

2 | Digitale techniek

Poortlogica

De ICT-wereld kan niet zonder *logica*. Het is de **booleaanse logica** die een computersysteem doet draaien. Computers rekenen immers 'binair' en beslissen 'booleaans'. Elke computergebruiker maakt dus - zonder het te beseffen - gebruik van de fascinerende algebra van George Boole^[8].

2.1 Logische functies

2.1.1 Algebra van Boole

De Brit George Boole ontwikkelde in het midden van de 19e eeuw een **booleaanse algebra** in een poging om via algebraïsche technieken met *logische uitdrukkingen* te kunnen experimenteren. Booleaanse algebra passen we onder meer toe bij het samenstellen van digitale elektronische schakelingen, zoals die in computers worden gebruikt. In de praktijk kunnen we de werking ervan zien in sommige zoeksystemen (webbrowsers) voor internetpagina's. Boole-algebra maakt gebruik van (ten minste) twee elementen: onwaar (FALSE) en waar (TRUE). We spreken dan ook wel van *twee-elementen-algebra*. Ze voorziet in twee binaire bewerkingen (AND en OR) en een unaire bewerking (NOT).

2.1.2 Waarheidstabel

Een **waarheidstabel** (truth tabel) is een wiskundige tabel die in het begin van de 20^e eeuw werd ontwikkeld. We gebruiken waarheidstabellen om te beslissen of een logische uitdrukking al dan niet waar is. Ze zijn nuttig om de waarheidwaarden weer te geven van logische operatoren (AND, OR, NOT) bij alle combinaties van de ingangsvariabelen.

[8] De Brit George Boole genoot zelf weinig onderwijs, maar hij slaagde erin om zichzelf op te werken tot een geslaagd wiskundige. Rond 1849 ontwikkelde hij zijn Booleaanse logica, de basis van de moderne digitale computerlogica.

Let op!

In waarheidstabellen kunnen we de waarheid of onwaarheid van een bewering op verschillende manieren aanduiden:

- Meestal gebruiken we een **T** voor *true* (waar) en **F** voor *false* (onwaar).
- Ook gebruiken we de **1** voor *waar* en **0** voor *onwaar*.

Een nadeel van waarheidstabellen is dat bij beweringen met veel variabelen de tabel snel te groot wordt. De grootte van de tabel groeit immers exponentieel met het aantal variabelen. Figuur 2.1 toont een waarheidstabel van een praktisch voorbeeld. Stel dat we willen kamperen tijdens de vakantie bij goed weer. We definiëren 2 onafhankelijke ingangsvariabelen: 'vrij zijn' en 'het regent niet'. De bewering: 'we gaan kamperen' is de afhankelijke uitgangsgrootheid. Zowel voor de beslissing (uitgangsvariabele) als voor de ingangsvariabelen zijn er slechts 2 mogelijkheden: het zijn daarom **binaire elementen**.

	ingangs- variabele 1	ingangs- variabele 2	beslissing
combinaties	'vrij zijn'	'het regent niet'	'we gaan kamperen'
1	onwaar	onwaar	onwaar
2	onwaar	waar	onwaar
3	waar	onwaar	onwaar
4	waar	waar	waar

△ Fig.2.1 | Waarheidstabel van een illustratief voorbeeld.
Bron: W.Van Wichelen

Dank zij de booleaanse schrijfwijze kunnen we de waarheidstabel van figuur 2.1 eenvoudiger opschrijven. We maken volgende afspraken:

- V = 'vrij zijn'
- N = 'het regent niet'
- K = 'we gaan kamperen'

combinaties	V	N	K
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

△ Fig.2.2 | Gebruikelijke voorstelling van een waarheidstabel.
Bron: W.Van Wichelen

2.1.3 Positieve logica

Elke logische bewering is een combinatie van slechts 4 basisfuncties: de **JA**-functie, de **NIET**-functie, de **EN**-functie en de **OF**-functie. Voor de praktische realisatie moeten we beschikken over logische 'nullen' en 'enen'. In de digitale techniek worden logische enen en nullen voorgesteld door twee verschillende *spanningswaarden*.

Definitie

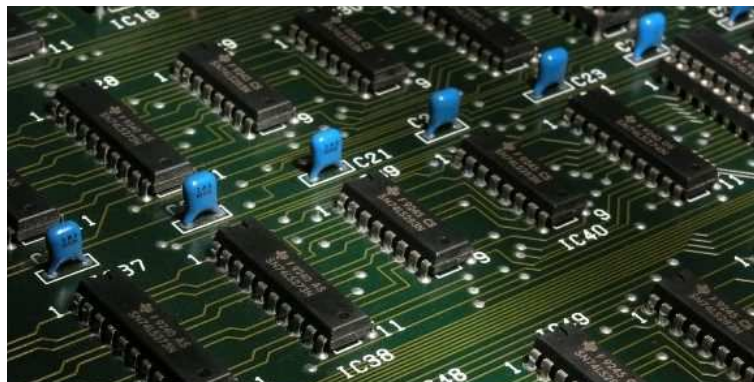
Wanneer de *meest positieve* spanning een 'logische één' voorstelt en de *minst positieve* spanning een 'logische nul' voorstelt, dan spreken we van **positieve logica**.

Positieve logica wordt het meest toegepast. Negatieve logica zal je niet veel tegenkomen. Later zullen we digitale schakelingen realiseren met digitale IC's^[9] die we voorzien van een voedingsspanning van +5 volt ten opzichte van massapotentiaal. Een gevolg hier van is dat een logische 1 overeenkomt met +5 volt en een logische 0 met 0 volt.

Let op!

In datasheets (gegevensbladen) vinden we meestal volgende benamingen terug: 'high' (**H**) en 'low' (**L**).

- hoog niveau (1) = H = +5 volt
- laag niveau (0) = L = 0 volt



△ Fig.2.3 | Familieportret van digitale IC's op een printkaart.
Bron: <https://gadgets.ndtv.com>

[9] Een geïntegreerde schakeling (IC, Integrated Circuit) bestaat uit verschillende elektronische componenten (transistors) op een enkel stuk halfgeleidermateriaal. Het eerste IC werd ontwikkeld in 1958. Tegenwoordig zitten er miljoenen transistoren in een IC van enkele vierkante millimeters groot...

2.2 Basispoorten

In de digitale elektronica zijn **logische poorten** bouwstenen die werken volgens de *booleaanse logica*. Ze werken met 2 signaalspanningen die kunnen worden geïnterpreteerd als één en nul. Digitale poorten kunnen één of meerdere ingangen hebben maar ze bezitten slechts één uitgang. Deze logische bouwstenen zijn voornamelijk opgebouwd uit transistoren, weerstanden en dioden. Digitale elektronica gebruikt van origine 5 volt-logica (TTL^[10]). Tegenwoordig gebruiken we, omwille van de energiezuinigheid, logica die gebruikmaakt van 3,3 volt. Bij CPU's hanteren we nog lagere spanningen zoals 1,8 volt.

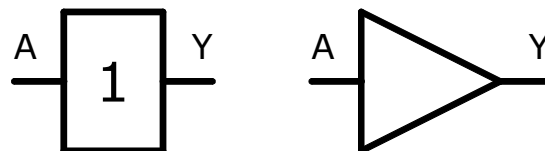
2.2.1 Ja-poort

Definitie

Een **JA-poort** of een *buffer* is een digitale poortschakeling (gate) met één ingang (A) en één uitgang (Y). De uitgang neemt steeds de toestand over van de ingang.

De JA-poort voert dus in feite geen logische functie uit. Deze poort gaan we inzetten als stroomversterkend element en om ingangen en uitgangen van elkaar te scheiden.

Symbool



△ Fig.2.4 | Links: Europees symbool voor de JA-poort
Rechts: Amerikaans symbool voor de JA-poort
Bron: W.Van Wichelen

[10] TTL-logica of 'transistor-transistorlogica' is een standaard voor digitale logica en werkt met geïntegreerde schakelingen (IC's) waarvan het typenummer meestal begint met '74'.

Waarheidstabel

	input	output
combinatie	A	Y
1	0	0
2	1	1

	input	output
combinatie	A	Y
1	L	L
2	H	H

△ Fig.2.5 | Waarheidstabel voor een JA-poort
Bron: W.Van Wichelen

Logische vergelijking

Booleaanse vergelijking van een **JA-poort**:

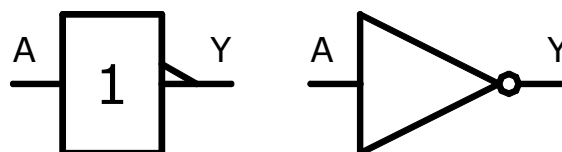
$$Y = A$$

2.2.2 NOT-poort

Definitie

Een **NOT-poort** is een digitale poortschakeling met één ingang (A) en één uitgang (Y). De uitgang neemt steeds de *inverse* toestand over van de ingang. Deze poort wordt dan ook wel *inverter* genoemd.

Symbool



△ Fig.2.6 | Links: Europees symbool voor de NOT-poort
Rechts: Amerikaans symbool voor de NOT-poort
Bron: W.Van Wichelen

Waarheidstabel

	input	output
combinatie	A	Y
1	0	1
2	1	0

	input	output
combinatie	A	Y
1	L	H
2	H	L

△ Fig.2.7 | Waarheidstabel voor een NOT-poort
Bron: W.Van Wichelen

Logische vergelijking

Booleaanse vergelijking van een **NOT-poort**:

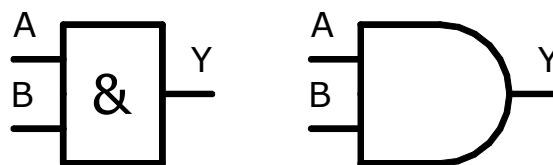
$$Y = \bar{A}$$

2.2.3 AND-poort

Definitie

Een **AND-poort** is een digitale poortschakeling met één uitgang en meerdere ingangen. Enkel wanneer *alle* ingangen hoog zijn, genereert deze functie een hoge uitgang. Van zodra één ingang laag is, zal de uitgang ook laag worden.

Symbool



△ Fig.2.8 | Links: Europees symbool voor de AND-poort
Rechts: Amerikaans symbool voor de AND-poort
Bron: W.Van Wichelen

Waarheidstabel

	input 1	input 2	output
comb.	A	B	Y
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

	input 1	input 2	output
comb.	A	B	Y
1	L	L	L
2	L	H	L
3	H	L	L
4	H	H	H

△ Fig.2.9 | Waarheidstabel voor een AND-poort met 2 ingangen. Bron: W.Van Wichelen

Logische vergelijking

Booleaanse vergelijking van een **AND-poort** (2 ingangen):

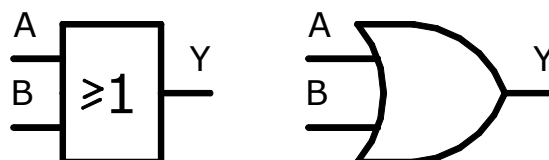
$$Y = A \cdot B$$

2.2.4 OR-poort

Definitie

Een **OR-poort** is een digitale poortschakeling met één uitgang en meerdere ingangen. Van zodra er één ingang hoog is, zal de uitgang ook hoog worden. De uitgang is alleen laag wanneer *alle* ingangen laag zijn.

Symbool



△ Fig.2.10 | Links: Europees symbool voor de OR-poort
Rechts: Amerikaans symbool voor de OR-poort
Bron: W.Van Wichelen

Waarheidstabel

	input 1	input 2	output
comb.	A	B	Y
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

	input 1	input 2	output
comb.	A	B	Y
1	L	L	L
2	L	H	H
3	H	L	H
4	H	H	H

△ Fig.2.11 | Waarheidstabel voor een OR-poort met 2 ingangen.
Bron: W.Van Wichelen

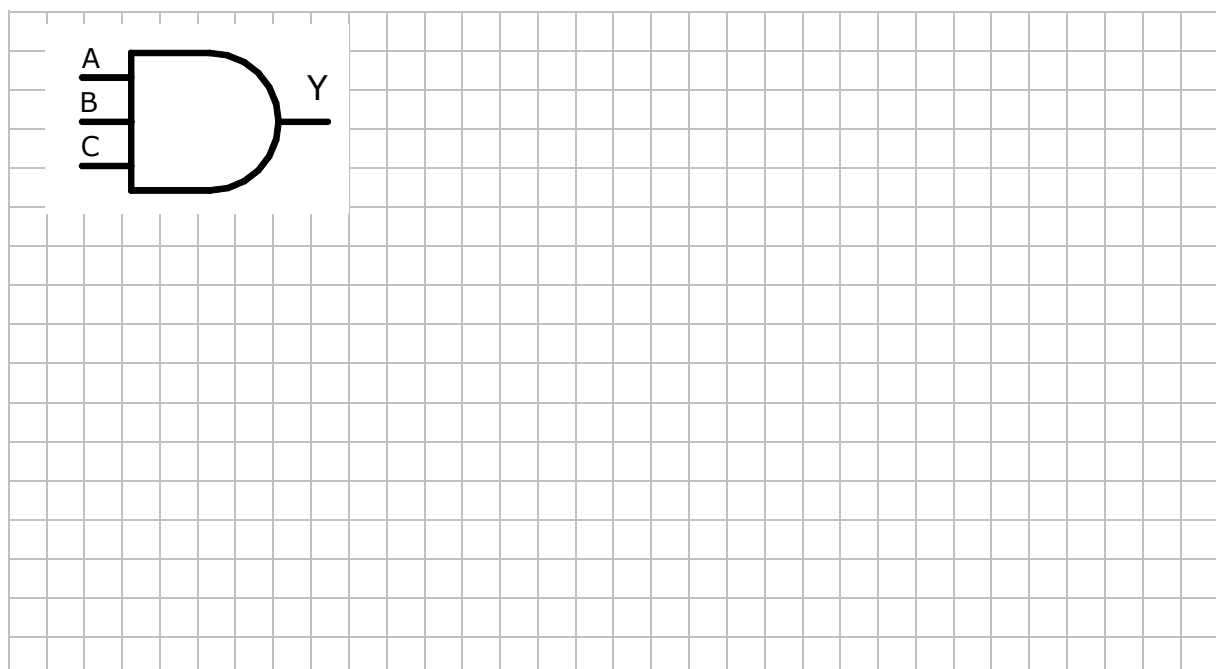
Logische vergelijking

Booleaanse vergelijking van een **OR-poort** (2 ingangen):

$$Y = A + B$$

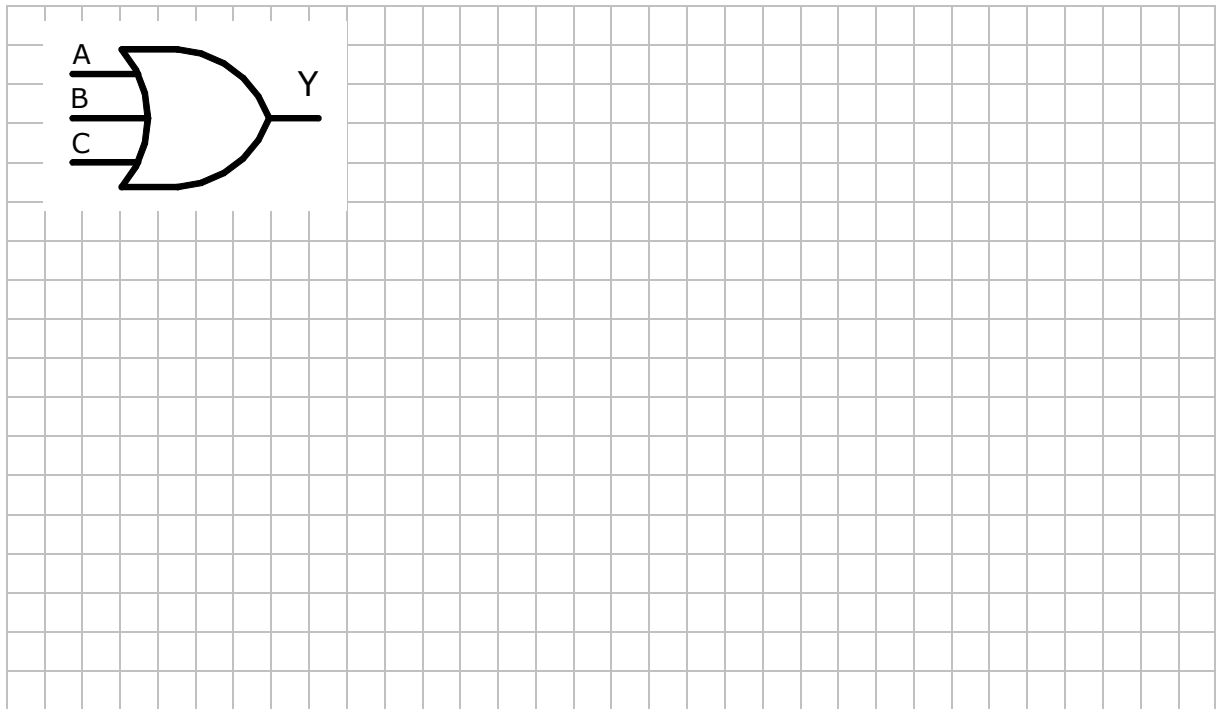
2.2.5 Oefeningen

1. Bestudeer onderstaande logische poort.
 - a. Stel de waarheidstabel op.
 - b. Schrijf de logische vergelijking op.



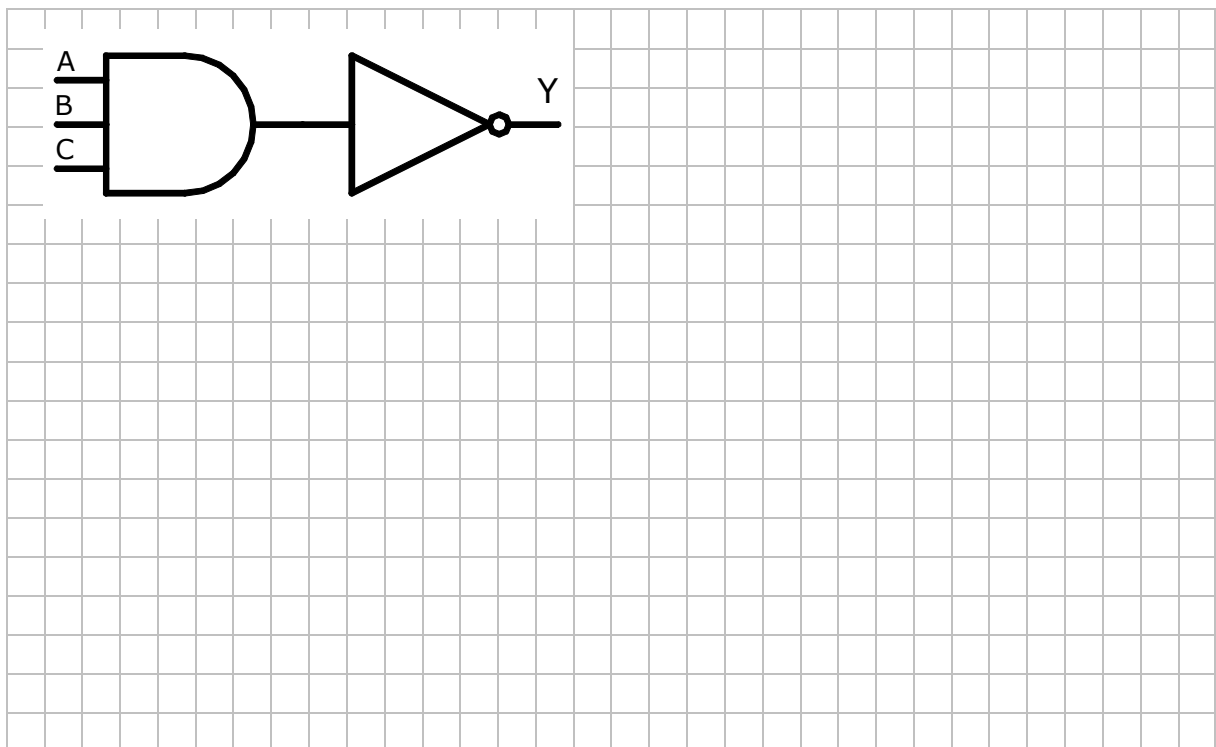
2. Bestudeer onderstaande logische poort.

