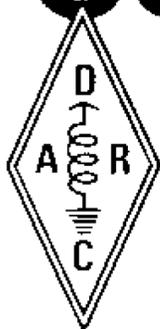


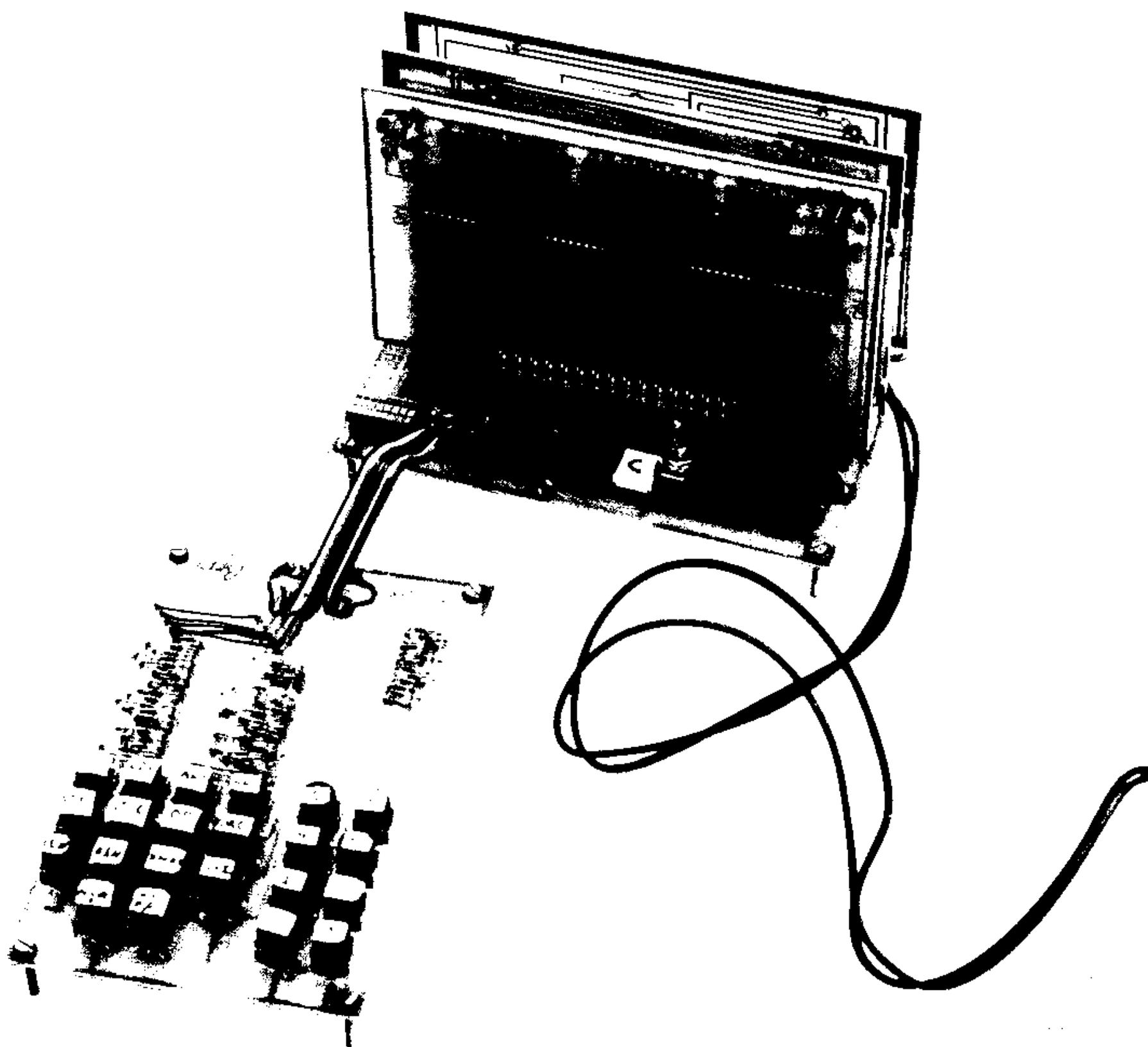
Cq

DL 6/84

1 P 7308 E
Juni



Clubzeitschrift des DARC
Fachorgan für den Amateurfunkdienst



1-Bit-Lerncomputer

Ein 1-Bit-Lerncomputer zum Einstieg in die Mikrocomputertechnik

Burkhard John, DK5JG, Zweibrückenstr. 35, 4000 Düsseldorf, 0211/23 44 89 und Volker Ludwig, DDØEU, Am Alten Bach 14 a, 4040 Neuss 21, 02107/7 02 26 als Sachbearbeiter „Amateurfunk in der Schule“ im Jugend- und Ausbildungsreferat des DARC

Zusammenfassung

Der Beitrag beschreibt die Funktion, den Aufbau und die Betriebspraxis inklusive Programmierung eines auf die wesentlichen Bestandteile reduzierten Mikrocomputersystems. Der Einstieg in die immer komplexer werdende Mikrocomputertechnik wird damit außerordentlich erleichtert. Dieses System wurde vom Jugend- und Ausbildungsreferat des DARC entwickelt und wird auf der Ham Radio in Friedrichshafen ausführlich vorgestellt.

1. Vom festen Schaltkreis zum Mikroprozessor

Folgende Aufgabe soll elektronisch gelöst werden: Wenn der Motor läuft UND das Auto in die Dunkelheit kommt, soll sich die Fahrzeugbeleuchtung einschalten. Dieser Bedingung liegt eine logische Verknüpfung zugrunde, die UND-Verknüpfung heißt, welche mit Hilfe einer fest verdrahteten Elektronik hergestellt werden kann. Abb. 1 zeigt dafür das Schaltzeichen.

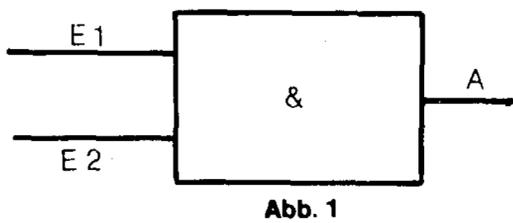


Abb. 1

Zur Ansteuerung der Eingänge E1 und E2 sind zwei Meßwertaufnehmer nötig. Folgende Spannungswerte sollen zugrunde gelegt werden:

Meßwertaufnehmer Fahrzustand:

Zündschloß aus: 0 Volt

Zündschloß ein: 12 Volt

Meßwertaufnehmer Helligkeit:

z. B. ein Fototransistor

Helligkeit: 0 Volt

Dunkelheit: 12 Volt.

Die Schaltung soll unter den oben angegebenen Bedingungen 12 V Spannung abgeben, z. B. zur Steuerung eines Relais für die Scheinwerfer.

Abb. 2 zeigt die Möglichkeiten, die eintreten können:

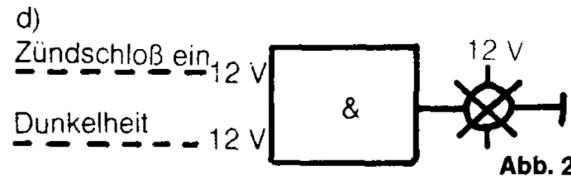
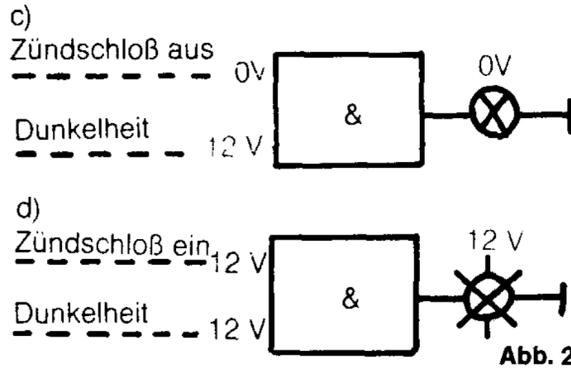
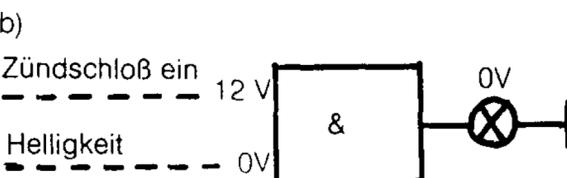
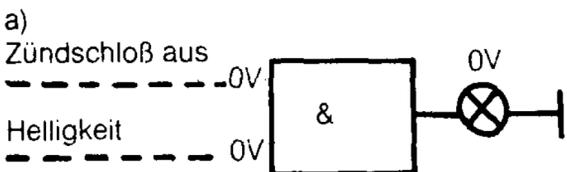


Abb. 2

Nur wenn an beiden Eingängen der UND-Schaltung eine Spannung von 12 V liegt, liegen am Ausgang auch 12 V. Eine mögliche Schaltung, die diese Bedingung in fest verdrahteter Elektronik verwirklicht, zeigt Abb. 3.

Geräte mit fester Verdrahtung haben den Vorteil der sofortigen Benutzbarkeit ohne erforderliche Grundkenntnisse.

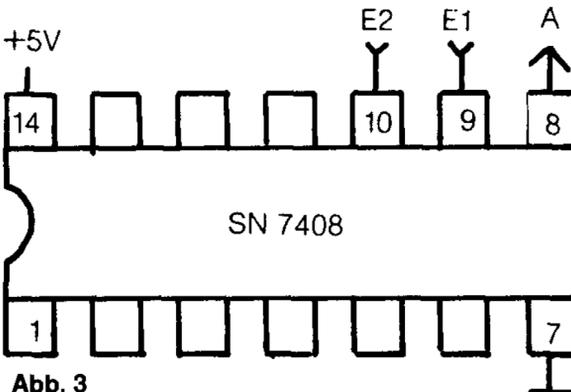
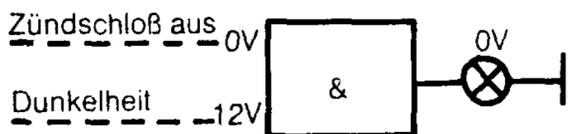


Abb. 3

Folgende zusätzliche Aufgabenstellung soll gelöst werden: Wenn das Auto parkt und es ist dunkel, soll sich die Parkleuchte einschalten.

Fall c aus Abb. 2 würde eintreten:



Das Gerät versagt, da sich die Parkleuchte nicht einschaltet. Baut man jedoch ein neues Gerät mit einer Schaltung, wie sie in Abb. 4 bzw. Abb. 5 dargestellt ist, kann das Problem gelöst werden.

Jede Aufgabenstellung benötigt ein eigenständiges Gerät.

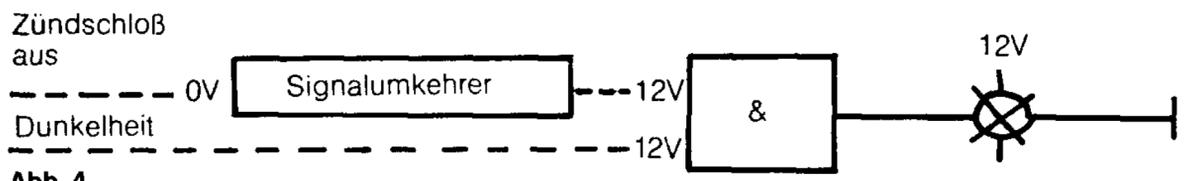


Abb. 4

Die beiden oben genannten Anwendungsbeispiele können durch ein einziges Gerät, dem ein Mikroprozessor zugrunde liegt, der zudem noch für viele andere Aufgaben zur Verfügung stehen würde, aufgebaut werden, ohne jedesmal die Schaltung des Gerätes verändern zu müssen.

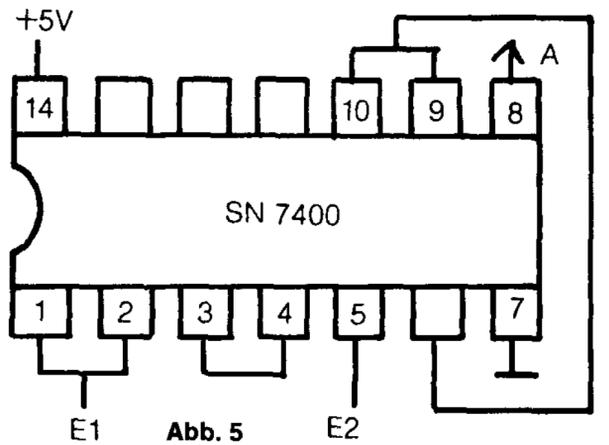


Abb. 5

Mikroprozessoren haben den Vorteil, daß in einem Gerät viele Anwendungen enthalten sind.

Der 1-Bit-Prozessor von Motorola MC 14500 schließt die Lücke zwischen der einfachen, fest verdrahteten Schaltungstechnik und der schwerer verständlichen Mikroprozessortechnik.

Der Umgang mit frei programmierbaren Mikroprozessoren verlangt umfangreiche Kenntnisse der Programmierung. Programmierung eines Mikroprozessors ist die sinnvolle Aneinanderreihung von für den Prozessor geeigneten Befehlen, die diesen veranlassen sollen, bestimmte Aufgaben zu lösen. Solche Befehle sind z. B. UND, ODER. Im Gegensatz zu herkömmlichen Mikroprozessoren beschränkt sich der MC 14500 auf einen Befehlssatz von lediglich 16 Befehlen. Sein Anwendungsbereich liegt deswegen im wesentlichen bei Steuerungen, wie z. B. die eines Elektromotors nach rechts oder links. Für den Anfänger bedeutet dies aber sicherlich eine Erleichterung.

Der MC 14500 ist ein 1-Bit-Prozessor.

Ein Bit, das ist die Nachrichteneinheit „ja“ oder „nein“ bzw. „0“ oder „1“, elektrisch jeweils verwirklicht durch z. B. 0 Volt bzw. 5 Volt. Somit ist ein 1-Bit-Prozessor ein System, das die Informationen „0“ bzw. „1“ mit Hilfe von Programmbefehlen in einer bestimmten Abfolge verarbeitet.

2. Eigenschaften des Lerncomputers

2.1 Anwendungstechnische Eigenschaften

In der Technik wird er z. B. zur Steuerung von Aufzuganlagen verwendet. Auch die technischen Eigenschaften des vorgestellten Computersystems sind darauf ausgerichtet. Es hat vier Eingangsleitungen, die entweder durch Schalter auf der Grundplatine (vgl. 3.4) oder durch angeschlossene Geräte beschaltet werden können. Auf diesem Weg kommen Signale in den Prozessor hinein. Weiter gibt es 8 Ausgabeleitungen, die sowohl an 8 Leuchtdioden (LED) (vgl. 3.1) als auch an entsprechende Anschlüsse einer Steckerleiste geführt sind. Über diese Leitungen können Ergebnisse oder Entscheidungen des Prozessors angezeigt bzw. an angeschlossene Geräte weitergegeben werden, die dann die Informationen in Steuervorgänge umsetzen, z. B. Steuerung eines Relais. Drei dieser Ausgänge sind zusätzlich mit drei weiteren Eingangsleitungen verbunden, so daß der Prozessor auf seine alten Ergebnisse, d. h. Daten, zurückgreifen kann, d. h., das System hat drei Datenspeicher. Zu einem Zeitpunkt kann nur die Information von einer Datenleitung verarbeitet werden, daher der Name „1-Bit-Prozessor“.

2.2 Eigenschaften als Lehr- bzw. Lernsystem

Nachdem ein Computerkenner den Teil 2.1 gelesen hat, wird er wahrscheinlich einwenden: „Das kann ein normaler 8-Bit-Mikroprozessor doch viel besser, was soll ich noch mit diesem Einfachstgerät?“

Für die Anwendungstechnik hat er recht. Das hier vorgestellte Computersystem dient aber einem anderen Zweck und ist deshalb mit anderen Mikroprozessoren kaum vergleichbar. Es richtet sich an Computeranfänger, die die Grundbegriffe der Schaltungstechnik und der Programmierung kennenlernen wollen. Die Schaltung wurde auf ein Mindestmaß zurückgeführt, damit der Einsteiger nicht durch raffinierte Schaltungstricks verwirrt wird und die Schaltung für ihn durchschaubar bleibt. Die Aufteilung der Schaltung auf die Platinen zerlegt das System in die grundlegenden Funktionsbaugruppen eines jeden Computersystems, d. h. Programmspeicher, Anzeige, zentrale Datenverarbeitungseinheit und Bedienungselemente (vgl. 3.).

Diese Aufteilung, zusammen mit der großzügigen Gestaltung der Platinenvorlagen, ermöglicht es, dieses Gerät selbst aufzubauen und auf Fehler zu untersuchen. Dies ist gleichzeitig ein Einstieg in die allgemeine Elektronik bzw. Digitaltechnik. Die schwer durchschaubare Fülle von einigen hundert verschiedenen Anweisungen im Befehlssatz

der üblichen 8-Bit-Prozessoren wird hier vermieden und durch 16 leicht verständliche Befehle ersetzt. Sie erleichtern das Erlernen der Grundprinzipien der Programmierung eines Computers. Wer dieses System durchschaut hat, der wird sich leicht auf einen umfangreicheren Befehlssatz umstellen können.

2.3 Eigenschaften der CPU

Der wichtigste Baustein in einem Computersystem ist die zentrale Datenverarbeitungseinheit (engl. Central Processing Unit – CPU). Die CPU C14500 ist erheblich einfacher aufgebaut als ihre 8-Bit-Kollegen. Sie hat nur 16 Anschlußbeine (Pins). Abb. 6 zeigt die Pinbelegung.

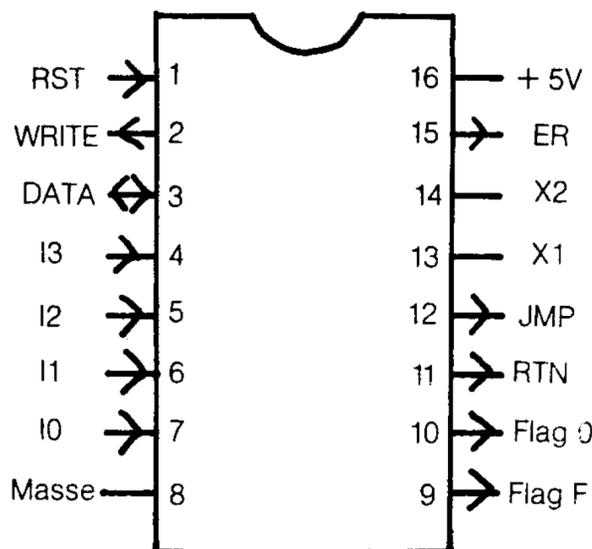


Abb. 6: Pinbelegung des IC MC 14500

Bedeutung der einzelnen Beine:

Pin 1: RESET Normalerweise ist dieses Bein über einen Widerstand mit 0V verbunden. Verbindet man es aber über einen Taster mit 5V, dann werden alle Informationen in der CPU gelöscht, und sie geht in den Ruhestand über (RESET-Taster).

Pin 2: WRITE „Schreiben“: Ausgangsleitung, die angibt, ob die CPU gerade eine Information über ihre Datenleitung herinholt oder ob sie über diese Leitung Daten ausgibt. Im ersten Fall ist die Spannung an diesem Bein 0V, während bei der Ausgabe 5V anliegen.

Pin 3: DATA Datenleitung: s. o., über die die Informationen sowohl beim Lesen als auch beim Schreiben von den bzw. zu den angeschlossenen Bausteinen des Computers geleitet werden. Da dieser Vorgang auf einer Leitung in beide Richtungen erfolgen kann, hat die Leitung auch den Namen „Bidirektionaler Datenbus“.

