

Zeitschrift: Technische Mitteilungen / Schweizerische Post-, Telefon- und Telegrafienbetriebe = Bulletin technique / Entreprise des postes, téléphones et télégraphes suisses = Bollettino tecnico / Azienda delle poste, dei telefoni e dei telegrafi svizzeri

Herausgeber: Schweizerische Post-, Telefon- und Telegrafienbetriebe

Band: 48 (1970)

Heft: 10

Artikel: Grundlagen der elektronischen Logikschaltung. 1. Teil

Autor: Wider, Peter Ferdinand

DOI: <https://doi.org/10.5169/seals-876077>

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 30.04.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

Zusammenfassung. *Der vorliegende Artikel vermittelt eine Einführung in die Grundlagen der elektronischen Logikschaltungen und deren mathematischen Hintergrund. Es werden die gebräuchlichsten Tore besprochen und anschliessend auf die Schaltungsanalyse und -synthese eingegangen. Ein kurzer Abriss der geschichtlichen Entwicklung, Hinweise auf weitergehende Logikrechenregeln und Beispiele logischer Schaltungen beschliessen die Ausführungen.*

Principes fondamentaux des circuits logiques électroniques

Résumé. *Le présent article est une initiation aux principes fondamentaux des circuits logiques électroniques et à leur fondement mathématique. Il présente les tores les plus usuels et aborde ensuite l'analyse et la synthèse des circuits. Un bref aperçu de l'évolution historique, des indications sur les règles plus poussées du calcul logique et des exemples de circuits logiques terminent les explications.*

Principio dei circuiti elettronici logici

Riassunto. *Il presente articolo contiene un'introduzione generale sui principi dei circuiti elettronici logici e le loro basi matematiche. Si descrivono le porte più in uso e in seguito si parla dell'analisi e della sintesi dei circuiti. Un breve squarcio storico dello sviluppo, accenni ad ulteriori regole dei calcoli logici ed esempi di circuiti logici concludono la relazione.*

1. Was ist Logik?

Logik ist die Lehre von der Folgerichtigkeit. Sie hat den Charakter:

Gegeben sei ein System mit den Eigenschaften x_1, x_2, \dots, x_n . Wenn sich das System logisch verhält, so kann es, gemäss seinen Eigenschaften, nur vorbestimmte Handlungen vornehmen. Zufallsprozesse spielen dabei keine Rolle.

Die Gesetze der Logik sind so aufgebaut und aufeinander abgestimmt, dass nur eindeutige Aussagen erhalten werden. Es erscheinen also nur Aussagen vom Typ:

Richtig – Falsch, Ja – Nein, Hier – Nicht hier, usw.

Aussagen wie «vielleicht», «ungefähr», usw. können in der linearen Logik nicht vorkommen. Wenn eine erste Annahme getroffen ist, so sind alle andern damit verbundenen Aussagen darausfolgend, sie sind determiniert.

Logik ist also ein Instrument, mit dessen Hilfe bestimmte (Eingangs-) Variablen so miteinander verbunden werden, dass voraussagbare (Ausgangs-) Variablen entstehen. Damit und nur damit ist es möglich, mathematisch den Wahrheitsgehalt von irgendwelchen Informationen oder Aussagen zu prüfen oder zu «errechnen». Diese mathematische Anwendung der Logik wird *Logistik* genannt.

Spricht man von elektronischer Logik, so versteht man darunter elektronische Schaltungen, die Probleme «logistisch lösen».

Anwendungsmöglichkeiten der Logik

In den Naturwissenschaften wird die Logik nach den Regeln der Booleschen Algebra angewendet.

In der elektronischen Logik müssen diese formalen Vorgänge in elektronische Schaltungen umgesetzt werden. Das geschieht mit den sog. Logikbausteinen und logischen Schaltungen. – Die Schaltungen werden nach logischen Gesetzen aufgebaut, um selbst wieder logistische Operationen ausführen zu können. Sie sind also selbst wieder Werkzeuge der reinen Logik.