- Stel de waarheidstabel op.
- Schrijf de logische vergelijking op.

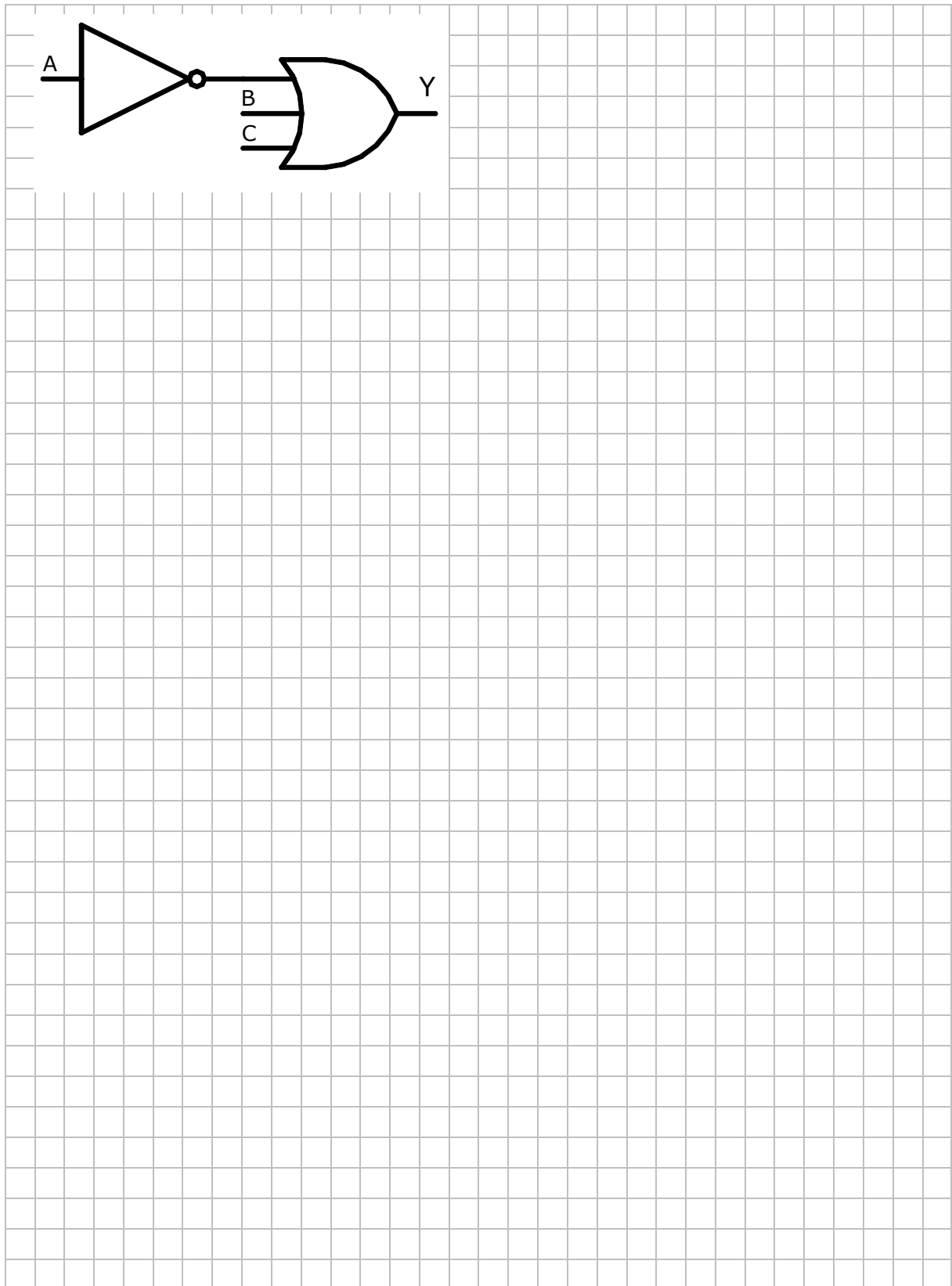


3. Bestudeer onderstaande logische poortschakeling.

- Stel de waarheidstabel op.
- Schrijf de logische vergelijking op.



4. Bestudeer onderstaande logische poortschakeling.
- Stel de waarheidstabel op.
 - Schrijf de logische vergelijking op.



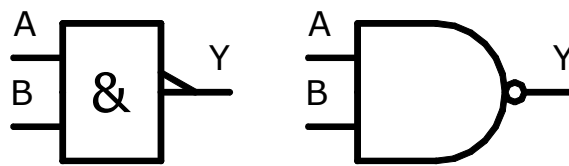
2.3 Afgeleide poorten

2.3.1 NAND-poort

Definitie

Een **NAND-poort** is een digitale poortschakeling met één uitgang en meerdere ingangen. De uitgang is altijd hoog behalve wanneer *alle* ingangen hoog zijn: dan wordt de uitgang laag.

Symbool



△ Fig.2.12 | Links: Europees symbool voor de NAND-poort
Rechts: Amerikaans symbool voor de NAND-poort
Bron: W.Van Wichelen

Waarheidstabel

	input 1	input 2	output
comb.	A	B	Y
1	0	0	1
2	0	1	1
3	1	0	1
4	1	1	0

	input 1	input 2	output
comb.	A	B	Y
1	L	L	H
2	L	H	H
3	H	L	H
4	H	H	L

△ Fig.2.13 | Waarheidstabel voor een NAND-poort met 2 ingangen. Bron: W.Van Wichelen

Logische vergelijking

Booleaanse vergelijking van een **NAND-poort** (2 ingangen):

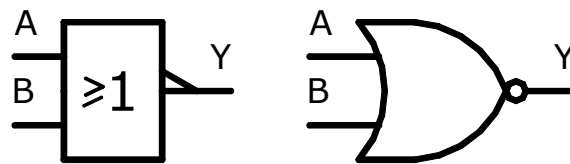
$$Y = \overline{A \cdot B}$$

2.3.2 NOR-poort

Definitie

Een **NOR-poort** is een digitale poortschakeling met één uitgang en meerdere ingangen. De uitgang is alleen hoog wanneer *alle* ingangen laag zijn. Bij alle andere ingangscombinaties is de uitgang laag.

Symbol



△ Fig.2.14 | Links: Europees symbool voor de NOR-poort
Rechts: Amerikaans symbool voor de NOR-poort
Bron: W.Van Wichelen

Waarheidstabel

	input 1	input 2	output
comb.	A	B	Y
1	0	0	1
2	0	1	0
3	1	0	0
4	1	1	0

	input 1	input 2	output
comb.	A	B	Y
1	L	L	H
2	L	H	L
3	H	L	L
4	H	H	L

△ Fig.2.15 | Waarheidstabel voor een NOR-poort met 2 ingangen. Bron: W.Van Wichelen

Logische vergelijking

Booleaanse vergelijking van een **NOR-poort** (2 ingangen):

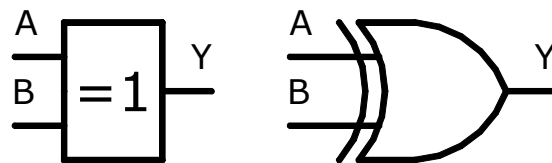
$$Y = \overline{A + B}$$

2.3.3 EXOR-poort

Definitie

Een **EXOR-poort** (exclusive OR) is digitale poortschakeling met één uitgang en twee ingangen. De uitgang is alleen hoog wanneer de ingangen *verschillend* zijn van elkaar. Bij gelijke ingangen is de uitgang laag.

Symbool



△ Fig.2.15 | Links: Europees symbool voor de EXOR-poort
Rechts: Amerikaans symbool voor de EXOR-poort
Bron: W.Van Wichelen

Waarheidstabel

	input 1	input 2	output
comb.	A	B	Y
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

	input 1	input 2	output
comb.	A	B	Y
1	L	L	L
2	L	H	H
3	H	L	H
4	H	H	L

△ Fig.2.16 | Waarheidstabel voor een EXOR-poort.
Bron: W.Van Wichelen

Logische vergelijking

Booleaanse vergelijking van een **EXOR-poort**:

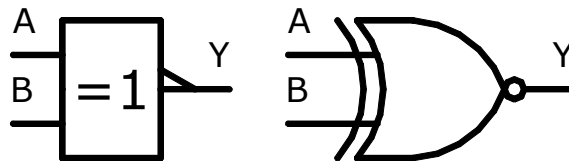
$$Y = \bar{A} \cdot B + A \cdot \bar{B}$$

2.3.4 EXNOR-poort

Definitie

Een **EXNOR-poort** (exclusive NOR) is digitale poortschakeling met één uitgang en twee ingangen. De uitgang is alleen hoog wanneer de ingangen *gelijk* zijn van elkaar. Bij ongelijke ingangen is de uitgang laag.

Symbol



△ Fig.2.17 | Links: Europees symbool voor de EXNOR-poort
Rechts: Amerikaans symbool voor de EXNOR-poort
Bron: W.Van Wichelen

Waarheidstabel

	input 1	input 2	output
comb.	A	B	Y
1	0	0	1
2	0	1	0
3	1	0	0
4	1	1	1

	input 1	input 2	output
comb.	A	B	Y
1	L	L	H
2	L	H	L
3	H	L	L
4	H	H	H

△ Fig.2.18 | Waarheidstabel voor een EXNOR-poort.
Bron: W.Van Wichelen

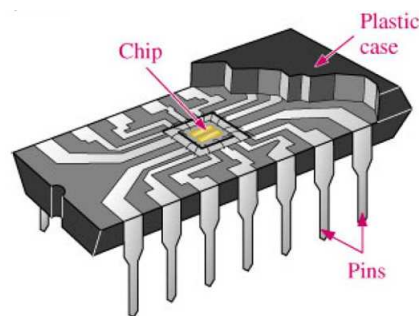
Logische vergelijking

Booleaanse vergelijking van een **EXNOR-poort**:

$$Y = A \cdot B + \bar{A} \cdot \bar{B}$$

2.3.5 IC-behuizingen

Digitale logische poorten brengt men op de markt in de vorm van geïntegreerde schakelingen (IC's). De schakelelektronica is ondergebracht op een chip van enkele mm² groot. De chip wordt via gouden draadjes verbonden met de pennen (pins) van de behuizing (package). Figuur 2.19 toont een doorsnede van een IC. Een veel voorkomende behuizing voor digitale bouwstenen is de 'dual in line package' (DIP). Dit type IC bezit 2 rijen aansluitpennen die mooi op een zekere afstand achter elkaar staan.



△ Fig.2.19 | Doorsnede van een IC met DIP-behuizing.
Bron: <https://slideplayer.com>

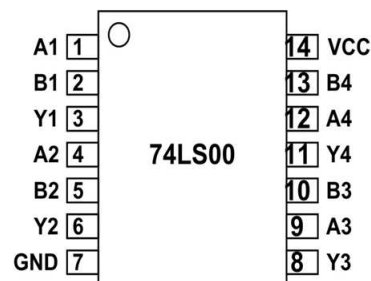
2.3.6 IC-basispoorten

De tabel van figuur 2.20 toont veel gebruikte basispoorten. Digitale IC's zijn ingedeeld in *families* volgens hun toegepaste technologie. Tot de populairste familie behoren de IC's die gebruik maken van 'Transistor Transistor Logic' (**TTL**). Deze IC's met type-nummer 74XX, zijn opgebouwd met klassieke *bipolaire junctie-transistoren*. Binnen de TTL-serie zit nog een interessante sub-familie: de *Low Power Schotky* die beter presteert dan de gewone TTL 74-reeks. IC's die bestaan uit *veldeffecttransistoren* behoren tot de zogenaamde **CMOS**-familie. Deze IC's verbruiken praktisch geen stroom maar ze zijn erg gevoelig voor statische ontladingen.

TABEL BASISPOORTEN

Poort	aantal ingangen	aantal poorten per IC	Low Power Schottky TTL	CMOS	High Speed CMOS
BUFFER	1	6	74LS07	4010	74HC4050
NOT	1	6	74LS04	4069	74HC04
AND	2	4	74LS08	4081	74HC08
	3	3	74LS11	4073	74HC11
	4	2	74LS21	4082	74HC21
OR	2	4	74LS32	4071	74HC32
	3	3		4075	74HC75
	4	2		4072	
NAND	2	4	74LS00	4011	74HC00
	3	3	74LS10	4023	74HC10
	4	2	74LS20	4012	74HC20
NOR	2	4	74LS02	4001	74HC02
	3	3	74LS27	4025	74HC27
	4	2	74LS23	4002	74HC002
EXOR	2	4	74LS86	4030	74HC86
EXNOR	2	4		4077	74HC266

△ Fig.2.20 | Overzicht van veel gebruikte basispoorten.
Bron: W.Van Wichelen

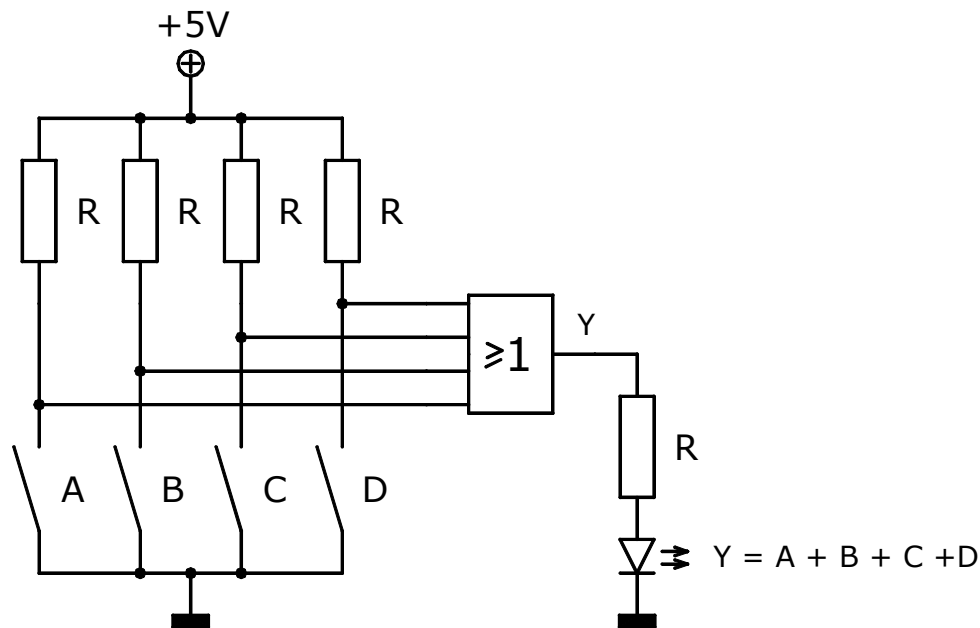


△ Fig.2.21 | Een low power TTL NAND-poort in vol ornaat met bijhorende pinout. Het 'bolletje' geeft pin 1 aan.
Bron: <https://components101.com>

2.3.7 Toepassingen

Controleschakeling met OR-poort

De schakeling van figuur 2.22 verwittigt de autobestuurder wanneer één of meerdere portieren niet goed gesloten zijn. Deursensoren A, B, C en D zijn drukschakelaars die een hoog niveau genereren in de geopende stand en een laag niveau in de geloten stand. Een signaallampje (LED) op het dashboard licht op indien minstens één portier open staat.