Pin 4: 13 Anschluß von Leitungen, die die Programmbefehle vom Programmspeicher an die CPU heranzuführen.
Pin 5: 12
Pin 6: 11
Pin 7: 10
Pin 8: Masse Masseanschluß der CPU.
Pin 9: FLAG F Ausgabe von 5 V-Impulsen aus der CPU, wenn der Befehl „NOPF“ (vgl. 2.5) ausgeführt wird. Sonst liegen hier 0V an.

Pin 10: FLAG 0 Hier tritt ein Impuls beim „NOP0“-Befehl auf (vgl. 2.5).

Pin 11: RTN Hier tritt ein Impuls beim „RTN“-Befehl auf (vgl. 2.5). Dieses Bein ist im System nicht beschaltet. Der Befehl ist auch nicht vorgesehen. Er kann für Systemerweiterungen verwendet werden.

Pin 12: JMP Hier tritt ein Impuls beim „JMP“-Befehl auf. Auch dieser Befehl ist nicht im System vorgesehen. Er kann für Systemerweiterungen verwendet werden.

Pin 13: X1
Pin 14: X2 An diese beiden Beine wird eine Widerstandskombination angeschlossen (1 MOhm, 56 kOhm), mit der die CPU den schnellen Arbeitstakt erzeugt.

Pin 15: ER Ausgang, an den das Ergebnisregister ER der CPU geführt ist. Im Ergebnisregister ER steht immer das Ergebnis, das durch die Ausführung eines Programmbefehls gewonnen wird. Dieses Bein ist mit der Eingangsleitung I0 verbunden, damit die Information des Ergebnisregisters wieder gelesen werden kann.

Pin 16: Betriebsspannung.

2.4 Innenleben der CPU:

Die Elektronik innerhalb der CPU besteht aus folgenden Teilen (Abb. 7).

1. Taktoszillator, der den schnellen Systemtakt (ca. 1 MHz) erzeugt.
2. Rechenwerk, in dem die Programmbefehle ausgeführt und die Informationen, die über die Datenleitung kommen, verarbeitet werden.
3. Ergebnisregister (ER), in dem die Rechenergebnisse gespeichert werden.
4. Befehlsdecodierung, sie unterscheidet und erkennt die 16 möglichen Programmbefehle und veranlaßt das Rechenwerk zur entsprechenden Befehlsausführung, bzw. sie steuert die Richtung der Datenleitung und die „Write“-Leitung und erzeugt die Impulse an den Beinen 9 bis 12.

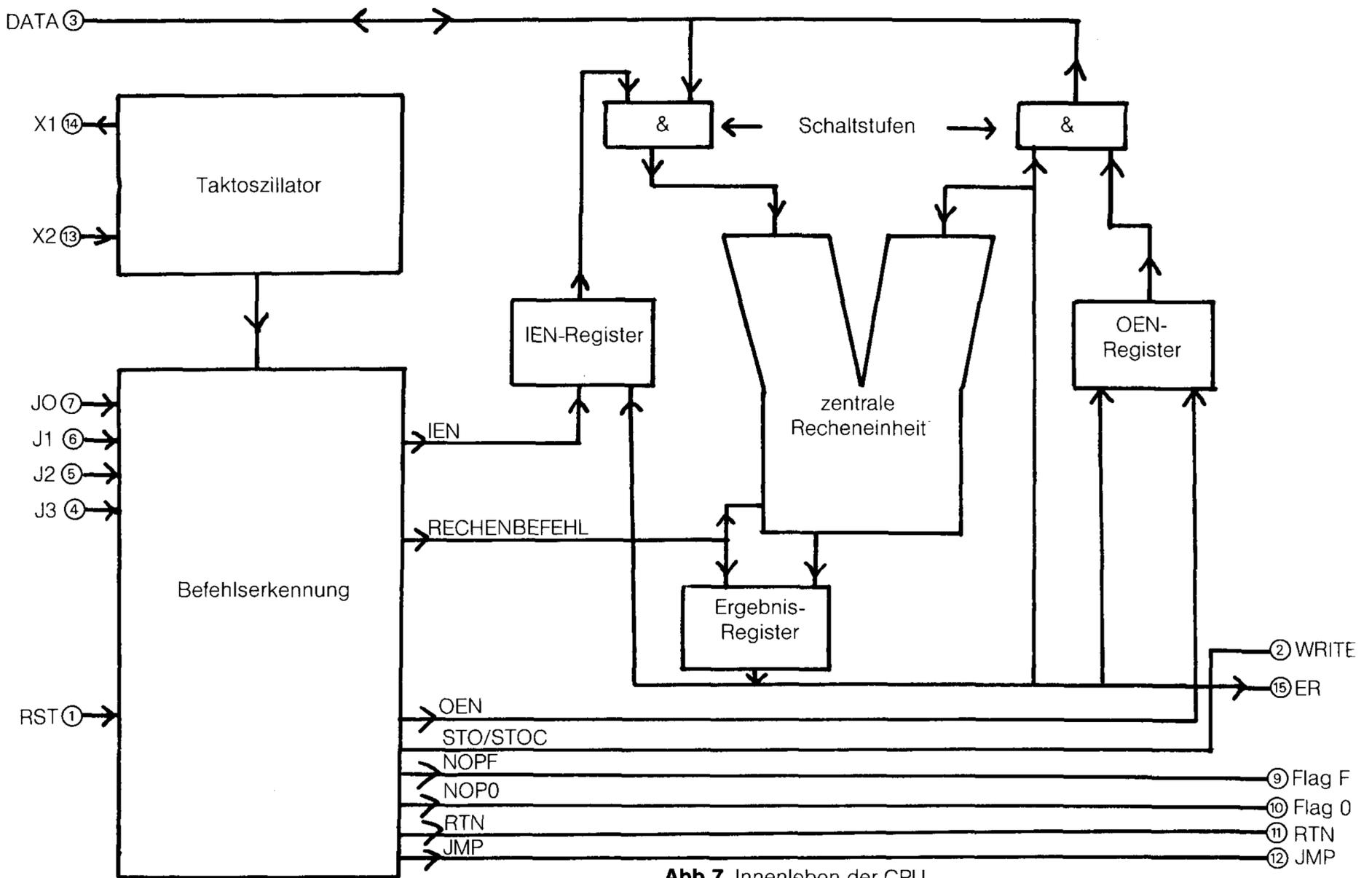


Abb 7. Innenleben der CPU

5. IEN-Register (Input ENable-Register (engl.)/Eingangs-Freigabe-Register). Der Zustand dieses Flip-Flops (vgl. 3.2) kann von der Recheneinheit gesetzt werden. Über die nachfolgende Schaltstufe kann das Einlesen von Informationen gesperrt werden.

6. OEN-Register (Output ENable-Register (engl.)/Ausgangs-Freigabe-Register). Dieses Flip-Flop arbeitet wie das IEN-Register, nur daß die Ausgabe von Informationen gesperrt werden kann.

2.5 Befehlssatz der CPU:

Die CPU kann 16 verschiedene Programmbefehle ausführen. Jeder hat eine Nummer, die durch eine 4stellige Zahl im Zweiersystem codiert ist (vgl. 3.2).

Befehls-

code	Name	CPU-Tätigkeit
0000	NOPO	„No Operation 0“ Die CPU gibt nur einen Impuls auf Pin 10 ab. Sonst geschieht nichts. Da Pin 10 nicht beschaltet ist, dient der Befehl nur dazu, in das Programm Lücken bzw. Pausen einzubauen.
0001	LD	„Load“ Die CPU holt („lädt“) eine In-

0010 LDC
„Load Complement“

0011 AND
Logische UND-Funktion

0100 ANDC
Logische UND-Funktion mit dem Gegenteil

0101 OR
Logische ODER-Funktion

0110 ORC
Logische ODER-Funktion mit dem Gegenteil

formation über die Datenleitung und speichert sie im Ergebnisregister ab.

Die CPU holt wieder eine Information, speichert aber das Gegenteil davon im Ergebnisregister. D. h., wenn auf der Datenleitung eine „1“ liegt, enthält das Ergebnisregister hinterher eine „0“ und umgekehrt.

Es wird die UND-Funktion gebildet aus dem Inhalt des Ergebnisregisters ER und der Information auf der Datenleitung. Das Ergebnis steht dann im Ergebnisregister. Hier steht eine „1“, wenn vorher im ER und auf der Datenleitung eine „1“ gestanden haben.

Es wird die UND-Funktion gebildet mit dem Inhalt des Ergebnisregisters und dem Gegenteil der Datenleitung. D. h., das Ergebnisregister ist nur dann „1“, wenn vorher im ER „1“ und auf der Datenleitung „0“ gestanden hat.

0101 OR
0110 ORC

0111 XNOR

1000 STO

1001 STOC

Logische ODER-Funktion
Die ODER-Funktion ergibt dann eine „1“ im ER, wenn vorher mindestens auf einer Leitung, ER oder Datenleitung, eine „1“ gestanden hat.
Logische ODER-Funktion mit dem Gegenteil der Datenleitung
Im ER steht jetzt eine „1“, wenn schon vorher im ER eine „1“ gestanden hat oder auf der Datenleitung eine „0“ war.

Logische Gleichwertigkeit („Äquivalenzfunktion“)
ER wird nur dann „1“, wenn vorher ER und Datenleitung gleich waren, d. h. entweder beide „0“ oder beide „1“.

„STORE“-Ausgabebefehl
Die CPU gibt die Information aus dem Ergebnisregister über einen der Ausgabekanäle aus.
„STORE Complement“-Ausgabebefehl für das Gegenteil
Die CPU gibt das Gegenteil des ER über die Datenleitung aus, d. h., wenn im ER eine „1“ steht, gibt sie eine „0“ aus bzw. umgekehrt.

1010 IEN „Input ENable“
Falls im ER eine „1“ steht, wird über das IEN-Register die Eingabe von Daten über die Datenleitung freigegeben. Falls ER „0“ enthält, wird die Eingabe gesperrt. Es werden dann keine Eingabebefehle ausgeführt.

1011 OEN „Output ENable“
Falls im ER eine „1“ steht, wird über das OEN-Register die Ausgabe von Informationen über die Datenleitung freigegeben. Falls ER „0“ enthält, wird die Ausgabe gesperrt. Es werden keine Aus-

gabebefehle mehr ausgeführt.

1100 JMP Der Befehl gibt nur einen Impuls an Pin 12 ab. Da Pin 12 nicht beschaltet ist, ist dieser Befehl im vorliegenden System nicht vorgesehen.

1101 RTN Der Befehl gibt nur einen Impuls an Pin 11 ab. Auch dieser Pin ist nicht beschaltet, und der Befehl ist nicht vorgesehen.

1110 SKZ „SKip on Zero“
Falls im ER eine „1“ steht, führt die CPU den nächsten Befehl normal aus. Falls im ER eine „0“ steht, führt sie

1111 NOPF „No OPeration F“
Bei diesem Befehl gibt die CPU auf Pin 9 einen Impuls aus, sonst geschieht nichts innerhalb der CPU. Pin 9 ist aber mit dem „Preset“-Eingang der Programmzähler IS 4029 verbunden. Der Impuls setzt also den Programmzähler (vgl. 3.2) wieder auf den Programmanfang.

(Fortsetzung folgt)

Akustische Spannungsanzeige

von Dr. Hans-Hellmuth Cuno (DL2CH), Waldetzenberg, Birkenstraße 11, 8411 Laaber, Telefon 0 94 98 / 13 17

Für einen stark sehbehinderten Amateur wurde das beschriebene Gerät als Abstimmhilfe für die Röhrendstufe seines Transceivers aufgebaut. Das Gerät gibt einen Ton ab, dessen Frequenz linear mit der angelegten Spannung zunimmt (Spannungs-Frequenz-Wandler). Dies altbekannte Prinzip wurde in der Schaltung (Bild 1) so realisiert, daß man den Eingang des Geräts nur mit den Anschlüssen des wohl in jedem Transceiver vorhandenen Anzeigeinstruments verbinden muß. Damit ist der Eingriff in das vorhandene Funkgerät minimal und kann bei Verwendung von Miniatur-Abgreifklemmen sogar völlig vermieden werden. Alle anderen Bestandteile wie Platine, Batterie, Lautsprecher und Schalter befinden sich in einem kleinen Gehäuse, das in der Station leicht Unterschlupf findet. Natürlich kann der Anschluß auch am SWR-Meter, Leistungsmesser etc. erfolgen.

Das Gerät war ein vollständiger Erfolg. Der OM hat jetzt immer die korrekte Abstimmung und damit volle Ausgangsleistung. Andere Amateure mit ähnlichen Problemen äußerten in QSOs mit dem OM ihr Interesse an dem Gerät, so daß seine Schaltung hier veröffentlicht wird. Ein probeweiser Einsatz an der eigenen KW-Station mit Röhren-PA und Linearendstufe ergab eine wesentliche Erleichterung und Beschleunigung des Abstimmens. Das rührt daher, daß das Auge nicht auf das Anzeigeinstrument fixiert ist und daß man bereits kleinste Spannungen hört, von denen am Instrument noch nichts zu sehen ist. So kann das Gerät auch normal sehenden Amateuren gute Dienste leisten.

Die Schaltung enthält nur preiswerte Standard-Bauteile. Die Eingangsspannung wird über einen RC-Tiefpaß auf den Eingang des Operationsverstärkers TAB1453A geführt. (Bei Verwendung anderer Verstärker achte man darauf, daß der Gleichtaktbereich der Eingänge bis zur negativen Speisepannung reicht!) Der Ausgang des Verstärkers steuert zwei pnp-Transistoren an, welche als Konstantstromquellen wirken. Der

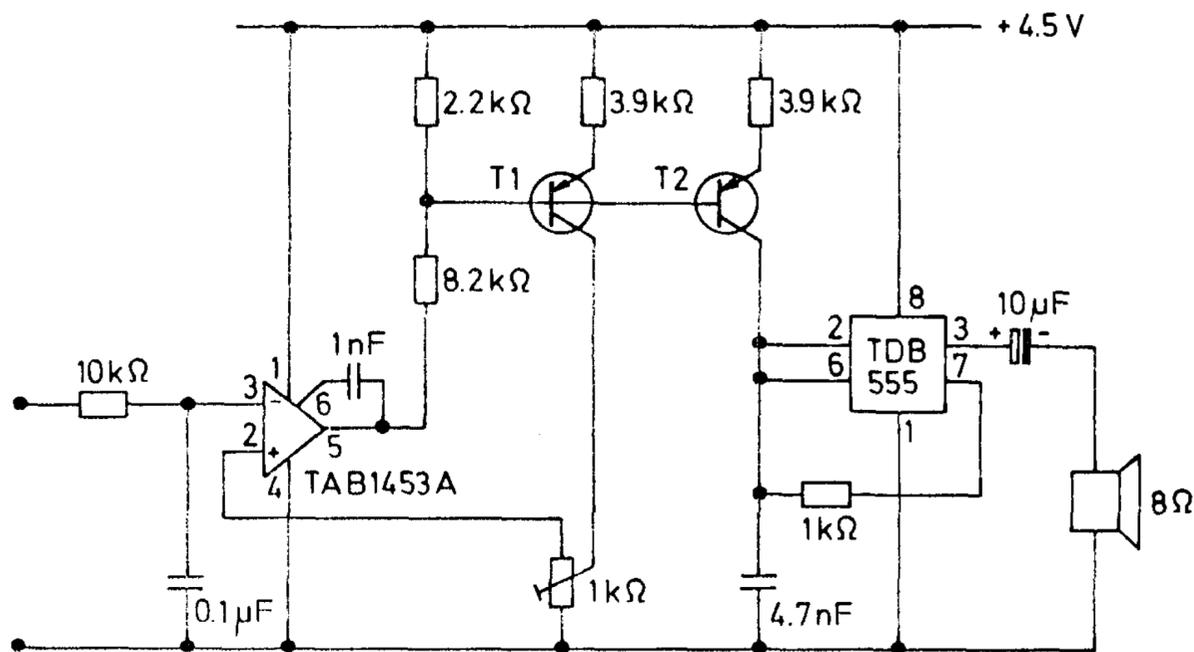


Abb. 1 Schaltung der akustischen Spannungsanzeige

Strom von T1 fließt in das Potentiometer, mit dessen Schleifer der Spannungsabfall und damit die Verstärkung der Eingangsstufe eingestellt wird. Der gleich große Strom durch T2 fließt in den Ladekondensator des Timer-IC TDB555. Beim Erreichen der oberen Schaltschwelle schaltet der IC und entlädt den Kondensator über den 1-kΩ-Vorwiderstand. Wenn die Spannung bis zur unteren Schaltschwelle abgesunken ist, schaltet der IC die Entladung wieder aus. Gleichzeitig wird am Ausgang ein rechteckförmiger Impuls abgegeben, der über den Kondensator den Lautsprecher ansteuert. Je höher die

Eingangsspannung ist, desto höher ist der Ladestrom des Kondensators, und desto schneller folgen die Entladungen aufeinander.

Der Eingangsstrom des Operationsverstärkers (wenige μA) ergibt einen geringen Spannungsabfall am Widerstand des Tiefpasses, der auch bei kurzgeschlossenem Eingang zur Erzeugung eines tiefen Tons führt. Die Schaltung hat dadurch keine Einsatzschwelle und reagiert bereits auf kleinste Spannungen.

Der Regelbereich des Potentiometers reicht für Eingangsspannungen bis ca. 90 mV aus. Das reicht für fast alle Drehspulinstrumente. Sollte die maximale Eingangsspannung höher liegen, so muß ein Spannungsteiler vor dem Eingang vorgesehen werden (Bild 2). Wegen des Eingangsstroms des Operationsverstärkers sollte R2 einen Wert von 10 k Ω nicht übersteigen. R1 wird dann für eine Eingangsspannung am Gerät von ca. 30 mV bemessen, die auf jeden Fall im Regelbereich des Potentiometers liegt.

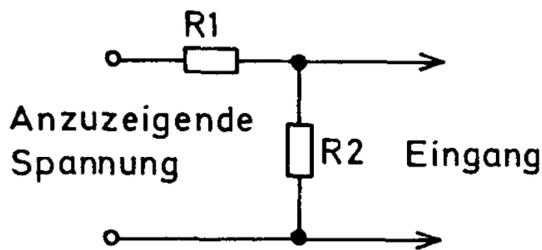


Abb. 2 Spannungsteiler für den Eingang

Ein 1-Bit-Lerncomputer zum Einstieg in die Mikrocomputertechnik (2)

Burkhard John, DK5JG, Zweibrückenstr. 35, 4000 Düsseldorf, 02 11/23 44 89 und Volker Ludwig, DDØEU, Am Alten Bach 14 a, 4040 Neuss 21, 02107/70226 als Sachbearbeiter „Amateurfunk in der Schule“ im Jugend- und Ausbildungsreferat des DARC

3. Beschreibung der Platinen

3.1. Anzeigeplatine JR-28

Es gibt einige Möglichkeiten, den Benutzer eines Computers erkennen zu lassen, was dieser ihm gerade anzeigt. Eine sehr bekannte, jedoch aufwendige Lösung ist ein Monitor oder Fernseher. Eine besonders einfache und zugleich preiswerte Lösung bieten Leuchtdioden, LED (Light Emitting Diode, engl. Licht aussendende Diode).

Das Schaltzeichen einer Leuchtdiode zeigt Abb. 8.

