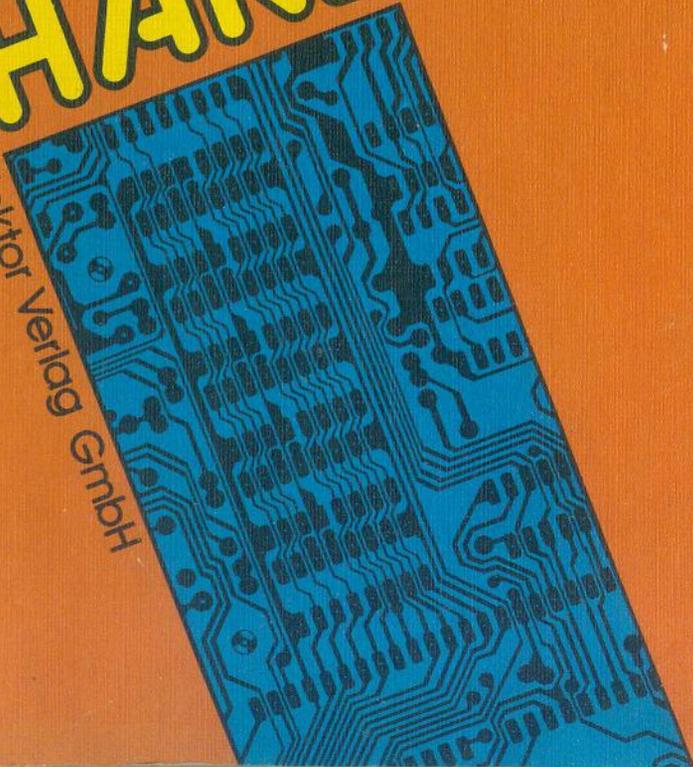


microprocessor

HARDWARE

Elektor Verlag GmbH



microprocessor-HARDWARE - Elektor Verlag GmbH

microprocessor
HARDWARE

G.H.K. Dam

ISBN 3-921608-30-9
Elektor Verlag GmbH, 5133 Gangelt 1

Über die Liefermöglichkeit der im Buch veröffentlichten Platinen gibt die jeweils gültige EPS-Liste (Elektor-Platinen-Service) in der Zeitschrift ELEKTOR Auskunft. Das gilt auch für weitere Informationen über den Elektor-Software-Service (ESS).

© 1983 Elektor Verlag GmbH, 5133 Gangelt 1

Die in diesem Buch veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen und Illustrationen sind urheberrechtlich geschützt. Ihre auch teilweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet. Alleiniges Nachdruckrecht für das holländische Sprachgebiet: Elektuur B.V., Beek (L), Holland; für Publikationen in englischer Sprache: Elektor Publishers Ltd., Canterbury, England; für das französische Sprachgebiet: Elektor sarl, Bailleul, Frankreich. Der Nachbau der veröffentlichten Schaltungen geschieht außerhalb der Verantwortlichkeit des Herausgebers.

Vorwort

In einem Mikrocomputersystem ist beides wichtig: die Hardware und die Software. Das eine ist ohne das andere nicht denkbar. Ein Mikrocomputersystem ist nur dann das oft zitierte und vielbestaunte Wunderwerk der Technik, wenn Hardware und Software optimal aufeinander abgestimmt sind. Die Software ist dabei wesentlich vielfältiger. Sie wird vom Mikrocomputersystem bestimmt und ist deshalb nicht universell für verschiedene Systeme verwendbar.

Anders ist es bei der Hardware. Für den Aufbau der unterschiedlichsten Systeme stehen nur wenige verschiedenartige Bausteine (z. B. CPU's) zur Verfügung. Es ist deshalb möglich, die Hardware so aufzubauen, daß sie mit geringen Änderungen an verschiedene Systeme anzupassen ist. Hierzu gibt das Buch wichtige Tips und Hinweise.

Die im Buch beschriebene Hardware basiert auf Schaltungsentwürfe zu den μ Computersystemen von Elektor. Ohne große Anpassungsschwierigkeiten sind die Schaltungen für Systeme geeignet, die mit den CPU's 6502, 6809, Z80 oder 8080 aufgebaut sind. In einigen Fällen sind kleine Änderungen erforderlich. Da jedoch alle erwähnten Prozessortypen mit dem gleichen Bussystem arbeiten, sind die Änderungen minimal und die beschriebenen Schaltungen universell einsetzbar.

Dekoder

Was ist DUS?

Was ist ein TUP?

Was ist 3k9?

Was ist 4μ7?

Der Elektor-Dekoder erläutert diese und andere häufig in Elektor verwendeten Begriffe, Abkürzungen und Bezeichnungen.

Halbleitertypen

Die Abkürzungen TUP-TUN, DUG-DUS findet man häufig in Elektorschaltungen. Sie beziehen sich auf universell verwendbare Transistoren und Dioden, die hinsichtlich der technischen Daten übereinstimmen und sich nur durch Gehäuseform und Anschlußbelegung unterscheiden. Die Mindestanforderung der TUP-TUN, DUG-DUS-Baulemente sind in den Tabellen I und II zusammengefaßt.

Tabelle I
Mindestanforderungen für
TUP und TUN.

U_{CEO} max.	20 V
I_C max.	100 mA
h_{FE} min.	100
P_{tot} max.	100 mW
f_T min.	100 MHz

TUN-Beispiele:

BC 107 (-8, -9), BC 147 (-8, -9),
BC 207 (-8, -9), BC 237 (-8, -9),
BC 137 (-8, -9), BC 347 (-8, -9),
BC 547 (-8, -9), BC 171 (-2, -3),
BC 182 (-3, -4), BC 382 (-3, -4),
BC 437 (-8, -9), BC 414

TUP-Beispiele:

BC 177 (-8, -9), BC 157 (-8, -9),
BC 204 (-5, -6), BC 307 (-8, -9),
BC 320 (-1, -2), BC 350 (-1, -2),
BC 557 (-8, -9), BC 251 (-2, -3),
BC 212 (-3, -4), BC 512 (-3, -4),
BC 261 (-2, -3), BC 416

Tabelle II
Mindestanforderungen für
DUG und DUS.

	DUG	DUS
U_R max.	20 V	25 V
I_F max.	35 mA	100 mA
I_R max.	100 μ A	1 μ A
P_{tot} max.	250 mW	250 mW
C_D max.	10 pF	5 pF

DUG-Beispiele:

OA 85, OA 91, OA 95, AA 116

DUS-Beispiele:

BA 127, BA 217, BA 317,
BAY 61, 1N914, 1N4148

Viele äquivalente Halbleiter haben unterschiedliche Typennummern. Um Beschaffungsschwierigkeiten eines speziellen Typs zu vermeiden, wird in Elektor – soweit möglich – eine universelle Typennummer angegeben. Als Beispiel möge das IC 741 dienen.

741 bedeutet: μ A 741,
LM 741, MC 741, MIC 741,
RM 741, SN 72741 usw.

Widerstands- und Kapazitätswerte

Die Angabe von Widerstands- und Kapazitätswerten erfolgt ohne Komma. Anstelle des Kommas werden international gebräuchliche Abkürzungen verwendet:

p (Piko)	= 10^{-12}
n (Nano)	= 10^{-9}
μ (Mikro)	= 10^{-6}
m (Milli)	= 10^{-3}
k (Kilo)	= 10^3
M (Mega)	= 10^6
G (Giga)	= 10^9

Einige Beispiele von Widerstands- und Kapazitätsangaben:

3k9 = 3,9 k Ω = 3900 Ω
0 Ω 33 = 0,33 Ω
4p7 = 4,7 pF
5n6 = 5,6 nF
4 μ 7 = 4,7 μ F

Belastbarkeit der Widerstände: $\frac{1}{4}$ Watt (falls nichts anders angegeben).

Die Spannungsfestigkeit der Folienkondensatoren sollte um ca. 20% höher liegen als die Betriebsspannung der Schaltung.

Gleichspannungsangaben

Die in einer Schaltung angegebenen Gleichspannungen sind als Richtwerte zu betrachten, d.h., die Meßwerte dürfen um $\pm 10\%$ abweichen. (Das Meßgerät sollte einen Innenwiderstand von ≥ 20 k Ω /V haben.)

Wichtige Hinweise für den Selbstbauer

1. Verwenden Sie bei selbstgebauten Geräten immer ein Kunststoffgehäuse, damit alle spannungsführenden Bauteile berührungssicher verpackt sind.
2. Wenn aufgrund bestimmter Gegebenheiten ein Metallgehäuse vorgeschrieben ist (z.B. als Abschirmgehäuse bei HF-Schaltungen) muß das Gehäuse in jedem Fall geerdet werden.
3. Alle 220 V-Anschlüsse sowie alle anderen Punkte, an denen die Wechselspannung 42 V oder die Gleichspannung 60 V übersteigt, müssen berührungssicher isoliert sein.
4. Das Netzkabel muß im Gehäuse durch eine Zugentlastung gesichert sein, damit man es nicht von den Transformatoranschlüssen abreißen kann. Ferner darf das Netzkabel nicht einfach durch eine Bohrung in das Gehäuse geführt werden. Damit das Kabel nicht durch die scharfen Kanten der Bohrung verletzt wird, ist die Bohrung unbedingt mit einer Gummitülle zu versehen. Das gilt insbesondere für alle Metallgehäuse.

Inhalt

Vorwort	3
Dekoder	4
Einleitung	6
Kapitel 1: Elekterterminal	
• ASCII-Tastatur plus Video-Interface gleich Schreiben auf dem Bildschirm (mit Beiträgen von T. Frankenmolen, B. Heisterkamp, R. Schneider und K.D. Vonneman)	7
• Mehr Seiten für das Elekterterminal	32
• VHF/UHF-TV-Modulator	38
Kapitel 2: Bus- und μP-Systeme, Speisungen	
• Elektors μ C-Systeme	43
• Bussystem und Busplatine	48
• Speisung für kleine und große μ P-Systeme (+5 V/5 A, -12 V/1 A)	51
• μ C-Schaltzerteil (+5 V/1 A, +12 V/100 mA, -5 V/100 mA)	55
Kapitel 3: Speicher	
• RAM/EPROM-Karte (8-K-RAM + 8-K-EPROM) (mit Beiträgen von A. Seul und G.J.W. Wiselius)	58
• Dynamische RAM-Karte (16 K in 8 ICs) (mit einem Beitrag von D. Paulsen)	70
Kapitel 4: Programmiergeräte und anders	
• 2716-Programmiergerät u.a. für 2650 und 8080 (von R. Péquet)	84
• EPROM-Programmiergerät 2716/32, passend für SC/MP und Junior (von P.R. Boldt)	90
• EPROM-Löcher	97
• PROM-Programmierer	99
• IPROM	104
• Einfache Programmier-Spannungsquelle	108
Kapitel 5: Kassetten-Interface	
• Drei mögliche Lösungen (mit Beiträgen von H.W. Braun und L. van Laren)	110
Kapitel 6: Diverses	
• Kapazitives Tastenfeld	125
• Spannungsausfall-Indikator	130
• Digital-Analog-Wandler (nach einer Idee von T. Basien und P. Habersetzer)	131
• RS-232-Schnittstelle	133
• Einfacher Zufallsgenerator	133
• Programmierbarer Adreßdekoder	136

Einleitung

Das Buch beschreibt Zusatzschaltungen und Erweiterungen für bestehende Mikrocomputersysteme. Das für die beschriebenen Schaltungen verwendete Bussystem läßt sich relativ einfach mit dem Systembus kombinieren. Dazu werden lediglich die gleichnamigen Anschlüsse miteinander verbunden. Ebenfalls berücksichtigt sind in diesem Buch die mittlerweile genormten Anschlußbedingungen der EPROM-Serie 27XX. Zwei Schaltungen können als "plug-in-System" in bereits vorhandene EPROM-Fassungen gesteckt werden, einmal als EPROM-Programmierer und zum anderen als Pseudo-ROM.

In der Regel sind die gängigen Personalcomputer mit einem eigenen Bildschirm ausgerüstet oder wenigstens mit der Möglichkeit versehen, einen Monitor anzuschließen. In vielen Fällen, insbesondere wenn man die Information einem größeren Kreis zugänglich machen will, ist dies nicht ausreichend. Hier würde ein Videoausgang Abhilfe schaffen.

Sollen außerdem noch mehrere Anwender gleichzeitig mit dem System arbeiten, sind mehrere Zusatzterminals erforderlich. Die im Buch angebotene Lösung bietet zu den kommerziellen Geräten eine gute Alternative: Das Elekterterminal arbeitet mit einem normalen Fernseher.

Die Software ersetzt in vielen Fällen die notwendige Hardware. So lassen sich z. B. Gatterfunktionen oder Zähler softwaremäßig nachvollziehen. Nicht immer bietet jedoch die Software-Lösung eine zufriedenstellende Lösung, so z. B. bei einem Software-Zufallsgenerator. Wo also die Software nicht ausreicht, steht immer noch die Hardware zur Verfügung. In Kapitel 6 sind zu diversen Problemen Hardware-Lösungen angegeben. Beispielsweise die Schaltung, die anzeigt, wenn die Versorgungsspannung kurzzeitig ausgefallen ist.

Welches Computerprogramm ist dazu schon in der Lage. Für die Hardware ist dies kein Problem. Die Schaltungen in diesem Buch beweisen es.

G.H.K. Dam

Kapitel 1:

Elekterterminal

ASCII-Tastatur plus Video-Interface gleich Schreiben auf dem Bildschirm

Das "Tüpfelchen auf dem i" ist für ein Mikrocomputersystem das Video-Interface, mit dem sich Informationen auf elegante Weise sichtbar machen lassen. Wird das Video-Interface von einem alphanumerischen Keyboard (Tastatur) ergänzt, so entsteht eine vollständige Ein- und Ausgabe-Einheit.

Wenn ein solches Gerät unabhängig vom Mikrocomputer arbeitet (z.B. als TV-Schreibmaschine), spricht man nicht mehr von einem Video-Interface, sondern nennt die gesamte Einheit Video-Terminal oder VDU (Video Display Unit). Mit den auf dem Markt erhältlichen "CRT-Controllern" läßt sich ein komplettes Video-Terminal aufbauen, ohne daß hierfür eine Unzahl an TTL-ICs und teuren Platinen notwendig ist. Das Elekterterminal kann zusammen mit einem TV-Empfänger entweder die Funktion eines Video-Interface (ohne Keyboard) oder die eines Video-Terminals übernehmen. Die gesamte Elektronik des Videoteils nimmt hierbei nicht mehr Raum als eine Eurokarte ein.

Für die Verständigung mit einem (Mikro-)Computer wird in der Regel eine alphanumerische Tastatur als Input Unit benutzt. Das hier beschriebene Keyboard ist der erste Baustein für das "Elekterterminal".

Der zweite Baustein ist das Video-Interface. Beide Einheiten zusammen lassen sich an einen Personalcomputer anschließen. Der Datenaustausch zwischen Computer und Elekterterminal geschieht seriell.

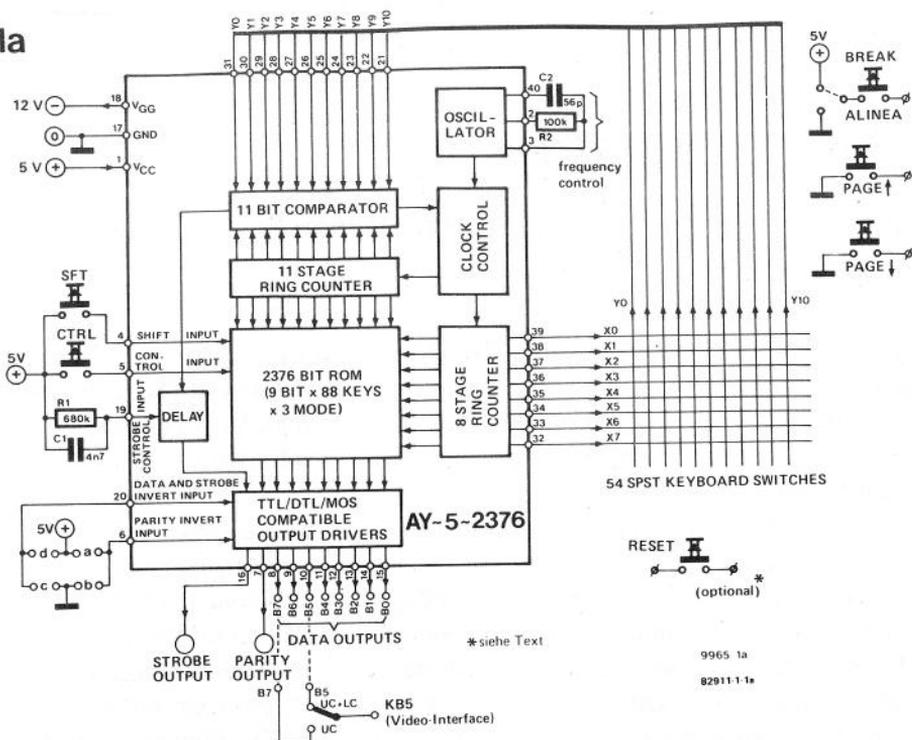
Ein alphanumerisches Keyboard ist eine Tastatur, die ähnlich wie bei einer Schreibmaschine Tasten für die Buchstaben des lateinischen Alphabets, die Dezimalziffern und die Satzzeichen und Hilfsfunktionen besitzt. Damit der Computer "verstehen", welches Zeichen (auch Charakter genannt) jeweils gemeint ist, muß ihm dies in Form unterschiedlicher Bitmuster mitgeteilt werden.

Im Laufe der Jahre wurden für diesen Zweck verschiedene Codes entwickelt;

der gebräuchlichste von ihnen ist der sogenannte ASCII-Code (ASCII = American Standard Code for Information Interchange).

Das ASCII-Code ist ein 8 Bit-Code, der das Most Significant Bit (MSB) als Paritätsbit benutzt. Die übrigen 7 Bit drücken das Zeichen oder die Hilfsfunktion aus. Die Anzahl der möglichen Bitmuster (Bytes) ist bei einem 7 Bit-Wort derart groß ($2^7 = 128$), daß neben den Bytes für Dezimalziffern, Buchstaben und

1a



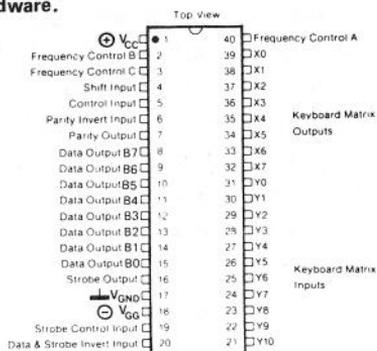
Satzzeichen noch eine stattliche Anzahl frei bleibt. Beim ASCII-Code sind diese Bytes für System-Control-Funktionen reserviert. In Tabelle 1 ist der vollständige ASCII-Code zusammen mit den Bezeichnungen der System-Control-Funktionen angegeben.

Keyboard-Schaltung

Im Prinzip läßt sich ohne weiteres ein Keyboard mit separaten Tasten für sämtliche der 128 möglichen Bytes aufbauen, die der ASCII-Code enthält. Ein solches Keyboard wäre jedoch nicht nur kostspielig, sondern auch recht unhandlich in der Bedienung. Die meisten Tasten werden deshalb mit zwei Funktionen belegt, so daß die Anzahl der benötigten Tasten beträchtlich schrumpft. Allerdings ist eine zusätzliche Taste notwendig, um zwischen den beiden Funktionen der übrigen Tasten umschalten zu können. Die Schaltung des Keyboards muß jeweils

Bild 1. Das Charakter-Generator-IC AY-5-2376 ist das Herzstück der Keyboard-Hardware.

1b



das Bitmuster erzeugen, das der ASCII-Code für das einzelne Zeichen vorsieht. Es verwundert kaum, daß solche "Keyboard Encoder" schon seit geraumer Zeit in IC-Form zur Verfügung stehen. Ein Beispiel ist das AY-5-2376 von General Instruments. Bild 1 zeigt blockschematisch die interne Schaltung und die Anschlußdaten dieses ICs, das zusammen mit

Tabelle 1a.

Character	Binary Bit 7 to Bit 0	Hexadecimal	Character	Binary Bit 7 to Bit 0	Hexadecimal
NUL	00000000	00	@	01000000	40
SOH	00000001	01	A	01000001	41
STX	00000010	02	B	01000010	42
ETX	00000011	03	C	01000011	43
EOT	00000100	04	D	01000100	44
ENQ	00000101	05	E	01000101	45
ACK	00000110	06	F	01000110	46
BEL	00000111	07	G	01000111	47
BS	00001000	08	H	01001000	48
HT	00001001	09	I	01001001	49
LF	00001010	0A	J	01001010	4A
VT	00001011	0B	K	01001011	4B
FF	00001100	0C	L	01001100	4C
CR	00001101	0D	M	01001101	4D
SO	00001110	0E	N	01001110	4E
SI	00001111	0F	O	01001111	4F
DLE	00010000	10	P	01010000	50
DC1	00010001	11	Q	01010001	51
DC2	00010010	12	R	01010010	52
DC3	00010011	13	S	01010011	53
DC4	00010100	14	T	01010100	54
NAK	00010101	15	U	01010101	55
SYN	00010110	16	V	01010110	56
ETB	00010111	17	W	01010111	57
CAN	00011000	18	X	01011000	58
EM	00011001	19	Y	01011001	59
SUB	00011010	1A	Z	01011010	5A
ESC	00011011	1B	[01011011	5B
FS	00011100	1C	\	01011100	5C
GS	00011101	1D]	01011101	5D
RS	00011110	1E	^	01011110	5E
US	00011111	1F	_	01011111	5F
SP	00100000	20	`	01100000	60
!	00100001	21	a	01100001	61
"	00100010	22	b	01100010	62
#	00100011	23	c	01100011	63
\$	00100100	24	d	01100100	64
%	00100101	25	e	01100101	65
&	00100110	26	f	01100110	66
'	00100111	27	g	01100111	67
(00101000	28	h	01101000	68
)	00101001	29	i	01101001	69
*	00101010	2A	j	01101010	6A
+	00101011	2B	k	01101011	6B
,	00101100	2C	l	01101100	6C
-	00101101	2D	m	01101101	6D
.	00101110	2E	n	01101110	6E
/	00101111	2F	o	01101111	6F
0	00110000	30	p	01110000	70
1	00110001	31	q	01110001	71
2	00110010	32	r	01110010	72
3	00110011	33	s	01110011	73
4	00110100	34	t	01110100	74
5	00110101	35	u	01110101	75
6	00110110	36	v	01110110	76
7	00110111	37	w	01110111	77
8	00111000	38	x	01111000	78
9	00111001	39	y	01111001	79
:	00111010	3A	z	01111010	7A
;	00111011	3B	{	01111011	7B
<	00111100	3C		01111100	7C
=	00111101	3D	}	01111101	7D
>	00111110	3E	~	01111110	7E
?	00111111	3F	DEL	01111111	7F

der Tastatur bereits das komplette Keyboard bildet. Ein Charakter-Generator wie das AY-5-2376 ist nichts anderes als ein ROM, in dem der ASCII-Code gespeichert ist. Die Adressierung des Speichers geschieht durch Drücken einer Taste. Die ebenfalls im IC integrierte Adreßlogik sorgt dafür, daß das richtige Byte an den Ausgängen erscheint. Die Verdrahtung des Tastensatzes vereinfacht sich, wenn die Tasten in Matrix-Form angeordnet werden. In Bild 2 ist angegeben, an welchen Matrixpunkten sich die einzelnen Tasten befinden. Außer den Matrix-Tasten sind in der Schaltung (Bild 1) noch sechs weitere Tasten eingezeichnet: eine Break-, zwei Page-, eine

Reset-, eine Shift- und eine Control-Taste. Die Break- und Page-Tasten sind für das Elekterterminal bestimmt, die Reset-Taste kann für eigene Zwecke benutzt werden. Die Funktionen der Matrix-Tasten hängen von der Shift- und der Control-Taste ab. Der genaue Zusammenhang geht aus Tabelle 2 hervor. Die Zeichen in den Zeilen N (Normal) geben an, welches Zeichen das IC bei Drücken dieser Taste ohne zusätzliche Umschaltfunktion erzeugt. In den Zeilen S und C sind die Zeichen aufgeführt, die bei gleichzeitigem Drücken der Taste S (Shift) bzw. C (Control) gelten. Viele Zeichen kommen in Tabelle 2 mehrfach vor; an einigen Matrixpunkten ist deshalb keine Taste erforderlich.

Verschiedene Charakter des ASCII-Code haben für das Elekterterminal eine abweichende Funktion erhalten. Tabelle 3 gibt an, um welche Charakter es sich im einzelnen handelt und welches ihre Funktion beim Elekterterminal ist. Bei anderer Verwendung des Keyboards kann natürlich die Originalfunktion beibehalten werden.

Mechanische Schalter sind nicht frei von Kontaktprellen. Um Fehlfunktionen auszuschließen, ist im IC eine Entprellschaltung integriert, deren Verzögerungszeit von den externen Komponenten R1 und C1 abhängt. Ferner können die IC-Anschlüsse 6 und 20 über die Drahtbrücke a, b, c und d entweder an Masse oder an die positive Speisespannung gelegt werden. Liegen sie an "1" (+5 V), dann sind die Signale an den Data-Ausgängen, am Strobe-Ausgang und am Paritätsausgang invertiert. Normalerweise liegen die Anschlüsse 6 und 20 an Masse ("0"), es sind dann nur die Brücken c und b vorhanden.

nur GROSSBUCHSTABEN

Großbuchstaben unterscheiden sich von Kleinbuchstaben im ASCII-Code nur durch den komplementären Wert des sechsten Bits (B5 am Data-Ausgang des Keyboard-Encoders). Bei Großbuchstaben hat dieses Bit den Wert 0, bei Kleinbuchstaben demnach den Wert 1

Tabelle 1b.

NUL	— null, or all zeros
SOH	— start of heading
STX	— start of text
ETX	— end of text
EOT	— end of transmission
ENQ	— enquiry
ACK	— acknowledge
BEL	— bell
BS	— backspace
HT	— horizontal tabulation
LF	— line feed
VT	— vertical tabulation
FF	— form feed
CR	— carriage return
SO	— shift out
SI	— shift in
DLE	— data link escape
DC1	— device control 1
DC2	— device control 2
DC3	— device control 3
DC4	— device control 4
NAK	— negative acknowledge
SYN	— synchronous idle
ETB	— end of transmission block
CAN	— cancel
EM	— end of medium
SUB	— substitute
ESC	— escape
FS	— file separator
GS	— group separator
RS	— record separator
US	— unit separator
SP	— space
DEL	— delete

Tabelle 1. In dieser Tabelle ist der vollständige ASCII-Code sowohl binär als auch in hexadezimaler Schreibweise angegeben.

	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	
X0												
X1												
X2	/	= -	FS						SP			
X3	/	0	*	P	@	← BS	{ [}]	CR	LF ↓		
X4	/	+ ;	? /	> .	< ,	M	N	B	V	C	X	Z
X5	/	L	K	J	H	G	F	D	S	A	Erase FF	ESC
X6	/	O	I	U	Y	T	R	E	W	Q	→ HT	VT ↑
X7	/) 9	(8	' 7	& 6	% 5	\$ 4	# 3	" 2	! 1		

9965 2
82911-2

(siehe Tabelle 1a).

Der Keyboard-Encoder AY-5-2376 hat außer den üblichen sieben ASCII-Bits einen Ausgang für ein achttes Bit (B7). Dieses achte Bit kann die Funktion des sechsten Bits übernehmen, obwohl dieser Sachverhalt im Datenblatt des Encoders nicht besonders hervorgehoben wurde. In verschiedenen Fällen vereinfacht der Gebrauch des Upper-Case-Ausgangs die Bedienung der Tastatur erheblich. Ein gutes Beispiel ist das Programmieren in NIBL-BASIC. Diese BASIC-Programmiersprache verarbeitet ausschließlich Großbuchstaben.

Die Verwendung des Upper-Case-Ausgangs vermeidet das ständige Betätigen der SHIFT-Taste, die Taste wird jetzt nur noch für die Eingabe spezieller Zeichen benötigt: der Bedienkomfort hat erheblich zugenommen. Eventuell kann man mit einem einpoligen Umschalter eine Wahlmöglichkeit für Groß- und Kleinbuchstaben sowie für Großbuchstaben allein schaffen (siehe Bild 1a).

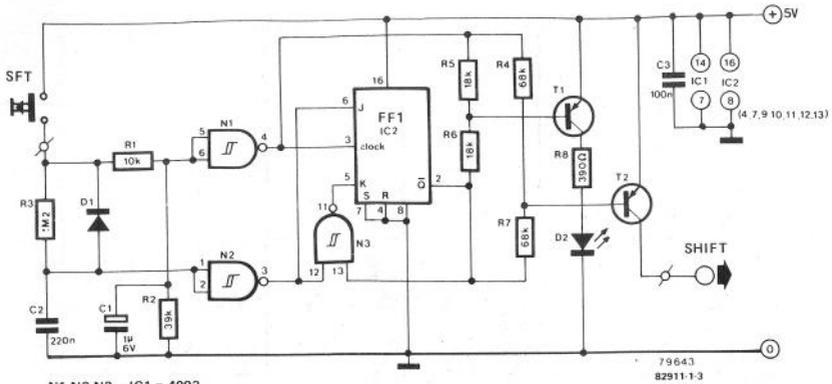
SHIFT-LOCK

Man kann den Anwendungsbereich der

Bild 2. Die Matrixpunkte, an denen sich eine Taste befindet, gehen aus diesem Bild hervor.

ASCII-Tastatur bedeutend erweitern, wenn man eine zusätzliche "SHIFT-LOCK"-Schaltung einbaut (Bild 3). Auch mit anderen Tastaturen kann man diese Schaltung einsetzen. Für diese Erweiterung muß keine zusätzliche Taste eingebaut werden, da die vorhandenen SHIFT-Tasten sowohl die normale SHIFT-als auch die SHIFT-LOCK-Funktion übernehmen. Maßgebend ist dabei, wie lange die Taste eingedrückt bleibt. Bei einem kurzen ($\leq 0,2$ s) Druck auf eine der SHIFT-Tasten bleibt der SHIFT-Ausgang der Schaltung aktiv, so daß durch einen zweiten Tastendruck die Funktion "SHIFT-LOCK" abgeschaltet wird. Bleibt eine SHIFT-Taste länger als 0,2 s gedrückt, so kehrt der SHIFT-Ausgang der Schaltung direkt in den Ruhezustand zurück.

Als Zeiteinheit dient die Zeitkonstante des RC-Gliedes R3/C2. Der Schmitt-Trigger kippt um, wenn die Spannung an C2 etwa 45% der Versorgungsspannung beträgt. Um die Arbeitsweise der Schaltung zu erklären, geht man am



N1,N2,N3 = IC1 = 4093

FF1 - 1/2 IC2 - 4027

D1 = DUS

T1, T2 = TUP

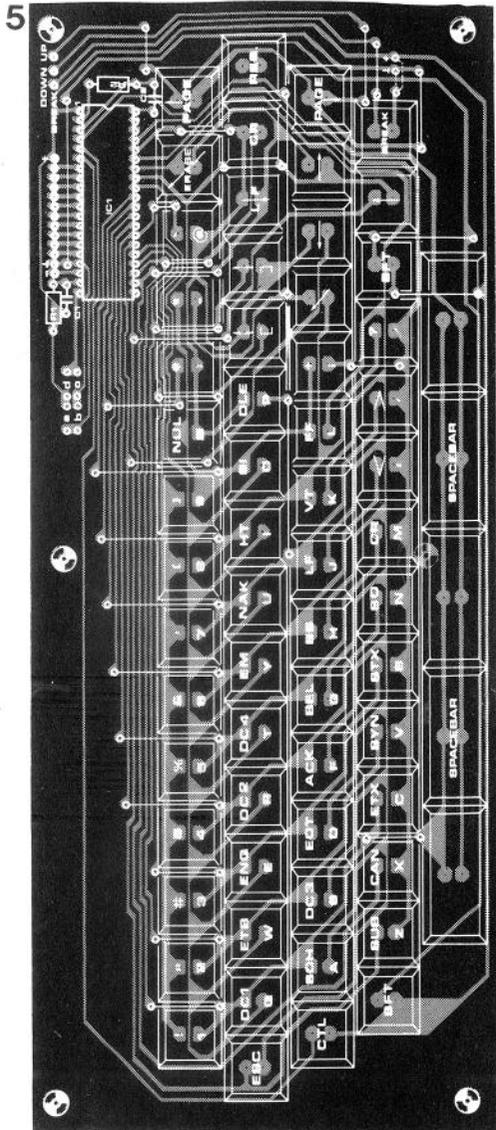
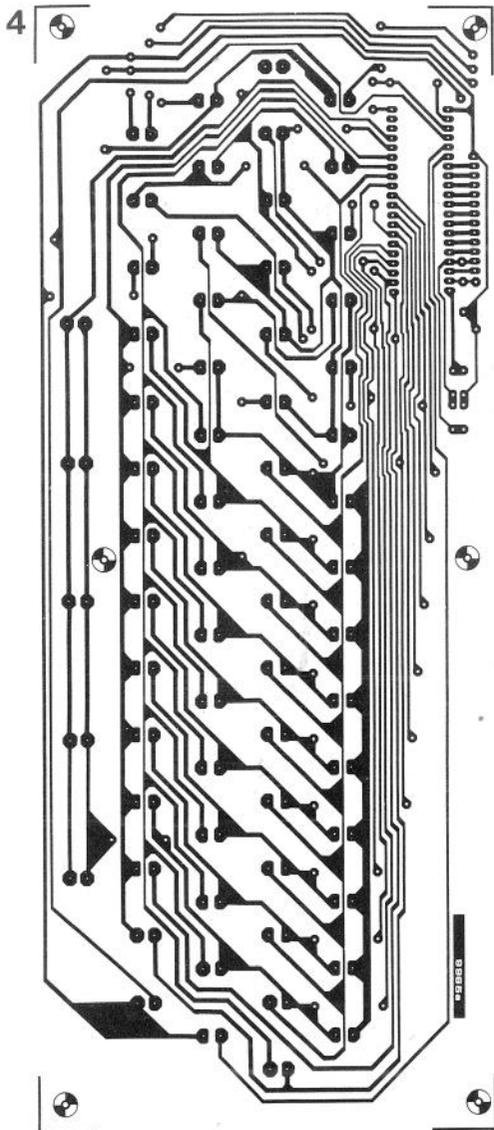
Bild 3. Die Shift-Lock-Schaltung.

Tabelle 2. Die Tabelle gibt Aufschluß über den Zusammenhang zwischen Keyboard-Matrix, Umschaltfunktionen und Charakter. Verschiedene Zeichen kommen mehrfach vor, so daß nicht für jeden Matrixpunkt eine eigene Taste benötigt wird.

Tabelle 2.

		y 0	y 1	y 2	y 3	y 4	y 5	y 6	y 7	y 8	y 9	y 10
x 0	C	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	DC1	DLE	SI
	S	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	DC1	@	←
	N	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	DC1	P	O
x 1	C	DLE	VT	FF	SO	CR	NAK	SYN	ETB	CAN	EM	SUB
	S	DLE		\	↑]	NAK	SYN	ETB	CAN	EM	SUB
	N	DLE	K	L	N	M	NAK	SYN	ETB	CAN	EM	SUB
x 2	C		FS	GS	RS	US				SP		US
	S	=	FS	GS	RS	US	<	>	,	SP	.	←
	N	-	FS	GS	RS	US	<	>	,	SP	.	←
x 3	C			DLE	US		BS	ESC	GS	CR	LF	RUB
	S		*	P	DEL		BS	{	}	CR	LF	OUT
	N	∅	:	p	←	@	BS	[]	CR	LF	
x 4	C					CR	SO	STX	SYN	ETX	CAN	SUB
	S	+	?	>	<	M	N	B	V	C	X	Z
	N	;	/	.	,	m	n	b	v	c	x	z
x 5	C	FF	VT	LF	BS	BEL	ACK	EOT	DC3	SOH	FF	ESC
	S	L	K	J	H	G	F	D	S	A	FF	ESC
	N	l	k	j	h	g	f	d	s	a	FF	ESC
x 6	C	SI	HT	NAK	EM	DC4	DC2	ENQ	ETB	DC1	HT	VT
	S	O	l	U	Y	T	R	E	W	Q	HT	VT
	N	o	i	u	y	t	r	e	w	q	HT	VT
x 7	C										RS	FS
	S)	('	&	%	\$	=	"	!	ESC	
	N	9	8	7	6	5	4	3	2	1	↑	\

besten von dem Zustand aus, in dem \bar{Q} logisch 1 ist. Drückt man die SFT-Taste kurz ein, so bleibt der Ausgang von N2 auf logisch 1. Entsprechend ist der J-Eingang des Flipflops "1" und der K-Eingang "0". Kurze Zeit nach dem Eindrücken wird der Ausgang von N1 logisch 0 und damit der SHIFT-Ausgang aktiv. Sobald man den Knopf losläßt, bekommt das Flipflop einen Triggerimpuls und der \bar{Q} -Ausgang wird "0".



Infolgedessen bleibt der SHIFT-Ausgang aktiv und am K-Eingang des Flipflops erscheint eine "1". Beim folgenden kurzen Druck auf die SHIFT-Taste kippt das Flipflop wieder zurück, und die SHIFT-Funktion ist abgeschaltet. Drückt man die Taste länger als 0,2 s, so sorgt N2 dafür, daß der \bar{Q} -Ausgang nicht "0" werden kann. Entsprechend beendet ein Loslassen der Taste die SHIFT-Funktion sofort. R1, C1 und C2 unterdrücken Prell-

Bild 4. Kupferseite der Keyboard-Platine.

Bild 5. Bestückungsseite der Keyboard-Platine (Bezugsquellen für die Taster teilt die Redaktion auf Anfrage mit).

Layout und Bestückungsseite sind um 50% verkleinert abgebildet.

impulse von der SHIFT-Taste. Die LED D2 zeigt an, ob die SHIFT-LOCK-Funktion aktiv ist.

Stückliste zu Bild 1a

Widerstände:

R1 = 680 k

R2 = 100 k

Kondensatoren:

C1 = 4n7

C2 = 56 p

Halbleiter:

IC1 = AY-5-2376

(General Instruments)

Sonstiges:

62 Tasten TKC Typ MM9

Auto-Repeat

Ein weiteres "Extra" macht den Anwendungsbereich der ASCII-Tastatur noch vielseitiger: die Dauerfunktion verschiedener Tasten. Ähnlich wie bei einer elektrischen Schreibmaschine ein bestimmter Buchstabe dauernd angeschlagen wird, so lange die entsprechende Taste gedrückt

ist, wiederholt auch die ASCII-Tastatur den Kode, wenn die zugehörige Taste längere Zeit gedrückt bleibt. Für diese Zusatzfunktion genügt lediglich ein geringer Eingriff ins Video-Interface, dessen Beschreibung folgt.

Platine und Bau

Damit der Bau des Keyboards sich so einfach wie möglich gestaltet, wurde hierfür eine Platine entworfen. Die Platine, deren Kupfer- und Bestückungsseite die Bilder 4 und 5 zeigen, nimmt die gesamte Hardware einschließlich der Tasten auf. Vorgehen ist die Platine für Tasten des Typs MM9-2 von TKC (Tokai Communication). Bevor man die Tastaturplatine bestellt, sollte man sich zuerst um die Tasten bemühen. Bild 6 gibt an, welche Taste an welcher Stelle sitzt. Die Montage der Tasten erfordert erhöhte Sorgfalt. Da die Tasten durch ihre elektrischen Anschlüsse an ihrem Platz gehalten werden, ist beim Einlöten darauf

Bild 6. So sind die einzelnen Zeichen und Hilfsfunktionen auf der Tastatur angeordnet.

Tabelle 3. Einige Charakter werden beim Elekterminal für andere Zwecke benutzt; sie sind in dieser Tabelle aufgeführt.

Tabelle 3.

CTL	+	L	=	FF	(FORM FEED)	=	home cursor + page clear
CTL	+	J	=	LF	(LINE FEED)	=	LF + cursor ↓
CTL	+	I	=	HT	(HORIZONTAL TAB)	=	cursor →
CTL	+	K	=	VT	(VERTICAL TAB)	=	cursor ↑
CTL	+	M	=	CR	(CARRIAGE RETURN)	=	CR + erasure to end of line
				FS	(FILE SEPARATOR)	=	home cursor ↖
CTL	+	⌘	=	ESC	(ESCAPE)	=	scroll up
CTL	+	⌛				=	CR (no erasure)
CTL	+	Z				=	erasure of current line

6



82911-1-6
9965-5

zu achten, daß sie alle genau senkrecht und in gleicher Höhe stehen. Es kann sonst leicht vorkommen, daß die eine oder andere Taste nach dem Drücken hängenbleibt.

Für die Verbindung zwischen Keyboard und Interface ist Flachkabel das geeignete Material; hierüber kann das Keyboard gleichzeitig mit Strom versorgt werden. Das Keyboard benötigt die beiden Speisespannungen +5 V und -12 V; die Stromaufnahme beträgt bei beiden Spannungen maximal 10 mA. Die Tastaturplatine ist nicht für die beschriebene Shift-Lock-Schaltung vorgesehen. Der Anschluß dieser Schaltung an die ASCII-Tastatur ist jedoch sehr einfach: Leiterbahn an Pin 4 des AY-5-2376 unterbrechen und zwischen Taste und IC die Shift-Lock-Schaltung anschließen.

Video-Interface

Ein Video-Interface wird in einem Mikro-Computersystem in der Regel direkt adressiert. Die Information kann also sehr schnell zum Bildschirm gelangen. Das nachfolgend beschriebene Interface arbeitet jedoch mit der seriellen Kommunikation. Dies hat den wesentlichen Vorteil, daß sich das Interface mit jedem Mikrocomputer kombinieren läßt, der einen seriellen Ausgang besitzt. Die meisten Mikrocomputer verfügen bereits über eine serielle Ausgaberroutine, so daß auch ein großer Teil der für das Schreiben auf dem Bildschirm notwendigen Software vorhanden ist. Den Benutzer des Terminals werden folgende wichtigen Eigenschaften am meisten interessieren:

- Die Speicherkapazität beträgt 1024 Zeichen (Character), die sich auf 16 Zeilen zu 64 Zeichen (=1 Seite) verteilen.
- Eine Erweiterung der Speicherkapazität auf mehrere Seiten ist möglich.
- Sechs verschiedene Schreibgeschwindigkeiten stehen zur Wahl: 75, 110, 150, 300, 600 und 1200 Baud.
- Das Interface verarbeitet 6- oder 7-Bit-ASCII-Code mit und ohne Paritätsbit und mit einem oder zwei Stopbits.

- Der Text wird weiß auf schwarzem Untergrund oder schwarz auf weißem Untergrund geschrieben.
- Der Benutzer hat durch einfache Cursorcontrol-Software uneingeschränkte Bewegungsfreiheit auf dem Schirm.

Alle genannten Eigenschaften gelten auch für den Einsatz als Terminal; hinzu kommt noch der Vorteil, daß sämtliche Informationen mit dem Keyboard eingegeben werden können, so daß für die Benutzung des Bildschirms keinerlei Software erforderlich ist. Dies eröffnet sogar die Möglichkeit, mit einem anderen Terminal über eine Fernsprechleitung zu kommunizieren. Auch ist der Einsatz des Bildschirms als Informationsgeber für Reklamezwecke, als Fahrplanauskunft usw. denkbar. Für die genannten Zwecke muß das Terminal lediglich mit einem Kassetten-Interface erweitert werden.

Blockschema

Da die Tastatur bereits beschrieben wurde, kann sich dieser Beitrag ganz dem Video-Interface widmen (Bild 7). Direkt mit dem Keyboard verbunden ist der UART. Diese Abkürzung bedeutet "Universal Asynchronous Receiver/Transmitter" oder wörtlich übertragen "Universeller asynchroner Sendeempfänger". Dieser Baustein (nur ein einziges LSI-IC) ist für den Informationsaustausch mit dem Keyboard, dem Videodisplay und der übrigen Außenwelt zuständig. Letzteres kann zum Beispiel ein Mikrocomputer sein. Der UART erfüllt eine Vielzahl wichtiger Funktionen; er wird nachfolgend noch ausführlicher besprochen.

Für den Informationsaustausch in einer bestimmten Geschwindigkeit benötigt der UART die Hilfe eines sogenannten Baudrate-Generators. Dieser erzeugt Signale mit Frequenzen, die um das 16fache höher als die gewünschte Baudrate (= Übertragungsgeschwindigkeit in Bit/s) liegen. Hier werden diese Signale durch Teilen der quartzesteuerten Oszillatorfrequenz des CRTC (Cathode Ray Tube Controller) erzeugt. Der CRTC darf

mit Recht als der wichtigste Baustein des Interface betrachtet werden. Er ist nämlich für die horizontale und vertikale Synchronisation im Videosignal, das Adressieren des Seitenspeichers und die Steuerung des Zeichengenerators zuständig; der CRTC wird deshalb ebenfalls noch Gegenstand ausführlicher Betrachtungen sein.

Der Seitenspeicher besteht aus einer Anzahl statischer RAMs. Während der Dauer eines Bilds (20 ms) wird dieser Speicher einmal vollständig ausgelesen. Der Zeichengenerator "übersetzt" die gespeicherten Informationen in für das Videodisplay verständliche Parallel-Informationen. Nach der Umsetzung in ein serielles Signal fügt der "Video-Combiner" das Video- und das Synchronisationssignal zusammen. Dieser liefert bereits das Videosignal ($5V_{SS}$ an 68Ω), das über den Video-Eingang oder den VHF-UHF-TV-Modulator einem gewöhnlichen TV-Empfänger zugeführt werden kann.

Der einzige noch verbleibende Block ist der CTL-Dekoder (CTL = Control). Dieses ROM setzt das vom UART kommende ASCII-Signal in Kommandos für den CRTC um, aus denen dieser erkennen kann, ob es sich jeweils um ein Steuersignal oder ein abzubildendes Zeichen handelt.

Spezielle Baueinheiten

In diesem Teil des Artikels werden UART, CRTC und Zeichen-(Character-)Generator ausführlich besprochen.

UART

Der Universelle Asynchrone Daten-Sendeempfänger besteht aus zwei in einem 40-poligen DIL-Gehäuse untergebrachten Chips. Ein Teil der Funktionen ist so kombiniert, daß die Anzahl der verfügbaren Pins nicht überschritten wird (siehe Bild 8). Der UART ist eine für die Datenübertragung häufig benutzte Einheit, da er selbständig digitale Informationen senden und empfangen kann. Die Sende- und Empfangsgeschwindigkeit hängt von Taktsignalen ab, die das

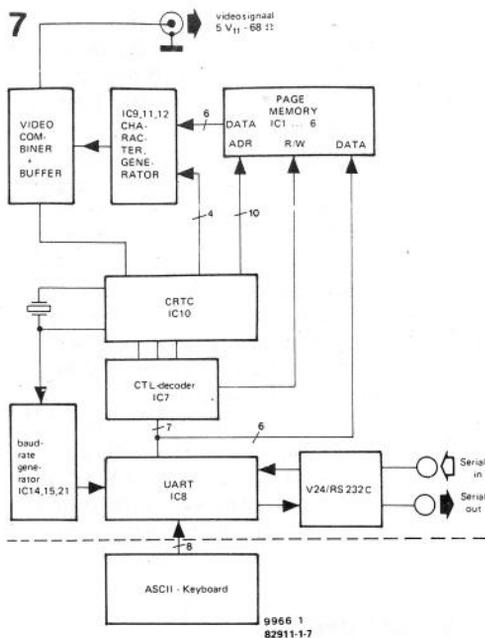
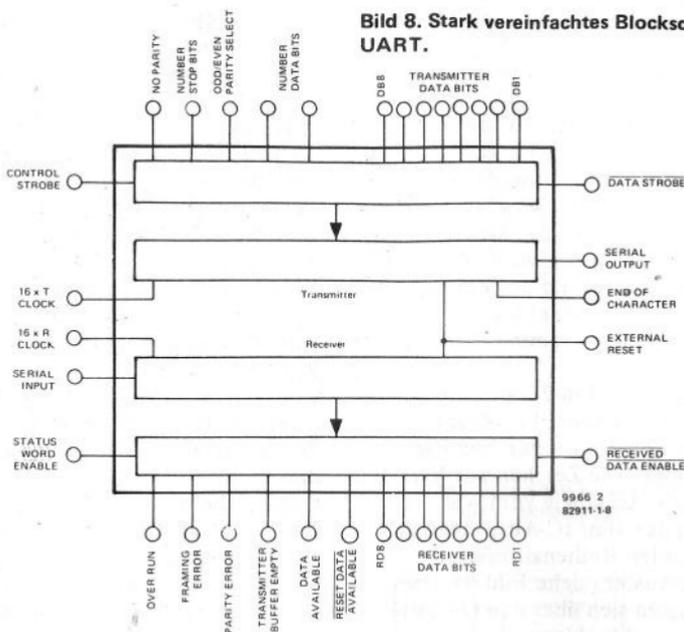


Bild 7. Blockscheema des kompletten Elekt terminals.

16fache der Baudrate beträgt. Da für Sender und Empfänger getrennte Takteingänge vorhanden sind, kann mit unterschiedlichen Sende- und Empfangsgeschwindigkeiten gearbeitet werden. Der UART eignet sich daher auch als Baudrate-Konverter.

Der Sendeseite des UART müssen Parallel-Informationen zugeführt werden, die zum Beispiel von einem ASCII-Key-board stammen. Diese Informationen werden mit Start- und Stopbits und eventuell einem Paritätsbit versehen. Der Benutzer hat die Wahl zwischen einem oder zwei Stopbits, Odd- oder Even-Parität und 5, 6, 7 oder 8 Bit langen Daten-Bytes. Die Einstellungen erfolgen für Sender und Empfänger gemeinsam. Nachdem der UART das Daten-Byte auf diese Weise ergänzt hat, wird die Information in serieller Form mit einer vom Taktsignal abhängigen Geschwindigkeit ausgesendet. Auf der Empfangsseite des UART findet der umgekehrte Vorgang statt. Die seriell empfangenen Informationen



werden von den Start- und Stopbits befreit, die Parität wird geprüft und eventuelle Fehler über den Parity-Error-Ausgang angezeigt. Schließlich kommt die Information an den Parallelausgängen in ihrer ursprünglichen Form wieder zum Vorschein.

Zeichengenerator

Weniger komplex als der UART, jedoch ebenso wichtig wie dieser ist der Zeichen-(Character-)Generator. Dieses IC ist nämlich für die Form verantwortlich, in der Buchstaben, Ziffern und andere Zeichen auf dem Bildschirm erscheinen. Im allgemeinen werden die Zeichen aus den Elementen einer Punktmatrix zusammengesetzt. Die gebräuchlichsten Arten sind die 5 x 7 und 7 x 9-Matrix. Beide Matrizen haben Vor- und Nachteile. Die 7 x 9-Matrix ermöglicht wegen ihrer höheren Punktzahl grundsätzlich einen feiner gerasterten Bildaufbau und daher eine bessere Lesbarkeit als die 5 x 7-Matrix. In der Praxis ist dies jedoch nur bei Einsatz von Video-Monitoren, nicht jedoch bei Verwendung gewöhnlicher TV-Empfänger zutreffend. Die höhere Punktzahl

der 7 x 9-Matrix hat nämlich eine größere Videobandbreite zur Folge. Bei den 64 Zeichen, die pro Zeile geschrieben werden, ist die Videobandbreite um einige Megahertz größer als die Empfängerbandbreite, was natürlich auf Kosten der Bildschärfe geht. Die 5 x 7-Matrix liefert zwar einfacher aufgebaute Zeichen, die Lesbarkeit kann bei 64 Zeichen pro Zeile und Verwendung eines gewöhnlichen TV-Empfängers mit Video-Eingang trotzdem als gut bezeichnet werden. Auch bei Zwischenschalten eines VHF/UHF-Modulators bleibt die Lesbarkeit durchaus akzeptabel. Im Elekterterminal ist deshalb ein

9

ROW ADDRESS	OUTPUTS
A ₃ A ₂ A ₁	O ₅ O ₄ O ₃ O ₂ O ₁
0 0 0	0 0 0 0 0
0 0 1	0 0 0 0 0
0 1 0	1 0 0 0 0
0 1 1	1 0 0 0 0
1 0 0	0 1 0 0 0
1 0 1	0 1 0 0 0
1 1 0	0 0 0 0 1
1 1 1	0 0 0 0 1

9966 3
82911-1-9

Bild 9. Die Zeichenmatrix besteht aus 5 Spalten mit einer Höhe von 8 Punkten. Die obere Reihe bleibt stets leer.

Zeichengenerator mit 5 x 7-Matrix zu finden.

Den Aufbau dieser Matrix zeigt Bild 9. Die Informationen in den fünf vertikalen Spalten liefert ein ROM, das durch den ASCII-Code adressiert wird. Die Informationen an den Ausgängen werden ferner von der Reihenadresse (Row Address) bestimmt; sie muß von der Steuerung des IC erzeugt werden. Bild 10 zeigt das Blockschaltbild des Zeichengenerators. Reihenadresse und ASCII-Code bilden zusammen die Gesamtadresse.

Die 9 Adreßbits bieten $2^9 = 512$ Kombinationsmöglichkeiten. Pro Zeichen werden 8 Bitkombinationen benötigt, so daß 64 verschiedene Zeichen zur Verfügung stehen. Abhängig vom ASCII-Code erscheint an den fünf IC-Ausgängen das zur betreffenden Reihenadresse gehörende Bitmuster (siehe Bild 9). Die Ausgänge lassen sich über den Output-Inhibit-Eingang blockieren; sie werden dann hochohmig (Tri-State). Es können daher mehrere Zeichengeneratoren parallelgeschaltet werden, so daß auch die übrigen 64 ASCII-Zeichen (Kleinbuchstaben und Sonderzeichen) wiedergegeben werden können.

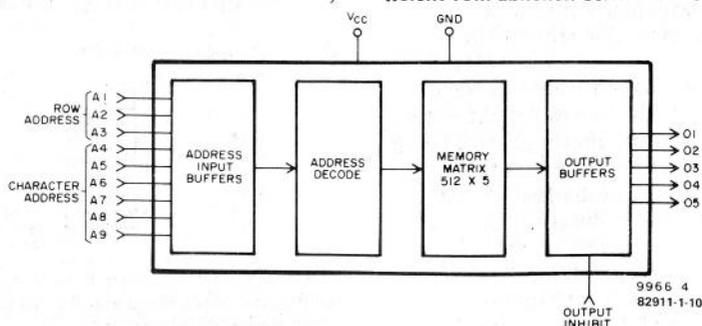
CRTC

Die meisten Hersteller von Mikroprozessoren brachten bereits einen CRTC auf den Markt. Fast alle Controller sind nur in Kombination mit einem bestimmten Mikroprozessor-Typ oder einer Prozessor-Familie einsetzbar. Der für das Elektterminal verwendete "Cathode Ray Tube Controller" (Bild 11) macht hiervon eine Ausnahme,

denn der SF.F 96364 von Sescosem führt alle notwendigen Funktionen selbständig oder zusammen mit einer kleinen Hilfschaltung aus. Diese Eigenschaft verhilft ihm zu einer größeren Vielseitigkeit als die meisten anderen Video-Controller. Eine der wichtigsten Funktionen ist die Synchronisation des TV-Bilds. Gesteuert von einem On-Chip-Quarzoszillator erzeugt der SF.F 96364 mit guter Näherung ein CCIR-gerechtes Synchronisationssignal. Das Synchronisationssignal enthält sowohl die Zeilen- als auch die Rastersynchronisationsimpulse. Der gleiche Generator steuert den Display-Counter, der für die Adressierung des Zeichengenerators (Reihenadresse) und der Speicher zuständig ist. Der Display-Counter liefert Signale an den Cursor- und den Page-End-Komparator. Das Ausgangssignal des Cursor-Komparators wird benötigt, um den Cursor an der richtigen Stelle, auf dem Bildschirm sichtbar zu machen. Mit dem Page-End-Komparator läßt sich der Adreßbereich auf mehrere Seiten ausweiten. Das RS-Signal wird dann dazu benutzt, den Übergang von einer Seite zur anderen an eine beliebige Stelle des Bilds zu legen. Das \overline{RP} -Signal dient als Taktsignal für den Zähler, der für die Erweiterung des Adreßbereichs notwendig ist. Im einzelnen wird die Ausweitung der Speicherkapazität auf maximal 4 Seiten in einem nachfolgenden Abschnitt beschrieben. Obwohl die Anzahl der aufgezählten

Bild 10. Der Zeichen-(Character-) Generator ist ein speziell für diesen Zweck programmiertes ROM; lediglich die Wortbreite von 5 Bit weicht vom üblichen Standard ab.

10



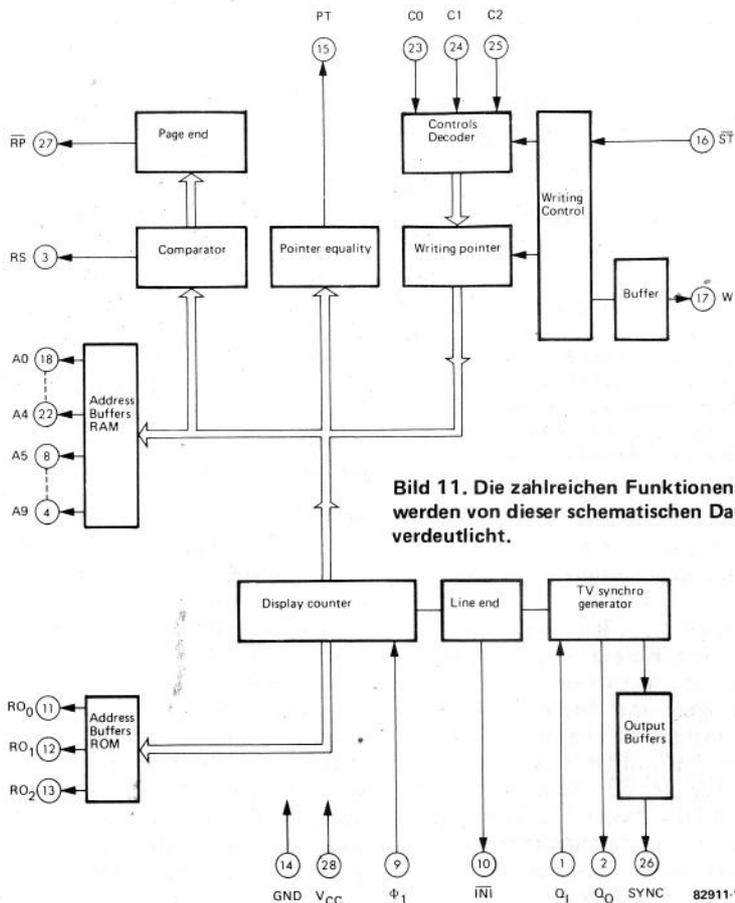


Bild 11. Die zahlreichen Funktionen des CRTC werden von dieser schematischen Darstellung verdeutlicht.

Funktionen bereits recht stattlich ist, fehlt noch das Wichtigste. Das Adressieren eines Speichers und das Erzeugen von Synchronisationssignalen ist nämlich sinnlos, solange nicht außerdem verschiedene Steuerfunktionen aufgeführt werden. Diese Aufgabe erfüllen der "Control Decoder", der "Writing Pointer" und der "Writing Controller". Der im Blockschema angegebene CTL-Dekoder (siehe Bild 7) liefert ein 3 Bit-Ausgangssignal, das zu einem internen Dekoder des SF.F 96463 gelangt. Abhängig von dem an den Anschlüssen C₀, C₁ und C₂ liegenden Bitmuster läßt der Dekoder den CRTC einen in der Tabelle 4 angegebenen Befehl ausführen. Die für die einzelnen Befehle benötigte Zeit ist zwar relativ lang, dafür werden

jedoch alle Operationen ohne störende Nebeneffekte ausgeführt. Dies ist nur möglich, wenn die Operation außerhalb des sichtbaren Bilds, also während des Zeilenrücklaufs oder in der Bildaustastlücke stattfinden.

Die Anzahl der Befehle läßt sich über das W(=Write)-Signal vergrößern. Von dieser Möglichkeit wird auch beim Elektterminal Gebrauch gemacht.

Schaltung

Die in Bild 12 gezeichneten 21 ICs bilden zusammen mit den übrigen Komponenten die komplette Schaltung des Video-Interface. Es fehlt nur noch das Keyboard mit seinem Encoder, damit das Elektterminal komplett ist. Erst wenn man

Tabelle 4.

	C ₂	C ₁	C ₀	Execution time ms
Page erase and cursor home (top-left)	0	0	0	132
End of line erase and cursor return (at left)	0	0	1	8.3
Line feed (cursor down)	0	1	0	8.3
Inhibition of the character sent	0	1	1	8.3
Cursor left (one position)	1	0	0	8.3
Erasure of cursor-line	1	0	1	8.3
Cursor up (one position)	1	1	0	8.3
Normal character	1	1	1	8.3

Tabelle 4. Übersicht über die Baudraten mit den zugehörigen theoretischen Teilfaktoren (Tab. 5a). Durch Runden der Teilfaktoren (Tab 5b) läßt sich ein Low-Cost-Baudrate-Generator mit einer Toleranz von 1% realisieren.

über beides verfügt, lassen sich die Möglichkeiten dieser Einheit voll ausschöpfen.

Der Speicher (IC1 . . . IC6) weist nur eine Breite von 6 Bit auf, so daß hierfür die preisgünstigen RAMs vom Typ 2102 A4 geeignet sind. Die Ziffer 4 am Schluß der Typenbezeichnung deutet auf ein "schnelles" IC hin; in diesem Fall beträgt die Zugriffszeit 450 ns. Eventuell können auch Low-Power-ICs verwendet werden; dies drückt den Strombedarf um ca. 30% herab, was insbesondere bei Erweiterung des Speichers auf mehrere Seiten ins Gewicht fällt. Die Typenbezeichnung lautet dann meistens 2102 AL4, sie kann abhängig vom Fabrikat geringfügig variieren.

Die im Speicher befindliche Information wird pro Bild achtmal ausgelesen, denn die Zeichen sind wie bereits beschrieben aus Punktreihen aufgebaut (siehe Bild 9). Da zu einer Zeile 64 Zeichen oder Zwischenräume gehören, muß der Speicher in entsprechend langen Blöcken abgetastet werden. IC10, der CRTC, sorgt dafür, daß dies in der richtigen Weise geschieht. Jeder Speicherblock wird achtmal hintereinander ausgelesen, wobei an den R-Ausgängen die jeweiligen Bitmuster für den Zeichengenerator liegen.

