

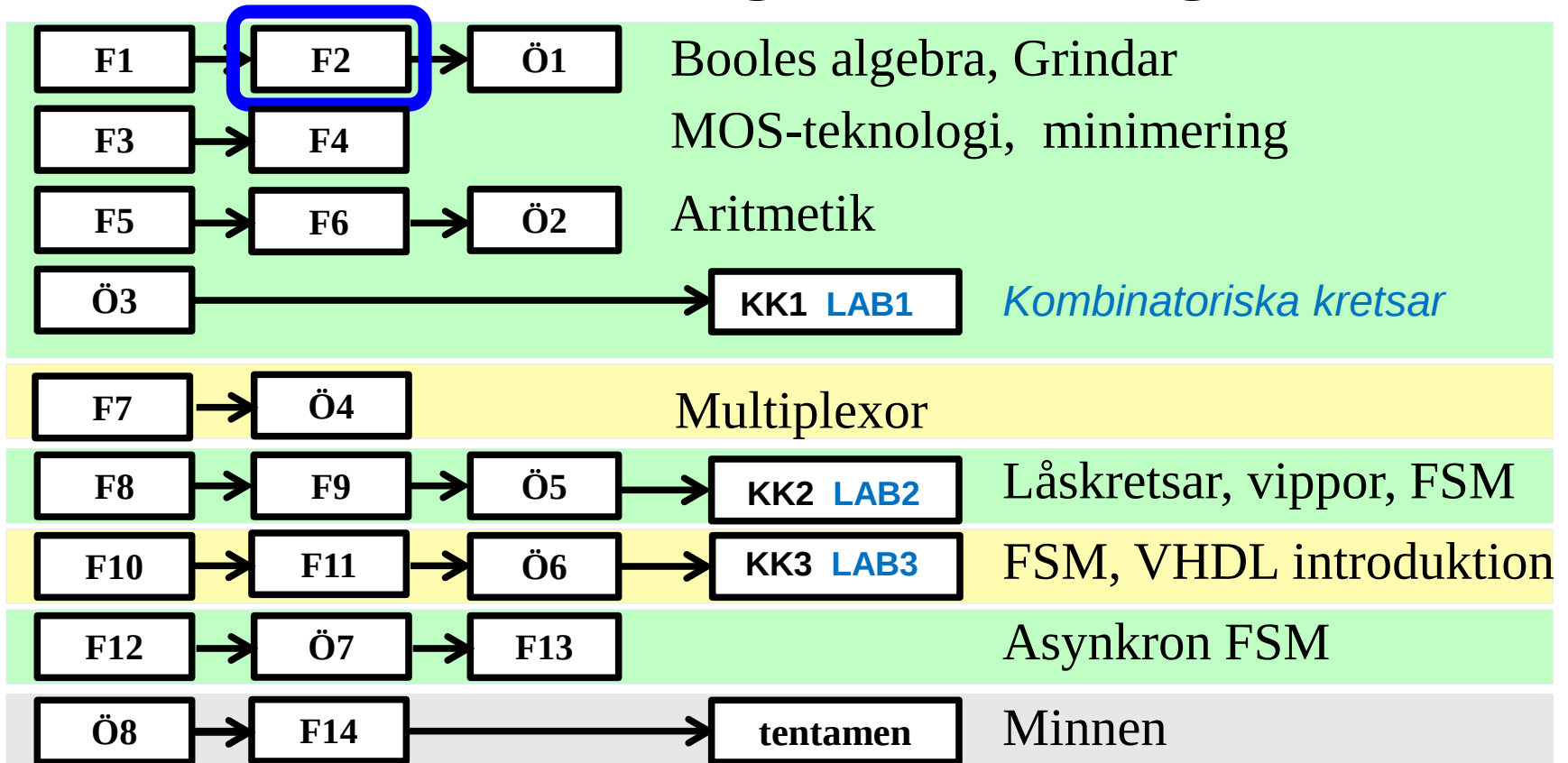
Digital Design IE1204

Kursomgång för Högskoleingenjörsinriktningarna:
Datateknik, Elektronik och Datorteknik.

**F2 : Logiska Grindar och
Kretsar, Boolesk Algebra**

`william@kth.se`

IE1204 Digital Design



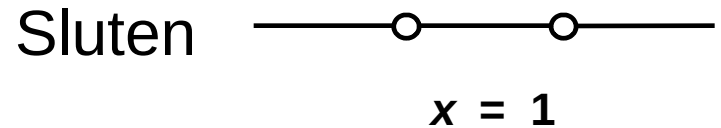
*Föreläsningar och övningar bygger på varandra! Ta alltid igen det Du missat!
Läs på i förväg – delta i undervisningen – arbeta igenom materialet efteråt!*

Detta har hänt ...

Talsystem: Decimala, hexadecimala, oktala, binära

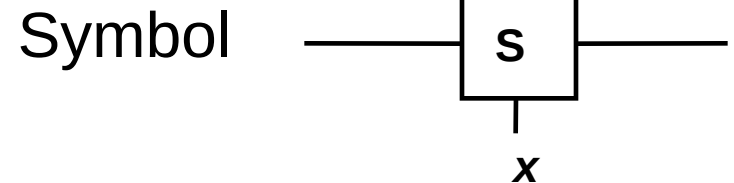
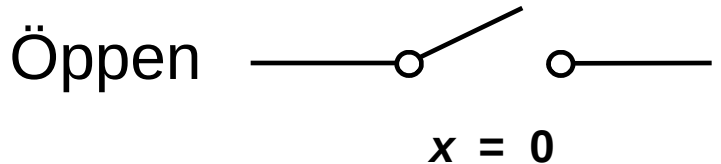
$$(175,5)_{10} = (AE.8)_{16} = (256.4)_8 = (10101110.1)_2$$

Switch



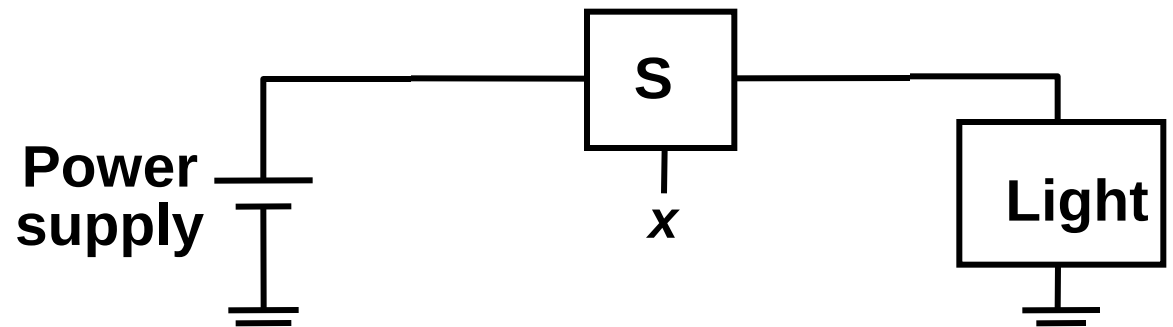
En switch har två lägen

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



Implementering av logiska funktioner

Switchen kan användas för att implementera logiska funktioner

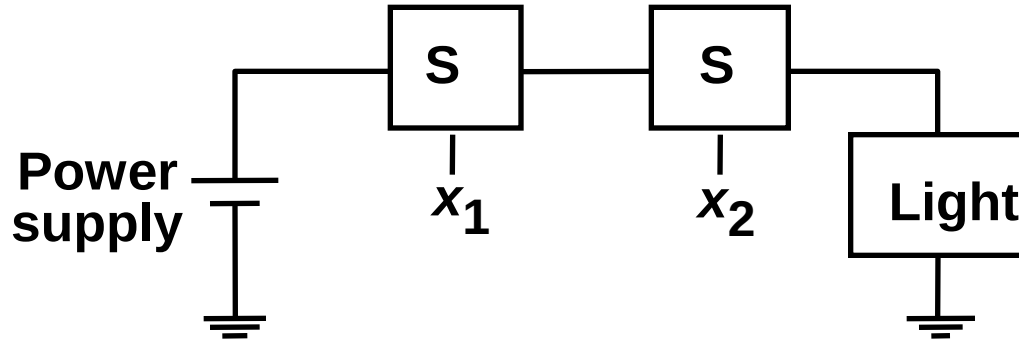


$L(x)$ är en logisk funktion
 x är en logisk variabel

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

Operation **AND** (OCH)

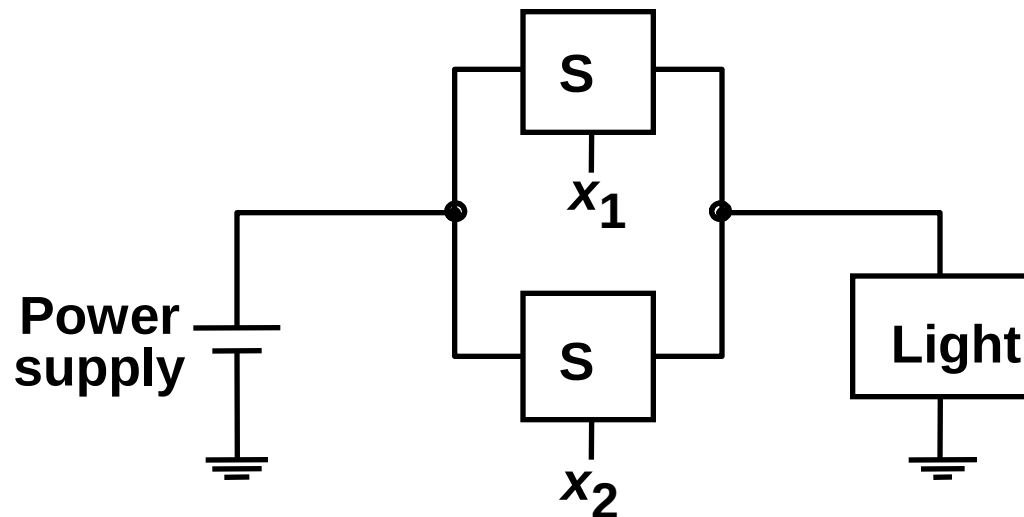
AND-operationen ($\bullet$) uppnås genom switchar som kopplas i serie