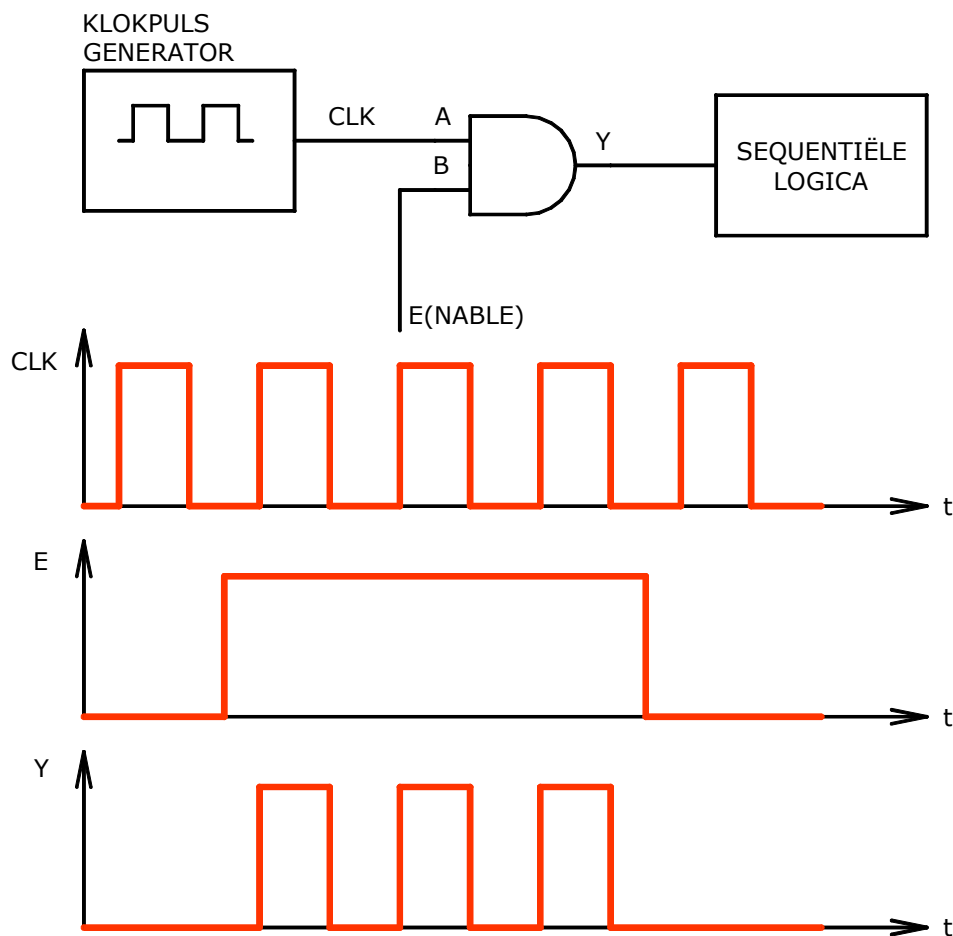
△ Fig.2.22 | OR-gate met 4 ingangen aan het werk...
Bron: W.Van Wichelen



△ Fig.2.23 | Signalering van een open portier op het dashboard van een wagen. Bron: <https://westpacparts.com>

AND-functie als blokkeerpoort

Een AND-poort kan op een gemakkelijke manier ingezet worden als *controlepoort*. De schakeling van figuur 2.24 laat toe dat een signaal al of niet doorgeschakeld kan worden. In ons voorbeeld levert een klokpulsgenerator de nodige synchronisatiepuls voor de sequentiële logica. Zolang de E-ingang^[11] laag gehouden wordt, zal de AND-poort blokkeren en het klokpulssignaal tegenhouden zodat de digitale logica niet kan werken. Maken we de E-ingang hoog dan worden de klokpulsen wel doorgeschakeld. De sequentiële logica wordt nu in staat gesteld om bijvoorbeeld het nodige telwerk te doen.



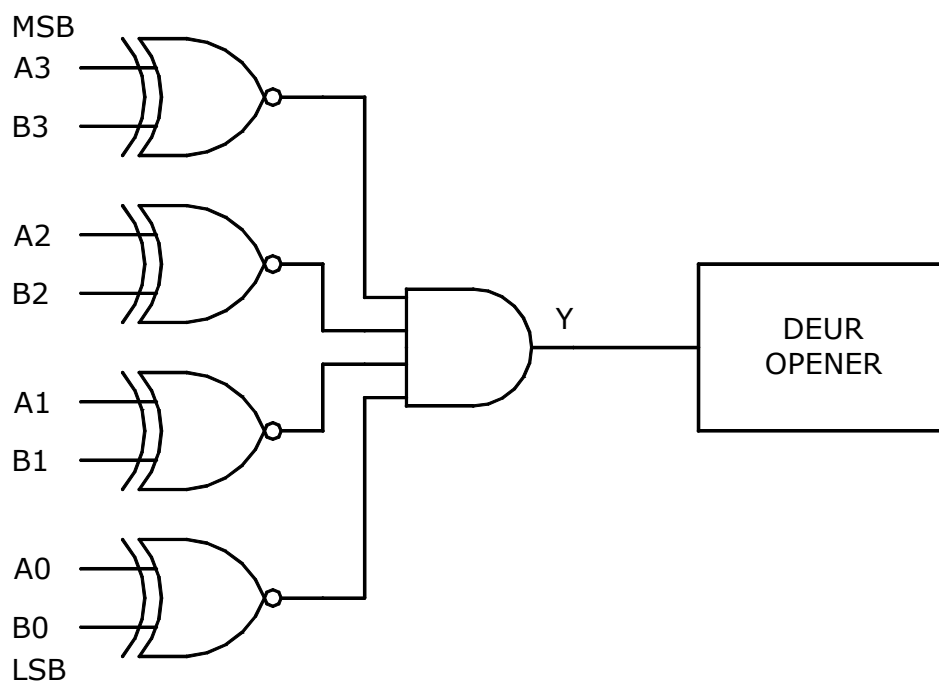
△ Fig.2.24 | AND-gate met 2 ingangen als blokkeerpoort.
Bron: W.Van Wichelen

[11] 'to enable', letterlijk vertaald: 'in staat stellen'.

Het is een veel gebruikte techniek om via een bijkomende E-aansluiting (enable-pin) een digitale schakeling al of niet te kunnen vergrendelen.

EXNOR als vergelijker

EXNOR-poorten kunnen gebruikt worden om twee getallen (A en B) te *vergelijken*. Laten we een principeschakeling voor een automatisch deurslot even van naderbij bekijken. De decimale getallen A en B zijn binair gecodeerd m.b.v. 4 bits. A is het decimale cijfer dat we via een numeriek klavier moet ingeven en B is een vast ingesteld cijfer dat alleen de eigenaar kent. Uitgang Y zal enkel hoog worden indien alle overeenkomende bits gelijk zijn aan elkaar. Dit hoge signaal zal via een geschikte interface de deur automatisch openen.



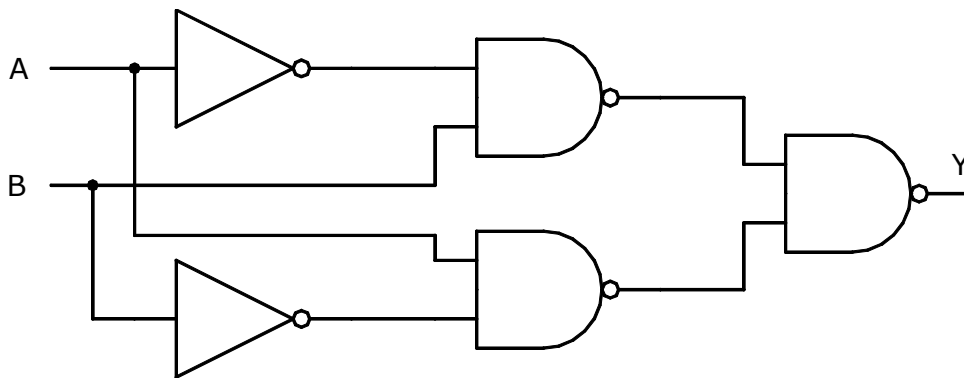
△ Fig.2.25 | EXNOR-gate met 4 ingangen als cijferslot.
Bron: W.Van Wichelen



△ Fig.2.26 | Voorbeeld van een numeriek klavier van een digitaal cijferslot. Bron: world wide web

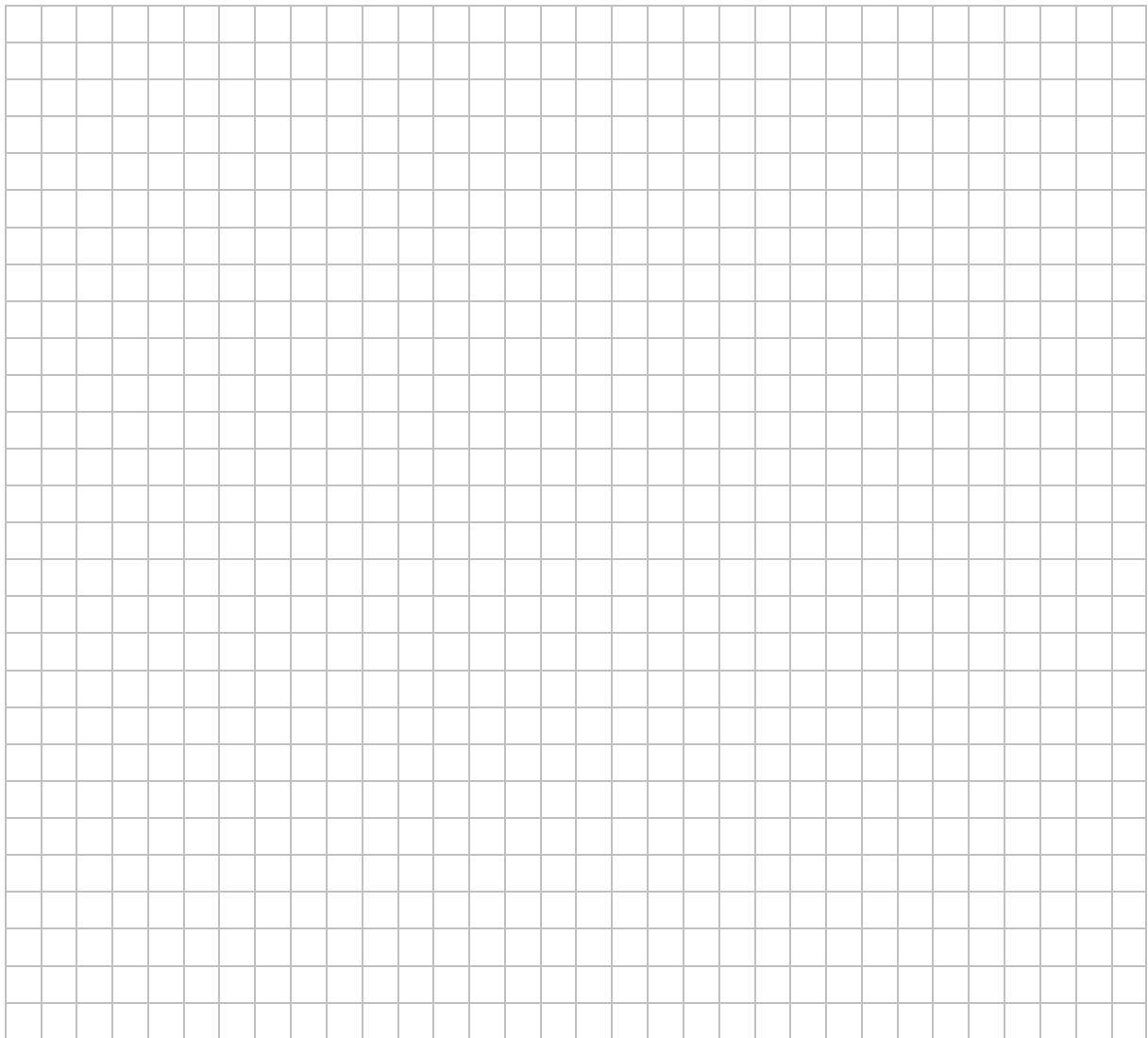
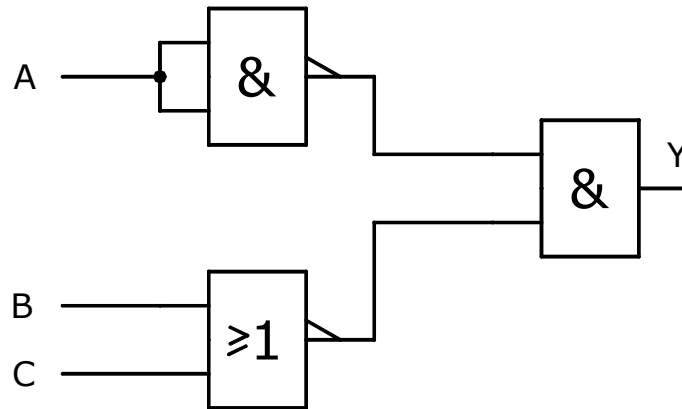
2.3.8 Oefeningen

1. Bestudeer onderstaande logische schakeling.
 - a. Stel de waarheidstabel op.
 - b. Stel de logische vergelijking op.
 - c. Herken je deze functie?



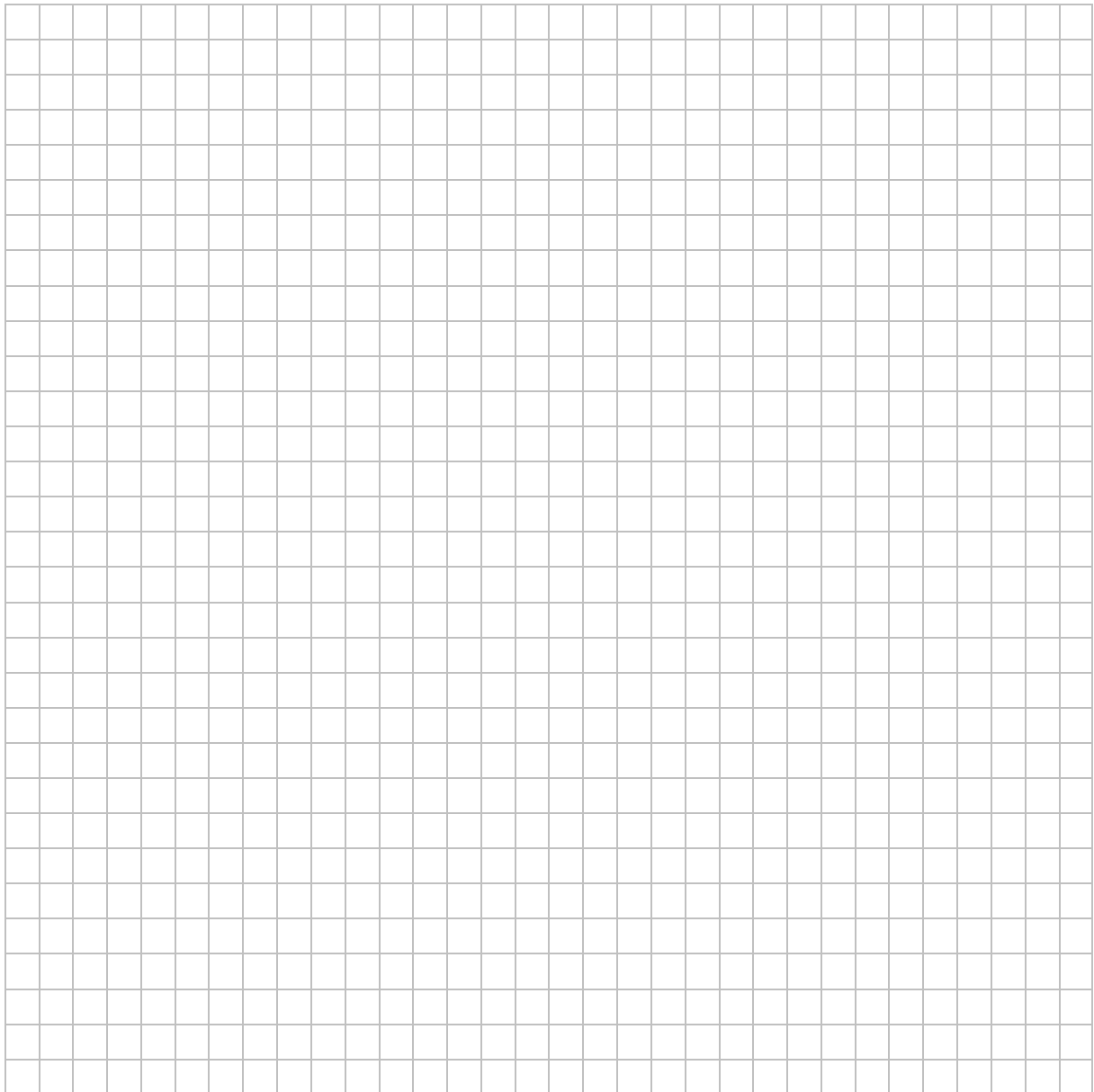
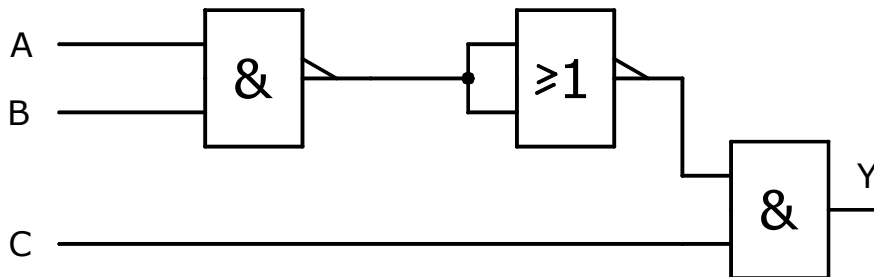
2. Bestudeer onderstaande logische schakeling.

- Stel de waarheidstabel op.
- Stel de logische vergelijking op.
- Herken je deze functie?

