

Boolean Algebra

D. Leeuw

6 december 2023

v.1.1.0



© 2023 Dennis Leeuw

Dit werk is uitgegeven onder de Creative Commons BY-NC-SA Licentie en laat anderen toe het werk te kopiëren, distribueren, vertonen, op te voeren, en om afgeleid materiaal te maken, zolang de auteurs en uitgever worden vermeld als maker van het werk, het werk niet commercieel gebruikt wordt en afgeleide werken onder identieke voorwaarden worden verspreid.

Over dit Document

0.1 Leerdoelen

Dit document geeft een basis uitleg over de boolean algebra in relatie tot de elektrotechniek. De verschillende Boolean functies worden beschreven aan de hand van elektronische schema's. De lezer leert dan ook wat boolean algebra is in de context van de elektrotechniek.

0.2 Voorkennis

- Om de uitleg te kunnen volgen die beschreven is bij de elektronische werking van de verschillende functies dient de lezer kennis te hebben van de werking van transistoren, weerstanden, spanning en stroom
- Het is handig als de lezer enigzins vertrouwd is met het lezen van stroomschema's

Inhoudsopgave

Over dit Document	i
0.1 Leerdoelen	i
0.2 Voorkennis	i
1 Introduction	1
2 NOT - Inverter	3
3 AND	5
4 NAND	7
5 OR	9
6 NOR	11
7 XOR	13
8 XNOR	15
9 Praktijkopdrachten	17
Index	19

Hoofdstuk 1

Introduction

In dit hoofdstuk gaan we een aantal van de hiervoor beschreven componenten gebruiken om computer logica op te bouwen. Kortom hoe maken we van elektrische spanningen logische circuits. De booleanse algebra, een tak van de wiskunde, onderscheidt zich van de rest van de wiskunde doordat de booleanse algebra alleen gebruik maakt van true of false, of zoals we in de computer techniek zeggen 1 en 0.

In de computertechniek gebruiken we de transistor als schakelaar. De basis van de transistor is de input. Door deze input te gebruiken schakelen we de transistor en komt er een bepaalde waarde (1 of 0) aan de output.

De gebruikte spanningen in de computer zijn 3,3 Vdc of 5 Vdc en dat is dan een 1. De 0 is staat gelijk aan geen spanning.

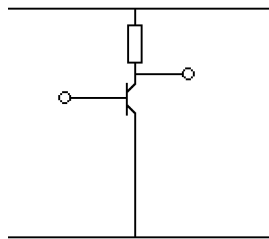
De booleanse algebra wordt ook gebruikt bij het programmeren van computers. Dan staan de waarden 1 of 0 vaak voor true of false. Een vergelijking is waar of niet waar. We zullen hier niet al te doen op ingaan. Wel belangrijk is het om te weten dat je een 1 dus kan lezen als waar en een 0 als niet waar.

Hoofdstuk 2

NOT - Inverter

De NOT werkt als een inverter. Dus als er een 1 op de ingang wordt aangeboden dan staat er een 0 op de uitgang en omgekeerd. De waarheidstabel voor de inverter is dus heel eenvoudig:

De NOT wordt gebouwd door gebruik van een enkele transistor en een weerstand (figuur 2.1).



Figuur 2.1: NOT circuit

Input	Output
0	1
1	0

Het symbool voor de NOT is weergegeven in figuur 2.2



Figuur 2.2: Symbool van een NOT

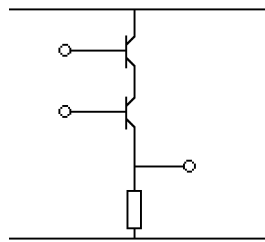
In de wiskunde of in programmeertalen kan je de not tegenkomen met de volgende symbolen: \neg !

Hoofdstuk 3

AND

De AND geeft aan dat wanneer beide waarden 1 zijn ook de uitkomst 1 is.

De AND wordt gebouwd door gebruik te maken van twee transistoren waarvan de beide basis de ingang vormen en een weerstand (figuur 3.1).



