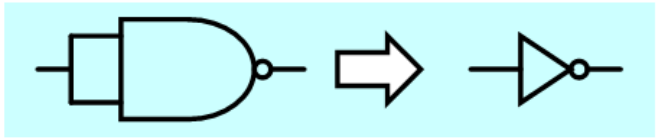
ESc201 : Introduction to Electronics

Digital Circuits

Amit Verma
Dept. of Electrical Engineering
IIT Kanpur

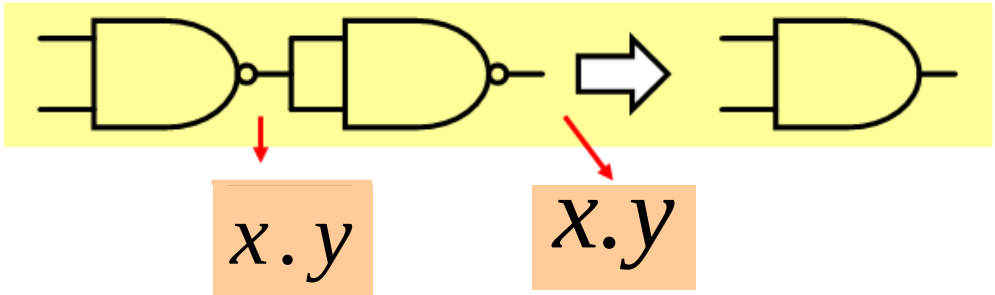
Implementation using only NAND gates (recap)

NAND to Inverter

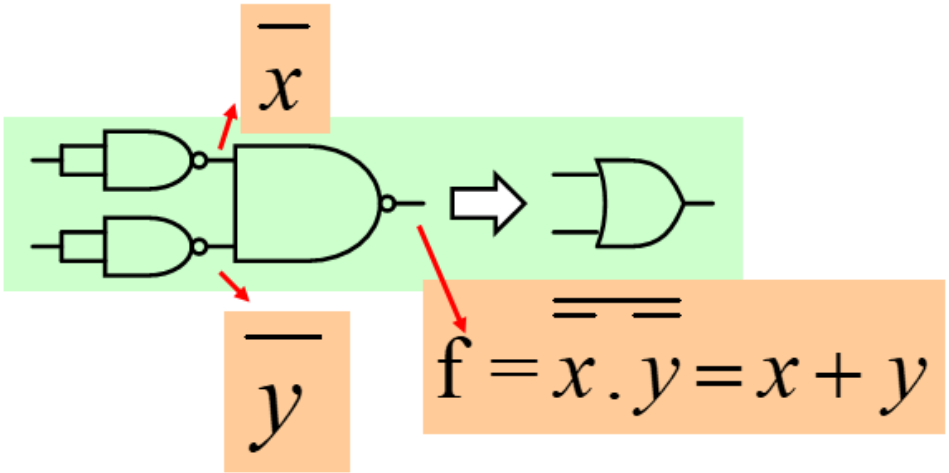


$$\overline{x.x} = \overline{x}$$

NAND to AND

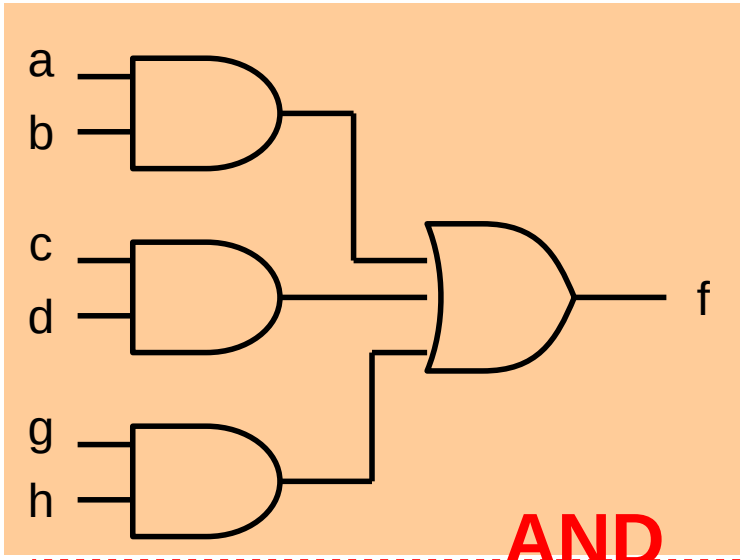


NAND to OR

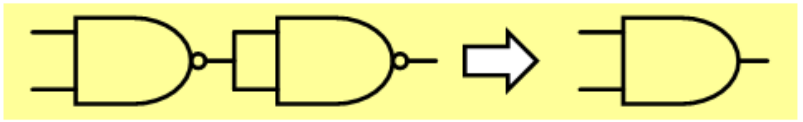


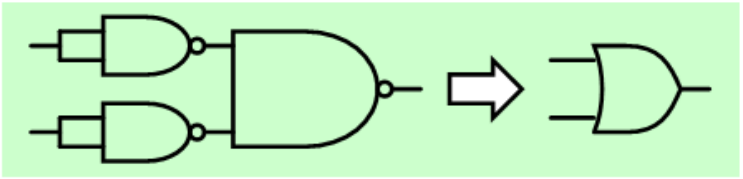
Implementation using only NAND gates (recap)

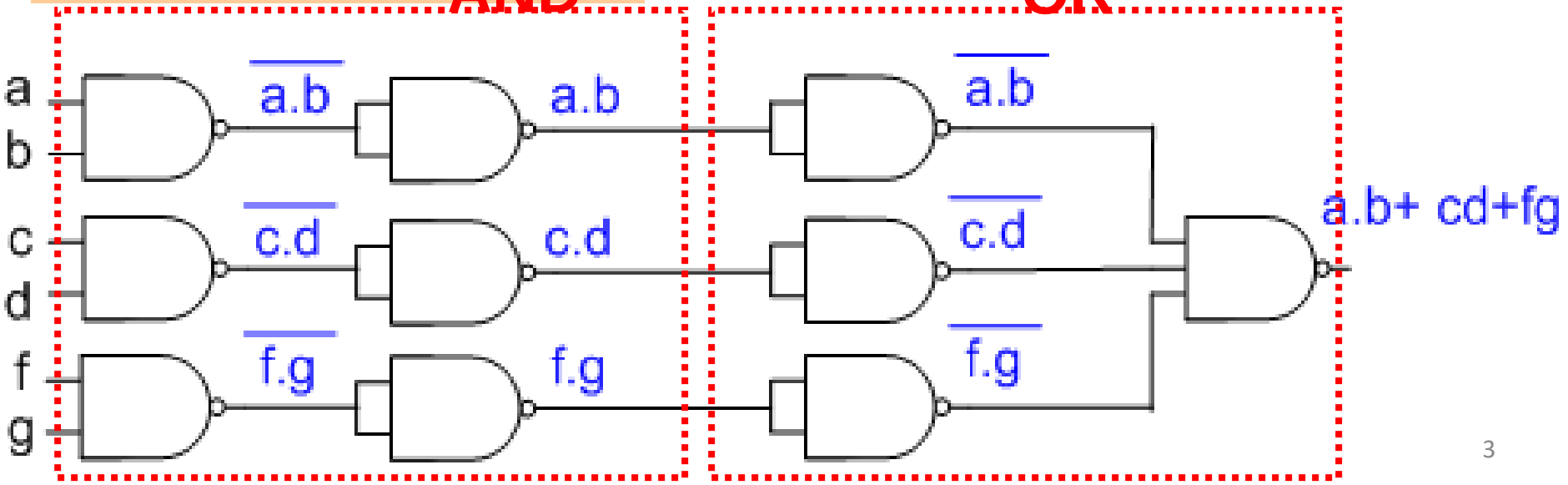
A SoP expression is easily implemented with NAND gates.

