

TABLE OF CONTENTS

Unit 1

Chapter - 1	Digital Logic Families	(1 - 1) to (1 - 15)
1.1	Classification of Logic Families	1 - 1
1.2	Digital IC Characteristics.....	1 - 1
1.3	TTL : Standard TTL Characteristics, Operation of TTL NAND Gate	1 - 4
1.4	CMOS : Standard CMOS Characteristics, Operation of CMOS NAND Gate	1 - 9
1.5	Comparison of TTL and CMOS	1 - 14
Chapter - 2	Signed Binary Representation and Arithmetic and Codes	(2 - 1) to (2 - 30)
2.1	Introduktion to Number Systems	2 - 1
2.2	Signed Binary Number Representation - Signed and True Magnitude, 1's Complement and 2's Complement Representation.....	2 - 9
2.3	Binary Arithmetic.....	2 - 10
2.4	Codes : BCD, Excess-3, Gray Binary Code and Their Conversion	2 - 13
2.5	IEEE Standard 754 Floating Point Number Representation .	2 - 25
Chapter - 3	Logic Minimization	(3 - 1) to (3 - 19)
3.1	Review of Boolean Algebra	3 - 1
3.2	Representation of Logic Functions	3 - 1
3.3	Simplification of Logical Functions, using K-Maps up to 4 Variables	3 - 1

Unit 2

Chapter - 4

Design using SSI Chips	(4 - 1) to (4 - 16)
4.1 Code Converters.....	4 - 1
4.2 Half - Adder, Full Adder, Half Subtractor, Full Subtractor, Binary Adder (IC 7483)	4 - 7
4.3 n-bit Binary Adder	4 - 15

Chapter - 5

Introduction to MSI Chips and Design using MSI Chips

(5 - 1) to (5 - 27)

5.1 Multiplexers (IC 74153) and Implementation of Logic Function using IC 74153	5 - 1
5.2 Decoder / Demultiplexer (IC 74138, IC 74238) and Implementation of Logic Function Using IC 74138	5 - 9
5.3 Encoder (IC 74147)	5 - 18
5.4 Binary Adder	5 - 21
5.5 BCD Adder	5 - 22
5.6 BCD Subtractor using IC 7483.....	5 - 24

Unit 3

Chapter - 6

Introduction to Sequential Circuits and Flip-Flops

(6 - 1) to (6 - 19)

6.1 Introduction.....	6 - 1
6.2 Flip-Flop : SR, JK, D, T.....	6 - 4
6.3 Preset and Clear	6 - 9
6.4 Master and Slave Flip Flop.....	6 - 10
6.5 Excitation Tables.....	6 - 11
6.5 Conversion from One FF to Another	6 - 12
6.7 Study of 7474 and 7476 Flip-Flop ICs	6 - 13

1

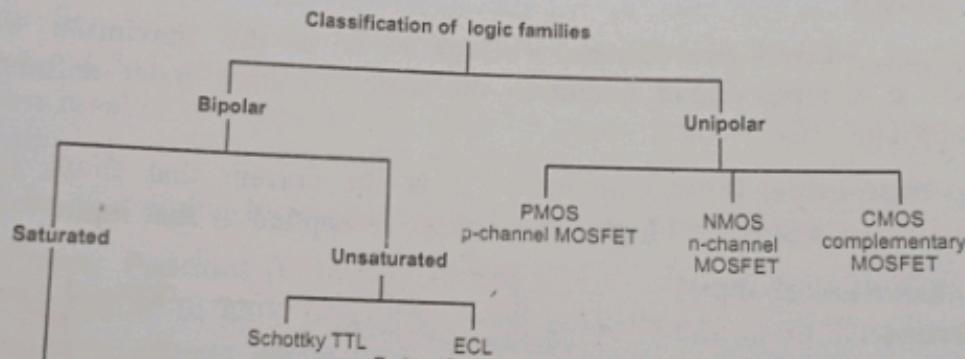
Digital Logic Families

1.1 : Classification of Logic Families

Q.1 What is logic family ? Give the classification of logic families.

[SPPU : May-10,12, Dec.-08, Marks 4]

Ans. : A digital logic family is a group of compatible devices with the same logic levels and supply voltages. According to components used in the logic family, digital logic families are classified as shown in the Fig. Q.1.1.



- RTL : Register Transistor Logic
- DTL : Diode Transistor Logic
- DCTL : Direct Coupled Transistor Logic
- I²L : Integrated Injection Logic
- HTL : High Threshold Logic
- TTL : Transistor Transistor Logic

Fig. Q.1.1 Classification of logic families

1.2 : Digital IC Characteristics

Q.2 State and explain any four characteristics of digital ICs.

[SPPU : Dec.-08,15, May-17, Marks 6]

Ans. : **Propagation Delay** : The propagation delay of a gate is basically the time interval between the application of an input pulse and the occurrence of the resulting output pulse. The propagation delay is a very

important characteristic of logic circuits because it limits the speed at which they can operate. The shorter the propagation delay, the higher the speed of the circuit and vice-versa.

Power Dissipation : The amount of power that an IC dissipates is determined by the average supply current, I_{CC} , that it draws from the V_{CC} supply. It is the product of I_{CC} and V_{CC} .

Current and Voltage Parameter

$V_{IH(min)}$ - **High-Level Input Voltage :** It is the minimum voltage level required for a logical 1 at an input. Any voltage below this level will not be accepted as a HIGH by the logic circuit.

$V_{IL(max)}$ - **Low-Level Input Voltage :** It is the maximum voltage level required for a logic 0 at an input. Any voltage above this level will not be accepted as a LOW by the logic circuit.

$V_{OH(min)}$ - **High-Level Output Voltage :** It is the minimum voltage level at a logic circuit output in the logical 1 state under defined load conditions.

$V_{OL(max)}$ - **Low-Level Output Voltage :** It is the maximum voltage level at a logic circuit output in the logical 0 state under defined load conditions.

I_{IH} - **High-Level Input Current :** It is the current that flows into an input when a specified high-level voltage is applied to that input.

I_{IL} - **Low-Level Input Current :** It is the current that flows into an input when a specified low-level voltage is applied to that input.

I_{OH} - **High-Level Output Current :**

It is the current that flows from an output in the logical 1 state under specified load conditions.

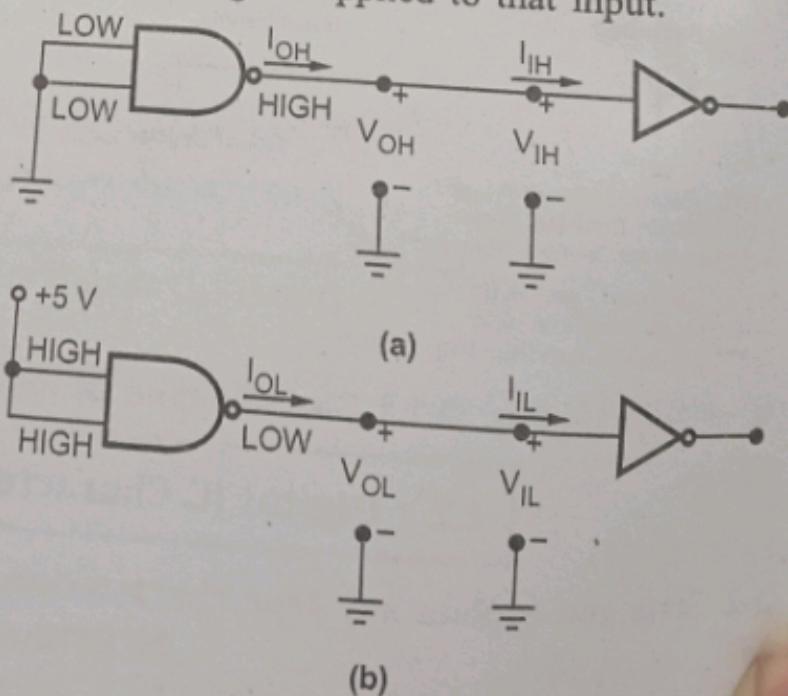


Fig. Q.2.1 Currents and voltages in the two logic states

I_{OL}- Low-Level Output Current : It is the current that flows from an output in the logical 0 state under specified load conditions.

Noise Margin : The noise immunity of a logic circuit refers to the circuit's ability to tolerate the noise without causing spurious changes in the output voltage. To avoid this problem due to noise, voltage level $V_{IH(min)}$ is kept at a few fraction of volts below $V_{OH(min)}$ and voltage level $V_{IL(max)}$ is kept above $V_{OL(max)}$, at the design time.

V_{NH} is the difference between the lowest possible HIGH output, $V_{OH(min)}$ and the minimum voltage, $V_{IH(min)}$ required for a HIGH input. This voltage difference, V_{NH} is called high-state noise margin. Similarly, we have low-state noise margin. It is the voltage difference between the largest possible low output, $V_{OL(max)}$ and the maximum voltage, $V_{IL(max)}$ required for a LOW input.

In short we can write as,

$$V_{NH} = V_{OH(min)} - V_{IH(min)} \quad \text{and} \quad V_{NL} = V_{IL(max)} - V_{OL(max)}$$

Fan-in and Fan-out : The maximum number of inputs of several gates that can be driven by the output of a logic gate is decided by the parameter called **fan-out**. In general, the fan-out is defined as the maximum number of inputs of the same IC family that the gate can drive maintaining its output levels within the specified limits.

The **fan-in** of a digital logic gate refers to the number of inputs.

Speed Power Product (Figure of Merit) : • In general, for any digital IC, it is desirable to have shorter propagation delays (higher speed) and lower values of power dissipation. There is usually a trade-off between switching speed and power dissipation in the design of a logic circuit i.e. speed is gained at the expense of increased power dissipation. Therefore, a common means for measuring and comparing the overall performance of an IC family is the **Speed-Power Product (SPP)**. It is also called **Figure of Merit**.

Operating Temperature Range : • It is the temperature range specified by the logic family within which devices are guaranteed to work reliably.

Power Supply Requirements : • Power supply requirements differ from logic family to family. For example, it is 5V for TTL family and 3-15 volts for CMOS family. Further more, power supply tolerance also depends on logic family. For example, for 74 series TTL family it is ± 0.25 V and for 54 series TTL family it is ± 0.5 V.

1.3 : TTL : Standard TTL Characteristics, Operation of TTL NAND Gate

Q.3 With neat circuit diagram explain the operation of two-input TTL NAND gates.

[SPPU : May-06, 10, 12, 13, 16, Dec.-07, Marks 8]

Ans. : The Fig. Q.3.1 (a) shows the circuit diagram of 2-input NAND gate. Its input structure consists of multiple-emitter transistor and output structure consists of totem-pole output. Here, Q_1 is an NPN transistor having two emitters, one for each input to the gate. Although this circuit looks complex, we can simplify its analysis by using the diode equivalent of the multiple-emitter transistor Q_1 , as shown in Fig. Q.3.1 (b). Diodes D_2 and D_3 represent the two E-B junctions of Q_1 and D_4 is the collector-base (C-B) junction.

The input voltages A and B are either LOW (ideally grounded) or HIGH (ideally +5 volts). If either A or B or both are low, the corresponding diode conducts and the base of Q_1 is pulled down to approximately 0.7 V. This reduces the base voltage of Q_2 to almost zero. Therefore, Q_2 cuts off. With Q_2 open, Q_4 goes into cut-off and the Q_3 base is pulled HIGH. Since Q_3 acts as an emitter follower, the Y output is pulled up to a HIGH voltage. On the other hand, when A and B both are HIGH, the emitter diode of

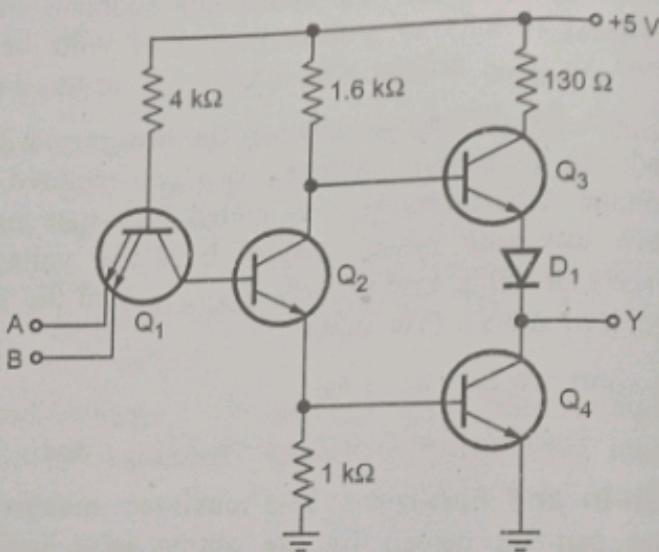


Fig. Q.3.1 (a) Two input TTL NAND gate

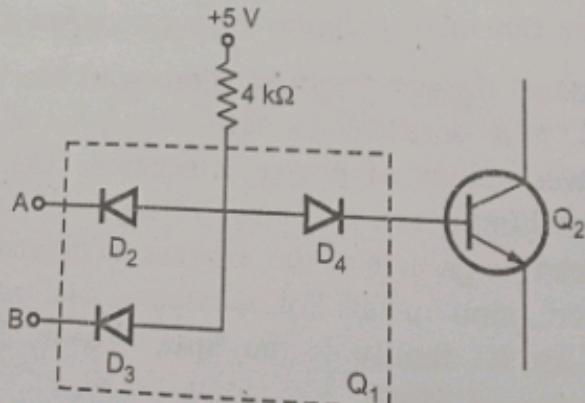
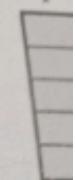


Fig. Q.3.1 (b) Diode equivalent for Q_1

Q_1 are reversed bias
 D_4 to go into forward
turn, Q_4 goes into
summarizes all input



Table

Without diode D_4 ,
is low. To prevent
base-emitter diode
when the output

Q.4 Draw three
its operation.

Ans. : The Fig.
as that of two
has three emit
input NAND g
logic 0; other
2-input NAND
NAND gate.

A	B
0	0
0	0
0	1
0	1
1	0
1	0
1	1
1	1

Table Q.4.
3-input

Q_1 are reversed biased making them off. This causes the collector diode D_4 to go into forward conduction. This forces Q_2 base to go HIGH. In turn, Q_4 goes into saturation, producing a low output. Table Q.3.1 summarizes all input and output conditions.

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Table Q.3.1 Truth table for 2-input NAND gate

Without diode D_1 in the circuit, Q_3 will conduct slightly when the output is low. To prevent this, the diode is inserted; its voltage drop keeps the base-emitter diode of Q_3 reverse-biased. In this way, only Q_4 conducts when the output is low.

Q.4 Draw three input standard TTL NAND gate circuit and explain its operation.

[SPPU : Dec.-08,10, Marks 8]

Ans. : The Fig. Q.4.1 shows the three input TTL NAND gate. It is same as that of two input TTL NAND gate except that its Q_1 (NPN) transistor has three emitters instead of two. Rest of the circuit is same. For three input NAND gate if all the inputs are logic 1 then and then only output is logic 0; otherwise output is logic 1. The operation is similar to the 2-input NAND gate. The Table Q.4.1 shows the truth table for 3-input NAND gate.

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

Table Q.4.1 Truth table of 3-input NAND gate

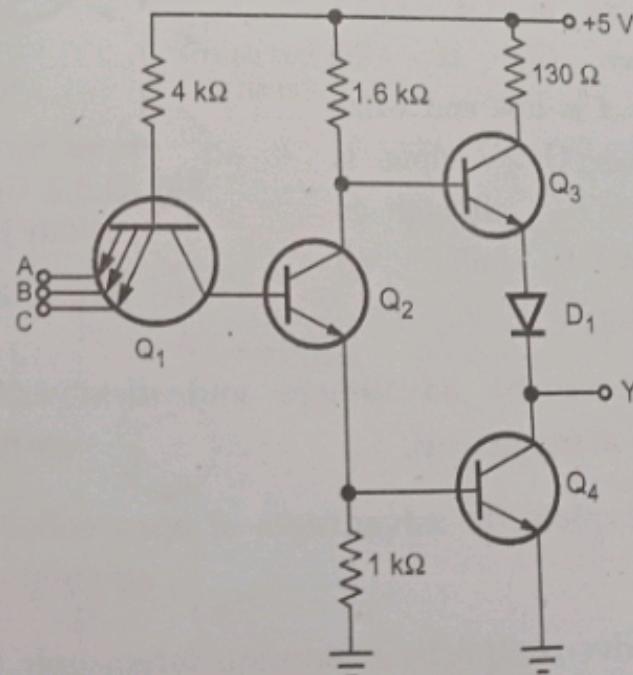


Fig. Q.4.1 Three input TTL NAND gate

Q.5 What do you mean by open collector output ? Explain with suitable circuit diagram ?

[SPPU : May-15, Marks 4]

Ans. : Some TTL devices provide another type of output called **open collector output**. The outputs of two different gates with open collector output can be tied together. This is known as **wired logic**.

Fig. Q.5.1 shows a 2-input NAND gate with an open-collector output eliminates the pull-up transistor Q_3 , D_1 and R_4 . The output is taken from the open collector terminal of transistor Q_4 .

Because the collector of Q_4 is open, a gate like this will not work properly until you connect an external pull-up resistor, as shown in Fig. Q.5.2.

When Q_4 is ON, output is low and when Q_4 is OFF output is tied to V_{CC} through an external pull up resistor.