Um die relativ niedrige Geschwindigkeit des Speichers aufzufangen, ist ferner ein

Zwischenspeicher (IC9) vorhanden. Der Seitenspeicher erhält dadurch genügend Zeit, das ASCII-Signal auf seine Ausgänge zu setzen. Von IC9 gelangt das ASCII-Signal zum Zeichengenerator, der anschließend ein aus 5 Bit bestehendes Signal an Schieberegister IC12 abgibt. Dieses Schieberegister sorgt dafür, daß die Bitmuster sichtbar gemacht werden können. Es wird von einem "Dot Clock"-Signal (dot = Punkt) gesteuert, dessen Frequenz bei 11 MHz liegt. Der Dot-Clock-Generator ist mit N17, N18 und N26 aufgebaut. Da alle acht Reihen eines Zeichens genau untereinanderstehen müssen, wird der Dot-Clock-Generator vom CRTC synchronisiert. Dies geschieht mit Hilfe des INT-Signals (siehe Bild 11), das nach dem 64. Zeichen auf logisch "0" geht. Der Dot-Clock-Generator wird hierdurch bis zum Ende des folgenden Zeilensynchronisationsimpulses blockiert. Vom Dot-Clock-Signal ist die gesamte Speicheradressierung abhängig; von diesem Signal wird nämlich mit einem 8-Teiler (IC13) auch das Letter-Clock-Signal abgeleitet. Dieses Signal steuert den Adreßzähler über den $\Phi 1$ -Eingang. Die Frequenz des Dot-Clock-Signals bestimmt die Breite der Zeichen: Je niedriger die Frequenz liegt, desto breiter erscheinen die Zeichen auf dem Schirm. Natürlich kann die Frequenz nicht beliebig niedrig sein, da das Bild dann über den Rand hinausgeschrieben wird. Andererseits hat eine zu hohe Frequenz eine schlechte Ausnutzung der Schirmfläche zur Folge, was sich nachteilig auf Schärfe und Lesbarkeit der Zeichen auswirkt. Mit C2 ist die Frequenz des Dot-Clock-Signals so

Spannungsteiler R14/R15 bestimmt das Amplitudenverhältnis zwischen Video- und Synchronisationssignal. Die Werte dieser Widerstände sind so gewählt, daß der Schwarzpegel bei ca. 30% liegt. Auf den Video-Combiner folgt eine Pufferstufe mit einer Ausgangsimpedanz von 68Ω, an die unmittelbar ein Koaxkabel angeschlossen werden kann. Bei richtiger Anpassung darf das Kabel, ohne daß Schwierigkeiten zu befürchten sind, bis zu 10 m lang sein; allerdings muß auch das andere Ende des Kabels mit der richtigen Impedanz abgeschlossen werden. Die Pufferstufe hebt den Schwarzpegel auf ungefähr 35% an. Dies läßt sich mit R14 und R15 korrigieren, hier wurde jedoch davon abgesehen, da dieser Effekt nicht störend in Erscheinung tritt. Der gesamte für den Aufbau des Videosignals zuständige Schaltungsteil ist damit besprochen. Übrig bleibt nur noch der Teil, der die Kommunikation mit der Außenwelt ermöglicht.

Als Wichtigstes ist hier der UART zu nennen; seine Eigenschaften und Aufgaben wurden bereits erläutert. Um die Übertragungsgeschwindigkeit festzulegen, muß dem UART ein Taktsignal mit einer um das 16fache höher als die Baudrate liegenden Frequenz zugeführt werden. Meistens verwendet man zur Erzeugung dieses Signals einen monolithisch integrierten Baudrate-Generator. Ein solches IC ist jedoch recht kostspielig; außerdem wird ein 1MHz-Quarz benötigt, um durch Teilung die gewünschten Taktfrequenzen zu erhalten. Es liegt nahe, den bereits für den CRTC vorhandenen Quarz auch zur Erzeugung der Baudrate-Taktsignale heranzuziehen. Dies läßt sich mit einem einstellbaren Teiler (IC14 und IC15) bewerkstelligen, dem das von N14 verstärkte CRTC-Oszillatorsignal zugeführt wird. (Geeignet sind ausschließlich ICs der Firmen Fairchild, National Semiconductor, Philips, Valvo und Toshiba.) Mit Schalter S2 kann die Übertragungsgeschwindigkeit gewählt werden. Die Baudrate-Frequenzen werden mit einer Toleranz von 1% eingehalten. Im allgemeinen lassen die für die Datenübertragung benutzten Systeme wesentlich größere Toleranzen zu, so daß

diese geringe Abweichung ohne Bedeutung ist. Aus Tabelle 5 geht der Zusammenhang zwischen der UART-Frequenz und den theoretischen (Tab. 5a) und tatsächlichen Teilfaktoren (5b) hervor. Der Hersteller des SF.F 96364 empfiehlt eine Quarzfrequenz von 1008 kHz, um Interferenzen mit der Netzfrequenz zu vermeiden. In der Praxis erfüllt auch ein 1 MHz-Quarz diese Aufgabe. Dabei kann jedoch eine Interferenz (Überlagerung) entstehen, die sich in Form eines langsam bewegenden horizontalen grauen Balken bemerkbar macht.

Gerundet auf ganze Zahlen ergeben sich für beide Quarzfrequenzen die gleichen Teilfaktoren, ohne daß die genannte Toleranz von 1% überschritten wird. Das Resultat ist ein Baudrate-Generator, der nur ungefähr 20% des Preises kostet, der für eine monolithische Ausführung bezahlt werden muß. Die Testergebnisse beweisen, daß dieser dem Hobby-Etat zugute kommende Kunstgriff keinerlei nachteilige Auswirkungen auf die Brauchbarkeit der Schaltung hat. Nachdem für das Taktsignal des UART gesorgt ist, steht der Kommunikation mit der Außenwelt nichts mehr im Weg. Grundlage für diese Kommunikation sind im

Tabelle 5a.

Baudrate	fUART	Teilfaktor 1000 MHz	Teilfaktor 1008 MHz
75	1200 Hz	833.33	840
110	1760 Hz	568.18	572.73
150	2400 Hz	416.67	420
300	4800 Hz	208.33	210
600	9600 Hz	104.17	105
1200	19200 Hz	52.08	52.50

Tabelle 5b.

Zusammenstellung der Teilfaktoren

Baudrate	1 MHz	1008 MHz
75	64 x 13	64 x 13 (+8)
110	44 x 13 (-4)	44 x 13
150	32 x 13	32 x 13 (+4)
300	16 x 13	16 x 13 (+2)
600	8 x 13	8 x 13 (+1)
1200	4 x 13	4 x 13

allgemeinen nicht die TTL-Spezifikationen, sondern die sogenannte RS-232C- oder V 24-Norm. Die beiden Normen sind praktisch identisch und gegeneinander austauschbar; sie legen fest, daß die Signalspannungen mindestens +5 V für log. "1" und - 5 V für log. "0" sowie maximal + 15 V für log. "1" und - 15 V für log. "0" betragen. Im Vergleich zu den TTL-Spannungen ist hier der Störabstand wesentlich größer. Beim vorliegenden Entwurf wurde ein Kompromiß zwischen der genannten Norm und den TTL-Spezifikationen angestrebt. Das Ergebnis ist ein diskret aufgebautes Interface, das sowohl der obigen Norm genügt als auch TTL-gerecht ist. Wenn der Ausgang ein TTL-Signal liefern soll, muß Diode D4 hinzugefügt werden. Diese Diode begrenzt die Ausgangsspannung auf -0,6 V; ohne Diode bewegt sich die Signalspannung zwischen +5 V und -12 V. Die Ausgangsimpedanz wurde hierbei niedrig gehalten, damit die Kabelanpassung ohne Schwierigkeiten möglich ist.

Wie bereits erwähnt, kann das Format des seriellen Ein- und Ausgangssignals eingestellt werden. Es hängt von den logischen Signalen an den Eingängen 35 . . . 39 des UART-IC ab; Tabelle 6 gibt Aufschluß über die verschiedenen Möglichkeiten.

Eine bevorzugte Einstellung (7 Bit pro Zeichen mit Even-Parity) ist in Tabelle 6 durch Rasterung hervorgehoben. Das Paritätsbit kann eventuell entfallen (No Parity Bit), da dieses nur für das abgehende Signal von Bedeutung ist. Der UART prüft zwar die Parität des ankommenden Signals, der Parity-Error-Ausgang ist jedoch nicht herausgeführt, so daß das Paritätsbit nur am Ort des Signalursprungs ausgewertet werden kann.

Die mit dem UART vom Keyboard oder über den seriellen Eingang empfangene Information wird über den Datenbus B0 . . . B6 an den Speicher und den CRTC weitergeleitet. Bevor die Datensignale den Speicher erreichen, muß der 7-Bit-ASCII-Code in einem 6-Bit-ASCII-Code umgesetzt werden. Dazu wird Bit 5 negiert und Bit 6 invertiert. Außerdem ermöglichen die Gatter N1 . . . N7 das schnelle Laden des Speichers mit dem Zwischenraum-Code (100000), so daß unter Mitwirkung des CRTC eine Zeile oder das gesamte Bild

Tabelle 6. Einstellmöglichkeiten des UART; die Tabelle gilt sowohl für den AY-5-1013 als auch für den MM 5303. Die gerasterte Konfiguration entspricht den in der Schaltung (Bild 12) gezeichneten Verbindungen.

Tabelle 6.

PIN	LEVEL	TRANSMITTED OR RECEIVED FORMAT
35	1	No parity bit
	0	Transmitted parity bit
39	1	Even parity
	0	Odd parity
36	1	2 stop bits
	0	1 stop bit
37	0	5 bits/character
38	0	
37	0	6 bits/character
38	1	
37	1	7 bits/character
38	0	
37	1	8 bits/character
38	1	

Tabelle 7.

Funktion	Taste	gleich mit
line-feed	LF	CTL J
carriage-return + erase to end of line	CR	CTL M
cursor up	VT	CTL K
cursor down	LF	CTL J
cursor left	BS	CTL H
cursor right	HT	CTL I
home cursor	FS	—
home cursor + page erase	FF	CTL L
scroll up (cursor down)	ESC	CTL [
carriage return (no erasure)	-- (GS)	CTL]
erase current line	-- (SUB)	CTL Z

auf einmal gelöscht werden kann. Dies sind einige der sogenannten Kontrollfunktionen, mit denen das Elektterminal reichhaltig ausgestattet ist. Die Kontrollfunktionen werden aus dem 7-Bit-ASCII-Code mit Hilfe eines ROM abgeleitet, im Blockschema (Bild 7) ist dieses als CTL-Dekoder bezeichnet. Der ASCII-Code wird auf die Adreßleitungen des ROM gesetzt; das Bitmuster an den Ausgängen dieses 256 x 4-Bit-ROM bestimmt, wie der CRTC die angelegte Information auswertet. Tabelle 4 faßt bereits die verschiedenen Möglichkeiten zusammen. Wird die Read-Write-Leitung des Speichers durch ein zusätzliches Bit gesteuert, so ergeben sich weitere Möglichkeiten; Tabelle 7 gibt hierüber Aufschluß. Die meisten dieser Funktionen sind am Keyboard als separate Taste herausgeführt. Der Bedienungskomfort des Terminals erreicht hierdurch ein angenehm hohes Niveau. Alle Funktionen mit Ausnahme von "Home Cursor" können auch mit den Control-Tasten und einem zugehörigen Zeichen abgerufen werden. Letzteres ist wichtig, wenn das speziell für dieses Terminal entworfene Keyboard nicht benutzt werden soll. Das PROM, das alle genannten Funktionen dekodiert, ist entsprechend Tabelle 8 programmiert. Dieses PROM wird von mehreren Herstellern unter der Typenbezeichnung 74S387 angeboten. Da der 7-Bit-Code 128 unterschiedliche Bitmuster ermöglicht, wird nur die Hälfte des PROM benutzt.

Tabelle 7. Der 4-Bit-Dekoder (PROM) eröffnet neben den in Tabelle 4 angeführten Funktionen weitere Möglichkeiten. Die Funktionen können mit speziellen Funktions-taster oder über die Control-Funktion des Keyboards aufgerufen werden.

Platine

Die Video-Interface-Platine des Elektterminals ist nicht viel größer als eine Eurokarte; sie ist nur einseitig mit Leiterbahnen bedruckt. Dies bedeutet allerdings, daß ungefähr 60 Drahtbrücken notwendig sind. Die Bilder 13 und 14 zeigen die Platine und den zugehörigen Bestückungsplan für die in diesem Artikel beschriebene Schaltung. An einer der Schmalseiten befindet sich die Kontaktleiste für das Keyboard, über die sämtliche Verbindungen, also auch die Speiseleitungen, laufen. Zu dieser Kontaktleiste ist auch der vom UART kommende Datenbus geführt; diese Anschlüsse werden jedoch erst im Erweiterungsstadium benötigt. Anstelle der Kontaktleiste genügt eventuell auch eine einfache Flachkabel-Verbindung. Über die zweite auf der Platine befindliche Kontaktleiste sind sämtliche Adreß- und Datenleitungen des Speichers sowie einige Schaltungspunkte zugänglich, die die Adressierung des Speichers ermöglichen. Hiervon wird jedoch ebenfalls erst im Erweiterungsstadium Gebrauch gemacht.

Die Programmierung des UART (IC8) erfolgt mit Hilfe von Drahtbrücken. Der Bestückungsaufdruck der Platine entspricht der in Tabelle 6 durch Rasterung hervorgehobenen Konfiguration. Es soll noch betont werden, daß das Elekterterminal eine in sich abgeschlossene Einheit bildet. Die Speicherkapazität kann bei Bedarf durch einfaches Aufstecken einer oder mehrerer Platinen vergrößert werden. Die einzige außerdem erforderliche Änderung ist das Entfernen einer Drahtbrücke.

3 x Auto-Repeat

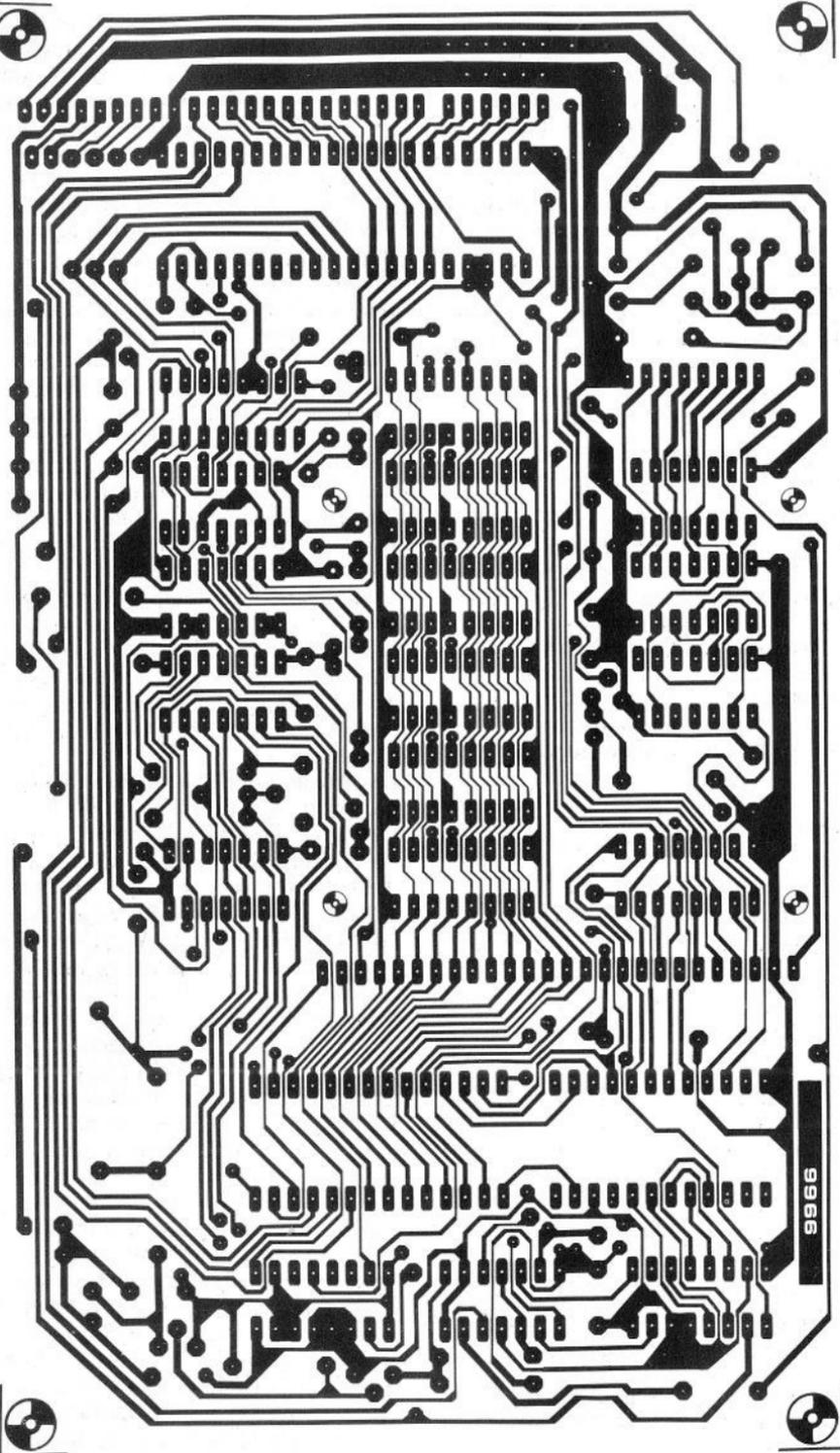
Viele Wege führen nach Rom. Übertragen auf die Auto-Repeat-Möglichkeit heißt das: Viele technische Lösungen führen zum Auto-Repeat; drei davon werden anschließend beschrieben. Die zwei ersten Schaltungen arbeiten nicht nur mit dem Elekterterminal, sondern auch mit anderen Tastaturen. Bevor nun die einzelnen Techniken betrachtet werden, ist zunächst ein kurzer Blick auf die Bedingungen für die Auto-Repeat-

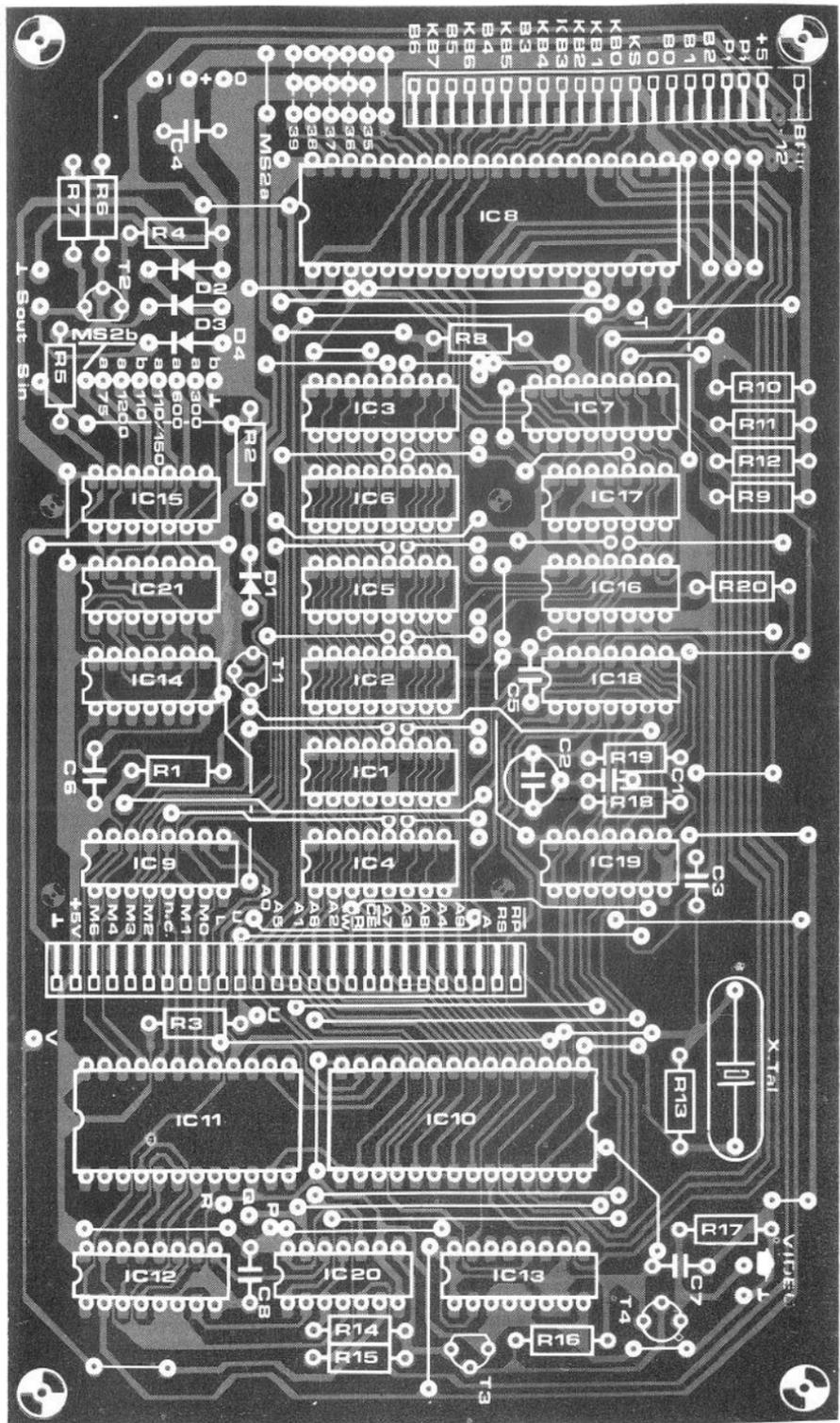
Tabelle 8.

Address	Positive logic			
	O ₃	O ₂	O ₁	O ₀
0 to 127	1	0	0	0
128 to 135	0	0	1	1
136	0	1	0	0
137	0	1	1	1
138	1	0	1	0
139	0	1	1	0
140	1	0	0	0
141	1	0	0	1
142 to 153	0	0	1	1
154	1	1	0	1
155	0	0	1	0
156	0	0	0	0
157	0	0	0	1
158, 159	0	0	1	1
160 to 254	1	1	1	1
255	0	0	1	1

Tabelle 8. Die Programmierung von IC7 ermöglicht die Dekodierung verschiedener Funktionen.

Schaltung interessant. Das Impulsdiagramm in Bild 15 verdeutlicht, um was es geht. Obwohl sich das Diagramm auf die Schaltung aus Bild 16 bezieht, kann man es auch für die allgemeine Erklärung der Auto-Repeat-Funktion heranziehen. Die Tastatur sendet beim Drücken einer Taste einen Strobeimpuls aus, dessen Impulslänge von der Tastendruckdauer abhängt. Diese Tatsache ist wie dafür geschaffen, beim längeren Drücken einer Taste mehrere Strobeimpulse auszulösen. Dadurch wird der Tastatur simuliert, daß mehrmals hintereinander die gleiche Taste einen Impuls auslöst. Dieses Verhalten ist in Bild 15 deutlich dargestellt. Ein kurzer Tastenanschlag (weniger als 2 Sekunden) erzeugt einen Strobe-Ausgangsimpuls, der ebenfalls unter der maximalen Dauer von 2 Sekunden bleibt (Bild 15a). Hält man die Taste längere Zeit gedrückt, folgen nach 2 Sekunden noch eine ganze Anzahl von Strobe-Ausgangsimpulsen. Und das so lange, bis die Taste losgelassen wird (Bild 15b). Das ermüdende Eintippen ganzer Reihen Leerräume, Punkte oder was auch sonst immer, gehört der Vergangenheit an. Das Wiederholen des Strobeimpulses kommt dadurch zustande, daß ein Oszillatorsignal nach 2 Sekunden den eigentlichen Strobeimpuls ersetzt. Eine der drei Lösungen zeigt Bild 16. Die Schaltung ist relativ aufwendig, doch erfüllt sie die genannte Aufgabe exakt. Der Oszillator ist mit dem Schmitt-Trigger N1 aufgebaut. Um das notwendige niederfrequente Signal zu erzeugen, ist der Transistor als Puffer hinzugefügt. Der monostabile Multivibrator sorgt für die Verzögerungszeit von 2 Sekunden, während N2, N4 und N5 zwischen Strobe- und Oszillatorsignal umschalten. Wie, das erläutert der folgende Abschnitt. Das Gatter N2 ist so lange gesperrt, wie das Strobe-Eingangssignal logisch 0 ist. Das gilt auch für den MMV (IC1). Geht nun das Strobe-Eingangssignal auf logisch 1, startet der MMV und der Ausgang Q wird "0". Die Folge ist, daß Gatter N2 weiterhin gesperrt bleibt. Nach 2 Sekunden kippt IC1 in den Ruhezustand zurück ($Q = "0"$ und $\bar{Q} = "1"$). Gatter N4, dessen Ausgang während der Monoflop-





Stückliste zu Bild 12

Widerstände:

R1 ... R3 = 4k7
R4 = 6Ω8
R5 = 2k2
R6 = 560 Ω
R7 = 270 Ω/1 W
R8 ... R12 = 1 k
R13 = 10 M
R14 = 150 Ω
R15, R18 = 390 Ω
R16 = 220 Ω
R17 = 68 Ω
R19 = 680 Ω
R20 = 4M7

Kondensatoren:

C1 = 27 pF (siehe Text)
C2 = 45 pF Trimmer
C3 = 10 pF
C4 ... C8 = 100 n

Halbleiter:

D1 ... D4 = DUS (D4, siehe Text)
T1 = BC 547B
T2 = BC 557B
T3 = BF 451
T4 = 2N2219
IC1 ... IC6 = 2102-1, 2102A4,
2102AL4
IC7 = SFC 71301E 1-0
(programmiert) o.a. z.B. 74S387,
MM 6300 (zu programmieren nach
Tabelle 8)

IC8 = AY-5-1013, MM 5303
IC9 = 74LS174*
IC10 = SF.F 96364 (Sescosem)
IC11 = R0-3-2513
IC12 = 74LS165*
IC13 = 74LS163*
IC14, IC15 = 4024 (siehe Text)
IC16 = 4011
IC17, IC21 = 4081
IC18 = 74LS04*
IC19 = 74LS00
IC20 = 74LS125*

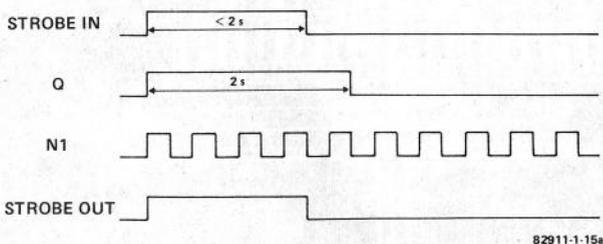
Sonstiges:

S1 = Schalter, einpolig
S2 = Stufenschalter 2 x 6
S3 = Schalter 1 x Um
Buchsenleiste Typ ITT-Cannon
G09A45C3DEAA
1 x 22 polig (Keyboard)
1 x 26 polig (Erweiterung)
Steckerleiste Typ ITT-Cannon
G09A45C4DCAA
1 x 22 polig
Die Steckverbindung kann auf das
richtige Maß geschnitten werden
X1 = Quarz 1008 kHz

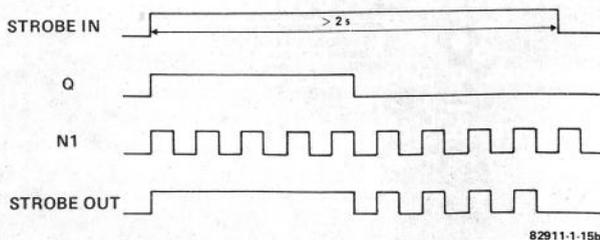
* Die Verwendung von Low Power Schottky-TTL-ICs ist zu empfehlen, TTL-ICs sind aber auch geeignet.

Bild 15. Impulsdiagramm für die Auto-Repeat-Funktion.

15a



15b



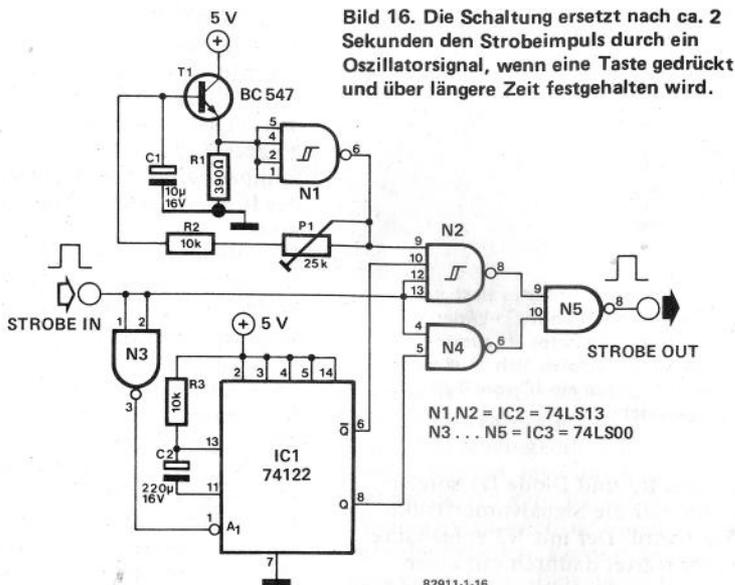


Bild 16. Die Schaltung ersetzt nach ca. 2 Sekunden den Strobeimpuls durch ein Oszillatorsignal, wenn eine Taste gedrückt und über längere Zeit festgehalten wird.

zeit von 2 Sekunden logisch 0 war, führt nun an seinem Ausgang logisch 1. Bleibt nun eine Taste länger als 2 Sekunden gedrückt, gibt Gatter N2 das Oszillatorsignal über N5 an den Strobe-Ausgang weiter. Am Ausgang stehen also Strobeimpulse mit einer Frequenz von ca. 3 Hz zur Verfügung. Erst wenn die gedrückte Taste in ihre Ruhelage zurückkehrt, wird der Strobe-Ausgangsimpuls wieder blockiert.

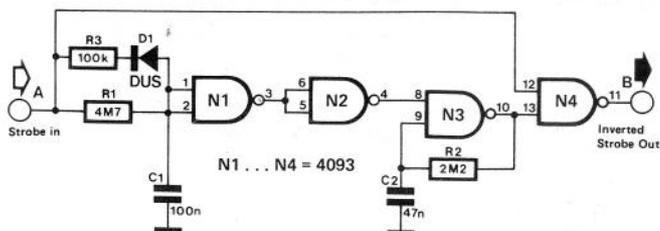
Schaltung aus Bild 17. Das Impulsdigramm aus Bild 15 ist auch für diese Schaltung zutreffend. Der einzige Unterschied besteht lediglich im Strobe-Ausgangssignal: Es ist gegenüber der Schaltung aus Bild 16 invertiert. Diese Tatsache macht das Gatter N12 im Video-Interface überflüssig. Die Schaltung in Bild 17 ist mit vier Schmitt-Triggern aufgebaut. Das Strobe-Eingangssignal gelangt über ein RC-Glied verzögert an den Schmitt-Trigger N1.

Alles in einem IC

Die in Bild 16 vorgestellte Schaltung benötigt insgesamt 3 ICs, um die Auto-Repeat-Funktion zu realisieren. Ein monostabiler Multivibrator lässt sich auch aus einzelnen Gattern aufbauen. Diese Lösung führt zu der einfacheren

Bild 17. Kompakter und einfacher wird der Aufbau für die Auto-Repeat-Funktion, wenn für den Aufbau der Schaltung ein Schmitt-Trigger-IC zur Verfügung steht. Die Dauer der Verzögerungszeit und die Wiederholfrequenz lassen sich durch andere Kondensatorwerte für C1 und C2 in gewissen Grenzen variieren.

17



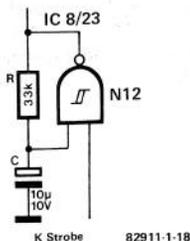


Bild 18. Ist die Tastatur festes Bestandteil des Terminals, sind nur ein Widerstand und ein Kondensator als zusätzliche Bauelemente erforderlich. Außerdem wird IC16 gegen ein IC vom Typ 4093 ausgetauscht.

Widerstand R3 und Diode D1 sorgen dafür, daß nur die Signalvorderflanke verzögert wird. Der mit N3 aufgebaute Oszillator startet dadurch mit einer Verzögerung von ca. 0,5 s, stoppt jedoch sofort, wenn die Taste losgelassen wird. Die Oszillatorfrequenz beträgt ca. 15 Hz. Beide Werte, Verzögerungszeit und Oszillatorfrequenz, lassen sich durch andere Kondensatorwerte für C1 und C2 individuell ändern.

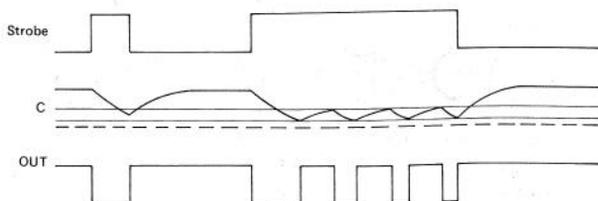
Ohne IC

Die beiden beschriebenen Schaltungen sind selbständige Einheiten, mit denen sich bei fast allen Tastaturen zusätzlich eine Auto-Repeat-Funktion realisieren läßt. Ist nun eine Tastatur ständig mit dem Video-Interface verbunden, kann man sie als Einheit des Systems betrachten und nicht mehr als eigenständige Schaltung. In diesem Fall ist es dann auch nicht mehr erforderlich, die Tastatur mit einer eigenen Auto-Repeat-Schaltung auszurüsten, sondern es genügt ein kleiner Eingriff ins Video-Interface und die Sache

ist perfekt. Der Eingriff ist einfach: Anstelle von Gatter N12 tritt ein verzögernd startender Oszillator. N12 gehört zu IC16, ein vierfaches NAND-Gatter 4011, das ersetzt wird durch ein Schmitt-Trigger-IC 4093. Beide Typen sind pin-kompatibel, so daß nach dem Austausch des ICs das Elekterminal wie gewohnt arbeiten muß.

Das ändert sich, wenn N12 zu einem Oszillator umgebaut ist. Der Schmitt-Trigger ist hierzu bestens geeignet. Bild 18 zeigt, wie aus dem Schmitt-Trigger N12 ein Oszillator wird. Dazu sind nur zwei Bauelemente erforderlich: ein Kondensator und ein Widerstand. Wenn der Umbau abgeschlossen ist, verhält sich N12 wie es Bild 19 zeigt. Eine nur kurz angeschlagene Taste erzeugt nicht wie bei den beiden anderen Schaltungen einen definierten Strobeimpuls. Die Dauer des Strobeimpulses am Eingang von N12 ist identisch mit der Impulsdauer am Ausgang. Nur ist der Ausgangsimpuls invertiert. Der Tastenanschlag muß allerdings so kurz sein, daß sich der Kondensator nicht bis zur oberen Triggerschwelle des Schmitt-Triggers entladen kann. Bleibt die Taste über eine längere Zeit gedrückt, unterschreitet die Kondensatorladung die obere Triggerschwelle. Ist die untere Triggerschwelle erreicht, schaltet N12 um, bis der Kondensator wieder die obere Triggerschwelle erreicht. Schmitt-Trigger N12 oszilliert also innerhalb der beiden Triggerschwellen (Hysterese), bis die Taste wieder in die Ruhestellung zurückkehrt. Einfacher geht's nicht mehr.

Bild 19. Das Impulsdiagramm macht die Funktion der Schaltung aus Bild 18 deutlich.



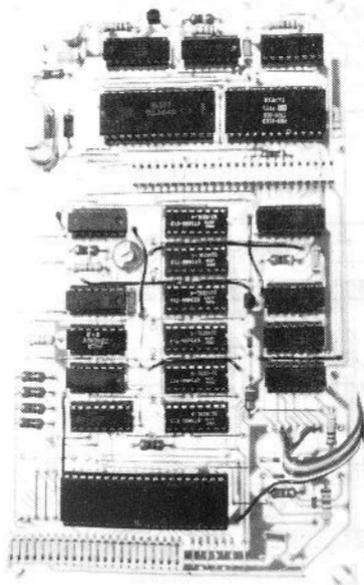
Anschluß an den TV-Empfänger

Nicht an jedem TV-Gerät ist ein Video-Eingang vorhanden. Wenn dies doch der Fall ist, muß die Videospannung des Terminals an die Empfindlichkeit des Video-Eingangs angepaßt werden. Dies geschieht am einfachsten wie folgt: Über ein gebräuchliches Koax-Antennenkabel (50 . . . 75Ω) wird das Signal zum TV-Gerät oder Monitor geführt. Auf der Seite des Wiedergabegeräts muß das Kabel niederohmig abgeschlossen werden. Hierzu eignet sich ein 100 Ω-Trimpoti, mit dem auch die Signalamplitude angepaßt werden kann. Diese Lösung ist bei Geräten mit bereits intern vorhandener niedriger Abschlußimpedanz nicht möglich; in solchen Fällen muß nach einer gleichwertigen Lösung Ausschau gehalten werden.

Wenn ein Video-Eingang fehlt, kann das TV-Gerät eventuell nachgerüstet werden. Allerdings läßt sich dafür kein allgemeingültiges Rezept angeben. Unter Umständen kann hier der Kundendienst des Herstellers weiterhelfen. Unbedingt zu beachten ist auf jeden Fall, daß das



Chassis des TV-Empfängers mit dem 220 V-Netz galvanisch in Verbindung steht. Um lebensgefährliche Situationen auszuschließen, muß das als Teil des Terminals benutzte TV-Gerät an einem Trenntrafo ausreichend hoher Leistung betrieben werden. Kommt der Anschluß über einen Video-Eingang nicht in Betracht, so erhält man auch mit einem VHF-UHF-TV-Modulator eine akzeptable Wiedergabe. Infolge der hohen Bandbreite des Modulationssignals ist bei diesem Verfahren eine geringe Einbuße an Schärfe (ca. 5%) unvermeidlich. Ob mit oder ohne Modulator, die Einstellung der Signalamplitude ist stets so vorzunehmen, daß die Synchronisation sowohl bei positiver als auch bei negativer Wiedergabe einwandfrei arbeitet. Kontrollieren läßt sich dies durch wiederholtes Umschalten vor S3; das Bild muß hierbei sofort "stehen". Voraussetzung ist natürlich, daß kein Fehler im TV-Gerät vorliegt.



Speisung

Die 5 V-Speisung wird vom Terminal bei Bestückung mit Standard-Speicher-ICs mit ca. 750 mA belastet; dieser Wert sinkt bei Verwendung von Low-Power-Speichern auf ca. 550 mA. Im negativen Speiseweig (-12 V) fließen höchstens 100 mA. Das Netzteil eines nicht erweiterten Mikrocomputers kann daher dank seiner Überkapazität die Speisung des Terminals übernehmen.

Mehr Seiten für das Elekterminal

Der Textspeicher des Elekterminal läßt sich ohne weiteres um einige Seiten erweitern. Da die Elekterminal-Platine für eine solche Erweiterung vorbereitet ist, bereitet der Anschluß der Speichererweiterung keinerlei Probleme. Mit der Erweiterung umfaßt die Textkapazität des Elekterminal vier Seiten.

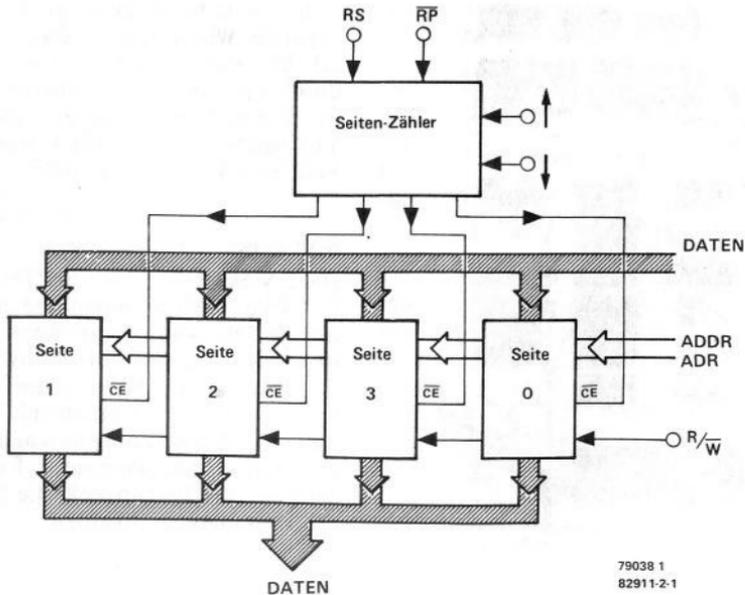
Eine Textspeicherkapazität von 16 Zeilen reicht in der Praxis nur selten aus, denn schon ein einfaches BASIC-Programm ist meistens umfangreicher. Eine Erweiterung des vorhandenen, nur eine Seite umfassenden Textspeichers erscheint deshalb dringend notwendig. Die Erweiterung auf mehrere Seiten macht eine Steuerung erforderlich; sie hat die Aufgabe, die auf dem Bildschirm erscheinende Seite auszuwählen. Dabei können im Bild entweder eine vollständige Seite oder aber die Teilstücke von zwei benachbarten Seiten sichtbar sein. Das Blockschema in Bild 1 macht die Struktur des erweiterten Textspeichers deutlich: Über die Chip-Enable-Eingänge (CE) der Speicher

steuert der Seitenzähler (page counter) die gewünschte Seite an. Die Informationen der Seiten 1, 2 und 3 sind auf der Erweiterungsplatine gespeichert, die Seite 0 befindet sich dagegen auf der Terminalplatine. Der Seitenzähler wird vom CRTC des Terminal und von den Up- und Down-Tastern des ASCII-Keyboards gesteuert.

Eine bequeme Handhabung der Seiten ist nur dann gegeben, wenn der Textspeicher folgende Eigenschaften besitzt:

1. Der Seitenzähler muß vorwärts und rückwärts zählen.

Bild 1. Blockschema des erweiterten Textspeichers mit dem Seitenzähler. Der Speicher für Seite 0 befindet sich auf der Terminalplatine.



79038 1
82911-2-1

- Nach der letzten Seite muß wieder die erste Seite im Bild erscheinen.
- In umgekehrter Richtung muß die letzte Seite auf die erste Seite folgen.
- Teilstücke von zwei benachbarten Seiten (z.B. die zweite Hälfte von Seite 2 und die erste Hälfte von Seite 3) müssen gleichzeitig ins Bild gebracht werden können.

Man kann sich den Textspeicher als eine in beiden Richtungen drehbare Litfaßsäule vorstellen, auf der nebeneinander mehrere gleich große Plakate (die Seiten) angebracht sind; für den Betrachter der Litfaßsäule ist jedoch nur ein Feld von der Breite eines Plakats (der Umfang einer Seite) sichtbar.

Seitenzähler

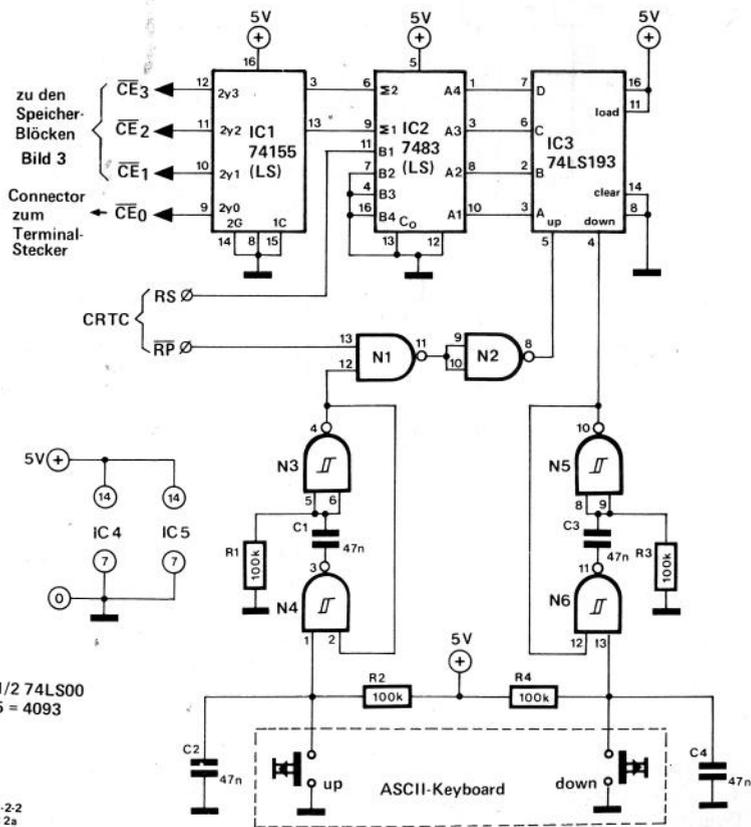
Aus dem Vorstehenden wird deutlich, daß der Seitenzähler ein wichtiges Bindeglied zwischen Terminal und

Speicher ist. Vor der Erklärung der Arbeitsweise sei noch einmal an den im Elektrotterminal befindlichen CRTC erinnert: Dieser enthält eine Page-End-Komparator, der die Signale RP und RS liefert.

Das Signal RS ändert seinen logischen Zustand, wenn beim Schreiben des Bilds eine Seitengrenze überschritten wird. Solange eine vollständige Seite im Bild erscheint, ist RS logisch 1. Werden jedoch die Teilstücke von zwei Seiten auf den Schirm geschrieben, dann gilt der untere Bildteil als "actual page". Während dieses Teilstücks ist R6 ebenfalls logisch 1, für die Dauer des oberen Teilstücks dagegen logisch 0. Sind zum

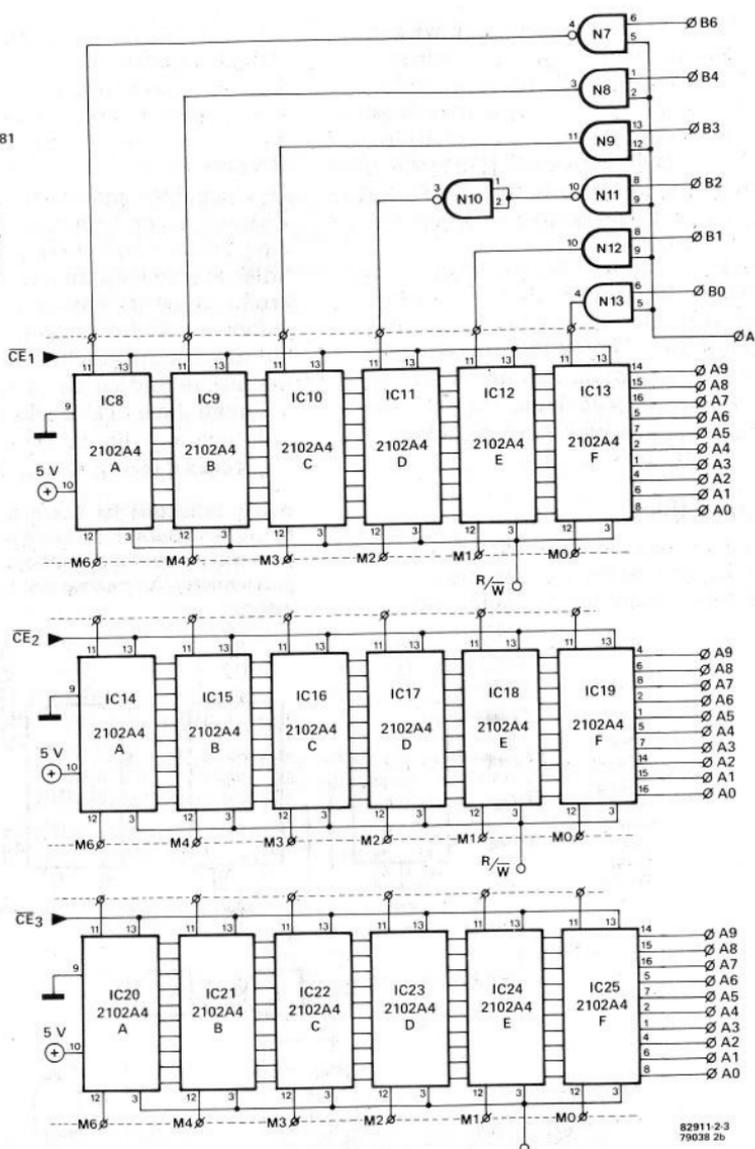
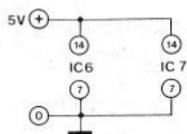
Bild 2. Schaltung der Textspeicher-Erweiterung. Die Bezeichnungen der Anschlüsse beziehen sich auf die in gleicher Weise bezeichneten Anschlüsse der Terminalplatte.

2



82911-2-2
79038 2a

N7,N10,N11 = IC6 = 4011
 N8,N9,N12,N13 = IC7 = 4081



82911-2-3
 79038 2b

Beispiel die Zeilen 7 ... 16 von Seite 2 und die Zeilen 1 ... 6 von Seite 3 auf dem Bildschirm sichtbar, dann ist RS während der ersten zehn Zeilen "0" und während der folgenden sechs Zeilen "1".

Der \overline{RP} -Ausgang des CRTIC liefert einen "0"-Impuls, wenn am unteren Bildrand eine Seitengrenze überschritten wird. Dieser Impuls erscheint nur, sofern durch Drücken des LF- oder des ESC-

Bild 3. Pro Seite sind 6 RAMs Speicher-Kapazität erforderlich.

Tasters der Übergang zur nächstfolgenden Seite notwendig wird. Die Signale \overline{RP} und RS werden zur Steuerung des Seitenzählers benutzt.

Im einzelnen besteht der Seitenzähler aus einem Vorwärts/Rückwärts-Zähler, einem 4-Bit-Paralleladdierer, und einem 2-nach-4-Demultiplexer. Durch einen

\overline{RP} -Impuls oder durch Drücken des Up-Tasters erhöht sich der Stand des Vorwärts/Rückwärts-Zählers um eins; bei Drücken des Down-Tasters wird er um eins herabgesetzt. Der Paralleladdierer ermittelt anschließend die binäre Summe von Zählerstand und RS-Signal. Die Summe (2 Bit) steuert schließlich über den Demultiplexer die CE-Eingänge der vier Speicherblöcke.

Bei der Darstellung einer vollständigen Seite im Bild ist RS logisch 1. Da diese "1" zum Zählerstand addiert wird, verschiebt sich die Nummerierung der Seiten um eins. Im Blockschema (Bild 1) sind die Seiten in der Folge angegeben, in der der Seitenzähler zählt. Die letzte Seite (Seite 0) ist die auf der Terminalplatine gespeicherte Seite.

Wenn die Teilstücke von zwei aufeinanderfolgenden Seiten im Bild erscheinen, ist RS während der Zeit logisch 0, in der das Teilstück der vorhergehenden Seite geschrieben wird; der Seitenzähler schaltet daher stets zum richtigen Zeitpunkt von der einen Seite zur nächsten um.

Der Aufbau und die Arbeitsweise der drei Seitenspeicher ist identisch mit dem im Artikel "Elekterterminal" beschriebenen, nur eine Seite umfassenden Speicher, so daß hierauf an dieser Stelle nicht näher eingegangen werden soll.

Bau

Wie aus Bild 2 und Bild 3 hervorgeht, ist die Schaltung der Textspeicher-Erweiterung relativ unkompliziert. Der Speicher selbst, der die drei zusätzlichen Seiten aufnimmt, besteht aus 18 RAMs vom Typ 2102A4 (Bild 3). Auch Low-Power-ICs des gleichen Typs (Bezeichnung 2102AL4) lassen sich verwenden; der Strombedarf wird hierdurch um ca. 30% reduziert.

Der Seitenzähler besteht aus IC1 (2-nach-4-Demultiplexer), IC2 (Paralleladdierer) und IC3 (Vorwärts/Rückwärts-Zähler). Die Taster "Up" und "Down" werden durch die Schaltungen mit N3/N4 bzw. N5/N6 entprellt. Die Bezeichnungen der Anschlüsse entsprechen den Be-

zeichnungen in der Schaltung des Elekterterminal.

Platine

Die Erweiterungsplatine (Bild 4) ist mit zwei Kontaktleisten versehen, über die sie bequem mit der Terminalplatine verbunden werden kann. Die 26-polige Leiste wird auf der Unterseite der Erweiterungsplatine montiert, damit sie in ihr Gegenstück auf der Terminalplatine paßt. Die Verbindungen B0 . . . B4, B6 und die Verbindungen zum Up- und Down-Taster laufen nicht über diese Kontaktleiste. Hierfür ist die zweite, 8-polige Kontaktleiste vorgesehen; ihre Anschlüsse sind mit den entsprechenden Punkten der anderen Kontaktleiste auf der Terminalplatine zu verbinden. Will man die Kontaktleisten einsparen, so können die Verbindungen natürlich auch direkt hergestellt werden.

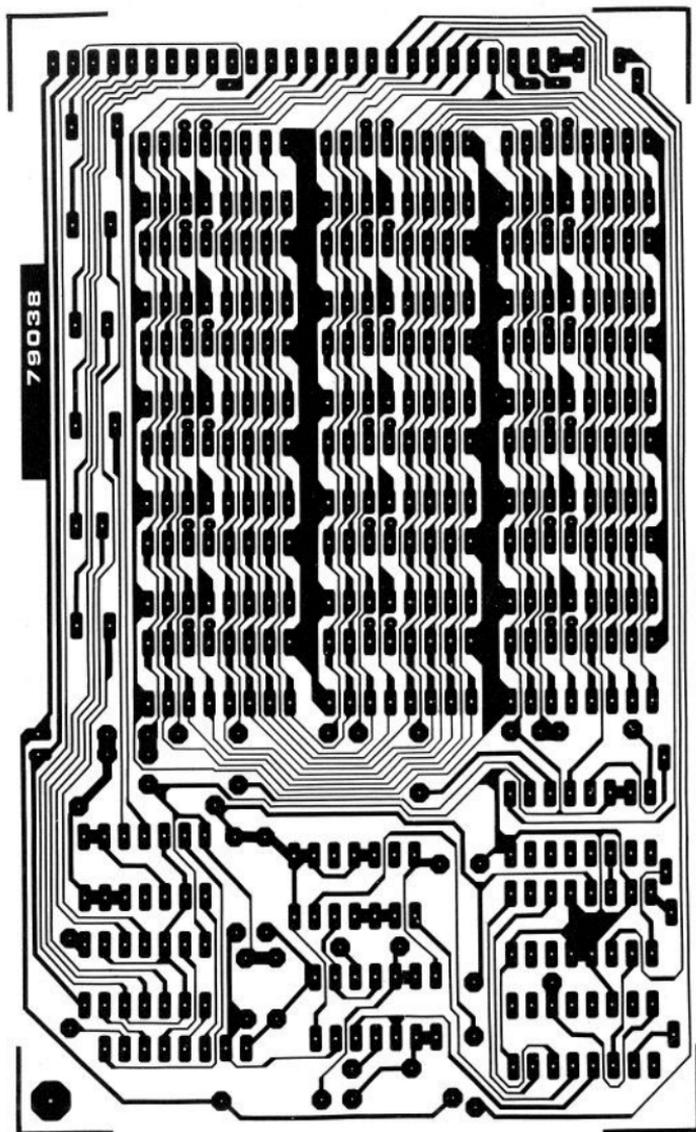
Nach Anschließen der Erweiterungsplatine muß noch eine kleine Änderung an der Terminalplatine vorgenommen werden: Die Kupferbahn von CE nach Masse ist zu unterbrechen; ihre genaue Lage geht aus Bild 5 hervor.

Die auf dem Keyboard bereits herausgeführten Anschlüsse des Up- und Down-Tasters können entweder direkt oder über die Terminalplatine mit der Erweiterungsplatine verbunden werden. Macht man von Kontaktleisten Gebrauch, so wird man die zweite Lösung vorziehen.

Bedienung

Die Konzeption des erweiterten Seitenspeichers bietet ein hohes Maß an Bedienungskomfort. Wenn bei der Arbeit mit dem Terminal die Kapazität der ersten Seite überschritten wird, geht das Terminal selbsttätig auf die nächste Seite über. Überschreitet man die Gesamtkapazität von 64 Zeilen, dann beginnt das Terminal wieder auf der ersten Seite. Die dort bereits vorhandene Information geht allerdings verloren.

Die Taster "Up" und "Down" dienen zum Weiterschalten der Seiten in Vorwärts- bzw. Rückwärtsrichtung. Mit dem



Stückliste zu den Bildern 2 und 3

Widerstände:

R1 ... R4 = 100 k

Kondensatoren:

C1 ... C4 = 47 n

Halbleiter:

IC1 = 74LS155*

IC2 = 74LS83*

IC3 = 74LS193

IC4 = 74LS00

IC5 = 4093

IC6 = 4011

IC7 = 4081

IC8 ... IC25 = 2102-1, 2102A4,
2102AL4

Sonstiges:

Fassungen für alle ICs:

21 Stück 16-p.-DIL

4 Stück 14-p.-DIL

Steckerleiste auf der Unterseite der Platine

(nicht unbedingt erforderlich):

GO9A45C4DBAA ITT-Cannon

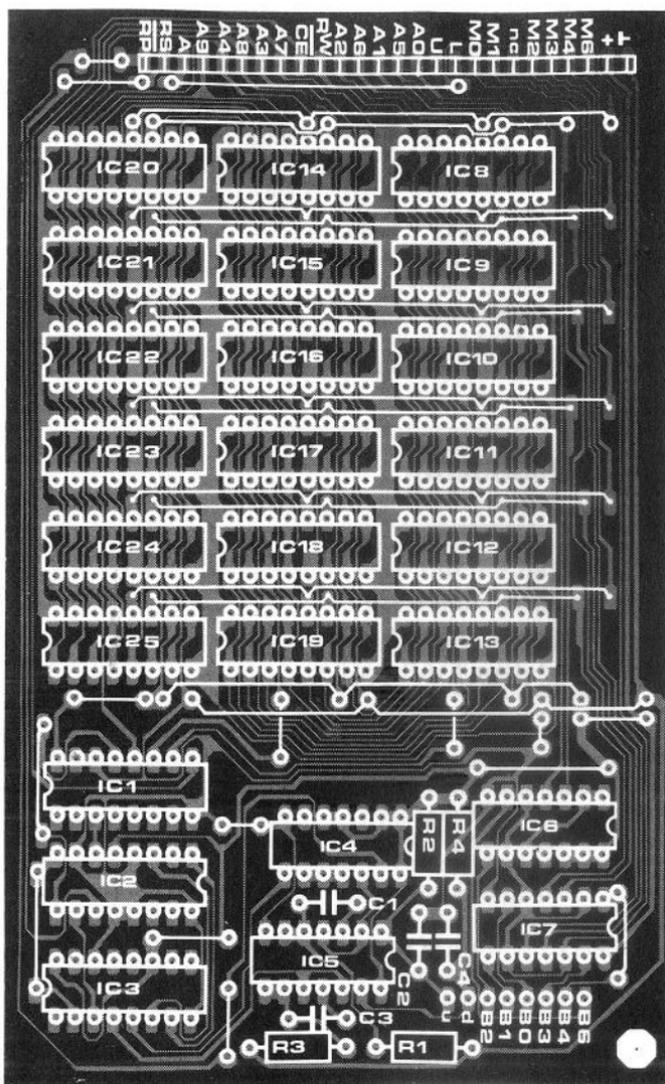
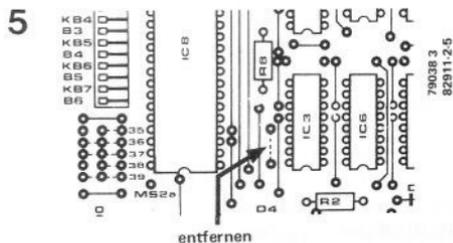


Bild 4. Platine und Bestückungsplan für die Textspeicher-Erweiterung.

oder, bei Verwendung von Flachbandkabel, GO9A45C4DCAA ITT-Cannon (auf Maß kürzen)

* Anmerkung:
Die Verwendung von LS-TTL-Typen wird empfohlen, Standard-TTL-Typen sind aber auch verwendbar.

Bild 5. Nach Anschließen der Erweiterungsplatine muß auf der Terminalplatine eine Drahtbrücke unterbrochen werden. Das Bild gibt ihre genaue Lage an.



Taster "ESC" kann der Text zeilenweise weitergeschaltet werden, so daß auch aneinander anschließende Teilstücke von zwei Seiten gleichzeitig ins Bild gebracht werden können. Wenn man in diesem Fall den Up- oder Down-Taster drückt, erscheint keine komplette Seite, sondern es werden die folgenden bzw. die vorangegangenen 16 Zeilen sichtbar. Auch die Taster "LF" kann zur Weiter-schaltung der Zeilen benutzt werden; dies hat jedoch eine Leerzeile auf dem Schirm und gleichzeitig das Löschen dieser Zeile aus dem Speicher zur Folge.

Speisung

Wird die Schaltung mit Speicher-ICs in Standardausführung aufgebaut, so beträgt die Stromaufnahme ungefähr 600 mA; sie sinkt auf etwa 400 mA, wenn man Low-Power-ICs verwendet. Sollte die vorhandene Elekterminal-Speisung der zusätzlichen Belastung nicht gewachsen sein, so muß ihre Leistung entsprechend erhöht werden.

VHF/UHF-TV-Modulator

Die Schaltung erzeugt ein mit dem zugeführten Videosignal moduliertes HF-Signal, dessen zahlreiche Harmonische sowohl in den VHF- als auch in den UHF-TV-Bereich fallen. Sie dient als Bindungsglied zwischen dem Video-Interface und dem TV-Empfänger.

Vor der Schaltungsbeschreibung und der Erklärung der Arbeitsweise sollen zuerst einige grundsätzliche Zusammenhänge betrachtet werden, deren Kenntnis hier notwendig erscheint.

In Bild 1a ist das Videosignal einer Bildzeile skizziert. Drei Amplitudenwerten kommt eine besondere Bedeutung zu: dem Weißwert, dem Schwarzwert und dem Synchronisationspegel. Die Begriffe Weiß- und Schwarzwert beziehen sich selbstredend auf die Bildhelligkeit, während der Synchronisationsimpuls dem Empfänger das Ende einer Zeile und den Beginn der nächsten Zeile signalisiert. Ein Videosignal besteht folglich aus den Zeilen- (und auch Raster-) Synchronisationsimpulsen und dem Bildinhalt. Zwischen zwei Zeilensynchronisationsimpulsen wird der Helligkeitsverlauf der Bildzeile übertragen; die Amplitude des Videosignals kann hier nur Werte zwischen dem Weiß- und dem Schwarzpegel annehmen. Aus Bild 1a geht ferner hervor, daß die Dauer einer Zeile einschließlich des Zeilensynchronisa-

tionsimpulses $64 \mu\text{s}$ beträgt (bei Videosignalen nach CCIR-Norm).

