Boolean Differential Calculus

Synthesis Lectures on Digital Circuits and Systems

Editor
Mitchell A. Thornton, ***Southern Methodist University***

The *Synthesis Lectures on Digital Circuits and Systems* series is comprised of 50- to 100-page books targeted for audience members with a wide-ranging background. The Lectures include topics that are of interest to students, professionals, and researchers in the area of design and analysis of digital circuits and systems. Each Lecture is self-contained and focuses on the background information required to understand the subject matter and practical case studies that illustrate applications. The format of a Lecture is structured such that each will be devoted to a specific topic in digital circuits and systems rather than a larger overview of several topics such as that found in a comprehensive handbook. The Lectures cover both well-established areas as well as newly developed or emerging material in digital circuits and systems design and analysis.

Boolean Differential Calculus
Bernd Steinbach and Christian Posthoff
2017

Embedded Systems Design with Texas Instruments MSP432 32-bit Processor
Dung Dang, Daniel J. Pack, and Steven F. Barrett
2016

Fundamentals of Electronics: Book 4 Oscillators and Advanced Electronics Topics
Thomas F. Schubert and Ernest M. Kim
2016

Fundamentals of Electronics: Book 3 Active Filters and Amplifier Frequency
Thomas F. Schubert and Ernest M. Kim
2016

Bad to the Bone: Crafting Electronic Systems with BeagleBone and BeagleBone Black, Second Edition
Steven F. Barrett and Jason Kridner
2015

Fundamentals of Electronics: Book 2 Amplifiers: Analysis and Design
Thomas F. Schubert and Ernest M. Kim
2015

Fundamentals of Electronics: Book 1 Electronic Devices and Circuit Applications
Thomas F. Schubert and Ernest M. Kim
2015

Applications of Zero-Suppressed Decision Diagrams
Tsutomu Sasao and Jon T. Butler
2014

Modeling Digital Switching Circuits with Linear Algebra
Mitchell A. Thornton
2014

Arduino Microcontroller Processing for Everyone! Third Edition
Steven F. Barrett
2013

Boolean Differential Equations
Bernd Steinbach and Christian Posthoff
2013

Bad to the Bone: Crafting Electronic Systems with BeagleBone and BeagleBone Black
Steven F. Barrett and Jason Kridner
2013

Introduction to Noise-Resilient Computing
S.N. Yanushkevich, S. Kasai, G. Tangim, A.H. Tran, T. Mohamed, and V.P. Shmerko
2013

Atmel AVR Microcontroller Primer: Programming and Interfacing, Second Edition
Steven F. Barrett and Daniel J. Pack
2012

Representation of Multiple-Valued Logic Functions
Radomir S. Stankovic, Jaakko T. Astola, and Claudio Moraga
2012

Arduino Microcontroller: Processing for Everyone! Second Edition
Steven F. Barrett
2012

Advanced Circuit Simulation Using Multisim Workbench
David Báez-López, Félix E. Guerrero-Castro, and Ofelia Delfina Cervantes-Villagómez
2012

Circuit Analysis with Multisim
David Báez-López and Félix E. Guerrero-Castro
2011

Microcontroller Programming and Interfacing Texas Instruments MSP430, Part I
Steven F. Barrett and Daniel J. Pack
2011

Microcontroller Programming and Interfacing Texas Instruments MSP430, Part II
Steven F. Barrett and Daniel J. Pack
2011

Pragmatic Electrical Engineering: Systems and Instruments
William Eccles
2011

Pragmatic Electrical Engineering: Fundamentals
William Eccles
2011

Introduction to Embedded Systems: Using ANSI C and the Arduino Development Environment
David J. Russell
2010

Arduino Microcontroller: Processing for Everyone! Part II
Steven F. Barrett
2010

Arduino Microcontroller Processing for Everyone! Part I
Steven F. Barrett
2010

Digital System Verification: A Combined Formal Methods and Simulation Framework
Lun Li and Mitchell A. Thornton
2010

Progress in Applications of Boolean Functions
Tsutomu Sasao and Jon T. Butler
2009

Embedded Systems Design with the Atmel AVR Microcontroller: Part II
Steven F. Barrett
2009

Embedded Systems Design with the Atmel AVR Microcontroller: Part I
Steven F. Barrett
2009

Embedded Systems Interfacing for Engineers using the Freescale HCS08 Microcontroller II: Digital and Analog Hardware Interfacing
Douglas H. Summerville
2009

Designing Asynchronous Circuits using NULL Convention Logic (NCL)
Scott C. Smith and JiaDi
2009

Embedded Systems Interfacing for Engineers using the Freescale HCS08 Microcontroller I: Assembly Language Programming
Douglas H.Summerville
2009

Developing Embedded Software using DaVinci & OMAP Technology
B.I. (Raj) Pawate
2009

Mismatch and Noise in Modern IC Processes
Andrew Marshall
2009

Asynchronous Sequential Machine Design and Analysis: A Comprehensive Development of the Design and Analysis of Clock-Independent State Machines and Systems
Richard F. Tinder
2009

An Introduction to Logic Circuit Testing
Parag K. Lala
2008

Pragmatic Power
William J. Eccles
2008

Multiple Valued Logic: Concepts and Representations
D. Michael Miller and Mitchell A. Thornton
2007

Finite State Machine Datapath Design, Optimization, and Implementation
Justin Davis and Robert Reese
2007

Atmel AVR Microcontroller Primer: Programming and Interfacing
Steven F. Barrett and Daniel J. Pack
2007

Pragmatic Logic
William J. Eccles
2007

PSpice for Filters and Transmission Lines
Paul Tobin
2007

PSpice for Digital Signal Processing
Paul Tobin
2007

PSpice for Analog Communications Engineering
Paul Tobin
2007

PSpice for Digital Communications Engineering
Paul Tobin
2007

PSpice for Circuit Theory and Electronic Devices
Paul Tobin
2007

Pragmatic Circuits: DC and Time Domain
William J. Eccles
2006

Pragmatic Circuits: Frequency Domain
William J. Eccles
2006

Pragmatic Circuits: Signals and Filters
William J. Eccles
2006

High-Speed Digital System Design
Justin Davis
2006

Introduction to Logic Synthesis using Verilog HDL
Robert B.Reese and Mitchell A.Thornton
2006

Microcontrollers Fundamentals for Engineers and Scientists
Steven F. Barrett and Daniel J. Pack
2006

Boolean Differential Calculus

Bernd Steinbach and Christian Posthoff

www.morganclaypool.com

ISBN: 9781627059220 paperback
ISBN: 9781627056175 ebook

DOI 10.2200/S00766ED1V01Y201704DCS052

A Publication in the Morgan & Claypool Publishers series
SYNTHESIS LECTURES ON DIGITAL CIRCUITS AND SYSTEMS

Lecture #52
Series Editor: Mitchell A. Thornton, *Southern Methodist University*
Series ISSN
Print 1932-3166 Electronic 1932-3174

Boolean Differential Calculus

Bernd Steinbach
Freiberg University of Mining and Technology, Germany

Christian Posthoff
The University of West Indies, Trinidad & Tobago

SYNTHESIS LECTURES ON DIGITAL CIRCUITS AND SYSTEMS #52

MORGAN & CLAYPOOL PUBLISHERS

ABSTRACT

The Boolean Differential Calculus (BDC) is a very powerful theory that extends the basic concepts of Boolean Algebras significantly.

