ESc201 : Introduction to Electronics

Digital Circuits-2

Dr. Y. S. Chauhan
Dept. of Electrical Engineering
IIT Kanpur

Goal of Simplification

In the SOP expression:

1. Minimize number of product terms

2. Minimize number of literals in each term

$$\text{Simplification} \;\Rightarrow\; \text{Minimization}$$

Minimization

$$y = \overline{x_1}.\overline{x_2}.x_3 + \overline{x_1}.x_2.x_3 + x_1.\overline{x_2}.x_3 + x_1.x_2.x_3$$

$$y = \overline{x_1}.x_3.(\overline{x_2} + x_2) + x_1.x_3.(\overline{x_2} + x_2)$$

$$y = \overline{x_1}.x_3 + x_1.x_3$$

$$y = (\overline{x_1} + x_1).x_3$$

$$y = x_3$$

Principle used: $x + \overline{x} = 1$

$$f = \overline{x}.\overline{y} + \overline{x}.y + x.\overline{y}$$

Apply the Principle: $x + \overline{x} = 1$ to simplify

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

$$f = \overline{x} + x.\overline{y}$$

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

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

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

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

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

Principle: $x + \overline{x} = 1$ and $x + x = x$

Need a systematic and simpler method

Karnaugh Map (K map) is a popular technique for carrying out simplification

It represents the information in problem in such a way that the two principles become easy to apply

Principle: $x + \overline{x} = 1$ and $x + x = x$

K-map representation of truth table

x	y	min term
0	0	$\overline{x} \cdot \overline{y}$ m0
0	1	$\overline{x} \cdot y$ m1
1	0	$x \cdot \overline{y}$ m2
1	1	$x \cdot y$ m3

x \ y	0	1
0	m_0	m_1
1	m_2	m_3

x	y	f_1
0	0	0
0	1	1
1	0	1
1	1	0

⇨

x \ y	0	1
0	0	1
1	1	0

$$f_2 = \sum(1,2,3)$$

x \ y	0	1
0	0	1
1	1	1

x \ y	0	1
0	1	0
1	0	1

$$f = \bar{x}.\bar{y} + x.y$$

3-variable K-map representation

x	y	z	min terms	
0	0	0	$\bar{x} . \bar{y} . \bar{z}$	m0
0	0	1	$\bar{x} . \bar{y} . z$	m1
0	1	0	$\bar{x} . y . \bar{z}$	m2
0	1	1	$\bar{x} . y . z$	m3
1	0	0	$x . \bar{y} . \bar{z}$	m4
1	0	1	$x . \bar{y} . z$	m5
1	1	0	$x . y . \bar{z}$	m6
1	1	1	$x . y . z$	m7

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

3-variable K-map representation

x	y	z	min terms	
0	0	0	$\overline{x} . \overline{y} . \overline{z}$	m0
0	0	1	$\overline{x} . \overline{y} . z$	m1
0	1	0	$\overline{x} . y . \overline{z}$	m2
0	1	1	$\overline{x} . y . z$	m3
1	0	0	$x . \overline{y} . \overline{z}$	m4
1	0	1	$x . \overline{y} . z$	m5
1	1	0	$x . y . \overline{z}$	m6
1	1	1	$x . y . z$	m7

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

x \ yz	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

x \ yz	00	01	11	10
0	0	1	1	0
1	0	1	1	0

What is the function represented by this K map?

x \ yz	00	01	11	10
0	1	0	1	0
1	0	1	1	0

$$f = \overline{x}.\overline{y}.\overline{z} + \overline{x}.y.z + x.\overline{y}.z + x.y.z$$

4-variable K-map representation

w	x	y	z	min terms
0	0	0	0	m_0
0	0	0	1	m_1
0	0	1	0	m_2
0	0	1	1	m_3
⋮	⋮	⋮	⋮	⋮
1	1	1	0	m_{14}
1	1	1	1	m_{15}

wx \ yz	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

wx \ yz	00	01	11	10
00	1	0	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	0

$$f = \overline{w}.\overline{x}.\overline{y}.\overline{z} + \overline{w}.\overline{x}.y.z + \overline{w}.x.\overline{y}.z + \overline{w}.x.y.z$$
$$+ w.x.\overline{y}.\overline{z} + w.x.y.\overline{z} + w.\overline{x}.\overline{y}.\overline{z}$$

Minimization using Kmap

$$f_2 = \sum(2,3)$$

x \ y	0	1
0	0	0
1	1	1

$$f = x.\overline{y} + x.y$$

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

$$f = x$$

Combine terms which differ in only one bit position. As a result, whatever is common remains.

x \ y	0	1
0	0	1
1	0	1

$$f = \overline{x}.y + x.y$$

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

$$\Rightarrow f = y$$

x \ y	0	1
0	1	0
1	1	0

$$\Rightarrow f = \overline{y}$$

x \ y	0	1
0	1	1
1	0	0

$$\Rightarrow f = \overline{x}$$

$$f_2 = \sum(1,2,3)$$

x \ y	0	1
0	0	1
1	1	1

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

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

$$= x\cdot\left(\overline{y} + y\right) + \left(\overline{x} + x\right)\cdot y$$

$$= x + y$$

The idea is to cover all the 1's with as few and as simple terms as possible

3-variable minimization

$$f = \overline{x}.\overline{y}.\overline{z} + \overline{x}.y.z + x.y.z + x.\overline{y}.z$$

x \ yz	00	01	11	10
0	1	0	1	0
1	0	1	1	0

$y.z$

$x.z$

$$f = \overline{x}.\overline{y}.\overline{z} + y.z + x.z$$

3-variable minimization

$$f = \bar{x}.\bar{y}.\bar{z} + \bar{x}.y.\bar{z} + x.y.z + x.\bar{y}.z$$

x \ yz	00	01	11	10
0	1	0	0	1
1	0	1	1	0

$$\bar{x}.\bar{z}$$

$$x.z$$

$$f = \bar{x}.\bar{z} + x.z$$

3-variable minimization

x \ yz	00	01	11	10
0	0	0	0	0
1	1	1	1	1

$$f = x.\overline{y}.\overline{z} + x.\overline{y}.z + x.y.z + x.y.\overline{z}$$

