AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-2 Lec-1



Avinash Sir

Today's target

- Analysis and Design of combinational circuits
- Types of Adders
- PYQ
- DPP

Gateway Classes

AKTU : *DIGITAL SYSTEM DESIGN (BEC-302)* Syllabus

UNIT-2 : Combinational circuits:

- Analysis and Design of combinational circuits
- MSI devices like comparators, multiplexers, demultiplexers, encoder, decoder,
- circuit realization using Multiplexers and decoders, half and full adders, subtractors, serial and parallel adders, BCD adder, barrel shifter and ALU.

Gateway Classes

Combinational Circuits

AKTU-2023-24

A combinational circuit is formed by the interconnection of logic gates. The output of the circuit depends only on the present values of the input signals. These circuits process binary input data and generate the required binary output data.



The circuit takes n input variables from an external source and produces m output variables, which are delivered to an external destination through the internal logic network.

A combinational circuit can be fully defined in two ways:

- By a truth table listing all possible input combinations and their corresponding outputs.
- By m Boolean functions, one for each output, expressed in terms of the n input variables.

- Examples: Adders, Subtractors, Multiplexers, Demultiplexers, Encoders, Decoders, Comparators.

Gateway Classes

Analysis Procedure

(Used when the circuit is already given, and we need to determine how it works.)

- Identify inputs and outputs of the circuit.
- Determine the Boolean function for each output in terms of the inputs by tracing logic gates.
- Simplify the Boolean expressions (using Boolean algebra or Karnaugh maps).
- Construct the Truth Table to list outputs for all possible input combinations.
- Interpret results to understand the circuit's function (e.g., is it an adder, comparator, etc.?).

Gateway Classes

Design Procedure

(Used when the required function is given, and we need to build the circuit.)

- Problem statement: Define the circuit's required behavior.
- List input and output variables:

  Let the number of inputs = n & number of outputs = m

- Form the Truth Table that relates inputs to outputs.
- Derive Boolean functions for each output variable from the truth table.
- Simplify Boolean expressions (using Boolean algebra, Karnaugh maps, or Quine–McCluskey method).
- Draw the logic diagram using the simplified Boolean expressions and logic gates.
- Implement and verify the design.

Gateway Classes

Design Procedure

(Used when the required function is given, and we need to build the circuit.)

- Problem statement: Define the circuit's required behavior.
- List input and output variables:

  Let the number of inputs = n & number of outputs = m

- Form the Truth Table that relates inputs to outputs.
- Derive Boolean functions for each output variable from the truth table.
- Simplify Boolean expressions (using Boolean algebra, Karnaugh maps, or Quine–McCluskey method).
- Draw the logic diagram using the simplified Boolean expressions and logic gates.
- Implement and verify the design.

Gateway Classes

Adder

- An Adder is a combinational circuit that performs binary addition. It is one of the basic building blocks of arithmetic circuits.
- There are mainly two basic types of adders:
  - Half Adder (HA)
  - Full Adder (FA)

Gateway Classes

Half -Adder

- A circuit that adds two 1-bit binary numbers (A and B). And has two outputs S(Sum) & Carry(C).

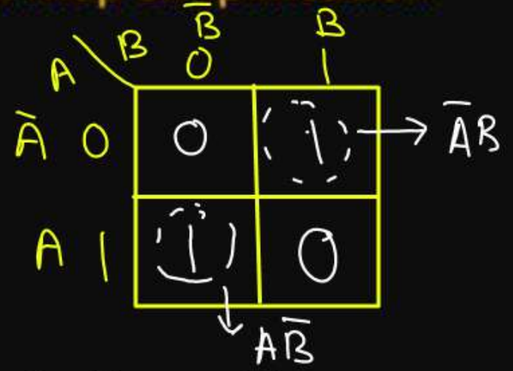


$$Sum\ (s) = A \oplus B = A\bar{B} + \bar{A}B$$

$$Carry\ (C) = AB$$

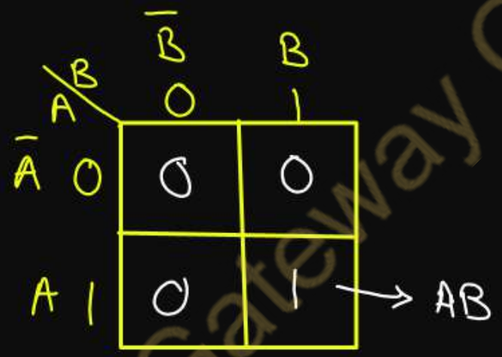
INPUTS		OUTPUTS	
A	B	Sum (S)	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- K Map Simplification



$$S = \bar{A}B + A\bar{B}$$

$$= A \oplus B$$



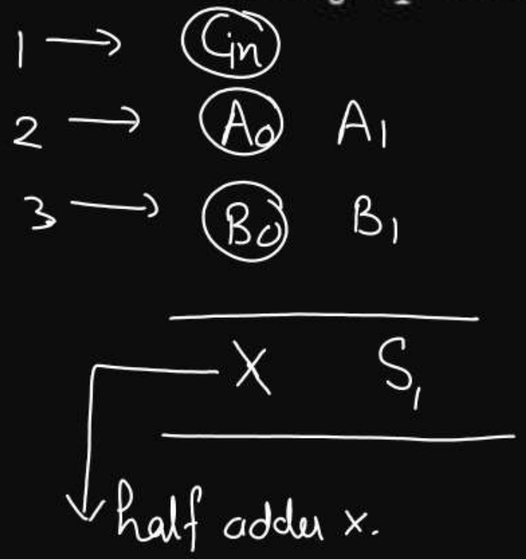
$$C = A \cdot B$$

Gateway Classes

Limitation of Half Adder

- Cannot add previous carry (works only for 2 inputs).
- Example Let we must ADD two 2- bit numbers

$A = A_0\ A_1$ and $B = B_0\ B_1$



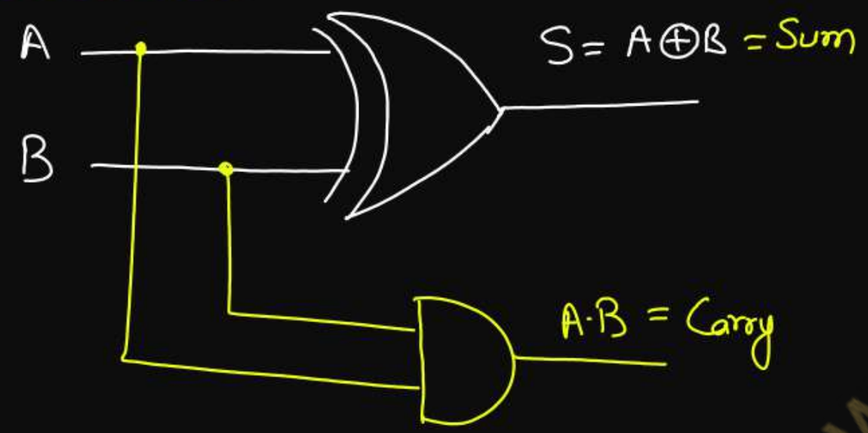
Gateway Classes

Logic Implementation

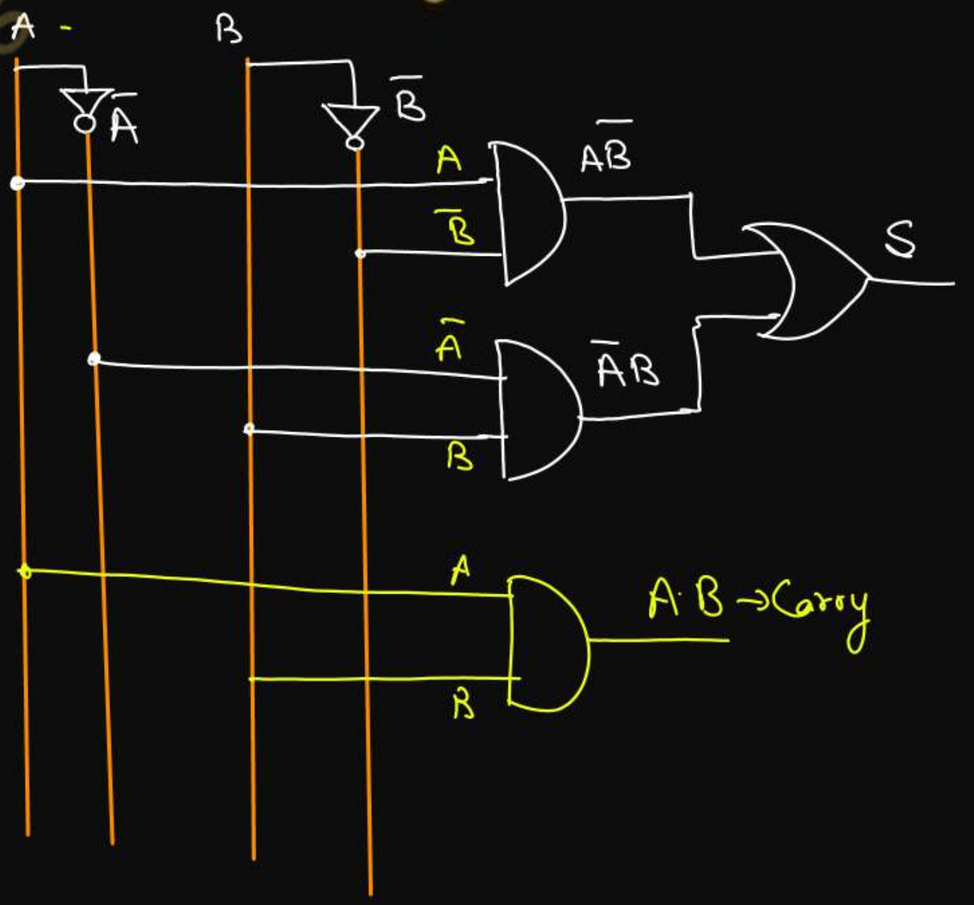
$S = A \oplus B = A\bar{B} + \bar{A}B$

$C = A \cdot B$

Using Ex-OR Gate



Using Basic Gates



Gateway Classes

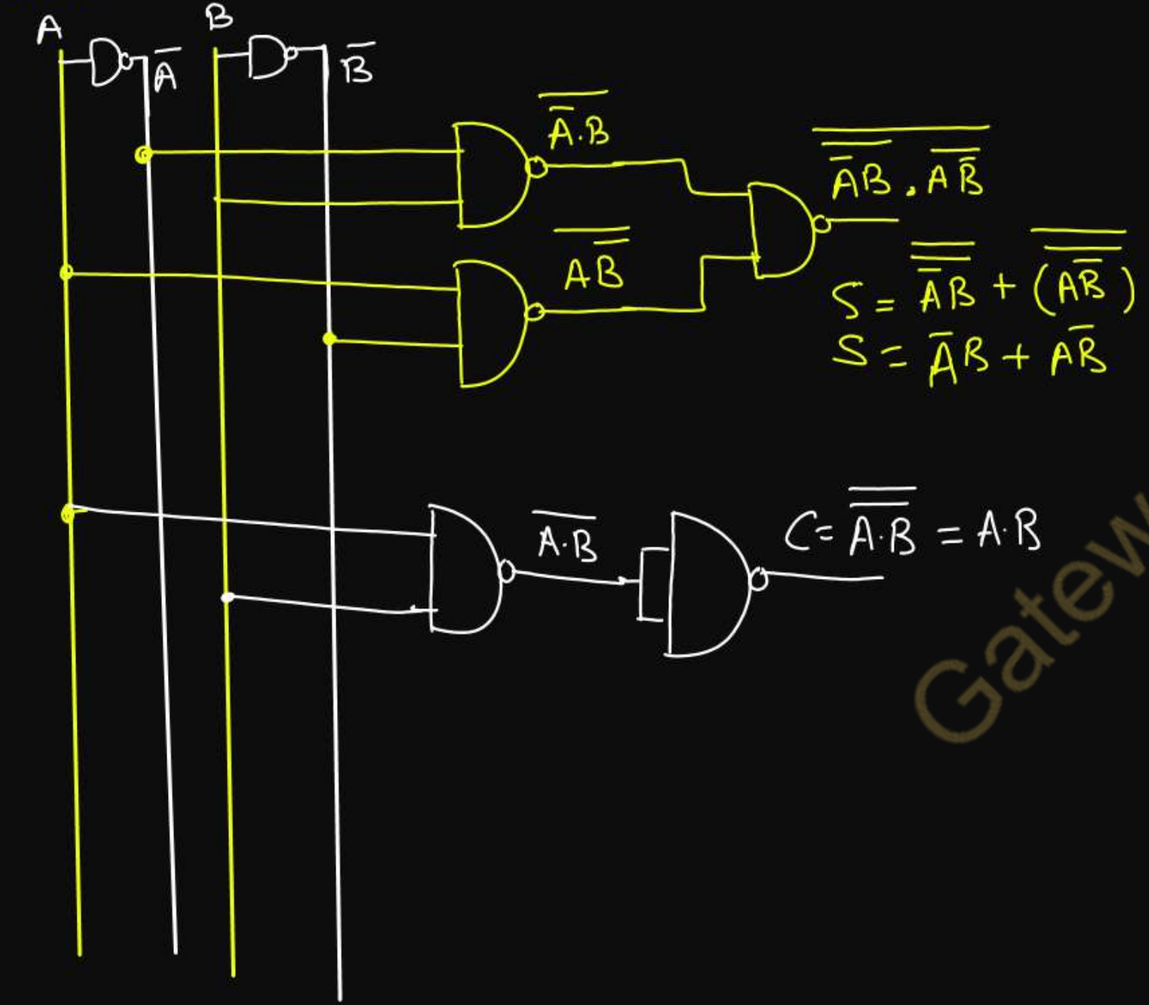
Logic Implementation

AKTU-2021-22

$C = A \cdot B$

Using NAND Gate

$S = \overline{\bar{A}B + A\bar{B}} = \overline{\bar{A}B} \cdot \overline{A\bar{B}}$

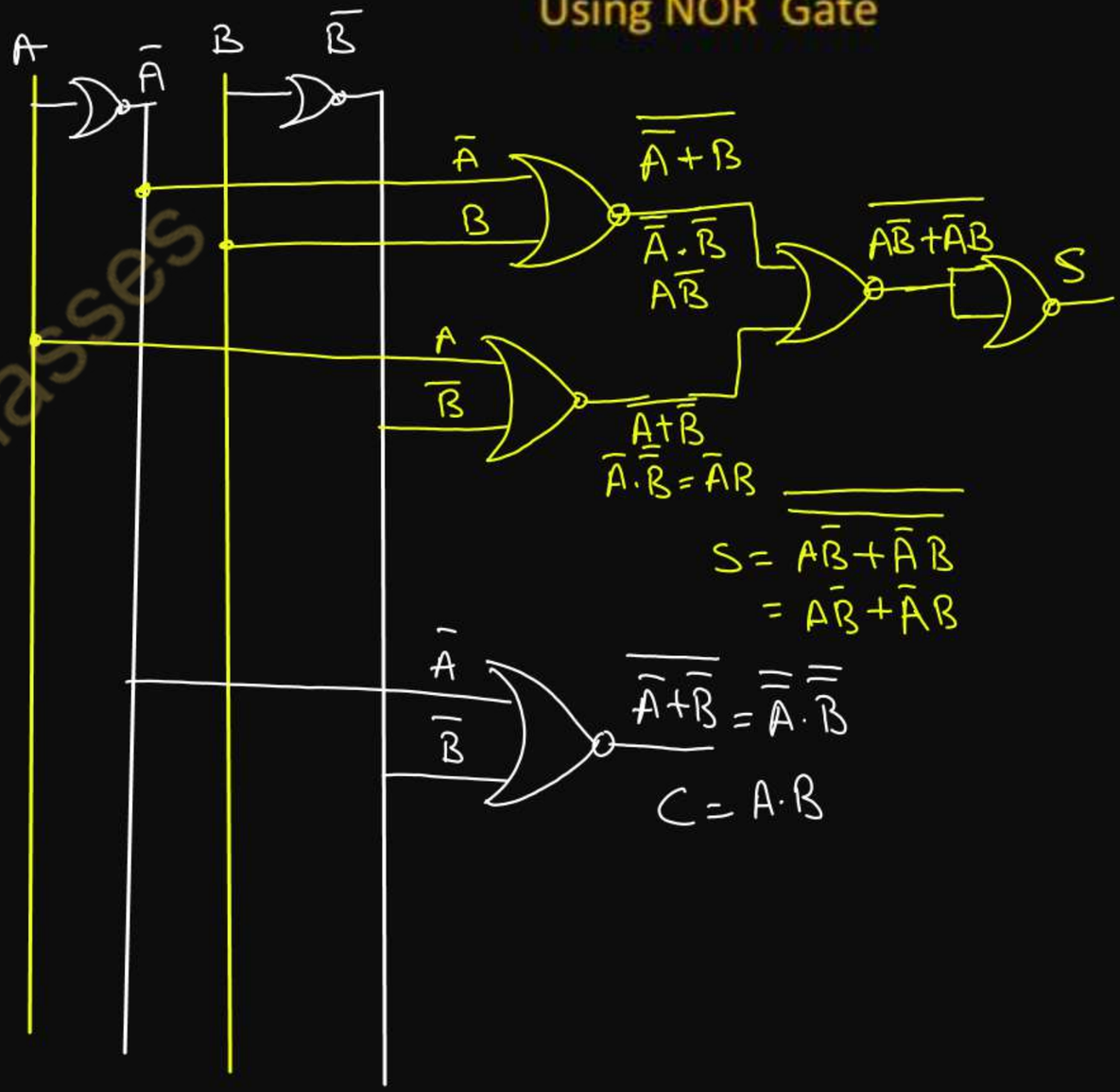


$S = \overline{\bar{A}B} + \overline{(A\bar{B})}$

$S = \bar{A}B + A\bar{B}$

$C = \overline{A \cdot B} = A \cdot B$

Using NOR Gate



$S = A\bar{B} + \bar{A}B$

$= A\bar{B} + \bar{A}B$

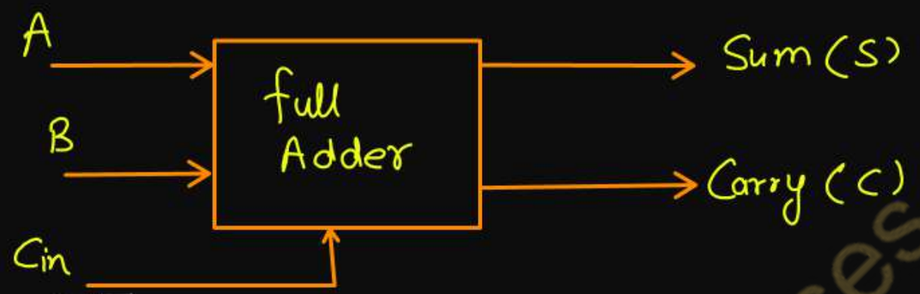
$C = A \cdot B$

Gateway Classes

Full Adder

AKTU-2023-24, 2021-22

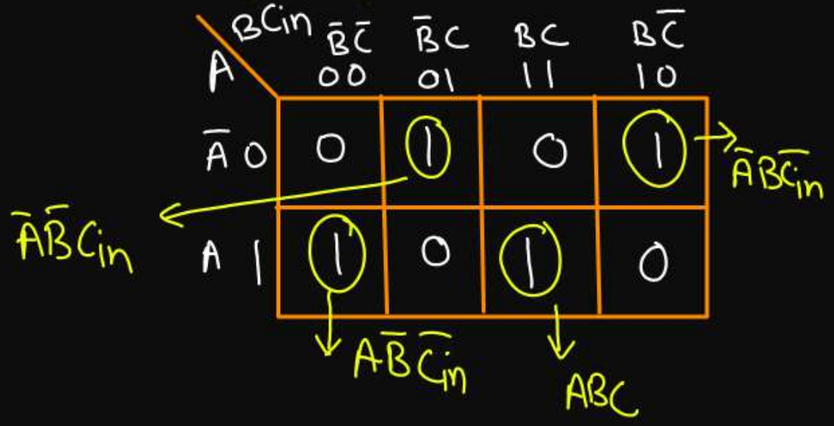
➤ A circuit that adds three 1-bit binary numbers (A, B, and Cin – previous carry) And has two outputs S(Sum) & Carry(C).

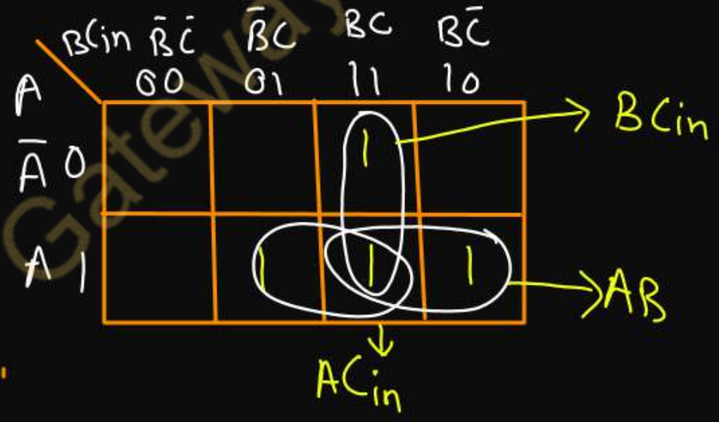


$$Sum(s) = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in} = A \oplus B \oplus C$$

$$Carry\ output\ (C) = AB + AC_{in} + BC_{in}$$

➤ K Map Simplification





INPUTS			OUTPUTS	
A	B	C_{in}	Sum (S)	Carry out (C)
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

Gateway Classes

$$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + AB\bar{C}_{in}$$

$$= C_{in}(\bar{A}\bar{B} + AB) + \bar{C}_{in}(\bar{A}B + A\bar{B})$$

$\bar{A}\bar{B} + AB$: X-NOR

$\bar{A}B + A\bar{B}$: XOR $= (A \oplus B)$

$$S = C_{in}(\bar{X}) + \bar{C}_{in}(X)$$

where $X = \bar{A}B + A\bar{B}$

$$S = C_{in}\bar{X} + \bar{C}_{in}X$$

$$= X \oplus C_{in}$$

$$\boxed{S = A \oplus B \oplus C_{in}}$$

X-OR = $\overline{\text{X-NOR}}$

$A \oplus B = \overline{A \odot B}$

$\bar{A}B + A\bar{B} = \overline{AB + \bar{A}\bar{B}}$

$= \overline{A\bar{B}} \cdot \overline{\bar{A}\bar{B}}$

$= (\bar{A} + \bar{B}) \cdot (\bar{\bar{A}} + \bar{\bar{B}})$

$= (\bar{A} + \bar{B}) \cdot (A + B)$

$= A\bar{A} + \bar{A}B + \bar{B}A + \bar{B}B$ → 0

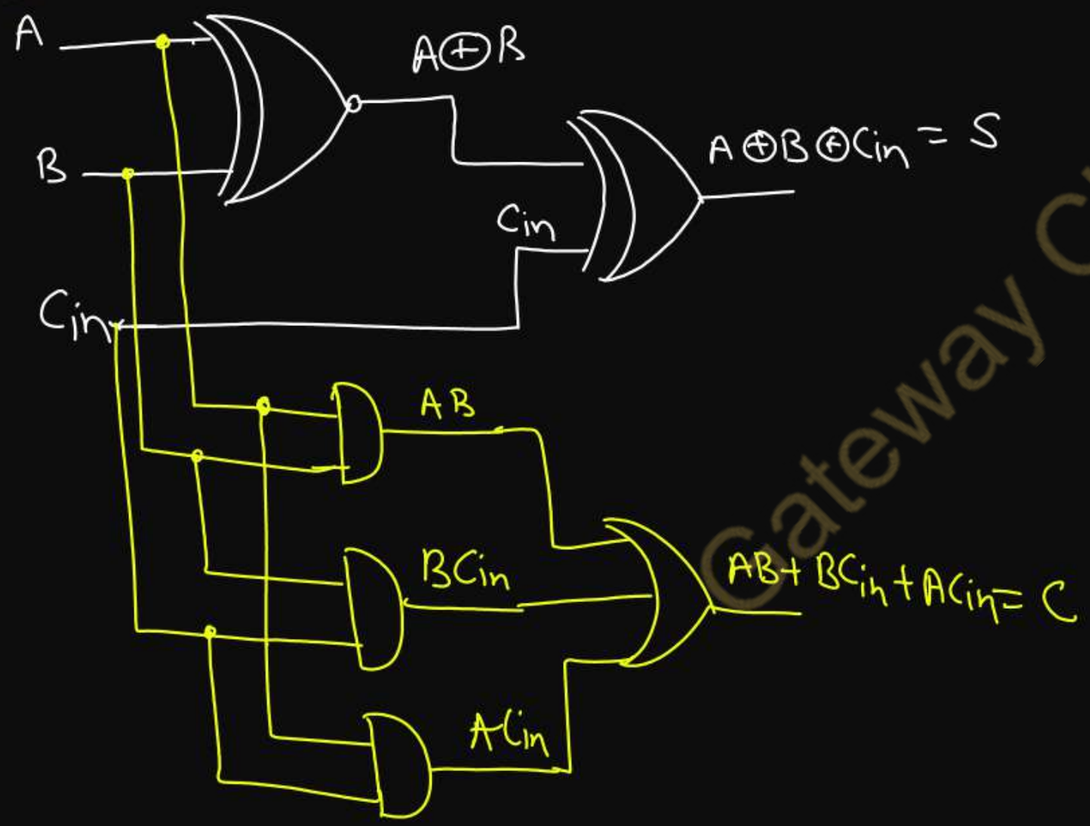
Gateway Classes

Logic Implementation

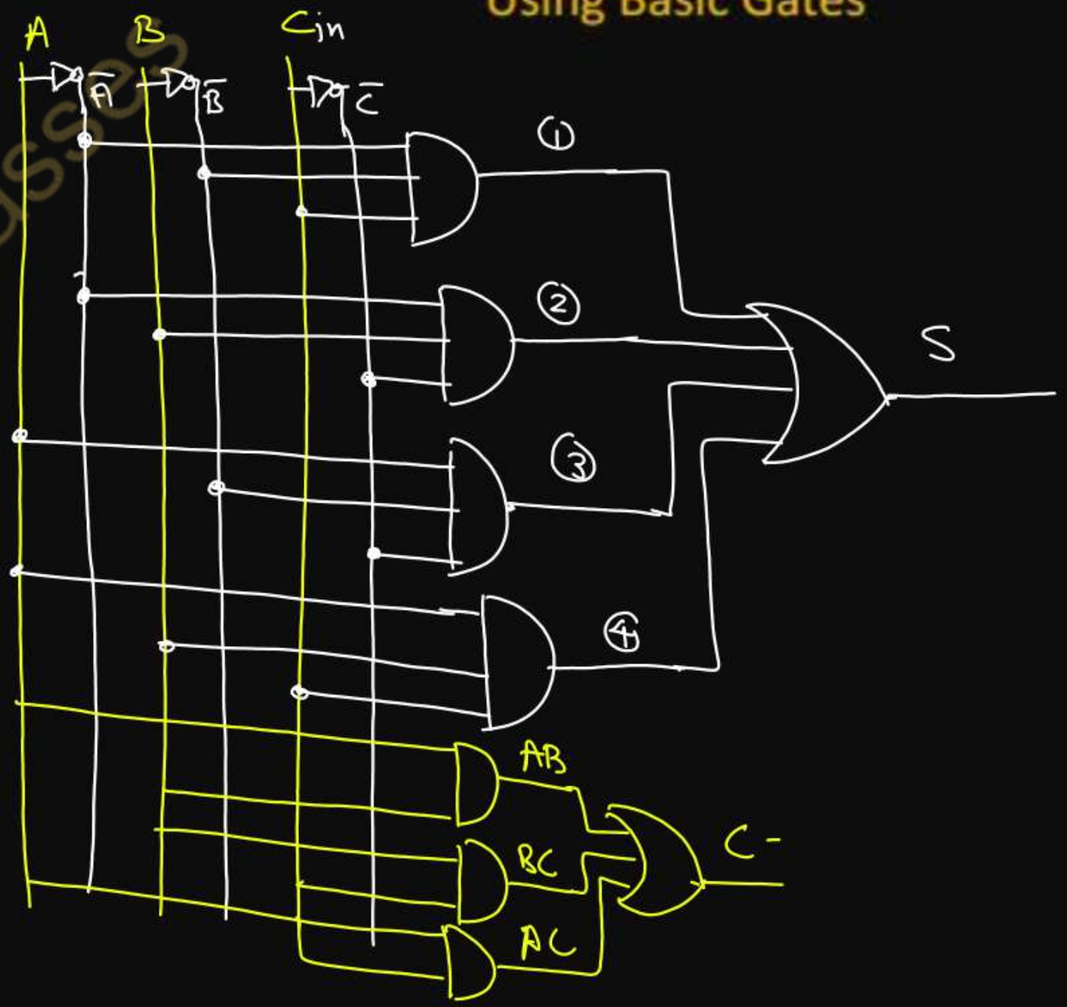
$$\text{Sum} = A \oplus B \oplus C_{in}$$
$$= \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$$
(terms ①, ②, ③, ④)

Carry output $C = AB + BC_{in} + AC_{in}$

Using Ex-OR Gate



Using Basic Gates



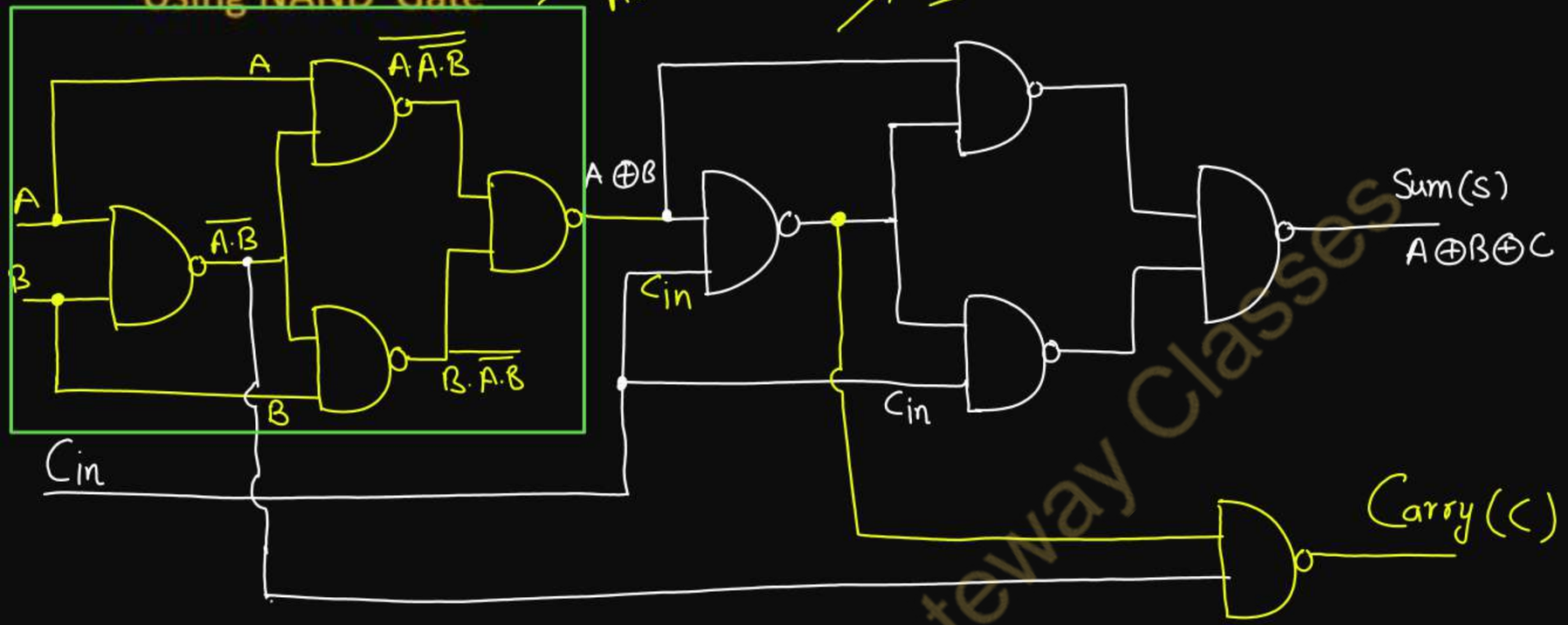
Gateway Classes

Logic Implementation

AKTU-2023-24

$S = A \oplus B \oplus C$

Using NAND Gate



$C = (\overline{A \cdot B}) \cdot \overline{(A \oplus B) C_{in}}$

$= AB + BC + AC$

Using NOR Gate

"Try yourself"

$S = \overline{(\overline{A \cdot A \cdot B}) \cdot (\overline{B \cdot A \cdot B})}$

$S = (\overline{A \cdot \overline{AB}}) + (\overline{B \cdot \overline{AB}})$

$= (A \cdot \overline{AB}) + (B \cdot \overline{AB})$

$= (A \cdot (\bar{A} + \bar{B})) + (B \cdot (\bar{A} + \bar{B}))$

$= (A\bar{A} + A\bar{B}) + (B\bar{A} + B\bar{B})$

$= A\bar{B} + B\bar{A}$

$A \oplus B$ ✗

Gateway Classes

Full Adder using NAND GATE

$$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$S = \overline{\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC}$$

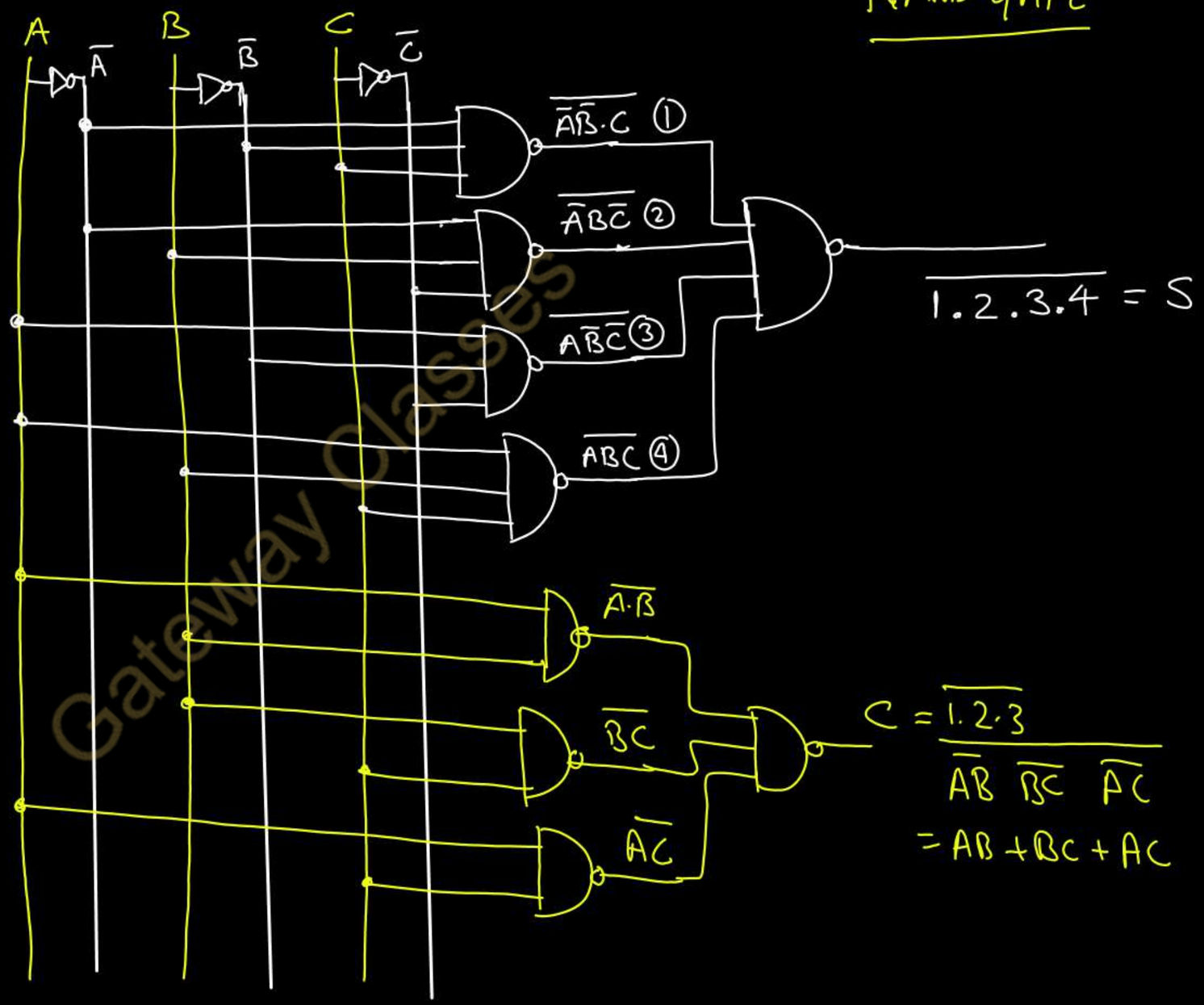
$$S = \overline{\bar{A}\bar{B}C} \cdot \overline{\bar{A}B\bar{C}} \cdot \overline{A\bar{B}\bar{C}} \cdot \overline{ABC}$$

