Chapter-03

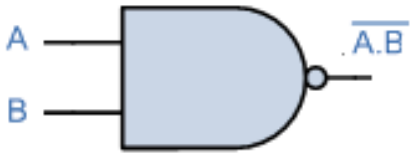
GATE LEVEL MINIMIZATION

NAND
and
NOR
IMPLEMENTATION

Universal Gates

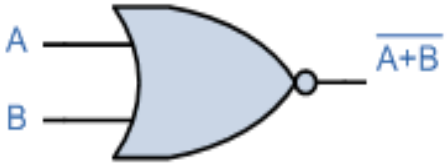
- A universal gate is a gate that can implement any Boolean function without the use of any other gate .
- The NAND and NOR gates are universal gates.
- The NAND and NOR gates are said to be *universal* gates because any logic circuit can be implemented with it.

NAND Gate



INPUT		OUTPUT
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

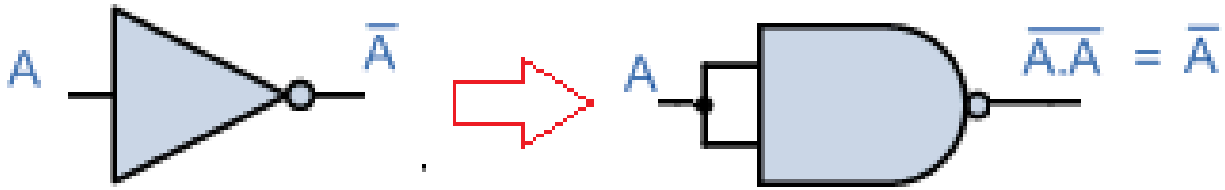
NOR Gate



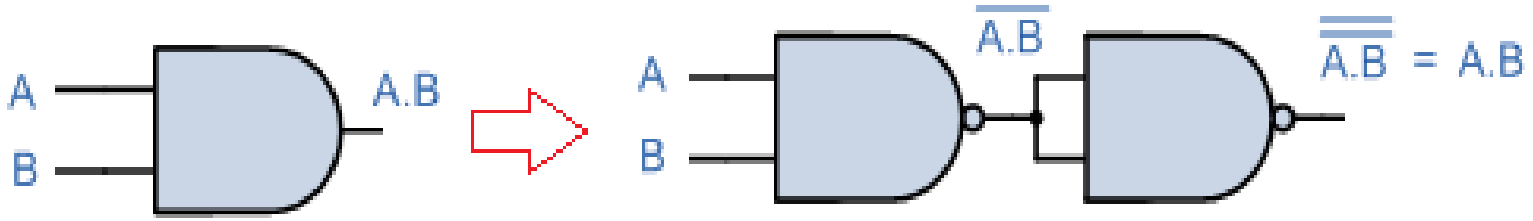
INPUT		OUTPUT
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

