Franzis Computer-Praxis

Klein

Rechner modular

Der NDR-Klein-Computer –
selbstgebaut und programmiert

Bus-Schaltkreise
Z80-CPU voll ausgebaut
Die CPU 64180
64-KByte-Speicher
Blumen mit Schleife
Serielles Interface
Der Floppy-Anschluß
Aufbau eines EPROM-Programmierers
Sound-Generator
16-Kanal-Analog/Digital-Umsetzer
D/A-Umsetzer
Assembler
Strukturierte Programmierung
Zeilenassembler und Disassembler
Gosi
Basic

Franzis'

Vorwort

Die Mikrocomputer dringen in alle Bereiche des Lebens ein. Daher kann es nur von Nutzen sein, sich mit dieser Technik vertraut zu machen. Für den Anfänger ist es aber schwierig, einen Einstieg zu finden. Es gibt eine Vielzahl fertiger Computer, die aber meist nur wenig durchschaubar sind. Schaltpläne sind nur selten für den Anwender erhältlich. Außerdem verwenden die Hersteller oft eigene integrierte Bausteine, deren Innenleben ein streng gehütetes Geheimnis bleibt. Das gleiche gilt für die Software.

In dem vorliegenden Buch wird ein Mikrocomputersystem vorgestellt, das dieses Übel beseitigen soll. Alle Mikrorechnerschaltungen sind durch im Handel erhältliche Leiterplatten unterstützt. Sogar eine Fernsehreihe existiert, die dieses System verwendet. Der hier vorgestellte Mikrorechner verdankt seinen Namen dieser Serie und heißt NDR-Klein-Computer.

Der Computer ist modular aufgebaut, das heißt, durch Kombination von verschiedenen Baugruppen kann man den Computer ganz nach Bedarf ausbauen. Das geht vom einfachen Z80-Rechner mit 4 KByte über einen voll ausgebauten Z80-Computer mit 1 MByte sogar bis zu einem 68020-Computer mit 4 MByte und Winchester. In diesem Buch wird der Aufbau des Z80-Computers behandelt, der am Schluß mit Floppy-Laufwerken ausgestattet werden kann und damit keinen Vergleich mit anderen kommerziellen Systemen zu scheuen braucht. Durch die Verwendung des CP/M-Betriebsystems ist auch eine Vielfalt an kompatibler und preisgünstiger Software verfügbar.

Besonderer Dank gilt auch Herrn Dr. Hans Hehl für die Durchsicht des Manuskripts. Durch das Sammeln seiner Erfahrungen beim Einsatz des NDR-Klein-Computers im Gymnasium Markt-Schwaben hat er wertvolle Beiträge zu diesem Buch geleistet.

Rolf-Dieter Klein, München

2 Kurze Einführung in die Digitaltechnik

Die Digitaltechnik vollständig abzuhandeln, würde den Rahmen des Buches sicher sprengen. Dennoch soll hier versucht werden, wenigstens einen kurzen Überblick zu vermitteln.

Im Jahr 1941 wurde in Deutschland die erste programmgesteuerte, elektronische Rechenanlage in Betrieb genommen. Das Gerät bestand aus 2600 höchst sinnvoll verschalteten Relais. Erdacht hatte diesen ersten Computer der Welt, der auch wirklich vollständig funktionierte, ein Mann namens Konrad Zuse. Er wollte damit die langwierigen und auch fehlerträchtigen Berechnungen im Bauingenieurwesen und auch anderswo automatisieren und damit sicherer und schneller durchführbar machen. Der Computer trug den Namen Z3. Er ist im deutschen Museum in München zu besichtigen.

2.1 Digitale Signale (Dr. Hans Hehl)

Die Z3 von Konrad Zuse war ein erstaunliches Gerät. Denn sie konnte neben den Grundrechenarten zum Beispiel auch Wurzeln ziehen. Das ist deshalb so erstaunlich, weil die Z3 dazu (natürlich in elementarer Form) alle Merkmale programmgesteuerter Universalrechenmachinen aufweisen mußte, wie sie auch heute noch gültig sind. Erfunden und in die Maschine eingebaut hat das alles im wesentlichen ein einziger Mann, eben Konrad Zuse, der dafür in den Jahren nach 1970 auch vielfältig geehrt wurde. Eine seiner wichtigen Ideen beim Bau der Z3 war die Verwendung eines Zahlensystemes, das besser an Maschinen angepaßt ist als unser gewöhnliches Zehnersystem. Intern rechnete die Z3 mit den sogenannten Dualzahlen. Weshalb das so gut war, soll gleich erklärt werden: Es hängt mit der Verwendung von Relais zusammen.

