JEPPIAAR INSTITUTE OF TECHNOLOGY

Self Belief | Self Discipline | Self Respect

QUESTION BANK

REGULATION :2017

YEAR : II

SEMESTER : 03

BATCH :2018-2022

DEPARTMENT

OF

INFORMATION TECHNOLOGY

JEPPIAAR INSTITUTE OF TECHNOLOGY

"Self-Belief | Self Discipline | Self Respect"

INSTITUTION VISION

Jeppiaar Institute of Technology aspires to provide technical education in futuristic technologies with the perspective of innovative, industrial and social application for the betterment of humanity.

INSTITUTION MISSION

- To produce competent and disciplined high quality professionals with the practical skills necessary to excel as innovative professionals and entrepreneurs for the benefit of the society.
- To improve the quality of education through excellence in teaching and learning, research, leadership and by promoting the principles of scientific analysis, and creative thinking.
- To provide excellent infrastructure, serene and stimulating environment that is most conducive to learning.
- To strive for productive partnership between the Industry and the Institute for research and development in the emerging fields and creating opportunities for employability.
- To serve the global community by instilling ethics, values and life skills among the students needed to enrich their lives.

JEPPIAAR INSTITUTE OF TECHNOLOGY

"Self-Belief | Self Discipline | Self Respect"

DEPARTMENT VISION

To facilitate the evolution of problem solving skills along with knowledge application in the field of Information Technology, understanding industrial and global requirements for the benefit of the society.

DEPARTMENT MISSION

- To produce creative and productive computing graduates in software development being aware of global requirements and maximize employability.
- To enhance evolution of professional skills and development of leadership traits among the students to grow into successful entrepreneurs.
- To offer students an advantageous infrastructure to apply their research thoughts and develop their technical expertise .
- To escalate the moral code and honesty in the professional activities.

Program Educational Objectives (PEOs)

PEO 1: To provide students with a fundamental knowledge in Science, mathematics and computing skills for creative and innovative application.

PEO 2: To enable students competent and employable by providing excellent Infrastructure to learn and contribute for the welfare of the society.

PEO 3: To channelize the potentials of the students by offering state of the art amenities to undergo research and higher education.

PEO 4: To evolve computing engineers with multi-disciplinary understanding and maximize Job Opportunities.

PEO 5: To facilitate students obtain profound understanding nature and social requirements and grow as professionals with values and integrity.

Program Specific Outcomes (PSOs)

PSO 1 : To create the ability to analyze and enhance coding skills by participating in various competitions.

PSO 2 : Students are able to provide solutions for Social Problems by creating Mobile Application Development using Android Studio and Chatbot.

PSO 3 : Students are able to deal with real time problems using Machine Learning Tools and Big data Analytics.

BLOOM'S TAXONOMY

Definition:

Bloom's taxonomy is a classification system used to define and distinguish different levels of human cognition like thinking, learning, and understanding.

Objectives:

- To classify educational learning objectives into levels of complexity and specificity. The classification covers the learning objectives in cognitive, affective and sensory domains.
- To structure curriculum learning objectives, assessments and activities.

Levels in Bloom's Taxonomy:

- **BTL 1 – Remember** - The learner recalls, restate and remember the learned information.
- **BTL 2 – Understand** - The learner embraces the meaning of the information by interpreting and translating what has been learned.
- **BTL3–Apply**-The learner makes use of the information in a context similar to the one in which it was learned.
- **BTL 4 – Analyze** - The learner breaks the learned information into its parts to understand the information better.
- **BTL 5 – Evaluate** - The learner makes decisions based on in-depth reflection, criticism and assessment.
- **BTL6–Create**-The learner creates new ideas and information using what has been previously learned.

TABLE OF CONTENTS

TABLE OF CONTENT

MA8353	DISCRETE MATHEMATICS	L	T	P	C
		4	0	0	4

OBJECTIVES:

- The primary objective of this course is to provide mathematical background and sufficient experience on various topics of discrete mathematics like logic and proofs, combinatorics, graphs, algebraic structures, lattices and Boolean algebra.
- This course will extend student‟s Logical and Mathematical maturity and ability to deal with abstraction and to introduce most of the basic terminologies used in computer science courses and application of ideas to solve practical problems.

