

Rolf-Dieter Klein, Joachim Arendt

Mikroelektronik im Fernsehen

Teil 3

Die vier Auszubildenden in der NDR-Fernsehserie „Mikroelektronik“ sind keineswegs Statisten. Obwohl das NDR-Klein-System in der Praxis erprobt ist, braucht man bei der Fernsehproduktion die ständige „Kontrolle“ durch die Adressaten. Insbesondere bedarf ein Lehrsystem der Lernkontrolle.

Die Auszubildenden (Funkelektroniker im zweiten Lehrjahr) sind Karin Bähre (Bild 1), Birger Schröder (Bild 2) und Ralph Schult (Bild 3); Jörg Lucassen ist Nachrichtengeräte-Mechaniker im ersten Lehrjahr (Bild 4). Sie spielen eine sehr aktive Rolle in dem Fernsehstück „Einführung in die Mikroelektronik“.



Bild 1. Karin Bähre

Arbeitsrhythmus, Tempo und Darstellungsformen wurden weitgehend durch sie als „Vertreter“ der Adressaten bestimmt. Und selbstverständlich stammen viele der praktischen Hardware-Beispiele und Bildschirmdarstellungen von den vier, teilweise sogar „live“ im Studio entwickelt.



Bild 2. Birger Schröder



Bild 3. Ralph Schult



Bild 4. Jörg Lucassen

Der Nichtstu-Befehl

In der sechsten Folge des Kurses werden Befehle in dualer „Schreibweise“ an den Pins, den Anschlußbeinen des Mikroprozessors Z80 „abgelesen“. Befehle in dualer Schreibweise rechnet man oft um und stellt sie sedezimal dar. Was heißt nun „duale“ und „sedezimale“ Schreibweise? Wir erklären das in drei Stufen. Mit dem Oszilloskop messen wir an den Anschlüssen der Datenaus- und -eingänge des Mikroprozessors Spannungen. Liegt an einem Anschlußbein eine Spannung zwischen 2,4 und 5 V, so definiert man diesen Spannungsbereich als „logisch eins“ und kennzeichnet diesen Zustand mit der Ziffer „1“.

Liegt an einem Anschlußbein eine Spannung zwischen 0 und 0,7 V, so definiert man diesen Spannungsbereich als „logisch null“ und kennzeichnet ihn mit der Ziffer „0“. Bitte lesen Sie dazu auch den Abschnitt 3.1 „Messen mit dem Multimeter“ auf Seite 37 von R.-D. Klein, „Microcomputer selbstgebaut und programmiert“, im folgenden „Begleitbuch“ genannt. Bei der Übertragung von Befehlen oder Daten beschränkt man sich stets auf diese beiden Spannungsbereiche; man kommt also bei ihrer Darstellung mit zwei Zeichen aus. Beispiele zeigen die Bilder 5 bis 7.

Das Dualsystem

„Zwei“ heißt lateinisch „duo“, danach nennt man ein Zeichensystem, das mit zwei Zeichen auskommt, „Dualsystem“, Computerbefehle können also im Dualsystem geschrieben werden. Hier drei Beispiele für den Prozessor Z80:

```
NOP = 0 0 0 0 0 0 0 0
OUT = 1 1 0 1 0 0 1 1
IN = 1 1 0 1 1 0 1 1
```

Mit dem Dualsystem kann man aber nicht nur Befehle oder sonstige Daten darstellen, sondern auch Zahlen schreiben und rechnen.

Daher müssen wir uns mit den Zahlensystemen noch etwas genauer befassen. Unser vertrautes Zehnersystem kommt mit zehn Zeichen aus, man kann nämlich mit den Ziffern

0 1 2 3 4 5 6 7 8 9

jede beliebig große oder kleine Zahl darstellen. Allerdings braucht man dazu ein zweites Darstellungsmittel: die Stellenwertschreibweise. Man spricht daher statt von „Zahlensystemen“ auch von „Stellenwertsystemen“. Was ist ein „Stellenwertsystem“? Bleiben wir zur Klärung dieses Begriffs zunächst bei un-

