AKTU

GATEWAY CLASSES

Digital System Design

B.Tech. 3rd Sem
EC & Allied

Exam/Result Oriented Content

- ✓ Recorded Video Lectures
- ✓ Pdf Notes
- ✓ PYQs

DIGITAL SYSTEM DESIGN

UNIT 2 : Combinational circuits

Today's Target

- Combinational circuits: steps for designing
- Half Adder
- Full Adder
- DPP
- University Questions

Combinational circuits: steps for designing

- Study required conditions
- Prepare Truth Table
- Identify Minterms and place in K-map
- Generate a reduced boolean expression.
- Realize using basic gates/ NAND gates (as required)

Example: Realize a circuit with four inputs (A,B,C,D) that provide output as logic 1 for all combination of inputs which are multiples of 3

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

$$Y = \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}D + AB\bar{C}\bar{D} + ABCD$$

$$= \bar{B}D(\bar{A}C + A\bar{C}) + B\bar{D}(\bar{A}C + A\bar{C}) + ABCD$$

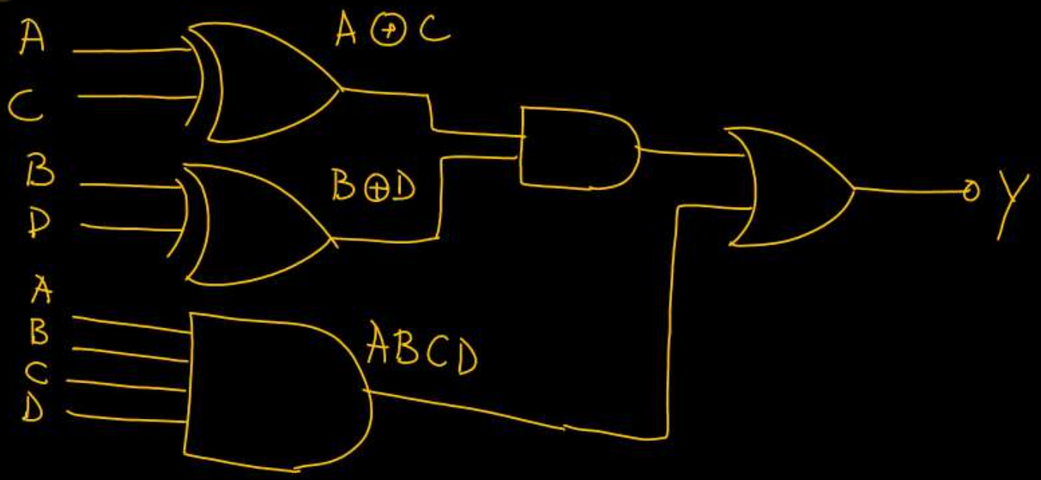
$$= \bar{B}D(A \oplus C) + B\bar{D}(A \oplus C) + ABCD$$

$$= (A \oplus C)(\bar{B}D + B\bar{D}) + ABCD$$

$$= (A \oplus C)(B \oplus D) + ABCD$$

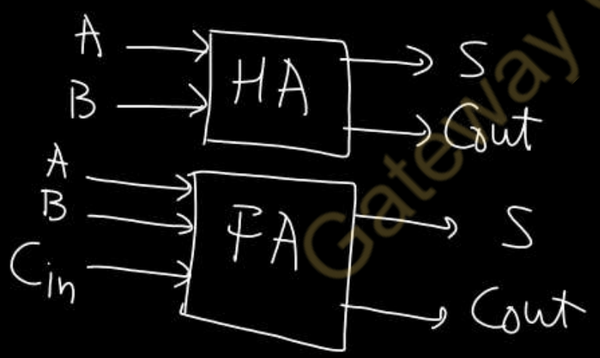
$$Y = \Sigma m(3, 6, 9, 12, 15)$$

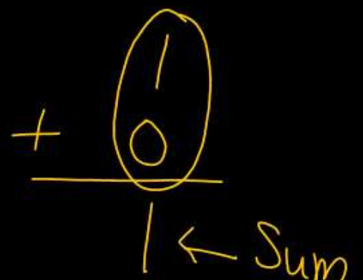
CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

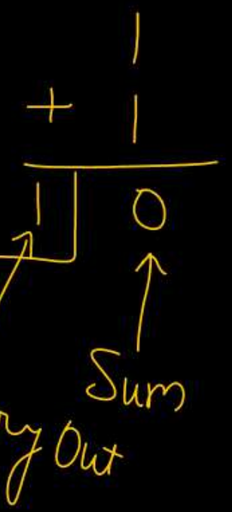


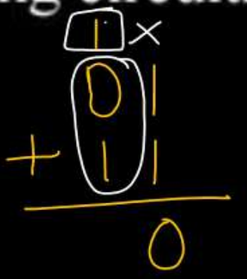
Binary Adder

- Binary Adders are arithmetic circuits in the form of half-adders and full-adders used to add together two binary digits.
- The addition of these two digits produces an output called the SUM of the addition and a second output called the CARRY or Carry-out, (C_{OUT}) bit according to the rules for binary addition.
- One of the main uses for the *Binary Adder* is in arithmetic and counting circuits.
- Binary Addition of Two Bits:
  - $0 + 0 = 0$
  - $0 + 1 = 1$
  - $1 + 0 = 1$
  - $1 + 1 = 0$ (carry = 1)









Binary Adder Block Diagram

A
+ B

Cout ← Sum

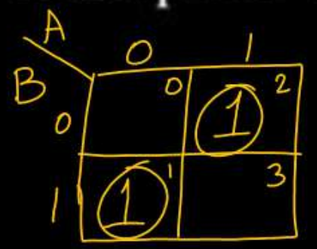
Half Adder Truth Table with Carry-Out

A	B	S	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = \Sigma m(1, 2)$

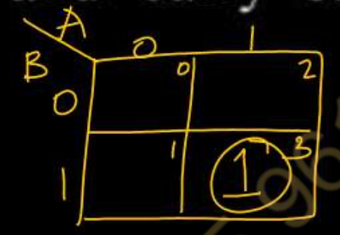
$Cout = \Sigma m(3)$

K-map for 'sum' and 'carry' & implementation using basic gates

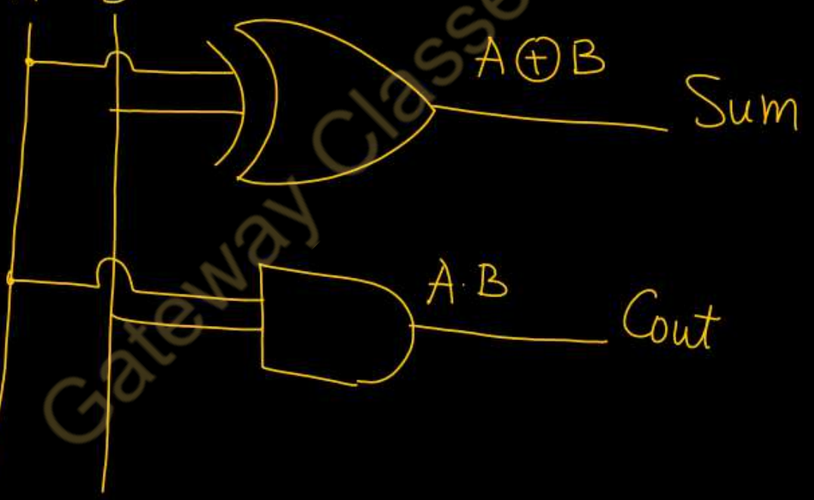


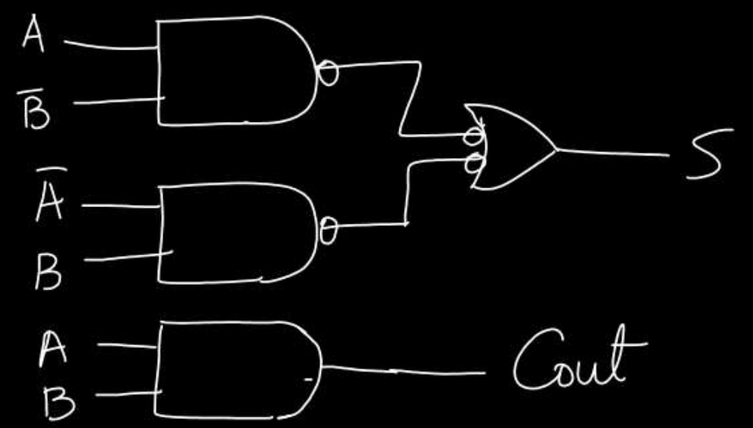
$S = A\bar{B} + \bar{A}B$ ✓

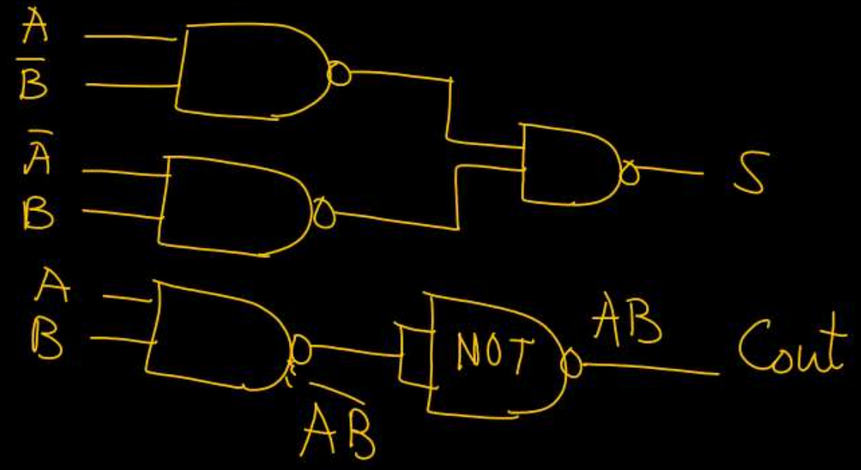
$= A \oplus B$ ×



$Cout = AB$



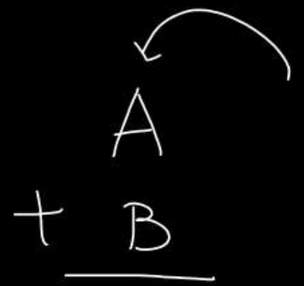




Disadvantage of the *Half Adder* circuit

- when used as a binary adder, is that there is no provision for a "Carry-in" from the previous circuit

- multiple bits of data cannot be added.



A Full Adder Circuit

- a full adder has three inputs.
- 2 bits are single bit data inputs A and B
- plus an additional *Carry-in* (C-in) input to receive the carry from a previous stage.

Cin
A
+ B

$\text{Cout} \leftarrow \overline{\text{SUM}}$

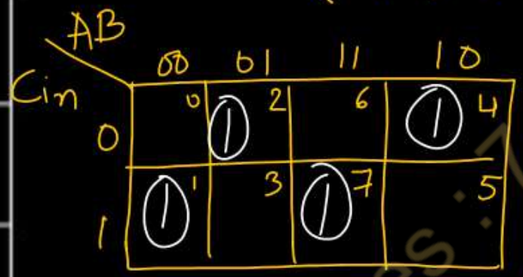
Full Adder Truth Table with Carry-Out

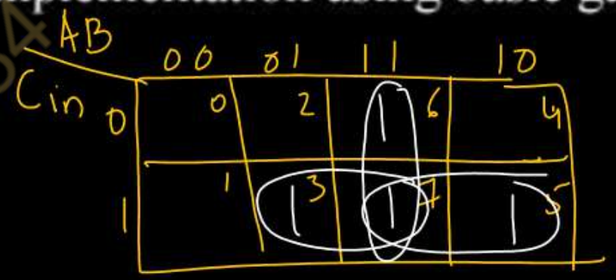
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

K-map for 'sum' and 'carry' & implementation using basic gates

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

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





$Cout = BCin + AB + A\,Cin$

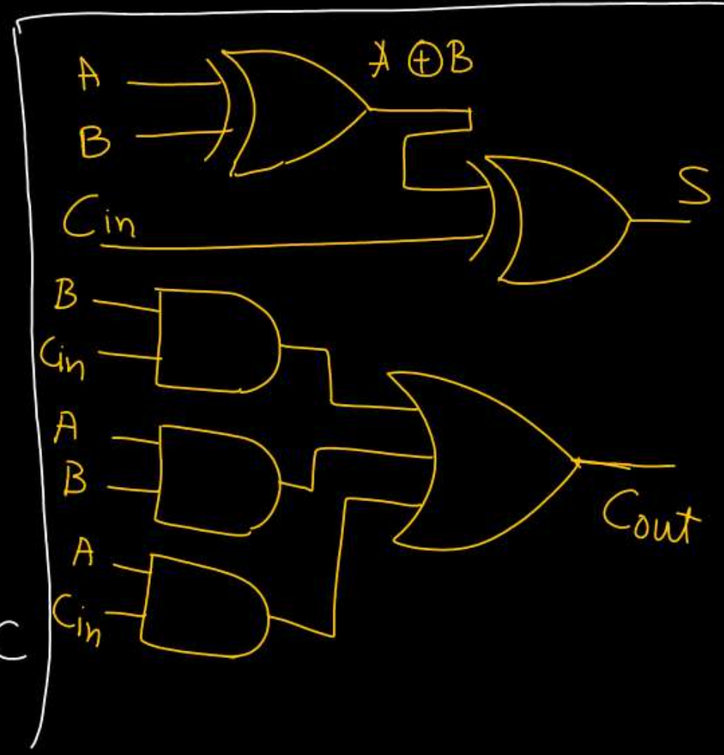
$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

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

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

$B \oplus C = X$ So $\overline{B \oplus C} = \bar{X}$

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



DPP

a) Design Half Adder circuit using NAND gates only.

b) Design Full Adder circuit using NAND gates only.

UNIVERSITY QUESTIONS

YEAR	QUESTION	MARKS
21-22	Design a full adder by constructing the truth table and simplify the output equations.	10 marks
20-21	Construct full adder using logic gates.	2 marks

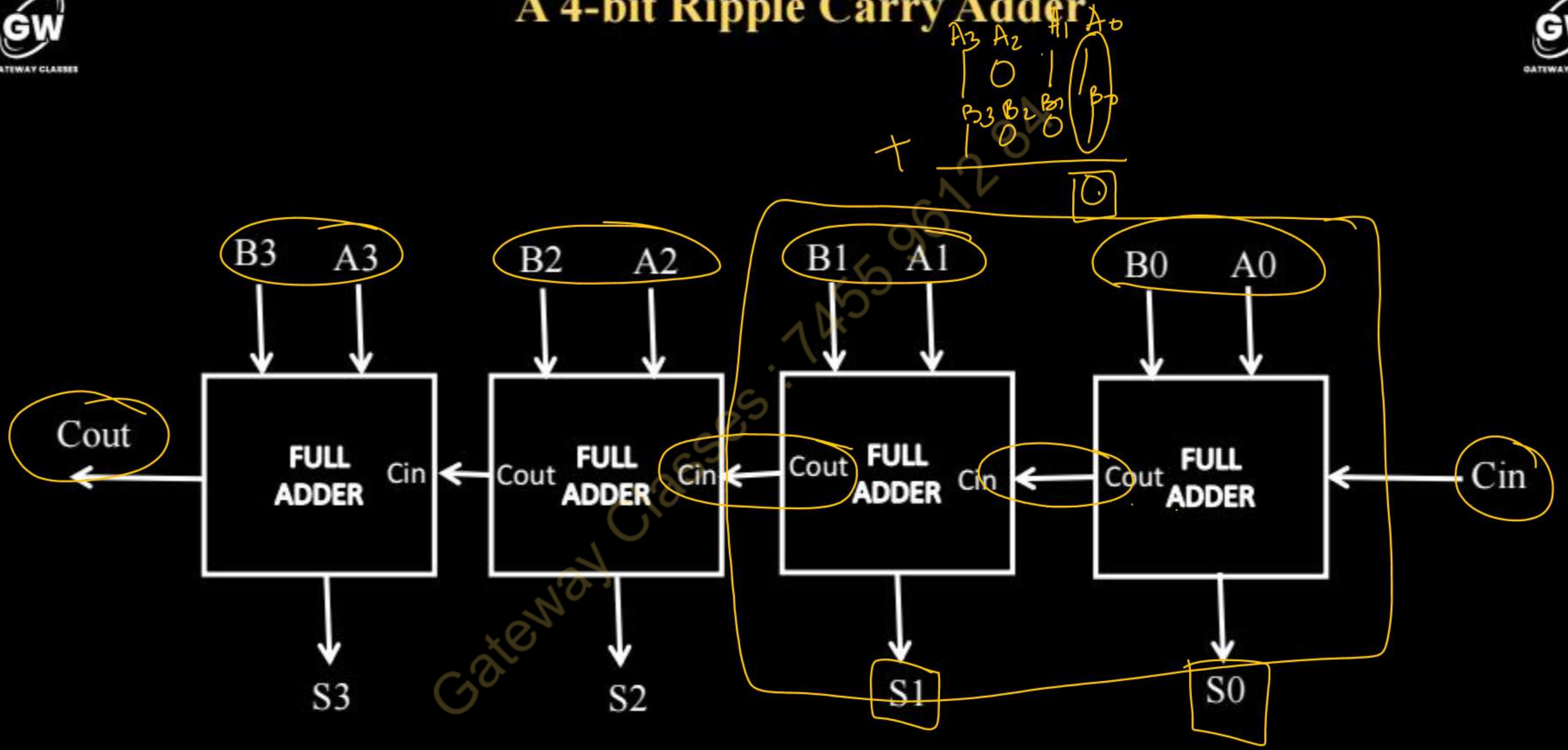
DIGITAL SYSTEM DESIGN

UNIT 2 : Combinational circuits

Today's Target

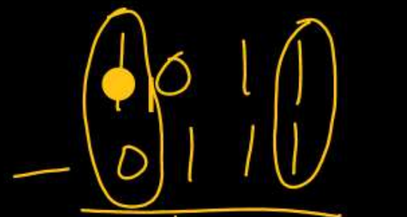
- A 4-bit Ripple Carry Adder
- Binary Subtractor
- Difference between Serial Adder and Parallel Adder
- Carry Look Ahead Adder
- DPP
- University Questions

A 4-bit Ripple Carry Adder



Binary Subtractor

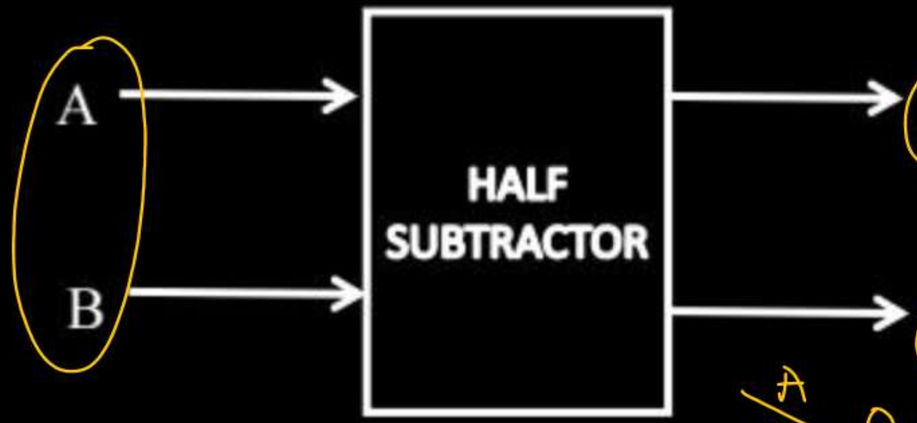
- produces an output which is the subtraction of two binary numbers.
- The binary subtractor produces a DIFFERENCE, D
- use a BORROW bit, B from the previous column.
- Binary subtraction Rules:
- 0 – 0 = 0
- 1 – 0 = 1
- 1 – 1 = 0
- 0 – 1 = 1, borrow = 1



Bin

A Half Subtractor Circuit

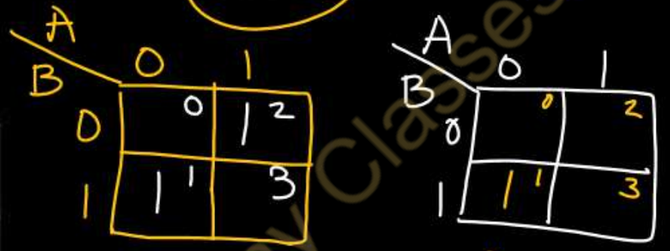
- A half subtractor is a logical circuit that performs a subtraction operation on two binary digits.
- The half subtractor produces a sum and a borrow bit for the next stage.



$D = \Sigma m(1, 2)$

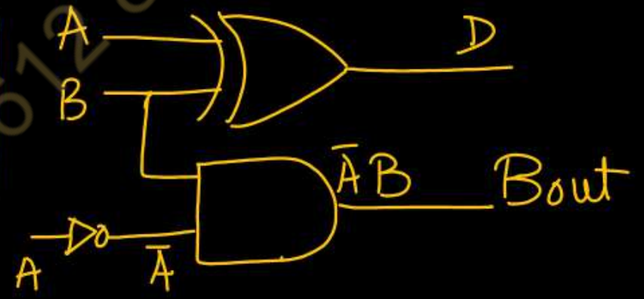
$Bout = \Sigma m(1)$

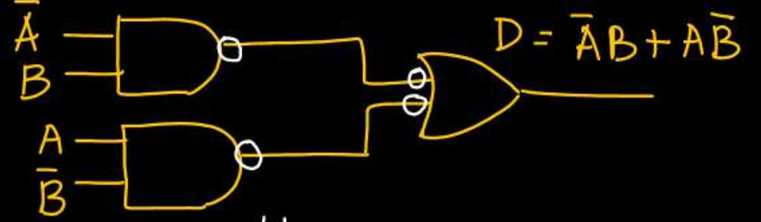
A	B	D	Bout
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

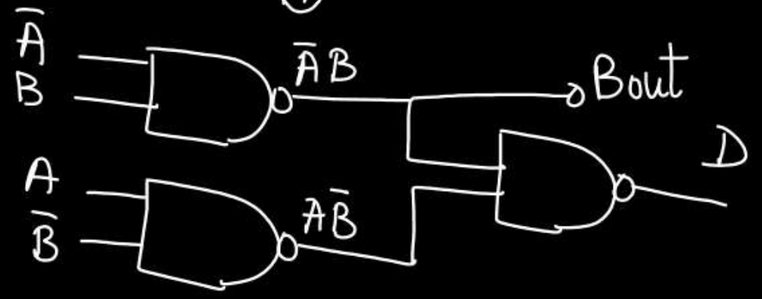


for $D = \bar{A}B + A\bar{B}$ ✓ for $B_{out} = \bar{A}B$

$= A \oplus B$ ×







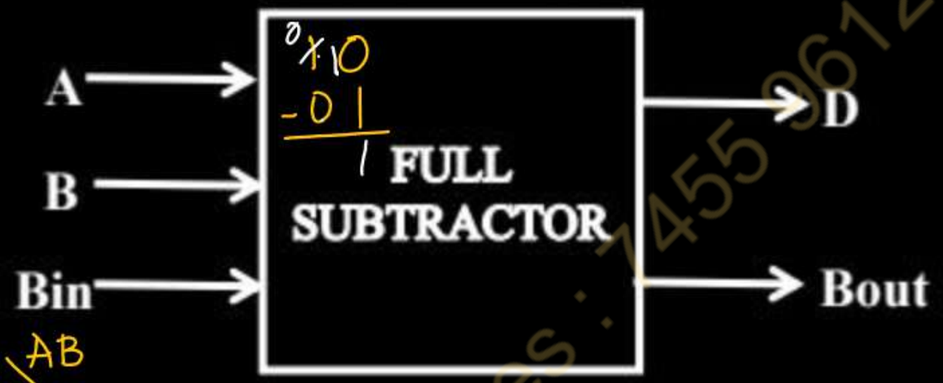
A Full Binary Subtractor Circuit

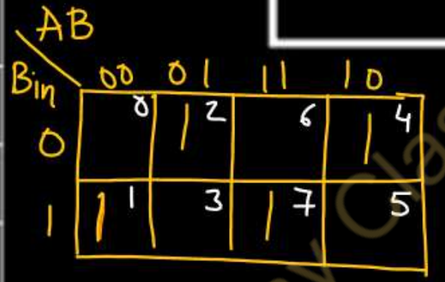
- a full subtractor has three inputs.
- The two single bit data inputs X (minuend) and Y (subtrahend)
- an additional *Borrow-in* (B-in) input to receive the borrow generated from a previous stage.

