STUDY MATERIAL
TOPIC – LOGIC GATES & COMBINATIONAL CIRCUITS

SUBJECT: COMPUTER APPLICATION **CLASS: XII**
DATE: 14.05.2020

Logic Gates

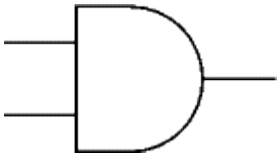
A logic gate is a building block of a digital circuit. Most logic gates have two inputs and one output and are based on Boolean algebra. At any given moment, every terminal is in one of the two binary conditions *false* (high) or *true* (low). False represents 0, and true represents 1. Depending on the type of logic gate being used and the combination of inputs, the binary output will differ. A logic gate can be thought of like a light switch, wherein one position the output is off—0, and in another, it is on—1. Logic gates are commonly used in integrated circuits (IC).

❖ Basic Logic gates

➢ AND gate

An AND gate can have two or more inputs, its output is true if all inputs are true. The output Q is true if input A AND input B are both true: **Q = A . B**

Input A	Input B	Output Q
0	0	0
0	1	0
1	0	0
1	1	1

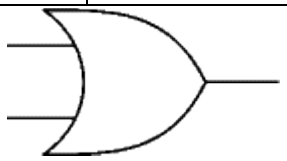


Logical symbol

➢ OR gate

An OR gate can have two or more inputs, its output is true if at least one input is true.
The output Q is true if input A OR input B is true (or both of them are true): **Q = A + B**

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	1

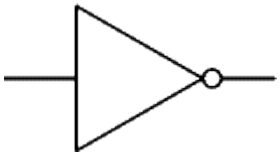


Logical symbol

➢ NOT gate (inverter)

A NOT gate can only have one input and the output is the inverse of the input. A NOT gate is also called an inverter.
The output Q is true when the input A is NOT true: **Q =** $\overline{A}$

Input A	Output Q
0	1
1	0



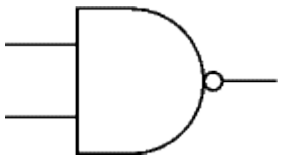
Logical symbol

❖ Universal gates

➢ NAND gate

This is an AND gate with the output inverted, as shown by the 'o' on the symbol output. A NAND gate can have two or more inputs, its output is true if NOT all inputs are true. The output Q is true if input A AND input B are NOT both true: **Q =** $\overline{A \,.\, B}$

Input A	Input B	Output Q
0	0	1
0	1	1
1	0	1
1	1	0

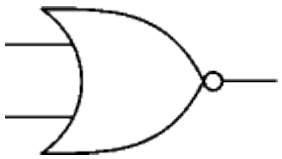


Logical symbol

➢ NOR gate

This is an OR gate with the output inverted, as shown by the 'o' on the symbol output. A NOR gate can have two or more inputs, its output is true if no inputs are true. The output Q is true if NOT inputs A OR B are true: **Q =** $\overline{A + B}$

Input A	Input B	Output Q
0	0	1
0	1	0
1	0	0
1	1	0



Logical symbol

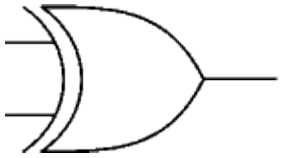
❖ Exclusive gates

➢ EX-OR gate

EXclusive-OR. This is like an OR gate but excluding both inputs being true. The output is true if inputs A and B are DIFFERENT. EX-OR gates can only have 2 inputs.

The output Q is true if either input A is true OR input B is true, *but not when both of them are true*: **Q = A ⊕ B**

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	0

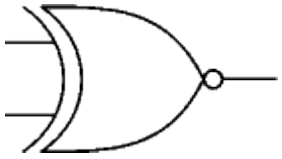


Logical symbol

➢ EX-NOR gate

EXclusive-NOR. This is an EX-OR gate with the output inverted, as shown by the 'o' on the symbol output. EX-NOR gates can only have 2 inputs. The output Q is true if inputs A and B are the *SAME* (both true or both false): **Q =** $\overline{A \oplus B}$

