

MODULE 3

Important topics

- Gray to binary circuit
- half adder and full adder circuits
- parallel adder-subtractor circuit
- 2-bit Magnitude Comparator,
- Decoder, Encoder, priority encoder,
- two-to-one-line multiplexer


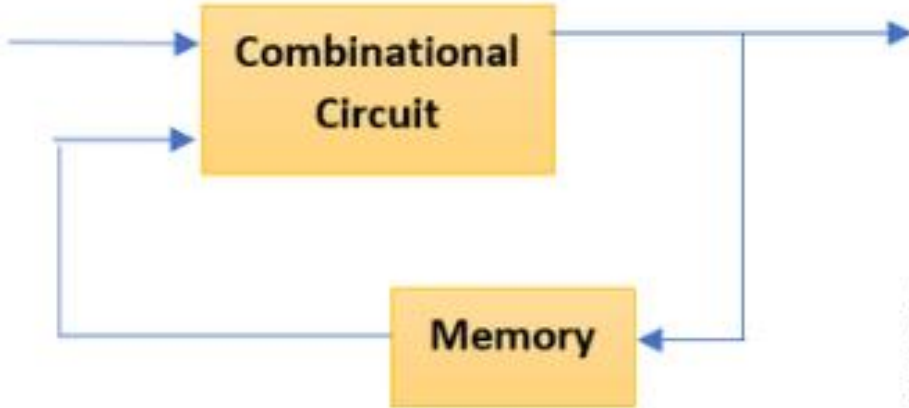
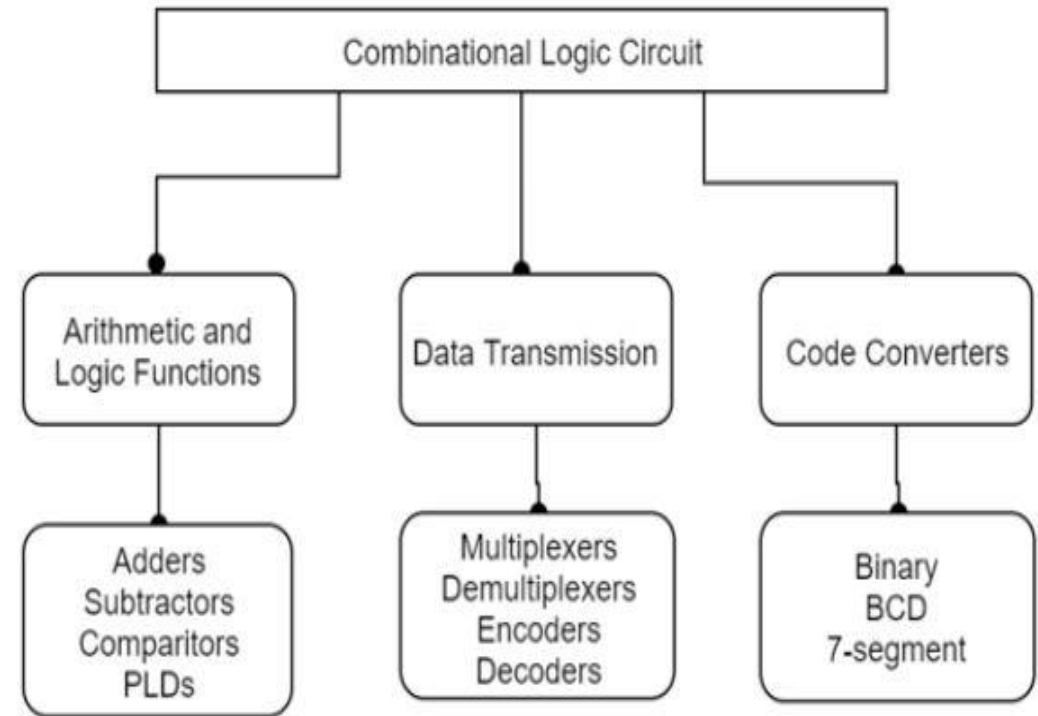
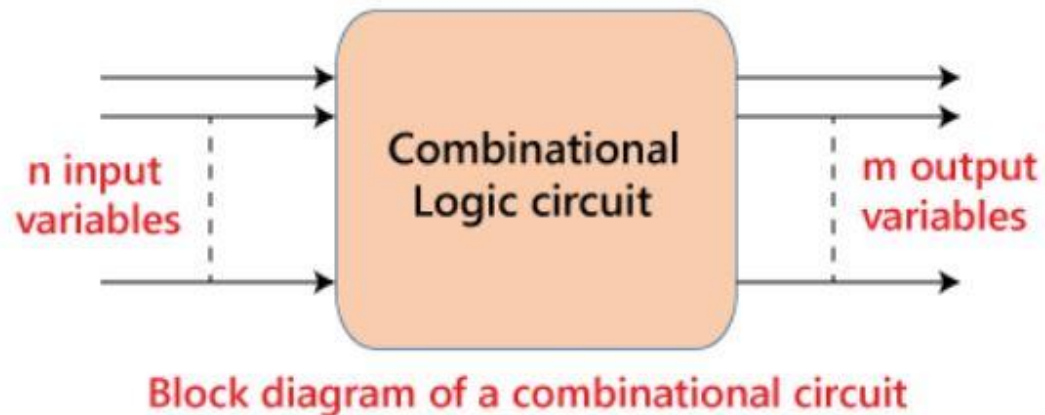
Combinational Circuit	Sequential Circuit
Output only depends on the present input	Output depends on present input and past output
Memory element is absent	Memory element is present
No clock signal is applied	Clock signal is required
	
Example - Half Adder, Full Adder, Multiplexer	Examples - Flipflop, Counters, Registers

Image Credits: Electronules

Combinational Logic circuit

- ✓ The combinational logic circuits are the circuits that contain different types of logic gates.
- ✓ Simply, a circuit in which different types of logic gates are combined is known as a **combinational logic circuit**.
- ✓ The output of the combinational circuit is determined from the present combination of inputs, regardless of the previous input.
- ✓ The input variables, logic gates, and output variables are the basic components of the combinational logic circuit.
- ✓ There are different types of combinational logic circuits, such as **Adder, Subtractor, Decoder, Encoder, Multiplexer, and De-multiplexer**.

