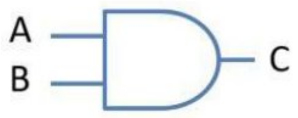
AND Gate

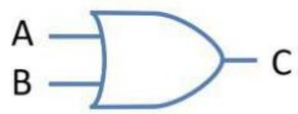
- An AND gate has two or more inputs but only one output.
- The output assumes the logic 1, only when each one of its inputs is at logic 1.
- The output assumes the logic 0 even if one of its inputs is at logic 0.
- The logic symbol & truth table are shown in below figure.
- Notation:- $C = A \cdot B$



A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

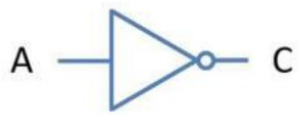
- An OR gate has two or more inputs but only one output.
- The output assumes the logic 0, only when each one of its inputs is at logic 0.
- The output assumes the logic 1 even if one of its inputs is at logic 1.
- The logic symbol & truth table are shown in below figure.
- Notation:- $C = A + B$



A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate

- A NOT gate (also called an *inverter*) has only one input & one output.
- It is a device whose output is always the complement of its input.
- The output assumes the logic 1, when its input is at logic 0.
- The output assumes the logic 0, when its input is at logic 1.
- The logic symbol & truth table are shown in below figure.
- Notation:- $C = \bar{A}$



A	C
0	1
1	0

NAND Gate (Universal Gate)

- NAND means NOT AND, i.e. the AND output is NOTed.
- The output assumes the logic 0, only when each one of its inputs is at logic 1.

- The output assumes the logic 1 even if one of its inputs is at logic 0.
- The logic symbol & truth table are shown in below figure.
- Notation:- $C = \overline{A \cdot B}$

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

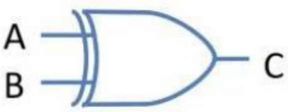
NOR Gate (Universal Gate)

- NOR means NOT OR, i.e. the OR output is NOTed.
- The output assumes the logic 1, only when each one of its inputs is at logic 0.
- The output assumes the logic 0 even if one of its inputs is at logic 1.
- The logic symbol & truth table are shown in below figure.
- Notation:- $C = \overline{A + B}$

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

EX-OR Gate

- An X-OR gate has two or more inputs but only one output.
- The output assumes the logic 1 when one and only one of its inputs assumes a logic 1.
- Under the conditions when both the inputs assume the logic 0, or when both the inputs assume the logic 1, the output assumes a logic 0.
- Since, an X-OR gate produces an output 1 only when the inputs are not equal, it is called an *anti-coincidence gate* or *inequality detector*.
- The logic symbol & truth table are shown in below figure.
- Notation:- $C = A \oplus B$



A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

EX-NOR Gate

- An X-NOR gate has two or more inputs but only one output.
- The output assumes the logic 0 when one and only one of its inputs assumes a logic 0.
- Under the conditions when both the inputs assume the logic 1, or when both the inputs assume the logic 1, the output assumes a logic 0.

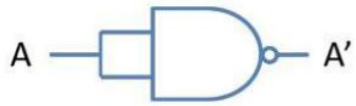
- Since, an X-NOR gate produces an output 1 only when the inputs are equal, it is called a *coincidence gate* or *equality detector*.
- The logic symbol & truth table are shown in below figure.
- Notation:- $C = A \odot B$