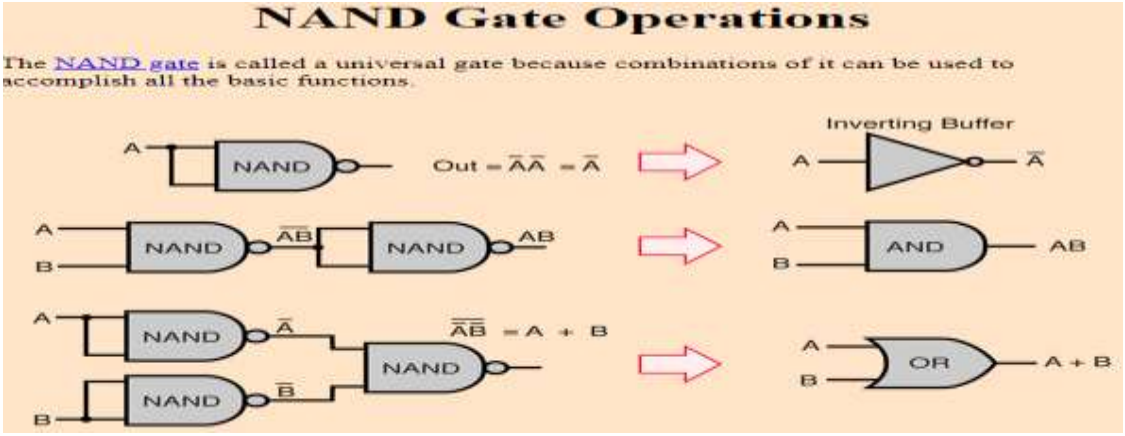
Input A	Input B	Output Q
0	0	1
0	1	0
1	0	0
1	1	1

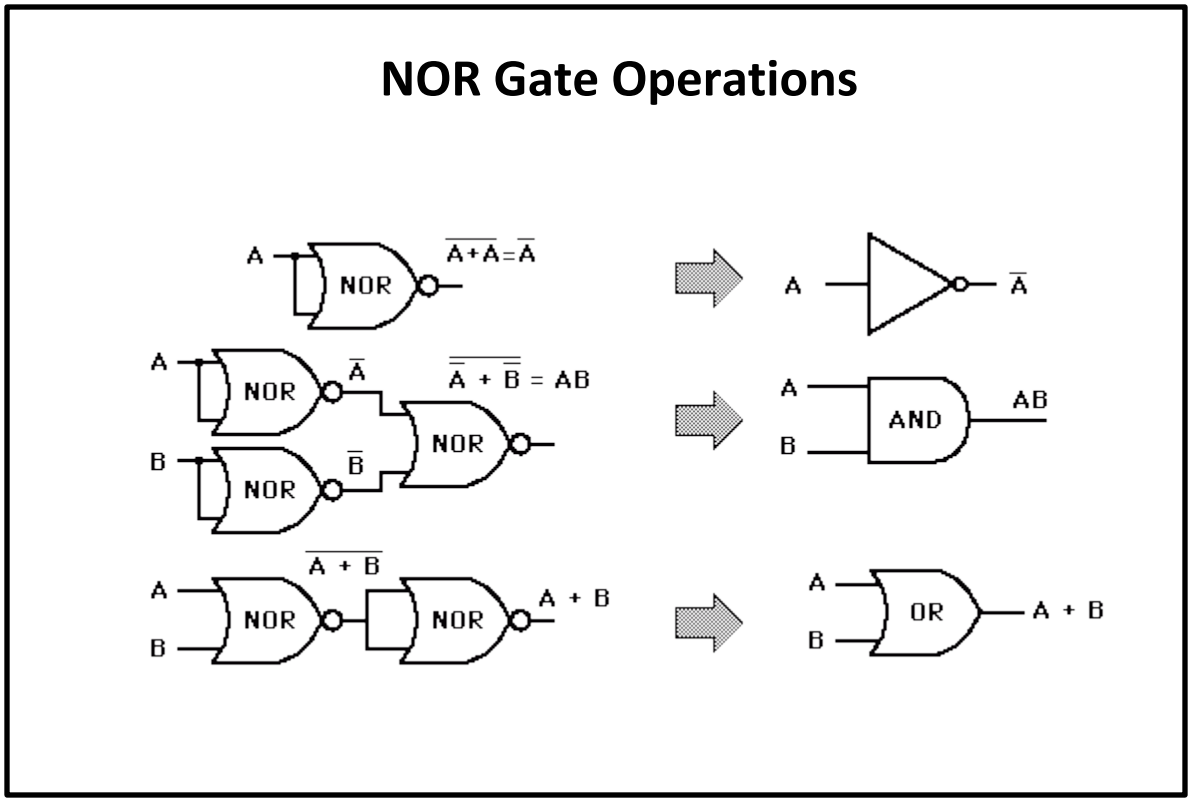


Logical symbol

✓ Why NAND and NOR gates are called Universal gates?