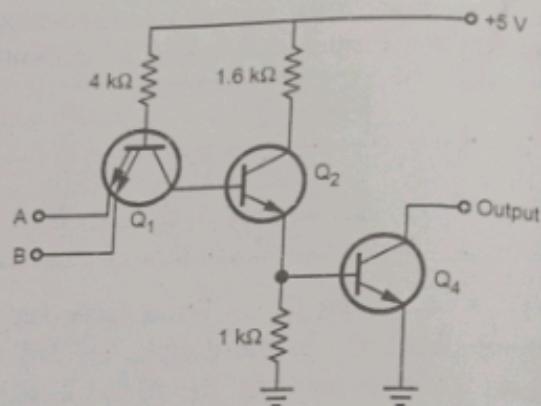


Fig. Q.5.1 Open collector 2-input TTL NAND gate

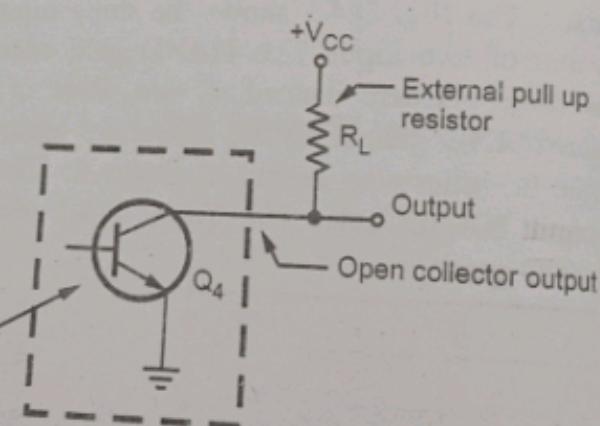


Fig. Q.5.2 Open collector output with pull-up resistor

Q.6 Give the advantages and disadvantages of totem-pole output stage arrangement.

[SPPU : May-05, Dec.-07, Marks 4]

OR Explain the advantages of open collector output.

[SPPU : Dec.-06,15, May-08,15 Marks 4]

OR Give comparison between totem-pole and open collector outputs.

Ans. :

Sr.No.	
1.	Output s transistor pull-down
2.	External required
3.	Output tied tog
4.	Operati

Table Q.6.1

Q.7 Draw and explain the circuit of an inverter with open collector output.

Ans. : The truth table for an inverter with open collector output is as follows:

Input (A)	Output (Y)
0	Open collector output
1	High (V _{CC})

The circuit configuration is shown in Fig. Q.5.2. The output is connected to an external pull-up resistor R_L which is connected to the power supply $+V_{CC}$. The output is labeled "Open collector output".

Enable(E) —

Ans. :

Sr.No.	Totem-pole	Open collector
1.	Output stage consists of pull-up transistor (Q_3), diode resistor and pull-down transistor (Q_4).	Output stage consists of only pull-down transistor.
2.	External pull-up resistor is not required.	External pull-up resistor is required for proper operation of gate.
3.	Output of two gates cannot be tied together.	Output of two gates can be tied together using wired AND technique.
4.	Operating speed is high.	Operating speed is low.

Table Q.6.1 Comparison of totem-pole and open collector output

Q.7 Draw and explain the circuit diagram of tri-state TTL NAND gate.

[SPPU : May-05, Dec.-05, 07, Marks 6]

Ans. : The tristate configuration is a third type of TTL output configuration. It utilizes the high-speed operation of the totem-pole arrangement while permitting outputs to be wired-ANDED (connected together). It is called tristate TTL because it allows three possible output stages : HIGH, LOW and high-impedance.

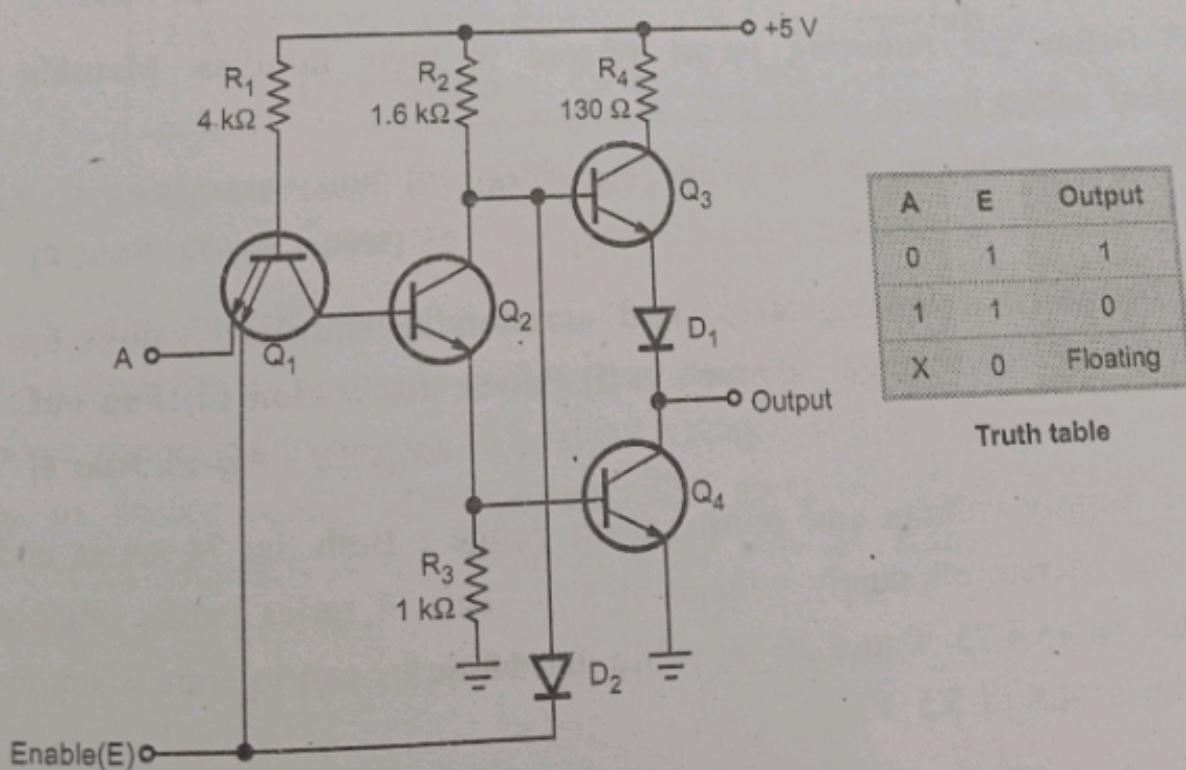


Fig. Q.7.1 Tristate TTL inverter

Fig. Q.7.1 shows the simplified circuit for tristate inverter. It has two inputs A and E.

A is the normal logic input whereas E is an ENABLE input. When ENABLE input is HIGH, the circuit works as a normal inverter. Because when E is HIGH, the state of the transistor Q_1 (either ON or OFF) depends on the logic input A, and the additional component diode is open circuited as its cathode is at logic HIGH. When ENABLE input is LOW, regardless of the state of logic input A, the base-emitter junction of Q_1 is forward biased and as a result it turns ON. This shunts the current through R_1 away from Q_2 making it OFF. As Q_2 is OFF, there is no sufficient drive for Q_4 to conduct and hence Q_4 turns OFF. The LOW at ENABLE input also forward-biases diode D_2 , which shunt the current away from the base of Q_3 , making it OFF. In this way, when ENABLE input is LOW, both transistors are OFF and output is at high impedance state.

Q.8 Explain the following characteristics of TTL logic families :

- i) Power dissipation ii) Noise margin iii) Propagation delay
- iv) Fan out.

OR Explain standard TTL characteristics in detail.

[SPPU : Dec.-08, 10, 12, May-10, 13, Marks 8]

OR Define the following terms related to logic families. Mention typical values for standard TTL family :

- i) Power dissipation ii) Fan-in iii) V_{IL} , V_{OH} iv) Noise margin.

[SPPU : May-12, Marks 8]

OR Define the following terms and mention its standard values for TTL family : i) Voltage parameters ii) Power dissipation iii) Fan out

[SPPU : May-16, Marks 6]

Ans. : Supply voltage and temperature range : Both the 74 series and 54 series operate on supply voltage of 5 V. The 74 series works reliably over the range 4.75 V to 5.25 V, while the 54 series can tolerate a supply variation of 4.5 to 5.5 V.

Voltage level
output logic
maximum
of power

For TTL
both are

Power
average

Fan-out
inputs

Q.9

Ans.
two
sourc
has
conne
toge

Voltage levels and noise margin : Table Q.8.1 shows the input and output logic voltage levels for the standard 74 series. The minimum and maximum values shown in the Table Q.8.1 are for worst case conditions of power supply, temperature and loading conditions.

Voltages	Minimum	Typical	Maximum
V_{OL}	—	0.2	0.4
V_{OH}	2.4	3.4	—
V_{IL}	—	—	0.8
V_{IH}	2.0	—	—

Table Q.8.1 Voltage levels

For TTL, Low state noise margin, V_{NL} and high state noise margin, V_{NH} both are equal and 0.4 V.

Power dissipation and propagation delay : A standard TTL gate has an average power dissipation of about 10 mW.

Fan-out : A standard TTL output can typically drive 10 standard TTL inputs. Therefore, standard TTL has fan-out 10.

1.4 : CMOS : Standard CMOS Characteristics, Operation of CMOS NAND Gate

**Q.9 Draw the structure of CMOS inverter gate. Explain its working.
[SPPU : Dec.-07, Marks 3; May-12, Marks 4]**

Ans. : Fig. Q.9.1 shows the basic CMOS inverter circuit. It consists of two MOSFETs in series in such a way that the P-channel device has its source connected to $+V_{DD}$ (a positive voltage) and the N-channel device has its source connected to ground. The gates of the two devices are connected together as the common input and the drains are connected together as the common output.

- When input is HIGH, the gate of Q_1 (P-channel) is at 0 V relative to the source of Q_1 i.e. $V_{gs1} = 0$ V. Thus, Q_1 is OFF. On the other hand, the gate of Q_2 (N-channel) is at $+V_{DD}$ relative to its source i.e. $V_{gs2} = +V_{DD}$. Thus, Q_2 is ON. This will produce $V_{OUT} \approx 0$ V.
- When input is LOW, the gate of Q_1 (P-channel) is at a negative potential relative to its source while Q_2 has $V_{gs} = 0$ V. Thus, Q_1 is ON and Q_2 is OFF. This produces output voltage approximately $+V_{DD}$.

Q.10 Explain with neat diagram two input CMOS NAND gate.

[SPPU : Dec.-12,14, May-14, Marks 8]

Ans. :

Fig. Q.10.1 shows CMOS 2-input NAND gate. It consists of two P-channel MOSFETs, Q_1 and Q_2 , connected in parallel and two N-channel MOSFETs, Q_3 and Q_4 connected in series.

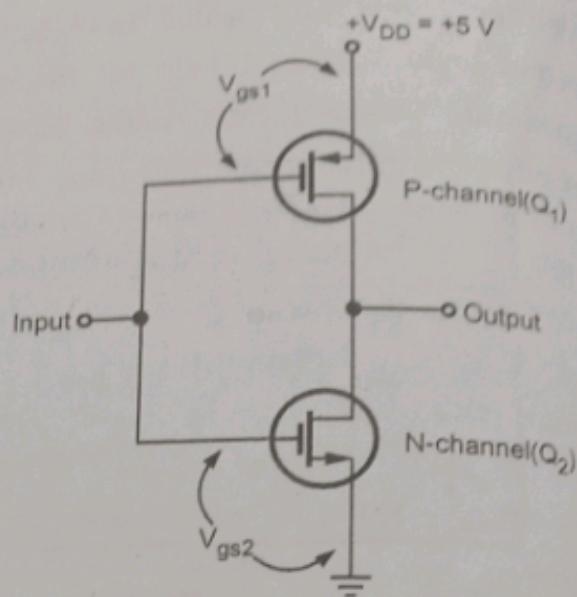
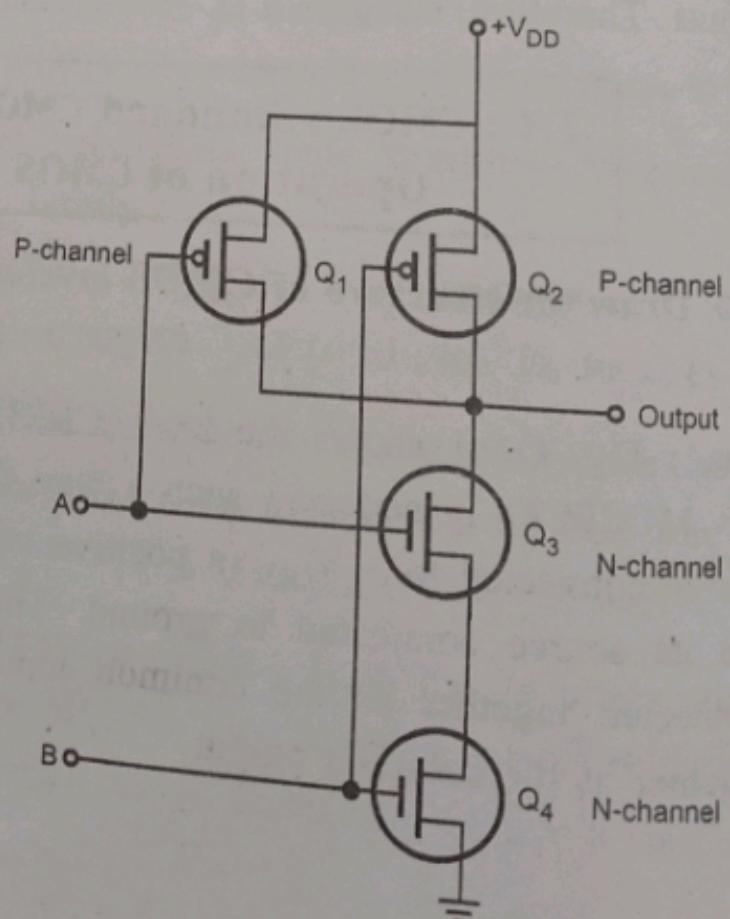


Fig. Q.9.1 CMOS inverter circuit



A	B	Q_1	Q_2	Q_3	Q_4	Output
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	1
1	0	OFF	ON	ON	OFF	1
1	1	OFF	OFF	ON	ON	0

Table Q.10.1 Truth table of NAND gate

Here, the gates of both P-channel MOSFETs are negative with respect to their sources, since the sources are connected to $+V_{DD}$. Thus, Q_1 and Q_2 are both ON. Since the gate - to - source voltages of Q_3 and Q_4 (N-channel MOSFETs) are both 0 V, those MOSFETs are OFF. The output is therefore connected to $+V_{DD}$ (HIGH) through Q_1 and Q_2 and is disconnected from ground. When $A = 0$ and $B = +V_{DD}$, Q_1 is on because $V_{GS1} = -V_{DD}$ and Q_4 is ON because $V_{GS4} = +V_{DD}$. MOSFETs Q_2 and Q_3 are off because their gate-to-source voltages are 0 V. Since Q_1 is ON and Q_3 is OFF, the output is connected to $+V_{DD}$ and it is disconnected from ground. When $A = +V_{DD}$ and $B = 0$ V, the situation is similar (not shown); the output is connected to $+V_{DD}$ through Q_2 and it is disconnected from ground because Q_4 is OFF. Finally, when both inputs are high ($A = B = +V_{DD}$), MOSFETs Q_1 and Q_2 are both OFF and Q_3 and Q_4 are both ON. Thus, the output is connected to the ground through Q_3 and Q_4 and it is disconnected from $+V_{DD}$. The Table Q.10.1 summarizes the operation of 2-input CMOS NAND gate.

Q.11 Explain with a neat diagram interfacing of TTL gate driving CMOS gates and vice-versa.

[SPPU : May-05,07,13, Dec.-05, Marks 6; Dec.-08,11, Marks 8]

OR How will you connect the output of CMOS logic circuit as an input to TTL logic circuit ? Explain your reason with suitable diagram.

[SPPU : Dec.-16, Marks 6]

Ans. : TTL Driving CMOS : The input current values for CMOS are extremely low compared with the output current capabilities of any TTL series. Thus, TTL has no problem meeting the CMOS input current requirements.

But when we compare the TTL output voltages with the CMOS input voltage requirements we find that :

$V_{OH(min)}$ for TTL $\ll V_{IH(min)}$ for CMOS for these situations TTL output must be raised to an acceptable level for CMOS. This can be done by connecting pull-up resistor at the output of TTL, as shown in the Fig. Q.11.1. The pull-up resistor causes the TTL output to rise to approximately 5 V in the HIGH state, thereby providing an adequate CMOS input voltage level.

TTL Driving HIGH Voltage CMOS : When output CMOS circuit is operating with V_{DD} greater than 5 V, the situation becomes more difficult. The outputs of many TTL devices cannot be operated at more than 5 V. In such cases some alternative arrangements are made. Two of them are discussed below :

- When the TTL output cannot be pulled up to V_{DD} , one can use open collector buffer as an interface between totem-pole TTL output and CMOS operating at $V_{DD} > 5$ V, as shown in the Fig. Q.11.2.