A	B	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

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

$Bout = \Sigma m(1,2,3,7)$





$$D = \bar{A}\bar{B}Bin + \bar{A}B\bar{Bin} + A\bar{B}\bar{Bin} + ABBin$$
$$= \bar{A}(\bar{B}Bin + B\bar{Bin}) + A(\bar{B}\bar{Bin} + BBin)$$
$$= \bar{A}(B \oplus Bin) + A\overline{(B \oplus Bin)}$$

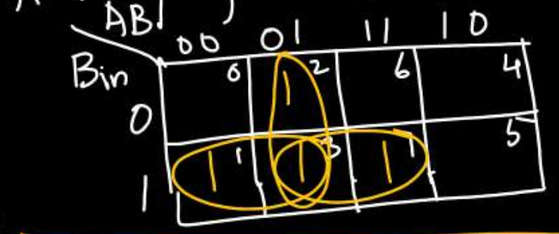
If $B \oplus Bin = X$

$D = \bar{A}X + A\bar{X}$

$= A \oplus X$

$$D = A \oplus B \oplus Bin$$

K-Map for Bout

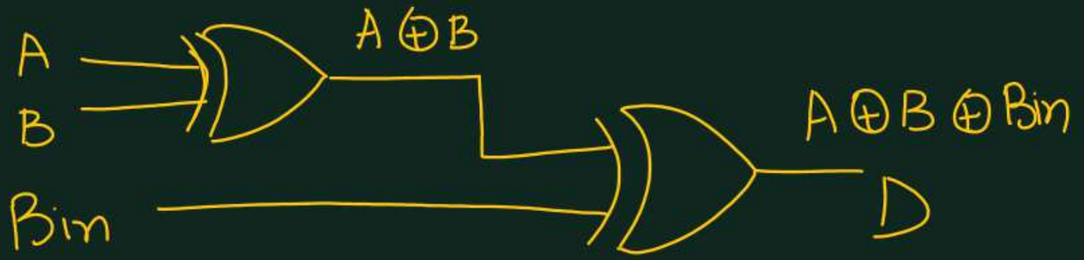


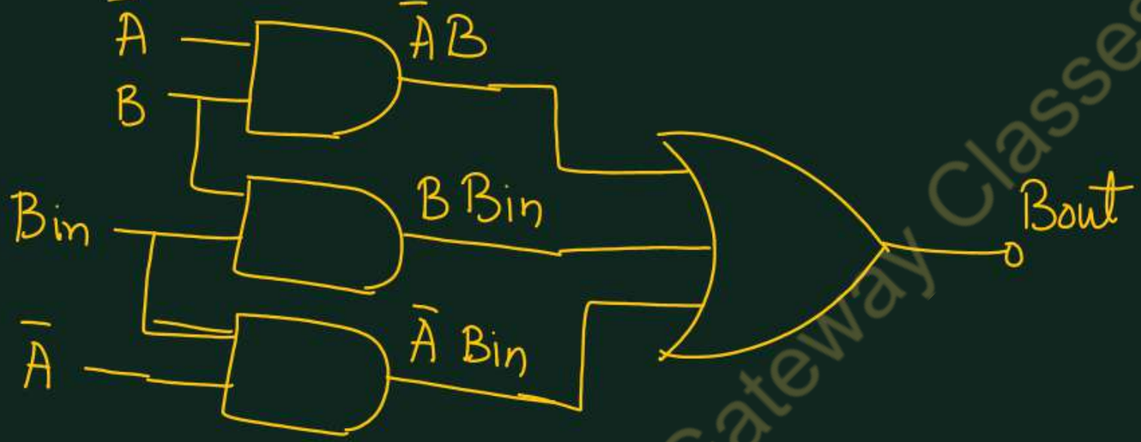
$$Bout = \bar{A}Bin + \bar{A}B + BBin$$

Using Logic Gates

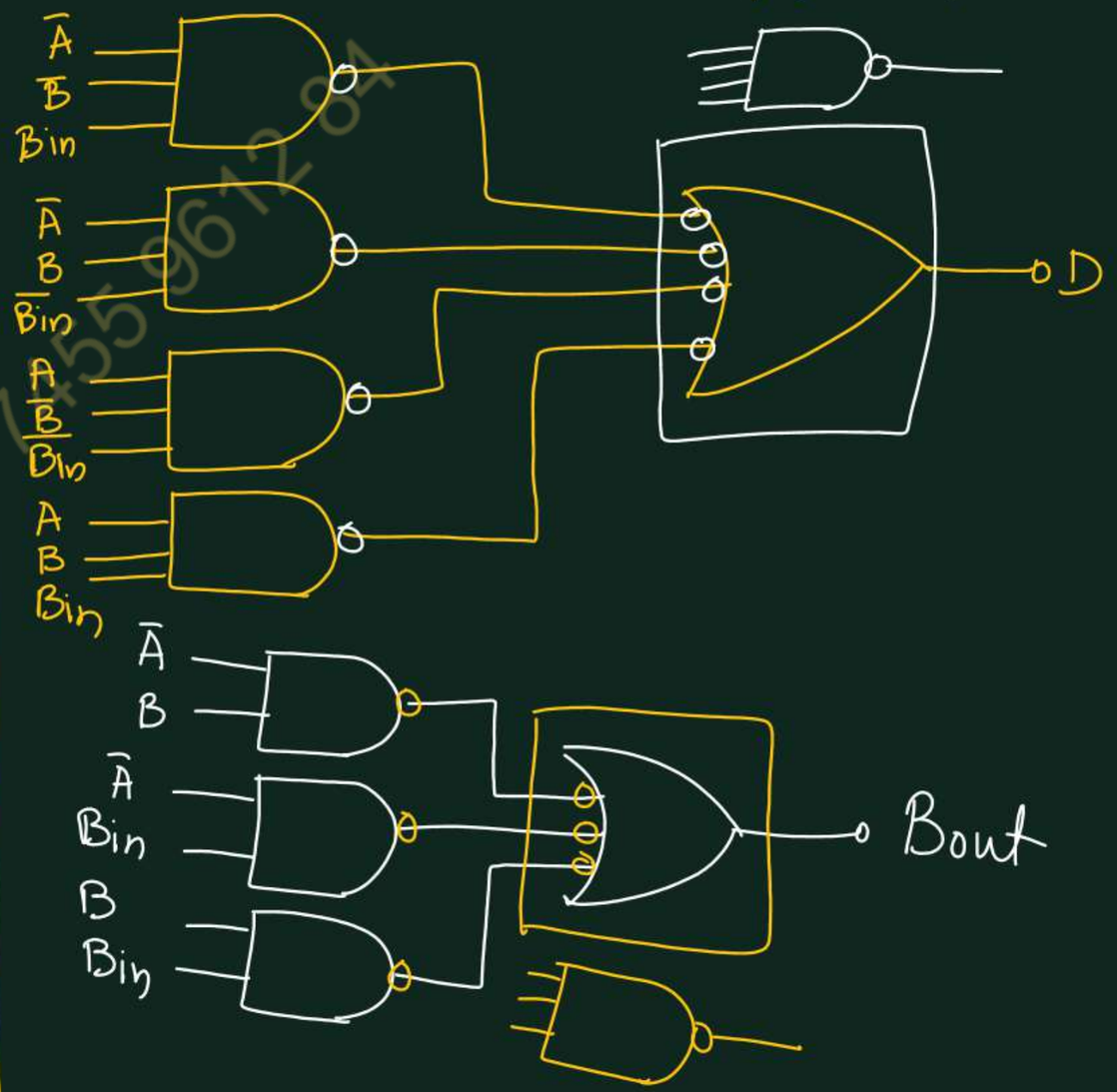
$D = A \oplus B \oplus Bin$

$Bout = \bar{A}B + \bar{A}Bin + BBin$



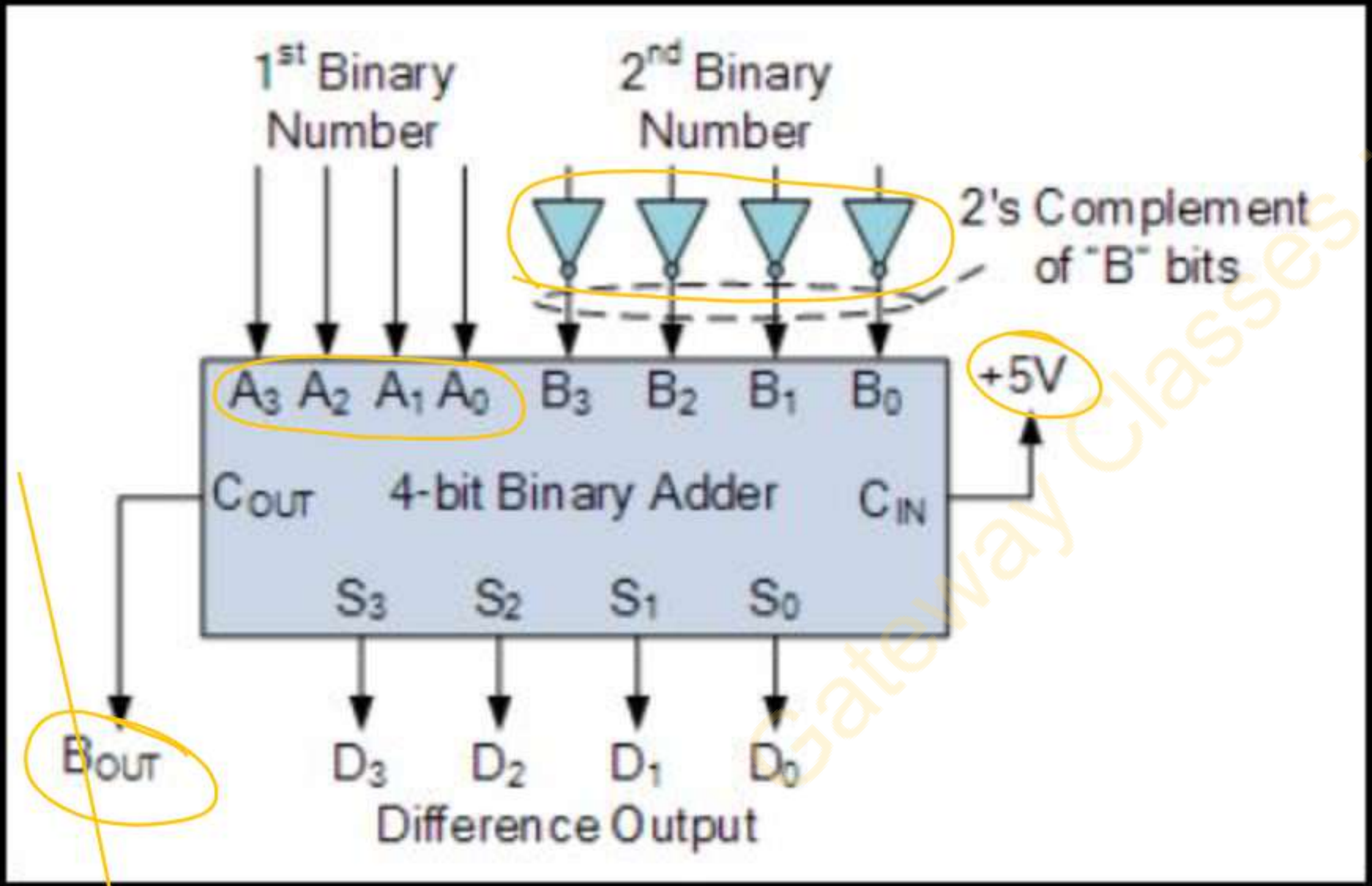


$D = \bar{A}\bar{B}Bin + \bar{A}B\bar{Bin} + A\bar{B}\bar{Bin} + ABBin$



An n-bit Binary Subtractor

- It uses an n-bit adder and n number of inverters (NOT Gates)
- the process of subtraction becomes an addition as we can use two's complement notation on all the bits in the subtrahend and setting the carry input of the least significant bit to a logic "1" (HIGH).



1011
− 0101

I : 1011
2's of II : + 1011
1 0110

1's + 1 = 2's

0101
1's 1010
+ 1
1011

Difference between Serial Adder and Parallel Adder:

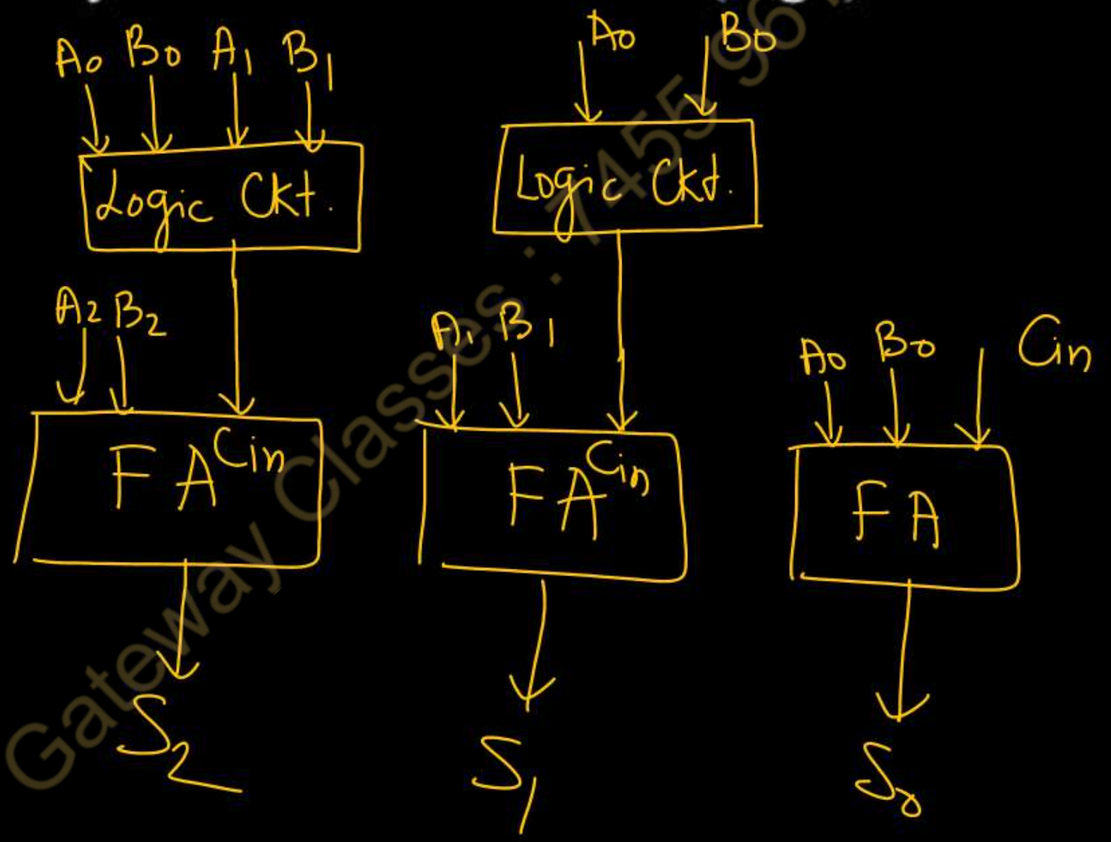
SERIAL ADDER	PARALLEL ADDER
It is used to add two binary numbers in serial form.	It is used to add two binary numbers in parallel form.
A serial adder uses shift registers.	A parallel adder uses registers with parallel loads.
It requires single full adder.	It requires multiple full adders
Carry flip-flop is used in serial adder.	Ripple carry adder is used in parallel adder.
Serial adder is a sequential circuit.	Parallel adder is a combinational circuit.
In serial adder, propagation delay is less.	In parallel adder, propagation delay is present from input carry to output carry.
Number of required full adder is fixed i.e. one.	Number of required full adder is equal to the number of bits in the binary number.

Difference between "Ripple Carry Adder" and Carry Look-ahead Generator

- In a ripple carry adder, the carry signal has to travel through each adder module to get to the end, causing long propagation delays.
- In a carry look-ahead adder, the carry logic is handled separately with shorter delays.
- A carry look-ahead adder reduces the propagation delay by introducing more complex hardware.

Carry Look Ahead Adder

- Carry Look Ahead Adder is an improved version of the ripple carry adder.
- It generates the carry-in of each full adder simultaneously without causing any delay.
- The time complexity of carry look ahead adder = Θ (logn).



DPP

GW GATEWAY CLASSES

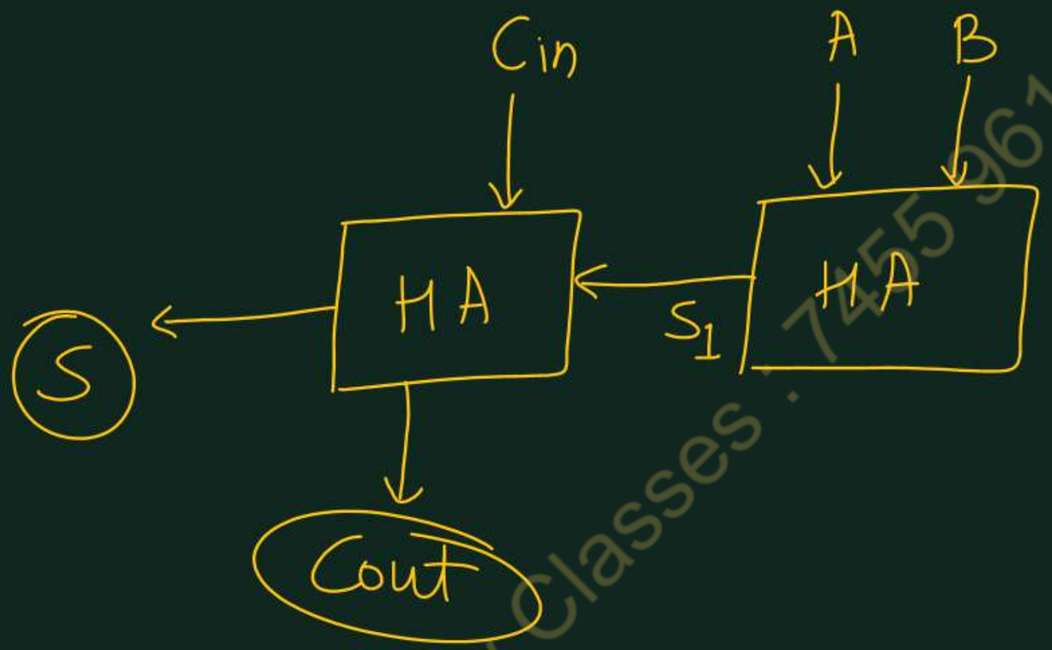
Implement using NAND gates only:

a) Half Subtractor

b) Full Subtractor

UNIVERSITY QUESTIONS

YEAR	QUESTION	MARKS
22-23	Draw a full adder using two half adders	2 marks
21-22	What is the role of subtractor in digital electronics?	2 marks
21-22	Construct half subtractor using NAND gates.	2 marks
21-22	Explain the concept of serial adder with accumulators.	10 marks
20-21	What is difference between "Ripple Carry Adder" and Carry Look-ahead Generator?	2 marks
19-20	Construct half subtractor using logic gates.	2 marks
19-20	Design a 4-bit parallel binary Adder/Subtractor circuit.	10 marks



UNIT 2 : Combinational circuits

Today's Target

- Magnitude Comparator
- 1 Bit, 2 Bit, 3 Bit, 4 Bit Comparator
- DPP
- University Questions

Digital Comparator

- A **digital comparator** or magnitude **comparator** is a hardware **electronic** device that takes two numbers as input in binary form and determines whether one number is greater than, less than or equal to the other number.
- **Comparators** are used in central processing units (CPUs) and microcontrollers (MCUs).
- One bit Comparator:

A	B	A=B	A>B	A<B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

A = 0	1	0	1
B = 1	0	0	1
A<B	A>B	A=B	A=B

$A<B: y_1 = \bar{A}B$

$A>B: y_2 = A\bar{B}$

$A=B: y_3 = \bar{A}\bar{B} + AB = A \odot B$

$$y_3 = \overline{y_1 + y_2}$$

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

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

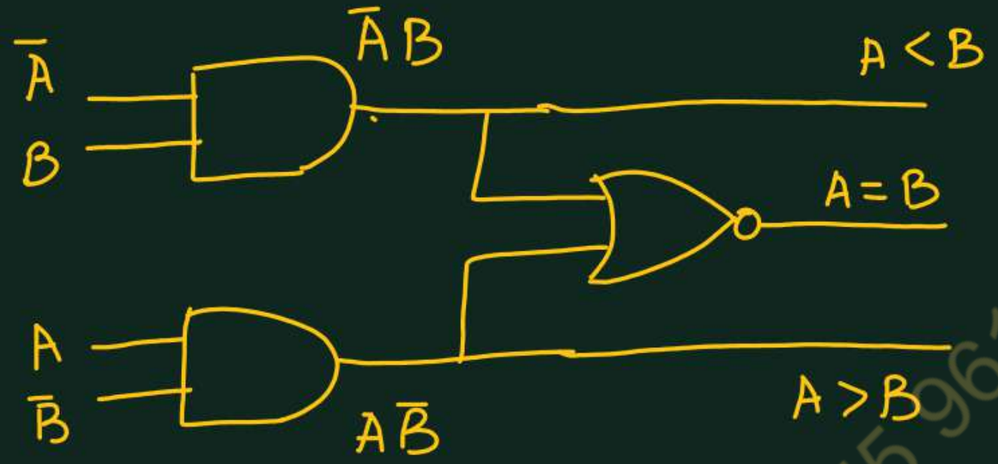
$$= (\bar{\bar{A}} + \bar{B})(\bar{A} + \bar{\bar{B}})$$

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

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

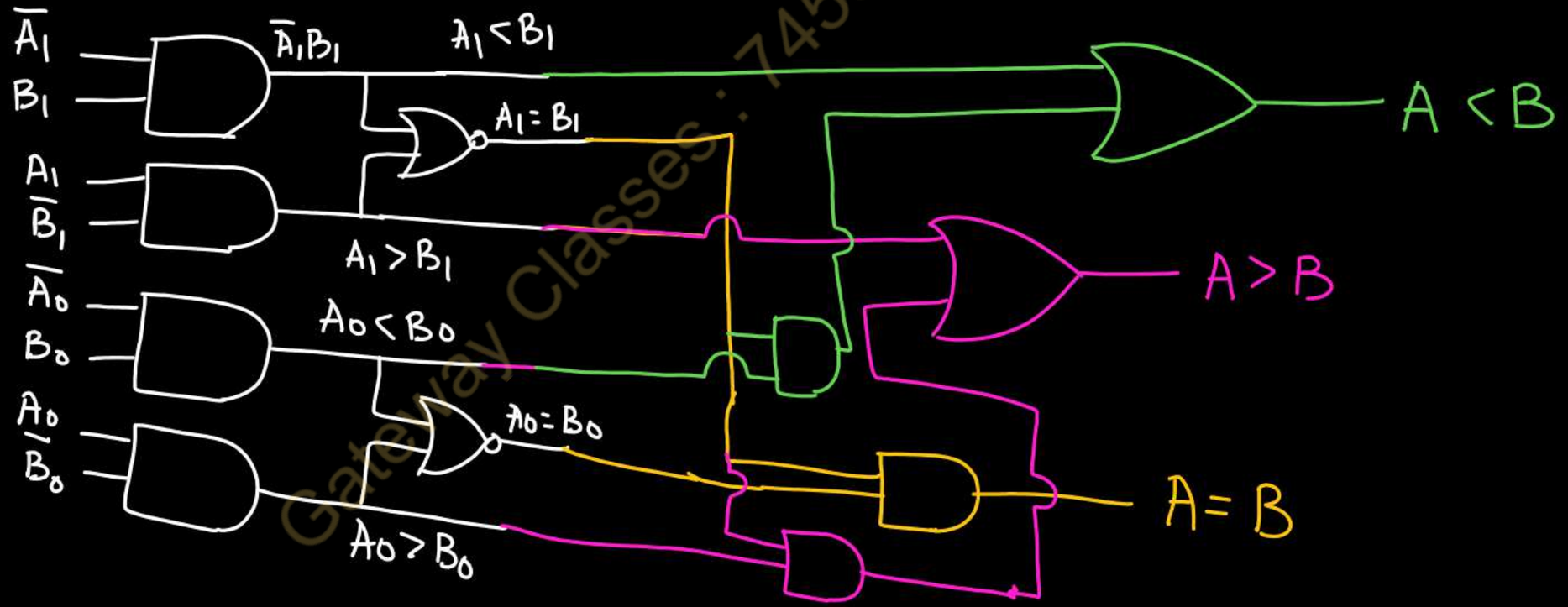
$$= A\bar{A} + AB + \bar{A}\bar{B} + B\bar{B}$$

