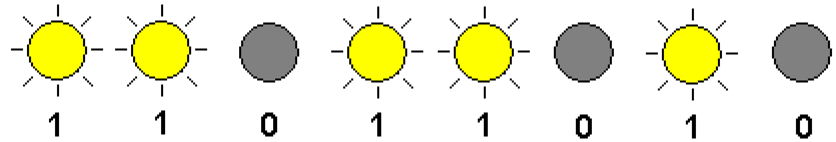
Control with binary code



Row with indicator lamps showing 8-bit number, Byte

Bin → Dec

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

$128+64+0+16+8+0+2+0 = 218$

Hex → Dec

$$\underbrace{1101}_{D}\ \underbrace{1010}_{A}$$

D	A
16^1	16^0
16	1

$13 \cdot 16 + 10 \cdot 1 = 218$

William Sandqvist william@kth.se

Dec – Bin – Hex – Oct

Dec	Bin	Hex	Dec	Bin	Hex
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

11011010

$$\overbrace{1101}^{\mathbf{D}}\;\overbrace{1010}^{\mathbf{A}} \qquad \overbrace{11}^{\mathbf{3}}\;\overbrace{011}^{\mathbf{3}}\;\overbrace{010}^{\mathbf{2}}$$

$218_{10} = 11011010_2 = DA_{16} = 332_8$

Ex 1.1c Decimal to Binary

binary weights: 1024 512 256 128 64 32 16 8 4 2 1

$71_{10} = ?_2$

Ex 1.1c Decimal to Binary

binary weights: 1024 512 256 128 64 32 16 8 4 2 1

$71_{10} = ?_2$

$71_{10} =$
$(64+7= 64+4+2+1) = 1000111_2$

Ex. 1.2a Binary to Decimal

binary weights: 1024 512 256 128 64 32 16 8 4 2 1

$$101101001_2 = ?_{10}$$

William Sandqvist william@kth.se

Ex. 1.2a Binary to Decimal

binary weights: 1024 512 256 128 64 32 16 8 4 2 1

$$101101001_2 = ?_{10}$$

$$101101001_2 =$$
$$(2^8+2^6+2^5+2^3+2^0=256+64+32+8+1)$$
$$=361_{10}$$

William Sandqvist william@kth.se

Ex 1.3c Binary/Octal/Hexadecimal

$$100110101_2 = ?_{16} = ?_8$$

Ex 1.3c Binary/Octal/Hexadecimal

$$100110101_2 = ?_{16} = ?_8$$

$$1\ 0011\ 0101_2 = 1\ 3\ 5_{16}$$

Ex 1.3c Binary/Octal/Hexadecimal

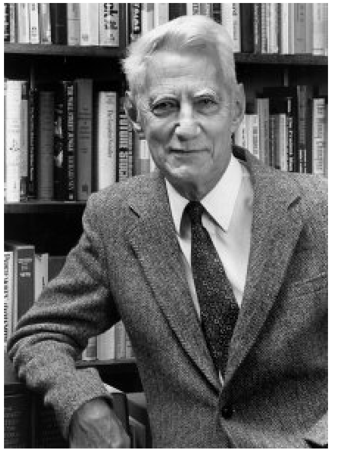
$$100110101_2 = ?_{16} = ?_8$$

$$1\ 0011\ 0101_2 = 1\ 3\ 5_{16}$$

$$100\ 110\ 101_2 = 4\ 6\ 5_8$$

Boolean algebra

George Boole
Mathematician
(1815-1864)

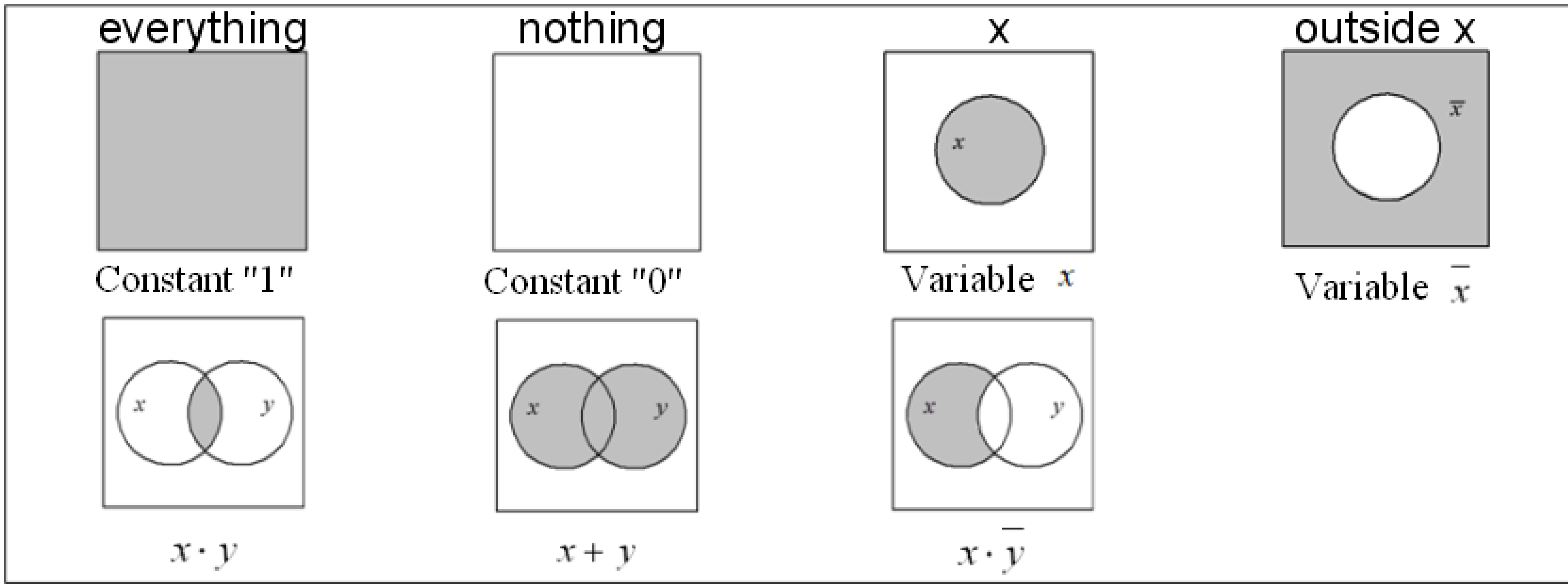


Claude Shannon
Mathematician/Eelectronics
engineer (1916 –2001)

By representing logical expressions in mathematical form, where the joining words OR and AND corresponded to a kind of addition and multiplication, it became possible to investigate with an algebra complicated logical statements and arguments, to determine if in the end they were true or false.

1938 Claude Shannon showed how to use the Boolean algebra to electrical contacts. Since then, Boolean algebra, is the main tool for all digital design.

Venn-diagram



x in common with y x together with y x in common with outside y

Ex. 3.2 De Morgans theorem with Venn diagram

Prove De Morgans theorem with the use of Venn Diagram.

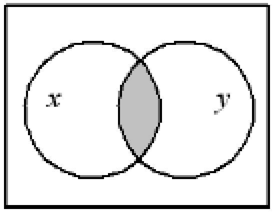
$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

Ex. 3.2 De Morgan

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

Ex. 3.2 De Morgan

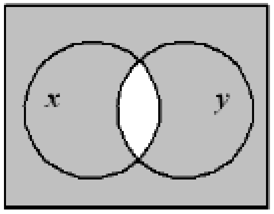
$$\overline{x \cdot y} = \overline{x} + \overline{y}$$



$x \cdot y$

Ex. 3.2 De Morgan

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

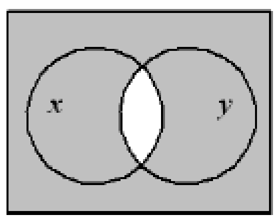


$\overline{x \cdot y}$

=

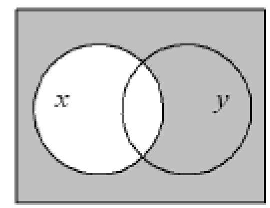
Ex. 3.2 De Morgan

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

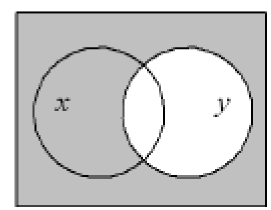


$\overline{x \cdot y}$

=



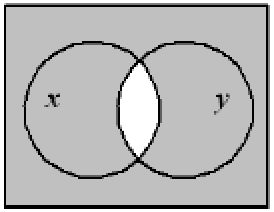
$\overline{x}$



$\overline{y}$

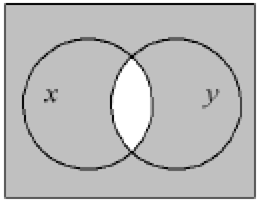
Ex. 3.2 De Morgan

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$



$\overline{x \cdot y}$

=

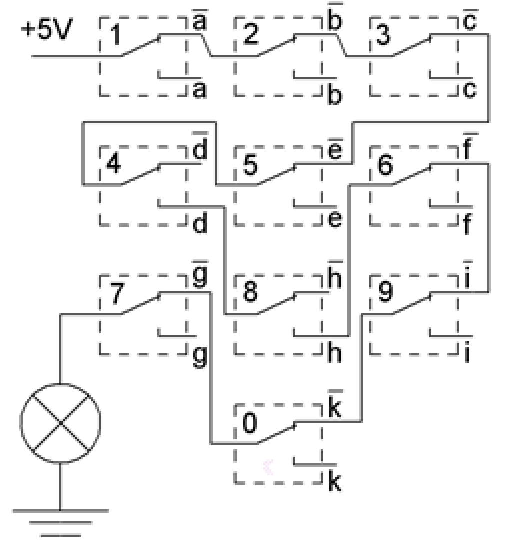


$\overline{x} + \overline{y}$

Now proven!