NOR gates and NAND gates have the particular property that any **one of them** can create any logical Boolean expression if appropriately designed. Meaning that you can create any logical Boolean expression using ONLY NOR gates or ONLY NAND gates. Other logical gates do not have this property.





Combinational Circuits

Combinational Circuits (CC) are circuits made up of different types of logic gates. A **logic gate** is a basic building block of any electronic circuit. The output of the combinational circuit depends on the values at the input at any given time. The circuits do not make use of any memory or storage device.

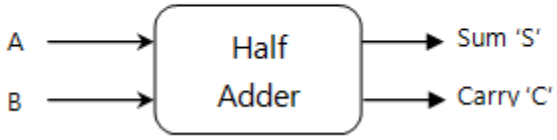
❖ The Adder

An adder is a digital circuit that is used to perform the addition of numeric values. It is one of the most basic circuits and is found in arithmetic logic units of computing devices. There are two types of adders. **Half adders** compute single digit numbers, while **full adders** compute larger numbers.

➢ Half Adder

The half adder adds two single digit binary numbers and forms the foundation for all addition operations in computing. If we have two single binary digits, A and B, then the half adder adds them with the circuit carrying two outputs, the sum and the carry. The carry represents any overflow from the addition of the two numbers. This is represented in the following block diagram figure:

Figure 1: Half Adder



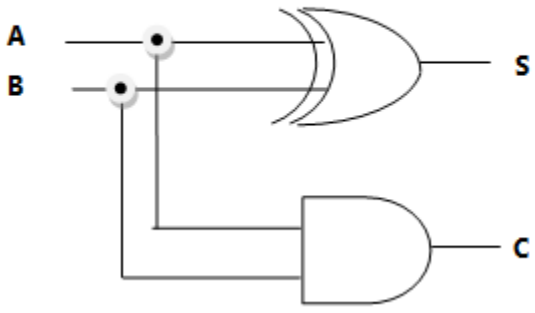
In addition, the following truth table demonstrates all the possible outputs for various input combinations of the half adder.

Table 1: Truth Table - Half Adder

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

The next figure represents the logic circuit of the half adder:

Figure 2: Half Adder- Logic Circuit

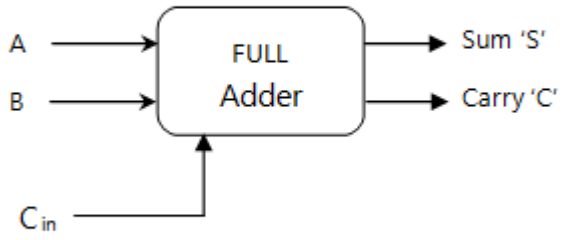


The sum S is represented by the Boolean Expression S = A'B + AB' = $\mathbf{A} \oplus \mathbf{B}$ and C = AB

➢ Full Adder

The full adder overcomes the disadvantages of the half adder in that it can add two single bit numbers in addition to the carry digit at its input as seen in this figure:

Figure : Full Adder

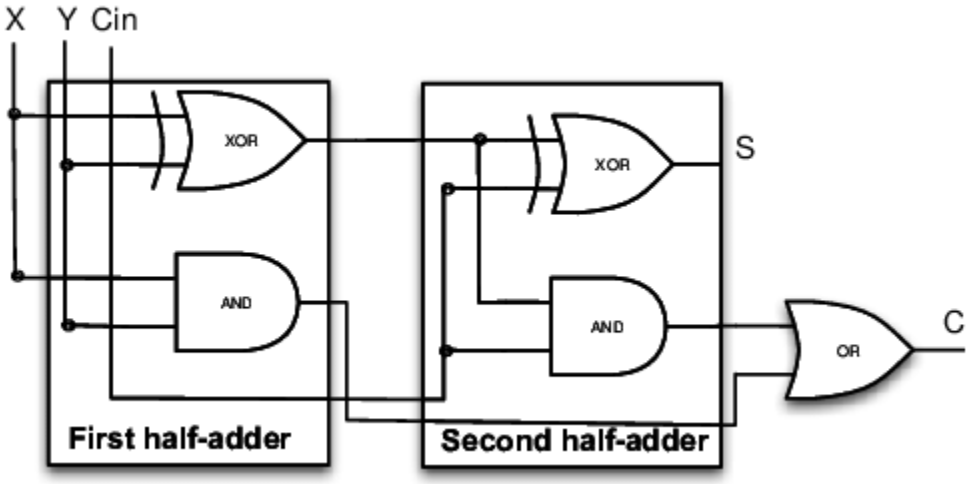


The next truth table shown here demonstrates all the possible outputs for various input combinations with the carry input digit:

Table 2: Truth Table - Full Adder

A	B	Cin	Co	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

Boolean expression for the full adder is S = A'B'Cin + A'BCin' + AB'Cin' + ABCin and C = A'BCin + AB'Cin + ABCin' + ABCin. This is where A and B are all the possible binary inputs and C is the carry in.



❖ Subtractors

A subtractor is used to subtract one number from another. Because we are dealing with binary digits, the 1s complement and 2s complement of the numbers are used to achieve this. Three bits are involved in performing the basic subtraction: the minuend (X), the subtrahend (Y) and the borrow (Bi), which is input from the previous bit. The outputs are the difference (D) and the borrow bit (Bout).

➢ Half Subtractor

When a subtraction is done between just two bits a half subtractor is used, similar to the half adder. The half subtractor's combinational circuit is represented in this image as well as the half subtractor table:

Figure 4: Half Subtractor – Logic Circuit

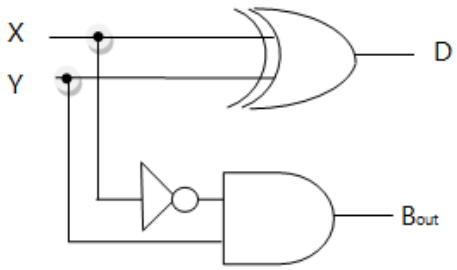


Table 3: Truth Table - Half Subtractor

X	Y	D=(X-Y)	Bout
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

The Boolean expressions are as follows:

D = X'Y + XY' = **X** $\oplus$ **Y**

Bout = X'Y

➢ Full Subtractor

The combinational circuit of the full subtractor performs a subtraction operation on three bits, the minuend, the subtrahend, and the borrow-in bits. The circuit generates two outputs comprising of the calculated difference, D and the borrow-out.

Figure 5: Full Subtractor – Logic Circuit

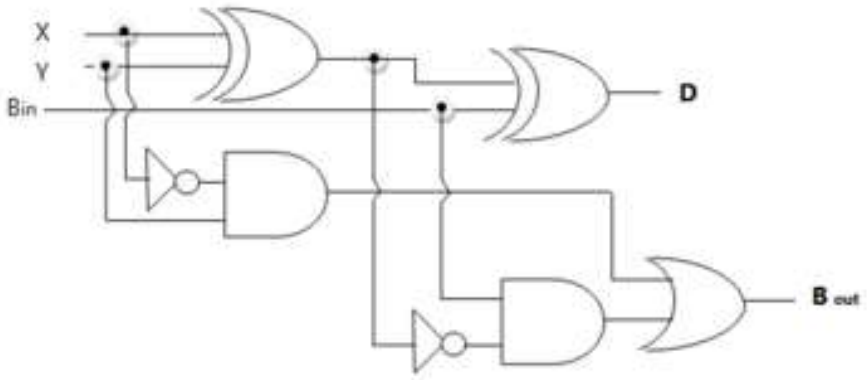


Table 4: Truth Table - Full Subtractor

X	Y	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Boolean expressions:

D = X'Y'Bin + X'YBin' + XY'Bin' + XYBin

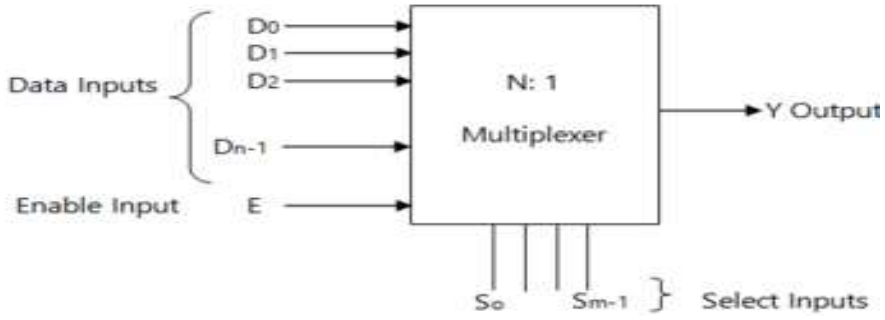
Bout = X'Y'Bin + X'YBin' + X'YBin + XYBin

❖ Multiplexers

Multiplexers are combinational circuits designed to select one of multiple data inputs and produce a single output. They're commonly used in communication transmissions.

The input lines are selected depending on the selection inputs called control lines. The binary state of these inputs can either be low '0' or high '1'. Multiplexers have an even number of data input lines D as 2N, with a corresponding number of control lines S.

Figure 6: Multiplexer



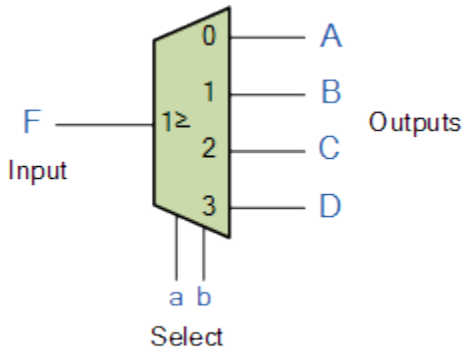
Multiplexers are designed at different levels. There are 2:1, 4:1, 16:1, and 32:1 multiplexers.

❖ Demultiplexer

The data distributor, known more commonly as a **Demultiplexer** or “Demux” for short, is the exact opposite of the Multiplexer we saw in the previous tutorial.

The *demultiplexer* takes one single input data line and then switches it to any one of a number of individual output lines one at a time. The **demultiplexer** converts a serial data signal at the input to a parallel data at its output lines as shown below.

The Demultiplexer Symbol



1-to-4 Channel De-multiplexer

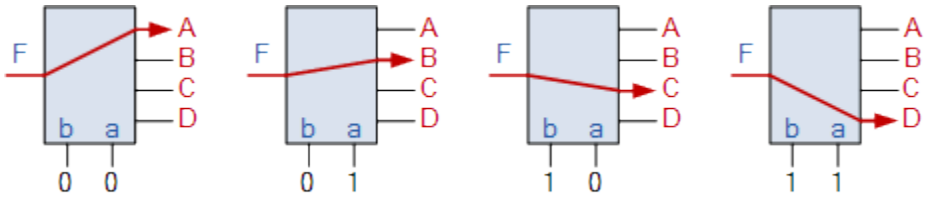
Output Select		Data Output Selected
a	b	
0	0	A
0	1	B
1	0	C
1	1	D

The Boolean expression for this 1-to-4 **Demultiplexer** above with outputs A to D and data select lines a, b is given as:

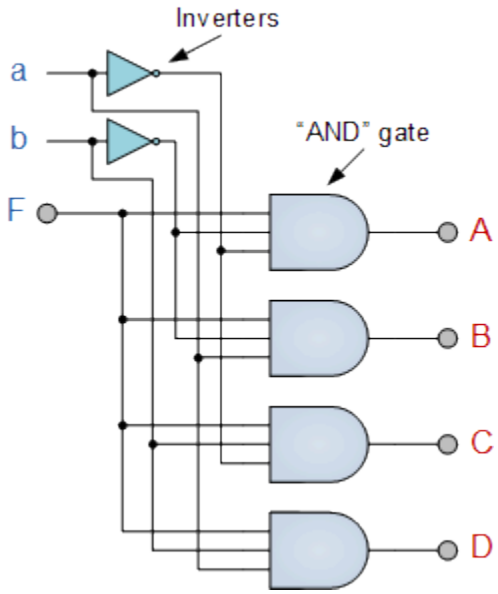
$$F = abA + abB + abC + abD$$

The function of the **Demultiplexer** is to switch one common data input line to any one of the 4 output data lines A to D in our example above. As with the multiplexer the individual solid state switches are selected by the binary input address code on the output select pins "a" and "b" as shown.