In Bild 1b ist die mit dem Videosignal aus Bild 1a modulierte Trägerwelle (z.B. eines Fernsehsenders) dargestellt. Hieraus wird deutlich, daß der Träger negativ amplitudenmoduliert ist: Die Trägeramplitude erreicht dann ihren höchsten Wert, wenn die Amplitude des Videosignals ihren niedrigsten Wert annimmt. Letzteres ist jeweils während des Synchronisationsimpulses der Fall. Bei TV-Sendern muß die Trägerwelle absolut sinusförmig verlaufen, da andere Schwingungsformen gleichbedeutend mit der Abstrahlung störender Ober- und Nebenwellen wären. Für einen TV-Modulator gelten jedoch andere Kriterien, weil das HF-Signal nicht einer Antenne, sondern dem Empfänger direkt zugeführt wird. Die Erzeugung möglichst vieler Harmonischer ist bei einem TV-Modulator sogar sinnvoll, damit der Empfangskanal freizügig gewählt werden kann. Als Träger dient bei diesem VHF-UHF-TV-Modulator

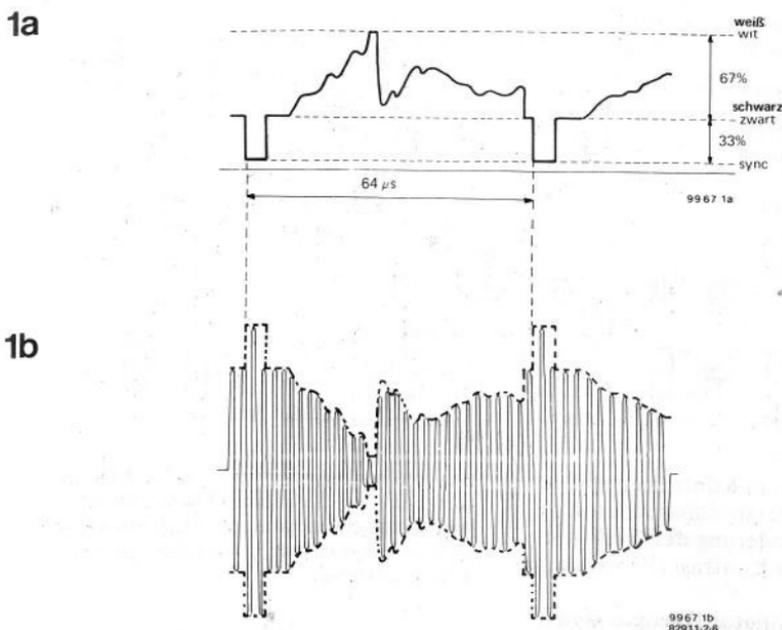


Bild 1. Prinzipieller Aufbau eines Videosignals (a) und der mit diesem Signal negativ amplitudenmodulierte Träger (b).

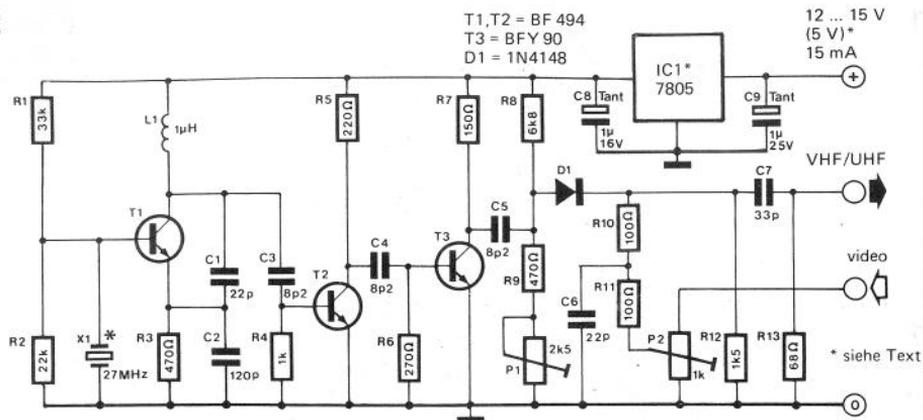
deshalb kein sinusförmiges, sondern ein nadelimpulsartiges Signal.

Zur Schwingungserzeugung wird ein preiswerter 27 MHz-Quarz benutzt. Das TV-Signal ist deshalb auf allen Frequenzen zu empfangen, die ein ganzzahliges Vielfaches von 27 MHz sind; auf diese Frequenzen fallen nämlich die Oberwellen. Das Oberwellenspektrum erstreckt sich bis etwa 1 GHz, so daß die VHF-Bereiche und der UHF-Bereich gleichermaßen erfaßt werden.

Schaltung

Die Schaltung des VHF/UHF-TV-Modulators (siehe Bild 2) ist nicht übermäßig kompliziert. Die Grundfrequenz erzeugt der mit T1 aufgebaute quarzstabilisierte Oszillator; als Quarz ist praktisch jeder CB-Funk-Quarz (27, ... MHz) geeignet. Die Stufen mit T2 und T3 formen aus dem Oszillatorsignal das gewünschte obertonreiche Nadelimpulssignal. Die Modulation des Trägersignals geschieht mit Diode D1. Über verschiedene Widerstände erhält diese Diode eine Vorspannung. Die Höhe der Vorspannung ist so bemessen, daß die Diode dann ständig leitet, wenn das Video-

signal null ist (diesen Wert nimmt das Videosignal bekanntlich während des Synchronisationsimpulses an). Steigt die Amplitude des Videosignals, so leitet die Diode nicht mehr während der gesamten Trägersignalperiode, sondern nur solange die Amplitude des Trägersignals höher als die Amplitude des Videosignals zuzüglich der Diodendurchlaßspannung ist. Als Resultat ergibt sich eine um so niedrigere Amplitude des Ausgangssignals, je höher die Amplitude des Videosignals ansteigt: Die Amplitudenmodulation ist negativ. Für die Einstellung der Modulation sind zwei Trimpoties vorhanden. Mit P1 läßt sich die Vorspannung der Modulator-diode ändern. Von diesem Poti hängt der Spannungswert ab, den das Videosignal annehmen muß, damit Diode D1 sperrt. P1 beeinflusst folglich den Weißwert des Modulators. Die Amplitude des modulierenden Videosignals kann dagegen mit P2 eingestellt werden; von diesem Poti ist der Modulationsgrad abhängig. Die beiden Potis sind mit dem



Helligkeits- und dem Kontrasteinsteller eines TV-Geräts vergleichbar; allerdings wirkt sich eine Änderung des Weißwerts (P1) auch auf den Kontrast (P2) aus und umgekehrt.

Das vom TV-Modulator erzeugte Signal unterscheidet sich auch noch in anderer Hinsicht vom abgestrahlten Signal eines Fernsehsenders. Bei letzterem muß zur Reduzierung der Übertragungsbandbreite eins der beiden Seitenbänder unterdrückt werden. Diese Maßnahme kann beim TV-Modulator entfallen, da ja eine genügende Anzahl freier Kanäle zur Verfügung steht. In Bild 3 ist zum Vergleich das Frequenzspektrum eines Fernsehsenders (a) und des TV-Modulators (b) dargestellt. Da vom TV-Modulator beide Seitenbänder auf die Antennenbuchse des Empfängers gelangen, kann dieser versehentlich auch auf das linke (falsche) Seitenband abgestimmt werden. Ein einwandfreies Bild ist jedoch nur bei Abstimmung auf das rechte Seitenband zu erwarten.

Bau und Einstellung

Für den VHF/UHF-TV-Modulator wurde eine Platine entworfen; Bild 4 zeigt die Kupfer- und die Bestückungsseite. Um unerwünschte kapazitive Kopplungen zu verhindern, sind auf der Platine verschiedene an Masse liegende Abschirmflächen vorhanden. Außerdem wurde das Layout so gestaltet, daß zwischen Oszillator und Modulator ein auf der

Bild 2. Schaltung des VHF/UHF-TV-Modulators. Die Grundfrequenz 27 MHz erzeugt ein stabiler Quarzoszillator; das Oszillatorsignal wird so geformt, daß Harmonische bis etwa 1 GHz entstehen.

Platine stehendes Abschirmblech angebracht werden kann. Die Stromversorgung des VHF/UHF-TV-Modulators kann mit einer unstabilierten Speisespannung (12 ... 15 V) oder mit einer stabilisierten Speisespannung (5 V) erfolgen. Im letzten Fall entfällt Stabilisator IC1; die Lötunkte für Eingang und Ausgang von IC1 sind dann zu überbrücken. Um die beiden Trimpoties richtig einzustellen, bedarf es keiner komplizierten Kunstgriffe. Zuerst ist der Modulatorausgang mit dem Antennen-eingang über Koaxkabel zu verbinden. Das Signal des TV-Modulators muß nun sowohl in den beiden VHF-Bereichen (Kanal 2 ... 4 bzw. 5 ... 12) als auch im UHF-Bereich (Kanal 21 ... 65) zu empfangen sein. In den VHF-Bereich fällt die 1. Oberwelle (2 · 27 MHz; Kanal 3); im UHF-Bereich liegen die Oberwellen im Abstand von jeweils 3 bis 4 Kanälen nebeneinander. Der Empfang des TV-Modulator-Signals macht sich durch vermindertes Rauschen (Schnee) und durch abnehmende Bildhelligkeit bemerkbar. Ist der Empfänger in etwa richtig abgestimmt, so kann dem TV-Modulator ein Videosignal zugeführt werden. Der

Stückliste

Widerstände:

R1 = 33 k
R2 = 22 k
R3, R9 = 470 Ω
R4 = 1 k
R5 = 220 Ω
R6 = 270 Ω
R7 = 150 Ω
R8 = 6k8
R10, R11 = 100 Ω
R12 = 1k5
R13 = 68 Ω
P1 = 2k5 (2k2) Trimpotentiometer
P2 = 1 k Trimpotentiometer

Kondensatoren:

C1 = 22 p
C7 = 33 p
C2 = 120 p
C3, C4, C5 = 8p2
C6 = 22 p
C8, C9 = 1 μ /25 V Tantal

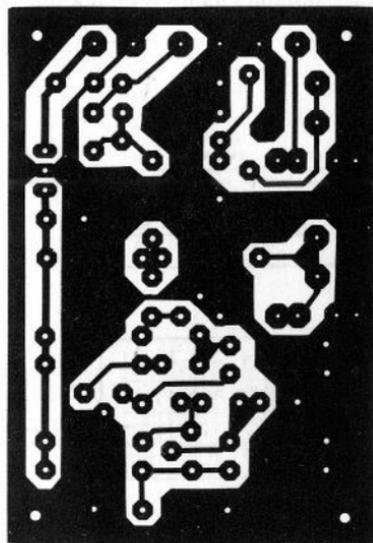
Halbleiter:

T1, T2 = BF 494
T3 = BFY 90
D1 = 1N4148
IC1 = 7805 (siehe Text)

Sonstiges:

L1 = 1 μ H
X1 = Quarz ca. 27 MHz

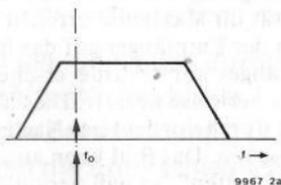
4



Videosignalgeber wird mit dem TV-Modulator ebenfalls über Koaxkabel verbunden.

Vor Beginn der Einstellung bringt man P1 am besten in Mittelstellung; der Schleifer von P2 wird auf Rechtsanschlag gedreht, sofern die Amplitude des

3a



3b

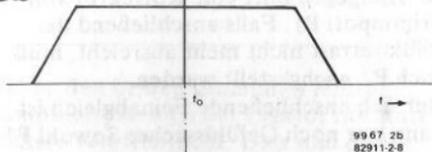
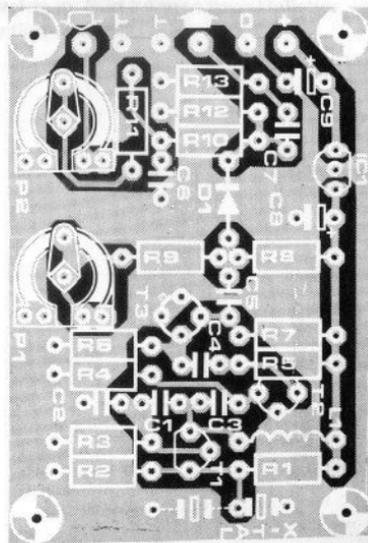


Bild 3. Frequenzspektrum eines Fernsehenders (a) und des TV-Modulators (b). Ein Seitenband des vom Fernsehenders ausgestrahlten Signals wird zur Reduzierung des Bandbreitenbedarfs unterdrückt; beim TV-Modulator ist dies nicht erforderlich.

Bild 4. Platine und Bestückungsplan für den VHF/UHF-TV-Modulator. Die an Masse liegenden Kupferflächen sorgen für eine Entkoppelung der einzelnen Stufen.



Videosignals $3 V_{ss}$ nicht überschreitet. Bei höheren Videospannungen kann P2 etwas zurückgenommen werden.

Liegt ein Videosignal am Modulator-eingang, so muß das Bild in irgendeiner (meistens verzerrten) Form auf dem Schirm sichtbar sein. Man stimmt nun den Empfänger so nach, daß die Bildqualität ihr Maximum erreicht.

Wenn der Empfänger auf das linke Seitenband abgestimmt wurde, erscheint das Bild möglicherweise negativ. Dies läßt sich leicht durch vorsichtiges Nachstimmen korrigieren. Das Bild kann auch vertikal "durchlaufen", so daß periodisch der schwarze Synchronisationsbalken sichtbar ist. Hiergegen hilft eine Korrektur von Trimpoti P1. Falls anschließend der Bildkontrast nicht mehr ausreicht, muß auch P2 nachgestellt werden.

Der sich anschließende Feinabgleich ist dann nur noch Gefühlssache: Sowohl P1

als auch P2 beeinflussen die Bildqualität. Falls das Bild während der Einstellung wieder negativ werden sollte, so ist dies nicht unbedingt auf eine fehlerhafte Abstimmung zurückzuführen. Auch eine Übersteuerung des Modulators kann für diesen Effekt verantwortlich sein. In diesem Fall genügt es, P2 geringfügig zurückzudrehen.

Auf zweierlei soll zum Schluß noch mit Nachdruck hingewiesen werden: Es ist unbedingt notwendig, die Platine des VHF/UHF-TV-Modulators in ein abschirmendes Gehäuse einzubauen, damit das HF-Signal nicht unkontrolliert abgestrahlt wird. Aus dem gleichen Grund ist dafür Sorge zu tragen, daß das HF-Signal nicht auf eine Antenne oder ein antennenähnliches Gebilde gelangt; wegen des extrem breiten Frequenzspektrums des HF-Signals wären nämlich die Folgen buchstäblich unübersehbar.

Kapitel 2:

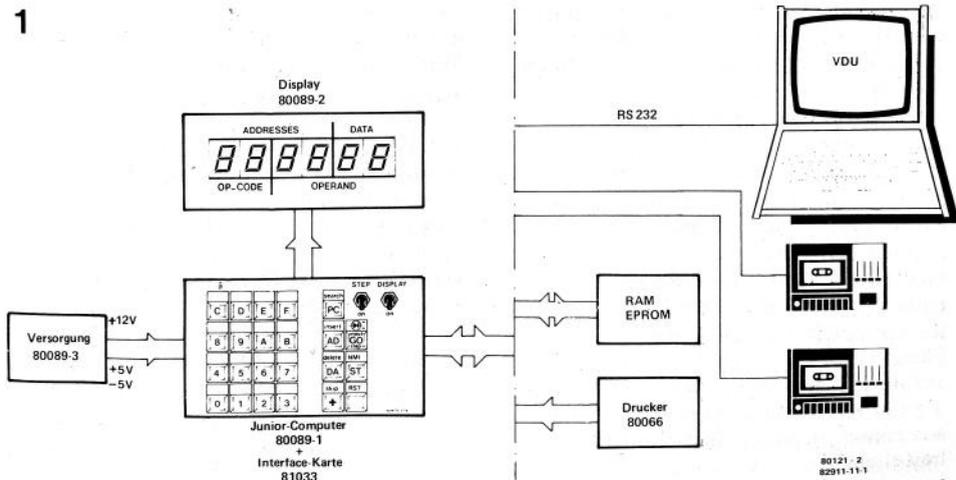
Bus-, μ P-Systeme, Speisungen

Der Selbstbau eines Mikrocomputers ist sicherlich ein lohnendes Vorhaben. Doch wer die Wahl hat, hat die Qual: Elektor veröffentlichte bisher fünf Selbstbau-Mikrocomputer-Systeme, so daß die Entscheidung schwerfällt. Dem Anfänger auf diesem Gebiet, dem die unterschiedlichen Konzeptionen der fünf Systeme weniger deutlich sind, werden hier die notwendigen Orientierungshilfen gegeben.

Die zwei Elektor-Mikrocomputersysteme die hier näher besprochen werden, tragen die Namen "SC/MP-System" und "Junior-Computer". Bevor die zwei Systeme einander gegenübergestellt werden, erscheint eine Randbemerkung notwendig: Dieses Kapitel will nur die unterschiedlichen Eigenschaften der Systeme aufzeigen, nicht jedoch einen generellen Überblick über die Einsatzmöglichkeiten von Mikrocomputern geben. Wenn bei der Beschreibung der einzelnen Systeme Anwendungen genannt sind, so bedeutet das natürlich nicht, daß das betreffende System nicht auch für einen anderen Zweck geeignet ist.

Neben den beiden genannten Mikrocomputersystemen hat Elektor noch drei weitere veröffentlicht. Dies sind der TV-Spielcomputer, der Intelktor (ein 16-bit-Schachcomputer) und die DCF-Computerschaltuhr. Bereits aus den Namen der Systeme geht hervor, daß sie eine bestimmte Funktion erfüllen und nicht universell einsetzbar sind. Deshalb sind sie auch im folgenden Kapitel nicht weiter erwähnt.

Bild 1. Links von der vertikalen Trennungslinie ist das Basissystem des Junior-Computers dargestellt. Die Erweiterungen rechts im Bild können nach Belieben hinzugefügt werden.



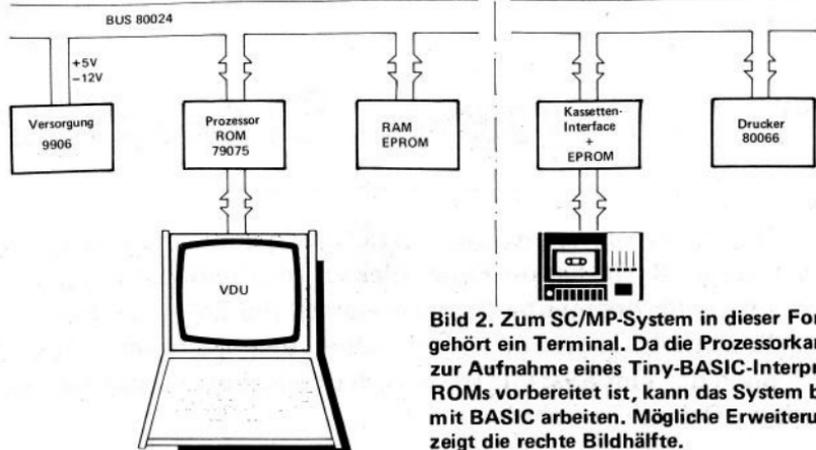


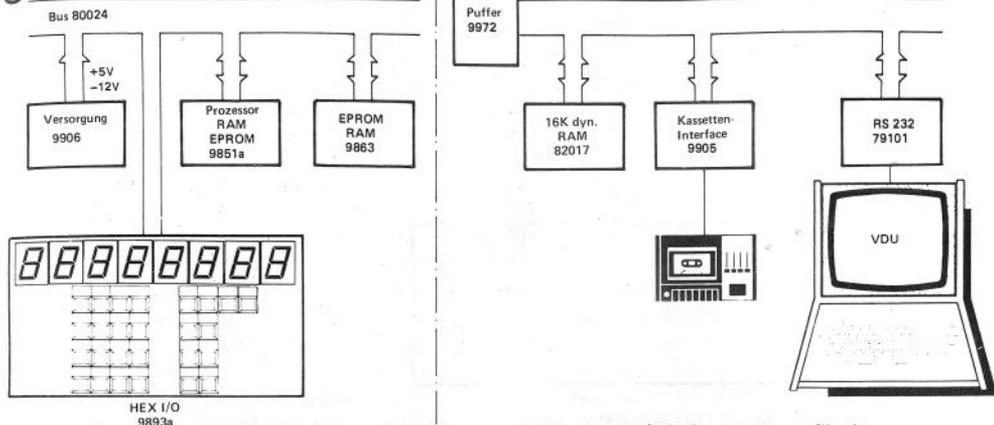
Bild 2. Zum SC/MP-System in dieser Form gehört ein Terminal. Da die Prozessorkarte zur Aufnahme eines Tiny-BASIC-Interpreter-ROMs vorbereitet ist, kann das System bereits mit BASIC arbeiten. Mögliche Erweiterungen zeigt die rechte Bildhälfte.

Das erste Elektor-Mikrocomputersystem ist das SC/MP-System. Dieser modular aufgebaute Mikrocomputer verdankt seinen Namen dem verwendeten Prozessor: SC/MP (sprich: Skemp) bedeutet "Simple to use, Cost effective Micro-Processor"; diese Abkürzung wurde dem Prozessortyp INS 8060 von seinem Hersteller (National Semiconductor) mit auf den Weg gegeben.

Die wichtigste Eigenschaft des SC/MP-Systems ist ein modularer Aufbau. Zur Hardware dieses Systems gehören eine Reihe von Platinen im Euroformat (10 cm x 16 cm); sie sind miteinander über einen gemeinsamen Bus verbunden. Ein Bus ist bekanntlich nichts anderes als ein Bündel von Verbindungsleitungen, die sich beim SC/MP-System ebenfalls in Form von Kupferbahnen auf Platinen befinden. Über den Bus werden die Kontakteleisten der einzelnen Systemplatinen parallelgeschaltet. Der modulare Aufbau verhilft dem SC/MP-System zu einer hohen Hardware-Flexibilität. In seiner niedrigsten Ausbaustufe besteht das System aus zwei Eurokarten. Erweitern läßt es sich einfach durch Einstecken von zusätzlichen Karten in den Bus. Nicht nur die Speicherkapazität (RAM und ROM) kann auf diese Weise erhöht werden, sondern die Erweiterung kann beispielsweise auch aus einem Interface für einen Drucker bestehen.

Der Junior besteht in seiner einfachsten Form aus nur einer einzigen Platine (abgesehen von der Speisung und dem Display). Eine zweite Platine, die Interfacekarte, macht den Junior-Computer zu einem vielseitig verwendbaren Mikrocomputer. Über eine Kontakteleiste kann die erweiterte Version mit einem Bussystem verbunden werden, daß auch SC/MP-kompatibel ist. In vielen Fällen sind deshalb die Erweiterungskarten für beide Systeme verwendbar. Für universelle Anwendungen sind sowohl das SC/MP-System als auch der Junior-Computer geeignet. Diese beiden Mikrocomputer können nicht nur "ernsthafte" Aufgaben wie zum Beispiel das Steuern einer Heizungsanlage übernehmen, sondern sie sind für allerlei Spiele ebenso gut geeignet. Daneben verfügen beide Systeme über Einrichtungen, die bei der Entwicklung von Programmen Hilfestellung leisten. Auch höhere Programmiersprachen gehören bei ihnen zum Bereich des Möglichen; für das SC/MP-System und für den Junior-Computer ist beispielsweise ein BASIC-Interpreter lieferbar. Beim SC/MP-System werden alle Daten, Befehle usw. über ein Terminal eingegeben. Dieses besteht aus einer schreibmaschinenähnlichen Tastatur und einem TV-Gerät oder einem Drucker. Um mit dem SC/MP-System sinnvoll arbeiten zu können, ist das Terminal unent-

3



behrlich. Geeignet ist unter anderem das in diesem Buch beschriebene "Elekterterminal".

Ein wichtiger Vorteil des Terminals ist die von ihm geschaffene Möglichkeit, mit dem Computer auf einfache Weise in einer höheren Programmiersprache kommunizieren zu können. Die elementaren Bausteine einer solchen Sprache sind nämlich nicht nur Ziffern, sondern auch Buchstaben und Sonderzeichen; eine alfanumerische Tastatur als Eingabe-Einheit ist deshalb unentbehrlich.

Der einfache Junior-Computer ist in seiner Grundform (ohne SC/MP-Systemkomponenten) relativ einfach ausgestattet, was sich natürlich auf die Anschaffungskosten sehr positiv auswirkt. Hier dienen 23 auf der Platine befindliche Taster als Eingabe-Einheit; das Display besteht aus 6 Sieben-Segment-Anzeigen. Die erweiterte Version kann ebenfalls an das Elekterterminal angeschlossen werden.

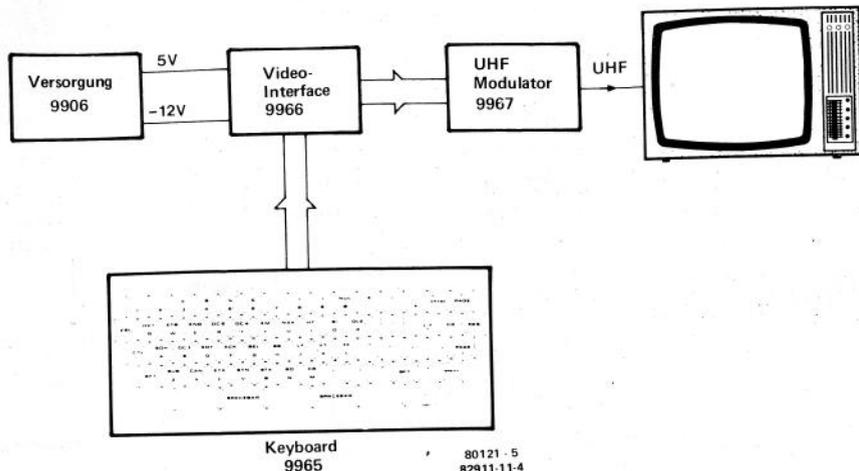
Mikroprozessoren

Wenn es um die Leistungsfähigkeit der einzelnen Mikroprozessor-Typen geht, erhitzen sich bisweilen die Gemüter. Der Prozessor ist schließlich das Herz des Computers; seine Eigenschaften bestimmen weitgehend die Technik des Programmierens (solange keine höhere Programmiersprache in den Computer

Bild 3. Das SC/MP-System kann für den Anfang auch mit einer hexadezimalen Ein- und Ausgabe aufgebaut werden. Erweiterungen sind hier ebenfalls rechts im Bild dargestellt.

"eingebaut" wird) sowie die Arbeitsgeschwindigkeit des Systems. Zunächst ist man geneigt anzunehmen, daß der Prozessor den Vorzug verdient, dessen Befehlsliste den größten Umfang hat, und der mit der höchsten Geschwindigkeit arbeitet. Dies ist jedoch nicht ganz richtig. Ein Prozessor mit einem Befehlsatz von mehreren hundert Befehlen läßt sich weniger leicht handhaben als ein anderer Typ mit übersichtlicherem Befehlsatz. Die Praxis hat gezeigt, daß der Programmierer möglichst alle Befehle und ihre Wirkungen im Gedächtnis haben muß, und dies ist natürlich umso schwieriger, je länger die Liste der Befehle ist. Eine höhere Geschwindigkeit des Mikroprozessors bringt nur dann uneingeschränkt Vorteile, wenn nicht gleichzeitig auch schnellere (und damit teure!) Speicherbausteine erforderlich sind. Die Bedeutung der Arbeitsgeschwindigkeit wird manchmal überschätzt; sie spielt in der Praxis erst bei Anwendung höherer Programmiersprachen oder bei sehr komplexen mathematischen Operationen eine Rolle.

Grundsätzlich ist jedoch ein schnellerer Prozessor einem langsamen vorzuziehen. Ein anderer wichtiger (wenn nicht sogar der wichtigste) Gesichtspunkt



für die Auswahl eines Prozessortyps ist die Menge der verfügbaren Software. Im allgemeinen arbeiten Programme eines fremden Systems nach kleinen Änderungen auch auf dem eigenen, sofern beide Systeme mit dem gleichen Prozessor bestückt sind. In dieser Hinsicht schneidet der 6502 am besten ab; hierzu trug wesentlich die Popularität des Mikrocomputers "KIM" bei. Doch auch andere mit dem 6502 arbeitende Mikrocomputer sind weit verbreitet. So kommt es, daß die Anzahl der allgemein zugänglichen 6502-Programme mit Abstand an der Spitze liegt.

Doch zurück zu den Elektor-Mikrocomputersystemen: Beide Systeme arbeiten mit anderen Prozessoren. Im SC/MP-System befindet sich der INS 8060 von National und im Junior-Computer der 6502 von Rockwell. Der SC/MP (INS 8060) ist der einfachste und langsamste, während der 6502 die meisten Möglichkeiten bietet und mit der höchsten Geschwindigkeit arbeitet.

Die relativ niedrige Arbeitsgeschwindigkeit des SC/MP hat dazu geführt, daß für das Elektor-SC/MP-System auch Prozessor-karten angeboten werden, die den Prozessortyp Z-80 enthalten. Die Entwicklung der Z-80-Karte geschah jedoch nicht auf Initiative von Elektor. Zum Abschluß wird nachfolgend der Hardware-Aufbau beider Elektor-Systeme kurz umrissen; ferner werden die ver-

Bild 4. Das Elektroterminal benutzt einen gewöhnlichen TV-Empfänger als Datensichtgerät. Auf seinen Schirm werden 16 Zeilen mit jeweils 64 Zeichen geschrieben.

schiedenen Ausbaumöglichkeiten genannt. Da das an dieser Stelle nur in Kurzform möglich ist, sind weiterführende Informationen den Büchern zu entnehmen, die das betreffende System beschreiben.

Junior-Computer

Der Junior-Computer gehört zu den sogenannten Einplatinen-Computern. Auch er benötigt deshalb eine Speisung, die hier die Spannungen +5 V, -5 V und +12 V liefern muß (siehe Bild 1). Das Tastenfeld für die Eingabe befindet sich auf der Platine. Informationen des Computers erscheinen auf 6 Sieben-Segment-Anzeigen. Das Display sitzt auf einer kleinen Hilfsplatine, die für eine bequeme Ablesung schräg auf die Hauptplatine montiert wird. Hauptplatine, Displayplatine und Speisung bilden das kleinste mögliche System. Die Erweiterung mit einer Interfacekarte für zwei Kassettenrekorder, einem Drucker, einer zusätzlichen I/O und Speicher ist leicht möglich. Auf der Hauptplatine des Junior-Computers ist eine Kontaktleiste angebracht, die die Verbindung zum Bus herstellt. Dies ist beispielsweise für die Erweiterung der RAM- oder ROM-Kapazität wichtig. Auf

der Junior-Computer-Platine befindet sich ein 1-KByte-RAM und ein 1-KByte-EPROM; letzteres enthält das Monitor-Programm.

Der Monitor des Junior-Computers ist recht komfortabel ausgestattet. Ein sogenannter Hex-Assembler ermöglicht die Verwendung von symbolischen Adressen bei Sprungbefehlen. Der Computer berechnet dann die absoluten Adressen selbst.

Will man in Assembler oder mit höheren Programmiersprachen arbeiten, so ist ein Terminal notwendig. Dieses läßt sich ebenfalls über die Kassetteninterface-Platine anschließen.

Hier noch einmal das Wichtigste zusammengefaßt: Das Minimalsystem des Junior-Computers wird im Hexadezimalcode (Ziffern 0 . . . F) programmiert. Für das Speichern von Programmen auf Band ist zusätzlich die Interface-Platine notwendig. Das Arbeiten mit Hilfsmitteln wie Editor, Assembler, Disassembler oder in höheren Programmiersprachen (zum Beispiel BASIC) erfordert zusätzliche EPROM-bzw. RAM-Kapazität.

Die Hard- und Software des kompletten Systems ist ausführlich in den Junior-Büchern 1 . . . 4 beschrieben.

SC/MP-System

Da das SC/MP-System modular aufgebaut ist, sind mehrere unterschiedliche Systemkonfigurationen möglich (siehe Bild 2).

Das Minimalsystem besteht aus zwei Karten: Der Prozessorkarte und der 8-K-EPROM-und-8-K-RAM-Karte. Auf der Prozessorkarte befindet sich auch ein Adreß- und Datenbuspuffer sowie ein RS-232-Interface für den Anschluß eines Terminals. Die 8-K-EPROM-und-8-K-RAM-Karte nimmt das in EPROMs gespeicherte Monitorprogramm und das RAM (den Arbeitsspeicher) auf. Das RAM kann ganz nach Wunsch von 1 K bis auf 8 K ausgebaut werden.

Das SC/MP-System läßt sich in dieser Form nur über ein Terminal bedienen. Auf dem aus zwei Karten bestehenden Minimalsystem können bereits Programme laufen, die in BASIC abgefaßt sind. Die Prozessorkarte ist nämlich mit

einer IC-Fassung ausgerüstet, in die das BASIC-Interpreter-ROM nur eingesteckt zu werden braucht. Auf Band lassen sich Programme aufzeichnen, wenn eine Kassetteninterface-Platine hinzugefügt wird. Mit Hilfe der Metallfoliendrucker-Interface-Karte können schließlich auch Listings von Daten oder Programmen ausgedruckt werden.

Die für das SC/MP-System benötigten Speisespannungen hängen von der Wahl der EPROMs ab. Die auf der 8-K-EPROM-und-8-K-RAM-Karte befindlichen EPROMs vom Typ 2716 arbeiten mit einer einzigen Spannung von +5 V. An Stelle des Typs 2716 kann auch der 2708 verwendet werden; dann halbiert sich allerdings nicht nur die EPROM-Kapazität der Karte, es sind zusätzlich zur Speisespannung +5 V die Spannungen +12 V und -5 V notwendig. Die EPROMs des Typs 5204 auf der Kassetteninterface-Karte machen zwei Speisespannungen erforderlich: +5 V und -12 V. Für das System ist eine Speisung vorhanden, die sich - abgesehen vom Netztrafo - auf der Busplatine befindet. Diese Speisung liefert die beiden auf dem SC/MP-Bus verfügbaren Speisespannungen +5 V und -12 V.

Außer den bis jetzt genannten Moduln sind auch noch einige Karten aus der Aufbauphase des SC/MP-Systems erhältlich. Ein kleineres SC/MP-System kommuniziert mit der Außenwelt über ein hexadezimalles Tastenfeld und 8 Sieben-Segment-Anzeigen (siehe Bild 3). Die Struktur dieses Systems zeigt weitgehende Ähnlichkeit mit dem Aufbau des Junior-Computers in seiner einfachsten Form. Aus den älteren Karten läßt sich ein System wie folgt zusammenstellen: Die Prozessorkarte und die Erweiterungskarte bilden den eigentlichen Computer. Die Datenleitungen sind nicht gepuffert, so daß ein Ausbau des Systems nur in Grenzen möglich ist. Plant man umfangreichere Erweiterungen, so kann eine Datenbuspuffer-Platine hinzugefügt werden. Das in EPROMs abgelegte Monitor-Programm umfaßt 1,5 KByte, während dem Benutzer ein 1-KByte-RAM als Arbeitsspeicher zur Verfügung

steht. Eine Erweiterung der RAM-Kapazität ist mit der RAM/EPROM-Karte oder der 16 K dynamischen RAM-Karte möglich.

Die Ein- und Ausgabe von Informationen erfolgt mit 26 Drucktasten und 8 Sieben-Segment-Anzeigen auf der Hex-I/O-Platine. Für das Aufzeichnen von Programmen auf Magnetband ist noch eine kleine Hilfsplatine notwendig. An dieses System läßt sich auch ein Terminal anschließen. In diesem Fall wird eine weitere kleine Hilfsplatine (RS-232-Interface) benötigt, während die Hex-I/O-Platine entfällt.

Bild 3 zeigt schematisch den Aufbau des Systems, das übrigens auch in dem Buch "µComputer für den Selbstbau, Band 1" beschrieben ist.

Aus Bild 4 ist schließlich ersichtlich, welche Moduln zum Elektrotterminal gehören. Eine genaue Beschreibung des

Terminals enthält auch der zweite Band "µComputer für den Selbstbau".

Universelle Erweiterungen

Es ist nicht selten, daß Computer-Freaks zwei Systeme zu Hause haben. Nun stehen sie bei jeder Erweiterung vor der Frage, ob sie die Erweiterung systemspezifisch oder universell aufbauen. Elektor hat sich jedenfalls für die zweite Möglichkeit entschieden. Die in diesem Buch veröffentlichten Schaltungen sind dann auch hardwaremäßig so konzipiert, daß sie in der Regel zu jedem Computersystem passen. Die Kompatibilität ist also keineswegs auf die beiden beschriebenen Systeme beschränkt. Es ist jedoch unumgänglich, daß verschiedentlich geringe Modifikationen notwendig sind, auf die dann hingewiesen wird.

Bussystem und Busplatine

Jedes modular aufgebaute Mikrocomputersystem benötigt ein sogenanntes Bussystem. Es vereinfacht den Computeraufbau enorm. Auch die Computersysteme von Elektor arbeiten mit einem solchen Bussystem. Die Ausführung hiervon ist so universell, daß die Busplatine auch für Computer mit unterschiedlichen Mikroprozessoren geeignet ist.

Alle Verbindungen in einem µP-System werden über den Systembus geführt. Bei einem modular aufgebauten Mikrocomputer besteht ein Systembus in der Regel aus einer länglichen Platine, auf der eine ganze Anzahl von Leiterbahnen parallel verlaufen. Eine solche Busplatine bildet häufig die Rückwand eines Systems und wird nicht zuletzt auch deshalb "backplane" genannt. Eine zutreffende Bezeichnung, wenn man bedenkt, daß alle Systemplatinen mit der meistens im hinteren Gehäuseteil montierten Busplatine untereinander verbunden sind.

Auch die Elektor-Computersysteme sind mit einer solchen Busplatine aufgebaut. Sie kann maximal fünf andere Systemplatinen (auch Steckkarten genannt)

aufnehmen.

Die Kontaktreihen rechts und links auf der Busplatine sind gespiegelt (siehe Bild 1). Das heißt: Will man mehrere Busplatinen hintereinanderschalten, liegen sich immer die Kontaktreihen gegenüber. Dadurch kann man die Busplatinen kreuzungsfrei mit 64 Drahtbrücken elektrisch miteinander verbinden. Wegen der Spiegelung der linken Kontaktreihe kann man dort allerdings keine Steckerleiste anbringen. Die Belegung der Anschlußstifte stimmt nicht!

Busdefinition

Voraussetzung für ein universell verwendbares Bussystem ist, daß vorher getroffene

Speisung für kleine und große μ P-Systeme

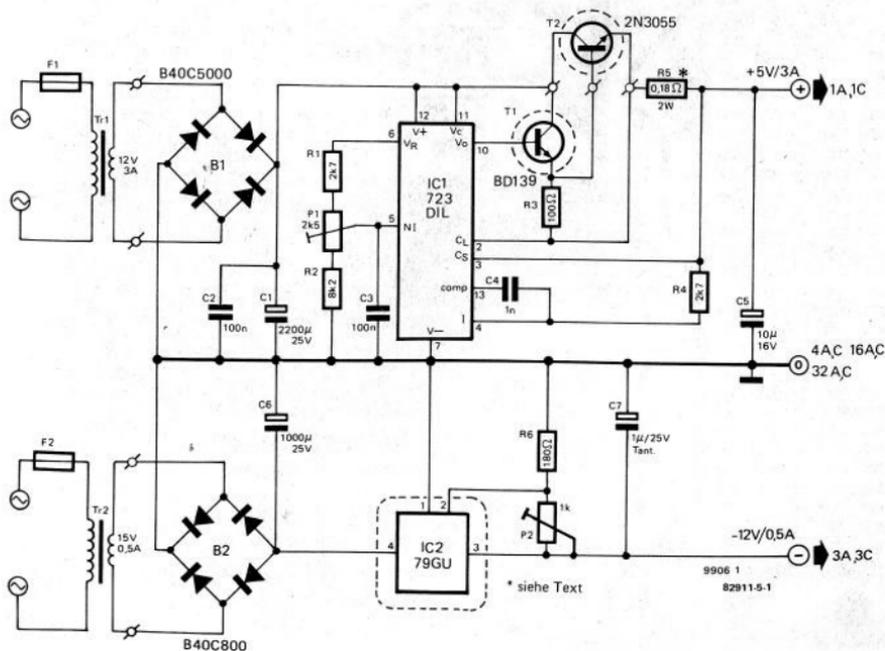
Für kleinere Computersysteme ist die hier beschriebene Speisung die Lösung. Die zur Schaltung gehörende Platine nimmt nicht nur die Stabilisierungsschaltung auf, sondern hat auch noch Platz für zwei 64-polige Konnektoren. Dadurch ist es möglich, daß die Platine zusammen mit zwei Eurokarten ein kompaktes Modul bildet.

Da die Möglichkeit besteht, die Platine mit einer oder mehreren Bus-Platinen zu verbinden, können auch umfangreichere Systeme mit dieser Speisung versorgt werden.

Die hier vorgestellte Schaltung ist so universell, daß sie für viele Systeme die Stromversorgung übernehmen kann. Die Ausgangsspannungen der meisten 5 V-IC-Stabilisatoren sind so eng toleriert, daß sie in der Regel auch den Anforderungen an die Einhaltung der vorgeschriebenen Speisespannung bei μ P-ICs genügen. Trotzdem kann mit einem "Worst Case"-Stabilisator (Exemplar, dessen Werte an der Grenze des Toleranzbereichs liegen) die Spannung an den

μ P-IC-Anschlüssen zu gering sein. Dies hat seine Ursache darin, daß bei den nicht unbedeutenden Strömen, die das Gesamtsystem aufnimmt, durchaus ins Gewicht fallende Spannungsverluste durch Leitungen und Platinbahnen entstehen. Es ist deshalb sehr von Vorteil, wenn sich die Speisespannung innerhalb eines bestimmten Bereichs um den Sollwert

Bild 1. Schaltung der kompletten Speise-Einheit für μ P-Systeme.



einstellen läßt, so daß sich solche Verluste ausgleichen lassen. Bei der Einstellung ist es wichtig darauf zu achten, das der maximale Wert für μP -ICs von 5,25 V nicht überschritten wird.

Der Strombedarf in einem Mikrocomputersystem ist je nach Ausbaustufe recht unterschiedlich. In jedem Fall muß die +5 V Versorgung den meisten Strom liefern. Dabei sind Ströme zwischen 3 A und 5 A keine Seltenheit.

Neben der bereits erwähnten Spannung von +5 V ist bei μP -Systeme auch eine -12 V Spannung häufig erforderlich.

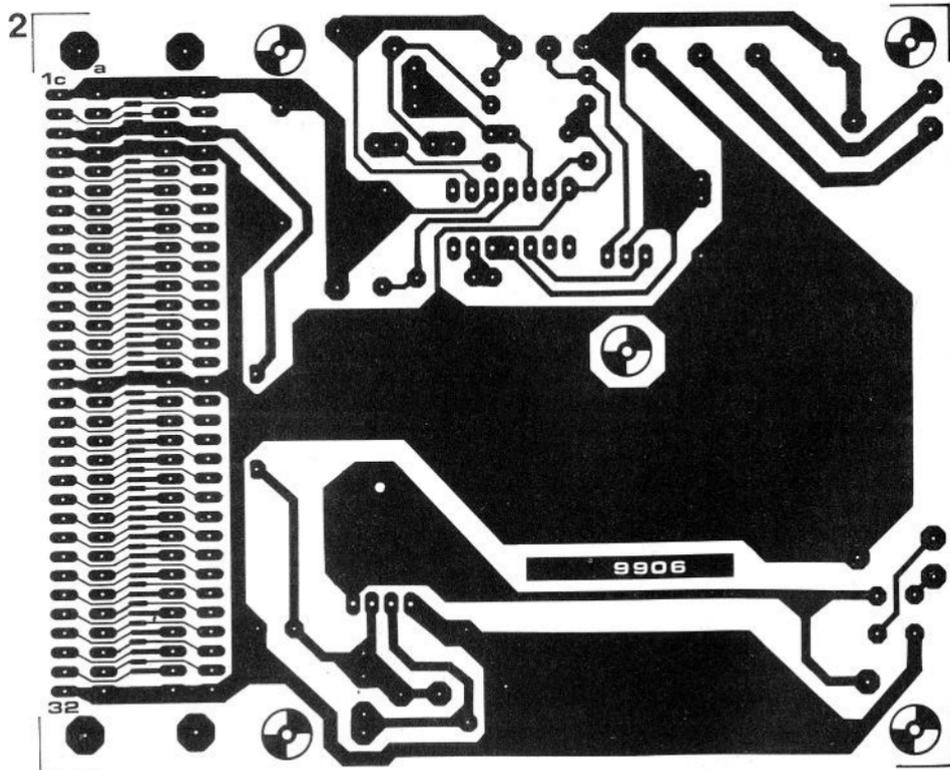
Schaltung

Eine Speisung, die die genannten Forderungen erfüllt, ist in Bild 1 angegeben. Für die Stabilisierung der 5 V-Spannung dient ein 723 (IC1); diese Spannung läßt sich mit Poti P1 zwischen 4,5 V und 5,5 V einstellen. Die Leistungstransistoren T1 und T2 sorgen dafür, daß der benötigte

hohe Strom entnommen werden kann. Die Strombegrenzung hängt von Widerstand R5 ab, sie setzt bei ca. 3 A ein. Für höhere Ausgangsströme kann der Wert von R5 auf 0,1 Ω /4 W herabgesetzt werden; der maximale Strom beträgt dann ca. 6 A. Natürlich müssen auch der Transformator, der Brückengleichrichter und der Siebelko C1 entsprechend angepaßt werden.

Die negative Spannung von -12 V liefert ein einstellbarer IC-Stabilisator vom Typ 79 G (IC2). Hier beträgt der maximale Strom 0,5 A, was für die meisten Systeme mehr als ausreicht. Die von IC2 stabilisierte Spannung kann mit Poti P2 exakt auf -12 V eingestellt werden.

In der Schaltung ist für die negative Spannung ein eigener Netztrafo vorgesehen. Natürlich läßt sich statt dessen auch ein gemeinsamer Trafo mit zwei getrennten Wicklungen verwenden. Soll die +5 V Spannung einen Strom von 5 A liefern können, muß der Trafo ca.



5,5 A abgeben können. Der Siebelko C1 wird auf $1000\mu \dots 4700\mu$ erhöht. Der Strombegrenzungswiderstand R5 erhält den Wert $0,12 \Omega/5 \text{ W}$. Um bei höheren Ausgangsströmen die Belastung des

Längstransistors möglichst gering zu halten, empfiehlt sich ein Paralleltransistor. Aus der Tabelle läßt sich die unterschiedliche Dimensionierung für 3 A und 5 A Ausgangsstrom entnehmen.

Stückliste zu Bild 1, 2 und 3

Widerstände:

R1, R4 = 2k7

R2 = 8k2

R3 = 100 Ω

R5 = 0,18 $\Omega/2 \text{ Watt}$ (siehe Text)

R6 = 180 Ω

P1 = 2k5

P2 = 1 k

Kondensatoren:

C1 = 2200 $\mu/25 \text{ V}$

C2, C3 = 100 n

C4 = 1 n

C5 = 10 $\mu/16 \text{ V}$

C6 = 1000 $\mu/25 \text{ V}$

C7 = 1 $\mu/25 \text{ V Tantal}$

Halbleiter:

IC1 = 723

IC2 = 79 G

T1 = BD 137, BD 139

T2 = 2N3055

B1 = B40C5000

B2 = B40C800

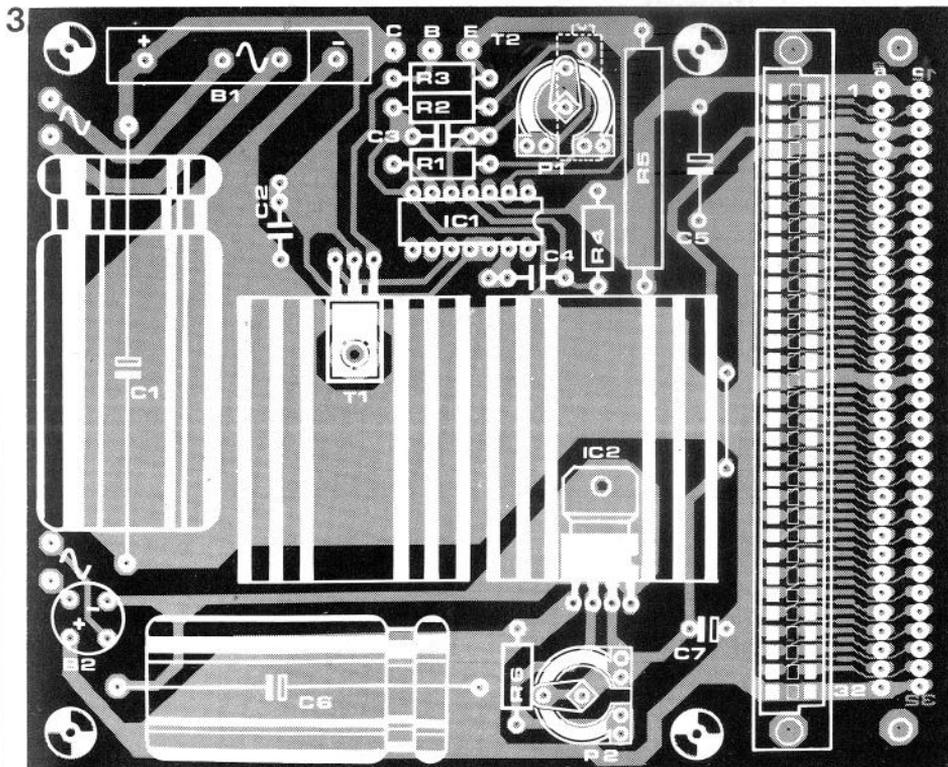
Sonstiges:

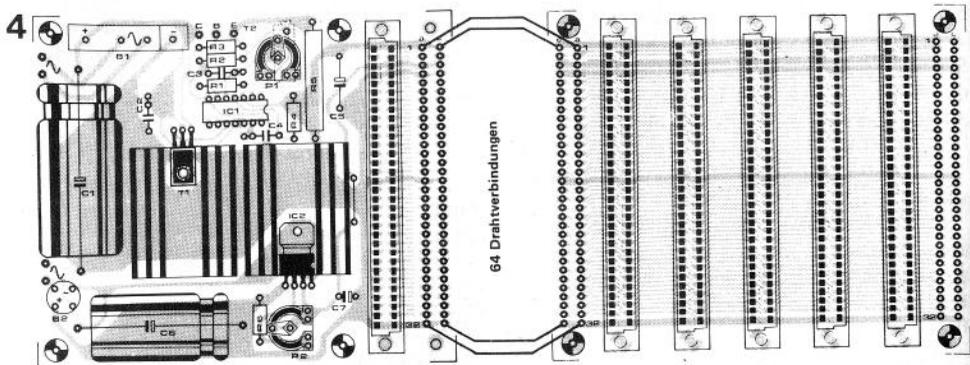
Tr1 = Trafo 12 V, 3 ... 4 A

Tr2 = Trafo 15 V, 0,5 A

Bild 2. Layout der Platine für die μP -Speise-Einheit.

Bild 3. Bestückungsplan für die Speiseplatine.





Platine und Aufbau

Auf der für die μP -System-Speisung entworfenen Platine (Bild 2 und 3) können mit Ausnahme von Leistungstransistor T2 und den Transformatoren sämtliche Bauteile untergebracht werden. Für T2 wird ein Kühlkörper mit einem maximalen thermischen Widerstand von $1,5^\circ\text{C/W}$ (bei $I_{\text{max}} = 3\text{ A}$) benötigt. Höhere Ströme machen eine stärkere Kühlung erforderlich. T1 und IC2 sind ebenfalls mit Kühlkörpern versehen; diese befinden sich jedoch auf der Platine.

Die Platine für die hier beschriebene Speisung weist die gleichen Abmessungen wie die Bus-Platine auf. Ferner läßt sich die Speise-Platine mit zwei zusätzlichen 64-poligen Kontaktleisten (Buchsenteil) ausrüsten, in die zum Beispiel eine weitere RAM-Karte gesteckt werden kann. In

Bild 4. Aus diesem Bild geht hervor, wie die Speisung mit der Bus-Platine verbunden wird.

Bild 4 ist skizziert, wie Speise- und Bus-Platine miteinander verbunden werden. Für die Drahtverbindungen verwendet man am besten flexible Litze; der Querschnitt der Speiseleitungen sollte dabei so groß wie möglich gewählt werden.

Die in Bild 4 skizzierte Anordnung bietet Platz für sechs Karten im Europa-Format. Außerdem ist ersichtlich, wo welche Verbindungen herzustellen sind. Für die Verbindungen sollte man vorzugsweise flexibles Kabel verwenden. Damit die Spannungsverluste in den Versorgungszuleitungen sehr klein bleiben, muß der Leitungsquerschnitt möglichst groß sein.

	$U_{\text{Tr.}}$ sek.	C1	T2	R5
3 A	12 V/ 3 A	2200 μ / 25 V	wie in Schaltbild 1 gezeigt	0,18 Ω /2 W
5 A	12 V/ 5,5 A	10000 μ 25 V		0,12 Ω / 5 W

μ C-Schaltnetzteil

Nicht immer sind in einem Mikrocomputersystem die Versorgungsspannungen von +5 V und -12 V ausreichend. Da in der Regel auch analoge Bauelemente zu einem solchen System gehören, ist auch eine +12 V-Speisung wünschenswert.

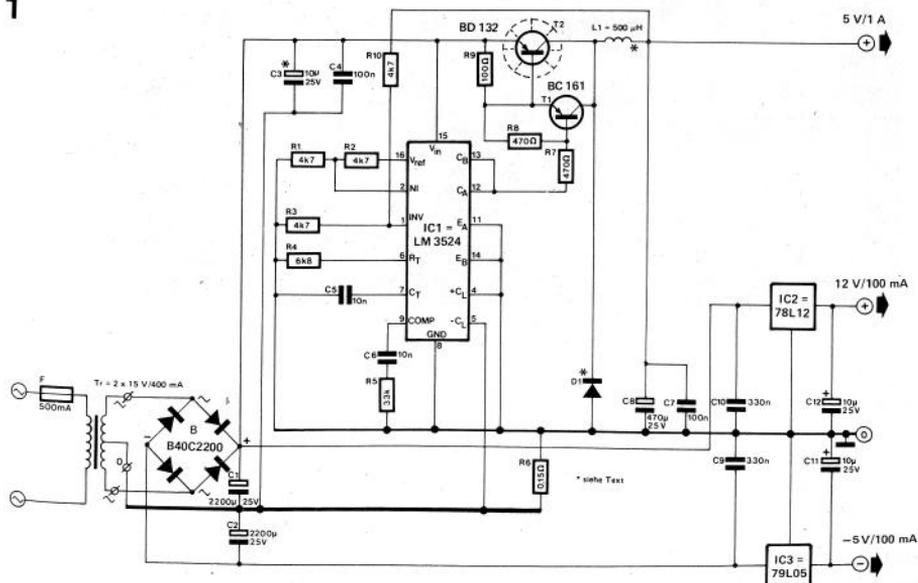
Außerdem benötigen einige Speicherbausteine auch eine -5 V Versorgungsspannung. Die gerade beschriebene Schaltung ist nur für normale Systeme mit +5 V und -12 V Versorgungsspannung geeignet. Die nachfolgende Schaltung liefert neben der +12 V-Speisung auch noch eine Spannung von maximal +5 V/1 A. Das μ C-Schaltnetzteil eignet sich von daher als separate Versorgungseinheit für eine zusätzliche Busplatine, die dann alle Versorgungsspannungen u.a. für die 16 K dynamische RAM-Karte aufweist.

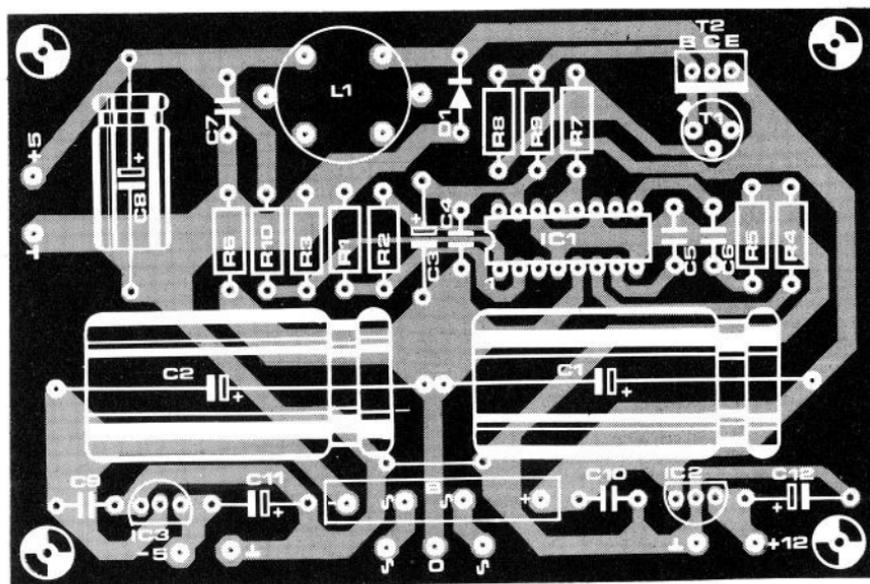
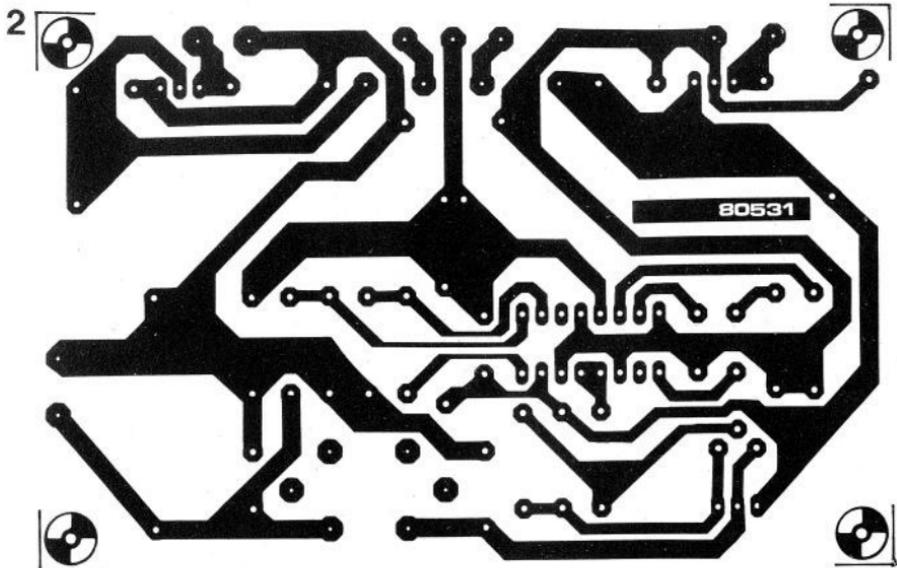
Das Schaltnetzteil (Bild 1) geht so sparsam mit der Energie um, daß unter normalen Betriebsbedingungen kein Bauelement gekühlt zu werden braucht. Obwohl das Netzteil bei einer Ausgangsspannung von +5 V einen Strom von 1 A liefern kann,

wird nur ein Trafo 2 x 15 V/0,4 A benötigt. Die Schaltfrequenz liegt knapp oberhalb der Hörgrenze; sie beträgt ungefähr 20 kHz. Die Regelung kommt dadurch zustande, daß IC1 die Einschaltdauer von T2 an den Belastungszustand anpaßt. Ausführlich wurden diese Zusammenhänge im Artikel "Schaltnetzteile" dargestellt, der im Januar 1980 in Elektor erschien.

Der Pulsbreitenmodulator 3524 wird von einer ganzen Reihe von Herstellern gefertigt (Texas Instruments, National Semiconductor, Silicion General u. a.), so daß er ohne Schwierigkeiten erhältlich sein dürfte. Schalttransistor T2 ist ein etwas schnellerer Typ als in herkömmlichen Netzteilen üblich; seine Eigenschaften sind jedoch wenig kritisch. Dies gilt auch für Diode D1. Die durch T2 und D1 fließenden Ströme bleiben bei normalen Betriebsbedingungen unter 2 A.

1





Die Drossel kann leicht selbst auf einen Schalenkern mit Luftspalt gewickelt werden, der für die Schaltfrequenz von 20 kHz geeignet ist. Für Testaufbauten des Schaltnetzteils wurde der Schalenkern-Typ B 65651-N250-A28 von Siemens verwendet; sein Durchmesser beträgt 18 mm, seine Höhe 11 mm. Aus dem A_L -Wert des Kerns (beim genannten

Typ ist $A_L = 250 \cdot 10^{-9} \text{ H}$) läßt sich leicht die benötigte Windungszahl n errechnen:

$$n = \sqrt{\frac{L}{A_L}} = \sqrt{\frac{500 \cdot 10^{-6} \text{ H}}{250 \cdot 10^{-9} \text{ H}}} = 44,72$$

Bei einem A_L -Wert von $250 \cdot 10^{-9} \text{ H}$ sind daher 45 Windungen notwendig.

Stückliste

Widerstände:

R1 . . . R3, R10 = 4k7

R4 = 6k8

R5 = 33 k

R6 = 0,15 Ω

R7, R8 = 470 Ω

R9 = 100 Ω

Kondensatoren:

C1, C2 = 2200 μ /25 V

C3, C11, C12 = 10 μ /25 V

C4, C7 = 100 n

Den Drahtdurchmesser wählt man so, daß die Wicklung den vorhandenen Raum voll ausfüllt. Im erwähnten Fall betrug der Drahtdurchmesser 0,5 mm.

Das Schaltnetzteil ist mit Hilfe von R6 gegen Kurzschlüsse geschützt. Läßt sich ein 0,15- Ω -Widerstand nicht beschaffen, dann stellt man ihn aus einem Stück Widerstandsdraht selbst her. Wird der Ausgang kurzgeschlossen, so erwärmt

C5, C6 = 10 n
C8 = 470 μ /25 V
C9, C10 = 330 n

Halbleiter:

D1 = BYW32

T1 = BC 161

T2 = BD 132

IC1 = 3524

IC2 = 78L12

IC3 = 79L05

B = B40C2200

außerdem:

Tr = Trafo 2 x 15 V/0,4 A

L1 = Drossel 500 μ H (siehe Text)

sich T2 stark. Um allen unvorhergesehenen Umständen zu begegnen, sollte deshalb T2 doch mit einem Kühlkörper versehen werden.

Die Stabilisierungen der Spannungen +12 V und -5 V sind konventionell aufgebaut; diese Spannungen sind als Hilfsspannungen für das verbreitete EPROM 2708 und für dynamische RAMs (z.B. 4116) bestimmt.

Kapitel 3:

Speicher

Universelle RAM/EPROM-Karte

Die hier vorgestellte RAM/EPROM-Karte ist so universell aufgebaut, daß sie prinzipiell und ohne große Veränderungen bei einer ganzen Reihe von Mikroprozessorsystemen eingesetzt werden kann. In einigen Fällen sind einige kleine Änderungen im Steuerteil notwendig. Diese Änderungen sind für zwei Anwendungsfälle ausführlich beschrieben.

Auf der neuen Karte ist nicht nur für 8K RAM's Platz, sondern auch für 4, 8 oder 16K EPROM's.

Das populäre RAM-IC 2114 besitzt im Vergleich zum 2102 und 2112 eine viermal höhere Speicherkapazität. Mit dem 2114 läßt sich daher ein Schreib-Lese-Speicher doppelter Kapazität bei gleichzeitig halber IC-Anzahl aufbauen.

Die Wahl des EPROM-IC-Typs ist ebenfalls nicht schwer. Hier kommen wegen ihrer großen Verbreitung nur die ICs der 27XX-Serie in Betracht. Die Pin-Kompatibilität der Typen 2708, 2716 und 2732 erlaubt, auf der gleichen Karte wahlweise eine EPROM-Gesamtkapazität von 4 K, 8 K oder 16 K unterzubringen. Dazu sind zwei "Handgriffe" notwendig: die Beschaltung der Pins 19 und 21 (die Einstellung der richtigen Versorgungsspannung) und die Festlegung der Adreßdekodierung mit Hilfe von Drahtbrücken.