① ② ③ 4

$$C = \overline{\overline{AB + BC + AC}}$$

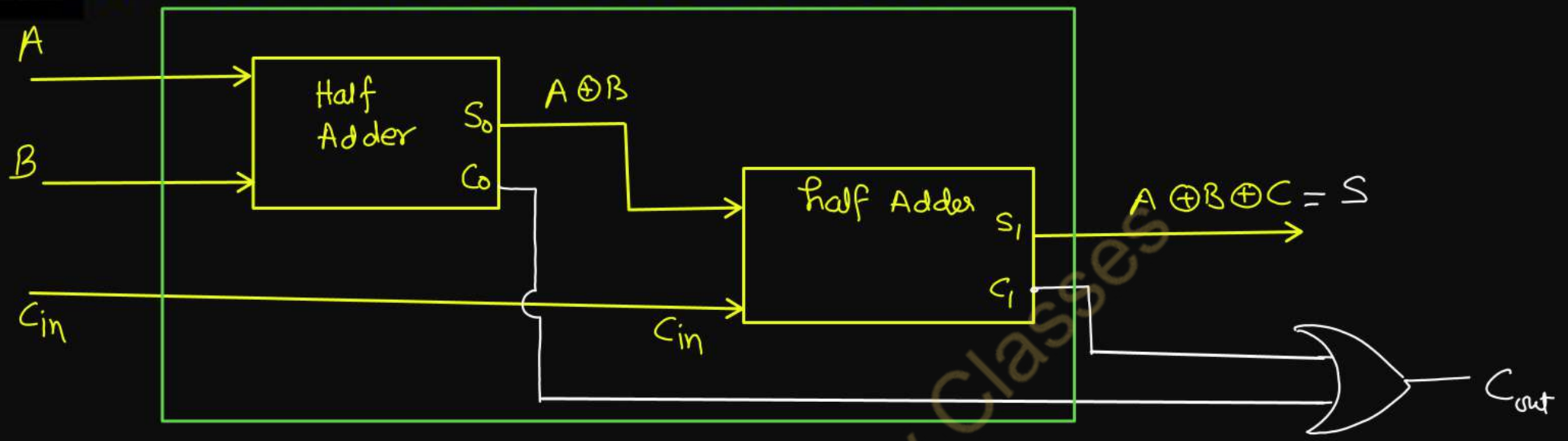
$$= \overline{\overline{AB} \cdot \overline{BC}\ \overline{AC}}$$

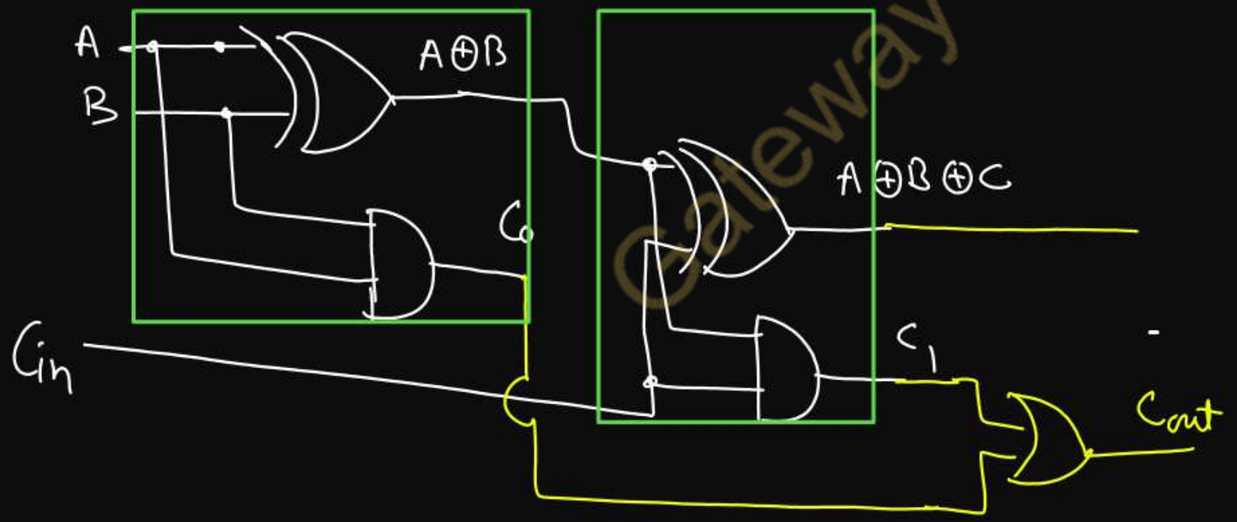
① ② ③



Gateway Classes

Logic Implementation for Full adder using half adder





Gateway Classes

AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-2 Lec-2

Gateway Classes



Today's target

- Types of Subtractors
- PYQ

Avinash Sir

AKTU : *DIGITAL SYSTEM DESIGN (BEC-302)* Syllabus

UNIT-2 : Combinational circuits:

- Analysis and Design of combinational circuits
- MSI devices like comparators, multiplexers, demultiplexers, encoder, decoder,
- circuit realization using Multiplexers and decoders, half and full adders, subtractors, serial and parallel adders, BCD adder, barrel shifter and ALU.

Gateway Classes

Subtractor

A subtractor is a combinational circuit that performs the arithmetic subtraction of two binary numbers.

Just like adders, there are two basic types:

- Half Subtractor
- Full Subtractor

Gateway Classes

Half Subtractor (HS)

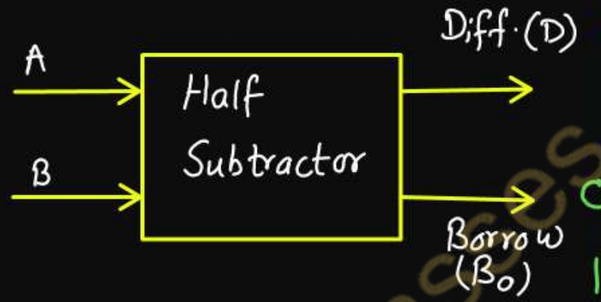
A half subtractor is a combinational circuit with two inputs and two outputs that subtracts one binary digit from another.

Inputs: A (minuend), B (subtrahend)

Outputs: Difference (D) Borrow (Bo)

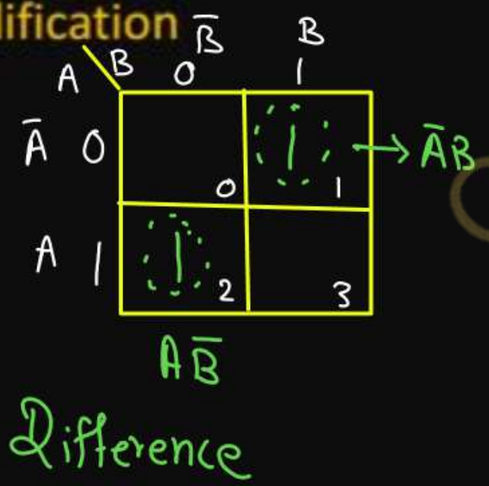
Difference $D = A\bar{B} + \bar{A}B = A \oplus B$

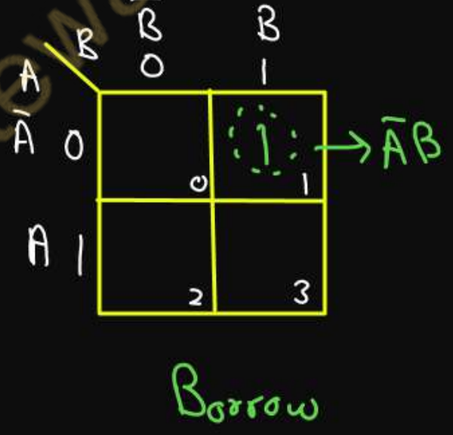
Borrow $B_o = \bar{A}B$



	INPUTS		OUTPUTS	
	A	B	Difference (D)	Borrow (B_0)
0	0	0	0	0
1	0	1	1	1
2	1	0	1	0
3	1	1	0	0

- K Map Simplification





Gateway Classes

Limitation of Half Subtractor

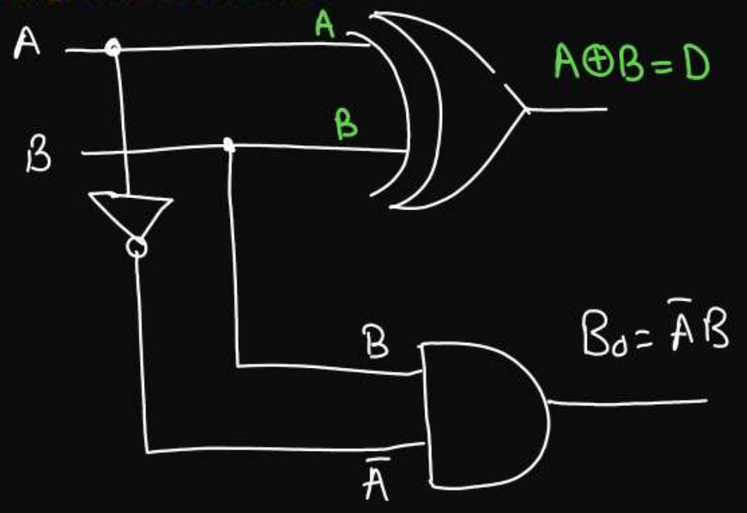
Half Subtractor does not consider a borrow from the previous stage → cannot be used alone for multi-bit subtraction.

Logic Implementation

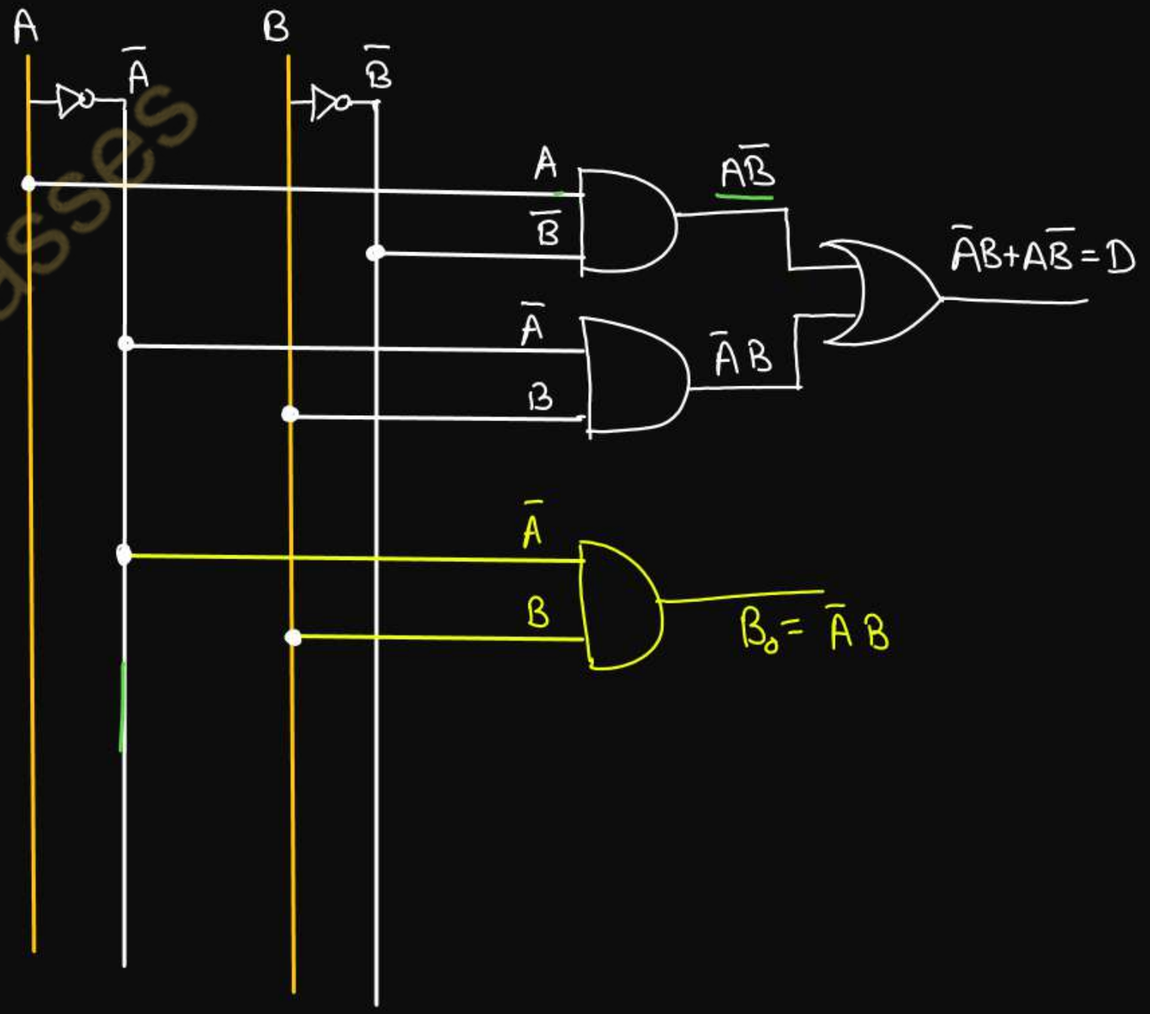
$$D = A\bar{B} + \bar{A}B = A \oplus B$$

$$B_0 = \bar{A}B$$

Using Ex-OR Gate



USING BASIC GATES



Gateway Classes

Logic Implementation Half Subtractor

$\bar{A}B$ $A\bar{B}$

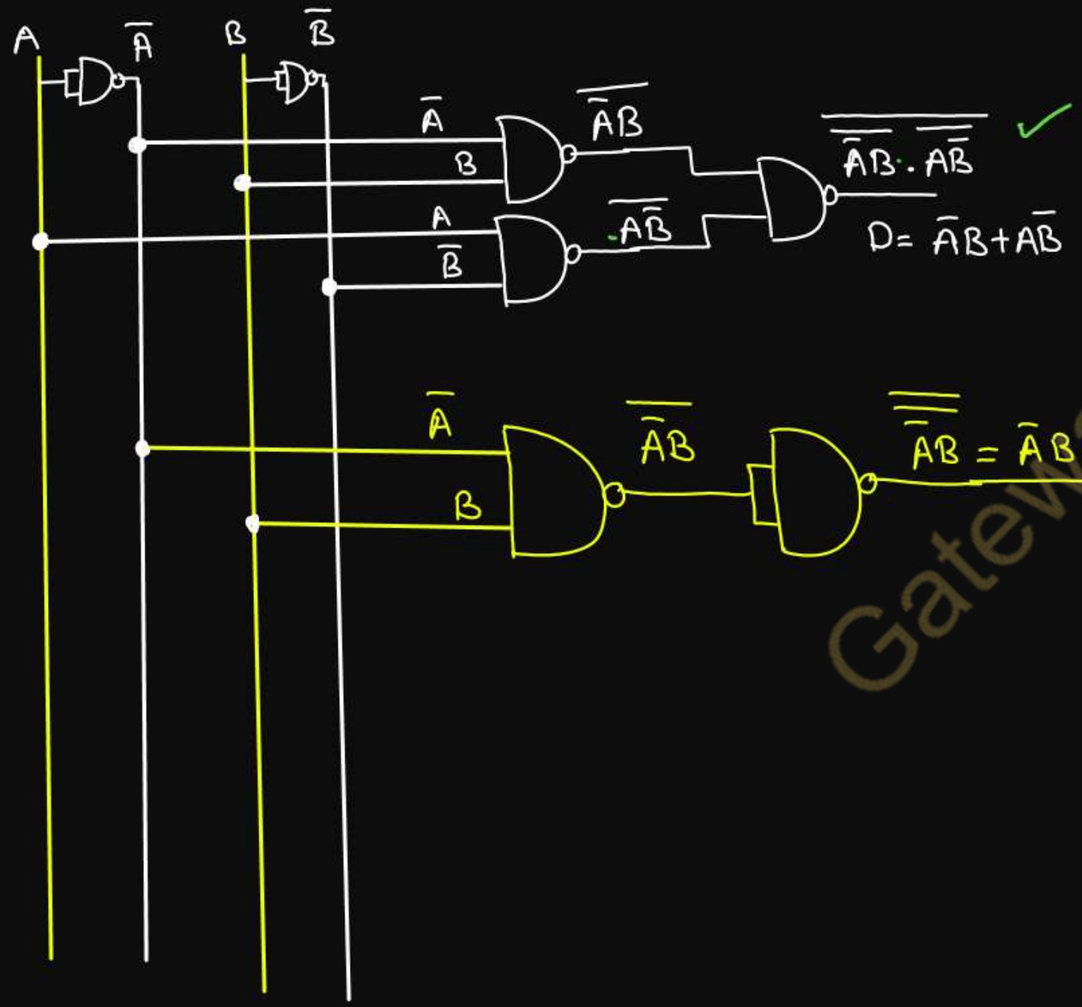
Using NAND Gate

$D = \overline{\bar{A}B + A\bar{B}}$

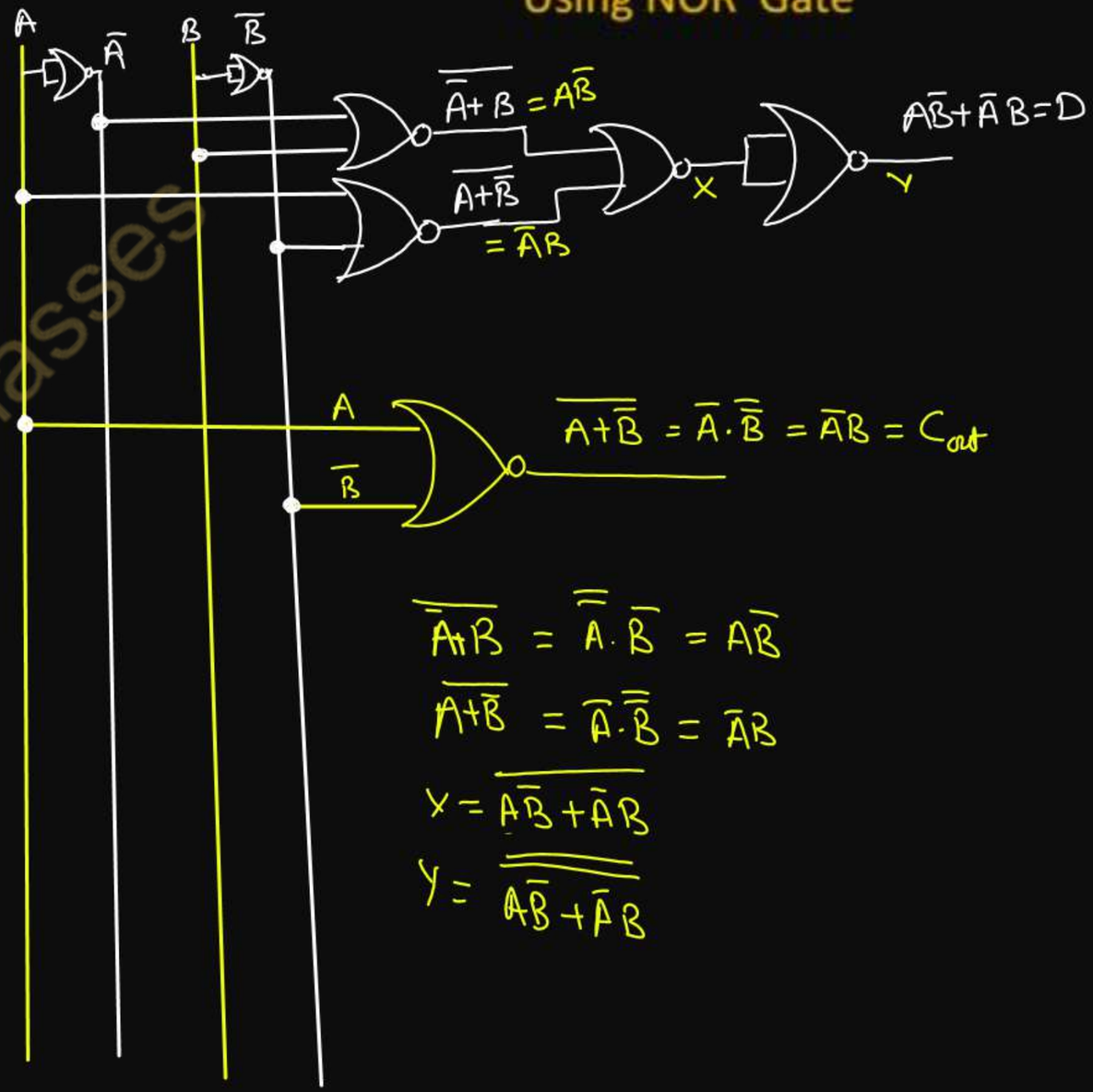
$= \overline{\bar{A}B} \cdot \overline{A\bar{B}}$ ✓

$B_o = \overline{\bar{A}B} = \overline{\bar{A} + B}$

$= \overline{A + \bar{B}}$



Using NOR Gate



$\overline{A+B} = \bar{A} \cdot \bar{B} = A\bar{B}$

$\overline{\bar{A}+\bar{B}} = \bar{A} \cdot \bar{B} = \bar{A}B$

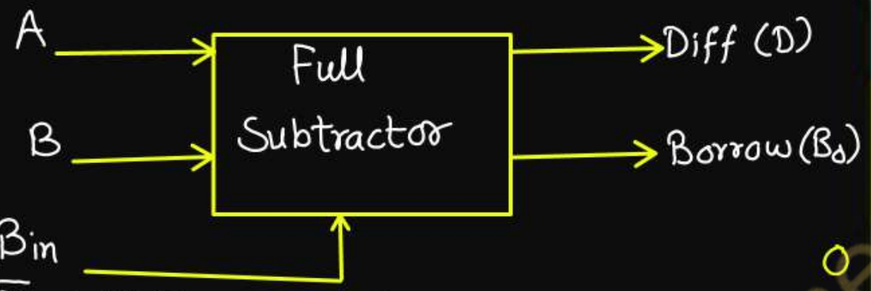
$X = \overline{A\bar{B} + \bar{A}B}$

$Y = \overline{\overline{A\bar{B} + \bar{A}B}}$

Gateway Classes

Full Subtractor

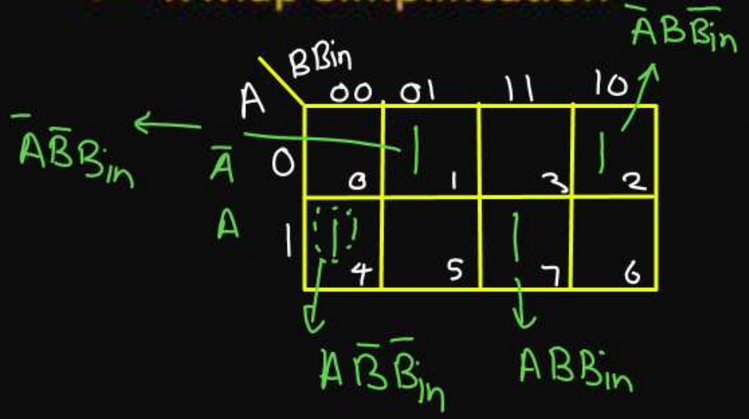
- A full subtractor subtracts three inputs: A (minuend), B (subtrahend), and a borrow-in (Bin) from the previous stage.



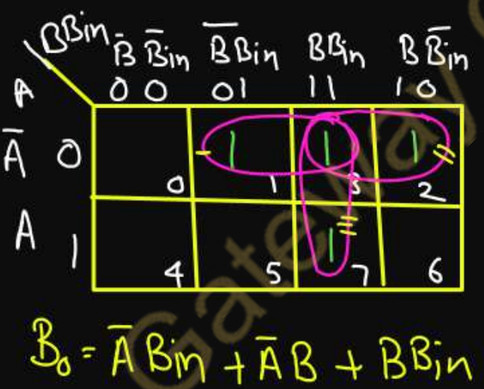
$$D = \bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + A\bar{B}\bar{B}_{in} + ABB_{in} = A \oplus B \oplus C_{in}$$

$$B_0 = \bar{A}B_{in} + \bar{A}B + BB_{in}$$

- K Map Simplification



Difference



$B_0 = \bar{A}B_{in} + \bar{A}B + BB_{in}$

Borrow

	INPUTS			OUTPUTS	
	A	B	B_{in}	Difference (D) $A-B-B_{in}$	Borrow (B_0)
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	0	1	1
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	0
6	1	1	0	0	0
7	1	1	1	1	1

Gateway Classes

$$D = \bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + A\bar{B}\bar{B}_{in} + AB B_{in}$$

$$D = B_{in}(\bar{A}\bar{B} + AB) + \bar{B}_{in}(\bar{A}B + A\bar{B})$$

$\bar{A}\bar{B} + AB$: X-NOR → $\bar{X}$

$\bar{A}B + A\bar{B}$: X-OR → X

$$D = B_{in}\bar{X} + \bar{B}_{in} \cdot X$$

$$D = X \oplus B_{in}$$

$$D = A \oplus B \oplus B_{in}$$

$$\overline{A \odot B} = A \oplus B$$

$$X = \bar{A}B + A\bar{B} = A \oplus B$$

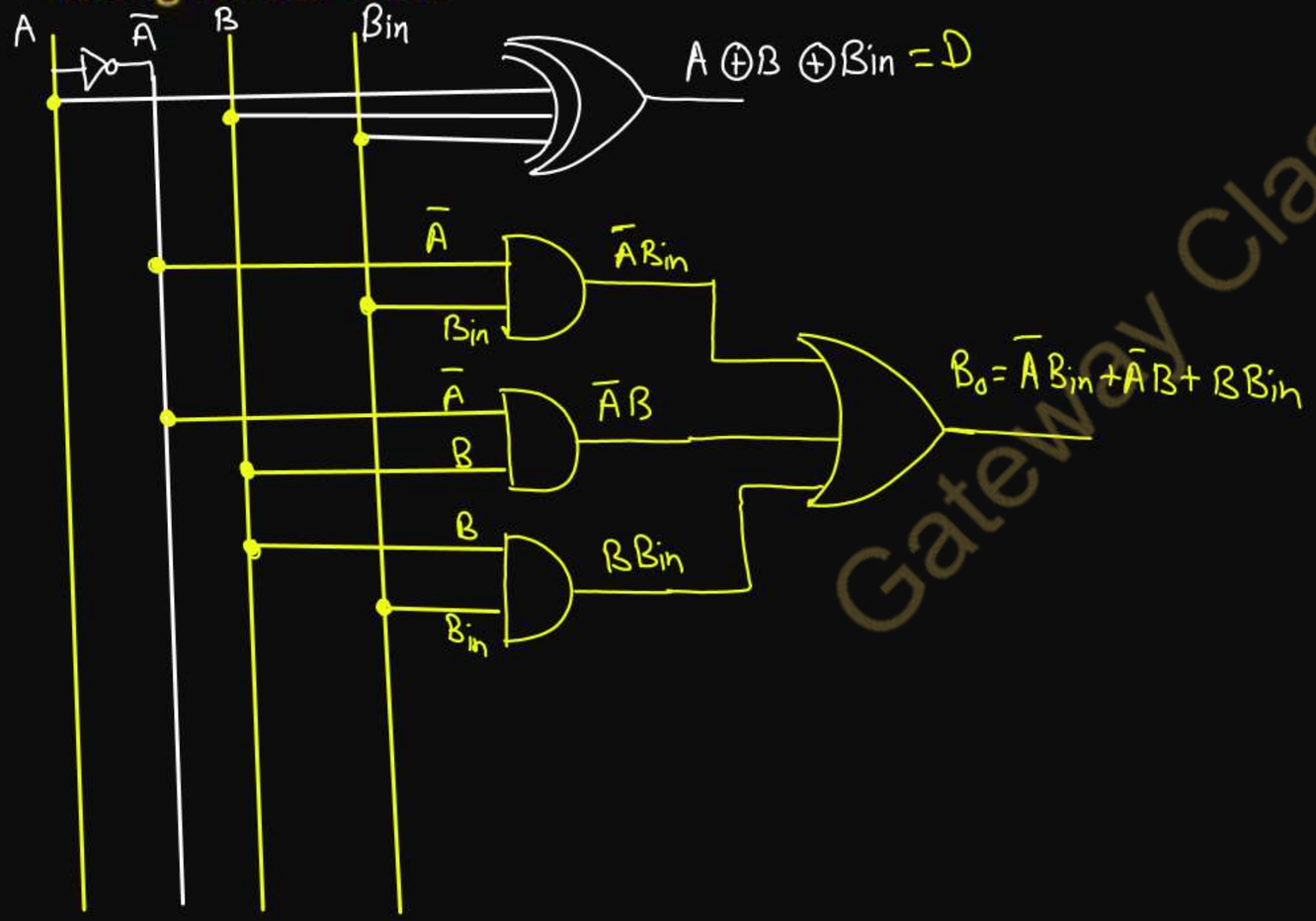
Gateway Classes

Logic Implementation

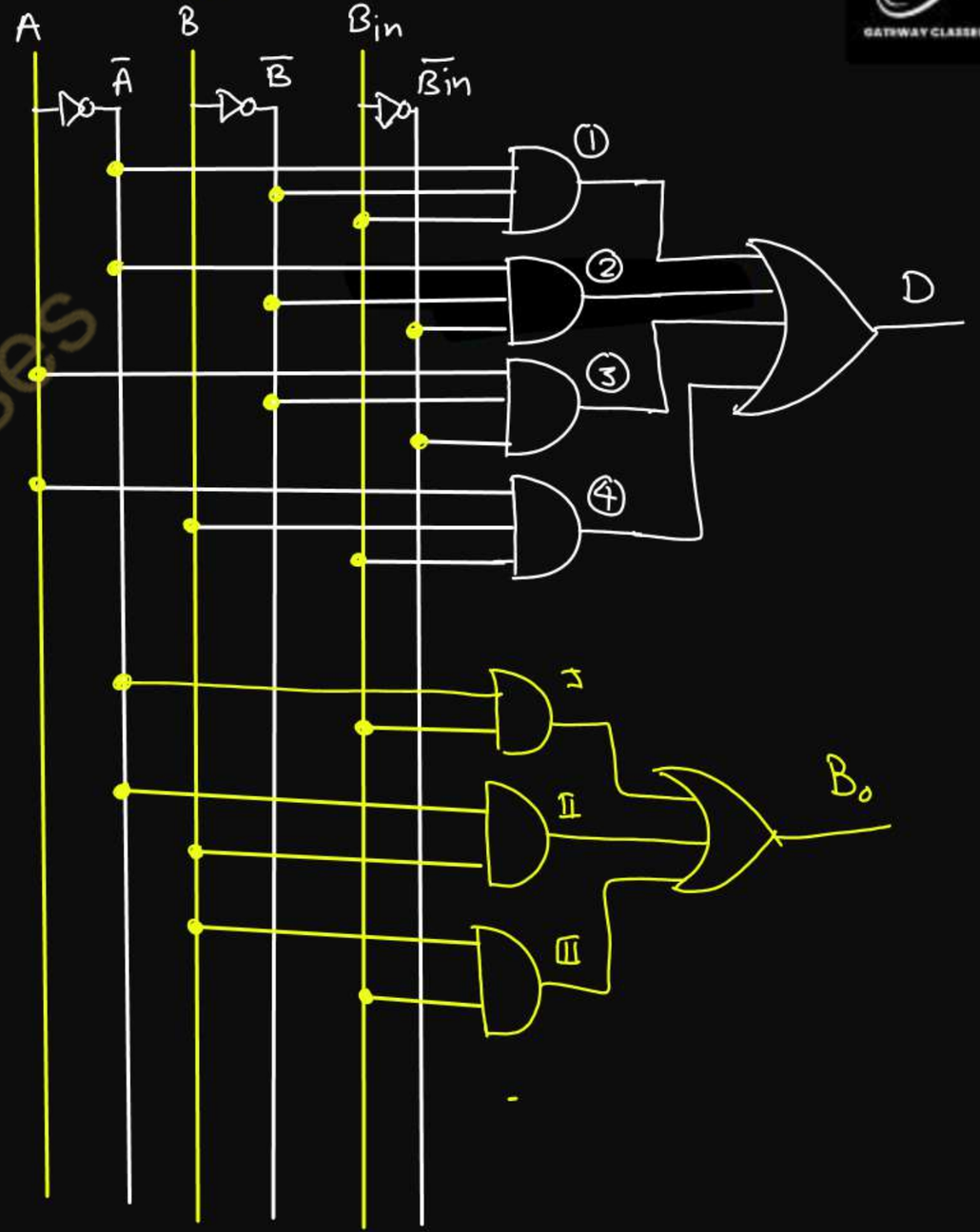
$$D = \underset{①}{\bar{A}\bar{B}B_{in}} + \underset{②}{\bar{A}B\bar{B}_{in}} + \underset{③}{A\bar{B}\bar{B}_{in}} + \underset{④}{AB B_{in}} = A \oplus B \oplus B_{in}$$

$$B_0 = \underset{I}{\bar{A}B_{in}} + \underset{II}{\bar{A}B} + \underset{III}{B B_{in}}$$