Ein Relais ist ja nichts weiter als ein elektrischer Schalter, der mit elektrischem Strom ein- und ausgeschaltet werden kann. Bei einem Relais macht also der Strom das, was man bei einer Taschenlampe mit dem Daumen von Hand machen muß: Man schaltet den Schalter der Lampe ein oder aus. Entsprechend wird die Lampe leuchten oder nicht. Eine Verbindung zu den Zahlen kann man schlagen, wenn man verabredet, daß zum Beispiel der Zustand der Taschenlampe Null sein soll, wenn sie ausgeschaltet ist und Eins, wenn sie eingeschaltet ist. So merkwürdig künstlich und willkürlich so eine Verabredung zunächst erscheint, die ganze Computerindustrie ist in gewissem Sinn darauf aufgebaut.

Ein Relais, eine Taschenlampe, überhaupt ein physikalisches Gerät, das zwei Zustände annehmen kann, von welchen man verabreden kann, daß der eine Zustand Null, der andere Eins bedeuten soll, das sind Beispiele für die Realisierung einer sogenannten binären Variablen.

Einer Taschenlampe sieht man nicht an, mit welcher Spannung die Glühbirne betrieben wird. Allerdings weiß man, daß sicher nicht so hohe Spannungen wie z. B. 220 V verwendet werden. Genauso sind die Spannungswerte (Pegel) bei einer elektrisch dargestellten binären Variablen (ein- oder ausgeschaltet) prinzipiell nicht vorgegeben. Sie hängen jeweils von der technischen Konzeption des Gerätes ab. Dem Binärwert 0 kann man zum Beispiel die Spannung 0 V ebenso zuordnen wie die Spannung − 12 V. Allgemein spricht man von einem L-Pegel (Low), wenn der Pegelwert näher bei „minus unendlich“ liegt und von einem H-Pegel (High), wenn der Pegelwert näher bei „plus unendlich" liegt. In der Praxis wird meist dem L-Pegel der Wert 0 der binären Variable und dem H-Pegel der Wert 1 zugeordnet (positive Logik).

Dualzahlen binär dargestellt

Wenn man begreifen will, wie Zahlen in einem Computer dargestellt werden, dann ist es zum Beispiel günstig, sich den Kilometerzähler eines Autos vorzustellen. Es sei ein Modell, das keine Hundert-Meter-Einteilung besitzt. Dann wird dort beim Fahren das Anzeigerad für die einzelnen Kilometer, also das Rad ganz rechts, von einer mit den Auto-Rädern verbundenen Welle gedreht und zeigt nacheinander 0, 1, 2, 3, 4, . . . Auf diesem Anzeigerad (und auf den anderen weiter links auch) befinden sich zehn Ziffern, die der Reihe nach gezeigt werden. Immer dann, wenn das Einerrad eine Umdrehung vollendet, also beim Umschalten von 9 auf 0, wird das benachbarte weiter links befindliche Rad um eine Ziffer weitergedreht. Stand es vorher auf 0, dann steht es nach einer solchen Situation auf 1. Dazu besitzt das Einerrad einen Mitnehmer, der das Zehnerrad dann mitnimmt. Also nach zehn gefahrenen Kilometern steht tatsächlich auch 10 auf dem Kilometerzähler. Das Zehnerrad zählt also mit, wie oft das Einerrad sich gedreht hat und merkt so an, wievielmal zehn Kilometer zurückgelegt wurden. Das Zehnerrad selbst besitzt ebenfalls einen Mitnehmer, der bei Vollendung einer Umdrehung das benachbarte Hunderterrad um eins weiterdreht. Und dieses Hunderterrad wiederum kann das Tausenderrad mitnehmen, was selbst wieder das Zehntausenderrad mitnimmt. Und so weiter.

