Boolesk Algebra og det binære talsystem - temahæfte informatik

Dette temahæfte introducerer to-talsystemet og logiske udtryk (Boolesk algebra). Vi oplever, at de almindelige regneregler også gælder i to-talsystemet, og vi prøver at nedskrive simple apparaters opførsel vha. booleske udtryk.

På en digital indgang kan en computer kun se forskel på, om en kontakt er tændt eller slukket. Men til gengæld er den hurtig og god til at regne.

Boolesk Algebra - oprindelse

I 1854 var der en engelsk matematiker og logiker ved navn George Boole (1815-1864) der lavede en algebraisk beskrivelse af logiske love - det der i dag kaldes Boolesk Algebra. George Boole var autodidakt (senere matematik professor ved Queens College i Irland) og det var diskussioner med filosoffer og logikere, der fik Boole i gang med at opstille de logiske regler.

1 Det binære talsystem (to-talssystemet)

1.1 Det binære talsystem og dets opbygning

Et simpelt regnestykke som 6 + 9 = 15 kan en computer udføre på et nano-sekund. Mennesket bruger ti-talsystemet når vi regner - tallene fra nul til ni. Men i sin grundform kan en computer kun forstå to tal, nemlig 0 og 1. Så for at computeren kan udføre regnestykket, bliver 6 og 9 lavet om til to-talsystemet. Computeren finder et facit og konverterer det fra to-talsystem tilbage til ti-talsystemet, så vi kan forstå det.

Når vi skal forstå computerens måde at regne på, er det godt lige at tænke lidt over hvordan vi selv gør når vi regner.
Vi begynder med ti-talsystemet. Når man skriver 3146 i vores normale ti-talsystem betyder det, at der er seks enere, fire tiere, en hundrede, og tre tusinder. Lægges de disse tal sammen, giver det 3146.

På skemaform kan det skrives på følgende vis:

1000	100	10	1
3	1	4	6

3000 + 100 + 40 + 6 = 3146

Et ciffers placering afgør dets betydning

På tilsvarende måde er det i to-talsystemet. Her tælles ikke fra 0 til 9, men fra 0 til 1. Et ciffer kan have værdien 1 eller 0. I ti-talsystemet kan et ciffer have værdien 0, 1, 2, 3, 4, 5, 6, 7, 8 eller 9. I to-talsystemet er der kun; 0 eller 1 - nul eller en - falsk eller sandt - high eller low - lys eller mørke - tændt eller slukket

Kun 0 eller 1. Tallet 6 i ti-talsystemet skrives som 0110 i to-talsystemet. To-talsystemet kaldes også det *binære talsystem* (binær betyder "sammensat af to"). Tallet 3 i ti-talsystemet skrives som 0011 i det binære talsystem.

1.1.1 Konvertering fra ti-talsystem til to-talsystem

Et tal i to-talsystemet konverteres til ti-talsystemet ved at indsætte to-tal-systems tal i tabellen herunder. Derefter lægges kolonneoverskrifterne sammen de steder, hvor der er "1" i cellen.
Således bliver tallet 0000|0101
i to-talsystemet til 5 i ti-talsystemet.

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	1

0 + 0 + 0 + 0 + 0 + 4 + 0 + 1 = 5

I ti-talsystemet hedder et enkelt tal et ciffer. I to-talsystemet kaldes det et bit. Normalt arbejdes der i to-talsystemet med 8 bit ad gangen - det kaldes en *byte*. 1 gigabyte er med andre ord 8 gigabit.

Der er placeret en lodret streg mellem de øverste 4 bit og de nederste fire bit. Det er gjort for at øge læsbarheden, men er ellers uden betydning for tallets værdi.

Øvelse - sidder du fast så snak med sidemanden
Konverter følgende binære tal til ti-talsystem

a) 0000 | 0110

128	64	32	16	8	4	2	1

b) 1100 | 0110

128	64	32	16	8	4	2	1

c) 0000 | 0000

128	64	32	16	8	4	2	1

d) 1111 | 1110

128	64	32	16	8	4	2	1

Øvelse - sidder du fast så snak med sidemanden
Konverter følgende decimaltal, til tal i to-talsystemet.

a) 4

128	64	32	16	8	4	2	1

b) 17

128	64	32	16	8	4	2	1

c) 130

128	64	32	16	8	4	2	1

d) 61

128	64	32	16	8	4	2	1

e) 10

128	64	32	16	8	4	2	1

r) 0

128	64	32	16	8	4	2	1

g) 254

128	64	32	16	8	4	2	1

h) 85

128	64	32	16	8	4	2	1

1.1.2 Binær addition

To binære tal lægges sammen på gammeldags måde ved at stille dem over hinanden

	0000 | 0001	altså	1
+	0000 | 0100		+ 4
=	0000 | 0101		= 5

Øvelse - sidder du fast så snak med sidemanden

Læg følgende tal sammen.

a) 0000 | 0110 + 0010 | 1000

	0	0	0	0	0	1	1	0
+	0	0	1	0	1	0	0	0
=								

b) 0100 | 0100 + 1010 | 1001 (sæt selv tallene ind i skemaet)

+								
=								

c) 0100 | 0101 + 1000 | 1000

+								
=								

Addition med "carry" (mente)

Når vi lægger to store tal sammen på den gammeldags metode (fra folke-skolen) skubber vi en ti'er, hvis resultatet af to cifre bliver for stort. Det kalder vi mente (carry på engelsk). Man begynder bagfra:

1)	2)	3)	4)	5)
		NB!	+1	+1
1625	1625	1625	1625	1625
+ 5621	+ 5621	+ 5621	+ 5621	+ 5621
= 6	= 46	= 246	= 246	= 7246

Øvelse - sidder du fast så snak med sidemanden

a) Læg følgende tal sammen på den gammeldags metode.
Begynd bagfra - tjek efterfølgende svaret med din lommeregner:

3 6 5 6 9	6 9 5 6 8	1 2 3 5 8 6
+ 5 5 3 2 0	+ 4 6 3 5 1	+ 6 8 0 4 8 1
=	=	=

Øvelse - læg nu følgende 2-talsystems tal sammen - "med carry"

a) 0001 | 0110 + 0010 | 0100 (sæt selv tallene ind i skemaet)

+								
=								

b) 0110 | 0110 + 0010 | 1011

+								
=								

1.1.3 Subtraktion

Metoden til at subtrahere (minus-stykker) i ti-talsystemet kan også anven-des i to-talsystemet. Man starter bagfra og så "låner" vi fra naboen. I to-talsystemet får man to, når man låner.