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

Operation OR (ELLER)

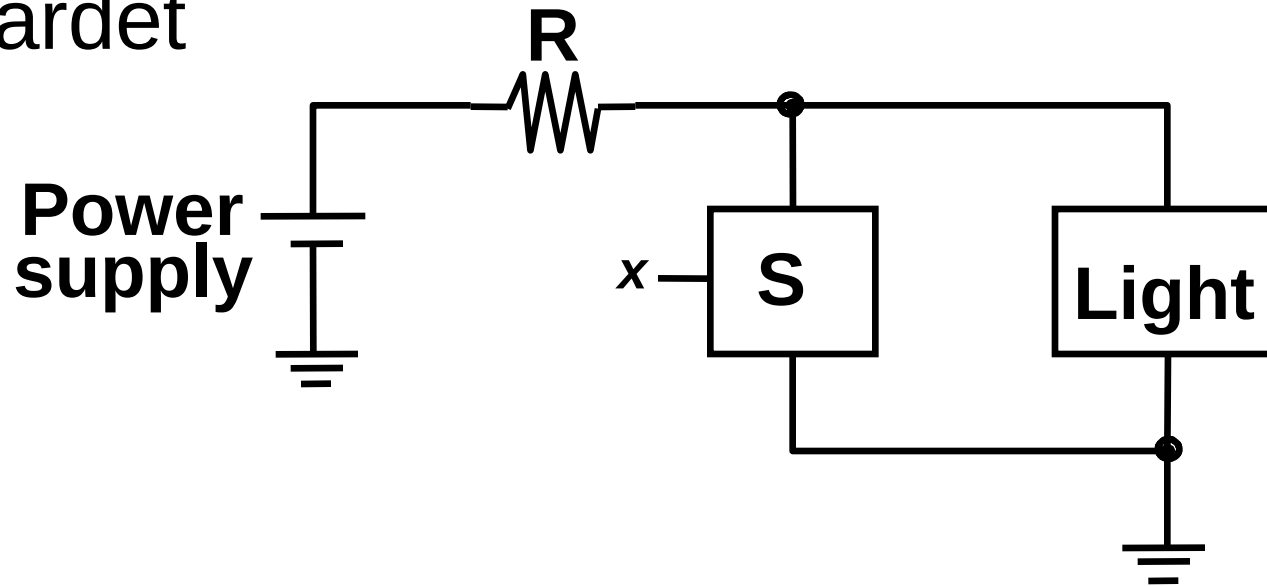
OR-operationen (+) uppnås genom switchar som kopplas parallellt



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

Operation NOT (ICKE)

NOT-funktionen inverterar det logiska värdet



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

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)

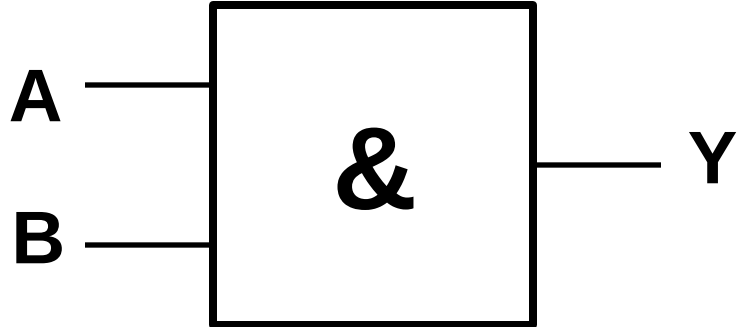
Logiska grindar

AND-grinden (OCH)

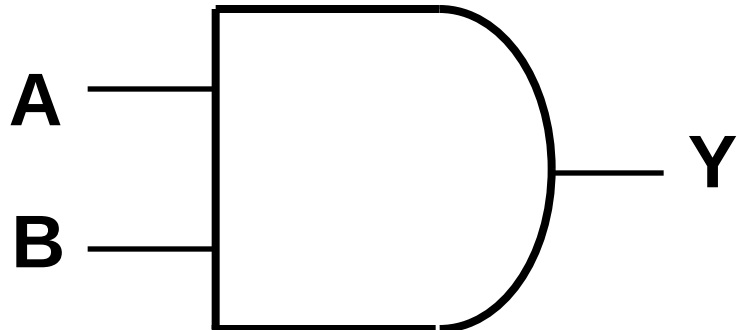
IEC Symbol
(International Electrotechnical Commission)

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

$$Y = A \cdot B$$



Traditional (American) Symbol



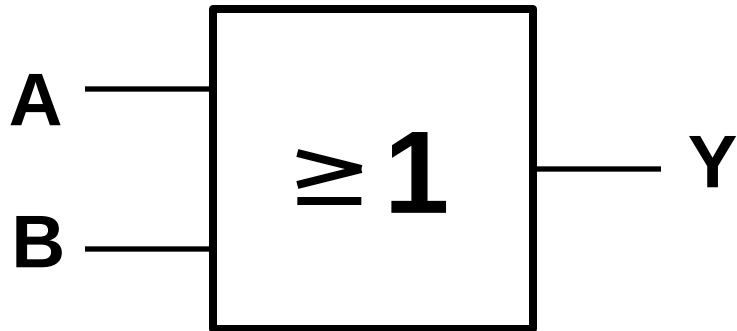
Logiska grindar

OR-grinden (ELLER)

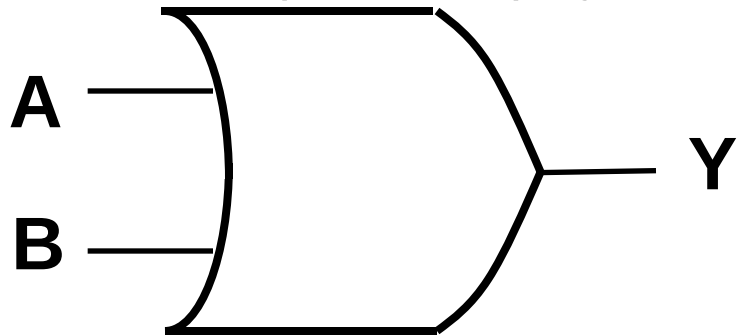
IEC Symbol
(International Electrotechnical Commission)

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

$$Y = A + B$$



Traditional (American) Symbol



Logiska grindar

inverterare NOT (ICKE)

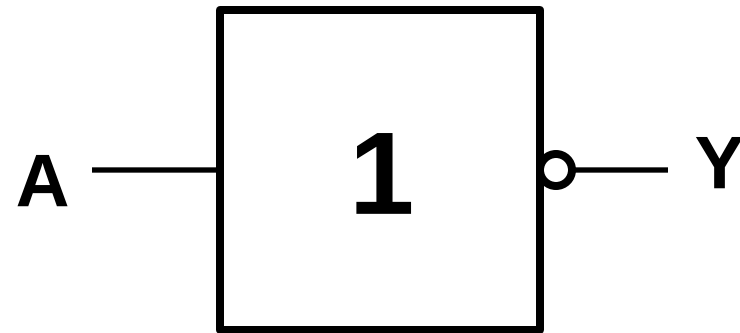
IEC Symbol

(International Electrotechnical Commission)