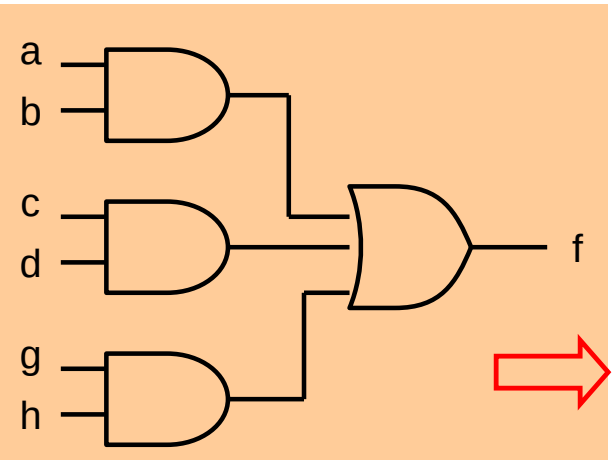


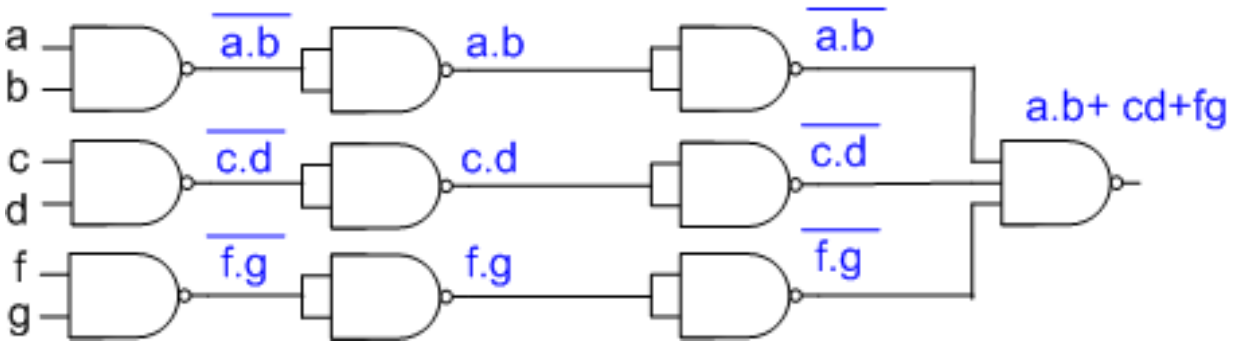
$$f = a.b + c.d + f.g$$

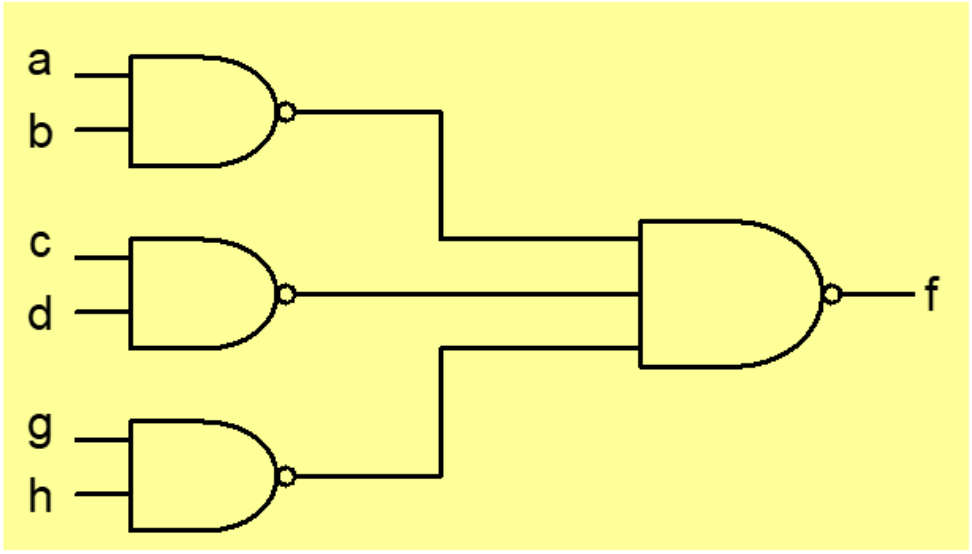








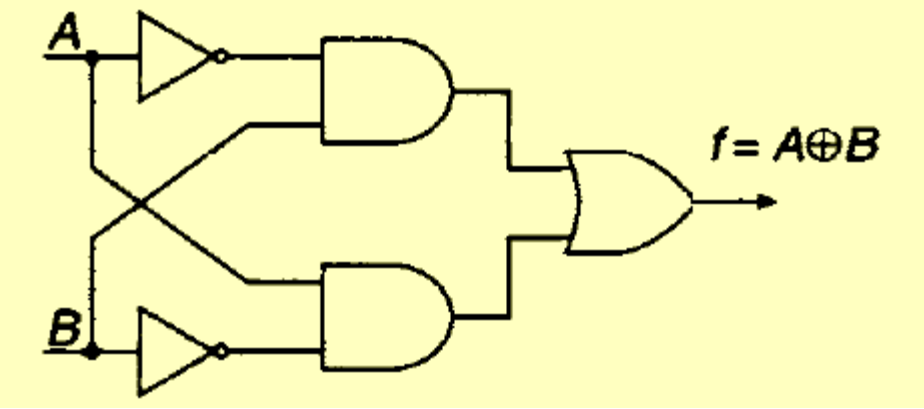


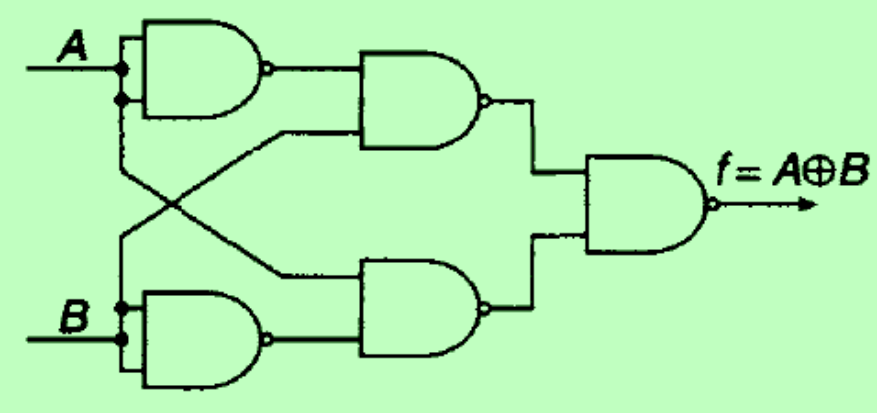


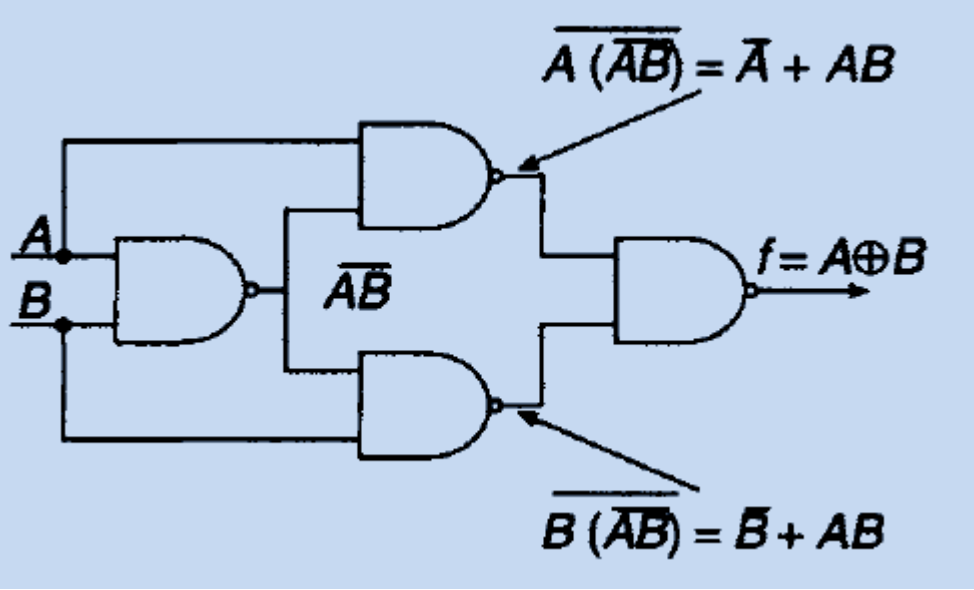
There is a one-to-one mapping between AND-OR network and NAND network

Often there is lot of further optimization that can be done