Using Ex-OR Gate



Using Basic Gates



Gateway Classes

Logic Implementation

AKTU-2021-22

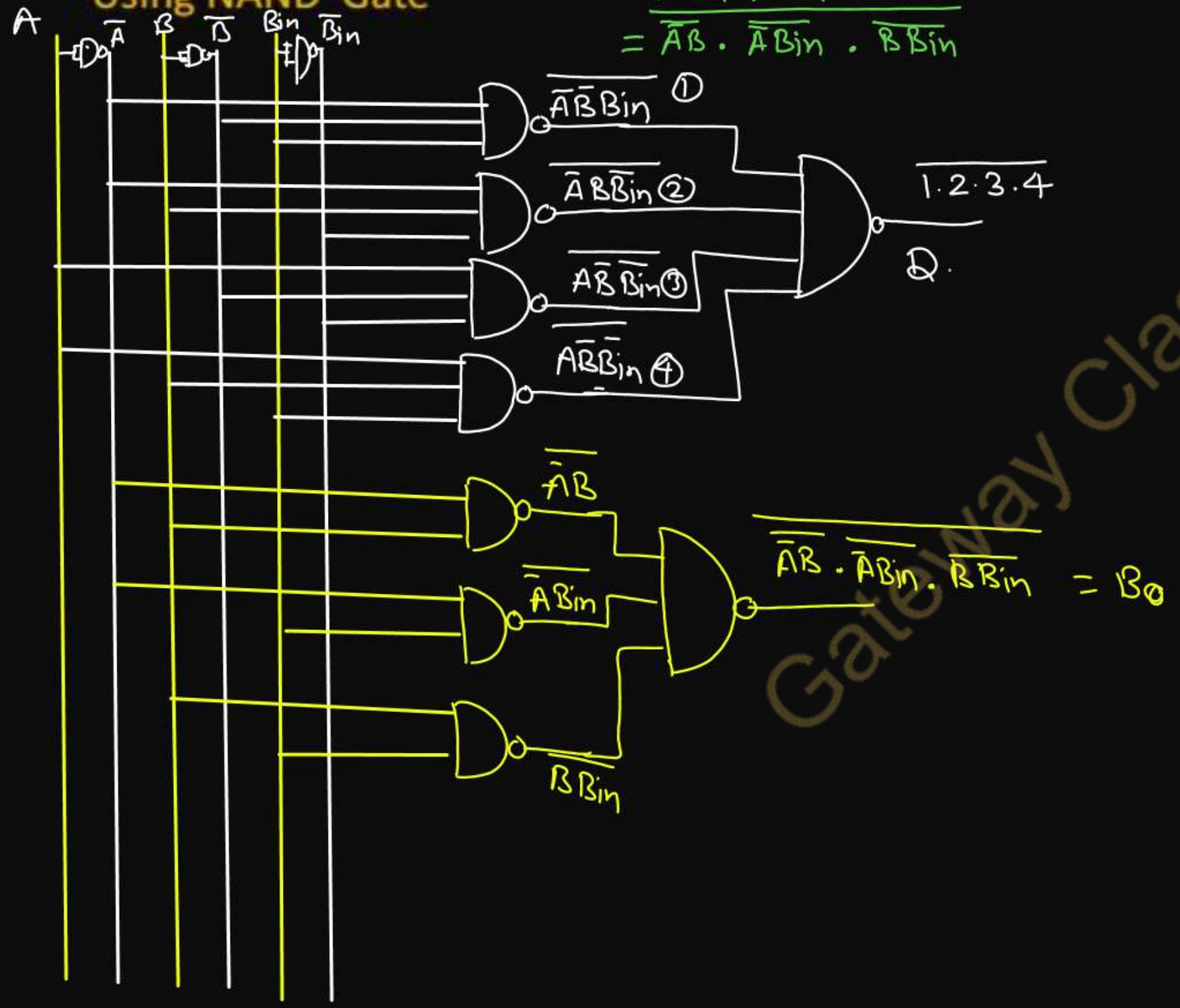
$$D = \overline{\bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + A\bar{B}\bar{B}_{in} + AB\bar{B}_{in}}$$

$$D = \overline{\bar{A}\bar{B}B_{in}} \cdot \overline{\bar{A}B\bar{B}_{in}} \cdot \overline{A\bar{B}\bar{B}_{in}} \cdot \overline{AB\bar{B}_{in}}$$

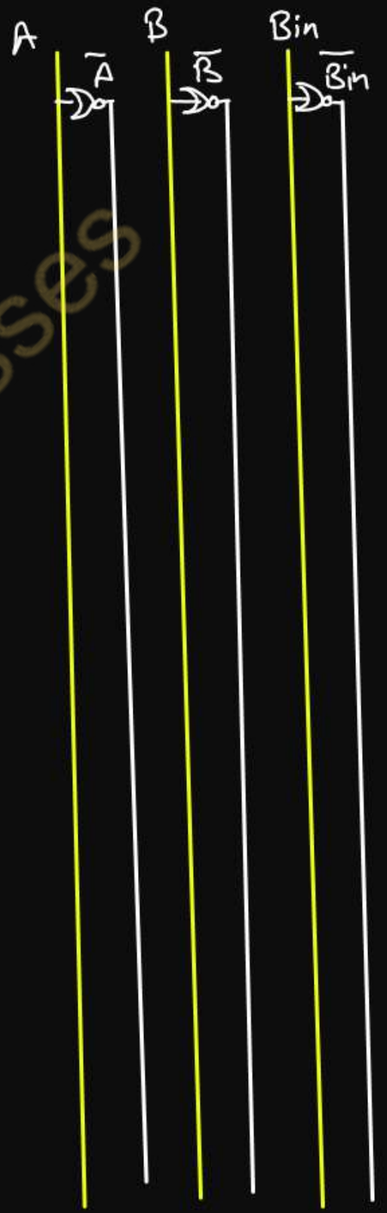
$$B_o = \overline{\bar{A}B + \bar{A}B_{in} + B\bar{B}_{in}}$$

$$= \overline{\bar{A}B} \cdot \overline{\bar{A}B_{in}} \cdot \overline{B\bar{B}_{in}}$$

Using NAND Gate



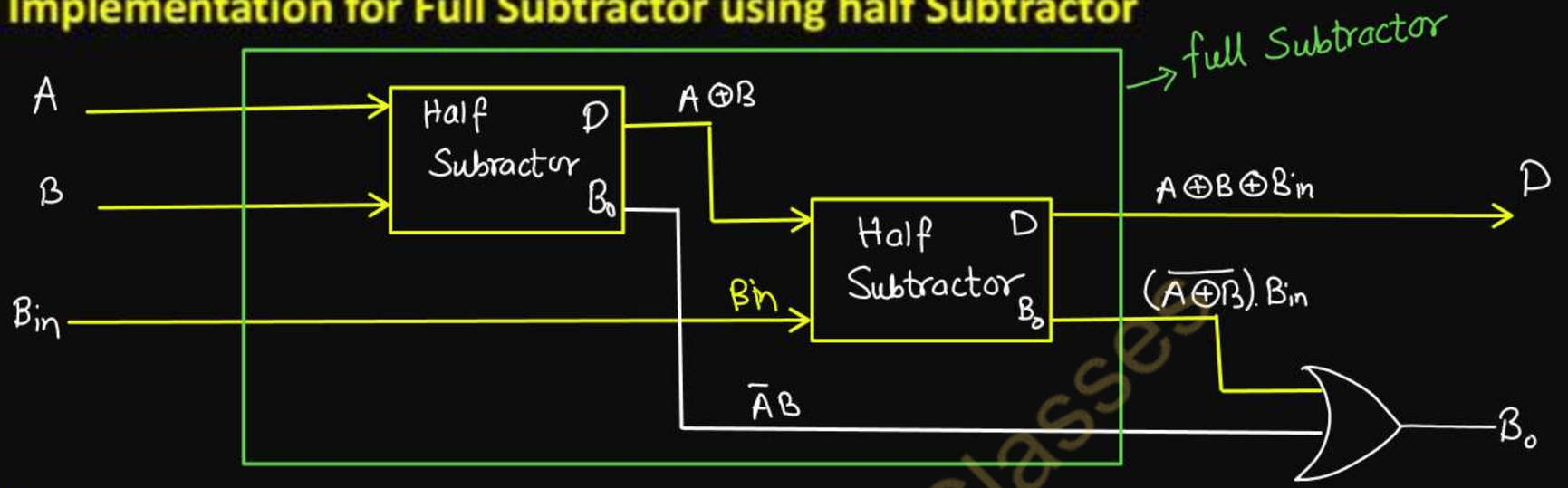
Using NOR Gate



"Do Yourself"

Gateway Classes

Logic Implementation for Full Subtractor using half Subtractor



For Borrow

$1+B_{in}=1$

$$= \left(\overline{A\oplus B}\right).B_{in} + \bar{A}B$$

$$= (A \odot B)B_{in} + \bar{A}B$$

$$= (\bar{A}\bar{B} + AB)B_{in} + \bar{A}B$$

$$= \bar{A}\bar{B}B_{in} + ABB_{in} + \bar{A}B(1+B_{in})$$

$$= \bar{A}\bar{B}B_{in} + ABB_{in} + \bar{A}B + \bar{A}BB_{in}$$

$$= BB_{in}(A+\bar{A}) + \bar{A}\bar{B}B_{in} + \bar{A}B \quad (A+\bar{A}=1)$$

$$= BB_{in} + \bar{A}\bar{B}B_{in} + \bar{A}B(1+B_{in})$$

$$= BB_{in} + \bar{A}\bar{B}B_{in} + \bar{A}B + \bar{A}BB_{in}$$

$$= BB_{in} + \bar{A}B_{in}(\bar{B}+B) + \bar{A}B \quad (\bar{B}+B=1)$$

$$\boxed{B_0 = BB_{in} + \bar{A}B_{in} + \bar{A}B}$$

Gateway Classes

AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Gateway Classes

Unit-2 Lec- 3

Today's target

- Serial and parallel adders
- BCD adder
- PYQ



Avinash Sir

UNIT-2 : Combinational circuits:

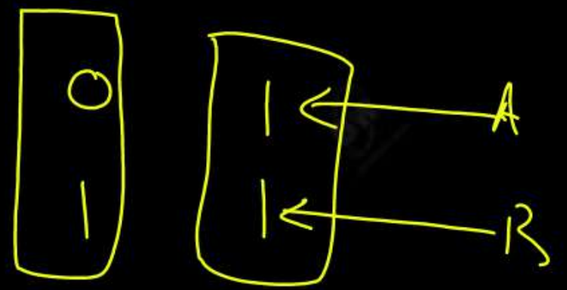
- Analysis and Design of combinational circuits
- MSI devices like comparators, multiplexers, demultiplexers, encoder, decoder,
- circuit realization using Multiplexers and decoders, half and full adders, subtractors, serial and parallel adders, BCD adder, barrel shifter and ALU.

Gateway Classes

Parallel Adder

A parallel adder is a digital circuit used to add two binary numbers simultaneously (in parallel). A parallel adder adds all the bits of two binary numbers at the same time using multiple full adders connected in parallel.

- Each bit of the numbers is added by a separate full adder.
- Carries generated from each stage are propagated to the next higher stage.
- Thus, it is much faster than a serial adder.

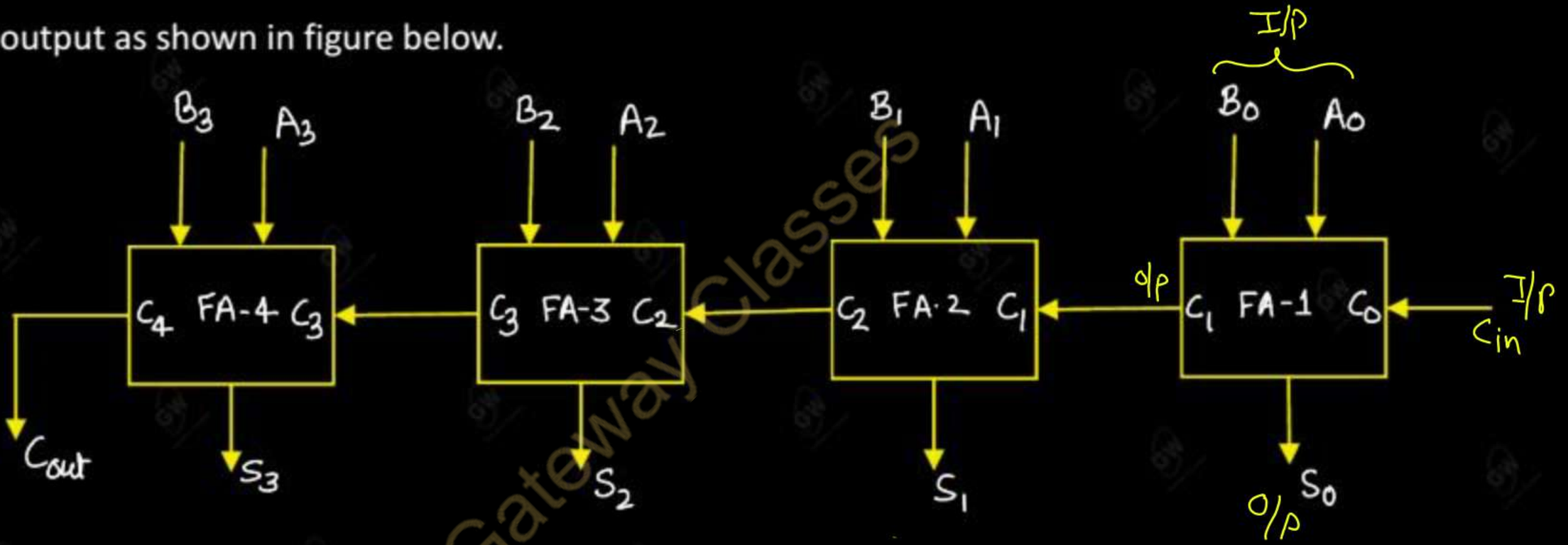


Gateway Classes

4-bit binary adder

Final o/p = C_{out} S_3 S_2 S_1 S_0

The 4-bit binary adder using full adder circuits is capable of adding two 4-bit numbers resulting in a 4-bit sum and a carry output as shown in figure below.



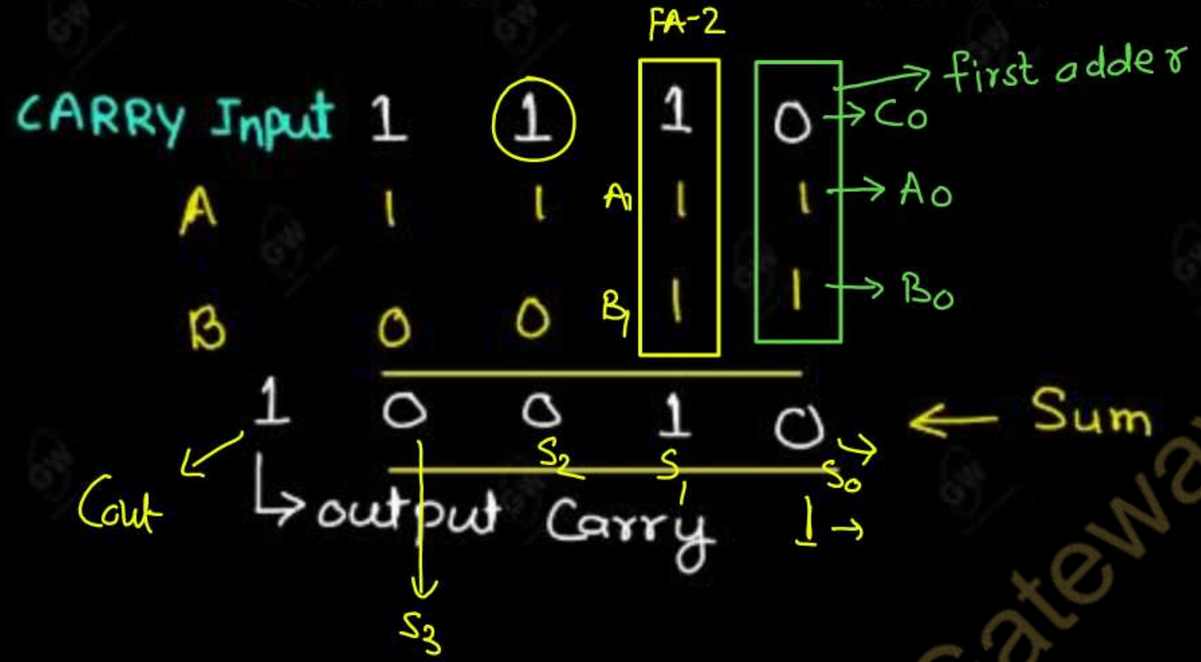
Gateway Classes

- Use 4 full adders in cascade.
- Carry-out of each stage is connected to carry-in of next stage.

4-bit binary adder

Let the 4-bit words to be added be represented by,

$A_3 A_2 A_1 A_0 = 1\ 1\ 1\ 1$ and $B_3 B_2 B_1 B_0 = 0\ 0\ 1\ 1$.



Gateway Classes

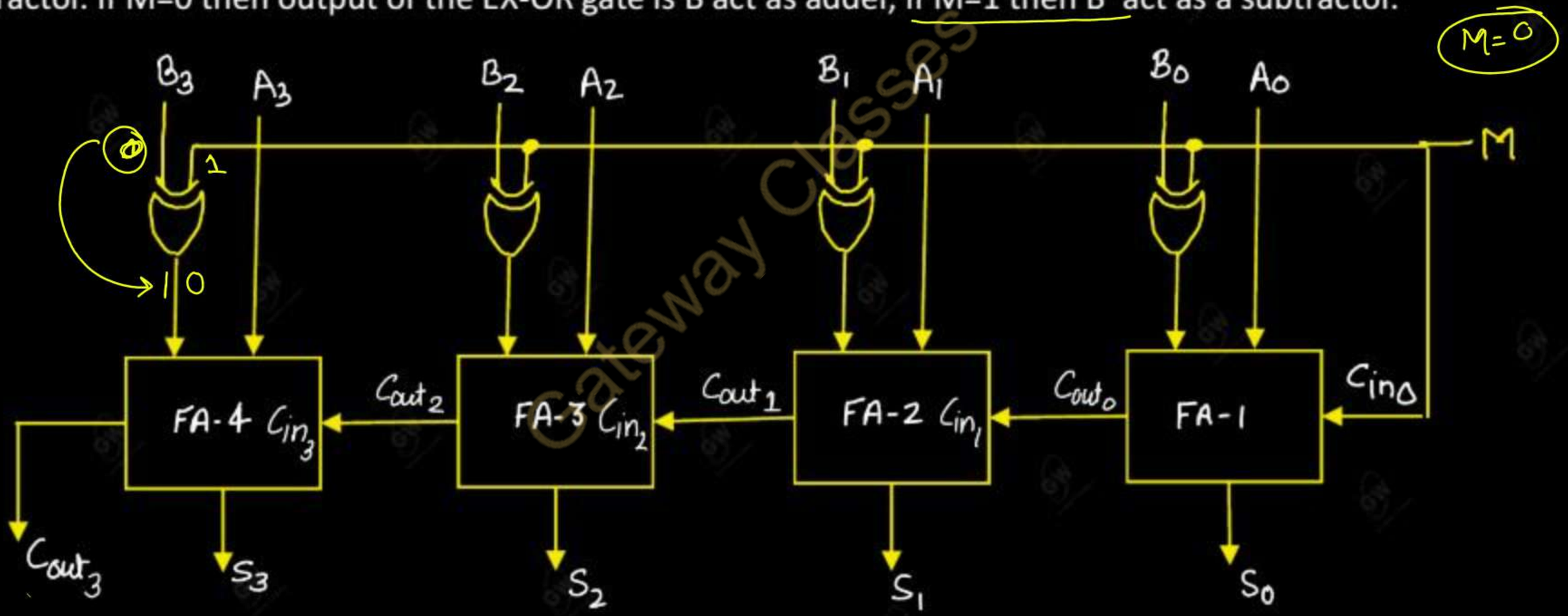
Advantages

- Very fast compared to serial adders (all bits processed simultaneously).
- Simple design using only full adders.
- Widely used in CPUs and ALUs.

Parallel binary Adder – Subtractor

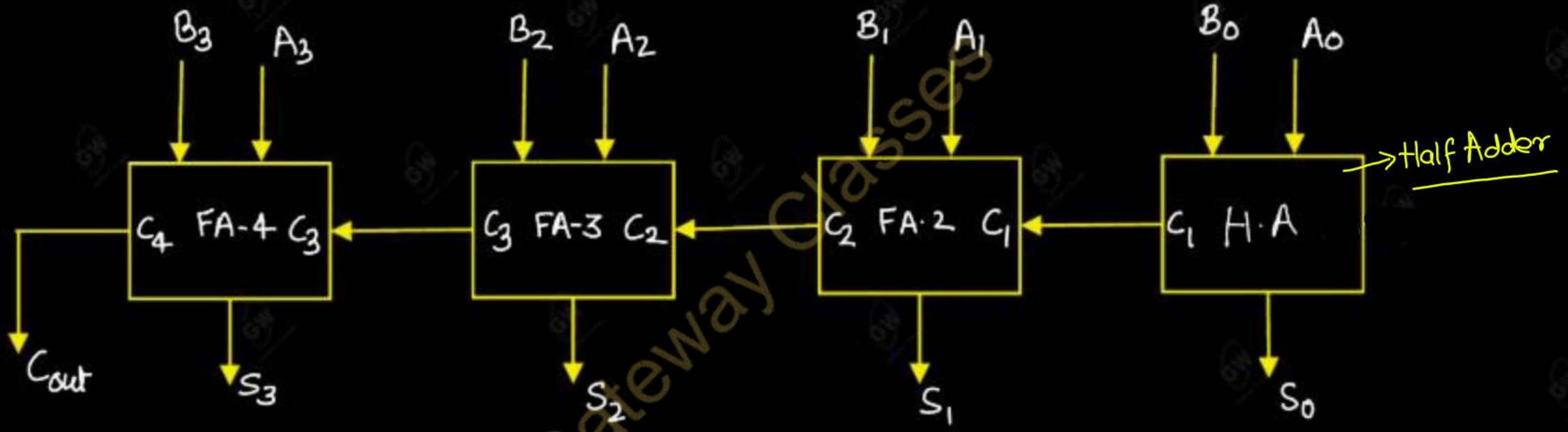


The addition and subtraction operations can be perform using a common adder circuit, where a EX-OR gate is connected in the second input along with the mode selection bit M. if M=0 the circuit act as a adder, M=1 then subtractor. If M=0 then output of the EX-OR gate is B act as adder, if M=1 then B' act as a subtractor.



Gateway Classes

Ex. Design a 4-bit adder using three full adder and one half adder.



Gateway Classes

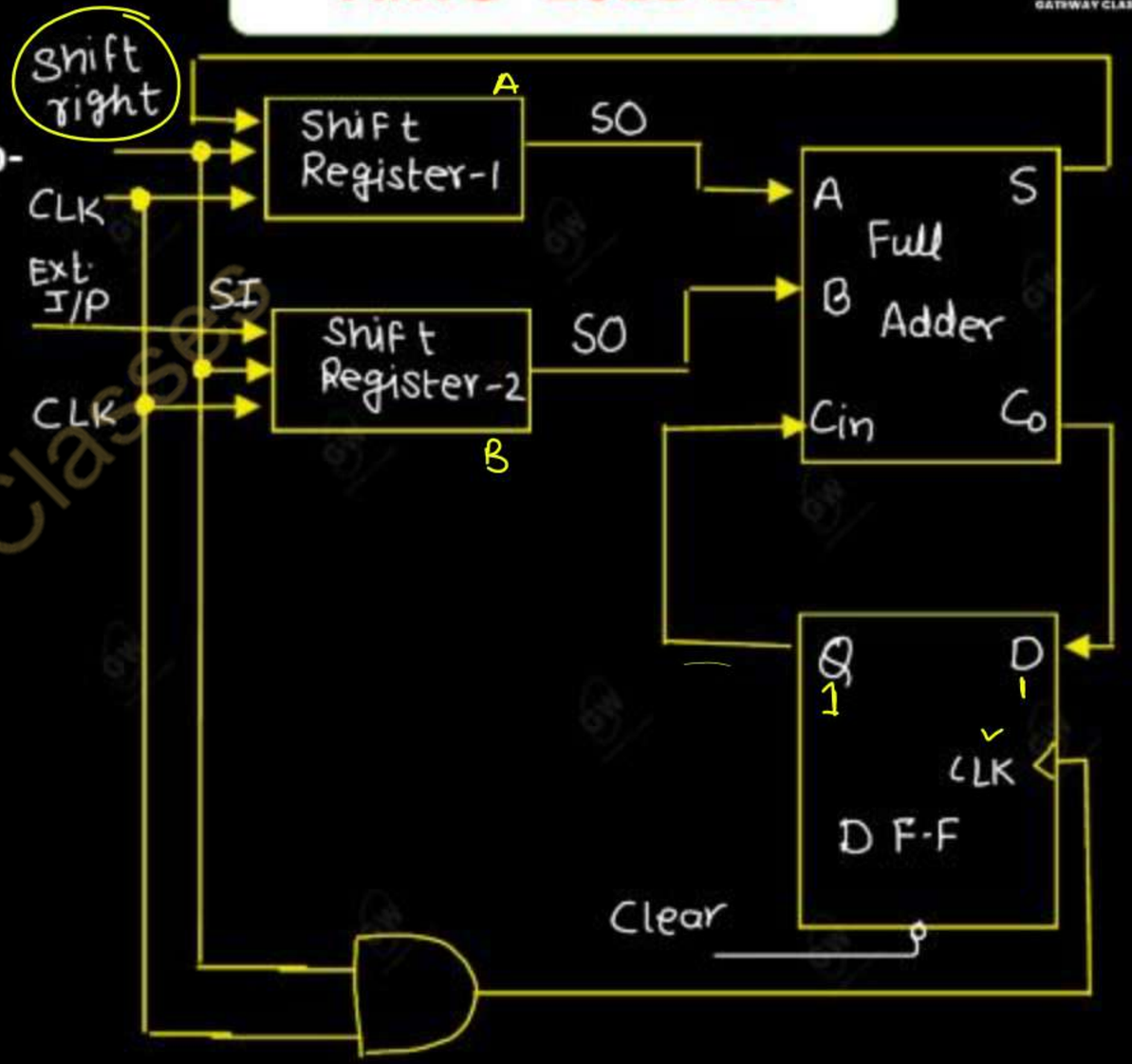
Addition Perform →

Serial Adder

AKTU- 2021-22

1. Definition and Circuit

- Uses two shift registers, one full adder, one D flip-flop, and an AND gate.
- The serial outputs of the registers feed into the full adder.
- The sum goes back into one register, and the carry is stored in the D flip-flop.
- Clock pulses (through the AND gate) control shifting and addition.



Gateway Classes

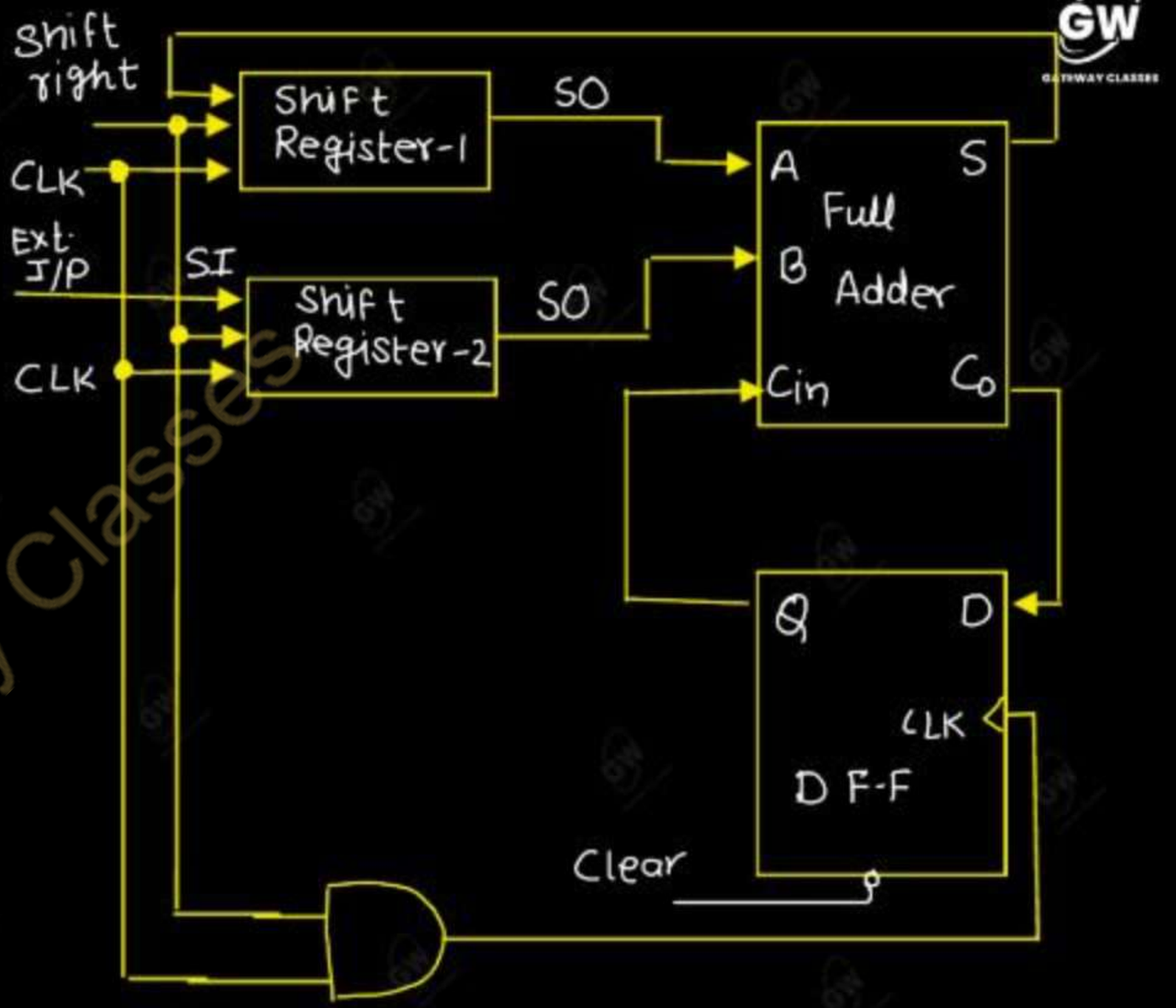
Serial Adder

2. Working Operation

Initially, the D flip-flop is cleared so Q= 0 and C_{in} = 0.

At each clock pulse:

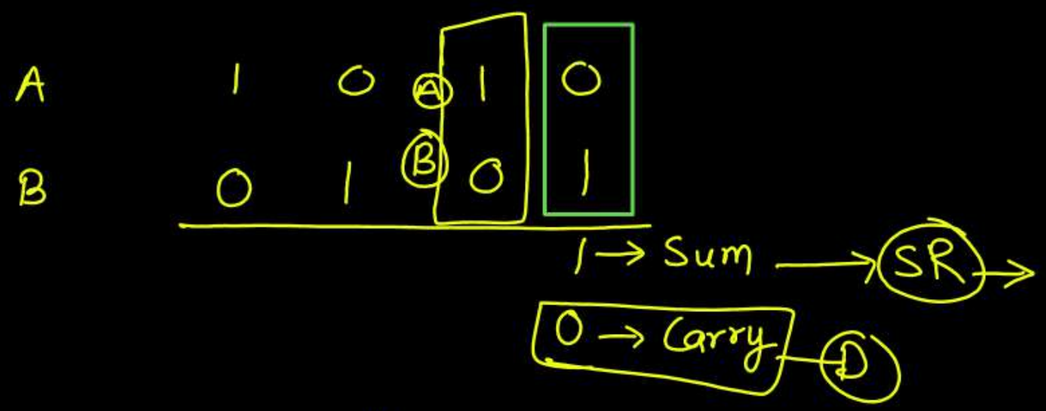
- The least significant bits (LSBs) from both registers are added.
- The full adder produces sum (S) and carry (Cout). Carry is stored in the D flip-flop for the next stage.
- Both registers shift right → next bits move to the output.
- This repeats until all bits are added.

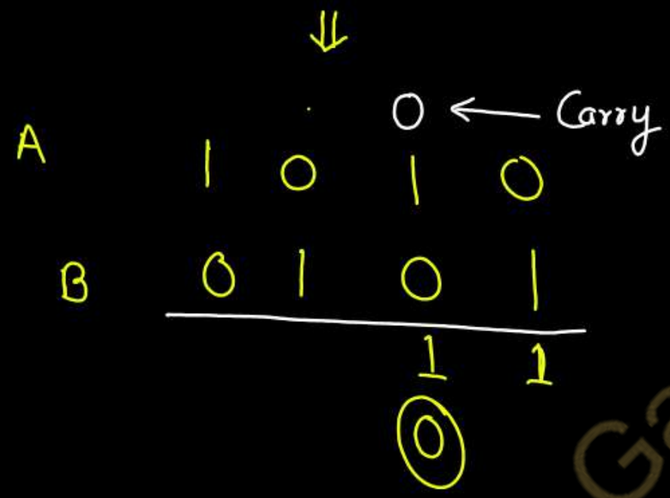


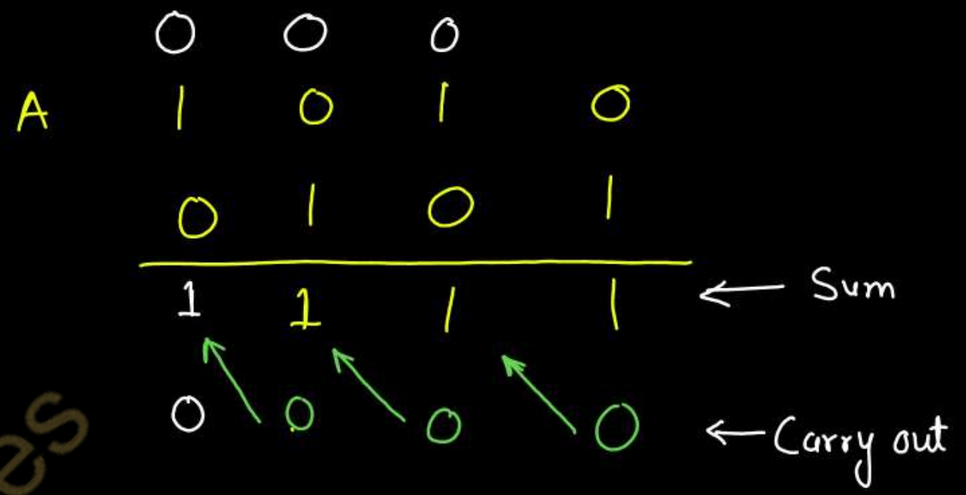
Gateway Classes

Example:-

Let







Gateway Classes

Serial Adder

3. Advantages

- Adds one pair of bits at a time (step-by-step).Needs only $n-1$ clock cycles to finish addition.
- Addition continues as long as shift control is enabled.
- Output sum is obtained in serial form.

4. Drawbacks

- Slow – takes more time compared to parallel adder.
- Circuit is more complex.
- Requires more components.
- Output is in serial form, so complete result is not available at once.

Gateway Classes

BCD Adder

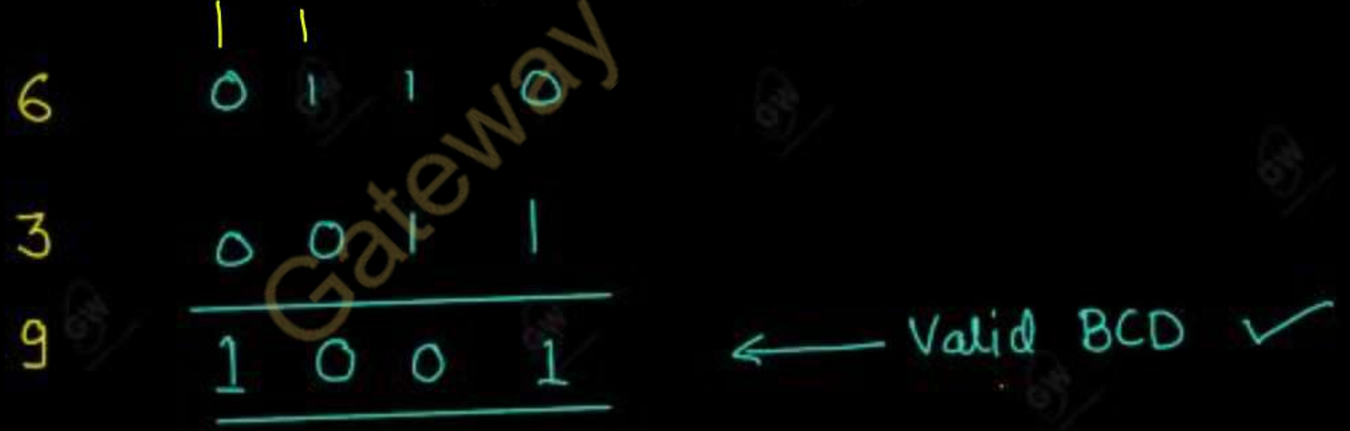


- In digital system, the decimal number is represented in the form of binary coded decimal (BCD).
- The ten digit (0-9) decimal numbers are represented by the binary digits.
- The circuit which add the two BCD number is called BCD adder. The BCD cannot be greater than 9.