$x.\overline{y}$

$x.y$

$$f = x.\overline{y} + x.y$$

x \ yz	00	01	11	10
0	0	0	0	0
1	1	1	1	1

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

x

x \ yz	00	01	11	10
0	1	1	1	1
1	0	0	0	0

$\bar{x}$

x \ yz	00	01	11	10
0	1	0	0	1
1	1	0	0	1

$\bar{z}$

x \ yz	00	01	11	10
0	0	1	1	0
1	0	1	1	0

z

x \ yz	00	01	11	10
0	1	0	0	1
1	1	1	1	1

$\bar{z}$

x

$$f = x + \bar{z}$$

Can we do this ?

x \ yz	00	01	11	10
0	0	0	0	0
1	1	1	1	0

Note that each encirclement should represent a single product term. In this case it does not.

$$
\begin{aligned}
f &= x.\bar{y}.\bar{z} + x.\bar{y}.z + x.y.z \\
&= x.\bar{y} + x.z
\end{aligned}
$$

We do not get a single product term. In general we cannot make groups of 3 terms.

Can we use kmap with the following ordering of variables?

x \ yz	00	01	10	11
0	0	0	0	0
1	0	1	1	0

Can we combine these two terms into a single term ?

$$f = x.\overline{y}.z + x.y.\overline{z}$$

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

Note that no simplification is possible.

x \ yz	00	01	10	11
0	0	1	0	1
1	0	0	0	0

These two terms can be combined into a single term but it is not easy to show that on the diagram.

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

Kmap requires information to be represented in such a way that it is easy to apply the principle $x + \bar{x} = 1$

4-variable minimization

wx \ yz	00	01	11	10
00	1	0	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	0

$\overline{w}.y.z$

$\overline{w}.x.z$

$w.\overline{y}.\overline{z}$

$$f = \overline{w}.y.z + \overline{w}.x.z + w.\overline{y}.\overline{z} + \overline{w}.\overline{x}.\overline{y}.\overline{z} + w.x.y.\overline{z}$$

Is this the simplest expression ?

wx \ yz	00	01	11	10
00	1	0	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	0

$$w.\ x.\bar{y}.\bar{z}+w.\ x.y.\bar{z}=w.\ x.\bar{z}$$

wx \ yz	00	01	11	10
00	1	0	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	0

$$w.\ \bar{x}.\bar{y}.\bar{z}+\bar{w}.\ \bar{x}.\bar{y}.\bar{z}=\bar{x}.\ \bar{y}.\bar{z}$$

4-variable minimization

wx \ yz	00	01	11	10
00	1	0	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	0

$\overline{x}.\overline{y}.\overline{z}$

$\overline{w}.y.z$

$w.x.\overline{z}$

$\overline{w}.x.z$

$w.\overline{y}.\overline{z}$

$$f = \overline{w}.y.z + \overline{w}.x.z + w.\overline{y}.\overline{z} + w.x.\overline{z} + \overline{x}.\overline{y}.\overline{z}$$

Is this the best that we can do ?

Cover the 1’s with minimum number of terms

wx \ yz	00	01	11	10
00	1	0	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	0

$$\mathrm{f} = \overline{w}.y.z + \overline{w}.x.z + w.\overline{y}.\overline{z} + w.x.\overline{z} + \overline{x}.\overline{y}.\overline{z}$$

wx \ yz	00	01	11	10
00	1	0	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	0

$$\mathrm{f} = \overline{w}.y.z + \overline{w}.x.z + w.x.\overline{z} + \overline{x}.\overline{y}.\overline{z}$$

4-variable minimization

wx \ yz	00	01	11	10
00	1	0	0	0
01	1	1	0	0
11	0	0	0	0
10	1	0	0	1

wx \ yz	00	01	11	10
00	1	0	0	0
01	1	1	0	0
11	0	0	0	0
10	1	0	0	1

$$f = \bar{w}.x.\bar{y} + w.\bar{x}.\bar{z} + \bar{w}.\bar{y}.\bar{z}$$

$$f = \bar{w}.x.\bar{y} + w.\bar{x}.\bar{z} + \bar{x}.\bar{y}.\bar{z}$$

Groups of 4

wx \ yz	00	01	11	10
00	0	1	0	0
01	1	1	1	1
11	0	1	0	0
10	0	1	0	0

$\overline{w}.x$

$\overline{y}.z$

wx \ yz	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	1	1	0
10	0	1	1	0

$x.z$

$w.z$

wx \ yz	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	1	0	0	1
10	0	0	0	0

$x.\overline{z}$

wx \ yz	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	0	0	0	0
10	0	1	1	0

$\overline{x}.z$

wx \ yz	00	01	11	10
00	1	0	0	1
01	0	0	0	0
11	0	0	0	0
10	1	0	0	1

$\overline{x}.\overline{z}$

wx \ yz	00	01	11	10
00	1	0	1	0
01	0	0	0	0
11	0	0	0	0
10	1	0	1	0

??