Die Speicherkarte ist selbstverständlich vollständig gepuffert, so daß die Belastung des Bussystems so niedrig wie möglich bleibt. Für den Adreßbus werden unidirektionale Puffer vom Typ 74LS241 verwendet. Diese ICs besitzen PNP-Eingänge und benötigen nur einen geringen Eingangsstrom. Die gleiche Eigenschaft haben die bidirektionalen Datenbuspuffer. Hier wird die Datenrichtung vom logischen Signal auf der gemeinsamen

Selektionsleitung bestimmt. Bei logisch 0 arbeiten die Datenbuspuffer im Schreibmodus, während eine logische 1 gleichbedeutend mit dem Lesemodus ist.

Solange die Speicherkarte nicht adressiert wird, bleiben die Datenbuspuffer ständig im Schreibmodus. Die Speicherkarte kann daher fremde Signale auf dem Datenbus nicht verfälschen. Beim Adressieren der Karte wird auf dem Lesemodus umgeschaltet, es sei denn, daß über N4 ein Schreibsignal eintrifft.

Die Drahtbrücken L-M und L-N ermöglichen es, die Speicherkarte an verschiedene Prozessorsysteme anzupassen. Die Drahtbrücke L-M ist beim SC/MP und dem 6502 erforderlich.

Anordnung der Speicher-Blöcke (Adreß-Organisation)

Die in dieser Schaltung gewählte Adreß-Organisation erlaubt eine große Flexibilität in der Verwendung der Speicher. Zunächst muß man wissen, daß IC5 den gesamten Adreßbereich in 4-KByte-Blöcke unterteilt, und daß N1 einen, zwei oder vier davon für die EPROMs und N2 zwei 4-KByte-Blöcke für die RAMs auswählt. Alle 4-K-Blöcke können im 64-K-Adreßbereich beliebig verteilt werden, wobei Doppelbelegungen selbstverständlich nicht möglich sind.

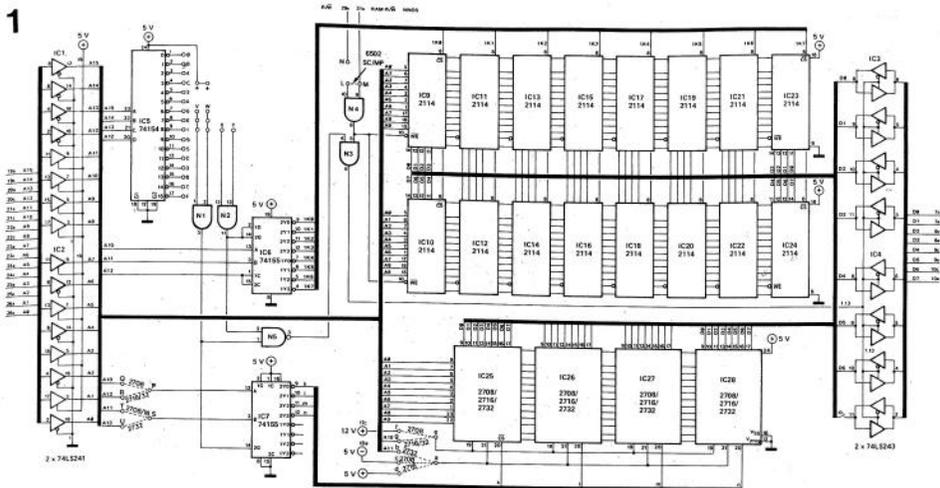


Bild 1. Schaltung der neuen RAM/EPROM-Karte. Im Bild ist angegeben, welche Verbindungen bei den einzelnen EPROM-Typen notwendig sind.

Über die Lage der 4-K-Blöcke im Adreßbereich bei der Verbindung bestimmter Ausgänge von IC5 mit den Eingängen der Gatter N1 und/oder N2 gibt Tabelle 3 Auskunft.

Dazu sind noch einige Erläuterungen notwendig:

RAM-Bereich

Die beiden 4-K-Blöcke werden von IC9 ... IC16 und IC17 ... IC24 so gebildet, daß einer der beiden Blöcke auf eine geradzahlige (0, 2, 4 ...) und der andere auf eine ungeradzahlige Seite gelegt wird. So ergibt sich zum Beispiel durch die Wahl von X = 4 und Y = 5 ein zusammenhängender Speicherbereich von Adresse 4000 (4000 ... 4FFF ist der erste 4-K-Block) bis Adresse 5FFF (5000 ... 5FFF ist der zweite 4-K-Block).

EPROM-Bereich (2708)

Vier dieser 1-K-EPROMs bilden einen 4-K-Block. Dazu wird einer der Ausgänge von IC5 mit N1 ("V") verbunden. Den anderen Eingang von N1 ("W") verbindet man entweder mit "V" oder über eine Drahtbrücke mit +5 V. Die 4-K-Adressierung wird weiter durch IC7 unterteilt,

das durch die Drahtbrücken P-Q und S-T mit den Adreßleitungen A10 und A11 verbunden ist. So ergibt sich schließlich die folgende Zuordnung der Adressen zu den EPROMs:

IC25	V000 ...	V3FF
IC26	V400 ...	V7FF
IC27	V800 ...	VBFF
IC28	VC00 ...	VFFF

EPROM-Bereich (2716)

Hier ist eine Adressierung mit 8 K (4 x 2 K) notwendig. "V" von N1 muß mit einem geradzahligen und "W" mit einem ungeradzahligen Ausgang von IC5 verbunden werden. Wenn also zum Beispiel V = 2 und W = 7 gewählt ist, dann ergeben sich folgende Adreßbereiche für die einzelnen EPROMs:

IC25	2000 ...	27FF
IC26	7000 ...	77FF
IC27	2800 ...	2FFF
IC28	7800 ...	7FFF

IC25 und IC27 sowie IC26 und IC28 bilden jeweils ein "4-K-Paar". Die angegebenen Adressenbereiche gelten nur, wenn in Bild 1 die Drahtbrücken P-R und S-T gelegt sind.

Tabelle 1

EPROM-Typ	Drahtbrücken	Reihenfolge der Adressierung	ab Seite
2708	P-Q, S-T e-f, a-c	IC25-26-27-28	0 ... 15
2716	P-R, S-T e-g, a-d	IC25-27-26-28 IC26-28-25-27	0,2,4,6,8,10,12,14 1,3,5,7,9,11,13,15
2732	P-R, S-U e-g, a-b	IC25 IC26 IC27 IC28	0,4,8 oder 12 1,5,9 oder 13 2,6,10 oder 14 3,7,11 oder 15

Tabelle 2. Beispiel für die Adressierung von RAM- und EPROM-Bereich

RAM		2708	EPROM 2716	2732
1k0 = 1000 ... 13FF	IC25 =	3000 ... 33FF	3000 ... 37FF	3000 ... 3FFF
1k1 = 1400 ... 17FF	IC26 =	3400 ... 37FF	4000 ... 47FF	4000 ... 4FFF
1k2 = 1800 ... 1BFF	IC27 =	3800 ... 3BFF	3800 ... 3FFF	5000 ... 5FFF
1k3 = 1C00 ... 1FFF	IC28 =	3C00 ... 3FFF	4800 ... 4FFF	6000 ... 6FFF
1k4 = 2000 ... 23FF				
1k5 = 2400 ... 27FF				
1k6 = 2800 ... 2BFF				
1k7 = 2C00 ... 2FFF				
Pins 9 und 5 von IC5 verbunden mit N2		Pin 14 von IC5 verbunden mit N1	Pins 14 und 3 von IC5 ver- bunden mit N1	Pins 14, 3, 7 und 11 von IC5 ver- bunden mit k, l, m, n.

Weicht man von der angegebenen Verdrahtung ab, erhalten die EPROMs andere Adressbereiche zugeteilt. Verbinden die Drahtbrücken die Punkte P-T und S-R, bilden IC25 und IC26 sowie IC27 und IC28 jeweils ein "4-K-Paar".

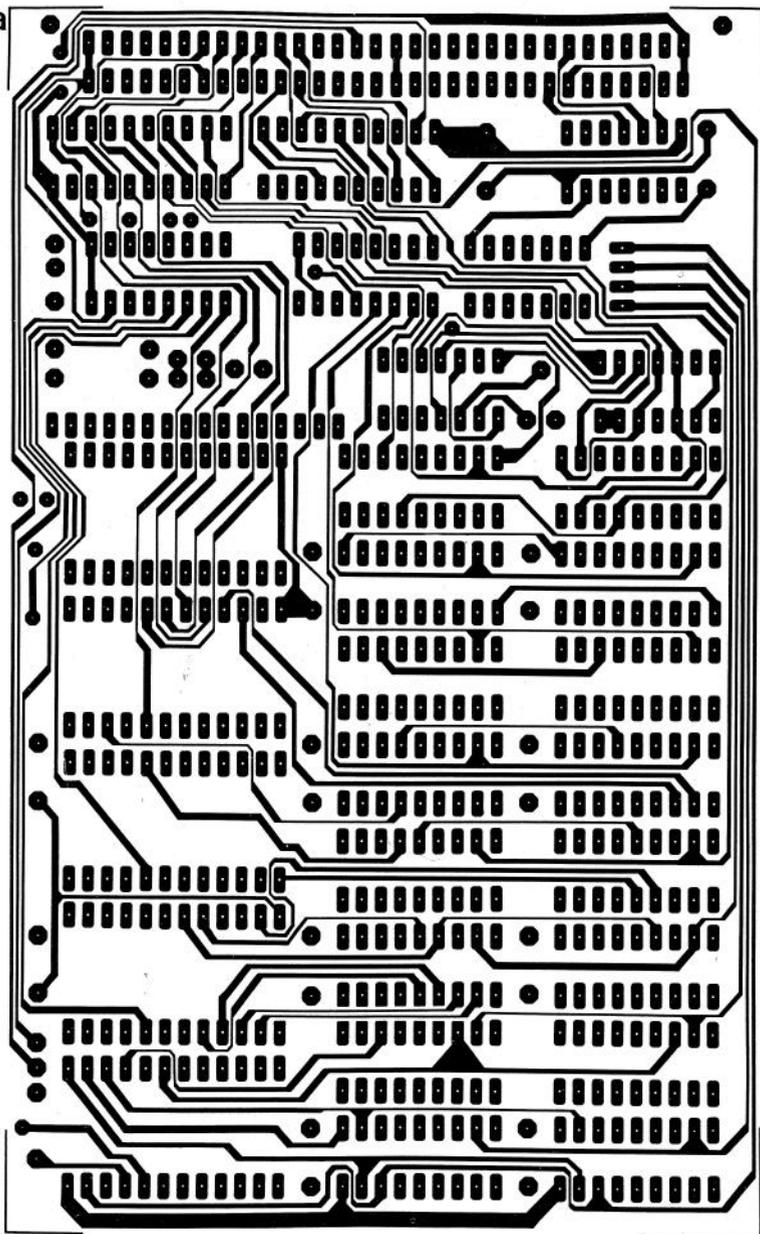
EPROM-Bereich (2732)

Jedes dieser ICs nimmt bereits einen 4-K-Adressbereich ein. Das entspricht also der Ansteuerung mit einem Ausgang von IC5. Die Unterteilung des Adressbereichs mit Hilfe von IC7 ist deshalb nicht nötig. Die gewünschte Verbindung zwischen Adressdekoder IC5 und den EPROMs IC25 ... IC28 stellt man einfach durch Drahtbrücken zwischen den Anschlüssen von IC5 und den jetzt freien Anschlüssen von IC7 (Pins 9 ... 12) her. Dabei gehört Pin 9 zu IC25, Pin 10 zu IC26 usw. Wenn IC7 nicht eingesetzt ist, ergibt sich allerdings ein Nachteil: Nach der Verdrahtung der "Seitenwahl"-Drahtbrücken mit IC25 bis IC28 kann man die Eingänge von N1 nicht mehr so einfach beschalten. Es ist dann besser, IC7 doch einzusetzen

und die Anwahl der EPROMs über die Adreßleitungen A12 und A13 (mit Hilfe der entsprechenden Drahtbrücken) vorzunehmen. In diesem Fall sind zwei Drahtbrücken zwischen den Punkten P-R und S-U einzusetzen (siehe dazu auch Tabelle 1).

Wenn nicht mehr als zwei EPROMs vom Typ 2732 eingesetzt sind, ist auch die Verdrahtung von IC5 mit N1 einfach: Genau wie bei den anderen EPROM-Typen legt man einfach Verbindungen zwischen die Ausgänge von IC5 und die Eingänge "V" und "W" von N1. Bei der Verwendung von mehr als zwei EPROMs reichen die beiden Eingänge von N1 nicht aus. Das Problem kann durch eine einfache ODER-Schaltung gelöst werden: Zunächst erhalten die Eingänge von N1 Pull-Up-Widerstände (nach +5 V). Dafür ist ausreichend Platz auf der Platine vorhanden. Danach werden die gewünschten Verbindungen zwischen IC5 und N1 mit Hilfe von Dioden hergestellt. Die Anoden der Dioden liegen dabei an N1. Zusätzliche Dioden am Eingang eines TTL-Gatters

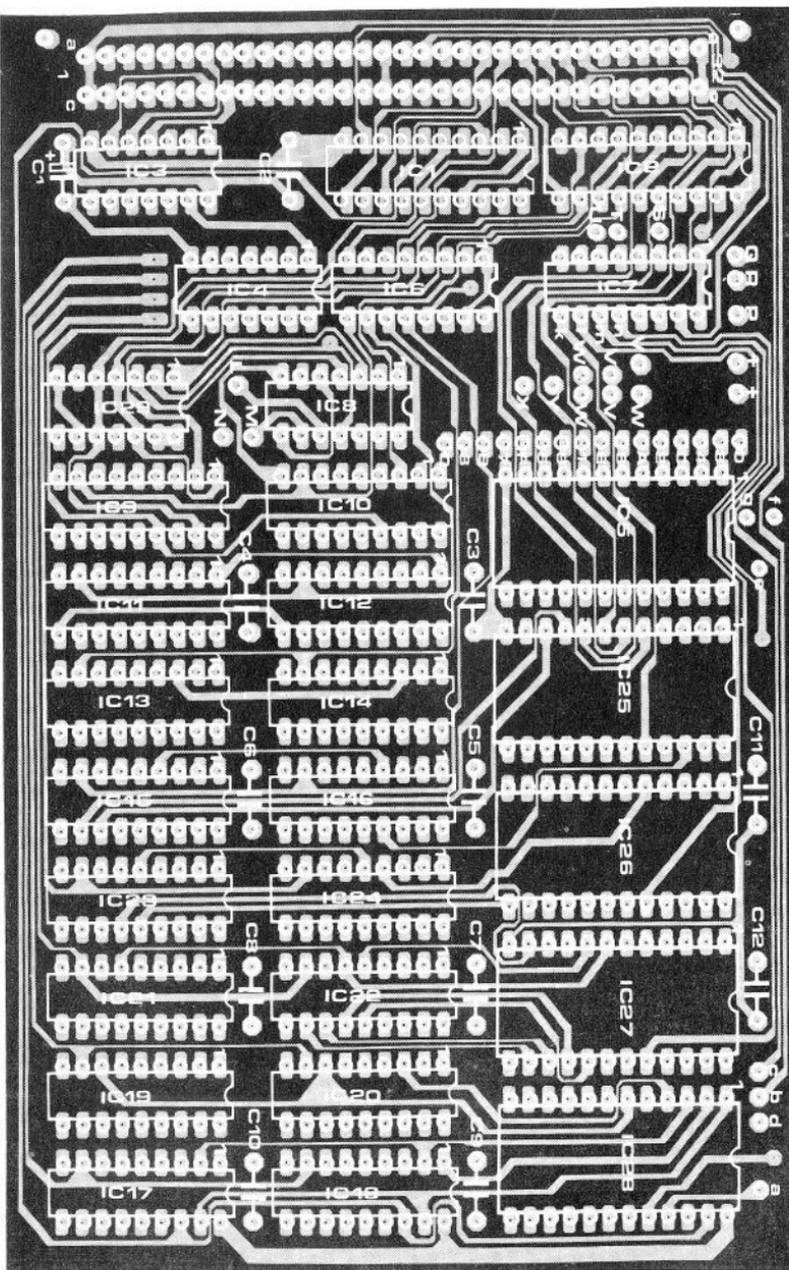
2a



sind zwar nicht "die feine Art", aber die Schaltung funktioniert. Die Reihenfolge der Seiten ist abhängig von der verwendeten Dekodierung. Wenn IC7 nicht eingesetzt ist, hängt die Reihenfolge von den Verbindungen von den

Ausgängen von IC5 mit den Punkten k, l, m, n ab. Jedes EPROM kann also auf einer beliebigen Seite liegen (selbstverständlich ist eine Doppelbelegung nicht möglich!). Sonst ist die Adressierung gemäß Tabelle 1 festgelegt.

2b



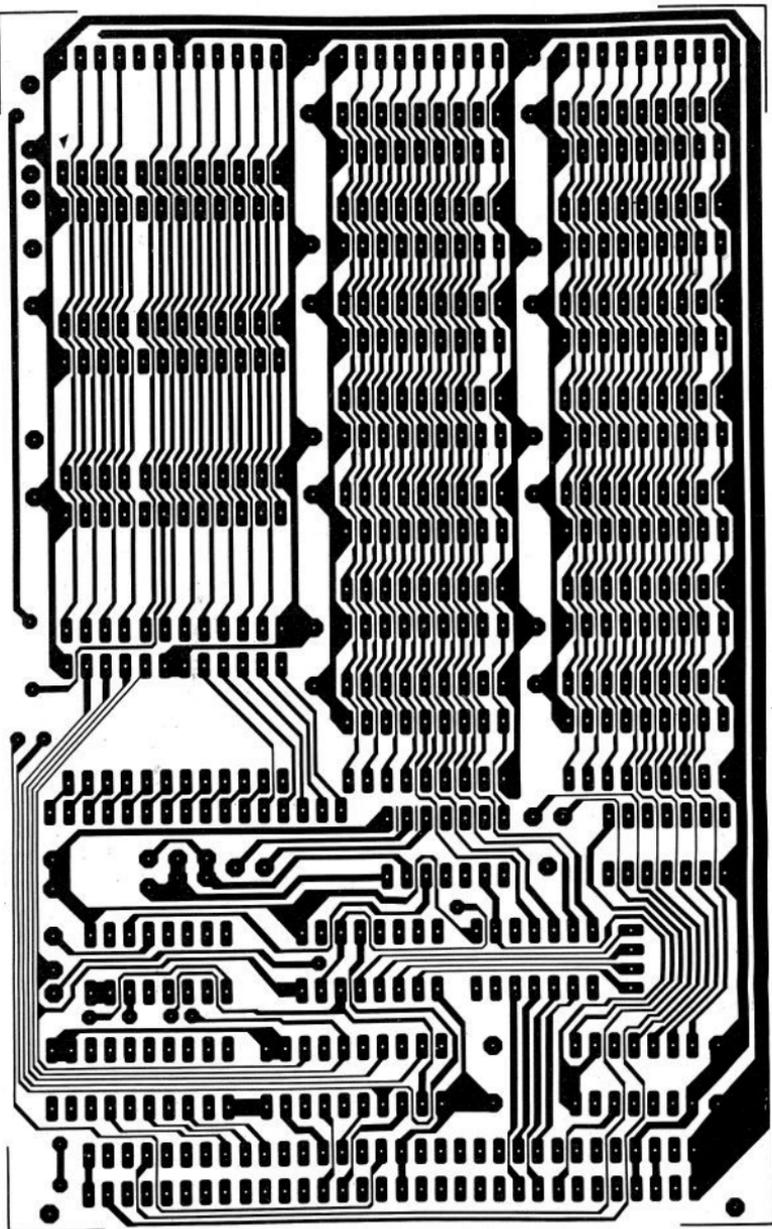
Drahtbrücken und unbenutzte Eingänge

Will man nicht den gesamten EPROM- oder RAM-Bereich adressieren, dürfen unbenutzte Eingänge der Gatter N1 und

N2 nicht offen bleiben; sie müssen mit +5 V verbunden werden.

Ein besonderes Augenmerk ist auf die Drahtbrücken an den Eingängen von IC7 und IC25 . . . IC28 zu richten. Die notwendigen Verbindungen sind vom

2c



verwendeten EPROM-Typ abhängig:

2708: P-Q, S-T, e-f, a-c

2716: P-R, S-T oder T-P, R-S, e-g, a-d

2732: P-R, S-U, e-g, a-b

Genau so wichtig, wie die gerade erwähn-
ten Drahtbrücken ist die Verbindung der

Bild 2. Platinenlayout und Bestückungsplan für die RAM/EPROM-Karte. Diese doppel-seitige Platine bietet Platz für RAM-ICs mit einer Gesamtkapazität von 8 K und für ROMs bis maximal 16 K.

Tabelle 3

IC5 Ausgang	4-KByte- Block Adreßbereich	2 x 4-KByte-RAM Bereich gewählt durch		4-KByte-EPROM- Bereich 2708- Typen gewählt durch	2 x 4-KByte-EPROM- Bereich 2716-Typen gewählt durch	
		X	Y	V	V	W
0	0000 ... 0FFF	0		0	0	
1	1000 ... 1FFF		1	1		1
2	2000 ... 2FFF	2		2	2	
3	3000 ... 3FFF		3	3		3
4	4000 ... 4FFF	4		4	4	
5	5000 ... 5FFF		5	5		5
6	6000 ... 6FFF	6		6	6	
7	7000 ... 7FFF		7	7		7
.
.
.
F	F000 ... FFFF	F	F	F	F	F

Stückliste

Widerstände:

2 x 1 k (siehe Text)

Kondensatoren:

C1 = 1 µ/10 V Tantal

C2 ... C7 = 100 n MKM

Halbleiter:

IC1, IC2 = 74LS241

IC3, IC4 = 74LS243

IC5 = 74(LS)154

IC6, IC7 = 74(LS)155

IC8 = 74LS08

IC9 ... IC24 = 2114

IC25 ... IC28 = 2708 oder 2716
oder 2732

IC29 = 74LS00

Steckerleiste:

64-polig DIN 41612

Masseleitung des Netzteils mit den Konnektoranschlüssen 4a und 4c, 16a und 16c sowie 32a und 32c. Diese Pins sind auf der Platine nicht miteinander verbunden.

Anpassungsschwierigkeiten?**Hier ist die Lösung!**

In der bisher beschriebenen Form ist die RAM/EPROM-Karte direkt für die in Elektor beschriebenen Systeme SC/MP- und Junior-Computer geeignet. Der Junior arbeitet mit der 6502-CPU. Das heißt jedoch leider nicht, daß in jedem

6502-System die RAM/EPROM-Karte ohne weiteres verwendbar ist. Als Beispiel ist hier nur der populäre Mikrocomputer von OSI (Ohio Scientific) erwähnt, der zum "Lesen und Schreiben" ein zusätzliches Signal benötigt.

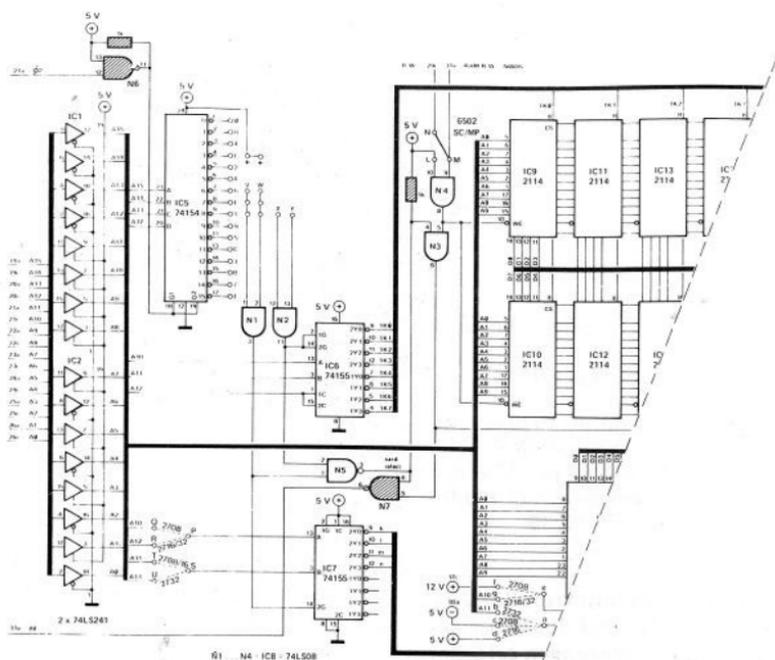
Ähnliches gilt auch für die Z80- und 8080-Systeme. Hier besteht die Gefahr, daß die Speicherkarte während des Ein- oder Ausgabezykluses neu adressiert wird.

In allen genannten Fällen ist also eine Anpassung der Speicherkarte unumgänglich. Die dazu erforderlichen Bauteile sind bereits auf der Karte enthalten. Es fehlen lediglich einige Widerstände für die offenen Kollektorausgänge beim OSI.

OSI-Anpassung

Das bereits erwähnte Zusatzsignal beim OSI-Mikrocomputer ist das dd-Signal; dd ist die Abkürzung von data-direction. Das dd-Signal setzt sich aus dem Φ2- und dem R/W-Signal zusammen und muß während des Lesezykluses logisch 0 sein. Ein solches Signal ist bereits auf der Speicherkarte vorhanden, allerdings ist es invertiert. Die erste Forderung lautet also: Vorhandenes Signal invertieren. Das dd-Signal muß noch eine zweite Forderung erfüllen, es muß dem OSI-Computer von einem Gatter mit offenem Kollektorausgang angeboten werden.

Das Φ2-Signal entsteht beim Anschluß



der Karte an den Junior-Computer durch das Kombinieren der Signale R/W und RAM-R/W. In OSI-Systemen steht jedoch das RAM-R/W-Signal nicht zur Verfügung, deshalb muß $\Phi 2$ auf andere Art und Weise entstehen. Bild 3 zeigt die Lösung für beide Forderungen.

Anstelle des 74LS00 (IC29) tritt ein 74LS03. Beide ICs haben die gleiche

Bild 3. In dem hier gezeigten Ausschnitt der Schaltung aus Bild 1 sind die notwendigen Änderungen für den Anschluß an einen OSI-Computer deutlich zu erkennen.

Pinbelegung. Der einzige Unterschied besteht darin, daß die NAND-Gatter des ICs 74LS03 offene Kollektorausgänge haben.

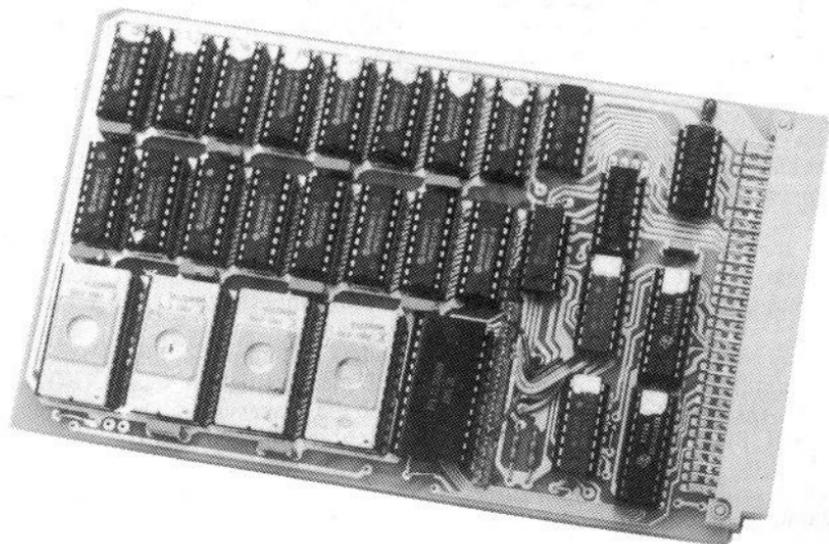


Tabelle 4

Leiterbahnen unterbrechen: :

IC29	pin 7 und 9
IC29	pin 5 und 10
IC29	pin 4 und 12
IC29	pin 10 und IC9 pin 17
IC29	pin 12 und IC9 pin 1
IC5	pin 18, 19 und Masse

Verbindungen herstellen (Platinenunterseite):

IC29	pin 5 und IC9 pin 17 (a7)
IC29	pin 4 und IC9 pin 1 (a6)
IC29	pin 3 und IC29 pin 10 (card select)
IC29	pin 9 und IC8 pin 5 (b-R/W)
IC29	pin 12 und konektor pin 27-a (02)
IC29	pin 8 und Konektor pin 11-a (dd)
IC29	pin 11 und IC5 pin 18, 19

Das $\Phi 2$ -Signal wird nun vom Bus abgenommen. Dazu sind auf der Platine allerdings einige Änderungen erforderlich (Bild 3). Das nun invertierte Signal wird dem Adreßdekoder IC5 zugeführt, wodurch die Adressierung der Speicherkarte nur noch während der positiven Phase von $\Phi 2$ möglich ist.

Sogenannte Buskonflikte sind damit ausgeschlossen. Am Ausgang des Gatters N3 steht das invertierte dd-Signal zur Verfügung, um die Datenbuspuffer zu steuern. Um nun das dd-Signal für den OSI-Computer zu erhalten, genügt es, dem NAND-Gatter N3 ein weiteres NAND-Gatter parallelzuschalten. Der Widerstand für den offenen Kollektorausgang dieses Gatters befindet sich bereits im OSI-Computer, so daß man ihn nicht zusätzlich auf der Speicherkarte unterbringen muß. Die Verbindung zwischen dd-Signal und Computer geschieht über den Bus. Hierfür steht der Anschlußpin 11a zur Verfügung, da er im Junior-Computer keine Funktion hat. Die Karte bleibt somit zwischen beide Systeme auswechselbar.

Schließlich muß noch der R/\bar{W} - mit dem $RAM-R/\bar{W}$ -Anschluß verbunden werden (Brücke N-M, Bild 3) und Anschlußpunkt L mit +5 V.

In Tabelle 4 sind die Änderungen aufgelistet; daraus geht hervor, welche Leiter-

bahn wo zu unterbrechen ist. Als letztes sind noch zwei 1 kOhm-Widerstände einzulöten. Der erste Widerstand an Pin 4 von IC8 und der zweite an Pin 11 von IC29. Der noch freie Anschluß der Widerstände kommt an +5 V, z.B. an Pin 14 der genannten ICs.

Z80-Anpassung

Der ursprüngliche Entwurf der RAM/EPROM-Karte geht davon aus, daß zur Zeit des Read- oder Write-Strobes der CPU sowohl gültige Adressen als auch die Richtung des Datentransfers (!) definiert sind. Das trifft zwar beim SC/MP und auch beim 6502, nicht jedoch beim Z80 zu. Man kann beim Z80 drei Zustände angeben, bei denen eine gültige Adresse ausgegeben wird: Normaler Speicherzugriff, Zugriff auf eine der 256 I/O-Adressen und Speicherzugriff während eines Refresh-Zyklus. Rechnet man noch die vierte Möglichkeit einer ungültigen Adresse hinzu, dann wären nur zwei CPU-Leitungen notwendig, um alle möglichen Adressenzustände zu beschreiben. Tatsächlich verwendet der Z80-Prozessor aber drei Leitungen:

\overline{MREQ} , wenn ein Speicherplatz gemeint ist,

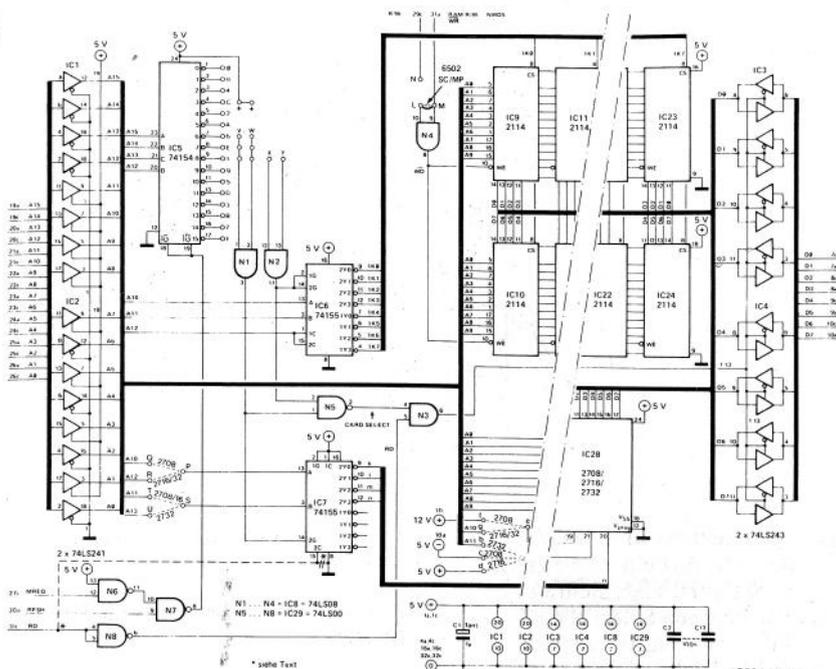
\overline{IORQ} , wenn ein Peripheriegerät gemeint ist und

RFSH, wenn dynamischen RAMs auf die Sprünge geholfen wird.

\overline{IORQ} kann man an dieser Stelle außer Acht lassen. Benötigt werden nur die Beschreibungen eines normalen Speicherzugriffs und des Refreshings.

Während eines normalen Speicherzugriffs gibt die CPU zunächst die Adressen aus. Kurze Zeit später folgen der \overline{MREQ} und in einem Lese-Zyklus der \overline{RD} -Strobe. Im Lese-Zyklus sind beide Signale synchron. Sind beide Signale beendet, dann hat die CPU die anliegenden Daten aufgenommen (gelesen).

Anders im Schreib-Zyklus. Hier setzt die CPU gleichzeitig mit dem \overline{MREQ} -Signal auch die ausgehenden Daten auf den Datenbus. Das \overline{WR} -Signal wird dagegen nicht gleichzeitig, sondern erst später aktiv. Damit ist es möglich, die aktive Flanke dieses Strobes direkt zum



82911-3-4

82134

Schreiben in einen Speicher zu verwenden (wenn man die Systempufferung so ausgelegt hat, daß der Datenbus auch wirklich zum Speicher durchschaltet, bevor der \overline{WR} -Strobe erscheint!). Das \overline{WR} -Signal wird gleichzeitig mit dem \overline{MREQ} -Signal abgeschaltet.

Während eines Refresh-Zyklus muß man einen Zugriff auf die Speicherkarte vermeiden. Der Zyklus läuft folgendermaßen ab: Zunächst wird das Refresh-Signal aktiv. Erst dann erfolgt ein Strobe mit dem \overline{MREQ} -Signal. \overline{RD} und \overline{WR} werden nicht benutzt, weil die CPU bei diesem Speicherzugriff die Daten ignoriert. Zusammenfassend ergeben sich für die Steuerung eines Speichersystems die folgenden Forderungen:

1. Der Speicher wird angesprochen, wenn \overline{MREQ} aktiv und \overline{RFSH} *nicht* aktiv ist.
2. Die Daten müssen am RAM liegen, *bevor* der \overline{WR} -Strobe aktiv ist.
3. Die Daten müssen auf dem Bus erscheinen, solange \overline{RD} aktiv und der Speicher selektiert ist. Sonst *nicht!*

Bild 4 zeigt die Schaltung der auf diese Weise modifizierten Karte. Forderung 1

Bild 4. Schaltung der für ein Z80-System modifizierten RAM/EPROM-Karte.

wird durch die Verknüpfung von \overline{MREQ} und \overline{RFSH} durch N6 und N7 erfüllt. Pin 8 von N7 wird nur dann "0", wenn die CPU einen Speicherplatz adressiert. Die Karte soll natürlich nicht bei jedem Speicherzugriff reagieren, sondern nur dann, wenn der auf ihr enthaltene Speicherausschnitt gemeint ist. Deshalb ist Pin 8 von N7 mit den Pins 18 und 19 von IC5 verbunden. Der Dekoder 74154 wird also nur dann eingeschaltet, wenn ein gültiger Speicherzugriff vorliegt. Seine Ausgänge schalten über N1 und N2 die CS-Dekoder IC6 und IC7 ein. Außerdem liefert der Ausgang von N5 ein Platinen-Select-Signal ("active high"). Forderung 2 wird dadurch erfüllt, daß die Karte im Ruhezustand die Daten vom BUS zum Speicher durchläßt. Die Daten liegen also auch an, wenn \overline{WR} erscheint.

Die RAM/EPROM-Karte verwendet statt eines \overline{RD} -Signals das \overline{WR} -Signal und schaltet die Daten auf den BUS, wenn

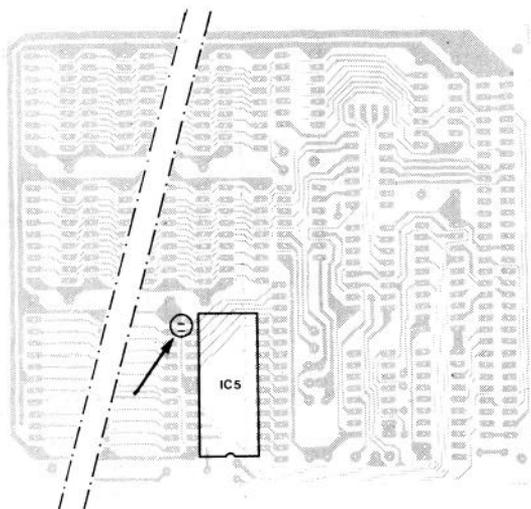


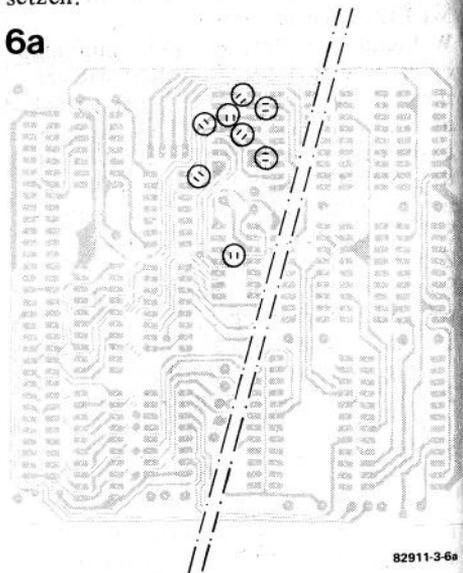
Bild 5a. Leiterbahnunterbrechung auf der Lötseite.

Bild 5b. Verbindungsleitungen auf der Lötseite.

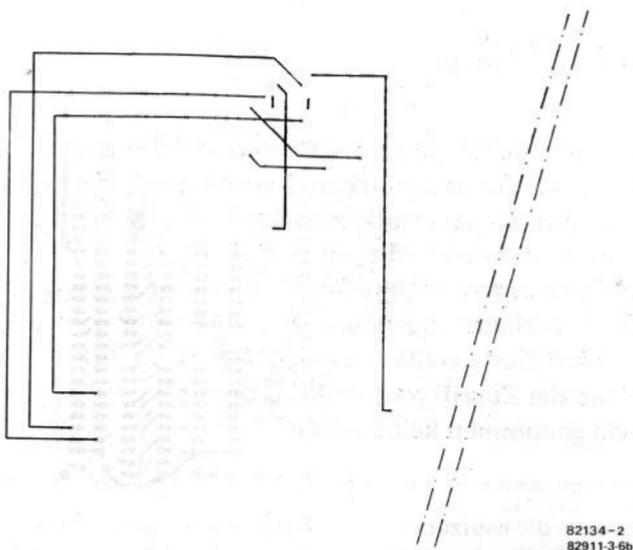
der Speicher selektiert und das \overline{WR} -Signal inaktiv ist. Auf ein Z80-System übertragen: Während eines Schreibzyklus liegt das Platinen-Select-Signal vor dem \overline{WR} -Strobe an. Die nichtmodifizierte Karte "sieht" Select und Nicht- \overline{WR} und nimmt deshalb einen Lesezyklus an. Bis zum Eintreffen des Schreibpulses \overline{WR} schaltet sie also die RAM-Daten auf den BUS. Zum einen liegen damit die BUS-Daten nicht am RAM an, wenn der \overline{WR} -Strobe erscheint (Forderung 2). Zum anderen wird der Datenbus bereits von den Puffern der CPU versorgt, weil die CPU-Steuerung den Schreibzyklus erkannt hat. Der BUS darf allerdings nicht aus zwei Richtungen gleichzeitig versorgt werden, sonst spielt er "verrückt". Je nach eingesetzten Treibern entstehen nämlich Stromspitzen auf der +5-V- und auf der GND-Leitung, die durchaus zu einem "Systemkollaps" führen können. Aus diesem Grunde wird im Z80-System das mittels N8 invertierte \overline{RD} -Signal (über N3 verknüpft mit dem Platinen-Select-Signal) für die Richtungssteuerung der DATA-BUS-Puffer verwendet. N4 dient zur Pufferung der \overline{WE} -Leitung, die immerhin 16 ICs versorgen muß. Die für die Modifikation notwendigen Inverter N6 und N8 und das NAND-Gatter N7 sind bereits auf der Platine in IC29 (74LS00) enthalten. In der

Originalschaltung liegen die dort ungenutzten Eingänge auf definierten Pegeln, um Übersprechen auf aktive Gatter zu vermeiden. Diese Anschlüsse muß jetzt durch die in den Bildern 4 und 6b angegebenen Verbindungsleitungen ersetzen.

6a



6b

82134-2
82911-3-6b

Nach Ausführung der erwähnten Änderungen ist die RAM/EPROM-Karte für den Einsatz in Z80-Systemen gerüstet. Insgesamt sind 7 neue Leitungen und 9 Unterbrechungen erforderlich. Bild 6a zeigt die Änderungen auf der Lötseite der Platine.

In Bild 5 ist zu erkennen, daß lediglich eine Leiterbahn auf der Bestückungsseite unterbrochen werden muß.

8080-Anpassung

Für die 8080-CPU gilt, daß die Anpassung der Speicherkarte genauso vorgenommen wird, wie es bereits bei der Z80-CPU

Bild 6. Leiterbahnunterbrechung auf der Bestückungsseite.

beschrieben ist. Der einzige Unterschied besteht lediglich darin, daß der 8080 kein $\overline{\text{RFSH}}$ -Signal zur Verfügung hat. Da nun der 8080 ohne $\overline{\text{RFSH}}$ -Signal mit der Speicherkarte nichts anfangen kann, wäre eine zusätzliche externe Refresh-Logik erforderlich.

Doch es geht auch anders. Es genügt, wenn man die Speicherkarte so an den 8080 anpaßt, wie es bei der Z80-Anpassung beschrieben ist. Dabei muß dann noch der RFSH-Eingang der Speicherkarte mit +5 V verbunden werden.

Dynamische RAM-Karte

Die Preise von dynamischen Speicher-ICs sind in der letzten Zeit derart gepurzelt, das es trotz der notwendigen Steuer-Elektronik sicherlich der Mühe wert ist, sie anstelle von statischen Speicher-ICs in einem Mikrocomputer-System einzusetzen. Mit nur acht dieser ICs kann man 16 K auf einer Euro-Karte unterbringen! Dabei ist auch noch genügend Platz für die Steuerlogik vorhanden. Außerdem braucht eine solche "dynamische" RAM-Karte weitaus weniger Strom als ihre statische "Schwester". Und die Zugriffszeit ist außerordentlich kurz. Nachteile gibt's im Grunde genommen keine (mehr).

Die Computersysteme, die heutzutage von Hobby-Programmierern eingesetzt werden, speichern Informationen in den meisten Fällen in "flüchtigen" Speichern: in statischen und (selten) in dynamischen. Die Daten in statischen RAMs bleiben so lange erhalten wie die Betriebsspannung anliegt. Bei dynamischen RAMs sind die Verhältnisse etwas komplizierter. Die

Daten müssen nämlich regelmäßig erneuert werden, damit sie nicht verlorengehen. Die inzwischen erfreulich niedrigen Preise für dynamische RAMs haben uns veranlaßt, ihre Vorteile als Speicher jetzt auch für den Hobby-Programmierer nutzbar zu machen.

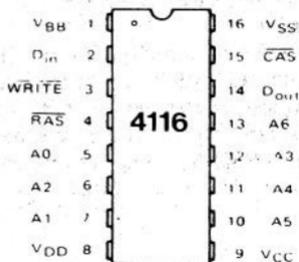
Dynamisch kontra statisch

Vor allem die Selbstbauer unter den Computer-Fans haben bisher fast ausschließlich statische RAMs verwendet. Die Gründe dafür liegen auf der Hand: Ein "statischer" Speicher ist wesentlich einfacher zu handhaben als ein dynamischer. Alle Funktionen sind gleich im IC enthalten. Man braucht keine externe Steuerung. Auch Timing-Probleme gibt's nicht, wenn die RAMs für die entsprechende Anwendung nur schnell genug sind.

Die Speicherzelle eines statischen RAM stellt im Prinzip eine Art Set-Reset-Flipflop dar. Ein solches Flipflop besteht aus mindestens 5 bis 6 Transistoren. Das bedeutet also eine ganze Menge Aufwand für ein Speicher-IC.

Ein dynamisches RAM enthält dagegen keine Flipflops, sondern Kondensatoren mit FET-Schaltern. Die Spannung über dem Kondensator gibt an, ob eine "0" oder eine "1" in der Zelle gespeichert wurde. Vergleicht man die Schaltungen einer statischen und einer dynamischen

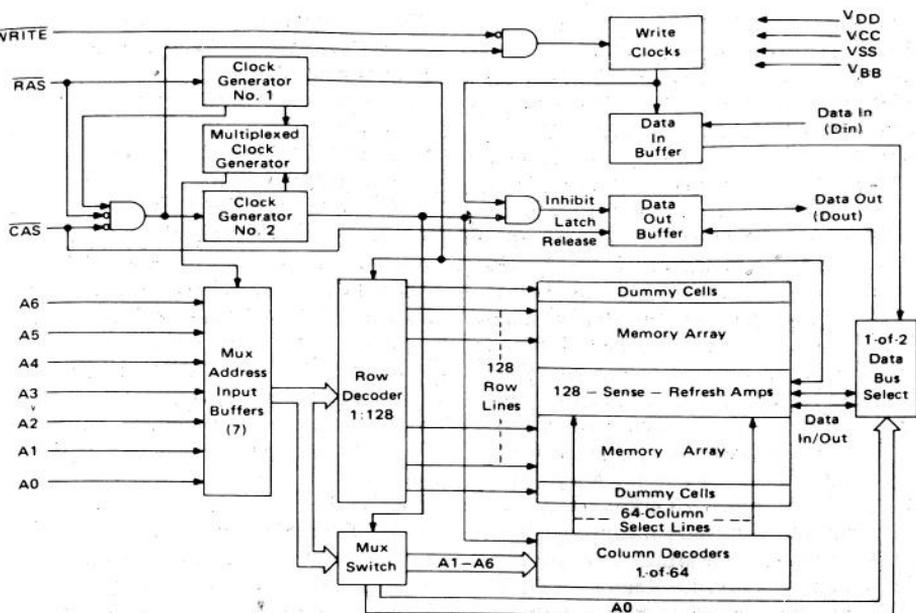
1 Pin-Belegung



PIN NAMES

A0 - A6	Address Inputs
CAS	Column Address Strobe
D _{in}	Data In
D _{out}	Data Out
RAS	Row Address Strobe
WRITE	Read/Write Input
V _{BB}	Power (-5 V)
V _{CC}	Power (+5 V)
V _{DD}	Power (+12 V)
V _{SS}	Ground

Bild 1. Die Pinbelegung des dynamischen RAM-ICs 4116. Das IC benötigt drei unterschiedliche Betriebsspannungen. Die Stromaufnahme ist allerdings sehr gering.



Speicherzelle miteinander, dann wird deutlich, daß die dynamische Speicherzelle viel weniger Aufwand erfordert als die statische. Man kann also auf der gleichen Chip-Fläche wesentlich mehr "dynamische" Speicherzellen unterbringen. Leider sind Kondensator und FET-Schalter nicht ganz ausreichend, um eine gut funktionierende dynamische Speicherzelle zu "bauen". Jeder Kondensator besitzt nämlich einen geringen Leckstrom, der dazu führt, daß die Spannung über dem Kondensator mit der Zeit ohne Einwirkung von außen sinkt. Die gespeicherten Daten würden also nicht ohne weiteres über längere Zeit hinweg in ihrem ursprünglichen Zustand zur Verfügung stehen. Von Zeit zu Zeit muß man deshalb den Daten "auf die Sprünge helfen" (Englisch: refresh). Das geht natürlich nur mit Hilfe einer zusätzlichen Steuer-Elektronik plus eines sehr genauen Timings, um rechtzeitig und schnell zu "refreshen".

Das ist jedoch nicht das einzige Problem. Da sehr viele Speicherplätze auf einem Chip untergebracht sind, ist auch die Adressierung nicht so einfach. Die Hersteller von dynamischen RAMs haben dieser Tatsache dadurch Rechnung ge-

Bild 2. Das Blockscha des 4116-RAMs. Der Speicher besteht aus einer Matrix von 128 Reihen ("rows") und 128 Spalten ("columns"). Zwischen der 64. und der 65. Reihe sitzen die Leseverstärker ("sense amplifiers").

tragen, daß sie einen gemultiplexten Adreßbus integrierten. Auch darauf muß man also achten.

Die Preise für dynamische RAMs sind im Vergleich zu ihren statischen Konkurrenten pro Speicherplatz so niedrig geworden, daß die Kosten für die Steuer-Elektronik demgegenüber nicht mehr ins Gewicht gefallen. Die Stromaufnahme ist jedenfalls wesentlich kleiner als bei einem statischen Speicher-System gleichen Umfangs. Allerdings benötigt unsere dynamische RAM-Karte im Gegensatz dazu drei Betriebsspannungen.

Alles in allem halten wir die dynamische RAM-Karte doch für eine echte Alternative zu einem statischen Speicher in einem Hobby-Computer-System.

Aufbau und Funktion eines dynamischen RAM-Speichers

Die dynamische RAM-Karte wird mit 4116-RAMs bestückt, mit preiswerten und gut erhältlichen Typen verschiedener

Hersteller. Es handelt sich dabei um ein 16384 x 1 bit dynamisches RAM, so daß man mit 8 dieser ICs einen 8 bit "breiten" 16-K-Speicher erhält. Das RAM-IC ist ziemlich schnell. Die Zugriffs- ("access"-) Zeiten von 150 . . . 300 ns sind auf dem 16poligen DIL-Gehäuse aufgedruckt. 16polig? Wie kann man 16 K darin unterbringen? Der Speicher des 4116-ICs ist als Matrix von 128 Spalten und 128 Reihen (128 x 128 = 16384 = 16 K) ausgeführt. Zur Adressierung eines Bit braucht man 14 Adreß-Bits, denn $2^{14} = 16384$: Sieben zur Wahl der Reihe und sieben zur Wahl der Spalte. Ein integrierter Spalten- und ein integrierter Reihen-Dekoder ordnen jeder 7-bit-Adresse eine Spalte und eine Reihe zu. Wie an der Pin-Belegung des 4116-RAM in Bild 1 zu erkennen ist, sind allerdings nur 7 Adreß-Eingänge vorhanden. Ein integrierter Multiplexer sorgt dafür, daß zunächst die Reihen-Adresse und dann die Spalten-Adresse verarbeitet wird. Um dem IC anzuzeigen, welche Adresse gerade "gemeint" ist, gibt ein negativer Impuls am \overline{RAS} - ("row address strobe") Eingang an, daß eine Reihen-Adresse anliegt. Dagegen steht ein negativer Impuls am \overline{CAS} - ("column address strobe") Eingang für "Spalten-Adresse liegt an". Da es sich um einen 1 bit "breiten" Speicher handelt, braucht man auch nur einen Daten-Ausgang und einen Daten-Eingang (D_{out} und D_{in}). Der logische Pegel am \overline{WRITE} -Eingang legt fest, ob ein Bit geschrieben oder gelesen werden soll. Schließlich bleiben noch die Betriebsspannungs-Anschlüsse VDD (+12 V), VCC (+5 V), VBB (-5 V) und VSS (0 V).

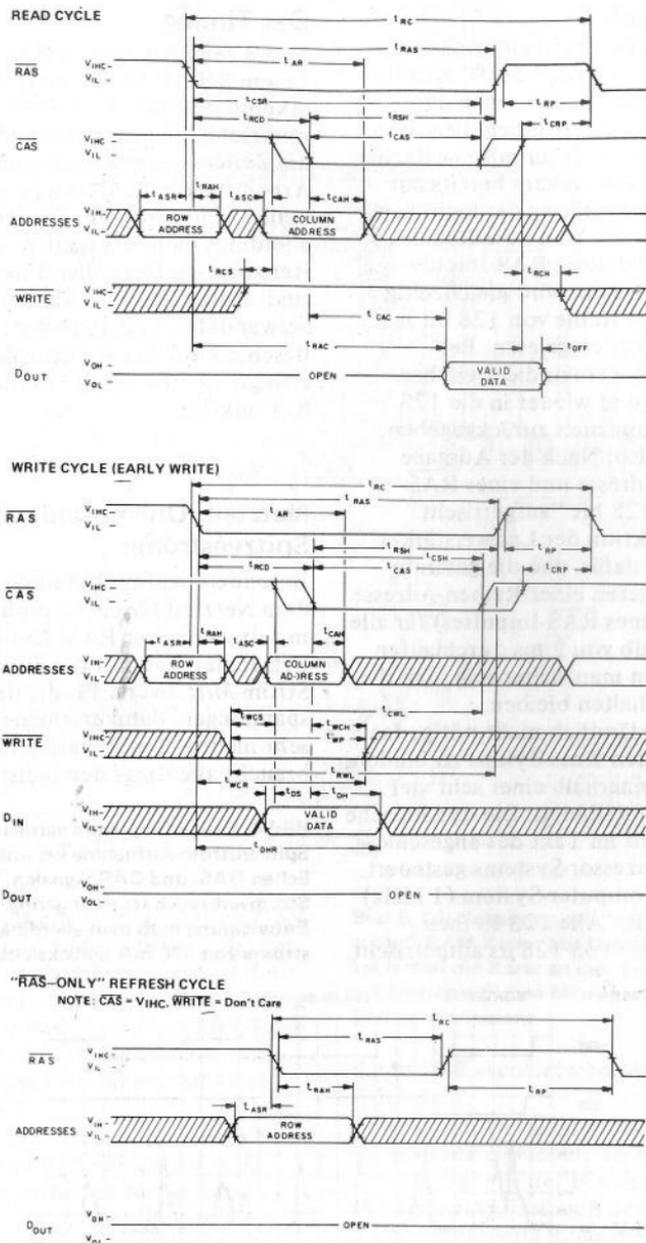
Außer den schon erwähnten Funktionen enthält das IC 128 Leseverstärker ("sense refresh amps"), die zwei Aufgaben erfüllen: Zum einen sorgen sie während des Refreshings für die Nachladung der Kondensatoren. Und zum anderen dienen sie als Verstärker beim Ein- und Auslesen der Speicherzellen. Jeder dieser Verstärker ist eine Art Flipflop. An jedem Eingang ist die Hälfte einer Reihe angeschlossen. Erhält das IC eine Adresse, dann werden die Daten aller Kondensatoren der Reihe so ausgelesen, daß jeder Kondensator über

den Eingang des Leseverstärkers entladen wird. Die auf diese Weise entstehende Spannungsänderung wird vom Flipflop "verstärkt" und zur Spaltenleitung zurückgeführt, so daß der ursprüngliche (verstärkte) logische Pegel wieder an den Kondensator gelangt. Der Leseverstärker enthält also nun dieselben Daten wie der zunächst entladene und danach wieder geladene Kondensator. Diese Methode ermöglicht das gleichzeitige Refreshing aller in den Kondensatoren gespeicherter logischer Pegel einer Reihe, wenn diese "angesprochen" wird. Um eine Vorstellung von der Kapazität eines solchen Kondensators zu bekommen: Bei einem 4116-RAM beträgt sie etwa 0,04 pF! Aus der vorangegangenen Beschreibung läßt sich bereits erkennen, welche Signale in welcher Reihenfolge auftreten müssen. Beim Auslesen eines Speicherplatzes wird zunächst eine 7-bit-Adresse an die Adreß-Eingänge gelegt. Anschließend gelangt ein Impuls an den \overline{RAS} -Eingang. Die Reihen-Adresse muß eine bestimmte Zeit lang anliegen. Danach kann die 7-bit-Spalten-Adresse angelegt werden. Anschließend erhält der \overline{CAS} -Eingang einen Impuls. Auch die Spaltenadresse muß eine bestimmte Zeit lang anliegen. Über den internen Ausgangspuffer gelangt danach der logische Pegel der gewählten Adresse an den Daten-Ausgang. Während dieser Prozedur muß am \overline{WRITE} -Eingang eine logische "1" liegen. Beim Schreiben geschieht (fast) dasselbe. Der einzige Unterschied: Der logische Pegel erscheint am Daten-Eingang. Und der \overline{WRITE} -Eingang liegt auf "0". Der zeitliche Ablauf der Signale läßt sich am besten aus dem Impulsdiagramm in Bild 3 ablesen.

Refresh:

Die notwendige "Aufmunterung"

Wie schon erwähnt, hat die Verwendung von Kondensatoren als Speicherelemente nicht nur Vorteile sondern auch Nachteile. Die Ladung eines solchen Kondensators "leckt". Sie läßt mit der Zeit nach, so daß die geladene Information langsam aber sicher verlorengeht. Die Information



muß aus diesem Grunde von Zeit zu Zeit aufgefrischt werden. Beim 4116-RAM muß dieses Refreshing mindestens einmal in 2 ms stattfinden. Wenn man bedenkt, daß die Zellenkapazität nur 0,04 pF beträgt, handelt es sich um eine vergleichs-

Bild 3. Das Impulsdiagramm für das Lesen, Schreiben und Refreshing des 4116-RAMs. Die Zeiten sind nicht zahlenmäßig angegeben, da sie sich von Typ zu Typ, abhängig von deren "Schnelligkeit", unterscheiden. Die Schnelligkeiten liegen allerdings sämtlich im Nanosekunden-Bereich.

weise lange Speicherzeit.

Das Refreshing ist glücklicherweise aufgrund der "Architektur" des IC ziemlich einfach zu realisieren. Zwischen den Speicherzellen liegen nämlich die schon erwähnten Leseverstärker mit zweifacher Funktion. Eine davon kam bereits zur Sprache: die Verstärkung der logischen Pegel in den Zellen. Wenn die Reihen-Adresse aufgrund eines $\overline{\text{RAS}}$ -Impulses eingelesen wird, dann wird gleichzeitig eine vollständige Reihe von 128 bit in die Leseverstärker eingelesen. Bei diesem Vorgang werden die logischen Pegel verstärkt und wieder in die 128 Reihen-Kondensatoren zurückgegeben. Das bedeutet also: Nach der Ausgabe einer Reihen-Adresse und eines RAS-Impulses sind 128 bit "aufgefrischt" – die zweite Funktion der Leseverstärker! Sorgt man also dafür, daß die gesamte Prozedur (Anbieten einer Reihen-Adresse und Ausgabe eines $\overline{\text{RAS}}$ -Impulses) für alle Reihen innerhalb von 2 ms durchlaufen wird, dann kann man sicher sein, daß die Daten im IC erhalten bleiben.

Es ist selbstverständlich nicht nötig, das Refreshing an den 2-ms-Zyklus zu binden. Es kann auch innerhalb einer sehr viel kürzeren Zeit stattfinden. Die dynamische RAM-Karte wird im Takt des angeschlossenen Mikroprozessor-Systems gesteuert. Für ein 6502-Computer-System (1 MHz) bedeutet das z.B.: Alle 128 Reihen werden innerhalb von 128 μs aufgefrischt.

Das Timing

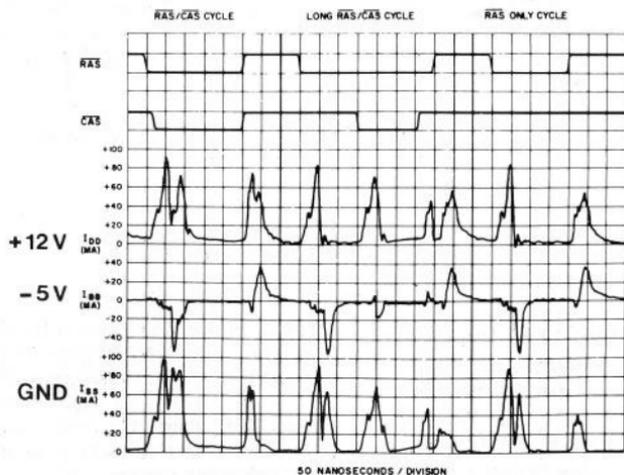
Bild 3 zeigt das Impulsdiagramm für Lesen, Schreiben und Refreshing. Man erkennt deutlich die zeitliche Abfolge der unterschiedlichen Signale. Hier sind auch die Zeiten angegeben, die man bei der Arbeit mit der RAM-Karte unbedingt berücksichtigen muß. Die Werte sind allerdings nicht angegeben, da sie von Hersteller zu Hersteller unterschiedlich sind. Außerdem sind die Zeiten vom verwendeten Typ abhängig. Die Beschreibung der Reihenfolge der Signale erfolgte bereits bei der Erläuterung der IC-Funktion.

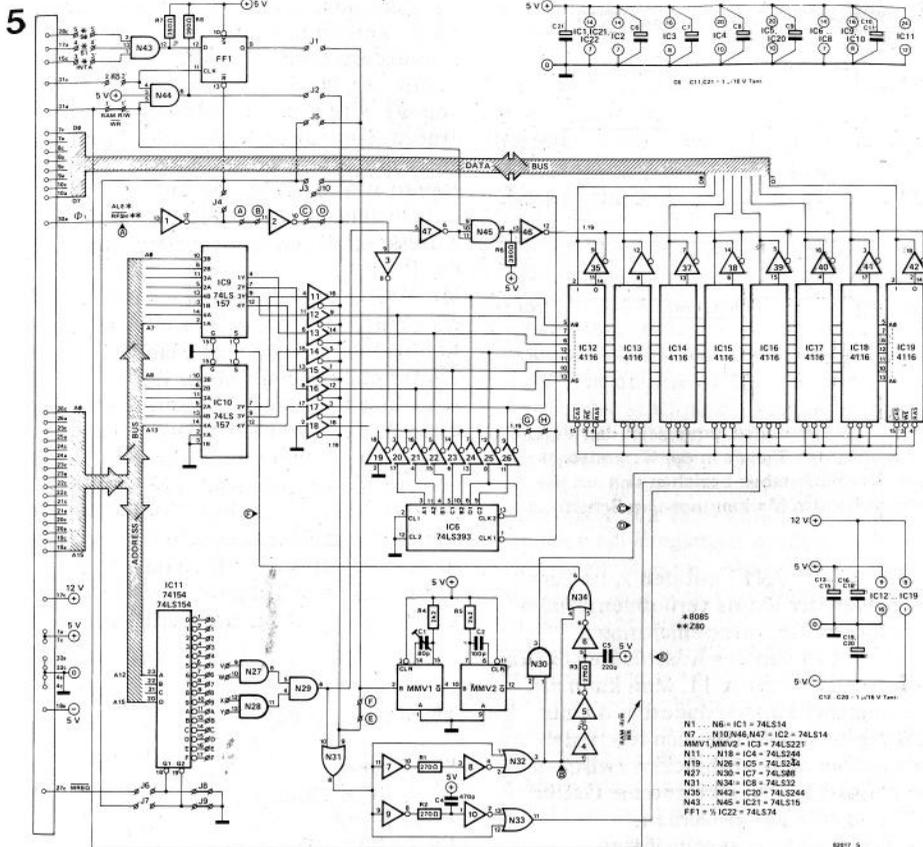
Netzteil: Unempfindlich gegen Spitzenströme

Besondere Aufmerksamkeit muß man dem Netzteil für ein Computer-System mit dynamischer RAM-Karte schenken. Betrachtet man nur die angegebenen Strom-Mittelwerte für die drei Betriebsspannungen, dann erscheinen sie zunächst sehr niedrig. Ein dynamisches RAM braucht allerdings den meisten Strom,

Bild 4. Dieses Diagramm verdeutlicht die Spitzenstrom-Aufnahme bei unterschiedlichen RAS- und CAS-Signalen. Der mittlere Stromverbrauch ist zwar gering. Bei der Entwicklung muß man allerdings Spitzenströme von 100 mA berücksichtigen!

