Module 3:Digital Electronics
Derived Gates

Quote of the day

"Never think you are nothing, never think you are everything, but think you are something and achieve anything ".

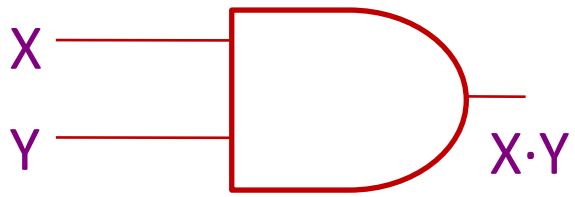
— Albert Einstein

Review basic gates

AND		
X	Y	X·Y
0	0	0
0	1	0
1	0	0
1	1	1

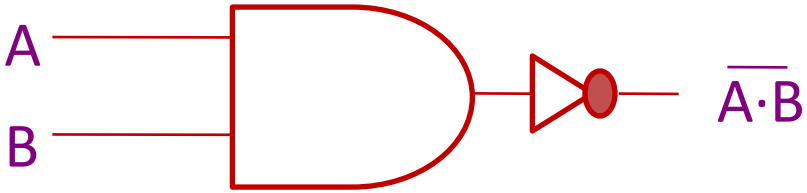
OR		
X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
Y	$\bar{Y}$
0	1
1	0

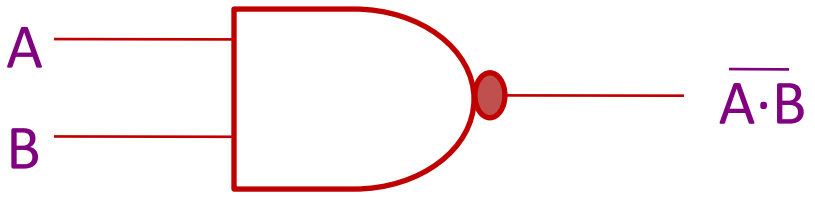


Derived gates

AND + NOT=NAND



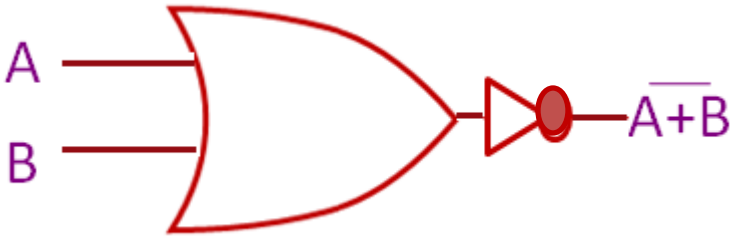
Logical Symbol



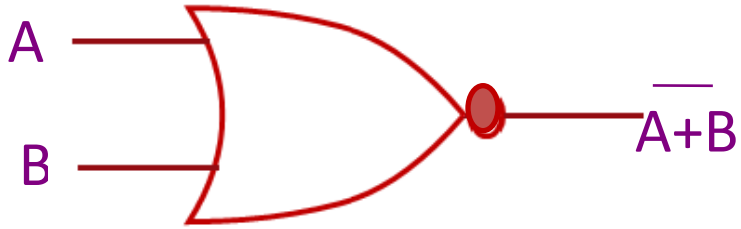
NAND		
A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