There are three different cases in BCD Addition

(i) Sum is less than or equal to 9 with carry 0

Consider the addition of two BCD numbers 6 and 3, The addition is performed as normal binary addition



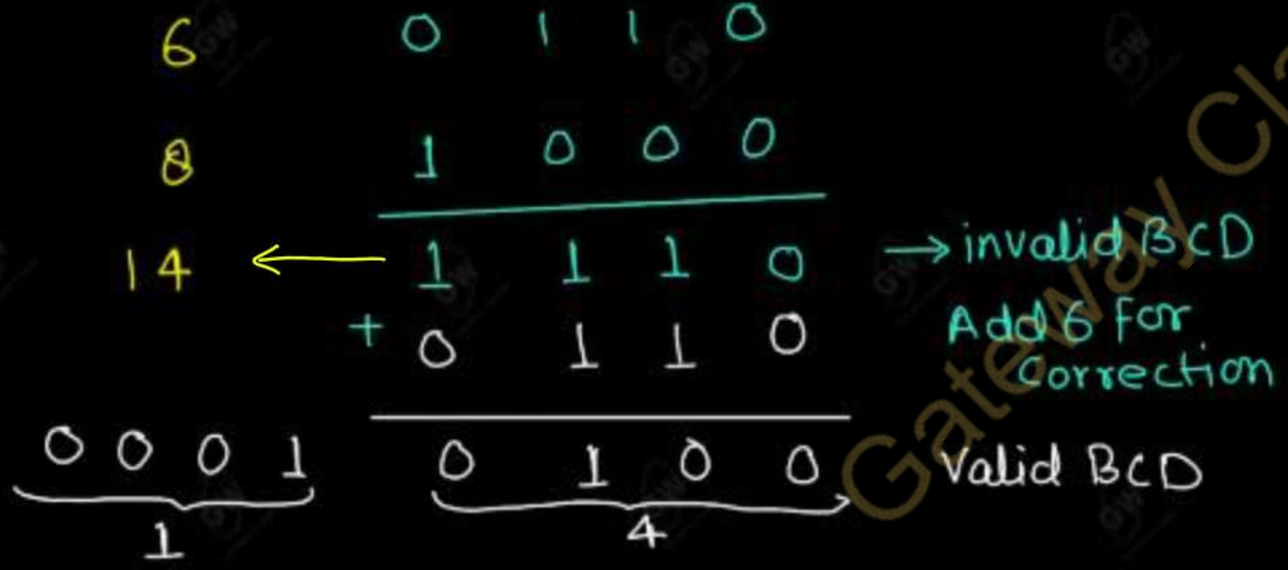
Gateway Classes

BCD Adder

1 4
0001 0100

ii) Sum is greater than 9 with carry 0

consider the number 6 and 8 in BCD. The sum is invalid BCD number, Add the sum with correction number 6



After addition of 6 carry is produced into the second decimal position.

iii) Sum equals 9 or less with carry 1

Consider the addition of 8 and 9 in BCD. The result 0001 0001 is valid BCD number but it is incorrect. Add 6 to get correct number.



Gateway Classes



The procedure for BCD addition is

1. Add two BCD numbers using ordinary binary addition.

2. If four bit sum is less than or equal to ~~zero~~ 9, then correction is not required.

3. If the four bit sum is greater than 9 or if carry is generated then add 0110.

Implementation of BCD Adder

We require 4-bit binary adder for initial addition, Logic circuit to detect sum greater than 9, and second 4 bit binary adder to add 0110.

The following truth table is used to design a circuit for the sum, which is greater than 9

Why 6?

- Because 10 (decimal) = 1010_2, and we want to "roll over" into the next decimal digit.
- Adding 6 shifts invalid results (1010 to 1111) into the next valid BCD range.
- This also generates a carry for the next decimal place.

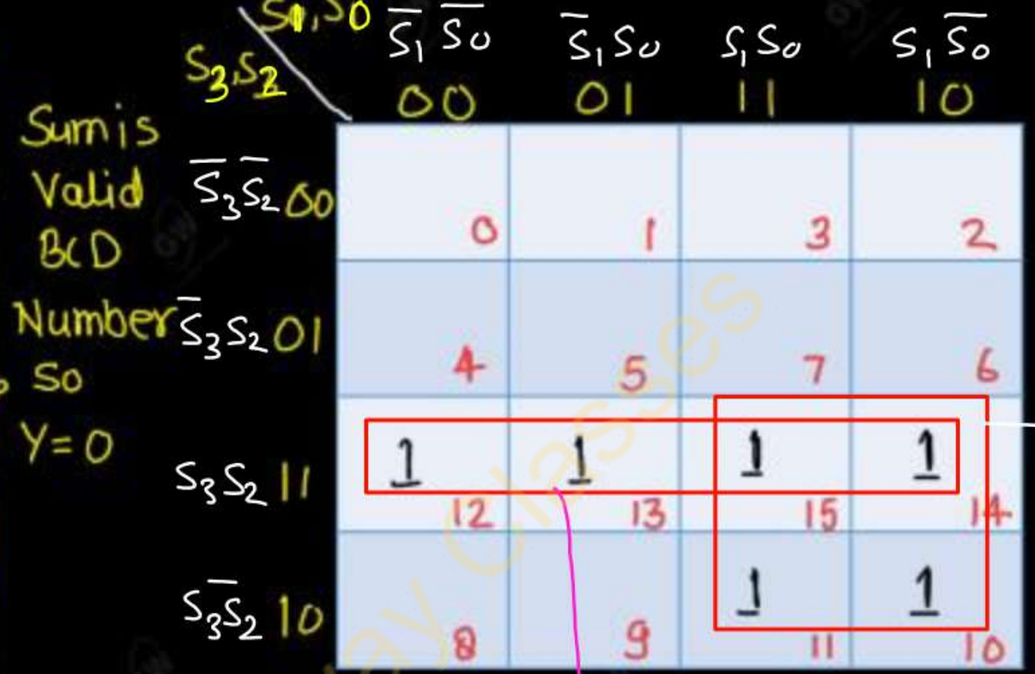
Gateway Classes

	Input				output
	S3	S2	S1	S0	Y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

Sum is Valid BCD Number so Y=0

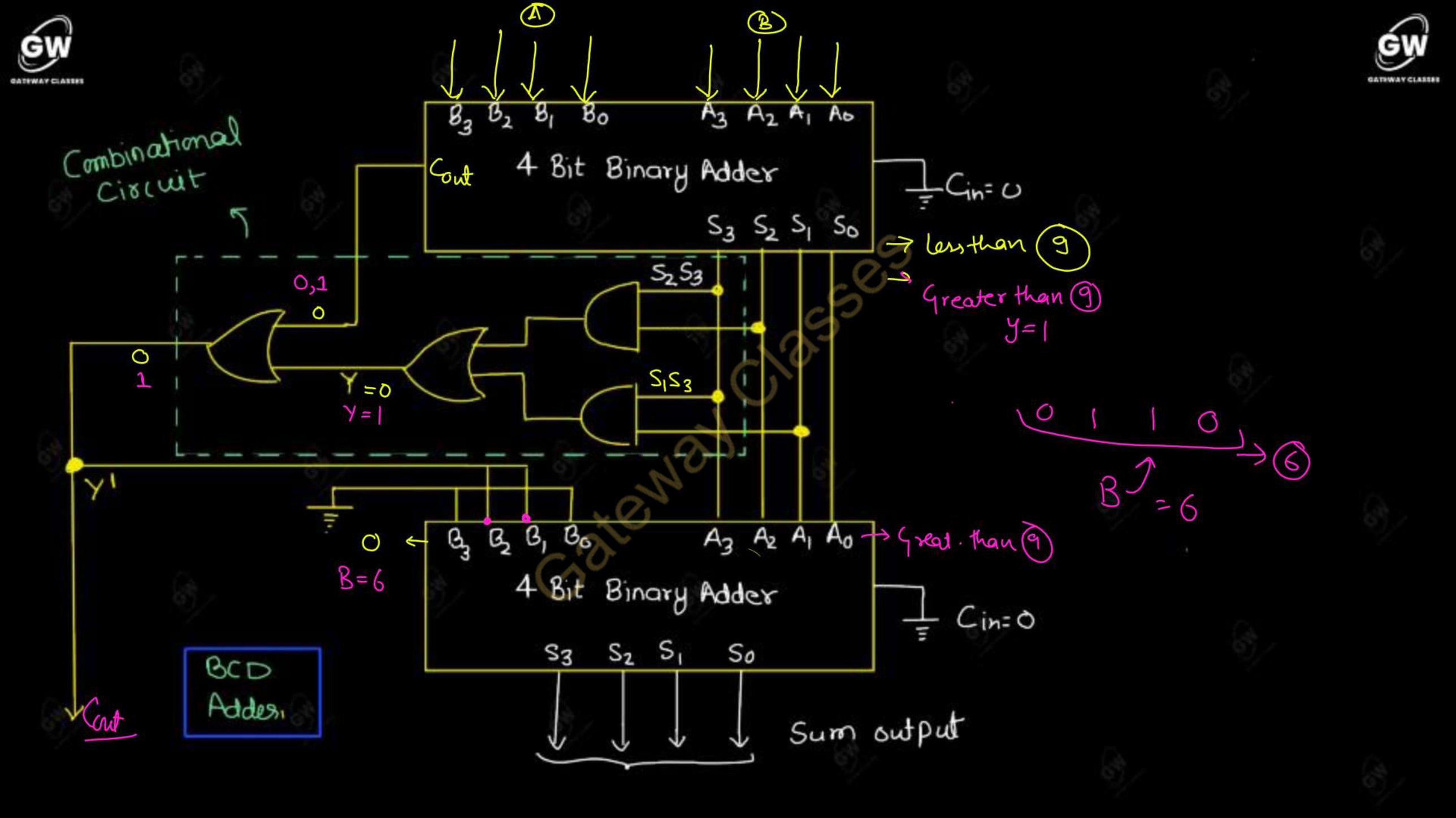
Sum is invalid BCD number Therefor Y=1

K-Map



$$Y = S_3S_2 + S_3S_1$$

Gateway Classes


A
B
B_3 B_2 B_1 B_0
A_3 A_2 A_1 A_0
4 Bit Binary Adder
Cout
C_{in} = 0
S_3 S_2 S_1 S_0
Combinational Circuit
0,1
0
0
1
Y = 0
Y = 1
S_2S_3
S_1S_3
Y1
less than 9
Greater than 9
Y = 1
0 1 1 0
6
B = 6
0
B = 6
Great. than 9
4 Bit Binary Adder
C_{in} = 0
S_3 S_2 S_1 S_0
Sum output
Cout
BCD Adder

AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-2 Lec-4



Today's target

- Comparators
- Decoders
- PYQ

Gateway Classes

Avinash Sir

UNIT-2 : Combinational circuits:

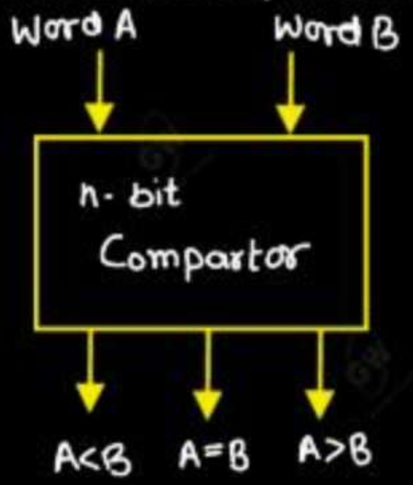
- Analysis and Design of combinational circuits
- MSI devices like comparators, multiplexers, demultiplexers, encoder, decoder,
- circuit realization using Multiplexers and decoders, half and full adders, subtractors, serial and parallel adders, BCD adder, barrel shifter and ALU.

Gateway Classes

Comparators

A comparator is a combinational circuit that compares two binary numbers (A and B) and determines their relative magnitudes, i.e., whether:

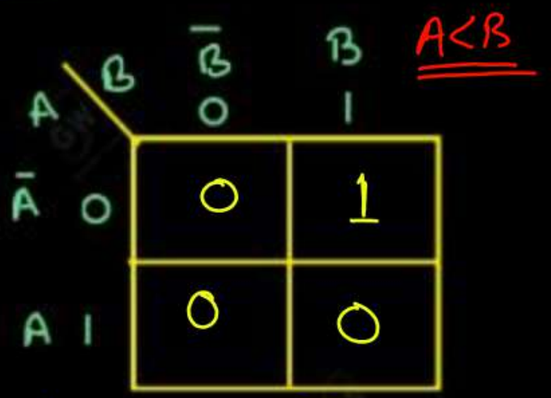
$$A = B, \quad A > B, \quad A < B$$

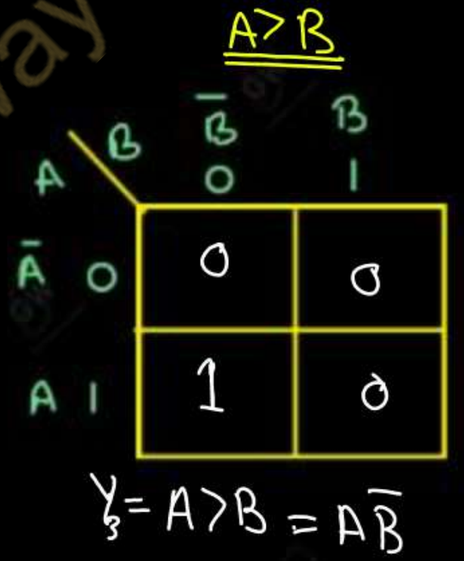


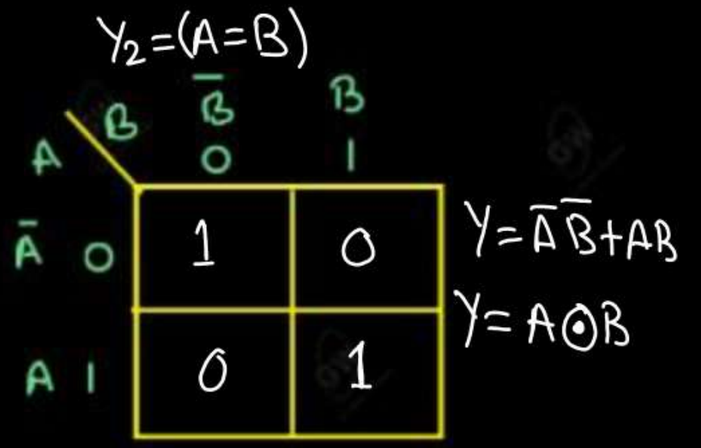
Basic 1-bit Comparator

For two 1-bit inputs A and B: Truth Table

INPUT		OUTPUT		
A	B	Y1 = A < B	Y2 = A = B	Y3 = A > B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0







Gateway Classes

Basic 1-bit Comparator

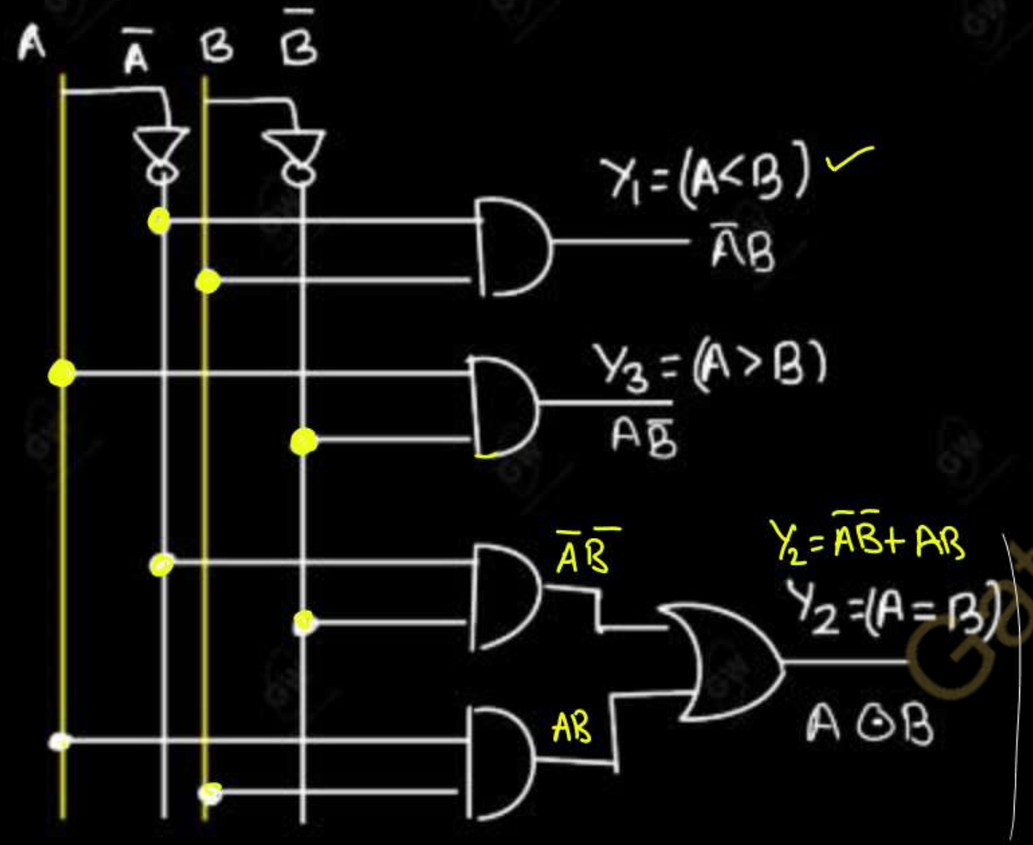
$Y_1 = (A < B) = \bar{A}B$

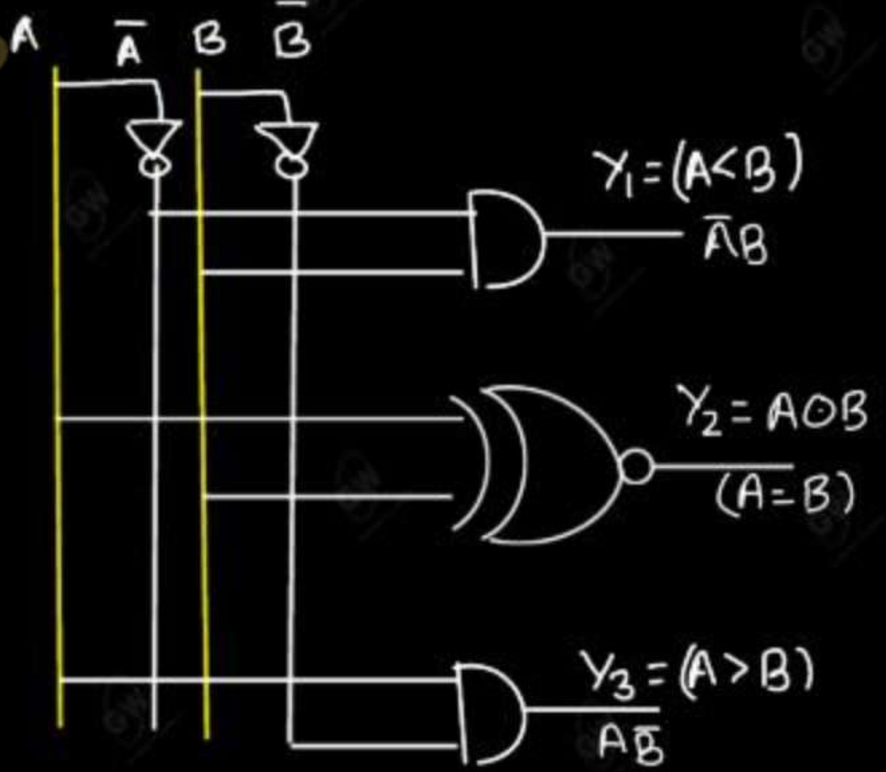
$Y_2 = (A = B) = A \odot B = \bar{A}\bar{B} + AB$

$Y_3 = (A > B) = A\bar{B}$

Realize using Ex-NOR Gate

Circuit Realization



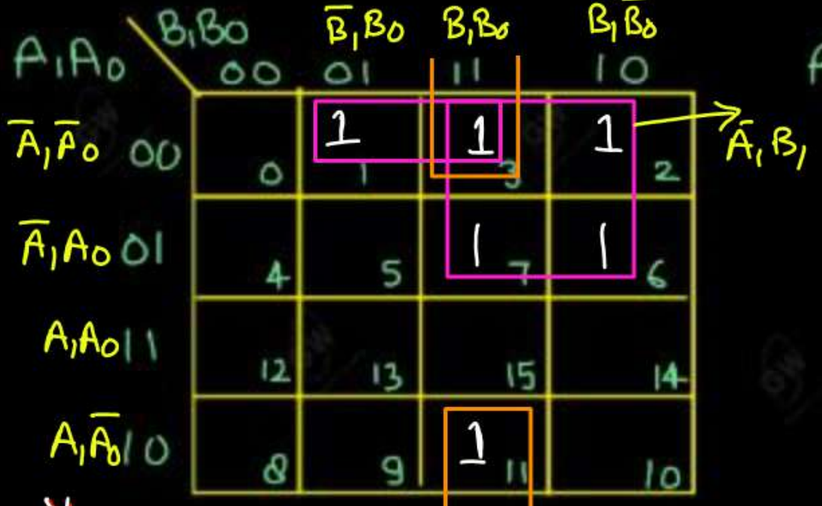


Gateway Classes

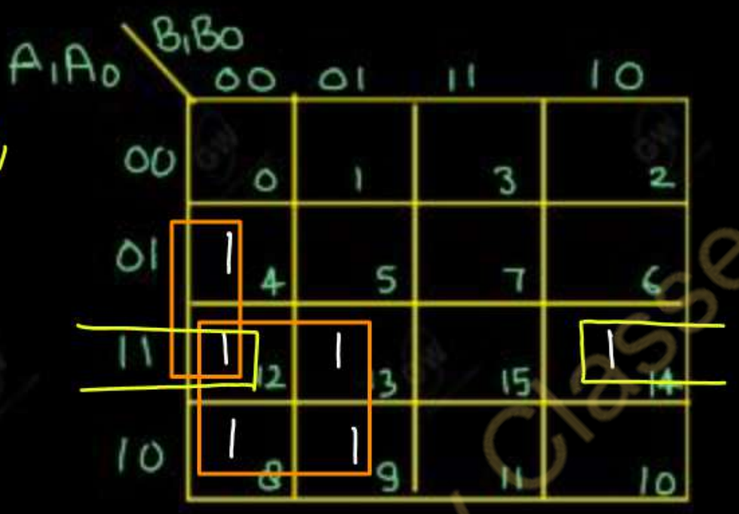
AKTU- 2024-25,23-24,21-22

Basic 2-bit Comparator

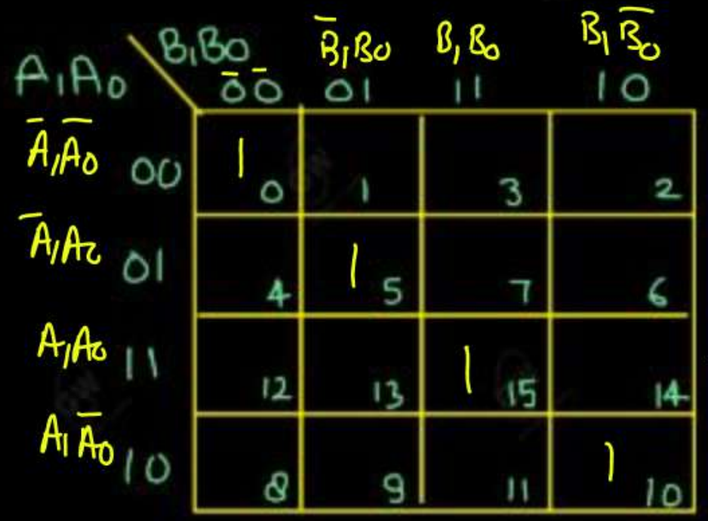
Let $A = A1\ A0$ and $B = B1\ B0$



$Y_1 = A<B = \bar{A}_1B_1 + \bar{A}_1\bar{A}_0B_0 + \bar{A}_0B_1B_0$



$Y_3 = A>B = A_1\bar{B}_1 + A_0\bar{B}_1\bar{B}_0 + A_1A_0\bar{B}_0$



$Y_2 = (A=B) = \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1B_0 + A_1A_0B_1B_0 + A_1\bar{A}_0B_1\bar{B}_0$

$= \bar{A}_1\bar{B}_1(\bar{A}_0\bar{B}_0 + A_0B_0) + A_1B_1(A_0B_0 + \bar{A}_0\bar{B}_0)$

$= (A_1B_1 + \bar{A}_1\bar{B}_1)(A_0B_0 + \bar{A}_0\bar{B}_0)$

$Y_2 = (A_1 \odot B_1)(A_0 \odot B_0)$

Truth table

	Input A1	Input A0	Input B1	Input B0	output A < B	output A = B	output A > B
0	0	0	0	0	0	1	0
1	0	0	0	1	1	0	0
2	0	0	1	0	1	0	0
3	0	0	1	1	1	0	0
4	0	1	0	0	0	0	1
5	0	1	0	1	0	1	0
6	0	1	1	0	1	0	0
7	0	1	1	1	1	0	0
8	1	0	0	0	0	0	1
9	1	0	0	1	0	0	1
10	1	0	1	0	0	1	0
11	1	0	1	1	1	0	0
12	1	1	0	0	0	0	1
13	1	1	0	1	0	0	1
14	1	1	1	0	0	0	1
15	1	1	1	1	0	1	0

Gateway Classes

Basic 2-bit Comparator

Circuit Realization

For $A<B$

$$\bar{A}_1\bar{A}_0B_0 + \bar{A}_1B_1 + \bar{A}_0B_1B_0 = Y_1$$

For $A>B$

$$A_0\bar{B}_1\bar{B}_0 + A_1A_0\bar{B}_0 + A_1\bar{B}_1 = Y_3$$

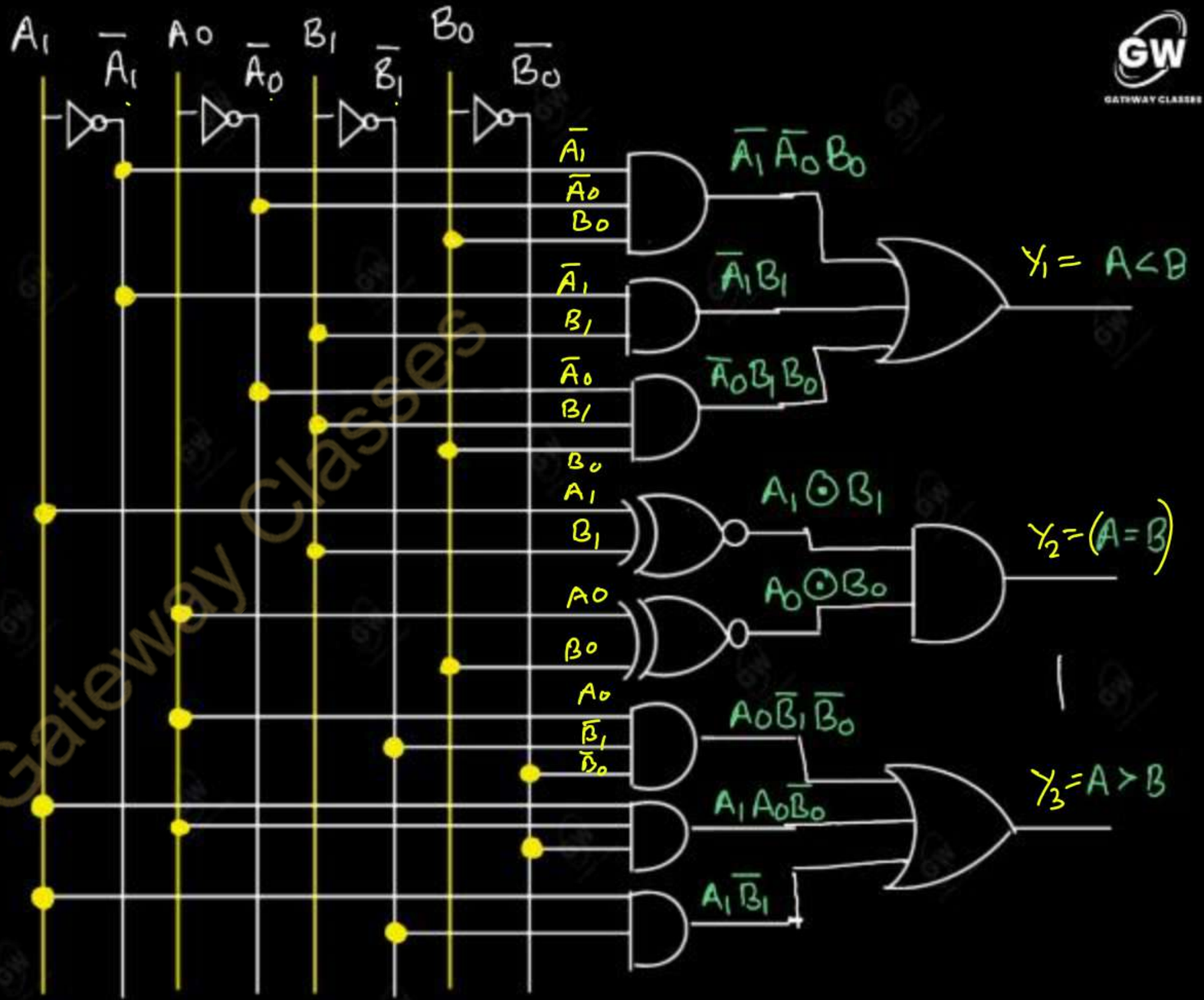
For $A=B$

$$= \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1B_0 + A_1A_0B_1B_0 + A_1\bar{A}_0B_1\bar{B}_0$$

$$= \bar{A}_0\bar{B}_0(\bar{A}_1\bar{B}_1 + A_1B_1) + A_0B_0(\bar{A}_1\bar{B}_1 + A_1B_1)$$

$$= (\bar{A}_1\bar{B}_1 + A_1B_1)(\bar{A}_0\bar{B}_0 + A_0B_0)$$

$$= (A_1 \odot B_1).(A_0 \odot B_0) = Y_2$$



Gateway Classes

n-bit Comparator

For larger numbers (like 4-bit or 8-bit), the same principle applies:

- Compare MSB first, then proceed bit by bit if equal.
- Usually implemented using cascaded 1-bit comparators.

IC 7485 is a common 4-bit magnitude comparator that provides outputs for $A>B$, $A=B$, and $A<B$.

Applications of Comparators

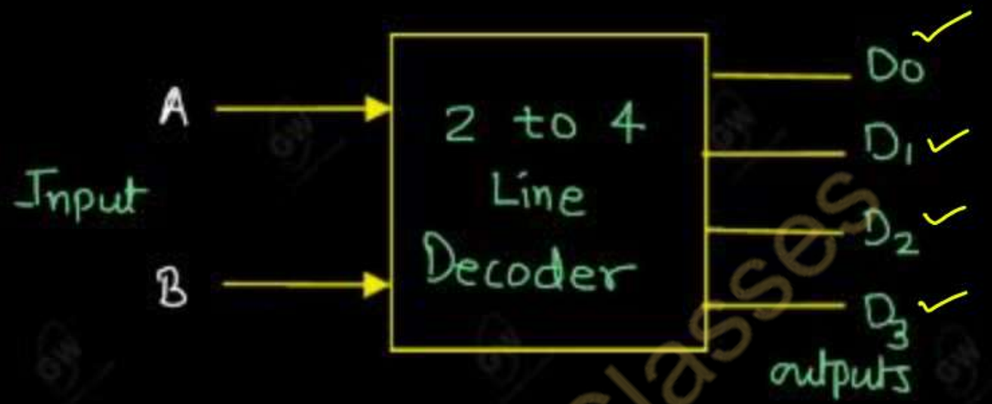
- Digital sorting (ordering numbers).
- Searching in digital systems.
- Used in ALUs to check conditions (equal, less, greater).
- Input matching in memory systems.

Gateway Classes

Decoder

A decoder is a combinational circuit that converts binary information from 'n' input lines into a maximum of 2^n unique output lines.