4





wenn ein Impuls am $\overline{\text{RAS}}$ - und/oder $\overline{\text{CAS}}$ -Eingang eintrifft. Bild 4 zeigt ein Beispiel für mögliche Stromspitzen. Dort ist deutlich zu erkennen, daß die verhältnismäßig großen Stromspitzen gerade während der Flanken der Rechteck-Signale auftreten. Die Spitzen erreichen dabei Werte von 100 mA pro IC!

Daraus wird wohl deutlich, daß man bei Verwendung von dynamischen RAM-Karten die Stromspitzen besonders beachten muß. Es ist allerdings nicht erforderlich, das Netzteil größer zu dimensionieren. Viel wichtiger ist die ausreichende Pufferung mittels Kondensatoren direkt an den IC-Pins.

Die Schaltung

In Bild 5 ist die Schaltung eines vollstän-

Bild 5. Die Schaltung der "dynamischen" RAM-Karte. Mit Drahtbrücken kann man die Karte an den Einsatz in unterschiedlichen Mikroprozessor-Systemen anpassen.

digen 16-K-dynamischen-RAM-Speichers dargestellt.

IC12 ... IC19 bilden zusammen den eigentlichen Speicher: 16 K x 8 bit. Die Daten-Eingänge der ICs sind direkt mit den Daten-Anschlüssen der Anschlußleiste verbunden (Bild ganz links). Die Daten-Ausgänge werden über Tri-State-Puffer an die Daten-Leitungen geführt. Die Adressen-Leitungen A0 ... A13 gelangen auf IC9 und IC10, die jeweils vier 2-nach-1-Multiplexer enthalten. Dadurch wird die Menge der Adressen-Leitungen auf sieben reduziert. Diese 7 Leitungen sind über die Tri-State-

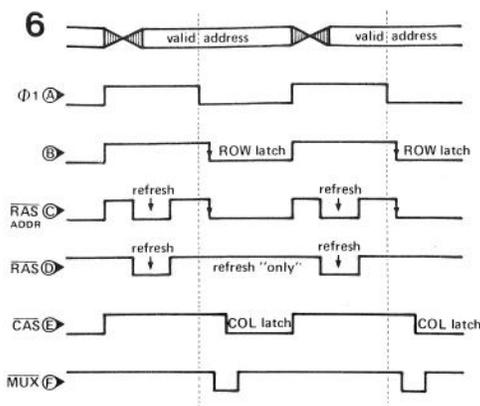


Bild 6. Das Impulsdiagramm zeigt den Verlauf der wichtigsten Signale in der Gesamtschaltung. Die Buchstaben beziehen sich auf die entsprechenden Markierungen im Schaltbild 5.

Puffer N11 . . . N17 mit den Adressen-Eingängen der RAMs verbunden. Außerdem führen die Adressenleitungen A12 . . . A15 von der Anschlußleiste zum Adressen-Dekoder IC11. Man kann also den Speicher einfach dadurch in einen beliebigen Adressbereich legen, daß entsprechende Drahtbrücken zwischen die Ausgänge von IC11 und die Gatter N27 und N28 gelegt werden. Zur Einleitung der regelmäßigen "Erfrischung" der Speicher-ICs dient IC6, das als 7-bit-Zähler geschaltet ist. Die Ausgänge dieses ICs sind über Tri-State-Puffer (N20 . . . N26) ebenfalls mit den Adressen-Eingängen der RAM-ICs verbunden. Während der Zeit, in der der Prozessor den Adreßbus nicht belegt, wird "gerefreshed". Über die Inverter N1, N2 und N3 sind der Takt-Eingang von IC6 und die Steuer-Eingänge der Tri-State-Puffer N20 . . . N26 mit dem Takt $\Phi 1$ des Prozessor-Systems verbunden. Beim Junior-Computer wird, während das Taktsignal $\Phi 1$ "1" ist, der Speicher *nicht* angesprochen. Bei anderen Systemen kann das anders sein. Die eingezeichneten Drahtbrücken gelten für den Betrieb im Junior-Computer-System. Doch zurück zu der Anmerkung, daß der Speicher zu bestimmten Zeiten bestimmt nicht angesprochen wird. Hier nutzen wir diese

Tatsache aus, um den Speicher während dieser Zeit zu refreshen. An Hand des Impulsdiagramms in Bild 6 wird das etwas deutlicher. Nach jeder positiven Flanke von $\Phi 1$ wird der Zähler-Inhalt um Eins erhöht. Die Puffer N11 . . . N17 sind ausgeschaltet, da sie über N2 und N3 angesteuert werden. Über die Puffer N20 . . . N26 gelangt in diesem Fall eine Adresse an die entsprechenden Eingänge der RAM-ICs.

Mit MMV1 und MMV2 ist eine Verzögerung eingebaut, die dafür sorgt, daß kurze Zeit nach der steigenden Flanke des Taktimpulses ein negativer Impuls an die RAS-Eingänge der RAMs gelegt wird. Auf diese Weise frischt ein positiver Taktimpuls eine komplette Reihe auf. Nachdem der Zähler also nach 128 Taktperioden einmal durchgezählt hat, sind alle 128 Reihen aufgefrischt. Ein vollständiger Refresh-Zyklus dauert immer 128 μs bei einer Taktfrequenz von 1 MHz. Die Verzögerungszeiten von

Stückliste zu Bild 5

Widerstände:

- R1 . . . R3 = 270 Ω
- R4, R5 = 2k2
- R6 . . . R8 = 390 Ω

Kondensatoren:

- C1 = 80 p Trimmer
- C2 = 100 p
- C4 = 470 p
- C5 = 120 p
- C6 . . . C21 = 1 $\mu/16$ V Tantal

Halbleiter:

- IC1, IC2 = 74LS14
- IC3 = 74LS221
- IC4, IC5, IC20 = 74LS244
- IC6 = 74LS393
- IC7 = 74LS08
- IC8 = 74LS32
- IC9, IC10 = 74LS157
- IC11 = 74LS154
- IC12 . . . IC19 = 4116 (250 ns)
- IC21 = 74LS15
- IC22 = 74LS74 (siehe Text)

außerdem:

64polige Messerleiste nach DIN 41612

Tabelle 1

6502	Z80	8085
1 - 1'	1 - 1'	1 - 1'
A - B	2 - 2'	2 - 2'
C - D	J2	3 - 3'
E - F	J3	4 - 4'
G - H	J4	5 - 5'
J8	J5	J1
J9	J6	J2
	J9	J4
IC22 entfällt		J6
	IC22 entfällt	J9
		J10

Tabelle 1. Die Übersicht zeigt, welche Drahtbrücken für den verwendeten Prozessor-Typ zu legen sind.

Tabelle 2

Ausgang IC5	Adresse 4-Kbyte-Block	A15	A14	A13	A12
0	0000 ... 0FFF	0	0	0	0
1	1000 ... 1FFF	0	0	0	1
2	2000 ... 2FFF	0	0	1	0
3	3000 ... 3FFF	0	0	1	1
4	4000 ... 4FFF	0	1	0	0
5	5000 ... 5FFF	0	1	0	1
6	6000 ... 6FFF	0	1	1	0
7	7000 ... 7FFF	0	1	1	1
8	8000 ... 8FFF	1	0	0	0
9	9000 ... 9FFF	1	0	0	1
A	A000 ... AFFF	1	0	1	0
B	B000 ... BFFF	1	0	1	1
C	C000 ... CFFF	1	1	0	0
D	D000 ... DFFF	1	1	0	1
E	E000 ... EFFF	1	1	1	0
F	F000 ... FFFF	1	1	1	1

Tabelle 2. Der Adressen-Bereich wird durch die Wahl der Verbindungsleitungen zwischen den Ausgängen von IC5 und den Punkten V, W, X und Y festgelegt. Jede Verbindung bezeichnet einen Block von 4 KByte. Möchte man 16 K benutzen, dann sind also 4 Verbindungen herzustellen.

MMV1 und MMV2 sind so gewählt, daß die Refresh-Zeit sicher in die Mitte der positiven Taktzeit fällt.

Das Ein- und Auslesen einer Adresse geschieht immer dann, wenn der $\Phi 1$ -Takt negativ ist. Die zur Steuerung dieses Vorgangs notwendige "Timing-Logik" erfordert eine genaue Dimensionierung. Denn die unterschiedlichen Flanken müssen unbedingt in der richtigen Reihen-

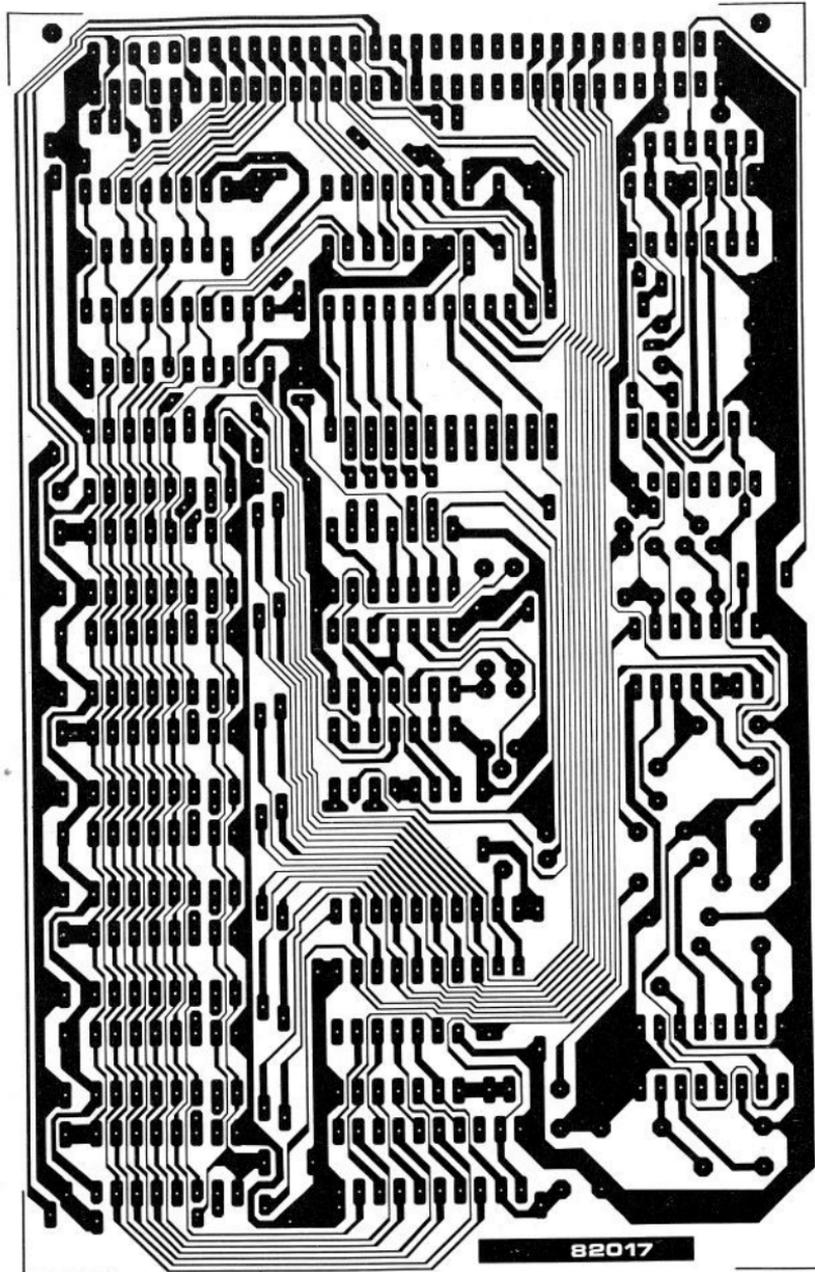
folge bei den RAM-ICs und den Multiplexern eintreffen. Diese Timing-Logik ist mit N4 ... N10, R1 ... R3 und C3 ... C5 aufgebaut.

Liegt eine Adresse zur Abfrage der RAMs an, dann wird der Ausgang von N29 aufgrund der Stellung des Adressendekoders logisch 0. Über N31, an dem auch $\Phi 1$ liegt, gelangt das Takt-Signal dann nach N7 und N9. Die abfallende Flanke eines Takt-Impulses wird von R1/C3 verzögert und über einige Gatter zu den \overline{RAS} -Eingängen geführt (siehe auch Bild 6). Das bedeutet: Die ersten sieben Adreßbits werden in die RAMs gelesen. Danach werden die Multiplexer von der durch R3/C5 verzögerten, abfallenden \overline{RAS} -Flanke umgeschaltet. Jetzt liegen also die nächsten sieben Adreßbits an den RAM-Eingängen. Mit einer abfallenden Flanke an den \overline{CAS} -Eingängen werden auch sie eingelesen. Diese Flanke wird ebenfalls aus der abfallenden Flanke des Takt-Impulses abgeleitet und von R2/C4 verzögert. Die \overline{WE} -Eingänge sind direkt mit den entsprechenden Pins der Anschlußleiste verbunden. Die bisher nicht erwähnten Gatter und das Flipflop im Schaltbild dienen zur Anpassung der RAM-Karte an ein anderes als ein 6502-Mikroprozessor-System.

Der Aufbau

In Bild 7 sind Layout und Bestückungsdruck der Platine dargestellt. Es sollte selbstverständlich sein, daß man bei einer solchen, doch relativ komplizierten Schaltung auch den Aufbau sehr sorgfältig vornimmt. Vor allen Dingen muß darauf geachtet werden, daß man nur die angegebenen Bauteile-Werte einsetzt. Wie gesagt: Das Timing ist hier besonders kritisch, und ein falscher Kondensatorwert wird sicherlich einiges durcheinanderbringen.

Der Einsatz verschiedener Drahtbrücken ist vom verwendeten Prozessor-Typ abhängig. Tabelle 1 ordnet die Drahtbrücken den unterschiedlichen Prozessoren zu. Bei 6502- und Z80-Systemen muß IC22 entfallen, da Flipflop FF1 nicht benötigt wird. Bei 8085-Systemen muß man einige



Punkte berücksichtigen: Im Unterschied zum Z80 liefert der 8085 nämlich kein Refresh-Signal. Dieses Signal gewinnt man in diesem Fall aus S_0 , S_1 und INTA (sie geben den Opcode-Fetch an). In der Zeit, die der Prozessor benötigt, um den Code

zu "entdecken", werden die RAMs aufgefrischt. Außerdem verfügt der 8085 über einen gemultiplexten Daten-Bus, der also auch Adressen "enthält". Die dynamische RAM-Karte ist allerdings nicht für einen solchen Bus geeignet. Beim 8085

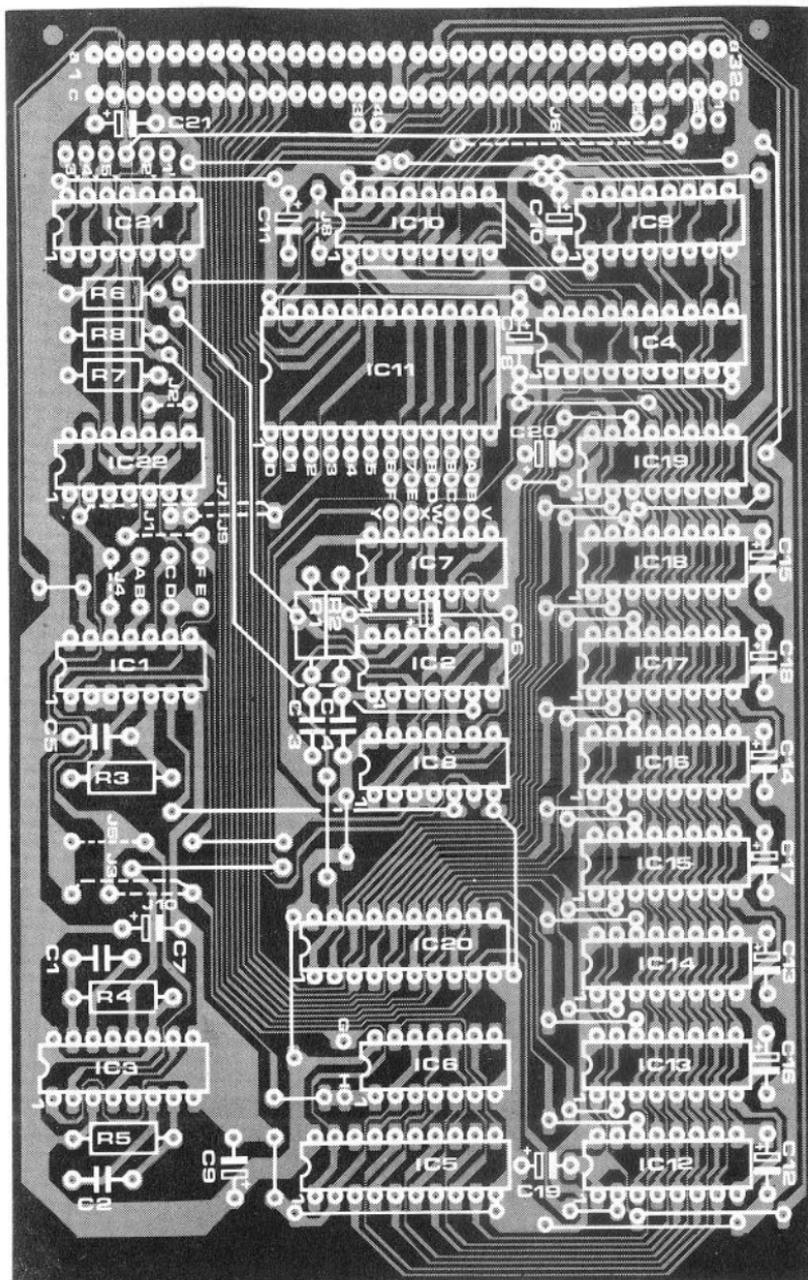


Bild 7. Layout und Bestückungsdruck der Platine für die "dynamische" RAM-Karte. Die Drahtbrücken sind gemäß Tabelle 1 zu legen. Um bei den Masseleitungen einen unzulässig hohen Spannungsverlust zu vermeiden, sind auf der Leiterbahnseite noch 4 zusätzliche Masseverbindungen notwendig.

muß deshalb der Datenbus bereits irgendwo im System "entmultiplext" sein.

Die Verbindungen zwischen den Punkten V, W, X, Y und den Ausgängen von IC11 legen den Adressbereich fest. Jeder

Tabelle 3

```

0010: 0004          ORG 0004
0020:
0030:
0040:          *** RAM TEST PROGRAM ***
0050:
0060:
0070:          DEFINITIONS
0080:
0090: 0004          BEG * 0000 BEGIN OF MEMORY
0100: 0004          END * 0002 END OF MEMORY
0110: 0004          CUR * 0006 CURRENT ADDRESS POINTER
0120: 0004          POINT * 000A MONITOR'S ADDRESS POINTER
0130: 0004          PATTERN * 000E CURRENT TEST PATTERN
0140: 0004          MONIT * 0100
0150:
0160:
0170: 0004 20 45 00 RAMTST JSR WZERO FILL WORKSPACE WITH 000
0180: 0007 20 54 00 JSR CURBEG CUR = BEG
0190:
0200: 000A 20 84 00 TSTA JSR WALK WALKING BIT ROUTINE
0210: 000D 00 20 BNE T5C BRANCH IF MEMORY CELL IS DEFECT
0220: 000F A9 FF LDAIM SFF TEST PATTERN FOR DOUBLE ADDRESSING
0230: 0011 01 E6 STAYI CUR
0240: 0013 20 50 00 JSR INCCHK INCREMENT AND CHECK CUR
0250: 0016 00 F2 BEC T5A TEST FINISHED
0260: 0018 20 45 00 JSR WZERO FILL WORKSPACE WITH 000
0270: 001B A6 02 LDA END CHECK FROM BOTTOM TO TOP
0280: 001D 06 E6 STX CUR
0290: 001F A6 03 LDA END +01
0300: 0021 06 E7 STX CUR +01
0310:
0320: 0023 20 84 00 T5B JSR WALK
0330: 0026 00 F2 BNE T5C BRANCH IF MEMORY CELL IS DEFECT
0340: 0028 A9 FF LDAIM SFF TEST PATTERN FOR DOUBLE ADDRESSING
0350: 002A 01 E6 STAYI CUR
0360: 002C 20 49 00 JSR DECCHK DECREMENT AND CHECK CUR
0370: 002F 00 F2 BEC T5B
0380: 0031 A9 00 LDAIM 000 DISPLAY "0000 XX" IF
0390: 0033 85 FA S'A POINT MEMORY IS O.K.
0400: 0035 85 FB S'A POINT +01
0410: 0037 4C 10 JC JMP MONIT0
0420:
0430: 003A A5 E6 T5C LDA CUR DISPLAY THE ADDRESS OF
0440: 003C 85 FA S'A POINT THE DEFECT MEMORY CELL
0450: 003E A5 E7 LDA CUR +01
0460: 0040 85 FB S'A POINT +01
0470: 0042 4C 10 JC JMP MONIT0
0480:
0490:
0500:          SUBROUTINES
0510:
0520:
0530: 0045 20 54 00 WZERO JSR CURBEG FILL THE MEMORY BETWEEN BEG & END
0540: 0048 A0 00 LDYIM 000 WITH 000
0550:
0560: 004A A9 00 WRZ LDAIM 000
0570: 004C 01 E6 STAYI CUR
0580: 004E 20 50 00 JSR INCCHK
0590: 0051 00 F7 BEC WRZ
0600: 0053 00 RTS
0610:
0620: 0054 A6 00 CURBEG LDA BEG CUR = BEG
0630: 0056 86 E6 STX CUR
0640: 0058 A6 01 LDA BEG +01
0650: 005A 86 E7 STX CUR +01
0660: 005E 00 RTS
0670:
0680: 0059 E6 E6 INCCHK INC CUR CUR = CUR+01
0690: 005F 00 02 BNE IA
0700: 0061 E6 E7 INC CUR +01
0710:
0720: 0063 38 IA SEC C=0 IF CUR >END
0730: 0064 A5 02 LDA END
0740: 0066 E5 E6 SBC CUR
0750: 0068 A5 03 LDA END +01
0760: 006A E5 E7 SBC CUR +01
0770: 006C 00 RTS
0780:
0790: 006D 38 DECCHK SEC CUR = CUR -01
0800: 006E A5 E6 LDA CUR
0810: 0070 E9 01 SBCIM 01
0820: 0072 85 E6 S'A CUR
0830: 0074 A5 E7 LDA CUR +01
0840: 0076 E9 00 SBCIM 000
0850: 0078 85 E7 S'A CUR +01
0860: 007A 38 SEC C=0 IF CUR < BEG
0870: 007B A5 E6 LDA CUR
0880: 007D E5 00 SBC BEG
0890: 007F A5 E7 LDA CUR +01
0900: 0081 E5 01 SBC BEG +01
0910: 0083 00 RTS
0920:
0930: 0084 A9 01 WALK LDAIM 001 INIT. PATTERN
0940: 0086 85 E5 S'A PATTERN
0950: 0088 A0 00 LDYIM 000
0960: 008A 01 E6 LDAI CUR IS STILL 000 IN THE CELL
0970: 008C 00 0F BNE WALKB IF NO, THEN BRANCH
0980: 008E A2 08 LDAIM 008 WALKING BIT COUNTER
0990:
1000: 0090 A5 E5 WALKA LDA PATTERN CURR. PATTERN INTO ACCU
1010: 0092 01 E6 STAYI CUR STORE IT IN MEMORY
1020: 0094 01 E6 CMPIY CUR DOES IT MATCH?
1030: 0096 00 05 DNE WALKB IF NO, THEN BRANCH
1040: 0098 06 E5 ASL PATTERN WALKING BITS!
1050: 009A 0A DEX
1060: 009C 00 03 BNE WALKA
1070:
1080: 009D 60 WALKB RTS
1090:
1100:

```

Tabelle 3. Ein Test-Programm für die Speicher-Karte. Start- und Ende-Adresse des zu überprüfenden Bereichs werden auf die Plätze 0000 . . . 0003 gesetzt. Das Programm beginnt bei 0004. Vor dem ersten Programmstart muß der Trimmer C1 in Mittelstellung stehen.

Ausgang des ICs entspricht einem Adressbereich von 4 Kbyte. Tabelle 2 zeigt die Einteilung des Bereiches. Da der dynamische Speicher 16 Kbyte groß ist, werden also vier Ausgänge von IC 11 mit den Punkten V, W, X und Y verbunden. Man kann den Speicher auf diese Weise in 4-K-Blöcken auf fast jede Stelle legen. Dabei ist allerdings darauf zu achten, daß nicht zweimal derselbe Code für A12 und A13 gewählt werden darf (siehe auch letzte Reihe von Tabelle 2). Das ist deshalb notwendig, weil diese beiden Adressen-Leitungen auch an den Adressen-Eingängen der RAMs liegen. Die vier Drahtbrücken müssen also immer so gelegt werden, daß die folgenden Kombinationen von A13 und A12 auftreten:

A13/A12 → 0/0; 0/1; 1/0; 1/1.

Aus dieser Angabe kann man leicht ablesen, welche Kombinationen erlaubt sind. *Gut*: z.B. 8000, 9000, A000 und B000. *Schlecht*: z.B. 0000, 4000, 8000 und C000, da hier in *allen* Blöcken die Kombination A13/A12 → 0/0 auftritt. Wie immer bei diesen "Filigran-Layouts" empfehlen wir auch hier, IC-Fassungen und einen "feinen" LötKolben zu verwenden. Achtung: Kalte Lötstellen sind besonders schlecht zu finden!

Wird die Karte im Junior-Computer-System eingesetzt, dann sind die benötigten Betriebsspannungen bereits vorhanden. Sonst kann man diese Spannungen mittels integrierter Spannungsregler selbst erzeugen. In Elektor findet man zahlreiche solcher Netzteil-Schaltungen.

TEST, TEST, TEST

Bevor man die Betriebsspannungen anschließt, sollte der Aufbau noch einmal sorgfältig kontrolliert werden; Trimmer C1 muß in Mittelstellung stehen. Erst danach kann die Karte auf den Bus eines Prozessor-Systems gesetzt werden. Die Reihenfolge des Anschlusses der Betriebsspannungen ist nicht kritisch, obwohl der Hersteller empfiehlt, erst die -5-V-Spannung einzuschalten. Damit soll verhindert werden, daß eventuelle Überspannungen des Netzteils den ICs schaden

können. Bei einem *gut* stabilisierten Netzteil braucht man allerdings auf diese Einschalt-Reihenfolge nicht zu achten. Beim Einschalten aller Betriebsspannungen sollte die Speicher-Karte jedenfalls sofort betriebsbereit sein. Diese Betriebsbereitschaft kann man ihr zwar nicht so ohne weiteres von außen ansehen. Aber ein kleines Test-Programm hilft da weiter. Tabelle 3 enthält ein solches Programm für den Junior-Computer. Man kann es übrigens auch zum Testen anderer RAM-Speicher verwenden. Nachdem das Programm eingegeben wurde, kann man auf 0000 (= ADL) und 0001 (= ADH) die Start-Adresse und auf 0002 (= ADL) und 0003 (= ADH) die End-Adresse des zu prüfenden Speicher-Bereichs setzen. Danach wird das Programm auf Adresse 0004 gestartet: Der Speicher wird mit 00 gefüllt. Anschließend überprüft das Programm, ob tatsächlich in der ersten Adresse des Bereichs 00 steht. Ist das der Fall, dann werden nacheinander 01, 02, 04, 08, 10, 20, 40 und 80 in diese Adresse geschrieben und danach wieder ausgelesen. Das bedeutet: Jedes Bit einer Adresse war einmal "1". Schließlich wird zum Aufspüren von Adressierungsfehlern noch FF in die Adresse gesetzt. Tritt nämlich irgendwo ein solcher Fehler auf, dann erscheint außerdem FF in einer anderen Adresse. Man kann den Adres-

Tabelle 4

Brücken auf der RAM-Karte: 1-1', 2-2', A-B, J2, J3, J5, J6, J9

IC22 entfällt!

Verbindungen auf der RAM-Karte:
5' nach +5 V, 3' nach C

Verbindungen vom Interface zur RAM-Karte:
Pin 13/MMV1 nach 4', Pin 9/MMV2 nach Pin/N43 12, Pin 1/MMV1 nach J4-A
Pin 12/N1, Pin 5/MMV2 nach J4-J3-J5,
Pins 3, 11, 10, 16, R1 und R2 nach +5 V,
Pin 8 nach Masse

Tabelle 4. Anpassungshinweise für den Gebrauch der RAM-Karte in Zusammenhang mit dem SC/MP.

sierungsfehler also leicht daran erkennen, daß irgendwo anders FF ausgelesen wird.

Die beschriebene Prozedur wird für jede Adresse durchgeführt, so daß das Programm bis zum Ende des Bereichs kommt. Dort angelangt, beginnt das "Spielchen" noch einmal von hinten nach vorne: Zunächst wird der Bereich mit $\emptyset\emptyset$ gefüllt und so weiter. Die Suche nach Adressierungsfehlern wird also auch noch einmal durchgeführt.

Hat das Programm keinen Fehler entdeckt, dann erscheint nach Ablauf Adresse $\emptyset\emptyset\emptyset$ mit dem niederwertigen Adreßbyte der eingegebenen Startadresse auf dem Display. Wird ein Fehler entdeckt, dann gibt das Display die fehlerhafte Adresse und den falschen Inhalt dieser Adresse an. Hat man den Fehler korrigiert und möchte das Programm weiterlaufen lassen, dann genügt die Eingabe der Adresse $\emptyset\emptyset A$ samt Start.

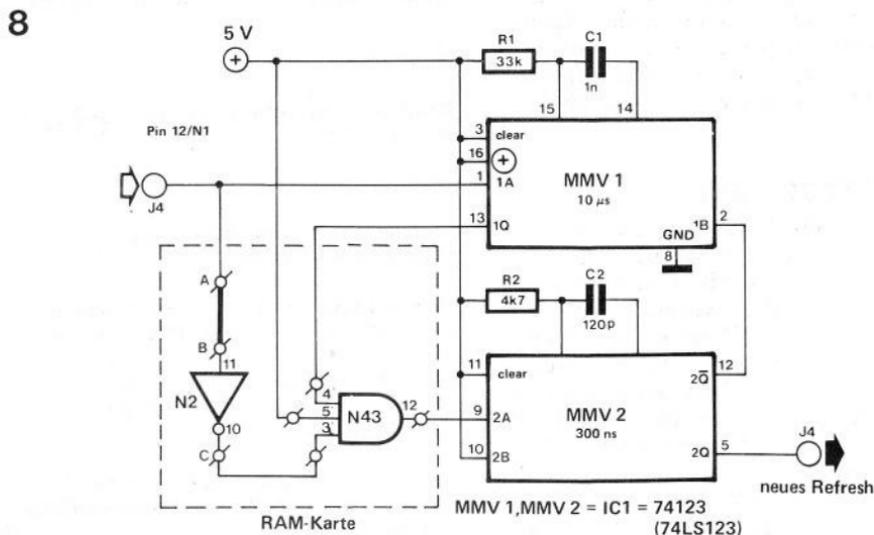
Zeigt das Programm sofort nach dem Start einen Fehler, dann ist die Einstellung des Trimmers C1 nicht korrekt. Das Refreshing erfolgt dann nicht zur rechten Zeit. In diesem Fall genügt es, Trimmer C1 anders einzustellen und das Programm erneut zu starten. In der Regel ist die Einstellung von C1 nicht kritisch und die dynamische RAM-Karte müßte

in einem relativ großen Einstellbereich fehlerfrei funktionieren. Der Abgleich von C1 ist dann abgeschlossen, wenn der Trimmer in der Mittelstellung des Einstellbereiches steht. Ist trotz mehreren Versuchen kein korrekter Abgleich möglich, bleibt nichts anderes übrig, als die Platine auf eventuelle Leiterbahnverbindungen, Lötbrücken, Montagefehler und defekte Bauelemente zu überprüfen.

SC/MP

Tabelle 1 zeigt, wie die dynamische RAM-Karte mit den Prozessortypen 6502, Z80 und 8085 zu verbinden ist. Andere Prozessoren werden zwar nicht erwähnt, können aber trotzdem mit dem dynamischen Speicher arbeiten. Dabei ist allerdings eine wichtige Tatsache von Bedeutung: Der Prozessor darf nicht durch einen Befehl abgeschaltet werden können, wie es z.B. das HALT-Kommando mit dem SC/MP praktiziert. Dadurch fällt auch der Refresh-Befehl weg und das führt dazu, daß Daten verlorengehen. In solchen Fällen muß eine Hilfsschaltung den Refresh-Befehl ersetzen. Die Schaltung in Bild 8 zeigt,

Bild 8. Interfaceschaltung für die Verbindung zwischen RAM-Karte und SC/MP.



das es hierzu keine großen Klimmzüge bedarf.

Das Interface kommt mit nur einem IC, zwei Widerständen und zwei Kondensatoren aus. Sie findet bei der dynamischen RAM-Karte ihren Anschluß zwischen den Punkten J4. Außerdem sind die in der Tabelle angegebenen Brücken und zusätzlichen Verbindungen zu legen. Die Schaltung enthält ein retriggerbares Monoflop, dessen Haltzeit auf etwa $10 \mu\text{s}$ eingestellt ist (MMV1). Treffen ständig NADS-Impulse ein, dann ist der Ausgang 1Q immer "1". Über N2 und N43 (bereits auf der RAM-Karte) wird das zweite

Monoflop "vorbereitet".

Bei jedem NADS-Befehl steht an 2Q ein 300 ns langer Impuls als Refresh-Befehl zur Verfügung. Treffen innerhalb von $10 \mu\text{s}$ keine NADS-Impulse ein, dann wird 1Q "0", über N43 wird das zweite Monoflop getriggert. Das Refresh-Signal gelangt über den Ausgang 2Q auch an MMV1 und triggert auf diese Weise das Monoflop nach. Ausgang 1Q wird $10 \mu\text{s}$ lang "1". Die Speicher werden also innerhalb von $10 \mu\text{s} \times 128 = 1,28 \text{ ms}$ ge"refreshed". Die Schaltung ist übrigens auch für andere Systeme mit "manuellem" Reset geeignet.

Kapitel 4: Programmiergeräte und anderes

2716-Programmiergerät

Programmieren ohne großen Aufwand

Das EPROM ist ein Erasable Programmable Read Only Memory, ein programmierbarer Festwertspeicher. Dieser hat gegenüber den anderen nicht flüchtigen Speicherbausteinen den angenehmen Vorteil, daß sich die Daten in ihnen wieder mit UV-Licht löschen lassen, wenn man ein anderes Programm abspeichern will. EPROMs finden in der letzten Zeit immer häufiger Verwendung, und das nicht nur in Computersystemen. Meist stellt sich für den Anwender immer wieder die Frage: "Wie bekomme ich mein EPROM programmiert?" Nun, wenn es sich um ein 2716 handelt, mit dem hier beschriebenen Entwurf. Ein Timer-IC 555, ein TTL-IC, einige diskrete Bauelemente und einer von mehreren möglichen Mikroprozessoren genügen, um das einfache EPROM-Programmiergerät aufzubauen.

Der Preisverfall beim 2716 ist enorm. Vor Jahresfrist mußte man für dieses EPROM noch ca. 80 DM auf den Ladentisch blättern, während heute ca. 15 DM genügen. Ein weiterer Vorteil kommt hinzu: das IC benötigt nur noch eine Speisepannung. Dadurch ist es relativ einfach, das EPROM in bestehende μ P-Systeme

einzuführen oder den 2708 dagegen auszutauschen. Was nun noch fehlt, ist eine einfache Programmiermöglichkeit für den 2716. Die Programmierbedingungen lassen sich aus dem Datendiagramm (Bild 1) leicht ableiten: es ist lediglich eine Programmierspannung von 25 V und ein Programmierimpuls mit normalem TTL-Niveau und 50 ms Dauer notwendig. Während der Programmierung dürfen die Signale \overline{CE} und \overline{OE} nicht aktiviert sein. Um zu überprüfen, ob die vorgenommene Programmierung in Ordnung ist, läßt man die Daten aus dem EPROM einfach wieder ausschreiben, ohne daß die Programmierspannung extra abgeschaltet werden muß.

In Verbindung mit bereits bestehenden μ P-Systemen (z.B. 2650, 8085) ist die Programmierung für den 2716 sehr einfach. Der vorhandene Speicherbaustein wird aus der Fassung entfernt, in die dann

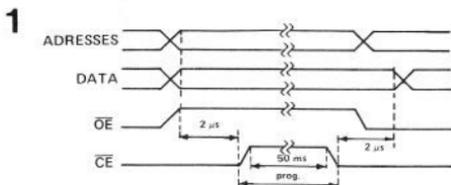


Bild 1. Ein EPROM 2716 läßt sich relativ einfach programmieren. Wenn die Programmierspannung von 25 V anliegt, genügt es, den \overline{OE} -Eingang für 50 ms auf logisch 1 zu legen, während der \overline{CE} -Eingang ebenfalls logisch 1 sein muß.

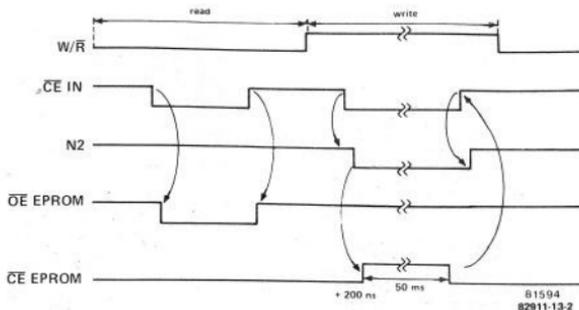


Bild 2. Das Programmiergerät erzeugt die notwendigen Steuersignale für die Programmierung des EPROMs 2716. Bei einem Read-Zyklus wird das \overline{OE} -Signal aktiviert (log. 0) und das \overline{CE} -Signal bleibt auf niedrigem Pegel (log. 0). Werden nun die Daten mit einem Write-Zyklus eingelesen, müssen beide Signale während der Programmierung logisch 1 sein.

eine kleine Hilfsschaltung (Bild 3) Platz findet. In die Hilfsschaltung selbst wird dann auch das ganze oder teilweise zu programmierende EPROM gesteckt. Nachdem dann die Hilfsschaltung noch mit vier zusätzlichen Verbindungen versehen ist, steht der Programmierung des 2716 nichts mehr im Wege. Die Programmierung selbst wird mit einem ganz normalen Write-Signal an die richtige Adresse gebracht. Ein anschließendes Read-Signal gibt Aufschluß darüber, ob die richtigen Daten abgespeichert sind. Danach folgt die Programmierung des nächsten (oder auch eines willkürlichen) Speicherplatzes. Selbstverständlich ist es auch möglich, nur einen einzigen Speicherplatz zu programmieren. Damit das alles so wie beschrieben funktioniert, muß der Prozessor eine Bedingung erfüllen: Während der Write-Instruktion muß man ihn für die erforderlichen 50 ms anhalten können. Der Eprommer ist für den Prozessor 2650 von Signetics vorgesehen, der hierfür einen \overline{OPACK} -Eingang hat. Handelt es sich um einen anderen Prozessor, kann man die Schaltung immer dann gebrauchen (meist mit kleinen Änderungen), wenn ein "Stopp-Eingang" vorhanden ist. Bei der

CPU 8085 ist das der Eingang READY. Die CPU 6502 läßt sich während des Write-Zyklus nicht anhalten. Der SC/MP hat zwar keinen \overline{OPACK} -Eingang, aber einen ähnlichen. Es ist der NHOLD-Anschluß, der allerdings gegenüber den \overline{OPACK} -Eingängen ein invertiertes Signal benötigt. Fazit: Das Programmiergerät ist auch für ein SC/MP-System geeignet. Das \overline{OPACK} -Signal hält den Prozessor an; dabei bleiben die Signale auf dem Daten- und Adreßbus unverändert. Während der 50 ms bleibt also das Adreß- und Datenbyte am EPROM stabil. In der Hilfsschaltung wird das \overline{OPACK} -Signal mit dem bekannten Timer-IC 555 erzeugt. Sobald die EPROM-Adresse aufgerufen ist, startet der gesamte Programmierzyklus. Infolgedessen geht das \overline{CE} -Signal auf logisch 0 (es ist dann aktiviert, Bild 2). Bei einem Read-Zyklus bleibt das

Tabelle 1

PINS \ MODE	\overline{CE}/PGM (18)	\overline{OE} (20)	V_{PP} (21)	V_{CC} (24)	OUTPUTS (9-11, 13-17)
Read	V_{IL}	V_{IL}	+5	+5	DOUT
Standby	V_{IH}	Don't care	+5	+5	High Z
Program	pulsed V_{IL} to V_{IH}	V_{IH}	+25	+5	DIN
Program Verify	V_{IL}	V_{IL}	+25	+5	DOUT
Program Inhibit	V_{IL}	V_{IH}	+25	+5	High Z

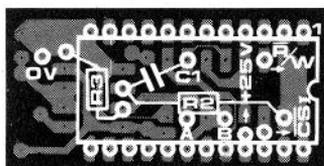
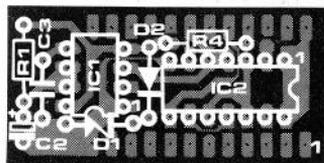
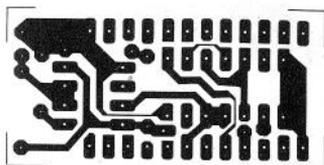


Bild 4. Platinenlayout und Bestückungsplan.
Achtung: Es dürfen bei den Anschlußpins 20 und 21 des DIL-Steckers keine leitenden Verbindungen entstehen. Das gilt auch für Pin 18, der über einen separaten Draht mit Pin 2 von IC2 zu verbinden ist. Im Platinaufdruck ist dieser Punkt mit $\overline{CS1}$ gekennzeichnet.

Stückliste

Widerstände:

- R1 = 180 k
- R2 = 150 k
- R3 = 22 k
- R4 = 1k8

Kondensatoren:

- C1 = 1 n
- C2 = 0,22 μ /16 V Tantal
- C3 = 10 n

Halbleiter:

- D1 = Z-Diode 5V6/0,4 W
- D2 = 1N4148
- IC1 = 555
- IC2 = 74LS00

außerdem:

- 24-polige IC-Fassung
- 24-poliger IC-Stecker

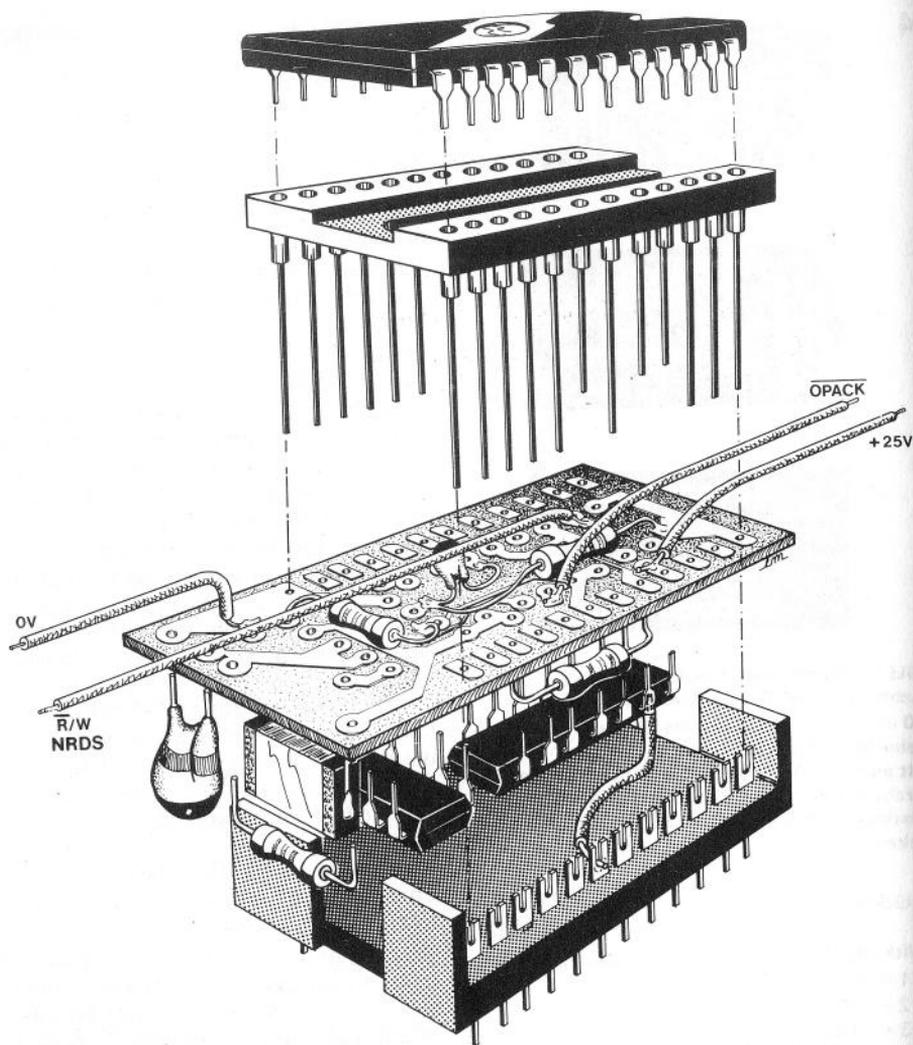
steht nun das \overline{CE} -Signal zur Verfügung; es ist für 50 ms logisch 1. Somit sind die Bedingungen für die Speicherung der anliegenden Daten erfüllt. Dies geht noch einmal aus Tabelle 1 hervor: Programmspannung = 25 V; \overline{OE} = log. 1 und \overline{CE} = log. 1.

Das Steuersignal \overline{OPACK} für den Prozessor kommt ebenfalls von IC 1. Es geht nach 50 ms wieder auf logisch 0 und der Prozessor setzt sein Programm fort.

Der gesamte Programmiervorgang läuft also folgendermaßen ab:

- Hilfsschaltung anstelle des EPROMs in die IC-Fassung stecken; EPROM in die Hilfsschaltung stecken und über zusätzliche Leitungen mit $\overline{R/W}$, \overline{OPACK} , der 25 V-Leitung (30 mA sind ausreichend) und Masse verbinden.
- Bei einem System mit der CPU 2650 ist das Programm in Tabelle 2 angegeben. Das ist die gesamte Software, die für die Programmierung notwendig ist.
- Abschließend werden die zusätzlichen Verbindungen zur 25 V-Leitung, Masse, $\overline{R/W}$ und \overline{OPACK} wieder gelöst, die Hilfsschaltung entfernt und das programmierte EPROM in die entsprechende IC-Fassung gesteckt. Einfacher geht es wohl kaum noch. Nun noch ein paar Tips, damit der Eprommer mit dem bestehenden System richtig zusammenpaßt:

Die Signale, die zum Eprommer gelangen, müssen zeitlich aufeinander abgestimmt sein. Das $\overline{R/W}$ -Signal muß vor oder zumindest gleichzeitig mit dem \overline{CE} -Signal anwesend sein. Das bedeutet für den 8085, es darf nicht \overline{WR} , sondern es muß S1 benutzt werden. Umgekehrt gilt auch, daß man \overline{CE} nicht mit \overline{WR} oder $\overline{R/W}$ kombinieren darf (beim 2650). Das ergibt sich aus der Forderung, daß man nur aus "gültigen Adressen" das \overline{CE} -Signal ableiten darf. Beim 2650 z.B. ist die Kombination \overline{OPREQ} und M/IO möglich. Nach jeder Programmierung muß die Schaltung in den Ausgangszustand zurückkehren können. Für diese Zeit sind mindestens 2,5 ms erforderlich, die im Programmbeispiel durch eine Verzögerungsschleife (Delay) berücksichtigt



ist. Verantwortlich für die notwendige Verzögerung ist in erster Linie das RC-Glied $C1/R2$. Die theoretische \overline{CE} -Impulszeit beträgt mit den angegebenen Werten von $R1$ und $C2$ 45 ms. Stehen Meßgeräte zur Verfügung (Frequenzmesser, Oszilloskop), kann man durch einen höheren Wert von $C1$ die Impulszeit etwas ausdehnen. Bei einem fehlenden Meßgerätepark sollte man allerdings auf diese Möglichkeit verzichten und mit der kürzeren Impulszeit von 45 ms vorliebnehmen. Die in Bild 2 abgebildete Signalfolge ist nicht korrekt dargestellt, da eine Delay-

Bild 4. Deutlicher als ein Foto oder eine ausführliche Beschreibung zeigt diese Zeichnung, worauf es ankommt. Erst die Platine auf beiden Seiten mit ebensolangen Stücken von blankem Montage draht versehen und verlöten! Anschließend kürzt man die Drähte auf das entsprechende Maß. Verwendet man für die IC-Fassung eine Wirewrap-Ausführung mit langen Anschlußpins, können die Drähte entfallen.

zeit von $2 \mu\text{s}$ nicht eingezeichnet ist (siehe Bild 1). Auch im Schaltungsaufbau sind die $2 \mu\text{s}$ nicht berücksichtigt. Im praktischen Test ist jedoch dadurch kein Nachteil entstanden, doch ist eine

Tabelle 2

0100	xx xx	Startadresse Daten		
0102	xx xx	Startadresse EPROM		
0104	xx	Anzahl Bytes		
76 60		PPSU	Disable Interrupt	} ist das EPROM leer?
0D 01 04		LODA	Anzahl Bytes in R1	
0D C1 02		LODA	} ist EPROM-Byte FF	
E4 FF		COMI		
98 17		BCFR	wenn \neq FF: Error	
59 77		BRNR	zur nächstfolgenden Adresse	
74 40		CPSU	Flag = 0	} EPROM programmieren
09 F1		LODR	Anzahl Bytes in R1	
04 FF		LODI	} 2,5 ms warten	
F8 7E		BDRR		
0D C1 00		LODA	Daten laden	
CDE1 02		STRA	Daten zum EPROM	
EDE1 02		COMA	ist EPROM richtig programmiert?	
98 02		BCFR	wenn Fehler: Error	
59 6F		BRNR	nächstes Byte ins EPROM	
40		HLT		

hundertprozentige Programmierung nicht in jedem Fall mehr gewährleistet.

Das Timer-IC 555 verursacht zwischen dem Eingangssignal an Pin 2 und dem Ausgangssignal an Pin 3 eine Verzögerung von mindestens 200 ns. Treffen nun mehrere ungünstige Zeitfaktoren zusammen, dann kann das $\overline{\text{OPACK}}$ -Signal zu spät zum Prozessor gelangen und die Programmierung ist dann nicht mehr möglich. Bei einer CPU 2650 (Taktfrequenz 1 MHz) muß das $\overline{\text{OPACK}}$ -Signal innerhalb von 600 ns anliegen. In diesem Fall ist also genügend Zeit vorhanden. Die CPU 8085 arbeitet mit einer Taktfrequenz von 3 MHz und läßt dem $\overline{\text{OPACK}}$ -Signal nur 100 ns Zeit. Das kann eventuell bereits zu kurz sein, was aber nicht unbedingt sein muß. Im ersten Fall bleibt nichts anderes übrig, als die Taktfrequenz herabzusetzen.

Betrachtet man in verschiedenen Datenbüchern die Pinbelegung des 2716, ist die Verwirrung perfekt. Sie machen näm-

lich unterschiedliche Angaben, welcher Pin der OE- und der $\overline{\text{CE}}$ -Eingang ist. Halten Sie sich an die vom Hersteller gemachten Angaben, dann wird das eine und andere EPROM korrekt arbeiten. Das gilt nicht im Zusammenhang mit der 8-K-RAM/8-K-EPROM-Karte. Hier bestätigt die Ausnahme wieder die Regel. Auf dieser Karte ist Pin 18 ($\overline{\text{CE}}$) der EPROMs mit Masse verbunden, so daß die beschriebenen Bedingungen nicht einzuhalten sind. Trotzdem kann das Programmiergerät auch an die RAM/EPROM-Karte angepaßt werden und zwar mit Hilfe von Anschluß 20, dem $\overline{\text{CS}}$ -Eingang (chip select). Außerdem ist dafür Sorge zu tragen, daß nur während des Signals "Adresse ist gültig" ein $\overline{\text{CS}}$ -Signal anliegt. Das ist auf der erwähnten Karte nicht der Fall. In Verbindung mit dem SC/MP-System sind hierfür die Signale NRDS und NWDS geeignet. Sie werden auf der RAM/EPROM-Karte den Anschlüssen 18 und 19 von IC5 zugeführt.

Programme für SC/MP und Junior selbst "schießen"

Hier wird ein EPROM-Programmiergerät für die Elektor-Hauscomputer SC/MP und Junior vorgestellt. Die Abmessungen sind sehr "bescheiden": Die gesamte Elektronik hat auf einer Euro-Karte Platz. Trotzdem kann man mit diesem Gerät sowohl 2716- als auch 2732-EPROMs programmieren. Die EPROMs können nicht nur programmiert, sondern deren Inhalt kann gleichzeitig auch ausgelesen werden.

Vielleicht tun wir hier etwas zu viel des Guten. Wieder ein EPROMer? Nun – der Unterschied zwischen dem 2716-Programmiergerät und der folgenden Schaltung ist beträchtlich. Die erstgenannte Schaltung war insbesondere auch für "Elektor-fremde" Prozessoren geeignet, während der neue "Programmierer" speziell für SC/MP- und 6502-Systeme entwickelt wurde. Mit diesem Gerät kann man sowohl 2716- als auch 2732-EPROMs programmieren und auslesen. Die Euro-Karte, auf der die Schaltung aufgebaut ist, enthält eine Messerleiste nach DIN, die genau auf den SC/MP-Bus und auf den Junior-Computer paßt.

Wie man solch ein 2716- (und auch 2732-)EPROM programmieren kann, wurde bereits beschrieben. Erwähnt wurde auch, daß gerade der 2716-Typ sehr preiswert zu kaufen und darüber hinaus leicht zu programmieren ist. Wir möchten deshalb nur kurz wiederholen, wie die Programmierung funktioniert. An Pin 21 des 2716 (Pin 20 des 2732) legt man eine Programmierspannung von 25 V. Ein Programmier-Impuls von 50 ms Länge am \overline{CE} -Eingang reicht dann aus, um die an den Eingängen liegenden Daten in die entsprechende Adresse zu laden.

Steht bereits ein Computer zur Verfügung (Wofür sonst möchte man schließlich EPROMs programmieren?), dann liegt es natürlich nahe, den Computer auch zum Programmieren von EPROMs einzusetzen. Es gibt eigentlich nur einen Aspekt, der dagegen spricht: Den Junior-Computer kann man nicht in einem Schreibzyklus

stoppen. Eine Adresse mit den darin enthaltenen Daten muß aber mindestens 50 ms am EPROM anliegen. Beim SC/MP ist das kein Problem! Für den Junior muß man deshalb einige Puffer vorsehen, in denen die Information zwischengespeichert werden kann. Die Platine enthält aus diesem Grunde die notwendigen Puffer. Bei Betrieb mit dem SC/MP-Prozessor kann man sie einfach mittels einiger Drahtstückchen überbrücken.

Die Schaltung

Bevor wir uns an die Programmierung begeben, ist es zunächst notwendig, die Funktionsweise des Gerätes zu beschreiben. Die Schaltung ist in Bild 1 dargestellt. Wie schon erwähnt, sind einige Puffer nötig, in denen die Daten und Adressen einige Zeit lang gespeichert werden können. Die Puffer sind in den IC1 . . . IC4 enthalten. Abhängig vom zu programmierenden EPROM-Typ müssen 11 (für 2716) und 12 (für 2732) Adreßbits in den Puffern zwischengespeichert werden. Deshalb wird die Leitung für das 12. Adreßbit über Schalter S1 wahlweise mit dem EPROM verbunden.

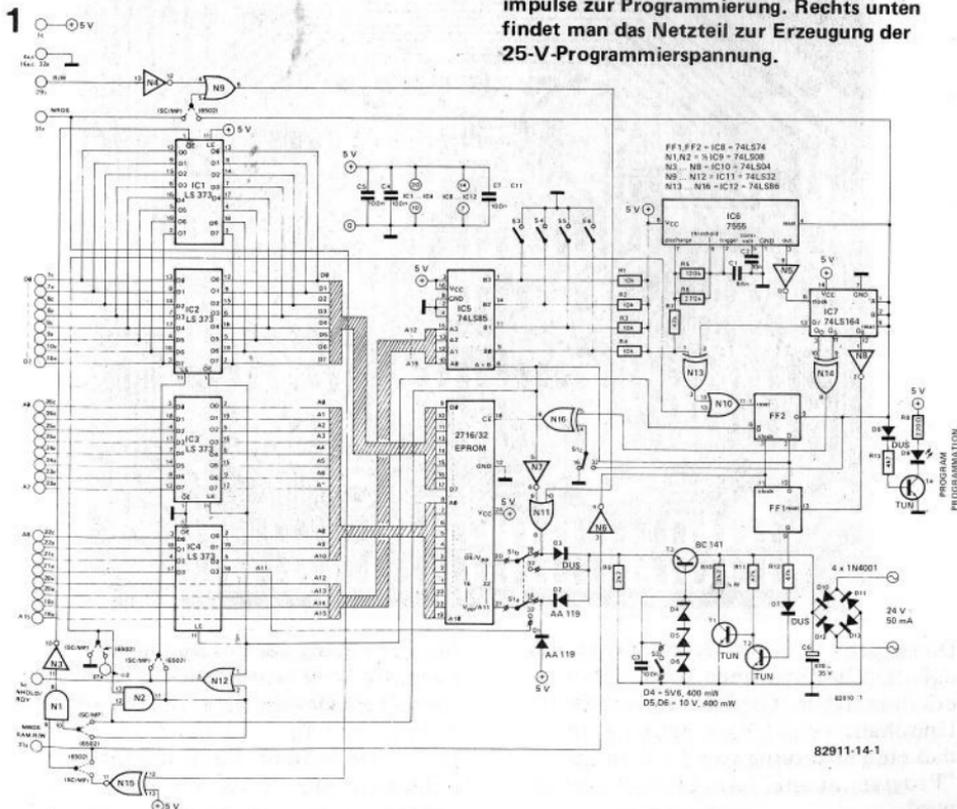
Die Daten-Leitungen D0 . . . D7 sind an zwei parallel geschaltete Zwischenspeicher ("Latch") angeschlossen (IC1 und IC2). Die Eingänge von IC1 sind mit den Ausgängen von IC2, und die Ausgänge von IC1 sind mit den Eingängen von IC2 verbunden. Auf diese Weise ist es möglich, die EPROM-Daten auch wieder über den Computer auszulesen. Außerdem sind noch einige Verbindungen zum Computer notwendig, die alle zum linken Rand von

Bild 1 geführt sind.

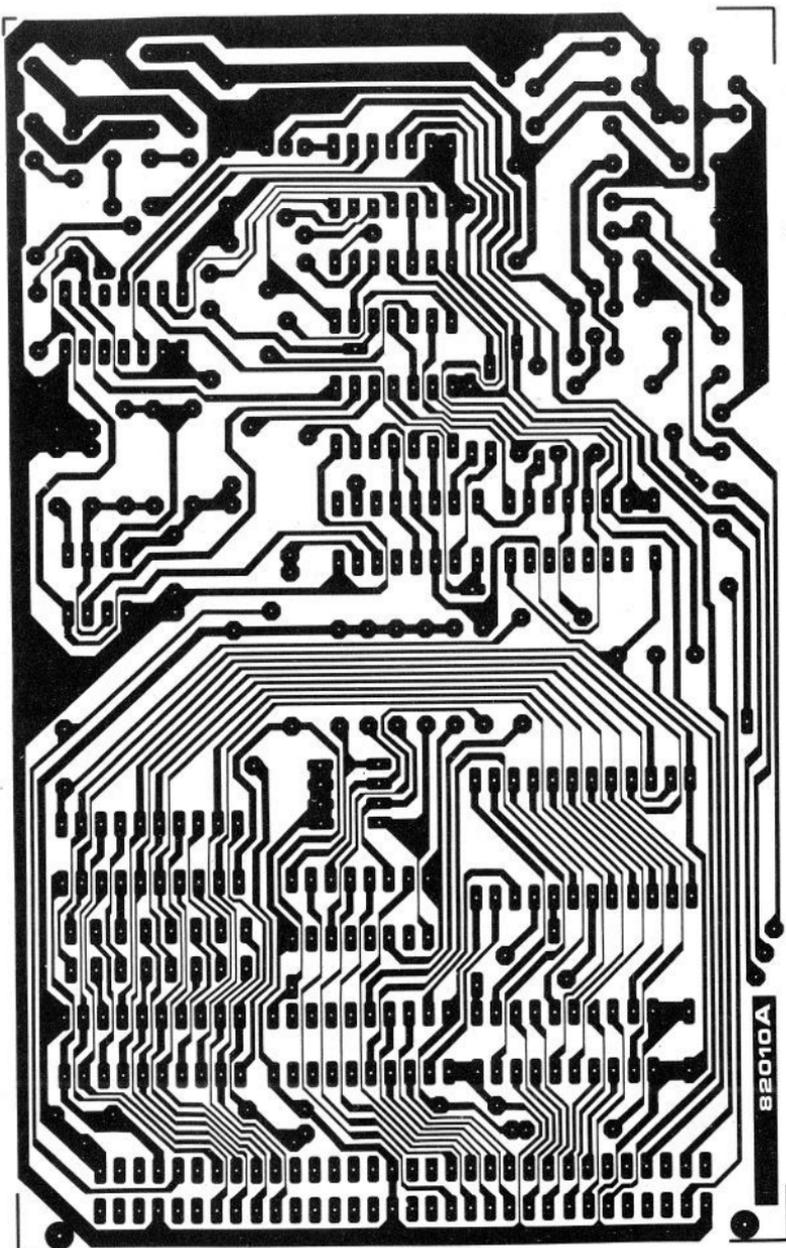
Die erforderliche Adressendekodierung übernimmt der 4-bit-Vergleicher IC5. Auf der einen Seite ist dieses IC mit den Adressenleitungen A12 . . . A15 verbunden. Auf der anderen Seite kann man den Adressbereich mit Hilfe der Schalter S3 . . . S6 einstellen. Ein geschlossener Schalter bildet eine logische Null. Die Schalter sind den Adressenleitungen folgendermaßen zugeordnet: S3 → A12, S4 → A13, S5 → A14 und S6 → A15. Ein Adressbereich umfaßt immer 4 K. Das scheint auf den ersten Blick sehr viel zu sein. Man muß aber bedenken, daß auch ein 4-K-EPROM (2732) programmiert werden kann. Bei der Programmierung eines 2716-EPROMs kann aus diesem Grunde sowohl der Bereich ab X000 als auch der ab X800 adressiert werden. IC6 ist ein astabiler Multivibrator mit Start/Stop-Möglichkeit. Die Dimensio-

nierung der frequenzbestimmenden Bauteile ist so ausgelegt, daß die Periodendauer 10 ms beträgt. Über einen Inverter ist der Ausgang des Multivibrators mit dem Schieberegister IC7 verbunden. Der Reset-Eingang von IC6 liegt am Q-Ausgang von FF2. Der Takt-Eingang dieses Flipflops, und auch der von FF1, erhält über N6 Impulse von Pin 36a der Messerleiste. Das bedeutet: Die Flipflops erhalten dann Taktimpulse, wenn der Prozessor einen Write-Befehl gibt. In dem Moment, wo der Adressbereich gefunden ist, wird der Ausgang "A = B" von IC5 logisch 1. Dieser Impuls wird an den Q-Ausgang von FF2 durchgeschaltet, und damit startet der Multivibrator IC6.