1)	2)	3)	to-tal systemet
	10 (vi låner ti)	10	10 (vi låner to stk)
2729	2~~7~~29	2~~7~~29	0000 | 1~~1~~01
- 1538	- 1538	- 1538	- 0000 | 0010
= 1	= 91	= 1191	= 0000 | 1011

Øvelse - træk følgende to tal fra hinanden

a) 0011 | 0110 - 0010 | 0100 (sæt selv tallene ind i skemaet)

-								
=								

b) 0110 | 0101 - 0010 | 0011

-								
=								

c) 1100 | 1011 - 0001 | 0101

-								
=								

Vigtig viden eller ligegyldig info

Det mest betydende ciffer i et to-talsystem, er bittet til venstre. Det betegnes *MSB,* som er en forkortelse af Most Significant Bit. I et 8-bit system, er bittet til højre, bit nummer 0. Derfor bliver 128-markøren til bit nummer 7. Bit 0 benævnes *LSB* (Least Significant Bit).

MSB							LSB
7	6	5	4	3	2	1	0
128	64	32	16	8	4	2	1

1.1.4 To's komplement - negative tal i to-talsystemet

I ti-talsystemet anvender vi et minustegn foran et tal, for at angive, at det er negativt.

+ 45 er et positivt tal   - 45 er et negativt tal

Ved -45 bruges ciffer tre's plads til fortegnet. Ved f.eks. -6318 er fortegnet placeret på ciffer fem's plads.

Det er også muligt i en computer med to-talsystemet. Men det er lidt bøvlet at detektere, hvilket bit der er fortegn. I nogle systemer anvender man derfor bit 7, som et fast fortegnsbit. Når bit 7 er 1, er der tale om et negativt tal.

En mere benyttet metode kaldes *to's komplement*. Den er hurtig for computeren at bruge, og samtidig muliggør metoden subtraktion ved at addere tallene. Det udnyttes at 60 - 45, kan skrives som +60 +(-45).

Her er et eksempel med tallet 45 i to-talsystemet:

128	64	32	16	8	4	2	1
0	0	1	0	1	1	0	1

0 + 0 + 32 + 0 + 8 + 4 + 0 + 1 = 45

Når to's komplement skal findes, toggles hvert bit, og der lægges 1 til. At *toggle* et bit, betyder at det inverteres - bittet vendes. 0 bliver til 1.

Ved at toggle bit 1 i ovennævnte tal, bliver værdien ændret fra 1 til 0. Her er alle bittene blevet togglet:

128	64	32	16	8	4	2	1
1	1	0	1	0	0	1	0

For at færdiggøre konverteringen skal der lægges 1 til. Her er to's komplement af det binære tal 45.

128	64	32	16	8	4	2	1
1	1	0	1	0	0	1	1

Det smarte er, at når de to tal lægges sammen fås resultatet af minus-stykket. 60-45 laves som:

+ to's komplement af 45:	+ 1101 | 0011
+ 60	+ 0011 | 1100
= 15	= 1 0000 | 1111

De nederste 8 bit af resultatet giver 15.

Øvelse - find to's komplement af følgende tal (først toggle, så + 1)

a) 36	0	0	1	0	0	1	0	0
komplement								

b) 70	0	1	0	0	0	1	1	0
komplement								

c) 0	0	0	0	0	0	0	0	0
komplement								

Bemærk at to's komplement af 0, bliver 0.

Øvelse - træk følgende to tal fra hinanden

a) 0011 | 0110 - 0010 | 0100

Find først to's komplement af det tal der trækkes fra (0010 | 0100).

tal	0	0	1	0	0	1	0	0
komp								

Læg derefter det første tal (0011 | 0110) til

komp								
+	0	0	1	1	0	1	1	0
=								

b) 0110 | 0101 - 0010 | 0011 (begynd med at finde to's komplementer)

tal	0	0	1	0	0	0	1	1
komp								
+								
=								

c) 1100 | 1011 - 0001 | 0101 (sæt selv tallene ind i skemaet)

tal								
komp								
+								
=								

Tjek resultaterne med øvelserne s. 5 - det er de samme regnestykker, så de resultatet skulle helst blive det samme.

1.1.5 Multiplikation

Multiplikation (gange-stykker) forløber problemfrit i forhold til regne-reglerne. At gange med 0 eller 1 er enkelt - det er de samme regler der anvendes i ti-talsystemet.

0000|0100 • 0000|0000 = 0 (4 • 0 = 0)

0000|0100 • 0000|0001 = 0000|0100 (4 • 1 = 4)

Skal der ganges med 2, svarer det til at lægge det oprindelige tal sammen med sig selv.

0000|0100 • 0000|0010 = 0000|1000 (4 • 2 = 8)

	0000 | 0100	altså	4
+	0000 | 0100		+ 4
=	0000 | 1000		= 8

Når der ganges med 3, så skal det oprindelige tal lægges sammen med sig selv, tre gange.

I computeren udnytter man dette forhold. Hvis der skal ganges med 2, skiftes de enkelte bit én gang til venstre (left shift). Hele tallet rykkes til venstre.

4	0000|0100	(4)
4•2	0000|1000	(8)
4•3	0001|0000	(16)
4•4	0010|0000	(32)

Det ene bit rykkes et hak til venstre, hver gang tallet ganges med 2

Det er på samme vis kommaet flyttes i normale regnestykker, når der ganges med ti - cifferene rykkes en plads til venstre.

Eksempel fra ti-talsystemet: 123,45 • 10 = 1234,5 (kommaet er flyttet en plads til højre).

Her er et eksempel fra to-talsystemet, hvor et tal ganges med to, ved at bittene rykkes en plads venstre (*left shift*):

0000|0100
0 0000|100 der sætters "0" ind på den tomme plads til højre

Så svaret bliver: 0 0000 | 1000

Omvendt, når der skal deles med to, skiftes tallet en gang til højre (*right shift*).

Vigtig viden eller ligegyldig info

......there are only 010 kind of people: Those who understand binary and those who don’t.

2 Boolesk Algebra

Foruden de normale regneregler i to-talsystemet findes der også Booleske regneregler - *Boolesk algebra*. Disse regneregler er en stor hjælp, når en computer skal programmeres.

2.1 Oprindelse

I 1854 var der en engelsk matematiker og logiker ved navn George Boole (1815-1864), der lavede en algebraisk beskrivelse af logiske love - det der i dag kaldes Boolesk Algebra. George Boole var autodidakt (senere matematik professor ved Queens College i Irland), og det var diskussioner med filosoffer og logikere, der fik Boole i gang med at opstille de logiske regler.

2.2 Logik

Boolesk algebra er logik-baseret. Der tales om "udsagn". Et udsagn kan være sandt eller falsk. Sandt/rigtigt kan symboliseres ved lys i en lampe på en sandhedsdetektor (ved løgn eller forkert svar, forbliver lampen slukket)

Et falsk udsagn: Peter kan trække vejret under vandet. 

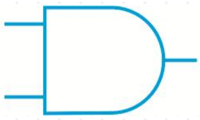
Et sandt udsagn: Peter kan drikke en cola. 