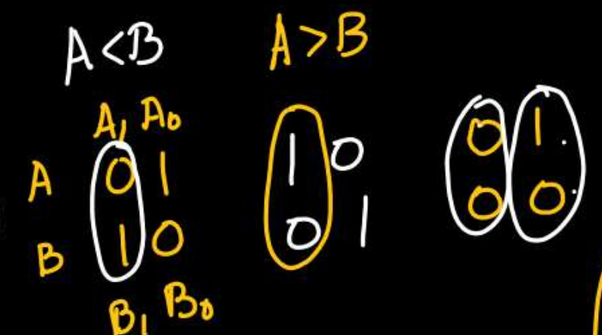
$$= AB + \bar{A}\bar{B}$$



2 BIT Digital Comparator

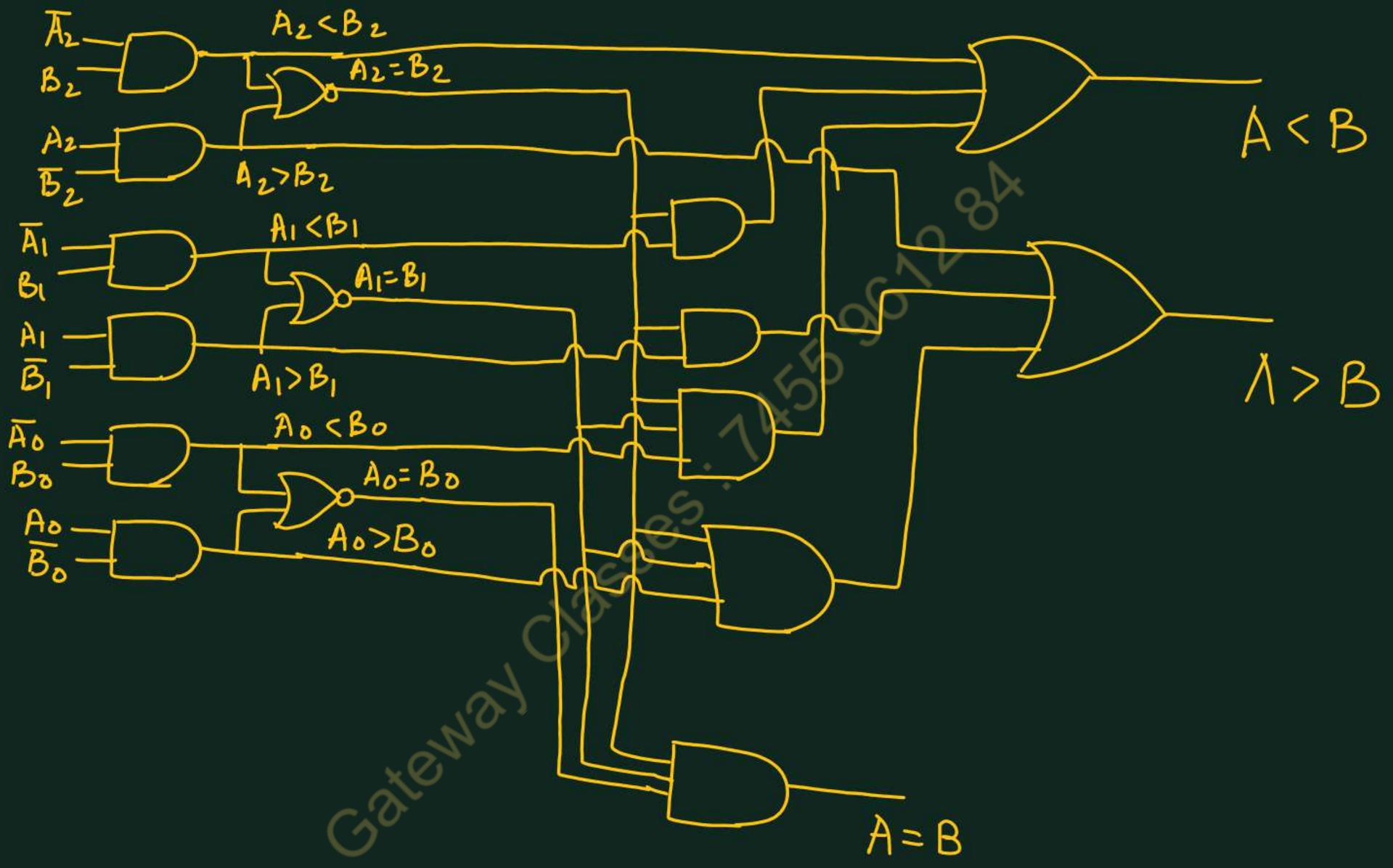
- Consider two 2-bit binary numbers A and B as; $A_1\ A_0$ & $B_1\ B_0$
- A=B if ; $A_1 = B_1$ (AND) $A_0 = B_0$
- A> B if ; $A_1 > B_1$ (OR) $A_1 = B_1$ (AND) $A_0 > B_0$
- A< B if ; $A_1 < B_1$ (OR) $A_1 = B_1$ (AND) $A_0 < B_0$





3 BIT Digital Comparator

- Consider two 3-bit binary numbers A and B as; $A_2\ A_1\ A_0$ & $B_2\ B_1\ B_0$
- A=B if ; $A_2 = B_2$ AND $A_1 = B_1$ AND $A_0 = B_0$
- A> B if ; $A_2 > B_2$ OR $A_2 = B_2$ AND $A_1 > B_1$ OR $A_2 = B_2$ AND $A_1 = B_1$ AND $A_0 > B_0$
- A< B if ; $A_2 < B_2$ OR $A_2 = B_2$ AND $A_1 < B_1$ OR $A_2 = B_2$ AND $A_1 = B_1$ AND $A_0 < B_0$



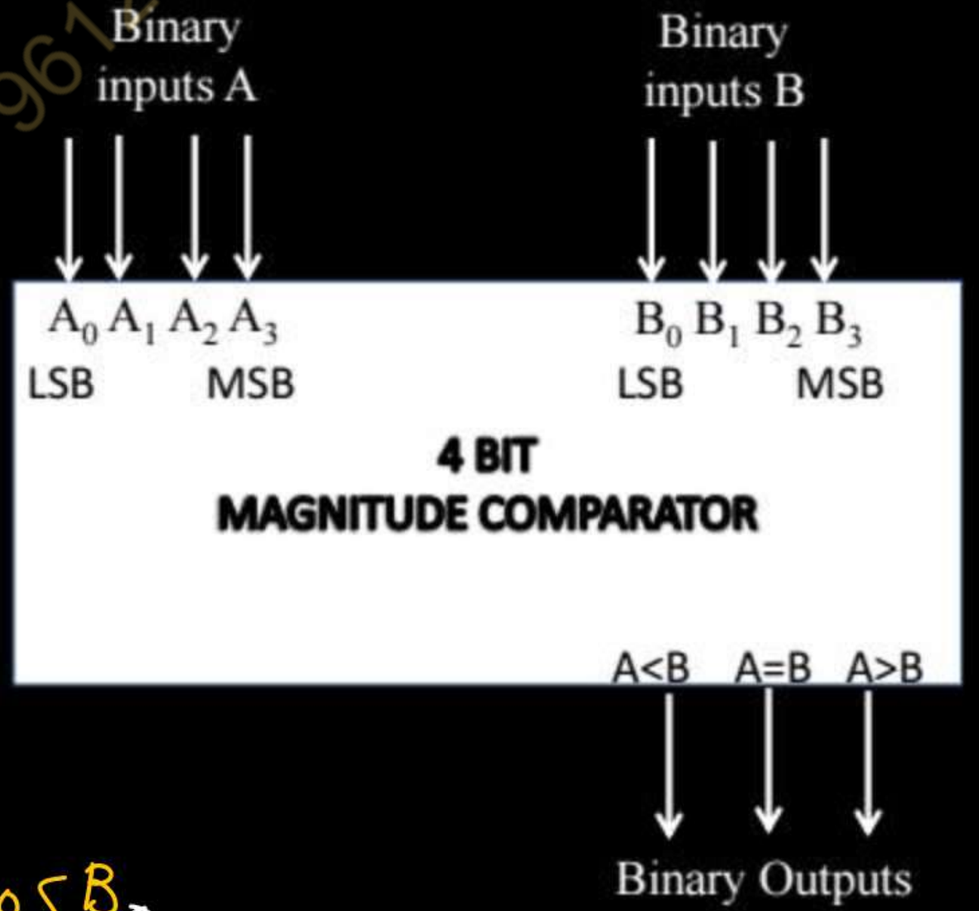
4 BIT Digital Comparator

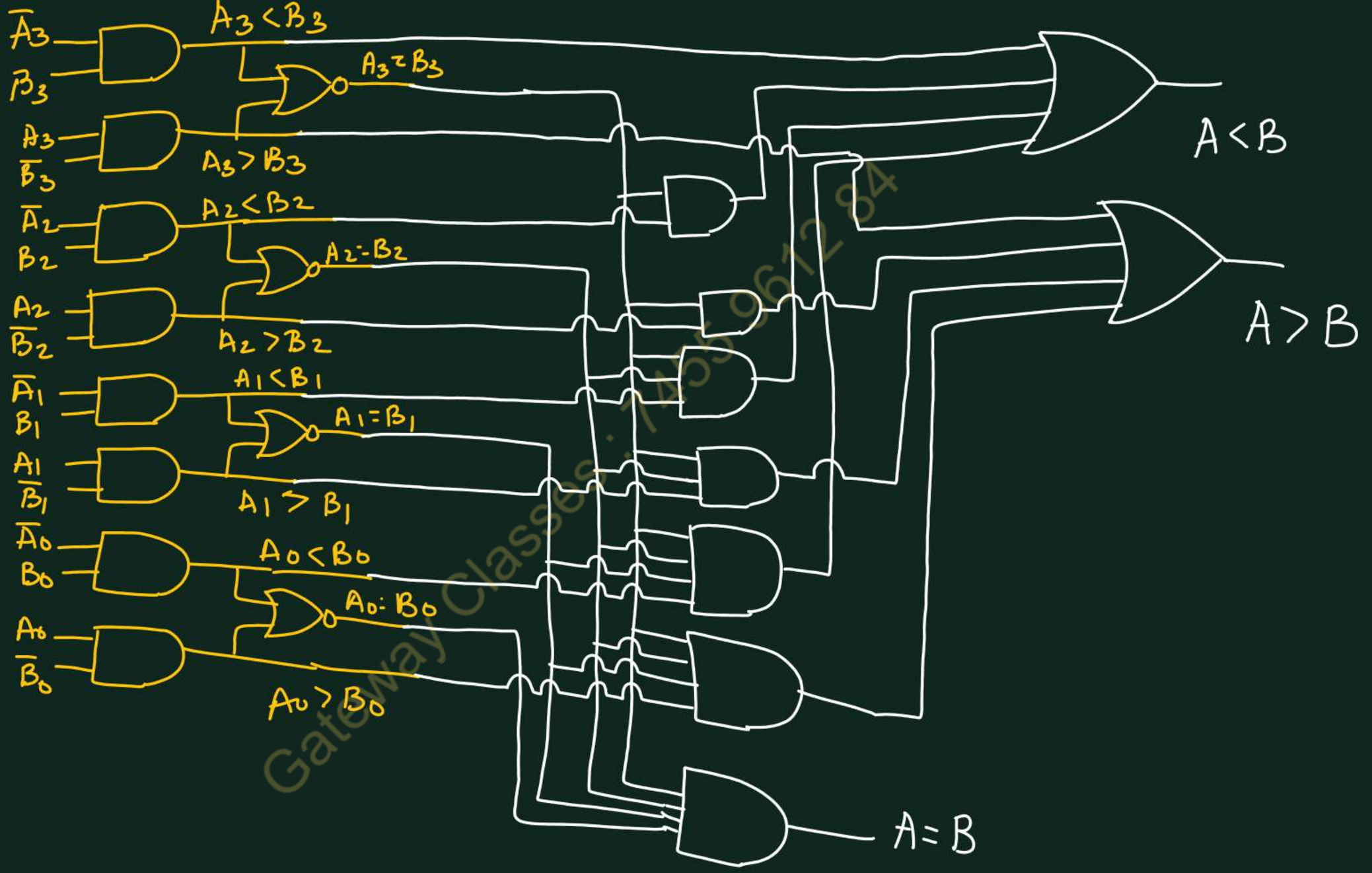
- Consider two 4-bit binary numbers A and B as; $A_3\ A_2\ A_1\ A_0$ & $B_3\ B_2\ B_1\ B_0$

- A=B if ; $A_3 = B_3$ AND $A_2 = B_2$ AND $A_1 = B_1$ AND $A_0 = B_0$

- A> B if ; $A_3 > B_3$ OR $A_3 = B_3$ AND $A_2 > B_2$
  OR $A_3 = B_3$ AND $A_2 = B_2$ AND $A_1 > B_1$
  OR $A_3 = B_2$ AND $A_2 = B_2$ AND $A_1 = B_1$ AND $A_0 > B_0$

- A< B if ; $A_3 < B_3$ OR $A_3 = B_3$ AND $A_2 < B_2$
  OR $A_3 < B_3$ AND $A_2 = B_2$ AND $A_1 < B_1$
  OR $A_3 = B_3$ AND $A_2 = B_2$ AND $A_1 = B_1$ AND $A_0 < B_0$





DPP

a) Design a 1 bit Magnitude Comparator.

b) Design a 2 bit Magnitude Comparator.

c) Design a 3 bit Magnitude Comparator.

d) Design a 4 bit Magnitude Comparator.

UNIVERSITY QUESTIONS

YEAR	QUESTION	MARKS
22-23	What is magnitude comparator? Design a Single-bit comparator circuit using logic gates.	7 marks
21-22	Design 2-bit magnitude comparator.	7 marks
19-20	Design a 4-bit comparator circuit using logic gates	7 marks

DIGITAL SYSTEM DESIGN

UNIT 2 : Combinational circuits

Today's Target

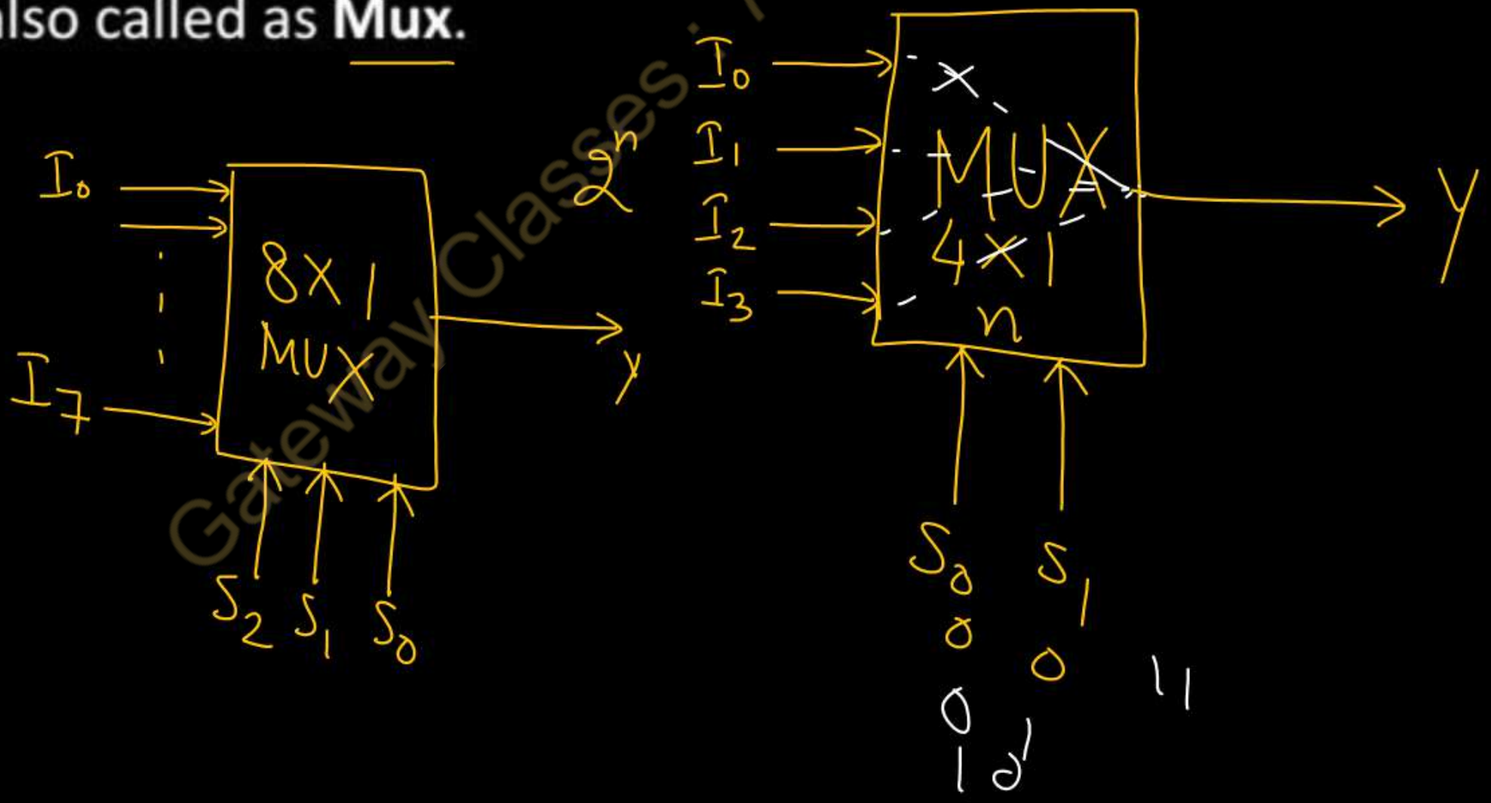
- MUX
- Implementation of Higher-order Multiplexers
- Numerical Practice of MUX
- DPP
- University Questions

UNIVERSITY QUESTIONS

YEAR	QUESTION	MARKS
22-23	Implement the function F = Σm(0,1,3,4,7,8,9,11,14,15) using 8:1 mux.	10 marks
20-21, 19-20	Implement a 4:1 multiplexer using 2:1 multiplexer.	2 marks
19-20	Implement a full adder by using 8:1 multiplexer.	10 marks
18-19	Implement the following Boolean function F (A, B, C, D)= (0, 1, 3, 4, 7, 8, 9, 11, 14,15) using (I) 4:1 MUX (ii) 2:1 MUX	10 marks

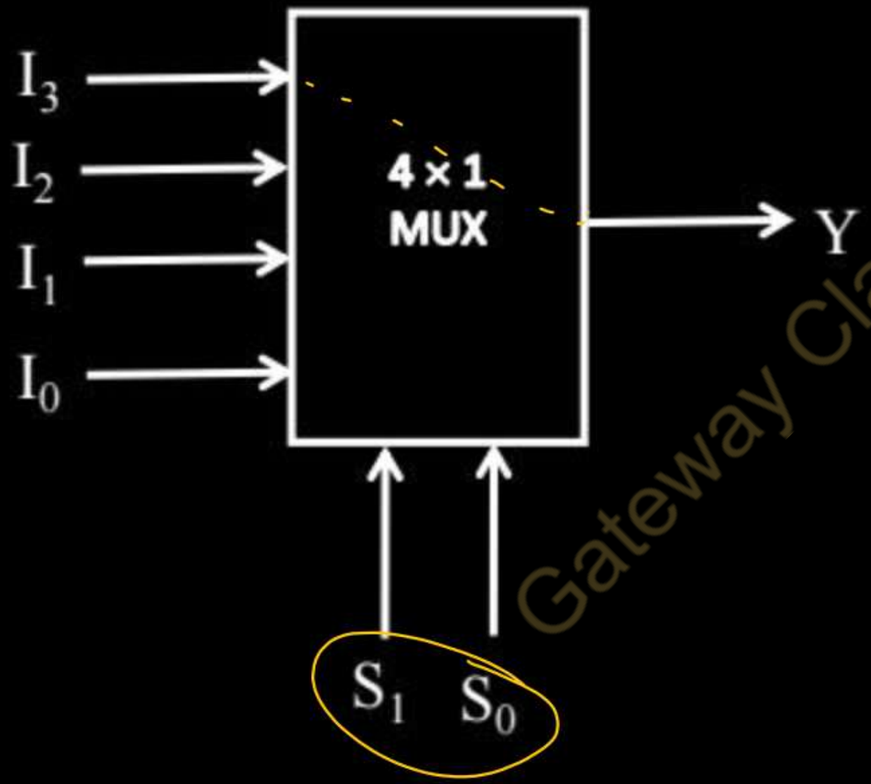
Multiplexers

- **Multiplexer** is a combinational circuit that has
- maximum of 2^n data inputs,
- 'n' selection lines and single output line.
- One of these data inputs will be connected to the output based on the values of selection lines.
- Multiplexer is also called as **Mux**.



4x1 Multiplexer

- four data inputs I_3, I_2, I_1 & I_0,
- two selection lines S_1 & S_0 and
- one output Y.
- The **block diagram** of 4x1 Multiplexer is shown in the following figure.