Groups of 8

wx \ yz	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	1	1	0
10	0	1	1	0

z

wx \ yz	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	0	0	0	0

x

wx \ yz	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

$\overline{z}$

wx \ yz	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

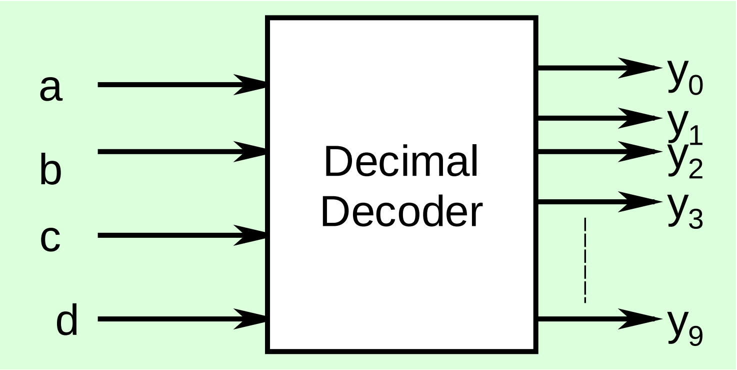
$\overline{x}$

Examples

wx \ yz	00	01	11	10
00	0	1	0	1
01	1	1	1	1
11	1	1	1	1
10	0	0	0	1

wx \ yz	00	01	11	10
00	0	1	0	1
01	1	1	0	1
11	1	1	1	1
10	0	0	0	1

Don’t care terms



Y_3

ab \ cd	00	01	11	10
00	0	0	1	0
01	0	0	0	0
11	x	x	x	x
10	0	0	x	x

$$y_3 = \overline{a}.\overline{b}.c.d$$

a	b	c	d	$y_0y_1y_2y_3y_4y_5y_6y_7y_8y_9$
0	0	0	0	1 0 0 0 0 0 0 0 0 0
0	0	0	1	0 1 0 0 0 0 0 0 0 0
0	0	1	0	0 0 1 0 0 0 0 0 0 0
0	0	1	1	0 0 0 1 0 0 0 0 0 0
0	1	0	0	0 0 0 0 1 0 0 0 0 0
0	1	0	1	0 0 0 0 0 1 0 0 0 0
0	1	1	0	0 0 0 0 0 0 1 0 0 0
0	1	1	1	0 0 0 0 0 0 0 1 0 0
1	0	0	0	0 0 0 0 0 0 0 0 1 0
1	0	0	1	0 0 0 0 0 0 0 0 0 1
1	0	1	0	x x x x x x x x x x
1	0	1	1	x x x x x x x x x x
1	1	0	0	x x x x x x x x x x
1	1	0	1	x x x x x x x x x x
1	1	1	0	x x x x x x x x x x
1	1	1	1	x x x x x x x x x x

Don’t care terms can be chosen as 0 or 1. Depending on the problem, we can choose the don’t care term as 1 and use it to obtain a simpler Boolean expression

Y_3

ab \ cd	00	01	11	10
00	0	0	1	0
01	0	0	0	0
11	x	x	x	x
10	0	0	x	x

$$y_3 = \overline{b}.c.d$$

Don’t care terms should only be included in encirclements if it helps in obtaining a larger grouping or smaller number of groups.

Product of Sum (PoS) Terms Representation

x	y	f_1
0	0	0
0	1	1
1	0	1
1	1	0

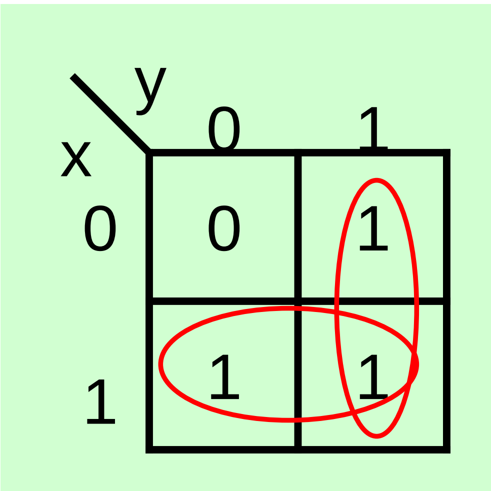
x	y	Max term	
0	0	$x + y$	M0
0	1	$x + \overline{y}$	M1
1	0	$\overline{x} + y$	M2
1	1	$\overline{x} + \overline{y}$	M3

$$\mathrm{f}_1 = (x + y) \cdot (\overline{x} + \overline{y})$$

$$\mathrm{f}_1 = M_0 \cdot M_3$$

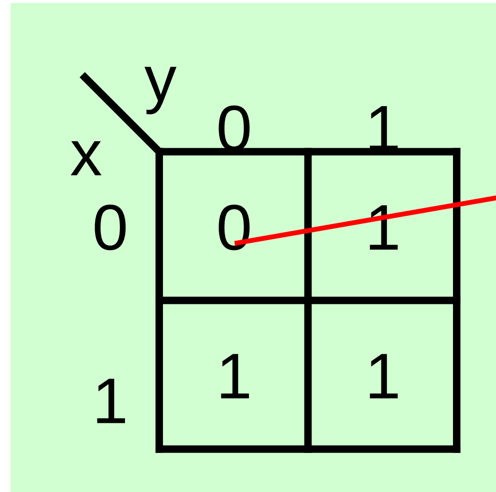
$$\mathrm{f}_1 = \prod (M_0, M_3)$$

Minimization of Product of Sum Terms using Kmap

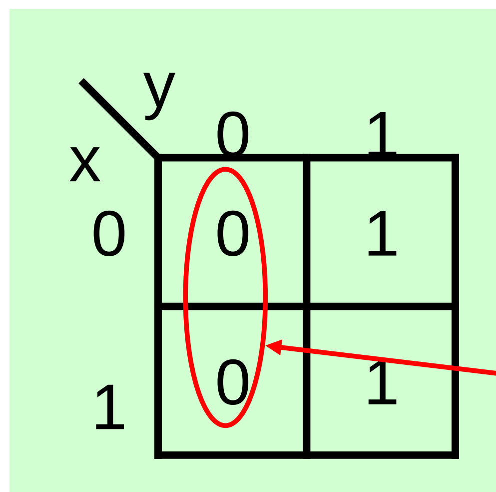


$$f = x + \bar{x}.y + x.y$$
$$= x + (\bar{x} + x).y$$
$$= x + y$$

Sum of Product (SoP)