A 2 to 4 Line Decoder



$D_0 = \bar{A}\bar{B}$

$D_1 = \bar{A}B$

$D_2 = A\bar{B}$

$D_3 = AB$

INPUT		OUTPUT			
A	B	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Gateway Classes

Decoder

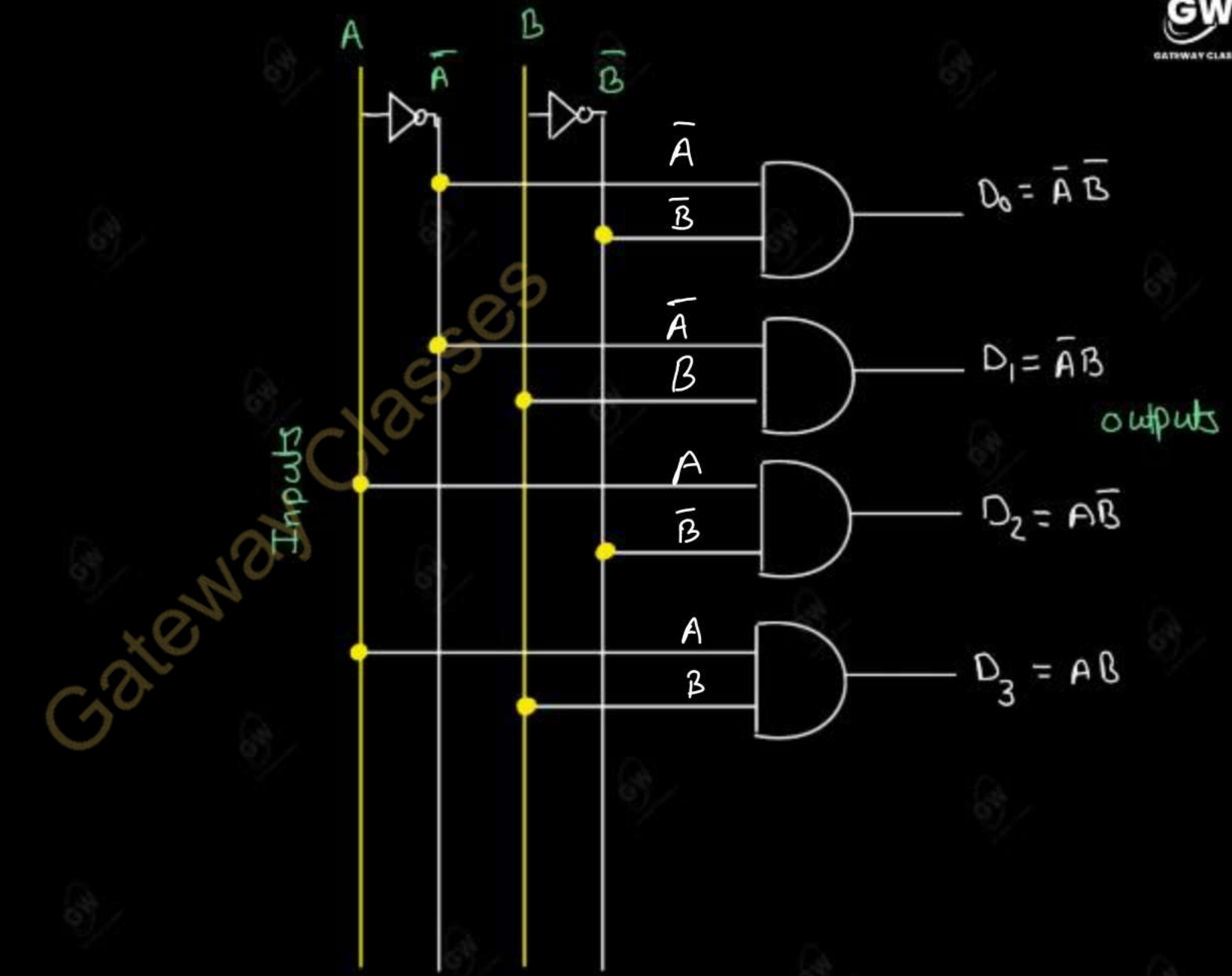
Circuit Realization

$D_0 = \bar{A}\,\bar{B}$

$D_1 = \bar{A}B$

$D_2 = A\bar{B}$

$D_3 = AB$

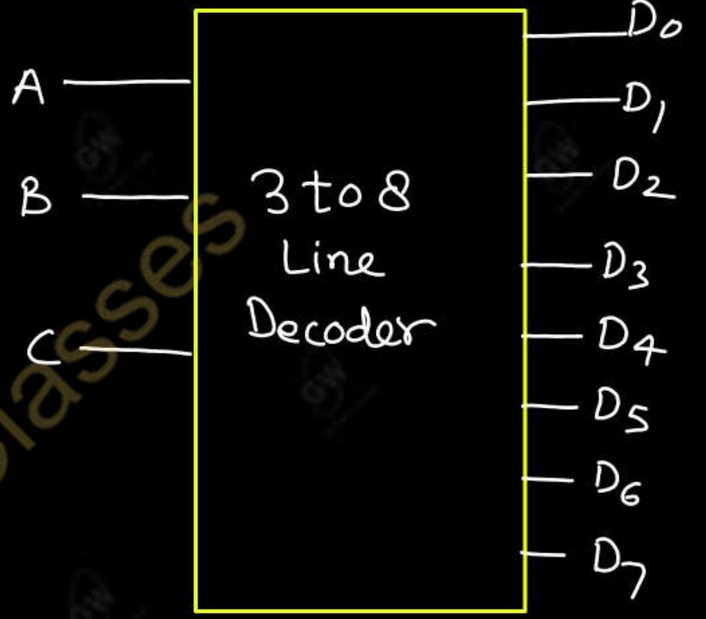


Gateway Classes

A 3 to 8 Line Decoder

$2^n = 2^3 = 8$ line

Input			output							
A	B	C	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



$D_0 = \bar{A}\bar{B}\bar{C}$

$D_1 = \bar{A}\bar{B}C$

$D_2 = \bar{A}B\bar{C}$

$D_3 = \bar{A}BC$

$D_4 = A\bar{B}\bar{C}$

$D_5 = A\bar{B}C$

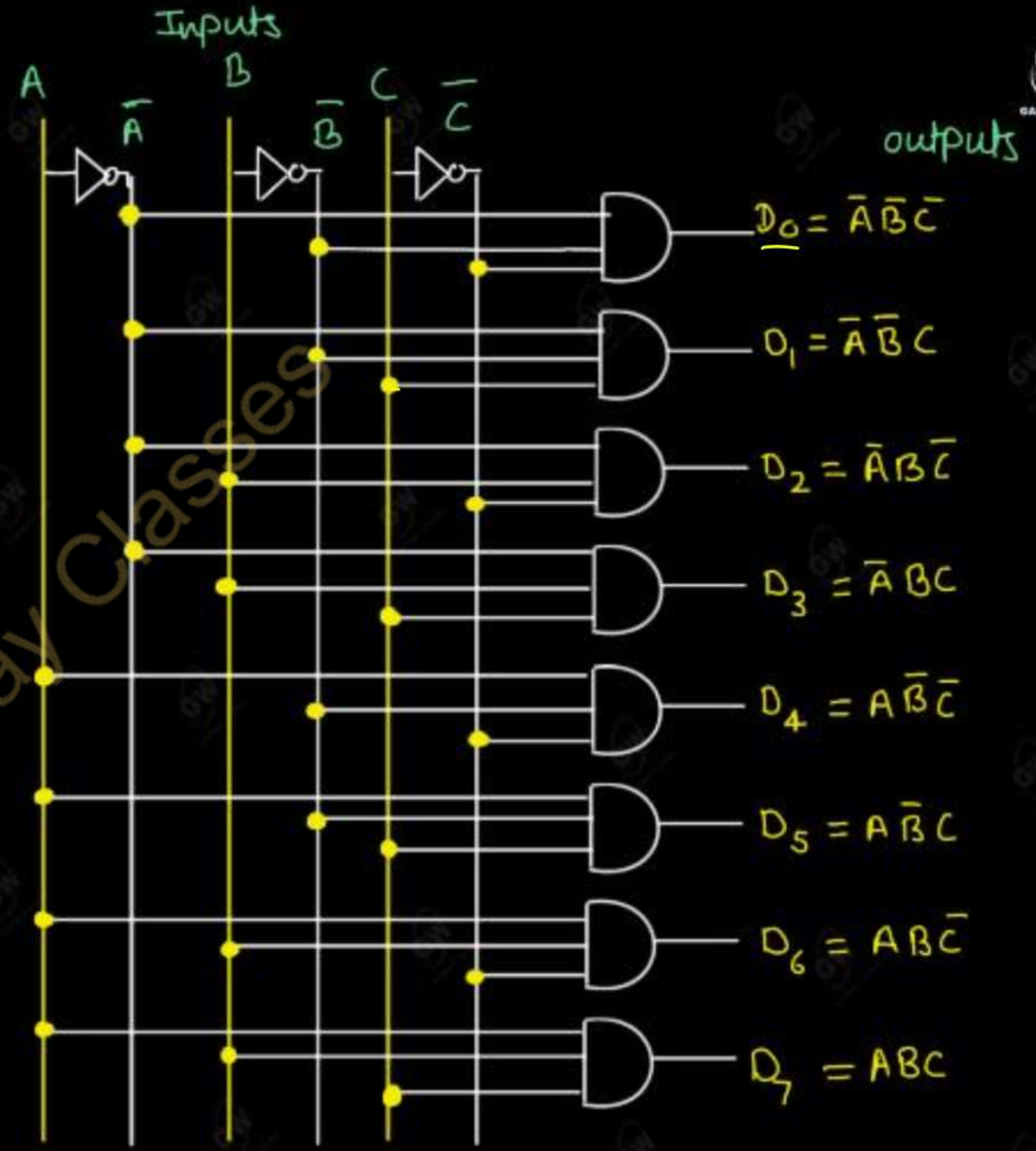
$D_6 = AB\bar{C}$

$D_7 = ABC$

Gateway Classes

A 3 to 8 Line Decoder

Circuit Realization



Gateway Classes

Design a 4 to 16 Line Decoder using 3 to 8 Line Decoder

Circuit Realization

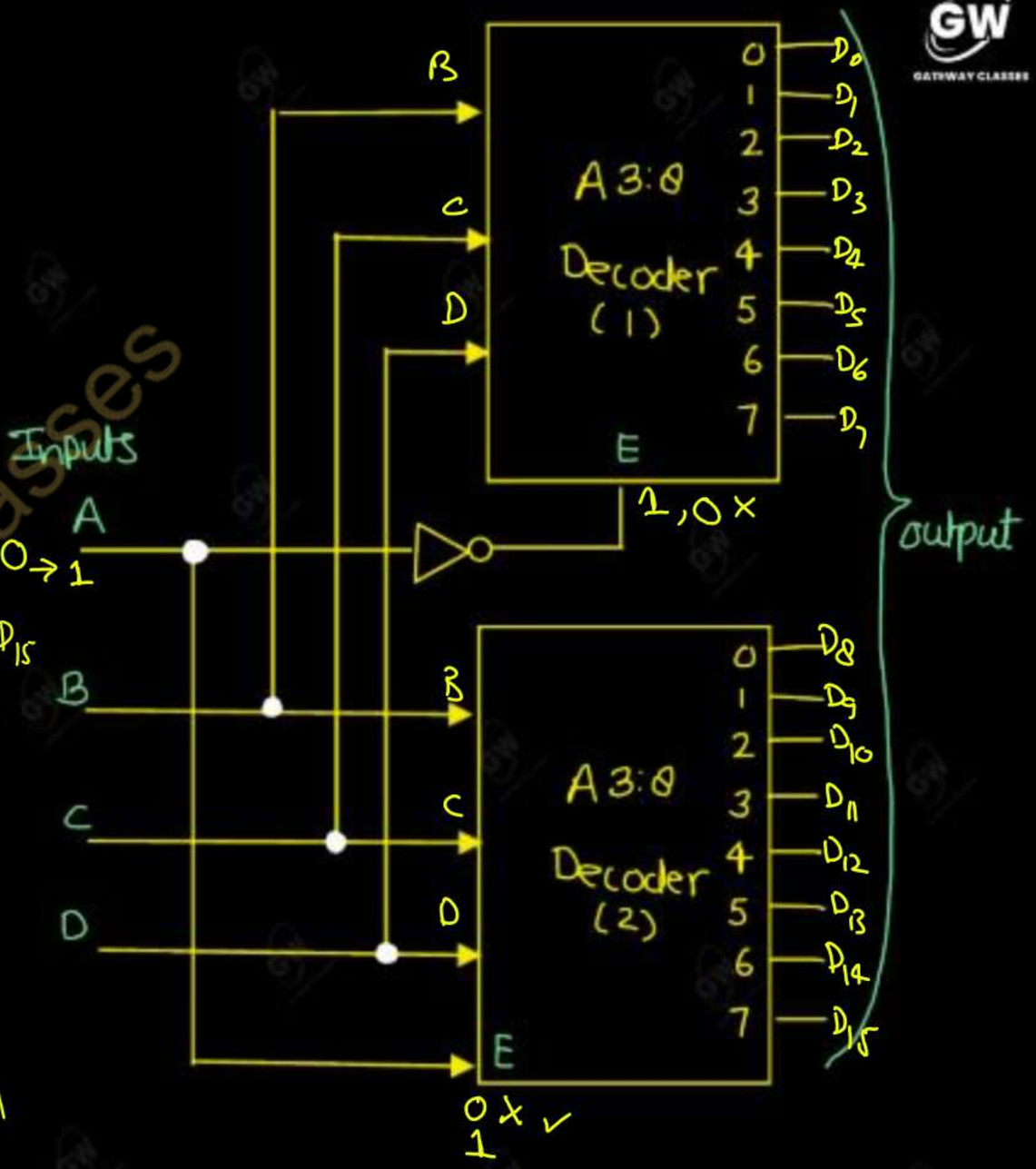
When $A3 = 0$

Decoder 1 active → generates outputs $D0$ to $D7$.

When $A3 = 1$

Decoder 2 active → generates outputs $D8$ to $D15$.

A	B	C	D	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}
0	0	0	0	1															
0	0	0	1		1														
0	0	1	0			1													
0	0	1	1				1												
0	1	0	0					1											
0	1	0	1						1										
0	1	1	0							1									
0	1	1	1								1								
1	0	0	0									1							
1	0	0	1										1						
1	0	1	0											1					
1	0	1	1												1				
1	1	0	0													1			
1	1	0	1														1		
1	1	1	0															1	
1	1	1	1																1

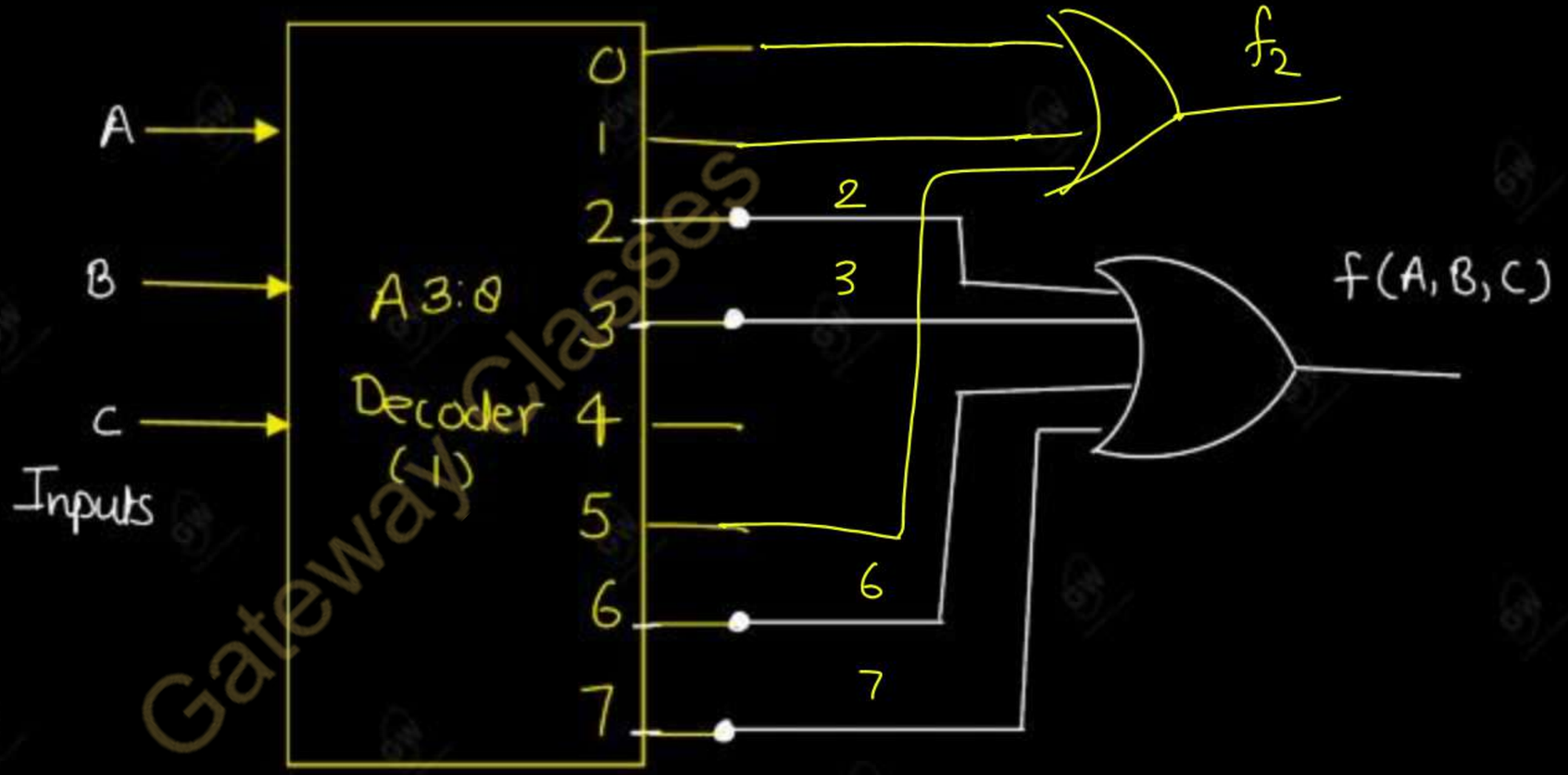


Gateway Classes

AKTU- 2022-23

Implement f (A,B,C) = m (2,3,6,7) using 3 to 8 line decoder

m (2,3,6,7)



Gateway Classes

Applications of Decoders

- Memory Address Decoding → Selecting memory location in RAM/ROM.
- Instruction Decoding → In CPUs, to decode machine instructions.
- 7-Segment Display Decoder → Converts binary-coded input to display digits.
- Demultiplexing → Used to connect one input line to one of many output lines.
- Digital communication systems.

Gateway Classes

AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-2 Lec-5



Avinash Sir

Gateway Classes

Today's target

- Encoders
- Multiplexers
- PYQ

UNIT-2 : Combinational circuits:

- Analysis and Design of combinational circuits
- MSI devices like comparators, multiplexers, demultiplexers, encoder, decoder,
- circuit realization using Multiplexers and decoders, half and full adders, subtractors, serial and parallel adders, BCD adder, barrel shifter and ALU.

Gateway Classes

MSI or Medium scale integrated circuits are a set of basic, elementary logic circuits. They are available as ICs (integrated chips) and implement specific, commonly used digital functions such as: Multiplexing and Demultiplexing ,Encoding and Decoding)

Encoders



An encoder is a combinational circuit that performs the reverse operation of a decoder. It usually has 2^n input lines and n output lines. The encoder converts active inputs into a unique binary code representing the position of the active input.

When one of the input lines is active (high), the encoder produces a unique binary output corresponding to that input's position.

Example: In an 8-to-3 encoder, if input I5 is active, the output is the binary code for 5 (101).

Types of Encoders

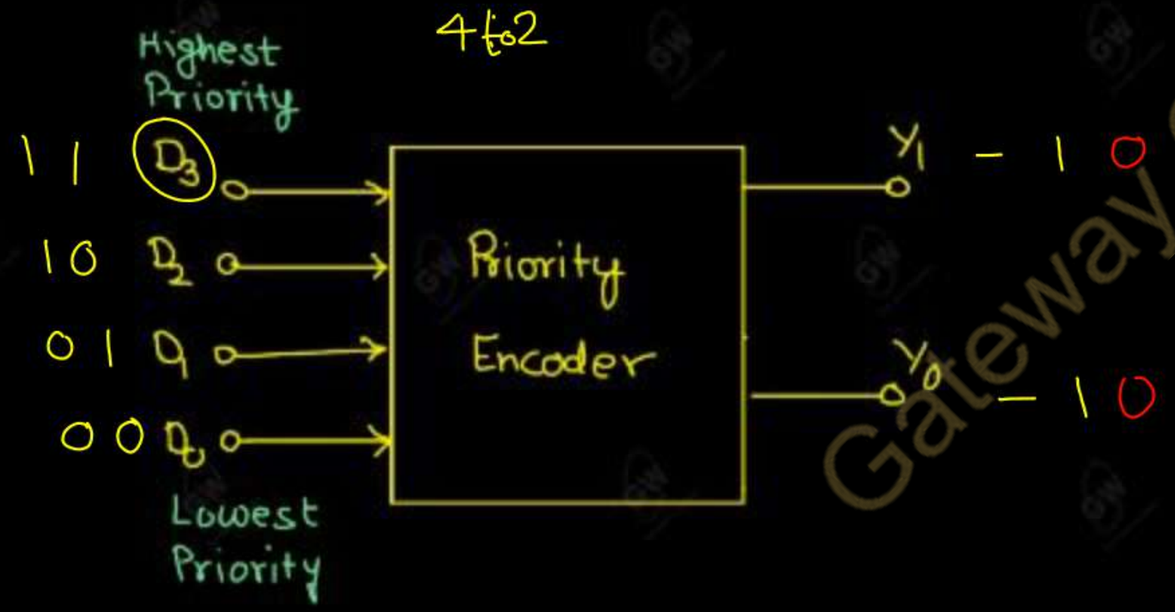
i. **Priority Encoder** (ii) **Decimal to BCD Encoder** (iii) **Octal/Hexadecimal to Binary Encoder:**

Gateway Classes

Priority Encoders

AKTU-2023-24

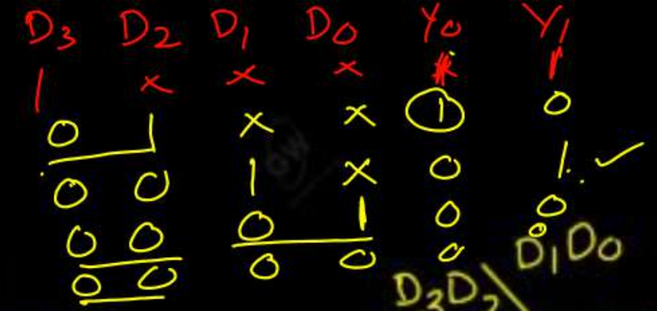
This is a special type of encoder. Priority is given to the input lines. If two or more input line are 1 at the same time, then the input line with highest priority will be considered. There are four input D0, D1, D2, D3 and two output Y0, Y1. Out of the four input D3 has the highest priority and D0 has the lowest priority. That means if D3 = 1 then Y1 = 1 irrespective of the other inputs.

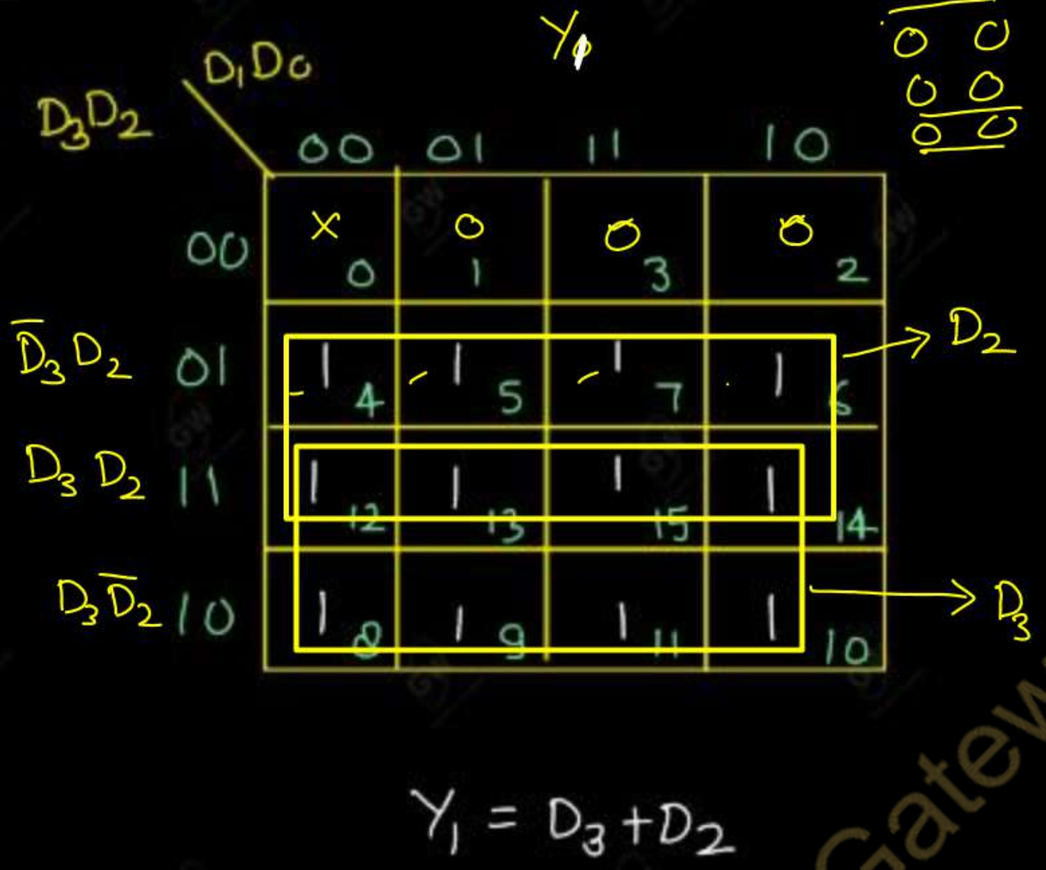


INPUT				OUTPUT	
D3 (Highest)	D2	D1	D0 (Lowest)	Y0	Y1
0	0	0	0	x	x
0	0	0	1	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1

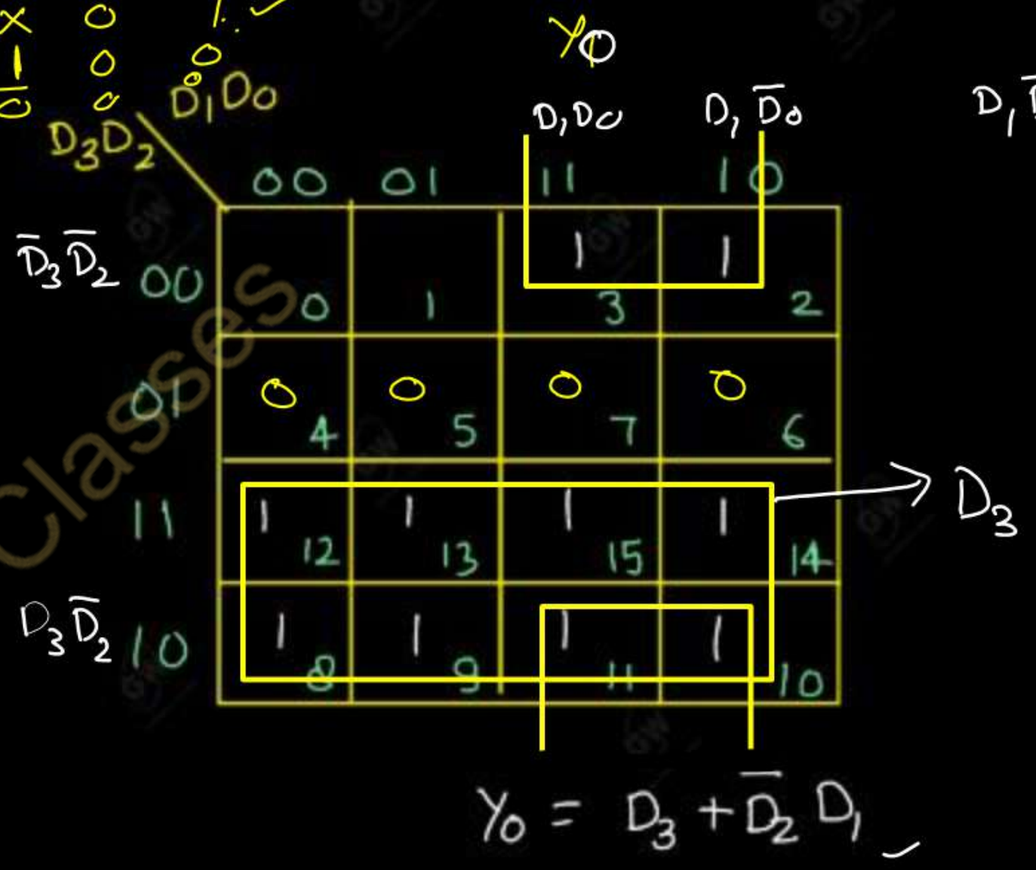
Gateway Classes

Priority Encoders





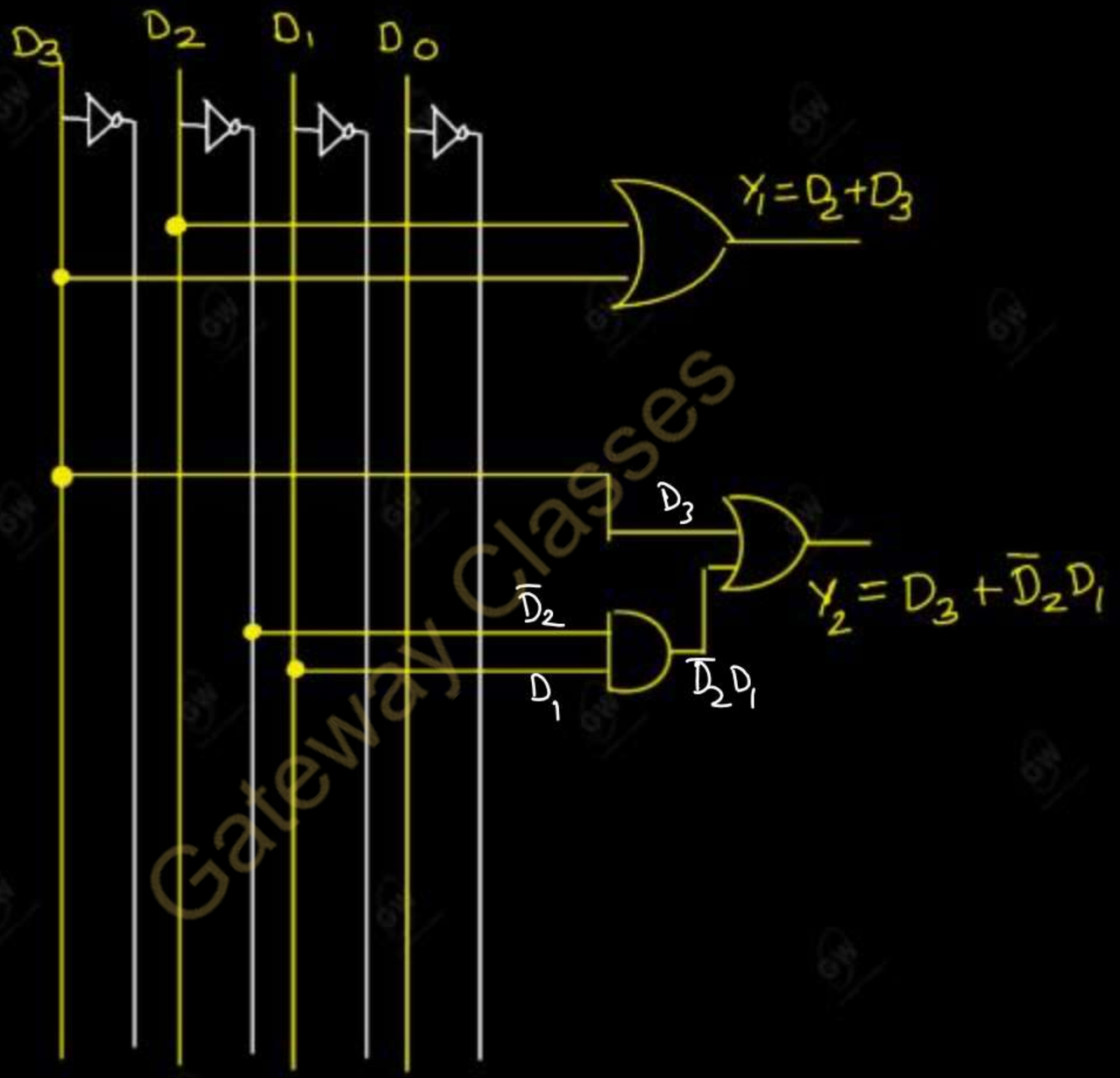
$$Y_1 = D_3 + D_2$$



$$Y_0 = D_3 + \bar{D}_2 D_1$$

Gateway Classes

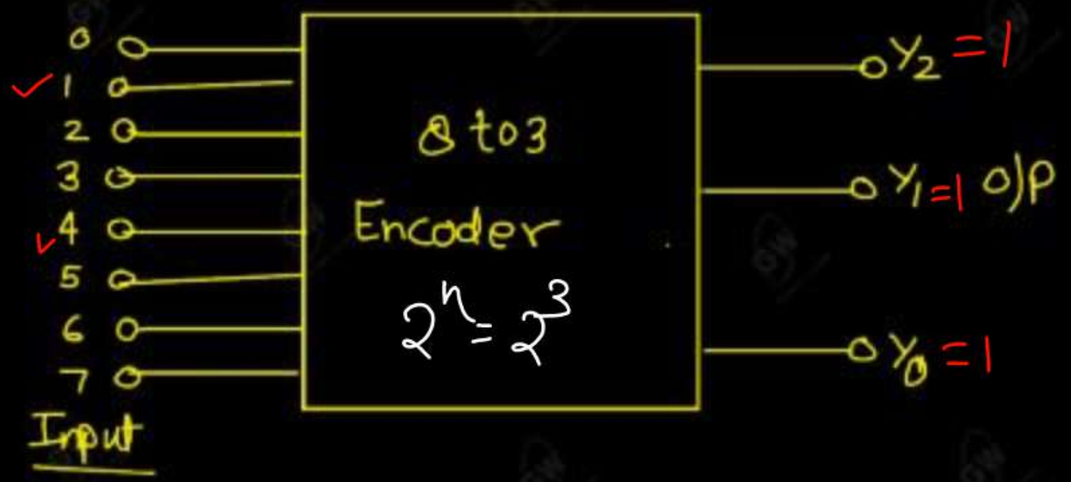
Priority Encoders



Gateway Classes

Octal to Binary Encoder (8 to 3 Encoder)

The 8 to 3 Encoder or octal to Binary encoder consists of 8 inputs: Y7 to Y0 and 3 outputs: A2, A1 & A0. Each input line corresponds to each octal digit value and three outputs generate corresponding binary code.



Input								Output		
D0	D1	D2	D3	D4	D5	D6	D7	Y2	Y1	10
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$Y2 = D7 + D6 + D5 + D4$

$Y1 = D7 + D6 + D3 + D2$

$Y0 = D7 + D5 + D3 + D1$

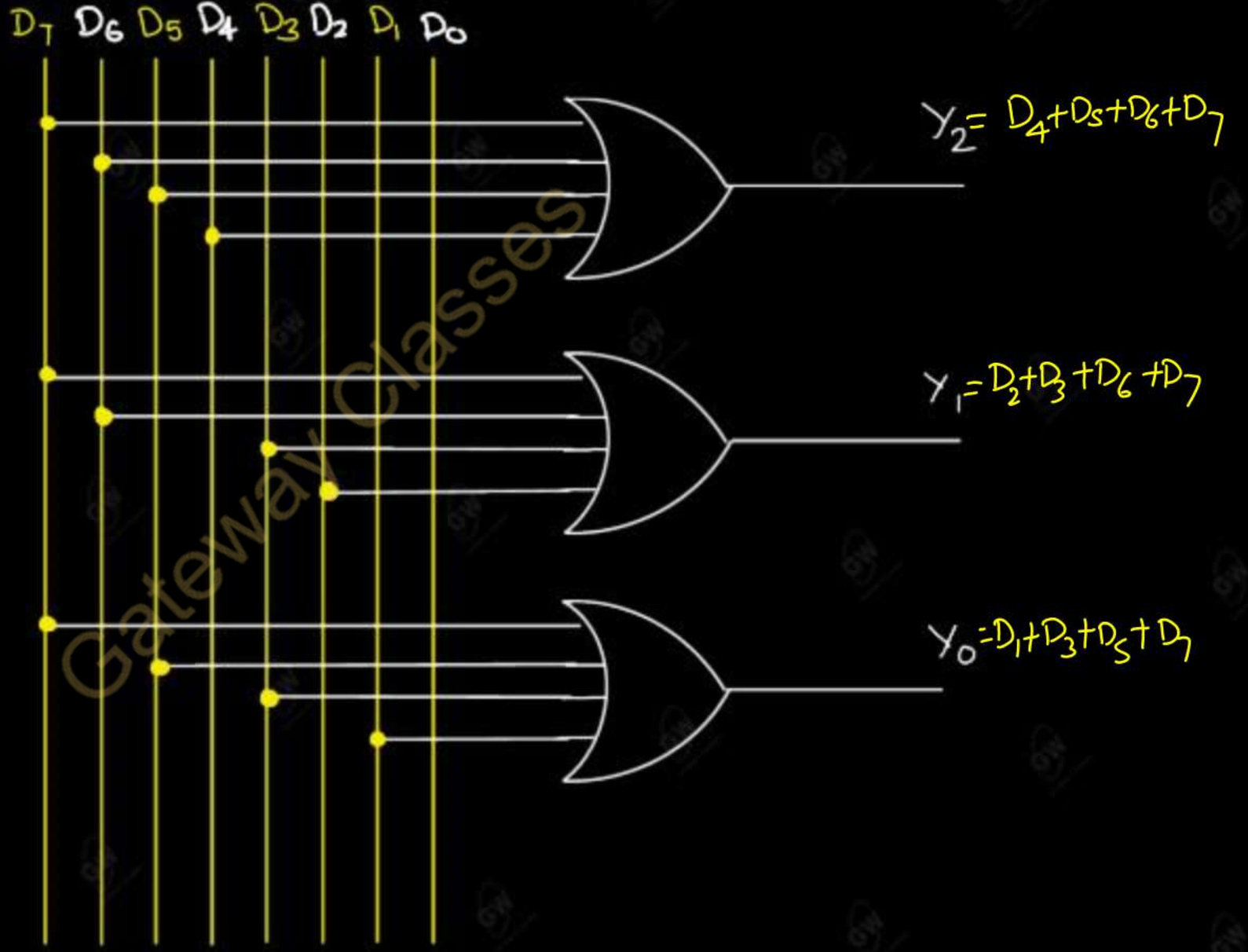
Gateway classes

Octal to Binary Encoder (8 to 3 Encoder)

Y2 = D7 + D6 + D5 + D4

Y1 = D7 + D6 + D3 + D2

Y0 = D7 + D5 + D3 + D1



Gateway Classes

Decimal to BCD Encoder

The decimal-to-binary encoder usually consists of 10 input lines and 4 output lines. Each input line corresponds to each decimal digit and 4 outputs correspond to the BCD code.