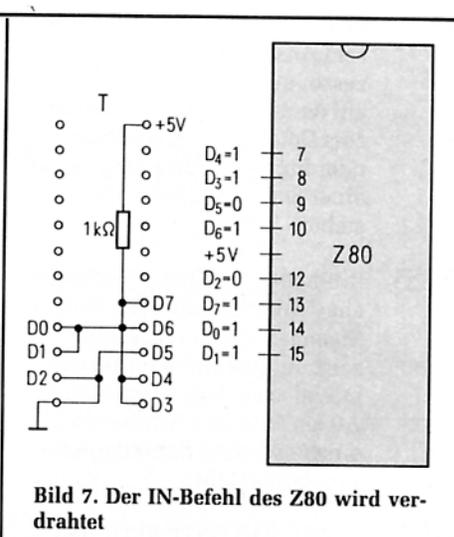
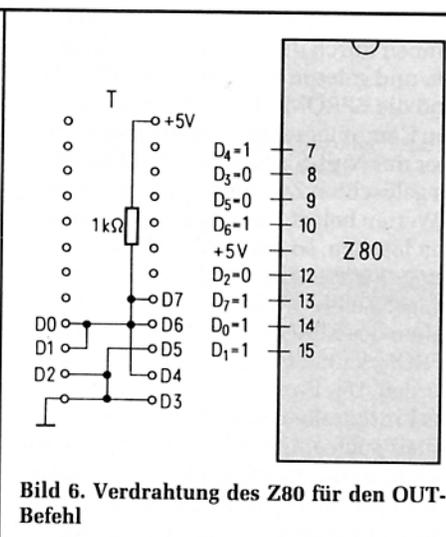
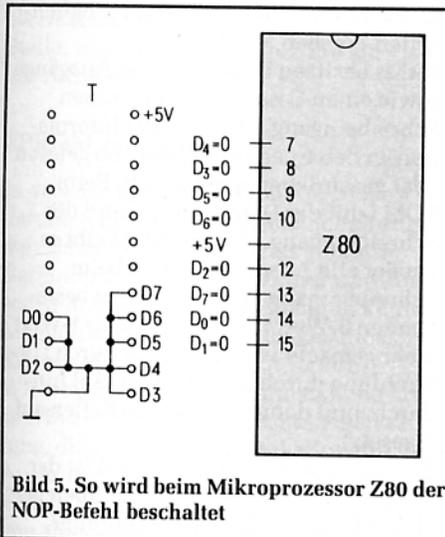


Bild 5. So wird beim Mikroprozessor Z80 der NOP-Befehl beschaltet

Bild 6. Verdrahtung des Z80 für den OUT-Befehl

Bild 7. Der IN-Befehl des Z80 wird verdrahtet

serem vertrauten Zehnersystem. Jede Ziffer einer Zahl steht an einer bestimmten Stelle und bekommt einen entsprechenden „Wert“: 7 9 7 3 „Siebentausend-neunhundert-dreiundsiebzig“ lesen wir diese Zahl. Das heißt: Die „7“ an der vierten Stelle (von rechts nach links gezählt) hat den Wert 7×1000 oder 7×10^3 , die „9“ an der dritten Stelle hat den Wert 9×100 oder 9×10^2 , die „7“ an der zweiten Stelle (immer von rechts nach links gezählt) hat den Wert 7×10 oder 7×10^1 und die 3 an der ersten Stelle hat den Wert 3×1 oder 3×10^0 .

Umrechnung ins Dezimalsystem

Entsprechend verfährt man beim Dual- oder Zweier-System: Jede „Stelle“ einer Dualzahl wird mit Potenzen von zwei multipliziert. Ein Beispiel:

1101 1011 wird zerlegt.

$$\begin{aligned}
 1 \times 128 &= 1 \times 2^7 \\
 1 \times 64 &= 1 \times 2^6 \\
 0 \times 32 &= 0 \times 2^5 \\
 1 \times 16 &= 1 \times 2^4 \\
 \\
 1 \times 8 &= 1 \times 2^3 \\
 0 \times 4 &= 0 \times 2^2 \\
 1 \times 2 &= 1 \times 2^1 \\
 1 \times 1 &= 1 \times 2^0
 \end{aligned}$$

Wenn man die Übersetzung der Stellenwerte der Dualzahl addiert, erhält man die Dezimaldarstellung dieser Zahl: $1101 1011 = 219$. Da 1101 1011 der IN-Befehl war, kann man also den IN-Befehl dezimal durch 219 wiedergeben. Die Berechnung einer Dualzahl aus einer Dezimalzahl mit Hilfe des „euklidischen Algorithmus“ haben wir im Februarheft der mc gezeigt.