$$f = x + y$$



$$f = y$$

x \ y	0	1
0	1	0
1	1	0

$$\Rightarrow f = \overline{y}$$

x \ y	0	1
0	1	1
1	0	0

$$\Rightarrow f = \overline{x}$$

x \ yz	00	01	11	10
0	1	0	0	1
1	0	1	1	0

$$\overline{x} + z$$

$$x + \overline{z}$$

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

$$\Rightarrow f = \overline{x}.\overline{z} + x.z$$

$$x + y + z$$

wx \ yz	00	01	11	10
00	0	1	0	1
01	1	1	1	1
11	1	0	1	1
10	0	0	0	0

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

$$\overline{w} + y + \overline{z}$$

$$\overline{w} + x$$

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

Example

Obtain the minimized PoS by suitably using don't care terms

wx \ yz	00	01	11	10
00	1	x	0	1
01	1	0	1	1
11	0	x	1	1
10	1	x	1	x

$$f = (w + x + \bar{z}).(\bar{w} + \bar{x} + y).(y + \bar{z})$$

Design Flow

System Description

↓

Truth Table

↓

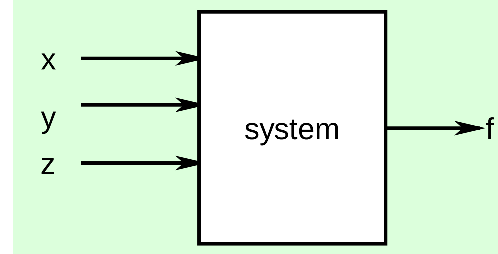
Boolean Expression

↓

Minimized Boolean Expression

↓

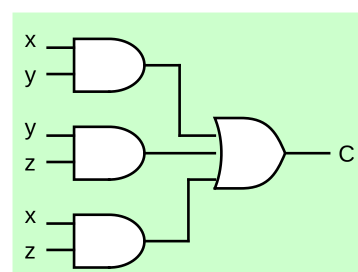
Gate Netlist



x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

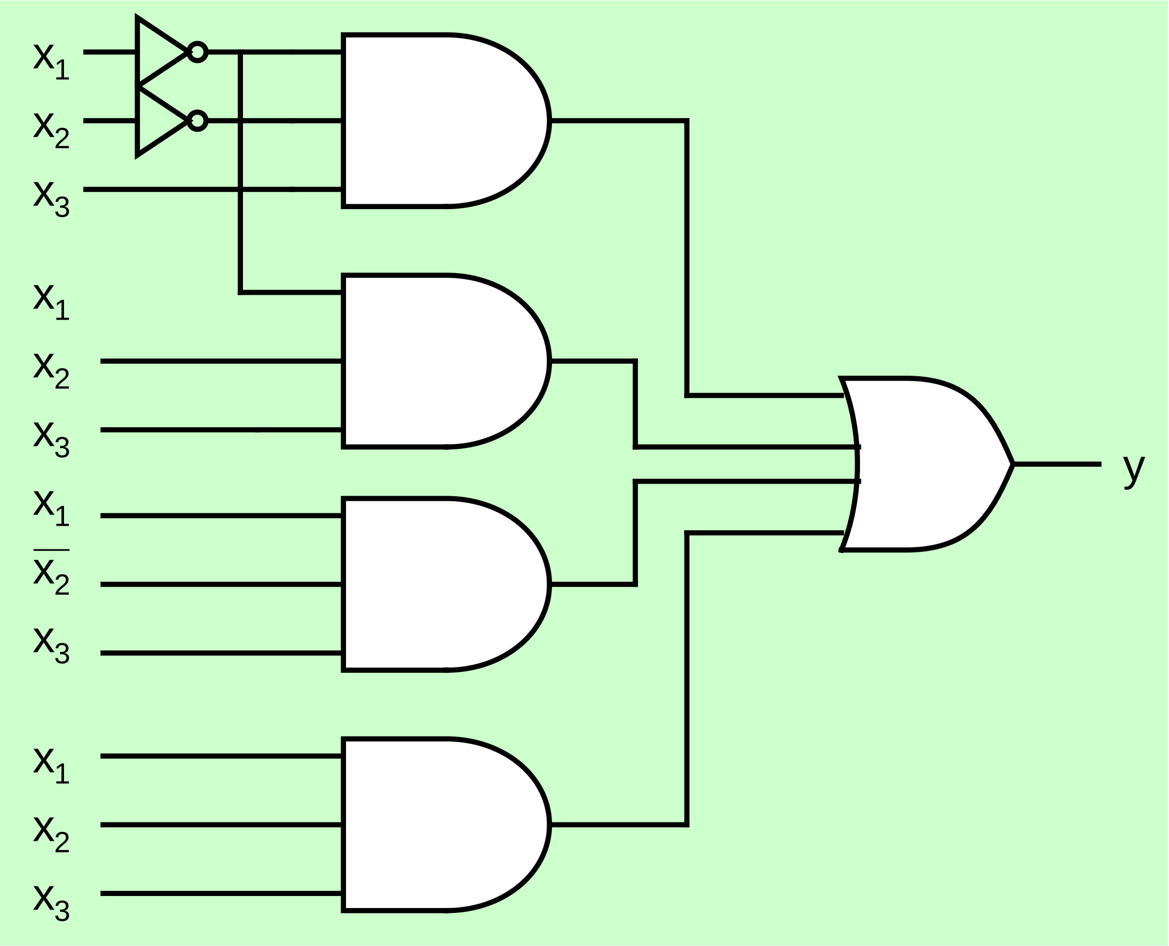
$$\mathrm{f} = \bar{x}.\bar{y}.z + \bar{x}.y.z + x.\bar{y}.z + x.y.z$$

$$\Rightarrow \mathrm{f} = \bar{x}.\bar{z} + x.z$$



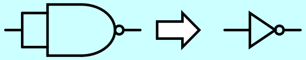
Mapping of Boolean expression to a Network of gates available in the library

$$y = \overline{x_1} . \overline{x_2} . x_3 + \overline{x_1} . x_2 . x_3 + x_1 . \overline{x_2} . x_3 + x_1 . x_2 . x_3$$