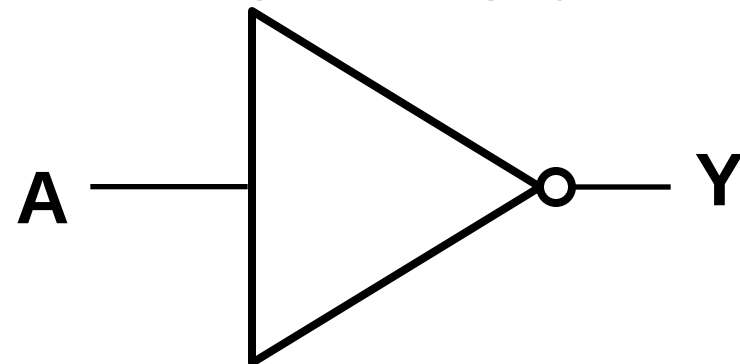
**Inverterare
(Inverter)**

A	Y
0	1
1	0

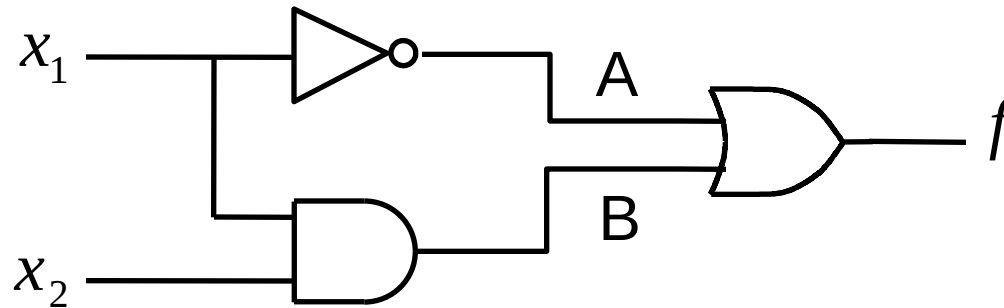
$$Y = \bar{A}$$



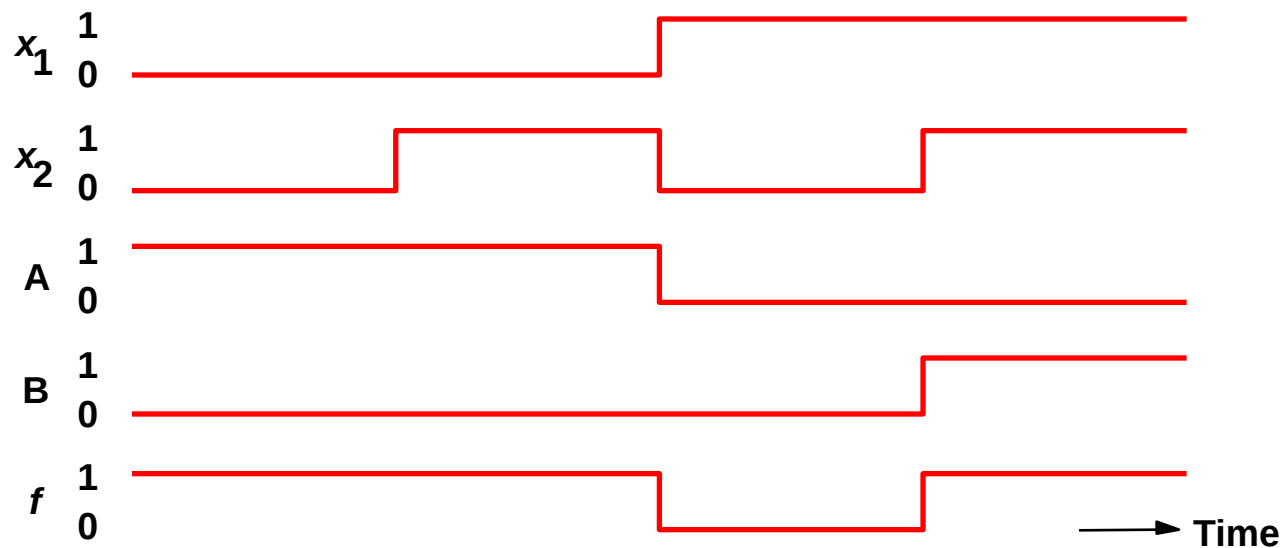
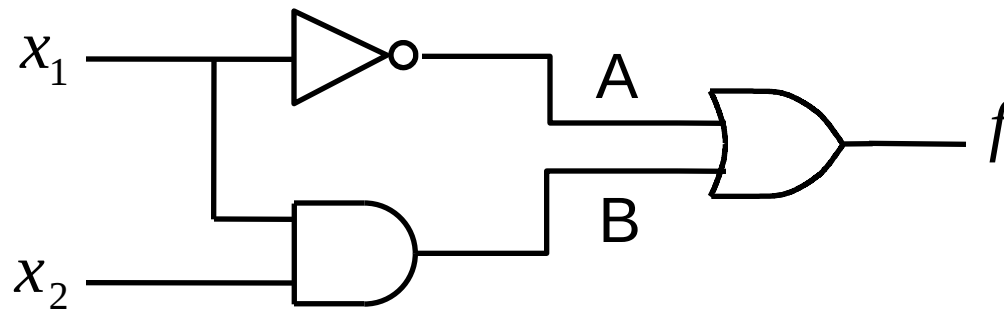
Traditional (American) Symbol



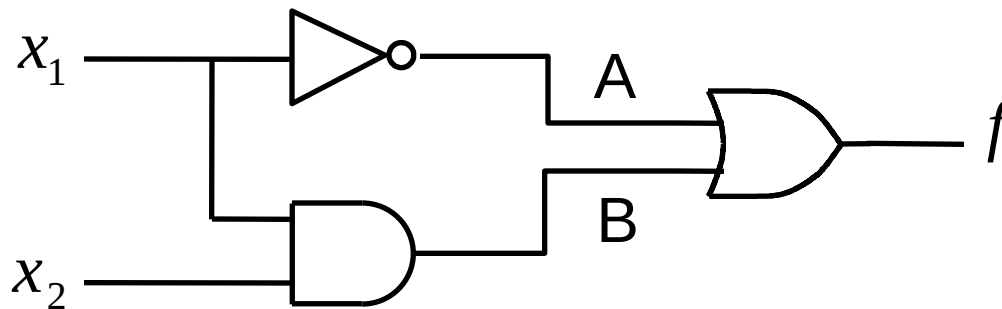
Vilken funktion har grindnätet?



Tidsdiagram



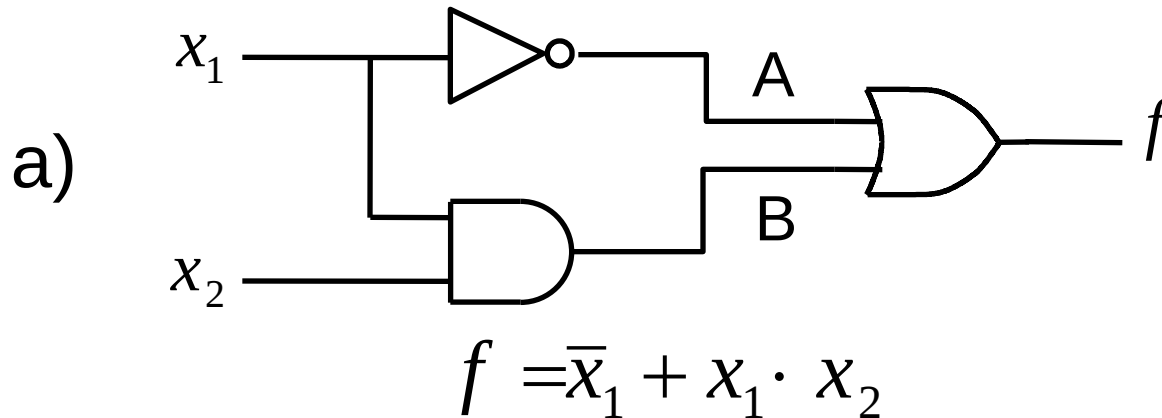
Sanningstabel



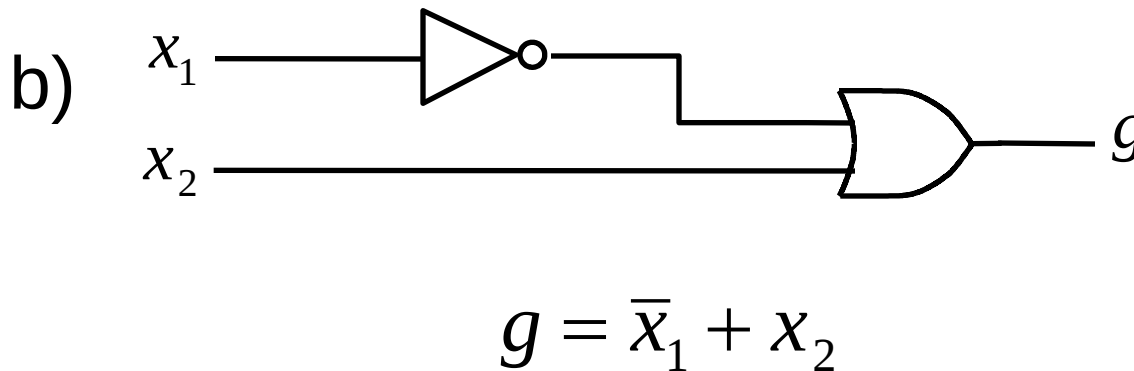
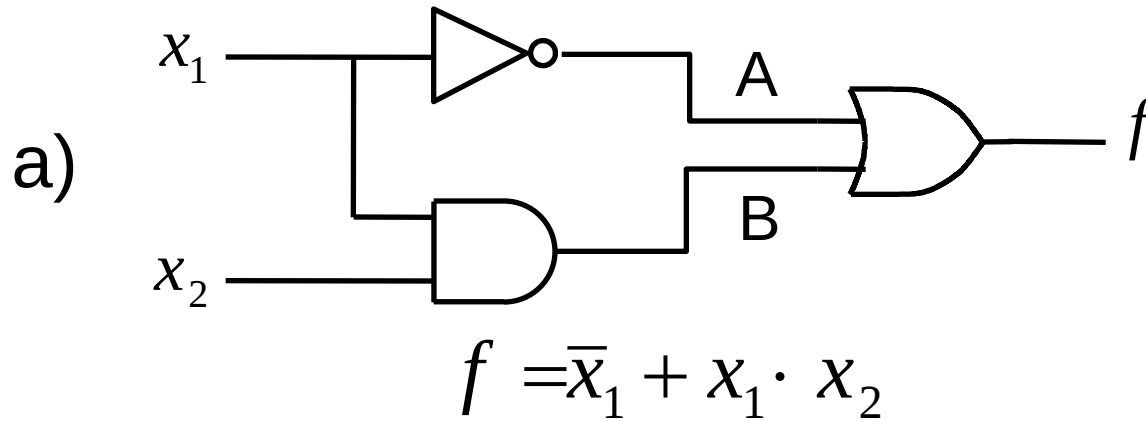
x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

A	B
1	0
1	0
0	0
0	1

Flera grindnät kan implementera samma funktion!



Flera grindnät kan implementera samma funktion!