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

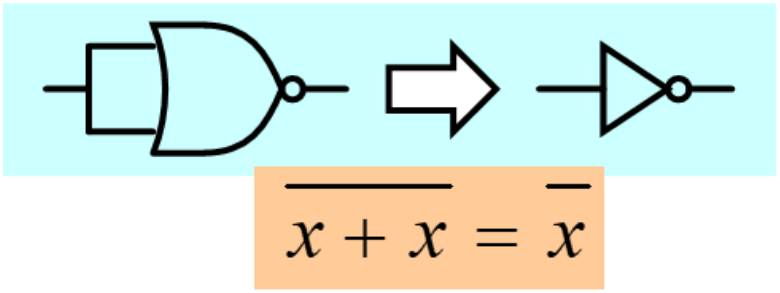




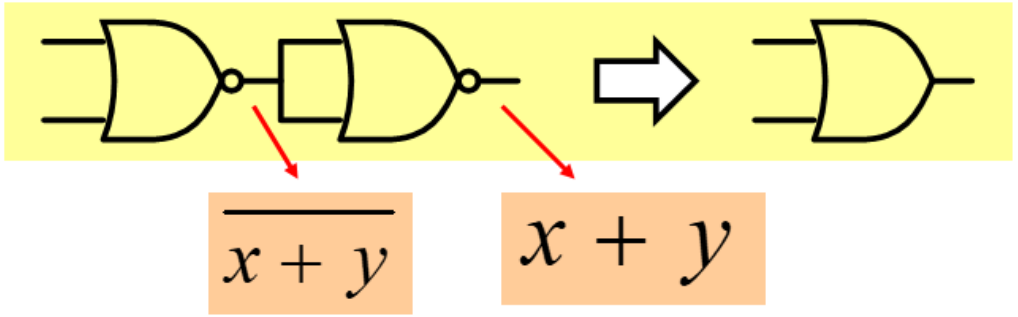


Implementation using only NOR gates (recap)

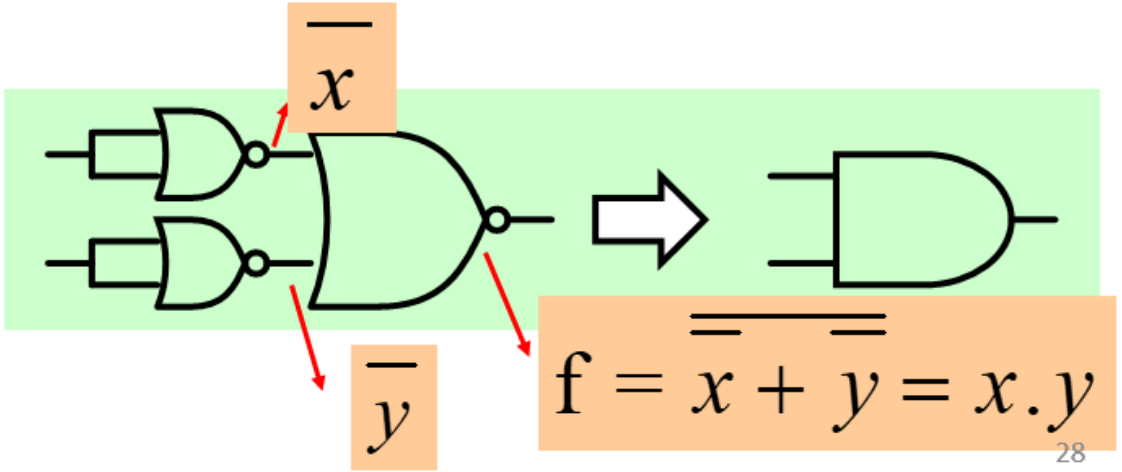
NOR to Inverter



NOR to OR



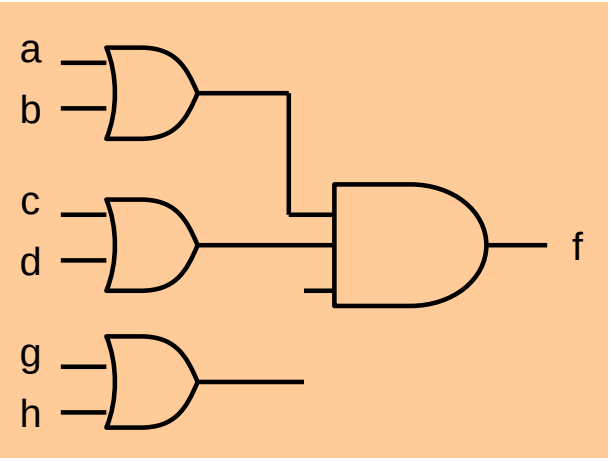
NOR to AND

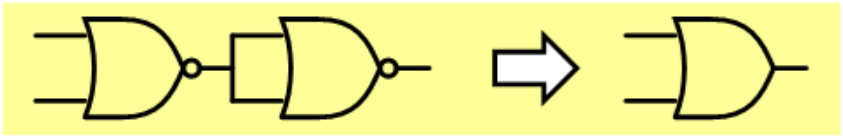


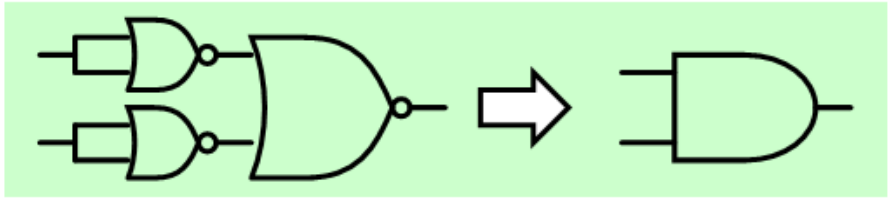
Implementation using only NOR gates (recap)