Daß die Anzeigeräder jeweils 10 Ziffern tragen, das rührt von unserer Gewohnheit her, im Zehnersystem zu rechnen. Eine ziemlich merkwürdige, aber durchaus mögliche Konstruktion eines solchen Kilometerzählers könnte darin bestehen, daß man auf die Anzeigeräder nur auf der einen Seite des Umfanges 0 und auf der anderen Hälfte 1 anschreibt und den Mitnahmemechanismus so gestaltet, daß beim Drehen von 1 auf 0 das weiter links befindliche Rad um eine halbe Umdrehung weiter gedreht wird. Das Einerrad zählt dann von 0 bis 1. Links daneben befindet sich das Zweierrad, das um eins weiter gedreht wird, wenn das Einerrad einmal ganz herum kommt und dabei der zweite Kilometer abgefahren wird. Nach zwei gefahrenen Kilometern steht dann 10 auf diesem merkwürdigen Zähler. Auch das Zweierrad dreht beim Übergang von 1 auf 0 ein weiter links befindliches Rad. Es sind hier einfach einmal für einen vierstelligen Zähler mit der verrückten Zweiereinteilung die Anzeigestellungen und die gefahrenen Kilometer aufgezählt:

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

An dieser Aufstellung kann man, wenn man scharf hinschaut, erkennen, daß in einer Kolonne von oben nach unten (links vom Gleichheitszeichen) immer nur die Ziffern 0 auf 1 auftauchen. Man könnte also für jede der vier Stellen eine binäre Variable hernehmen und diese vier Variablen dann jeweils so schalten, wie es das Muster aus Nullen und Einsen verlangt, das gerade auf dem Kilometerzähler erscheint. Zum Beispiel leuchtet bei vier nebeneinanderliegenden Taschenlampen genau die ganz rechts außen liegende. Dann kann man sagen, daß damit die Zahl Eins binär dargestellt ist. Wenn alle vier Lampen eingeschaltet sind, dann ist damit die Zahl 15 binär dargestellt. Jeder Kombination von „Ein" und „Aus" entspricht also ganz natürlich eine Zahl.

Mathematiker nennen Gebilde, wie sie in der linken Spalte auftauchen, Dualzahlen, wenn sie betonen wollen, daß sie in einem System arbeiten, das nur die Ziffern 0 und 1 benutzt. Wie eben gesagt, kann man also die Dualzahlen zum Beispiel mit einer geeigneten Anzahl von Taschenlampen binär darstellen. Zuse benutzte in seiner Z3 Relais, um damit Dualzahlen binär in seiner Maschine darzustellen.

Noch etwas Theorie

Ein Zahlensystem mit den beiden Ziffern 0 und 1 unterscheidet sich von unserem gewohnten Zehnersystem durch einen wesentlich kleineren Abstand der Stellenwerte. Was besagt dies aber? Grundsätzlich können wir je nach Art der Anordnung von Zeichen für Zahlen Additionssysteme und Positionssysteme unterscheiden.

Ein Additionssystem ist zum Beispiel das römische Zahlensystem. In ihm wird das Jahr 1768 als MDCCLXVIII dargestellt. Die eigentliche Zahl ergibt sich durch Addition der einzelnen Zahlzeichen. Das heutzutage benützte Positionssystem (auch Stellenwertsystem genannt) wertet dagegen die Stellung des Zahlzeichens mit aus.

Ein Beispiel soll dies verdeutlichen: In dem römischen Zeichen III für die Zahl Drei hat jede der drei Ziffern den Zahlwert 1 und die Zahl ergibt sich durch die Addition der drei einzelnen Zahlenwerte.

Im dekadischen Positionssystem ergibt die Zeichenanordnung 111 die Zahl Einhundertelf. Alle Zeichen haben den gleichen Zahlwert 1, aber einen unterschiedlichen Stellenwert. Der niedrigste

Tabelle 2.1.1 Dezimalzahl und Einschaltkombination

Dezimalzahl	Kombination
1	0 0 0 0 0 0 0 1
2	0 0 0 0 0 0 1 0
3	0 0 0 0 0 0 1 1
4	0 0 0 0 0 1 0 0
5	0 0 0 0 0 1 0 1
6	0 0 0 0 0 1 1 0
7	0 0 0 0 0 1 1 1
8	0 0 0 0 1 0 0 0
9	0 0 0 0 1 0 0 1
10	0 0 0 0 1 0 1 0
.	.
31	0 0 0 1 1 1 1 1
.	.
255	1 1 1 1 1 1 1 1

Stellenwert (Einer) steht ganz rechts, dann folgen Zehner und Hunderter. Jeder Stellenwert beträgt ¹⁄₁₀ des links von ihm stehenden.

Beim Binärsystem sind die Stellenwerte die Potenzen der Zahl (Basis) 2, also die Zahlen 1, 2, 4, 8, 16, 32, 64, . . . usw.

