH THEN B=B+1
200 POKE M,20H:M=Z:POKE M,43H
210 PLOT 10,35:'B':'PUNKTE':GOTO 30
220 S=P(I)+D(I)
230 IF PEEK(S)<20H GOTO 250
240 P(I)=S:GOTO 100
250 S=P(I)-R(I):GOTO 60
260 FOR I=1 TO 16:FOR J=1 TO 27
270 ?':':NEXT?:NEXT
280 DIM P(3),R(3),C(3),D(3)
290 FOR S=0 TO 24 STEP 8
300 RESTORE:FOR J=1 TO 5
310 READ A,B,Y:GOSUB 450
320 NEXT:FOR J=1 TO 3
330 READ A,B,X:GOSUB 470
340 NEXT:NEXT
350 READ A,B,Y:GOSUB 450
360 READ Y:GOSUB 450
370 FOR J=1 TO 3:P(I)=3CA0H+I
380 C(I)=2EH:R(I)=1:NEXT
390 M=3CA7H:Z=M:B=-1:GOTO 30
410 DATA 5,8,5,21,27,5,0,2,9
420 DATA 7,12,9,18,27,9,5,9,13
430 DATA 1,5,17,0,8,0
440 DATA 0,27,9,-24
450 FOR I=A TO B:PLOT I,Y+S
460 PLOT 55-I,Y+S:NEXT:RETURN
470 FOR I=A TO B:PLOT X,I+S
480 PLOT 55-X,I+S:NEXT:RETURN

```

Tafel 10: Wichtige Adressen

3C00/1H	enthält Programmbeginn
3C02/3H	enthält Programmlänge
3C04/5H	enthält RAMTOP
3C06H	enthält Zeilenzahl
3C08/9H	enthält Kursorposition
3C80H	Beginn des Bildspeichers
Unterprogramme	
28H	Ausgabe des A-Registers auf den Bildschirm
36H	Abschalten der Bildschirmanzeige durch DI; der Computer arbeitet dann wesentlich schneller; PRINT und INPUT schalten die Anzeige wieder ein
56H	Bildschirm löschen und formatisieren
0C4H	Tastatureingabe eines Zeichens ins A-Register
0E0H	Tastaturabfrage (INKEY), im A-Register steht 0 oder das gedrückte Zeichen
0D8FH	Tonbandausgabe (SAVE)
0E6CH	Tonbandeingabe (LOAD)

Telefonbuch

Mit diesem Programm sollen die Möglichkeiten der Zeichenkettenverarbeitung dargestellt werden.

```

10 REM 'TELEFONBUCH-GOTO 30'
20 DIM N(10,10),T(10,10),S(10)
30 INPUT 'EINGABE:1 SUCHEN:2',E:
40 GOTO E*1000
1000 Z=Z+1:INPUT 'NAME';N(Z):
1010 INPUT 'TELEFON';T(Z):
1020 GOTO 30
2000 INPUT 'NAME';S:
2010 FOR I=1 TO Z
2020 IF S=N(Z)'TELEFON';T(I):
2030 NEXT?:GOTO 30

```

Maschinenkodeunterprogramme

Mit der DATA-Anweisung können Maschinenkodeunterprogramme einfach in ein Basicprogramm eingebunden werden. Vor der Ausführung muß es mit POKE an das Ende des RAM-Bereichs gebracht werden. Das Beispiel schreibt die Buchstaben MC auf den Bildschirm.