Logic Gates using only NAND Gates

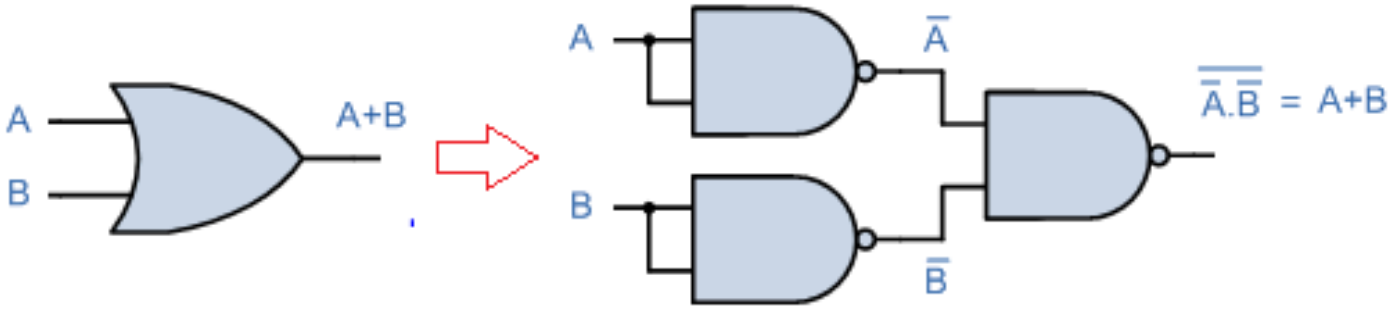
- NOT Gate



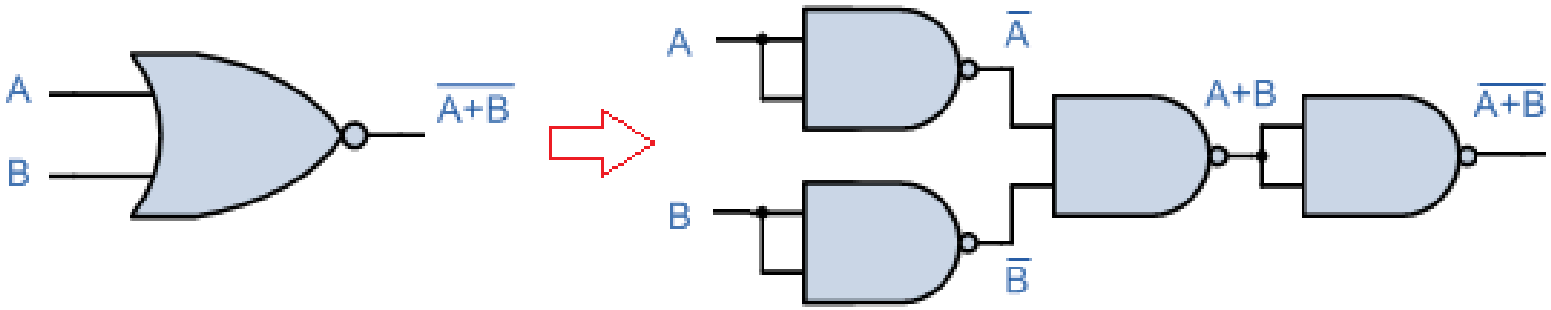
- AND Gate



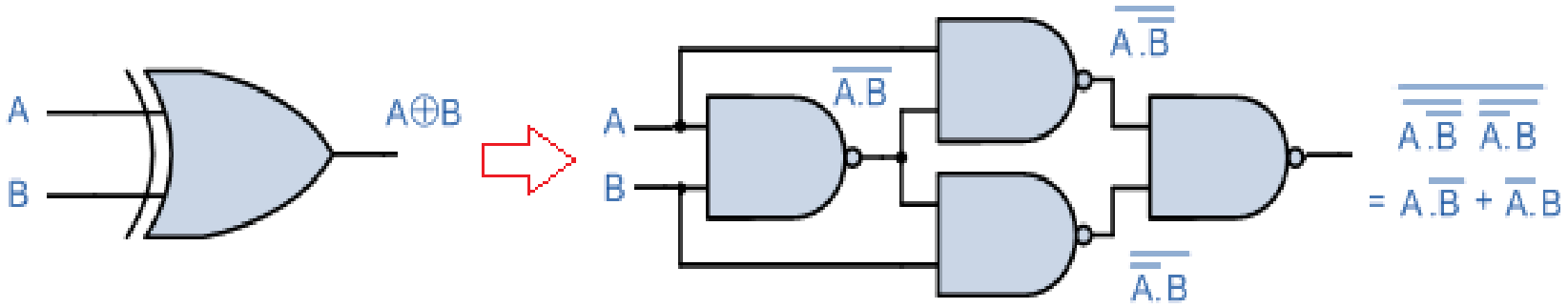
- OR Gate



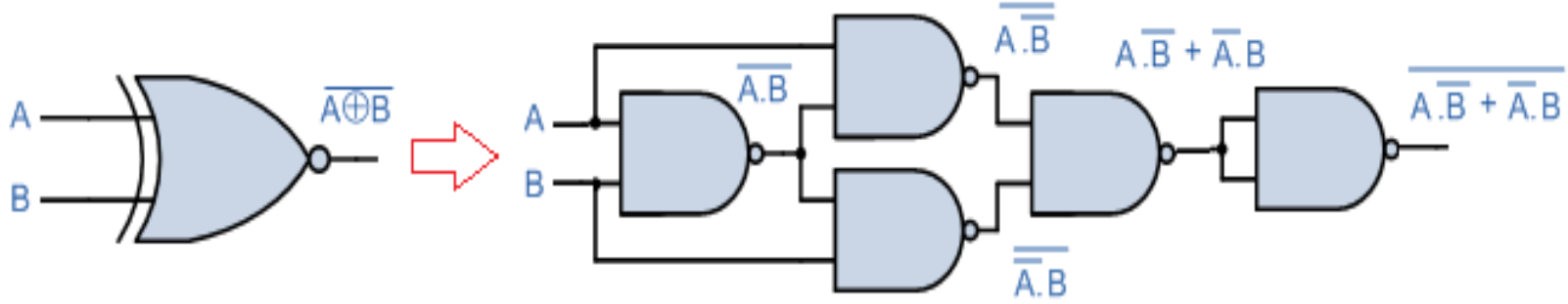
- NOR Gate



- EX-OR Gate

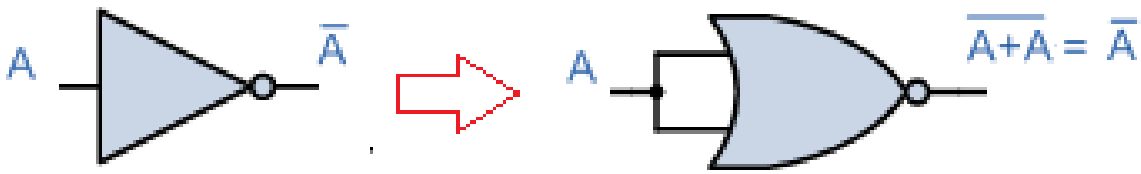


- EX-NOR Gate

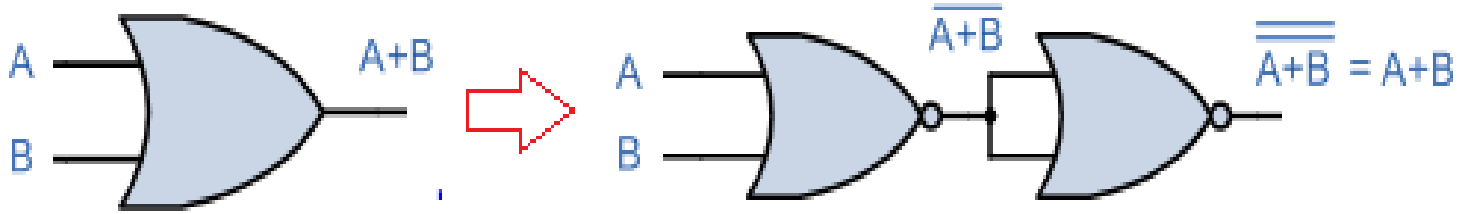


Logic Gates using only NOR Gates

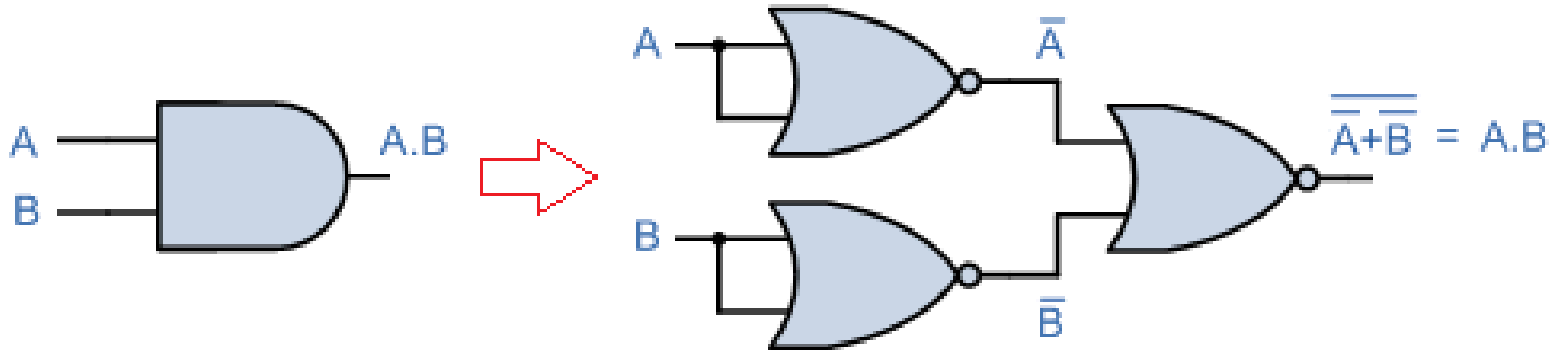
