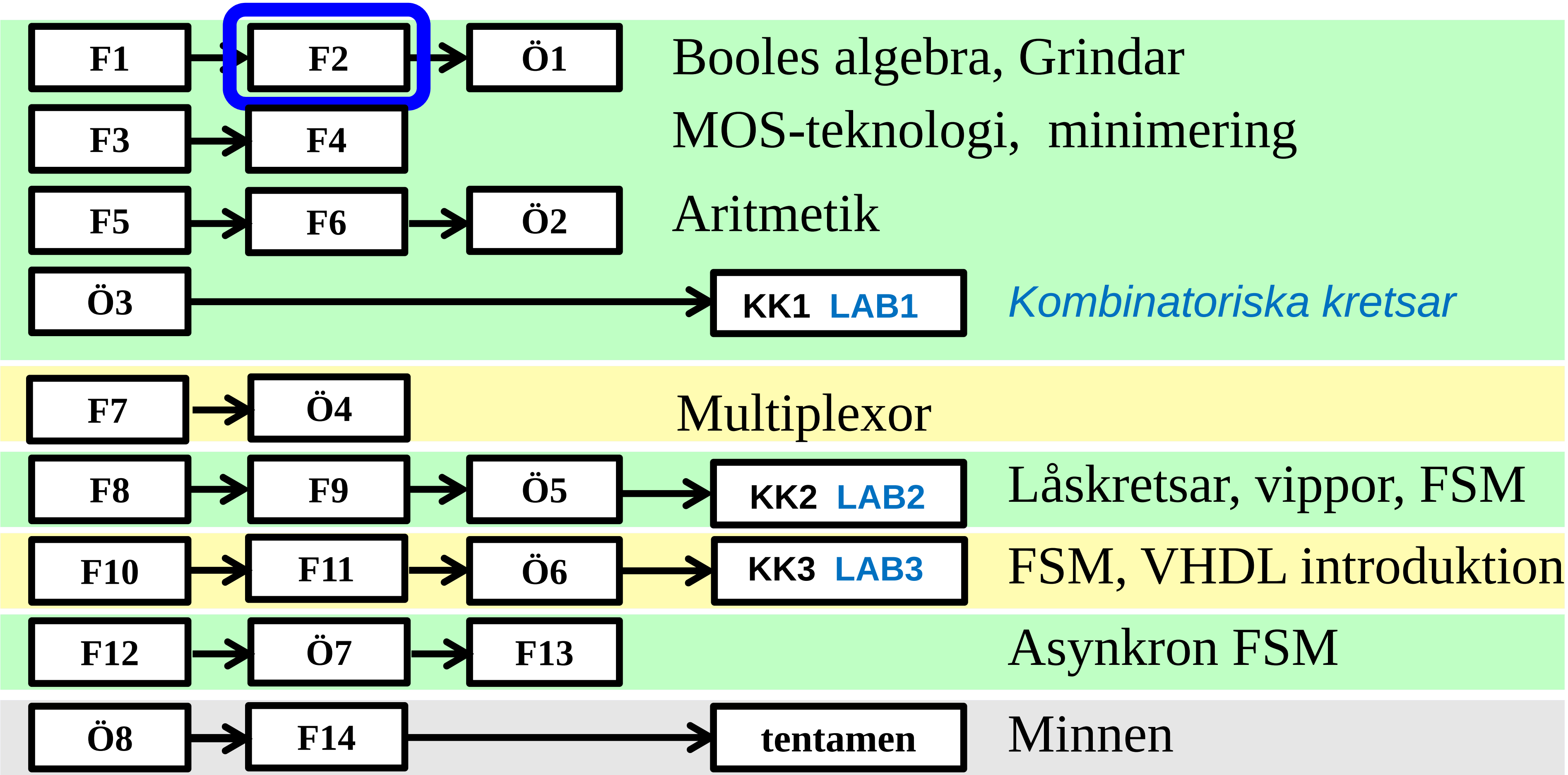
Digital Design IE1204

Baserat på föreläsningsbilder av William Sandqvist

F2 : Logiska Grindar och Kretsar, Boolesk Algebra

Carl-Mikael Zetterling
bellman@kth.se

IE1204 Digital Design



Föreläsningar och övningar bygger på varandra! Ta alltid igen det Du missat!
Läs på i förväg – delta i undervisningen – arbeta igenom materialet efteråt!

Detta har hänt …

Talsystem: Decimala, hexadecimala, oktala, binära

$$(175,5)_{10} = (AE.8)_{16} = (256.4)_8 = (10101110.1)_2$$

Switch

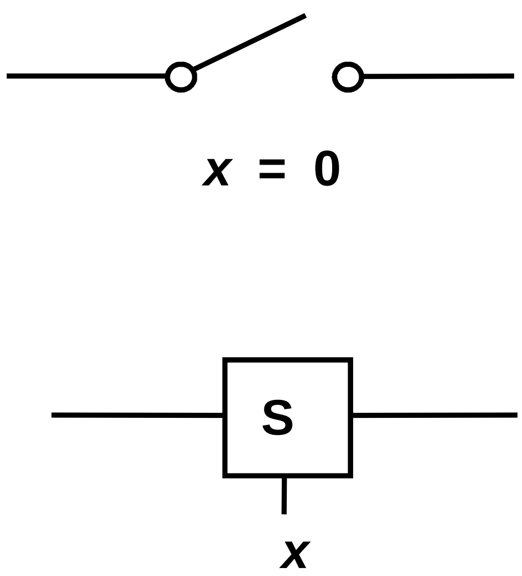
En switch har två lägen

- Sluten/Till (Closed/On)
- Öppen/Från (Open/Off)

Sluten

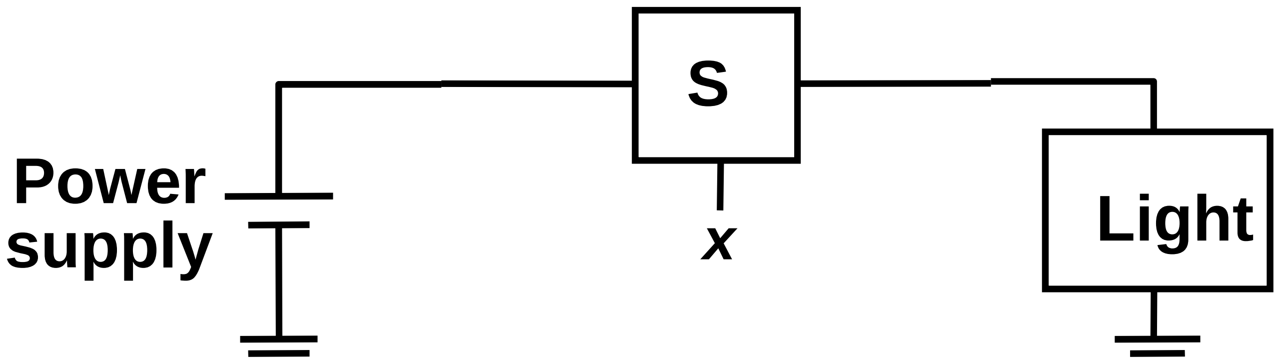
Öppen

Symbol



Implementering av logiska funktioner

Switchen kan användas för att implentera logiska funktioner



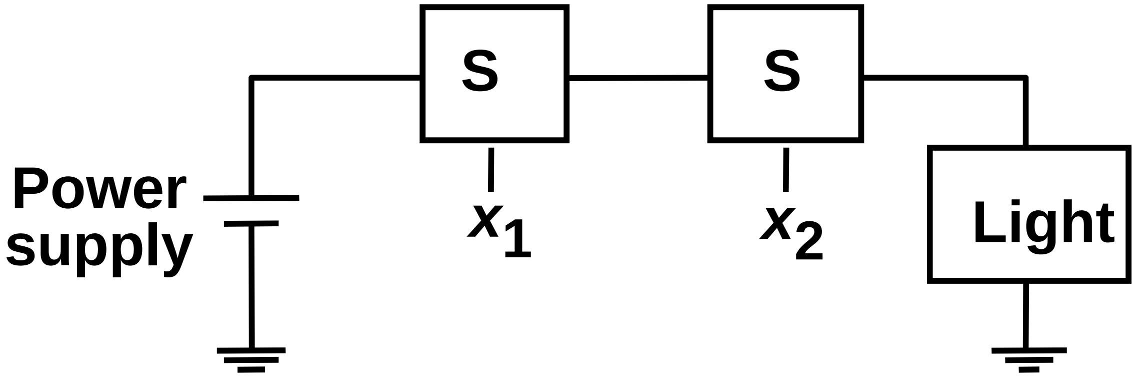
L(x) är en logisk funktion

x är en logisk variabel

$$L(x) = \begin{cases} 0 & \text{Light Off} \\ 1 & \text{Light On} \end{cases}$$

Operation **AND** (OCH)

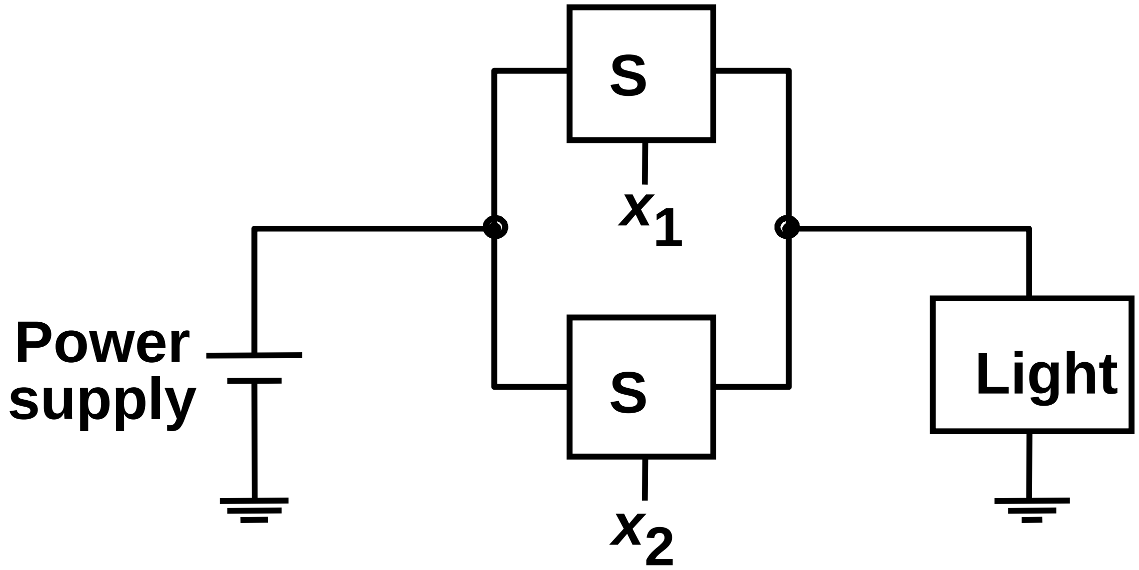
AND-operationen (•) uppnås genom switchar som kopplas i serie



$$L(x_2, x_1) = x_1 \cdot x_2$$

Operation OR (ELLER)