- The second alternative is to use level translator circuit, such as the 40104. This is a CMOS chip that is designed to take a low-voltage input (e.g. from TTL) and

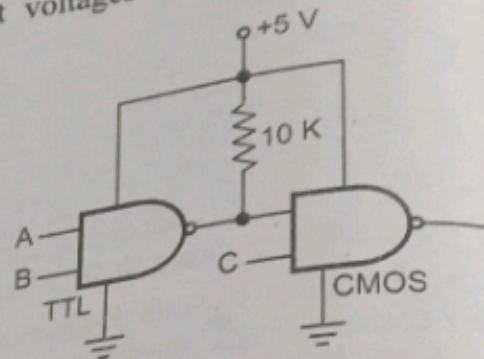


Fig. Q.11.1 TTL driving CMOS using external pull-up resistor

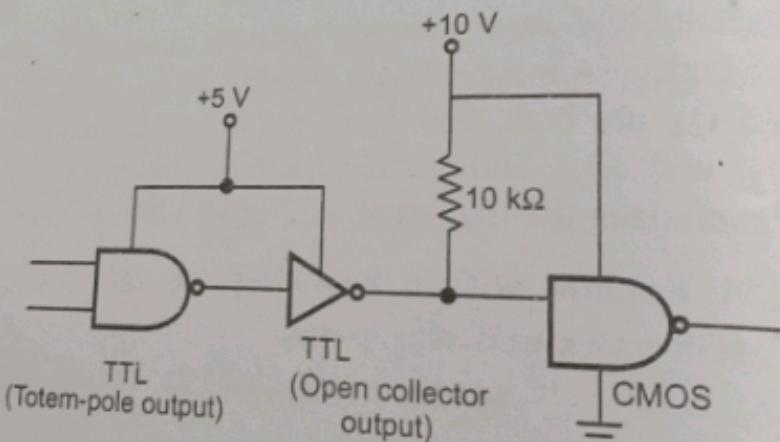


Fig. Q.11.2 Open collector buffer used as interface circuit

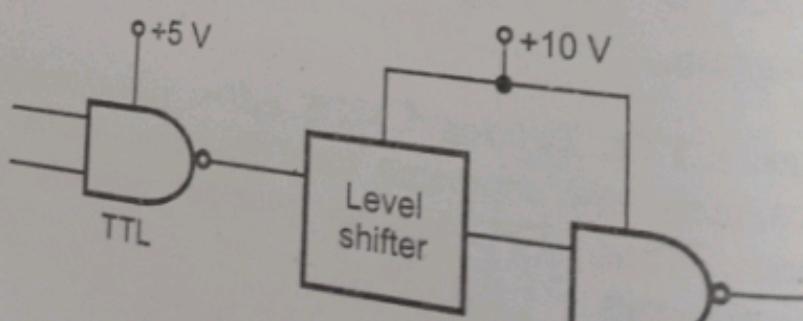


Fig. Q.11.3 Level shifter used as interface

translate it to high voltage output for CMOS. Fig. Q.11.3 shows the circuit arrangement.

CMOS Driving TTL : CMOS outputs can easily supply enough voltage (V_{OH}) to satisfy the TTL input requirement in the HIGH state (V_{IH}). CMOS outputs can supply more than enough current (I_{OH}) to meet the TTL input current requirements (I_{IH}). Thus no special consideration is required for CMOS driving TTL in the HIGH state.

CMOS output voltage (V_{OL}) satisfies TTL input requirement in the LOW state (V_{IL}). However, the current requirements in the LOW state are not satisfied. The TTL input has a relatively high input current in the LOW state (1.6 mA) and CMOS output current at LOW state (I_{OL}) is not sufficient to drive even one input of the TTL. In such situations some type of interface circuit is needed between the CMOS and TTL devices.

In Fig. Q.11.4 the CMOS 4050 B, non-inverting buffer is used as an interfacing circuit. It has an output current rating of $I_{OL(max)} = 3$ mA which satisfies the TTL input current requirement.

HIGH Voltage CMOS Driving TTL : Some IC manufacturers have provided several 74LS

TTL devices that can withstand input voltages as high as 15 V. These devices can be driven directly from CMOS outputs operating at $V_{DD} = 15$ V. However, most TTL inputs cannot handle more than 7 V and so interface is necessary if they are to be driven from high-voltage CMOS.

In such situations voltage level

translators are used. Fig. Q.11.5 Level translation using CMOS buffer

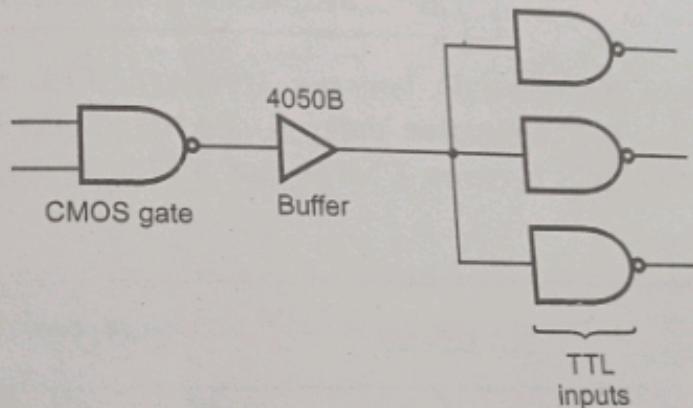
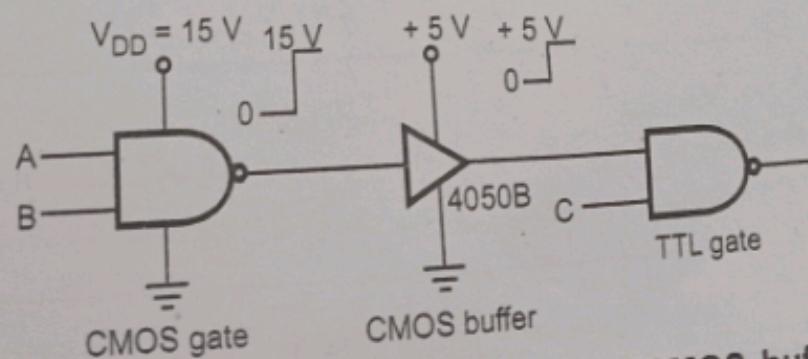


Fig. Q.11.4 CMOS driving TTL in LOW state using buffer



They convert the high-voltage input to a 5 V output that can be connected to TTL.

Fig. Q.11.5 shows how the 4050B can be used to perform this level translation between 15 V and 5 V.

Q.12 Why is a buffer required between some CMOS outputs and TTL inputs ?

[SPPU : May-13, Marks 2]

Ans. : The current requirement of CMOS output at LOW state are not satisfied by TTL input. The TTL input has a relatively high input current in the LOW state (1.6 mA) and CMOS output current at LOW state (I_{OL}) is not sufficient to drive even one input of the TTL. Hence buffer is used for interfacing.

1.5 : Comparison of TTL and CMOS

Q.13 Differentiate between standard TTL and CMOS logic circuit w.r.t. i) Propagation delay ii) FANOUT iii) Figure of merit. List the differences between CMOS and TTL.

[SPPU : May-15,16, Marks 6]

Ans. :

Sr.No.	Parameter	CMOS	TTL
1.	Device used	n-channel and p-channel MOSFET	Bipolar junction transistor
2.	$V_{IH(\min)}$	3.5 V	2 V
3.	$V_{IL(\max)}$	1.5 V	0.8 V
4.	$V_{OH(\min)}$	4.95 V	
5.	$V_{OL(\max)}$	0.005 V	2.7 V
6.	High level noise margin	$V_{NH} = 1.45$ V	0.4 V
7.	Low level noise margin	$V_{NL} = 1.45$ V	0.4 V
			0.4 V

it can be connected

perform this level

S outputs and
ay-13, Marks 2]state are not
input current
W state (I_{OL})
uffer is usedic circuit
List the
Marks 6]

8.	Noise immunity	Better than TTL	Less than CMOS
9.	Propagation delay	70 ns	10 ns
10.	Switching speed	Less than TTL	Faster than CMOS
11.	Power dissipation per gate	0.1 mW	10 mW
12.	Speed power product	0.7 pJ	100 pJ
13.	Fan-out	50	10
14.	Power supply voltage	3 - 15 V	Fixed 5 V
15.	Power dissipation	Increase with frequency	Increase with frequency
16.	Application	Portable instrument where battery supply is used.	Laboratory instruments.

Table Q.13.1 Comparison between TTL and CMOS families

END... ↗

Unit 1**2****Signed Binary Representation and Arithmetic and Codes****2.1 : Introduction to Number Systems****Concepts : Number Systems**

- Number system is a basis for counting various items.
- The decimal number system has 10 digits : 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.
- Modern computers communicate and operate with binary numbers which use only the digits 0 and 1.
- When decimal quantities are represented in the binary form, they take more digits.
- For large decimal numbers people have to deal with very large binary strings and therefore, they do not like working with binary numbers. This fact gave rise to three new number systems : Octal, Hexadecimal and Binary Coded Decimal (BCD).
- Each number system has r set of characters. For example, in decimal

Radix (Base) r	Characters in set
2 (Binary)	0, 1
3	0, 1, 2
4	0, 1, 2, 3
5	0, 1, 2, 3, 4
6	0, 1, 2, 3, 4, 5
7	0, 1, 2, 3, 4, 5, 6
8 (Octal)	0, 1, 2, 3, 4, 5, 6, 7
9	0, 1, 2, 3, 4, 5, 6, 7, 8
10 (Decimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
:	
:	
16 (Hexadecimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Table 2.1 Radix and character set

- In general we characters in Table 2.1.
- Table 2.2 shows hexadecimal

De

Tab

Q.1
of c

Ans
can
illu

number system r equals to 10 has 10 characters from 0 to 9, in binary number system r equals to 2 has 2 characters 0 and 1 and so on.

- In general we can say that, a number represented in radix r , has r characters in its set and r can be any value. This is illustrated in Table 2.1.
- Table 2.2 shows the relationship between decimal, binary, octal and hexadecimal.

Decimal	Binary	Octal	Hexadecimal
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

Table 2.2 Relation between decimal, binary, octal and hexadecimal numbers

Q.1 Convert $(1101.101)_2$ to decimal number and explain the process of conversion.

Ans. : • By adding each digit of a binary number in a power of 2 we can find the decimal equivalent of the given binary number. This is illustrated in Fig. Q.1.1.

$$\begin{array}{c}
 \begin{array}{ccccccccc}
 & 2^3 & 2^2 & 2^1 & 2^0 & . & 2^{-1} & 2^{-2} & 2^{-3} \\
 \text{N} = & \boxed{1} & 1 & 0 & 1 & . & 1 & 0 & 1 \\
 & 1 \times 2^3 & 1 \times 2^2 & 0 \times 2^1 & 1 \times 2^0 & . & 1 \times 2^{-1} & 0 \times 2^{-2} & 1 \times 2^{-3} \\
 \text{N} = & 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 & + & 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} & = (13.625)_{10}
 \end{array}
 \end{array}$$

Fig. Q.1.1

Q.2 Convert $(5632.471)_8$ to decimal number and explain the process of conversion.

Ans. : By adding each digit of an octal number in a power of 8 we can find the decimal equivalent of the given octal number. This is illustrated in Fig. Q.2.1.

$$\begin{array}{c}
 \begin{array}{ccccccccc}
 & 8^3 & 8^2 & 8^1 & 8^0 & . & 8^{-1} & 8^{-2} & 8^{-3} \\
 \text{N} = & \boxed{5} & 6 & 3 & 2 & . & 4 & 7 & 1 \\
 & 5 \times 8^3 & 6 \times 8^2 & 3 \times 8^1 & 2 \times 8^0 & . & 4 \times 8^{-1} & 7 \times 8^{-2} & 1 \times 8^{-3} \\
 \text{N} = & 5 \times 8^3 + 6 \times 8^2 + 3 \times 8^1 + 2 \times 8^0 & + & 4 \times 8^{-1} + 7 \times 8^{-2} + 1 \times 8^{-3} & = (2970.611328)_{10}
 \end{array}
 \end{array}$$

Fig. Q.2.1

Q.3 Convert $(3FD.84)_{16}$ into decimal number and explain the process of conversion.

Ans. : By adding each digit of a hexadecimal number in a power of 16 we can find decimal equivalent of the given hexadecimal number.

$$\begin{array}{c}
 \begin{array}{ccccccccc}
 & 16^2 & 16^1 & 16^0 & . & 16^{-1} & 16^{-2} \\
 \text{N} = & \boxed{3} & F & D & . & 8 & 4 \\
 & 3 \times 16^2 & F \times 16^1 & D \times 16^0 & . & 8 \times 16^{-1} & 4 \times 16^{-2} \\
 \text{N} = & 3 \times 16^2 + F \times 16^1 + D \times 16^0 & + & 8 \times 16^{-1} + 4 \times 16^{-2} \\
 = & 3 \times 16^2 + 15 \times 16^1 + 13 \times 16^0 & + & 8 \times 16^{-1} + 4 \times 16^{-2} & = (1021.515625)_{10}
 \end{array}
 \end{array}$$

Q.4 What is the radix called in case of decimal, binary, octal and hexadecimal number systems ?

Ans. : The radix or base for above number system are shown in Table 1.1.

Q.5 Convert 10101101.0111 to octal equivalent and explain the conversion process.

Ans. :

Step 1 : • Make group of 3-bits starting from LSB for integer part and MSB for fractional part, by adding 0s at the end, if required.

Step 1.

0	1	0	1	0	1	1	0	1	.	0	1	1	1	0	0
2	5	5							.	3	4				

Step 2.

Adding 0 to make a group of 3-bits

Adding 0s to make a group of 3-bits

Binary (Base 2)
Octal (Base 8)

Step 2 : • Write equivalent octal number for each group of 3-bits.
 $\therefore (10101101.0111)_2 = (255.34)_8$

Q.6 Convert $(125.62)_8$ to binary and explain the conversion process.

Ans. :

Step 1 : Write equivalent 3-bit binary number for each octal digit.

Step 2 : Remove any leading or trailing zeros.

1	2	5	.	6	2
0	0	1	0	1	0
1	0	1	0	1	0

Octal (Base 8)

Step 1.

Step 2.

Leading zeros Trailing zero

$$\therefore (125.62)_8 = (1010101.11001)_2$$

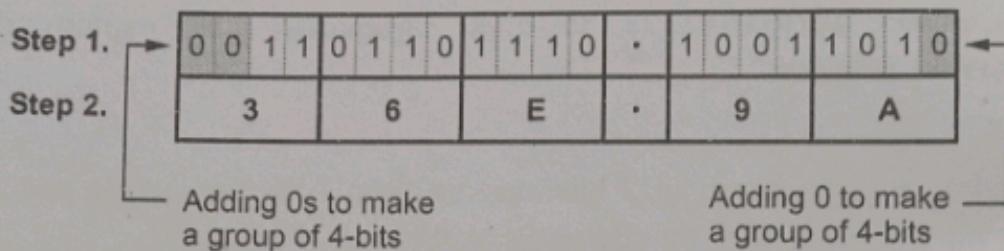
Q.7 Convert $1101101110 \cdot 1001101$ to hexadecimal equivalent and explain the conversion process.

Ans. :

Step 1 : Make group of 4-bits starting from LSB for integer part and MSB for fractional part, by adding 0s at the end, if required.

Step 2 : Write equivalent hexadecimal number for each group of 4-bits.

$$\therefore (1101101110.1001101)_2 = (36E.9A)_{16}$$



Q.8 Convert the following : $(62.7)_8 = (?)_{16} = (?)_2$.

Ans. :

6	2	.	7	Octal number
0	0	1	1	Binary number
3	2	.	E	Hex number

$$\therefore (62.7)_8 = (110010.1110)_2 = (32.E)_{16}$$

Q.9 Convert the base - 7 number $(35614)_7$ to base - 12.

Ans. : Step 1 : Convert to decimal

$$\begin{aligned}
 (35614)_7 &= \\
 3 \times 7^4 + 5 \times 7^3 + 6 \times 7^2 + 1 \times 7^1 + 4 \times 7^0 &= \\
 &= 7203 + 1715 + 294 + 7 + 4 \\
 &= 9223
 \end{aligned}$$

12	9223	7
12	768	0
12	64	4
12	5	5
12	0	

Step 2 : Convert to base 12

$$\therefore (35614)_7 = (5407)_{12}$$

Q.10 Convert $(735.25)_{10} = (?)_{16}$

Ans. :

[SPPU : May-17, Marks 2]

16	735	15 - F	4
16	45	13 - D	
16	2	2	

0

$$\therefore (735)_{10} = (2DF)_{16}$$

$$\therefore (735.25)_{10} = (2DF.4)_{16}$$

$$0.25 \times 16 = 4.0$$

Q.
S.
i)

Q.11 Convert $(101011.111011)_2 = (?)_8 = (?)_{16}$

ECE [SPPU : May-17, Marks 2]

Ans. :

$$\begin{array}{r} 101 \ 011 \cdot 111 \ 011 \\ \hline 5 \quad 3 \quad 7 \quad 3 \\ 0010 \ 1011 \cdot 1110 \ 1100 \\ \hline 2 \quad B \quad E \quad C \end{array}$$

$$(101011.111011)_2 = (53.73)_8 = (2B.EC)_{16}$$

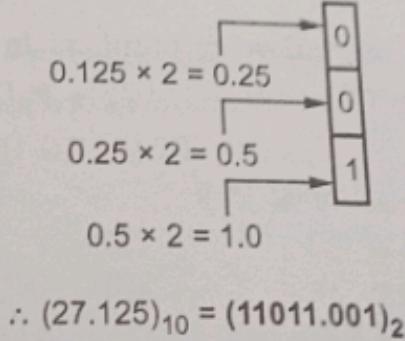
Q.12 Do the following conversions :

- i) $(27.125)_{10} = (?)_2$ ii) $(3A.2F)_{16} = (?)_{10}$ iii) $(1101.0011)_2 = (?)_{10}$
ECE [SPPU : Dec.-16, Marks 6]

Ans. :

i)

2	27	1
2	13	1
2	6	0
2	3	1
2	1	1
		0



$$\begin{aligned} \text{i)} \quad (3A.2F)_{16} &= 3 \times 16^1 + 10 \times 16^0 + 2 \times 16^{-1} + 15 \times 16^{-2} \\ &= 48 + 10 + 0.125 + 0.05859375 \\ &= (58.18359375)_{10} \end{aligned}$$

$$\begin{aligned} \text{ii)} \quad (1101.0011)_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} \\ &\quad + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} \\ &= 8 + 4 + 1 + 0.125 + 0.0625 = (13.1875)_{10} \end{aligned}$$

Q.13 Convert the following binary numbers to octal then to decimal.
Show the steps of conversions.

