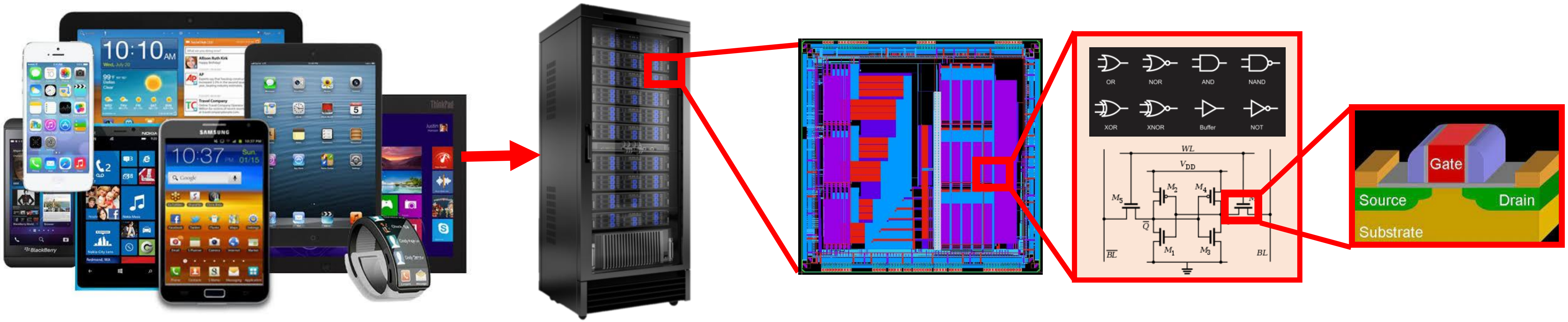
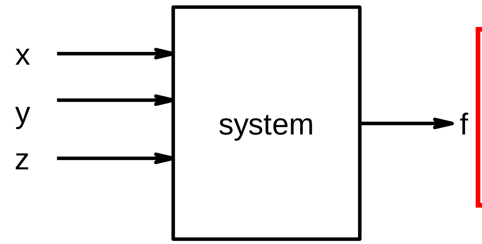


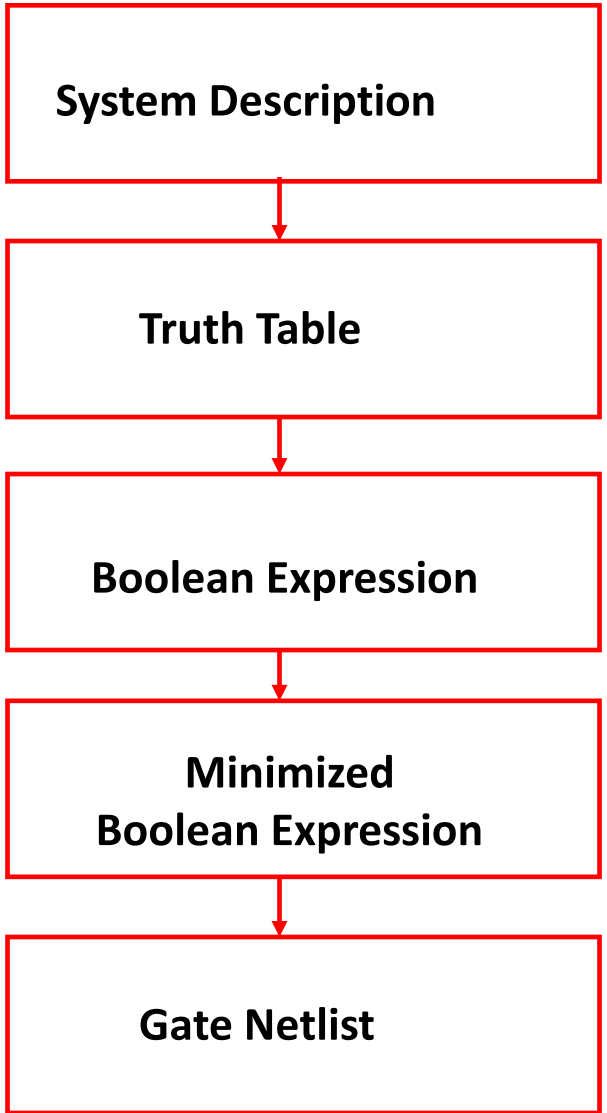
ESC201: Introduction to Electronics

Module 6: Digital Circuits

Dr. Shubham Sahay,
Associate Professor,
Department of Electrical Engineering,
IIT Kanpur