Der grosse Vorteil der logischen Schaltungen liegt darin, dass man damit die Probleme elektronisch, das heisst mit

kleinem Zeitverbrauch, behandeln kann. Die Computer sind eine typische Verwirklichung solcher Schaltungen.

2. Rechensysteme

Wir rechnen normalerweise mit dem Dezimalsystem, dessen Einheit die Zahl 10 ist. Jede Zahl im Dezimalsystem kann ein «Gewicht» zwischen 0 und 9 annehmen.

Es gibt und gab aber auch noch andere Zahlensysteme, zum Beispiel das 12er-System der alten Ägypter, das auch in Europa heimisch war, oder das 2er- und das 3er-System, die heute durch die Computertechnik grossen Aufschwung erleben. Um logische Schaltungen elektronisch realisieren zu können, kann im Prinzip irgendein Zahlensystem verwendet werden. Es hat sich aber gezeigt, dass das 2er-System (auch Binär- oder Dualsystem genannt) gegenüber dem Dezimalsystem grosse Vorteile hat. Vereinzelt wird auch das 3er-System (Ternärsystem) verwendet.

Binär- und Ternärsysteme haben sich in der Technik der logischen Schaltungen deshalb solcherweise durchgesetzt, weil sie zur Darstellung einer Zahl nur 2 beziehungsweise 3 elektrische Zustände erfordern. – Warum soll denn das ein Vorteil sein, man benötigt für diese Systeme ja viel mehr Stellen zur Darstellung einer Zahl als beim Dezimalsystem?

Bei nachfolgender Zahl benötigt das Dezimalsystem 4, das Ternärsystem 8 und das Binärsystem gar 13 Zahlen:

4358 (Dezimal) = 12222102 (Ternär) = 1000100000110 (Binär)

Der Grund ist darin zu suchen, dass elektrische Schaltungen, die nur zur Realisierung binärer Werte gebraucht werden, sehr einfach und vor allem sehr billig und fehler-sicher gebaut werden können, denn binär bedeutet hier positiv/negativ oder Strom/kein Strom. Schaltungen für ternäre Systeme weisen die Werte positiv/Null/negativ auf. Sie sind also nicht mehr ganz so einfach wie binäre Schaltungen, aber immer noch viel einfacher und billiger als solche zur Realisierung dezimaler Systeme. Sie werden nur in Spezialfällen angewendet.

Tabelle I

1920	0. Generation	Realisierung mit Relais	Schaltzeit ca. 10 ms
1945	1. Generation	Realisierung mit Röhren	Schaltzeit ca. 1 µs
1955	2. Generation	Realisierung mit Halbleitern	Schaltzeit ca. 100 ns
1965	3. Generation	Realisierung mit integrierten Schaltungen	Schaltzeit ca. 10 ns

3. Was sind logische Schaltungen?

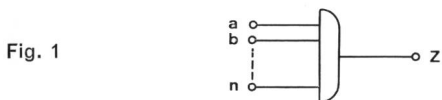
Die mathematischen Grundlagen der logischen Schaltungen wurden erstmals 1938 von Shannon beschrieben. Sie resultieren direkt aus der Booleschen Algebra, auf die in diesem Zusammenhange noch zurückzukommen sein wird. Solche Schaltungen lassen sich auf verschiedene Arten verwirklichen, und man kann wie bei den Computern von «Generationen» sprechen. In der *Tabelle I* sind die ungefähren Entstehungsdaten der einzelnen Generationen, ihre Realisierungsart und ihre Schaltgeschwindigkeit angegeben.

4. Aus was für Elementen bestehen logische Schaltungen?

Sie bestehen aus den sogenannten Logikgliedern. Diese enthalten in ihrer einfachsten Form jeweils nur eine logische Funktion. Sie können mit allen vorgenannten Generationen verwirklicht werden.

