AKTU Festival Offer

GATEWAY CLASSES

B.Tech II-Year EC & EC Allied

DSD : Digital System Design

Unit-1 Lec-1



Avinash Sir

Today's target

- Introduction to Number System
- Basic of Number System Conversion
- PYQ
- DPP

Gateway Classes

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

UNIT-1 : Logic simplification and combinational logic design

- Number Systems
- Binary arithmetic
- Signed magnitude representation
- Binary codes, code conversion
- Review of Boolean algebra and Demorgans theorem
- SOP & POS forms
- Canonical forms
- Karnaugh maps up to 5 variables
- Tabulation method.

Gateway Classes

Number system : A number system is a way to represent and express numbers using a set of symbols (digits). The most commonly used number systems in digital technology are

(1) Binary → Base-2 (0,1)

(2) Decimal

(3) Octal

(4) Hexadecimal

Base or radix:

The number of symbols used in a number system is called base or radix of the number system.

Eg: In decimal number system – Base is 10

In Binary number system – Base is 2

In Hexadecimal number system – Base is 16 etc

Gateway Classes

Number System	Base / Radix	Digits Used	Example
Binary	2	0, 1	$(101111)_2$
Octal	8	0, 1, 2, 3, 4, 5, 6, 7 (0-7)	$(47525)_8$
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	$(316875)_{10}$
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	$(9B21A)_{16}$

NOTE : A = 10, B = 11, C = 12, D = 13, E = 14, F = 15

Gateway Classes

Common Terminology in Digital Electronics

2 Nibble = 8 bit = 1 Byte

1 word = 2Byte = 16 bit = 4 Nib

1 D.W = 2word = 4 Byte = 32 bit

Bit: A binary digit is called a bit. Example: 0 or 1

Nibble: A group of four bits is called a Nibble. Example: 0110, 1110 etc.

Byte: A group of eight bits is called a byte. Example: 01101101, 11010011 etc.

MSB LSB

Word:- A group of two bytes (16 bits) is called a Word.

Double word:- A group of two words (4 bytes or 32 bits) is called a double Word.

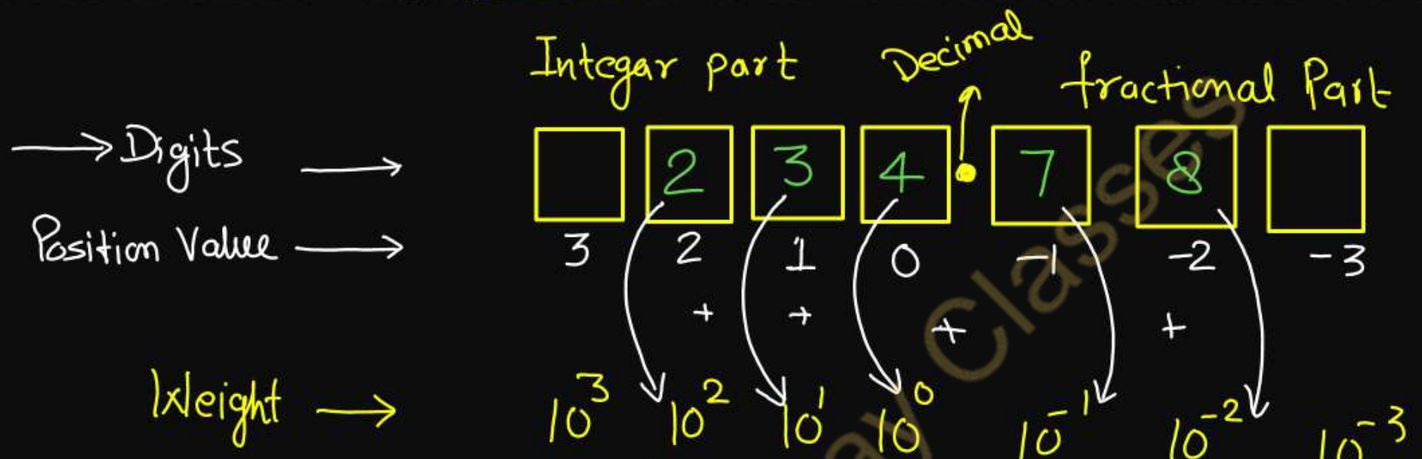
Most significant bit (MSB): The left most bit of a binary number which has the highest place value is called most significant bit.

Least significant bit (LSB): The right most bit of a binary number which has the least place value is called least significant bit.

Gateway Classes

Decimal Number system

Decimal number system is the commonly used number system. It is known as base (or radix) ten systems because it has ten digits 0 to 9 that can be used to represent any number.



Conversion to Decimal

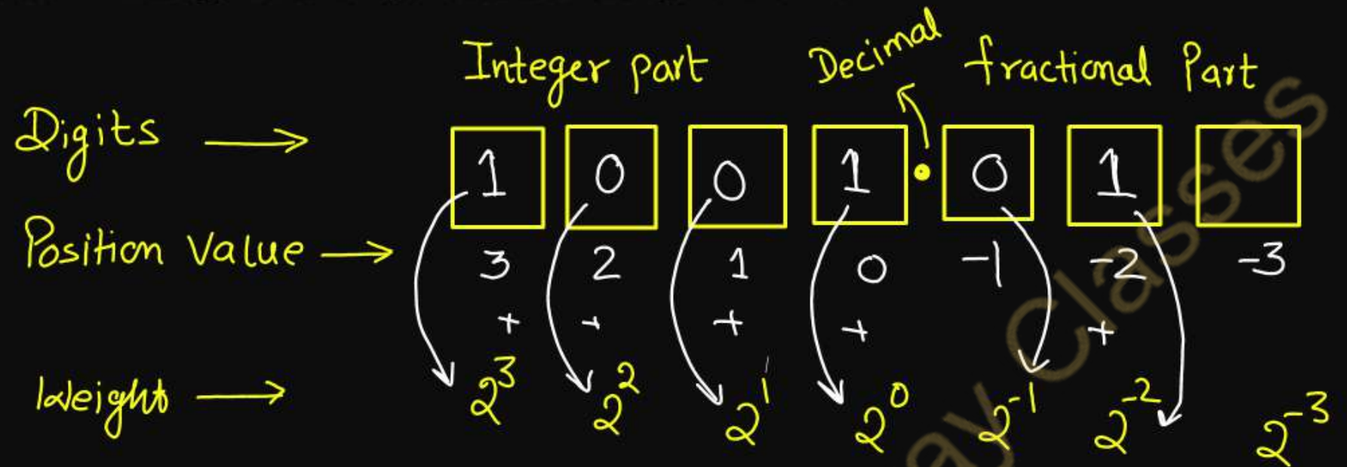
Example :- $234.78 = (2 \times 10^2) + (3 \times 10^1) + (4 \times 10^0) + (7 \times 10^{-1}) + (8 \times 10^{-2})$

$= 200 + 30 + 4 + 0.7 + 0.08$

Gateway Classes

Binary Number system

The binary number has a radix of 2. As r = 2, only two digits are needed, and these are 0 and 1. In binary system weight is expressed as power of 2.

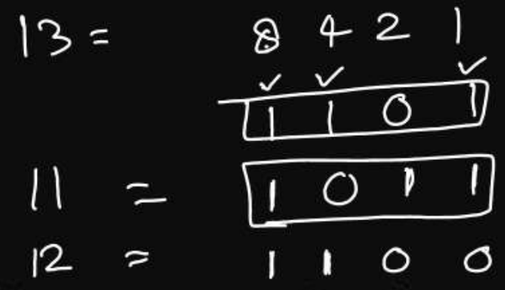


Example :- $(1001.01)_2 = [(1) \times 2^3] + [(0) \times 2^2] + [(0) \times 2^1] + [(1) \times 2^0] + [(0) \times 2^{-1}] + [(1) \times 2^{-2}]$

$(1001.01)_2 = [1 \times 8] + [0 \times 4] + [0 \times 2] + [1 \times 1] + [0 \times 0.5] + [1 \times 0.25]$

$(1001.01)_2 = (9.25)_{10}$

Gateway Classes



Octal Number system

(0-7)

Octal and hexadecimal, are used for representing large binary numbers. Octal systems use a base or radix of 8. It uses first eight digits(0- 7) of decimal number system.

Hexa Decimal Number System

(0-9) (A-F) = 16

The hexadecimal numbering system has a base of 16. There are 16 symbols. The decimal digits 0 to 9 are used as the first ten digits as in the decimal system, followed by the letters A, B, C, D, E and F, which represent the values 10, 11,12,13,14 and 15 respectively.

Decimal	Binary (2^3 2^2 2^1 2^0 = 8 4 2 1)	octal	Hexadec.
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Gateway Classes

Methods of Number system conversion

- Universal Method of Number system conversion



- Easy Method for some Number system conversion

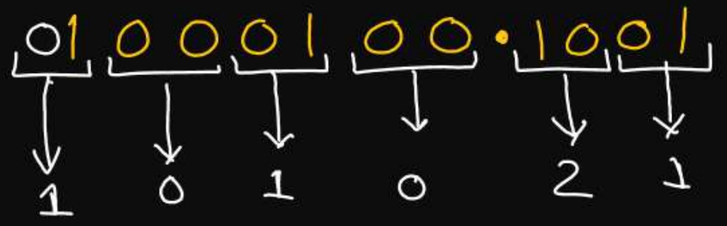
Base	Binary	Base
4, 8, 16	2	4, 8, 16

Base	Group of binary digits
4	2
8	3
16	4

Gateway Classes

(1) Binary to base 4 conversion

Example : $(1000\ 100\ .\ 1001)_2 = (?)_4$



$(1010.21)_4$

(1) Binary to base 4 conversion

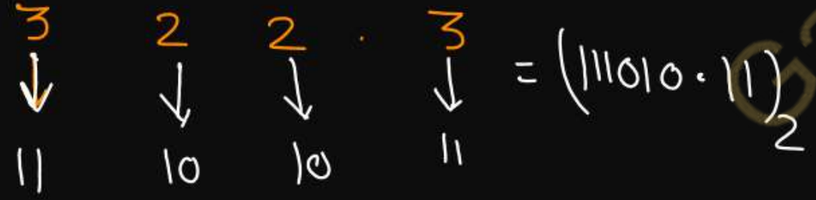
Example : $(110101\ .\ 10)_2 = (?)_4$



$(311.2)_4$

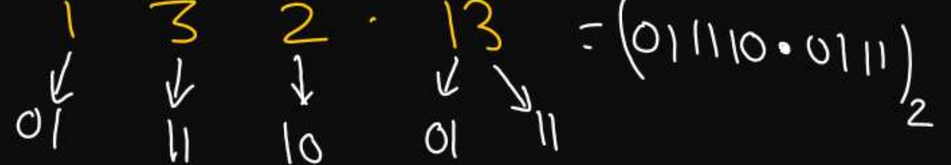
(2) base 4 to binary conversion

Example : $(322.3)_4 = (?)_2$



(2) base 4 to binary conversion

Example : $(132.13)_4 = (?)_2$



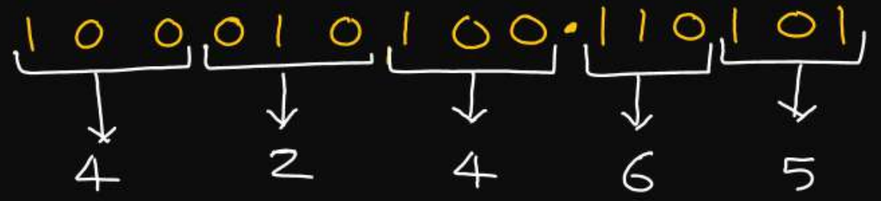
Gateway Classes

$2^3 = 2^n \Rightarrow n = 3$

AKTU-2024-25

(1) Binary to octal conversion

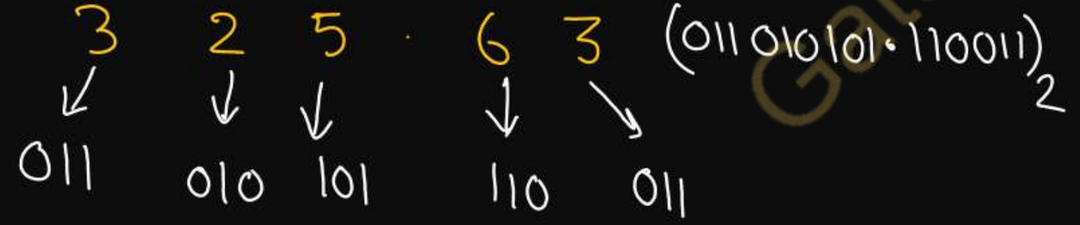
Example : $(1000\ 10100\ .\ 110101)_2 = (?)_8$



$(424.65)_8$

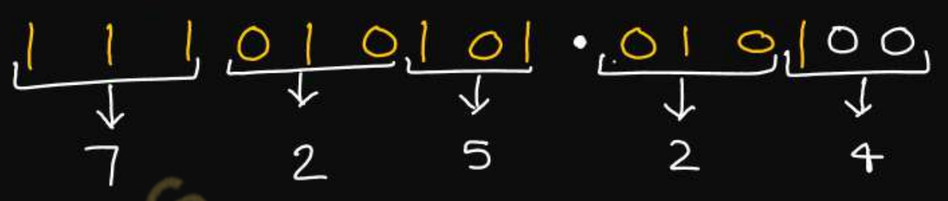
(2) Octal to binary conversion

Example : $(325.63)_8 = (?)_2$



(1) Binary to octal conversion

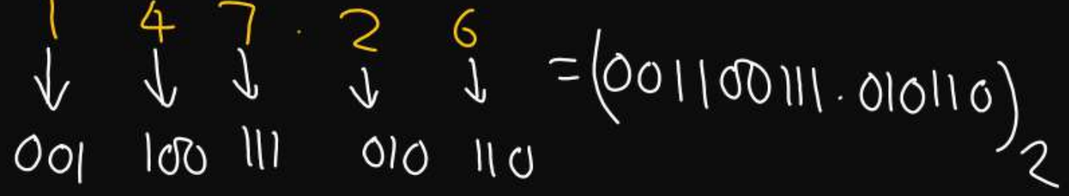
Example : $(1110\ 10101\ .\ 0101)_2 = (?)_8$



$(725.24)_8$

(2) Octal to binary conversion

Example : $(147.26)_8 = (?)_2$



Gateway Classes

AKTU-2024-25

(3) Hexadecimal to binary conversion

Example : $(ABC)_{16} = (?)_2$

A B C

1010 1011 1100

$(101010111100)_2$

Example : $(2F3\blacksquare)_{16} = (?)_2$

AKTU-2023-24

2 F 3

0010 1111 0011

$(001011110011)_{16}$

2^4

AKTU-2023-24

Hexadecimal to binary conversion

Example : $(A3B\blacksquare)_{16} = (?)_2$

A 3 B

1010 0011 1011

$(1010001110 11)_{16}$

(4) Binary to Hexadecimal conversion

Example : $(1000\ 10100\ .\ 110101)_2 = (?)_{16}$

0001 0001 0100 . 1101 0100

1 1 4 13(D) 4

$(114 \cdot D4)_{16}$

Gateway Classes

(5) Octal to Hexadecimal conversion

Easy Method : Octal → Binary → Hexadecimal

Example : $(642)_8 = (?)_{16}$

6 4 2

110 100 010 ↷ Binary

0001 1010 0010 ↷ Hexa

1 A 2

$(1A2)_{16}$

(6) Hexadecimal to octal

Easy Method : Hexadecimal → Binary → Octal

Example : $(1AF.BD)_{16} = (?)_8$

1 A F . B D ↷ B

0001 1010 1111 1011 1101

001 101 011 111 . 101 111 010

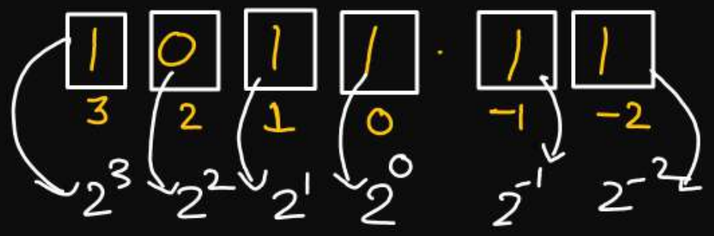
6 5 7 5 7 2

$(657.572)_8$

Gateway Classes

Any base to decimal conversion

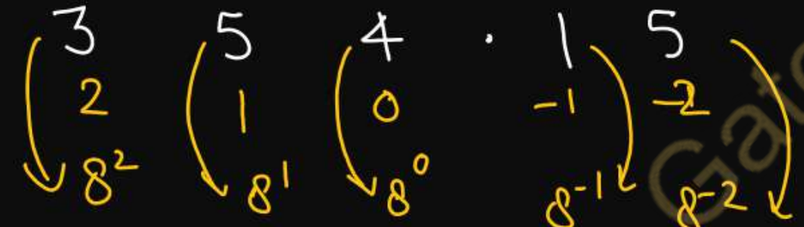
(1) Binary to decimal : Example $(1011.11)_2 = (?)_{10}$



$(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) + (1 \times 2^{-1}) + (1 \times 2^{-2})$

$8 + 0 + 2 + 1 + 0.5 + 0.25 = (11.75)_{10}$ Ans.

(2) Octal to decimal : Example $(354.15)_8 = (?)_{10}$



$(3 \times 8^2) + (5 \times 8^1) + (4 \times 8^0) + (1 \times 8^{-1}) + (5 \times 8^{-2})$

$= 192 + 40 + 4 + 0.125 + 0.078125$

$= (236.203125)_{10}$

Gateway Classes

(3) Hexadecimal to decimal : Example $(AB2.2C)_{16} = (?)_{10}$

(10) A (11) B 2 . 2 C (12)

2 1 0 -1 -2

16^2 16^1 16^0 16^{-1} 16^{-2}

$(10 \times 16^2) + (11 \times 16^1) + (2 \times 16^0) + (2 \times 16^{-1}) + (12 \times 16^{-2})$

$(2560) + (176) + 2 + 0.125 + 0.046875$

$= (2738.171875)_{10}$

(4) Other conversion : Example $(213.23)_4 = (?)_{10}$

$(2\ 1\ 3 . 2\ 3)$

$4^2\ 4^1\ 4^0\ 4^{-1}\ 4^{-2}$

$= (2 \times 4^2) + (1 \times 4^1) + (3 \times 4^0) + (2 \times 4^{-1}) + (3 \times 4^{-2})$

$= 32 + 4 + 3 + 0.5 + 0.1875 = (39.6875)_{10}$

Gateway Classes

(5) Other conversion : Example $(123)_9 = (?)_{10}$

AKTU-2023-24

1	2	3	Digits.
2	1	9	P.V.
9^2	9^1	9^0	weight

$(1 \times 9^2) + (2 \times 9^1) + (3 \times 9^0)$

$81 + 18 + 3$

$(102)_{10}$

Gateway Classes

Decimal to any Base conversion

(1) Decimal to Binary

Example : $(319.6875)_{10} = (?)_2$

I f

Integer part

		Remainder
2	319	
2	159	1
2	79	1
2	39	1
2	19	1
2	9	1
2	4	1
2	2	0
2	1	0
	0	1

Fractional part

$0.6875 \times 2 = 1.375$ → 1

$0.375 \times 2 = 0.75$ → 0

$0.75 \times 2 = 1.50$ → 1

$0.50 \times 2 = 1.00$ → 1

$= (100111111.1011)_2$

Gateway Classes

AKTU-2023-24

(2) Decimal to Octal : Example $(378.93)_{10} = (?)_8$

		R
8	378	
8	47	2
8	5	7
	0	5

(572)

0.93 × 8	= 7.44	7
0.44 × 8	= 3.52	3
0.52 × 8	= 4.16	4
0.16 × 8	= 1.28	1
0.28 × 8	= 2.24	2

(73412)

$= (572.73412)_8$

AKTU-2023-24,22-23

(3) Decimal to Hexadecimal :

Example (i) $(256)_{10} = (?)_{16}$

16	256	
16	16	0
16	1	0
	0	1

$= (100)_H$

		R
16	454	
16	28	6
16	1	12 → C
	0	1

(1C6)

Example (i) $(454.523)_{10} = (?)_{16}$

0.523 × 16	= 8.368	8
0.368 × 16	= 5.888	5
0.888 × 16	= 14.208	14 (E)
0.208 × 16	= 3.328	3

$= (1C6.85E3)_H$

Gateway Classes

(4) Other conversion : Example $(2230)_4 = (?)_3$

2 2 3 0

3 2 1 0

4^3 4^2 4^1 4^0

Decimal.

① 10 → 172 → ③

$(2 \times 4^3) + (2 \times 4^2) + (3 \times 4^1) + (0 \times 4^0)$

$128 + 32 + 12 + 0$

$= (172)_{10}$

		R
3	172	
3	57	1
3	19	0
3	6	1
3	2	0
	0	2

$= (20101)_3$

Gateway Classes

UNIT-1 DPP-1

Q. Convert the following

(i) Binary $(111101110)_2$ to Octal — Ans:- $(756)_8$

(ii) $(5A9.B4)_{16} = (?)_2$ — Ans:- $(10110101001.10110100)_2$

(iii) $(3102.12)_4$ to decimal — Ans:- $(210.375)_{10}$

(iv) Decimal number 37 to binary — Ans:- 100101

(v) Decimal number 24.6 to binary — Ans:- 11000.10011

(vi) Decimal number 35.45 to octal — Ans:- $(43.34631)_8$

Gateway Classes

AKTU Festival Offer

GATEWAY CLASSES

B.Tech II-Year EC & EC Allied

DSD : Digital System Design

Unit-1 Lec-2



Avinash Sir

Today's target

- Binary arithmetic
- PYQ
- DPP

Gateway Classes

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

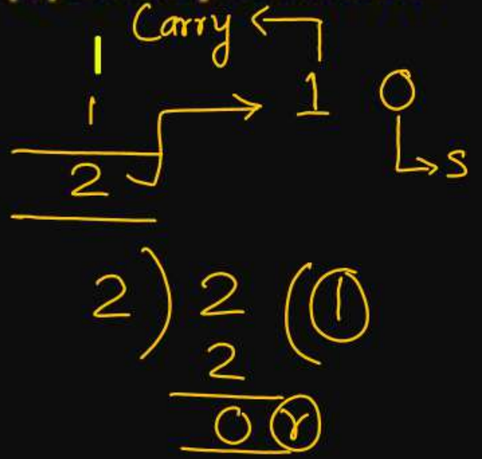
UNIT-1 : Logic simplification and combinational logic design

- Number Systems
- Binary arithmetic
- Signed magnitude representation
- Binary codes, code conversion
- Review of Boolean algebra and Demorgans theorem
- SOP & POS forms
- Canonical forms
- Karnaugh maps up to 5 variables
- Tabulation method.