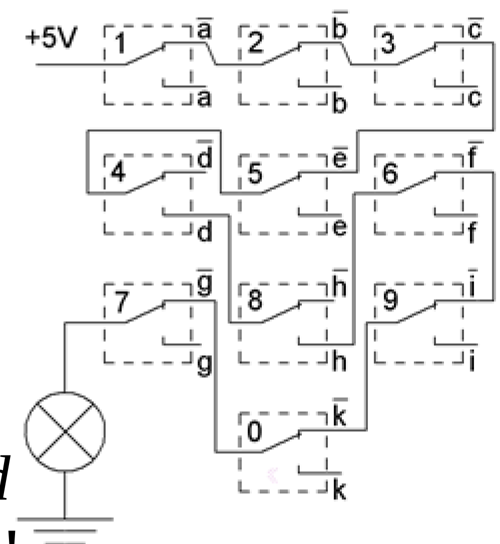
(Ex. 5.1) How to open the code-lock? (=minterm)

Which buttons should be simultaneously pressed in order to light up the lamp?
(= open up the lock)



(Ex. 5.1) How to open the code-lock? (=minterm)

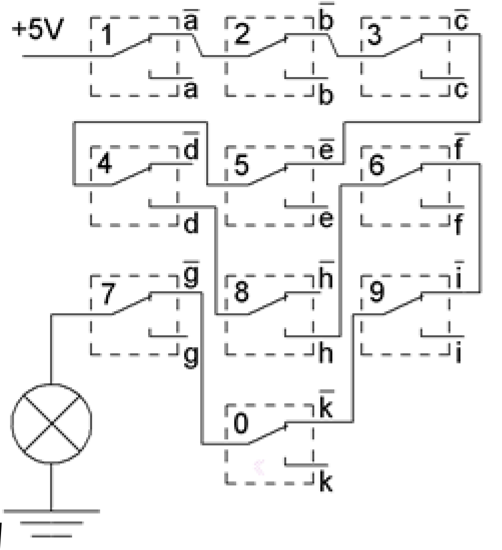
Which buttons should be simultaneously pressed in order to light up the lamp?
(= open up the lock)



Answer: `4,d` and `8,h` but you must simultaneously *avoid* pressing `a b c e f g i` and `k`!

(Ex. 5.1) How to open the code-lock? (=minterm)

Which buttons should be simultaneously pressed in order to light up the lamp?
(= open up the lock)

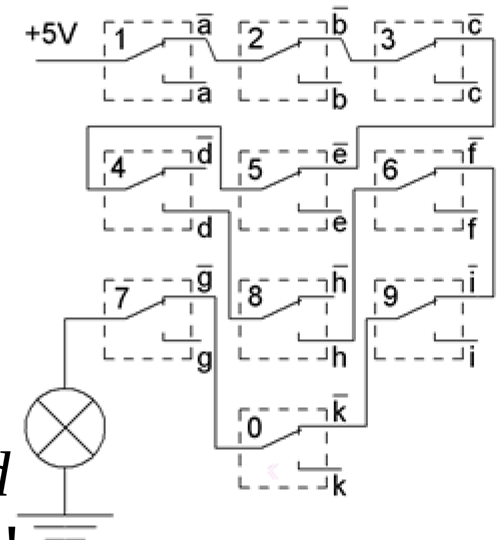


Answer: `4,d` and `8,h` but you must simultaneously *avoid* pressing `a b c e f g i` and `k`!

$$T = \overline{a} \cdot \overline{b} \cdot \overline{c} \cdot d \cdot \overline{e} \cdot \overline{f} \cdot \overline{g} \cdot h \cdot \overline{i} \cdot \overline{k}$$

(Ex. 5.1) How to open the code-lock? (=minterm)

Which buttons should be simultaneously pressed in order to light up the lamp?
(= open up the lock)



Answer: `4,d` and `8,h` but you must simultaneously *avoid* pressing `a b c e f g i` and `k`!

$$T = \bar{a}\cdot\bar{b}\cdot\bar{c}\cdot d\cdot\bar{e}\cdot\bar{f}\cdot\bar{g}\cdot h\cdot\bar{i}\cdot\bar{k}$$

A product-term with *all* variables is called a **minterm**.

Ex 3.3 Venn Diagram

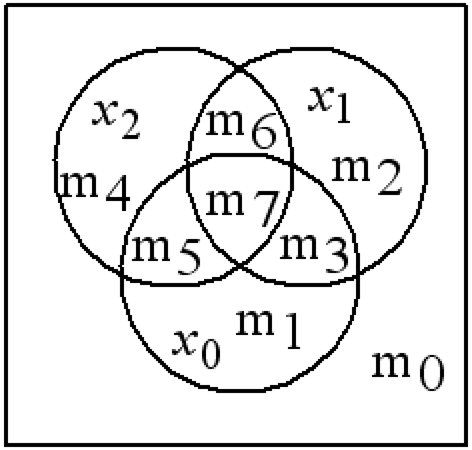
a) Draw a Venn Diagram for three variables and mark all truth table minterms in the diagram.

b) Minimize this function with the help of the Venn Diagram.

$$f = \overline{x}_2\overline{x}_1 x_0 + \overline{x}_2 x_1 x_0 + x_2\overline{x}_1 x_0 + x_2 x_1\overline{x}_0 + x_2 x_1 x_0$$

Ex. 3.3a Truth Table – Venn diagram

x_2	x_1	x_0	
0	0	0	m_0
0	0	1	m_1
0	1	0	m_2
0	1	1	m_3
1	0	0	m_4
1	0	1	m_5
1	1	0	m_6
1	1	1	m_7

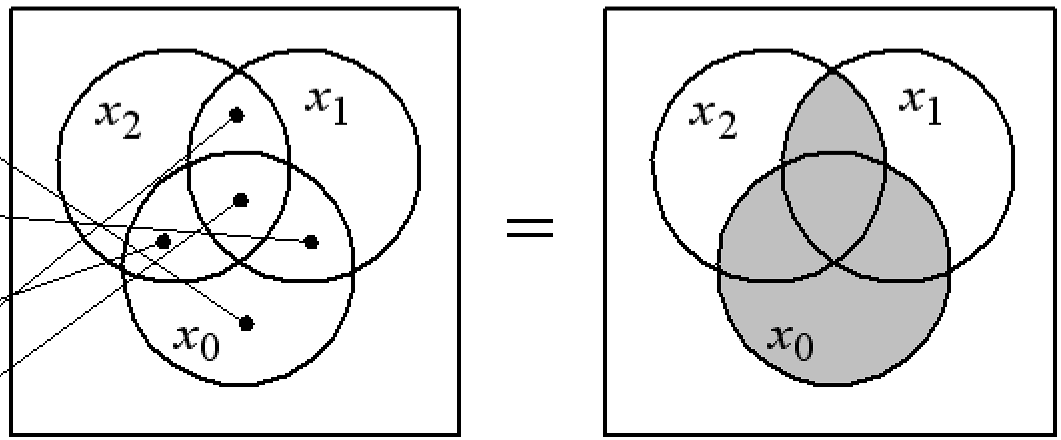


Ex. 3.3b simplified expression

Orginal expression.

$$\boxed{f = \bar{x}_2\bar{x}_1 x_0 + \bar{x}_2 x_1 x_0 + x_2\bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 x_1 x_0}$$

x_2	x_1	x_0		
0	0	0	0	
0	0	1	1	$\bar{x}_2\bar{x}_1 x_0$
0	1	0	0	
0	1	1	1	$\bar{x}_2 x_1 x_0$
1	0	0	0	
1	0	1	1	$x_2\bar{x}_1 x_0$
1	1	0	1	$x_2 x_1 \bar{x}_0$
1	1	1	1	$x_2 x_1 x_0$