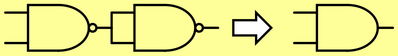
Implementation using only NAND gates

NAND to Inverter



$$\overline{x.x} = \overline{x}$$

NAND to AND

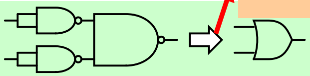


$$\overline{x.y}$$

$$x.y$$

NAND to OR

$$\overline{x}$$

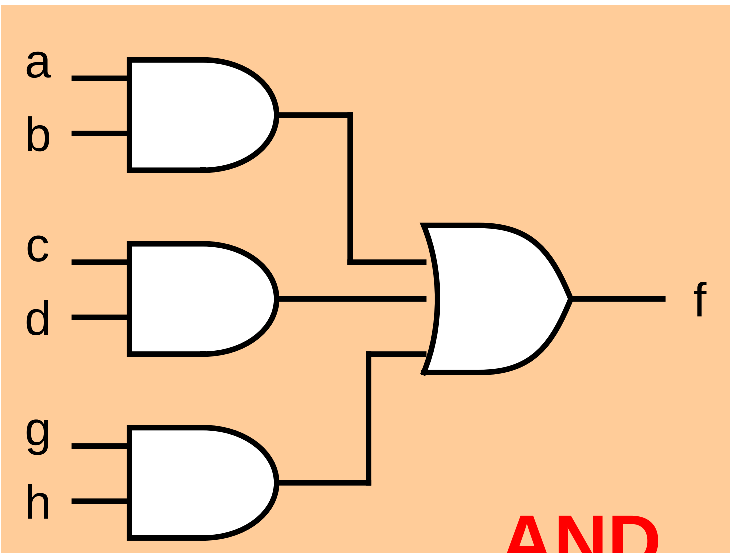


$$\overline{y}$$

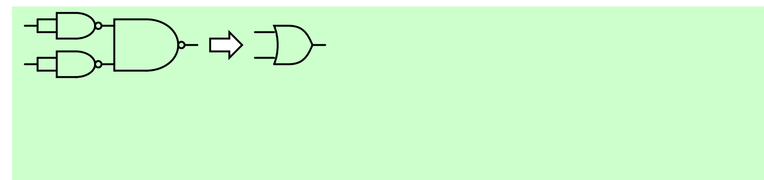
$$f = \overline{\overline{x}.\overline{y}} = x + y$$

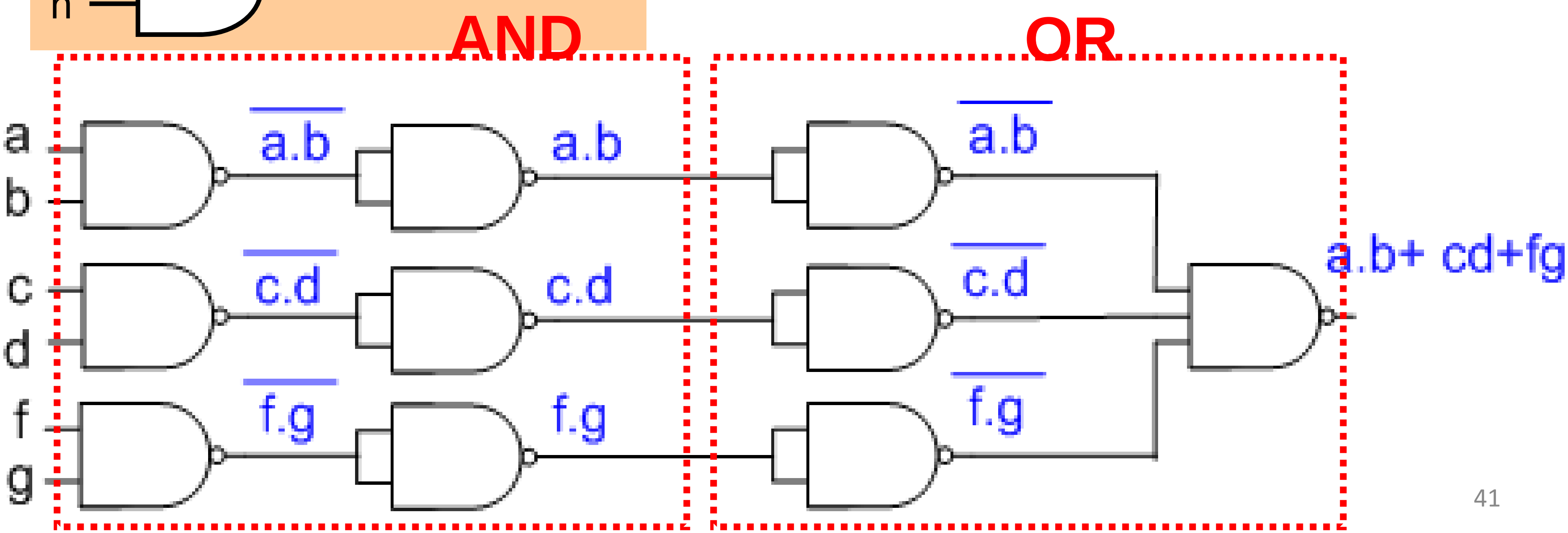
Implementation using only NAND gates

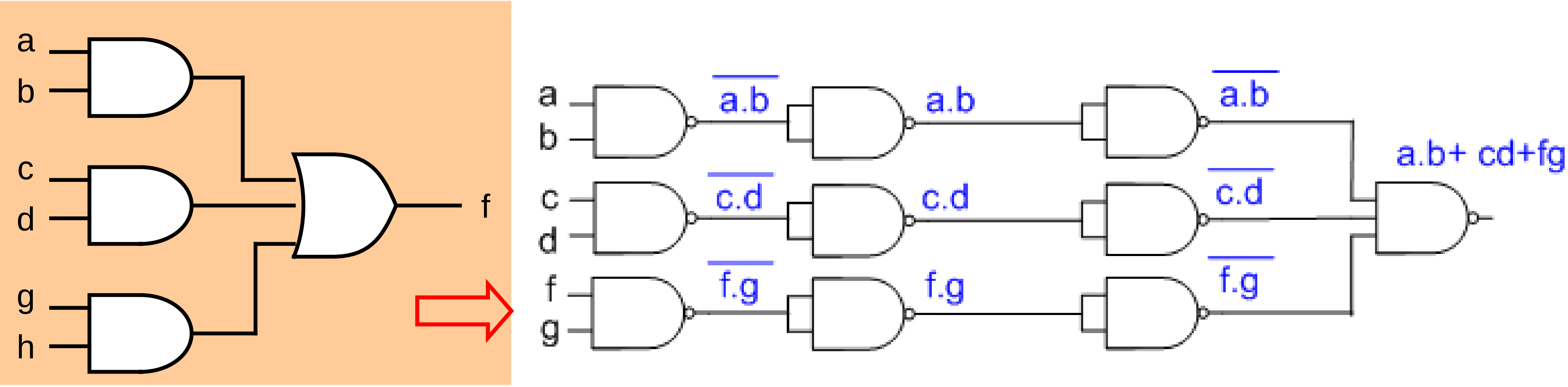
A SoP expression is easily implemented with NAND gates.