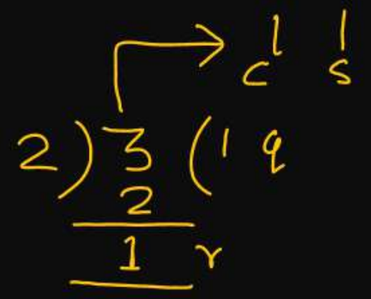
Gateway Classes

Binary Arithmetic

(1) Binary Addition



	SUM	CARRY
0 + 0 =	0	0
0 + 1 =	1	0
1 + 0 =	1	0
1 + 1 =	0	1
1 + 1 + 1 =	1	1



Note:

(i) 1 + 1 = 10

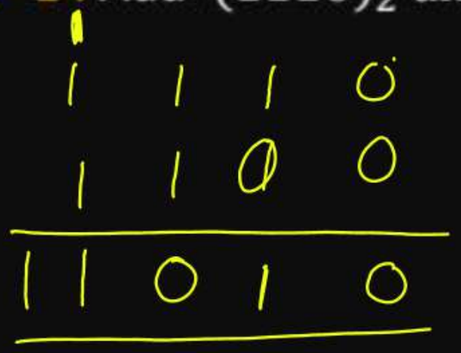
SUM = 0 Carry = 1

(ii) 1 + 1 + 1 = 11

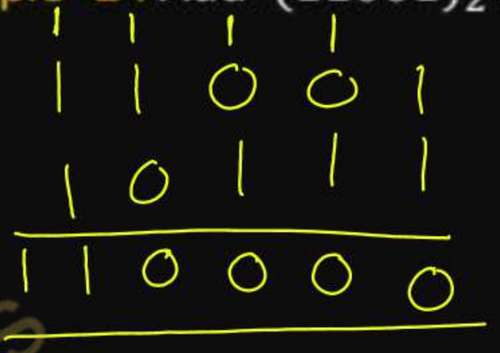
SUM = 1 Carry = 1

Gateway Classes

Example - 1 : Add $(1110)_2$ and $(1100)_2$



Example -2 : Add $(11001)_2$ and $(10111)_2$



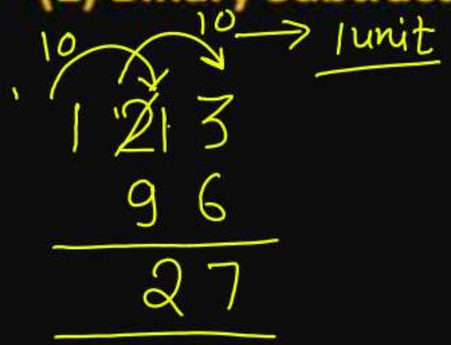
Example - 3 : Add binary number $(1110.110)_2$ to $(1100.011)_2$



Gateway Classes

Binary Arithmetic

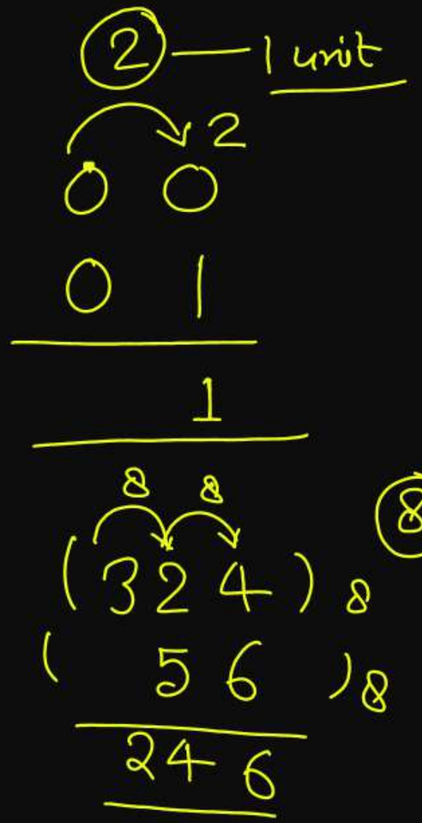
(1) Binary Subtraction



	Subtraction	Borrow
0 - 0 =	0	0
1 - 0 =	1	0
1 - 1 =	0	0
0 - 1 =	1	1

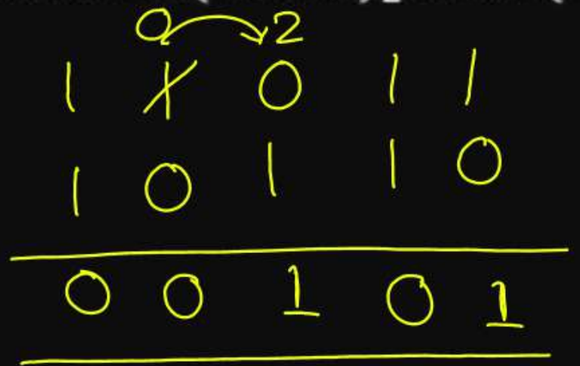
Note:

$$0 - 1 = 10 - 1$$
$$= 2 - 1$$
$$= 1$$

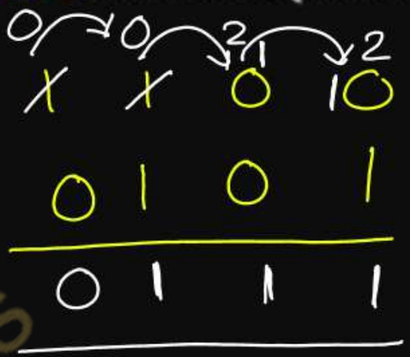


Gateway Classes

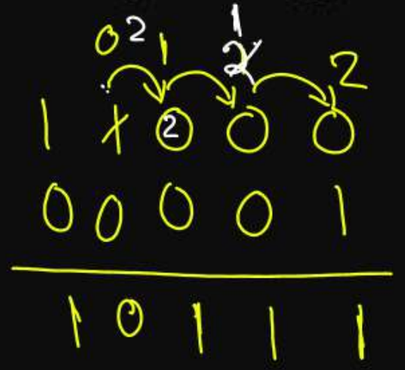
Example -1 : Subtract $(10110)_2$ from $(11011)_2$



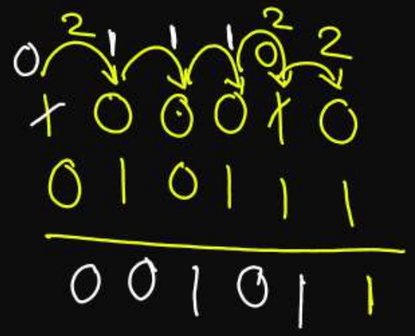
Example -3 : Subtract $(0101)_2$ from $(1100)_2$



Example - 2 : Subtract $(00001)_2$ from $(11000)_2$



Example - 4 : Subtract $(010111)_2$ from $(100010)_2$



Gateway Classes

Complements

Complements are used in the digital computers in order to simplify the subtraction operation and for the logical manipulation

r - Complement → 2's 8's 10's 16's

(r-1) - Complement → 1's 7's 9's 15's

Binary Octal Decimal Hex decimal

3AB

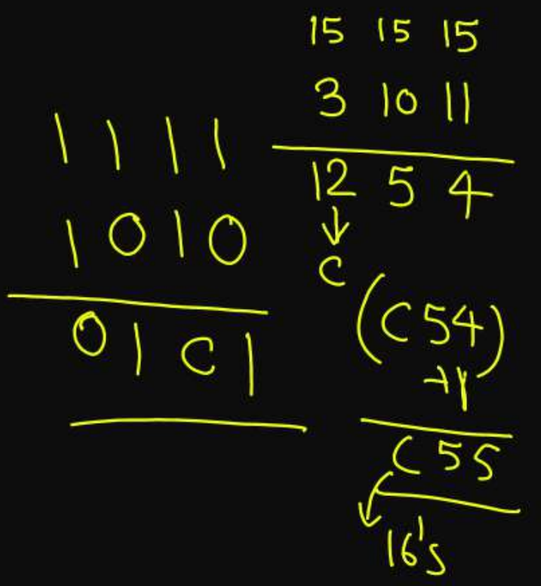
For Binary System

(i) 1' S complement

(ii) 2 ' S complement

1' S complement





Given Number

1010

11010101

1001

1' S complement

0101 ✓

00101010

0110

Gateway Classes

2' S complement

2' S complement = 1' S complement + 1

Given	1' S complement	2' S complement
1011	0100	0100
0100		1
		0101 → 2's Complement



10110010 → 01001101

01001101
+1
01001110 2's Complement

Gateway Classes

Binary Subtraction 1' S complement

$A - B = A + (-B)$ ← (-) → Complement

STEP – 1 : Find the 1'S complement of the number to be subtracted

STEP – 2 : Perform addition

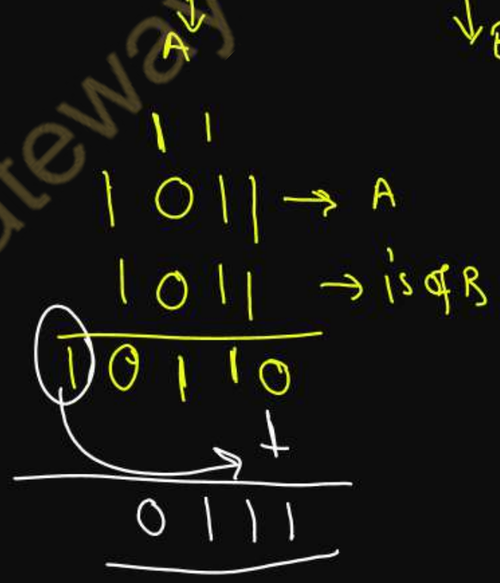
STEP – 3 : CASE – 1 : If the final carry is 1, then add it to the result obtained in step-2

CASE – 2 : If the final carry is zero, result obtain in step-2 is negative and in the 1'S complement form

Example – 1 : Subtract using 1' S complement $(1011)_2 - (0100)_2$

B = 0100

1's Complement of B = 1011



Gateway Classes

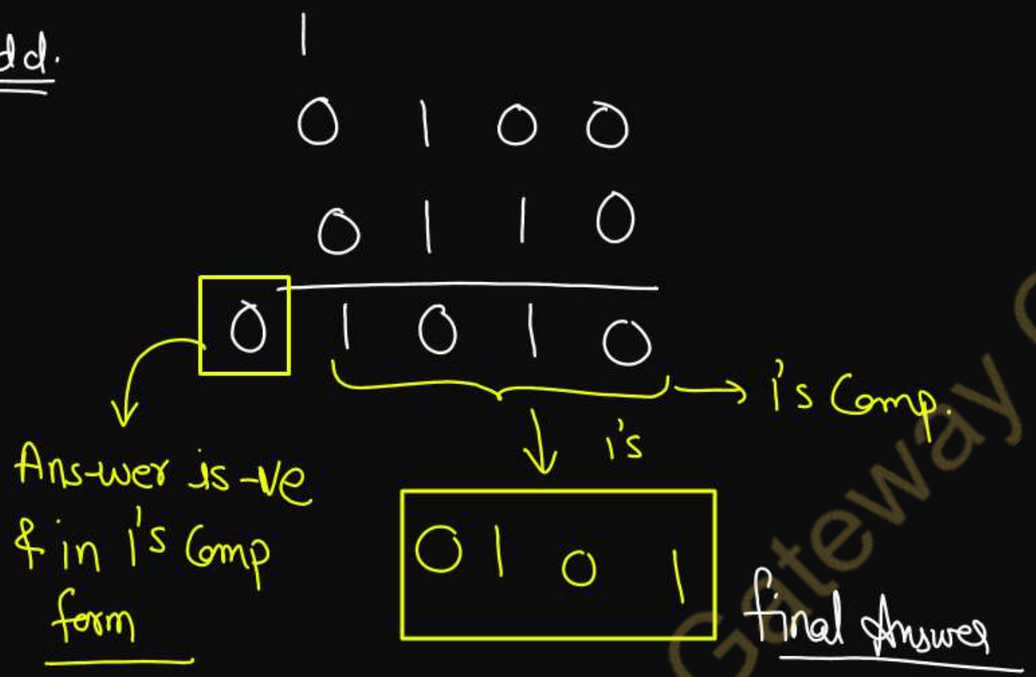
Example – 2 : Subtract using 1' S complement $(0100)_2 - (1001)_2 \rightarrow -(0101)_2$

↳A ↳B

B → 1001

1's Comp of B → 0110

Add.



Gateway Classes

Binary Subtraction 2' S complement

$A - B = A + (-B)$

STEP – 1 : Find the 2' S complement of number to be subtracted

STEP – 2 : Perform addition

STEP – 3 : CASE – 1 : If the final carry is [redacted] generated, then the result is positive and is in its true form

CASE – 2 : If the final carry is not generated, then the result is negative and is in its 2' S complement form.

↳ 0

Example – 1 : Subtract using 2' S complement $(1001)_2 - (0101)_2$ → $(0100)_2$ Ans

↳ B

B

1's of B

2's " "

0 1 0 1

1 0 1 0

+1

1 0 1 1 – 2's

A ↙ ↘

1 1

1 0 0 1

1 0 1 1

1 0 1 0 0

↳ discard

Gateway Classes

Example – 2 : Subtract using 2' S complement $(0100)_2 - (1001)_2 \rightarrow -(0101)_2$

A ↵ B ↵

B ⟶ 1 0 0 1

1's Comp of B → 0 1 1 0

+ 1

2's Comp → 0 1 1 1

2's Comp

0 1 0 0

0 1 1 1

0 1 0 1 1

↵ If Carry is 0

Answer is -ve and in 2's Comp form.

2's Comp of 1 0 1 1

0 1 0 0 ⟶ 1's

+ 1

0 1 0 1 final Answer

Gateway Classes

For Decimal system System

(i) 9' S complement

(ii) 10 ' S complement

9' S complement

Given Number	9' S complement
2467	9999 − 2467 = 7532 → 9's 7532 + 1 = 7533 → 10's
75468	99999 − 75468 = 24531 → 9's 24531 + 1 = 24532 → 10's

Gateway Classes

10' S complement

10' S complement = 9' S complement + 1

Given Number	9' S complement	10' S complement
4862	5137	5137 + 1 = 5138 (10's)
65432	34567	34567 + 1 = 34568 (10's)

Gateway Classes

Binary Subtraction 9' S complement

A - B = A + (-B)

STEP – 1 : Find the 9'S complement of B

STEP – 2 : Add A and 9'S complement of B

STEP – 3 : CASE – 1 : If the final carry is 1, then add it to the result obtained in step-2

CASE – 2 : If the final carry is zero, result obtain in step-2 is negative and in the 9'S complement form

Example – 1 : (437 – 235) using 9's

A ← 437 → B

B = 235

9's = 999
235
764

1 1
437
764
1 201
1
202
Ans.

Gateway Classes

Example – 2 :

$235 - 437$

B

9's of B =

$$\begin{array}{r} 999 \\ 437 \\ \hline 562 \\ \hline \end{array}$$

$$\begin{array}{r} 235 \\ 562 \\ \hline 0797 \\ \hline \end{array}$$

↵

-ve
9's Comp

Ans ⇒ 9's Comp of 797

$$\begin{array}{r} 999 \\ 797 \\ \hline 202 \\ \hline \end{array}$$

Answer = $-(202)_{10}$

Gateway Classes

Binary Subtraction 10' S complement

A - B = A + (-B)

STEP – 1 : Find the 10' S complement B

STEP – 2 : Add A and 10' S complement B

STEP – 3 : CASE – 1 : If the final carry is generated, then the result is positive and is in its true form

CASE – 2 : If the final carry is not generated, then the result is negative and is in its 2' S complement form.

Example – 1 : (437 - 235) using 10's

↳ B

10's of B →

```
  9 9 9
  2 3 5
-------
  7 6 4
    +1
-------
  7 6 5   → 10's
```

```
  4 3 7
  7 6 5
-------
1 2 0 2   Answer
```

↳ discard

Gateway Classes

Example – 2 :

235 – 437

10's of 437 =

```
  999
  437
 ----
  562
   +1
 ----
  563   → 10's
 ----
```

```
  235
  563
 ----
 0 798
 ----
```

↳ -ve

10's

```
  999
  798
 ----
  201   → 9's
   +1
 ----
  202   → 10's
 ----
```

Final Answer = –(202) Ans.

Gateway Classes

Digital System Design

UNIT -1 L 2 : DPP

Q. Perform following operation as indicated.

I. Subtract using 1's complement: $(10111)_2 - (110011)_2$

II. find the 1's and 2's compliment of $(010100)_2$

III. Determine 2's complement of $(1010.110)_2$.

IV. Convert $(25.125)_{10}$ into Hexadecimal number.

V. Add binary number $(1011)_2$ and $(1111)_2$.

VI. Subtract using 10's complement: $(9754)_{10} - (364)_{10}$

VII. Subtract using 1's complement: $(10111)_2 - (110011)_2$

Gateway Classes

AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-1 Lec-3



Today's target

- Signed magnitude representation
- Binary Codes
- PYQ
- DPP

Gateway Classes

Avinash Sir

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

UNIT-1 : Logic simplification and combinational logic design

- Number Systems
- Binary arithmetic
- Signed magnitude representation
- Binary codes, code conversion
- Review of Boolean algebra and Demorgans theorem
- SOP & POS forms
- Canonical forms
- Karnaugh maps up to 5 variables
- Tabulation method.

Gateway Classes

Signed magnitude representation

$2^n = 2^8 = 256$
(0 - 255)
-127 to +127

AKTU-2023-24

It is a method for storing signed binary numbers (numbers that can be either + or -). In this system, one bit of the binary number is dedicated to the sign, and the remaining bits represent the number's magnitude (its absolute value).

Sign Bit: The most significant bit (MSB), or the leftmost bit, is the sign bit. A 0 for +ve & 1 For – ve Number.

Magnitude Bits: The remaining bits represent the magnitude of the number in standard binary.

For example, using an 8-bit system:

Sign bit

- The decimal number +53 would be represented as 0 0 1 1 0 1 0 1
- The decimal number −53 would be represented as 1 0 1 1 0 1 0 1.

Range :- For a 8-digit number range is 0 to 255. But in signed magnitude representation range will be -127 to +127.

Hence largest negative number is -127 or 1 1 1 1 1 1 1 1 &

largest positive number is +127 or 0 1 1 1 1 1 1 1

Gateway Classes

Signed Complement Numbers

When representing negative numbers in binary, two common systems are used:

1's Complement Representation

2's Complement Representation

Both methods allow positive and negative numbers to be stored in binary format.

Gateway Classes

1. 1's Complement

Positive numbers → same as normal binary.

Negative numbers → take the binary of the absolute value, then invert all bits (0 → 1, 1 → 0).

Example (with 4-bit system):

Represent +5 and –5.

+5 in binary (4-bit) = 0101

–5 → take 5 (0101), invert → 1010

2's Complement Representation

Positive numbers: written as normal binary.

Negative numbers: Take binary of positive number.

Find 1's complement (invert).

Add 1.

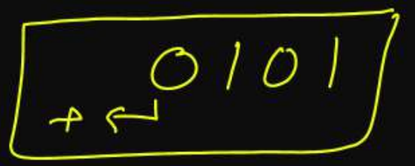
Example (using 4 bits): +5 = 0101

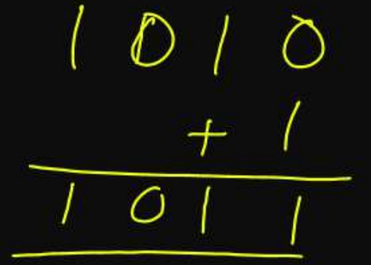
To get –5:

Invert → 1010

Add 1 → 1010 + 1 = 1011

So –5 = 1011 (2's complement form).



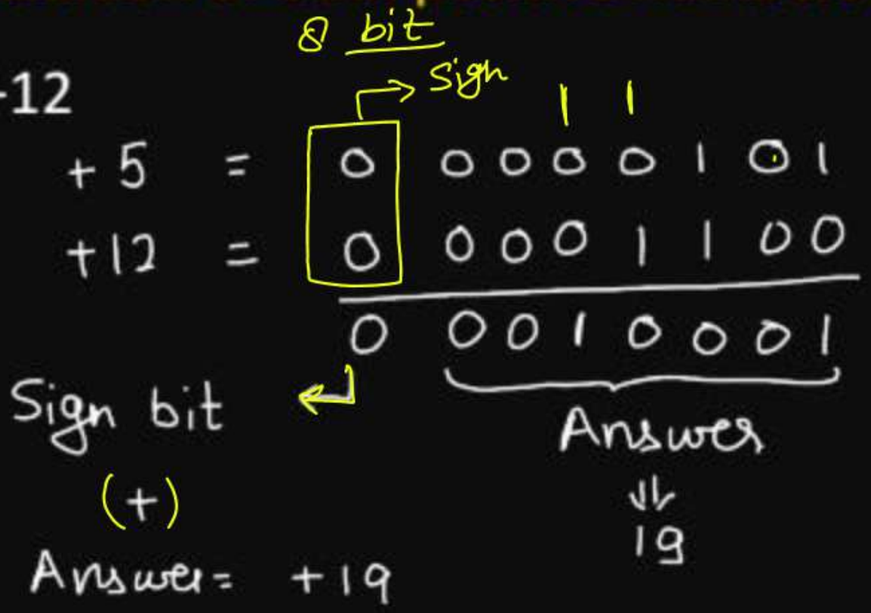


Gateway Classes

Addition of signed Numbers

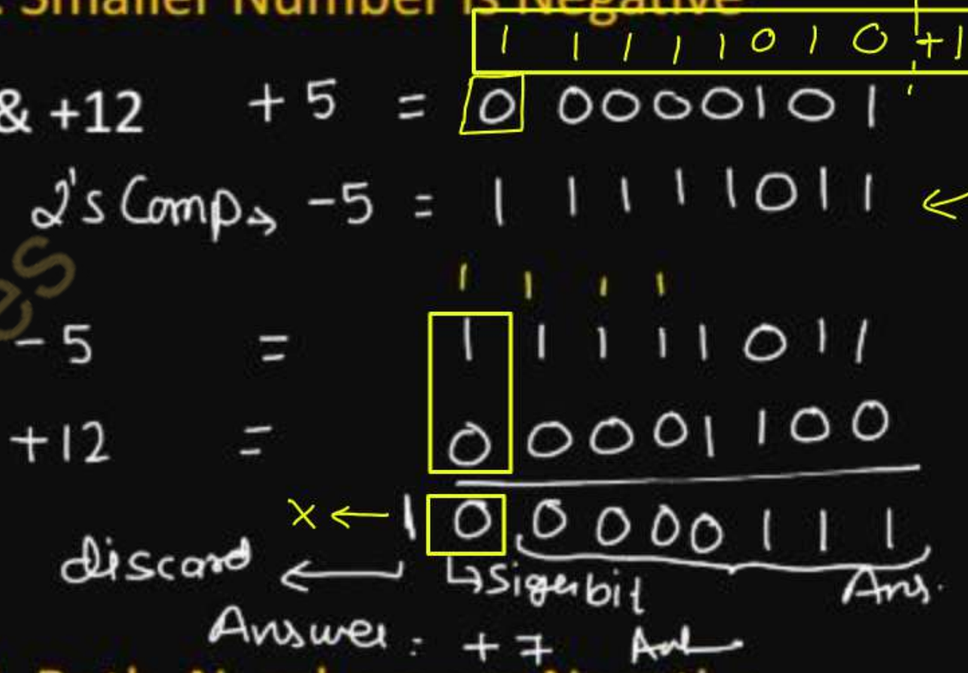
Case-1 : Addition of both positive numbers

Ex. +5 & +12



Case2: Smaller Number is Negative

Ex. -5 & +12



Case3: Large Number is Negative

Ex. + 5 & -12

Case 4: Both Numbers are Negative

Ex. -5 & - 12

Gateway Classes

+5 & -12

+5 → 0 0000101

-12 → 2's Comp of +12

+12 → 0 0001100

1's Comp → 1 1 1 1 0 0 1 1

+ 1

2's Comp = 1 1 1 1 0 1 0 0 → -12

+5 → 0 0000101

-12 → 1 1 1 10100

1 1 1 1 1 0 0 1

Sign bit ← Answer (-)

↓ 2's Comp of Answer

∴ Final Answer 1 1 1 1 0 0 1 → 2's comp.

−1

1 1 1 1 0 0 0 → 1's Comp

0 0 0 0 1 1 1

final Answer −7

Ex:

−5 , −12

↳2's ↳2's

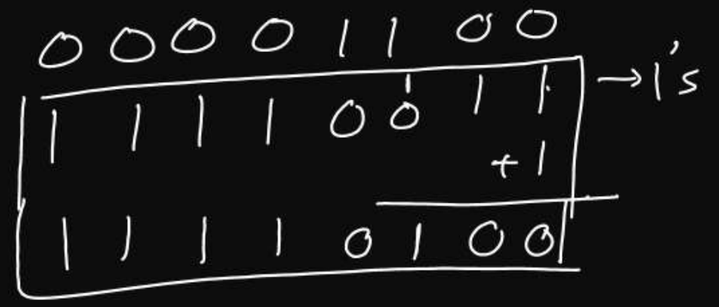
Sign - 1

Answer 2's Comp

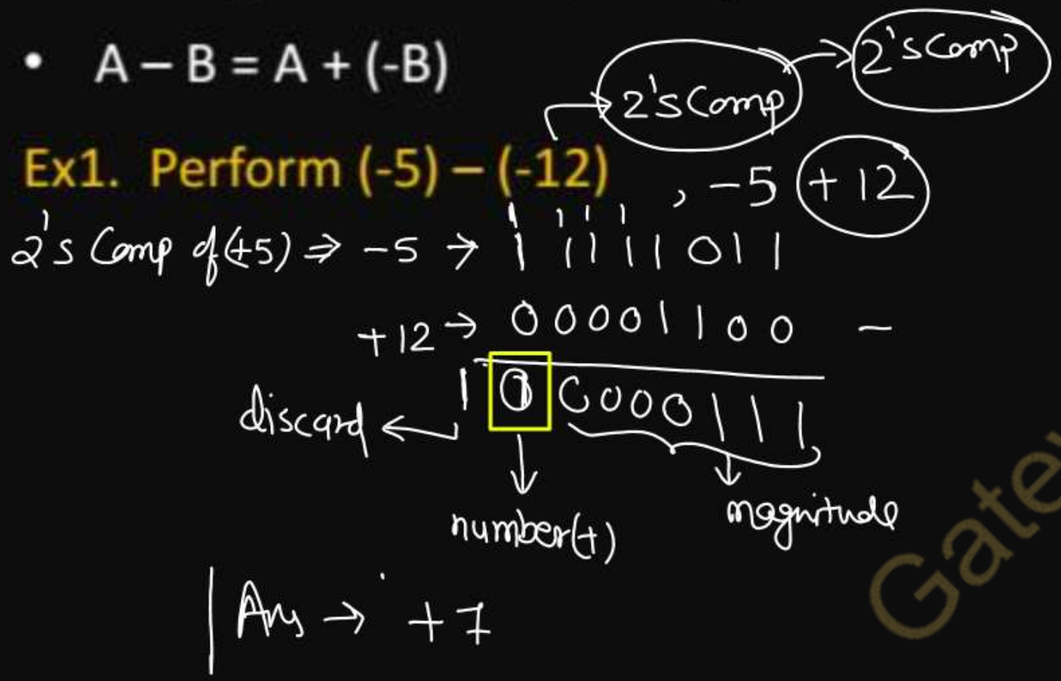
Gateway Classes

Subtraction of signed Numbers

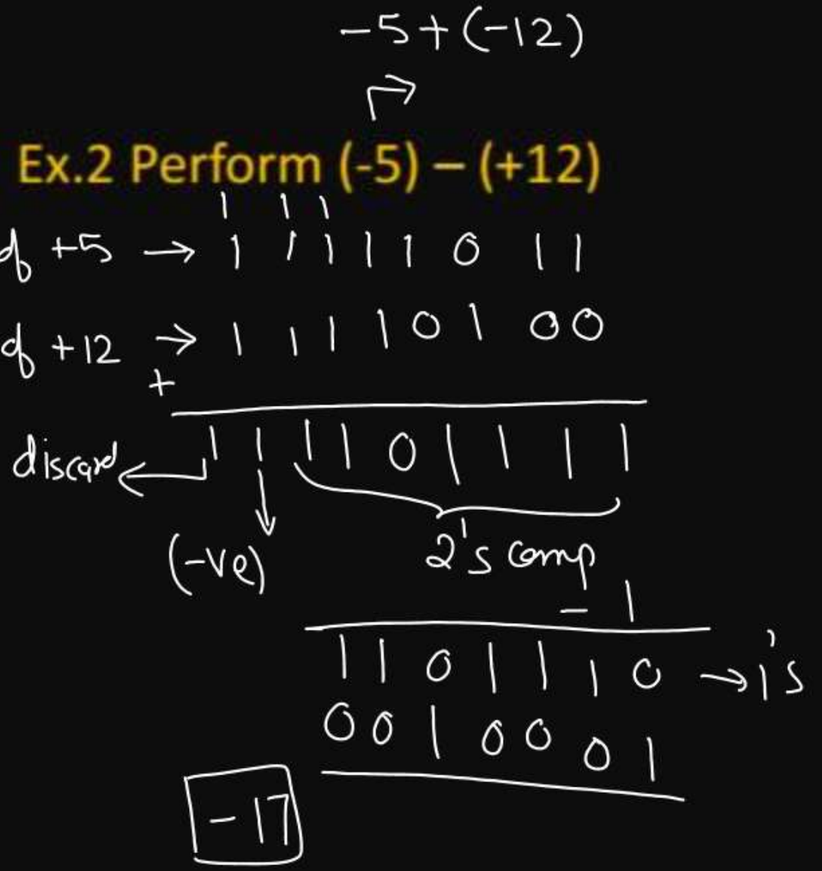
- Take 2's complement of subtrahend
- Add it to the minuend (including sign bits).
- A carry out of the sign bit position should be discarded.
- A – B = A + (-B)



Ex1. Perform (-5) – (-12)



Ex.2 Perform (-5) – (+12)



Gateway Classes

Binary Codes

- A binary code is a system of representing information (numbers, characters, symbols) using 0s and 1s.
- Each element of the information is represented by a group of bits (code word).
- Used in digital systems, computers, and communication.