Das Sedezimal-System

Man kann mit Hilfe der Stellenwertdarstellung und einer beliebig festzulegenden Zahl von Ziffern jedes beliebige Wort und jede denkbare Zahl darstellen. Wegen seiner bequemen Teilbarkeit durch 2 hat sich in der Mikroprozessortechnik neben dem Dualsystem das Sedezimalsystem durchgesetzt. „sedecem“ ist ein lateinisches Wort und heißt auf deutsch sechzehn. Das Sedezimalsystem – oft falsch „Hexadezimal-“ oder „Hexsystem“ genannt – braucht zur Darstellung von Zahlen und anderen Begriffen sechzehn Zeichen. Man nimmt zur Sedezimaldarstellung die zehn Ziffern 0 1 2 3 4 5 6 7 8 9 und die Buchstaben A B C D E F. Vergleichen wir einmal in einer Liste die Darstellung der Zahlen von eins bis zwanzig in den drei Zahlen- oder Stellenwertsystemen:

Dezimal	Dual	Sedezimal
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	1 0000	10
17	1 0001	11
18	1 0010	12
19	1 0011	13
20	1 0100	14

Man erkennt wohl sehr gut, wie leicht sich eine verhältnismäßig lange duale Darstellung in eine knappe sedezimale Darstellung umwandeln läßt. Ein Beispiel dafür findet man in der Februarangabe der mc.

Bei der Frage der dualen Darstellung von Befehlen haben wir uns zugleich noch einmal die Lage der Anschlußbeine der Datenleitungen des Mikroprozessors Z80 beziehungsweise Z80A, der 4-MHz-Version dieses Prozessortyps, bewußt gemacht. In der Sendung werden noch vier weitere, sehr wichtige Anschlüsse des Z80 gezeigt, an die wir uns noch einmal erinnern sollten (Bild 8):

RD (READ = Lesen): An diesem Anschluß gibt der Prozessor Null-Signale aus, wenn er Daten aus dem Speicher oder von einem Port lesen will.

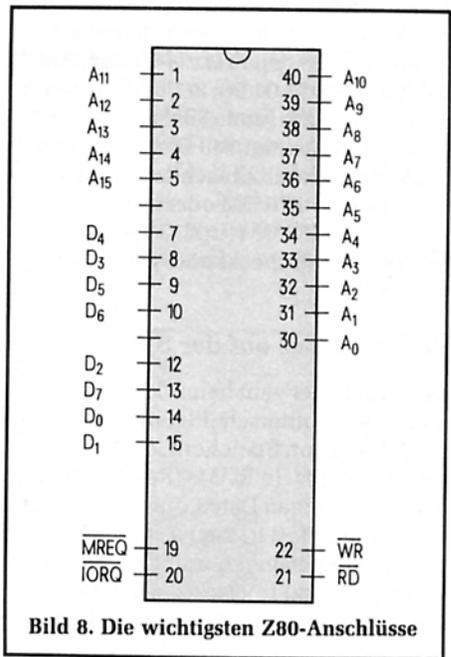


Bild 8. Die wichtigsten Z80-Anschlüsse

WR (WRITE = Schreiben): An diesem Anschluß sendet der Prozessor ein 0-Signal aus, wenn auf dem Daten-Bus des Prozessors Daten für den Speicher oder für einen Ausgabe-Port an einer Schnittstelle bereitstehen.

MREQ (MEMORY REQUEST = Speicher-Anforderung): Ein Null-Signal an diesem Anschluß zeigt an, daß auf dem Adreßbus, also an den Anschlüssen A0 bis A15 eine Adresse für eine Lese- oder Schreiboperation bereitsteht.

IORQ (INPUT/OUTPUT REQUEST = Ein/Ausgabe-Anforderung): Ein Null-Signal an diesem Anschluß zeigt an, daß an den Adreßleitungen A0 bis A7, also an den „unteren“ Adreßleitungen eine von 2^8 , das sind 256 Adressen für eine Lese- oder Schreiboperation von oder zu einem von 256 möglichen Ports bereitsteht.

Die vollständige Pinbelegung findet sich im Begleitbuch auf S. 397, Abb. 9.1.