- i) 11011100.101010 ii) 01010011.010101 iii) 10110011

ECE [SPPU : Dec.-17, Marks 6]

Ans. :

i) $\begin{array}{r} \underline{011} & \underline{011} & \underline{100} \\ 3 & 3 & 4 \end{array} . \begin{array}{r} \underline{101} & \underline{010} \\ 5 & 2 \end{array}$ Binary
Octal

$$(334.52)_8 = 3 \times 8^2 + 3 \times 8^1 + 4 \times 8^0 + 5 \times 8^{-1} + 2 \times 8^{-2}$$

$$= 192 + 24 + 4 + 0.625 + 0.03125 = (220.65625)_{10}$$

ii) $\begin{array}{r} \underline{001} & \underline{010} & \underline{011} \\ 1 & 2 & 3 \end{array} . \begin{array}{r} \underline{010} & \underline{101} \\ 2 & 5 \end{array}$ Binary
Octal

$$(123.25)_8 = 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 + 2 \times 8^{-1} + 5 \times 8^{-2}$$

$$= 64 + 16 + 3 + 0.25 + 0.078125 = (83.328125)_{10}$$

iii) $\begin{array}{r} \underline{010} & \underline{110} & \underline{011} \\ 2 & 6 & 3 \end{array}$ Binary
Octal

$$(263)_8 = 2 \times 8^2 + 6 \times 8^1 + 3 \times 8^0 = 128 + 48 + 3 = (179)_{10}$$

Q.14 Convert the following numbers in binary form :

(i) $(125.12)_{10} = (?)_2$ (ii) $(337.025)_8 = (?)_2$ (iii) $(5DB.FA)_{16} = (?)_2$
[SPPU : Dec.-18, Marks 6]

Ans. :

i) $(125.12)_{10} =$

Integer part		Fractional part
2	125	$1 \times 0.12 \times 2 = 0.24$
2	62	$0 \times 0.24 \times 2 = 0.48$
2	31	$1 \times 0.48 \times 2 = 0.96$
2	15	$1 \times 0.96 \times 2 = 1.92$
2	7	$1 \times 1.92 \times 2 = 1.84$
2	3	$1 \times 1.84 \times 2 = 1.68$
2	1	$1 \times 1.68 \times 2 = 1.36$
	0	

$$\therefore (125.12)_{10} = (1111101.0001111)_2$$

ii) $(337.025)_{10} = (011\ 011\ 111.000\ 010\ 101)_2$
 iii) $(5DB.FA)_{18} = (0101\ 1101\ 1011.1111\ 1010)_2$

Unsolved Examples

Q.15 Convert the following numbers as indicated :

i) $(62.7)_8 = (?)_{16} = (?)_2$, ii) $(BC64)_{16} = (?)_{10} = (?)_2$ iii) $(111011)_2 = (?)_5$

Ans. : i) $(32.E)_{16}$, $(110010.111)_2$ ii) $(48228)_{10} = (1011110001100100)_2$
 iii) $(214)_5$

Q.16 Convert the following decimal numbers into their equivalent hexadecimal numbers and octal numbers.

- 1) 936 2) 1507 3) 23.56 4) 1.025 5) 100.5

[SPPU : May-13, Marks 10]

Ans. : 1) $(3A8)_{16}$, $(1650)_8$, 2) $(5E3)_{16}$, $(2743)_8$, 3) $(17.8F2)_{16}$, $(27.436)_8$,
 4) $(1.066)_{16}$, $(1.0146)_8$, 5) $(64.8)_{16}$, $(144.4)_8$

Q.17 Convert the following hexadecimal numbers into their equivalent octal numbers and binary numbers.

- 1) A72E 2) BD6.7 3) 0.AF54 4) DF 5) FF

[SPPU: May-13, Marks 10]

Ans. : 1) $(123456)_8$, $(1010011100101110)_2$,
 2) $(5726.34)_8$, $(101111010110.0111)_2$,
 3) $(0.53652)_8$, $(0.1010111010100)_2$,
 4) $(337)_8$, $(11011111)_2$, 5) $(3FF)_8$, $(11111111)_2$

Q.18 Convert the following numbers, show all steps :

i) $(2598.675)_{10} = (?)_{16}$ ii) $(110101.101010)_2 = (?)_8$

iii) $(A72E)_{16} = (?)_8$

[SPPU : Dec.-15, Marks 6]

Ans. : i) $(2598.675)_{10} = (A26.ACC)_{16}$

ii) $(110101.101010)_2 = (65.52)_8$

iii) $(A72E)_{16} = (1010011100101110)_2 = (123456)_8$

2.2 : Signed Binary Number Representation - Signed and True Magnitude, 1's Complement and 2's Complement Representation

Q.19 State different ways to represent negative binary number.

Ans. : There are three ways to represent negative numbers :

- Signed - magnitude representation.
- 1's complement representation.
- 2's complement representation.

Q.20 Find 1's complement of $(11010100)_2$.

Ans. :

Number
NOT operation
1's complement of number

The 1's complement of a binary number is the number that results when we change all 1's to zeros and the zeros to ones.

Q.21 Find 2's complement of $(11000100)_2$.

Ans. : The 2's complement is the binary number that results when we add 1 to the 1's complement.

1	1	0	0	0	1	0	0	Number
0	0	1	0	1	0	1	1	
+ 1								

1	1	1	1	1	0	1	1	Carry
0	0	1	1	1	0	1	1	
+ 1								
0	0	1	1	1	1	0	0	2's complement of number

Q.22 What is signed magnitude representation ? Represent + 9 and - 9 using 8-bit signed - magnitude form.

OR What do you mean by sign magnitude representation ? Discuss.

Ans. : In signed magnitude representation most significant bit is used to code sign of a number (1 = negative, 0 = +) and remaining bits are used to represent magnitude of the number in the range of $+ 2^{n-1}$ to $- 2^{n-1}$.

$$+9 = 0000 \ 1001$$

$$-9 = 1000 \ 1001$$

Q.23 Represent decimal number "- 13" in all three methods of negative binary number representation using eight bits.

Ans. :

1	0	0	0	1	1	0	1
---	---	---	---	---	---	---	---

- 13 in sign magnitude representation

1	1	1	1	0	0	1	0
---	---	---	---	---	---	---	---

- 13 in 1's complement representation

1	1	1	1	0	0	1	1
---	---	---	---	---	---	---	---

- 13 in 2's complement representation

2.3 : Binary Arithmetic

Q.24 Explain the binary addition operation.

Ans. :

1. Add bits column-wise starting from LSB with carry if any.
2. Put the sum at the bottom of the same column.
3. Put the carry, if any, on the top of next column.

Q.25 Add $(28)_{10}$ and $(15)_{10}$ by converting them into binary.

Ans. : Using decimal to binary conversion technique we have,

$$(28)_{10} = (011100)_2 \text{ and } (15)_{10} = (01111)_2$$

Sign Extension	+	Carry				$(28)_{10}$	
		1	1	1			
		0	0	1	1	0	0
		0	0	0	1	1	1
		0	1	0	1	0	1
						+ $(15)_{10}$	
						$\frac{(43)_{10}}$	
						Result : Binary equivalent of $(43)_{10}$	

Q.26 State the procedure to perform binary subtraction using 1's complement method.

Ans. : The operation $A - B$ is performed using 1's complement method as follows :

1. Take 1's complement of B.
2. Result $\leftarrow A + 1$'s complement of B.
3. If carry is generated then the result is positive and in the true form.
Add carry to the result to get the final result.

4. If carry is not generated then the result is negative and in the 1's complement form.

Q.27 Perform $(28)_{10} - (15)_{10}$ using 6-bit 1's complement representation.

Ans. : $(28)_{10} = (011100)_2$ $(15)_{10} = (001111)_2$

$(15)_{10}$

1's complement of $(15)_{10}$

Sign Extension	<table border="1" style="border-collapse: collapse; width: 100px;"> <tr><td>1</td><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>+ </td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td></td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>+ </td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td></td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> </table>	1	1						0	1	1	1	0	+	1	1	0	0	0		1	0	0	1	1	+	0	0	1	1	0		0	0	1	1	0	Carry Binary equivalent of $(28)_{10}$ 1's complement of 15, i.e. $(-15)_{10}$ Result Add end around carry Final result : Binary equivalent of $(13)_{10}$	$(28)_{10}$ $+ (-15)_{10}$ $\hline (13)_{10}$
1	1																																						
	0	1	1	1	0																																		
+	1	1	0	0	0																																		
	1	0	0	1	1																																		
+	0	0	1	1	0																																		
	0	0	1	1	0																																		

Q.28 Perform $(15)_{10} - (28)_{10}$ using 6-bit 1's complement representation.

Ans. : $(15)_{10} = (001111)_2$ $(28)_{10} = (011100)_2$

 	Binary equivalent of $(28)_{10}$ Carry 1's complement of $(28)_{10}$
------	--

 	Carry Binary equivalent of $(15)_{10}$ 1's complement of $(28)_{10}$ Result = Binary equivalent of $(-13)_{10}$	$(15)_{10}$ $+ (-28)_{10}$ $\hline (-13)_{10}$
------	--	--

Verification

1's complement of result (Binary equivalent of $(13)_{10}$)

Q.29 State the procedure to perform binary subtraction using 2's complement method.

$(28)_{10}$ $(-15)_{10}$ $(13)_{10}$

Ans. : The operation $A - B$ is performed using 2's complement method as follows :

1. Take 2's complement of B.
2. Result $\leftarrow A + 2$'s complement of B.
3. If carry is generated then the result is positive and in the true form. In this case, carry is ignored.
4. If carry is not generated then the result is negative and in the 2's complement form.

Q.30 Perform the following operations using 2's complement method.

i) $-(48)_{10} - (23)_{10}$ ii) $-(48)_{10} - (-23)_{10}$ [SPPU : Dec.-13, May-19, Marks 6]

Ans. :

i) $-(48)_{10} = 1010000$
$-(23)_{10} = 101001$

Carry
$-(48)_{10}$
$-(23)_{10}$
Result = $-(71)_{10}$

Sign extension	+	1	1	0	1	0	0	0	0	0	0
		1	1	0	1	0	0	0	1		
		1	0	1	1	1	0	0	1		

$\therefore (48)_{10} - (23)_{10} = -(71)_{10}$

ii) $-(48)_{10} - (-23)_{10} = -(48)_{10} + (23)_{10}$

$$-(48)_{10} = 1010000$$

$$(23)_{10} = 010111$$

Carry
$-(48)_{10}$
$(23)_{10}$
Result = $-(25)_{10}$

+	1	1	0	0	1	1	1	1	1	1	1
	1	0	1	0	0	0	0	0	0	0	0
	0	0	1	0	1	1	1	1	1	1	1

$\therefore -(48)_{10} - (-23)_{10} = -(25)_{10}$

Q.31 Subtract $(27.50)_2$ from $(68.75)_2$ using 2's complement method.

[SPPU : Dec.-14, May-17, Marks 4]

Ans. : Step 1 : Convert numbers to binary

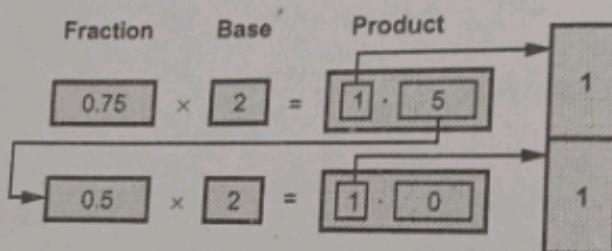
Logic D

Q.32

Ans. 3

2	27	1	
2	13	1	$0.5 \times 2 = 1.0$
2	6	0	$\therefore (27.50) = (11011.1)_2$
2	3	1	
2	1	1	
	0		

2	68	0
2	34	0
2	17	1
2	8	0
2	4	0
2	2	0
2	1	1
	0	



$$\therefore 68.75 = (1000100.11)_2$$

Step 2 : Take 2's complement of $(110111)_2$, V

0	1	1	0	1	1	1	·	1
1	0	0	1	0	0	0	·	0
+								
1	0	0	1	0	0	1	·	1

$$68.75 - 27.50 = (10100101)_2$$

Carry
 $(68.75)_{10}$
 2's complement of $(27.5)_{10}$
 Result

Q.32 Do $(7F)_{16} = (5C)_{16}$ using 2's complement method.

Ans. : $(7F)_{16} = (1111111)_2$ and $(5C)_{16} = (1011100)_2$

1	0	1	1	1	0	0
0	1	0	0	0	1	1
0	1	0	0	1	0	0

$(5C)_{16}$
1's complement
2's complement

1	1	1	1	1			Carry
	1	1	1	1	1	1	$(7F)_{16}$
+ 0	1	0	0	1	0	0	2's complement of $(5C)_{16}$
Discard carry	0	1	0	0	0	1	Result = (23)H

Q.33 Perform 2's complement arithmetics of :

- i) $(7)_{10} - (11)_{10}$ ii) $(-7)_{10} - (11)_{10}$ iii) $(-7)_{10} + (11)_{10}$

[SPPU : May-15, Marks 6]

Ans. : $(7)_{10} = (0111)_2$ $(11)_{10} = (01011)_2$

0	1	1	1
1	0	0	0
1	0	0	1

$(7)_{10}$

1's complement of 7
2's complement of 7

0	1	0	1	1
1	0	1	0	0
1	0	1	0	1

$(11)_{10}$

1's complement of 11
2's complement of 11

i) $(7)_{10} - (11)_{10}$

	1	1	1	1		Carry
+ 0	0	0	1	1	1	$(7)_{10}$
+ 1	0	1	0	1		2's complement of 11
+ 1	1	1	0	0		Result is in 2's complement form

ii) $(-7)_{10} - (11)_{10}$

1				1		Carry
Sign extension	→ 1	1	0	0	1	2's complement of 7
	1	0	1	0	1	2's complement of 11
+	1	0	1	1	0	Result is in 2's complement form

iii) $(-7)_{10} + (11)_{10}$

1	1			1	1	Carry
Sign extension	→ 1	1	0	0	1	2's complement of 7
	0	1	0	1	1	$(11)_{10}$
Discard carry	0	0	1	0	0	Result = 4

2.4 : Codes : BCD, Excess-3, Gray Binary Code and Their Conversion

Q.34 Give the classification of binary codes.

Ans. : The Fig. Q.34.1 shows the classification of codes.

- The codes are broadly classified as

1. Weighted codes	2. Non-weighted codes
3. Reflective codes	4. Sequential codes
5. Alphanumeric codes	6. Error detecting and correcting codes.

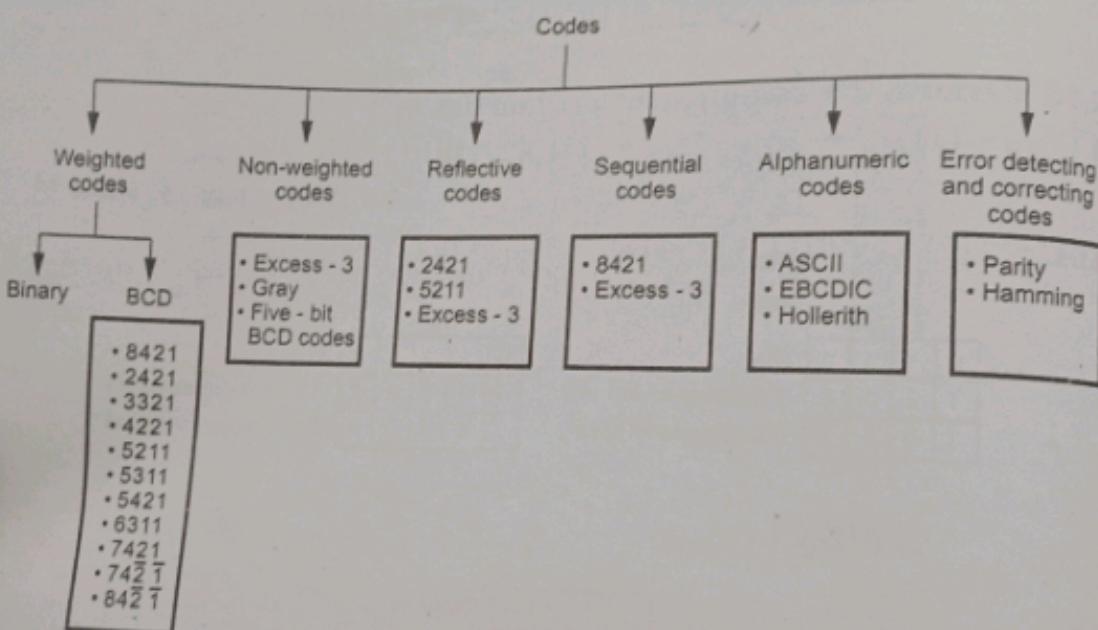


Fig. Q.34.1 Classification of codes

Q.35 Define weighted, non-weighted, reflective and sequential codes with examples.

Ans. : Weighted codes : • In weighted codes, each digit position of the number represents a specific weight.

• In weighted binary code each bit has a weight 8, 4, 2 or 1 and each decimal digit is represented by a group of four bits.

Non-weighted codes : • Non-weighted codes are not assigned with any weight to each digit position, i.e., each digit position within the number is not assigned fixed value.

• Excess-3 and gray codes are the non-weighted codes.

Reflective codes : • A code is said to be reflective when the code for 9 is the complement for 0, the code for 8 is complement for 1, 7 for 2, 6 for 3 and 5 for 4.