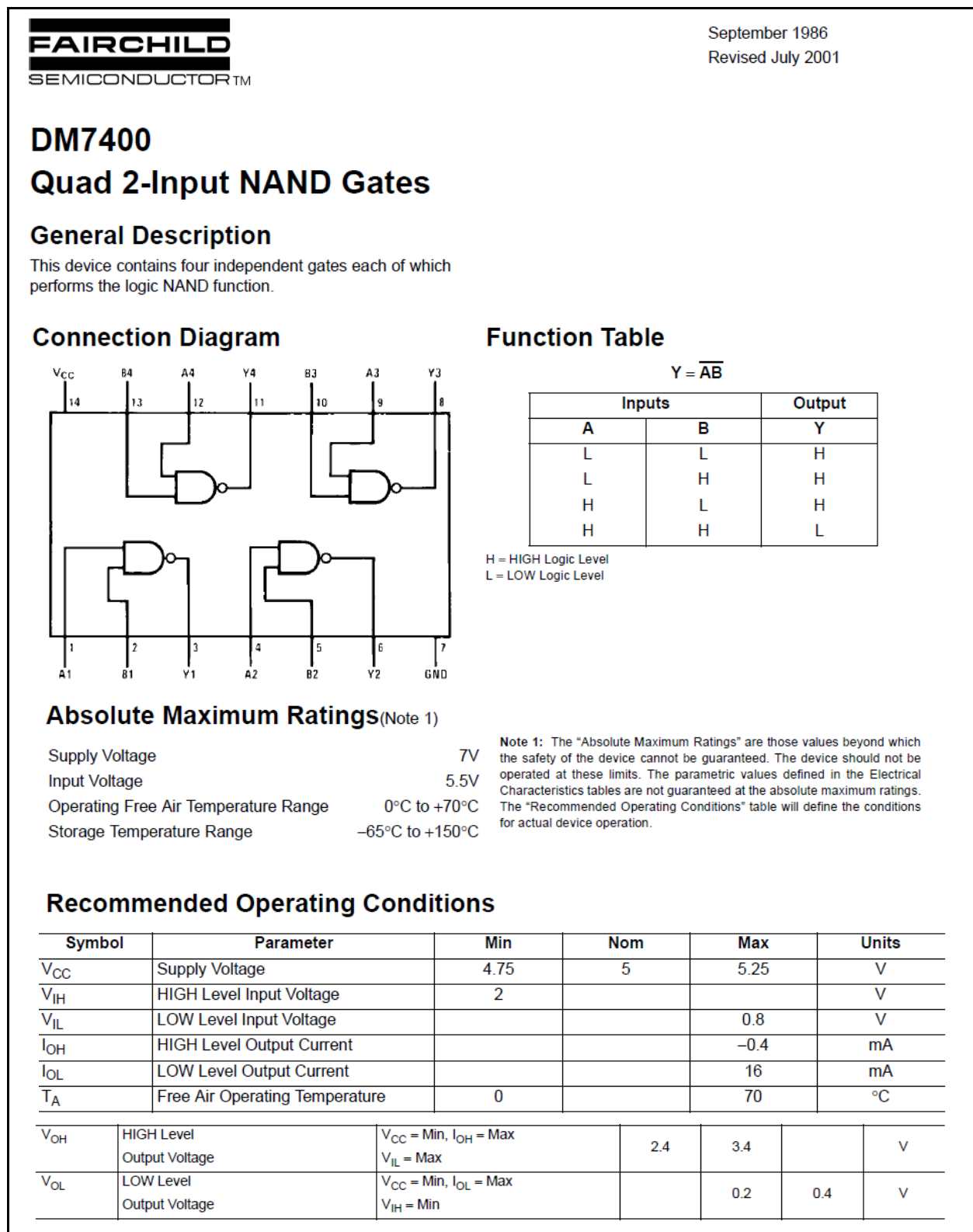


3. Bestudeer onderstaande logische schakeling.

- Stel de waarheidstabel op.
- Stel de logische vergelijking op.
- Herken je deze functie?



2.3.9 Datasheet TTL-poort



△ Fig.2.27 | Overzicht van de voornaamste gegevens uit een datasheet van het IC 7400.
Bron: www.fairchildsemi.com

Belangrijke parameters

- $V_{CC(nom)}$: nominale voedingsspanning.
TTL: 5V
High Speed CMOS: van 2V tot 6V
- $V_{IH(min)}$: de minimum spanning aan de ingang die nog als een logische '1' wordt gezien.
TTL: 2V
High Speed CMOS: 3,15V (bij $V_{CC}=4,5V$)
- $V_{IL(max)}$: de maximum spanning aan de ingang die nog als een logische '0' wordt gezien.
TTL: 0,8V
High Speed CMOS: 1.35V (bij $V_{CC}=4,5V$)
- $I_{OH(max)}$: de maximum stroom die een uitgang van een poort mag leveren bij een hoog niveau ('1').
TTL: 0,4mA (current sourcing)
High Speed CMOS: 25mA (current sourcing)
- $I_{OL(max)}$: de maximum stroom die in een uitgang van een poort mag ontvangen bij een laag niveau ('0').
TTL: 16mA (current sinking)
High Speed CMOS: 25mA (current sinking)
- $V_{OH(nom)}$: de nominale uitgangsspanning bij een hoog niveau (logische '1')
TTL: 3,4V
High Speed CMOS: ca. V_{CC} (onbelast)
- $V_{OL(nom)}$: de nominale uitgangsspanning bij een laag niveau (logische '0')
TTL: 0,2V
High Speed CMOS: ca. 0V

2.3.10 Sinking en sourcing

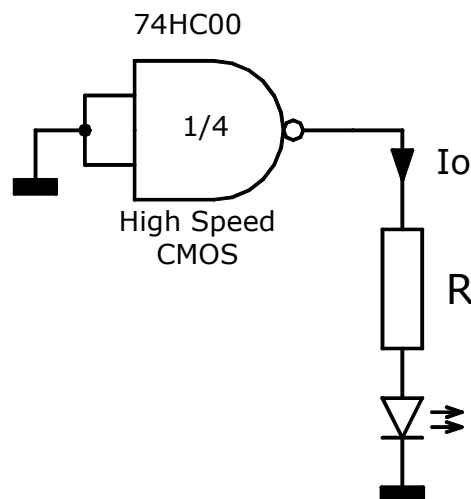
Wanneer we met digitale poorten spelen, is het wenselijk dat we de logische niveau's aan de ingang en de uitgang van de verschillende poorten op één of andere manier kunnen 'uitlezen'. De meest eenvoudige methode is gebruik te maken van een lichtgevende diode: de LED. Let echter wel op! Je kan de LED niet zomaar aansluiten op bijvoorbeeld een uitgang van een NAND-poort van het IC '7400'. Stel dat je bij een hoog niveau een rode LED wil laten oplichten. Typisch verbruikt een rode LED ongeveer 5mA bij een klemspanning van ongeveer 1,8 volt. Een TTL-poort kan bij een hoog uitgangsniveau van +3,4V een stroom leveren van 0,4 mA. Zowel de spanning als de stroom zijn niet aangepast: de poort kan onvoldoende stroom leveren en de uitgangsspanning is te hoog.

Principe 'current sourcing'

De poort zelf zorgt voor de stroom die de belasting (in ons voorbeeld een rode LED) vraagt.

High Speed CMOS-poorten (74HC-reeks) zijn in staat om continu ongeveer 20mA aan een belasting te leveren.

Deze manier van aansturen wordt zelden toegepast bij klassieke TTL-poortschakelingen (74LS-reeks).

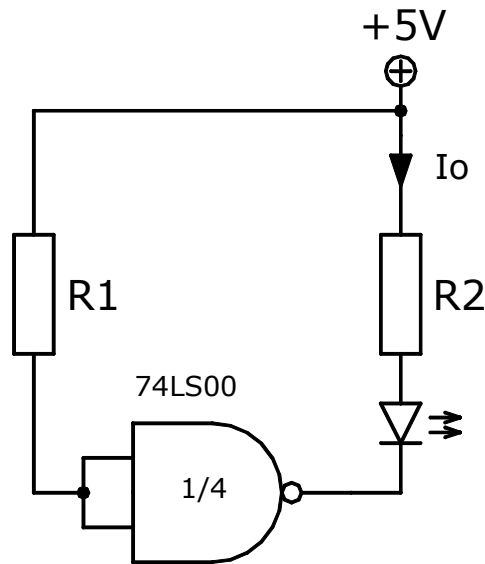


△ Fig.2.28 | Principe van current sourcing.
Bron: W.Van Wichelen

Principe 'current sinking'

De poort ontvangt de stroom die de belasting (in ons voorbeeld een rode LED) vraagt. De stroom wordt geleverd door de voedingsspanning.

Deze manier van aansturen wordt meestal toegepast bij TTL-poorten van bijvoorbeeld de 74LS-serie. Een nadeel van deze methode is dat de LED zal oplichten bij een logisch laag niveau.



△ Fig.2.29 | Principe van current sinking.
Bron: W.Van Wichelen

R1: 'pull up-weerstand'

R2: voorschakelweerstand (stroombegrenzing)

Laboproef 1

Realiseer een AND-functie met drie ingangen met het IC '7400' (74LS00). Stel de waarheidstabel op en verifieer ze met metingen. Visualiseer de logische toestand van de uitgang met een blauwe LED. Pas het principe van 'current sinking' toe.

3 | Digitale techniek

Logische vergelijkingen



Logische vergelijkingen laten toe om een bepaalde logische beweringen eenduidig te beschrijven. Via booleaanse algebra kunnen we deze vergelijkingen vereenvoudigen en minimaliseren. Dankzij deze techniek kunnen we het aantal digitale IC's in digitale systemen tot een minimum beperken. Let echter wel op dat je bij het oplossen van booleaanse vergelijkingen de regels van de klassieke algebra niet altijd zomaar mag toepassen!

3.1 Boole algebra

3.1.1 Basiswetten

De drie basiswetten van de Booleaanse algebra zijn dezelfde als in de klassieke algebra: de *commutatieve* wetten, de *associatieve* wetten en de *distributieve* wetten.

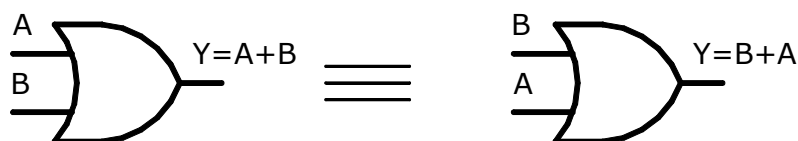
Commutativiteit

De *volgorde* waarin de termen van een **som** staan in een logische vergelijking is onbelangrijk:

$$A + B = B + A$$

De *volgorde* waarin de factoren van een **product** staan in een logische vergelijking is onbelangrijk:

$$A \cdot B = B \cdot A$$



△ Fig.3.1 | Voorbeeld van commutativiteit.
Bron: W.Van Wichelen

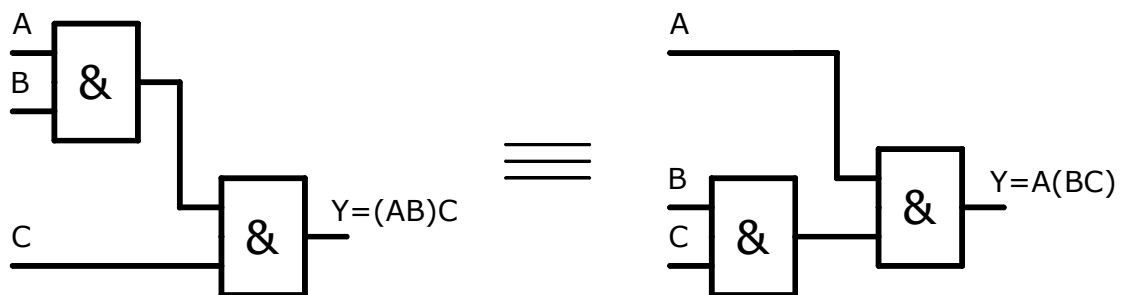
Associativiteit

Je mag steeds termen *groeperen* in een logische vergelijking:

$$(A + B) + C = A + (B + C)$$

Je mag steeds factoren *groeperen* in een logische vergelijking:

$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

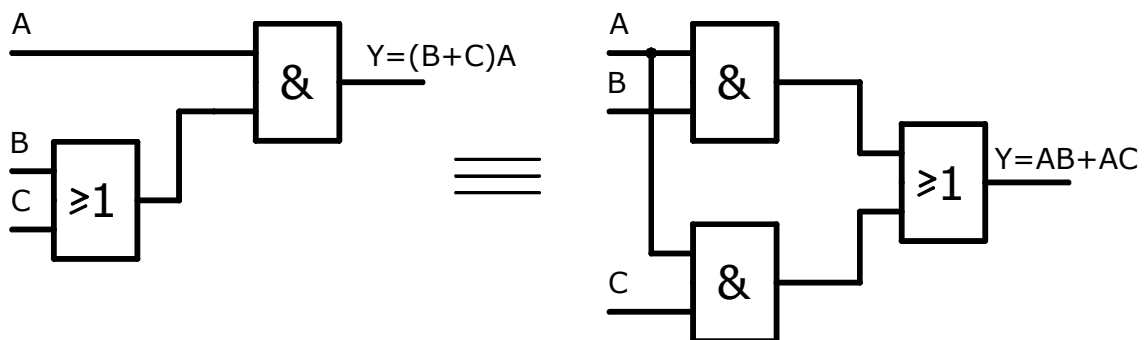


△ Fig.3.2 | Voorbeeld van associativiteit.
Bron: W.Van Wichelen

Distributiviteit

Je mag de haakjes uitwerken zoals bij de klassieke algebra:

$$A \cdot (B + C) = A \cdot B + A \cdot C$$



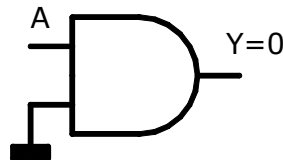
△ Fig.3.3 | Voorbeeld van distributiviteit.
Bron: W.Van Wichelen

3.1.2 Basisregels vermenigvuldiging

Logisch vermenigvuldigen met 0

Als in een product één of meer factoren '0' zijn, dan is het resultaat altijd '0':

$$A \cdot 0 = 0$$

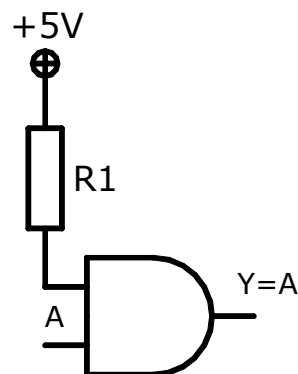


△ Fig.3.4 | Voorbeeld van logisch vermenigvuldigen met '0'.
Bron: W.Van Wichelen

Logisch vermenigvuldigen met 1

Als in een product één of meer factoren '1' zijn, dan hebben die factoren geen invloed op het resultaat:

$$A \cdot 1 = A$$

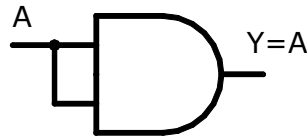


△ Fig.3.5 | Voorbeeld van logisch vermenigvuldigen met '1'.
Bron: W.Van Wichelen

Logisch vermenigvuldigen met zichzelf

Als in een product één of meer factoren *dezelfde* zijn, dan volstaat het die factoren slechts één keer te noteren:

$$A \cdot A = A$$

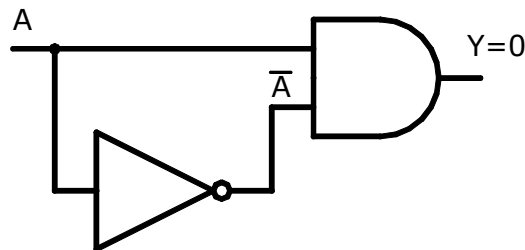


△ Fig.3.6 | Voorbeeld van logisch vermenigvuldigen met zichzelf. Bron: W.Van Wichelen

Logisch vermenigvuldigen met het inverse

Als in een product twee factoren elkaars *tegengestelde* zijn, dan is het resultaat steeds '0':

$$A \cdot \bar{A} = 0$$



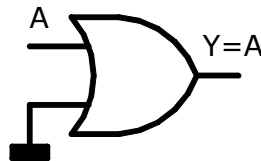
△ Fig.3.7 | Voorbeeld van logisch vermenigvuldigen met het inverse. Bron: W.Van Wichelen

3.1.3 Basisregels optelling

Logisch optellen bij 0

Als in een som één of meer termen '0' zijn, dan hebben die termen geen invloed op het resultaat:

$$A + 0 = A$$

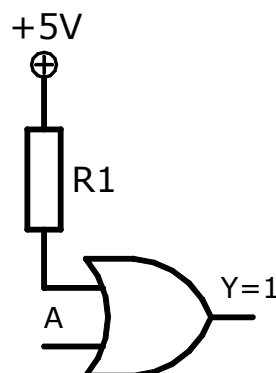


△ Fig.3.8 | Voorbeeld van logisch optellen met '0'.
Bron: W.Van Wichelen

Logisch optellen bij 1

Als in een som één of meer termen '1' zijn, dan is het resultaat altijd '1':

$$A + 1 = 1$$

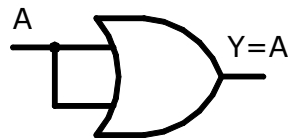


△ Fig.3.9 | Voorbeeld van logisch optellen met '1'.
Bron: W.Van Wichelen

Logisch optellen bij zichzelf

Als in een som één of meer termen *dezelfde* zijn, dan volstaat het die term slechts één keer te noteren:

$$A + A = A$$

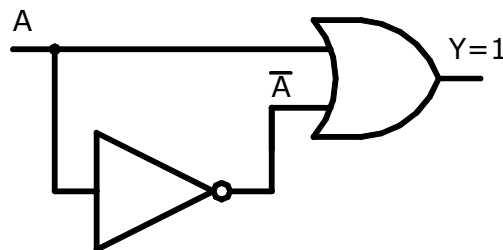


△ Fig.3.10 | Voorbeeld van logisch optellen met zichzelf.
Bron: W.Van Wichelen

Logisch optellen bij het inverse

Als in een som twee termen elkaars *tegengestelde* zijn, dan is het resultaat steeds '1':

$$A + \bar{A} = 1$$



△ Fig.3.11 | Voorbeeld van logisch optellen met inverse.
Bron: W.Van Wichelen

3.1.4 Tweemaal inverteren

Let op!