Der Kurs auf Kassetten

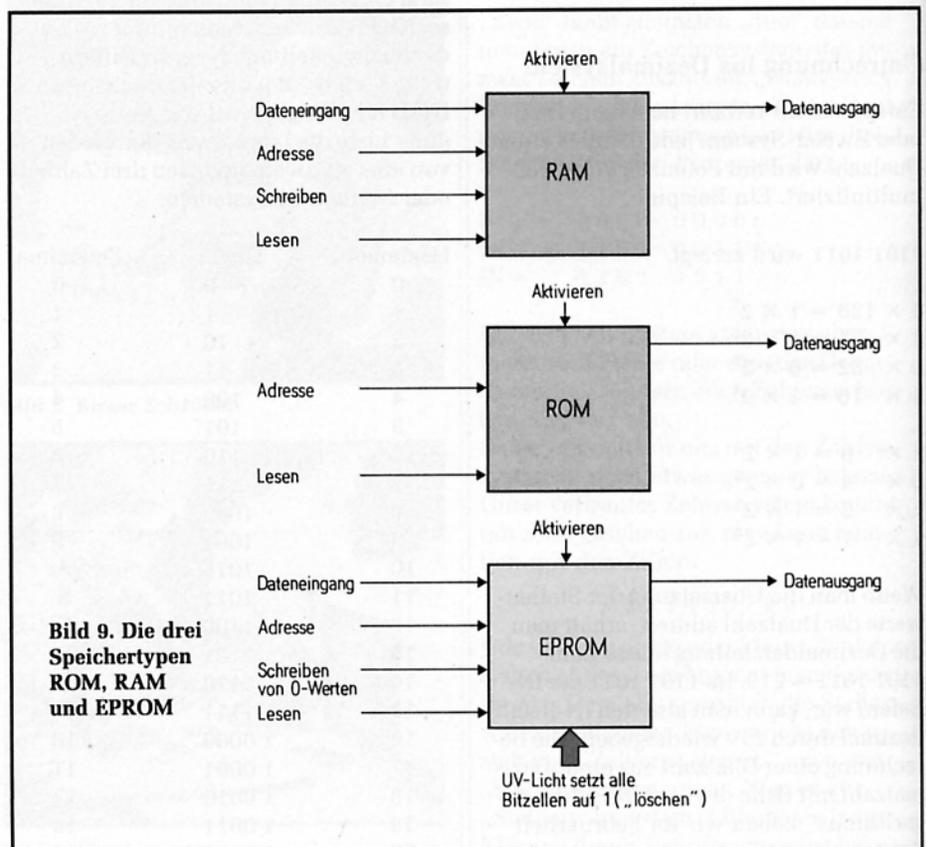
Für die betriebsinterne berufliche Aus- und Weiterbildung, aber auch für Hobby-Clubs und Interessierte, für ihre private Nutzung des Kurses außerhalb der Sendezeiten oder in Ländern, in denen der Kurs gerade nicht ausgestrahlt wird, hat die Produktionsfirma Studio Hamburg Kassetten produziert. Wer sich für den Erwerb der Kassetten interessiert, wende sich an den Franzis-Verlag, Abt. ZV, Postfach 37 01 20, 8000 München 37. Die zwei Kassetten (Beta, VHS oder Video-2000) kosten 248 DM; die Bestellung erfolgt durch Einsenden eines Schecks über 250 DM oder durch Überweisen von 250 DM (inkl. 2 DM Porto) auf unser Postscheckkonto München 813 75-809.

Dem Speicher auf der Spur

Ohne Speicher geht beim Computer nichts. Man unterscheidet im Prinzip zwei Arten von Speichern, die ROMs und die RAMs. In ROMs (Read Only Memory) können Daten nur gelesen werden, sie sind fest in dem IC programmiert, zum Beispiel durch einen metallischen Aufdampfungsprozeß bei der Herstellung des ICs. In einem RAM (Random Access Memory) sind die Daten