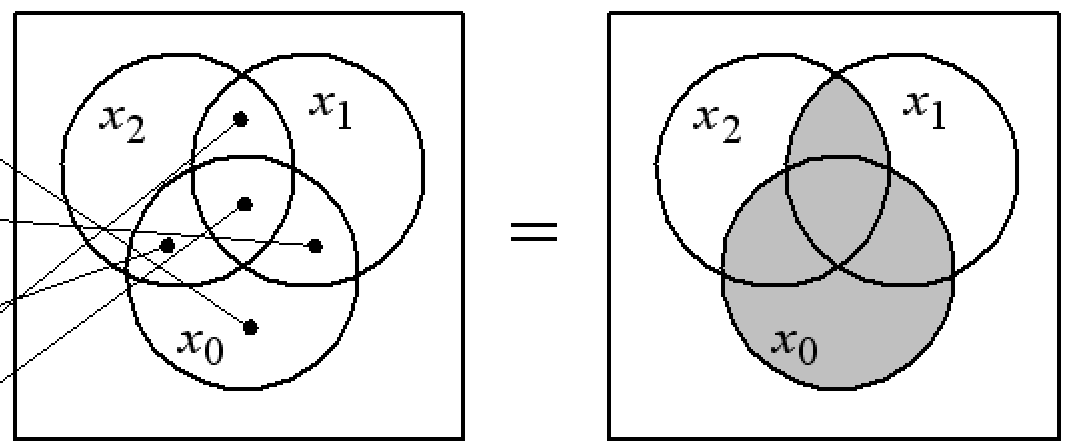


Ex. 3.3b simplified expression

Orginal expression.

$$f = \bar{x}_2\bar{x}_1x_0 + \bar{x}_2x_1x_0 + x_2\bar{x}_1x_0 + x_2x_1\bar{x}_0 + x_2x_1x_0$$

x_2	x_1	x_0		
0	0	0	0	
0	0	1	1	$\bar{x}_2\bar{x}_1x_0$
0	1	0	0	
0	1	1	1	$\bar{x}_2x_1x_0$
1	0	0	0	
1	0	1	1	$x_2\bar{x}_1x_0$
1	1	0	1	$x_2x_1\bar{x}_0$
1	1	1	1	$x_2x_1x_0$



Simplified!

$$f = x_2x_1 + x_0$$

Boole’s algebra rules

Logical addition "+", **OR**, and logical multiplication "×", **AND**, broadly follows the usual normal algebraic distributive, commutative and associative laws (with one exception).

Distributiva lagarna	$A \cdot (B + C) = A \cdot B + A \cdot C$ $A + (B \cdot C) = (A + B) \cdot (A + C)$ *Exception!*
Kommutativa lagarna	$A \cdot B = B \cdot A$ $A + B = B + A$
Associativa lagarna	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$ $(A + B) + C = A + (B + C)$

Theorems

Rules

$A \cdot A = A \qquad A \cdot 0 = 0 \qquad A + 0 = A$

$A + A = A \qquad A \cdot 1 = A \qquad A + 1 = 1$

Some theorems

Absorption	$A + A \cdot B = A$ $A \cdot (A + B) = A$
Consensus	$A \cdot B + \overline{A} \cdot C =$ $A \cdot B + \overline{A} \cdot C + B \cdot C$
de Morgan	$\overline{(A + B)} = \overline{A} \cdot \overline{B}$ $\overline{(A \cdot B)} = \overline{A} + \overline{B}$

Ex. 4.1(a, b, c, h) Boolean algebra

4.1

a) $f = a \cdot \overline{c} \cdot d + a \cdot d$

b) $f = a \cdot (\overline{b} + \overline{a} \cdot c + a \cdot b)$

c) $f = a + \overline{b} + \overline{a} \cdot b + \overline{c}$

d) $f = (a + b \cdot \overline{c}) \cdot (\overline{a} \cdot \overline{b} + c)$

e) $f = (a + \overline{b}) \cdot (\overline{a} + b) \cdot (a + b)$

f) $f = \overline{a} \cdot \overline{b} \cdot c + a \cdot b \cdot c + \overline{a} \cdot b \cdot c$

g) $f = \overline{a} \cdot b \cdot \overline{c} + \overline{a} \cdot b \cdot \overline{d} + c \cdot d$

h) $f = a + (\overline{\overline{a}} \cdot \overline{\overline{b}})$

i) $f = \overline{\overline{\overline{a} + \overline{\overline{a}} \cdot b + c}}$

Ex. 4.1a

$$f = a \cdot \bar{c} \cdot d + a \cdot d \quad = \{ \mathit{factor}\ \mathrm{ad} \} = a \cdot d \cdot (\bar{c} + 1) = a \cdot d$$

Ex. 4.1b

$$f = a \cdot (\overline{b} + \overline{a} \cdot c + a \cdot b) \quad = a \cdot \overline{b} + a \cdot \overline{a} \cdot c + a \cdot a \cdot b =$$

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

Ex. 4.1**c**

$$f = a + \overline{b} + \overline{a} \cdot b + \overline{c} \quad =$$

Ex. 4.1c

$$f = a + \boxed{\overline{b}} + \overline{a} \cdot b + \overline{c} \quad = a + \boxed{(a + \overline{a}) \cdot \overline{b}} + \overline{a} \cdot b + \overline{c} =$$

$$= a + \boxed{a \cdot \overline{b} + \overline{a} \cdot \overline{b}} + \overline{a} \cdot b + \overline{c} =$$

$$= a + a \cdot \overline{b} + \overline{a} \cdot (\overline{b} + b) + \overline{c} =$$

$$= \ldots \; a + \overline{a} \ldots = 1$$

Ex. 4.1h

$$f = a + (\overline{\overline{a} \cdot \overline{b}}) \quad =$$

Ex. 4.1**h**

$$f = a + (\overline{\overline{a} \cdot \overline{b}}) \quad = \{deMorgan\} = a + \overline{\overline{a}} + \overline{\overline{b}} = a + b$$

Ex. 4.4 De Morgan

4.4

$$\overline{(a+b+\bar{c})(a+\bar{b}+\bar{c})(\bar{a}+\overline{\bar{b}c}+b\bar{c})}$$

Ex. 4.4

$$\overline{(a+b+\bar{c})(a+\bar{b}+\bar{c})(\bar{a}+\boxed{\overline{\bar{b}c}}+b\bar{c})} =$$

$$= \overline{(a+b+\bar{c})(a+\bar{b}+\bar{c})(\bar{a}+b+\bar{c}+b\bar{c})} =$$

$$= \overline{(a+b+\bar{c})(a+\bar{b}+\bar{c})(\bar{a}+b+\bar{c})} =$$

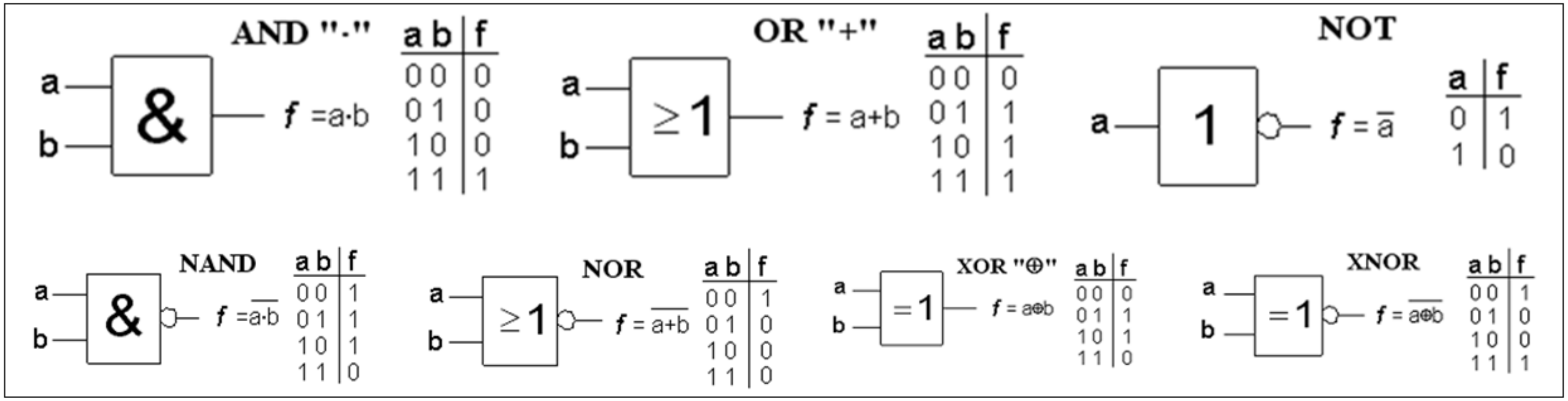
$$\overline{(a+b+\bar{c})}+\overline{(a+\bar{b}+\bar{c})}+\overline{(\bar{a}+b+\bar{c})} =$$