Een variabele die *tweemaal* geïnverteerd wordt, keert terug tot haar oorspronkelijke toestand:

$$\bar{\bar{A}} = A$$

3.2 Theorema van De Morgan

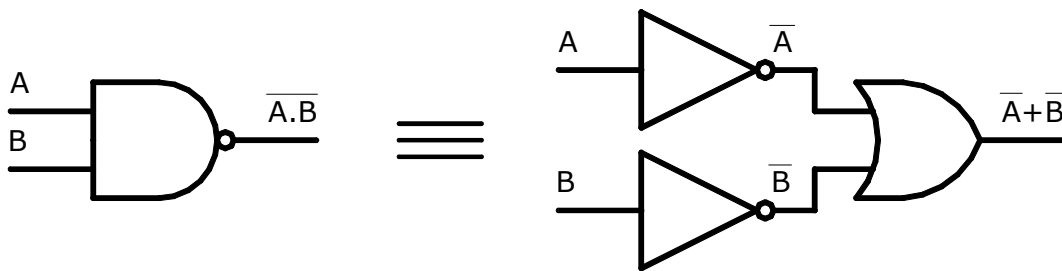
Het *theorema van De Morgan*^[12] formuleert twee wetten in de booleaanse logica die een verband leggen tussen AND en OR en de negatie (NOT).

3.2.1 Van product naar som

Let op!

Met booleaanse algebra is het mogelijk een logisch *product* om te vormen naar een logische *som*:

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$



△ Fig.3.12 | Van logische som naar logisch product.
Bron: W.Van Wichelen

Uit bovenstaande uitdrukking kunnen we afleiden dat we met NOR-poorten een *AND-functie* kunnen realiseren:

$$\Rightarrow \overline{A \cdot B} = \overline{A} + \overline{B}$$

$$\Leftrightarrow \overline{\overline{A \cdot B}} = \overline{\overline{A} + \overline{B}}$$

$$\Leftrightarrow A \cdot B = \overline{\overline{A} + \overline{B}}$$

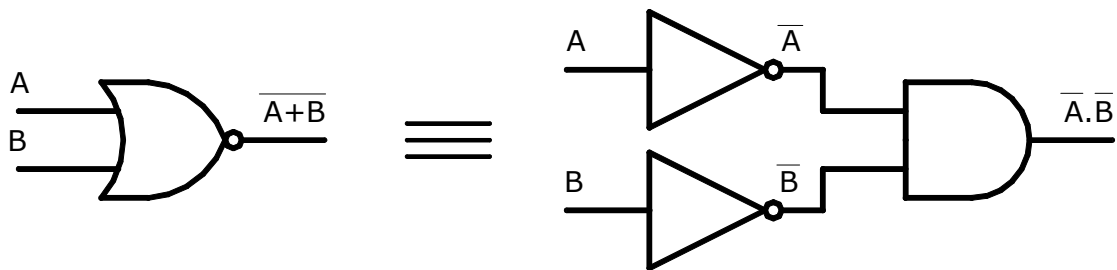
[12] Augustus De Morgan (1806–1871) was een Britse wiskundige en logicus. Bij het formuleren van zijn stelling werd hij beïnvloed door de ontwikkeling van de Booleaanse algebra door George Boole. Voor hem formuleerden Aristoteles en middeleeuwse denkers reeds deze wet.

3.2.2 Van som naar product

Let op!

Met booleaanse algebra is het mogelijk een logische *som* om te vormen naar een logisch *product*:

$$\overline{A+B} = \overline{A} \cdot \overline{B}$$



△ Fig.3.13 | Van logisch product naar logische som.
Bron: W.Van Wichelen

Uit bovenstaande uitdrukking kunnen we afleiden dat we met NAND-poorten een *OR-functie* kunnen realiseren:

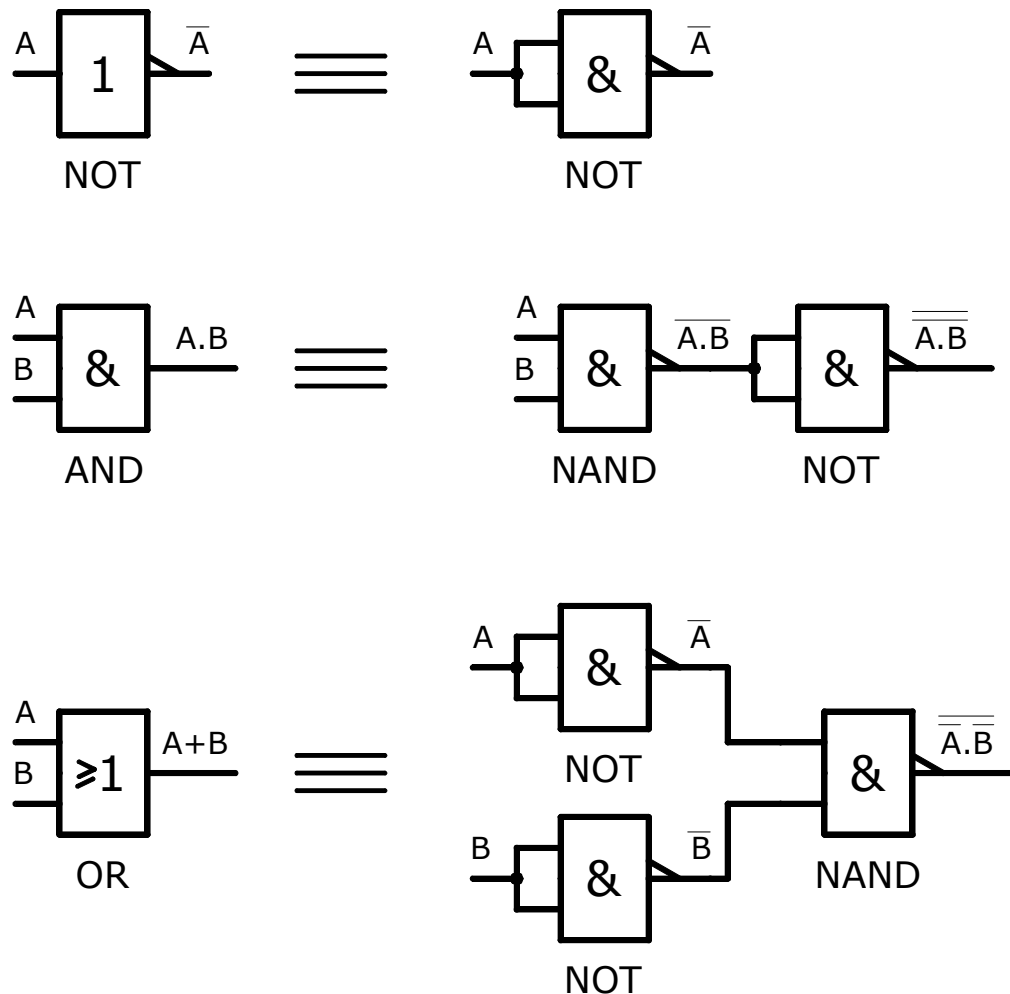
$$\Rightarrow \overline{A+B} = \overline{A} \cdot \overline{B}$$

$$\Leftrightarrow \overline{\overline{A+B}} = \overline{\overline{A} \cdot \overline{B}}$$

$$\Leftrightarrow A+B = \overline{\overline{A} \cdot \overline{B}}$$

3.2.3 Praktisch nut

Omdat we booleaanse functies gemakkelijk kunnen omvormen kiezen fabricanten van geïntegreerde schakelingen (IC's) ervoor om de schakellogica uit te voeren met enkel **NAND**-poorten. Hier hebben ze uiteraard wel goede redenen voor. NAND-technologie is makkelijker te produceren en dus ook goedkoper. Figuur 3.14 toont de realisatie van de drie basisfuncties NOT, OR en EN met enkel NAND-poorten.

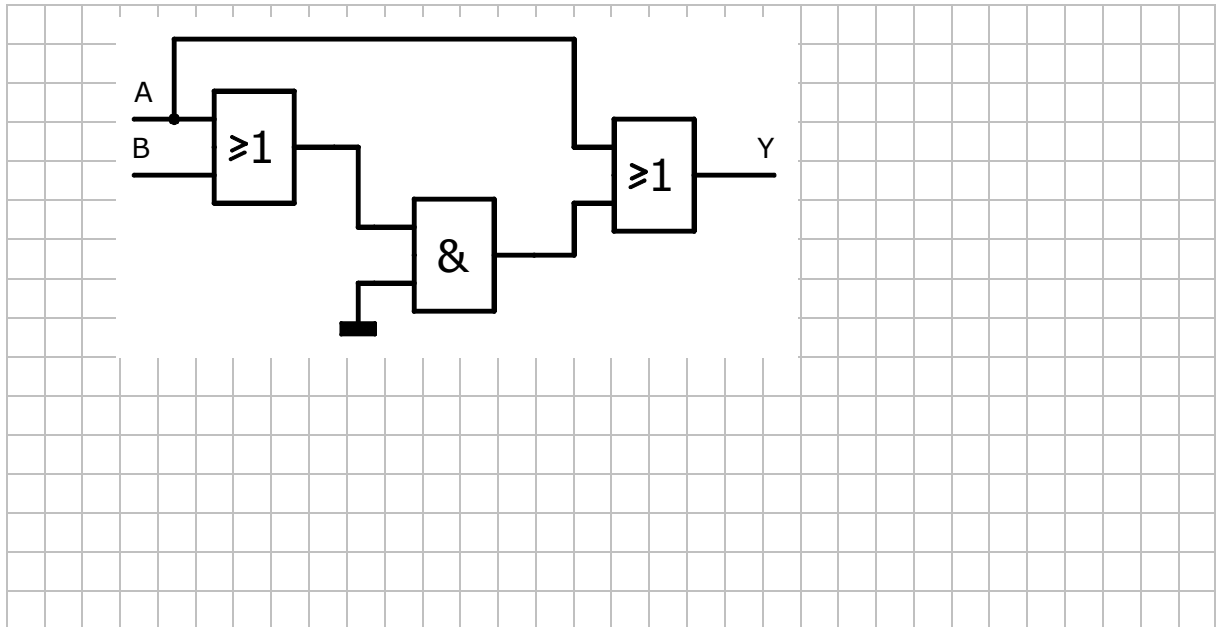


△ Fig.3.14 | De drie basisfuncties door enkel gebruik te maken van NAND-poorten. Bron: W.Van Wichelen

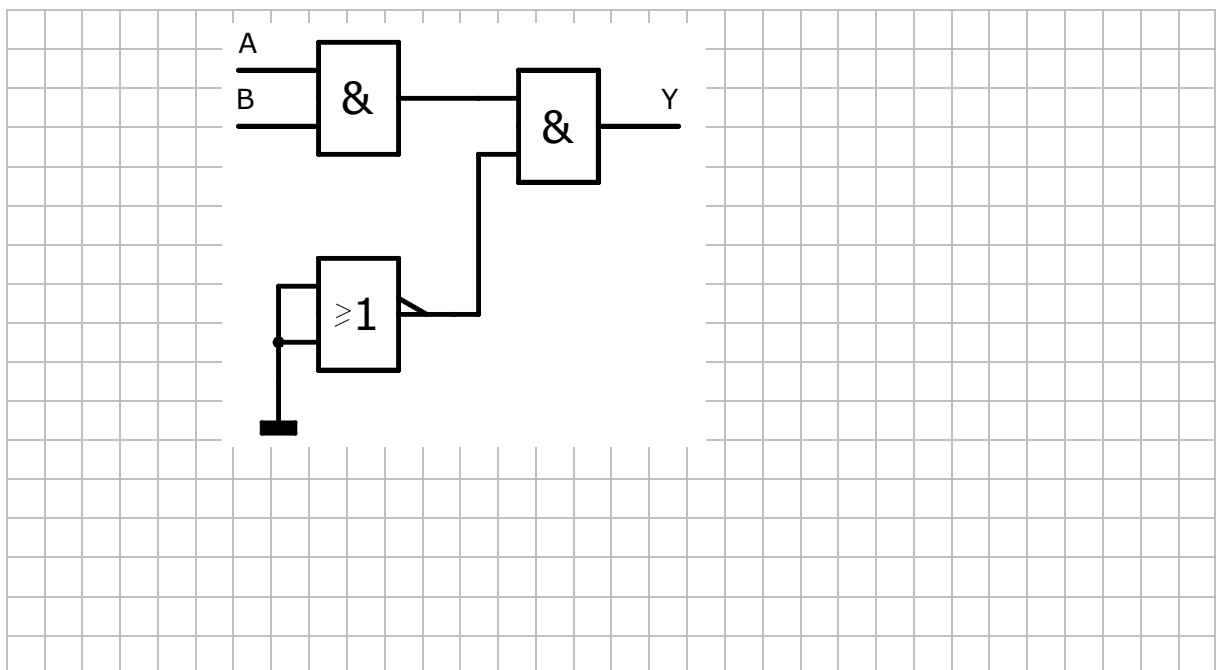
3.2.4 Oefeningen

Reeks 1

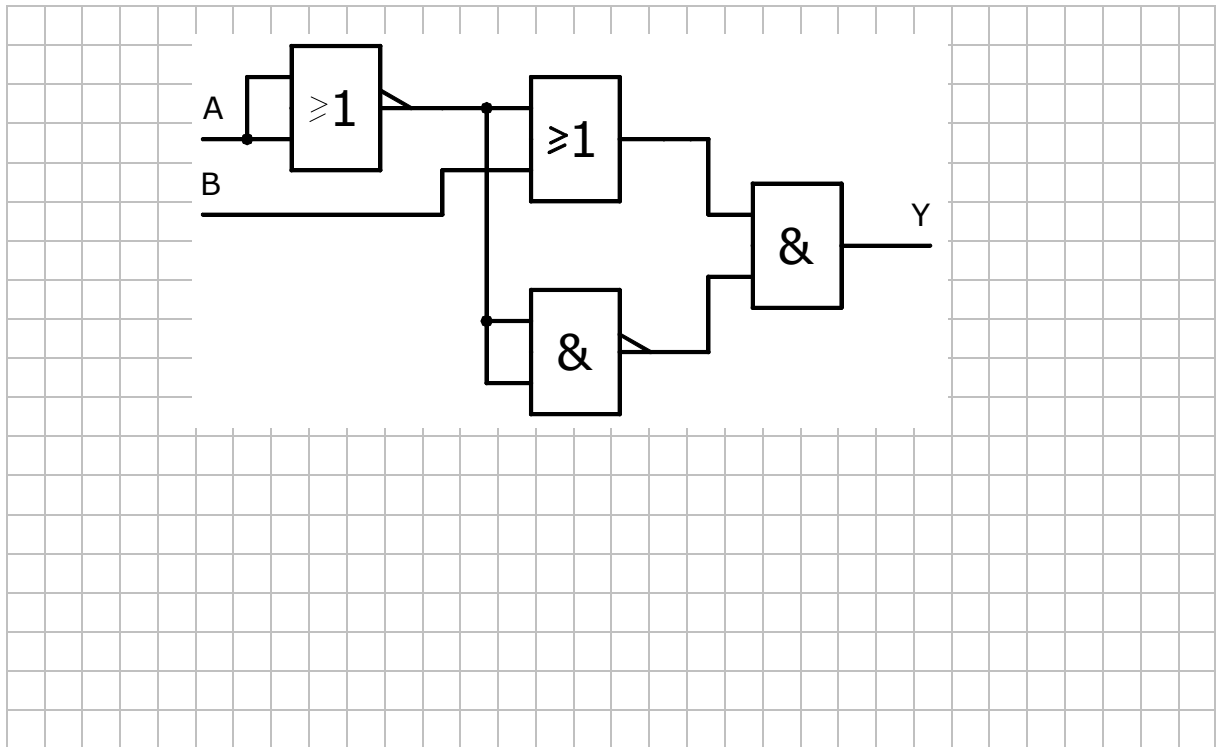
1. Bestudeer onderstaande logische schakeling.
 - a. Stel de logische vergelijking op.
 - b. Vereenvoudig door gebruik te maken van de basisregels.



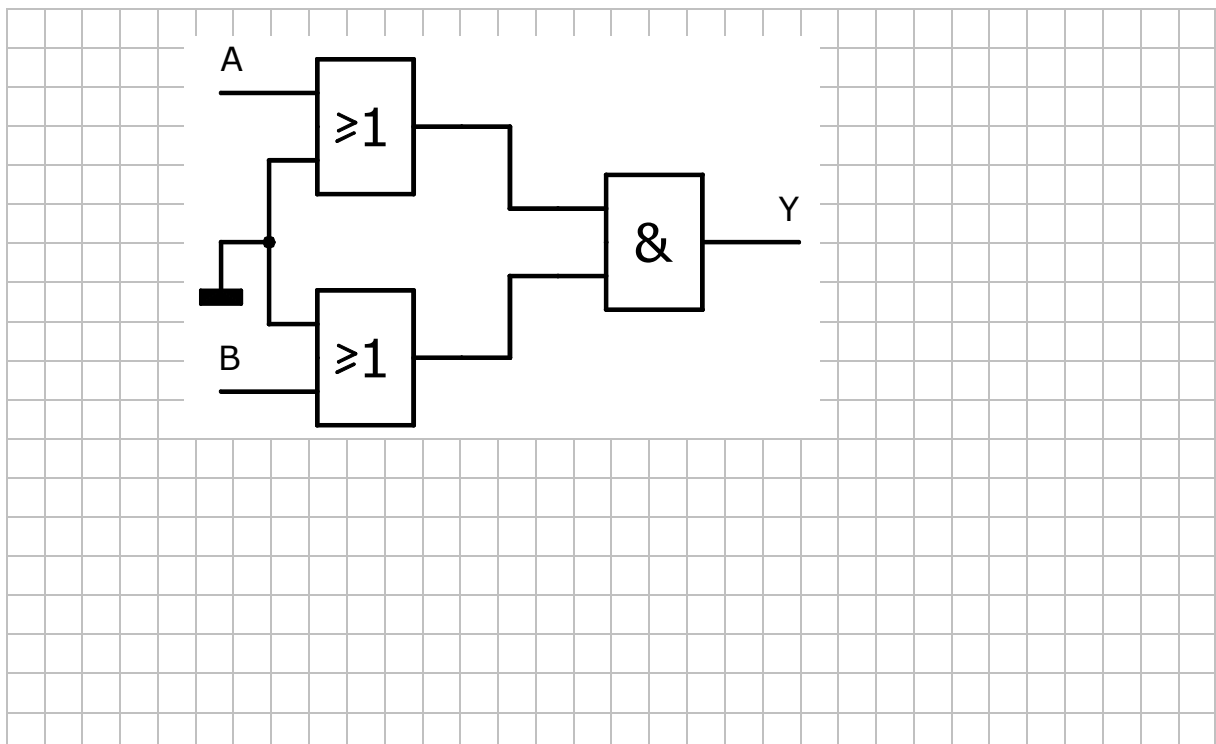
2. Bestudeer onderstaande logische schakeling.
 - a. Stel de logische vergelijking op.
 - b. Vereenvoudig door gebruik te maken van de basisregels.