Acht Taschenlampen oder ein Byte?
Wir verwenden nun acht Taschenlampen, die wir in eine Reihe legen und beliebig ein- bzw. ausschalten. Damit ergeben sich $2^8 = 256$ Einschaltkombinationen. Wenn die Zuordnung dieser Kombinationen zu den Zahlen 0 bis 255 mit der Rechenregel „Dezimalzahl ergibt sich durch Aufsummieren der Zweierpotenzen" durchgeführt wird, dann ergibt sich das Schema wie beim Kilometerzähler. In *Tabelle 2.1.1* sind einige Dezimalzahlen und die entsprechenden Kombinationen der Werte 0 und 1 aufgeführt (eingeschaltet = 1, ausgeschaltet = 0), die sich beim Zählen wie vorhin ergeben würden.

Die Zuordnung kann nun überprüft werden. Die Kombination 0 0 0 1 1 1 1 1 ergibt als Summe der Zweierpotenzen die Zahl 31.

$$0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$
$$0 + 0 + 0 + 16 + 8 + 4 + 2 + 1$$

Umgekehrt kann aus einer Dezimalzahl zwischen 0 und 255 die zugehörige Dualzahl ermittelt werden, indem fortlaufend die Zweierpotenzen, beginnend bei 2^7, von der Zahl bzw. vom übrig bleibenden Rest abgezogen werden. Würde die Differenz negativ, so wird eine Null aufgeschrieben und die nächst kleinere Zweierpotenz verwendet. Ist die Differenz positiv, so wird eine 1 aufgeschrieben und mit dem Rest weiter gearbeitet. Probieren wir dies mit der Zahl 18 aus.

Tabelle 2.1.2 Gegenüberstellung der drei Stellenwertsysteme

Dezimal	Sedezimal	Dual
0	0	0000 0000
1	1	0000 0001
2	2	0000 0010
3	3	0000 0011
.	.	.
8	8	0000 1000
9	9	0000 1001
10	A	0000 1010
11	B	0000 1001
12	C	0000 1100
13	D	0000 1101
14	E	0000 1110
15	F	0000 1111
16	10	0001 0000
17	11	0001 0001
.	.	. .
94	5E	0101 1110
.	.	. .
171	AB	1010 1011
.	.	. .
255	FF	1111 1111

18 − 128 = ? geht nicht, also 0
18 − 64 = ? geht nicht, also 0
18 − 32 = ? geht nicht, also 0
18 − 16 = 2 geht, also 1
2 − 8 = ? geht nicht, also 0
2 − 4 = ? geht nicht, also 0
2 − 2 = 0 geht, also 1
0 − 1 = ? geht nicht, also 0

Die Kombination für die Zahl 18 lautet also von oben nach unten: 0 0 0 1 0 0 1 0.

Eine solche Kombination von Nullen und Einsen nennt man „Byte". Dieses Byte besteht aus acht Bits. Ein Bit, abgeleitet von „binary digit", ist die kleinste Darstellungseinheit für Binärdaten. Ein Bit repräsentiert eine Binärstelle in einem Byte.

Für Umwandlungsübungen seien noch einige Beispiele angegeben.

85 = 0 1 0 1 0 1 0 1
138 = 1 0 0 0 1 0 1 0
255 = 1 1 1 1 1 1 1 1

2.2 Der Treiber und Logikschaltungen (Dr. Hans Hehl)

Vom Relais zum Transistor

Genug der Mathematik, nun sei diskutiert, wie das Ein- und Ausschalten der Glühbirnchen unserer acht Taschenlampen automatisiert werden kann. 1941 verwendete K. Zuse dazu Relais. Aber so ein Relais kann nicht Hunderte von Schaltvorgängen pro Sekunde durchführen, die Kontakte sind dazu zu träge. Es ist schon eigenartig, daß nur 7 Jahre später, also 1948 ein Ersatz für das langsame Relais entdeckt wurde. Die Amerikaner Bardeen, Brattain und Shockley entdeckten den Transistoreffekt an einem Germaniumkristall und erhielten dafür 1956 den Nobelpreis für Physik.

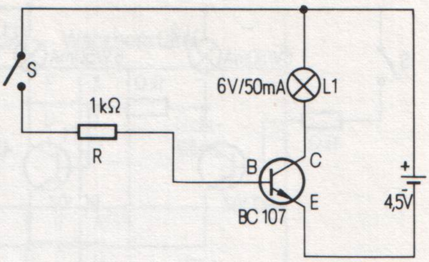
Mit dem Transistor stand ein Bauteil zur Verfügung, das keine mechanischen Teile enthält, sehr schnell schalten kann und weniger Strom als der Elektromagnet eines Relais benötigt. *Abb. 2.2.1* zeigt den Schaltplan eines Transistorschalters.