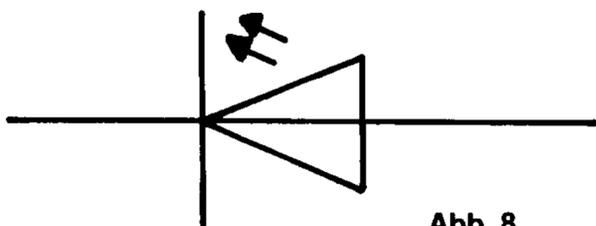


Abb. 8

Die Platine enthält 27 Leuchtdioden, die alle in diesem Mikroprozessorsystem wichtigen Signale anzeigen.

Abb. 9 gibt den entsprechenden Ausschnitt aus dem Gesamtschaltbild wieder.

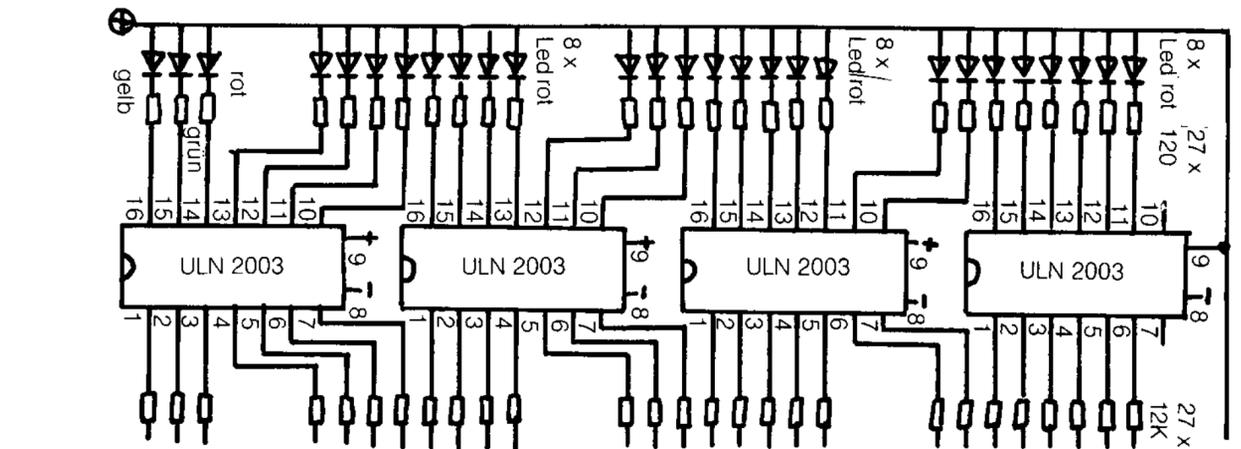
Das Prinzip der Ansteuerung der 27 Leuchtdioden durch die vier integrierten Schaltungen ULN 2003 läßt sich auf einen Verstärker pro Leuchtdiode zurückführen. Vor den Eingang ist ein 12-kOhm-Widerstand geschaltet, damit der Steuerstrom möglichst klein wird. Der Verstärker liefert mehr Strom, als die Dioden vertragen. Deshalb begrenzt der 120-Ohm-Widerstand den Leuchtdiodenstrom. Die 31-Pol-Steckerleiste stellt die Verbindung zur Grundplatine her.

Bedeutung der LEDs, von links

1. Systemtakt
2. Ergebnisregister
3. Datenleitung
4. - 11. Eingänge 7 bis 0
12. - 19. Ausgänge 7 bis 0
20. - 23. Befehlscode im Programmspeicher
24. Ein- oder Ausgabebefehl
25. - 27. Nummer der Ein- oder Ausgabeleitung

3.2. Speicherplatine JR-26

Unter „Speicher“ versteht man z.B. eine elektronische Schaltung, die sich 1 Bit als „0“ oder „1“ in Form von 0 V oder 5 V merkt. Es gibt viele Speichermöglichkeiten, eine gängige ist das Flip-Flop. (Der Name deutet darauf hin, daß etwas kippt – to flip – oder zurückkippt – to flop back). Abb. 10 zeigt ein solches Flip-Flop als Setz-Rücksetz-Flip-Flop. Ist dieses Flip-Flop im Flip-Zustand, so zeigt das der Ausgang des Flip-Flop z.B. mit einer „1“ an.



Nur ein Signal an Eingang E2 kann die gespeicherte Information ändern.

Ein solches Flip-Flop benötigt somit zwei Eingänge, einen Ausgang sowie zwei Anschlüsse für die Stromversorgung. Durch einen Schaltungskniff, der hier nicht erläutert werden soll, gelingt es, beide Eingänge zu einem gemeinsamen Eingang zusammenzufassen. Benutzt man viele solcher „D-Flip-Flops“ als Speicher in einem IC, so hat das den Vorteil der Ersparnis von Leitungen. Allerdings wird zum Einschreiben (Write) von Informationen noch eine zusätzliche Leitung benötigt, die aber für alle Speicher gemeinsam ist. Würde man vier solcher Speicher parallel schalten, erhielte man einen 4-Bit-Speicher, der 16 verschiedene Zustände speichern kann. Dies kann man mathematisch mit Hilfe des Zweiersystems erfassen.

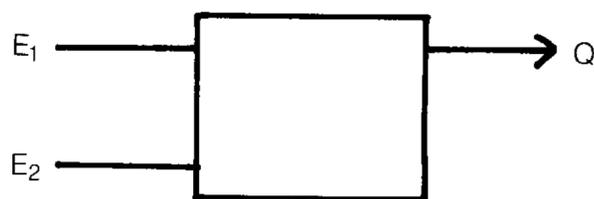


Abb. 10

Die folgende Gegenüberstellung hilft bei der Übersetzung des Zweiersystems in das geläufigere Zehnersystem. Vier Flip-Flops sind hintereinandergeschaltet, von rechts nach links gelesen ergibt sich ihre Stelle.

Stelle des Flip-Flops	4	3	2	1
Wert im Zweiersystem	2 ³	2 ²	2 ¹	2 ⁰
Wert im Zehnersystem	8	4	2	1

Stehen alle Flip-Flops auf 1, ist die Summe der Werte im Zehnersystem 8 + 4 + 2 + 1 = 15. Von 0 bis 15 sind 16 Zustände, da die Flip-Flops folgende Kombinationen eingehen können:

0	0000		
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001
5	0101	10	1010
		11	1011
		12	1100
		13	1101
		14	1110
		15	1111

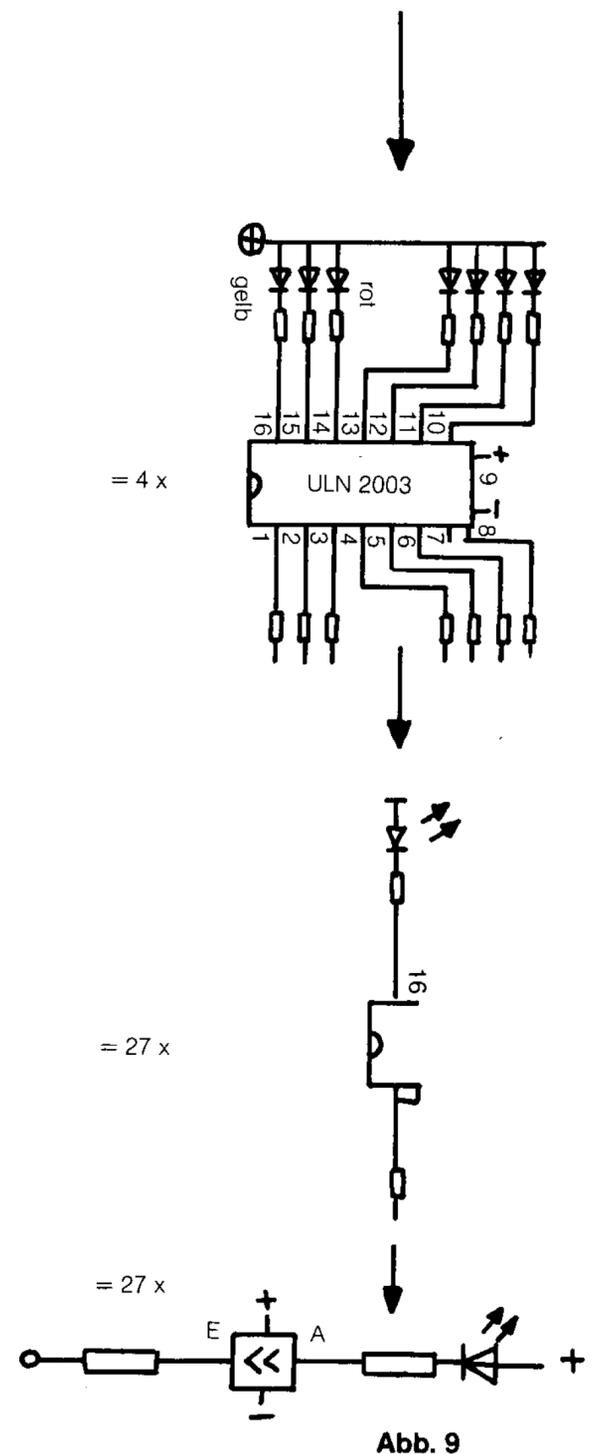


Abb. 9

Aufgrund der vorangegangenen Beschreibung ist es leicht einzusehen, daß, unter Zugrundelegung von zwei Leitungen für die Stromversorgung, 4 Leitungen für die Eingänge, eine Leitung für „Write“ und 4 Lei-

tungen für die Ausgänge, ein solcher Speicher in einem Gehäuse mit z.B. 16 Beinen untergebracht werden kann. Daraus folgt, daß man für jeden Speicher zwei Anschlüsse, einmalig drei Anschlüsse für die Write-Leitung und für die Stromversorgung benötigt. Im Falle eines 4-Bit-Speichers also $2 \times 4 + 3 = 11$ Anschlüsse, im Falle eines 256-Bit-Speichers $2 \times 256 + 3 = 515$ Anschlüsse.

Das Speicher-IC 2112 ist ein 256 x 4-Bit-Speicher.

Nach obiger Rechnung benötigt dieses Speicher-IC $2 \times 4 \times 256 + 3 = 2051$ Anschlüsse! Wie gelingt es aber, mit Hilfe von 16 Beinen 4 x 256 Speicherplätze ein- bzw. auszulesen?

Mit Hilfe von 8 Adreßeingängen können 256 Speicherplätze angewählt werden:

Wie in der obigen Gegenüberstellung erläutert die folgende Gegenüberstellung die mathematische Zuordnung:

Stelle des Adreßeingangs	8	7	6	5	4	3	2	1
Wert im Zweiersystem	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Wert im Zehnersystem	128	64	32	16	8	4	2	1
Die Summe der Werte im Zehnersystem	$128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$							

Zählt man von 0 bis 255, erhält man 256 mögliche Zustände.

Würde man ein solches System in ein Gehäuse mit 16 Beinen integrieren, benötigt man mit 8 Adreßleitungen, 4 Aus- und Eingängen sowie 3 Leitungen für die Stromversorgung und Write 19 Leitungen. Durch einen weiteren Schaltungskniff, der hier wiederum nicht erläutert werden soll, gelingt es, die Ausgänge auch als Eingänge zu benutzen. Legt man den Write-Eingang auf „0“, so wirkt der Anschluß als Eingang, liegt der Write-Eingang auf „1“, so wirkt der Anschluß als Ausgang. Abb. 11 zeigt die Pinbelegung des Speichers 2112. Da 4 Ein-/Ausgänge vorhanden sind, können 4 x 256 Bit als Speicher unterschieden werden.

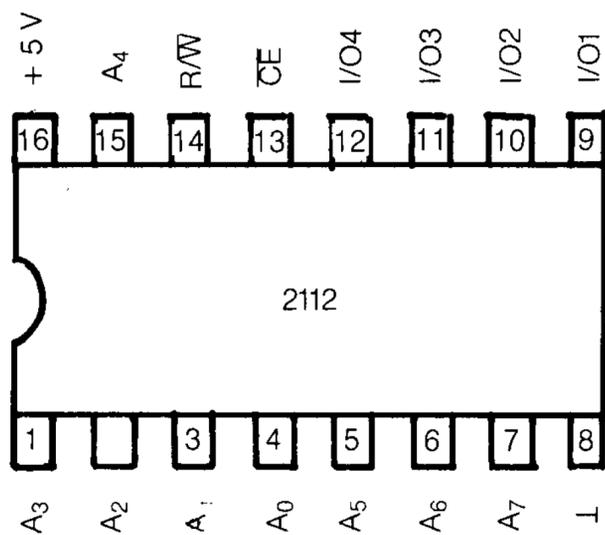


Abb. 11

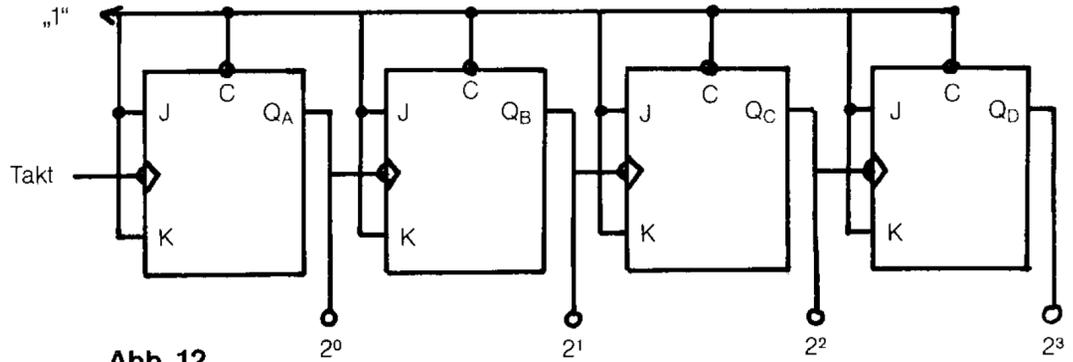


Abb. 12

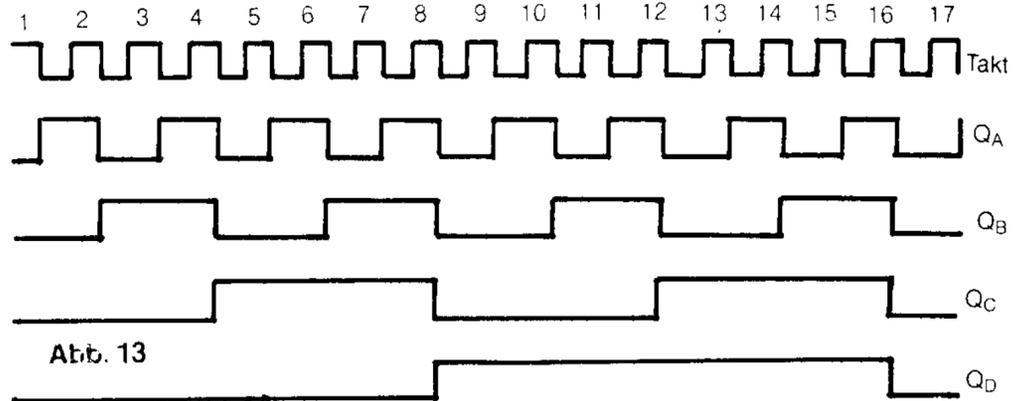


Abb. 13

Der 4-Bit-Binärzähler CD 4029 soll die Ansteuerung der 8 Adreßeingänge des Speichers übernehmen.

Um jeden der 256 Speicher des 2112 nacheinander anwählen zu können, muß man im Zweiersystem bis 256 zählen können. Hierfür benötigt man einen 8-Bit-Binärzähler. Im System sind dies zwei hintereinandergeschaltete 4-Bit-Binärzähler. Für einen 4-Bit-Zähler benötigt man 4 spezielle J-K-Flip-Flops, die hintereinandergeschaltet sind. Ein solcher Zähler ist in Abb. 12 gezeigt.

In Ausgangsstellung (Reset) sind alle Zähler-Flip-Flops „0“. Der 1. Taktimpuls auf den Eingang läßt das 1. Flip-Flop kippen, am Ausgang erscheint eine „1“. Bei einem 2. Taktimpuls kippt das 1. Flip-Flop zurück und setzt dadurch das 2. Flip-Flop. Bei einem 3. Taktimpuls kippt wieder das 1. Flip-Flop, das 2. Flip-Flop bleibt jedoch gesetzt.

Bei einem 4. Taktimpuls wird das 1. Flip-Flop zurückgesetzt. Dadurch wird das 2. Flip-Flop zurückgesetzt und hierdurch wird das 3. Flip-Flop gesetzt. Bei einem 5. Taktimpuls ... Abb. 13 gibt das dazugehörige Impulsdigramm wieder.

Der 4-Bit-Binärzähler CD 4029 ist programmierbar.

Neben seinen Ausgängen Qa, Qb, Qc und Qd enthält der CD 4029 4 „Data“-Eingänge A, B, C, D (vgl. Abb. 14), die es z.B. mit Hilfe von Schaltern erlauben, binär jeden beliebigen Zählerstand vorzuwählen, ab dem dann der Zähler weiterzählt. Dies ist in Abb. 15 durch jeweils 4 Schalter der Programmadressenschalter möglich.

Zusammenfassung unter Zugrundelegung der Abb. 16:

Die Speicherplatine enthält zum einen den Programmzähler, der es erlaubt, die Programmschritte in der richtigen Reihenfolge

ablaufen zu lassen. Er besteht aus zwei Bausteinen CD 4029. Dies sind setzbare Vor- oder Rückwärtszähler mit je 4 Bit, hier mit Hilfe von Pin 10 als Vorwärtszähler geschaltet. Sie bilden zusammen einen 8-Bit-Zähler, dessen Anfangsstand mit dem Adreßwahlschalter und dem „Preset“-Taster auf der Grundplatte voreingestellt werden kann. Zum zweiten enthält sie den Programmspeicher. Es sind zwei Schreib-Lesespeicher vom

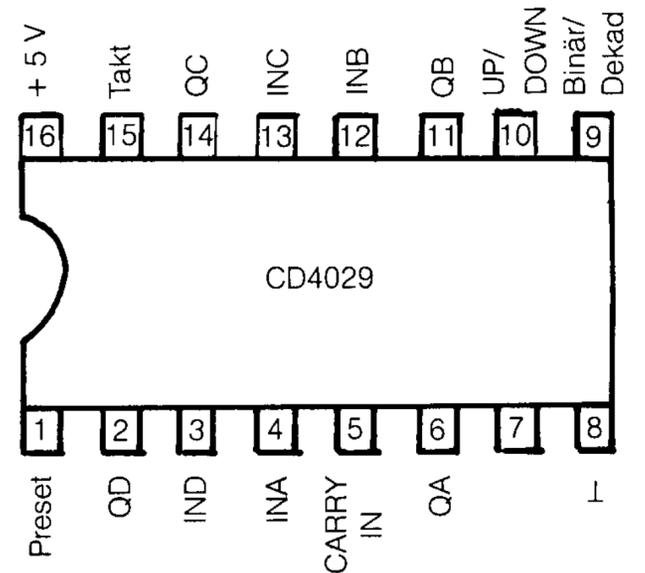


Abb. 14

Typ 2112 mit einer Speicherkapazität von je 256 mal 4 Bit. Ihre Adreßleitungen sind mit den 8 Zählerausgängen verbunden.

Ein 2112 speichert den Befehlscode, der andere die Ein-/Ausgabeadresse, eingestellt durch die Befehl-, I/O- und Adreßschalter.