Duplicate!

$$= \boxed{\bar{a}\bar{b}c}+\bar{a}bc+a\bar{b}c = \boxed{\bar{a}\bar{b}c}+\boxed{\bar{a}\bar{b}c}+\bar{a}bc+a\bar{b}c =$$

$$= \bar{a}c(\bar{b}+b)+\bar{b}c(a+\bar{a}) = \bar{a}c+\bar{b}c$$

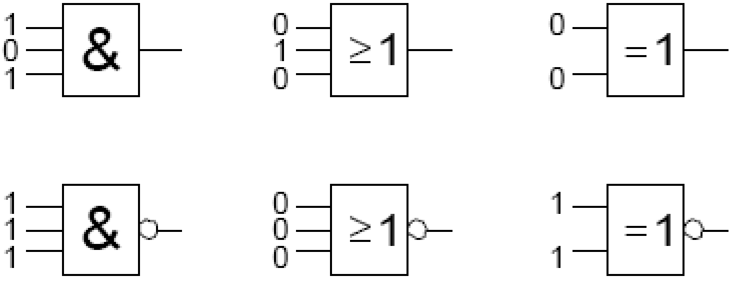
Logic gates



William Sandqvist william@kth.se

(Ex. 4.5a) Gates

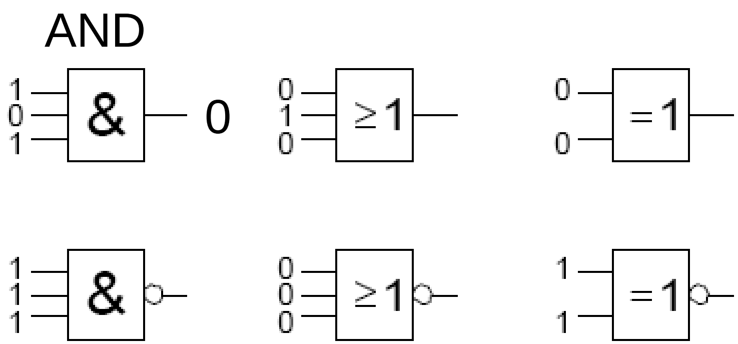
Enter the name and output 1/0 for the following six gate types when the input signals are as shown in the figure.



William Sandqvist william@kth.se

(Ex. 4.5a) Gates

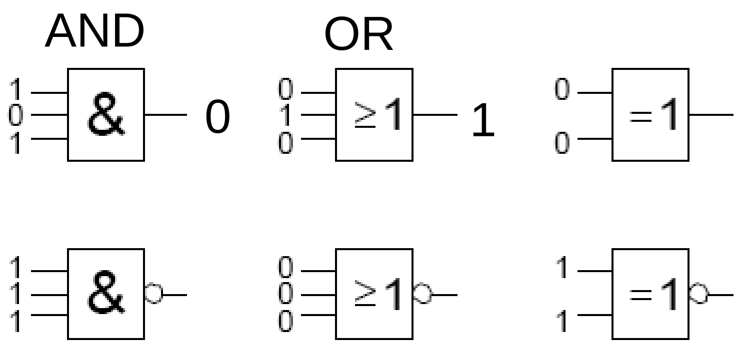
Enter the name and output 1/0 for the following six gate types when the input signals are as shown in the figure.



William Sandqvist william@kth.se

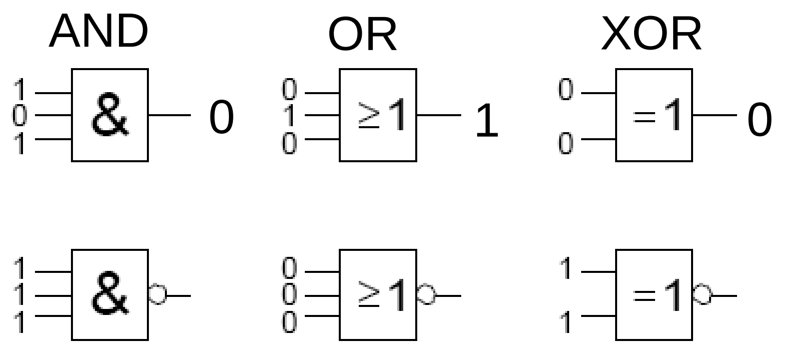
(Ex. 4.5a) Gates

Enter the name and output 1/0 for the following six gate types when the input signals are as shown in the figure.



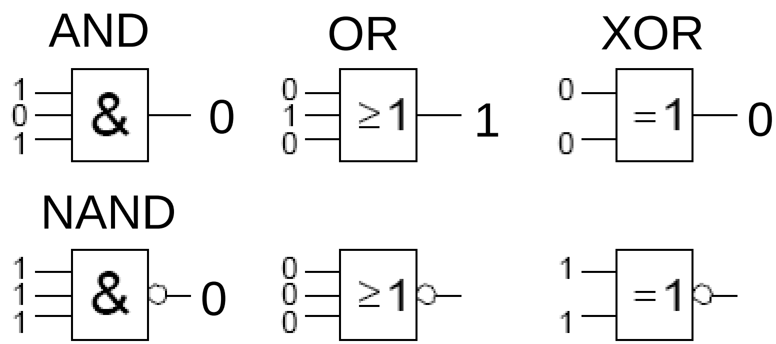
(Ex. 4.5a) Gates

Enter the name and output 1/0 for the following six gate types when the input signals are as shown in the figure.



(Ex. 4.5a) Gates

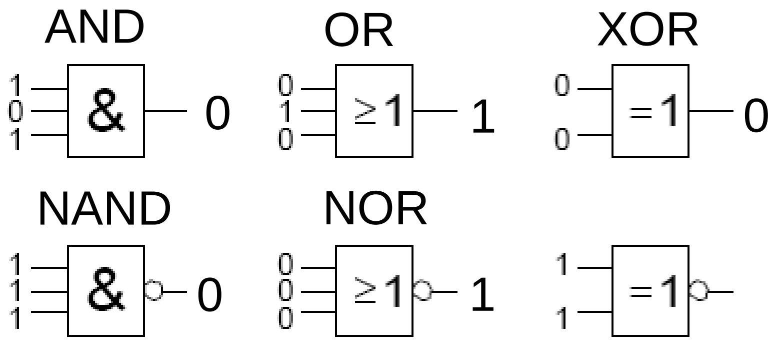
Enter the name and output 1/0 for the following six gate types when the input signals are as shown in the figure.



William Sandqvist william@kth.se

(Ex. 4.5a) Gates

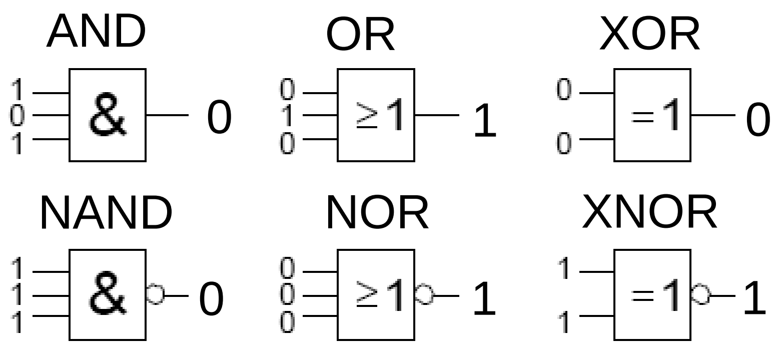
Enter the name and output 1/0 for the following six gate types when the input signals are as shown in the figure.