So eine Schaltung wird zum Beispiel benötigt, wenn ein Computer Lampen, Motoren usw. schalten soll, um also eine Verbindung zur Außenwelt zu schaffen. Bevor wir die Teile der Schaltung näher betrachten, müssen wir uns jedoch dem Problem der Stromrichtung zuwenden.

Man hatte vor der Zeit der Elektronenröhren und der Halbleiter einfach die Stromrichtung vom Pluspol der Spannungsquelle über den Verbraucher zum Minuspol festgesetzt (technische Stromrichtung). Die normalen Träger der Elektrizität, die Elektronen, fließen aber vom Minuspol über den Verbraucher zum Pluspol, wie man erst später entdeckte. Wir verwenden hier diese Elektronenflußrichtung.

Wichtigster Teil der Treiberschaltung nach *Abb. 2.2.1* ist der Transistor (BC-107). Er besteht im Inneren aus drei Halbleiterschichten mit wechselnder Leitfähigkeit. Halbleiter, zum Beispiel Silizium oder Germanium, leiten den Strom schlechter als Metalle. Werden geringste Mengen eines anderen Metalles (z. B. Antimon) hinzugefügt, verändert sich die Leitfähigkeit der Schicht erheblich. Verfolgen wir nun den Elektronenfluß durch den Transistor. Vom Minuspol der Batterie fließen die Elektronen zum Emitter E (durch eine Pfeilspitze gekennzeichnet, die aus dem Kreissymbol heraus zeigt, npn-Typ). Wieviel Elektronen nun zum Collektor C (bzw. Kollektor) und damit durch die Lampe L1 fließen können, hängt vom Stromfluß am Steuereingang (Basis) B ab, der vom Emitter über Basis, Widerstand R, geschlossenen Schalter S zum Pluspol der Batterie fließt. Ein geringer Emitter-Basis-Strom bewirkt einen großen Emitter-Kollektor-Strom, man spricht von einer „Stromverstärkung". Großer Strom bedeutet aber nach dem Ohmschen Gesetz einen kleinen Widerstand der Emitter-Kollektor-Strecke, der Transistor „schaltet durch". Die Stromverstärkung beträgt einige „Hundertfache".

Abb. 2.2.1 Treiberschaltung mit einem Transistor



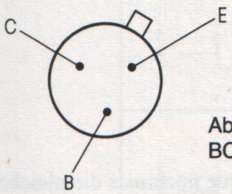


Abb. 2.2.2 Anschlußschema des Transistors BC-107

Nach der Theorie zur Praxis. Verwendet werden kann der Transistor BC-107 oder ein vergleichbarer npn-Typ (wie z. B. BC-109 und andere). Die Kennzeichnung der drei Anschlußdrähte des Transistors ergibt sich aus *Abb. 2.2.2*. Der Draht dicht neben dem Gehäusevorsprung ist der Emitter, wobei man den Transistor von unten, also von der Anschlußdrahtseite her anschaut.

Achtung: wird der Basis-Widerstand R überbrückt, liegt die Spannung der Batterie voll an Emitter und Basis an, ein großer Strom fließt und eine Zuleitung im Transistor schmilzt durch.

Löten: Übung macht den Meister

Wenn Sie die im Buch angegebenen Schaltungen aufbauen wollen, dann verwenden Sie bitte einen kleinen Lötkolben mit etwa 20 W Leistung und eine feine Dauerlötspitze, die nicht verzundert. Reine Kupferspitzen sind weniger geeignet. Die Spitze reinigt man vor jedem Lötvorgang mit einem feuchten Spezialschwämmchen oder mit einem Baumwollappen. Als Lot wird ein 1 mm dünner Lötdraht verwendet, der im Inneren ein Flußmittel auf Harzbasis enthält. Säurehaltige Flußmittel wie Lötfette oder sogar Salzsäure dürfen auf gar keinen Fall verwendet werden, da die Säurereste eine Korrosion der Leiterbahnen und Bauteile bewirken. Zum Löten erwärmen Sie mit der Lötkolbenspitze solange die zu verbindenden Teile, bis das gleichzeitig an die Teile gehaltene Lot schmilzt und die Teile überzieht. Die Lötstelle darf bis zum Erkalten nicht bewegt werden. Eine gute Lötstelle verbindet die Bauteile mit nur wenig Lötzinn, das eine hellglänzende Oberfläche besitzt. Üben Sie dies nicht mit Ihren Bausatzplatinen, sondern an versilberten Schaltdrahtresten oder Kupferlitze. Alles Handwerkszeug und das Lötzinn sollten Sie im Elektronikfachhandel kaufen, damit Sie sicher sind, daß Ihre Arbeitsmittel auch geeignet sind.