Derived gates

- OR + NOT=NOR



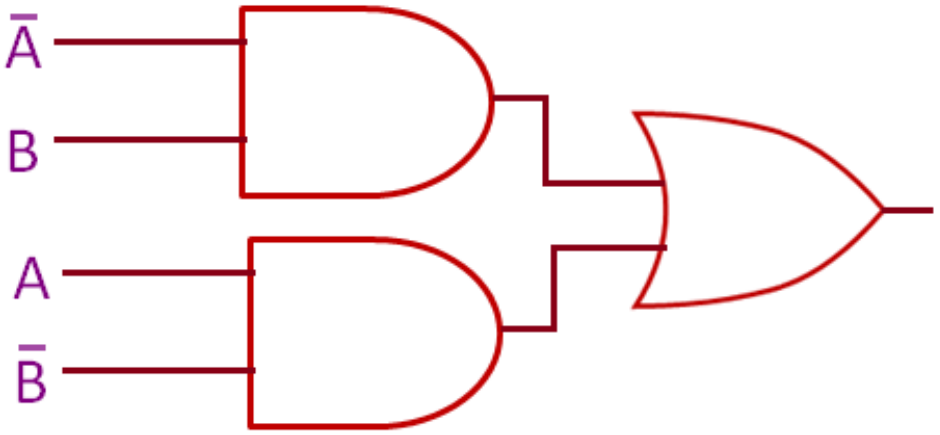
Logical Symbol



NOR		
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Derived gates

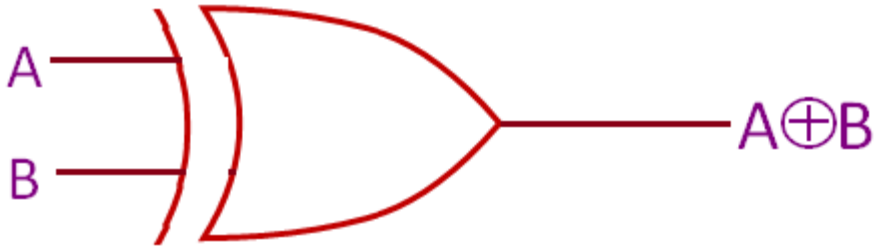
- Exclusive-OR →EX-OR



$\bar{A}B + A\bar{B}$

Boolean Expression for EX-OR

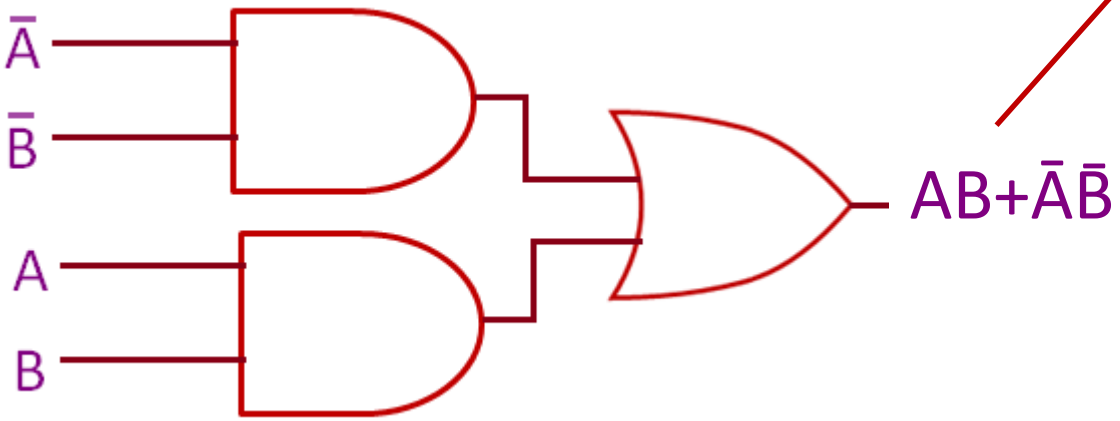
- Logical Symbol



EX-OR		
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

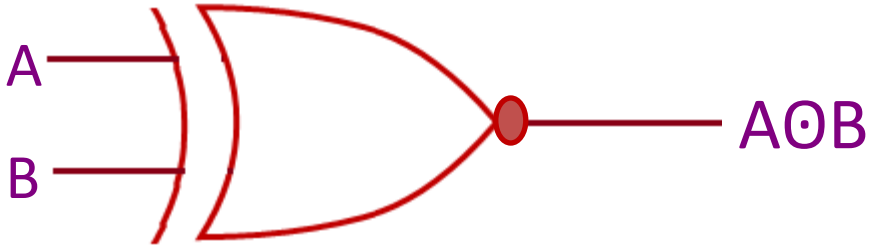
Derived gates

- Exclusive-NOR →EX-NOR



Boolean Expression for EX-NOR

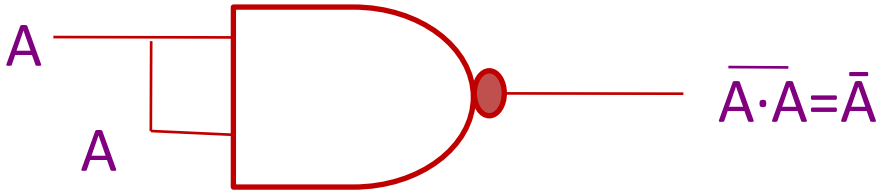
- Logical Symbol



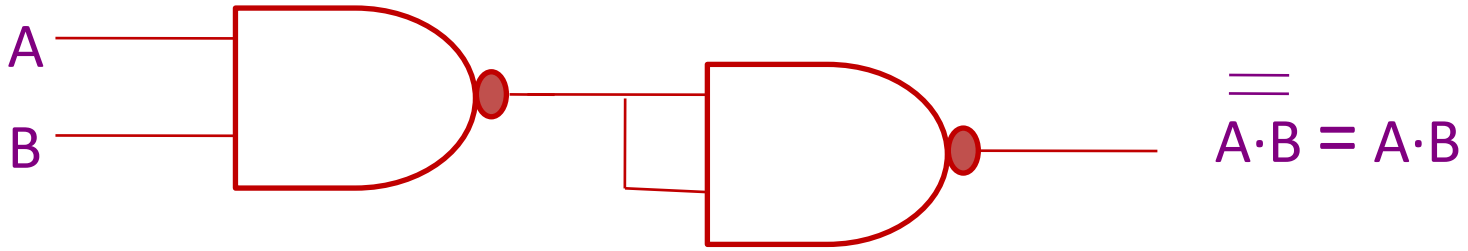
EX-NOR		
A	B	A⊙B
0	0	1
0	1	0
1	0	0
1	1	1

Universal gates

- NAND and NOR gates are called as universal gates because we can implement all gates using these gates.
- NAND as Universal gates
- NOT using NAND