William Sandqvist william@kth.se

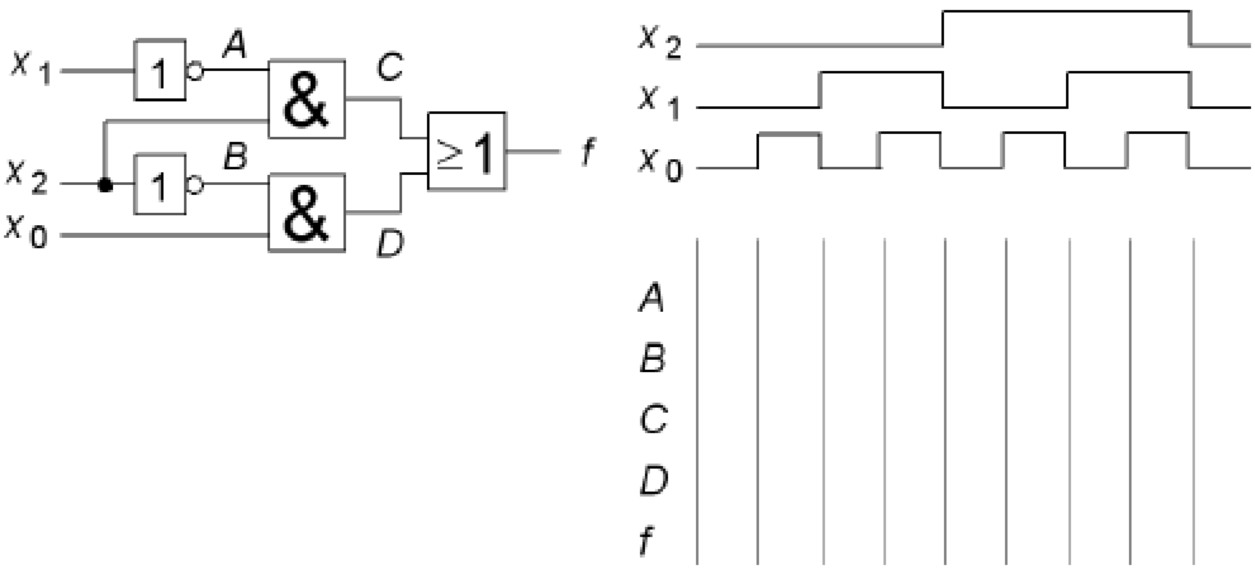
(Ex. 4.5a) Gates

Enter the name and output 1/0 for the following six gate types when the input signals are as shown in the figure.