- Like 2421, codes 5211 and excess-3 are also reflective. The 8421 code is not reflective.

Sequential codes : • In sequential codes each succeeding code is one binary number greater than its preceding code.

- The 8421 and excess-3 are sequential, whereas the 2421 and 5211 codes are not.

Q.36 What are error detecting and correcting codes ?

Ans. : • To maintain the data integrity between transmitter and receiver, extra bit or more than one bit are added in the data. These extra bits allow the detection and sometimes correction of error in the data.

- The data along with the extra bit/bits forms the code. Codes which allow only error detection are called **error detecting codes** and codes which allow error detection and correction are called **error detecting and correcting codes**.

Q.37 What is BCD code ? What are rules for BCD addition ?

Ans. : • BCD is an abbreviation for binary coded decimal.

- BCD is a numeric code in which each digit of a decimal number is represented by a separate group of 4-bits. The most common BCD code is 8-4-2-1 BCD.
- It is called 8-4-2-1 BCD because the weights associated with 4 bits are 8-4-2-1 from left to right. This means that, bit-3 has weight 8, bit-2 has weight 4 bit-1 has weight 2 and bit-0 has weight 1.
- The Table Q.37.1 shows the 4-bit 8-4-2-1 BCD code used to represent a single decimal digit.
- In multidigit coding, each decimal digit is individually coded with 8-4-2-1 BCD code, as shown in the Fig. Q.37.1. Total 8-bits are required to encode 58_{10} in 8-4-2-1 BCD.

Decimal	5	8
8-4-2-1 BCD	0 1 0 1	1 0 0 0

Fig. Q.37.1

Rules for BCD addition

- BCD addition rules are as follows :

- Add two BCD numbers using ordinary binary addition.
- If four-bit sum is equal to or less than 9, no correction is needed. The sum is in proper BCD form.

- If the four-bit sum is greater than 9 or if a carry is generated from the four-bit sum, the sum is invalid.
 - To correct the invalid sum, add 6 (0110_2) to the four-bit sum and ignore carry of this sum.
 - Add 1(0001_2) to the next higher-order BCD digit.

Q.38 Represent the unsigned decimal number 965 and 672 in BCD and then show the steps necessary to find their sum.

Ans. :

		11		
1001	0110	0101	(965) ₁₀	
+ 0110	0111	0010	(672) ₁₀	
1111	1101	0111		
0110	0110			
1 0101	1 0011	0111	Add 6 for correction	
	1			
0001 0110	0011	0111	Add carry	
1 6	3	7	BCD Sum	
			Equivalent decimal	

DECODE®

ode
2
0
0
1
1
0
1
0
1
0
1
0
1

de
from the
bit sum

in BCD

Q.39 What are the rules for BCD subtraction using 9's complement?

Ans. :

Step 1 : Find the 9's complement of a negative number.

Step 2 : Add two numbers using BCD addition (Minuend + 9's complement of subtrahend).

Step 3 : If carry is not generated result is negative and find the 9's complement of the result, otherwise result is positive and add carry to the result.

Q.40 Perform the following subtractions of BCD numbers using 9's complement : i) $68 - 24$ ii) $24 - 29$

Ans. : i)

Step 1 : 9's complement of 24 = $99 - 24 = 75$

Step 2 : Add 68 and 9's complement of 24.

$ \begin{array}{r} 1 \ 1 \\ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \\ + 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \\ \hline 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \end{array} $	(68) BCD
$ \begin{array}{r} 1 \ 1 \\ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \\ + 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \\ \hline 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \end{array} $	(75) 9's complement of (24)BCD

$ \begin{array}{r} 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \\ + 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \\ \hline 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \end{array} $	Invalid BCD numbers
$ \begin{array}{r} 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \\ + 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \\ \hline 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \end{array} $	Add 6 in each digit
$ \begin{array}{r} 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \\ + 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \\ \hline 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \end{array} $	Add end around carry
$ \begin{array}{r} 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \\ \hline 4 \qquad\qquad\qquad 4 \end{array} $	Result

Since there is a carry, result is positive and true

$$\therefore (68)_{BCD} - (24)_{BCD} = (44)_{BCD}$$

ii)

Step 1 : 9's complement of 29 = $99 - 29 = 70$

Step 2 : Add 24 and 9's complement of 70

1	1					
0	0	1	0	0	1	0
0	1	1	1	0	0	0
Carry	0	1	0	0	1	0

(24)BCD (70) 9's complement of (29) BCD
 (94)BCD

Since carry is zero the answer is negative.

- Step 3 :** Take 9's complement of answer
 $9's \text{ complement of } 94 = 99 - 94 = (5) \text{ BCD}$
 $\therefore (24) \text{ BCD} - (29) \text{ BCD} = -(5) \text{ BCD}$

Unsolved Examples

Q.41 Perform $(46)_{10} - (22)_{10}$ in BCD using 9's complement.

Q.42 Perform $(24)_{10} - (56)_{10}$ in BCD using 9's complement.

Q.43 What are the rules for BCD subtraction using 10's complement?

Ans. : Steps for subtraction of BCD numbers using 10's complement method

Step 1 : Find the 10's complement of a negative number.

Step 2 : Add two numbers using BCD addition (Minuend + 10's complement of subtrahend).

Step 3 : If carry is not generated result is negative and find the 10's complement of the result, otherwise result is positive and discard carry.

Q.44 Perform $(46)_{10} - (22)_{10}$ in BCD using 10's complement.

Ans. :

Step 1 : Find 10's complement of 22.

$$\text{10's complement of } 22 = \text{9's complement of } 22 + 1$$

$$= (99 - 22) + 1 = 78$$

Step 2 : Add 46 and 10's complement of 22.

Disc

Since ther

\therefore

Q.45 P

Ans. :

Step 1

Step 2

Since

Step

Q.46

(i) 2

Ans.

	1								
+ 46	0	1	0	0	0	1	1	0	
- 22	0	1	1	1	1	0	0	0	
24	1	0	1	1	1	1	1	0	
	1	1	1	1	1	1			
	1	0	1	1	1	1	1	0	
	+	0	1	1	0	0	1	1	0
Discard carry	<input checked="" type="checkbox"/>	0	0	1	0	0	1	0	0
			2			4			

Carry

 $(46)_{BCD}$ $(78)_{10}$'s complement of $(22)_{BCD}$

Carry

Invalid BCD numbers

Add 6 in each digit

Result

Since there is a carry the result is positive and true.

$$\therefore (46)_{BCD} - (22)_{BCD} = (24)_{BCD}$$

Q.45 Perform $(24)_{10} - (56)_{10}$ in BCD using 10's complement.

Ans. :

Step 1 : Find 10's complement of 56.

$$\begin{aligned} \text{10's complement of } 56 &= \text{9's complement of } 56 + 1 \\ &= (99 - 56) + 1 = 44 \end{aligned}$$

Step 2 : Add $(24)_{10}$ and 10's complement of 56.

			1			
24		0	0	1	0	0
- 56		0	1	0	0	0
- 32	Carry	0	0	1	1	0
		0	1	0	1	0

Carry

 $(24)_{BCD}$ 10's complement of $(56)_{BCD}$ $(68)_{BCD}$

Since carry is 0 the answer is negative.

Step 3 : Take 10's complement of answer.

$$\begin{aligned} \text{10's complement of } 68 &= \text{9's complement of } 68 + 1 \\ &= (99 - 68) + 1 = 32 \end{aligned}$$

$$\therefore (24)_{10} - (56)_{10} = -(32)_{10}$$

Q.46 Find 9's and 10's complement of the following decimal numbers :

- (i) 24,681,234 (ii) 63,325,600

Ans. : i) $99999999 - 24681234 = 75318765$ 9's complement

$75318765 + 1 = 75318766$ 10's complement

ii) $99999999 - 63325600 = 36674399$ 9's complement

$36674399 + 1 = 36674400$ 10's complement

Q.47 Write a note on Excess-3 code.

Ans. : • The excess-3 code can be derived from the natural BCD code by adding 3 to each coded number.

- For example, decimal 12 can be represented in BCD as 0001 0010. Now adding 3 to each digit we get Excess-3 code as 0100 0101 (12 in decimal). It is a **non-weighted code**.

- Table Q.47.1 shows excess-3 codes to represent single decimal digit. It is **sequential code** because we get any code word by adding binary 1 to its previous code word as shown in the Table Q.47.1.
- In excess-3 code we get 9's complement of a number by just complementing each bit. Due to this excess-3 code is called **self-complementing code or reflective code**.

Q.48 Represent the decimal number 6 in Excess-3 code.

Ans. :

+	0 1 1 0	6 in BCD
	0 0 1 1	3
	1 0 0 1	6 in Excess-3

Q.49 Convert the decimal number 25 into binary format, excess-3 format and BCD format.

☞ [SPPU : May-18, Marks 3]

Logic Design
Ans. :

Q.50 Sta

Ans. : T

• Add tv

• If car

Q.51 Pe

Ans. :

a) $8 + 6$

Q.52 S

Ans. :

• Comp

• Add

• If ca

If ca

Ans.

2	25	1
2	12	0
2	6	0
2	3	1
2	1	1
	0	

$$(25)_{10} = (11001)_2 = (0010\ 0101)_{BCD}$$

$$= (0101\ 1000)_{\text{excess-3}}$$

Q.50 State the rules for Excess-3 addition.

Ans. : To perform excess-3 addition we have to

- Add two excess-3 numbers using binary addition.
 - If carry = 1 \rightarrow add 3 (0 0 1 1)₂ to the sum of two digits.
= 0 \rightarrow subtract 3 (0 0 1 1)₂ from the sum.

Q.51 Perform the excess-3 addition of a) 8, 6 b) 1, 2.

ANS.:

a) $8 + 6$	<table border="1"> <tr> <td>1</td><td>1</td><td>1</td><td></td><td>Carry</td></tr> <tr> <td></td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr> <td>+ 1</td><td>0</td><td>0</td><td>1</td><td></td></tr> <tr> <td>0 0 1 1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr> <td>0 1 0 0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td></td><td>4</td><td></td><td></td></tr> </table>	1	1	1		Carry		1	0	1	1	+ 1	0	0	1		0 0 1 1	0	0	1	1	0 1 0 0	0	1	1	1	1		4		
1	1	1		Carry																											
	1	0	1	1																											
+ 1	0	0	1																												
0 0 1 1	0	0	1	1																											
0 1 0 0	0	1	1	1																											
1		4																													

$1 + 2$	<table border="1"> <tr><td>1</td><td></td><td></td><td></td></tr> <tr><td></td><td>0</td><td>1</td><td>0</td></tr> <tr><td></td><td>+</td><td>0</td><td>1</td></tr> <tr><td>Carry</td><td>0</td><td>1</td><td>0</td></tr> <tr><td></td><td>-</td><td>0</td><td>0</td></tr> <tr><td></td><td></td><td>1</td><td>1</td></tr> </table>	1					0	1	0		+	0	1	Carry	0	1	0		-	0	0			1	1	Carry
1																										
	0	1	0																							
	+	0	1																							
Carry	0	1	0																							
	-	0	0																							
		1	1																							
		Excess-3 for 1																								
		Excess-3 for 2																								
		Carry = 0, hence																								
		Sub 3																								
		Excess-3 for 3																								
		Result in decimal																								

Q.52 State the rules for Excess-3 subtraction.

Ans. : To perform excess-3 subtraction we have to

- Complement the subtrahend.
 - Add complemented subtrahend to minuend.
 - If carry = 1 Result is positive. Add 3 ($0\ 0\ 1\ 1$)₂ and end-around carry.
If carry = 0 Result is negative. Subtract 3 ($0\ 0\ 1\ 1$)₂.

Q.53 Perform the excess-3 subtraction of a) $8 - 5$, b) $5 - 8$.

Ans. : a) $8 - 5$

$ \begin{array}{r} 1 & 1 & 1 & 1 \\ + & 1 & 0 & 1 & 1 \\ \hline \text{Carry} & 1 & 0 & 0 & 1 & 0 \\ & 0 & 0 & 1 & 1 \\ \hline & 0 & 1 & 0 & 1 \\ \hline & 0 & 1 & 1 & 0 \end{array} $					
Excess-3 for 8					
Complement of 5 in excess-3					
Add 3					

b) $5 - 8$

$ \begin{array}{r} 1 & 0 & 0 & 0 \\ + & 0 & 1 & 0 & 0 \\ \hline \text{Carry} & 0 & 1 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 1 \\ \hline & 1 & 0 & 0 & 1 \end{array} $					
Excess-3 for 5					
Complement of 8 in excess-3					

Subtract 3

Excess-3 for -3

Q.54 Write a short note on gray code.

Ans. :

- Gray code is a non-weighted code and is a special case of unit-distance code.
- In unit-distance code, bit patterns for two consecutive numbers differ in only one bit position. These codes are also called cyclic codes.
- The Table Q.54.1 shows the bit patterns assigned for gray code from decimal 0 to decimal 15.

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

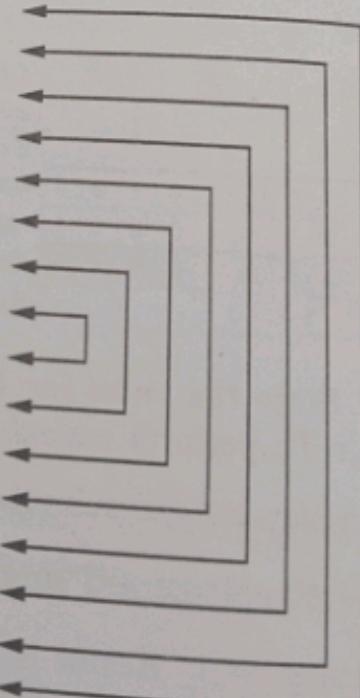


Table Q.54.1 Gray code

- As shown in groups differ

• Reflective P
Notice that
mirror imag
significant b
 0_{10} through

Q.55 State th

Ans. : Steps

1. The Most
the Most
down MS

2. To obtain
between
down the

3. Repeat s
binary d

Q.56 Conv

Ans. :

Q.57 Stat

Ans. : 1.
binary nu

2. To ob
between

3. Repea
previo

- * As shown in the Table Q.54.1 for gray code any two adjacent code groups differ only in one bit position.

* **Reflective Property :** The gray code is also called reflected code. Notice that the two least significant bits for 4_{10} through 7_{10} are the mirror images of those for 0_0 through 3_{10} . Similarly, the three least significant bits for 8_{10} through 15_{10} are the mirror images of those for 0_{10} through 7_{10} .

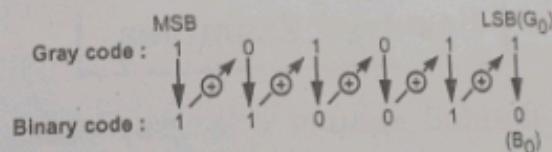
Q.55 State the rules for gray to binary code conversion.

Ans. : Steps for gray to binary code conversion

1. The Most Significant Bit (MSB) of the binary number is the same as the Most Significant Bit (MSB) of the gray code number. So write down MSB as it is.
2. To obtain the next binary digit, perform an exclusive-OR-operation between the bit just written down and the next gray code bit. Write down the result.
3. Repeat step 2 until all gray code bits have been exclusive-ORed with binary digits.

Q.56 Convert gray code 101011 into its binary equivalent.

Ans. :



$$(101011)_{\text{gray}} = (110010)_2$$

Q.57 State the rules for binary to gray code conversion.

Ans. : 1. The MSB of the gray code is the same as the MSB of the binary number. So write down MSB as it is.

2. To obtain the next gray digit, perform an exclusive-OR-operation between previous and current binary bit. Write down the result.
3. Repeat step 2 until all binary bits have been exclusive-ORed with their previous ones.

Q.58 Encode the decimal number 46 to Gray code.

Ans. : $(46)_{10} = (101110)_2$

Binary code :	1	- ⊕ -	0	- ⊕ -	1	- ⊕ -	1	- ⊕ -	1	- ⊕ -	0
Gray code :	1	↓	1	↓	1	↓	0	↓	0	↓	1

∴ $(46)_{10} = (111001)_{\text{Gray}}$

Q.59 Convert the decimal number 27 into : i) Binary ii) Excess-3
iii) Gray iv) HEX. [SPPU : May-16, Marks 6]

Ans. :

2	27.	1
2	13	1
2	6	0
2	3	1
2	1	1
	0	

∴ $(27)_{10} = (11011)_2 = (1B)_{\text{HEX}}$

$(27)_{10} = (0010 \ 0111)_{\text{BCD}}$
 $= (0101 \ 1010)_{\text{Excess - 3}}$

Binary :	1	- ⊕ -	1	- ⊕ -	0	- ⊕ -	1	- ⊕ -	1
Gray :	1	↓	0	↓	1	↓	1	↓	0

∴ $(27)_{10} = (10110)_{\text{Gray}}$

Unsolved Examples

Q.60 Represent the decimal number 6 in gray code.

Ans. : 0101

2.5 : IEEE Standard 754 Floating Point Number Representation

Q.61 Explain the IEEE754 standards for representing floating point numbers.

[SPPU : May-13,14,16, Dec.-15]

Ans. : To accommodate very large integers and very small fractions, a computer must be able to represent numbers and operate on them in such a way that the position of the binary point is variable and is automatically adjusted as computation proceeds. In this case, the binary point is said to float, and the numbers are called floating-point numbers.

- The standards for representing floating point numbers in 32-bits and 64-bits have been developed by the Institute of Electrical and Electronics Engineers (IEEE), referred to as IEEE 754 standards. Fig. Q.61.1 shows these IEEE standard formats.