Types of Binary Codes

(A) Weighted Codes

Each bit position has a fixed weight.

Ex. BCD code, 8421 code , 2421 code

(B) Non-Weighted Codes

Do not use positional weights.

Gray Code, Excess Code

(C) Alphanumeric Codes

Represent letters, numbers, and symbols.

ASCII Code , EBCDIC code

Gateway Classes

Binary Codes

BINARY CODES - FLOW CHART

BINARY CODES

Weighted Codes

- BCD (8421)
- 2421 Code

Non-Weighted Codes

- Excess-3
- Gray Code

Alphanumeric Codes

- ASCII (7/8-bit)
- EBCDIC (8-bit)

Error Control Codes

- Parity Bit (Error Detect)
- Hamming Code (Detect + Correct)

Gateway Classes

BCD Code (8421 code)

Ten + 6 → invalid

- BCD (Binary Coded Decimal)Decimal digits (0–9) are represented by 4-bit binary numbers.
- But with 4 bits we can represent 0000 to 1111 (16 numbers). In BCD remaining six numbers (1010 to 1111) are invalid.

Conversion of Decimal to BCD

(1) 78 (ii) 236

7 → 0111

8 → 1000

BCD Code → 0111 1000

2 → 0010, 3 → 0011, 6 → 0110

BCD code

0010 0011 0110

Decimal	Binary	BCD (8421)
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	0001 0000
11	1011	0001 0001
12	1100	0001 0010
13	1101	0001 0011
14	1110	0001 0100
15	1111	0001 0101

Same as Binary

Gateway Classes

Advantages of BCD Code

- Easy Conversion → Direct conversion between decimal digits and binary form.
- Human-friendly → Representation is closer to decimal, making it easier for humans to read.
- Reduces coding errors → Each decimal digit is separately encoded, making debugging easy.

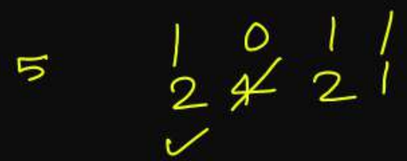
Disadvantages of BCD Code

- Not space-efficient → Uses 4 bits for only 10 digits (0000–1001). Six combinations (1010–1111) are unused.
- Slower arithmetic → Addition and subtraction are more complex compared to pure binary and having different rules
- Less memory-efficient → Compared to pure binary, it consumes more storage.

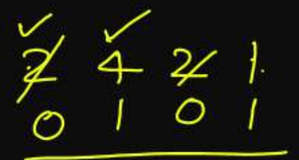
Applications of BCD Code

- Digital Displays → 7-segment displays in calculators, digital clocks, counters.
- Financial Calculators → Used where decimal accuracy is needed (avoids binary rounding errors).
- Digital Systems → Where decimal data input/output is common (e.g., billing machines, digital meters).

Gateway Classes



2421 code



- Weighted code with weights 2,4,2,1.
- Two position has the same weights
- There could be two possible bit pattern for some decimal numbers
- But we can use only one

- Ex. 5 → 1011 (2+0+2+1=5).

Other codes are 3321 , 4221, 5211, 5421, 6311, 7421,

Decimal	Binary	(2421 Code)
0	0000	0 0 0 0
1	0001	0 0 0 1
2	0010	0 0 1 0
3	0011	0 0 1 1
4	0100	0 1 0 0
5	0101	1 0 1 1
6	0110	1 1 0 0
7	0111	1 1 0 0
8	1000	1 1 1 0
9	1001	1 1 1 1

Gateway Classes

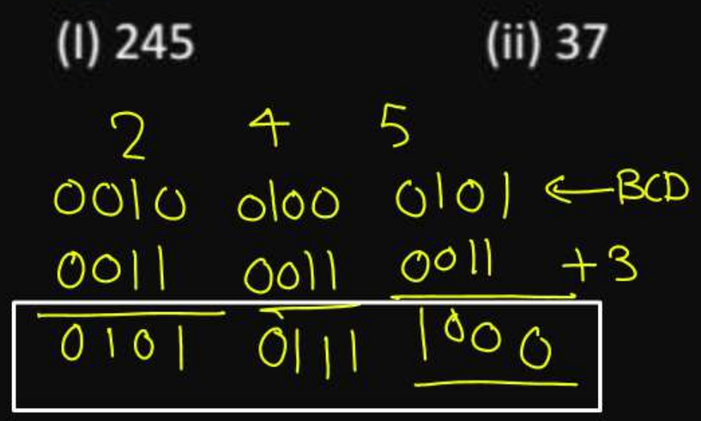
Excess 3 code

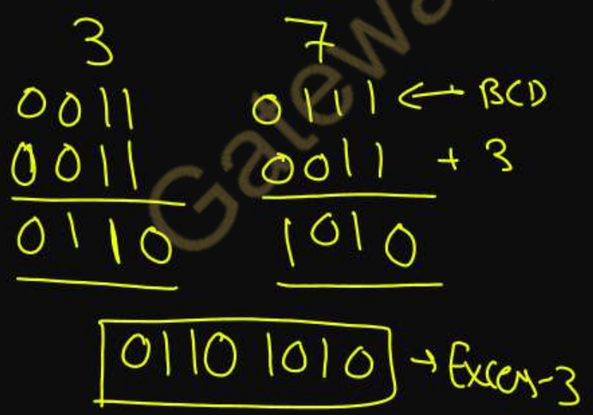
- Derived from BCD by adding 3 (0011) to each digit.
- Example:

  Decimal 5 → BCD (0101) + 0011 = 1000
- Self-complementing code (9's complement easy).

Ex:-

(I) 245 (ii) 37



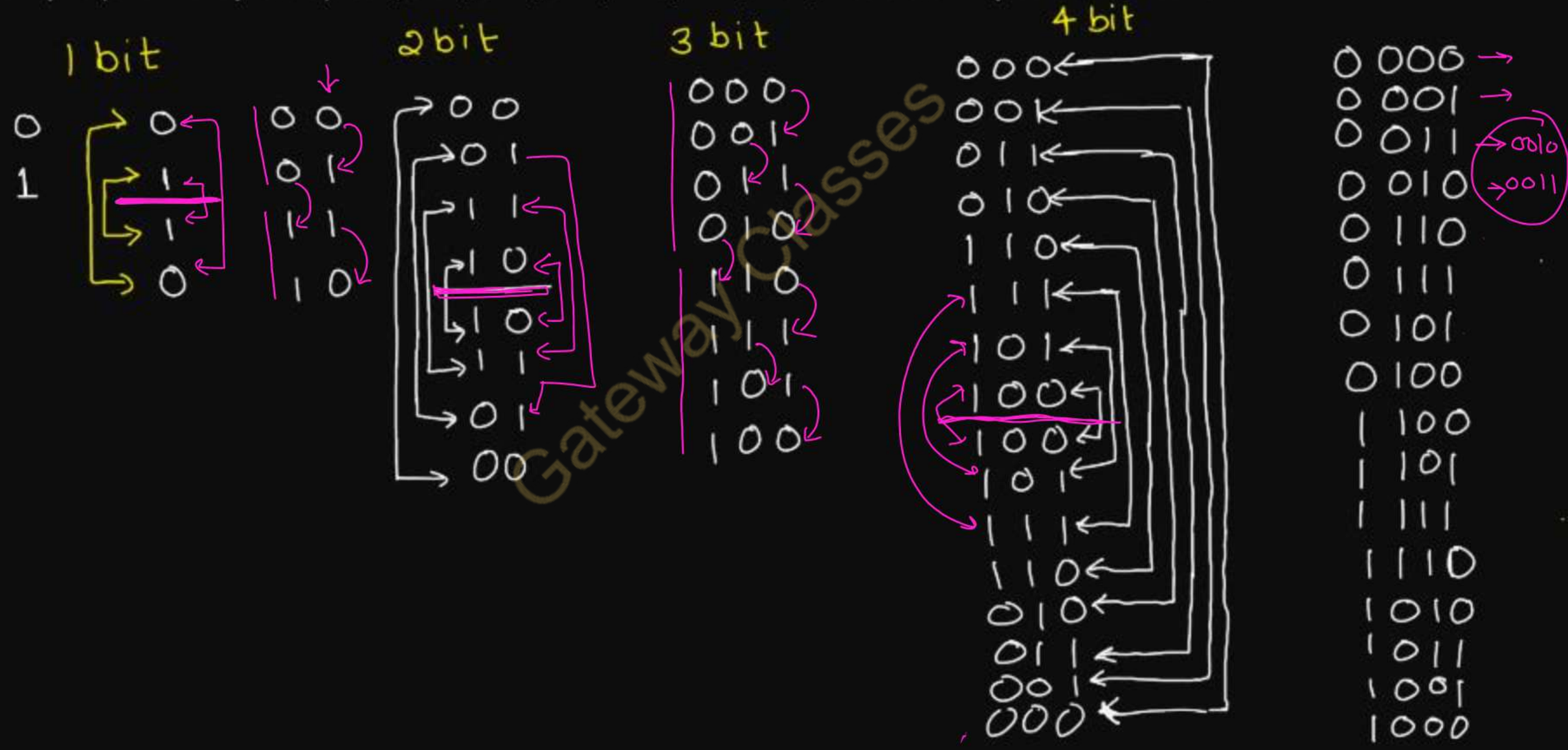


Decimal	BCD	Excess 3
0	0000 +0011 (3)	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

Gateway Classes

Gray Code (Reflected Binary Code)

- Consecutive numbers differ by only one bit (minimizes errors). Also known as unit distance code or cyclic codes
- Example (3-bit Gray code):000, 001, 011, 010, 110, 111, 101, 100Used in position encoders.



Gateway Classes

Advantages of Gray Code

- Error Reduction → Only 1 bit changes, minimizing errors in transmission.
- Simple hardware implementation → Especially useful in position sensors.
- Used in Karnaugh Maps (K-Maps) → Simplifies logic minimization since adjacent cells differ by only 1 bit.

Disadvantages of Gray Code

- Not suitable for arithmetic → Cannot directly add or subtract in Gray code.
- Conversion needed → Must convert to/from binary for arithmetic operations.
- More complex for large systems compared to binary.

Applications of Gray Code

- Rotary and Linear Encoders → To measure shaft positions without errors.
- Karnaugh Maps (K-Maps) → For minimizing Boolean expressions.
- Error Detection → Reduces chances of errors during bit transitions in communication.
- Digital Systems → Used where precision and error-free transitions are important.

Gateway Classes

Digital System Design

UNIT -1 L 3 : DPP

Q. Perform following operation as indicated.

I. Add using Sign numbers 7 & 12 (ii) 9 & -13

II. Subtract using Sign numbers -13 & (-14)

III. Convert 243 in

BCD code, Excess 3 code , 2421 code

Gateway Classes

AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-1 Lec-4



Avinash Sir

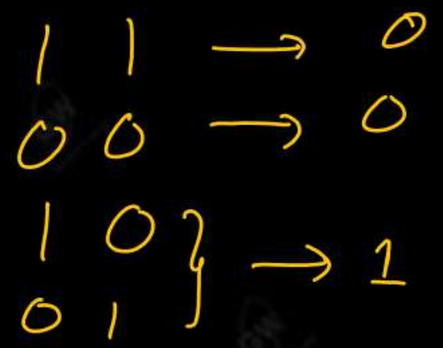
Today's target

- Code Conversions
- Review of Boolean Algebra and Laws
- De Morgan Theorem
- PYQ , DPP

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

UNIT-1 : Logic simplification and combinational logic design

- Number Systems
- Binary arithmetic
- Signed magnitude representation
- Binary codes, code conversion
- Review of Boolean algebra and Demorgans theorem
- SOP & POS forms
- Canonical forms
- Karnaugh maps up to 5 variables
- Tabulation method.



GRAY CODE CONVERSION

(i) Binary → Gray Code Conversion

Rule: 1. The most significant bit (MSB) of Gray code = MSB of Binary code.

2. Each next Gray code bit = XOR of previous Binary bit and current Binary bit.

Example: Convert Binary 1011 to Gray Code

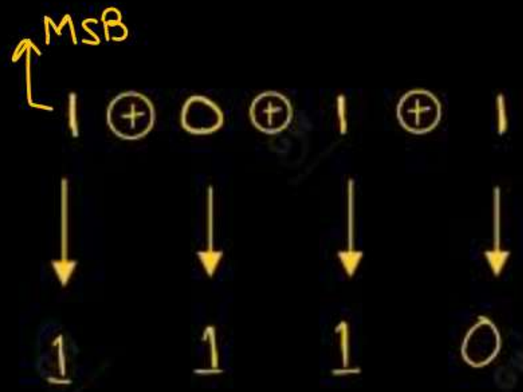
MSB remains same → 1

2nd bit = 1 ⊕ 0 = 1

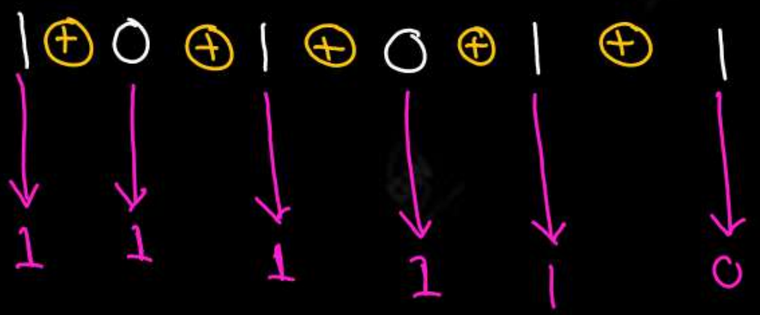
3rd bit = 0 ⊕ 1 = 1

4th bit = 1 ⊕ 1 = 0

Gray Code = 1110



Ex. Convert 101011 into Gray Code



(111110) Gray Code.

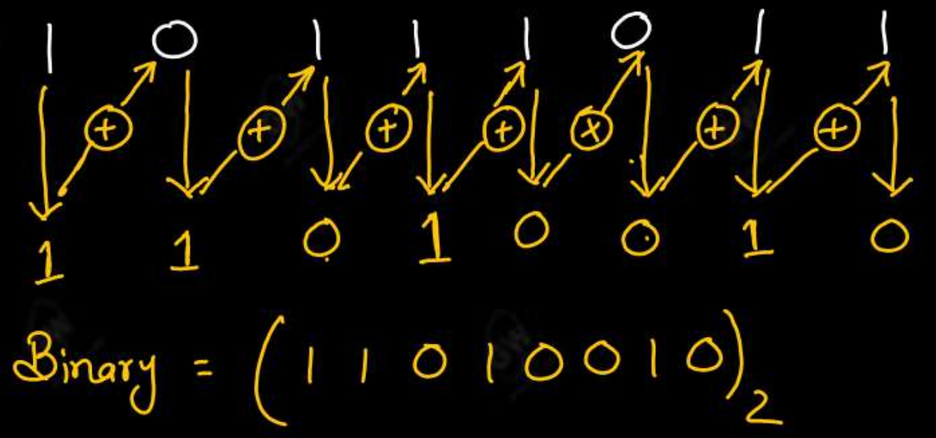
GRAY CODE CONVERSION

(ii) Gray Code → Binary Conversion

Rule:

The MSB of Binary = MSB of Gray.

Each next Binary bit = Previous Binary bit ⊕ Current Gray bit.



Example:

Convert Gray 1110 to Binary

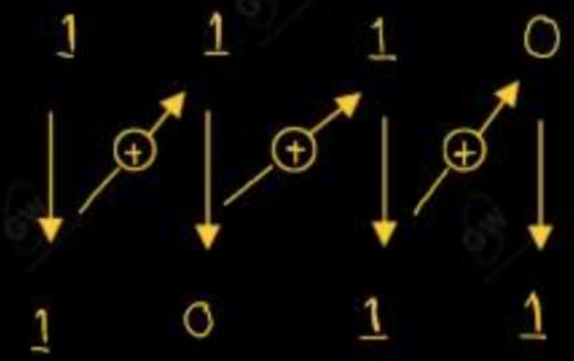
MSB remains same → 1

2nd bit = 1 ⊕ 1 = 0

3rd bit = 0 ⊕ 1 = 1

4th bit = 1 ⊕ 0 = 1

Binary = 1011



Ex. Convert 10111011

Binary ↔ Binary Coded Decimal (BCD)

(a) Binary → BCD

Binary → decimal → BCD

BCD represents each decimal digit using 4 bits.

0101 ← → 1001

Example: Decimal 59

Binary = 111011 (6 bits)

BCD = 0101 1001 (two groups of 4 bits → "5" and "9")

5 9

(b) BCD → Binary

BCD → Dec → Binary

Convert each 4-bit group into decimal, then into full binary.

Binary ↔ Excess-3 (XS-3) Code

Excess-3 is a self-complementary code, each decimal digit = BCD + 3.

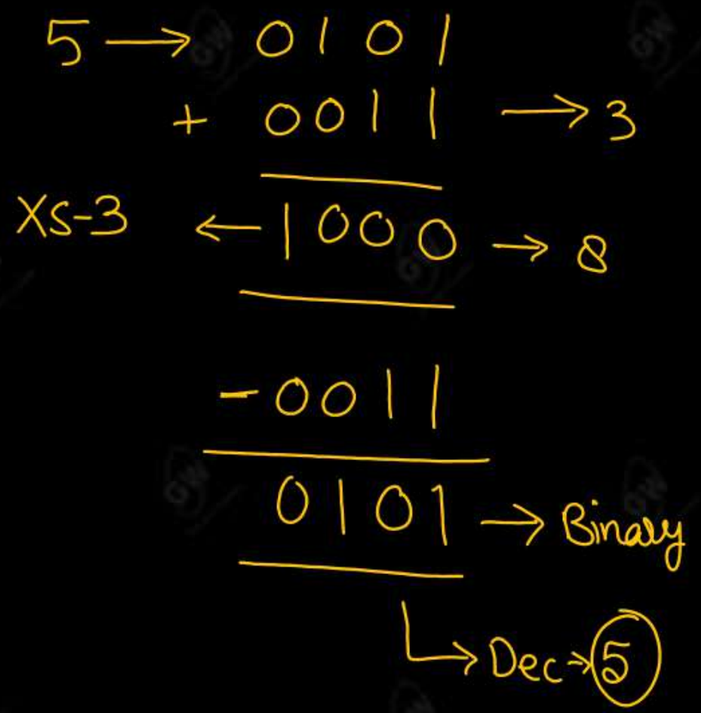
(a) Decimal → XS-3

Example: Decimal 5

BCD = 0101

Add 3 = 1000

XS-3 = 1000



(b) XS-3 → Decimal

Example: XS-3 = 1000

Subtract 3 → 0101 (5 in decimal)

Boolean Algebra

- Boolean Algebra is used to analyze and simplify digital (logic) circuits.
- Each variable in Boolean algebra has values either 0 (Low) or 1 (High).

Low → 0 → false

High → 1 → True

- The basic operations used in Boolean algebra are:

1 + 0 = 1 1 . 0 = 0 1 → 0

(a) Logical addition (b) Logical multiplication (c) Complementation

- Laws of Boolean Algebra

(1) Commutative Law (2) Associative Law (3) Distributive Law

(4) AND Laws (5) OR Laws (6) Inversion Law

(7) Absorption Law (8) De Morgan's Law

(1) Commutative Law

The commutative law states that changing the sequence of the variables does not have any effect on the output of a logic circuit.

(a) $A \cdot B = B \cdot A$ (b) $A + B = B + A$

(2) Associative Law

This law states that the order in which the logic operations are performed is irrelevant as their effect is the same.

(a) $(A \cdot B) \cdot C = A \cdot (B \cdot C)$ (b) $(A + B) + C = A + (B + C)$

(3) Distributive Law

$$A \cdot (B + C) = A \cdot B + A \cdot C$$



$\overline{\overline{A}} = A$

$A = 1$

$\overline{A} = 0$

$1 \cdot 0 \rightarrow 0$

(4) AND Laws

These laws use the AND operation. Therefore, they are called AND laws.

(a) $A \cdot 0 = 0$ (b) $A \cdot 1 = A$ (c) $A \cdot A = A$ (d) $A \cdot \bar{A} = 0$

Any var. ↵ ↳ always

(5) OR Laws

These laws use the OR operation. Therefore, they are called OR laws.

(a) $A + 0 = A$ (b) $A + 1 = 1$ (c) $A + A = A$ (d) $A + \bar{A} = 1$

$0+1=1$
$1+1=1$

$1+1=1$
$0+0=0$

Let $A = 1, 0$
$\bar{A} = 0, 1$
$1+0=1$

(6) Inversion Law

The inversion law states that double inversion of a variable results in the original variable itself

$$\overline{\overline{A}} = A$$

(7) Absorption Law

This Law enables a reduction in a complicated expression to a simple one by absorbing like terms

$$A + AB = A \Rightarrow A(1+B) = A$$

$1+B \rightarrow 1$

M.Imp

AKTU - 2023-24

(8) De Morgan's Law

The two important laws have been suggested by De Morgan's.

(i) This law states that the complement of a product is equal to the addition of the complements.

$$\overline{AB} = \overline{A} + \overline{B} \qquad \overline{ABC} = \overline{A} + \overline{B} + \overline{C}$$

(ii) This law states that the complement of a sum is equal to the product of the complements.

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

Ex.

$$A + BC = (A+B)(A+C)$$

Proof.

$$(A+B)(A+C) = A \cdot A + A \cdot C + B \cdot A + B \cdot C$$

$$= A + A \cdot C + B \cdot A + B \cdot C$$

$$= A(1+C) + B \cdot A + B \cdot C$$

$$= A + BA + BC$$

$$= A(1+B) + BC$$

$$= A + BC$$

Example – 1 : Simplify the expression $\underset{①}{A} + \bar{A}B + \underset{③}{AB}$ using Boolean algebra.

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

$= A + B$

Ans.

Rule :- $A + \bar{A} = 1$

$\bar{A}B + A(1 + B)$

$A + \bar{A}B$

Example – 2 : Simplify the Boolean function using Boolean algebra theorems

$$A'B'C' + A'BC' + AB'C' + ABC'$$

$= A'C'(B' + B) + AC'(B' + B)$

$= A'C' + AC'$

$= C'(A + A')$

$= C'$

$\because B + B' = 1$

$\because A + A' = 1$

Ans.

Example – 3 : Simplify the Boolean function F = XY + XY'Z + YZ' using Boolean algebra.

$= XY(Z+Z') + XY'Z + YZ'$ ← $XY[Z+\bar{Z})$

$= XYZ + XYZ' + XY'Z + YZ'$

$= XZ(Y+Y') + YZ'(X+1)$ ← $Y+Y'=1$, $X+1=1$

$F = XZ + YZ'$

Ans.

Dis Proof.

Example – 4 : Simplify the following expression Y = (A+B)(A+C).

$Y = (A+B)(A+C)$

$= A \cdot A + A \cdot C + B \cdot A + B \cdot C$

$= A + AC + BA + BC$

$= A(1+C) + BA + BC$

$= A + AB + BC = AC(1+B) + BC$

$Y = A + BC$

SOP and POS Representation for Logic Expression

Sum of Product → Var → A, B, C

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

Any logic expression may be expressed in the following two forms

(i) Sum of Product form (SOP)

$f(A, B, C) = \underbrace{AB}_{Cx} + \underbrace{BC}_{Ax} + \underbrace{AC}_{Bx}$ → Product →

Standard or Canonical SOP form

Each product term consists of all the variables in the complemented or un-complemented form.

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

Product of Sum

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

(ii) Product of Sum form (POS)

$f(A, B, C) = \underbrace{(A+B)}_{Cy} . \underbrace{(B+C)}_{Ax} . \underbrace{(A+C)}_{Bx}$

Standard or Canonical POS form

Each sum term consists of all the variables in the complemented or un-complemented form.