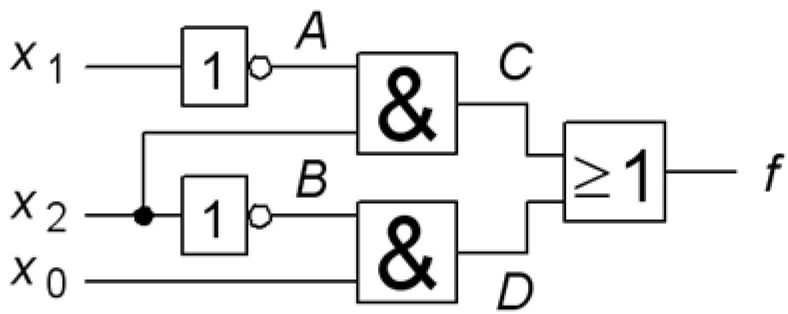


William Sandqvist william@kth.se

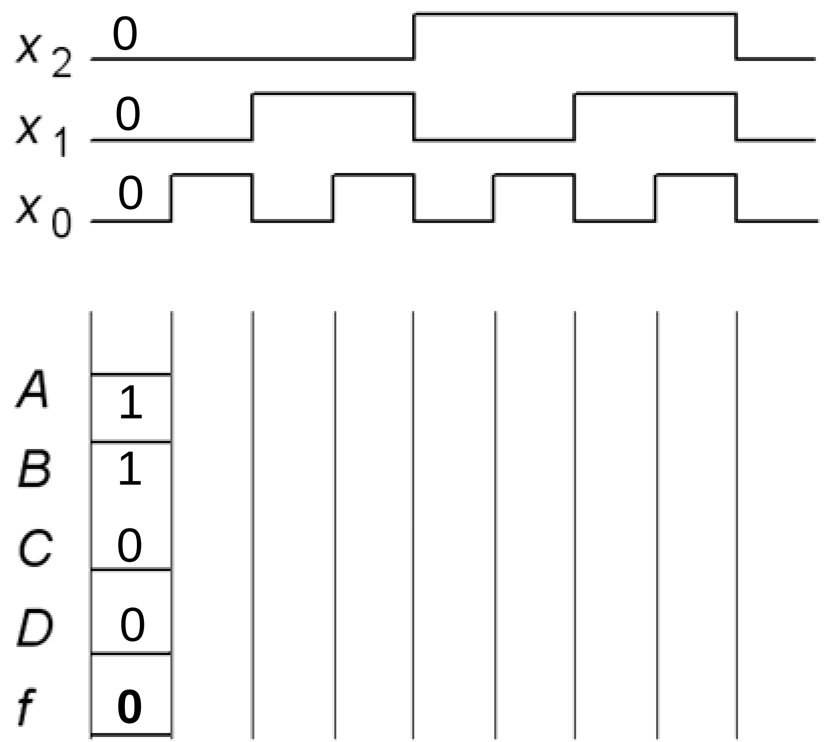
Ex. 4.7 Timing diagram and Truth Table



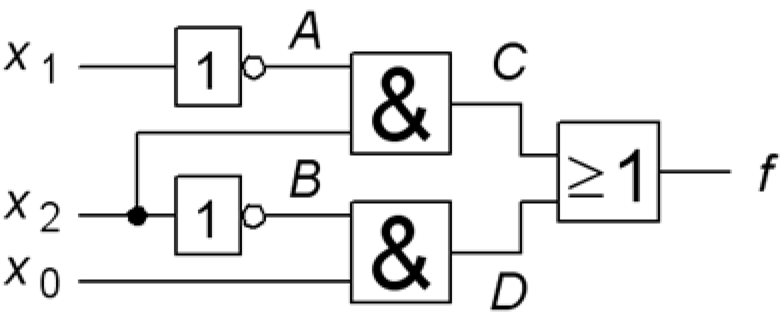
Ex. 4.7

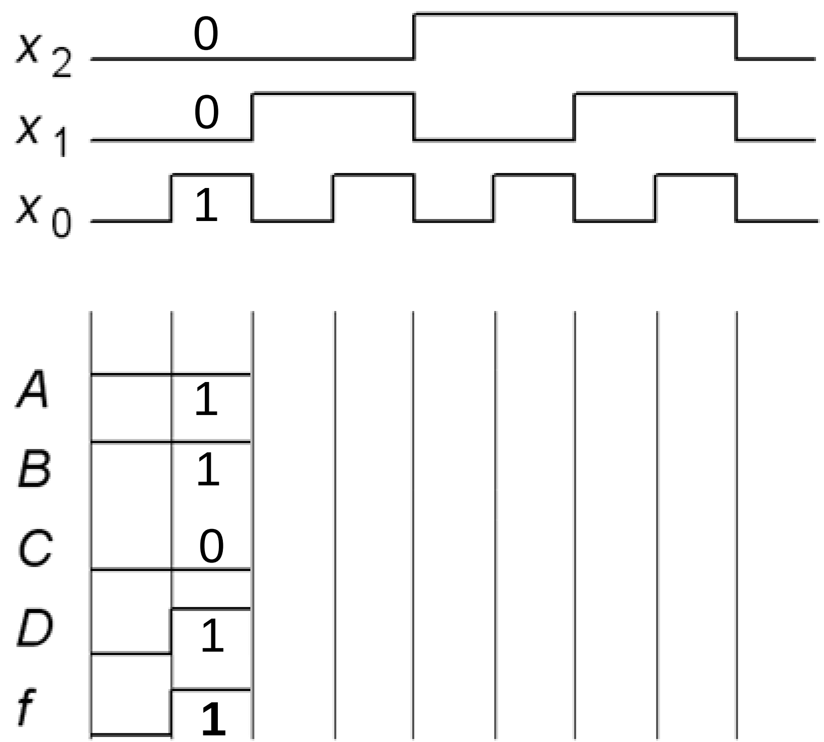


x_2	x_1	x_0	f
0	0	0	0
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



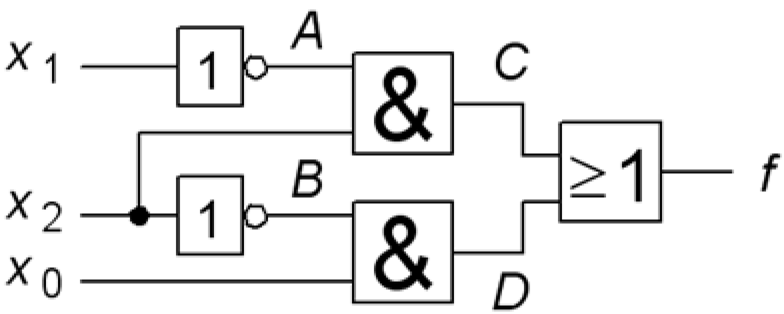
Ex. 4.7

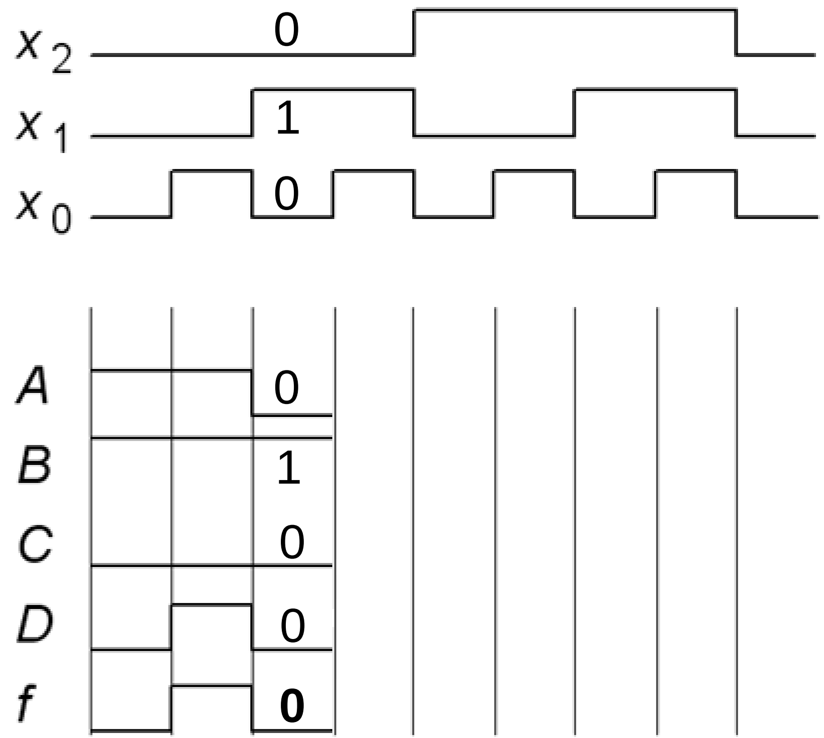




x_2	x_1	x_0	f
0	0	0	0
0	0	1	1
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Ex. 4.7

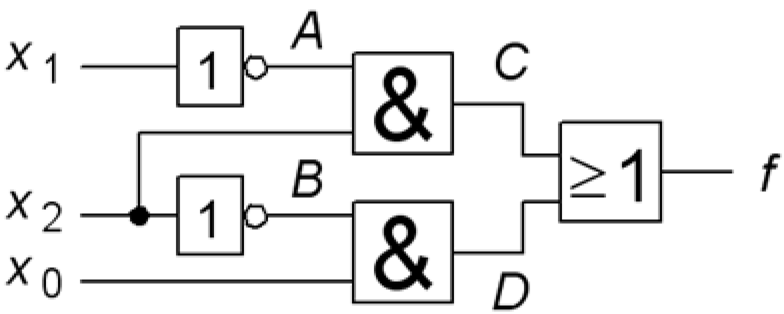




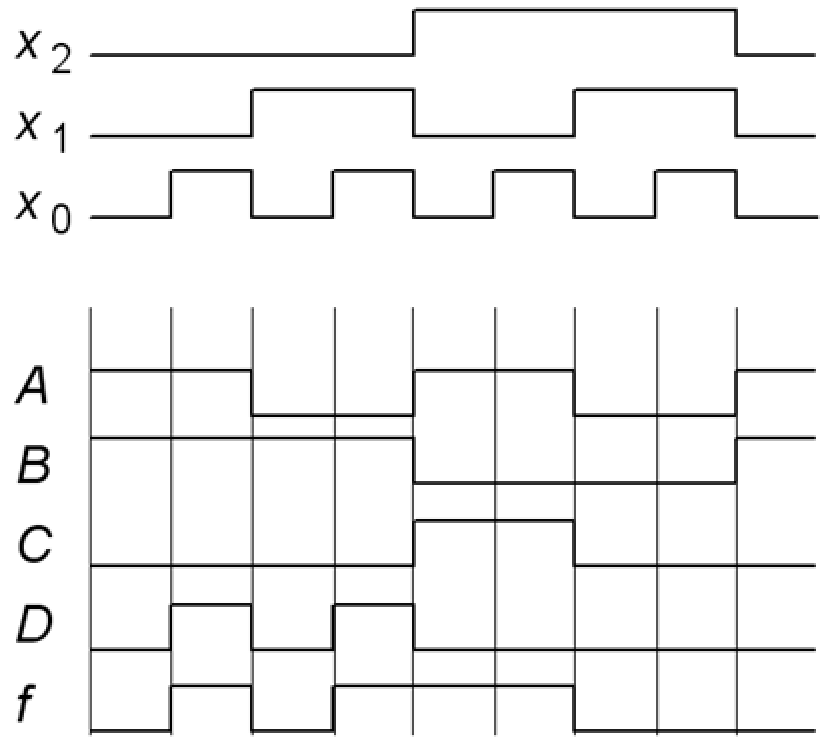
x_2	x_1	x_0	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

William Sandqvist william@kth.se

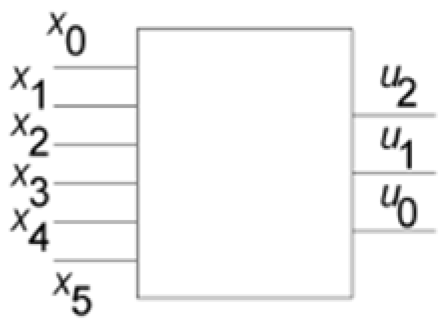
Ex. 4.7



x_2	x_1	x_0	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0



Ex. 4.12 From text to Boolean equations



A combinatorical circuit with six input signals x_5, x_4, x_3, x_2, x_1 and three output signals u_2, u_1, u_0, is described in this way:

- $u_0 = 1$ if and only if ”either both x_0 and x_2 are 0 or x_4 and x_5 are different”
- $u_1 = 1$ if and only if ”x_0 and x_1 are equal and x_5 is the inverse of x_2”
- $u_2 = 0$ if and only if ”x_0 is 1 and some of x_1 … x_5 is 0”

ÖH 4.12

$u_0 = 1$ if and only if

AND

” either both x_0 and x_2 are 0 NOT

XOR

or x_4 and x_5 are different”

XOR

$$u_0 = \overline{x}_0 \cdot \overline{x}_2 \oplus (x_4 \oplus x_5)$$

ÖH 4.12

$u_1 = 1$ if and only if

” x_0 and x_1 are equal $\boxed{\text{and}}$ x_5 is the inverse of x_2 ”

XNOR **AND** **XOR**

$$u_1 = \overline{x_0 \oplus x_1} \cdot (x_5 \oplus x_2)$$

$$= (x_0 x_1 + \bar{x}_0 \bar{x}_1) \cdot (x_5 \bar{x}_2 + \bar{x}_5 x_2)$$

ÖH 4.12

NOT

$\boxed{u_2 = 0}$ if and only if

” x_0 is 1 and $\boxed{\text{some of } x_1 \ldots x_5}$ is 0 ”

AND **OR** **NOT**

$$\overline{u}_2 = x_0 \cdot (\overline{x}_1 + \overline{x}_2 + \overline{x}_3 + \overline{x}_4 + \overline{x}_5)$$

$$\Rightarrow \quad u_2 = \overline{x_0 \cdot (\overline{x}_1 + \overline{x}_2 + \overline{x}_3 + \overline{x}_4 + \overline{x}_5)} =$$

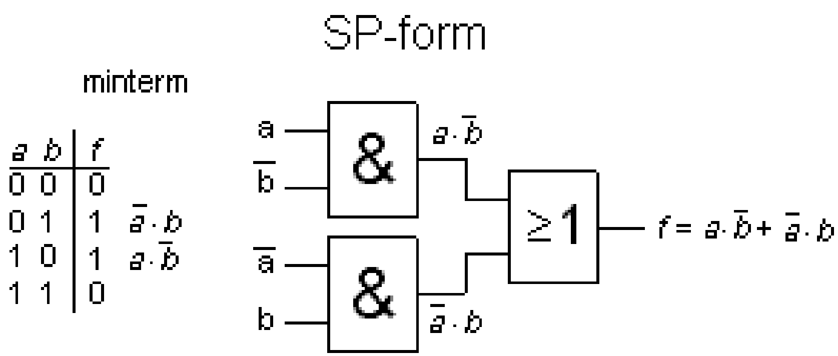
$$= \overline{x}_0 + \overline{(\overline{x}_1 + \overline{x}_2 + \overline{x}_3 + \overline{x}_4 + \overline{x}_5)} =$$

$$= \overline{x}_0 + x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5$$

Logic circuits of SoP-form

All logical functions can be realized by using gate types AND and OR combined in two steps. Here we assume that the input variables are also available in inverted form. If not, then you of course use inverters NOT.