Its applications are based on Boolean spaces $\mathbb{B}$ and $\mathbb{B}^n$, Boolean operations, and basic structures such as Boolean Algebras and Boolean Rings, Boolean functions, Boolean equations, Boolean inequalities, incompletely specified Boolean functions, and Boolean lattices of Boolean functions. These basics, sometimes also called switching theory, are widely used in many modern information processing applications.

The BDC extends the known concepts and allows the consideration of *changes* of function values. Such changes can be explored for pairs of function values as well as for whole subspaces. The BDC defines a small number of *derivative* and *differential* operations. Many existing theorems are very welcome and allow new insights due to possible transformations of problems. The available operations of the BDC have been efficiently implemented in several software packages.

The common use of the basic concepts and the BDC opens a very wide field of applications. The roots of the BDC go back to the practical problem of testing digital circuits. The BDC deals with changes of signals which are very important in applications of the analysis and the synthesis of digital circuits. The comprehensive evaluation and utilization of properties of Boolean functions allow, for instance, to decompose Boolean functions very efficiently; this can be applied not only in circuit design, but also in data mining. Other examples for the use of the BDC are the detection of hazards or cryptography. The knowledge of the BDC gives the scientists and engineers an extended insight into Boolean problems leading to new applications, e.g., the use of Boolean lattices of Boolean functions.

KEYWORDS

Boolean Differential Calculus, derivative operation, differential operation, Boolean Algebra, Boolean Ring, Boolean function, Boolean equation, Boolean lattice, applications, XBOOLE

Contents

Introduction

Boolean (two-valued, dual, binary) systems have a long history and tradition in science and technology. In our daily life we normally meet the decimal number system that uses ten digits which get their meaning in a number based on powers of 10. The representation by means of dual numbers only needs two digits 0 and 1. Instead of the powers of 10 now powers of 2 are taken into consideration.

Arithmetic using dual numbers has been completely presented by G. W. Leibniz (*Explication de l' Arithmétique Binaire* (Histoire de l'Academie Royale des Sciences 1703), published in Paris 1705). In 1854, George Boole published a precedent-setting book which describes a system that formalizes the *laws of correct thinking* (Boole [1854, Reprint: 2010]). He was working with the two values *true* and *false* and created a propositional calculus which today is called *Boolean Algebra*. The next famous name is Claude Shannon. He proved that Boolean Algebra and dual arithmetic could be used to simplify the arrangement of the electromechanical relays that were used then in telephone call routing switches. Next, he expanded this concept proving that it would be possible to use arrangements of relays to solve problems in Boolean Algebra. Using this property of electrical switches to implement logic is the fundamental concept that underlies all electronic digital computers. Shannon's work became the foundation of digital circuit design. The restriction to only two digits and the utilization of the ideas of Boole and Shannon are the keys that Conrad Zuse was able to realize the first computer.

Since then a gigantic building of theory and applications has been created that requires careful and comprehensive studies. Switching theory deals with Boolean structures using the elements of $\mathbb{B} = \{0, 1\}$ and covers a wide field of applications. However, there are applications which require information about the effects of changing the values of variables and functions. One of them is the test of combinational circuits. Investigations of this topic by Reed [1954], Huffman [1958], and Akers [1959] can be seen as the roots of the *Boolean Differential Calculus* (BDC).

Basics of the BDC were published by A. Thayse and his group (Davio et al. [1978], Thayse [1981]). Comprehensive studies were carried out by a research group at the Chemnitz University of Technology where the authors are coming from. Results of this research were published in German in the monograph about the Boolean Differential Calculus by Bochmann and Posthoff [1981]. Chapters in Posthoff and Steinbach [2004] and Sasao and Butler [2010] summarize the main definitions of the BDC and present some selected applications. An extensive article about the BDC and its applications (Steinbach and Posthoff [2010a]) was published in the *Journal of Computational and Theoretical Nanoscience*.

The BDC was successfully applied in the last years to solve many problems. This success is not only based on the theoretical background of the BDC, but also on the efficient implementation of the derivative operations as part of the XBOOLE library as shown by Steinbach [1992]. The book Steinbach and Posthoff [2009] contains many examples and exercises where the BDC and the XBOOLE-Monitor (freely available over the Internet) were efficiently used. This book also provides the solution for all exercises and supplements the textbook Posthoff and Steinbach [2004].

The naming of several concepts of the BDC has been based on *formal analogies* with the *differential* and *integral calculus* for real numbers and functions. However, it should not be forgotten that we are dealing with finite algebraic structures without concepts like *limits* etc. Nevertheless, these analogies can be extended to create a *Boolean Integral Calculus*. The basics of this inverse calculus were already provided in the monograph Bochmann and Posthoff [1981]. The Boolean Integral Calculus and especially methods to solve Boolean differential equations have been presented in the Ph.D. thesis of Steinbach [1981]. The book Steinbach and Posthoff [2013a], published by Morgan & Claypool, explains these methods, their practical calculations using the XBOOLE-Monitor, and demonstrates many applications.

We also can consider the *possibility* of changing the values of variables and functions by using differentials of variables $\mathrm{d}x_i$ and functions $\mathrm{d}f_j$. Using such differentials we get an excellent possibility to describe graphs and model their properties.

The BDC basically deals with changing the values of Boolean functions. Extensions to multivalued variables and functions are given in Yanushkevich [1998], and in combination with binary concepts already in Bochmann and Posthoff [1981]. Some comments according to this approach are repeated in Bochmann [2008].

Replacing a single Boolean function by a lattice of such functions is an important source for finding optimal solutions, especially in circuit design. Therefore, derivative operations of lattices become a new problem. A recent research result of Steinbach and Posthoff [2013b, 2015] and Steinbach [2013] is the extension of the BDC to lattices of Boolean functions. These extensions and promising applications are included into this presentation of the Boolean Differential Calculus.

It is our aim to present the theory of the BDC and many applications in a well understandable manner. The classification of the single derivatives as a special case of the vectorial derivatives contributes to this aim. The reader should always try to understand the concepts and do as many examples as possible. To this end we added several exercises at the end of the chapters and give the associated solutions in the final chapter. We hope that this book helps the readers to solve their relevant problems more efficiently.

Bernd Steinbach and Christian Posthoff
December 2016

CHAPTER 1

Basics of Boolean Structures

1.1 LATTICES AND FUNCTIONS

We use the set $\mathbb{B} = \{0, 1\}$ with two different elements 0 and 1 as the initial point. An order relation $\leq$ can be defined as follows:

$$0 \leq 0 \,, \quad 0 \leq 1 \,, \quad 1 \leq 1 \,.$$

This order relation allows the finding of the minimum and the maximum of any two elements:

$$\begin{aligned} \min(0,0) &= 0 \,, \quad \min(0,1) = 0 \,, \quad \min(1,0) = 0 \,, \quad \min(1,1) = 1 \,, \\ \max(0,0) &= 0 \,, \quad \max(0,1) = 1 \,, \quad \max(1,0) = 1 \,, \quad \max(1,1) = 1 \,. \end{aligned}$$

Instead of *minimum* and *maximum* the operation signs $\wedge$ and $\vee$ are used, together with the names *conjunction* and *disjunction* instead of minimum and maximum:

$$\min(x, y) = x \wedge y \,, \quad \max(x, y) = x \vee y \,.$$

This results in Table 1.1.