$f = g$

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

Boolesk algebra

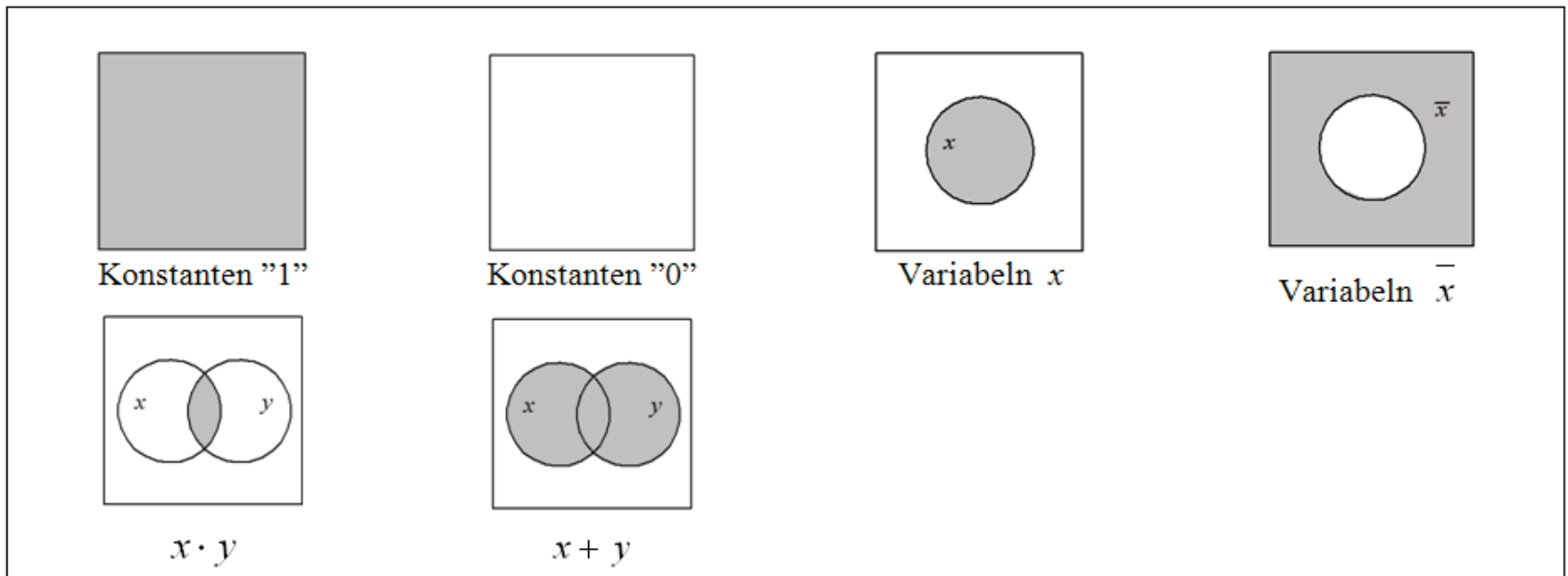
- Eftersom flera grindnät kan implementera samma funktion, så vill man hitta den ***mest kostnadseffektiva implementeringen***
- Grindnäten kan bli mycket stora
- En matematisk bas behövs så att automatiseringen av grindnätsoptimering kan genomföras med datorer

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 $\bar{x} = 1$	(4b) If $x = 1$, then $\bar{x} = 0$

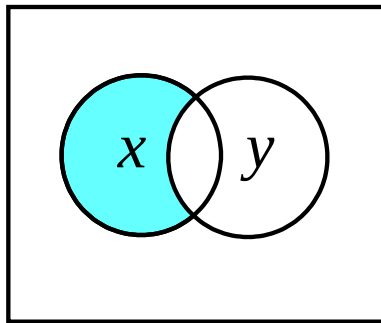
Venn-diagram

Venn-diagrammet kan användas för att illustrera logiska operationer

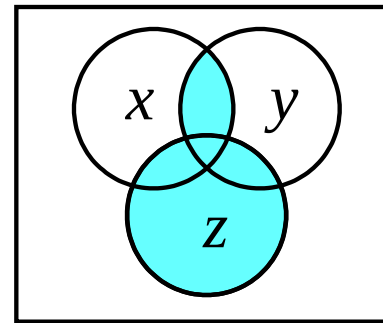


Venn-diagram

Venn-diagrammet kan användas för att illustrera logiska operationer

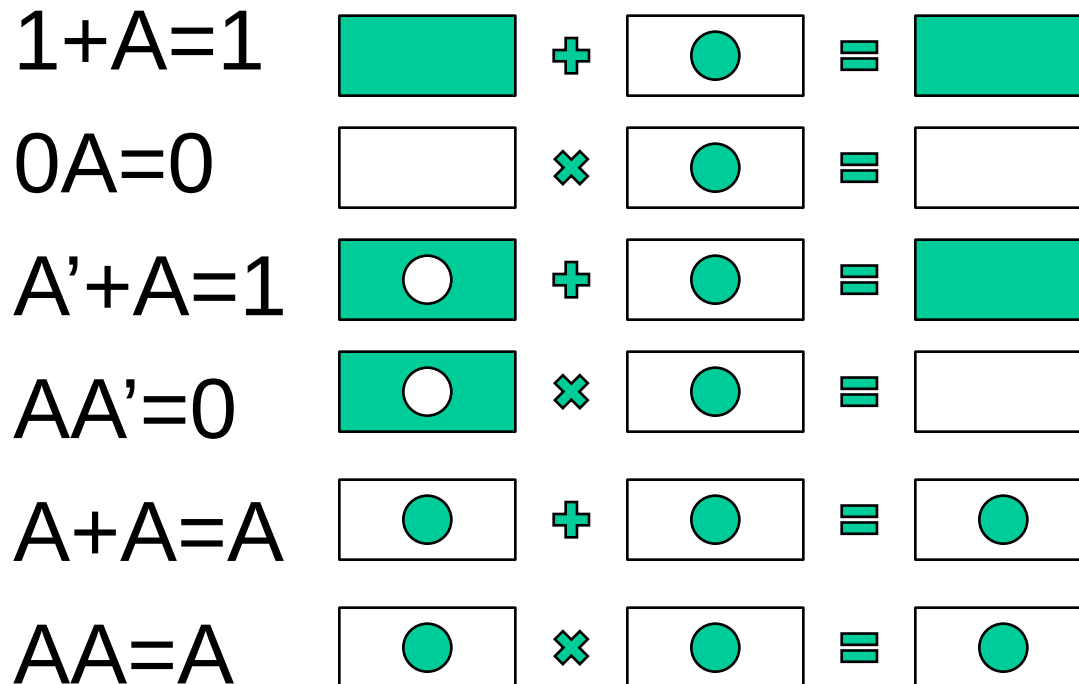


$$x \cdot \bar{y}$$



$$x \cdot y + z$$

Boolesk algebra med Venn-diagram



Booles algebra

enkla räkneregler

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

Räknelagar	
(5a) $x \cdot 0 = 0$	(5b) $x + 1 = 1$
(6a) $x \cdot 1 = x$	(6b) $x + 0 = x$
(7a) $x \cdot x = x$	(7b) $x + x = x$
(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) $x + y = y + x$	<i>kommutativ</i> <i>associativ</i> <i>distributiv</i> <i>absorption</i>
(11a) $x \cdot (y \cdot z) = (x \cdot y) \cdot z$	(11b) $x + (y + z) = (x + y) + z$	
(12a) $x \cdot (y + z) = x \cdot y + x \cdot z$	(12b) $x + y \cdot z = (x + y) \cdot (x + z)$	
(13a) $x + x \cdot y = x$	(13b) $x \cdot (x + y) = x$	
(14a) $x \cdot y + x \cdot \bar{y} = x$	(14b) $(x + y) \cdot (x + \bar{y}) = x$	<i>DeMorgan</i>
(15a) $\overline{x \cdot y} = \bar{x} + \bar{y}$	(15b) $\overline{x + y} = \bar{x} \cdot \bar{y}$	
(16a) $x + \bar{x} \cdot y = x + y$	(16b) $x \cdot (\bar{x} + y) = x \cdot y$	<i>consensus</i>
(17a) $x \cdot y + y \cdot z + \bar{x} \cdot z$ $= x \cdot y + \bar{x} \cdot z$	(17b) $(x + y) \cdot (y + z) \cdot (\bar{x} + z)$ $= (x + y) \cdot (\bar{x} + z)$	

Exempel

Bevisa konsensuslagen (17a)

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

Bevis av en av konsensuslagarna

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

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

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

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

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

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

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

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 sanningsstabell 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
		0	1
		1	0
		1	1

1 OK

1 OK

0 ej OK

1 OK



Vild gissning:

Varna för **tryck** på samtidigt som **kran** stängd.