Bild 1. Die Schaltung des EPROMs. In der Mitte des Bildes erkennt man das EPROM, darüber den Adreßdekor. Links davon sind die Puffer für Daten und Adressen dargestellt. Ganz rechts befindet sich die Logik zur Erzeugung der unterschiedlichen Steuerimpulse zur Programmierung. Rechts unten findet man das Netzteil zur Erzeugung der 25-V-Programmierspannung.



82911-14-1



Dies geschieht unter der Voraussetzung, daß die Flipflops einen Write-Befehl erhalten haben. Gleichzeitig mit dem Umschalten von FF2 kippt FF1, so daß eine Spannung von 25 V an den "Programmierpin" des EPROMs gelegt wird.

Zur Erzeugung der Programmiervspannung von 25 V wurde ein kleines Netzteil entwickelt, das im wesentlichen aus den Transistoren T1 . . . T3 und den Dioden D4 . . . D6 besteht. Ein Brückengleichrichter mit Siebelko ist ebenfalls vorhanden, so daß man nur noch eine Wechsel-

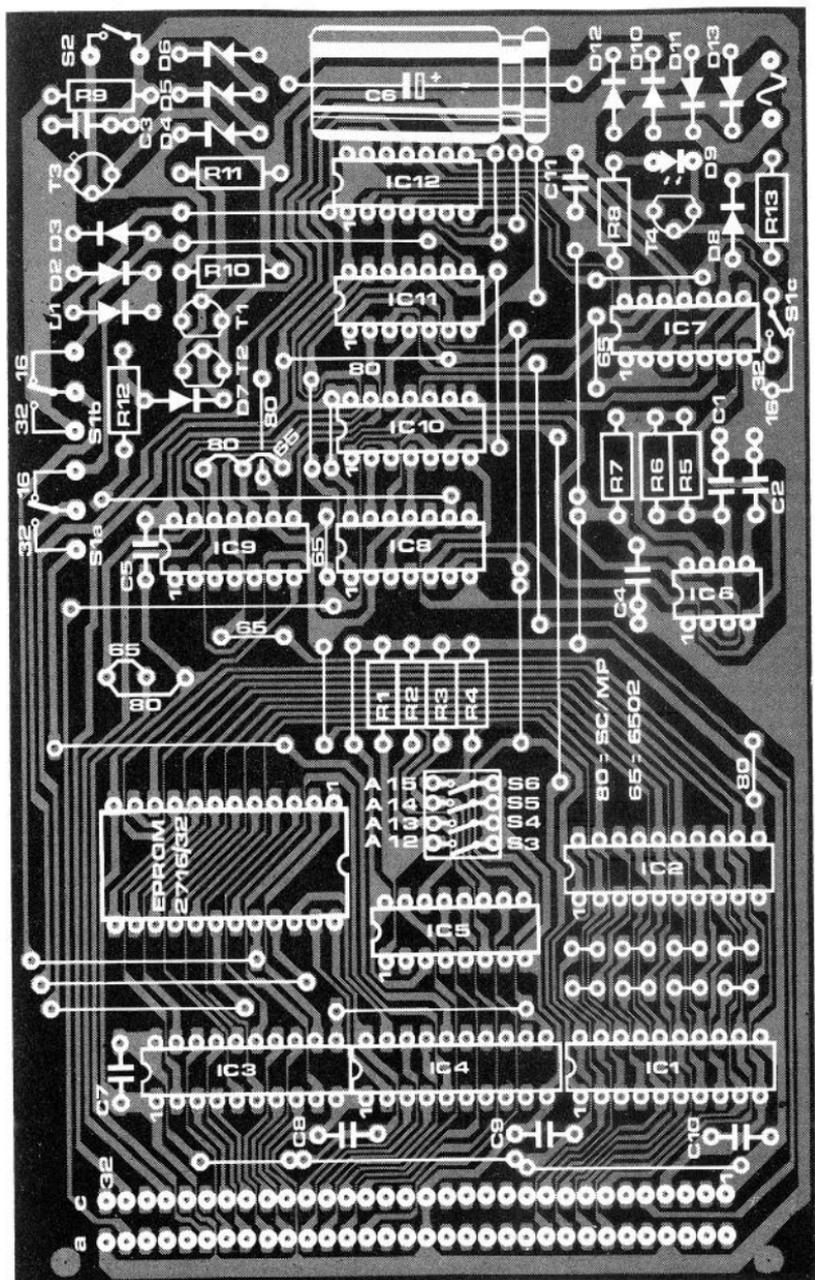


Bild 2. Layout und Bestückungsplan der Platine für den EPROMer. Die Fassung für das EPROM sollte vom Null-Kraft-Typ sein.

spannung von 24 V anschließen muß. Der Q-Ausgang von FF1 schaltet die 25-V-Spannung ein und aus. Mit Schalter S2

kann man zwischen "Programmieren/Lesen" und "Lesen" umschalten. Ist der Schalter geschlossen, dann wird auch die Programmierspannung zu Null (Betriebsart "Lesen"). In der anderen Schalterstellung ist die Betriebsart "Programmieren/Lesen" gewählt.

Stückliste

Widerstände:

R1 . . . R4 = 10 k
R5 = 120 k
R6 = 270 k
R7, R11, R12 = 47 k
R8 = 220 Ω
R9 = 2k2
R10 = 2k2/½ W
R13 = 4k7

Halbleiter:

D1, D2 = AA 119
D3, D7, D8 = DUS
D4 = Z-Diode 5V6/0,4 W
D5, D6 = Z-Diode 10 V/0,4 W
D9 = LED
D10 . . . D13 = 1N4001
T1, T2, T4 = TUN
T3 = BC 141
IC1 . . . IC4 = 74LS373
IC5 = 74LS85
IC6 = 7555
IC7 = 74LS164
IC8 = 74LS74
IC9 = 74LS08
IC10 = 74LS04
IC11 = 74LS32
IC12 = 74LS86

Kondensatoren:

C1 = 68 n
C2 = 10 n
C3 . . . C5 = 100 n
C6 = 470 µ/35 V
C7 . . . C11 = 100 n

außerdem:

S1 = Umschalter, doppelpolig
S2 = Ein/Aus-Schalter
S3 . . . S6 = 4 Miniatur-Schalter
im DIL-Gehäuse
IC-Test-Fassung 24polig
("Null Kraft")
64polige Messerleiste mit abgebo-
genen Anschlüssen nach DIN 41612

Doch nun wieder zurück zur Funktion des EPROMers: Die Programmierspannung ist eingeschaltet, und der Multi-vibrator IC6 "läuft". Im Rhythmus der Impulse von IC6 wird IC7 nun mit Einsen vollgeschrieben. 10 ms nachdem die 25-V-Spannung eingeschaltet wurde, wird Ausgang Q0 logisch 1. Dadurch erhält auch der Eingang \overline{CE} des EPROMs eine "1". Durch die Verknüpfung der Aus-

gänge Q0 und Q5 mittels N14 liegt der "1"-Pegel 50 ms lang am \overline{CE} -Eingang an. 10 ms später wird Ausgang Q6 "1", und FF1 schaltet die Programmierspannung wieder ab. Während der gesamten Zeit sind die Daten und Adressen in den Latches gespeichert. Am Ausgang von FF2 ist über Transistor T4 eine LED angeschlossen, die während der Programmierung des EPROMs aufleuchtet. Beim Lesen der Daten funktioniert die Hardware der Schaltung nicht so ohne weiteres, da der Prozessor keinen Write-Befehl gibt. In diesem Fall werden die Ausgänge von IC2 in den hochohmigen Zustand (einer von "Tri States") geschaltet, und IC1 wird freigegeben. Das läuft ungefähr darauf hinaus, daß die vom EPROM gelieferten Daten praktisch direkt mit den Datenleitungen des Prozessors verbunden sind. Schalter S1 dient zur Einstellung auf den EPROM-Typ. In der gezeichneten Stellung von S1 (2716) liegt die Programmierspannung an Pin 21, gleichzeitig ist der \overline{OE} -Eingang mit N11 verbunden. Wenn man den Schalter umstellt (2732), ist die Adressenleitung A11 mit Pin 21 verbunden. Die Programmierspannung liegt an Pin 20. Mit Hilfe einiger Gatter (N1 . . . N4, N9, N11, N12 und N15) werden, abhängig vom gewählten Prozessor, aus den vorhandenen Steuersignalen weitere Steuerungssignale für die Schaltung abgeleitet.

Die Euro-Karte

In Bild 2 ist die Platine für den EPROMer dargestellt. Erfahrene Computer-Bauer werden sicherlich keine Schwierigkeiten bei der Bestückung haben. Die EPROM-Fassung sollte von sehr guter Qualität sein, da sie häufig benutzt wird. Ein paar Mark mehr als gewöhnlich sind in diesem Fall kein rausgeworfenes Geld! Am besten eignet sich eine sogenannte Null-Kraft-Fassung. Man setzt das IC in den Sockel und stellt den Kontakt mit einem Hebel-druck her.

Die Karte ist wie erwähnt für zwei unterschiedliche Systeme ausgelegt: SCMP und Junior. Abhängig davon müssen die entsprechenden Drahtbrücken auf der Platine gelegt werden. Die Verbindungen

zum Bus sind alle auf eine DIN-Leiste geführt. Man braucht nur noch eine Wechselspannung von 24 V anzuschließen.

SC/MP + EPROMer

Besitzer des SC/MP-Systems können den EPROMer ganz einfach anwenden. Nach Einstellung der gewünschten Adresse auf der Karte wird zunächst S2 geschlossen. Dann setzt man das EPROM in die Fassung, steckt anschließend die Karte in die Bus-Leiste und beginnt mit der Programmierung. Voraussetzung dafür ist, daß die Programmiervoltage anliegt und daß S2 wieder geöffnet wird. Nach der Programmierung sollte man S2 wieder schließen.

Eine spezielle Software ist für den EPROMer in diesem Fall nicht notwendig; ELBUG reicht hier völlig aus. Möchte man einen Speicherplatz programmieren, dann genügt es, MO . . . YYYY einzugeben. YYYY ist die zu programmierende Adresse. Bei einem leeren EPROM erscheint "FF" als Antwort auf dem Display. Soll auf diesen Platz beispielsweise 08 gesetzt werden, dann muß man 08 eingeben. Und die Daten sind programmiert.

Sollen größere Daten-Blöcke einprogrammiert werden, dann empfiehlt sich eine Speicherung im RAM des Computers. Anschließend werden die Informationen mit einem Block-Transfer-Befehl in das EPROM geladen (BL . . . SSSS, EEEE, BBBB - S = Start-Adresse, E = End-Adresse, B = Adresse, in die der Block geladen wird).

Das Lesen des EPROM-Inhalts geschieht auf die gleiche Weise, als ob man eine normale Adresse lesen möchte.

Junior + EPROMer

Um den EPROMer mit dem Junior-Computer betreiben zu können, braucht man ein spezielles Programm. Dieses Programm ist in Tabelle 1 aufgelistet. Es beginnt bei 0200 und endet bei 0277. Hat man das Programm geladen, dann kann der EPROMer auf die Erweiterungs-Leiste gesteckt werden. Der EPROMer ist jetzt betriebsbereit.

Man kann das Programm aus Tabelle 1

auch in ein EPROM laden, so daß es nicht jedes Mal, wenn der EPROMer in Betrieb genommen werden soll, eingegeben werden muß. Achtung: Die absoluten Sprungadressen im Programm müssen in diesem Fall angepaßt werden! Eine Möglichkeit besteht darin, das TM-EPROM auf der Interface-Karte mit dem Programm zu laden. Dieses EPROM hat nämlich noch einige freie Plätze. In den Adressen 0C80 . . . 0FFF steht überall FF. Im EPROM selbst sind das die Plätze 480 . . . 7FF. Läßt man das Programm statt bei 0200, wie in Tabelle 1, bei 0C80 beginnen, kann das Programm in das TM-EPROM geladen werden. In diesem Fall muß man allerdings die absoluten Sprungadressen im Programm anpassen (alle 3-Byte-Befehle, die mit 02 enden). Im folgenden geben wir eine kurze Beschreibung der drei Programmteile PROGRAM, DUPLICATE und VERIFY.

Die Programmier-Routine

Auf die Speicherplätze MOVH, MOVH (0004, 0005) werden das linke und das rechte Byte der Adresse, mit der man die Programmierung beginnen möchte, gesetzt. Danach wird die Programmier-Routine gestartet (in Tabelle 1 → 0200). Auf dem Display erscheinen nun die Adresse und die Daten, die in dieser Adresse stehen. FF erscheint, wenn das EPROM leer ist. Will man beispielsweise A9 programmieren, dann wird erst A gedrückt (die Anzeige im Display ändert sich nicht) und anschließend 9. Das Display erlischt für etwa 70 ms. Danach erscheinen Adresse und Daten A9 wieder auf dem Display. A9 ist in den entsprechenden Speicherplatz geladen.

Drückt man danach die Plus-Taste, dann erscheint die nächste Adresse auf dem Display. Auf die zuvor beschriebene Art können Daten geladen werden. Möchte man nur Daten lesen, dann braucht nur die Plus-Taste gedrückt werden, oder man kehrt durch Betätigung der Reset-Taste ins Monitor-Programm zurück.

Die Duplicate-Routine

Möchte man ein Programm, das irgendwo

Tabelle 1

```

0010: 0200          ORG 50200
0020:
0030:          DATE : 10-7-'81
0040:
0050:          PAGE ZERO DATA BUFFERS :
0060:
0070:
0080: 0200          SAL * 50000 DATA BLOCK START ADDRESS
0090: 0200          SAH * 50001
0100: 0200          EAL * 50002 DATA BLOCK END ADDRESS + 1
0110: 0200          EAH * 50003
0120: 0200          MOVL * 50004 EPROM PROGRAM START ADDRESS
0130: 0200          MOVH * 50005
0140:
0150: 0200          INH * 500F9 DISPLAY BUFFER ( DATA )
0160: 0200          POINTL * 500FA " " ( ADDRESS L )
0170: 0200          POINTH * 500FB " " ( ADDRESS H )
0180:
0190:          EXTERNAL SUBROUTINES :
0200:
0210: 0200          GETBYT * $1D6F
0220: 0200          SCAND * $1D88
0230:
0240:          JUNIOR MONITOR START :
0250:
0260: 0200          RESET * $1C1D
0270:
0280:
0290:          PROGRAM START ADDRESS : 50200 ( PROG )
0300:          DUPLICATE START ADDRESS : 50222 ( DUPL )
0310:          VERIFY START ADDRESS : 50233 (-VERIFY )
0320:
0330:
0340:          *****
0350:          EPROM-PROGRAMMER
0360:          *****
0370:
0380: 0200 20 55 02 PROG JSR TRF TRANSFER MOVL(H) TO POINTL(H)
0390: 0203 A0 00 PRGR LDYIM $80 CLEAR Y-REGISTER
0400: 0205 91 FA LDAYI POINTL GET DATA SPECIFIED BY POINTL(H) AND
0410: 0207 85 F9 STAZ INH STORE THIS IN DISPLAY BUFFER INH
0420: 0209 20 6F 1D JSR GETBYT READ TWO HEXKEYS AND STORE THEIR VALUE IN THE
0430: ACCUMULATOR, RETURN WITH H#1 IF
0440: ONLY HEXKEYS WERE DEPRESSED.
0450: IF A COMMAND KEY WAS DEPRESSED,
0460: RETURN WITH H#0
0470: 020C 10 07 BPL PR COMMAND KEY DEPRESSED?
0480: 020E A0 00 LDYIM $00 CLEAR Y-REGISTER
0490: 0210 91 FA STAYI POINTL PROGRAM THE CONTENTS OF THE ACCUMULATOR IN
0500: THE EPROM MEMORY LOCATION
0510: SPECIFIED BY POINTL(H)
0520: 0212 4C 03 02 JMP PRGR
0530: 0215 C9 12 PR CMPIM $12
0540: 0217 D0 15 BNE PRGR +KEY?
0550: 0219 E6 FA INCZ POINTL INCREMENT ADDRESS BY ONE
0560: 021B D0 E6 BNE PRGR
0570: 021D E6 FB INCZ POINTH
0580: 021F 4C 03 02 JMP PRGR
0590: 0222 20 55 02 DUPL JSR TRF TRANSFER MOVL(H) TO POINTL(H)
0600: 0225 A0 00 DU LDYIM $80 GET DATA SPECIFIED BY SAL(H)
0610: 0227 91 00 LDAYI SAL PROGRAM THE CONTENTS OF THE ACCUMULATOR IN
0620: 0229 91 FA STAYI POINTL THE EPROM MEMORY LOCATION
0630: SPECIFIED BY POINTL(H)
0640: 022B 20 5E 02 JSR INCMNT INCREMENT SAL(H) AND POINTL(H) BY ONE
0650: 022E D0 F5 BNE DU NOT LAST ADDRESS
0660: 0230 4C 1D 1C JMP RESET RETURN TO JUNIOR MONITOR
0670:
0680: 0233 20 55 02 VERIFY JSR TRF TRANSFER MOVL(H) TO POINTL(H)
0690: 0236 A0 00 VER LDYIM $00 GET DATA SPECIFIED BY POINTL(H)
0700: 0238 B1 FA LDAYI POINTL COMPARE THIS DATA WITH DATA SPECIFIED BY SAL(H)
0710: 023A D1 0F CMPYI SAL DATA EQUAL?
0720: 023C F0 0F BEQ NEXT
0730: 023E 28 88 1D ANYKEY JSR SCAND DISPLAY EPROM ADDRESS AND DATA
0740: 0241 D0 FB BNE ANYKEY ANY KEY DEPRESSED?
0750: 0243 28 88 1D JSR SCAND DISPLAY EPROM ADDRESS AND DATA
0760: 0246 D0 FE BNE ANYKEY ANY KEY DEPRESSED?
0770: 0248 20 88 1D NKEY JSR SCAND DISPLAY EPROM ADDRESS AND DATA
0780: 024B F0 FB BEQ NKEY NO KEY DEPRESSED?
0790: 024D 20 5E 02 NEXT JSR INCMNT INCREMENT SAL(H) AND POINTL(H) BY ONE
0800: 024E D0 E4 BNE VER NOT LAST ADDRESS?
0810: 0250 4C 1D 1C JMP RESET RETURN TO JUNIOR MONITOR
0820:
0830:          *****
0840:          SUBROUTINES
0850:          *****
0860:
0870: 0255 A5 04 TRF LDAH MOVL
0880: 0257 85 FA STAZ POINTL TRANSFER MOVL TO POINTL
0890: 0259 A5 05 LDAH MOVH
0900: 025B 85 FB STAZ POINTH TRANSFER MOVH TO POINTH
0910: 025D 60 RTS
0920:
0930: 025E 20 88 1D INCMNT JSR SCAND DISPLAY FOR ABOUT SMS POINTH, POINTL
0940: AND INH ( = EPROM ADDRESS AND DATA
0950: ON THIS ADDRESS )
0960: 0261 E6 00 INCZ SAL INCREMENT SAL(H) BY ONE
0970: 0263 D0 02 BNE INCDA
0980: 0265 E6 01 INCZ SAH
0990: 0267 E6 FA INCDA INCZ POINTL INCREMENT POINTL(H) BY ONE
1000: 0269 D0 02 BNE COMP
1010: 026B E6 FB INCZ POINTH
1020: 026D A5 01 COMP LDAH SAL
1030: 026F C5 03 CMPZ EAH COMPARE EAH WITH SAH
1040: 0271 D0 04 BNE RTRN EAH NOT EQUAL SAH?
1050: 0273 A5 00 LDAH SAL
1060: 0275 C5 02 CMPZ EAL COMPARE EAL WITH SAL
1070: 0277 60 RTRN RTS

```

Tabelle 1. Dieses Programm erlaubt den Betrieb des EPROMs zusammen mit dem Junior-Computer.

im Speicherbereich des Computers steht, in ein EPROM laden, dann wird die Duplicate-Routine verwendet. Auf die Speicherplätze SAL, SAH (0000).

0001) wird die Start-Adresse des zu kopierenden Programms gesetzt. In EAL, EAH (0002, 0003) legt man die End-Adresse + 1. Und auf MOVL, MOVH (0004, 0005) wird die EPROM-Adresse gesetzt, in der das erste Byte des zu kopierenden Programm-Blocks stehen soll.

Danach kann man das Duplikat-Programm starten (Adresse 0222), und das Display erlischt. Nach einigen Programmschritten "meldet" sich das Display aber mit der Anzeige des programmierten Speicherplatzes zurück. Auf diese Weise kann man den Programmiervorgang auf dem Display verfolgen. Allerdings ist die Anzeige immer nur sehr kurz und schwach!

Ist der Programmiervorgang beendet, dann meldet sich der Junior mit der Angabe der letzten Adresse + 1 zurück.

Die Verify-Routine

Mittels dieser Routine kann ein bestimm-

ter Programm-Block mit dem EPROM-Inhalt verglichen werden. Auch in diesem Fall werden Start-Adresse, End-Adresse + 1 und Ziel-Adresse genau wie zuvor beschrieben eingegeben. Danach startet man das Programm auf 0233.

Wird nun ein Fehler entdeckt, stoppt das Programm, und auf dem Display wird die Adresse angezeigt, in der der Fehler steckt. Auch die Daten, die in diesem Moment an der Stelle stehen, werden angezeigt. Drückt man die RES- oder ST-Taste, dann läuft das Programm weiter.

Ist alles kontrolliert, meldet sich der Junior wieder mit der letzten Adresse + 1 zurück. Gleichzeitig ist also auch die Rückkehr ins Monitor-Programm erfolgt.

Schließlich bleibt nur noch zu sagen, daß mit Hilfe des EPROMers ein nützliches Gerät zur Verfügung steht, mit dem man EPROMs sehr leicht selbst programmieren kann. In der Praxis wird sich die Nützlichkeit sehr schnell erweisen.

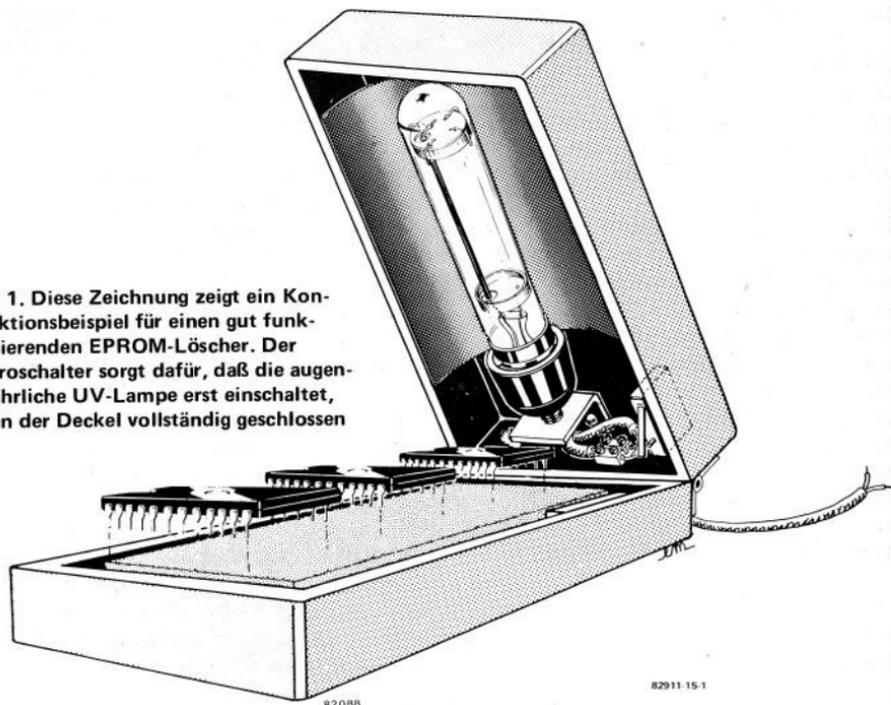
EPROM-Löcher

Ohne Festwertspeicher, sprich ROM, PROM oder EPROM, läuft kein Computer. Auch nicht der kleinste Hobbycomputer. Für Letztgenannten verwendet man meist EPROMs, die sich bei Bedarf löschen und mit neuem Inhalt versehen lassen. Nach dem beschriebenen Programmiergerät ist es logisch, wenn auch von dem ebenso wichtigen Löschen die Rede ist. Die beschriebene Löschmethode arbeitet zuverlässig und ist außerdem preiswert anzuwenden. Vorab aber eine wichtige Warnung: Das für das EPROM-Radieren benötigte UV-Licht ist sehr augenschädigend. Also aufgepaßt!

Über PROMs und EPROMs wurde schon viel geschrieben, jedenfalls, was das Programmieren anbelangt. Im Gegensatz zu den nur einmal programmierbaren PROMs lassen sich EPROMs durch Belichten mit speziellem UV-Licht wieder löschen und danach neu program-

mieren. Die in Elektor am häufigsten verwendeten EPROMs sind die Typen 2708 und 2716. Normalerweise löscht man diese Bausteine mit Hilfe einer Spezial-UV-Lampe. Es geht aber auch anders. So beginnt beispielsweise der Löschvorgang bei dem

Bild 1. Diese Zeichnung zeigt ein Konstruktionsbeispiel für einen gut funktionierenden EPROM-Löschgerät. Der Mikroschalter sorgt dafür, daß die augengefährliche UV-Lampe erst einschaltet, wenn der Deckel vollständig geschlossen ist.



82088

82911 15-1

bekanntem EPROM 2716, sobald es einer Lichtquelle ausgesetzt wird, die Strahlung mit einer Wellenlänge kleiner als 400 nm aussendet. Auch die Sonne und Leuchtstofflampen liefern Anteile in diesem Bereich.

Allerdings hat es sich gezeigt, daß ein 2716 bei fortwährender Beleuchtung mit normalen TL-Leuchtstofflampen erst nach etwa 3 Jahren gelöscht ist. Bei strahlendem Sonnenschein hingegen geht es etwas schneller. Schon nach einer Woche direkter Sonnenbestrahlung ist die Information in der Regel verloren. Daher sollte man das Fenster eines EPROMs auch nach dem Programmieren mit einem Aufkleber abdecken, um ein allmähliches unfreiwilliges Löschen durch Sonnen- und Lampenlicht zu vermeiden.

Für eine erwünschte Löschung sind die vorgenannten Lichtquellen weniger geeignet, es sei denn, man nimmt seine EPROMs in den Urlaub an Spaniens Sonnenküste mit. Eine praktikablere Lichtquelle müßte UV-Licht mit einer Wellenlänge von 253,7 nm und einer

Intensität von 12 mW/cm^2 liefern, um EPROMs in 15 bis 20 Minuten zu löschen.

Solche Lichtquellen befinden sich in den käuflich erhältlichen EPROM-Löschgeräten, die meist auch noch eine kleine Schaltung enthalten. Für den Amateur sind diese Geräte aber unrentabel, da zu teuer. Eine preiswertere Gelegenheit haben wir in der UV-Lampe TUV 6 W der Firma Philips gefunden, die UV-Strahlung mit der genau passenden Wellenlänge ausstrahlt. Diese Lampe wird normalerweise für Sterilisationsgeräte (Keimtöter) verwendet und kann in jedem größeren Elektro-Fachgeschäft gekauft oder zumindest bestellt werden. Das Lämpchen ist von länglicher Bauform mit normalem Edison-Schraubsockel und läßt sich sehr vorteilhaft ohne Vorschaltgerät betreiben. Bevor Sie aber jetzt zum nächsten Laden eilen, um Ihre Bestellung aufzugeben, noch eine ernste Warnung: *Niemals Lampe einschalten, ohne Augen zu schützen oder abzuschirmen. Das ausgesandte UV-Licht kann Ihr Sehvermögen bleibend beein-*

trächtigen. Bestrahlung der Haut kann Rötung und Juckreiz (Sonnenbrand) verursachen.

Um Blindheit und Verbrennungen zu vermeiden, ist es unbedingt erforderlich, die Lampe in ein lichtdichtes Kästchen einzubauen. Wegen der Wärmeentwicklung der Lampe sollte das Kästchen nicht zu klein sein. Bild 1 zeigt ein Konstruktionsbeispiel. Am Deckel wird die Lampenfassung befestigt, wobei über der Lampe noch ein Reflektor, z.B. aus Weißblech, anbracht werden sollte. Auf Bodenplatte des Kästchens ist dann Platz für die zu löschenden EPROMs. Die Sicherheit ist am besten dadurch zu gewährleisten,

daß man die Lampe durch einen Mikroschalter einschaltet, der erst bei Schließen des Deckels betätigt wird. Bevor man die EPROMs in den Löscher legt, sollte man sie auf ein Stück Leitschaumgummi stecken.

Unter die Lampe ist Platz genug, um vier EPROMs in der gleichen Zeit zu löschen. Die Löschzeit beträgt etwa 30 Minuten, wenn der Abstand zwischen Lampe und EPROMs nicht größer als etwa 10 mm ist. Diese Zeit gilt für die meistverwendetsten EPROMs. Eine Ausnahme bildet das TMS 2516 der Firma Texas Instruments. Es muß gut zwei Stunden belichtet werden.

PROM-Programmierer

Ein PROM ist ein Programmable Read Only Memory, ein programmierbarer Festwertspeicher. Diese Bausteine haben bereits seit Jahren auch in Selbstbauprojekten Eingang gefunden. Um ein PROM mit den Informationen zu füllen, die später nur noch ausgelesen werden können, dafür aber auch bei Wegfall der Speisespannung erhalten bleiben, wird ein PROM-Programmiergerät benötigt. Ein solches Gerät ist recht kostspielig, da das Programmierverfahren stets strengen Vorschriften unterworfen ist. Deshalb überlassen Hobby-Elektroniker diese Prozedur meistens dem Hersteller oder Vertreter der PROMs.

PROMs mit relativ geringer Speicherkapazität können aber auch selbst programmiert werden. Im einfachsten Fall geschieht dies bitweise von Hand; der dafür notwendige Zeitaufwand fällt bei Einzelstücken weniger ins Gewicht. Selbstverständlich müssen auch hier die vorgeschriebenen Strom-, Spannungs- und Zeitwerte eingehalten werden.

Dieses Programmiergerät ist für das PROM 82S23 von Signetics, ein verbreiter 32 x 8-bit-Typ, bestimmt.

Auch das PROM 82S123 kann programmiert werden; es hat im Gegensatz zum 82S23 keine Open-Collector- sondern

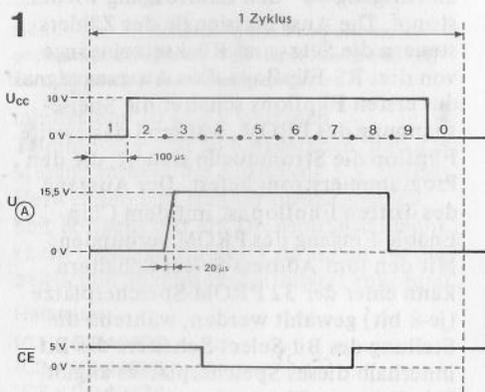


Bild 1. Spannungsverhältnisse beim Programmieren eines Bits.

Tri-State-Ausgänge. Die vom Hersteller vorgeschriebenen Werte werden vom Programmiergerät voll erfüllt. In Bild 1 sind die wichtigsten Signale gezeichnet. Spannung U_{CC} ist die Speisespannung des PROM; \overline{CE} ist das Signal, das an den Chip-Enable-Eingang gelegt werden muß. Das eigentliche Programmiersignal ist die Spannung $U(A)$. Ihre Anstiegszeit muß nach Vorschrift des Herstellers zwischen 10 μs und 50 μs liegen; sie beträgt bei diesem Programmierer ca. 20 μs . Das Programmieren jedes Bit dauert eine

bestimmte Zeit. Um die Reihenfolge der verschiedenen Signaländerungen zu verdeutlichen, wurde in Bild 1 die Zeitachse in Abschnitte von $100 \mu\text{s}$ unterteilt.

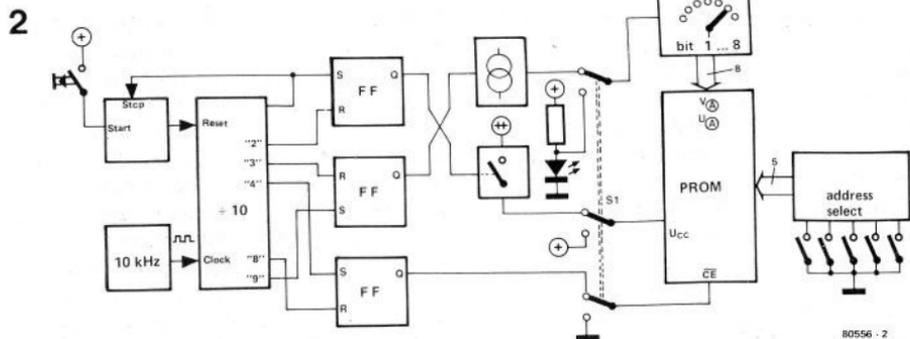
Bild 2 zeigt das Blockschema des Selbstbau-Programmiergerätes. Als Zeitgeber dient ein Taktgenerator mit einer Frequenz von 10 kHz . Diese Frequenz ist gleichbedeutend mit einer Periodendauer des Taktsignals von $100 \mu\text{s}$. Das Taktsignal wird einem Dezimalzähler zugeführt.

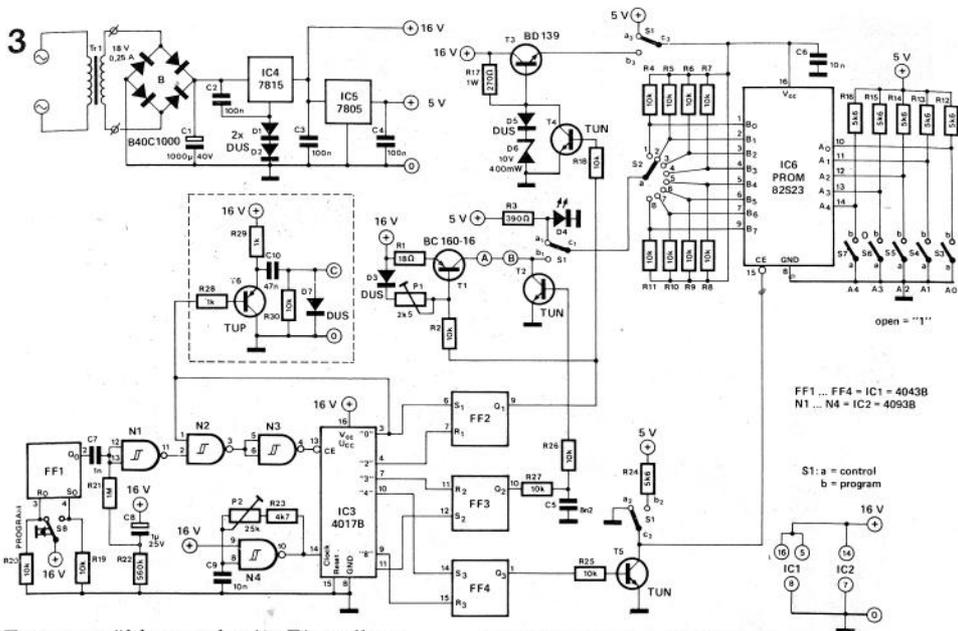
An den zehn Ausgängen des Zählers erscheint der Reihe nach jeweils ein Impuls mit einer Breite von $100 \mu\text{s}$. Der Zähler, der selbsttätig in seiner Nullstellung stoppt, wird von Hand gestartet. Nach dem ersten Taktpuls geht Ausgang "0" auf logisch 0, so daß der Zählengang freigegeben wird. Der Zähler zählt so lange, bis eine logische 1 an Ausgang "0" den Zählvorgang wieder stoppt. Die Ausgangssignale des Zählers steuern die Setz- und Rücksetzeingänge von drei RS-Flipflops. Das Ausgangssignal des ersten Flipflops schaltet die Speisepannung des PROM, während das zweite Flipflop die Stromquelle steuert, die den Programmierstrom liefert. Der Ausgang des dritten Flipflops ist mit dem Chip-Enable-Eingang des PROM verbunden. Mit den fünf Address-Select-Schaltern kann einer der 32 PROM-Speicherplätze (je 8 bit) gewählt werden, während die Stellung des Bit-Select-Schalters das Bit innerhalb dieses Speicherplatzes angibt. Schaltet man S1 um, dann wird der Inhalt der gewählten 1-bit-Zelle von einer LED angezeigt. Die LED leuchtet auf,

wenn der Inhalt logisch 1 ist. Bild 3 zeigt die Schaltung des PROM-Programmierers. Das PROM ist hier mit IC6 bezeichnet, IC3 ist der 10-Zähler, und Gatter N4 arbeitet als Taktgenerator. Nach Einschalten der Speisepannung sorgen C8, R21 und R22 dafür, daß der Zähler automatisch auf Null gesetzt wird. Die Stromquelle, die den Programmierstrom liefert, ist mit Transistor T1 aufgebaut. Sie wird von Flipflop FF2 ein- und ausgeschaltet. T2 schließt den Ausgang der Stromquelle nach Masse kurz, wenn FF3 gesetzt ist. Die Anstiegszeit der so erzeugten Programmierspannung $U_{(A)}$ hängt von C5, R26 und R27 ab; sie beträgt hier ca. $20 \mu\text{s}$. Flipflop FF2 schaltet außer T1 gleichzeitig T4 und legt damit entweder 10 V oder 0 V an den Speisepannungsanschluß des PROM. FF4 steuert über T5 den Chip-Enable-Eingang.

In Bild 4 ist die Platine mit dem zugehörigen Bestückungsplan wiedergegeben. Der Bau des PROM-Programmierers ist unkritisch; empfohlen wird lediglich die Verwendung von IC-Fassungen. Für IC6 (das zu programmierende PROM) ist natürlich stets eine Fassung erforderlich. Die Bauelemente R28 . . . R30, T6, D7 und C10 können entfallen, wenn ein Frequenzzähler vorhanden ist. Der in Bild 3 gestrichelt umrahmte Schaltungsteil dient nämlich ausschließlich zum Einstellen der Taktfrequenz auf den richtigen Wert. Doch auch ohne

Bild 2. Blockschaltbild des PROM-Programmierers.





Frequenzzähler macht die Einstellung kaum Mühe: C8 wird hierzu kurzgeschlossen und die Brücke A-B zwischen der Stromquelle und T2 entfernt. An Ausgang C legt man ein Vielfachmeßinstrument, das auf den Gleichspannungsbereich 10 V eingestellt ist; der Innenwiderstand des Instruments muß

Bild 3. Der im Schaltbild des Programmiers gestrichelt umrahmte Teil dient ausschließlich zum Einstellen der Taktfrequenz.

mindestens 20 kΩ/V betragen. Trimpoti P2 ist so einzustellen, daß das Meßinstrument die Spannung 5 V anzeigt. Verwenden läßt sich für diese Einstellarbeit übrigens nur ein Drehspulinstrument (kein Digitalvoltmeter!), da es hier auf den zeitlichen Spannungsmittelwert ankommt.

Stückliste

Widerstände:

- R1 = 18 Ω
- R2, R4 ... R11, R18, R19, R20, R25, R26, R27, R30 = 10 k
- R3 = 390 Ω
- R12 ... R16, R24 = 5k6
- R17 = 270 Ω/1 W
- R21 = 1 M
- R22 = 560 k
- R23 = 4k7
- R28, R29 = 1 k
- P1 = 2k5-Trimpoti
- P2 = 25-k-Trimpoti

Kondensatoren:

- C1 = 1000 µ/40 V
- C2, C3, C4 = 100 n MKM
- C5 = 8n2
- C6, C9 = 10 n MKM
- C7 = 1 n MKM
- C8 = 1 µ/25 V
- C10 = 47 n MKM

Halbleiter:

- IC1 = CD 4043B
- IC2 = CD 4093B
- IC3 = CD 4017B
- IC4 = 7815
- IC5 = 7805
- IC6 = 82S23 (Valvo/Signetics)
- D1, D2, D3, D5, D7 = DUS
- D4 = LED
- D6 = Zenerdiode 10 V/0,4 W
- T1 = BC 160-16
- T2, T4, T5 = TUN
- T3 = BD 139
- T6 = TUP

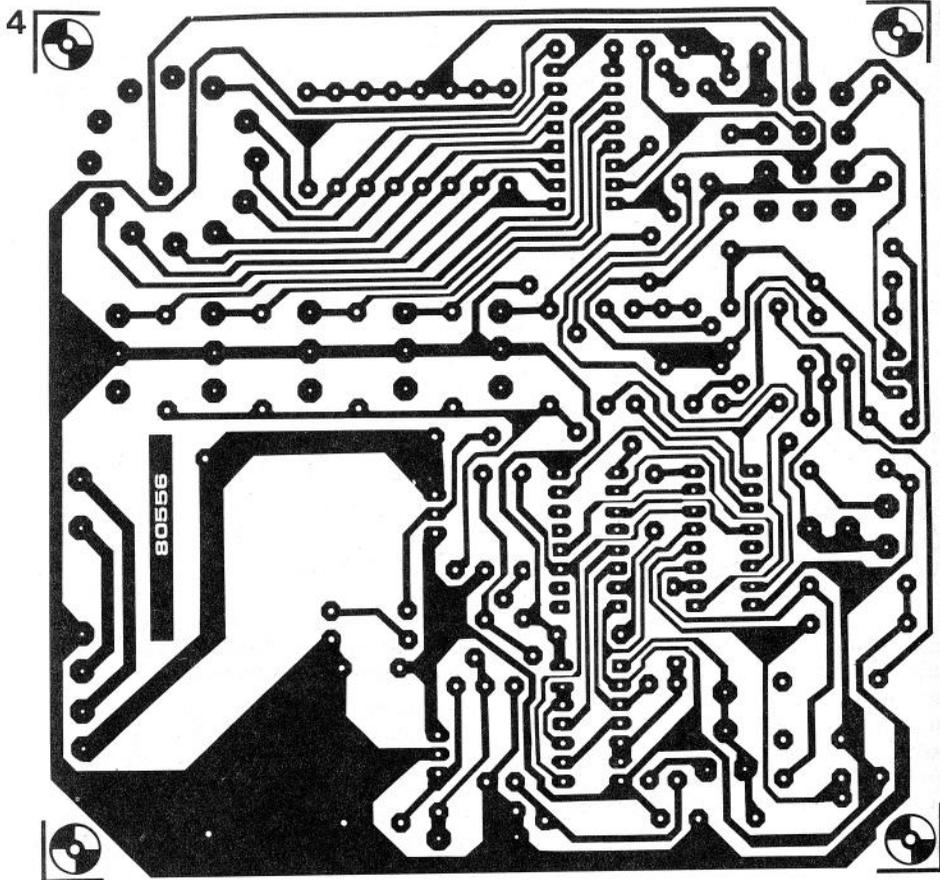
außerdem:

- Brückengleichrichter B40C1000
- dreipoliger Umschalter
- Drehgeber 12 Stellungen
- S3 ... S7 = einpoliger Schalter
- einpoliger Schalter (Digitast)
- Netztrafo 18 V/0,25 sek.

FF1 ... FF4 = IC1 = 4043B
N1 ... N4 = IC2 = 4093B

S1: a = control
b = program

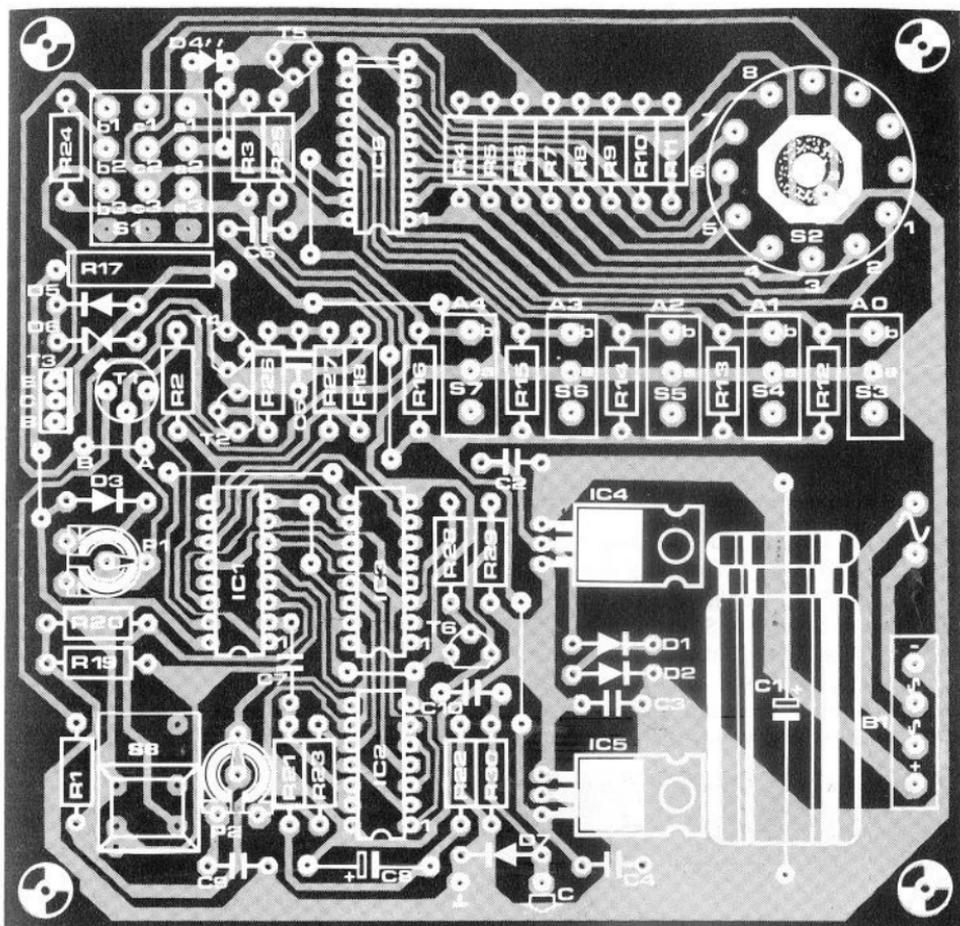
open = "1"



Die Bedienung des Programmiergeräts ist denkbar einfach. Beim Einsetzen des PROM in seine Fassung muß die Speisepannung ausgeschaltet sein; ferner muß S1 in Stellung "Kontrolle" (a) stehen. Mit S3 . . . S7 wird die Adresse des 8-bit-Worts und mit S2 das gewünschte Bit aus diesem Wort gewählt. S1 darf erst in Stellung "Programmieren" (b) gesetzt werden, nachdem man sich vergewissert hat, daß S3 . . . S7 sowie S2 auch tatsächlich in der gewünschten Stellung stehen. Der Programmiervorgang läuft ab, sobald Taster S8 gedrückt wird. In die gewählte Zelle wird dann eine logische 1 gesetzt. Bei unprogrammierten PROMs steht in sämtlichen Zellen logisch 0, so daß Nullen nicht programmiert werden brauchen. Zum Schluß kontrolliert man

Bild 4. Platinenlayout und Bestückungsplan für den PROM-Programmierer.

mit Hilfe der LED (S1 in Stellung a) noch einmal den gesamten Speicherinhalt. Wie schon erwähnt, leuchtet die LED auf, wenn in einer Zelle logisch 1 steht. Wer einen Frequenzzähler besitzt, kann diesen an den Ausgang von N4 legen und die Frequenz 10 kHz unmittelbar einstellen. Das Meßgerät wird anschließend auf Gleichstrommessung umgeschaltet und in Serie mit einem Widerstand $180 \Omega/0,5 \text{ W}$ zwischen den Kollektor von T1 und Masse gelegt. Mit P1 stellt man den angezeigten Strom auf 50 mA ein. Während des Programmierens fließt dann ein Strom von 65 mA, denn auch

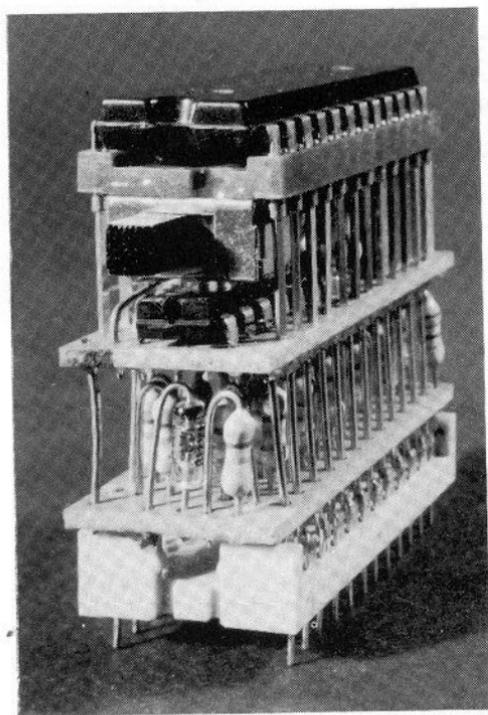


hier wird der zeitliche Mittelwert gemessen. Damit ist das Einstellen des Programmierers beendet. Der Kurzschluß von C8 wird wieder aufgehoben und die Brücke A-B wieder angebracht.

IPROM

Für Computer-Anwender ist es sicherlich sehr hilfreich, ein Programm auch permanent speichern zu können. Sonst geht es beim Ausschalten der Betriebsspannung verloren und muß später erst wieder eingegeben werden. EPROMs und ROMs sind zwar ausgezeichnete Festwertspeicher, benötigen aber ein spezielles Programmiergerät. Das IPROM, alias "Pseudo-ROM", hingegen wird einfach in einen vorhandenen EPROM-Sockel gesteckt, und schon kann man das Programm wie bei einem RAM einfach laden – ohne Programmiergerät. Ein spezielles Löschergerät ist ebenfalls überflüssig. "Alte" Daten werden einfach überschrieben.

Das IPROM verbindet die Vorzüge von RAM und ROM. Es besteht aus einem RAM, das nur sehr wenig Strom verbraucht. Man kann deshalb den Speicherinhalt über eine sehr lange Zeit aufbewahren. Kompakt-Kassette und/oder Floppy-Disk-Speicher können das auch. Ihr Einsatz erfordert allerdings wesentlich mehr



Aufwand. Auch ein EPROM wäre als Speicher geeignet. Ein wesentlicher Nachteil ist jedoch, daß man nur mit Mühe den Inhalt ändern kann.

Deshalb bietet sich das IPROM als Alternative an: Ein RAM mit geringem Stromverbrauch, gespeist aus Batterien oder Akkus, wenn der Computer ausgeschaltet ist. Um das Ganze auch in eine handliche Form zu bringen, muß man sich etwas einfallen lassen. Schließlich wird es aber doch gelingen, wie die Bauzeichnung (Bild 2) zeigt. Die Konstruktion erinnert an das 2716-Programmiergerät. Auch hier wurde Wert darauf gelegt, daß die Gehäuseabmessungen klein sind. Das IPROM hat eine Speicherkapazität von 2 K und die gleiche Pinzahl wie das bekannte 2716-EPROM. Es ist dazu pincompatibel. Man kann das IPROM also als ROM (EPROM), aber auch als RAM verwenden. Beim Abschalten der Betriebsspannung bleiben alle Informationen erhalten. Ein Einsatz in einem anderen Computer ist deshalb gar kein Problem. Vielleicht kann sich auf diese Weise eine lebhaftere Software-Korrespondenz entwickeln. Die Bundespost würde sich freuen – der Empfänger sicher auch.

Man kann das IPROM auch als Master-EPROM in einem Programmiergerät für 2716-EPROMs einsetzen. Dann spart man sich das mühsame Von-Hand-Programmieren. Schließlich darf die eigentliche Absicht des Entwicklers nicht

Die Schaltung

Bild 1 zeigt die Schaltung des IPROMs. Als RAM ist hier der Typ HM6116LP von Hitachi eingesetzt. Der interne Aufbau und die geringe Stromaufnahme im "Stand-By-Betrieb" machen dieses IC besonders geeignet. Die Daten können in 8 bit Breite ein- und ausgelesen werden, so daß man den Datenbus ohne weiteres an viele Computersysteme anschließen kann. Die Stromaufnahme im Ruhezustand beträgt nur einige μA . Aus diesem Grund können sogenannte Knopfzellen die Schaltung monatelang versorgen. Akkus in Knopfzellen-Form sind noch besser geeignet, da sie ständig nachgeladen werden. Auf diese Weise hat man sogar die Möglichkeit, das IPROM jahrelang mit den Akkus zu betreiben.

Um das IPROM anstelle eines ROMs in der Schaltung einsetzen zu können, muß man einige Punkte beachten: Das IPROM benötigt eine Betriebsspannung von 5 V aus dem Computersystem. Eine rote LED dient hier als Indikator. Die Widerstände R1 und R2 bestimmen die Größe des LED-Stroms. Außerdem legen sie fest, wann T1 aufgesteuert wird. Das Verhältnis der beiden Widerstände ist so gewählt, das T1 bei einer Betriebsspannung über 4...4,5 V leitet. Dann liegen die Eingänge von N4 auf "0". Damit sind die Gatter N1...N3 für die an den Anschlüssen 18, 20 und 21 liegenden Informationen geöffnet. Der R/W-Eingang des RAMs ist über einen Schalter mit dem Ausgang von N3 verbunden. Ist dieser Schalter geöffnet, dann kann das RAM nur ausgelesen werden.

Fällt die Betriebsspannung des Computers aus, dann wird das RAM über D3 aus zwei in Reihe geschalteten Knopfzellen versorgt. Diode D2 trennt in diesem Fall die Batterie- von der Versorgungsspannungslleitung des Computers. Da T1 bei Batteriebetrieb sperrt, liegen die RAM-Eingänge CE, OE und R/W über D3, R6 und die Gatter N1...N4 auf "1". Das RAM ist in diesem Fall nicht aktiv.

Die Widerstände R8...R29 legen alle RAM-Eingänge auf "0"-Pegel, wenn die Betriebsspannung abgeschaltet ist. Ein nicht definierter Pegel an den Eingängen

würde nämlich zu einem erhöhten Stromverbrauch aus den Batterien führen. Wie schon erwähnt ist die Schaltung so ausgelegt, daß man sie anstelle eines 2716-EPROMs einsetzen kann. Das RAM ist "schnell" genug (etwa 250 ns plus 40 ns für die CMOS-Gatter), so daß es in praktisch alle Computer-Systeme für den Hausgebrauch paßt. Ein Pin unterscheidet sich allerdings vom 2716-EPROM: Pin 21 der Fassung, in die das EPROM gesteckt werden soll, muß über ein Stückchen Draht mit der R/W-Leitung verbunden sein. Als Batterien oder Akkus sind alle Typen geeignet, die Knopfzellenform mit den Abmessungen 11,5 x 5 mm haben. Man kann also alle Quecksilberoxid-, Silberoxid- oder Alkali-Mangan-Typen einsetzen. In diesem Fall muß aber R5 entfernt werden. Der Versuch, Batterien

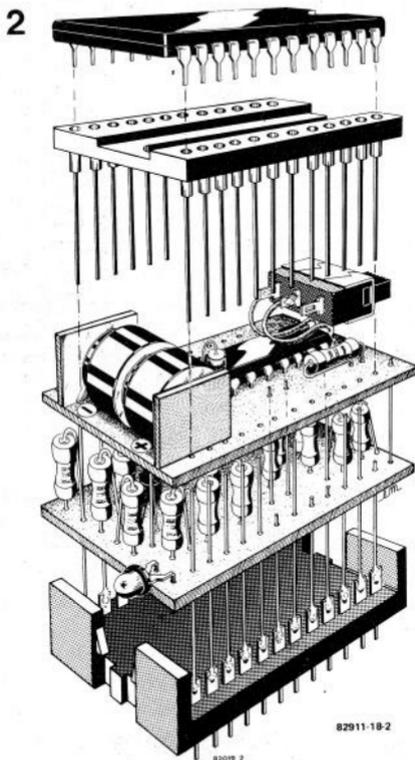


Bild 2. Der Aufbau des IPROMs ist sehr kompakt. So läßt es sich aber ohne weiteres anstelle eines 2716-EPROMs in vorhandene Fassungen stecken. Man sollte bei der Verdrahtung besonders sorgfältig vorgehen.

3

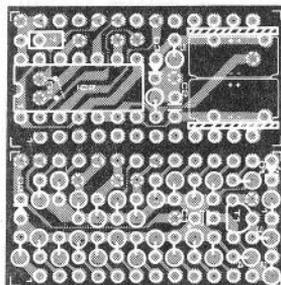
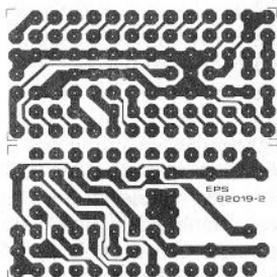


Bild 3. Layouts und Bestückungspläne der beiden IPROM-Platinen. Sie werden zwischen Wire-Wrap-Fassung und DIL-Stecker gesetzt und miteinander verlötet. Das RAM steckt man schließlich in die Fassung.

Stückliste

Widerstände (1/8 W):

R1 = 220 Ω

R2 = 56 Ω

R3 = 1 k

R4 = 4k7

R5 = 2k2 nur in Verb. mit NiCd-Akkus

R6 . . . R26 = 47 k

Kondensatoren:

C1 = 4 μ 7/16 V Tantal

C2 = 10 n ker.

Halbleiter:

D1 = LED rot 3 mm

D2, D3 = DUG

T1 = BC 547B

IC1 = HM 6116LP (Hitachi)

IC2 = 4071B

außerdem:

24polige Wire-Wrap-Fassung

24poliger DIL-Stecker

2 NiCd-Akkus oder Batterien

(Knopfzellen) ϕ 11,5 mm x 5 mm

z.B. V675PX (Quecksilberoxid), V76HS

(Silberoxid), V13GA, V13GS, 20 DK (NiCd)

— alle VARTA

S1 = 1poliger Schalter

4

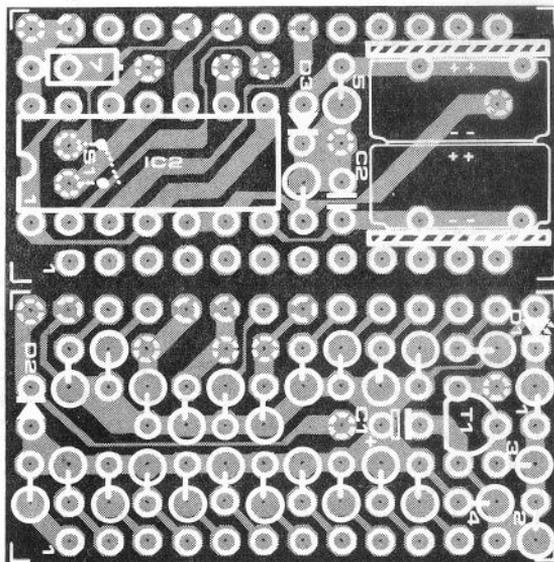


Bild 4. Hier ist der Bestückungsplan vergrößert dargestellt. Sonst muß man sich zum Betrachten vom Bild 3 extra eine Lupe kaufen.

aufzuladen, ist meist nicht sehr erfolgreich. Also – R5 nur einsetzen, wenn Akkus verwendet werden.

Der Aufbau

Bild 2 zeigt sehr gut den Aufbau des IPROMs. Sorgfältige Arbeit ist unbedingt vonnöten. Schließlich soll das Ganze so kompakt wie möglich aussehen und auch störungsfrei funktionieren. Zunächst bestückt man die Platinen (siehe Bild 3). Danach erfolgt der Zusammenbau "von oben nach unten" gemäß Bild 2. Den Schalter kann man mit Zweikomponentenkleber auf die Unterseite der Fassung kleben. Diese Fassung sollte ein Wire-Wrap-Typ sein, der längere und stabilere Anschlußbeinchen als normale Fassungen hat. Die Akkus

oder Batterien müssen fest auf der Platine sitzen. Am besten eignen sich hier Typen mit Lötflächen, die mit doppel-seitigem Klebeband auf der Platine befestigt und mit zwei Platinenstückchen zusätzlich gegen Herausfallen gesichert werden. Die Verbindungen der Fassung, der Platinen und schließlich des DIL-Steckers werden ebenfalls in der Reihenfolge von oben nach unten festgelötet. Hat man sich davon überzeugt, daß es keine Kurzschlüsse in dem Aufbau gibt, dann kann auch das RAM eingesetzt werden.

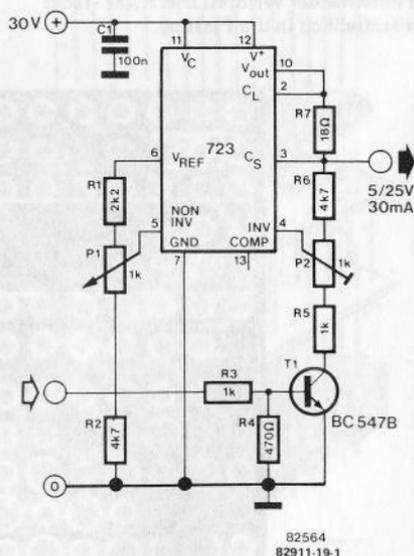
Wichtig ist vor allem, daß das IPROM nur bei ausgeschaltetem Computer eingesetzt und ebenso herausgenommen wird. Batterien sollte man jedes Jahr auswechseln.

Einfache Programmier-Spannungsquelle

Der "Preisverfall" bei den EPROMs im Laufe der letzten Jahre hat dazu geführt, daß allorts fleißig selbst programmiert wird. Zum Programmieren des bekannten 2716 braucht man eine Spannung von 25 V. Das bedeutet: Normalerweise trägt die Spannung am "Programmierpin" 5 V. Beim Programmieren wird diese Spannung auf 25 V vergrößert. Es ist übrigens bei einigen EPROM-Typen auch möglich, bei der Kontrolle des frisch Programmierten diese 25-V-Spannung eingeschaltet zu lassen. Sonst muß ständig zwischen den beiden Pegeln umgeschaltet werden.

Ein gutes EPROM-Netzteil muß deshalb einfach, schnell, genau und kurzschlußfest sein. Was die Schnelligkeit betrifft, geben die Hersteller im allgemein zwischen 0,5 und 2 μ s an. Was die Genauigkeit betrifft, darf die Gleichspannung beim Einschalten weder überschwingen noch unter dem Nennwert bleiben. Alle diese Forderungen lassen sich sehr gut mit dem bekannten Spannungsregler-IC 723 erfüllen.