Treiber mal zwei

Wir benötigen zu Versuchszwecken wieder unsere Treiberschaltung aus dem ersten Abschnitt, *Abb. 2.2.1*. Die Lampe leuchtete, wenn der Schalter geschlossen wurde.

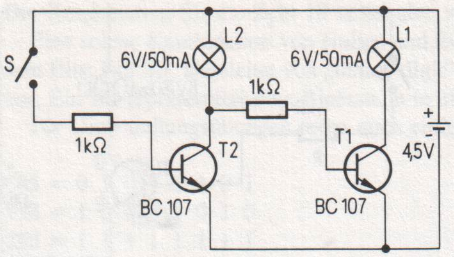


Abb. 2.2.3 NICHT-Glied: Doppelte Treiberschaltung

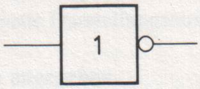


Abb. 2.2.4 Schaltsymbol des NICHT-Gliedes

Diese Treiberschaltung sei jetzt erweitert. Anstelle des Schalters wurde nochmals die gleiche Treiberschaltung eingesetzt. *Abb. 2.2.3* zeigt die neue Schaltung. Nach dem Anschließen der Batterie leuchtet Lampe L1. Wird aber der Schalter S geschlossen, so erlischt Lampe L1 und L2 leuchtet. Warum?

Lampe L1 leuchtet zunächst, da ein kleiner Steuerstrom vom Transistor T1 über Widerstand und Glühfaden von L2 fließen kann, ohne daß L2 leuchtet. Da der Schalter offen ist, kann kein Steuerstrom bei Transistor T2 und damit auch kein Strom von dessen Emitter zum Kollektor fließen. Der Transistor T2 besitzt in diesem Zustand keinen Einfluß auf die Schaltung. Schließen wir aber den Schalter, so bewirkt der entstehende Steuerstrom einen großen Stromfluß durch T2 und die Lampe L2. Wir könnten auch Emitter und Kollektor von T2 mit einem Draht überbrücken, denn großer Stromfluß bedeutet kleinen Widerstand (Spannung konstant). Jetzt fließt kein Steuerstrom mehr durch T1, weil die Spannung am Kollektor von T2 fast ganz auf Null abgesunken ist. L1 erlischt deshalb.

Wir bezeichnen nun den Schalter als Eingang, dessen offenen Schalterzustand mit der Zahl 0 und den geschlossenen mit 1. Die Lampe L1 wird zum Ausgang erklärt. Den Leuchtzustand kennzeichnen wir mit der Zahl 1. Dann erhalten wir folgenden Zusammenhang:

Schalter	Lampe L1
0	1
1	0

Das Eingangssignal erscheint am Ausgang invertiert, also genau umgekehrt. So eine invertierende Schaltung wird als NICHT-Glied bezeichnet. *Abb. 2.2.4* zeigt das Schaltsymbol der invertierenden Schaltung.

Nicht nur NICHT

Ein Gesichtspunkt ist besonders interessant. Man kann das Verhalten einer solchen Schaltung, wie die des Inverters, einerseits am konkreten Objekt studieren und andererseits das Wesentliche daran, daß nämlich ein Null-Zustand an der Eingabe in einen Eins-Zustand an der Ausgabe verwandelt wird und ein Eins-Zustand an der Eingabe in einen Null-Zustand an der Ausgabe, in einer Tabelle ganz kurz und trocken notieren. *Abb. 2.2.5* zeigt einfach an, was welchem Eingabewert an der Ausgabe durch die verwendete Schaltung zugeordnet wird. Der Inverter hatte an seiner Eingabe nur eine binäre Variable, den einen Schalter. Es gibt nun Schaltungen (die