Design Flow

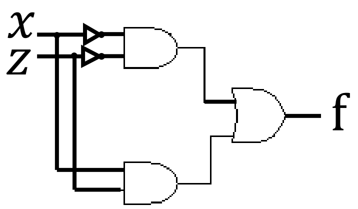




x	y	z	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

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

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



K-map Representation of Truth Table

x	y	min term
0	0	$\bar{x} . \bar{y}$ m0
0	1	$\bar{x} . y$ m1
1	0	$x . \bar{y}$ m2
1	1	$x . y$ m3

x \ y	0	1
0	m_0	m_1
1	m_2	m_3

Example

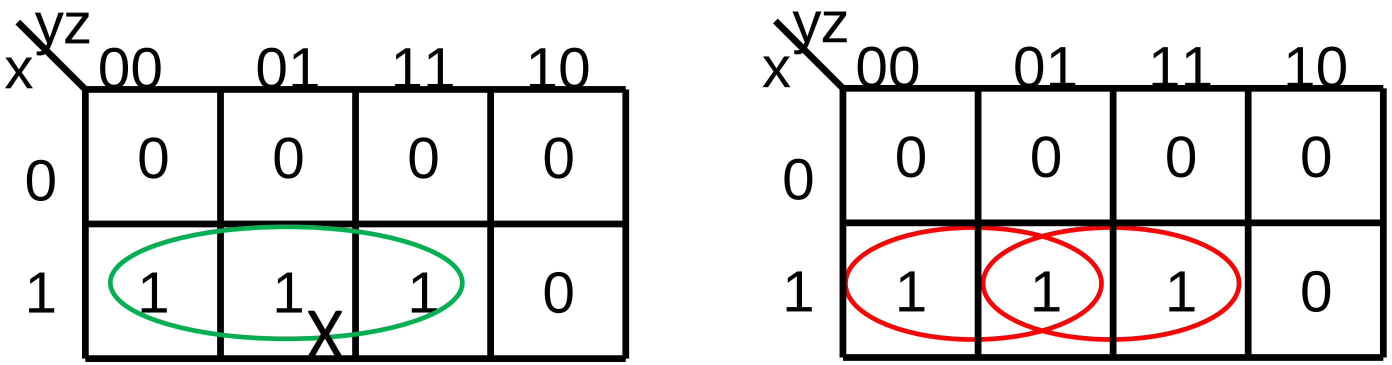
x	y	f_1
0	0	0
0	1	1
1	0	1
1	1	0

$\Rightarrow$

x \ y	0	1
0	0	1
1	1	0

Can We Group 3 Min Terms?

Can we do this ?



Note that each encirclement should represent a single product term.

In this case it does not represent a single product term**.**

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

We do not get a single product term.

Grouping in 3 terms will not help in minimization of terms in function.

Reordering of Numbering not Beneficial for Simplification

Can we use K-map with the following ordering of variables?

NOT A GOOD IDEA!

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 ?

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

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

Note that no simplification is possible here.

K-map requires variable to change one bit between adjacent cells

- Continued -

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

NOT A GOOD IDEA!

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 $x + \bar{x} = 1$

Example

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

$\overline{x}$

Example

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

$\overline{z}$

Example

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

z

Example

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

$\overline{z}$

x

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

4-variable Minimisation

Example

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

$\overline{w}.y.z$

$\overline{w}.x.z$

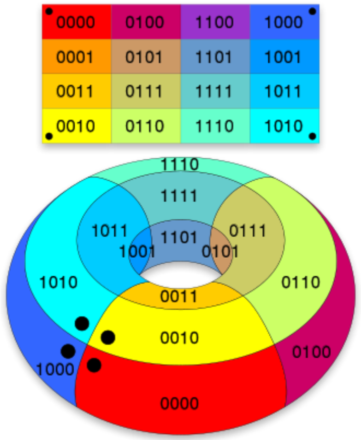
$w.\overline{y}.\overline{z}$

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

But is this the simplest expression ?