4.1 UND-Funktion (AND)

Das Symbol für das Und-Tor (And-gate) zeigt *Figur 1*:



Die Und-Funktion eines Gliedes besagt, dass die Werte der Eingänge a, b, c ... n, die nur die Werte 0 und 1 annehmen können, miteinander *multipliziert* werden müssen, um den Wert des Ausgangs Z zu erhalten.

$$a \cdot b \cdot c \cdot \dots \cdot n = Z$$

Ein Und-Tor mit zwei Eingängen ist in *Figur 2* dargestellt.



Tabelle II

a	b	Z
0	0	0
0	1	0
1	0	0
1	1	1

Die *Tabelle II* wird Wahrheitstabelle genannt. Sie zeigt, dass nur dann am Ausgang ein Signal erscheint, wenn alle Eingänge ein Signal aufweisen.

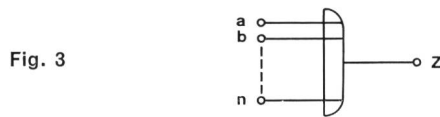
Ein praktisches Beispiel dafür ist die Aussage:

Ich gehe skifahren, wenn es schön ist *und* wenn ich Geld habe.

Das heisst ich gehe nur dann skifahren, wenn beide Bedingungen erfüllt sind.

4.2 ODER-Funktion (OR-gate)

Das entsprechende Symbol zeigt *Figur 3*:



Die Oder-Funktion eines Gliedes besagt, dass die Zustandswerte der Eingänge *addiert* werden müssen, um den Ausgangszustand zu erhalten.

$$a + b + c + \dots + n = Z$$

Für zwei Eingänge folgt also:

Tabelle III

a	b	Z
0	0	0
0	1	1
1	0	1
1	1	1

Die *Tabelle III* zeigt, dass mindestens ein Eingang ein Signal aufweisen muss, um am Ausgang den Funktionswert 1 zu erzeugen.

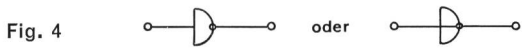
Praktisches Beispiel:

Ich gehe skifahren, wenn Rita *oder* Elisabeth mitkommt.

Das heisst mindestens eine der Damen muss mitkommen, damit ich skifahren gehe. Die Schwierigkeiten, die entstehen können, wenn beide mitkommen, bleiben in dieser Rechnung allerdings unberücksichtigt.

4.3 Nicht-Funktion (Negation, Inversion)

Symbol für diese Funktion ist *Figur 4*.



Diese Funktion sagt aus, dass das Eingangssignal umgekehrt (invertiert, negiert) wird.

$$\bar{a} = z$$

Wahrheitstabelle (Tabelle IV)

Tabelle IV

a	Z
0	1
1	0

Praktisches Beispiel:

Ich gehe nur skifahren, wenn ich *nicht* müde bin.

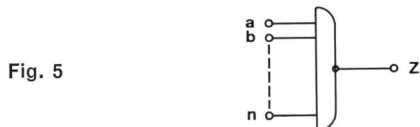
Tabelle V

Müdigkeit = a	Skifahren = Z
nein	ja
ja	nein

4.4 Zusammengesetzte logische Funktionen

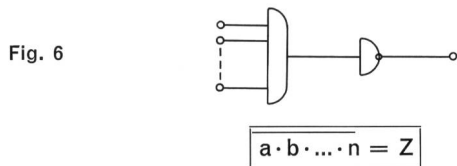
4.4.1 Nand (Not-and, Nicht-Und)

Symbol (Fig. 5)



Zusammengesetzt aus einem Und-Tor dem ein Nicht-Tor nachgeschaltet ist.

Schaltung mit Grundbausteinen (Fig. 6):



Für zwei Eingänge folgt aus Tabelle VI:

Tabelle VI

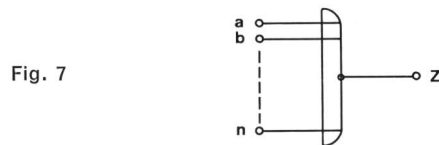
a	b	a · b	Z
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Der Ausgang wird dann zu 1, wenn mindestens ein Eingang eine Null aufweist.