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

Conversion of a logical expression to standard SOP or POS form

Example – 1 : Convert the expression $Y = A\,B + \bar{A}\,C + B\,C$ into the standard SOP form.

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

$$= ABC + AB\bar{C} + \bar{A}BC + \bar{A}\bar{B}C + ABC + \bar{A}BC$$

$$= ABC + AB\bar{C} + \bar{A}BC + \bar{A}\bar{B}C \rightarrow \text{minterm}$$

$A \cdot B \rightarrow C \rightarrow (C+\bar{C}) \rightarrow 1$

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

$BC \rightarrow A \rightarrow (A+\bar{A})$

$A + A = A$

Ans

$0 \leftarrow C \cdot \bar{C}$ $\quad B \cdot \bar{B}$ $\quad A \cdot \bar{A}$

$C \quad B \quad A$

$\therefore A + BC = (A+B)(A+C)$

Example – 2 : Convert the expression $Y = (A+B)(A+C)(B+\bar{C})$ to standard POS form.

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

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

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

Maxterm

Ans

SOP

Minterm and Maxterm

Minterm

- A product term which contains all the function variables either in complemented form or in un-complemented form is called a Minterm. ABC , ABC̄ ...
- Each individual term in the standard SOP form is called a Minterm

Maxterm

- A sum term which contains all the function variables either in complemented form or in un-complemented form is called a Maxterm.
- Each individual term in the standard POS form is called a Maxterm

Minterm { 0 — Comp ; 1 → UnComp

Max. 0 — UnComp ; 1 → Comp

Maximum number of minterms = 2^n

Maximum number of maxterms = 2^n

Where n = number of variables

Variables		Minterms	Maxterms
A	B	m_i	M_i
0	0	$m_0 = \overline{A}\ \overline{B}$	$M_0 = A + B$
0	1	$m_1 = \overline{A}\ B$	$M_1 = A + \overline{B}$
1	0	$m_2 = A\ \overline{B}$	$M_2 = \overline{A} + B$
1	1	$m_3 = AB$	$M_3 = \overline{A} + \overline{B}$

Logic Gates

- The most basic element of digital circuits are the logic gates.
- Boolean functions may be practically implemented by using Logic gates.
- Electronic gates require a power supply.

LOGIC GATES

fundametal

Basic gates

(i) NOT gate

(ii) AND gate

(iii) OR gate

Universal gates

(i) NAND gate NOT + AND

(ii) NOR gate NOT + OR

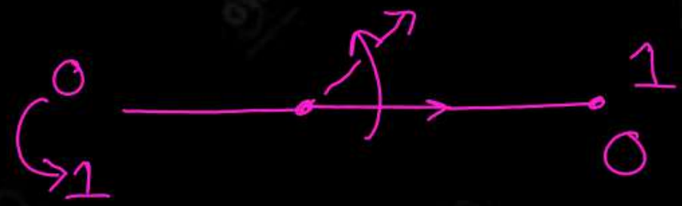
Special purpose gate

(i) EXOR gate

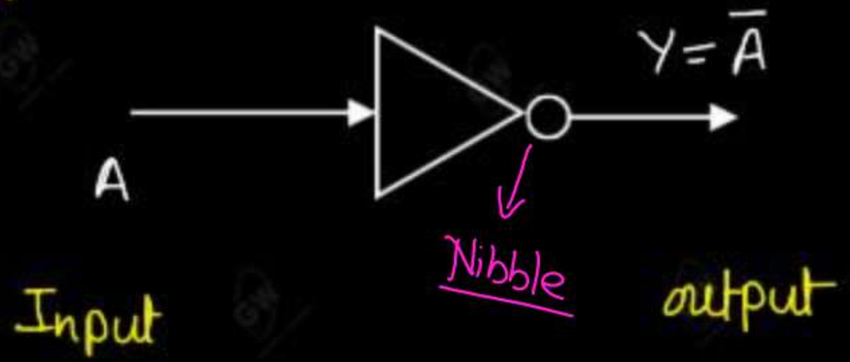
(ii) EXNOR gate

(1) NOT Gate

- The NOT gate is also known as an inverter.
- Output is the inverted version or complement of its input.
- NOT gate has only one input and one output.



Symbol

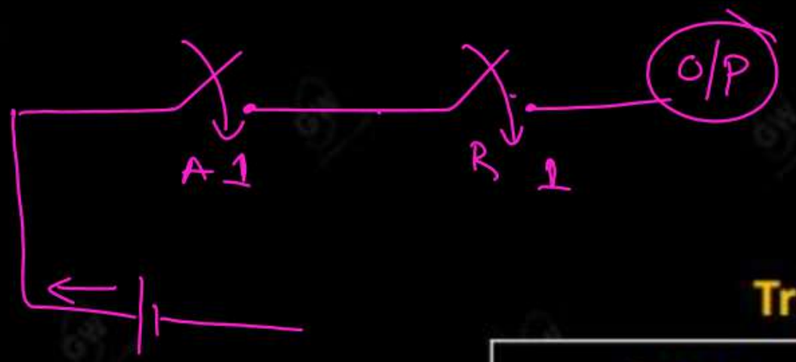


Truth Table

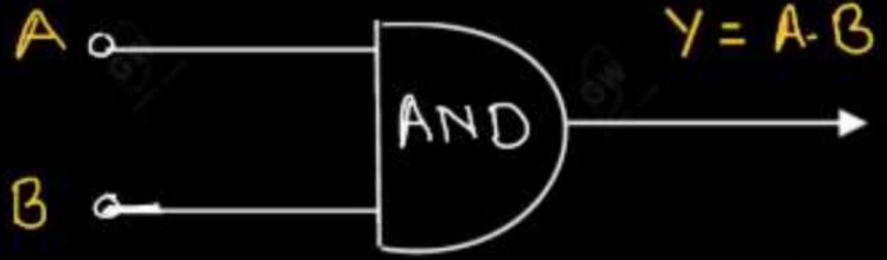
Input	Output
A	$Y = \bar{A}$
0	1
1	0

(2) AND gate

- The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high.
- It performs logical multiplication on its inputs.



Symbol



Truth Table

Input		Output
A	B	Y = A . B
0	0	0
0	1	0
1	0	0
1	1	1

(3) OR gate

- The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high.
- It performs logical addition on its inputs.



Symbol



Truth Table

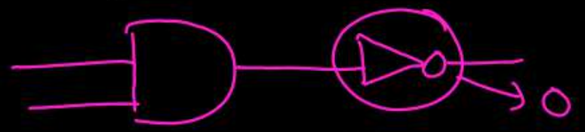
Input		Output
A	B	Y = A + B
0	0	0
0	1	1
1	0	1
1	1	1

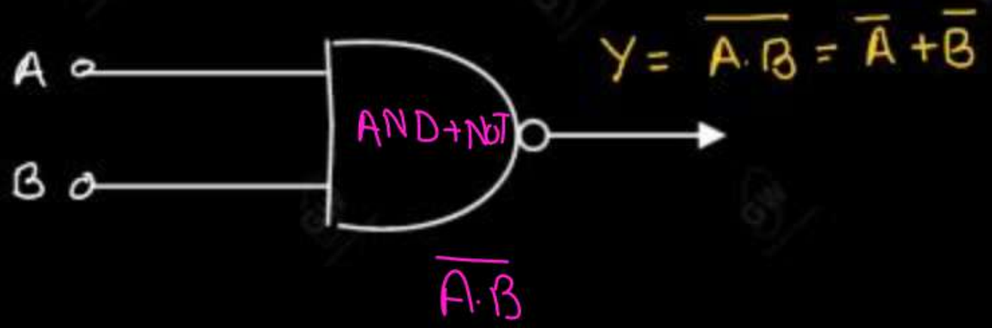
(4) NAND gate

Universal

- NOT + AND gate
- The input of all NAND gates are high if any of the input are low
- The small circle represents inversion

Symbol





Truth Table

Input		Output	
A	B	A·B	$Y = \overline{A \cdot B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

(5) NOR gate

OR + NOT

- Not + OR Gate
- The outputs of all NOR gates are low if any of the inputs are high.

A+B

Symbol



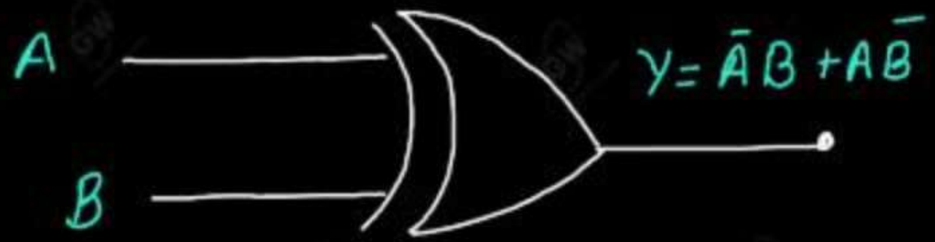
Truth Table

Input		Output	
A	B	OR	$Y = \overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

(6) EXOR gate

- The EXOR gate is a circuit which will give a high output if both the input are different
- $\oplus$ is used to show EXOR operation

Symbol $\oplus$



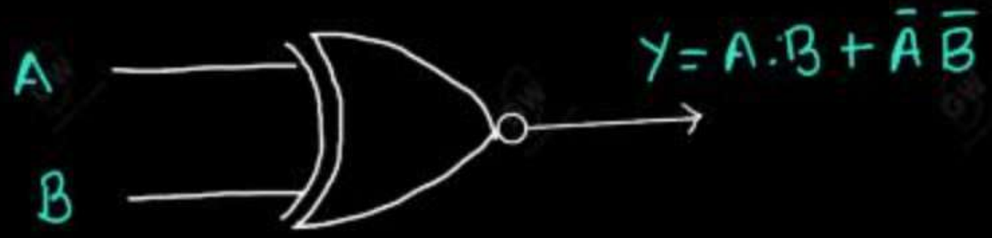
Truth Table

Input		Output
A	A	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

(7) EX – NOR gate

- The EX-NOR gate is a circuit which will give a high output if both the input are same.
- $\odot$ is used to show EX-NOR gate.

Symbol



Truth Table

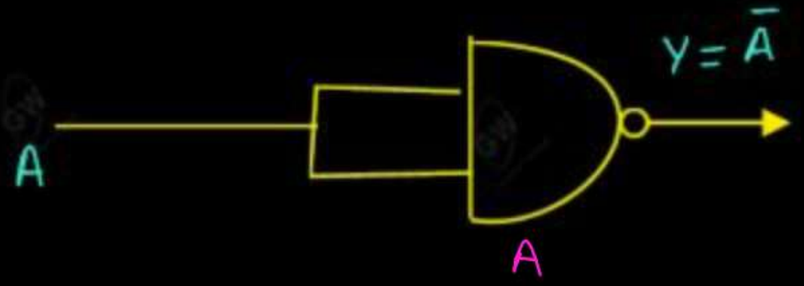
Input		Output
A	A	$Y = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

Universal Gates

- NAND and NOR gates are called as universal gates because it is possible to implement any Boolean expression with the help of only NAND gates or only NOR gates.
- A user can build any logical gates or combinational circuit with the help of only NAND gates or only NOR gates.
- NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families.

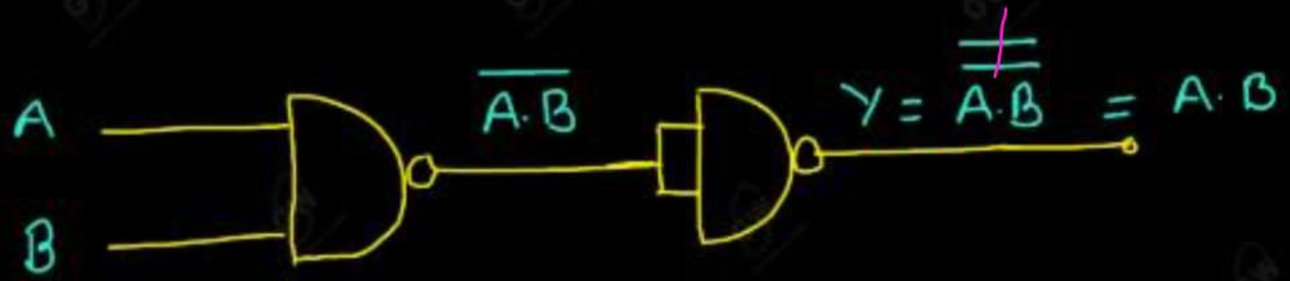
NAND Gate as a Universal Gate

(1) NOT gate using NAND gate

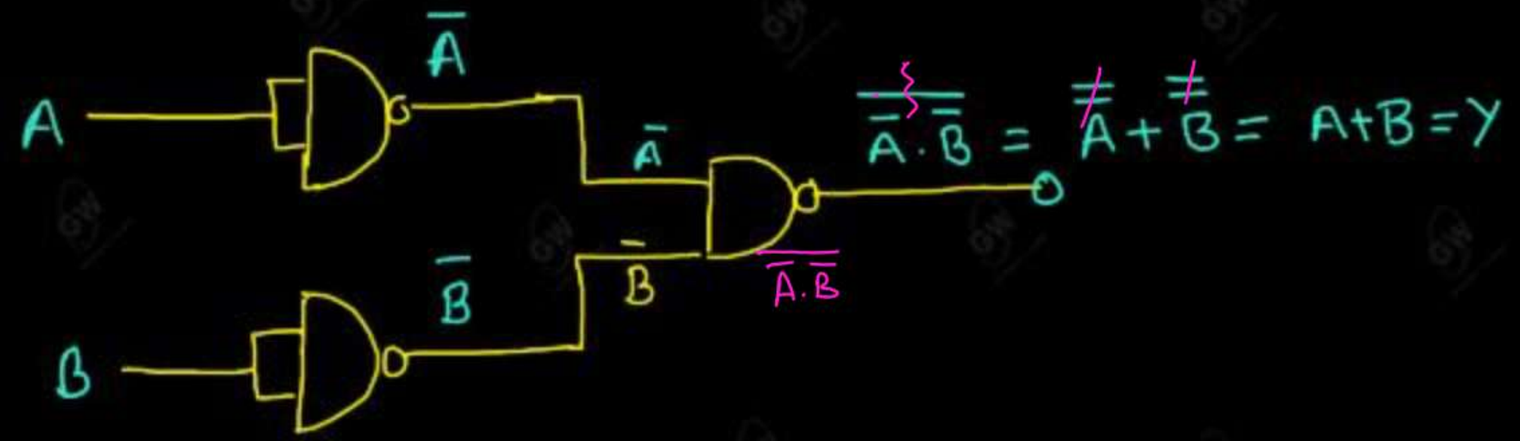


$A \cdot B$

(2) AND gate using NAND gate

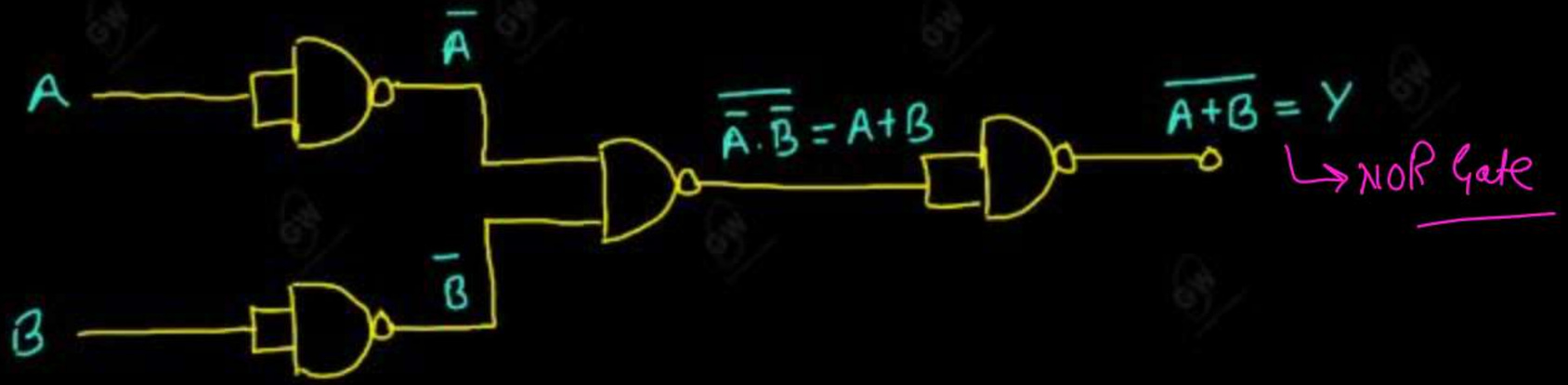


(3) OR gate using NAND gate

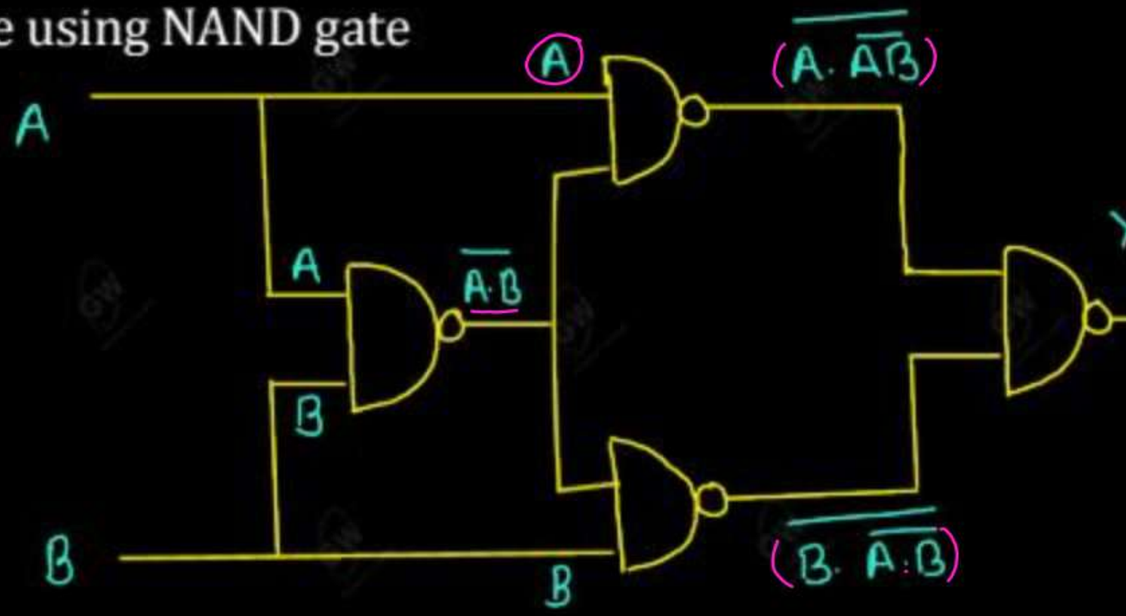


(4) NOR gate using NAND gate

AKTU-2024-25



(5) EXOR gate using NAND gate



$$Y = \overline{\overline{A \cdot \overline{AB}} \cdot \overline{B \cdot \overline{AB}}} = A \cdot \overline{AB} + B \cdot \overline{A \cdot B}$$

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

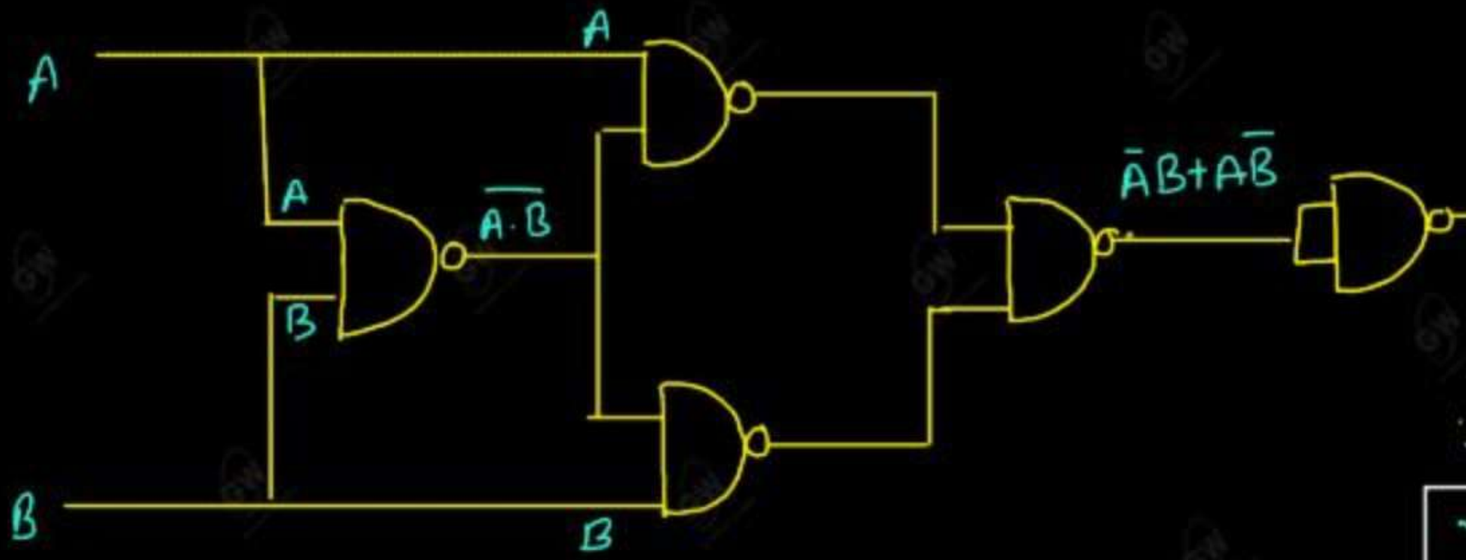
$$Y = A(\bar{A} + \bar{B}) + B(\bar{A} + \bar{B})$$

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

($A\bar{A} = 0$, $B\bar{B} = 0$)

$$\boxed{Y = A\bar{B} + \bar{A}B} \rightarrow \text{Ex-OR}$$

(6) EXNOR gate using NAND gate



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

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

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

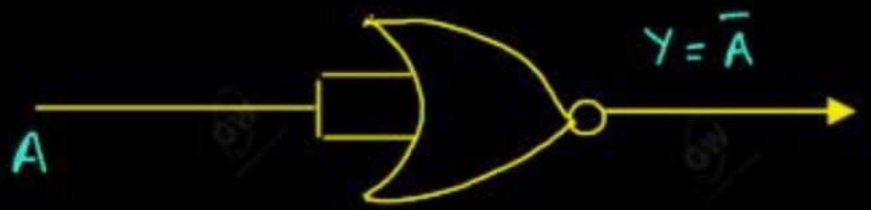
$$Y = A\bar{A} + AB + \bar{B}\bar{A} + B\bar{B}$$

($A\bar{A} = 0$, $B\bar{B} = 0$)