- NOT Gate



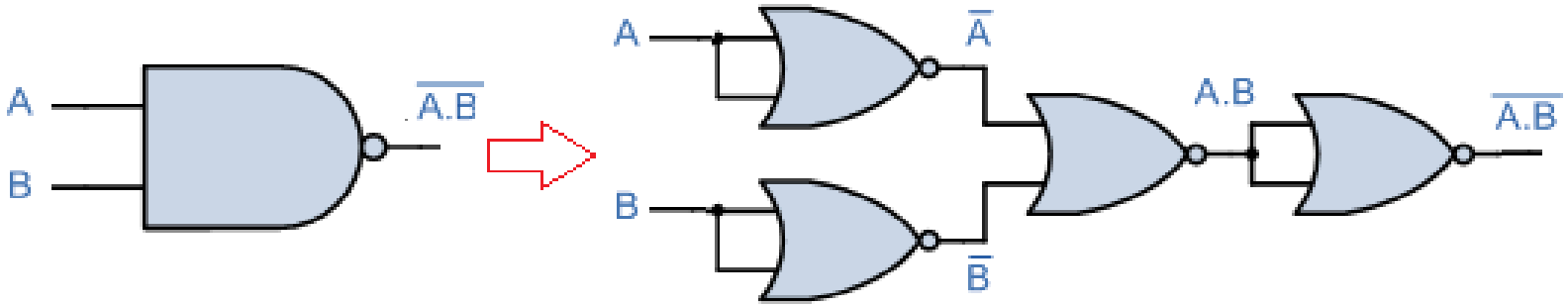
- OR Gate



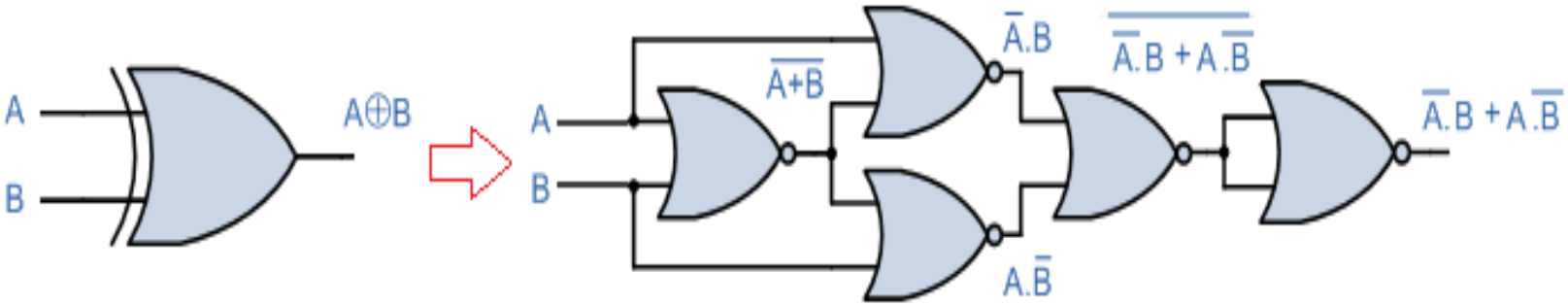
- AND Gate



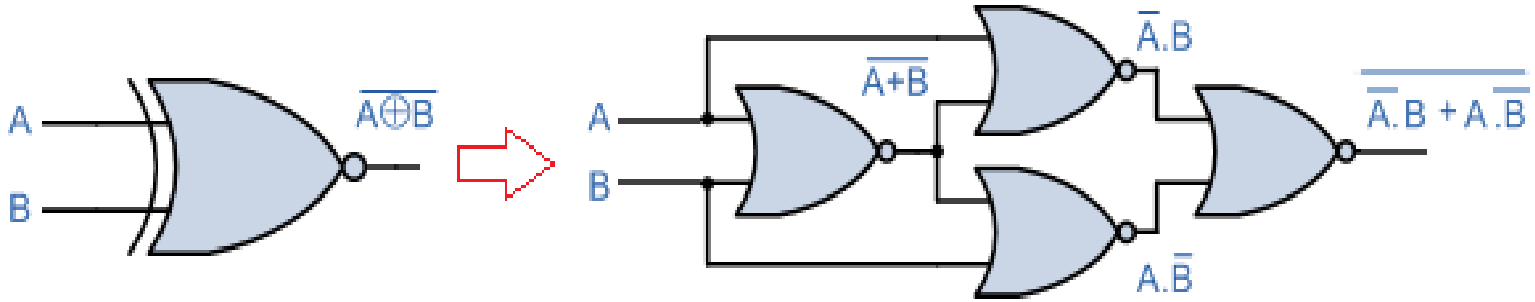
- NAND Gate



- EX-OR Gate



- EX-NOR Gate



Equivalent Gates

- Note that a bubble denotes complementation (inverter) and two bubbles along the same line represent double complementation, so both can be removed.