Praktisches Beispiel: Kajaks schwimmen, wenn sie höchstens eine Person tragen müssen.

4.4.2 Nor (Not-or, Nicht-oder)

Symbol (Fig. 7):



Zusammengesetzt aus einem Oder-Tor mit nachgeschaltetem Nicht-Tor.

$$a + b + \dots + n = Z$$

Für zwei Eingänge folgt aus Tabelle VII:

Tabelle VII

a	b	a + b	Z
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Der Ausgang wird zu Null, wenn mindestens ein Eingang eine 1 aufweist.

Praktisches Beispiel:

Ich gehe nur dann nach Y skifahren, wenn weder A noch B dort sind.

4.4.3 Äquivalenz-Tor (Gleich-Tor)

Symbol für zwei Eingänge (Fig. 8):



Die Schaltung aufgebaut mit Grundbausteinen zeigt Figur 9:

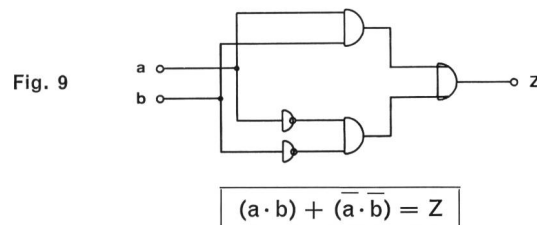


Tabelle VIII

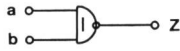
a	b	a · b	$\bar{a} \cdot \bar{b}$	Z
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Praktisches Beispiel: Fernsehsendung «Dopplet oder Nüt»

Exklusiv-Oder-Tor

Symbol für zwei Eingänge (Fig. 10):

Fig. 10



Die Schaltung verwirklicht mit Grundbausteinen (Fig. 11):

Fig. 11

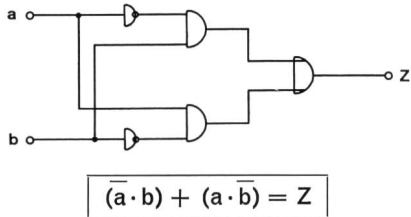


Tabelle IX

a	b	$\bar{a} \cdot b$	$a \cdot \bar{b}$	Z
0	0	0	0	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

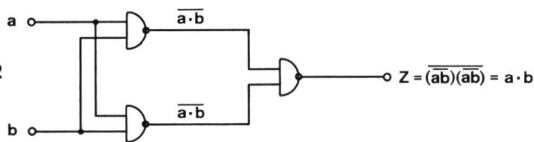
Praktisches Beispiel: Herr X ist Spezialist. Darum kann er nur in den Firmen A oder B arbeiten, wo solche Leute gebraucht werden.

Da die Nand und die Nor-gates negierende Wirkungen haben, lassen sich mit ihnen auch Und- und Oder-Tore bauen, denn es gilt:

$$Z = x = \overline{\overline{x}}$$

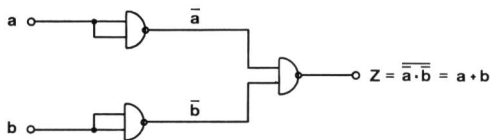
Und ist in Fig. 12 in Nand-Technik dargestellt:

Fig. 12



Oder zeigt Fig. 13 in Nand-Technik.