Hur kan följande sanningsstabell 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

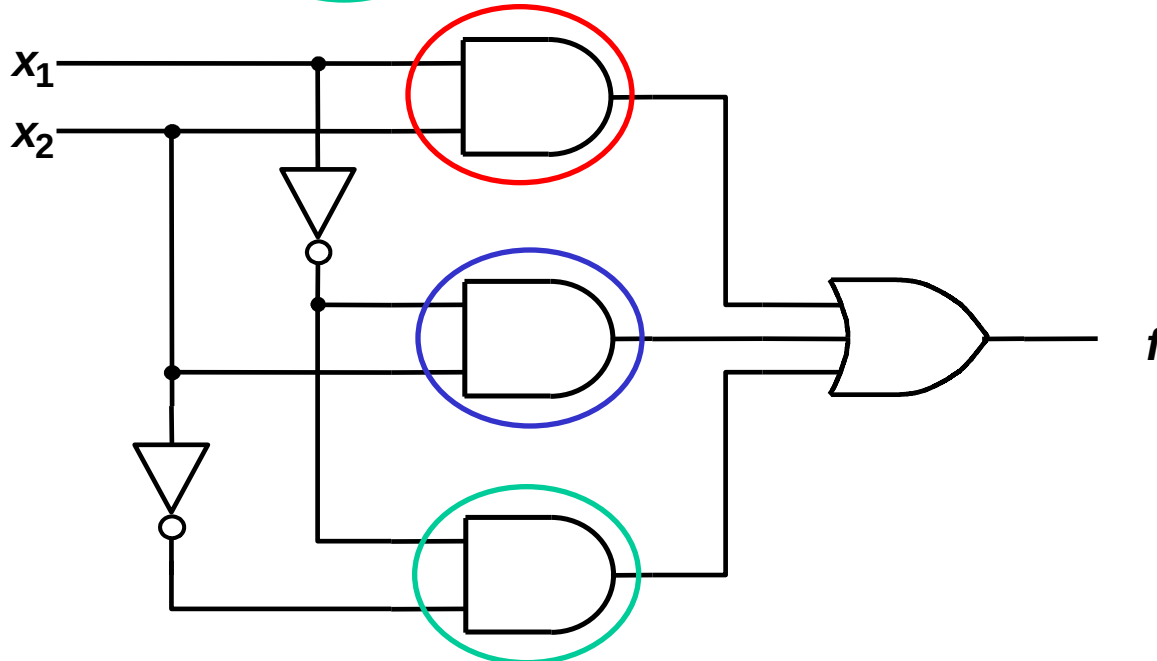
1. Ta fram den logiska funktionen.

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

Hur kan följande sanningstabel implementeras med logiska grindar?

2. Gör en direkt implementering av den logiska funktionen.

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



Hur kan följande sanningsstabell implementeras med logiska grindar?

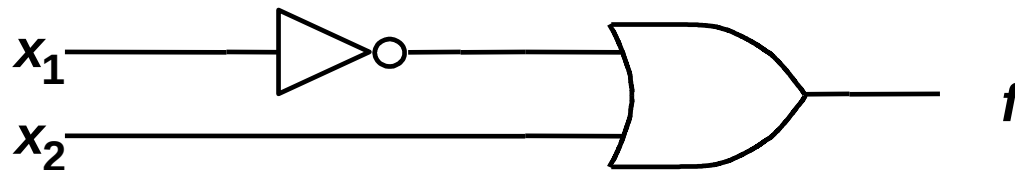
2. (Bättre) Minimera den logiska funktionen

$$\begin{aligned} f &= \bar{x}_1\bar{x}_2 + \bar{x}_1x_2 + x_1x_2 \\ &= \bar{x}_1\bar{x}_2 + \bar{x}_1x_2 + \bar{x}_1x_2 + x_1x_2 && \text{Lägg till redundant term } \bar{x}_1x_2 \text{ (7b)} \\ &= \bar{x}_1(\bar{x}_2 + x_2) + (\bar{x}_1 + x_1)x_2 && \text{Distribution (12a)} \\ &= \bar{x}_1 \cdot 1 + 1 \cdot x_2 && (8b) \\ &= \bar{x}_1 + x_2 \end{aligned}$$

Hur kan följande sanningsstabell implementeras med logiska grindar?

3. Implementera den minimerade funktionen

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



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 Maxtermmer

Row number	x_1	x_2	x_3	= 1 Minterm	= 0 Maxterm
0	0	0	0	$m_0 = \bar{x}_1\bar{x}_2\bar{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \bar{x}_1\bar{x}_2x_3$	$M_1 = x_1 + x_2 + \bar{x}_3$
2	0	1	0	$m_2 = \bar{x}_1x_2\bar{x}_3$	$M_2 = x_1 + \bar{x}_2 + x_3$
3	0	1	1	$m_3 = \bar{x}_1x_2x_3$	$M_3 = x_1 + \bar{x}_2 + \bar{x}_3$
4	1	0	0	$m_4 = x_1\bar{x}_2\bar{x}_3$	$M_4 = \bar{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\bar{x}_2x_3$	$M_5 = \bar{x}_1 + x_2 + \bar{x}_3$
6	1	1	0	$m_6 = x_1x_2\bar{x}_3$	$M_6 = \bar{x}_1 + \bar{x}_2 + x_3$
7	1	1	1	$m_7 = x_1x_2x_3$	$M_7 = \bar{x}_1 + \bar{x}_2 + \bar{x}_3$

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 Products **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

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

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

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

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 **Maxterm** och mellan **SP** och **PS**

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

$$M_0 = \overline{m}_0 = \overline{\overline{x}_1 \cdot \overline{x}_2 \cdot \overline{x}_3} = \overline{\overline{x}_1} + \overline{\overline{x}_2} + \overline{\overline{x}_3} = x_1 + x_2 + x_3$$

(mha DeMorgan 15a)

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

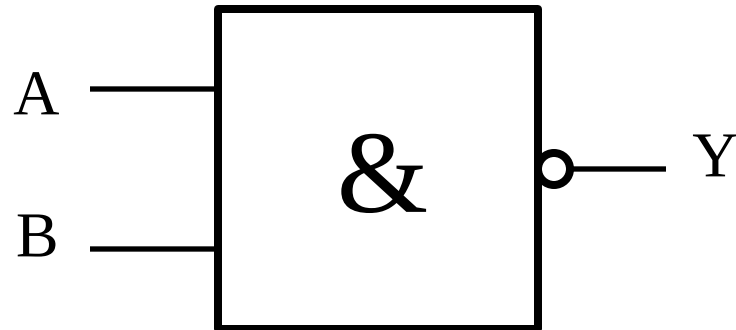
$$f = \sum m(1, 4, 5, 6) = \prod M(0, 2, 3, 7)$$

Logiska grindar NAND-grinden

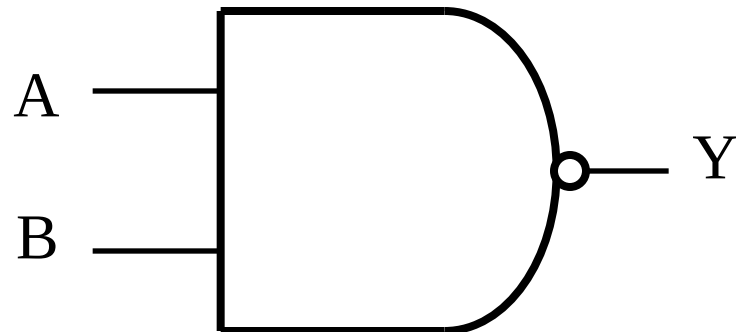
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

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

IEC Symbol
(International Electrotechnical Commission)



Traditional (American) Symbol

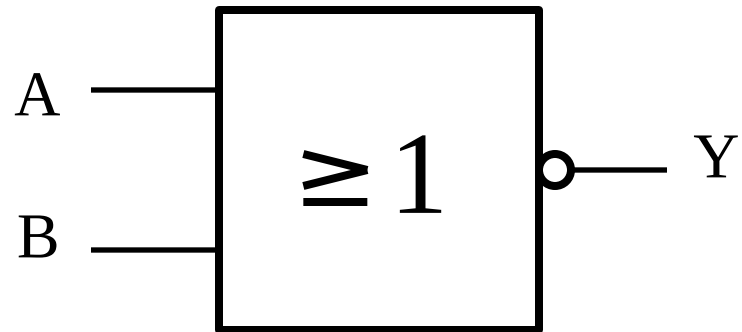


Logiska grindar NOR-grinden

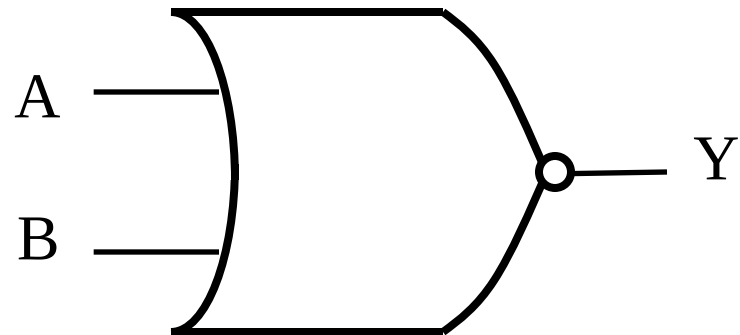
IEC Symbol
(International Electrotechnical Commission)

