# Logic Gate

## □ Introduction:

Digital system are constructed using logic gates.

The logic gate is the most basic building block any digital system. of making decision including computers. Each one of the logic gates is a piece of hardware or an electric circuit that can be used to implement some basic logic expression. It is an electronic circuit with one or many inputs and only one output.

Most logic gates have two inputs and one output. At any given instances, every input or output terminal is in one of the two logic conditions.

- True or False which describes logic value.
- 1 or 0 provides binary value
- High or low describes voltage level.

- ON or OFF describes the switch position.

The common use of logic gate is to act as switches although they don't have any moving part. Gates opens to pass on logic or close to shut to keep it off.

Gates are classified as three types:

- Basic Gates or Primary logic gates. (AND, OR, NOT)
- Derived logic Gates.
- Universal logic Gates. (NAND, NOR)

## □ Primary logic Gates:

Primary logic Gates are the fundamental building blocks of digital circuit They perform basic logical operations.

### 1. AND Gate:

AND gate is a logic circuit having two or more inputs and one output The output of an AND gate is high only when all of its input are high state, In other cases the output is low.

AND gate diagram:

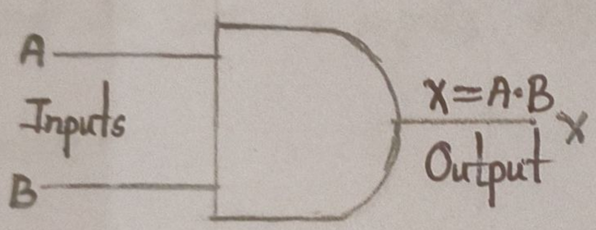


figure: (A) symbol equation of AND gate

Block diagram:

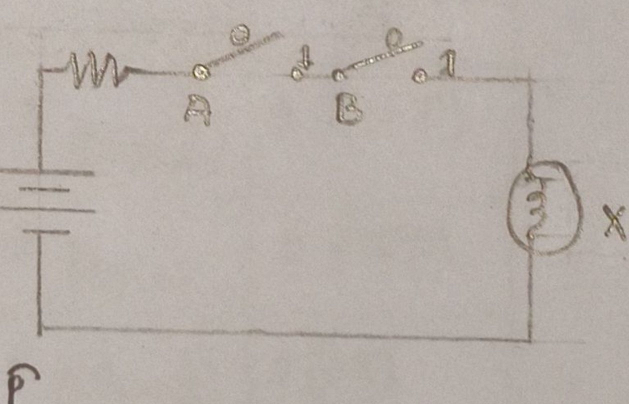


Figure: (b) Switching circuit of AND gate.

The logical operation of AND gate can be expressed with the help of an table which includes all input combination and corresponding outputs. Such table of input /output relations is known as the truth table In this case, there are 2 inputs to AND gates to provide $2^2 = 4$ input combination in the truth table. The truth table may be extended to any number of inputs. for 3 inputs, there are 8 combination. and 16 for 4 inputs. AND gate truth table is given below:

| Input (switch) | | Output |
|---|---|---|
| A | B | $X = A \cdot B$ |
| 0 (Open) | 0 (Open) | 0 (OFF) |
| 0 (Open) | 1 (Closed) | 0 (OFF) |
| 1 (closed) | 0 (open) | 0 (OFF) |
| 1 (Closed) | 1 (Closed) | 1 (ON) |

figure: (c) Truth-table of AND gate

**Note:** The AND gate produces a logic 1 if and only if all the inputs are at logic 1.

AND gate can be implemented using switches. Let us now consider an electric circuit consisting of Battery, Resistor, Switches and Lamp to explain the AND gates function. If both switches are closed then only lamp is on. If anyone or both switches are open then the lamp will be off. (For figure: (b)).

Example:

A traffic light with a green light will only turn on (Output = 1) if both the "car sensor" (input A) and the "pedestrian sensor" (input B) are activated (input = 1).

X = A AND B this a boolean expression.