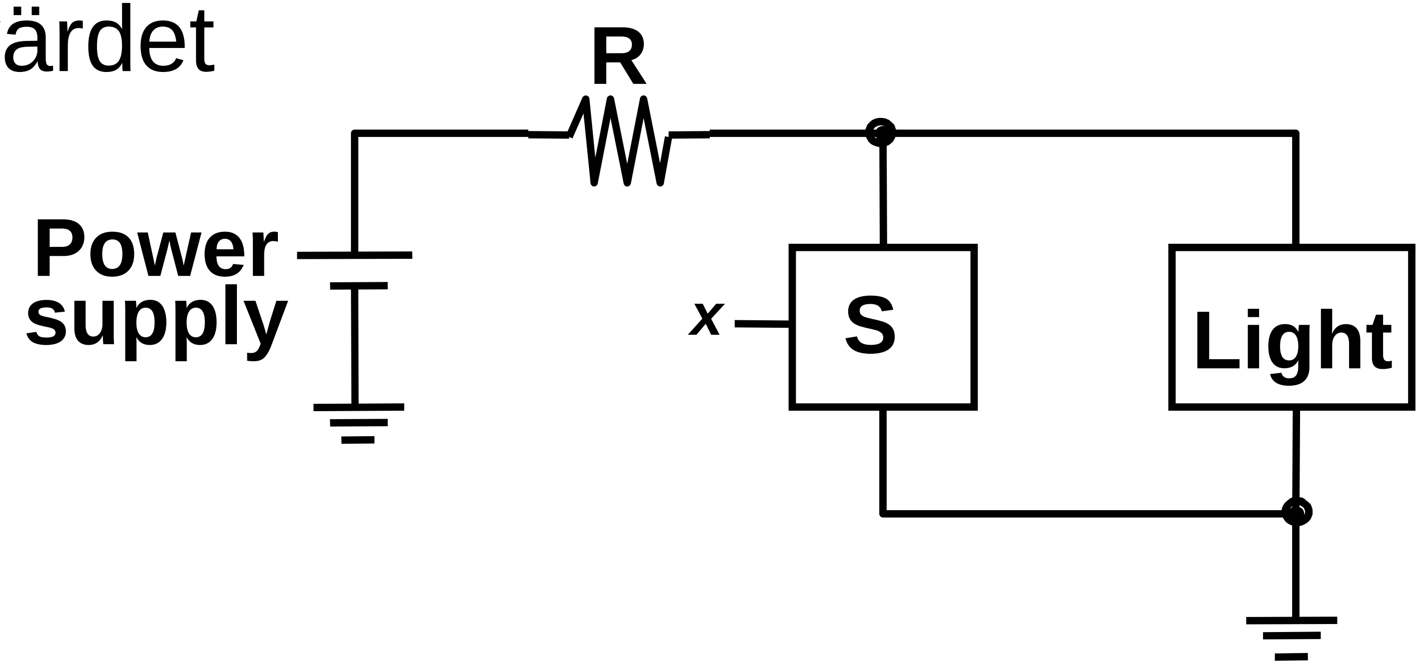
OR-operationen (+) uppnås genom switchar som kopplas parallellt



$$L(x_2, x_1) = x_1 + x_2$$

Operation NOT (ICKE)

NOT-funktionen inverterar det logiska värdet



$$L(x) = \overline{x}$$

Sanningstabell

En logisk funktion kan också beskrivas genom en *sanningstabell (truth table)*

x_1	x_2	$x_1 \cdot x_2$	$x_1 + x_2$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

AND OR

1 står för sann (true)

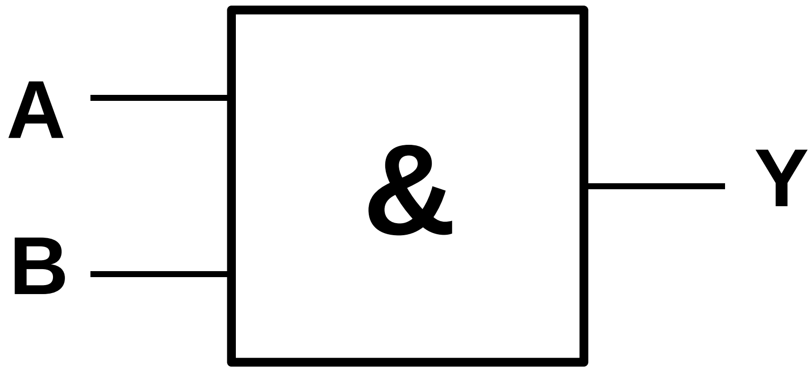
0 står för falsk (false)

Logiska grindar
AND-grinden (OCH)

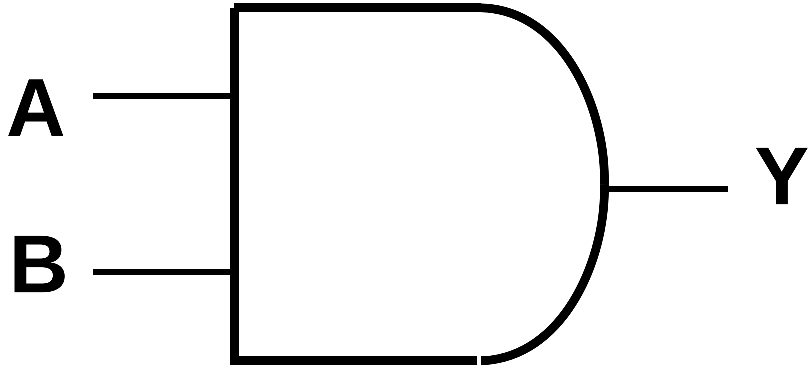
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$$Y = A \cdot B$$

IEC Symbol
(International Electrotechnical Commission)



Traditional (American) Symbol

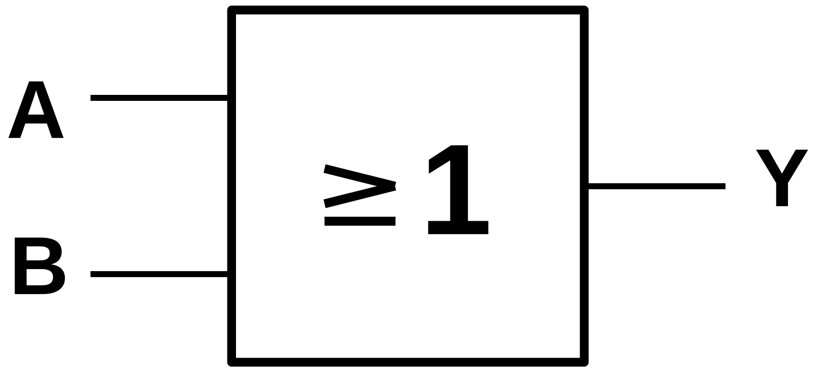


Logiska grindar
OR-grinden (ELLER)

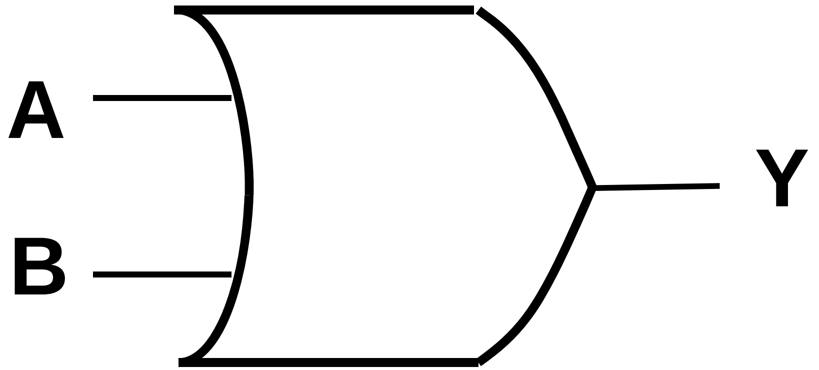
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

$$Y = A + B$$

IEC Symbol
(International Electrotechnical Commission)



Traditional (American) Symbol



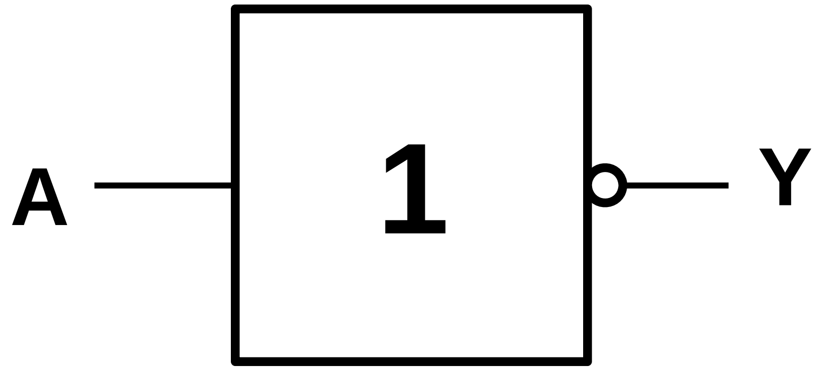
Logiska grindar
inverterare NOT (ICKE)

Inverterare (Inverter)

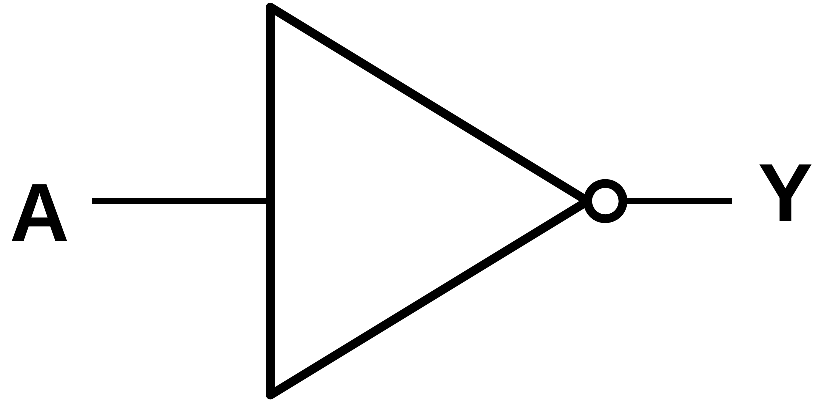
A	Y
0	1
1	0

$$Y = \overline{A}$$

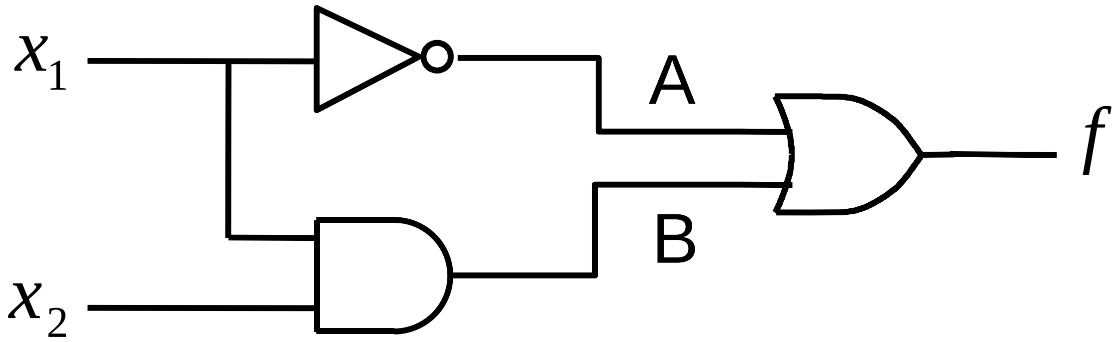
IEC Symbol
(International Electrotechnical Commission)



