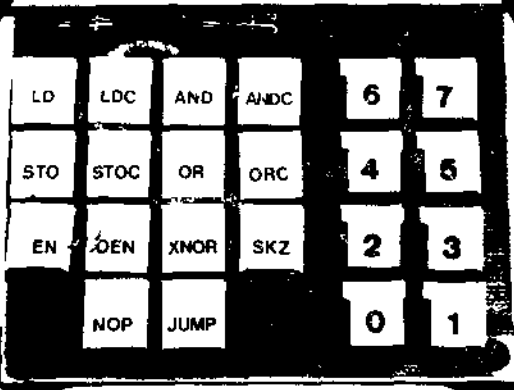
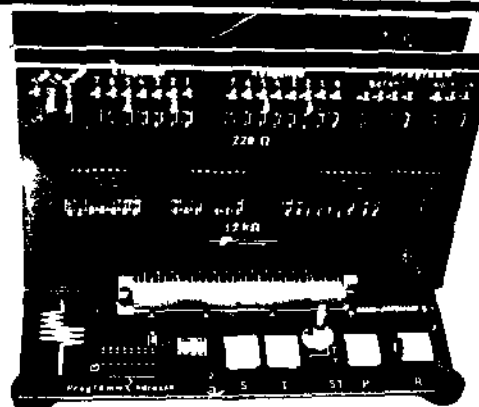
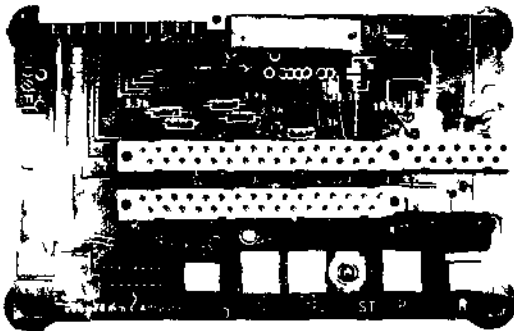
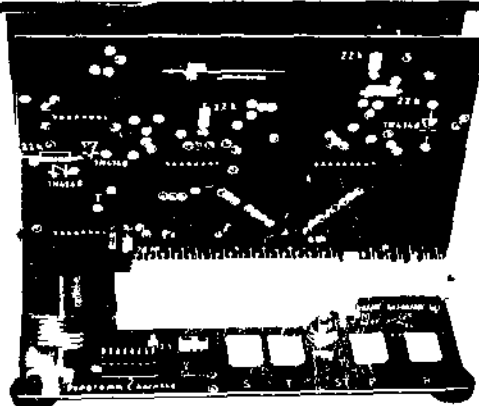
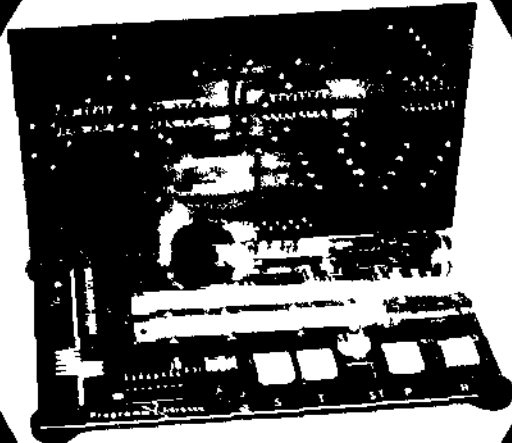


WDR

FAST ALLES UBER DEN
WDR-1-Bit-Computer



VOLKER LUDWIG
KLAUS PASCHENDA U.A.

Inhaltsverzeichnis

1.	Vorbemerkungen	1
2.	Der Aufbau des WDR-1-Bit-Computers (Hardware)	8
2.1.	Der prinzipielle Aufbau	8
2.2.	Die Funktionseinheiten	13
2.2.1.	Der Mikroprozessor MC 14500	13
2.2.2.	Der Eingabebaustein CD 4051	21
2.2.3.	Der Ausgabebaustein MC 14599	23
2.2.4.	Die Eingabetastatur	24
2.2.5.	Der Programmspeicher 2112	26
2.2.6.	Der Programmzähler CD 4029	28
2.2.7.	Die Anzeigeeinheit	30
2.2.8.	Die Takteinheit	31
2.2.9.	Die Stromversorgung	35
2.2.10.	Der 6-fach-Inverter CD 4049	36
2.2.11.	Das ODER-Glied	36
2.3	Das Zusammenwirken der Funktionseinheiten	37
3.	Der Zusammenbau	43
3.1.	Bausatz und Werkzeug	43
3.2.	Die Anzeigeplatine	45
3.3.	Die Grundplatine	50
3.4.	Die Speicherplatine	53
3.5.	Die Prozessorplatine	55
3.6.	Die Tastatur	56
4 .	Die Programmierung des WDR-1-Bit-Computers (Software)	59
4.1.	Der Befehlssatz des Mikroprozessors	59
4.2.	Das Eingeben von Programmen über DIL-Schalter	61
4.3.	Einfache Programme	63
4.3.1.	Programme zur Eingabe von Daten	63
4.3.2.	Programme zur Ausgabe von Daten	65
4.4.	Es wird bequemer: Die Tastatur	66
4.5.	Weitere Programme	66
4.5.1.	Logische Verknüpfungen	66
4.5.2.	Programmverzweigungen	70
4.5.3.	Programme zur Darstellung von Flipflops	73
4.5.4.	Programme zu arithmetischen Funktionen	82
4.5.5.	Simulationsprogramme	95
4.6.	Programme zur Steuerung angeschlossener Peripherie	102
4.6.1.	Motorsteuerung	104
4.6.2.	Sortieranlage	105
4.6.3.	Steuerung eines Roboters	108
4.6.4.	Ampelsteuerung	111
4.6.5.	Morsezeichengenerator	113
4.6.6.	Siebensegmentanzeige	114
4.6.7.	Bohrer	115
4.6.8.	Plotter	116
	Anhang	119

Volker Ludwig DDOEU
Klaus Paschenda DG5YCK

unter Mitarbeit von

Heinz Schepers
Hermann-Josef Terplane
Dr. Klaus Grannemann
Burkhard John
Hermann Komar
Ludwig Meinersen

FAST ALLES ÜBER DEN WDR-1-BIT-COMPUTER

Einige Hinweise in eigener Sache

Die Darstellungen von Schaltungen und Verfahren erfolgen ohne Rücksicht auf bestehende Patente, da sie nur für Amatuer- und Lehrzwecke gedacht sind. Ausdrücklich untersagt wird die gewerbliche Nutzung.

Sämtliche Rechte an Text und Bildern bleiben vorbehalten. Die Vervielfältigung einzelner Teile zu Lehrzwecken ist unter Verweis auf die Quelle erlaubt.

Trotz sorgfältiger Prüfung aller Darstellungen, Schaltungen und Verfahren sind Fehler nicht auszuschließen. Daher kann weder eine Garantie für Fehlerlosigkeit gegeben werden noch die juristische Verantwortung oder irgendeine Haftung übernommen werden.

Titelbild: Klaus Paschenda

Vertrieb: DATANorf
Am alten Bach 14a
4040 Neuss 21

Neuss und Recklinghausen 1986

1. VORBEMERKUNGEN

SO IST DER WDR-1-BIT-COMPUTER ENTSTANDEN

Nach dem Aufbau von über 50 Computern auf Z-80-Basis mit Teilnehmern an der Volkshochschule Düsseldorf durch Burkhard John und Volker Ludwig zeigten die Erfahrungen, daß weniger als 10% der Teilnehmer solche komplexen Computer verstanden hatten. Somit konnte man einen solchen Computerbau Schülern der Sekundarstufe 1 erst recht nicht zumuten.

Auf der Suche nach einem einfacheren System stieß man auf den Motorola-Baustein 14500. Der Computer, von seinen beiden Konstrukteuren nach dem Gesichtspunkt der didaktischen Vereinfachung konstruiert, wurde von Schülern erfolgreich aufgebaut und programmiert.

Nach Schlagzeilen in der Presse über den erfolgreichen Computerbau durch Schüler beauftragte der Redakteur des WDR-Schulfernsehens, Alexander Rimmelman, die Konstrukteure des Lerncomputers, Drehbücher über sechs Folgen mit dem Titel "Bit und Byte - Wir bauen einen Computer" zu schreiben.

SO IST DIESES BUCH ENTSTANDEN

Nach ersten Veröffentlichungen über den Lerncomputer wurde der Themenbereich als Diplomarbeit an der Fachhochschule Münster an Heinz Schepers und Hermann-Josef Terglane vergeben. Diese Arbeit wurde im Autorenteam zu dem vorliegenden Buch überarbeitet, wobei weitere Veröffentlichungen eingeflossen sind.

DAS AUTORENTEAM

Volker Ludwig und Klaus Paschenda
 Heinz Schepers und Hermann-Josef Terglane
 Dr. Klaus Grannemann
 Burkhard John
 Hermann Komar
 Ludwig Meinersen

EIN STÜCK GESCHICHTE

Den Beginn des modernen "Computer-Zeitalters" markieren die von Konrad Zuse gebaute Z1, Z2 und Z3, die von Howard Aiken konstruierte Rechenmaschine Mark I und die von J.P. Eckert und J.W. Manchley konzipierte Rechanlage ENIAC.

Die Z3 war der erste über Lochkarten gesteuerte elektromagnetische Rechner; Schaltelemente waren Relais. Die von Zuse initiierte Entwicklung konnte in Deutschland infolge des Krieges aber erst Jahre später wieder aufgegriffen werden. Anders in den USA, hier wurden die neuen Ideen sofort von der Industrie umgesetzt.

Der Bau der Mark I wurde beispielsweise durch den damaligen Chef von IBM, Thomas J. Watson, ermöglicht; er steckte eine Million Dollar in das Projekt.

ENIAC war die erste elektronische Großrechenanlage. Sie enthielt zirka 18000 Röhren und 1500 Relais, wog 30t, nahm einen ganzen Saal in Anspruch und verbrauchte 174kW. Ihre Programmierung erfolgte über Schalttafeln, die je nach Programm unterschiedlich verdrahtet wurden.

LINEARE LOCHSTREIFEN

Auch die Z3 wurde wie alle damaligen Rechner extern gesteuert. Ihr Programm war auf Lochstreifen codiert. Der Lochstreifen wurde abgspult, so daß der Anlage sämtliche Operationen nacheinander eingegeben wurden und sie das Programm linear abarbeiten konnte. Verzweigungen, logische Abfragen waren nicht möglich.

Moderne Computer weisen hingegen eine flexible Speicherprogrammierung auf. Dabei werden die einzelnen Befehle codiert, mit einer "Adresse" versehen und in einem Speicher abgelegt.

Die Codierung erfolgt üblicherweise in binärer Form. Der Rechner kann nun die Anweisungen linear abarbeiten; er kann aber auch logische Entscheidungen, eine Auswahl treffen, wenn bestimmte Bedingungen erfüllt oder nicht erfüllt sind. Einzelne Programmteile können so übersprungen, ausgelassen werden, im Programm können Verzweigungen enthalten sein.

Die grundlegenden Kennzeichen speicher-programmierter Rechner, die der Mathematiker von Neumann als erster in seinem Entwurf eines Universalrechners, der "Princeton-Maschine" formulierte, wollen wir am Beispiel dieser Maschine erläutern.

WIRKSAME WERKE

Der grundlegende Aufbau der meisten heutigen Rechenanlagen entspricht dem Konzept der Princeton-Maschine. Sie läßt sich durch folgende Beschreibung kennzeichnen:

Die Von-Neumann-Maschine besteht prinzipiell aus fünf Funktionseinheiten, sogenannten "Werken" oder "Einheiten", und zwar aus dem

- Rechen- und Logikwerk
- dem Leitwerk
- dem Speicherwerk
- dem Eingabewerk und
- dem Ausgabewerk.

Die ersten zwei Werke faßt man unter dem Begriff "Zentraleinheit" zusammen. Das Rechen- und Logikwerk führt arithmetische und logische Operationen durch. Das Leitwerk, auch Steuerwerk genannt, steuert den Austausch von Daten und Befehlen zwischen dem Rechen- und Logikwerk und den anderen Funktionseinheiten, es sichert einen geordneten Programmablauf. Wie dies im einzelnen abläuft, werden wir an späterer Stelle an unserem Prozessor beispielhaft erklären.

Die Übertragungswege, über die Daten zwischen den Funktionseinheiten und der Zentraleinheit ausgetauscht werden, nennt man Busse.

Alle Daten und Befehle, die von der Maschine verarbeitet werden sollen, müssen binär codiert sein. Sie werden als Folge von Nullen und Einsen dargestellt, weil sich dies physikalisch leicht realisieren läßt.

ZAHLREICHE ZELLEN

Im Unterschied zu Rechnern mit externer Programmsteuerung werden hier das Programm und die Daten in einem gemeinsamen Speicher abgelegt, in sogenannten "Zellen". Damit jederzeit auf die Zelleninhalte zugegriffen werden kann, wird jeder einzelnen Zelle eine bestimmte "Adresse", eine Nummer, zugeordnet. Die Zellen werden dazu fortlaufend durchnummeriert.

Ein solcher Speicher ist mit einem Regal vergleichbar, das eine gewisse Anzahl von Schließfächern enthält. Jedes Fach ist durch eine im Dualsystem dargestellte Zahl eindeutig gekennzeichnet. Ist beispielsweise die Zahl 0011 im Fach mit der Adresse 0010 gespeichert, kann der Speicherinhalt jederzeit abgerufen werden. Adresse und Inhalt einer Speicherzelle sind also zu unterscheiden.

Da die Befehle eines Programmes immer in aufeinanderfolgenden Zellen gespeichert werden, muß nur die Adresse des ersten Befehls bekannt sein, um das Programm starten und abzuarbeiten zu können. Die Adresse des nächsten Befehls liefert ein Programmzähler. Er erhöht die alte Adresse jeweils um Eins. Dies entspricht einem linearen Programmablauf.

Diese sequentielle Abarbeitung eines Programmes kann durch "Sprungbefehle" durchbrochen werden. Es gibt bedingte und unbedingte Sprungbefehle. Ein bedingter Sprung wird nur dann ausgeführt, wenn eine bestimmte Bedingung erfüllt ist. Andernfalls zählt der Programmzähler wie üblich weiter. Liegt hingegen ein unbedingter Sprung vor, wird der Programmzähler automatisch mit der neuen Adresse geladen.

Diese Eigenschaften, die für die Von-Neumannsche Struktur speicher-programmierter Rechner charakteristisch sind, werden von dem 1-Bit-Computer alle realisiert.

AUS EINER INNOVATION WIRD EINE NEUE INDUSTRIELLE REVOLUTION

Die zweite industrielle Revolution geht auf die Erfindung des Transistors im Jahre 1948 zurück. Dieses Bauelement hat nach und nach die Elektronenröhre ersetzt. Der Ablöseprozess verlief zunächst eher langsam. Erst mit dem Übergang zum Silizium als Grundmaterial begann eine explosionsartige Entwicklung, wie sie in der Technik bisher ohne Beispiel ist. Etwa um 1960 wurde nämlich die Möglichkeit erkannt, daß man, statt einzelne Transistoren herzustellen und zu verdrahten, auch auf einem kleinen Silizium-Chip mehrere mikroskopisch kleine Komponenten zusammenfassen ("integrieren") kann. Dies war der Beginn der Mikroelektronik.

Man vermutet, daß die Mikroelektronik den Motor zur dritten industriellen Revolution darstellt. Es gibt kaum einen Bereich, der nicht von der Entwicklung der Mikroelektronik beeinflusst wird. Von der Kommunikations- und Datentechnik bis zur Unterhaltungselektronik, von der Prozeßautomatisierung bis zur Medizin- und Verfahrenstechnik, Überall wird der technische Fortschritt durch die Mikroelektronik geprägt.

Ob nun zweite oder dritte industrielle Revolution, offensichtlich vollzieht sich momentan ein sprunghafter Wandel in Industrie, Dienstleistungsgewerbe und Handel. Treibende Kraft dieses Wandels ist die Mikroelektronik mit den damit verbundenen neuen Techniken und Produkten.

Weil dieser Wandel einschneidende Auswirkungen auf das berufliche, das gesellschaftliche und private Leben aller, insbesondere aber der jungen Menschen haben wird, muß auch die Schule auf diese Herausforderung reagieren.

DER NEUE BILDUNGSaufTRAG

Es gehört heute "zum Bildungsauftrag der Schule, allen Schülern frühzeitig - nach Möglichkeit bereits in der Sekundarstufe I - entsprechende Grundlagen in geeigneter Form zu vermitteln."

Über die konkrete Ausgestaltung, die geeignete Form, dieses Bildungsauftrages liegen sich die Pädagogen und Didaktiker noch im Clinch, im Computer-Clinch.

Schaut man sich in den einzelnen Bundesländern um, zeigt sich, daß es nur für die Sekundarstufe II in allen Bundesländern ein eigenständiges Unterrichtsfach "Informatik" gibt. Über eine Grundbildung Informatik in der Sekundarstufe I, die immerhin von ca. 70% aller Schüler besucht wird, denken die meisten Kultusminister noch nach. Ausnahme sind die Länder Bayern und Baden-Württemberg sowie Berlin, in denen es schon fixierte Vorstellungen gibt.

UNSER RAHMENKONZEPT

Gegenstand unseres Konzeptes ist die Einführung einer informationstechnischen Bildung für alle Schüler und Jugendlichen. Analog zu unserem Schul- bzw. Bildungssystem ist es gegliedert in :

- * eine informationstechnische Grundbildung für alle Schüler
- * eine vertiefende informationstechnische Bildung in Form der Informatik
- * eine berufsbezogene informationstechnische Bildung.

In der Sekundarstufe I soll - entsprechend dieser Empfehlung - die informationstechnische Grundbildung im Rahmen bestehender Unterrichtsfächer vermittelt werden. Infrage kommen dabei solche Fächer, "die induktiven, deduktiven und statistischen Methoden Vorrang einräumen und sich formalisierter Verfahren und Darstellungen bedienen."

Lange Rede, kurze Interpretation: In erster Linie sind damit wohl Physik, Mathematik und Technik gemeint.

Ein eigenständiges Fach Informatik bleibt der gymnasialen Oberstufe vorbehalten, was nicht ausschließt, "daß Informatikangebote auch schon in den Wahlpflichtunterricht des Sekundarbereichs I aufgenommen werden."

Um dies für die Sekundarstufe I ansatzweise zu lösen, werfen wir einen Blick in die Richtlinien für den Informatikunterricht der gymnasialen Oberstufe (Richtlinien für die gymnasiale Oberstufe in Nordrhein-Westfalen, Informatik, Hrsg.: Der Kultusminister des Landes NRW, Köln 1981, S. 29). Dort finden wir die folgenden vier Lernbereiche:

- Lernbereich I:
Algorithmik
- Lernbereich II:
Daten und Datenstrukturen
- Lernbereich III:
Hard- und Softwaresysteme
- Lernbereich IV:
Realisierung, Probleme und Auswirkungen der praktischen Datenverarbeitung.

WIE SIEHT DAS DIDAKTISCHE KONZEPT DIESES PROJEKTES AUS?

Entmystifizierung des Computers ist hier kein bloßes Schlagwort, hier wird ernst damit gemacht, und zwar handgreiflich. Es wird gebohrt, gelötet, verdrahtet und getestet. Nicht graue Theorie über das Innenleben einer CPU wird vermittelt, sondern handfestes Wissen, das Schülern der Sekundarstufe I zumutbar ist. Ein üblicher 8-Bit-Prozessor ist halt sehr viel komplexer aufgebaut als ein 1-Bit-Prozessor. Um beispielsweise die grundsätzlichen Komponenten eines Z-80 kennenzulernen, bedarf es schon eines umfangreichen Vokabulars:

Computer Control Unit, Program Control Unit, Programmzähler, Adreß Register, Akkumulator, Zwischenspeicher, Index-Register, Stackpointer, Flag Register, Takt, Steuer-Bus, Daten-Bus, Adreß-Bus, usw. . Die Erklärung, wie diese Komponenten nun zusammenspielen, wollen wir uns, und vor allem den Schülern, ersparen.

Didaktische Reduktion ist hier gefordert. Dem entspricht die Wahl eines 1-Bit Prozessors, des MC 14500 von Motorola.

WAS KANN DER WDR-1-BIT-COMPUTER?

Im wesentlichen erfüllt er Steueraufgaben. Er kann eine Ampelanlage, einen Roboter, einen Aufzug und andere Geräte steuern. Er kann aber auch zur physikalischen Modellbildung herangezogen werden, wenn Boolesche bzw. Schaltalgebra Unterrichtsthema ist. Die wichtigsten logischen Verknüpfungen, die NICHT-, UND- sowie die ODER-Verknüpfung, sind in der CPU unter anderem realisiert.

Er stellt einen vollwertigen Mikroprozessor dar, dessen Funktionsweise Schülern auf Grund seines relativ einfachen Aufbaus erklärt werden kann und die auch verstanden wird. Der Prozessor 14500 schließt die Lücke zwischen der einfachen, aber teuren, weil fest verdrahteten Schaltungstechnik und den preiswerten, aber schwer verständlichen Mikroprozessoren. Er ist sozusagen das "Missing link" zwischen diesen beiden Extremen.

Natürlich kann er dafür auch sehr viel weniger als seine größeren Brüder, sein Repertoire umfaßt nur 16 Befehle. Die beherrschen Schüler dann aber auch alle!

Der WDR-1 Bit-Computer ist modular aufgebaut, er besteht aus verschiedenen Funktionseinheiten, auf die an späterer Stelle ausführlich eingegangen wird (Siehe: Kapitel 2) Der modulare Aufbau hat einen Nachteil: Der Computer bzw. der Bausatz wird dadurch teurer. Er wurde aber aus didaktischen bzw. methodischen Gründen gewählt: Der Computer kann so im arbeitsteiligen Gruppenunterricht von den Schülern selbst gebaut werden.

Programmiert wird der Computer in Maschinensprache, mit Tastatur in Assembler. Auch dies ist sinnvoll, wenn die grundsätzliche Arbeitsweise eines Computers im Vordergrund steht. Die wenigsten Schüler werden jemals im Berufsleben eigene professionelle Programme schreiben müssen, weder in Basic, Pascal, Logo, Elan oder sonst einer Sprache. Durch die Maschinensprache erfahren sie aber, wie Computer Daten wirklich verarbeiten.

DIE KONZEPTION DES WDR 1-BIT-COMPUTERS

Ein Computer ist eine funktionsfähige Steuereinheit aus einem Prozessor und Zusatzbausteinen, je nach Bedarf. Ein Prozessor besteht aus einer vollständig programmgesteuerten Rechnerschaltung. Einen besonders einfachen Prozessor stellt der 1-Bit Mikroprozessor MC 14500 dar. Es ist ein 1-Bit-Prozessor, weil er zu einem Zeitpunkt nur 1 Bit verarbeiten kann. 1 Bit, das ist die Information "ja" oder "nein" bzw. "1" oder "0", verwirklicht durch 0 Volt bei "0" und 5 Volt bei "1". Von einem Mikrocomputer spricht man, wenn der Prozessor ein Mikroprozessor ist, der Prozessor also in einer winzigen, mikroskopisch kleinen integrierten Schaltung realisiert ist und die Zusatzbausteine ebenfalls integrierte Schaltungen sind. Moderne Mikrocomputer arbeiten z.B. mit 8 oder 16 Bit-Mikroprozessoren. Sie sind so komplex und kompakt gebaut, daß sie sich für den Bau und die elementare Programmierung eines Computers in der Schule nicht eignen.

Der WDR-1 Bit-Computer dagegen stellt für Schüler der Sekundarstufe I und II das didaktisch Einfache dar, da er auf seine wesentlichen Teile reduziert worden ist. Bei einer weiteren Verringerung von Funktionsteilen würde er als Computer nicht mehr richtig arbeiten können. Für den schulischen Bedarf wurden die Verbindungen zwischen den integrierten Schaltungen im Gegensatz zu den professionellen Mikrocomputern nicht miniaturisiert, sondern so weit vergrößert, daß der Computer für den Schüler wieder greifbar wird. Der WDR-1-Bit-Computer ist ein Computer zum Anfassen, deshalb fehlt ihm auch ein Gehäuse. Eine weitere Vereinfachung stellt sein modularer Aufbau in 4 steckbare Funktionseinheiten dar. Dies bedeutet für den Schüler eine zusätzliche Verringerung des Komplexitätsgrades beim Begreifen des Prinzips eines Mikrocomputers. Die genannte Vereinfachung sowie der modulare Aufbau ermöglichen so einen sehr sicheren Nachbau des WDR-1-Bit-Computers durch Schüler unter Anleitung des Lehrers ohne Verwendung spezieller Hilfsmittel. Der Schüler soll sogar Löcher für die Aufnahme der Bauteile in die Platine selbst bohren. Bei dieser Tätigkeit gewinnt er Vertrauen beim Zusammenbauen eines für ihn zunächst noch unbekanntes Computers mit Hilfe einer ihm bekannten Fertigkeit.

Die Verwendung von nur 14 Befehlen (16 sind möglich) erleichtert zusätzlich das Erlernen der Grundprinzipien der Programmierung eines Computers, da herkömmliche Computer einen Befehlsvorrat von einigen 100 Befehlen aufweisen, deren Kenntnis und Anwendung sehr viele Anwender überfordern. Mit diesen 14 Befehlen können mittels einfachster Programme alle den Computer aufbauenden Elemente simuliert werden. So können seine logischen Bausteine wie UND, ODER, NICHT oder seine Speicher und seine Zähler dargestellt werden: Der Computer erklärt sich selbst.

Es liegt auf der Hand, daß ein für die Schule konzipierter, auf das didaktisch Einfache zurückgeführter Mikrocomputer im Preis-Leistungsverhältnis nicht mit modernen Mikrocomputern konkurrieren kann, zumal er auch in seinem Aufbau bewußt groß gehalten worden ist. Wegen seiner Einfachheit ermöglicht er jedoch schon in der Sek. I einen schülergemäßen und elementaren Einstieg in den Aufbau, die Wirkungsweise und die Programmierung von Mikrocomputern.

WELCHE LERNZIELE KÖNNEN IN EINEM UNTERRICHTSPROJEKT ERREICHT WERDEN?

Wir haben diese Frage gestellt, obwohl wir sie nur "grob" beantworten wollen. Eine Auflistung aller Feinziele wäre lang und wenig sinnvoll. Die Schülergruppe, die die Anzeigeplatine zusammenbaut, wird andere Ziele erreichen als die Gruppe, die sich die Speicherplatine vorgenommen hat. Auf den jeweiligen Schwerpunkt kommt es also an. Begnügen wir uns deshalb mit einigen wenigen (ergänzungsbedürftigen) Grobzielen.

- * Das unseres Erachtens wichtigste Ziel: Die Schüler sollen durch den Bau eines eigenen Computers, am Bau ihres Computers regen Anteil nehmen, "motiviert" sein! In allen bisherigen Unterrichtsprojekten war dies der Fall, obwohl einzelne Platinen nicht immer auf Anhieb funktionierten.
- * Weiter sollen die Schüler ein gewisses handwerkliches Geschick beim Bau erlangen. Dieses Ziel gilt sicherlich weniger für Gymnasiasten, um so wichtiger ist es für Hauptschüler. Für diese Gruppe steht es in engem Zusammenhang mit dem ersten Ziel. Wie weit der Lehrer auf die einzelnen Bauteile des Computers, auf Widerstände, Dioden, Kondensatoren und Chips eingehen soll, muß der Lehrer vor Ort entscheiden. Hier kann nur allgemein gefordert werden, daß die Schüler das Verhalten der wichtigsten Bauelemente des 1-Bit-Computers kennen, insbesondere das des Mikroprozessors kennenlernen.
- * Ein Verständnis des Zusammenwirkens der einzelnen Bauteile und der Funktionsgruppen des Computers ist das nächste Ziel. Dazu gehört ein Vergleich der Grobstruktur des 1-Bit-Computers mit größeren Systemen.
- * Um schließlich programmieren zu können, muß man die Befehle kennen, ihre Codierung und ihre Wirkungsweisen, speziell die vorhandenen logischen Verknüpfungen.

Obwohl sich noch weitere Grobziele auflisten lassen, z.B. Binärsystem, Peripherie etc. heißt unsere letzte Forderung, die Schüler sollen ein Grundverständnis der Komponenten der CPU und ihres Zusammenwirkens erlangen. Dies heißt, sie sollen exemplarisch die Architektur eines Rechners kennenlernen.

DER WDR-COMPUTER-SCHEIN

Wer den 1-Bit-Computer erfolgreich baut, der erhält einen (oder auch mehrere) Computerscheine! Nach Beendigung des erfolgreichen Aufbaus und der Programmierung des WDR-1-Bit-Computers bekommen die Schüler einen Computerschein der Klasse 1 entsprechend 1 Bit. Der Computerschein ist so ausgelegt, daß er auch für andere Klassen ausgegeben werden kann. Der Computerschein ist im Anhang als Kopiervorlage vorgestellt, er kann aber auch farbig in Klassenstärke von der Redaktion des Schulfernsehens des WDR, Appelloplatz 1, 5000 Köln 1 angefordert werden.

2. DER AUFBAU DES WDR-1-BIT-COMPUTERS (HARDWARE)

In diesem Abschnitt soll auf den technischen Aufbau, d.h. auf die Bauteile und ihre Verdrahtung eingegangen werden. Der englische Fachausdruck dafür ist "Hardware". Zunächst wird der prinzipielle Aufbau beschrieben. Abb. 2 zeigt das Blockschaltbild des WDR-1-Bit-Computers. Es soll als Beilage zum Kapitel 2.1 dienen.

2.1. DER PRINZIPIELLE AUFBAU

Der WDR-1-Bit-Computer besitzt vier Eingangsleitungen, die entweder durch Schalter oder über einen Expansionsstecker durch angeschlossene Geräte beschaltet werden können. Auf diesem Wege kommen Signale in den Computer hinein.

Weiter gibt es acht Ausgangsleitungen, die ebenfalls an den Expansionsstecker geführt sind. Über diese Leitungen können Ergebnisse oder Entscheidungen des Computers an angeschlossene Geräte weitergegeben werden, die dann die Informationen in Steuervorgänge umsetzen. Drei dieser Ausgänge sind zusätzlich mit drei weiteren Eingangsleitungen verbunden, so daß der Computer auf seine alten Ergebnisse zurückgreifen kann, d.h., das System hat drei Datenspeicher.

Der Mikroprozessor besitzt ein Ergebnisregister, welches immer das Ergebnis, das durch die Ausführung eines Programmschrittes gewonnen wird, enthält. Dieses Ergebnisregister ist mit noch einer Eingangsleitung verbunden, damit die Information des Ergebnisregisters wieder gelesen werden kann.

Der Computer besitzt also acht Eingangs- und acht Ausgangsleitungen. Zur Kontrolle der Informationen dieser Leitungen sind sie mit jeweils einer Leuchtdiode verbunden, die zusammen einen Teil der Anzeige darstellen.

Die Eingangsleitungen sind mit einem Eingabebaustein und die Ausgangsleitungen mit einem Ausgabebaustein verbunden. Der Eingabebaustein schickt über die Datenleitung die Information einer der acht Eingänge an den Mikroprozessor. Genauso erhält der Ausgabebaustein über dieselbe Datenleitung eine Information vom Mikroprozessor, die er weiter an einen der acht Ausgänge schickt. Über eine weitere Leitung kann der Ausgabebaustein den Auftrag erhalten, diese Information wieder zu löschen. Dazu muß man einen Löschtaster betätigen, der auch mit dem Mikroprozessor verbunden ist und auch dessen Information löscht. Damit gesichert ist, ob der Eingabebaustein eine Information an den Mikroprozessor senden soll oder ob der Ausgabebaustein eine Information vom Mikroprozessor aufnehmen soll, gibt es eine Schreib-/Leseleitung, die den Mikroprozessor mit den beiden Bausteinen verbindet. Beim Auftrag "Lesen" arbeitet nur der Eingabebaustein und beim Auftrag "Schreiben" arbeitet nur der Ausgabebaustein.

Über die Eingabetastatur können die Befehle eingegeben werden. Von den sechzehn möglichen Befehlen sind nur vierzehn ausgeführt, um das System übersichtlicher zu machen. Im Gegensatz zu den üblichen Computern sind die vierzehn Befehle fest verdrahtet, d.h., für jeden Befehl gibt es eine Taste. Dies wäre bei einem Befehlssatz von einigen hundert Anweisungen, wie sie größere Computer haben, nicht mehr zweckmäßig.

Damit der Prozessor die vierzehn im System verwirklichten Befehle unterscheiden kann, werden vier Leitungen benötigt.

Je nachdem, ob eine Leitung "0" oder "1" führt, sind $2^4=16$ Möglichkeiten vorhanden, wie folgende Darstellung im Dualsystem es zeigt: 0000; 0001; 0010; 0011; 0100; 0101; 0110; 0111; 1000; 1001; 1010; 1011; 1100; 1101; 1110; 1111.

Die meisten Befehle beziehen sich auf einen der acht Eingänge oder einen der acht Ausgänge, die somit zu Adressen werden. Daher gibt es acht weitere Tasten, die die Nummern der Ein- und Ausgänge enthalten. Diese Ein-/Ausgabeadressen sind also ebenfalls fest verdrahtet. Für sie werden drei Leitungen zur Unterscheidung der Adressen benötigt ($2^3=8$): 000; 001; 010; 011; 100; 101; 110; 111. Sie bedeuten in dieser Reihenfolge: 0; 1; 2; 3; 4; 5; 6; 7. Dieses Binärsystem wird auch Dualsystem genannt. Jede Stelle besitzt eine Wertigkeit. Die letzte Stelle, d.h. das rechte Bit, hat die Wertigkeit 1. Nach vorne verdoppelt sich die Wertigkeit pro Stelle. Sie ist wichtig bei der Umrechnung einer Dualzahl in eine Dezimalzahl. Eine "0" bedeutet kein Wert vorhanden und eine "1" bedeutet Wert vorhanden. Die vorhandenen Werte werden addiert. Die Dualzahl 101 z.B. ist die Dezimalzahl 5: $1*4+0*2+1*1=5$.

Eine vierte Leitung existiert, ist aber nicht mit der Eingabetastatur verbunden. Sie kann zur Erweiterung der Ein-/Ausgabeadressen dienen, denn mit vier Leitungen können sechzehn Ein-/Ausgabeadressen unterschieden werden.

Die Eingabetastatur enthält also vierzehn Tasten für die Befehle und acht Tasten für die Ein-/Ausgabeadressen. Mit Hilfe dieser zweiundzwanzig Tasten und den sieben Leitungen können nun Programme erstellt werden. Wie bei den Ein- und Ausgängen sind die sieben Leitungen mit jeweils einer Leuchtdiode verbunden. Diese zeigen an, um welchen Befehl und um welche Ein-/Ausgabeadresse es sich bei den einzelnen Programmschritten handelt.

Ein Programm bekommt erst seine große Bedeutung, wenn man die einzelnen Programmschritte speichern kann. Da ein Programmschritt aus einem Befehl, durch vier Leitungen codiert, und einer Ein-/Ausgabeadresse, durch drei Leitungen codiert, besteht, müssen die Programmspeicher sieben Informationen parallel, d.h. gleichzeitig, aufnehmen.

Die Speicherbausteine in dem WDR-1-Bit-Computer sind jeweils 256 x 4-Bit-Speicher. Die zweite Zahl, d.h. die "4", bezeichnet die Anzahl der Bits, die der Programmspeicher zu einem Zeitpunkt parallel aufnehmen kann. Bei sieben Informationen brauchen wir somit zwei Speicherbausteine, so daß wir einen 256 x 8-Bit-Speicher erhalten. Die erste Zahl, d.h. die "256", bezeichnet die Anzahl der Bits, die der Programmspeicher auf einem der acht Eingänge seriell, d.h. nacheinander unter verschiedenen Programmadressen, aufnehmen kann. Sie gibt also die maximale Anzahl der Programmschritte an, die gespeichert werden können. In diesem Fall darf ein Programm nicht mehr als 256 Programmschritte haben.

Der eine Programmspeicher beinhaltet die Befehle und der andere Programmspeicher enthält die Ein-/Ausgabeadressen. Das Einlesen erfolgt mit Hilfe eines Schreib-Tasters. Der Befehlsspeicher ist mit dem Mikroprozessor verbunden und der Adressenspeicher ist mit dem Eingabe- und dem Ausgabebaustein verbunden.

Damit die einzelnen Programmschritte in der richtigen Reihenfolge gespeichert sind und bei Bedarf in eben dieser

Reihenfolge ausgegeben werden können, benötigen wir einen Programmzähler. Jeder Programmschritt hat eine eigene Programmadresse. Wir müssen 256 verschiedene Adressen anwählen können. Dies ist mit einem 8-Bit-Binärzähler zu realisieren ($2^8=256$). Beide Programmspeicher sind mit diesem Programmzähler über acht Leitungen verbunden. In dem 1-Bit-Computer sind zwei 4-Bit-Binärzähler hintereinander geschaltet.

Normalerweise beginnt der 8-Bit-Zähler bei der Adresse 0 ($0 \cdot 128 + 0 \cdot 64 + 0 \cdot 32 + 0 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 = 0$) und zählt im Dualsystem hoch bis zur Adresse 255 ($1 \cdot 128 + 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 255$). Danach fängt er wieder bei 0 an. Man kann jedoch mit Hilfe eines Tasters jederzeit zum Zählerstand 0 zurückkehren. Dieser Tastendruck kann aber auch durch einen Programmschritt, bzw. durch einen Befehl, ersetzt werden, den man z.B. am Ende eines Programmes setzt.

Damit der Programmzähler zählen kann, braucht er einen Takt. Diesen Takt bekommt er vom Mikroprozessor. Durch einen Taktwahlschalter kann die Schnelligkeit des Taktes eingestellt werden. Dabei gibt es drei Stufen:

In der Stellung "Schnell-Takt" kommt der Takt allein vom Mikroprozessor.

In der Stellung "Langsam-Takt" wird das Taktsignal eines Timers wirksam. Über ein Potentiometer (einstellbarer Widerstand) kann der Takt noch stufenlos variiert werden.

In der Stellung "Hand-Takt" kommt nur dann ein Taktimpuls, wenn man einen dafür vorgesehenen Taster betätigt.

Eine Leuchtdiode zeigt uns die Taktimpulse an. Außer dem Programmzähler ist noch der Eingabebaustein mit dem Takt verbunden.

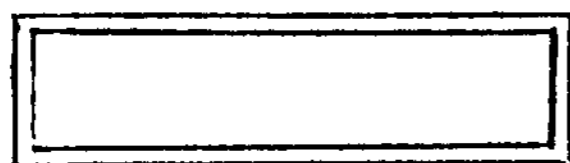
Die Anzeigeeinheit besteht aus vier Bausteinen und 27 Leuchtdioden. Fünfzehn Leuchtdioden zeigen den Informationsgehalt der Ein- und Ausgänge, vier den Befehlscode, drei die Nummer der Ein- oder Ausgabelitung im Dualsystem, eine den Inhalt des Ergebnisregisters und eine den Takt an. Eine Leuchtdiode ist dann noch mit der Datenleitung verbunden.

Es wurde hier bewußt auf englische Fachausdrücke der Datenverarbeitung verzichtet, um dem Anfänger den Eintritt in die Mikroprozessortechnik zu erleichtern.

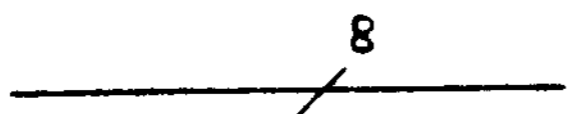
Abb. 1: Erläuterungen zu den Abb. 2 und 3:



Funktionseinheiten mit integrierten Bausteinen



Bedienungselemente



8 Leitungen

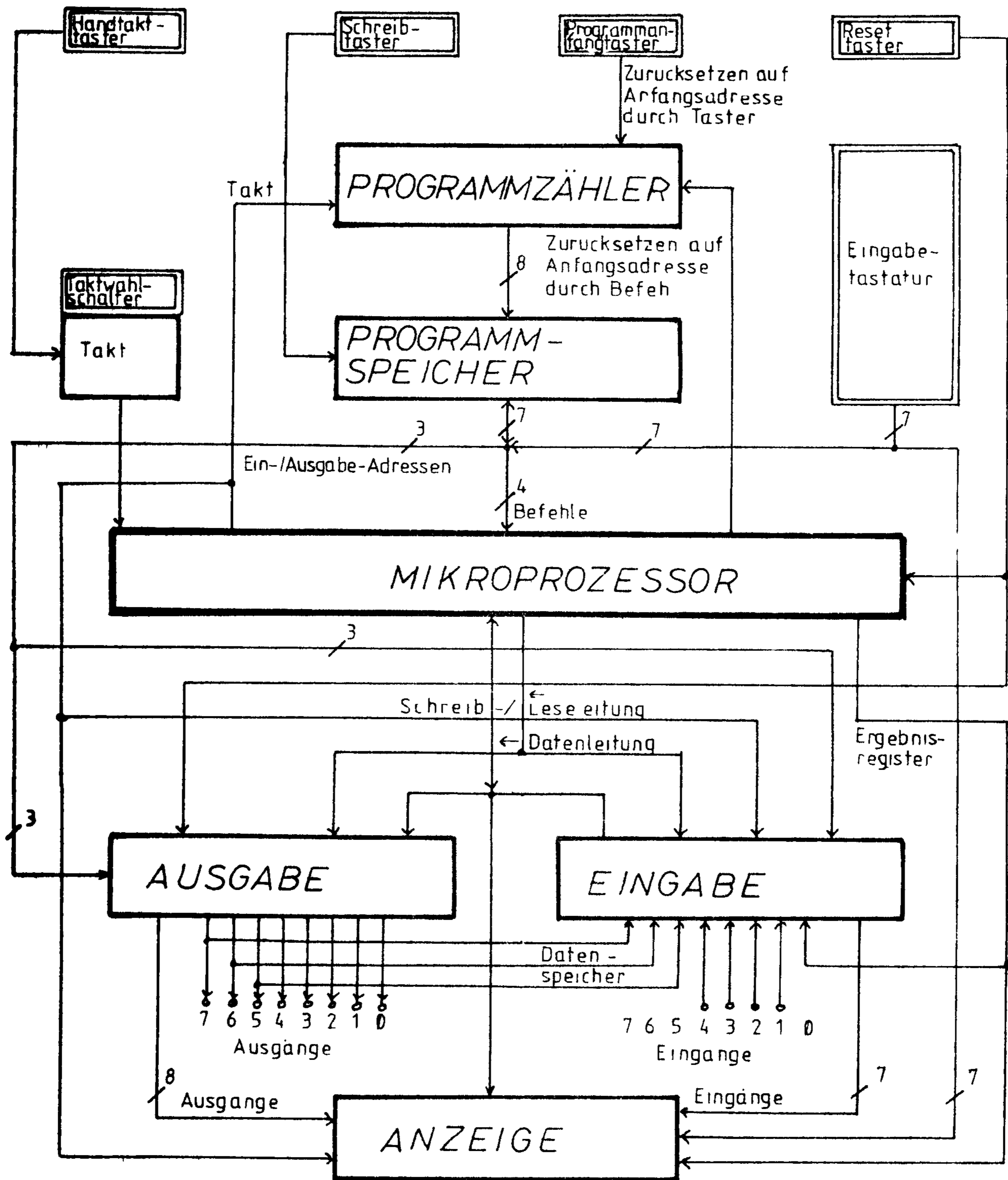


Abb. 2: Blockschaltbild des WDR-1-Bit-Computers

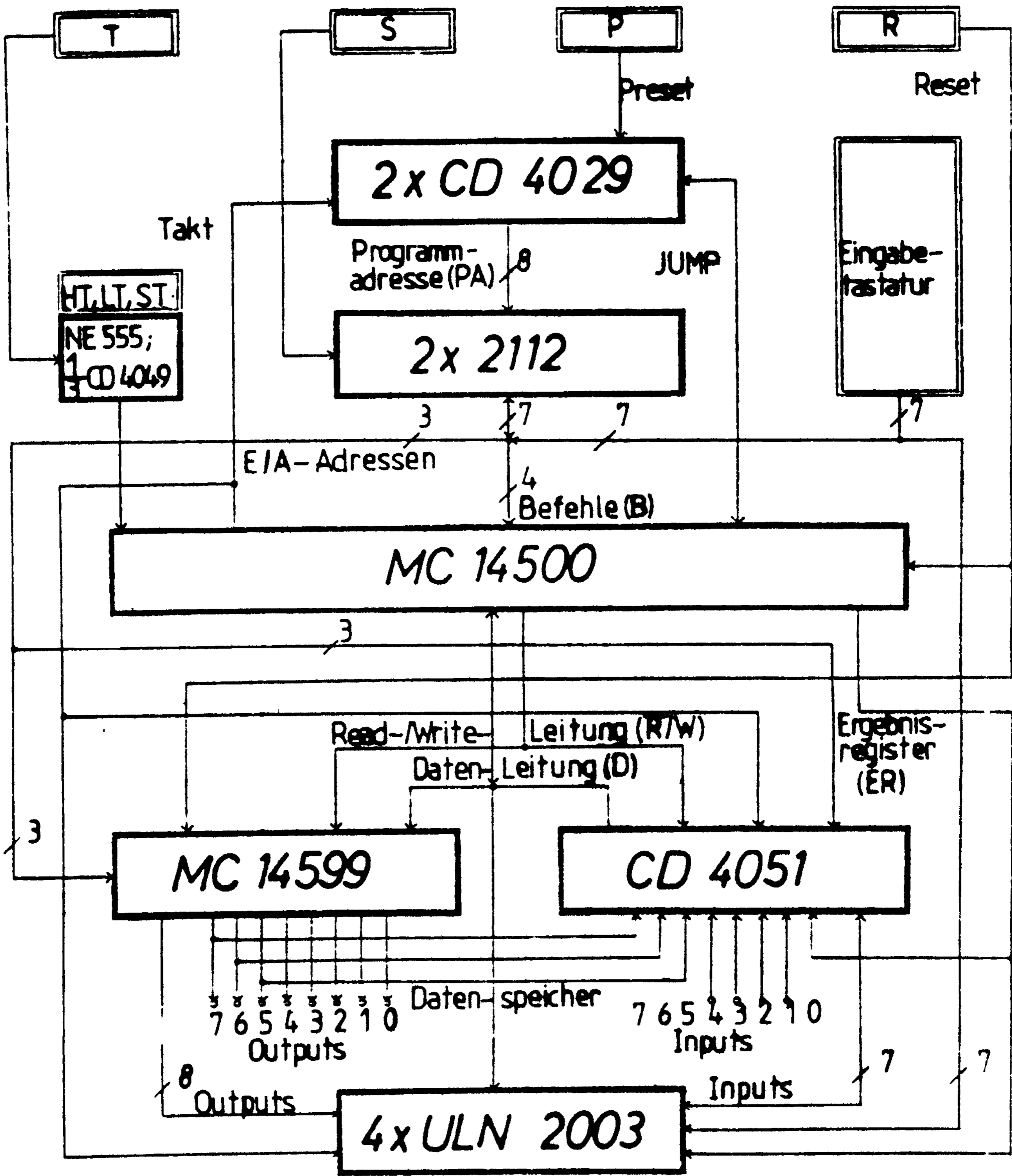


Abb. 3: Blockschaltbild des WDR-1-Bit-Computers (technisch)

Abb. 3 zeigt das gleiche Blockschaltbild wie Abb. 2. Hier sind nur teilweise die deutschen Ausdrücke durch die englischen Fachausdrücke ersetzt. Die dafür verwendeten Abkürzungen stehen in Klammern dahinter. Außerdem sind in Abb. 3 die Funktionseinheiten durch die Typenbezeichnungen, d.h. durch die Typennummern, der integrierten Bausteine ersetzt.

Durch zwei Blockschaltbilder wird dem Anfänger der Zugang zu den englischen Fachausdrücken erleichtert. Zusätzlich kann er schnell nachschauen, welche Aufgabe z.B. der Baustein mit der Typennummer 4051 hat.

2.2. DIE FUNKTIONSEINHEITEN

2.2.1. DER MIKROPROZESSOR MC 14500

Der wichtigste Baustein in einem Computersystem ist der Mikroprozessor. Er wird auch zentrale Datenverarbeitungseinheit (engl. Central Processing Unit - CPU) genannt. Die CPU MC 14500 ist erheblich einfacher aufgebaut als ihre 8-Bit-Kollegen. Abb. 4 zeigt die Pinbelegung des MC 14500.

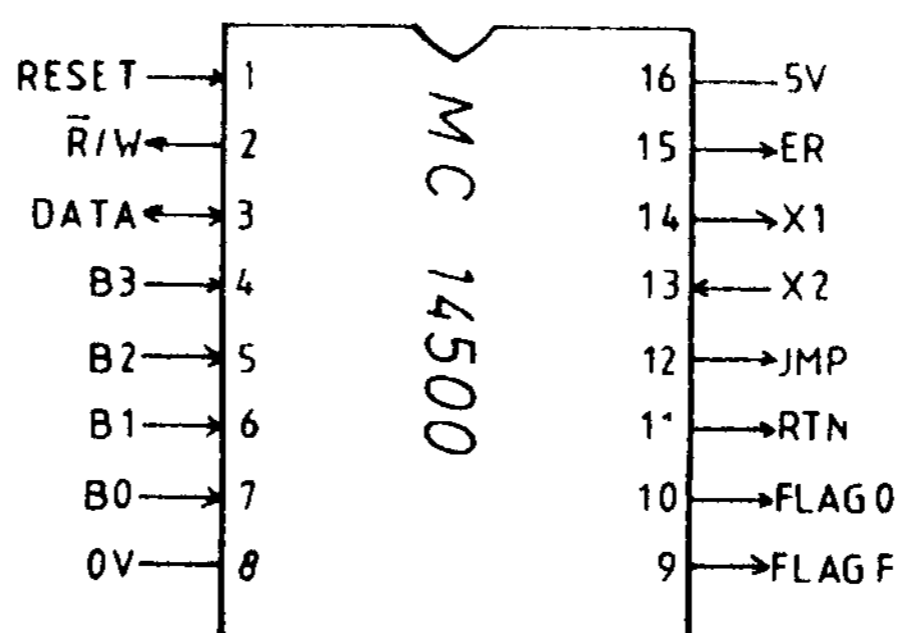


Abb. 4: Pinbelegung des MC 14500




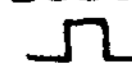
Bedeutung der einzelnen Beine:

- Pin 1: RESET Rücksetzeingang:
Normalerweise ist dieses Bein über einen Widerstand mit 0 V verbunden. Verbindet man es aber mit Hilfe eines Tasters (Reset-Taster) mit 5 V, dann werden alle Informationen in der CPU gelöscht, und diese geht in die Ruhstellung über.
- Pin 2: \bar{R}/W Schreib /Leseleitung:
Ausgangsleitung, die angibt, ob die CPU gerade eine Information über ihre Datenleitung hereinholt ("Lesen") oder ob sie über diese Leitung Daten ausgibt ("Schreiben"). Im ersten Fall ist die Spannung an diesem Bein 0 V, während bei der Ausgabe 5 V anliegen. Der Querbalken über dem R ist ein Negationszeichen und bedeutet hier "0"-aktiv. (Kein Querbalken bedeutet dann "1"-aktiv.) Er soll hier ausdrücken, daß die CPU bei 0 V "liest" und bei 5 V "schreibt".
- Pin 3: DATA Datenleitung:
Leitung, über die die Informationen sowohl beim Lesen als auch beim Schreiben von dem Eingabebaustein bzw. zu dem Ausgabebausteins des Computers geleitet werden. Da dieser Vorgang auf einer Leitung in beide Richtungen erfolgt, hat die Leitung auch den Namen "Bidirektionaler Datenbus".
- Pin 4: B3,
- Pin 5: B2,
- Pin 6: B1,
- Pin 7: B0 Befehlsleitungen:
Anschluß von Leitungen, die die Befehle vom Programmspeicher an die CPU heranzuführen.
- Pin 8: 0 V Masseanschluß
- Pin 9: FLAG F Kennzeichen für den NOP F-Befehl:
Ausgabe eines 5 V Impulses (Steuersignal) aus der CPU, wenn der Befehl NOP F ausgeführt wird. Sonst liegen hier 0 V an.
Dieses Bein ist im WDR-1 Bit-Computer nicht beschaltet.
- Pin 10: FLAG O Kennzeichen für den NOP O-Befehl:
Ausgabe eines 5 V-Impulses aus der CPU, wenn der Befehl NOP O ausgeführt wird. Sonst liegen auch hier 0 V an. Dieses Bein ist auch nicht beschaltet.
- Pin 11: RTN ReTurN, Zurückkehren:
Hier tritt ein 5 V Impuls beim Befehl RTN auf:
Dieses Bein ist auch nicht beschaltet.
- Pin 12: JMP JuMP, Springen:
Hier tritt ein 5 V-Impuls beim Befehl JMP auf. Dieser wird zum Rücksetzen des Programmzählers benutzt:

- Pin 13: X2 Takteingang für den Taktoszillator der CPU:
Hier wird von außen ein Takt eingepreßt.
- Pin 14: X1 Taktausgang des Taktoszillators der CPU:
Bausteine des Computers, die einen Takt zur Synchronisation benötigen, sind mit diesem Taktausgang verbunden.
- Pin 15: ER Ergebnisregister:
Ausgang, an den das Ergebnisregister der CPU geführt ist. Dort steht immer das Ergebnis, das durch die Ausführung des letzten Programmschrittes gewonnen wird.
- Pin 16: 5 V Betriebsspannung

Zum besseren Verständnis des 1-Bit-Computers soll der innere Aufbau der CPU an einem Prinzipschaltbild (Abb. 5) erläutert werden. Dafür muß zunächst der Befehlssatz der CPU kennengelernt werden. Die CPU kann sechzehn verschiedene Befehle ausführen. Jeder hat ein Codewort, das durch eine vierstellige Zahl im Dualsystem ausgedrückt wird. Hier soll nur kurz auf die Wirkung der einzelnen Befehle eingegangen werden. Eine ausführliche Beschreibung erfolgt im Kapitel 4.1.

Der Befehlssatz der CPU:

Befehlscode	Befehl	Befehlsart	Wirkung des Befehls
B3 B2 B1 B0			
0 0 0 0	NOP 0	Steuerbefehl	FLAG 0 = 
0 0 0 1	LD	Eingabebefehl	ER = DATA
0 0 1 0	LDC	Eingabebefehl	ER = $\overline{\text{DATA}}$
0 0 1 1	AND	Rechenbefehl	ER = ER * DATA
0 1 0 0	ANDC	Rechenbefehl	ER = ER * $\overline{\text{DATA}}$
0 1 0 1	OR	Rechenbefehl	ER = ER + DATA
0 1 1 0	ORC	Rechenbefehl	ER = ER + $\overline{\text{DATA}}$
0 1 1 1	XNOR	Rechenbefehl	ER = ER * DATA + $\overline{\text{ER * DATA}}$
1 0 0 0	STO	Ausgabebefehl	DATA = ER
1 0 0 1	STOC	Ausgabebefehl	DATA = $\overline{\text{ER}}$
1 0 1 0	IEN	Freigabebefehl	IEN-Register = DATA
1 0 1 1	OEN	Freigabebefehl	OEN-Register = DATA
1 1 0 0	JMP	Steuerbefehl	JMP FLAG = 
1 1 0 1	RTN	Steuerbefehl	RTN FLAG = 
1 1 1 0	SKZ	Programmverzweigungsbefehl	Falls ER = 0: übernächster Befehl Falls ER = 1: nächster Befehl
1 1 1 1	NOP F	Steuerbefehl	FLAG F = 

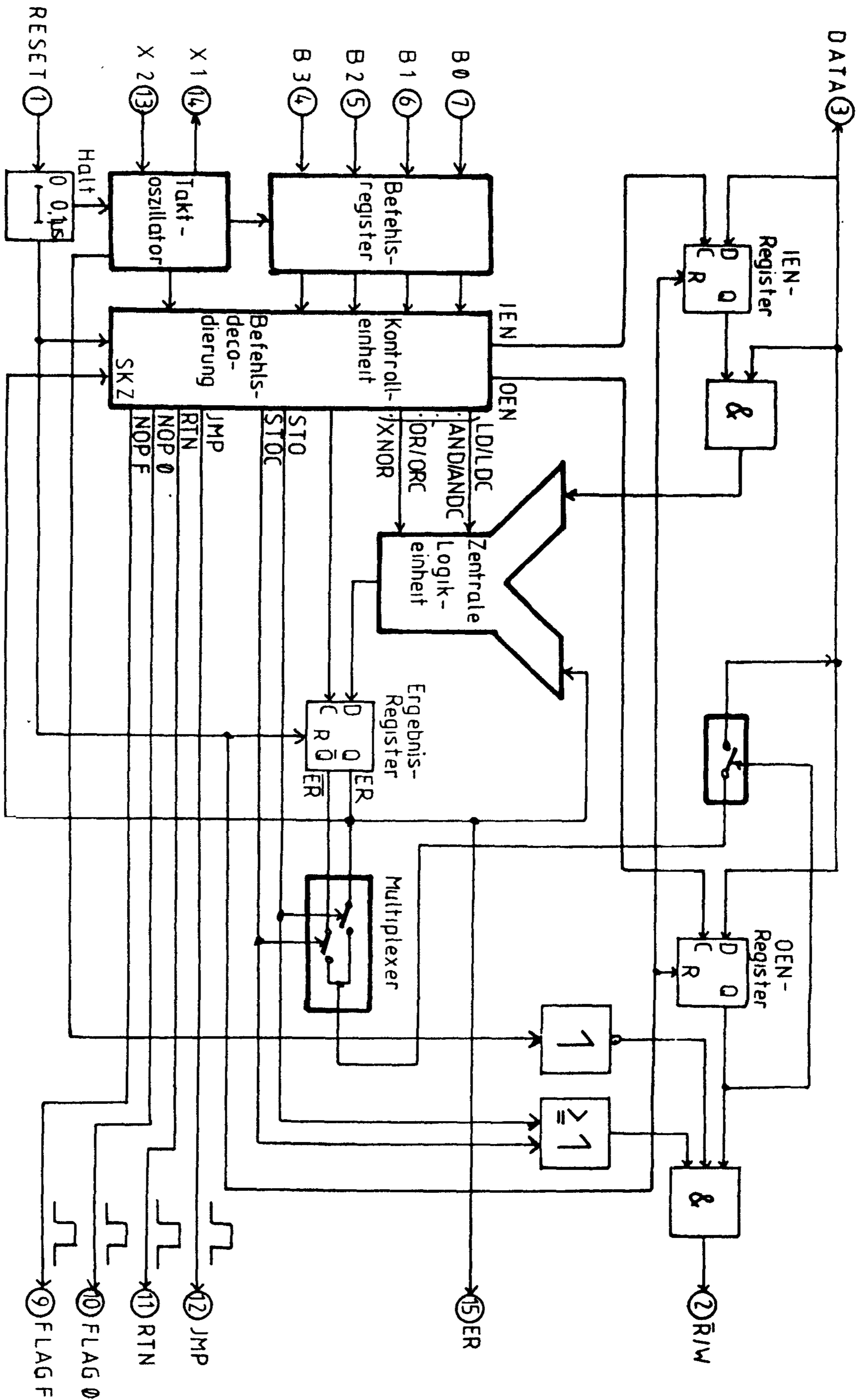


Abb. 5: Prinzipschaltbild des Mikroprozessors MC 14500

In der letzten Spalte stehen die Wirkungen der Befehle als arithmetische Anweisungen. Eine arithmetische Anweisung enthält von links nach rechts:

1. den Namen einer Speicherstelle (Register oder Leitung), die das Ergebnis aufnimmt,
2. das Zuordnungszeichen (kein Gleichheitszeichen !) "=",
3. einen arithmetischen Ausdruck aus Namen der Speicherstellen und Rechenzeichen ("*" bedeutet logische UND-Verknüpfung, "+" bedeutet logische ODER-Verknüpfung) oder das Symbol für einen 5 V-Impuls.

Die Negationszeichen bedeuten hier nicht "0"-aktiv, sondern geben das Komplement, d.h. das Gegenteil, einer Information an. Das Komplement von 0 ist 1 und umgekehrt.

Die Elektronik innerhalb der CPU besteht aus folgenden Teilen:

1. Taktoszillator

Der Takt wird in einem eigenen Kapitel (Siehe: Kapitel 2.2.8.) beschrieben

2. Befehlsregister

Die Befehle werden der CPU durch die vier Befehlseingänge (B0, B1, B2 und B3) zugeführt und mit der negativen Taktflanke von X1 ins Befehlsregister übernommen.

Negative Taktflanke bedeutet Übergang von 5 V auf 0 V des Taktes. Dies geschieht z.B. in der Stellung "Hand-Takt" beim Drücken des Takt-Tasters.

3. Kontrolleinheit

Die Befehle gelangen vom Befehlsregister zur Kontrolleinheit und werden dort decodiert. Liegt ein Steuerbefehl an, so wird ein 5 V-Impuls mit der Dauer einer Taktperiode, beginnend mit der negativen Taktflanke von X1, erzeugt. Dieses Steuersignal (FLAG) erscheint, je nach Befehl, an einem der vier Steuerausgänge (Pin 9...12). Ist der Befehl ein Logikbefehl oder ein Eingabebefehl, so wird er zur Zentralen Logikeinheit weitergeleitet.

Von besonderer Charakteristik ist der Programmverzweigungsbefehl SKZ. Er hat folgende Wirkung: Steht das Ergebnisregister auf "1", so ist er bedeutungslos, d.h., das Programm läuft normal weiter.

Steht das Ergebnisregister auf "0", so wird der nächste Befehl ignoriert, und das Programm läuft bei dem übernächsten Befehl weiter. Die Entscheidung, ob der nächste Befehl ausgeführt wird oder ob er übersprungen wird, findet in der Kontrolleinheit statt. Zu diesem Zweck ist der Ausgang Q des Ergebnisregisters mit der Kontrolleinheit verbunden.

4. Zentrale Logikeinheit

Die Zentrale Logikeinheit hat als Eingänge die Datenleitung und den Ausgang des Ergebnisregisters. Die beiden Bits, die über diese Eingänge in die Zentrale Logikeinheit gelangen, werden durch eine logische Funktion, vorgegeben durch den jeweiligen logischen Befehl, verknüpft. Das Ergebnis dieser Operation wird an das Ergebnisregister weitergeleitet. Bei einem Eingabebefehl wird der Inhalt der Datenleitung bzw. dessen Komplement an das Ergebnisregister weitergeleitet.

5. Ergebnisregister

Das Ergebnisregister speichert immer das Ergebnis der letzten Logikoperation (beim Logikbefehl), bzw. die Information der Datenleitung (beim Eingabebefehl). Das Ergebnisregister ist ein D-Flipflop (Verzögerungs-Flipflop, engl. Delay-D). Die jeweils am Eingang D liegende Information (hier: das Ergebnis der Zentralen Logikeinheit) wird bei einem D-Flipflop zu einem bestimmten Zeitpunkt (hier: bei der positiven Taktflanke von X1 am Takteingang C) eingespeichert und erscheint am Ausgang Q. Der invertierte Ausgang wird mit \bar{Q} bezeichnet. Der Ausgang Q des Ergebnisregisters ist an Pin 15 geführt. Positive Taktflanke bedeutet Übergang von 0 V auf 5 V des Taktes. Dies geschieht z.B. in der Stellung "Hand-Takt" beim Loslassen des Takt-Tasters.

6. Die Eingabeschaltung und das IEN-Register

Das IEN-Register ist ein Eingangs-Freigabe-Register (engl. Input ENable-IEN). Es ist ein D-Flipflop mit dem Eingang D, verbunden mit der Datenleitung, und dem Eingang C, verbunden mit der Kontrolleinheit. Beim Befehl IEN kommt an Eingang C ein Taktimpuls. Nur dann wird die Information der Datenleitung am Eingang D eingelesen, gespeichert und zum Ausgang Q durchgeschaltet. Wie beim Ergebnisregister wird die Information bei der positiven Taktflanke von X1 eingespeichert. Der Inhalt des IEN-Registers kann nur durch einen neuen IEN-Befehl geändert werden.

Der Ausgang des IEN-Registers ist mit dem Eingang eines UND-Gliedes (Schaltzeichen: &) verbunden. Am zweiten Eingang des UND-Gliedes liegt die Datenleitung. Ist die Eingabe nicht freigegeben, d.h., liegt am Ausgang des IEN-Registers eine "0", so erscheint auch am Ausgang des UND-Gliedes eine "0", unabhängig vom Zustand der Datenleitung. Ist die Eingabe freigegeben, d.h., liegt am Ausgang des IEN-Registers eine "1", so erscheint am Ausgang des UND-Gliedes immer der Zustand der Datenleitung. Eine Wahrheitstabelle soll diese Zusammenhänge noch einmal verdeutlichen:

Eingänge des UND-Gliedes		Ausgang des UND-Gliedes
IEN-Register	Datenleitung	
0	0	0
0	1	0
1	0	0
1	1	1

Das IEN-Register gibt also den Datenweg zur Zentralen Logikeinheit frei, wenn es eine "1" speichert. Hervorzuheben ist, daß das IEN-Register, genau wie das im folgenden beschriebene OEN-Register, über die Datenleitung geladen wird.

7. Die Ausgabeschaltung und das OEN-Register

Das OEN-Register ist ein Ausgangs-Freigabe-Register (engl. Output ENable - OEN). Dieses D-Flipflop arbeitet wie das IEN-Register. Nur erscheint am Eingang C ein Taktimpuls beim Befehl OEN. Durch eine "1" in diesem Register wird die Ausgabe von Daten ermöglicht. Daten werden ausgegeben bei den beiden Ausgabebefehlen STO und STOC. Je nach Ausgabebefehl sendet die Kontrolleinheit für die Dauer einer Taktperiode, beginnend mit

der negativen Taktflanke von X1, eine "1" auf die STO - oder auf die STOC-Leitung. Diese beiden Leitungen gehen auf ein ODER-Glied (Schaltzeichen: 1). Liegt einer der beiden Befehle an, so ist der Ausgang des ODER-Gliedes "1". Der Ausgang dieses ODER-Gliedes ist auf einen Eingang eines UND-Gliedes geführt. Ein zweiter Eingang ist über einen Inverter (Schaltzeichen: 1 mit am Ausgang nachgesetztem Kreis) mit dem Taktoszillator verbunden. Am dritten Eingang liegt der Ausgang Q des OEN-Registers. Nur wenn im OEN-Register eine "1" gespeichert ist, ein Ausgabebefehl anliegt, d.h., wenn am Ausgang des ODER-Gliedes eine "1" erscheint, und der Takt "0" ist, d.h., wenn am Ausgang des Inverters eine "1" anliegt, so ist auch am Ausgang des UND-Gliedes eine "1". Dieser Ausgang steuert den Eingabe- und den Ausgabebaustein. Deshalb ist er über Pin 2 mit der Schreib-/Leseleitung verbunden. Zwei Wahrheitstabellen sollen die Zusammenhänge noch einmal verdeutlichen:

Eingänge des ODER-Gliedes		Ausgang des ODER-Gliedes
STO-Leitung	STOC-Leitung	
0	0	0
0	1	1
1	0	1
-	-	-

Eingänge des UND-Gliedes			Ausgang des UND-Gliedes
OEN-Register	Takt	ODER-Glied	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Bei dem Befehl STO wird der Inhalt des Ergebnisregisters (Ausgang Q) und bei dem Befehl STOC wird der invertierte Inhalt des Ergebnisregisters (Ausgang Q) über die Datenleitung ausgegeben, wenn die Ausgabe freigegeben ist. Die Datenausgabe wird in Abb. 5 durch drei symbolische Schalter dargestellt. Das Schließen der Schalter wird durch die STO-Leitung, durch die STOC-Leitung und durch das OEN-Register bestimmt. Eine "1" schließt den Schalter. Die Schalter ermöglichen hinsichtlich der Datenleitung drei Ausgangszustände der Ausgangsschaltung: "1", "0" (beide niederohmig) und "Ausgang offen" (hochohmig). Solche Schaltungen werden TSL-Schaltungen (Drei-Zustands-Logik, engl. Tri State Logik - TSL) genannt. Eine Wahrheitstabelle soll den Ausgangszustand der Ausgangsschaltung bzw. den Inhalt der Datenleitung für die verschiedenen Zustände der drei Schalter aufzeigen:

STO-Leitung	STOC-Leitung	OEN-Register	Ausgangszustand bzw. Datenleitung
0	0	0	Ausgang offen
0	0	1	Ausgang offen
0	1	0	Ausgang offen
0	1	1	ER
1	0	0	Ausgang offen
1	0	1	ER
-	-	0	-
-	-	1	-

Liegt der Ausgangszustand "Ausgang offen" vor, so ist die Ausgangsschaltung von der Datenleitung entkoppelt. Das hat den Vorteil, daß dann die Dateneingabe, die auch über die Datenleitung stattfindet, nicht gestört werden kann. Was aber passiert, wenn der Ausgangszustand "0" oder "1" ist, wie es Zustand 4 und 6 der Wahrheitstabelle zeigen? Dabei müssen wir zwei Fälle unterscheiden: Ist der Takt "0", so liegt am Pin 2 der CPU und damit auch auf der Schreib-/Leseleitung eine "1". Dadurch ist der Eingabebaustein gesperrt und der Ausgabebaustein freigegeben. Ist der Takt "1", so arbeitet der Eingabebaustein auch nicht. Er wird dann über Pin 14 der CPU (Taktausgang) gesperrt. Auf diese Weise wird erreicht, daß die CPU nie gleichzeitig Daten aufnehmen und Daten ausgeben kann, was sonst zu Kurzschlüssen führen könnte.

8. Der RESET-Eingang

Durch das Betätigen des RESET-Tasters erscheint am Pin 1 der CPU ein 5 V-Signal. Dieses Signal geht auf ein Ausschalt-Verzögerungsglied. Eine Signaländerung am Eingang dieses Verzögerungsgliedes von 0 V auf 5 V bewirkt eine sofortige Signaländerung am Ausgang von 0 V auf 5 V. Eine Signaländerung am Eingang von 5 V auf 0 V bewirkt erst nach einer bestimmten Verzögerungszeit eine Signaländerung am Ausgang von 5 V auf 0 V. Die Verzögerungszeit beträgt hier ca. 0,1 μ s (millionstel Sec.). Der Ausgang des Verzögerungsgliedes ist unter anderem mit dem Taktoszillator verbunden. Ein 5 V-Impuls an Pin 1 setzt den Takt von X1 (Oszillatorausgang) auf "1" und hält diesen Zustand solange, bis der Ausgang des Verzögerungsgliedes wieder auf 0 V geht. Die eigentliche Aufgabe des RESET-Signals besteht aber darin, Informationen zu löschen. Zu diesem Zweck ist der Ausgang des Verzögerungsgliedes auch mit der Kontrolleinheit und mit den drei D-Flipflops (Ergebnisregister, IEN-Register und OEN-Register) verbunden. Über einen 5 V-Impuls an die Kontrolleinheit wird ein eventuell anliegendes Steuersignal (FLAG) wieder auf 0 V gesetzt. Genauso werden die drei D-Flipflops über einen speziellen Eingang R durch einen 5 V-Impuls zurückgesetzt. Die Inhalte der Register sind danach gelöscht, d.h., an ihren Q-Ausgängen erscheint eine "0".

2.2.2. DER EINGABEBAUSTEIN CD 4051

Der Eingabebaustein CD 4051 ist hier ein 1-aus-8-Multiplexer. Multiplexer haben die Aufgabe, aus verschiedenen angebotenen Daten die gewünschten Daten auszuwählen und über die Ausgänge weiterzuleiten. Der 4051 hat acht Eingänge (E0 ... E7) und einen Ausgang (Datenleitung). Vier dieser Eingänge (E1, E2, E3 und E4) sind mit jeweils einem Schalter verbunden. Durch Schließen eines Schalters wird der dazugehörige Eingang auf "1" gesetzt. Ist der Schalter offen, so ist der Eingang über einen Pull down-Widerstand (Widerstände, die mit 0 V verbunden sind, heißen Pull down-Widerstände) mit 0 V verbunden, d.h., der Eingang wird auf "0" gesetzt. Liegt an der betrachteten Leitung 5 V, so hat der Widerstand praktisch keinen Einfluß. Wird auf die Leitung kein Signal gegeben, so zieht der Widerstand sie auf 0 V. Dadurch wird erreicht, daß die Leitung nur zwei Zustände zuläßt: 0 V ("0") und 5 V ("1"). Auf demselben Prinzip beruhen die Pull up-Widerstände. Diese Widerstände sind mit einer Spannungsquelle (hier: 5 V) verbunden und ziehen die Leitung, wenn kein Signal anliegt, auf 5 V.

Die vier Schalter sind in einem Dual-In-Line-Gehäuse (DIL-Gehäuse) integriert. Abb. 6 zeigt die Pinbelegung des CD 4051.

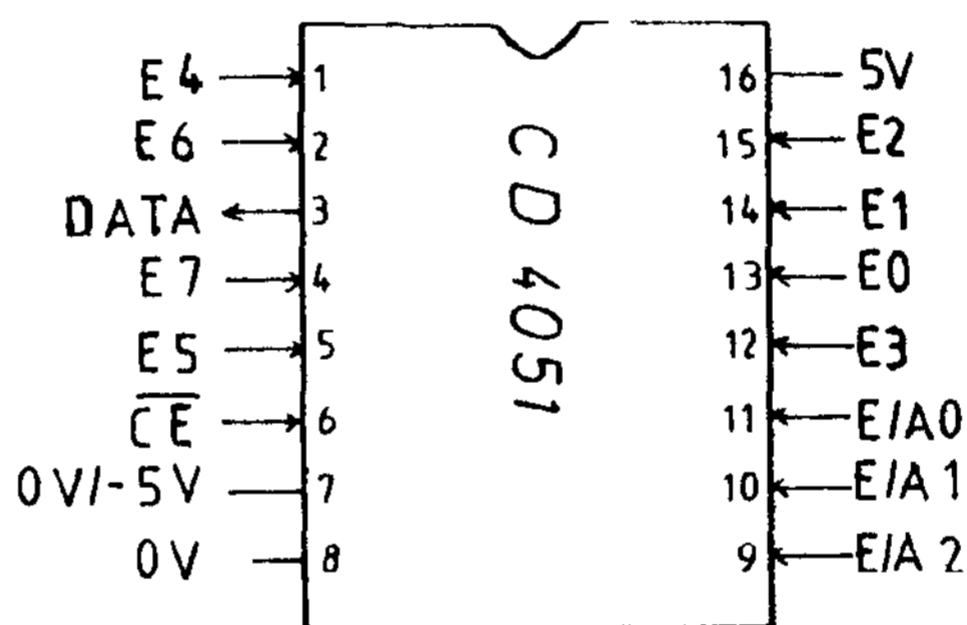


Abb. 6: Pinbelegung des CD 4051

Bedeutung der einzelnen Beine:

- Pin 1 : E4 Eingangsleitung Nr. 4
- Pin 2 : E6 Eingangsleitung Nr. 6
- Pin 3 : DATA Die Datenleitung ist der Ausgang des Bausteins. Die Information einer der acht Eingänge wird über diese Leitung an die CPU geschickt.
- Pin 4 : E7 Eingangsleitung Nr. 7
- Pin 5 : E5 Eingangsleitung Nr. 5
- Pin 6 : CE Chip Enable, Chip befähigen:
Ein 0 V-Signal an diesem Eingang bewirkt ein Arbeiten des Bausteins (Chip). Liegen hier 5 V an, so ist der 4051 nicht in Betrieb und seine Ausgabe ist gesperrt. Daten sollen nur ausgegeben werden, wenn die CPU Daten über die Datenleitung hereinholt und wenn kein Taktimpuls anliegt. In dem Augenblick ist der Takt "0" und die Schreib-/Leseleitung "0". Diesen Zustand

erkennt die Schaltung mit Hilfe eines ODER-Gliedes, Abb.7, an dessen beiden Eingängen die Schreib-/ Leseleitung und der Taktausgang von der CPU angeschlossen sind. Das ODER-Glied gibt nur dann durch eine "0" an Pin 6 die Schaltung frei, wenn beide Eingänge "0" sind. Eine Wahrheitstabelle soll die möglichen Zustände noch einmal aufzeigen:

Eingänge des ODER Gliedes		Ausgang des ODER- Gliedes
\bar{R}/W	Takt	\bar{CE}
0	0	0
0	1	1
1	0	1
1	1	1

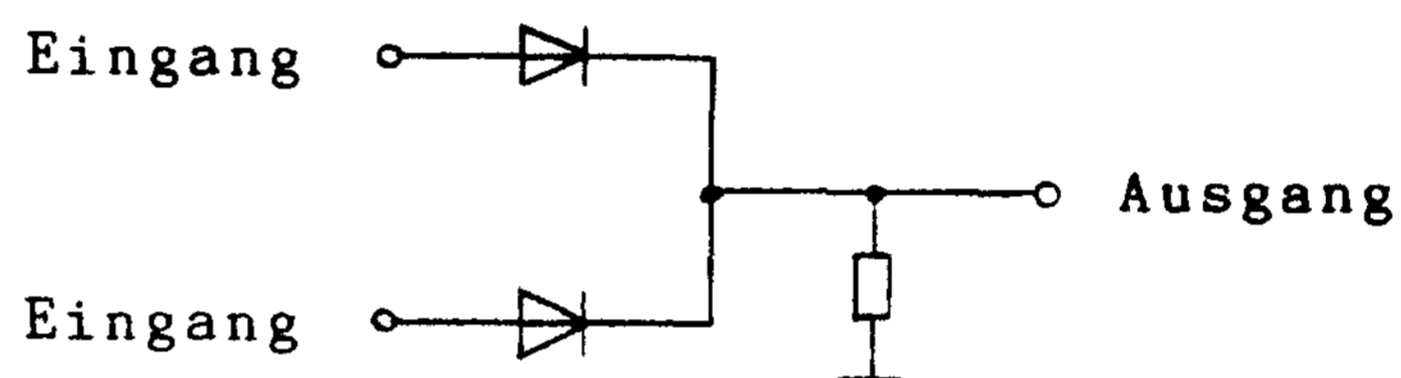


Abb. 7: Aufbau eines ODER-Gliedes aus einzelnen Bauelementen

- Pin 7 : 0V/ 5V Da sich der 4051 im Digitalbetrieb befindet, liegen hier 0 V an. Im Analogbetrieb würden hier -5 V anliegen.
- Pin 8 : 0V Masseanschluß
- Pin 9 : E/A 2,
- Pin 10: E/A 1,
- Pin 11: E/A 0 Die Entscheidung, welche der acht Eingangsleitungen die Information für die Datenleitung liefert, treffen die Ein-/ Ausgabeadressen. Mit diesen drei Leitungen kann man acht verschiedene Adressen anwählen. (Siehe: Kapitel 2.1.)
- Pin 12: E3 Eingangsleitung Nr. 3
- Pin 13: E0 Eingangsleitung Nr. 0
- Pin 14: E1 Eingangsleitung Nr. 1
- Pin 15: E2 Eingangsleitung Nr. 2
- Pin 16: 5V Betriebsspannung

2.2.3. DER AUSGABEBAUSTEIN MC 14599

Der Ausgabebaustein MC 14599 ist hier ein 1-auf-8-Demultiplexer mit Speicherwirkung. Demultiplexer haben die Aufgabe, das am Eingang erscheinende Signal zu einem bestimmten Ausgang durchzuschalten. In diesem Fall wird die Information der Datenleitung, je nach Ein-/Ausgabeadresse, auf eine der acht Ausgangsleitungen geschickt und solange gehalten, bis die Information entweder durch Drücken des Reset-Tasters gelöscht wird oder bis eine andere Information auf diese Leitung kommt. Abb. 8 zeigt die Pinbelegung des MC 14599.

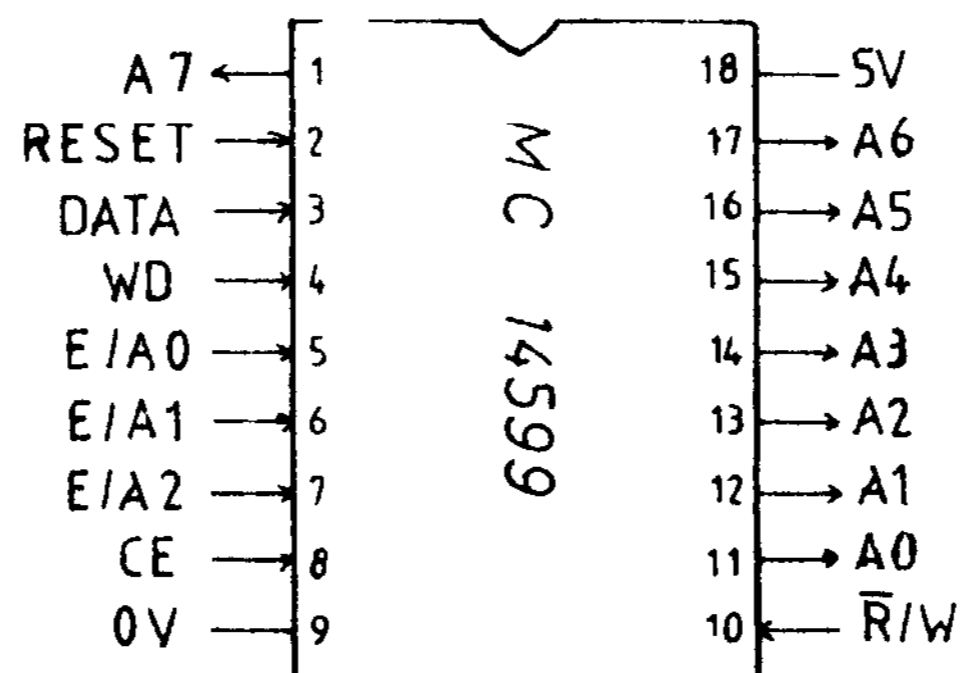


Abb. 8: Pinbelegung des MC 14599

Bedeutung der einzelnen Beine:

- Pin 1: A7 Ausgangsleitung Nr. 7
- Pin 2: RESET Normalerweise ist dieses Bein über einen Widerstand mit 0 V verbunden. Verbindet man es aber mit Hilfe eines Tasters (Reset-Taster) mit 5 V, dann werden alle Informationen an den Ausgangsleitungen gelöscht.
- Pin 3: DATA Die Datenleitung ist der Eingang des Bausteins. Die Information für einen der acht Ausgänge kommt über diese Leitung von der CPU herein.
- Pin 4: WD Write Disabled, Schreiben unfähig machen: Dieses Bein ist mit dem Takt verbunden, da der 14599 immer dann Daten ausgeben soll, wenn er über Pin 8 angesprochen wird.
- Pin 5: E/A 0,
- Pin 6: E/A 1,
- Pin 7: E/A 2 Die Entscheidung, welche der acht Ausgangsleitungen die Information der Datenleitung erhält, treffen auch hier, wie beim Eingabebaustein, die Ein-/Ausgabeadressen.
- Pin 8: CE Chip Enable, Chip befähigen: Ein 5 V-Signal an diesem Eingang bewirkt eine Freigabe des Bausteins. Da der 14599 nur arbeiten soll, wenn die CPU Daten über die Datenleitung ausgibt, ist Pin 8 mit der Schreib-/ Leseleitung verbunden. Bei der Datenausgabe ist diese Leitung nämlich auf 5 V.
- Pin 9: 0V Masseanschluß

Pin 10: \bar{R}/W Da der 14599 hier nur als Ausgabebaustein verwendet werden soll, kann dieser Anschluß direkt mit Pin 8 verbunden werden. Bei 0 V an Pin 10 würde der 14599 einen Eingabebaustein darstellen.

Pin 11: A0 Ausgangsleitung Nr. 0
 Pin 12: A1 Ausgangsleitung Nr. 1
 Pin 13: A2 Ausgangsleitung Nr. 2
 Pin 14: A3 Ausgangsleitung Nr. 3
 Pin 15: A4 Ausgangsleitung Nr. 4
 Pin 16: A5 Ausgangsleitung Nr. 5
 Pin 17: A6 Ausgangsleitung Nr. 6
 Pin 18: 5V Betriebsspannung

2.2.4. DIE EINGABETASTATUR

Die Eingabetastatur enthält vierzehn Tasten für die Befehle und acht Tasten für die Ein-/ Ausgabeadressen. Der Befehlscode und die Adressen werden durch eine Diodenmatrix codiert. Jeder Befehl und jede Adresse besteht aus Zahlen im Binärsystem, deren Stelle durch 0 V ("0") bzw. 5 V ("1") dargestellt werden. Was aber ist eine "Codierung durch eine Diodenmatrix"? Abb. 9 zeigt dafür ein Beispiel. Der Befehl IEN ist durch 1010 und der Befehl OEN durch 1011 codiert. Zur Codierung ist es nur notwendig, die Stellen, die "0" werden sollen, mit 0 V zu verbinden, denn durch Pull up-Widerstände sind alle Leitungen schon mit 5 V verbunden. Wird der Taster für den IEN-Befehl gedrückt, so liegt an Pin 4, 5, 6 und 7 der CPU die Bitfolge 1 10 an und die Kontrolleinheit der CPU erkennt, daß der Befehl IEN gemeint ist. Genauso wie der IEN-Befehl und der OEN-Befehl sind auch die anderen zwölf Befehle mit Hilfe der Diodenmatrix codiert. Für jede "0" im Befehlssatz wird also eine Diode benötigt. Somit sind 31 Dioden nötig.

An dem Beispiel in Abb. 9 wird auch deutlich, warum Dioden erforderlich sind. Wird nicht der IEN- sondern der OEN-Taster gedrückt, so wären ohne Dioden die Leitungen Nr. 6 und Nr. 4 miteinander gekoppelt und die falsche Kombination 1010 würde anliegen.

Die Dioden dienen also zur Entkopplung der einzelnen Tasten und Leitungen.

Die Ein-/ Ausgabeadressen sind ebenfalls durch eine Diodenmatrix verdrahtet. Sie sind im Dualsystem codiert (Siehe: Kapitel 2.1.).

Man erkennt, daß man für die Ein-/ Ausgabeadresse Nr. 7 keinen Taster benötigt, da sie durch 111 codiert ist und man keine Leitung auf 0 V herunterziehen braucht. Der Taster ist trotzdem vorhanden, wenn er auch nicht beschaltet ist, um einen unerfahrenen Programmierer nicht zu verwirren.

Für die Ein-/ Ausgabeadressen sind die Leitungen Nr. 0, 1 und 2 vorgesehen. Es sind 12 Dioden für die Codierung der Adressen nötig. Für eine Erweiterung der Adressen kann die Leitung Nr. 3 verwendet werden (Siehe: Kapitel 2.1.).

Zu jedem Befehl gehört eine Ein-/ Ausgabeadresse. Sie ergeben zusammen einen Programmschritt. Der Befehl gelangt zur CPU und die dazugehörige Ein-/ Ausgabeadresse gelangt zum Eingabe- und

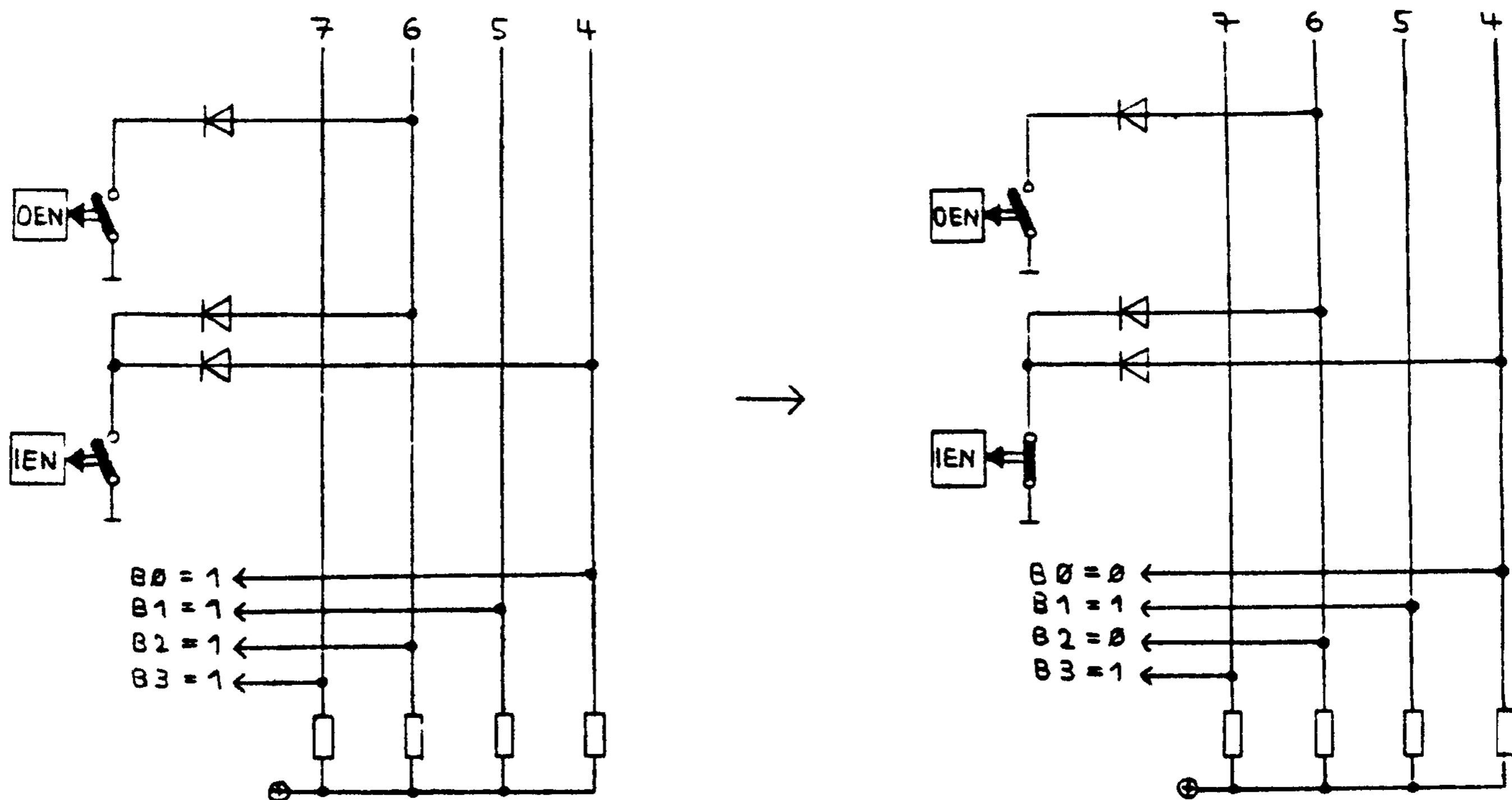


Abb. 9: Beispiel für die Codierung zweier Befehle

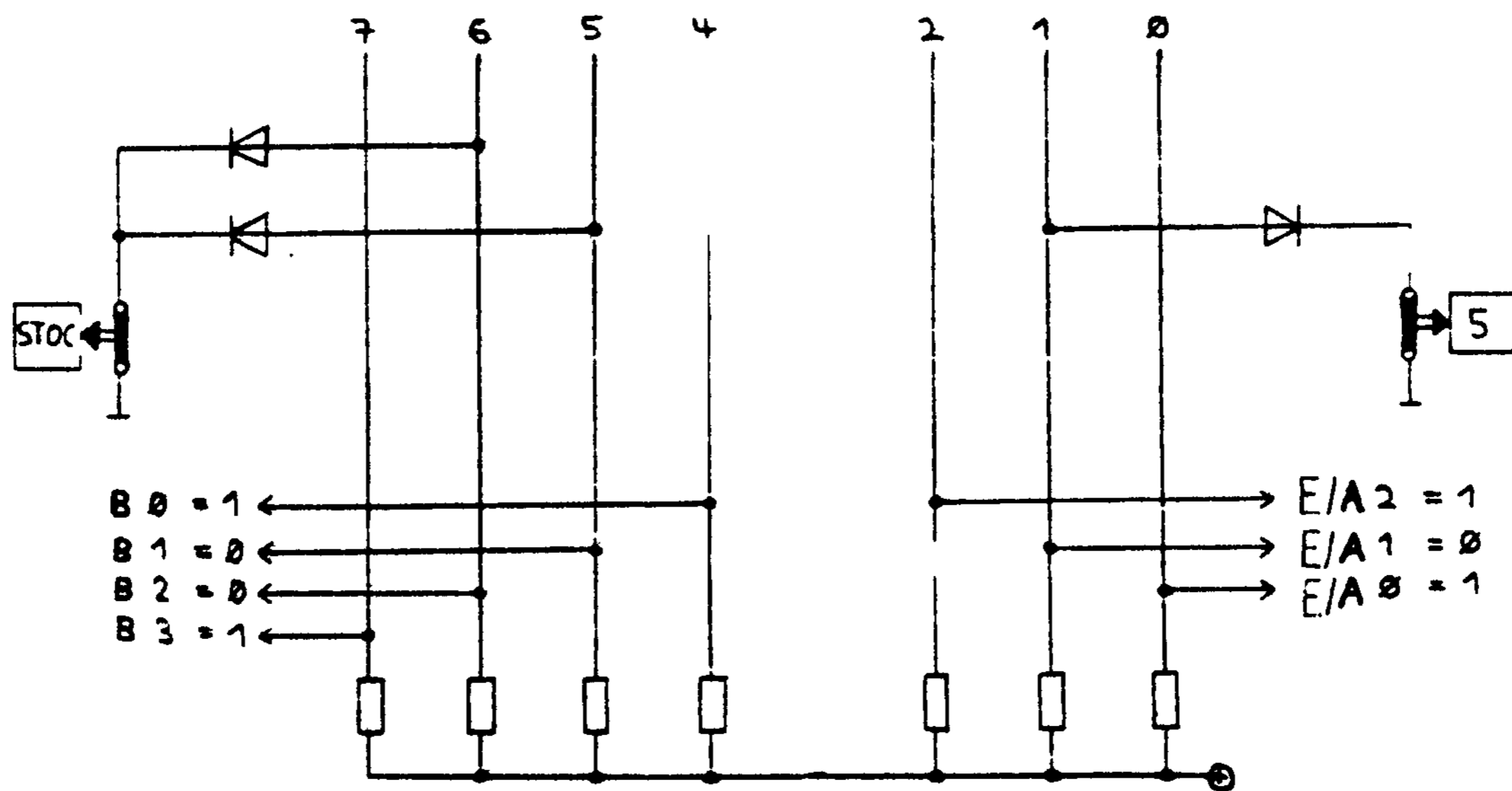


Abb. 10: Beispiel für die Codierung eines Programmschrittes

zum Ausgabebaustein. Abb. 10 zeigt die Codierung eines Befehls mit seiner Adresse. Als Beispiel wird der Programmschritt STOC 5 verwendet. Der Befehl STOC wird durch die Bitfolge 1001 codiert und die Ein-/ Ausgabeadresse Nr.5 ist im Dualsystem die Kombination 101.