- At any instant of time, the output of the combinational circuits depends only on the present input terminals.
- The combinational circuit doesn't have any backup or previous memory. The present state of the circuit is not affected by the previous state of the input.
- The n number of inputs and m number of outputs are possible in combinational logic circuits.



Binary Adder

- ✓ The binary adder is a combinational circuit that can perform summation of the input binary numbers.
- ✓ The most common or basic arithmetic operation is the addition of binary digits.
- ✓ A combination circuit which performs the additions of two bits is called a half adder while that performs the addition of three bits is a full adder.

1. Half Adder : Half Adder is a combinational logic circuit which is designed by connecting one EX-OR gate and one AND gate. The half adder circuit has two inputs: A and B, which add two input digits and generates a carry and a sum.

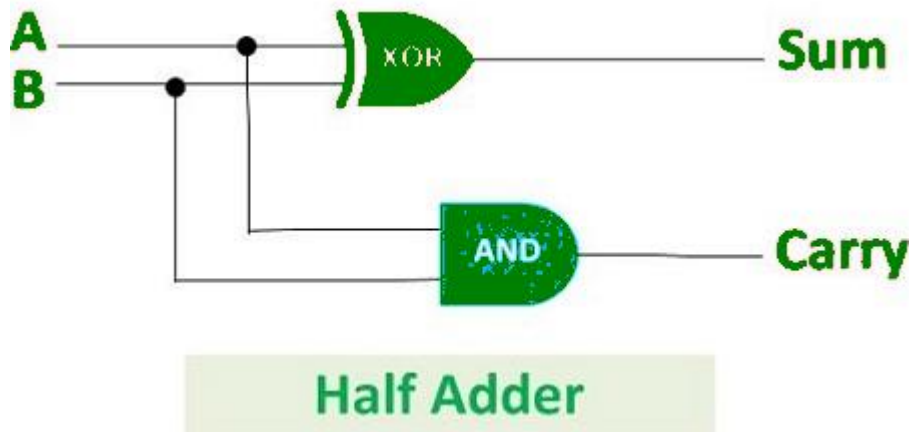
The output obtained from the EX-OR gate is the sum of the two numbers while that obtained by AND gate is the carry. There will be no forwarding of carry addition because there is no logic gate to process that.

Logical Expression :

$$\text{Sum} = A \text{ XOR } B$$

$$\text{Carry} = A \text{ AND } B$$

Truth Table			
Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Parameters	Half Adder	Full Adder
Description	Half Adder is a combinational logic circuit that adds two 1-bit digits. The half adder produces a sum of the two inputs.	A full adder is a combinational logic circuit that performs an addition operation on three one-bit binary numbers. The full adder produces a sum of the three inputs and carry value.
Previous carry	The previous carry is not used.	The previous carry is used.
Inputs	In Half adder, there are two input bits (A, B).	In full adder, there are three input bits (A, B, C-in).
Outputs	The generated output is of two bits-Sum and Carry from the input of 2 bits.	The generated output is of two bits-Sum and Carry from the input of 3 bits.

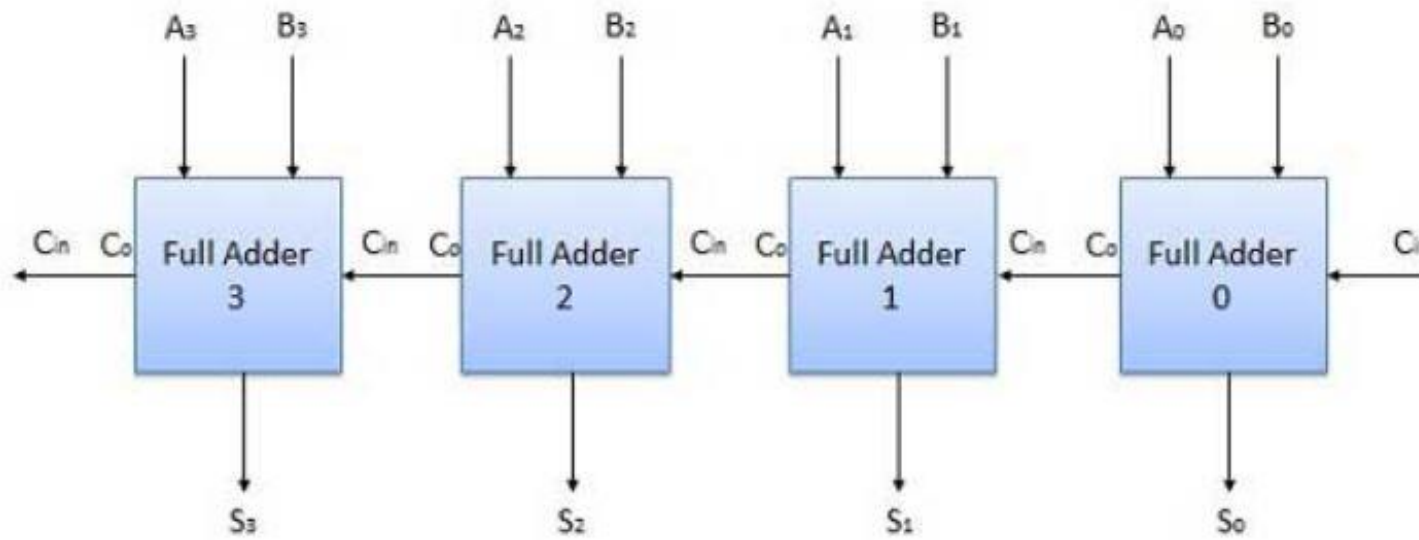
Used as	A half adder circuit cannot be used in the same way as a full adder circuit.	A full adder circuit can be used in place of a half adder circuit.
Feature	It is simple and easy to implement	The design of a full adder is not as simple as a half adder.
Logical Expression	Logical Expression for half adder is : $S = a \oplus b$; $C = a * b$.	Logical Expression for Full adder is : $S = a \oplus b \oplus C_{in}$; $C_{out} = (a * b) + (C_{in} * (a \oplus b))$.
Logic gates	It consists of one EX-OR gate and one AND gate.	It consists of two EX-OR, two AND gates, and one OR gate.
Applications	It is used in Calculators, computers, digital measuring devices, etc.	It is used in Multiple bit addition, digital processors, etc.