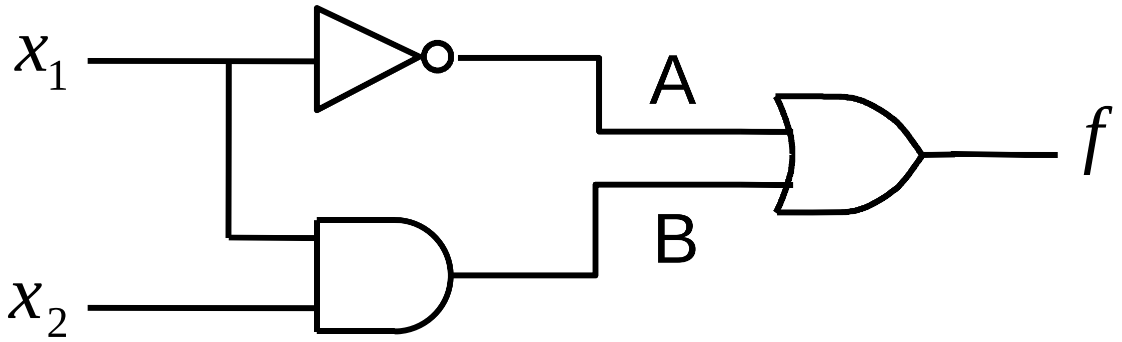
Traditional (American) Symbol

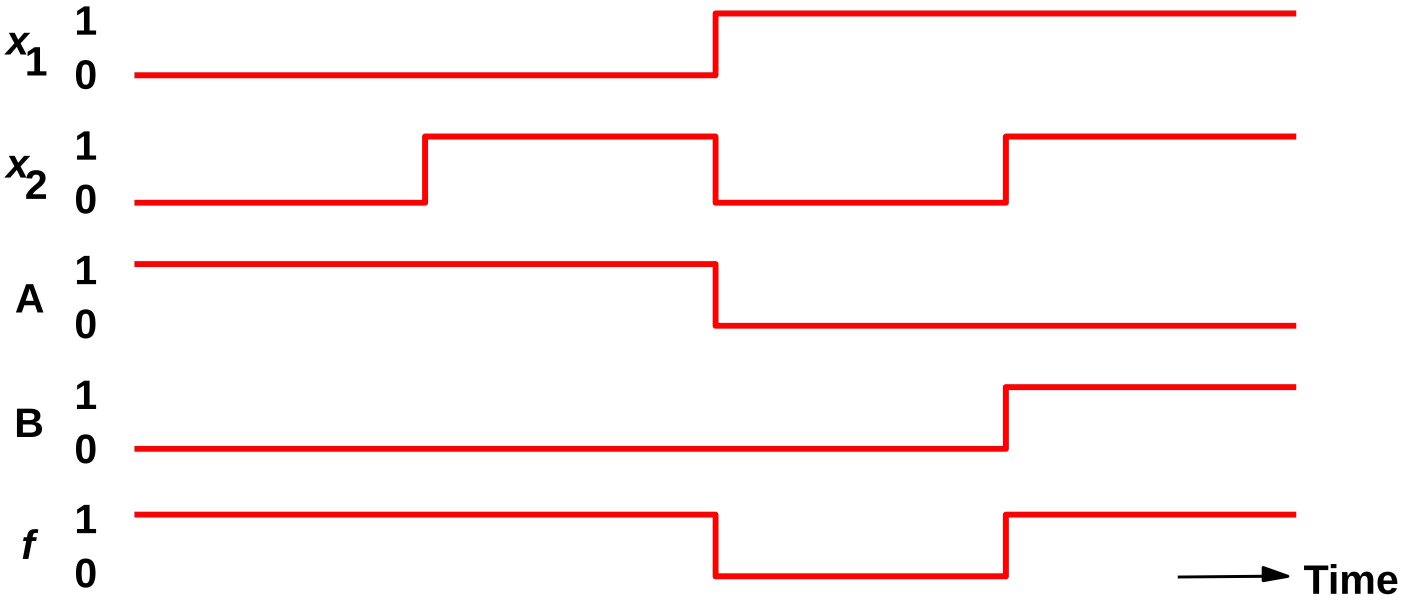


Vilken funktion har grindnätet?

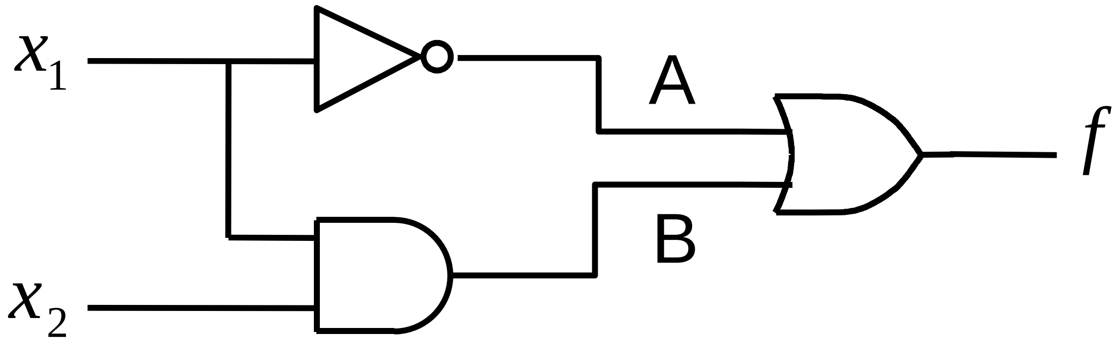


Tidsdiagram





Sanningstabell

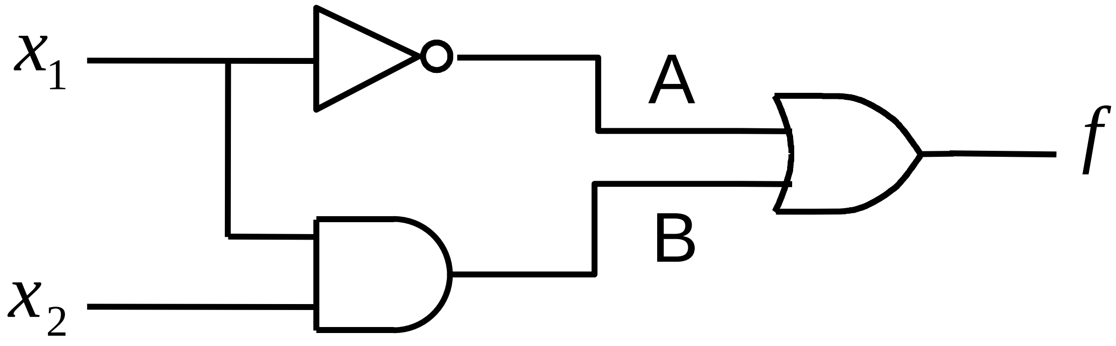


x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

A	B
1	0
1	0
0	0
0	1

Flera grindnät kan implementera samma funktion!

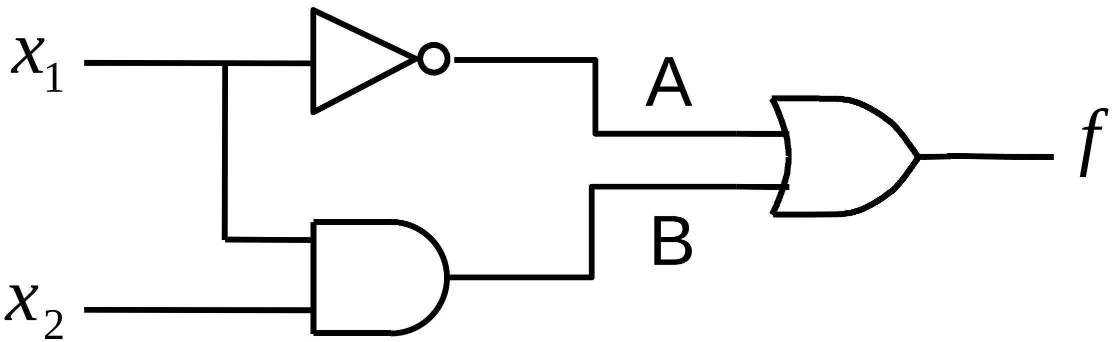
a)



$$f = \overline{x}_1 + x_1 \cdot x_2$$

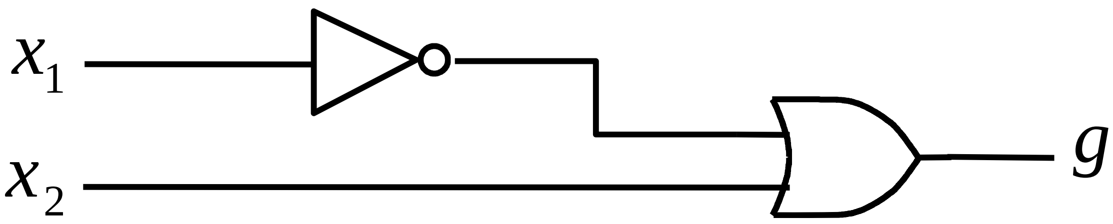
Flera grindnät kan implementera samma funktion!

a)



$$f = \overline{x}_1 + x_1 \cdot x_2$$

b)



$$g = \overline{x}_1 + x_2$$

f* = *g

x_1	x_2	f	g
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

Boolesk algebra

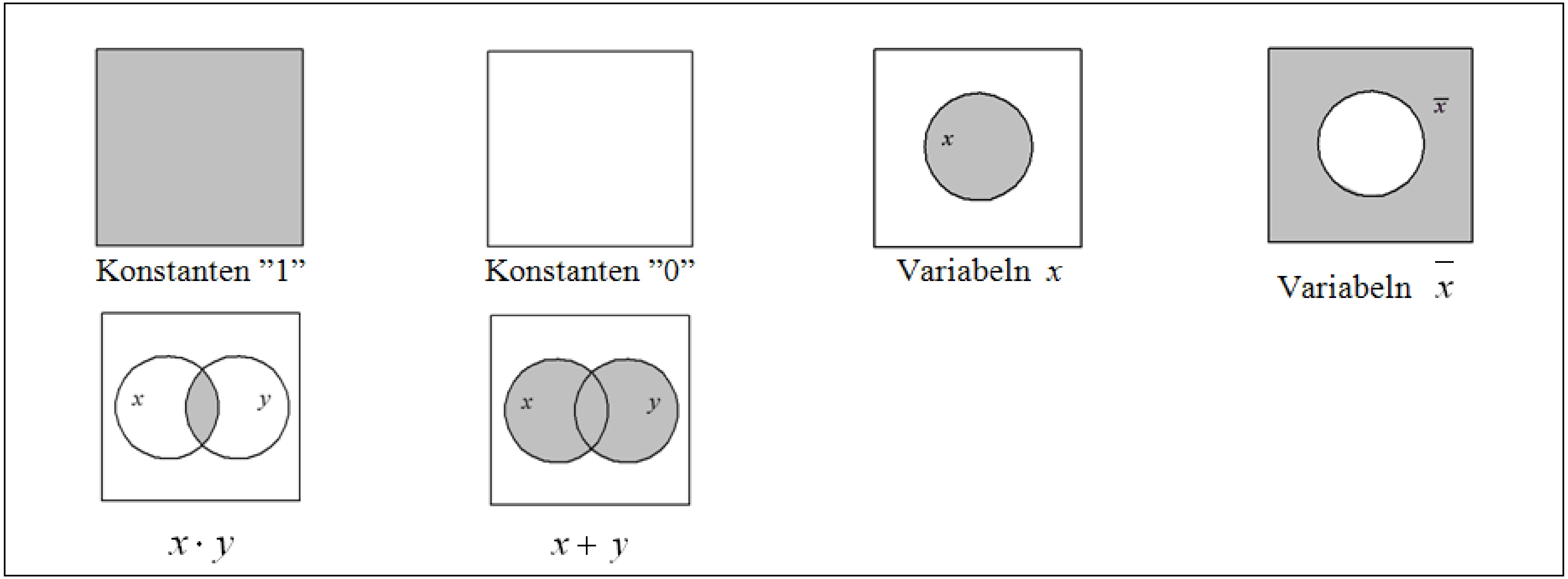
- Eftersom flera grindnät kan implementera samma funktion, så vill man hitta den ***mest kostnadseffektiva implementeringen***
- Grindnäten kan bli mycket stora
- En matematisk bas behövs så att automatiseringen av grindnätsoptimering kan genomföras med datorer

Booles algebra axiom

Axiom			
(1a)	$0 \cdot 0 = 0$	(1b)	$1 + 1 = 1$
(2a)	$1 \cdot 1 = 1$	(2b)	$0 + 0 = 0$
(3a)	$0 \cdot 1 = 1 \cdot 0 = 0$	(3b)	$1 + 0 = 0 + 1 = 1$
(4a)	If $x = 0$, then $\overline{x} = 1$	(4b)	If $x = 1$, then $\overline{x} = 0$

