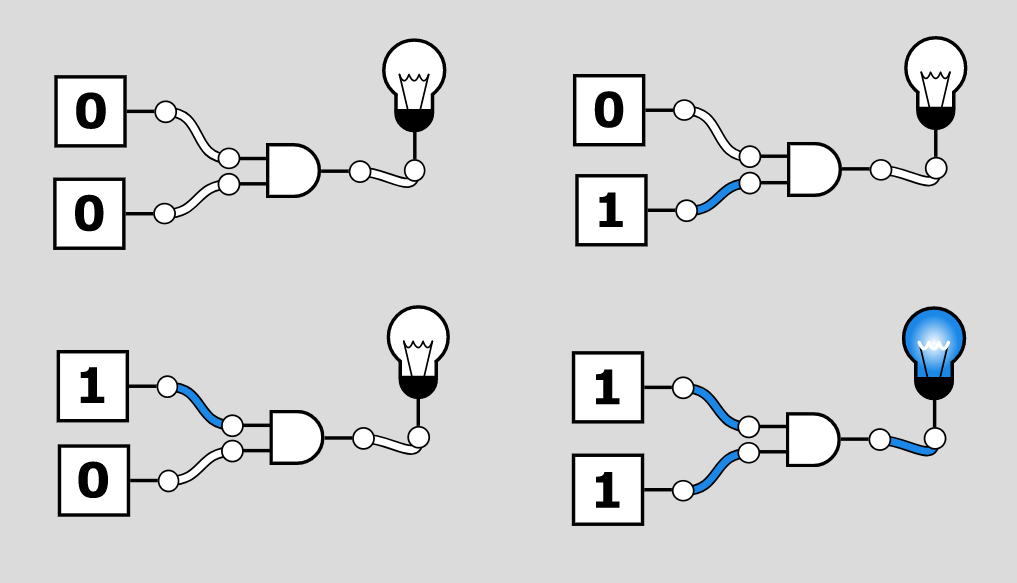
**Shashwat Tripathi**

**D10A 58**

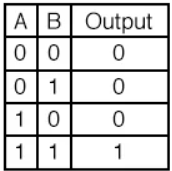
**BATCH: C**

**LAB EXPERIMENT NO: 2**

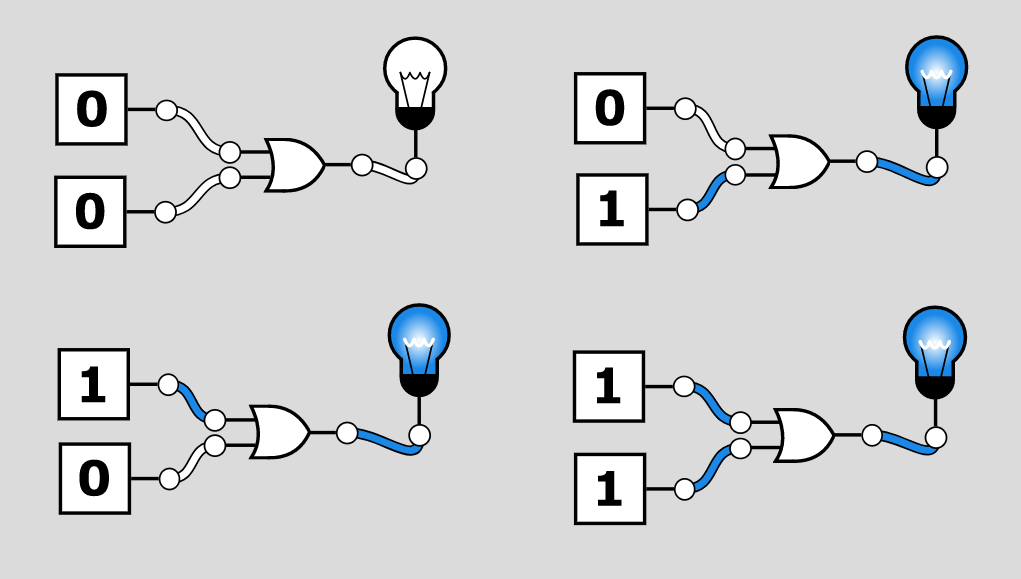
**AND GATE:** The output state of the AND gate will always be low when any of the input states is low. Simply, if any input value in the AND gate is set to 0, then it will always return low output(0). The logic or Boolean expression for the AND gate is the logical multiplication of inputs denoted by a full stop or a single dot as A.B=Y