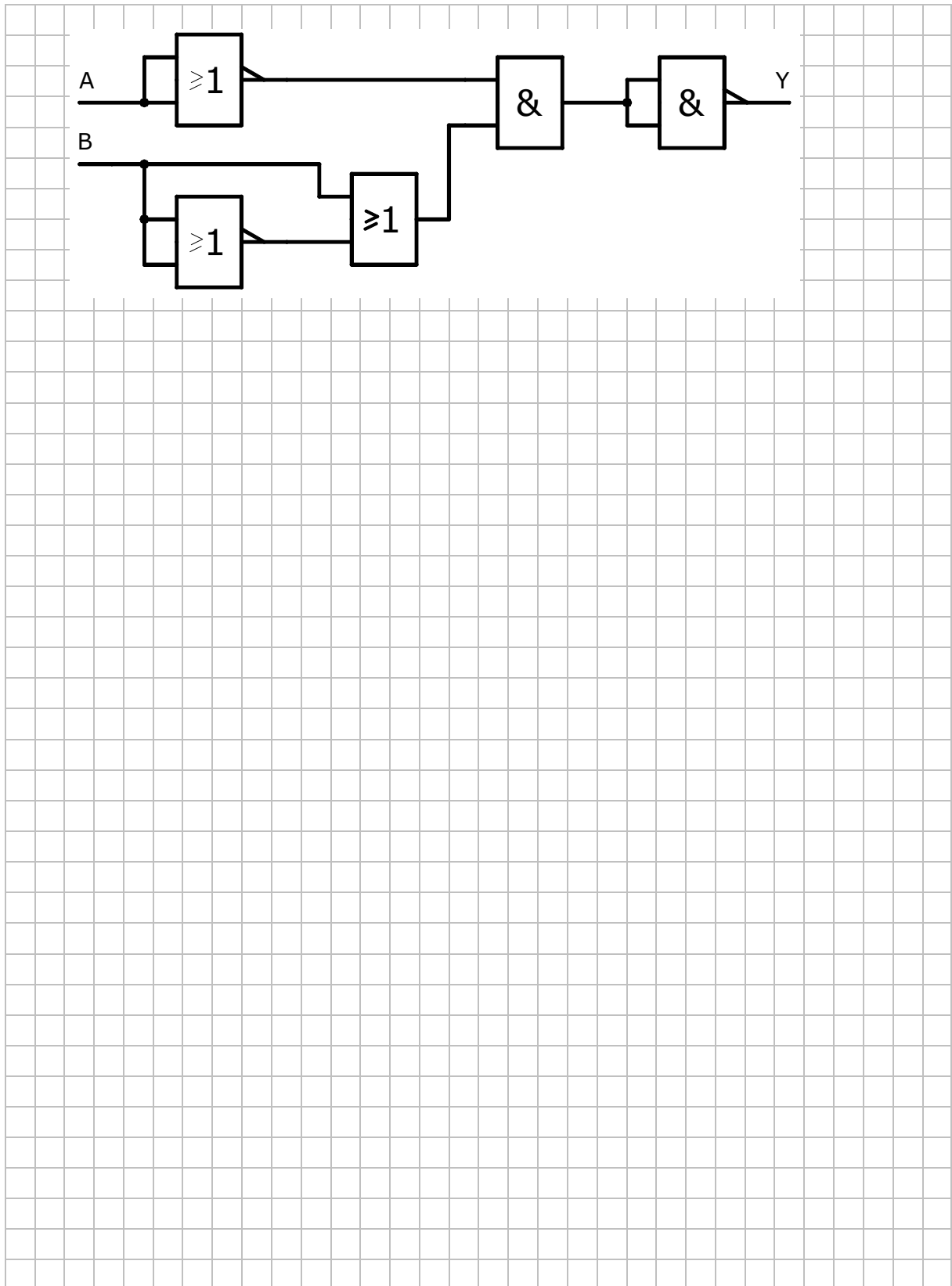
3. Bestudeer onderstaande logische schakeling.
- Stel de logische vergelijking op.
 - Vereenvoudig door gebruik te maken van de basisregels.



4. Bestudeer onderstaande logische schakeling.
- Stel de logische vergelijking op.
 - Vereenvoudig door gebruik te maken van de basisregels.

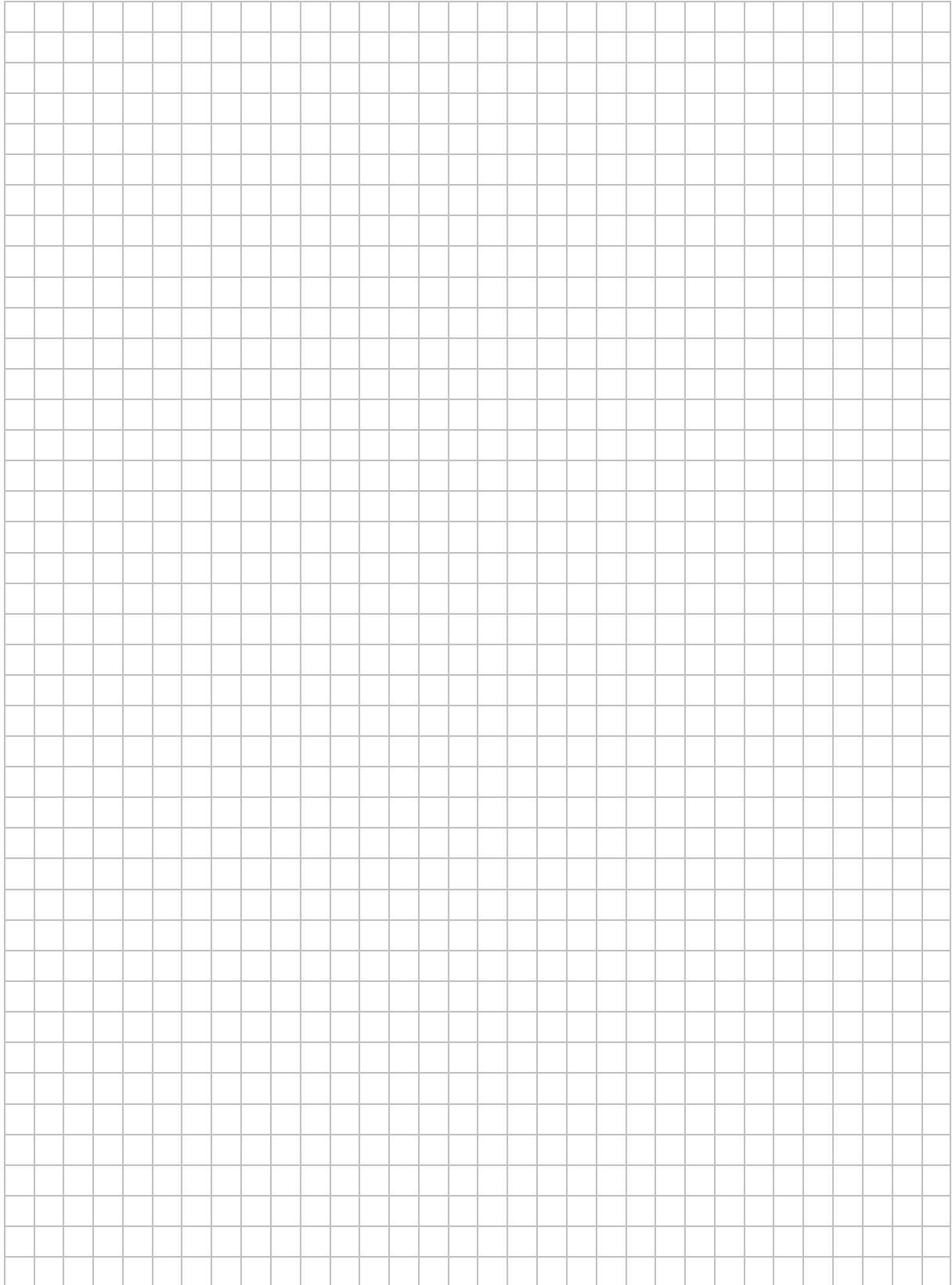


5. Bestudeer onderstaande logische schakeling.
- Stel de logische vergelijking op.
 - Vereenvoudig door gebruik te maken van de basisregels.

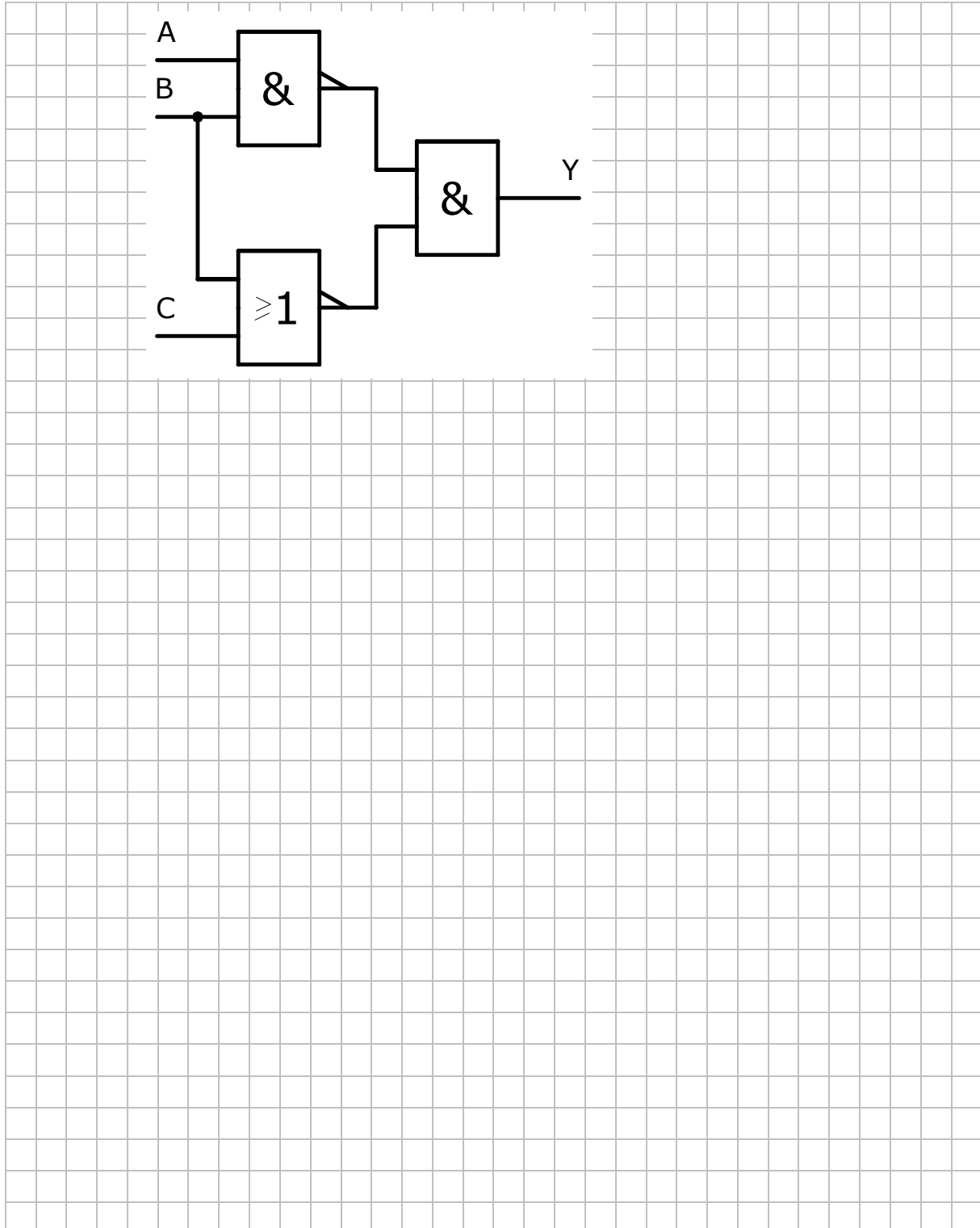


Reeks 2

1. Realiseer een digitale schakeling die voldoet aan de vergelijking: $Y = \overline{A} \cdot B + C$. Je mag enkel gebruik maken van het IC 74LS00. Maak gebruik van de stelling van De Morgan.



2. Bestudeer onderstaande logische schakeling.
- Stel de logische vergelijking op.
 - Pas het theorema van De Morgan toe en vereenvoudig de vergelijking.
 - Teken het vereenvoudigde schema met de basispoorten.
 - Realiseer de schakeling met 4 TTL poorten van het IC 74LS00.

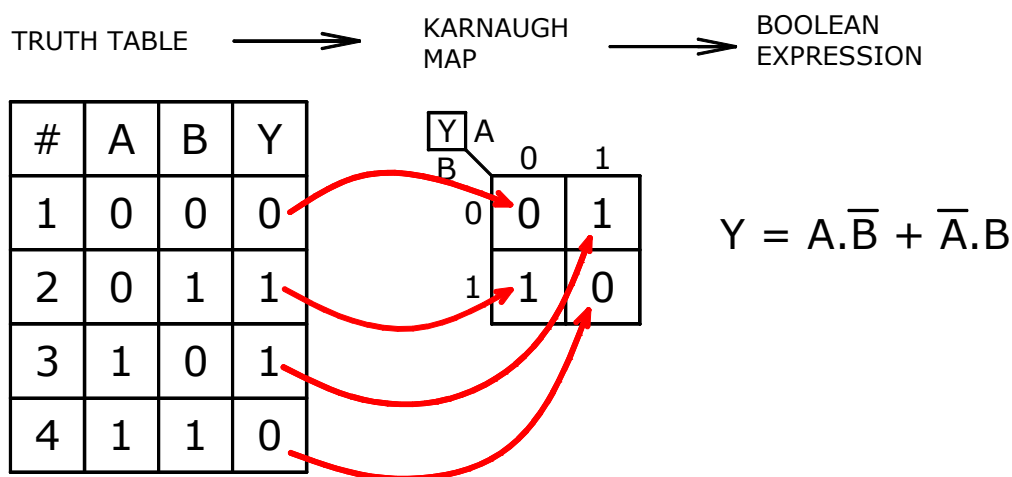


3.3 Karnaugh-diagrammen

Eerder leerden we om logische vergelijkingen te vereenvoedigen met booleaanse algebra. ICT-technici - die niet erg vertrouwd zijn met de regels en theorema's van de Boole-algebra - maken liever gebruik van een *grafische* methode om tot de meest eenvoudige logische uitdrukking te komen. Doordat het menselijk brein gemakkelijk patronen kan herkennen, helpt een zg. **karnaugh-diagram**^[13] om snel op te zoeken welke termen gecombineerd kunnen worden. We kunnen nu visueel logische vergelijkingen, met maximaal 6 logische variabelen, minimaliseren. Het stelt ons in staat om digitale logica op te leveren met een minimum aantal elektronische componenten, waardoor de kans op defecte poorten aanzienlijk vermindert. De alzo ontworpen schakelingen bezitten kleinere afmetingen en zijn goedkoper om te produceren.

3.3.1 Karnaugh-diagram voor 2 variabelen

De waarheidstabel toont de status (0 of 1) van de uitgang (Y) voor iedere mogelijke combinatie van de ingangsvariabelen (A en B). Een karnaugh-diagram geeft dezelfde informatie maar iets anders voorgesteld. Elke ingangscombinatie uit de waarheidstabel stemt overeen met een welbepaald vakje in het karnaugh-diagram. Figuur 3.15 illustreert hoe dit in zijn werk gaat.



△ Fig.3.15 | Omzetting van een waarheidstabel naar een karnaugh-diagram voor 2 booleaanse variabelen.
Bron: W.Van Wichelen

[13] Het karnaugh-diagram werd uitgevonden in 1950 door Maurice Karnaugh, een telecommunicatie-ingenieur bij Bell Labs.

3.3.2 Karnaugh-diagram voor 3 variabelen

#	A	B	C	Y
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	1
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

Y	C	AB			
		00	01	11	10
0		0	0	1	1
1		1	1	1	1

$$Y = A.B.\bar{C} + A.\bar{B}.\bar{C} + \bar{A}.\bar{B}.C + \bar{A}.B.C + A.B.C + A.\bar{B}.C$$

△ Fig.3.16 Omzetting van een waarheidstabel naar een karnaugh-diagram voor 3 booleaanse variabelen.
Bron: W.Van Wichelen

3.3.3 Karnaugh-diagram voor 4 variabelen

#	A	B	C	D	Y
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	0
4	0	0	1	1	1
5	0	1	0	0	1
6	0	1	0	1	1
7	0	1	1	0	1
8	0	1	1	1	1
9	1	0	0	0	0
10	1	0	0	1	1
11	1	0	1	0	0
12	1	0	1	1	1
13	1	1	0	0	0
14	1	1	0	1	0
15	1	1	1	0	0
16	1	1	1	1	0

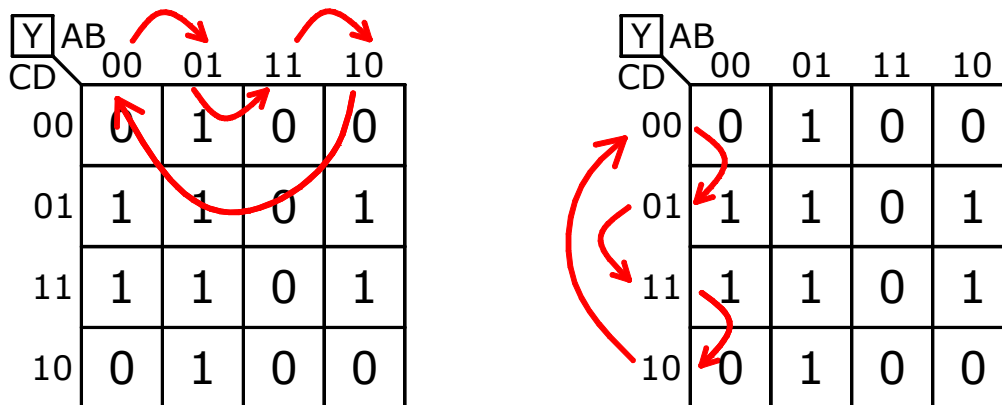
Y	CD	AB			
		00	01	11	10
00		0	1	0	0
01		1	1	0	1
11		1	1	0	1
10		0	1	0	0

$$Y = \bar{A}.B.\bar{C}.\bar{D} + \bar{A}.\bar{B}.\bar{C}.D + \bar{A}.B.\bar{C}.D + A.\bar{B}.\bar{C}.D + \bar{A}.\bar{B}.C.D + \bar{A}.B.C.D + A.\bar{B}.C.D + \bar{A}.B.C.\bar{D}$$

△ Fig.3.17 Omzetting van een waarheidstabel naar een karnaugh-diagram voor 4 booleaanse variabelen.
Bron: W.Van Wichelen

3.3.4 Regels voor minimalisatie

Wanneer je een karnaugh-diagram van naderbij bekijkt dan merk je dat bij de overgang van twee naast elkaar gelegen vakken er slechts één variabele is die van toestand wijzigt. Ook voor de grensvakken (uiterst links, uiterst rechts, bovenaan en onderaan) is dit het geval. Dankzij deze schikking is het mogelijk om de logische vergelijking te vereenvoudigen of te minimaliseren.