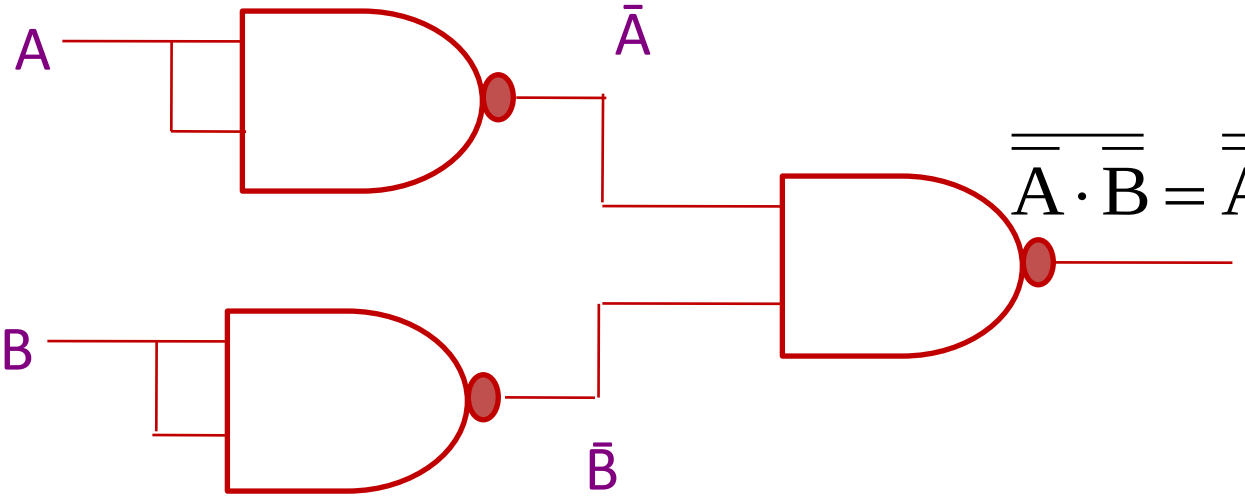


- AND using NAND



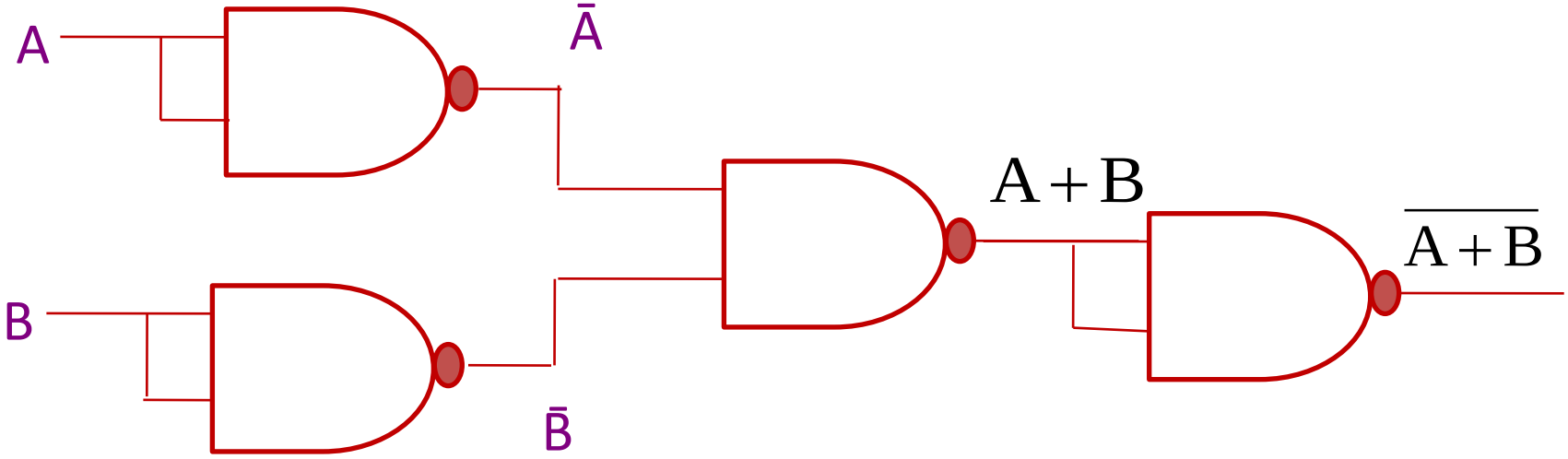
NAND as Universal gates

- OR using NAND

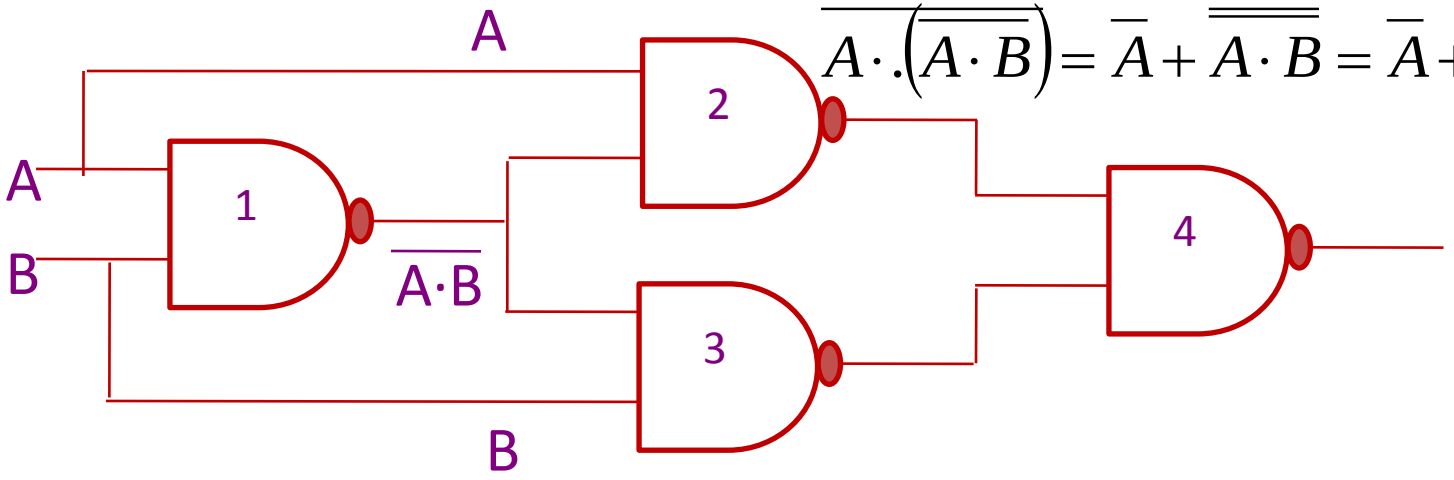


$$\overline{\bar{A}\cdot\bar{B}} = \bar{\bar{A}} + \bar{\bar{B}} = A + B$$

- NOR using NAND



EX-OR using NAND



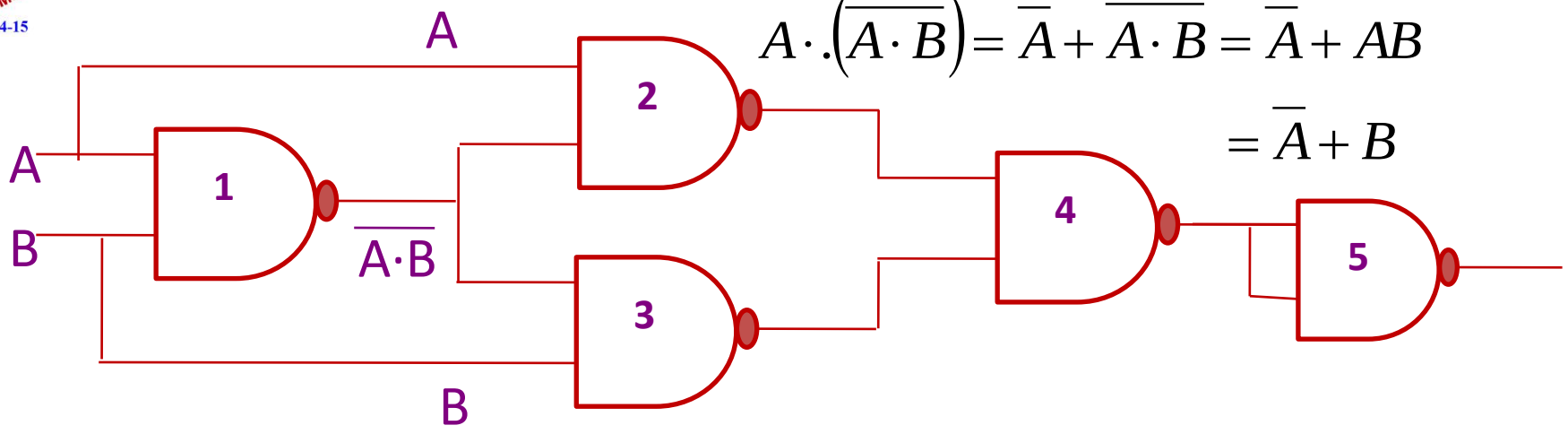
$$\overline{A\cdot\overline{(A\cdot B)}}=\overline{A}+\overline{\overline{A\cdot B}}=\overline{A}+AB$$

$$\overline{B\cdot\overline{(A\cdot B)}}=\overline{B}+\overline{\overline{A\cdot B}}=\overline{B}+AB$$