UNIT I LOGICANDPROOFS 12

Propositional logic – Propositional equivalences - Predicates and quantifiers – Nested quantifiers – Rules of inference - Introduction to proofs – Proof methods and strategy.

UNIT II COMBINATORICS 12

Mathematical induction – Strong induction and well ordering – The basics of counting – The pigeonhole principle – Permutations and combinations – Recurrence relations – Solving linear recurrence relations – Generating functions – Inclusion and exclusion principle and its applications

UNIT III GRAPHS 12

Graphs and graph models – Graph terminology and special types of graphs – Matrix representation of graphs and graph isomorphism – Connectivity – Euler and Hamilton paths.

UNIT IV ALGEBRAIC STRUCTURES 12

Algebraic systems – Semi groups and monoids - Groups – Subgroups – Homomorphism‟s – Normal subgroup and cosets – Lagrange‟s theorem – Definitions and examples of Rings and Fields.

UNIT V LATTICES AND BOOLEAN ALGEBRA 12

Partial ordering – Posets – Lattices as Posets – Properties of lattices - Lattices as algebraic systems – Sub lattices – Direct product and homomorphism – Some special lattices – Boolean algebra.

TOTAL PERIODS: 60

OUTCOMES:

After completing this course, students should demonstrate competency in the following topics:

- Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations, functions, and integers.

- Evaluate elementary mathematical arguments and identify fallacious reasoning (not just fallacious conclusions).
- Synthesize induction hypotheses and simple induction proofs.
- Prove elementary properties of modular arithmetic and explain their applications in Computer Science, for example, in cryptography and hashing algorithms.
- Apply graph theory models of data structures and state machines to solve problems of connectivity and constraint satisfaction, for example, scheduling.
- Apply the method of invariants and well-founded ordering to prove correctness and termination of processes and state machines.
- Derive closed-form and asymptotic expressions from series and recurrences for growth rates of processes.
- Calculate numbers of possible outcomes of elementary combinatorial processes such as permutations and combinations.
- Concepts and properties of the algebraic structures such as groups, rings and fields and lattices and Boolean Algebra

TEXTBOOKS:

1. Rosen, K.H., "Discrete Mathematics and its Applications", 7th Edition, Tata McGraw Hill Pub. Co. Ltd., New Delhi, Special Indian Edition, 2011.
2. Tremblay, J.P. and Manohar.R, " Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw Hill Pub. Co. Ltd, New Delhi, 30th Reprint, 2011.

REFERENCES:

1. Grimaldi, R.P. "Discrete and Combinatorial Mathematics: An Applied Introduction", 4th Edition, Pearson Education Asia, Delhi, 2007.
2. Lipschutz, S. and Mark Lipson., "Discrete Mathematics", Schaum‟s Outlines, Tata McGraw Hill Pub. Co. Ltd., New Delhi, 3rd Edition, 2010.
3. Koshy, T. "Discrete Mathematics with Applications", Elsevier Publications, 2006.

MA8353 – Discrete Mathematics

UNIT I –LOGICS AND PROOFS
Study of Propositional logic – Propositional equivalences - Predicates and quantifiers – Nested quantifiers – Rules of inference - Introduction to proofs – Proof methods and strategy.