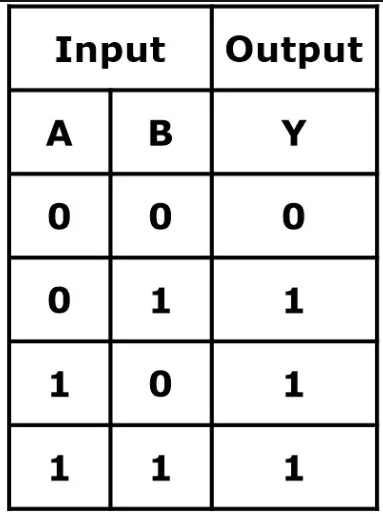
**TRUTH TABLE-**



**OR GATE:** The output state of the OR gate will always be low when both of the inputs states is low. Simply, if any input value in the OR gate is set to 1, then it will always return high-level output(1).The logic or Boolean expression for the OR gate is the logical addition of inputs denoted by plus sign(+) as A+B=Y

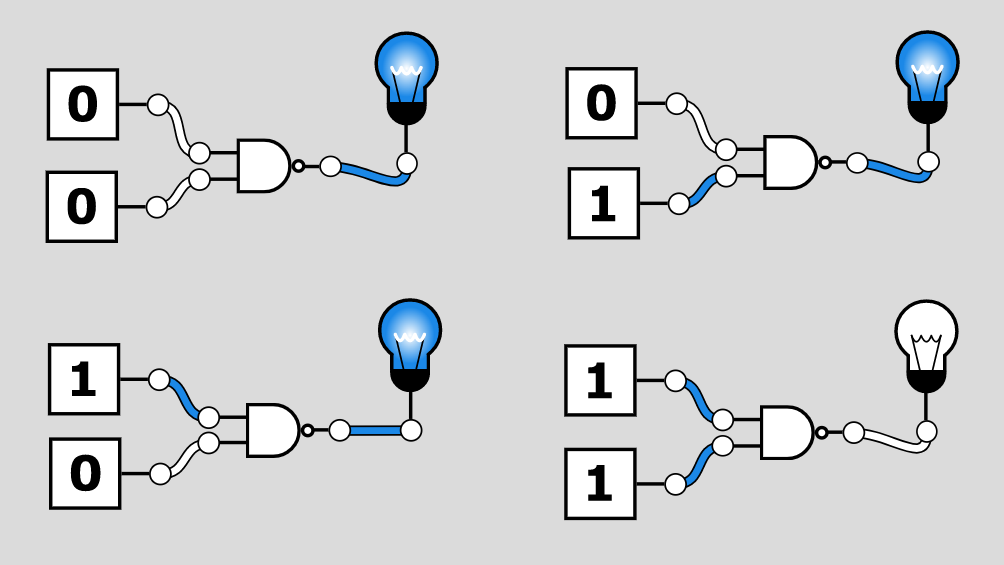


**TRUTH TABLE-**

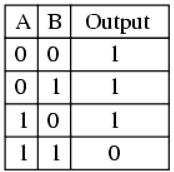
****

**NAND GATE:** The NAND gate is the universal gate. It means all the basic gates such as AND, OR, and NOT gate can be constructed using a NAND gate. The NAND gate is the combination of the NOT-AND gate. The output state of the NAND gate will be low only when all the inputs are high.The logic or Boolean expression for the NAND gate is the complement of logical multiplication of inputs denoted by a full stop or a single dot as