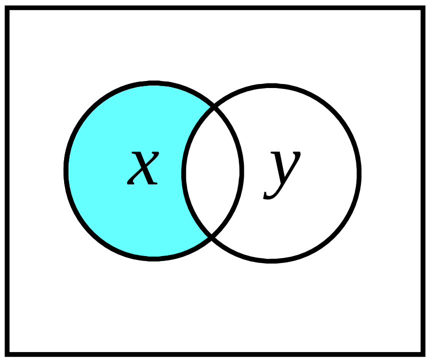
Venn-diagram

Venn-diagrammet kan användas för att illustrera logiska operationer

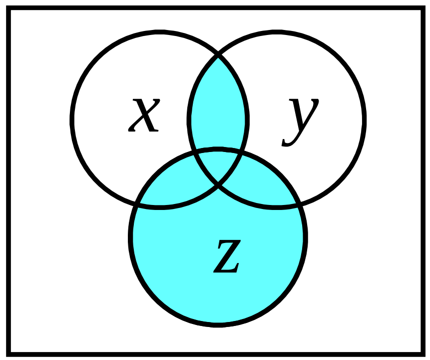


Venn-diagram

Venn-diagrammet kan användas för att illustrera logiska operationer



$x \cdot \overline{y}$



$x \cdot y + z$

Boolesk algebra med Venn-diagram

$1+A=1$

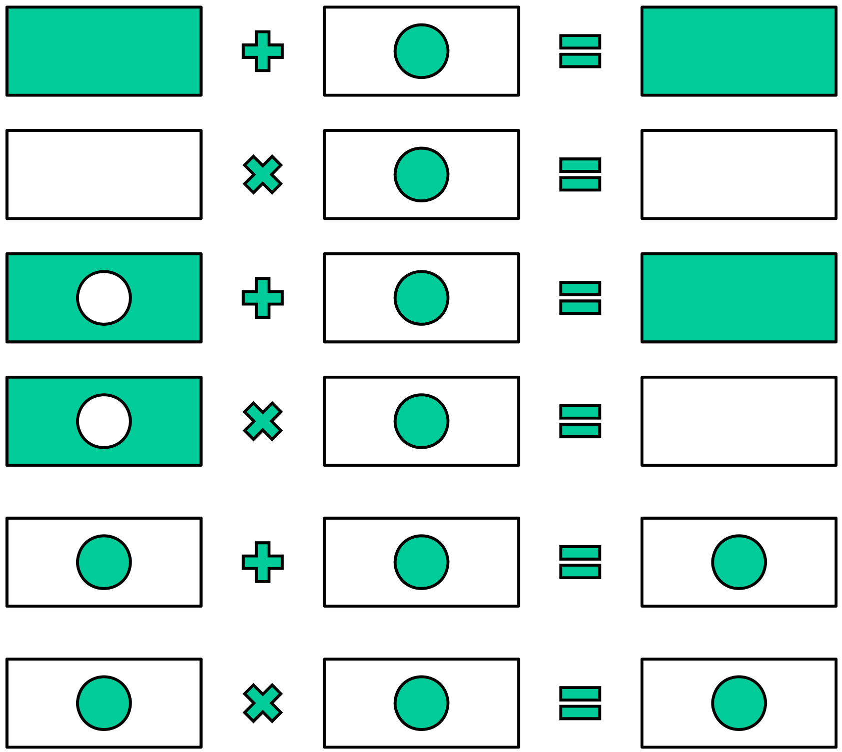
$0A=0$

$A'+A=1$

$AA'=0$

$A+A=A$

$AA=A$



Booles algebra
enkla räkneregler

Med axiomen som bas kan man formulera nya lagar (teorem)

Räknelagar			
(5a)	$x \cdot 0 = 0$	(5b)	$\mathrm{x} + 1 = 1$
(6a)	$x \cdot 1 = x$	(6b)	$\mathrm{x} + 0 = \mathrm{x}$
(7a)	$x \cdot x = x$	(7b)	$\mathrm{x} + \mathrm{x} = \mathrm{x}$
(8a)	$x \cdot \overline{x} = 0$	(8b)	$x + \overline{x} = 1$
(9a)	$\overline{\overline{x}} = x$		

Dualitetsprincipen

Har man ett giltigt booleskt samband så får man ett annat giltigt samband om man samtidigt byter

– alla 0:or mot 1:or och alla 1:or mot 0:or

– alla AND mot OR och alla OR mot AND

x_1	x_2	$x_1 \cdot x_2$	x_1	x_2	$x_1 + x_2$
0	0	0	1	1	1
0	1	0	1	0	1
1	0	0	0	1	1
1	1	1	0	0	0

Boolesk algebra
Räknelagar med flera variabler

Räknelagar				
(10a)	$x \cdot y = y \cdot x$	(10b)	$x + y = y + x$	*kommutativ*
(11a)	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	(11b)	$x + (y + z) = (x + y) + z$	*associativ*
(12a)	$x \cdot (y + z) = x \cdot y + x \cdot z$	(12b)	$x + y \cdot z = (x + y) \cdot (x + z)$	*distributiv*
(13a)	$x + x \cdot y = x$	(13b)	$x \cdot (x + y) = x$	*absorption*
(14a)	$x \cdot y + x \cdot \overline{y} = x$	(14b)	$(x + y) \cdot (x + \overline{y}) = x$	
(15a)	$\overline{x \cdot y} = \overline{x} + \overline{y}$	(15b)	$\overline{x + y} = \overline{x} \cdot \overline{y}$	*DeMorgan*
(16a)	$x + \overline{x} \cdot y = x + y$	(16b)	$x \cdot (\overline{x} + y) = x \cdot y$	
(17a)	$x \cdot y + y \cdot z + \overline{x} \cdot z = x \cdot y + \overline{x} \cdot z$	(17b)	$(x + y) \cdot (y + z) \cdot (\overline{x} + z) = (x + y) \cdot (\overline{x} + z)$	*consensus*

Exempel

Bevisa konsensuslagen (17a)

- med Venn-diagram (se övning 1)
- med algebraisk manipulation

Bevis av en av konsensuslagarna

17 a) $x \cdot y + \bar{x} \cdot z = x \cdot y + y \cdot z + \bar{x} \cdot z$

$$x \cdot y + y \cdot z + \bar{x} \cdot z \quad \text{(höger led)}$$

$$= \quad x \cdot y \cdot (z + \bar{z}) + (x + \bar{x}) \cdot y \cdot z + \bar{x} \cdot (y + \bar{y}) \cdot z$$

$$= \quad x \cdot y \cdot z + x \cdot y \cdot \bar{z} + x \cdot y \cdot z + \bar{x} \cdot y \cdot z + \bar{x} \cdot y \cdot z + \bar{x} \cdot \bar{y} \cdot z$$

$$= \quad x \cdot y \cdot z + x \cdot y \cdot \bar{z} + \bar{x} \cdot y \cdot z + \bar{x} \cdot \bar{y} \cdot z$$

$$= \quad x \cdot y \cdot (z + \bar{z}) + \bar{x} \cdot z \cdot (y + \cdot \bar{y})$$

$$= \quad x \cdot y + \bar{x} \cdot z \quad (= \text{vänster led})$$

Notationsalternativ

Olika författare använder olika notationer!

$\overline{x}$	x', $!x$, $/x$, $\neg x$
$x \cdot y$	xy, $x \wedge y$
$x + y$	$x \vee y$

Analys och **Syntes**

Syntes

– Konstruktion av ett grindnätverk som implementerar en given logisk funktion

Analys

– Framtagandet av den logiska funktionen för ett existerande grindnätverk

Hur kan följande sanningstabell implementeras med logiska grindar?

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

(Varför denna sanningstabell?)

Grunka/grej …

Tryck på/av 1/0 **Kran** öppen/stängd 1/0

x_1	x_2	$f(x_1, x_2)$	
0	0	1	OK
0	1	1	OK
1	0	0	ej OK
1	1	1	OK

Vild gissning:
Varna för ***tryck*** **på** samtidigt som ***kran*** **stängd**.

Hur kan följande sanningstabell implementeras med logiska grindar?

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

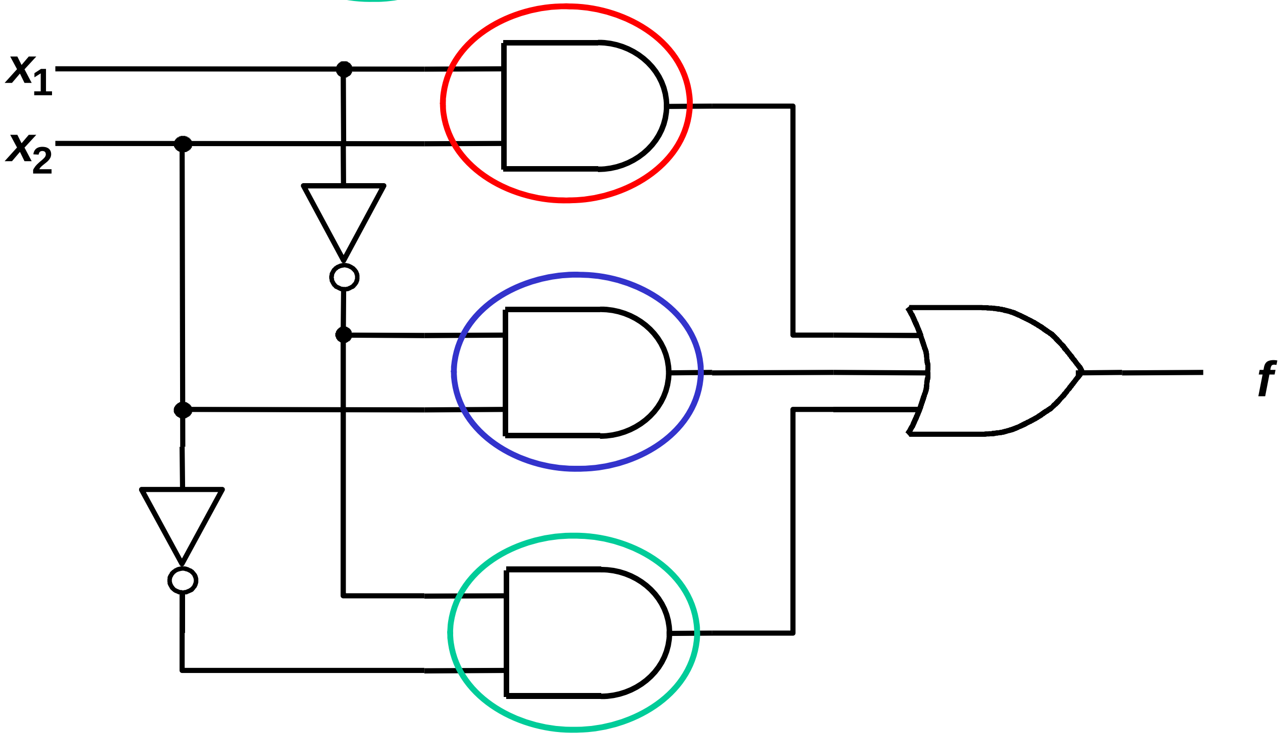
1. Ta fram den logiska funktionen.

$$f = \bar{x}_1\bar{x}_2 + \bar{x}_1 x_2 + x_1 x_2$$

Hur kan följande sanningstabell implementeras med logiska grindar?

2. Gör en direkt implementering av den logiska funktionen.

$$f = \overline{x}_1\overline{x}_2 + \overline{x}_1 x_2 + x_1 x_2$$



Hur kan följande sanningstabell implementeras med logiska grindar?

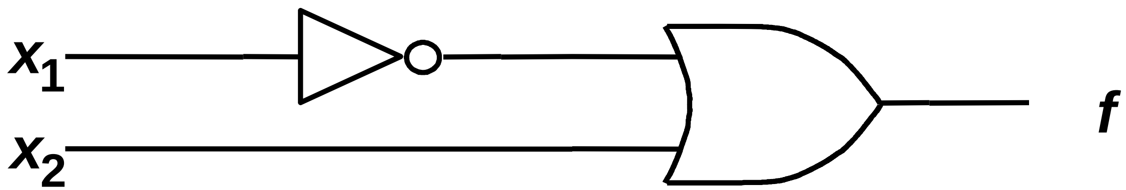
2. (Bättre) Minimera den logiska funktionen

$$
\begin{aligned}
f &= \bar{x}_1\bar{x}_2 + \bar{x}_1 x_2 + x_1 x_2 \\
&= \bar{x}_1\bar{x}_2 + \bar{x}_1 x_2 + \bar{x}_1 x_2 + x_1 x_2 && \text{Lägg till redundant term } \bar{x}_1 x_2 \text{ (7b)} \\
&= \bar{x}_1(\bar{x}_2 + x_2) + (\bar{x}_1 + x_1)x_2 && \text{Distribution (12a)} \\
&= \bar{x}_1 \cdot 1 + 1 \cdot x_2 && \text{(8b)} \\
&= \bar{x}_1 + x_2
\end{aligned}
$$

Hur kan följande sanningstabell implementeras med logiska grindar?

3. Implementera den minimerade funktionen

$$f = \overline{x}_1 + x_2$$



Mycket enklare implementering!

Diskussion: Algebraisk manipulering

- Algebraisk manipulering av logiska uttryck kan leda till effektiva implementeringar
- Men: För större nätverk kan det bli mycket svårt att identifiera möjliga optimeringar

Vi behöver en metod som fungerar för alla kombinatoriska nätverk!