OUTPUT OF GATE 4:

$$\overline{(\overline{B}+A)}\cdot\overline{(\overline{A}+B)}=\overline{(\overline{B}+A)}+\overline{(\overline{A}+B)}=\overline{\overline{B}}\cdot\overline{A}+\overline{\overline{A}}\cdot\overline{B}=\overline{A}B+A\overline{B}$$

EX-NOR using NAND



$$\overline{A\cdot\left(\overline{A\cdot B}\right)}=\overline{A}+\overline{\overline{A\cdot B}}=\overline{A}+AB$$

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

$$\overline{B\cdot\left(\overline{A\cdot B}\right)}=\overline{B}+\overline{\overline{A\cdot B}}=\overline{B}+AB$$

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

OUTPUT OF GATE 4:

$$\left(\overline{\overline{B}+A}\right)\cdot\left(\overline{\overline{A}+B}\right)=\left(\overline{\overline{B}+A}\right)+\left(\overline{\overline{A}+B}\right)=\overline{\overline{B}}\cdot\overline{A}+\overline{\overline{A}}\cdot\overline{B}=\overline{A}B+A\overline{B}$$

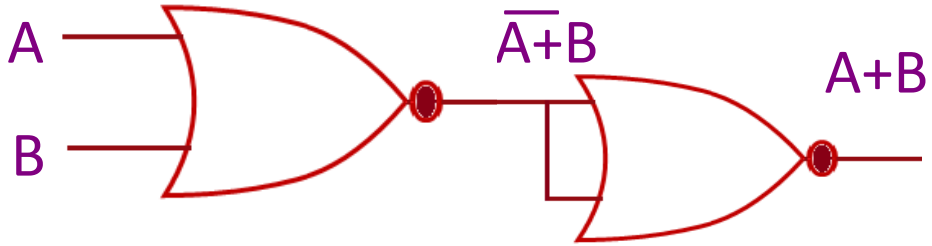
OUTPUT OF GATE 5:

$$\left(\overline{\overline{A}B+A\overline{B}}\right)=\left(\overline{\overline{A}B}\right)\cdot\left(\overline{A\overline{B}}\right)=\left(A+\overline{B}\right)\cdot\left(\overline{A}+B\right)$$