```

10 RESTORE
20 FOR I=3FF0H TO 3FF6H
30 READ X:POKE I,X:NEXT
40 Y=USR(3FF0H)
50 REM 'MC-UNTERPROGRAMM'
60 DATA
3EH,4DH,0EFH,3EH,43H,0EFH,0C9H

```

Labyrinth

Als letztes folgt ein Bildschirmspiel (Tafel 9). 'C' (=43H) kann mit den Tasten V, T, F und G bewegt werden. Das Ziel ist, alle Punkte (=2EH) aufzusammeln. Dabei muß den Zeichen 'X' (=58H) ausgewichen werden. Das Programm benötigt 1 Kbyte Zusatzspeicher und läuft mit 18 Zeilen.

Literatur

- [1] Schiller, E.: Basic-Heimcomputer, radio fernsehen elektronik, Berlin 34 (1985) 1, S. 13-18
Berichtigungen: Heft 4 (1985) S. 205 und Heft 6 (1985) S. 341
- [2] Benning, K.: Amateurschaltkreise RAM S 256 C/D, RAM S 214 D, S 2141 D und S 2142 D, Funkamateure, Berlin 34 (1985) 4, S. 181-184

8-Kbit-Vollgrafik für Basic-Heimcomputer

Dipl.-Ing. WERNER SCHMIDT

Mitteilung aus der Interessengemeinschaft Heimcomputer der IH Dresden

Schaltungsprinzip

Der (8K×1)-bit-Bildwiederholpeicher wird beim Lesen von zwei 8-bit-Zählern adressiert (Bild 1). Der erste dieser Zähler, der durch den Bildpunktaktgenerator (2,91 MHz) angesteuert und als Aufwärtszähler betrieben wird, liefert mit seinen niederwertigen sieben Bits 128 Bildpunkte je Zeile (geschrieben von links nach rechts). Am Zeilenende wird der Zähler durch den Zeilensynchronimpuls SZ des Rechners synchronisiert und zurückgesetzt. Der zweite Zähler wird direkt von dem Zeilensynchronimpuls SZ angesteuert und als Abwärtszähler betrieben. Seine oberen sechs Bits erzeugen 64 Zeilen (geschrieben von oben nach unten). Der Bildsynchronimpuls SB setzt den Zähler auf FFH, damit das obere Ende des Grafikadreßbereiches dem oberen Bildrand zugeordnet wird. Der Nulldurchgang steht dann für den un-

Als Ergänzung zu dem in [1] beschriebenen Basic-Heimcomputer wurde eine 8-Kbit-Vollgrafik entwickelt, die in einem zweiten Bildwiederholpeicher untergebracht ist und wahlweise zugeschaltet werden kann. Dadurch ergeben sich interessante Anwendungsmöglichkeiten, vor allem für die Lösung mathematischer und physikalischer Aufgaben. Als notwendig erwiesen sich dazu eine eigene Bildschirmsteuerung und ein 13 bit breiter Adreßumschalter.

teren Bildrand. Damit entspricht dieses Grafikkonzept in der Zählrichtung von links unten nach rechts oben (Bild 2) dem in der Mathematik üblichen X-Y-Koordinatensystem und erleichtert die Programmierung von Darstellungen mathematischer Funktionen.

Die gelesene Bildpunktinformation wird mit den verschiedenen Austastsignalen für die Dunkelastung des Feldaußenrandes gekoppelt und über einen Einstellregler für den Grauwert und eine Diode dem BAS-Signal des Rechners zugeführt. Der Einstellregler erlaubt eine Einstellung des Grafikkontrastes, wobei der Weißwert der Grafik mit der des Rechners identisch ist. Eingeschrieben wird die Grafikinformation durch die Basicanweisung POKE Z,I. Während der Ausführung dieser Anweisung wird, selektiert durch die obersten drei Bits von Z, der Speicheradreßbus kurzzeitig auf den

Adreßbus des Rechners geschaltet und die Information I (0 ≙ grau, 1 ≙ weiß) eingeschrieben.

Beschreibung der einzelnen Baugruppen

Taktgenerator und Bildpunktzähler (Bild 3)

Die Einstellung der Frequenz des TTL-Oszillators erfolgt visuell nach der Breite des Grafikfeldes. Nach 128 Bildpunkten erhält Bit 7 des Bildpunktzählers (bezeichnet mit R7) H-Pegel, so daß der rechte Feldaußenrand dunkel getastet wird. Der in diesem Zeitabschnitt vom Rechner gelieferte Impuls SZ triggert einen monostabilen Multivibrator, dessen Ausgangsimpuls eine Breite von 10 µs aufweist. Dieser tastet den linken Feldaußenrand dunkel und synchronisiert den Bildpunkt-Taktgenerator, indem er ihn für die Dauer des Impulses sperrt und gleichzeitig den Zähler zurücksetzt.

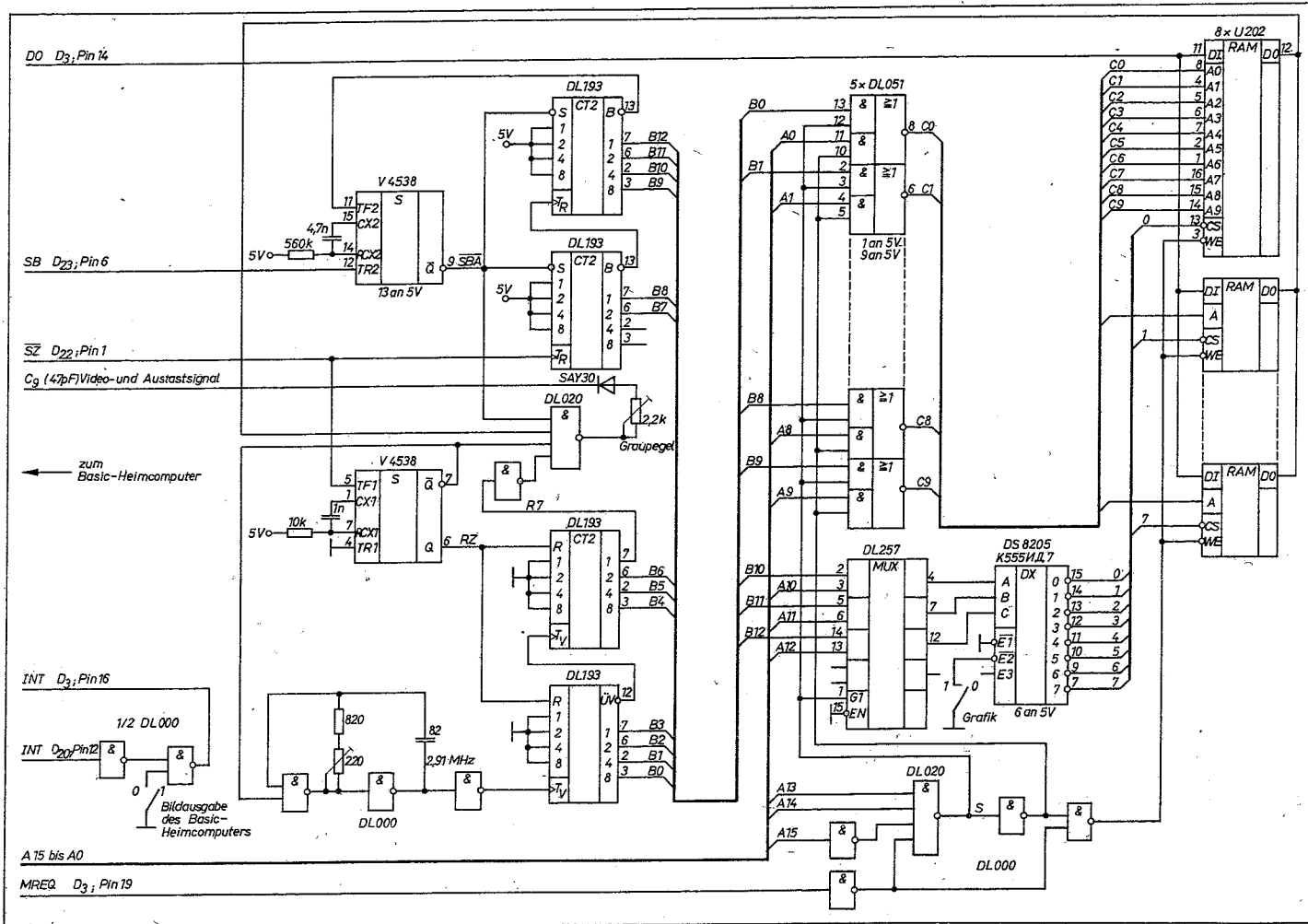


Bild 1: Stromlaufplan der 8-Kbit-Vollgrafik.

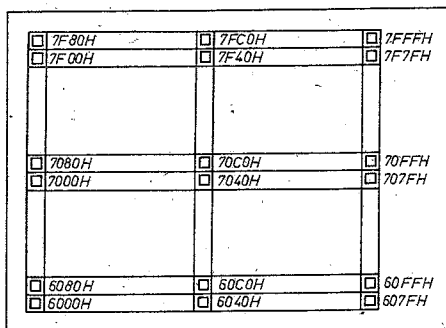


Bild 2: Aufbau des Grafikfeldes mit Angabe einiger ausgewählter Adressen. Die Adresse 7040 H ist vorzugsweise als Koordinatenmittelpunkt geeignet. Bezogen auf diese Adresse, geht der Wertebereich des normierten x von -60...60 und des normierten y von -30...30.

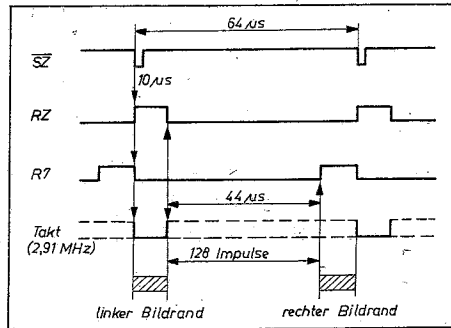


Bild 3: Horizontale Bildschirmsteuerung

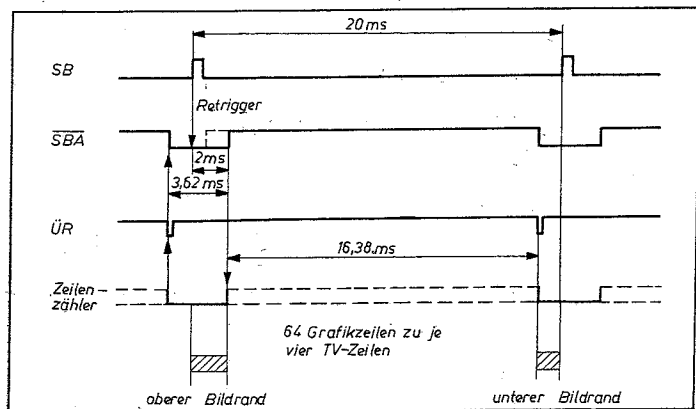


Bild 4: Vertikale Bildschirmsteuerung

Grafikzeilenzähler (Bild 4)

Annähernd quadratische Grafikfeldelemente werden erzielt, wenn diese eine Höhe von vier Fernsehzeilen aufweisen. Aus diesem Grunde wird der Zähler erst ab Bit 2 für die BWS-Adressierung genutzt. Die freien Bits 1 und 0 könnten prinzipiell bei gleichzeitiger Vervielfachung der Bildpunkttaffrequenz und einer entsprechenden RAM-Erweiterung für eine höhere Grafikaufklärung eingesetzt werden. Dem steht allerdings der begrenzte Zahlenbereich des Rechners entgegen, der keine Adressierung über die Adresse 7FFF H hinaus erlaubt.

Zur Synchronisation mit der Rechnerbildschirmsteuerung wird die Retriggereigenschaft des verwendeten monostabilen Multivibrators genutzt. Dieser wird zunächst nach Abschluß der untersten Grafikzeile durch den Übertragsimpuls beim Nulldurchgang des Rückwärtszählers getriggert und gibt einen Impuls mit der Breite von 2 ms ab. Wie aus Bild 4 hervorgeht, trifft während dieses Intervalls der vom Rechner bereitgestellte Impuls SB ein, der die Retriggerung auslöst, so daß sich eine effektive Impulsbreite von 3,62 ms ergibt. Dieser als SBA bezeichnete Impuls tastet einmal den oberen und den unteren Feldaußenrand dunkel. Zum anderen setzt er über den Ladeeingang S den Zähler auf FF H. Die Freigabe des Zählers erfolgt nach Ablauf des Intervalls SBA. Für den Fall, daß die Zählung im Grafikfeld links oben beginnen und rechts unten enden soll, ist der Zähler durch SBA auf 0 H zu setzen und

Tafel 1: Programm zum Löschen des Grafikfeldes