Table 1.1: Conjunction and disjunction

x_1	x_2	$x_1 \wedge x_2$	$x_1 \vee x_2$	$\min(x_1, x_2)$	$\max(x_1, x_2)$
0	0	0	0	0	0
0	1	0	1	0	1
1	0	0	1	0	1
1	1	1	1	1	1

It means that we can start with the order relation $\leq$ and define the conjunction and the disjunction as the minimum and the maximum of two values. It is, however, also possible to define these two operations $\wedge$ and $\vee$ by means of Table 1.1 and take the conjunction for the minimum and the disjunction for the maximum. This is justified because in each row the value $x_1 \wedge x_2$ is less than the value of $x_1 \vee x_2$. The two approaches are completely equivalent.

Now we formulate the following laws.

1. **Commutativity:**

$$x_1 \vee x_2 = x_2 \vee x_1 \,, \tag{1.1}$$
$$x_1 \wedge x_2 = x_2 \wedge x_1 \,, \tag{1.2}$$

2. **Associativity:**

$$x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3 = x_1 \vee x_2 \vee x_3 \,, \tag{1.3}$$
$$x_1 \wedge (x_2 \wedge x_3) = (x_1 \wedge x_2) \wedge x_3 = x_1 \wedge x_2 \wedge x_3 \,, \tag{1.4}$$

3. **Idempotence:**

$$x \vee x = x \,, \tag{1.5}$$
$$x \wedge x = x \,, \tag{1.6}$$

4. **Absorption:**

$$x_1 \vee (x_1 \wedge x_2) = x_1 \,, \tag{1.7}$$
$$x_1 \wedge (x_1 \vee x_2) = x_1 \,. \tag{1.8}$$

The absorption laws connect the two operations $\wedge$ and $\vee$. The other laws include only one operation.

Table 1.2 shows the proof of one of these laws; this *table method* is not difficult and very typical. We enumerate all the vectors of $\mathbb{B} \times \mathbb{B}$ and calculate the left and the right side of the equation. The equality of the first and the last column show that the equation is correct.

Table 1.2: Proof of the absorption law (1.7)

x_1	x_2	$x_1 \wedge x_2$	$x_1 \vee (x_1 \wedge x_2)$
0	0	0	**0**
0	1	0	**0**
1	0	0	**1**
1	1	1	**1**

These four laws define the algebraic structure of a *lattice*.

Now we extend this concept and build vectors with n components where each component will take one of these two values.

Definition 1.1 Boolean Space. For $\mathbb{B} = \{0, 1\}$,

$$\mathbb{B}^n = \{\mathbf{x} \,|\, \mathbf{x} = (x_1, x_2, \ldots, x_{n-1}, x_n),\ x_i \in \mathbb{B},\ i = 1, \ldots, n\} \tag{1.9}$$

is the set of all binary vectors of length n, the *Boolean space* $\mathbb{B}^n$.

It can also be understood as the *cross product*

$$\mathbb{B}^n = \underbrace{\mathbb{B} \times \ldots \times \mathbb{B}}_{n \text{ times}} = \{\mathbf{x} \,|\, \mathbf{x} = (x_1, \ldots, x_n),\ x_i \in \mathbb{B},\ i = 1, \ldots, n\} \,.$$

Here are two small examples:

$$\begin{aligned}\mathbb{B}^2 &= \{(00), (01), (10), (11)\}\ ,\\ \mathbb{B}^4 &= \{(0000), (0001), (0010), (0011), (0100), (0101), (0110), (0111),\\ &\qquad (1000), (1001), (1010), (1011), (1100), (1101), (1110), (1111)\}\ .\end{aligned}$$

Since there are n components and each component can take on two values, $\mathbb{B}^n$ contains 2^n different elements (vectors).

For $\mathbb{B}^n$ we introduce a *partial order relation* in the following way: for $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ the $\leq$ of $\mathbb{B}$ has to be applied for each component:

$$\mathbf{x} \leq \mathbf{y} \quad \text{if and only if} \quad x_i \leq y_i \quad \text{for} \quad i = 1, \ldots, n\ .$$

The effect of this definition explains the name *partial order*: for some vectors we get $\mathbf{x} \leq \mathbf{y}$, for instance $(0010) \leq (1011)$, but for other vectors it can be possible that there we have neither $\mathbf{x} \leq \mathbf{y}$ nor $\mathbf{y} \leq \mathbf{x}$, for instance $\mathbf{x} = (1001)$, $\mathbf{y} = (0110)$. These vectors are not comparable.

Nevertheless, this relation allows again the definition of the *minimum and the maximum of two vectors* by applying the conjunction and the disjunction of $\mathbb{B}$ component by component:

$$\begin{aligned}\mathbf{x} &= (10011100)\ ,\\ \mathbf{y} &= (00110101)\ ,\\ \mathbf{x} \wedge \mathbf{y} &= (00010100)\ ,\\ \mathbf{x} \vee \mathbf{y} &= (10111101)\ .\end{aligned}$$

The properties of a lattice are again satisfied, $(\mathbb{B}^n, \wedge, \vee)$ is also a *lattice*.

As we can see by this example, neither the minimum nor the maximum must be equal to one of the given vectors. When all the vectors of $\mathbb{B}^n$ are included, then the minimum is equal to the vector $\mathbf{0} = (0, 0, \ldots, 0)$, the maximum is equal to $\mathbf{1} = (1, 1, \ldots, 1)$, and these two vectors are also elements of $\mathbb{B}^n$.

Now a subset S of $\mathbb{B}^n$ is considered: $S \subset \mathbb{B}^n$. The conjunction of all elements of S is the minimum $\min(S)$ of S, the disjunction of all elements of S is the maximum $\max(S)$ of S. The vectors $\min(S)$ and $\max(S)$ can be an element of S, but not necessarily. If the conjunction and the disjunction of any two elements of $S \subset \mathbb{B}^n$ are again elements of S then S is a *sublattice* of $\mathbb{B}^n$.

For the representation of Boolean functions a second relation will be considered, an *order relation* where all the elements are comparable. The easiest way is the use of the decimal equivalent of a dual vector. When n is set to 3, for instance, then we get

$$(000), (001), (010), (011), (100), (101), (110), (111)$$

in this order, and this corresponds to the dual representation of the numbers $0, 1, 2, 3, 4, 5, 6, 7$. It is often named *lexicographic order*. Very often such vectors will be used as the argument vectors of Boolean functions, and it is possible to assign the variables x_1, x_2, x_3 to such vectors.

Now everything is prepared to define *Boolean functions*.

Definition 1.2 Boolean Function. Each unique mapping from $\mathbb{B}^n$ into $\mathbb{B}$ is a *Boolean function* of n variables.

Since $\mathbb{B}^n$ has 2^n elements, there are 2^{2^n} different functions of n variables. Important are the elementary functions for $n = 1$ and $n = 2$. We start with $n = 1$. Then $2^n = 2$ and $2^{2^n} = 4$. Therefore we have two values for the single variable and four different functions. Table 1.3 shows the four different functions of one variable x, $f_i(x)$ will be used for the respective function values:

$$f_0(x) = 0(x)\ , \quad f_1(x) = x\ , \quad f_2(x) = \overline{x}\ , \quad \text{and} \quad f_3(x) = 1(x)\ .$$

Table 1.3: Boolean functions of one variable

x	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

When we read the function values vertically, then we see the binary vectors of $\mathbb{B}^2$:

$$(f_i(0), f_i(1)): \quad (00) \quad (01) \quad (10) \quad (11)\ .$$

The functions $0(x)$ and $1(x)$ are constant $= 0$ and $= 1$, resp., the variable x indicates that these functions depend on one variable.