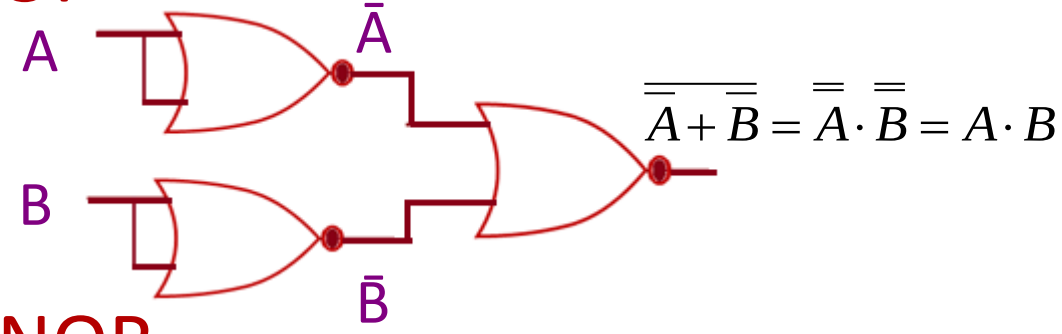
NOR as universal gate

- NOT using NOR

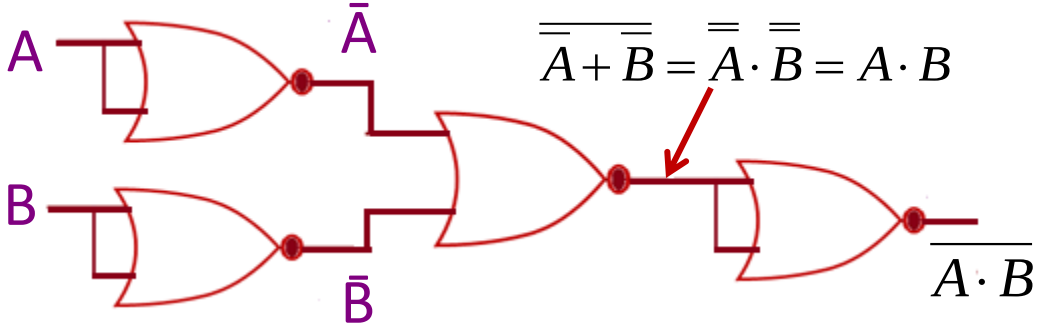
- OR using NOR



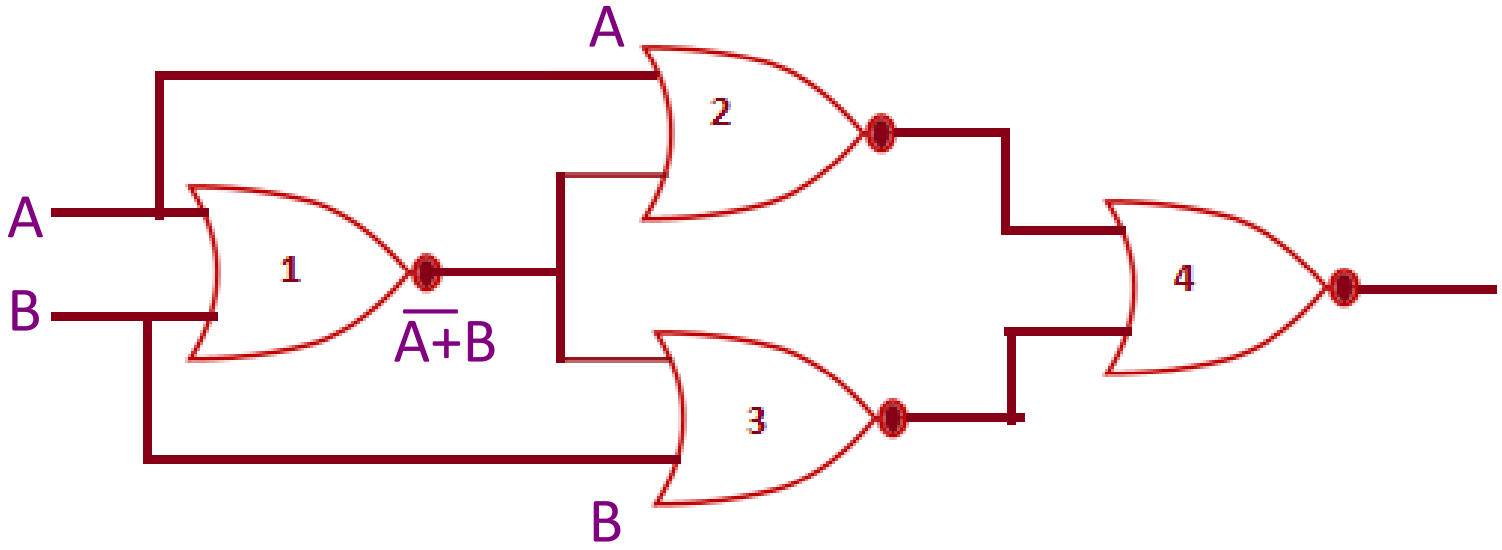
- AND using NOR



- NAND using NOR



EX-NOR using NOR



OUTPUT OF GATE 2:

$$\overline{A+.\left(\overline{A+B}\right)}=\overline{A}\cdot\overline{\overline{A+B}}=\overline{A}\cdot\left(A+B\right)=\overline{A}\cdot A+\overline{A}\cdot B=\overline{A}\cdot B$$

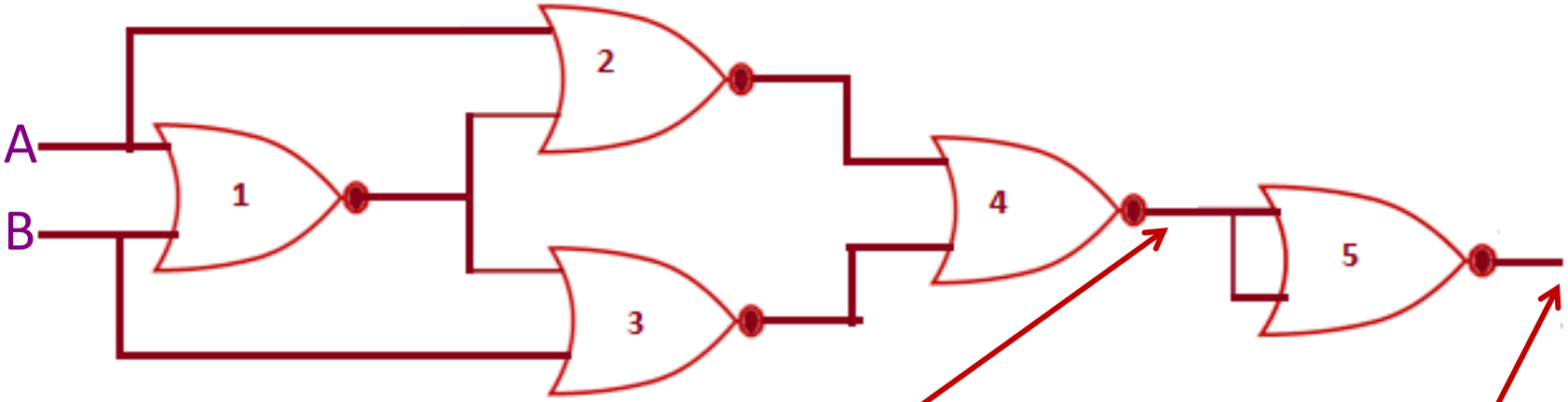
OUTPUT OF GATE 3:

$$\overline{B+.\left(\overline{A+B}\right)}=\overline{B}\cdot\overline{\overline{A+B}}=\overline{B}\cdot\left(A+B\right)=\overline{B}\cdot A+\overline{B}\cdot B=A\cdot\overline{B}$$

OUTPUT OF GATE 4:

$$\left(\overline{\overline{A}\cdot B+A\cdot\overline{B}}\right)=\left(\overline{\overline{A}\cdot B}\right)\cdot\left(\overline{A\cdot\overline{B}}\right)=\left(\overline{\overline{A}}+\overline{B}\right)\cdot\left(\overline{A}+\overline{\overline{B}}\right)=\left(A+\overline{B}\right)\cdot\left(\overline{A}+B\right)$$

EX-OR using NOR



- Output of Gate 4

$$(A+\overline{B})\cdot(\overline{A}+B)$$

- Output Gate of 5

$$\overline{(A+\overline{B})\cdot(\overline{A}+B)}=\overline{(A+\overline{B})}+\overline{(\overline{A}+B)}=\overline{A}\overline{\overline{B}}+\overline{\overline{A}}\overline{B}=\overline{A}B+A\overline{B}$$

Half adder using basic gates

- Half adder is a circuit which adds two one bit binary data.
- The result of half adder generates sum(S) and carry (C).
- Observe the truth table.
- $S=A \oplus B$ and $C=A \cdot B$,
- $S=\bar{A} \cdot B + A \cdot \bar{B}$