Mintermer och **Maxtermer**

- En **minterm** är en ***produktterm*** för en logisk funktion där ***alla*** variabler av den logiska funktionen måste vara representerade
- En **maxterm** är en ***summaterm*** för en logisk funktion där ***alla*** variabler av den logiska funktionen måste vara representerade

Mintermer och **Maxtermer**

Row number	x_1	x_2	x_3	**= 1** Minterm	**= 0** Maxterm
0	0	0	0	$m_0 = \overline{x}_1\overline{x}_2\overline{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \overline{x}_1\overline{x}_2 x_3$	$M_1 = x_1 + x_2 + \overline{x}_3$
2	0	1	0	$m_2 = \overline{x}_1 x_2\overline{x}_3$	$M_2 = x_1 + \overline{x}_2 + x_3$
3	0	1	1	$m_3 = \overline{x}_1 x_2 x_3$	$M_3 = x_1 + \overline{x}_2 + \overline{x}_3$
4	1	0	0	$m_4 = x_1\overline{x}_2\overline{x}_3$	$M_4 = \overline{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\overline{x}_2 x_3$	$M_5 = \overline{x}_1 + x_2 + \overline{x}_3$
6	1	1	0	$m_6 = x_1 x_2\overline{x}_3$	$M_6 = \overline{x}_1 + \overline{x}_2 + x_3$
7	1	1	1	$m_7 = x_1 x_2 x_3$	$M_7 = \overline{x}_1 + \overline{x}_2 + \overline{x}_3$

Introduktion **SP** och **PS**

Följande logisk funktion
ska beskrivas med ett booleskt uttryck

Row number	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Sum of **P**roducts **SP** (SOP)

Row number	x_1	x_2	x_3	$f(x_1, x_2, x_3)$	
0	0	0	0	0	
1	0	0	1	1	m_1
2	0	1	0	0	
3	0	1	1	0	
4	1	0	0	1	m_4
5	1	0	1	1	m_5
6	1	1	0	1	m_6
7	1	1	1	0	

$$f = \overline{x}_1\overline{x}_2x_3 + x_1\overline{x}_2\overline{x}_3 + x_1\overline{x}_2x_3 + x_1x_2\overline{x}_3 = \sum m(1,4,5,6)$$

Sum - of - Products

En *summa av produkter (sum-of-products)* är en logisk funktion *f* som bildas genom att summera produkttermerna så att *f* blir 1 om *en* av produkttermerna blir 1.

- Följande förkortningar används SOP (engelska) och SP (svenska)

I *SOP-normalformen* är alla produkttermer mintermer

- Det benämns även som *disjunktiv normalform*

Products - of - Sums

Row number	x_1	x_2	x_3	$f(x_1, x_2, x_3)$	
0	0	0	0	0	M_0
1	0	0	1	1	
2	0	1	0	0	M_2
3	0	1	1	0	M_3
4	1	0	0	1	
5	1	0	1	1	
6	1	1	0	1	
7	1	1	1	0	M_7

$$f = (x_1 + x_2 + x_3) \cdot (x_1 + \overline{x}_2 + x_3) \cdot (x_1 + \overline{x}_2 + \overline{x}_3) \cdot (\overline{x}_1 + \overline{x}_2 + \overline{x}_3) = \prod M(0,2,3,7)$$

Products - of - Sums

En *produkt av summor (product-of-sums)* är en logisk funktion *f* som bildas genom en produkt av summatermerna så att *f* blir 0 om *en* av sumtermer blir 0.

- **Följande förkortningar används POS (engelska) och PS (svenska)**

I POS-*normalformen* är alla sumtermer maxtermer

- **Det benämns även som *konjunktiv normalform***

Dualitet mellan **Mintermer** och **Maxtermer** och mellan **SP** och **PS**

- Till varje **minterm** finns det en motsvarande **maxterm** $f = \overline{m_i} = M_i$

$$M_0 = \overline{m_0} = \overline{\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3} = \bar{\bar{x}}_1 + \bar{\bar{x}}_2 + \bar{\bar{x}}_3 = x_1 + x_2 + x_3$$

(mha DeMorgan 15a)

- Till varje **SP** finns det en motsvarande **PS**

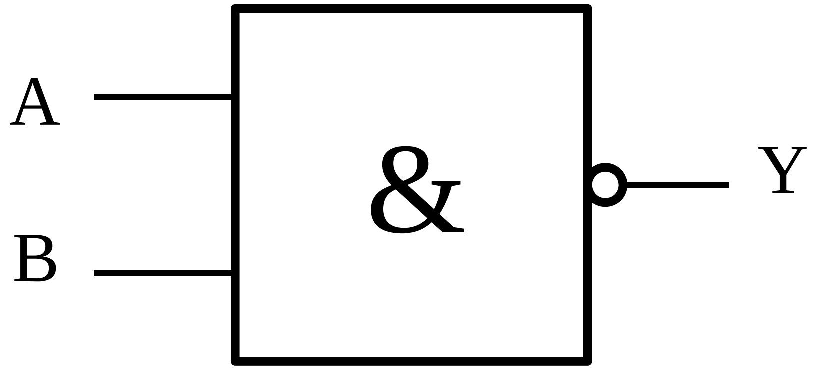
$$f = \sum m(1,4,5,6) = \prod M(0,2,3,7)$$

Logiska grindar NAND-grinden

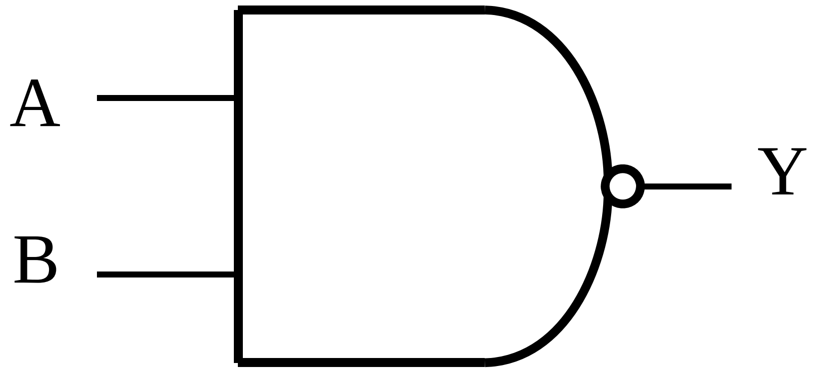
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

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

IEC Symbol
(International Electrotechnical Commission)



Traditional (American) Symbol

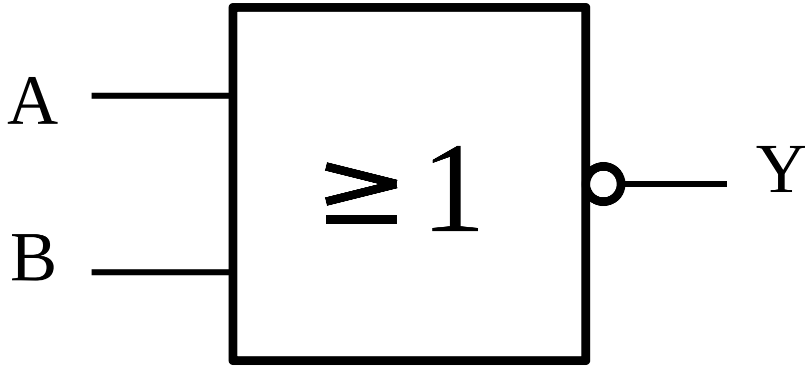


Logiska grindar NOR-grinden

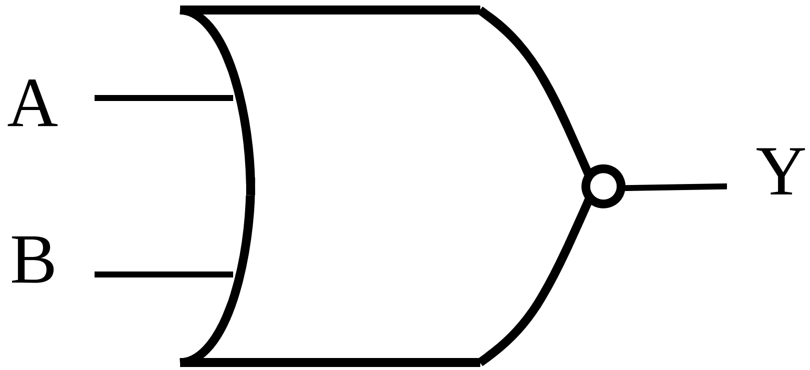
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

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

IEC Symbol
(International Electrotechnical Commission)

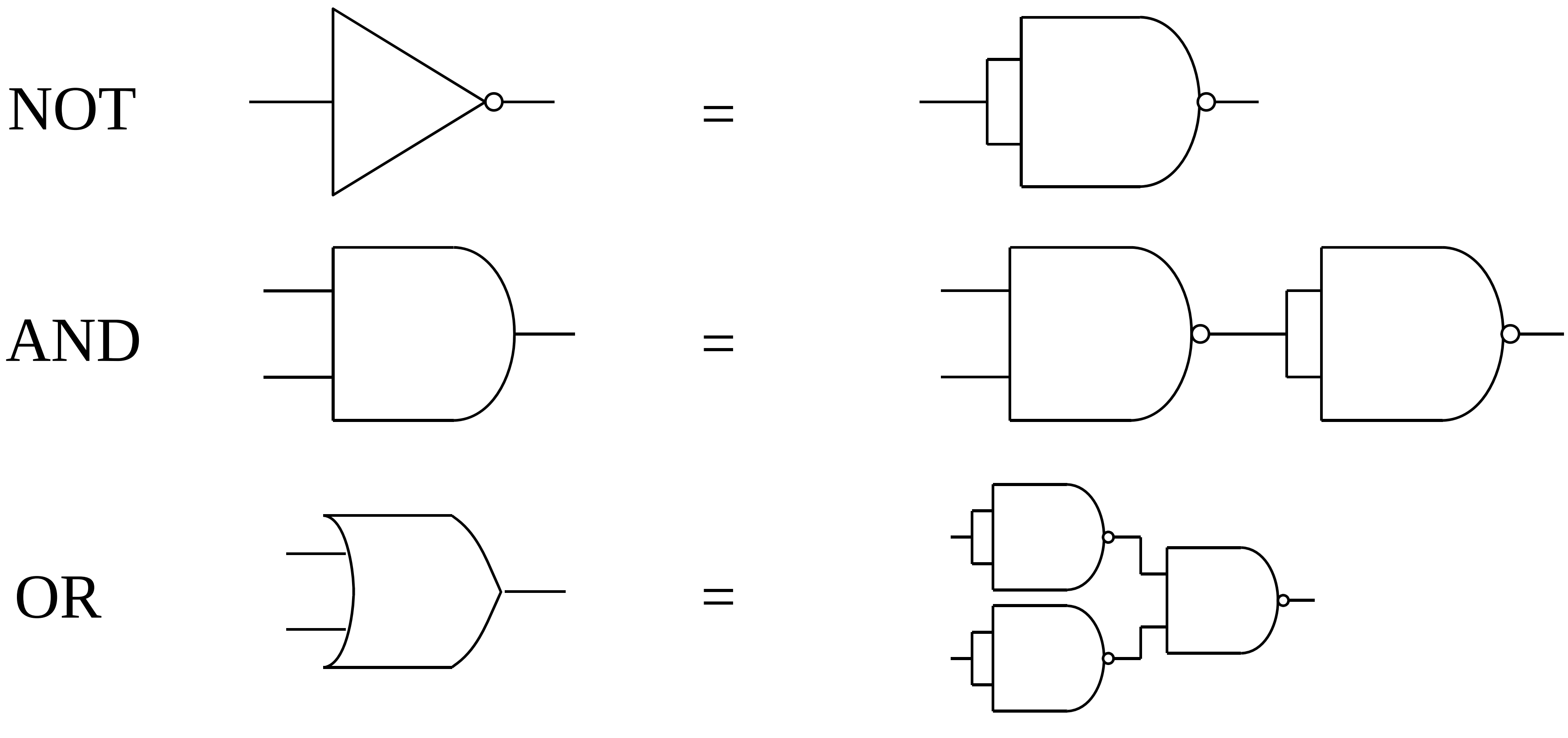


Traditional (American) Symbol

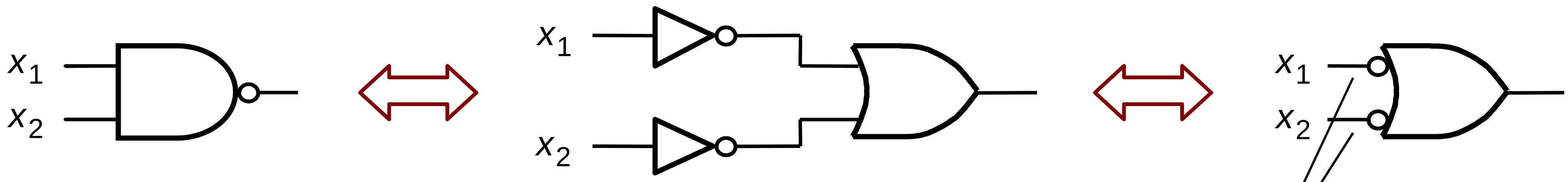


Bara en typ av grind behövs!