```
10 Z = 6000 H           (Anfangsadresse,
                        links unten)
20 POKE Z,1             (1 ≙ weiß)
30 IF Z = 7FFF H END    (Endadresse,
                        rechts oben)
40 Z = Z + 1; GOTO 20
```

Tafel 2: Programm für die Darstellung einer horizontalen Geraden

```
10 Z = 70A0 H           (0 ≙ grau)
20 POKE Z,0
30 IF Z = 70F0 H END
40 Z = Z + 1; GOTO 20
```

Tafel 3: Programm für die Darstellung einer vertikalen Geraden

```
10 Z = 63B0 H
20 POKE Z,0
30 IF Z = 78B0 H END
40 Z = Z + 128; GOTO 20
```

Tafel 4: Programm für die Darstellung der Funktion $y = x/2 + 10$

```
10 X = -60
20 Y = X/2 + 10
30 Z = 7040 H + 128 * Y + X
40 POKE Z,0
50 PRINT X,Y,/
60 X = X + 1
70 IF Z < 7F7F H GOTO 20
80 END
```

Tafel 5: Programm für die Darstellung der Funktion $y = 5x^2/220 - 0,66x - 10$

```
10 X = -30
20 B = X * 66/100 + 10
30 Y = X * X * 5/220 - B
40 Z = 7040 H + 128 * Y + X
7040 H: Adresse des
Koordinaten-
mittelpunktes

50 POKE Z,0
60 PRINT X,Y,/
70 IF X = 60 END
80 X = X + 1; GOTO 20
```

Tafel 6: Programm für die Darstellung der Funktion $y = x^3/200 - 2x$

```
10 GOSUB 100
20 X = -25
30 Y = X * X * X * Y / 200 - 2 * X
40 Z = 7040 H + 128 * Y + 2 * X
(X-Achse um
Faktor 2
gedehnt)

50 POKE Z,0; PRINT X,Y,/
60 IF X = 25 END
62 A = 0
63 A = A + 1
64 IF A < 100 GOTO 63
70 X = X + 1; GOTO 30
100 Z = 6000 H
110 POKE Z,1
120 IF Z = 7FFF H GOTO 140
130 Z = Z + 1; GOTO 110
140 Z = 7FC0 H
150 POKE Z,0
160 IF Z = 6040 H GOTO 180
170 Z = Z - 128; GOTO 150
180 Z = 7000 H
190 POKE Z,0
200 IF Z = 707F H RETURN
210 Z = Z + 1; GOTO 190
(Zeitschleife)
```

Tafel 7: Programm für die Darstellung von zwei sich schneidenden Geraden

```
(mit Anfangslöschung, Koordinatenkreuz, Ausgabe
der Schnittpunktskoordinaten und universell wirken-
der Begrenzung des zu adressierenden Grafikbe-
reiches)

1 A = 6000 H;           (80 H = 128)
B = 7F7F H;
N = 80 H
(2 GOTO 4)
3 GOSUB 100
4 INPUT 'M= ,M,/      (Koordinatenmittelpunkt,
                        z. B. 7040 H)

5 GOSUB 200
10 X = -1 * D
20 Y = X/2 + 10        (Gerade I)
25 GOSUB 300
30 V = Y
40 Y = X                (Gerade II)
45 GOSUB 300
50 IF Y = V PRINT X,Y,/
60 IF = N - D END
70 X = X + 1; GOTO 20

100 Z = A                (UP Grafikfeld löschen)
110 POKE Z,1
120 IF Z = B + N RETURN
130 Z = Z + 1; GOTO 110

200 C = (M - A) / N      (UP Koordinatenkreuz)
210 D = M - A - N * C
220 E = A + D; F = M - D
230 POKE E,0            (X-Achse)
240 E = E + N