Half Adder			
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

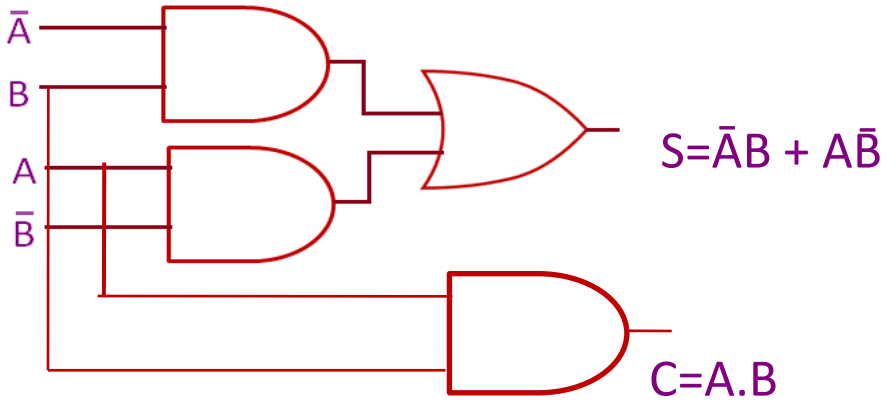
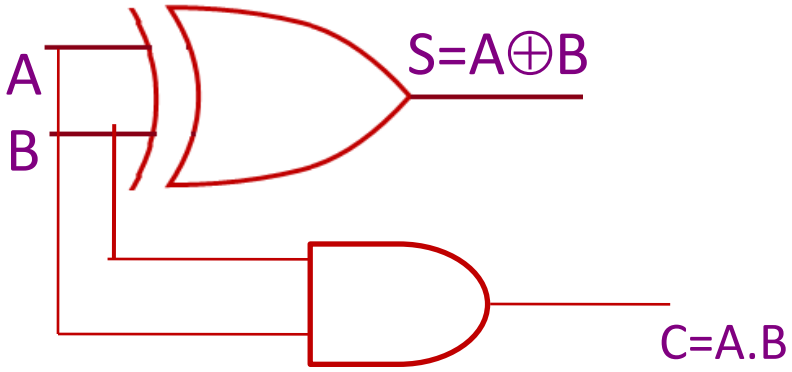


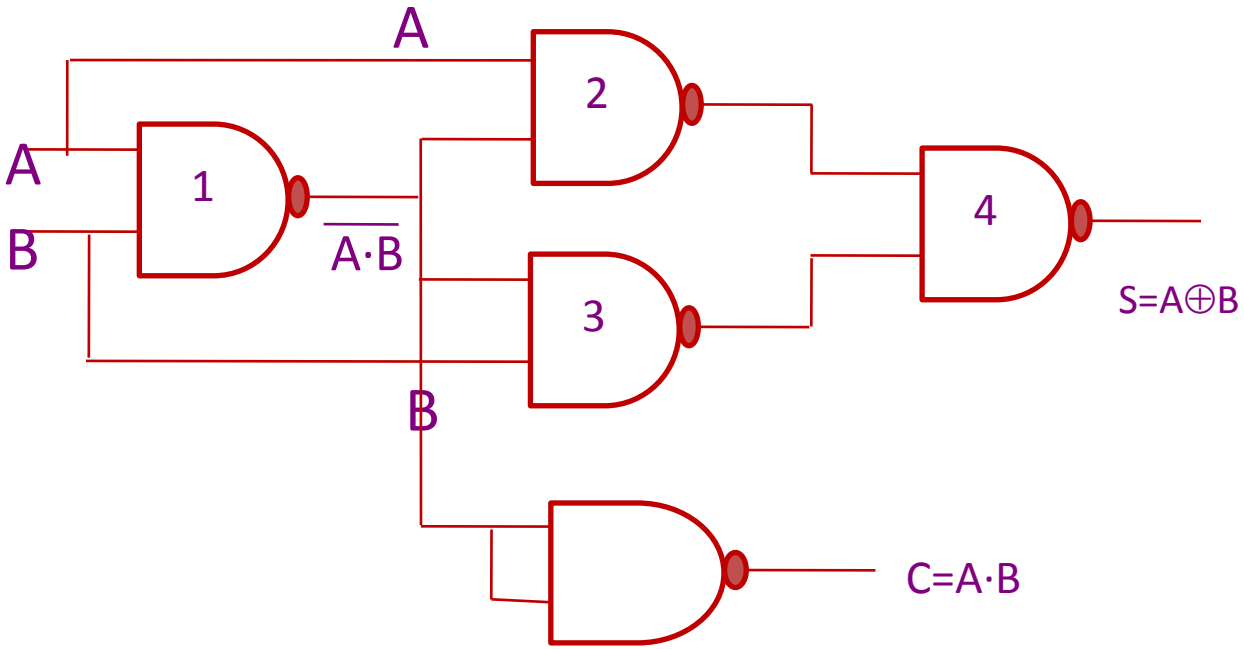
Fig. Logic Diagram using basic gates

Half adder using Logic gates

- Use EX-OR gate for sum and AND gate to generate carry



- Half adder using NAND



- Half adder does not add the carry generated from previous stage.

Full Adder

- Full adder is the circuit which adds two bits along with carry in(C_{in}) and generates Sum(S) and carry out (C_{out}).
- From Truth Table Writing equation for S and C_{out}.
- $S=\bar{A}\bar{B}C_{in}+\bar{A}B\bar{C}_{in}+A\bar{B}\bar{C}_{in}+ABC_{in}$
- $S=(\bar{A}\bar{B}+AB)C_{in}+(\bar{A}B+A\bar{B})\bar{C}_{in}$
- $S=(A \odot B)C_{in}+(A \oplus B)\,\bar{C}_{in}$