The function $f_2(x)$ converts the two values into each other; it is called **negation** and indicated by a line above the variable, i.e., by $\overline{x}$:

$$\overline{0} = 1\ , \qquad \overline{1} = 0\ . \tag{1.10}$$

Literal can be used as a *common name* for variables and negated variables. The twofold use of the negation reproduces the original value:

$$\overline{\overline{x}} = x\ .$$

The functions of two variables can be seen in Table 1.4.

The set of all possible Boolean functions of two variables is equal to $\mathbb{B}^4$ and satisfies the axioms of a lattice.

There are two *constant* functions: $f_0(x) = 0(x_1, x_2)$ and $f_{15}(x_1, x_2) = 1(x_1, x_2)$. The function $f_1(x_1, x_2)$ is already known as **conjunction**:

$$f_1(x_1, x_2) = x_1 \wedge x_2\ . \tag{1.11}$$

Table 1.4: Boolean functions of two variables

x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Very often $\wedge$ is omitted, and we simply write $f_1(x_1, x_2) = x_1 x_2$. The main property of this function: it is only equal to 1 if *both* arguments are equal to 1 ($x_1 = x_2 = 1$). This has been seen already when the minimum for $\mathbb{B}$ has been defined.

We are also familiar with $f_7(x_1, x_2)$, the **disjunction**:

$$f_7(x_1, x_2) = x_1 \vee x_2 \ . \tag{1.12}$$

The important property of this function: $f_7(x_1, x_2) = x_1 \vee x_2 = 0$ only holds for $x_1 = 0$ *and* $x_2 = 0$ (remember the maximum for $\mathbb{B}$).

Two functions are very appropriate to indicate the *inequality* and the *equality* of the arguments (see Table 1.4) by their function values:

$$f_6(x_1, x_2) = x_1 \oplus x_2 \quad \textbf{(antivalence)} \tag{1.13}$$

and

$$f_9(x_1, x_2) = x_1 \odot x_2 \quad \textbf{(equivalence)} \ . \tag{1.14}$$

It can be seen that the vector of the antivalence (0110) can be transformed into the vector of the equivalence (1001) and vice versa by means of the negation:

$$\overline{x_1 \odot x_2} = x_1 \oplus x_2 \ , \qquad \overline{x_1 \oplus x_2} = x_1 \odot x_2 \ . \tag{1.15}$$

Two more functions are very often used in circuit design:

$$f_8(x_1, x_2) = \mathrm{NOR}(x_1, x_2) = \overline{x_1 \vee x_2} \tag{1.16}$$

and

$$f_{14}(x_1, x_2) = \mathrm{NAND}(x_1, x_2) = \overline{x_1 \wedge x_2} \ . \tag{1.17}$$

Each reader must ensure that he knows all these functions and can use them properly. It can be taken into consideration that the length of binary vectors representing a Boolean function always is a power of 2: $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, ...

The main problem that must be addressed for many applications is the *exponential complexity*: for $n = 3$ there are already $2^{2^3} = 2^8 = 256$ functions of 3 variables, etc.

In order to cope with these problems formulas will be used. Each correct formula must be built by means of the following steps.

Definition 1.3 Construction of Formulas.

1. The constants 0 and 1 and the single variables $x_1, \ldots, x_n$ are formulas.
2. If F is a formula then $\overline{F}$ is also a formula.
3. If F_1 and F_2 are formulas then

$$(F_1 \wedge F_2)\,, \quad (F_1 \vee F_2)\,, \quad (F_1 \oplus F_2)\,, \quad \text{and} \quad (F_1 \odot F_2)$$

   are also formulas.
4. Each formula can be built when the second and the third rule are applied finitely many times.

Example 1.4 Construction of a Formula of Four Variables.

- We start with x_1, x_2, x_3 and x_4 .
- Thereafter, we build $F_1 = (x_1 \vee x_2)$ and $F_2 = (x_3 \oplus x_4)$.
- Next, we set $F_3 = \overline{F}_2$.
- Finally:

$$f(x_1, x_2, x_3, x_4) = (F_1 \wedge F_3) = (x_1 \vee x_2) \wedge \overline{(x_3 \oplus x_4)}\,.$$

The formula is a nice abbreviation, however, it does not really show the function values. In order to find these values we must go back to the 16 possible combinations of values and calculate each value of $F(x_1, x_2, x_3, x_4)$ by means of the respective parts of the formula. And this brings back the problem of very large tables; see Table 1.5 for the example.

Example 1.5 Lattice of Boolean Functions. The functions $f_1(x_1, x_2, x_3)$, $f_2(x_1, x_2, x_3)$, and $f_3(x_1, x_2, x_3)$ will be given as follows:

$$\begin{array}{lcccccccc} f_1(x_1, x_2, x_3): & (1 & 0 & 1 & 0 & 0 & 1 & 0 & 0)\,, \\ f_2(x_1, x_2, x_3): & (1 & 0 & 1 & 0 & 1 & 1 & 0 & 0)\,, \\ f_3(x_1, x_2, x_3): & (1 & 0 & 1 & 1 & 1 & 1 & 0 & 0)\,. \end{array}$$

Table 1.5: The function $f = (x_1 \vee x_2) \wedge \overline{(x_3 \oplus x_4)}$

x_1	x_2	x_3	x_4	$x_1 \vee x_2$	$\overline{x_3 \oplus x_4}$	$f(x_1, x_2 x_3, x_4)$
0	0	0	0	0	1	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	1	1	1
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	0	1	1	0	0
1	0	1	0	1	0	0
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	1	1

It can be seen that $f_1 \leq f_2$, $f_1 \leq f_3$ and $f_2 \leq f_3$. This means that

$$\min(f_1, f_2, f_3) = f_1 \ , \quad \max(f_1, f_2, f_3) = f_3 \ .$$

The $\wedge$ and the $\vee$ of any of these three functions is again one of these three functions. The set $\{f_1, f_2, f_3\}$ is *closed* with regard to these two operations and therefore a lattice.

Another possibility is the following *ternary vector*:

$$(011 - 0 - 10) \ .$$

Sometimes it is also written as

$$(011 \ \Phi \ 0 \ \Phi \ 10) \ .$$

The dash $(-)$ or the Φ can be arbitrarily replaced by 0 or 1, in this way we get four functions that satisfy the axioms of a lattice. In circuit design such a lattice is generated by *don't-care conditions*.

Now we split the algebraic background into two directions: *Boolean Algebras* and *Boolean Rings*.