Die Eingabetastatur kann auch durch sieben Schalter ersetzt werden. Man muß dann per Hand den Code für jeden Programmschritt einstellen. Durch Schließen eines Schalters wird die dazugehörige Leitung auf "0" gesetzt. Ist der Schalter offen, so ist die Leitung über einen Pull up-Widerstand mit 5V verbunden, d.h., die Leitung wird auf "1" gesetzt.

2.2.5. DER PROGRAMMSPEICHER 2112

Der Programmspeicher 2112 ist ein 256 * 4-Bit-Speicher mit wahlfreiem Zugriff (engl. Random Access Memory-RAM). Das RAM stellt einen Schreib-/ Lese-Speicher dar, d.h., der Speicherinhalt kann beliebig oft verändert werden. Neben dem RAM gibt es noch unter anderem einen Nur-Lese-Speicher (engl. Read Only Memory-ROM). Das ROM ist ein Festwertspeicher, d.h., der gespeicherte Inhalt kann nicht verändert werden. Es ist einmalig - bei seiner Herstellung - programmiert.

Die Speicherzellen des 2112 sind D-Flipflops (Siehe: Kapitel 2.2.1). Der Speicherinhalt geht leider beim Abschalten der Versorgungsspannung verloren.

Für die Speicherung eines Befehls und seiner Ein-/ Ausgabeadresse werden sieben Speicherzellen benötigt. Daher werden zwei Speicher 2112 (a) und 2112 (b) verwendet. Abb. 11 zeigt die Pinbelegung der beiden Bausteine.

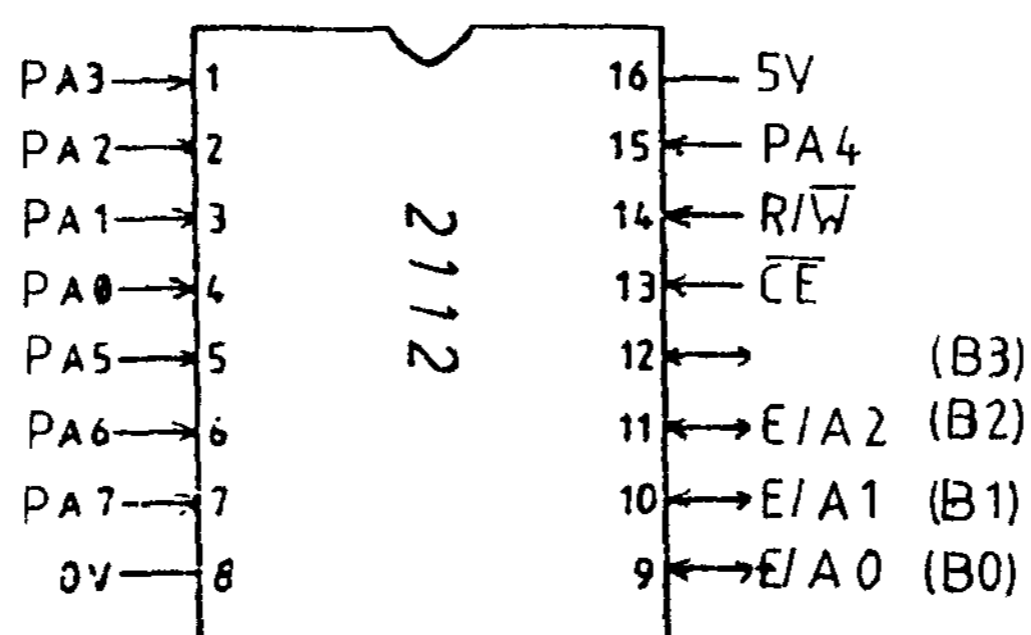


Abb.11: Pinbelegung des 2112 a (2112 b)
 "A" steht für Ausgang
 "PA" steht für Programmadresse

Bedeutung der einzelnen Beine:

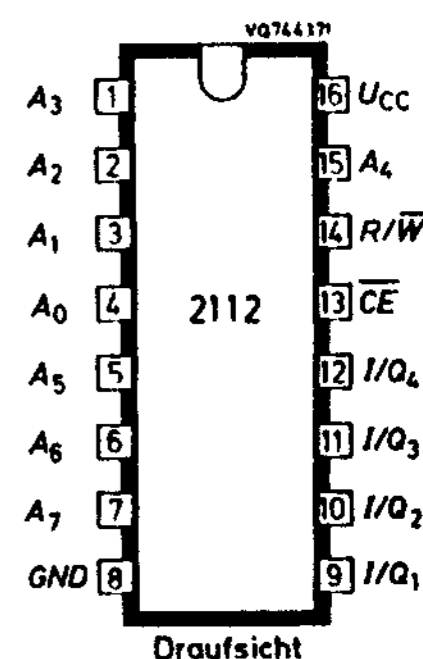
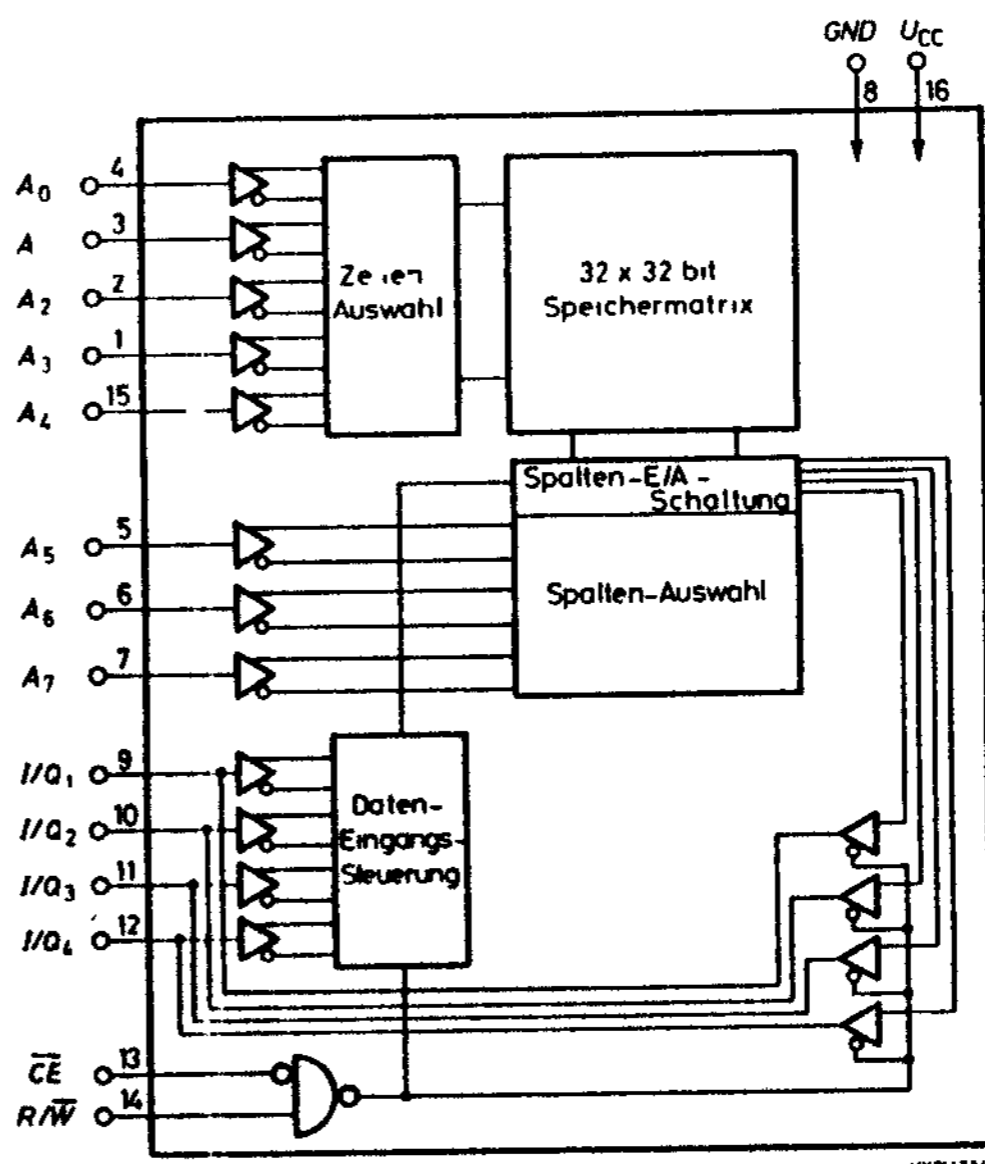
- Pin 1: PA3,
 Pin 2: PA2,
 Pin 3: PA1,
 Pin 4: PA0,
 Pin 5: PA5,
 Pin 6: PA6,
 Pin 7: PA7,
 Pin 15: PA4 Programmadressen
 Pin 8: 0V Masseanschluß
 Pin 9: E/A0 (B0),
 Pin 10: E/A1 (B1),
 Pin 11: E/A2 (B2),
 Pin 12: - (B3) Beim 2112 (a) liegen an Pin 9 ... 11 die Ein-/Ausgabeadressen an. Pin 12 kann für eine Erweiterung vorgesehen werden. Beim 2112 (b) liegen an Pin 9 ... 12 die Befehle an. Abhängig von Pin 14 wirken diese vier Anschlüsse entweder als Eingänge, d.h., die anliegenden Bits werden eingespeichert, oder als Ausgänge, d.h., die gespeicherten Werte werden ausgelesen, wobei sich der Inhalt der Speicherzellen nicht ändert. Damit bei der Ausgabe von einer "1" kein Kurzschluß entsteht, wenn von der Eingabetastatur durch Drücken eines Tasters eine "0" über dieselbe Leitung kommt, sind die acht Leitungen zur Eingabetastatur mit acht Schutzwiderständen versehen. Über diese Widerstände können dann die 5V Spannungsdifferenz zwischen der "0" und der "1" abfallen.
 Pin 13: \overline{CE} Chip Enable, Chip befähigen:
 Ein 0V-Signal an diesem Eingang bewirkt ein Arbeiten des Bausteins. Da der 2112 immer betriebsbereit sein soll, ist dieses Bein mit Masse verbunden.
 Pin 14: R/\overline{W} Normalerweise ist dieses Bein über einen Widerstand mit 5V verbunden. In diesem Zustand können Speicherinhalte über Pin 9 ... 12 ausgelesen werden. Verbindet man Pin 14 mit Hilfe eines Tasters (Schreib-Taster) mit 0V, dann sind Pin 9 ... 12 Eingänge, d.h., die Bits, die dort anliegen, werden eingespeichert. Alte Speicherinhalte werden dabei überschrieben.
 Pin 16: 5V Betriebsspannung

2112 2112-1 2112-2

256 x 4 bit-RAM, statisch

Technologie: N-Kanal-MOS
 Zugriffszeit: 2112: max. 1000 ns
 2112-1: max. 500 ns
 2112-2: max. 650 ns

Speisespannung: + 5 V
 Speisestrom: max. 70 mA
 2 Freigabeeingänge
 „three state“-Ausgänge
 TTL-kompatibel
 Gehäuse: F, N



2.2.6. DER PROGRAMMZÄHLER CD 4029

Die beiden Programmzähler CD 4029 (a) und CD 4029 (b) stellen hier je einen 4-Bit-Binärzähler dar, die so miteinander verbunden sind, daß sie zusammen einen 8-Bit-Binärzähler ergeben. Dieser soll die Ansteuerung der acht Eingänge für die Programmadressen der Speicher (Pin 1 ... 7 und Pin 15 des 2112) übernehmen.

Der 8-Bit-Zähler beginnt bei der Adresse 0, zählt hoch bis zur Adresse 255 und fängt wieder bei 0 an zu zählen.

Es muß beachtet werden, daß der Zähler keine besondere Rücksetzmöglichkeit hat. Das Rücksetzen auf die Programmadresse 0 geschieht mit Hilfe des Preset-Schalters.

Abbildung 12 zeigt die Pinbelegung der Bausteine.

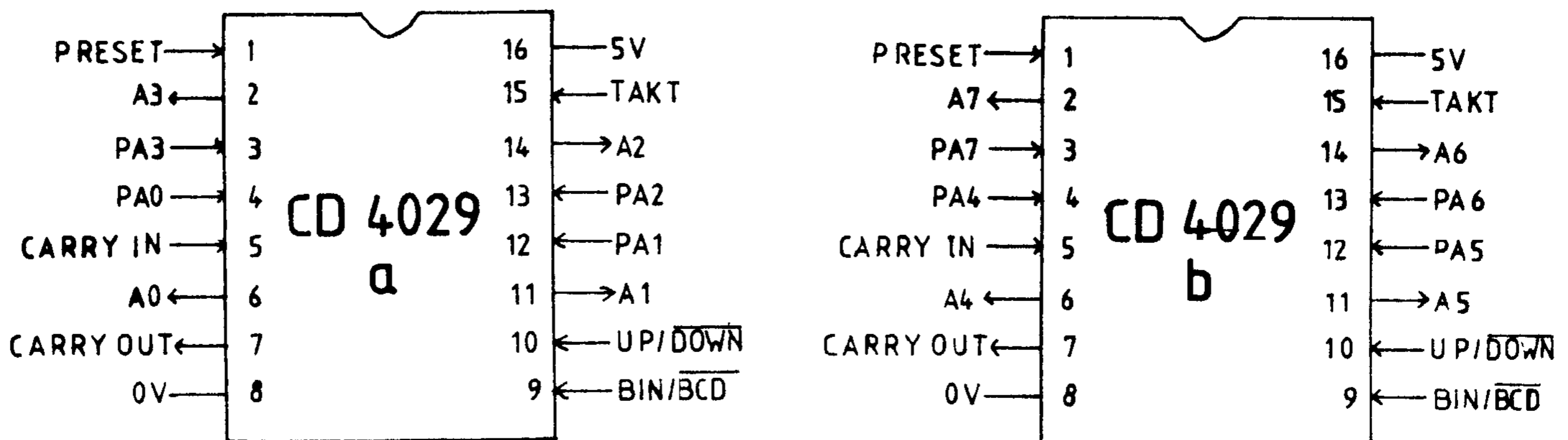


Abb. 12: Pinbelegung des CD 4029 (a) und CD 4029 (b)

Bedeutung der einzelnen Beine:

Pin 1: PRESET Normalerweise ist dieses Bein über einen Widerstand mit 0 V verbunden. Verbindet man es aber mit 5 V, so wird die Programmadresse zurückgesetzt, d.h., der Zähler zählt bei 0 weiter. Man kann Pin 1 mit Hilfe des Preset-Tasters oder durch den Befehl JUMP mit 5 V verbinden. Durch den Befehl JUMP wird in der CPU ein 5V-Impuls mit der Dauer einer Taktperiode erzeugt (siehe: Kapitel 2.2.1.). Dieses Steuersignal erscheint an Pin 9 der CPU. Von dort erhalten der 4029 (a) und der 4029 (b) über eine Leitung den 5V-Impuls.

Pin 2: PA3 (PA7)

Pin 6: PA0 (PA4)

Pin 11: PA1 (PA5)

Pin 14: PA2 (PA6) Beim 4029 (a) liegt an Pin 2, 6, 11 und 14 die untere Hälfte der Programmadresse. Beim 4029 (b) liegt hier die obere Hälfte der Programmadresse.

Pin 3: 0V

Pin 4: 0V

Pin 12: 0V

Pin 13: 0V

- Pin 5: CARRY IN, Übertrag-Eingang
- Pin 7: CARRY OUT, Übertrag-Ausgang:
 Pin 5 des 4029 (a) ist mit 0V verbunden. Pin 7 des 4029 (a) ist mit Pin 5 des 4029 (b) verbunden. Pin 7 des 4029 (b) ist offen gelassen. Mit Hilfe dieser Anschlüsse gelingt es, aus den beiden 4-Bit-Zählern einen 8-Bit-Zähler zu bauen. Liegt Pin 5 an 0V, so zählt der Baustein bei jedem Taktimpuls. Dies ist beim 4029 der Fall. Während er von 0 (PA0=PA1=PA2=PA3=0) bis 15 (PA0=PA1=PA2=PA3=1) zählt, liegen an Pin 7 des 4029 (a) und damit an Pin 5 des 4029 (b) 5V an. Der 4029 (b) zählt also während dieser Zeit nicht (PA4=PA5=PA6=PA7=0). Ist der 4029 (a) bei 15 angekommen, so geht Pin 7 des 4029 (a) und damit Pin 5 des 4029 (b) auf 0V. Beim nächsten Taktimpuls zählen also beide Bausteine. Aus Programmadresse 15 (PA0=PA1=PA2=PA3=1, PA4=PA5=PA6=PA7=0) wird Programmadresse 16 ((PA0=PA1=PA2=PA3=0, PA4=1, PA5=PA6=PA7=0). Danach liegen an Pin 7 des 4029 (a) und an Pin 5 des 4029 (b) wieder 5V an und der 4029 (b) zählt nicht mehr. Durch Steuern der 4-Bit-Zähler mit einem gemeinsamen Takt ergibt sich ein voll synchroner 8-Bit-Zähler.
- Pin 9: Bin/BCD Binär-Zähler/BCD-Zähler:
 Da der 4029 hier ein Binär-Zähler ist, liegt dieses Bein an 5V. Binär-Zähler zählen von 0 bis 15 und fangen dann wieder bei 0 an. Würde Pin 9 an 0V liegen, so wäre der 4029 ein BCD-Zähler (Binärcode für Dezimal-Ziffern - engl. Binary Coded Decimal - BCD). BCD-Zähler zählen von 0 bis 9 und fangen wieder bei 0 an. (Ist er bei 9 angekommen, so geht Pin 7 auf 0V bis nach dem nächsten Taktimpuls.)
- Pin 10: UP/DOWN Vorwärts-Zähler/Rückwärts-Zähler:
 Da der 4029 hier ein Vorwärts-Zähler ist, liegt dieses Bein an 5V. Vorwärts-Zähler zählen von 0 bis 15 (bzw. bis 9) und fangen wieder bei 0 an. Rückwärts-Zähler zählen von 15 (bzw. von 9) bis 0 herunter und fangen dann wieder oben an.
- Pin 15: TAKT Dieses Bein ist mit Pin 14 der CPU verbunden. Der 4029 schreitet um eine Zählung fort bei jeder positiven Taktflanke von X1 (Pin 14 der CPU), wenn die Zählung freigegeben ist.
- Pin 16: Betriebsspannung

2.2.7. DIE ANZEIGEEINHEIT

Die Anzeigeeinheit besteht aus vier Bausteinen ULN 2003 und 25 Leuchtdioden. Eine Leuchtdiode (Schaltzeichen: Diode mit zwei Pfeilen) ist eine Licht aussendende Diode (engl. Light Emitting Diode-LED). 15 Leuchtdioden zeigen den Informationsgehalt der Ein- und Ausgänge, vier den Befehlscode, drei die Nummer der Ein- oder Ausgabeleitung im Dualsystem, eine die Information auf der Datenleitung, eine den Inhalt des Ergebnisregisters und eine den Takt.

Ein ULN 2003 ist im Prinzip ein 7-fach-Inverter. Er hat sieben Eingänge und sieben Ausgänge. Jeder Eingang ist über einen Inverter mit einem Ausgang verbunden. Die Inverter haben einen offenen Kollektor (engl. open collector). Ein solcher Schaltkreis bewirkt folgendes: Liegt am Eingang des Inverters eine "1", so liegt am Ausgang eine "0". Liegt aber am Eingang eine "0", so ist der Ausgangszustand normalerweise undefiniert. Eine Schaltung, die an diesen Ausgang angeschlossen werden kann, bestimmt dann den Ausgangszustand des Inverters.

Abb. 13 zeigt das Prinzip der Ansteuerung einer Leuchtdiode.



Abb. 13: Prinzip der Ansteuerung einer Leuchtdiode

Hier ist der Ausgang über einen Widerstand und eine Leuchtdiode mit 5V verbunden, so daß er auf 5V gezogen wird. Der Widerstand dient dazu, den Strom der Leuchtdiode zu begrenzen. Damit auch der Ausgangsstrom des Bausteins, z.B. des Ausgabebausteins, der mit dem Inverter verbunden ist, nicht zu groß wird, ist vor dem Inverter noch ein Widerstand geschaltet.

Mit Hilfe dieser Schaltung wird folgendes erreicht: Liegt am Eingang des Inverters eine "1", so liegt am Ausgang eine "0", d.h., die Leuchtdiode leuchtet. Leuchten bedeutet "1". Liegt am Eingang des Inverters eine "0", so liegt am Ausgang eine "1", d.h., die Leuchtdiode leuchtet nicht, was eine "0" bedeutet. Der Zustand der Leuchtdiode zeigt uns also die Information der Leitung an, die vor den Inverter geschaltet ist.

Bei 25 Schaltungen, die alle nach dem in Abb. 13 gezeigten Prinzip aufgebaut sind, können 25 Informationen angezeigt werden. Da jeder ULN 2003 sieben Inverter enthält, benötigt man vier Bausteine. Die Pinbelegung der vier Bausteine kann in Abb. 25 nachgeschaut werden.

Abb. 14 zeigt den prinzipiellen Aufbau eines Bausteins.

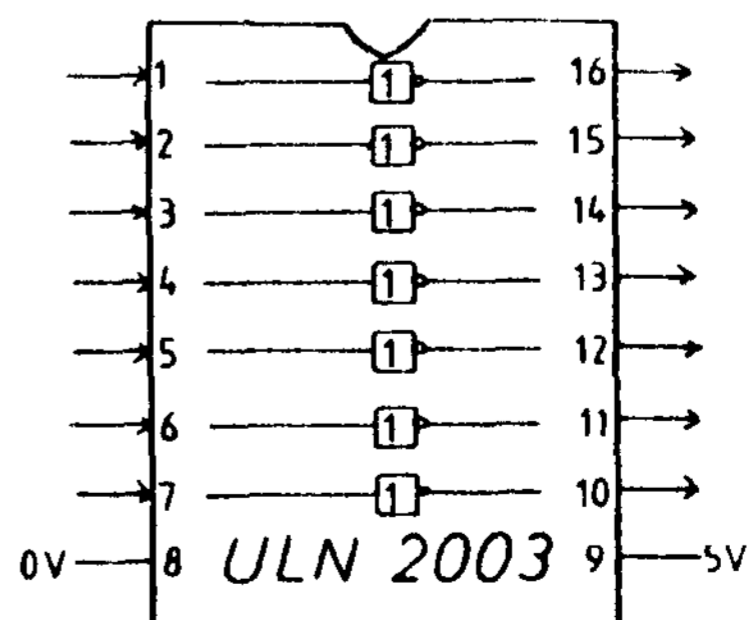


Abb. 14: Prinzipieller Aufbau des ULN 2003

2.2.8. DIE TAKTEINHEIT

Drei Baugruppen sind für die Erzeugung des Taktes zuständig: Eine bistabile Kippstufe, ein Timer und der Taktoszillator der CPU.

Welche Baugruppe gerade ausschlaggebend für den Takt ist, bestimmt die Stellung des Taktwahlschalters.

1. Stellung "Hand-Takt"

In der Stellung "Hand-Takt" wird das Taktsignal mit einer bistabilen Kippstufe erzeugt. Abb. 15 zeigt den Aufbau einer bistabilen Kippstufe mit Taster.

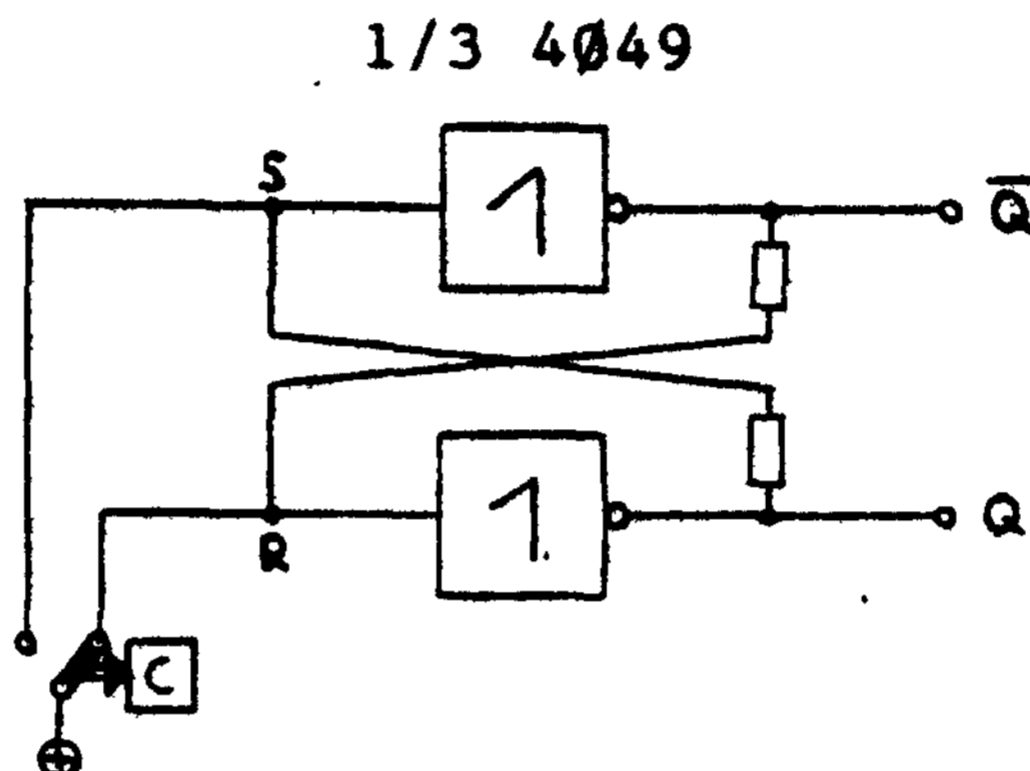


Abb. 15: Bistabile Kippstufe mit Taster

Normalerweise ist der Eingang R (Reset) der bistabilen Kippstufe über den Takt-Taster mit 5 V verbunden. Dadurch ist der Ausgang Q auf "0". Der Eingang S (Set), der mit dem Ausgang \bar{Q} verbunden ist, liegt auch auf "0" und damit der Ausgang Q auf "1".

Drückt man den Takt-Taster, so ist der Eingang S mit 5 V verbunden. Dadurch ist der Ausgang \bar{Q} auf "0", der Eingang R, der mit dem Ausgang \bar{Q} verbunden ist, auf "0" und der Ausgang Q auf "1". Nach dem Loslassen des Takt-Tasters ist der ursprüngliche Zustand wieder hergestellt. Während des Drückens beziehungsweise Loslassens gibt es eine kurze Übergangszeit, in der keiner der beiden Eingänge R oder S mit 5 V verbunden ist. Während dieser Zeit bestimmen die Ausgänge Q und \bar{Q} die Werte der Eingänge R und S über die beiden Rückkopplungen von \bar{Q} nach R und Q nach S. Der letzte Zustand der bistabilen Kippstufe bleibt also erhalten. Dadurch wird verhindert, daß Störungen wie Prellen, Kontakt-schwierigkeiten beim Betätigen des Takt-Taster auftreten.

In den beiden Rückkopplungszweigen befinden sich zwei Schutzwiderstände. Beim Drücken des Tasters springt z.B. das Signal am Eingang S von "0" auf "1". Der Ausgang \bar{Q} und damit der Eingang R reagieren nicht sofort auf diese Änderung, da das Durchlaufen des Signals durch den Inverter eine gewisse Zeit beansprucht (Laufzeit). Genauso lange braucht das Signal am Eingang R, um über den anderen Inverter auf den Ausgang Q zu gelangen. Erst nach einer Verzögerungszeit, die etwa gleich der doppelten Laufzeit ist, springt also das Signal am Ausgang Q von "0" auf "1". Während dieser Verzögerungszeit liegt am Eingang S

ein "1" und am Ausgang Q eine "0". Da aber Ausgang Q mit Eingang S verbunden ist, muß ein Widerstand dazwischen geschaltet werden, damit kein Kurzschluß entsteht. Die gleiche Problematik zeigt sich auch beim Loslassen des Tasters. Es muß also auch zwischen Eingang R und Ausgang \bar{Q} ein Schutzwiderstand geschaltet werden.

Der Ausgang Q der bistabilen Kippstufe kann durch den Taktwahlschalter mit Pin 13 der CPU verbunden werden.

2. Stellung "Langsam-Takt"

In der Stellung "Langsam-Takt" wird das Taktsignal eines Timers wirksam. Hier wird der Timer NE 555 eingesetzt, der mit den Widerständen R_1 und R_2 und dem Kondensator C beschaltet ist:

Abb. 16.

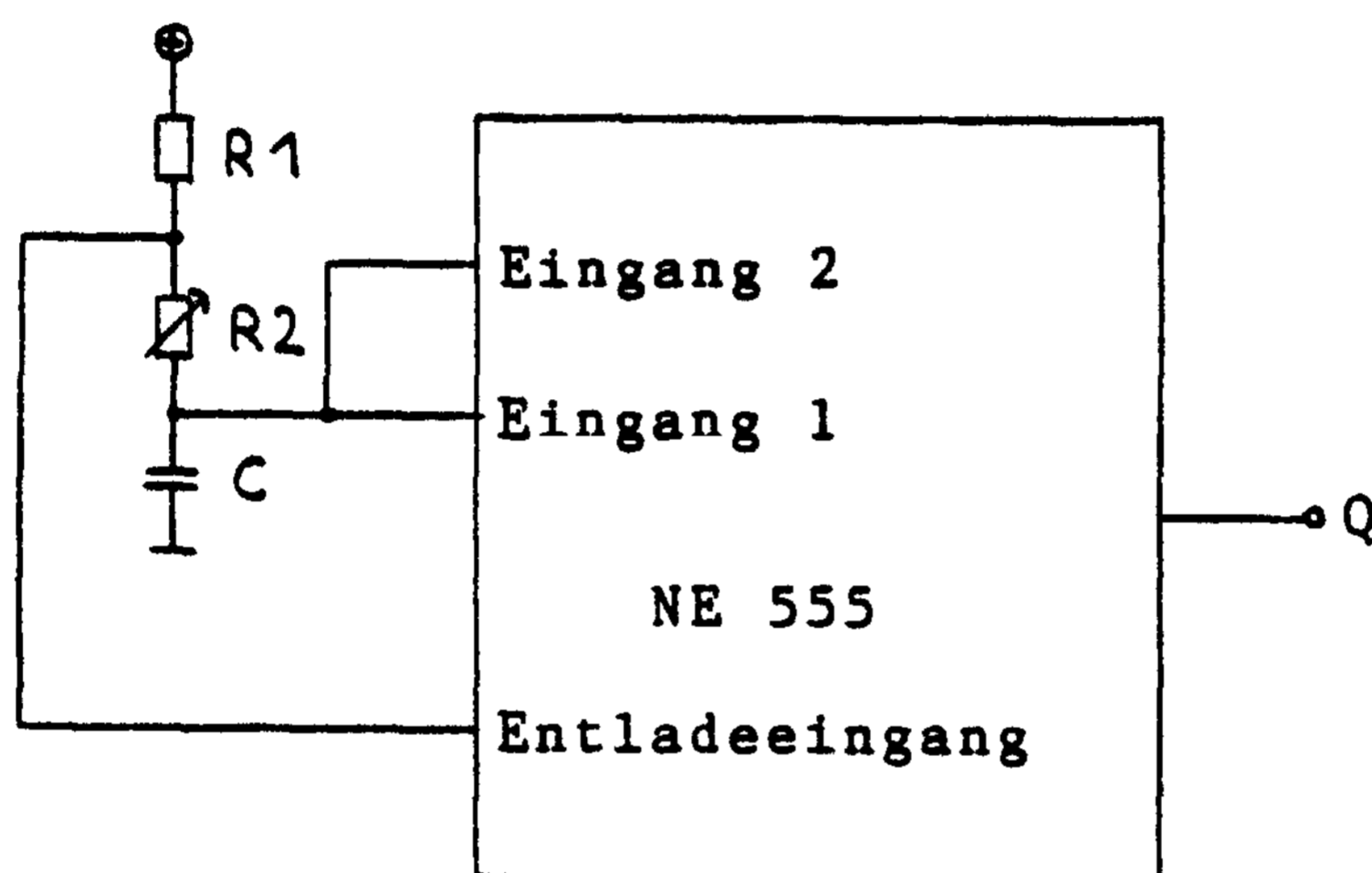


Abb. 16: Timer NE 555 mit prinzipieller Beschaltung

Dieser Kondensator wird von der Betriebsspannung über die beiden Widerstände aufgeladen, d.h., die Spannung am Kondensator steigt. Diese Spannungen liegen an zwei Eingängen des Timers. Im Timer werden sie mit zwei fest vorgegebenen Spannungen verglichen: Spannung am Eingang 1 mit $1/3$ der Betriebsspannung, Spannung am Eingang 2 mit $2/3$ der Betriebsspannung. Es sind also drei Fälle zu unterscheiden: 1. Die Kondensatorspannung ist kleiner $1/3$ der Betriebsspannung, am Ausgang des Timers liegt eine "1". 2. Die Kondensatorspannung liegt zwischen den beiden Vergleichsspannungen des Timers, der Ausgangszustand des Timer ändert sich nicht. 3. Ist die Kondensatorspannung größer als $2/3$ der Betriebsspannung, geht der Ausgang Q des Timers auf "0". Dadurch wird der Kondensator über den Widerstand R_2 durch den Entladeeingang entladen, d.h., die Kondensatorspannung sinkt. Wird die untere Vergleichsspannung erreicht, geht der Ausgang des Timers wieder auf "1". Der Entladevorgang stoppt und der Aufladevorgang setzt wieder ein. Auf diese Weise ändert sich der Ausgangszustand periodisch. Dabei werden die Dauer der beiden Ausgangszustände durch die Größen der äußeren Beschaltung bestimmt. Große Werte bewirken z.B. eine langsame Änderung. Die

Geschwindigkeit kann am besten mit Hilfe eines variablen Widerstandes (Potentiometer) eingestellt werden. Da der Widerstand R_2 am Auflade- und am Entladevorgang beteiligt ist, eignet sich dieser besonders gut dafür.

Der Zusammenhang zwischen der Ausgangsfrequenz f (Anzahl der Taktperioden pro Sekunde) und der äußeren Beschaltung (R_1 , R_2 , C), wie sie in Abb. 16 dargestellt ist, lautet als Formel

$$f = \frac{1,44}{(R_1 + 2 \cdot R_2) \cdot C}$$

Anhand der Pinbelegung des Timers (Abb. 17) wird noch kurz die Bedeutung der einzelnen Beine erläutert. Näheres zum Timer ist im Anhang zu finden.

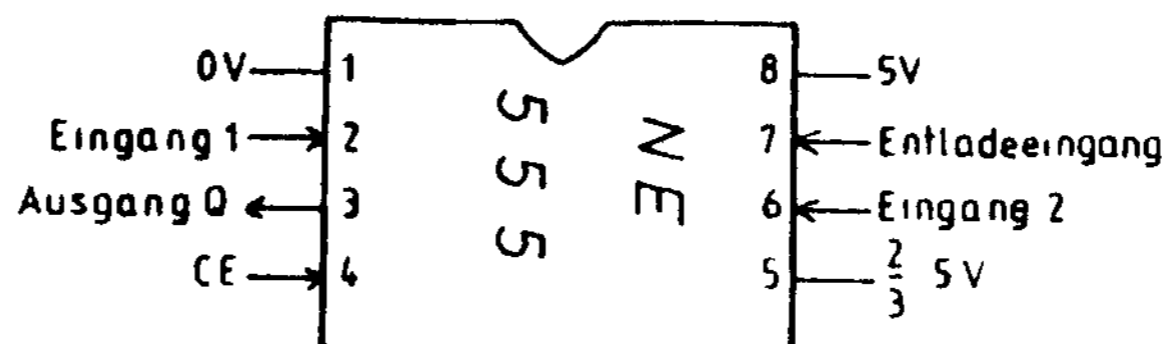


Abb. 17: Pinbelegung des Timers NE 555

Pin 1:	0 V	Masseanschluß
Pin 2:	Eingang 1	Die hier anliegenden Spannung wird mit 1/3 der Betriebsspannung verglichen.
Pin 3:	Q	Dies ist der Ausgang des Timers. Er ist über einen hochohmigen Widerstand mit Pin 13 der CPU verbunden.
Pin 4:	CE	Chip Enable, Chip befähigen: Ein 5 V Signal an diesem Eingang bewirkt ein Arbeiten des Bausteins. Damit der NE 555 betriebsbereit ist, wird dieses Signal mit 5 V verbunden.
Pin 5:	2/3 von 5 V	2/3 der Betriebsspannung im unbeschalteten Zustand dieses Bauelements.
Pin 6:	Eingang 2	Die hier anliegende Spannung wird mit 2/3 der Betriebsspannung verglichen.
Pin 7:	Entladeeingang	Beim Entladevorgang wird der Kondensator über einen Widerstand (hier R_2) und über dieses Beinchen entladen. Während dieser Zeit ist dieser Anschluß also intern mit Masse verbunden.
Pin 8:	5 V	Betriebsspannung

3. Stellung "Schnell-Takt"

In der Stellung "Schnell-Takt" kommt der Takt allein vom Taktoszillator der CPU. Mit dem Taktwahlschalter wird der Ausgang X1 des Taktoszillators (Pin 14 der CPU) über einen Widerstand mit dem Eingang X2 des Taktoszillators (Pin 13 der CPU) verbunden, d.h. rückgekoppelt. Die Werte dieses Rückkopplungswiderstandes bestimmen die Taktfrequenz: Abb. 18.

Die Unterteilung der beiden Achsen sind logarithmische, d.h., die Striche zwischen z. B. 10k und 100k bedeuten entsprechend 15k; 20k; 30k; 40k; ...90k. Die Striche zwischen z.B. 100k und 1M bedeuten entsprechend 150k; 200k; 300k; ...; 900k. Auf diese Art und Weise ist das ganze Schaubild unterteilt.

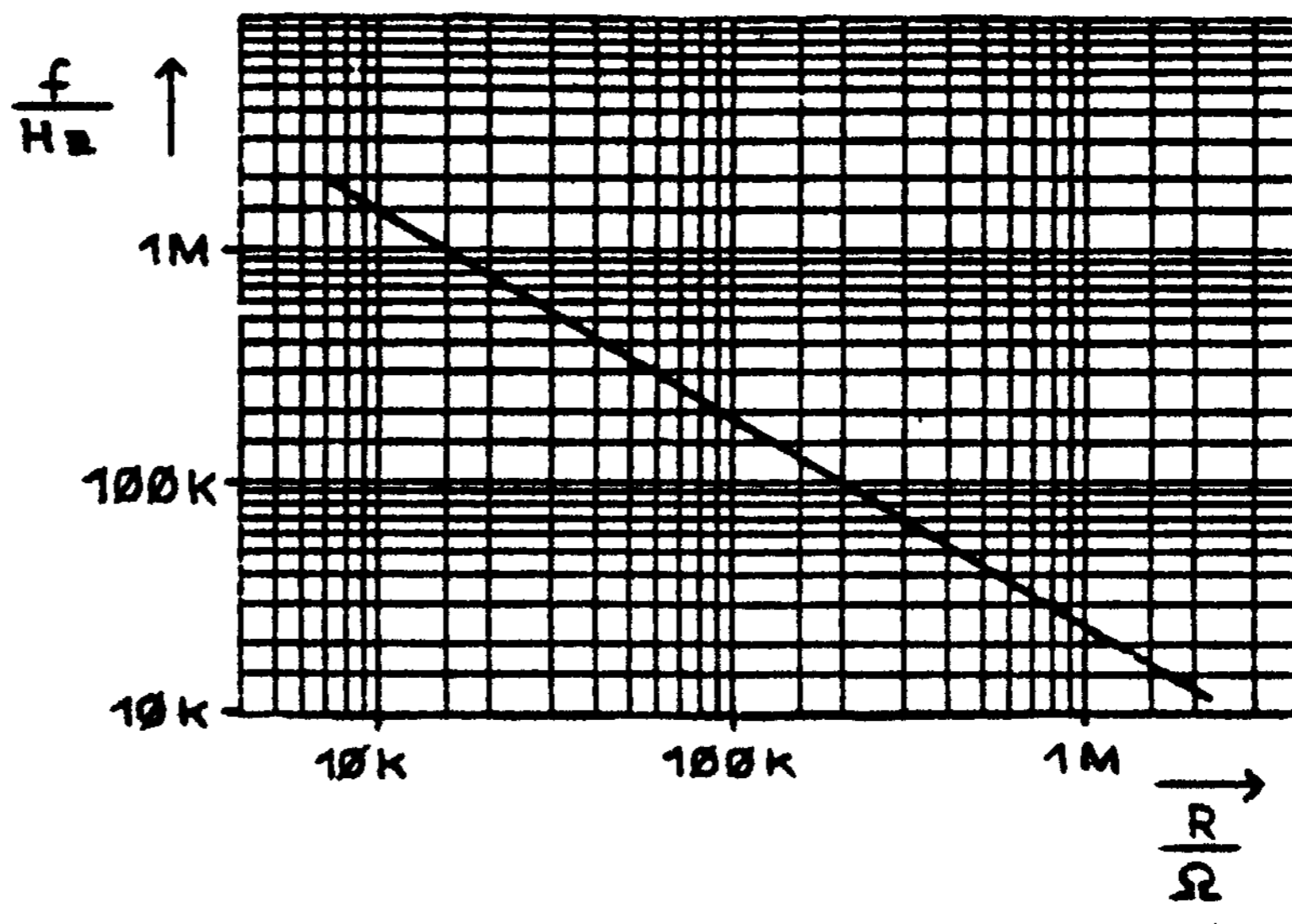


Abb. 18: Abhängigkeit der Taktfrequenz f vom Widerstand R für den Taktoszillator der CPU

Die Zusammenschaltung der gesamten Takteinheit des WDR-1-Bit-Computers zeigt Abb. 19.

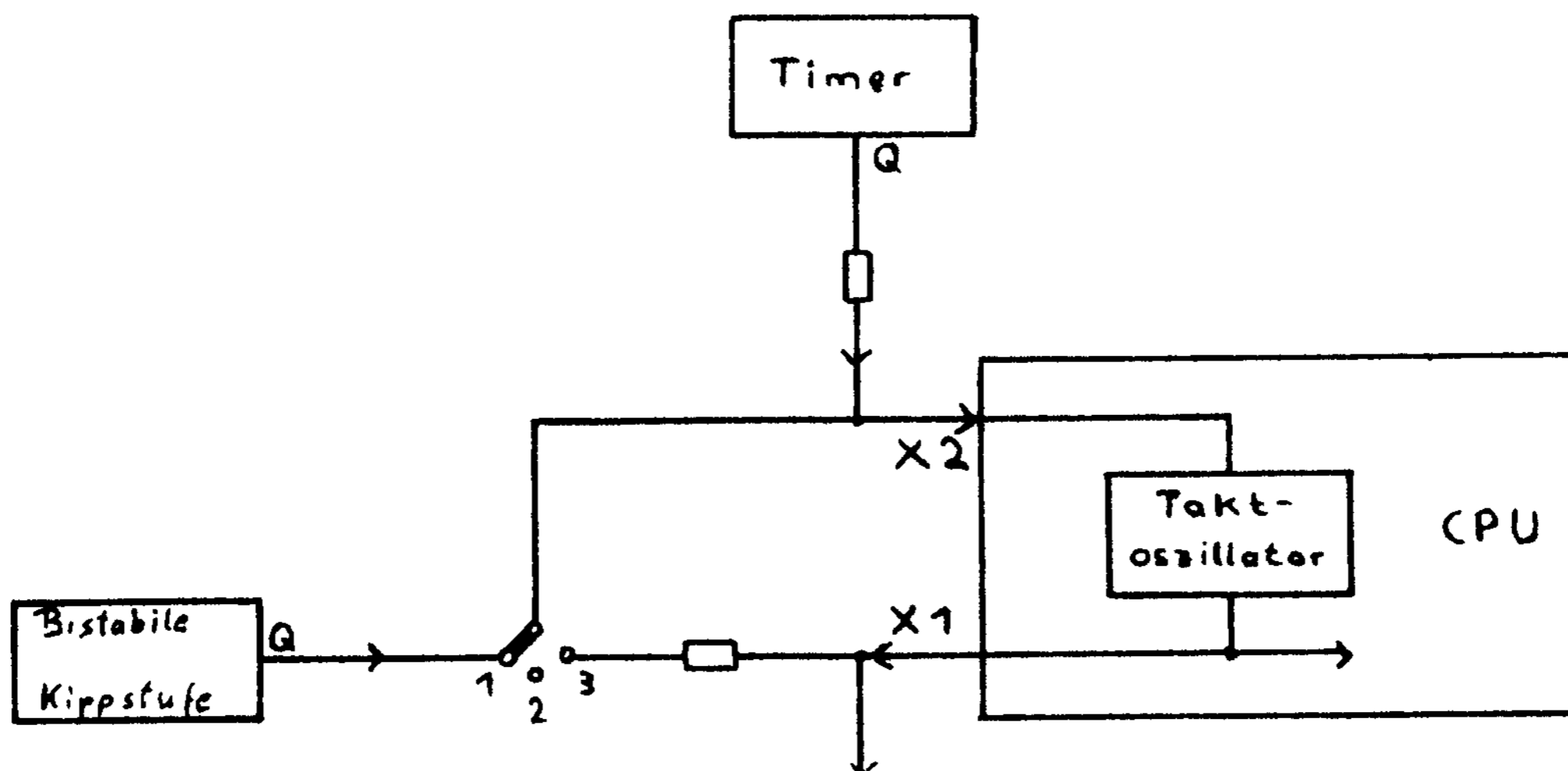


Abb. 19: Die Takteinheit

Folgende Betriebsmöglichkeiten ergeben sich:

1. In Stellung 1 (Hand-Takt) und 2 (Langsam-Takt) wird dem Taktoszillator ein fremdes Taktsignal an Eingang X2 eingeprägt. Dieses erscheint an X1 invertiert. Von dort beziehen alle Bausteine, die einen Takt benötigen, ihr Taktsignal.

In Stellung 1 sind die bistabile Kippstufe und der Timer mit X2 verbunden. Der Timer ist nur "lose" angekoppelt, da zwischen ihm und X2 ein hochohmiger Widerstand liegt. (Hochohmige Ankopplung bedeutet lose Ankopplung) Da der Ausgang Q der bistabilen Kippstufe niederohmig ist, wird dessen Taktsignal wirksam. In Ruhestellung der bistabilen Kippstufe, d.h., wenn der Takt-Taster nicht betätigt wird, liegt am Ausgang Q der bistabilen Kippstufe und damit am Eingang X2 eine "0".. Am Ausgang X1 liegt dann eine "1".

In der Stellung 2 ist nur der Timer mit X2 verbunden. Dessen Taktsignal erscheint invertiert an X1.

2. In Stellung 3 (Schnell-Takt) wird das eigene Taktsignal der CPU über einen Widerstand rückgekoppelt. Auch hier wird das Signal des Timers wegen der losen Ankopplung nicht wirksam. Der Rückkopplungswiderstand muß also wesentlich kleiner sein als der Ankopplungswiderstand des Timers.

2.2.9. DIE STROMVERSORGUNG

Die Stromversorgung wird benötigt, um aus der Netzspannung (220 V Wechselspannung) die Betriebsspannung (5 V Gleichspannung) zu erzeugen.

Die Stromversorgung besteht aus einem dem WDR-1-Bit-Computer vorzuschaltenden Transformator mit einer Ausgangsspannung von ca. 9 V. Er sollte mindestens 350 mA Strom liefern können. Die Abb.20 zeigt die Elemente der Stromversorgung ohne den Transformator.

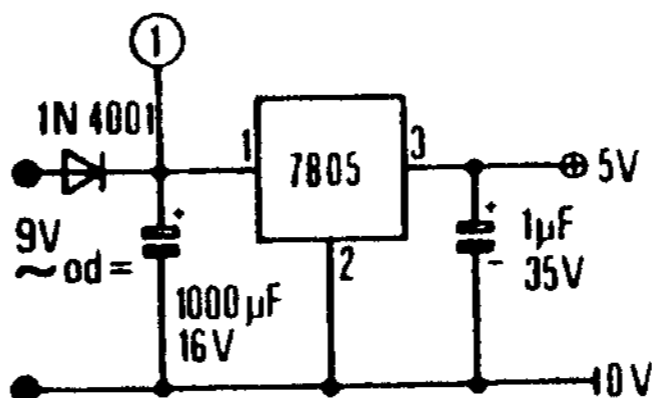


Abb. 20 Die Stromversorgung

Die Stromversorgung besteht aus einer Diode, zwei Kondensatoren und aus einem Spannungsregler.

Die Diode richtet die Wechselspannung gleich. (Wird eine Gleichspannung von ca. 9 V als Stromversorgung für den WDR-1-Bit-Computer benutzt, dient die Diode als Verpolungsschutz).

An der Diode fallen im leitenden Zustand ca. 0,6 V Spannung ab. Ist die Ausgangsspannung des Transformators um 0,6 V größer als die Kondensatorspannung, so leitet die Diode und der Kondensator wird aufgeladen, also steigt die Kondensatorspannung. Die maximale Kondensatorspannung beträgt also $12,7 \text{ V} - 0,6 \text{ V} = 12,1 \text{ V}$.

Über die nachfolgende Schaltung, dem WDR-1-Bit-Computer, würde der Kondensator aber immer wieder teilweise entladen, abgesehen davon, daß diese Spannung zu groß wäre. Um die durch den Verbraucher entstehenden Schwankungen der Kondensatorspannung aufzufangen, wird ein Spannungsregler hinter den Kondensator geschaltet. Der Spannungsregler 7805 hat drei Anschlüsse: Pin 1 ist der Eingang (hier: ca. 12,1V), Pin 2 ist mit Masse verbunden und Pin 3 ist der Ausgang (5 V). Die Ausgangsspannung des Spannungsreglers ist die stabilisierte Betriebsspannung für den WDR-1-Bit-Computer. Ein Kondensator ist zwischen Pin 2 und Pin 3 geschaltet, um Störungen, die durch Ein- und Ausschaltvorgänge verursacht werden, zu unterdrücken. Auch wenn der Spannungsregler bei Betrieb des WDR-1-Bit-Computers heiß wird, ist es nicht nötig, den Spannungsregler zum Schutz gegen Übertemperatur mit einem Kühlkörper zu versehen.

2.2.10. DER 6-FACH-INVERTER CD 4049

Die beiden Inverter des 1-Bit-Computers, die einen Teil der Takteinheit darstellen, sind Bestandteile des Bausteins CD 4049. Der 4049 ist ein 6-fach-Inverter.

Abb. 21 zeigt den Aufbau eines Bausteins.

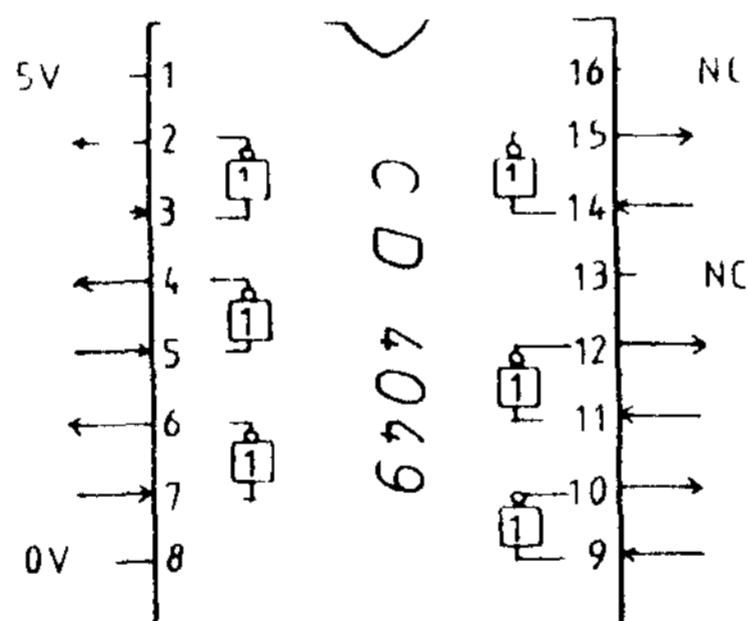


Abb. 21: Aufbau des CD 4049

Für die Takteinheit werden nur zwei Inverter benötigt. Gewählt wurden die Inverter 1 (Pin 3 und 2) und Inverter 6 (Pin 14 und 15). Pin 13 und 16 sind beim 4049 nicht angeschlossen (engl.: Not connected = NC). Die Eingänge der unbenutzten Inverter werden an Masse oder an die Betriebsspannung angeschlossen, damit diese einen definierten Eingangszustand haben.

2.2.11. DAS ODER-GLIED

Das ODER-Glied des 1-BIT-Computers, das an Pin 6 des CD 4051 angeschlossen ist, kann der Einfachheit halber aus einzelnen Bauelementen aufgebaut werden. Eine entsprechende Schaltung wurde bereits in Abb. 7 vorgestellt.

2.3. DAS ZUSAMMENWIRKEN DER FUNKTIONSEINHEITEN

Abb. 22 zeigt den Gesamtschaltplan des WDR-1-Bit-Computers. Aus ihm kann die Dimensionierung der Bauelemente entnommen werden. Die Eingabe von Programmen erfolgt durch einfache Schalter, die in einer weiteren Stufe durch eine Tastatur ersetzt werden können.

Da der Gesamtschaltplan (Abb.22) für einen Anfänger doch sehr komplex wirkt, sind in den Abb. 23 bis 25 Auszüge aus dem Gesamtschaltplan dargestellt. Dort wird das Zusammenwirken der wichtigsten Bausteine, also des Mikroprozessors, des Eingabebausteins, des Ausgabebausteins, der Programmspeicherbausteine und der Programmzählerbausteine, gezeigt, wobei jeweils andere Bausteine in den Mittelpunkt der Betrachtung rücken.

Abb. 23 zeigt die Betrachtung des Eingabebausteins, Abb. 24 zeigt die Beschaltung des Ausgabebausteins und Abb. 25 zeigt die Beschaltung der Programmspeicher- und Programmzählerbausteine. Eine gesonderte Abbildung, in der der Mikroprozessor in den Mittelpunkt rückt, erfolgt nicht, da dies schon im Gesamtschaltplan geschieht.

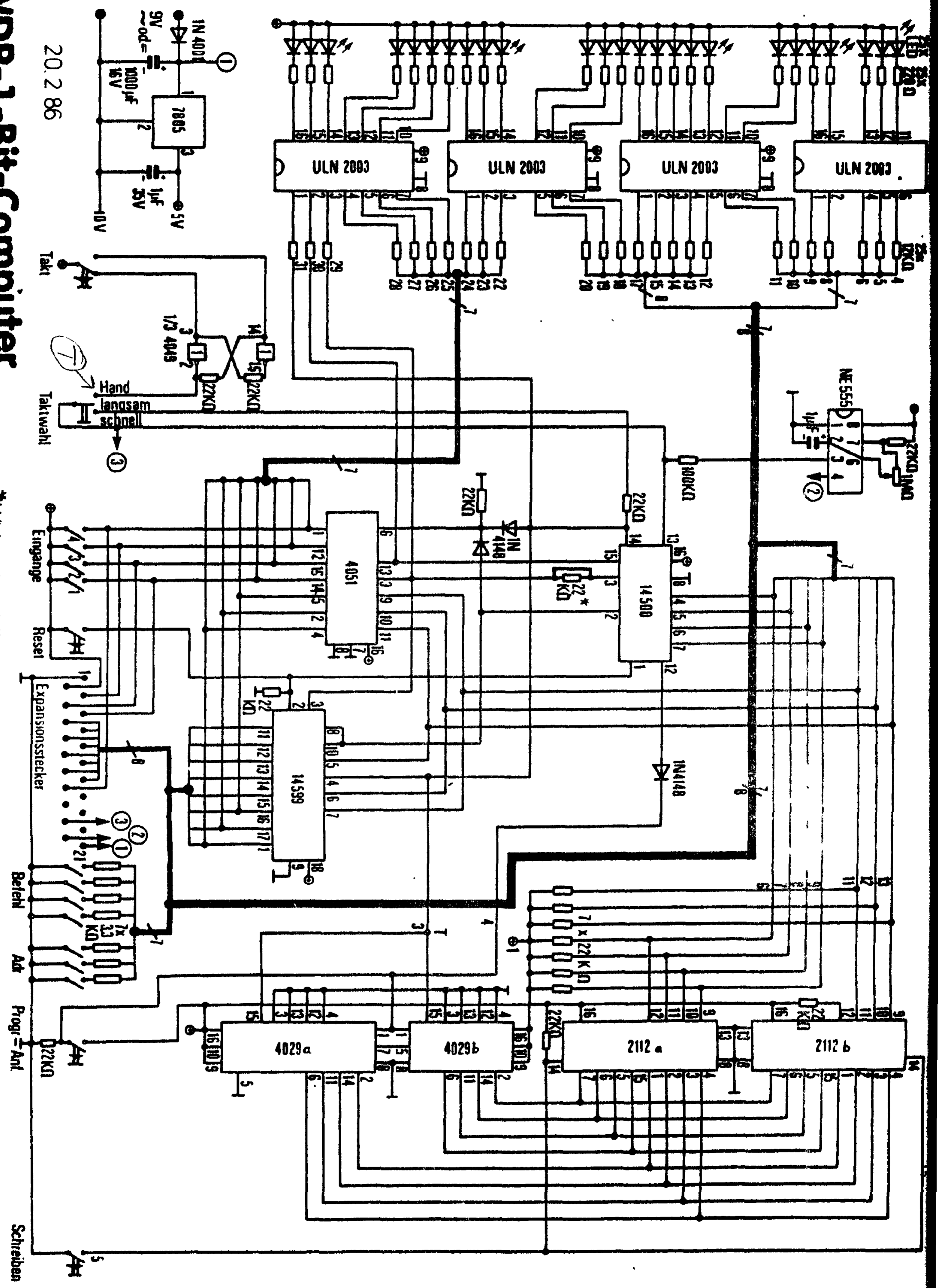
WICHTIGER HINWEIS: Der 22 Kiloohm-Widerstand der Datenleitung zwischen Pin 3 des 14500, Pin 3 des 4051 und Pin 3 des 14599 sollte fehlen, d.h. durch eine Drahtbrücke ersetzt werden, da bei einigen Eingabebausteinen 4051 der WDR-1-Bit-Computer im Schnelltakt nicht läuft. Er war ursprünglich als Schutzwiderstand gedacht.

Für den Expansionsstecker gilt folgende Anschlußbelegung:

Anschluß 1:	0 V	Masseanschluß
Anschluß 2:	5 V	Betriebsspannung
Anschluß 3:	E 3	Eingangsleitung Nr. 3
Anschluß 4:	E 2	Eingangsleitung Nr. 2
Anschluß 5:	E 1	Eingangsleitung Nr. 1
Anschluß 6:	A 7	Ausgangsleitung Nr. 7
Anschluß 7:	A 6	Ausgangsleitung Nr. 6
Anschluß 8:	A 5	Ausgangsleitung Nr. 5
Anschluß 9:	A 4	Ausgangsleitung Nr. 4
Anschluß 10:	A 3	Ausgangsleitung Nr. 3
Anschluß 11:	A 2	Ausgangsleitung Nr. 2
Anschluß 12:	A 1	Ausgangsleitung Nr. 1
Anschluß 13:	A 0	Ausgangsleitung Nr. 0
Anschluß 14:	E 4	Eingangsleitung Nr. 4
Anschluß 18:	Eingang für eine digitale Abfrage	
Anschluß 20:	Chip-Enable-Anschluß vom NE 555	
Anschluß 21	Stromversorgungsanschluß für Peripheriebausteine.	

Das Zusammenwirken der Funktionseinheiten wird durch den Takt gesteuert. Welche Baugruppen angesprochen werden, wird durch die Befehlsart bestimmt. Die Freigabebefehle und der Programmverzweigungsbefehl wurden bereits behandelt. Daher soll hier nur das Zusammenwirken der Funktionseinheiten bei den Eingabebefehlen, den Ausgabebefehlen, den Rechenbefehlen und den Steuerbefehlen erklärt werden. Wir gehen hier davon aus, daß bereits ein Programm im Programmspeicher steht und die Ein- und die Ausgabe freigegeben ist. Wie ein Programm eingegeben wird,

WDR-1-1-Bit-Computer



* Widerstand überbrücken!

Abb. 22: Schaltplan des WDR-1-1-Bit-Computers

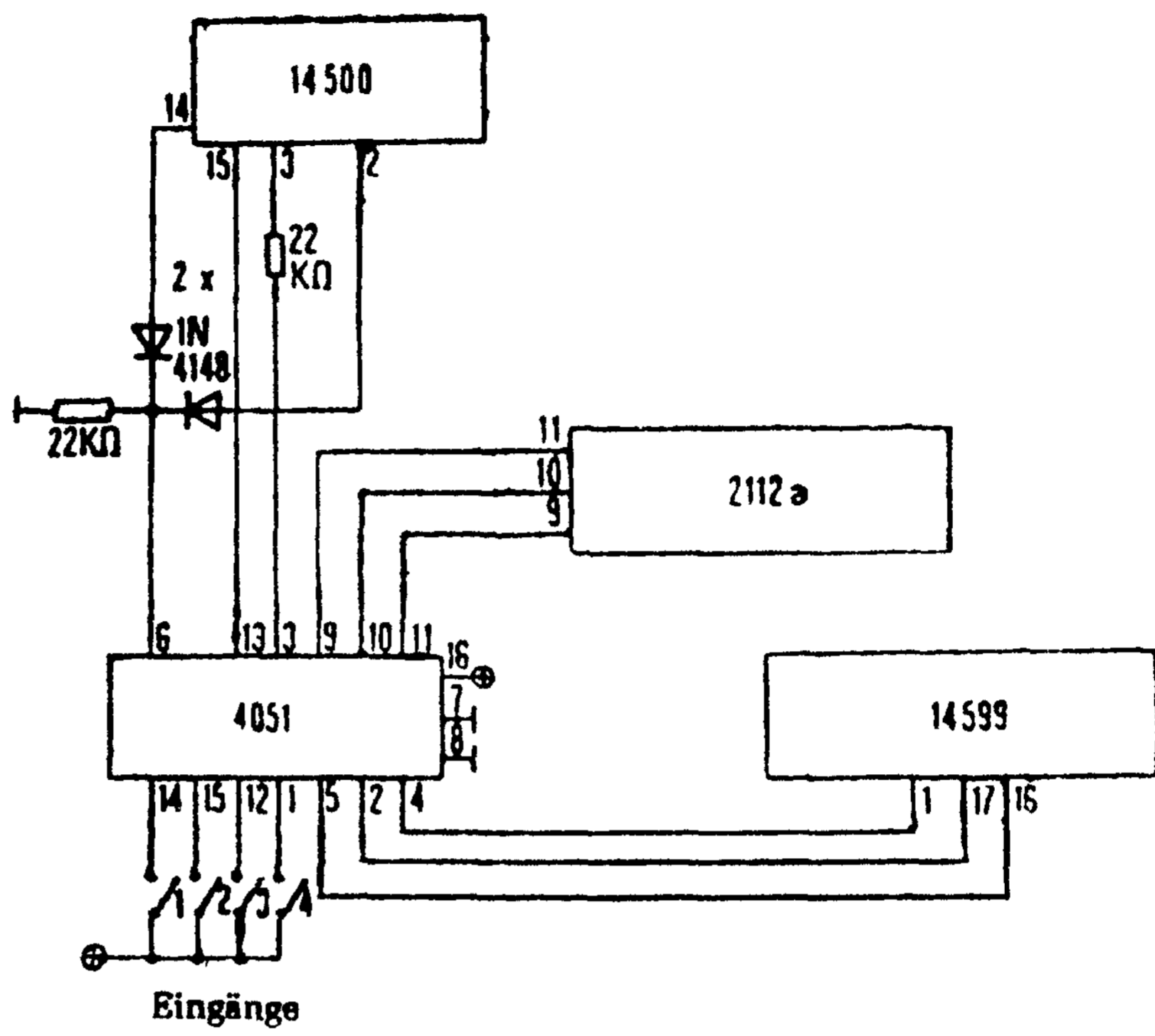


Abb. 23: Eingabeblock

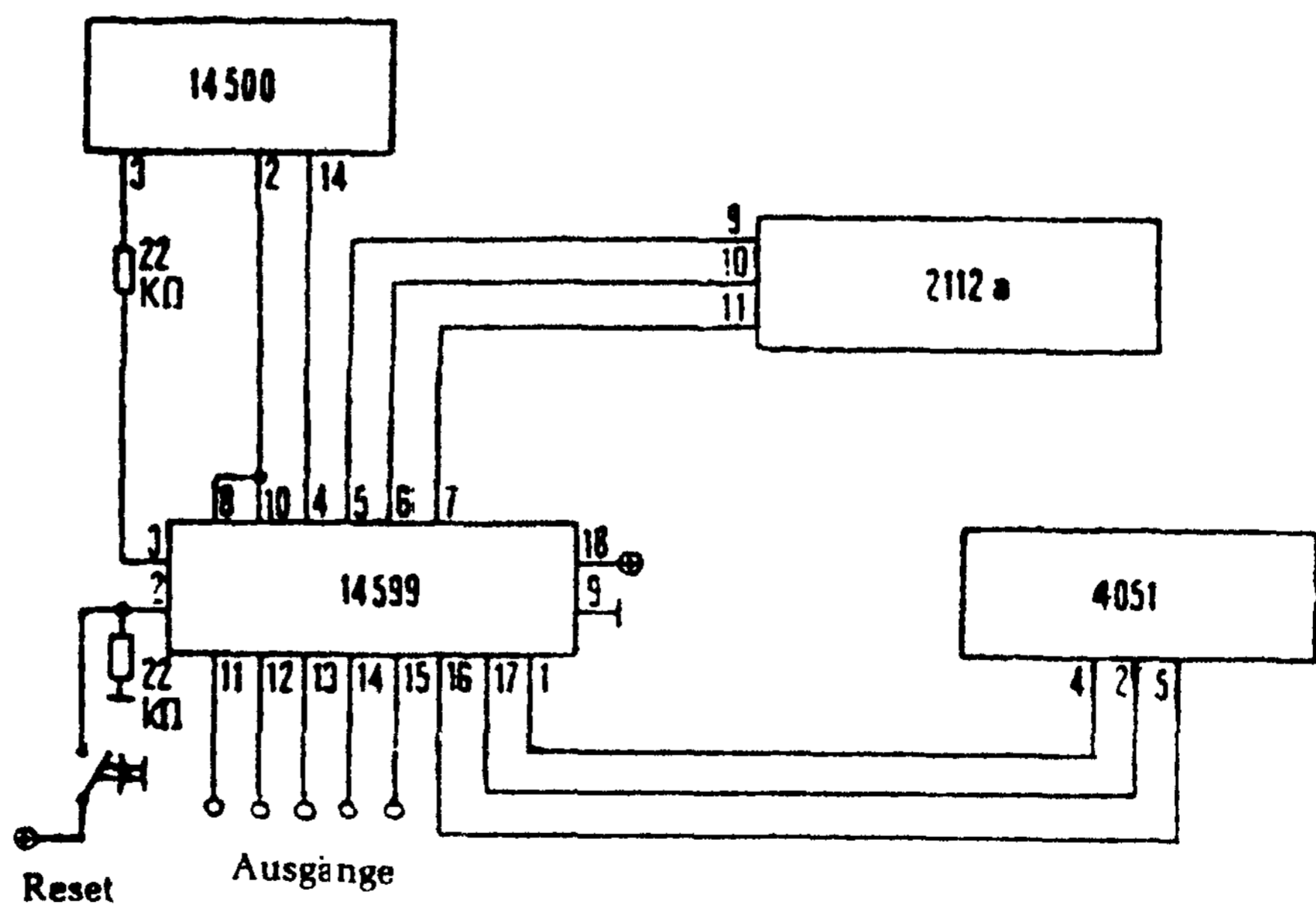


Abb. 24: Ausgabeblock

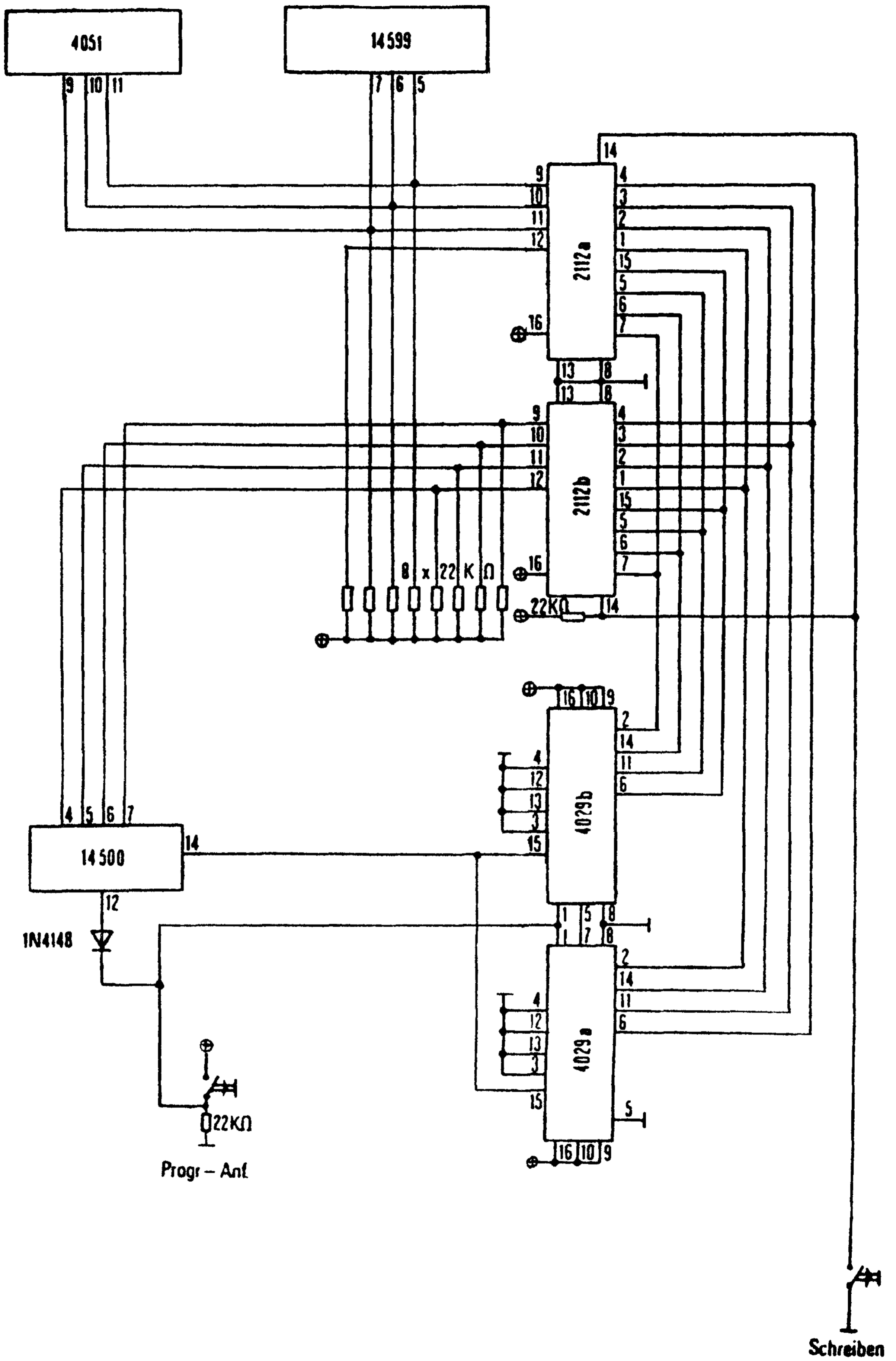


Abb. 25: Zähler- und Speicherblock

soll hier nicht erläutert werden (Siehe: Kapitel 4.2.). Für die Abarbeitung eines Programmschrittes wird immer nur eine Taktperiode benötigt.

1. Taktzustand "1"

Der Programmzähler steht auf einer bestimmten Adresse. Der Programmspeicher gibt den zu dieser Adresse gehörenden Programmschritt aus. Der Befehl gelangt zum Mikroprozessor und die Ein-/Ausgabeadresse zum Eingabebaustein und zum Ausgabebaustein.

2. Negative Taktflanke

Der Befehl wird in das Befehlsregister des Mikroprozessors eingelesen.

3. Taktzustand "0"

In der Kontrolleinheit des Mikroprozessors wird der Befehl decodiert.

Eingabebefehl: Die Schreib/Leseleitung enthält eine "0". Somit ist der Eingabebaustein frei und der Ausgabebaustein gesperrt. Im Eingangsbaustein wird eine der acht Eingangsleitungen mit der Datenleitung verbunden. Dabei bestimmt die Ein-/Ausgabeadresse die Nummer der Eingangsleitung. Die Zentrale Logikeinheit des Mikroprozessors erhält die Information der Datenleitung und damit die Information des betreffenden Eingangs. Außerdem erhält die Zentrale Logikeinheit von der Kontrolleinheit die Anweisung, diese Information entweder direkt zu ihrem Ausgang weiterzuleiten oder sie erst zu invertieren und dann weiterzuleiten.

Logikbefehl: In gleicher Weise wie beim Eingabebefehl gelangt hier auch die Information einer Eingangsleitung über die Datenleitung zur Zentralen Logikeinheit. Außerdem gelangt der Inhalt des Ergebnisregisters über den zweiten Dateneingang in die Zentrale Logikeinheit. Je nach Logikbefehl werden die Werte der beiden Eingänge logisch miteinander verknüpft. Das Ergebnis dieser logischen Verknüpfung erscheint am Ausgang der Zentralen Logikeinheit.

Ausgabebefehl: Die Schreib-/Leseleitung enthält eine "1". Somit ist der Ausgabebaustein frei und der Eingabebaustein gesperrt. Der Mikroprozessor schaltet, je nach Ausgabebefehl, entweder den Inhalt des Ergebnisregisters oder dessen Komplement auf die Datenleitung. Die Information der Datenleitung gelangt zum Ausgabebaustein. Dort wird sie auf einen der acht Ausgänge gesetzt. Dabei bestimmt wieder die Eingabe-/Ausgabeadresse die Nummer der Ausgangsleitung. Die Information an diesem Ausgang bleibt solange erhalten, bis sie gelöscht wird oder durch eine andere Information überschrieben wird.

Steuerbefehl: Die Schreib-/Leseleitung enthält eine "0". In gleicher Weise wie beim Eingabebefehl und beim Logikbefehl gelangt auch hier die Information einer Eingangsleitung über die Datenleitung zur Zentralen Logikeinheit. Diese wird jedoch nicht bearbeitet. Je nach Steuerbefehl gelangt an einen der Steuerausgänge des Mikroprozessors eine "1". Da nur der zum Steuerbefehl JMP zugehörige Steuerausgang beschaltet ist, wird

dieser auch hier nur näher behandelt. Dieser Steuerausgang ist mit dem Programmzähler verbunden. Eine "1" auf dieser Leitung setzt die Programmadresse auf 0.

4. Positive Taktflanke

Der Programmzähler schreitet um eine Zählung fort. Bei einem Eingabebefehl und bei einem Logikbefehl wird die Information vom Ausgang der Zentralen Logikeinheit in das Ergebnisregister gespeichert.

Bei einem Ausgabebefehl geht die Schreib-/Leseleitung wieder auf "0".

5. Taktzustand "1"

Der Eingabebaustein ist gesperrt, da der Takt "1" ist. Der Ausgabebaustein ist gesperrt, da die Schreibleseleitung "0" ist. (Der Programmzähler leitet die neue Programmadresse zum Programmspeicher weiter. Dieser gibt den zu dieser Adresse gehörenden Programmschritt aus.)

6. Negative Taktflanke

War der letzte Befehl ein Steuerbefehl, so wird jetzt das zugehörige Steuersignal zurückgenommen, d.h., die "1" an dem Steuerausgang wird wieder zu "0".

(Der neue Befehl wird in das Befehlsregister des Mikroprozessors eingespeichert.)

3. DER ZUSAMMENBAU

3.1. BAUSATZ UND WERKZEUG (aus (14))

Grundsätzlich ist der WDR-1-Bit-Computer so konzipiert, daß alle Bauteile bis auf die fertig geätzten Platinen im Elektronikfachhandel einzeln erhältlich sind. Da diese Beschaffungsart aber sehr umständlich ist, halten wir es für sinnvoll, einen kompletten Bausatz einschließlich der Platinen bei der Firma raffel-elektronic (vgl. Anhang) zu beziehen. Ersatzteile können jederzeit nachgekauft werden.

Damit die Kosten für den Bausatz niedrig gehalten werden, enthält er entsprechend den vier Funktionseinheiten des Computers vier Einzeltüten mit den jeweiligen unsortierten Bauelementen. Wir werden auf ihren Inhalt näher eingehen, wenn wir die einzelnen Funktionsgruppen aufbauen, die Anzeige-, die Speicher-, die Prozessoreinheit und die Grundplatine. Der Bausatz enthält nicht die Tastatur. Dennoch werden wir ihren Aufbau und ihre Funktion beschreiben, da sie das Programmieren erheblich vereinfacht. Die Tastatur ist ebenfalls als Bausatz erhältlich, jedoch bei der Firma DATANorf (vgl. Anhang).

Nur 5 Dinge braucht der Computerbauer, zumindest in unserem Falle:

- eine Bohrmaschine mit Bohrständer und Bohrern
- einen Lötkolben mit Lötzinn
- einen Seitenschneider
- eine Säge
- ein Meßgerät

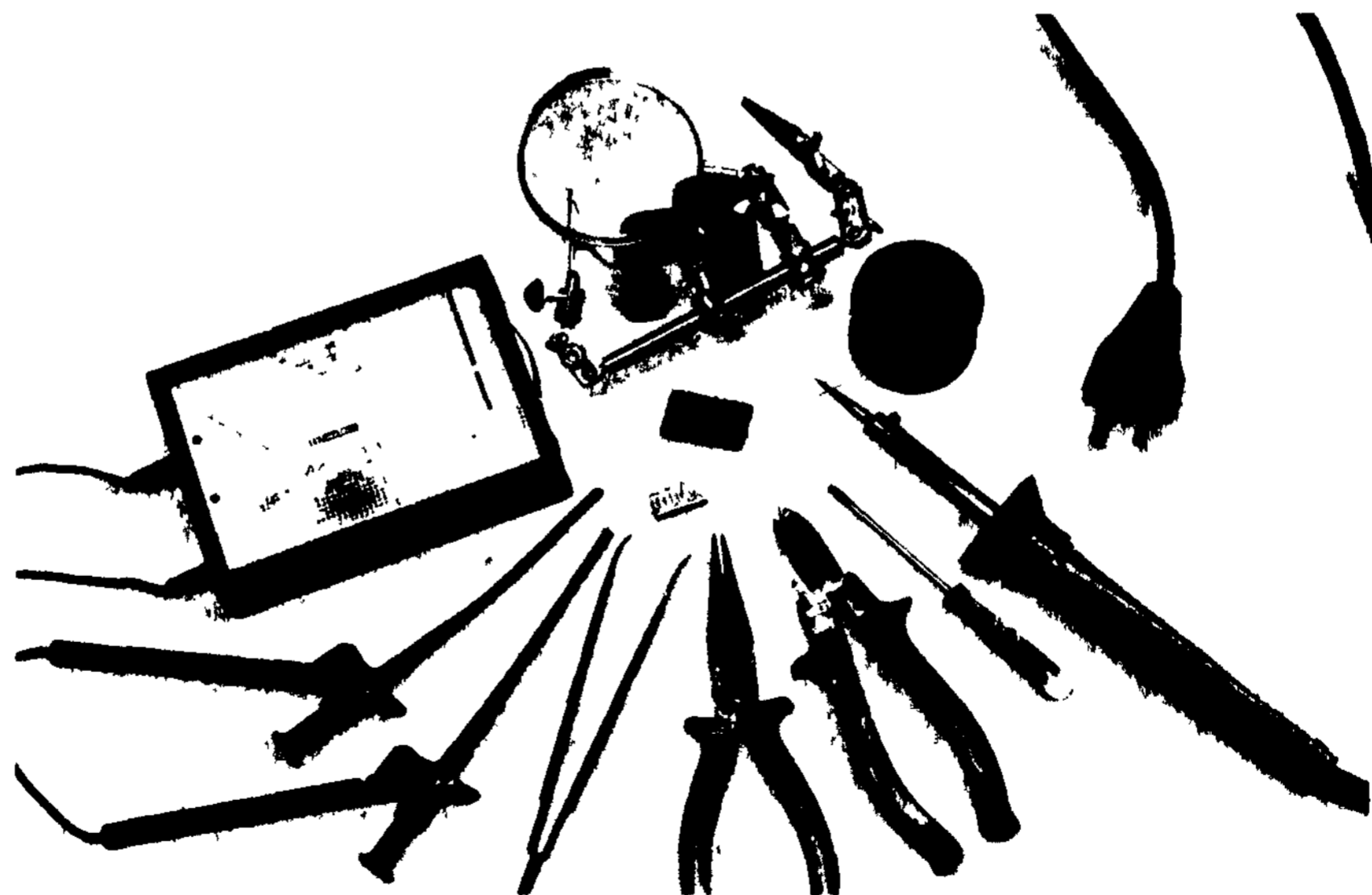


Abb. 26: Dinge, die der Computerbauer braucht

Die Bohrmaschine

Im Prinzip eignet sich jede Bohrmaschine, sofern das Bohrfutter die Bohrer von 1 mm bis 5 mm Durchmesser fassen kann. Bewährt haben sich Kleinbohrmaschinen für Niedervoltbetrieb, weil sie handlicher und sehr viel leiser sind.

Für den Bau des vorliegenden Computers werden die Metallbohrerstärken 1mm, 1,5 mm, 2 mm und 5 mm benötigt.

Der LötKolben

Bei der Wahl des LötKolbens ist man eingeschränkter. Hier eignen sich nur ElektroniklötKolben mit feiner Lötspitze und einer Leistung um 20 Watt. Werden andere LötKolben verwendet, besteht die Gefahr, daß die Platinen zerstört werden. Bei zu groben Lötspitzen entstehen sehr leicht Lötfehler. Bei zu geringer Leistung, unter 15 Watt, fließt das Lötzinn nicht mehr richtig, bei zu hoher Leistung, über 25 Watt, verdampft das im Lötdraht enthaltene Flußmittel (Colophonium) sehr schnell. Dann besteht die Gefahr, daß die Lötstelle schlecht wird. Als Lötdraht eignet sich feiner Elektroniklötdraht von ca. 1 mm Durchmesser.

Lötübungen: auf das Gefühl kommt es an

Für diejenigen, die noch nie mit einem LötKolben gearbeitet haben, empfehlen wir, aus Teilen nicht mehr verwendeter Elektronikgeräte, wie Radio, Fernseher, Computer o.ä., einzelne Bauteile herauszulöten. Am besten eignen sich eingelötete Bauelemente mit nur zwei Anschlüssen. Dazu wird auf der einen Seite des Bauteileträgers, der Platine, mit einer Pinzette ein Anschluß angefaßt, auf der anderen Seite wird die Lötstelle so lange mit der LötKolbenspitze erhitzt, bis sich der Anschluß "butterweich" herausziehen läßt. Auf diese Weise erhält man sehr schnell das richtige Gefühl für eine genügend heiße Lötstelle.

Um das fehlerfreie Anlöten zweier Verbindungsstellen zu üben, klemmen wir einen Anschluß eines ausgelöteten Bauteils in einen kleinen Schraubstock und versuchen dann, ein anderes Bauteil anzulöten (siehe Anhang). Wenn dies mehrfach gelungen ist, können wir mit dem Bau unseres Computers beginnen.

Der Seitenschneider

Einen Elektronik-Seitenschneider benutzen wir, um überstehende Anschlüsse der Bauteile fachgerecht abzuschneiden. Nimmt man eine gewöhnliche Kneifzange, riskiert man, die Leiterbahn durch die Hebelwirkung beim Kneifen abzureißen.

Die Säge

Um einzelne Steckerleisten zu kürzen, reicht eine feine Handsäge.

Das Meßgerät

Zur Kontrolle des gelungenen Computeraufbaus brauchen wir ein einfaches Vielfachmeßgerät. Die unterste Preisklasse von ca. 25 DM für ein solches Meßgerät ist vollkommen ausreichend. Es sollte allerdings die Meßbereiche für Volt, Ampere und Ohm umfassen.

Der Gebrauch eines Oszilloskopes ist zwar sehr anschaulich, jedoch in unserem Fall eigentlich überflüssig. Mit diesen 5 Geräten kommen wir aus. Erleichtern können wir uns die Arbeit allerdings, wenn wir zusätzlich noch einige andere Geräte benutzen :

- Ein Biegewerkzeug, um die Anschlußdrähte einzelner Bauteile so zu biegen, daß sie problemlos in die entsprechenden Löcher der Platine passen.
- Eine Pinzette mit stumpfer, geriffelter Spitze, um kleine Anschlüsse zurechtzubiegen.
- Einen Kleinschraubstock, um die abzusägenden Bauteile festzuklemmen.

3.2. DIE ANZEIGEPLATINE

Der Bausatz enthält unter anderem eine Tüte mit den Teilen für die Anzeigeeinheit. Sie sind in der Abb. 27 dargestellt und aufgelistet.

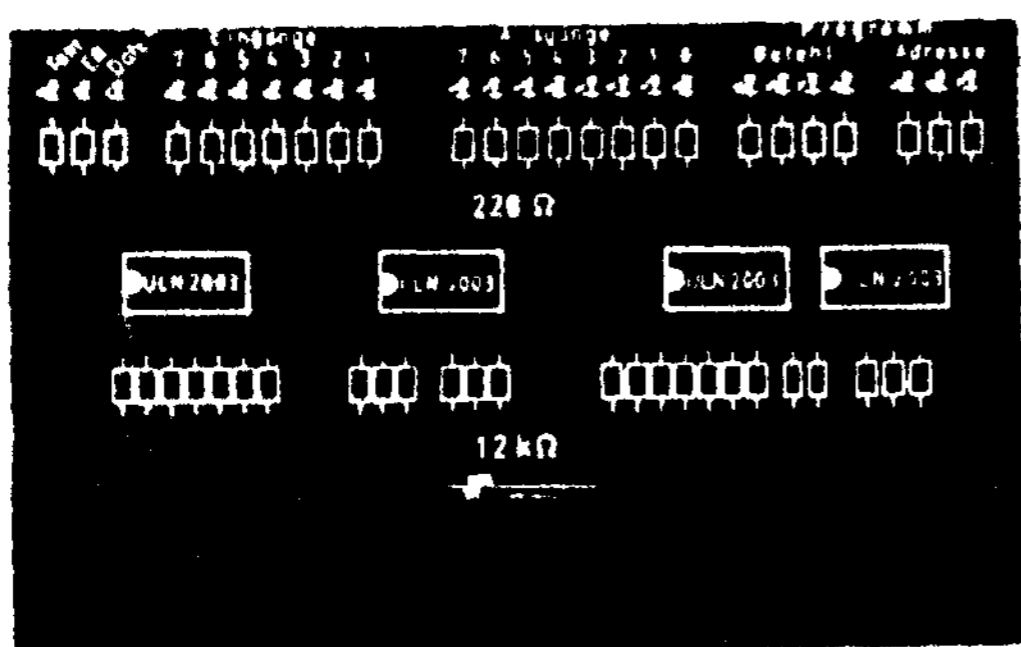


Abb. 27: Bauteile für die Anzeigeeinheit: Platine, Widerstände, ICs, Fassungen für die ICs, Leuchtdioden, Steckerleiste.

Der erste Arbeitsschritt besteht darin, diese Platine zu bohren. Dazu legen wir sie mit der Lötseite nach oben (vgl. Abb. 28) unter den Bohrer im Bohrständer.

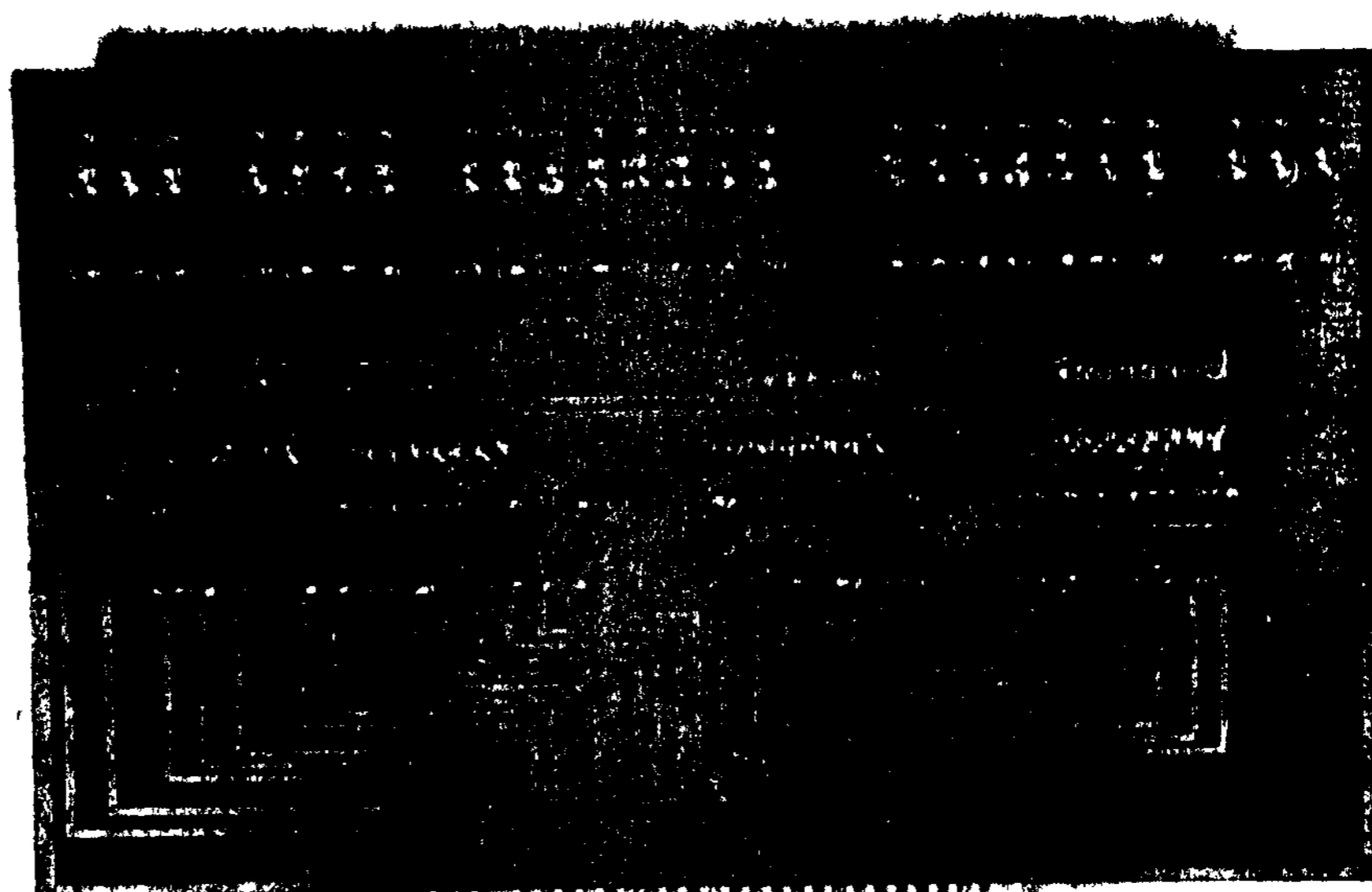


Abb. 28: Lötseite der Anzeigeplatine

Auf der Lötseite erkennen wir deutlich die Lötungen mit den Bohrpunkten. Beim Bohren mit dem 1 mm Bohrer müssen wir diese Bohrpunkte genautreffen, dann hat der Bohrer automatisch die richtige Führung durch das Loch im Lötauge. Er rutscht dann nämlich nicht mehr unkontrolliert auf dem Lötauge herum, sondern bleibt in dem kleinen kupferfreien Punkt im Zentrum des LötAuges.

Die Bauteile werden eingesetzt

Wenn sämtliche Lötungen durchbohrt sind, setzen wir die Bauteile ein. Der Aufdruck der Bestückungsseite gibt uns an, welche Bausteine wohin gehören.

Fangen wir mit den Widerständen an. Es gibt hier zwei verschiedene Werte, zum einen 220 Ohm, zum anderen 12 Kiloohm, die jeweils bestimmt codiert sind (vgl. Abb.29). Die Bedeutung der einzelnen Ringe in Abhängigkeit ihrer Farbe und Position ist im Anhang genau aufgeführt.

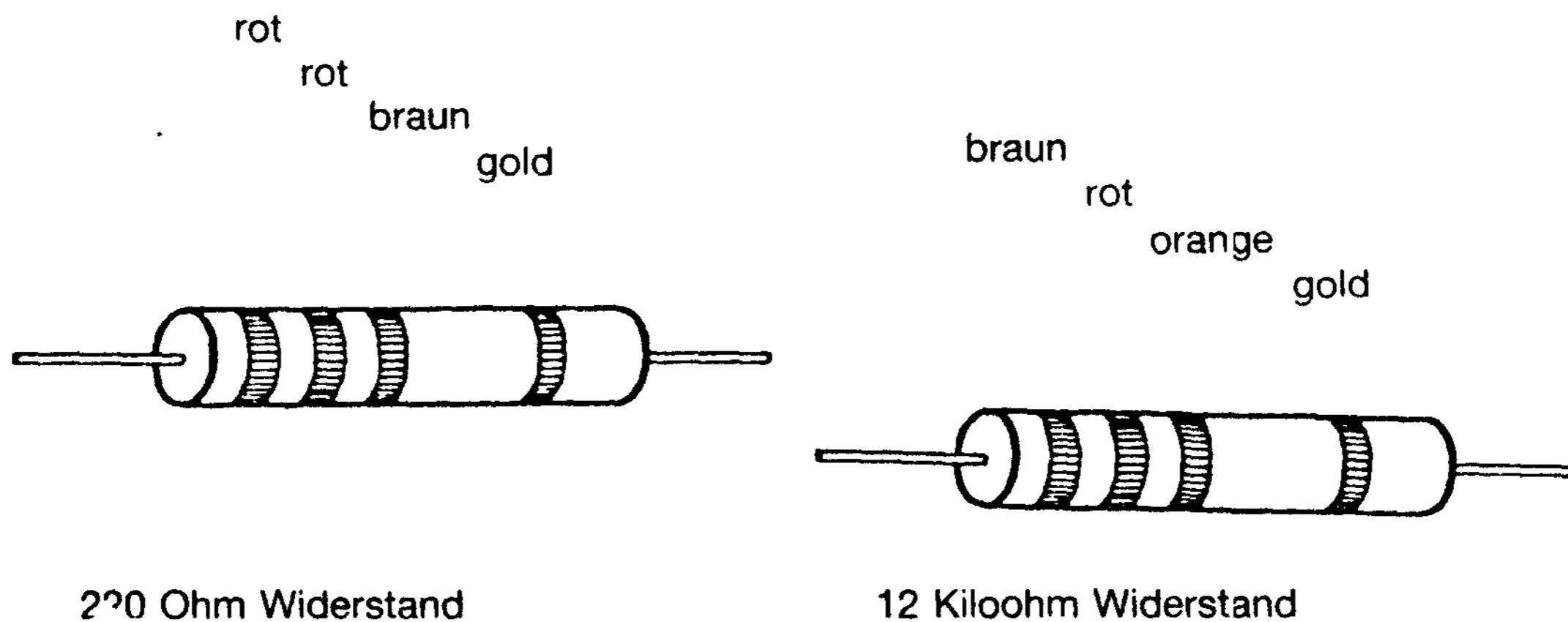


Abb. 29: Widerstände mit ihrer Codierung

Biegen wir zunächst die Anschlüsse der 220-Ohm-Widerstände, die den Farbcode "rot-rot-braun-gold" tragen, auf die richtige Länge, so daß sie mühelos in die Bohrlöcher passen. Dies geht besonders schnell mit dem Biegewerkzeug.