Die Widerstände sind „Pull up“-Widerstände für alle Leitungen, d.h., sie verbinden die entsprechenden IC-Beine mit 5 V, um Störimpulse zu unterdrücken, wenn die Leitungen nicht gebraucht werden.

Die Verbindung zur Grundplatte stellt eine 21polige Steckerleiste her.

3.3 CPU-Platine JR-25

Sie enthält fünf integrierte Schaltungen, zwei Dioden und sieben Widerstände. Abb. 17 zeigt die Schaltung der CPU-Platine JR 25.

Die CPU MC14500 ist schon in Teil 2 beschrieben worden.

Die beiden Widerstände (1 MOhm, 56 kOhm) links daneben liegen parallel, wenn der Taktwahlschalter auf „schnell“ steht. Die CPU erzeugt dann mit dem eingebauten Taktgenerator den Schnelltakt mit ca. 1 MHz.

Die beiden Inverter mit den vier Widerständen (vgl. Abb. 18) und den beiden Dioden bilden ein S-R-Flip-Flop, dessen Eingänge über den „Takt“-Taster abwechselnd mit 5 V verbunden werden. Schon nach kürzester Berührung des Kontaktes springt das Flip-Flop um und behält seinen Zustand, auch wenn der Schalter noch nicht zuverlässig geschlossen ist. Damit ist das Prellen des Schalters unterdrückt. Wenn der Taktwahlschalter auf „Einzelschritt“ steht, ist Pin 13 der CPU mit dem Ausgang des Flip-Flops verbunden, und der CPU-Takt wird durch das Umschalten des Flip-Flops bestimmt.

Der MC14599 ist der Ausgabespeicher, der die 8 Ausgabeleuchtdioden steuert. Dieser Baustein verarbeitet nur dann die Daten auf der Datenleitung (Pin 3), wenn die CPU einen Ausgabebefehl (STO, STOC) ausführt. Dann ist die „Write“-Leitung (Pin 10) aktiv. Außerdem muß die Ein-/Ausgabeleitung des Programmspeichers „1“ sein, und der Takt muß „0“ sein. Die beiden letzten Bedingungen werden von der Schaltung aus den beiden Invertern und dem ODER-Gatter geprüft (Pin 8). Die Entscheidung, auf welche der 8 LED's (Pin 1, 11–17) die Daten ausgegeben werden, geschieht durch die drei Adreßleitungen (Pin 5, 6, 7).

Der CD4051 ist immer dann aktiv, wenn die CPU Daten über ihre Datenleitung hereinholt. In dem Augenblick ist der Takt „0“, die „Write“-Leitung ist „0“ und auch die Ein-/Ausgabeleitung des Programmspeichers. Diesen Zustand erkennt die Schaltung aus den beiden ODER-Gattern und gibt durch eine „0“ an Pin 6 die Schaltung frei. Die Auswahl der Eingangsleitungen (Pin 13, 14, 15, 12, 1, 5, 2, 4) geschieht über die drei Adreßleitungen (Pin 9, 10, 11) vom Programmspeicher. Die Ein- bzw. Ausgänge mit den Adressen 5, 6 und 7 sind

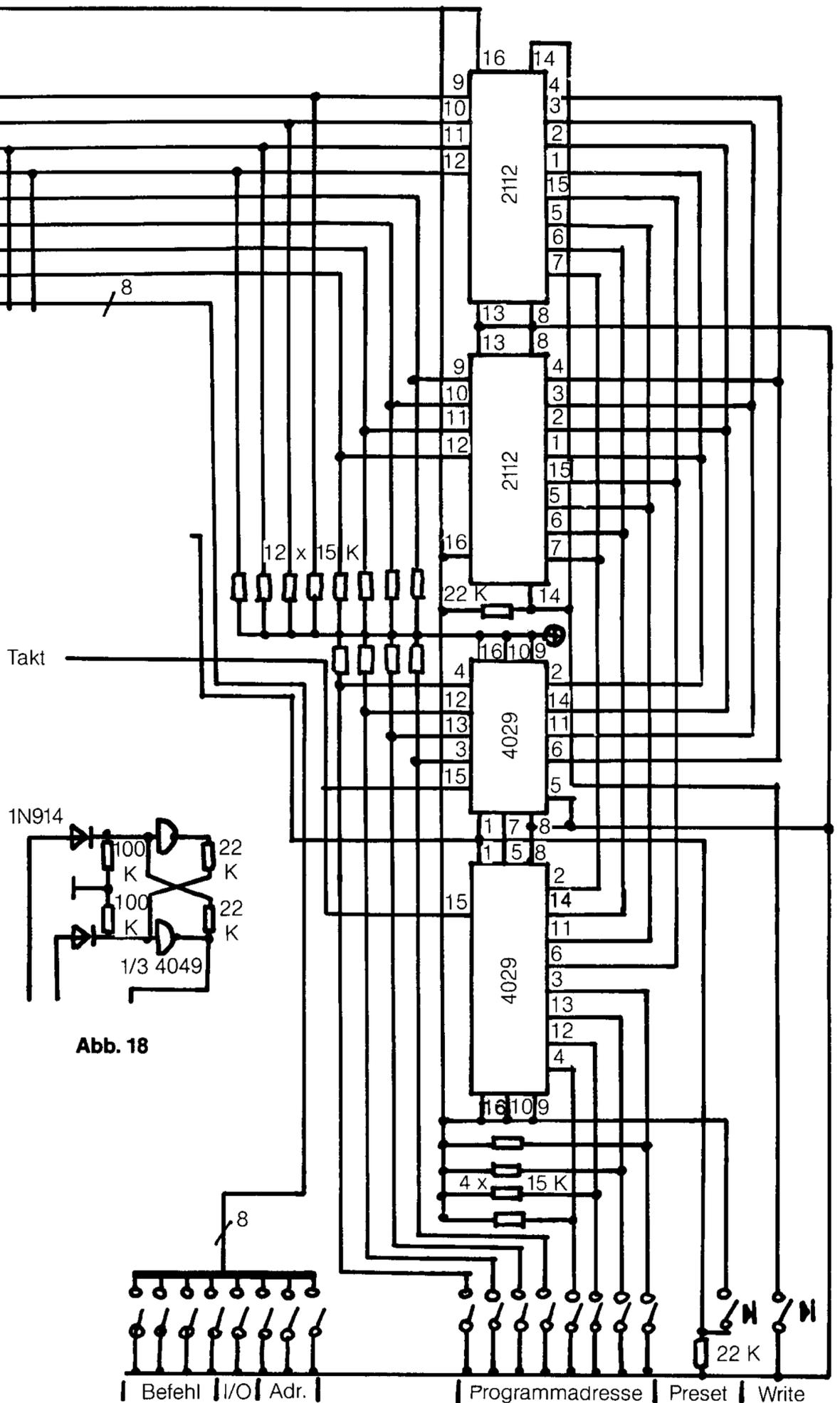


Abb. 18

Abb. 16

miteinander verbunden. Die Ausgänge können normal aber auch wieder von der CPU gelesen werden. Dadurch entstehen drei Datenspeicher. In gleicher Weise lassen sich weitere Datenspeicher am Expansionsstecker auf die Grundplatte schalten, denn die Eingänge 1 bis 4 sind auf die Grundplatte geführt. Der Eingang mit der Adresse 0 (Pin

13) ist mit dem ER-Pin der CPU verbunden, um das ER wieder lesen zu können.

Der CD4049 enthält 6 Inverter, von denen in der Schaltung 4 benutzt werden. Der CD4071 enthält 4 ODER-Schaltungen, von denen drei benutzt sind. Die Verbindung zur Grundplatte stellen eine 31polige und eine 13polige Steckerleiste her.

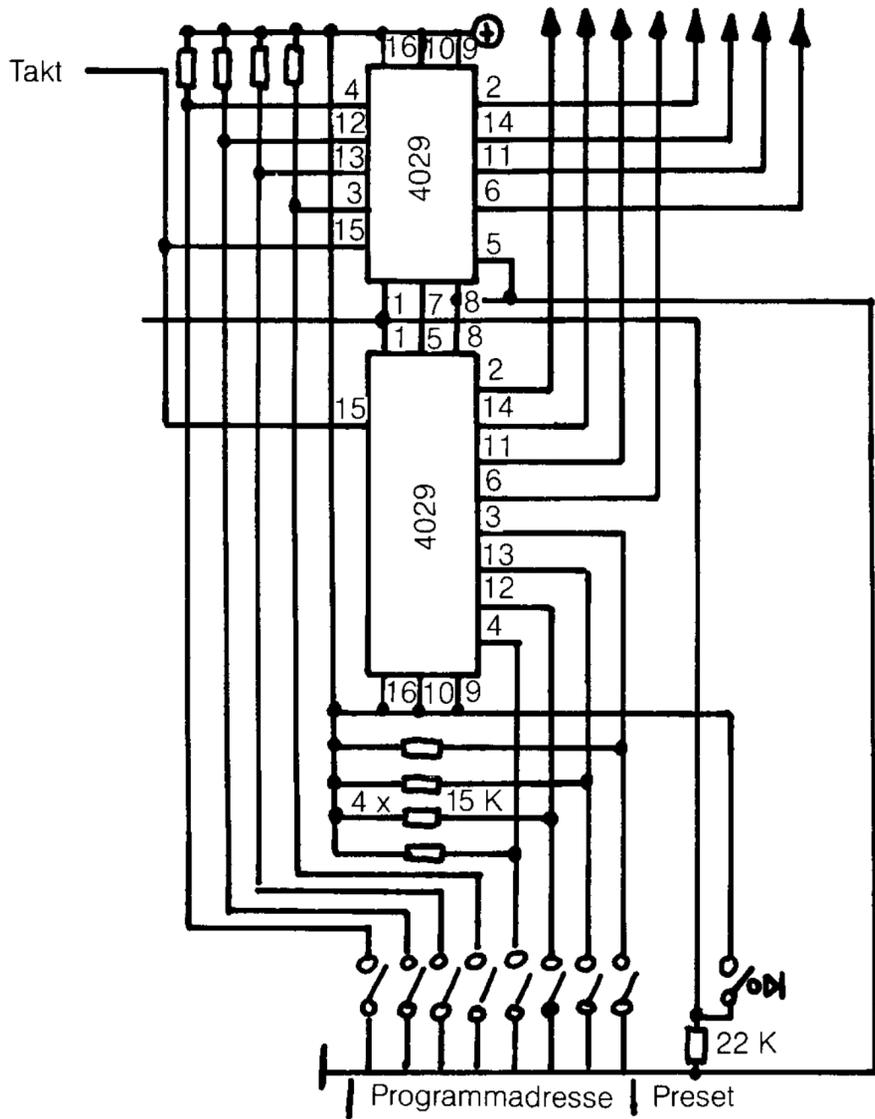


Abb. 15

3.4 Grundplatine JR-27

Sie enthält alle Verbindungsleitungen zwischen den vorbeschriebenen Platinen und trägt die entsprechenden Buchsenleisten. Eine Gleichrichterdiode und ein Spannungstabilisator 7805 bilden die Stromversorgung (vgl. Abb. 19).

Das System arbeitet dadurch mit 7 V bis 12 V Gleich- oder Wechselstrom. Der 7805 wird mit einem Blechwinkel gekühlt. Vorsicht, hier sind Kurzschlüsse zu anderen Platinen möglich. Das Timer-IC 555 ist als Rechteckgenerator geschaltet (vgl. Abb. 20). Der Ausgang des ICs ist über den 100-kOhm-Widerstand an der CPU angeschlossen. Dadurch ist die Ankoppelung schwach, so daß nur in Mittelstellung des Taktwahlschalters (Langsamtakt), wenn weder Handtakt noch Schnelltakt der CPU eingeschaltet sind, das Signal des 555 wirksam werden kann. Falls für einen speziellen Fall der Frequenzregelbereich des Potentiometers nicht ausreichen sollte, kann der Takt, durch Ändern des 10-Mikrofarad-Kondensators oder des Potis und des 120-kOhm-Widerstandes, beliebig angepaßt werden. Verkleinerung der Werte erhöht die Taktfrequenz. Der Expansionsstecker, links hinten, dient zum Anschluß weiterer Geräte, z.B. eines Tongenerators.

An der Vorderseite finden sich die Bedienungsschalter.

Von links:

- 8-Bit-DIL-Schalter zur Vorwahl der Adresse, siehe JR-26.
- 16-Pol-IC-Fassung für die Steckverbindung zu JR-29 oder DIL-Schalter zur Programmeingabe.
- 4-Bit-DIL-Schalter für die Eingangssignale 1 bis 4.
- „Write“-Taster zum Einschreiben eines Befehls in den Programmspeicher.
- „Takt“-Taster zum Weiterschalten des Programmzählers, „Handtakt“. Dies muß ein Um-Schalter sein!
- Taktwahlschalter. Schalter 1 mal Um mit Mittelstellung. Vorne: Schnellgang, ca. 1 MHz. Mittelstellung: langsamer Takt aus 555. Hinten: Handtakt.
- „Preset“-Taster zur Einstellung des Programmzählers auf die vorgewählte Adresse.
- „Reset“-Taster. Er bringt das System in den Ruhezustand.

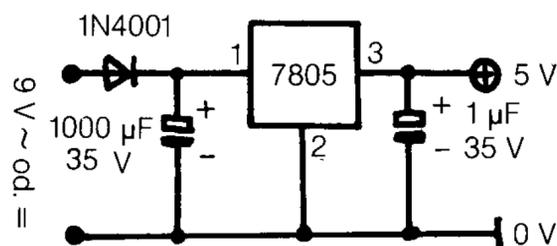


Abb. 19

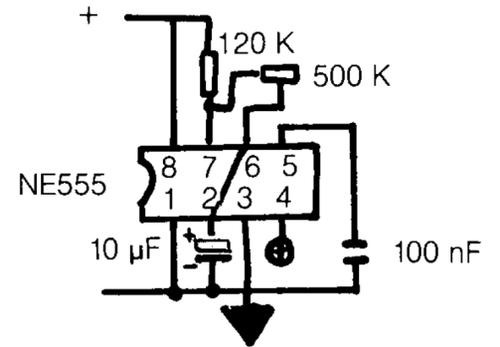


Abb. 20

3.5 Eingabetastatur JR-29

Sie enthält 14 Tasten für die Befehle und 8 Tasten für die Ein-/Ausgabeadressen. Der Befehlscode und die Adressen werden durch eine Diodenmatrix codiert. Jeder Programm-befehl besteht aus Zahlen im Zweiersystem, deren Stellen durch 0 V („0“) bzw. 5 V („1“) dargestellt werden. Zur Codierung ist es nur notwendig, die Stellen, die „0“ werden sollen, mit 0 V zu verbinden, denn durch „Pull up“-Widerstände auf der Speicherplatine sind alle Eingänge des Programmspeichers schon mit 5 V verbunden. Die Dioden sind nötig, um die verschiedenen Tasten zu entkoppeln. Abb. 21 gibt ein Beispiel für die Codierung des „IEN-Befehls“.

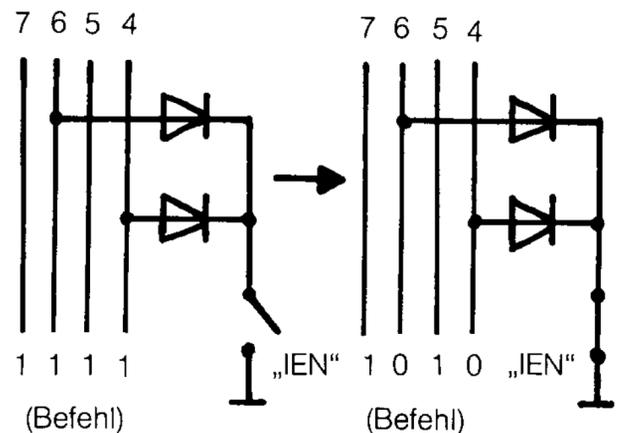


Abb. 21

Damit der Prozessor entscheiden kann, ob es sich um einen Ein- oder Ausgabebefehl handelt, muß für Bit 3 eine weitere Diode zugeschaltet werden, wenn es sich um einen Eingabebefehl handelt, damit dieses Bit auf „0“ gesetzt werden kann. Abb. 22 zeigt diese Codierung am Beispiel „IEN“.

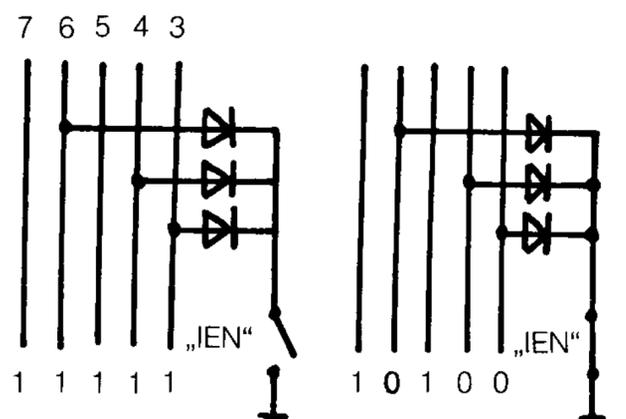


Abb. 22

Die vollständige Codierung mit Hilfe von Dioden für den Befehl „IEN 0“ zeigt Abb. 23.