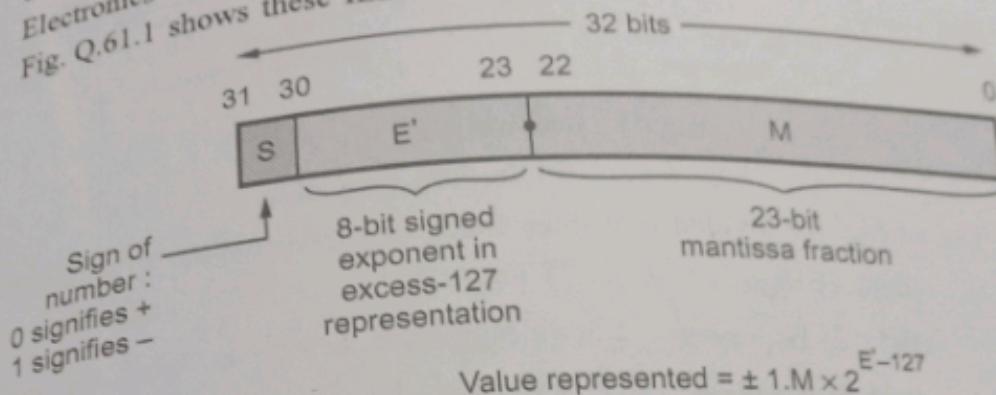


Fig. Q.61.1 (a) Single-precision

- The 32-bit standard representation shown in Fig. Q.61.1 (a) is called a **single-precision representation** because it occupies a single 32-bit word. The 32-bits are divided into three fields as shown below :
 - (field 1) Sign $\leftarrow 1$ - bit
 - (field 2) Exponent $\leftarrow 8$ - bits
 - (field 3) Mantissa $\leftarrow 23$ - bits
- Instead of the signed exponent, E, the value actually stored in the exponent field is $E' = E$ (scaling factor) + bias.
- In the 32-bit floating point system (single precision), bias is 127. Hence $E' = E$ (scaling factor) + 127. This representation of exponent is also called as the *excess-127 format*.
- The end values of E' , namely, 0 and 255, are used to indicate the floating point values of exact zero and infinity, respectively in single precision.
- Thus range of E' for normal values in single precision is $0 < E' < 255$. This means that for 32-bit representation the actual exponent E is in the range $-126 \leq E \leq 127$.
- The 64-bit standard representation shown in Fig. Q.61.1 (b) is called a **double-precision representation** because it occupies two 32-bit words.

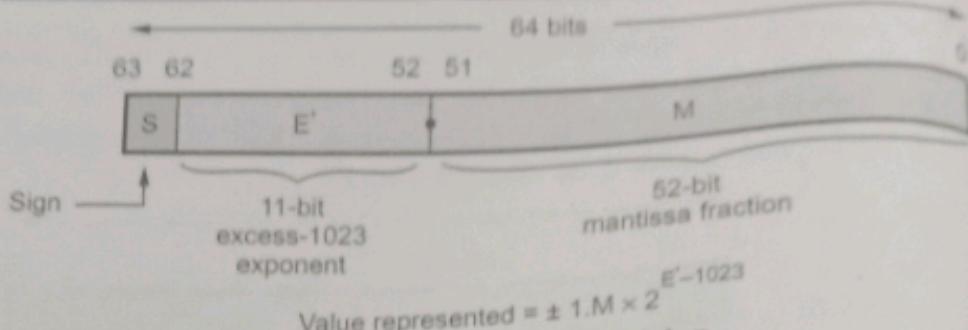


Fig. Q.61.1 (b) Double precision

The 64-bits are divided into three fields as shown below :

- (field 1) Sign \leftarrow 1 - bit
- (field 2) Exponent \leftarrow 11 - bit
- (field 3) Mantissa \leftarrow 52 - bits

- In the double precision format value actually stored in the exponent field is given as
$$E' = E + 1023$$
- Here, bias value is 1023 and hence it is also called excess-1023 format.
- The end values of E' , namely, 0 and 2047, are used to indicate the floating point exact values of exact zero and infinity, respectively.
- Thus the range of E' for normal values in double precision is $0 < E' < 2047$. This means that for 64-bit representation the actual exponent E is in the range.

Normalization Process

To represent the number in floating point format, first binary point is shifted to right of the first bit and the number is multiplied by the correct scaling factor to get the same value. The number is said to be in the **normalized form** and is given as

$$111101.1000110 \rightarrow 1.\underbrace{11101100110}_{\text{Significant digits}} \times 2^{\underbrace{5}_{\text{Scaling factor}}} \quad \boxed{\text{Exponent}}$$

Fig. Q.61.2 Number represented in normalized form

Q.62 Represent the following in single precision floating point format as well as double precision format.

- i) 17.125
- ii) 12.5

Ans. : i) 17.125

Step 1 : Decimal to Binary conversion

$$(17)_{10} = (10001)_2 \quad 0.125 \times 2 = 0.25 \quad 0$$

$$\therefore (17.125)_{10} = (10001001)_2 \quad 0.25 \times 2 = 0.5 \quad 0$$

$$\therefore \quad \quad \quad \quad \quad 0.5 \times 2 = 1.0 \quad 1$$

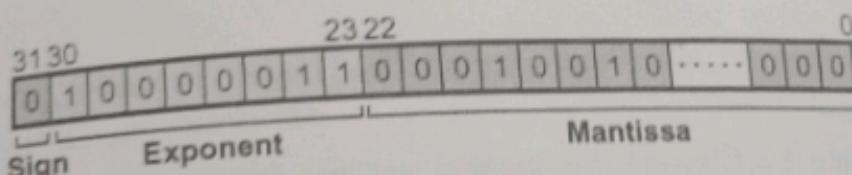
Step 2 : Normalization

$$10001.001 = 1.0001001 \times 2^4$$

Step 3 : Single precision representation

$$S = 0, E = 4 \text{ and } M = 0001001$$

$$E' = 127 + 4 = (131)_{10} = (10000011)_2$$



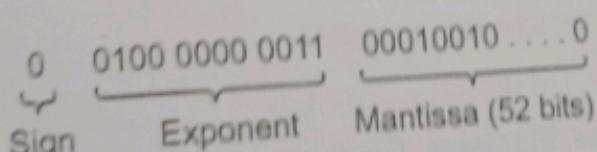
Step 4 : Double precision format

Bias for double precision format = 1023

$$\therefore E' = E + 1023 = 4 + 1023 = (1027)_{10} = (0100\ 0000\ 0011)_2$$

\therefore Number in double precision format is given as

ii) 12.5



Step 1 : Decimal to Binary conversion

$$\therefore (12)_{10} = (1100)_2 \quad 0.5 \times 2 = 1.0 = 1$$

$$\therefore 12.5 = 1100.1$$

Step 2 : Normalization

$$1100.1 = 1.1001 \times 2^3$$

Step 3 : Single precision representation

$S = 0$, $E = 3$ and $M = 1001$

$$E' = 127 + 3 = (130)_{10} = (10000010)_2$$

Sign	Exponent	Mantissa
1	10000010	010000100000.....000

Step 4 : Double precision format

Bias for double precision format = 1023

$$E' = E + 1023 = 3 + 1023 = (1026)_{10} = (0100\ 0000\ 001)_2$$

∴ Number in double precision format is given as

Sign Exponent Mantissa (52 bits)

Q.63 Represent following number in single precision format :
 $(0.625)_{10}$.

[SPPU : May-13, Marks 10]

Ans. : Step 1 : Convert the decimal number into binary format.

Fractional part :

$$\begin{array}{r}
 0.625 \times 2 = 1.25 \\
 0.25 \times 2 = 0.5 \\
 0.5 \times 2 = 1.0
 \end{array}
 \quad \begin{array}{r}
 1 \\
 0 \\
 1
 \end{array}
 \quad \begin{array}{l}
 \text{MSD} \\
 \downarrow \\
 \text{LSD}
 \end{array}$$

$$\therefore (0.625)_{10} = (0101)_2$$

Step 2 : Normalize the number

$$0.1010 = 1.010 \times 2^{-1}$$

Step 3 : Single precision format

For the given number

$$S = 0, E = -1, M = 01$$

Bias for single precision format = 127

$$E' = E + 127$$

$$= -1 + 127 = (126)_{10}$$

$$= (01110100)_2$$

∴ Number in single precision format is given as

0 01110100 010.....0
Sign Exponent Mantissa(23 bits)

END... ↗

Unit 1

3

Logic Minimization

3.1 : Review of Boolean Algebra

Postulates and Theorems

- The postulates and theorems of Boolean algebra as shown in Table 3.1.

Postulates	(a)	(b)
Postulate 1	Result of each operation is either 0 or 1 $\in B$	
Postulate 2 (Identity)	$A + 0 = A$	$A \cdot 1 = A$
Postulate 3 (Commutative)	$A + B = B + A$	$AB = BA$
Postulate 4 (Distributive)	$A(B + C) = AB + AC$	$A + BC$ $= (A + B)(A + C)$
Postulate 5 (Complement)	$A + \bar{A} = 1$	$A \cdot \bar{A} = 0$
Theorems	(a)	(b)
Theorem 1 (Idempotency)	$A + A = A$	$A \cdot A = A$
Theorem 2	$A + 1 = 1$	$A \cdot 0 = 0$
Theorem 3 (Involution)		$\bar{\bar{A}} = A$
Theorem 4 (Absorption)	$A + AB = A$	$A(A + B) = A$
Theorem 5	$A + \bar{A}B = A + B$	$A \cdot (\bar{A} + B) = AB$
Theorem 6 (Associative)	$A + (B + C)$ $= (A + B) + C$	$A(BC) = (AB)C$

Table 3.1 Postulates and basic theorems of Boolean algebra

DeMorgan's Theorems

1. $\overline{AB} = \overline{A} + \overline{B}$

2. $\overline{A+B} = \overline{A} \cdot \overline{B}$

Principle of Duality

The principle of duality theorem says that, starting with a Boolean relation, you can derive another Boolean relation by

1. Changing each OR sign to an AND sign
2. Changing each AND sign to an OR sign and
3. Complementing any 0 or 1 appearing in the expression.

For example : Dual of relation $A + \overline{A} = 1$ is $A \cdot \overline{A} = 0$

Q.1 Prove that $A + \overline{AB} = A + B$

Ans. :

$$\begin{aligned}
 A + \overline{AB} &= A + AB + \overline{AB} && \text{by Theorem : 4(a)} \\
 &= A + B \cdot (A + \overline{A}) && \text{by Postulate : 4(a)} \\
 &= A + B \cdot 1 && : 5(a) \\
 &= A + B && : 2(b)
 \end{aligned}$$

Q.2 Prove the following equations using Boolean algebra.

i) $XY + XYZ + XY\bar{Z} + \bar{X}YZ = Y(X + Z)$

ii) $\overline{ABC}\bar{D} + BCD + B\bar{C}\bar{D} + B\bar{C}D = B(\bar{D} + \bar{C})$

[SPPU : Dec.-11, June-11, Marks 6]

Ans. : i) $xy + xyz + xy\bar{z} + \bar{x}yz = xy(1 + z + \bar{z}) + \bar{x}yz = xy + \bar{x}yz$
 $= y(x + \bar{x}z) = y(x + z)$
 $\therefore A + \overline{AB} = A + B \dots \text{Proved}$

ii) $\overline{ABC}\bar{D} + BCD + B\bar{C}\bar{D} + B\bar{C}D = BCD(\bar{A} + 1) + B\bar{C}(\bar{D} + D)$
 $= BCD + B\bar{C} \quad \because \bar{A} + 1 = 1 \text{ and } \bar{A} + A = 1$
 $= B(\bar{C} + C\bar{D}) = B(\bar{C} + \bar{D}) \quad \because A + \overline{AB} = A + B$
 $\dots \text{Proved}$

Q.3 Using Boolean algebra show that :

$$\text{i) } \overline{ABC}\overline{D} + B\overline{C}\overline{D} + B\overline{C}\overline{D} + B\overline{C}D = B(\overline{D} + \overline{C})$$

$$\text{ii) } AB + \overline{AC} + A\overline{BC} (AB + C) = 1$$

$$\text{iii) } \overline{AB}\overline{CD} + \overline{ABC}\overline{D} + ABD = BD$$

EE^W [SPPU : Dec.11, Marks 10]

Ans. : i) Refer Q. 2 (ii).

$$\text{ii) } AB + \overline{AC} + A\overline{BC} (AB + C) = AB + \overline{A} + \overline{C} + AAB\overline{B}C + A\overline{B}CC$$

$$\because \overline{AC} = \overline{A} + \overline{C}$$

$$= AB + \overline{A} + \overline{C} + A\overline{B}C \quad \because A\overline{A} = 0$$

$$= \overline{A} + AB + \overline{C} + \overline{A} + A\overline{B}C \quad \because A = A + A$$

$$= \overline{A} + B + \overline{C} + \overline{A} + \overline{B}C \quad \because A + \overline{AB} = A + B$$

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

$$= \overline{A} + B + \overline{C} + \overline{B} \quad \because A + \overline{AB} = A + B$$

$$= 1 \quad \because B + \overline{B} = 1 \dots \text{Proved}$$

$$\text{iii) } \overline{AB}\overline{CD} + \overline{ABC}\overline{D} + ABD = \overline{ABD}(\overline{C} + C) + ABD$$

$$= \overline{ABD} + ABD \quad \because A + \overline{A} = 1$$

$$= BC(\overline{A} + A)$$

$$= BC \quad \because A + \overline{A} = 1 \dots \text{Proved}$$

3.2 : Representation of Logic Functions

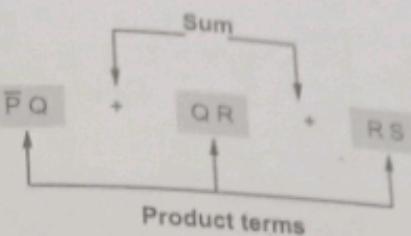
Concepts

- Each occurrence of a variable in either a complemented or an uncomplemented form in the Boolean function is called a literal.
- A product term is defined as either a literal or a product (also called conjunction) of literals.
- A sum term is defined as either a literal or a sum (also called disjunction) of literals.
- Literals and terms are arranged in one of the two forms :
 - Sum Of Product form (SOP) and
 - Product Of Sum form (POS)

DECODE®

- A Sum Of Products (SOP) is a group of product terms ORed together.
- Example**

$$f(P, Q, R, S) =$$



: Dec.11, Marks 10]

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

$$\because \bar{A}\bar{C} = \bar{A} + \bar{C}$$

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

$$\therefore A = A + A$$

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

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

= 1 ... Proved

$$\therefore A + \bar{A} = 1$$

1 ... Proved

ted or an
ral.

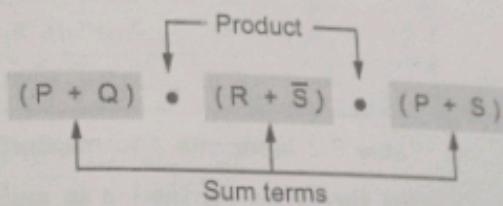
also called

so called

- A product of sums is any groups of sum terms ANDed together.

Example

$$f(P, Q, R, S) =$$



- If each term in SOP form contains all the literals then the SOP form is known as **canonical SOP form**.
- Each individual term in the canonical SOP form is called **minterm**.
- If each term in POS form contains all the literals then the POS form is known as **canonical POS form**.
- Each individual term in the canonical POS form is called **maxterm**.
- The POS and SOP functions derived from the same truth table are logically equivalent.
- In terms of minterms and maxterms we can then write

$$f(A, B, C) = m_0 + m_1 + m_3 + m_4 + m_6 + m_7 = M_2 + M_5$$

$$\therefore f(A, B, C) = \sum m(0, 1, 3, 4, 6, 7) = \pi M(2, 5)$$

Variables			Minterms	Maxterms
A	B	C	m_i	M_i
0	0	0	$\bar{A} \bar{B} \bar{C} = m_0$	$A + B + C = M_0$
0	0	1	$\bar{A} \bar{B} C = m_1$	$A + B + \bar{C} = M_1$
0	1	0	$\bar{A} B \bar{C} = m_2$	$A + \bar{B} + C = M_2$
0	1	1	$\bar{A} B C = m_3$	$A + \bar{B} + \bar{C} = M_3$
1	0	0	$A \bar{B} \bar{C} = m_4$	$\bar{A} + B + C = M_4$
1	0	1	$A \bar{B} C = m_5$	$\bar{A} + B + \bar{C} = M_5$
1	1	0	$A B \bar{C} = m_6$	$\bar{A} + \bar{B} + C = M_6$
1	1	1	$A B C = m_7$	$\bar{A} + \bar{B} + \bar{C} = M_7$

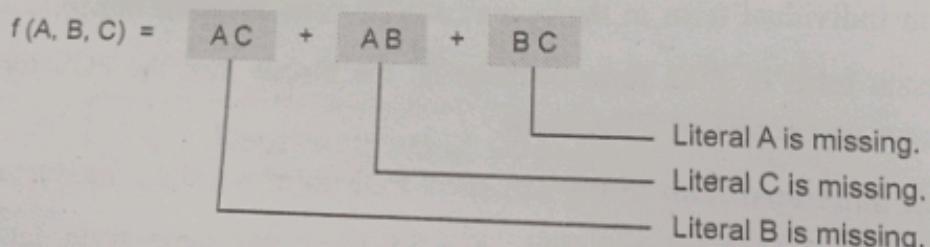
Table 3.2 Minterms and maxterms for three variables

Q.4 Convert the given expression in canonical SOP form.