1.2 BOOLEAN ALGEBRAS

We have already considered the four properties of a lattice, now we can add some more laws. The verification of these laws follows the same procedure as before, we enumerate all possibilities and check the left and the right side of the equations. The variables of the next formulas always take their values from $\mathbb{B}$, but all of them can be replaced by vectors when the operations are performed component by component. The 0 must be replaced by the vector $\mathbf{0} = (00\ldots0)$ and the 1 by $\mathbf{1} = (11\ldots1)$ when vector operations are computed:

1. **Distributivity:**

$$x_1 \vee (x_2 \wedge x_3) = (x_1 \vee x_2) \wedge (x_1 \vee x_3)\ , \tag{1.18}$$
$$x_1 \wedge (x_2 \vee x_3) = (x_1 \wedge x_2) \vee (x_1 \wedge x_3)\ , \tag{1.19}$$

2. **Neutral elements:**

$$0 \vee x = x\ , \tag{1.20}$$
$$1 \wedge x = x\ , \tag{1.21}$$

3. **Complement:**

$$x \vee \overline{x} = 1\ , \tag{1.22}$$
$$x \wedge \overline{x} = 0\ , \tag{1.23}$$

4. **De Morgan's Laws:**

$$\overline{x_1 \wedge x_2} = \overline{x}_1 \vee \overline{x}_2\ , \tag{1.24}$$
$$\overline{x_1 \vee x_2} = \overline{x}_1 \wedge \overline{x}_2\ . \tag{1.25}$$

Both these four laws and the four laws of a lattice (see page 4) remain true when the operations $\vee$ and $\wedge$ are exchanged. When these four laws are added to the laws of a lattice then all the axioms of a **Boolean Algebra** are satisfied. As a summary we get the following.

Theorem 1.6 Boolean Algebras. *The structures*

$$(\mathbb{B}, \wedge, \vee, ^{-}, 0, 1),\ (\mathbb{B}, \vee, \wedge, ^{-}, 1, 0),\ (\mathbb{B}^n, \wedge, \vee, ^{-}, \mathbf{0}, \mathbf{1}),\ \textit{and}\ (\mathbb{B}^n, \vee, \wedge, ^{-}, \mathbf{1}, \mathbf{0})$$

are Boolean Algebras.

It is possible that different formulas describe the same function. This is already indicated by the transformation rules from above. It will be later on a problem to find formulas that satisfy a given criterion.

Up until now we reached the following point:

- a Boolean function can be defined by a table; and
- we can find the function table when a formula has been given.

We still need a possibility to find a formula for a function which is given by a table. We go back to Table 1.5 and consider the lines with the function value 1 for

$$f = (x_1 \vee x_2) \wedge \overline{(x_3 \oplus x_4)}\ .$$

Table 1.6: The disjunctive normal form of f

x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$	Conjunction
0	1	0	0	1	$\overline{x}_1\, x_2\, \overline{x}_3\, \overline{x}_4$
0	1	1	1	1	$\overline{x}_1\, x_2\, x_3\, x_4$
1	0	0	0	1	$x_1\, \overline{x}_2\, \overline{x}_3\, \overline{x}_4$
1	0	1	1	1	$x_1\, \overline{x}_2\, x_3\, x_4$
1	1	0	0	1	$x_1\, x_2\, \overline{x}_3\, \overline{x}_4$
1	1	1	1	1	$x_1\, x_2\, x_3\, x_4$

We use, for instance, the first row $(x_1x_2x_3x_4) = (0100)$ and assign a negated variable to the value 0 and a non-negated variable to the value 1. These (negated or non-negated) variables are combined by $\wedge$. For the given vector we get $\overline{x}_1x_2\overline{x}_3\overline{x}_4$. In this way the conjunction will result in the value 1 exactly for the given vector; for the other vectors of $\mathbb{B}^4$ the conjunction results in 0. This will be done for all the given lines of the table, the resulting six conjunctions are combined by $\vee$, the resulting disjunction of conjunctions is called the **disjunctive normal form**. It is *uniquely defined* except the order of the conjunctions. This, however, does not matter because of the commutativity of $\vee$. The other vectors which are not in the table result in $f = 0$. In this way we get a formula for the representation of a Boolean function which is given by Table 1.6:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) = {} & \overline{x}_1x_2\overline{x}_3\overline{x}_4 \vee \overline{x}_1x_2x_3x_4 \vee x_1\overline{x}_2\overline{x}_3\overline{x}_4 \vee \\ & x_1\overline{x}_2x_3x_4 \vee x_1x_2\overline{x}_3\overline{x}_4 \vee x_1x_2x_3x_4\ . \end{aligned} \tag{1.26}$$

Special denominations are sometimes necessary for the constant functions which are equal to 0 or 1 for all possible vectors $\mathbf{x}$. If it is necessary to show the variables that are important in a given context then we write $0(x_1, x_2, x_3, x_4)$ or $1(x_1, x_2, x_3, x_4)$; if the context is clear we might simply use $\mathbf{0}$ or $\mathbf{1}$ for these functions.

A second approach results in the **conjunctive normal form**. We select the binary vectors which are assigned to the function value 0 (see Tables 1.5 and 1.7). Here the non-negated variables are used for the value 0 in the vector, the negated variables are used when the component of the vector is equal to 1. The resulting literals are combined by $\vee$. Each disjunction is equal to 0 for the respective binary vector. The resulting disjunctions are combined by $\wedge$. Since the conjunction with 0 always results in 0, finally the conjunction of all these disjunctions produces all the values 0 of f. In this way we have built the **conjunctive normal form**. It is also *uniquely defined* except the order of the disjunctions:

$$
\begin{aligned}
f(x_1, x_2, x_3, x_4) = {} & (x_1 \vee x_2 \vee x_3 \vee x_4)(x_1 \vee x_2 \vee x_3 \vee \overline{x}_4)(x_1 \vee x_2 \vee \overline{x}_3 \vee x_4) \\
& (x_1 \vee x_2 \vee \overline{x}_3 \vee \overline{x}_4)(x_1 \vee \overline{x}_2 \vee x_3 \vee \overline{x}_4)(x_1 \vee \overline{x}_2 \vee \overline{x}_3 \vee x_4) \\
& (\overline{x}_1 \vee x_2 \vee x_3 \vee \overline{x}_4)(\overline{x}_1 \vee x_2 \vee \overline{x}_3 \vee x_4)(\overline{x}_1 \vee \overline{x}_2 \vee x_3 \vee \overline{x}_4) \\
& (\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3 \vee x_4) \; . \qquad (1.27)
\end{aligned}
$$

If there is a given Boolean function and we specify the value of *one* variable the result is a **subfunction**.

Table 1.7: The conjunctive normal form of f

x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$	Disjunction
0	0	0	0	0	$(x_1 \vee x_2 \vee x_3 \vee x_4)$
0	0	0	1	0	$(x_1 \vee x_2 \vee x_3 \vee \overline{x}_4)$
0	0	1	0	0	$(x_1 \vee x_2 \vee \overline{x}_3 \vee x_4)$
0	0	1	1	0	$(x_1 \vee x_2 \vee \overline{x}_3 \vee \overline{x}_4)$
0	1	0	1	0	$(x_1 \vee \overline{x}_2 \vee x_3 \vee \overline{x}_4)$
0	1	1	0	0	$(x_1 \vee \overline{x}_2 \vee \overline{x}_3 \vee x_4)$
1	0	0	1	0	$(\overline{x}_1 \vee x_2 \vee x_3 \vee \overline{x}_4)$
1	0	1	0	0	$(\overline{x}_1 \vee x_2 \vee \overline{x}_3 \vee x_4)$
1	1	0	1	0	$(\overline{x}_1 \vee \overline{x}_2 \vee x_3 \vee \overline{x}_4)$
1	1	1	0	0	$(\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3 \vee x_4)$

Definition 1.7 Cofactors. Let $f(x_1, x_2, \ldots, x_n) = f(x_i, \mathbf{x}_1)$ be a function of n variables. Then the *negative cofactor*

$$f^0_{x_i}(\mathbf{x}_1) = f(x_i = 0, \mathbf{x}_1) \qquad (1.28)$$

and the *positive cofactor*

$$f^1_{x_i}(\mathbf{x}_1) = f(x_i = 1, \mathbf{x}_1) \qquad (1.29)$$

are the two subfunctions with regard to x_i.