Wir empfehlen, die Widerstände in gleicher Ausrichtung ihrer Farben in die Platine einzusetzen. So erleichtern wir eine eventuelle Fehlerüberprüfung. Da wir die Platine zum Löten nun umdrehen müssen, winkeln wir vorher die herausstehenden Anschlußdrähte leicht ab. Dann fallen die Widerstände nicht heraus. Außerdem liegen die Bauteile eng an der Platine an. Dadurch wird verhindert, daß bei mechanischer Belastung die Bauteile durchgedrückt werden und so Leiterbahnen zerreißen.

Ist dies getan, löten wir einen Widerstand nach dem anderen an. Zuerst plazieren wir die heiße Lötspitze so, daß das entsprechende Lötauge und der Anschlußdraht im Lötbereich gleichzeitig erhitzt werden. Dann erst führen wir ein wenig Lötzinn zu. Wenn das Lötzinn auf dem Lötauge gut verlaufen ist, lassen wir die Lötstelle abkühlen. Ist eine Lötstelle nicht heiß genug geworden, eine sogenannte "kalte" Lötstelle, kann der elektrische Strom nicht gut fließen. Sieht sie so aus, wie ganz rechts in Abb. 30, dann müssen wir noch einmal nachlöten. Die Abb. 30 faßt die geschilderten Vorgänge noch einmal zusammen.

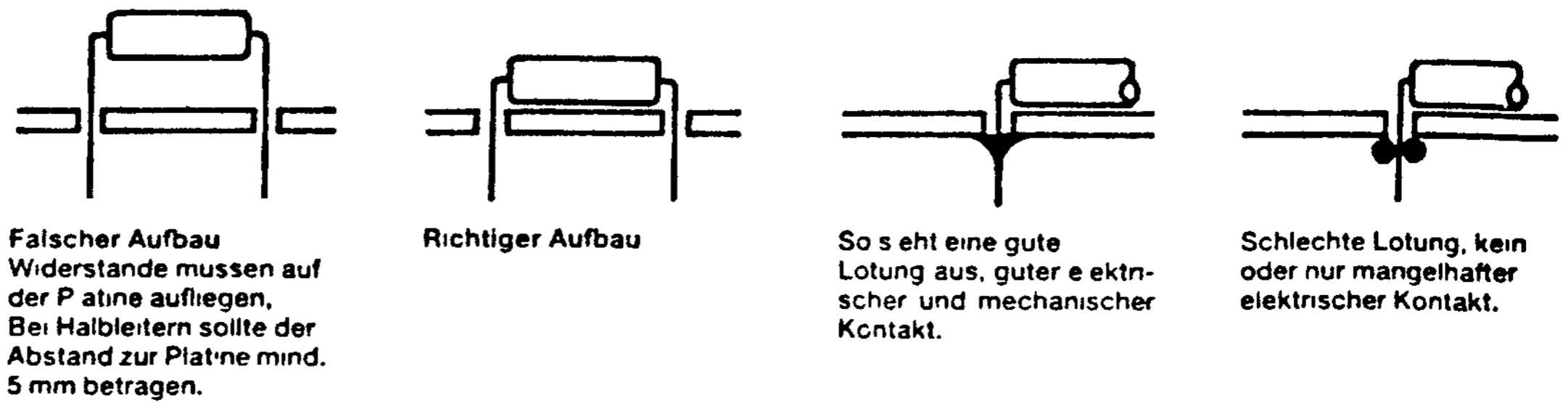


Abb. 30: Einsetzen und Einlöten von Bauelementen

Sind alle 25 Widerstände richtig eingelötet, werden ihre überstehenden Enden mit dem Seitenschneider abgekniffen. Die abgekniffenen Drahtenden sollten wir aufbewahren. Wir benötigen sie noch an späterer Stelle (für anzufertigende Durchkontaktierungen).

Den gleichen Vorgang wiederholen wir mit den 12-Kiloohm-Widerständen. Diese 25 Widerstände haben den Farbcode "braun-rot-orange-gold".

Als nächstes setzen wir die Leuchtdioden (LEDs) ein. Dabei müssen wir auf die Polung (vgl. Abb.31) und die Farben der LEDs achten. Der kurze Anschluß ist die Kathode, die mit dem Minuspol verbunden werden muß. Die LEDs werden so in die Platine eingesetzt, daß die kurzen Anschlüsse an die 220-Ohm-Widerstände gebracht werden. Also stecken wir die langen Anschlüsse in die obere Lochreihe, die leitend miteinander verbunden ist. Die gelbe LED findet ihren polrichtigen Platz in den Löchern bei "Daten", sie ist also die dritte von links. Ganz links, bei "Takt", wird eine grüne LED eingesetzt. Die restlichen drei grünen LEDs sind die "Adressen"-LEDs, sie befinden sich ganz rechts. In den restlichen Löchern werden die roten LEDs platziert.

Auch diesmal werden die Anschlüsse leicht abgewinkelt und anschließend wie folgt angelötet: Zuerst werden alle oberen Anschlüsse angelötet und dann erst die unteren. Wir erreichen dadurch, daß die Wärmeaufnahme der LEDs beim Einlöten in Grenzen gehalten wird, denn sie sind sehr hitzeempfindlich.

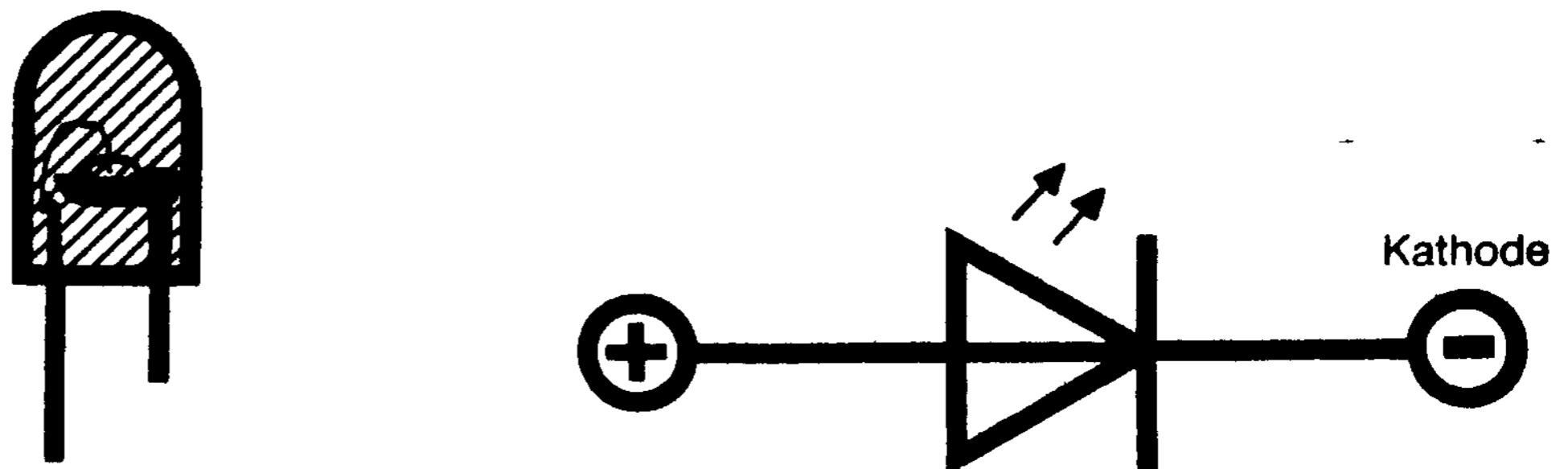


Abb. 31: Gehäuse, Anschlüsse und Polung einer LED

Nun sind weniger wärmeempfindliche Bauteile an der Reihe, die Fassungen für die Integrierten Schaltkreise (ICs) und die Steckerleiste. Beim Einsetzen der IC-Fassungen ist darauf zu achten, daß ihre Markierungszeichen (ein Halbkreis) mit denen des Aufdrucks auf der Platine übereinstimmen, sich also links befinden. Dadurch wird einer späteren Verpolung der ICs vorgebeugt. Vor dem Einlöten vergewissern wir uns, daß alle (!) Anschlüsse durchgesteckt sind. Andernfalls muß der gesamte Sockel wieder ausgelötet werden, und dies ist sehr aufwendig. Dann biegen wir wenigstens zwei diagonal weit auseinanderliegende Anschlüsse jedes Sockels so um, daß der Sockel eng an der Platine anliegt. So verhindern wir ein Durchdrücken der Anschlüsse beim Einsetzen der ICs und ein damit verbundenes Abreißen von Leiterbahnen.

Wenn wir die IC-Sockel-Anschlüsse einlöten, kann es trotz vorhandenem Lötstopplack passieren, daß zwei benachbarte Anschlüsse durch Lötzinn miteinander verbunden werden. Dann erwärmen wir die Lötstelle noch einmal und trennen diese ungewollte Verbindung gleichzeitig mit Hilfe einer angespitzten (!) Bleistiftmine. (Didaktisch-methodische Anmerkung: Ein Bleistift ist ein in der Schule vorhandener üblicher Gegenstand, auch wenn er in den seltensten Fällen angespitzt ist. Es ist nicht zu befürchten, daß die Graphitschicht, die beim Auftrennen der ungewollten Lötbrücke möglicherweise entsteht, beim WDR-1-Bit-Computer im Schulbetrieb zu Störungen führt.) Die ICs werden jetzt noch nicht eingesetzt. Übrig bleibt noch die Steckerleiste. Abb. 32 zeigt den Lötanschluß der Steckerleiste von der Seite.

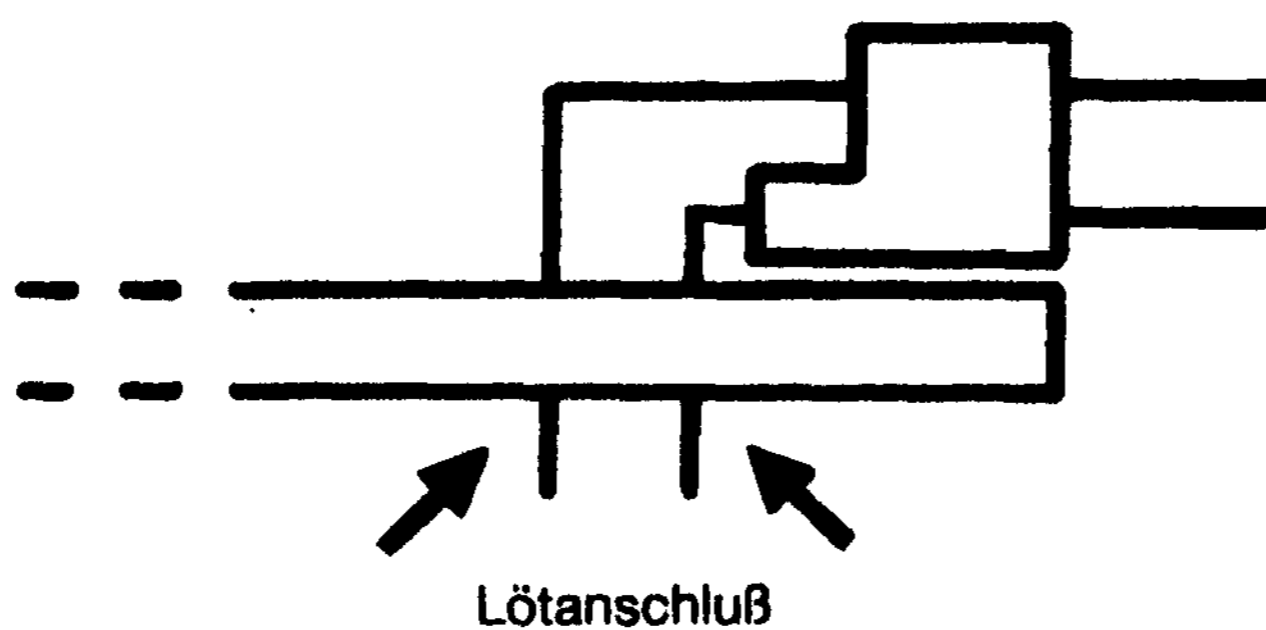


Abb. 32: Lötanschluß der Steckerleiste von der Seite

Nach dem Durchstecken der Lötanschlüsse kontrollieren wir, ob auch wirklich alle Anschlüsse auf der Lötseite zu sehen sind. Das Anlöten der Steckerleiste sollte uns nun leichtfallen. Abb. 33 zeigt die fertige Anzeigeplatine.

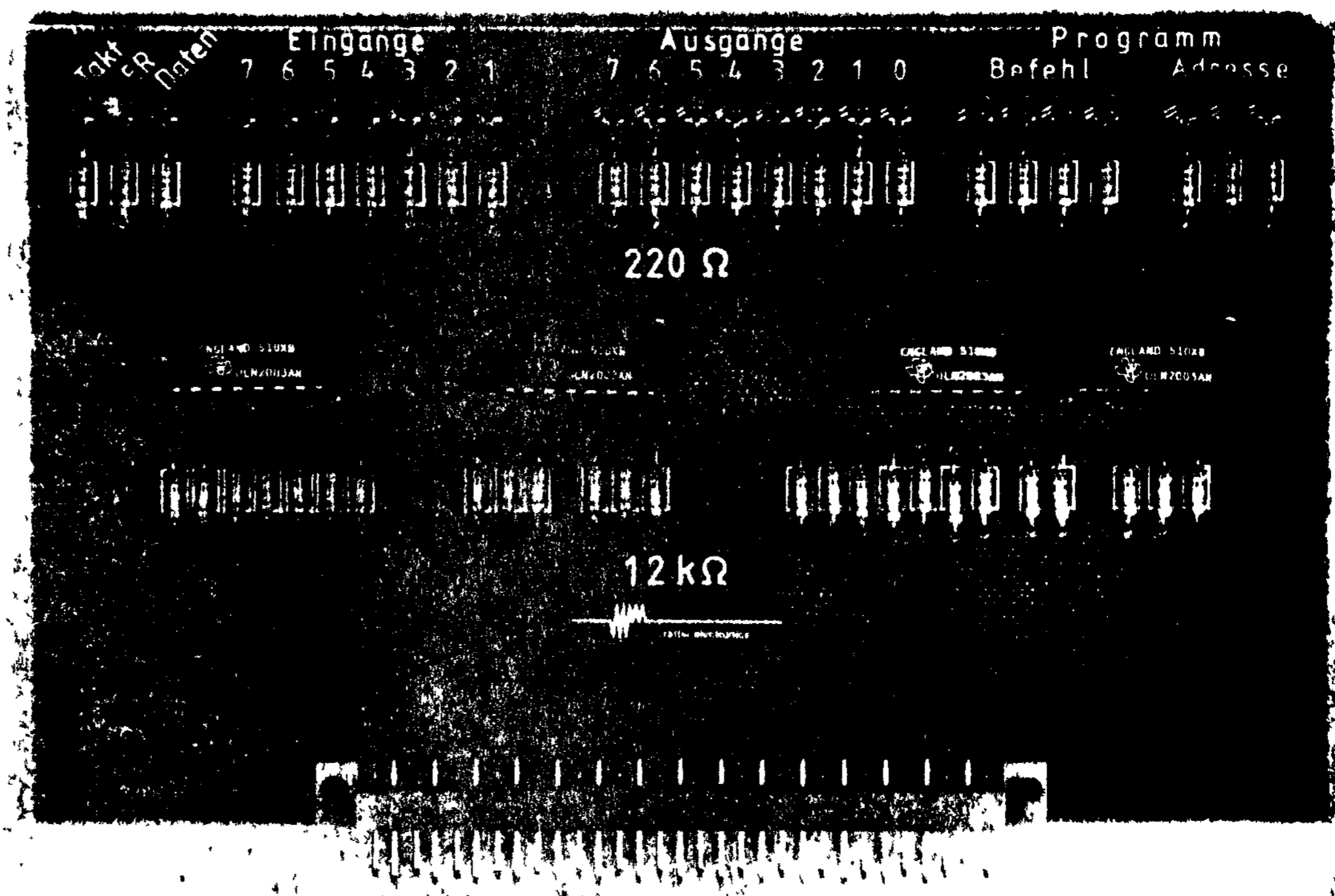


Abb. 33: Fertiggestellte Anzeigeplatine

Fehlerprüfung nach Fertigstellung

Haben wir fehlerfrei gearbeitet? Um dies zu überprüfen, testen wir die fertiggestellte Anzeigeplatine. Dazu benötigen wir eine Spannungsquelle mit 5 Volt Gleichspannung. Eine 4,5-Volt-Flachbatterie kann allerdings genauso verwendet werden. Beim Testen lernen wir gleichzeitig das Schaltbild lesen. Zuerst überprüfen wir die Leuchtdioden: Wir führen den Pluspol unserer Spannungsquelle mit Hilfe eines roten Kabels an einen oberen Anschluß einer beliebigen LED. Den Minuspol der Spannungsquelle dürfen wir auf keinen Fall an den anderen Anschluß der LED legen. Die hier verwendeten LEDs vertragen höchstens einen Strom von je 20 mA. Um sie nicht zu zerstören, müssen wir beim Testen den dazugehörenden Vorwiderstand von 220 Ohm als Strombegrenzer einbeziehen. Wir legen also den Minuspol an den Widerstand an. Um eine Polverwechslung möglichst auszuschließen, verwenden wir hierfür ein schwarzes Kabel. Diese LED muß nun leuchten. Indem wir den Minuspol an sämtlichen Widerständen vorbeiführen, überprüfen wir die Funktionsfähigkeit aller LEDs und Widerstände. Wenn eine LED nicht aufleuchtet, kann dies folgende Ursachen haben:

- Es liegt eine kalte Lötstelle vor. Wir löten nach.
- Die LED ist verpolt. Nach dem Umpolen der Testanordnung müßte sie leuchten. Ist dies der Fall, wird die Leuchtdiode umgelötet.
- Der Widerstand ist falsch. Wir kontrollieren zunächst die Farbringe. Stimmen diese, so messen wir den Widerstand mit dem Vielfachmeßgerät im Ohmbereich. Weicht dieser stark vom erwarteten Wert ab, so wird ein neuer Widerstand mit dem Wert 220 Ohm eingesetzt.
- Die LED ist defekt. Wir wechseln die LED aus.

Wenn alle LEDs ansprechen, suchen wir sie und ihre Vorwiderstände im Schaltbild und markieren sie farbig, z.B. gelb.

Nun können wir die noch fehlenden vier ICs, Treiber-ICs polrichtig einsetzen, d.h., ihre Markierungen müssen mit denen der Fassungen übereinstimmen. Sie haben die Aufgabe, den

geringen Steuerstrom für die LEDs aus den einzelnen Funktionseinheiten des Computers zu verstärken (die LEDs zu treiben). Daher nennt man solche ICs auch Treiber-ICs.

Wir überprüfen sie zusammen mit ihren Vorwiderständen, indem wir die Spannung diesmal an die Steckerleiste führen. Diese Vorwiderstände haben die Aufgabe, den für die Verstärker zu hohen Ansteuerstrom zu begrenzen. Pin 1, das ist der ganz rechte auf der Steckerleiste, ist der Pluspol. Wir markieren ihn am Stecker rot. Pin 2, also der Pin daneben ist der Minuspol. Wir markieren den Stecker an dieser Stelle schwarz. Mit Hilfe der beiden Kabel stellen wir eine Verbindung zwischen Pin 1 sowie Pin 2 und der Spannungsquelle her. Ein zweites rotes Kabel wird mit dem Pluspol verbunden und nacheinander an die restlichen Stifte der Steckerleiste geführt. Nun müssen alle LEDs der Reihe nach aufleuchten. Gibt es eine LED die nicht aufleuchtet? Folgende Fehler können vorliegen:

- Ein IC ist beschädigt. Wir tauschen es mit einem Nachbar-IC aus. Ist der Fehler an dieser Stelle behoben, so muß er nun an einer anderen Stelle auftauchen.
- Der Vorwiderstand ist defekt, hat einen falschen Wert, oder ist fehlerhaft eingelötet.
- Die dazugehörige Leiterbahn ist unterbrochen. Dies überprüfen wir mit dem Ohmmeter.

Leuchten beim Testen eines Treibers zwei oder mehrere LEDs gleichzeitig auf, so suchen wir unter dem IC nach einer Lötbrücke und entfernen sie.

Die Anzeigeplatine ist nun betriebsbereit. Anschließend suchen wir noch die vier ICs ULN 2003 und die dazugehörigen Vorwiderstände im Schaltbild und markieren sie mit der gelben Farbe für die Elemente der Anzeigeplatine.

3.3. DIE GRUNDPLATINE

Der erste Arbeitsschritt für die Fertigstellung der Grundplatine besteht wieder darin, die Löcher von der Lötseite her, dies ist die Seite ohne Bestückungsaufdruck, mit einem 1-mm-Bohrer zu bohren. Die fünf großen Bohrpunkte werden mit einem 5-mm-Bohrer bearbeitet. Für die mittelgroßen Bohrpunkte (neben den Bezeichnungen "HT", "LT" und "ST" sowie "1M") nehmen wir einen 1,5-mm-Bohrer. Die Grundplatine hat auf beiden Seiten Leiterbahnen, die zum Teil elektrisch verbunden werden müssen. Wir setzen hierfür an den mit weißen Kreisen markierten Stellen Durchkontaktierungen ein. Dazu benutzen wir z.B. die beim Aufbau der Anzeigeplatine abgekniffenen Drahtenden der Widerstände. Das Schema einer Durchkontaktierung zeigt Abb. 34.

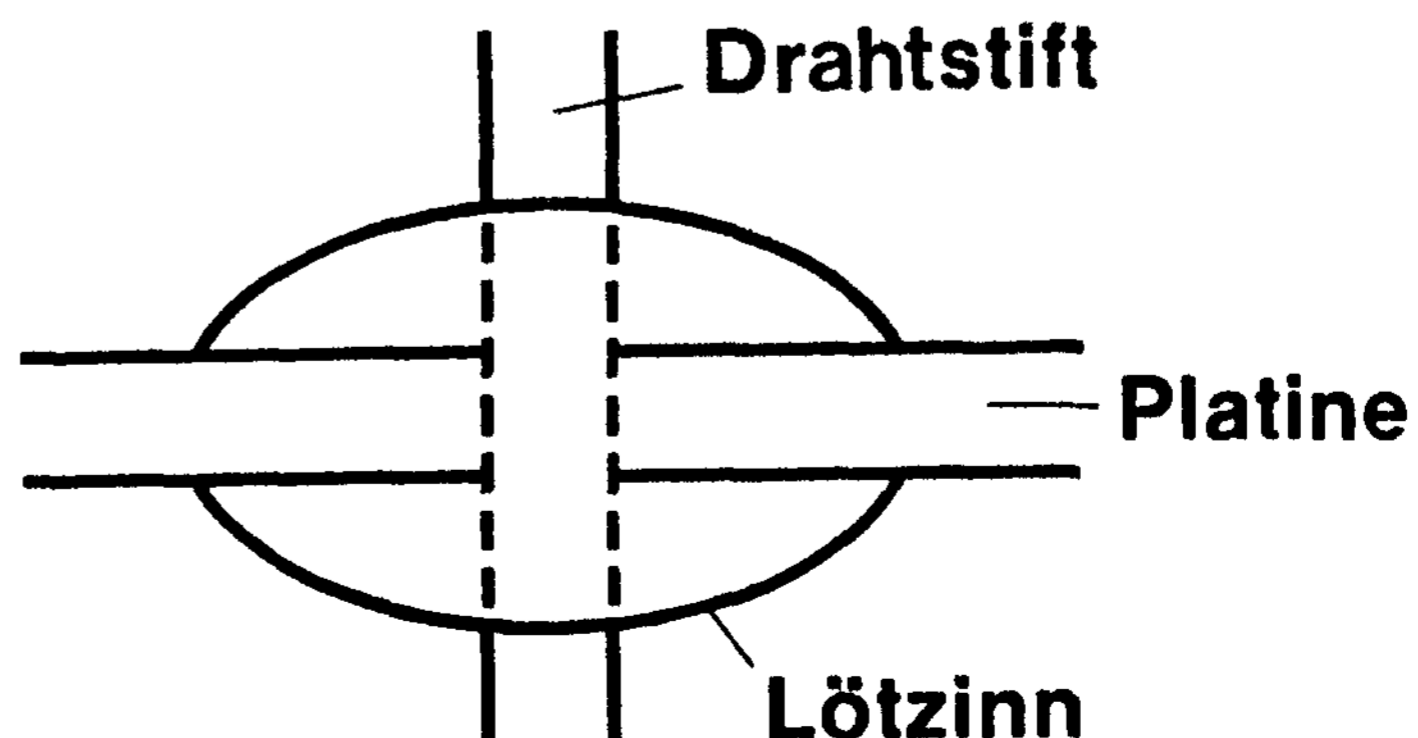


Abb. 34: Schema einer Durchkontaktierung

Damit die Durchkontaktierungen auf der Platineunterseite genügend lang herausragt, legen wir unter die Platine einige Geldstücke. Wir löten so alle Drahtstifte auf der Bestückungsseite fest, biegen sie leicht um und wenden die Platine. Jetzt können wir die Durchkontaktierungen auch auf der Lötseite zu Ende führen, ohne daß die Stifte herausfallen oder herausgezogen werden. Überstehende Drahtenden der Durchkontaktierungen werden danach abgeschnitten. Einige der Buchsenleisten müssen vor dem Einlöten an ihren Enden gekürzt werden: eine 13-polige und die beiden 31-poligen Buchsenleisten sowie die 21-polige Steckerleiste. Eine Pinzette hilft, die vielen Anschlüsse einer Buchsenleiste in die vielen Löcher zu führen, manchmal stehen sie je nach vorheriger Lagerung etwas seitlich ab. Beim Einlöten der Steckerleiste sowie einiger anderer Bauteile müssen wir darauf achten, daß einige Anschlüsse zusätzlich auch auf der Bestückungsseite angelötet werden müssen. Dies erkennen wir an den kupfernen Lötäugen. Jetzt werden die 3,3-Kiloohm-Widerstände (Farbcodierung: "orange-orange-rot-gold") und der 100-Kiloohm-Widerstand (Farbcodierung: "braun-schwarz-gelb-rot") eingesetzt und verlötet. Es folgen die Kondensatoren. Sie müssen so eingesetzt werden, daß ihre aufgedruckten Polmarkierungen dem Bestückungsaufdruck der Platine entsprechen. Wenn wir die Sockel, Schalter und Taster einsetzen, achten wir wieder darauf, daß auch diese Bauteile wieder eng an der Platine anliegen. Jeder einzelne Taster muß so eingesetzt werden, daß der "Drehpunkt" des Tasters hinten liegt, der Taster sich also damit vorne absenken läßt. Auch den Spannungskonstanter 7805 und die Diode müssen wir polrichtig einlöten (Siehe Abb. 35).

+ Bauteile

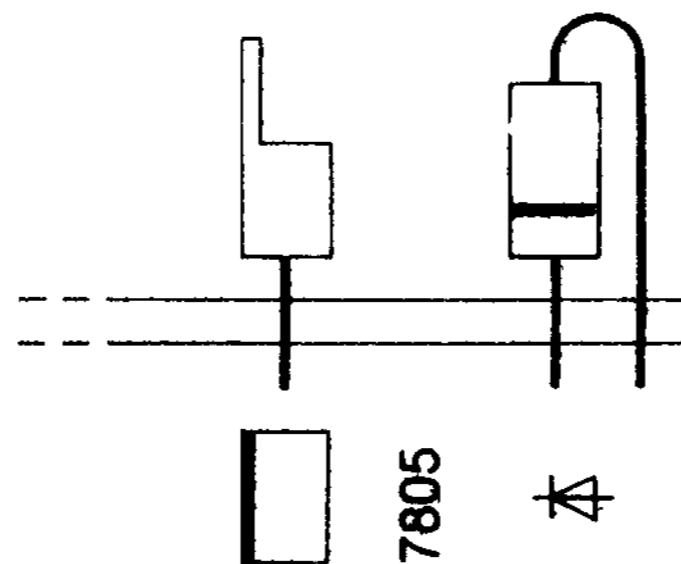


Abb. 35: Spannungskonstanter und Diode

Diese Abbildung zeigt auch, daß die Diode stehend eingesetzt werden sollte. Es fehlen noch das Einlöten des 1-Megaohm-Potentiometers und das Einsetzen der 5 Gummifüße in die 5-mm-Löcher. Die Platine ist fertig aufgebaut, wenn kein kupfernes Lötauge auf beiden Seiten der Platine mehr zu sehen ist.

Abb. 36 zeigt die fertige Grundplatine.

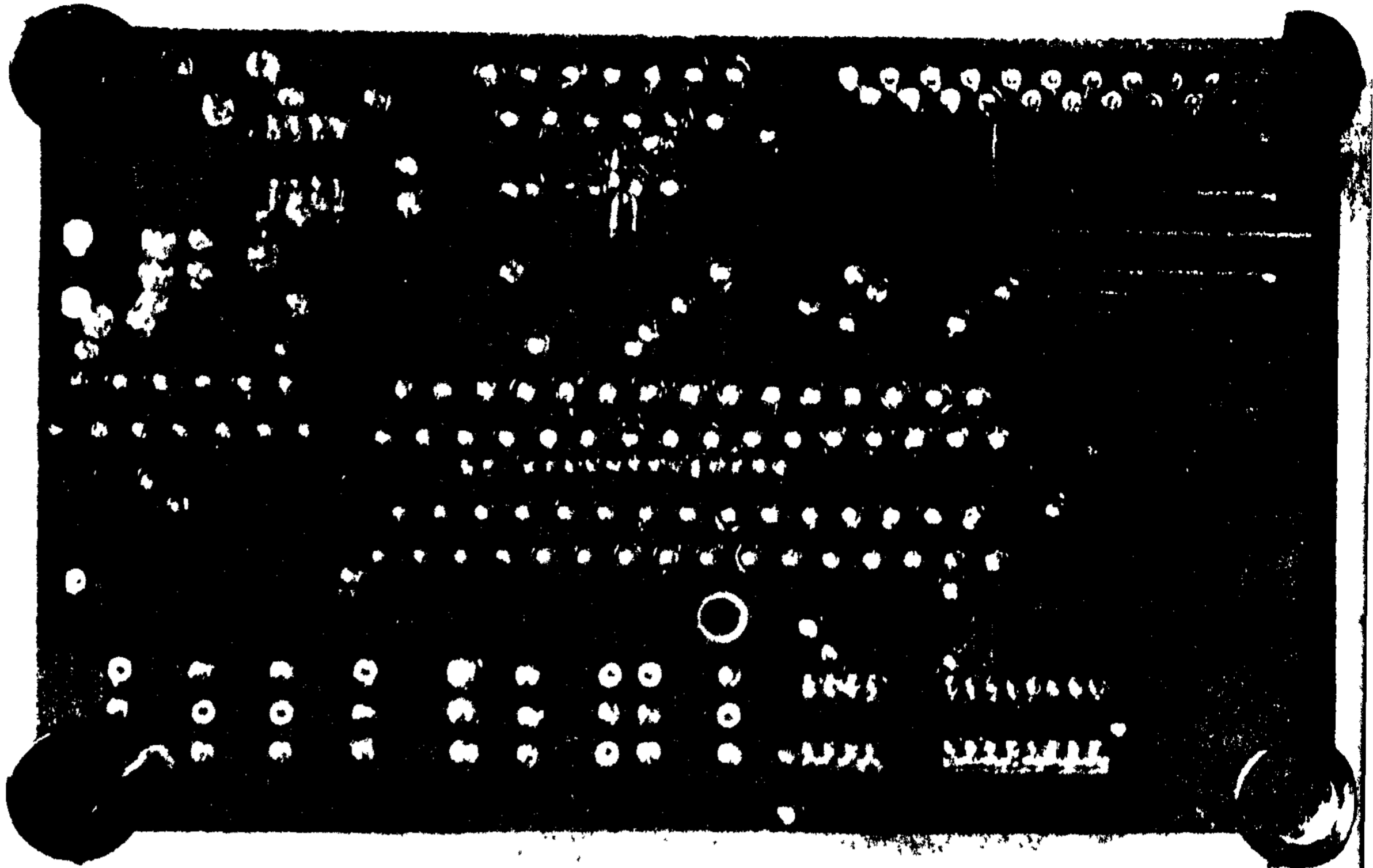


Abb. 36: Fertiggestellte Grundplatine von der Rückseite

Haben wir auch diesmal fehlerfrei gearbeitet? Um dies herauszufinden, legen wir an den Eingang des Spannungskonstanters (+ und - Aufdruck der Platine in der Nähe der Diode) eine Wechselspannung von etwa 9 Volt. An der Kathode der Diode liegen jetzt ca. 12 Volt Gleichspannung an (Siehe Kap. 2.2.9.). Dies überprüfen wir mit unserem Vielfachmeßgerät. Messen wir nur eine Spannung von etwa 4 Volt, so haben wir vergessen, den 1000-Mikrofarad-Kondensator auf der Bestückungsseite anzulöten. Ist jedoch alles in Ordnung, so können wir statt der Wechselspannung auch eine Gleichspannung von etwa 9 Volt an den Eingang der Stromversorgung anlegen. In diesem Falle arbeitet die Diode als Verpolungsschutz. Jedes handelsübliche Steckernetzgerät für die Versorgung von Transistorgeräten mit einem max. Strom von mindestens 350 mA und einer Spannung von ca. 9 Volt eignet sich für unseren Computer.

Nun überprüfen wir die Ausgangsspannung der Stromversorgung an den Buchsenleisten: An Buchse 1 jeder Buchsenleiste liegt jeweils die 5-Volt-Spannung, an Buchse 2 aller 31poligen und der alleinstehenden 13-poligen Buchsenleisten liegt die Masse. Dabei ist zu beachten, daß die alleinstehende 13polige Buchsenleiste umgekehrt eingelötet worden ist.

Folgende Fehler können vorliegen, wenn wir nicht 5 Volt an diesen Meßpunkten nachweisen können:

- Der Spannungskonstanter arbeitet nicht, weil er beispielsweise defekt ist oder falsch eingelötet wurde.
- Der 1- μ F-Kondensator ist verpolt worden.

Im Schaltbild suchen wir die Elemente der Stromversorgung (links unten) und markieren sie rosa. Nun können wir unbesorgt den Timer NE 555 (das Taktgeber-IC) in seine Fassung setzen und seine Wirkung im Verbund mit dem 1- μ F-Kondensator, dem 22-Kilohm-Widerstand und dem 1-Megaohm-Potentiometer testen. Dazu greifen wir am Ausgang Q des Taktgeber-ICs, dies ist sein

Anschluß 3 mit der Verbindung zu dem 100-Kiloohm-Widerstand, die Ausgangsspannung ab. Je nach Stellung des Potentiometers kann der Zeiger dem Takt folgen.

Der Taktgeber arbeitet nicht?

- Der Kondensator ist verpolt.
- Das IC ist falsch eingesetzt.

Am Anschluß 4 oder am Anschluß 8 des IC's liegt keine Betriebsspannung.

Die Elemente des Timers markieren wir im Schaltbild grün. Auf die fertige Grundplatine können wir nun die Anzeigeplatine stecken. Dabei zeigt sich, ob diese beiden Platinen schon zusammenspielen. Wir können die Eingangsleuchtdioden 1 bis 4 ein- bzw. ausschalten, indem wir die vier Eingangsschalter, die in dem 8poligen Sockel stecken, betätigen.

3.4. DIE SPEICHERPLATINE

Die zweiseitig geätzte Speicherplatine wird wieder von der Lötseite her mit einem 1-mm-Bohrer bearbeitet. Als erstes werden die Durchkontaktierungen gesetzt und verlötet. Die überstehenden Drahtenden kneifen wir wieder ab, außer dem Stift, der sich rechts unter dem Testpunkt "T" befindet. Jetzt setzen wir die 22-Kiloohm-Widerstände ("rot-rot-orange-gold") ein und verlöten sie. Es folgen die Sockel für die ICs und die Steckerleiste. Die ICs setzen wir erst ein, wenn wir die Platine auf Kurzschlüsse hin überprüft haben. Dazu stecken wir sie hinten auf die Grundplatine (vgl. Abb.37).

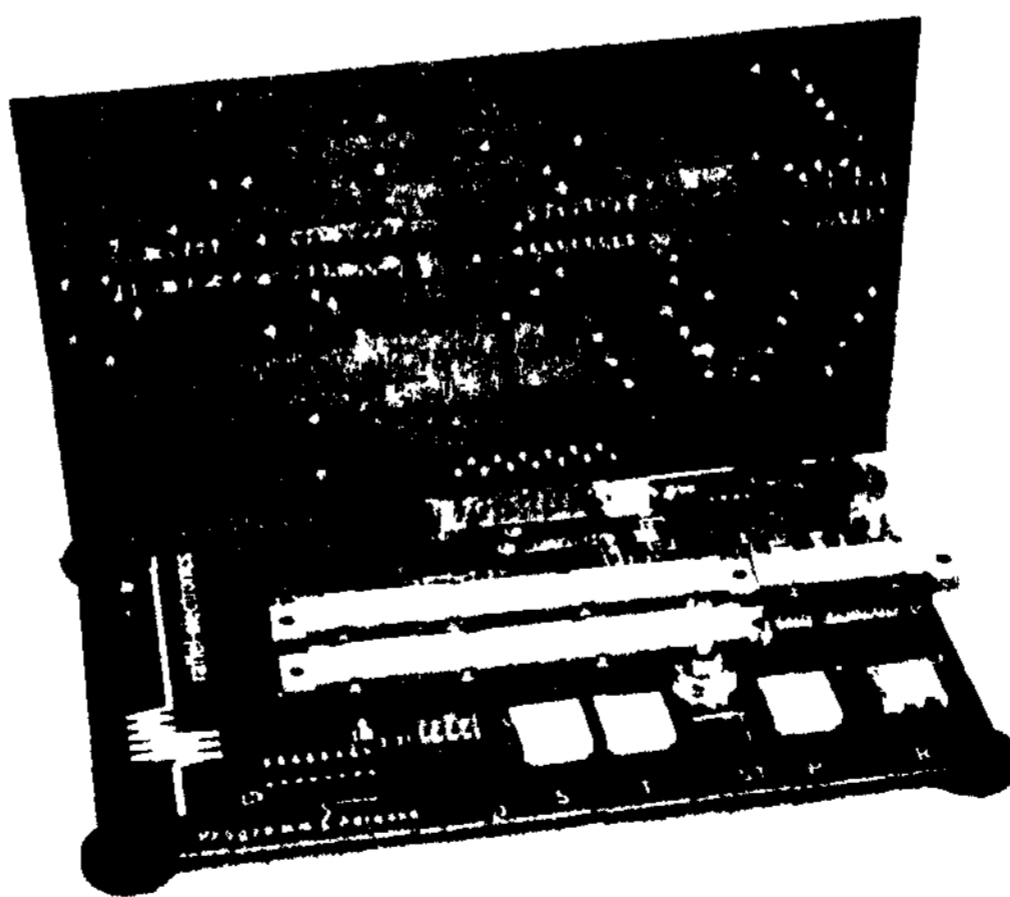


Abb. 37: Grundplatine mit aufgesteckter Speicherplatine

Vorne stecken wir die Anzeigeplatine dazu. Leuchten die vier Eingangs-LEDs wieder, so hat die Speicherplatine keinen Kurzschluß zwischen der 5 V-Leitung und der Masse. Liegt jedoch

ein Kurzschluß vor, so hilft das Vielfachmeßgerät als Ohmmeter bei der Kurzschlußsuche. Wir nehmen die Speicherplatine wieder heraus und setzen die beiden Speicher-ICs, die 2112, gemäß dem Bestückungsaufdruck in ihre Fassungen. Nach dem erneuten Einsetzen in die Grundplatine zeigen die 7 Programm-LEDs der Anzeigeplatine nun zufällige Werte an.

Um einen bestimmten Programminhalt anzuzeigen, legen wir die Information "0000 000" an die 7 Speichereingänge. Dies geschieht mit Hilfe von 7 Schaltern im 16poligen Sockel der Grundplatine. Der vierte Schalter von rechts hat keine Bedeutung in unserem System.

Im Schaltbild kennzeichnen wir die beiden Speicher-ICs und die 7 Schalter (vier Befehls- und drei Adreßschalter) mit einem roten Stift. Aus dem Schaltbild geht hervor, daß an den Eingängen der Speicher, das sind die Anschlüsse 9 bis 11 bzw. 9 bis 12, 0 Volt anliegen, wenn die Schalter geschlossen sind. Wir drücken nun auf den Schreibtaster "S" der Grundplatine. Dadurch wird die eingestellte Information "0000 000" in den Speicher übernommen. Die 7 Programm-LEDs auf der Anzeigeplatine müssen nun erloschen sein. Ist dies nicht der Fall, so sind folgende Fehler denkbar:

- Die Schalter wurden nicht geschlossen. Das kann mit dem Ohmmeter überprüft werden.
- Die Speicher ICs sind verpolt.
- Die Speicher-ICs sind defekt.
- Der Schreibtaster arbeitet nicht. Auch hier hilft das Ohmmeter bei der Fehlersuche.

Wir überprüfen, ob unsere beiden Speicher auch die Information "1111 111" behalten können. Dazu öffnen wir die 7 Schalter und betätigen wieder den Schreibtaster. Jetzt müssen alle 7 Programm-LEDs leuchten. Ist dies nicht der Fall, so sind folgende Fehler möglich:

- Ein Speicher-IC ist defekt. Ein Tausch der Speicher-ICs bestätigt die Vermutung.
- Die Zuleitung von den Speichern zu der Anzeigeplatine ist unterbrochen.
- Der Widerstand am entsprechenden Speichereingang ist nicht wirksam. Wir überprüfen die Lötstelle bzw. seinen Wert. Das Aufsuchen dieses 22-Kiloohm-Widerstandes im Schaltbild verdeutlicht, daß dieser den Speichereingang immer dann auf "1" setzt, wenn der dazugehörige Schalter geöffnet ist.

Unter den Speicher-ICs mit den dazugehörigen Widerständen erkennen wir im Schaltbild die beiden Zähler-ICs CD 4029. Die Anschlüsse 15 sind zusammengeschaltet. Dies sind die Takteingänge für die Zähler. Wir markieren im Schaltbild die beiden ICs ebenfalls rot. Diese beiden ICs setzen wir in die Fassungen der Speicherplatine ein und testen diese nun zu Ende. Wir verbinden den Ausgang des Timers (Anschluß 3) auf der Grundplatine mit dem Testpunkt "T" der Speicherplatine. Er ist der genannte Takteingang für die Zähler. Je nach Taktgeschwindigkeit des Taktgebers können wir das wechselnde Aufleuchten der Programmleuchtdioden verfolgen. Ist dies nicht der Fall, so sind folgende Fehler möglich:

- Das Taktgeberpotentiometer ist zu schnell eingestellt.
- Die Zähler-ICs sind verpolt.
- Die Zähler-ICs sind defekt.
- Die Stromversorgung an den Sockeln ist unvollständig. Hier kann man nachmessen.
- Der Speicherinhalt ist immer der gleiche. Diesen können wir inzwischen verändern.

Die Speicherplatine ist damit fertig gebaut.

3.5. DIE PROZESSORPLATINE

Abbildung 38 zeigt die Bestückungsseite der Prozessorplatine.

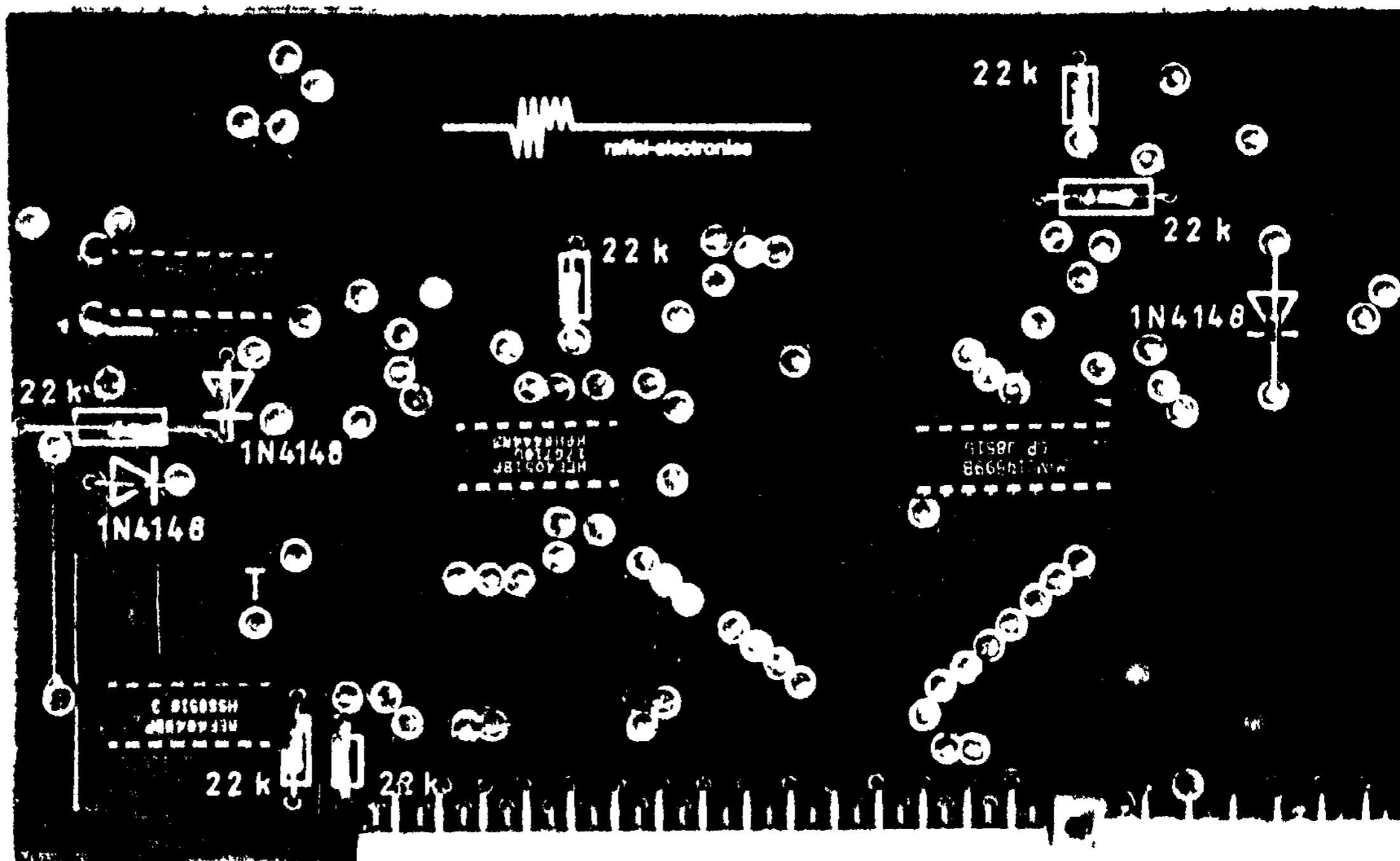


Abb. 38: Bestückungsseite der Prozessorplatine

Auch sie ist wieder von beiden Seiten geätzt. Daraus folgt, daß wir sie wieder durchkontaktieren müssen. Danach löten wir die 22-Kiloohm-Widerstände ein. Bei den drei Dioden müssen wir vor dem Einlöten wieder auf das polrichtige Einsetzen achten. Wir schlagen vor, die IC-Sockel in einer bestimmten Reihenfolge einzusetzen: zuerst den 18poligen und dann die 16poligen. So verhindern wir, daß z.B. ein 16poliger Sockel an den Platz des 18poligen versehentlich gerät.

Die Steckerleisten müssen vor dem Einsetzen auf das richtige Maß gekürzt werden. Die fertig aufgebaute Prozessorplatine stecken wir zunächst noch ohne ICs zusammen mit den anderen beiden Platinen in unsere Grundplatine. Verändert sich die Helligkeit der Eingangs-LEDs nicht, so liegt kein Kurzschluß vor. Die ICs setzen wir in einer ganz bestimmten Reihenfolge ein, und zwar das IC CD 4049 als erstes. Es enthält 6 Signalumkehrer, sogenannte Inverter, von denen allerdings nur zwei benötigt werden. Das Betätigen eines Tasters kann viele Impulse verursachen, wenn der Taster prellt. Wir benötigen für einen Taktschritt jedoch nur einen Einzelimpuls. Die mit Hilfe der beiden Inverter aufgebaute Schaltung entprellt den Takttaster "T" auf der Grundplatine, der beim Betätigen so nur einen definierten Impuls erzeugt. Im Schaltbild markieren wir die beiden Inverter (sie liegen neben der Stromversorgung) blau. Die Entprellschaltung funktioniert so: Die Tasterstellung im Schaltbild legt an den Eingang (Pin 3) des Inverters eine "1". Das invertierte Signal erzeugt am Ausgang (Pin 2) eine "0". Somit liegt über dem 22-Kiloohm-Widerstand am Eingang des anderen Inverters (Pin 14) eine "0". Diese erscheint invertiert

als "1" am Ausgang dieses Inverters (Pin 13) und somit als Einzelimpuls. Betätigen wir den Taster, so kehren sich sämtliche Zustände um. Um zu erkennen, ob die Handtaktschaltung richtig arbeitet, verbinden wir den Testpunkt "T" der Speicherplatine mit dem Testpunkt "T" der Prozessorplatine. Dann überprüfen wir den Taster mehrmals. Dabei leuchtet die Takt-LED im Rhythmus des Taktes, den wir mit der Hand erzeugen: die LED ist aus, wenn der Taster gedrückt ist, sie leuchtet, wenn wir den Taster loslassen. *an Tisch!* Geschieht dies nicht?

- Der Taster arbeitet nicht, weil er schlechte Kontakte hat. Hier hilft das Ohmmeter.

- Das IC ist defekt!

- Die Leiterbahn zwischen dem Testpunkt "T" der Prozessorplatine und dem Anschluß 31 der Buchsenleiste für die Anzeigeplatine ist unterbrochen.

Wir können nun die Leitung zwischen den beiden Testpunkten entfernen. Danach setzen wir das IC MC 14500, ein und testen seinen internen Taktgenerator (Siehe Abb. 5). Gleichzeitig überprüfen wir dabei die Funktion des Taktwahlschalters. Die Takt-LED leuchtet, wenn der Taktwahlschalter auf Handtakt "HT" gestellt wird. Betätigen wir den Takttaster, erlischt sie. Wir bringen jetzt den Taktwahlschalter in die mittlere Stellung, auf Langsamtakt "LT". Je nach Stellung des Potentiometers für den NE 555 erkennen wir das Aufleuchten der Takt-LED. In der Stellung Schnelltakt "ST" wirkt der prozessorinterne Taktgenerator. Dann leuchtet die LED ca. 1 Million mal in der Sekunde auf. Für das menschliche Auge hat dies den Anschein, als ob die LED dauernd leuchte. Die Takt-LED leuchtet allerdings nicht mehr so hell.

Ersetzen wir den 100 Kiloohm-Widerstand an Pin 3 des NE 555 durch einen mit dem Wert von 560 Kiloohm, so verringern wir den Einfluß des Langsamtaktes auf den Schnelltakt.

Wenden wir uns dem Schaltbild zu. Dort markieren wir das IC 14500 blau. Es gibt dort nur noch zwei weiße Flecke. Es ist der noch fehlende Eingabebaustein und der Ausgabebaustein, die ICs 4051 und 14599. Wir markieren beide ICs im Schaltbild blau und setzen sie in ihre Sockel auf der Prozessorplatine ein. Ob sie, und damit unser nun fertig aufgebauter Computer, funktionieren, können wir ohne Programm nicht mehr testen. Dies wird in Kap. 4 geschehen.

3.6. DIE TASTATUR

Die Tastatur ist kein Luxus. Wir können unseren Computer auch ohne sie aus methodischen Gründen mit Hilfe der Programmschalter programmieren. Dies ist für den Anfang erwünscht, damit Lernende den jeweiligen Programmschritt, sein zugeordnetes Bitmuster bei den Eingabeschaltern und die dazugehörige Leuchtdiodenanzeige interpretieren können. Später, wenn dies beherrscht wird, ist diese Eingabe jedoch sehr umständlich. Abb. 39 zeigt die Tastaturplatine.

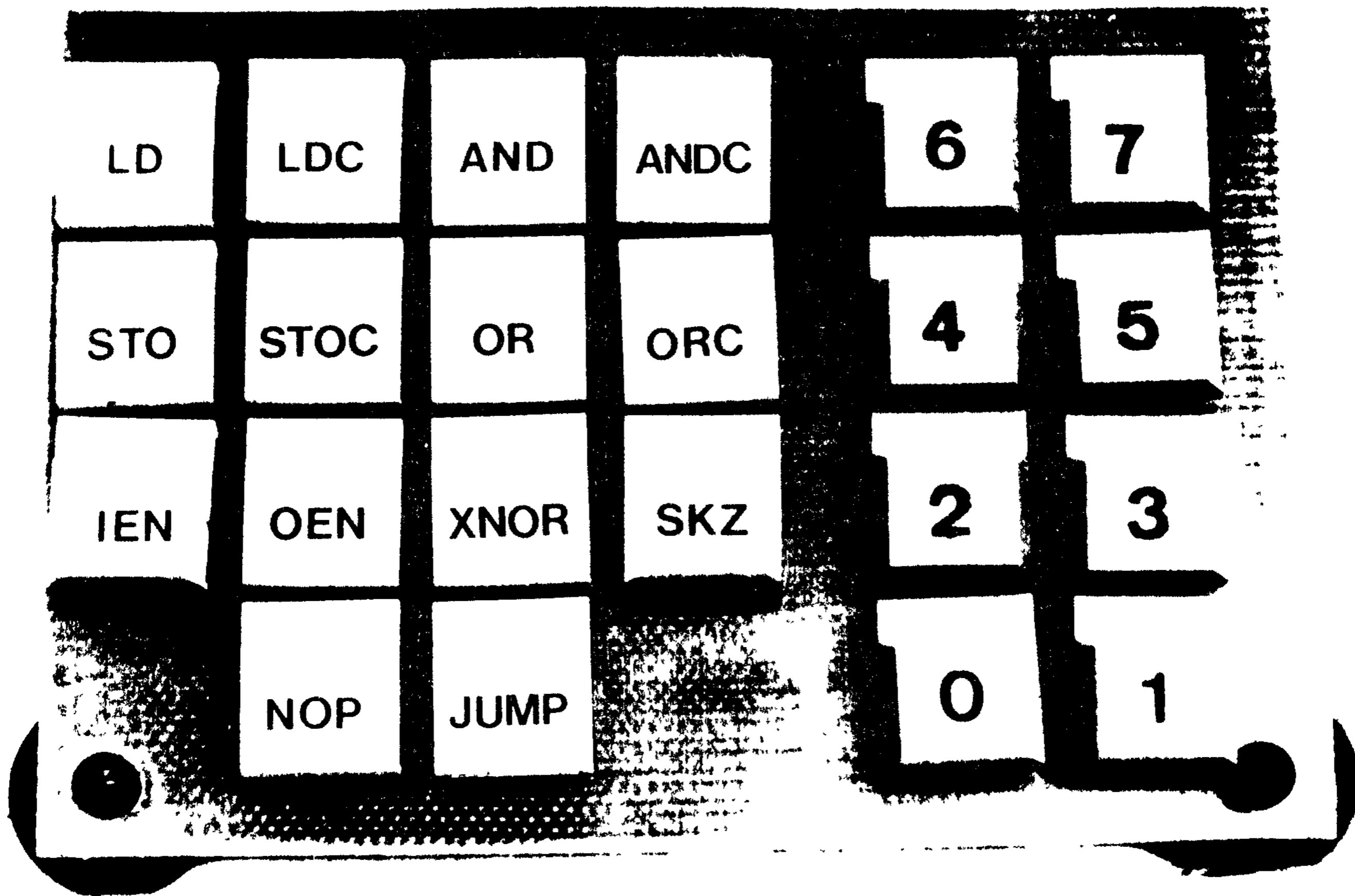


Abb. 39: Die Tastatur

Zum Bohren benötigen wir für die Füße den 5-mm Bohrer, für alle anderen Teile den 1-mm-Bohrer. Der Bestückungsaufdruck zeigt (aus Platzgründen) die Diodenschaltzeichen nur unvollständig. Die kurzen Querstriche entsprechen den Kathoden. Nach dem Einlöten der Dioden werden die beiden Drahtbrücken gesetzt. Bevor wir die Taster einsetzen, müssen wir ihre zwei Plastikstifte auf der Unterseite entfernen. Wir setzen die Taster so ein, daß ihre "Drehpunkte" in Richtung Dioden zeigen. Den Anschluß der fertig aufgebauten Tastatur an den WDR-1-Bit-Computer zeigt die Abb. 40.

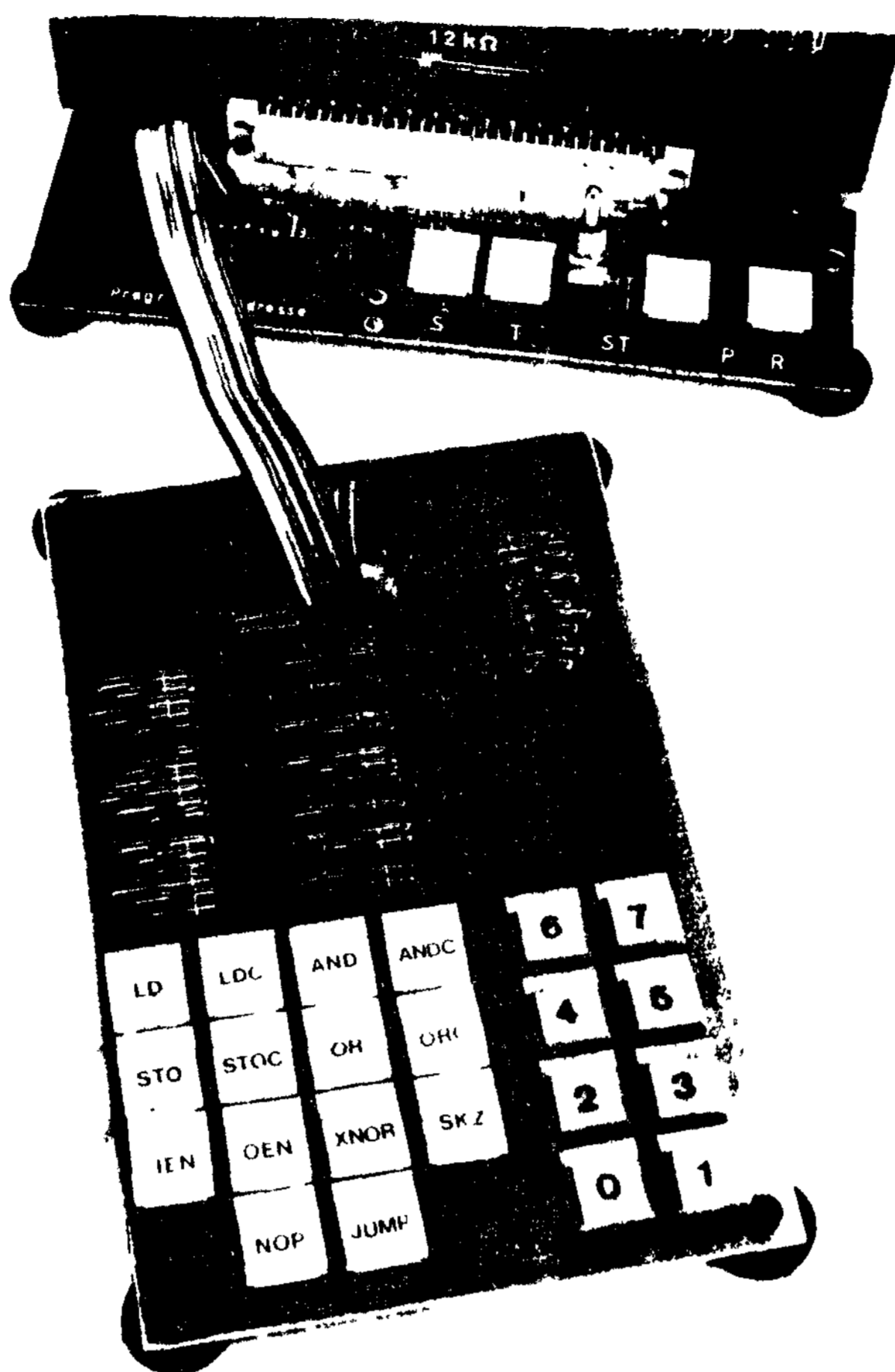


Abb. 40: Anschluß der DATANorf-Tastatur an den WDR-1-Bit-Computer.

Vor dem Einlöten des Flachbandkabels in die Tastatur bzw. den mitgelieferten IC-Sockel müssen ihre Enden kurz (!) abisoliert werden. Zuerst löten wir das Kabel in die Tastatur ein. Die Reihenfolge der Farbe beim Flachbandkabel spielt keine Rolle. Die linke Ader des Kabels stellt nach dem Einlöten die Masseleitung dar, die anderen Adern sind die vier Programm- und drei Adreßleitungen in der Reihenfolge von links nach rechts. An das andere Ende des Flachbandkabels löten wir einen IC-Sockel nach folgendem Gesichtspunkt ein: Lötunkte sind die Fassungsunkte oben auf dem Sockel, also nicht die unteren Sockelanschlüsse. Der Masseleitung des Kabels gehört in die untere Reihe, am besten nach links unten. Die anderen Adern löten wir in der oben angegebenen Reihenfolge links oben beginnend an. Wenn wir das Kabel auf der Tastatur noch zusätzlich befestigen, kann es sich nicht mehr so leicht lösen und an den Lötstellen abreißen.

4. DIE PROGRAMMIERUNG DES WDR-1-BIT-COMPUTERS: DIE SOFTWARE

Nun ist es soweit, nachdem wir unseren Computer - hoffentlich erfolgreich - aufgebaut und getestet haben, werden wir ihn jetzt programmieren. Die einfacheren Programme simulieren logische Verknüpfungen und deren Anwendung, z.B. Flipflops. Andere Programme führen von einer Motorsteuerung bis hin zu komplexen Robotersteuerungen. Dabei zeigt sich der WDR-1-Bit-Computer von seiner starken Seite: Seine Programme sind einfach und überschaubar, weil sein Befehlssatz überblickbar bleibt. Er demonstriert das Wesen der industriellen Revolution durch Mikroprozessoren. In Maschinensprache geschriebene Programme zeigen Lernenden in elementarer Form, wie Roboter und andere Maschinen steuerbar sind.

In diesem Abschnitt wollen wir die Programmierung des WDR-1-Bit-Computers behandeln. Der englische Fachausdruck dafür ist "Software". Zunächst muß aber der Befehlssatz der CPU näher kennengelernt werden. Dies geschieht in Kapitel 4.1.

4.1. DER BEFEHLSSATZ DES MIKROPROZESSORS

Befehlscode	Name	CPU-Tätigkeit
0000	NOP	No OPERATION - Keine Operation Die Kontrolleinheit gibt nur einen 5 V-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. Dieser Befehl heißt eigentlich NOPO. Da aber der Befehl NOPF in diesem System nicht existiert, soll der Befehl NOPO zur Vereinfachung NOP genannt werden.
0001	LD	Load - Laden Die CPU holt eine Information über die Datenleitung und speichert sie, wenn die Eingabe freigegeben ist, im Ergebnisregister. Ist die Eingabe nicht freigegeben, so wird dort eine "0" gespeichert.
0010	LDC	Load Complement - Komplement laden Die CPU holt eine Information über die Datenleitung und speichert das Komplement, wenn die Eingabe freigegeben ist, im Ergebnisregister. Ist die Eingabe nicht freigegeben, so wird dort eine "1" gespeichert.
0011	AND	UND-Funktion Es wird die UND-Funktion gebildet aus dem Inhalt des Ergebnisregisters und der Information auf der Datenleitung. Das Ergebnis steht dann im Ergebnisregister. Hier steht nur dann eine "1", wenn vorher im Ergebnisregister und auf der Datenleitung eine "1" gestanden hat und die Eingabe freigegeben ist.
0100	ANDC	UND-Funktion mit dem Komplement der Datenleitung Es wird die UND-Funktion gebildet aus dem Inhalt des Ergebnisregisters und dem Gegenteil der Information auf der Datenleitung. Ins Ergebnisregister kommt nur dann eine "1", wenn dort vorher eine "1" und auf der Datenleitung eine "0" gestanden hat, oder die Eingabe gesperrt ist.

0101	OR	<p>ODER-Funktion</p> <p>Es wird die ODER-Funktion gebildet aus dem Inhalt des Ergebnisregisters und der Information auf der Datenleitung. Ins Ergebnisregister kommt nur dann eine "0", wenn dort vorher eine "0" und auf der Datenleitung eine "0" gestanden hat oder die Eingabe gesperrt ist.</p>
0110	ORC	<p>ODER-Funktion mit dem Komplement der Datenleitung</p> <p>Es wird die ODER-Funktion gebildet aus dem Inhalt des Ergebnisregisters und dem Gegenteil der Information auf der Datenleitung. Ins Ergebnisregister kommt nur dann eine "0", wenn dort vorher eine "0" und auf der Datenleitung eine "1" gestanden hat und die Eingabe freigegeben ist.</p>
0111	XNOR	<p>eXclusive NOR - Äquivalenzfunktion</p> <p>Es wird die Äquivalenzfunktion gebildet aus dem Inhalt des Ergebnisregisters und der Information auf der Datenleitung. Ins Ergebnisregister kommt nur dann eine "1", wenn dort vorher das Gleiche gestanden hat wie auf der Datenleitung, oder wenn im Ergebnisregister eine "0" stand und die Eingabe gesperrt ist.</p>
1000	STO	<p>STOre - Speichern</p> <p>Die CPU gibt den Inhalt des Ergebnisregisters über die Datenleitung aus, wenn die Ausgabe freigegeben ist. Ist die Ausgabe nicht freigegeben, so wird nichts ausgegeben. In keinem Fall ändert sich der Inhalt der Ergebnisregisters.</p>
1001	STOC	<p>STOre Complement - Komplement speichern</p> <p>Die CPU gibt das Gegenteil des Inhaltes des Ergebnisregisters über die Datenleitung aus, wenn die Ausgabe freigegeben ist. Auch hier bleibt der Inhalt des Ergebnisregisters erhalten.</p>
1010	IEN	<p>Input ENable - Eingabe freimachen</p> <p>Die CPU holt eine Information über die Datenleitung und speichert sie im IEN-Register. (Siehe: Kapitel 2.2.1)</p> <p>Der Inhalt des Ergebnisregisters ändert sich nicht.</p>
1011	OEN	<p>Output ENable - Ausgabe freimachen</p> <p>Die CPU holt eine Information über die Datenleitung und speichert sie im OEN-Register. (Siehe: Kapitel 2.2.1)</p> <p>Der Inhalt des Ergebnisregisters bleibt dabei erhalten.</p>
1100	JMP	<p>JuMP - Springen</p> <p>Die Kontrolleinheit der CPU gibt einen 5 V-Impuls auf Pin 12 ab. Dieser Impuls setzt den Programmzähler auf die Programmadresse 0, denn Pin 12 ist über eine Diode mit Pin 1 des Programmzählers verbunden. Die Diode verhindert, daß beim Betätigen des Programmanfangs-Tasters der JMP-Ausgang der CPU zerstört wird.</p>
1101	RTN	<p>Dieser Befehl wird hier nicht benutzt. Wäre er vorgesehen, würde die Kontrolleinheit der CPU einen 5V-Impuls auf den hier nicht beschalteten PIN 11 abgeben. Außerdem würde der nächste Befehl übersprungen werden zum übernächsten.</p>

1110	SKZ	SKip if Zero - Überspringen, wenn Null. Falls im Ergebnisregister eine "1" steht, führt die CPU den nächsten Befehl normal aus. Falls dort eine "0" steht, überspringt die CPU den nächsten Befehl und geht zum übernächsten weiter.
1111	NOPF	No OPeration Function - keine Operationsfunktion Dieser Befehl ist hier auch nicht vorgesehen. Er würde einen 5V-Impuls von der CPU an den hier nicht beschalteten PIN 9 hervorrufen.

Besonderheiten:

Die Steuerbefehle NOP, JMP und der Programmverzweigungsbefehl SKZ benötigen keine Ein-/Ausgabeadresse, da sie keine Information der Datenleitung brauchen und auch keine Information über sie ausgehen. Anstatt einer bestimmten Ein-/Ausgabeadresse wird in diesem Abschnitt in Programmen hinter diesen Befehl ein x geschrieben. Dieses x kann also 0, 1, 2, 3, 4, 5, 6, oder 7 heißen. Außerdem bleibt auch bei diesen Befehl der Inhalt des Ergebnisregisters erhalten. Die Eingangsleitung Nr. 0 (E0) ist mit dem Ergebnisregister verbunden (Siehe: Abb. 2). Die Ein-/Ausgabeadresse 0 stellt in Zusammenhang mit den Eingabebefehlen und den Rechenbefehlen eine weitere Besonderheit dar. Der Programmschritt LD 0 ist ein überflüssiger Befehl. Er hat folgende Wirkung: Der Inhalt des Ergebnisregisters wird in das Ergebnisregister gebracht. Der Programmschritt LDC 0 invertiert immer den Inhalt des Ergebnisregisters, d.h., er bildet das Komplement. Bei den Rechenbefehlen wird der Inhalt des Ergebnisregisters mit sich selbst verknüpft. Bei den Programmschritten ORC 0 und XNOR 0 gelangt immer eine "1" in das Ergebnisregister. Beim Programmschritt ANDC 0 gelangt immer eine "0" in das Ergebnisregister. Bei den Programmschritten AND 0 und OR 0 wird der Inhalt des Ergebnisregisters nicht verändert. Voraussetzung hierbei ist, daß die Eingabe freigegeben ist.

4.2. DAS EINGEBEN VON PROGRAMMEN ÜBER DIL-SCHALTER

Wir geben jeden einzelnen Programmschritt ein, indem wir die vier Befehlsschalter und die drei Adressenschalter betätigen, entsprechend dem Aufdruck auf der Grundplatine links unten. Diese Schalter sind Teil des 8-fach DIL-Schalters, der sich auf dem 16-poligen Sockel der Grundplatine befindet. Die Schalter 1-4 dienen der Programmeingabe, die Schalter 6-8 der Adresseingabe. Der Schalter 5 wird also nicht benutzt.

Das mit den Schaltern eingestellte Bitmuster erscheint, nachdem wir den Schreibtaster betätigt haben, auf den 7 Programmleuchtdioden der Anzeigeplatine. Bevor der nächste Programmschritt eingegeben werden kann, betätigen wir noch den Taktaster, dadurch wird der Zählerstand um 1 erhöht. Damit ergibt sich das folgende Muster, um einzelne Programmschritte in den WDR-1-Bit-Computer einzuschreiben:

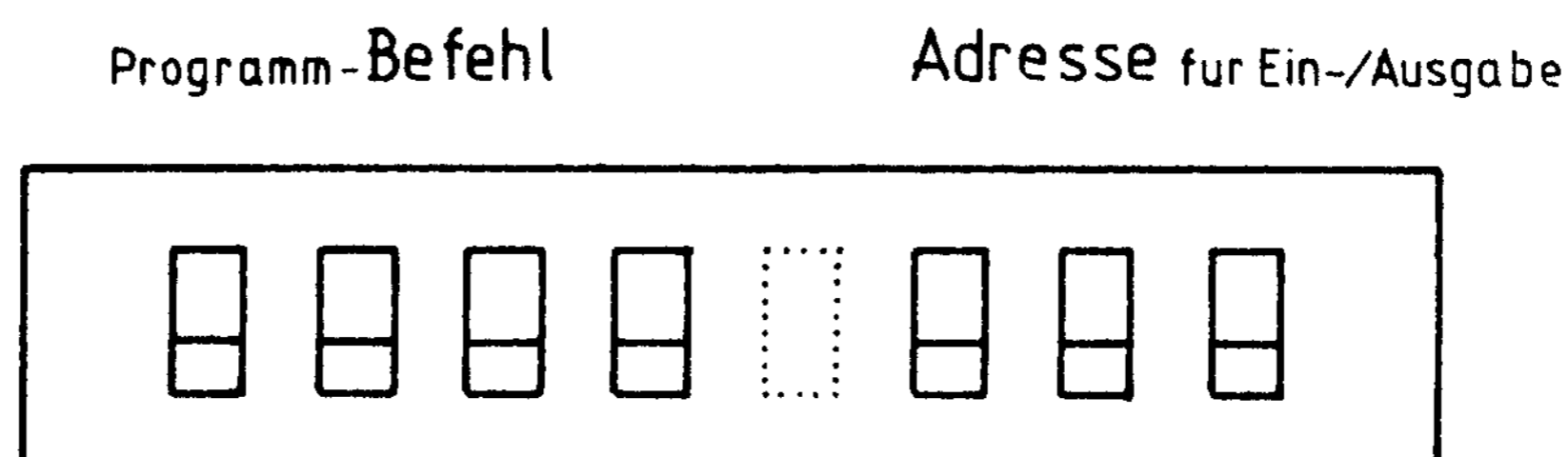


Abb. 41: Schema der Eingabe eines Programmschritts

Wenn wir den Computer eingeschaltet haben, stehen in seinem Speicher irgendwelche zufälligen Informationen, die uns stören können wenn wir programmieren. Wir setzen daher mit dem Programmschritt NOP 0 alle Speicherzellen auf Null:

1. Taktwahlschalter auf Schnelltakt "ST" stellen.
2. NOP 0, also das Bitmuster 0000 000, am 8-fach DIL-Schalter einstellen.
3. Den Schreibtaster "S" betätigen, wodurch alle Programmspeicher auf 0 gesetzt werden.
4. Den Taktwahlschalter auf Handtakt "HT" stellen.
5. Den Programmanfangstaster "P" betätigen. Dadurch wird der Programmzähler auf 0 gesetzt.
6. Den Rücksetztaster "R" drücken, wodurch alle Informationen in den Registern des Prozessors und des Ausgabebausteins gelöscht werden.

Der 5. Schritt sorgt also dafür, daß jedes Programm bei der Programmadresse 0 bzw. binär "0000 0000" beginnt. Aus Gründen der Vereinfachung werden wir bei unseren Programmen den Stand des Programmzählers nicht binär sondern dezimal angeben, da unser Computer den Wert des Zählers nicht anzeigt.

Nach den ersten sechs Vorbereitungsschritten, die der besseren Übersicht beim Programmieren dienen, folgen hieraus drei zusätzliche Schritte, um den Computer für weitere Eingaben durch den Programmierer vorzubereiten.

FREIGABE FÜR AUS- UND EINGABE

Die Ein- und die Ausgabe von Daten in die Logikeinheit (LU = logical unit) bzw. aus der LU muß erst mit Hilfe zweier Register, des IEN- und des OEN-Registers, freigegeben werden.

Das IEN-Register wird durch den IEN-Befehl (Eingangsfreigabe-Befehl) Input-ENable-Befehl geladen.

Ist im IEN-Register einen "0" gespeichert, so können keine Daten in die CPU eingelesen werden, denn am Ausgang des IEN-Registers liegt dann immer eine "0", unabhängig von ihren tatsächlichen Wert. Ist im IEN-Register hingegen eine "1" gespeichert, so werden alle Daten korrekt eingelesen.

Um zu gewährleisten, daß Eingangssignale auch tatsächlich in die Logikeinheit gelangen, muß dafür gesorgt werden, daß im IEN-Register eine "1" steht.

Wir müssen also dafür sorgen, daß einer der Eingänge einen von uns erwünschten logischen Zustand hat, eine "1" oder eine "0". Betrachten wir den Eingang EO. Er ist direkt mit dem Ausgang des ER verbunden. Wir können daher mit dem Befehl "IEN 0" den Pegel des FR in das IEN-Register laden. Bleibt noch zu erreichen, daß im ER eine 1 steht. Dies erreichen wir mit dem Befehl ORC 0.

Ist die Eingabe gesperrt, so liest die CPU das mit dem Befehl ORC 0 eine "0" vom Ausgang des UND-Gatters ein. (Siehe Kapitel 2.2.1.-Eingangsschaltung der CPU). Die "0" wird in der Logikeinheit invertiert, also zur "1" und dann mit dem Inhalt des ER ODER-verknüpft. Das Ergebnis dieser Verknüpfung ist immer eine "1", es wird ins ER übernommen und kann dann über Pin 15 der CPU über die Eingabeeinheit und über die Datenleitung ins IEN-Register geladen werden. Danach ist die Eingabe von Daten freigegeben.

Mit der Befehlsfolge ORC 0, IEN 0 können wir also erreichen, daß Signale über die Datenleitung in die CPU eingelesen werden. Wir müssen nun noch dafür sorgen, daß unser Computer Daten auch ausgeben kann. Hierfür ist das OEN-Register in Verbindung mit dem OEN-Befehl zuständig (OEN steht für Output ENable, also für Ausgangsfreigabe. Das OEN-Register wird in gleicher Weise wie das IEN-Register über die Datenleitung geladen. Steht im OEN-Register eine "1", so können Daten mit Hilfe des STO- oder STOC-Befehls über die Datenleitung an die Ausgabeeinheit gelangen. (Siehe Kapitel 2.2.1.-Ausgangsschaltung der CPU).

Es ist also auch notwendig, das OEN-Register mit einer "1" zu laden, um den Mikroprozessor und damit unseren Computer zur Programmbearbeitung vorzubereiten. Dieses Vorbereiten nennt man Initialisieren.

Zusammenfassend lautet unser Initialisierungsprogramm INIT also wie folgt:

INIT

Programmadresse	Programmschritt Befehl	Adresse
001	ORC	0
002	IEN	0
003	OEN	0

Dieses Initialisierungsprogramm besteht aus drei Programmschritten. Es steht immer am Anfang eines jeden Programms und wird in Zukunft mit INIT abgekürzt:

000 INIT

Und so werden diese drei Programmschritte in den Computer eingegeben:
7. ORCO, also das Bitmuster 0110 000, am 8-fach DIL-Schalter einstellen und den Schreibtaster betätigen.

8. Danach wird der Takt-Taster einmal betätigt. Hierdurch springt der Programmzähler auf die Programmadresse 1. Gleichzeitig wird der Programmschritt ORC 0 von der Programmadresse 0 ausgeführt.

9. In gleicher Weise wie ORC 0 werden auch IEN 0, binär: 1010 000, und OEN 0, binär: 1011 000, eingeben.

(In Programmen, in denen keine Daten eingegeben werden brauchen, kann der Programmschritt IEN 0 fehlen. Genauso kann OEN 0 fehlen, wenn keine Daten ausgegeben werden sollen.)

Von großer Bedeutung ist, daß der ER-Ausgang mit dem Eingang E0 verbunden ist und daß der Inhalt des ER durch die Schritte 2 und 3 nicht verändert wird.

Ausgang und Eingang sind durch diese Initialisierung nun freigegeben. Erst jetzt werden alle weiteren Programmschritte korrekt ausgeführt.

4.3 EINFACHE PROGRAMME

4.3.1. PROGRAMME ZUR EINGABE VON DATEN

In den WDR-1-Bit-Computer können Informationen mit Hilfe des 4-fach DIL-Schalter oder über die Anschlüsse 3 bis 5 und 14 des Expansionssteckers mit den Eingabebefehlen LD oder LDC eingegeben werden. Diese Anschlüsse sind, wie das Schaltbild zeigt, miteinander

verbunden. Schließen wir die 4 Eingangsschalter oder legen wir eine positive Spannung von 5 Volt an die genannten Anschlüsse des Expansionssteckers, so leuchten die entsprechenden 4 Eingangsleuchtdioden.

Das folgende Programm wird eingegeben, es zeigt uns, wie der logische Pegel des Eingangs gelesen und im Ergebnisregister abgespeichert wird:

****Eingang E1 einlesen****

```
00 INIT      INIT=Initialisierung, also die Programmschritte
              *
              00 ORC 0 , binär: 0110 000
              01 IEN 0 , binär: 1010 000
              02 OEN 0 , binär: 1011 000
              *
03 LD        Lade den Pegel des Eingangs 1, binär: 0001 001
```

Wenn wir das Programm starten, indem wir den Zähler mit dem Programmvorwahltaster "P" auf den Programmstart setzen und dann z.B. den Taktaster viermal betätigen, zeigt uns die "ER"-Leuchtdiode über den Inhalt des Ergebnisregisters an, welcher Eingangspegel am Eingang 1 gerade anliegt.

Wenn wir nun weitertakten, durchläuft der Computer unnötigerweise sämtliche restlichen 252 Speicherstellen. Um dies zu verhindern, lautet der nächste Befehl: Springe an den Anfang zurück, also Jump (JMP):

```
04 JMP x     binär 1100 xxx Das x bedeutet eine beliebige Adresse.
              Eine Adressenangabe erübrigt sich bei diesem Befehl, da
              weder die Eingabe- noch die Ausgabeeinheit
              angesprochen werden.
```

Soll der einzulesende Eingangswert umgekehrt werden, setzen wir anstelle des Befehls LD (Laden) den Befehl LDC (Laden Komplement). Der Programmschritt 03 lautet dann:

```
03 LDC 1 , binär: 0010 001
```

Dieser Befehl erspart zusätzliche Hardware (in diesem Falle den Einbau eines Inverters), wenn aus programmtechnischen Gründen invertierte Signale benötigen.

Bisher haben wir nur den Pegel des Eingangs E1 eingelesen. Wollen wir die Pegel der anderen drei Eingänge (E2-E4) in den Computereingaben, ändern wir die Adresse: LD 3 (binär 0001 011) lädt z.B. den Pegel des Eingangs E3.

4.3.2 PROGRAMME ZUR AUSGABE VON DATEN

Um die im Computer verarbeiteten Eingangssignale auszugeben, wenden wir die Ausgabebefehle STO bzw. STOC an. Dadurch wird der sich im Ergebnisregister befindende Pegel bzw. dessen Komplement auf die Ausgänge A0 bis A7, das sind die Anschlüsse 6 - 13 des Expansionssteckers, gelegt und dort gespeichert. Daher der Befehlsname Store.

Auch hier gilt wieder, daß die Ausgangsadresse mit dem Befehl eingegeben werden muß, will man einen bestimmten Ausgang anwählen.

Ist z.B. der Wert des Ergebnisregisters "1", und soll der Inhalt des Ergebnisregisters auf den Ausgang 4 gelesen werden, so lautet der entsprechende Befehl STO 4.

Wie bei der Eingabe kann auch bei der Ausgabe das Komplement gebildet werden. STOC 4 gibt also das Gegenteil des Wertes des Ergebnisregisters auf A4.

Das folgende Programm schaltet den Ausgang A4 immer wieder ein und aus. Es handelt sich um einen Rechteckgenerator oder astabilen Multivibrator.

****Rechteckgenerator****

```
00 INIT
03 STO 4   binär: 1000 100
04 STOC 4   "    1001 100
05 JMP x   "    1100 xxx
```

Soll das Programm automatisch ablaufen, und wollen wir es dabei beobachten, stellen wir den Taktwahlschalter auf Langsamtakt "LT". Durch Variieren des Potentiometers kann die Frequenz der Rechteckpulse geändert werden. Dabei beobachten wir, daß das Tastverhältnis zwischen hell und dunkel der Ausgangsleuchtdiode unsymmetrisch ist. Soll unser Rechteckgenerator dazu benutzt werden, gleich lange Impulse zu liefern, müssen wir es ändern. Die Binärwerte übertragen wir einfach aus dem obigen Programm: **** Rechtecksgenerator ****

****Rechteckgenerator mit symmetrischem Tastverhältnis****

```
00 INIT
03 STO 4
04 NOP x
05 NOP x
06 NOP x
07 NOP x
08 STOC 4
0  JMP x
```

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:

****Lauflicht****

```
00 INIT
03 STO 0
04 STO 1
05 STO 2
06 STO 3
07 STO 4
```

```

08 STO 5
09 STO 6
13 STO 7
11 STOC 7
12 STOC 6
13 STOC 5
14 STOC 4
15 STOC 3
16 STOC 2
17 STOC 1
18 STOC 0
19 JMP x

```

Natürlich können wir unser Programm um weitere 237 Programmschritte erweitern. Dann allerdings empfehlen wir die Benutzung der Tastatur.

4.4. ES WIRD BEQUEMER: DIE TASTATUR

Die Eingabe von Programmen über den DIL-Schalter erweist sich zu Beginn als sehr lehrreich, jedoch auch als sehr unkomfortabel. Zudem muß immer vor der Eingabe eines Programmschrittes das Bitmuster dieses Programmschrittes ermittelt werden. Daher benutzen wir in Zukunft die Tastatur, um Programme einzugeben.

Zunächst entfernen wir den 8-poligen DIL-Schalter und setzen an seine Stelle den Tastaturstecker so ein, daß die 7 Anschlüsse in der hinteren Reihe stecken, der eine Anschluß also vorne liegt. Die Tastatur ist damit schon gebrauchsfähig.

Um einen Programmschritt einzugeben, drücken wir den gewünschten Befehlstaste und den Adressentaster gleichzeitig (!). Zusätzlich betätigen wir mit der anderen Hand den Schreibtaster. Damit ist der Programmschritt schon gespeichert. Nach der Betätigung des Taktasters können wir auf dieselbe Art den nächsten Programmschritt eingeben.

4.5. WEITERE PROGRAMME

4.5.1. LOGISCHE VERKNÜPFUNGEN

In Kapitel 2.2.1. haben wir die wichtigsten logischen Verknüpfungen kennengelernt. In diesem Kapitel sollen logische Verknüpfungen mittels eines Programmes mit dem Computer verwirklicht werden.

Gleichzeitig sollen die Rechenbefehle AND, ANDC, OR, ORC und XNOR kennengelernt werden.

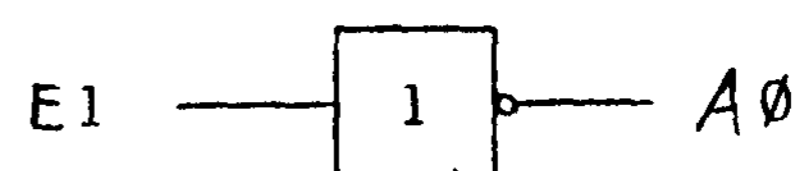
Die Eingänge einer logischen Schaltung sind E1, E2, E3 bis E4. Das Ergebnis der logischen Verknüpfung wird auf A0 angezeigt. Ist ein Programm eingegeben, so wird der Taktwahlschalter auf Stellung "Schnell-Takt" gestellt. Betätigt man die Schalter für die Eingänge, so erscheint an A0 sofort das Ergebnis der logischen Verknüpfung.

1. NICHT-Glied (Inverter)

Wahrheitstabelle:

E1	A0
0	1
1	0

Schaltzeichen:



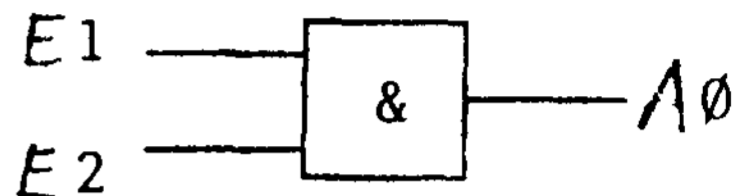
Programm: 00 INIT
 03 LD 1
 04 STOC 0
 05 JMP x

2. UND-Glied mit zwei Eingängen

Wahrheits_tabelle: E2 E1 A0
 0 0 0
 0 1 0
 1 0 0
 1 1 1

Programm: 00 INIT
 03 LD 1
 04 AND 2
 05 STO 0
 06 JMP x

Schaltung:

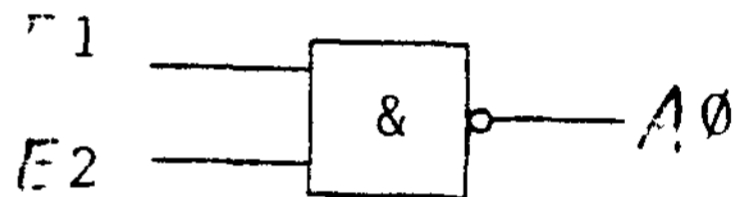


3. NAND-Glied mit zwei Eingängen

Wahrheitstabelle: E2 E1 A0
 0 0 1
 0 1 1
 1 0 1
 1 1 0

Programm: 00 INIT
 03 LD 1
 04 AND 2
 05 STOC 0
 06 JMP x

Schaltung:

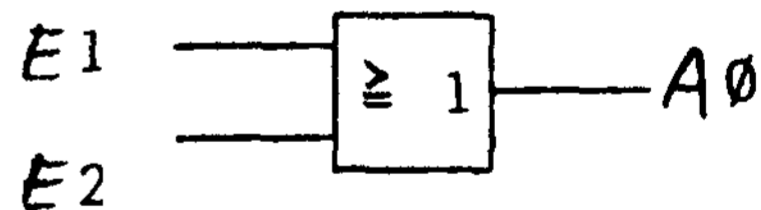


4. ODER-Glied mit zwei Eingängen

Wahrheitstabelle: E2 E1 A0
 0 0 0
 0 1 1
 1 0 1
 1 1 1

Programm: 00 INIT
 03 LD 1
 04 OR 2
 05 STO 0
 06 JMP x

Schaltung:

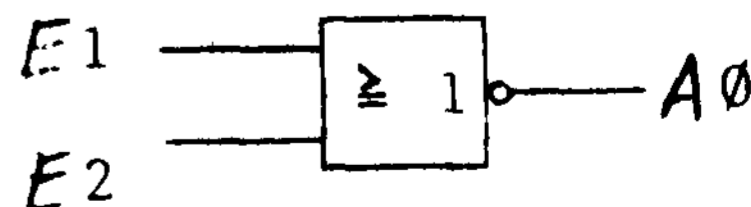


5. NOR-Glied mit zwei Eingängen

Wahrheitstabelle: E2 E1 A0
 0 0 1
 0 1 0
 1 0 0
 1 1 0

Programm: 00 INIT
 03 LD 1
 04 OR 2
 05 STOC 0
 06 JMP x

Schaltung:

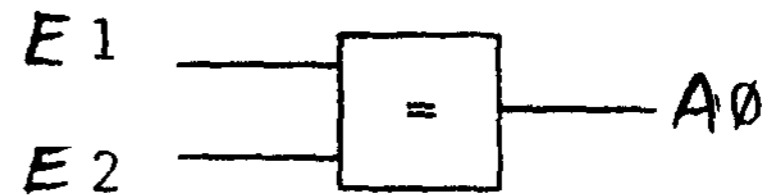


6. Äquivalenz-Glied (Exklusiv-NOR-Glied) mit zwei Eingängen

Wahrheitstabelle:

E2	E1	A0
0	0	1
0	1	0
1	0	0
1	1	1

Schaltung:



Programm:

```

00 INIT
03 LD 1
04 XNOR 2
05 STO 0
06 JMP x

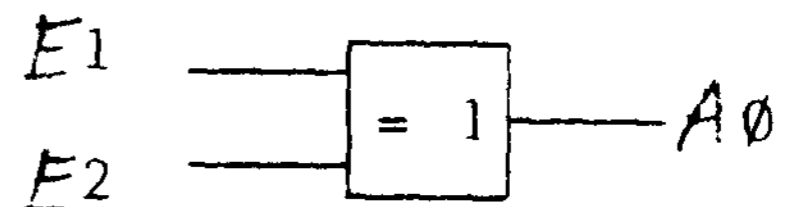
```

7. Antivalenz-Glied (Exklusiv-ODER-Glied) mit zwei Eingängen

Wahrheitstabelle:

E2	E1	A0
0	0	0
0	1	1
1	0	1
1	1	0

Schaltung:



Programm:

```

00 INIT
03 LD 1
04 XNOR 2
05 STOC 0
06 JMP x

```

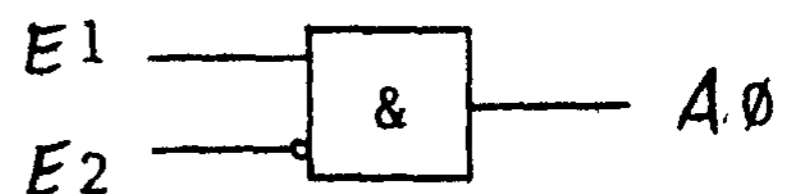
8. Inhibitions-Glied

Die Inhibition ist eine besondere Art der UND-Verknüpfung. Ein Eingangszustand wird vor der UND-Verknüpfung negiert.

Schaltung:

Wahrheitstabelle:

E2	E1	A0
0	0	0
0	1	1
1	0	0
1	1	0



Programm:

```

00 INIT
03 LD 1
04 ANDC 2
05 STO 0
06 JMP x

```

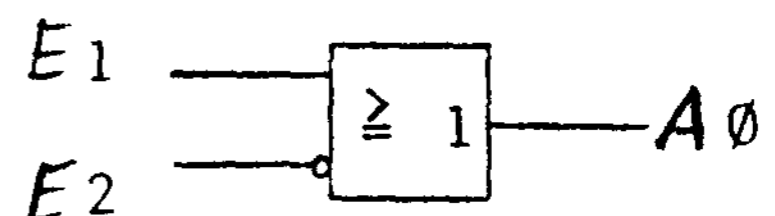
9. Implikations-Glied

Die Implikation ist eine besondere Art der ODER-Verknüpfung. Ein Eingangszustand wird vor der ODER-Verknüpfung negiert.