A	B	C
0	0	1
0	1	0
1	0	0
1	1	1

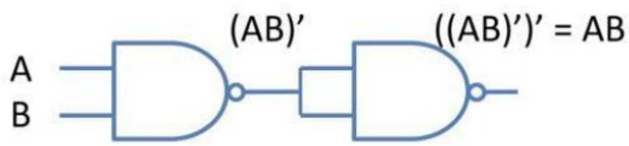
Basic Gates as Universal Gates

- ***Implementation of NOT, AND & OR gates using NAND gate only***

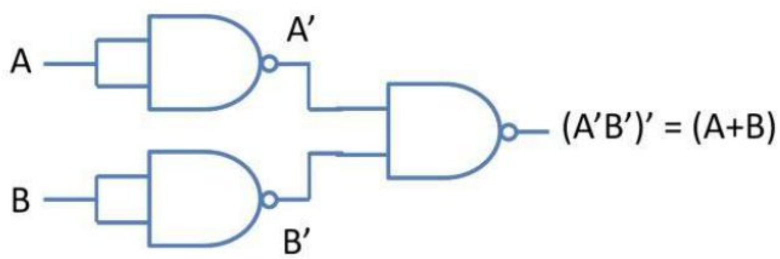
1. NOT using NAND gate
   - A NAND gate can also be used as an inverter by tying all its input terminals together and applying the signal to be inverted to the common terminal.



2. AND using NAND gate
   - NAND means NOT AND, i.e. the AND output is NOTed.
   - So, a NAND gate is combination of an AND gate and a NOT gate.

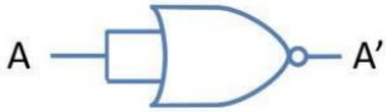


3. OR using NAND gate
   - By inverting inputs in NAND gate, a OR gate is constructed via De Morgan's theorem.
   - $\overline{\overline{A}\,\overline{B}} = \bar{\bar{A}} + \bar{\bar{B}} = A + B$

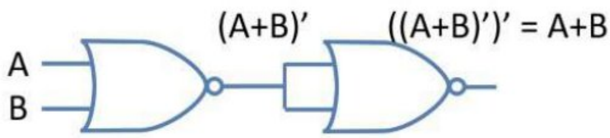


- ***Implementation of NOT, AND & OR gates using NOR gate only***

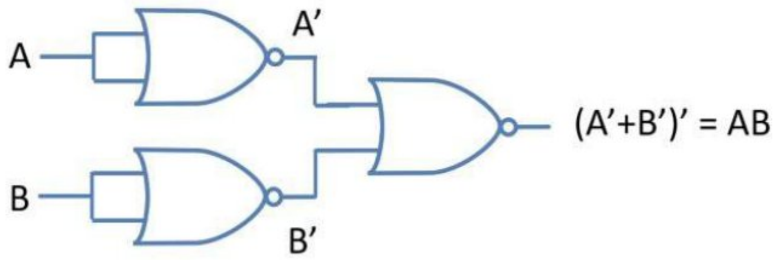
1. NOT using NOR gate
   - A NOR gate can also be used as an inverter by tying all its input terminals together and applying the signal to be inverted to the common terminal.



2. OR using NOR gate
    - NOR means NOT OR, i.e. the OR output is NOTed.
    - So, a NOR gate is combination of an OR gate and a NOT gate.



3. AND using NOR gate
    - By inverting inputs in NOR gate, a AND gate is constructed via De Morgan’s theorem.
    - $\overline{\bar{A} + \bar{B}} = \bar{\bar{A}}\,\bar{\bar{B}} = A\,B$



Boolean Algebra Laws

- AND laws
    1. $A \cdot 0 = 0\ (Null\ Law)$
    2. $A \cdot 1 = A\ (Identity\ Law)$
    3. $A \cdot A = A$
    4. $A \cdot \bar{A} = 0$
- OR laws
    1. $A + 0 = A\ (Null\ Law)$
    2. $A + 1 = 1\ (Identity\ Law)$
    3. $A + A = A$
    4. $A + \bar{A} = 1$

- Commutative laws
    1. $A + B = B + A$
    2. $A \cdot B = B \cdot A$

- Associative laws
    1. $(A + B) + C = A + (B + C)$
    2. $(A \cdot B)C = A(B \cdot C)$

- Distributive laws
    1. $A(B + C) = AB + AC$
    2. $A + BC = (A + B)(A + C)$

- Redundant Literal Rule
  1. $A + \bar{A}B = A + B$
  2. $A(\bar{A} + B) = AB$

- Idempotence laws
  1. $A \cdot A = A$
  2. $A + A = A$

- Absorption laws
  1. $A + AB = A$
  2. $A(A + B) = A$

De Morgan's Theorem

1. Law 1 : $\overline{A + B + C} = \bar{A}\,\bar{B}\,\bar{C}$
   - This law states that the complement of a sum of variables is equal to the product of their individual complements.