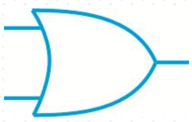
Morlille kan ikke flyve 

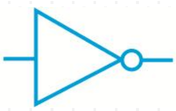
Simplificeret kan man sige, at det eneste en computer gør, er at kigge på en række udsagn (nogle input) og derefter udføre en ordre (output), der er afhængig af udsagnet.

2.3 Boolesk Algebra og gates

Når der tegnes rutediagrammer, har hver kasse/symbol sin egen betydning. På sammen måde anvendes der i Boolesk algebra en række symboler, der repræsenterer forskellige Booleske regler (logiske operatorer). Det giver et godt overblik symbolerne, og ved at kombinere symbolerne, kan man kombinere forskellige logiske regler.
Der er grundlæggende tre Boolesk e udtryk. Til hvert udtryk hører et symbol og en sandhedstabel.

AND A B  Q

OR A B  Q

Inverter A  Q (også kaldet NOT gate)

Gates virkemåde

Udgangen af en gate er enten ”0” eller ”1” (falsk eller sandt).
Udgangen kaldes Q.
Det er indgangsforholdene (kaldet A, B) der bestemmer hvad udgangen skal være. Indgangen kan være ”0” eller ”1”.

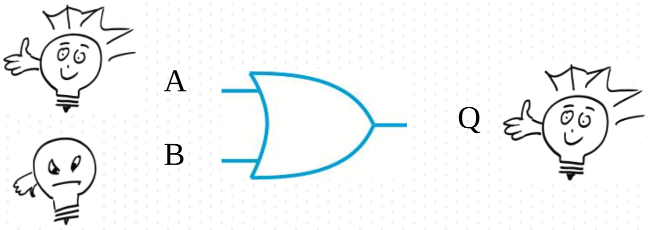
2.3.1 OR-gate

Udgangen, Q på en OR gate er altid 1 - på nær når begge indgange er 0.

Med andre ord, så er udgangen på en OR gate 1, når indgang A eller indgang B er 1.

På engelsk bliver det A or B, deraf navnet OR-gate.

Hvis A = 1 og B = 0, bliver Q = 1:



Det modsatte er også tilfældet. Hvis A = 0 og B = 1, så bliver Q = 1 (hvis A OR B er 1, bliver Q = 1). De forskellige kombinationer af A og B kan skrives op i en sandhedstabel.

B	A	Q
0	0	0
0	1	1
1	0	1
1	1	1

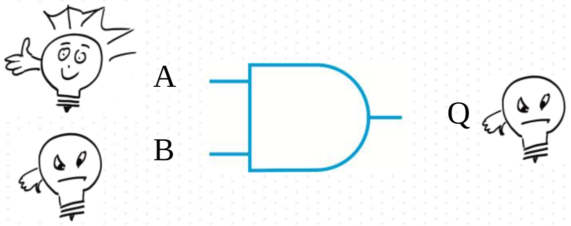
Sandhedstabel for en OR gate

2.3.2 AND-gate

Tilsvarende haves en kombination, hvor både A og B skal være 1, før ud-gangen, Q også bliver 1.

Med andre ord; udgangen på en AND gate altid er 0, på nær når A og B = 1
Det bliver på engelsk til A and B

Hvis A = 1 og B = 0, bliver Q = 0 (både A og B skal være 1, før Q bliver 1):



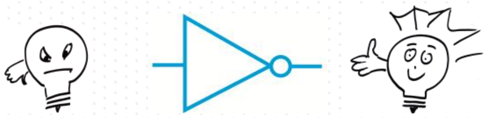
Modsat, hvis A = 0 og B = 1, forbliver Q = 0. De forskellige kombinationer af A og B kan skrives op i en sandhedstabel.

B	A	Q
0	0	0
0	1	0
1	0	0
1	1	1

Sandhedstabel for en AND gate

2.3.3 NOT - Inverter

Den sidste gate-type er en inverter. En inverter inverterer - med andre ord så vender Inverteren indgangssignalet. Hvis indgangen er 1 bliver udgangen 0. Hvis indgang er 0, bliver udgangen 1.



A	Q
0	1
1	0

Sandhedstabel for en Inverter

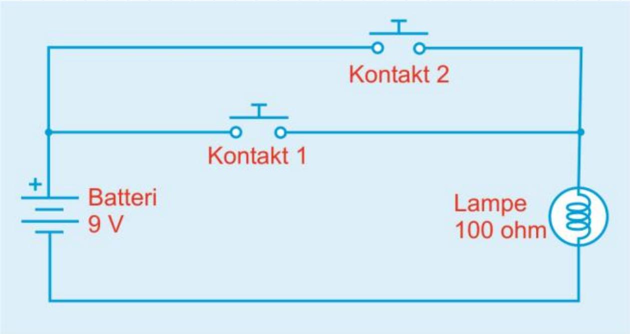
Udgangen er det modsatte af indgangen. Så hvis A er 1, så er Q "not 1".
I denne situation kaldes det også for ”not A”.
not A skrives som $\overline{A}$. A med en bjælke over. En bjælke over et boolesk udtryk betyder at udtrykket skal inverteres.

2.3.4 Sandhedstabeller vist med el-diagram

De booleske gates kan også vises med et simpelt el-diagram. Forestil dig en lommelygte med et batteri, en kontakt og en lampe. Når der trykkers på knappen (kontakten) lyser lampen.

Summeøvelse - snak med sidemanden

Kig på el-diagrammet. Når der trykkes på en af kontakterne (f.eks. Kontakt 1), lyser lampen.



El-diagrammet viser en lommelygte med batteri, to kontakter og en lampe

Hvis der trykkes på den anden kontakt, Kontakt 2 lyser lampen også.
Hvis der trykkes på begge kontakter samtidig, lyser lampen.
Er ingen kontakter trykkede, er der ingen lys.

Kredsløbet kan sammenlignes med en gate og der kan laves en sandheds-tabel. Når en kontakt aktiveres (når der trykkes på en kontakt), svarer det til at indgang A på en gate er 1.

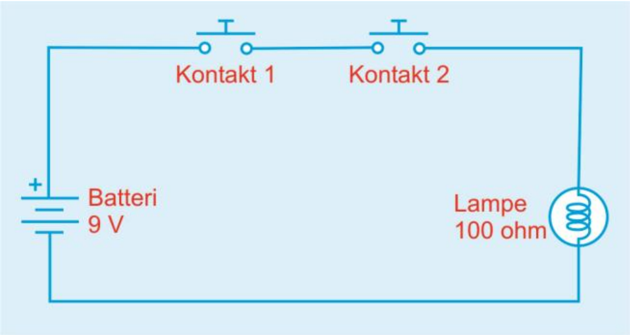
a) Prøv de forskellige kontaktkombinationer og skriv resultatet i tabellen

B	A	Q
0	0	
0	1	
1	0	
1	1	

b) Hvilken gate opfører sig på tilsvarende måde?

Summeøvelse - snak med sidemanden

Når man trykker på Kontakt 1, sker der ingen ting (ingen lys).
Hvordan får man lys i pæren?