Demultiplexer Output Line Selection



4 Channel Demultiplexer using Logic Gates

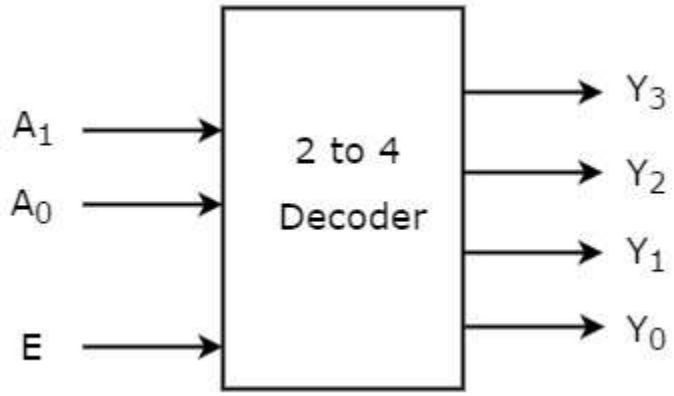


❖ Decoder

Decoder is a combinational circuit that has 'n' input lines and maximum of 2n output lines. One of these outputs will be active High based on the combination of inputs present, when the decoder is enabled. That means decoder detects a particular code. The outputs of the decoder are nothing but the min terms of 'n' input variables lineslines, when it is enabled.