△ Fig.3.18 | Schikking van de vakken in een karnaugh-diagram voor 4 booleaanse variabelen.
Bron: W.Van Wichelen

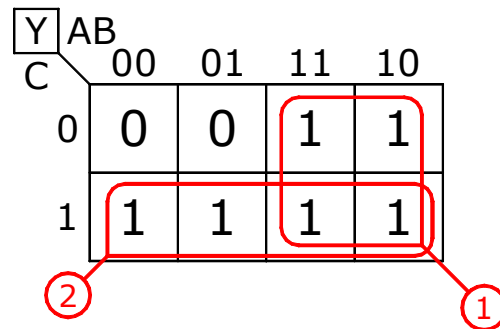
Aangrenzende vakken vormen een zg. **lus**. Het is de bedoeling om zo groot mogelijke lussen te maken in een poging om zoveel mogelijk enen te vangen. Hierbij houden we ons aan volgende regels:

Regels voor minimalisatie

- We maken steeds een zo *groot mogelijke* lus met vakken die **enen** bevatten en we herhalen dit tot *alle* enen in een bepaalde lus zijn opgenomen.
- Elke lus moet een *rechthoek* of een *vierkant* vormen met **1, 2, 4** of **8** aangrenzende vakken.
- Noteer enkel de variabelen die *constant* zijn binnen alle vakken in een lus.
- Lussen mogen elkaar *overlappen*.
- Je moet alle enen tenminste één keer aflezen.

3.3.5 Minimalisatie met een karnaugh-map

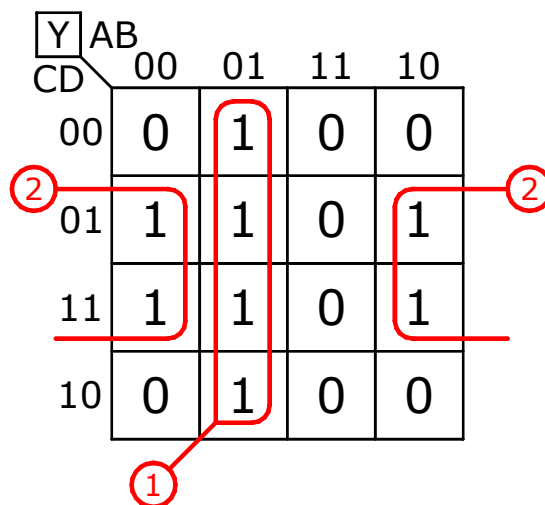
Voorbeeld 1



△ Fig.3.19 Minimalisatie met een karnaugh-diagram voor 3 booleaanse variabelen. Bron: W.Van Wichelen

Bij het aflezen van lus 1 brengen we enkel de constanten in rekening: $Y_1 = A$. Zo levert lus 2: $Y_2 = C$. Finaal kunnen we schrijven: $Y = A + C$.

Voorbeeld 2



△ Fig.3.20 Minimalisatie met een karnaugh-diagram voor 4 booleaanse variabelen. Bron: W.Van Wichelen

We lezen lus 1 af: $Y_1 = \bar{A} \cdot B$

We lezen lus 2 af: $Y_2 = \bar{B} \cdot D$

De minimale logische vergelijking wordt: $Y = \bar{A}B + \bar{B}D$

3.3.6 Inverse van een functie

Via een karnaugh-diagram is het eenvoudig het inverse van een bepaalde booleaanse uitdrukking te bepalen. Het volstaat om de nullen af te lezen in plaats van de enen. Het voorbeeld van figuur 3.21 maakt dit duidelijk.

Y	AB	00	01	11	10
CD	00	0	1	0	0
01	1	1	1	0	1
11	1	1	1	0	1
10	0	1	1	0	0

△ Fig.3.21 Afleiden van de inverse functie met een karnaugh-diagram voor 4 booleaanse variabelen.
Bron: W.Van Wichelen

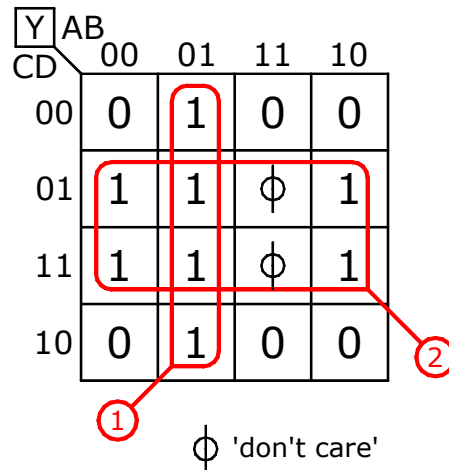
We lezen lus 1 af: $\overline{Y}_1 = A \cdot B$

We lezen lus 2 af: $\overline{Y}_2 = \overline{B} \cdot \overline{D}$

De minimale inverse logische vergelijking wordt: $\overline{Y} = AB + \overline{B}\overline{D}$

3.3.7 Gebruik van DON'T CARE

Bij het ontwerpen van digitale combinatorische logica kan het voorkomen dat niet alle vakken van een karnaugh-map bepaald zijn. Er kunnen toestanden voorkomen die zowel '0' als '1' mogen zijn. Dit zijn de zg. *DON'T CARE's*. Bij het uitlezen mag je zelf beslissen of een '1' of een '0' plaats in het betreffende vak.



△ Fig.3.22 Uitlezen van een karnaugh-diagram voor 4 booleaanse variabelen met 'don't care's'.
Bron: W.Van Wichelen

We lezen lus 1 af: $Y_1 = \bar{A} \cdot B$

We lezen lus 2 af: $Y_2 = D$

De minimale logische vergelijking wordt: $Y = \bar{A} \cdot B + D$

3.3.8 Uitgewerkt voorbeeld

Digitale stemmachine

Een deliberatiecommissie moet beslissen tot het al dan niet geven van een herexamen (Y). De commissie bestaat uit 4 leden: 3 leerkrachten (A, B en C) en de directeur (D). Indien de directeur 'ja' ('1') stemt, dan volstaat nog 1 extra ja-stem om de te delibereren leerling een herexamen te geven. Zegt de directeur 'nee' ('0'), dan moeten alle leerkrachten 'ja' stemmen om onze arme student toch nog zijn herexamen te kunnen geven. Ontwerp een digitale schakeling om de stemming automatisch te laten verlopen.

Waarheidstabel & karnaugh-map

#	A	B	C	D	Y
1	0	0	0	0	0
2	0	0	0	1	0
3	0	0	1	0	0
4	0	0	1	1	1
5	0	1	0	0	0
6	0	1	0	1	1
7	0	1	1	0	0
8	0	1	1	1	1
9	1	0	0	0	0
10	1	0	0	1	1
11	1	0	1	0	0
12	1	0	1	1	1
13	1	1	0	0	0
14	1	1	0	1	1
15	1	1	1	0	1
16	1	1	1	1	1

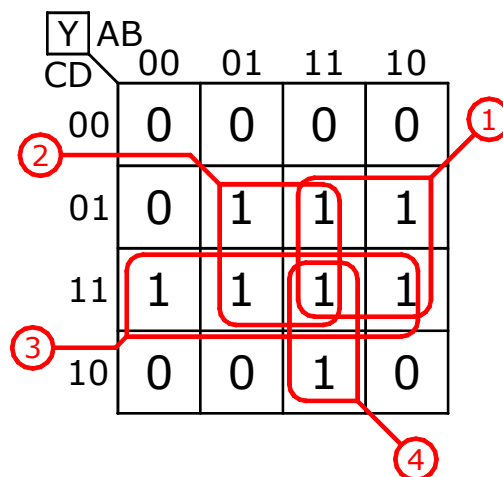
Y	AB	00	01	11	10
CD	00	0	0	0	0
01	0	1	1	1	1
11	1	1	1	1	1
10	0	0	1	0	0

△ Fig.3.23 Truth table en karnaugh-map van de digitale stemmachine. Bron: W.Van Wichelen

Zonder kennis van vereenvoudigingstechnieken zou de schakeling bestaan uit een OR-poort met 8 ingangen waarbij elke ingang connectie maakt met een AND-poort met 4 ingangen. Uit de waarheidstabel leiden we immers volgende vergelijking af:

$$Y = \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BCD + A\bar{B}\bar{C}D + A\bar{B}CD + AB\bar{C}D + ABC\bar{D} + ABCD$$

Minimalisatie voor Y



△ Fig.3.24 Aflezen van de enen in de karnaugh-map van de digitale stemmachine. Bron: W.Van Wichelen

We lezen lus 1 af: $Y_1 = A \cdot D$

We lezen lus 2 af: $Y_2 = B \cdot D$

We lezen lus 3 af: $Y_3 = C \cdot D$

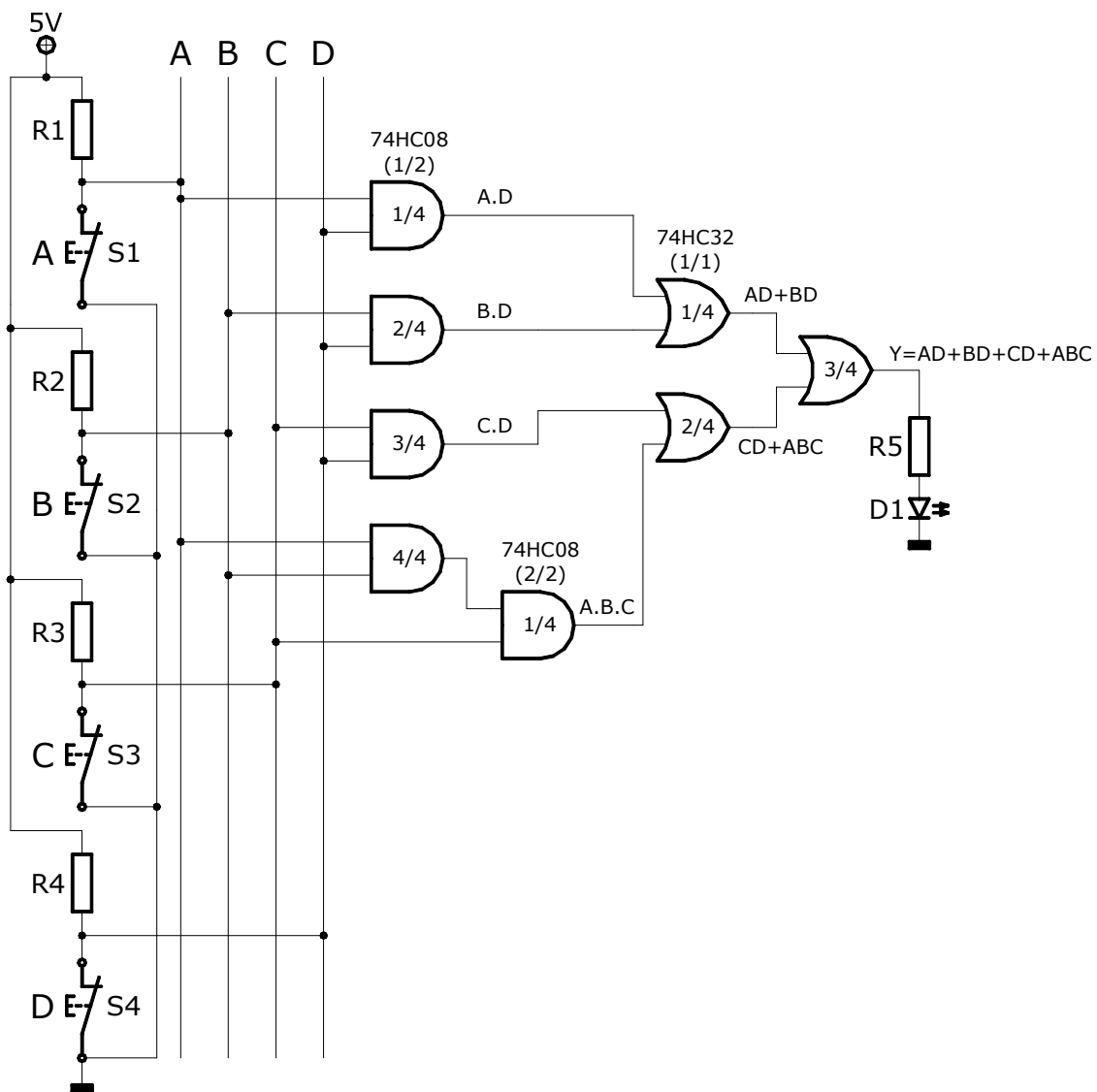
We lezen lus 4 af: $Y_4 = A \cdot B \cdot C$

De minimale logische vergelijking wordt: $Y = AD + BD + CD + ABC$

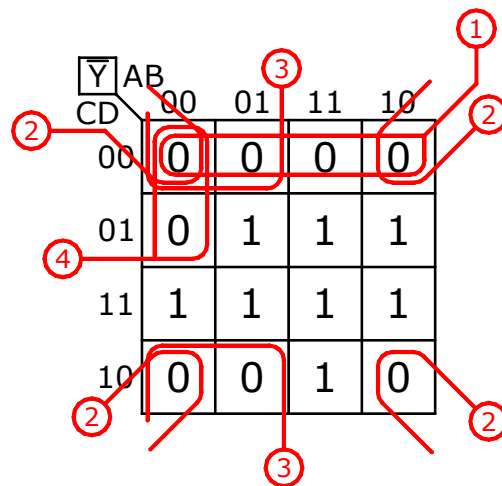
Laboproef 2

Ontwerp een digitale stemmachine om een deliberatie-commissie te automatiseren. De commissie bestaat uit drie vakleerkrachten (A, B en C) en de directeur (D). Wanneer de directeur zich onthoudt, dan beslist de meerderheid van de leerkrachten tot het geven van een herexamen (Y). Wanneer de directeur meestemt dan is de uitkomst steeds een herexamen. Stel de waarheidstabel op en verifieer ze met metingen. Visualiseer de logische toestand van de uitgang met een blauwe LED. Pas het principe van 'current sinking' toe. Maak gebruik van miniatuur drukknoppen (S1,..S4).

Praktische realisatie - current sourcing

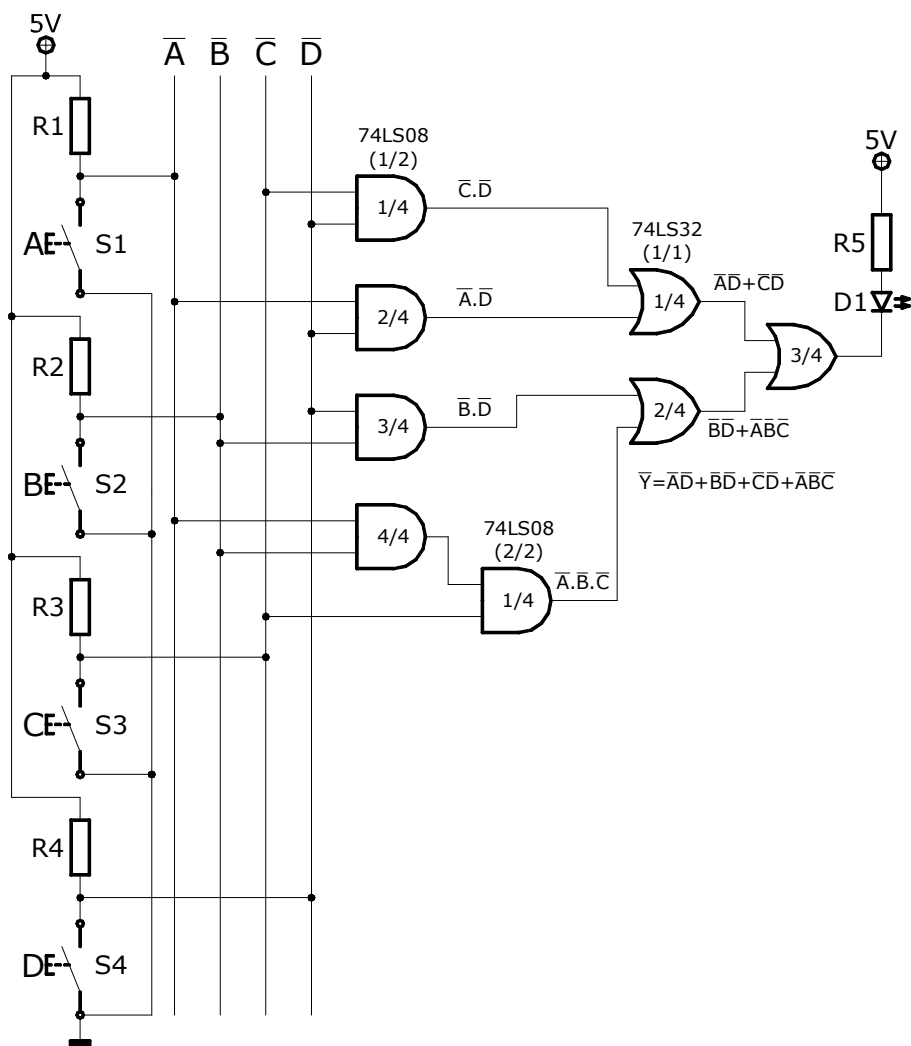


△ Fig.3.25 | Praktische realisatie met CMOS-poorten van de digitale stemmachine. Bron: W.Van Wichelen