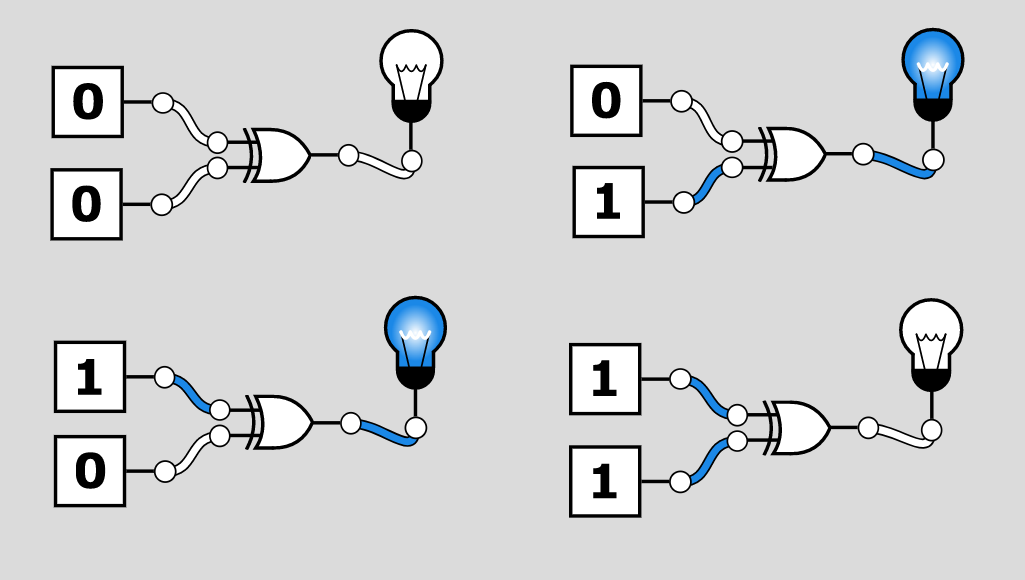
(A.B)'=Y



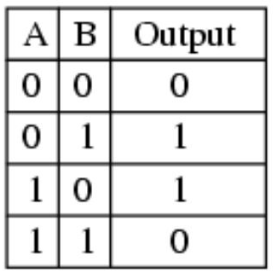
**TRUTH TABLE-**

****

**XOR GATE-** The Exclusive-OR gate or XOR gate is achieved by combining standard logic gates together. XOR gate is used extensively in error detection circuits, computational logic comparators and arithmetic logic circuits. The Exclusive OR gate gives an output only if its two inputs are dissimilar, namely if one of them is high (one) and the other is low (zero).

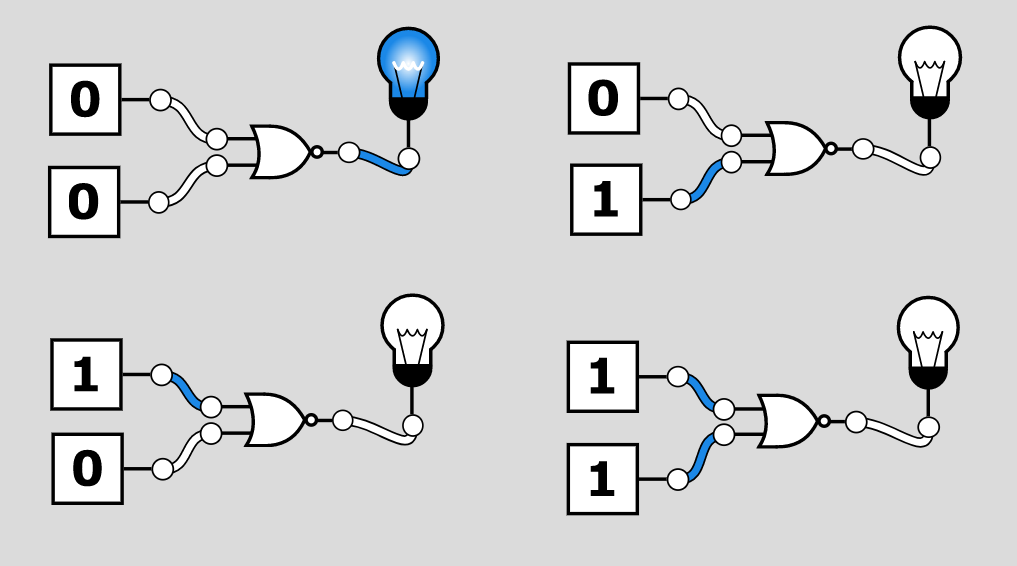


**TRUTH TABLE-**

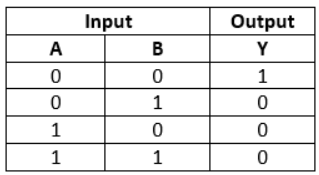


**NOR GATE:** The NOR gate is also a universal gate. So, we can also form all the basic gates using the NOR gate. The NOR gate is the combination of the NOT-OR gate. The output state of the NOR gate will be high only when all of the inputs are low. Simply, this gate returns the complement result of the OR gate.