Fig. 13



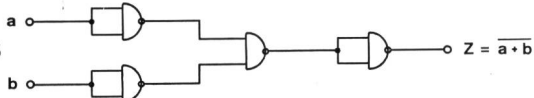
In Nicht-Technik (Fig. 14):

Fig. 14



Nor in Nand-Technik ist in Fig. 15 gezeigt

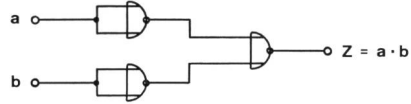
Fig. 15



Diese Funktionen können aber auch in Nor-Technik hergestellt werden, so zeigt Fig. 16:

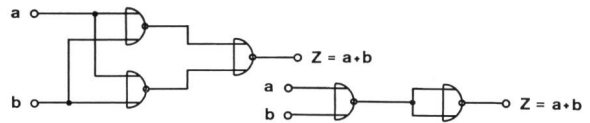
Und

Fig. 16



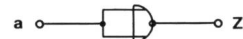
während Oder in Fig. 17 dargestellt wird.

Fig. 17



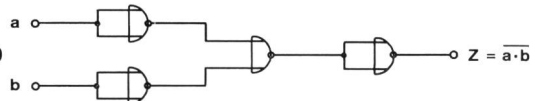
Die Nicht-Schaltung ist in Fig. 18, Nand in Fig. 19 gezeichnet

Fig. 18



Nand:

Fig. 19



5. Die Boolesche Algebra

Um mit diesen Torschaltungen überhaupt sinnvoll arbeiten zu können, müssen wir sie mathematisch in den Griff bekommen. Es hat sich gezeigt, dass die Boolesche Algebra hierfür bestens geeignet ist. Georg Boole, ein englischer Mathematiker, lebte von 1813-1864. Mit dieser nach ihm benannten Algebra stellte er ein algebraisches System auf, das selbst wieder Teil der Mengenlehre ist.

- In diesem System wird mit den «Zahlen» (Mengen, Eingänge, Eingangsvariable) a, b, c...n, gerechnet.
- Die einzelnen Zahlen dürfen nur mit + oder mit · verknüpft werden.

Andere Schreibweisen für +: \vee, \cup, und

Andere Schreibweisen für ·: $\times, *, \wedge, \cap, \text{oder}$

- Solcherart verknüpfte Zahlen ergeben «Resultate» (Mengen, Ausgänge, Ausgangsvariable) A, B, C...N (hier meist Z genannt.)
- Alle in einer solchen Rechnung vorkommenden Werte können nur die Zustände (Werte, Gewichte) 0 (\emptyset) oder 1 (U) annehmen.
- Zu jeder Zahl gibt es eine Komplementärzahl, die die Negierung der ursprünglichen Zahl genannt wird. Mathematische Schreibweise:

Gegebene Zahl a, negierte Zahl \bar{a} ($\neg a, a'$)

das heisst $a = 1$, wenn $\bar{a} = 0$

$a = 0$, wenn $\bar{a} = 1$

daraus folgt aber auch

$$\overline{\overline{a}} = a$$

Aus diesen Axiomen und Voraussetzungen ergeben sich folgende Rechengesetze:

Addition:

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= 1 \end{aligned}$$

Theoreme:

$$\begin{aligned} a + 0 &= a \\ a + 1 &= 1 \\ a + a &= a \\ a + \overline{a} &= 1 \end{aligned}$$

Multiplikation:

$$\begin{aligned} 0 \cdot 0 &= 0 \\ 0 \cdot 1 &= 0 \\ 1 \cdot 0 &= 0 \\ 1 \cdot 1 &= 1 \end{aligned}$$

$$\begin{aligned} a \cdot 0 &= 0 \\ a \cdot 1 &= a \\ a \cdot a &= a \\ a \cdot \overline{a} &= 0 \end{aligned}$$

Gesetze:

Kommutatives Gesetz: $a + b = b + a$
 $a \cdot b = b \cdot a$

Assoziatives Gesetz: $(a + b) + c = a + (b + c)$
 $(a \cdot b) \cdot c = a \cdot (b \cdot c)$

Distributives Gesetz: $a(b + c) = (a \cdot b) + (a \cdot c)$
 $a + (b \cdot c) = (a + b) \cdot (a + c)$

De Morgans Theorem: $\overline{a + b} = \overline{a} \cdot \overline{b}$
 $\overline{a \cdot b} = \overline{a} + \overline{b}$

6. Die Schaltungsanalyse

Unter Beachtung dieser Gesetze können wir nun beliebige Zusammenschaltungen der schon beschriebenen Logikbausteine entschlüsseln, oder deren Übertragungsfunktion errechnen. Man geht dabei so vor, dass der Einfluss der jeweiligen Torcharakteristik auf die betreffenden Eingangsvariablen schrittweise berechnet wird (Fig. 20).