Minimalisatie voor inverse functie $\bar{Y}$ 

△ Fig.3.26 | Aflezen van de nullen in de karnaugh-map van de digitale stemmachine. Bron: W.Van Wichelen

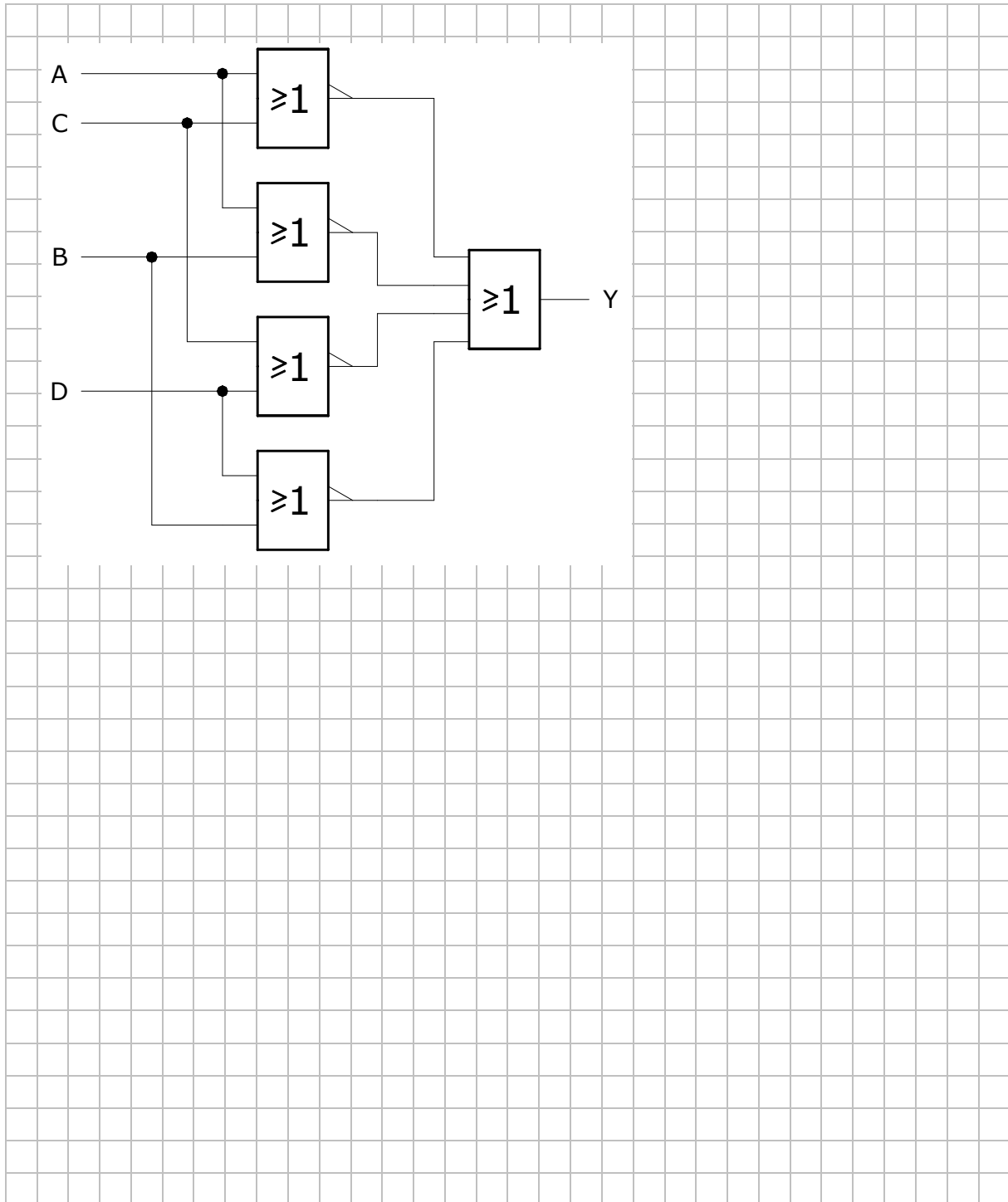
Praktische realisatie - current sinking



△ Fig.3.27 | Praktische realisatie met TTL-poorten van de digitale stemmachine. Bron: W.Van Wichelen

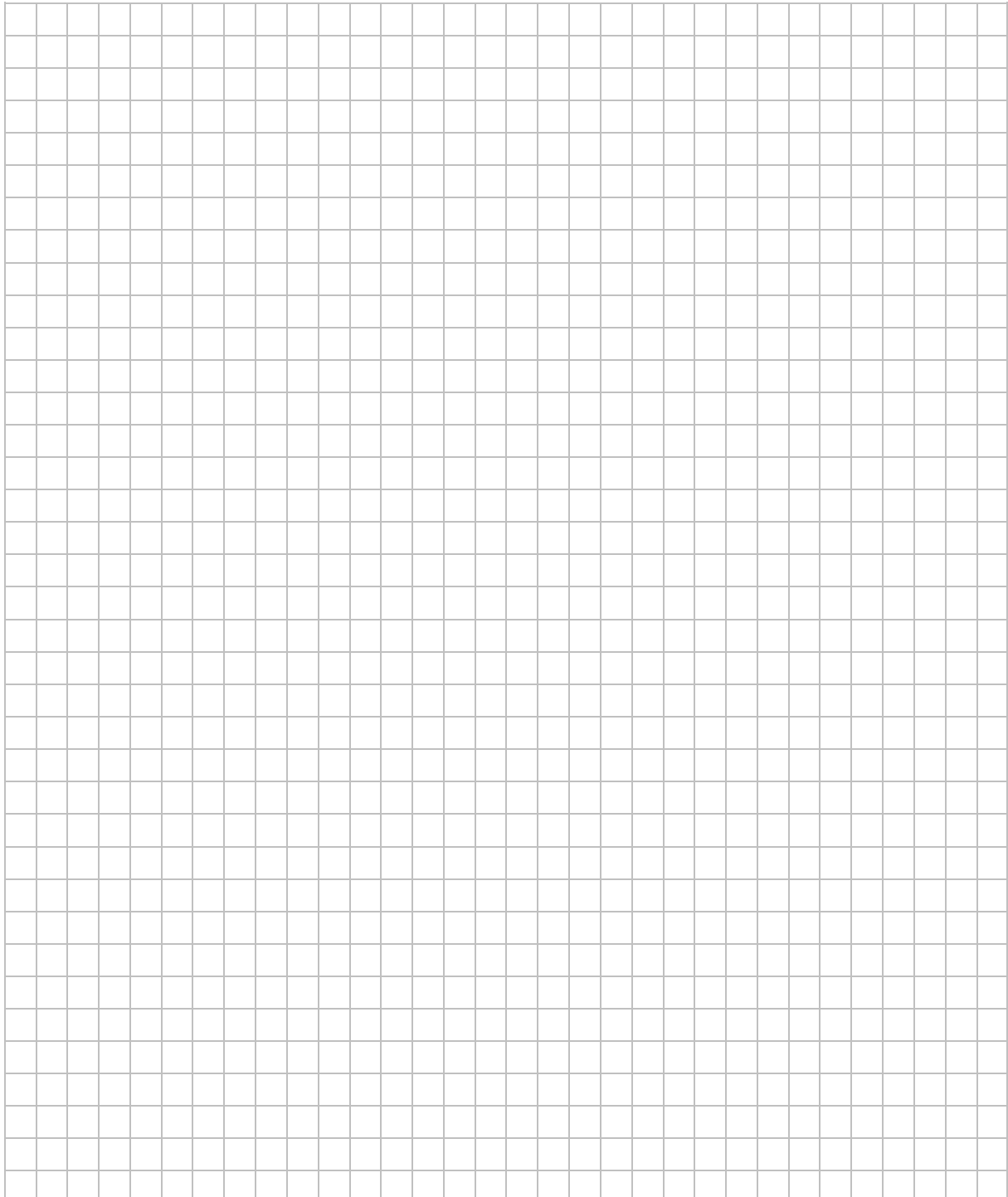
3.3.9 Oefeningen

1. Bestudeer onderstaande logische schakeling.
 - a. Stel de waarheidstabel op.
 - b. Stel de logische vergelijking op.
 - c. Vereenvoudig m.b.v. een karnaugh-map.
 - d. Teken de eenvoudigste realisatie.

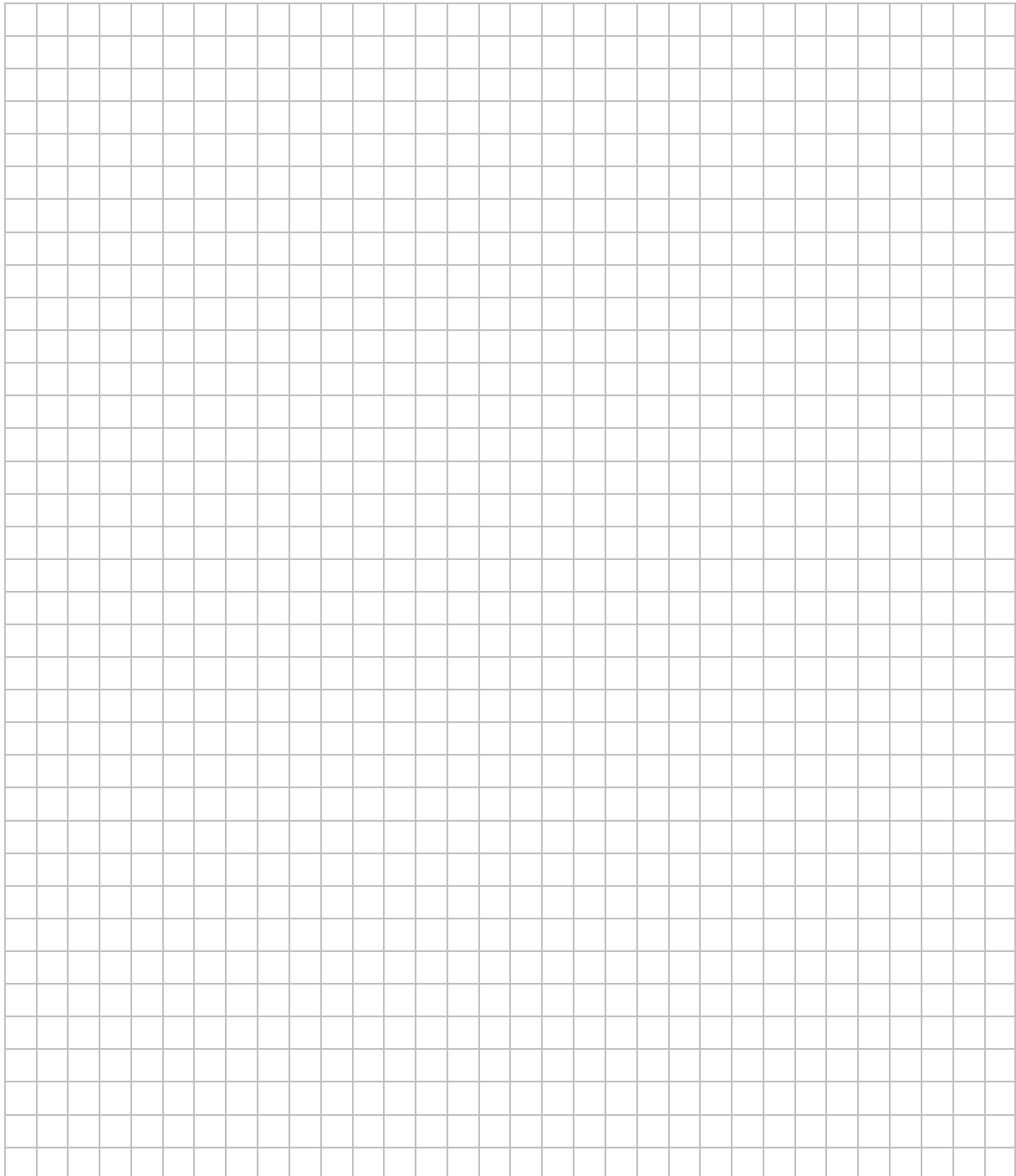




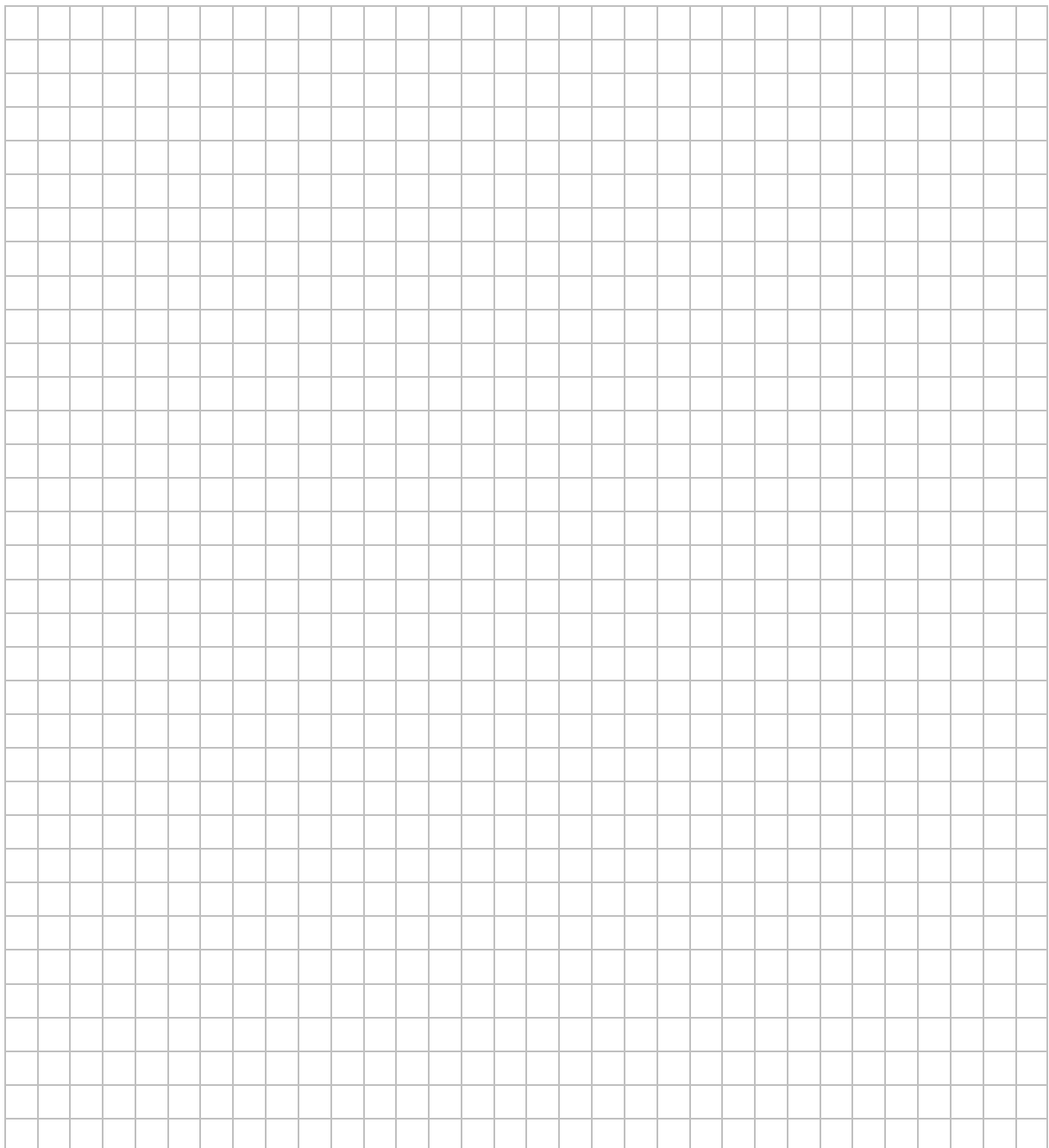
2. Een toestel (Y) wordt bediend door twee drukknoppen (A en B).
Ontwerp een combinatorische NAND-poortschakeling die er voor zorgt dat het toestel enkel en alleen inschakelt als de twee schakelaars een verschillende stand aannemen.
- Stel de waarheidstabel op.
 - Noteer het karnaugh-diagram.
 - Stel de vergelijking op.
 - Realiseer de schakeling met enkel NAND-poorten.



3. Twee verbruikers (X en Y) worden bediend door drie schakelaars (A, B en C). Verbruiker X moet werken zodra A en B ingedrukt zijn. Verbruiker Y moet werken zodra A en C ingedrukt zijn. Ontwerp een combinatorische schakeling die enkel gebruik maakt van het IC 74LS00.
- Stel de waarheidstabel op.
 - Noteer het karnaugh-diagram.
 - Stel de vergelijkingen op.
 - Realiseer de schakeling met enkel NAND-poorten.



4. Een elektronische thermometer meet de temperatuur van een aquarium. Het uitgangssignaal bestaat uit 4 bits en is de binaire voorstelling van de temperatuur van 0°C tot 15°C . Ontwerp een combinatorische schakeling waarvan de uitgang (Y) hoog wordt voor temperaturen hoger dan 9°C en laag wordt voor temperaturen lager dan 10°C . Maak gebruik van de basispoorten met actief hoge ingangen.
- Stel de waarheidstabel op.
 - Maak een karnaugh-map.
 - Stel de vergelijking op.
 - Teken de realisatie met basispoorten.



3.4 Herhalingsvragen

- http://electrivial.atspace.eu/oefenreeksen/reeks11_elektronica.html
- http://electrivial.atspace.eu/oefenreeksen/reeks12_elektronica.html
- http://electrivial.atspace.eu/oefenreeksen/reeks13_elektronica.html