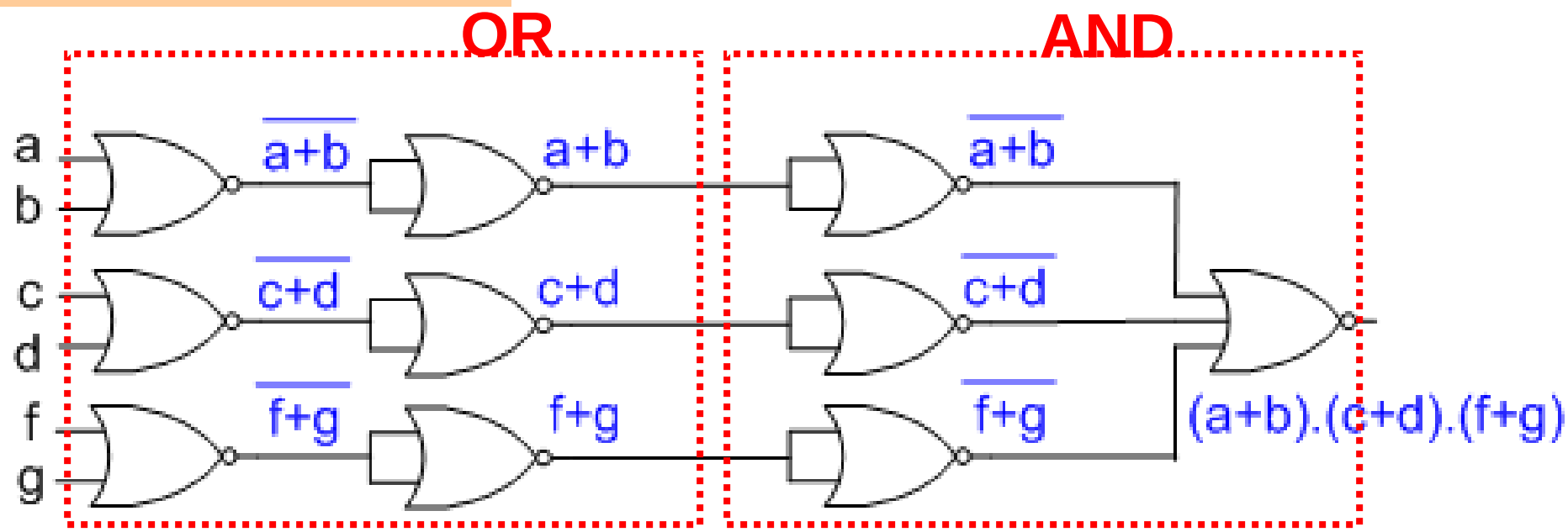
To implement using NOR gates, it is easiest to start with minimized Boolean expression in POS form