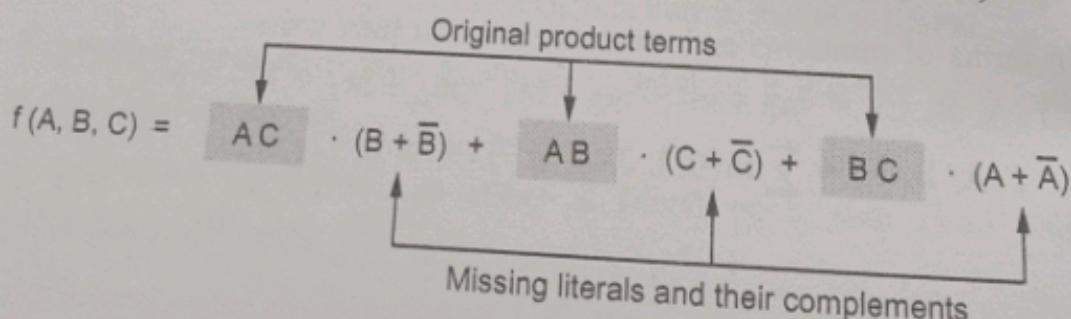
$$f(A, B, C) = AC + AB + BC$$

Ans. :

Step 1 : Find the missing literal/s in each product term.



Step 2 : AND product term with (missing literal + its complement).



Step 3 : Expand the terms and reorder literals.

$$\text{Expand : } f(A, B, C) = A C B + A C \bar{B} + A B C + A B \bar{C} + B C A + B C \bar{A}$$

$$\text{Reorder : } f(A, B, C) = A B C + A \bar{B} C + A B C + A B \bar{C} + A B C + \bar{A} B C$$

Note After having sufficient practice student should expand product term and reorder literals in it in a single step.

Step 4 : Omit repeated product terms.

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

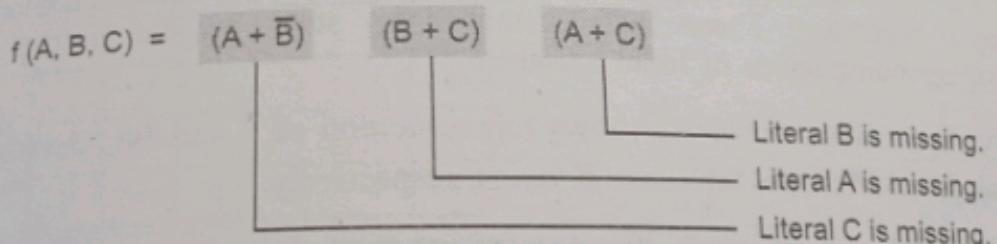
$$\therefore f(A, B, C) = A B C + A \bar{B} C + A B \bar{C} + \bar{A} B C$$

Q.5 Convert the given expression in canonical POS form.

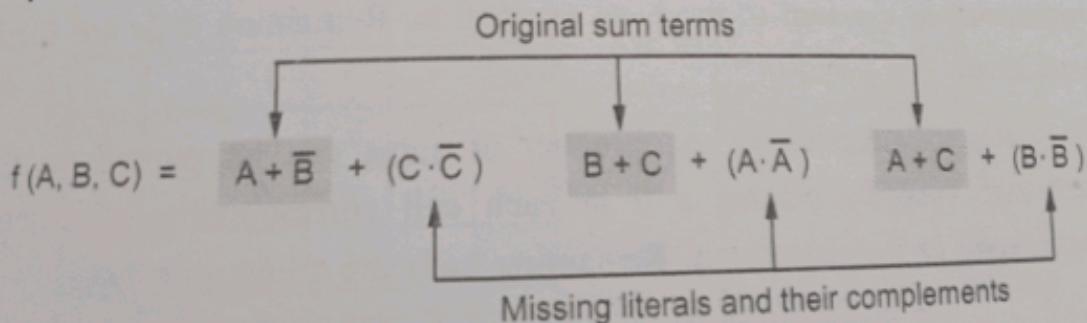
$$f(A, B, C) = (A + \bar{B})(B + C)(A + C)$$

Ans. : Find the missing literal/s in each sum term.

Step 1 : Find the missing literal/s in each sum term.



Step 2 : OR sum term with (missing literal • its complement).



Step 3 : Expand the terms and reorder literals.

Expand :

Since $A + BC = (A + B)(A + C)$ we have,

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

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

Reorder :

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

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

Step 4 : Omit repeated sum terms.

Repeated sum terms

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

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

3.3 : Simplification of Logical Functions, using K-Maps up to 4 Variables

Q.6 What is K-map ? How to represent truth table on K-map ?

Ans. : • K-map (Karnaugh map) is a graphical way to represent a truth table and it is a method by which we can simplify boolean expressions.

- Each Karnaugh map consists of 2^n cells - where n represents number of input variables. These cells contain the state of the output for corresponding states of input variables.
- Fig. Q.6.1 (a), (b) and (c) shows representation of 2-variable, 3-variable and 4-variable truth tables on K-maps, respectively.
- A Boolean expression in the sum of products form can be plotted on the Karnaugh map by placing a 1 in each cell corresponding to a term (**minterm**) in the sum of products expression. Remaining cells are filled with zeros.
- A Boolean expression in the product of sums can be plotted on the Karnaugh map by placing a 0 in each cell corresponding to a term (**maxterm**) in the expression. Remaining cells are filled with ones.

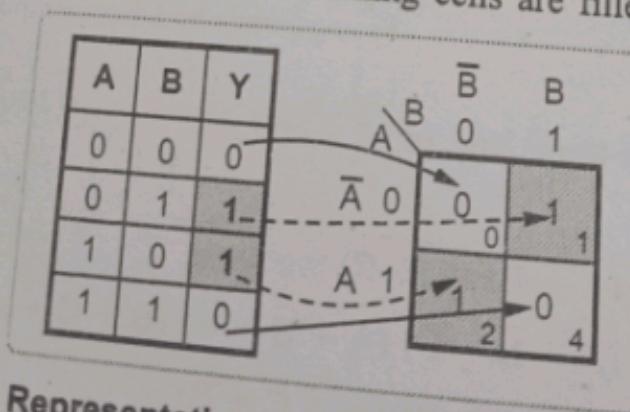


Fig. Q.6.1 (a) Representation of 2-variable truth table on K-map

DECODE®

No.	A	B	C	Y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

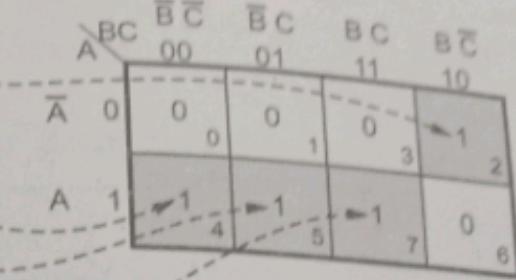
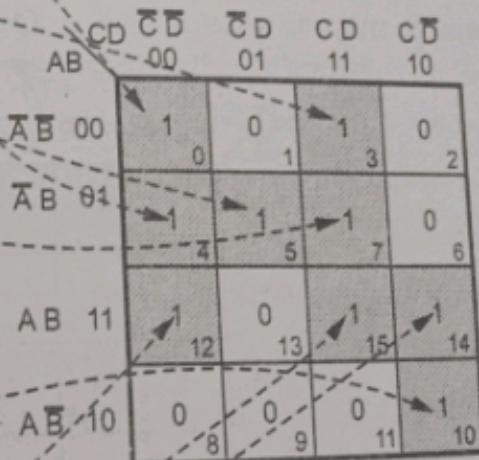


Fig. Q.6.1 (b) Representation of 3-variable truth table on K-map

No.	A	B	C	D	Y
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1



(c) Representation of 4-variable truth table on K-map

Fig. Q.6.1 Plotting truth table on K-map

Q.7 Plot Boolean expression $Y = A\bar{B}\bar{C} + A\bar{B}C + \bar{A}\bar{B}C$ on the Karnaugh map.

Ans. : The expression has 3-variables and hence it can be plotted using 3-variable as in Fig. Q.7.1.

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

Fig. Q.7.1

Q.8 Plot Boolean expression.

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

Ans. : The expression has 4-variables and hence it can be plotted using 4-variable map as shown in Fig. Q.8.1.

		$A + B + C + \bar{D}$		$A + B + \bar{C} + \bar{D}$			
		$C\bar{D}$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$		
		$A+B$	1	0	0	1	
		$A+\bar{B}$	1	1	1	0	$\cdots A + \bar{B} + \bar{C} + D$
		$\bar{A}+\bar{B}$	4	5	7	6	$\cdots \bar{A} + \bar{B} + \bar{C} + D$
		$\bar{A}+B$	1	0	1	0	
			12	13	15	14	
			8	9	11	10	$\bar{A} + \bar{B} + C + \bar{D}$

Fig. Q.8.1

$$(A + B + C + \bar{D}) = M_1, (A + \bar{B} + \bar{C} + D) = M_6, (A + B + \bar{C} + \bar{D}) = M_3, \\ (\bar{A} + \bar{B} + C + \bar{D}) = M_{13}, (\bar{A} + \bar{B} + \bar{C} + D) = M_{14}$$

Q.9 What is a pair in K-map? How is it useful in simplification of Boolean expression?

Ans. • A pair is a group of two adjacent cells in a Karnaugh map. It cancels one variable in a K-map simplification.

• Fig. Q.9.1 (a) shows a pair of 1's that are horizontally adjacent to each other; the first represents $\bar{A}\bar{B}C$ and the second represents $\bar{A}BC$. Note that in these two terms only the B variable appears in both normal and complemented form (\bar{A} and C remain unchanged). Thus the expression reduces to $\bar{A}C$.

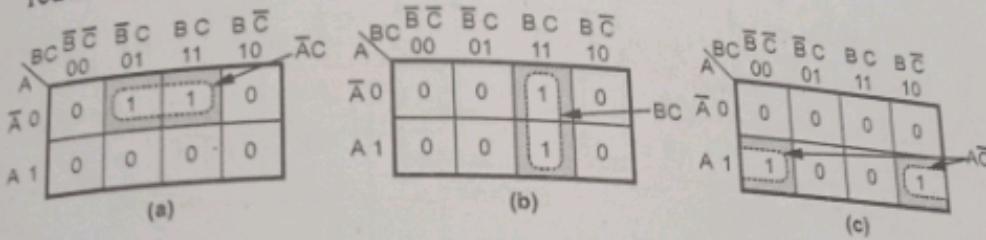


Fig. Q.9.1

- Fig. Q.9.1 (b) shows an example of two vertically adjacent 1s. These two can be combined to eliminate A variable since it appears in both its uncomplemented and complemented forms. Thus the expression reduces to BC.
- In a Karnaugh map the corresponding cells in the leftmost column and rightmost column are considered to be adjacent. Thus, the two 1s in these columns with a common row can be combined to eliminate one variable. This is illustrated in Fig. Q.9.1 (c).
- Here variable B has appeared in both its complemented and uncomplemented forms and hence eliminated to give expression as $\bar{A}C$.
- Same pairing rules can be applied for 2 or 4 variable K-maps. In 4-variable K-maps, the corresponding cells in the top row and bottom row are considered to be adjacent.

Q.10 What is quad in K-map? How is it useful in simplification of Boolean expression?

Ans. • Quad is a group of four adjacent cells in a Karnaugh map. It cancels two variables in a K-map simplification.

• Fig. Q.10.1 shows several examples of quad

	BC $\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
A	00	01	11	10
\bar{A}	0	0	0	0
A	1	1	1	1

(a) $Y = A$

	CD $\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
AB	00	01	11	10
$\bar{A}\bar{B}$	00	01	11	10
AB	00	01	11	10
$A\bar{B}$	00	01	11	10

(c) $Y = BD$

	CD $\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
AB	00	01	11	10
$\bar{A}\bar{B}$	00	01	11	10
AB	00	01	11	10
$A\bar{B}$	00	01	11	10

(d) $Y = A\bar{D}$

	CD $\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
AB	00	01	11	10
$\bar{A}\bar{B}$	00	01	11	10
AB	00	01	11	10
$A\bar{B}$	00	01	11	10

(e) $Y = \bar{BD}$

Fig. Q.10.1

Q.11 What is octet ? How is it useful in simplification of Boolean expression ?

Ans. : Octet is a group of eight adjacent cells in a Karnaugh map. It cancels three variables in a K-map simplifications.

Fig. Q.11.1 shows several examples of octet. (Refer Fig. Q.11.1 on next page)

Q.12 State the rules for simplifying logic function using K-map.

Ans. : Rules for simplifying logic function using K-map are :

1. Group should not include any cell containing a zero.
2. The number of cells in a group must be a power of 2, such as 1, 2, 4, 8 or 16.
3. Group may be horizontal, vertical but not diagonal.
4. Cell containing 1 must be included in at least one group.
5. Groups may overlap.
6. Each group should be as large as possible to get maximum simplification.
7. Groups may be wrapped around the map. The leftmost cell in a row may be grouped with the rightmost cell and the top cell in a column may be grouped with the bottom cell.

AB	CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	00	01	11	10	
$\bar{A}\bar{B}$	00	0	0	0	0
$\bar{A}\bar{B}$	01	1	1	1	1
$\bar{A}\bar{B}$	11	1	1	1	1
$\bar{A}\bar{B}$	10	0	0	0	0

(a) $Y = B$

AB	CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	00	01	11	10	
$\bar{A}\bar{B}$	00	0	1	1	0
$\bar{A}\bar{B}$	01	0	1	1	0
$\bar{A}\bar{B}$	11	0	1	1	0
$\bar{A}\bar{B}$	10	0	1	1	0

(b) $Y = D$

AB	CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	00	01	11	10	
$\bar{A}\bar{B}$	00	1	1	1	1
$\bar{A}\bar{B}$	01	0	0	0	0
$\bar{A}\bar{B}$	11	0	0	0	0
$\bar{A}\bar{B}$	10	1	1	1	1

(c) $Y = \bar{B}$

AB	CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	00	01	11	10	
$\bar{A}\bar{B}$	00	1	0	0	1
$\bar{A}\bar{B}$	01	1	0	0	1
$\bar{A}\bar{B}$	11	1	0	0	1
$\bar{A}\bar{B}$	10	1	0	0	1

(d) $Y = \bar{D}$

Fig. Q.11.1 Examples of combining octets of adjacent ones

8. A cell may be grouped more than once. The only condition is that every group must have at least one cell that does not belong to any other group. Otherwise, redundant terms will result.
9. We need not group all don't care cells, only those that actually contribute to a maximum simplification.
10. All above rules are stated considering the SOP simplification. In case of POS simplification all rules are same except 0 (zero) takes place of 1 (one).

Q.13 What do you mean by don't care conditions ?

Ans. :

- In some logic circuits, certain input conditions never occur, therefore the corresponding output never appears. In such cases the output level is not defined, it can be either HIGH or LOW. These output levels are indicated by 'X' or 'd' in the truth tables and are called don't care outputs or don't care conditions or incompletely specified functions.

- A circuit designer is free to make the output for any "don't care" condition either a '0' or a '1' in order to produce the simplest output expression.

Q.14 Simplify the following function

$$f_1(A, B, C, D) = \sum m(0, 3, 5, 6, 9, 10, 12, 15)$$

[SPPU : May-13, Marks 4]

Ans. :

$$\begin{aligned} f_1 &= \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} \\ &\quad + A\bar{B}\bar{C}\bar{D} + ABCD + A\bar{B}\bar{C}D + A\bar{B}CD \\ &= \bar{A}\bar{B}(\bar{C}\bar{D} + CD) + \bar{A}B(\bar{C}D + C\bar{D}) \\ &\quad + AB(\bar{C}\bar{D} + CD) + A\bar{B}(\bar{C}D + C\bar{D}) \\ &= \bar{A}\bar{B}(C \oplus D) + \bar{A}B(C \oplus D) \\ &\quad + AB(C \oplus D) + A\bar{B}(C \oplus D) \\ &= (\bar{A}\bar{B} + AB)(C \oplus D) + (\bar{A}B + A\bar{B})(C \oplus D) \\ &= (A \oplus B)(C \oplus D) + (A \oplus B)(C \oplus D) \end{aligned}$$

		CD	00	01	11	10
AB		00	1		1	
		01		1		1
		11	1		1	
		10		1		1

Fig. Q.14.1

Substituting $A \oplus B = A$ and $C \oplus D = B$ we have

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

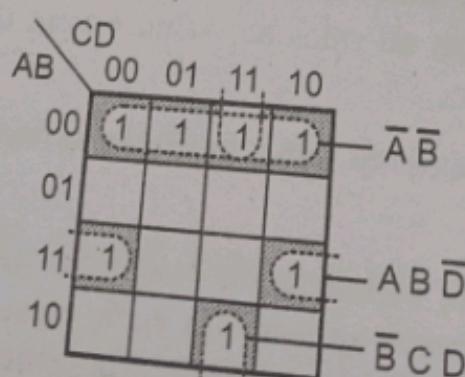
$$\therefore f_1 = (A \oplus B) \odot (C \oplus D)$$

Q.15 Simplify the following function

$$f_3(A, B, C, D) = \sum m(0, 1, 2, 3, 11, 12, 14)$$

[SPPU : May-13, Marks 4]

Ans. :



$$\therefore f_3 = \bar{A}\bar{B} + AB\bar{D} + B\bar{C}D$$

Fig. Q.15.1

minimization
it care
it output

Q.16 Solve the following using minimization technique
 $Z = f(A, B, C, D) = \sum (0, 2, 4, 7, 11, 13, 15)$

[SPPU : Dec.-09, Marks 5]

Ans. :

		CD	00	01	11	10	
		AB	00	01	11	10	
			1			1	$\bar{A}\bar{B}\bar{D}$
			1		1		$\bar{A}C\bar{D}$
				1	1		$B\bar{C}D$
						1	$\bar{A}BD$
						1	$A\bar{C}D$

$$\therefore Z = \bar{A}\bar{B}\bar{D} + \bar{A}\bar{C}\bar{D} + B\bar{C}D + \bar{A}BD + A\bar{C}D$$