This expression means that the output X is true when both input A and B are true.

## 2. NOT Gate:

NOT gate has one input and one-output. It is a logic circuit whose output is always the compliment of the input. The NOT gate is popularly known as inverter. It performs logical inversion or complementation.

### NOT gate diagrams:

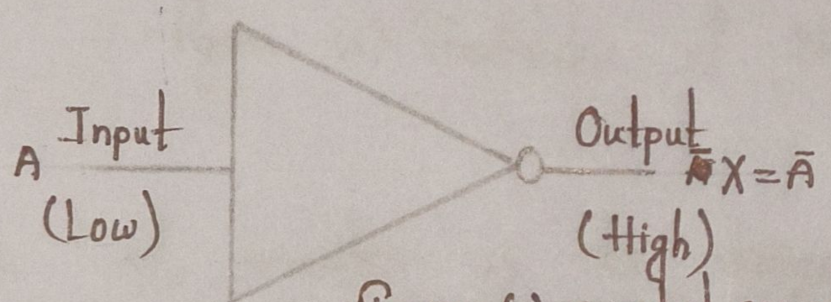


figura: (a) symbol equation of NOT gate

### Block diagram:

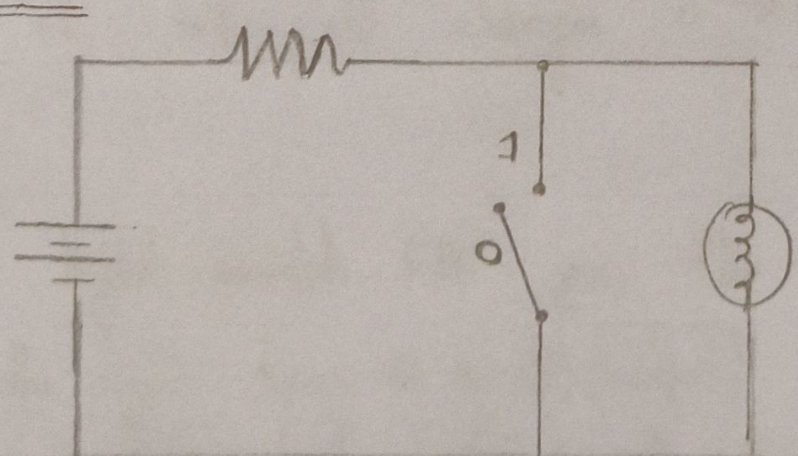


figura: (b) Switching circuit of NOT gate.

The logical operation of ~~Not~~ NOT gate can be expressed with the help of truth table. NOT gate truth table is given below:

| Input (Switch) | Output (Lamp) |
|---|---|
| A | $X = \bar{A}$ |
| 0 (Open) | 1 (ON) |
| 1 (Closed) | 0 (OFF) |

figure: (c) Truthtable of NOT gate.

The purpose of inverter is to change one Logic level to opposite level. The low level at input produces a High level and vice versa. In terms of bit it changes 0 to 1.

~~For~~ **Example:**

Imagine a light switch. When you flip the switch (the input), the light turns on or off (output). The NOT gate works similarly, but instead of turning a light on or off, it inverts the signal.

3. OR Gate:

An OR gate is a logic circuit with two or more inputs and one output. The OR gate performs logical addition

OR Gate diagrams:

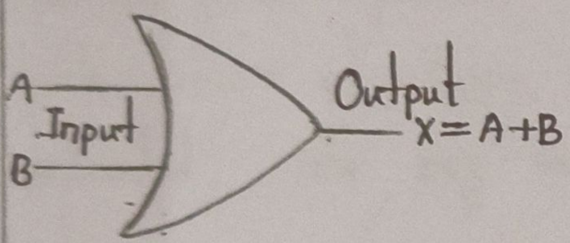


figure: (a) symbol equation of OR gate

Block diagram:

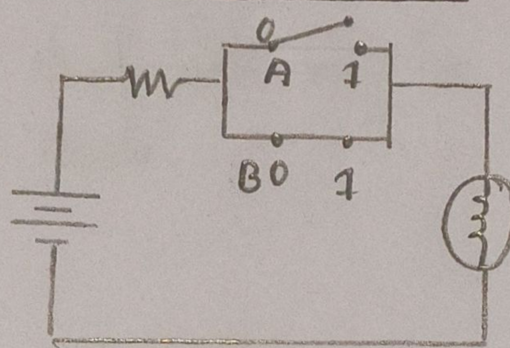


figure: (b) Switching circuit of OR gate

OR gate truthtable is given below:

| Input | | Output |
|---|---|---|
| A(switch) | B(switch) | X=A+B |
| 0 (Open) | 0 (Open) | 0 (OFF) |
| 0 (Open) | 1 (Closed) | 1 (ON) |
| 1 (Closed) | 0 (Open) | 1 (ON) |
| 1 (Closed) | 1 (Closed) | 1 (ON) |

figure:(c) Truth table of OR Gate

OR gate can be implemented using switches. Let us now consider electronic circuit consisting of Battery, Resistor, Switches and Lamp to explain the OR function. If both switches are open then lamp will be off. If any one or both switches are closed then the lamp is on. The OR gate produces logic 1. If either one or both are at logic 1. The truthtable may be extended to any number of inputs.

Example:

Imagine you have two switches controlling a light. If you turn on either switch (or both), the light turns on either switch (or both), the light turns on This is essentially how an OR gate works.

**Derived Gates:** Derived gates, also known as complex gates, are logic gates that are constructed using combinations of three basic gates. These gates perform more complex and logical operation than basic gates and widely used in digital circuit design.

**1. NAND Gate:** NAND gate is a combination of AND and NOT gates The NAND gate provides AND function with inverted output

**NAND Gate diagram:**

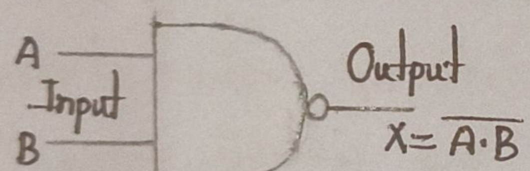


figure: (a) symbolic equation of NAND gate

**Block diagram:**

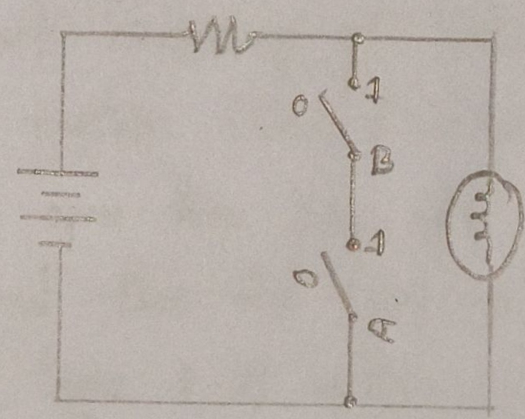


figure: (b) Switching circuit of NAND gate.

The NAND gate produces a logic 1. If either one or both the logic at 0 else the output is at logic 1.

The truthtable of NAND gate is given below:

| Input | | Output |
|---|---|---|
| A (Switch) | B (Switch) | $X = \overline{A \cdot B}$ |
| 0 (Open) | 0 (Open) | 1 (ON) |
| 0 (Open) | 1 (Closed) | 1 (ON) |
| 1 (Closed) | 0 (Open) | 1 (ON) |
| 1 (Closed | 1 (Closed) | 0 (OFF) |

figure: (c) Truth table of NAND gate.

NAND gate can be implemented using switches. Let us now ~~easil~~ consider an electric circuit consisting of Battery, Resistor, Switches and Lamp to explain NAND function. If any one or both switches are open then lamp will be on. If both switches are closed then the lamp will be ~~op~~ off. The NAND gate produces logic 1. If either one of the inputs both inputs are at logic 0 else the output is at logic 1. The truth table may be extended to any number of inputs.