AND-OR logic, SoP-form



One can realize the gate circuit directly from the truth table. Each "1" in the table is a minterm. The function is the sum of these minterms. One says that the function is expressed in the SoP form (Sum of Products).

However, there may exist a simpler circuit with fewer gates that does the same job.

Ex. 5.2 SoP and PoS standard format

5.2

A logic function has this Truth Table:

$a\ b\ c$	f
0 0 0	1
0 0 1	0
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	0
1 1 1	1

Write the function in SoP normal form:

$f(a, b, c) =$

Write the function in PoS normal form:

$f(a, b, c) =$

Ex. 5.2 SoP-form

A logical function has the following truth table. Specify the function of SoP-normal form (sum of products).

a	b	c	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Ex. 5.2 SoP-form

A logical function has the following truth table. Specify the function of SoP-normal form (sum of products).

a b c	f	
0 0 0	1	$\bar{a}\bar{b}\bar{c}$
0 0 1	0	
0 1 0	0	
0 1 1	0	
1 0 0	1	$a\bar{b}\bar{c}$
1 0 1	1	$a\bar{b}c$
1 1 0	0	
1 1 1	1	abc

Ex. 5.2 SoP-form

A logical function has the following truth table. Specify the function of SoP-normal form (sum of products).

a	b	c	f	
0	0	0	1	$\bar{a}\bar{b}\bar{c}$
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	1	$a\bar{b}\bar{c}$
1	0	1	1	$a\bar{b}c$
1	1	0	0	
1	1	1	1	abc

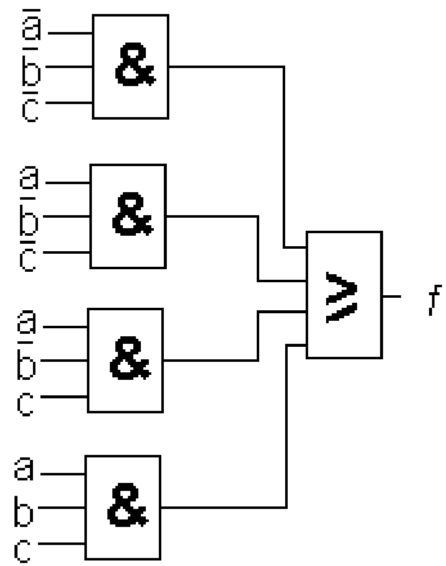
$$f = \bar{a}\cdot\bar{b}\cdot\bar{c} + a\cdot\bar{b}\cdot\bar{c} + a\cdot\bar{b}\cdot c + a\cdot b\cdot c$$

Ex. 5.2 SoP-form

A logical function has the following truth table. Specify the function of SoP-normal form (sum of products).

a	b	c	f	
0	0	0	1	$\bar{a}\bar{b}\bar{c}$
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	1	$a\bar{b}\bar{c}$
1	0	1	1	$a\bar{b}c$
1	1	0	0	
1	1	1	1	abc

$$f = \bar{a}\cdot\bar{b}\cdot\bar{c} + a\cdot\bar{b}\cdot\bar{c} + a\cdot\bar{b}\cdot c + a\cdot b\cdot c$$

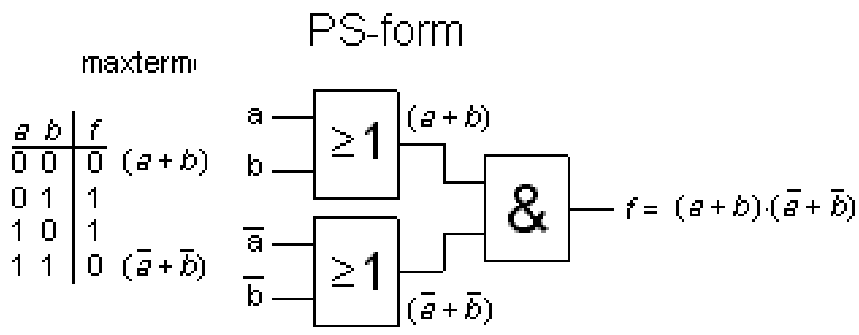


William Sandqvist william@kth.se

Logik circuits of PoS-form

OR-AND logic, PoS form

Alternatively, one can focus on the truth table 0s. If a gate circuit reproduces the function 0's correct then of course the 1's are right too!



Thus, if the function is to be "0" for a particular variable combination (a, b) for example (0, 0) one forms the sum (a + b). This sum could only be "0" for the combination (0, 0).

Such a sum is called a ***maxterm***. The function is expressed as a product of all such maxterms. Each maxterm contributes with a 0 from the truth-table. The function is said to be expressed in the ***PoS form*** *(Product of Sums)*.

Ex. 5.2 PoS-form

A logical function has the following truth table. Specify the function of PoS-normal form (product of sums).

a	b	c	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Ex. 5.2 PoS-form

A logical function has the following truth table. Specify the function of PoS-normal form (product of sums).

a	b	c	f	
0	0	0	1	
0	0	1	0	$(a+b+\bar{c})$
0	1	0	0	$(a+\bar{b}+c)$
0	1	1	0	$(a+\bar{b}+\bar{c})$
1	0	0	1	
1	0	1	1	
1	1	0	0	$(\bar{a}+\bar{b}+c)$
1	1	1	1	

Ex. 5.2 PoS-form

A logical function has the following truth table. Specify the function of PoS-normal form (product of sums).

$$f = (a+b+\overline{c})\cdot(a+\overline{b}+c)\cdot(a+\overline{b}+\overline{c})\cdot(\overline{a}+\overline{b}+c)$$

a	b	c	f	
0	0	0	1	
0	0	1	0	$(a+b+\overline{c})$
0	1	0	0	$(a+\overline{b}+c)$
0	1	1	0	$(a+\overline{b}+\overline{c})$
1	0	0	1	
1	0	1	1	
1	1	0	0	$(\overline{a}+\overline{b}+c)$
1	1	1	1	

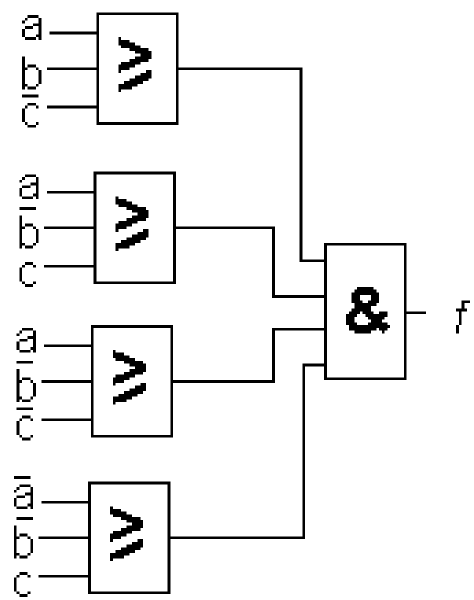
William Sandqvist william@kth.se

Ex. 5.2 PoS-form

A logical function has the following truth table. Specify the function of PoS-normal form (product of sums).

$$f = (a+b+\overline{c})\cdot(a+\overline{b}+c)\cdot(a+\overline{b}+\overline{c})\cdot(\overline{a}+\overline{b}+c)$$

a	b	c	f	
0	0	0	1	
0	0	1	0	$(a+b+\overline{c})$
0	1	0	0	$(a+\overline{b}+c)$
0	1	1	0	$(a+\overline{b}+\overline{c})$
1	0	0	1	
1	0	1	1	
1	1	0	0	$(\overline{a}+\overline{b}+c)$
1	1	1	1	

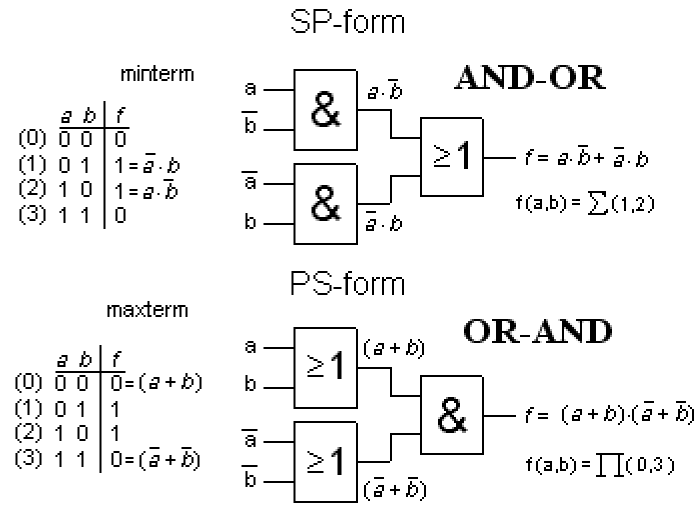


$\sum$ and Π

The SoP and PoS-formats are usually simplified by writing the min- and maxterms serial numbers:

$f(a,b) = \sum m(1,2)$

$f(a,b) = \Pi M(0,3)$



Ex. 5.3 SoP and PoS -form

A minimized function is given on SoP form (Sum of Products).
Specify this function with minterms on SoP normal form, and with maxterms on PoS (Product of Sums) normal form.

$$f(x, y, z) = x\overline{y} + y\overline{z} + \overline{x}z$$

Ex. 5.3

$$f(x,y,z) = x\overline{y} + y\overline{z} + \overline{x}z$$

$$= x\overline{y}(z+\overline{z}) + (x+\overline{x})y\overline{z} + \overline{x}(y+\overline{y})z =$$

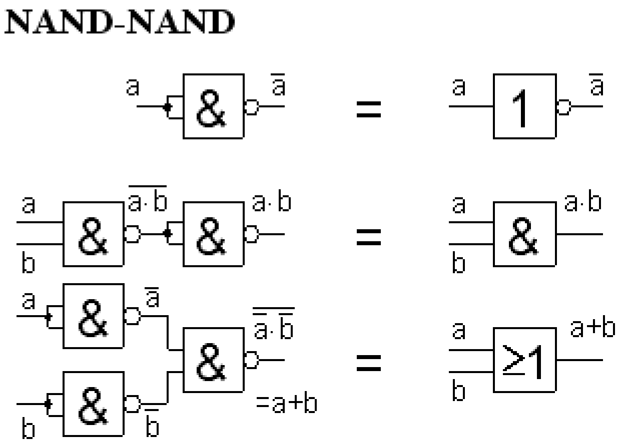
$$= x\overline{y}z + x\overline{y}\,\overline{z} + xy\overline{z} + \overline{x}y\overline{z} + \overline{x}yz + \overline{x}\,\overline{y}z$$

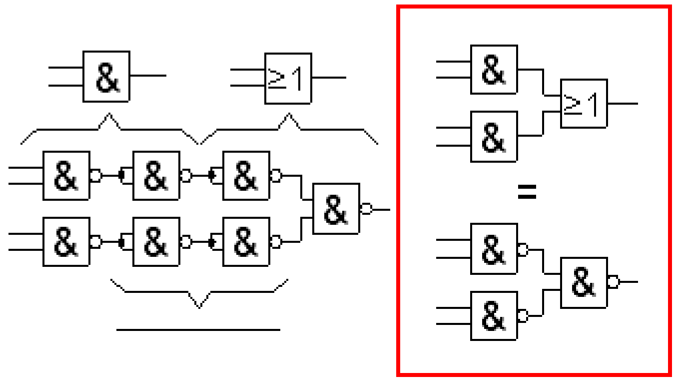
$$\Rightarrow \quad f(x,y,z) = \sum m(001, 010, 011, 100, 101, 110) = \sum m(1,2,3,4,5,6)$$

$$\Rightarrow \quad f(x,y,z) = \prod M(0,7) = (x+y+z)(\overline{x}+\overline{y}+\overline{z})$$

complete logic NAND-NAND

OR, AND and NOT can be made with NAND gates. For logic functions in the SoP format, you can directly replace the AND-OR circuits with NAND-NAND circuits.



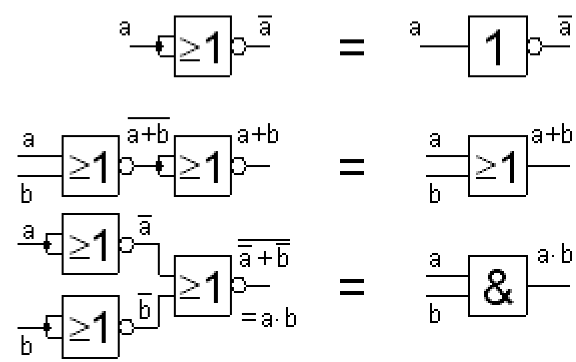


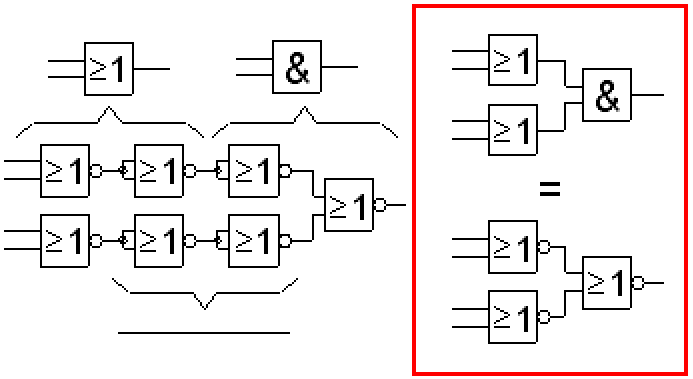
The cost in number of gates will be the same!

complete logic NOR-NOR

OR, AND and NOT can also be made using NOR gates. For logic functions in the PoS form, you can replace the OR-AND circuit with NOR-NOR "straight off".

NOR-NOR





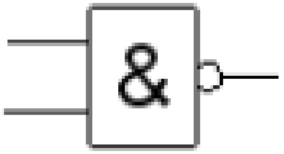
The cost, the number of gates, will be the same!

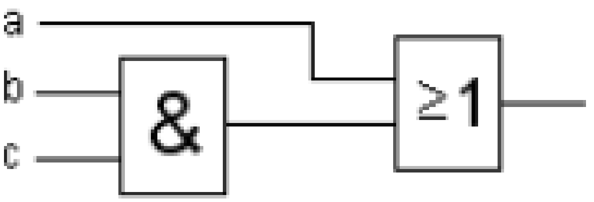
William Sandqvist william@kth.se

Ex. 5.5 NAND-gates

5.5

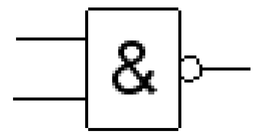
Change to NAND gates!

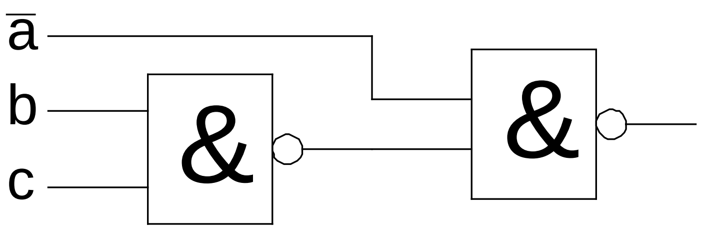




Ex. 5.5

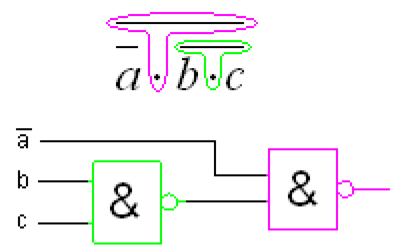
Change to
NAND gates





Algebraically:

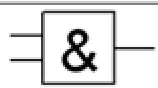
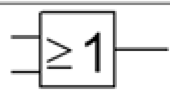
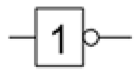
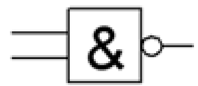
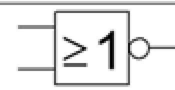
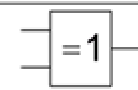
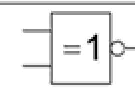
$$a+b\cdot c=\overline{\overline{a+b\cdot c}}=\overline{\overline{a}\cdot\overline{b\cdot c}}$$



William Sandqvist william@kth.se

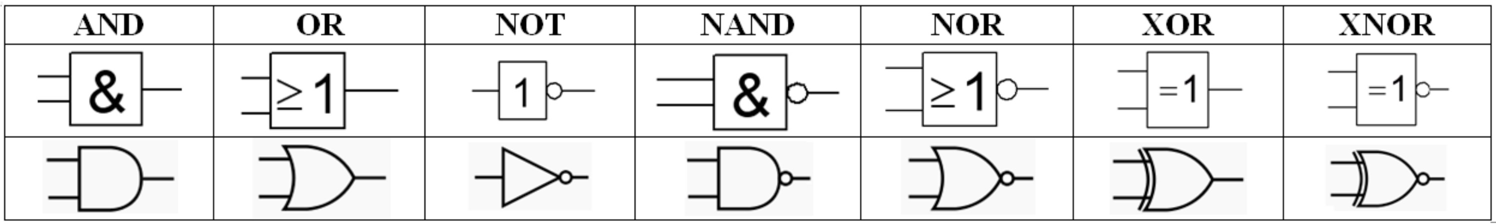
(Ex. 4.11) European and American Symbols

Try out yourself …

(Ex. 4.11) European and American Symbols

Try out yourself …



William Sandqvist william@kth.se