Figuur 3.1: AND circuit

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Het symbool in een stroomschema voor de AND is weergegeven in figuur 3.2



Figuur 3.2: Symbool van een AND

In wiskundige notaties en bij het programmeren kunnen de volgende symbolen gebruikt worden om een AND aan te geven (afhankelijk van de programmeertaal): \wedge & && ·

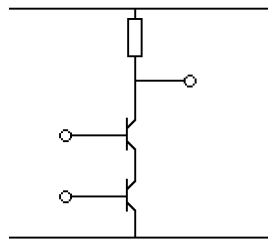
Hoofdstuk 4

NAND

De NAND geeft een negatief resultaat als beide ingangen 1 zijn, de uitgang is dan 0.

De NAND-technologie kan je tegen komen in flash memory.

De NAND wordt gebouwd door gebruik te maken van twee transistoren waarvan de beide basis de ingang vormen en een weerstand (figuur 4.1).



Figuur 4.1: NAND circuit

Input 1	Input 2	Output
0	0	1
0	1	1
1	0	1
1	1	0

Het symbool voor de NAND is weergegeven in figuur 4.2



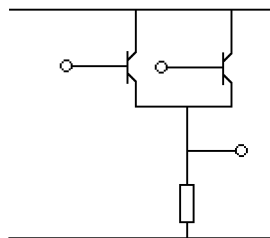
Figuur 4.2: Symbool van een NAND

Hoofdstuk 5

OR

De OR geeft aan of één of beide ingangen 1 zijn, als dat het geval is dan is de uitgang 1.

De OR wordt gebouwd door gebruik te maken van twee transistoren die parallel aan elkaar staan, de beide basis vormen de ingang en er is een weerstand (figuur 5.1).



Figuur 5.1: OR circuit

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

Het symbool voor de OR is weergegeven in figuur 5.2



Figuur 5.2: Symbool van een OR

In de wiskunde of in programmeertalen kan je de OR tegen komen met de volgende symbolen: \vee \parallel

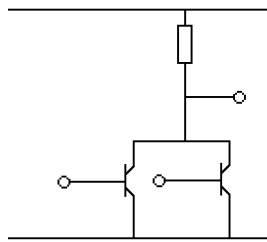
Hoofdstuk 6

NOR

De NOR geeft aan dat beide ingangen 0 zijn, als dat het geval is dan is de uitgang 1.

De NOR-technologie kan je tegen komen in flash memory.

De NOR wordt gebouwd door gebruik te maken van twee transistoren die parallel aan elkaar staan, de beide basis vormen de ingang en er is een weerstand (figuur 6.1).



Figuur 6.1: NOR circuit

Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	0

Het symbool voor de NOR is weergegeven in figuur 6.2



Figuur 6.2: Symbool van een NOR

Hoofdstuk 7

XOR

De XOR geeft alleen als één van beide ingangen 1 is een 1 op de uitgang. De XOR functie wordt veel gebruikt in de cryptografie.

Er zijn vele manieren om een XOR te bouwen, er is dan ook geen circuit opgenomen. Zie Wikipedia voor mogelijke oplossingen.

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

Het symbool voor de XOR is weergegeven in figuur 7.1



Figuur 7.1: Symbool van een XOR

In de wiskunde of in programmeertalen kan je de XOR tegenkomen met de volgende symbolen: \oplus

Hoofdstuk 8

XNOR

De XNOR geeft als één van beide ingangen 1 is en de andere 0 een 0 op de uitgang.

Er zijn vele manieren om een XNOR te bouwen, er is dan ook geen circuit opgenomen. Zie Wikipedia voor mogelijke oplossingen.

Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	1

Het symbool voor de XNOR is weergegeven in figuur 8.1



Figuur 8.1: Symbool van een XNOR

Hoofdstuk 9

Praktijkopdrachten

Met een breadboard kunnen de verschillende boolean functies nagebouwd worden. Sluit op de uitgang van de transistor een weerstand en een led aan en kies voor de basis van elke transistor een juiste weerstand en spanning om een 1 aan te kunnen bieden.

Index

AND, [5](#)

Booleanse algebra, [1](#)

Cryptografie, [13](#)

Inverter, [3](#)

NAND, [7](#)

NOR, [11](#)

NOT, [3](#)

OR, [9](#)

XNOR, [15](#)

XOR, [13](#)