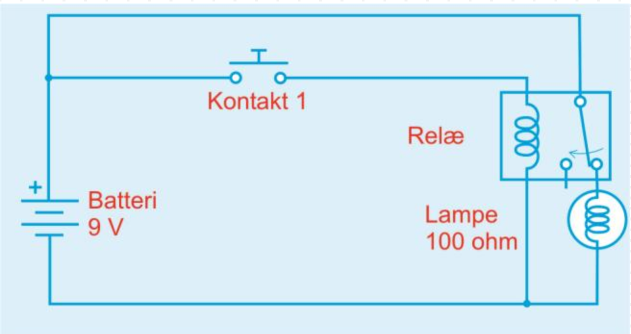
a) Prøv de forskellige kontaktkombinationer og skriv resultatet i tabellen.

B	A	Q
0	0	
0	1	
1	0	
1	1	

b) Hvilken gate opfører sig på tilsvarende måde?

Inverter kredsløbet lavet med et mekanisk relæ

Herunder er vist et kredsløb, der simulerer en inverters opførsel.
Pæren lyser hele tiden - selvom Kontakt 1 ikke er trykket ned.



En inverters virkemåde kan vises ved brug af et omskifterrelæ

Når man trykker på Kontakt 1, aktiveres relæet (Relæ) og omskifter-kontak-ten inde i relæet svinger væk fra lampen.
Det er lidt ligesom et skiftespor til et tog.
Pæren mister forbindelsen og slukkes.

Med andre ord: når der trykkes på knappen, slukker pæren.

Nyttig viden eller ligegyldig info

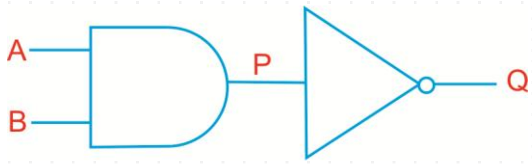
Når man er stifter og bestyrelsesformand for af et af verdens største it-firmaer (Microsoft), er det lidt fedt at hedde noget med gates - Bill Gates.

2.4 Kombinationer af Gates

Ved at kombinere gates, kan funktionaliteten af kredskøbet udbygges, og simple controllere kan bygges. Det er nemt at lave en ”fyld badekar controller” der tjekker om der er vand nok i badekarret, og hvis der er, give signal om at lukke vandhanen.
Det er også muligt at lave en kompressor controller. Man kan lave en tæller (f.eks. til display af ventenummeret i Bilkas informations kø), eller man kan lave en vaskemaskine controller.

2.4.1 Opbygning af sandhedstabeller

Her er to gatekombineret. En AND-gate er sat samen med en Inverter.
Udgangen af AND-gaten (kaldet P) er brugt som indgang på Inverteren.



Skal der laves en sandhedstabel, begynder man fra input (A, B) og bevæger sig mod output (Q). Fra venstre mod højre.
Efter A og B er det første vi møder udgangen P. Sandhedstabellen for AND-gaten skrives op med P som resultat.
Det fungerer lidt som en mellem regning:

B	A	P	Q
0	0	0	
0	1	0	
1	0	0	
1	1	1	

Sandhedstabellen bygges op med én gate ad gangen

Bagefter tilføjes en kolonne Q, og nu brugers P kolonnen som input til Inverteren.

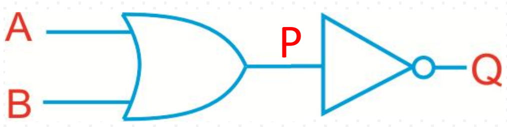
B	A	P	Q
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Q udfyldes med P som input.

Kolonne Q er med andre ord blot det modsatte af P. Set udefra er det en sandhedstabel med A og B som input og Q som output.

Ovenstående sandhedstabel er meget anvendt. Den gælder en gate, der hedder NAND gate (navnet er sammensat af "not" og "and")

Summeøvelse - snak med sidemanden
Lav en sandhedstabel for nedenstående gate-kredsløb



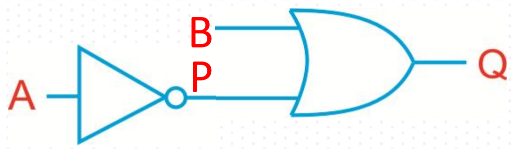
Kredsløbet betegnes også NOR gate.

B	A	P	Q
0	0		
0	1		
1	0		
1	1		

Først udfyldes kolonne P. Når det er gjort kan kolonne Q udfyldes

Kredsløbet betegnes også NOR gate.

Summeøvelse - snak med sidemanden
Lav en sandhedstabel for nedenstående gate-kredsløb



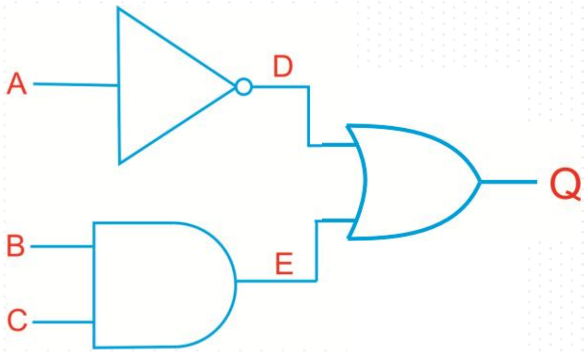
B	P	B	Q
0		0	
0		1	
1		0	
1		1	

Først udfyldes udgangen af Inverteren, P

Når P er udfyldt, bruges P og B som input til OR-gaten og Q nedskrives.

Summeøvelse - snak med sidemanden

Lav en sandhedstabel for nedenstående gate-kredsløb



C	B	E	A	D	Q
0	0		0		
0	0		1		
0	1		0		
0	1		1		
1	0		0		
1	0		1		
1	1		0		
1	1		1		

Opgaven løses ved at udfylde en kolonne ad gangen

Begynd med at udfylde E og D.

Når det er gjort udfyldes Z på baggrund af E og D

Kompressorens virkemåde

En kompressor er et apparat der fungerer ved, at den tænder en luftpumpe, der fylder luft i en tank.
Når trykket i tanken er 6 bar eller mere (input A = 1), slukkes luftpumpen.
Når trykket i tanken kommer under 6 bar (input A = 0) tændes luftpumpen.
Kompressoren virker kun, når der er tændt (B = 1) på 230 volt kontakten på væggen.
En sandhedstabel for luftpumpens funktionalitet vil se sådan ud:

B (230 V)	A (Tryk)	Q (Luftpumpe)
0	0	0
0	1	0
1	0	1
1	1	0

Sandhedstabellen passer ikke med nogen standard gates

Summeøvelse - snak med sidemanden
Fyld badekarret. Badekaret (A) kan være fyldt (1) eller ikke fyldt (0).
Når badekaret er fyldt, skal vandhanen (Q) lukkes (0).
Er badekaret ikke fyldt, skal vandhanen (Q) være åben (1)
Lav en sandhedstabel for systemet

A (badekar)	Q (vandhane)
0	
1	

Tegnes systemet med et rutediagram fylder det en hel side. Anvendes pseudokode fylder det seks linjer. Med Boolesk algebra kan det skrives på to linjer.