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

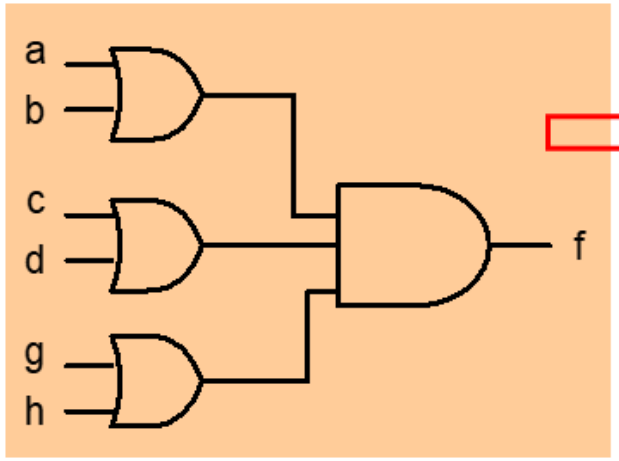


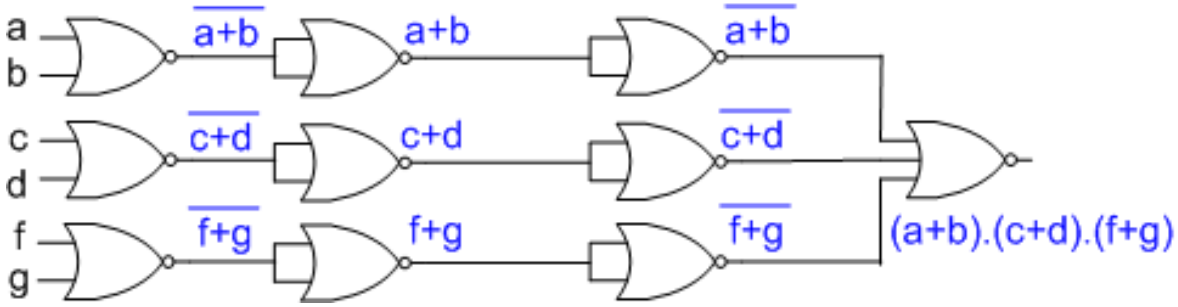


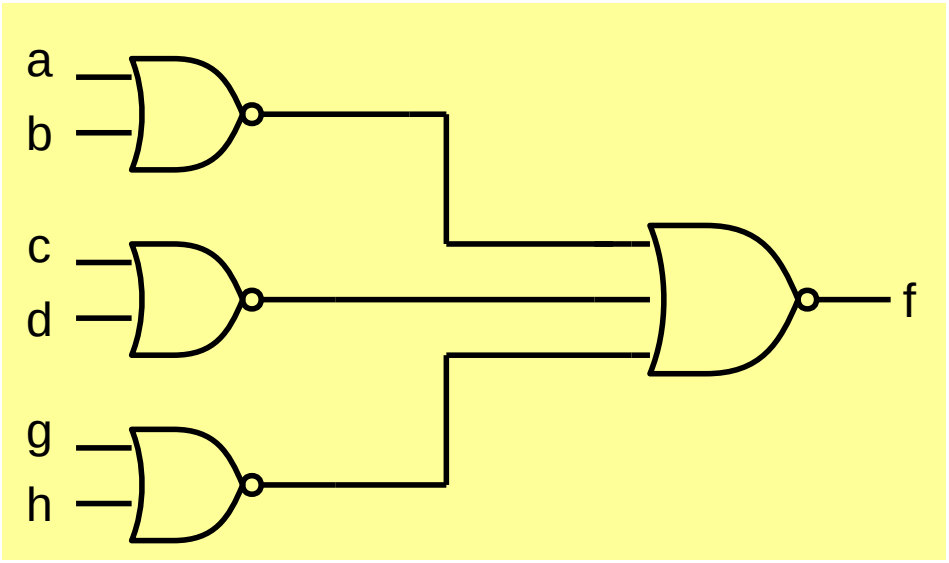




$$f = (a+b).(c+d).(f+g)$$



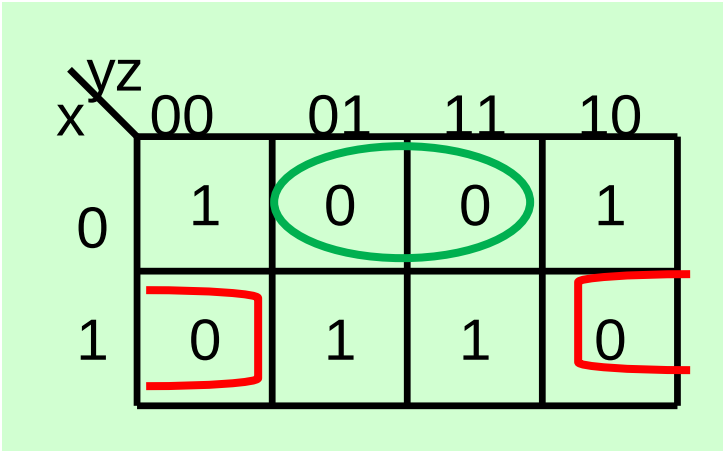




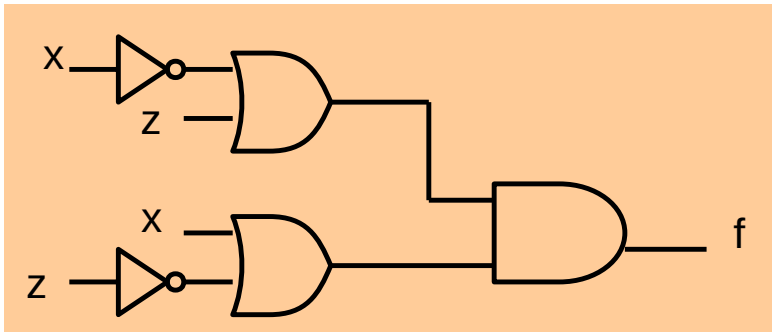
There is a one-to-one mapping between OR-AND network and NOR network

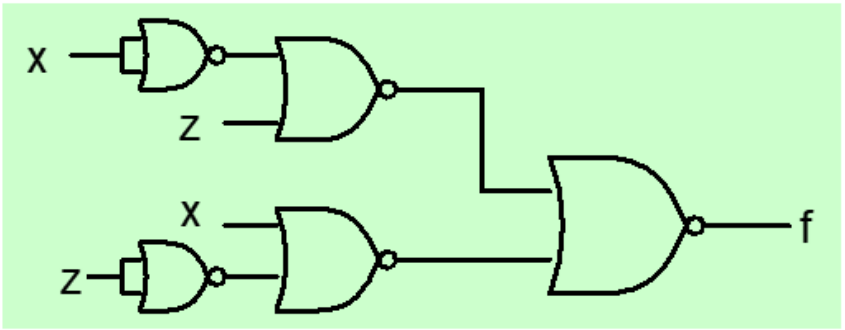
To implement SoP expression using NOR gates, determine first the corresponding PoS expression and then follow the procedure outlined earlier

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



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





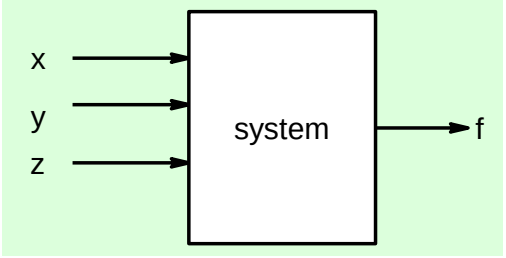
Similarly PoS expression can be implemented as NAND network by first converting it to SoP expression and then following the procedure outlined earlier

How do we get the chocolate?

Digital Circuits

Design Flow

System Description



Truth Table

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

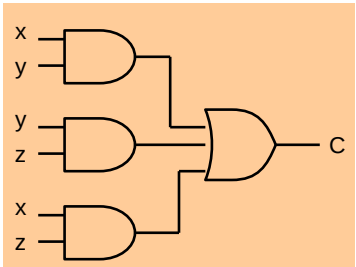
Boolean Expression

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

Minimized Boolean Expression

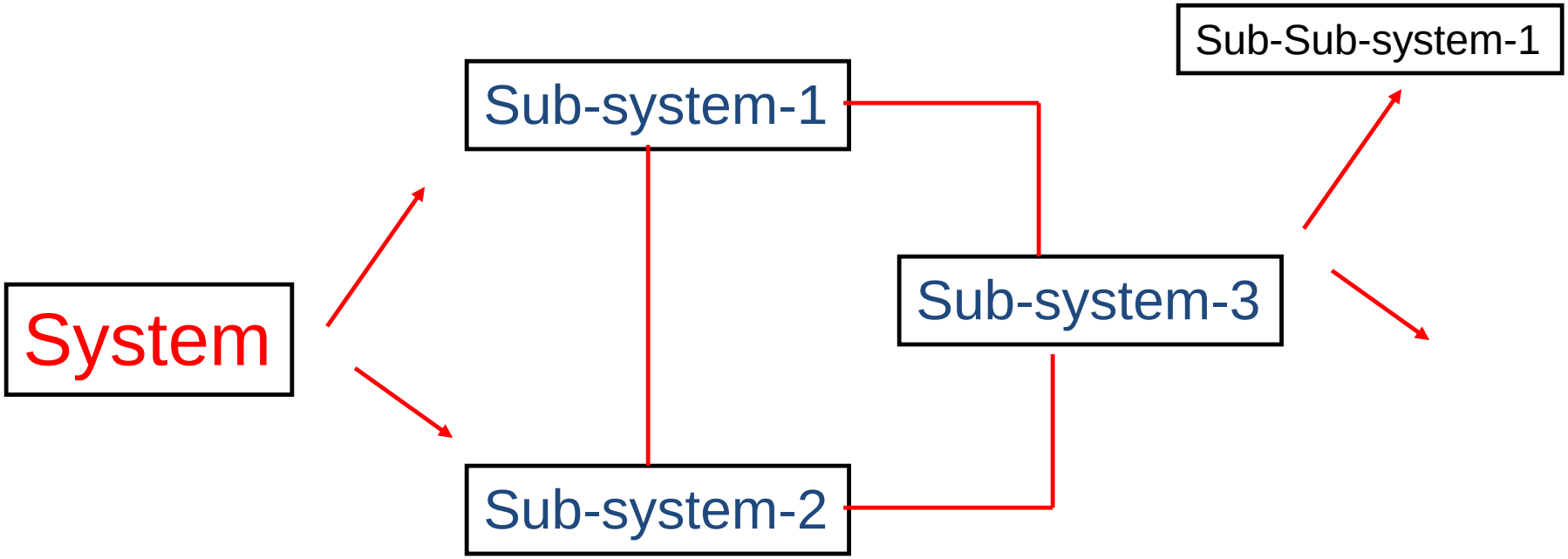
$$\Rightarrow f = \bar{x}.\bar{z} + x.z$$