$B3 = D9 + D7 + D5 + D3 + D1$

$B2 = D7 + D6 + D3 + D2$

$B1 = D7 + D6 + D5 + D4$

$B0 = D9 + D8$

Input										Output			
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	B0	B1	B2	B3
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

Gateway Classes

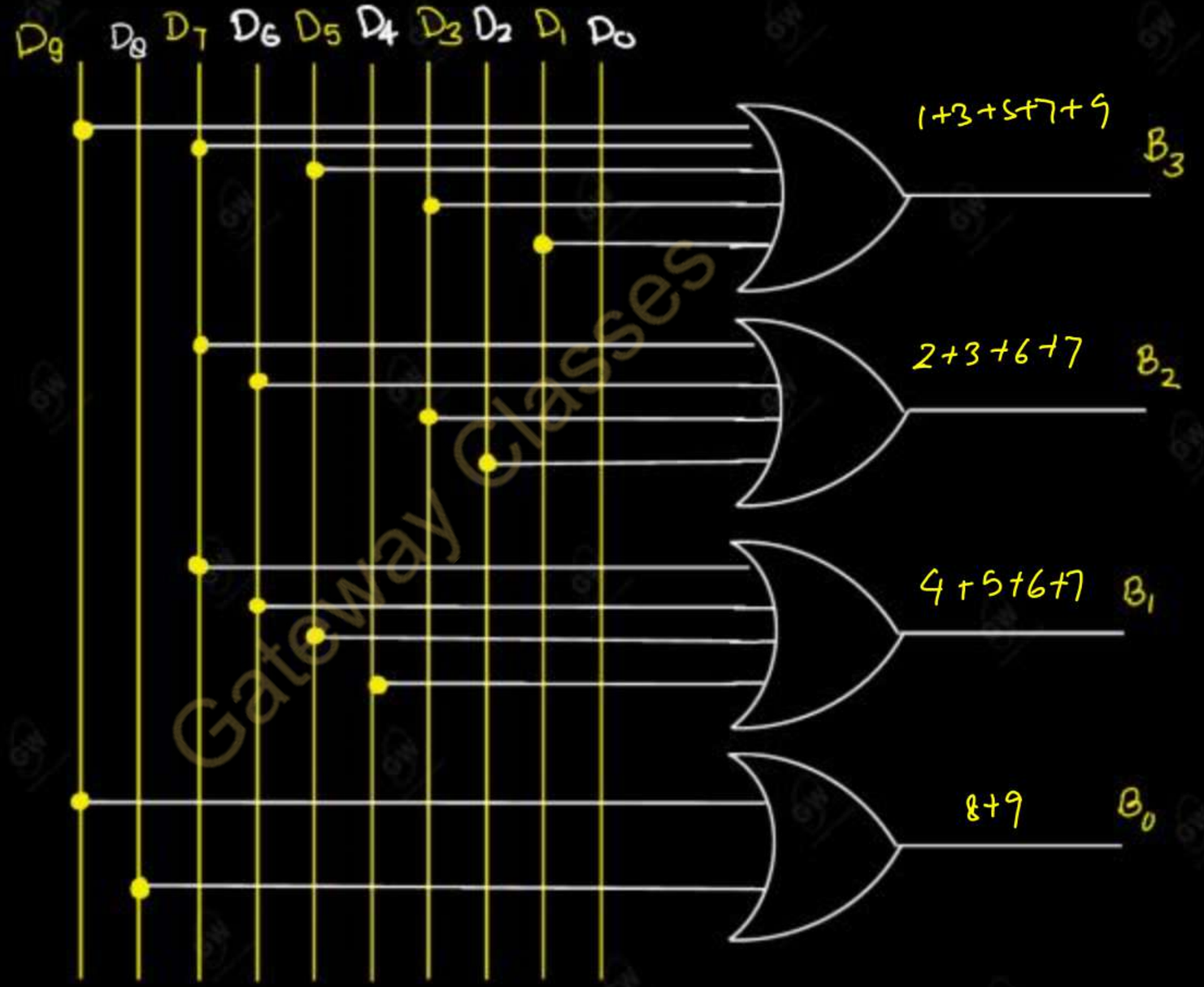
Decimal to BCD Encoder

B3 = D9 + D7 +D5 +D3 + D1

B2 = D7 + D6 + D3 +D2

B1 = D7 + D6 + D5 +D4

B0 = D9 + D8



Gateway Classes

Application of Encoders

- Encoders are used to translate the decimal values to the binary in order to perform binary functions such as addition, subtraction, multiplication, etc.
- Other applications especially for Priority Encoders may include detecting interrupts in microprocessor applications.

Advantages of Encoders in Digital Logic

- Reduction in the number of lines to transmit information from multiple I/p to a single o/p.
- Improved reliability By converting multiple inputs into a single serial code
- Improve performance by reducing time required to transmit information.

Disadvantages of Encoders in Digital Logic

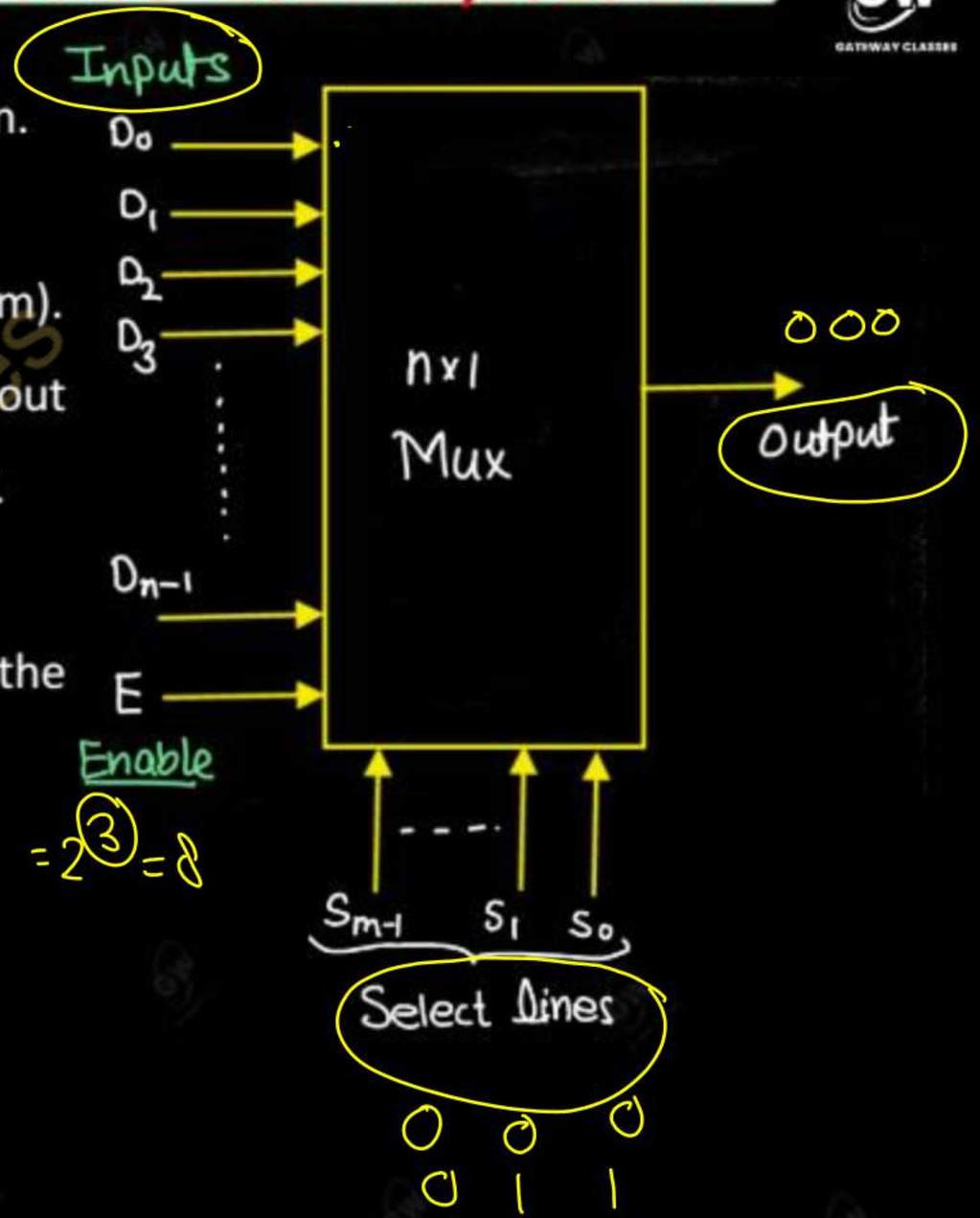
- Encoders are typically more complex than multiplexers , and require additional components.
- Suitable for applications where a parallel set of inputs must be converted into a serial code.
- Limited flexibility

Gateway Classes

GW GATEWAY CLASSES

Multiplexers (Data Selector / Many to one)

- There are n-data inputs, one output and m select inputs with $2^m = n$.
- It selects one of the n data inputs and routes it to the output.
- The selection of one of the n inputs is done by the selected inputs(m).
- Depending on the digital code applied at the selected inputs, one out of n data sources is selected and transmitted to the single output Y.
- E is enable (strobe) input which is useful for the cascading.
- E is generally an active low terminal that means it will perform the required operation when it is low.



$2^{(m)}$ = Input $= 2^{(3)} = 8$

m = Select line

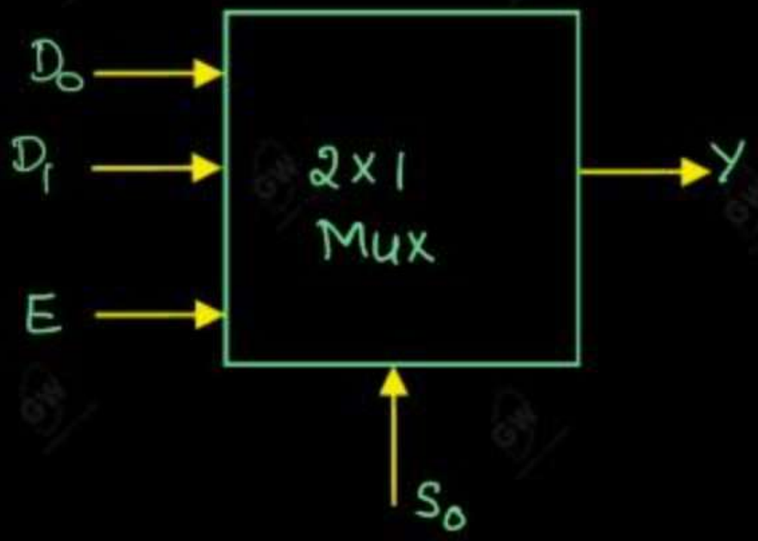
- Multiplexers are classified As

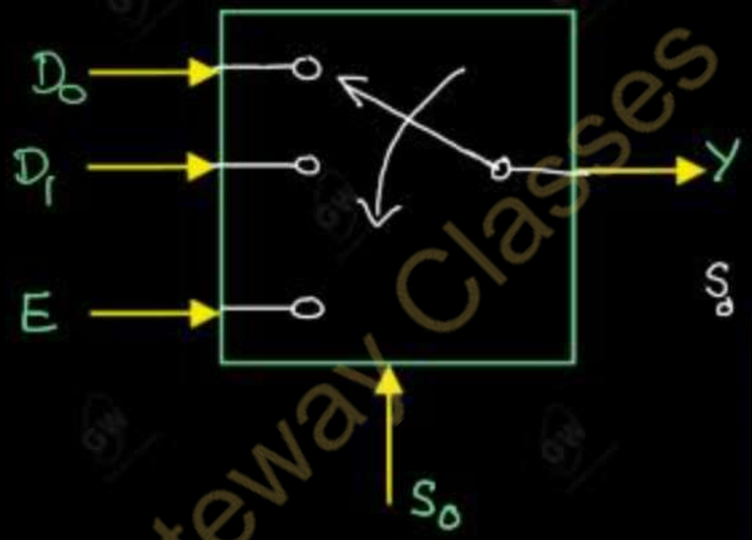
  2:1 , 4:1 , 8:1 , 16: 1 , 32:1 Mux

Gateway Classes

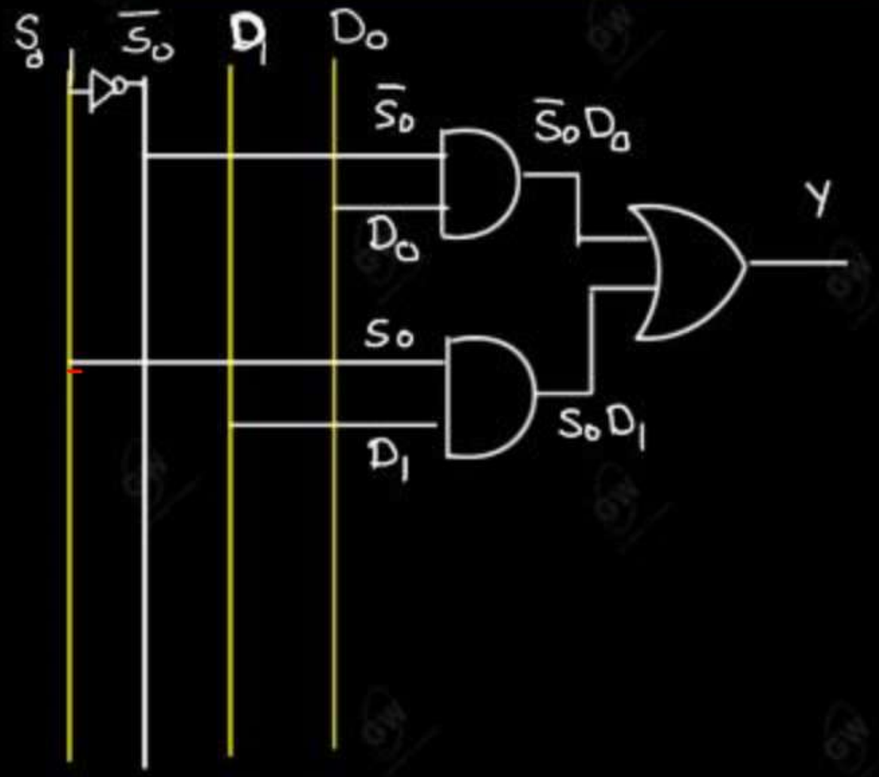
2:1 Multiplexer

The 2x1 mux has two input lines, one output line, and a single selection line.





INPUT			OUTPUT
S0	D_0	D_1	Y
0	0	x	0
0	1	x	1
1	x	0	0
1	x	1	1



Logical Expression of 2x1 Mux

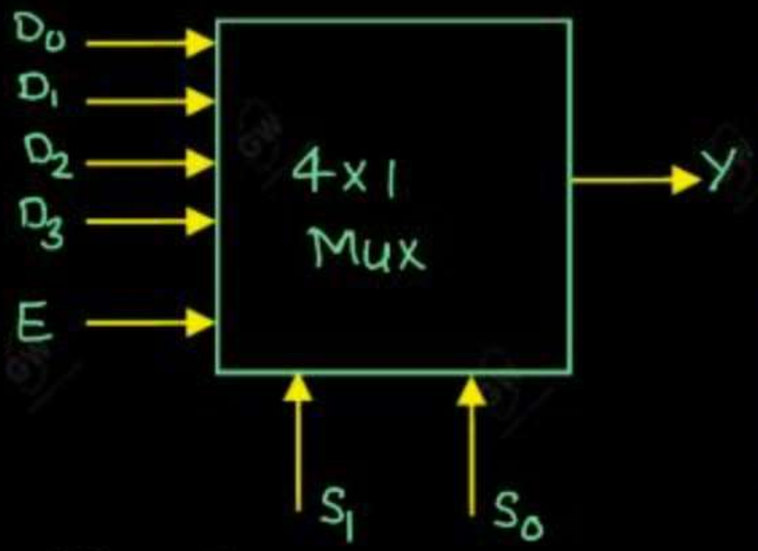
$$Y = \bar{S}_0 D_0 + S_0 D_1$$

Gateway Classes

4:1 Multiplexer

It is a multiplexer that has 4 inputs and a single output and 2 select lines. $4 = 2^{2} \rightarrow$ S.L

The number of the Selection lines given by $m = \log_2 n$



INPUT		OUTPUT
S0	S1	Y
0	0	D0
0	1	D1
1	0	D2
1	1	D3

From Truth table

$Y = D_0$ for $S_1S_0 = 00$

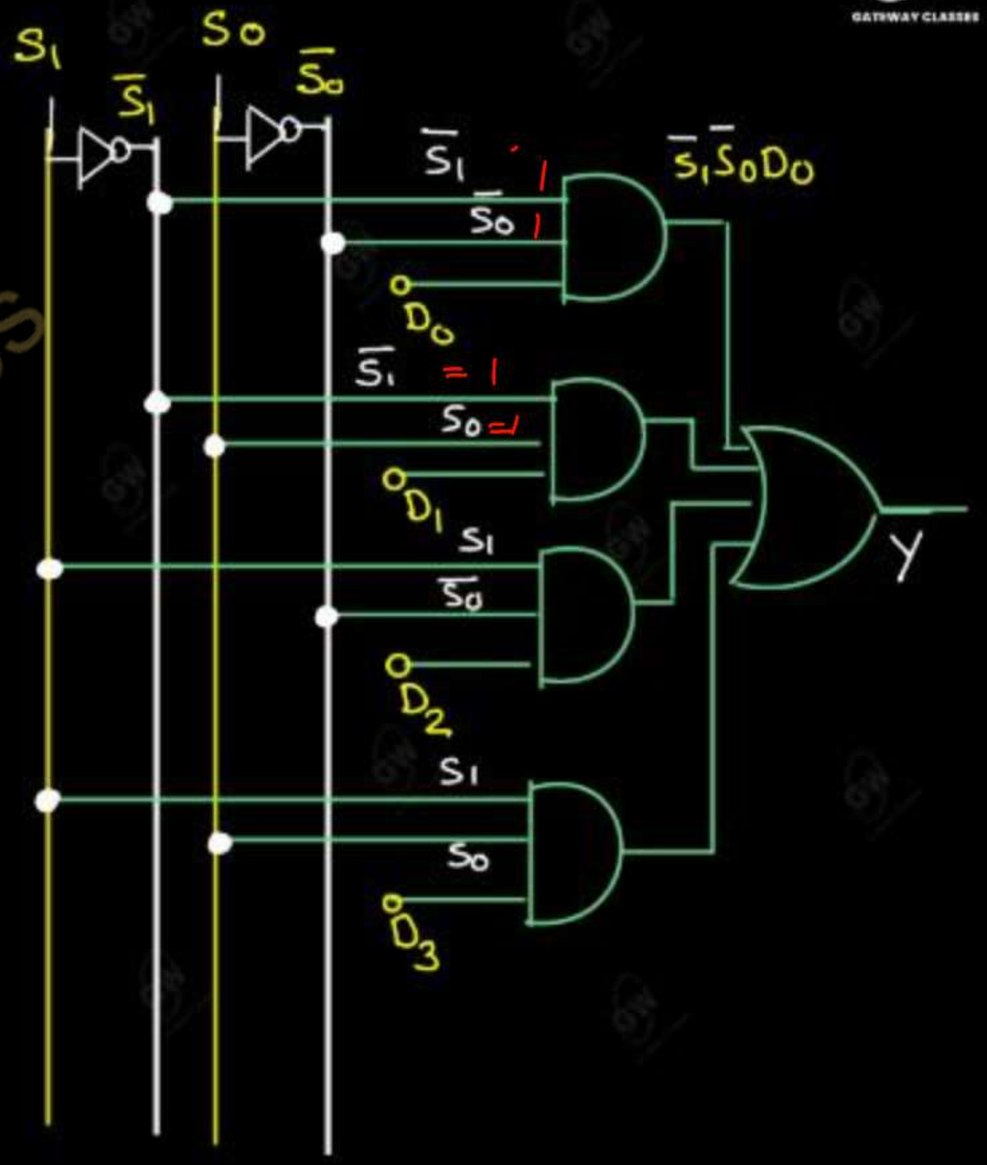
$Y = D_1$ for $S_1S_0 = 01$

$Y = D_2$ for $S_1S_0 = 10$

$Y = D_3$ for $S_1S_0 = 11$

So output will be high when Select input is high (D_0, D_1, D_2 ---)
So the output is in SOP form as

$$Y = \bar{S}_1\bar{S}_0D_0 + \bar{S}_1S_0D_1 + S_1\bar{S}_0D_2 + S_1S_0D_3$$



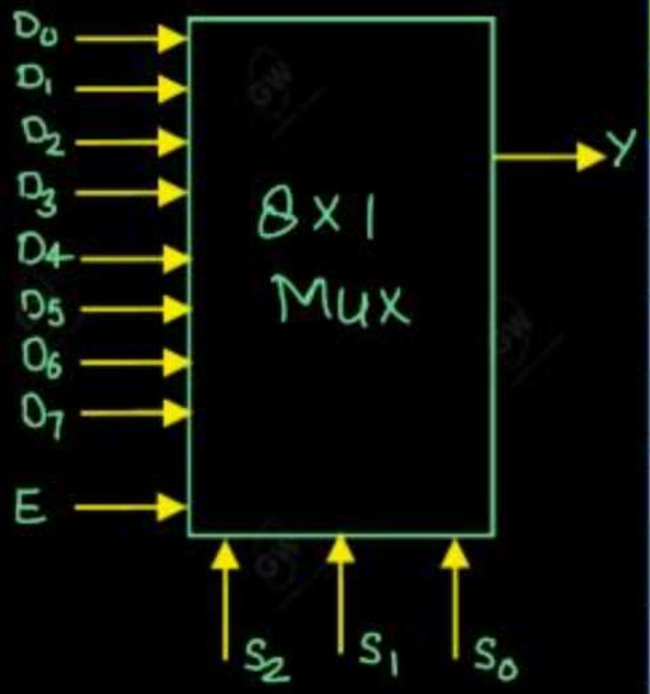
Gatewayclasses

8:1 Multiplexer

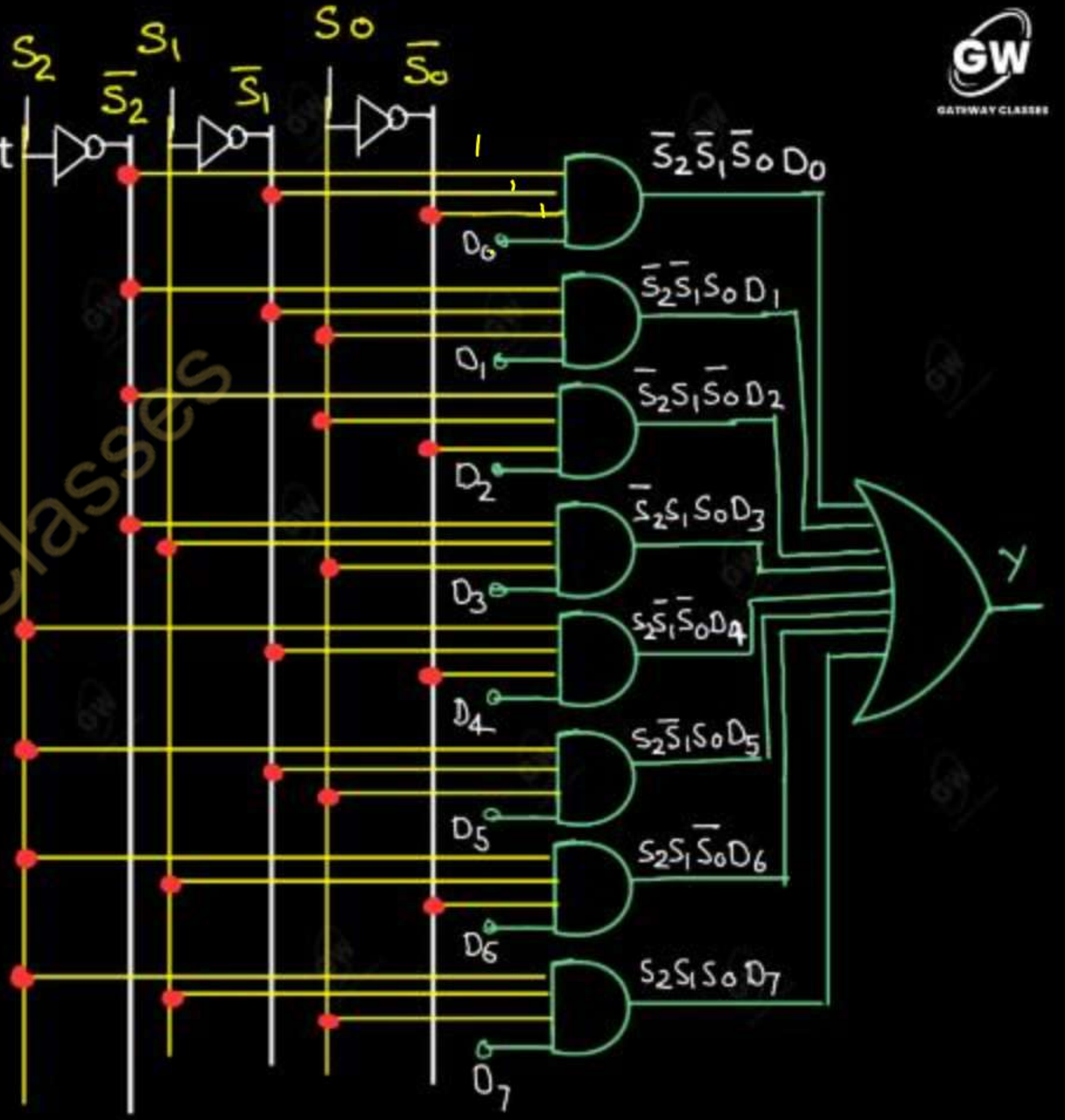
$Y = \bar{S}_0\bar{S}_1\bar{S}_2D_0 + S_0\bar{S}_1\bar{S}_2D_1 + \bar{S}_0S_1\bar{S}_2D_2 + S_0S_1\bar{S}_2D_3 + \bar{S}_0\bar{S}_1S_2D_4 + S_0\bar{S}_1S_2D_5 + \bar{S}_0S_1S_2D_6 + S_0S_1S_2D_7$

It is a multiplexer that has 8 inputs and a single output and 3 select lines. $8 = 2^3$ → S.L.

The number of the Selection lines given by $m = \log_2 n$



INPUT			OUTPUT
S0	S1	S2	Y
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

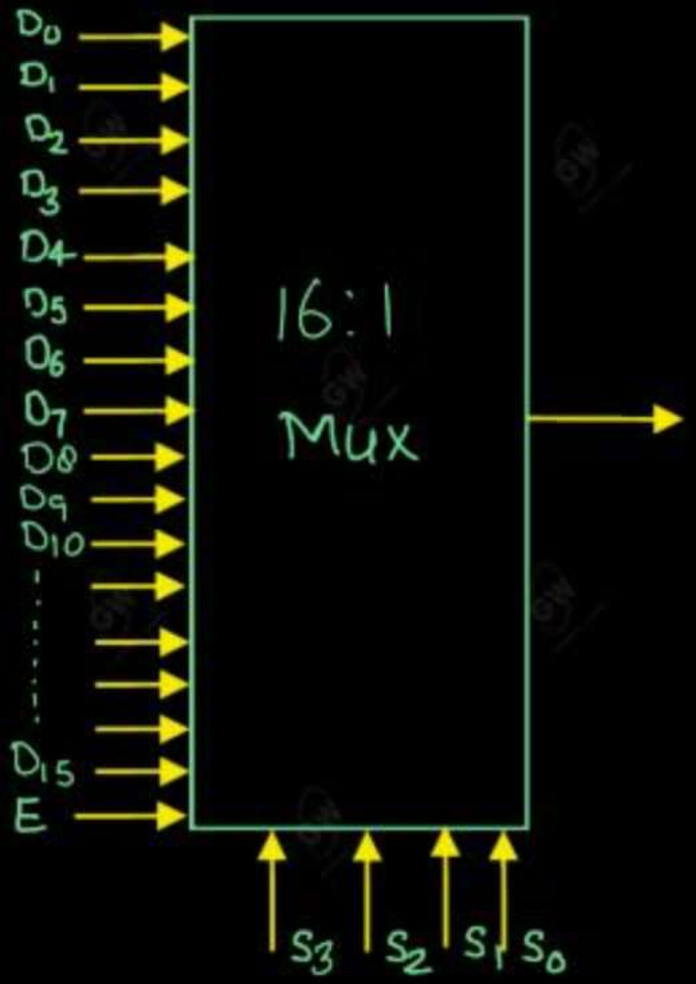


Gateway Classes

16:1 Multiplexer

It is a multiplexer that has 16 inputs and a single output and 4 select lines. $16 = 2^{\oplus} =$ S.L

The number of the Selection lines given by $m = \log_2 n$



INPUT				OUTPUT
S0	S1	S2	S3	Y
0	0	0	0	D0
0	0	0	1	D1
0	0	1	0	D2
0	0	1	1	D3
0	1	0	0	D4
0	1	0	1	D5
0	1	1	0	D6
0	1	1	1	D7
1	0	0	0	D8
1	0	0	1	D9
1	0	1	0	D10
1	0	1	1	D11
1	1	0	0	D12
1	1	0	1	D13
1	1	1	0	D14
1	1	1	1	D15

$\bar{S}_0\bar{S}_1\bar{S}_2\bar{S}_3 D_0$

$\bar{S}_0 S_1 \bar{S}_2 S_3 D_1$

Gateway Classes

AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-2 Lec-6

Gateway Classes



Avinash Sir

Today's target

- Circuit realization using Multiplexers
- PYQ

UNIT-2 : Combinational circuits:

- Analysis and Design of combinational circuits
- MSI devices like comparators, multiplexers, demultiplexers, encoder, decoder,
- circuit realization using Multiplexers and decoders, half and full adders, subtractors, serial and parallel adders, BCD adder, barrel shifter and ALU.

Gateway Classes

Multiplexer tree

32:1 → 4:1

$\frac{32}{4} = 8$, $\frac{8}{4} = 2$, $\frac{2}{4} = 0.5 \simeq 1$

8+2+1 = 11

A multiplexer tree (or mux tree) is a hierarchical way to build a large multiplexer (many-to-one selector) out of smaller multiplexers.

For Example "An 8:1 multiplexer can be made with two 4:1 multiplexers and one 2:1 multiplexer. The two 4:1 outputs feed into the 2:1, giving total select bits 3."