The K-Map Folds on Itself

K-map drawn on a torus, and in a plane. The dot-marked cells are adjacent.
- *source: Wikipedia*



Example (continued)

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

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

Example (continued)

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

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

Example (continued)

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

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

$\overline{w}.y.z$

$\overline{w}.x.z$

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

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

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

Is this the best that we can do ?

Cover the 1's with Minimum Number of Groupings

Example (continued)

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

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

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

Redundant

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

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

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

Equivalent Solutions

Example

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

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

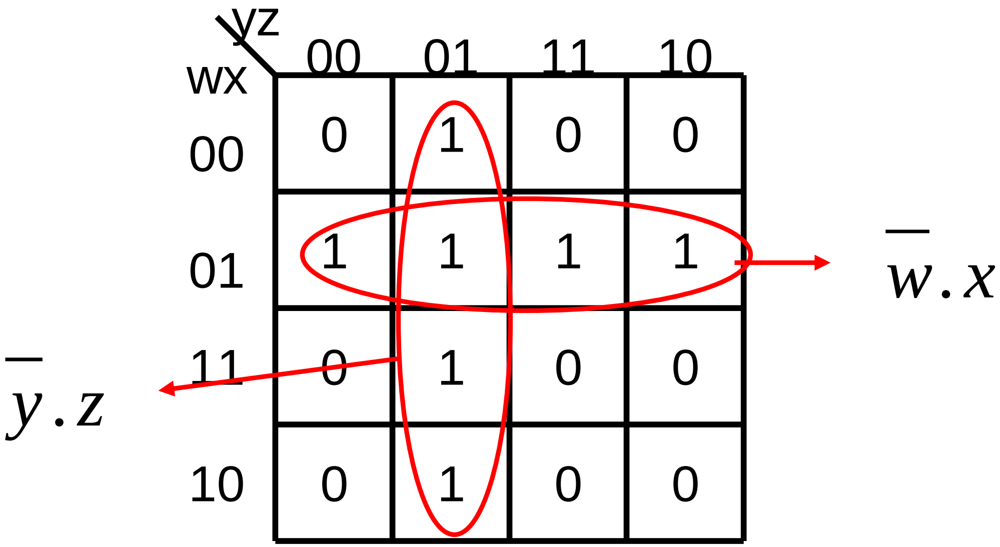
wx \ yz	00	01	11	10
00	1	0	0	0
01	1	1	0	0
11	0	0	0	0
10	1	0	0	1

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

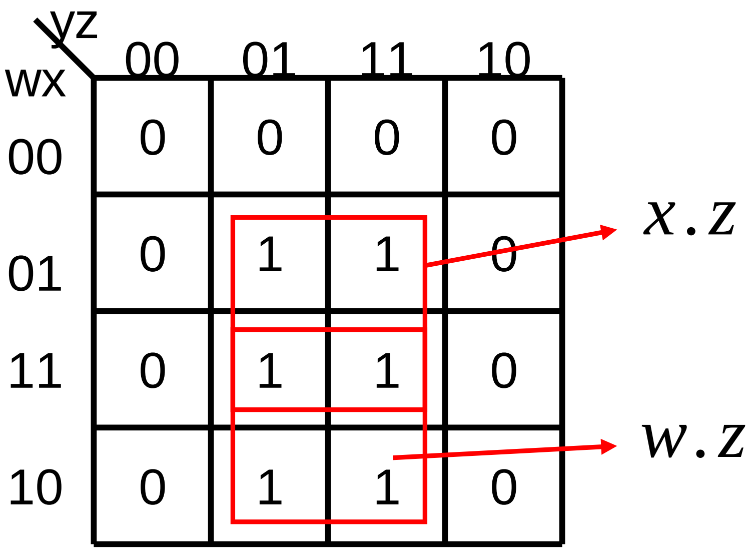
Both solutions are equivalent.

Groupings of Four

Example



Example



Example

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

$x.\bar{z}$

Example

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

$\bar{x}.z$

Example

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

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

Example

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

??