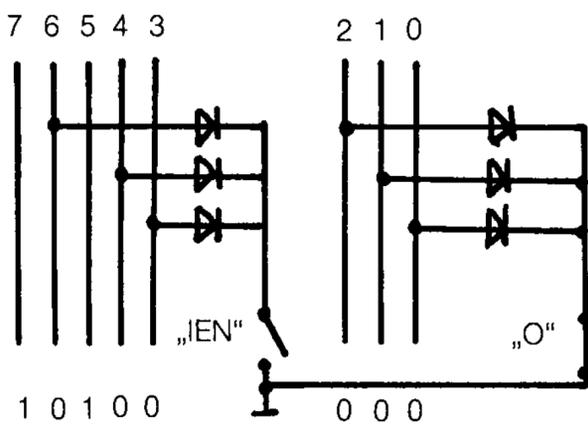


Abb. 23

Dies erklärt auch, warum auf der Platine der „NOPF“- und der „7“-Taster nicht beschaltet sind. In beiden Fällen sind alle Stellen „1“. Die Taster sind aber trotzdem vorhanden, um die Eingabe dieser Befehle zu erleichtern. Die Verbindung zur Grundplatine bildet ein kurzes Flachbandkabel mit angelöteter IC-Fassung. (Fortsetzung folgt)

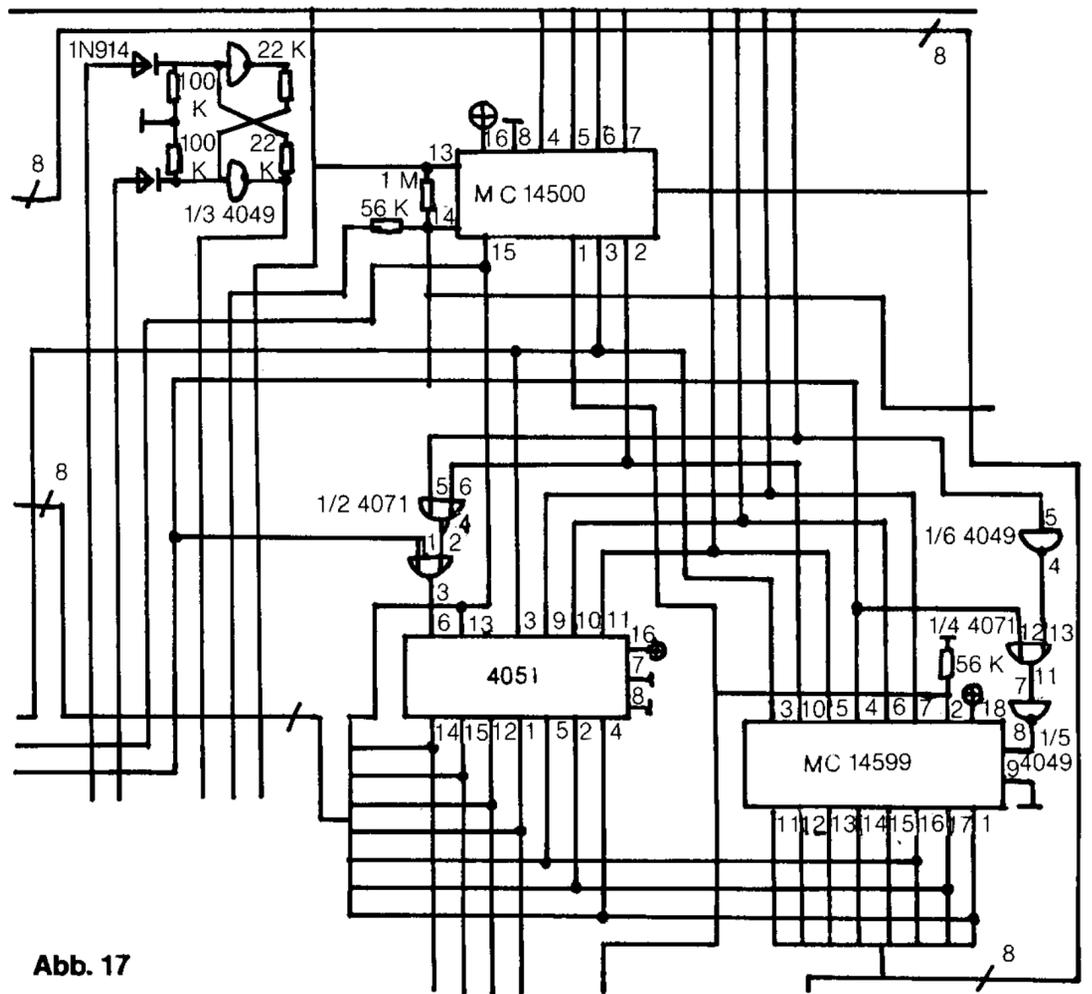


Abb. 17

Intermodulationsfester Preselektor für 1,5 – 30 MHz

Michael Martin, DJ7VY, Alsenstraße 6, 1000 Berlin 39, und Richard Waxweiler, DJ7VD, Florastraße 8, 1000 Berlin 41

Ein Preselektor für den Kurzwellenbereich kann als Zusatzgerät das Empfänger-Großsignalverhalten entscheidend verbessern. Vornehmlich die „modernen“ Amateur-Kurzwellenempfänger mit Breitbandeingängen benötigen eine abstimmbare Vorselektion, die selbst sehr intermodulationsarm sein muß. Nur so kann man verhindern, daß der in einem Band von mehreren MHz steckende ungeheure Wust von „dicken“ Signalen den oftmals zu schwach ausgelegten Empfänger-mischer übersteuert.

Allgemeines

Unbefriedigende Großsignaleigenschaften eines Empfängers lassen sich durch geeigneten Umbau des Eingangsteils verbessern. Das werden die meisten Gerätebesitzer aber nicht durchführen, weil es ihnen zu schwierig, zu zeitraubend, zu aufwendig oder mangels geeigneter Meßgeräte nicht möglich erscheint. Manche Hersteller versuchen das mangelnde Großsignalverhalten ihrer Produkte zu bemänteln, indem sie einen umschaltbaren Abschwächer in den Eingang des Empfängers legen. Ein 20-dB-Dämpfungsglied verbessert zwar den Interceptpunkt, das Maß für das Großsignalverhalten, um 20 dB, womit die Intermodulationsprodukte um 60 dB abnehmen, hat aber auch eine Absenkung der erwünschten Signale um 20 dB zur Folge. Eine Verbesserungsmöglichkeit, die den Empfänger unangeta-

stet läßt, ist die Vorschaltung eines schmalbandigen Preselektors, der sich als ein selektives Dämpfungsglied auffassen läßt. Erwünschte Signale dämpft er (fast) nicht, unerwünschte dagegen erheblich. Für diesen Vorteil muß man den Nachteil zusätzlicher Bedienungselemente in Kauf nehmen.

Schaltung

Es sind sehr viele Preselektor-Schaltungsvarianten denkbar und auch manche veröffentlicht worden [1, 2, 3, 4]. Das hier vorgestellte Modell zeichnet sich durch besondere Einfachheit und Intermodulationsfestigkeit aus und ist eine einkreisige, passive Tiefpaßschaltung (Abb. 1). Eingang und Ausgang stehen in einem Transformationsverhältnis von eins zu eins und besitzen eine Nennimpedanz von 50 Ω. Die Koppelkondensatoren

am Eingang dienen der Transformation von 50 Ω auf höhere Werte, wogegen jene am Ausgang in gleicher Weise wieder herabtransformieren. Sie sind in drei Stufen schaltbar ausgeführt (15, 37, 83 pF), um sich den Bedingungen des ganzen Abstimmbereichs besser anpassen zu können (Schalter 1). Für eine annähernd konstante Durchgangsdämpfung muß bei tiefer werdender Frequenz, also steigender Drehkondensator-kapazität, die Kapazität der Koppelkondensatoren zunehmen. Anderenfalls steigen die Durchgangsdämpfung und die Betriebsgüte des Schwingkreises zu sehr an, während jedoch auch die Weitabselektion und die Trennschärfe zunehmen. Der Schalter 1 beeinflusst diese miteinander verkoppelten Parameter. Abb. 2 zeigt die Dämpfungsverläufe für vier ausgewählte Amateur-bänder.

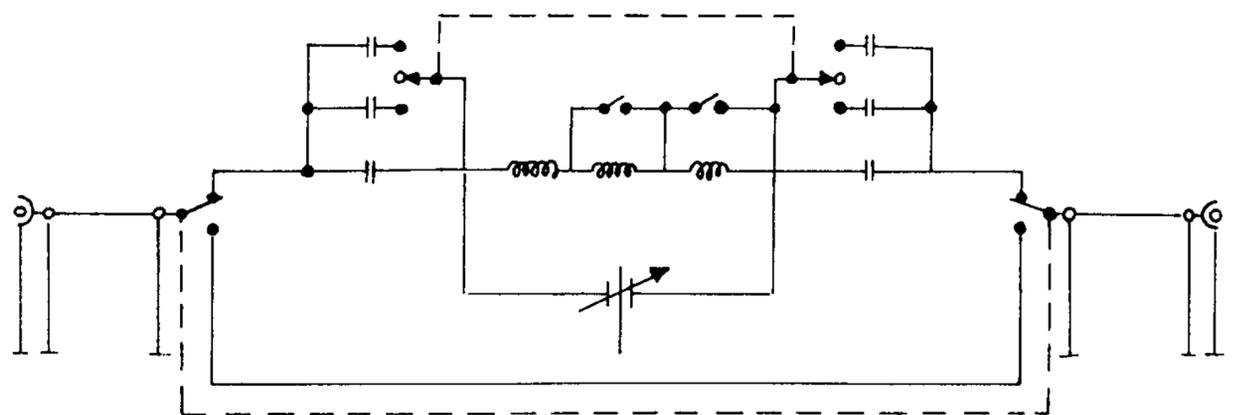


Abb. 1: Schaltung des intermodulationsfesten Preselektors 1,5–30 MHz

Ein 1-Bit-Lerncomputer zum Einstieg in die Mikrocomputertechnik (3)

Burkhard John, DK5JG, Zweibrückenstr. 35, 4000 Düsseldorf, 0211/23 44 89, und Volker Ludwig, DDØEU, Am Alten Bach 14a, 4040 Neuss 21, 0 21 07 / 7 02 26, als Sachbearbeiter „Amateurfunk in der Schule“ im Jugend- und Ausbildungsreferat des DARC.

4. Allgemeine Aufbauhinweise

Die Abb. 24 zeigt das Blockschaltbild des Minimalsystems, Abb. 25 gibt den Schaltplan des 1-Bit-Prozessors wieder. Die Verwirklichung des Schaltplans geschieht in modularer Form auf Europakarten.

Alle integrierten Schaltungen und DIL-Schalter werden auf Fassungen gesetzt. Die Buchsenleisten auf der Oberseite der Grundplatine sind so angelegt, daß die dazugehörigen Platinen nicht vertauscht werden können. Die 13polige Steckerleiste ist zum Hinzufügen peripherer Geräte gedacht, z.B. eines Tongenerators. Die Leiste wird auf die Oberseite der Platine gelötet.

Die 21polige Buchsenleiste nimmt die Speicherkarte JR 26 auf.

Die 31polige und zusätzlich 13polige Buchsenleisten tragen die CPU-Platine JR 25.

Auf die 31polige Buchsenleiste paßt die Anzeigeplatine JR 28.

Der Sockel des mittleren 8-Pol-DIL-Schalters nimmt wahlweise einen 8fach-DIL-Schalter oder den Verbindungsstecker der Verbindung zur Eingabeplatine JR 29 auf.

Die Platinen werden mit 1-mm-Bohrern gebohrt. Vor dem Bestücken der Bauteile sollten auf den doppelseitigen Platinen die Durchkontaktierungen gesetzt werden. Dazu eignen sich abgeschnittene Widerstandsbeine besonders gut, die auf beiden Seiten festgelötet werden. Die entsprechenden Punkte sind auf den Bestückungsplänen markiert. An einigen Stellen werden zur Durchkontaktierung gleich die Anschlußbeine der eingesetzten Bauteile benutzt.

Die Durchkontaktierungen sorgfältig prüfen, evtl. mit dem Ohmmeter, denn sie sind Fehlerquellen, die man später kaum noch findet. Häufige Fehler sind kalte Lötstellen oder Lötzinnbrücken.

Für das Poti zur Geschwindigkeitseinstellung ist kein Befestigungspunkt vorgesehen. Es wird direkt mit den Anschlüssen an die Punkte unter der Platine gelötet. Der Spannungsstabilisator sollte mit einem kleinen Kühlwinkel gekühlt werden. Vorsicht, Kurzschluß zu anderen Platinen ist möglich.

Es empfiehlt sich, vor dem Auswechseln einer Platine die Betriebsspannung abzuschalten.

Die meisten Widerstandswerte sind nicht kritisch und können auch durch ähnliche Werte ersetzt werden, z.B. 10 kOhm statt 12 kOhm.

Zur eventuellen Fehlersuche genügt ein normales Vielfach-Meßgerät.

Die Taster auf der Grundplatine haben Kunststoffstifte an der Unterseite. Sie sollten abgefeilt werden.

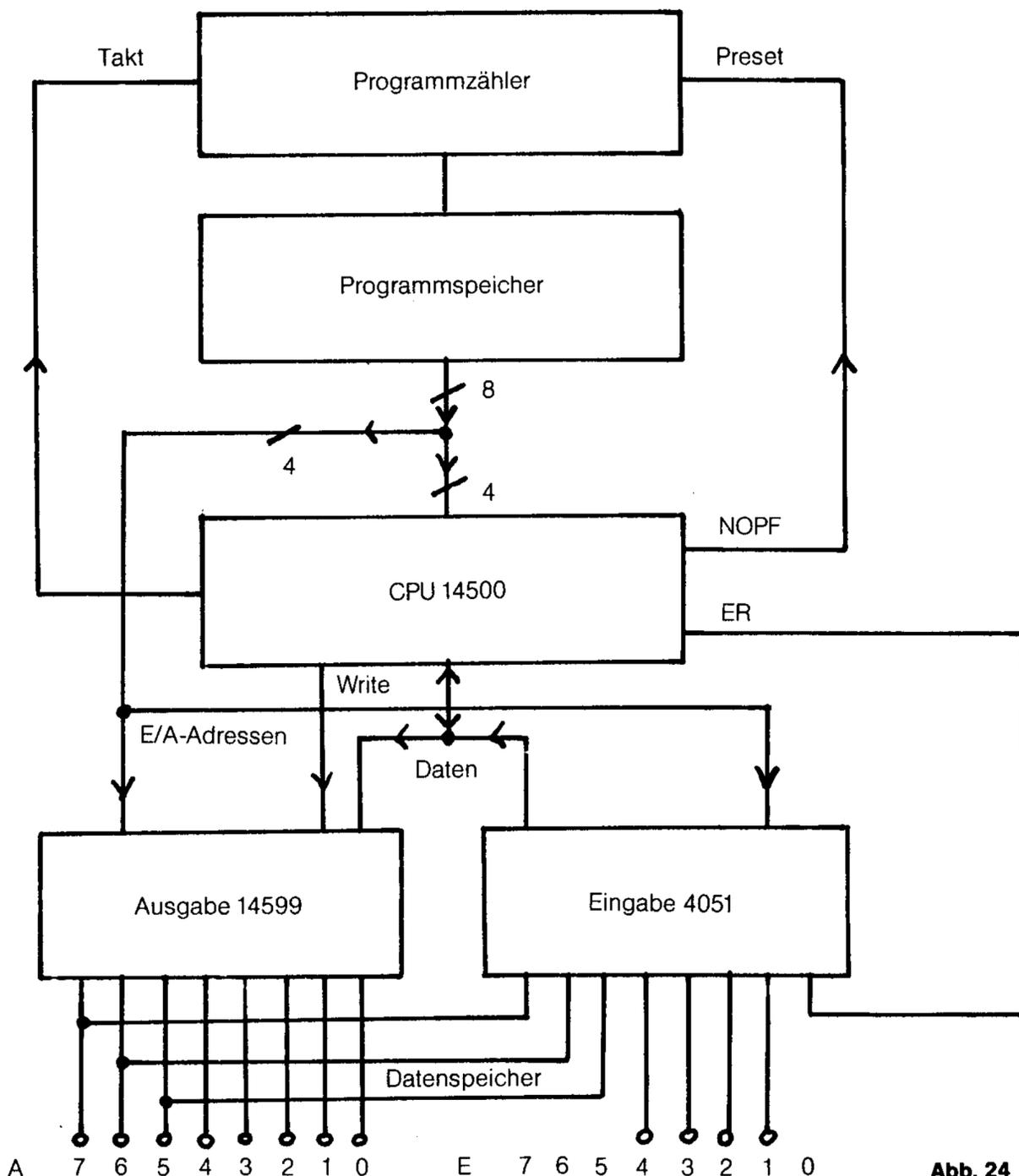


Abb. 24

5. Leitfaden für die Aufbauarbeiten

Um schon während des Aufbaues einige Funktionen prüfen zu können, empfiehlt sich folgende Reihenfolge bei der Arbeit.

5.1 Bestücken der Grundplatine JR-27

Anlegen der Betriebsspannung.

Kontrolle der 5 V Konstanterspannung an Pin 1 (+) und Pin 2 (-) jeweils an der 21poligen Federleiste für die Speicherplatine, den 31poligen Leisten für die CPU- und die Anzeigeplatine.

Überprüfen der Umschaltfunktion des „Takt“-Tasters mit dem Voltmeter.

5.2 Bestücken und Einsetzen der Anzeigeplatine in die Grundplatine.

Zur Kontrolle der Funktion der LEDs und Treiber-ICs:

Abgreifen der +5 V von Pin 1 der 21poligen Leiste und Heranführen dieser Spannung an die Anschlüsse 4 bis 15 und 17 bis 31 der Anzeigeplatine.

Nun müssen alle LEDs nacheinander leuchten.

Die LEDs der Eingänge 1 bis 4 können jetzt mit den vier Eingangs-DIL-Schaltern geschaltet werden.

5.3 Bestücken und Einsetzen der CPU-Platine.

Der Taktwahlschalter wird auf „Handtakt“ geschaltet. Die „Takt“-LED muß nun leuchten. Betätigt man nun den „Takt“-Taster mehrmals, dann sollte die „Takt“-LED entsprechend blinken. In Mittelstellung des Taktwahlschalters (Langsamtakt) blinkt die LED von selbst. Die Geschwindigkeit kann mit dem Poti auf der Grundplatine eingestellt werden. In Stellung „Schnelltakt“ wird die LED etwas dunkler.

5.4 Bestücken und Einsetzen der RAM-Platine.

In Stellung „Langsamtakt“ sollten einige Programm-LEDs abwechselnd leuchten.

5.5 Bestücken und Ankoppeln der Eingabetastatur an den Programmiersockel.

Alle Dioden sollten vor dem Aufbau auf

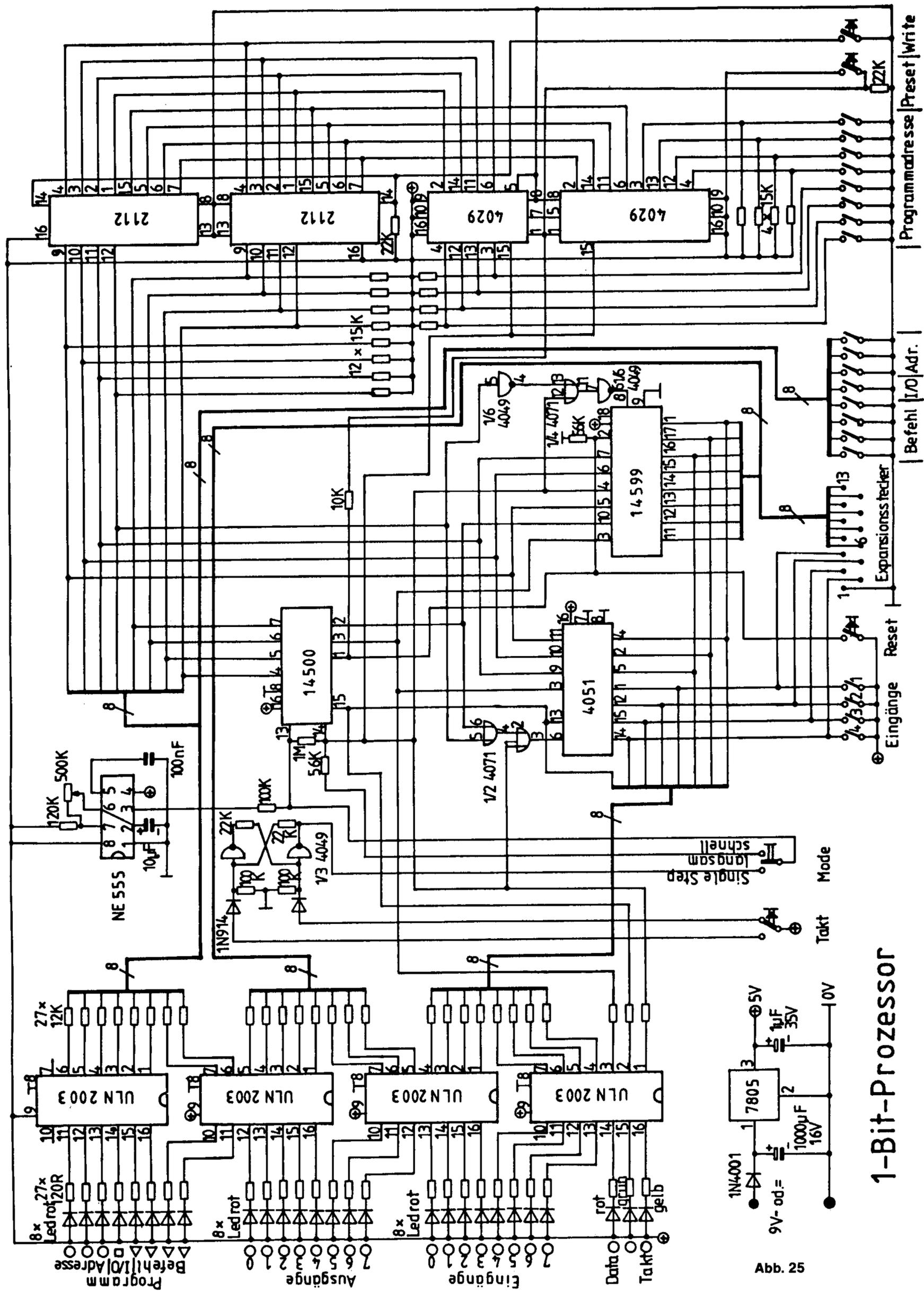


Abb. 25

1-Bit-Prozessor

ihre Funktion hin geprüft worden sein, mit einem Ohm-Meter, wichtig bei billigen Sonderangeboten!