variabel untergebracht, das heißt, sie können durch den Computer geschrieben und gelesen werden. Eine dritte Art sind die EPROMs. Daten können durch den Computer eingeschrieben werden, aber nur Null-Signale. Das EPROM ist im gelöschten Zustand daher mit lauter 1-Werten belegt. Will man die Information löschen, so muß UV-Licht auf ein Quarz-Fenster gestrahlt werden. Nach einiger Zeit (etwa 20 Minuten) sind alle Zellen des EPROMs wieder auf 1. Das EPROM kann dann erneut programmiert werden. Der Programmiervorgang, also das Einschreiben von Null-Signalen, dauert auch einige Minuten. Daher kann man EPROMs nicht wie RAM-Bausteine einsetzen, wo es auf schnellen Datenaustausch ankommt, sondern eher wie ROMs. Die meisten Bausteine mit festen Programmen sind bei uns EPROMs, da sie sich sehr leicht herstellen lassen und die Information dauerhaft (etwa 10 Jahre) speichern. Die Information wird übrigens als elektrische Ladung gespeichert. Bild 9 zeigt nochmals eine Zusammenfassung der Speichertypen. Jeder der Speicher besitzt einen Adreßeingang, da ja unterschiedliche Zellen in dem Speicher angesprochen werden sollen. Ein Speicher mit einer Kapazität von 1024 Bit z. B. müßte 10 Adreßleitungen besitzen, da sich mit 10 Leitungen

2^{10} Kombinationen, also 1024 Möglichkeiten ergeben. RAMs besitzen Datenein- und Ausgänge sowie einen Leseeingang und einen Schreibeingang, über den die Information an den Speicher gelangt, ob gelesen oder geschrieben werden soll. Beim ROM fehlt der Dateneingang und der Schreibeingang. Beim EPROM gibt es wieder alle Anschlüsse, aber beim Schreibvorgang kann aus einem bestehenden 0-Wert im Speicher kein 1-Wert mehr gemacht werden, außer durch UV-Strahlung durch den Quarzdeckel hindurch, und dann werden alle Zellen auf 1 gesetzt. Ein weiteres wichtiges Element ist der Decoder. Er kommt sowohl im Speicher als auch außerhalb in der Computerschaltung vor. Er soll aus einer Adresse ein Auswahlsignal machen, das über eine einzelne Leitung eine Speicherzelle oder einen Speicherbaustein anwählt. Bild 10 zeigt eine Wahrheitstabelle. Die Stellen, die mit einem x markiert sind, bedeuten, daß der logische Wert an diesem Eingang sowohl 0 als auch 1 sein kann, ohne das Verhalten der Verknüpfung zu verändern. Liegt zum Beispiel der Code 011 an den Eingängen ABC und sind G1 und G2 auf 0 und G3 auf 1, so liegt am Ausgang 6 ein 0-Signal. Wenn man die Adresse an ABC



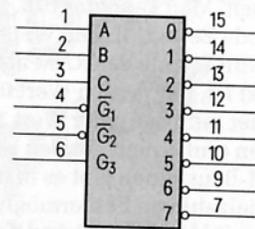
als Dualzahl liest, mit der höherwertigen Stelle, so lag der duale Wert 110 an, also die Adresse 6.

EPROM macht Musik

In dieser Folge der Sendereihe geht es darum, eine kleine Interface-Schaltung aufzubauen. Interface nennt man die Verbindung zwischen Rechner und Aussenwelt. Bisher hatte der Rechner keine Möglichkeit, außer über ein Oszilloskop Signale an die Umwelt zu liefern. Wir wollen den Rechner musizieren lassen. Der Prozessor Z80 besitzt eine Leitung, die für die Kommunikation mit der Aussenwelt gedacht ist, die Leitung $\overline{\text{IORQ}}$. Daß die Leitung in Abhängigkeit von einem Programm gesteuert werden kann, haben wir ja schon bei der Folge „Nichtstu-Befehl“ gesehen. Mit einem größeren Programm ist es möglich, Musik an dieser Leitung zu erzeugen. Das Programm ist in einem EPROM untergebracht. Für diese Folge werden, falls alle Versuche durchgeführt werden sollten, drei EPROMs benötigt: das EPROM MUO, RWT und MUM. Die SBC2-Karte besitzt in dieser Ausbauphase noch keinen RAM-Speicher, sondern nur vier leere 24polige Fassungen für RAMs und EPROMs. Das EPROM mit dem Namen MUO wird nun in die Fassung 0 gesteckt. Es ist nicht gleichgültig, in welche Fassung man das EPROM steckt, da sonst das Programm nicht arbeitet. Daher bitte auf die Beschriftung achten. Nun kann die SBC2-Karte wieder an die Spannungsversorgung angeschlossen werden. Mit einem Oszilloskop kann man nun die Pulsformen am Prozessor beobachten. Wir suchen nur den Anschluß mit der Bezeichnung $\overline{\text{IORQ}}$, das ist das Pin Nr. 20. Dort wird der Eingang des Oszilloskopes angeschlossen. Die Masse des Oszilloskopes wird zuvor mit der Masse der POW5V-Baugruppe verbunden, z. B. kann man die Masseleitung an dem Minuspol des Elkos auf der POW5V-Baugruppe anklammern.