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

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

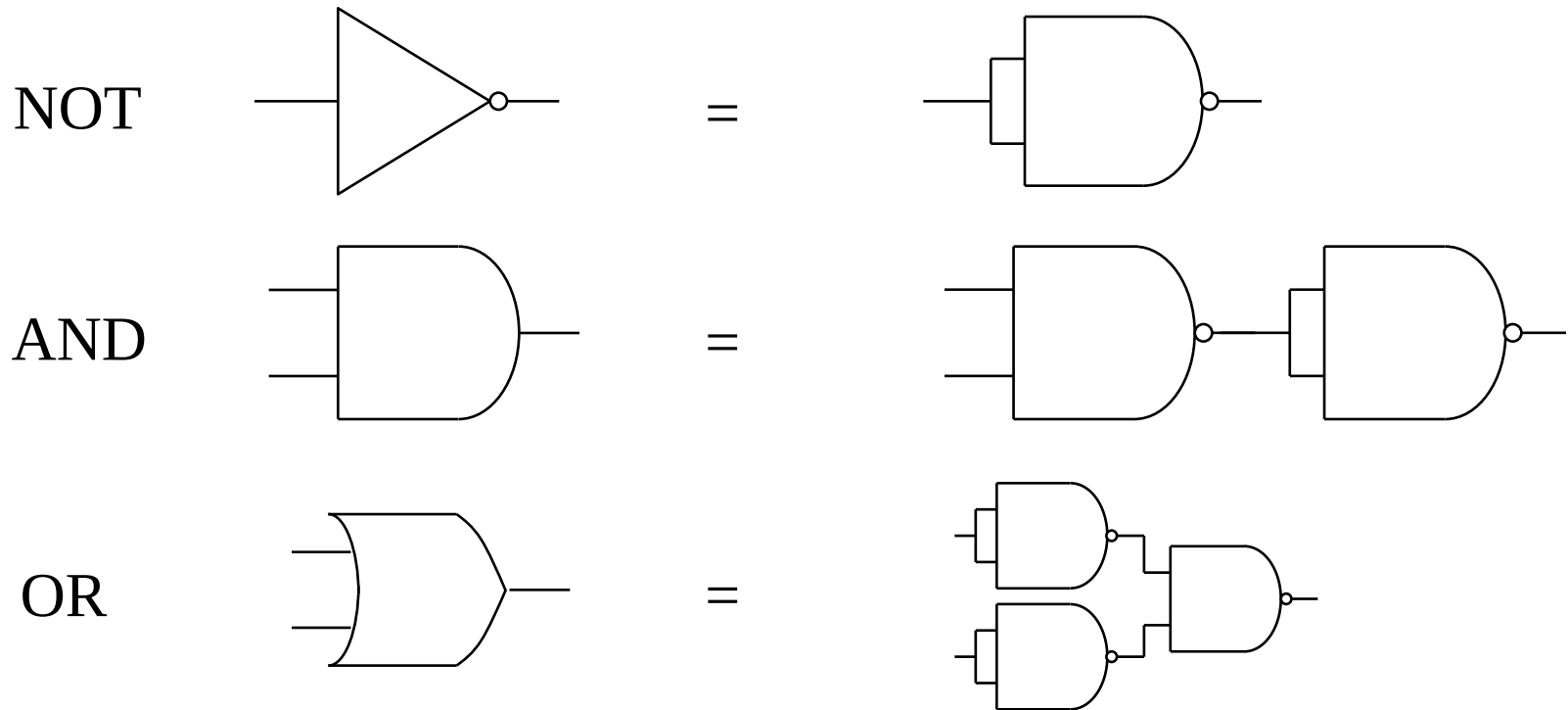


Traditional (American) Symbol

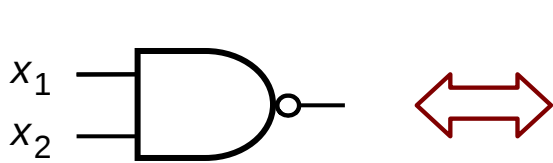


Bara en typ av grind behövs!

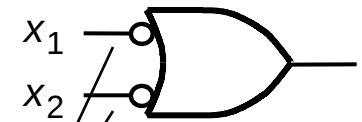
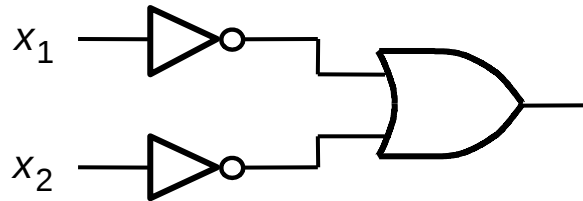
För att implementera en boolesk funktion behövs det bara **NAND**- eller **NOR**-grindar



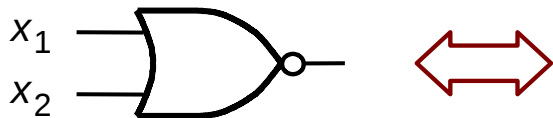
DeMorgans teorem - bubbelgrindar



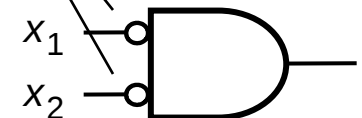
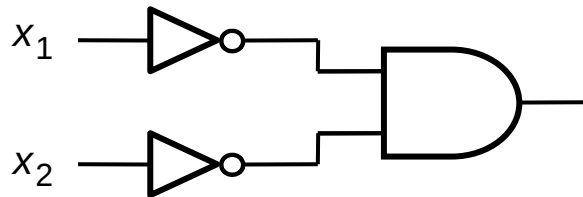
En NAND är en
bubbel-OR



Inverterade ingångar



En NOR är en
bubbel-AND



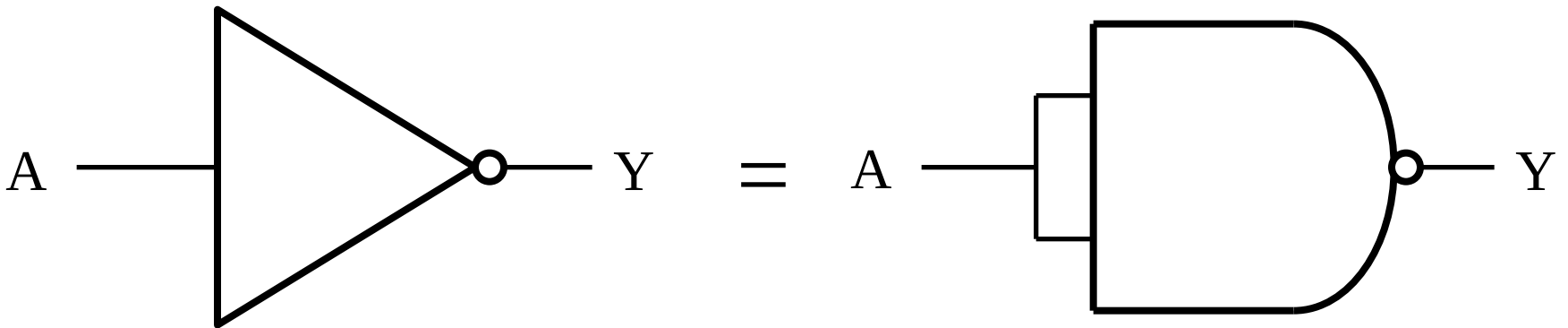
$$(a) \overline{\bar{x}_1 \bar{x}_2} = \bar{x}_1 + \bar{x}_2$$

(DeMorgan (15a))

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

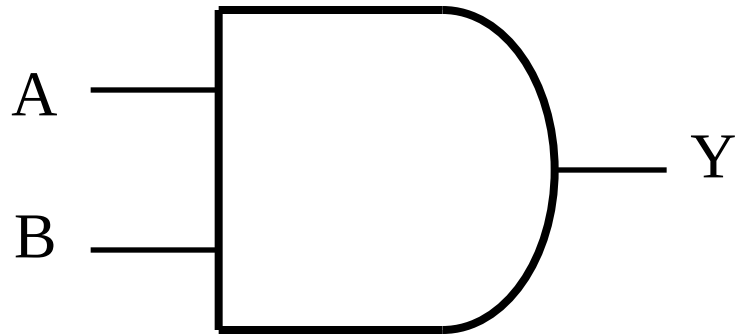
(DeMorgan (15b))