250 IF E < B GOTO 230
255 G = 0
260 POKE F + G,0        (Y-Achse)
265 G = G + 1
270 IF G < N GOTO 260
280 RETURN

300 Z = M + N * Y + X    (UP Darstellung der Funk-
                        tion y = f(x))
310 IF Z < A GOTO 340
320 IF Z > B GOTO 340
330 POKE Z,0; RETURN
340 Z = M; GOTO 330
(untere BWS-Grenze)
(obere BWS-Grenze)
```

der Zähler selbst als Vorwärtszähler zu betreiben.

Adreßumschalter

Für die unteren zehn Bits wurde die IS DL 051 D, für die restlichen drei Bits der Multiplexer DL 257 D benutzt. Das Umschaltensignal wird durch Dekodierung der Rechneradressen A13, A14 und A15 gewonnen, die bei jeder POKE-Anweisung mit Z im Bereich 6000 H bis 7FFF H entsprechend aktiviert werden.

Bildwiederholtspeicher

Die Auswahl der acht Speicherschaltkreise erfolgt durch Dekodierung der Adressen

Halbleitertechnik

● Mit CHEMFET wird ein Halbleiterbauelement bezeichnet, das als Sensor arbeitet und mit dem bestimmte Ionenarten in einer Flüssigkeit nachgewiesen und nach Auswertung auch angezeigt werden können. Der Sensor kann auf bestimmte Ionen abgestimmt werden, wobei eine in das Transistorsystem integrierte Membran nur auf Ionen des jeweils interessierenden Stoffes reagiert. Dabei wird die Schwellspannung des Transistors verändert. Diese Änderung wird elektrisch ermittelt und einem A-D-Wandler zugeführt. Je nach verwendeter Membran lassen sich verschiedene Ionenarten erkennen.

C10, C11 und C12. Der Eingang E2 des Dekoders ermöglicht ein einfaches Abschalten der Grafik, z. B. während der Programm-entwicklung. Eine wesentliche Vereinfachung im Aufbau ist durch den Einsatz von zwei IS U 214 oder U 224 zu erreichen. Der Adreßdekoder müßte dann durch den Multiplexer DL 251 D ersetzt und die Grafikabschaltung an dessen Strobeingang angebracht werden, falls dieser nicht zum Takten benötigt wird. In der vorliegenden Schaltung erwies sich ein Takten des Dekoders als nicht erforderlich. Das Signal WE wird aus dem Umschaltensignal erzeugt. Wesentlich ist, daß WE erst dann aktiv wird, wenn die Adreßumschaltung abgeschlossen ist. Es kann daher nicht mit S direkt verbunden werden. In der praktischen Erprobung stellte sich heraus, daß auf das Signal WR des Rechners bei der Bildung von WE ohne Nachteile verzichtet werden konnte.

Abschalten der Bildausgabe des Rechners

Dies erfordert einen Eingriff in den Rechner und beruht auf der Sperrung der INT-Anforderung durch den CTC-Schaltkreis. Damit wird die für die Bildschirmsteuerung vorgesehene Interruptserviceroutine nicht aufgerufen, so daß der Rechner das Grafikprogramm mit uneingeschränkter Geschwindigkeit abarbeiten kann. Dieser Umstand ist besonders beim Löschen des Grafikspeichers von Bedeutung, da jede Zelle über Basicanweisungen einzeln gelöscht wird. Der Löschvorgang für das gesamte Grafikfeld läßt sich damit von 50 s auf 30 s verkürzen.

Programmbeispiele

Die Tafeln 1 bis 7 enthalten die Programme für die Darstellung von mathematischen Funktionen auf dem Bildschirm und zum Löschen des Grafikfeldes. Wie bei allen Speicherschreibenweisungen ist auf eine exakte Eingrenzung des zu adressierenden Bereiches zu achten, da andernfalls die Gefahr besteht, daß die außer Kontrolle geratenen Schreibanweisungen das im Arbeitsspeicher stehende Programm zerstören.

Literatur

- [1] Schiller, E.: Basic-Heimcomputer. radio fernsehen elektronik, Berlin 34 (1985) 1, S. 13-18
- [2] Streng, K. K.: Daten digitaler integrierter Schaltkreise. Berlin: Militärverlag der DDR 1985.

● Mit dem Magnetfeld-Czochralski-Verfahren lassen sich qualitativ hochwertige Kristalle mit optimalem Sauerstoffgehalt herstellen. Neben Si kann auch GaAs als Halbleitermaterial benutzt werden. Zum Einbringen der Sauerstoffatome in das Einkristallgitter wird ein vertikales oder horizontales Magnetfeld herangezogen. Die Kristallziehgeschwindigkeit konnte auf 1,6 bis 2,0 mm/min erhöht werden. Der Tiegel wird beim Ziehen gedreht. Durch die Einhaltung eines richtigen Temperaturprofils können die Defekte niedrig gehalten werden. Die Wärme ist am Rande der Schmelzzone am größten. Von oben nach unten findet entlang der Achse eine Temperaturverringering statt.

Basic-Heimcomputer

Ing. MICHAEL WOLTER

Beim Nachbau des Basic-Heimcomputers BCS-3 nach [1] und [2] gab es einige Probleme mit der Bildschirmsteuerung. In der Version 2.4 arbeitete der Rechner anfangs nicht. Eine Kontrolle der Pegel an der CPU ergab L-Pegel am /WAIT-Eingang. Nachdem entgegen der Berichtigung in [3] Pin 11 von D_{1,2} mit Pin 6 von D₂₂ verbunden wurde, arbeitete der Computer ohne Beanstandung. Beim Umrüsten auf die Version 3.1 von [2] trat ein Versatz des Bildschirmaufbaues auf. Der Zeilenanfang begann im letzten Drittel des Bildschirms. Änderungen in der Impulsverlängerung der IS D₂₂ lösten das Problem nicht. Ein Vergleich der Bildschirminterruptroutinen (BINT) beider Versionen ergab, daß in der Version 2.4 dreimal soviel Befehle zwischen Beginn der Routine und der /WAIT-Auslösung zur Zeilensynchronisation wie in der Version 3.1 abzuarbeiten waren (Version 2.4: 121 Taktzyklen, Version 3.1: 42 Taktzyklen). Aus diesem Grunde wurde die BINT-Routine auf die Adresse 03F H (vier freie By-

tes) vorverlegt und eine Zählschleife eingefügt.

Die Länge der Verzögerung wird in das B-Register geladen. Im vorliegenden Fall lag das Optimum bei 5.

Das geänderte Programm zeigt Tafel 1, in Tafel 2 sind die zu ändernden Bytes angegeben. Der Computer arbeitet mit dem geänderten Programm ohne Beanstandungen.

Als günstig hat sich die Auslegung von R₃₂ als ein von außen zugängliches Potentiometer erwiesen. Damit läßt sich das Bild ohne Probleme in der Horizontalen einstellen und ein gelegentlich auftretendes Bildwackeln (keine quarzstabilisierte Frequenz) kompensieren.