The resulting subfunctions do not depend on the respective variable anymore.

Example 1.8 Cofactors With Regard to x_1. We use the function (1.27) as an example and set $x_1 = 0$:

$$f^0_{x_1}(\mathbf{x}_1) = f(x_1 = 0, x_2, x_3, x_4) = (x_2 \vee x_3 \vee x_4)(x_2 \vee x_3 \vee \overline{x}_4)(x_2 \vee \overline{x}_3 \vee x_4)$$
$$(x_2 \vee \overline{x}_3 \vee \overline{x}_4)(\overline{x}_2 \vee x_3 \vee \overline{x}_4)(\overline{x}_2 \vee \overline{x}_3 \vee x_4) \ .$$

For $x_1 = 1$ the following function comes into existence:

$$f^1_{x_1}(\mathbf{x}_1) = f(x_1 = 1, x_2, x_3, x_4) = (x_2 \vee x_3 \vee \overline{x}_4)(x_2 \vee \overline{x}_3 \vee x_4)(\overline{x}_2 \vee x_3 \vee \overline{x}_4)$$
$$(\overline{x}_2 \vee \overline{x}_3 \vee x_4) \ .$$

It can be seen that some vectors of the function table are selected, all the vectors with a special value in the first component.

We go back to the disjunctive normal form (see (1.26)) and use the distributive law after the collection of the disjunctions of the normal form according to x_1:

$$f(x_1, x_2, x_3, x_4) = \overline{x}_1(x_2\overline{x}_3\overline{x}_4 \vee x_2x_3x_4) \vee$$
$$x_1(\overline{x}_2\overline{x}_3\overline{x}_4 \vee \overline{x}_2x_3x_4 \vee x_2\overline{x}_3\overline{x}_4 \vee x_2x_3x_4) \ . \tag{1.30}$$

This can be abbreviated by

$$f(x_1, x_2, x_3, x_4) = \overline{x}_1 f^0_{x_1} \vee x_1 f^1_{x_1} = \overline{x}_1 f^0_{x_1} \oplus x_1 f^1_{x_1} \tag{1.31}$$

and has been introduced by *C. Shannon*. $f^0_{x_1}$ and $f^1_{x_1}$ are the *two* cofactors from above. Each Boolean function can be decomposed using the following.

Theorem 1.9 Shannon Decomposition. *Each Boolean function $f(\mathbf{x}) = f(x_i, \mathbf{x}_1)$ can be represented by*

$$f(x_i, \mathbf{x}_1) = \overline{x}_i f(x_i = 0, \mathbf{x}_1) \vee x_i f(x_i = 1, \mathbf{x}_1) \tag{1.32}$$
$$= \overline{x}_i f(x_i = 0, \mathbf{x}_1) \oplus x_i f(x_i = 1, \mathbf{x}_1) \tag{1.33}$$

for any $x_i \in \mathbf{x}$, where $\mathbf{x}_1 = \mathbf{x} \setminus x_i$.

The cofactors $f^0_{x_1}(\mathbf{x}_1) = f^0_{x_1}(x_2, x_3, x_4)$ and $f^1_{x_1}(\mathbf{x}_1) = f^1_{x_1}(x_2, x_3, x_4)$ of (1.31) depend on x_2, x_3, and x_4. As a border case it can happen that $f^0_{x_1}(\mathbf{x}_1) = f^1_{x_1}(\mathbf{x}_1)$. This is an example for the *independence of a function of a variable* because we get step-by-step

$$\begin{aligned} f(x_1, x_2, x_3, x_4) &= \overline{x}_1 f^0_{x_1}(\mathbf{x}_1) \vee x_1 f^1_{x_1}(\mathbf{x}_1) \\ &= \overline{x}_1 f^0_{x_1}(\mathbf{x}_1) \vee x_1 f^0_{x_1}(\mathbf{x}_1) \\ &= (\overline{x}_1 \vee x_1) f^0_{x_1}(\mathbf{x}_1) \\ &= f^0_{x_1}(x_2, x_3, x_4) \ . \end{aligned} \tag{1.34}$$

This independence can also be seen when we go back to Table 1.4. Here we get for $f_3(x_1, x_2)$:

$$f_3(x_1, x_2) = x_1\overline{x}_2 \vee x_1x_2 = x_1(\overline{x}_2 \vee x_2) = x_1 \ .$$

We get a formula without x_2, nevertheless the function f_3 is a function of x_1 and x_2.

Functions also can be embedded into larger spaces using one additional variable. The reverse application of the Shannon decomposition realizes this expansion. We can assign the function $f(x_1, \ldots, x_n)$ of n variables to both the negative and the positive cofactors of a Shannon decomposition with regard to x_{n+1}

$$\begin{aligned} f(x_1, \ldots, x_n, x_{n+1} = 0) &= f(x_1, \ldots, x_n) \ , \\ f(x_1, \ldots, x_n, x_{n+1} = 1) &= f(x_1, \ldots, x_n) \ , \end{aligned}$$

and get as expansion

$$f(x_1, \ldots, x_n, x_{n+1}) = \overline{x}_{n+1} f(x_1, \ldots, x_n) \vee x_{n+1} f(x_1, \ldots, x_n) \tag{1.35}$$

the function $f(x_1, \ldots, x_n, x_{n+1})$ of $n + 1$ variables.

1.3 BOOLEAN RINGS

The algebraic structure of a ring requires an *addition* and a *multiplication*. For each Boolean space $\mathbb{B}^n$ there are two Boolean Rings. For the first Boolean Ring, as the addition the antivalence $\oplus$ will be taken, the conjunction $\wedge$ is used as multiplication. The second (dual) Boolean Ring uses as addition the equivalence $\odot$ and the disjunctions $\vee$ as multiplication. The following axioms must be satisfied.

1. **Commutativity:**

$$x_1 \oplus x_2 = x_2 \oplus x_1 \ , \tag{1.36}$$
$$x_1 \odot x_2 = x_2 \odot x_1 \ , \tag{1.37}$$

2. **Associativity:**

$$x_1 \oplus (x_2 \oplus x_3) = (x_1 \oplus x_2) \oplus x_3 = x_1 \oplus x_2 \oplus x_3 \ , \tag{1.38}$$
$$x_1 \odot (x_2 \odot x_3) = (x_1 \odot x_2) \odot x_3 = x_1 \odot x_2 \odot x_3 \ , \tag{1.39}$$

3. **Zero element:**

$$x \oplus 0 = x \ , \tag{1.40}$$
$$x \odot 1 = x \ , \tag{1.41}$$

4. **Inverse element:**

$$x \oplus x = 0 \ , \tag{1.42}$$
$$x \odot x = 1 \ , \tag{1.43}$$

5. **Unit element:**

$$x \wedge 1 = x \ , \tag{1.44}$$
$$x \vee 0 = x \ , \tag{1.45}$$

6. **Distributivity:**

$$x_1 \wedge (x_2 \oplus x_3) = (x_1 \wedge x_2) \oplus (x_1 \wedge x_3) \ , \tag{1.46}$$
$$x_1 \vee (x_2 \odot x_3) = (x_1 \vee x_2) \odot (x_1 \vee x_3) \ , \tag{1.47}$$

4. **Idempotence:**

$$x \wedge x = x \ , \tag{1.48}$$
$$x \vee x = x \ . \tag{1.49}$$

The idempotence characterizes *Boolean Rings*. The comparison of the upper line in above axioms (valid for the first Boolean Ring) and the associated line below (valid for the second Boolean Ring) show both the duality of the operations ($\wedge \Leftrightarrow \vee$) as well as ($\oplus \Leftrightarrow \odot$) and the opposite roles of the values 0 and 1.