Inverterare med NAND

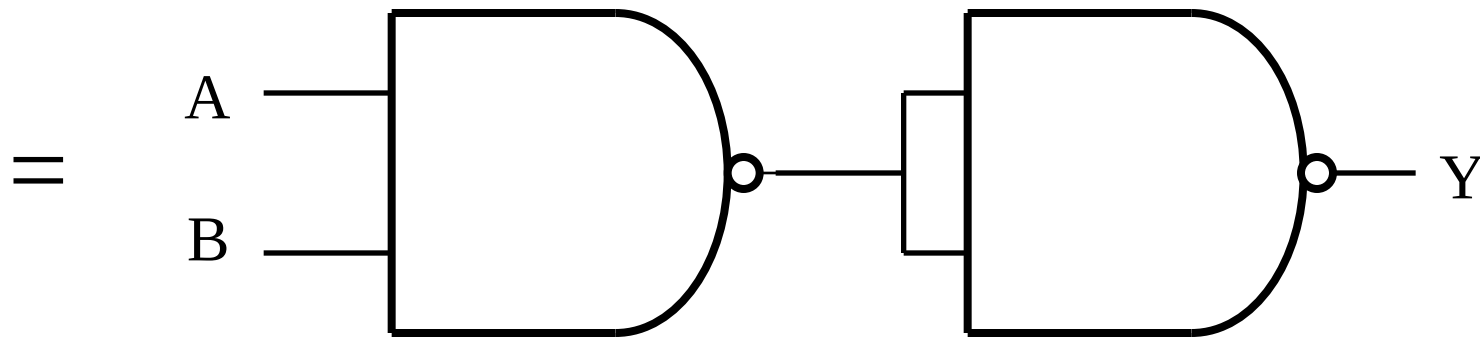


$$Y = \overline{A} = \overline{A \cdot 1}$$

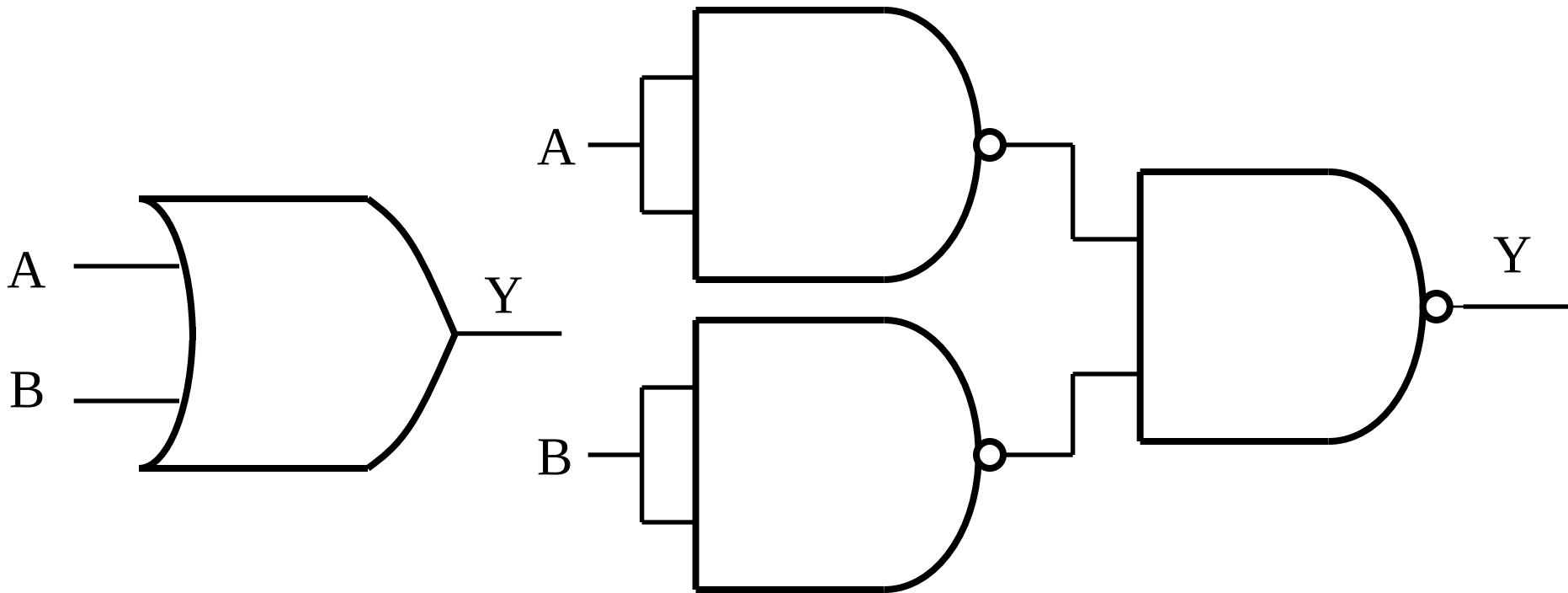
AND-grind med NAND-grindar



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

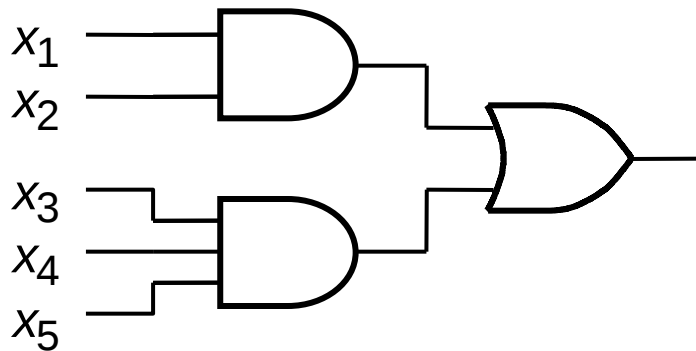


OR-grind med NAND-grindar

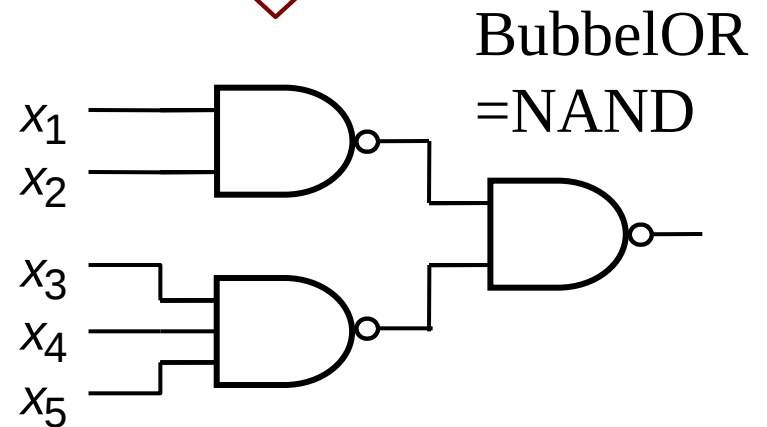
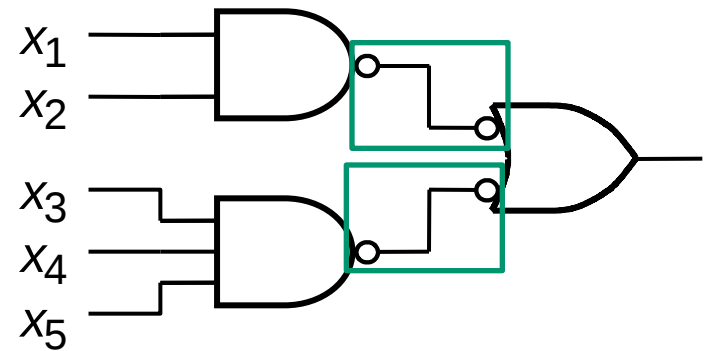


$$Y = A + B = \overline{\overline{A + B}} = \overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A} \cdot \overline{A} \cdot \overline{B} \cdot \overline{B}}$$

Logiska funktioner med bara NAND



AND-OR funktion



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:

$\{\text{AND, OR, NOT}\} \rightarrow (\text{DeMorgan}) \rightarrow \{\text{AND, NOT}\} \rightarrow \{\text{NAND}\}$

$\{\text{OR, AND, NOT}\} \rightarrow (\text{DeMorgan}) \rightarrow \{\text{OR, NOT}\} \rightarrow \{\text{NOR}\}$

Logiska grindar XOR-grinden

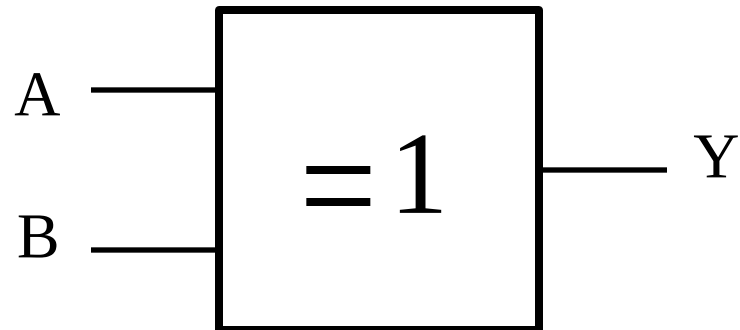
Exklusivt ELLER

IEC Symbol

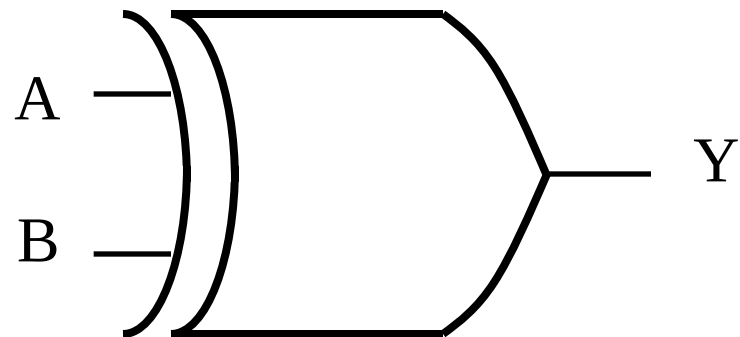
(International Electrotechnical Commission)

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

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



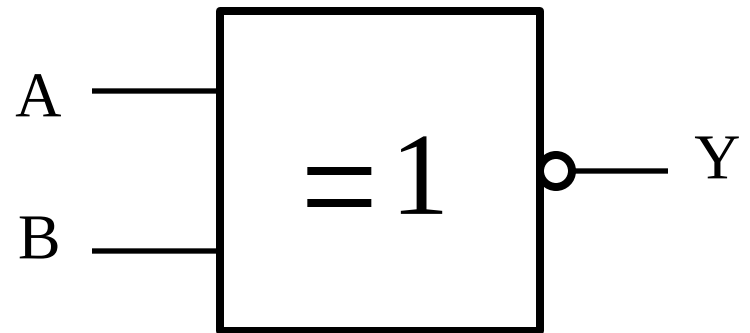
Traditional (American) Symbol



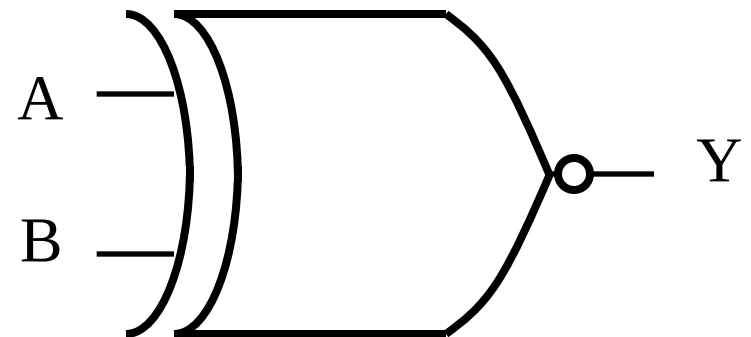
Logiska grindar XNOR-grinden

IEC Symbol
(International Electrotechnical Commission)