Tafel 2: Angabe der zu ändernden Bytes

Adresse	neuer Inhalt (alles hexadezimal)
002 A	3 F
003 F	08
0040	DB FA FE 04 20 EC D9 06
0048	05 10 FE 32 00 14

Tafel 1: Geändertes Programm

Adresse	Operatonskode	Mnemonic	Bemerkungen
003 F	08	BINT: EXAF	
0040	DBFA	IN OFAH	
0042	FE04	CMP 4	
0044	20EC	JRNZ OECH	
0046	D9	EXX	
0047	0605	LDB, 05 H	Zähler laden
0049	10FE	Z: DJNZ Z	Zähler
004 B	320014	LD(1400 H), A	/WAIT-Auslösung
004 E	87	ADD A	Beginn des ungeänderten Bereiches

Literatur

- [1] Schiller, E.: Basic-Heimcomputer. radio fernsehen elektronik, Berlin 34 (1985) 1, S. 13-18
- [2] Schiller, E.: Erweiterungen für Basic-Heimcomputer. radio fernsehen elektronik, Berlin 35 (1986) 9, S. 559-563
- [3] Berichtigung. radio fernsehen elektronik, Berlin 34 (1985) 4, S. 205

Stellungnahme des Autors

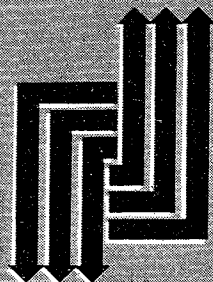
Am Pin 1 von D₂₂ liegen negative Impulse mit einer Breite von etwa 4 µs an. Diese Impulse schalten am Bildanfang über D_{26,3} das WAIT-Signal ab, das durch den Befehl LD A,(1400H) ausgelöst wird. Damit wird erreicht, daß die Erzeugung des Bildschirmbildes immer an der gleichen Stelle beginnt. Dieses Verfahren hat bei allen mir bekannten Nachbauten zur vollsten Zufriedenheit funktioniert.

Bei Herrn Wolter wird vermutlich das Signal vom Pin 1 von D₂₂ auf Grund eines technischen Fehlers nicht ordnungsgemäß erscheinen. Alle angeführten Maßnahmen beseitigen zwar die Folgen, nicht aber die Ursache des Defektes. Auch das von ihm beobachtete Bildwackeln resultiert nicht aus der schlechten Frequenzkonstanz des Taktgenerators, sondern aus der unsicheren Überlappung von WAIT-Abfrage und Zeilenimpuls bei seiner Schaltungsvariante.

Vorteilhaft kann allerdings manchmal die Verschiebbarkeit des Bildes sein, da bei schlecht justierten Fernsehempfängern ein Buchstabe am rechten Rand verschwindet.

Eckhard Schiller

Bausteine der Automatisierungstechnik



VEB VERLAG TECHNIK BERLIN

Neuerscheinung

Bausteine der Automatisierungstechnik

Herausgegeben vom Institut für Fachschulwesen der DDR. Gesamtedaktion und Federführung: Dr.-Ing. Heinz Kolb und Dr.-Ing. Günter Graichen. 368 Seiten, 342 Bilder, 47 Tafeln, Kunstleder, DDR 25,- M, Ausland 35,- DM. Im Fachbuchhandel erhältlich. Bestellangaben: 5536124/Kolb, Bausteine

Umfassendes Fachschullehrbuch über die in den Automatisierungseinrichtungen benötigten Bauelemente und Funktionseinheiten (elektronische, elektromechanische, pneumatische und hydraulische). Die Darstellung ist auf die praktische Anwendung orientiert und auf den Nutzer von Einrichtungen der Automatisierungstechnik zugeschnitten. Entsprechend der zunehmenden Bedeutung elektronischer/mikroelektronischer Bauelemente und Bausteine für die Automatisierungstechnik bilden diese einen Schwerpunkt im Buch.

VEB VERLAG TECHNIK BERLIN



wendet werden (funktionale Programmiersprachen). Es ist jedoch sinnvoll, eine Bibliothek von standardisierbaren Verarbeitungsprozeduren zu schaffen, die von einer verbreiteten höheren Programmiersprache für den Hostrechner, d.h. einer imperativen Sprache, als externe Prozeduren nutzbar sind. Dieses Vorgehen entspricht dem vorgesehenen Einsatz des μ PD 7281 als schneller Coprozessor für preiswerte Hostsysteme (Personalcomputer).

Zusammenfassung

Die vorgestellte Einwicklungsumgebung für den Signalprozessor μ PD 7281 unterstützt Entwurf und Test von Hard- und Software für diesen Prozessor auf in der DDR produzierten Personalcomputern. Neben der Möglichkeit, Personalcomputer zu hochleistungsfähigen Signalverarbeitungssystemen auszubauen, ist zusätzlich eine praktische Einbeziehung der Datenflußarchitektur in die Aus- und Weiterbildung gegeben.

Literatur

- [1] Zoicas, A.; Look, G.: Datengesteuerter Prozessor beschleunigt die Bildverarbeitung. *Elektronik*, München 34 (1985) 3, S. 39-43
- [2] μ PD 7281 Image Pipelined Processor. Product Description, NEC Electronics (Europe) GmbH, 1986
- [3] Bildprozessorkarte verarbeitet 20 Millionen Befehle/s. *Elektronik*, München 36 (1987) 2, S. 32
- [4] Giloi, W. K.: Datenflußprinzipien in Rechnerarchitekturen. NTG-Fachberichte Band 80. Berlin: VDE-Verlag 1982
- [5] Rieken, R.: Einige Aspekte der Datenflußarchitektur. *Nachrichtentechnik-Elektronik*, Berlin 38 (1988) 5, S. 173-177
- [6] Anwenderdokumentation PLZ/SYS. VEB Robotron Buchungsmaschinenwerk Karl-Marx-Stadt, 1984
- [7] Hübner, U., Held, U.: Programmiersprache PLZ für 16-bit-Mikroprozessoren. *Nachrichtentechnik - Elektronik*, Berlin 36 (1986) 11, S. 408 bis 410
- [8] Rehm, W.: Universelles 16-bit-System USS 8000. *radio fernsehen elektronik*, Berlin 34 (1985) 5, S. 282-285
- [9] Rieken, R.: Bildverarbeitungssystem mit Mehrrechneranordnung auf der Basis von 16-

bit-Prozessoren. Tagungsbericht der 1. Internationalen Fachtagung „Automatische Bildverarbeitung“ Berlin, 1985

- [10] Rieken, R.: Pipeline-Prozessor-Debugger 7281. Dokumentation zur Version 3.2. Sektion Informationstechnik der TU Karl-Marx-Stadt, 1988
- [11] Srinii, V. P.: An Architectural Comparison of Dataflow Systems. *IEEE-Computer*, New York 19 (1986) 9, S. 68-88
- [12] Treleaven, P. C.; Brownbridge, D. R.; Hopkins, R. P.: Data-Driven and Demand-Driven Computer Architecture. *Computing Surveys*, New York 14 (1982) 1, S. 93-143
- [13] Kerner, H.: Funktionales Programmieren und Datenflußrechner. *Elektronische Rechenanlagen*, München 25 (1983) 6, S. 143-151
- [14] Backus, J.: Function-level computing. *IEEE-Spectrum*, Philadelphia 19 (1982) 8, S. 22-27
- [15] Arvind; Gostelow, K. P.: The U-Interpreter. *IEEE-Computer*, New York 15 (1982) 2, S. 42-49