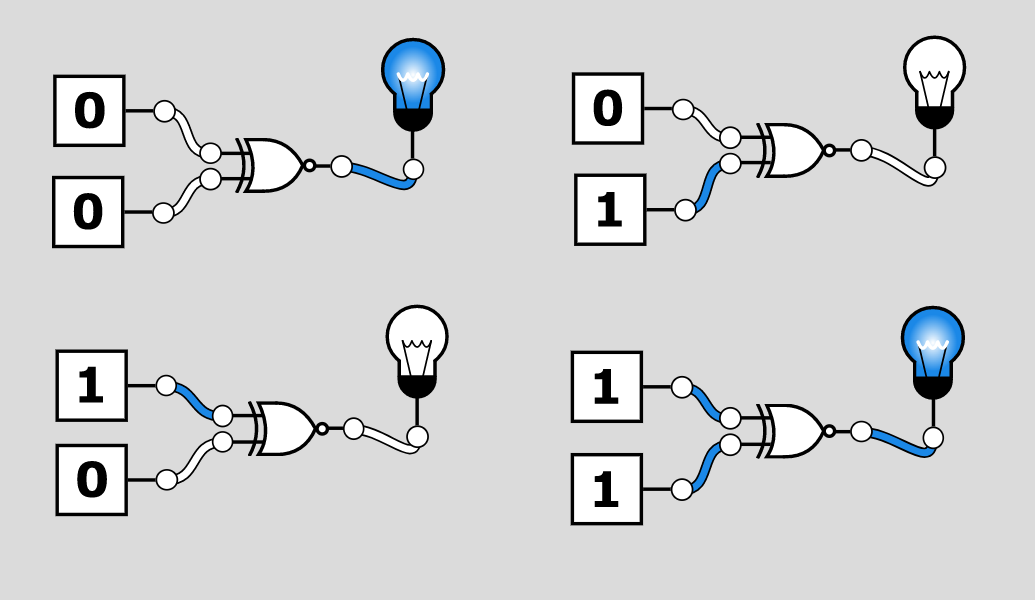
The logical or Boolean expression for the NOR gate is the complement of logical multiplication of inputs denoted by the plus sign as (A+B)'=Y



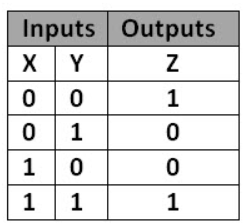
**TRUTH TABLE-**

****

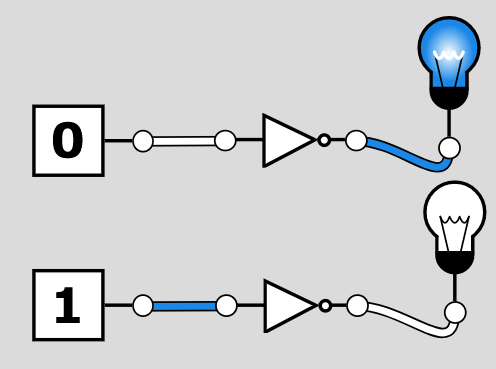
**XNOR GATE:** The XNOR gate is the complement of the XOR gate. It is a hybrid gate. Simply, it is the combination of the XOR gate and NOT gate. The output level of the XNOR gate is high only when both of its inputs are the same, either 0 or 1. The symbol of the XNOR gate is the same as XOR, only complement sign is added.