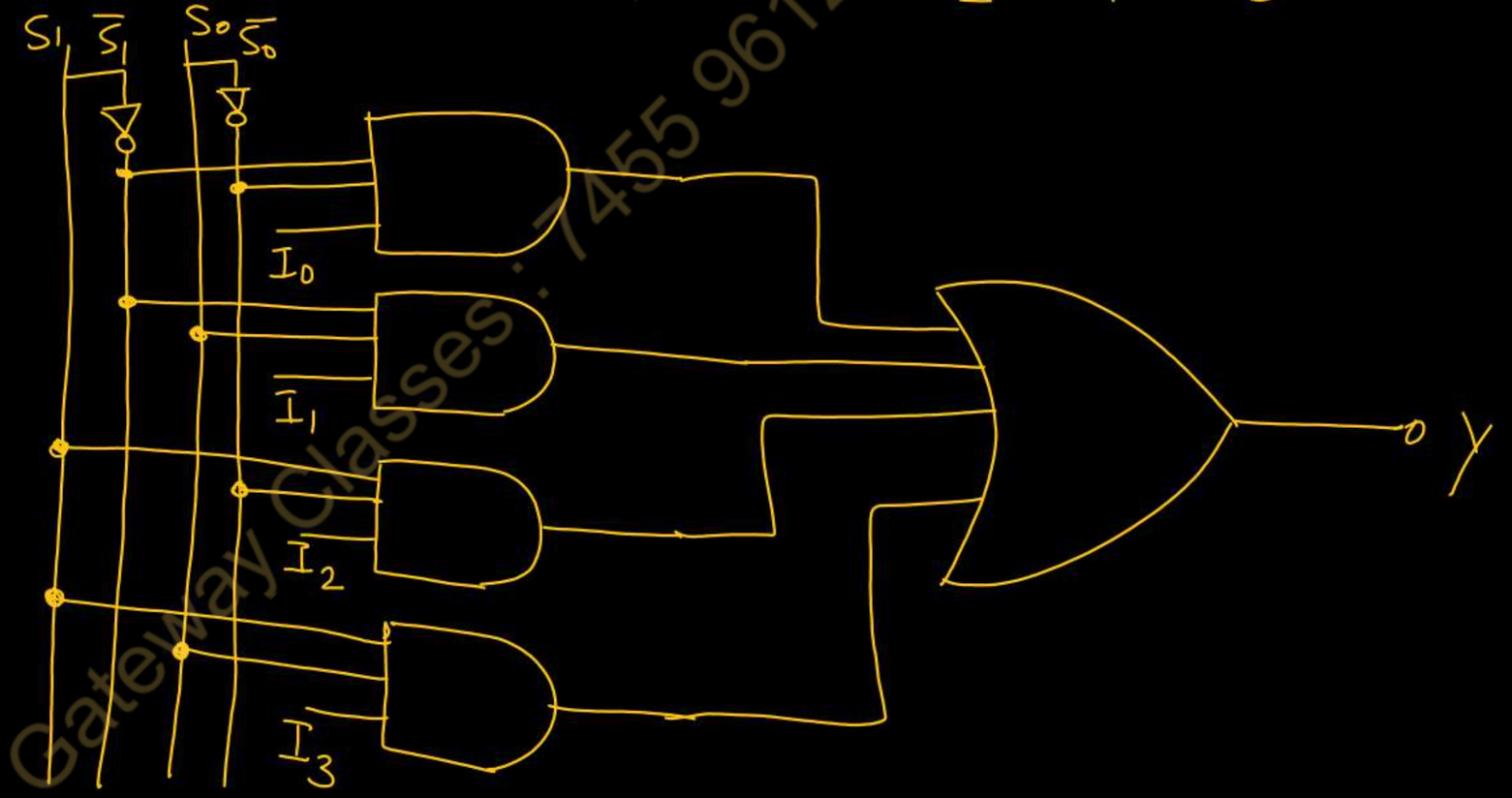
One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines.

Truth table of 4x1 Multiplexe

Selection Lines		Output
S1	S0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Boolean function for output :

$$Y = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$$

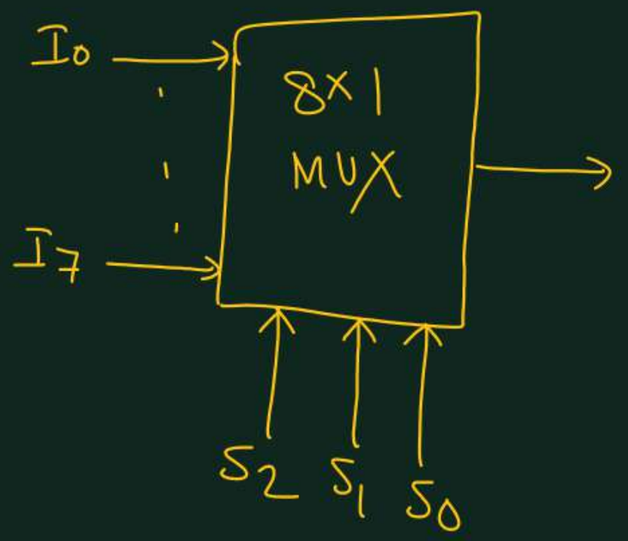


8 × 1 MUX

I/P = 8

Select = 3

O/P = 1



S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

$$Y = \bar{S}_2\bar{S}_1\bar{S}_0 I_0 + \bar{S}_2\bar{S}_1 S_0 I_1 + \bar{S}_2 S_1 \bar{S}_0 I_2 + \bar{S}_2 S_1 S_0 I_3 + S_2\bar{S}_1\bar{S}_0 I_4 + S_2\bar{S}_1 S_0 I_5 + S_2 S_1 \bar{S}_0 I_6 + S_2 S_1 S_0 I_7$$

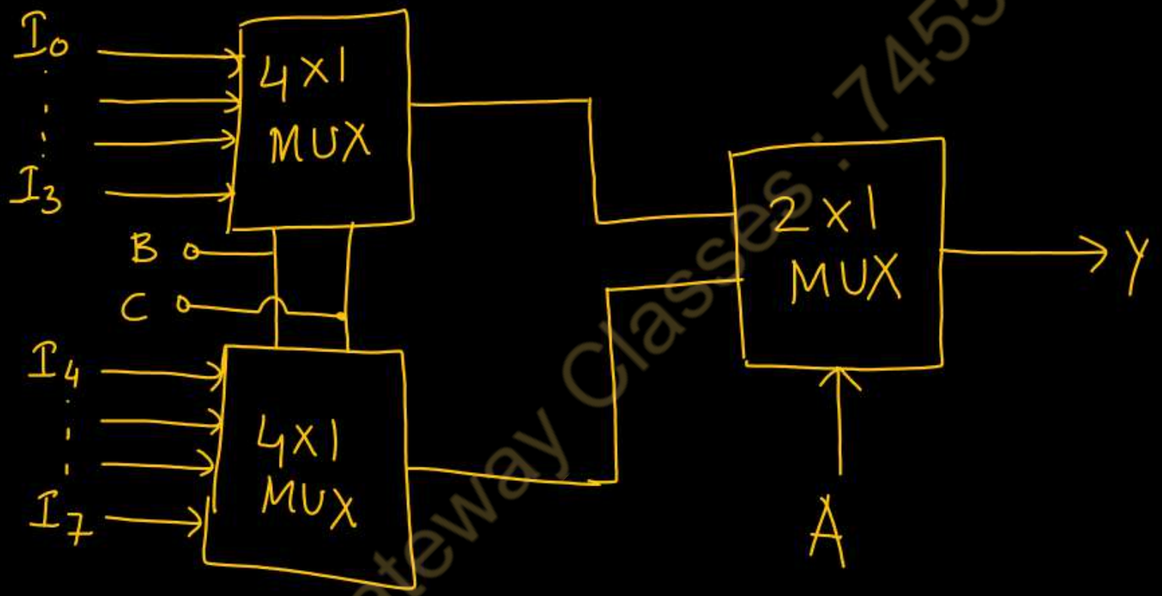
Implementation of Higher-order Multiplexers.

- 8x1 Multiplexer using 4x1 Multiplexers and 2x1 Multiplexer.

$$\text{No. of MUX Req.} = \frac{8\times1}{4\times1} = 2$$

8x1 : $n = 3$ A (B C)

4x1 : $n = 2$



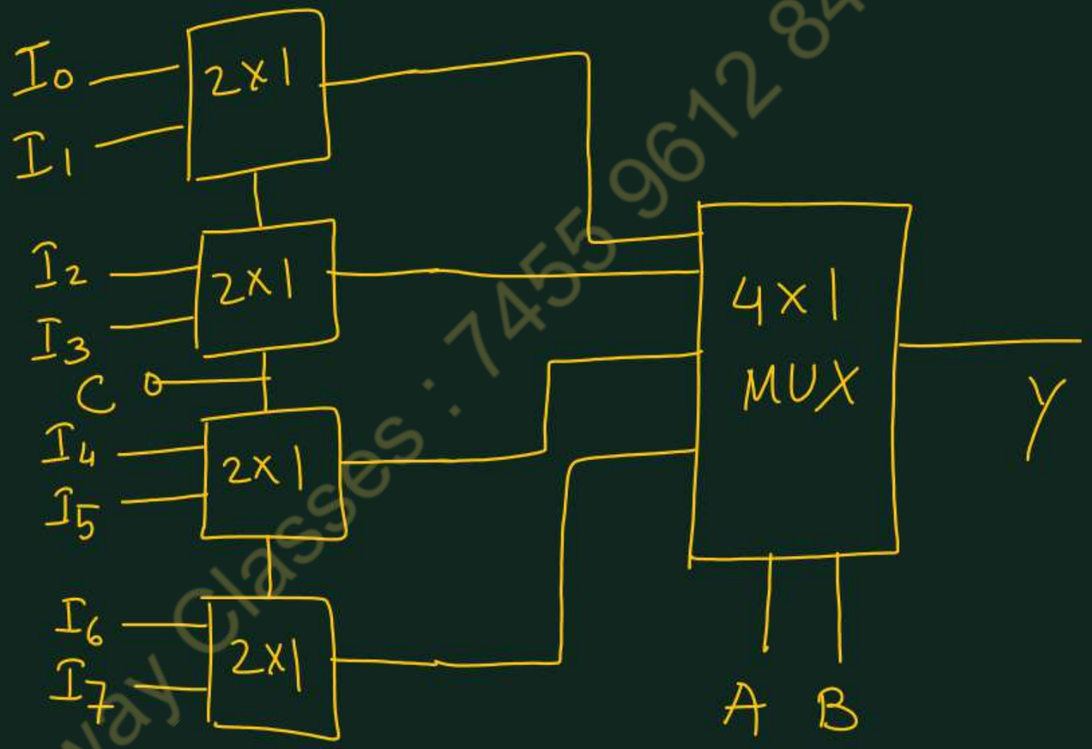
8×1 MUX using 2×1 MUX & 4×1 MUX

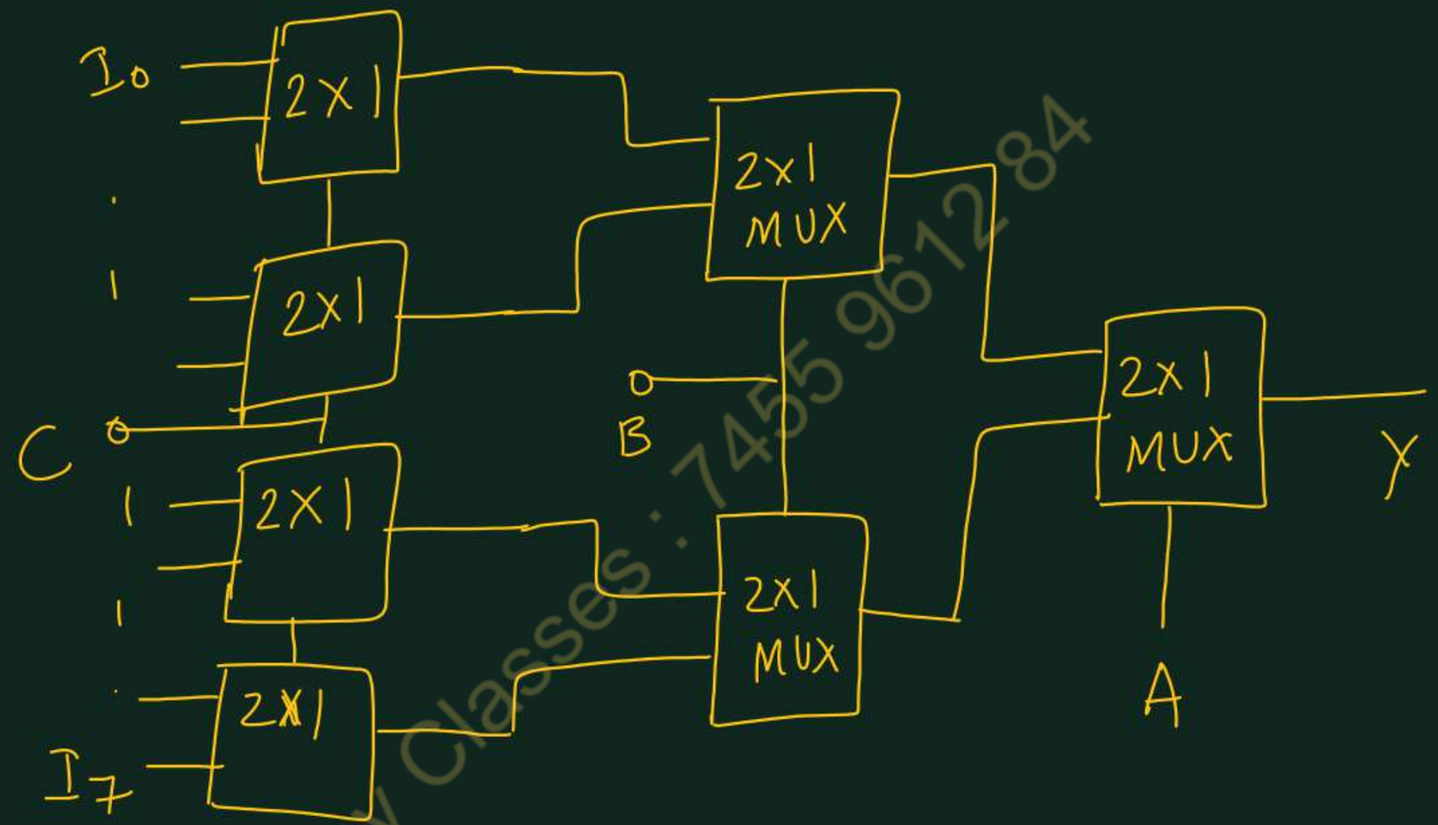
$$\text{no. of req. MUX} = \frac{8\times1}{2\times1} = 4$$

A B C

8×1 : n = 3

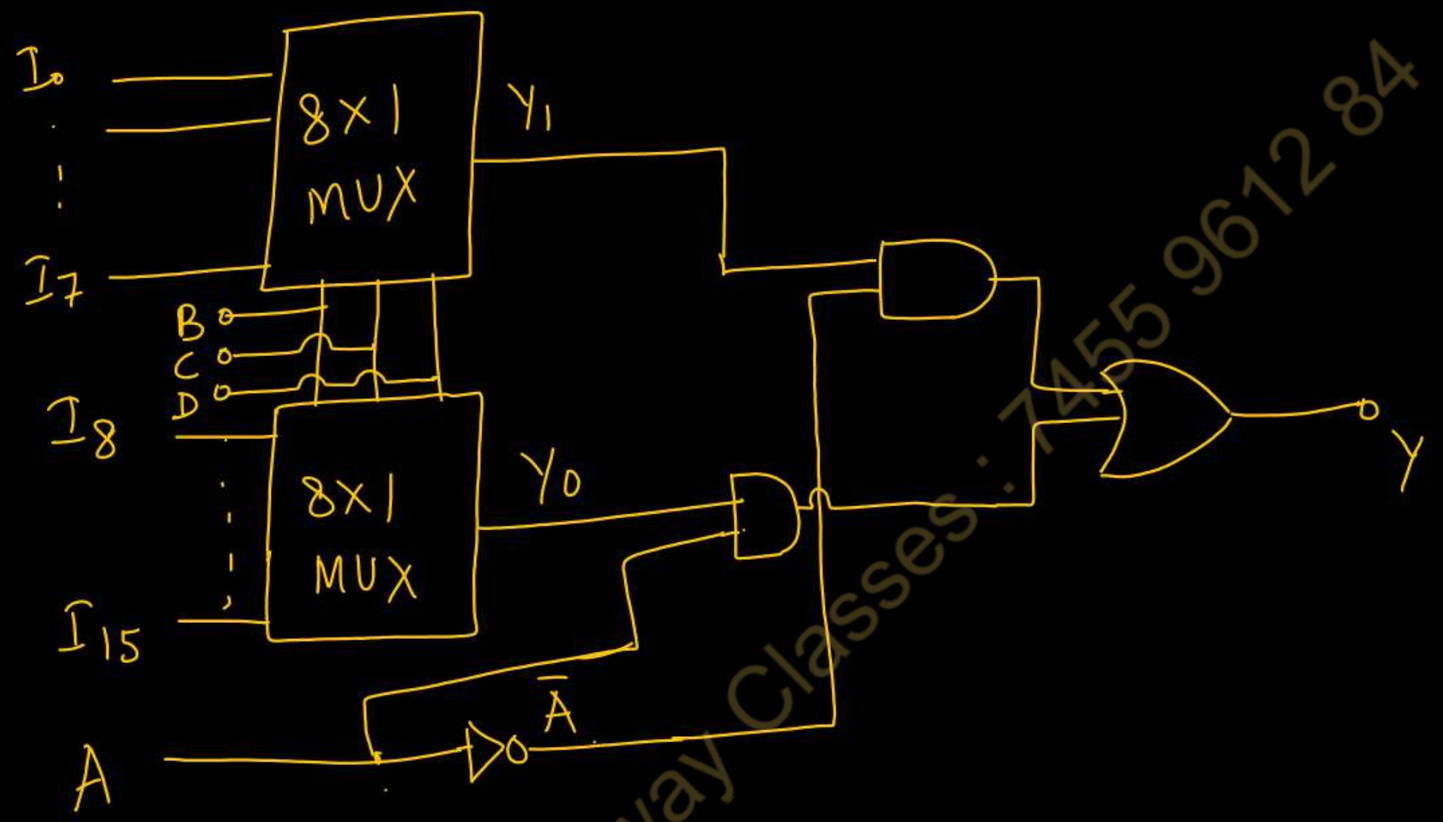
2×1 : n = 1





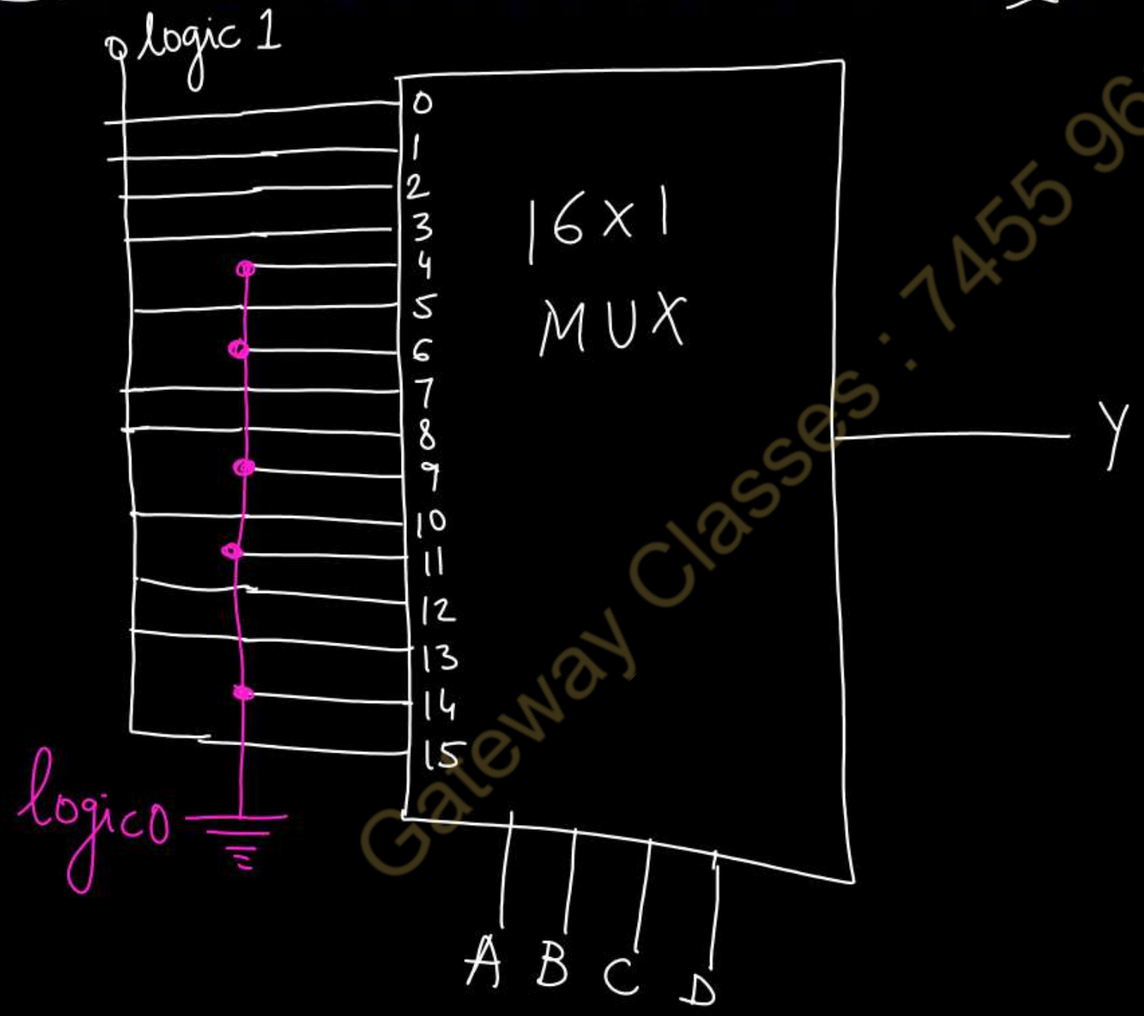
A B C

16x1 Multiplexer using 8x1

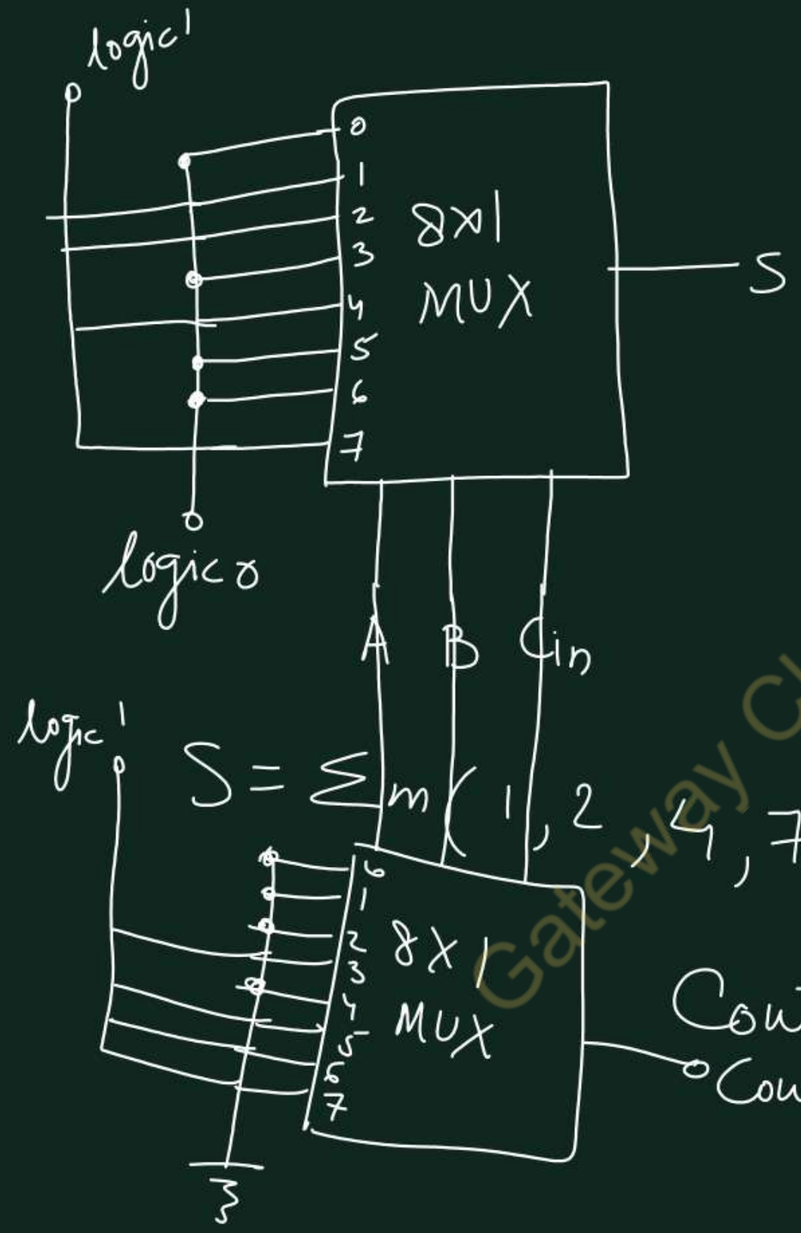


Design multiplexer implementations for the following functions using the numerical method.