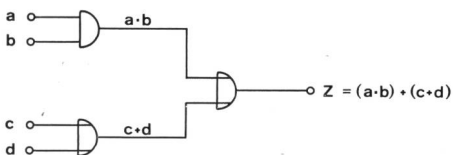


Fig. 20

7. Die Schaltungssynthese

Um eine logische Schaltung synthetisieren zu können, muss die Übertragungsfunktion oder müssen zumindest die Schaltbedingungen bekannt sein. Hievon wird die sogenannte Wahrheitstabelle erstellt, das heißt man variiert alle Eingangsvariablen auf alle möglichen Arten und bestimmt die sich jeweils daraus ergebenden Ausgangsvariablen. Die Eingänge werden dabei sinnvollerweise so

verändert, dass eine binomische Reihe entsteht. Aus dieser Tabelle werden dann alle jene Zeilen herausgezogen, deren Ausgangswert zu Eins (für die Oder-Form) beziehungsweise zu Null wird (für die Und-Form). Damit kann wiederum die sogenannte Normalfunktion erzeugt werden.

a	b	c	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Oder-Form

$$\begin{aligned} \overline{a} \cdot \overline{b} \cdot \overline{c} \\ a \cdot \overline{b} \cdot \overline{c} \\ a \cdot b \cdot \overline{c} \end{aligned}$$

Und-Form

$$\begin{aligned} a + b + c \\ a + b + \overline{c} \\ a + \overline{b} + \overline{c} \\ a + \overline{b} + c \\ \overline{a} + \overline{b} + c \\ \overline{a} + \overline{b} + \overline{c} \end{aligned}$$

Oder-Normalform:

$$\overline{a}\overline{b}\overline{c} + a\overline{b}\overline{c} + a\overline{b}c = Z$$

Diese Funktion kann mit Hilfe der vorgenannten Rechenregeln vereinfacht werden zu

$$(a + b) \cdot \overline{c} = Z$$

Regel: Alle Ziffern der herausgezogenen Zeilen mit dem Ausgangswert Eins werden durch die Variablen selbst ersetzt, während alle Ziffern mit dem Ausgangswert Null durch die Komplementwerte der Variablen ersetzt werden. Die Ziffern der Zeilen werden miteinander multipliziert und die so entstandenen Produkte addiert.

Und-Normalfunktion

$$\overline{(a + b + c)} \cdot \overline{(a + b + \overline{c})} \cdot \overline{(a + \overline{b} + \overline{c})} \cdot \overline{(a + \overline{b} + c)} = Z$$

Diese Funktion vereinfacht, ergibt ebenfalls

$$(a + b) \cdot \overline{c} = Z$$

Regel: Alle Ziffern der herausgezogenen Zeilen mit dem Ausgangswert Null werden durch die Variablen ersetzt, während alle Ziffern mit dem Ausgangswert Eins durch die Komplementwerte der Variablen ersetzt werden.

Die einzelnen Ziffern der Zeilen werden addiert und die so entstandenen Summen miteinander multipliziert.

Daraus kann nun direkt die Schaltung gefunden werden (Fig. 21):

$$\underbrace{(a + b)}_{\text{Oder-Tor}} \cdot \underbrace{\overline{c}}_{\text{Nicht-Tor}} = Z$$

Und-Tor

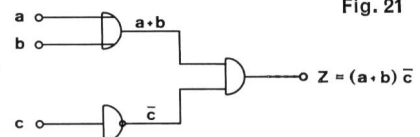


Fig. 21

(Fortsetzung folgt)