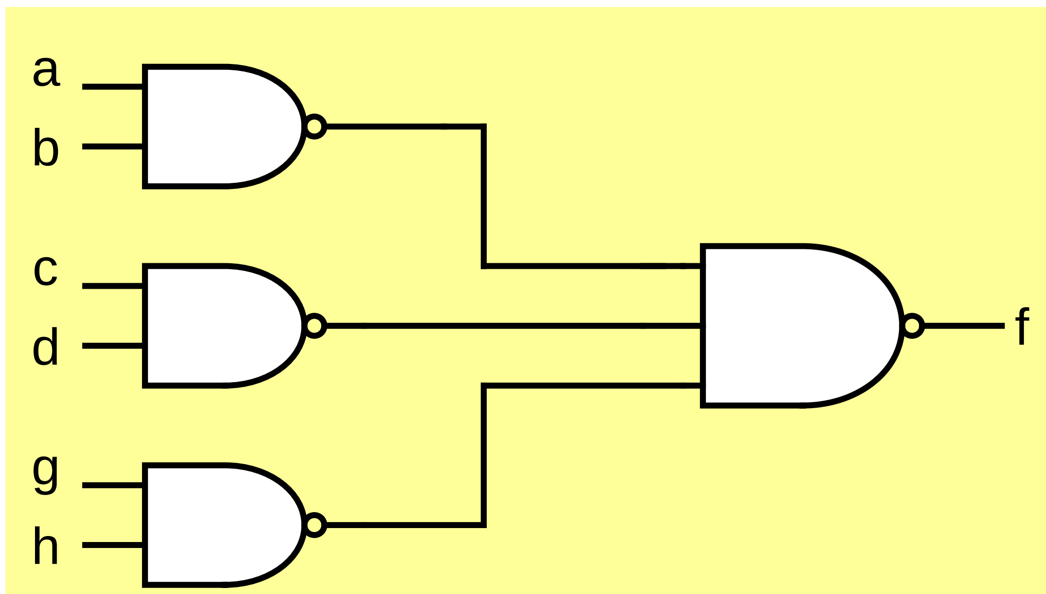


$$f = a.b + c.d + f.g$$





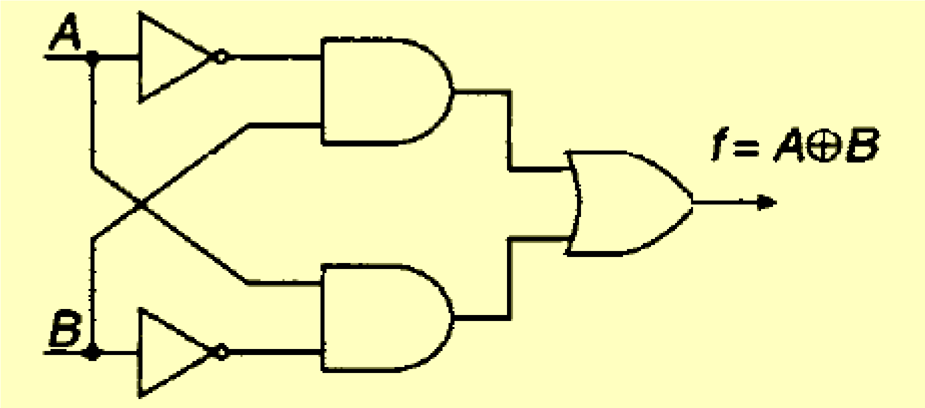


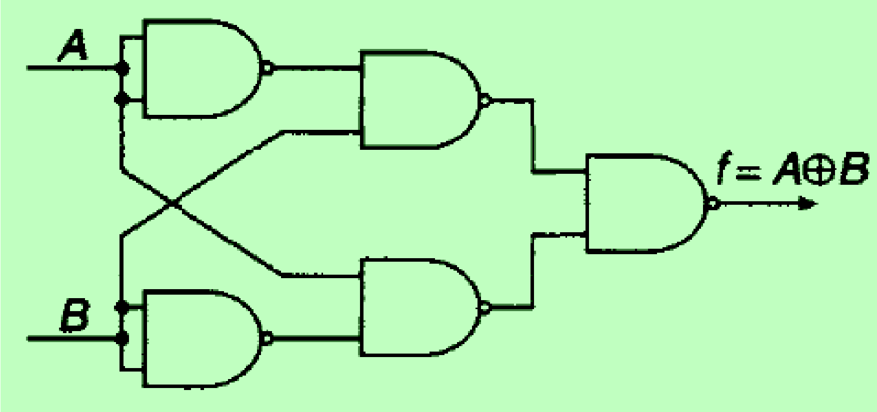


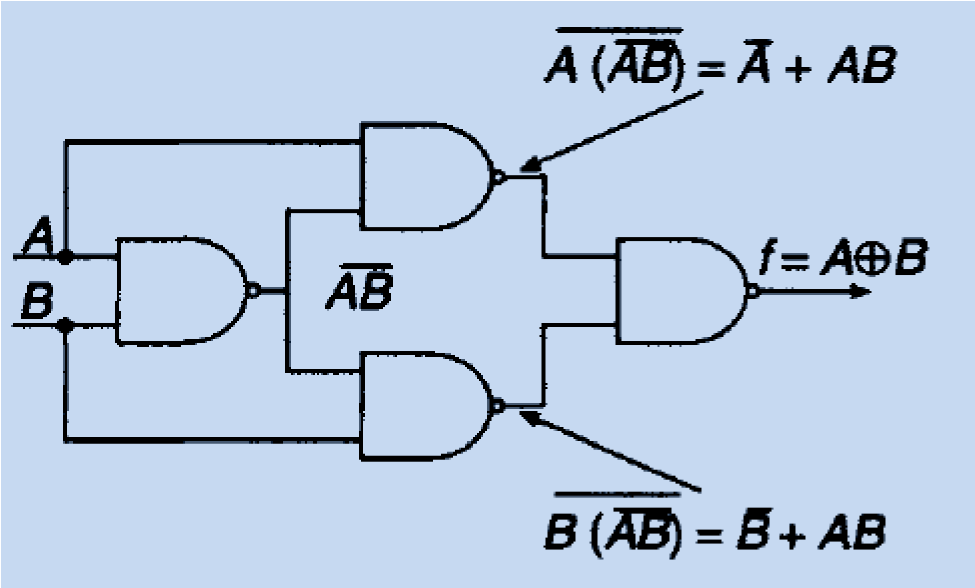
There is a one-to-one mapping between AND-OR network and NAND network