För att implementera en boolesk funktion behövs det bara **NAND-** eller **NOR-**grindar



DeMorgans teorem - bubbelgrindar

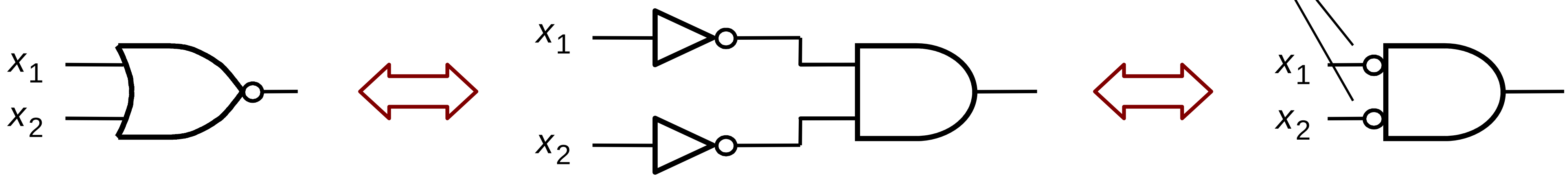


En NAND är en bubbel-OR

(a) $\overline{x_1 x_2} = \bar{x}_1 + \bar{x}_2$

(DeMorgan (15a))

Inverterade ingångar

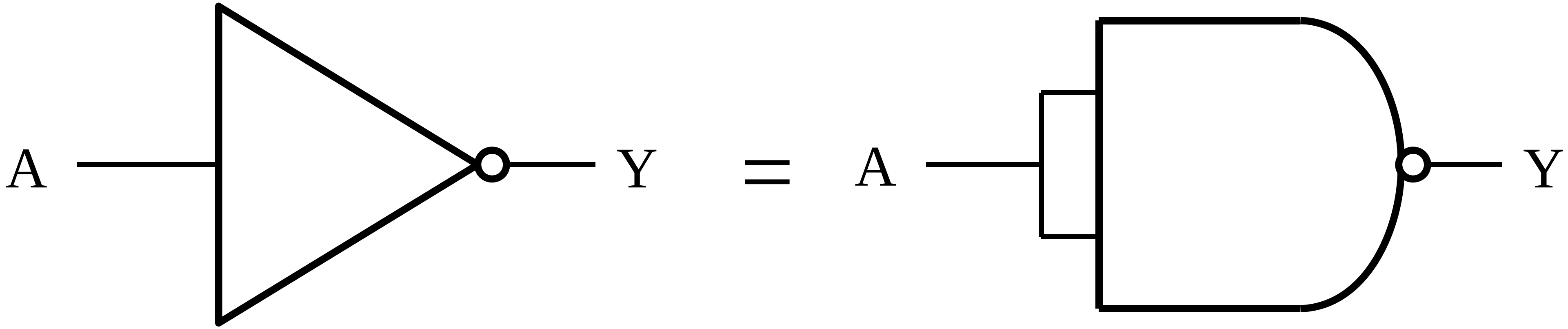


En NOR är en bubbel-AND

(b) $\overline{x_1 + x_2} = \bar{x}_1 \bar{x}_2$

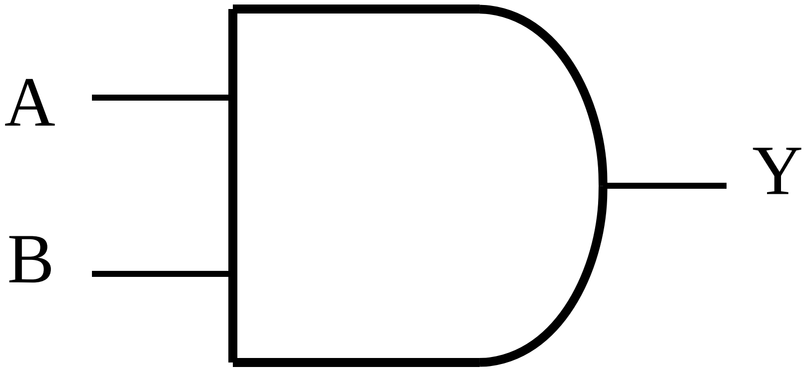
(DeMorgan (15b))

Inverterare med NAND



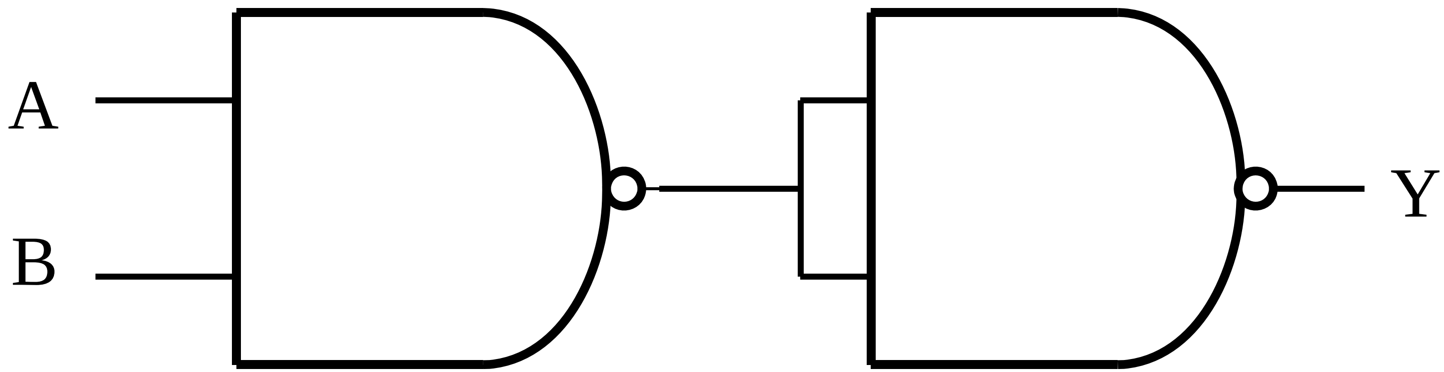
$$Y = \overline{A} = \overline{A \cdot A}$$

AND-grind med NAND-grindar

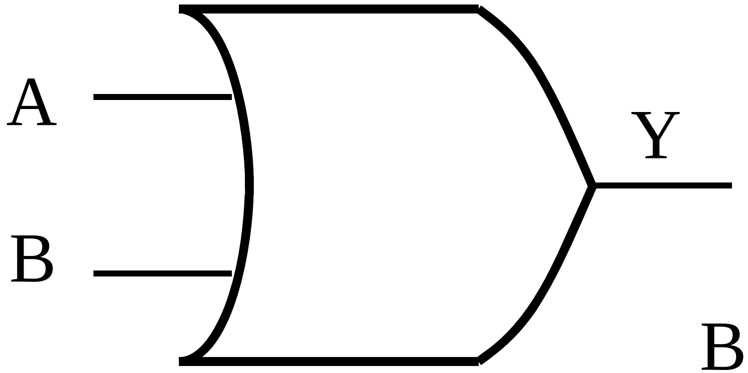


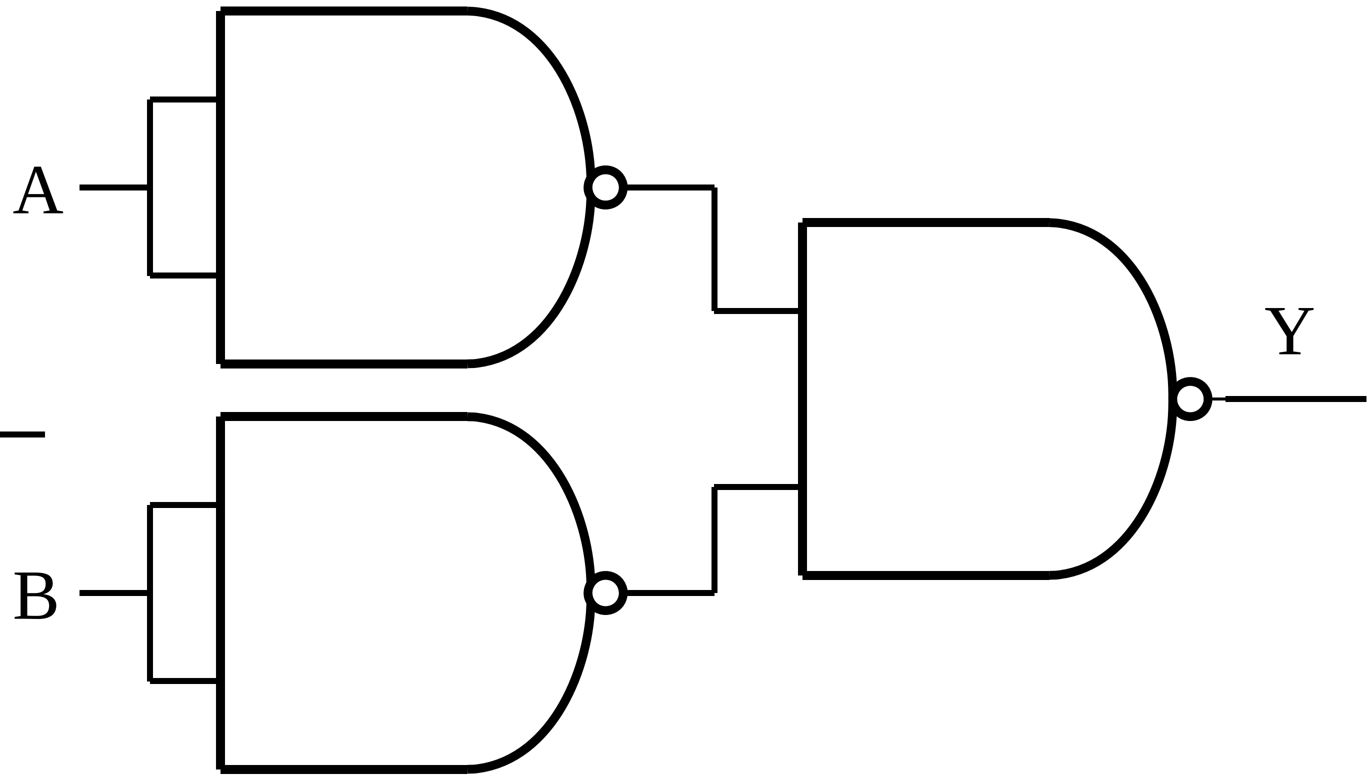
$$Y = A \cdot B = \overline{\overline{A \cdot B}}$$