Schaltung:

Wahrheitstabelle:

E2	E1	A0
0	0	1
0	1	1
1	0	0
1	1	1



Programm:

```

00 INIT
03 LD 1
04 ORC 2
05 STO 0
06 JMP x

```


9. Vergleich zweier logischer Schaltungen

Als Beispiel nehmen wir das 1. de Morgansche Gesetz, dessen allgemeine Gültigkeit unser Computer überprüfen soll.

Es lautet: $\overline{E1 \vee E2} = \overline{E1} \wedge \overline{E2}$

Das Programm:

** 1. de Morgansches Gesetz **

*

```
00 INIT
03 LD 1
04 OR 2
05 STOC 7
06 LDC 1
07 ANDC 2
08 XNOR 7
09 STO 0
10 JMP x
```

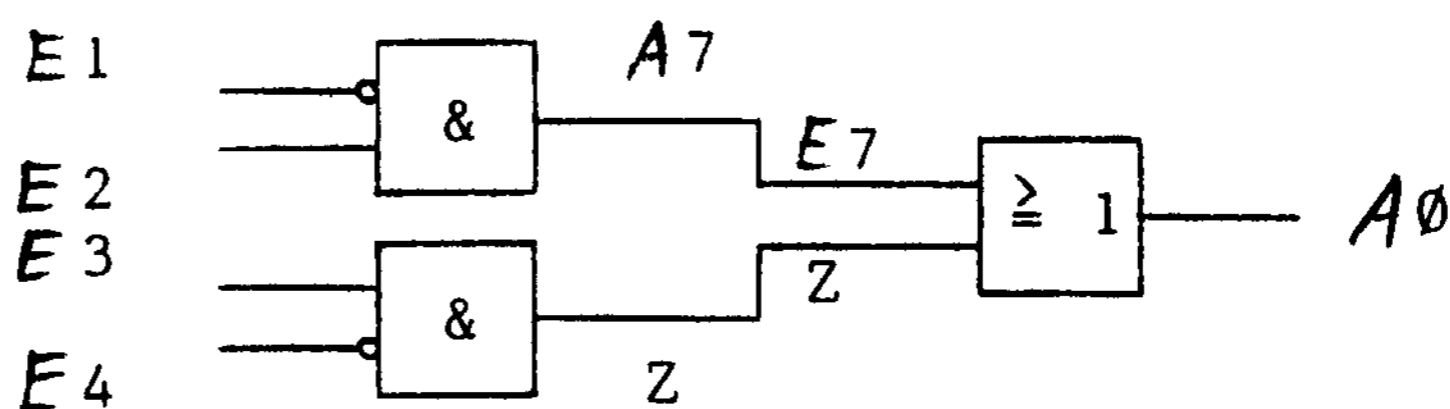
Im 5. Programmschritt wird das Komplement der Verknüpfung $E1 \vee E2$ zwischengespeichert. Als Zwischenspeicher stehen die drei Datenspeicher A5, A6 und A7 zur Verfügung. Im 8. Schritt werden dann die beiden Seiten der Gleichung miteinander verglichen. Um das Gesetz zu überprüfen, betätigen wir die Eingangsschalter E1 und E2. Das Ergebnis dieses Vergleichs liegt am Ausgang A0 an. Leuchtet die Ausgangsleuchtdiode, dann liegt Gleichheit vor.

10. Logische Schaltung mit mehreren Gliedern

Möchte man mehrere Glieder miteinander verbinden, so muß beachtet werden, daß das Ergebnis jeder parallelen logischen Verknüpfung, bis auf das Ergebnis der letzten parallelen logischen Verknüpfung, zwischengespeichert werden muß.

Im folgenden Beispiel wird der Datenspeicher A7 benutzt. Dort erscheint immer das Ergebnis der ersten logischen Verknüpfung. Das Ergebnis der zweiten logischen Verknüpfung wird willkürlich "Z" genannt.

Schaltung:



Wahrheitstabelle:

E4	E3	E2	E1	A7/E7	Z	A0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	1
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	1
1	0	1	1	0	0	0
1	1	0	0	0	0	0
1	1	0	1	0	0	0
1	1	1	0	1	0	1
1	1	1	1	0	0	0

```

Programm:      00 INIT
                03 LDC  1
                04 AND  2
                05 STO  7
                06 LD   3
                07 ANDC 4
                08 OR   7
                09 STO  0
                10 JMP  x
  
```

4.5.2. PROGRAMMVERZWEIGUNGEN

Mit Hilfe des IEN-Befehls, des OEN-Befehls und des SKZ-Befehls ist es möglich, innerhalb eines Programmes nur bestimmte Programmblöcke zu bearbeiten, andere Programmabschnitte werden dagegen ignoriert. Dem IEN-Befehl kommt dabei nur eine geringe Bedeutung zu. Hier sollen also nur der OEN-Befehl und der SKZ-Befehl behandelt werden.

Um die Programmverzweigung deutlich zu machen, werden die Programme mit Hilfe eines Flußdiagrammes dargestellt. Ein Flußdiagramm gibt den strukturellen Verlauf eines Programmes wieder. Es besteht aus Marken, durch Kreise oder Ellipsen gekennzeichnet, aus Abfragen, durch Karos gekennzeichnet und aus Zuordnungen, durch Rechtecke gekennzeichnet. Marken stehen immer am Programmanfang (Abkürzung: A), am Programmende und dort, wo Programmzweige wieder zusammenlaufen. Abfragen stehen immer dort, wo Entscheidungen nötig sind. Deswegen verzweigt sich hier das Programm in zwei Programmzweige. Die Abfrage kann nur mit "ja" oder "nein" beantwortet werden. Je nachdem wird der eine oder der andere Programmzweig bearbeitet. In den Zuordnungen stehen arithmetische Anweisungen.

1. Die WENN-DANN-Programmierung

Die Benutzung der Bezeichnung WENN-DANN sagt aus: WENN ein Zustand eingetreten ist, DANN soll ein bestimmter Programmabschnitt bearbeitet werden. Ist der Zustand dagegen nicht erreicht, so wird der bestimmte Programmabschnitt ignoriert. Die WENN-DANN-Beziehung soll an einem Beispiel verdeutlicht werden:

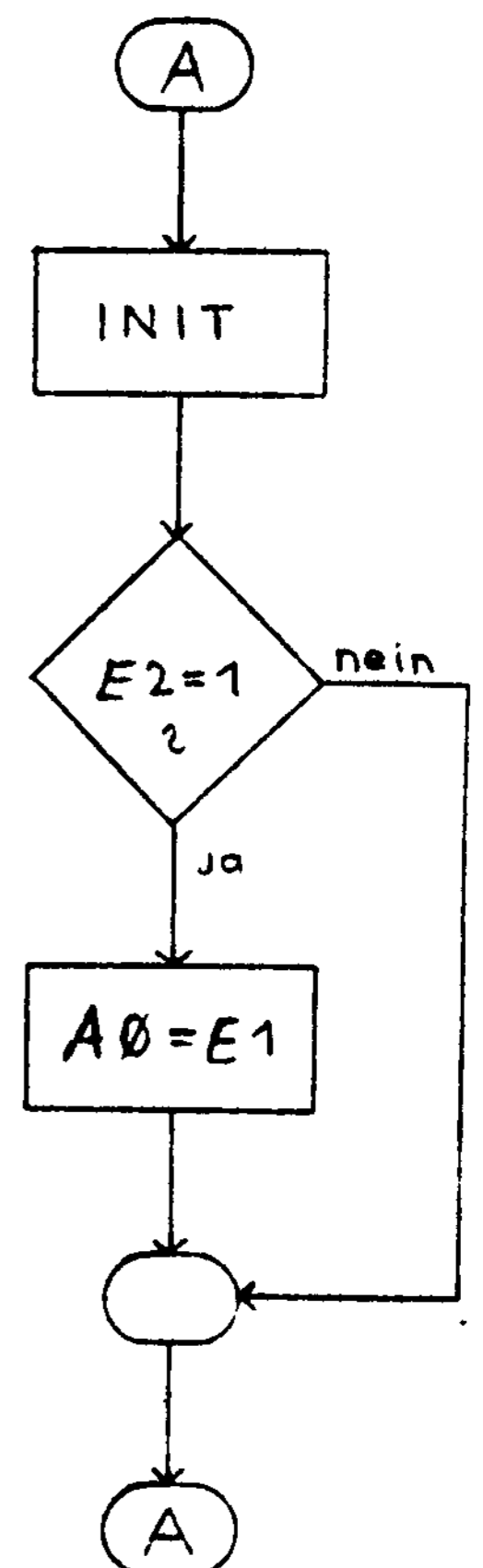
WENN die Ausgabe durch den Eingang E2 freigegeben ist, DANN soll der logische Pegel des Eingangs E1 am Ausgang A0 erscheinen. Ist die Ausgabe nicht freigegeben, so erscheint der logische Pegel des Eingangs E1 nicht.

Nebenstehend das Flußdiagramm!

```

Programm:      00 INIT
                03 OEN  2
                04 LD   1
                05 STO  0
                06 ANDC 0
                07 STO  0
                08 JMP  x
  
```

Stellung: Schnell-Takt



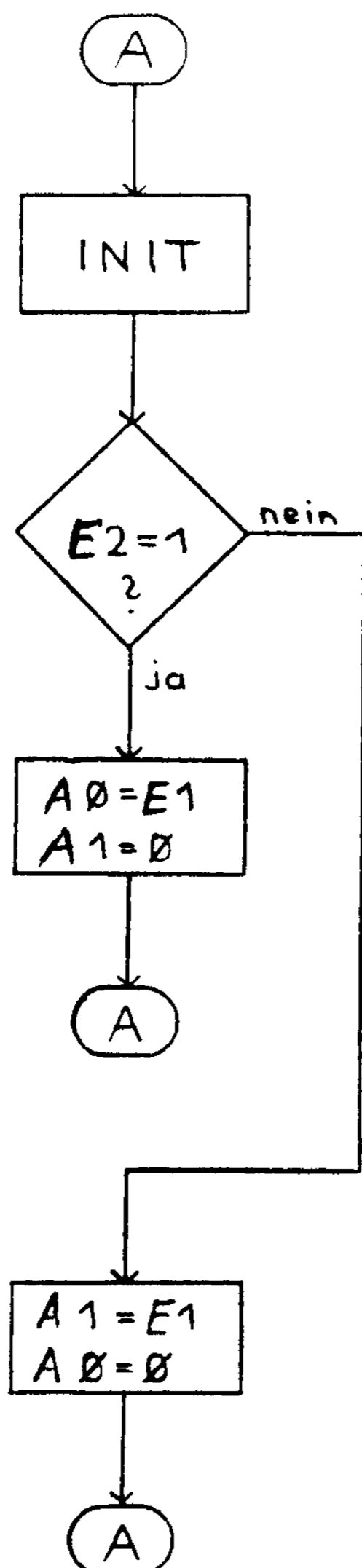
Die WENN-DANN-Beziehung wird mit Hilfe des OEN-Befehls möglich gemacht. OEN 2 bewirkt folgendes: Die Information des Eingangs E2 wird in das OEN-Register geladen. Steht dann dort eine "1", so wird die Information des Eingangs E1 auf den Ausgang A0 gesetzt. Steht dort jedoch eine "0", so ist die Ausgabe nicht freigegeben und STO 0 wird ignoriert. Dieses hat die gleiche Wirkung wie ein Überspringen dieses Programmschrittes. (ANDC 0 und STO 0 bewirken ein Löschen der Information des Ausgangs A0, damit nicht der Eindruck entsteht, daß eine Information ausgegeben wird, wenn die Ausgabe gesperrt ist.) Ist dieses Programm Teil eines anderen Programmes, d.h., soll am Ende dieses Programmes nicht zurückgesprungen werden, sondern sollen weitere Programmteile bearbeitet werden, so muß die Ausgabe wieder freigemacht werden.

2. Die WENN-DANN-SONST-Programmierung

Die Benutzung der Bezeichnung WENN DANN SONST sagt aus: WENN ein Zustand eingetreten ist, DANN soll ein bestimmter Programmabschnitt bearbeitet werden; SONST soll ein anderer Programmabschnitt bearbeitet werden. DIE WENN-DANN-SONST-Beziehung soll auch an einem Beispiel erläutert werden:

WENN die Ausgabe durch den Eingang E2 freigegeben ist, DANN soll der logische Pegel des Eingangs E1 am Ausgang A0 erscheinen; SONST, d.h., wenn die Ausgabe nicht freigegeben ist, soll er am Ausgang A1 erscheinen.

Flußdiagramm nebenstehend!



Programm:

00	INIT
03	LD 1
04	STO 5
05	OEN 2
06	STO 0
07	ANDC 0
08	STO 1
09	LDC 2
10	OEN 0
11	LD 5
12	STO 1
13	ANDC 0
14	STO 0
15	JMP x

Stellung: Schnell-Takt

Die WENN-DANN-SONST-Beziehung wird mit Hilfe zweier OEN-Befehle möglich gemacht. OEN 2 bewirkt das Gleiche wie bei der WENN-DANN-Beziehung. Er entscheidet über die Freigabe des ersten Programmblocks (Programmadressen 06...08). LDC 2 und OEN 0 stellen für sich auch eine WENN-DANN-Beziehung dar. Sie

entscheiden über die Freigabe des zweiten Programmblöcks (Programmadressen 11...14). Da immer nur einer der beiden Programmblöcke bearbeitet wird, ergeben beide zusammen eine WENN-DANN-SONST-Struktur.

3. Die SOLANGE-Programmierung

Die Benutzung der Bezeichnung SOLANGE sagt aus: SOLANGE ein bestimmter Zustand nicht eingetreten ist, soll das Programm in einer Warteschleife laufen. Ist der Zustand eingetreten, so wird das restliche Programm durchlaufen. Die SOLANGE-Beziehung soll ebenfalls an einem Beispiel erklärt werden:

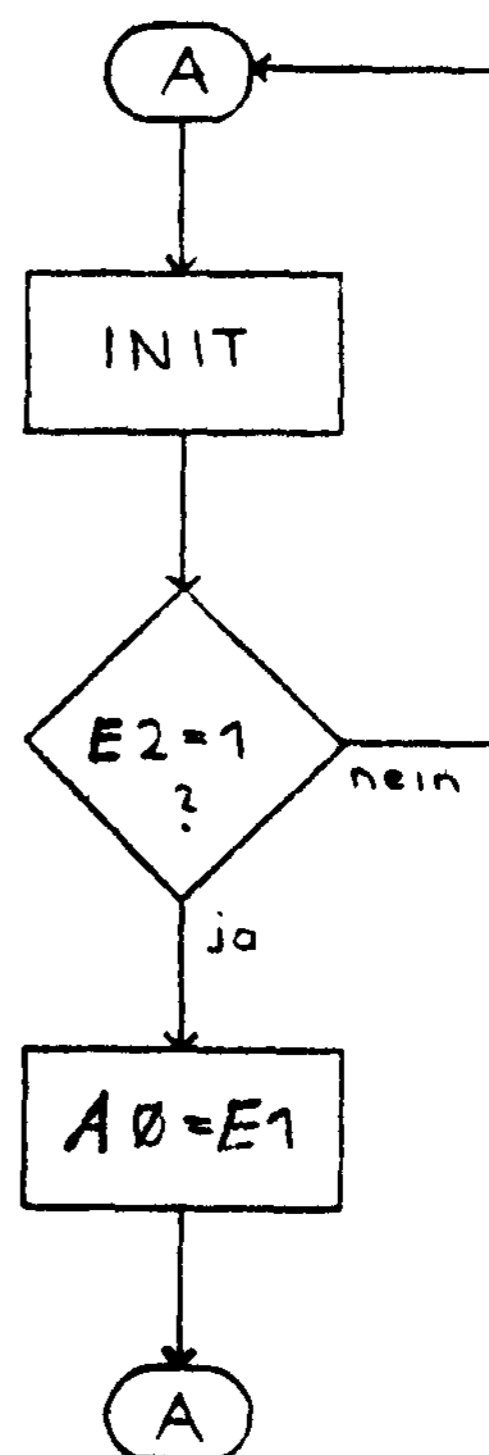
SOLANGE die Ausgabe durch den Eingang E2 nicht freigegeben ist, soll der logische Pegel des Eingangs E1 am Ausgang A0 nicht erscheinen.

Flußdiagramm nebenstehend!

Programm:	00 INIT
	03 LDC 2
	04 SKZ x
	05 JMP x
	06 LD 1
	07 STO 0
	08 ANDC 0
	09 STO 0
	10 JMP x
Stellung:	Schnell-Takt

Die SOLANGE-Beziehung wird mit Hilfe des SKZ-Befehls und des JMP-Befehls möglich gemacht. Steht nach LDC 2 eine "1" im Ergebnisregister, so wird JMP x (Programmadresse 05) ausgeführt und das Programm startet wieder von neuem. Dies geschieht solange, bis mit LDC 2 eine "0" ins Ergebnisregister gelangt. Dann wird JMP x ignoriert und das restliche Programm bearbeitet.

Beim Eingeben des Programmes ist darauf zu achten, daß vor dem Einschreiben von SKZ x eine "0" im Ergebnisregister steht, da sonst nach JMP x der Programmzähler auf der Programmadresse 00 steht.

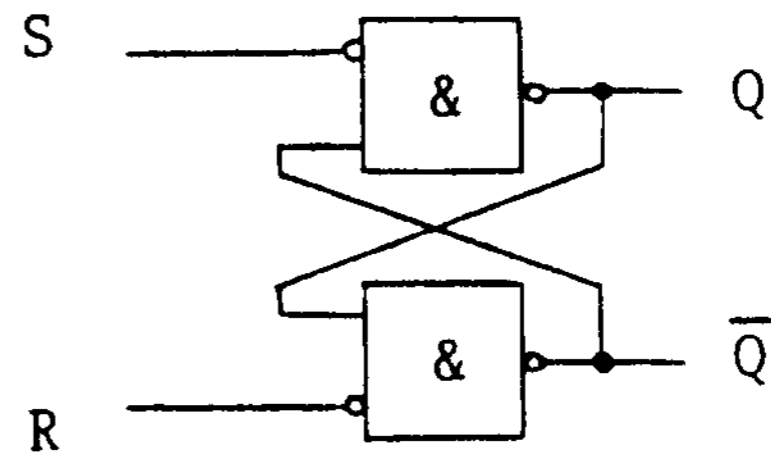


4.5.3. Programme zur Darstellung von Flipflops

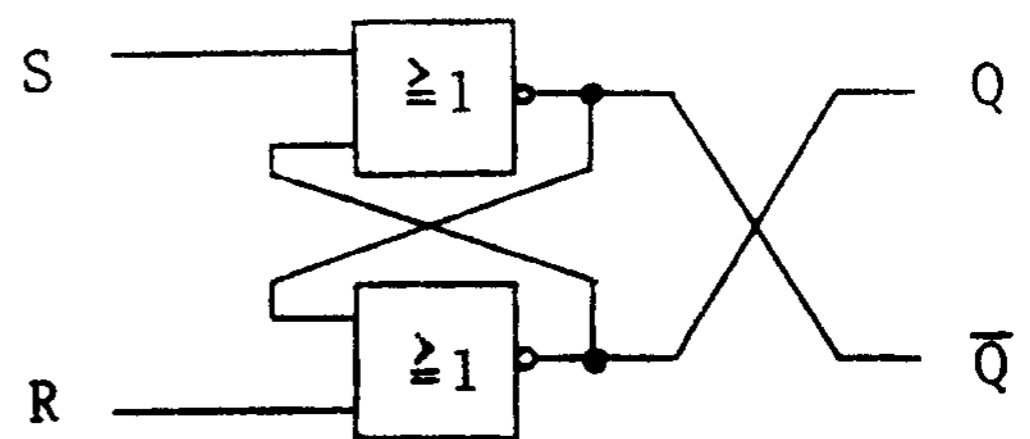
Drei Flipflop-Typen, das RS-Flipflop, das D-Flipflop und das JK-Flipflop sollen mit Hilfe von Programmen dargestellt werden. Alle Programme laufen in Stellung "Schnell-Takt" ab.

1. Ungetaktetes RS-Flipflop

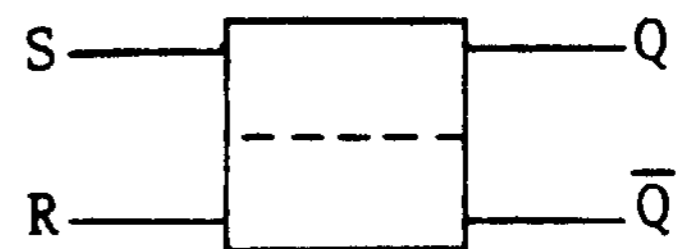
Schaltung (aufgebaut aus NAND-Gliedern):



Schaltung (aufgebaut aus NOR-Gliedern):



Schaltzeichen:



Wahrheitstabelle:

R	S	Q	\bar{Q}
0	0	Q-	\bar{Q} -
0	1	1	0
1	0	0	1
1	1	-	-

Ein RS-Flipflop kann aus NAND-Gliedern und aus NOR-Gliedern aufgebaut werden. Beide Flipflops haben einen nicht zulässigen Fall: $R=S=1$. Weiterhin haben sie einen Speicherfall: $R=S=0$. Hierbei bleibt der alte Ausgangszustand erhalten. Alte Zustände

werden in diesem Kapitel durch ein Minuszeichen gekennzeichnet. Alte Ausgangszustände heißen also Q - und \bar{Q} -.

Springt man vom nicht zulässigen Fall zum Speicherfall, so ist der Ausgangszustand ein Zufallsergebnis. Er ist davon abhängig, welches der beiden Glieder schneller umschaltet. Daher sollte dieser Fall vermieden werden. Im Programm wird dieser Fall durch den Speicherfall ersetzt.

Neue Wahrheitstabelle:

R	S	Q	\bar{Q}
0	0	Q -	\bar{Q} -
0	1	1	0
1	0	0	1
1	1	Q -	\bar{Q} -

Flußdiagramm nebenstehend!

Grundsätzlich liegt dem Flipflop eine WENN DANN Beziehung zugrunde. Um das Programm für ein Flipflop zu schreiben, müssen wir die in der Wahrheitstabelle enthaltenen Ein- und Ausgänge auf unseren Computer übertragen. Festlegung der Zuordnungen im Programm: $R = E2$, $S = E1$, $Q = A0$, $\bar{Q} = A1$.

```

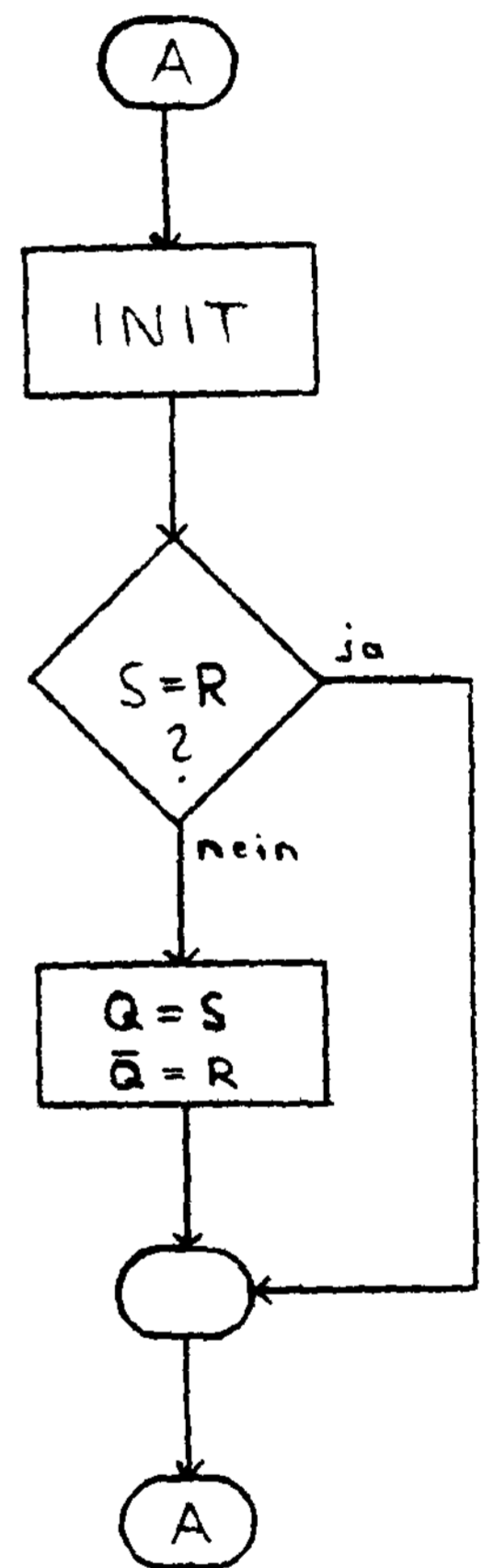
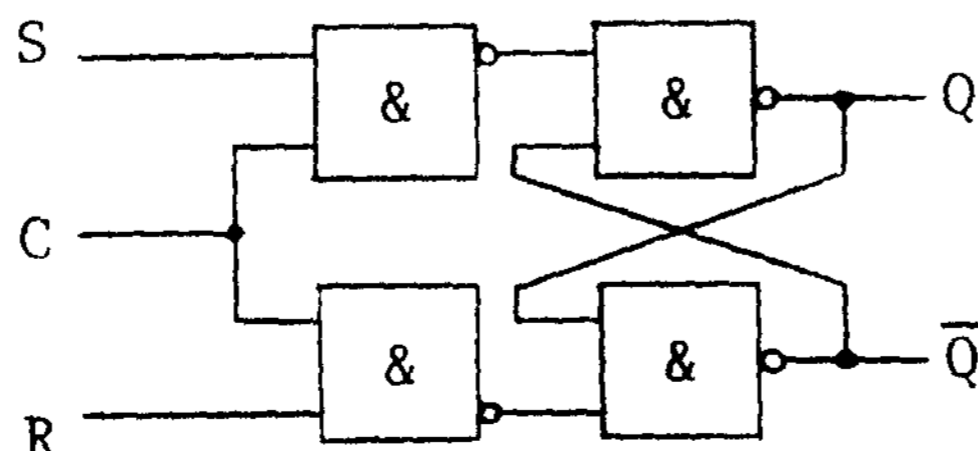
00 INIT                               ** Flipflop **
03 LD  1
04 STO 5
05 LDC 2
06 XNOR 5
07 OEN 0
08 LD  5
09 STO 0
10 STOC 1
11 JMP  x

```

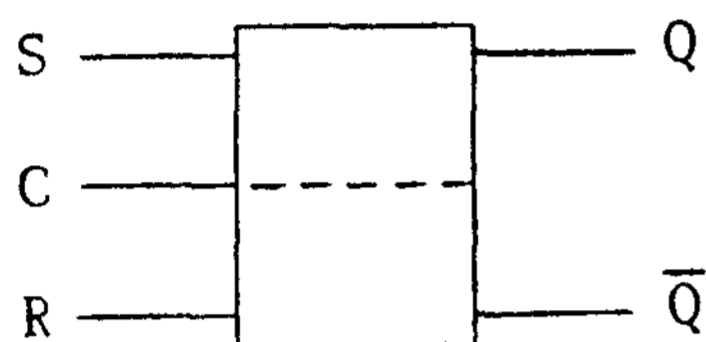
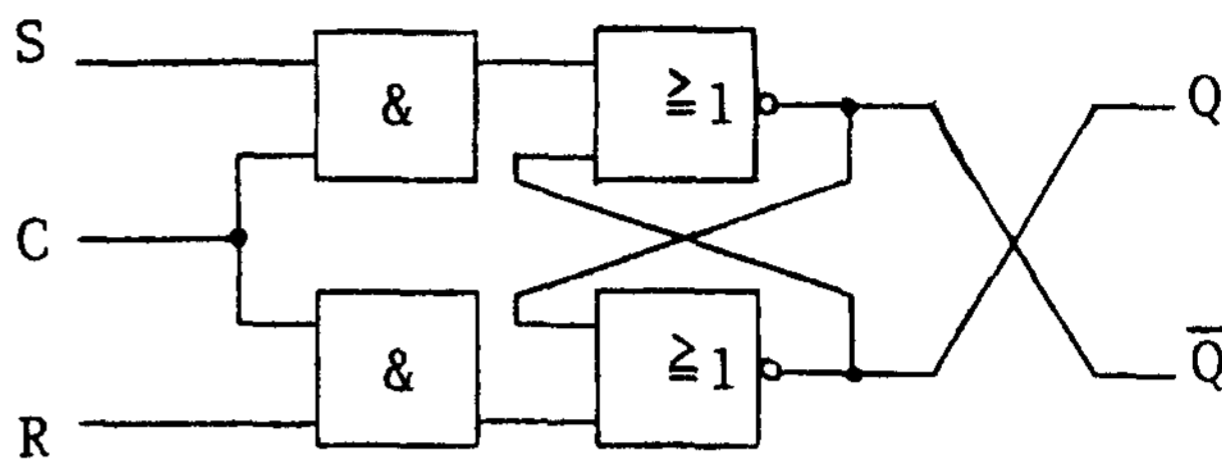
Durch LD 1 und STO 5 wird der Pegel des Eingangs $S = E1$ eingelesen und auf dem Ausgang $A5$ zwischengespeichert. Mit LDC 2 wird der invertierte Pegel des Eingangs $R = E2$ in das Ergebnisregister geladen. XNOR 5 vergleicht nun den Inhalt des Ergebnisregisters mit dem Ausgang $A5$. Falls $R = "0"$ und $S = "1"$ oder $R = "1"$ und $S = "0"$ ist, so ist das Ergebnis dieser logischen Verknüpfung eine "1". Ansonsten ist das Ergebnis eine "0". Das Ergebnis wird in das OEN-Register geladen. Im ersten Fall ist die Ausgabe freigegeben und mit LD 5, STO 0 und STOC 1 gelangt der zwischengespeicherte Wert von S auf $Q = A0$ und dessen Komplement auf $\bar{Q} = A1$. Im zweiten Fall geschieht an den Ausgängen nichts, da die Ausgabe gesperrt ist.

2. Taktzustandsgesteuertes RS-Flipflop

Schaltung (aufgebaut aus NAND-Gliedern):



Schaltung (aufgebaut aus NOR-Gliedern mit UND-Gliedern im Eingang)



Wahrheitstabelle:

C	R	S	Q	Q̄
0	0	0	Q-	Q̄-
0	0	1	Q-	Q̄
0	1	0	Q-	Q̄
0	1	1	Q-	Q̄
1	0	0	Q-	Q̄-
1	0	1	1	0
1	1	0	0	1
1	1	1	Q-	Q̄-

Schaltzeichen nebenstehend!

Beim taktzustandsgesteuerten RS-Flipflop gibt es als dritten Eingang einen Takt (engl. Clock - C). Ist der Takt "0", so gilt der Speicherfall. Ist der Takt "1", so gilt die Wahrheitstabelle für das ungetaktete RS-Flipflop. Es gibt also auch hier einen nicht zulässigen Fall: C = R = S = 1. Auch hier wird im Programm dieser Fall durch den Speicherfall ersetzt, was in der Wahrheitstabelle schon berücksichtigt wurde.

Flußdiagramm nebenstehend!

Festlegung der Zuordnungen im Programm:

C = E3, R = E2, S = E1, Q = A0, Q̄ = A1

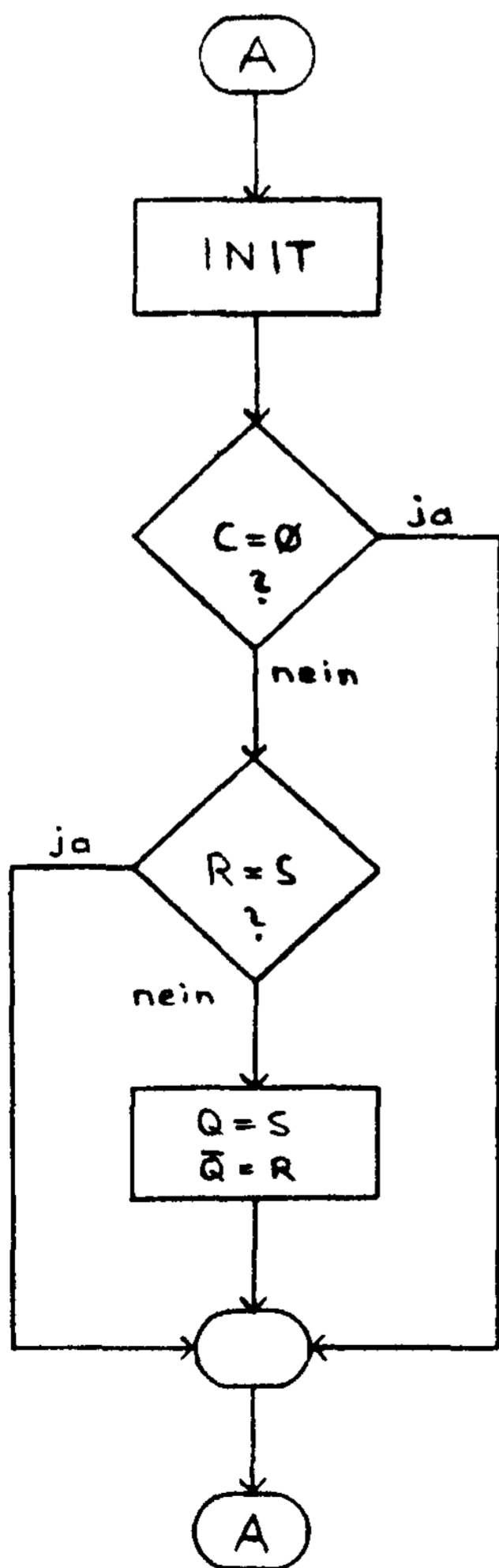
Programm:

```

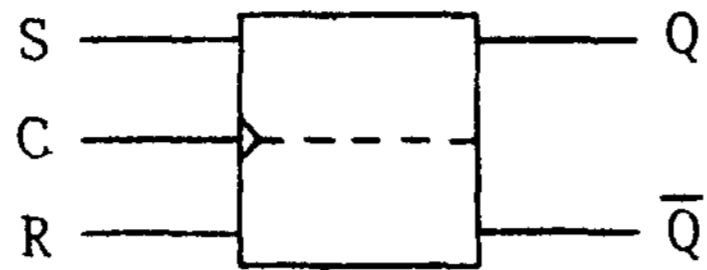
00 INIT
03 LD 1
04 STO 5
05 LDC 2
06 XNOR 5
07 AND 3
08 OEN 0
09 LD 5
10 STO 0
11 STOC 1
12 JMP x
    
```

3. Taktflankengesteuertes RS-Flipflop

Beim taktzustandsgesteuerten RS-Flipflop wurde beim Takt zwischen "0" und "1" unterschieden. Diese Unterscheidung wird beim taktflankengesteuerten RS-Flipflop nicht mehr gemacht. Hier sind die positive Taktflanke, d.h., Übergang von "0" auf "1" (↗), und die negative Taktflanke, d.h., Übergang von "1" auf "0" (↘), von Bedeutung.



RS-Flipflop, das bei positiver Taktflanke schaltet:
Schaltzeichen:



Wahrheitstabelle:

Bei der positiven Taktflanke gilt die Wahrheitstabelle für das ungetaktete RS-Flipflop. Ansonsten gilt der Speicherfall.

Flußdiagramm nebenstehend!

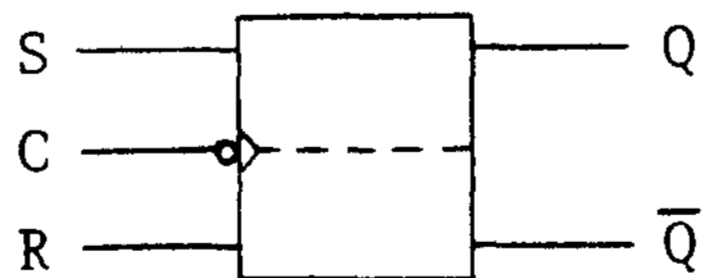
Festlegung der Zuordnungen im Programm:

C- = E5, C = E3, R = E2, S = E1, Q = A0, \bar{Q} = A1

```

Programm:      00 INIT
                03 LD   5
                04 STO  6
                05 LD   3
                06 STO  5
                07 ANDC 6
                08 STO  6
                09 LD   1
                10 STO  7
                11 LDC  2
                12 XNOR 7
                13 AND  6
                14 OEN  0
                15 LD   7
                16 STO  0
                17 STOC 1
                18 JMP  x
    
```

RS-Flipflop, das bei negativer Taktflanke schaltet:
Schaltzeichen:



Wahrheitstabelle:

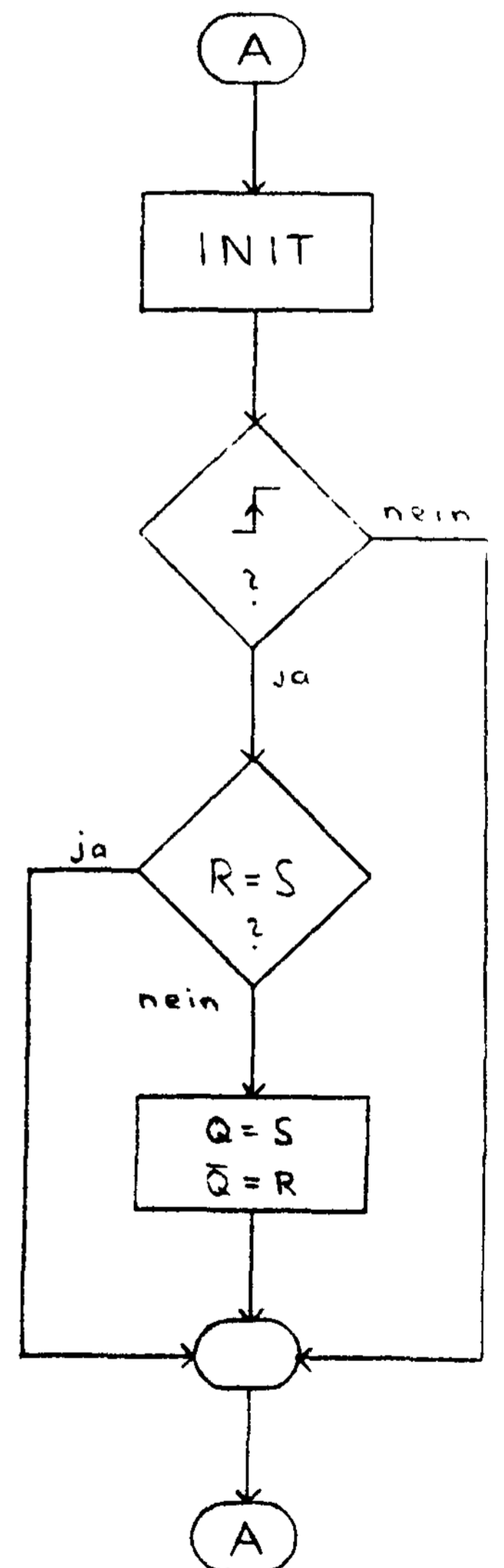
Bei der negativen Taktflanke gilt die Wahrheitstabelle für das ungetaktete RS-Flipflop. Ansonsten gilt der Speicherfall.

Gegenüber dem RS-Flipflop, das bei positiver Taktflanke schaltet, brauchen hier nur wenige Änderungen vorgenommen zu werden:

Flußdiagramm: Aus \uparrow wird \downarrow .

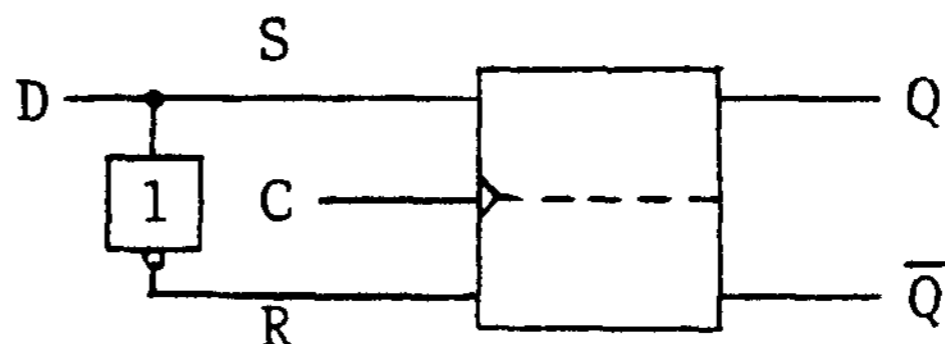
```

Programm:      07 ORC  6
                08 STOC 6
    
```



4. Taktflankengesteuertes D-Flipflop

Bildung eines D-Flipflops aus einem RS-Flipflop:



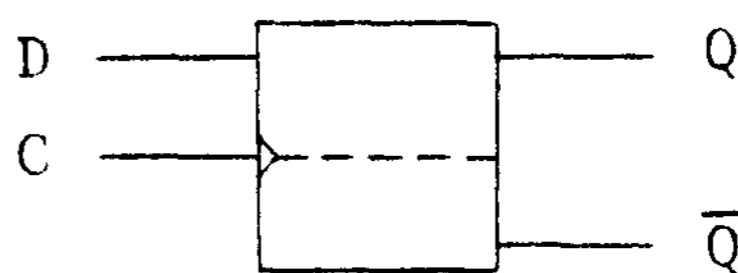
Der Zustand des Eingangs D wird mit der positiven oder mit der negativen Taktflanke des Eingangs C eingelesen und auf den Ausgang Q gesetzt.

Wahrheitstabelle:

D	Q	\bar{Q}
0	0	1
1	1	0

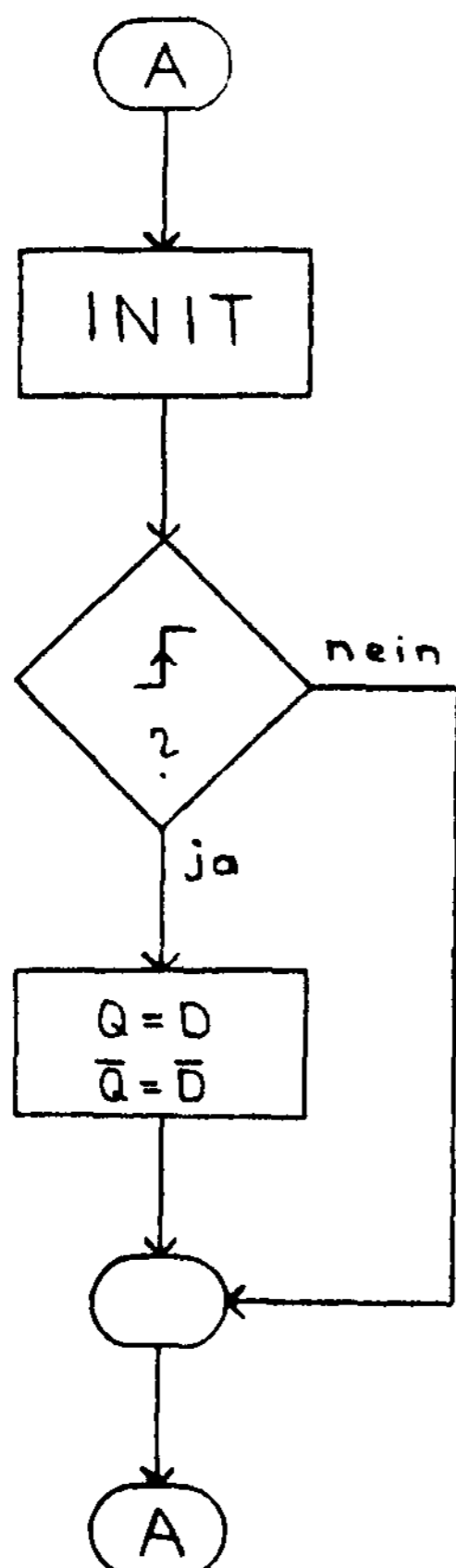
D-Flipflop, das bei positiver Taktflanke schaltet:

Schaltzeichen:



Flußdiagramm nebenstehend!

Festlegung der Zuordnungen im Programm:
C- = E5, C = E3, D = E1, Q = A0, \bar{Q} = A1



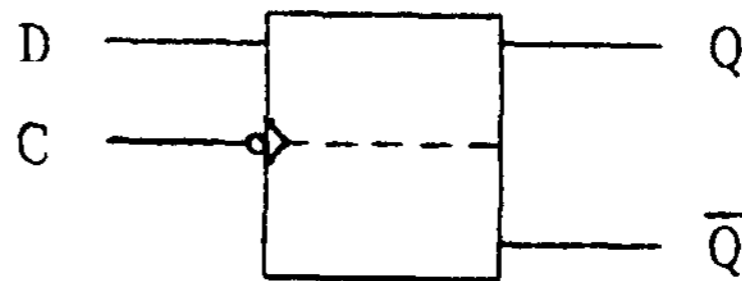
Programm:

```

00 INIT
03 LD 5      C (alt) wird umgeladen
04 STO 6     und zwischengespeichert.
05 LD 3      C (neu) wird geladen
06 STO 5     und zwischengespeichert.
07 ANDC 6    Feststellen, ob der Takt von "0" nach
              "1" geht,
08 OEN 0     nur wenn dies der Fall ist, werden die
              nachfolgenden Ausgabebefehle durch-
              geführt.
09 LD 1      D wird geladen.
10 STO 0     Der Wert von D wird in Q ausgegeben.
11 STOC 1    Der invertierte Wert von Q wird in Q-bar
              ausgegeben.
12 JMP x
  
```

D-Flipflop, das bei negativer Taktflanke schaltet:

Schaltzeichen:



Änderungen gegenüber dem D-Flipflop, das bei positiver Taktflanke schaltet:
 Flußdiagramm: Aus ∇ wird ∇ .

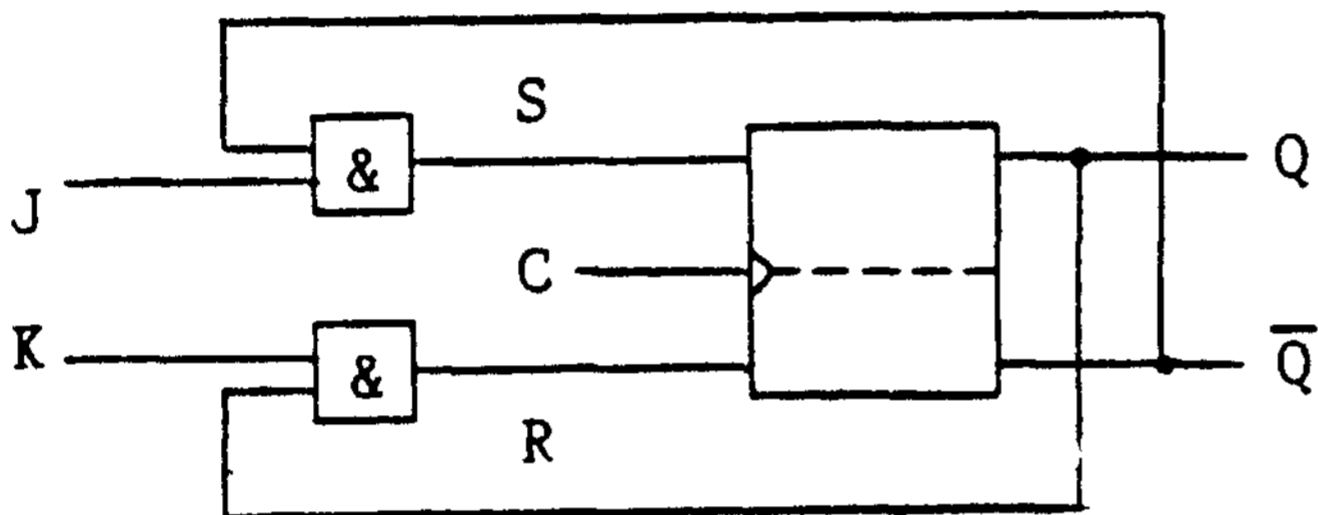
```

    Programm      07 ORC  6
                  08 LDC  0
                  09 OEN  0
                  .
                  .
    
```

Die restlichen Programmschritte rutschen gegenüber dem vorherigen Programm um eine Programmadresse weiter.

5. Taktflankengesteuertes JK-Flipflop

Bildung eines JK-Flipflops aus einem RS-Flipflop:



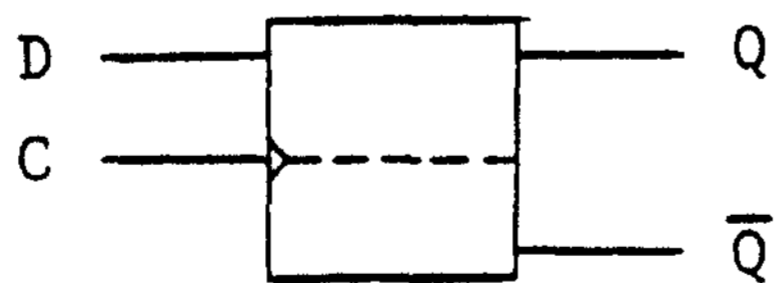
Wahrheitstabelle:

J	K	Q	\bar{Q}
0	0	Q^-	\bar{Q}^-
0	1	0	1
1	0	1	0
1	1	\bar{Q}^-	Q^-

Das Verhalten eines JK-Flipflops läßt sich am besten durch eine logische Gleichung ausdrücken:
 $Q = J \cdot \bar{Q}^- + K \cdot Q^-$

JK-Flipflop, das bei positiver Taktflanke schaltet:

Schaltzeichen:



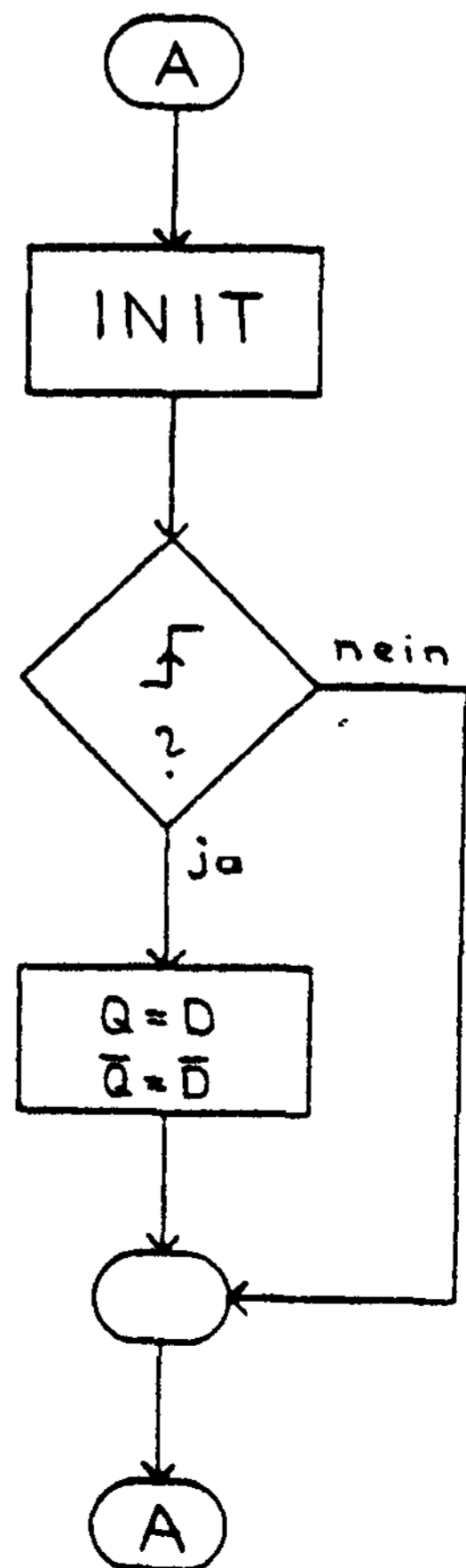
Flußdiagramm nebenstehend!

Festlegung der Zuordnungen im Programm:

$Q^- = E7$, $C^- = E5$, $C = E3$, $K = E2$, $J = E1$, $Q = A0$, $\bar{Q} = A1$

```

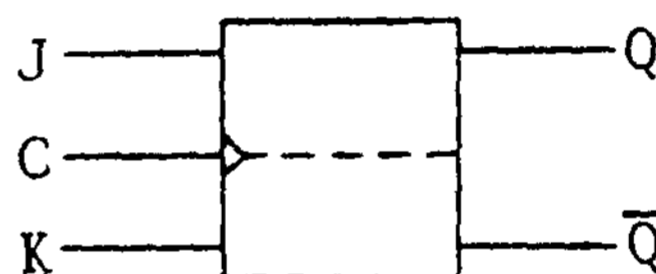
    Programm:
    00 INIT
    03 LD  5
    04 STO 6
    05 LD  3
    06 STO 5
    
```



07	ANDC	6	
08	OEN	0	
09	LD	1	J wird geladen
10	ANDC	7	und mit Q (alt) AND-verknüpft
11	STO	6	Das Ergebnis der Operation wird zwischengespeichert.
12	LDC	2	Das invertierte Signal von K wird geladen
13	AND	7	und mit Q (alt) AND-verknüpft.
14	OR	6	Das Ergebnis der letzten Operation wird mit dem zwischengespeicherten Ergebnis OR-verknüpft. Das Ergebnis ist Q.
15	STO	7	Q wird neu zwischengespeichert.
16	STO	0	Q wird auf Ausgang A0 ausgegeben.
17	STOC	1	Q wird auf Ausgang A1 ausgegeben.
18	JMP	x	

JK-Flipflop, das bei negativer Taktflanke schaltet:

Schaltzeichen:



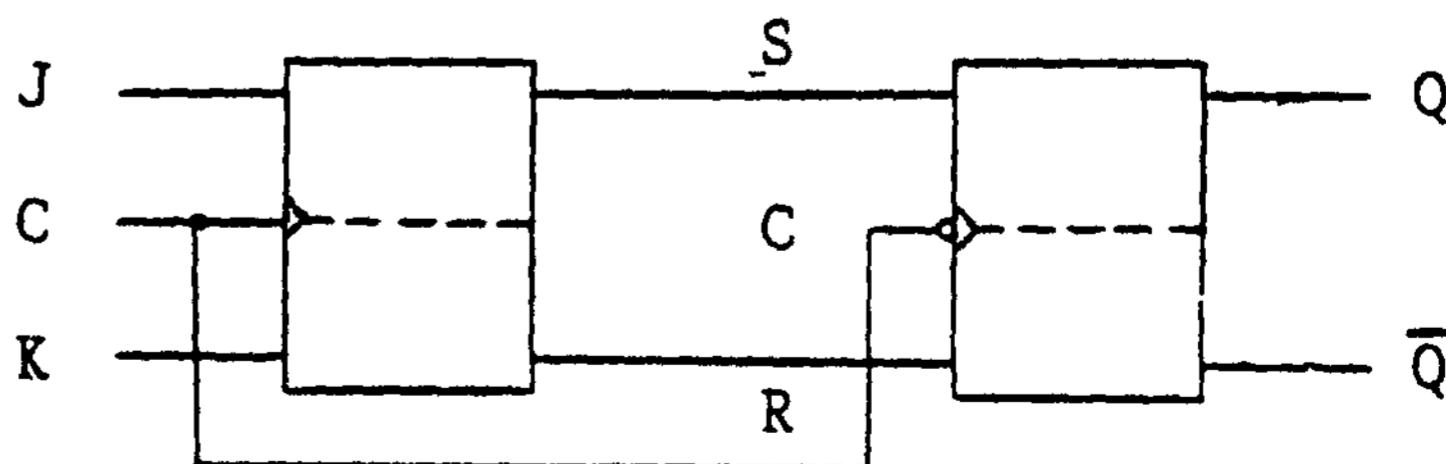
Änderungen gegenüber dem JK-Flipflop, das bei positiver Taktflanke schaltet:
Flußdiagramm: Aus \downarrow wird \uparrow .

Programm:	07	ORC	6
	08	LDC	0
	09	OEN	0

Die restlichen Programmschritte rutschen gegenüber dem vorherigen Programm um eine Programmadresse weiter.

6. JK-Master-Slave-Flipflop

Bildung eines JK-Master-Slave-Flipflops aus einem JK-Flipflop und einem RS-Flipflop:



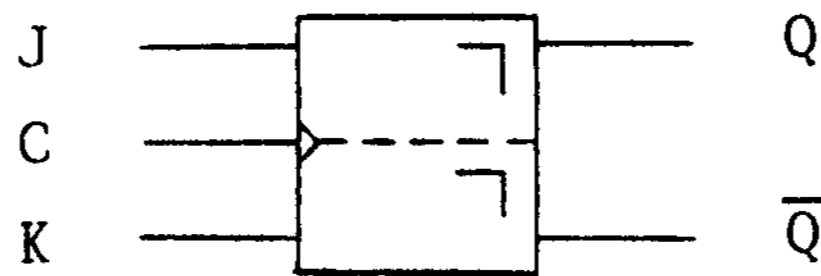
JK-Master-Slave-Flipflops nehmen bei der positiven Taktflanke das Eingangssignal auf. Dieses wird zwischengespeichert und erst bei der negativen Taktflanke zum Ausgang durchgeschaltet. Selbstverständlich gibt es auch JK-Master-Slave-Flipflops, die mit der negativen Taktflanke das erste Flipflop (Master-Flipflop) schalten. Dann schaltet die positive Taktflanke das zweite Flipflop (Slave-Flipflop).

Wahrheitstabelle:

Es gilt die Wahrheitstabelle für das einfache taktflankengesteuerte JK-Flipflop.

JK-Master-Slave-Flipflop, das bei positiver Taktflanke das Master-Flipflop schaltet:

Schaltzeichen:



Flußdiagramm nebenstehend!

Festlegung der Zuordnungen im Programm:

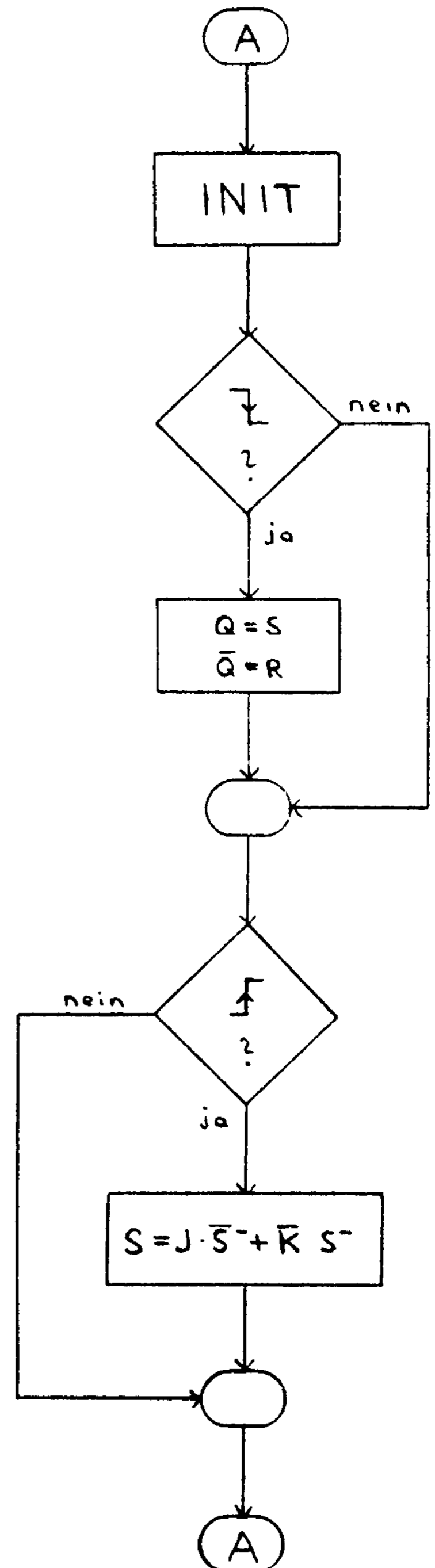
S- = E7, C- = E5, C = E3, K = E2, J = E1, Q = A0, \bar{Q} = A1

Programm:

```

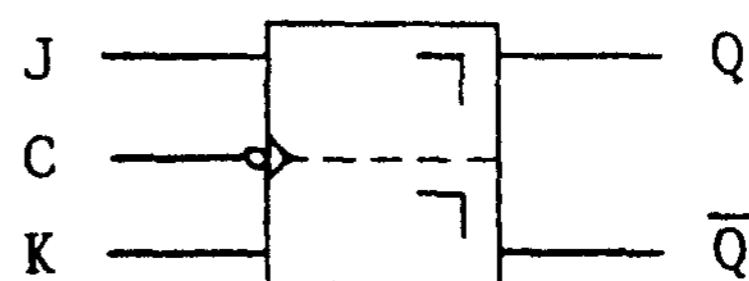
00 INIT
03 LD 5
04 STOC 6
05 LD 3
06 STO 5
07 XNOR 6
08 STO 6
09 ANDC 5
10 OEN 0
11 LD 7
12 STO 0
13 STOC 1
14 LD 6
15 AND 5
16 OEN 0
17 LD 1
18 ANDC 7
19 STO 6
20 LDC 2
21 AND 7
22 OR 6
23 STO 7
24 JMP x

```



JK-Master-Slave-Flipflop, das bei negativer Taktflanke das Master-Flipflop schaltet:

Schaltzeichen:



Änderungen gegenüber dem JK-Master-Slave-Flipflop, das bei positiver Taktflanke das Master-Flipflop schaltet:

Flußdiagramm: Aus ∇ wird ∇ und umgekehrt.

Programm: 09 AND 5
 15 ANDC 5

In diesem Kapitel wurden bisher nur WENN-DANN-Strukturen in den Programmen benutzt. Folgende zwei Programme zur Darstellung von JK-Master-Slave-Flipflops haben eine WENN-DANN-SONST-Struktur und eine SOLANGE-Struktur.

JK-Master-Slave-Flipflop, das bei positiver Taktflanke das Master-Flipflop schaltet:

Flußdiagramm nebenstehend!

Festlegung der Zuordnungen im Programm:

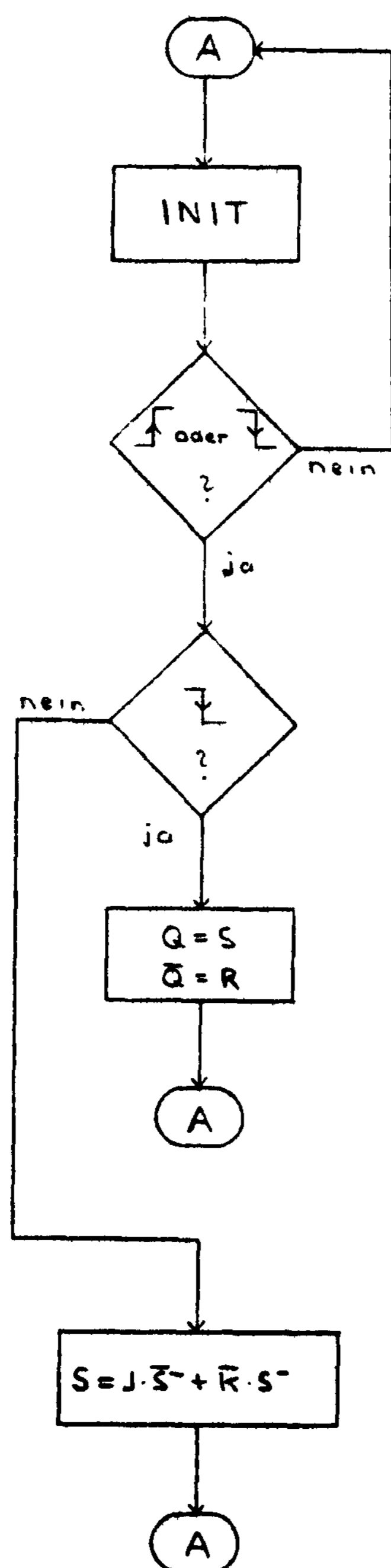
S- = E7, C- = E5, C = E3, K = E2, J = E1, Q = A0, \bar{Q} = A1

Programm:

```

00 INIT
03 LD 5
04 STO 6
05 LD 3
06 STO 5
07 XNOR 6
08 SKZ x
09 JMP x
10 LDC 5
11 OEN 0
12 LD 7
13 STO 0
14 STOC 1
15 OEN 5
16 LD 1
17 ANDC 7
18 STO 6
19 LDC 2
20 AND 7
21 OR 6
22 STO 7
23 JMP x

```



Es muß geprüft werden, ob eine Taktflanke vorliegt. Sind der alte und der neue Taktzustand gleich, so muß die XNOR-Verknüpfung eine "1" ergeben, damit nach SKZ x der Programmschritt JMP x ausgeführt wird. Dann lag nämlich weder eine steigende noch eine fallende Taktflanke vor.

JK-Master-Slave-Flipflop, das bei negativer Taktflanke das Master-Flipflop schaltet:

Änderungen gegenüber dem JK-Master-Slave-Flipflop, das bei positiver Taktflanke das Master-Flipflop schaltet:

Flußdiagramm: Aus \bar{F} wird F .

Programm:	10	OEN	5
	11	LD	7
	12	STO	0
	13	STOC	1
	14	LDC	5
	15	OEN	0

4.5.4. Programme zu arithmetischen Funktionen

Die Entstehung eines Programmes zu arithmetischen Funktionen: Bei den Programmen zu arithmetischen Funktionen ist das Entwickeln von Flußdiagrammen nicht immer zweckmäßig. Stattdessen stellt man meistens mit Hilfe von Wahrheitstabellen logische Gleichungen auf, die man dann zu vereinfachen versucht. Aus diesen logischen Gleichungen kann dann das Programm geschrieben werden. Alle Programme laufen in Stellung "Schnell-Takt" ab.

Der Übersicht wegen soll für die Äquivalenzfunktion ein neues Rechenzeichen eingeführt werden:

$$x_1 * x_2 + \bar{x}_1 * \bar{x}_2 : x_1 \equiv x_2$$

Für die Antivalenzfunktion gilt dann entsprechend:

$$\overline{x_1 \equiv x_2} = \overline{x_1} \equiv x_2$$

Es gelten im übrigen auch für die logischen Funktionen die Rechenregeln der Algebra, wie z.B. das Distributivgesetz:

$$x_1 * (x_2 + x_3) = x_1 * x_2 + x_1 * x_3$$

Sehr wichtig sind die Gesetze von De Morgan:

$$\overline{x_1 * x_2} = \overline{x_1} + \overline{x_2} \quad \text{und} \quad \overline{\overline{x_1} + \overline{x_2}} = \overline{x_1} * \overline{x_2}$$

Diese Gesetze gelten auch für Verknüpfungen von mehr als zwei Variablen.

1. Addierer

Halbaddierer:

Ein Halbaddierer kann zwei Dualziffern (D1 und D2) addieren. Die höherwertige Stelle, d.h., das linke Bit, vom Ergebnis ist der Übertrag (Carry out-C out). Die niederwertige Stelle wird Z genannt.

Wahrheitstabelle:

D2	D1	C out	Z
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Logische Gleichungen:

$$Z = \overline{D1 \oplus D2}$$

$$C\ out = D1 * D2$$

Festlegung der Zuordnungen im Programm:

$$D2 = E2, D1 = E1, Z = A0, C\ out = A1$$

Programm:

```
00 INIT
03 LD 1
04 XNOR 2
05 STOC 0
06 LD 1
07 AND 2
08 STO 1
09 JMP x
```

Volladdierer:

Ein Volladdierer kann drei Dualziffern (D1, D2 und C in) addieren. Bei der Addition von zwei Dualzahlen wird der Übertrag vom letzten Ergebnis (Carry in-C in) mit addiert. Die beiden Stellen vom Ergebnis werden wieder C out (Carry out) und Z genannt.

Durch Verschachteln von mehreren Volladdierern können zwei mehrstellige Dualzahlen addiert werden. Der Carry out eines Volladdierers ist dabei der Carry in für den nächsten Volladdierer. Für das niederwertigste Bit kann auch ein Halbaddierer verwendet werden.

Wahrheitstabelle:

C in	D2	D1	C out	Z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Logische Gleichungen:

$$Z = D1 \oplus D2 \oplus C \text{ in}$$

$$C \text{ out} = D1 * D2 + D1 * C \text{ in} + D2 * C \text{ in} = (D2 + C \text{ in}) * D1 + D2 * C \text{ in}$$

Festlegung der Zuordnungen im Programm:

$$C \text{ in} = E3, D2 = E2, D1 = E1, Z = A0, C \text{ out} = A1$$

Programm:

```

00 INIT
03 LD 1
04 XNOR 2
05 XNOR 3
06 STO 0
07 LD 2
08 OR 3
09 AND 1
10 STO 7
11 LD 2
12 AND 3
13 OR 7
14 STO 1
15 JMP x

```

Mit Hilfe des IEN-Befehls kann dieses Programm auch kürzer geschrieben werden:

```

10 IEN 2
11 OR 3
12 STO 1
13 JMP x

```

Die Programmschritte IEN 2 und OR 3 bewirken folgendes: Der Wert des Eingangs E2 gelangt mit IEN 2 in das IEN-Register des Mikroprozessors und damit an einen Eingang des UND-Gliedes der Eingangsschaltung (Siehe: Kapitel 2.2.1.). Mit OR 3 gelangt der Wert des Eingangs E3 an den anderen Eingang des UND-Gliedes und wird dort mit dem Inhalt des IEN-Registers UND-verknüpft. Das Ergebnis dieser Operation wird mit dem Inhalt des Ergebnisregisters in der Zentralen Logikeinheit ODER-verknüpft. Mit diesen beiden Programmschritten wurde also folgendes durchgeführt:

$$ER = E2 * E3 + ER$$

Da mit IEN 2 der Inhalt des Ergebnisregisters nicht verändert wird, braucht das Ergebnis der letzten Operation (LD 2, OR 3 und AND 1) nicht zwischengespeichert zu werden. Darin liegt auch der große Vorteil des IEN-Befehls. (Für weitere Programmteile muß die Eingabe wieder freigemacht werden.)

Addierer:

Hier wird ein Halbaddierer mit einem Volladdierer verschachtelt. Das folgende Programm ist also eine Kombination der beiden letzten Programme. Es sollen zwei zweistellige Zahlen miteinander addiert werden: $Z = D1 + D2$.

Die beiden höchstwertigen Bits (engl. Most Significant Bit-MSB) liegen an den Eingängen E2 und E4 und die beiden niederwertigsten Bits (engl. Least Significant Bit - LSB) liegen an den Eingängen E1 und E3.

Festlegung der Zuordnungen im Programm:

$D2(\text{MSB}) = E4$, $D2(\text{LSB}) = E3$, $D1(\text{MSB}) = E2$, $D1(\text{LSB}) = E1$, $Z1 = A0$,
 $Z2 = A1$, $C \text{ out} = A2$

Programm:

```
00 INIT
03 LD 1
04 XNOR 3
05 STOC 0
06 LD 1
07 AND 3
08 STO 7
09 XNOR 2
10 XNOR 4
11 STO 1
12 LD 4
13 OR 7
14 AND 2
15 IEN 4
16 OR 7
17 STO 2
18 JMP x
```

2. Umwandlung positiver und negativer Binärzahlen

Im Dualcode gibt es nur positive Zahlen. Um auch negative Zahlen darstellen zu können, brauchen wir einen neuen Code. Der Zweier-Komplement-Code (2KC) ist dabei ein häufig benutzter Code. Dessen positiver Bereich ist gleich dem Dualcode. Die werthöchste Stelle wird als Vorzeichenstelle angesehen. Positive Zahlen sind durch eine 0, negative Zahlen durch eine 1 in der ersten Stelle von links gekennzeichnet. (Computer arbeiten bei der Zahlendarstellung stets mit festgelegter Stellenzahl. Die mögliche werthöchste Stelle ist somit stets bekannt und kann als Vorzeichenstelle verwendet werden, ohne daß Irrtümer entstehen.)

Bei der Umrechnung von positiven in negative Zahlen und umgekehrt braucht nur das Komplement der umzurechnenden Zahl gebildet werden, d.h., die Zahl wird invertiert und mit 1 addiert.

Dezimal	Dual	2KC
7	111	0111
6	110	0110
5	101	0101
4	100	0100
3	011	0011
2	010	0010
1	001	0001
0	000	0000
-1	-	1111
-2	-	1110
-3	-	1101
-4	-	1100
-5	-	1011
-6	-	1010
-7	-	1001
-8	-	1000

Im folgenden Programm sollen positive Zahlen in negative Zahlen und umgekehrt umgewandelt werden. Die umzuwandelnde Zahl liegt an den Eingängen E1 ... E4. Die umgewandelte Zahl wird an den Ausgängen A0 ... A3 angezeigt.

Wahrheitstabelle:

E4	E3	E2	E1	A3	A2	A1	A0
0	1	1	1	1	0	0	1
0	1	1	0	1	0	1	0
0	1	0	1	1	0	1	1
0	1	0	0	1	1	0	0
0	0	1	1	1	1	0	1
0	0	1	0	1	1	1	0
0	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	1
1	1	1	0	0	0	1	0
1	1	0	1	0	0	1	1
1	1	0	0	0	1	0	0
1	0	1	1	0	1	0	1
1	0	1	0	0	1	1	0
1	0	0	1	0	1	1	1
(1	0	0	0	1	0	0	0)

Logische Gleichungen:

$$A0 = E1$$

$$A1 = \overline{E1} \oplus E2 = \overline{E1} \oplus \overline{E2}$$

$$A2 = \overline{E1} * \overline{E2} \oplus E3$$

$$A3 = \overline{E1} * \overline{E2} * \overline{E3} \oplus E4$$

Programm:

```
00 INIT
03 LD 1
04 STO 0
05 XNOR 2
06 STOC 1
```

```

07 LDC 1
08 ANDC 2
09 XNOR 3
10 STO 2
11 LDC 1
12 ANDC 2
13 ANDC 3
14 XNOR 4
15 STO 3
16 JMP x

```

Hat die umzuwandelnde Zahl mehr als vier Stellen, so wird das Verfahren mit einer Wahrheitstabelle unhandlich. Dann empfiehlt es sich, das Programm nach der Umrechnungsvorschrift (Komplementbildung) aufzubauen: Für jede Stelle der umzurechnenden Zahl wird eine Wahrheitstabelle aufgestellt, beginnend mit der niederwertigsten Stelle. Die Wahrheitstabelle für die zweitniederwertigste Stelle kann auch für alle höherwertigen Stellen verwendet werden.

3. Subtrahierer

Da beim Subtrahieren auch negative Zahlen entstehen können, werden die Ergebnisse im Zweier-Komplement-Code dargestellt. Wie beim Addierer gibt es auch hier Halbsubtrahierer und Vollsubtrahierer. Das folgende Programm stellt eine Kombination dieser beiden Schaltungen dar. Es sollen zwei zweistellige Dualzahlen voneinander abgezogen werden:
 $z = D2 - D1$.

Festlegung der Zuordnungen im Programm:

$D2(\text{MSB}) = E4$, $D2(\text{LSB}) = E3$, $D1(\text{MSB}) = E2$, $D1(\text{LSB}) = E1$, $Z1 = A0$, $Z2 = A1$, Vorzeichen = $A2$

Wahrheitstabelle:

	E4	E3	E2	E1	A2	A1	A0
	0	0	0	0	0	0	0
	0	0	0	1	1	1	1
	0	0	1	0	1	1	0
	0	0	1	1	1	0	1
	0	1	0	0	0	0	1
	0	1	0	1	0	0	0
	0	1	1	0	1	1	1
	0	1	1	1	1	1	0
	1	0	0	0	0	1	0
	1	0	0	1	0	0	1
	1	0	1	0	0	0	0
	1	0	1	1	1	1	1
	1	1	0	0	0	1	1
	1	1	0	1	0	1	0
	1	1	1	0	0	0	1
	1	1	1	1	0	0	0

Logische Gleichungen:

$$A0 = \overline{E1} \oplus E3 = \overline{E1} \oplus E3$$

$$A1 = C \text{ in} \oplus E2 \oplus E4$$

$$A2 = C \text{ out} = (C \text{ in} \oplus E2) \oplus E4 \oplus C \text{ in} \oplus E2$$

$$C \text{ out} = E1 \oplus \overline{E3} = 00 \oplus E1$$

$$(C \text{ in} \oplus E1 \oplus \overline{E3})$$

$$(C \text{ in} \oplus E1 \oplus E3)$$

Programm:

```

00 INIT
03 LDC 1
04 XNOR 3
05 STO 0
06 AND 1
07 STO 7
08 XNOR 2
09 XNOR 4
10 STO 1
11 LD 7
12 OR 2
13 ANDC 4
14 IEN 2
15 OR 7
16 STO 2
17 JMP x

```

4. Inkrementieren

Inkrementieren heißt eine 1 addieren.

Die an den Eingängen E1 ... E4 liegende Dualzahl wird mit 1 addiert. Das Ergebnis erscheint an den Ausgängen A0 ... A3. Liegt an den Eingängen eine 15 an, so ist das Ergebnis 16. Da 16 aber nicht dargestellt werden kann, erscheint an den Ausgängen eine 0. Zusätzlich wird dieser Überlauf durch eine "1" am Ausgang A5 angezeigt.

Wahrheitstabelle für die niederwertigste Stelle:

E1	C out	A0
0	0	1
1	1	0

Wahrheitstabelle für die zweitniederwertigste Stelle:

C in	E2	C out	A1
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Für alle höherwertigen Stellen gilt die letzte Wahrheitstabelle.

Logische Gleichungen:

$$\begin{array}{ll}
 A0 = \overline{E1} & C \text{ out} = E1 \\
 A1 = \overline{C \text{ in} \oplus E2} & C \text{ out} = C \text{ in} * E2 \\
 A2 = \overline{C \text{ in} \oplus E3} & C \text{ out} = C \text{ in} * E3 \\
 A3 = \overline{C \text{ in} \oplus E4} & A5 = C \text{ out} = C \text{ in} * E4
 \end{array}$$

Das C in einer Gleichung ist gleich dem C out der vorherigen Zeile.

Programm:

```

00 INIT
03 LD 1
04 STOC 0
05 XNOR 2
06 STOC 1
07 LD 1
08 AND 2
09 STO 7
10 XNOR 3
11 STOC 2
12 LD 7
13 AND 3
14 STO 7
15 XNOR 4
16 STOC 3
17 LD 7
18 AND 4
19 STO 5
20 JMP x

```

5. Dekrementieren

Dekrementieren heißt 1 subtrahieren.

Das Programm zum Dekrementieren läuft im Prinzip wie das Programm zum Inkrementieren ab. Der Überlauf, der bei dem Übergang von 0 auf 15 entsteht, wird wieder durch eine "1" am Ausgang A5 angezeigt.

Wahrheitstabellen:

E1	C out	A0
0	1	1
1	0	0

C in	E2	C out	A1
0	0	0	0
0	1	0	1
1	0	1	1
1	1	0	0

Logische Gleichungen:

$$\begin{array}{ll}
 A0 = \overline{E1} & C \text{ out} = \overline{E1} \\
 A1 = \overline{C \text{ in} \oplus E2} & C \text{ out} = C \text{ in} * \overline{E2} \\
 A2 = \overline{C \text{ in} \oplus E3} & C \text{ out} = C \text{ in} * \overline{E3} \\
 A3 = \overline{C \text{ in} \oplus E4} & A5 = C \text{ out} = C \text{ in} * \overline{E4}
 \end{array}$$

Programm:

```

00 INIT
03 LDC 1
04 STO 0
05 XNOR 2
06 STOC 1
07 LDC 1
08 ANDC 2
09 STO 7
10 XNOR 3
11 STOC 2
12 LD 7
13 ANDC 3
14 STO 7
15 XNOR 4
16 STOC 3
17 LD 7
18 ANDC 4
19 STO 5
20 JMP x
    
```

6. Multiplizierer

Es sollen zwei zweistellige Dualzahlen (D1 und D2) miteinander multipliziert werden. D1 liegt an den Eingängen E1 und E2 und D2 liegt an den Eingängen E3 und E4. Das Ergebnis erscheint an den Ausgängen A0...A3.

Schaltungsprinzip nebenstehend!

Wahrheitstabelle:

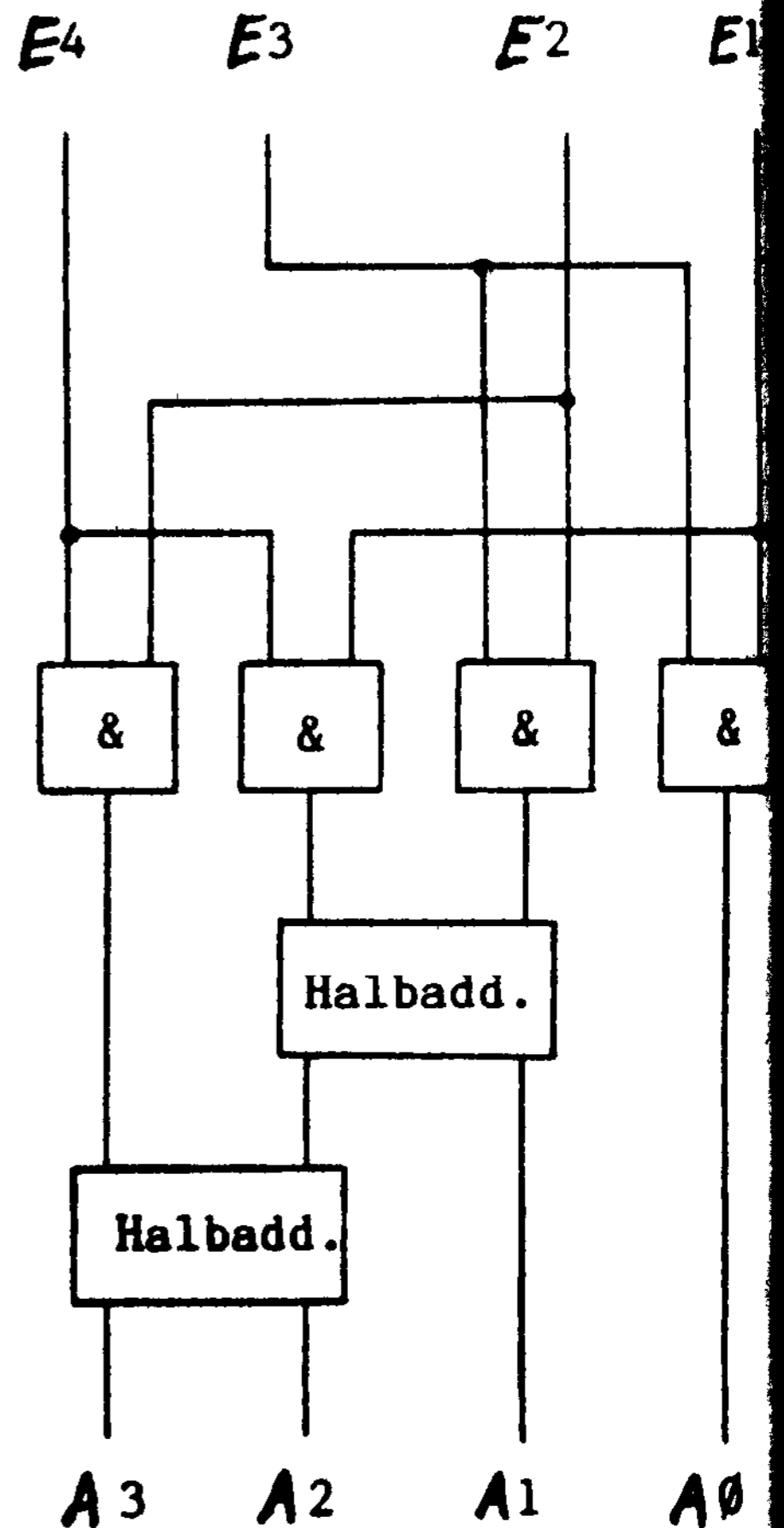
E3	E1	A0
0	0	0
0	1	0
1	0	0
1	1	1

E1*E4	E2*E3	C out	A1
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

C in	E2*E4	A3A	A2
0	0	0	0
0	1	0	1
1	0	-	-
1	1	1	0

Logische Gleichungen:

$$\begin{aligned}
 A0 &= E1 \cdot E3 \\
 A1 &= \overline{E1 \cdot E4} \oplus \overline{E2 \cdot E3} & C\ out &= E1 \cdot E4 \cdot E2 \cdot E3 \\
 A2 &= \overline{C\ in} \cdot E2 \cdot E4 = C\ in \oplus \overline{E2 \cdot E4} & (C\ in &= E1 \cdot E4 \cdot E2 \cdot E3) \\
 A3 &= C\ in \cdot E2 \cdot E4 = C\ in & (C\ in &= E1 \cdot E4 \cdot E2 \cdot E3)
 \end{aligned}$$



Programm:

```

00 INIT
03 LD 1
04 AND 3
05 STO 0
06 LD 1
07 AND 4
08 STO 7
09 LD 2
10 AND 3
11 XNOR 7
12 STOC 1
13 LD 7
14 AND 2
15 AND 3
16 STO 3
17 ORC 2
18 ORC 4
19 STOC 2
20 JMP x

```

7. Paritätsprüfung

Es soll die Parität einer vierstelligen Dualzahl ermittelt werden. Ist die Anzahl der "Einsen", die an den vier Eingängen E1...E4 liegen, gerade, so soll auf dem Ausgang AO eine "0" ausgegeben werden. Ansonsten soll dort eine "1" erscheinen.

Wahrheitstabelle:

E4	E3	E2	E1	AO
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Logische Gleichung:

$$AO = \overline{E1 \oplus E2 \oplus E3 \oplus E4}$$

Programm:

```

00 INIT
03 LD 1
04 XNOR 2
05 XNOR 3
06 XNOR 4
07 STOC 0
08 JMP x

```

8. Komparator

Es sollen zwei zweistellige Dualzahlen (D1 und D2) miteinander verglichen werden. D1 liegt an den Eingängen E1 und E2 und D2 liegt an den Eingängen E3 und E4. Ist D1 größer als D2, so soll das der Ausgang A0 durch eine "1" anzeigen. Ist D1 kleiner als D2, so soll das der Ausgang A1 durch eine "1" anzeigen. Sind D1 und D2 gleich groß, so sollen beide Ausgänge eine "0" anzeigen.

Das Programm beginnt mit der Untersuchung der höherwertigen Stellen E2 und E4. Ist einer der beiden größer als die andere, so brauchen die niederwertigen Stellen E1 und E3 nicht zu untersucht werden. Bei Gleichheit werden E1 und E3 untersucht. Daher eignet sich hier ein Flußdiagramm zur Erstellung des Programmes.

Flußdiagramm:

Programm:

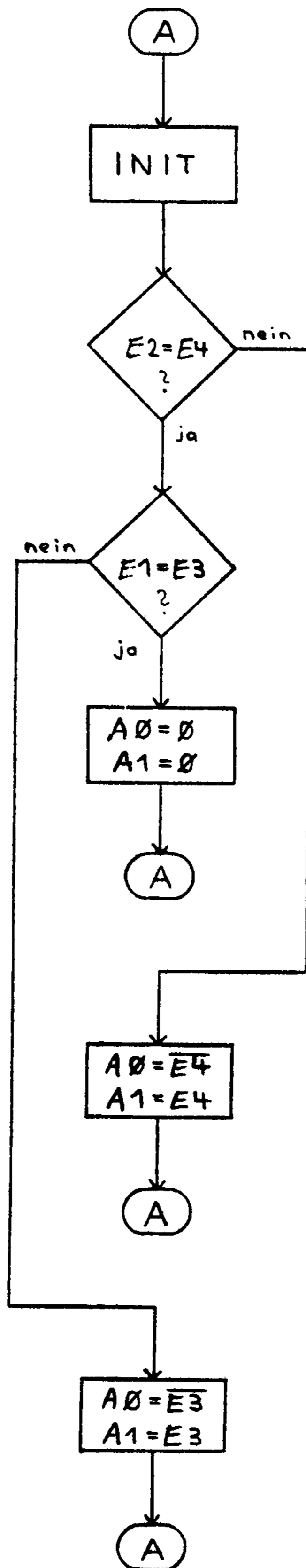
```

00 INIT
03 LDC 4
04 XNOR 2
05 STOC 7
06 OEN 0
07 LD 4
08 STO 1
09 STOC 0
10 OEN 7
11 LD 1
12 ANDC 3
13 STO 0
14 LDC 1
15 AND 3
16 STO 1
17 JMP x

```

9. Umcodierer

Die an den Eingängen E1...E3 liegende Dualzahl soll in die entsprechende Dezimalzahl umcodiert werden. Die Nummer des Ausgangs, an dem eine "1" erscheint, ist gleich dieser Dezimalzahl.



Wahrheitstabelle:

E3	E2	E1	A0	A1	A2	A3	A4	A5	A6	A7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Logische Gleichungen:

$$A0 = \overline{E1} * \overline{E2} * \overline{E3}$$

$$A1 = E1 * \overline{E2} * \overline{E3}$$

$$A2 = \overline{E1} * E2 * \overline{E3}$$

$$A3 = E1 * E2 * \overline{E3}$$

$$A4 = \overline{E1} * \overline{E2} * E3$$

$$A5 = E1 * \overline{E2} * E3$$

$$A6 = \overline{E1} * E2 * E3$$

$$A7 = E1 * E2 * E3$$

Programm:

```

00 INIT
03 LDC 1
04 ANDC 2
05 ANDC 3
06 STO 0
07 LD 1
08 ANDC 2
09 ANDC 3
10 STO 1
11 LDC 1
12 AND 2
13 ANDC 3
14 STO 2
15 LD 1
16 AND 2
17 ANDC 3
18 STO 3
19 LDC 1
20 ANDC 2
21 AND 3
22 STO 4
23 LD 1
24 ANDC 2
25 AND 3
26 STO 5
27 LDC 1
28 AND 2
29 AND 3
30 STO 6
31 LD 1
32 AND 2
33 AND 3
34 STO 7
35 JMP x

```

10. Zähler

Es soll nun ein Programm geschrieben werden, in dem die Anzahl der Eingänge (E1, E2, E3 und E4), die eine "1" anliegen haben, festgestellt werden soll. Die Nummer des Ausgangs, an dem eine "1" erscheint, soll gleich dieser Anzahl sein.

Wahrheitstabelle:	E4	E3	E2	E1	A0	A1	A2	A3	A4
	0	0	0	0	1	0	0	0	0
	0	0	0	1	0	1	0	0	0
	0	0	1	0	0	1	0	0	0
	0	0	1	1	0	0	1	0	0
	0	1	0	0	0	1	0	0	0
	0	1	0	1	0	0	1	0	0
	0	1	1	0	0	0	1	0	0
	0	1	1	1	0	0	0	1	0
	1	0	0	0	0	1	0	0	0
	1	0	0	1	0	0	1	0	0
	1	0	1	0	0	0	1	0	0
	1	0	1	1	0	0	0	1	0
	1	1	0	0	0	0	1	0	0
	1	1	0	1	0	0	0	1	0
	1	1	1	0	0	0	0	1	0
	1	1	1	1	0	0	0	0	1

Logische Gleichungen:

$$A0 = \overline{E1} \cdot \overline{E2} \cdot \overline{E3} \cdot \overline{E4}$$

$$\begin{aligned} A1 &= E1 \cdot \overline{E2} \cdot \overline{E3} \cdot \overline{E4} + \overline{E1} \cdot E2 \cdot \overline{E3} \cdot \overline{E4} + \overline{E1} \cdot \overline{E2} \cdot E3 \cdot \overline{E4} + \overline{E1} \cdot \overline{E2} \cdot \overline{E3} \cdot E4 = \\ &= (E1 \cdot \overline{E2} + \overline{E1} \cdot E2) \cdot \overline{E3} \cdot \overline{E4} + (\overline{E3} \cdot \overline{E4} + \overline{E3} \cdot E4) \cdot \overline{E1} \cdot \overline{E2} = \\ &= (\overline{E1 \oplus E2}) \cdot \overline{E3} \cdot \overline{E4} + (\overline{E3 \oplus E4}) \cdot \overline{E1} \cdot \overline{E2} = \\ &= (\overline{E1 \oplus E2}) + \overline{E3} + \overline{E4} + (\overline{E3 \oplus E4}) + \overline{E1} + \overline{E2} = \\ &= ((E1 \oplus E2) + E3 + E4) \cdot ((E3 \oplus E4) + E1 + E2) \end{aligned}$$

$$\begin{aligned} A2 &= E1 \cdot E2 \cdot \overline{E3} \cdot \overline{E4} + E1 \cdot \overline{E2} \cdot E3 \cdot \overline{E4} + \overline{E1} \cdot E2 \cdot E3 \cdot \overline{E4} + E1 \cdot \overline{E2} \cdot \overline{E3} \cdot E4 + \overline{E1} \cdot E2 \cdot \overline{E3} \cdot E4 + \\ &\quad \overline{E1} \cdot \overline{E2} \cdot E3 \cdot E4 = \\ &= (E2 \cdot \overline{E3} + \overline{E2} \cdot E3) \cdot E1 \cdot \overline{E4} + (E2 \cdot \overline{E4} + \overline{E2} \cdot E4) \cdot \overline{E1} \cdot E3 + (E1 \cdot \overline{E2} + \overline{E1} \cdot E2) \cdot \overline{E3} \cdot E4 = \\ &= (\overline{E2 \oplus E3}) \cdot E1 \cdot \overline{E4} + (\overline{E2 \oplus E4}) \cdot \overline{E1} \cdot E3 + (\overline{E1 \oplus E2}) \cdot \overline{E3} \cdot E4 = \\ &= (\overline{E2 \oplus E3}) + \overline{E1} + \overline{E4} + (\overline{E2 \oplus E4}) + \overline{E1} + \overline{E3} + (\overline{E1 \oplus E2}) + \overline{E3} + \overline{E4} = \\ &= ((E2 \oplus E3) + E1 + E4) \cdot ((E2 \oplus E4) + E1 + E3) \cdot ((E1 \oplus E2) + E3 + E4) \end{aligned}$$

$$\begin{aligned} A3 &= E1 \cdot E2 \cdot E3 \cdot \overline{E4} + E1 \cdot E2 \cdot \overline{E3} \cdot E4 + E1 \cdot \overline{E2} \cdot E3 \cdot E4 + \overline{E1} \cdot E2 \cdot E3 \cdot E4 = \\ &= (E3 \cdot \overline{E4} + \overline{E3} \cdot E4) \cdot E1 \cdot E2 + (E1 \cdot \overline{E2} + \overline{E1} \cdot E2) \cdot E3 \cdot E4 = \\ &= (\overline{E3 \oplus E4}) \cdot E1 \cdot E2 + (\overline{E1 \oplus E2}) \cdot E3 \cdot E4 = \end{aligned}$$

$$A4 = E1 \cdot E2 \cdot E3 \cdot E4$$

Die logischen Gleichungen werden nacheinander in ein Programm umgesetzt. Zwischenergebnisse werden auf dem Ausgang A7 gespeichert. Da die Operation $E1 \oplus E2$ und $E3 \oplus E4$ mehrmals vorkommen, werden ihre Ergebnisse auf den Ausgängen A5 und A6 festgehalten.