- $Z = F(A,B,C,D) = \sum m(0,1,2,3,5,7,8,10,12,13,15)$

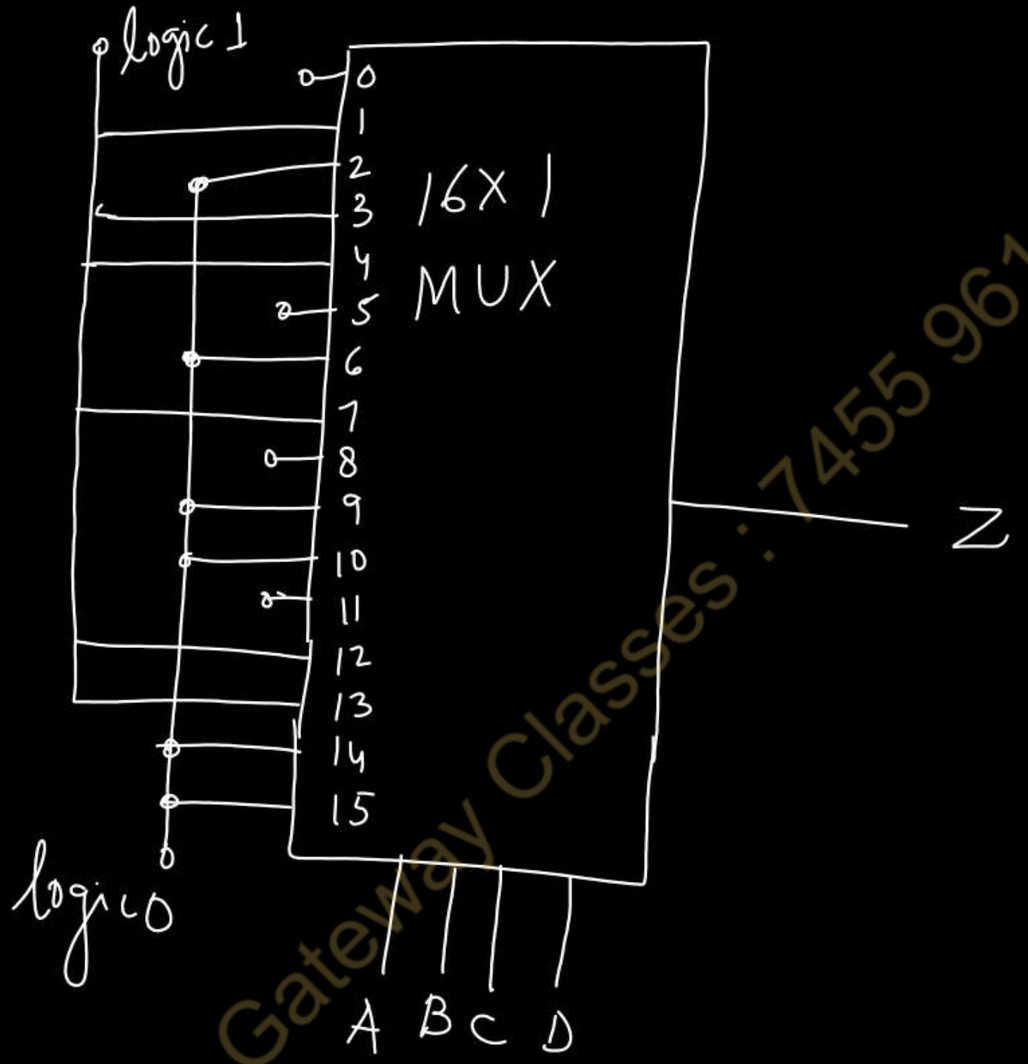


Full Adder

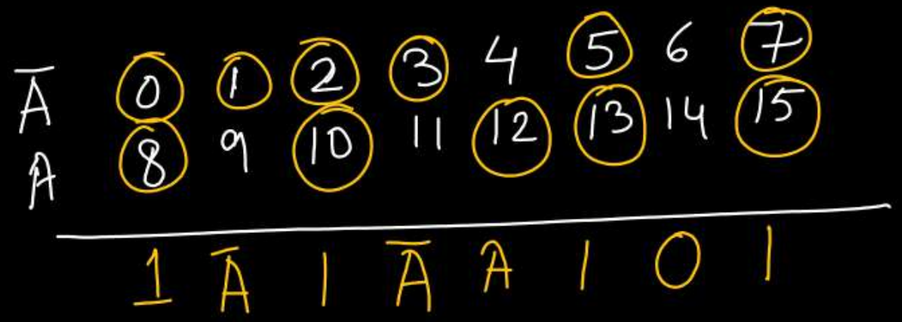


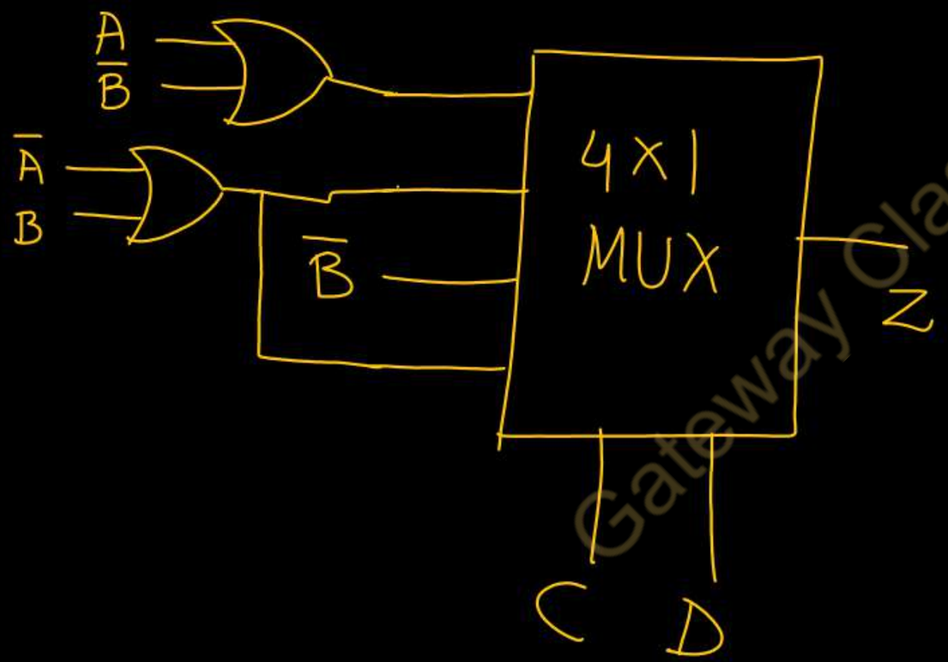
A	B	Cin	S	Cout
0	0	0	0	
0	0	1	1	
0	1	0	1	
0	1	1	0	1
1	0	0	1	
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

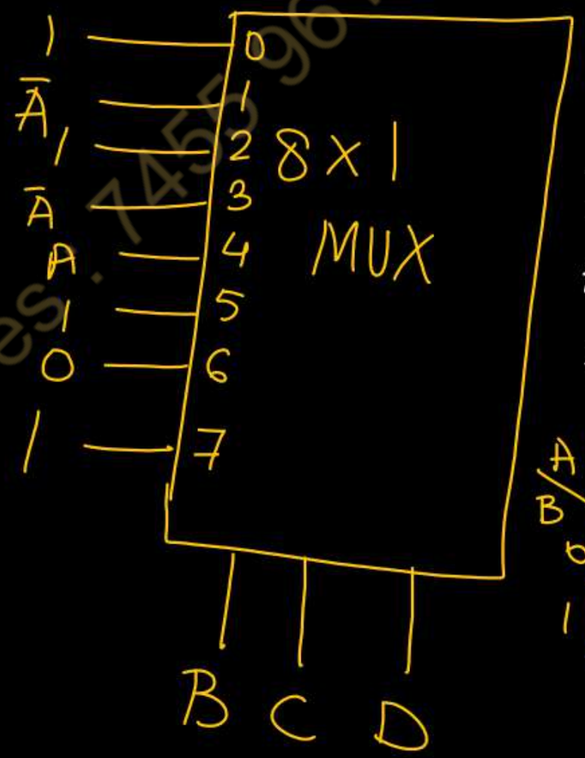
$$Z = F(A,B,C,D) = \sum m(1,3,4,7,12,13) + d(0,5,8,11)$$

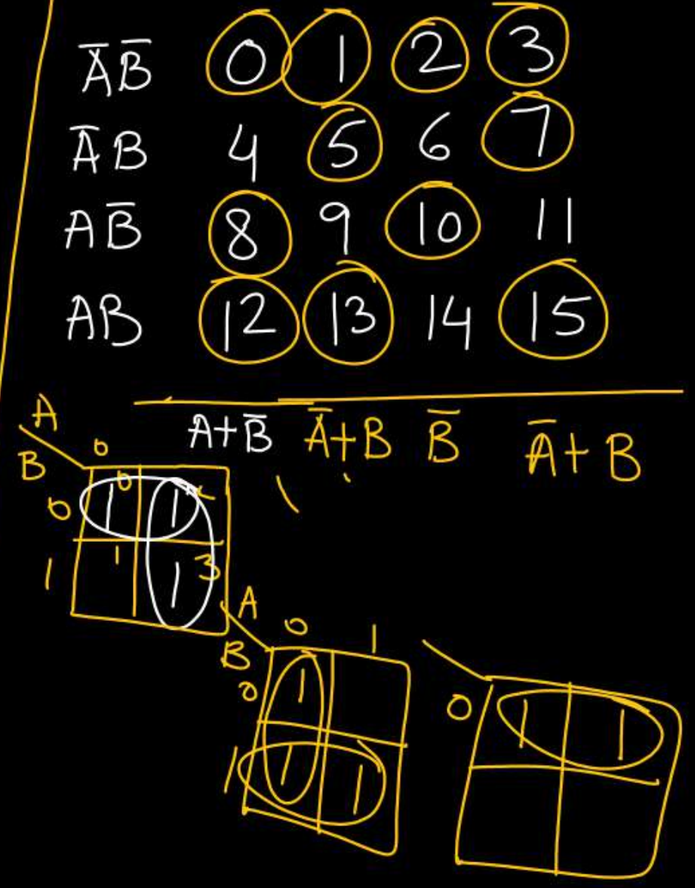


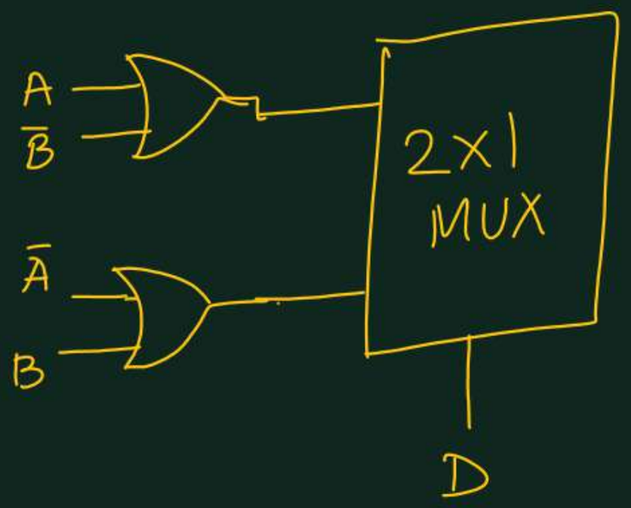
Implement Z= F(A,B,C,D) = ∑m(0,1,2,3,5,7,8,10,12,13,15) using 8×1 MUX







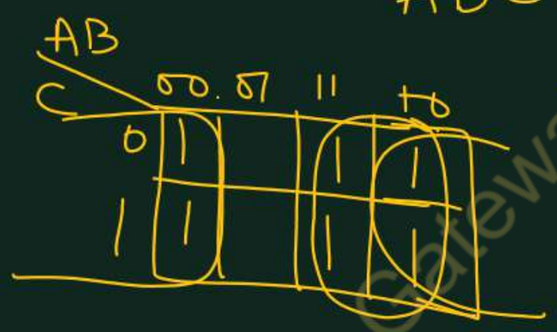




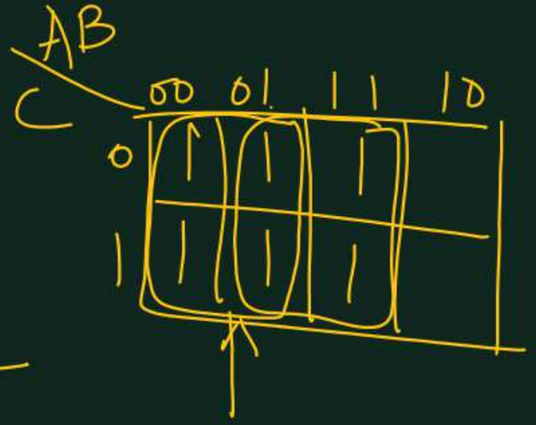
$\bar{A}\bar{B}\bar{C}$ 0
$\bar{A}\bar{B}C$ 1
$\bar{A}B\bar{C}$ 2
$\bar{A}BC$ 3
$A\bar{B}\bar{C}$ 4
$A\bar{B}C$ 5
$AB\bar{C}$ 6
ABC 7

$\bar{D}$ D

0 1
2 3
4 5
6 7
8 9
10 11
12 13
14 15



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



DPP

a) Design a 16 × 1 MUX using basic logic gates.

b) Z= F(A,B,C,D) = A'C'D' + BC'D' + AB'C' + A'BC'D + A'B'CD' using 16 × 1 MUX and 8 × 1 MUX

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

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

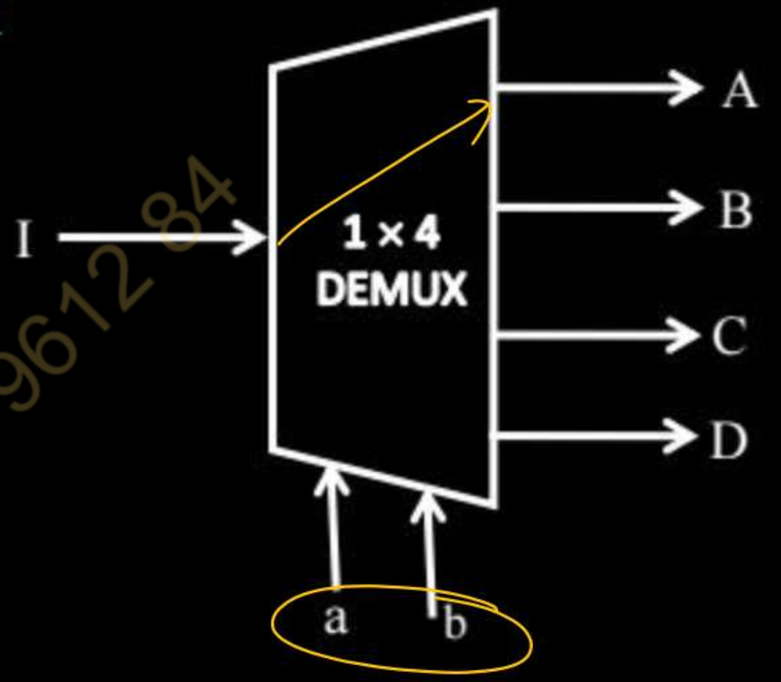
DIGITAL SYSTEM DESIGN

UNIT 2 : Combinational circuits

Today's Target

- DEMUX
- Implementation of Higher-order DeMultiplexers
- Multiplexed Display
- Pixel Oriented & Character Oriented
- Display Driver

DEMULTIPLEXER



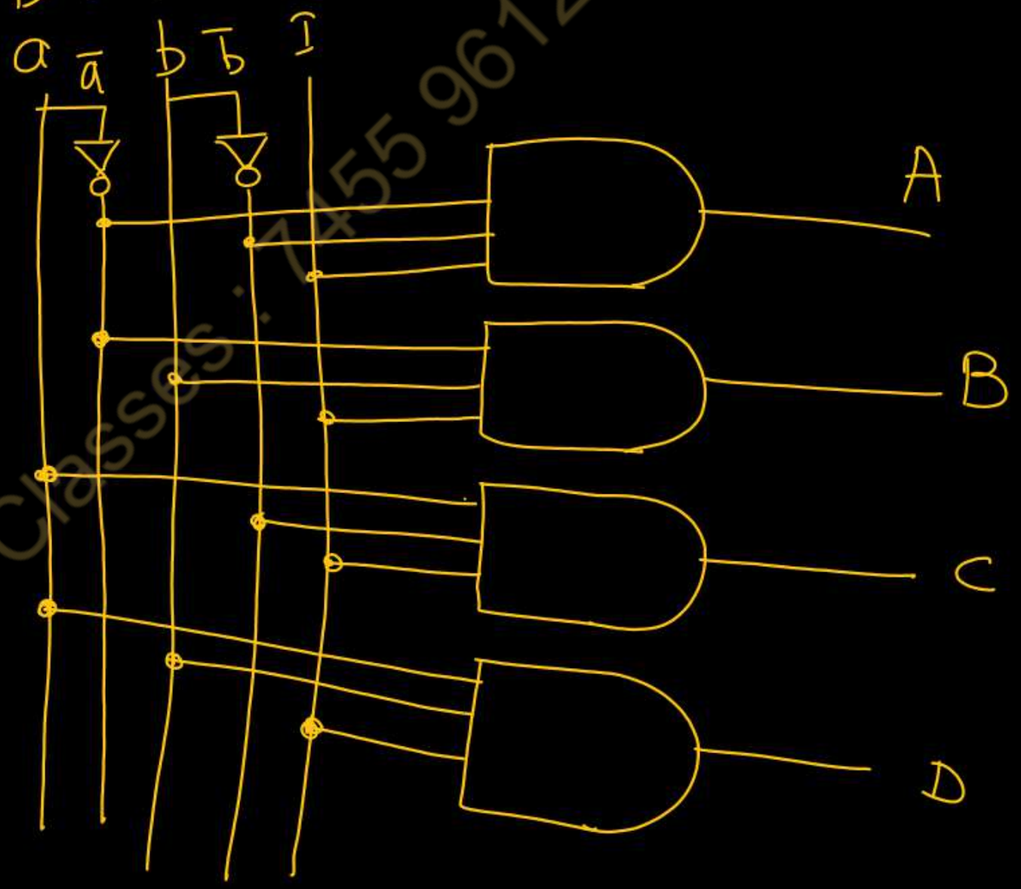
- inverse of the multiplexer
- single data input
- n address inputs
- 2^n outputs
- The address input determine which data output is going to have the same value as the data input.
- other data outputs will have the value 0.
- abbreviation used is DEMUX.

TRUTH TABLE AND CIRCUIT DIAGRAM

Data Input	Select Inputs		Outputs			
I	a	b	A	B	C	D
I	0	0	I	0	0	0
I	0	1	0	I	0	0
I	1	0	0	0	I	0
I	1	1	0	0	0	I

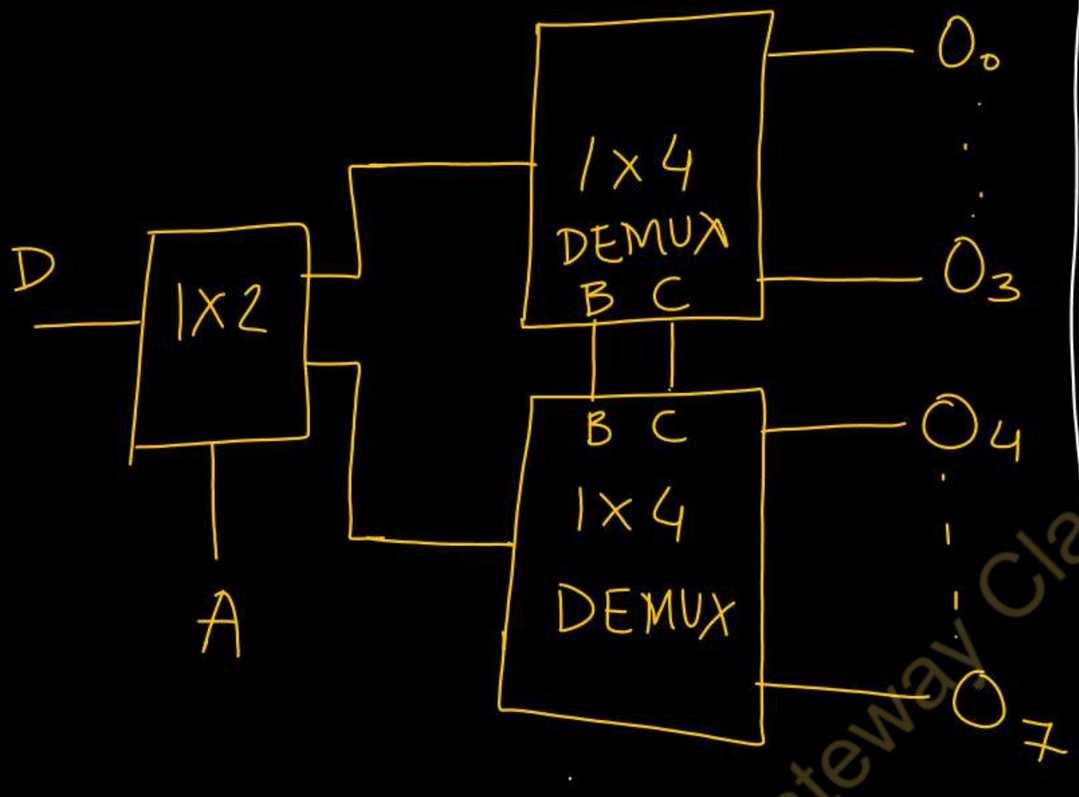
$A = \bar{a}\,\bar{b}\,I \quad B = \bar{a}\,b\,I \quad C = a\,\bar{b}\,I$