Die Schaltung sieht im Prinzip wie eine normale 5-V-Stabilisierung aus. Das ist



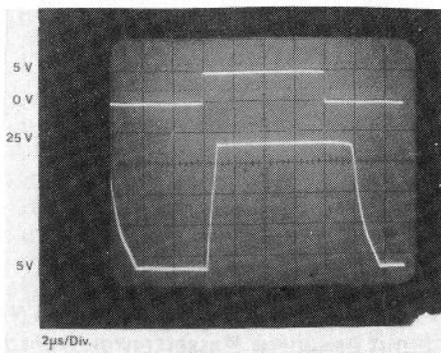
auch richtig! Denn die Referenzspannung von Pin 6 wird mittels P1 auf 5 V eingestellt und an den Plus-Eingang des internen Operationsverstärkers gelegt. Solange T1 sperrt, liegt auch die gesamte

Ausgangsspannung am Minus-Eingang (Pin 4) des internen OpAmps, und am Ausgang der Schaltung liegt 5 V. R7 legt den Einsatz der Strombegrenzung fest.

Die Programmierspannung wird dadurch erzeugt, daß Pin 4 eine andere Spannung erhält. Liegt dort nämlich ein entsprechend dimensionierter Spannungsteiler an, dann kann die Ausgangsspannung größer als 5 V werden. Dieser Spannungsteiler besteht aus R5, R6 und P2; und er wird mittels T1 eingeschaltet. Sobald dieser Transistor eine Basisspannung erhält, erscheint am Ausgang der Schaltung die Programmierspannung von 25 V.

Der Abgleich ist leicht durchzuführen: Basis von T1 nicht ansteuern → mit P1 Ausgang auf 5 V einstellen. Basis von T1 ansteuern (R3 an 5 V) → mit P2 Ausgang auf 25 V einstellen. Das ist alles!

Das Foto zeigt den Zusammenhang zwischen Steuersignal an R3 (oben) und Ausgangssignal (unten). Das Spannungsregler-IC ist deshalb so "schnell", weil die Kompensation fehlt (Pin 13). Ein



Kondensator an dieser Stelle würde dazu führen, daß die Flanken des Ausgangssignals weniger steil verlaufen. Außerdem fällt sofort auf, daß es nach Ausbleiben des Steuersignals etwa 2 μ s dauert, bis das Ausgangssignal wieder auf 5 V abfällt. Das liegt am "langsamen" Transistor T1. Schneller wird's nur dann, wenn man dort einen Schalttransistor (z.B. BSX 20), einen CMOS-Schalter (z.B. 4066) oder einen V-FET (z.B. BS 170) einsetzt. Der CMOS-Schalter stellt vermutlich die preiswerteste Lösung dar.

Kapitel 5: Kassetten-Interface

Von langsam bis schnell

Ein Kassetten-Interface benötigt wohl jeder Mikrocomputer-Besitzer, damit er seinen Kassettenrekorder als "Massenspeicher und Langzeitgedächtnis" einsetzen kann. Dabei ist der Kansas-City-Standard nicht die einzige Möglichkeit Computersignale in "digitale Audiosignale" umzusetzen. In diesem Kapitel möchten wir drei Möglichkeiten davon vorstellen.

Als Interface bezeichnet man gewöhnlich die Hardware, über die Peripherie-Geräte mit einem Computer verbunden sind. Ein Kassetten-Interface ist demnach eine Schaltung, die den Anschluß eines Kassettenrekorders an einen (Mikro-)Computer ermöglicht. Die kostengünstige Kompaktkassette kann mit Hilfe des Interfaces als Massenspeicher für Informationen in einer für den Mikrocomputer "lesbaren" Form dienen. Selbstverständlich lassen sich auch Spulen-Tonbandgeräte verwenden; in der Praxis reicht jedoch normalerweise auch für die längsten Programme die Kapazität einer C 60-Kassette aus, so daß größere Spulen lediglich unhandlicher wären. Hinzu kommt, daß ein Kassetten-Interface besonders niedrige Anforderungen an die elektroakustischen Eigenschaften des Rekorders stellt; für den genannten Zweck ist daher jeder "Low Cost"-Kassettenrekorder geeignet.

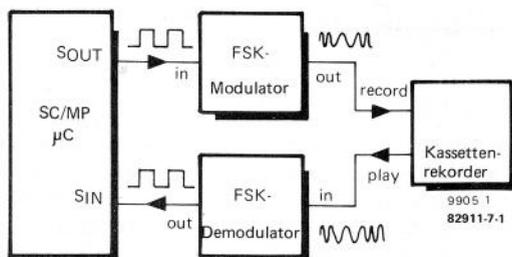
Ein Interface kommt nicht ohne Software aus. Die Software sorgt dafür, daß der Speicherinhalt des Mikrocomputers seriell ein- und ausgelesen werden kann, während die Hardware die serielle Informationen in Audio-Signale umsetzt und später die Audio-Signale in serielle Informationen rückwandelt.

Prinzip

Im Prinzip kann die Interface-Hardware aus einem FSK-Modulator (FSK = Frequency Shift Keying) und einem FSK-Demodulator bestehen (siehe Bild 1). Der FSK-Modulator erzeugt ein sinusförmiges Ausgangssignal mit einer Frequenz von 2400 Hz, wenn an seinem Eingang eine logische "1" liegt. Ist daß

Bild 1. Schematische Darstellung des Kassetten-Interface-Prinzips.

1



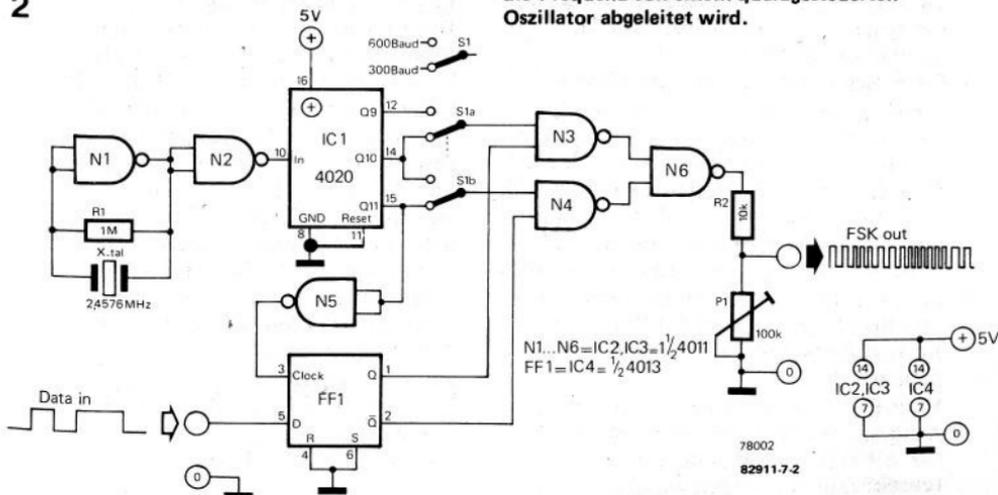
Eingangssignal dagegen logisch "0", so beträgt die Ausgangsfrequenz 1200 Hz. Die Modulatorsignale werden einfach mit dem Kassettenrekorder auf Band aufgezeichnet.

Bei umgekehrtem Informationsfluß gelangen die Rekorder-Ausgangssignale zum FSK-Demodulator. Dieser liefert bei einer Frequenz von 2400 Hz eine "1" und bei 1200 Hz eine "0" an seinem Ausgang. Die wieder in ihre ursprüngliche Form umgewandelten Signale werden auf den Serieneingang des μ computers gegeben; die Soft- und Hardware sorgt in diesem Fall dafür, daß die seriell angelieferten Daten parallel (als Bytes) in den Speicher geschrieben werden.

Die Festlegung der beiden genannten Frequenzen ist nicht willkürlich. Es existieren hierfür verschiedene "Standards"; häufig handelt es sich um den sogenannten Kansas City Standard (KCS). Diese Norm schreibt die Frequenzen 2400 Hz und 1200 Hz sowie eine Übertragungsgeschwindigkeit von 300 Bd (Baud) vor. Mit diesen Frequenzen läßt sich jeder beliebige Rekorder (jedes Diktiergerät usw.) zur Datenaufzeichnung verwenden; außerdem ist im Prinzip die Datenübertragung über das Fernsprechnet (Frequenzbereich 300 ... 3000 Hz) möglich.

Die Übertragungsgeschwindigkeit wird in der Regel nicht vom Kassetten-Interface

2

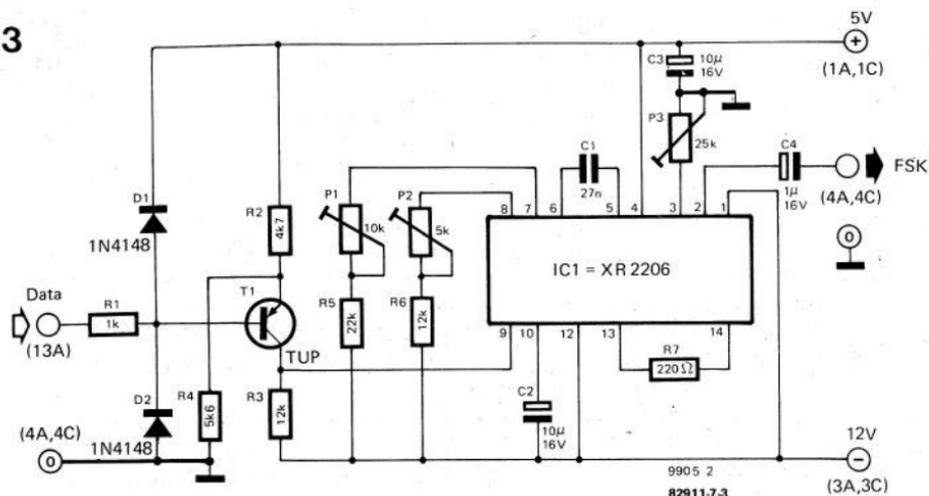


selbst bestimmt, sondern von einem Seriell/Parallel-Umsetzer (UART oder ACIA) oder von der Software. Welche Lösung nun zur Anwendung kommt, hängt von den Möglichkeiten der CPU ab. Mit den genannten Frequenzen kann die Übertragungsgeschwindigkeit maximal 1200 Baud betragen. Bei einer Geschwindigkeit von 1200 Bd dauert übrigens die Ein- oder Ausgabe eines 64 k-Bytes umfassenden Programms (gesamte Speicherkapazität des SC/MP-Systems) immerhin noch ca. 10 Minuten.

2 x FSK-Modulator

Der Aufbau von FSK-Modulatoren ist relativ einfach. Lediglich der Abgleich ist etwas schwierig. Deshalb folgt zunächst die Beschreibung einer Schaltung, die ohne Abgleichelemente konzipiert ist. Erst dann beschäftigen wir uns mit einem einstellbaren Modulator. Es ist klar, daß beide Lösungen ihre Vor- und Nachteile haben. Welche das sind, geht aus der nachfolgenden Beschreibung hervor. Zu der Modulatorschaltung ohne Einstellmöglichkeit aus Bild 2 genügen wenige Bauelemente. Die von einem quarzgesteuerten Oszillator erzeugte Frequenz (hier 2,4576 MHz) wird von einem

Bild 2. Bei diesem einfachen Kansas-City-Modulator ist kein Abgleich erforderlich, weil die Frequenz von einem quarzgesteuerten Oszillator abgeleitet wird.



14-stufigen Binärteiler (IC1) herabgeteilt. An Ausgang Q10 steht eine Frequenz von 2400 Hz zur Verfügung, während die Frequenz an Ausgang Q11 genau die Hälfte, also 1200 Hz beträgt.

Da der Oszillator quarzstabilisiert ist, sind auch diese beiden Frequenzen äußerst konstant. Abhängig vom Signal am Daten-Eingang ("0" oder "1") gelangt eine der beiden Frequenzen zum Ausgang. Das Umschalten der Frequenz erfolgt durch Flipflop FF1 zusammen mit den beiden Gattern N3 und N4. Da das Flipflop vom 1200 Hz-Signal getaktet wird, besteht das FSK-Signal stets aus vollständigen Perioden des 1200 Hz- bzw. des 2400 Hz-Signals. Dies ist notwendig, um die spätere Demodulation des FSK-Signals zu erleichtern.

Steht Schalter S1 in der gezeichneten Stellung, so eignet sich der Modulator für eine Übertragungsgeschwindigkeit von 300 Bd (Baud = Bits pro Sekunde). Nach Umschalten von S1 kann die Geschwindigkeit 600 Bd betragen. In diesem Fall werden die Frequenzen 2400 Hz und 4800 Hz zum Ausgang geschaltet, so daß die Erkennbarkeit gleich groß wie bei 300 Bd ist. Die Anzahl der Perioden pro Bit bleibt nämlich bei beiden Geschwindigkeiten gleich.

Vom Einstellpoti P1 hängt die Höhe der Ausgangsspannung ab. Wenn der Modulator mit einem Tonbandgerät oder Kassettenrekorder verbunden werden soll, muß

Bild 3. FSK-Modulator für Übertragungsgeschwindigkeiten bis 1200 Baud. Das sinusförmige Ausgangssignal des Modulators läßt sich in der Amplitude einstellen.

möglicherweise ein Tiefpaßfilter zwischen Modulatorausgang und TB-Eingang geschaltet werden. Ein einfaches RC-Glied, dessen Grenzfrequenz bei ca. 5 kHz liegt, reicht für diesen Zweck aus. Die angegebene Quarzfrequenz von 2,4576 MHz ist trotz ihres "krummen" Werts durchaus handelsüblich. Man ist jedoch keineswegs an diese Frequenz gebunden, sondern kann bei abweichender Oszillatorfrequenz eventuell andere Teilerausgänge benutzen und mit anderen FSK-Frequenzen arbeiten.

Der FSK-Modulator erzeugt von beiden Frequenzen immer komplette Perioden. Eine Tatsache, die den Aufbau eines Demodulators sehr vereinfacht. Der Nachteil dieser Schaltung ist die relativ geringe Übertragungsgeschwindigkeit von nur 300 Baud. Dadurch wird der Operator, speziell beim Übertragen von langen Programmen, immer zu Zwangspausen veranlaßt. Um diese Zeiten möglichst gering zu halten, kann die Schaltung aus Bild 2 auch eine Übertragungsgeschwindigkeit von 600 Baud erzeugen. Dadurch verdoppeln sich die Frequenzen auf 2400 Hz und 4800 Hz, und das kann für einen preiswerten Kassettenrekorder bereits zu schnell sein. Erst recht für die Datenübertragung per

Fernsprechnetz. Um all diese Schwierigkeiten auszuschließen, müßte es also möglich sein, mit einer noch höheren Übertragungsgeschwindigkeit von z.B. 1200 Baud die niedrigen Modulationsfrequenzen von 1200 Hz und 2400 Hz zu benutzen. Diese Möglichkeit besteht tatsächlich, sie ist mit der Schaltung aus Bild 3 realisiert. Dieser FSK-Modulator ist mit dem bekannten Funktionsgenerator-IC XR 2206 aufgebaut. Die Speisenspannungen der Schaltung betragen +5 V und -12 V. Um das IC mit TTL-Signalen steuern zu können, wurde Transistor T1 hinzugefügt.

In Bild 7 sind das Eingangssignal (a) und das Ausgangssignal (b) des Modulators grafisch dargestellt. Die Amplitude des Ausgangssignals läßt sich mit Poti P3 einstellen und an die Eingangsempfindlichkeit des Rekorders anpassen. Mit den Potis P1 und P2 können die Frequenzen des Audio-Signals (1200 Hz bzw. 2400 Hz) eingestellt werden. Steht kein geeigneter Frequenzmesser zur Verfügung, so kann zum Abgleich auch, wie nachfolgend beschrieben, die CPU selbst dienen.

Jede CPU benötigt für das genaue Timing einen quartzesteuerten Taktgenerator. Ferner ist von jeder Instruktion die Zeit bekannt, welche für die Durchführung notwendig ist. Mit diesen Kenntnissen ist es möglich, ein Programm zu schreiben, das exakt die erforderlichen Frequenzen von 1200 Hz und 2400 Hz erzeugt. Falls der Mikrocomputer einen programmier-

4

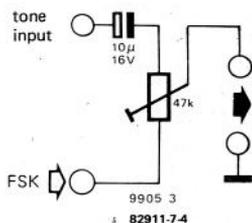


Bild 4. Mit dieser einfachen Hilfsschaltung kann der Modulator aus Bild 3 eingestellt werden. An den Eingang tone-input muß ein geeignetes Signal vom Mikrocomputer gelegt werden, damit der Frequenzvergleich möglich ist.

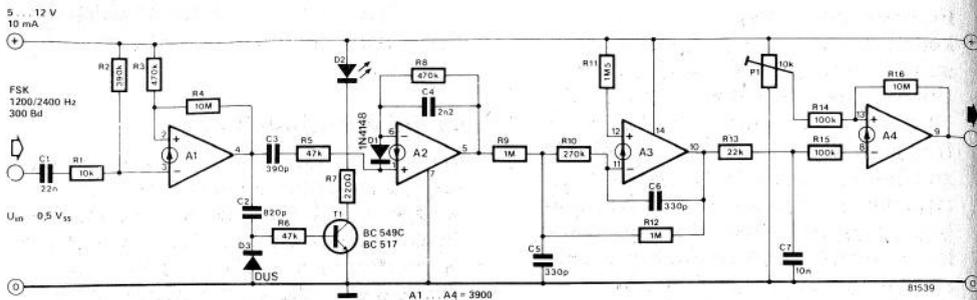
baren Timer benutzt, ist der Abgleich mit dem Hilfsprogramm noch einfacher.

Modulator-Einstellung

In den vorhandenen Mikrocomputer muß zuerst das Testprogramm eingegeben werden, so daß eine genügend konstante Signalfrequenz von 1200 Hz oder 2400 Hz zur Verfügung steht. Diese Rechtecksignale dienen dann zur Einstellung des Modulators. Man verbindet dazu den Modulator-Ausgang mit dem Tonausgang des Mikrocomputers mit der in Bild 4 angegebenen Meßschaltung. Am Ausgang der Meßschaltung wird ein hochohmiger Kopfhörer oder ein Kassettenrekorder mit Aussteuerungsanzeige angeschlossen. Nach dem Start des 1200 Hz-Programms legt man an den Eingang des Modulators eine logische "0"; Poti P3 steht dabei in Stellung maximaler Amplitude. Im Kopfhörer sind nun verschiedene Frequenzen hörbar: der 1200 Hz-Ton des Programms, der vom Modulator erzeugte Ton sowie die Differenzfrequenz von beiden. Poti P1 wird auf eine möglichst niedrige Differenzfrequenz eingestellt. Die Schwebung, die das Aussteuerungsinstrument des Rekorders anzeigt, ist dann minimal. Die Frequenzen von Programm und Modulator stimmen nach dieser Porzedur sehr genau überein. Die Einstellung der 2400 Hz-Frequenz wird auf die gleiche Weise vorgenommen. Natürlich muß das Programm dann einen 2400 Hz-Ton erzeugen, während der Modulator-Eingang auf logisch "1" liegt.

Kansas-City-Demodulator

Dieser Kansas-City-Demodulator (Bild 5) verwendet als Besonderheit eine "Ladungspumpe". Keine Angst — das ist nichts Mechanisches, sondern eine spezielle Schaltungstechnik. Zunächst einmal ein Überblick: Jeder Operationsverstärker hat eine andere Funktion. A1 arbeitet als Komparator und formt auf diese Weise aus dem, was ihm vom Kassettenrekorder angeboten wird, ein "sauberes" Rechtecksignal. A2 ist die "Ladungspumpe", die das



Komparatorsignal mit einer Frequenz von 1200 oder 2400 Hz in eine kleine oder große Spannung umsetzt. A3 bildet ein Tiefpaßfilter das dekodierte Signal. Schließlich ist mit A4 ein zweiter Komparator aufgebaut, der an seinem Ausgang eine Rechteckspannung zur Verfügung stellt. Die Stufe mit T1 zeigt an, ob das Eingangssignal ausreichend groß ist. Das war sozusagen die Schaltungsbeschreibung im Schnelldurchgang.

Eine Detailbeschreibung ist allerdings auch notwendig, denn die hier verwendeten Operationsverstärker funktionieren etwas anders als gewohnt. Das IC 3900 enthält nämlich 4 Verstärker, die nicht auf Eingangs-Spannungen, sondern auf Eingangs-Ströme reagieren. Der Ausgang der ersten Komparators A1 liegt beispielsweise im Ruhezustand auf Massepotential. Der in den invertierenden Eingang (über R2) fließende Strom ist in diesem Fall größer als der in den nicht-invertierenden Eingang (über R3) fließende Strom.

Die Ladungspumpe funktioniert folgendermaßen: Liegt kein Signal am Eingang an, dann entlädt C4 sich über R8. Der Ausgang von A2 liegt also an Masse. Eine positive Flanke des Komparatorausgangssignals hat einen kurzen Stromimpuls am nicht-invertierenden Eingang von A2 zur Folge (über C3). Wie bei jedem Operationsverstärker, stellt sich auch hier ein Gleichgewichtszustand ein, der dazu führt, daß ein ebenso großer Strom in den invertierenden Eingang fließt. Dieser Strom kann nur über C4 fließen. Er lädt diesen Kondensator teilweise auf. Mit jedem neuen Stromimpuls wird C4 etwas weiter aufgeladen, so daß die Ausgangsspannung von A2 in gleichem

Bild 5. Für den einfachen Kansas-City-Demodulator ist der Abgleich unproblematisch. Es ist lediglich nur eine Einstellung erforderlich.

Maße ansteigt. Über R8 entlädt sich der Kondensator wieder. Die Ausgangsspannung von A2 sinkt. Je mehr Impulse eintreffen, desto größer wird die Ausgangsspannung der "Ladungspumpe" sein. Auch die negativen Flanken des Komparatorausgangssignals tragen über D1 zur Ladung von C4 und damit zur Erhöhung der Ausgangsspannung von A2 bei.

A3 ist ein aktives Tiefpaßfilter. Die Eckfrequenz ist so gewählt, daß sie gerade über der maximal auftretenden Signalfrequenz von 150 Hz liegt. Zur Erklärung: An A3 liegen die Daten bereits mit einer Baudrate von 300 Bd an. 300 Bd entsprechen 150 Hz!

Da das dekodierte Signal hinter dem Tiefpaßfilter nur eine geringe Amplitude und weniger steile Flanken besitzt, ist der Komparator A4 als Impulsformer eingesetzt. Sollte die Phasenlage des Ausgangssignals nicht stimmen, so ist R14 an den invertierenden und R15 an den nicht-invertierenden Eingang von A4 anzuschließen.

Die Stromaufnahme der Schaltung ist etwas von der Betriebsspannung (letztere sollte genauso groß wie die der nachfolgenden Schaltung sein) abhängig.

P1 wird so eingestellt, daß logisch 0 und logisch 1 bei einem Eingangssignal aus 8 Perioden mit 2400 Hz und 4 Perioden mit 1200 Hz genau gleich "lang" sind. Dieses Impulsmuster kann man leicht mittels eines kleinen Programms durch den Modulator erzeugen lassen.

Genau wie der Modulator aus Bild 3 ist

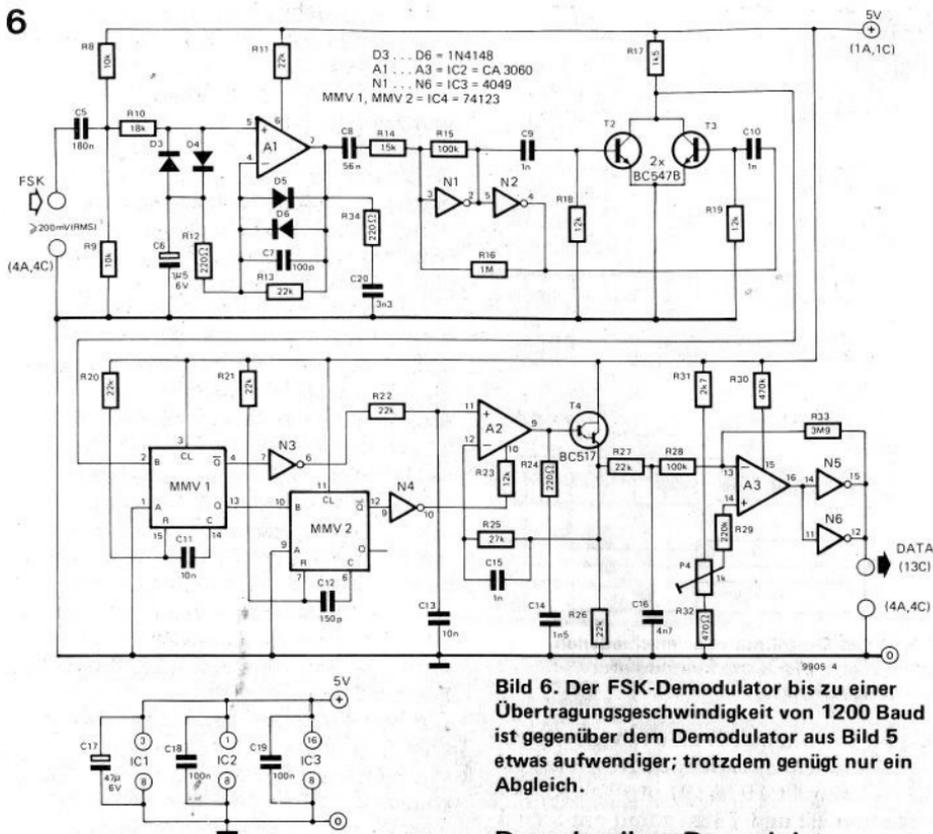


Bild 6. Der FSK-Demodulator bis zu einer Übertragungsgeschwindigkeit von 1200 Baud ist gegenüber dem Demodulator aus Bild 5 etwas aufwendiger; trotzdem genügt nur ein Abgleich.

Der schnellere Demodulator

Mehr Komfort bedeutet in der Regel mehr Aufwand. So auch hier. Die in Bild 6 dargestellte Demodulatorschaltung ist schneller gegenüber der Schaltung aus Bild 5, dafür aber auch aufwendiger. Bedeutend komplizierter als der Modulator ist die zur Demodulation des FSK-Signals benötigte Schaltung, die Bild 6 zeigt. Da ihre Arbeitsweise von sonst für diesen Anwendungszweck gebräuchlichen Prinzipien (z.B. PLL) abweicht, erscheint es angebracht, diese an Hand der in Bild 7 skizzierten Spannungsverläufe etwas näher unter der Lupe zu nehmen.

Das FSK-Signal (b in Bild 7) liegt am Eingang des Demodulators. Die als Begrenzer wirksame Verstärkerstufe mit OpAmp A1 clippt das Signal symmetrisch (c); dieses Signal wird einem Schmitt-Trigger (N1, N2) zugeführt. Das Ausgangssignal des Schmitt-Triggers

auch dieser Demodulator nur für eine relativ geringe Übertragungsgeschwindigkeit geeignet. Hier gilt das Motto: Langsam, aber sicher.

Für Mikrocomputer-Anwender die es eilig haben, sind 300 Baud viel zu langsam. Eine Übertragungsgeschwindigkeit von 1200 Baud kommt ihnen da schon eher entgegen. Dabei spielen allerdings die Aufnahmequalität und das verwendete Bandmaterial eine wichtige Rolle mit. Das ist verständlich, wenn man bedenkt, daß bei 1200 Baud lediglich eine Periode von 1200 Hz bzw. zwei Perioden von 2400 Hz auf dem Band abgespeichert sind. Es genügt nur ein Dropout und das entsprechende Bit ist verschwunden. In der Praxis ist das kein ungewöhnlicher Fehler. Folglich gehört in dieses Kapitel auch die Beschreibung eines Demodulators, der die Übertragungsgeschwindigkeit von 1200 Baud ohne Schwierigkeiten verarbeiten kann.

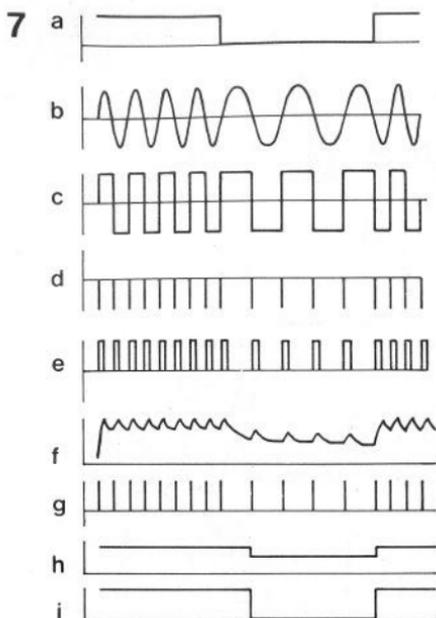


Bild 7. Das Diagramm von verschiedenen Spannungen, die in der Demodulatorschaltung auftreten.

gelangt sowohl direkt als auch invertiert zu einem differenzierenden RC-Netzwerk (C9, R18 und C10, R19). An den Kollektoren von T2 und T3 ist somit ein Signal vorhanden, das die doppelte Frequenz des ursprünglichen, am Eingang des Schmitt-Triggers liegenden Signals aufweist (d). Dieses Signal steuert ein nachfolgendes Monoflop. Am Ausgang des Monoflops erscheinen Impulse von konstanter Breite (e), die einerseits einem integrierenden Netzwerk (R22, C13) zugeführt werden und andererseits ein zweites Monoflop steuern.

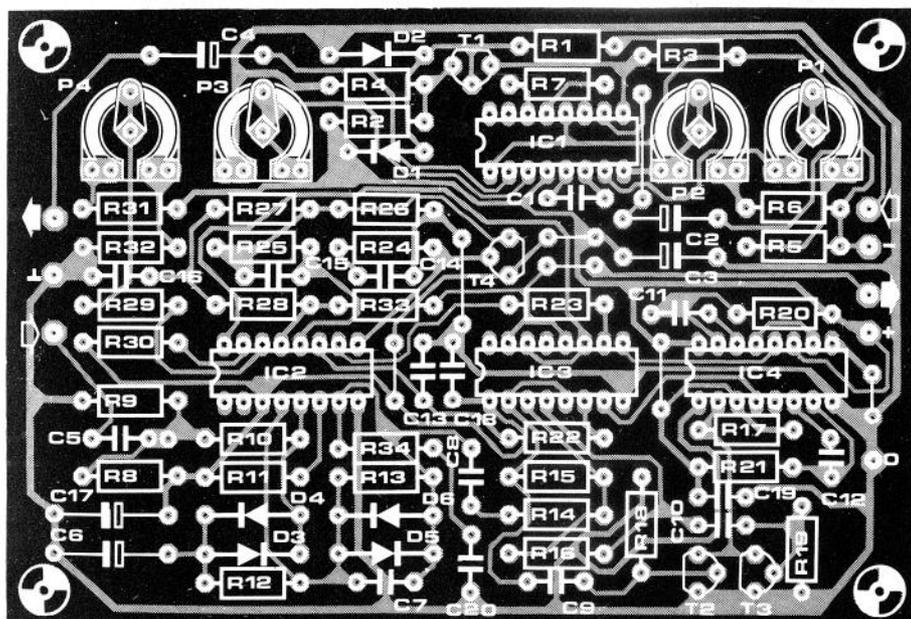
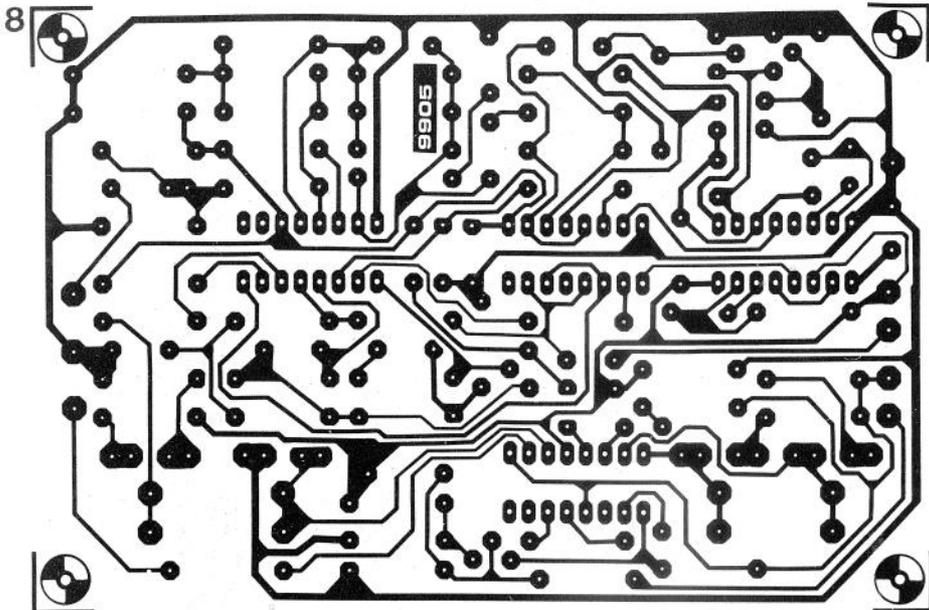
Die Spannung an Kondensator C13 (f) ist ein Kriterium für die Frequenz des Eingangssignals; dieser Kondensator erhält nämlich bei hoher Eingangsfrequenz (2400 Hz) die doppelte Ladungsmenge pro Zeiteinheit zugeführt, als diese bei niedriger Eingangsfrequenz (1200 Hz) der Fall ist. Um die relativ geringe Differenz der Gleichspannungskomponente an Kondensator C13 in digitale Informationen umzusetzen, wird von einer Sample-Hold-Schaltung Gebrauch gemacht; sie ist mit A2 und T4 sowie mit

C14 als Speicherelement aufgebaut. Diese Stufe tastet während der vom zweiten Monoflop erzeugten Sample-Impulse (g) die Spannung an Kondensator C13 ab und speichert deren Wert bis zum nächsten Sample-Impuls. Ist die Frequenz des Demodulator-Eingangssignals konstant, so ändert sich auch die Spannung am Ausgang der Sample-Hold-Schaltung (Emitter von T4) nicht. Tritt dagegen eine Frequenzänderung ein, dann nimmt die Spannung an C13 einen anderen Wert an; mit dem nächsten Sample-Impuls ändert sich auch die Ausgangsspannung der Sample-Hold-Schaltung (h). Am Ausgang der Sample-Hold-Schaltung ist bereits ein digitales Signal vorhanden; die Flankensteilheit und Amplitude dieses Signals ist jedoch so gering, daß das Signal von einem Komparator (A3 mit N5 und N6) auf TTL-Niveau gebracht werden muß. Die Einstellung des Modulators beschränkt sich übrigens auf die Festlegung der Komparator-Schaltweise; auch hier leistet ein einfaches Programm Hilfestellung.

Das Hilfsprogramm muß die in Bild 7a und 7c gezeigten Signale erzeugen. Das symmetrische Rechtecksignal aus Bild 7a steuert den Modulator aus Bild 3. Das vom Computer produzierte Signal (Bild 7c) kann direkt an den Demodulator angeschlossen werden. Damit die Einstellung optimal ist, muß die Periodenzeit des Rechtecksignals aus Bild 7a immer das geradzahlige Vielfache der 1200-Hz-Periode betragen, mindestens jedoch das Zweifache.

Demodulator-Einstellung

Führt man dem Modulator ein "symmetrisches" Eingangssignal zu, das heißt ein Signal, bei dem hohe Töne (2400 Hz) und tiefe Töne (1200 Hz) genau gleich lang sind, dann muß auch das Ausgangssignal symmetrisch sein. Diese Einstellung erfolgt mit P4. Ein symmetrisches Rechtecksignal, dessen Amplitude zwischen Null und dem Wert der Speisespannung springt, weist als Spannungsmittelwert die halbe Speisespannung auf. Schließt man daher an den Ausgang ein Gleichspannungsvoltmeter an, so zeigt dieses



bei richtiger Einstellung von P4 die Hälfte der Speisespannung (2,5 V) an. Weitere Einstellungen sind am Demodulator nicht erforderlich.

Wer diesem Abgleich nicht hundertpro-

Bild 8. Platinenlayout und Bestückungsplan für die Modulatorschaltung aus Bild 3 und für den Demodulator aus Bild 6.

zentig vertraut und ein Oszilloskop zur Verfügung hat, kann selbstverständlich den Abgleich damit kontrollieren und die Symmetrie exakt einstellen. Mit Hilfe eines Zweikanal-Oszilloskops wird die Einstellung noch genauer. Es besteht dann nämlich die Möglichkeit, das Impuls/Pausenverhältnis des Ausgangssignals dem des Eingangssignals anzupassen.

Platine

Für den Modulator aus Bild 3 und den Demodulator aus Bild 6 steht eine Platine zur Verfügung (Bild 8), die beide Schaltungen aufnehmen kann. Die Bauteilnumerierung beginnt in Bild 3 und setzt sich in Bild 6 fort. Dies gilt auch für den Bestückungsaufdruck der Platine. Bild 9 zeigt die Verbindung zwischen Interface und Mikrocomputer.

Die schnellste Version

Eine Übertragungsgeschwindigkeit von 1200 Baud ist schon ganz beachtlich, doch ist auch hier noch eine Steigerung möglich. Durch die Verdoppelung der FSK-Frequenz kann man die Übertragungsgeschwindigkeit auf 2400 Baud erhöhen. Doch das ist noch nicht alles. Mit einem weiteren Trick, nämlich der Verwendung des Kassettenrekorders als Digitalrekorder, sind Übertragungsgeschwindigkeiten von 4800 Baud oder gar 9600 Baud möglich. Es stellt sich dann jedoch die Frage, wie es um die Zuverlässigkeit bestellt ist.

Die Schaltung, die wir in diesem Abschnitt vorstellen, hat den Vorteil, ausgezeichnet zu funktionieren. Dafür ist sie – theoretisch jedenfalls – weniger zuverlässig. Das hängt damit zusammen, daß die für Audio-Anwendungen entwickelten Rekorder für diese Zwecke weniger geeignet sind als spezielle, professionelle Typen. Diese arbeiten nämlich mit wesentlich höherer Bandgeschwindigkeit, und das Band wird ständig bis zur Sättigung ausgesteuert. Ein Effekt, den der Hersteller eines Audio-Rekorders unbedingt vermeiden muß, damit keine Verzerrungen entstehen. Außerdem

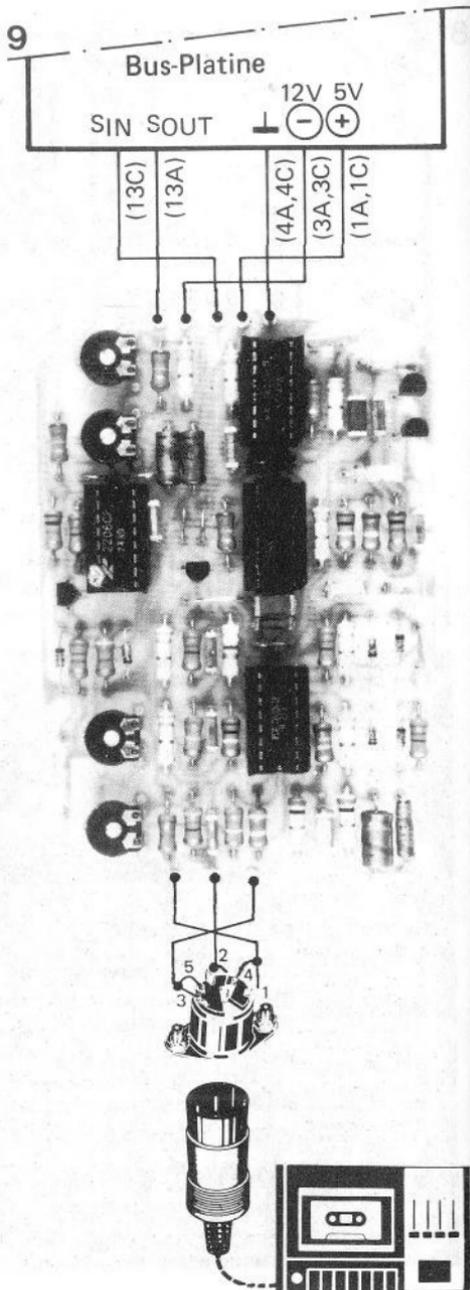


Bild 9. Auf diese Weise wird das Interface mit dem Mikrocomputer verbunden. Die Anschlußbezeichnungen stimmen mit dem in diesem Buch beschriebenen Bussystem überein. Die Abkürzungen **S_{IN}** und **S_{OUT}** bedeuten serial-in bzw. serial-out.

Stückliste zu Bild 8

Widerstände:

R1 = 1 k
R2 = 4k7
R3, R6, R18, R19, R23 = 12 k
R4 = 5k6
R5, R11, R13, R20, R21, R22, R26,
R27 = 22 k
R7, R12, R24, R34 = 220 Ω
R8, R9 = 10 k
R10 = 18 k
R14 = 15 k
R15, R28 = 100 k
R16 = 1 M
R17 = 1k5
R25 = 27 k
R29 = 220 k
R30 = 470 k
R31 = 2k7
R32 = 470 Ω
R33 = 3M9
P1 = 10 k
P2 = 5 k
P3 = 25 k
P4 = 1 k

Kondensatoren:

C1 = 27 n
C2, C3 = 10 μ /16 V
C4 = 1 μ /16 V
C5 = 180 n
C6 = 1,5 μ /6 V
C7 = 100 p
C8 = 56 n
C9, C10, C15 = 1 n
C11, C13 = 10 n
C12 = 150 p
C14 = 1n5
C16 = 4n7
C17 = 47 μ /6 V
C18, C19 = 100 n
C20 = 3n3

Halbleiter:

D1 . . . D6 = 1N4148
T1 = BC557 (TUP)
T2, T3 = BC547 (TUN)
T4 = BC517
IC1 = XR 2206
IC2 = CA 3060
IC3 = 4049
IC4 = 74123

wirkt der "Bias"-Strom in einem Audio-Rekorder der Verzerrung entgegen und beeinflusst deshalb den Frequenzgang ungünstig – für Datenübertragung jedenfalls! Alles in allem ist also ein Audio-

Rekorder weniger als Datenspeicher für digitale Informationen geeignet. Trotzdem ist die Zuverlässigkeit des hier beschriebenen Interfaces mit derjenigen der zuvor erwähnten Verfahren vergleichbar. Und – da die Übertragungsgeschwindigkeit wesentlich größer ist – kann man durchaus von einem Fortschritt sprechen. Bereits vor etwa 10 Jahren wurde von einem gewissen Herrn Tarbell vorgeschlagen, den Manchester-II-Kode für Hobby-Anwendungen einzusetzen. Dieser Kode war zu der Zeit bei den "Profis" bekannt und auch verbreitet. Zwar verwendete fast jeder Anwender eine eigene Variante, aber im Prinzip unterscheiden sich die Spezial-Versionen nicht voneinander.

Der Manchester-Kode besitzt ein Reihe von Vorteilen: Er ist sehr wirkungsvoll. Man kann sehr leicht ein Taktsignal darin "wiederfinden". Die Schaltung braucht nicht unnötig kompliziert sein. Der Wirkungsgrad beträgt 50%. Das bedeutet: Die Baudrate der Daten darf genauso groß wie die höchste vorkommende Frequenz sein. Modernere Kodes erreichen zwar die doppelte Baudrate (im Verhältnis zur höchsten Frequenz), allerdings nur mit wesentlich komplizierteren Schaltungen. Ein weiterer Vorteil: Die tiefste vorkommende Frequenz ist gleich der halben Maximalfrequenz. An den Frequenzgang des Rekorders werden also keine allzu hohen Anforderungen gestellt. In der Praxis bedeutet das: Bei einer Übertragungsgeschwindigkeit von 4800 Baud muß der Rekorder seine obere Grenzfrequenz bei 4800 Hz haben. Liegt die Grenzfrequenz bei 10 kHz, dann sind theoretisch Übertragungsgeschwindigkeiten bis zu 9600 Baud möglich.

Manchester II

Den Manchester-Kode kann man auf unterschiedliche Weise beschreiben. Wir beginnen mit der leichten, so daß es nachher nicht mehr schwer fällt, auch die anderen zu verstehen.

Wir gehen zunächst von einem Taktsignal von beispielsweise 4800 Hz aus. Das Manchester-Signal *muß* bei jeder abfallenden Flanke des Taktsignals

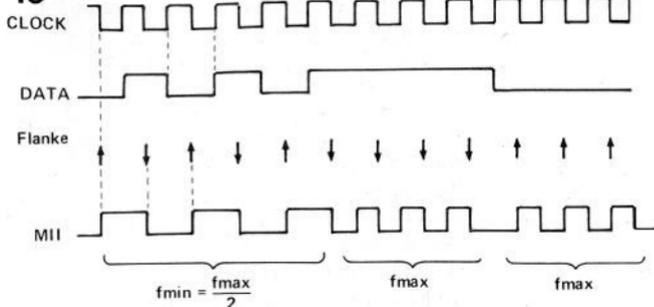


Bild 10. Der Manchester-Kode wird aus einer Information ("DATA") und einem Taktsignal ("CLOCK") gebildet. Die Information legt fest, ob eine negative oder eine positive Flanke erforderlich ist. Ändert sich die Information nicht, dann müssen zusätzlich Flanken "zwischen geschaltet" werden.

ebenfalls eine Flanke besitzen. Ist die Information in dem Augenblick "0", dann muß es eine positive Flanke sein, im anderen Fall eine negative. Solange die Information sich periodisch ändert, treten keine Probleme mit dieser Vorschrift auf. Das Manchester-Signal besitzt die halbe Takt-Frequenz (siehe Bild 10). Was geschieht jedoch, wenn die Information konstant bleibt? Nun – in diesem Fall können nicht ständig negative oder positive Flanken ausgesendet werden. Die Lösung des Problems: Auf halbem Wege zwischen positiven oder negativen Flanken werden zusätzliche (negative oder positive) Flanken hinzugefügt. Die Frequenz entspricht in diesem Fall der Taktfrequenz, sie ist also maximal.

Ausgehend von der zuvor erwähnten Vorschrift läßt sich ohne weiteres eine komplizierte Schaltung ausdenken, die die erforderlichen Flanken mittels logischer Gatter und Flipflops als Speicherbausteine erzeugt. Glücklicherweise ist solch eine komplizierte "Mimik" nicht notwendig. Ein einziges Exklusiv-NOR-Gatter genügt!

Bild 11 zeigt, warum: Aus der Wahrheitstabelle kann man ablesen, daß immer dann, wenn am Eingang A eine "1" liegt, das Signal B unverändert übertragen wird. Liegt dort jedoch eine "0" an, dann erscheint am Ausgang das invertierte Signal \bar{B} . Legt man nun das Taktsignal an Eingang B und die Daten an Eingang A, dann wird das Taktsignal anhängig von den Daten invertiert oder nicht. In Bild 10 kann man diesen Zusammenhang bereits erkennen: Die Information Null invertiert das Taktsignal. Bei

"1" ist es genau umgekehrt.

In der Technik verwendet man für das Verb "invertieren" häufig auch den Begriff "die Phase um 180° drehen". Das bedeutet hier: Die Phase des Taktsignals wird entweder um 180° (invertieren) oder um 0° (nicht invertieren) gedreht. Auf diese Weise kann man das Manchester-Signal auf eine zweite Art beschreiben: Der Takt wird durch die Daten in der Phase moduliert. Manchester II gehört also zur Gruppe der Codes, die eine Phasenmodulation beinhalten. Hier handelt es sich um einen Zwei-Phasen-Kode. Das heißt: Die Modulation erfolgt mit lediglich zwei Phasen, nämlich 0° und 180°.

Eine dritte Beschreibung ist sehr anschaulich: Sobald eine längere Periode auftritt, müssen sich die Daten beim Dekodieren ändern. Entweder in eine "1", wenn die längere Periode "1" ist, oder in eine "0", wenn diese Periode "0" ist.

Dekoder

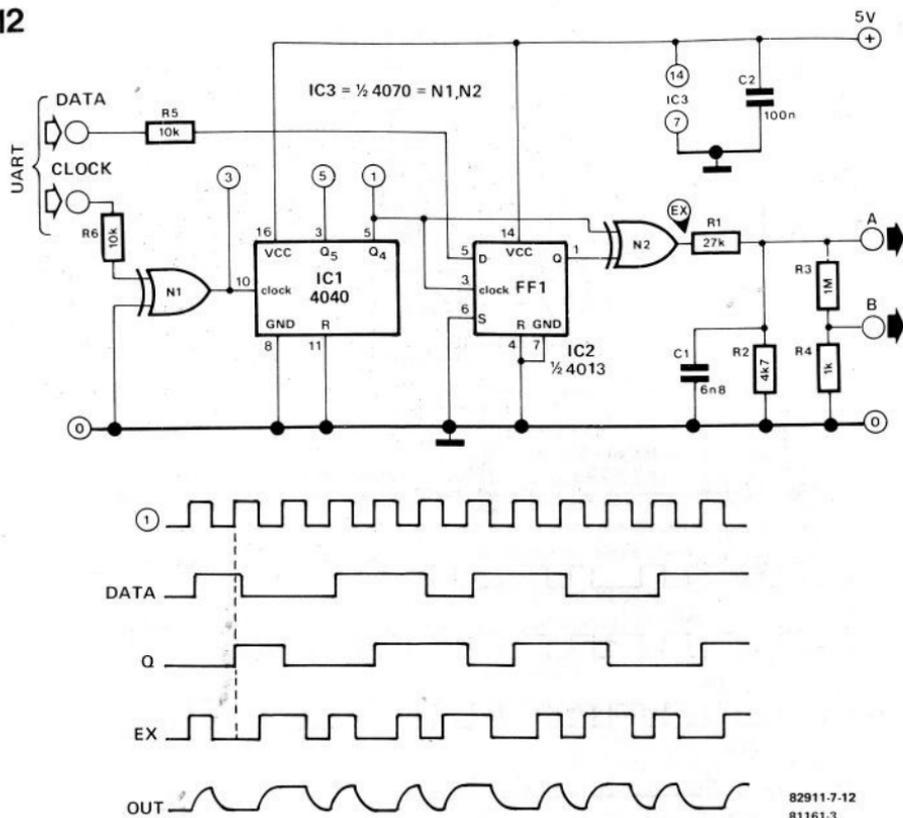
Ausgehend von der letztgenannten Erklärung ist es nur noch ein kleiner Schritt zur Entwicklung eines Dekoders. Der Encoder, der aus Daten und Taktsignal das Manchester-Signal bildet, wurde bereits erwähnt. Dabei handelt es sich einfach um ein Exklusiv-NOR-Gatter.

11



A	B	C	A = 1	A = 0
0	0	1		
0	1	0		
1	0	0	B/C	B/C
1	1	1	1	1
			0	0
			1	1

Bild 11. Die Wahrheitstabelle eines Exklusiv-NOR-Gatters. Dieses Gatter eignet sich als Encoder des Kassetten-Interface.



Der Dekoder ist leider etwas komplizierter.

Bei jeder längeren Periode ändert sich die Information. Diese längere Periode kann man leicht mit einem monostabilen Multivibrator (MMV) erfassen. Tritt eine solche Periode auf, dann gibt der MMV einen Impuls ab. Mit diesem Impuls wird der aktuelle Pegel des Manchester-II-Signals in einem Flipflop gespeichert. Am Ausgang des Flipflops liegt dann die zurückgewonnene Information. Eine längere Periode, die "1" war, zeigt in jedem Fall an, daß die Information ebenfalls "1" werden muß, und umgekehrt. So einfach ist das!

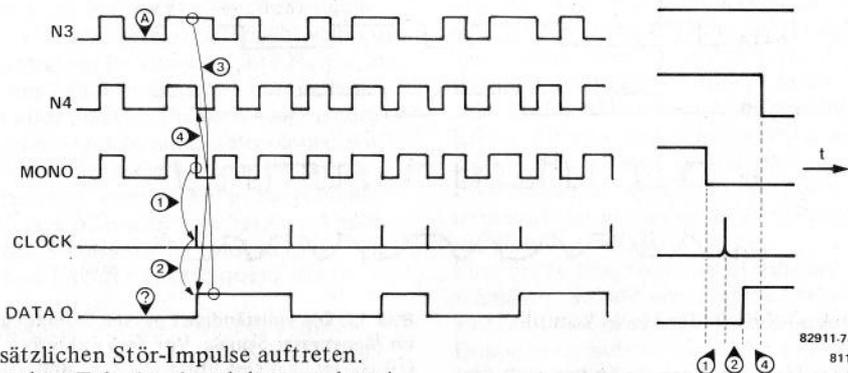
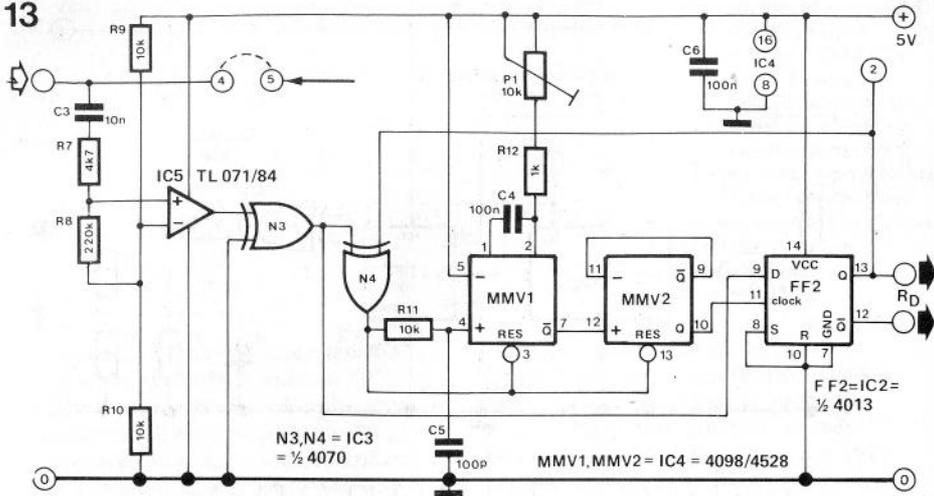
Die Schaltung

Die Beschreibung von Encoder und Dekoder kann kurz ausfallen, da die prinzipiellen Funktionsweisen bereits

Bild 12. Die vollständige Encoder-Schaltung für Manchester-Signale. Vor dem Exklusiv-OR-Gatter liegt eine Stufe, die Takt und Daten synchronisiert. Im Vergleich zu Bild 1 erscheint der Code am Ausgang dieser Schaltung invertiert.

erklärt wurden. Bild 12 zeigt die Schaltung des Encoders. Die Daten stammen normalerweise aus einem UART. In jedem Fall muß der Takt ein ganzes Vielfaches der Baudrate sein. Bei einem 16x-Takt beträgt die Taktfrequenz also $16 \times 4800 \text{ Hz} = 76,8 \text{ kHz}$, bei einem 64x-Takt ergibt sich $307,2 \text{ kHz}$. IC3 ist ein Binär-Teiler, der durch 16 teilt (oder durch 64, wenn Ausgang 2 anstelle von Ausgang 5 genommen wird). Flipflop FF1 sorgt dafür, daß sich die Information genau mit der positiven Flanke des geteilten Taktsignals ändert. Diese Maßnahme ist deshalb notwendig, damit im Manchester-II-Signal keine

82911-7-12
81161-3



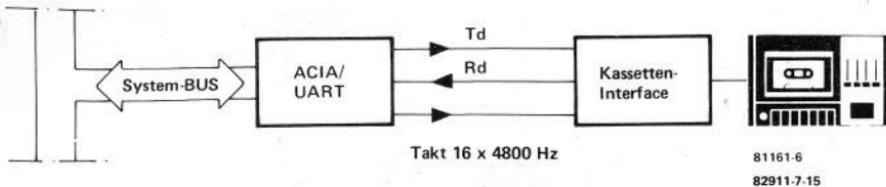
zusätzlichen Stör-Impulse auftreten. Aus dem Taktsignal und den synchronisierten Daten wird auf die bekannte Weise das Manchester-Signal erzeugt. Da hier anstelle eines Exklusiv-NOR- ein Exklusiv-OR-Gatter eingesetzt wurde, erscheint der Kode zwar invertiert am Ausgang. Das ändert aber nichts an der prinzipiellen Funktion.

Mittels R1 und C1 werden die Rechtecke etwas abgerundet. Der Encoder enthält zwei Ausgänge: Ausgang A wird vorzugsweise verwendet, Ausgang B liefert ein abgeschwächtes Signal, das man an den Mikrofon-Eingang eines Rekorders legen kann.

Bild 13 zeigt die Schaltung des Dekoders. Das Rekordersignal wird einem Komparator zugeführt, der daraus zunächst einmal eine rechteckförmige Spannung formt. N3 "möbelt" die Flanken noch

Bild 13. Der Dekoder für Manchester-Signale. Um Ursache und Wirkung deutlich zu machen, ist ein Teil des Impulsdiagramms vergrößert dargestellt. Die Pfeile geben die Abfolge der Signale an. Achtung: Beim Einschalten der Betriebsspannung kann die Information einmalig verlorengehen.

etwas auf. Das Manchester-Signal gelangt an den Daten-Eingang von Flipflop 2 und gleichzeitig an Monoflop 1, das die längeren Perioden "entdecken" muß. An dieser Stelle gibt es noch ein Problem, das im folgenden erläutert und durch den Einsatz von N4 gelöst wird. Nehmen wir einmal an, daß N4 nicht invertiert. Mit der ersten besten positiven Flanke startet Monoflop 1. Ist die Periode kurz, dann werden beide Monoflops zurückgesetzt, so daß das Eingangssignal



ohne Pausen. Die hier beschriebene Schaltung ist jedoch für asynchrone Übertragung ausgelegt und deshalb für synchrone Übertragung *nicht* ohne weiteres geeignet.

Kleine Unterschiede (einige Procente) in der Geschwindigkeit zwischen Aufnahme und Wiedergabe sind zulässig, da auf jedes Byte asynchroner Übertragung eine Pause folgt, die den UART wieder synchronisiert. Bei synchroner Übertragung muß man einen anderen Dekoder einsetzen (beispielsweise mit PLL), der das Taktsignal zurückgewinnt und auf diese Weise Geschwindigkeitsunterschiede

Bild 15. Auf diese Weise kann man das Kassetten-Interface in ein Mikroprozessor-System integrieren.

minimal hält. Das Blockscheema in Bild 15 zeigt die prinzipielle Verwendung des Kassetten-Interfaces in einem Mikroprozessor-System.

Im Prinzip eignet sich die Schaltung für beliebige Geschwindigkeiten, sowohl kleiner als auch größer als 4800 Baud. Möchte man also eine andere Übertragungsgeschwindigkeit einstellen, dann sind C1 und C4 entsprechend anzupassen.

Kapitel 6:

Diverses

Kapazitives Tastenfeld

Berühren erlaubt!

Tastenfelder, Keyboards, oder wie sie sonst noch heißen mögen, sind im Zeitalter der "Computerisierung" auch aus der Hobby-Elektronik nicht mehr wegzudenken. Möchte man seinen Personal-Computer selbst bauen, dann wird auch irgendwann solch ein Tastenfeld benötigt. Die Frage stellt sich: selbst bauen oder kaufen? Nun – der gestandene Hobby-Elektroniker wird nicht lange überlegen und zum LötKolben greifen. Jedoch: Der Preis für Tasten, beispielsweise für die bekannten Digitaster, ist nicht ohne. Also doch was Fertiges kaufen?

Allen Unentschlossenen bietet Elektor eine preiswerte Alternative an: Das kapazitive Tastenfeld ohne bewegliche Teile, das trotzdem oder gerade deshalb zuverlässig funktioniert. Ein gutes Beispiel für die einfache Lösung eines komplizierten elektro "mechanischen" Problems auf elektronische Weise.

Gerade bei kleinen Mikroprozessor-Systemen benötigt man in den meisten Fällen ein Tastenfeld mit etwa 10 bis 20 Tasten. Ein Preis von etwa 2,30 DM für einen Digitaster schlägt also ganz schön zu Buche. Außerdem unterliegen

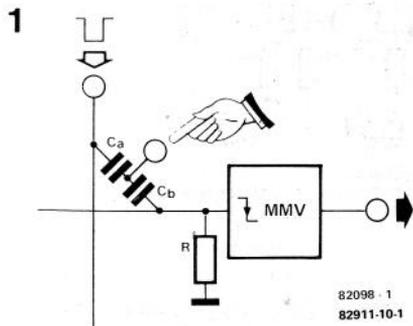


Bild 1. Diese Zeichnung macht die Funktionsweise eines kapazitiven Tastenfelds deutlich.

die Tasten einem gewissen mechanischen Verschleiß. Aus diesen Gründen hat man sich Gedanken über die Realisierung anderer Taster-Prinzipien gemacht. So gibt es beispielsweise Tastenfelder aus leitendem Gummi, mit Hall-Elementen und nicht zuletzt mit Sensor-Tasten. Zu der letzten Kategorie zählt auch das kapazitive Tastenfeld. Die Tasten bestehen aus metallischen Flächen, die als Kondensatoren wirken. Eine Berührung mit dem Finger löst einen Schaltvorgang aus. Der gesamte Aufbau des kapazitiven Tastenfelds ist denkbar unkompliziert und nicht aufwendig. Mit Hilfe eines kleinen Programms kann man Kontaktentprellung und Dekodierung realisieren.