Taktschalter auf „Handtakt“ stellen. Mit NOPO 0 und „Write“ müssen alle Programm-LEDs ausgehen. Mit NOPO 1 bis NOPO 7 und „Write“ zeigen die drei Adreß-LEDs rechts die Zahlen 1 bis 7 im binären Zahlensystem. Einmal „Preset“ betätigen. Während die Tasten NOPO 0 bis 7 gedrückt werden, jeweils einmal „Write“ und „Takt“ schalten. Nach einem neuen „Preset“ müssen durch mehrfache „Takt“-Betätigung die einprogrammierten Speicherinhalte wieder sichtbar werden. Bei Überprüfung der restlichen Befehlstaster zusammen mit „Write“, wie oben, muß das Bitmuster des entsprechenden Befehls auf den übrigen 5 Befehls-LEDs erscheinen.

5.6 Nach erfolgreicher Arbeit müßte der 1-Bit-Prozessor jetzt arbeitsfähig sein. Die weitere Prüfung bringt nun die Programmierung.

6. Die Programmierung des 1-Bit-Lerncomputers

Abb. 26 zeigt die Anordnung der Bedienelemente des 1-Bit-Lerncomputers. Um den Programmierer nicht durch zufällige Speicherinhalte, die auf den Ausgabe-LEDs erscheinen, zu verwirren, empfiehlt es sich, alle Speicher auf „0“ zu setzen, so daß alle LEDs erloschen sind:

1. Taktwahlschalter auf Stellung schnell stellen.
2. Betätigung von NOPO 0 und gleichzeitig „Write“-Taster (1-2 Sekunden).
3. Taktwahlschalter auf Handtakt zurück-schalten.

Ein Betätigen des Takttasters bestätigt nun, daß alle Speicher auf „0“ gesetzt sind. Mit dem Preset-Taster wird der Prozessor auf die Anfangsadresse 00000000 gesetzt.

Mit dem Reset-Taster werden alle Flip-Flops in der CPU zurückgesetzt.

6.1. Initialisierung heißt Vorbereitung der CPU zur Programmbearbeitung

Damit die CPU die Befehle, die sie bearbeiten soll, überhaupt ausführen kann, müs-

sen die Eingabe- und Ausgabe-Flip-Flops gesetzt werden, indem der IEN-Befehl (Input-ENable = Eingang freimachen) und der OEN-Befehl (Output-ENable = Ausgang freimachen) dem Prozessor mitgeteilt werden. Damit er diese Befehle überhaupt ausführen kann, muß das Ergebnisregister auf „1“ gesetzt werden. Dies gelingt mit Hilfe des ORC0-Befehls. Somit lautet das Initialisierungsprogramm:

Adresse	Befehl	Bedeutung des Befehls	Kurzform
00000000	ORC 0	Ergebnisregister auf 1 setzen	ER<----,1"
00000001	IEN 0	Eingabe freimachen	IEN<----ER (= „1“)
00000010	OEN 0	Ausgabe freimachen	OEN<----ER (= „1“)

Eingabe des Programms:

1. Taktwahlschalter auf „Handtakt“ stellen.
 2. Adreßwahlschalter auf 00000000 schalten.
 3. „Preset“- und „Reset“-Taster betätigen.
 4. „ORC“- und „0“-Tasten gleichzeitig drücken und mit der anderen Hand die „Write“- und die „Clock“-Taste je einmal betätigen.
 5. In gleicher Weise die anderen Befehle eingeben.
- Zum Starten des Programms:
6. „Preset“- und „Reset“-Tasten drücken.
 7. Taktwahlschalter in Mittelstellung bringen.
 8. Die Geschwindigkeit mit dem Taktpoti einstellen.

6.2 Einlesen von Informationseinheiten in den Lerncomputer

Informationen können mit den Schaltern des 4fach-DIL-Schalters oder von den Eingängen des Expansionssteckers mit LD oder mit LDC eingelesen werden. Ein Setzen jedes der 4fach-DIL-Schalter führt zum Leuchten der entsprechenden Eingabe-LEDs. Eine „1“ an den Eingängen E1-E3 des Expansionssteckers führt ebenso zum Leuchten der Eingabe-LEDs. Dazu müssen die 4fach-DIL-Schalter auf „0“ stehen. Das Programm zum Einlesen des Eingangs E1 sieht wie folgt aus:

Adresse	Befehl	
00000000	ORC 0	
00000001	IEN 0	
00000010	OEN 0	
00000011	LD 1	Der logische Pegel des

Eingangs 1 wird gelesen und im ER abgespeichert (ER<----E1). Bei E1 = „1“ ist ER = „1“. Bei E1 = „0“ ist ER = „0“. Der Zustand des ER ist an der ER-LED zu erkennen.

Manchmal ist es zweckmäßig, den einzulesenden Eingangspegel in seinem Wert umzukehren, zu invertieren. Dies geschieht mit LDC. Das Programm dazu ist ähnlich wie das vorangegangene:

00000000	ORC 0
00000001	IEN 0
00000010	OEN 0
00000011	LDC 1

Der logische Pegel des Eingangs 1 wird gelesen und invertiert im ER abgespeichert (ER<----E1). Bei E1 = „1“ ist ER = „0“, bei E1 = „0“ ist ER = „1“.

Der Querstrich über ER bringt zum Ausdruck, daß der unter dem Strich stehende Begriff in seinem Wert invertiert, d.h. umgekehrt wird.

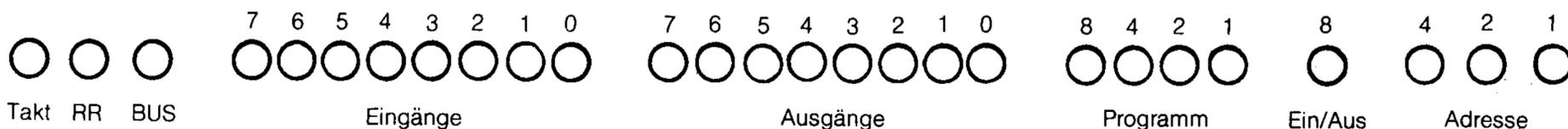
Variationen: Statt LD 1 bzw. LDC 1 kann LD 2, LDC 2, LD 3 oder LDC 3 eingegeben werden.

6.3 NOPF bedeutet Rücksprung zum Programmmanfang

Um zu vermeiden, daß sämtliche freien Speicherplätze nach dem Abarbeiten des Programms durchlaufen werden müssen, wird der NOPF-Befehl an das Programmende gesetzt:

00000000	ORC 0
00000001	IEN 0
00000010	OEN 0
00000011	LD 2
00000100	NOPF

Dieser Befehl bewirkt, wie das Betätigen des Presettasters, daß an den Programmmanfang gesprungen wird. Der Programmmanfang wird mit Hilfe des Adreßwahlschalters eingestellt.



Ein- und Ausgabe des 1-Bit-Prozessors

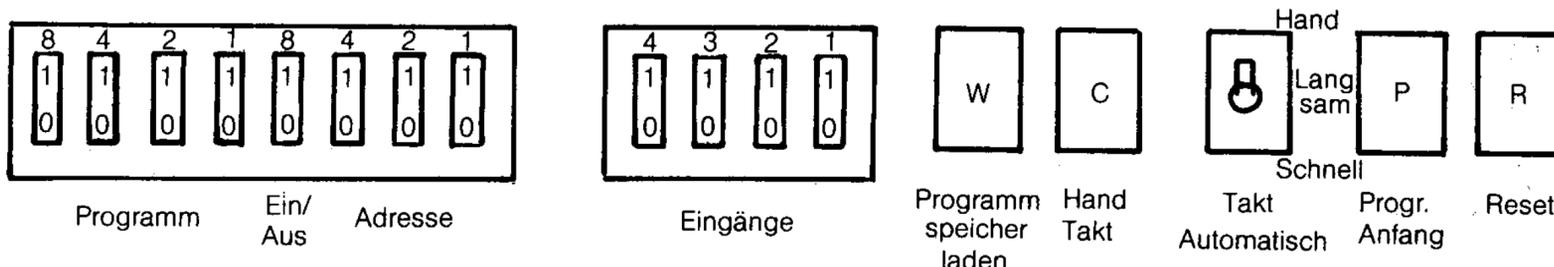


Abb. 26

Diesen Sachverhalt zeigt Abb. 27.

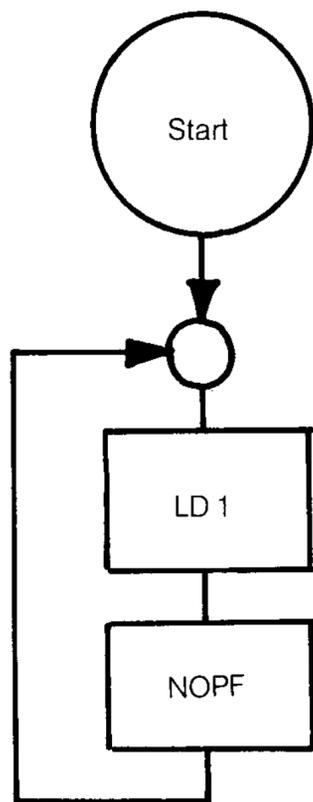


Abb. 27

6.4 Ausgabe von Informationen aus dem Prozessor

Mit dem Befehl STO kann eine Information aus dem Prozessor auf einen der 8 Ausgänge ausgegeben werden. Dazu ist folgendes Programm nötig:

```
00000000 ORC 0
00000001 IEN 0
00000010 OEN 0
00000011 LD 1
00000100 STO 0 Der Inhalt des ER wird
                  auf den Ausgang 0
                  gelegt (A0<---ER)
00000101 NOPF
```

Um die anderen Ausgänge anzusteuern, muß in dem Programm statt STO 0 STO 1, STO 2, STO 3, STO 4, STO 5, STO 6 oder STO 7 eingegeben werden.

Das folgende Programm bewirkt das Leuchten aller Ausgangs-LEDs.

```
00000000 ORC 0
00000001 IEN 0
00000010 OEN 0
00000011 LD 1
00000100 STO 0
00000101 STO 1
00000110 STO 2
00000111 STO 3
00001000 STO 4
00001001 STO 5
00001010 STO 6
00001011 STO 7
00001100 NOPF
```

Eine weitere Ausgabemöglichkeit bietet der STOC-Befehl:

```
00000000 ORC 0
00000001 IEN 0
00000010 OEN 0
00000011 LD 1
00000100 STOC 0 Der invertierte Inhalt des
                  ER wird auf den Aus-
                  gang gegeben. (A0<---
                  ER)
```

00000101 NOPF

Da die Ausgänge beliebig beschaltet werden können, ist es möglich, ein Lauflicht in den verschiedensten Variationen zu programmieren. Das folgende Programm zeigt ein solches Lauflicht:

```
00000000 ORC 0
00000001 IEN 0
00000010 OEN 0
00000011 LD 1
00000100 STO 0
00000101 STO 1
00000111 STO 2
00001000 STO 3
00001001 STO 4
00001010 STO 5
00001011 STO 6
00001100 STO 7
00001101 STOC 7
00001110 STOC 6
00001111 STOC 5
00010000 STOC 4
00010001 STOC 3
00010010 STOC 2
00010011 STOC 1
00010100 STOC 0
00010101 NOPF
```

Eine andere Möglichkeit besteht darin, eine Ausgangs-LED blinken zu lassen. Das Programm hierfür kann so aussehen:

```
Programm Rechteckgenerator:
00000000 ORC 0 LED aus
00000001 IEN 0
00000010 OEN 0
00000011 LD 1
00000100 STO 0 LED für 4 Takte an
00000101 STO 0
00000110 STO 0
00000111 STO 0
00001000 STOC 0 LED für 4 Takte aus
00001001 STOC 0
00001010 STOC 0
00001011 STOC 0
00001100 NOPF
```

Startet man das Programm, so kann man am Expansionsstecker, Ausgang A0, ein Rechtecksignal abgreifen, dessen Frequenz mit Hilfe des Potentiometers einstellbar ist. Es ist leicht einzusehen, daß auch Rechtecksignale mit anderen Tastverhältnissen programmiert werden können.

6.5 Die Befehle AND, ANDC, OR, ORC und XNOR

Um eine AND-Funktion am Prozessor zu bilden, muß folgendes Programm eingegeben werden:

```
INIT INITIALisierung (ORC 0, IEN 0, OEN 0)
LD 1
AND 2 Verknüpft den Zustand von Eingang
      E2 mit dem Zustand des ER in Form
      einer UND-Funktion: ER<---E2 * ER
      (*: Kurzschreibweise für UND).
      Das Ergebnis wird der neue Inhalt des
      ER.
STO 0 Das Verknüpfungsergebnis wird auf
      Ausgang A0 angezeigt.
```

NOPF

In verkürzter Form liest sich das Programm auch so:

```
INIT
LD 1 ER<---E1
AND 2 ER<---E2 * ER
STO 0 A0<---ER
```

Startet man das Programm im Schnelltaktverfahren und betätigt man die Eingangsschalter E1 und E2, so leuchtet die Ausgangs-LED am Ausgang 0 nur, wenn der eine UND der andere Schalter auf „1“ stehen. In allen anderen Fällen ist die Ausgangs-LED aus.

Die folgenden Programme bilden die jeweiligen Funktionen, die mit Hilfe der Eingangsschalter nachvollzogen werden können.

NAND:

```
INIT
LD 1 ER<---E1
AND 2 ER<---E2 * ER
STOC 0 A0<---ER
```

NOPF

OR:

```
INIT
LD 1 ER<---E1
OR 2 ER<---E2 + ER
```

+ : Kurzschreibweise für die ODER-Verknüpfung

```
STO 0 A0<---ER
NOPF
```

NOR:

```
INIT
LD 1 ER<---E1
OR 2 ER<---E2 + ER
STOC 0 A0<---ER
NOPF
```

XNOR:

```
INIT
LD 1 ER<---E1
XNOR 2 Vergleiche E2 mit ER
      ER = „1“ <---ER = E2,
      ER = „0“ <---ER ≠ E2
STO 0 A0<---ER
NOPF
```

Die Wahrheitstabelle zum Exklusiv NOR lautet somit:

E1	ER alt	ER neu
0	0	1
0	1	0
1	0	0
1	1	1

Immer wenn das Datenbit auf dem Datenbus gleich dem Datenbit im ER ist, lautet das neue Ergebnis im ER „1“ (vgl. 2.5).

Das Programm für ein XOR (ausschließendes ODER) lautet:

```
INIT
LD 1 ER<---E1
XNOR 2 Vergleiche E1 mit E2
STOC 0 A0<---ER
NOPF
```

6.6 Logische Funktionen können verkettet werden

Möchte man zwei AND-Funktionen und eine OR-Funktion hintereinander darstellen, muß lediglich beachtet werden, daß das Ergebnis jeder logischen Verknüpfung zwischengespeichert werden muß. Wie das geschieht, zeigt das folgende Programmbeispiel.

```

INIT
LD 1 ER<---E1
AND 2 ER<---ER * E2
STO 5 Zwischenspeicherung der ersten
logischen Verknüpfung E1 * E2
---->E/A5
LD 3 ER<---E3
AND 4 ER<---ER * E4
OR 5 Verwendung des zwischengespei-
cherten Ergebnisses ER<---(E3 *
E4)+ (E1 * E2)
STO 1 A1<---ER
NOPF
  
```

Die Zwischenspeicherung von Ergebnissen kann im vorliegenden System nur auf den Ausgängen 5 bis 7 geleistet werden, da nur diese Ausgänge mit den entsprechenden Eingängen verdrahtet sind.

6.7 Der OEN-Befehl ermöglicht eine WENN-DANN-Programmierung

Mit Hilfe des OEN-Befehls ist es möglich, innerhalb eines Programms nur bestimmte Programmblöcke durchlaufen zu lassen, andere Programmabschnitte bearbeitet der Prozessor dagegen nicht. Die Benutzung der Bezeichnung WENN - DANN sagt aus: Wenn ein Zustand eingetreten ist, dann soll ein bestimmter Programmabschnitt durchlaufen werden. Ist der Zustand dagegen nicht erreicht, so wird der bestimmte Programmabschnitt nicht bearbeitet. Im folgenden wird ein Beispiel für ein Wenn-Dann-Programm beschrieben: Wenn die Zündung ausgeschaltet ist, dann sollen sich die Parkleuchten einschalten.

Abb. 28 zeigt ein Flußdiagramm

Das Programm folgt:

```

INIT
LDC 1 ER<---E1
Laden des Wertes des Zündschloß-
schalters an E1 (vergl. 1.)
OEN 0 OEN<---ER
Sofern OEN = „1“, Freigabe der Aus-
gabe
STO 1 A1<---ER
Parkleuchte an Ausgang A1
anschalten
ORC 0 ER<---1
ER wird auf „1“ gesetzt
OEN 0 OEN<---ER
Ausgabe wird freigegeben
  
```

Läßt man das Programm z.B. im Einzelschrittverfahren durchlaufen, kann man es überprüfen.

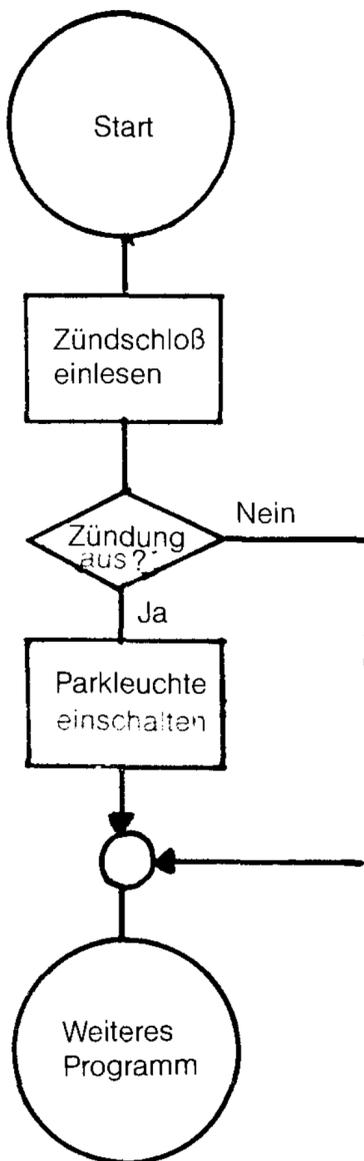


Abb. 28

Wenn es hell ist, dann soll die Fahrtbeleuchtung des PKW ausgeschaltet werden, ist ein weiteres Beispiel für eine Wenn-Dann-Beziehung:

```

INIT
STO 2 A2<---1
Damit das Programm sinnvoll ist,
muß die Beleuchtung vorher einge-
schaltet sein.
LDC 2 ER<---E2
Laden des Wertes des Helligkeits-
aufnehmers an Eingang E2
OEN 0 OEN<---ER
Sofern OEN = „1“, Freigabe der Aus-
gabe
STOC 2 A2<---ER
Fahrtbeleuchtung an Ausgang A2
abschalten
ORC 0 ER<---1
ER wird auf „1“ gesetzt
OEN 0 OEN<---ER
Ausgabe wird freigegeben.
  