Fig. Q.16.1

Q.17 Using K-map convert the following standard POS expression into a minimum POS expression, a standard SOP expression and minimum SOP expression

$$(A' + B' + C + D) (A + B' + C + D) (A + B + C + D')$$

$$(A + B + C' + D') (A' + B + C + D') (A + B + C' + D).$$

[SPPU : Dec.-15, Marks 6]

Ans. : $(\bar{A} + \bar{B} + C + D) (A + \bar{B} + C + D) (A + B + C + \bar{D}) (A + B + \bar{C} + \bar{D})$
 $(\bar{A} + B + C + \bar{D}) (A + B + \bar{C} + D) = \pi M(12, 4, 1, 3, 9, 2)$
 $= \pi M(1, 2, 3, 4, 9, 12)$

Minimum POS Expression using K-map

		CD	C+D	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$	
		AB	00	01	11	10	
			0	0	0	0	$(A + B + \bar{C})$
			0				$(\bar{B} + C + D)$
			0				
				0			$(B + C + \bar{D})$

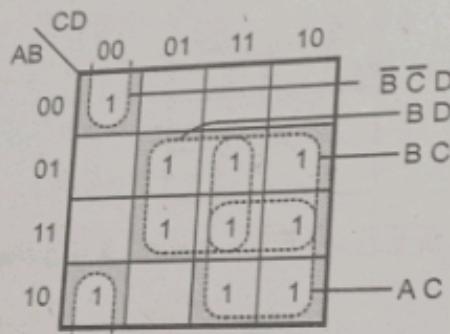
$$\therefore F(A, B, C, D) = (A + B + \bar{C})(\bar{B} + C + D)(B + C + \bar{D})$$

Ans. :
K-map simi

Standard SOP Expression

$$\begin{aligned}\pi M(1, 2, 3, 4, 9, 12) &= \Sigma m(0, 5, 6, 7, 8, 10, 11, 13, 14, 15) \\ &= \overline{AB}\overline{CD} + \overline{AB}\overline{C}D + \overline{ABC}\overline{D} + \overline{ABC}D + A\overline{B}\overline{C}\overline{D} \\ &\quad + A\overline{B}\overline{C}D + A\overline{B}CD + AB\overline{C}\overline{D} + ABC\overline{D} + ABCD\end{aligned}$$

Minimum SOP Expression



$$\therefore F(A, B, C, D) = \overline{B}\overline{C}D + BD + BC + AC$$

Q.18 Minimize the following function using K-map and implement using basic logic gates $f(A, B, C, D) = \sum m(0, 1, 2, 4, 8, 9, 12, 13) + d(3, 6, 7)$.

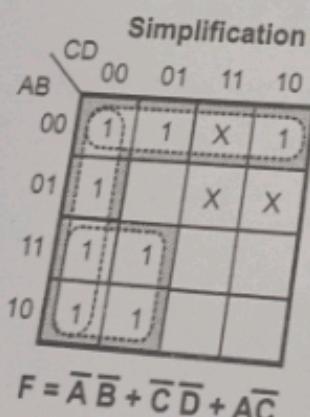
Ans. :

[SPPU : May-14, Marks 6]

Q.20 Min
Impleme

Ans. :

K-map s



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

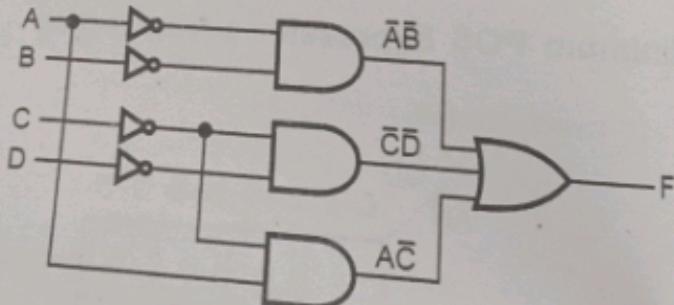
Implementation

Fig. Q.18.1

Q.19 Minimize the following equation using K-map and realize it using NAND gates only.

$$y = \sum m(0, 1, 2, 3, 5, 7, 8, 9, 11, 14)$$

[SPPU : Dec.-11, Marks 10]

f (A, B)

DECODE®

Ans. :
K-map simplification

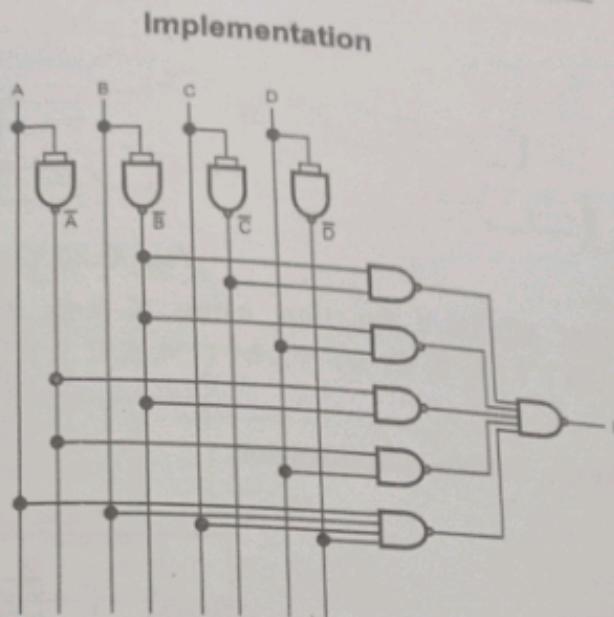
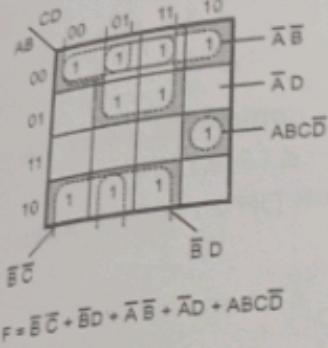


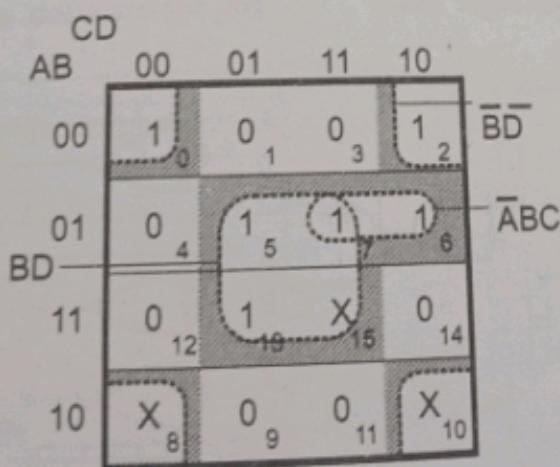
Fig. Q.19.1

Q.20 Minimize $f(A, B, C, D) = \sum m(0, 2, 5, 6, 7, 13) + d(8, 10, 15)$
Implement using NAND gates.

[SPPU : May-07, Marks 8]

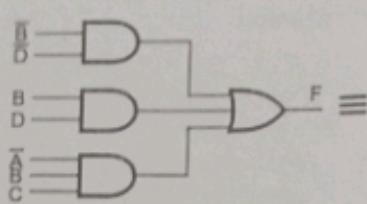
Ans. :

K-map simplification :



$$f(A, B, C, D) = \overline{B} \overline{D} + B D + \overline{A} B C$$

Implementation :



Implementation using NAND gates :

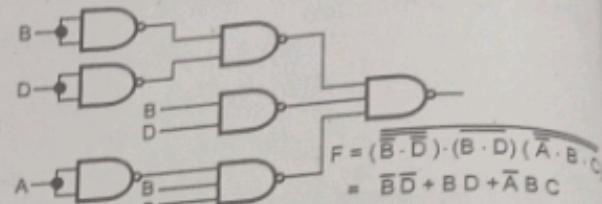


Fig. Q.20.1

Q.21 Minimize function using K-map and implement using NOR gate : $f(A, B, C, D) = \pi M(1, 4, 5, 6, 7, 8, 12) + d(3, 9, 11, 14)$
 [SPPU : Dec.-07, Marks 6]

Ans. : K-map simplification :

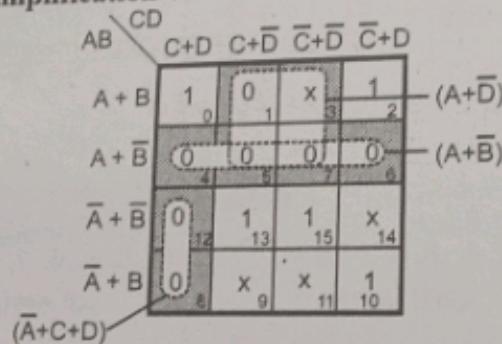
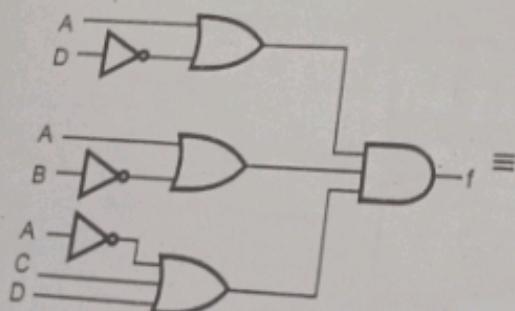


Fig. Q.21.1

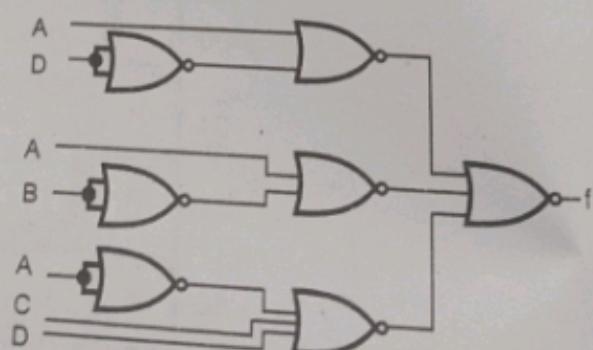
$$\therefore f(ABCD) = (A + \overline{D})(A + \overline{B})(\overline{A} + C + D)$$

Implementation using basic gates



(a)

Implementation using only NOR gates



$$(b)$$

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

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

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

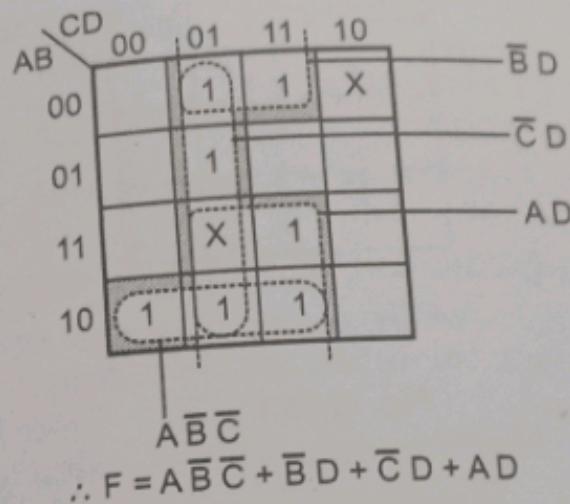
Fig. Q.21.2

Q.22 Minimize the following function using K-map and implement using basic logic gates.

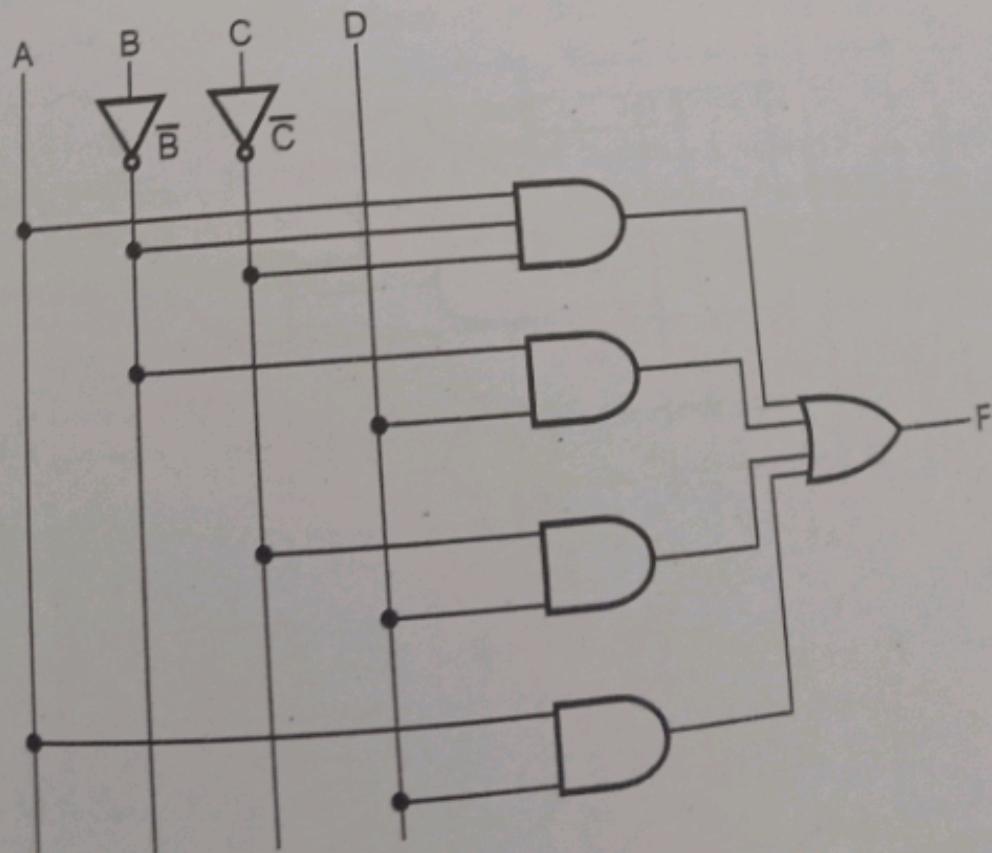
$$F(A, B, C, D) = \sum m(1, 3, 5, 8, 9, 11, 15) + d(2, 13).$$

U.G.P [SPPU : Dec.-17, Marks 6]

Ans. : K-map Simplification :



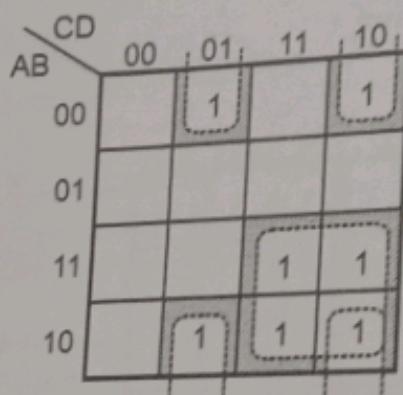
Implementation :



Q.23 Use K-map minimization technique to realize following expression using minimum number of gates.

$$Y = \sum m(1, 2, 9, 10, 11, 14, 15). \quad [SPPU : May-19, Marks 6]$$

Ans. :



$$Y = \overline{B} \overline{C} D + \overline{B} C \overline{D} + A C$$

Fig. Q.23.1

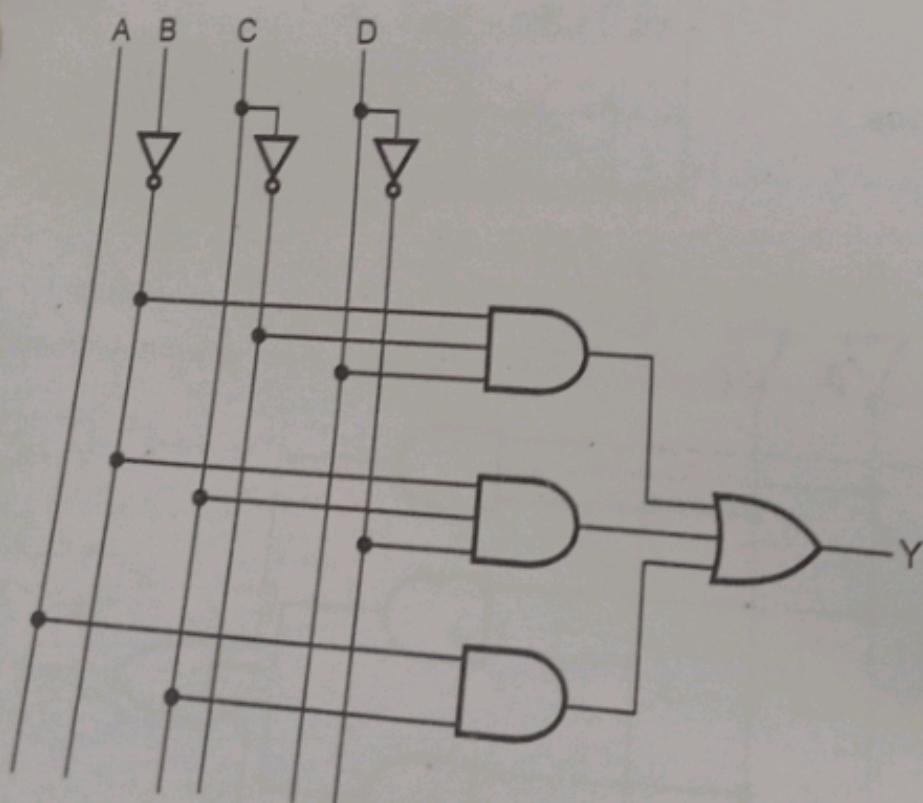


Fig. Q.23.2

4

Q.1 State the

Ans. : Step
input code a
Step 2 : For
expression i
Step 3 : Re

Q.2 Design

Ans. : Ste
Excess-3 c
can be de
number. F
0010. Now
in decimal
code conv

END... ↗