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



Traditional (American) Symbol

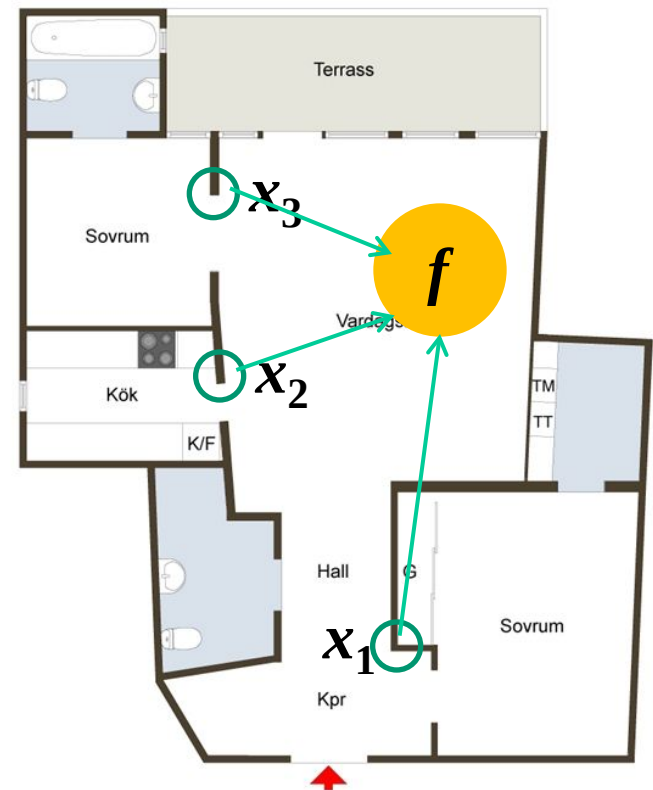
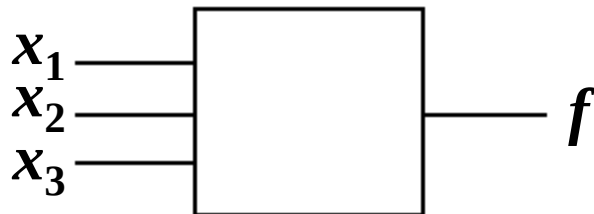


$$Y = \overline{A \oplus B} = \overline{A} \cdot \overline{B} + A \cdot B$$

Exempel: Trevägs ljuskontroll

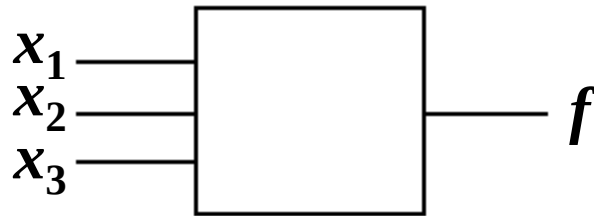
Brown/Vranesic: 2.8.1

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



Viss avvikelse kan förekomma. Skala och mått kan avvika från verkligheten.

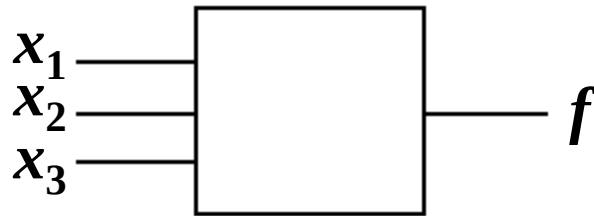
Trevägs ljuskontroll



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	

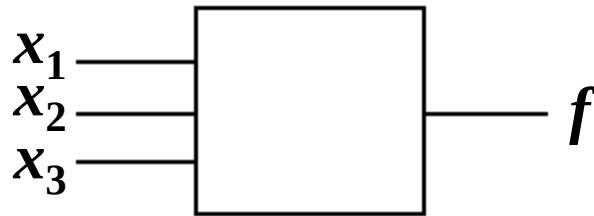
Trevägs ljuskontroll



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	

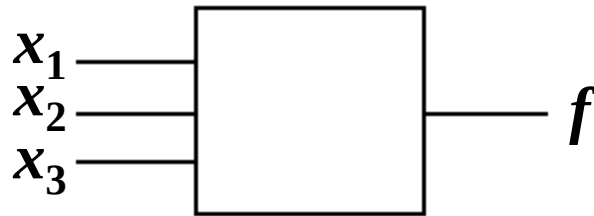
Trevägs ljuskontroll



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	

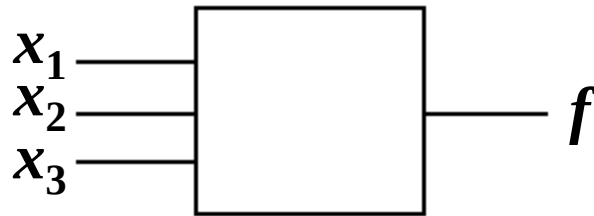
Trevägs ljuskontroll



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



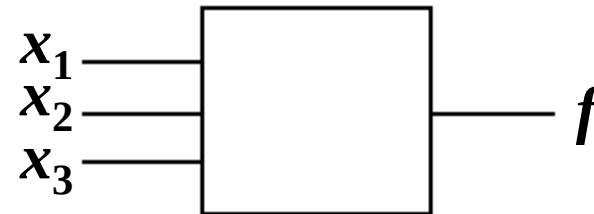
Man ska alltid kunna ändra
ljuset genom att ändra en
valfri strömbrytare.

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



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

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

eller

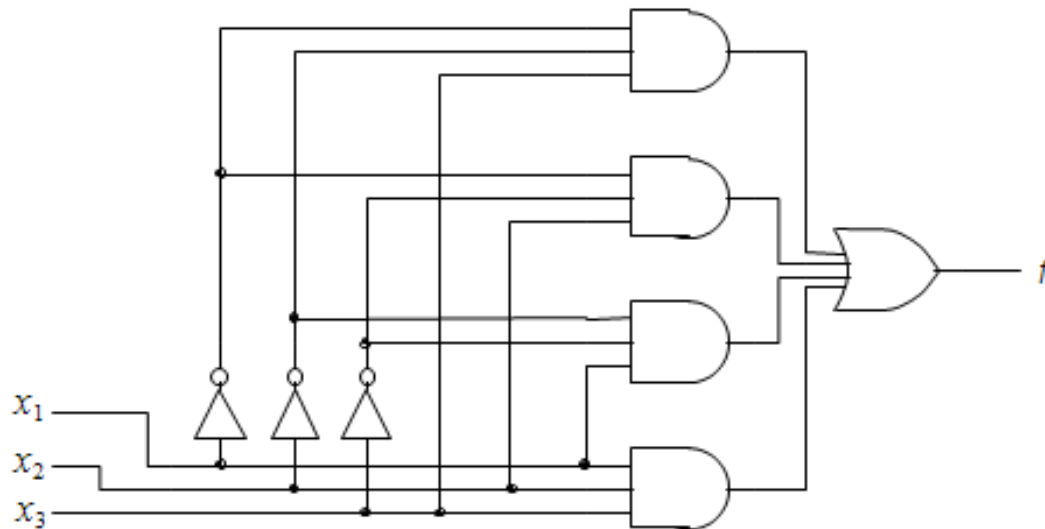
$$f = \prod M(0,3,5,6) =$$

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

Trevägs ljuskontroll



$$f = \sum m(1,2,4,7) = \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2\bar{x}_3 + x_1\bar{x}_2\bar{x}_3 + x_1x_2x_3$$

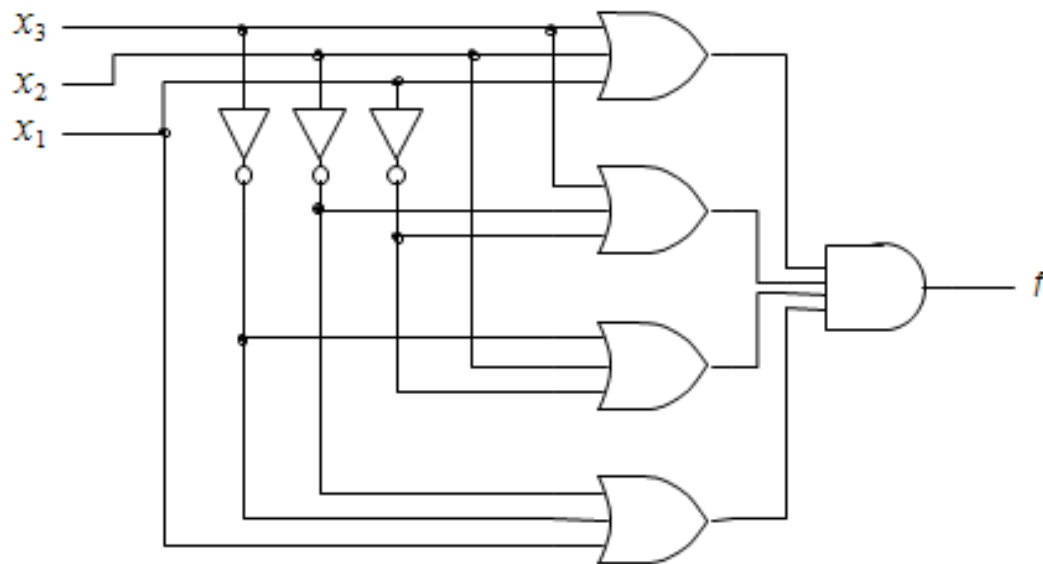


(a) Sum-of-products realization

Trevägs ljuskontroll



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



(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)**