Groupings of Eight

Example

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

Example

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

Example

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

$\overline{z}$

Example

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

$\overline{x}$

Example

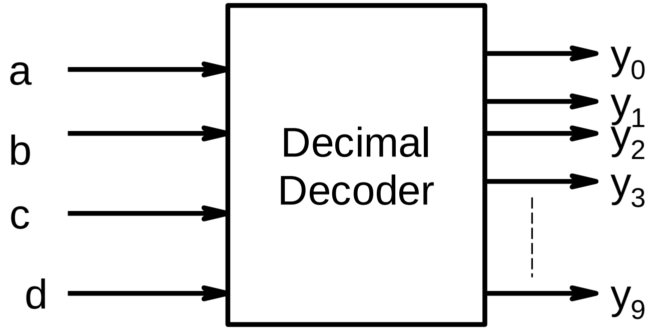
wx \ yz	00	01	11	10
00	0	1	0	1
01	1	1	1	1
11	1	1	1	1
10	0	0	0	1

Example

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

Example



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

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

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

Choice of Don’t Care Terms

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

Example

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

x ≡ 1

x ≡ 0

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

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

Minimisation of PoS Terms using K-map

Example

x \ y	0	1
0	0	1
1	1	1

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

Example

x \ y	0	1
0	0	1
1	1	1

$f = x + y$

Example

x \ y	0	1
0	0	1
1	0	1

$f = y$

Example

x \ y	0	1
0	1	0
1	1	0

$\Rightarrow f = \overline{y}$

Example

x \ y	0	1
0	1	1
1	0	0

$\Rightarrow f = \overline{x}$

Example

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

$\overline{x}.+z$

$x+\overline{z}$

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

Example

$x+y+z$

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

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

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

$\overline{w}+x$

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

Choosing x to be 0 to optimise PoS

Obtain the minimized PoS by suitably making don't care terms to be zero.

Rest of don't care(s) are to be chosen as one.

Example

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

For grouping you assume:

- all selected x^s are '0'
- not selected x is '1'

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

Design Flow

System Description

↓

Truth Table

↓

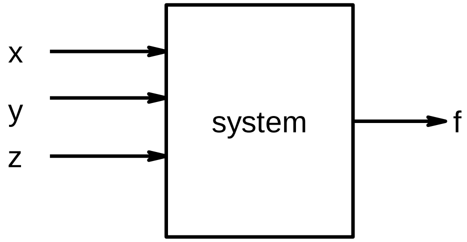
Boolean Expression

↓

Minimized Boolean Expression

↓

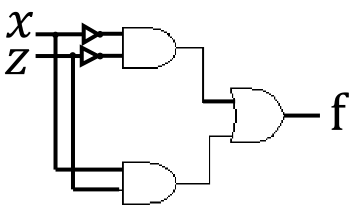
Gate Netlist



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

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

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



Boolean Functions

So far we have been looking at SoP or PoS forms

For example:

$\mathrm{f} = \overline{x}\cdot\overline{z} + x\cdot z$ or $\mathrm{f} = (\overline{x} + z)\ \ (x + \overline{z})$

This requires AND, OR and NOT operations.

First choice use “Gates” to implement AND, OR and NOT operations.

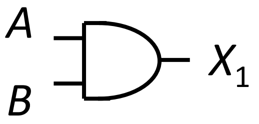
(The Boolean function and interactions are gating the transmission of 1)

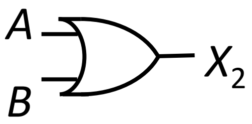
There may be other ways to implement the functions as well!

