hogeschool Windesheim

Project SMU

Slimme Meter Uitlezer

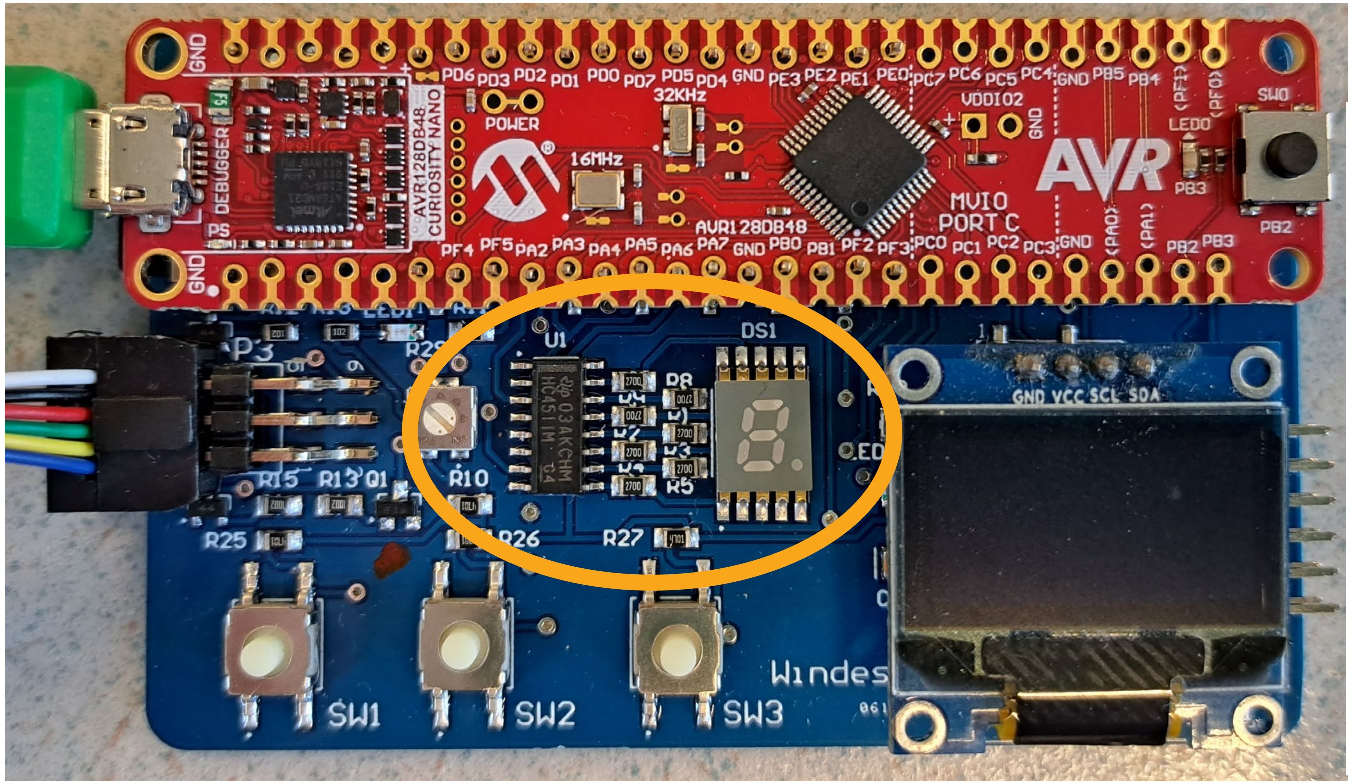
Docenten:

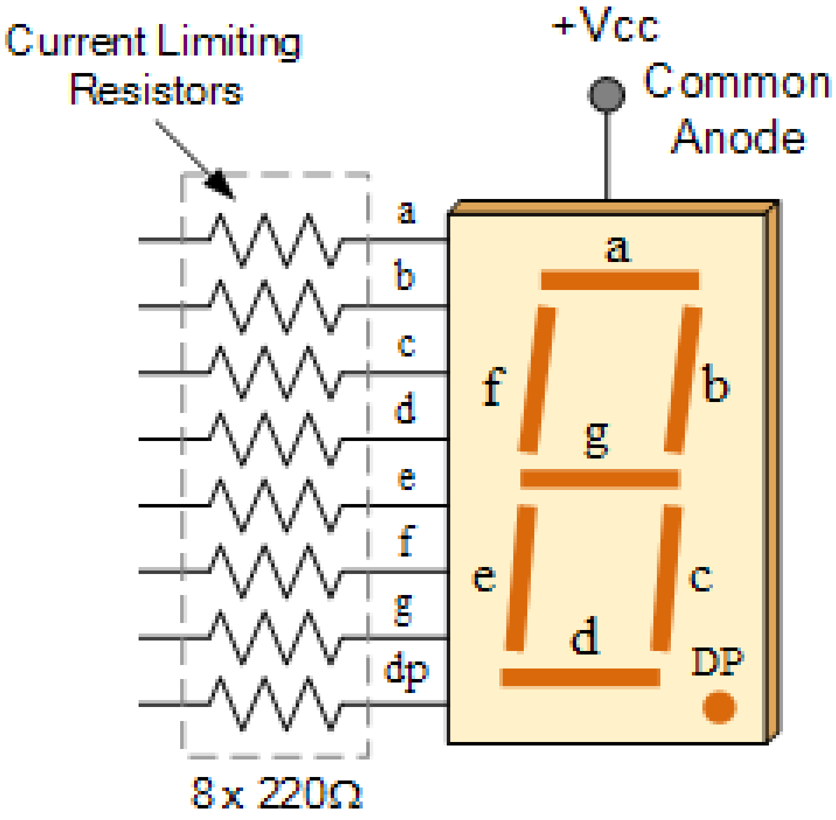
- Jan Kampen: j.kampen@windesheim.nl
- Anke Kuijk: a.kuijk@windesheim.nl
- Richard Rosing: r.rosing@windesheim.nl

Wat gaan we doen?

Week 1:	Kennismaken met de Slimme Meter en de SMU
Week 2:	Het meten van spanning en stroom
Week 3:	Digitale signalen en schakelaars
Week 4:	**Booleaanse algebra en het 7-segmentsdisplay**
Week 5:	De microcontroller programmeren
Week 6:	Seriële communicatie, het OLED display en de SD-kaart
Week 7:	Alles afmaken
Week 8:	Toetsing

Het 7-segmentsdisplay + decoder





Om decoder aan te sturen: binaire getallen

Binair getal	Macht	Decimaal getal
00001	2^0	1
00010	2^1	2
00100	2^2	4
01000	2^3	8
10000	2^4	16

- Elke één in een binair getal komt overeen met een macht van twee.
- Je begint aan de rechterkant met 2^0 en elke plek die je naar links schuift wordt de macht 1 hoger.

Bij een binair getal met meerdere enen tel je de bijbehorende machten bij elkaar op om de decimale waarde te vinden:

$1011 = 2^3 + 2^1 + 2^0 = 8 + 2 + 1 = 11$

Van decimaal naar binair

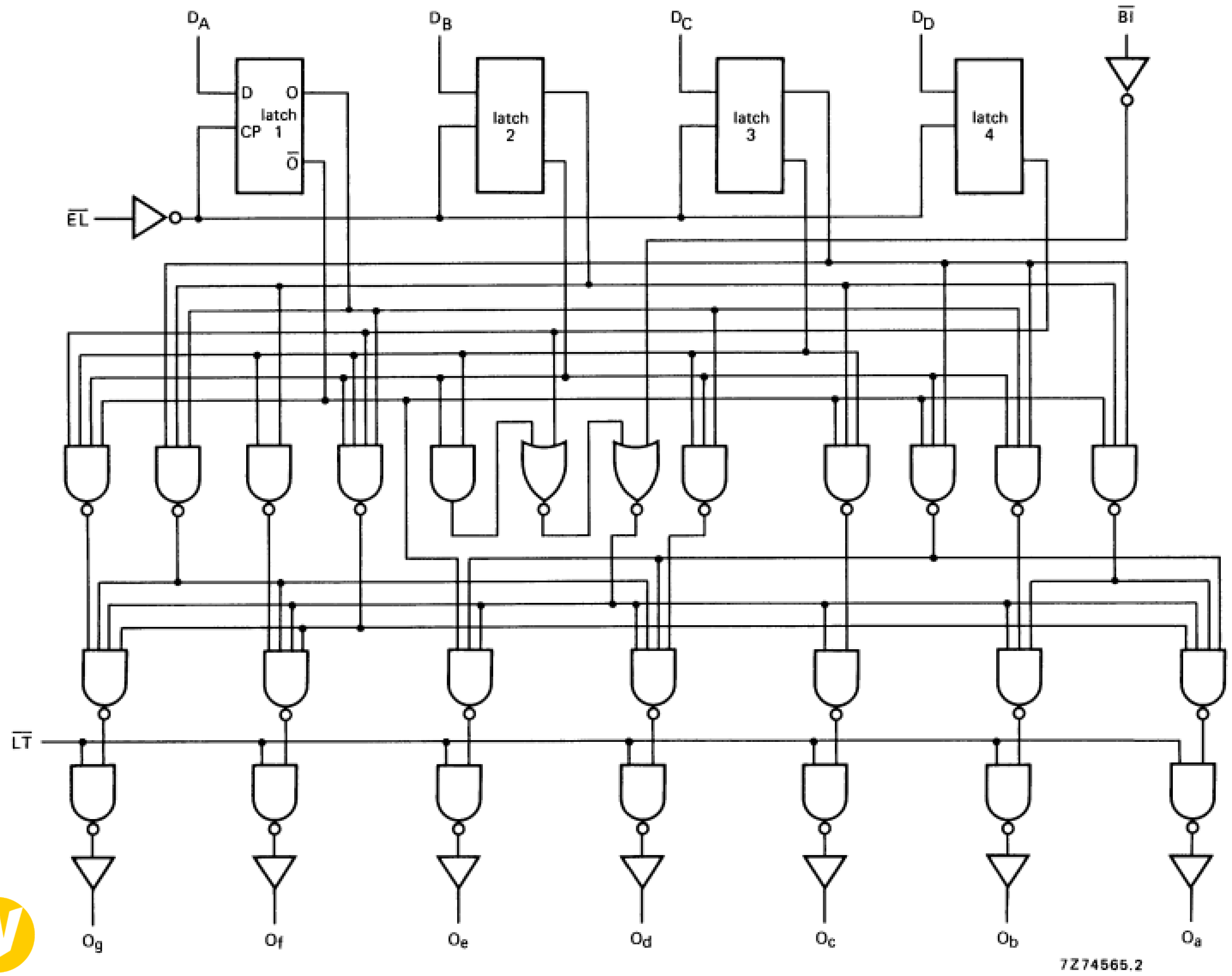
Binair getal	Macht	Decimaal getal
00001	2^0	1
00010	2^1	2
00100	2^2	4
01000	2^3	8
10000	2^4	16

Wil je een decimaal getal omzetten naar een binair getal, dan kijk je welke machten van twee erin passen, te beginnen bij de grootste macht.

Het decimale getal 23 wordt op deze manier:

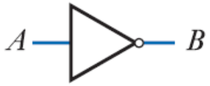
- De grootste macht die in 23 past is 2^4 = 16
- 23 – 16 = 7, de grootste macht die hierin past is 2^2 = 4
- 7 – 4 = 3, hier past 2^1 = 2 in
- 3 – 2 = 1, hier past 2^0 = 1 in.
- $2^4 + 2^2 + 2^1 + 2^0$ = 10111

Binnen in de decoder: combinatorische logica



Een combinatie van de logische poorten die je vorige week geleerd hebt!