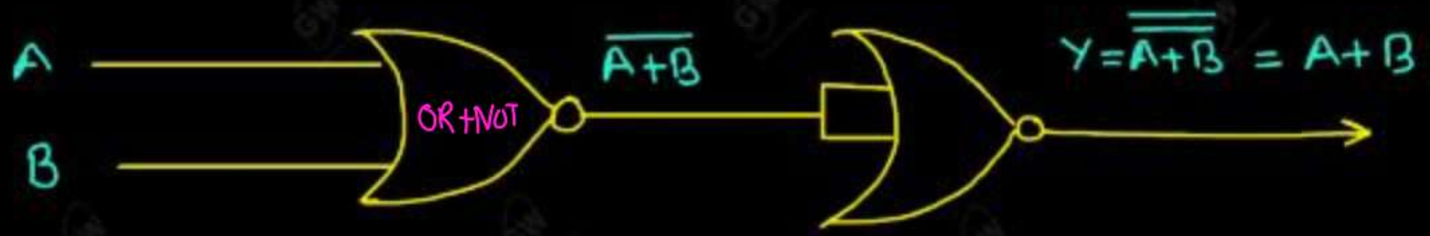
$$\boxed{Y = AB + \bar{A}\bar{B}}$$

NOR gate as a universal gate

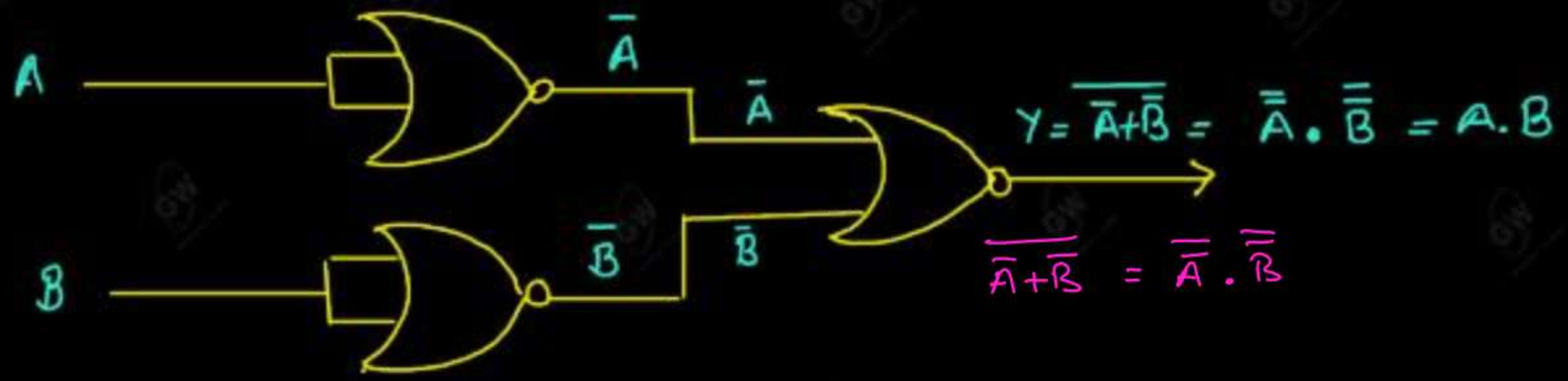
(1) NOT gate using NOR gate



(2) OR gate using NOR gate

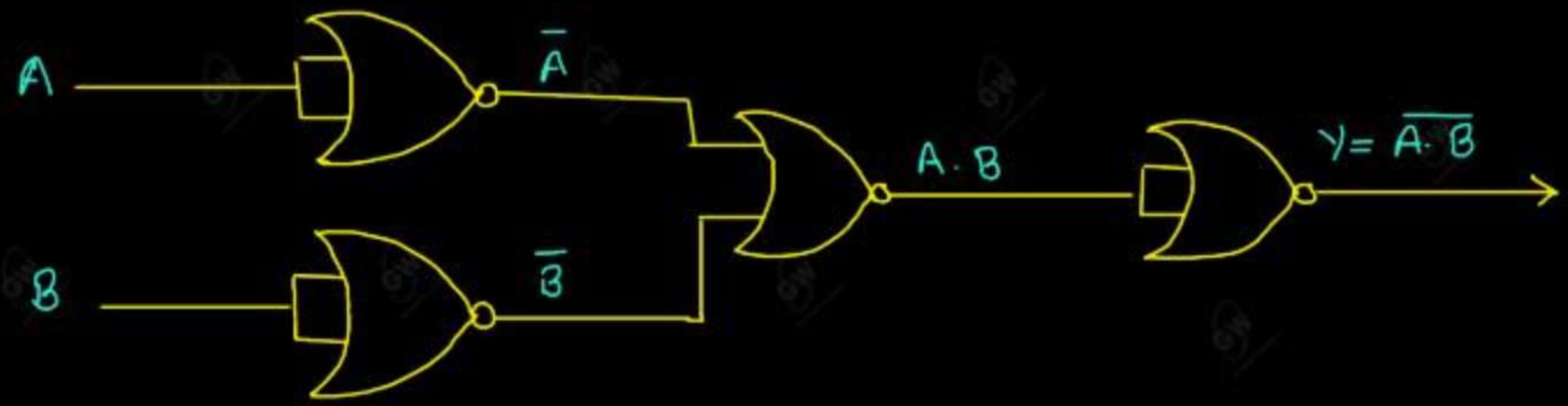


(3) AND gate using NOR gate

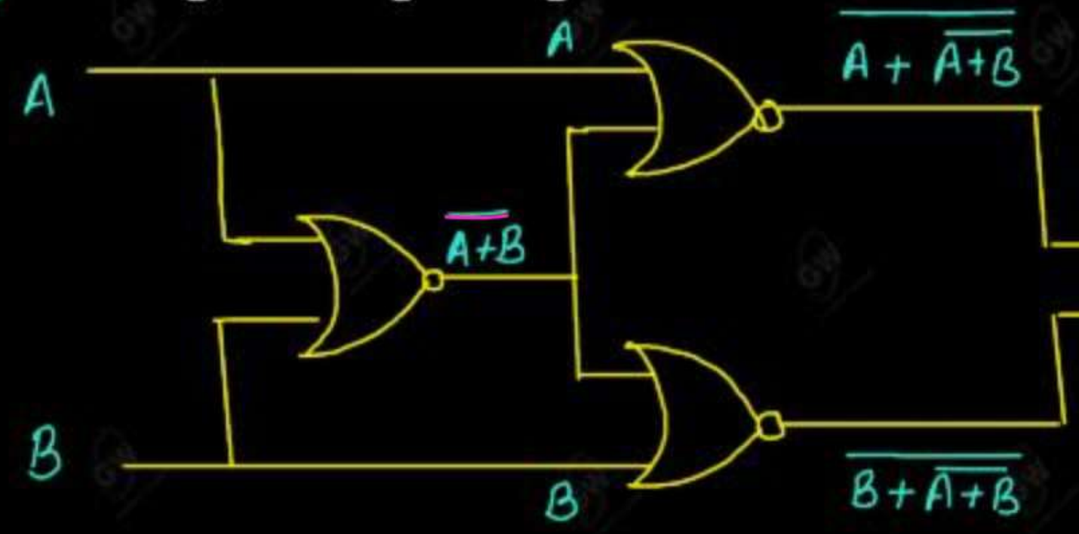


(4) NAND gate using NOR gate

$\overline{A \cdot B}$



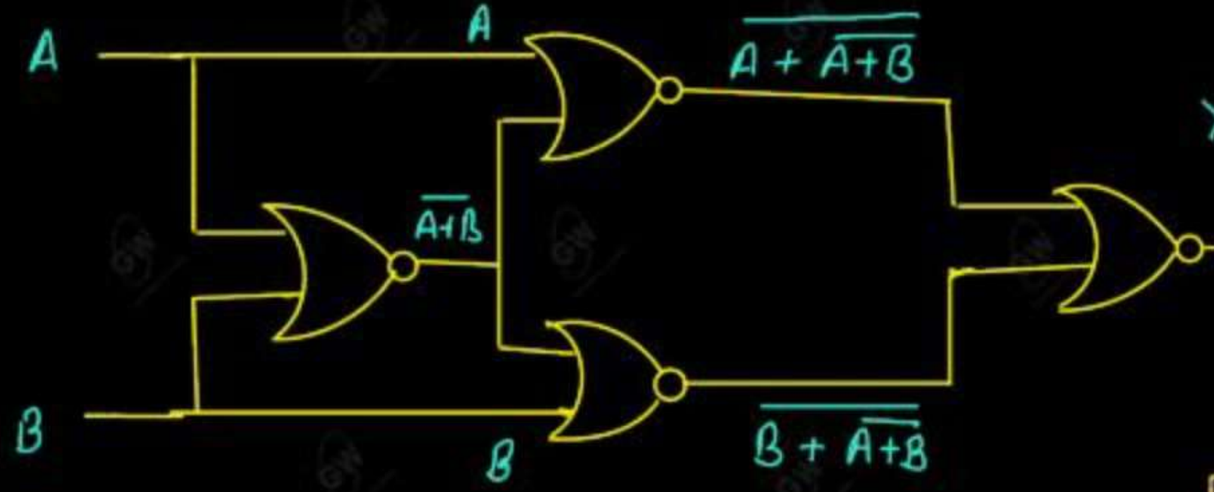
(5) EX-OR gate using NOR gate



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

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

(6) EXNOR gate using NOR gate



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

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

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

AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-1 Lec-5



Today's target

- Karnaugh maps up to 5 variables
- PYQ
- DPP

Avinash Sir

Gateway Classes

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

UNIT-1 : Logic simplification and combinational logic design

- Number Systems
- Binary arithmetic
- Signed magnitude representation
- Binary codes, code conversion
- Review of Boolean algebra and Demorgans theorem
- SOP & POS forms
- Canonical forms
- Karnaugh maps up to 5 variables
- Tabulation method.

Gateway Classes

Karnaugh Map (K-map)

- The K-map is a pictorial representation of truth table of the Boolean function and used to minimize the Boolean function
- K-map matrix contain 2^n cell, where n is the total number of variables

K-map

For SOP form

Each Cell contain a Minterm

0	Variable with complement $\bar{A}$
1	Variable without complement A

For POS form

Each cell contain a Maxterm

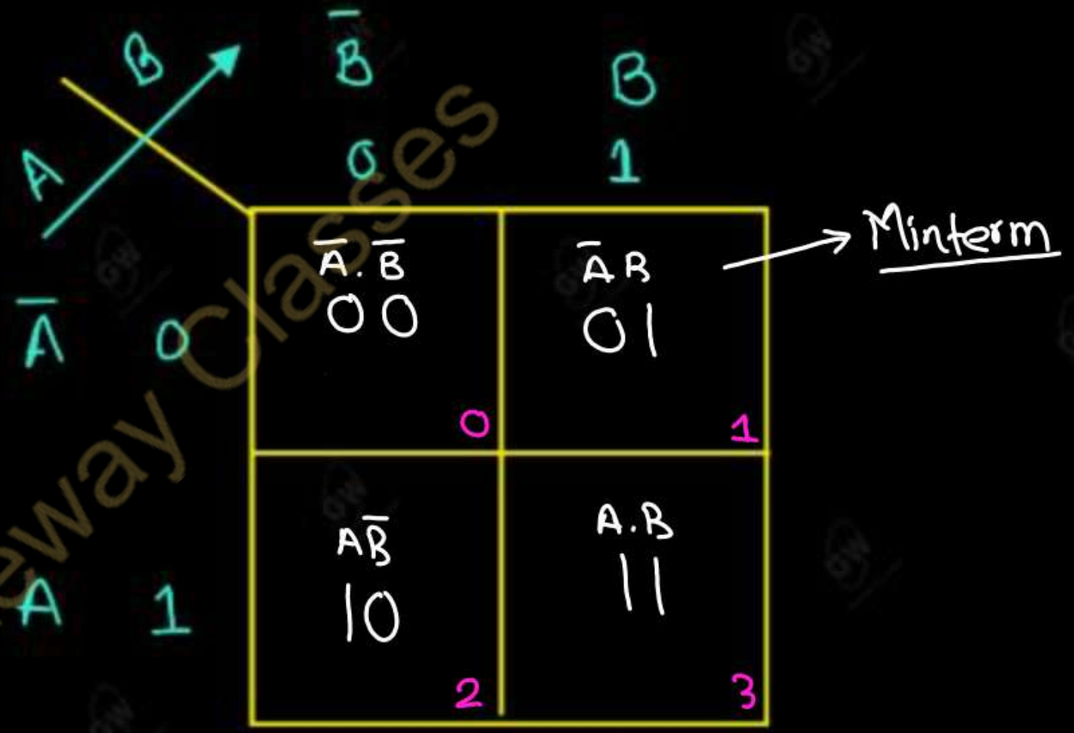
A	0 Variable without complement
$\bar{A}$	1 Variable with complement

Gateway Classes

K-Map for SOP Form

(1) For two variable A and B

- No of cells in K-map = $2^2 = 4$
- Each cell contain a Minterm



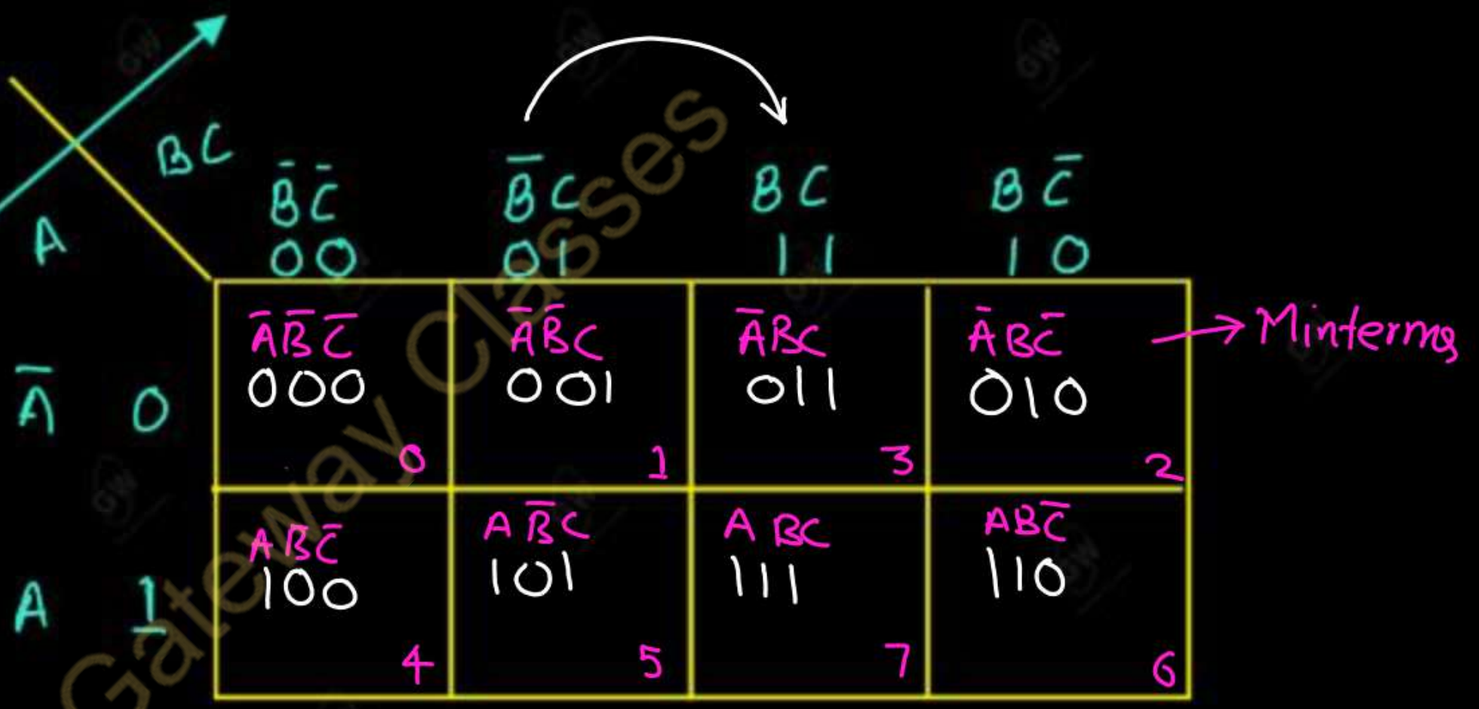
Gateway Classes

(2) For three variables A , B , C

- No of cells in K-map = $2^3 = 8$
- Each cell contain a Minterm

2nd last → Last
Last → 2nd Last

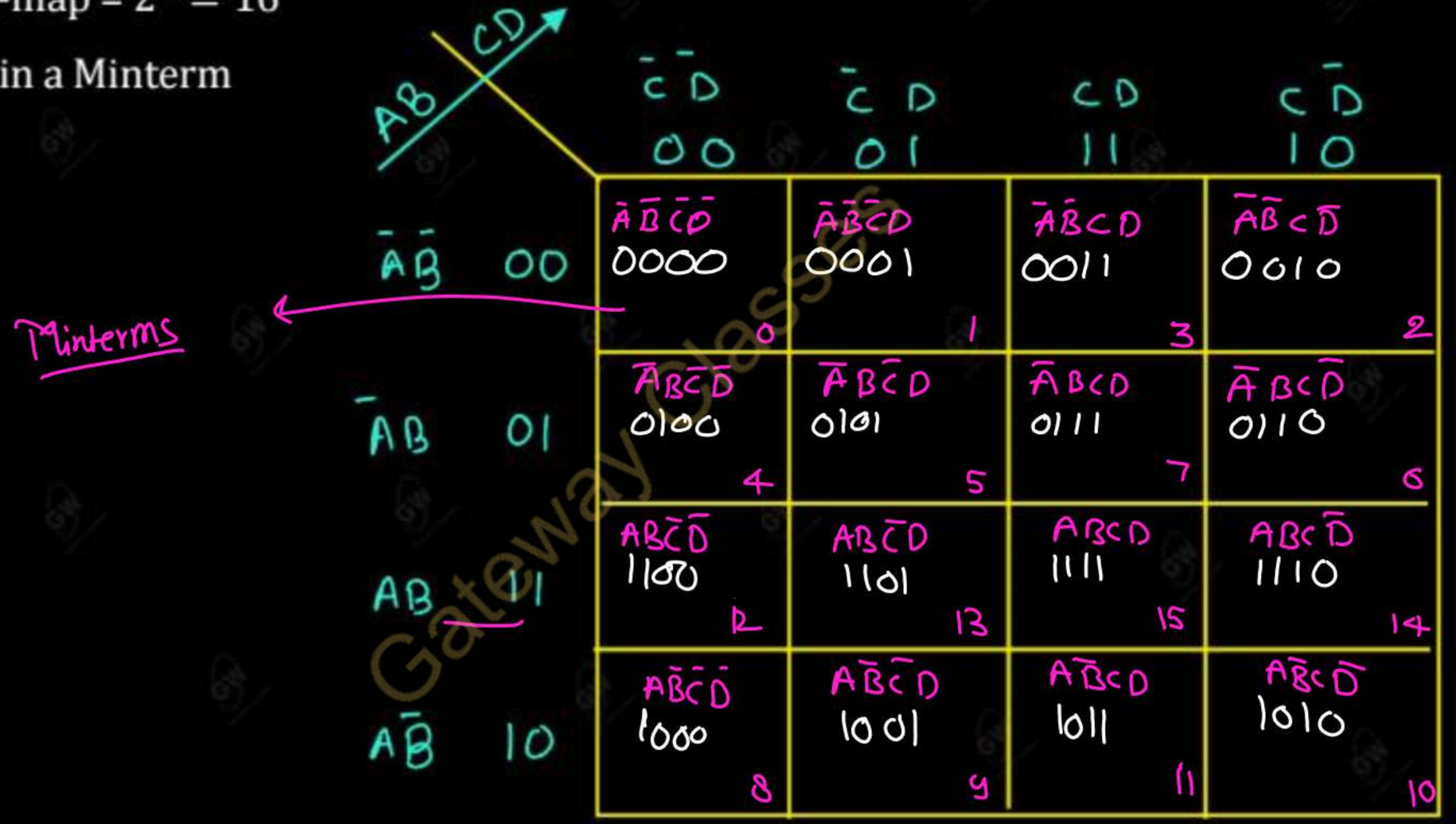
Gray code



Gateway Classes

(3) For four variables A , B , C , D

- No of cells in K-map = $2^4 = 16$
- Each cell contain a Minterm



Gateway Classes

Rule of Minimization

- Put 1 for each Minterm and 0 for each maxterm
- Formation of Groups : Form minimum number of groups and Form big size group

(1) 16 Group = 2^4 (4 Variables minimise)

(2) 8 Group = 2^3 (3 Variables minimise) octacte

(3) 4 Group = 2^2 (2 Variables minimise) Quad

(4) 1 Group = 2^0 (0 Variables minimise)

(5) 2 Group = 21 (1 4 4) Pair

GatewayClasses

Q.1 Simplify the following function using K-map $F(A, B) = \Sigma m(0, 1, 3)$

$F(A,B) = \Sigma m(0,1,3)$

$f(A,B) = \bar{A}\bar{B} + \bar{A}B + AB$

Minterms

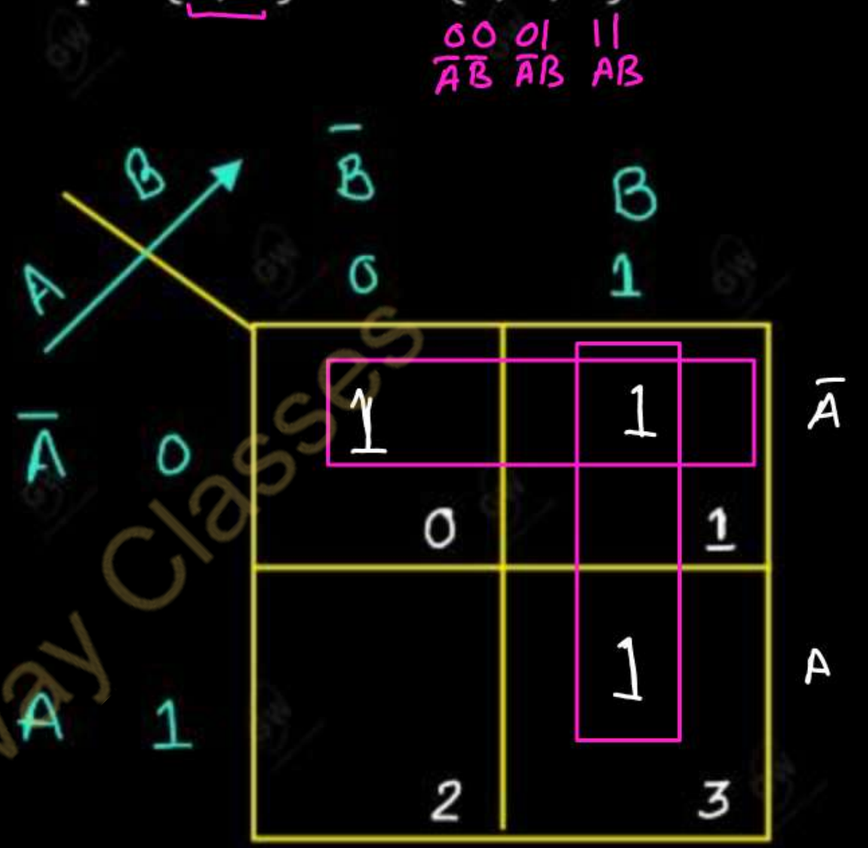
$\bar{A}(\bar{B}+B) + AB$

$\bar{A} + AB$

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

$1.(\bar{A}+B)$

$F = \bar{A} + B$



Gateway Classes

Q.2 Simplify the following function using K-map $F(A, B) = \Sigma m(0, 1, 2, 3)$

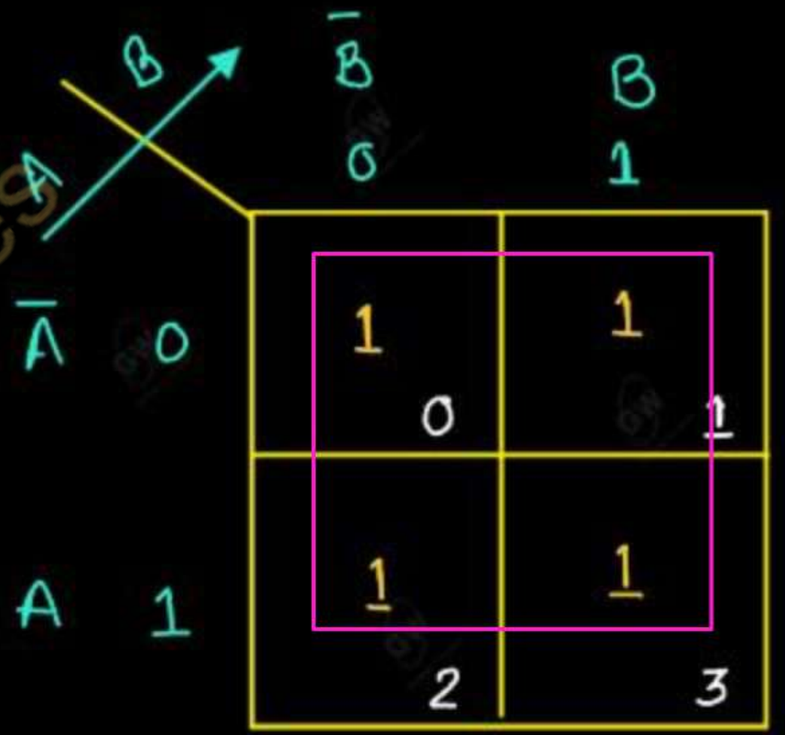
$\bar{A}\bar{B} + \bar{A}B + A\bar{B} + AB$
00 01 10 11

$F(A,B) = 1$

Diagonal ×

Vert. ✓

Horiz. ✓



Gateway Classes

Q.3 Simplify the following function using K-map $F(A, B, C) = \Sigma m(0, 1, 3, 6, 7)$

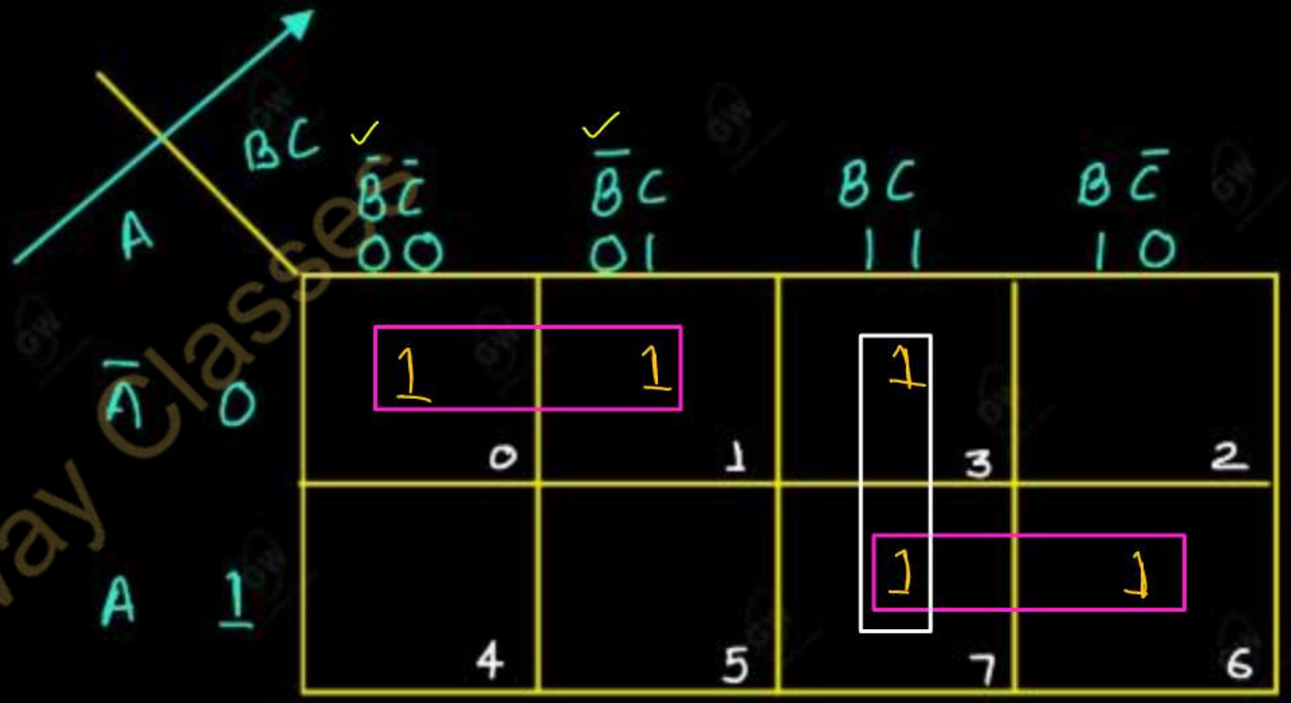
000 001 011 110 111

$\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + AB\bar{C} + ABC$

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

$\therefore$

No of Groups = No of minterms



Gateway Classes

Q.4 Simplify the logic function using K-map

AKTU-2022-23

$$Y = \sum m\,(0, 2, 3, 4, 6, 7, 9, 11, 16, 18, 19, 20, 22, 23, 25, 27)$$

$\bar{A}(B, C, D, E)$ $\quad A = 0$, $\bar{A}$

$$F = \bar{B}\bar{E} + \bar{B}D + B\bar{C}E$$

BC \ DE	$\bar{D}\bar{E}$ 00	$\bar{D}E$ 01	DE 11	$D\bar{E}$ 10
$\bar{B}\bar{C}$ 00	1 (0)	(1)	1 (3)	1 (2)
$B\bar{C}$ 01	1 (4)	(5)	1 (7)	1 (6)
BC 11	(12)	(13)	(15)	(14)
$B\bar{C}$ 10	(8)	1 (9)	1 (11)	(10)