Der Spannungsbereich des Oszilloskopes wird auf 2V/cm gestellt und die Triggerrichtung auf Automatik. Mit dem Zeitbassschalter kann nun der richtige Bereich eingestellt werden. Dabei wird man unregelmäßig hin- und herschwankende Pulsgruppen entdecken, das ist unser Musikstück. Um einen Lautsprecher an den Rechner anschließen zu können, braucht man einen Verstärker. Bild 11 zeigt den kompletten Schaltplan. Für den Aufbau eignet sich z. B. eine kleine Lochrasterplatte. Daß der Transi-

Bild 10. Der Decoder 74LS138 mit Wahrheits-Tabelle



A	B	C	\overline{G}_1	\overline{G}_2	G_3	0	1	2	3	4	5	6	7
X	X	X	1	X	X	1	1	1	1	1	1	1	1
X	X	X	X	1	X	1	1	1	1	1	1	1	1
X	X	X	X	X	0	1	1	1	1	1	1	1	1
0	0	0	0	0	1	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
0	1	0	0	0	1	1	1	0	1	1	1	1	1
1	1	0	0	0	1	1	1	1	0	1	1	1	1
0	0	1	0	0	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	1	0	1	1	1
0	1	1	0	0	1	1	1	1	1	1	0	1	1
1	1	1	0	0	1	1	1	1	1	1	1	0	1

stor allein noch nicht als Verstärker reicht, haben wir schon in der Sendung gezeigt. Die Pulse der CPU sind zu kurz, um den Transistor richtig durchzuschalten. Wer will, kann übrigens auch einmal andere Anschlüsse mit dem Eingang des Transistortreibers verbinden. Dabei gibt es eine Überraschung: Es ertönt manchmal wesentlich lautere Musik, jedoch in unterschiedlichen Klangfarben, warum?

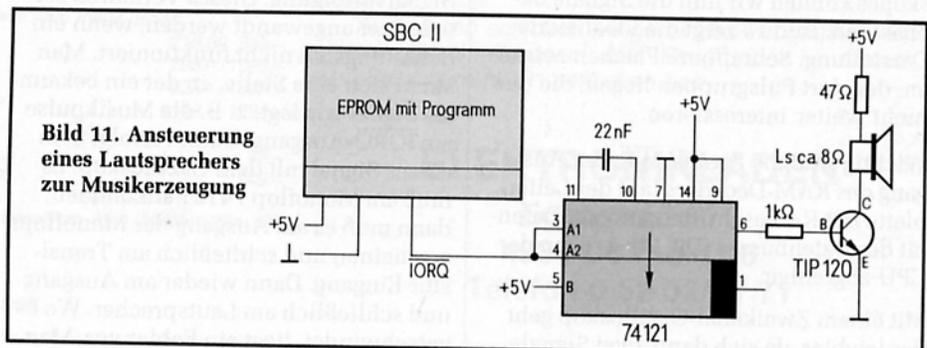
Der Z80 durchläuft das Musikprogramm immer wieder. Dabei ändern sich die Daten, wie auch Adressen ähnlich rhythmisch wie das Musikstück. Warum sind hier die Pulse nicht zu kurz? Weil Adressen und Daten auf Pegeln längere Zeit verweilen können; mit dem Oszilloskop kann man sich Aufschluß verschaffen. Die bessere Methode jedoch ist, ein Monoflop zu verwenden. Das Monoflop macht aus den kurzen Pulsen des $\overline{\text{IORQ}}$ -Ausgangs längere. Wie lang sie werden, bestimmt der Kondensator am Monoflop. Wird er vergrößert, so vergrößern sich auch die Pulsbreiten.

Das RAM wird eingebaut

Bisher arbeitete das Programm noch ohne zusätzliche RAM-Speicher. Wir wollen jetzt den Speicher ausbauen, um später auch komplizierte Programme laufen lassen zu können. Das Programm MUO ist so ausgeklügelt, daß es ohne RAM-Baustein auskommt. Der RAM-Baustein HM 6116 wird in die Fassung 2 gesteckt, Fassung 0 und 3 sind noch leer. Ein zweites RAM könnte auch schon jetzt in die Fassung 3 gesetzt werden, ist jedoch bis zur Folge „Schreiben lernen“ noch nicht nötig.