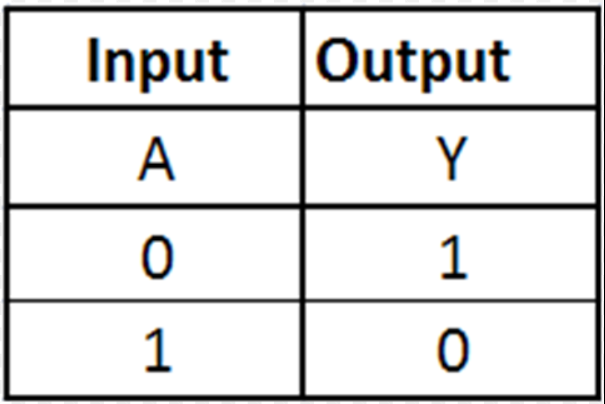
**TRUTH TABLE-**

****

**NOT GATE:** The NOT gate is the most basic logic gate of all other logic gates. NOT gate is also known as an inverter or an inverting Buffer. NOT gate only has one input and one output. When the input signal is "Low", the output signal is "High" and when the input signal is "High", the output is "Low". The Boolean expression for the NOT gate is as follows: A'=Y



**TRUTH TABLE-**

****

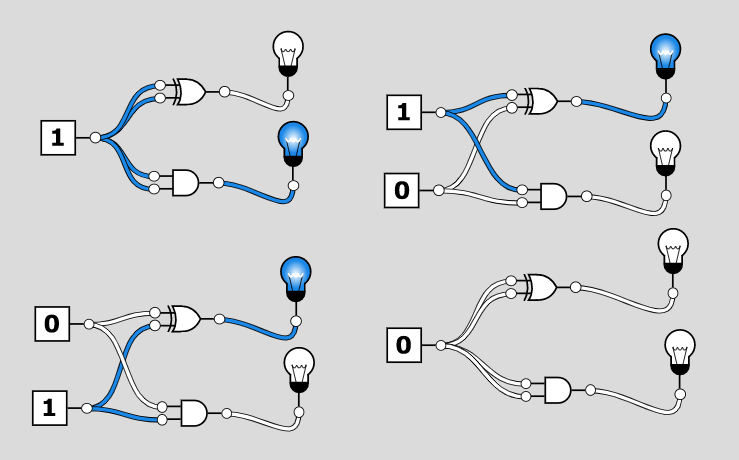
**HALF-ADDER**

The Half-Adder is a basic building block of adding two numbers as two inputs and produce out two outputs. The adder is used to perform OR operation of two single bit binary numbers. The augent and addent bits are two input states, and 'carry' and 'sum 'are two output states of the half adder.

The SOP form of the sum and carry are as follows:

Sum = x'y+xy'

Carry = xy



**TRUTH TABLE**

| **A** | **B** | **SUM** | **CARRY** |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |

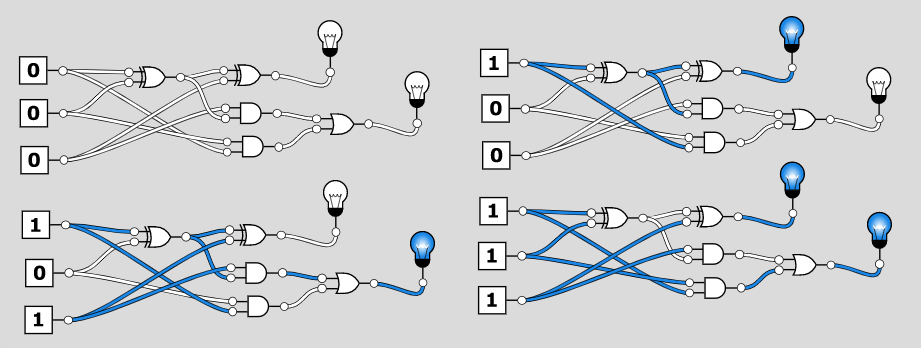
**FULL-ADDER**

The half adder is used to add only two numbers. To overcome this problem, the full adder was developed. The full adder is used to add three 1-bit binary numbers A, B, and carry C. The full adder has three input states and two output states i.e., sum and carry.