```

Läßt man das Programm ebenfalls im Einzelschritt oder mit angehängtem NOPF durchlaufen, kann man es nachvollziehen.

Auch dieses Programm ist in Abb. 29 in einem Flußdiagramm dargestellt.

Beide Programme können wie folgt miteinander verschachtelt werden:

```

INIT
STO 2 A2<---1
Damit das Programm sinnvoll ist,
  
```

```

muß die Beleuchtung vorher einge-
schaltet sein.
LDC 1 ER<---E1
Laden des Wertes des Zündschloß-
schalters an E1
OEN 0 OEN<---ER
Sofern OEN = „1“, Freigabe der Aus-
gabe
STO 1 A1<---ER
Parkleuchte an Ausgang A1
anschalten
ORC 0 ER<---1
ER wird auf „1“ gesetzt
OEN 0 OEN<---ER
Ausgabe wird freigegeben
LD 2 ER<---E2
Laden des Wertes des Helligkeits-
aufnehmers an Eingang E2
OEN 0 OEN<---ER
Sofern OEN = „1“, Freigabe der Aus-
gabe
STOC 2 A2<---ER
Fahrtbeleuchtung an Ausgang A2
abschalten
ORC 0 ER<---1
ER wird auf „1“ gesetzt
OEN 0 OEN<---ER
Ausgabe wird freigegeben.
  
```

Das dazugehörige Flußdiagramm zeigt Abb. 30.

Für die Programmierung einer Wenn-Dann-Beziehung gilt:

1. Lade den Eingangszustand.

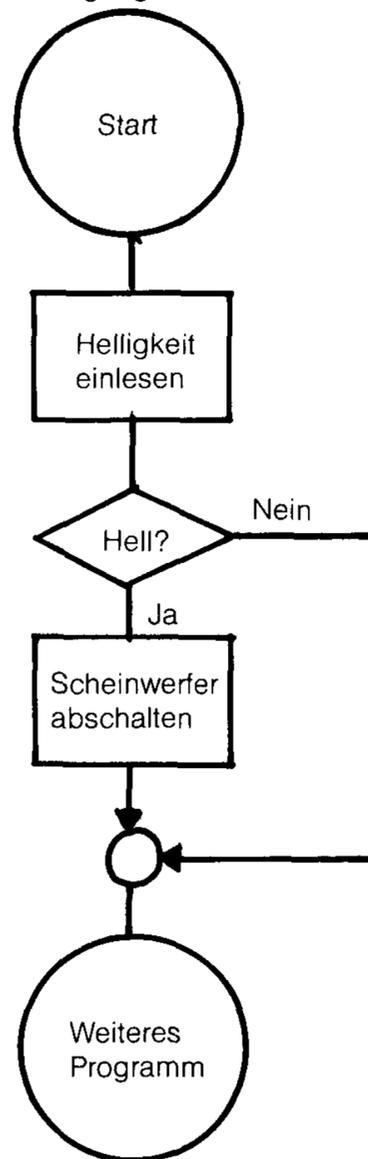


Abb. 29

2. Lade das OEN-Register mit dem Zustand des ER. Sofern OEN = „1“, Freigabe der Ausgabe.
3. Führe den Befehlsblock durch.
4. Setze OEN mit Hilfe von ORC auf 1, damit anschließende Befehle durchgeführt werden können.

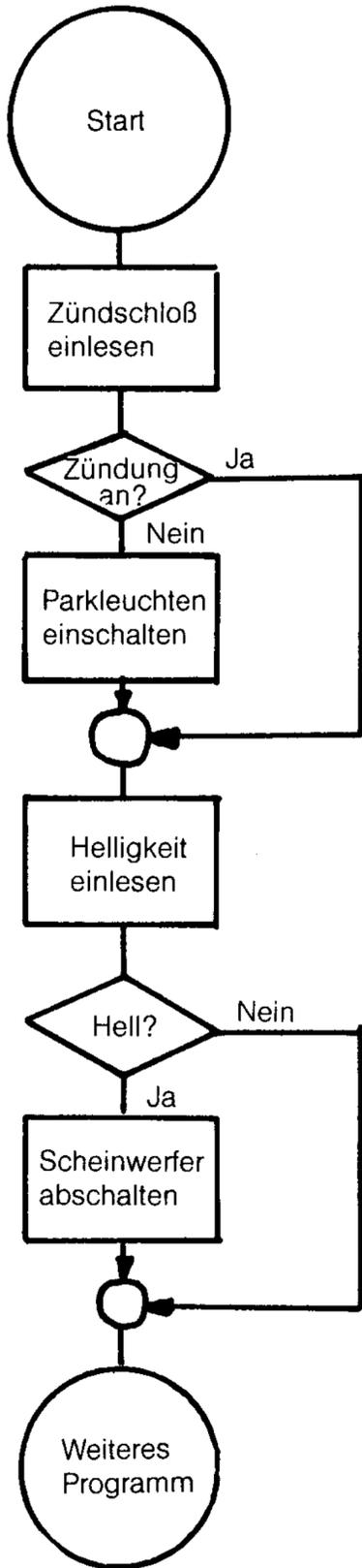


Abb. 30

6.8. Die Wenn-Dann-Sonst-Programmierung

In den Beispielen der Wenn-Dann-Programmierung waren keine Anweisungen enthalten, was geschehen soll, wenn das Auto fährt oder es hell ist. Die Wenn-Dann-Sonst-Struktur enthält ein weiterführendes Programm, z.B., die Parkbeleuchtung einzuschalten.

Das Flußdiagramm dieser Wenn-Dann-Sonst-Beziehung ist in Abb. 31 vorgestellt.

Die Programmierung für eine Wenn-Dann-Sonst-Anweisung lautet:
1. Lade den Eingangszustand.

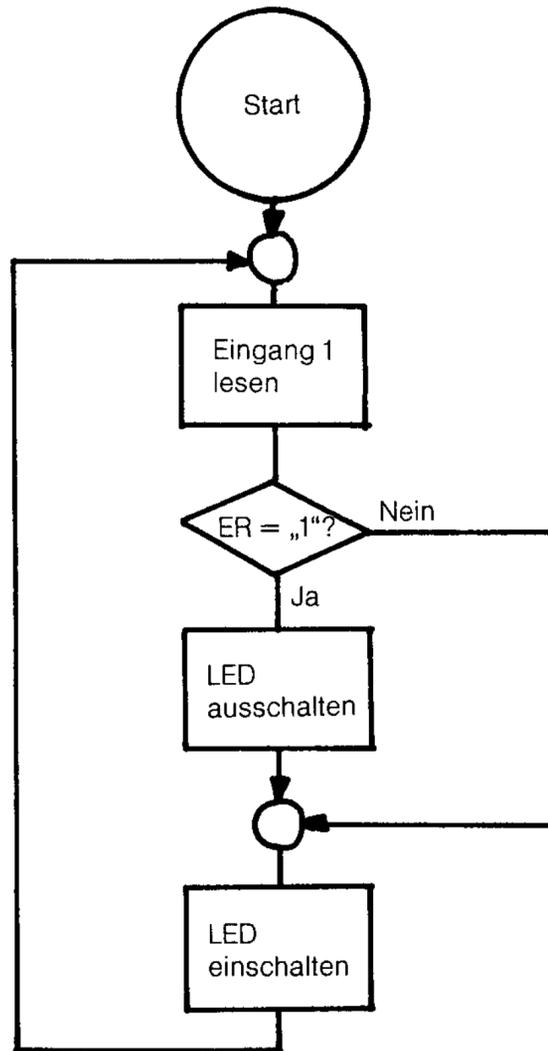


Abb. 31

2. Setze das Komplement des Eingangszustandes in den Zwischenspeicher.
3. Lade das OEN mit dem Zustand des ER.
4. Führe den ersten Befehlsblock durch.
5. Lade das OEN mit dem Wert des Zwischenspeichers.
6. Führe den 2. Befehlsblock durch.
7. Setze OEN mit Hilfe von ORC auf 1, damit anschließende Befehle durchgeführt werden können.

Mit Hilfe dieser Programmierung soll folgendes Problem gelöst werden: Wenn es hell ist oder der Motor läuft, dann soll sich die Parkleuchte ausschalten, sonst soll sich die Parkleuchte einschalten.

INIT

LD 1 ER<---E1

OR 2 ER<---ER + E2

STOC 5 E/A5<---ER

OEN 0 Freigabe Befehlsblock 1

STOC 2 Parkleuchte aus

OEN 5 Freigabe Befehlsblock 2 durch OEN<---E/A5

STOC 2 Parkleuchte an

ORC 0 ER<---ER + ER

Der Befehl führt immer zu einer „1“ in ER

OEN 0 OEN<---„1“

Natürlich kann ein kürzeres Programm die gestellte Aufgabe schneller lösen. Ein solches Programm ist jedoch zur Veranschauli-

chung der Wenn-Dann-Sonst-Struktur nicht zu gebrauchen.

Wie bei der Wenn-Dann-Struktur ist eine Verschachtelung von Wenn-Dann-Sonst-Strukturen möglich.

6.9. Die Programmierung mit Hilfe des SKZ-Befehls

Der Befehl Skip, wenn ER = Zero bedeutet:

überspringe, wenn ER = „0“.

Enthält zur Zeit des SKZ-Befehls das ER eine „0“, so wird der nachfolgende Befehl nicht ausgeführt. Das folgende Programm zum Ein- bzw. Abschalten eines Rechteckgenerators soll den Einsatz des SKZ-Befehls verdeutlichen. Zunächst folgt ein Flußdiagramm in Abb. 32.

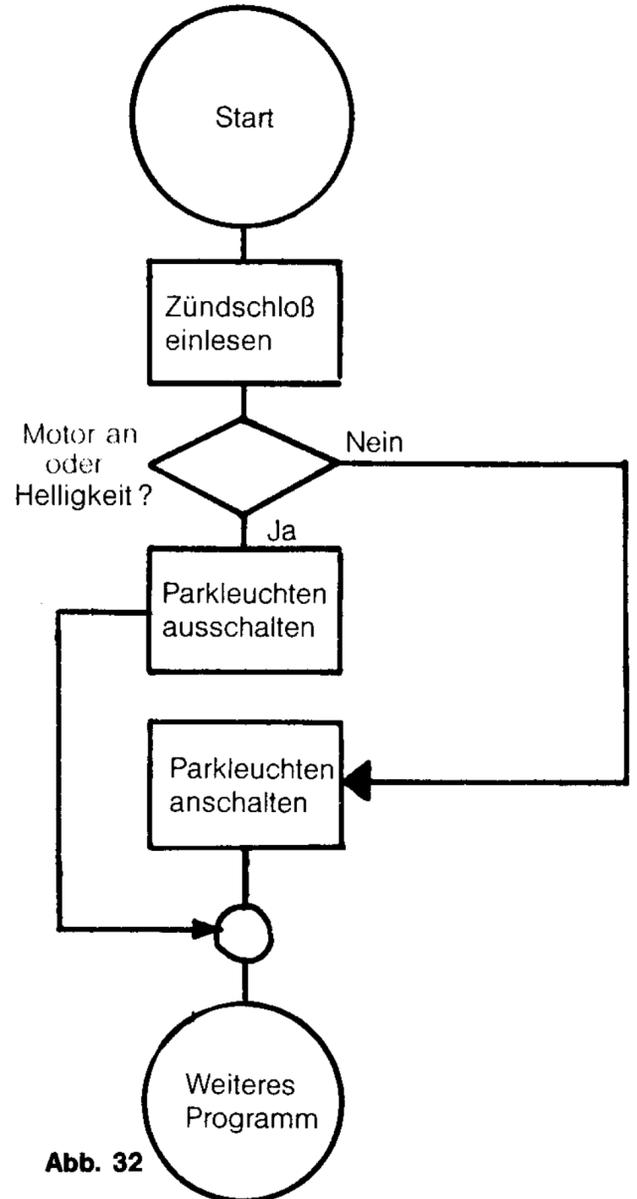


Abb. 32

Aus dem Flußdiagramm läßt sich folgende Programmieranweisung ableiten:

1. Lade den Eingangszustand
2. Setze SKZ
3. Führe den 1. Befehl aus
4. Führe den 2. Befehl aus.

Das Programm lautet somit:

INIT

LD 1 ER<---E1

SKZ überspringe den nächsten Befehl, wenn ER = „0“

STOC 1 ER--->A1

STO 1 ER--->A1

NOPF

(Fortsetzung folgt)

Ein 1-Bit-Lerncomputer zum Einstieg in die Mikrocomputertechnik (4)

Burkhard John, DK5JG, Zweibrückenstr. 35, 4000 Düsseldorf, 0211/23 44 89, und Volker Ludwig, DDØEU, Am Alten Bach 14a, 4040 Neuss 21, 02107/70226, als Sachbearbeiter „Amateurfunk in der Schule“ im Jugend- und Ausbildungsreferat des DARC.

6.10. Mit Hilfe von logischen Funktionen werden Flip-Flops gebildet.

Drei Flip-Flop-Typen, das S-R-Flip-Flop, das D-Flip-Flop und das JK-Flip-Flop, sollen, weil sie zum Verständnis des Lerncomputers gehören (vgl. 3.2 ff), mit Hilfe von kurzen Programmen dargestellt werden.

Um ein Set-Reset-Flip-Flop auf dem 1-Bit-Lerncomputer zu realisieren, müssen zunächst die Ein- und Ausgänge des S-R-Flip-Flops den Ein- und Ausgängen des Prozessors zugeordnet werden.

S	R	Zwischenspeicher	Q
E2	E1	E/A5	A3

Nun läßt sich das Programm erklären:

```

INIT
LD 2 ER<---E2
    Set-Eingang lesen
STO 5 ER--->E/A5
    Gelesenen Wert von S zwischen-
    speichern
LDC 1 ER<---ET
    Reset-Eingang lesen
XNOR 5 ER<---ER XNOR E/A5
    Vergleiche E1 mit E2
OEN 0 OEN<---ER
    Ausgabe abschalten, wenn S=R
LD 5 ER<---E/A5
    Zwischenspeicher lesen
STO 3 ER--->A3
    Q=S auf den Ausgang geben
XNOR 0 ER<---ER XNOR ER
    ER auf „1“ setzen
OEN 0 OEN<---ER
    Ausgänge einschalten
NOPF
    
```

Nach Eingabe des Programms wird auf Schnelltakt geschaltet. Ein abwechselndes Betätigen der beiden Eingabeschalter läßt die Speicherfunktion erkennen. Für das bessere Verständnis der Vorgänge sollen bei den vier möglichen Eingangszuständen S = „0“/R = „0“, S = „1“/R = „0“, S = „0“/R = „1“ und S = „1“/R = „1“ die jeweiligen Pegel über den Kurzerklärungen eingetragen werden, wie sie auch im Einzeltaktverfahren nachvollzogen werden können.

Die Pegel bei S=R: S = „0“/R = „0“ haben folgende Werte.

```

INIT
    0 0
LD 2 ER<---E2
    Set-Eingang lesen
    0 0
STO 5 ER--->E/A5
    Gelesenen Wert von S zwischen-
    speichern
    
```

```

    1 0
LDC 1 ER<---ET
    Reset-Eingang lesen
    0 1 0
XNOR 5 ER<---ER XNOR E/A5
    Vergleiche E1 mit E2
    0 0
OEN 0 OEN<---ER
    Ausgabe abschalten, wenn S=R
    0 0
LD 5 ER<---E/A5
    Zwischenspeicher lesen
    0 0
STO 3 ER--->A3
    Q=S auf den Ausgang geben: Aus-
    gabe ist gesperrt
    1 0 0
XNOR 0 ER<---ER XNOR ER
    ER auf „1“ setzen
    1 1
OEN 0 OEN<---ER
    Ausgabe einschalten
NOPF
    Die Pegel für S = „0“ und R = „1“ lauten:
INIT
    0 0
LD 2 ER<---E2
    Set-Eingang lesen
    0 0
STO 5 ER--->E/A5
    Gelesenen Wert von S zwischen-
    speichern
    0 1
LDC 1 ER<---ET
    Reset-Eingang lesen
    1 0 0
XNOR 5 ER<---ER XNOR E/A5
    Vergleiche E1 mit E2
    1 1
OEN 0 OEN<---ER
    Ausgabe abschalten, wenn S=R
    0 0
LD 5 ER<---E/A5
    Zwischenspeicher lesen
    0 0
STO 3 ER--->A3
    Q=S auf den Ausgang geben: Aus-
    gabe ist geöffnet
    1 0 0
XNOR 0 ER<---ER XNOR ER
    ER auf „1“ setzen
    1 1
OEN 0 OEN<---ER
    Ausgänge einschalten
NOPF
    
```

Für die Eingangszustände S = „1“ und R = „0“ lauten die Pegel:

```

INIT
    1 1
LD 2 ER<---E2
    Set-Eingang lesen
    1 1
STO 5 ER--->E/A5
    Gelesenen Wert von S zwischen-
    speichern
    1 0
LDC 1 ER<---ET
    Reset-Eingang lesen
    1 1 1
XNOR 5 ER<---ER XNOR E/A5
    Vergleiche E1 mit E2
    1 1
OEN 0 OEN<---ER
    Ausgang Q abschalten, wenn S=R
    1 1
LD 5 ER<---E/A5
    Zwischenspeicher lesen
    1 1
STO 3 ER--->A3
    Q=S auf den Ausgang geben: Aus-
    gabe ist geöffnet
    1 1 1
XNOR 0 ER<---ER XNOR ER
    ER auf „1“ setzen
    1 1
OEN 0 OEN<---ER
    Ausgänge einschalten
NOPF
    Die Pegel für S = R: S = „1“ und R = „1“
    sehen wie folgt aus:
INIT
    1 1
LD 2 ER<---E2
    Set-Eingang lesen
    1 1
STO 5 ER--->E/A5
    Gelesenen Wert von S zwischen-
    speichern
    0 1
LDC 1 ER<---ET
    Reset-Eingang lesen
    0 0 1
XNOR 5 ER<---ER XNOR E/A5
    Vergleiche E1 mit E2
    0 0
OEN 0 OEN<---ER
    Ausgang Q abschalten, wenn S=R
    1 1
LD 5 ER<---E/A5
    Zwischenspeicher lesen
    
```

```

1 0
STO 3 ER---->A3
      Q=S auf den Ausgang geben: Aus-
      gang ist gesperrt
1 1 1
XNOR 0 ER<----ER XNOR ER
      ER auf „1“ setzen
1 1
OEN 0 OEN<---ER
      Ausgänge einschalten

```

NOFF
 Will man ein D-Flip-Flop mit Hilfe eines Programms verwirklichen, muß zunächst wieder eine Zuordnung der Ein- und Ausgänge vorgenommen werden.