Um die prinzipielle Arbeitsweise von Speichern nochmals zu verstehen, und auch um den Speicher ganz grob zu testen, wollen wir ein weiteres Experiment anschließen. Dazu wird anstelle des MUO-EPROMs das EPROM mit der Bezeichnung RWT eingesteckt. RWT ist die Abkürzung für Read-Write-Test, also Lese-Schreib-Test. Um einen Speicher zu testen, muß man in ihm ein Datenwort schreiben und es anschließend

Bild 11. Ansteuerung eines Lautsprechers zur Musikerzeugung



wieder auslesen. Man kann das tun, indem man in jede Zeile z. B. den Wert 0 einschreibt. Nun könnte das RAM aber, wenn es defekt ist, immer den Wert 0 behalten. Daher muß dann der Wert 1 eingeschrieben und erneut gelesen werden. Bei RAM-Bausteinen gibt es in der Praxis eine Vielzahl von Fehlermöglichkeiten, und ausführliche Speicher-Test-Programme sind sehr kompliziert.

Im Prinzip müßte man für eine vollständige Prüfung jede mögliche Datenkombination in einen Speicher einschreiben. Bei 8196 Speicherzellen sind das aber 2 Hoch 8196 Möglichkeiten, eine astronomische Zahl, die innerhalb der Lebensdauer des Universums nicht zu testen ist. In der Praxis beschränkt man sich daher auf eine Auswahl dieser Kombinationen. Käufliche RAM-Bausteine können also theoretisch nie vollständig getestet sein.

Zum Glück sind die verwendeten RAM-Bausteine heutzutage sehr zuverlässig und Speicher-Fehler nur sehr selten.

Wir wollen also nur das Prinzip kennenlernen und den Test mit einem Oszilloskop durchführen. Dazu wird vom RWT-EPROM jeweils ein Datenwort mit 0- und eines mit 1-Signalen in eine einzige Speicherzelle geschrieben, nämlich die erste in dem RAM-Baustein auf Fassung 2.

Das Oszilloskop wird diesmal mit einer sogenannten externen Triggerung betrieben. Das heißt, wann der Schreibstrahl losläuft, wird durch einen Impuls an dem sogenannten Trigger-Eingang des Oszilloskopes ausgelöst. Dieser Eingang wird mit dem $\overline{\text{IORQ}}$ -Ausgang der SBC2-Karte verbunden. Das RWT-EPROM signalisiert eine Testphase durch einen Impuls auf der Leitung $\overline{\text{IORQ}}$. Dadurch wird es möglich, die sehr komplizierten Impulsgruppen, die wir messen wollen, immer auf die gleiche Stelle auf den Schirm des Oszilloskopes zu bringen. Bild 12 zeigt die prinzipielle Meßanordnung. Mit dem Meßeingang des Oszilloskopes können wir nun die Signale beobachten. Bild 13 zeigt die idealisierte Darstellung. Schraffierte Flächen zeigen an, daß dort Pulsgruppen liegen, die uns nicht weiter interessieren.

Wichtig sind die Signale $\overline{\text{WR}}$, der Ausgang des RAM-Decoders (auf der Leiterplatte mit RA beschriftet) und ein Datenbit des Datenbusses (D0...7), z. B. an der CPU abgreifbar.