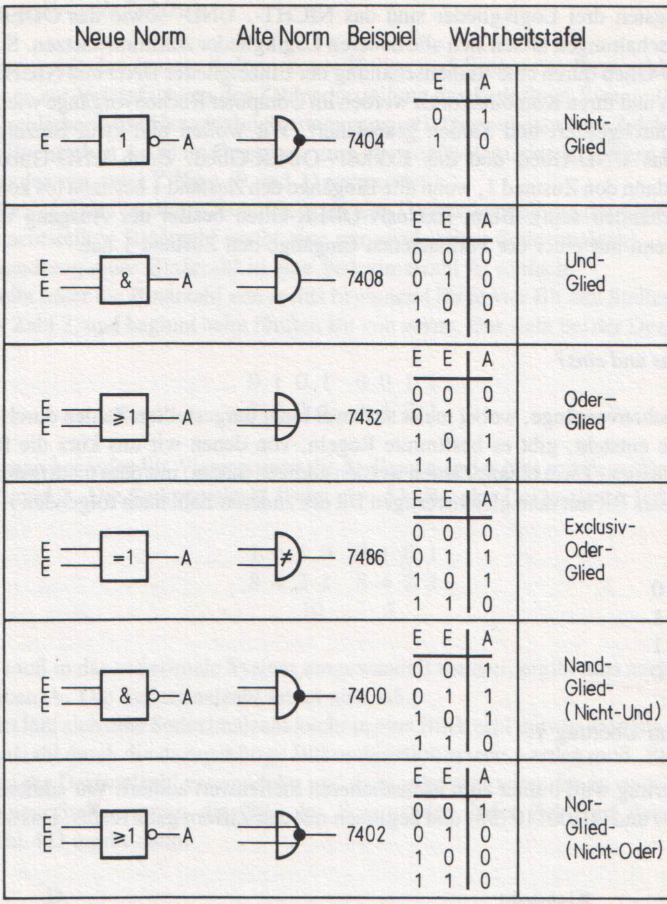


Abb. 2.2.5 Einige Logikglieder und ihre Wahrheitstafeln

neben dem Inverter eine der Grundlagen der Computerei überhaupt bilden), die zwei oder mehrere Eingänge haben und die jeder Eingangskombination von Nullen und Einsen genau ein zugehöriges Ergebnis (0 oder 1) am Ausgang zuordnen. Zum Beispiel gibt es Schaltungen mit zwei Eingängen und einem Ausgang, wo genau dann eine 1 am Ausgang erscheint, wenn der eine Eingang UND der andere Eingang den Zustand 1 besitzen; in allen anderen Fällen erscheint eine Null am Ausgang. Eine solche Schaltung, sie kann in vielen Varianten aufgebaut werden, heißt ihrem Verhalten entsprechend UND-Schaltung oder in der Fachsprache der Digitaltechniker UND-Glied. *Abb. 2.2.5* zeigt sowohl die Zuordnungstabelle, die zum UND-Glied gehört, als auch seine abstrakten Schaltsymbole nach alter und neuer Norm. Neben diesem digitalen Schalt-Glied werden noch andere wichtige aufgeführt, deren Verhalten aus der zugehörigen Zuordnungstabelle abgelesen werden kann. Die Namen solcher Schaltglieder sind vom Verhalten abgeleitet. Solche Schalt-Glieder mit mehreren Eingängen heißen auch „Verknüpfungen“, weil sie die Zustände der Eingänge hernehmen und zu einem Ausgangssignal verknüpfen.

Die wichtigsten drei Logikglieder sind das NICHT-, UND- sowie das ODER-Glied. Aus diesen Grundschaltungen lassen sich alle anderen Logikglieder zusammensetzen. So entsteht das NICHT-UND-Glied durch eine Reihenschaltung der Einzelglieder UND und NICHT. Mit diesen Logikgliedern und ihren Kombinationen werden im Computer Rechenvorgänge wie Addition und Subtraktion durchgeführt und Zahlen gespeichert. Wir wollen nun zwei Einzelglieder näher betrachten, das UND-Glied und das Exklusiv-ODER-Glied. Beim UND-Glied besitzt der Ausgang nur dann den Zustand 1, wenn alle Eingänge den Zustand 1 besitzen (es können mehrere Eingänge vorhanden sein). Beim Exklusiv-ODER-Glied besitzt der Ausgang nur dann den Zustand 1, wenn nur einer der vorhandenen Eingänge den Zustand 1 hat.

Wieviel ist eins und eins?

Für binäre Rechenvorgänge, wobei meist aus zwei binär dargestellten Zahlen durch Verknüpfungen eine neue entsteht, gibt es bestimmte Regeln, von denen wir uns kurz die für die binäre Addition anschauen. Zwei binäre Zahlen werden addiert, indem, mit dem niedrigsten Stellenwert beginnend, jedes Bit mit dem gleichwertigen Bit der anderen Zahl nach folgenden Regeln addiert wird:

0 + 0 ergibt 0
0 + 1 ergibt 1
1 + 0 ergibt 1
1 + 1 ergibt 0,

aber mit einem Übertrag 1.