- [16] Dennis, J. B.; Gao, G. R.; Todd, K. W.: Modeling the Weather with a Data Flow Supercomputer. *IEEE-Transactions on Computers*, New York 33 (1984) 7, S. 592-603
- [17] Hanaki, S.; Temma, T.: Template-Controlled Image Processor (TIP) Project. *Multicomputers and Image Processing - Algorithms and Programs*. London: Academic Press 1982

Ergänzungen für Basic-Heimcomputer BCS-3

Dipl.-Ing. TORSTEN MUSIOL

Maschinenkode-Editor

Der erweiterte Heimcomputer BCS-3 [2] besitzt zwar einen gegenüber der ersten Version wesentlich verbesserten Basicinterpreter, jedoch ist die Lösung anspruchsvoller Aufgaben auf Grund des Interpreterprinzips und des Fehlens eines schnellen Massenspeichers kaum möglich. Soll der Computer jedoch als Steuergerät für verschiedenste Anwendungen bzw. zur Entwicklung von kleineren Mikrorechnersteuerungen verwendet werden, muß die Möglichkeit gegeben sein, Maschinenprogramme editieren und testen zu können. Durch die Einbindung von Maschinenprogrammteilen können so auch Basicprogramme wesentlich verbessert werden.

Der hier beschriebene bildschirmorientierte Maschinenkode-Editor in der Version für 29 Zeichen/Zeile beansprucht zusammen mit dem noch vorzustellenden EPROM-Programmiermodul einen Speicherraum von weniger als 1 Kbyte. Sie sind somit auf einer zusätzlichen IS U 555 unterzubringen, die in diesem Beispiel im Bereich F000H bis F3FFH selektiert wurde. Nach Eingabe des Maschinenkodes (s. Tafel 1) und Start des Basicprogramms (Tafel 2) kann der Editor auch für einen anderen Speicherbereich lauffähig gemacht werden.

Denkbar wäre z. B. das Ende eines, natürlich erweiterten, RAM-Bereiches, wobei dann zwar der zusätzliche EPROM entfällt, das Programm aber ständig von Kassette geladen werden müßte.

Voraussetzung für den Betrieb des Editors ist die Einstellung der Zeilenanzahl von mindestens 20. Der Aufruf erfolgt im einfachsten

Der in [1] vorgestellte und nach [2] erweiterte Basic-Heimcomputer BCS-3 stellt einen guten Kompromiß zwischen Bauelementeaufwand und Leistung dar. Im folgenden Beitrag sollen weitere Ergänzungen beschrieben werden, die ebenfalls unter der Prämisse eines möglichst geringen Bauelementeeinsatzes entwickelt wurden.

Fall mit Hilfe der Basicfunktion USR [z. B. mit $Y = USR(0F000H)$]. Möglich ist auch die Verwendung einer übergeordneten Prozedur zur Kommandoingabe, die jedoch etwas umfangreichere Änderungen im EPROM D₆ erfordert.

Der Editor meldet sich mit der Ausschrift EDIT>. Von dieser Stelle aus sind die Kommandos S(ubstitute), M(ove), G(o), T(est) und Q(uit) möglich.

- Kommando S(ubstitute): S<adr>

<adr> gibt die Anfangsadresse des zu editierenden Speicherbereichs an. Grundsätzlich werden Daten mit zwei und Adressen mit vier Stellen eingegeben, wobei die in Basic übliche Kennzeichnung der Hexadezimalzahlen mit H entfällt.

Nach Eingabe des Kommandos erscheint ein Ausdruck des Speicherbereichs so auf dem Bildschirm (8 Zeilen mit je 8 byte), daß in der ersten Zeile ein Cursor > auf die gewünschte Speicherzelle zeigt, wobei die Anfangsadresse jeder Zeile ein Vielfaches von 8 ist. Damit wird die Orientierung im Speicherbereich wesentlich erleichtert. Unterhalb dieses 8 x 8-Feldes erscheint die jeweils aktuelle Speicheradresse, hinter der der neu zu schreibende Wert eingegeben wird.

Fehlerhafte Eingaben können wie auch bei der Adressenwahl mit Hilfe der Delete-Taste korrigiert werden, oder es wird einfach fortlaufend geschrieben, wobei die zuerst eingegebenen Ziffern bis auf zwei bzw. vier Stellen verschwinden. Nach der Eingabe eines Bytes rückt der Cursor automatisch eine Stelle weiter. Eine schnelle Bewegung des Cursors nach oben, unten,

links und rechts ist mit Hilfe der Tasten O, L, I und P möglich. Die Lage dieser Tasten erlaubt eine recht bequeme Cursorsteuerung. Ebenso würde sich eine getrennte Anordnung von vier Tasten, die den oben genannten parallel geschaltet werden, anbieten. Bei Überschreiten der ersten bzw. letzten Zeile rollt das Feld eine Zeile nach unten bzw. oben. Nach Betätigung der Taste Q (Quit) wird ein Rücksprung zum Kommandoingabestatus EDIT> ausgeführt.

- Kommando M(ove): M<adr1> -<adr2> -><adr3>

Mit diesem Kommando wird der Speicherbereich von <adr1> bis <adr2> auf den Bereich mit der Anfangsadresse <adr3> kopiert. Die Zeichen - nach Eingabe von <adr1> und -> nach Eingabe von <adr2> werden vom Programm geliefert. Die beiden Speicherbereiche dürfen sich überlappen.

- Kommando G(o): G<adr>

Durch dieses Kommando wird ein Programm mit der Anfangsadresse <adr> gestartet. Wird ein RETURN-Befehl erreicht, erfolgt der Rücksprung zum Kommandoingabestatus.

- Kommando T(est): T<adr>

Dieses Kommando führt wie das Kommando G(o) ein auf der Adresse <adr> beginnendes Programm aus. Beim Rücksprung werden hier jedoch in einer Zeile die Inhalte der Register in der Reihenfolge AF, BC, DE und HL ausgegeben.

- Kommando Q(uit): Q

Es erfolgt ein Rücksprung zum aufrufen-

Tafel 1:
Hexadezimalausdruck
(Bereich F000H bis
F3FFH)

F000	CD 56 00 21 39 F0 06 07 7E EF 23 10 FB CD C4 00
F010	FE 51 C8 21 03 F0 E5 FE 5A D0 FE 40 D8 6F CD 33
F020	F1 D8 7D EB FE 53 28 18 FE 4D CA C7 F0 FE 47 28
F030	07 FE 54 C0 01 8E F3 C5 E9 1E 1E 45 44 49 54 3E
F040	E5 EB CD 56 00 7B E6 0F FE 08 38 02 D6 08 67 7B
F050	E6 F8 5F 24 2E 01 22 10 3C 01 B2 3C 03 03 03 25
F060	20 FA C5 0E 08 CD B3 F2 C5 0E 1D 2A 08 3C 09 22
F070	08 3C C1 0D 20 EF C1 D1 3E 3E 02 21 C0 3E 22 08
F080	3C EF CD B2 F1 EB CD 2D F1 F5 C5 37 3F 0E 02 CC
F090	E6 F0 CD 1D F1 C1 30 03 F1 18 EB F1 20 0C 73 03
F0A0	ED 43 08 3C CD AD F1 0B 3E 50 CD C9 F1 EB 18 C8
F0B0	3E 2D EF CD 33 F1 D8 EB ED 52 30 06 EB CD 1D F1
F0C0	18 F1 23 E5 C1 EB C9 CD B0 F0 3E 2D EF 3E 3E EF