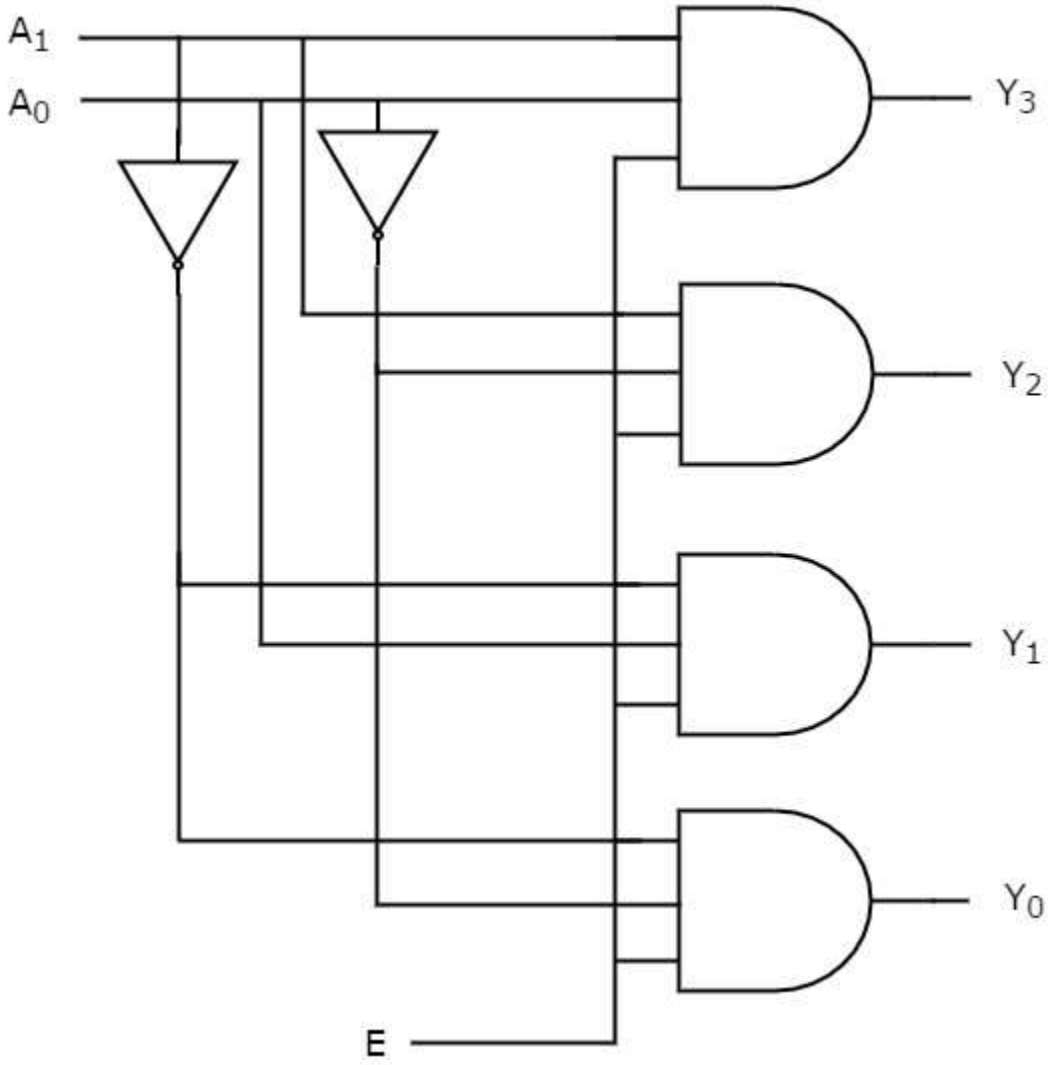
➢ 2 to 4 Decoder

Let 2 to 4 Decoder has two inputs A1 & A0 and four outputs Y3, Y2, Y1 & Y0. The block diagram of 2 to 4 decoder is shown in the following figure.



INPUT		OUTPUT			
A_1	A_0	Y_3	Y_2	Y_1	Y_0
x	x	0	0	0	0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

The circuit diagram of 2 to 4 decoder is shown in the following figure.



E is the enable.

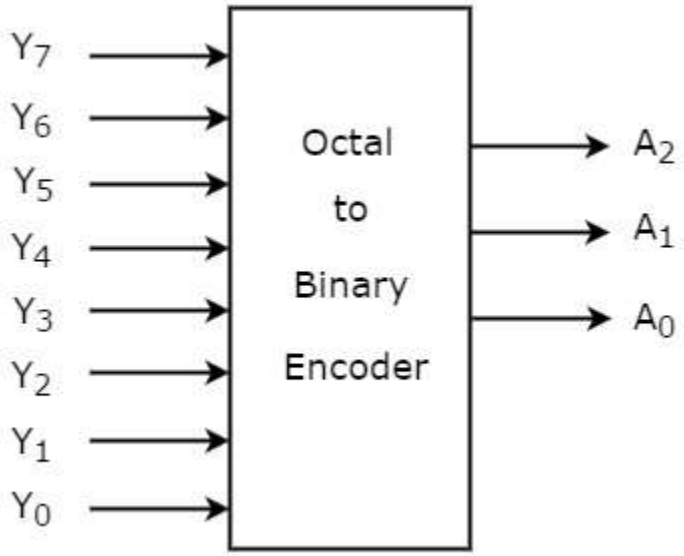
A standard decoder typically has an additional input called Enable. Output is only generated when the Enable input has value 1; otherwise, all outputs are 0. Only a small change in the implementation is required: the Enable input is fed into the AND gates which produce the outputs.

❖ Encoder

An Encoder is a combinational circuit that performs the reverse operation of Decoder. It has maximum of 2n input lines and 'n' output lines. It will produce a binary code equivalent to the input, which is active High. Therefore, the encoder encodes 2n input lines with 'n' bits. It is optional to represent the enable signal in encoders.

➢ Octal to Binary Encoder

Octal to binary Encoder has eight inputs, Y7 to Y0 and three outputs A2, A1 & A0. Octal to binary encoder is nothing but 8 to 3 encoder. The block diagram of octal to binary Encoder is shown in the following figure.



At any time, only one of these eight inputs can be '1' in order to get the respective binary code.

The Truth table of octal to binary encoder is shown below.