$D = a\,b\,I$

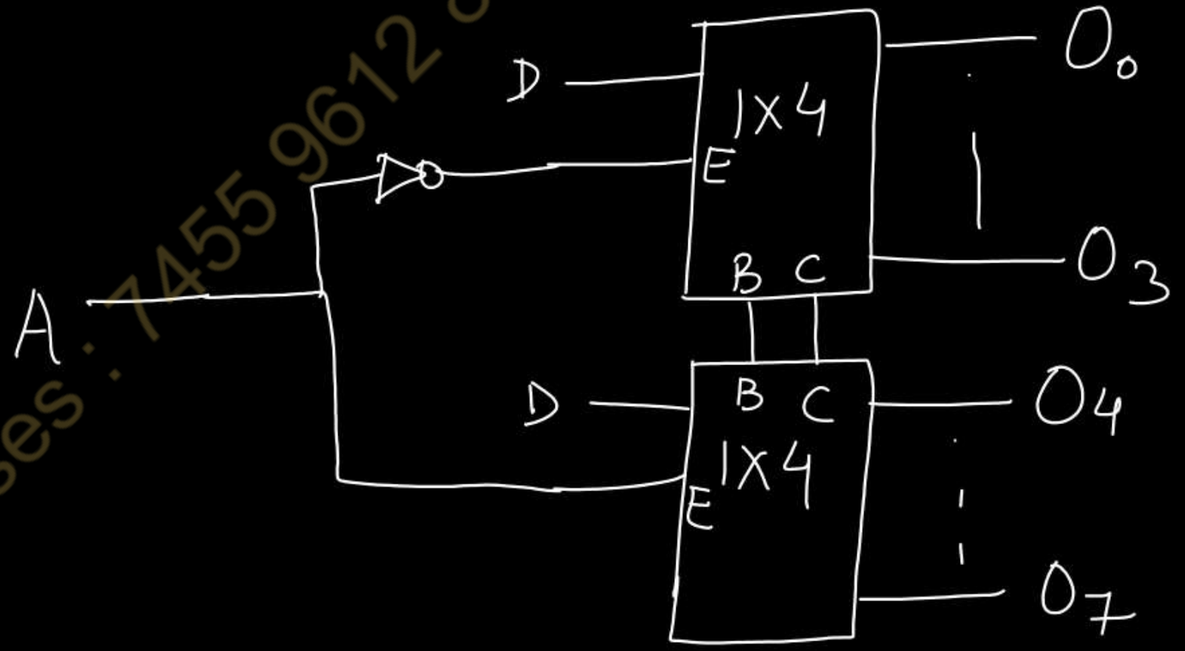


HIGHER ORDER DEMUX USING LOWER ORDER DEMUX

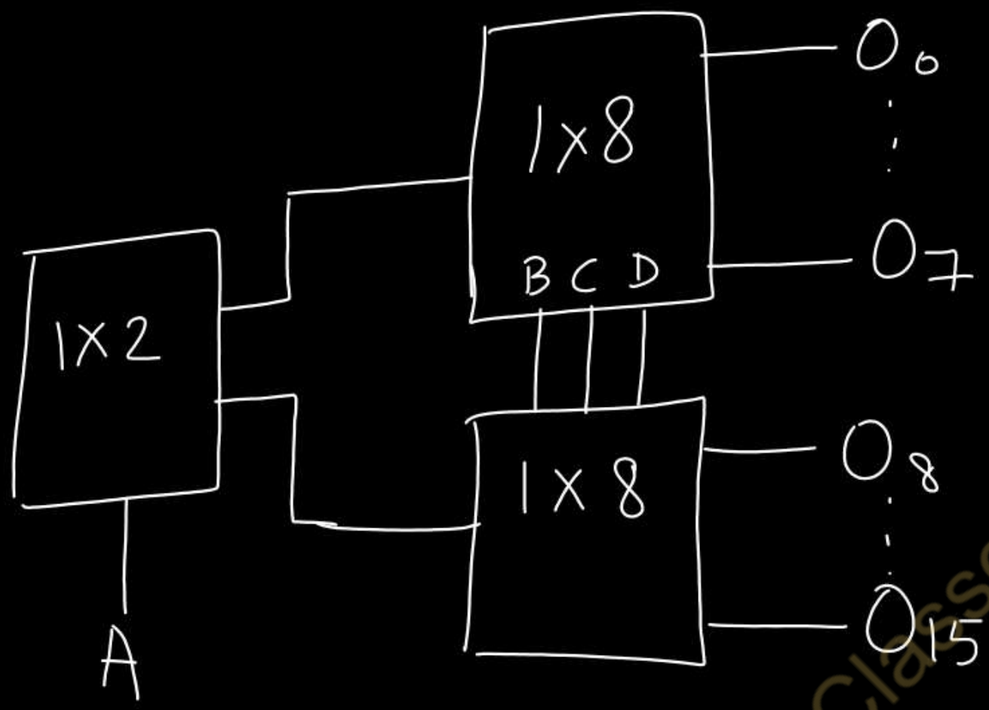
1 × 8 USING 1 × 4 DEMUX



A B C

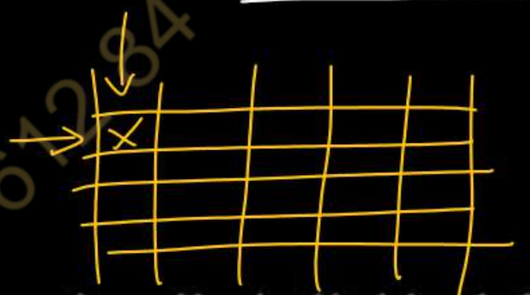


1 × 16 USING 1 × 8 DEMUX



Multiplexed display

- Multiplexed displays are electronic display devices where the entire display is not driven at one time.
- Instead, sub-units of the display are multiplexed
- typically, rows or columns for a dot matrix display or
- individual characters for a character oriented display, occasionally individual display elements



- These are driven one at a time, but the electronics and the persistence of vision combine to make the viewer believe the entire display is continuously active.
- Advantages compared to a non-multiplexed display:

➢ Fewer wires (often, far fewer wires) are needed

➢ simpler driving electronics can be used

➢ both lead to reduced cost

➢ reduced power consumption

Two Broad Categories

- character-oriented displays
- pixel-oriented displays

Character-oriented displays

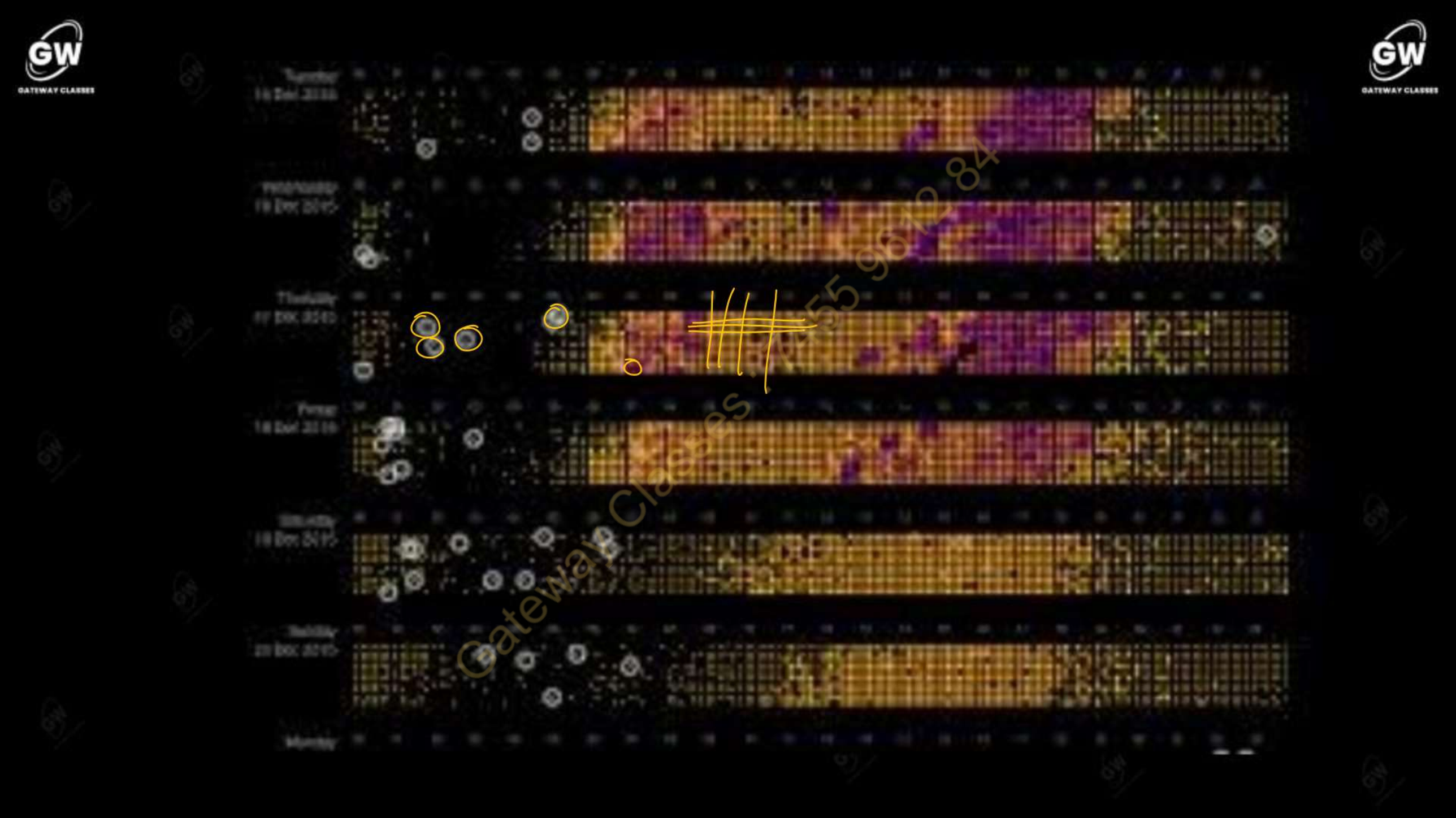
- display an entire character at one time.
- various segments of each character are connected in a two-dimensional diode matrix and will only illuminate if both the "row" and "column" lines of the matrix are at the correct electrical potential.
- If the display had been built with every segment being individually connected, the display would have required 49 wires just for the digits, with more wires being needed for all of the other indicators that can be illuminated.
- By multiplexing the display, only seven "digit selector" lines and seven "segment selector" lines are needed.
- The extra indicators are arranged as if they were segments of an additional digit or two or extra segments of existing digits and are scanned using the same multiplexed strategy as the real digits.
- A keyboard matrix circuit has a very similar arrangement as a multiplexed display


Gateway Classes : 7455 9612 84

Pixel-oriented displays

- In dot matrix displays, individual pixels are located at the intersections of the matrix's "row" and "column" lines and each pixel can be individually controlled.
- For a typical 1024×768 (XGA) computer screen, 2,359,296 wires would be needed for non-multiplexed control. That many wires would be completely impractical.

- By arranging the pixels into a multiplexed matrix, only 1792 wires are needed.
- Pixel-oriented displays may drive a single pixel at a time or an entire row or column of pixels simultaneously.
- Active-matrix LCD provide a storage element at each pixel so that the pixel continues to display the correct state even when not being actively driven.



Break up

- Because most multiplexed displays do not present the entire display simultaneously, they are subject to "break up" if :

➢ the observer's point of regard is in motion.

➢ People with nystagmus (involuntary eye movement) are much more likely to experience the effect.

➢ provoked by chewing hard candy which causes vibration of the user's eyes.

➢ The multiplexed nature of a display can also be revealed by observing it through a mechanical stroboscope.


alamy
Gateway Classes 7455 9612 84
alamy
alamy

Display driver

- A **display driver** is usually a semiconductor integrated circuit
- provides an interface function between a microprocessor or a microcontroller or general-purpose peripheral interface and a particular type of display device, e.g. LCD, LED, CRT, ETC.
- accept commands and data using serial or parallel interface and generate signals with suitable voltage, current, timing and demultiplexing to make the display show the desired text or image.
- The display driver may itself be an application-specific microcontroller and may incorporate RAM, Flash memeory, EEPROM, ROM
- Example of a display driver IC is the Hitachi HD44780 LCD controller, KS0108, SSD1815 and ST7920 .

DIGITAL SYSTEM DESIGN

UNIT 2 : Combinational circuits

Today's Target

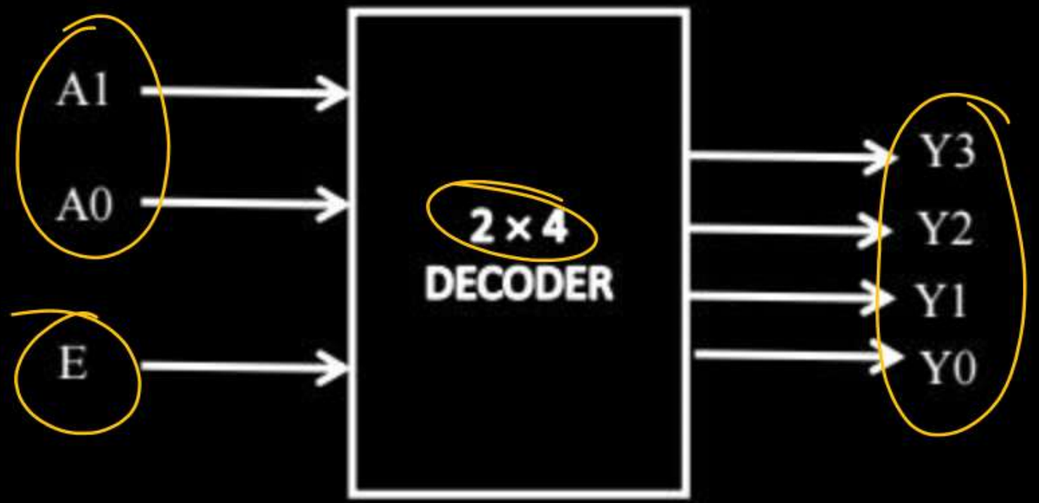
- Decoder ✓
- Implementation of Higher-order Decoders ✓
- Numerical Practice
- DPP
- University Questions

UNIVERSITY QUESTIONS

YEAR	QUESTION	MARKS
22-23	Explain Decoder with neat diagram. Implement the logic expression $Y = \Sigma m(2, 4, 6, 7)$ using decoder.	10 marks

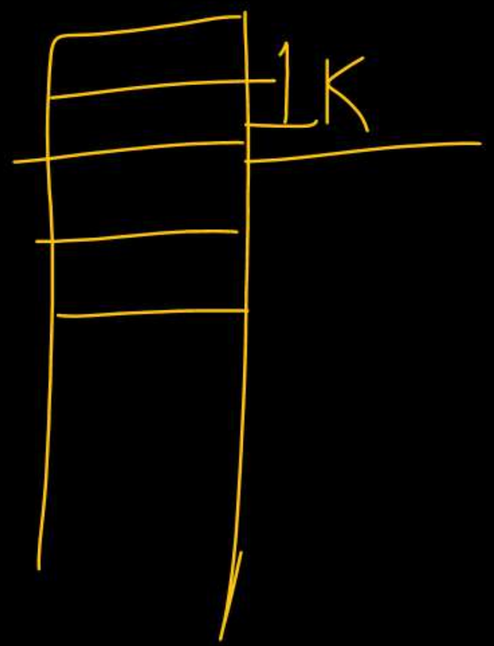
DECODER

- a combinational logic circuit that converts binary information from the n coded inputs to a maximum of 2^n unique outputs.
- Some decoders also have one or more "enable" inputs.
- n input signals
- 2^n unique output signals



APPLICATIONS

- data multiplexing and data demultiplexing
- seven segment displays
- Address decoders for memory
- Address decoders for port mapped I/O

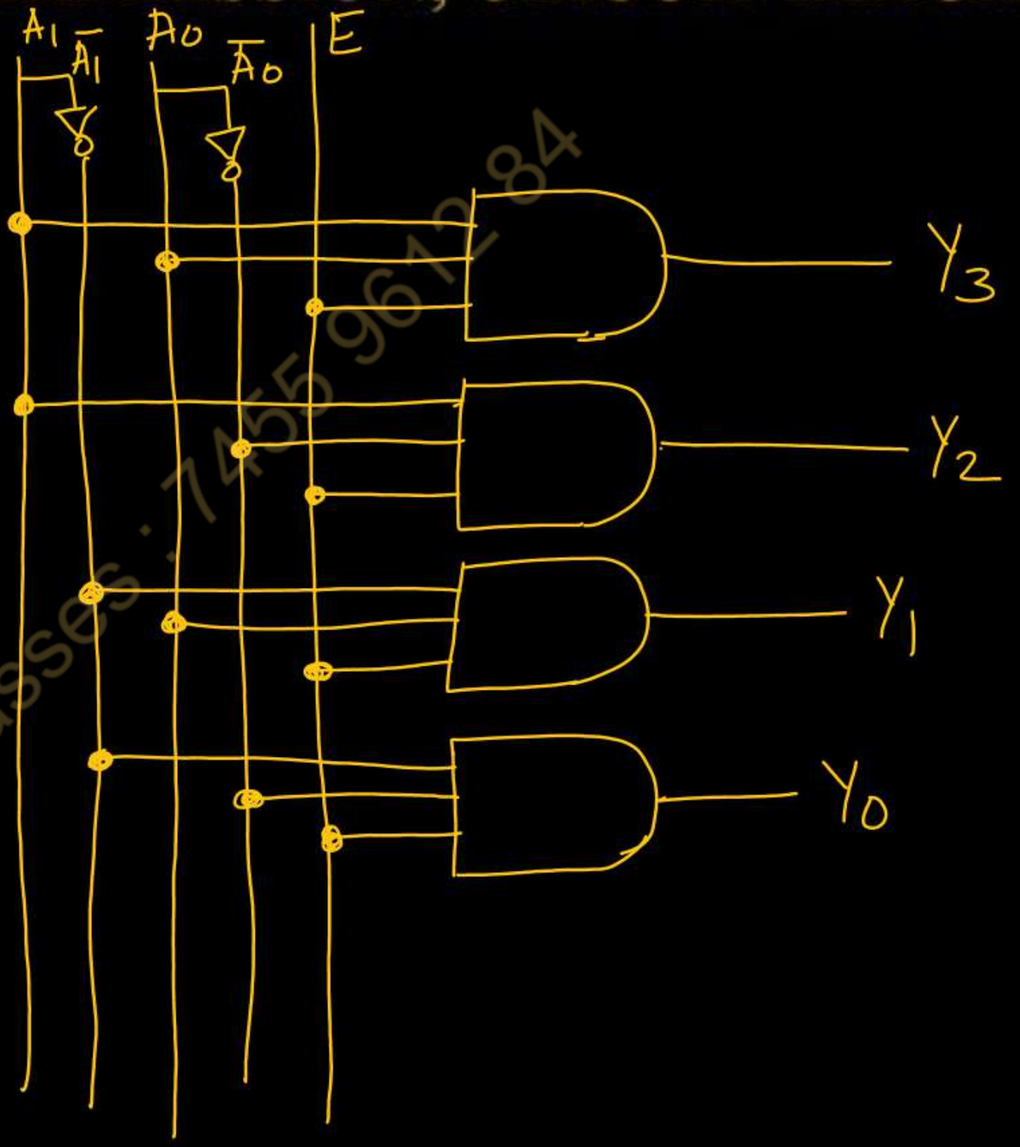


TRUTH TABLE, BOOLEAN EXPRESSION, CIRCUIT DIAGRAM

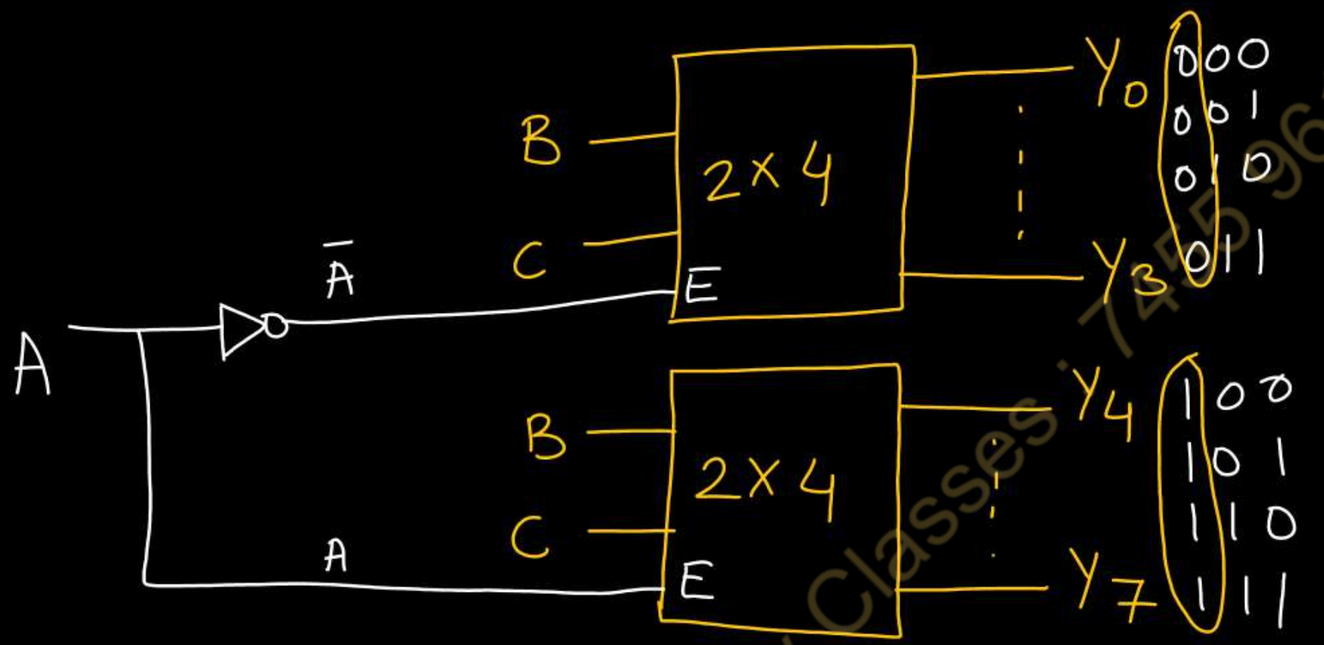
ENABLE	INPUTS		OUTPUTS			
E	A1	A0	Y3	Y2	Y1	Y0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

$Y_3 = A_1 A_0 E$ $\quad Y_1 = \bar{A}_1 A_0 E$

$Y_2 = A_1 \bar{A}_0 E$ $\quad Y_0 = \bar{A}_1 \bar{A}_0 E$



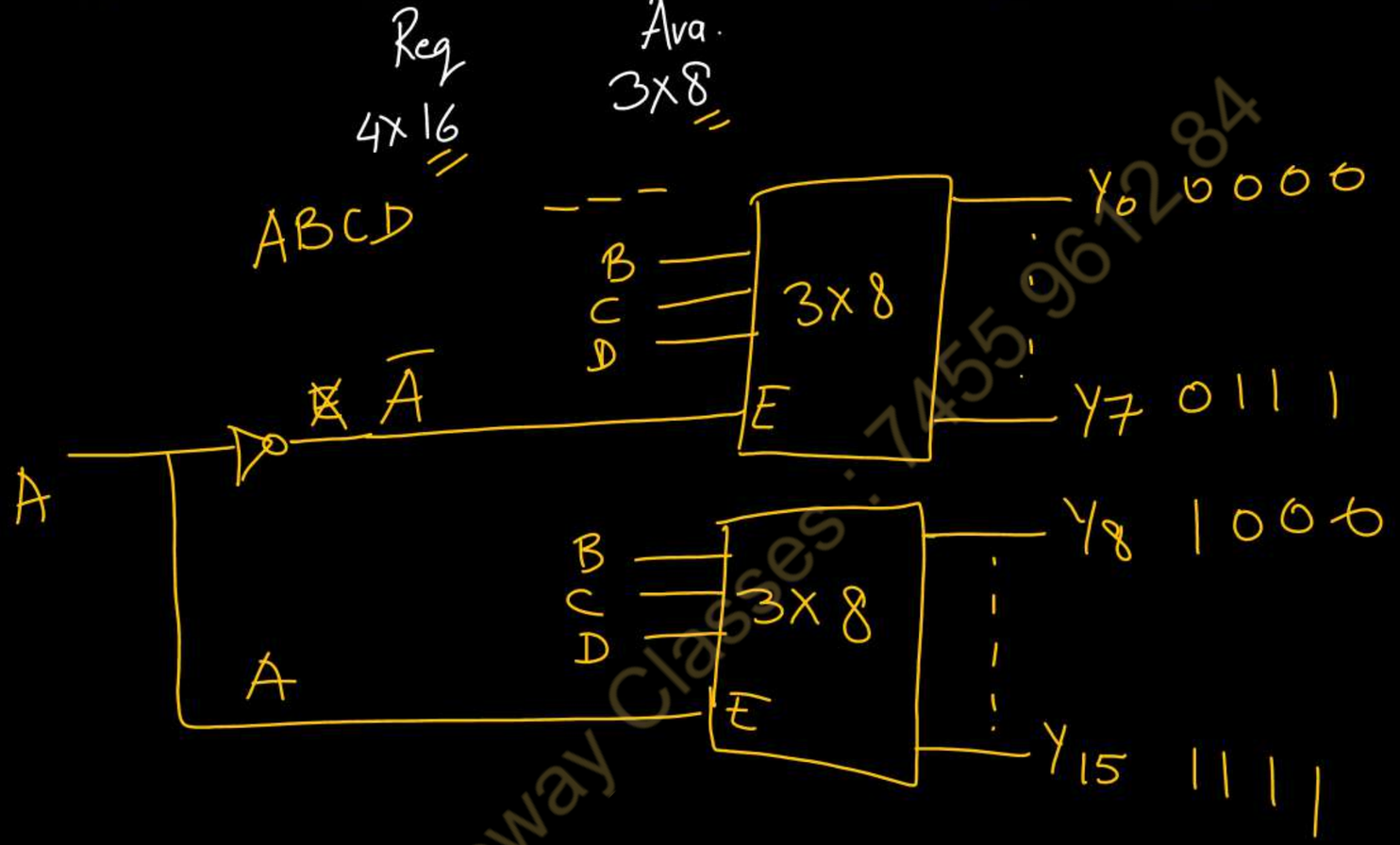
Implementing 3×8 line Decoder by using 2×4 decoders