Gate Netlist



This design approach becomes difficult to use

General Approach

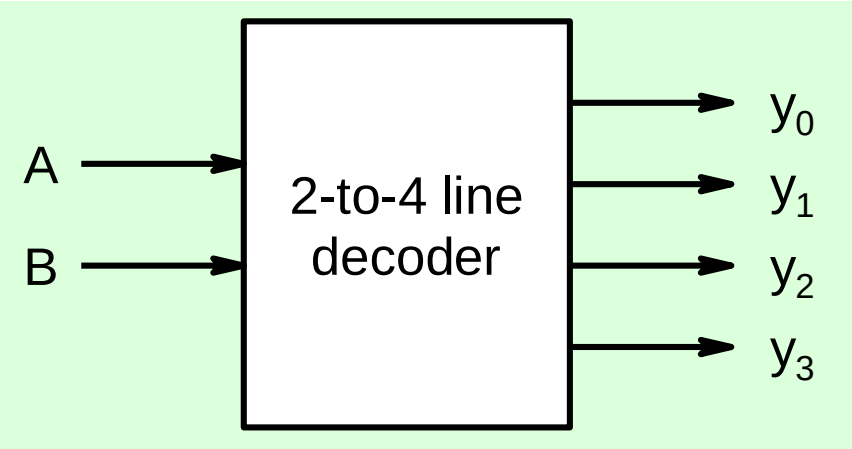


There are certain sub-systems or blocks that are used quite often such as :

1. **Decoders, Encoders**
2. **Multiplexers**
3. **Adder/Subtractors, Multipliers**
4. **Comparators**
5. **Parity Generators**
6. **……..**

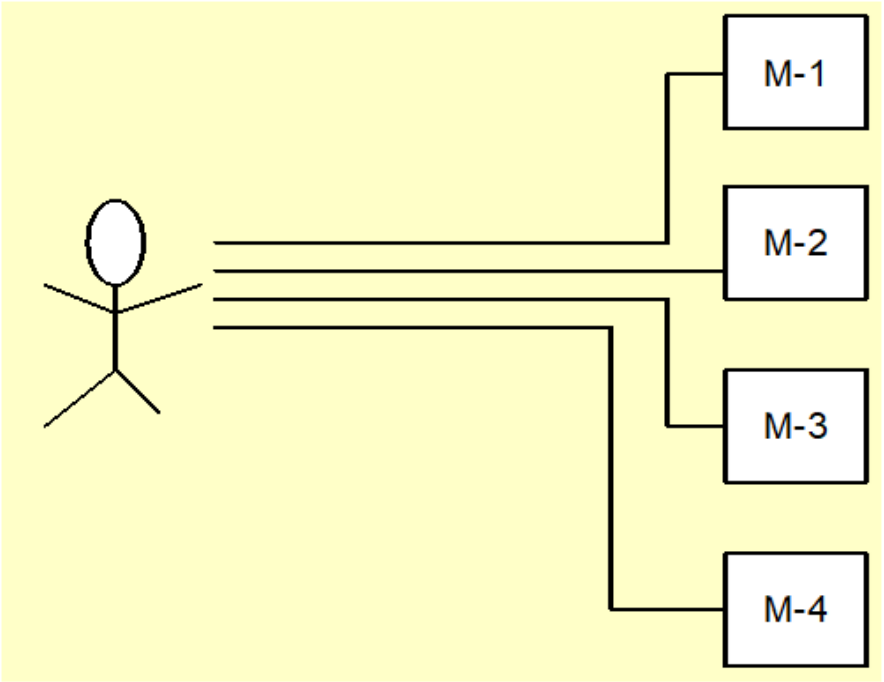
Decoders

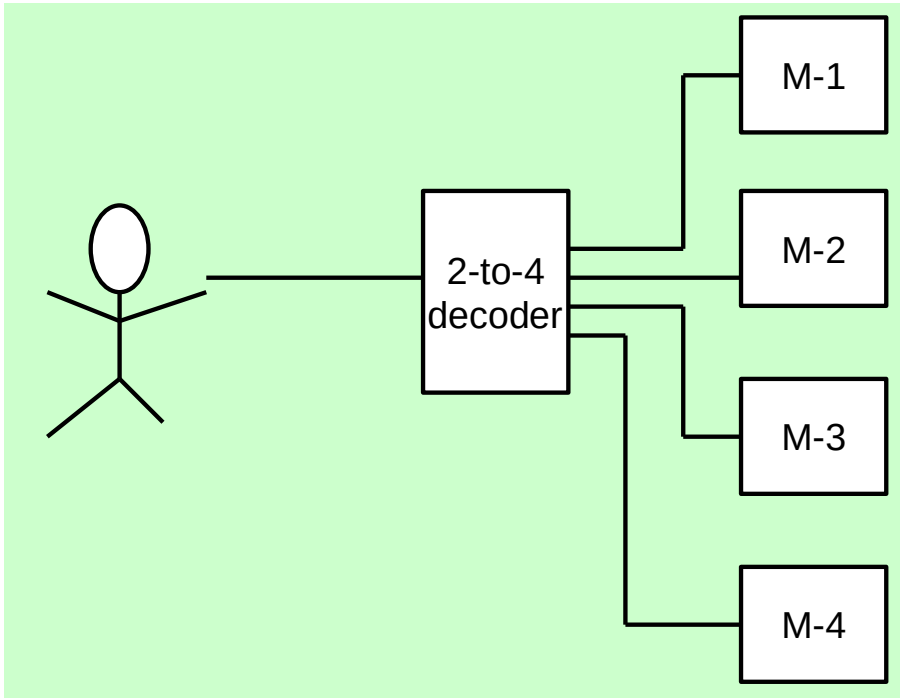
In general maps a smaller number of inputs to a larger set of outputs



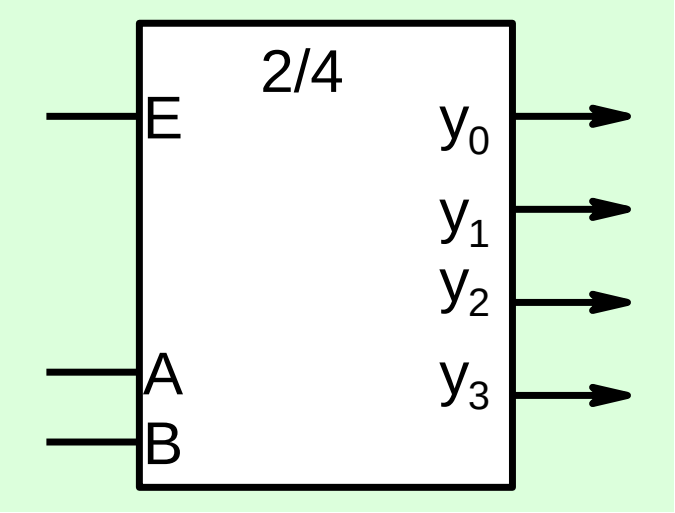
B	A	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Example