Inputs								Outputs		
Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

From Truth table, we can write the **Boolean functions** for each output as

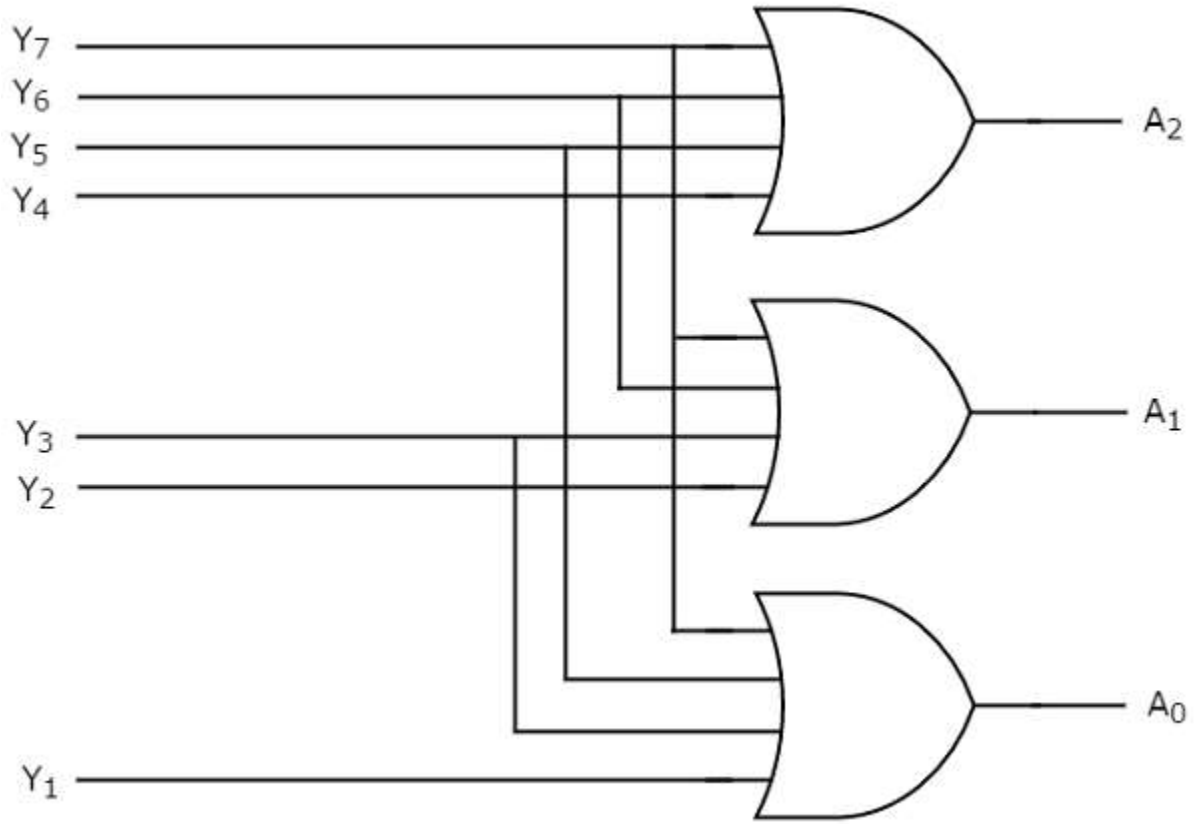
$$A_2=Y_7+Y_6+Y_5+Y_4A2=Y7+Y6+Y5+Y4$$

$$A_1=Y_7+Y_6+Y_3+Y_2A1=Y7+Y6+Y3+Y2$$

$$A_0=Y_7+Y_5+Y_3+Y_1A0=Y7+Y5+Y3+Y1$$

We can implement the above Boolean functions by using four input OR gates.

The circuit diagram of octal to binary encoder is shown in the following figure.



❖ Reference:

- J. Bhattacharya
- Modern Computer Applications by Rajiv Mathur
- http://hyperphysics.phy-astr.gsu.edu/hbase/Electronic/nand.html
- https://www.google.com/search?q=why+nandNOR+gate+is+universal&tbm=isch&ved=2ahUKEwi48YftpK7pAhUeMrcAHWHXA90Q2-cCegQIABAA&oq=why+nandNOR+gate+is+universal&gs_lcp=CgNpbWcQA1CvoARY_qoEYJCuBGgAcAB4AIABnAGIAcUDkgEDMC4zmAEAoAEBqgELZ3dzLXdpei1pbWc&sclient=img&ei=bpG6XriCNZ7k3LUP4a6P6A0#imgrc=_QyvaXogdb1c-M
- https://www.electronics-tutorials.ws/logic/logic_10.html
- https://study.com/academy/lesson/basic-combinational-circuits-types-examples.html
- https://www.elprocus.com/introduction-to-combinational-logic-circuits/
- https://www.tutorialspoint.com/computer_logical_organization/combinational_circuits.htm
- https://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR
- https://www.electronics-tutorials.ws/logic/universal-gates.html

***

PRITHWISH DE