How To Calculate no. of Mux Required in a specific condition

Ex.:- How many MUX required to design 16:1 Mux using 2:1 MUX or 4:1 MUX

16:1 MUX By 2:1

2|16 , 8

2|8 , 4

2|4 , 2

2|2 , 1

8+4+2+1 = 14

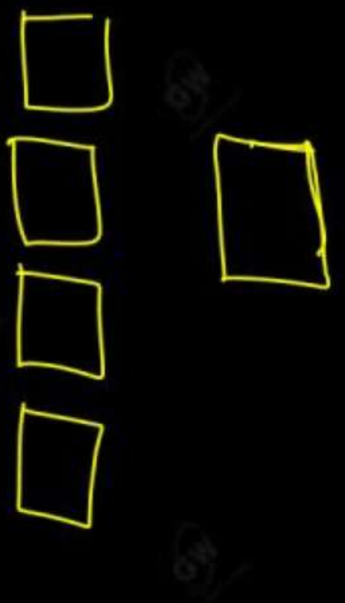


4:1

$\frac{16}{4} = 4$

$\frac{4}{4} = 1$

Total 4+1



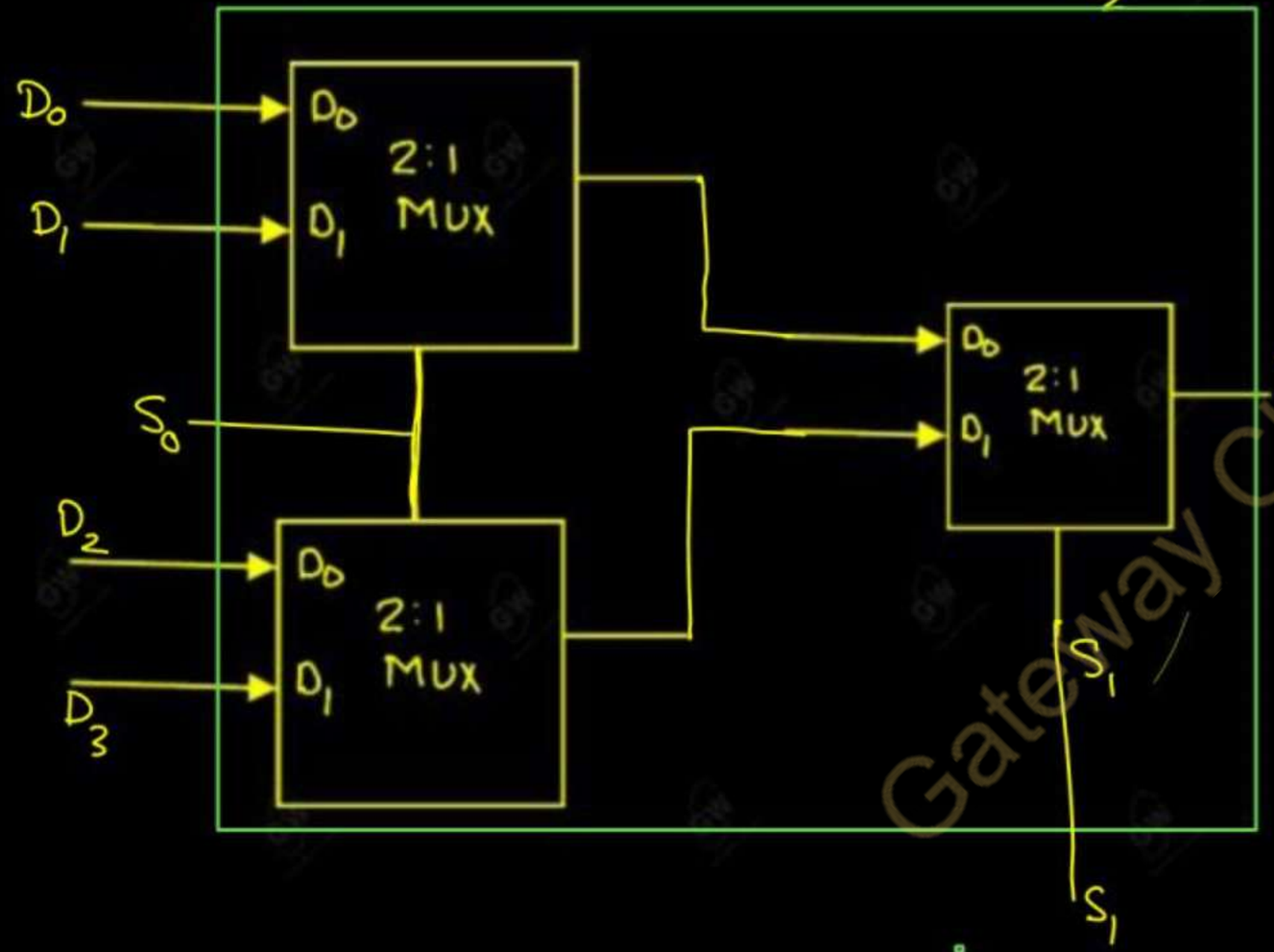
Gateway Classes

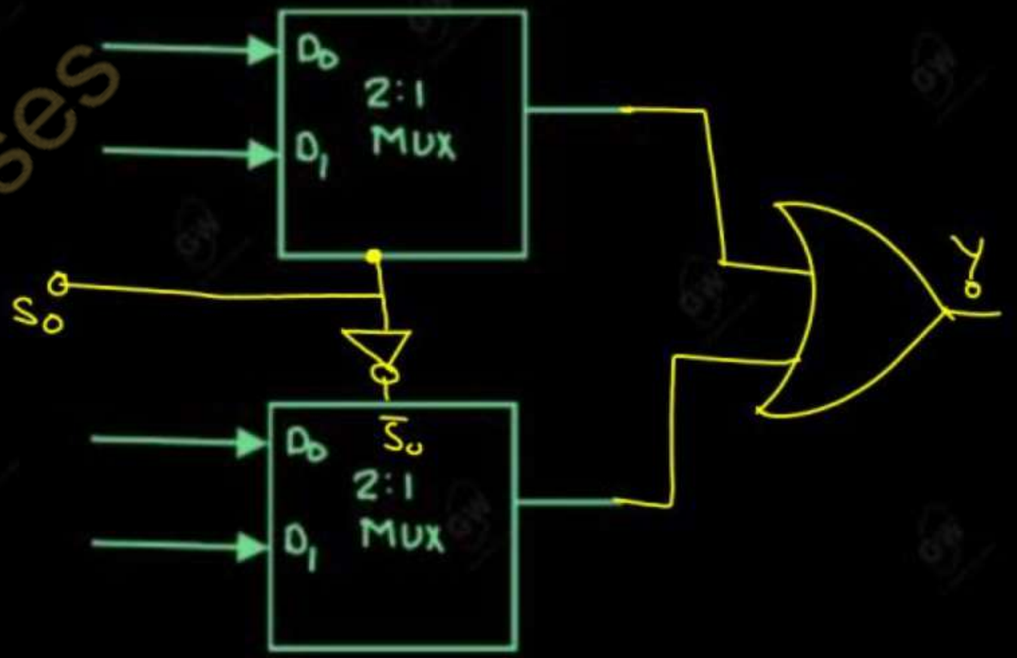
Q.1 . Design 4:1 MUX using 2:1 MUX

$\frac{4}{2} = 2$

$\frac{2}{2} = 1$

(2+1) = 3





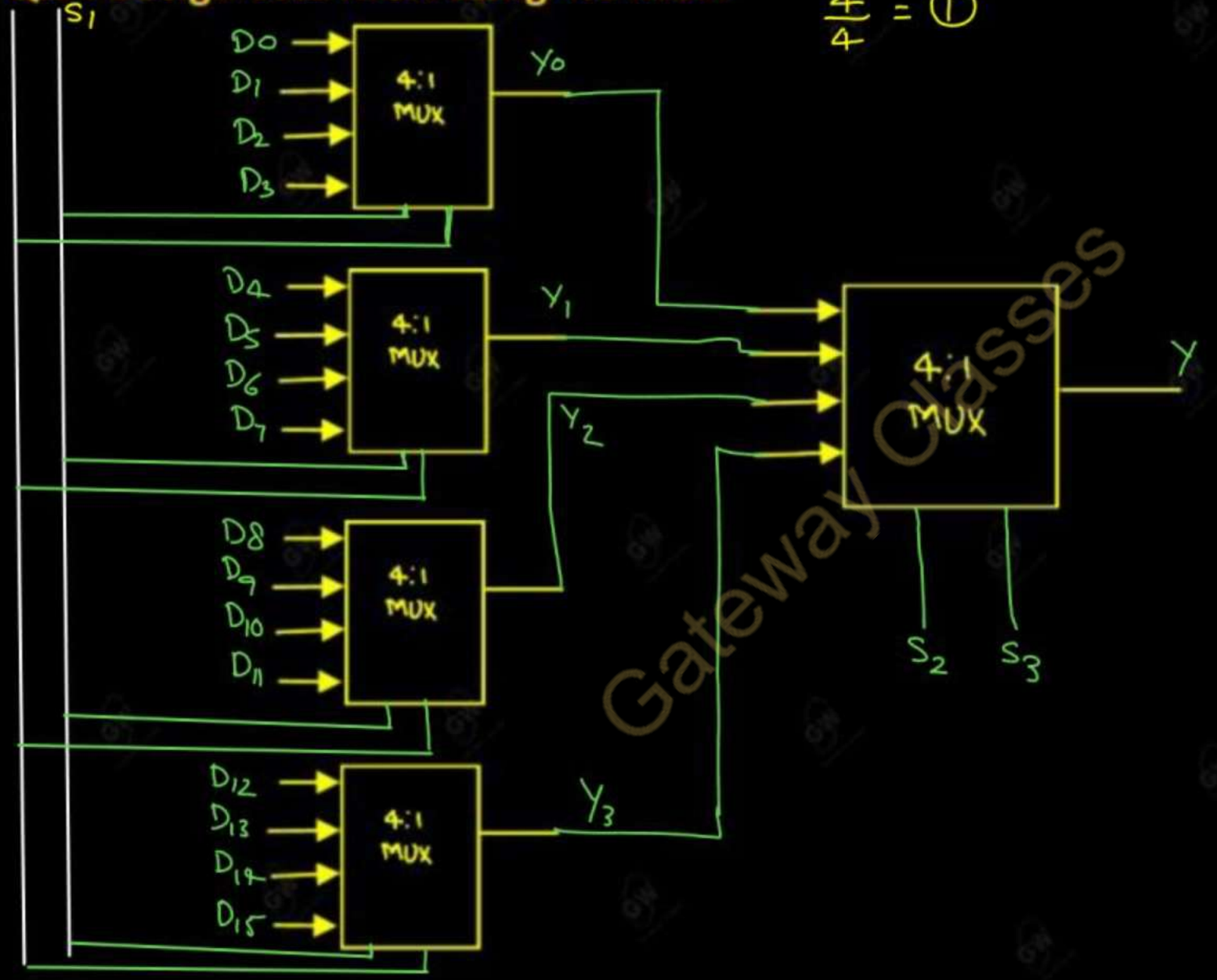
Gateway Classes

Q.2 . Design 16:1 MUX using 4:1 MUX

$\frac{16}{4} = 4$

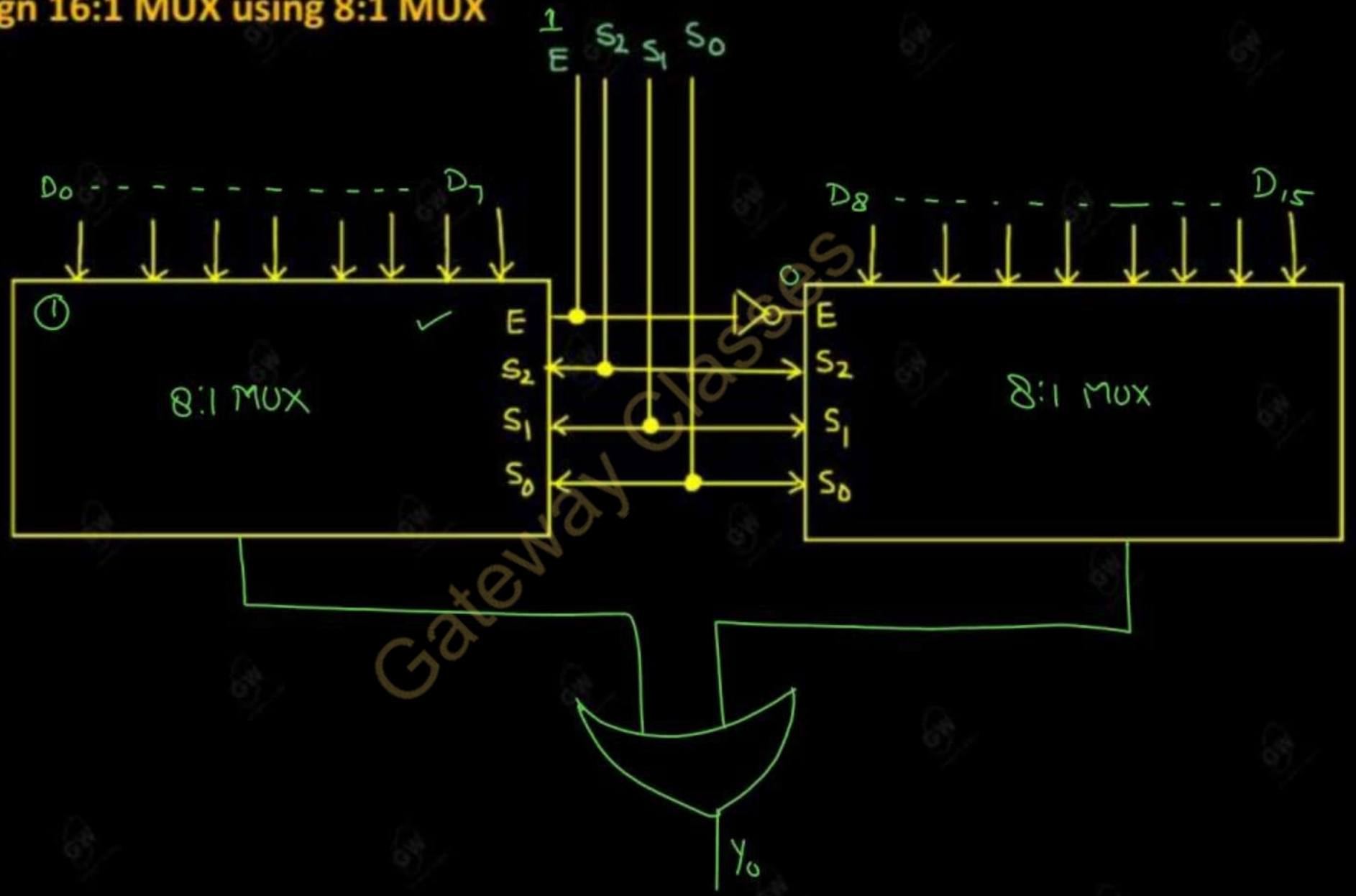
$\frac{4}{4} = 1$

AKTU- 2024-25



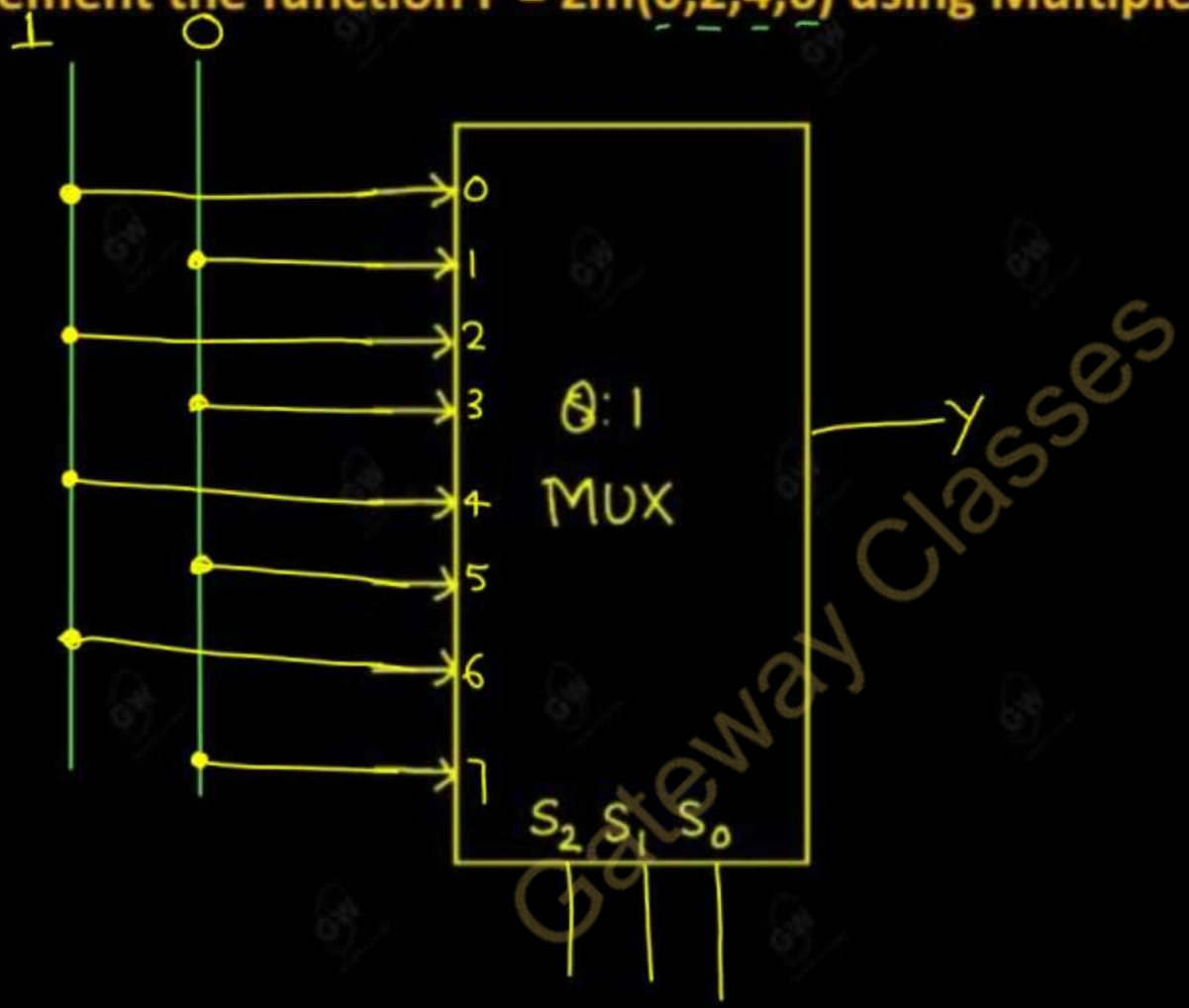
Gateway Classes

Q.3 . Design 16:1 MUX using 8:1 MUX



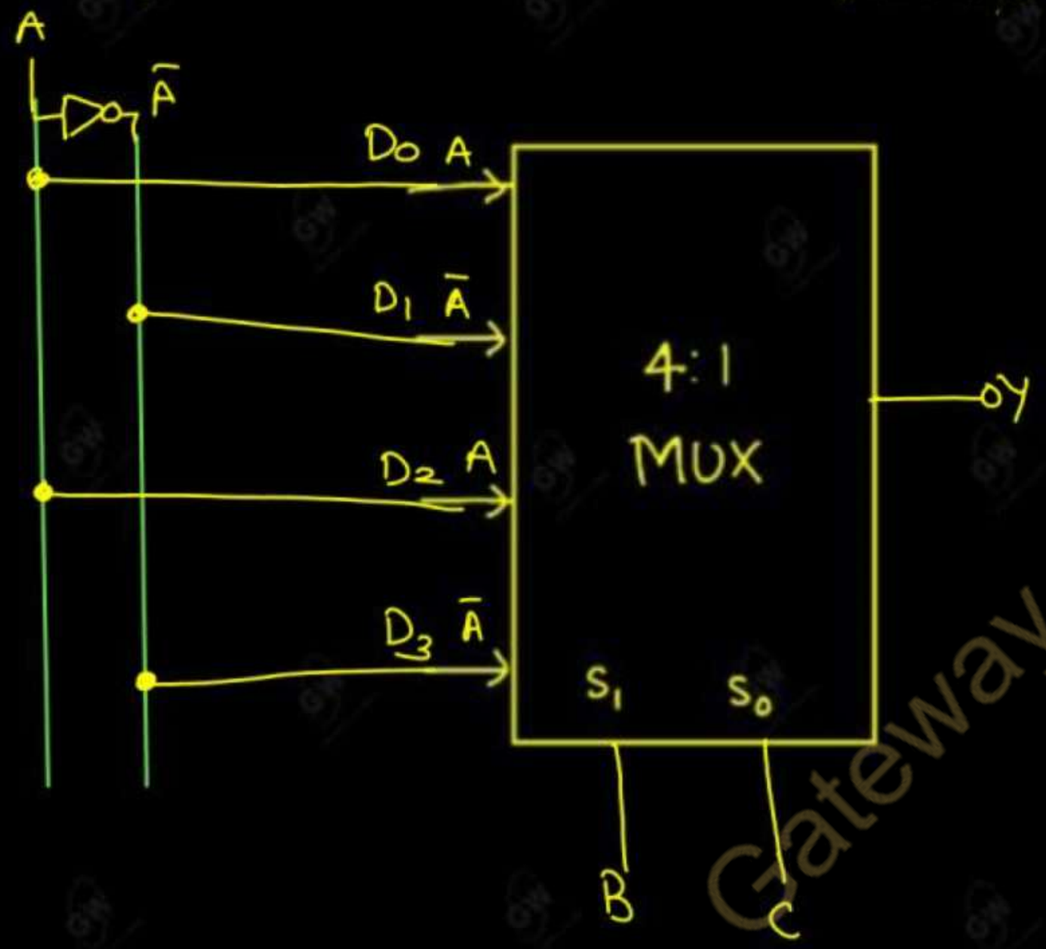
Gateway Classes

Q.4 . Implement the function $F = \Sigma m(0,2,4,6)$ using Multiplexer.



Gateway Classes

Q.5 . Implement the function F = Σm(1,3,4,6) using 4:1 multiplexer.



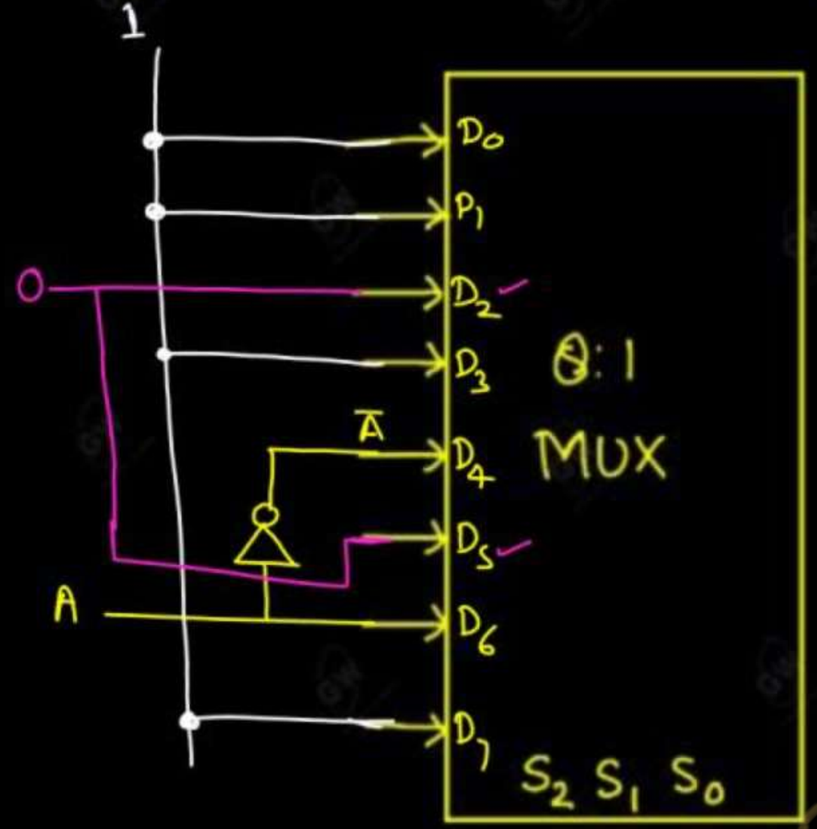
	D_0	D_1	D_2	D_3
$\bar{A}$	0	1	2	3
A	4	5	6	7
I/P to MUX	A	$\bar{A}$	A	$\bar{A}$

"Design Table"

Gateway Classes

Q.6 . Implement the function F = Σm(0,1,3,4,7,8,9,11,14,15) using 8:1 MUX.

AKTU- 2022-23



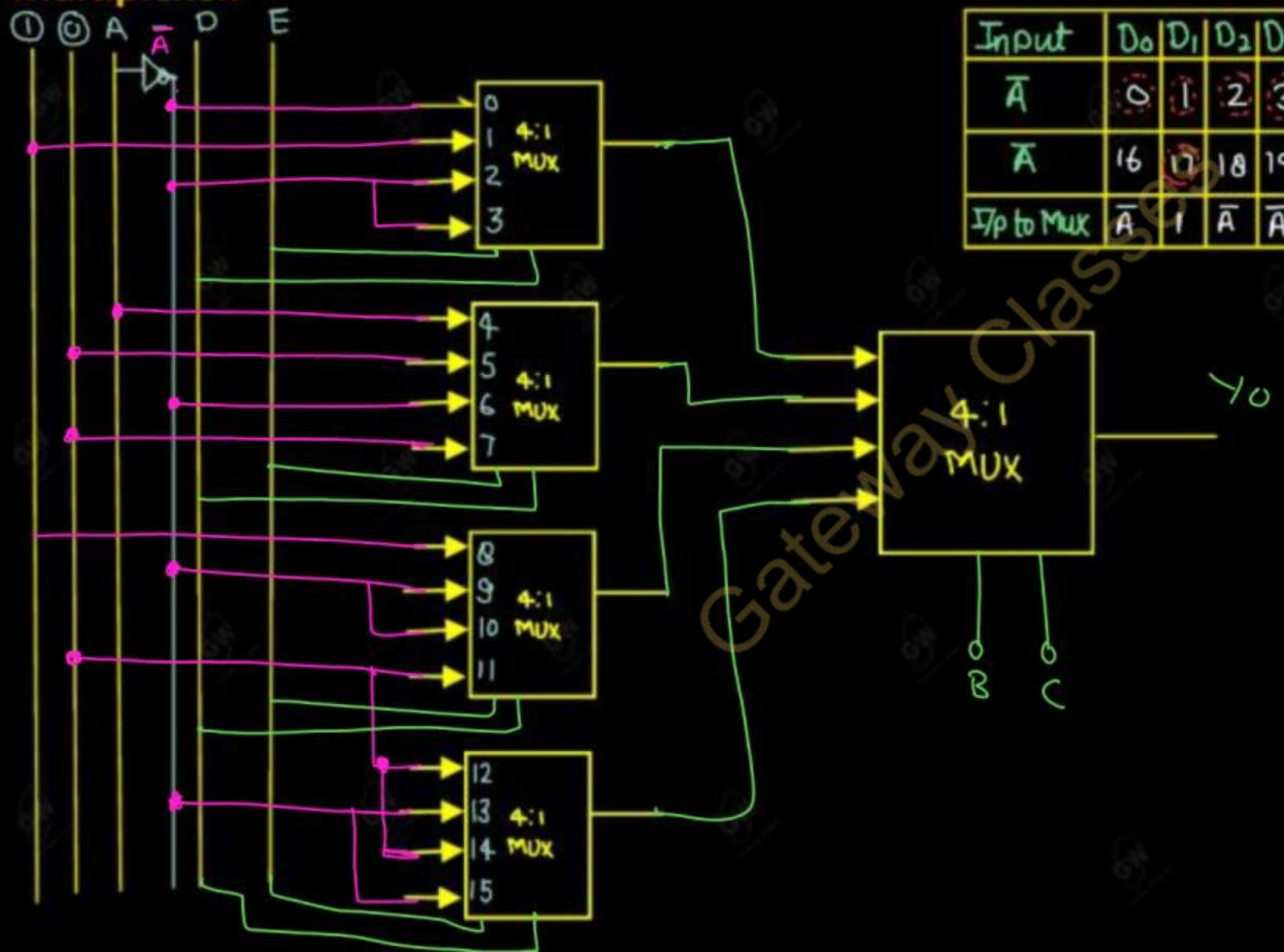
	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$\bar{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	1	1	0	1	$\bar{A}$	0	A	1

Gateway Classes

Q.7 . Implement the function F(A,B,C,D,E) = Σm(0,1,2,3,6,8,9,10,13,15,17,20,24) using 4:1 Multiplexer.

$\frac{16}{4} = 4$, $\frac{4}{4} = 1$

Input	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}
$\bar{A}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\bar{A}$	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
I/p to Mux	$\bar{A}$	1	$\bar{A}$	$\bar{A}$	A	0	$\bar{A}$	0	1	$\bar{A}$	$\bar{A}$	0	0	$\bar{A}$	0	$\bar{A}$



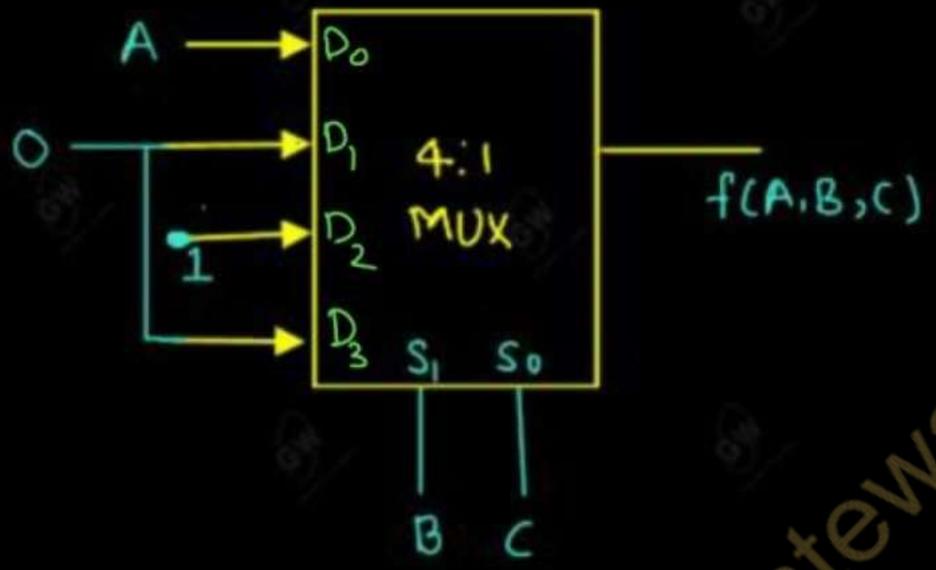
Gateway Classes

Q.8 . Implement the function F(A,B,C) = Π (0,1,3,5,7) using 4:1 Multiplexer.

POS

$(A+B+C)(\bar{A}+\bar{B}+C)$

$F(A,B,C) = \Sigma m(2,4,6)$

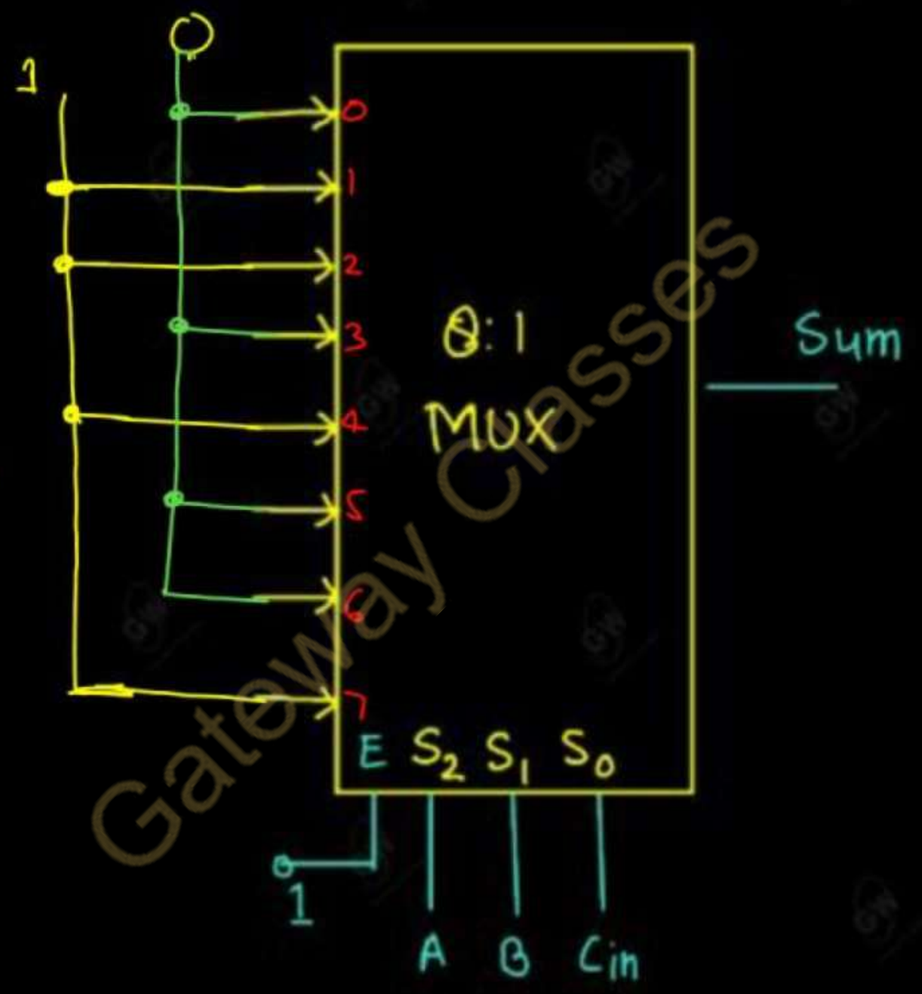


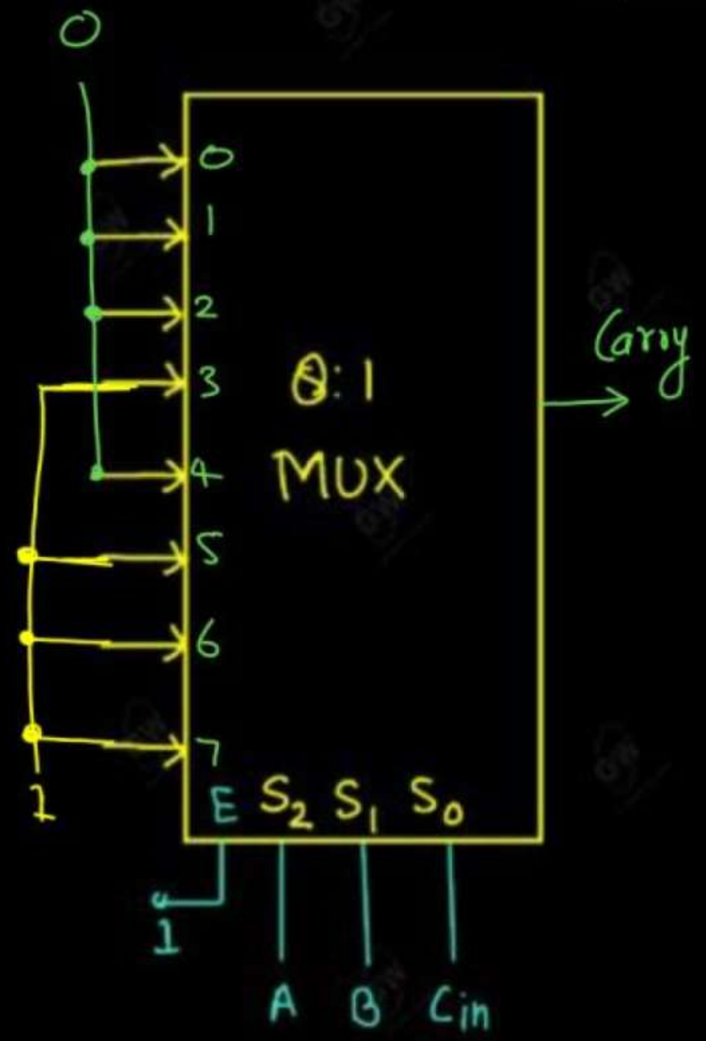
Input	D_0	D_1	D_2	D_3
$\bar{A}$	0	1	2	3
A	4	5	6	7
I/P to MUX	A	0	1	0

Gateway Classes

Q.9 . Implement Full Adder using 8:1 Multiplexer.