Programm:

```
00 INIT ;
03 LDC 1
04 ANDC 2
05 ANDC 3
06 ANDC 4
07 STO 0
08 LD 1
```

```
09 XNOR 2
10 STO 5
11 OR 3
12 OR 4
13 STO 7
14 LD 3
15 XNOR 4
16 STO 6
17 OR 1
18 OR 2
19 AND 7
20 STOC 1
21 LD 2
22 XNOR 3
23 ORC 1
24 OR 4
25 STO 7
26 LD 2
27 XNOR 4
28 OR 1
29 ORC 3
30 AND 7
31 STO 7
32 LD 5
33 OR 3
34 ORC 4
35 AND 7
36 STOC 2
37 LDC 6
38 AND 1
39 AND 2
40 STO 7
41 LDC 5
42 AND 3
43 AND 4
44 OR 7
45 STO 3
46 LD 1
47 AND 2
48 AND 3
49 AND 4
50 STO 4
51 JMP x
```

4.5.5. Simulationsprogramme

In diesem Kapitel sollen weitere Programme wie z.B. Schaltungen behandelt werden.

1. Wechselschaltung

Eine Wechselschaltung ist eine Schaltung mit zwei Schaltern (S1 und S2). Wird ein Schalter betätigt, so ändert sich der Ausgangszustand Z der Schaltung. Der Anfangszustand ist willkürlich gewählt.

Festlegung der Zuordnungen im Programm:

S2 = E2, S1 = E1, Z = A0

Wahrheitstabelle:

E2	E1	A0
0	0	0
0	1	1
1	1	0
1	0	1

Logische Gleichung:

$$A0 = \overline{E1 \oplus E2}$$

Programm:

```
00 INIT
03 LD 1
04 XNOR 2
05 STOC 0
06 JMP x
```

Stellung: Schnell-Takt

2. Kreuzschaltung

Eine Kreuzschaltung ist eine Schaltung mit mehreren Schaltern (hier drei: S1, S2 und S3). Wird ein Schalter betätigt, so ändert sich der Ausgangszustand Z der Schaltung. Der Anfangszustand ist wieder willkürlich gewählt.

Festlegung der Zuordnungen im Programm:

S3 = E3, S2 = E2, S1 = E1, Z = A0

Wahrheitstabelle:

E3	E2	E1	A0
0	0	0	0
0	0	1	1
0	1	1	0
0	1	0	1
1	1	0	0
1	1	1	1
1	0	1	0
1	0	0	1

Logische Gleichung:

$$A0 = E1 \oplus E2 \oplus E3$$

Programm:

```
00 INIT
03 LD 1
04 XNOR 2
05 XNOR 3
06 STO 0
07 JMP x
```

Stellung: Schnell-Takt

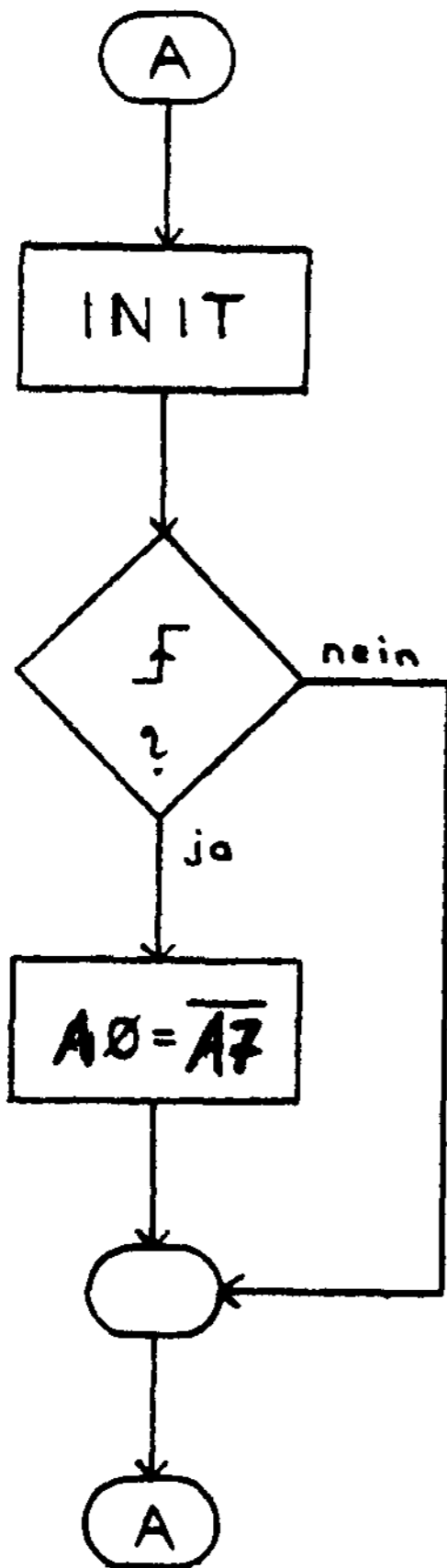
3. Tasterschaltung

Die Tasterschaltung ist eine Schaltung mit mehreren Tastern (hier vier: T1, T2, T3 und T4). Wird ein Taster kurz geschlossen, so ändert sich der Ausgangszustand Z der Schaltung.

Festlegung der Zuordnungen im Programm:

T4 = E4, T3 = E3, T2 = E2, T1 = E1, Z = A0

Flußdiagramm nebenstehend!



Programm:

```

00 INIT
03 LD 5
04 STO 6
05 LD 1
06 OR 2
07 OR 3
08 OR 4
09 STO 5
10 ANDC 6
11 OEN 0
12 LD 7
13 STOC 7
14 STOC 0
15 JMP x
  
```

Stellung: Schnell-Takt

4. Zwei-aus-drei-Schaltung

Zwei-aus-drei-Schaltungen werden bei mit Risiken behafteten Anlagen verwendet. Eine Abschaltung der Anlage soll nur dann erfolgen, wenn mindestens zwei der drei Gefahrenmelder (G1, G2 und G3) die Gefahr anzeigen, da Gefahrenmelder auch defekt sein können. Die Gefahrenmelder geben bei Gefahr eine "1". Die Abschaltung der Anlage soll erfolgen, wenn am Ausgang Z der Zwei-aus-drei-Schaltung eine "1" erscheint.

Festlegung der Zuordnungen im Programm:

G3 = E3, G2 = E2, G1 = E1, Z = A0

Wahrheitstabelle:

E3	E2	E1	A0
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Logische Gleichungen:

$$A0 = (E1+E2) \cdot E3 + E1 \cdot E2$$

Programm:

```
00 INIT
03 LD 1
04 OR 2
05 AND 3
06 IEN 1
07 OR 2
08 STO 0
09 JMP x
```

Stellung: Schnell-Takt

5. Multivibrator

Flipflops sind bistabile Kippstufen, d.h., sie haben zwei stabile Ausgangszustände (Q und \bar{Q}). Multivibratoren sind astabile Kippstufen, d.h., sie haben keinen stabilen Ausgangszustand. Sie kippen von einem nichtstabilen Zustand in den anderen nichtstabilen Zustand und wieder zurück. Zum Kippen ist kein von außen kommendes Signal erforderlich.

Festlegung der Zuordnungen im Programm:

$$Q = A0, \bar{Q} = A1$$

Programm:

```
00 INIT
03 LD /
04 STOC 7
05 STOC 0
06 STO 1
07 NOP x
.
.
.
yy NOP x
yy+1 JMP x
```

yy ist eine beliebige Programmadresse. $yy+1$ ist die auf yy folgende Programmadresse.

Durch die Anzahl der Leerbefehle NOP kann die Frequenz variiert werden. Bei hohen Frequenzen ist die Stellung "Schnell-Takt" zu wählen und bei niedrigen Frequenzen ist die Stellung "Langsam-Takt" zu wählen, wobei im letzteren Fall die Frequenz noch zusätzlich durch das Potentiometer verändert werden kann. Das Ausgangssignal ist ein Rechtecksignal, da die beiden Zustände $Q = "0"$ und $Q = "1"$ gleich lange dauern (Siehe: Kapitel 4.3.2.).

6. Monoflop

Monoflops sind monostabile Kippstufen, d.h., sie haben einen stabilen und einen nichtstabilen Ausgangszustand. Ein Kippen der Schaltung in den nichtstabilen Zustand ist nur durch ein von außen zugeführtes Signal S möglich. Im stabilen Zustand liegt am Ausgang Q eine "0". Im nichtstabilen Zustand liegt dort eine "1".

Festlegung der Zuordnungen im Programm:

$S = E1$, $Q = A0$, $\bar{Q} = A1$

Programm:

```

00 INIT
03 LD 5
04 STO 6
05 LDC 1
06 STO 5
07 QRC 6
08 SKZ x
09 JMP x
10 STOC 0
11 STO 1
12 NOP x
.
.
.
yy NOP x
yy+1 STO 0
yy+2 STOC 1
yy+3 JMP x

```

Durch die Anzahl der Leerbefehle NOP kann die Dauer des nichtstabilen Zustandes variiert werden. Danach kippt die Schaltung selbständig in den stabilen Zustand zurück. Es kann "Schnell-Takt" oder "Langsam-Takt" gewählt werden.

7. Elektronischer Würfel

Es soll ein elektronischer Würfel dargestellt werden. Ein Schalter soll den Würfelvorgang ersetzen. Die Nummer des Ausgangs, an dem beim Betätigen des Schalters (E1) eine "1" erscheint, ist gleich der "gewürfelten" Zahl. Zunächst wird ein Zähler gebildet, der immer von 1 bis 6 zählt. Wird dann der Eingang E1 auf "1" gesetzt, so soll der derzeitige Zählerstand erhalten bleiben. Damit alle Ausgangszustände, d.h. alle Zahlen, gleich wahrscheinlich sind, muß jeder Zählerstand gleich lange an den Ausgängen stehen.

Programm:

```

00 INIT
03 LD 1
04 SKZ x
05 JMP x
06 STOC 1
07 STO 6
08 NOP x
09 NOP x
10 NOP x
11 NOP x
12 LD 1
13 SKZ x

```

14 JMP x
15 STOC 2
16 STO 1
17 NOP x
18 NOP x
19 NOP x
20 NOP x
21 LD 1
22 SKZ x
23 JMP x
24 STOC 3
25 STO 2
26 NOP x
27 NOP x
28 NOP x
29 NOP x
30 LD 1
31 SKZ x
32 JMP x
33 STOC 4
34 STO 3
35 NOP x
36 NOP x
37 NOP x
38 NOP x
39 LD 1
40 SKZ x
41 JMP x
42 STOC 5
43 STO 4
44 NOP x
45 NOP x
46 NOP x
47 NOP x
48 LD 1
49 SKZ x
50 JMP x
51 STOC 6
52 STO 5
53 JMP x

Stellung: Schnell-Takt

8. Fließbandsteuerung

Der Antrieb eines Fließbandes soll so gesteuert werden, daß Pakete, die auf diesem Fließband laufen, zu einer Entnahmestelle gebracht werden. Die Entnahmestelle befindet sich auf dem Fließband zwischen zwei Lichtschranken (L1 und L2). Die Steuerung des Fließbandes geschieht mit Hilfe dieser beiden Lichtschranken und einem Entnahmeggerät. Der Abstand der beiden Lichtschranken voneinander ist etwas größer als die größtmögliche Breite eines Paketes. Normalerweise läuft das Fließband schnell vorwärts bis ein Paket die erste Lichtschranke unterbricht. Dann läuft das Fließband langsam vorwärts. Ist die Lichtschranke wieder geschlossen, so stoppt das Fließband und das Entnahmeggerät bekommt einen Impuls zum Entnehmen des Paketes vom Fließband. Ist das Paket richtig entnommen, so wird das Fließband durch einen Impuls vom Entnahmeggerät auf "Schnell-

Vorwärts" gestellt. Ist jedoch das Paket wieder auf das Fließband gefallen bzw. gar nicht entnommen worden, so muß es wieder an die Entnahmestelle gebracht werden, wenn eine der beiden Lichtschranken unterbrochen wurde. Die erste Lichtschranke (L1) steuert das Paket langsam vorwärts zur Entnahmestelle und die zweite Lichtschranke (L2) steuert das Paket langsam rückwärts.

Ist eine Lichtschranke geschlossen, entspricht das einer "1".

Ist eine Lichtschranke unterbrochen, entspricht das einer "0". Impulse werden durch eine "1" dargestellt .

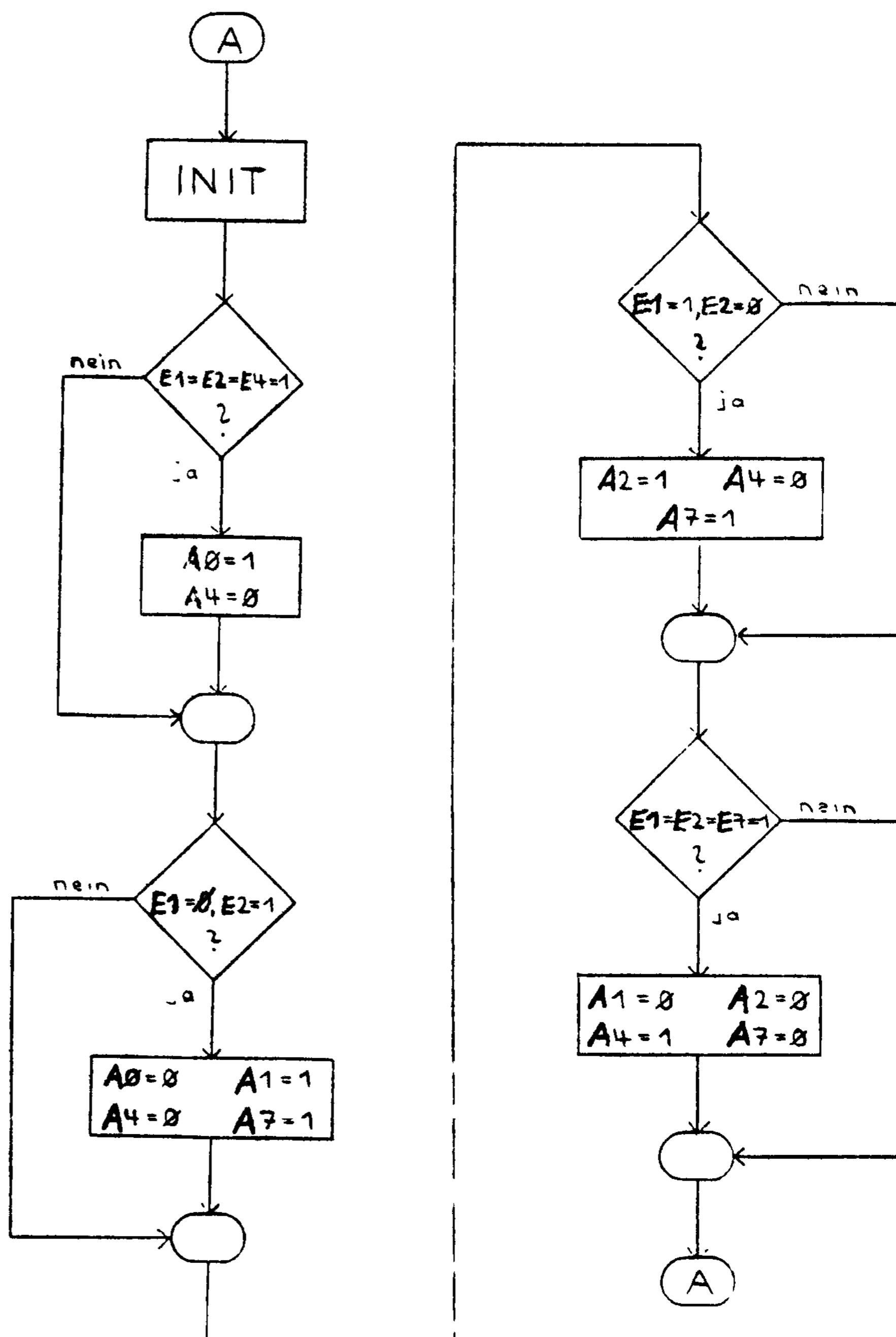
Festlegung der Zuordnungen im Programm:

Impuls vom Entnahmegerät = E4, Lichtschranke 2 (L2) = E2, Lichtschranke 1 (L1) = E1

Fließband:

Schnell-Vorwärts = A0, Langsam-Vorwärts = A1, Langsam-Rückwärts = A2, Impuls für Entnahmegerät = A4

Flußdiagramm:



Programm:

```
00 INIT
03 LD 1
04 AND 2
05 AND 3
06 OEN 0
07 STOC 4
08 STO 0
09 LD 2
10 ANDC 1
11 OEN 0
12 STOC 0
13 STOC 4
14 STO 1
15 STO 7
16 LD 1
17 ANDC 2
18 OEN 0
19 STOC 4
20 STO 2
21 STO 7
22 LD 7
23 AND 1
24 AND 2
25 OEN 0
26 STOC 1
27 STOC 2
28 STO 4
29 STOC 7
30 JMP x
```

Stellung: Schnell-Takt

4.6. PROGRAMME ZUR STEUERUNG ANGESCHLOSSENER PERIPHERIE

In Kapitel 4.5.5. haben wir unter anderem eine Fließbandsteuerung simuliert. Wir haben also kein Fließband angeschlossen, sondern nur die Steuerungsvorgänge an den Leuchtdioden beobachtet.

In diesem Kapitel wollen wir nun z. B. einen Motor an den Computer anschließen und die Steuerungsvorgänge direkt beobachten. Geräte, die an einen Computer angeschlossen werden, nennt man Peripherie.

Wie schon vorher dargelegt, ist es eines der wesentlichen Kennzeichen der dritten industriellen Revolution, daß Computer die Steuerung von Maschinen übernehmen. Genau das kann auch unser Computer. Grundlage für den Antrieb von Robotern sind Elektromotoren, genauer gesagt, Schrittmotoren. Beginnen wir daher mit der Steuerung eines Elektromotors.

Alle Elektromotoren müssen mit weit höheren elektrischen Strömen betrieben werden, als unser Computer sie an seinen Ausgängen liefert. Daher schalten wir einen oder auch mehrere Transistorverstärker, ein sogenanntes Interface, zwischen Computer und Motor. Das Interface, an dem der Motor angeschlossen ist, wird mit dem Expansionsstecker verbunden.

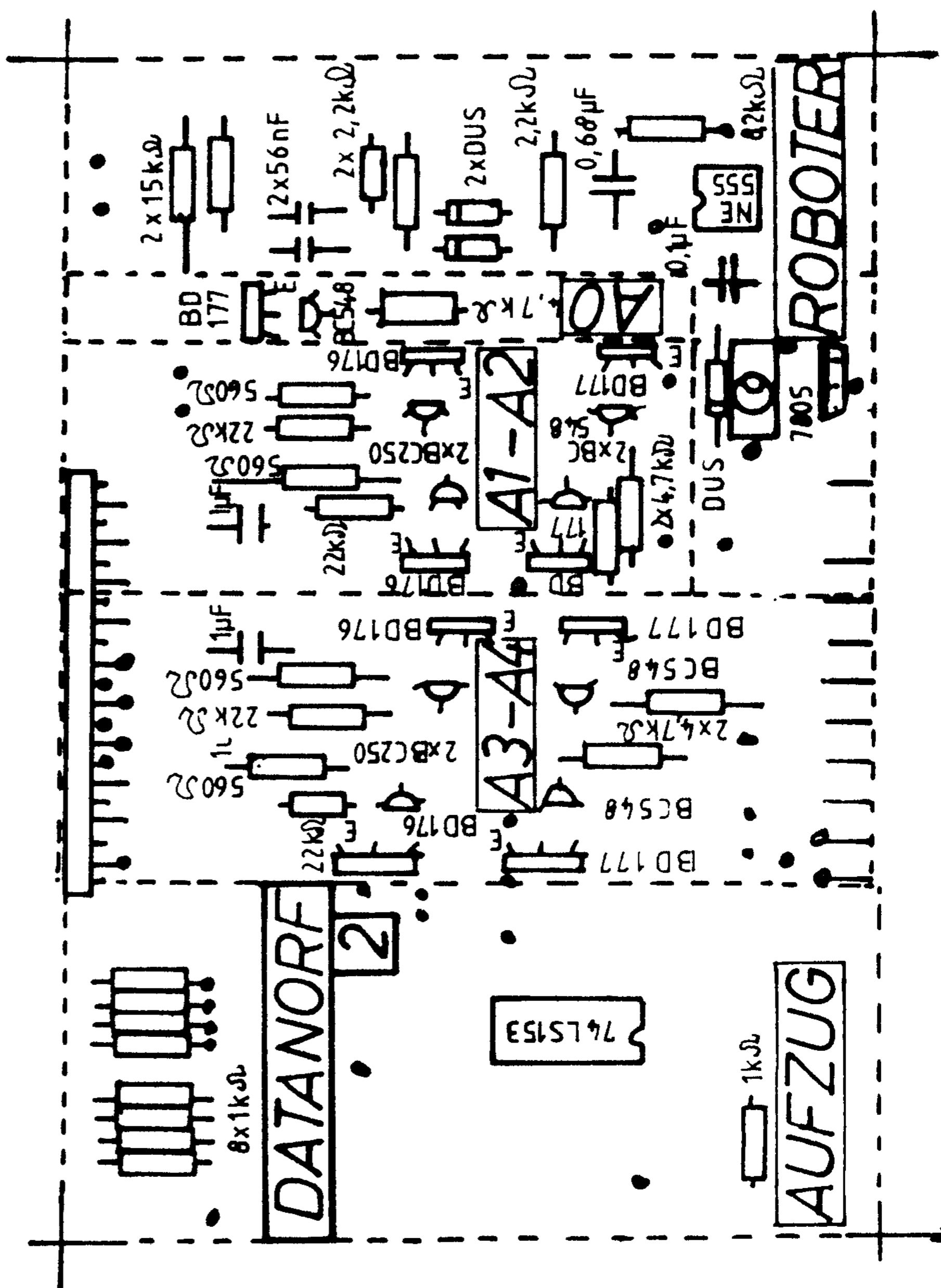


Abb. 42: Verstärker für den Anschluß an den WDR-1-Bit Computer

Um möglichst vielseitig zu sein, benutzen wir ein speziell für den WDR-1-Bit-Computer entwickeltes Interface für Elektromotoren, das DATANorf-Interface (vergl. Bezugsquellen). Dieses Interface bietet den Vorteil, daß es je nach Bedarf ausgebaut bzw. erweitert werden kann. In der ersten Ausbaustufe "das Schalten eines Gerätes" wird das Feld A0 bestückt. Die zweite Ausbaustufe, Feld A1/A2 ist schon leistungsfähiger. Beispielsweise kann damit ein Motor umgepolt werden. Insgesamt können wir mit den vier vorhandenen Ausbaustufen alle Geräte aus dem Fischertechnik computing- Baukasten steuern.

4.6.1. MOTORSTEUERUNG

Ziel ist es zunächst, eine Radarantenne zu steuern. Eine Radarantenne dreht sich ein Stück in eine bestimmte Richtung, stoppt dann und dreht sich ein Stück in die entgegengesetzte Richtung. Dazu müssen wir einen Elektromotor ein- und ausschalten, sowie umpolen. Umpolen heißt, wir machen aus einer Linksdrehung des Motors eine Rechtsdrehung, bzw. umgekehrt. Die Radaranlage bauen wir aus Legobauteilen (Siehe Abb. 43) auf.

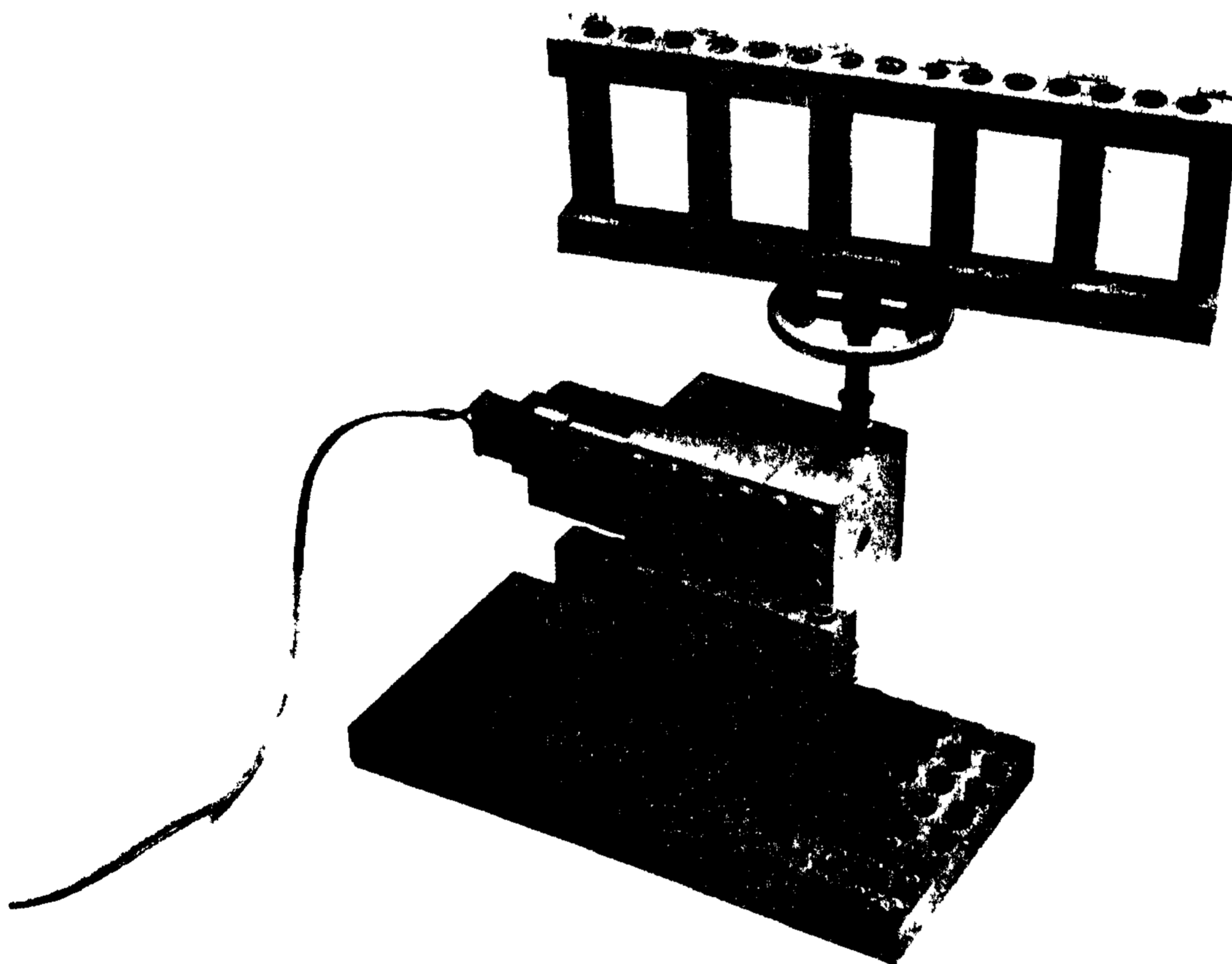


Abb. 43 Radarantenne

Den Motor verdrahten wir mit dem Ausgang A0 des Interfaces. Zunächst schalten wir den Motor an und aus.

**** Motor schalten ****

*** Betriebsart: Handtakt ***

00 INIT

03 STO 0 Motor wird eingeschaltet

04 STOC 0 Motor wird ausgeschaltet

05 JMP x

Soll das Programm automatisch ablaufen, stellen wir den Taktschalter auf Langsamtakt und fügen "Verzögerungen" ein, da der Motor sonst den schnellen Befehlsänderungen nicht folgen kann:

```

** Motor schalten **
* Betriebsart: Langsamtakt *
00  INIT
03  STO  0    Motor wird eingeschaltet
04  NOP  0    Bei diesem Schritt verändert sich nichts: es
                entsteht eine Pause.
zz  NOP  0
zz+1 STOC 0   Motor wird ausgeschaltet
zz+2 JMP  x

```

(Es ist darauf zu achten, daß die Anzahl der Programmschritte hier zz+2, 256 nicht überschreitet)

Mit Hilfe des nächsten Programms können wir den Motor umpolen. Da das Feld A0 des Interfaces einen Motor oder Elektromagne nur ein- bzw. ausschalten kann, verdrahten wir nun den Motor mit den Ausgängen des Feldes A1-A2. Damit sind die Zuordnungen im Programm festgelegt:

Motor linksdrehen = A1, Motor rechtsdrehen = A2.

```

** Motor umpolen **
* Betriebsart: Langsamtakt *
00  INIT
03  STO  1    Motor linksdrehen an
04  STOC 1    Motor aus
05  STO  2    Motor rechtsdrehen an
06  STOC 2    Motor aus
07  JMP  x

```

(Sollte die Drehrichtung des Motors nicht mit dem Programm übereinstimmen, muß der Motor umgepolt werden.)

In dieses Programm können natürlich wieder "Verzögerungen" eingebracht werden, so daß eine Radarantenne richtig simuliert wird. Außerdem ist darauf zu achten, daß der Motor vor dem Umschalten durch das Programm ausgeschaltet wird. Andernfalls zerstören wir die Transistoren unseres Interfaces.

4.6.2. SORTIERANLAGE

Eine Anlage, die aus Fischertechnikbauteilen zusammensetzt ist (Siehe: Abb. 44), soll "Werkstücke" mit zwei verschiedenen Längen sortieren. Sortiert wird, indem ein Schlitten mit Hilfe eines Motors bewegt wird.

Die Längenabfrage geschieht mittels zwei parallel geschalteter Schalter S1 und S2 mit einem gemeinsamen Ausgang. Der Ausgang ist über diese Schalter mit 5 V verbunden. In Ruhestellung liegt der Ausgang auf "1", da beide Schalter in diesem Zustand

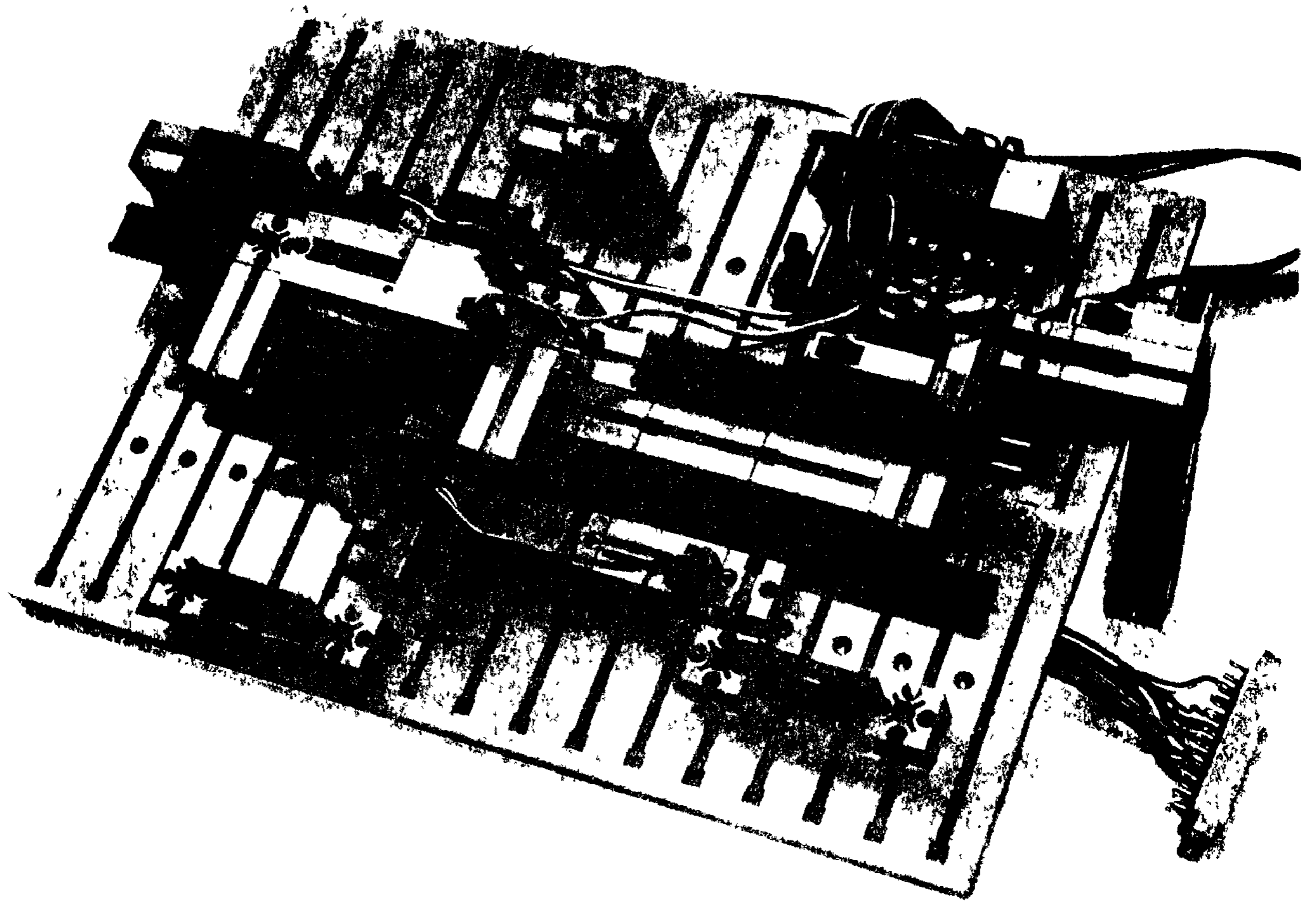


Abb. 44: Sortieranlage

geschlossen sind. Wird einer von ihnen geöffnet, ändert sich am Ausgang zunächst nichts, da sie im Sinne einer ODER-Schaltung miteinander verdrahtet sind. Damit am Ausgang eine "0" anliegt, wenn beide Schalter geöffnet sind, muß der Ausgang noch über einen Pull down Widerstand von der Größe 1 Kiloohm mit Masse verbunden sein. Der Ausgang dieser Schaltung ist mit dem Eingang E3 des Computers verbunden. Abb. 45 zeigt die Schaltung für die Längenabfrage.

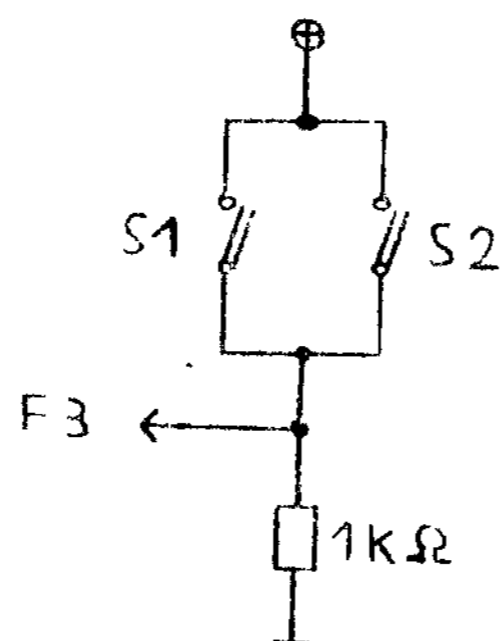


Abb. 45: Schaltung für die Längenabfrage

Die Längenabfrage erkennt einen langen Baustein daran, daß beim Transport des Bausteins beide Schalter betätigt werden, also eine "0" anliegt. Beim Transport eines kurzen Bausteins ist immer einer der beiden Schalter geöffnet, es liegt also eine "1" an.

Hier die Festlegung der Zuordnungen für die Abfrage des Schlittens im Programm:

Start des Schlittens und stoppen rechts = E4, Längenabfrage = E3, Stoppen des Schlittens links = E2, Schalter zum Abschalten = E1,

Motor linksdrehen = A1, Motor rechtsdrehen = A2.

**** Sortieranlage ****

*** Definition der Prozeduren ***

* In Prozedur WARTEN	sind	E/A6 = 0 und E/A7 = 0
* In Prozedur TESTEN	sind	E/A6 = 1 und E/A7 = 0
* In Prozedur LÄNGE ERKENNEN	sind	E/A6 = 0 und E/A7 = 1
* In Prozedur ZURÜCKFAHREN	sind	E/A6 = 1 und E/A7 = 1

*** Werkstücke sortieren ***

00 INIT

*** Prozedur WARTEN**

03 LDC 7

04 ANDC 6

05 LD 4

06 STO 1

07 STO 6

*** In Prozedur TESTEN**

08 LDC 7

09 AND 6

10 ANDC 3

11 OEN 0

12 STOC 1

13 STO 2

14 STOC 6

15 STO 7

16 LD 2

*** In Prozedur LÄNGE ERKENNEN**

17 AND 6

18 ANDC 7

19 OEN 0

20 STOC 1

21 STO 2

22 STO 6

23 STO 7

*** In Prozedur ZURÜCKFAHREN**

24 LDC 6

25 AND 7

26 AND 4

27 OEN 0

28 STOC 2

29 STO 1

30 STO 6

31 LD 7

32 AND 6

33 AND 1

34 OEN 0

35 STOC 1
 36 STOC 2
 37 STOC 6
 38 STOC 7
 39 JMP x

4.6.3. STEUERUNG EINES ROBOTERS

Bauen wir unsere Sortieranlage zu einem "Teach-in"-Roboter um. Er soll lernen, Werkstücke an einem bestimmten Platz aufzunehmen (Materialeinkauf), sie zu einer anderen Stelle hinzutransportieren (Produktion) und sie von hier an eine dritte Stelle zu bringen (Versand).

Der Teach-in-Roboter hat für diese Aufgabenstellung zwei Motoren, einen Elektromagneten und zwei Potentiometer. Gegenüber dem von Fischertechnik angegebenen Aufbauvorschlag haben wir ihn jedoch leicht verändert: Wir haben die beiden Potentiometerabfragen weggelassen, da unser Computer nicht in der Lage ist, die Ergebnisse von Analog-Digital-Wandlern zu verarbeiten. Wir müssen daher einen anderen Weg gehen. Für die Abfrage des Drehwinkels setzen wir auf die Schneckenwelle eine Seiltrommel mit Doppelnocke, die einen darunter montierten Schalter betätigt. Um die Lage des Schlittens abzufragen, fügen wir unter die Aluminiumschiene 10 Winkelstücke ein (vgl. Bezugsquellen), die wiederum einen Schalter betätigen und so die Lage des Schlittens dem Computer mitteilen. Beiden Abfragen ist gemeinsam, daß sie nicht mehr den Anfangs- bzw. Endzustand des zu bewegendem Teils registrieren, sondern daß die Anzahl der Schalterbetätigungen abgezählt wird. Allerdings besitzt der WDR-1-Bit-Computer hierfür keinen speziellen Zähler. Sein einziger Zähler ist der Programmzähler, der für diese Vorgänge mitbenutzt wird, wollen wir keinen zusätzlichen Zähler aufbauen. Es reicht aus, das DATANorf-Interface im Feld "Robot" um einen Timer NE 555 mit dazugehörigen passiven Bauteilen zu ergänzen. Der NE 555 ist hierbei so beschaltet, daß er die beiden Schalter des Roboters entprellt und so nur Einzelimpulse an den Eingang des Computerprogrammzählers gelangen. Wenn wir den Programmzähler als Impulzzähler verwenden wollen, müssen wir dafür sorgen, daß der automatisch ablaufende Langsam-Takt des Computers abgestellt wird. Dies erreichen wir, indem wir über den Anschluß 20 des Expansionssteckers an den Pin 4 des Taktgenerators eine "0" legen. Dies geschieht automatisch durch Anstecken des DATANorf-Interfaces, das eine Verbindung zum Ausgang A5 des Computers herstellt.

Nach der Initialisierung muß dann zuerst der Langsam-Takt wieder freigegeben werden. Dies geschieht mit dem Befehl STO 5. Jetzt arbeitet der Computer wieder wie gewohnt.

Hier das Prinzip des TEACH-IN: Mit STOC 5 stellen wir den Langsamtakt ab. Jeder Zählimpuls aus einem der beiden Schalter gelangt nun über den Anschluß 18 des Expansionssteckers an den Prozessor und damit an seinen Programmzähler. Jedem Zählimpuls wird ein NOP-Befehl zugeordnet. Soll dieser Vorgang beendet werden, geben wir STO 5 ein. Auf diese Art und Weise läßt sich jeder "Koordinatenpunkt" im Bewegungsraum des Elektromagneten erreichen, wir müssen nur die entsprechenden Programmschritte eingeben.

Das folgende Programm veranlaßt den Roboter zwei verschieden große Metallstücke mit Hilfe des Elektromagneten zu versetzen. Damit wir den Roboter leichter in eine durch ein Programm definierte Ausgangslage bringen können (in unserem Beispiel ist die Startposition links unten), enthält er noch zwei weitere Schalter, die die beide Motoren ausschalten können.

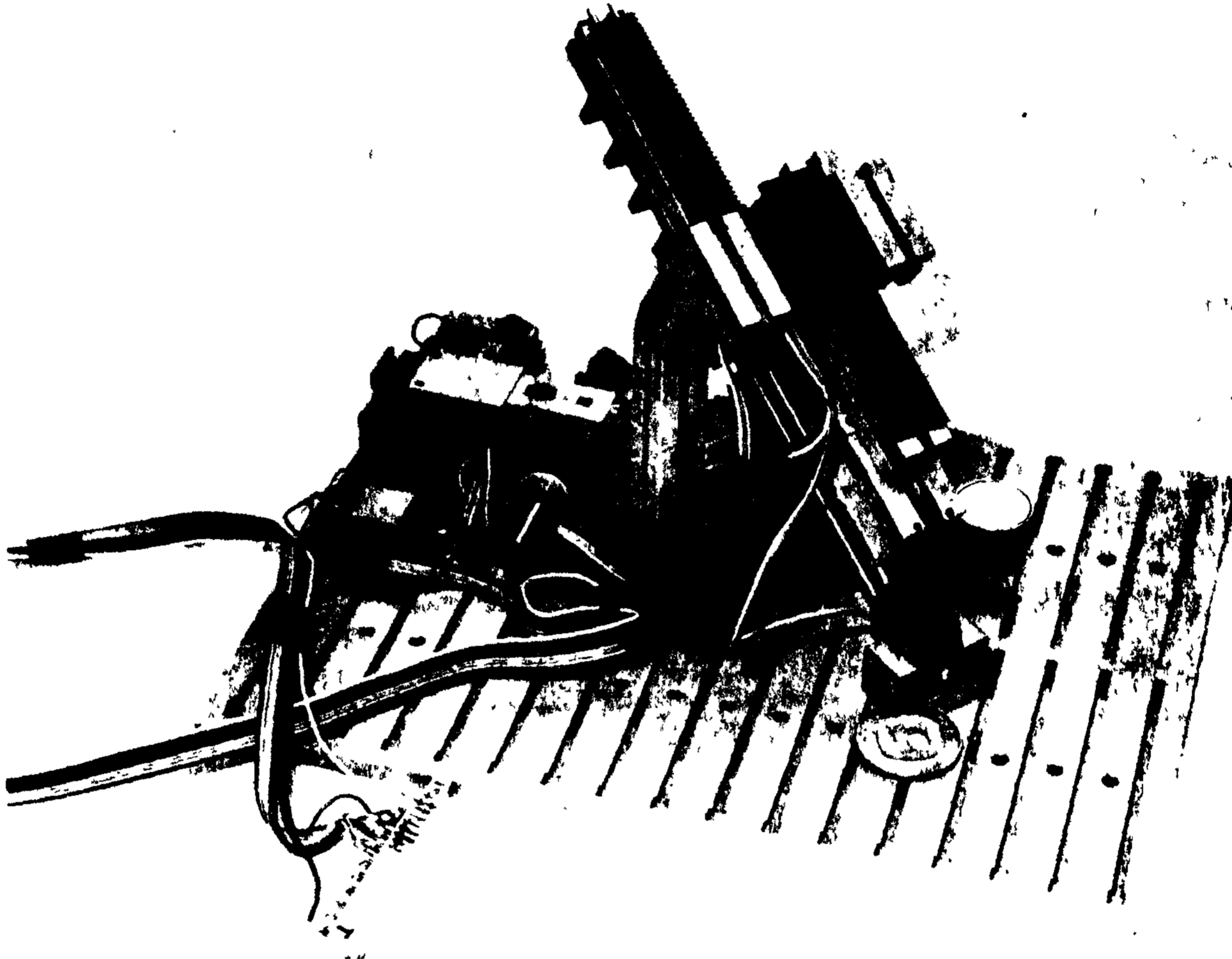


Abb. 46: Teach-in-Roboter mit digitaler Abfrage

Da das Programm sehr lang ist, ist die Erklärung des Programmes gleich bei den Programmschritten beigefügt.

```

***"TEACH-IN"-ROBOTER ***
** Zuordnungen **
* Startposition      :links unten
* Elektromagnet     :A0
* Rechtsdrehen      :A1
* Linksdrehen       :A2
* Tieffahren        :A3
* Hochfahren        :A4
* Wenn A5 = 1       :Takt vom Timer des Computers
* Wenn A5 = 0       :Takt vom Roboter
*
00 INIT              * Magnet an                * Magnet ein
* Magnet ein        * hoch ein                * hoch ein
* hoch ein          73 STO 5                145 STO 5
03 STO 5            74 STOC 3                146 STOC 3
04 STO 0            75 STO 0                147 STO 0
05 STO 4            76 STO 4                148 STO 4
06 STOC 5           77 STOC 5                149 STOC 5
07 NOP x /4 mal    78 NOP x /4 mal          150 NOP x /4 mal

```

* hoch aus	* hoch aus	* hoch aus
* rechts ein	* links ein	* links ein
11 STO 5	82 STO 5	154 STO 5
12 STOC 4	83 STOC 4	155 STOC 4
13 STO 1	84 STO 2	156 STO 2
14 STOC 5	85 STOC 5	157 STOC 5
15 NOP x /20 mal	86 NOP x /10 mal	158 NOP x /9 mal
* rechts aus	* links aus	* links aus
* tief an	* tief an	* tief an
35 STO 5	96 STO 5	167 STO 5
36 STOC 1	97 STOC 2	168 STOC 2
37 STO 3	98 STO 3	169 STO 3
38 STOC 5	99 STOC 5	170 STOC 5
39 NOP x /4 mal	100 NOP x /4 mal	171 NOP x /4 mal
* tief aus	* tief aus	* tief aus
* Magnet aus	* Magnet aus	* Magnet aus
* hoch an	* hoch an	* hoch an
43 STO 5	104 STO 5	175 STO 5
44 STOC 3	105 STOC 3	176 STOC 3
45 STOC 0	106 STOC 0	177 STOC 0
46 STO 4	107 STO 4	178 STO 4
47 STOC 5	108 STOC 5	179 STOC 5
48 NOP x /4 mal	109 NOP x /4 mal	180 NOP x /4 mal
* hoch aus	* hoch aus	* hoch aus
* links an	* rechts an	* links an
52 STO 5	113 STO 5	184 STO 5
53 STOC 4	114 STOC 4	185 STOC 4
54 STO 2	115 STO 1	186 STO 2
55 STOC 5	116 STOC 5	187 STOC 5
56 NOP x /9 mal	117 NOP x /20 mal	188 NOP x /10 mal
* links aus	* rechts aus	* links aus
* tief an	* tief ein	* tief ein
65 STO 5	137 STO 5	198 STO 5
66 STOC 2	138 STOC 1	199 STOC 2
67 STO 3	139 STO 3	200 STO 3
68 STOC 5	140 STOC 5	201 STOC 5
69 NOP x /4 mal	141 NOP x /4 mal	202 NOP x /4 mal
* tief aus	* tief aus	* tief aus
		205 STO 5
		206 STOC 3
		207 JMP x

Ein Programm mit nur wenig verschiedenen Programmbefehlen, aber vielen Programmschritten. Es eignet sich für die Durchführung eines Projekts "Robotics". Die einzelnen Programmabschnitte können arbeitsteilig geschrieben, eingegeben, ausgetestet und koordiniert werden. Das Programm kann in einem EPROM gespeichert bezogen werden, (vergl. Bezugsquellen). Dann ist der Gesamt Ablauf des Programms im Unterricht realistisch.

Ein Programm, das ohne ein EPROM zu benutzen in eine Unterrichtsreihe paßt, wenn es verkürzt eingegeben wird, ist das folgende.

4.6.4. AMPELSTEUERUNG

Natürlich können wir eine Ampelanlage auf unseren Ausgangsleuchtdioden simulieren. Allerdings fällt die Zuordnung zu den einzelnen Signallampen schwer. Um also ein möglichst naturgetreues Modell zu erhalten, greifen wir wieder auf Bauelemente des Fischertechnik computing-Baukastens zurück. Als Treiber für die Signallampen benutzen wir z.B. das IC SN 74LS241 auf einer Pufferplatine (vgl. Bezugsquellen).

Unsere Ampel ist eine Fußgängerampel, d.h. sie stoppt die Autofahrer, wobei sie gleichzeitig den Fußgängern grünes Licht gibt und umgekehrt. Ein Taster ermöglicht die Betätigung durch einen Fußgänger, sie läuft aber auch automatisch auf Langsamtakt, wenn wir den Eingangsschalter E1 betätigen. Jede Signallampe ist so geschaltet, daß eine "0" an einem Ausgang diese anschaltet! Wird die Ampelanlage mit der Pufferplatine verbunden, sind somit zunächst alle Lampen angeschaltet. Dies ändert sich erst mit Ablauf des Programms .

*** Ampel 1 ***

**Festlegung der Zuordnungen im Programm **

* Eingang E1 :Tasterabfrage
 * Schalter E1 :Dauerbetrieb
 * Druckknopf für den Fußgänger = E1
 * Signale für die Fußgänger
 * Ausgang A0 :Grün
 * Ausgang A1 :Gelb
 * Ausgang A2 :Rot

*
 * Signale für die Autofahrer
 * Ausgang A3 :Grün
 * Ausgang A4 :Gelb
 * Ausgang A5 :Rot
 *

00 INIT

* Tasterabfrage *

03 STOC 3 Grün für die Autofahrer
 04 STOC 2 Rot für die Fußgänger
 05 LDC 1 Taster abfragen
 06 SKZ x War der Taster nicht gedrückt, ist das Ergebnisregister = 1, der nächste Befehl wird ausgeführt. Dadurch wird an den Programmstart "Tasterabfrage" gesprungen. Wurde der Taster gedrückt oder wurde der Schalter E1 betätigt, so ist das Ergebnisregister = 0, der nächste Befehl wird ignoriert. Dadurch wird zur Routine "Signalumschaltung" gesprungen.

07 JMP x

* Signalumschaltung *

08 ORC 0 Das Ergebnisregister ist "0", wenn der vorherige Zweig durchlaufen wird.
 09 STO 3 Grün für die Autofahrer ausschalten
 10 STOC 4 Gelb für die Autofahrer
 11 STOC 1 Gelb für die Fußgänger
 12 NOP x 10 Takte Pause
 21 NOP x

LS 241 → LS 240

22	STO	4	Gelb für die Autofahrer ausschalten
23	STOC	5	Rot für die Autofahrer
24	STO	2	Rot für die Fußgänger ausschalten
25	STO	1	Gelb für die Fußgänger ausschalten
26	STOC	0	Grün für die Fußgänger
27	NOP	x	60 Takte Pause
86	NOP	x	
87	STO	0	Grün für die Fußgänger ausschalten
88	STOC	1	Gelb für die Fußgänger
89	STOC	4	Gelb für die Autofahrer
90	NOP	x	10 Takte Pause
99	NOP	x	
100	STO	1	Gelb für die Fußgänger ausschalten
101	STOC	2	Rot für die Fußgänger
102	STO	5	Rot für die Autofahrer ausschalten
103	STO	4	Gelb für die Autofahrer ausschalten
104	STOC	3	Grün für die Autofahrer
105	NOP	x	25 Takte Pause
129	NOP	x	
130	JMP	x	

Läßt man die Pausen weg, so kann das Programm, die Pausen für die einzelnen Signalphasen berücksichtigend, im Hand-Takt durchlaufen werden. Soll dagegen das komplette Programm in den Computer geladen werden, empfiehlt sich wieder die EPROM-Version (siehe Bezugsquellen).

Das folgende Ampelprogramm berücksichtigt die Tatsache, daß der Fußgänger kein Gelb bekommt.

Ampel 2

Festlegung der Zuordnungen im Programm

*Druckknopf für den Fußgänger = E1

*Lichtsignale für den Fußgänger: Grün = A0, Rot = A1

*Lichtsignale für die Autofahrer: Grün = A2, Gelb = A3, Rot = A4

*Bemerkungen: F = Fußgänger, A = Autofahrer

00	INIT		
03	STO	0	
04	STO	3	A. Grün an
05	STO	4	F. Rot an
06	LDC	1	
07	SKZ	x	
08	JMP	x	
09	ORC	0	
10	STO	2	A. Grün aus
11	STOC	3	A. Gelb an
12	NOP	x	
.			
.	NOP	x	Gelbphase (20 Takte)
.			
31	NOP	x	
32	STO	3	A. Gelb aus
33	STOC	4	A. Rot an
34	NOP	x	
.			
.	NOP	x	Fußgängerampel springt nicht sofort um (10 Takte)

falsch

```

43 NOP x
44 STO 1 F. Rot aus
45 STOC 0 F. Grün an
46 NOP x
.
. NOP x F. Grünphase (60 Takte)
.
105 NOP x
106 STO 0 F. Grün aus
107 STOC 1 F. Rot an
108 NOP x
.
. NOP x Autofahrerampel springt nicht sofort um, damit
auch Fußgänger, die sich noch auf der Straße
befinden, noch gefahrlos über die Straße können
(30 Takte).
137 NOP x
138 STOC 3 A. Gelb an
139 NOP x
.
. NOP x Rot-Gelb-Phase (10 Takte)
.
148 NOP x
149 STO 4 A. Rot aus
150 STO 3 A. Gelb aus
151 STOC 2 A. Grün an
152 NOP x
.
. NOP x A. Grünphase (104) Takte
.
255 NOP x Lange Grünphase, damit nicht durch dauerndes
Betätigen des Druckknopfs der Autoverkehr
blockiert wird.

```

Auch für dieses Programm gilt wieder: läßt man die Pausen weg, so kann das Programm, die Pausen für die einzelnen Programmphasen berücksichtigend, im Handtakt durchlaufen werden. Soll dagegen das komplette Programm in den Computer geladen werden, empfiehlt sich wieder die EPROM-Version (vergl. Bezugsquellen).

4.6.5. MORSEZEICHENGENERATOR

Ein Morsezeichengenerator (vgl. Bezugsquellen) arbeitet mit dem bekannten NE 555. Die bestückte Platine wird mit einer Buchsenleiste versehen (Siehe: Abb.47). Die Stromversorgung für den Morsezeichengenerator wird von den Anschlüssen 1 und 2 des Peripheriesteckers an die mit "+" und "-" gekennzeichneten Stellen geführt, der Anschluß 4 des NE 555 wird z.B. mit Ausgang A0 des Computers verbunden. STO 0 schaltet den Tonerzeuger ein und STOC 0 schaltet ihn aus. Nach der Initialisierung bewirkt ein STO 0 und dahinter ein STOC 0 einen Morsepunkt, dreimal STO 0 und dahinter ein STOC 0 ergibt einen Morsestrich. Eine Buchstabenpause entsteht durch Anfügen von zwei STOC 0-Programmschritten. Der Morsezeichengenerator in Folge 6 der Fernsehreihe "Bit und Byte-Wir bauen einen Computer" gibt das Amateurfunkrufzeichen "DDOEU" aus.

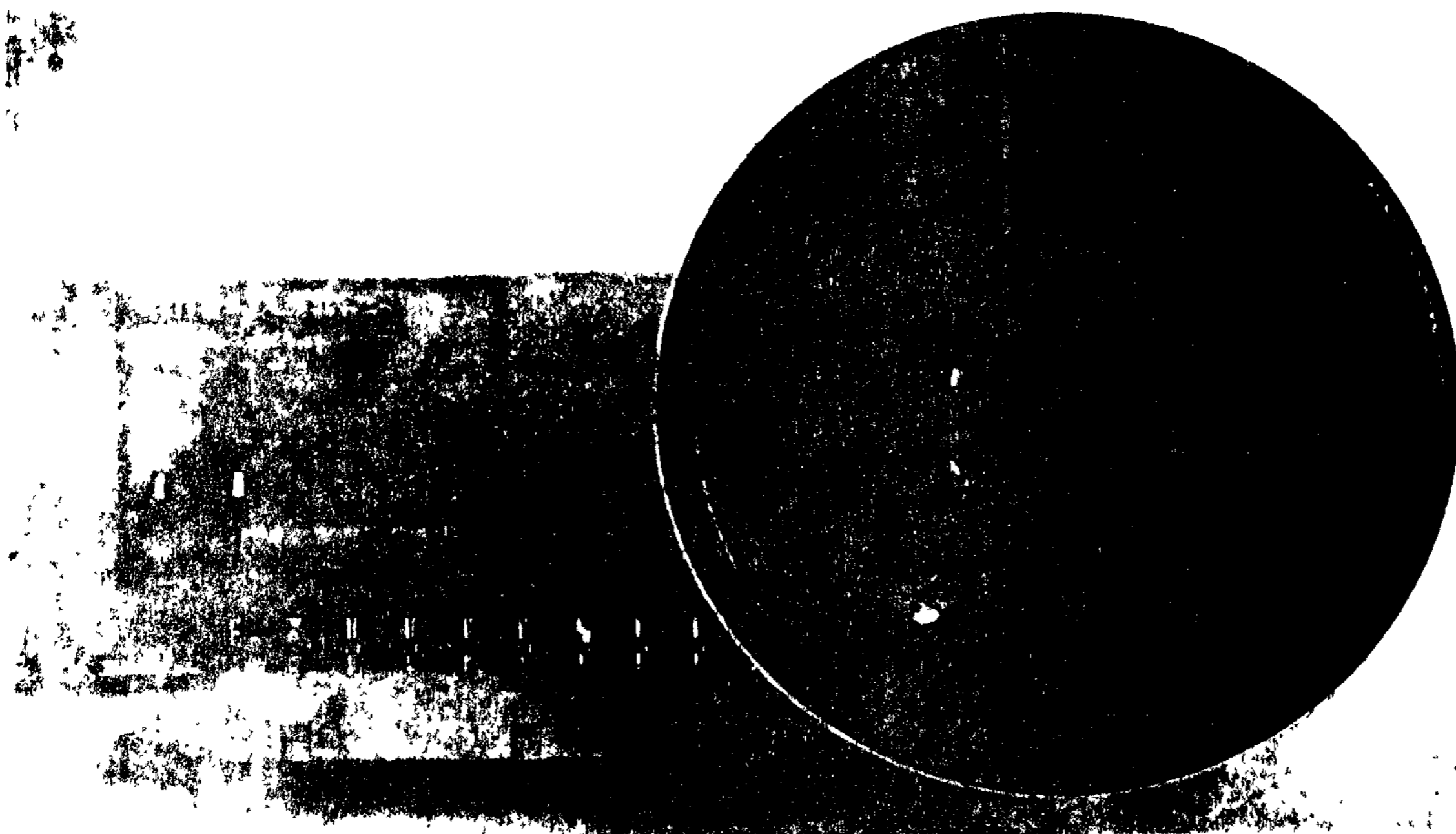


Abb. 47: Zeichengenerator

4.6.6. SIEBENSEGMENT-ANZEIGE

Abb. 48 zeigt die aufgebaute Platine für eine Siebensegmentanzeige (vergl. Bezugsquellen). In dem IC sind 8 invertierende Treiber enthalten, deren Eingänge mit den Ausgängen A0-A7 des Computers verbunden sind. 7 Treiber (A1 - A7) sind Verstärker für die 7 Leuchtdiodensegmente, denen jeweils ein Widerstand vorgeschaltet ist. Der 8. Treiber (A0) schaltet den Leistungstransistor, dessen Kollektor die gemeinsame Kathode der Siebensegmentanzeige auf "0" setzt. Durch diese Beschaltung ist ein flackerfreier Bildaufbau für die Darstellung alphanumerischer Zeichen gewährleistet.

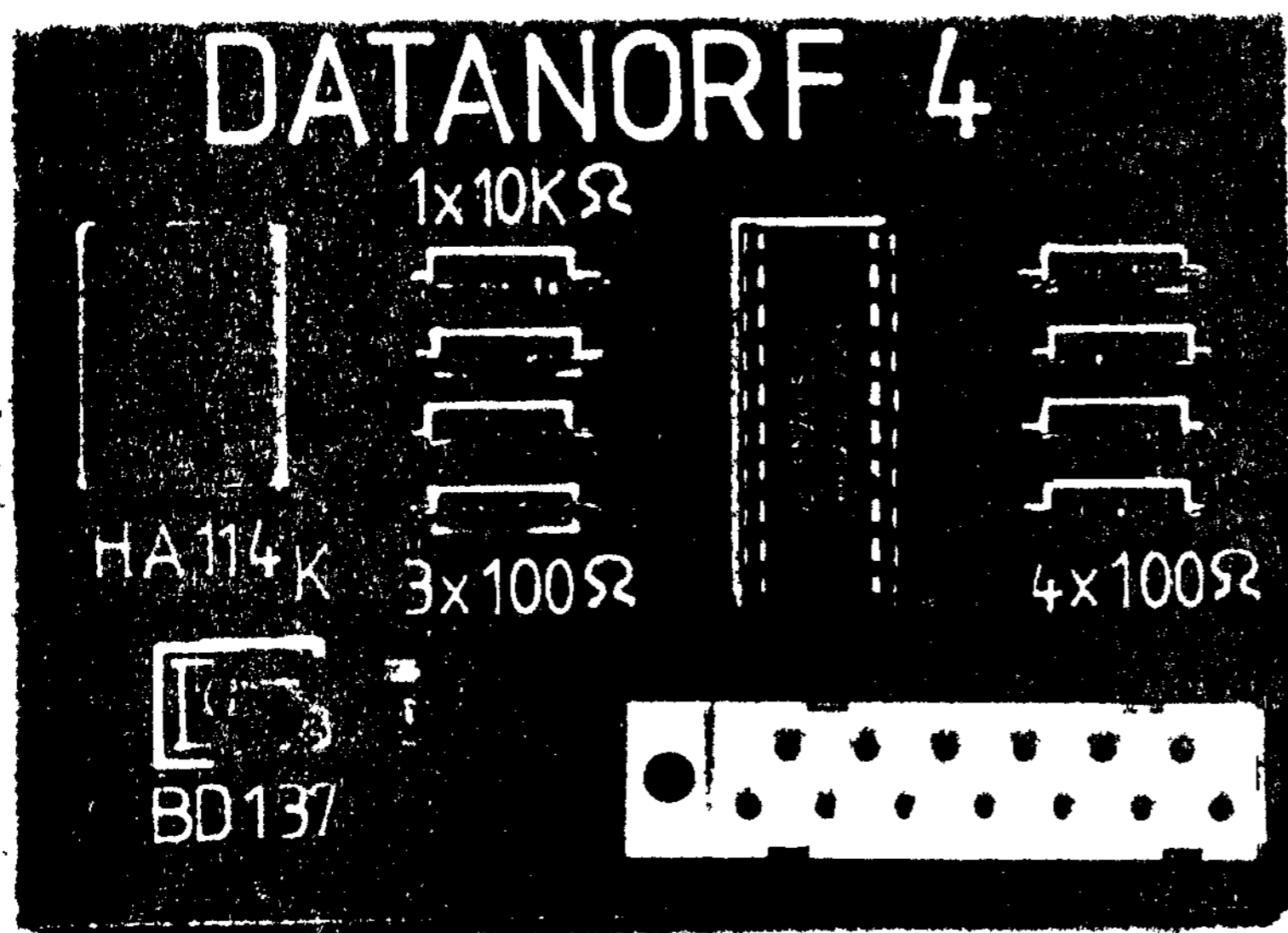


Abb. 48: Platine für die Siebensegmentanzeige

4.6.7. BOHRER

Task
 Das Fischertechnik-Bohrrobotermodell hat einen Motor für die Vorlage der Werkstücke sowie einen zweiten für das Heranführen des Bohrers an das Werkstück. Der Bohrvorgang selbst wird simuliert. Für die Erkennung der Endstellungen des Bohrvorschubs werden zwei ~~Schalter~~ abgefragt. Für die Programmierung der Steuerung des Bohrvorschubs reicht die Anwendung eines RS-Flipflops, das über die Schalter S1 und S2 getaktet wird: Der Bohrvorschubmotor wird nach dem Anhalten des Vorlagemotors eingeschaltet und führt die Vorlage solange nach unten, bis der Schalter S1, der ihn abschaltet, erreicht ist. Nach dem Bohren wird der Bohrvorschubmotor so wieder eingeschaltet, daß er den Vorschub nach oben führt. Ist der obere Schalter S2 erreicht, wird der Bohrvorschub gestoppt und das Drehen des Vorlagemotors für die zu bohrenden Werkstücke veranlaßt. Ein dritter Schalter, der durch die Drehung des Vorlagemotors betätigt wird, beendet die Vorlage des zu bohrenden Werkstücks und teilt dem Computer so die Lage des Werkstückes mit. Nach dem Bohren muß der Vorlagemotor bei geschlossenem Schalter S3 gestartet werden. Der Vorlagemotor muß sich weiterdrehen, obwohl der Schalter S3 jetzt geöffnet ist. Erst wenn S3 wieder geschlossen wird, muß der Vorlagemotor wieder angehalten werden.

Bohrroboter

Festlegung der Zuordnung im Programm

- * Ausgang A0 : Werkstückvorlage drehen
- * Ausgang A1 : Bohrkopf runterfahren
- * Ausgang A2 : Bohrkopf hochfahren
- * (Achtung: A1 und A2 dürfen nicht gleichzeitig "1" sein !!!)
- * *Task*
- * Eingang E1 : ~~Schalter~~ Vorlage (*F1*)
- * Eingang E2 : ~~Schalter~~ unten (*F2*)
- * Eingang E3 : ~~Schalter~~ oben (*F3*)
- * E/A7 = 0 : Prozedur drehen
- * E/A7 = 1 : Prozedur bohren
- * In Prozedur bohren : E/A6 = 0 : Prozedur runter
E/A6 = 1 : Prozedur hoch
- * In Prozedur drehen : E/A6 = 0 : drehen 0
E/A6 = 1 : drehen 1

00 INIT	13 LD 1	24 LD 7
* Prozedur drehen	14 STOC 0	25 AND 6
03 LDC 7	15 STOC 6	26 OEN 0
04 ANDC 6	16 STO 7	27 STO 2
05 OEN 0	*Prozedur runter	28 LD 3
06 STO 0	17 LD 7	29 STOC 2
07 LD 1	18 ANDC 6	30 STOC 7
08 STOC 6	19 OEN 0	31 STOC 6
*Prozedur drehen 1	20 STO 1	32 JMP x
09 LDC 7	21 LD 2	* Ende
10 AND 6	22 STOC 1	
11 OEN 0	23 STO 6	
12 STO 0	*Prozedur hoch	

4.6.8. PLOTTER

Ein Plotter ist ein Zeichengerät. Wir wollen nun die Schrift "1-bit" auf ein Blatt Papier malen. Der Fischertechnik-Plotter hat zwei Motoren und eine Vorrichtung zum Zeichnen. Beide müssen angesteuert werden, damit der Plotter zeichnen kann. Für die Abfrage der Koordinaten der Stellung des Schlittens für den Zeichenstift werden Zählimpulse über das Interface an den Programmzähler gegeben. Für die digitale Abfrage werden eine Seiltrommel auf die Schneckenwell des Motors für das Drehen des Schreibarms und Winkelstücke auf die Aluminiumschiene zur Führung des Schlittens für den Plotterstift angebracht, die von Schaltern abgefragt werden. (Vergl. Abb. 49). Für das Plotten der Schrift "1-bit" ist ein umfangreiches Programm von 159 Programmschritten nötig. Daher ist die Erklärung des Programms gleich beigefügt.

Plotter

Festlegung der Zuordnungen im Programm

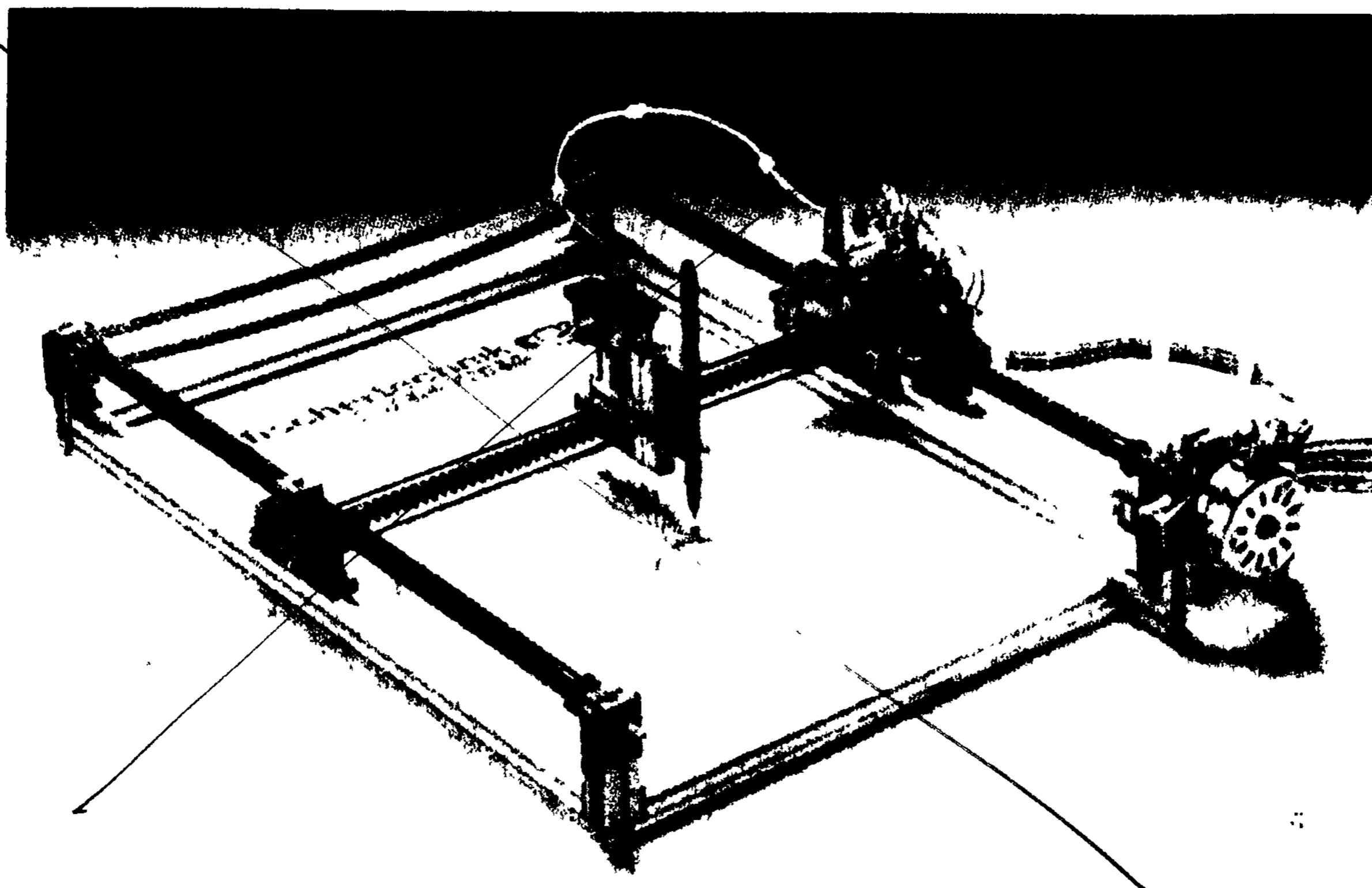
*Start: Links unten, A0: Stift, A1: Links fahren, A2: Rechts fahren, *A3: Tief fahren, A4: Hoch fahren, A5: 1 = Takt vom Prozessor, 2 = Takt vom Modell

00 INIT	* "b" schreiben, 4	* rechts aus
* "1" Schreiben 1	* links aus	*Stift runter
* Stift runter	* hoch an	* hoch an
* hoch an	56 STO 5	102 STO 5
03 STO 5	57 STOC 1	103 STOC 2
04 STO 0	58 STO 4	104 STO 0
05 STO 4	59 STOC 5	105 STO 4
06 STOC 5	60 NOP x /7 mal	106 STOC 5
07 NOP x /7 mal	* "i" anfahren 1	107 NOP x /7 mal
* "1" schreiben 2	* Stift hoch	* "t" schreiben 2
* hoch aus	* hoch aus	* Stift hoch
* links an	* rechts an	* hoch aus
14 STO 5	67 STO 5	* rechts an
15 STOC 4	68 STOC 0	114 STO 5
16 STO 1	69 STOC 4	115 STOC 0
17 STOC 5	70 STO 2	116 STOC 4
* hoch aus	71 STOC 5	117 STO 2
* Stift hoch	72 NOP x /6 mal	118 STOC 5
* "i" anfahren 1	* "i" anfahren 2	* "t" schreiben 3
18 STO 5	* rechts aus	* rechts aus
19 STOC 4	* tief an	* tief an
20 STOC 0	78 STO 5	119 STO 5
21 STO 3	79 STOC 2	120 STOC 2
22 STOC 5	80 STO 3	121 STO 3
23 NOP x /3 mal	81 STOC 5	122 STOC 5
* "b" anfahren	* "i" Punkt	123 STOC 2
* Stift hoch	* Stift runter	124 NOP x
* rechts an	* weiter tief	* "t" schreiben 3
26 STO 5	82 STO 5	* tief aus
27 STO 2	83 STO 0	* Stift runter
28 STOC 0	84 STOC 5	* links an
29 STOC 5	* "i" Rest 1	125 STO 5
30 NOP x /6 mal	* Stift hoch	126 STOC 3
* "b" schreiben 1	* weiter tief	127 STO 0
* Stift runter	85 STO 5	128 STO 1
* weiter rechts	86 STO 0	129 STOC 5

36 STO 5	87 STOC 5	130 NOP x /2 mal
37 STO 0	88 NOP x	* Anfang anfahren 2
38 STOC 5	* "i" Rest 2	* Stift hoch
39 NOP x /3 mal	* Stift runter	* links aus
* "b" schreiben 2	* weiter tief	* tief an
* rechts aus	89 STO 5	132 STO 5
* tief an	90 STO 0	133 STOC 0
42 STO 5	91 STOC 5	134 STOC 1
43 STOC 2	92 NOP x /3 mal	135 STO 3
44 STO 3	* "t" anfahren	136 STOC 5
45 STOC 5	* Stift hoch	137 NOP x /4 mal
46 NOP x /3 mal	* tief aus	* Anfang anfahren 2
* "b" schreiben 3	* rechts an	* tief aus
* tief aus	95 STO 5	* links an
* links an	96 STOC 0	141 STO 5
49 STO 5	97 STOC 3	142 STOC 3
50 STOC 3	98 STO 2	143 STO 1
51 STO 1	99 STOC 5	144 STOC 5
52 STOC 5	100 NOP x /2 mal	145 NOP 0 /12 mal
53 NOP x /3 mal	* "t" schreiben 1	* links aus
		157 STO 5
		158 STOC 1
		159 JMP x

Die im WDR-1-Bit-Computer liegenden Möglichkeiten haben wir bei weitem noch nicht erschöpft. Der interessierte Leser und Computerbauer findet sicher noch viele weitere Beispiele.

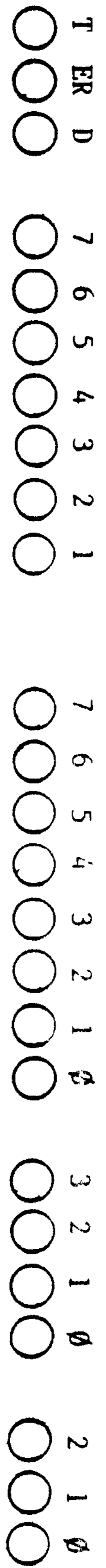
Abb. 49: Plotter



Anhang

Übersicht über die Bedienungselemente und LED-Anzeige	120
Internationaler Farbcode für Widerstände	121
Hexadezimal-Dual-Dezimal-Tabelle	122
Dauerhafte Speicherung von Programmen	123
Der Timer NE 555 als Rechteckgenerator	124
Datenblatt Timer NE 555	128
Minimalprojekt "Bit und Byte - Wir bauen einen Computer"	132
Beispiel: Durchführung einer Unterrichtseinheit im WPF-Bereich (9./10. Schuljahr) in einer Gesamtschule	133
Die Simulation des WDR-1-Bit-Computers mit dem Schulcomputer	144
Schüler bauen sich Computer	145
Zur Sendereihe "Bit und Byte"	146
Der Computerschein	148
Erweiterungen zum WDR-1-Bit-Computer	149
Literaturhinweise	191
Bezugsquellen	192

Übersicht über die Bedienungselemente und die LED-Anzeige



CPU

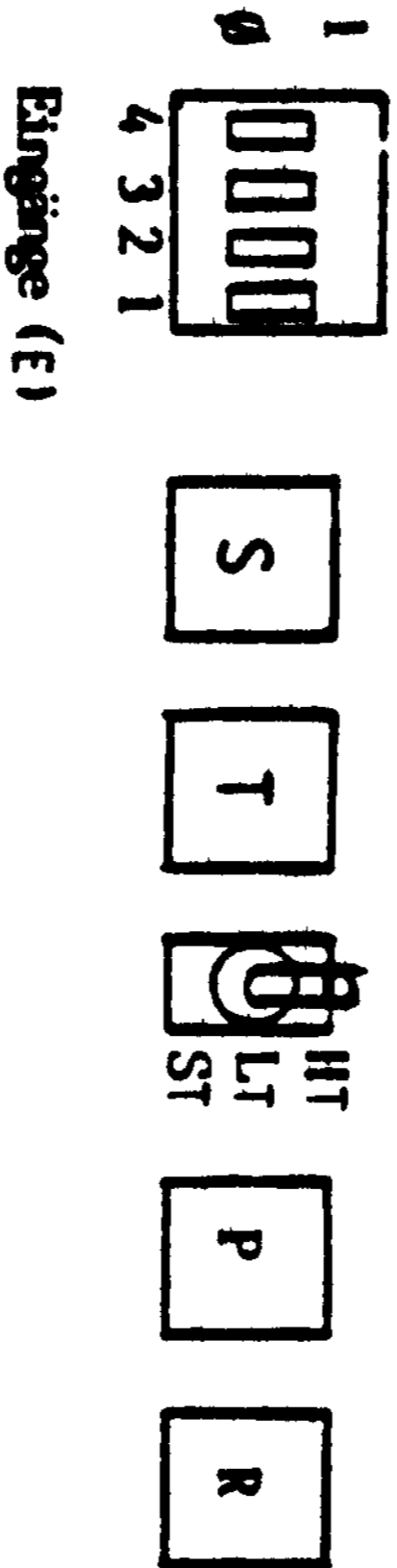
Eingänge (E)

Ausgänge (A)

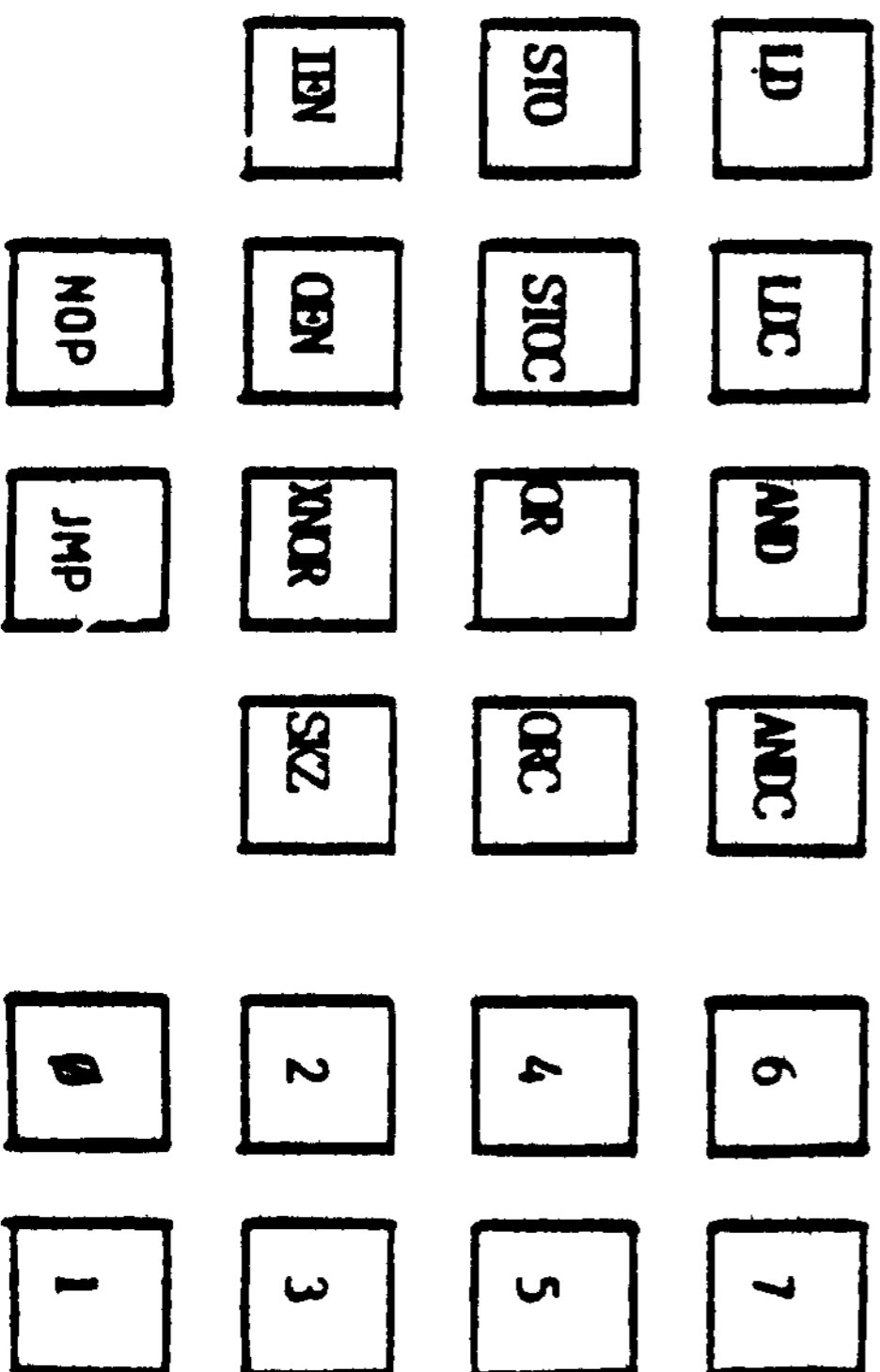
Befehl

E/A-Adresse

Programmschritt



Eingänge (E)



INTERNATIONALER FARBCODE FÜR WIDERSTÄNDE

Auf den Widerständen ist der Widerstandswert aufgedruckt. Er ist durch mehrere Farbringe codiert, gemäß dem internationalen Farbcode.

Die Farbringe haben von links nach rechts folgende Bedeutung:

Ring 1: 1. Stelle des Wertes

Ring 2: 2. Stelle des Wertes

Ring 3: Multiplikator bzw. Anzahl der anzuhängenden Nullen

Ring 4: Toleranzbereich



1., 2. und 3. Ring Toleranz
Wertangabe

Beispiel: Ein Widerstand hat folgende Farbringe:

braun - schwarz - grün - gold

1 0 00000 5%

Es ist also ein Widerstand mit dem Wert $10 \cdot 10^5$ Ohm = 1M Ohm mit dem Toleranzbereich 5%.

Internationaler Farbcode					
	1. Ring	2. Ring	3. Ring	4. Ring = Toleranz	
schwarz	0	0	-	Farbe:	
braun	1	1	0	braun	1 %
rot	2	2	00	rot	2 %
orange	3	3	000	gold	5 %
gelb	4	4	0000	silber	10 %
grün	5	5	00000	ohne	20 %
blau	6	6	000000		
violett	7	7			
grau	8	8			
weiß	9	9			

HEXADEZIMAL-DUAL-DEZIMAL-TABELLE

Die langwierige Umrechnung von Dezimalzahlen in Dualzahlen hat zur Einführung der Hexadezimalzahlen geführt. Im Hexadezimalsystem werden sechzehn Ziffern benötigt. Verwendet werden zunächst einmal die bekannten zehn Ziffern von 0 bis 9 des Dezimalsystems. Für die Zahlenwerte von 10 bis 15 hat man die Buchstaben A bis F genommen. (Die Umwandlung von Hexadezimalzahlen in Dezimalzahlen erfolgt nach dem vom Dualsystem her bekannten Prinzip. Die Wertigkeit versechzehnfacht sich aber von hinten nach vorne pro Stelle.) Jede mit vier Dualstellen darstellbare Zahl kann durch eine Hexadezimalzahl dargestellt werden. Je vier Dualstellen ergeben also eine Hexadezimalstelle, d.h. durch eine zweistellige Hexadezimalzahl kann eine achtstellige Dualzahl ausgedrückt werden.

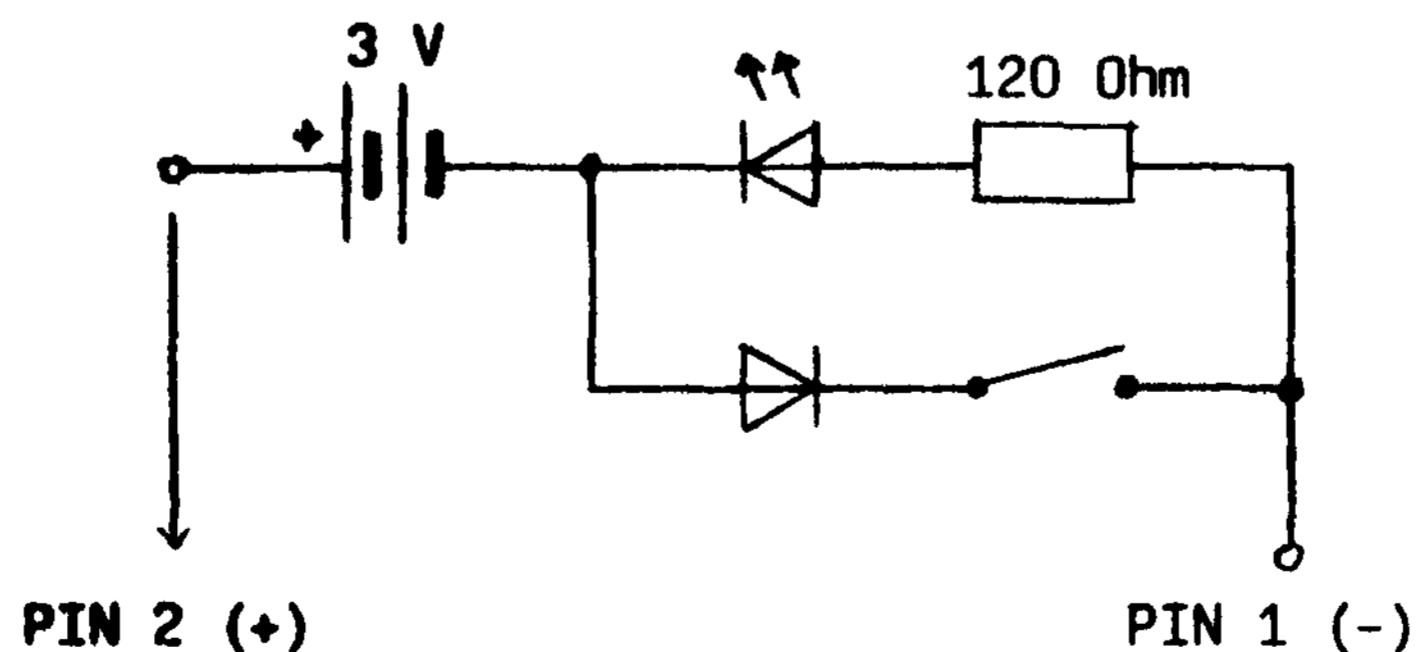
Hexadezimalzahl	Dualzahl	Dezimalzahl
00	00000000	0
01	00000001	1
02	00000010	2
03	00000011	3
04	00000100	4
05	00000101	5
06	00000110	6
07	00000111	7
08	00001000	8
09	00001001	9
0A	00001010	10
0B	00001011	11
0C	00001100	12
0D	00001101	13
0E	00001110	14
0F	00001111	15
10	00010000	16
11	00010001	17
.	.	.
.	.	.
.	.	.
9F	10011111	159
A0	10100000	160
.	.	.
.	.	.
.	.	.
FE	11111110	254
FF	11111111	255

DAUERHAFTE SPEICHERUNG VON PROGRAMMEN

Bei den einfachsten Programmen, die aus wenigen Programmschritten bestehen, ist der Verlust des Programmes durch das Ausschalten des Gerätes nicht tragisch, da es relativ schnell wieder eingegeben werden kann. Anders sieht es dagegen bei umfangreicheren Programmen aus. Hier ist ein Programmverlust nicht mehr vertretbar, wenn z.B. bei der Entwicklung eines umfangreicheren Steuerungsprojektes dieses unter großem Zeitaufwand immer wieder neu eingeben werden muß. Eine einfach durchzuführende Ergänzung auf der Speicherplatine ergibt eine Abhilfe. Da der Speicher bei Stromausfall seinen Inhalt verliert, muß dafür gesorgt werden, daß in einem solchen Fall eine Batterie die Stromversorgung rechtzeitig übernimmt. Noch günstiger ist ein Nickel-Cadmium-Akku, der fest auf die Platine gelötet wird. Die Abbildung unten zeigt das Prinzip der Erweiterung. Da die Kapazität der genannten Akkus 100 mAh beträgt und die Speicher einen Strom von etwa 10 mA benötigen, reicht diese Anordnung sogar für einen etwas längeren Transport von wenigstens sechs Stunden aus, wenn vorher dafür gesorgt war, daß die Speicherplatine mindesten 10 Stunden im Computer steckte und geladen wurde, da die Schaltung so konzipiert ist, daß der Ladestrom etwa 10 mA beträgt. Mit einem Schalter wird der Akku nur dann dazugeschaltet, wenn er geladen werden soll oder ein Programm gesichert werden soll.

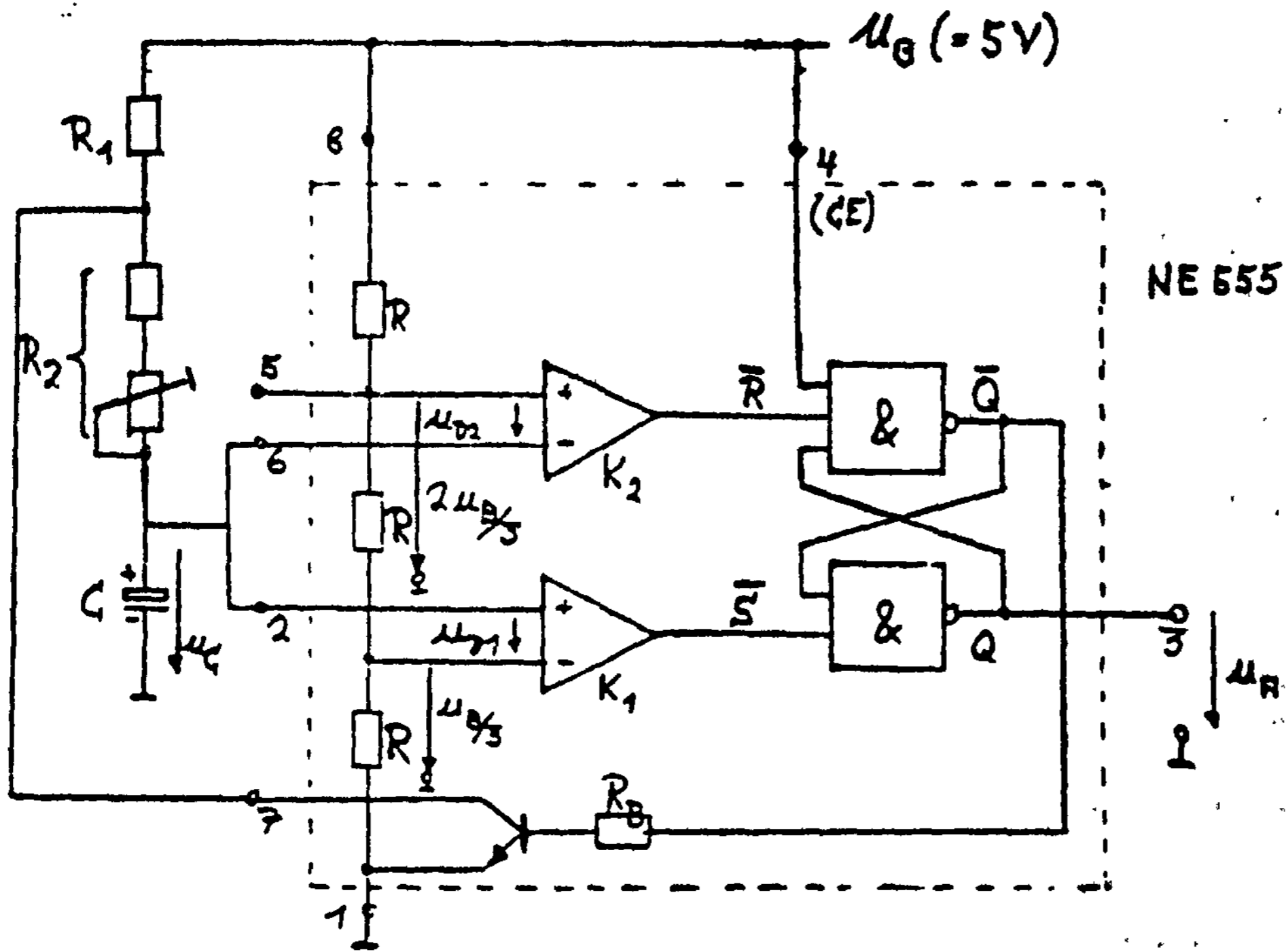
Eine andere Möglichkeit besteht in der Verwendung einer RAM/EPROM-Platine, die gegen die vorhandene Speicherkarte ausgewechselt wird. Diese Platine ermöglicht die Ausdehnung von Programmen bis zu 2 KiloByte (1 Byte = 8 Bit) Speicherkapazität. Weiterhin können unter Verwendung eines EPROMS dauerhaft komplexere Programme z.B. für Demonstrationszwecke untergebracht werden. Das "Brennen" der EPROMS gelingt mit Hilfe des WDR-1-Bit-Computers unter Verwendung einer Zusatzplatine (s. DATANorf-Bausätze).