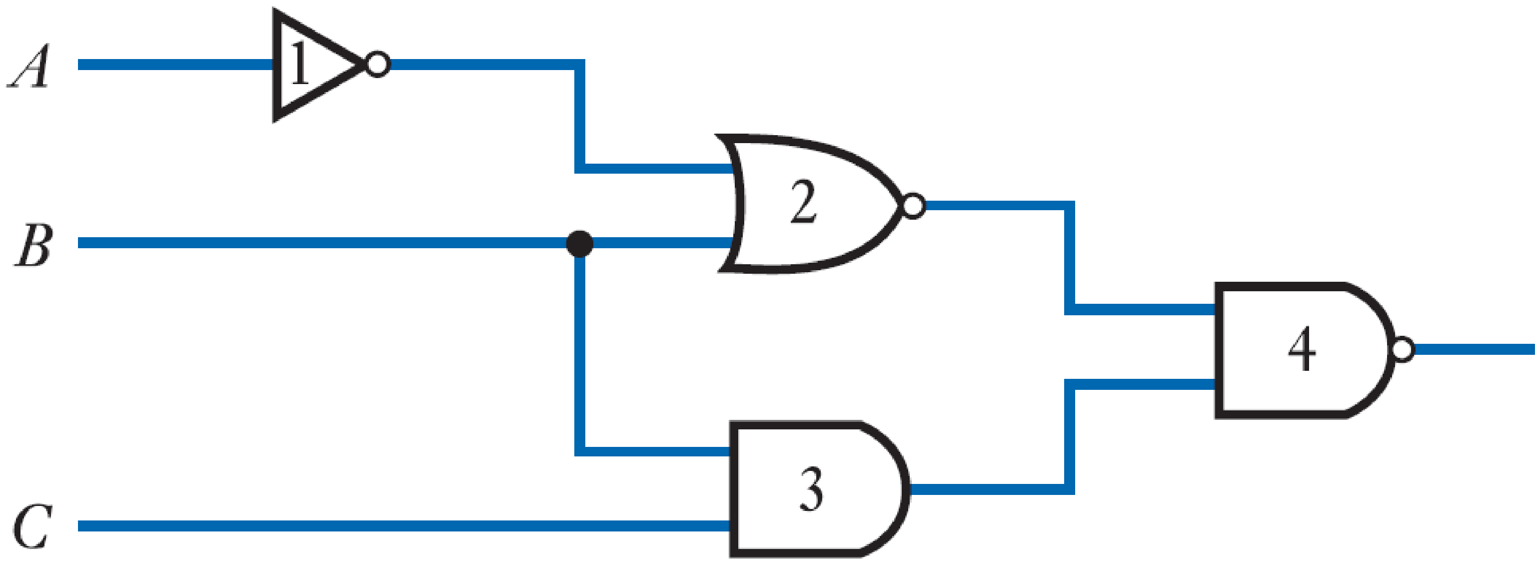
Logische poorten: notatie en functie

Function	Symbol	Alternative symbol	Boolean expression	Truth table
Buffer		1	$B = A$	A | B 0 | 0 1 | 1
NOT		1	$B = \overline{A}$	A | B 0 | 1 1 | 0
AND		&	$C = A \cdot B$	A B | C 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		≥1	$C = A + B$	A B | C 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1

Function	Symbol	Alternative symbol	Boolean expression	Truth table
NAND		&	$C = \overline{A \cdot B}$	0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR		≥1	$C = \overline{A + B}$	A B | C 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0
Exclusive OR		=1	$C = A \oplus B$	A B | C 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0
Exclusive NOR		=1	$C = \overline{A \oplus B}$	A B | C 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