The SOP form can be obtained with the help of K-map as:

Sum = x' y' z+x' yz+xy' z'+xyz

Carry = xy+xz+yz



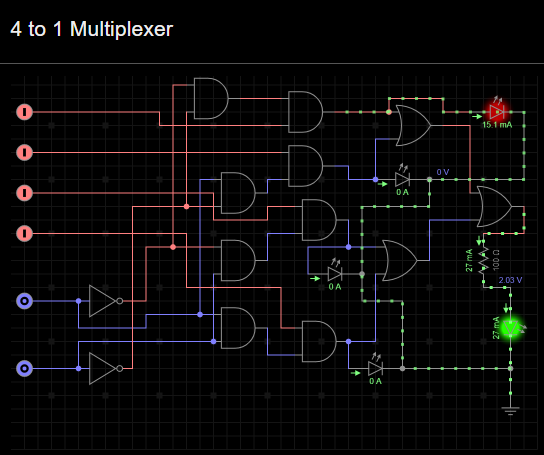
**TRUTH TABLE**

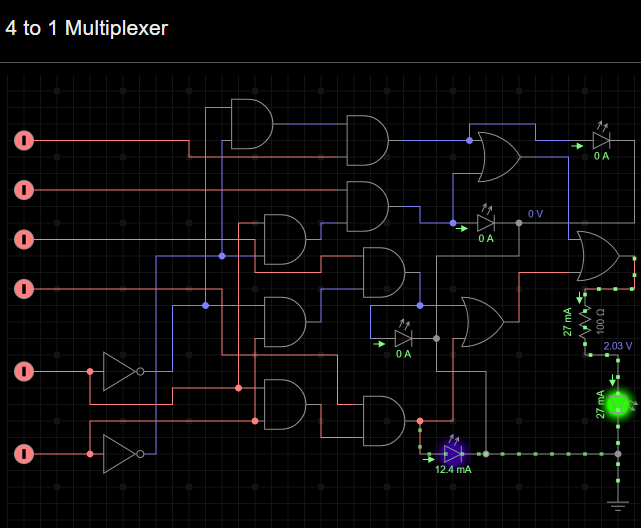
| **A** | **B** | **C** | **SUM** | **CARRY** |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**MULTIPLEXER**

A digital logic circuit which is capable of accepting several inputs and generating a single output is known as multiplexer or MUX. Thus, the multiplexer is a type of data selector which takes many inputs and gives a selected output. In a multiplexer, there are 2n input lines and 1 output line, where n is the number of select lines.

Therefore, a multiplexer is a combinational circuit which is designed to switch one of the many input lines to a single output line by the use of a control signal. For this reason, the multiplexer is also referred to as a many to one circuit.

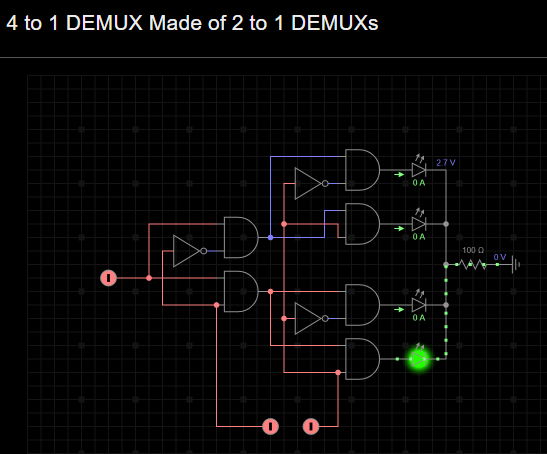


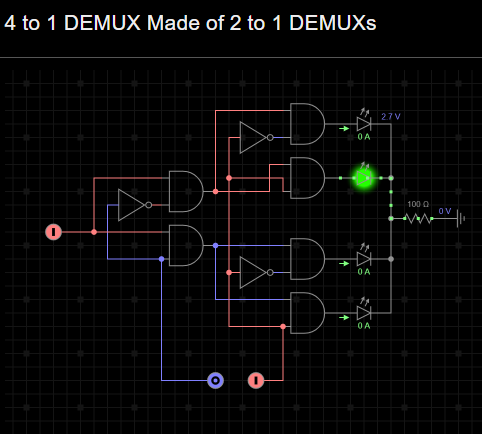
****

**DEMULTIPLEXER**

A digital combinational circuit which takes one input signal and generates multiple output signals is known as demultiplexer or DEMUX. As it distributes a single input signal over many output lines, hence it is also referred to as a type of data distributor.

In a demultiplexer, there is only 1 input line and 2n output lines. Where, n denotes the number of select lines. Therefore, it can be noted that a demultiplexer reverses the operation of a multiplexer.

****

****