$A = 1$, A

BC \ DE	$\bar{D}\bar{E}$ 00	$\bar{D}E$ 01	DE 11	$D\bar{E}$ 10
$B\bar{C}$ 00	1 (16)	(17)	1 (19)	1 (18)
$B\bar{C}$ 01	1 (20)	(21)	1 (23)	1 (22)
BC 11	(28)	(29)	(31)	(30)
$B\bar{C}$ 10	(24)	1 (25)	1 (27)	(26)

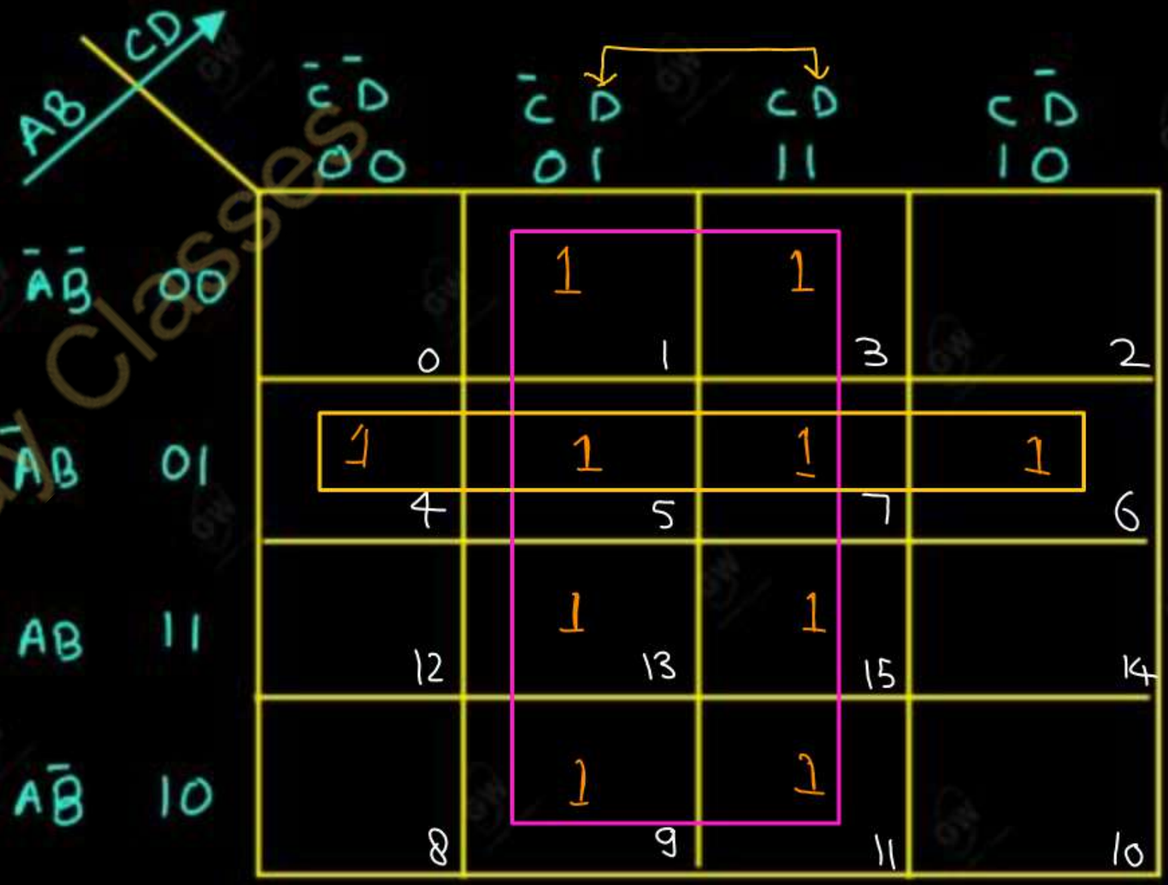
Gateway Classes

Q.5 Simplify the following function using K-map F (A, B, C, D) = Σ (1, 3, 4, 5, 6, 7, 9, 11, 13, 15).
Also implement the simplified function using basic gates only.

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

D → Octat

$\bar{A}B$ → Quad



Gateway Classes

Q.6 Simplify the following function using K-map

AKTU-2024- 25

$$F(A, B, C, D) = \Sigma(8, 9, 10, 11, 13, 15, 16, 18, 21, 24, 25, 26, 27, 30, 31).$$

$$F = B\bar{C} + \bar{A}BE + ABD + A\bar{B}C\bar{E} + A\bar{B}CD\bar{E}$$

E

$A = 0$ $\bar{A}$

BC \ DE	$\bar{D}\bar{E}$ 00	$\bar{D}E$ 01	DE 11	$D\bar{E}$ 10
$\bar{B}\bar{C}$ 00	0	1	3	2
$B\bar{C}$ 01	4	5	7	6
BC 11	12	13	15	14
$B\bar{C}$ 10	8	9	11	10

$A = 1$ A

BC \ DE	$\bar{D}\bar{E}$ 00	$\bar{D}E$ 01	DE 11	$D\bar{E}$ 10
$\bar{B}\bar{C}$ 00	16	17	19	18
$B\bar{C}$ 01	20	21	23	22
BC 11	28	29	31	30
$B\bar{C}$ 10	24	25	27	26

Gateway Classes

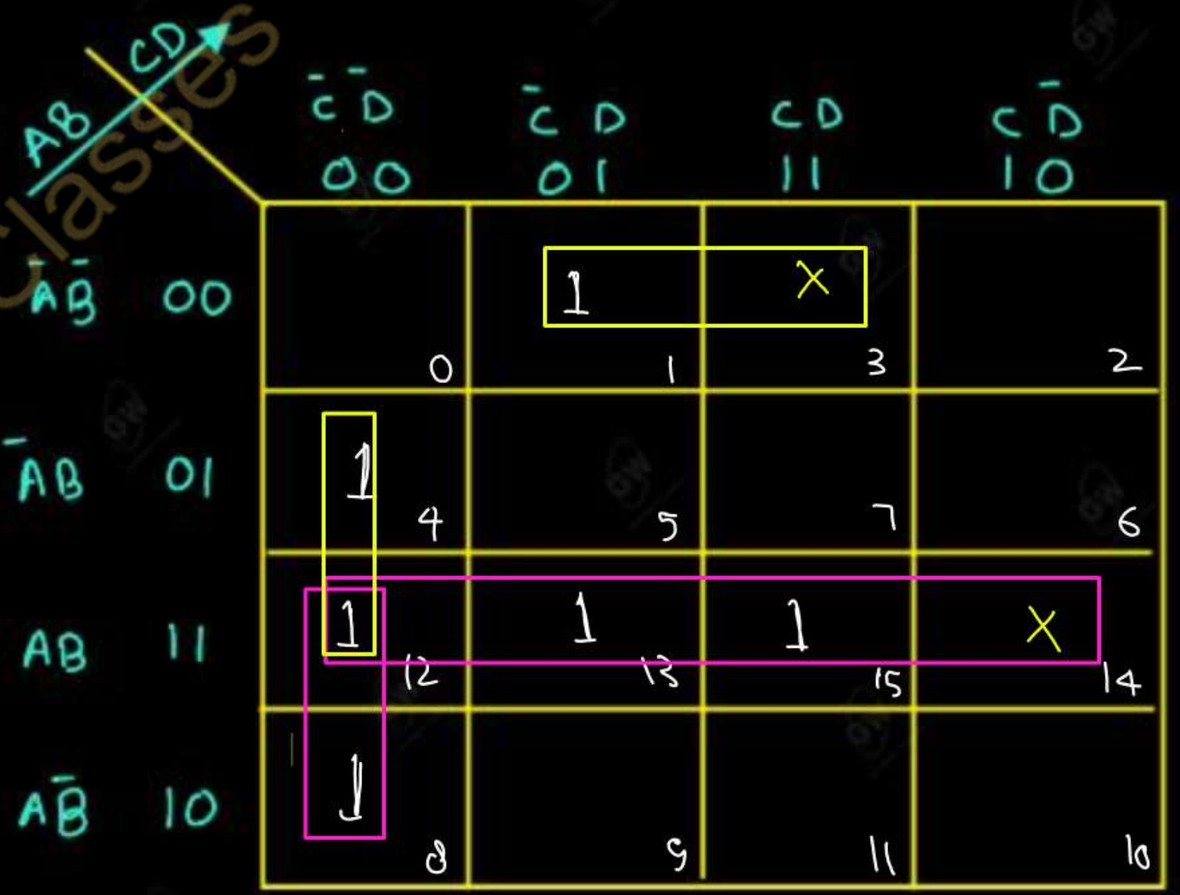
Don't Care Condition

Combination of inputs on which the output may or may not depends are called Don't Care Condition

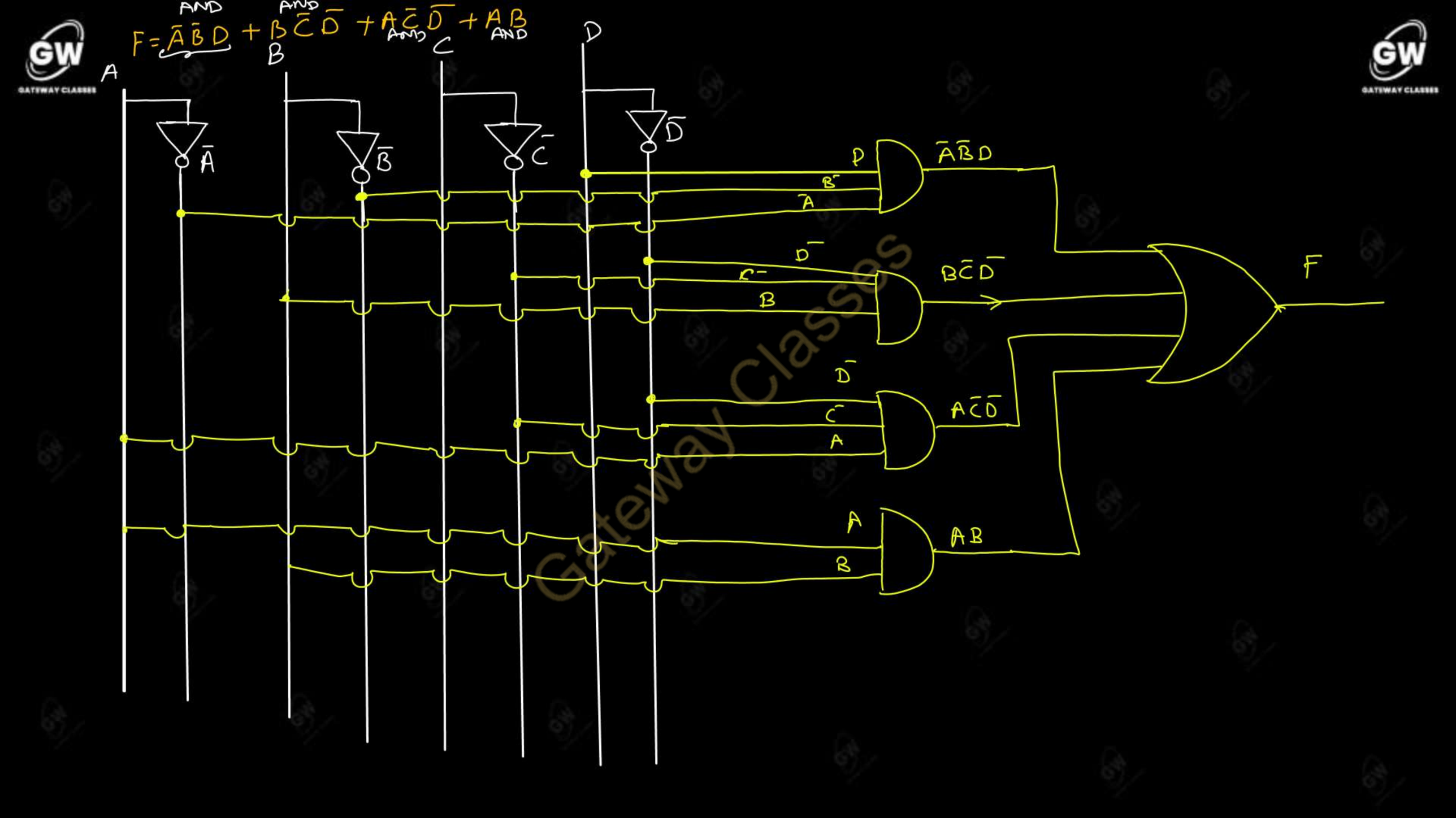
Q.7 Minimize using K-map and realize output using gates.

$$F(A, B, C, D) = \Sigma m(1, 4, 8, 12, 13, 15) + d(3, 14)$$

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

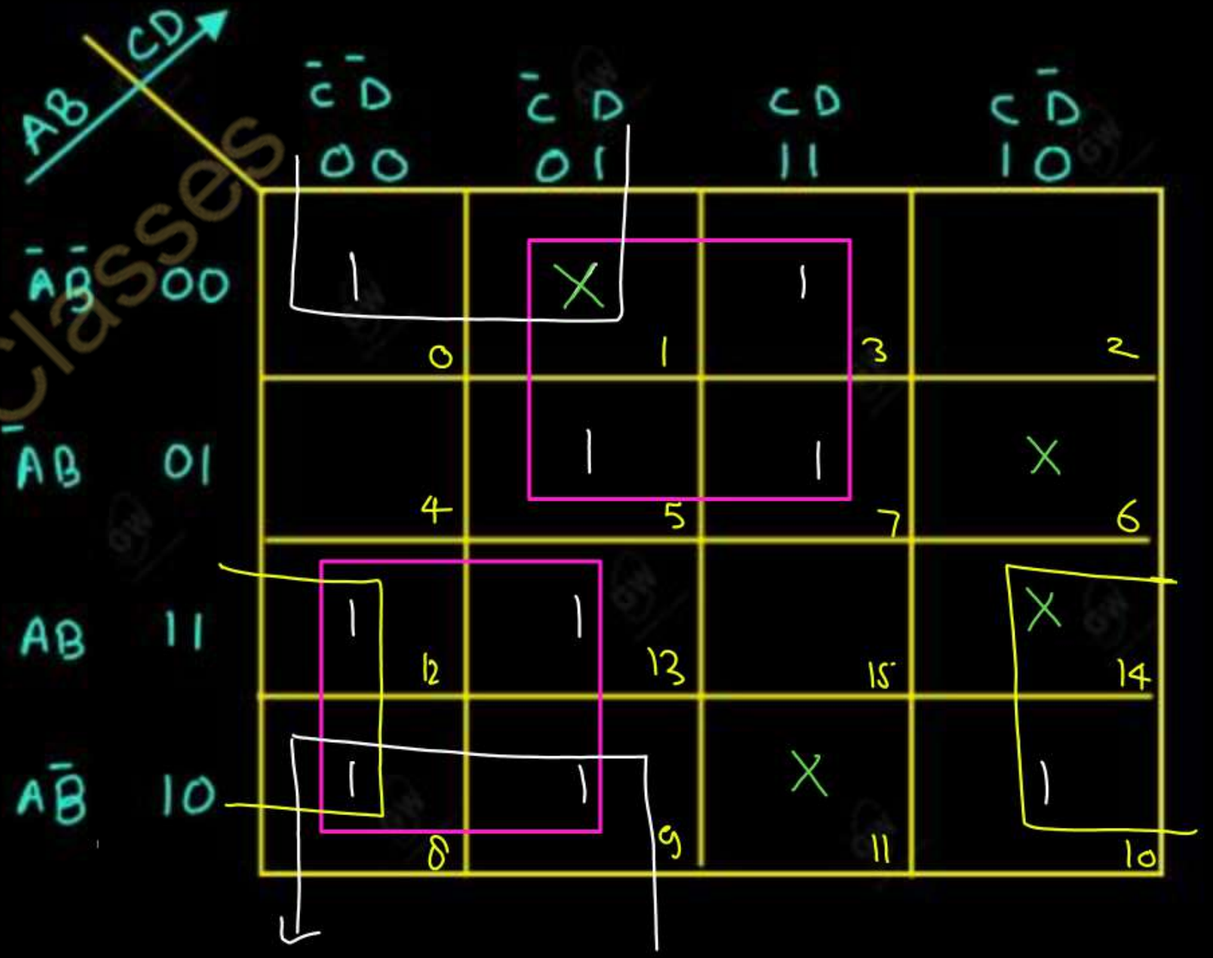


Gateway Classes

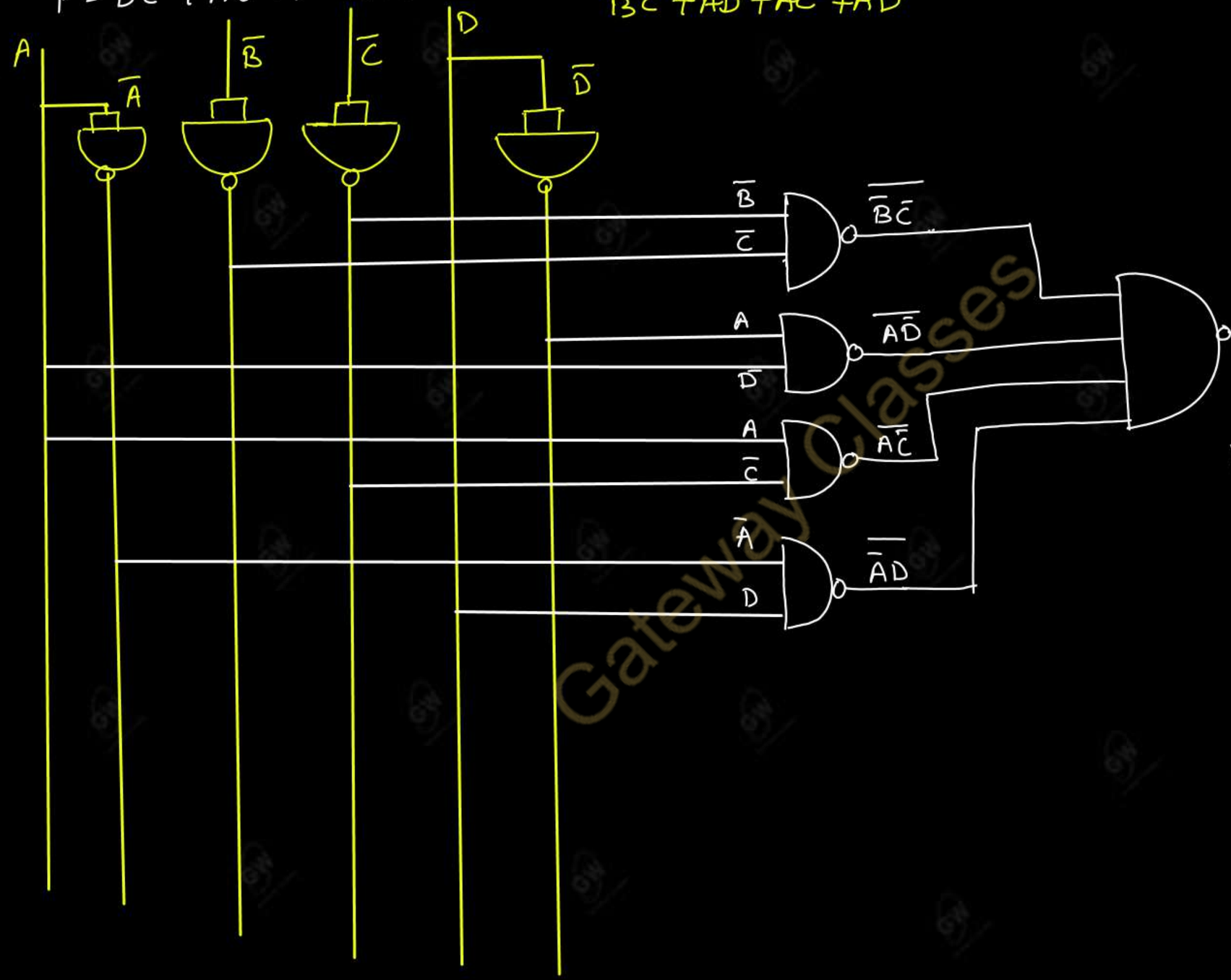

F = ĀB̄D + BC̄D̄ + AC̄D̄ + AB
AND
AND
AND
AND
A
B
C
D
Ā
B̄
C̄
D̄
D
B̄
Ā
ĀB̄D
D̄
C̄
B
BC̄D̄
D̄
C̄
A
AC̄D̄
A
B
AB
F

Q.8 Simplify the function $F = \Sigma m\,(0,3,5,7,8,9,10,12,13) + \Sigma d\,(1,6,11,14)$ using K-map and implement the simplified expression using NAND gates only

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



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



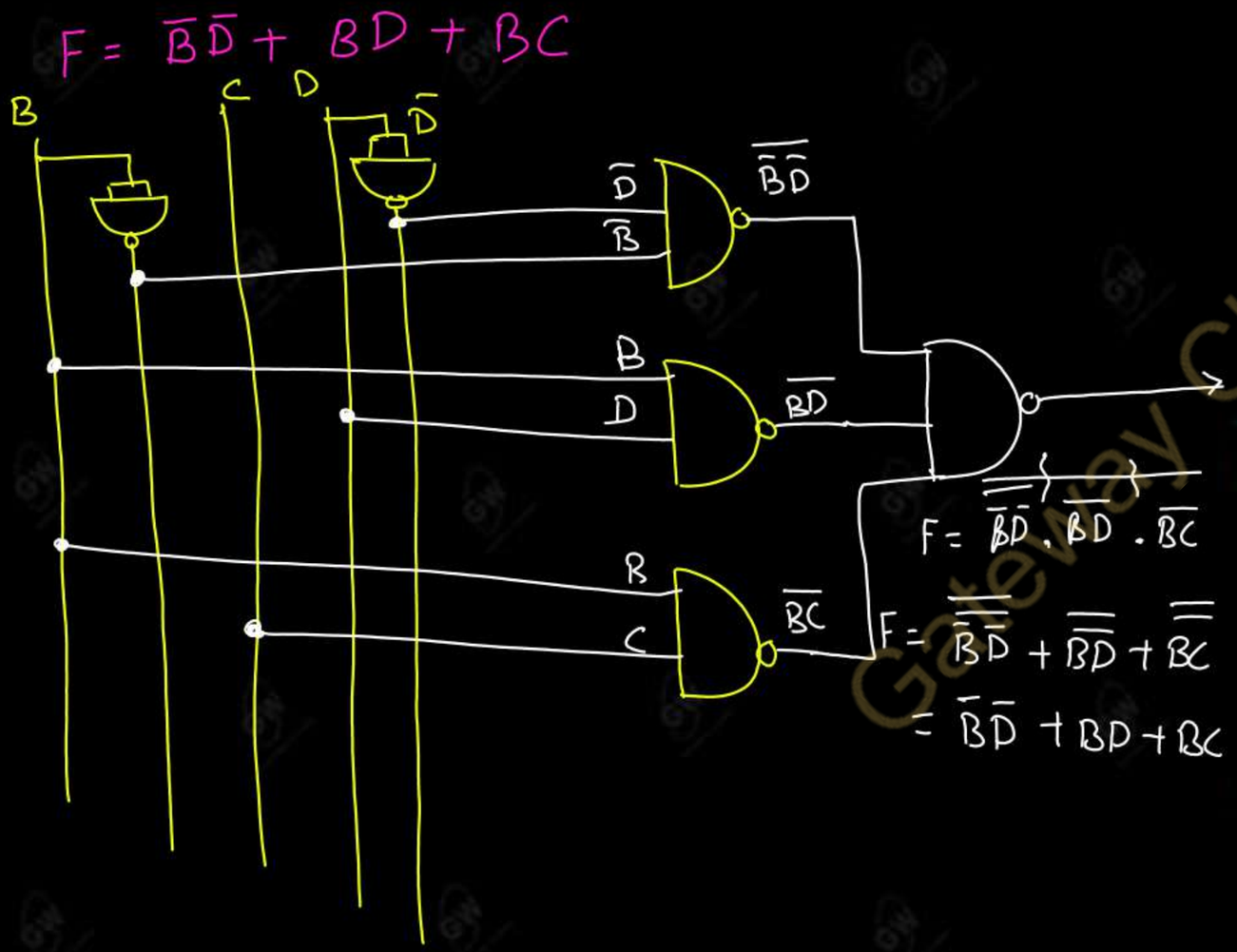
$\overline{\overline{\bar{B}\bar{C}} \cdot \overline{A\bar{D}} \cdot \overline{A\bar{C}}\ \overline{\bar{A}D}}$