- Two NOT gates in series are same as a buffer because they cancel each other as $\bar{\bar{A}} = A$

- An AND gate is equivalent to an inverted-input NOR gate.

- A NAND gate is equivalent to an inverted-input OR gate.

- An OR gate is equivalent to an inverted-input NAND gate.

- A NOR gate is equivalent to an inverted-input AND gate.

NAND Implementation

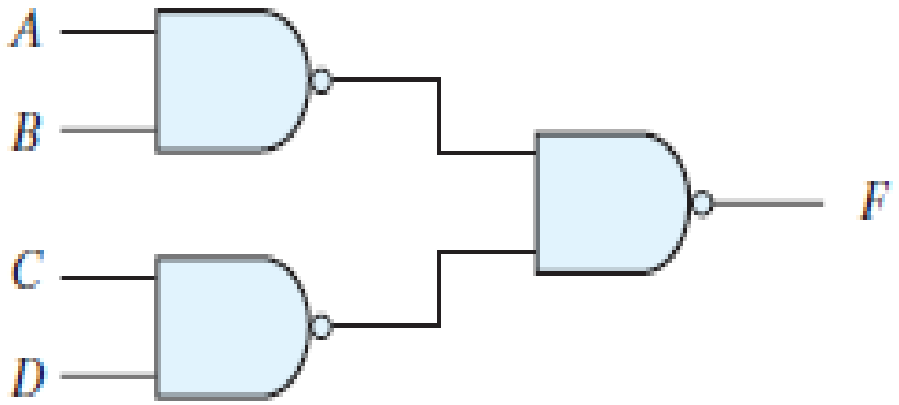
- The implementation of Boolean functions with NAND gates requires that the functions be in sum-of-products form.

F = AB + CD

$$F = AB + CD$$

$$= \overline{\overline{AB + CD}}$$

$$= \overline{\overline{AB} \cdot \overline{CD}}$$



NOR Implementation

- The implementation of Boolean functions with NOR gates requires that the functions be in product-of-sums form.

$$F = (A + B)(C + D)$$

$$F = (A + B).(C + D)$$

$$= \overline{\overline{(A + B).(C + D)}}$$

$$= \overline{\overline{(A + B)} + \overline{(C + D)}}$$