Summeøvelse - snak med sidemanden
Motoren (Q) på havetraktor kan kun starte, når der er benzin (B) i tanken og nøglekontakten (A) er tændt.

Opbyg en sandhedstabel for systemet:

B (Benzin)	A (nøgle)	Q (Motor)
0	0	
0	1	
1	0	
1	1	

Sandhedstabel for en motoren på en havetraktor

Summeøvelse - snak med sidemanden
En køler i et øl-anker skal holde temperaturen på +12°C. Når temperaturen i fadølsanlægget er over +12°C (A=1), tændes køleren (Q).
Når temperaturen er under +12°C, slukkes køleren.
Køleren virker kun, når der er tændt (B=1) på 230 Volt kontakten på væggen, og når der er CO_2 i kuldsyreflasken (C=1).

Opbyg en sandhedstabel for systemet

C (CO_2)	B (230 V)	A (Temp)	Q (Køleren)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Summeøvelse - snak med sidemanden

El-radiatorens virkemåde. Hvis hovedafbryderen i el-skabet er tændt, så fungerer en el-radiator ved, at den tænder automatisk, når temperaturen i stuen kommer under 20˚C. Når temperaturen kommer over 20˚C igen, slukkes radiatoren. Hvis hovedafbryderen i el-skabet er slukket, er el-radiatoren også slukket, uanset hvad temperaturen i rummet er.

a) Bestem hvad der er A og B og Q.

b) Fremstil en sandhedstabel for systemet.

B	A	Q
0	0	
0	1	
1	0	
1	1	

Når A og B er fastlagt, kan sandhedstabellen udfyldes

Summeøvelse - snak med sidemanden

Lav en sandhedstabel for et køleskab, der er tilsluttet en stikkontakt i væg-gen.

Når temperaturen i et køleskab er over 5°C skal kompressoren tændes - hvis altså stikkontakten er tændt. Når temperaturen er under 5°C, skal kompressoren slukkes.

a) Bestem hvad der er A og B og Q.

b) Fremstil en sandhedstabel for systemet.

B	A	Q
0	0	
0	1	
1	0	
1	1	

Sandhedstabel for et køleskab

2.5 Booleske operatorer

Forskellige gates opfylder forskellige booleske regneregler.
"•" anvendes når udtryk AND'es sammen.
"+" anvendes når udtryk OR'es sammen.

$1 \bullet 0 = 0$ (det er en AND gate) $1 + 0 = 1$ (det er en OR gate)

$1 \bullet 1 = 1$ $1 + 1 = 1$ (ikke 10, fordi det er OR)

$\overline{0} = 1$ (stegen over 0 læses som "0 negeret") $\overline{1} = 0$

"0 negeret" kan også siges som "ikke 0".
"1 negeret" siges som "ikke 1."

Vær opmærksom på, at nu anvendes "•" som AND (og ikke som gange-tegn). Det samme gælder plus-tegnet. I Boolesk algebra betyder 1 + 1 en OR-gate - hvis den ene eller den anden er "1", så skal udgangen være "1".

Summeøvelse - snak med sidemanden

Udregn følgende Booleske udtryk

a) $0 \bullet 1 =$

b) $1 \bullet 1 =$

c) $1 + 1 =$

d) $0 + 1 =$

e) $1 \bullet \overline{0} =$

f) $(1 \bullet 0) + (1 \bullet 1) =$

g) $\overline{(1 \bullet 0)} + (1 \bullet 1) =$

h) $(0 \bullet 1) + (1 \bullet 1) + (0 \bullet 0) =$

i) $(1 + 0) \bullet 0 + 0 + (1 \bullet \overline{1} \bullet 1) =$

j) $1 \bullet (1 + \overline{0}) + 1 + \overline{(1 + 1)} =$

Logiske operatorer fra hverdagen
Skal badekar-eksemplet skrives med logiske operatorer, kommer det til at se således ud:

$$\overline{\text{Vandhane\_åben}} \quad = \quad \text{Badekar\_fyldt}$$

Vores sandhedstabel så sådan her ud:

A (badekar)	Q (vandhane)
0	1
1	0

Sandhedstabel for vandhane og badekar

Eksemplet med kompressoren havde følgende sandhedstabel

B (230 V)	A (Tryk)	Q (Luftpumpe)
0	0	0
0	1	0
1	0	1
1	1	0

Sandhedstabel for en hobby kompressor

Luftpumpen er tændt når trykket er lavt, og der er 230 V.

Det giver følgende booleske udtryk: $Q = \overline{A} \bullet B$

Summeøvelse - snak med sidemanden
a) Find sandhedstabellen for en OR-gate frem og opstil et boolesk udtryk for den.

b) Find sandhedstabellen for en AND-gate frem og opstil et boolesk udtryk for den.

Summeøvelse - snak med sidemanden
Vaskemaskine. Lågen i en vaskemaskine skal blokeres (Q = 0), hvis der er vand (A) i vaskemaskinen, og man prøver at åbne lågen. Kredsløbet skal også blokere lågen, hvis tromlen (B) kører.

a) Lav en sandhedstabel for systemet

b) Lav et lille gatekredsløb der opfører sig som beskrevet.

c) Beskriv kredsløbet med et boolesk udtryk.

Summeøvelse - snak med sidemanden
Bevægelsessensor. Lyset i en carport skal tændes, hvis der er bevægelse foran en bevægelsessensor. Kredsløbet skal kun virke, når der er mørkt (måles med en lyssensor).

a) Lav en sandhedstabel for systemet

b) Lav et lille gatekredsløb, der opfører sig som beskrevet.

c) Beskriv kredsløbet med et boolesk udtryk.

Summeøvelse - snak med sidemanden

Brug boolesk regneregler på følgende udtryk

Husk at "+" angiver OR (ikke addition), og at "•" angiver AND (ikke gange).

Eksempel 0001 | 0110 + 0010 | 0100 . De to tal skal OR'es sammen

	0	0	0	1	0	1	1	0
+	0	0	1	0	0	1	0	0
=	0	0	1	1	0	1	1	0

a) 0000 | 0110 + 0010 | 1000 (sæt selv tallene ind i skemaet)

+								
=								

b) 0110 | 1100 + 1010 | 1001 (sæt selv tallene ind i skemaet)

+								
=								

c) 0001 | 0110 + 1010 | 0100

+								
=								

d) 0110 | 0110 + 1010 | 1011

+								
=								

e) 0001 | 0100 + 0010 | 1010

+								
=								

f) 1100 | 0110 + 1010 | 0000

+								
=								

Øvelse - sidder du fast så snak med sidemanden

Et netstik til en stuelampe stikkes i en stikkontakt. Når der tændes og sluk-kes på stikkontakten, tændes og slukkes lampen. MEN på ledningen er der monteret en afbryder. Når ledningsafbryderen er tændt, fungerer stikkontakt-kontakten normalt. Hvis ledningsafbryderen er slukket, kan lampen ikke tændes.

a) Opstil en sandhedstabel for systemet

b) Konstruer et lille gatekredsløb, der opfører sig på tilsvarende måde.

c) Beskriv kredsløbet med booleske udtryk.