N-Bit Parallel Adder

- ✓ The Full Adder is capable of adding only two single digit binary number along with a carry input.
- ✓ But in practical we need to add binary numbers which are much longer than just one bit.
- ✓ To add two n-bit binary numbers we need to use the n-bit parallel adder.
- ✓ It uses a number of full adders in cascade.
- ✓ The carry output of the previous full adder is connected to carry input of the next full adder.

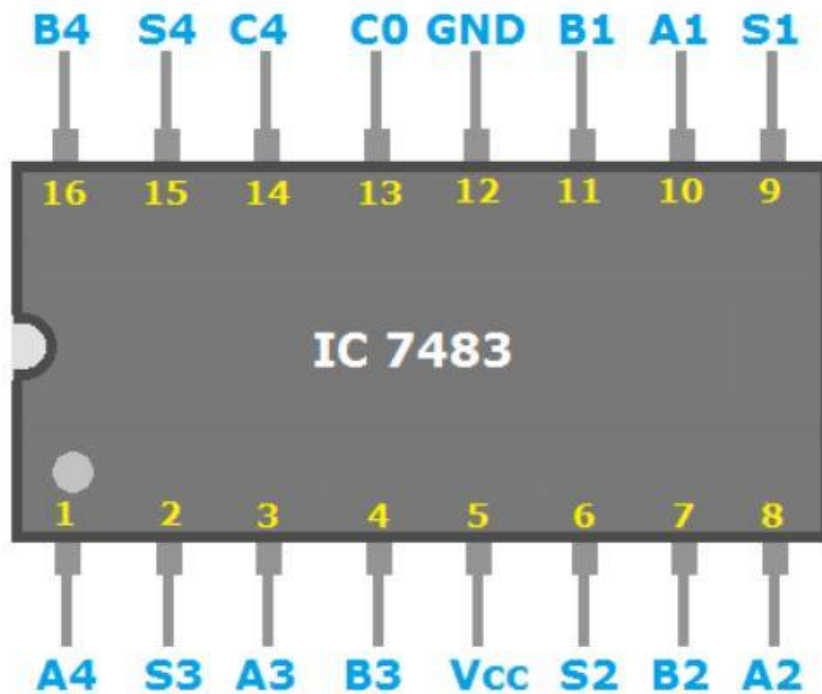
4 Bit Parallel Adder

- Full Adder is capable of adding only two single digit binary number along with a carry input.
- But in practical we need to add binary numbers which are much longer than just one bit.
- To add two n-bit binary numbers we need to use the n-bit parallel adder.
- It uses a number of full adders in cascade. The carry output of the previous full adder is connected to carry input of the next full adder.
- In the block diagram, A_0 and B_0 represent the LSB of the four bit words A and B.
- Hence Full Adder-0 is the lowest stage. Hence its C_{in} has been permanently made 0.
- The four bit parallel adder is a very common logic circuit.



- ✓ Interconnection of four full adder circuits to provide a 4 bit parallel adder
- ✓ Carries are connected in a chain through the full adders
- ✓ S outputs generate the required sum bits
- ✓ When the 4 bit full adder circuit is enclosed within an IC package, it has 4 terminals for the augend bits, four terminals for the addend bits, for terminals for the sum bits and two terminals for the input and output carries
- ✓ N bit parallel adder requires N full adders

IC 7483 PIN CONFIGURATION

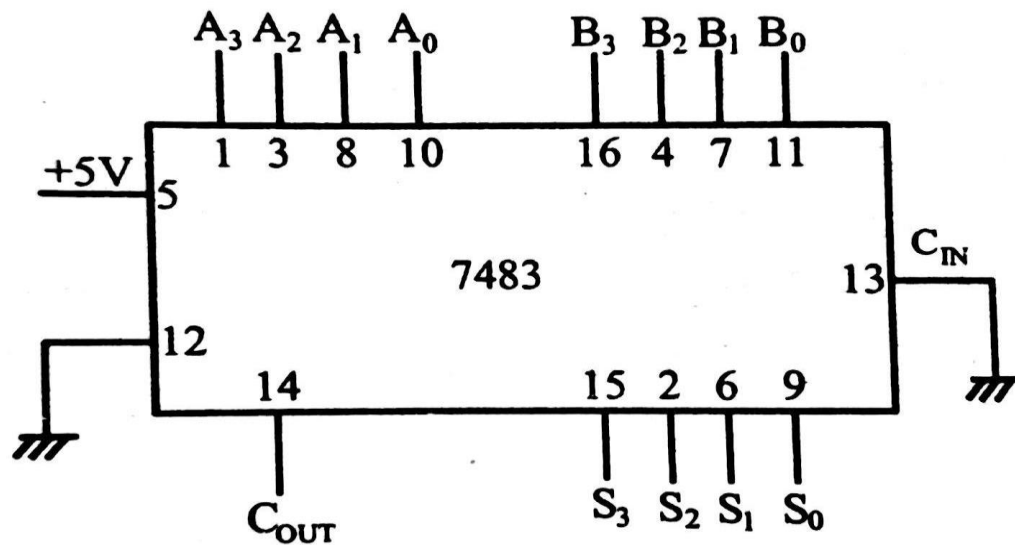


IC 7483 Pin Diagram

Input Data A				Input Data B				Addition				
A4	A3	A2	A1	B4	B3	B2	B1	Cout	S4	S3	S2	S1
1	0	0	0	0	0	1	0	0	1	0	1	0
1	0	0	0	1	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	1	0	1	0
0	0	0	1	0	1	1	1	0	1	0	0	0
1	0	1	0	1	0	1	1	1	0	1	0	1
0	1	1	0	0	0	1	1	0	1	0	0	1
1	1	1	0	1	1	1	1	1	1	1	0	1
1	0	1	0	1	1	0	1	1	0	1	1	1