Mit einem Zweikanal-Oszilloskop geht das leichter, da sich dann zwei Signale

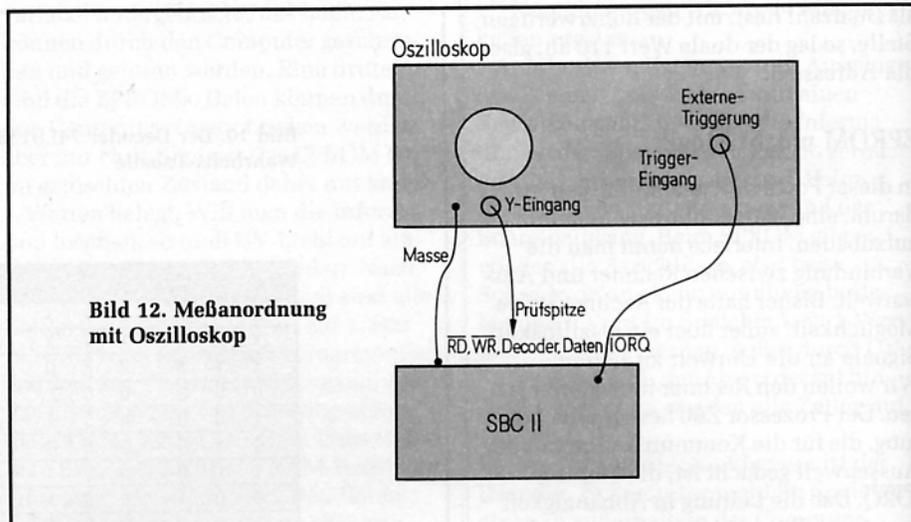


Bild 12. Meßanordnung mit Oszilloskop

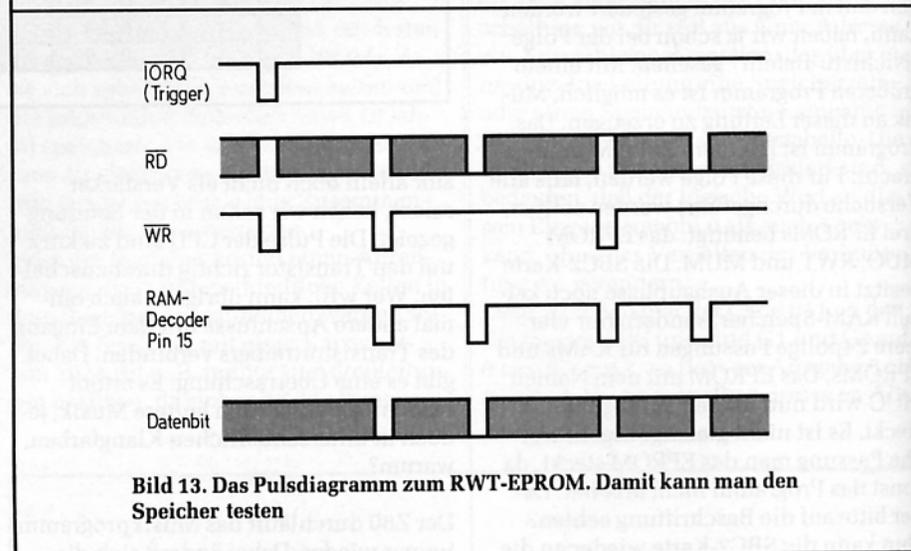


Bild 13. Das Pulsdiagramm zum RWT-EPROM. Damit kann man den Speicher testen

gleichzeitig darstellen lassen, z. B. der Ausgang des RAM-Decoders, der angibt, wo die interessanten Stellen liegen und eine Leitung des Datenbusses. Nach diesem Test kann man das EPROM MUM einstecken und hört nun Musik vierstimmig. Das Programm MUM läuft nur, wenn das RAM eingesteckt ist.

Fehlermöglichkeiten

Eine der wichtigsten Methoden ist die Signalverfolgung. Dieses Verfahren sollte immer angewandt werden, wenn ein Schaltungsteil nicht funktioniert. Man sucht sich eine Stelle, an der ein bekanntes Signal vorliegt, z. B. die Musikpulse am $\overline{\text{IORQ}}$ -Ausgang. Dann verfolgt man dieses Signal mit dem Oszilloskop. Es muß am Monoflop 74121 ankommen; dann muß es am Ausgang des Monoflops erscheinen und schließlich am Transistor-Eingang. Dann wieder am Ausgang und schließlich am Lautsprecher. Wo es verschwindet, liegt ein Fehler vor. Man

prüft alle Beschaltungen und Leitungen, die an diese Stelle führen. Da es sich meist um Verdrahtungsfehler handelt, sind Fehler so schnell zu finden.

Fortsetzung folgt

Materialliste für die Folge „EPROM macht Musik“

1. POW5V-Bausatz und Transformator
2. SBC-II-Bausatz
3. Lochrasterplatte ca. 100 mm × 70 mm
4. IC 74 121 (Monoflop)
5. 14polige IC-Fassung
6. kleiner Lautsprecher, 8 Ω
7. Widerstand 47 Ω, ¼ W
8. Widerstand 1kΩ, ¼ W
9. Transistor TIP 120 (oder TIP 110)
10. EPROM mit MUO-Programm*
11. EPROM mit RWT-Programm*
12. EPROM mit MUM-Programm*
13. Verbindungslitze

* vom Franzis-Software-Service