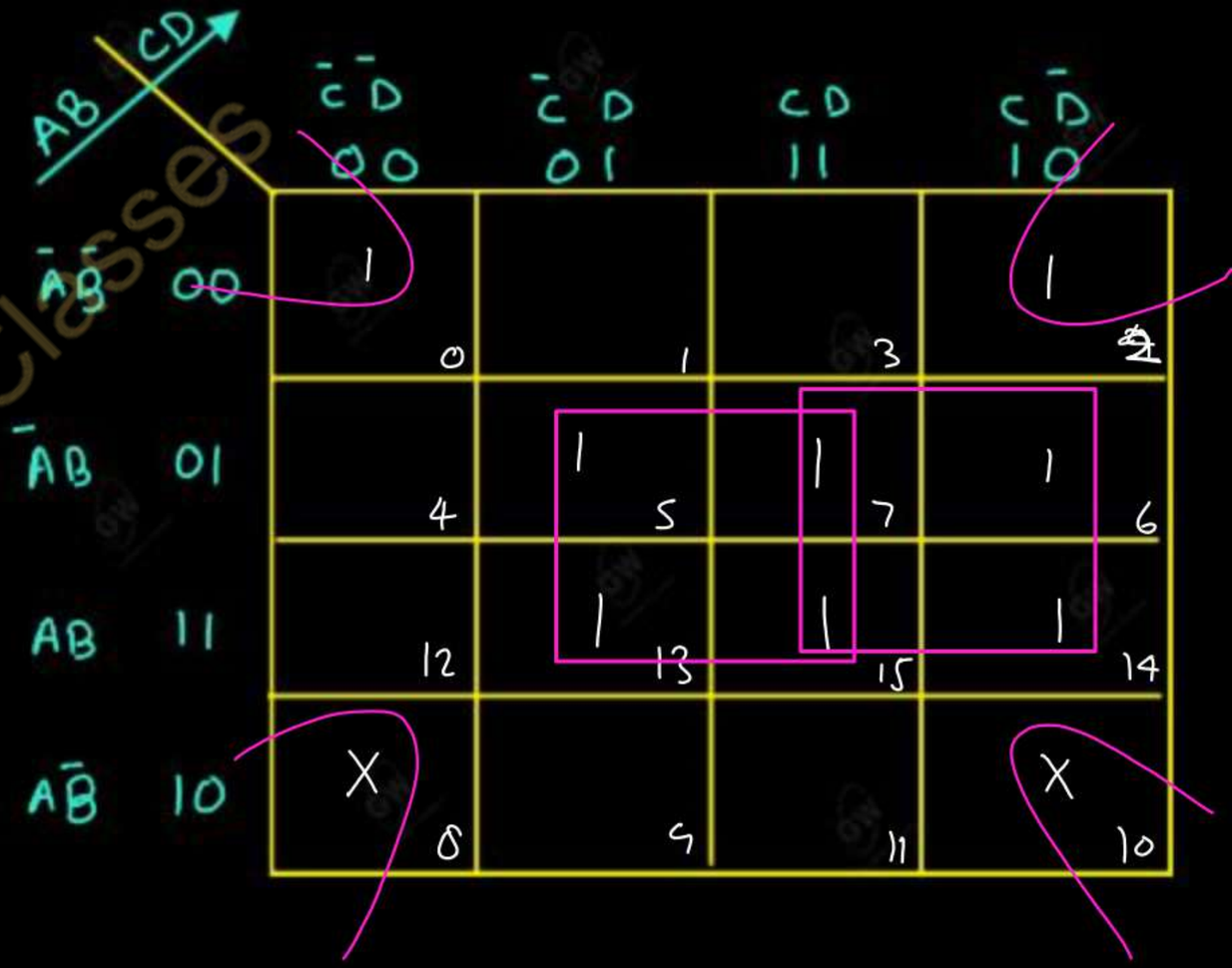
$F = \overline{\overline{\bar{B}\bar{C}}} + \overline{\overline{A\bar{D}}} + \overline{\overline{A\bar{C}}} + \overline{\overline{\bar{A}D}}$

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

Gateway Classes

Q.9 Simplify the function F (A, B, C, D) = Σ m(0, 2, 5, 6, 7, 13, 14, 15) + d (8, 10) using K-map and implement the simplified function using NAND gates only.





Gateway Classes

Q.10 - F (A, B, C, D, E) = Σm (0,1,2,4,5,6,10,13,14,18,21,22,24,26,29,30). Simplify the function with help of K-map and realize the simplified function using basic logic gates.

Do Yourself

A = 0

BC \ DE	$\bar{D}\bar{E}$ 00	$\bar{D}E$ 01	DE 11	$D\bar{E}$ 10
$\bar{B}\bar{C}$ 00				
$B\bar{C}$ 01				
BC 11				
$B\bar{C}$ 10				

A = 1

BC \ DE	$\bar{D}\bar{E}$ 00	$\bar{D}E$ 01	DE 11	$D\bar{E}$ 10
BC 00				
BC 01				
BC 11				
BC 10				

Gateway Classes

Q.11 Simplify the following Boolean expression using K-map.

$$f(V, W, X, Y, Z) = \sum m(0,7,8,9,12,13,15,16,22,23,30,31)$$

A B C D E

$$F = \bar{A}\bar{C}\bar{D}\bar{E} + \bar{A}B\bar{D} + \bar{A}CDE + A\bar{B}C\bar{D}\bar{E} + ACD$$

$\bar{A}$, A = 0

BC \ DE	$\bar{D}\bar{E}$ 00	$\bar{D}E$ 01	DE 11	$D\bar{E}$ 10
$\bar{B}\bar{C}$ 00	1 (0)	1	3	2
$B\bar{C}$ 01	4	5	1 (7)	6
BC 11	1 (12)	1 (13)	1 (15)	14
$B\bar{C}$ 10	1 (8)	1 (9)	11	10

A = 1

BC \ DE	$\bar{D}\bar{E}$ 00	$\bar{D}E$ 01	DE 11	$D\bar{E}$ 10
$\bar{B}\bar{C}$ 00	1 (16)	17	19	18
$B\bar{C}$ 01	20	21	1 (23)	1 (22)
BC 11	28	29	1 (31)	1 (30)
$B\bar{C}$ 10	24	25	27	26

Gateway Classes

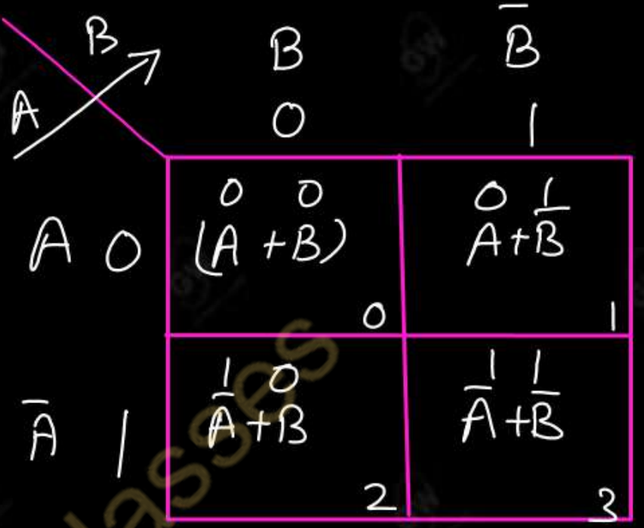
K-Map for POS Form

(1) For two variable A and B

- No of cells in K-map = $2^2 = 4$
- Each cell contain a Maxterm

0 ⟶ Un complemented

1 → Complemented.



Gateway Classes

(2) For three variables A , B , C

- No of cells in K-map = $2^3 = 8$
- Each cell contain a Maxterm

A \ BC	$B+C$ (00)	$B+\bar{C}$ (01)	$\bar{B}+\bar{C}$ (11)	$\bar{B}+C$ (10)
A (0)	$A+B+C$ 0	$A+B+\bar{C}$ 1	$A+\bar{B}+\bar{C}$ 3	$A+\bar{B}+C$ 2
$\bar{A}$ (1)	$\bar{A}+B+C$ 4	$\bar{A}+B+\bar{C}$ 5	$\bar{A}+\bar{B}+\bar{C}$ 7	$\bar{A}+\bar{B}+C$ 6

Gateway Classes

(3) For three variables A , B , C , D

- No of cells in K-map = $2^4 = 16$
- Each cell contain a Maxterm



Gateway Classes

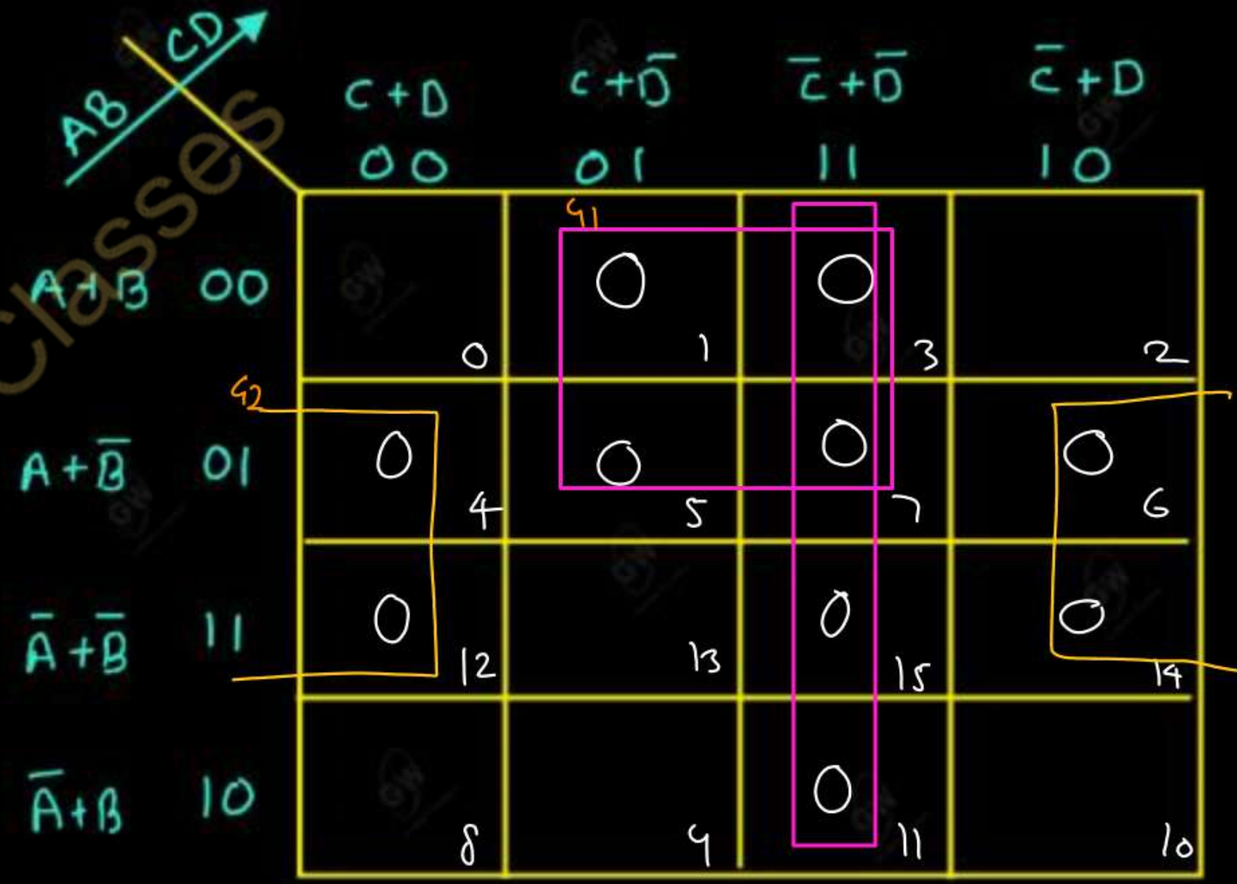
ΣM

GWQ.1 Simplify the Boolean expression using K-map and implement the simplified expression using NOR GATEs only F (A, B, C,D)=Π M(1,3,4,5,6,7,11,12,14,15)

0001 → (A+B+C+D̄)

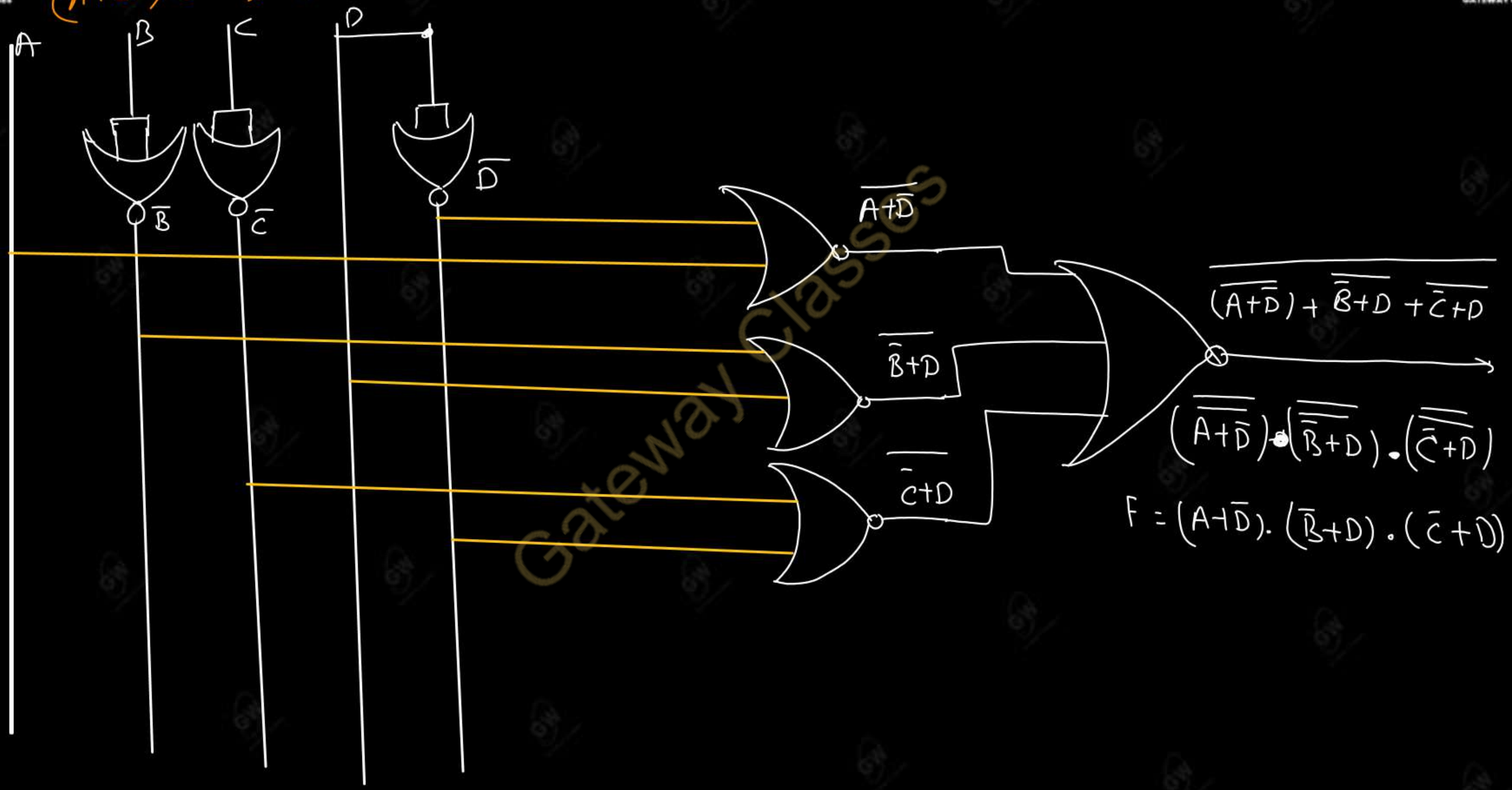
0011 → (A+B+C̄+D̄)

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



Gateway Classes

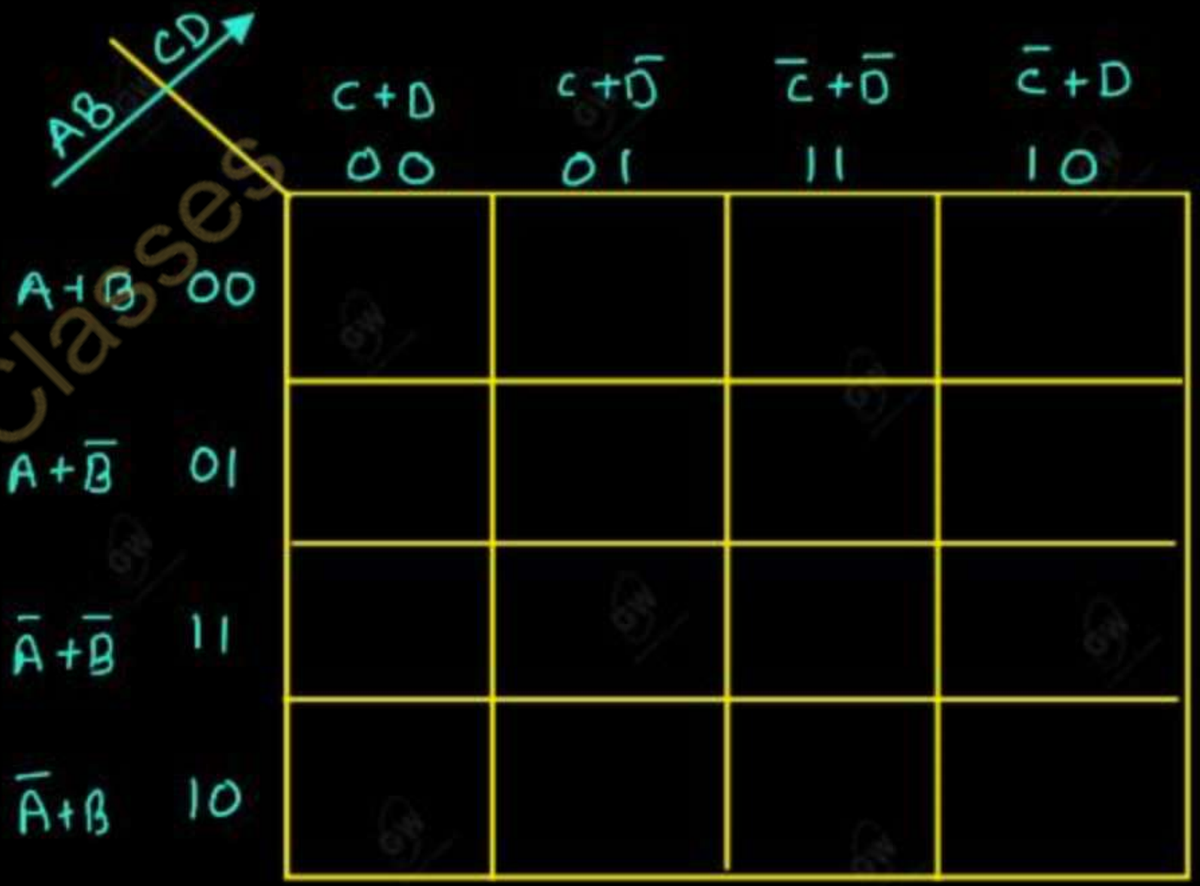
$(A+\bar{D})(\bar{B}+D)(\bar{C}+\bar{D})$



Gateway Classes

Q.2 Simplify the function F= ΠM (0, 2, 10, 11, 12, 14, 15)• Π d (5,13)using K-map and implement the simplified expression using NOR gates only.

Try yourself



Gateway Classes

Q.3 Minimize using K-map and realize using NOR gates only

$$F(A, B, C, D) = \Pi M(3, 4, 5, 7, 9, 13, 14, 15).\ d(0, 2, 8).$$

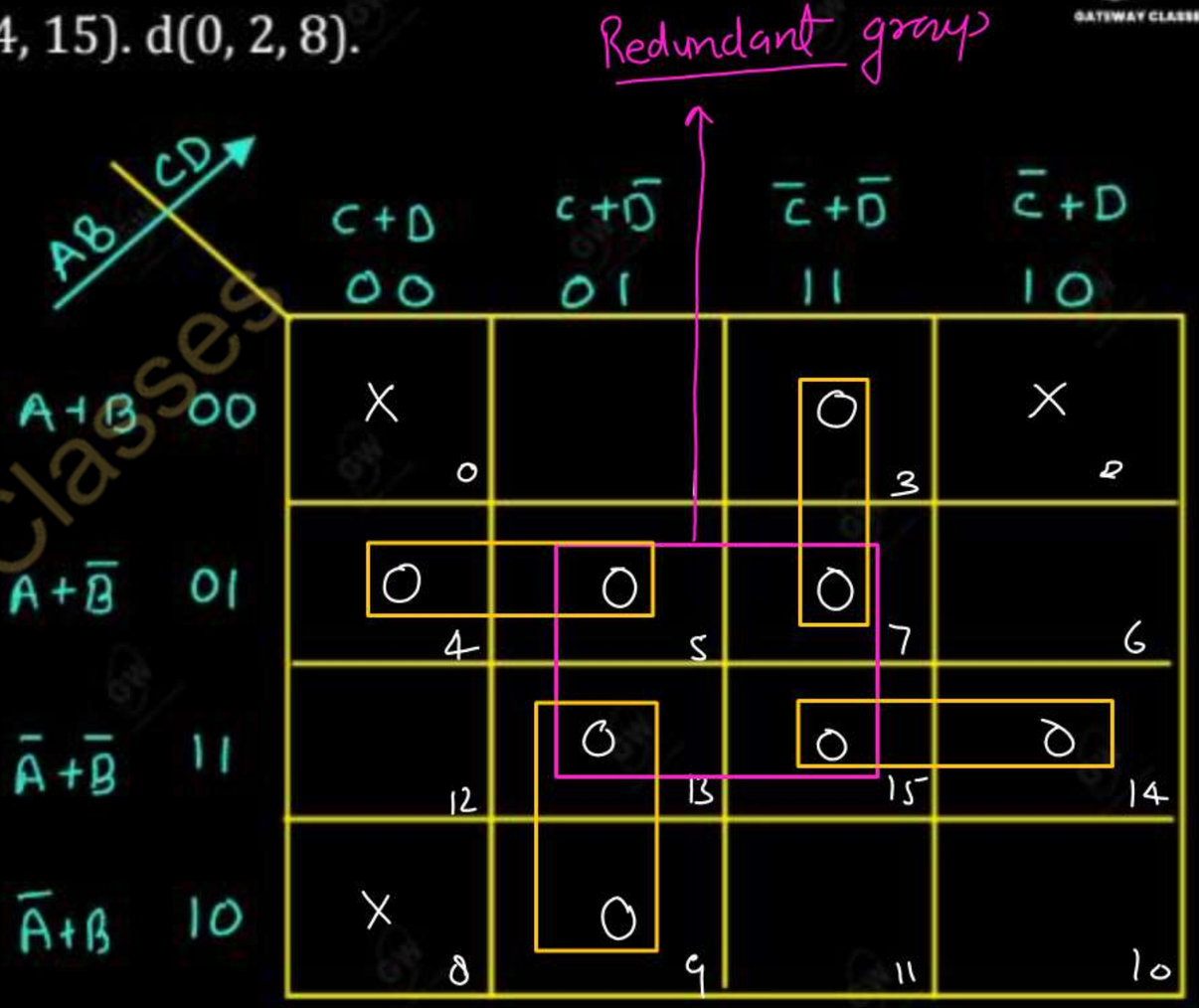
$$F = (A+\bar{B}+C) \cdot (A+\bar{C}+\bar{D}) \cdot (\bar{A}+C+\bar{D}) \cdot (\bar{A}+\bar{B}+\bar{C}) \left(\cdot (\overline{D}+\bar{B}) \right)_{\times}$$

Extra

Redundant Group (Ignore)

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

Ap



Gateway Classes

Unit-1 Lec-5

DPP

Q.1 State De Morgan's Law.

Q.2 Define minterm and maxterm.

Q.3 Minimize the following standard POS expression using K Map $Y = \Pi M (0, 2, 3, 5, 7)$

Q.4 Minimize the following expression using K Map $Y = \Sigma M (1, 5, 6, 7, 11, 12, 13, 15)$

Q.5 Simplify the Boolean Function $F = \sum m (0,1,2,5,7,8,9,10,13,15) + d (12,14)$ using K-map and implement the simplified expression using NAND gates only.

Gateway Classes

AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-1 Lec-6



Today's target

- Tabulation method.
- PYQ
- DPP

Gateway Classes

Avinash Sir

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

UNIT-1 : Logic simplification and combinational logic design

- Number Systems
- Binary arithmetic
- Signed magnitude representation
- Binary codes, code conversion
- Review of Boolean algebra and Demorgans theorem
- SOP & POS forms
- Canonical forms
- Karnaugh maps up to 5 variables
- Tabulation method.