Logisch schema en Booleaanse expressies

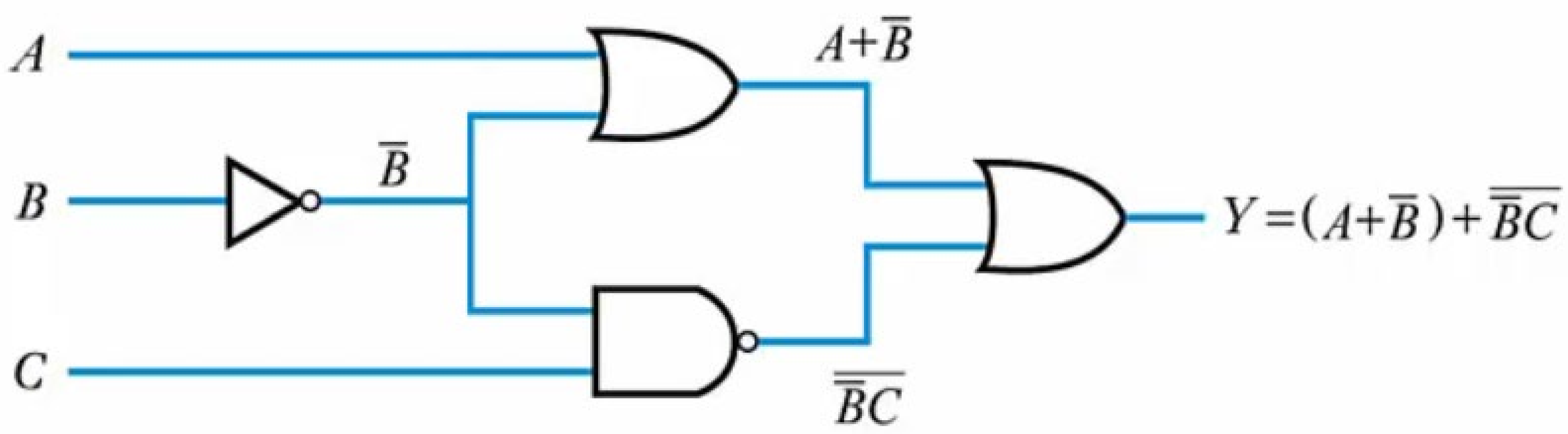
Van schema naar expressie



Logisch schema en Booleaanse expressies

Van expressie naar schema:

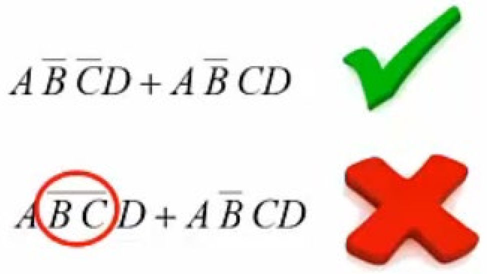
- Let op de volgorde van de Booleaanse algebra!
  (haakjes → AND-poorten → OR-poorten)
- Let op de lengte van de streepjes!