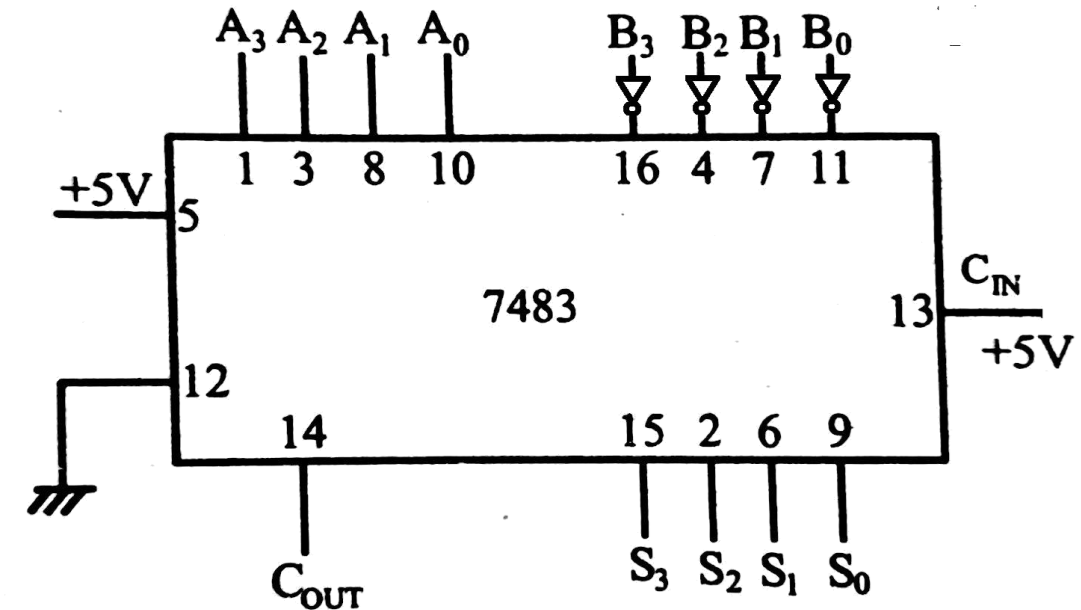
4 bit binary adder

- ✓ A binary adder can be constructed with full adders connected in series with the output carry from each adder connected to the next full adder in the chain



4 bit binary subtractor

- ✓ It can be constructed with full adders with inverters placed between each data input B and the corresponding input of full adder
- ✓ Input carry must be equal to 1 when performing subtraction

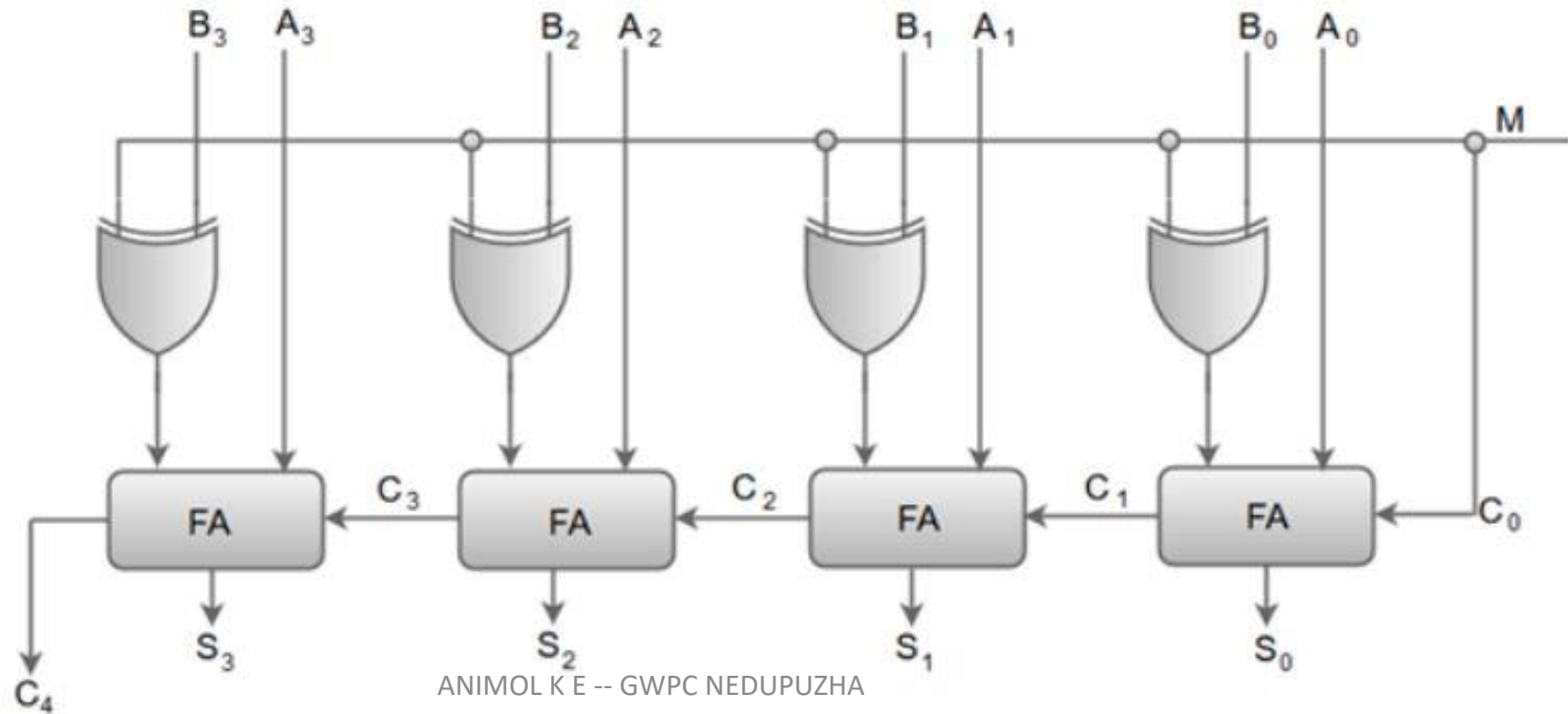


Binary Adder-Subtractor

- ✓ In Digital Circuits, A **Binary Adder-Subtractor** is one which is capable of both addition and subtraction of binary numbers in one circuit itself.
- ✓ The operation being performed depends upon the binary value the control signal holds. It is one of the components of the ALU (Arithmetic Logic Unit).
- ✓ This Circuit Requires prerequisite knowledge of Exor Gate, Binary Addition and Subtraction, Full Adder.
- ✓ Lets consider two 4-bit binary numbers A and B as inputs to the Digital Circuit for the operation with digits
- ✓ A0 A1 A2 A3 for A
- ✓ B0 B1 B2 B3 for B

- ✓ The circuit consists of 4 full adders for performing operation on 4-bit numbers.
- ✓ There is a control line M that holds a binary value of either 0 or 1 which determines that the operation being carried out is addition or subtraction
- ✓ The mode input M controls the operation
- ✓ when $M = 0$ the circuit is an adder and $M = 1$, the circuit becomes subtractor

4 bit adder-subtractor:



Binary Multiplier

- ✓ A **binary multiplier** is a combinational logic circuit or digital device used for **multiplying two binary numbers**.
- ✓ The two numbers are more specifically known as **multiplicand** and **multiplier** and the result is known as a **product**.
- ✓ The multiplicand & multiplier can be of various bit size. The product's bit size depends on the bit size of the multiplicand & multiplier
- ✓ Binary multiplication method is same as decimal multiplication. Binary multiplication of more than 1-bit numbers contains 2 steps.
- ✓ The 1st step is single bit-wise multiplication known as partial product and the 2nd step is adding all partial products into a single product.