F0D0	CD 33 F1 D8 E5 ED 52 E1 38 03 ED B0 C9 09 2B EB
F0E0	09 2B EB ED B8 C9 E5 97 21 14 3C 77 23 77 ED 5B
F0F0	24 3C 1A FE 47 30 21 FE 30 38 1D FE 3A 38 06 FE
F100	41 38 15 D6 07 E6 0F 13 06 02 CD FF 0B 23 23 0D
F110	20 E0 5E 2B 56 97 E1 C9 00 00 00 37 E1 E5 2A 24
F120	3C 22 08 3C 06 08 36 20 23 10 FB E1 C9 C5 E5 0E
F130	02 18 04 C5 E5 0E 04 51 06 00 2A 08 3C 22 24 3C
F140	22 08 3C 36 5F CD C4 00 04 2A 08 3C FE 5C 20 0A
F150	2B 05 28 2D 36 20 05 2B 1B E6 FE 47 30 14 79 90
F160	3C DE 2B 15 20 FC 54 5D 1B C5 06 00 ED B0 C1 51
F170	18 E4 C8 51 28 08 FE 7F 20 DA CD E6 F0 38 B6 18
F180	02 FE 7F E1 C1 C9 D5 C5 21 14 3C 73 2B 72 79 FE
F190	04 28 01 23 16 02 97 ED 6F F6 30 FE 3A 38 02 C6
F1A0	07 EF 15 20 F1 23 0D 0D 20 EA C1 D1 C9 C5 0E 02
F1B0	18 03 C5 0E 04 E5 CD 86 F1 2A 08 3C 23 79 FE 02
F1C0	28 01 23 22 08 3C E1 C1 C9 D5 F5 3E 20 02 F1 ED
F1D0	5B 10 3C CD DC F1 ED 53 10 3C D1 C9 FE 49 20 16
F1E0	0B 0B 0B 2B 15 C0 16 08 1D 20 05 01 9A 3C 18 6E
F1F0	D5 11 24 00 18 5D FE 50 20 1C 03 03 03 23 14 7A
F200	FE 09 C0 16 01 1C 7B FE 09 20 05 01 29 3E 18 7A
F210	D5 11 24 00 18 37 FE 4F 20 11 C5 01 08 00 97 ED
F220	42 C1 1D 28 39 D5 11 3C 00 18 28 FE 4C 20 12 C5
F230	01 08 00 09 C1 1C 7B FE 09 28 4F D5 11 3C 00 18
F240	0C FE 51 20 04 E1 E1 E1 C9 00 00 00 C9 E5 C5 E1
F250	19 18 06 E5 C5 E1 97 ED 52 E5 C1 E1 D1 C9 D5 C5
F260	E5 21 05 3E 11 41 3E 01 86 01 ED B8 21 80 3C 22
F270	08 3C 22 24 3C 0E 04 CD E6 F0 21 08 00 EB 97 ED
F280	52 EB CD B3 F2 E1 C1 D1 1C C9 D5 C5 E5 21 BC 3C
F290	11 80 3C 01 B6 01 ED B0 21 24 3E 22 08 3C 22 24
F2A0	3C 0E 04 CD E6 F0 21 08 00 19 EB CD B3 F2 E1 C1
F2B0	D1 1D C9 CD B2 F1 06 08 1A 13 D5 5F CD AD F1 D1
F2C0	10 F6 C9 CD 56 00 FB 21 87 F3 06 07 7E EF 23 10
F2D0	FB 21 C6 F2 E5 CD C4 00 FE 54 20 06 97 11 00 00
F2E0	18 40 FE 52 20 10 CD 5E F3 C8 CD 7C F3 21 00 00
F2F0	01 00 04 ED B0 C9 FE 50 20 5F CD 5E F3 C8 3E 79
F300	D3 F9 EB 3E 32 11 00 E0 01 00 04 E5 F5 3E FF 3D
F310	20 FD F1 ED A0 EA 0C F3 E1 3D 20 E9 3E F9 D3 F9
F320	37 EB CD 7C F3 21 00 E0 01 00 04 FD 21 00 00 1A
F330	38 02 3E FF ED A1 F5 28 08 FD 23 3E 1E EF CD B2
F340	F1 13 F1 EA 2F F3 FD E5 D1 7A B3 C8 CD B2 F1 3E
F350	45 EF 3E 52 EF 3E 52 EF C9 FE 51 C0 E1 C9 CD 33
F360	F1 30 02 97 C9 3E 2D EF D5 EB 01 FF 03 09 EB CD
F370	B2 F1 3E 3F EF CD C4 00 FE 4E D1 C9 F3 06 40 0E
F380	FF 0D 20 FD 10 F9 C9 1E 1E 50 52 4F 4D 3E E5 D5
F390	C5 F5 3E 1E EF 06 04 D1 CD B2 F1 10 FA C9 FB F5
F3A0	3A BF 13 C8 47 2B F9 F1 ED 4D 00 00 00 00 00 00
F3B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
F3C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
F3D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
F3E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
F3F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

konventioneller Weise genutzt. Dabei wurde davon ausgegangen, daß zur Beschleunigung der Operationen Lesen, Testen und Programmieren ein Abschalten der Bildschirmausgabe auf jeden Fall sinnvoll ist. Durch entsprechende Programmierung der CPU und der CTC wird erreicht, daß das INT-Signal während dieser Operationen eine statische Information über den Betriebszustand (Lesen, Programmieren) trägt.

Auf der CPU-Platte ist zur Entkopplung des WAIT-Signals eine Diode zwischen Pin 3 von D_{26.3} und Pin 24 der CPU einzufügen. Die für dynamische RAMs erforderliche Refreshzeit ist zwischen den WAIT-Zyklen vorhanden. Sollten bei der Verwendung von Bastlerschaltkreisen trotzdem Probleme auftreten, läßt sich die Dauer der WAIT-Impulse bei einer entsprechend erhöhten Anzahl der Programmierzyklen verringern.

Es werden insgesamt 50 Programmierzyklen durchlaufen. Im allgemeinen ist jedoch schon eine geringere Anzahl völlig ausreichend. Es ist aber hier schon möglich, die Anzahl der Programmierzyklen durch Änderung der Zelle F304H zu verringern und bei Bedarf den Programmierbefehl zu wiederholen.

Der zu programmierende EPROM wird ab Adresse E000H selektiert. Auf die Wiedergabe der Adreßdekodierlogik wird hier bewußt verzichtet, da die konkrete Realisierung stark von der bisherigen Ausbaustufe des Rechners und von den zur Verfügung stehenden Bauelementen abhängt.

Es ist zu beachten, daß der EPROM bei eingeschaltetem Rechner nur bei geöffnetem Schalter S₁ gesteckt bzw. gezogen werden darf.

Die Programmiersoftware beginnt in der Version nach Tafel 1 auf Adresse F2C3H und kann analog dem Editor mit Hilfe der Funktion USR aufgerufen werden. Eine Verschiebung des Editors auf einen anderen Bereich wird natürlich auch hier wirksam. Der Kommandoeingabestatus wird durch die Aufschrift PROM gekennzeichnet. An dieser Stelle ist die Eingabe der Kommandos R(ead), T(est), P(rogramming) und Q(uit) möglich.

• Kommando R(ead): R<adr>

Mit <adr> wird die Anfangsadresse des Bereichs angegeben, in den der Inhalt des EPROM kopiert werden soll. Um eine unbeabsichtigte Zerstörung von Speicherbereichen zu vermeiden, gibt das Programm in der Kommandozeile die Endadresse aus und wartet auf eine Bestätigung. Mit der Eingabe von N kann das Kommando abgebrochen werden. Bei jeder anderen Tastenbetätigung wird der Inhalt des EPROM auf den angegebenen Bereich kopiert.

• Kommando T(est): T

Es wird ein Blank-Check durchgeführt. Die Adressen nicht gelöschter Zellen und deren Gesamtanzahl werden ausgegeben. Da während des Tests der Bildschirm nicht beschrieben wird, sind nach Abschluß nur die letzten Adressen und die Gesamtanzahl fehlerhafter Zellen in hexadezimaler Form sichtbar (je nach Einstellung der Zeilenanzahl). Im allgemeinen ist diese Information jedoch ausreichend, da der EPROM dann entweder nicht gelöscht oder defekt ist. Ist der Test erfolgreich, er-

Tafel 2: Basicprogramm zur Verschiebung des Maschinenkodes