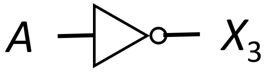
The Basic Gates

Popular introductory gates to represent Boolean operations: {AND, OR and NOT}

Gate







Operation

AND | OR | NOT

Algebraic Represetnation

$X_1 = A \cdot B = A \wedge B$

$X_2 = A + B = A \vee B$

$X_3 = \overline{A} = A' = \neg A$

Truth Table

A	**B**	$\boldsymbol{X_1}$
0	0	0
0	1	0
1	0	0
1	1	1

A	**B**	$\boldsymbol{X_1}$
0	0	0
0	1	1
1	0	1
1	1	1

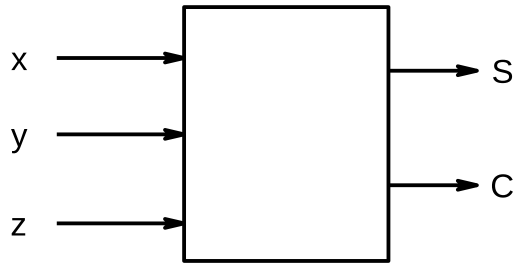
A	$\boldsymbol{X_3}$
0	1
1	0

The <u>three gates together</u> form a basis set for representing Boolean relationship.

AND and OR gates with more than two inputs is a possibility and often used.

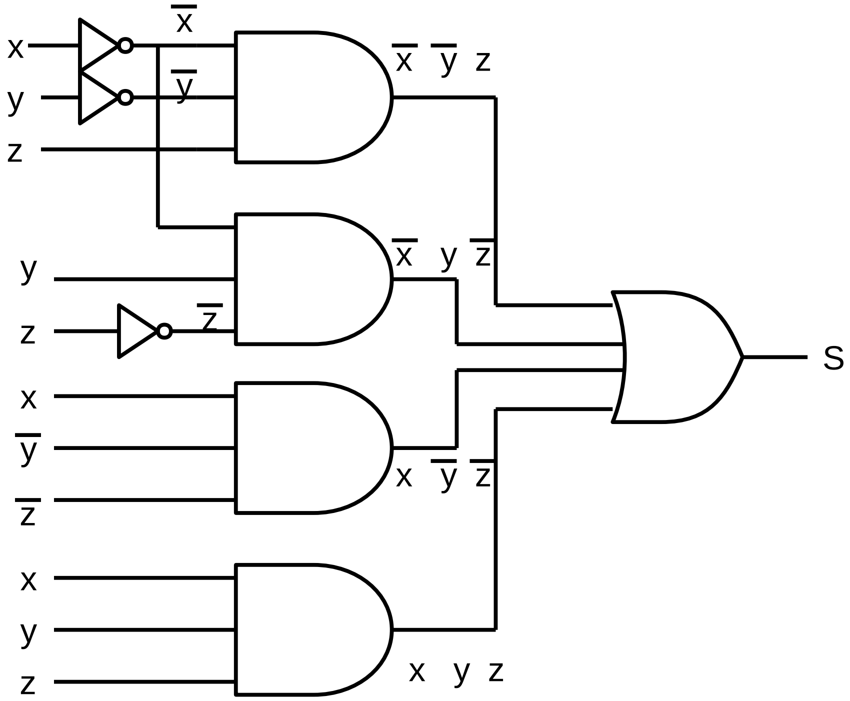
Implementing Boolean expressions using gates

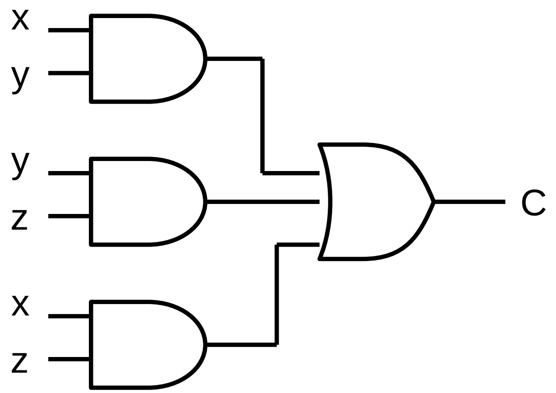
Example



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

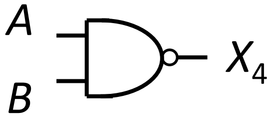
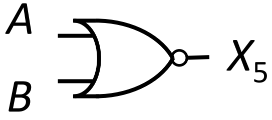
$$C = x.y + x.z + y.z$$





Popular Universal Gates

Two gates are popular for implementing Boolean Logic in hardware

	NAND	NOR
Gate		
Operation	NAND	NOR
Algebraic Represetnation	$X_4 = \overline{A \cdot B} = \overline{A} + \overline{B}$	$X_5 = \overline{A + B} = \overline{A} \cdot \overline{B}$