Theorem 1.10 Boolean Rings. *The structures*

$$(\mathbb{B}, \oplus, \wedge, 0, 1),\ (\mathbb{B}, \odot, \vee, 1, 0),\ (\mathbb{B}^n, \oplus, \wedge, \mathbf{0}, \mathbf{1}),\ \textit{and}\ (\mathbb{B}^n, \odot, \vee, \mathbf{1}, \mathbf{0})$$

are Boolean Rings.

The rule (to be checked as before)

$$x_1 \vee x_2 = x_1 \oplus x_2 \oplus x_1\, x_2 \tag{1.50}$$

connects antivalence and disjunction. The complement can be eliminated by

$$\overline{x} = 1 \oplus x. \tag{1.51}$$

A next normal form, the **antivalence normal form**, can be constructed by means of the disjunctive normal form. Since two conjunctions are always different, it is possible to find at least one variable x_i which appears non-negated in one conjunction C_1 and negated in C_2:

$$C_1 = x_i C' \text{ and } C_2 = \overline{x}_i C''\ .$$

$C_1 = 1$ and $C_2 = 1$ hold for different vectors, therefore $C_1 \wedge C_2 = 0$; such conjunctions are **orthogonal** to each other. Subsequently this concept of **orthogonality** will also be used for sets of vectors.

The property $C_1\, C_2 = 0$ simplifies the rule (1.50)

$$C_1 \vee C_2 = C_1 \oplus C_2 \oplus C_1\, C_2$$

to

$$C_1 \vee C_2 = C_1 \oplus C_2$$

when the variables are replaced by conjunctions. This relationship is generalized by the following.

Theorem 1.11 Orthogonality of Conjunctions. *Let C_i, $i = 1, \ldots, k$, be k conjunctions. If*

$$C_i \wedge C_j = 0 \qquad \forall\, i \neq j,\ 1 \leq i, j \leq k \tag{1.52}$$

then

$$\bigvee_{i=1}^{k} C_i = \bigoplus_{i=1}^{k} C_i\ . \tag{1.53}$$

All conjunctions (minterms) of a disjunctive normal form are orthogonal to each other. Hence, due to Theorem 1.11 the disjunction sign of the disjunctive normal form (1.26) simply can be replaced by the antivalence without any change of the specified function:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) = {} & \overline{x}_1 x_2 \overline{x}_3 \overline{x}_4 \oplus \overline{x}_1 x_2 x_3 x_4 \oplus x_1 \overline{x}_2 \overline{x}_3 \overline{x}_4 \oplus \\ & x_1 \overline{x}_2 x_3 x_4 \oplus x_1 x_2 \overline{x}_3 \overline{x}_4 \oplus x_1 x_2 x_3 x_4 \ . \end{aligned} \tag{1.54}$$

This antivalence normal form is also *uniquely defined* except the order of the conjunctions.

Finally, in this normal form the negation can be eliminated by using the rule (1.51), and the distributivity law (1.46) will be applied thereafter. For instance:

$$\overline{x}_1 x_2 x_3 x_4 = (1 \oplus x_1) x_2 x_3 x_4 = x_2 x_3 x_4 \oplus x_1 x_2 x_3 x_4 \ .$$

Further simplifications might be possible with other conjunctions. If there are, for instance, two equal conjunctions then we get

$$C_1 \oplus C_1 = 0 \quad \text{and} \quad F \oplus 0 = F \ .$$

Hence, two equal conjunctions can be deleted.

In this way, the antivalence normal form (1.54) can be expressed by the positive polarity Reed-Muller polynomial:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) = {} & x_1 \oplus x_2 \oplus \\ & x_1 x_2 \oplus x_1 x_3 \oplus x_1 x_4 \oplus x_2 x_3 \oplus x_2 x_4 \oplus \\ & x_1 x_2 x_3 \oplus x_1 x_2 x_4 \ . \end{aligned} \tag{1.55}$$

It can be seen that the conjunctions consist of one variable, two variables, and three variables. It is a property of the transformed function $f(x_1, x_2, x_3, x_4)$ that the conjunction of four variables does not occur in the expression (1.55).

There is also the possibility to switch between Boolean Rings and Boolean Algebras by eliminating $\oplus$:

$$x_1 \oplus x_2 = \overline{x}_1 x_2 \vee x_1 \overline{x}_2 \ . \tag{1.56}$$

In order to transfer the conjunctive normal form into an **equivalence normal form**, we need the following two rules:

$$D_1 \wedge D_2 = D_1 \odot D_2 \odot (D_1 \vee D_2) \ , \tag{1.57}$$

$$\overline{x} = 0 \odot x \ . \tag{1.58}$$

In the same way as has been done for the antivalence normal form it can be concluded that for two different disjunctions D_i and D_j always $D_i \vee D_j = 1$ holds, and we get

$$D_1 \wedge D_2 = D_1 \odot D_2 \ ,$$

and generalized the

Theorem 1.12 Orthogonality of Disjunctions. *Let D_i, $i = 1, \ldots, k$, be k disjunctions. If*

$$D_i \vee D_j = 0 \qquad \forall\, i \neq j,\ 1 \leq i, j \leq k \tag{1.59}$$

then

$$\bigwedge_{i=1}^{k} D_i = \bigodot_{i=1}^{k} D_i \ . \tag{1.60}$$

Remaining negated variables can be eliminated using (1.58) and the distributive law (1.47).

This second Boolean Ring $(\mathbb{B}, \odot, \vee, 1, 0)$ has the same properties as the ring with the antivalence and the conjunction $(\mathbb{B}, \oplus, \wedge, 0, 1)$. Therefore only the first ring will be used.

1.4 BOOLEAN EQUATIONS AND INEQUALITIES

We start this complex by an example which is easy to understand and shows the problem.

Example 1.13 Solving a Boolean Equation. Let the equation

$$x_1 \vee x_2 = x_3 \wedge x_4 \tag{1.61}$$

be given. Since we are dealing with Boolean functions, the equality can only have the format

$$0 = 0 \quad \text{or} \quad 1 = 1 \ .$$

$x_1 \vee x_2 = 0$ holds for $x_1 = 0, x_2 = 0$. $x_3 \wedge x_4 = 0$ holds for $x_3 = 0,\ x_4 = 0;\ x_3 = 0,\ x_4 = 1;\ x_3 = 1,\ x_4 = 0;$ hence, we have the following set of solution vectors with the components (x_1, x_2, x_3, x_4):

$$\{(0000), (0001), (0010)\} \ .$$

Now the identity $1 = 1$ has to be explored, and, according to the definition of $\vee$ and $\wedge$ we get the following solutions:

$$\{(0111), (1011), (1111)\} \ .$$

Altogether, this equation has six solution vectors.

Definition 1.14 Boolean Equation. Let $\mathbf{x} = (x_1, \ldots, x_n)$, $f(\mathbf{x})$ and $g(\mathbf{x})$ be two Boolean functions, then

$$f(\mathbf{x}) = g(\mathbf{x}) \tag{1.62}$$

is a *Boolean equation* of n variables. The vector $\mathbf{b} = (b_1, \ldots, b_n)$ is a *solution* of this equation if $f(\mathbf{b}) = g(\mathbf{b})$ (i.e., $f(\mathbf{b}) = g(\mathbf{b}) = 0$ or $f(\mathbf{b}) = g(\mathbf{b}) = 1$).