```

1 REM 'START MIT GOTO 10!'
10 INPUT 'ANFANGSADRESSE:': A
20 PRINT:PRINT
30 INPUT 'H-BYTE DER ZIELADRESSE:': Z
40 DP=0F0H-Z
50 FOR I=A TO A+3FFH
60 B0=PEEK(I)
70 RESTORE
80 FOR J=1 TO 6
90 READ X
100 IF B0#X GOTO 140
110 B2=PEEK(I+2)
120 IF (B2<0F0H) OR (B2>0F3H) GOTO 140
130 I=I+2: POKE I,B2-D: J=6
140 NEXT:NEXT
150 POKE A+5,0F0H-D
160 POKE A+15H,0F0H-D
170 POKE A+36H,0F3H-D
180 POKE A+2C9H,0F3H-D
190 POKE A+2D3H,0F2H-D
200 DATA 0C3H,0CAH,0CCH,0CDH,0E2H,0EAH
210 END

```

den Programm bzw. bei Aufruf durch die Funktion USR zum Basicinterpreter.

EPROM-Programmiermodul

Der EPROM-Programmiermodul in der hier vorgestellten Version gestattet das Lesen, Testen und Programmieren von EPROMs des Typs U 555. Die Beschränkung auf diesen momentan am leichtesten erhältlichen Typ erwies sich im Sinne eines minimalen Bauelementeaufwandes und der Idee, Maschinenkode-Editor und Programmiersoftware in nur einem zusätzlichen U 555 unterzubringen, als notwendig. Die Programmierschaltung (s. Bild) arbeitet nach dem in [3] beschriebenen Prinzip, d. h., nach Aktivierung einer EPROM-Zelle wird ein WAIT-Signal mit einer Dauer von 1ms erzeugt, währenddessen die Programmierspannung am Schaltkreis anliegt. Im Unterschied zur Schaltung nach [3] wird jedoch die Umschaltung zwischen Lese- und Programmierbetrieb durch die Software vorgenommen. Um zusätzliche Bauelemente einzusparen, wird dazu das INT-Signal in etwas un-

Fachtagung „Kommunikations- und Computertechnik“ 1989, Technische Universität Dresden

Asynchroner Transfermodus – ein neuer Weg zum Breitband-ISDN

Angenäherte Analyse der Warteschlange mit beschränkter Kapazität und Wartezeit

Paketkommunikationsmodul zur Kopplung von Datennetzen

Verkehrseigenschaften von ISDN-Koppelfeldern

Leitungsvermittelte und paketvermittelte Nachrichtennetze, Alternative oder Fusion

Es werden zwei experimentelle Vermittlungssysteme, das modulare digitale Vermittlungssystem MODIVA und ein Paketnetzknotten beschrieben und hinsichtlich der Leistungsfähigkeit der Steuerungen untersucht. Daraus werden erste Schlußfolgerungen für die Gestaltung zukünftiger Nachrichtennetze gezogen.

Erste Betriebserfahrungen mit einem transparenten Transportsystem zur Realisierung eines lokalen Kommunikationsnetzes

Methoden zur Erhöhung der Störsicherheit lokaler Netze

Funktechnik und Mikroelektronik

Messung und Simulation der breitbandigen Übertragungseigenschaften von Mobilfunkkanälen

In der Arbeit wird über Ausbreitungsmessungen berichtet, und es wird ein Funkkanalsimulator vorgestellt, bei dem über 12 durch unabhängige Rauschsignale modulierte Ausbreitungspfade die zeitvariante frequenzselektive Übertragungsfunktion des breitbandigen Mobilfunkkanals nachgebildet werden kann.

Funkortung von UKW-Verkehrsfunkstationen

In diesem Beitrag wird über ein Verfahren zur Standortbestimmung von UKW-Verkehrsfunkstationen berichtet, das auf dem Prinzip der Fremddortung basiert und hinsichtlich der Funktionsfähigkeit bei der Bestimmung von Feststationsstandorten getestet wurde.

Die Signalstruktur des satellitengestützten Ortungssystems GPS

Von dem satellitengestützten Ortungssystem NAVSTAR/GPS wird erwartet, daß es wegen der damit erzielbaren hohen Ortungsgenauigkeit und des relativ geringen Aufwandes beim Nutzer eine weltweite Anwendung im Verkehrswesen findet. Abweichend von anderen Ortungssystemen hat NAVSTAR/GPS eine besondere Signalstruktur, die Trägerschwingungen und die Codes für die Entfernungsmessung umfaßt. Die Struktur wird erläutert und ihre Auswertung abgeleitet.

Informationstheoretische Bewertung digitaler Diversity-Übertragungsverfahren bei nicht-selektiven Kanalfuktuationen

Digitale Signalverarbeitung – Philosophie und Anwendungen

Entwicklung eines Digitalkodes für Farbbildsignale

Frequenzeinstellung und Modulation von Verzögerungsleitungs-Oszillatoren auf der Basis akustischer Oberflächenwellen

Zuverlässigkeitsmodell für fehlertolerante Systeme

Modell eines periodischen stochastischen Prozesses

folgt ein sofortiger Rücksprung zum Kommandoeingabestatus.

- Kommando P(rogramming): P<adr>

Mit <adr> wird die Quelladresse angegeben. Um auch hier Eingabefehler auszu-schließen, kann analog zum Kommando R(ead) ein Abbruch erfolgen. Der Programmiervorgang dauert etwa 2min, ihm folgt eine Verifikation. Im Fehlerfall werden die Quelladressen und die Gesamtzahl falsch programmierter Zellen auf dem Bildschirm ausgegeben.

- Kommando Q(uit): Q

Es erfolgt ein Rücksprung zum aufrufen-den Programm.

Tastenfunktion Pause

Bei längeren Ausgaben, die die jeweils ein-gestellte Zeilenanzahl überschreiten, macht es sich störend bemerkbar, daß das Bild nicht angehalten werden kann. Dieser Nachteil kann durch einige kleinere Änderungen beseitigt werden. Hierbei wird die Interruptfä-higkeit des bisher nicht verwendeten vierten Kanals der CTC ausgenutzt. Auf der CPU-Platte muß dazu der Eingang CTRG3 (Pin 20 der CTC) vom /M1-Signal getrennt und auf eine zusätzliche Taste (gegen Masse) gelegt werden. Dabei muß auf die Verbindung zwi-schen Pin 14 der CTC und Pin 27 der CPU ge-achtet werden. Im EPROM D₆ sind folgende Korrekturen notwendig:

- 0026H: C7 01 Programmierung CTC/
Kanal 3

- 002EH: 9E F3 Anfangsadresse
Interruptroutine
- 0038 H 10 05 DJNZ 003FH
- 003 FH D1 POP DE
- 13 INC DE
- E9 JMP(HL)

Eine eventuelle Verschiebung muß beim Ein-trag in die Interruptadrestabelle berücksich-tigt werden.

Nach Durchführung dieser Änderungen kann mit Hilfe der Taste Pause ein Programm un-terbrochen und damit der Bildschirm ange-halten werden. Nach Betätigen der Space-Taste wird das Programm fortgesetzt.

Basicprogramm zum Verschieben des Maschinenkodes

Mit Hilfe des in Tafel 2 wiedergegebenen klei-nen Basicprogramms läßt sich der in Tafel 1 ab-gedruckte Maschinenkode für andere Spei-cherbereiche lauffähig machen. Es muß betont werden, daß das Programm für den Maschi-nenkode nach Tafel 1 die gewünschten Ergeb-nisse bringt, aber nicht verallgemeinerbar ist. Der Start des Programms erfolgt mit GOTO 10, falls der zu ändernde Maschinenkode vor RAMTOP steht.

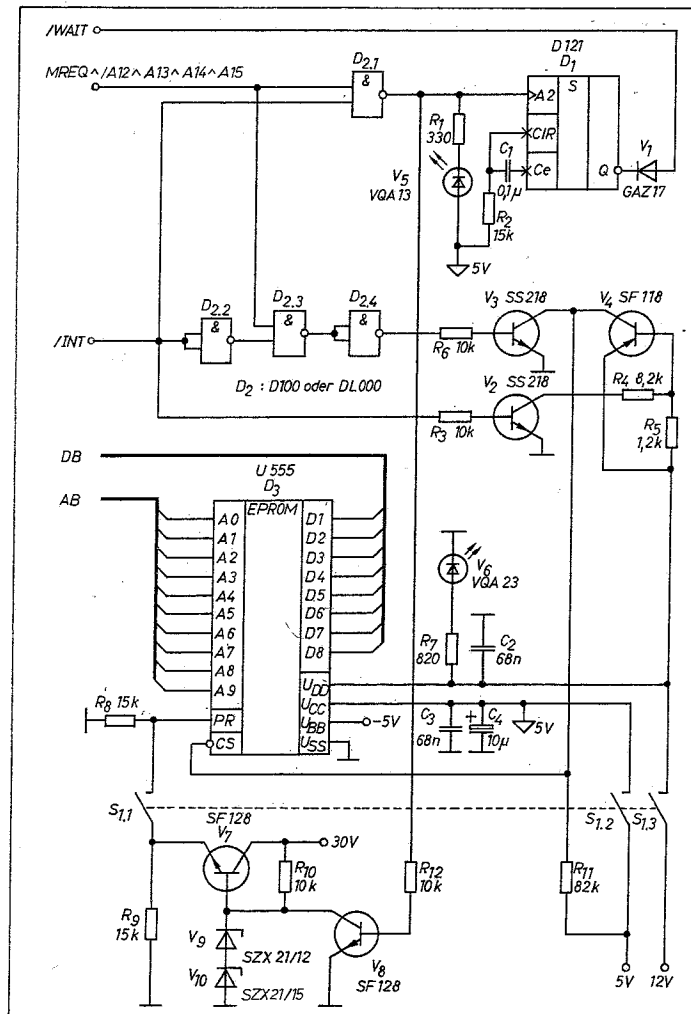
Für ANFANGSADRESSE muß die Adresse eingegeben werden, ab der der in Tafel 1 wie-dergegebene Maschinenkode geladen wurde. Anschließend ist der höherwertige Teil der Zieladresse einzugeben, d. h., der niederwertige Teil bleibt 0H.

Das ab ANFANGSADRESSE stehende Pro-gramm kann dann auf den gewünschten RAM-Bereich übertragen oder in einem EPROM gespeichert werden.

Die mit Hilfe des Pro-gramms aus [2] er-mittelten Prüfsum-men lauten 7CDB, 5C96, 6EE7 und 5CF2.

Literatur

- [1] Schiller, E.: Basic-Heimcomputer. ra-dio fernsehen elek-tronik, Berlin 34 (1985) 1, S. 13–18
- [2] Schiller, E.: Erwei-terungen für Basic-Heimcomputer. ra-dio fernsehen elek-tronik, Berlin 35 (1986) 9, S. 559–563
- [3] Kramer, M.: Prakti-sche Mikrocomputertechnik. Berlin: Militärverlag der DDR 1987



EPROM-
Programmiermodul