Gateway Classes

Quine–McCluskey method (tabulation method)

- It is a systematic approach for minimizing Boolean functions, particularly useful when dealing with functions involving more than four variables.
- This method is advantageous over Karnaugh Maps (K-maps) in such scenarios due to its algorithmic nature, making it suitable for both manual calculations and computer implementations .

Advantages

- Systematic Approach: Provides a clear, step-by-step procedure for minimization.
- Scalability: More effective than K-maps for functions with a large number of variables.
- Computer Implementation: Its algorithmic nature makes it suitable for computer-based logic minimization tools.

Limitations

- Computational Complexity: The method can become computationally intensive for functions with a very large number of variables, as the number of possible prime implicants can grow exponentially .

Gateway Classes

Prime Implicant (PI)

prime implicant is a group of a minterms which cannot be combined with any other minterm or groups.

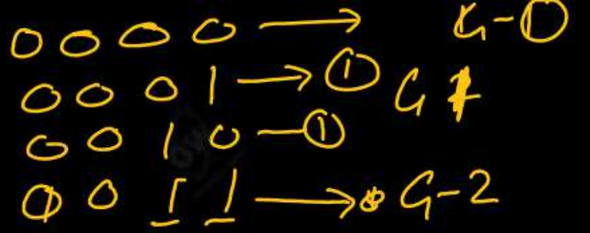
Essential Prime Implicant (EPI)

The essential prime implicant is a prime implicant in which one or more minterms are unique ,i.e it consist at least one minterm which is not contained in any other prime implicant. Tne Quine-McCluskey technique consists of two parts:

(i) First is to find all the prime implicants by an exhaustive search.

(ii) To identify the essential prime implicants obtained in part i and select from them the remaining prime implicants which can give the perfectly minimized expression.

Gateway Classes



The method involves the following key steps:

1) List Minterms: Begin by listing all the minterms (i.e., combinations of variables where the function outputs of the Boolean function.

2) Group Minterms: Organize these minterms based on the number of ones in their binary representation.

3) Combine Minterms: Iteratively combine minterms that differ by only one bit, replacing the differing bit with a dash (–) to indicate a don't-care condition. This process identifies the prime implicants.

4) Prime Implicant Chart: Construct a chart to determine which prime implicants cover which minterms.

5) Select Essential Prime Implicants: Identify and select essential prime implicants that cover minterms not covered by any other implicant.

6) Derive Simplified Expression: Use the selected prime implicants to write the minimized Boolean expression.

This method ensures a systematic and deterministic approach to Boolean function minimization, which is particularly beneficial for complex functions .

Gateway Classes

Q.1 Simplify the following function using Tabular method

$$F(A, B, C, D) = \Sigma m(0, 1, 3, 7, 8, 9, 11, 15)$$

0000
0001
0011
0111
1000
1001
1011
1111

Group	Minterm	A	B	C	D	
G-0	m_0	0	0	0	0	✓
G-1	m_1	0	0	0	1	✓
	m_8	1	0	0	0	✓
G-2	m_3	0	0	1	1	✓
	m_9	1	0	0	1	✓
G-3	m_7	0	1	1	1	✓
	m_{11}	1	0	1	1	✓
G-4	m_{15}	1	1	1	1	✓

Group	Minterm	A	B	C	D	
G0	$m_0 - m_1$	0	0	0	—	✓
	$m_0 - m_8$	—	0	0	0	✓
G1	$m_1 - m_3$	0	0	—	1	✓
	$m_1 - m_9$	—	0	0	1	✓
	$m_8 - m_9$	1	0	0	—	✓
G2	$m_3 - m_7$	0	—	1	1	✓
	$m_3 - m_{11}$	—	0	1	1	✓
	$m_9 - m_{11}$	1	0	—	1	✓
G3	$m_7 - m_{15}$	—	1	1	1	✓
	$m_{11} - m_{15}$	1	—	1	1	✓

Gateway Classes

Exp $F = \bar{B}\bar{C} + \bar{B}D + CD$

Final Simplified Expression

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

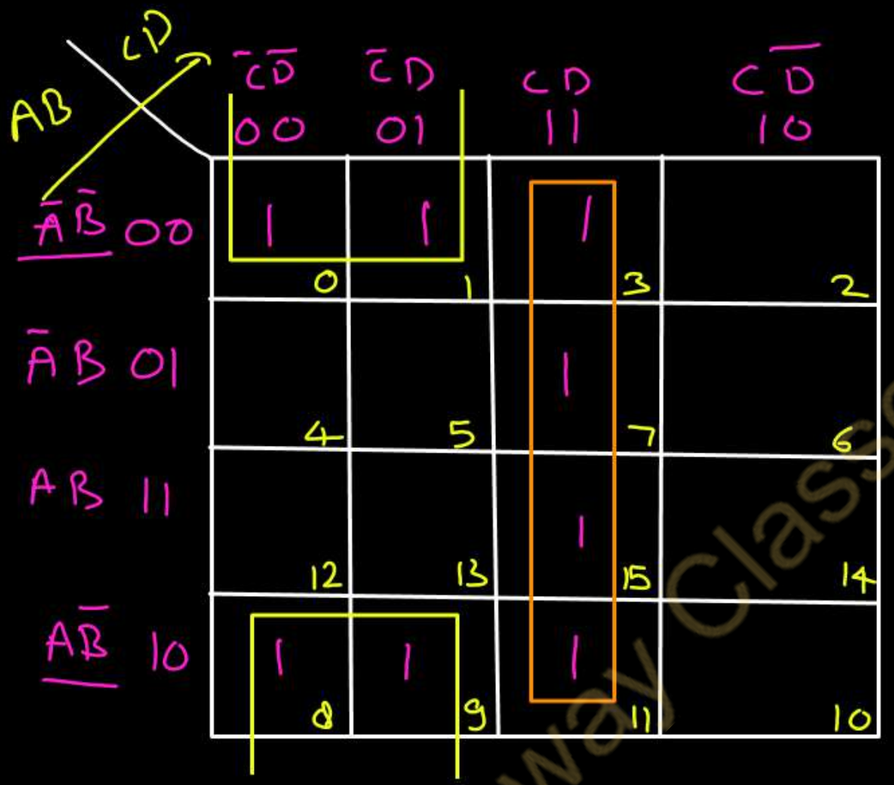
Group	Minterm	A	B	C	D	
G0	m_0-m_1-m_8-m_9	—	0	0	—	$\bar{B}\bar{C}$
	m_0-m_8-m_1-m_9	—	0	0	—	$\bar{B}\bar{C}$
G1	m_1-m_3-m_9-m_{11}	—	0	—	1	$\bar{B}D$
	m_1-m_9-m_3-m_{11}	—	0	—	1	$\bar{B}D$
G2	m_3-m_7-m_{11}-m_{15}	—	—	1	1	CD
	m_3-m_{11}-m_7-m_{15}	—	—	1	1	CD

PI

P.I	Minterms involve in PI	Essential P.I Given Minterms							
		0	1	3	7	8	9	11	15
$\bar{B}\bar{C}$	0,1,8,9	(X)	X			(X)	X		
$\bar{B}D$	1,3,9,11		X	X			X	X	
CD	3,7,11,15			X	(X)			X	(X)

Gateway Classes

Verified by K map



$F = \bar{B}\bar{C} + CD$

OK

Gateway Classes

Q.2 Simplify the following function using Tabular method

Practice $Y(A, B, C, D) = \Sigma m(0, 1, 2, 3, 5, 7, 8, 9, 11, 14)$

Group	Minterm	A	B	C	D	
	m_0					
	m_1					

Group	Minterm	A	B	C	D	

Gateway Classes

Group	Minterm	A B C D	

Gateway Classes

Q.3 Simplify the logic function using Tabular method

$$f(A, B, C, D) = \sum m\,(0, 1, 4, 6, 8, 9, 10, 12) + d(5, 7, 14).$$

Group	Minterm	A	B	C	D	
G_0	m_0	0	0	0	0	✓
G_1	m_1	0	0	0	1	✓
	m_4	0	1	0	0	✓
	m_8	1	0	0	0	✓
G_2	m_5*	0	1	0	1	✓
	m_6	0	1	1	0	✓
	m_9	1	0	0	1	✓
	m_{10}	1	0	1	0	✓
	m_{12}	1	1	0	0	✓
G_3	m_7*	0	1	1	1	✓
	m_{14}*	1	1	1	0	

0	0 0 0 0
1 ✓	0 0 0 1
4 ✓	0 1 0 0
6 ✓	0 1 1 0
8 ✓	1 0 0 0
9 ✓	1 0 0 1
10 ✓	1 0 1 0
12 ✓	1 1 0 0
5	0 1 0 1
7	0 1 1 1
A	1 1 1 0

Group	Minterm	A	B	C	D	
G_0	$m_0 - m_1$	0	0	0	—	✓
	$m_0 - m_4$	0	—	0	0	✓
	$m_0 - m_8$	—	0	0	0	✓
G_1	$m_1 - m_5$	0	—	0	1	✓
	$m_1 - m_9$	—	0	0	1	✓
	$m_4 - m_5$	0	1	0	—	✓
	$m_4 - m_6$	0	1	—	0	✓
	$m_4 - m_{12}$	—	1	0	0	✓
	$m_8 - m_9$	1	0	0	—	✓
	$m_8 - m_{10}$	1	0	—	0	✓
	$m_8 - m_{12}$	1	—	0	0	✓
G_2	$m_5 - m_7$	0	1	—	1	✓
	$m_6 - m_7$	0	1	1	—	✓
	$m_6 - m_{14}$	—	1	1	0	✓
	$m_{10} - m_{14}$	1	—	1	0	✓
	$m_{12} - m_{14}$	1	1	—	0	✓

Gateway Classes

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

$\underline{0}, \underline{1}, (\underline{4}, \underline{6}), \underline{8}, \underline{9}, \underline{10}, \underline{12}$

Don't use don't care in this table.

Group	Minterm	A	B	C	D	
G0	$m_0 - m_1 - m_4 - m_5$	0	—	0	—	$\times \bar{A}\bar{C}$
	$m_0 - m_1 - m_8 - m_9$	—	0	0	—	$\times \bar{B}\bar{C}$
	$m_0 - m_4 - m_1 - m_5$	0	—	0	—	$\times \bar{A}\bar{C}$
	$m_0 - m_4 - m_8 - m_{12}$	—	—	0	0	$\times \bar{C}\bar{D}$
	$m_0 - m_8 - m_1 - m_9$	—	0	0	—	$\times \bar{B}\bar{C}$
	$m_0 - m_8 - m_4 - m_{12}$	—	—	0	0	$\times \bar{C}\bar{D}$
G1	$m_4 - m_5 - m_6 - m_7$	0	1	—	—	$\bar{A}B$
	$m_4 - m_6 - m_5 - m_7$	0	1	—	—	$\bar{A}B$
	$m_4 - m_6 - m_{12} - m_{14}$	—	1	—	0	$B\bar{D}$
	$m_4 - m_{12} - m_6 - m_{14}$	—	1	—	0	$B\bar{D}$
	$m_8 - m_{10} - m_{12} - m_{14}$	1	—	—	0	$A\bar{D}$
	$m_8 - m_{12} - m_{10} - m_{14}$	1	—	—	0	$A\bar{D}$

PI	minterm involved in PI	Essential PI							
		0	1	4	6	8	9	10	12
$\bar{A}\bar{C}$	0, 1, 4, 5	×	×	×					
$\bar{B}\bar{C}$	0, 1, 8, 9	×	×			×	×		
$\bar{C}\bar{D}$	0, 4, 8, 12	×		×		×			×
$\bar{A}B$	4, 5, 6, 7			×	×				
$B\bar{D}$	4, 6, 12, 14			×	×				×
$A\bar{D}$	8, 10, 12, 14					×		×	×

$\bar{B}\bar{C} = 0, 1, 8, 9$

$A\bar{D} = 8, 10, 12, 14$

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

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

Ans.

GW GATEWAY CLASSES
Gateway Classes

Q.4 Minimize $y = \sum m(0,1,2,3,5,6,7,8,14,15) + d(4,11,13)$ using tabular method

AKTU-2022-23

GATEWAY CLASSES

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

Grp	Minterm	A	B	C	D	
G_0	m_0	0	0	0	0	✓
G_1	m_1	0	0	0	1	✓
	m_2	0	0	1	0	✓
	m_4	0	1	–	0	✓
	m_8	1	0	0	0	✓
G_2	m_3	0	0	1	1	✓
	m_5	0	1	0	1	✓
	m_6	0	1	1	0	✓
G_3	m_7	0	1	1	1	✓
	m_{11}	1	0	1	1	✓
	m_{13}	1	1	0	1	✓
	m_{14}	1	1	1	0	✓
G_4	m_{15}	1	1	1	1	✓

Grp	Minterm	A	B	C	D	
G_0	$m_0 - m_1$	0	0	0	–	✓
	$m_0 - m_2$	0	0	–	0	✓
	$m_0 - m_4$	0	–	0	0	✓
$\bar{B}\bar{C}\bar{D}$	$m_0 - m_8$	–	0	0	0	✗
G_1	$m_1 - m_3$	0	0	–	1	✓
	$m_1 - m_5$	0	–	0	1	✓
	$m_2 - m_3$	0	0	1	–	✓
	$m_2 - m_6$	0	–	1	0	✓
	$m_4 - m_5$	0	1	0	–	✓
	$m_4 - m_6$	0	1	–	0	✓
G_2	$m_3 - m_7$	0	–	1	1	✓
	$m_3 - m_{11}$	–	0	1	1	✓
	$m_5 - m_7$	0	1	–	1	✓
	$m_5 - m_{13}$	–	1	0	1	✓
	$m_6 - m_7$	0	1	1	–	✓
	$m_6 - m_{14}$	–	1	1	0	✓
G_3	$m_7 - m_{15}$	–	1	1	1	✓
	$m_{11} - m_{15}$	1	–	1	1	✓
	$m_{13} - m_{15}$	1	1	–	1	✓
	$m_{14} - m_{15}$	1	1	1	–	✓

G.P	Minterm	A	B	C	D	
G_0	0-1-2-3	0	0	–	–	✓
	0-1-4-5	0	–	0	–	✓
	0-2-1-3	0	0	–	–	✓
	0-2-4-6	0	–	–	0	✓
	0-4-1-5	0	–	0	–	✓
	0-4-2-6	0	–	–	0	✓
G_1	1-3-5-7	0	–	–	1	✓
	1-5-3-7	0	–	–	1	✓
	2-3-6-7	0	–	1	–	✓
	2-6-3-7	0	–	1	–	✓
	4-5-6-7	0	1	–	–	✓
	4-6-5-7	0	1	–	–	✓
G_2	3-7-11-15	–	–	1	1	✗ (CD)
	3-11-7-15	–	–	1	1	✗ (CD)
	5-7-13-15	–	1	–	1	✗ (BD)
	5-13-7-15	–	1	–	1	✗ (BD)
	6-7-14-15	–	1	1	–	✗ (BC)
	6-14-7-15	–	1	1	–	✗ (BC)

0-8 ⇒ $\bar{B}\bar{C}\bar{D}$ 6-7-14-15 ⇒ BC

3-7-11-15 ⇒ CD , 5-7-13-14 ⇒ BD

Gatewayclasses

$F = \bar{A} + \bar{B}\bar{C}\bar{D} + BC + CD + BD$

$\bar{A} = 0, 1, 2, 3, 4, 5, 6, 7$

$\bar{B}\bar{C}\bar{D} = 0, 8$

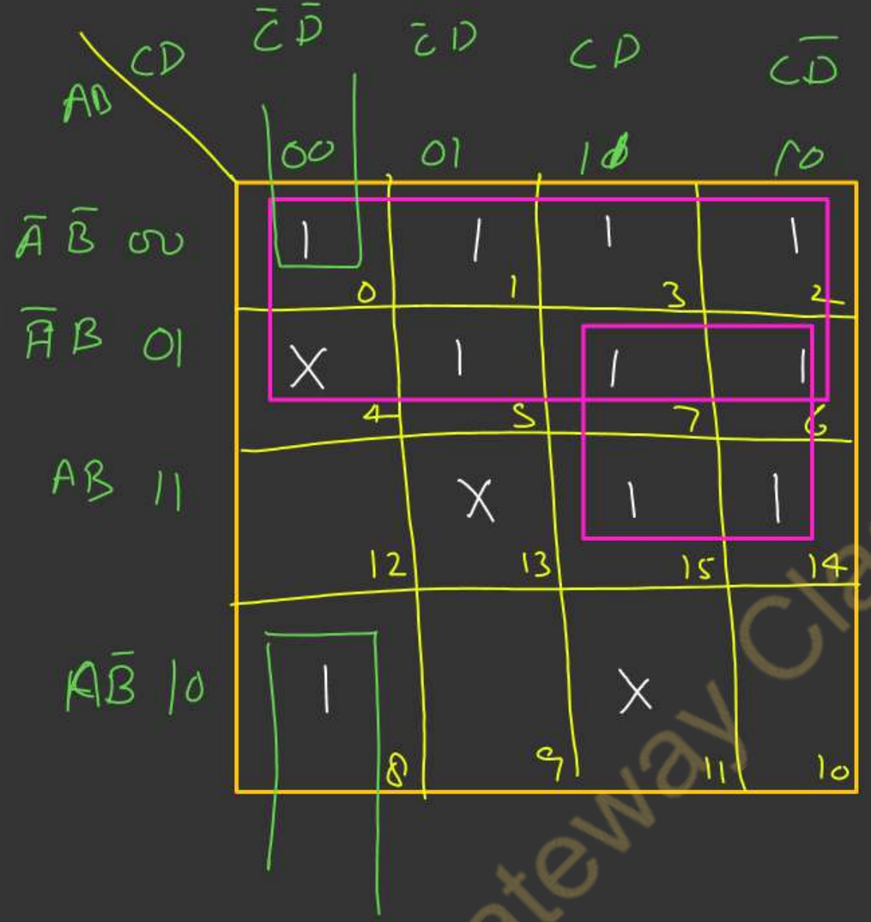
GrP	Minterm	A	B	C	D
G_0	0-1-2-3-4-5-6-7	0	—	—	—
	0-1-2-3-4-6-5-7	0	—	—	—
	0-1-4-5-2-3-6-7	0	—	—	—
	0-1-4-5-2-6-3-7	0	—	—	—
	0-2-1-3-4-5-6-7	0	—	—	—
	0-2-1-3-4-6-5-7	0	—	—	—
	0-2-4-6-1-3-5-7	0	—	—	—
	0-2-4-6-1-5-3-7	0	—	—	—
	0-4-1-5-2-3-6-7	0	—	—	—
	0-4-1-5-2-6-3-7	0	—	—	—
	0-4-2-6-1-3-5-7	0	—	—	—
	0-4-2-6-1-5-3-7	0	—	—	—

$\bar{A}$

PI	Minterm involved	Minterm used									
		0	1 ✓	2 ✓	3 ✓	5 ✓	6 ✓	7 ✓	8 ✓	14 ✗	15 ✗
$\bar{B}\bar{C}\bar{D}$	0, 8	X							X		
BC	6, 7, 14, 15						X	X		X	X
CD	3, 7, 11, 15				X			X			X
BD	5, 7, 13, 14					X		X		X	
$\bar{A}$	0, 1, 2, 3, 4, 5, 6, 7	X	X	X	X	X	X	X			

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

GW GATEWAY CLASSES
Gateway Classes



$$\text{Octet} = \bar{A} + BC + \bar{B}\bar{C}\bar{D}$$

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

Gateway Classes

Q.5 Minimize y = Π M (2,7,8,9,10,12) IN SOP form using tabular method

or $Y = \Sigma m(0,1,3,4,5,6,11,13,14,15)$ 0-15

0	0	0	0	0
1	0	0	0	1
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
11	1	0	1	1
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

GRP	Minterm	A	B	C	D
G_0	m_0	0	0	0	0 ✓
G_1	m_1	0	0	0	1 ✓
	m_4	0	1	0	0 ✓
G_2	m_3	0	0	1	1 ✓
	m_5	0	1	0	1 ✓
	m_6	0	1	1	0 ✓
G_3	m_{11}	1	0	1	1 ✓
	m_{13}	1	1	0	1 ✓
	m_{14}	1	1	1	0 ✓
G_4	m_{15}	1	1	1	1 ✓

GrP	Minterm	A	B	C	D	
G_0	$m_0 - m_1$	0	0	0	—	✓
	$m_0 - m_4$	0	—	0	0	✓
G_1	$m_1 - m_3$	0	0	—	1	✗ → $\bar{A}\bar{B}D$
	$m_1 - m_5$	0	—	0	1	✓
	$m_4 - m_5$	0	1	0	—	✓
	$m_4 - m_6$	0	1	—	0	✗ → $\bar{A}B\bar{D}$
G_2	$m_3 - m_{11}$	—	0	1	1	✗ → $\bar{B}CD$
	$m_5 - m_{13}$	—	1	0	1	✗ → $B\bar{C}D$
	$m_6 - m_{14}$	—	1	1	0	✗ → $BC\bar{D}$
G_3	$m_{11} - m_{15}$	1	—	1	1	✗ → ACD
	$m_{13} - m_{15}$	1	1	—	1	✗ → ABD
	$m_{14} - m_{15}$	1	1	1	—	✗ → ABC

GrP	Minterm	A	B	C	D	
G_0	0-1-4-5	0	—	0	—	$\bar{A}\bar{C}$
	0-4-1-5	0	—	0	—	$\bar{A}\bar{C}$

PI

P.I

Gateway Classes

$F = \bar{A}\bar{B}D + \bar{A}B\bar{D} + \bar{B}CD + BC\bar{D} + ACD + ABD + ABC + \bar{A}\bar{C}$

PI	Minterm involved	Minterm used									
		0 ✓	1 ✓	3	4 ✓	5 ✓	6	11	13	14	15
$\bar{A}\bar{B}D$	1, 3		X	X							
$\bar{A}B\bar{D}$	4, 6				X		X				
$\bar{B}CD$	3, 11			X				X			
$BC\bar{D}$ ✓	6, 14						X			X	
$B\bar{C}D$	5, 13					X			X		
ACD	11, 15							X			X
ABD	13, 15								X		X
ABC	14, 15									X	X
$\bar{A}\bar{C}$	0, 1, 4, 5	X	X		X	X					

$\bar{A}\bar{C} = 0, 1, 4, 5$

3, 6, 11, 13, 14, 15

$\bar{B}CD = 3, 11$

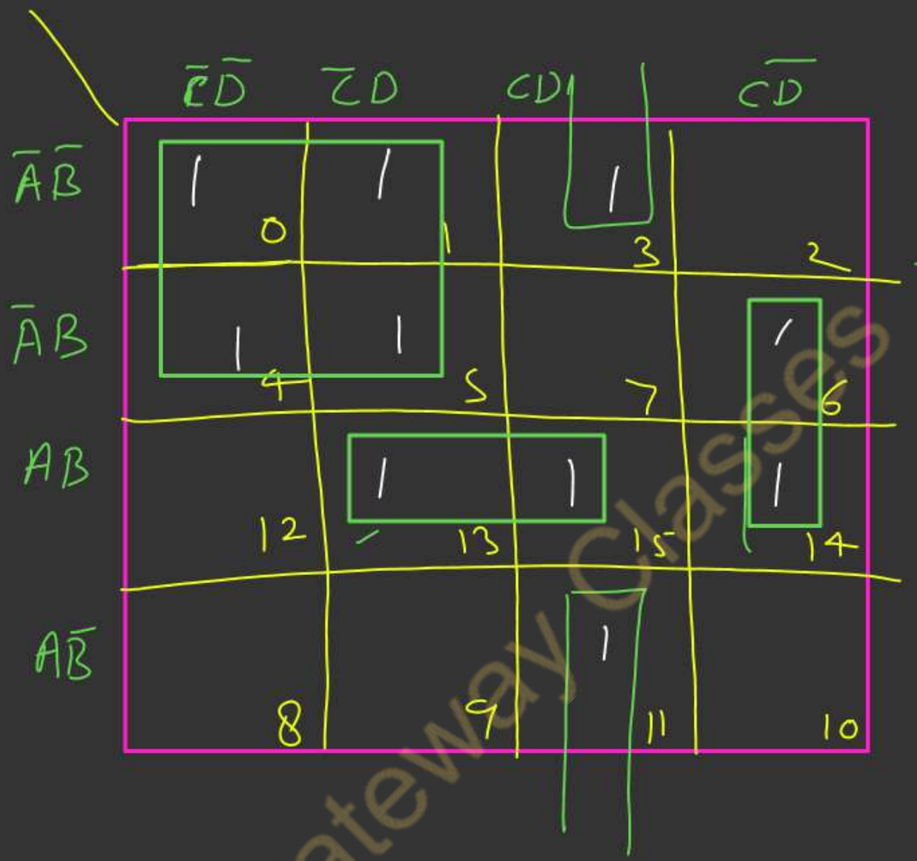
$BC\bar{D} = 6, 14$

$ABD = 13, 15$

$F = \bar{A}\bar{C} + \bar{B}CD + BC\bar{D} + ABD$

Ans

Gateway Classes



$$F = \bar{A}\bar{C} + ABD + \bar{B}CD + B\bar{C}\bar{D}$$

Gateway Classes

Unit-1 Lec-6

DPP

Q.1 Simplify the following function using Tabular method

$Y(A, B, C, D) = \Sigma m(2, 4, 8, 11, 15) + d(1, 10, 12, 13)$

Q.2 Simplify the following function using Tabular method

$Y(A, B, C, D) = \Sigma m(0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$

Gateway Classes

Gateway Classes

Empowering Learners, Transforming Futures

Thank You

Gateway Classes



Helpline No- 7819 0058 53