It is easily possible to reduce the considerations to homogeneous equations.

Theorem 1.15 Homogeneous Equations. *The equation $f(\mathbf{x}) = g(\mathbf{x})$ is equivalent to the following two equations:*

$$\textit{homogeneous restrictive equation}: \qquad f(\mathbf{x}) \oplus g(\mathbf{x}) = 0\,, \tag{1.63}$$

$$\textit{homogeneous characteristic equation}: \qquad f(\mathbf{x}) \odot g(\mathbf{x}) = 1\,. \tag{1.64}$$

The original equation and the two homogeneous equations have the same solution set.

It is very easy to eliminate *inequalities*. The inequality

$$f(\mathbf{x}) \leq g(\mathbf{x}) \tag{1.65}$$

is equivalent to the two equations

$$f(\mathbf{x}) \wedge \overline{g(\mathbf{x})} = 0\,, \tag{1.66}$$

$$\overline{f(\mathbf{x})} \vee g(\mathbf{x}) = 1\,. \tag{1.67}$$

As done before, these equivalent transformations can be applied to binary vectors and particularly to Boolean functions.

1.5 LISTS OF TERNARY VECTORS (TVL)

The use of ternary vectors started in the 1970s by Zakrevskij [1975] and was further developed in order to create numerical methods for Boolean problems. They will be defined as a pooling of Boolean vectors.

Let two conjunctions $x_1\,x_2\,x_3\,x_4$ and $\overline{x}_1\,x_2\,x_3\,x_4$ be given; they are equal to 1 for the two binary vectors (1111) and (0111). This can be abbreviated by (−111). The symbol − can be replaced by 0 and by 1, the ternary vector represents two binary vectors.

In the background the laws of a Boolean Algebra are used:

$$x_1\,x_2\,x_3\,x_4 \vee \overline{x}_1\,x_2\,x_3\,x_4 = (x_1 \vee \overline{x}_1)\,x_2\,x_3\,x_4 = x_2\,x_3\,x_4\,. \tag{1.68}$$

The resulting conjunction is equal to 1 for $x_2 = x_3 = x_4 = 1$ and any value of x_1.

When a function is given by a disjunctive form (i.e., several conjunctions are combined by $\vee$), then each conjunction will be translated into a ternary vector, and the different ternary vectors are collected in a list or matrix (TVL).

Example 1.16 TVL of a Boolean Function in Disjunctive Form. The TVL

$$\mathrm{D}(f) = \begin{array}{ccc} x_1 & x_2 & x_3 \\ \hline 1 & - & 0 \\ - & 1 & 1 \\ 1 & 0 & 1 \\ \hline \end{array}$$

represents the disjunctive form of the function

$$f(x_1, x_2, x_3) = x_1\,\overline{x}_3 \vee x_2\,x_3 \vee x_1\,\overline{x}_2\,x_3\ .$$

The set of all Boolean vectors $\mathbf{x}$ with $f(\mathbf{x}) = 1$ is collected in this list.

When we look at the function $f(x_1, x_2, x_3, x_4) = x_1 \vee x_2 \vee x_3$ in $\mathbb{B}^4$ then we get in a first step three ternary vectors, the respective sets of Boolean vectors, however, overlap each other.

$$\mathrm{D}(f) = \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \\ \hline 1 & - & - & - \\ - & 1 & - & - \\ - & - & 1 & - \\ \hline \end{array} \tag{1.69}$$

This is an uncomfortable situation and can be avoided when we introduce the **orthogonality** of ternary vectors.

Definition 1.17 Orthogonality. Let $S(\mathbf{t})$ be the set of all $\mathbf{x} \in \mathbb{B}^n$ which can be generated by means of the ternary vector $\mathbf{t}$. Two vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ are *orthogonal* to each other ($\mathbf{t}_1 \perp \mathbf{t}_2$) if $S(\mathbf{t}_1) \cap S(\mathbf{t}_2) = \emptyset$.

Theorem 1.18 Orthogonality. *The property* $\mathbf{t}_1 \perp \mathbf{t}_2$ *holds if and only if for at least one component* i *the combination*

$$t_{1,i} = 0, t_{2,i} = 1 \quad \text{or} \quad t_{1,i} = 1, t_{2,i} = 0 \tag{1.70}$$

exists.

This very useful property can easily be established and tested. The orthogonal TVL

$$\mathrm{ODA}(f) = \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \\ \hline 1 & - & - & - \\ 0 & 1 & - & - \\ 0 & 0 & 1 & - \\ \hline \end{array} \tag{1.71}$$

generates the same set of Boolean vectors as (1.69). The form predicate ODA indicates that this TVL can, due to Theorem 1.11, alternatively be used to describe a Boolean function in disjunctive form $\mathrm{D}(f)$ or in antivalence form $\mathrm{A}(f)$.

SUMMARY

In this chapter we discussed very briefly some important terms of Boolean Algebras. The Boolean spaces $\mathbb{B}^n$ are built from the elements 0 and 1 or vectors of these elements. Boolean functions are unique mappings from $\mathbb{B}^n$ into $\mathbb{B}$. Boolean expressions of variables and operations describe in a compact manner Boolean functions. A Boolean equation will be built from two Boolean functions and has a set of Boolean vectors as solution. The Boolean Algebra is widely used to specify, design, analyze, and test digital circuits and systems.

EXERCISES

1.1 Use the table method to prove the associative law (1.4).

1.2 Determine the function table of the function:

$$f(x_1, x_2, x_3, x_4) = (x_1 \vee \overline{x}_2) \wedge ((x_1 \oplus x_3) \vee \overline{(x_2 \oplus x_4)}) \ . \tag{1.72}$$

1.3 A lattice of Boolean functions is specified by Table 1.8 which contains three don't-care values Φ.

(a) How many functions belong to this lattice?

(b) Describe each function of this lattice by a formula in disjunctive normal form.

(c) Apply the idempotence law (1.5) and the absorption law (1.8) to find simplified disjunctive forms of these functions.

(d) Find all pairs of functions of this lattice for which the conjunction results in the minimum and the disjunction in the maximum of all these functions.

(e) Which function of this lattice has the simplest formula in disjunctive form?

Table 1.8: Incompletely specified function that generates a lattice of Boolean functions

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	Φ
1	0	0	0
1	0	1	Φ
1	1	0	Φ
1	1	1	1

1.4 Which binary vectors belong to the solution of the Boolean equation:

$$(x_1 \vee \overline{x}_3) \wedge x_2 = (x_3 \oplus x_4) \vee x_1 \ . \tag{1.73}$$

Specify the solution set S:

(a) by a list of binary vectors (BVL), and

(b) by an orthogonal list of ternary vectors (TVL).

1.5 The following TVL is not orthogonal.

x_1	x_2	x_3	x_4
1	–	0	0
–	1	1	–
0	–	0	–
–	–	1	1

(1.74)

(a) Which ternary vectors of the TVL (1.74) are not orthogonal to each other?

(b) Assume that the TVL (1.74) represents a function in disjunctive form. Construct the associated ODA-form.

(c) Assume that the TVL (1.74) represents a function in antivalence form. Construct the associated ODA-form.