Often there is lot of further optimization that can be done

Consider implementation of XOR gate $f = \overline{A}.B + A.\overline{B}$

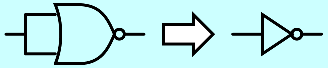






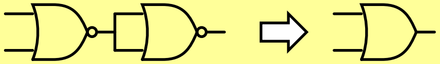
Implementation using only NOR gates

NOR to Inverter



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

NOR to OR

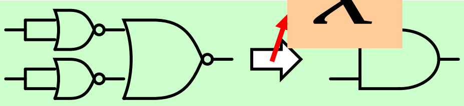


$$\overline{x+y}$$

$$x+y$$

NOR to AND

$$\overline{x}$$



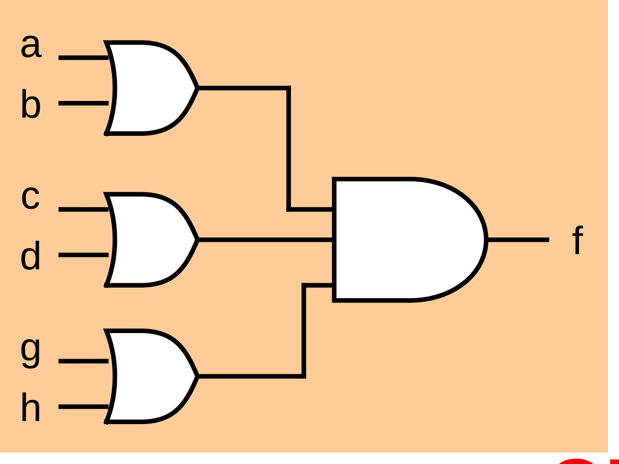
$$\overline{y}$$

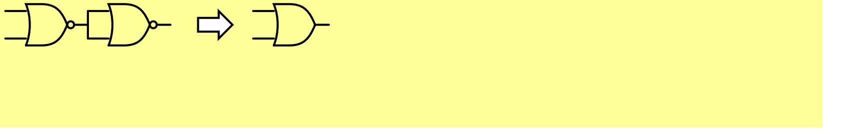
$$f=\overline{\overline{x}+\overline{y}}=x.y$$

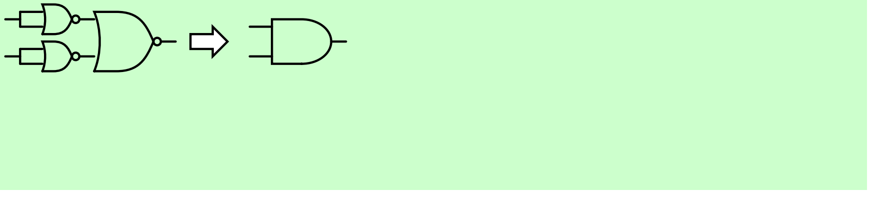
Implementation using only NOR gates

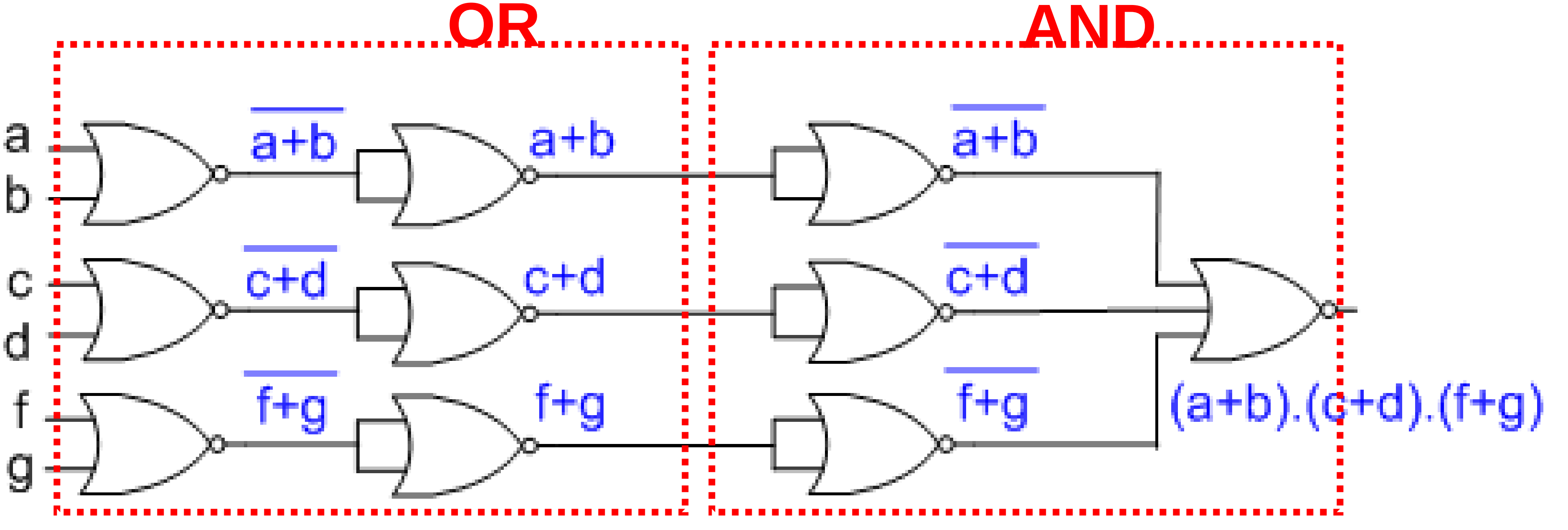
To implement using NOR gates, it is easiest to start with minimized Boolean expression in POS form