$$S=(A \oplus B) \oplus C_{in}$$

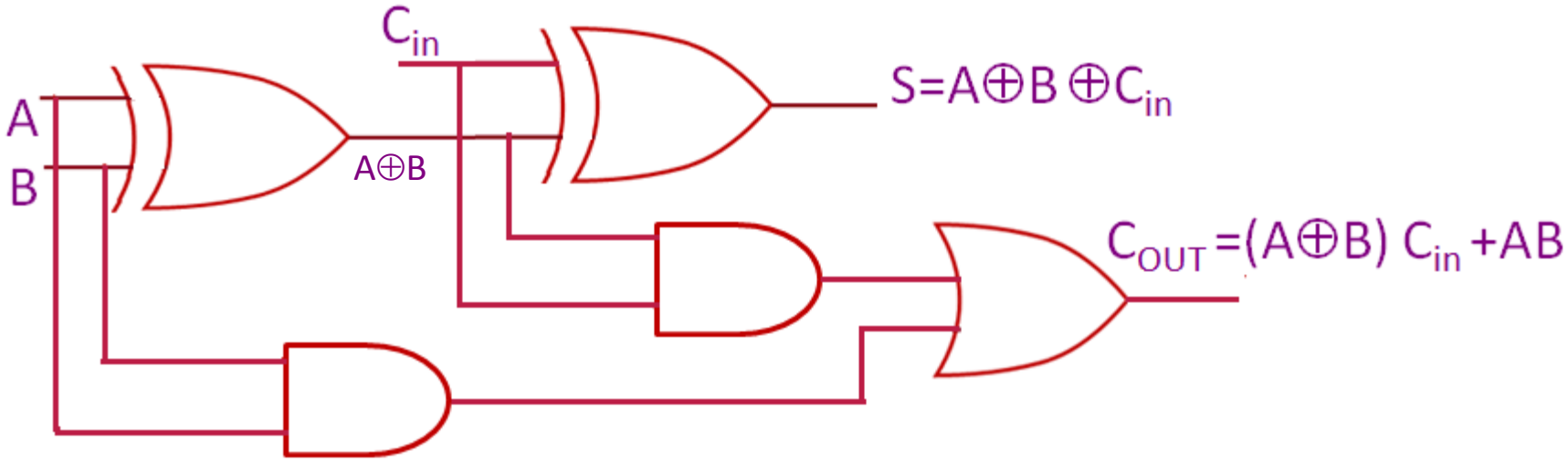
Now writing for C_{out}

- $C_{out}=\bar{A}BC_{in}+A\bar{B}C_{in}+AB\bar{C}_{in}+ABC_{in}$
- $C_{out}=\bar{A}BC_{in}+A\bar{B}C_{in}+AB(\bar{C}_{in}+C_{in})$
- $C_{out}=(\bar{A}B+A\bar{B})C_{in}+AB$

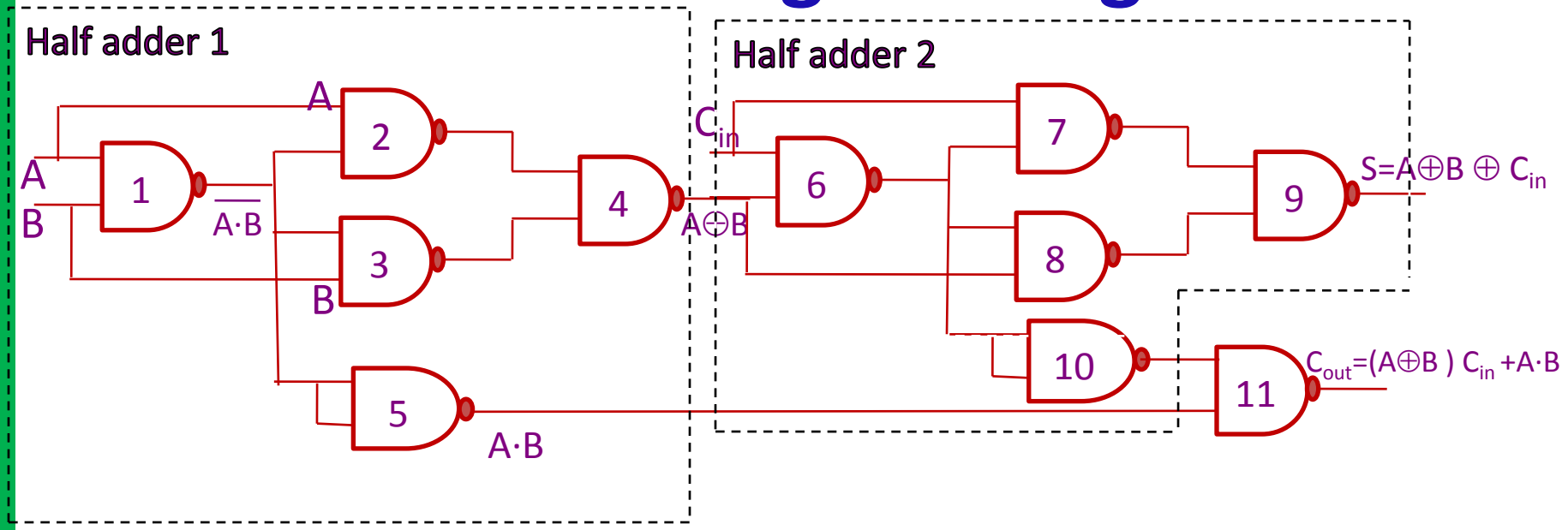
$$C_{out}=(A \oplus B)\,C_{in}+AB$$

Full Adder				
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full adder using Logic gates



Full adder using NAND gates



Reduction of carry out expression using Boolean algebra

Consider the expression for carry out.

$$C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB(\bar{C}_{in} + C_{in})$$

$$C_{out} = \bar{A}BC_{in} + A(\bar{B}C_{in} + B)$$

$$C_{out} = \bar{A}BC_{in} + A(C_{in} + B)$$

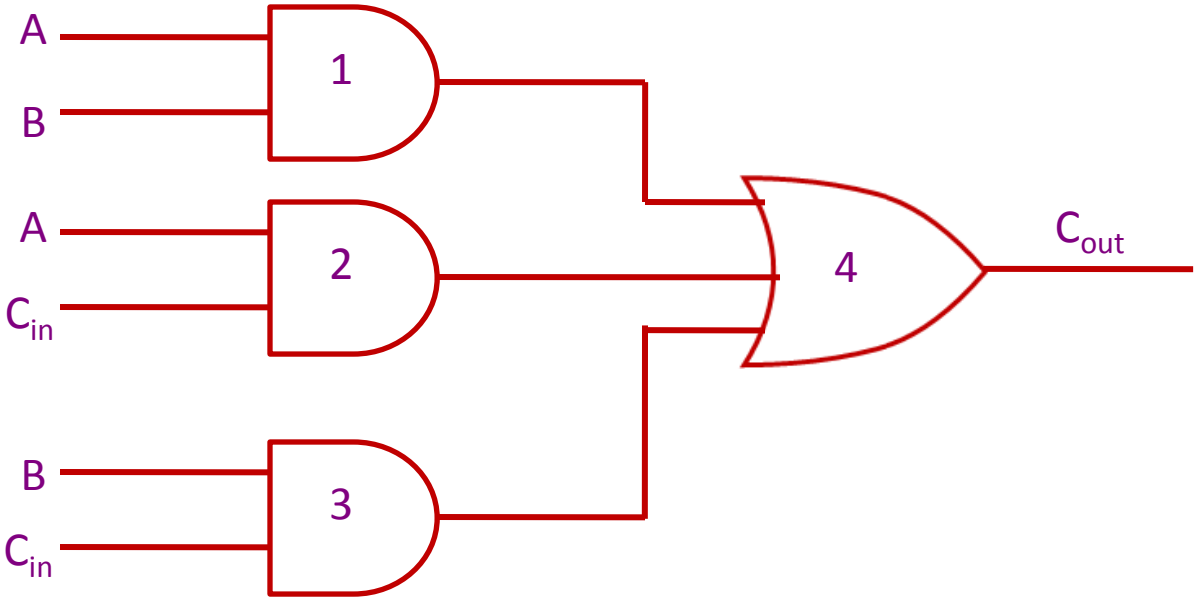
$$C_{out} = (\bar{A}B + A)C_{in} + AB = (B + A)C_{in} + AB$$