Dieser Übertrag wird immer zum nächsthöheren Stellenwert addiert. Wir addieren die Zahlen 00111111 (63) und 00100110 (38) und beginnen mit den Ziffern ganz rechts. Das schaut dann so aus.

Dezimalzahl	Binärzahl	
63	0 0 1 1 1 1 1 1	
+ 38	0 0 1 0 0 1 1 0	
11	1 1 1 1 1	Übertrag
101	0 1 1 0 0 1 0 1	

Es sei nun überlegt, wie man mit Logikgliedern eine solche Addition verwirklichen kann.

Vergleicht man die Rechenregeln mit den Wahrheitstafeln in *Abb. 2.2.5*, so entsprechen diese der Verknüpfung eines Exklusiv-ODER-Gliedes, das die Summe zweier Bits bildet, mit einem UND-Glied, das den Übertrag ermittelt. Für jeden weiteren Stellenwert benötigt man aber den vorhergehenden Übertrag, der mit der Summe verknüpft werden muß. Um zwei Bits zu addieren, brauchen wir also vier UND-, zwei NICHT- und drei ODER-Glieder. Für ein Byte brauchen wir dann acht solcher Logikgruppen.

Hexerei oder sedezimale Zahlen

Um nicht immer mit den langen Kolonnen aus Nullen und Einsen bei Binärzahlen arbeiten zu müssen, gibt es zur Vereinfachung der Zahlendarstellung das Sedezimal-System (Sechzehnersystem oder fälschlicherweise Hexadezimalsystem genannt). Dieses benutzt 16 Zeichen (die Ziffern 0–9 und die Buchstaben A–F) im Gegensatz zum Dezimalsystem, das 10 Ziffern (0–9) oder das Binärsystem, das nur zwei Ziffern (0 und 1) verwendet.

Die *Tabelle 2.1.2* des letzten Abschnitts zeigt eine Gegenüberstellung der drei Stellenwertsysteme. Eine achtstellige Binärzahl ergibt eine nur zweistellige Sedezimalzahl.

Die Umwandlung einer Binärzahl in eine Sedezimalzahl ist einfach:

Man schreibt unter die Binärzahl von rechts beginnend für je vier Bit den Stellenwert, also die Potenzen der Zahl 2, und beginnt beim fünften Bit von vorne. Das sieht bei der Dezimalzahl 83 so aus:

```
0 1 0 1   0 0 1 1
8 4 2 1   8 4 2 1
```

Nun addiert man bei jeder Bit-Vierergruppe die Stellenwerte der Bits mit dem Wert 1, das ergibt die Zahlen 5 und 3. Die Sedezimalzahl lautet also 53. Bei der Dezimalzahl 165 ergibt sich:

```
1 0 1 0   0 1 0 1
8 4 2 1   8 4 2 1
  10         5
```

Die Zahl 10 muß in das sedezimale System umgewandelt werden, ergibt also nach *Tabelle 2.1.2* den Buchstaben A. Die Sedezimalzahl lautet also A5.

Umgekehrt läßt sich eine Sedezimalzahl leicht in eine Binärzahl umwandeln, da nur jede Stelle der Sedezimalzahl durch die dazugehörige Bitkombination ersetzt werden muß. Buchstaben muß man vorher in die Dezimalzahl verwandeln, und dann setzt man unter den in zwei Vierergruppen angeschriebenen Stellenwerten die Bits der Summanden jeder Zahl auf den Wert 1. Die Sedezimalzahl 4 D ergibt dann:

```
   4         D
   4        13
8 4 2 1   8 4 2 1
0 1 0 0   1 1 0 1
```

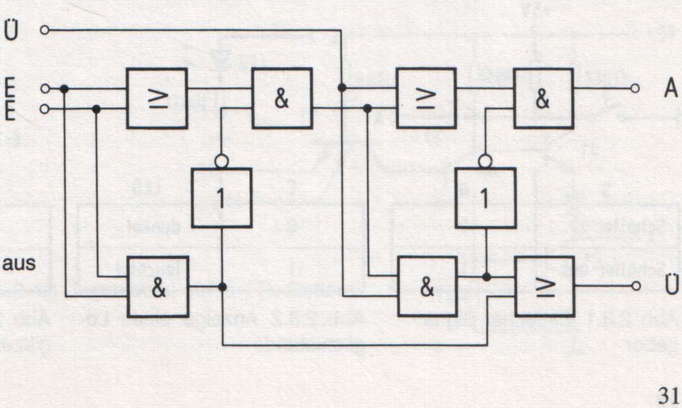


Abb. 2.2.6 Ein Addierer, aus Logikgliedern aufgebaut