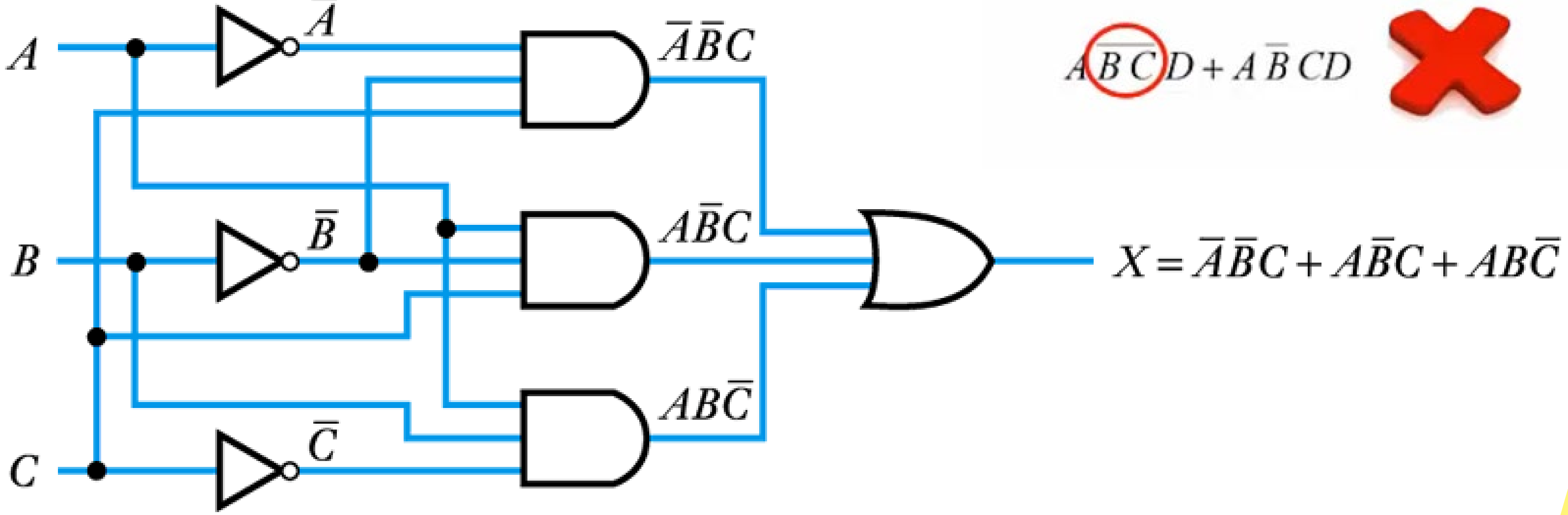


Logisch schema vanuit een waarheidstabel

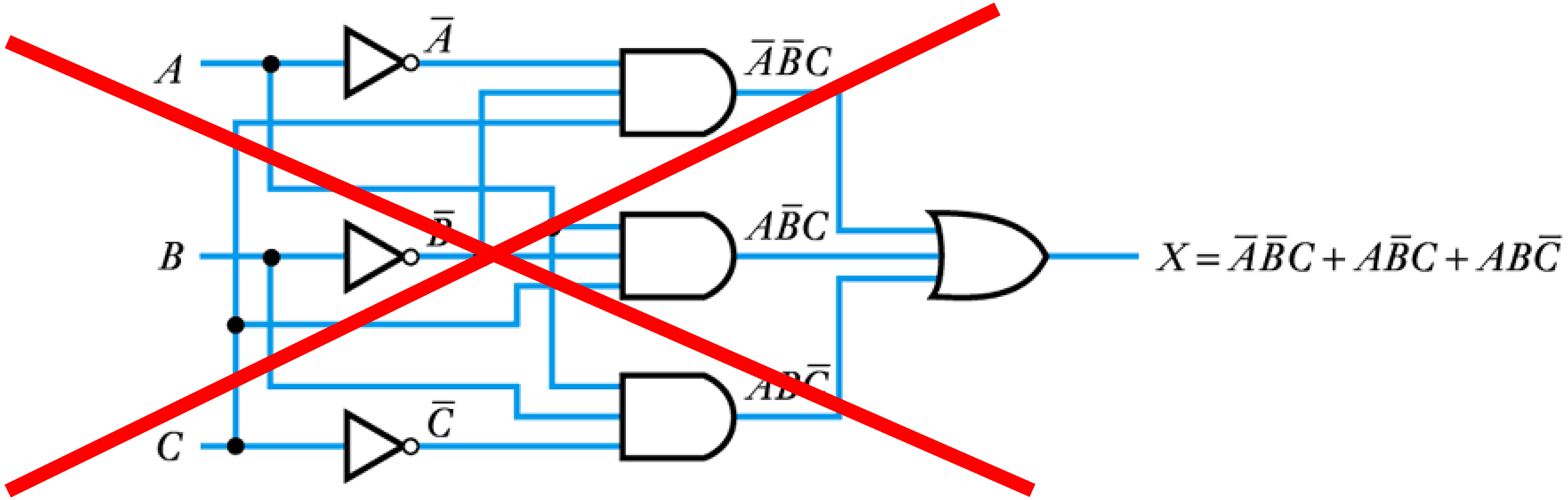
A	B	C	X	
0	0	0	0	
0	0	1	1	$\bar{A}\bar{B}C$
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	1	$A\bar{B}C$
1	1	0	1	$AB\bar{C}$
1	1	1	0	

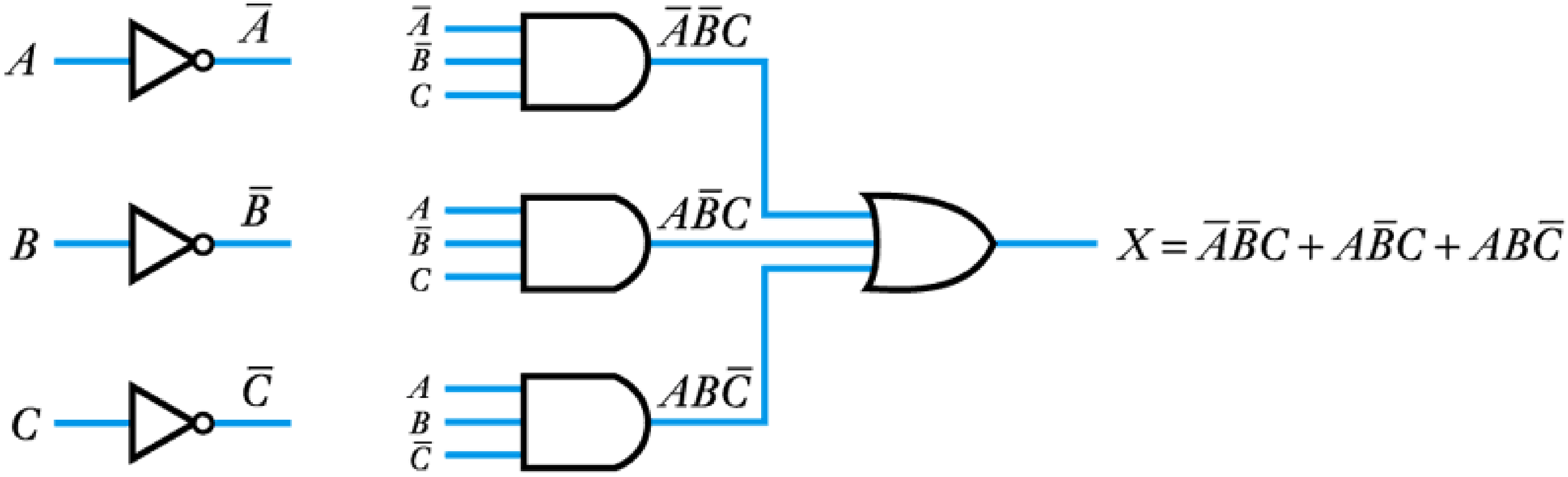
1. Schrijf de mintermen op bij de regels waarvan de output gelijk is aan ‘1’.
2. De bijbehorende Booleaanse expressie is de som van deze mintermen: $X = \bar{A}\,\bar{B}C + A\bar{B}C + AB\bar{C}$