A	B	C	A+B+C	(A+B+C)′	A′	B′	C′	A′B′C′
0	0	0	0	**1**	1	1	1	**1**
0	0	1	1	**0**	1	1	0	**0**
0	1	0	1	**0**	1	0	1	**0**
0	1	1	1	**0**	1	0	0	**0**
1	0	0	1	**0**	0	1	1	**0**
1	0	1	1	**0**	0	1	0	**0**
1	1	0	1	**0**	0	0	1	**0**
1	1	1	1	**0**	0	0	0	**0**

   - Hence, Law 1 is proved from the above truth table.

2. Law 2 : $\overline{A\,B\,C} = \bar{A} + \bar{B} + \bar{C}$
   - This law states that the complement of a product of variables is equal to the sum of their individual complements.

A	B	C	A B C	(A B C)′	A′	B′	C′	A′+B′+C′
0	0	0	0	**1**	1	1	1	**1**
0	0	1	0	**1**	1	1	0	**1**
0	1	0	0	**1**	1	0	1	**1**
0	1	1	0	**1**	1	0	0	**1**
1	0	0	0	**1**	0	1	1	**1**
1	0	1	0	**1**	0	1	0	**1**
1	1	0	0	**1**	0	0	1	**1**
1	1	1	1	**0**	0	0	0	**0**

Hence, Law 2 is proved from the above truth table.

Reduction of Boolean Expression

1.
$$
\begin{aligned}
f &= A[B + \bar{C}\,(\overline{AB + A\bar{C}})] \\
&= A\big[B + \bar{C}\,\big(\overline{AB}\;\overline{A\bar{C}}\big)\big] && \text{(De Morgan's Theorem)} \\
&= A[B + \bar{C}\,(\bar{A} + \bar{B})(\bar{A} + C)] && \text{(De Morgan's Theorem)} \\
&= A[B + \bar{C}\,(\bar{A}\bar{A} + \bar{A}C + \bar{B}\bar{A} + \bar{B}C)] && \text{(Distributive Law)} \\
&= A[B + \bar{C}\,(\bar{A} + \bar{A}C + \bar{A}\bar{B} + \bar{B}C)] && (A' \cdot A' = A') \\
&= A(B + \bar{C}\bar{A} + \bar{C}\bar{A}C + \bar{C}\bar{A}\bar{B} + \bar{C}\bar{B}C) && \text{(Distributive Law)} \\
&= A(B + \bar{A}\bar{C} + 0 + \bar{A}\bar{B}\bar{C} + 0) && (C \cdot C' = 0) \\
&= AB + A\bar{A}\bar{C} + A\bar{A}\bar{B}\bar{C} \\
&= AB && (A \cdot A' = 0)
\end{aligned}
$$

2.
$$
\begin{aligned}
f &= A + B[AC + (B + \bar{C})D] \\
&= A + B[AC + BD + \bar{C}D] && \text{(Distributive Law)} \\
&= A + BAC + BBD + B\bar{C}D && \text{(Distributive Law)} \\
&= A + ABC + BD + B\bar{C}D && (B \cdot B = B) \\
&= A(1 + BC) + BD(1 + \bar{C}) \\
&= A \cdot 1 + BD \cdot 1 && (1 + A = A) \\
&= A + BD
\end{aligned}
$$

Common Number Systems

- There are mainly four number systems which are used in digital electronics platform.

  1. Decimal number system
  - The decimal number system contains ten unique symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
  - The base or radix is 10.
  - 9's and 10's complements are possible for any decimal number.

  2. Binary number system
  - The binary number system contains two unique symbols 0, 1.
  - The base or radix is 2.
  - 1's and 2's complements are possible for any binary number.

  3. Octal number system
  - The octal number system contains eight unique symbols 0, 1, 2, 3, 4, 5, 6, 7.
  - The base or radix is 8.
  - 7's and 8's complements are possible for any octal number.

  4. Hexadecimal number system
  - The hexadecimal number system contains sixteen unique symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
  - The base or radix is 16.
  - 15's and 16's complements are possible for any hexadecimal number.