Schaltung für die Stromversorgung mit einem Akku

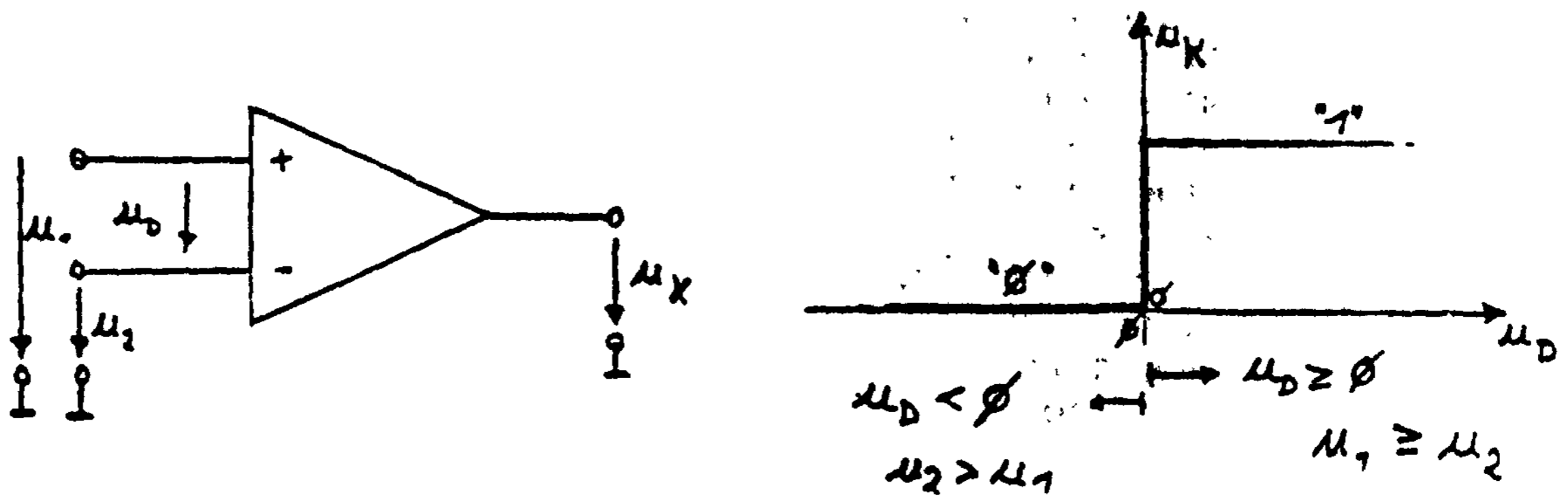


der Steckerleiste

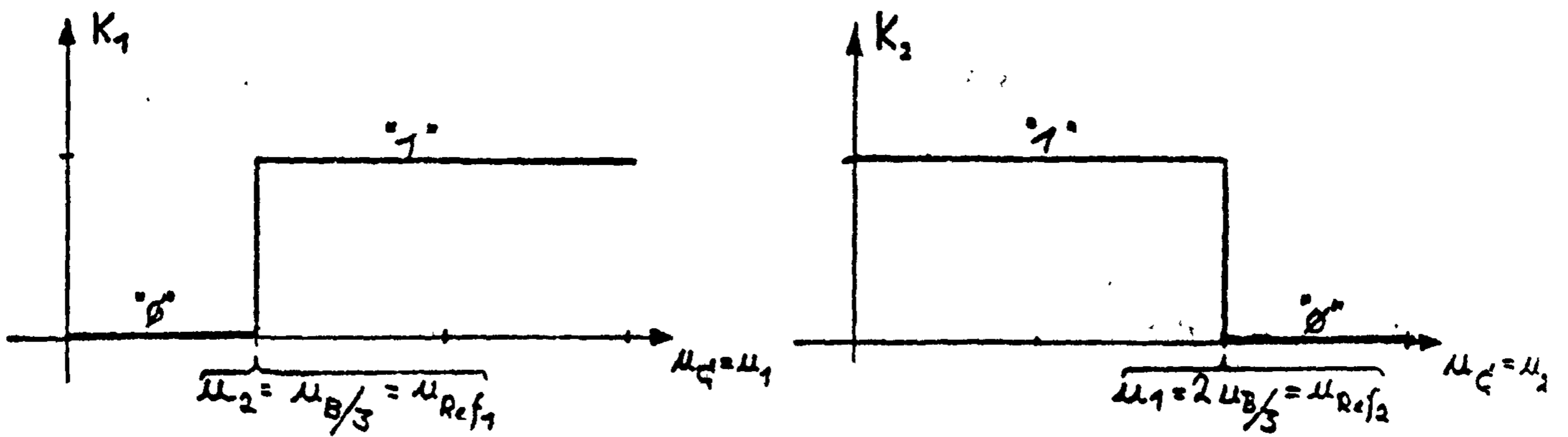
Der Timer NE 555 als Rechteckgenerator



Der Timer NE 555 mit seiner äußeren Beschaltung



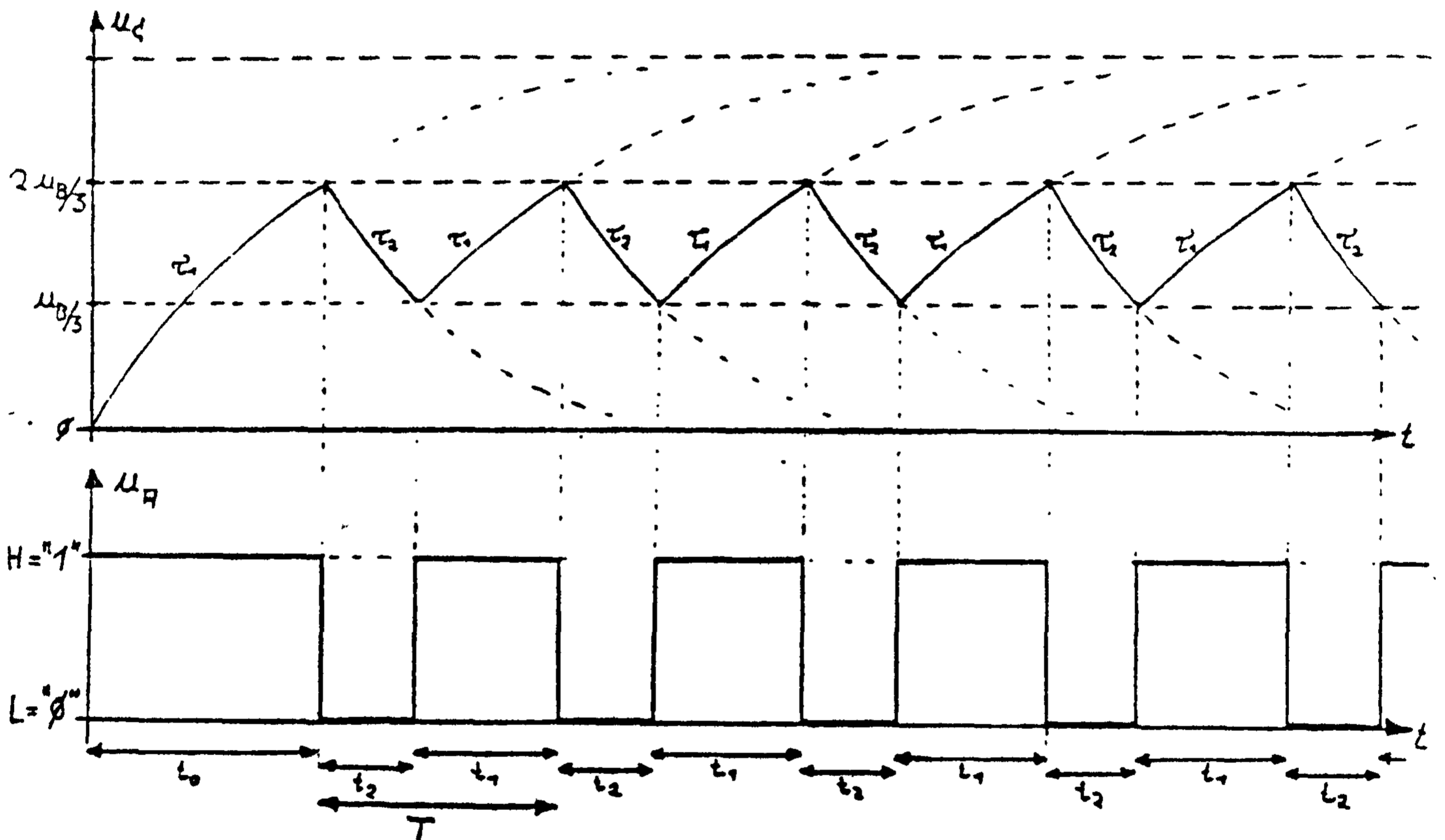
Komparator mit Übertragungskennlinie $u_K = f(u_D)$



Die Übertragungskennlinien der beiden Komparatoren

μ_C	$\bar{S} (=K_1)$	$\bar{R} (=K_2)$	$Q (= \mu_D)$
—	—	—	—
$< \mu_{B/3}$	\emptyset	1	1
$\geq \mu_{B/3}$ und $\leq 2\mu_{B/3}$	1	1	Q^- (Speicherfall)
$> 2\mu_{B/3}$	1	\emptyset	\emptyset

Wahrheitstabelle für das RS-Flipflop

Spannungsdiagramm $u_C = f(t)$ und $u_D = f(t)$ τ_1 = Aufladezeitkonstante

$$\tau_1 = (R_1 + R_2) \cdot C$$

Der Kondensator C wird durch die Betriebsspannung u_B über R_1 und R_2 aufgeladen.

 τ_2 = Entladezeitkonstante

$$\tau_2 = R_2 \cdot C$$

Der Kondensator C wird über R2 entladen. Der Entladestrom fließt über Pin 7 des Timers und den durchgeschalteten Transistor nach Masse.

t_0 = Aufladezeit des Kondensators direkt nach dem Anlegen der Betriebsspannung

Dies ist ein einmaliger Vorgang. Die Zeit t_0 trägt daher nicht zur Periodendauer T bei.

t_1 = Aufladezeit des Kondensators

t_2 = Entladezeit des Kondensators

T = Periodendauer der Ausgangsspannung $T = t_1 + t_2$

Berechnung der Aufladezeit t_1 :

Der Aufladevorgang eines Kondensators wird durch die folgende Gleichung beschrieben:

$$u_c = (1 - e^{-t/\tau}) \cdot u_B$$

Die Kondensatorspannung braucht hier nur den Wert $\frac{2}{3} u_B$ in der Zeit t_1 zu erreichen und hat zu Beginn des Aufladevorgangs bereits den Wert $\frac{1}{3} u_B$. Daraus ergibt sich folgende neue Gleichung:

$$\frac{1}{3} u_B = (1 - e^{-t_1/\tau_1}) \cdot \frac{2}{3} u_B$$

Diese Gleichung wird nach t_1 aufgelöst:

$$t_1 = \tau_1 \cdot \ln 2$$

Berechnung der Entladezeit t_2 :

Der Entladevorgang eines Kondensators wird durch folgende Gleichung beschrieben:

$$u_c = e^{-t/\tau} \cdot u_B$$

Die Kondensatorspannung braucht hier nur den Wert $\frac{1}{3} u_B$ in der Zeit t_2 zu erreichen und hat zu Beginn des Entladevorganges den Wert $\frac{2}{3} u_B$. Daraus ergibt sich folgende neue Gleichung:

$$\frac{1}{3} u_B = e^{-t_2/\tau_2} \cdot \frac{2}{3} u_B$$

Diese Gleichung wird nach t_2 aufgelöst:

$$t_2 = \tau_2 \cdot \ln 2$$

Berechnung der Frequenz f des Rechtecksignals:

Die Periodendauer T beträgt:

$$T = t_1 + t_2 = (\tau_1 + \tau_2) \cdot \ln 2 = (R_1 + 2 \cdot R_2) \cdot C \cdot \ln 2$$

Die Frequenz f ist der Kehrwert der Periodendauer:

$$f = \frac{1}{T} = \frac{1}{(R_1 + 2 \cdot R_2) \cdot C \cdot \ln 2} = \frac{1,44}{(R_1 + 2 \cdot R_2) \cdot C}$$



PRECISION TIMER

FEATURES

- Timing From Microseconds Through Hours
- Operates In Both Astable And Monostable Modes
- Adjustable Duty Cycle
- High Current Output Can Source Or Sink 200mA
- Output Can Drive TTL
- Temperature Stability Of 0.005% Per °C
- Normally On And Normally Off Output

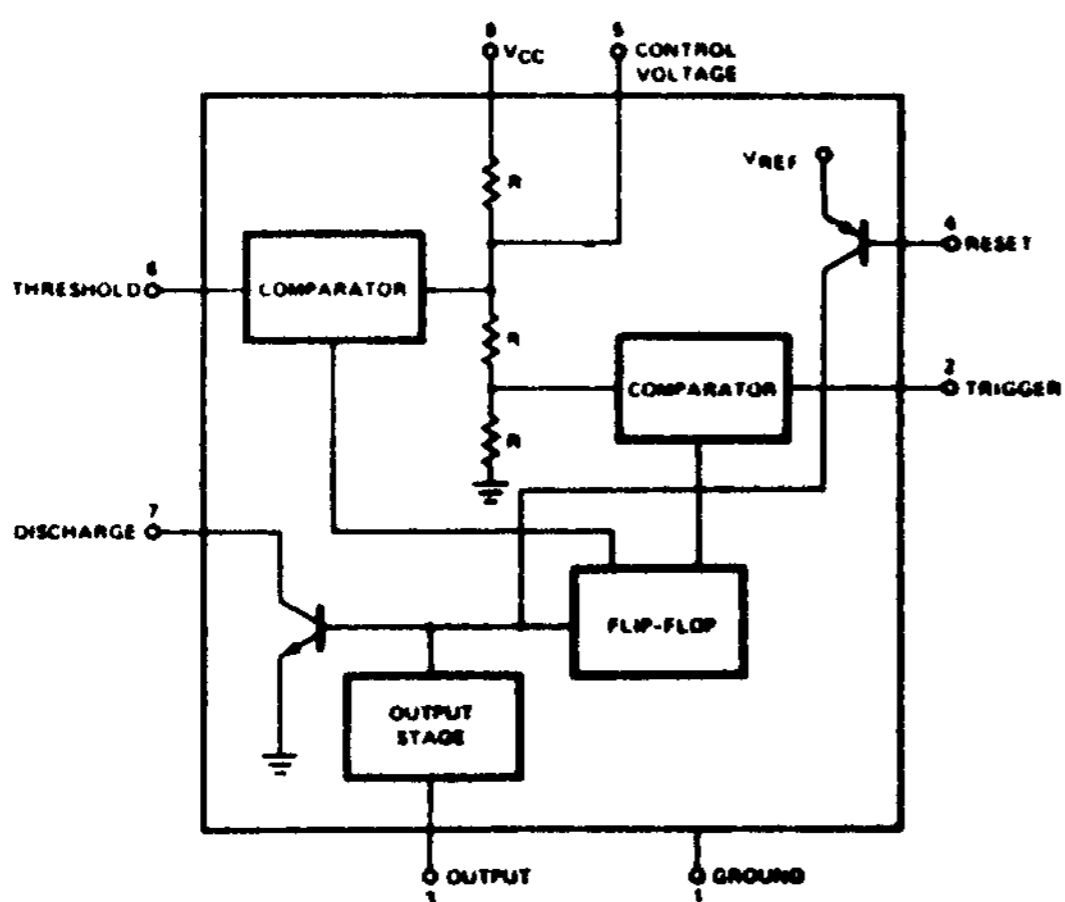
APPLICATIONS

- Precision Timing
- Pulse Generation
- Sequential Timing
- Time Delay Generation
- Pulse Width Modulation
- Pulse Position Modulation
- Missing Pulse Detector

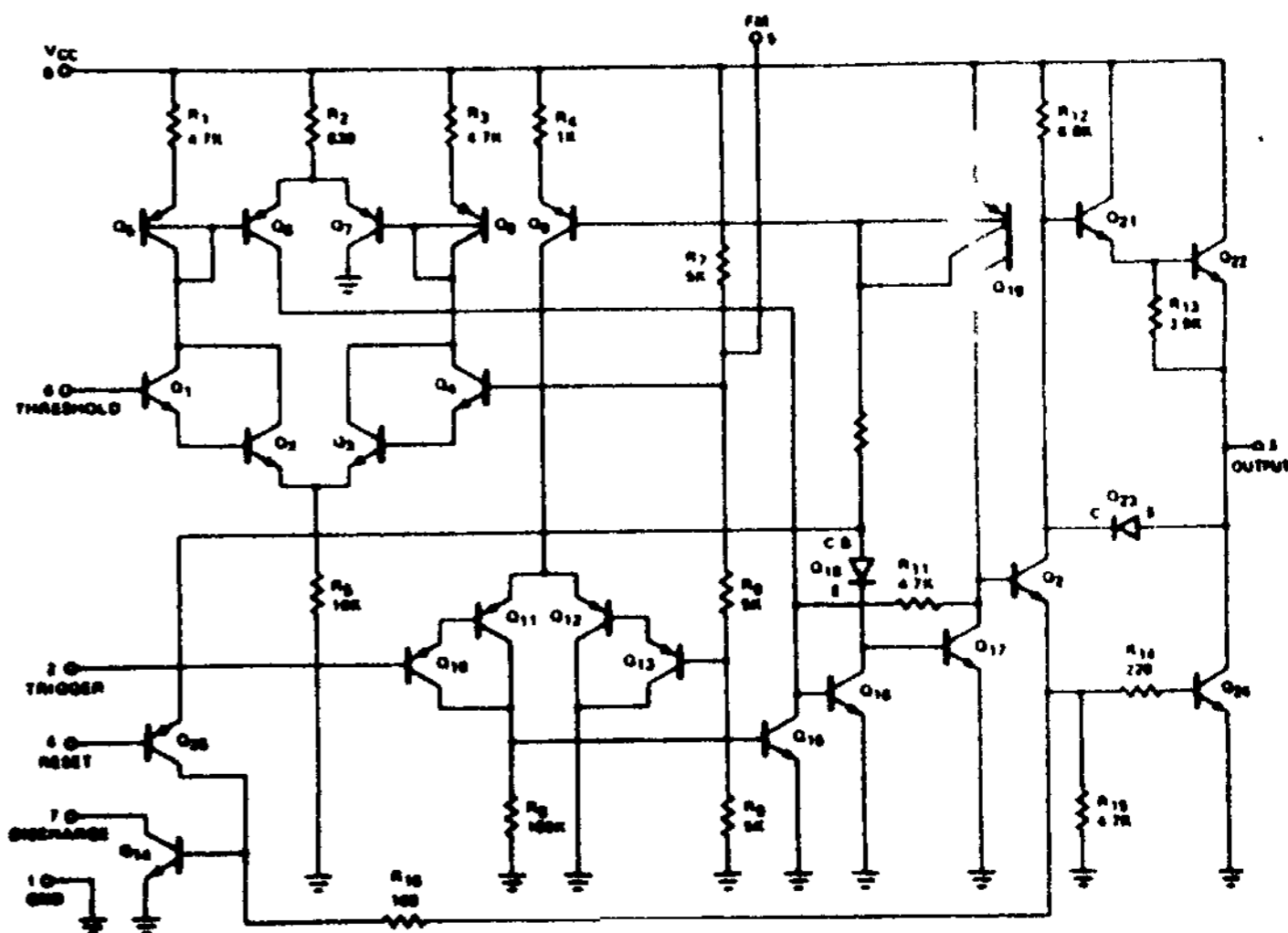
GENERAL DESCRIPTION

The NE/SE 555 monolithic timing circuit is a highly stable controller capable of producing accurate time delays, or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For a stable operation as an oscillator, the free running frequency and the duty cycle are both accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output structure can source or sink large currents or drive TTL circuits.

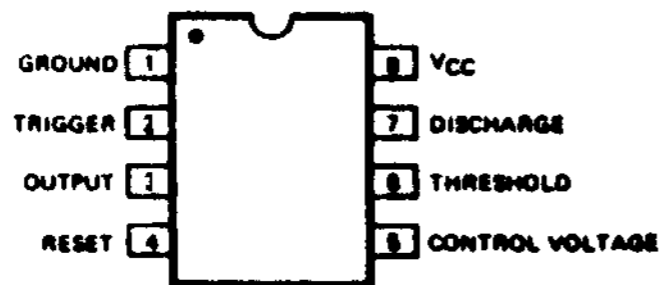
BLOCK DIAGRAM



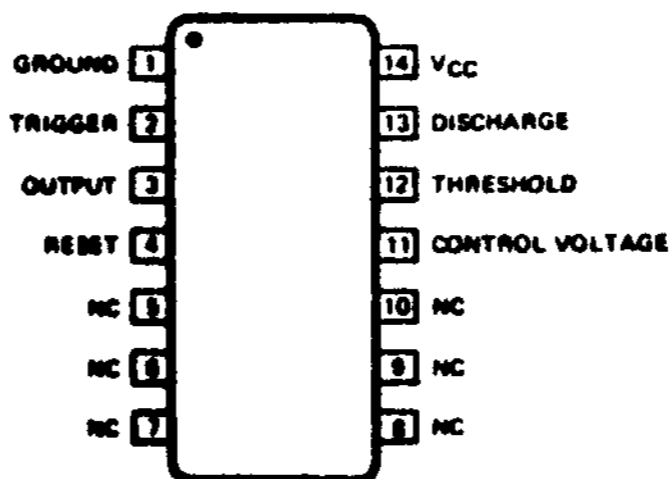
EQUIVALENT CIRCUIT



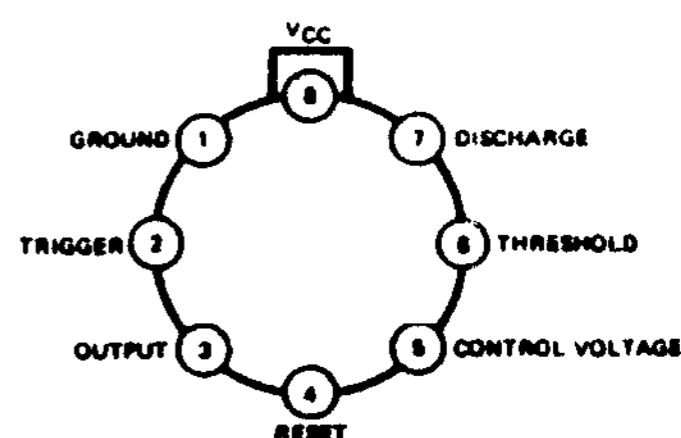
PIN CONFIGURATIONS (TOP VIEW)



8-PIN DIP



14-PIN HERMETIC DIP



TO-99

ABSOLUTE MAXIMUM RATINGS

Supply Voltage	+18V
Power Dissipation	600mW
Operating Temperature Range	
NE555	0°C to +70°C
SE555	-55°C to +125°C
Storage Temperature Range	-65°C to +150°C
Lead Temperature (Soldering, 60 seconds)	+300°C

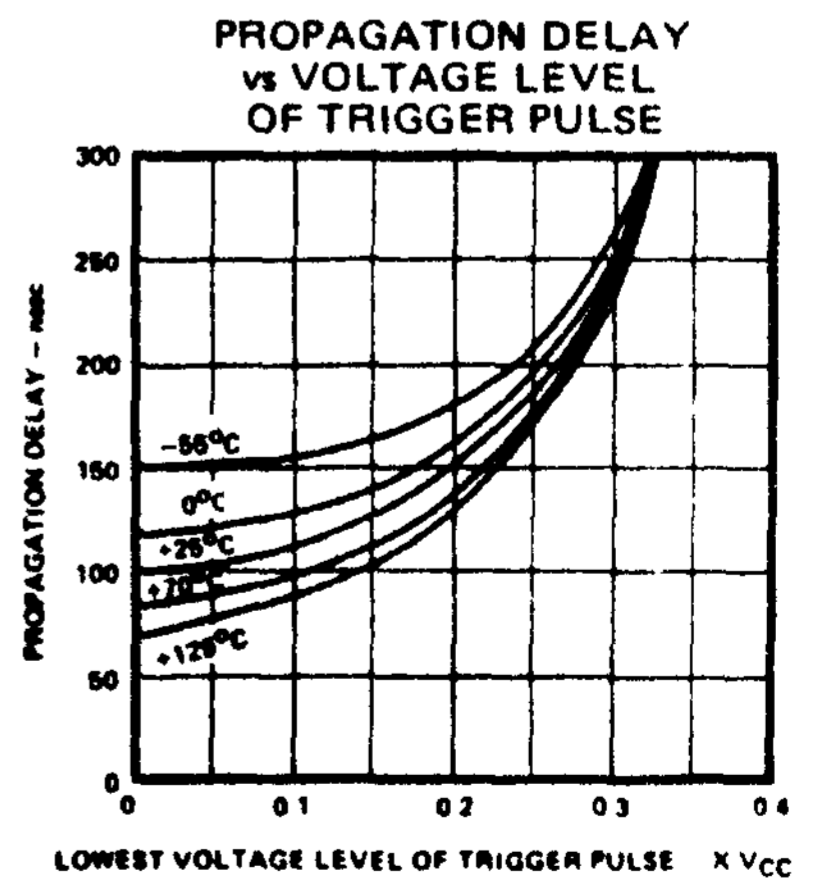
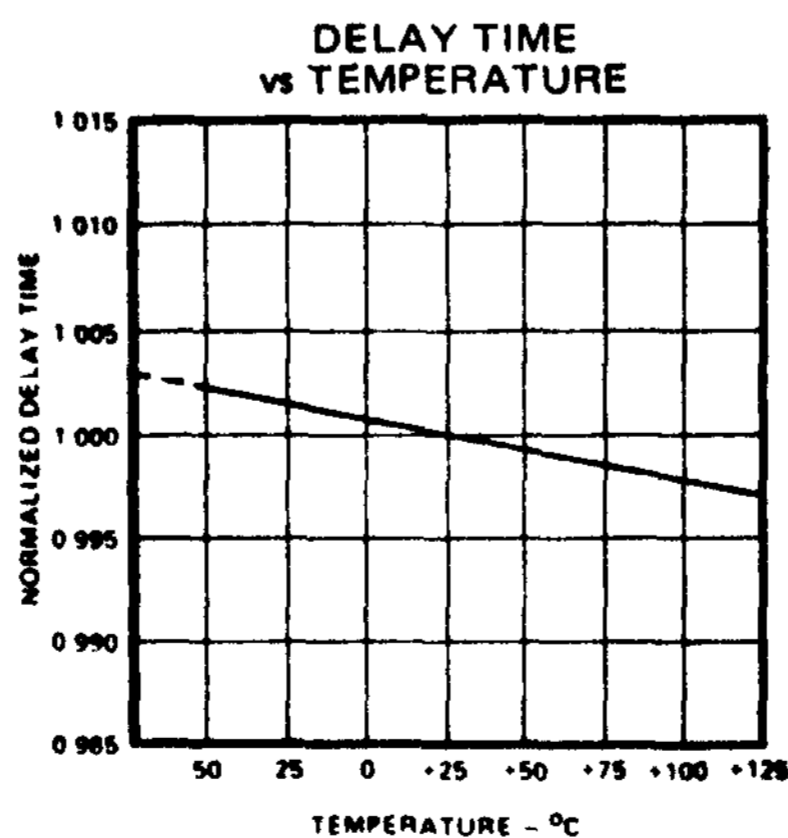
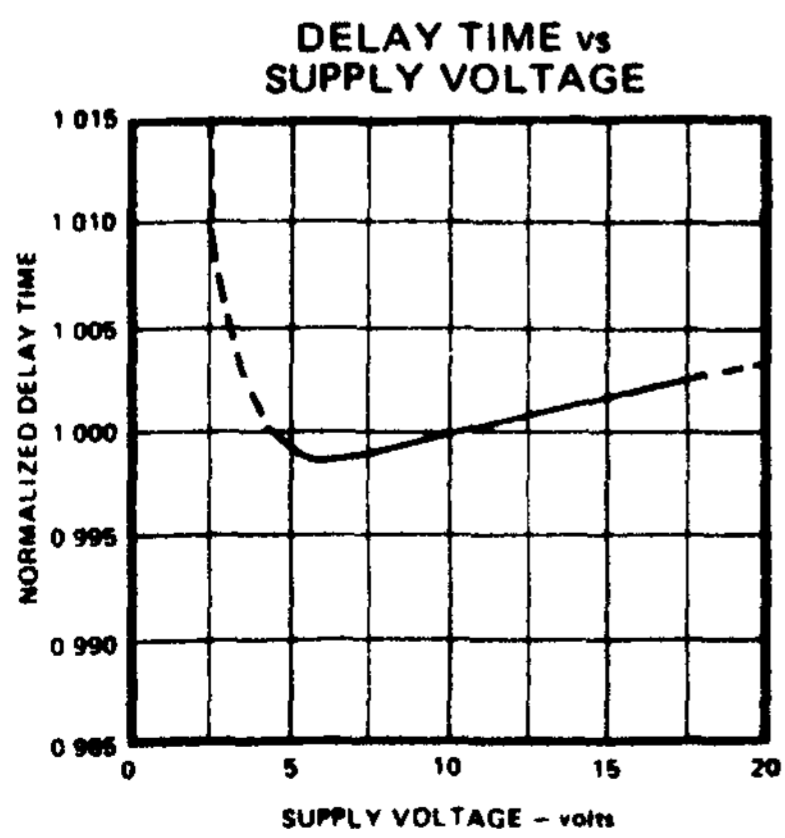
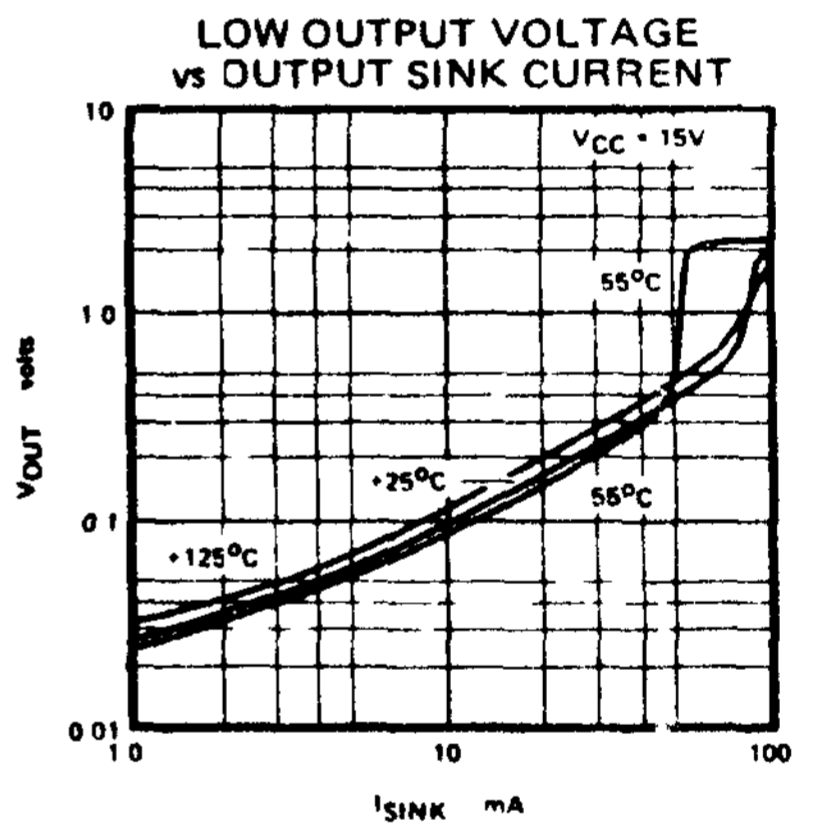
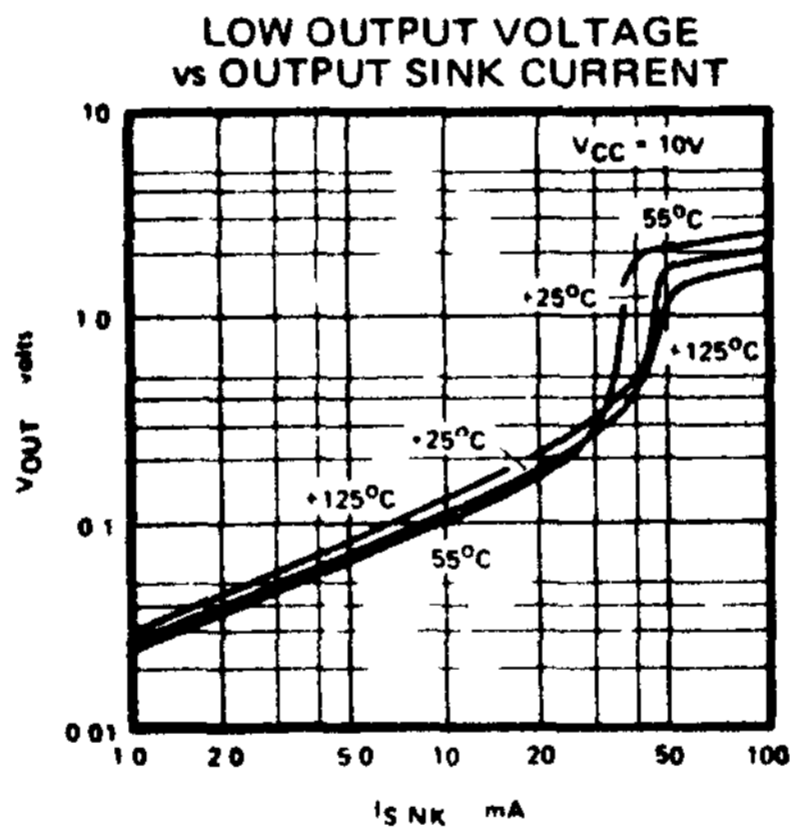
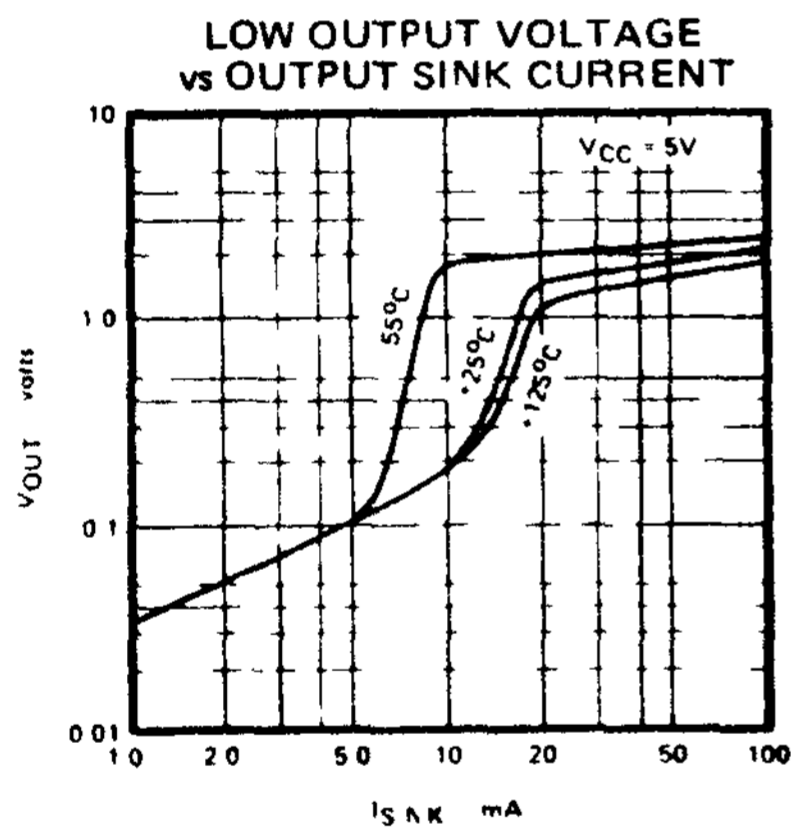
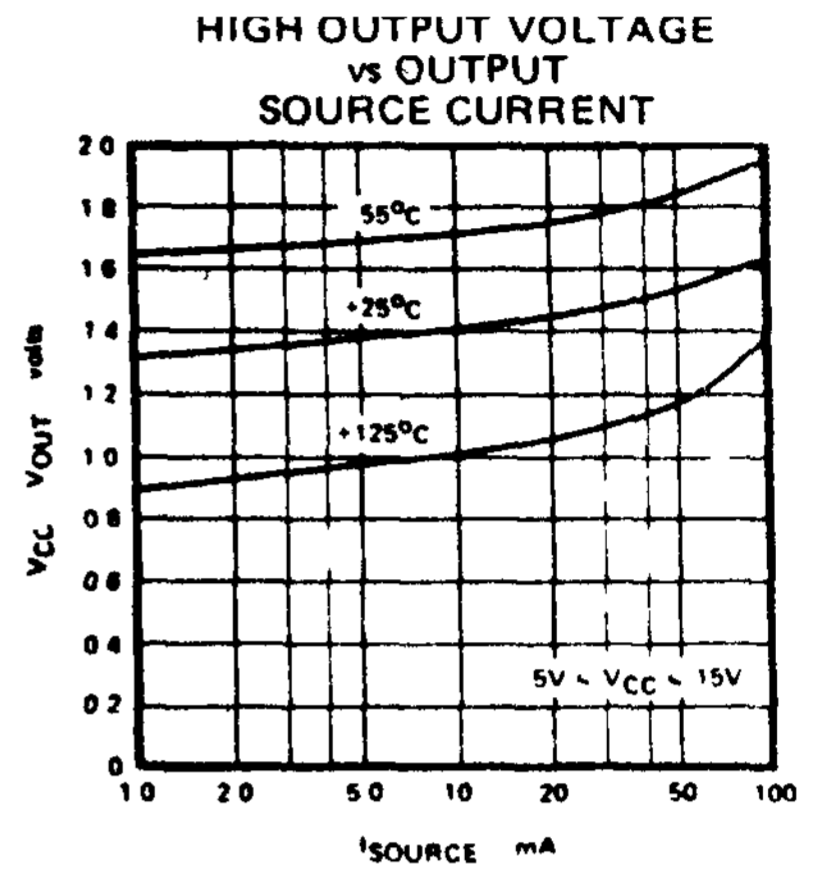
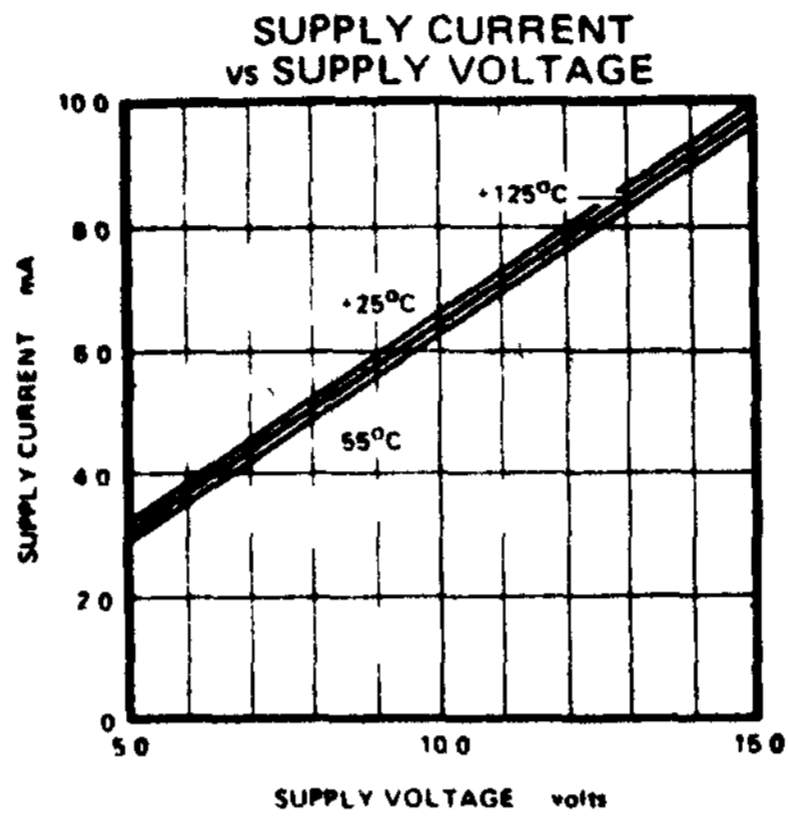
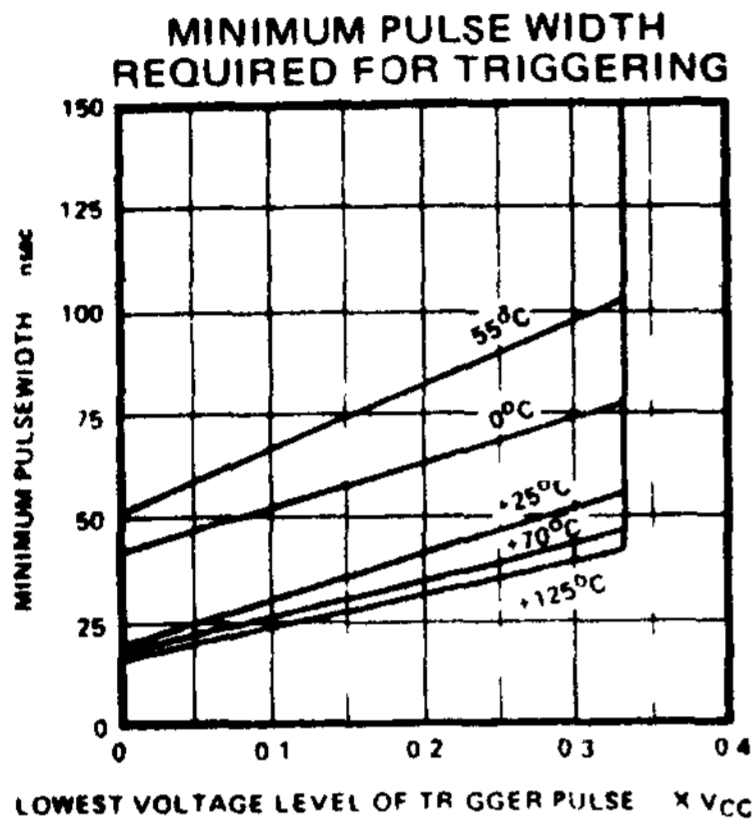
ELECTRICAL CHARACTERISTICS (T_A = 25°C, V_{CC} = +5V to +15 unless otherwise specified)

PARAMETER	TEST CONDITIONS	SE 555			NE 555			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
Supply Voltage		4.5		18	4.5		18	V
Supply Current	V _{CC} = 5V R _L = ∞		3	5		3	6	mA
	V _{CC} = 15V R _L = ∞		10	12		10	15	mA
Timing Error	Low State, Note 1							
	R _A , R _B = 1kΩ to 100kΩ							
	C = 0.1 μF Note 2							
Initial Accuracy			0.5	2		1		%
Drift with Temperature			30	100		50		ppm/°C
Drift with Supply Voltage			0.005	0.02		0.01		%/Volt
Threshold Voltage			2/3			2/3		X V _{CC}
Trigger Voltage	V _{CC} = 15V	4.8	5	5.2		5		V
	V _{CC} = 5V	1.45	1.67	1.9		1.67		V
Trigger Current			0.5			0.5		μA
Reset Voltage		0.4	0.7	1.0	0.4	0.7	1.0	V
Reset Current			0.1			0.1		mA
Threshold Current	Note 3		0.1	25		0.1	25	μA
Control Voltage Level	V _{CC} = 15V	9.6	10	10.4	9.0	10	11	V
	V _{CC} = 5V	2.9	3.33	3.8	2.6	3.33	4	V
Output Voltage Drop (low)	V _{CC} = 15V							
	I _{SINK} = 10mA		0.1	0.15		0.1	25	V
	I _{SINK} = 50mA		0.4	0.5		0.4	75	V
	I _{SINK} = 100mA		2.0	2.2		2.0	25	V
	I _{SINK} = 200mA		2.5			2.5		V
	V _{CC} = 5V							
	I _{SINK} = 8mA		0.1	0.25				V
	I _{SINK} = 5mA					25	35	V
Output Voltage Drop (high)	I _{SOURCE} = 200mA		12.5			12.5		V
	V _{CC} = 15V							
	I _{SOURCE} = 100mA							
	V _{CC} = 15V	13.0	13.3		12.75	13.3		V
	V _{CC} = 5V	3.0	3.3		2.75	3.3		V
Rise Time of Output			100			100		nsec
Fall Time of Output			100			100		nsec

NOTE 1: Supply Current when output high typically 1 mA less.

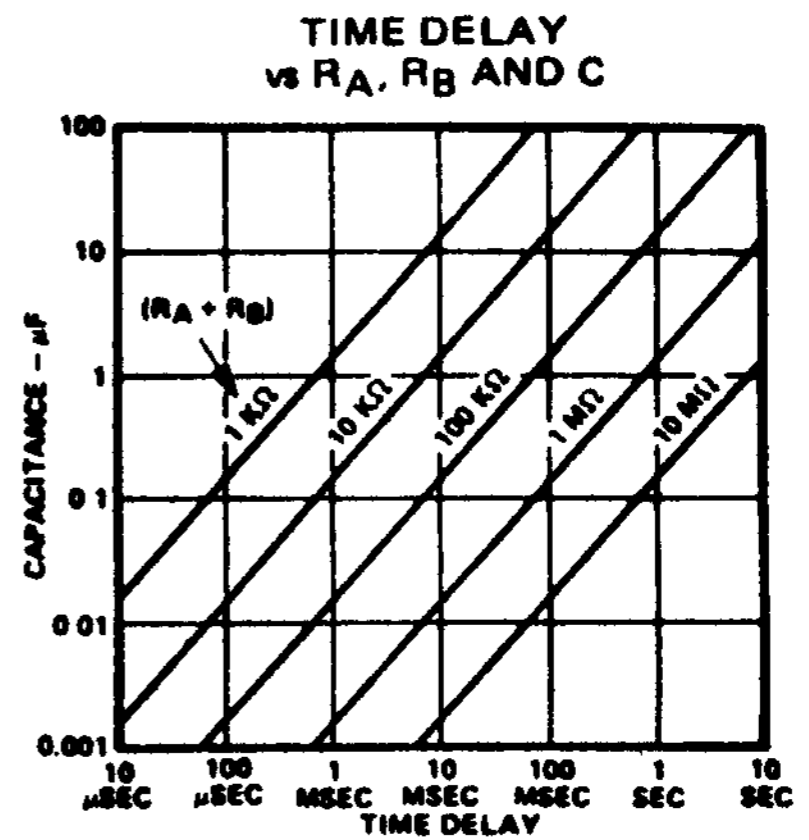
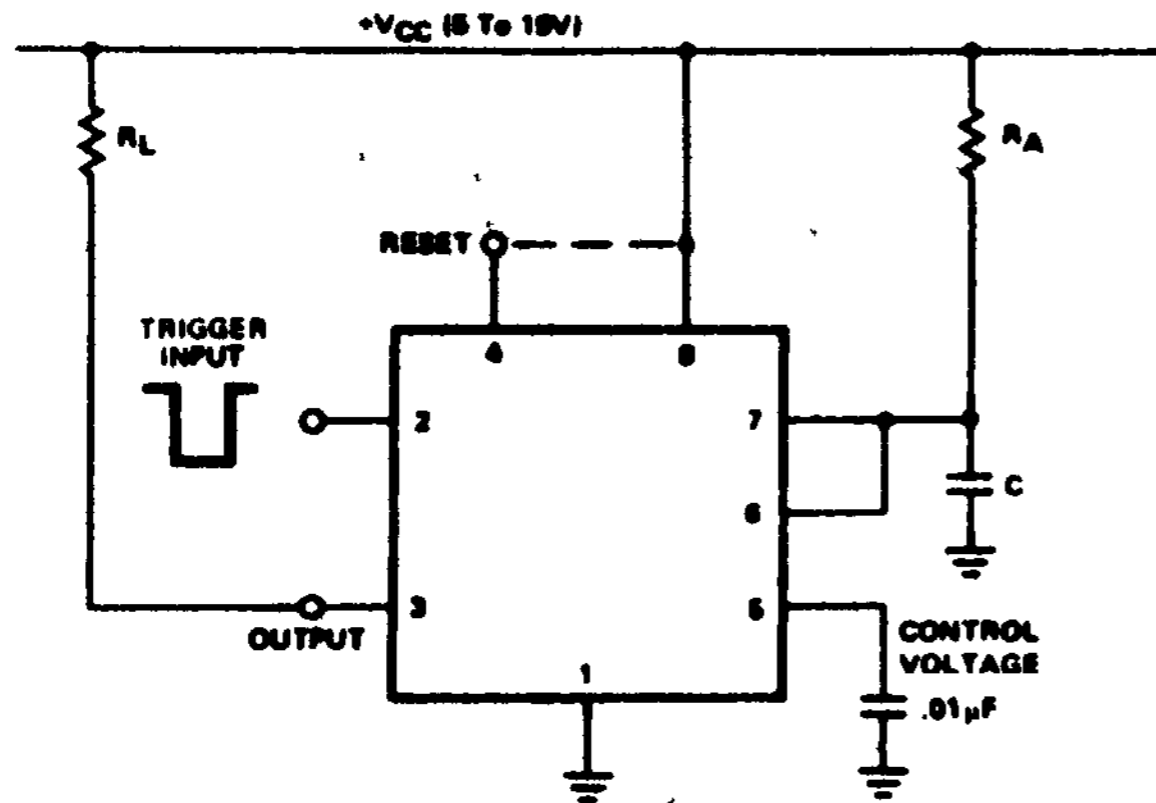
NOTE 2: Tested at V_{CC} = 5V and V_{CC} = 15V.NOTE 3: This will determine the maximum value of R_A + R_B. For 15V operation, the max. total R = 20 megohm.

TYPICAL CHARACTERISTICS



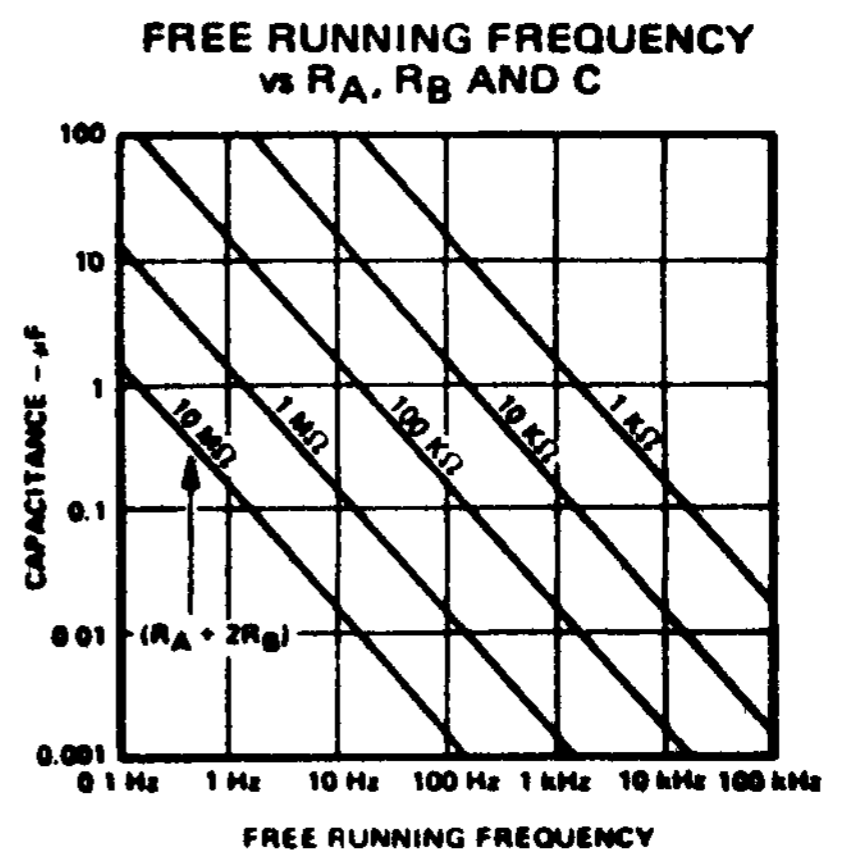
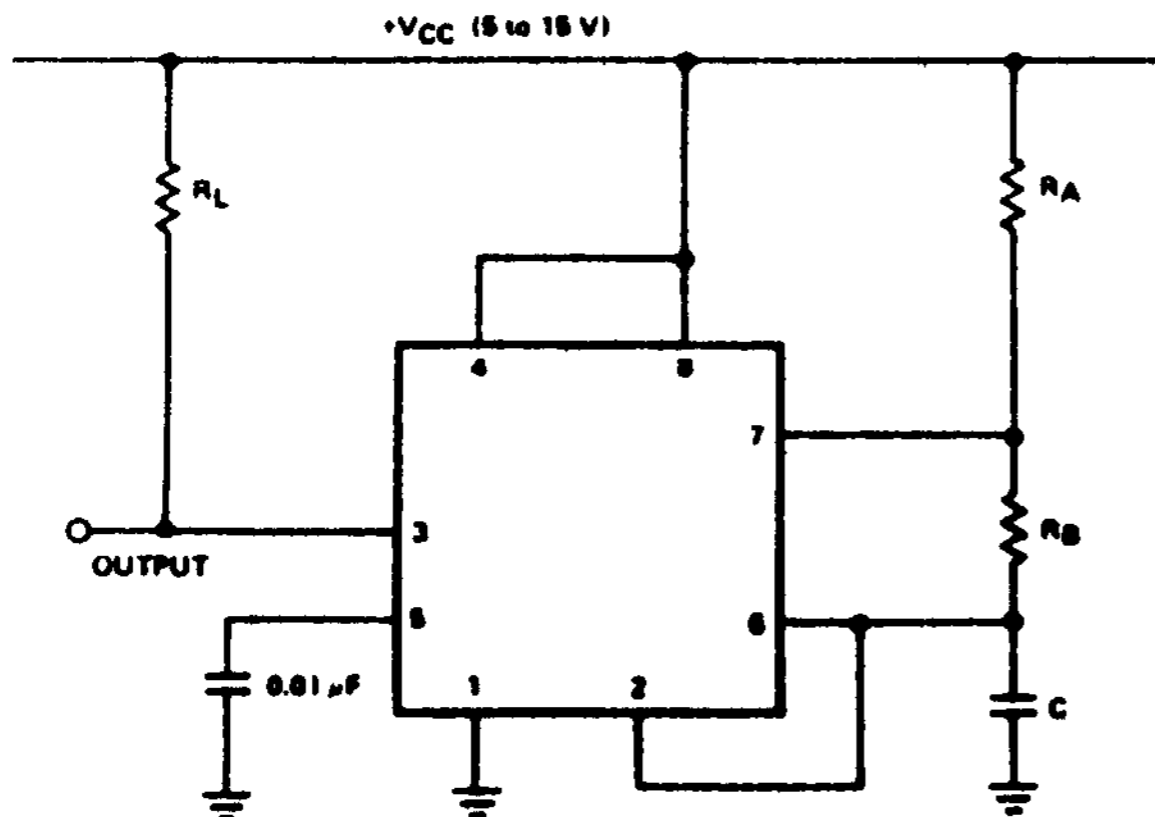
APPLICATION INFORMATION

MONOSTABLE OPERATION



In this mode of operation, the timer functions as a one-shot. Initially the external capacitor (C) is held discharged by a transistor inside the timer. Upon application of a negative trigger pulse to pin 2, the flip-flop is set which releases the short circuit across the external capacitor and drives the output high. The voltage across the capacitor, now, increases exponentially with the time constant $\tau = R_A C$. When the voltage across the capacitor equals $2/3 V_{CC}$, the comparator resets the flip-flop which in turn discharges the capacitor rapidly and drives the output to its low state.

ASTABLE OPERATION



The circuit can also be connected so as to trigger itself and free run as a multivibrator. The external capacitor charges through R_A and R_B and discharges through R_B only. Thus the duty cycle may be precisely set by the ratio of these two resistors. In this mode of operation, the capacitor charges and discharges between $1/3 V_{CC}$ and $2/3 V_{CC}$. As in the triggered mode, the charge and discharge times, and therefore the frequency are independent of the supply voltage.

The frequency of oscillation is given by: $f = \frac{1}{T} = \frac{1.46}{(R_A + 2R_B) C}$

ORDERING INFORMATION

TYPE	ORDER PART NUMBER	TEMPERATURE RANGE	PACKAGE
555	NE 555 V	0°C to 70°C	8 Pin DIP
555	NE 555 T		TO-99
555	SE 555 V	-55°C to 125°C	14 Pin Hermetic DIP
555	SE 555 T		TO-99

Minimalprojekt "Bit und Byte - Wir bauen einen Computer"

Vorschlag zu einer Durchführung im Wahlpflichtfach-Bereich
(9./10. Schuljahr)

Die hier angeführten Themen sind hierarchisch geordnet, bedürfen jedoch unbedingt spezifischen Ergänzungen entsprechend den in den einzelnen Bundesländern zu erwartenden Richtlinien.

- | | |
|---|----------|
| 1. Auslöten von elektronischen Bauteilen | 0,5 Std. |
| 2. Bauteilekunde, auf einzelne
Unterrichtsstunden verteilt | 2 Std. |
| 3. Vorführung der 6-teiligen WDR-Fernsehsendung
zu je 15 Minuten auf einzelne Stunden verteilt | 1,5 Std. |
| 4. Bau des WDR-1-Bit-Computers
(10 Schüler pro Computern inkl. Tastatur) | 6 Std. |
| 5. Inbetriebnahme des WDR-1-Bit-Computers | 6 Std. |
| 6. Programmierung des WDR-1-Bit-Computers | 4 Std. |

Mögliche Weiterführung:

7. Steuerung der Peripherie
8. Bau und Programmierung eines Robotermodells
9. Besichtigung eines Betriebes, der mit
Industrierobotern produziert
10. Soziokulturelle Aspekte der industriellen
Revolution

BEISPIEL: DURCHFÜHRUNG EINER UNTERRICHTSEINHEIT IM WPF-BEREICH (9./10. SCHULJAHR) IN EINER GESAMTSCHULE

Ein möglicher didaktischer Aufbau zu "Bit und Byte - Wir bauen einen Computer"

Die vorliegende Unterrichtseinheit wurde im Wahlpflichtfach-Unterricht der Jahrgänge 9 und 10 an der Gesamtschule Ost in Bremen durchgeführt. Mit 18 Schülern wurden in drei Gruppen von je 6 Schülern drei WDR-1-Bit-Computer gebaut. Es ist ratsam, nicht mehr als zehn Schüler an einem Computer arbeiten zu lassen! Jeder Schüler war für eine Platine verantwortlich. Das Material wurde getrennt nach Platinen bereitgestellt. Während der gesamten Unterrichtseinheit waren die Schüler sehr motiviert.

Zeitlicher Ablauf:

1. Allgemeine Einführung	1 Std.
2. Diskussion von Problemen unserer Zeit (Computerisierung, Qualifikation, Datenschutz, Arbeitsmarkt)	2 Std.
3. Die elektronische Datenverarbeitung	1 Std.
4. Auslöten von elektronischen Bauteilen	1 Std.
5. Lehrgang "Diode und Transistor"	2 Std.
6. Aufbau eines Halbleiters	1 Std.
7. Der Farbcode von Widerständen	1 Std.
8. Test	1 Std.
9. Vorführung des WDR-Films	1,5 Std.
10. Bau des WDR-1-Bit-Computers	ca. 10 Std.
11. Programmieren	ca. 7 Std.
12. Bau von Peripheriegeräten	

Einige Hilfen zur Durchführung

1. Allgemeine Einführung

Zeit: 1 Std.

- Unterrichtsgespräch
- Frage: Was bedeute die Abkürzung EDV?
 Antwort: Elektronische Datenverarbeitung
- Frage: Was bedeutet Computer?
 Antwort: Das Wort Computer kommt aus dem Lateinischen "computare = rechnen" und heißt "Rechner".
- Frage: Wo sind Computer eingesetzt?
 Antwort: Antworten als Tafelbild sammeln.

Computertyp	Zweck	Ausgeführte Arbeit
Bordcomputer	Blindflug	Steuern
Großrechner	Verwaltungsarbeit	z.B. Lohnabrechnung
Heimcomputer	Unterhaltung	Spiele

2. Texte: Anregungen zur Diskussion zu Problemfragen unserer Zeit

Zeit: 2 Std.

Text lesen und abschnittsweise besprechen. Sozial-umweltkritische Gedankengänge werden angesprochen.

3. Die "elektronische Datenverarbeitung"

Zeit: 1 Std.

- Unterrichtsgespräch:
- a) Historische Entwicklung (siehe Vorbemerkungen)
 - b) Grundfunktionen der EDV:
 Eingabe Verarbeitung Ausgabe
 - c) Rechnen mit Dualzahlen
- Tafelbild:
 Einüben Dual in Dezimal und Dezimal in Dual

Umrechnung

Stellenwertigkeit	128	64	32	16	8	4	2	1	Dez
Dualzahl	0	0	1	0	0	1	1	1	= 39
	1	0	0	0	1	0	0	1	= 137

4. Auslöten von elektronischen Bauteilen

Zeit: 1 Std.

- a) Bestückte Leiterplatten werden "ausgeschlachtet"
 z.B. aus alten Fernsehgeräten
- b) Die Bauteile werden benannt und einsortiert.
 Widerstände (große für höhere Belastung)
 Trimmwiderstände (einstellbar von 0 bis Aufdruckwert)
 Kondensatoren (Metallpapier, Elektrolyt mit Polung)
 Spulen (für Hochfrequenz, Übertrager, Trafo)
 Dioden (Farbringseite ist Kathode = Minuspol)
 Transistoren (Bauelemente mit drei oder vier Anschlüssen)

6. Lehrgang Diode und Transistor

Zeit: 2 Std.

- a) Die Schüler sortieren elektronische Bauteile. Die Bauteile werden dem Arbeitsbogen 1 zugeordnet.
- b) Die Schaltsymbole werden an die Tafel gezeichnet und geübt.
- c) Auf Arbeitsbogen 2 wird zur Wiederholung der "einfache Stromkreis" gezeichnet. Eine Diode wird in den Stromkreis geschaltet, einige Glühlampen leuchten. Die Diode wird umgedreht. Die Schüler sollen die Funktion der Diode als elektrisches Ventil erkennen.
- d) Der Arbeitsbogen 3 wird bearbeitet (Auszug aus "Elektronik", IPN, Kiel). Ergebnis: Es befinden sich zwei Dioden im Transistor. Der Lehrer erläutert stark vereinfacht: Wenn ein sehr kleiner Basis-Emitter-Strom fließt, dann hebt sich die Diode in der Basis-Kollektor-Strecke auf. Die Elektronen aus dem Kollektor werden mitgerissen. Jetzt kann ein viel größerer Strom vom Kollektor zum Emitter fließen. Nimmt der recht kleine Basisstrom zu, steigt der sowieso schon größere Kollektor-Emitter-Strom ebenfalls an. Hierauf beruht die verstärkende Wirkung des Transistors.

Auf den nächsten Seiten finden sich die folgenden Arbeitsbögen und ein Test:

Die Löt- und Bestückungstechnik in der Elektronik

Stückliste

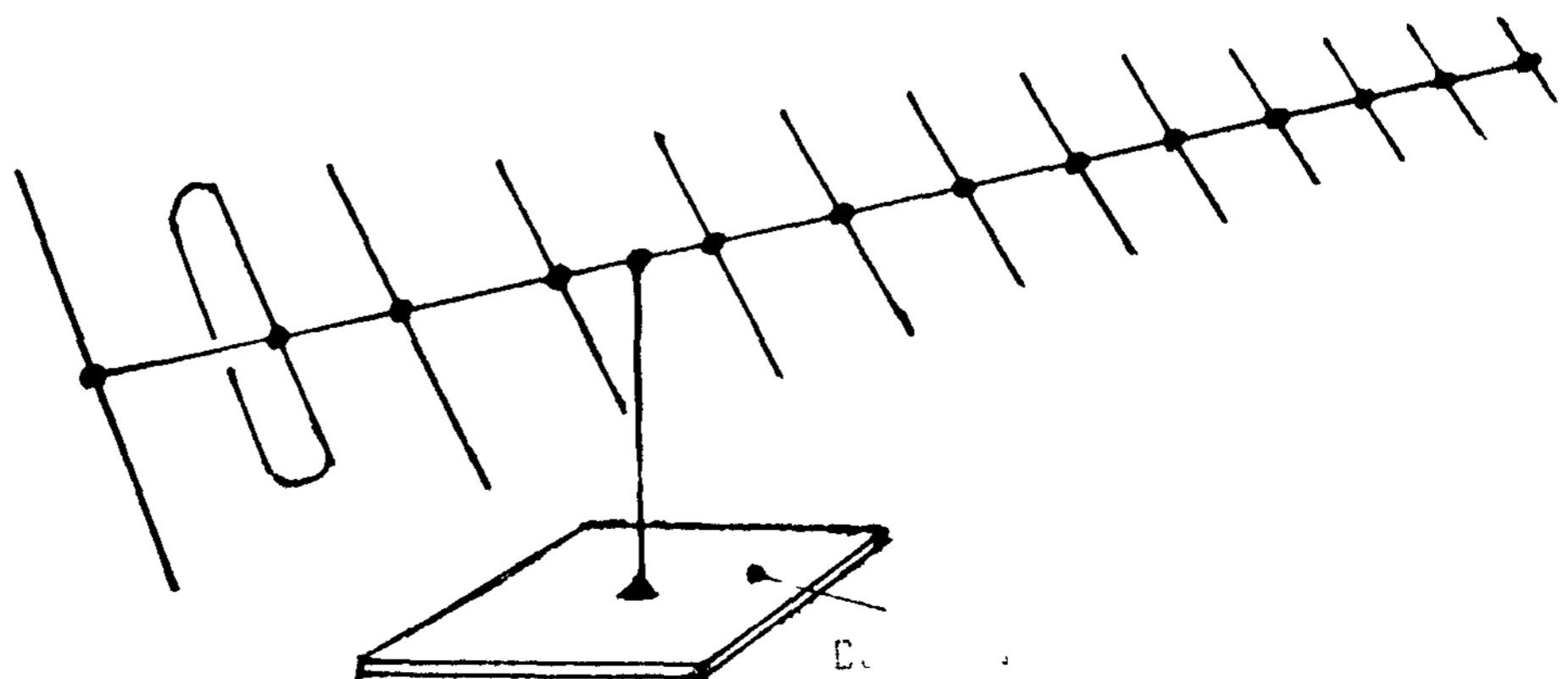
Stromkreis mit Diode

Diode und Transistor

Aufbau eines Halbleiters

Der NPN-Transistor

Test



Lötübung Antenne

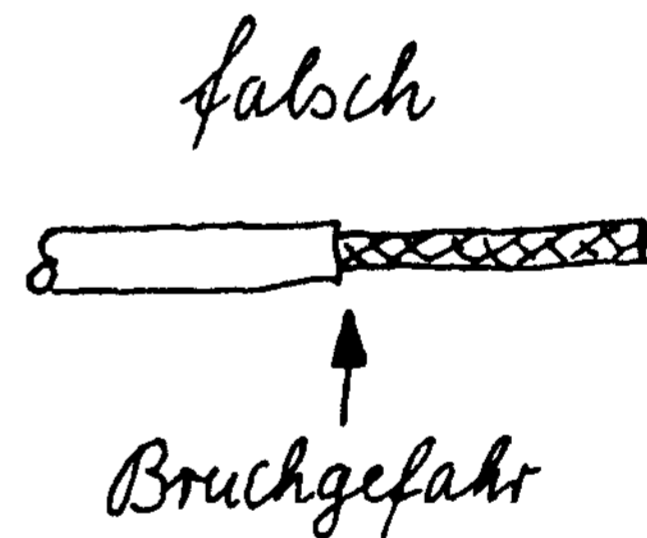
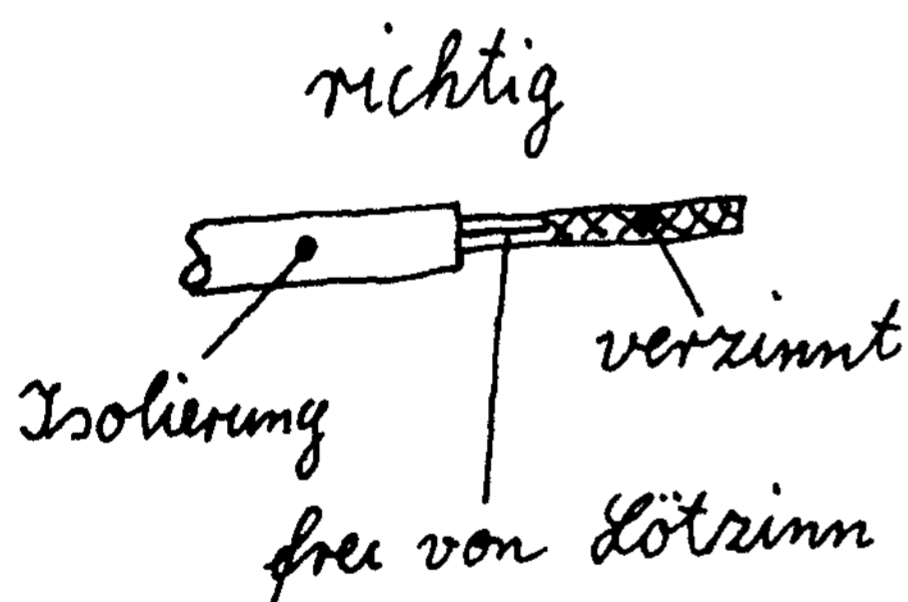
Die Löt- und Bestückungstechnik in der Elektronik

Zum Löten benötigen wir einen Lötkolben von 15 bis 30 Watt. Bei der Platinenbestückung eignet sich sehr gut der sogenannte 16 W Mini-Tip Lötkolben. Es wird Lötzinn mit Kolophonium von 1,5 mm Durchmesser verwendet. Das Kolophonium schützt die Lötstelle beim Erhitzen vor Oxydation, d.h. der Sauerstoff in der Luft kann sich nicht mit dem Kupfer der Platine verbinden, da das Kolophonium als Flußmittel einen Schutzfilm bildet.

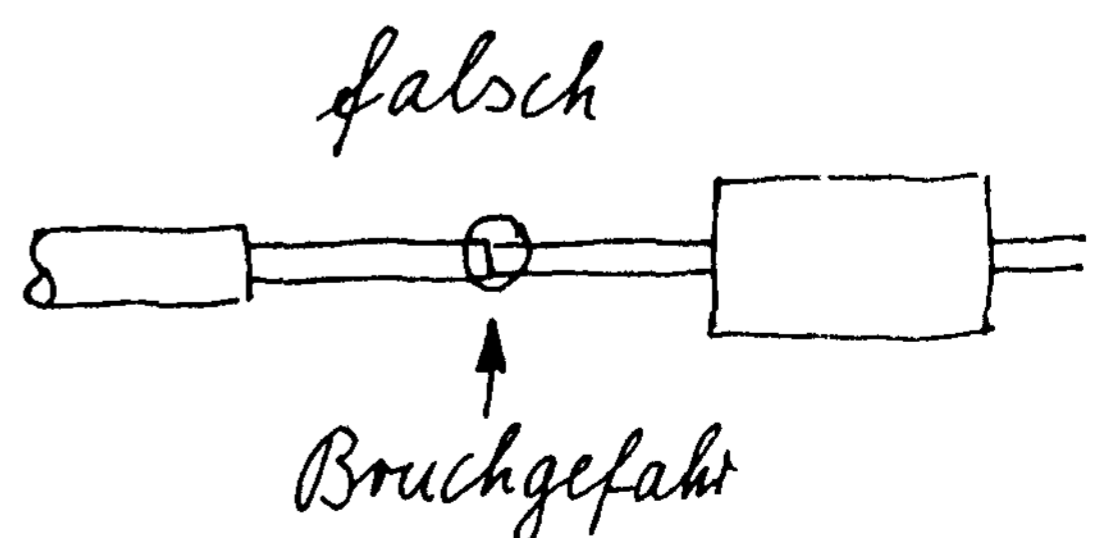
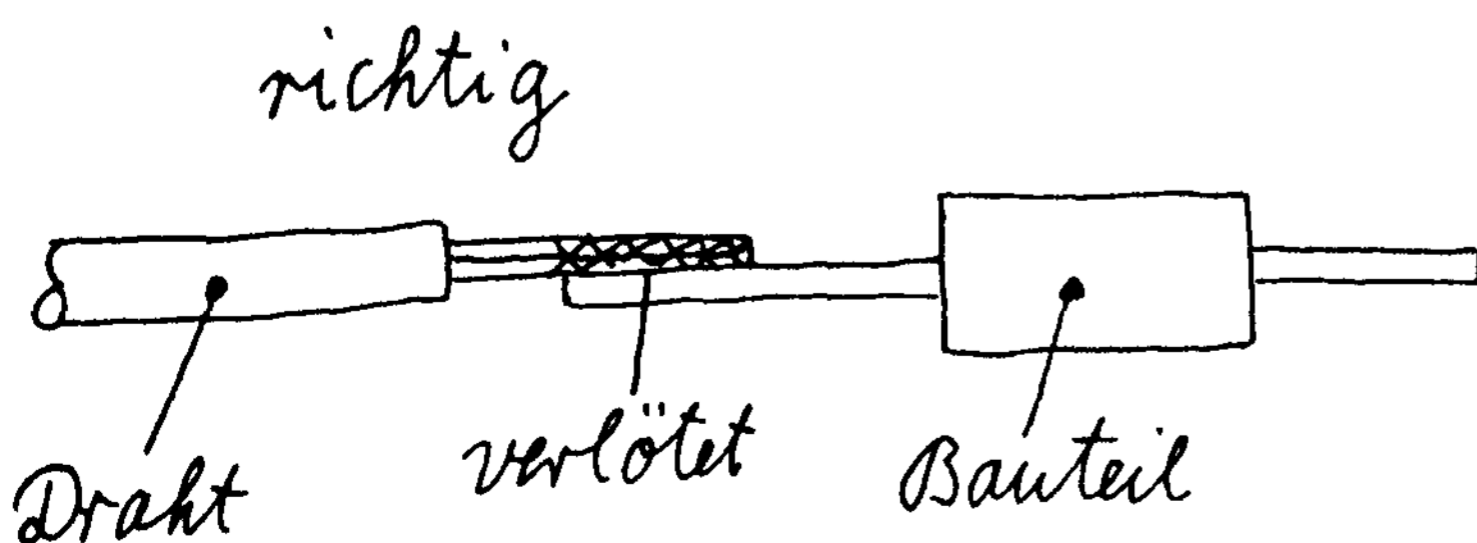
Löten will gelernt sein!

1. Es ist darauf zu achten, daß die Lötkolbenspitze immer verzinnt ist. Durch Abreiben mit einem feuchten Schwamm oder Lappen müssen der Zunder und das überschüssige Kolophonium entfernt werden. Dieser Reinigungsvorgang ist während des Lötens öfters zu wiederholen.

2. Zum Verzinnen von isolierten Drahtenden wird zuerst die Isolierung entfernt. Dies kann mit einem Seitenschneider geschehen, indem die Isolierung am Umfang zweimal eingeschnitten und dann abgezogen wird. Sind die frei liegenden Litzendrähte oxydiert, müssen sie mit einem Messer blank gekratzt und anschließend verdreht (eingedreht) werden. Beim Verzinnen wird die Lötkolbenspitze unter den Draht gelegt und das Lötzinn auf dem Draht leicht hin- und hergerieben, bis es schmilzt und die Litzendrähte umfließt. Dann werden Lötzinn und Draht in Richtung Drahtende weggezogen. Beim Verzinnen von Litzendrähten darf das Lötzinn nicht ganz bis an die Isolierung fließen, da sonst dort eine große Bruchgefahr entsteht.

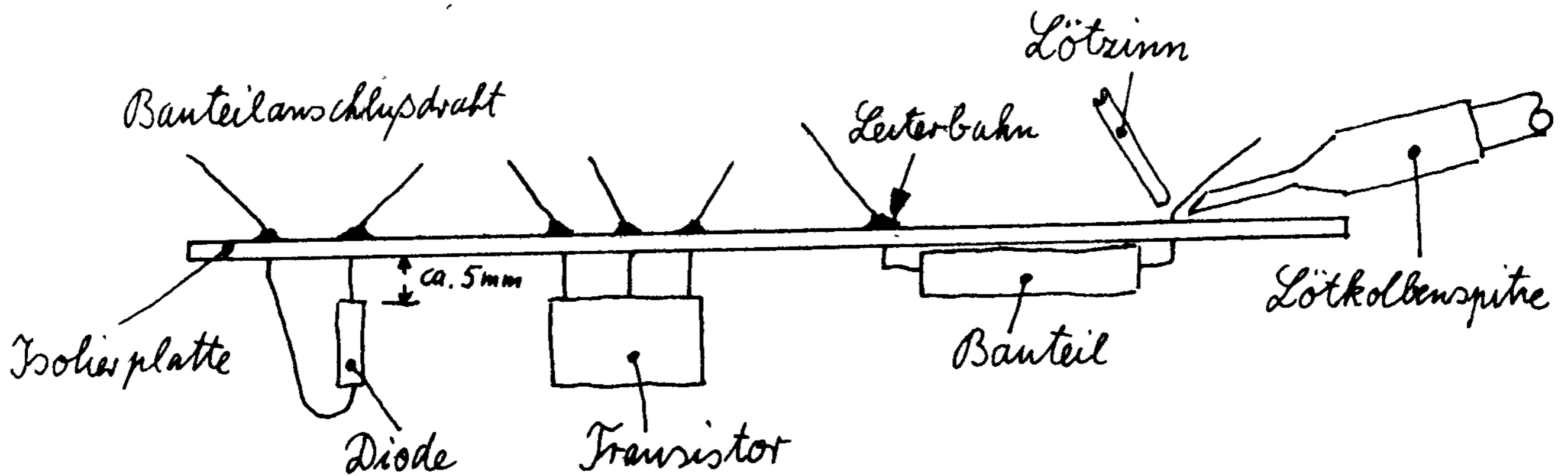


3. Werden zwei Drähte oder ein Bauteilanschluß mit einem Draht verlötet, müssen vorher die beiden Teile getrennt verzinnt werden. Dann werden die verzinnten Enden zusammengelegt und mit dem Lötkolben unter Zuführung von Lötzinn verschmolzen. Die beiden Teile dürfen bis zum Erkalten des Lötzinns nicht bewegt werden.



4. Bei der Bestückung der Leiterplatten mit den Bauteilen werden die Anschlußdrähte auf der Leiterbahnseite um etwa 45° nach außen scharf abgebogen. Die Bauteile liegen auf der isolierenden Seite der Platine auf. Ausgenommen davon sind Transistoren und Dioden, die einen Abstand von ca. 5 mm zur Platine haben müssen, damit die Kristalle von der Löttemperatur (ca. 200 bis 250 Grad) nicht zerstört werden. Die Widerstandsbezeichnungen sollten alle in einer Richtung liegen. Bei den Kondensatoren ist beim Bestücken unter Umständen auf die richtige Polung zu achten.

5. Mit der LötKolbenspitze werden Leiterbahn und Bauteileanschlußdraht gleichzeitig berührt. Das Lötzinn wird zum Teil auf der erhitzten Leiterbahn und zum Teil am Anschlußdraht abgeschmolzen, so daß eine gute Zinnverbindung entsteht. Der ganze Lötvorgang darf vier Sekunden nicht überschreiten, da sonst die Bauteile zu heiß werden und sich die Leiterbahnen von der Platine lösen können. Wenn einige Lötstellen schwer zugänglich sind, werden die Bauteileanschlußdrähte am besten mit dem Seitenschneider vorher gekürzt. Nach dem Löten werden grundsätzlich alle überstehenden Anschlußdrähte abgezwickelt, um auszuschließen, daß sich ein Anschlußdraht ungelötet unter einer Lötugel verbirgt. Das überschüssige Kolophonium auf den Lötstellen wird nicht entfernt, es schützt die Lötstelle vor Oxydation und wirkt sich vorteilhaft bei einem eventuellen Auslöten der Bauteile aus.


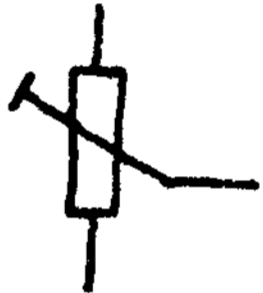


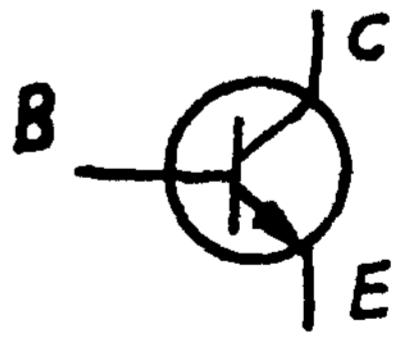




Lehrung: Funktion von Diode und TransistorArbeitsbogen 1

Name: _____

Klasse: _____ Datum: _____

Stückliste

Stück	Benennung	Schaltsymbol
	Widerstand	
	Trimm-Widerstand	
	Lichtempfindlicher Widerstand (LDR)	
	Diode	
	Transistor $B \cong$ Basis $E \cong$ Emitter $C \cong$ Collector	
	Batterie	
	Glühlampe	

Lehrung: Funktion von Diode und Transistor

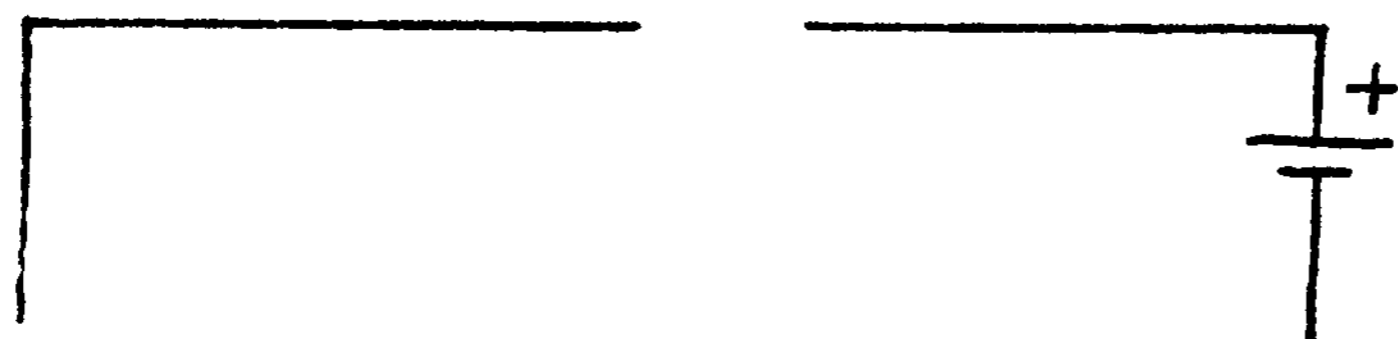
Arbeitsbogen 2

Name: _____

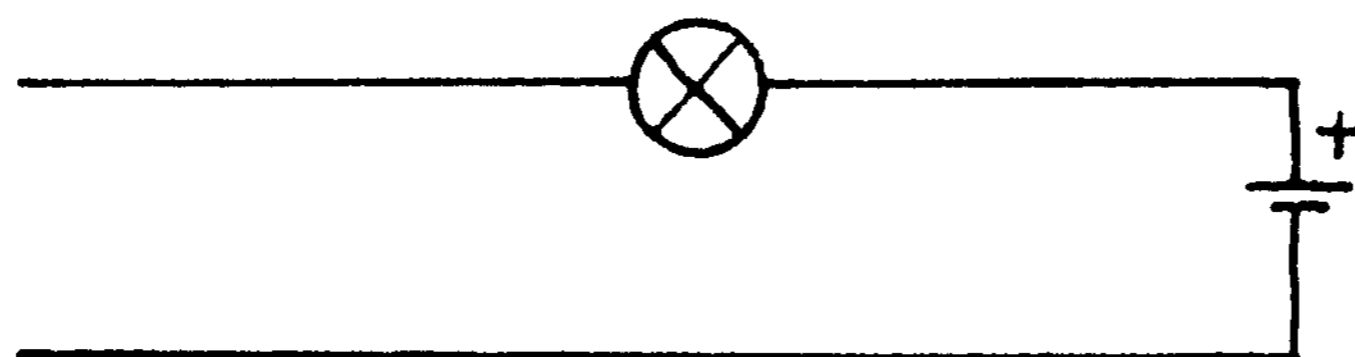
Klasse: _____

Datum: _____

Der einfache Stromkreis

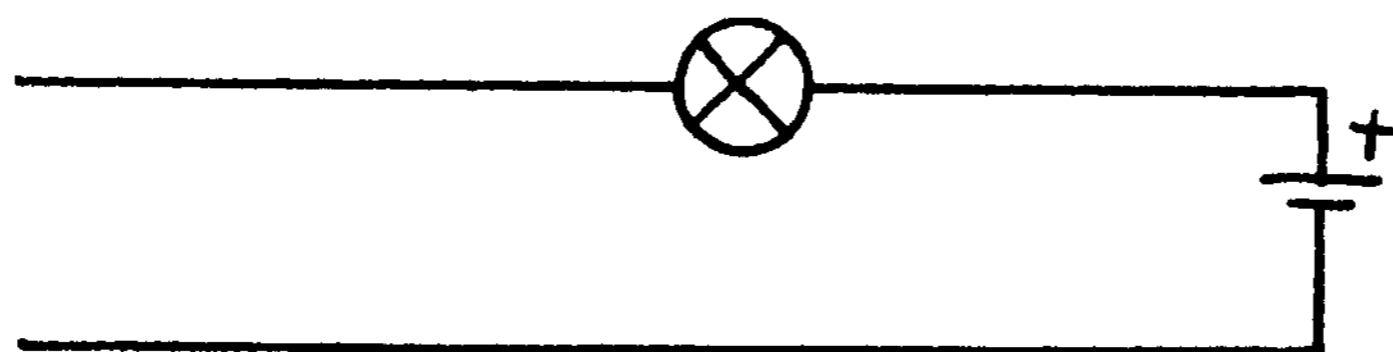


Diode im Stromkreis



Beobachtung: _____

Diode im Stromkreis



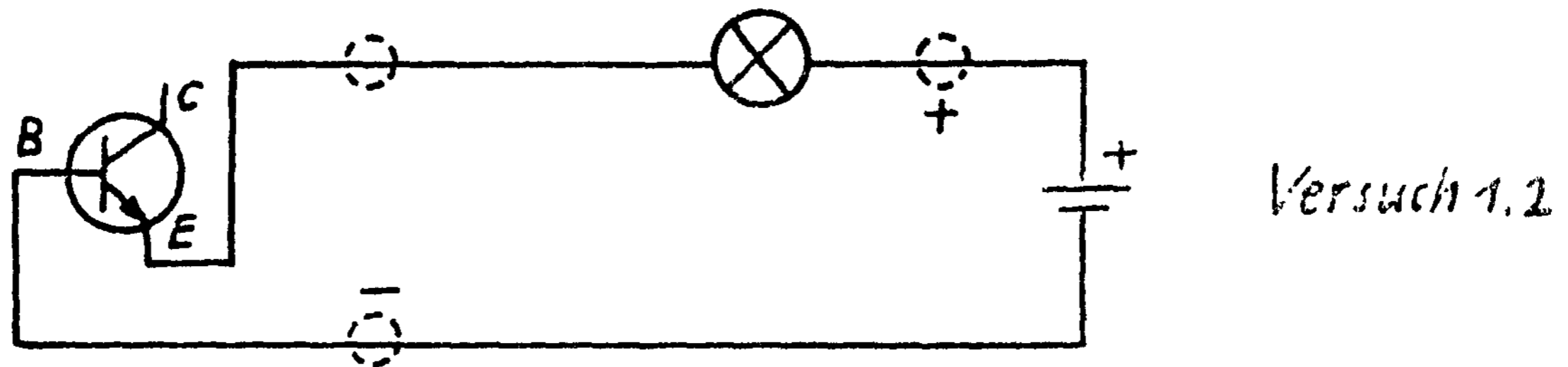
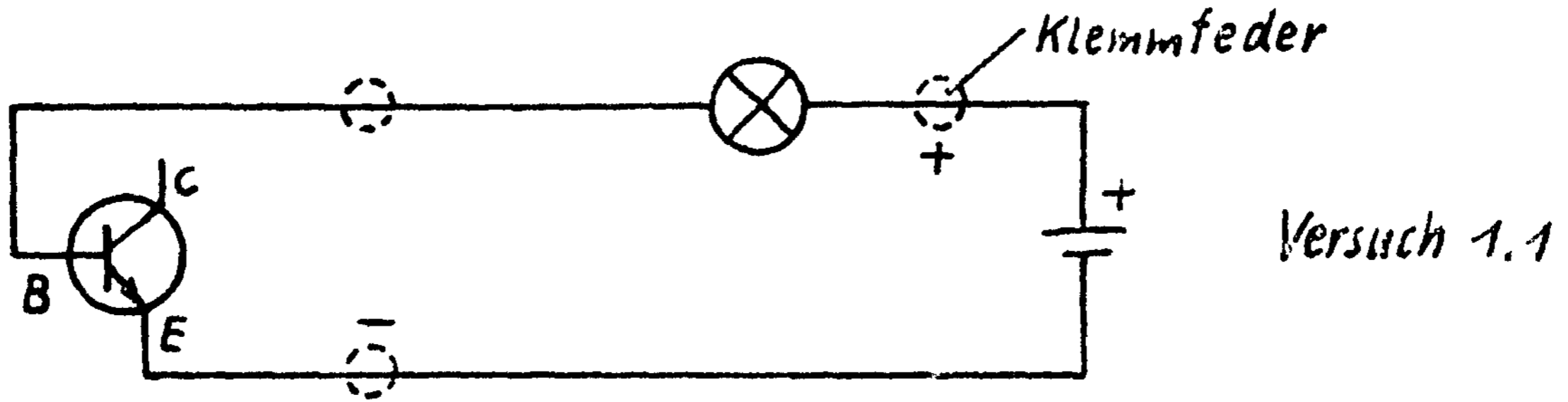
Beobachtung: _____

Lehrgang: Funktion von Diode und Transistor

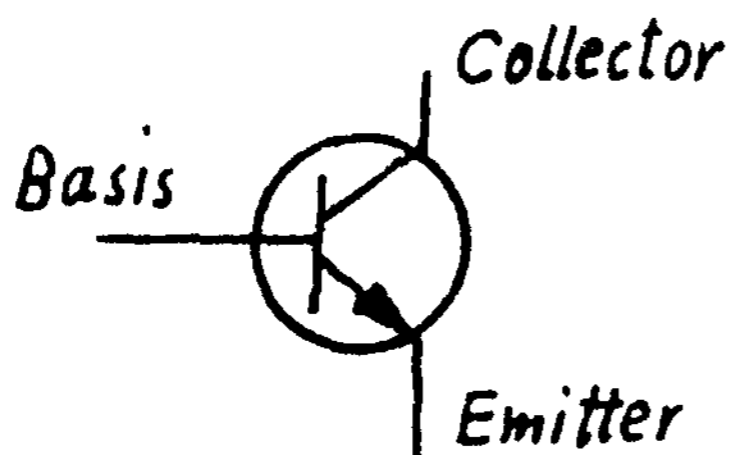
Arbeitsbogen 3

Name: _____

Klasse: _____ Datum: _____

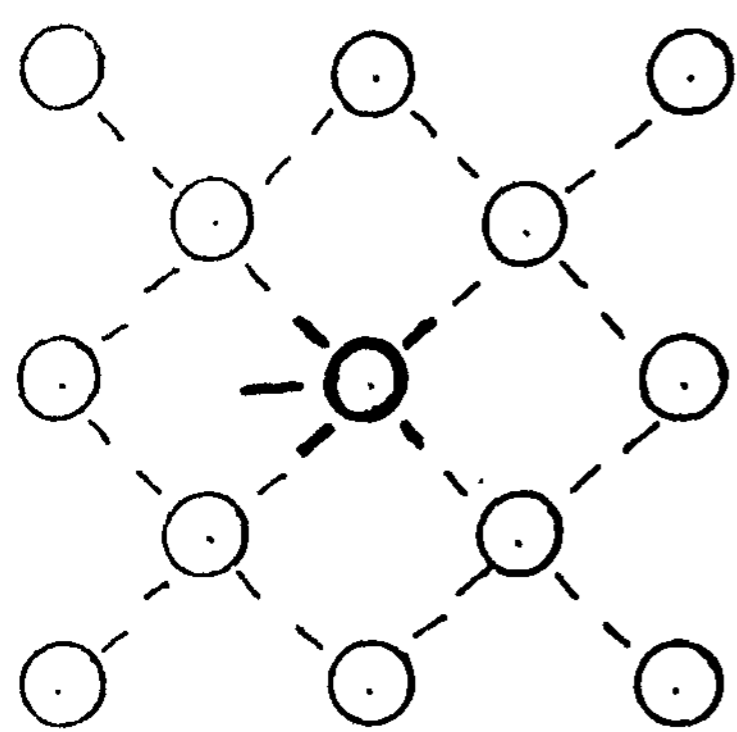
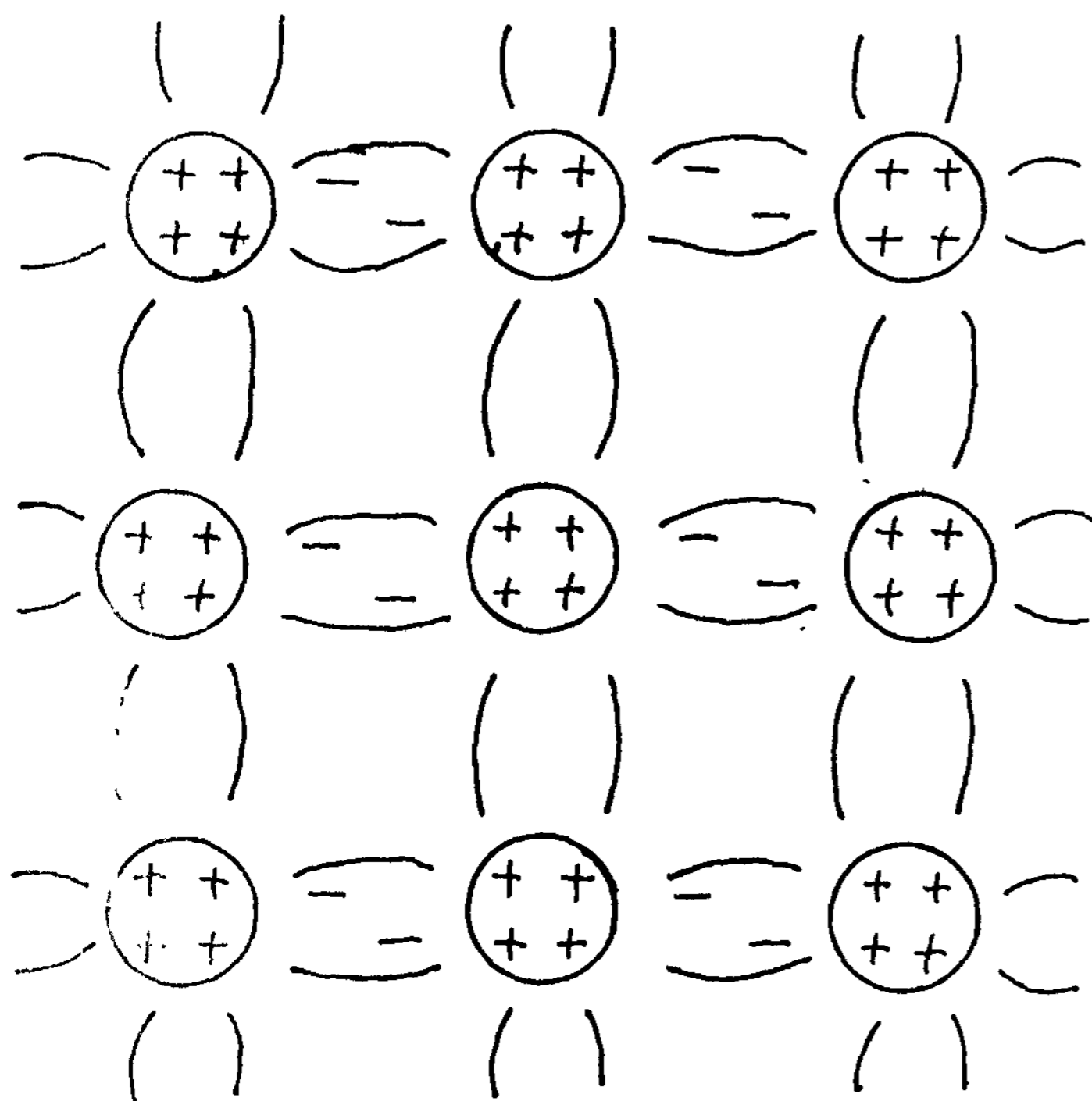


Versuch Nr.	Anschlußstellen des Transistors		Glühlampe leuchtet	Sperr-/durchlaß-richtung	Diode keine Diode
	an +	-			
1.1	B	E			
1.2	E	B			
2.1	B	C			
2.2	C	B			
3.1	C	E			
3.2	E	C			



o C Zeichne die
B o Dioden ein!
o E

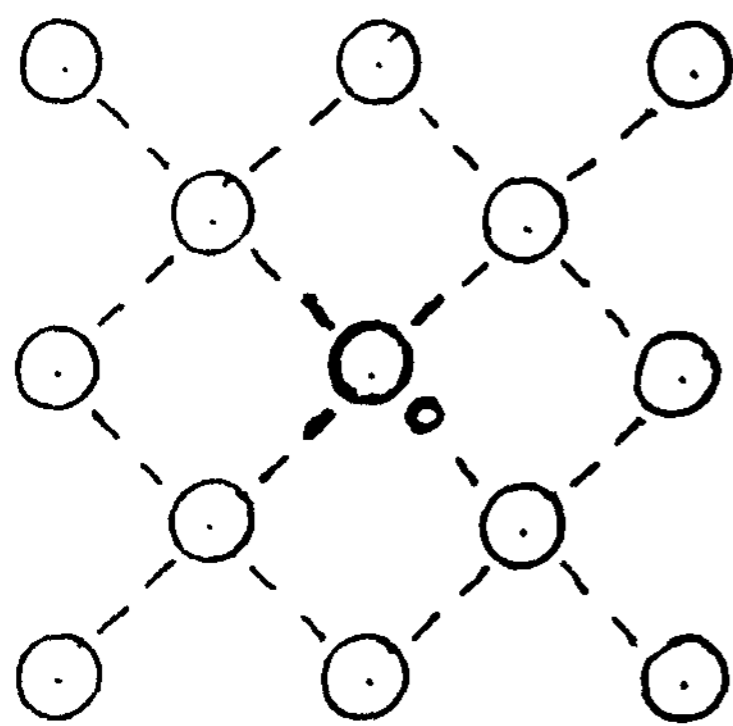
7. Aufbau eines Halbleiters



Hineinbringen von Fremdatomen (dotieren)
1 Fremdatom auf 10^6 Atome

Wird Antimon (fünfwertig) zugesetzt,
herrschen überschüssige Elektronen vor.

