

Fach: Automation

##### Thema: Steuerungsgrundlagen

##### 

##### Logische Grundbausteine

##### Kapitel: Schaltalgebra

##### Autor: Roman Moser

**Inhaltsverzeichnis**

[1. Introduction 2](#_Toc242285159)

[2. Gesetze der Schaltalgebra 3](#_Toc242285160)

[2.1 Disjunktion (ODER-Funktion) 3](#_Toc242285161)

[2.2 Konjunktion (UND-Funktion) 3](#_Toc242285162)

[2.3 Negation (NICHT-Verknüpfung) 3](#_Toc242285163)

[2.4 Rechenregeln 3](#_Toc242285164)

[2.5 Tasks - Vereinfachungen 7](#_Toc242285165)

# Introduction



The mid-nineteenth-century mathematician

George Boole invented a system of algebra for the mathematical analysis of logic.

Today we use his Boolean algebra to express the functions of our logic gates in digital electronic circuits, but students familiar with the theory of sets will recognise much of the arithmetic which today electronics engineers think of as their own.

We can break down the logic functions used in

electronics into three fundamental types and then use combinations of these to generate other functions.



**Tasks**

1. Read the text about George Boole carefully and try to find out the English equivalent of the verbs given in the table. Do it without the aid of a dictionary!

|  |  |
| --- | --- |
| English: | German: |
| (to) break down | aufgliedern |
| (to) recognise | erkennen |
| (to) think of as | etwas als etwas betrachten |
| (to) generate | erzeugen |
| (to) invent | erfinden |
| (to) express | darstellen, ausdrücken |

2) The writer mentions three fundamental types of logic functions. Write down the names of these functions!

AND, OR, NOT

# Gesetze der Schaltalgebra

Die Schaltalgebra handelt von binären Variablen und logischen Operatoren.

Die drei grundlegenden logischen Operatoren sind UND, ODER und NICHT.

Damit ist es möglich, sogenannte Schaltfunktionen kompakt darzustellen und mit ihnen zu rechnen.

Die Zeichen für die Operatoren sind wie folgt:

|  |  |  |  |
| --- | --- | --- | --- |
|  | UND | ODER | NICHT |
| übliche Darstellung: |  |  |  |
| Amerikanische Darstellung: |  |  |  |

Wir werden uns an die „übliche“ Notation halten!

## Disjunktion (ODER-Funktion)

Beispiel:

|  |  |  |
| --- | --- | --- |
| X | 1 | ? |
| 0 | 1 | 1 |
| 1 | 1 | 1 |

≥1

?

X

1

## Konjunktion (UND-Funktion)

## Negation (NICHT-Verknüpfung)

## Rechenregeln

### Vertauschungsregel (Kommutatives Gesetz)

&

Q1

X2

X1

&

Q1

X1

X2

### Verbindungsregel (Assoziatives Gesetz)

&

Q1

X1

X3

X2

&

Q1

X3

&

X1

X2

### Verteilungsregel (Distributives Gesetz)

Q1

≥1

&

X1

&

X3

X2

&

Q1

X1

≥1

X2

X3

Herleitung:

### Absorptionsregel

Herleitung:

Herleitung:

Merke:

* Eine UND-Verknüpfung bindet stets stärker als eine ODER-Verknüpfung!

z.B.

* Klammern haben vor UND-Verknüpfungen Vorrang.
* Negationszeichen haben vor Klammern Vorrang.

### De Morgansches Gesetz

Beweis:

Q1

&

1

X1

1

X2

Q1

≥1

1

X1

1

X2

|  |  |  |
| --- | --- | --- |
| X2 | X1 | Q1 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

|  |  |  |
| --- | --- | --- |
| X2 | X1 | Q1 |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

&

Q1

X1

X2

≥1

Q1

X1

X2

Daraus ergibt sich, dass alle Verknüpfungsschaltungen …

* … mit **ODER** und **NICHT** …
* … nur mit **NOR** …
* … mit **UND** und **NICHT** …
* … nur mit **NAND** …

… Gliedern aufgebaut werden können!

## Tasks - Simplifications

|  |  |  |
| --- | --- | --- |
| I2 | I1 | Q1 |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

***Guided example***

From the truth table you can see that .



De Morgan’sches Gesetz