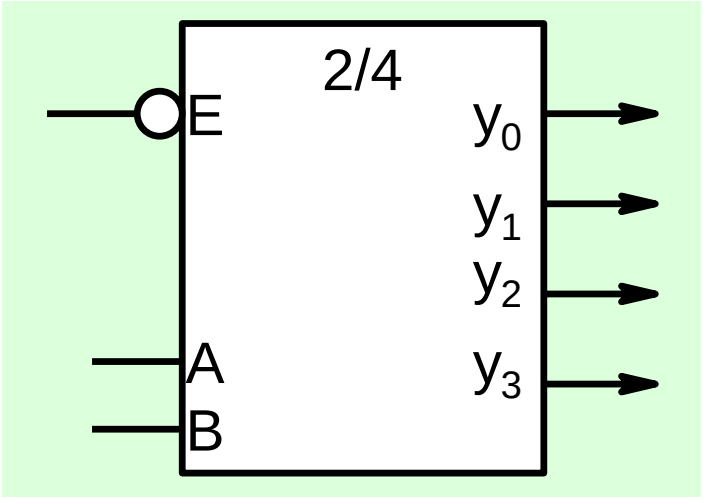




Decoder with Enable Input

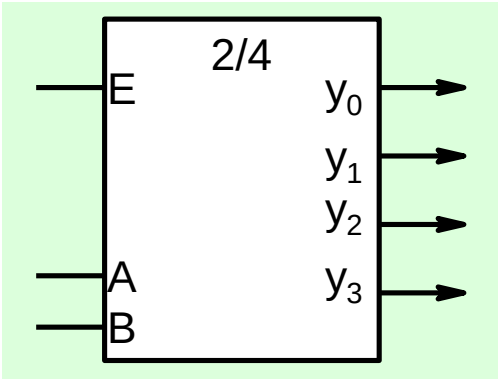


E	B	A	Y_0	Y_1	Y_2	Y_3
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1



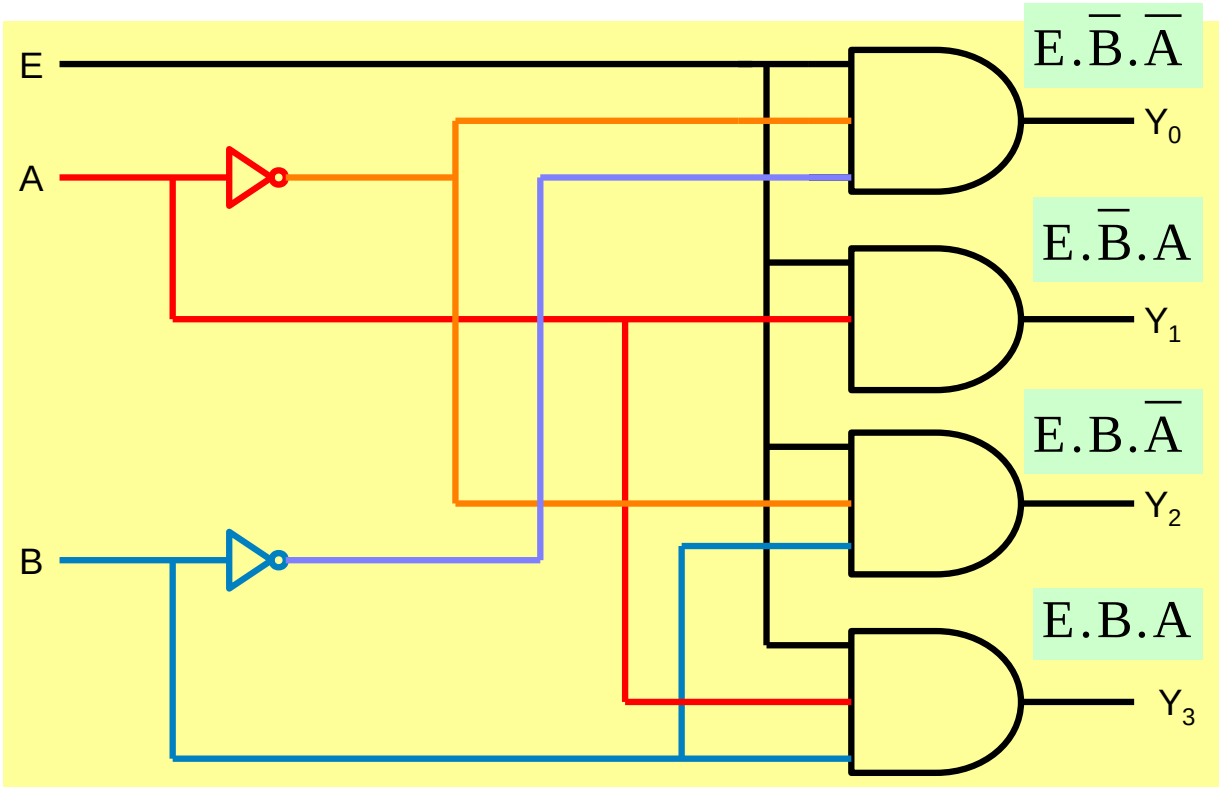
E	B	A	Y_0	Y_1	Y_2	Y_3
1	x	x	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1

Decoder: gate Implementation



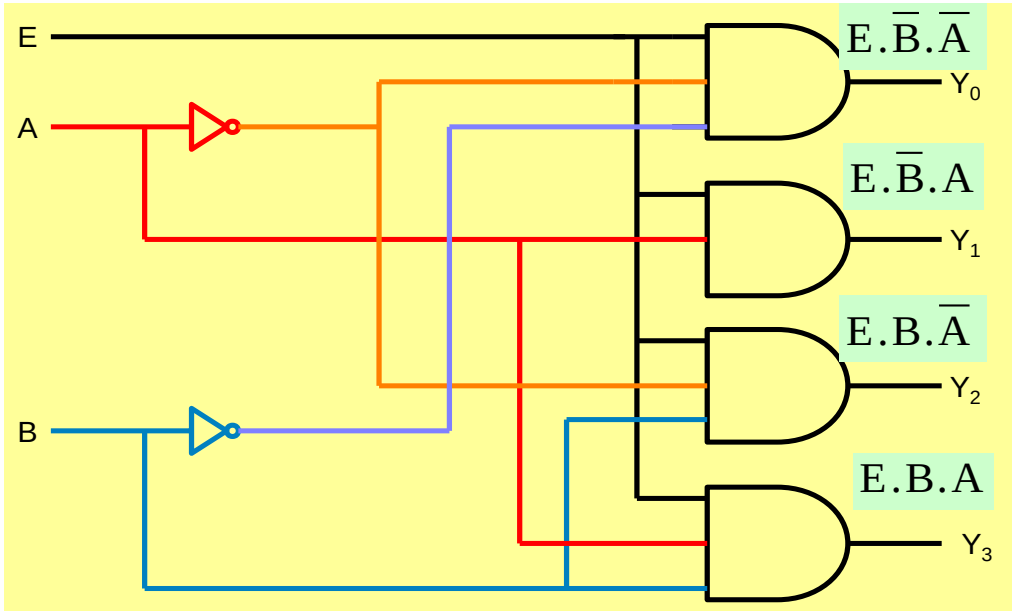
E	B	A	Y_0	Y_1	Y_2	Y_3
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

$$Y_0 = E.\overline{B}.\overline{A}\ ; Y_1 = E.\overline{B}.A ; Y_2 = E.B.\overline{A}\ ; Y_3 = E.B.A$$