Notatie zonder spaghetti:





Schakelalgebra

Wanneer we vanuit een waarheidstabel een Booleaanse vergelijking opstellen, wordt dit altijd een lange som van mintermen:

$$E = B\overline{C}\,\overline{D} + \overline{A}BD + ABD + BC\overline{D} + \overline{B}CD + \overline{A}\,\overline{B}\,\overline{C}D + A\overline{B}\,\overline{C}D$$

Het is mogelijk deze vergelijking korter en overzichtelijker te maken met behulp van Booleaanse wetten en regels.

Bijkomend voordeel: hoe korter de vergelijking, hoe minder componenten er nodig zijn om de schakeling te bouwen!

Boolean identities

AND function	OR function	NOT function
$0 \cdot 0 = 0$	$0 + 0 = 0$	$\overline{0} = 1$
$0 \cdot 1 = 0$	$0 + 1 = 1$	$\overline{1} = 0$
$1 \cdot 0 = 0$	$1 + 0 = 1$	$\overline{\overline{A}} = A$
$1 \cdot 1 = 1$	$1 + 1 = 1$	
$A \cdot 0 = 0$	$A + 0 = A$	
$0 \cdot A = 0$	$0 + A = A$	
$A \cdot 1 = A$	$A + 1 = 1$	
$1 \cdot A = A$	$1 + A = 1$	
$A \cdot A = A$	$A + A = A$	
$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$	

Boolean laws

Commutative law	**Absorption law**
$AB = BA$	$A + AB = A$
$A + B = B + A$	$A(A + B) = A$
Distributive law	**De Morgan's law**
$A(B + C) = AB + AC$	$\overline{A + B} = \overline{A} \cdot \overline{B}$
$A + BC = (A + B)(A + C)$	$\overline{A \cdot B} = \overline{A} + \overline{B}$
Associative law	**Note also**
$A(BC) = (AB)C$	$A + \overline{A}B = A + B$
$A + (B + C) = (A + B) + C$	$A(\overline{A} + B) = AB$

Schakelalgebra

We gaan de volgende vergelijking vereenvoudigen:

$$E = B\overline{C}\,\overline{D} + \overline{A}BD + ABD + BC\overline{D} + \overline{B}CD + \overline{A}\,\overline{B}\,\overline{C}D + A\overline{B}\,\overline{C}D$$

1. Neem termen die maar één variabele verschillen samen:
   $E = B\bar{D}(\bar{C} + C) + BD(\bar{A} + A) + \bar{B}\bar{C}D(\bar{A} + A) + \bar{B}CD$
2. Gebruik de wet: $\bar{A} + A = 1$
   $E = B\bar{D} + BD + \bar{B}\bar{C}D + \bar{B}CD$
3. Herhaling van stappen:
   $E = B(\bar{D} + D) + \bar{B}D(\bar{C} + C) = B + \bar{B}D$
4. Gebruik de wet: $A + \bar{A}B = A + B$
   $E = B + D$

Aan de slag!

- Ga naar leren.windesheim.nl

  (zoek de cursus EDPD.22, project SMU)

- Voer de opdrachten van week 4 uit.

- Ben je klaar? Ga vast verder met de voorbereiding van week 5!