Truth Table

A	**B**	**X_1**
0	0	1
0	1	1
1	0	1
1	1	0

A	**B**	**X_1**
0	0	1
0	1	0
1	0	0
1	1	0

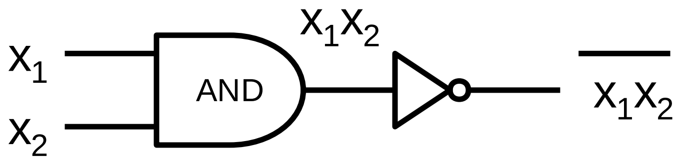
<u>Each of these two gates</u> form a basis set for representing Boolean relationship

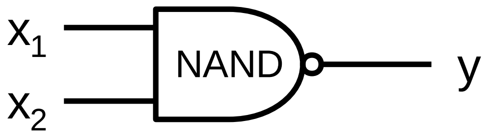
They are examples of **Universal Gates** to implement Boolean functions.

More than two inputs NAND and NOR gates is a possibility and are often used.

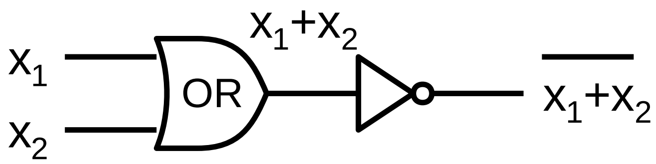
Parsing the NAND and NOR Gates

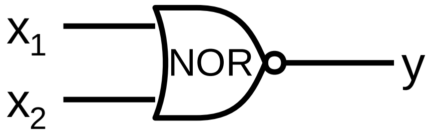
NAND: $y = \overline{x_1 . x_2}$





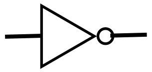
NOR: $y = \overline{x_1 + x_2}$





Basic Boolean Operations with NAND

NOT operation

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

AND operation

x

y

$x \cdot y$ → x y $\qquad \overline{x.y} \qquad x.y$

OR operation

x

y

$x + y$ → x y $\qquad \overline{x} \qquad \overline{y} \qquad \overline{\overline{x}.\overline{y}} = x + y$