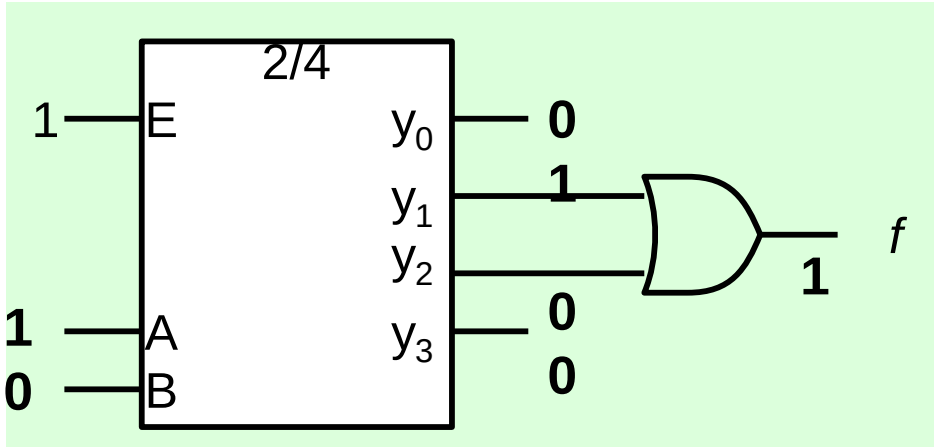
A n to 2^n decoder is a minterm generator

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



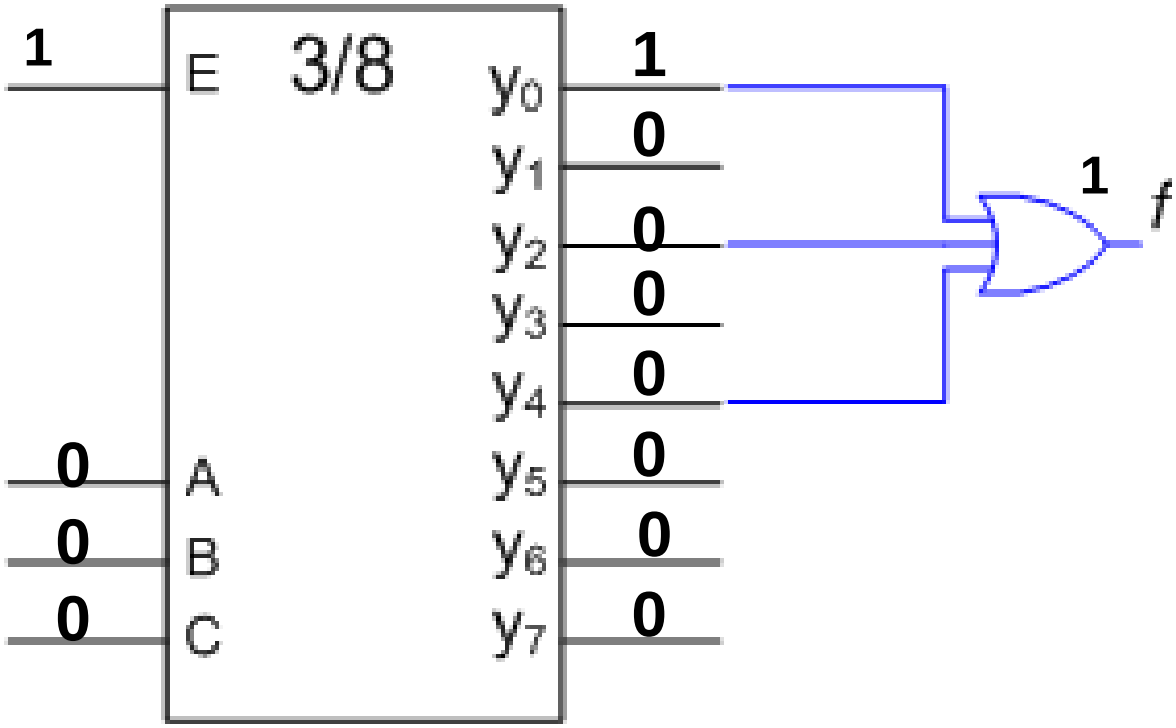
It can be used to implement any combinational circuit

B	A	f_1
0	0	0
0	1	1
1	0	1
1	1	0



Implementation of a 3-variable function with a 3-to-8 decoder

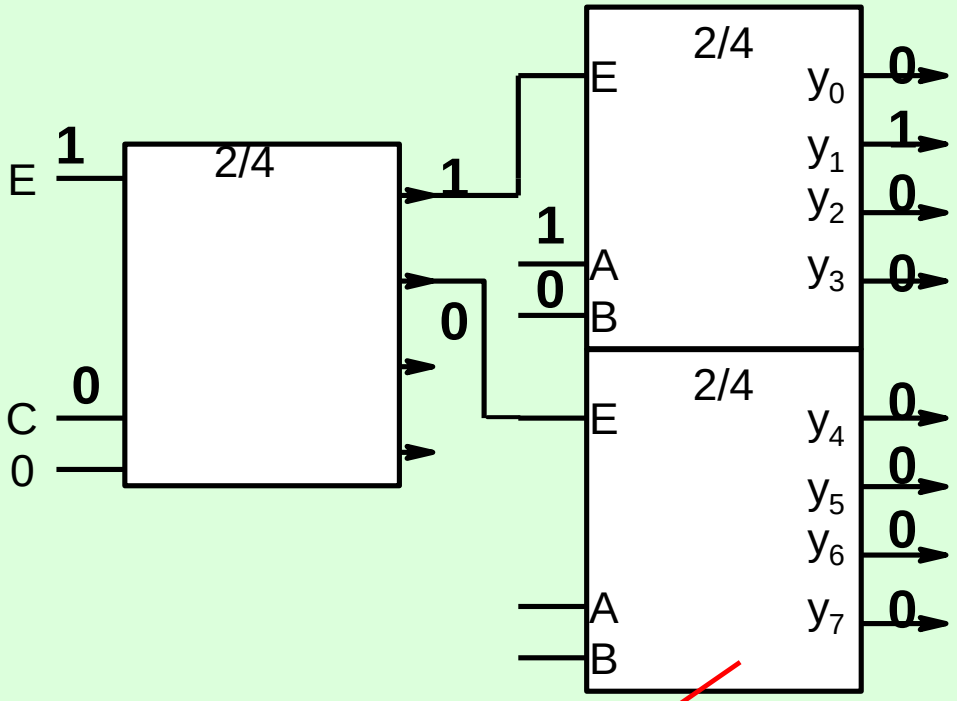
C	B	A	*f*
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0



Although it is easy to implement any combinational circuit with this method , it is often very inefficient in terms of gate utilization. Note that this method does not require any minimization.

3/8 decoder using 2/4 decoders

E	C	B	A	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1



E	B	A	Y_0	Y_1	Y_2	Y_3
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

How many 2/4 decoders are required to implement a 4/16 decoder ?