$$f = (a+b).(c+d).(f+g)$$

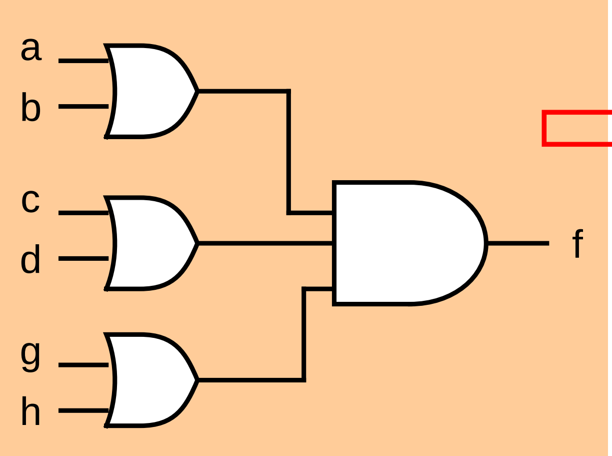


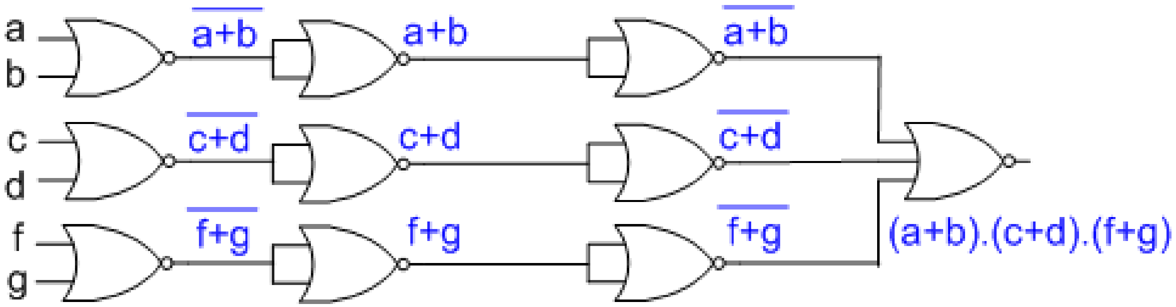


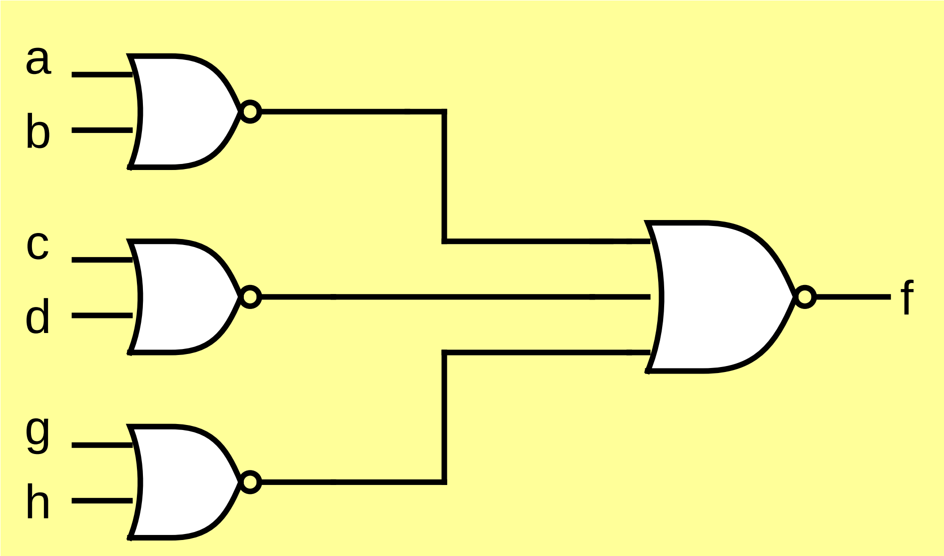




$$f = (a+b).(c+d).(f+g)$$







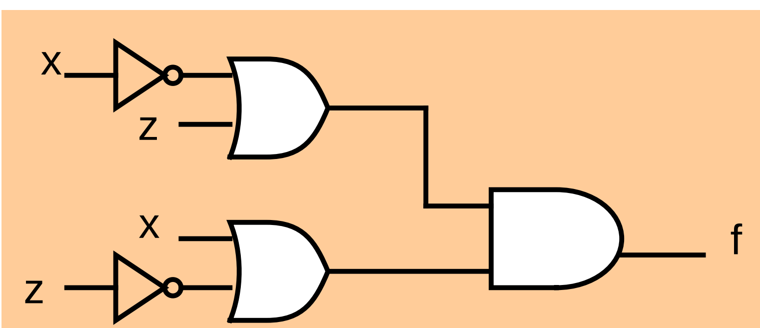
There is a one-to-one mapping between OR-AND network and NOR network

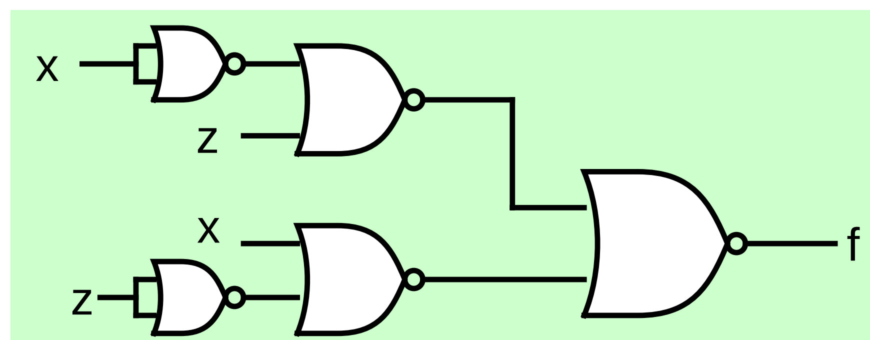
To implement SoP expression using NOR gates, determine first the corresponding PoS expression and then follow the procedure outlined earlier

Implement $f(x,y,z) = \bar{x}\,.\,\bar{z} + x\,.\,z$ using NOR gates

x \ yz	00	01	11	10
0	1	0	0	1
1	0	1	1	0

$\Rightarrow\ f = (\bar{x} + z).(x + \bar{z})$





Similarly PoS expression can be implemented as NAND network by first converting it to SoP expression and then following the procedure outlined earlier

How do we get the chocolate?