Es ist n-Material entstanden, N-leitend.

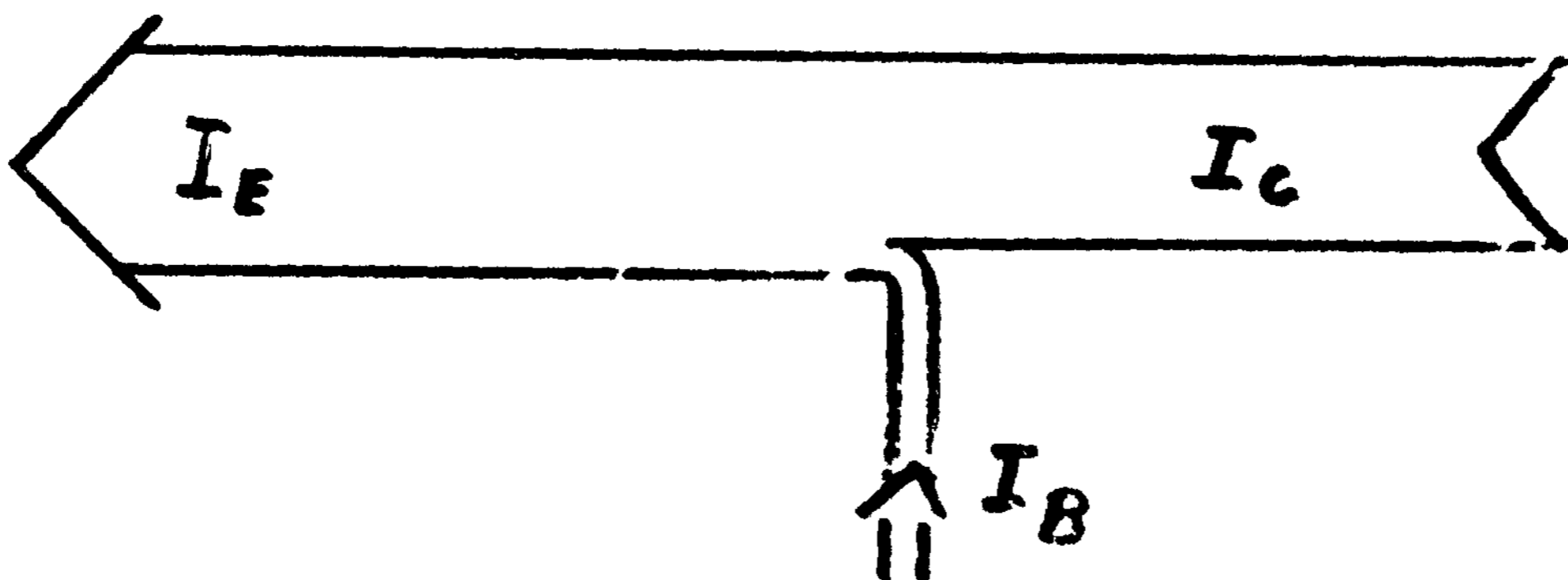
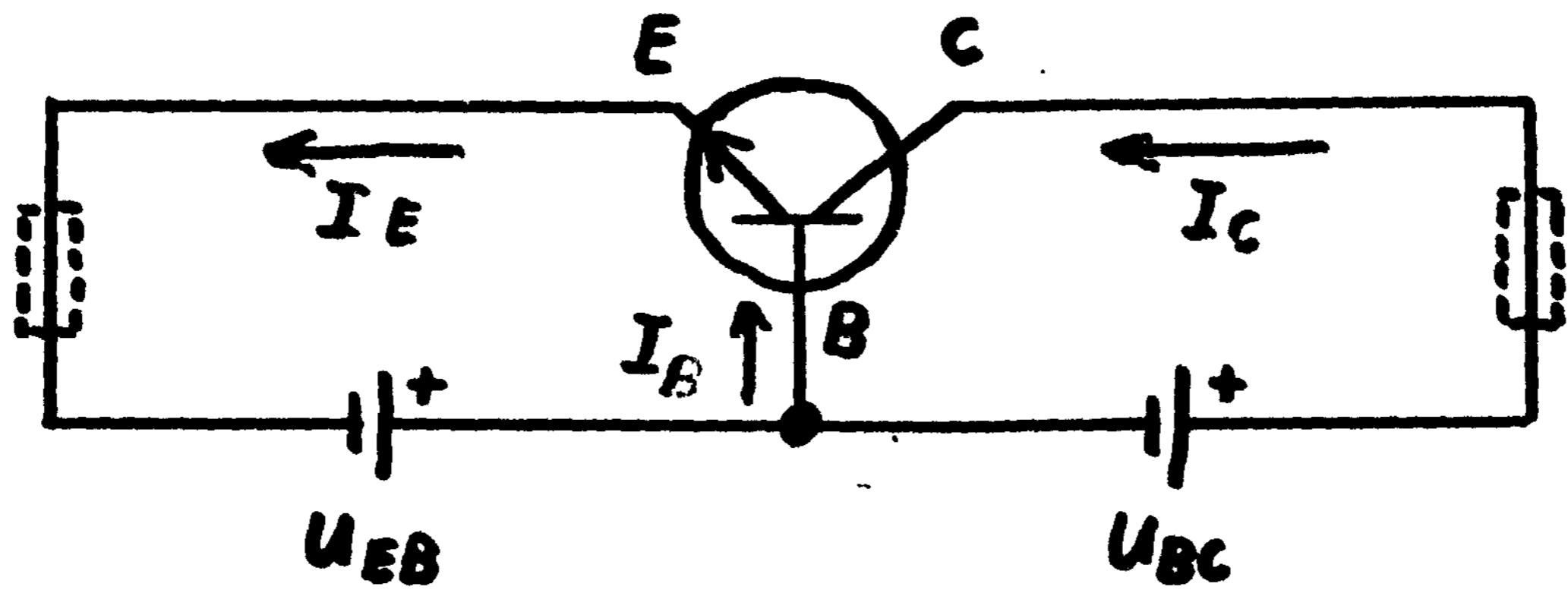
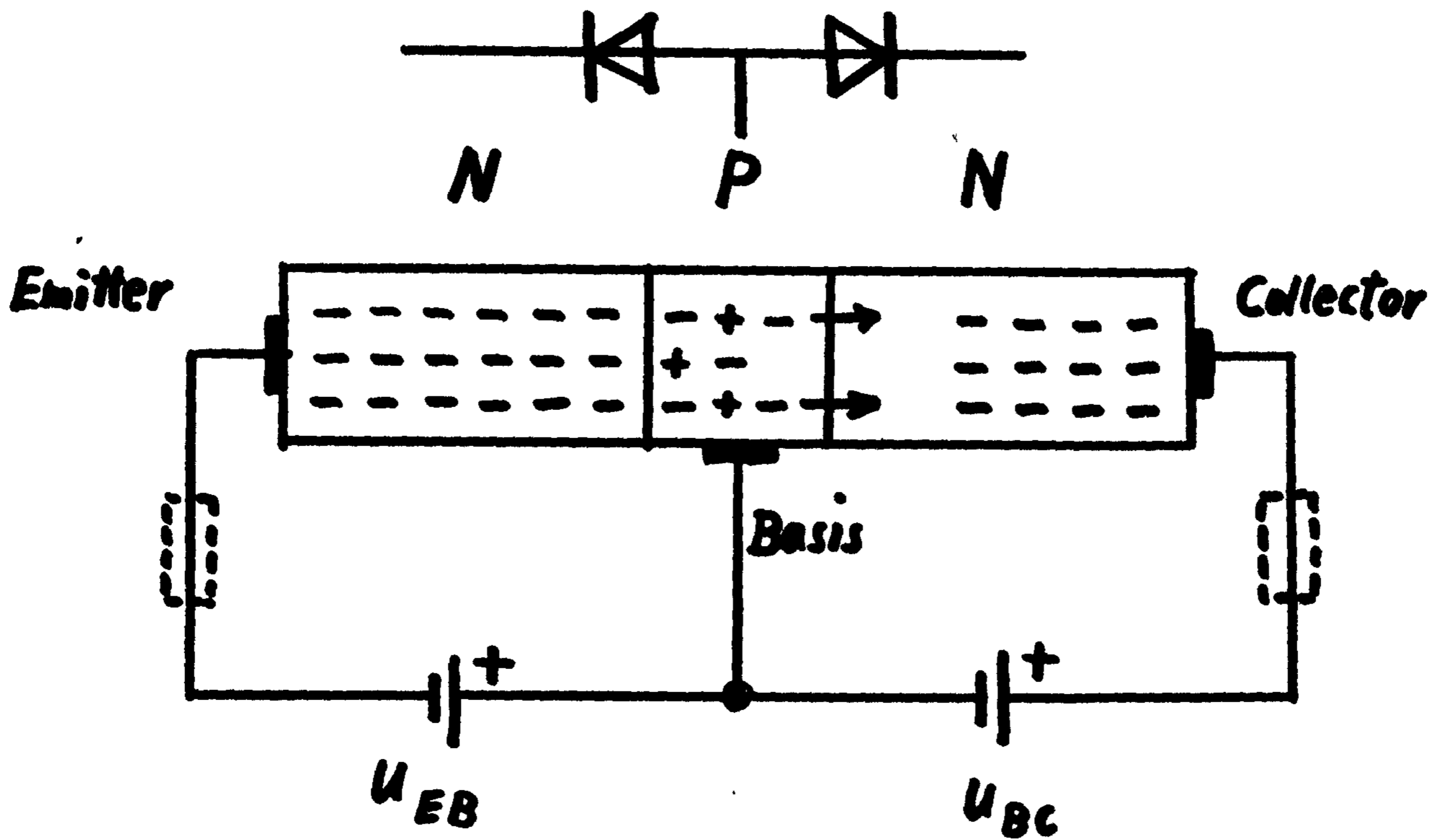


Wird Indium (dreiwertig) zugesetzt,
entsteht eine Bindungslücke, ein
Mangel-Elektron oder Loch.

Liegt ein äußeres Feld an, so wandert
das Loch entgegengesetzt zur
Bewegungsrichtung der Elektronen.

Es ist p-Material entstanden, P-leitend.

Der NPN - Transistor



Test zur Unterrichtseinheit "Wir bauen einen Computer"

Name:..... Klasse:..... Datum:.....

1. Die Technisierung schreitet unaufhörlich fort. Als Mitbeteiligter würde ich folgende Gesichtspunkte bei der Ausarbeitung neuer Techniken berücksichtigen: 6
2. Wo sind Computer eingesetzt und was soll mit ihnen verrichtet werden? Nenne mindestens 6 Beispiele! 6
3. Was bedeutet EDV und wie sieht das Prinzip aus? 3
4. Ist Dir die Funktion des "Abakus" bekannt? Wenn ja, wie ist die Rechentechnik? 3
5. Wann wurden die ersten Rechenautomaten konstruiert? 2
6. Der Computer kennt nur 0 und 1. Der Stellenwert der Dualzahlen kommt durch die Potenzen der Basiszahl 2 zustande. Es ist daher eine Umrechnung vom Dezimalsystem zum Dualsystem und umgekehrt notwendig.
 - a) Wandle folgende Dezimalzahlen in Dualzahlen um!
14 530 101 64 97 222 6
 - b) Wandle folgende Dualzahlen in Dezimalzahlen um!
1101 0011 1001 11111 10101 5
7. Beim Feinlöten in der Elektronik sind einige wichtige Punkte zu beachten. Zähle mindestens drei dieser Punkte auf. 3
8. Es sind bestimmte Symbole elektronischen Bauteilen zugeordnet. Zeichne die Symbole für: Batterie, Transistor (NPN, PNP), Trimmwiderstand und Diode. 5
9. Ermittle von drei Widerständen die Widerstandswerte und bestimme den Toleranzbereich mit der Farbcode-Tabelle! 6
10. Wodurch entstehen Elektronenüberschuß beziehungsweise Elektronenmangel im Halbleitermaterial? Ordne die Bezeichnung n-Material und p-Material zu! 5

Gesamtpunktzahl: 50

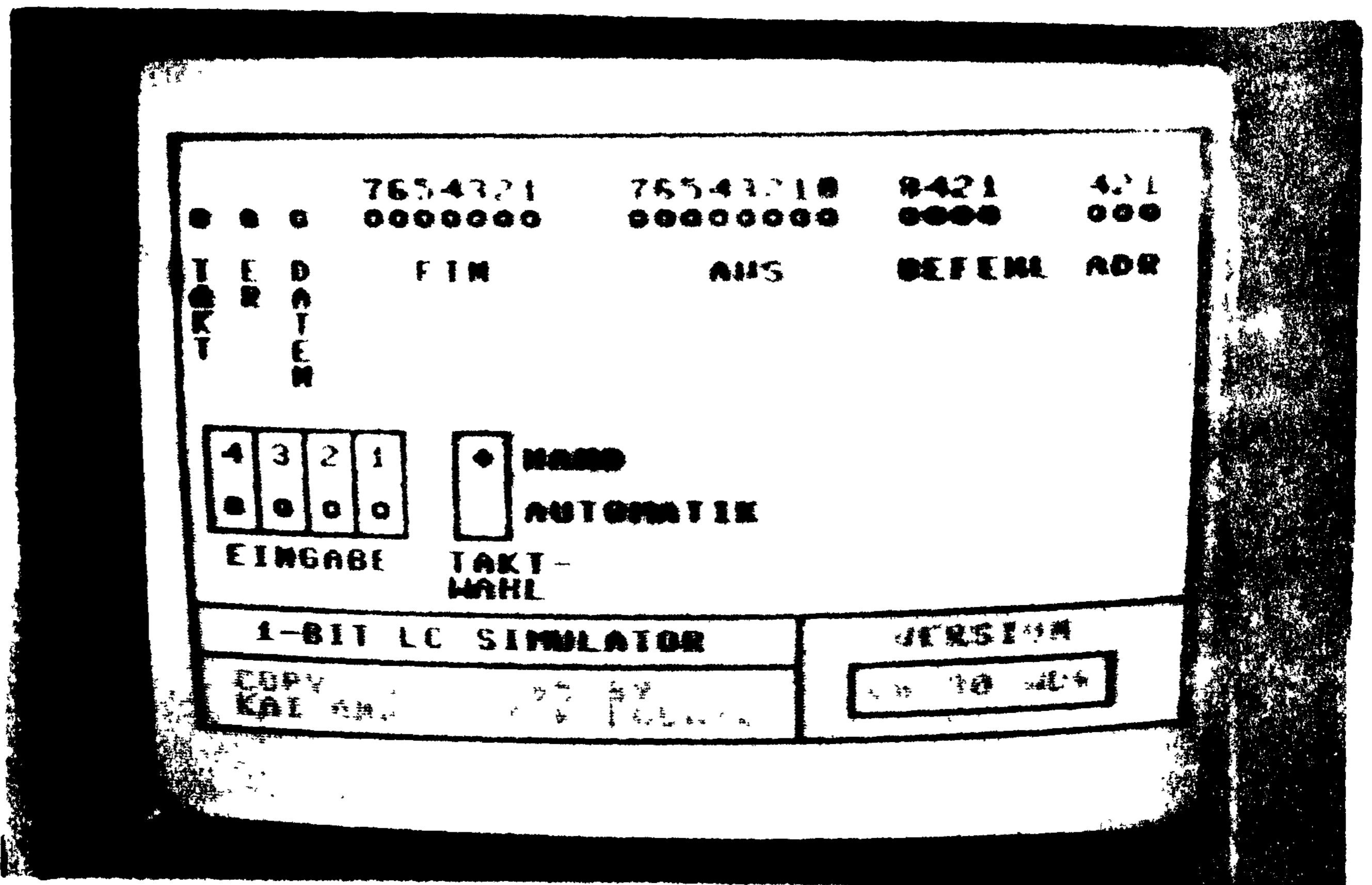
DIE SIMULATION DES WDR-1-BIT-COMPUTERS MIT DEM SCHULCOMPUTER

Der Einsatz eines Schulcomputers z.B. C 64, C128 oder Apple II zur Simulation des WDR-1-Bit-Computers soll keine preiswerte Ersatzlösung für dessen Aufbau sein. Es würde so die originale Begegnung mit einem Computer, die das Hauptziel der WDR-Sendereihe ist, verloren gehen.

Allerdings kann bei der Durchführung der Unterrichtseinheit "Wir bauen und programmieren einen Computer" eine organisatorische Schwierigkeit auftreten. Wenn z.B. aus finanziellen Gründen nur ein oder zwei WDR-Computer aufgebaut wurden, müssten während der Programmierphase 10 oder noch mehr Schüler an ein oder zwei WDR-Computern arbeiten.

Aus diesem Grunde sind Simulationen für verschiedene Rechner geschrieben worden, die mit Hilfe einer Floppystation eingelesen werden. Diese Programme sind rechnerresident, d.h. es ist der Erwerb nur einer Diskette nötig.

Kurzbeschreibung des Simulationsprogramms: Das Simulationsprogramm arbeitet vollkommen Menugesteuert. Die Programmeingabe, die Programmausführung in Hand- oder Schnelltakt sowie die Eingangsschaltersimulation gelingen mit Hilfe der Computertastatur. Der Zustand der 22 LEDs wird auf dem Bildschirm angezeigt. Natürlich ist das Simulationsprogramm, da es in BASIC geschrieben ist, nicht mehr so schnell wie der WDR-1-Bit-Computer. Deshalb ist es auch nicht für Steuerungszwecke z.B. eines Roboters geeignet. Hier wird wieder der WDR-Computer eingesetzt.



Schüler bauen sich Computer

Von unserem Redaktionsmitglied Bruno Seifert

Düsseldorf – „Ich kann's gar nicht abwarten bis zum nächsten Mal“, sagt ein 14-jähriger Krauskopf und unterdrückt nur mühsam seine Enttäuschung darüber, daß Lehrer Volker Ludwig die gerade geleisteten Unterrichtsfortschritte erst in der nächsten Woche begutachten will. Auch den anderen Schülern der 9. und 10. Klasse der Dusseldorfer Hauptschule an der Gotenstraße ist der nachmittägliche Computer-Unterricht viel zu schnell vorübergegangen. Seit Beginn des Schuljahres lernen sie alle mit Feuereifer, wie man sich selbst einen Computer baut. Motivationsprobleme sind ein Fremdwort.

Die freiwillige Schulbeschäftigung mit dem Rechner mag auf den ersten Blick nicht ungewöhnlich scheinen. Dennoch ist der Einzug der Prozessoren-Technik in die Dusseldorfer Hauptschule eine Besonderheit. Denn zum einen sehen die Richtlinien des Kultusministers für Hauptschüler keinen Computer-Unterricht vor. Selbst an den Gymnasien des Landes lassen sich die gelernten Informatiker an einer Hand abzählen, müssen Autodidakten in die Lücke springen. Und zum anderen fehlt es an Geld und Material.

Deutschlands größter Computer-Hersteller Nixdorf spendiert zwar für die höheren Schulzweige schon mal den einen oder anderen Rechner, die Hauptschulen bleiben in Sachen Computer allerdings Notstandsgebiet. Auch an Eigenmitteln gibt der Schuletat derzeit nichts her – im Gegensatz zu anderen Ländern wie etwa Japan, die ihre Schüler keineswegs ohne jeden Schimmer von EDV-Kenntnissen ins Berufsleben entlassen. Daß in Deutschland nun eine Dusseldorfer Hauptschule mit dem Eigenbau von Prozessoren für ein gutes Beispiel sorgt, ist in erster Linie ein Verdienst von Volker Ludwig.

Der Neusser Physikpädagoge, seit vier Jahren Autodidakt elektronischer Medien, arbeitete zwei Jahre an schulgeeigneten Rechnerprogrammen. Mit seinem Kollegen Burkhard John hat er im Urlaub einen Prototyp entwickelt, der ganz auf didaktisch

einfache Formen reduziert ist. Es handelt sich um einen 1-Bit-Prozessor, der modular in Funktionsgruppen aufgebaut ist. Er besteht aus fünf Platinen zur Aufnahme von Prozessor, Speicher, Anzeigen und Eingabetastatur. Durch seinen Bau lernen die 14- bis 15-jährigen in elementaren Grundschritten Aufbau und Funktionsweise eines Gerätes mit Maschinensprache. Sie wiederum ist Grundlage für die Programmiersprache „Basic“.

Die Kosten für die Computerelemente in Höhe von 150 Mark haben die Eltern aller 24 in zwei Kursen teilnehmenden Schüler übernommen. Volker Ludwig wertet dies auch als Bestätigung für seinen eingeschlagenen Kurs. Die im Vergleich zu Billigangeboten auf dem Fertigcomputermarkt relativ hohe Summe ist zudem ein Sonderpreis. Zustande kommt er durch eine Initiative des „Deutscher Amateur und Radio Club“ (DARC), dem auch Ludwig und John angehören. Bisher hat der DARC nach ihren Angaben etwa 150 präparierte und um je 35 Mark verbilligte Bausätze an Hauptschulen in der ganzen Bundesrepublik geschickt.

Daß die Amateurfunker dies nicht ganz uneigennützig machen, gibt Volker Ludwig zu. In einer jüngsten Analyse haben die DARC-Verantwortlichen nämlich herausgefunden, daß jugendlicher Clubnachwuchs ausbleibt: Computerspielen zieht offenbar besser als Morsen. Auf dem Umweg über das elektronische Lern- und Spielzeug verspricht die Mitgliederwerbung offenbar mehr.

Über den praktischen Wert ihres Computer-Unterrichts sind die Schüler verständlicherweise noch geteilter Meinung. Andre Westermann (14) denkt schon jetzt an eine spätere berufliche Laufbahn als Computertechniker, läßt sich dabei von seinem Vater und von Fachzeitschriften inspirieren. Für Rainer Jagielski dagegen ist Elektronik erst einmal Hobby, spätere Verwendung natürlich nicht ausgeschlossen. Vorerst reizt aber mehr der Spaß an der Freud': zum Beispiel an der computergesteuerten Lichtorgel im trauten Heim.

ZUR SENDEREIHE "BIT UND BYTE"

Zu den Inhalten der einzelnen Folgen des WDR-Schulfernsehfilms (aus (2))

Folge 1: Jede Menge Chips

Im ersten Teil werden Produktionsabschnitte beim Bau eines professionellen Computers gezeigt: Bohren und Ätzen von Platinen, Bestücken mit IC's, Löten und Überprüfen. Nach der Erklärung eines Chips als winziges Siliziumscheibchen mit tausenden von Schaltungselementen wird ein Mikroprozessor vorgestellt. Unter dem Hinweis auf negative und positive Auswirkungen der Chips werden Einsatzmöglichkeiten von Mikroprozessoren, die zur Automatisierung in der Industrie geführt haben, gezeigt: ein Bestückungsautomat für Fernsehplatinen, ein Schweißroboter und ein Herstellungsautomat für Fernsehgehäuse. Der Vergleich eines Fernsehers auf Röhrenbasis mit einem modernen auf der Grundlage der Verwendung von IC's hebt noch einmal die Bedeutung der Chips hervor. In einem anderen Anwendungsbereich werden ein Dialogterminal, ein Textautomat, ein Kontoauszugsdrucker, eine Codiermaschine, ein Bargeldautomat und ein BTX-Programm kommentiert. Im zweiten Teil werden das Team, der Computer und das für den Bau notwendige Werkzeug vorgestellt. Einfache Lötübungen beenden die Sendung.

Folge 2: Aller Anfang ist leicht: Die Anzeige

In dieser Folge geht es um Aufbau und Test der Anzeigeplatine. Vorgestellt werden zunächst die Baugruppen des Computers und die Bauteile der Anzeigeplatine. Die Anzeigeplatine wird gebohrt, die ersten Bauteile werden eingelötet. Nach einem Hinweis auf schlechte Lötstellen als Hauptfehlerquelle dieses Computerbaus werden die restlichen Bauteile eingelötet. Die Leuchtdioden und die Verstärker werden getestet und im Schaltbild gesucht. Ein Abschlußtest der Anzeigeplatine beendet die Folge.

Folge 3: Vergeißmeinnicht: Der Speicher

Aufbau und Test der Grund- und Speicherplatinen sind Inhalt der Folge 3. Nach der Vorstellung der Grundplatine und deren Bauteile wird diese aufgebaut. Die Stromversorgung und der Taktgenerator werden getestet und im Schaltbild zugeordnet. In einem zweiten Teil werden die Bauteile der Speicherplatine gezeigt und diese Platine dann zusammengesetzt. Nach dem Testen der Speicher und deren Zuordnung im Schaltbild werden die Zähler überprüft und im Schaltbild aufgesucht.

Folge 4: Einer für alle: Der Prozessor

Die Folge 4 ist dem Aufbau und dem Test der Prozessorplatine gewidmet. Nach der Vorstellung der neuen Bauteile wird diese aufgebaut und einem ersten Test unterzogen. Nach der Zuordnung der Inverter im Schaltbild wird der Hand-Takt überprüft. Mit dem Einsetzen der übrigen integrierten Schaltkreise kann diese Platine nur noch mit Hilfe von Programmen getestet werden.

Vor der Eingabe des ersten Programms wird das Zurücksetzen des Computers auf einen definierten Anfangszustand und die Aktivierung des Computers mit Hilfe von drei Befehlen (ORC, IEN, OEN) gezeigt.

Folge 5: 001, Das erste Programm

Um die Programmierung des Computers geht es in der Folge 5: Nach der Einführung zweier weiterer Befehle (LD, STO) wird die Handhabung einer Tastatur zur Erleichterung der Programmeingabe vorgestellt. Ein Lauflichtprogramm wird durch den Rücksprungbefehl (JUMP) abgeschlossen. Die Vorstellung des OR-Befehls steht exemplarisch für weitere logische Befehle.

Folge 6: Befehle nach draußen: Die Peripherie

Die letzte Folge steht im Zeichen der Peripherie zum Computer. Nach Vorübungen zur Steuerung von Motoren, die mit der Steuerung einer Radarantenne abschließt, werden Beispiele für weitere Peripheriesteuern genannt: ein Morsegenerator, eine Kojak-Sirene und ein Leuchtturm. Die Darstellung eines Bohrroboters zielt auf eine gehobene Programmierenebene hin. Weitere Roboter- und Maschinensteuerungsbeispiele sind eine Sortiersanlage, ein Teach-in-Roboter und ein Plotter. Die Vorstellung des Computerscheins soll das Interesse für das Erreichen eines solchen Scheines wecken.

» MICROCOMPUTER IN DER SCHULE «
WESTDEUTSCHER RUNDFUNK KÖLN/SCHULFERNSEHEN

Computer- schneid- klasse

hat den WDR-1-Bit-Computer
 gebaut und programmiert.

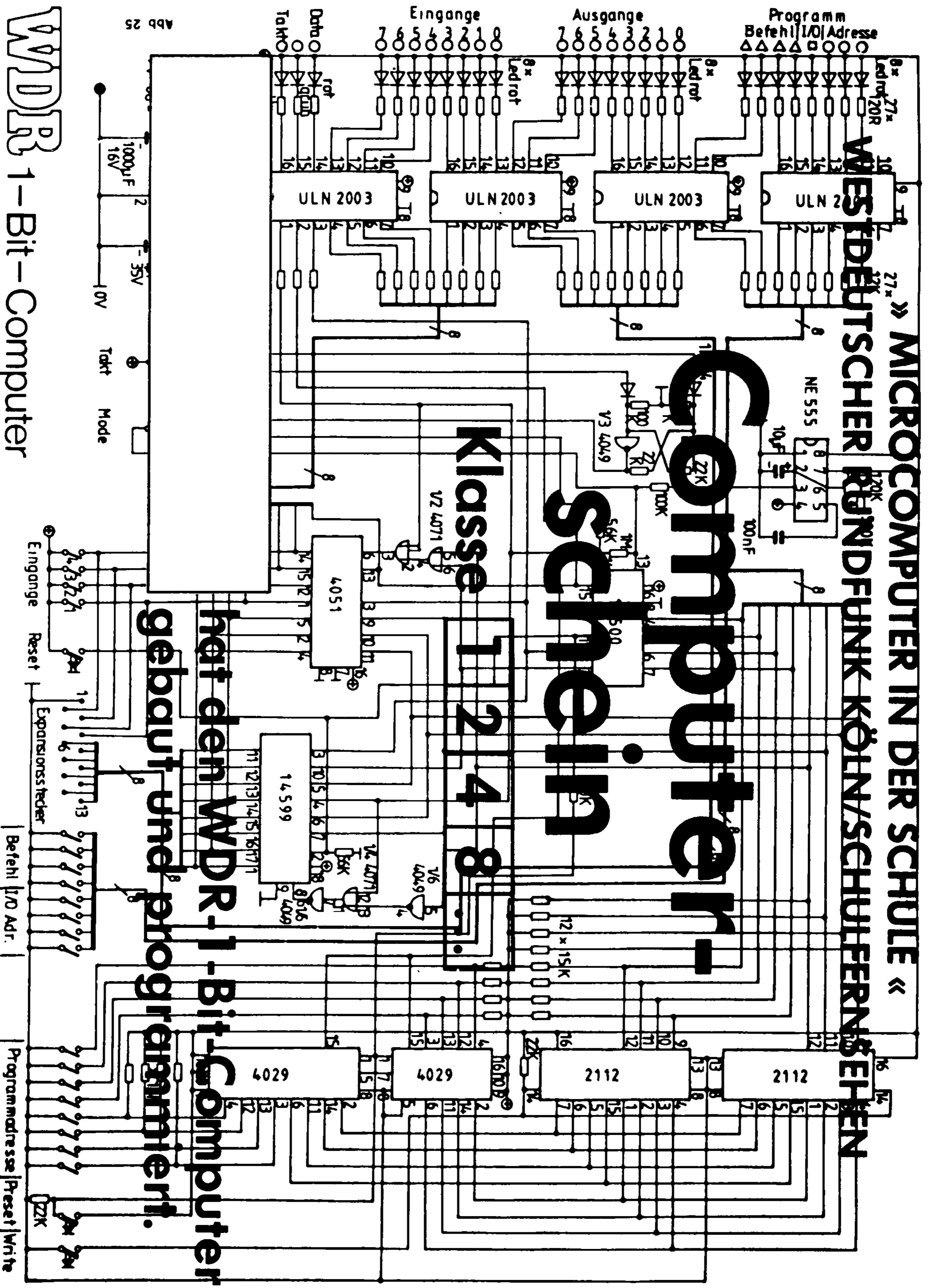


Abb 25

WDR 1-Bit-Computer

Eingänge | Reset | Expansionsstecker | Befehl | I/O Adr. | Programmadresse | Preset | Write

Erweiterungen zum WDR-1-Bit-Computer

Eingabetastatur für den WDR-1-Bit-Computer	150
Interface WDR-1-Bit-Computer/fischertechnik-computing	152
Interface C-64/fischertechnik-computing	159
Pufferplatine für 8 Ausgänge	163
Siebensegmentanzeige	164
Motorsteuerung	166
Schrittmotorsteuerung	168
RAM/EPROM-Platine	170
Morsezeichengenerator	173
Musikbox	174
C 64-Adapter	176
Relaisplatine	177
Ausgangserweiterung	178
Analog-Digitalwandler	182
Eingangserweiterung	184
Programmierbare Adressenumschalter für die RAM/EPROM-Platine	186
Steuerung von zwei Schrittmotoren	188

DATANorf 1: EINGABETASTATUR FÜR DEN WDR-1-Bit-COMPUTER

1. Beschreibung der Tastatur:

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 Programmbefehl besteht aus Zahlen im Zweiersystem, deren Stellen durch 0 V ("0") bzw. 5 V ("1") dargestellt werden. Zur Codierung ist es nun notwendig, die Stellen, die "0" werden sollen, mit 0 V zu verbinden, denn durch die 22 Kilo-Ohm-"Pull-up"-Widerstände auf der Speicherplatine sind alle Programmspeicher schon mit 5 V verbunden.

Die Dioden sind nötig, um die verschiedenen Tasten zu entkoppeln.

2. Der Schaltplan:

Dieser ist als Anlage beigelegt.

3. Bestückungsplan:

Dieser ist als Bestückungsaufdruck auf die Platine gedruckt.

4. Tastaturaufdruck:

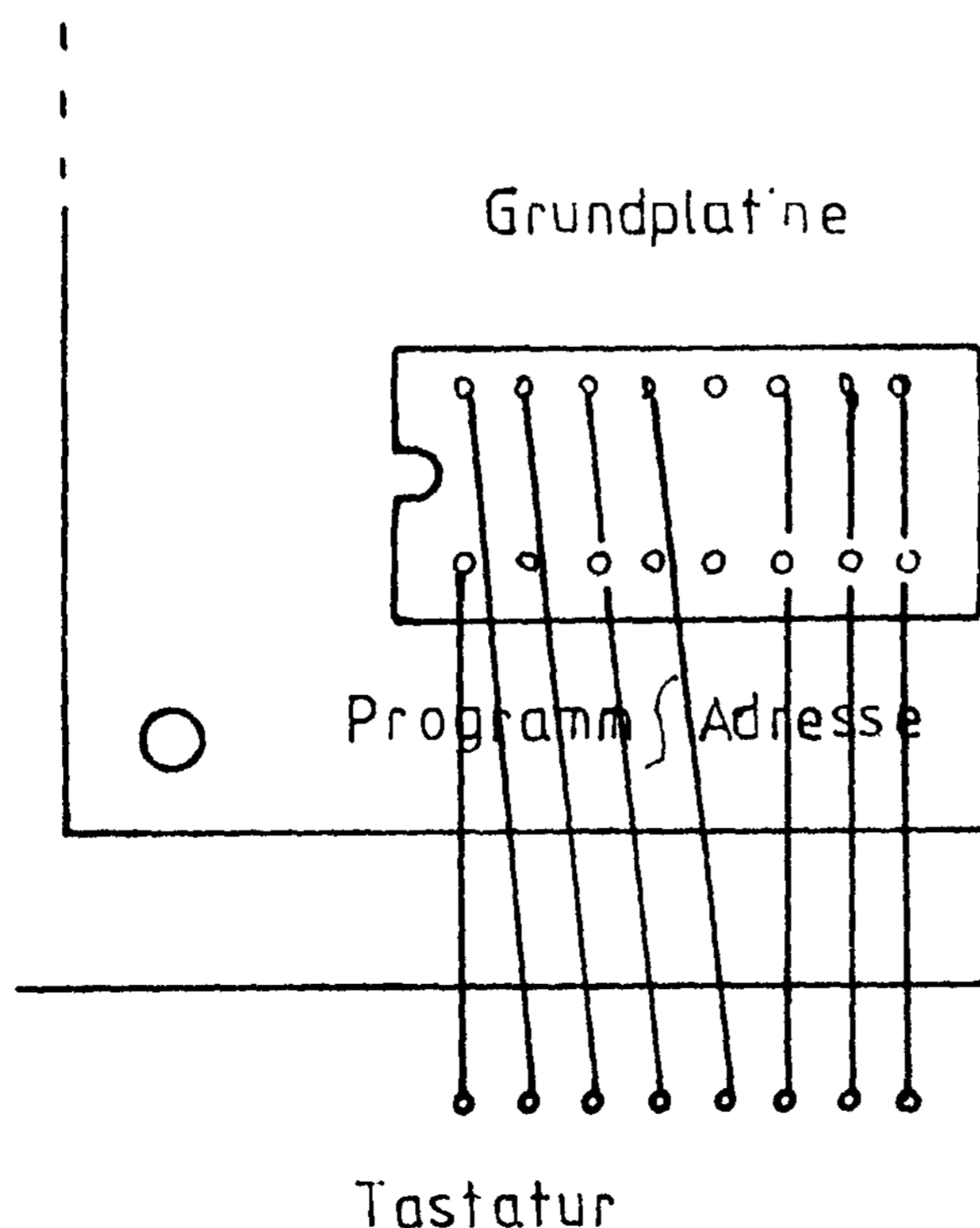
Vgl. Abb. 26 Praxis Schulfernsehen 111/1985 S. 99

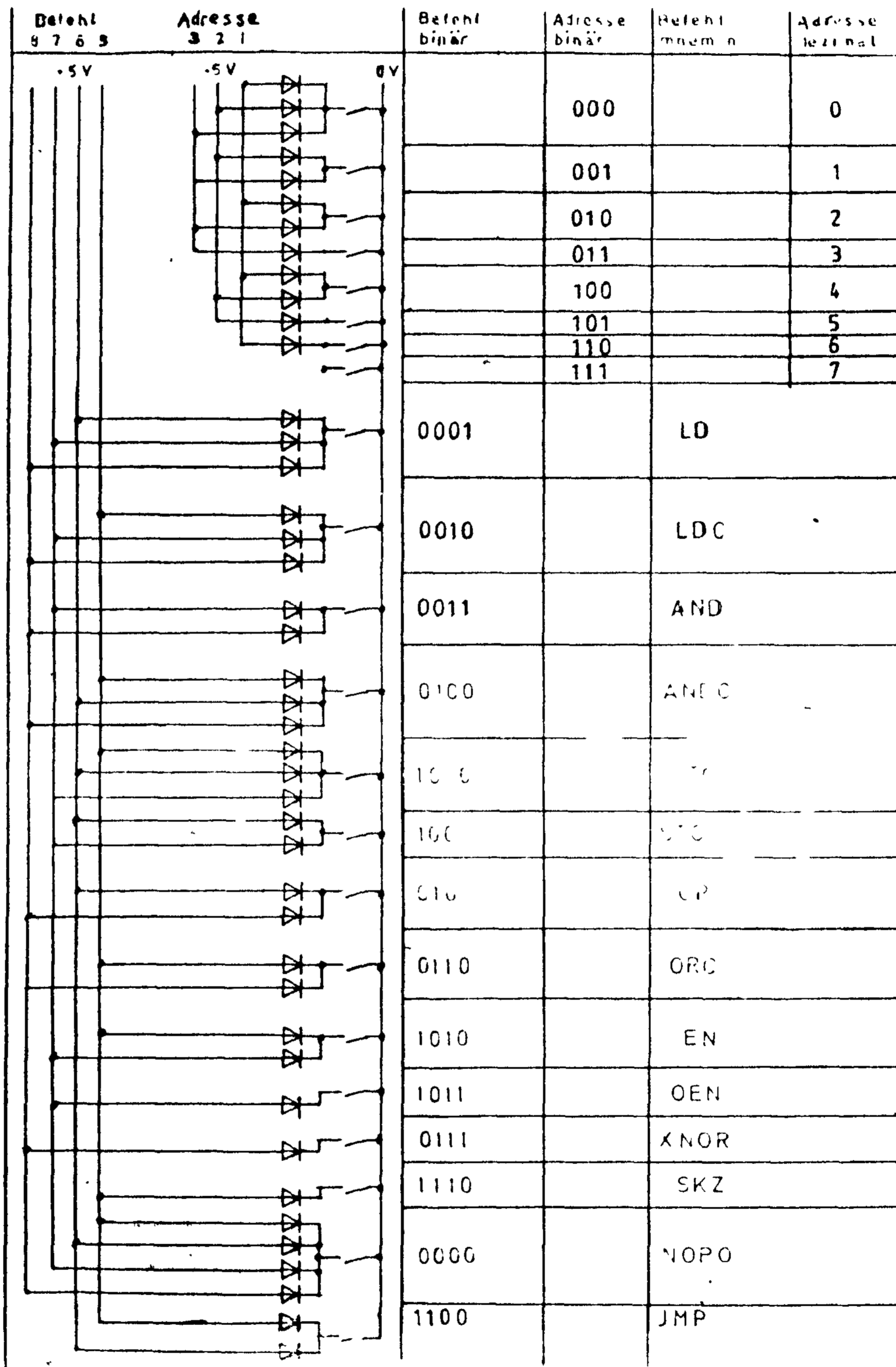
5. Didaktische Information:

Auf einen eleganten Prägeaufdruck für die 22 Tasterkappen wurde primär nicht aus Kostengründen verzichtet, obwohl sich das Produkt so, wenn auch entscheidend teurer, aus kosmetischen Gründen besser verkaufen würde.

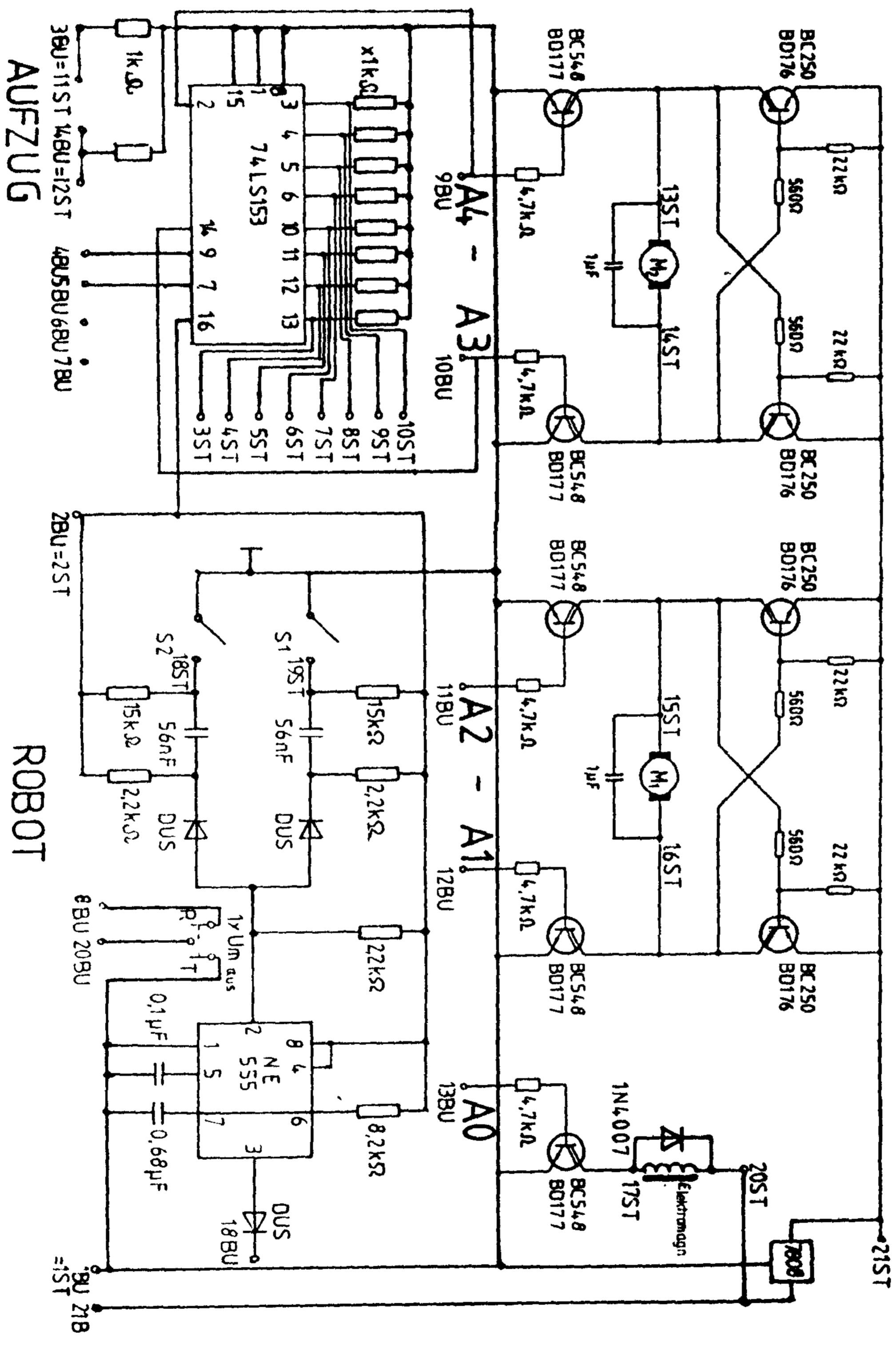
Im Vordergrund stand die Intention, der Lerngruppe die Möglichkeit zu geben, die Diodenmatrix eigenständig zu erforschen und so z.B. mit Hilfe von Tageslichtprojektorstiften, Aufklebern etc. den Tastern eine eigene Prägung zu geben.

Eine Prägeversion für die Tastkappen der Tastatur wird auf Anfrage deligiert. DATANorf 2, Version WDR-1-Bit-Computer





DATANORF2: Interface WDR-1-Bit-Computer/fischertechnik-computing



Fertigstellung und Inbetriebnahme der DATANorf 2-Interfaceplatine für den Betrieb von Fischertechnik-Computing-Modellen mit dem WDR-1-Bit-Computer

Für die Inbetriebnahme einiger Modelle werden 10 Fischertechnik-Winkelsteine, die im Fischertechnik-Set nicht enthalten sind, benötigt. Diese können unter DATANorf 20 bestellt werden. Ebenfalls können die 13- bzw. 21-poligen Buchsenleisten für den Anschluß der Modelle an das Interface bezogen werden.

1. Vorarbeiten

1.1 Die Lötaugen für die Buchsenleiste werden nicht gebohrt! Die Lage der Buchsenleiste ist aus dem Bestückungsaufdruck ersichtlich. Bohren der Platine DATANORF 2 mit 1mm-Bohrer. Bohren der Löcher für die zwei GummifüÙe mit 5 mm-Bohrer.

1.2 Steckerleiste einsetzen und beidseitig verlöten.

1.3.1 Buchsenleiste an die Unterseite anlöten.

1.3.2 Buchsenleiste an der Oberseite an den angegebenen Stellen, mit Hilfe von Drahtenden verlöten.

1.4 2 GummifüÙe einsetzen und gegebenenfalls festkleben.

2. Fertigstellen des Feldes A0 für den Betrieb des Elektromagneten

2.1 Bestücken des Feldes mit den Bauteilen: 1 Widerstand 4,7 kOhm (gelb, violett, rot, gold), Transistor BC 548 und BD 177, Diode 1N4007, Durchkontaktierung I im Feld ROBOT mit Hilfe eines Drahtendes. (Ermittlung des Emitters E des Transistors BD 177: Den Transistor mit der Beschriftungsseite nach oben legen; zeigen die drei Anschlüsse zum Betrachter, so ist der linke Anschluß der Emitter).

2.2 Einlöten der Bauteile.

2.3 Inbetriebnahme: Aufstecken der Interfaceplatine auf den Peripheriestecker des WDR-1-Bit-Computers. Anschluß des Elektromagneten an Stift 17 und Stift 20 der Steckerleiste. Das Programm zur Aktivierung des Elektromagneten lautet: INIT, STO 0. Mit STOC 0 wird der Elektromagnet wieder abgeschaltet.

3. Fertigstellen des Feldes A1-A2 für den Betrieb des ersten Motors

3.1 Bestücken des Feldes mit den Bauteilen: 2 Widerstände 560 Ohm (grün, blau, braun, gold), 2 Widerstände 4,7 kOhm, 2 Widerstände 22 kOhm (rot, rot, orange, gold), 1 Kondensator 0,82 Mikrofarad, den Transistoren 2 BC 250, 2 BC 548, 2 BD 176, 2 BD 177.

3.2 Bestücken des Feldes ROBOT mit dem Spannungskonstanter 7808. Er wird so eingesetzt, daß die Metallplatte zum Schalteraufdruck zeigt.

3.3 Setzen der Durchkontaktierungen II in Feld A3-A4, III, IV, V, VIa, VIIa. (Die Durchkontaktierungen VIa und VIIa bewirken die Spannung des Konstanters, die von VIb und VIIb die der Eingangsspannung des WDR-1-Bit-Computers an dem Motor).

3.4 Einlöten der Bauteile (zum Teil auf beiden Seiten der Platine).

3.5 Inbetriebnahme: Anschluß des Motors 1 an Stift 15 und Stift 16 der Steckerleiste. Das Interface aufstecken. Das Programm INIT, STO 1 läÙt den Motor in eine Richtung drehen. Mit anschließendem STOC 1 wird der Motor abgeschaltet. Daraufhin

bewirkt STO 2 das Drehen des Motors 1 in die andere Richtung. Mit STOC 2 wird der Motor 1 wieder abgeschaltet.

4. Fertigstellen des Feldes A3-A4 für den Betrieb des zweiten Motors

4.1 Bestücken des Feldes wie unter 3.1.

4.2 Setzen der Durchkontaktierungen VIII und IX.

4.3 Einlöten der Bauteile.

4.4 Inbetriebnahme: Anschluß des Motors 2 an Stift 13 und 14 der Steckerleiste. Das Programm STO 3 läßt den Motor 2 in eine Richtung drehen. Mit anschließendem STOC 3 wird der Motor 2 abgeschaltet. Daraufhin bewirkt STO 4 das Drehen in die andere Richtung.

Mit STOC 4 wird der Motor wiederum abgeschaltet.

5. Fertigstellen des Feldes ROBOT für den Betrieb der Fischertechnik-Computing Versionen "Teach-in Roboter" und "Plotter".

5.1 Bestücken des Feldes mit den Bauteilen: 2 Widerstände 15 k Ω (braun, grün, orange, gold), 3 Widerstände 2,2 k Ω (rot, rot, rot, gold), 1 Widerstand 8,2 k Ω (grau, rot, rot, gold) 2 Kondensatoren 56 nF, 1 Kondensator 0,68 μ F, 1 Kondensator 0,1 μ F, 2 Universalsiliziumdioden (DUS), 1 Sockel für den Timer NE 555 mit Fassung (Die Kerbe des Bestückungsaufdrucks muß mit der Kerbe der Fassung übereinstimmen, damit das IC entsprechend seiner Markierung richtig eingesetzt werden kann).

5.2 Setzen der Durchkontaktierungen X, XI, XII.

5.3 Setzen der Durchkontaktierung XIII im Feld A3-A4.

5.4 Einlöten der Bauteile und Einsetzen des NE 555.

5.5 Inbetriebnahme: Anschluß der beiden Fischertechniktaster an Stift 18 und Stift 19 der Steckerleiste, gemeinsame Masse an Stift 1. Aufstecken der Interfaceplatine auf den Peripheriestecker des WDR-1-Bit-Computers.

5.5.1 Testen der alten Betriebsart: Interfaceschalter auf Mittelstellung bringen. Eingabe des Testprogramms: INIT, STO 5, STO 0, 10 mal NOP 0, STOC 0, NOP 0, JMP x. In Stellung "LT" läuft das Programm langsam ab.

5.5.2 Testen der Funktion des monostabilen Multivibrators, der einen positiven Impuls von ca. 2 ms Dauer bei jeder Tasterbetätigung erzeugt: Interfaceschalter auf "T" (Test) schalten. Das Programm läuft nun wahlweise durch beide Fischertechniktaster getaktet ab. Wenn bei dem vorgegebenen Programm kein Programmschritt übersprungen wird, arbeiten die Fischertechniktaster nun prellfrei.

5.5.3 Testen der digitalen Abfrage durch den Roboter: Interfaceschalter nach "R" (Roboterbetrieb) schalten. Nun muß ein einmaliges Drücken des entsprechenden Fischertechnik-Tasters den Programmzähler um 1 erhöhen. Nach der Ausführung des Befehls STO 5 läuft das Programm wieder durch den Computer getaktet ab, bis es nach Erreichen des Befehls STOC 5 wieder durch den Taster des Roboters getaktet werden muß.

6. Fertigstellen des Feldes Aufzug

6.1 Bestücken des Feldes mit dem Sockel für den SN 74 LS 153 (die Kerbe des IC-Sockels auf die Kerbe des Sockelaufdrucks bringen), den 9 Widerständen 1 k Ω (braun, schwarz, rot, gold).

6.2 Setzen der Durchkontaktierungen XIV und XV (Feld A3-A4), XVII (Feld Roboter), XVI-XXII, XXIIIa und XXIVa, XXV-XXVII, XXVIII und XXX (Feld A3-A4).

6.3 Einlöten der Bauteile teilweise beidseitig.

6.4 Entfernen der Durchkontaktierung III. Damit wird verhindert, daß bei der Aktivierung der Ausgänge A3 und A4 mit "1" die Transistoren des Feldes A3-A4 in einen unzulässigen Betrieb geraten.

6.5 Inbetriebnahme: Aufstecken des Interfaces auf den Peripheriestecker des WDR-1-Bit-Computers. Anschluß der Aufzugtaster S0, S1, S2, S3, S4, S5, S6 und S7 an die Stifte 3 bis 10 des Interfaces und gemeinsam an Stift 2. Eingeben des Testprogramms für die Abfrage der Taster: INIT, ST03, ST0C 3, ST0 4 und ST0 3. An den Programmstart gehen. Taster S3 bewirkt eine 1 an E1, S4 an E2 des Computers. Das Programm bis zur Ausführung von ST0 3 taktet. Nun wirken S2 bzw. S5 an E1 bzw. E2. Das Programm bis zum Ende taktet. Nun wirken S0 bzw. S7 an E1 bzw. E2.

7. Anschluß des Bohrroboters

Vorarbeiten: Das IC 74 LS 153 entfernen. Die Durchkontaktierungen XXIII b und XXIV b setzen.

Die Motoren, Lampen und Taster werden wie folgt an eine 21-polige Buchsenleiste (nicht im Lieferumfang enthalten) angeschlossen:

Pin 2 : oberer Taster, 3 (Dieser Anschluß ist mit + 5 Volt des WDR-1-Bit-Computers verbunden) ; unterer Taster, 3; Vorlagetischtaster, 3.

Pin 9 : Vorlagetischtaster, 1. Dieser Anschluß wird an Eingang E1 des WDR-Computers herangeführt, wenn das IC 74 LS 153 entfernt wurde und die Durchkontaktierung XXIII b gesetzt wurde.

Pin 10: unterer Taster, 1. Dieser Anschluß wird an Eingang E2 des WDR-Computers herangeführt, wenn die Durchkontaktierung XXIV b gesetzt wurde.

Pin 11: oberer Taster, 1. Dieser Anschluß wird an E3 des WDR-Computers herangeführt.

Pin 15: Bohrmotor, Lampe grün

Pin 16: Bohrmotor, Lampe grün

Pin 17: Vorlagetischmotor, Lampe rot

Pin 21: Vorlagetischmotor, Lampe rot

8. Anschluß des Aufzugs

Vorarbeiten: Die Durchkontaktierungen XXIII b sowie XXIV b entfernen und die Durchkontaktierungen XXIII a und XXIVa setzen. Je einen Anschluß der beiden 4,7 kOhm-Widerstände ablöten.

Der Motor und die 6 Taster werden wie folgt an eine 21-polige Buchsenleiste (nicht im Lieferumfang enthalten) angeschlossen:

Pin 2 : mit den Anschlüssen 3 aller 6 Taster

Pin 4 : mit dem Anschluß 1 des Wahltasters für die 2. Etage

Pin 5 : mit dem Anschluß 1 des Wahltasters für die 1. Etage

Pin 6 : mit dem Anschluß 1 des Wahltasters für das Erdgeschoß

ST03
ST0C3
ST04
ST03

Wahltaster
Drehisole S1 an E1
auf S6 an E2

Pin 7 : mit dem Anschluß 1 des Abfragetasters für das Erdgeschoß
 Pin 8 : mit dem Anschluß 1 des Abfragetasters für die 1. Etage
 Pin 9 : mit dem Anschluß 1 des Abfragetasters für die 2. Etage
 Pin 15 : mit dem Motor, Lampe
 Pin 16 : mit dem Motor, Lampe

9. Anschluß der Sortieranlage

Vorarbeiten: Wie bei 7.

Definition der Taster:

1 Taster (links oben; startet den Sortiervorgang): S ob
 3 Taster (mitte: links für die linke Begrenzung des Schlittens; mitte für das Anhalten des Schlittens; rechts für die rechte Begrenzung des Schlittens) S mi li, S mi mi, S mi re
 2 Taster (unten: links, rechts für die Längenabfrage) S un li, S un re

Der Motor, die 6 Taster, die Lampen und 2 Dioden werden wie folgt an eine 21-polige Buchsenleiste (nicht im Lieferumfang enthalten) angeschlossen.

Pin 2 : mit den Anschlüssen 3 S mi li, 3 S ob, 3 S mi mi, 3 S mi re, 1 S un re, 1 s un li.
 Pin 9 : mit dem Anschluß 1 S mi mi
 Pin 10: mit dem Anschluß 1 S mi li
 Pin 11: mit dem Anschluß 2 S un re
 Pin 12: mit dem Anschluß 1 S mi re und 1 S ob
 Pin 15: mit dem Motor und den Lampen grün, rot sowie gelb
 Pin 16: mit dem Motor, Lampe grün und den Anoden der beiden Dioden. Die eine Kathode (der Anschluß mit dem Markierungsring) geht an die rote Lampe, die andere Kathode geht an die gelbe Lampe.

10. Anschluß des Plotters

Vorarbeiten:

Die originale Potentiometerabfrage wird für die WDR-1-Bit-Computerversion nicht benutzt.

Anbringen der digitalen Abfragen:

1. Auf die Laufschiene für den Schreibstifttransport werden 11 Winkelsteine (vgl. Datanorf 20) auf der Seite, die dem Elektromagneten gegenüberliegt, geschoben und gegen Verrutschen gesichert.

2. An einen grauen Baustein 15*15*15 mit Zapfen über dem Getriebe des Plotterschlittens wird ein roter Baustein 15*15*7,5 angebracht. Auf der gegenüberliegenden Seite dieses roten Bausteins wird ein grauer Baustein 30*15*15 mit Hilfe seines Zapfens befestigt. Auf dessen Unterseite wird ein weiterer grauer Baustein 30*15*15 mit Hilfe seines Zapfens geschoben. Am anderen Ende dieses Bausteins wird ein Taster so befestigt, daß er bei Bewegung des Schlittens ~~die~~ ^{er} 11 Winkelstücke betätigen kann. *werden*

3. Auf die verlängerte Welle für die Schnecke zum Treiben des schwarzen Zahnrades wird eine rote Flachnabe 25 mm Durchmesser

so befestigt, daß ein darunter angebrachter Taster bei einer Drehung der Schneckenachse um 360 Grad zweimal betätigt wird.

Die Motoren, Taster und Lampen und der Elektromagnet werden wie folgt an eine 21-polige Buchsenleiste (nicht im Lieferumfang enthalten) angeschlossen:

Pin 2 : Anschlüsse 3 Taster für die digitalen Abfragen
 Pin 13: Motor auf der Laufschiene, rote Lampe
 Pin 14: Motor auf der Laufschiene, rote Lampe
 Pin 15: Motor für den Drehvorgang, grüne Lampe
 Pin 16: Motor für den Drehvorgang, grüne Lampe
 Pin 17: Elektromagnet, *gelbe Lampe*
 Pin 18: Anschluß 1 Taster für die Winkelsteinabfrage
 Pin 19: Anschluß 1 Taster für die Abfrage mit der roten Flachnabe
 Pin 20: Elektromagnet, *gelbe Lampe*

11. Anschluß der Ampelanlage

Die Ampelanlage wird mit Hilfe der Pufferplatine DATANorf 3 betrieben, die an den Expansionsstecker des WDR-1-Bit-Computers Pin 1-13 gesteckt wird. Weiterhin wird ein 1 kOhm-Widerstand benötigt.

Die 6 Lampen der Ampelanlage werden wie folgt an eine 13-polige Buchsenleiste (nicht im Lieferumfang enthalten) beschaltet:

Pin 1 : Anschluß eines 1 kOhm-Widerstandes
 Pin 2 : Ampel mit Taster : Lampe grün, gelb und rot
 Pin 5 : Anschluß des anderen Endes des 1 kOhm-Widerstandes
 Pin 8 : Ampel mit Taster : rote Lampe
 Pin 9 : Ampel mit Taster : gelbe Lampe
 Pin 10: Ampel mit Taster : grüne Lampe
 Pin 11: Ampel ohne Taster: rote Lampe
 Pin 12: Ampel ohne Taster: gelbe Lampe
 Pin 13: Ampel ohne Taster: grüne Lampe

12. Anschluß des Teach-in-Roboters

Die Potentiometerabfrage wird nicht benötigt. Für die digitale Abfrage der Höhenposition des Elektromagneten werden 10 Winkelsteine benötigt, die unter die Schiene für die Höhenverstellung des Elektromagneten befestigt werden (vgl. Schulcomputer 2-5/ 85 Abb. 18).

Für die digitale Abfrage der Seitenposition des Elektromagneten wird auf die Achse mit dem roten Schneckenrad eine rote Seiltrommel so befestigt, daß bei einer Umdrehung ein darunterliegender Taster 2 mal betätigt wird (vgl. Schulcomputer 2-5/ 85 Abb. 18).

Die Motoren, Taster, Lampen und der Elektromagnet werden wie folgt an eine 21-polige Buchsenleiste (nicht im Lieferumfang enthalten) angeschlossen:

Pin 1 : Anschluß 3 beider Taster für die digitalen Abfragen
 Pin 13: Lampe rot, Motor für den Höhentransport
 Pin 14: Lampe rot, Motor für den Höhentransport
 Pin 15: Lampe grün, Motor für den Seitentransport

DATANorf 2: VERSION C 64

Fertigstellung und Inbetriebnahme der DATANorf 2 - Interfaceplatine für den Betrieb von Fischertechnik-computing-Modellen und einem Aufzug aus Fischertechnik-Bausteinen mit dem C 64

(Für die Inbetriebnahme wird der DATANorf 10-Adapter und ein Spannungsversorgungsgerät ca. 12 V, ca. 300-400 mA mit einem Klinkenstecker 2,5 mm benötigt; für das Betreiben der Roboter werden 10 Fischertechnik-Winkelsteine, die im Fischertechnik-Set nicht enthalten sind, benötigt. Diese können unter DATANorf 20 bestellt werden.)

1. Vorarbeiten

1.1 Die Lötaugen für die Buchsenleiste werden nicht gebohrt! Die Lage der Buchsenleiste ist aus dem Bestückungsaufdruck ersichtlich. Bohren der Platine DATANORF 2 mit 1mm-Bohrer. Bohren der Löcher für die zwei GummifüÙe mit 5 mm-Bohrer.

1.2 Steckerleiste einsetzen und beidseitig verlöten.

1.3.1 Buchsenleiste an die Unterseite anlöten.

1.3.2 Buchsenleiste an der Oberseite an den angegebenen Stellen mit Hilfe von Drahtenden verlöten.

1.4 2 GummifüÙe einsetzen und gegebenenfalls festkleben.

2. Fertigstellen des Feldes A0 für den Betrieb des Elektromagneten

2.1 Bestücken des Feldes mit den Bauteilen: 1 Widerstand 4,7 kOhm (gelb, violett, rot, gold), Transistor BC 548 und BD 177, Diode 1N4007, Durchkontaktierung I im Feld ROBOT mit Hilfe eines Drahtendes. (Ermittlung des Emitters E des Transistors BD 177: Den Transistor mit der Beschriftungsseite nach oben legen; zeigen die drei Anschlüsse zum Betrachter, so ist der linke Anschluß der Emitter).

2.2 Einlöten der Bauteile.

2.3 Inbetriebnahme: Aufstecken der Interfaceplatine DATANorf 2 auf den Adapter DATANorf 10 bei ausgeschaltetem !!! C 64, der seinerseits auf den Userport des C 64 gesteckt wurde.

Wichtig: Vor Anschluß der Stromversorgung ca. 9 V an den Adapter DATANorf 10 mit Hilfe eines Klinkensteckers 2,5 mm (die Spitze des Klinkensteckers zeigt positive Spannung) muß folgendes Abschaltprogramm aller Ausgänge auf 0 eingegeben werden:

**** Abschaltprogramm****

10 POKE 56579,255

20 POKE 56577,0

Jetzt kann die Stromversorgung hergestellt werden. Anschluß des Elektromagneten an Stift 17 und Stift 20 der Steckerleiste. Das Programm zur Aktivierung des Elektromagneten in Zeile 20 ändern: POKE 56577,1. Mit 20 POKE 56577,0 wird der Elektromagnet wieder ausgeschaltet.

3. Fertigstellen des Feldes A1-A2 für den Betrieb des ersten Motors

3.1 Bestücken des Feldes mit den Bauteilen: 2 Widerstände 560

Ohm (grün, blau, braun, gold), 2 Widerstände 4,7 kOhm, 2 Widerstände 22 kOhm (rot, rot, orange, gold), 1 Kondensator 0,82 Mikrofarad, den Transistoren 2 BC 250, 2 BC 548, 2 BD 176, 2 BD 177.

3.2 Bestücken des Feldes ROBOT mit dem Spannungskonstanter 7808. Er wird so eingesetzt, daß die Metallplatte zum Schalteraufdruck zeigt.

3.3 Setzen der Durchkontaktierungen II in Feld A3- A4,III (statt dieser Durchkontaktierung kann die Platinenunterseite über den im Lieferumfang enthaltenen Schalter, der jedoch für die C 64-Version nicht benutzt wird, mit der Platinenoberseite an Punkt III verbunden werden- dann entfällt die ROBOT-Version für den WDR-1-Bit-Computer),IV,V,VIa,VIIa.(Die Durchkontaktierungen VIa und VIIa bewirken die Spannung des Konstanters, die von VIb und VIIb die der Eingangsspannung des WDR-1-Bit-Computers an dem Motor).

3.4 Einlöten der Bauteile (zum Teil auf beiden Seiten der Platine).

3.5 Inbetriebnahme unter Verwendung des DATANorf 10-Adapters: Abschaltprogramm eingeben.Stromversorgung herstellen. Anschluß des Motors 1 an Stift 15 und Stift 16 der Steckerleiste. Das Programm in Zeile 20 ändern : 20 POKE 56577,2 .Es läßt den Motor in eine Richtung drehen. Mit anschließendem 20 POKE 56577,0 wird der Motor abgeschaltet. Daraufhin bewirkt 20 POKE 56577,4 das Drehen des Motors 1 in die andere Richtung.

4. Fertigstellen des Feldes A3-A4 für den Betrieb des zweiten Motors

4.1 Bestücken des Feldes wie unter 3.1.

4.2 Setzen der Durchkontaktierungen VIII und IX.

4.3 Einlöten der Bauteile.

4.4 Inbetriebnahme unter Verwendung des DATANorf 10 - Adapters:Abschaltprogramm eingeben.Herstellen der Stromversorgung. Anschluß des Motors 2 an Stift 13 und 14 der Steckerleiste. Das Programm in Zeile 20 POKE 56577,8 läßt den Motor 2 in eine Richtung drehen. Mit anschließendem 20 POKE 56577,0 wird der Motor 2 abgeschaltet. Daraufhin bewirkt 20 POKE 56577,16 das Drehen in die andere Richtung.

5. Fertigstellen des Feldes ROBOT für den Betrieb der Fischertechnik-Computing Versionen "Teach-in Roboter" und "Plotter".

5.1 Bestücken des Feldes mit den Bauteilen: 2 Widerstände 15 kOhm (braun,grün,orange,gold), 3 Widerstände 2,2 kOhm (rot,rot,rot,gold), 1 Widerstand 8,2 kOhm (grau,rot,rot,gold) 2 Kondensatoren 56 nF, 1 Kondensator 0,68 uF, 1 Kondensator 0,1 uF, 2 Universalsiliziumdioden, 1 Sockel für den Timer NE 555 mit Fassung (Die Kerbe des Bestückungsaufdrucks muß mit der Kerbe der Fassung übereinstimmen, damit das IC entsprechend seiner Markierung richtig eingesetzt werden kann). Nach der Bestückung bleiben neben der Durchkontaktierung XII der Schalter und über diesem die Universaldiode DUS unbesetzt. Statt der DUS wird eine Drahtbrücke eingelötet. Der Schalter wird für die C 64-Version nicht benötigt.

5.2 Setzen der Durchkontaktierungen X,XI,XII.

5.3 Setzen der Durchkontaktierung XIII im Feld A3-A4.

5.4 Einlöten der Bauteile und Einsetzen des NE 555.

5.5 Inbetriebnahme: Anschluß der beiden Fischertechnikschalter an Stift 18 und Stift 19 der Steckerleiste, gemeinsame Masse an Stift 1. Verwendung des DATANorf 10-Adapters ohne Stromversorgung.

5.5.1 Testen der Funktion des monostabilen Multivibrators, der einen positiven Impuls von ca. 2 ms Dauer bei jeder Schalterbetätigung erzeugt, mit Hilfe eines Programms :

```
** Testprogramm ROBOTER
10 FOR I=8192 TO 8227
20 READ A
30 POKE I,A
40 NEXT I
50 SYS 8192
100 DATA 32, 68,229,169, 16,141, 14,221,169, 5,141,
4,221,169, 0,141
110 DATA 5,221,169, 33,141, 14,221,173, 4,221,208,251,169,
73, 32,210
120 DATA 255, 76, 3, 32
```

Nach dem Starten des Programms erscheint nun wahlweise durch beide Fischertechnikschalter getaktet nach 5-maligem Betätigen der Schalter ein "I" auf dem Bildschirm. Dann arbeiten die Fischertechnik-Schalter prellfrei.

6. Fertigstellen des Feldes Aufzug

6.1 Bestücken des Feldes mit dem Sockel für den SN 74 LS 153 (die Kerbe des IC-Sockels auf die Kerbe des Sockelaufdrucks bringen), den 9 Widerständen 1 kOhm (braun,schwarz,rot,gold).

6.2 Setzen der Durchkontaktierungen XIV und XV (Feld A3-A4), XVII (Feld Roboter), XVI-XXII, XXIIIa und XXIVa, XXV-XXVII, XXVIII und XXX (Feld A3-A4).

6.3 Einlöten der Bauteile teilweise beidseitig.

6.4 Entfernen der Durchkontaktierung III. Damit wird verhindert, daß bei der Aktivierung der Ausgänge A3 und A4 mit "1" die Transistoren des Feldes A3-A4 in einen unzulässigen Betrieb geraten.

6.5 Inbetriebnahme unter Verwendung des DATANorf 10-Adapters ohne Stromversorgung: Anschluß der Aufzugschalter S0,S1,S2,S3,S4,S5,S6 und S7 an die Stifte 3 bis 10 des Interfaces und gemeinsam an Stift 2 . Eingeben des Testprogramms für die Abfrage der Eingänge E3 und E4 des Interfaces:

```
** Testprogramm für die Aufzugsteuerung
10 POKE 56579,63
20 POKE 56577,0
30 PRINT CHR$(147);
40 A=PEEK(56577)
50 IF 64 AND A THEN PRINT "E3 = 1":GOTO 70
60 PRINT "E3 = 0"
70 IF 128 AND A THEN PRINT "E4 = 1":GOTO 90
80 PRINT "E4 = 0"
90 PRINT CHR$(19);
100 GOTO 40
```

Nach dem Starten des Programms erscheint auf dem Bildschirm die Information, daß E3=0 und E4=0 ist. Es bewirkt die Schalter S3 und S4, daß E3=1 und E4=1 auf dem Bildschirm angezeigt wird, weil Buchse 5 und 4 des Interfaces diese Spannungsänderungen

erfahren und dies vom Programm gelesen wird.

Das Programm ändern, damit die Zustände von E2 und E5 angezeigt werden :

20 POKE 56577,8

In den Zeilen 50-80 E3 durch E2 sowie E4 durch E5 ersetzen.

Starten des Programms, Betätigen der Schalter S2 und S5.

Das Programm ändern, damit die Zustände von E1 und E6 angezeigt werden :

20 POKE 56577,16

In den Zeilen 50-80 E2 durch E1 sowie E5 durch E6 ersetzen.

Starten des Programms, Betätigen der Schalter S1 und S6 .

Das Programm ändern, damit die Zustände von E0 und E7 angezeigt werden :

20 POKE 56577,24

In den Zeilen 50-80 E1 durch E0 sowie E6 durch E7 ersetzen.

Starten des Programms, Betätigen der Schalter S0 und S7.

DATANorf 3: PUFFERPLATINE FÜR 8 AUSGÄNGE

Eigenschaften:

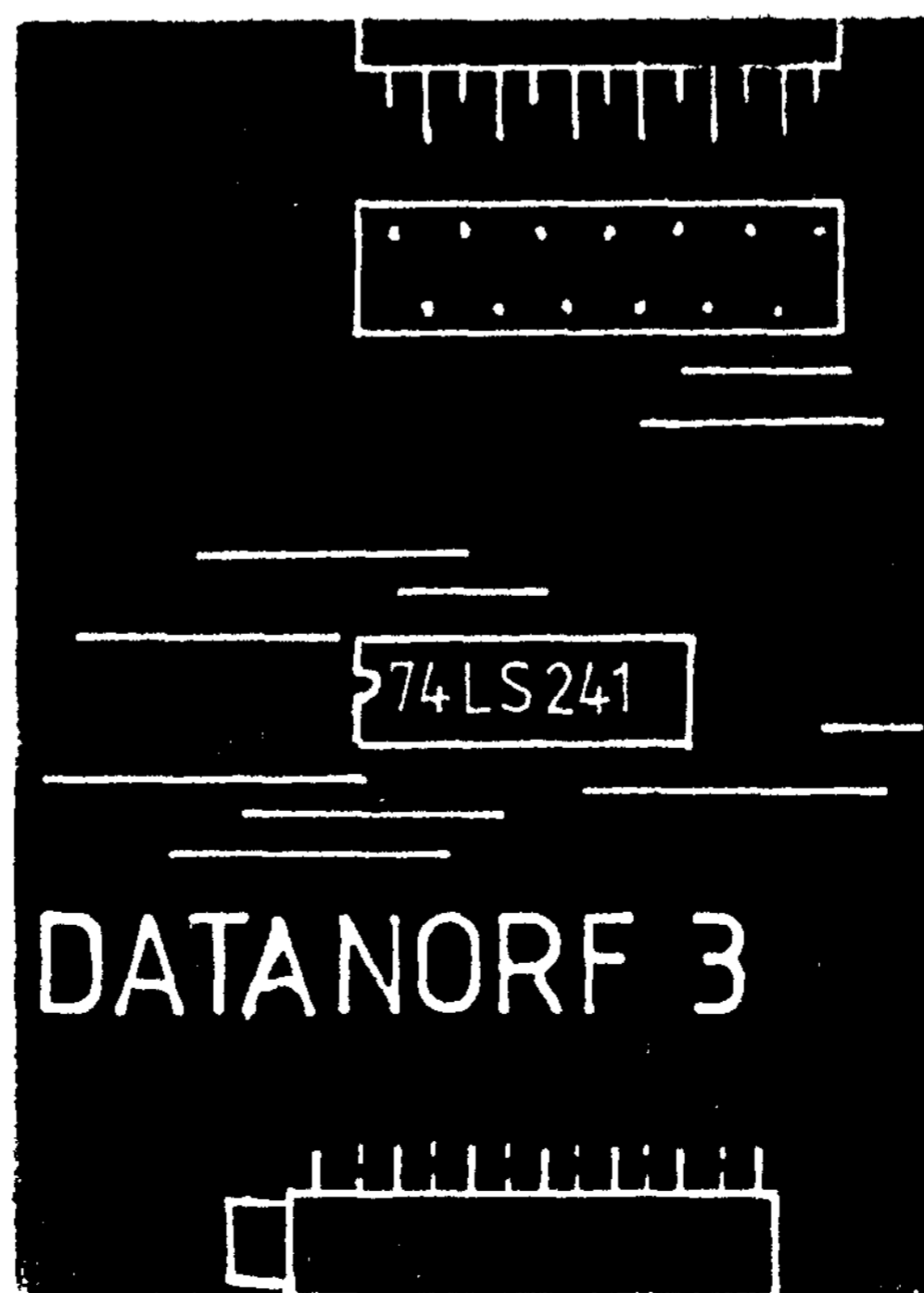
Die Pufferplatine wird mit Hilfe der 13-poligen Buchsenleiste auf den Expansionsstecker des WDR-1-Bit-Computers gesteckt. Die Stromversorgung geschieht durch den WDR-1-Bit-Computer. Sie verleiht den 8 Ausgängen einen maximalen Ausgangstrom von 15 bis 24 mA bei 5 V. Die 8 Ausgänge werden mit Hilfe einer 13-poligen Steckerleiste nach außen geführt. Zusätzlich können mit Hilfe einer 13-poligen Buchsenleiste weitere 8 Ausgänge parallel zu den vorhandenen betrieben werden. Eingänge sind nicht beschaltet. Durch Tausch des ICs 74 LS 244 mit einem 74LS240 können die 8 Ausgänge invertiert gepuffert betrieben werden. Bei Verwendung zweier hintereinandergesteckter Pufferplatten können mit Hilfe der beiden genannten ICs 8 nichtinvertierte und 8 invertierte Ausgänge gleichzeitig erzeugt werden.

Aufbau:

Die 13-polige Buchsenleiste wird entsprechend dem Bestückungsaufdruck hinter der 13. Buchse gekürzt und mit Sekundenkleber oben auf die Platine geklebt. Die Anschlüsse werden mit Hilfe der mitgelieferten Drahtenden mit der Platinenunterseite verbunden und verlötet. Nach Verlöten der 10 Drahtbrücken, des IC-Sockels, der 13-poligen Stecker- und der 13-poligen Buchsenleiste wird das IC seitenrichtig eingesetzt.

Inbetriebnahme:

Nach dem Aufbau der Pufferplatine kann diese an den 21-poligen Expansionsstecker des betriebsbereiten WDR-1-Bit-Computers bündig mit Pin 1 gesteckt oder auch wieder entfernt werden, ohne daß die Komponenten gefährdet werden.



DATANorf 4: SIEBENSEGMENTANZEIGE

Eigenschaften:

Die 7-Segmentanzeige wird bündig zu Pin 1 auf den Expansionsport des WDR-1-Bit-Computers aufgesteckt. Die Stromversorgung geschieht durch den WDR-1-Bit-Computer. Die 7-Segmentanzeige kann seriell alphanumerische Zeichen, die aus 7 Segmenten aufgebaut sind, erzeugen, wenn sie durch ein geeignetes Programm angesteuert wird. Die Anzahl der Zeichen richtet sich nach dem verwendeten Speicher. Der Bildaufbau eines alphanumerischen Zeichens kann sowohl demonstriert werden, er kann aber auch flackerfrei erfolgen.

Aufbau:

Der Aufbau geschieht entsprechend dem Bestückungsplan.

Inbetriebnahme:

Mit Eingabe des Programms "Test der 7-Segmentanzeige" leuchten nach dem 5. Programmschritt nacheinander alle 7 Segmente der Reihe nach auf.

Test der 7-Segmentanzeige

*Einzeltakt

INIT

STO 0

STO 1

STO 2

STO 3

STO 4

STO 5

STO 6

STO 7

Mögliche Fehlerquellen:

Es leuchten zwei Segmente gleichzeitig auf:

Prüfung mit dem Ohmmeter: zwei Leiterbahnen sind vor oder nach dem IC miteinander verbunden.

Es leuchtet ein Segment nicht auf:

Prüfung mit dem Ohmmeter. Ein Segment der Siebensegmentanzeige ist defekt? Der dazugehörige Widerstand 100 Ohm ist defekt bzw. nicht richtig eingelötet oder vertauscht? Das IC ist defekt? Die zu dem Segment gehörende Leiterbahn ist vor oder hinter dem IC unterbrochen? Die Buchsenleiste ist nicht richtig angelötet? Der Expansionsstecker des WDR-1-Bit-Computers ist nicht richtig verlötet?

Es leuchtet kein Segment auf:

Prüfung des ICs und des Transistors ohne Auslötarbeiten mit dem Voltmeter in folgender Reihenfolge:

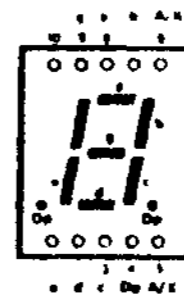
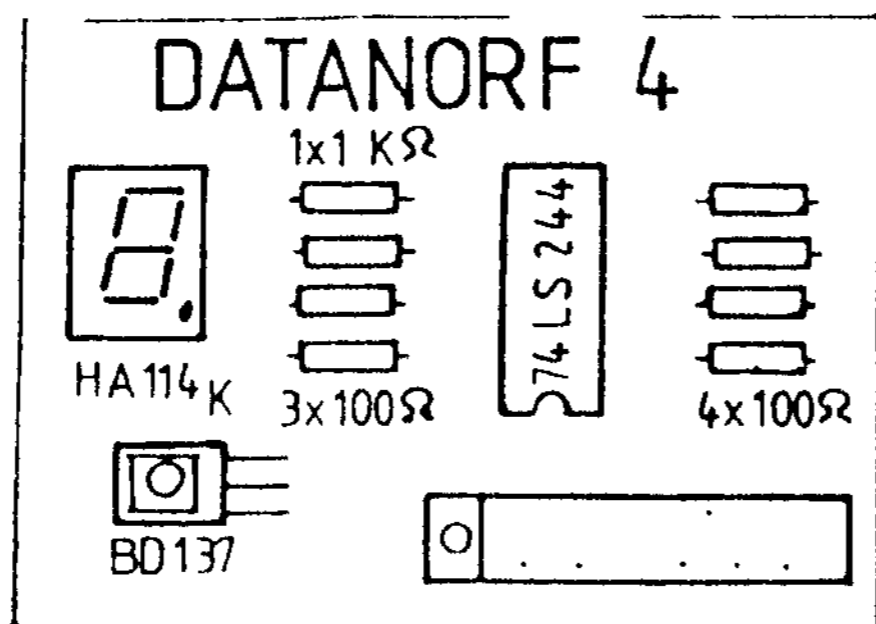
Liegt an Pin 20 des ICs 5 V an? Liegen nach dem Programmschritt STO 0 an der Basis des Transistors (das Bein, das zur 7-Segmentanzeige schaut) bzw. an Pin 12 oder Pin 8 des ICs und an Pin 13 der Buchsenleiste 5 V an? Liegt an Pin 5 der 7-Segmentanzeige Masse an, wenn der Pluspol des Voltmeters mit Pin

2 der Buchsenleiste verbunden wurde und liegt Pin 10 des ICs ebenfalls auf Masse?

Liegen nach dem Programmschritt ST0 1 Pin 6 und Pin 14 des ICs sowie Pin 8 der 7-Segmentanzeige auf 5 Volt?

Prüfung der 7-Segmentanzeige:

Pin 5 der 7-Segmentanzeige mit Masse verbinden. Einen Widerstand 100 Ohm mit 5 V verbinden. Das andere Ende des Widerstandes nacheinander an Pin 1-3 und 7-10 der 7-Segmentanzeige verbinden. Nun müssen die einzelnen Segmente nacheinander aufleuchten.



Standard	Low Power Schottky	Low Power High Speed
Standard	µs	
Max Ausgangsstrom bei I _{OL}	15 mA	
Max Ausgangsstrom bei I _{OH}	24 mA	
Max Reststrom I _{OL}	20 µA	
Max Reststrom I _{OH}	20 µA	
Typ Verzögerungszeit	12 ns	
Typ Leitungsaufnahme	135 mW	

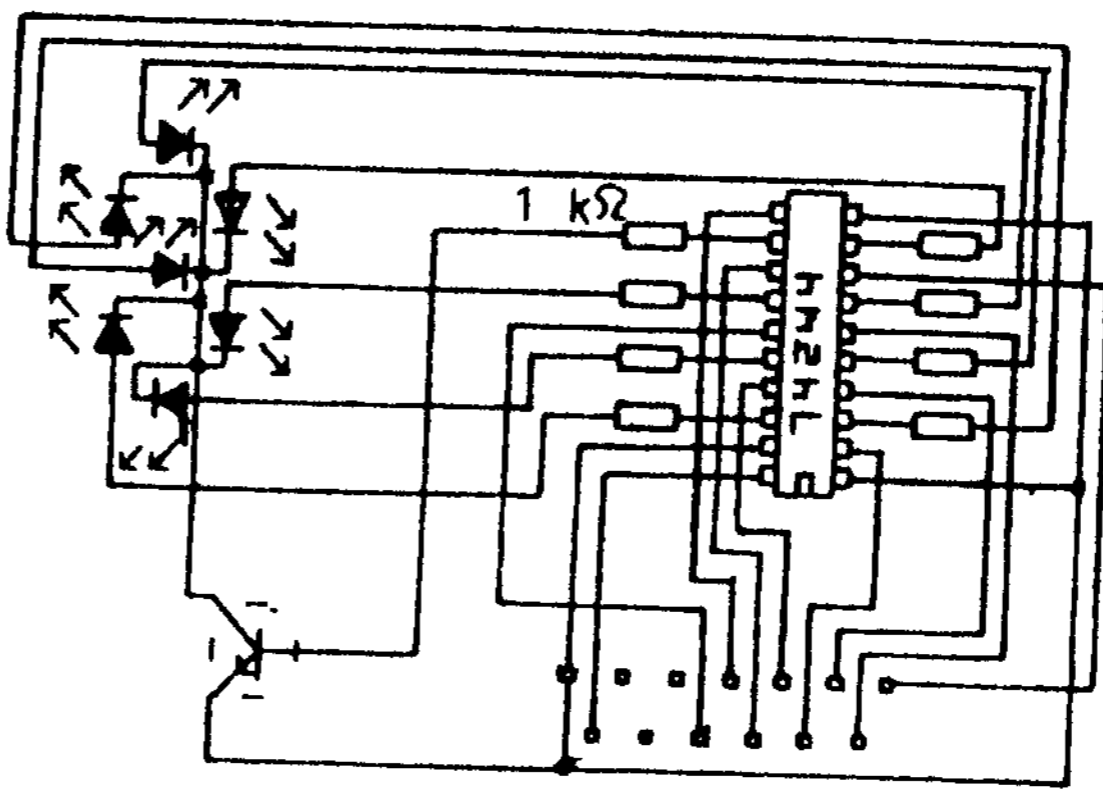
Input	Output
H	A
H	B
H	C
H	D
H	E
H	F
H	G

Positive Logik

Modus	Zustand	Leitwert
Eingangs D	L	0.5
	H	0.5
Ausgangs	L	1.0
	H	60.0

© nichtverändernde Version des LS 240

ACH (Bus-Leitungsanforder) (Tri-State)
 SN74LS244
 SN54LS244
 SNM1LS244



DATANorf5: MOTORSTEUERUNG

Eigenschaften:

Ein Elektromotor (mit einer Spannung 1,5V bis max. 45 V) kann (bis zu 3 A bzw. bei einer Leistungsaufnahme bis max. 30 W bei Verwendung von Kühlkörpern) mit Hilfe eines geeigneten Programms ein- und ausgeschaltet sowie umgepolt werden.

Aufbau:

Zunächst wird die 13-polige Buchsenleiste neben Pin 13 gekürzt und die Unterseite auf die Platine an der dafür vorgesehenen Stelle angelötet.

Entsprechend dem Bestückungsplan werden die 4,7 kOhm-Widerstände (gelb, violett, rot, gold) eingesetzt und verlötet. Zwei der abgekniffenen Drahtenden stellen die Verbindung zwischen der Oberseite der Buchsenleiste und der Platine her. Nach dem Einlöten der 22-kOhm-Widerstände (rot, rot, orange, gold) werden die restlichen Bauteile entsprechend dem Bestückungsplan eingelötet und die Drahtenden abgekniffen.

Inbetriebnahme:

Die Platine für die Motorsteuerung wird auf den Expansionsstecker des WDR-1-Bit-Computers bündig zu Pin 1 gesteckt. Je nach Betriebsspannung des Elektromotors wird an der 4-poligen Schraubklemme bei "-" und "+" eine externe Spannungsversorgung hergestellt. (Sehr geeignet für den Test ist die Niedervoltbohrmaschine und ihre Spannungsversorgung, mit deren Hilfe man die zu testende Platine gebohrt hat). Der Elektromotor selbst wird an die beiden restlichen Pole der 4-poligen Schraubklemme angeschlossen. Mit Hilfe des Testprogramms "Motortest 1" muß sich der Motor in Gang setzen lassen.

****Motortest 1****

*Betrieb: Handtakt bis STO 3

```
INIT
STO 3
```

Der Motor läuft nicht?

Mit Hilfe des Voltmeters werden folgende Messungen durchgeführt: Pin 10 der Buchsenleiste und die Basis des rechten BC 548 (neben der Beschriftung "DATANorf" müssen 5V führen. Am mittleren Bein des rechten BD 177 und des rechten BD 176 sowie an der Basis (mittleres Bein) des linken BC 250 müssen 0 V Spannung liegen. Am mittleren Bein des linken BD 177 und am mittleren Bein des linken BD 176 müssen 5 V liegen. Ebenso muß an der Basis des rechten BC 250 5 V liegen.

Liegt an der Basis des rechten BC 548 5 V an und die Kollektoren des rechten BD 176 und BD 177 (jeweils mittleres Bein) haben keine 0 V, so müssen beide Transistoren ausgetauscht werden. Liegt nach dem Austausch der Transistoren an den Kollektoren 0 V an und läuft der Motor immer noch nicht, müssen der linke BC 250 und BD 176 ersetzt werden.

Nun wird die Umpolung des Motors mit Hilfe des Programms "Motortest 2" getestet.