$=$



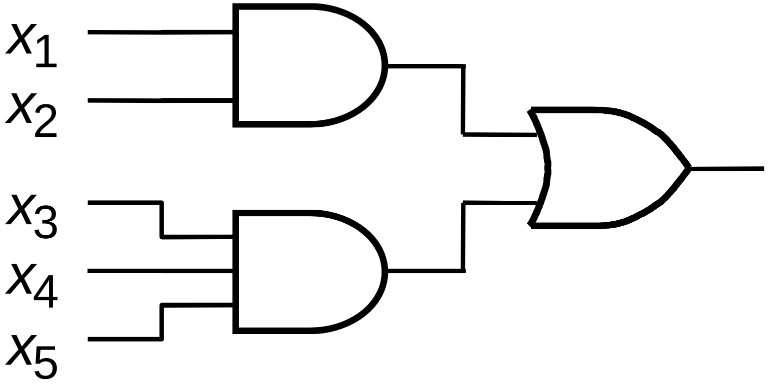
OR-grind med NAND-grindar



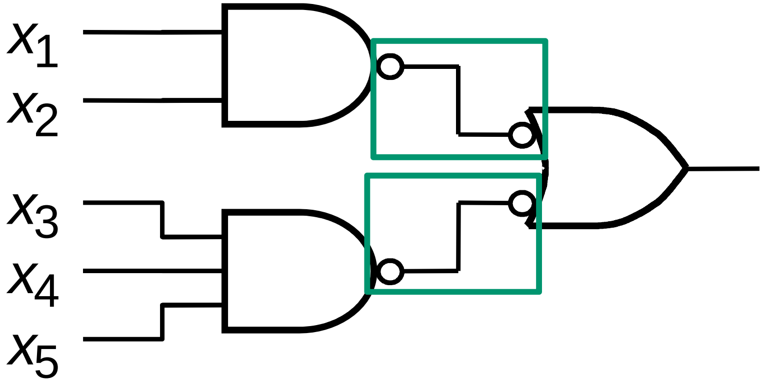


$$Y = A + B = \overline{\overline{A + B}} = \overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A \cdot A} \cdot \overline{B \cdot B}}$$

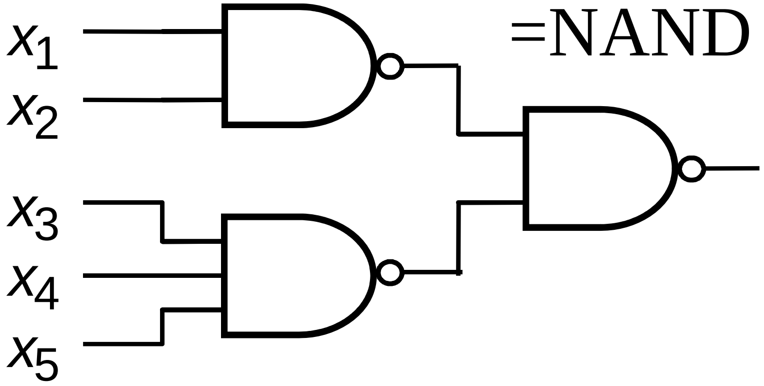
Logiska funktioner med bara NAND



AND-OR funktion



BubbelOR
=NAND



Universella mängder av grindar

En mängd (eng. set) av grindar kallas universell eller komplett om alla kombinatoriska system kan beskrivas mha detta set.

Exempel på universella grind-mängder:

{AND, OR, NOT} -> (DeMorgan) -> {AND,NOT} -> {NAND}

{OR, AND, NOT} -> (DeMorgan) -> {OR,NOT} -> {NOR}

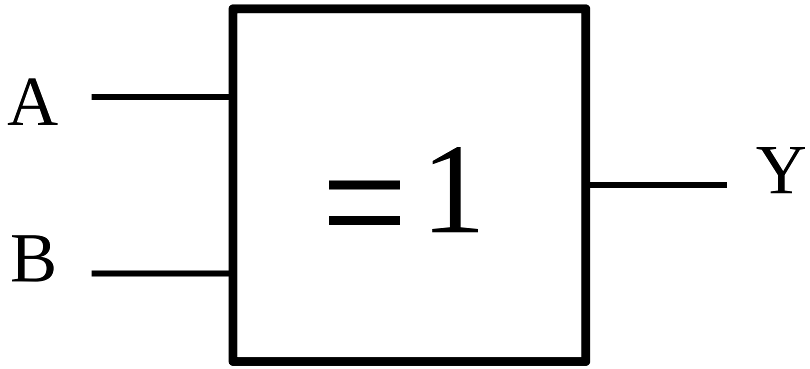
Logiska grindar XOR-grinden

Exklusivt ELLER

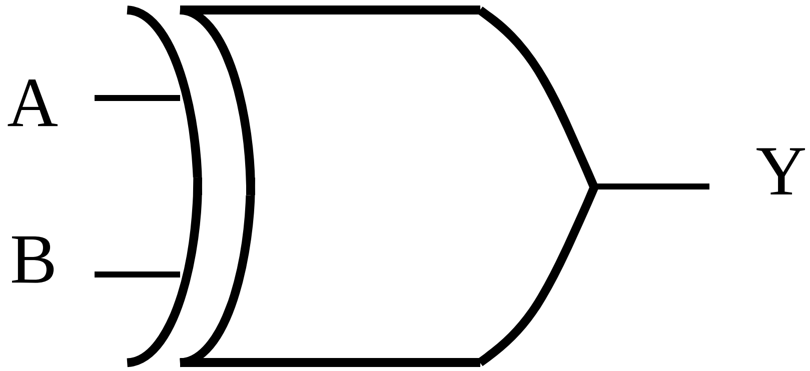
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

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

IEC Symbol
(International Electrotechnical Commission)

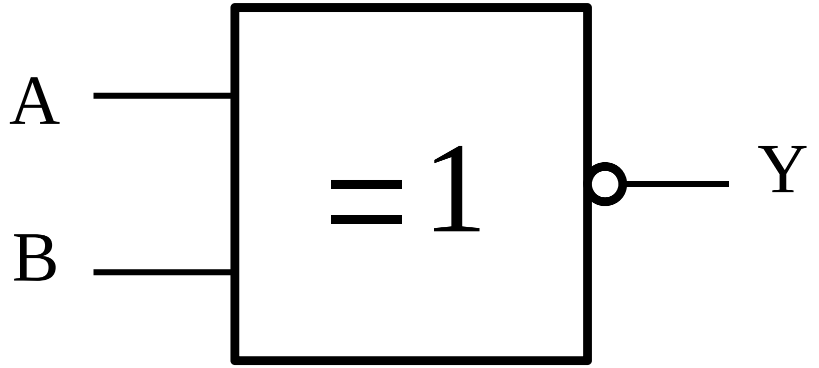


Traditional (American) Symbol



Logiska grindar XNOR-grinden

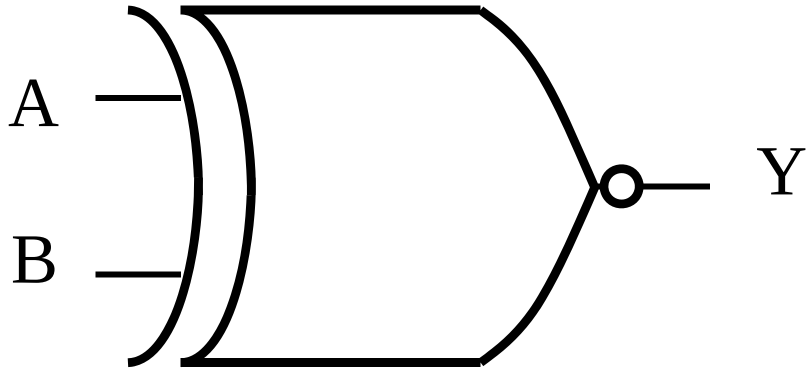
IEC Symbol
(International Electrotechnical Commission)



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

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

Traditional (American) Symbol



Hur många olika grindar finns det?

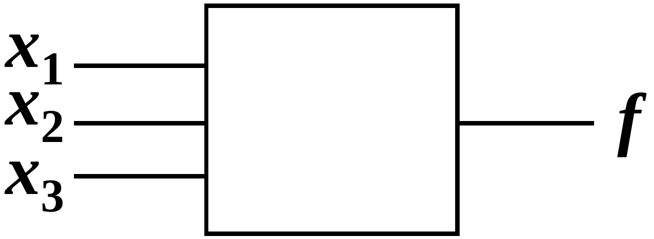
2 ingångar → 4 tillstånd, vart och ett kan ha två värden → $2^4 = 16$

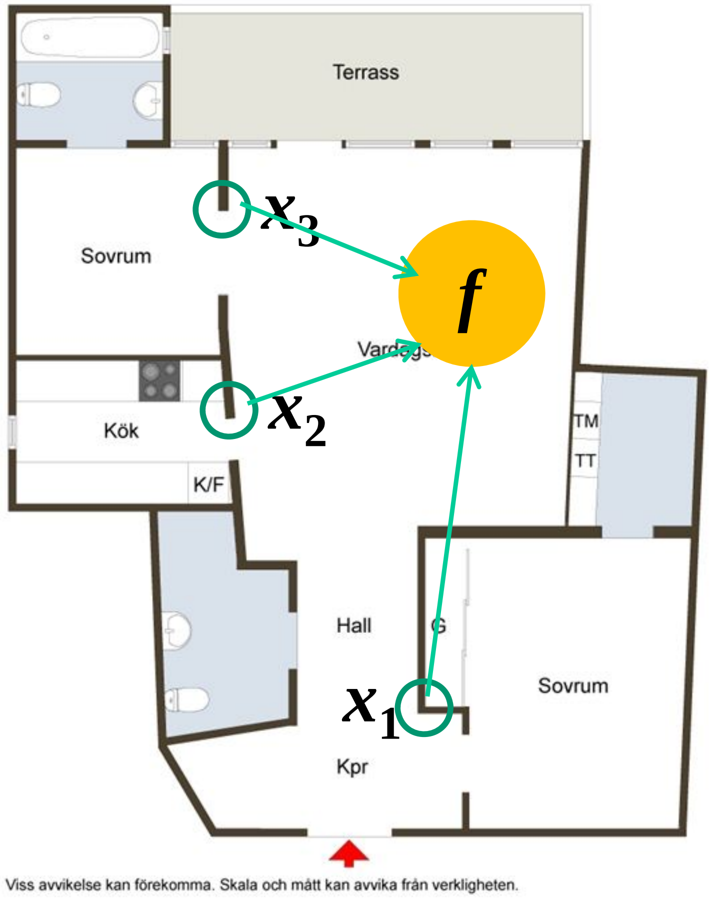
Inputs		16 possible output functions															
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
x	y		AND					XOR	OR	NOR	XNOR					NAND	
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Exempel: Trevägs ljuskontroll

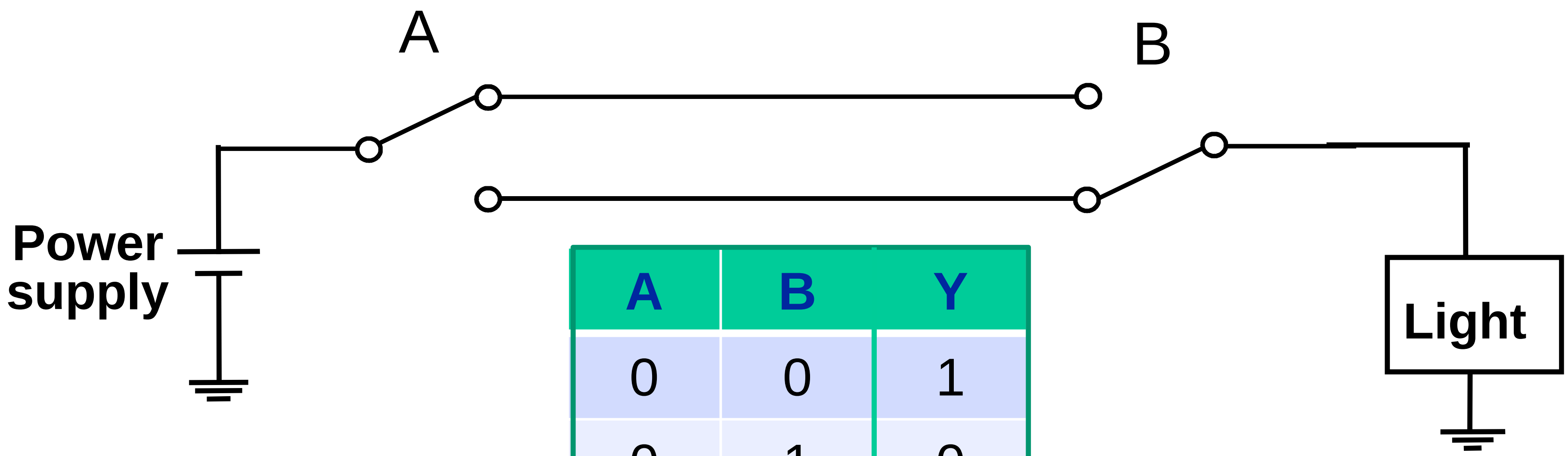
Brown/Vranesic: 2.8.1

Antag att vi behöver kunna tända/släcka vardagsrummet från tre olika ställen.