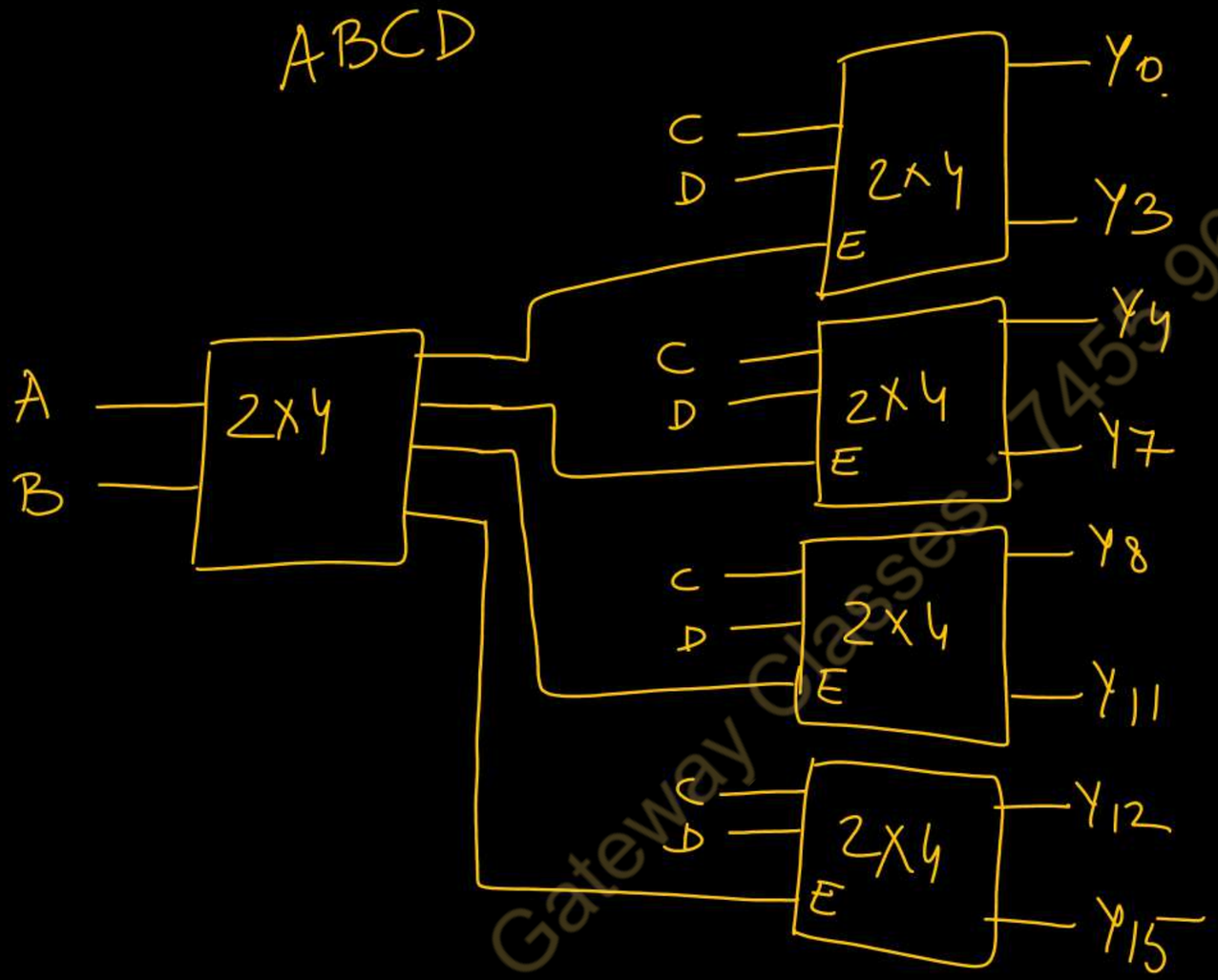
Req.
3×8
A B C

Available
2×4
B C

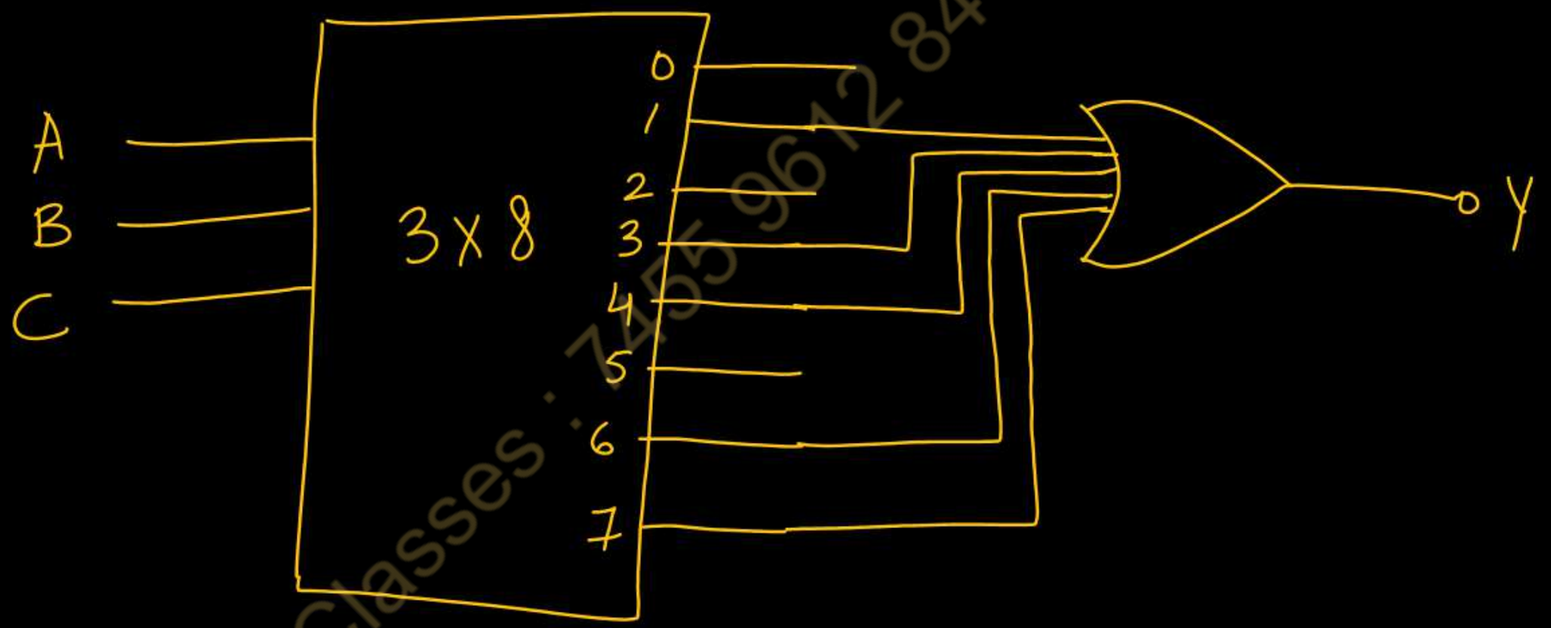
Implementing 4×16 line Decoder by using 3×8 decoders



4×16 line Decoder using 2×4 decoders



Implement the logic expression $Y=\Sigma m(1,3,4,6,7)$ using decoder.

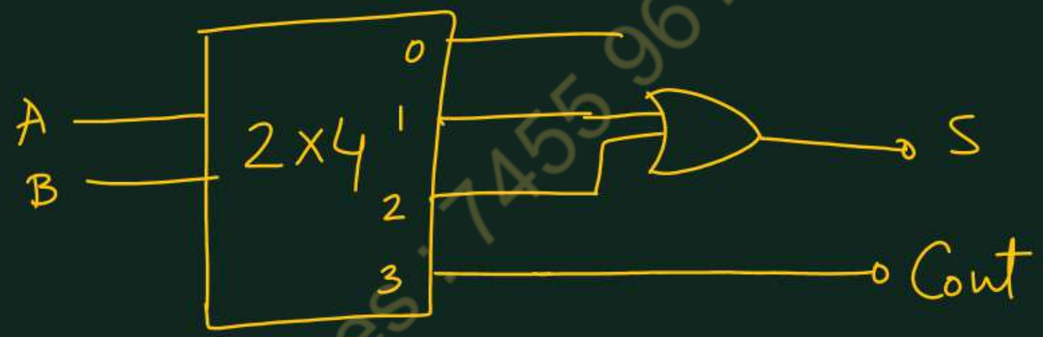


Half Adder using Decoder

A	B	S	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = \Sigma m(1, 2)$

$Cout = \Sigma m(3)$

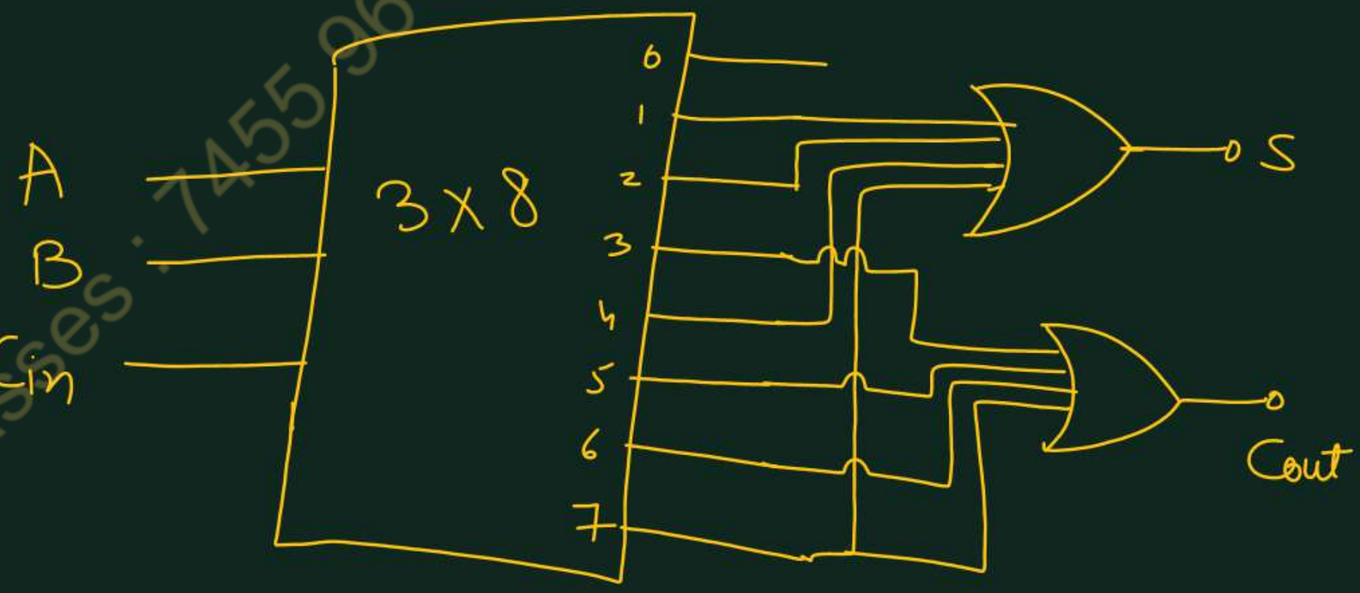


Full Adder

A	B	C_{in}	S	C_{out}
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

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

$C_{out} = \Sigma m(3, 5, 6, 7)$



DPP

a) Implement the logic expression $Y = \Sigma m(0,2,3)$ using decoder.

b) Implement the half adder using decoder.

DIGITAL SYSTEM DESIGN

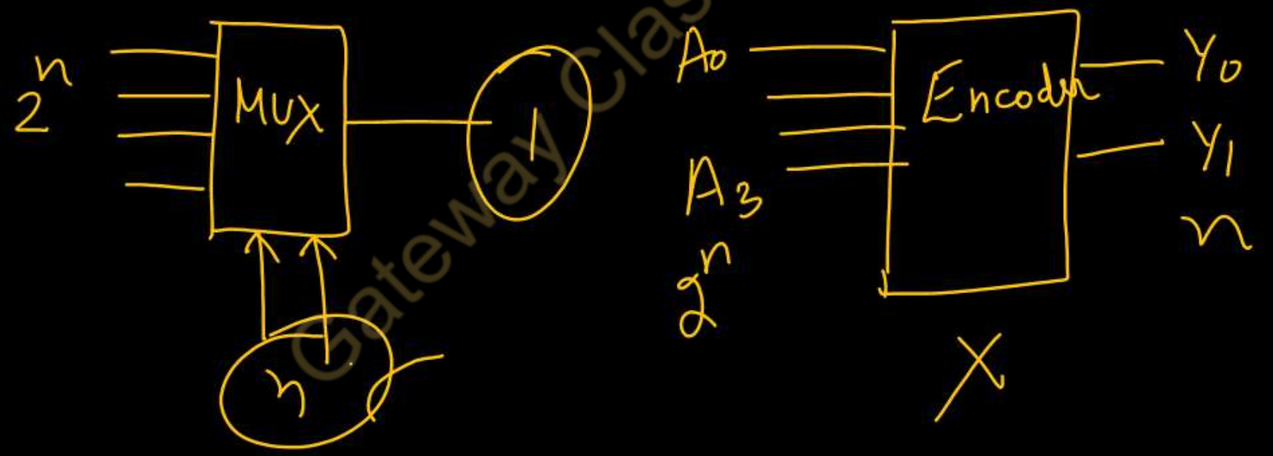
UNIT 2 : Combinational circuits

Today's Target

- Digital Encoder
- 4×2, 8×3, Priority Encoders
- University Questions

UNIVERSITY QUESTIONS

YEAR	QUESTION	MARKS
22-23	What is the difference between Multiplexer and Encoder	2 marks
18-19	Write short notes on priority encoder.	7 marks

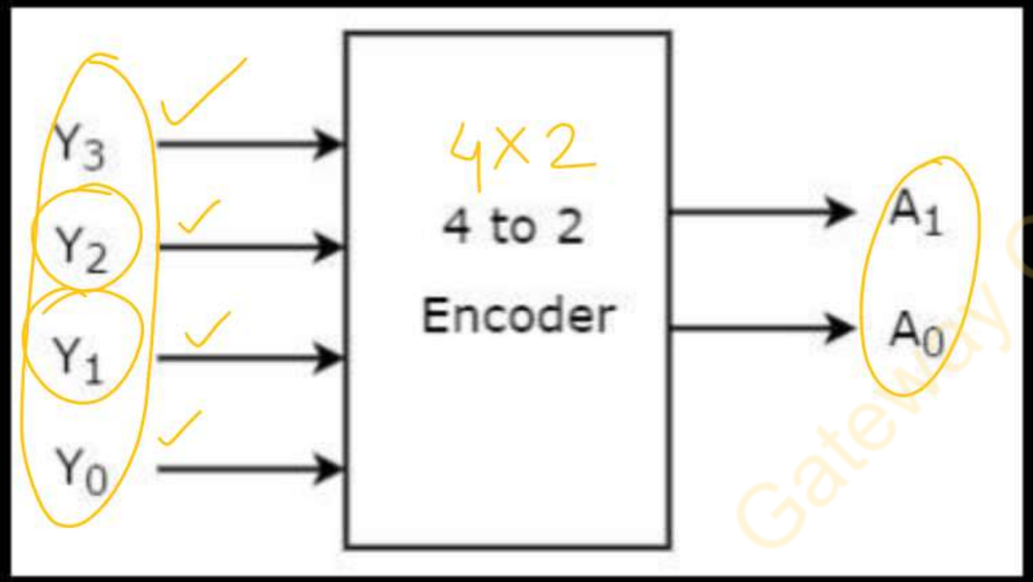


Digital Encoder

- a combinational circuit that generates a specific code at its outputs such as binary or BCD in response to one or more active inputs.
- reverse operation of Decoder
- 2^n input lines
- enable signal is optional in encoders

4 to 2 Encoder

- four inputs Y_3, Y_2, Y_1 & Y_0
- two outputs A_1 & A_0
- **block diagram** is shown
- At any time, only one of these 4 inputs can be '1' in order to get the respective binary code at the output.



Truth table

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

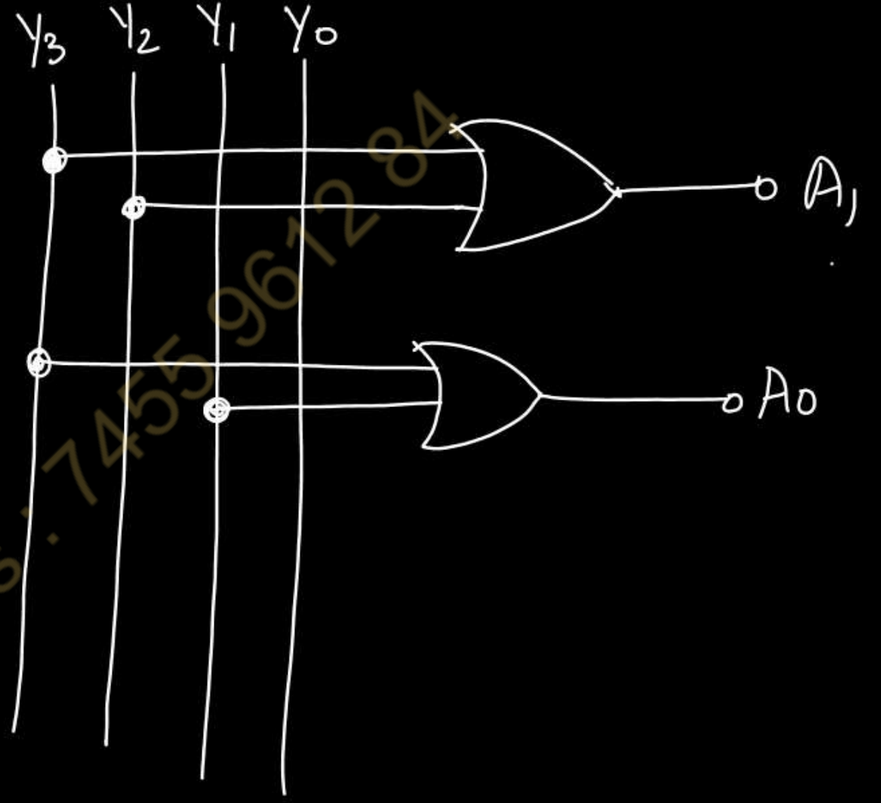
Boolean Equations

$A_1 = Y_3 + Y_2$

$A_0 = Y_3 + Y_1$

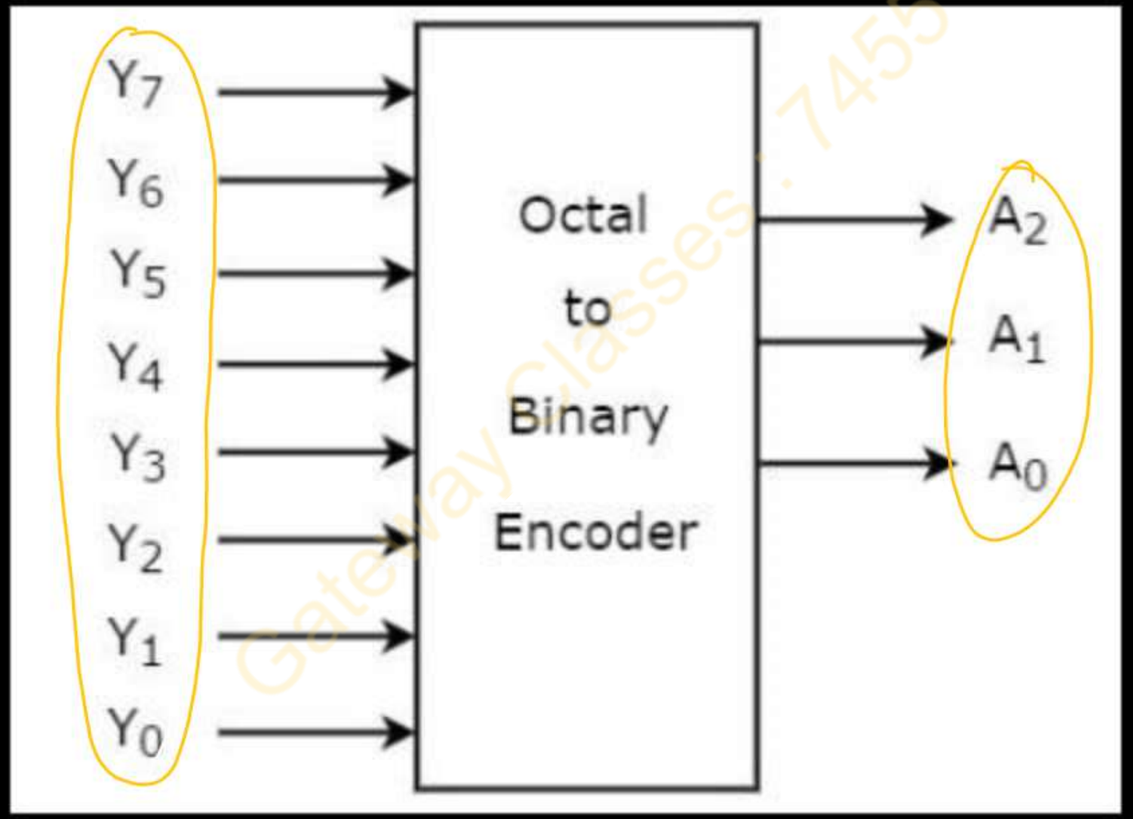
Implementation of 4 to 2 Encoder

$$A_1 = Y_3 + Y_2$$
$$A_0 = Y_3 + Y_1$$



Octal to Binary Encoder

- eight inputs, Y_7 to Y_0
- three outputs A_2, A_1 & A_0
- **block diagram** of octal to binary Encoder



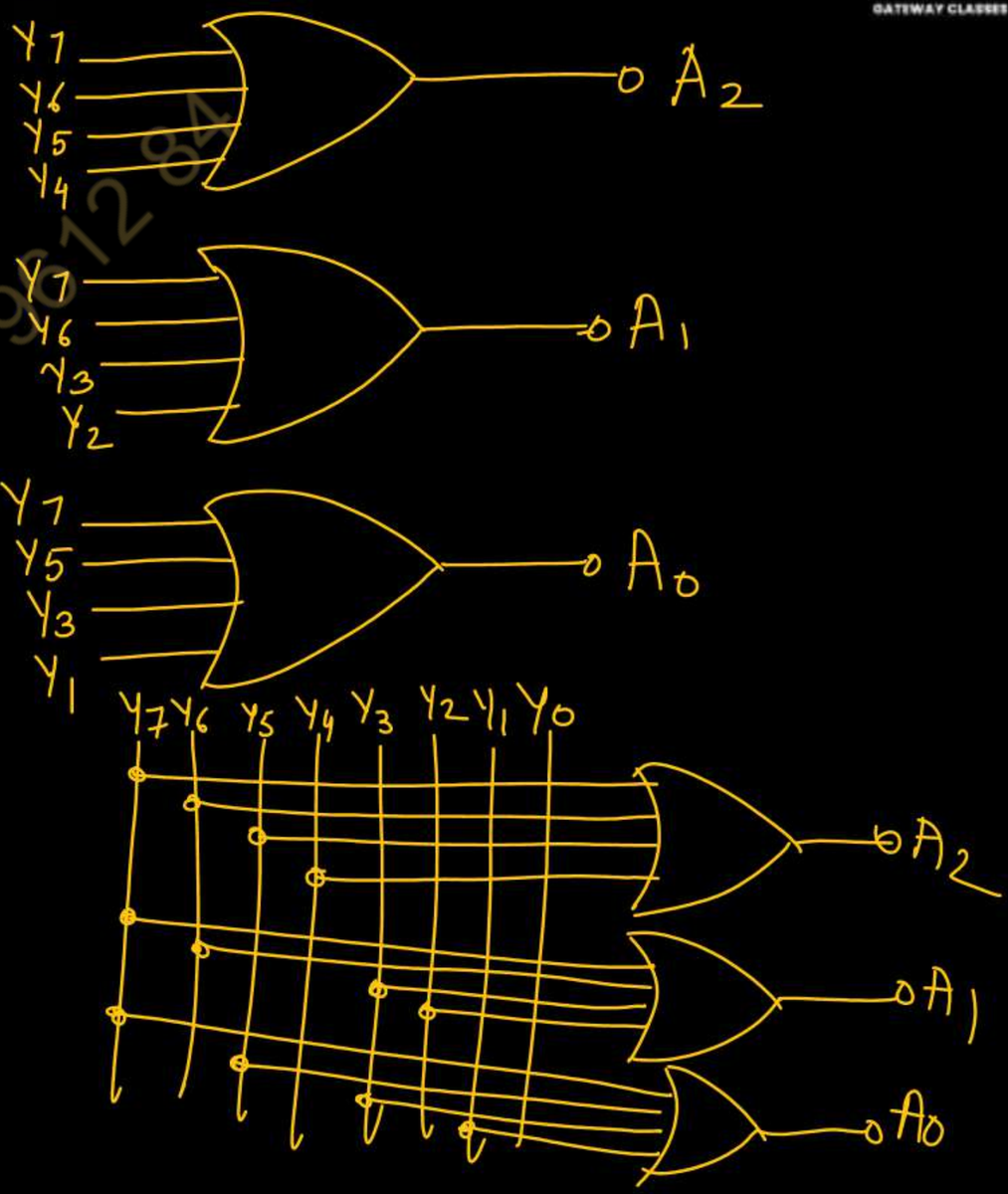
Truth table & Boolean Equations

INPUTS								OUTPUTS		
Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	A2	A1	A0
1	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0