****Motortest 2****

*Betrieb: Handtakt bis "STO 4"

```
INIT
STO 3
```

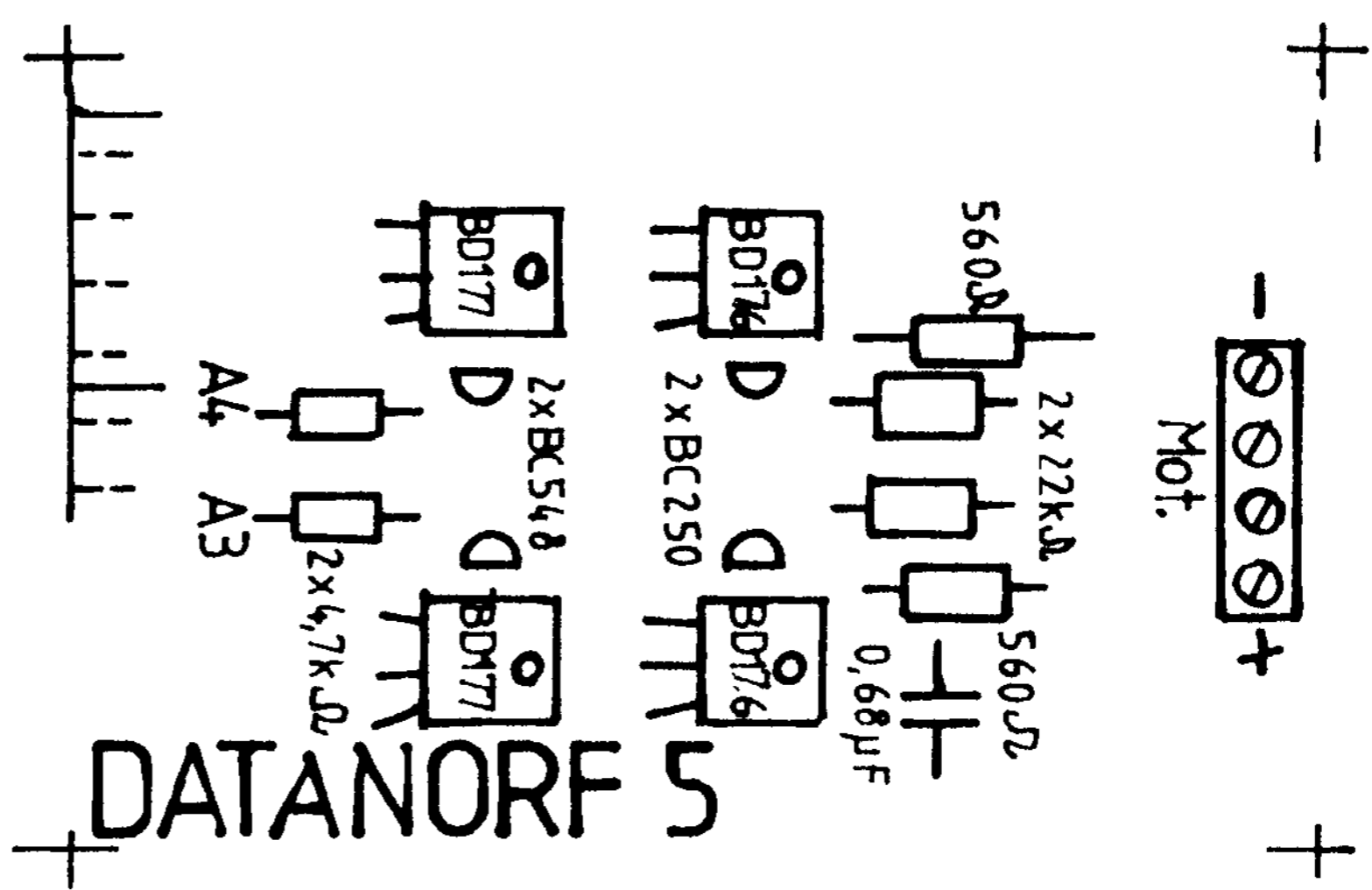
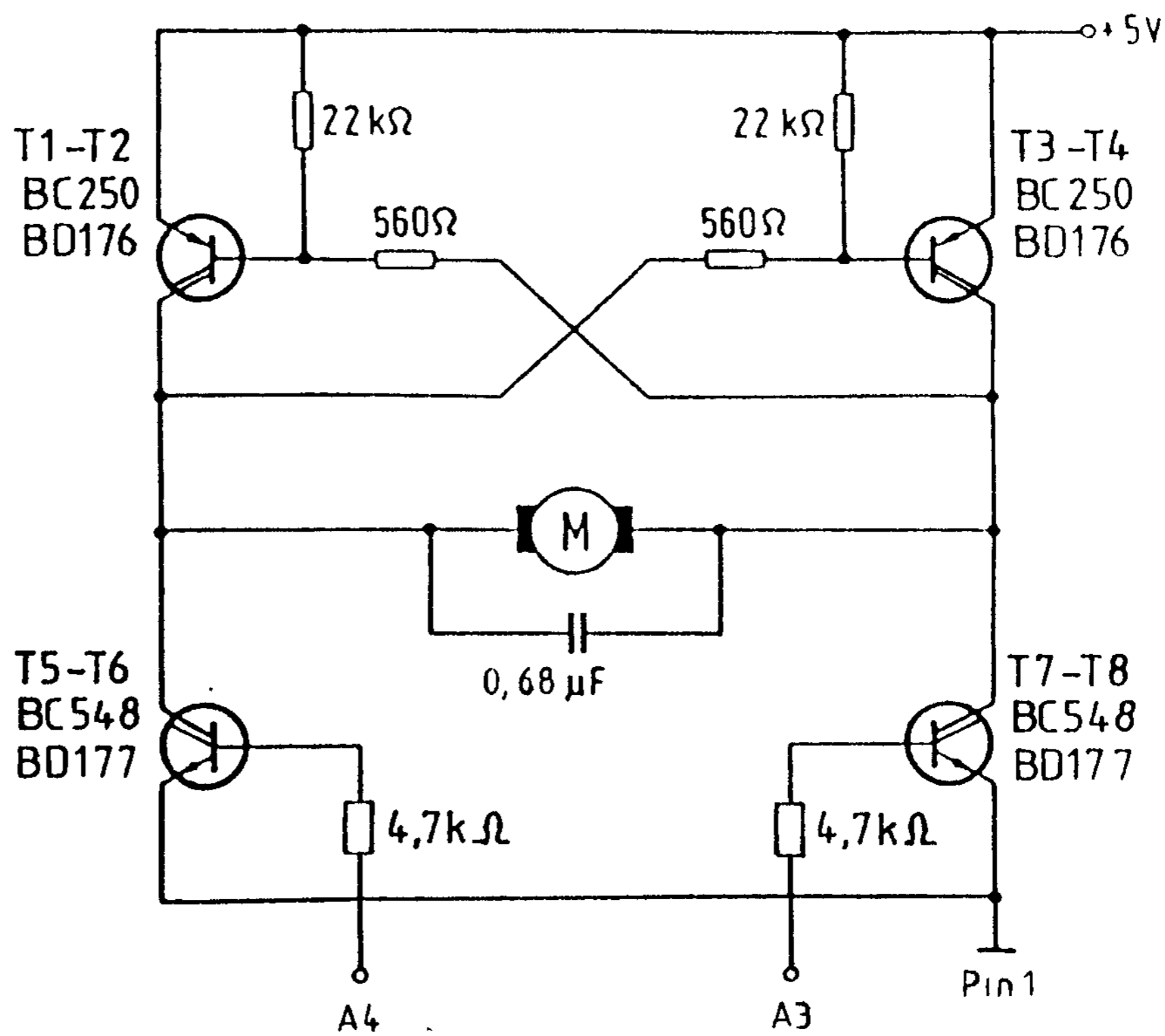
STOC 3

STO 4

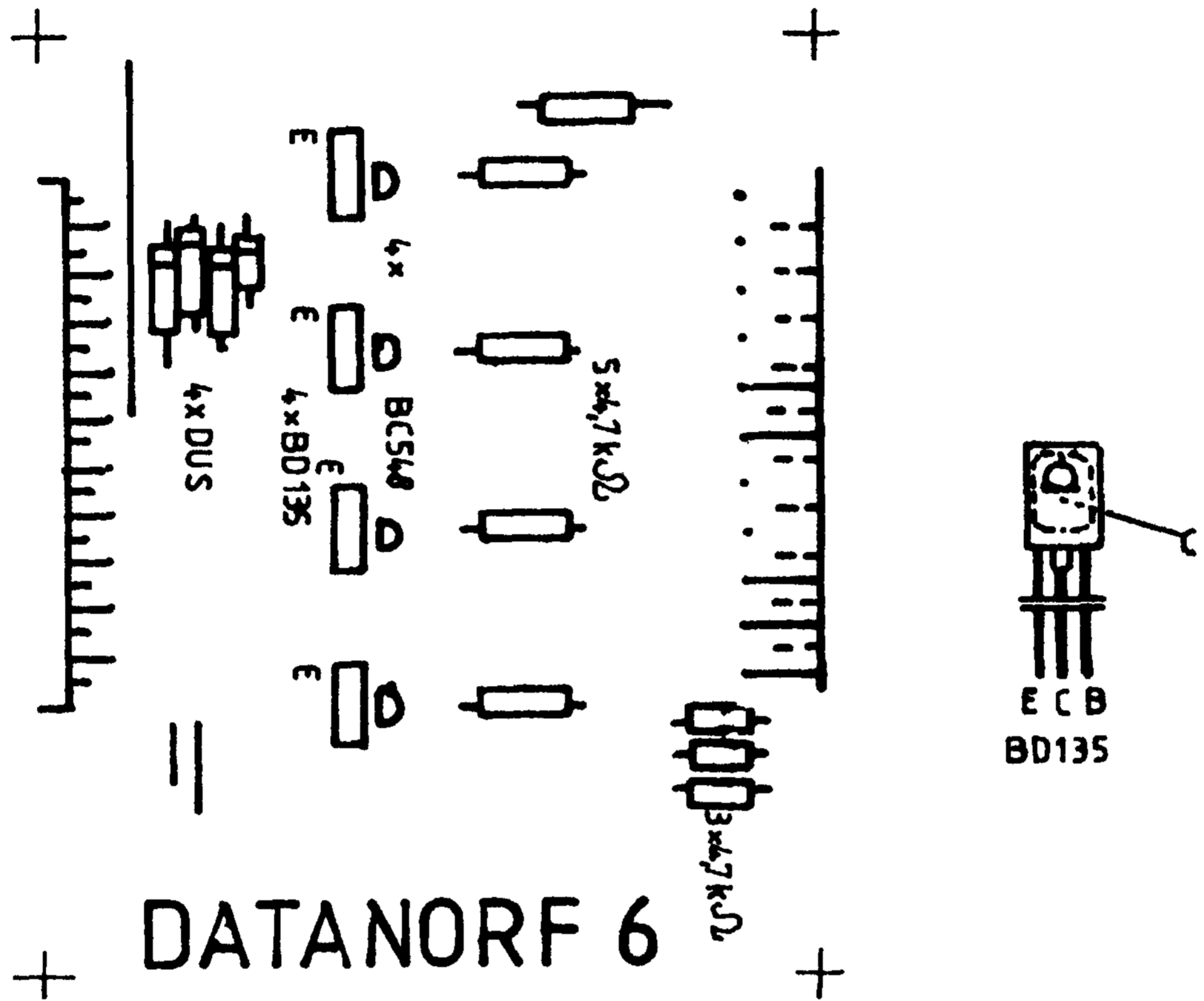
Der Motor läßt sich nicht unpolen?

Pin 9 der Buchsenleiste und die Basis des linken BC 548 müssen 5V führen. Am mittleren Bein des linken BD 177 und des linken BD 176 sowie an der Basis (mittleres Bein) des rechten BC 250 müssen 0 V Spannung liegen. Am mittleren Bein des rechten BD 177 und am mittleren Bein des rechten BD 176 müssen 5 V liegen. Ebenso muß an der Basis des linken BC 250 5 V liegen.

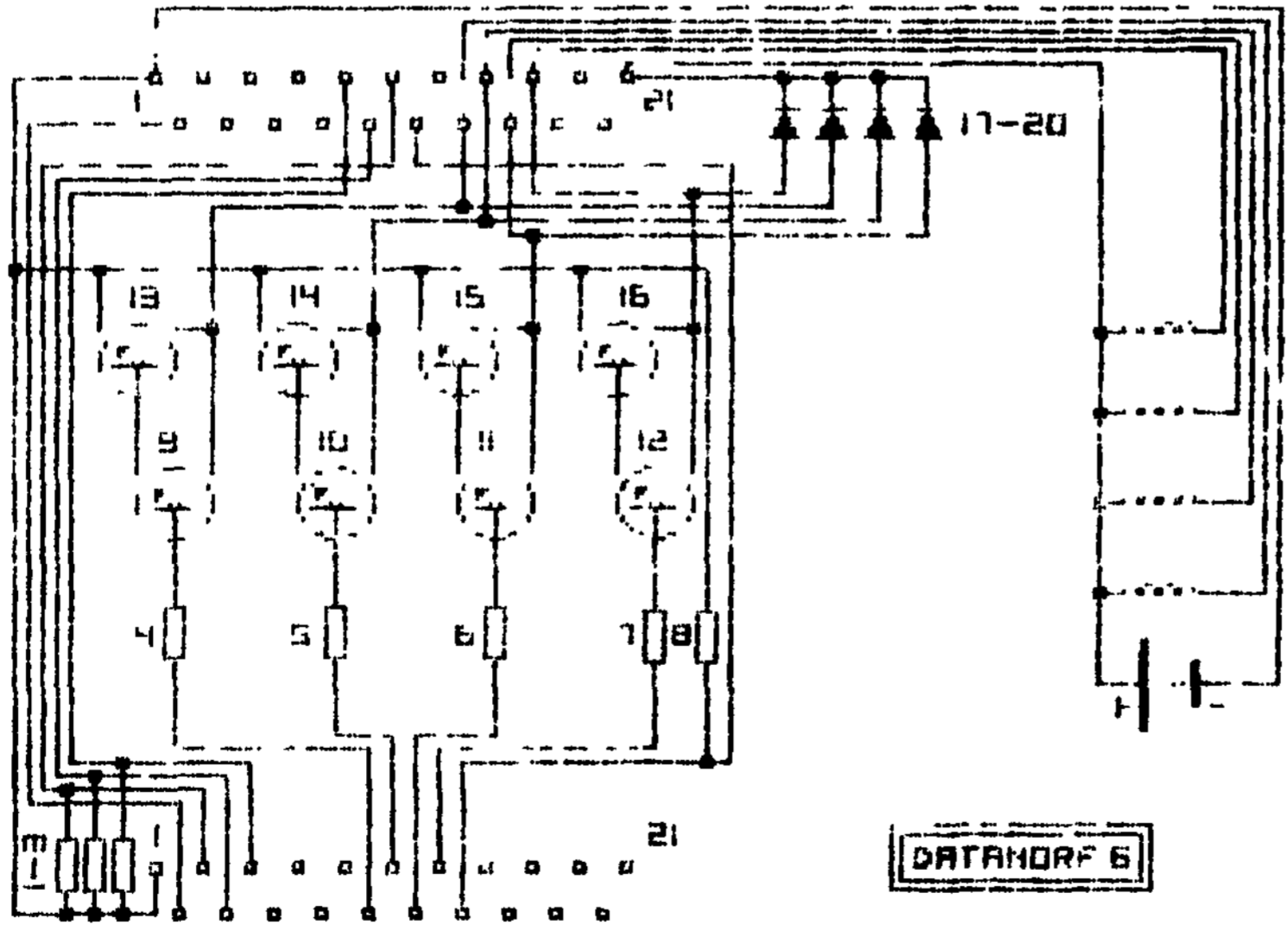
Liegt an der Basis des linken BC 548 5 V an und die Kollektoren des linken BD 176 und und BD 177 (jeweils mittleres Bein) haben keine 0 V, so müssen beide Transistoren ausgetauscht werden. Liegt nach dem Austausch der Trasistoren an den Kollektoren 0 V an und läuft der Motor immer noch nicht, müssen der rechte BC 250 und BD 176 ersetzt werden.



DATANORF 5



DATANORF 6



DATANorf 6: SCHRITTMOTORSTEUERUNG

Eigenschaften:

Ein 4-Strang-Unipolar-Schrittmotor SMS 35-48 (Conrad, Hirschau Best. Nr. 539015) kann mit Hilfe eines geeigneten Programms gesteuert werden. Ebenso können 4 Elektromotoren ein- bzw. ausgeschaltet werden. Die Steuerung von 4 Relais ist ebenfalls möglich. An einer 21-poligen Steckerleiste sind zusätzlich die Stromversorgungsanschlüsse des WDR-1-Bit-Computers und die vier Eingänge herausgeführt.

Aufbau:

Zunächst wird die Unterseite der 21-poligen Buchsenleiste auf der Platine verlötet. Nach dem Einlöten der Widerstände und dem Abkneifen der Drahtenden werden die fünf Verbindungen von der Buchsenleiste zur Platine gezogen und verlötet. Die restlichen Bauteile werden entsprechend dem Bestückungsplan verlötet.

Inbetriebnahme:

Da bei einer Funktionsüberprüfung mit einem Schrittmotor eine Fehlfunktion nicht immer eindeutig erkannt wird, empfiehlt sich das Austesten des Interfaces auf andere Weise.

Nach dem Aufstecken auf den Expansionsstecker wird das Testprogramm "Test der Schrittmotorsteuerung" eingegeben und die Funktion der Ausgänge mit Hilfe eines Niedervoltelektromotors (Bohrmaschine, mit der die Platine gebohrt wurde) überprüft. Die Spannungsquelle für den Elektromotor wird am Minuspol mit dem Minuspol des WDR-1-Bit-Computers verbunden. Eine Leitung des Elektromotors ist dauerhaft mit dem Pluspol der Spannungsquelle verbunden. Die andere Leitung des Elektromotors wird nacheinander mit den Anschlüssen 17 bis 14 der 21-poligen Steckerleiste des Interfaces verbunden.

Test der Schrittmotorsteuerung

*Einzeltakt

```
INIT
STO 0
STOC 0
STO 1
STOC 1
STO 2
STOC 2
STO 3
STOC 3
```

Bei dem Programmschritt STO 0 muß der Elektromotor an Pin 17 herangeführt werden, er muß nun laufen. Mit dem nächsten Programmschritt muß er wieder stoppen.

Bei dem Programmschritt STO 1 muß der Elektromotor an Pin 16 herangeführt werden, er muß nun laufen. Mit dem nächsten Programmschritt muß er wieder stoppen.

Bei dem Programmschritt STO 2 muß der Elektromotor an Pin 15 herangeführt werden, er muß nun laufen. Mit dem nächsten Programmschritt muß er wieder stoppen.

Bei dem Programmschritt STO 3 muß der Elektromotor an Pin 14 herangeführt werden, er muß nun laufen. Mit dem nächsten Programmschritt muß er wieder stoppen.

DATANorf 7: RAM/EPROM-PLATINE

Eigenschaften:

Die RAM/EPROM-Platine kann die RAM-Platine des WDR-1-Bit-Computers ersetzen und bietet folgende Erweiterungen:

Bei Betrieb mit dem 6116 als RAM oder dem 2716 als EPROM stehen 2 KByte Speicherplätze, also die achtfache Menge zur Verfügung, die in 8 Blöcken mit Hilfe eines 3-fach DIL-Schalters adressiert werden können. Mit Hilfe von DATANorf 19 ist eine automatische Adressenumschaltung über geeignete Software möglich. Mit Hilfe des Eprommerzusatzes DATANorf 31 können Programme mit Hilfe des WDR-1-Bit-Computers in EPROMS gelegt werden.

Aufbau:

Die Bauteile werden entsprechend dem Bestückungsaufdruck eingesetzt. Zunächst werden die Widerstände eingelötet, damit man deren abgekniffene Beine als Durchkontaktierungen benutzen kann. Bei Verwendung der Karte als RAM-Version muß die Leiterbahn unter dem 15 KOhm-Widerstand, der neben dem 100 Ohm-Widerstand liegt, unterbrochen werden.

Inbetriebnahme:

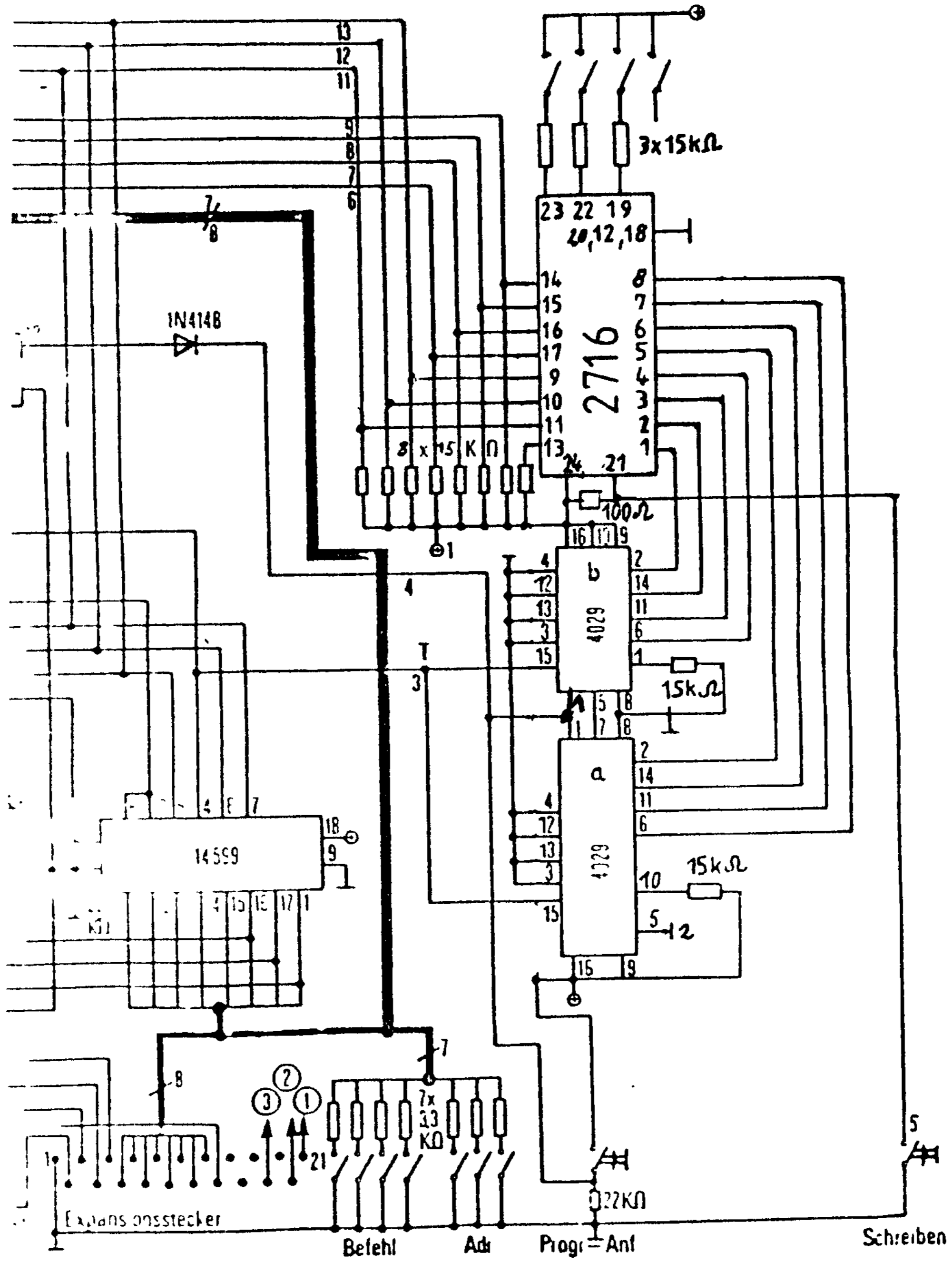
Die RAM/EPROM-Platine wird als RAM-Version wie die normale RAM-Karte des WDR-1-Bit-Computers programmiert. Die Codierung der Schalter ist unten angegeben.

Bei der Verwendung der RAM/EPROM-Platine als EPROM-Version ist eine Programmierung nicht mehr möglich, da das Programm bereits im EPROM vorhanden ist. Mit Hilfe der Rücksetz- und Programmvorwahltaster wird an den Anfang des abzurufenden Programms gesprungen.

Über die Auswahl der in einem EPROM vorhandenen möglichen Programme informiert die folgende Zusammenstellung:

Inhalt des EPROMS		Kodierung der Schalter
Adresse	Inhalt	
0000 - 00FF	Aufzug	x000
0100 - 01FF	Musik	x001
0200 - 02FF	Sortieranlage	x010
0300 - 03FF	Fußgängerampel	x011
0400 - 04FF	Bohrroboter	x100
0500 - 05FF	Schweißroboter	x101
0600 - 06FF	"Teach-in"-Roboter	x110
0700 - 07FF	Plotter	x111

DATANORF 7



DATANorf 8: MORSEZEICHENGENERATOR

Eigenschaften:

Mit Hilfe geeigneter Programme können Morsezeichen erzeugt werden, die mit Hilfe eines Lautsprechers hörbar gemacht werden.

Aufbau:

Die 13-polige Buchsenleiste wird neben Pin 13 gekürzt und an der Unterseite an die Platine gelötet. Nach dem Einsetzen des Widerstandes 2,2 KOhm (rot, rot, rot, gold) werden die Drahtenden abgekniffen. Sie werden für die Drahtbrücke und die eine Verbindung von der Oberseite der Buchsenleiste zur Platine benötigt. Wenn die restlichen Bauteile eingelötet sind, wird der NE 555 in den Sockel gesteckt und der Lautsprecher auf die Platine geklebt. Zum Schluß werden die beiden Verbindungen zum Lautsprecher angelötet.

Inbetriebnahme:

Der Test geschieht mit Hilfe des Testprogramms "Test des Morsezeichengenerators".

****Test des Morsezeichengenerators****

*Einzeltakt

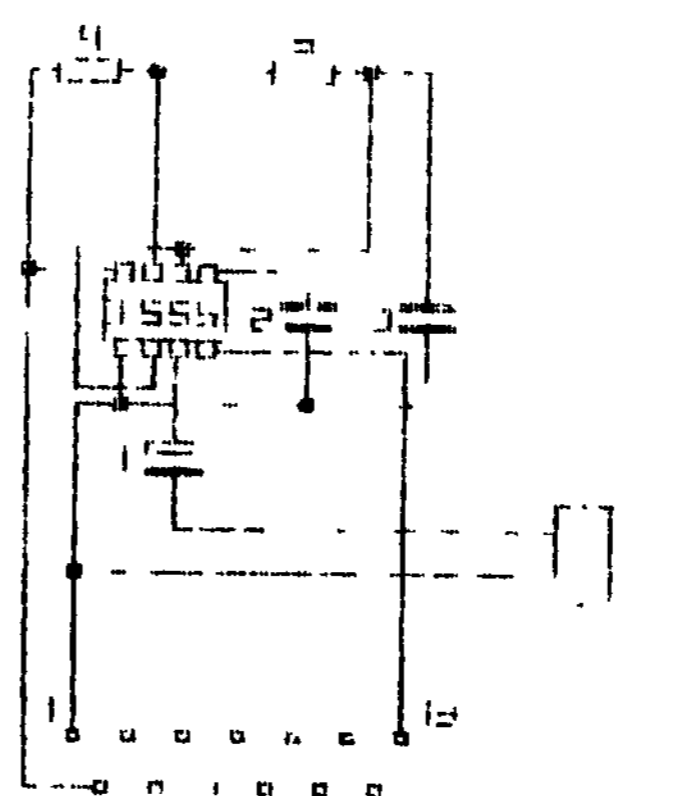
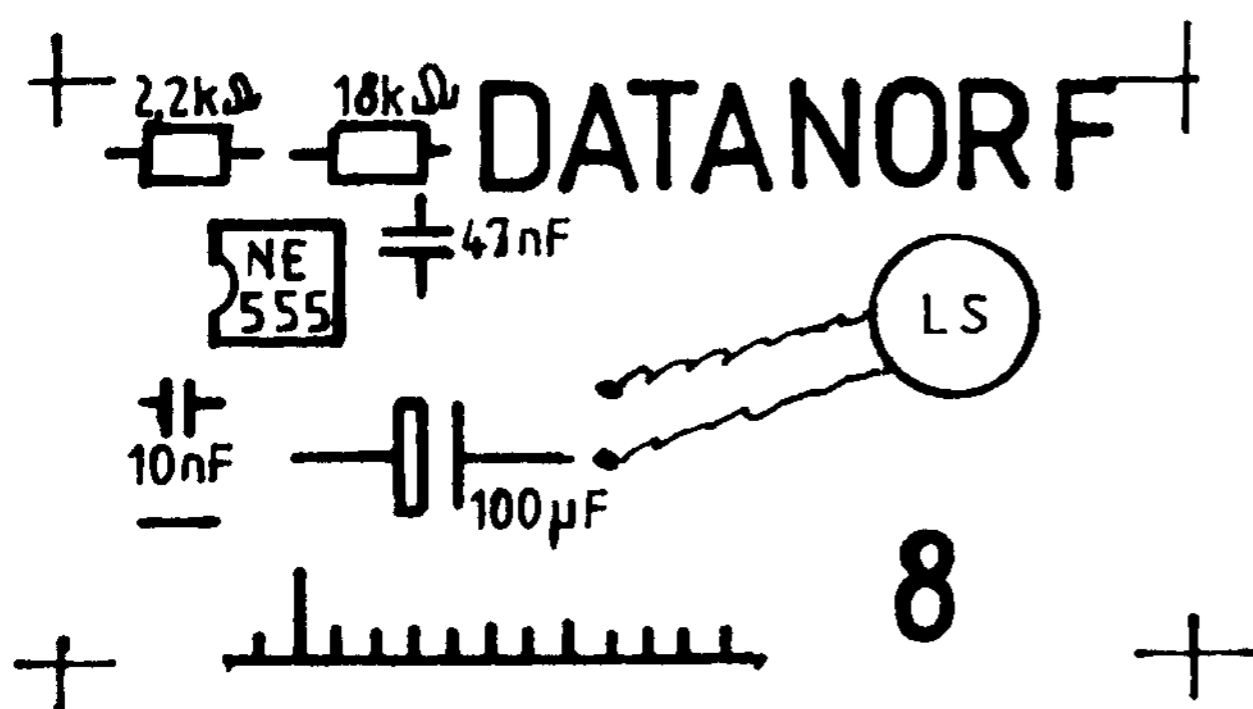
INIT

STO 0

Der Lautsprecher bleibt stumm?

Mit Hilfe des Voltmeters wird überprüft, ob an Pin 4 und Pin 8 des ICs 5 V anliegen, wenn das Programm auf dem Programmschritt STO 0 verharret. Mit Hilfe des Ohmmeters werden die Leiterbahnen überprüft. Das IC wird ersetzt. Die passiven Bauteile nachlöten und den Lautsprecher mit Hilfe des Ohmmeters überprüfen (4-8 Ohm).

Detaillierte Hinweise zur Erzeugung komplexer Wörter mit Hilfe von Morsezeichen finden sich in den Veröffentlichungen cQ-DL 6-10 1984.



DATANorf 9: Musikbox

Eigenschaften:

Mit Hilfe geeigneter Programme können einfache Melodien erzeugt werden, die mit Hilfe eines Lautsprechers hörbar gemacht werden. Die Stromversorgung geschieht durch den WDR-1-Bit-Computer. Ein Lautstärkereglert ist vorhanden.

Aufbau:

Der Aufbau geschieht entsprechend dem Bestückungsplan. Die Codierung der Farbringe der Widerstände gelingt mit Hilfe der Tabelle im Anhang. Der Lautsprecher wird aufgeklebt und mit Hilfe zweier Drähte mit der Platine verbunden. Die Drähte für die Stromversorgung sowie die Drähte für die 8 Steuereingänge werden entsprechend dem unteren Hinweis mit der mitgelieferten Buchsenleiste verlötet.

Inbetriebnahme:

Die Inbetriebnahme gelingt mit Hilfe des Testprogramms 1 "Morsezeichengenerator"

** Testprogramm 1 Morsezeichengenerator**

Langsamtakt

INIT

STO 0

STO 1

STO 2

STO 3

STO 4

STO 5

STO 6

STO 7

JMP x

Der Lautsprecher bleibt generell stumm?

Das Potentiometer wird verändert. Mit Hilfe des Voltmeters wird überprüft, ob an Pin 8 des ICs 5 V anliegt. Das IC wird entfernt und das Testprogramm 2 eingegeben.

Testprogramm 2 Morsezeichengenerator

Langsamtakt

INIT

STO 0

STOC 0

JMP x

Das Programm wird gestartet und eine elektrische Verbindung von Ausgang A0 (Pin 13 des Expansionssteckers) des WDR-1-Bit-Computers zu dem Mittelabgriff des Potentiometers auf der Musikboxplatine hergestellt. Schweigt der Lautsprecher noch immer, sollten die beiden Transistoren ausgewechselt werden bzw. der Lautsprecher mit Hilfe des Ohmmeters durchgemessen werden (4-8 OHM). Arbeitet der Verstärkerteil nach diesem Test, muß das IC ersetzt werden.

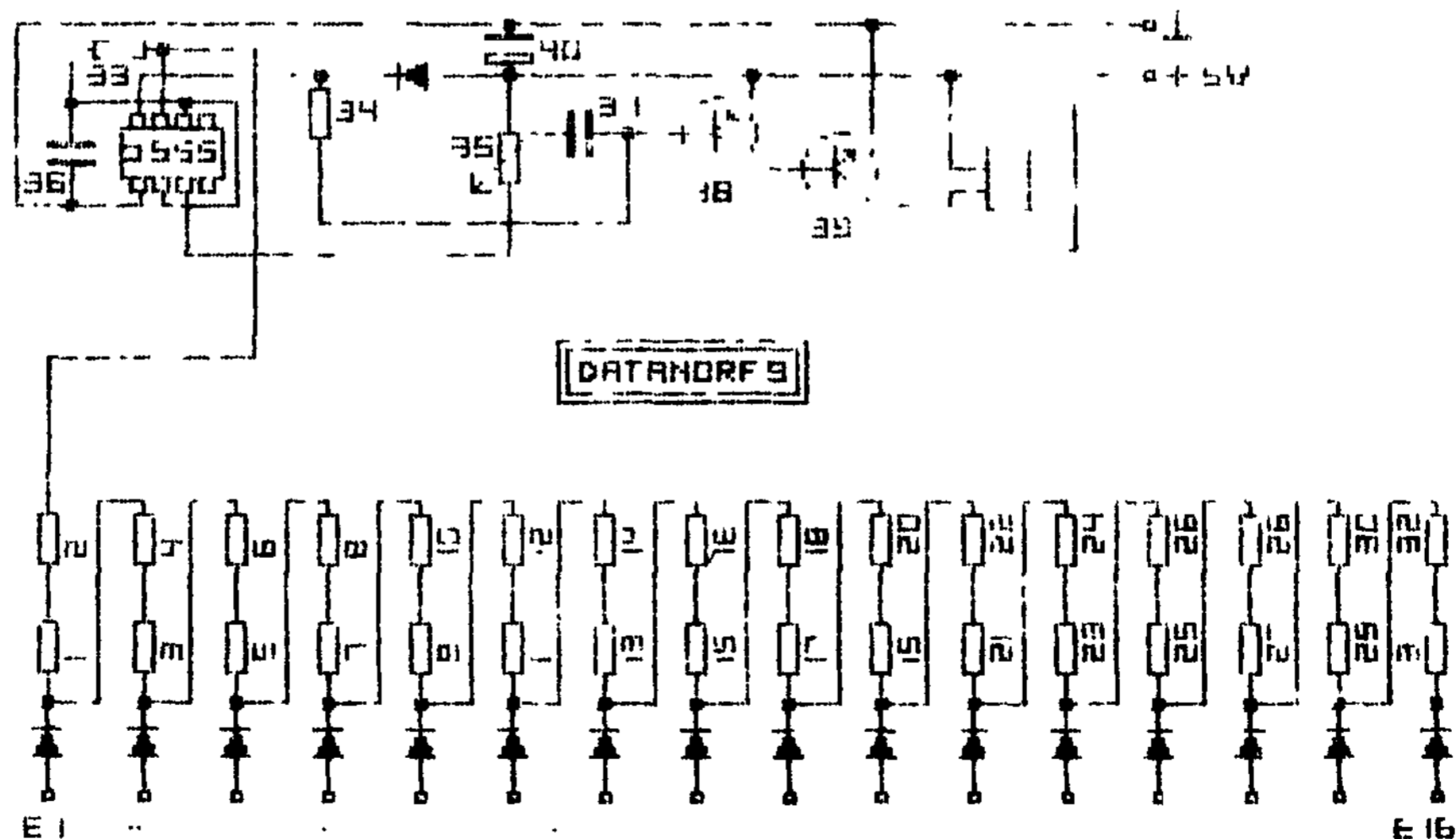
Der Lautsprecher bleibt ab einem bestimmten Programmschritt stumm? Dann ist die Verbindung zwischen E1 bis E 16 an einer Stelle unterbrochen.

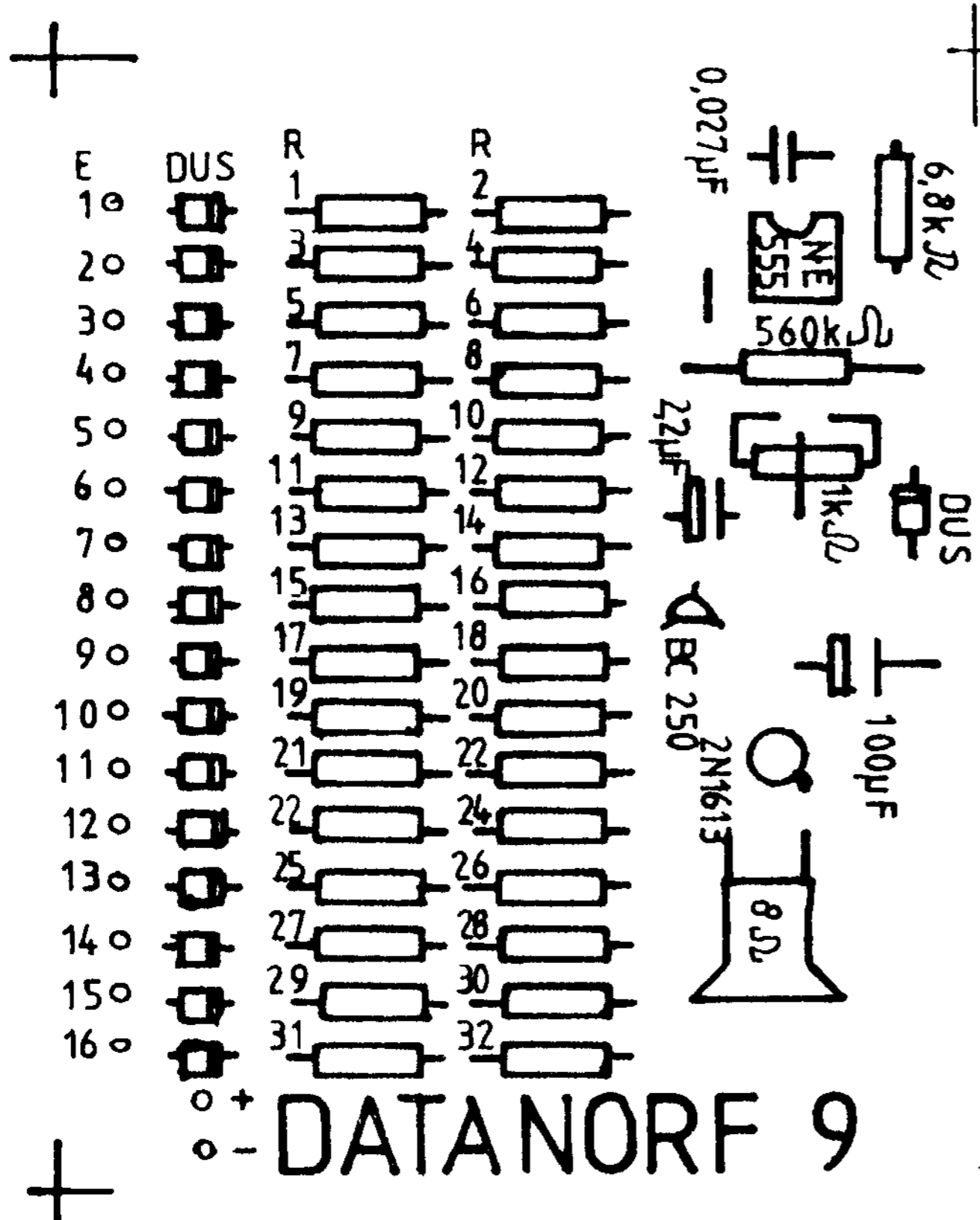
Die verwendeten Widerstände erzeugen Töne, die nicht ideal zu nennen sind.

R 1 = 2,7 kOhm	R 2 = 27 kOhm
R 3 = 1,8 kOhm	R 4 = 2,7 kOhm
R 5 = 2,2 kOhm	R 6 = 1,5 kOhm
R 7 = 470 Ohm	R 8 = 1,8 kOhm
R 9 = 100 Ohm	R10 = 3,9 kOhm
R11 = 1,2 kOhm	R12 = 1,8 kOhm
R13 = 22 Ohm	R14 = 3,9 kOhm
R15 = 22 Ohm	R16 = 3,9 kOhm
R17 = 680 Ohm	R18 = 3,9 kOhm
R19 = 470 Ohm	R20 = 3,9 kOhm
R21 = 82 Ohm	R22 = 3,3 kOhm
R23 = 680 Ohm	R24 = 4,7 kOhm
R25 = 120 Ohm	R26 = 4,7 kOhm
R27 = 560 Ohm	R28 = 3,9 kOhm
R29 = 560 Ohm	R30 = 5,6 kOhm
R31 = 560 Ohm	R32 = 5,6 kOhm
R33 = 6,8 kOhm	R34 = 560 kOhm
R35 = 100 kOhm-Poti	
C36 = 0,027 uF	C37 = 2,2 uF
C40 = 100 uF, 9V	
T38 = BC 250	T39 = 2N1613

Für die im EPROM von DATANorf 7 abgelegte Melodie wurden die Eingänge der DATANorf 9-Platine wie folgt mit den Ausgängen des WDR-1-Bit-Computers verbunden:

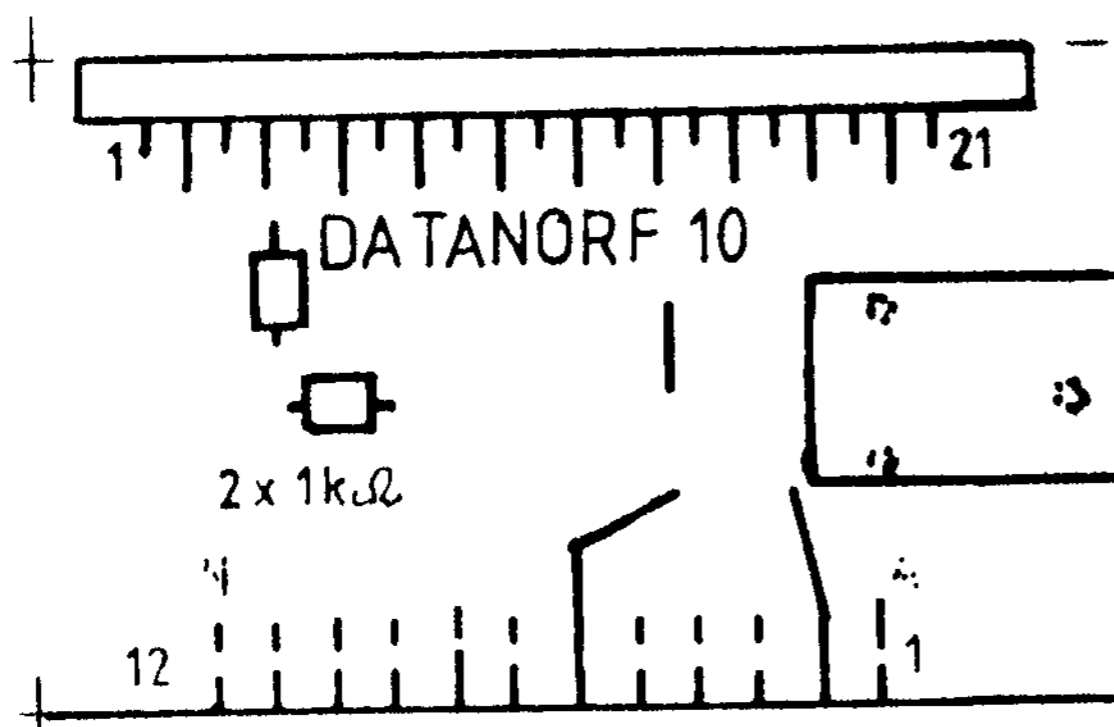
E16 mit A0
 E14 mit A1
 E12 mit A2
 E11 mit A3
 E 9 mit A4
 E 7 mit A5
 E 6 mit A6
 E 4 mit A7





DATANorf 10: C 64-Adapter

Der Adapter ermöglicht das Betreiben der WDR-1-Bit-Computerperipherie DATANorf 2,3,4,5,6,8,9,11,12,13,14,22 und 23. Der Zusammenbau geschieht entsprechend dem Bestückungsaufdruck. Hierbei ist lediglich zu beachten, daß eine Anschlußreihe des Steckers für den Userport für den C 64 an der Unterseite der Platine aus Stabilitätsgründen vollständig angelötet wird. Auf der Oberseite dieses Steckers müssen lediglich zwei Drahtbrücken angebracht werden. Die Drahtbrücken rekrutieren sich aus den abgekniffenen Enden der eingelöteten Widerstände. Je nach gewünschter Stromversorgung kann mit Hilfe eines 2,5 mm Klinkensteckers (Spitze = +) diese hinzugeschaltet werden.



DATANorf 11: Relaisplatine

Eigenschaften: Die Relaisplatine enthält 1 Relais mit je einem Arbeits- und Ruhekontakt. Der Ansteuer Eingang E0 - E7 ist wählbar.

Aufbau:

Zunächst wird die 13-polige Buchsenleiste neben Pin 13 gekürzt und die Unterseite auf die Platine an der dafür vorgesehenen Stelle angelötet. Entsprechend dem Bestückungsplan werden die Bauteile eingelötet. Ein abgekniffenes Drahtende stellt die Verbindung zwischen der Oberseite der Buchsenleiste Pin 1 und der Platine her. Soll der Ausgang A0 zur Ansteuerung dienen, so muß eine weitere Verbindung von der Buchsenleiste Pin 13 zur Platine hergestellt werden. Ebenfalls muß auf der Platinenunterseite mit Hilfe eines Lötkecks oder einer Drahtbrücke die Unterbrechung für die Eingangsauswahl hergestellt werden..

Inbetriebnahme:

Die Inbetriebnahme geschieht nach dem Aufstecken der Relaisplatine bündig zu Pin 1 des Expansionssteckers des WDR-1-Bit-Computers mit Hilfe des Programms "Relaistest".

****Relaistest****

*Einzeltakt

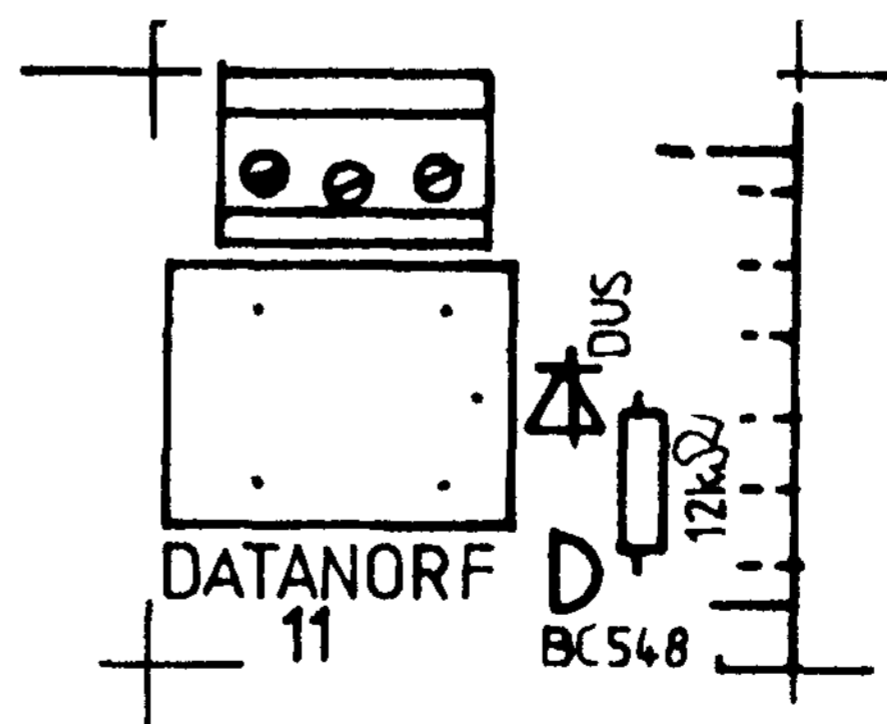
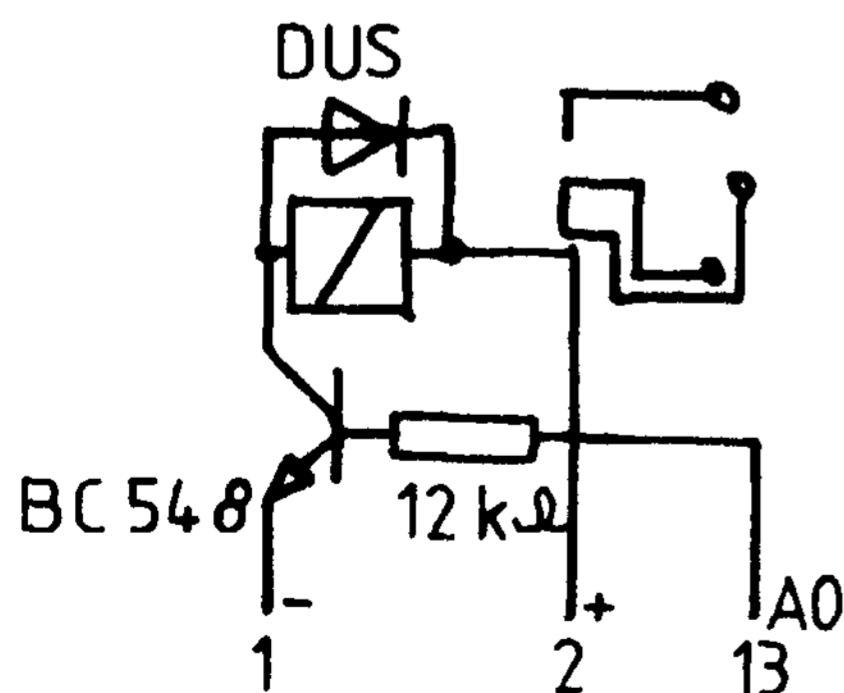
INIT

STO 0

STOC 0

Das Relais zieht nicht an?

Die Diode ist verpolt. Mit Hilfe des Voltmeters wird an der Basis des Transistors (mittleres Bein) nach dem Programmschritt 5 V gemessen. Ist dies der Fall, muß der Transistor ausgewechselt werden.



Die Pinbelegung der 21-poligen Steckerleiste:

Pin 1: Masse
 Pin 2: 5 V
 Pin 3: A 19 (=A7 des WDR-1-Bit-Computers)
 Pin 4: A 18 (=A6 des WDR-1-Bit-Computers)
 Pin 5: A 17 (=A5 des WDR-1-Bit-Computers)
 Pin 6: A 16 (=Pin 17 des ICs)
 Pin 7: A 15 (=Pin 16 des ICs)
 Pin 8: A 14 (=Pin 15 des ICs)
 Pin 9: A 13 (=Pin 14 des ICs)
 Pin 10: A 12 (=Pin 13 des ICs)
 Pin 11: A 11 (=Pin 11 des ICs)
 Pin 12: A 10 (=Pin 10 des ICs)
 Pin 13: A 9 (=Pin 9 des ICs)
 Pin 14: A 8 (=Pin 8 des ICs)
 Pin 15: A 7 (=Pin 7 des ICs)
 Pin 16: A 6 (=Pin 6 des ICs)
 Pin 17: A 5 (=Pin 5 des ICs)
 Pin 18: A 4 (=Pin 4 des ICs)
 Pin 19: A 3 (=Pin 3 des ICs)
 Pin 20: A 2 (=Pin 2 des ICs)
 Pin 21: A 1 (=Pin 1 des ICs)

Jeder der Ausgänge kann nun nach folgendem Muster mit Hilfe eines Voltmeters überprüft werden:

****Ansprechen des Ausganges 1****
 INIT
 STO 4

****Ansprechen des Ausganges 2****
 INIT
 STO 0
 STO 4

****Ansprechen des Ausganges 3****
 INIT
 STO 1
 STO 4

****Ansprechen des Ausganges 4****
 INIT
 STO 0
 STO 1
 STO 4

****Ansprechen des Ausganges 5****
 INIT
 STO 2
 STO 4

****Ansprechen des Ausganges 6****
 INIT
 STO 0
 STO 2
 STO 4

180

****Ansprechen des Ausgangs 7****

INIT
STO 1
STO 2
STO 4

****Ansprechen des Ausgangs 8****

INIT
STO 0
STO 1
STO 2
STO 4

****Ansprechen des Ausgangs 9****

INIT
STO 3
STO 4

****Ansprechen des Ausgangs 10****

INIT
STO 0
STO 3
STO 4

****Ansprechen des Ausgangs 11****

INIT
STO 1
STO 3
STO 4

****Ansprechen des Ausgangs 12****

INIT
STO 0
STO 1
STO 3
STO 4

****Ansprechen des Ausgangs 13****

INIT
STO 2
STO 3
STO 4

****Ansprechen des Ausgangs 14****

INIT
STO 0
STO 2
STO 3
STO 4

****Ansprechen des Ausgangs 15****

INIT
STO 1
STO 2
STO 3
STO 4

****Ansprechen des Ausgangs 16****

INIT
STO 0
STO 1
STO 2
STO 3
STO 4

****Ansprechen des Ausgangs 17****

INIT
STO 5

****Ansprechen des Ausgangs 18****

INIT
STO 6

****Ansprechen des Ausgangs 19****

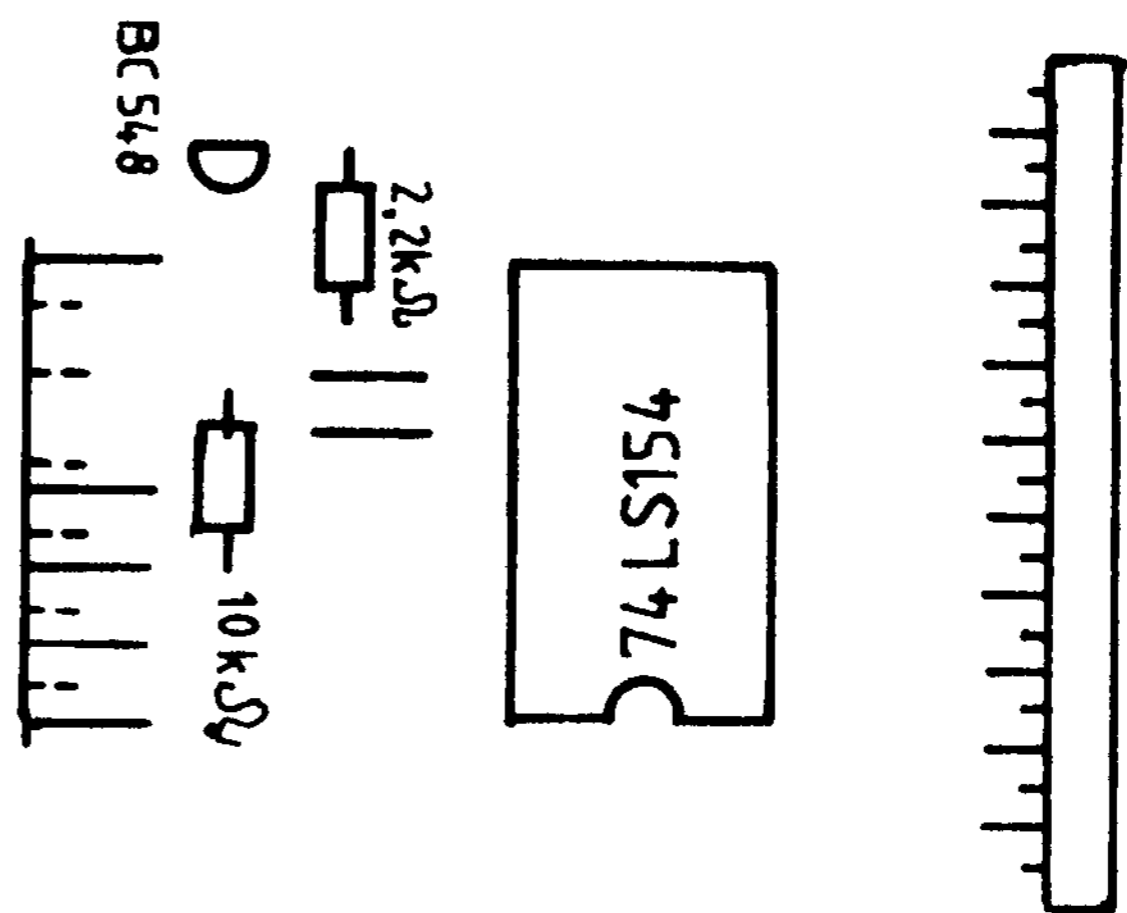
INIT
STO 7

****Ansprechen von zwei Ausgängen (Beispiel A1 und A17)****

INIT
STO 4
STO 5

****Ansprechen von vier Ausgängen (Beispiel A16, A17, A18, A19)****

INIT
STO 0
STO 1
STO 2
STO 3
STO 4
STO 5
STO 6
STO 7



DATA NORF 12

DATANorf 13: ANALOG-DIGITALWANDLER

Eigenschaften:

Die Stromversorgung des Wandlers geschieht mit Hilfe des WDR-1-Bit-Computers, wenn die DATANorf 14 Eingangserweiterung benutzt wird. Der Analog-Digitalwandler hat 8 Stufen und ist aus didaktisch-methodischen Gründen ohne IC aufgebaut. Eine analoge Spannung am Eingang wird in Stufen von ca. 0,7 V in binäre Signale unterteilt. Jeder Schritt wird mit Hilfe einer LED angezeigt. Zum Einlesen des Binärwertes in den WDR-1-Bit-Computers ist die Eingangserweiterung DATANorf 14 nötig.

Aufbau:

Zunächst wird die Unterseite der 21-polige Buchsenleiste auf die Platine an der dafür vorgesehenen Stelle gelötet. Die Widerstände 1 KiloOhm (braun, schwarz, rot, gold) werden eingelötet. Die abgekniffenen Drahtenden stellen die Verbindung zwischen der Oberseite der Buchsenleiste und der Platine her. Die restlichen Bauteile werden entsprechend dem Bestückungsplan eingelötet.

Inbetriebnahme:

An den Punkten 1-3 des Wandlers wird ein Potentiometer, z. B. 1 KOhm angeschlossen. Der Wandler darf nicht direkt auf den Expansionsstecker des WDR-1-Bit-Computers gesteckt werden. Die Versorgungsspannung kann dem WDR-1-Bit-Computer wie folgt entnommen werden: Masse von Pin 1 des Expansionssteckers, ca. 12 V vom Pluspol des 1000 MikroFarad-Kondensators der Grundplatine. Durch Verändern des Potentiometers kann die Digitalisierung des analogen Signals an den LEDs beobachtet werden.

Es leuchtet keine LED?

Die LEDs sind verpolt? Die Zenerdioden sind verpolt? Es liegt keine Spannung am Kollektor des 2N1613, das ist das Gehäuse? Mit Hilfe des Voltmeters wird am Emitter des Transistors 2N 1613 (das ist das Bein an der Markierung am Gehäuse) gemessen, ob durch das Verändern des Potentiometers eine Veränderung der Spannung hervorgerufen wird. Ist dies nicht der Fall, muß der 2N 1613 ausgewechselt werden. Liegt am Kollektor des 2N 1613 eine Spannung an, so muß der oberste BC 548 ausgewechselt werden. Wenn die LED richtig eingelötet ist, muß die oberste LED nun leuchten.

Es leuchtet nur die oberste LED?

Sind die anderen LEDs oder die Zenerdioden verpolt? Mit Hilfe des Voltmeters wird an der Kathode der Diode, die hinter dem Emitter des 2n 1613 folgt, gemessen, ob bei Änderung des Potentiometers eine Spannungsänderung auftritt. Ist dies nicht der Fall, so muß die Unterbrechung gesucht werden, die Diode umgedreht eingelötet werden oder ersetzt werden.

Die Verwendung des Wandlers in Verbund mit der Eingangserweiterung DATANorf 14 für den WDR-1-Bit-Computer ist mit Hilfe eines Programmes möglich.

Ist die Eingangsspannung = 0 V, so sind alle Transistoren des Wandlers gesperrt, die Ausgänge des Wandlers sind alle "1", die Eingangs-LEDs E1 - E3 sind "1", E4 = "0" Um diesen Zustand zu erfassen, müssen die vier Eingänge gelesen werden:

****Einlesen der Zustände des Analog/Digitalwandlers****

Schnelltakt

```

INIT
LD 1
AND 2
AND 3
AND 4
ANDC 4
STO 0
JMP x
    
```

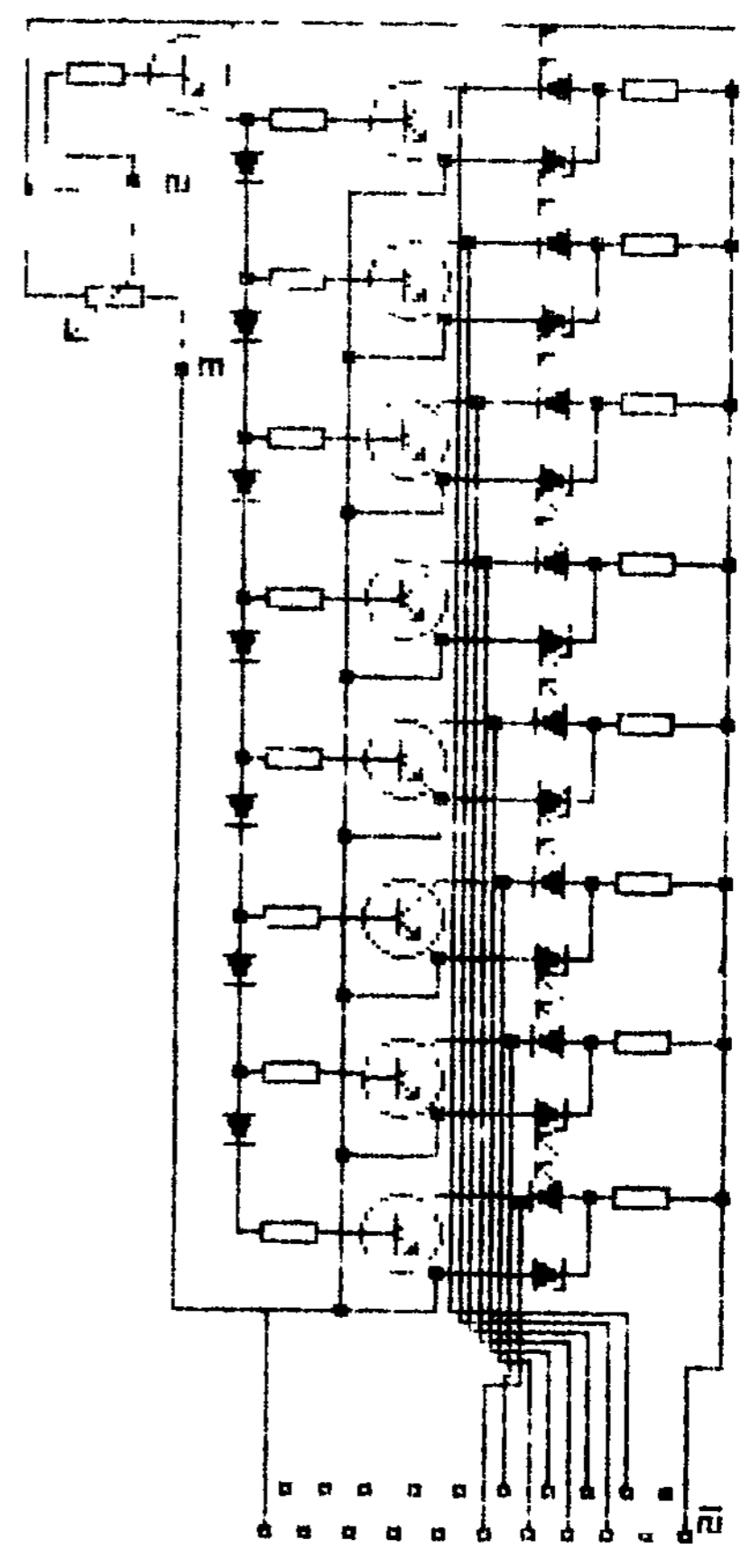
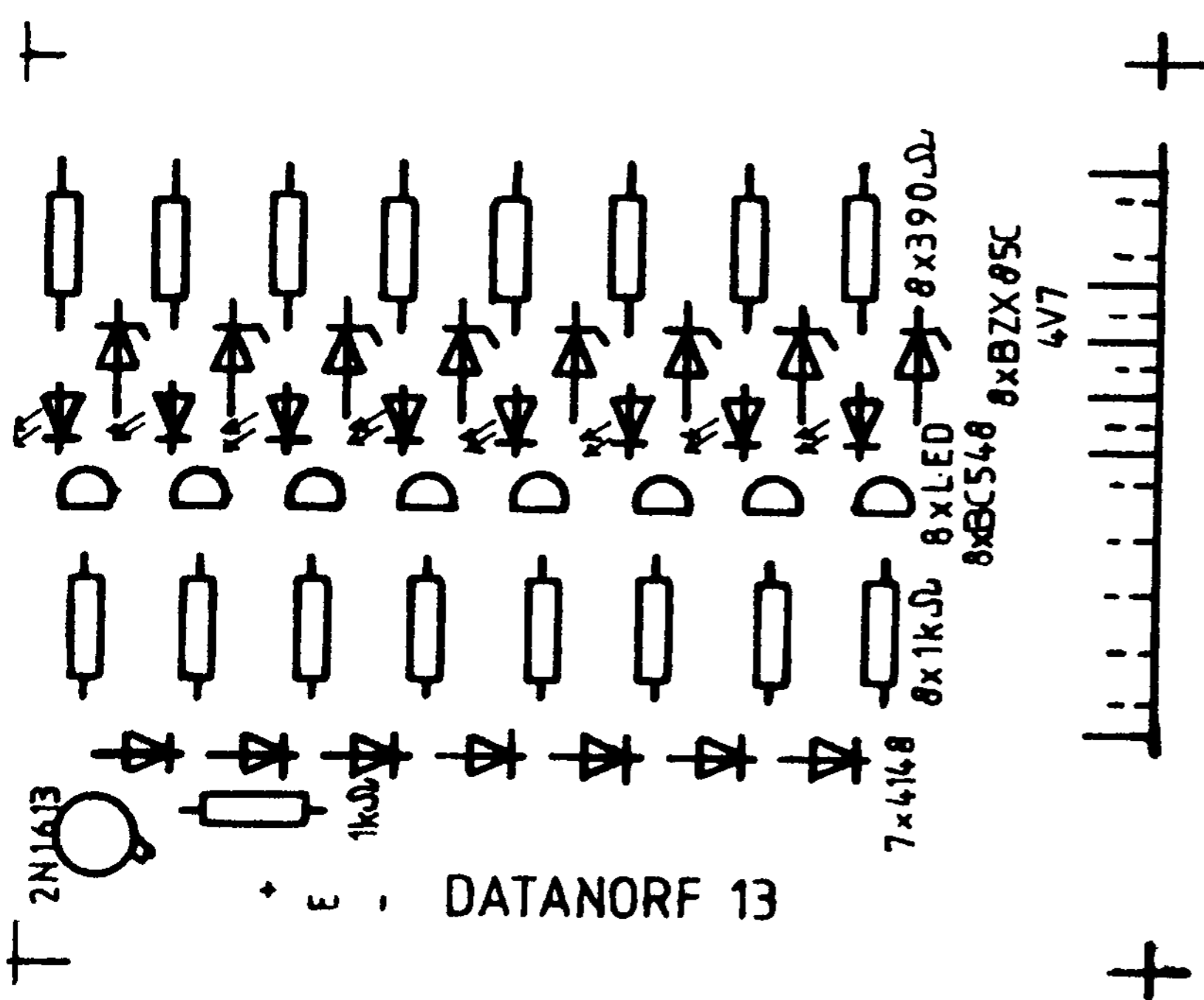
Steigt die Eingangsspannung auf ca. =0,7 V, so öffnet der 1. Transistor, nun sind alle Engänge "1". Das obige Programm wird nach STO 0 um folgende Programmschritte erweitert:

```

LD 1
AND 2
AND 3
AND 4
STO 1
    
```

und wieder mit JMP abgeschlossen.

Entsprechend diesen Beispielen können die anderen analogen Zustände in den WDR-1-Bit-Computer eingelesen werden.



Mit dem Aufstecken des Interfaces leuchten somit die Eingangs-LEDs außer E4 des WDR-1-Bit Computers. Mit Hilfe einer Verbindung zur Masse wird nun I7 an Pin 11 der Steckerleiste des Interfaces aktiviert. Als Folge davon erlöschen die Eingangs-LEDs E1, E2, E3 während E4 leuchtet.

Nun wird die Verbindung zu I6 hergestellt. Es leuchten jetzt die Eingangs-LEDs E4 und E1.

Nun wird die Verbindung zu I5 hergestellt. Es leuchten jetzt die Eingangs-LEDs E4 und E2.

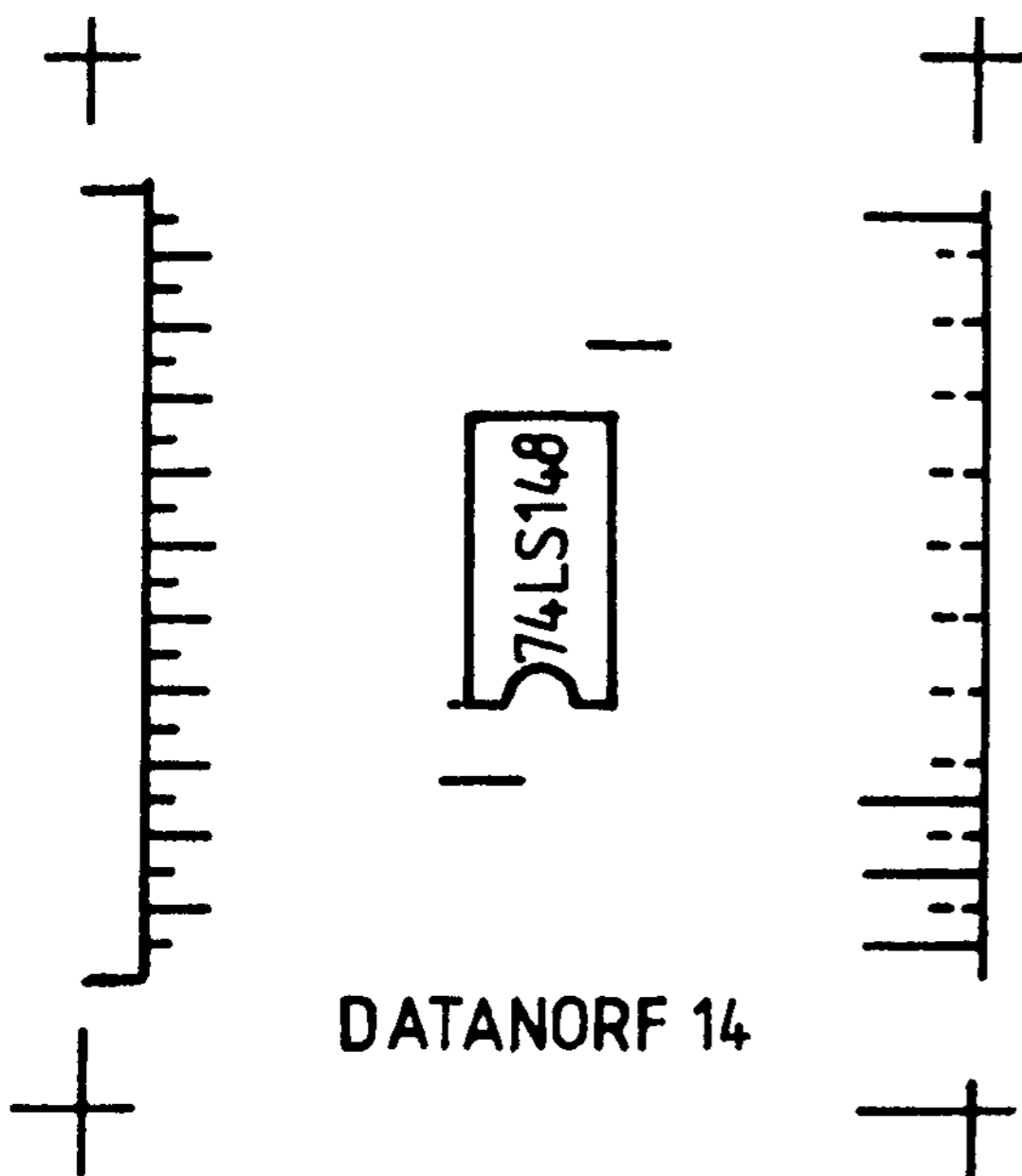
Nun wird die Verbindung zu I4 hergestellt. Es leuchten jetzt die Eingangs-LEDs E4, E2 und E1.


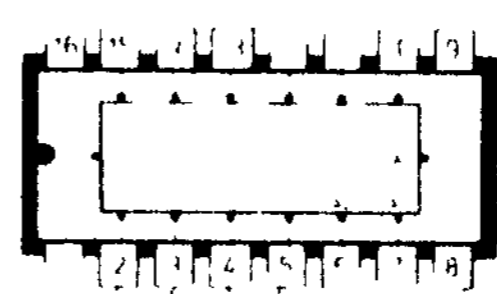
Nun wird die Verbindung zu I3 hergestellt. Es leuchten jetzt die Eingangs-LEDs E4 und E3.

Nun wird die Verbindung zu I2 hergestellt. Es leuchten jetzt die Eingangs-LEDs E4, E3 und E1.

Nun wird die Verbindung zu I1 hergestellt. Es leuchten jetzt die Eingangs-LEDs E4, E3, und E2.

Zuletzt wird die Verbindung zu I0 hergestellt. Es leuchten nun alle vier Eingangs-LEDs.



	Standard	Low Power Schottky	Schottky	Low Power	High Speed
Eingangskapazitäten	ja	ja			
Typ. Impulsverzögerungszeit	10 ns	15 ns			
Typ. Leistungsaufnahme	190 mW	80 mW			

Wahrheitstabelle

E1	Inputs							Outputs				
	0	1	2	3	4	5	6	7	A2	A1	A0	GSE0
H	X	X	X	X	X	X	X	X	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L
L	X	X	X	X	X	X	L	H	L	H	L	L
L	X	X	X	X	L	H	H	L	L	H	L	L
L	X	X	X	L	H	H	H	L	L	H	L	L
L	X	X	L	H	H	H	H	L	L	L	L	L
L	X	L	H	H	H	H	H	L	L	L	L	L
L	L	H	H	H	H	H	H	L	L	L	L	L

Positive Logik

Meßpunkt	Zustand	Lastfaktoren			
Eingänge	0	1,0	1,0		
	alle anderen	2,0	2,0		
Ausgänge	L	10,0	20,0		
	H	20,0	20,0		

Binärer 8 zu 3 Prioritätsenkoder

SN74148	SN74LS148
SN54148	SN54LS148
SN84148	SN84LS148

DATANORF 19: PROGRAMMIERBARER ADRESSENUNTSCHALTER FÜR DIE RAM/EPROM-PLATINE DATANorf 7

1. Aufbau der Platine:

Bis auf den 8-poligen Sockel werden alle Bauteile auf der Bestückungsseite eingelötet.

Der 8-polige Sockel dient als Stecker für den Einsatz auf der RAM/EPROM-Platine anstelle des 4-fach DIL-Schalters. Vor dem Anlöten des 8-poligen Sockels auf der DATANorf 19-Platine wird dieser auf der Oberseite von den überstehenden Stegen befreit und verzinnt. Auch die entsprechende Stelle auf der Lötseite der Platine wird verzinnt. Danach werden die verzinnten Teile des Sockels mit den verzinnten Teilen der Platinenlötseite mit Hilfe des LötKolbens verbunden. Die beiden Kabel werden bei "E" und "I" angelötet, die beiden anderen Enden werden an die beigefügten Stecker angelötet.

22

2. Ergänzungen auf der Prozessorplatine:

2.1 Die Durchkontaktierung unter Pin 8 des IC 14500 durch einen der drei mitgelieferten Stifte für die beiden Stecker ersetzen. Dies ist die Masseverbindung für die DATANorf 19-Platine.

2.2 Über Pin 9 des IC 14500 (ohne die neben Pin 9 auf der Platinenoberseite verlaufende Leiterbahn zu verletzen) wird eine Bohrung für den 2. Stift angefertigt. Der Stift wird an der Unterseite umgebogen und mit Pin 9 des IC 14500, dies ist der Flag F-Ausgang, verbunden.

2.3 Die Leiterbahn von Pin 1 des IC 14500 zum Pin2 des IC 14599 hat zwei Durchkontaktierungen. Eine der Durchkontaktierungen durch den 3. Stift ersetzen. Dieser Stift ist mit dem Reset-Taster der Grundplatine verbunden.

3. Inbetriebnahme der DATANorf 19-Platine:

Entfernen des 4-fach DIL-Schalters. Aufsetzen der Platine auf den nun freien Sockel, so daß die Platine teilweise das EPROM verdeckt.

Verbinden des Masseanschlusses, der Resetleitung und des Eingangs "F" mit dem FLAG F mit Hilfe der drei Stecker.

4. Funktionsweise des Adreßumschalters:

Mit Reset wird der Adreßwahlschalter so geschaltet, daß seine drei Ausgänge auf "0" liegen. Jetzt wird der Bereich 0000 - 00FF angesprochen.

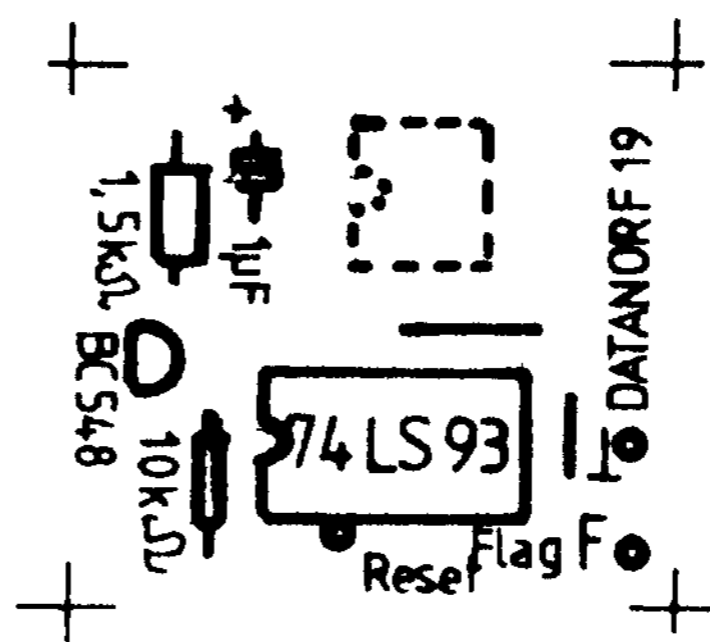
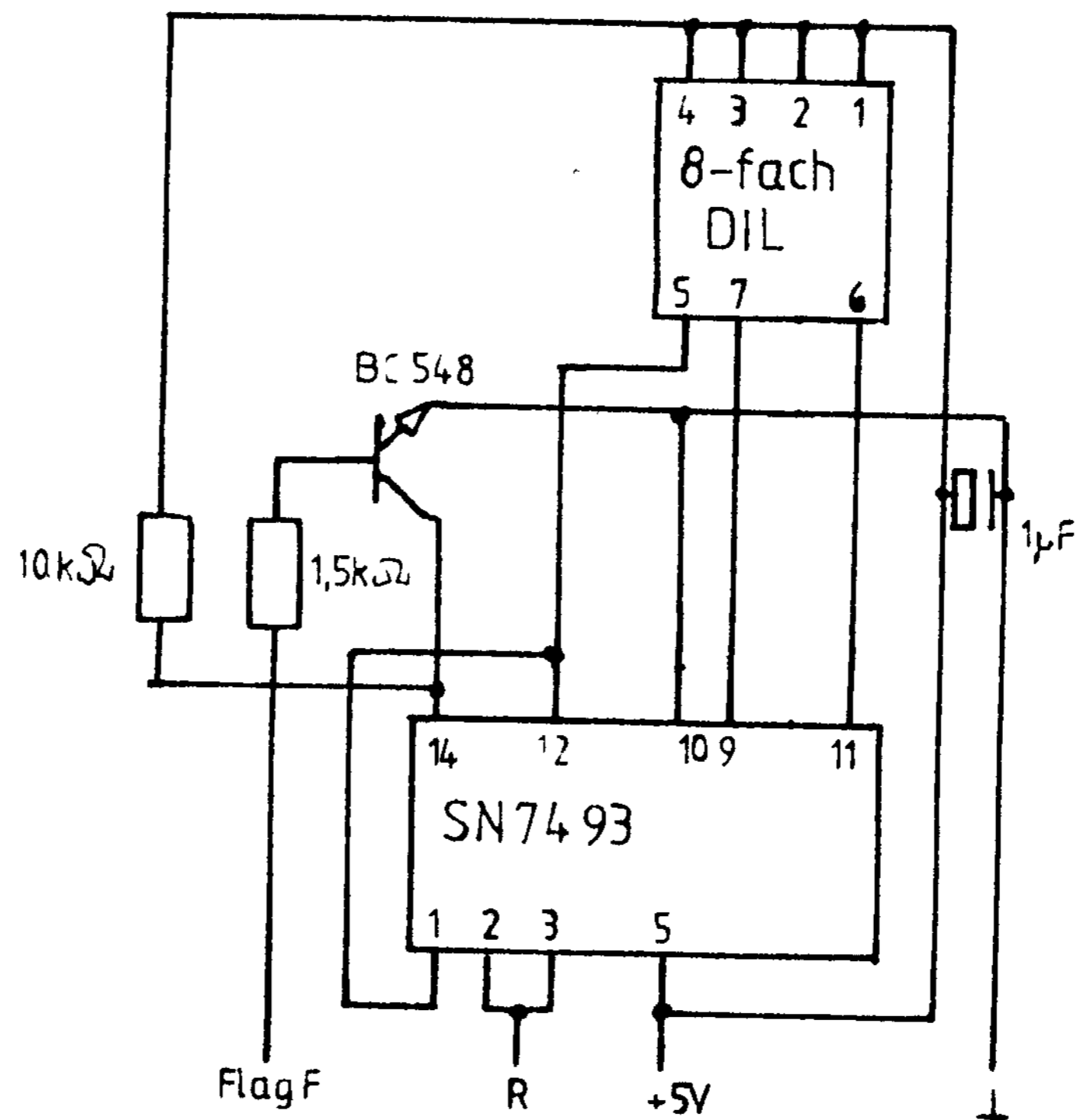
Durch Einfügen eines "F" in eine der Speicherstellen dieses Bereichs wird beim Ablauf des Programms hierdurch der Flag F-Ausgang der CPU angesprochen, der 4-Bit-Binärzähler wird hierdurch um 1 erhöht. Dadurch wird auf den Bereich 0100-01FF umgeschaltet. Usw.

5. Programmierung der Bereichsumschaltung:

Keine der Tasten der Eingabetastatur drücken. Dadurch entsteht das Bitmuster "FF" (hexadezimal). Nun den Schreibtaster betätigen.

Es empfiehlt sich, diesen Befehl vor einem JMP-Befehl zu setzen, sonst kann diese Stelle im Programm nicht erreicht werden. Steht

jedoch vor dem JMP-Befehl ein SKZ-Befehl, kann man in Abhängigkeit von der Eingangsbedingung beim Programmablauf im Speicherbereich bleiben bzw. in den nächsten Bereich umschalten.



DATANORF 22: Steuerung von zwei Schrittmotoren

Aufbau:

1. Bohren mit 1mm Bohrer.
2. Bohren mit 5mm Bohrer für die beiden GummifüÙe.
3. Einlöten der IC- Fassungen, der Kondensatoren ,der 1 mm Stifte ,der 21 poligen Steckerleiste (nicht der Buchsenleiste) und der Widerstände.
Einige Anschlüsse müssen auch auf der Platinenoberseite (Platinenoberseite) angelötet werden.
4. Die abgekniffenen Drahtenden als Durchkontaktierungen verwenden und verlöten.
5. Anlöten der 10 geradzahligen Anschlüsse der 21- poligen Buchsenleiste auf der Platinenunterseite. Herstellen der Verbindungen der ungeradzahligen Anschlüsse der 21- poligen Buchsenleiste mit der Platine mit Hilfe von Drahtenden.
6. Einsetzen der ICs .
7. Anlöten der Steckerchen für die 1mm Stifte an die Schrittmotoren (12 Volt, ca 350 mA). Die Verwendung der Steckerchen wird empfohlen, damit bei Bedarf ein Leistungsteil DATANORF 220 auf die Platine DATANORF 22 aufgesteckt werden kann.

Inbetriebnahme mit Hilfe des WDR-1-Bit-Computers:

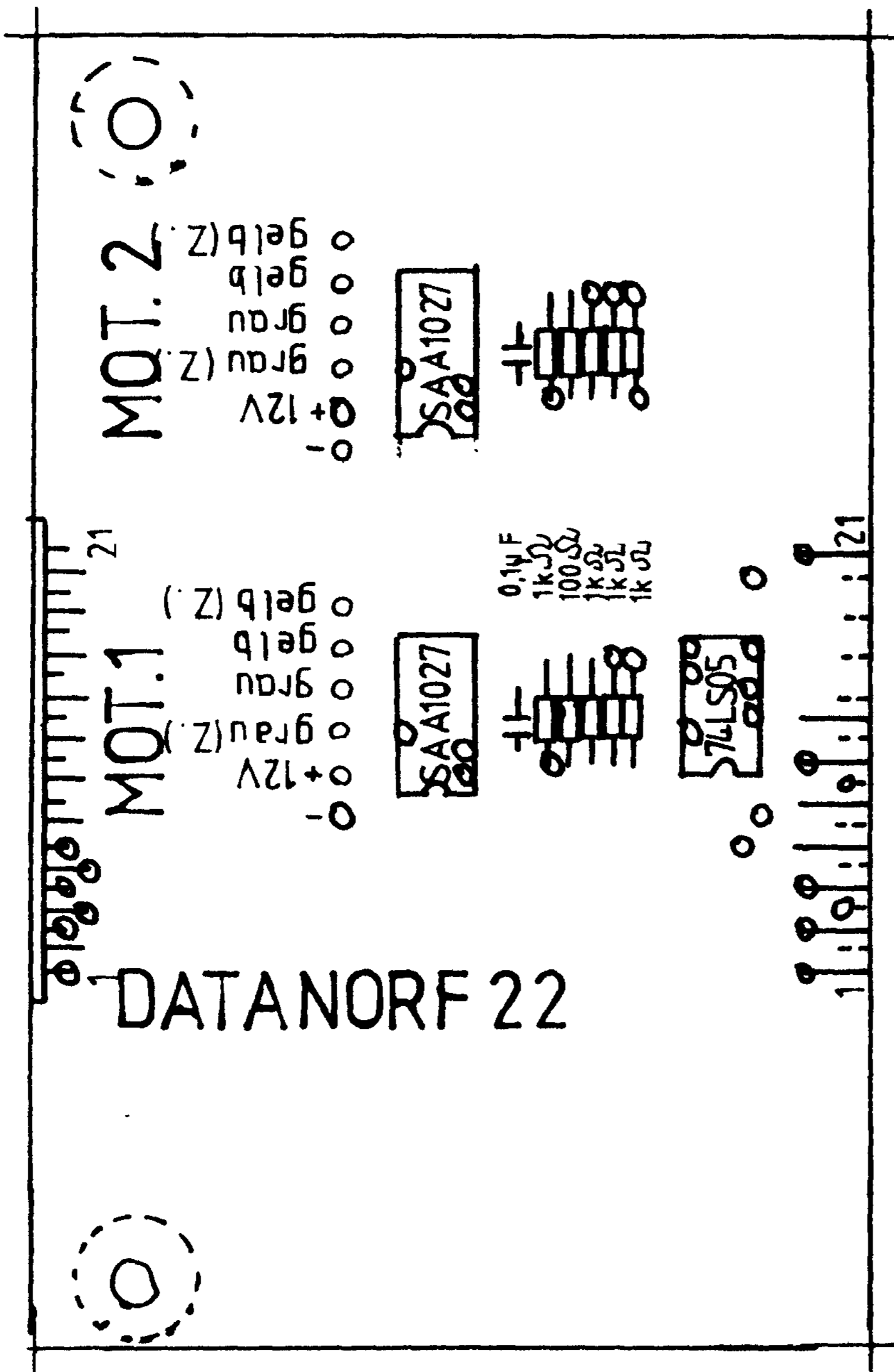
Das Programm INIT
 STO 3
 STOC 3
 JMP läßt den Schrittmotor 1 nach rechts drehen.

Mit einem zusätzlichen STO 4 ändert sich die Drehrichtung. Ein STO 5 stoppt den Motor.

Das Programm INIT
 STO 0
 STOC 0
 JMP läßt den Schrittmotor 2 nach rechts drehen.

Mit einem zusätzlichen STO 1 ändert sich die Drehrichtung. Ein STO 2 stoppt den Motor.

Am 21-poligen Stecker können die 4 Eingänge sowie die Ausgänge A6 und A7 benutzt werden.



LITERATURHINWEISE

- (1) Burkhard John, Volker Ludwig, Ein 1-Bit-Lerncomputer zum Einstieg in die Mikrom Computertechnik, cq-DL 6,7,8,9/1984 DARC-Verlag Baunatal, Postfach 1155, 3507 Baunatal 1
- (2) Ulrike Ludwig, Bit und Byte - Wir bauen einen Computer: Informationen für Lehrer, Praxis Schulfernsehen, Nr. 109/110 und Nr. 111, 1985, Verlagsgesellschaft Schulfernsehen (VGS), Köln
- (3) V. Ludwig, Der WDR-1-Bit-Computer in der Sek. I - Erfahrungen beim Aufbau, der Programmierung und Anwendung zur Steuerung eines Bargeldautomaten, NiU Heft 10 1985, Themenheft "Computer im Physikunterricht, Friedrich Verlag in Velber in Zusammenarbeit mit Klett
- (4) B. John, V. Ludwig, Der 1-Bit-Lerncomputer im Unterricht der Sekundarstufe I, Lehrer Journal 11, 1984, S. 465ff., Oldenbourg/Prögel
- (5) V. Ludwig, Computer im Schulfernsehen, micro computer colleg, 4,1986, Bertelsmann Fachzeitschriften GmbH, München
- (6) Der KM des Landes NRW als Herausgeber, Für den Anfang ein Bit, in "Leben und Lernen in der Hauptschule", S. 51ff., 1984, Völklinger Str. 49, 4000 Düsseldorf
- (7) Alexander Rimmelmann, Der kleine Bruder: Der WDR-1-Bit-Computer, Praxis Schulfernsehen Nr. 105, 1985, S. 4f., Verlagsgesellschaft Schulfernsehen, Köln
- (8) Gregory, Dellande, MC 14500B Industrial Control Unit Handbuch, Motorola 1977
- (9) Baumann, 6502 simuliert 14500, mc 10/82, S. 54ff., Franzis-Verlag, München
- (10) Elektor 6/79, S. 56ff., Elektor Verlag Gangel
- (11) Thompson, "Inside the Micro", Unilab Ltd. 1982, Blackburn UK.
- (12) V. Ludwig, Wir bauen einen Computer, der einen Bargeldautomaten steuert, pc-profi als Beilage zu "Kleiner Wirtschaftsspiegel, Heft 7 und 8, 1985 Sparkassenverlag Stuttgart
- (13) A. Rimmelmann, Bit und Byte - Wir bauen einen Computer, Teil I SchulPraxis plus SchulComputer, Nr. 2-3 1985
- (14) V. Ludwig, A. Rimmelmann, Bit und Byte - Wir bauen einen Computer, Teil II - III, SchulPraxis und SchulComputer, Nr. 2-4 und 2-5 1985
- (15) Heinz Schepers, Hermann-Josef Terplane, "Aufbau und Untersuchungen an einem 1-Bit-Computer mit dem IC MC 14500", Diplomarbeit FH Münster
- (16) V. Ludwig, Computer selbst gebaut in "Arbeiten und Lernen", Heft 47 mit dem Titel Informationstechnische Grundbildung, 1986

BEZUGSQUELLEN

Bausatz WDR-1-Bit-Computer:

~~reffel electronics, Gethaer Str. 16, 4030 Ratingen, Preis 256,00 DM
incl. MwSt.~~

Tastatur und Peripherie:

DATANorf 1:	Bausatz Tastatur mit Anschlußkabel und Stecker	79.70 DM
DATANorf 2:	Bausatz Interface für Motorsteuerungen der Roboter- und Maschinenmodelle des Fischertechnik-Computing-Systems	100.00 DM
DATANorf 3:	Bausatz Peripheriepuffer mit Anschlußstecker (u.a. für den Betrieb der Ampelanlage)	34.00 DM
DATANorf 4:	Bausatz 7-Segmentsteuerung mit Anschlußstecker	41.00 DM
DATANorf 5:	Bausatz Motorsteuerung mit Anschlußstecker	30,00 DM
DATANorf 6:	Bausatz Schrittmotorsteuerung mit Anschlußstecker	34.00 DM
DATANorf 7:	Bausatz RAM/EPROM-Platine mit EPROM für Steuerungen der Fischertechnik-Computing-Modelle, Aufzug, Ampelanlage sowie Musikbox	62.00 DM
DATANorf 8:	Bausatz Morsezeichengenerator mit Anschlußstecker	28.00 DM
DATANorf 9:	Bausatz Musikbox mit Anschlußstecker	46.00 DM
DATANorf 10:	Bausatz Adapter zum Anschluß der WDR-1-Bit-Computerperipherie von DATANorf an C 64	27.00 DM
DATANorf 11:	Bausatz Relaisplatine mit Anschlußstecker	26.00 DM
DATANorf 12:	Bausatz Ausgangserweiterung (8 auf 20) mit Anschlußstecker	34.00 DM
DATANorf 13:	Bausatz Analog-Digital-Wandler mit Anschlußstecker	33.00 DM
DATANorf 14:	Bausatz Eingangserweiterung (4 auf 8) mit Anschlußstecker	26.00 DM
DATANorf 15:	Diskette mit Simulationsprogramm "WDR-1-Bit-Computer" (Commodore C64/C128, Apple, andere auf Anfrage)	45.00 DM
DATANorf 17:	EPROM mit 8 Discolightprogrammen	20.00 DM
DATANorf 18:	EPROM mit Programm "Löcherstrom beim Transistor"	20.00 DM

DATANorf 19:	Bausatz Programmierbarer Adressenumschalter für die RAM/EPROM/Platine	19.00 DM
DATANorf 20:	10 Fischertechnikbausteine für eine digitale Abfrage von Modellen aus fischertechnik-Computing-Baukasten	8.00 DM
DATANorf 21:	Bausatz Adapter zum Anschluß des WDR-1-Bit-Computers an das ELSA-Didakt-System	23.00 DM
DATANorf 22:	Bausatz Steuerung von zwei Schrittmotoren incl. Anschlußstecker	49.00 DM
DATANorf 23:	Bausatz Steuerung für einen Schrittmotor incl. Anschlußstecker	34.00 DM
DATANorf 30:	Bausatz Adapter wie DATANorf 10 für den Apple	12.00 DM
DATANorf 31:	Bausatz EPROM-Brenner zum WDR-1-Bit-Computer für die RAM/EPROM-Platine	

Alle Bausätze von DATANorf sind mit Schaltungsunterlagen bzw. Hinweisen für den Betrieb versehen. Die Preise sind ohne Porto und Verpackung. Lieferung erfolgt nach Zahlungseingang auf DATANorf Postscheckamt Essen, Kontonr.301832-431 Am alten Bach 14a, 4040 Neuss 21