INPUTS			OUTPUTS		
A	B	C_{in}	Sum (S)	Carry out (C)	
0	0	0	0	0	D_0
0	0	1	1	0	D_1
0	1	0	1	0	D_2
0	1	1	0	1	D_3
1	0	0	1	0	D_4
1	0	1	0	1	D_5
1	1	0	0	1	D_6
1	1	1	1	1	D_7





Gateway Classes

AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-2 Lec-7



Today's target

- Circuit realization using De Multiplexers
- PYQ
- DPP

Gateway Classes

Avinash Sir

AKTU : *DIGITAL SYSTEM DESIGN (BEC-302)* Syllabus

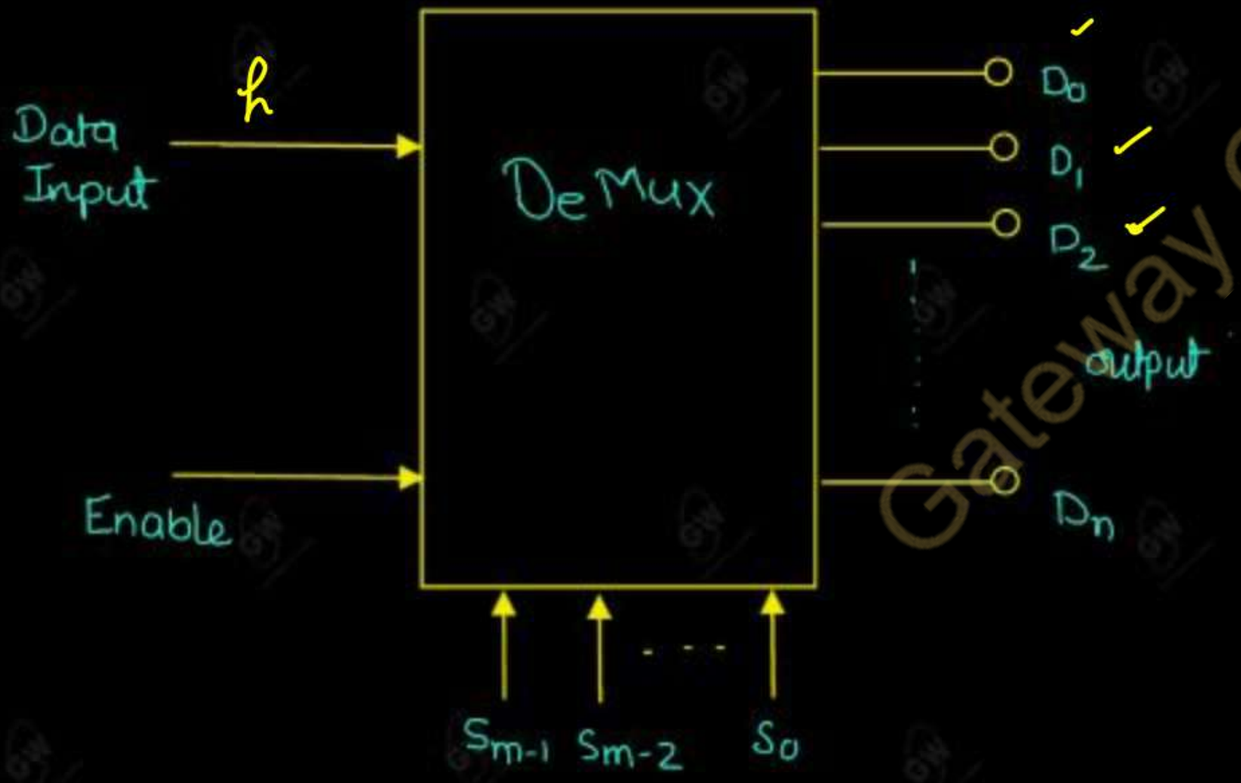
UNIT-2 : Combinational circuits:

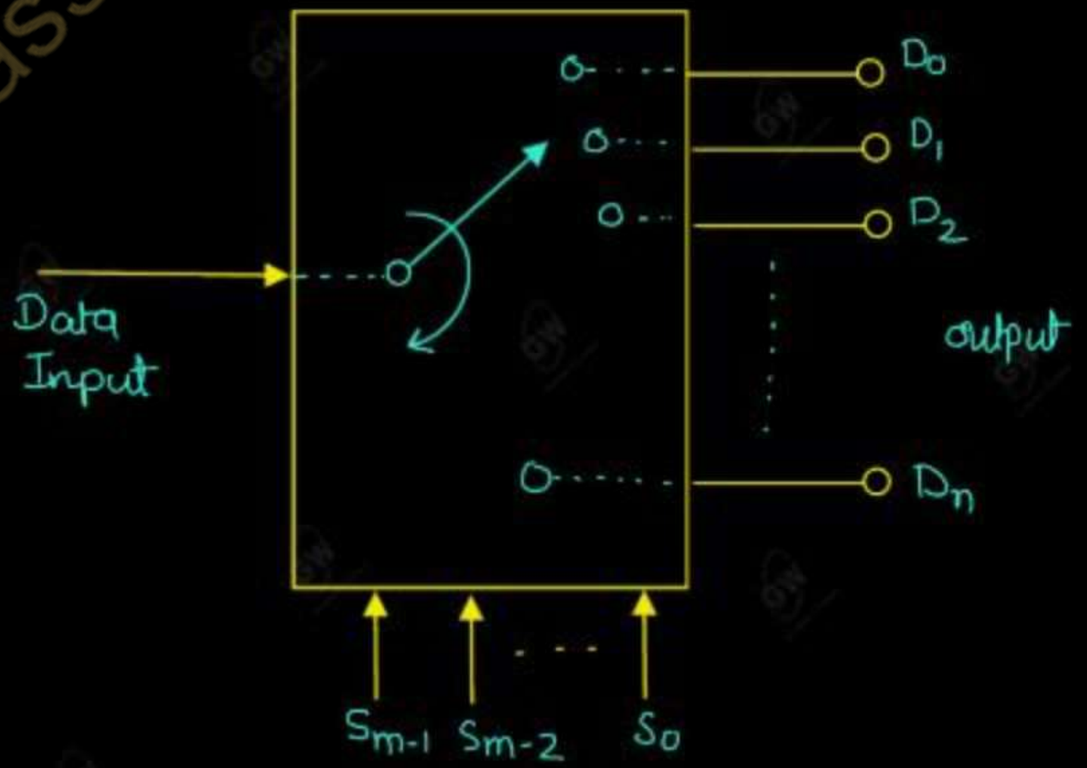
- Analysis and Design of combinational circuits
- MSI devices like comparators, multiplexers, demultiplexers, encoder, decoder,
- circuit realization using Multiplexers and decoders, half and full adders, subtractors, serial and parallel adders, BCD adder, barrel shifter and ALU.

Gateway Classes

Demultiplexer

A demultiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs. It has only one input, n outputs, m select input. At a time only one output line is selected by the select lines and the input is transmitted to the selected output line.

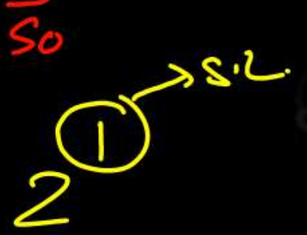




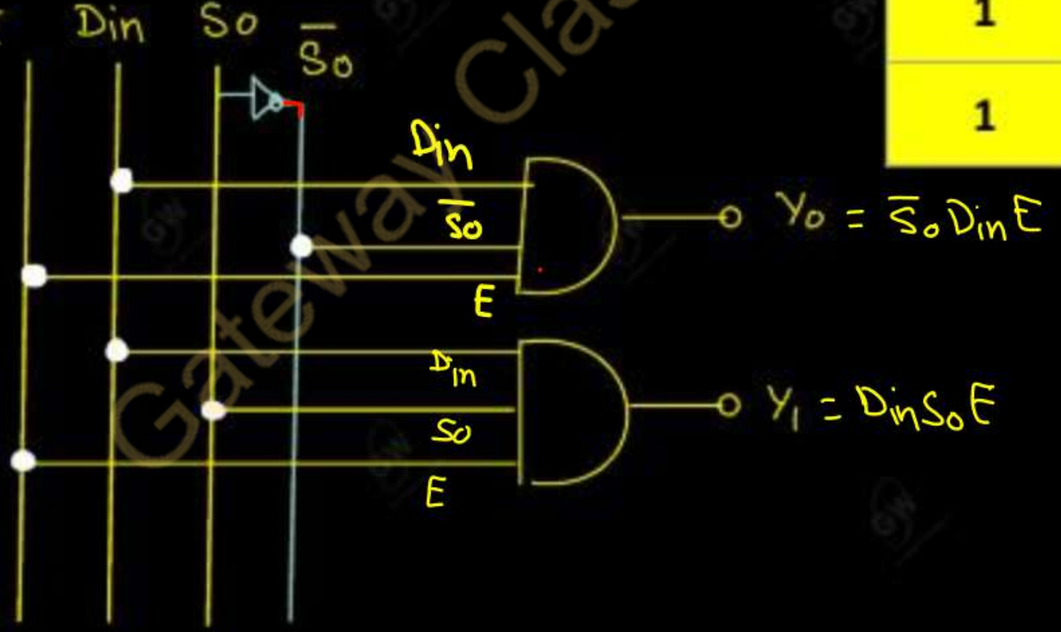
Gateway Classes

1 : 2 Demultiplexer



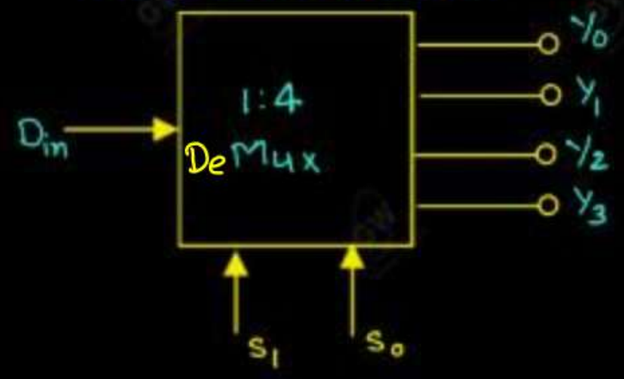


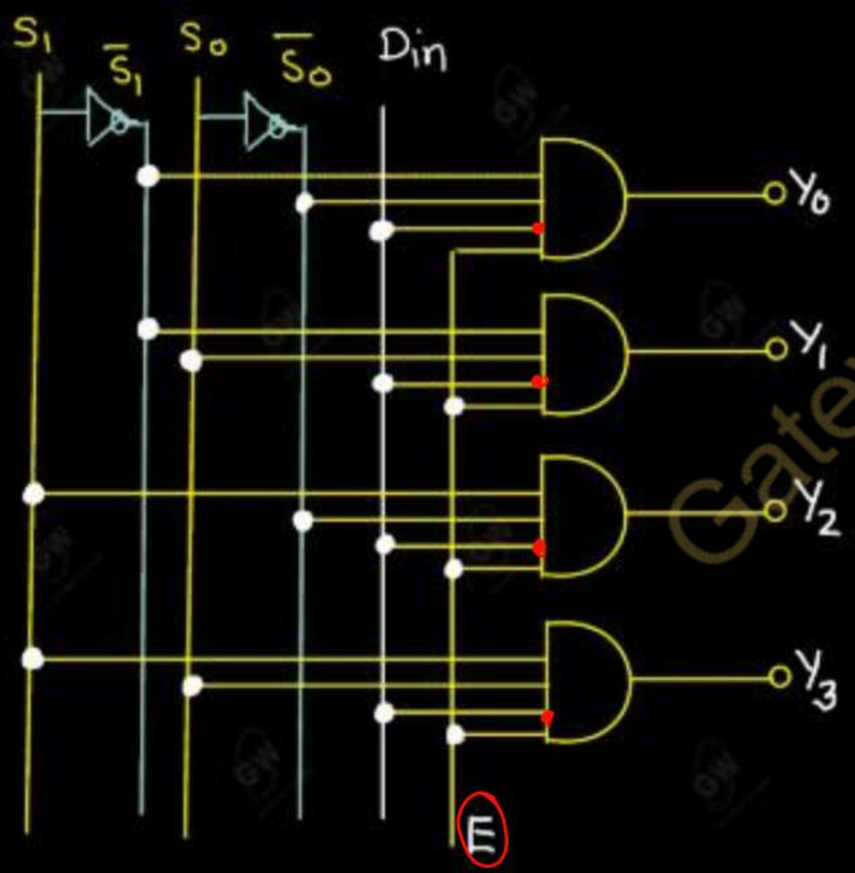
Enable	INPUTS	OUTPUTS	
E	S0	Y1	Y0
0	X	0	0
1	0	0	Din
1	1	Din	0



Gateway Classes

1 : 4 Demultiplexer





Enable	INPUTS	Select Line		Output				
E	Din	S1	S0	Y0	Y1	Y2	Y3	
1	0	0	0	0	0	0	0	$\bar{S}_1\bar{S}_0 D_{in}$ Y_0
1	1	0	0	1	0	0	0	
1	0	0	1	0	0	0	0	$\bar{S}_1 S_0 D_{in}$ Y_1
1	1	0	1	0	1	0	0	
1	0	1	0	0	0	0	0	$S_1\bar{S}_0 D_{in}$ Y_2
1	1	1	0	0	0	1	0	
1	0	1	1	0	0	0	0	$S_1 S_0 D_{in}$ Y_3
1	1	1	1	0	0	0	1	

Gateway Classes

1 : 8 Demultiplexer

2 (3) → S.L.

$Y_0 = \bar{S}_2 \bar{S}_1 \bar{S}_0 Din$

$Y_1 = \bar{S}_2 \bar{S}_1 S_0 Din$

$Y_2 = \bar{S}_2 S_1 \bar{S}_0 Din$

$Y_3 = \bar{S}_2 S_1 S_0 Din$

$Y_4 = S_2 \bar{S}_1 \bar{S}_0 Din$

$Y_5 = S_2 \bar{S}_1 S_0 Din$

$Y_6 = S_2 S_1 \bar{S}_0 Din$

$Y_7 = S_2 S_1 S_0 Din$

En.	I/P	Select		Output								
E	S2	S1	S0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	
0	X	X	X	0	0	0	0	0	0	0	0	
1	0	0	0	D_{in}	0	0	0	0	0	0	0	$\bar{S}_2 \bar{S}_1 \bar{S}_0 D_{in}$
1	0	0	1	0	D_{in}	0	0	0	0	0	0	$\bar{S}_2 \bar{S}_1 S_0 D_{in}$
1	0	1	0	0	0	D_{in}	0	0	0	0	0	$\bar{S}_2 S_1 \bar{S}_0 D_{in}$
1	0	1	1	0	0	0	D_{in}	0	0	0	0	$\bar{S}_2 S_1 S_0 D_{in}$
1	1	0	0	0	0	0	0	D_{in}	0	0	0	$S_2 \bar{S}_1 \bar{S}_0 D_{in}$
1	1	0	1	0	0	0	0	0	D_{in}	0	0	$S_2 \bar{S}_1 S_0 D_{in}$
1	1	1	0	0	0	0	0	0	0	D_{in}	0	$S_2 S_1 \bar{S}_0 D_{in}$
1	1	1	1	0	0	0	0	0	0	0	D_{in}	$S_2 S_1 S_0 D_{in}$

GW GATEWAY CLASSES
Gateway Classes

1 : 8 Demultiplexer

$Y_0 = \bar{S}_2 \bar{S}_1 \bar{S}_0 Din$

$Y_1 = \bar{S}_2 \bar{S}_1 S_0 Din$

$Y_2 = \bar{S}_2 S_1 \bar{S}_0 Din$

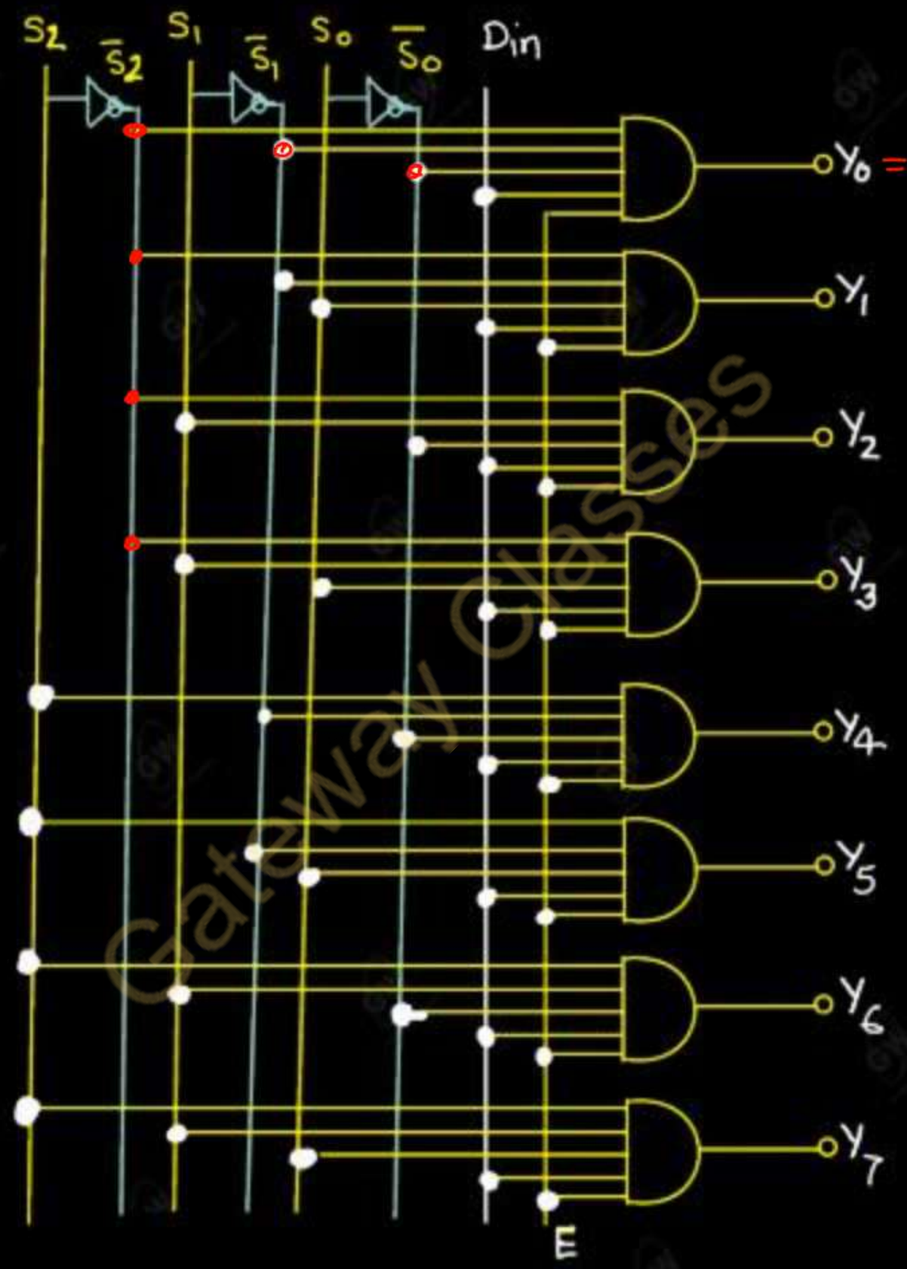
$Y_3 = \bar{S}_2 S_1 S_0 Din$

$Y_4 = S_2 \bar{S}_1 \bar{S}_0 Din$

$Y_5 = S_2 \bar{S}_1 S_0 Din$

$Y_6 = S_2 S_1 \bar{S}_0 Din$

$Y_7 = S_2 S_1 S_0 Din$



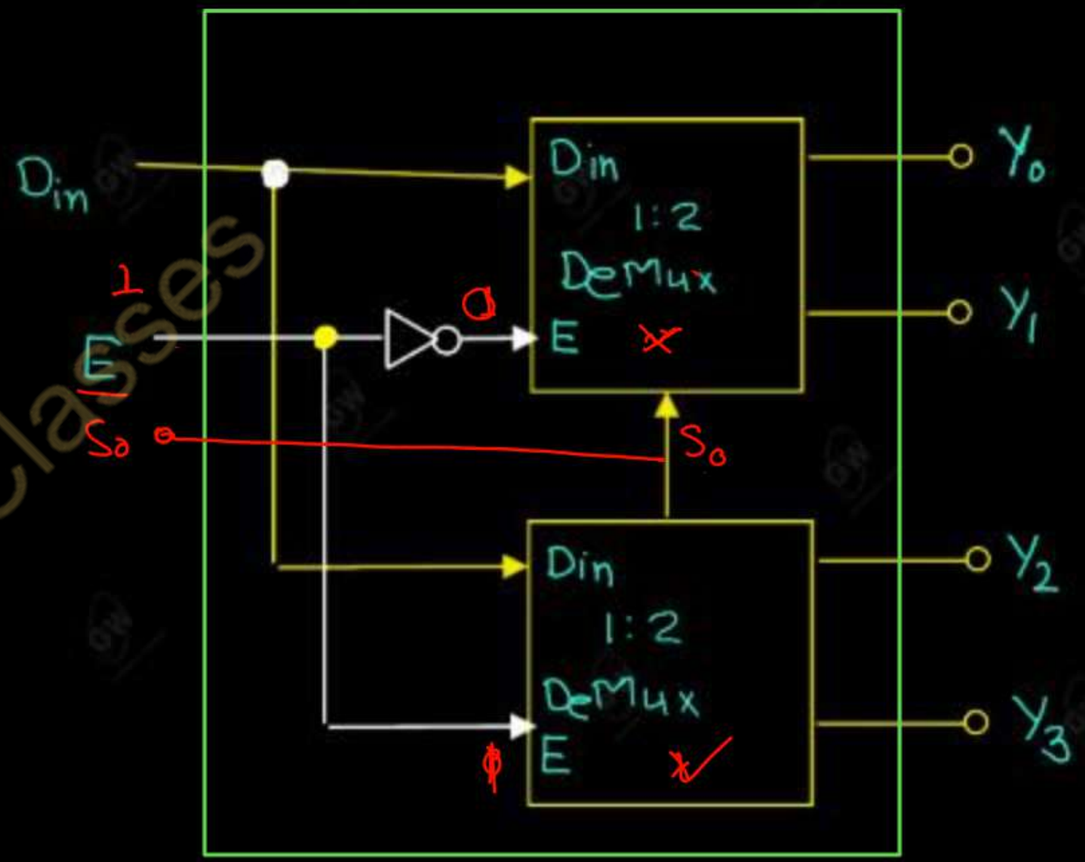
$Y_0 = \bar{S}_2 \bar{S}_1 \bar{S}_0 Din\, E$

Gateway Classes

Q.1 . Design 1: 4 line Demultiplexer using 1: 2 Demux.

Truth Table

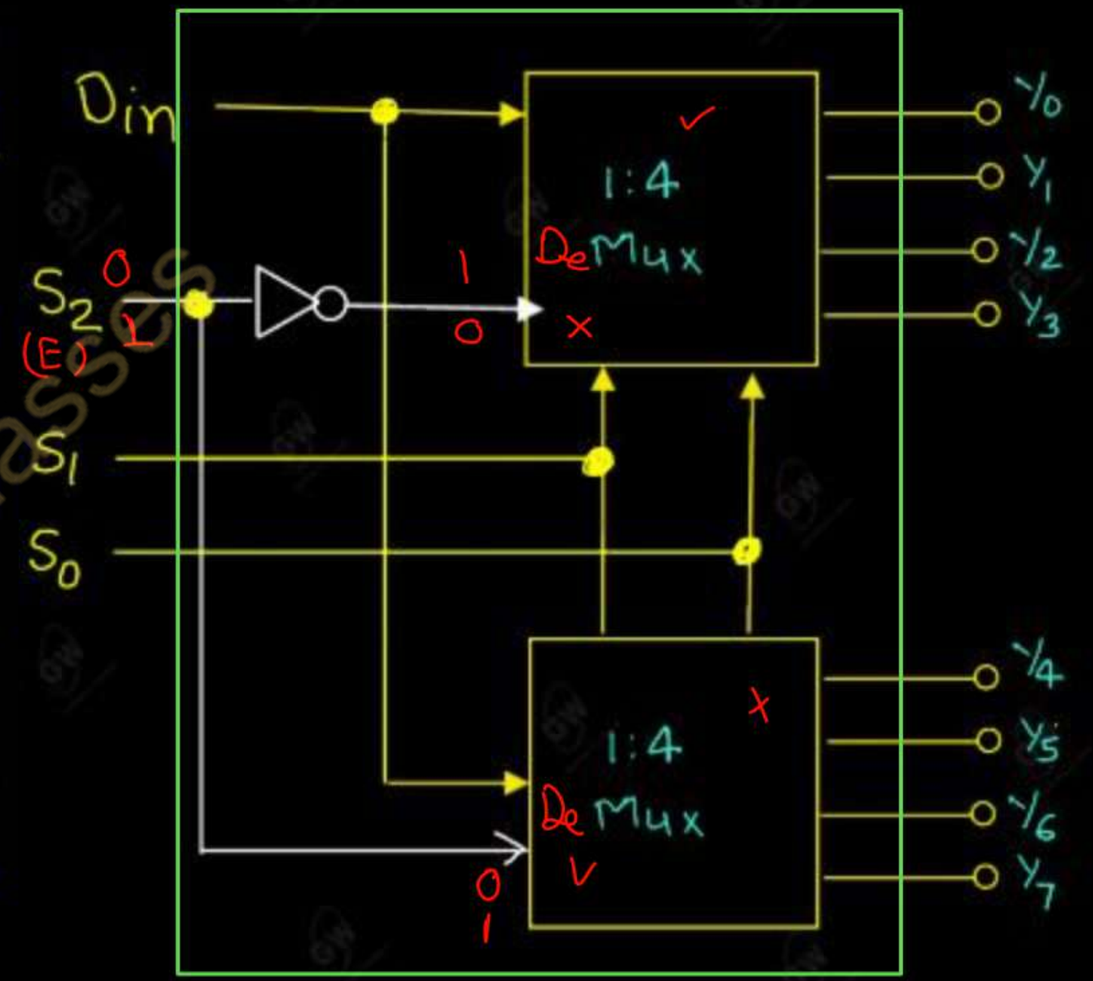
ENABLE	Select I/P		output				
	S_1	S_0	Y_0	Y_1	Y_2	Y_3	
0	0	0	D_{in}	0	0	0	Demux 1 Activated
0	0	1	0	D_{in}	0	0	
1	1	0	0	0	D_{in}	0	Demux 2 Activated
1	1	1	0	0	0	D_{in}	



Gateway Classes

Q.3 . Design 1 : 8 line Demultiplexer using 1: 4 Demux.

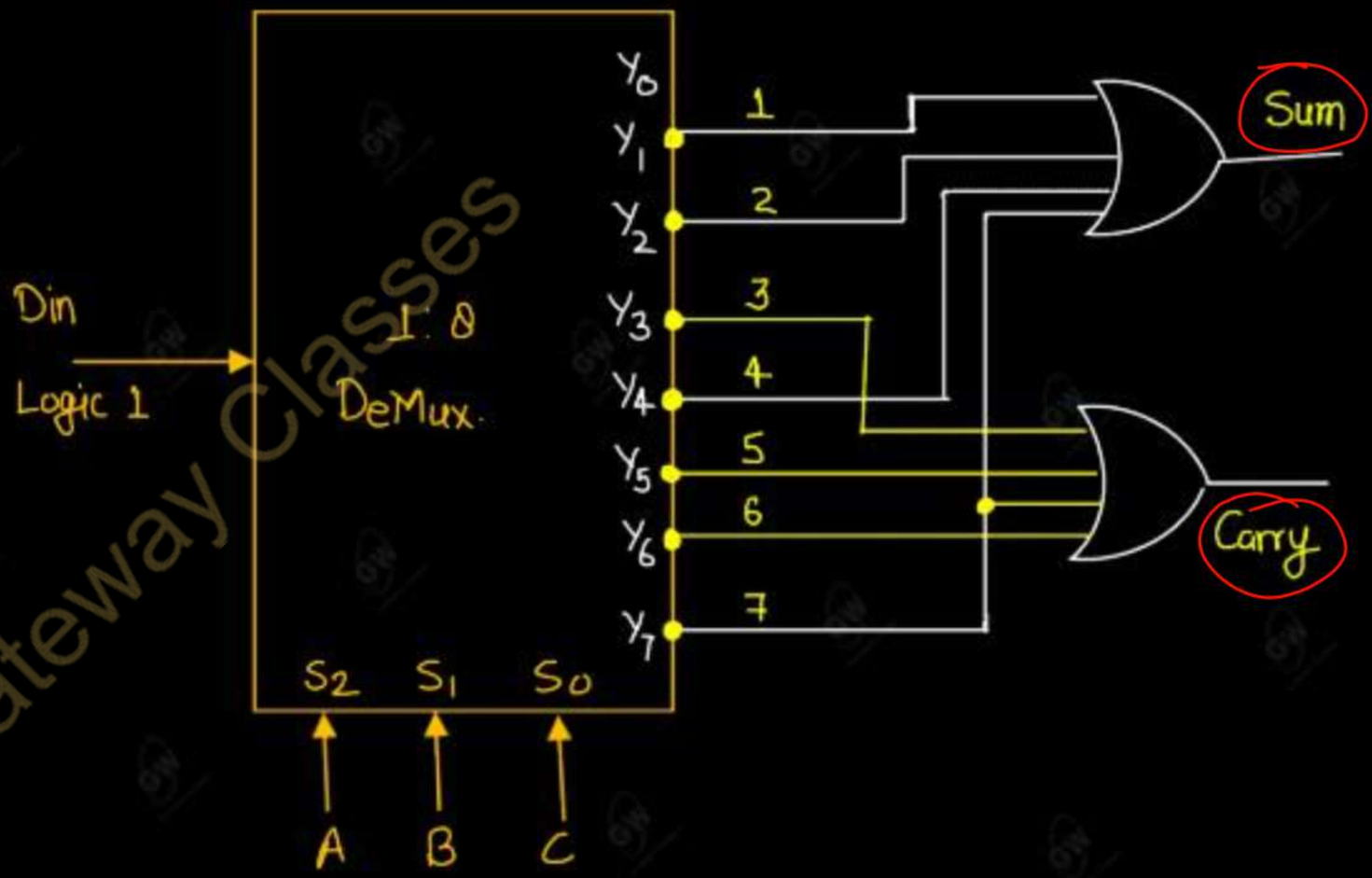
Select Inputs			Outputs							
S_2	S_1	S_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	Din	0	0	0	0	0	0	0
0	0	1	0	Din	0	0	0	0	0	0
0	1	0	0	0	Din	0	0	0	0	0
0	1	1	0	0	0	Din	0	0	0	0
1	0	0	0	0	0	0	Din	0	0	0
1	0	1	0	0	0	0	0	Din	0	0
1	1	0	0	0	0	0	0	0	Din	0
1	1	1	0	0	0	0	0	0	0	Din



Gateway Classes

Q.4 . Implement a full adder using Demultiplexer.

	INPUTS			OUTPUTS	
	A	B	C_{in}	Sum (S)	Carry out (C)
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1



$Sum = \Sigma m(1,2,4,7)$

$Carry = \Sigma m(3,5,6,7)$

Gateway Classes

DPP

Q.1 . Design 1: 32 line Demultiplexer using 1: 4 Demultiplexer.

Q.2 Implement the following expression using a multiplexer:

$F(A,B,C) = \Sigma m\ (0,2,3,6,8,\ 9\ ,\ 11,12,\ 14)$

Q.3 Implement the following function using a 4 to 16 line decoder:

$F = \Sigma m\ (1,2,4,7,8,\ 11,12,\ 13)$

Gateway Classes

AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-2 Lec-8



Today's target

- Barrel Shifter
- ALU
- PYQ

Gateway Classes

Avinash Sir

UNIT-2 : Combinational circuits:

- Analysis and Design of combinational circuits
- MSI devices like comparators, multiplexers, demultiplexers, encoder, decoder,
- circuit realization using Multiplexers and decoders, half and full adders, subtractors, serial and parallel adders, BCD adder, barrel shifter and ALU.

Gateway Classes

Barrel Shifter

A barrel shifter is a combinational circuit that can shift or rotate a binary number by a specific number of bits in a single operation.

Unlike normal shifters (which shift bit by bit), barrel shifters perform the shift in parallel, making them very fast.

Features

- Can shift data left or right.
- Can perform both logical and circular (rotational) shifts.
- Controlled by select lines which decide how many positions to shift.
- Used in multipliers, dividers, floating-point arithmetic, and processors.

Gateway Classes

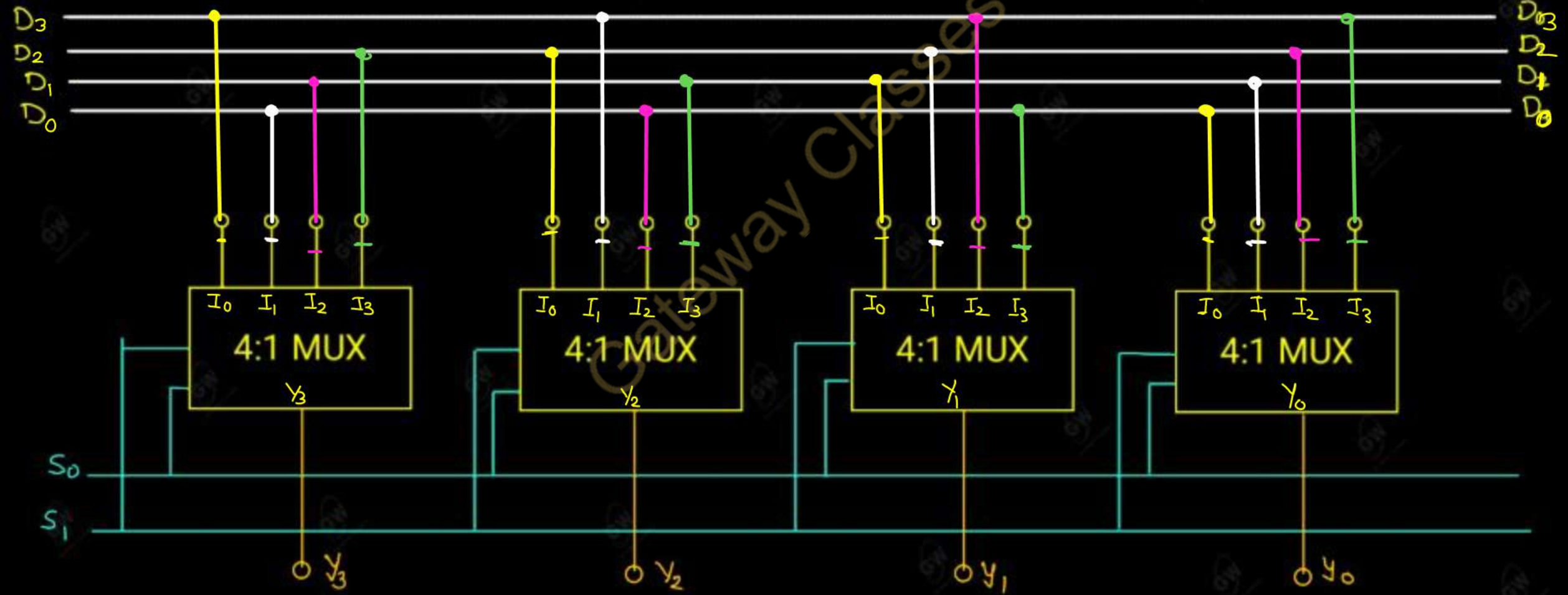
4 bit Barrel Shifter :-

Input: 4-bit number D3 D2 D1 D0

Control: 2 select lines S1 S0 → decides shift amount

Output: 4-bit shifted number

S_0	S_1	Y_3	Y_2	Y_1	Y_0
0	0	D_3	D_2	D_1	D_0
0	1	D_0	D_3	D_2	D_1
1	0	D_1	D_0	D_3	D_2
1	1	D_2	D_1	D_0	D_3



Gateway Classes

ALU

An Arithmetic Logic Unit (ALU) circuit element is a combinational digital electronic circuit that performs arithmetic and bitwise logic operations on integer binary numbers.



ALU plays an important role as it executes and processes all the instructions, performs calculations, manipulates binary data, and performs various decision-making operations.

Gateway Classes

Features of ALU

- Can do arithmetic operations: addition, subtraction, (sometimes multiplication, division)
- do logical operations: AND, OR, NOT, XOR, XNOR, etc.
- Contains two sub-units:
    - AU = Arithmetic Unit
    - LU = Logic Unit
- Receives control signals guiding which operation to perform
- After operation, result is sent to memory or output unit

Functions of ALU

- Arithmetic operations: addition, subtraction, multiplication, division
- Logical operations: AND, OR, NOT, etc.
- Comparison operations: equal to, less than, greater than, etc.
- Shift operations: bit shifts (left shift, right shift) to manipulate data at bit level

Gateway Classes

Main Components / Sub-Units

Arithmetic Unit (AU)

- **Adder:** adds two binary numbers (half-adder, full-adder, ripple carry adder, etc.)
- **Subtractor:** often implemented using 2's complement method to subtract binary numbers
- **Multiplier & Divider:** in more advanced ALUs, dedicated circuits for multiply/divide

Logic Unit (LU)

- Uses logic gates: AND, OR, NOT, NAND, NOR, XOR, XNOR
- Can be composed in combinations to carry out complex logic operations
- Some gates have special uses:
    - XOR = "inequality detector" (gives 1 if inputs differ)
    - XNOR = "equality detector" (gives 1 if inputs are same)

Gateway Classes

How ALU Works (Working Principle)

Inputs & control signals

- Receives binary data inputs
- Receives control signals (from control unit) that tell which operation to do

Operation selection & execution

- Based on control signals, ALU picks which sub-unit to use (arithmetic, logic, shift, etc.)
- Performs the operation

Output result

- Sends result to memory or output units

Gateway Classes

Gateway Classes

Empowering Learners, Transforming Futures

GATEWAY CLASSES

Thank You

Gateway Classes

GET IT ON Google Play

Download on the App Store

Helpline No- 7819 0058 53