Øvelse - sidder du fast så snak med sidemanden

Konstruer et lille gate kredsløb, der giver alarm, hvis lygterne i en gammel bil er tændt, selvom der ikke er tænding på (altså beep-beep, hvis man glemmer at slukke lyset, når man har kørt en tur).

Kredsløbet skal også give alarm, når der er tænding på, men med samtidig slukkede lygter (altså også en beep-beep, når der kørers en tur med sluk-kede lygter). Beskriv kredsløbet med booleske udtryk.

a) Opstil en sandhedstabel for systemet.

b) Beskriv kredsløbet med booleske udtryk.

c) Kredsløbets populære navn er *ulighedsdetektor*. Prøv at Google "basic logic gates" og klik på topmenu punktet "Billeder".
Find det symbol, der passer med den fundne sandhedstabel.

Nyttig viden eller ligegyldig info

Der ligger et godt SRP projekt i at kombinere argumentationsteknik (Toulmins argumentations-model) med Boolesk algebra. Det er Engelsk A (eller Dansk A) og matematik B eller A (boolesk algebra kan bruges som supplerende stof til matematik).
Tag f.eks. Romneys tale til det 12. konvent i 2012. Der er meget patos, men også lidt logos, logik (han taler om præsident Obama):
- This president can ask us to be patient.
- This president can tell us it was someone else's fault.
- This president can tell us that the next four years he'll get it right.
- But this president cannot tell us that you are better off today than when he took office.

Finder man Obamas tale fra samme år, kan der laves en god sammenligning.

2.6 Booleske regneregler

Rækkefølgen: Når man regner med booleske udtryk løser man parenteserne først (ligesom vi plejer - lidt som en mellemregning).

$(1 \cdot 0) + (1 \cdot 1) = 0 + 1$ efterfølgende løser man udtrykket $0 + 1 = 1$

AND (•) "binder" mere end OR (+)
Vi begynder med at udregne AND (•) først, og bagefter udregner vi OR (+).

Det kender vi fra den normale matematik, hvor vi udregner gange og divi-sion først, inden tallene lægges sammen.

I ti-talsystemet udregnes:

2•10+4•3 normalt ved at først at sige: 2 • 10 = 20 og 4 • 3 = 12.

Når det er gjort, lægges de to tal sammen. 20 + 12 = 32.

Så 2•10+4•3 bliver 32.

I boolesk algebra skal det se således ud:

1 • 0 + 1 • 1 udregnes ved først at AND’e 1 • 0 = 0 og 1 • 1 = 1.

Når det er gjort, OR'es de to tal sammen. 0 + 1 = 1.

Med andre ord er: 1 • 0 + 1 • 1 = 1.

Øvelse - sidder du fast så snak med sidemanden
Vis at følgende udtryk giver ”1”:

a) $0 \bullet 1 + 1 \bullet 1 =$

b) $1 \bullet 1 + 0 =$

c) $1 \bullet 1 + 0 \bullet 1 + \overline{1} \bullet 0 + 1 + 0 =$

d) $(1 \bullet 0) + 0 + (0 \bullet \overline{1}) + 1 + \overline{(1 \bullet 1)} =$

Inverteringsbjælken
Inverteringsbjælken fungerer som en parentes. Det bjælken dækker, skal inverteres (det, der står i ly af bjælken, hvis det regner, skal vendes om).

$\overline{0}$ bliver til 1 og $\overline{1}$ bliver til 0

$\overline{(1 + 1)}$ bliver til $\overline{(1)}$ fordi vi udregner parentesen først, og $\overline{1}$ bliver til 0.

Nyt eksempel: $1 + \overline{(1 \bullet 1)} + 0$.

Først løses parentesen. Det er $\overline{1 \bullet 1}$ og det bliver $\overline{1}$.

$\overline{1}$ er 0. Så nu står der : 1 + 0 + 0

Nu kan udregningen afsluttes: 1 + 0 + 0 = 1

De Morgans regel - ophævelse af Inverteringsbjælke
I ti-talsystemet kan en negativ parentes fjernes ved at ændre fortegnet.

y = -(x - 3) hvilket er det samme som y = - x +3

En Inverteringsbjælke, der dækker to pladser, kan fjernes ved at den "falder ned". De fortegn, den dækker (de fortegn stumperne fra bjælken rammer, når den falder ned), skal ændres.

$\overline{1 + 1}$ er det samme som $\overline{1} \bullet \overline{1}$ Bjælken er delt i to og fortegnet under den, er ændret

$\overline{1 \bullet 1}$ er det samme som $\overline{1} + \overline{1}$

Eksempel: $1 + \overline{1 \bullet 1} + 0 = 1 + \overline{1} + \overline{1} + 0$

Det er også i orden at udregne det der står under bjælken

Eksempel: $1 + \overline{1 \bullet 1} + 0 = 1 + \overline{1} + 0$

To bjælker går ud med hinanden

$\overline{\overline{0}} + 1 = 0 + 1$

Ved en lang og en kort bjælke oven på hinanden, lader man først den længste bjælke falde ned.

$\overline{\overline{1} + 1} + 0$ først lader man den største bjælke falde ned: $\overline{\overline{1}} \bullet \overline{1} + 0$

De to første bjælker går ud med hinanden, og det bliver: $1 \bullet \overline{1} + 0$

Nu kan bjælken over 1 fjernes: $1 \bullet 0 + 0$

Til sidst kommer udregning: 1 • 0 + 0 bliver til 0 + 0 = 0

Øvelse - sidder du fast så snak med sidemanden

Vis at følgende udtryk kan reduceres ned til 0

a) $\overline{1 + 0} =$

b) $(0 + \overline{1}) + \overline{\overline{0}} + \overline{1 \cdot 1} =$

c) $\overline{(1 + 0)} \cdot 0 + \overline{1} + \overline{(1 \cdot 1 \cdot 1)} + 0 =$

d) $\overline{1 \cdot \overline{0} \cdot 1} + 0 + (\overline{1} + 0) =$

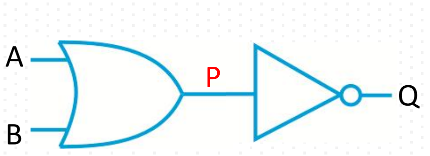
Bogstaverne fra sandhedstabellerne indsættes

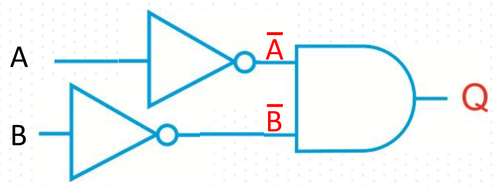
I boolesk algebra anvendes A, B, C osv. for de forskellige indgange når man tegner gates. Hver indgang kan være 1 eller 0.