Die Funktion

Die Funktion des kapazitiven Tasten-

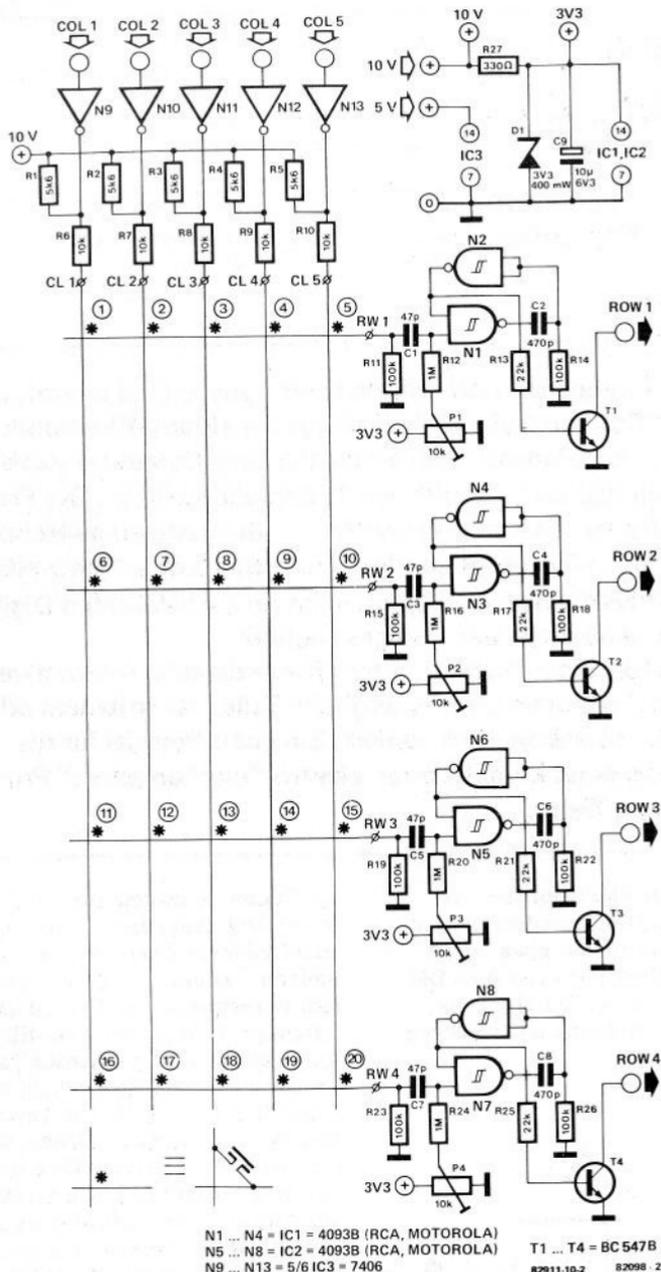


Bild 2. Die Schaltung eines kapazitiven Tastenfelds mit 20 Sensortasten. Der Aufwand an Elektronik ist gering, da das Tastenfeld von einem Mikroprozessor gesteuert wird.

felds läßt sich am besten an Hand des Bildes 1 erläutern. Zwischen zwei Anschlußleitungen liegen zwei Kondensatoren in Reihe. Der Knotenpunkt bildet

gleichzeitig den Sensorkontakt. Einer der beiden Kondensatoren (C_b) ist an den Eingang eines monostabilen Multivibrators angeschlossen, während dem anderen Kondensator (C_a) eine Impulsfolge zugeführt wird. Bleibt der Sensorkontakt unberührt, dann gelangt der Impuls über das differenzierende Netzwerk C_a , C_b , R an den Eingang des Monoflops. Das Monoflop reagiert auf die erste Flanke des differenzierten Impulses und gibt daraufhin einen Impuls bestimmter Länge ab. Der Triggerpegel des MMV ist so gewählt, daß der über C_a und C_b eintreffende Impuls gerade groß genug ist, um den MMV zu triggern. Berührt man den Sensorkontakt, dann wird praktisch ein Kondensator und/oder ein Widerstand gegen Erde geschaltet. Dadurch sinkt der Impuls am Eingang des MMV unter die Triggerschwelle. Der monostabile Multivibrator liefert keinen Ausgangsimpuls mehr. Auf diese Weise kann man ein vollständiges Tastenfeld aufbauen. Ein kurzes Programm sorgt dafür, daß die Impulse erzeugt werden, und stellt gleichzeitig fest, wann ein Impuls "ausfällt". Ein solcherart aufgebautes Tastenfeld ist nicht nur preiswert, sondern auch praktisch verschleißfrei.

Die Schaltung

Bild 2 zeigt die Schaltung eines kapazitiven Tastenfelds mit 20 Sensortasten. Sie sind in einer Matrix aus vier Reihen und fünf Spalten angeordnet. Jeder Kreuzungspunkt einer Reihe mit einer Spalte ist als kapazitive Taste ausgeführt. Die Spalten werden über die Inverter $N_9 \dots N_{13}$ mit Impulsen "gefüttert". Die Inverter sind vom Open-Collector-Typ. Man erhält also durch Anlegen der Pull-Up-Widerstände $R_1 \dots R_5$ eine Signalspannung von 10 V an den Inverter-Ausgängen. Die Impulse für die Reihen-Steuerung werden vom Mikroprozessor geliefert. Die Reihen müssen selbstverständlich nacheinander angesteuert werden. An jede Reihe ist ein monostabiler Multivibrator angeschlossen, der aus

zwei Schmitt-Trigger, einem Kondensator und einem Widerstand besteht. Die hier verwendeten CMOS-Typen haben eine besonders große Eingangsimpedanz, die man gegenüber dem am Eingang liegenden Widerstand vernachlässigen kann. Die Hysterese der Schmitt-Trigger wird durch die Verwendung einer kleinen Betriebsspannung (3,3 V) auf weniger als 400 mV reduziert. Das ist deshalb notwendig, weil die über die kapazitiven Tasten geführten Impulse nicht sondernlich groß sind. Die Impulsdauer des MMV ergibt sich für N_1 und N_2 zu $R_{14} \cdot C_2$ und für N_3 und N_4 zu $R_{18} \cdot C_4$ und so weiter. Die an den Eingängen liegenden Hochpaßfilter C_1/R_{12} , C_3/R_{16} und so weiter sieben Brumm und andere elektrische "Verunreinigungen" aus. Als Puffer/Inverter liegt am Ausgang jedes MMV ein Transistor mit offenem Kollektor. Die Potentiometer an den Eingängen der MMV gestatten die Einstellung eines Gleichspannungspegels. Dieser Pegel wird so gewählt, daß der eintreffende Impuls (von einer Spalte) gerade unterhalb der niedrigen Triggerschwelle U_T zu liegen kommt. Auf diese Weise erreicht man, daß der MMV einen Impuls abgibt, wenn die Sensortasten *nicht* berührt werden. "Positive" Impulse an den Spalten werden von $N_9 \dots N_{13}$ invertiert. Die MMVs reagieren also auf die erste abfallende Flanke eines Impulses an "CL1 ... CL5". In Bild 3 sind zur Veranschaulichung einige der vorkommenden Signalformen eingezeichnet. Während der ersten beiden Impulse wird der Sensorkontakt nicht berührt. Die nächsten drei Impulse sind kleiner, der Kontakt wird also berührt. In diesem Bild kann man auch deutlich erkennen, wohin der Gleichspannungspegel $U=$ gelegt werden muß.

Der Abgleich

Abgleich ist vielleicht etwas zu viel gesagt. Denn es geht nur um die Einstellung der Trimpotis. Zunächst stellt man alle Potis so ein, daß ihre Schleifer an +3,3 V liegen. Nun wird eine Rechteckspannung an die erste Spalte gelegt. Man dreht P_1 langsam so weit zurück, bis

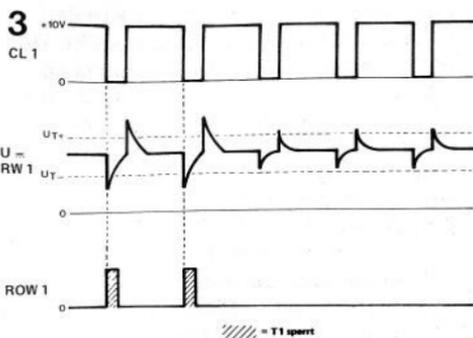


Bild 3. Ein Beispiel für die vorkommenden Signalformen, wenn Impulse auf Spalte 1 liegen. Während der ersten 2 Impulse ist die Taste 1 nicht berührt, während der folgenden 3 Impulse ist die Taste berührt.

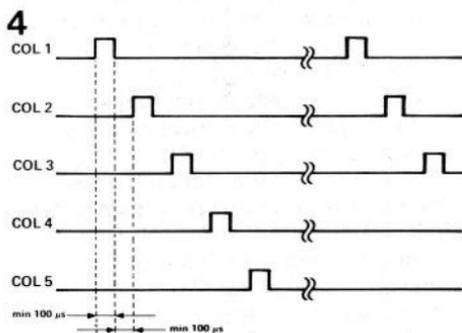


Bild 4. Auf diese Weise muß die Steuerung der unterschiedlichen Spalten geschehen.

der MMV auf die negative Flanke des Rechteckimpulses (also die positive Flanke am Eingang von N1) reagiert. Wird nun eine Taste berührt, dann darf der MMV nicht mehr triggern. Das Trimpoti dreht man noch etwas weiter, so daß die Schaltung gerade auf eine Tastenberührung reagiert. Dieser Abgleich wird für jede Reihe getrennt durchgeführt.

Ansteuerung

Bild 4 zeigt, wie die Impulse zur Ansteuerung aussehen müssen. Man kann sie direkt vom angeschlossenen Mikroprozessor-System ausgeben lassen. Nach der positiven Flanke jedes Impulses kann der Mikroprozessor "nachsehen", ob eine der Reihen-Ausgänge innerhalb der MMV-Zeit "0" ist. In diesem Fall wurde die Taste zwischen der Spalte,

auf der der ausgesendete Impuls liegt, und der Reihe, auf der kein Impuls "entdeckt" wurde, berührt.

Die MMV-Zeit beträgt in der angegebenen Dimensionierung etwa 50 μ s. Man kann sie jedoch einfach dadurch ändern, daß der 470-p-Kondensator ausgetauscht wird.

Für eine gute Funktion ist eine Kontaktentprellung unerlässlich. Auch sie kann der Mikroprozessor übernehmen. Bild 5 zeigt das Flußdiagramm für eine mögliche Programmierung. Wird keine Taste berührt, dann läuft das Programm in Schleife B ab. In dieser Schleife "wartet" der Prozessor zunächst 10 ms, besser gesagt: er macht in dieser Zeit etwas anderes, steuert beispielsweise das Display. Daran anschließend wird ein Impuls auf Spalte 1 gegeben, der Prozessor fragt die Reihen ab, gibt danach einen Impuls auf Spalte 2 und so weiter. Sind alle 5 Spalten abgefragt, und wurde nirgendwo eine "0" (kein Impuls) entdeckt, dann "weiß" der Prozessor, daß keine Taste berührt wurde. Das Programm läuft weiter in Schleife B.

Wurde allerdings eine "1" entdeckt, dann verläßt der Prozessor die Schleife und wartet zunächst wieder 10 ms lang. Anschließend überprüft der Prozessor, ob die Taste immer noch berührt wird. Ist das der Fall, dann verläßt der Prozessor dieses Programm (Pfeil unten) und springt in ein anderes, das die Information "Taste immer noch gedrückt" weiter verarbeitet. Danach geht's über Label KEY zurück ins Programm in Bild 5. Wird die Taste auch weiterhin berührt, dann bleibt der Prozessor in Schleife A. So lange bis die Taste losgelassen wird. Nun wartet der Prozessor wieder in Schleife B, bis eine Taste berührt wird. Auf diese Weise erreicht man eine Kontaktentprellung von minimal 10 ms. In der Praxis erwies sich dieser Wert als ausreichend. Man benötigt insgesamt 9 I/O-Leitungen des Mikroprozessors: 5 Ausgänge für die Spalten und vier Eingänge für die Reihen.

Die Tasten

Das Tastenfeld wird am besten aus

5

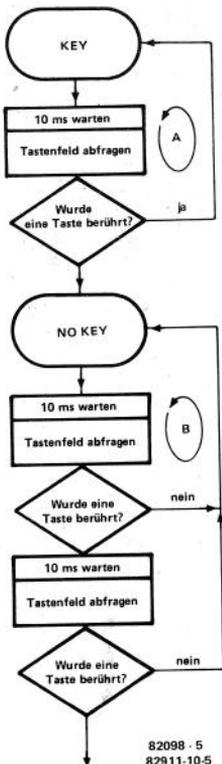


Bild 5. Das Flußdiagramm gibt einen Hinweis, wie ein Programm zur Tasten-Abfrage und zur Kontaktentprellung aussehen kann.

einem Stückchen doppelseitig kaschierter Platine hergestellt. Auf die eine Seite zeichnet man 4 mal 5 Flächen mit den Abmessungen 1,5 cm x 1,5 cm mit jeweils 5 mm Zwischenraum (siehe auch Bild 6). Auf die entgegengesetzte Seite wird das gleiche Muster aufgezeichnet mit dem Unterschied, daß jede Fläche durch einen schmalen Streifen in zwei Teile geteilt wird. Außerdem erhält eine der beiden so entstandenen Flächen eine Lötinsel. Die anderen Flächen werden durch einen schmalen Kupferstreifen miteinander verbunden.

Bei der Platinen-Herstellung muß man sehr sorgfältig arbeiten, weil die richtige Funktion von der Kapazität der Tasten abhängt. Da die Kapazität aller Tasten genau gleich sein soll, müssen auch die geometrischen Abmessungen möglichst genau übereinstimmen. Das bedeutet

6

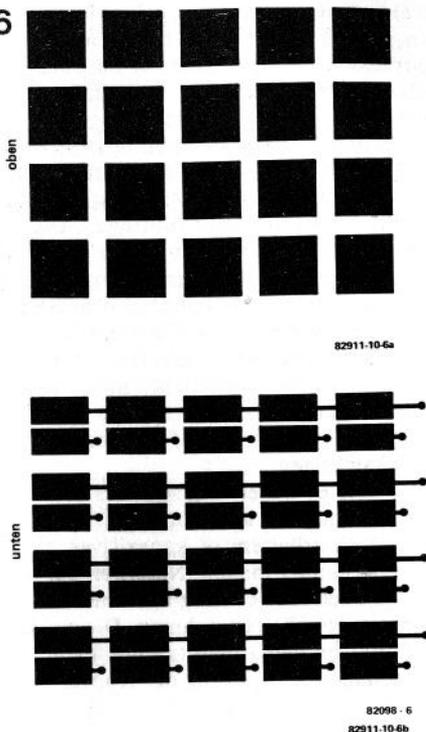


Bild 6. Der Aufbau des Tastenfelds. Beim Layout der doppelseitigen Platine muß man sehr genau vorgehen, da alle Kontaktflächen-Kapazitäten genau gleich groß sein müssen.

auch, daß die Kupferflächen auf beiden Seiten sich genau gegenüber liegen müssen.

Nach dem Ätzen der Platine, die selbstverständlich auch die Elektronik enthalten kann, werden die Lötinseln pro Spalte miteinander verbunden (siehe Bild 7). Auch hier muß man genau arbeiten. Am besten wird dünner Kupferlackdraht (beispielsweise von 0,2 mm Durchmesser) verwendet, dessen Isolierlack an den Lötstellen mit einem heißen LötKolben und Lötzinn abgebrannt wird. Die Verbindungsleitungen müssen sehr straff gespannt sein! Über die Sensorflächen auf der Platinen-Oberseite kann man eine durchsichtige Klebefolie spannen (je dünner, desto besser!). Auf diese Weise werden die Flächen nicht beschädigt, die Kapazität bleibt auf Dauer erhalten. Man kann die Kupfer-

flächen mit Abreibebuchstaben beschriften, so daß immer die richtige Taste gedrückt wird. Die Schaltung darf auf jeden Fall erst nach dem Auf- und/oder Einbau abgeglichen werden.

Um Störeinflüsse von außen fernzuhalten, kann man zusätzlich ein Abschirmblech genau parallel in etwa 2 cm Abstand von der Platinen-Unterseite anbringen. Die Ausrichtung muß deshalb genau parallel sein, weil sonst einige Tasten empfindlicher als andere sein könnten. Das Blech verbindet man mit Masse. Schließlich noch zwei praktische Tips: Die Reihe- und Spalten-Anschlußdrähte müssen so kurz wie möglich sein. Auch das Platinen-Layout für die MMV muß "kurze Wege" berücksichtigen.

Mit dieser Beschreibung sollte es ohne weiteres möglich sein, ein gutes und zuverlässig arbeitendes, kapazitives Tastenfeld aufzubauen. Natürlich kann man je nach Anwendung auch mehr oder weniger Tasten nehmen. Die richtige Funktion hängt hauptsächlich von der genauen Ausführung der Kupfer-

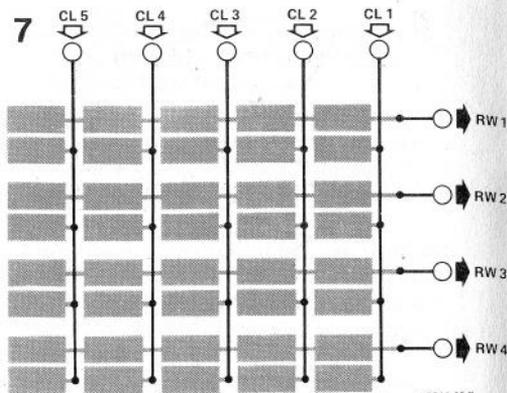
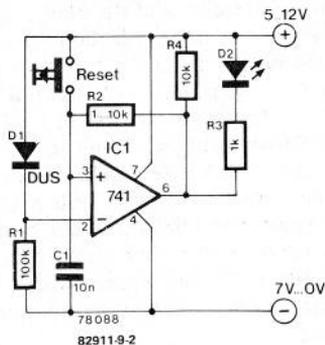


Bild 7. Alle Flächen einer Spalte werden untereinander mittels dünnem Kupferdraht verbunden. Die Reihen-Flächen sind untereinander durch Kupferbahnen verbunden.

flächen ab. Man kann Fehlfunktionen unter Umständen durch eine andere Einstellung der Trimpotis oder Änderung der Tastenabmessungen korrigieren. Im Elektor-Labor leistet die Schaltung jedenfalls schon geraume Zeit gute Dienste.

Spannungsausfall-Indikator

Für eine Reihe elektronischer Systeme ist das kurzzeitige Ausfallen der Speisenspannung von unangenehmer Wirkung. Insbesondere gilt dies für die Schreib-Lese-Speicher (RAM) in Mikrocomputern, die auch durch einen kurzen Störimpuls auf der Netzspannung völlig "in Verwirrung" geraten können. Lassen sich solche Störungen nicht absolut unterbinden, so kommt man ohne Spannungsausfall- und Störimpuls-Indikator kaum aus. Für diesen Zweck leistet die nebenstehende Schaltung gute Dienste. Sie zeigt durch Aufleuchten einer LED an, daß ein kurzzeitiger Ausfall der Speisenspannung eingetreten ist oder daß der Speisespannung ein Störimpuls überlagert war.



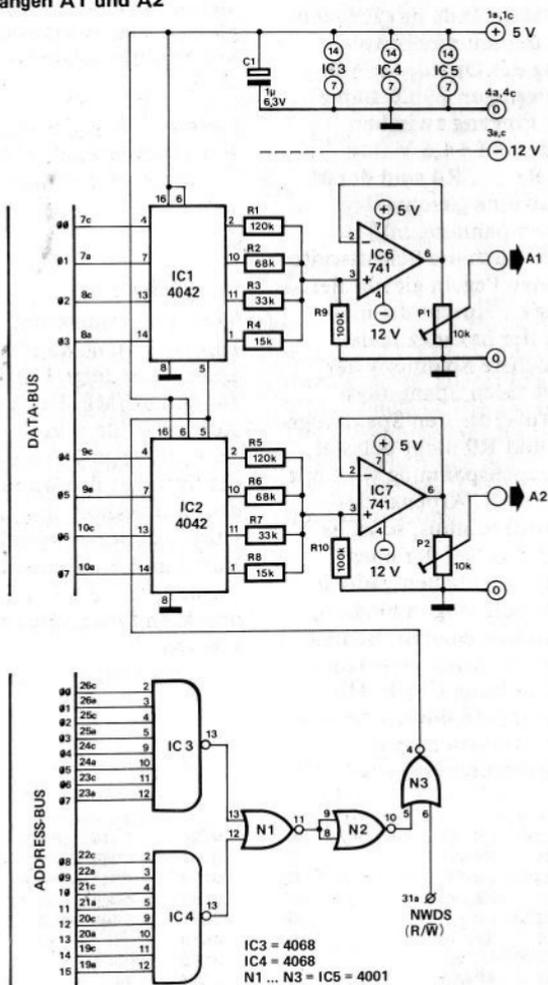
Digital-Analog-Wandler

Mit einigen billigen und leicht erhältlichen CMOS-ICs ist es möglich, einen einfachen Digital-Analog-Wandler aufzubauen. In Verbindung mit einem Mikrocomputer kann mit Hilfe von Software ein digitales Signal in ein analoges umgesetzt werden.

Bild 1. Vollständige Schaltung des Digital-Analog-Wandlers. Die von den Ausgängen der ICs 1 und 2 abgegebenen Spannungen werden mit dem Widerstandsnetzwerk R1 ... R4 (bzw. R5 ... R8) addiert und über die Puffer IC6 und IC7 den Ausgängen A1 und A2 zugeführt.

Da Mikrocomputersysteme auf großes Interesse stoßen und Erweiterungen immer wieder gewünscht werden, erfolgt hier die Vorstellung einer für viele Zwecke

1



nützlichen Ergänzung: ein Digital-Analog-Wandler.

Bild 1 zeigt die Schaltung des Wandlers. Den eigentlichen Wandlerteil bilden die mit den Speicherausgängen von IC1 (Q0 ... Q3) verbundenen Widerstände R1 ... R4. Die Eingänge des Speichers (D0 ... D3) sind mit dem Databus des μ P-Systems verbunden, über den die Informationen des Prozessors in den Speicher gelesen werden. Anhängig vom Bitmuster an den Eingängen liegen an den Ausgängen jeweils verschiedene Widerstände (R1 ... R4) an logisch "1" oder logisch "0". Bei CMOS-ICs und 5 V Versorgungsspannung bedeuten diese Zustände +5 V bzw. 0 V. Zusammen bilden nun die Widerstände mit R9 einen Spannungsteiler, der am nichtinvertierenden Eingang des Opamps IC6 liegt. Abhängig vom jeweiligen Zählerstand befindet sich der Eingang zwischen 0 V (log. "0") oder auf +4,6 V (log. "1"). Die Widerstände R1 ... R4 sind derart dimensioniert, daß eine gleichmäßig abgestufte Treppenspannung mit 16 Schritten entsteht; auch die Schrittweite ist bei den einzelnen Pegeln gleich. Bei dieser Anordnung entspricht die niedrigste Spannung der hexadezimalen Zahl 0 und die höchste Spannung der Zahl F. IC6 bildet einen Spannungsfollower als Pufferstufe, die den Spannungsteiler R1 ... R4 und R9 nicht belastet. Die Höhe der Treppenspannung wird mit dem Poti P1 eingestellt. Arbeitet der μ P mit einem 8-bit-Databus, so ist es erforderlich, zwei Wandler der eben beschriebenen Art aufzubauen, jedoch wird die Adreßdekodierung in beiden Fällen (für 1 Byte) beibehalten. Bedingt durch die Art des Adreßdekoders lautet die Adresse der Schaltung FFFF. Mit einer Anpassung des Adreßdekoders sind auch andere Adressen möglich. Ein in IC5 nicht gebrauchtes Gatter

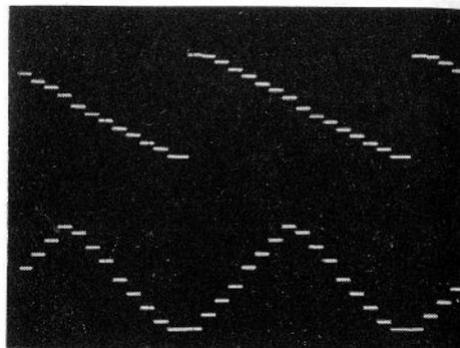


Bild 2. Hier sind die mit dem in Tabelle 1 angegebenen Programm erzeugten Signalformen dargestellt. Der Ausgang A1 liefert eine in 16 Stufen abfallende Spannung, A2 hingegen eine in 8 Schritten ansteigende und in 8 Schritten abfallende Treppenspannung.

kann zur Invertierung eines Adreßbits Verwendung finden, so daß prinzipiell 17 verschiedene Adressen zur Verfügung stehen.

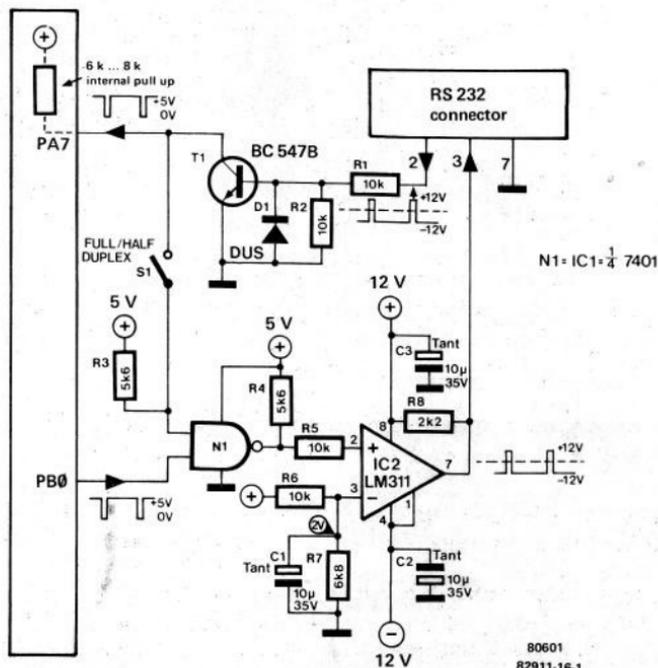
Programm

Viele Programme kommen für eine Digital-Analog-Wandlung in Frage; beispielsweise zeigt Tabelle 1 ein solches für den SC/MP. Beginnend mit der Adresse 0C00 schreibt das Programm eine 16 Byte lange Tabelle Byte für Byte in die Speicher des Wandlers. Dieser Vorgang wiederholt sich laufend, so daß am Ausgang ein periodisches Signal erscheint. Bild 2 gibt ein Beispiel für die Signalformen, die mit dem in Tabelle 1 abgedruckten Programm erzeugt werden können.

Tabelle 1

0C00	C4FF	LDI FF	TAB 0F00	0F	0C0C	C410	LDI10	D7
0C02	31	XPAL1		2E	0C0E	C80B	ST COUNT	86
0C03	C4FF	LDIFF		4D	0C10	\$2 C601	LD @1 (1)	95
0C05	35	XPAH1		6C	0C12	C900	ST00 (2)	74
0C06	C40F	LDI0F		8B	0C14	B805	DLD COUNT	53
0C08	36	XPAH2		AA	0C16	98F1	JZ\$1	32
0C09	\$1 C400	LDI00		C9	0C18	90F6	JMP \$2	11
0C0B	32	XPAL2		E8	0C1A	00	COUNT	00

RS-232-Interface



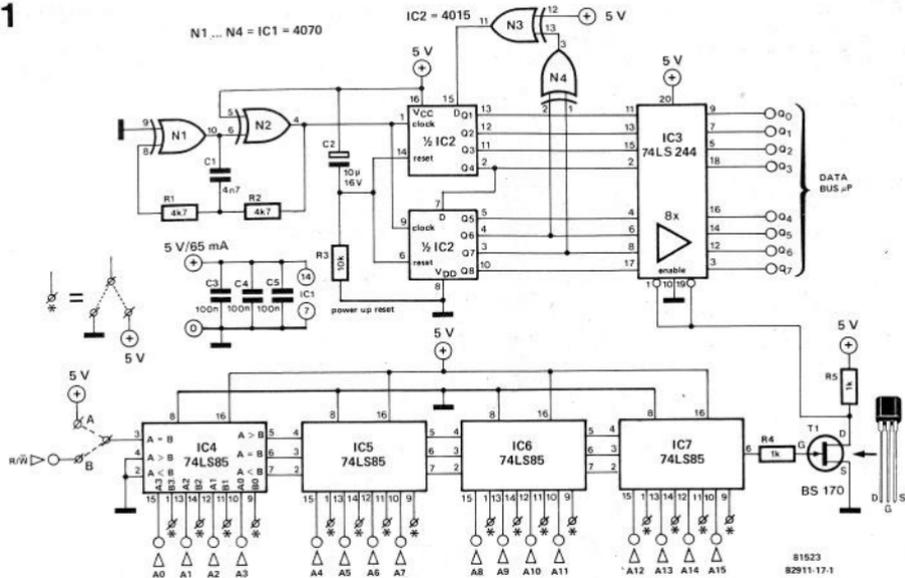
Computersysteme bestehen bekanntlich aus der Zentraleinheit und der Peripherie. Um beides problemlos miteinander koppeln zu können, müssen die Signalpegel an den Schnittstellen übereinstimmen. Wenn dies nicht der Fall ist, wird ein sogenanntes Interface benötigt. Als Norm für serielle Schnittstellen hat

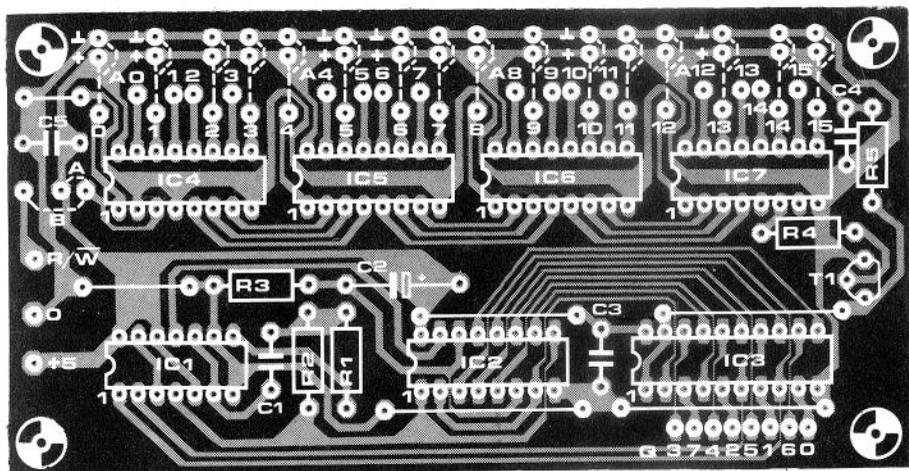
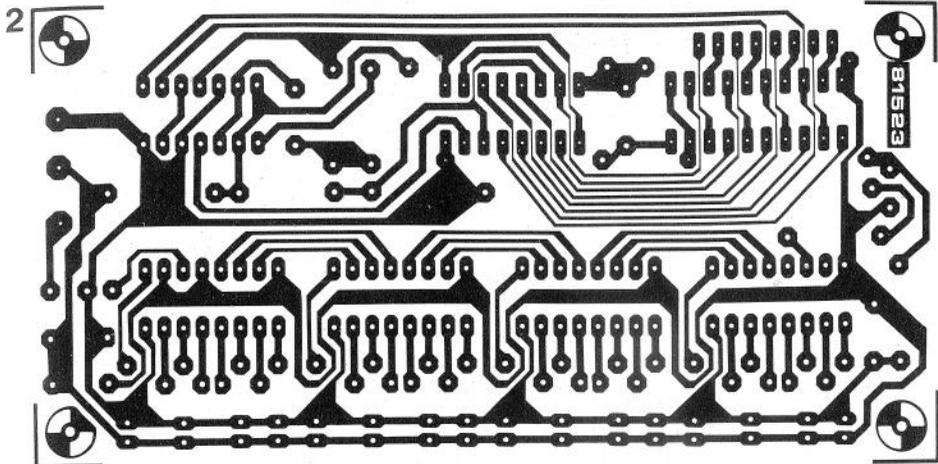
sich die RS-232-/V-24-Norm weltweit durchgesetzt. Dieses Interface paßt TTL-Signale an die genannte Norm an. Die Anschlußbezeichnungen PA7 und PB0 beziehen sich auf den Elektor-Junior-Computer; das Interface ist aber auch für andere Computer verwendbar.

Einfacher Zufallsgenerator

Für viele Computerspiele benötigt man einen Zufallsgenerator. Mit Hilfe einer kleinen Subroutine im Programm kann man wohl eine Pseudo-Zufallszahl erzeugen, aber ein "echter" Zufallsgenerator ist das doch nicht, wie sich spätestens nach einigen Spielen zeigt. Besser ist es, einen getrennten Zufallsgenerator zu verwenden, dessen Taktfrequenz von

der des Mikrocomputers unabhängig ist. Wenn das erzeugte Bitmuster auch noch eine ausreichend lange Zykluszeit aufweist, kann man wohl davon ausgehen, daß der Zufall ins Spiel kommt. Der einfache Zufallsgenerator, um den es hier geht, enthält ein Acht-bit-Schieberegister IC2 (Bild 1). Die Ausgänge Q6 und Q7 sind über die EXOR-Gatter N3 und N4





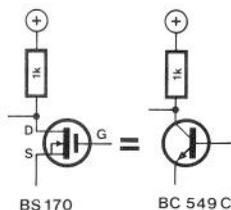
wird der R/\bar{W} -Eingang mit Pin 3/IC4 verbunden, bei einem negativen Strobe ist ein Inverter dazwischen zu schalten. Die Stromversorgung der Schaltung erfolgt mit 5 V bei einem Strombedarf 65 mA, was in vielen Fällen eine Versorgung durch das Netzteil des Mikrocomputers zuläßt. Auf der Platine (Bild 2) ist Platz für vier Adreßdekoder, es können auch weniger sein, wenn unvollständige Adreßdekodierung ausreicht. Anstelle der dann nicht benötigten ICs sind Drahtbrücken einzulöten, die die Anschlüsse 4 und 5, 3 und 6 sowie 2 und 7 miteinander verbinden.

Frage:

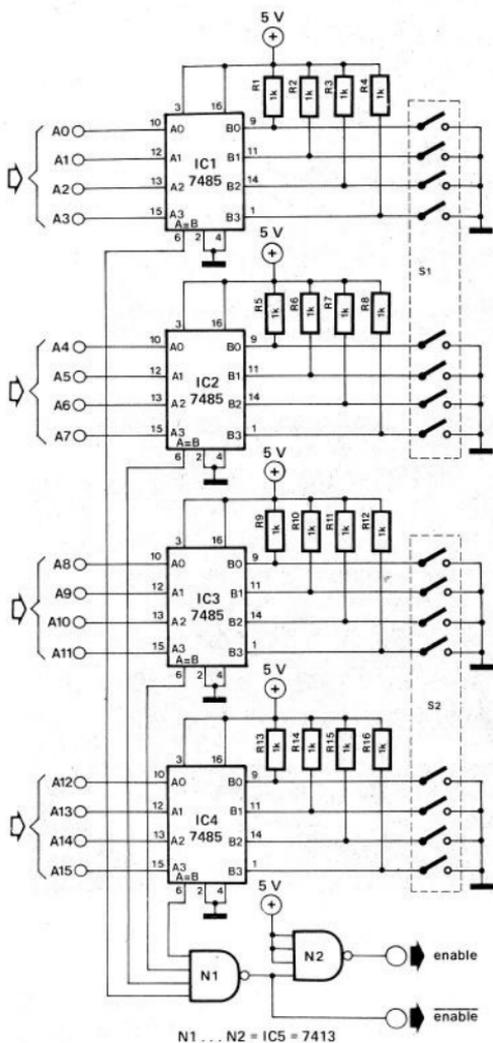
Was tun, wenn es bei den VMOS-Transistoren BS 170 wider Erwarten Beschaffungsschwierigkeiten gibt?

Antwort: Stattdessen einen bipolaren Transistor einsetzen, wie es Bild 3 zeigt.

3



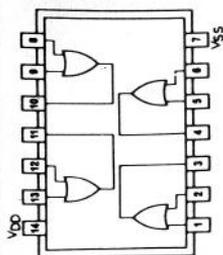
Programmierbarer Adreßdekode



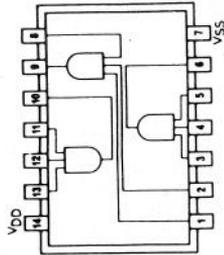
78056
82911-9-1

Diese Schaltung leistet gute Dienste beim Experimentieren mit Mikrocomputern. Aus dem Adreßbereich eines 64 Byte-Speichers kann ein bestimmtes Byte ausgewählt und die Adresse des gewünschten Bytes mit den Schaltern S1 und S2 programmiert werden. Sobald die gewünschte Adresse auf dem Adreßbus erscheint, spricht die Schaltung an. Die Möglichkeit, das Ausgangssignal in invertierter Form zu erhalten, ist ebenfalls vorgesehen. Man kann die Schaltung auch verwenden, um Seiten eines bestimmten Formats zu adressieren. Läßt man beispielsweise IC1 und IC2 weg, dann können Bereiche von 1/4 k Byte dekodiert werden.

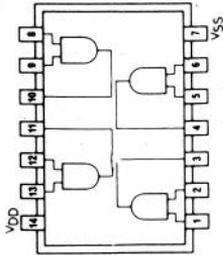
Die Schaltung ist nur dann einsetzbar, wenn das System über einen gepufferten Adreßbus verfügt. Die Ursache liegt am hohen "fan in" von IC ... IC4.



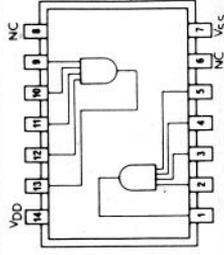
4071



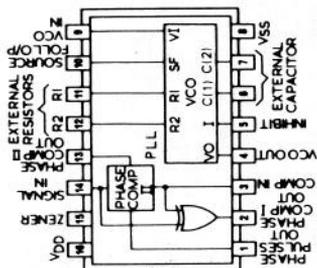
4073



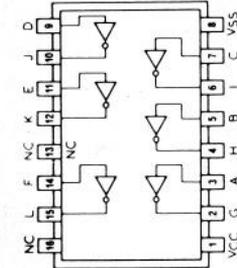
4081



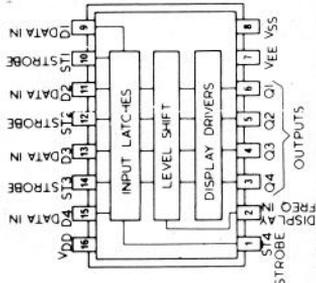
4082



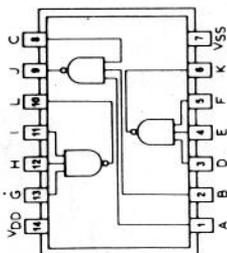
4046



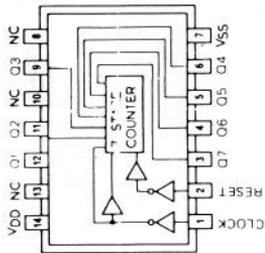
4049



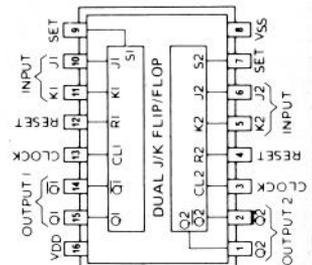
4054



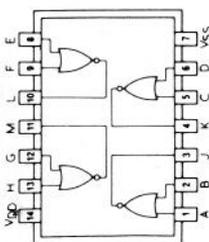
4023



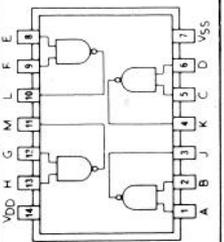
4024



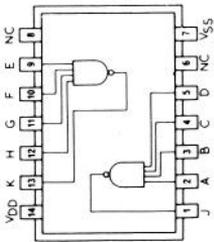
4027



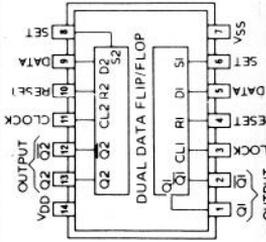
4001



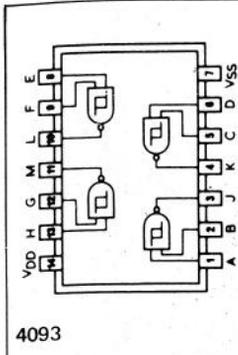
4011



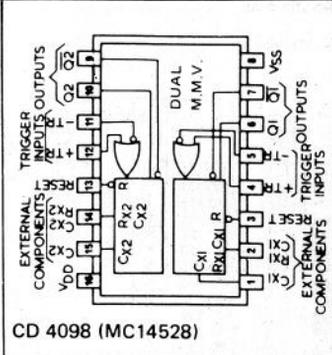
4012



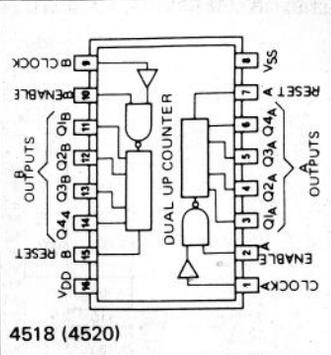
4013



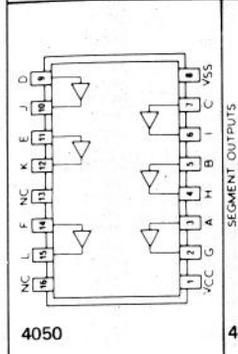
4093



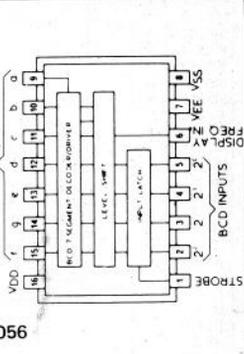
CD 4098 (MC14528)



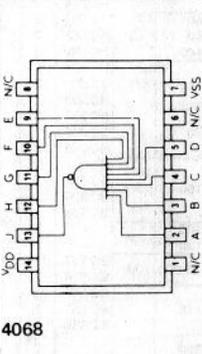
4518 (4520)



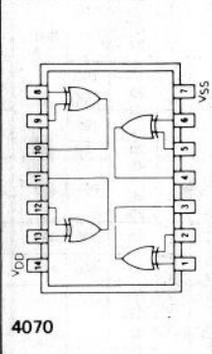
4050



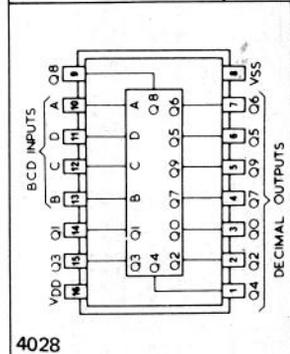
4056



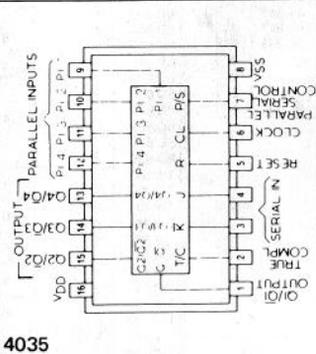
4068



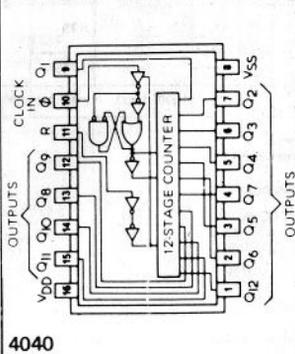
4070



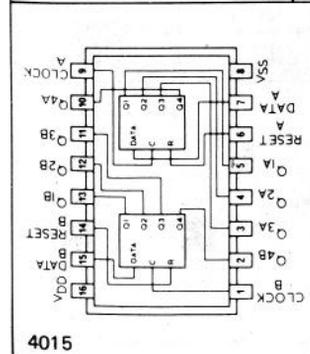
4028



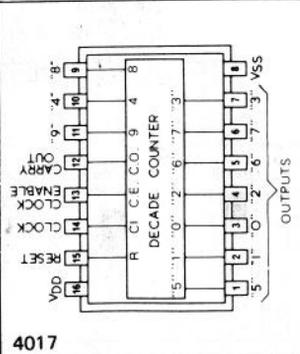
4035



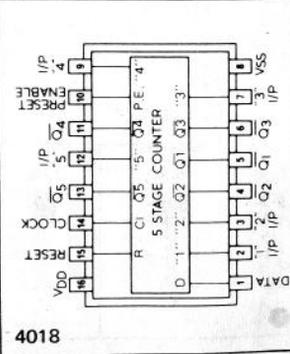
4040



4015



4017



4018

Achtung: Die Bezeichnung vor der Typennummer bezieht sich auf den Hersteller, z.B. CD 4001 (RCA), MC 14001 (Signetics), SCL 4001 (Solid State Scientific), SIL 4001 (Siitek).

TRANSISTOR-ÜBERSICHT: NF- und Universaltypen

Typ	PNP NPN	UCEO max (V)	Ic max (mA)	Pmax (mW)	bei hFE/Ic (mA)		Komple- mentär- zu:	Bild				
BC 107	N	45	100	300	> 110	2	BC 177	1				
BC 108	N	20					BC 178	1				
BC 109	N	40	3700	100	> 40	100	BC 179	1				
BC 140	N	20					BC 160	1				
BC 141	N	60					BC 161	1				
BC 160	P	40					BC 140	1				
BC 161	P	60					BC 141	1				
BC 177	P	45					BC 107	1				
BC 178	P	25					BC 108	1				
BC 179	P	20					BC 109	1				
BC 182	N	50					> 70	300	> 100	2	BC 212	2
BC 183	N	30					> 100				BC 213	2
BC 184	N	50	> 100	BC 214	2							
BC 212	P	50	> 60	BC 182	2							
BC 213	P	30	> 80	BC 183	2							
BC 214	P	30	> 140	BC 184	2							
BC 237	N	45	500	800	> 100	100	BC 307				2	
BC 238	N	20					BC 308				2	
BC 239	N	20					BC 309				2	
BC 307	P	45					BC 237				2	
BC 308	P	25					BC 238	2				
BC 309	P	20					BC 239	2				
BC 327	P	45					BC 337	2				
BC 328	P	25					BC 338	2				
BC 337	N	45					BC 327	2				
BC 338	N	25					BC 328	2				
BC 414	N	50	100	300	> 100	2	—	2				
BC 416	P	50	100	300	> 120	2	—	2				
BC 516	P	30	400	625	> 30.000	20	BC 517	1)				
BC 517	N	30	400	625	> 30.000	20	BC 516	1)				
BC 546	N	65	100	500	> 110	2	BC 556	2				
BC 547	N	45					BC 557	2				
BC 548	N	30					BC 557	2				
BC 549	N	30					BC 558	2				
BC 550	N	45					—	2				
BC 556	P	65					BC 546	2				
BC 557	P	45					BC 547	2				
BC 558	P	30					BC 548	2				
BC 559	P	30					—	2				
BC 560	P	45					—	2				
BC 639	N	80	1000	1000	> 40	150	BC 640	3				
BC 640	P	80	1000	1000	> 40	150	BC 639	3				

Anmerkungen:

1) Darlington

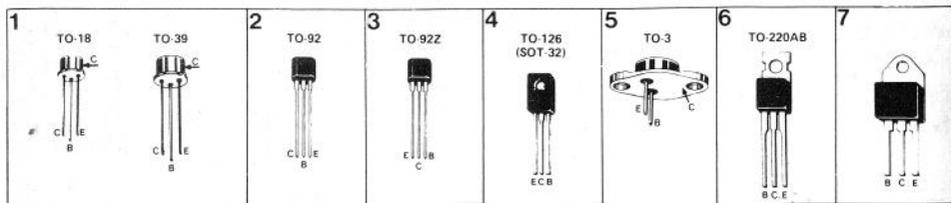
2) UCEO max.:

... A = 60 V

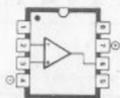
... B = 80 V

... C = 100 V

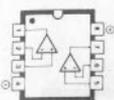
Typ	PNP NPN	UCEO max (V)	Ic max (A)	Pmax (W)	hFE/ bei Ic		Komple- mentär- zu:	Bild					
BD 131	N	45	3	15	> 40	0,15 A	BD 132	4					
BD 132	P						BD 131	4					
BD 135	N						BD 136	4					
BD 136	P						BD 135	4					
BD 137	N	60	1	8	> 40	0,15 A	BD 138	4					
BD 138	P						BD 137	4					
BD 139	N						BD 140	4					
BD 140	P						BD 139	4					
BD 169	N	80	1,5	20	> 20	3 A	BD 170	4					
BD 170	P						BD 169	4					
BD 183	N						15	117	> 20	3 A	—	5	
BD 233	N						45	2	25	40	0,15 A	BD 234	4
BD 234	P	BD 233	4										
BD 235	N	BD 236	4										
BD 236	P	BD 235	4										
BD 237	N	80	2	30	> 25	1 A	BD 238	4					
BD 238	P						BD 237	4					
BD 239	N						BD 240	4					
BD 240	P						BD 239	4					
BD 241	N	45	3	40	> 30	0,3 A	BD 242	6					
BD 242	P						BD 241	6					
BD 243	N						BD 244	6					
BD 244	P						BD 243	6					
BD 245	N	60	6	65	> 40	1 A	BD 246	6					
BD 246	P						BD 245	6					
BD 247	N						BD 248	6					
BD 248	P						BD 247	6					
BD 249	N	25	10	80	> 25	1,5 A	BD 249	7					
BD 250	P						BD 249	7					
BD 250	P						25	125	> 25	1,5 A	BD 249	7	
BD 250	P						25	125	> 25	1,5 A	BD 249	7	
BD 435	N	32	4	36	> 85	0,5 A	BD 436	4					
BD 436	P						BD 435	4					
BD 437	N						BD 438	4					
BD 438	P						BD 437	4					
BD 439	N	45	4	36	> 40	0,5 A	BD 440	4					
BD 440	P						BD 439	4					
BD 441	N						BD 442	4					
BD 442	P						BD 441	4					
BD 643	N	80	8	62,5	> 750	3 A	BD 644	7					
BD 644	P						BD 643	7					
BD 645	N						BD 646	7					
BD 646	P						BD 645	7					
BD 675	N	60	4	40	> 20	0,5 A	BD 676	7					
BD 676	P						BD 675	7					
BD 677	N						BD 678	7					
BD 678	P						BD 677	7					
BD 679	N	40	3	40	> 25	1 A	BD 680	4					
BD 680	P						BD 679	4					
TIP 31	N						25	100 m	0,3	> 20	10 mA	—	1
TIP 32	N						25	100 m	0,3	> 20	10 mA	—	1
TIP 33	N	70	15	100	> 20	4 A	TIP 32	6					
TIP 34	P						TIP 31	6					
TIP 35	N						TIP 34	7					
TIP 36	P						TIP 33	7					
TIP 41	N	100	15	125	> 20	5 A	TIP 36	7					
TIP 42	P						TIP 35	7					
TIP 122	N						TIP 42	6					
TIP 127	N						TIP 41	6					
TIP 142	N	25	100	115	> 20	4 A	TIP 127	6					
TIP 147	N						TIP 122	6					
TIP 2955	P						TIP 147	7					
TIP 3055	N						TIP 142	7					
2N3055	N	70	15	115	> 20	4 A	TIP 127	6					
MJ 2955	N						TIP 147	7					
MJ 2955	P						TIP 142	7					
2N3055	P						TIP 147	7					
2N2955	P	25	100 m	0,3	> 20	10 mA	—	1					



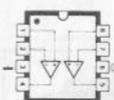
Lineare ICs



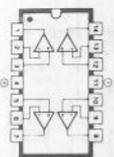
301
318
709
741
CA 3130
CA 3140
LF 355/356/357
TL 071/081



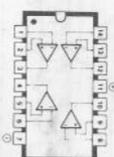
1458
4558



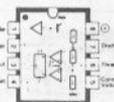
LM 387
NE 542



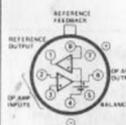
LM 324
TL 074
TL 084



RC 4136

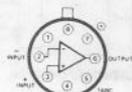


555

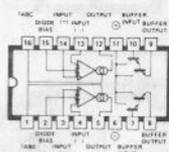


LM 10C

CA 3080



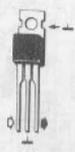
LM 13600



Eingang
Ausgang

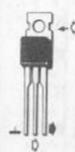
Alle Gehäuse: Ansicht von oben

Spannungsregler-ICs



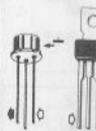
7805
7806
7808
7812
7815
7818
7824

$I_{out} = 1 \text{ A}$



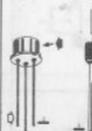
7905
7906
7908
7912
7915
7918
7924

$I_{out} = -1 \text{ A}$



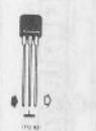
78M05
78M06
78M08
78M12
78M15
78M18
78M24

$I_{out} = 500 \text{ mA}$



79M05
79M06
79M08
79M12
79M15
79M18
79M24

$I_{out} = -500 \text{ mA}$



78L05
78L06
78L08
78L12
78L15
78L18
78L24

$I_{out} = 100 \text{ mA}$



79L05
79L06
79L08
79L12
79L15
79L18
79L24

$I_{out} = -100 \text{ mA}$



$U_{out} = 5 \text{ V}$

LM 309K
 $I_{out} = 1 \text{ A}$

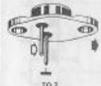
LM 323K
 $I_{out} = 3 \text{ A}$



$U_{out} = -5 \text{ V}$

$I_{out} = -3 \text{ A}$

$U_{out} = 1,2 \text{ V} \dots 37 \text{ V}$



LM 317K
 $I_{out} = 1,5 \text{ A}$



LM 723

$I_{out} = 200 \text{ mA}$

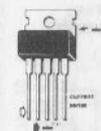
$U_{out} = \dots 37 \text{ V}_{max.}$

(pas de U_2)

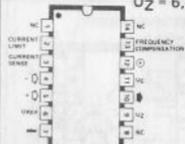
$U_{ref} = 7,15 \text{ V}$

$U_Z = 6,2 \text{ V}$

$U_{out} = 2,85 \text{ V} \dots 40 \text{ V}$



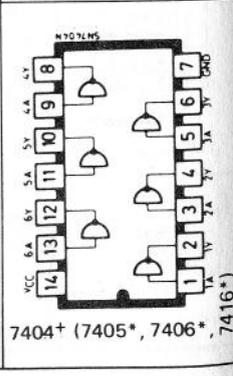
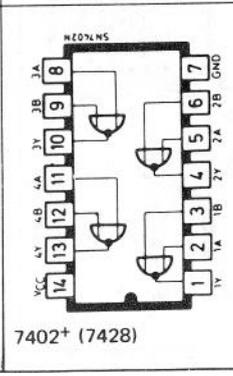
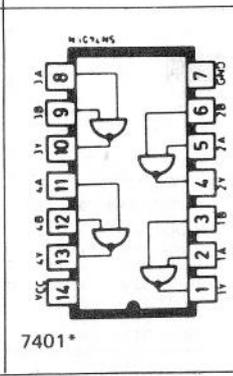
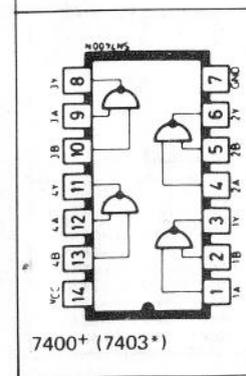
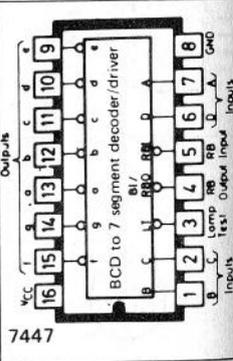
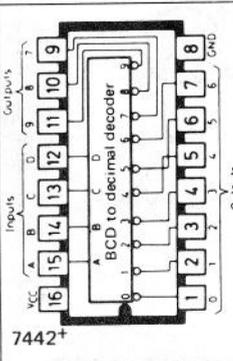
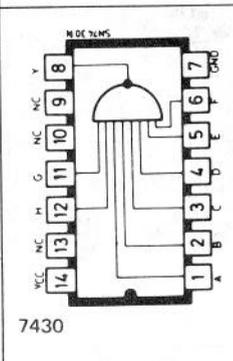
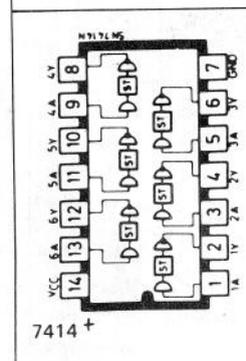
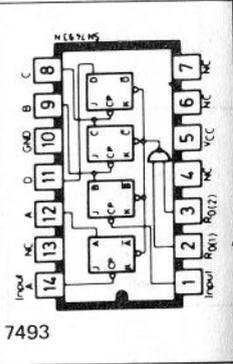
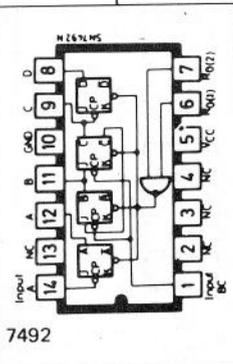
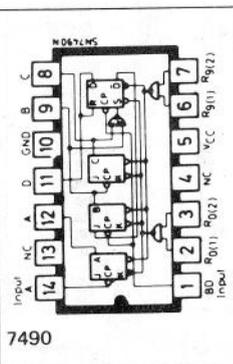
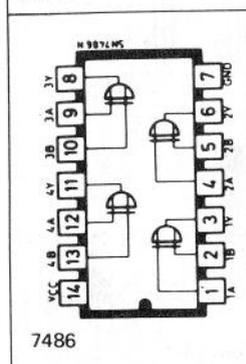
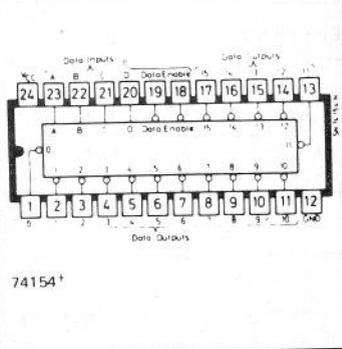
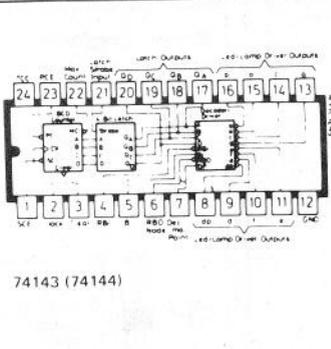
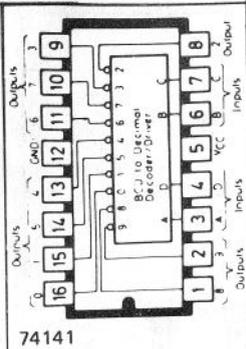
L 200
 $I_{out} = 2 \text{ A}$

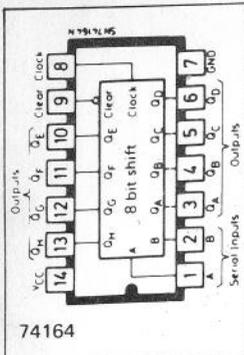


Eingangsspannungsbereich:

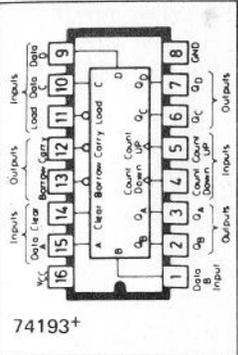
7805 = 8 V ... 35 V
7806 = 9 V ... 35 V
7808 = 11 V ... 35 V
7812 = 15 V ... 35 V
7815 = 18 V ... 35 V
7818 = 21 V ... 35 V
7824 = 27 V ... 40 V

7905 = -8 V ... -35 V
7906 = -9 V ... -35 V
7908 = -11 V ... -35 V
7912 = -15 V ... -35 V
7915 = -18 V ... -35 V
7918 = -21 V ... -35 V
7924 = -27 V ... -40 V

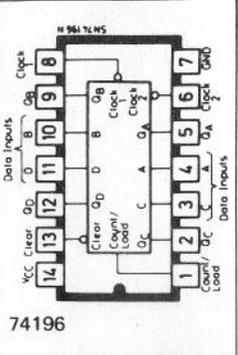




74164

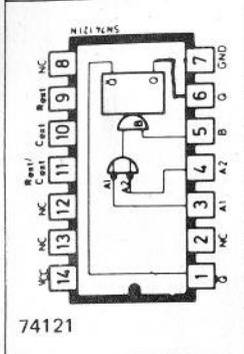


74193+

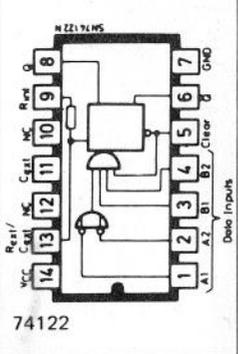


74196

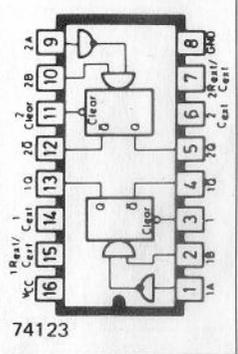
* open collector
 + auch in CMOS (pinkompatibel)
 erhältlich von Teledyne
 Semiconductor und National
 Semiconductor



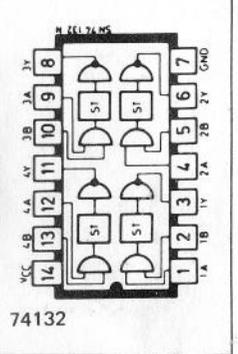
74121



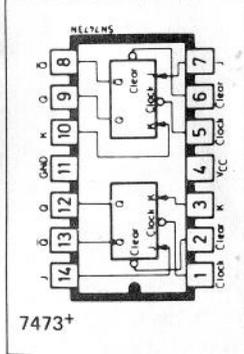
74122



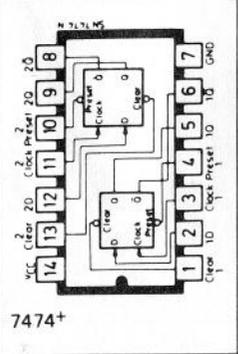
74123



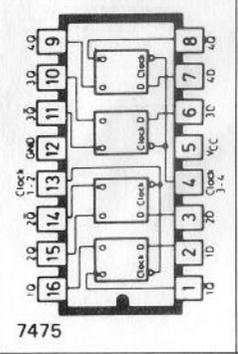
74132



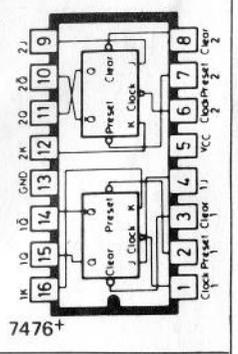
7473+



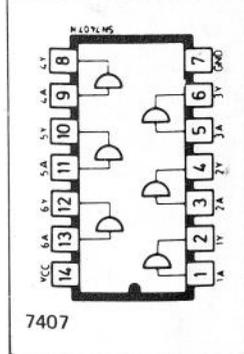
7474+



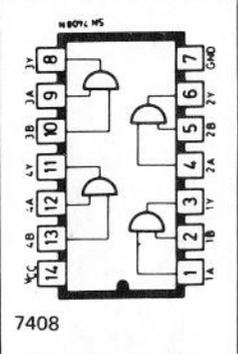
7475



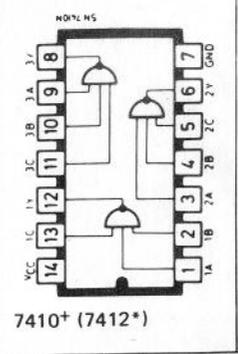
7476+



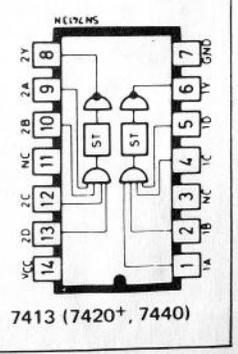
7407



7408



7410+ (7412*)



7413 (7420+, 7440)

hardware Beschläge (Holzbranche), Normteile (Maschinenbaubranche), Bausteine, Baugruppen und Geräte, vornehmlich auf dem Gebiet der Nachrichtenverarbeitung; . . . — so umschreibt ein Wörterbuch der Datenverarbeitung den Begriff "hardware". μ Computer-Hardware sind neben dem Computer selbst Schaltungen und Geräte, die zum Anschluß an bereits bestehende Mikrocomputersysteme geeignet sind. Dabei ist der Schaltungsaufbau so universell, daß sie mit nur wenig Änderungen zu verschiedenen Mikrocomputersystemen passen.

μ Computer-Hardware bietet Ihnen u. a.

- Elekterminal mit ASCII-Tastatur und Fernseher als Datensichtgerät
- Speisung für kleine und große μ P-Systeme
- 16 K. dynamische RAM-Karte
- EPROM-Programmiergerät
- Kassetten-Interface
- RS-232-Interface
- u. a. m.

μ Computer-Hardware gibt Ihnen die Möglichkeit, mehr aus Ihrem Mikrocomputer zu machen: einen kompletten Personalcomputer

Elektor Verlag GmbH

5133 Gangel 1

ISBN 3-921608-30-9