$A_2 = Y_7 + Y_6 + Y_5 + Y_4$

$A_1 = Y_7 + Y_6 + Y_3 + Y_2$

$A_0 = Y_7 + Y_5 + Y_3 + Y_1$



Priority Encoder

- 4 to 2 priority encoder
- four inputs Y_3, Y_2, Y_1 & Y_0
- two outputs A_1 & A_0
- Y_3 has the highest priority
- Y_0 has the lowest priority
- if more than one input is '1' at the same time, the output will be the binary code corresponding to the input, which is having **higher priority**.
- one more **output, V** ensures the code available at outputs is valid or not.
- If at least one input of the encoder is '1', then V=1
- If all the inputs of encoder are '0', then V=0

Truth table of 4 to 2 priority encoder

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

K-maps for A1 & A0

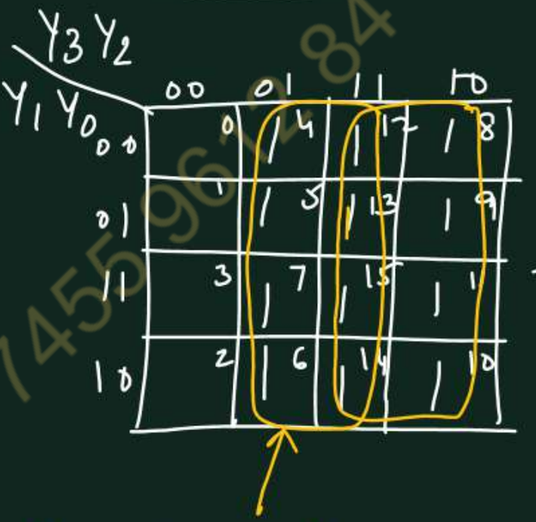
$A_1 = Y_2 + Y_3$

$A_0 = Y_3 + \overline{Y}_2 Y_1$

$V = Y_3 + Y_2 + Y_1 + Y_0$

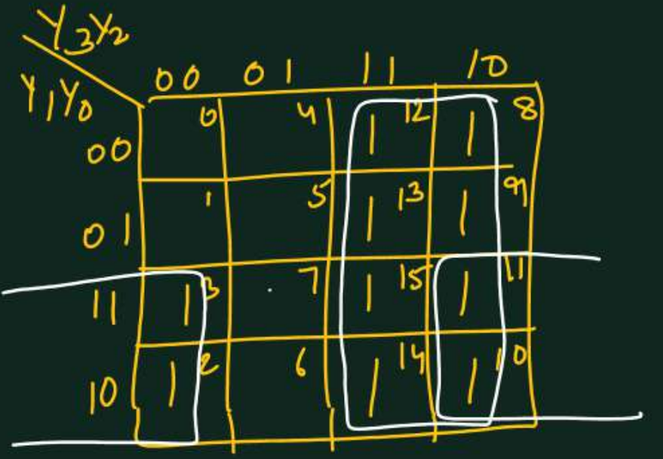
I/P						
Y_3	Y_2	Y_1	Y_0	A_1	A_0	V
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	1	1	1
0	1	0	0	1	0	1
0	1	0	1	1	1	1
0	1	1	0	1	1	1
0	1	1	1	1	0	1
1	0	0	0	1	0	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	0	1
1	1	0	0	1	0	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	0	1

for A_1



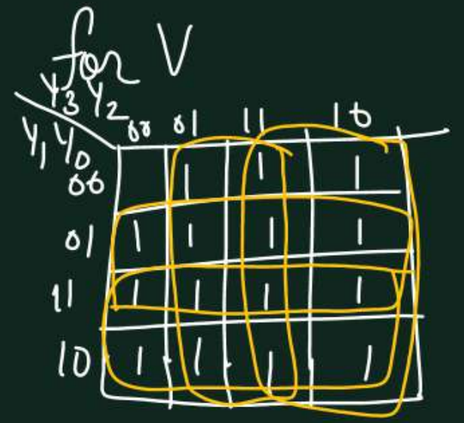
$A_1 = Y_2 + Y_3$

for A_0



$A_0 = Y_3 + \overline{Y_2} Y_1$

for V



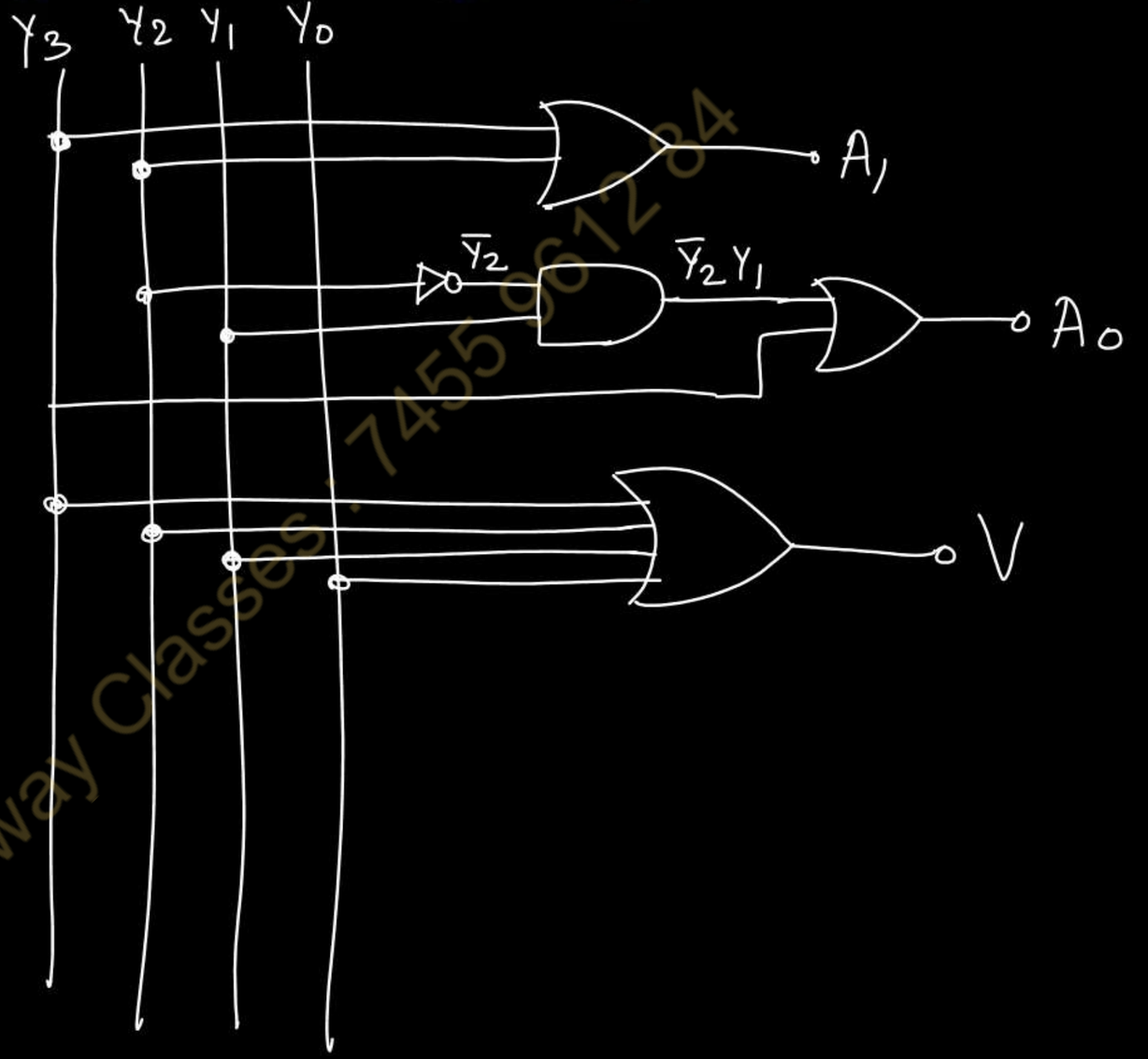
$V = Y_3 + Y_2 + Y_1 + Y_0$

Circuit Diagram of 4 to 2 priority encoder

$A_1 = Y_3 + Y_2$

$A_0 = Y_3 + \bar{Y}_2 Y_1$

$V = Y_3 + Y_2 + Y_1 + Y_0$



DIGITAL SYSTEM DESIGN

UNIT 2 : Combinational circuits

Today's Target

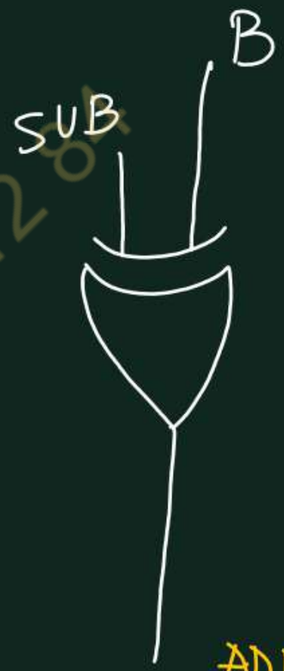
- Binary Adder & Subtractor ✓
- Decimal or BCD Adder ✓

Binary Adder & Subtractor

- Binary addition and subtraction can be done using single circuit.
- This can reduce the cost and space.
- Subtraction is done by 2's compliment method.
- An extra input 'sub' is introduces such that;
- for addition: sub = 0
- For subtraction: sub = 1

$$\begin{array}{rr} 1 & 1001 \\ 11 & -\ 0011 \\ \hline & \end{array} \qquad \begin{array}{r} 9 \\ -\ 3 \\ \hline +6 \end{array}$$

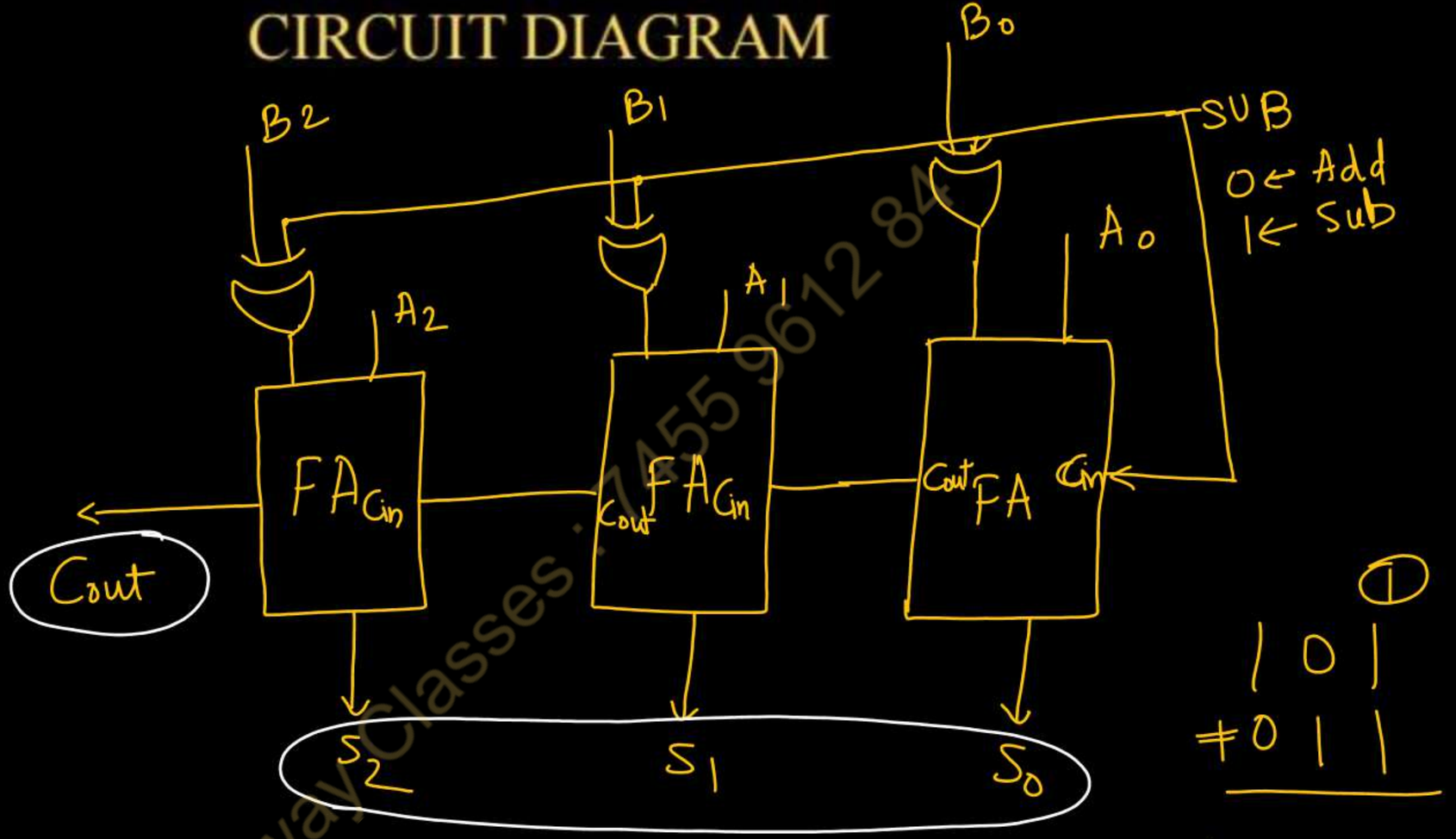
$$\begin{array}{rr} 1 & 1001 \\ 2's & 1101 \\ \hline & 11\,0110 \end{array} \qquad \begin{array}{rr} & 0011 \\ 1's & 1100 \\ +* & \quad\ 1 \\ \hline 2's & 1101 \end{array}$$



	SUB	B	o/p
ADD	0	0	0
	0	1	1
SUB	1	0	1
	1	1	0

1's

CIRCUIT DIAGRAM



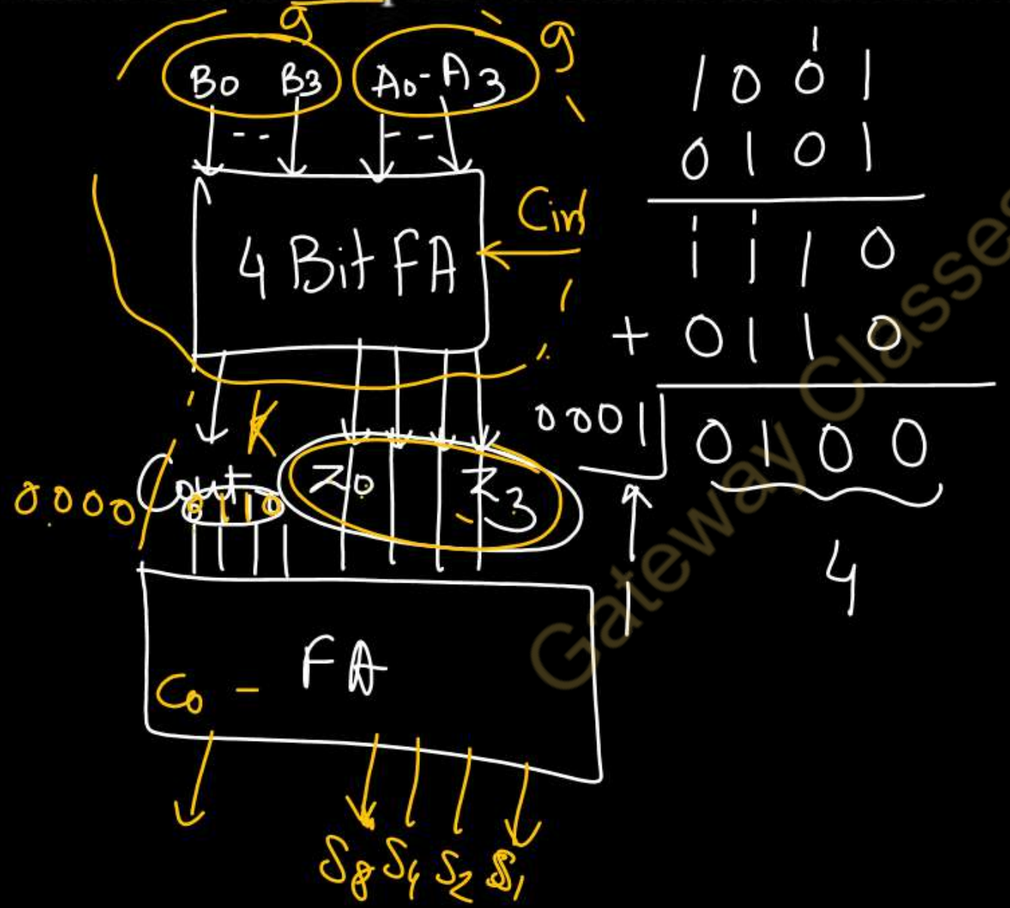
$$
\begin{array}{r}
1\ 0\ 1 \\
+\ 0\ 1\ 1 \\
\hline
\end{array}
\qquad (1)
$$

$$
\begin{array}{rl}
1\text{'s} & 1\ 0\ 0 \\
 & +\quad (1) \\
\hline
2\text{'s} & 1\ 0\ 1
\end{array}
$$

$$
\begin{array}{r}
1\ 0\ 1 \\
-\ 0\ 1\ 1 \\
\hline
\end{array}
$$

Decimal or BCD Adder

- The BCD-Adder is used in the computers and the calculators that perform arithmetic operation directly in the decimal number system.
- The BCD-Adder accepts the binary-coded form of decimal numbers.
- The Decimal-Adder requires a minimum of nine inputs and five outputs.



$$\begin{array}{r} 1001 \\ 0101 \\ \hline 1110 \\ +0110 \\ \hline 0001\ 0100 \end{array} \qquad \begin{array}{r} 9 \\ +5 \\ \hline 14 \end{array}$$

4

0–9 0000 : 1001

Invalid BCD

1010
1011
1100
1101
1110
1111

Binary Sum					BCD Sum					Decimal
K	Z8	Z4	Z2	Z1	C	S8	S4	S2	S1	Result
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10

$$D = K + Z_8Z_4 + Z_8Z_2$$

Binary Sum					BCD Sum					Decimal
K	Z8	Z4	Z2	Z1	C	S8	S4	S2	S1	
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

16 8 4 2 1

10000

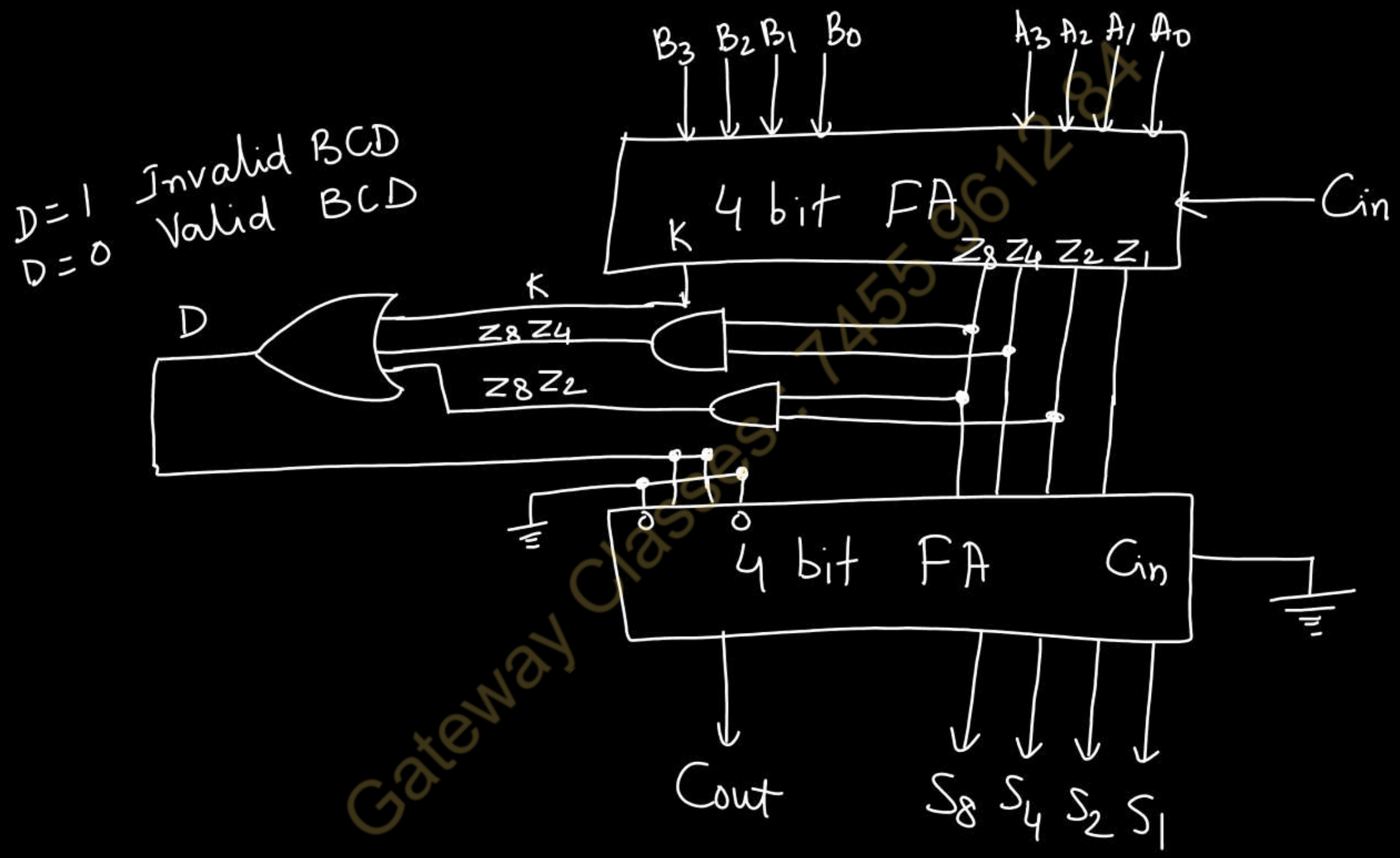
10

0001 0000

Boolean Expression for Cout

$$\text{Cout} = K + Z_8 Z_4 + Z_8 Z_2$$

Circuit Diagram for BCD Adder



AKTU Full Courses (Paid)

Download Gateway Classes Application

From Google Play store

All Subjects

Link in Description

Thank You