Example: Imagine a simple security system with sensors: a door sensor and window. We want an alarm to trigger only when both the door and window open.

**2. NOR Gate:** NOR gate is combination of OR and NOT gates. The NOR gates provides OR function with inverted output.

**NOR Gate diagram:**

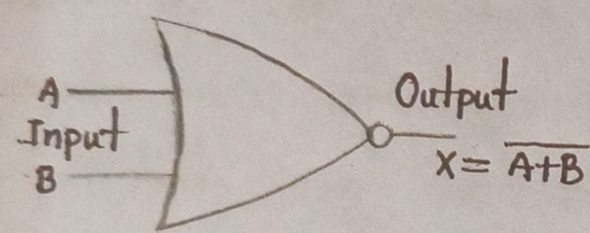


figure: (a) symbolic equation of OR gate.

**Block diagram:**

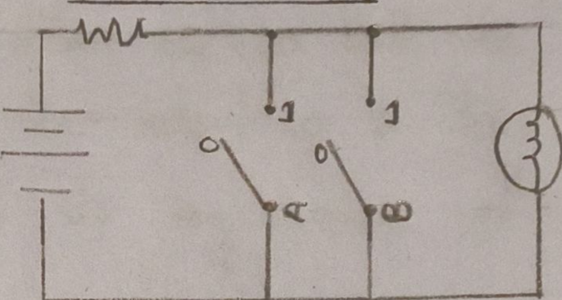


figure: (b) Switching circuit of NOR gate.

The NOR gate produces a logic 0, if either one or both the input are at Logic 1 else the output is at Logic 0.

The truthtable of NOR gate is given below:

~~From~~ fig:

| Input | | Output |
|---|---|---|
| A (switch) | B (switch) | X= $\overline{A+B}$ |
| 0 (Open) | 0 (Open) | 1 (ON) |
| 0 (Open) | 1 (Closed) | 0 (OFF) |
| 1 (Closed) | 0 (Open) | 0 (OFF) |
| 1 (Closed) | 1 (Closed) | 0 (OFF) |

figure: (c) Truth table of NOR gate.

NOR gate can be implemented using switches. Let us now consider an electronic circuit consisting of Battery, Resistor, Switches and Lamp to explain the NOR function. If any one or both the switches are closed then lamp will be off. If both switches are open then the lamp is ON. The NOR produces 0, if either one or both the inputs are at Logic 1 else the output is 0. The truthtable may be extended to any number of inputs.

<u>Example:</u> Conisider a home security system with two sensors: one for a window and one for a door. The system is designed to trigger an alarm only when both sensors detect an intrusion (both inputs are 0). In this case a NOR gate can be used to implement logic. :

- If both window or door sensors are inactive (0), the NOR gates output will be 1, triggering the alarm.

- If either or both sensors are active (1), the NOR gates output will be 0. and the alarm will not be triggerred.

3. XOR (Exclusive OR) Gate: Exclusive OR gate is basically designed to exclude the standard condition of standard OR so as to generate real binary addition. An XOR gate is a two inputs and one outputs logic circuit.

XOR gate diagram:

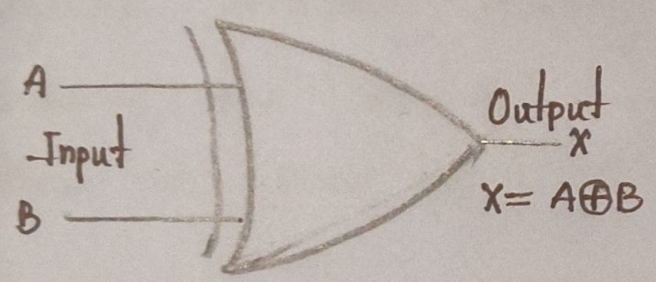


figure: (a) symbol equation of XOR gate

Block diagram:

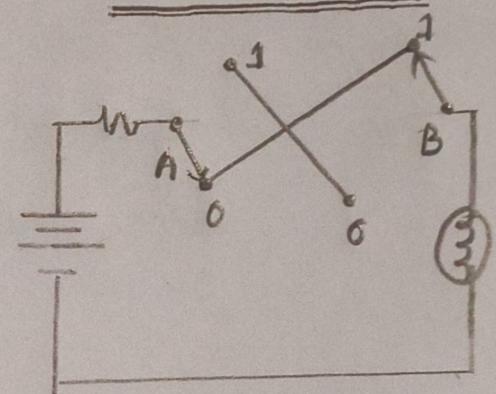


figure: (b) switching circuit of XOR gate.

The XOR gate produces a logic 1, only when the two inputs are at opposite levels.

The truth table of XOR gate is given below:

| Input | | Output |
|---|---|---|
| A | B | $X = A \oplus B$ |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

figure: (c) Truth table of XOR gate.

The XOR gate is known as inequality detector gate. When more than two inputs are to be XORed then multiple two inputs XOR gates may be used. The output of a multiple-input XOR logic function is at logic "1" when the number of 1s in the input sequence is odd and at logic "0" when number of 1s in the input sequence is even. including zero. That is, an all 0s input sequence also produces a logic "0" at the output. Three or more inputs XOR do not exist.

<u>Example:</u> Two switches, A and B, controlling a light. The light will only turn on if one of the switches is on, but not both. This exactly the behavior of an XOR gate.

**4. XNOR Gate:** XNOR is obtained by the combination of NOT and XOR gates. XNOR gate is a two inputs and one output XOR gate with active low output. The logic equation for two input XOR gate given by $X = \overline{A\bar{B}+\bar{A}B}$

**XNOR Gate diagram:**

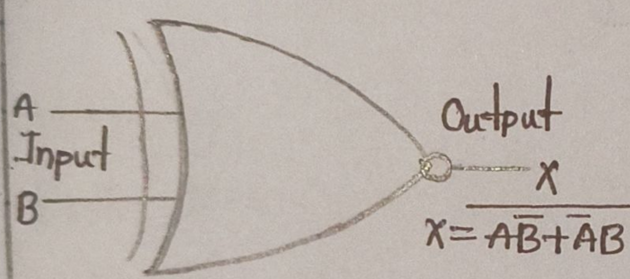


figure: (a) symbol equation of XNOR gate

**Block diagram:**

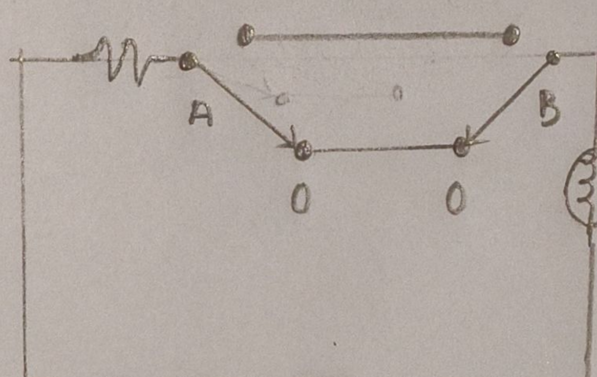


figure: (b) switching circuit of XNOR gate.

The XNOR gate produces a logic 1, only when both the inputs are at the same logic levels.

The truth table of XNOR gate is given below:

| Input | | Output |
|---|---|---|
| A | B | $X=\overline{A\bar{B}+\bar{A}B}$ |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

figure: (c) Truth table of XNOR gate.

XNOR gate is also known as equality detector gate.

<u>Example:</u> Consider two switches, A and B, controlling a light. The light will only turn on both switches are in the same position (either both up or both down). This is an example of XNOR gate in action

## <u>□ Universal Gate:</u>

Both NAND gate and NOR gate can be used as universal gate

<u>NAND Gate (Universal) diagram:</u>