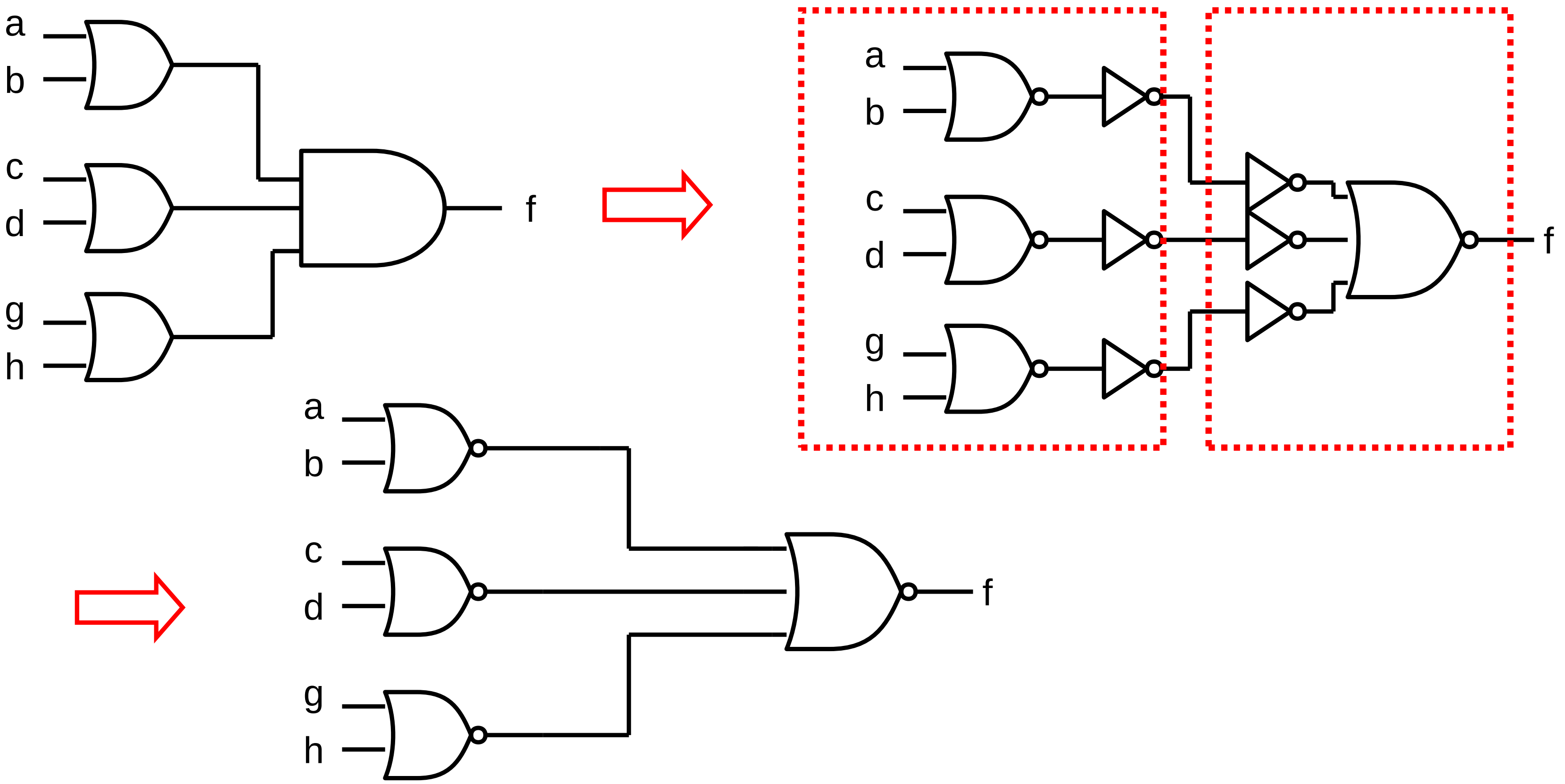
Exercise

Implement NOT, AND and OR with NOR gates

Implementing Boolean Function with Universal Gates

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

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



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

Similarly, there is a one-to-one mapping between AND-OR network and NAND network.

Example

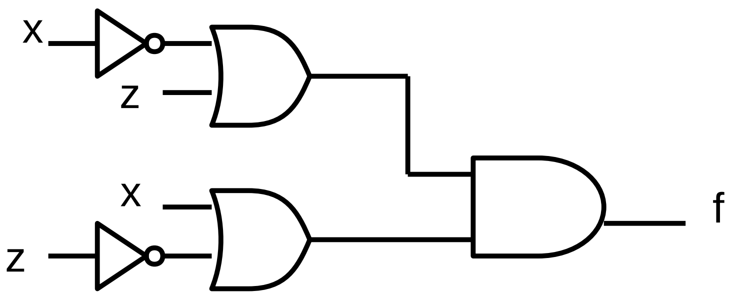
Implement f(x,z) $= \overline{x}.\overline{z} + x.z$ using NOR gates

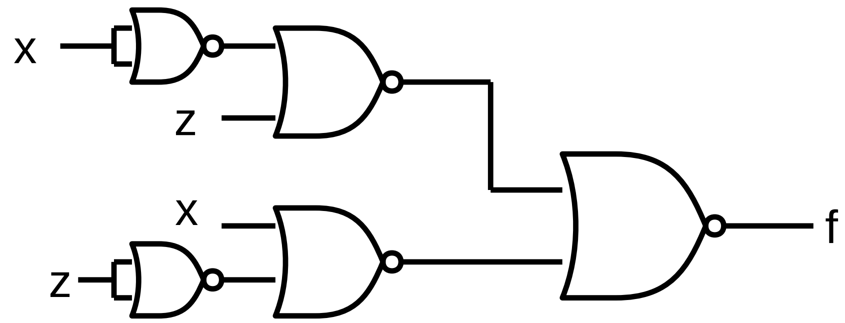
This is XNOR function!

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

Find out PoS

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





Similarly SoP expression can be implemented as NAND network.

→ first convert to SoP expression → then follow procedure outlined earlier