$$C_{out} = AB + Ac_{in} + Bc_{in}$$

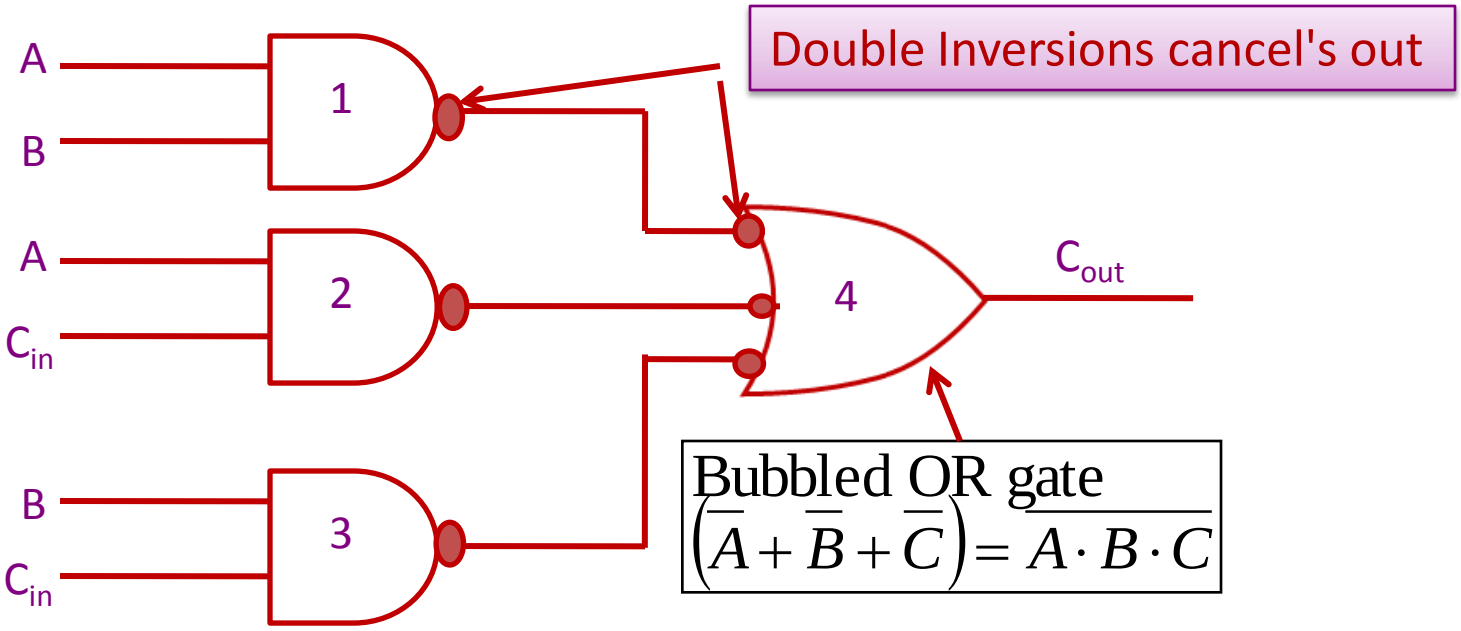
- This expression consists of product terms which are summed together to form Boolean expression. Such expressions are referred as sum of product form(SOP) equations.
- Now let us implement this expression using basic gates(AND-OR-NOT).

Implementation of SOP equation using Basic Gates

- $C_{out} = AB + Ac_{in} + Bc_{in}$
- To Implement this expression using basic gates we require three AND gates and one OR gate.

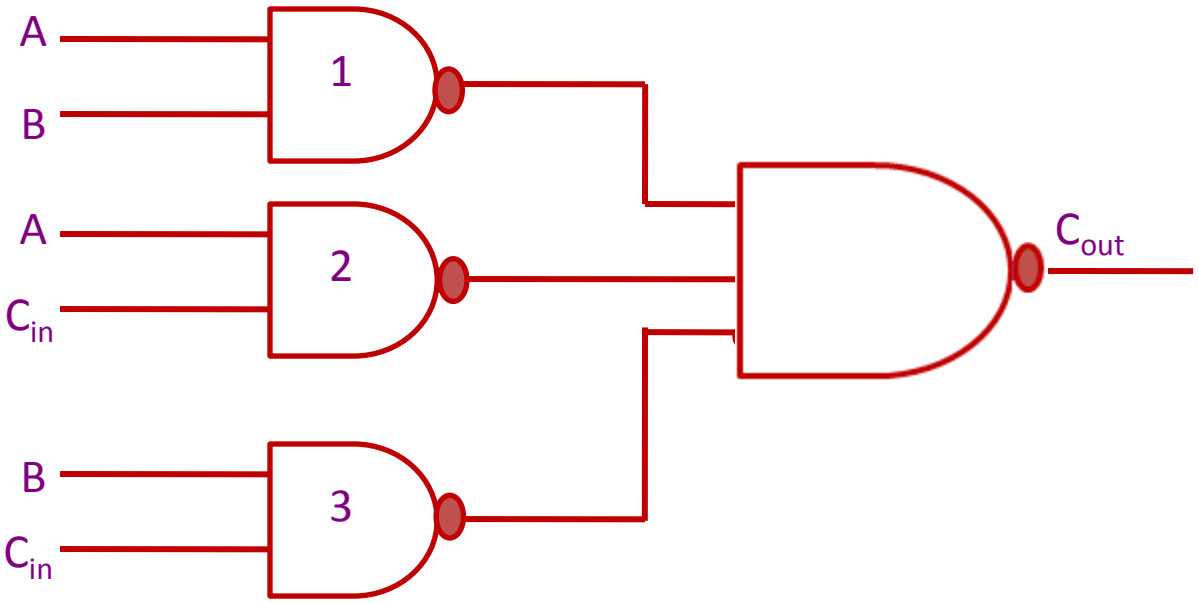


Implementation of SOP equation using NAND Gates



Bubbled OR gate is equivalent to NAND gate, So Replace

Implementation of SOP equation using NAND Gates



Bubbled OR gate is equivalent to NAND gate, So Replace