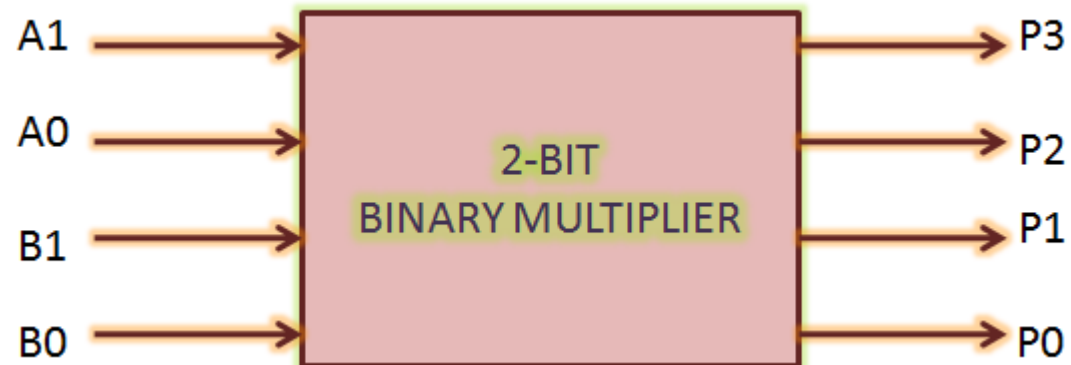
$$\begin{array}{r} 110 = 6 \\ \times 011 = 3 \\ \hline 110 \quad ; 110 \times 1 \\ 110x \quad ; 110 \times 1 \\ 000xx \quad ; 110 \times 0 \\ \hline 10010 = 18 \end{array}$$

Types of Binary Multipliers:

- **2×2 Bit Multiplier**
- **3×3 Bit Multiplier**
- **4×4 Bit Multiplier**

2×2 Bit Multiplier

This multiplier can multiply two numbers having bit size = 2 i.e. the multiplier and multiplicand can be of 2 bits. The product bit size will be the sum of the bit size of the input i.e. **2+2=4**.



Binary Multiplier

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

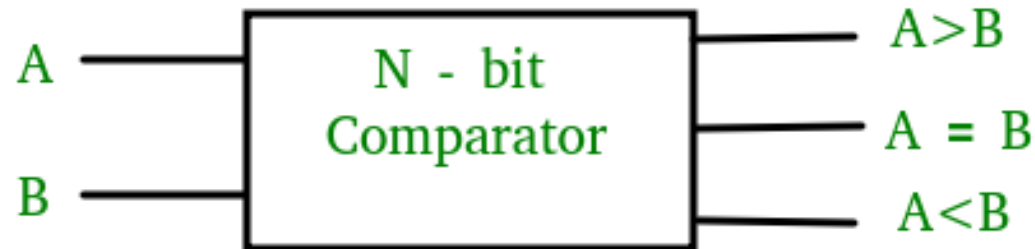
$$1 \times 1 = 1$$

- ✓ There are two binary digits A1A0 and B1B0
- ✓ A1A0 is the multiplicand. B1B0 is the multiplier.
- ✓ The first product obtained from multiplying B0 with the multiplicand is called as partial product 1 and the second product obtained from multiplying B1 with the multiplicand is known as the partial product 2
- ✓ As the number of bits increases, keep shifting each successive partial product to the left by 1 bit.
- ✓ In the end, we add the digits while keeping in mind the carry that might generate.

A1	A0	
B1	B0	
<hr/>		
	A1B0	A0B0
A1B1	A0B1	X
<hr/>		
A1B1+C	A0B1+A1B0	A0B0

Magnitude Comparator

- ✓ A magnitude digital Comparator is a combinational circuit that **compares two digital or binary numbers** in order to find out whether one binary number is equal, less than, or greater than the other binary number.
- ✓ Circuit consists of two inputs one for A and the other for B and have three output terminals, one for $A > B$ condition, one for $A = B$ condition, and one for $A < B$ condition.



7.18.1 1-bit Magnitude Comparator

The logic for a 1-bit magnitude comparator: Let the 1-bit numbers be $A = A_0$ and $B = B_0$.

If $A_0 = 1$ and $B_0 = 0$, then $A > B$.

Therefore,

$$A > B: G = A_0 \bar{B}_0$$

If $A_0 = 0$ and $B_0 = 1$, then $A < B$.

Therefore,

$$A < B: L = \bar{A}_0 B_0$$

If A_0 and B_0 coincide, i.e. $A_0 = B_0 = 0$ or if $A_0 = B_0 = 1$, then $A = B$.

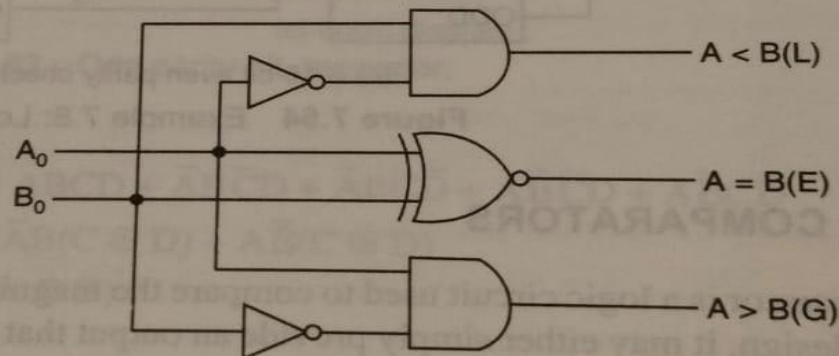
Therefore,

$$A = B: E = A_0 \odot B_0$$

The truth table and the logic diagram for the 1-bit comparator are shown in Figure 7.56. The logic expressions for G , L , and E can also be obtained from the truth table.