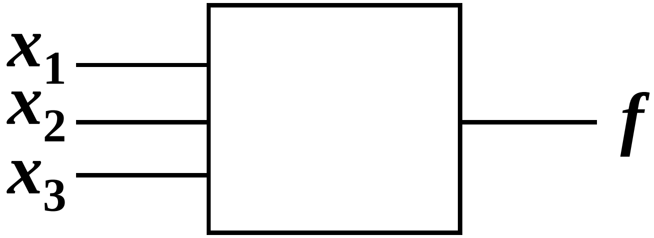


Exempel: Tvåvägs ljuskontroll = trappströmbrytare



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

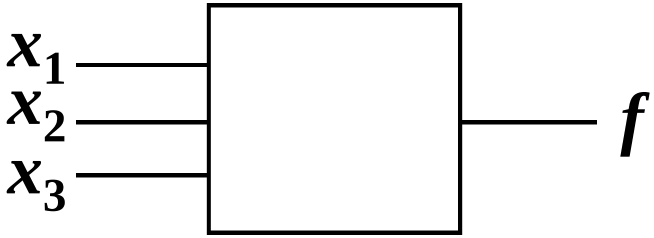
Trevägs ljuskontroll



Man ska alltid kunna *ändra* ljuset genom att *ändra* en valfri strömbrytare.

x_1	x_2	x_3	f
0	0	0	0
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

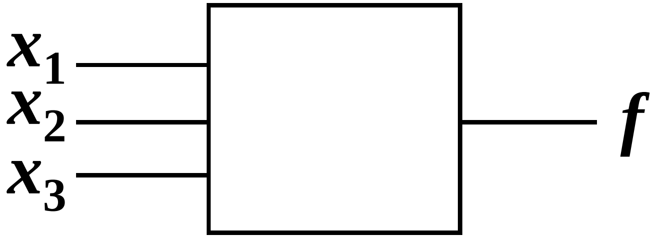
Trevägs ljuskontroll



Man ska alltid kunna ändra ljuset genom att ändra en valfri strömbrytare.

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	
1	0	0	1
1	0	1	
1	1	0	
1	1	1	

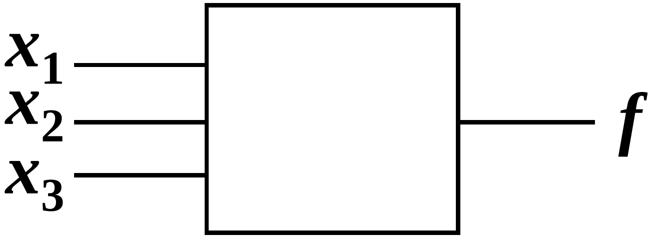
Trevägs ljuskontroll



Man ska alltid kunna ändra ljuset genom att ändra en valfri strömbrytare.

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	
1	1	0	
1	1	1	

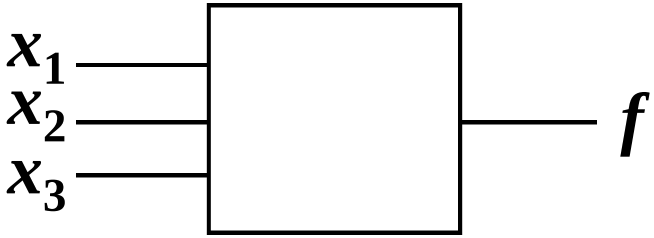
Trevägs ljuskontroll



Man ska alltid kunna ändra ljuset genom att ändra en valfri strömbrytare.

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	

Trevägs ljuskontroll



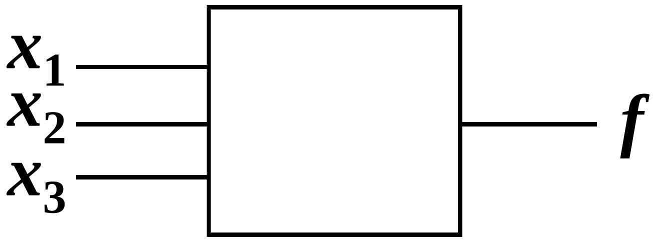
Man ska alltid kunna ändra ljuset genom att ändra en valfri strömbrytare.

Sanningstabellen stämmer nu med förutsättningarna!

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Trevägs ljuskontroll

x_1	x_2	x_3	f	
0	0	0	0	M_0
0	0	1	1	m_1
0	1	0	1	m_2
0	1	1	0	M_3
1	0	0	1	m_4
1	0	1	0	M_5
1	1	0	0	M_6
1	1	1	1	m_7



$$f = \sum m(1,2,4,7) =$$

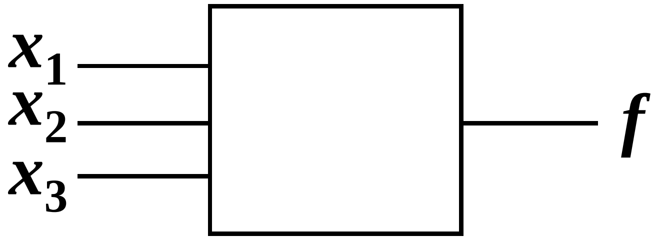
$$= \overline{x}_1\overline{x}_2 x_3 + \overline{x}_1 x_2 \overline{x}_3 + x_1\overline{x}_2\overline{x}_3 + x_1 x_2 x_3$$

eller

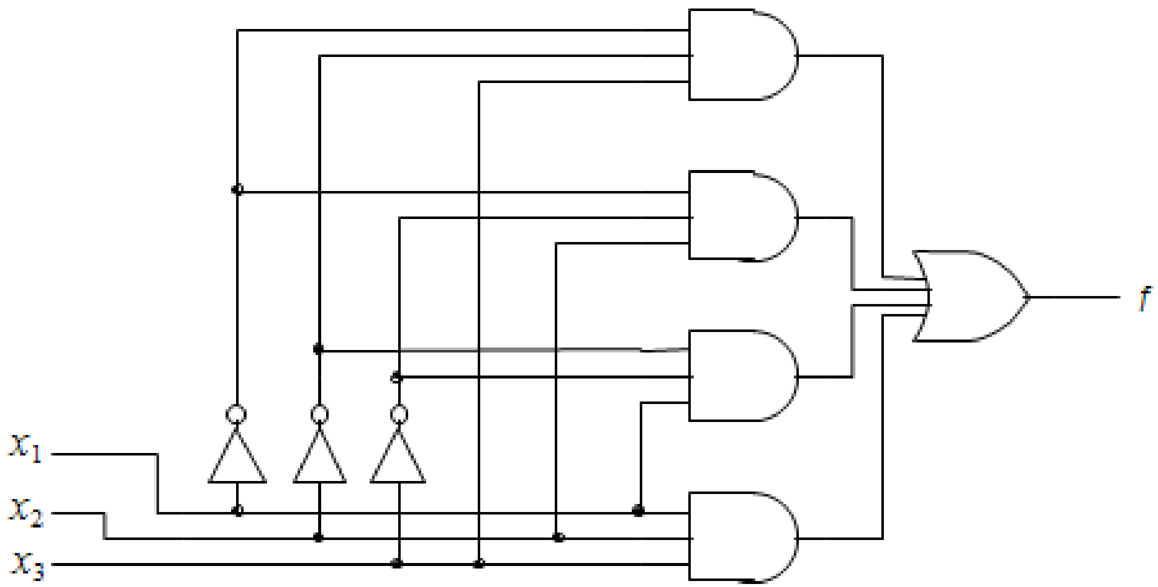
$$f = \prod M(0,3,5,6) =$$

$$= (x_1 + x_2 + x_3)\cdot(x_1 + \overline{x}_2 + \overline{x}_3)\cdot(\overline{x}_1 + x_2 + \overline{x}_3)\cdot(\overline{x}_1 + \overline{x}_2 + x_3)$$

Trevägs ljuskontroll

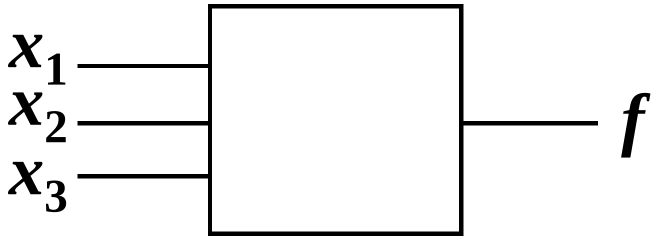


$$f = \sum m(1,2,4,7) = \bar{x}_1\bar{x}_2 x_3 + \bar{x}_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 \bar{x}_3 + x_1 x_2 x_3$$

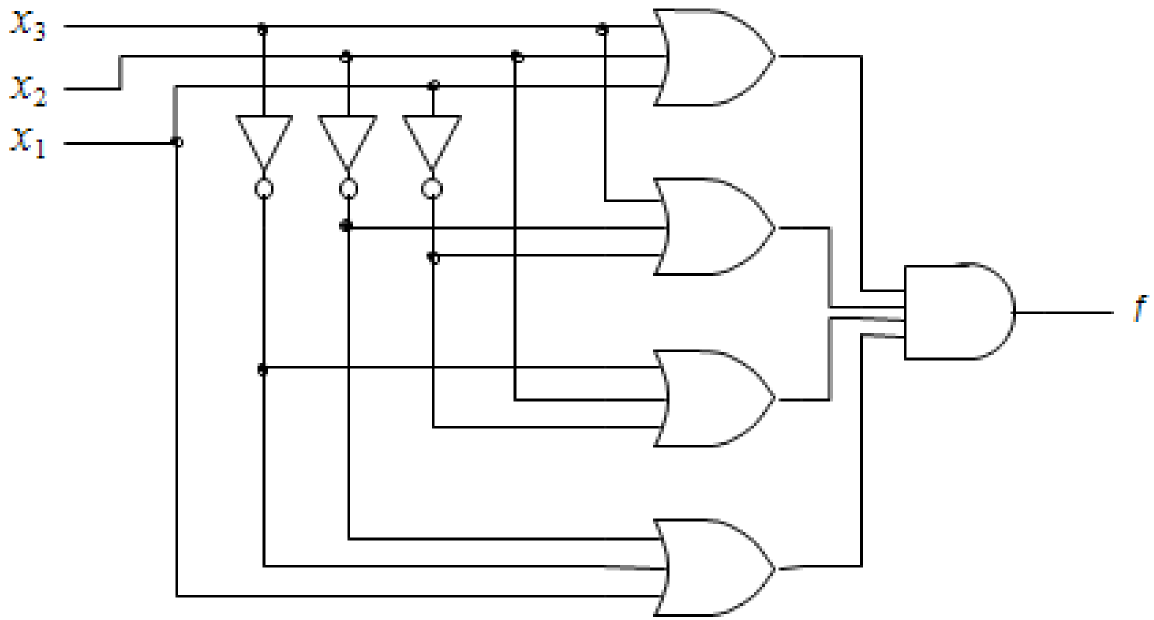


(a) Sum-of-products realization

Trevägs ljuskontroll



$$f = \prod M(0,3,5,6) = (x_1 + x_2 + x_3) \cdot (x_1 + \overline{x}_2 + \overline{x}_3) \cdot (\overline{x}_1 + x_2 + \overline{x}_3) \cdot (\overline{x}_1 + \overline{x}_2 + x_3)$$



(b) Product-of-sums realization

Sammanfattning

- **Logiska funktioner kan beskrivas med boolesk algebra**
- **Det finns logiska grindar för de vanliga booleska funktioner**
- **En logisk funktion kan uttryckas och skrivas om mha boolesk algebra till**

  - SOP-form (Summa av min-termer)

  eller

  - POS-form (Produkt av max-termer)

Två övningsgrupper på svenska

- Tillfälle 1: Efternamn A - K
  Första övningen 1/11 8-10 i **sal 205**
- Tillfälle 2: Efternamn L - Ö
  Första övningen 1/11 13-15 i **sal B**

- **Två salar bokade men bara den med lägst nummer används: B < 205 < 208 < 308**