Skrives regnereglerne op med bogstaver (ligesom man normalt gør med formler), kommer det til at se således ud:

$\overline{1 + 0} = \overline{1} \bullet \overline{0}$ med bogstaver bliver det til $\overline{A + B} = \overline{A} \bullet \overline{B}$

Herefter kan udtrykket tegnes med gates og en sandhedstabel kan laves:





B	A	P	Q
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

B	A	$\overline{B}$	$\overline{A}$	Q
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Det bliver til $\overline{A + B}$ $\overline{A} \bullet \overline{B}$

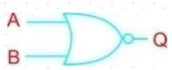
Når de to sandhedstabeller sammenlignes (de to Q-rækker er ens), betyder det, at kredsløbene er ens.

En bjælke, der dækker et helt udtryk $\overline{A+B}$, er en inverter på udgangen.

En bjælke, der kun dækker et bogstav $\overline{A}$, er en inverter på en indgang.

Vigtig viden eller ligegyldig info

En OR-Gate med en inverter på udgangen, kaldes en NOR-gate



Øvelse - sidder du fast så snak med sidemanden

a) Tegn de to Boelske udtryk med gates

$\overline{A \bullet B}$ og $\overline{A} + \overline{B}$

b) Lav de tilhørende sandhedstabeller

B	A	P	Q
0	0		
0	1		
1	0		
1	1		

B	A	$\overline{B}$	$\overline{A}$	Q
0	0			
0	1			
1	0			
1	1			

Øvelse - sidder du fast så snak med sidemanden

En anden boolesk regneregel er: $A + A \bullet B = A$

a) Tegn udtrykket "A + A•B" med gates

b) Lav en sandhedstabel for udtrykket

B	A	P	Q
0	0		
0	1		
1	0		
1	1		

Øvelse - sidder du fast så snak med sidemanden

a) Opskriv sandhedstabellen for følgende udtryk: $Q = \overline{C} \bullet (B + A)$

C	B	A	$\overline{C}$	(B + A)	Q
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Øvelse - sidder du fast så snak med sidemanden

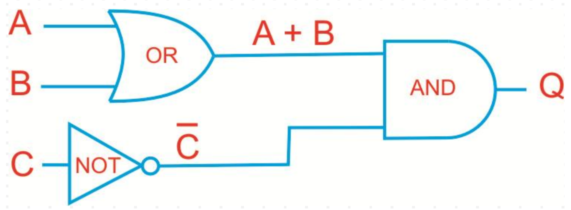
a) Opskriv sandhedstabellen for følgende udtryk: $Q = B + (C \bullet \overline{A})$

C	B	A	$\overline{A}$	$(C \bullet \overline{A})$	Q
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

b) Tegn kredsløbet med gates

Øvelse - sidder du fast så snak med sidemanden

Herunder er vist et gate-kredsløb. Opstil den tilhørende sandhedstabel



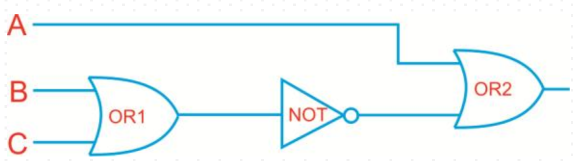
C	B	A			Q
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

b) Opstil det booleske udtryk

Øvelse - sidder du fast så snak med sidemanden

a) Opstil det booleske udtryk bag følgende kredsløb

C B A				Q
0 0 0				
0 0 1				
0 1 0				
0 1 1				
1 0 0				
1 0 1				
1 1 0				
1 1 1				



Opsummering af regneregler og skrivemåde
Først et par regneregler - er man i tvivl kan reglen altid testes med en sandhedstabel:

1) En Inverteringsbjælken kan falde ned og ændre fortegn $\overline{A + B} = \overline{A} \bullet \overline{B}$

2) Optræder en variabel alene (OR) kan de andre slettes $A + A \bullet B = A$

3) En operator OR'ed med sig selv, giver det oprindelige $A + A = A$

4) En operator AND'ed med sig selv, giver det oprindelige $A \bullet A = A$

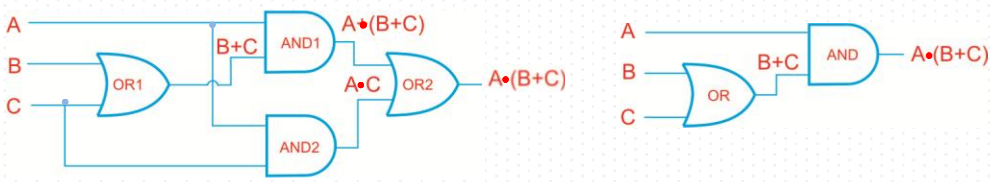
5) Man kan gange (AND) ind i en parentes: $A \bullet (B + C) = A \bullet B + A \bullet C$

6) Man kan OR en parameter ind i en parentes: $A + (B \bullet C) = (A+B) \bullet (A+C)$

Eksempel på reduktionsstykke $Q = A \bullet (B + C) + A \bullet C$
først AND'es parentesen ud (regel 5): $Q = A \bullet B + A \bullet C + A \bullet C$

$Q = A \bullet B + A \bullet C + A \bullet C$ ens operatorer samles (regel 3): $A \bullet B + A \bullet C$

$Q = A \bullet B + A \bullet C$ ”A” sættes uden for en parentes (regel 5): $Q = A \bullet (B+C)$
Tegnes det sidste eksempel, ser det således ud:



Det der kommer igennem de to kredsløb, er det samme.

Øvelse - sidder du fast så snak med sidemanden
Reducer følgende udtryk:

a) $Q = \overline{A + \overline{B}}$

b) $Q = B + \overline{A \bullet B}$

c) $Q = (A \bullet B) + C \bullet \overline{D} + A$

d) $Q = (A \bullet B) + (A \bullet \overline{C})$ (begynd med at sættes A udenfor en parentes)

e) $Q = \overline{A}B\overline{C} + AB\overline{C} + A\overline{B}C + ABC$ (ABC er det samme som A•B•C)

f) $Q = \overline{ABC} + AB\overline{C} + A\overline{B}C + ABC$

Vigtig viden eller ligegyldig info

I dette hæfte anvendes matematiske tegn som udtryk for de logiske sammenhænge, som det gøres i elektronikindustrien (Elektronik Ståbi, Teknisk Forlag, Mogens Boman & Jan Fialla, ISBN 87-571-1075-1).
I Jørgen Eberts bog (Boolesk Algebra, Systime, ISBN 87-7351-339-3) anvendes andre symboler.