A_0	B_0	L	E	G
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

(a) Truth table



(b) Logic diagram

2-Bit Magnitude Comparator:

- ✓ A comparator used to compare two binary numbers each of two bits is called a 2-bit Magnitude comparator.
- ✓ It consists of four inputs and three outputs to generate less than, equal to, and greater than between two binary numbers.

INPUT				OUTPUT		
A1	A0	B1	B0	A<B	A=B	A>B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

7.18.2 2-bit Magnitude Comparator

The logic for a 2-bit magnitude comparator: Let the two 2-bit numbers be $A = A_1A_0$ and $B = B_1B_0$.

1. If $A_1 = 1$ and $B_1 = 0$, then $A > B$ or
2. If A_1 and B_1 coincide and $A_0 = 1$ and $B_0 = 0$, then $A > B$. So the logic expression for $A > B$ is

$$A > B : G = A_1\bar{B}_1 + (A_1 \odot B_1)A_0\bar{B}_0$$

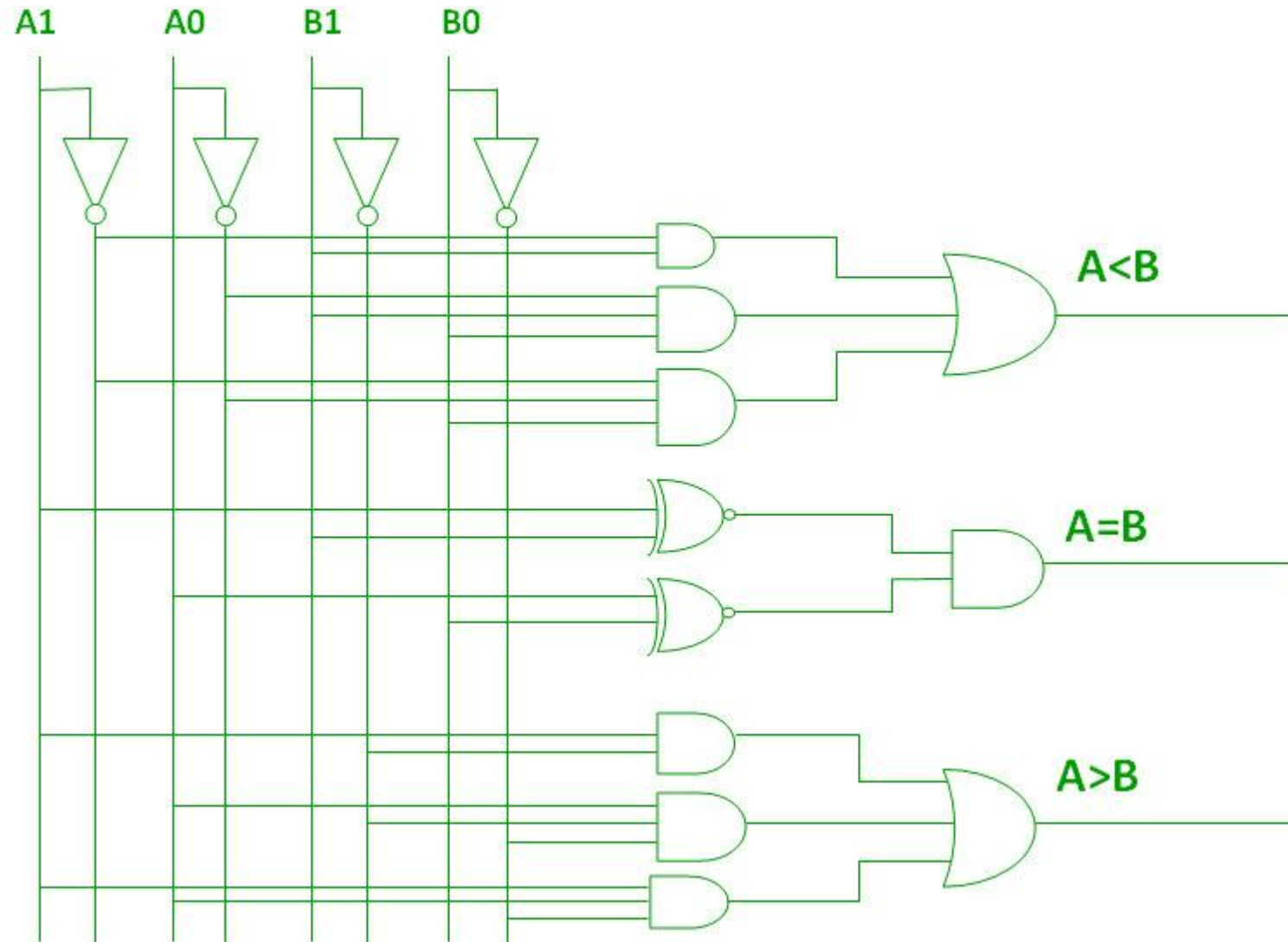
1. If $A_1 = 0$ and $B_1 = 1$, then $A < B$ or
2. If A_1 and B_1 coincide and $A_0 = 0$ and $B_0 = 1$, then $A < B$. So the expression for $A < B$ is

$$A < B : L = \bar{A}_1B_1 + (A_1 \odot B_1)\bar{A}_0B_0$$

If A_1 and B_1 coincide and if A_0 and B_0 coincide then $A = B$. So the expression for $A = B$ is

$$A = B : E = (A_1 \odot B_1)(A_0 \odot B_0)$$

2-Bit Magnitude Comparator:



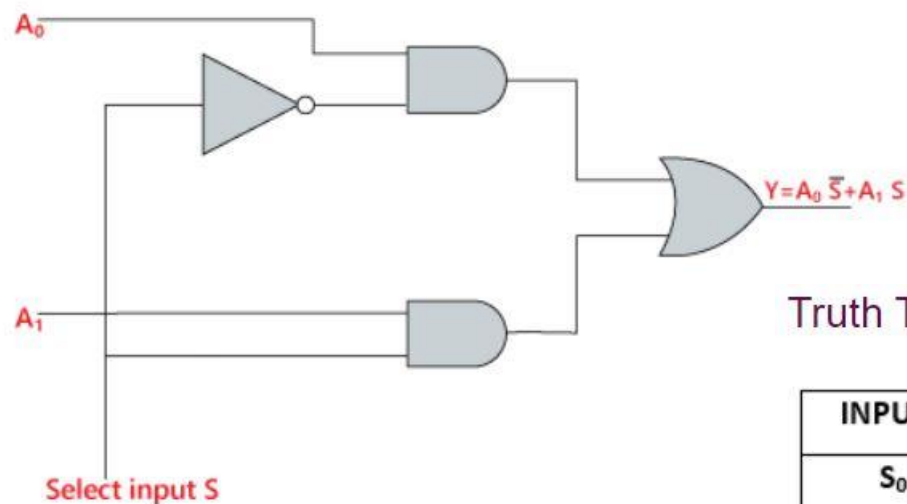
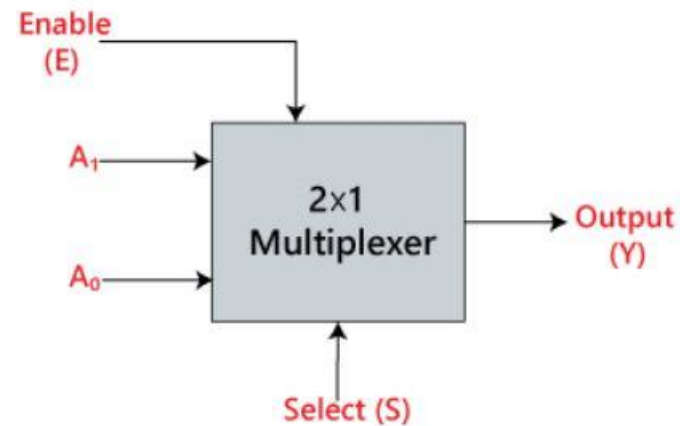
MULTIPLEXER

- A multiplexer is a combinational circuit also called as data selector, that has 2^n input lines and a single output line. Simply, the multiplexer is a multi-input and single-output combinational circuit.
- The binary information is received from the input lines and directed to the output line. On the basis of the values of the selection lines, one of these data inputs will be connected to the output.
- It is a Combinational Logic Circuit with more than one input line, one output line and more than one select line.
- It accepts the binary information from several input lines or sources and depending on the set of select lines, a particular input line is routed onto a single output line.

2: 1 Multiplexer

- A 2-to-1 multiplexer consists of two inputs A0 and A1, one select input S and one output Y.
- Depending on the select signal, the output is connected to either of the inputs.

- ✓ The logic level applied to the S input determines which AND gate is enabled, so that its data input passes through the OR gate to the output. $Y = S_0' \cdot A_0 + S_0 \cdot A_1$
- ✓ When $S=0$, AND gate 1 is enabled and AND gate 2 is disabled so $Y=A_0$
- ✓ When $S=1$, AND gate 2 is enabled and AND gate 1 is disabled so $Y=A_1$



Truth Table:

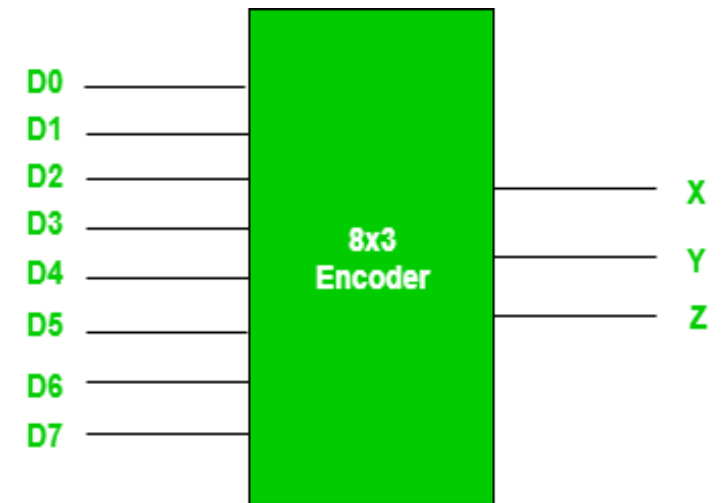
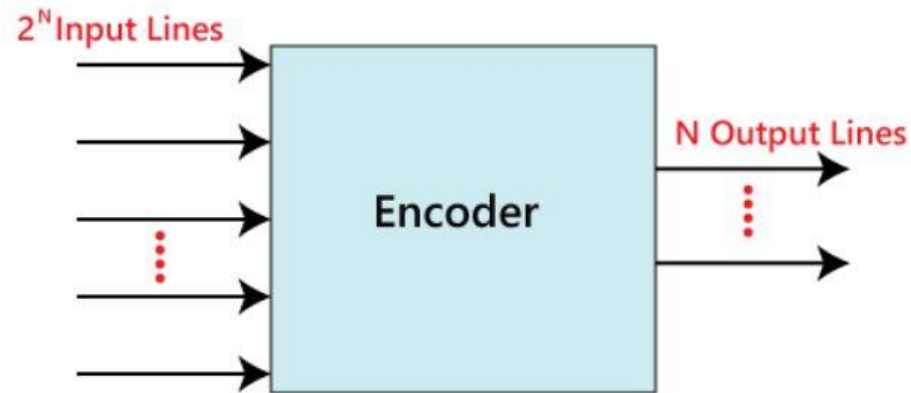
INPUTS	Output
S_0	Y
0	A_0
1	A_1

ENCODER and DECODER

- Binary code of N digits can be used to store 2^N distinct elements of coded information. This is what encoders and decoders are used for.
- Encoders convert 2^N lines of input into a code of N bits and Decoders decode the N bits into 2^N lines.

ENCODER

- An encoder is a combinational circuit that converts binary information in the form of a 2^N input lines into N output lines, which represent N bit code for the input. For simple encoders, it is assumed that only one input line is active at a time.



- As an example, let's consider **Octal to Binary** encoder. As shown in the following figure, an octal-to-binary encoder takes 8 input lines and generates 3 output lines.

Truth Table –

D7	D6	D5	D4	D3	D2	D1	D0	X	Y	Z
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

As seen from the truth table, the output is 000 when D0 is active; 001 when D1 is active; 010 when D2 is active and so on.