Takt	D	Zwischenspeicher	Taktspeicher	Q
E1	E2	E/A5	E/A6	A3

Das Programm erklärt sich wie folgt:

```

INIT
LD 6 ER<----E/A6
      Taktspeicher abfragen
STO 5 ER---->E/A5
      Zwischenspeichern
LD 1 ER<----E1
      Takt einlesen
STO 6 ER---->E/A6
      In den Taktspeicher geben
ANDC 5 ER<----ER * E/A5
      Verknüpfe den Takt mit dem Kom-
      plement des alten Takts mit logisch
      UND
OEN 0 OEN<---ER
      Ausgabe freigeben, wenn „1“ gefun-
      den, d.h., der Takt ist von „0“ auf „1“
      geschaltet worden
LD 2 ER<----E2
      D-Eingang lesen
STO 3 ER---->A3
      Auf den Q-Ausgang setzen
ORC 0 ER<----ER + ER
      ER auf „1“ setzen
OEN 0 OEN<---ER
      Ausgänge einschalten
NOFF

```

Mit den entsprechenden Eingangsschaltern kann das Verhalten des D-Flip-Flops im Schnelltakt erkundet werden.

Auch an dieser Stelle sollen zum besseren Verständnis die Pegel bei den vier möglichen Eingangszuständen über den Kurzerklärungen aufgeführt werden. Für Takt = „0“ und D = „0“ lauten sie: (Takt = E1, D = E2)

```

INIT
0 0
LD 6 ER<----E/A6
      Taktspeicher abfragen
0 0
STO 5 ER---->E/A5
      Zwischenspeichern
0 0
LD 1 ER<----E1

```

```

Takt einlesen
0 0
STO 6 ER---->E/A6
      In den Taktspeicher geben
0 0 1
ANDC 5 ER<----ER * E/A5
      Verknüpfe den Takt mit dem Kom-
      plement des alten Takts mit logisch
      UND
0 0
OEN 0 OEN<---ER
      Ausgabe freigeben, wenn „1“ gefun-
      den
0 0
LD 2 ER<----E2
      D-Eingang lesen
0 0
STO 3 ER---->A3
      Auf den Q-Ausgang setzen
1 0 1
ORC 0 ER<----ER + ER
      ER auf „1“ setzen
1 1
OEN 0 OEN<---ER
      Ausgänge einschalten
NOFF
      Für Takt = „0“ und D = „1“ gilt (Takt = E1,
      D = E2):
INIT
0 0
LD 6 ER<----E/A6
      Taktspeicher abfragen
0 0
STO 5 ER---->E/A5
      Zwischenspeichern
0 0
LD 1 ER<----E1
      Takt einlesen
0 0
STO 6 ER---->E/A6
      In den Taktspeicher geben
0 0 1
ANDC 5 ER<----ER * E/A5
      Verknüpfe den Takt mit dem Kom-
      plement des alten Taktes mit
      logisch UND
0 0
OEN 0 OEN<---ER
      Ausgabe freigeben, wenn „1“ gefun-
      den
1 1
LD 2 ER<----E2
      D-Eingang lesen
1 0
STO 3 ER---->A3
      Auf den Q-Ausgang setzen (Aus-
      gang ist gesperrt)
1 1 0
ORC 0 ER<----ER + ER
      ER auf „1“ setzen
1 1

```

```

OEN 0 OEN<---ER
      Ausgänge einschalten
NOFF
      Für Takt = „1“ und D = „0“ lauten die
      Pegel (Takt = E1, D = E2):
INIT
0 0
LD 6 ER<----E/A6
      Taktspeicher abfragen
0 0
STO 5 ER---->E/A5
      Zwischenspeichern
1 1
LD 1 ER<----E1
      Takt einlesen
1 1
STO 6 ER---->E/A6
      In den Taktspeicher geben
1 1 1
ANDC 5 ER<----ER * E/A5
      Verknüpfe den Takt mit dem Kom-
      plement des alten Takts mit logisch
      UND
1 1
OEN 0 OEN<---ER
      Ausgabe freigeben, wenn „1“ gefun-
      den
0 0
LD 2 ER<----E2
      D-Eingang lesen
0 0
STO 3 ER---->A3
      Auf den Q-Ausgang setzen
1 0 1
ORC 0 ER<----ER + ER
      ER auf „1“ setzen
1 1
OEN 0 OEN<---ER
      Ausgänge einschalten
NOFF
      Für den letzten Fall, Takt = „1“ und D = „1“,
      werden die Pegel:
INIT
0 0
LD 6 ER<----E/A6
      Taktspeicher abfragen
0 0
STO 5 ER---->E/A5
      Zwischenspeichern
1 1
LD 1 ER<----E1
      Takt einlesen
1 1
STO 6 ER---->E/A6
      In den Taktspeicher geben
1 1 1
ANDC 5 ER<----ER * E/A5
      Verknüpfe den Takt mit dem Kom-
      plement des alten Takts mit logisch
      UND
1 1

```

OEN 0 OEN<---ER
Ausgabe freigeben, wenn „1“ gefunden
1 1

LD 2 ER<---E2
D-Eingang lesen
1 1

STO 3 ER--->A3
Auf den Q-Ausgang setzen
1 1 0

ORC 0 ER<---ER + ER
ER auf „1“ setzen
1 1

OEN 0 OEN<---ER
Ausgänge einschalten

NOPF

Das Programm für ein JK-Flip-Flop ist etwas aufwendiger, da drei Eingänge verarbeitet werden müssen. Zunächst wiederum die Zuordnung der Ein- und Ausgänge:

J	K	Takt	Zwischenspeicher	Taktspeicher	Q
E2	E3	E1	E/A5	E/A6	A3

Das Programm hat folgende Erklärung:

INIT

LD 6 ER<---E/A6
Taktspeicher abfragen

STO 5 ER--->E/A5
Zwischenspeichern

LD 1 ER<---E1
Takt einlesen

STO 6 ER--->E/A6
In den Taktspeicher geben

ANDC 5 ER<---ER * E/A5
Verknüpfe den Takt mit dem Komplement des alten Takts mit logisch UND, prüfe, ob „0“-->„1“-Übergang vorliegt

OEN 0 OEN<---ER
Ausgabe freigeben, wenn „1“ gefunden

ANDC 3 ER<---ER * A3
Verknüpfe ...

STO 5 ER--->E/A5

LDC 3 ER<---E3

AND 2 ER<---ER * E2

OR 5 ER<---ER + E/A5

STO 3 ER--->A3

ORC 0 ER<---ER + ER

OEN 0 OEN<---ER

NOPF

Mit Hilfe der drei Eingabeschalter kann das Verhalten eines JK-Flip-Flops studiert werden.

7. Der 1-Bit-Prozessor als Rufzeichengeber

Der 1-Bit-Prozessor wird durch ein einfaches Programm zum Rufzeichengenerator. Es besteht aus nur sechs verschiedenen Befehlen.

Bedeutung der Befehle:

ORC 0 Definiert das Ergebnisregister

IEN 0 Schaltet die Eingänge ein

OEN 0 Schaltet die Ausgänge ein

STO 1 Schaltet den Morsegenerator ein

STOC 1 Schaltet den Morsegenerator aus

NOPF 0 Springt zum Programmanfang zurück

Erläuterung des Programms:

Nach der Initialisierung bewirken 3 mal STO 1 und dahinter ein STOC 1 einen Morsestrich mit anschließender Pause. Ein Morsepunkt mit Pause wird durch die Programmfolge STO 1, STOC 1 erreicht. Eine Buchstabenpause entsteht durch Anfügen von zwei STOC 1-Befehlen. Längere Pausen ergeben sich durch entsprechend viele STOC 1-Befehle hintereinander.

Notwendige Lötarbeit: Der Steuereingang eines Tongenerators muß mit Pin 12 des Expansionssteckers verbunden werden.

Programmbeispiel Anfang eines Rufzeichens (DL ...):

1. ORC 0 Systeminitialisierung
2. IEN 0
3. OEN 0
4. STO 1
5. STO 1 | DAH
6. STO 1 |
7. STOC 1
8. STO 1 | DIT
9. STOC 1
10. STO 1 | DIT
11. STOC 1
12. STOC 1 PAUSE
13. STOC 1
14. STO 1 | DIT
15. STOC 1
16. STO 1 |
17. STO 1 | DAH
18. STO 1 |
19. STOC 1
20. STO 1 | DIT
21. STOC 1
22. STO 1 | DIT
23. STOC 1
24. STOC 1 PAUSE
25. STOC 1

... usw.

Letzter Befehl NOPF0 als Rücksprung zum Programmanfang

8. Programm: „Elektronische Morsetaste für den 1-Bit-Prozessor JR25 - JR29“:

- 1 ORC 0 Vorbereitung des Prozessors
- 2 IEN 0 Eingänge einschalten
- 3 OEN 0 Ausgänge einschalten
- 4 LD 1 Punktkontakt auf Eingang 1 abfragen
- 5 STO 5 Punkt speichern auf Aus-/Eingang 5
- 6 LD 2 Strichkontakt auf Eingang 2 abfragen
- 7 STO 6 Strich speichern auf Aus-/Eingang 6
- 8 OR 5 Feststellen, ob ein Punkt oder

Strich gespeichert ist. Falls ja, ist das Ergebnisregister = 1

9 LDC 0 Invertieren des Ergebnisregisters

10 SKZ Falls Ergebnisregister = 0, nächsten Befehl überspringen

11 NOPF Rücksprung zum Anfang, falls Ergebnisregister = 1, also weder Strich noch Punkt

12 STOC 1 Ton einschalten. Hier steht im Ergebnisregister 0, deshalb wird eine 1 ausgegeben

13 STO 5 Löschen des Punktspeichers

14 LDC 6 Nachsehen, ob ein Punkt gespeichert ist

15 OEN 0 Falls ein Strich da, Ausgänge abschalten

16 NOPO Punktzeit abwarten

17 NOPO Eine Punkt- bzw. Pausenzeit ist 13 CPU-Takte, d.h. 13 Programmschritte

18 NOPO

19 NOPO

20 NOPO

21 NOPO

22 NOPO

23 NOPO

24 STOC 1 Falls Punkt, Ton aus

25 SKZ Falls Strich, nächsten Befehl überspringen

26 NOPF Ende des Punktes, Rücksprung zum Anfang

27 NOPO Im Ergebnisregister steht jetzt 0

28 LDC 0 Invertieren des Ergebnisregisters

29 OEN 0 Einschalten der Ausgänge

30 STOC 6 Löschen des Strichspeichers

31 NOPO Restliche Strichzeit abwarten

32 NOPO Gesamte Strichzeit: 39 Programmschritte

33 NOPO

34 NOPO

35 NOPO

36 NOPO

37 NOPO

38 NOPO

39 NOPO

40 NOPO

41 NOPO

42 NOPO

43 NOPO

44 NOPO

45 NOPO

46 NOPO

47 NOPO

48 NOPO

49 NOPO

50 NOPO

51 STOC 1 Ton abschalten

52 NOPF Ende des Striches, Rücksprung zum Anfang

Der Punktkontakt des Wabblers wird an Eingang 1, Pin 5 des Expansionssteckers, der Strichkontakt an Eingang 2, Pin 4, angeschlossen. Der Tongenerator wird an Ausgang 1, Pin 12, gelegt.

9. Verwendete Teile

9.1 Halbleiter
1 MC 14500 Motorola

1 MC 14599 Motorola
 1 CD 4049
 1 CD 4051
 1 CD 4071
 2 CD 4029
 2 RAM 2112
 4 ULN 2003
 1 7805
 1 NE 555
 27 LED (21 rot, 4 grün, 2 gelb)
 54 Dioden 1N4148 o.ä.
 1 Diode 1N4001

9.2 Widerstände

27 220 Ohm
 29 12 kOhm
 12 15 kOhm (Die 8 15-kOhm-Pull-up-Widerstände für die Datenleitungen der RAMs durch 8 22-kOhm-Widerstände ersetzen.)
 8 3,3 kOhm
 4 120 kOhm
 1 56 kOhm
 1 1 MOhm
 1 1-MOhm-Poti

9.3 Kondensatoren

1 100nF
 1 1µF
 1 10µF Tantal
 1 220µF oder mehr

9.4 Sonstiges

22 Taster für JR 29 mit entsprechendem Raster
 2 8-Pol-IC-Fassungen
 1 14-Pol-IC-Fassung
 13 16-Pol-IC-Fassungen
 1 18-Pol-IC-Fassung
 1 8fach-DIL-Schalter
 1 4fach-DIL-Schalter
 1 16-Pol-DIL-Stiftleiste für 16-Pol-Fassung
 1 13-Pol-Buchsenleiste
 1 21-Pol-Buchsenleiste
 2 31-Pol-Buchsenleisten
 2 13-Pol-Steckerleisten
 1 21-Pol-Steckerleiste
 2 31-Pol-Steckerleisten
 4 Schadow Digitaster Umschalter
 1 Schalter Ein-Aus-Ein

5 Schrauben und Muttern M3 als Füße für die BUS-Platine
 2 Lötnägel
 1 15 cm 9-Pol-Flachbandkabel

9.5 Platinen

Je eine Platine JR 25, JR 26, JR 27, JR 28, JR 29

10. Bezugsquellen und Literatur:

Der Platinensatz JR 25 - 29 kann für 25 DM + 2,50 DM Porto/Verpackung bei den Autoren (DK5JG) bezogen werden (Scheck).

Bis auf die beiden IC's MC-14500 und MC-14599 sind alle Bauteile gängig und somit fast überall erhältlich.

Beim Verfasser sind eine Anzahl der Spezial-IC's MC-14500 und MC-14599 vorrätig, die günstig erworben werden konnten. Sie werden als Satz (14500 und 14599) gegen 30 DM (Scheck) und Freiumschlag abgegeben (DDØEU).

Eine ausführliche Beschreibung des Motorola MC-14500 ist das „Industrial Control Unit MC 14500 Handbuch“ in deutscher Sprache. Weitere Literatur ist u.a. MC 10/82, S. 54 ff., Franzis Verlag München.

R. L. Drake: Umstieg auf Satelliten-Empfänger

Mehr Dichtung als Wahrheit war in der letzten Zeit über R. L. Drake zu hören, den bekannten Hersteller von Amateurfunkgeräten aus Miamisburg/USA. Dabei ist Öffentlichkeitsarbeit das einzige, was bei Drake nicht so ganz geklappt hat. Vielleicht hatte man auch einfach keine Zeit dazu. Denn in den letzten drei Jahren verzehnfachte die Firma ihren Umsatz auf 65 Mio. US-\$ und erweiterte im gleichen Zeitraum den Mitarbeiterstab von 160 auf mehr als 500 Angestellte. Es wurde ein neues Verwaltungsgebäude bezogen; auf dem alten Gelände an der 540 Richard Street zog dafür das Service-Center ein. In neuen Werkhallen von nunmehr 10.000 qm werden seit einiger Zeit höchst erfolgreich die auf dem Amateurfunkmarkt gewonnenen Erfahrungen in einer innovativen Branche umgesetzt: mit Satelliten-Empfängern konnte sich Drake in den USA auf dem heiß umkämpften Markt einen Anteil von 30% erobern. Alles von der Blechbearbeitung über den CAD-gestützten Platinenentwurf bis hin zur automatisierten Fertigung und dem Service wird im eigenen Hause gemacht. Und der Boom hält an – eine Lagerhaltung ist nicht notwendig. So sehr reißen sich die Distributoren um die begehrte Drake-Ware. Sie liefern diese an Händler weiter, die dann komplette Empfangssysteme zusammenstellen. Verständlicherweise wurde dank des überwältigenden Erfolges in der Satellitentechnik die Produktion von Amateurfunkgeräten eingestellt. Der Service für die auf dem Gebrauchtmärkte immer noch recht hoch gehandelten Drake-Empfänger und -Transceiver ist selbstverständlich auch in den nächsten Jahren gesichert.

Tip: Verbesserungen am Yaesu FT-480R/FT-780R

1. Das FT-480R wird bezüglich des Frequenzbereiches und der Schrittweiten in 5 Versionen hergestellt. Mein gebrauchtes erstandenes Gerät war die US-Version. Zum Umbau in die DL-Version sind einige Dioden umzulöten, die am Prozessor-IC µPD 1511, teilweise unter dem IC, angebracht sind. Einige Dioden ließen sich nur stehend einlöten. Im Schaltplan sind alle Dioden eingezeichnet. Dazu findet man in der englischsprachigen Betriebsanleitung die Beschriftungstabelle. Zusammen mit den folgenden Schaltungsänderungen erweist sich Type D als vorteilhaft.

2. Die interessanteste Änderung betrifft den Scanner. Leider besitzt der Scanner keine Start/Stop-Automatik, sondern muß von Hand immer wieder neu gestartet werden. Nachdem ich lange vergebens die „Low“- und „High“-Zustände am Prozessor-IC gemessen hatte, versuchte ich es mit einem kleinen Wagnis. Was passiert, wenn man den einen oder anderen Anschluß abklemmt? Ein verblüffendes Resultat: Das Gerät arbeitet einwandfrei im gewünschten Autostartbetrieb, wenn man an der Steckleiste J 03, die zum Prozessor µPD 1511 führt, den Anschluß „Scan Stop“ herauszieht oder abkneift. Das abgekneifene Kabel wird mit Isolierband versehen und fristet danach ein nutzloses Dasein. Der IC-Eingang bleibt dabei durch einen 22-kOhm-Widerstand gegen Masse vor undefinierten Zuständen und statischen Aufladungen sicher geschützt. Für Autostartbetrieb schaltet man auf „Busy“

und drückt die „Up“-Taste. Der Squelch sollte eventuell etwas weiter zuge dreht werden. Nunmehr wird das ganze Band kontinuierlich abgesucht – der Scanner funktioniert einwandfrei und stoppt jeweils 6 Sekunden auf jeder belegten Frequenz. Nach dieser kleinen Änderung hat man fast das Gefühl, ein neues Gerät zu besitzen

3. Die Rasterung im Frequenzwahlknopf kann leicht unangenehm werden. Außerdem ist sie lauter als manches schwache HF-Signal. Wenn man die Frontplatte vorzieht, kann man leicht am Schleifschalter eine Feder und ein Kügelchen entfernen. Mobilisten können die Frequenz gegen versehentliches Verstellen sichern, indem sie erst „Memory“ und danach „Memory Recall“ drücken.

4. Leider ist es mir nicht gelungen, das Gerät hundertprozentig fuchsjagdtauglich zu machen. Da die LED-Zeile keine genauen Peilungen zuläßt, habe ich am Anschlußstift bei D 1030 / C 10274 die entsprechende Analogspannung gemessen. Erst bei stärkeren Signalen erhält man eine Anzeige, schwache Signale sind damit also nicht zu peilen. Das Trimpoti VR 1004 dient zur Einstellung des S-Meters, bringt aber keine Verbesserung bei schwachen Signalen.

Eine wirkungsvolle RF-Gain-Regelung läßt sich hingegen gut verwirklichen, indem man an das Trimpoti VR 1001 einen Draht anlötet und den Anschluß zu einer Klinkensteckerbuchse führt. An dieser Buchse sollte man dann im Bedarfsfall ein 5-kOhm-Poti anschließen. Damit liegt dieses Poti parallel zum VR 1001. Die Eingangsempfindlichkeit läßt sich damit so weit herabregeln, daß man noch auf geringste Entfernung peilen kann.

Matthias Beier, DD4WW