$A \bullet B$	and	Konjunktion	$A \wedge B$
$\overline{A}$	invertering	Negation	$\neg A$
$A + B$	or	Disjunktion	$A \vee B$

2.7 Reduktion med Karnaughkort

Logiske reduktionsopgaver kan løses grafisk med Karnaughkort. Sandhedstabellen skrives op og indsættes i derefter en matrice. Det er ofte nemmere end at foretage udregningen.

Eksempel på reduktion af: $Q = \overline{A}B\overline{C} + AB\overline{C} + A\overline{B}C + ABC$

C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Hver celle repræsenterer Q i en linje af sandhedstabellen

Karnaughkort for		$\overline{BA}$	$\overline{B}A$	BA	$B\overline{A}$
		00	01	11	10
$\overline{C}$	0	0	0	1	1
C	1	0	1	1	0

Den markerede celle er når C=1, B=0 og A=1 dvs. $C\overline{B}A = 1$

For at reducere sammles alle ”1” der står ved siden ag hinanden. Det vil sige at ”1” grupperes i Karnaughkortet.

Karnaughkort for		$\overline{BA}$	$\overline{B}A$	BA	$B\overline{A}$
		00	01	11	10
$\overline{C}$	0	0	0	1	1
C	1	0	1	1	0

Det ses at når C = 0 (dvs. $\overline{C}$) er det lige meget hvad "A" er, i kolonne BA og $B\overline{A}$, bare B = 1.

Så første led i løsningen er, at $Q = \overline{C} \cdot B$ + ...

Næste gruppe med "1" findes:

Karnaughkort for		$\overline{B}\,\overline{A}$	$\overline{B}A$	BA	$B\overline{A}$
		00	01	11	10
$\overline{C}$	0	0	0	1	1
C	1	0	1	1	0

Det ses at når C = 1, er det lige meget hvad "B" er i kolonne $\overline{B}A$ og BA, når bare A = 1.

Det giver leddet, $C \cdot A$ (dvs. Q=1 når $C \cdot A$ - altså Q=1 når C=1 AND A=1)

Så med det andet led, bliver løsningen $Q = \overline{C} \cdot B + C \cdot A$

Nu kan der ikke laves flere sløjfer - vi er færdige. <u>Så det reducerede udtryk er $Q = \overline{C}B + CA$</u>

2.8 Karnaughkort regler

Ved brug af Karnaughkort skal man altid lave de største sløjfer først. Sløjferne skal være rektangulære eller kvadratiske. Alle ettaller skal sløjfes. Samme ettal må gerne indgå i flere sløjfer.

a) Reducer følgende udtryk med Karnaughkort: $Q = \overline{A}B\overline{C} + AB\overline{C} + \overline{A}BC + ABC$

Begynd med at indskrive sandhedstabellen i Karbaughkortet.
Det første 1-tal skal skrives i rækken for C=0. I kolonnen for B=1 og A=0 (dvs. $B\overline{A}$ - ”her”)

C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Karnaughkort for		$\overline{B}\,\overline{A}$	$\overline{B}A$	BA	$B\overline{A}$
		00	01	11	10
$\overline{C}$	0				her
C	1				

b) Reducer følgende udtryk med Karnaughkort: $Q = \bar{A}B\bar{C} + AB\bar{C} + \overline{AB}C + A\bar{B}C$

C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Karnaughkort for		$\overline{B}\overline{A}$	$\overline{B}A$	BA	$B\overline{A}$
		00	01	11	10
$\overline{C}$	0				
C	1				

c) Reducer følgende udtryk med Karnaughkort: $Q = \overline{ABC} + \overline{AB}C + ABC + \bar{A}BC$

C	B	A	Q
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Karnaughkort for		$\overline{B}\overline{A}$	$\overline{B}A$	BA	$B\overline{A}$
		00	01	11	10
$\overline{C}$	0				
C	1				

d) Reducer følgende udtryk med Karnaughkort: $Q = A\overline{BC} + AB\bar{C} + A\bar{B}C + ABC$

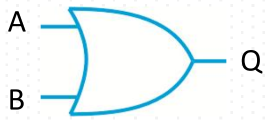
C	B	A	Q
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

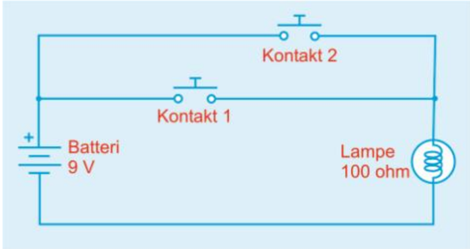
Karnaughkort for		$\overline{B}\overline{A}$	$\overline{B}A$	BA	$B\overline{A}$
		00	01	11	10
$\overline{C}$	0		1	1	
C	1		1	1	

Her er et værktøj der kan hjælpe med reduktionsstykkerne:
http://www.softpedia.com/get/Others/Home-Education/KarnaughMap.shtml

OR gate

B	A	Q
0	0	0
0	1	1
1	0	1
1	1	1

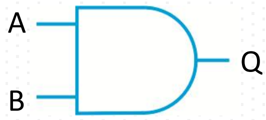


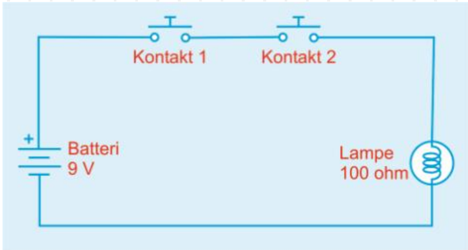


1 + 0 = 1, 1 + 1 = 1, 0 + 0 = 0

AND gate

B	A	Q
0	0	0
0	1	0
1	0	0
1	1	1

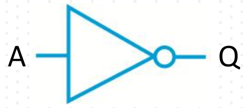


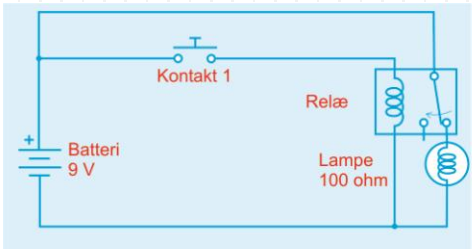


1 • 0 = 0, 1 • 1 = 1, 0 • 0 = 0

Inverter (NOT gate)

A	Q
0	1
1	0





$\overline{0}$ = 1 (stegen over 0, læses som "0 negeret") $\overline{1}$ = 0

---

1) En Inverteringsbjælken kan falde ned og ændre fortegn $\overline{A + B} = \overline{A} \bullet \overline{B}$

2) Optræder en variabel alene (OR) kan de andre slettes $A + A \bullet B = A$

3) En operator OR'ed med sig selv, giver det oprindelige $A + A = A$

4) En operator AND'ed med sig selv, giver det oprindelige $A \bullet A = A$

5) Man kan gange (AND) ind i en parentes: $A \bullet (B + C) = A \bullet B + A \bullet C$

6) Man kan OR en parameter ind i en parentes: $A + (B \bullet C) = (A+B) \bullet (A+C)$