ENCODER

- The combinational circuits that change the binary information into N output lines are known as **Encoders**.
- The binary information is passed in the form of 2^N input lines.
- The output lines define the N-bit code for the binary information
- At a time, only one input line is activated for simplicity. The produced N-bit output code is equivalent to the binary information.

4 to 2 line Encoder:

- ✓ In 4 to 2 line encoder, there are total of four inputs, i.e., Y_0 , Y_1 , Y_2 , and Y_3 , and two outputs, i.e., A_0 and A_1 .
- ✓ In 4-input lines, one input-line is set to true at a time to get the respective binary code in the output side.
- ✓ The 4 to 2 Encoder consists of **four inputs Y_3 , Y_2 , Y_1 & Y_0 and two outputs A_1 & A_0** .
- ✓ At any time, only one of these 4 inputs can be '1' in order to get the respective binary code at the output.

$$A_1 = Y_3 + Y_2$$

$$A_0 = Y_3 + Y_1$$



INPUTS				OUTPUTS	
Y3	Y2	Y1	Y0	A1	A0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

- ✓ One of the main disadvantages of standard digital encoders is that they can generate the wrong output code when there is more than one input present at logic level “1”.
- ✓ For example, if we make inputs Y_1 and Y_2 HIGH at logic “1” both at the same time, the resulting output is neither at “01” or at “10” but will be at “11” which is an output binary number that is different to the actual input present.
- ✓ Also, an output code of all logic “0”s can be generated when all of its inputs are at “0” OR when input Y_0 is equal to one.
- ✓ Normal encoder will not work when more than one input to the encoder are high
- ✓ One simple way to overcome this problem is to “Prioritise” the level of each input pin.
- ✓ So if there is more than one input at logic level “1” at the same time, the actual output code would only correspond to the input with the highest designated priority.
- ✓ Then this type of digital encoder is known commonly as a **Priority Encoder** or **P-encoder** for short.

PRIORITY ENCODER

- The **Priority Encoder** solves the problems mentioned above by allocating a priority level to each input.
- The *priority encoders* output corresponds to the currently active input which has the highest priority.
- So when an input with a higher priority is present, all other inputs with a lower priority will be ignored.

Applications

- It is used to reduce the no. of wires and connections required for electronic circuit designing that have multiple input lines. Example keypads and keyboards.
- Used in controlling the position in the ship's navigation and robotics arm position.
- Used in the detection of highest priority input in various applications of microprocessor interrupt controllers.

4 to 2 Priority Encoder

This is also referred to as 4-bit priority, which consists of 4 inputs and 2 output lines. Since an encoder contains 2^n input lines and n output lines. The third output is 'V', which is considered as a valid bit indicator and it is set to 1 when more than one input line is high or active (1).

If the valid bit is equal to '0', then all the inputs are '0'. In this case, the other 2 output lines are considered as don't care conditions denoted by 'X'

D3	D2	D1	D0	A	B	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

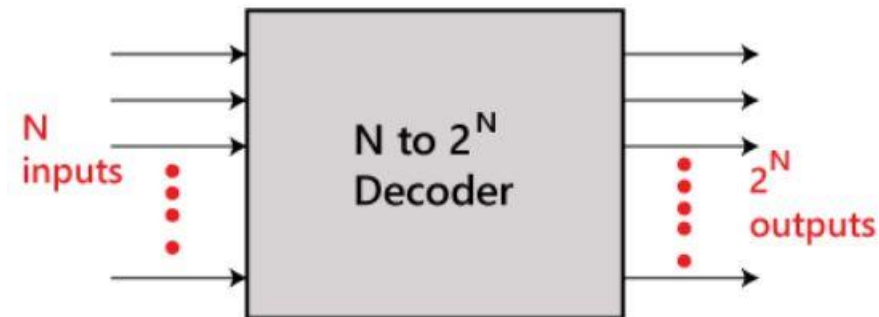
$$A = D2D3' + D3$$

$$B = D2'D1 + D3$$

$$V = D0 + D1 + D2 + D3$$

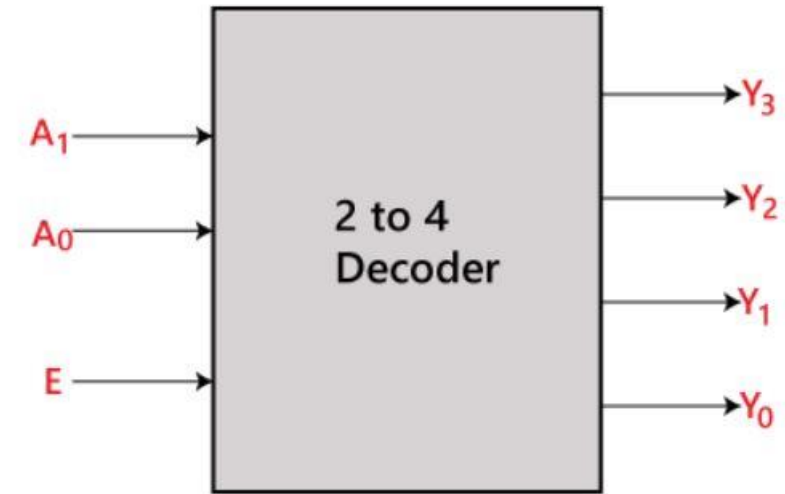
DECODER

- **Decoder** is a combinational circuit that has 'n' input lines and maximum of 2^n 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 lines when it is enabled
- At a time, only one input line is activated for simplicity. The produced 2^N -bit output code is equivalent to the binary information.



2 to 4 line decoder

- ✓ In the 2 to 4 line decoder, there is a total of three inputs, i.e., A_0 , and A_1 and E and four outputs, i.e., Y_0 , Y_1 , Y_2 , and Y_3 .
- ✓ For each combination of inputs, when the enable 'E' is set to 1, one of these four outputs will be 1.



Truth Table:

Enable	INPUTS		OUTPUTS			
	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

$$Y_3 = E \cdot A_1 \cdot A_0$$

$$Y_2 = E \cdot A_1 \cdot A_0'$$

$$Y_1 = E \cdot A_1' \cdot A_0$$

$$Y_0 = E \cdot A_1' \cdot A_0'$$