PART A

Q.No.	Questions
1.	**Define Proposition. (BTL1)** A proposition or a statement is a declarative sentence or assertion that is either true or false, but not both. **Example :"**6>7" (false) is a proposition **"**The sun sets in the east" (true) is a proposition
2	**Define tautology and contradiction.(BTL1)** A statement formula which is always true irrespective of the truth values of the individual variables is called a tautology. **Example:** $p \vee \neg p$ is a tautology. A statement formula which is always false is called contradiction or absurdity. **Example:** $p \wedge \neg p$ is a contradiction.
3	**Define atomic and compound statements.(BTL1)** A proposition or statement is atomic if it cannot be broken into simple propositions. A proposition obtained by combining two or more propositions by means of logical connectives is

	called a compound proposition or statement.
4	**Write the symbolic representation for "Students can access the internet from the campus only if they are computer science students or only if they are not fresher"s". (BTL3)** P: Students can access the internet from the campus Q: They are computer science students R: They are not fresher"s The symbolic representation is $P \rightarrow (Q \vee \neg R)$
5	**Give the converse, contra positive, and inverse of the statement "If there is rain , then I buy an umbrella". Also give its symbolic representation. (BTL3)** Let p: There is rain q: I buy an umbrella The given statement is $p \rightarrow q$ **CONTRAPOSITIVE:** $\neg q \rightarrow \neg p$ " If I do not buy an umbrella then there is no rain" **CONVERSE :** $q \rightarrow p$ "If I buy an umbrella then there is rain" **INVERSE:** $\neg p \rightarrow \neg q$ "If there is no rain then I do not buy an umbrella".
6	**Write down the converse, contra positive and inverse of the conditional statement " The home team wins whenever it is raining". (BTL3)** Let p: It is raining q: Home team wins The given statement is $p \rightarrow q$

CONTRAPOSITIVE: $\neg q \rightarrow \neg p$

"If the home team does not win, then it is not raining"

CONVERSE : $q \rightarrow p$

"If it is raining then the home team wins"

INVERSE: $\neg p \rightarrow \neg q$

"If it is not raining then the home team does not win"

7 **When do you say that two compound propositions are equivalent? (BTL2)**

Two propositions P and Q are equivalent iff $P \leftrightarrow Q$ is a tautology. It is denoted by the symbol $P \Leftrightarrow Q$

8 **Find the truth value of** $p \rightarrow \neg q$ **. (BTL2)**

P	Q	$\neg q$	$p \rightarrow \neg q$
T	T	F	F
T	F	T	T
F	T	F	T
F	F	T	T

9 **Construct a truth table for the compound proposition** $(p \rightarrow q) \rightarrow (q \rightarrow p)$ **BTL3**

P	Q	$p \rightarrow q$	$q \rightarrow p$	$(p \rightarrow q) \rightarrow (q \rightarrow p)$
T	T	T	T	T
T	F	F	T	T
F	T	T	F	F
F	F	T	T	T

10 **Construct a truth table for the compound proposition** $(p \rightarrow q) \leftrightarrow (\neg p \rightarrow \neg q)$ **(BTL3)**

		P	q	$\neg p$	$\neg q$	$p \rightarrow q$	$(\neg p \rightarrow \neg q)$	$(p \rightarrow q) \leftrightarrow (\neg p \rightarrow \neg q)$
		T	T	F	F	T	T	T
		T	F	F	T	F	T	F
		F	T	T	F	T	F	F
		F	F	T	T	T	T	T

11 **Using truth table, show that the proposition $p \vee \neg(p \wedge q)$ is a tautology. (BTL3)**

P	Q	$p \wedge q$	$\neg(p \wedge q)$	$p \vee \neg(p \wedge q)$
T	T	T	F	T
T	F	F	T	T
F	T	F	T	T
F	F	F	T	T

12 **Show that $(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R)$ is a tautology.** **(BTL3)**

Let S= $(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R)$

P	Q	R	$Q \rightarrow R$	$P \rightarrow Q$	$P \rightarrow R$	$P \rightarrow (Q \rightarrow R)$	$(P \rightarrow Q) \rightarrow (P \rightarrow R)$	S
T	T	T	T	T	T	T	T	T
T	T	F	F	T	F	F	F	T
T	F	T	T	F	T	T	T	T
T	F	F	T	F	F	T	T	T
F	T	T	T	T	T	T	T	T
F	T	F	F	T	T	T	T	T
F	F	T	T	T	T	T	T	T
F	F	F	T	T	T	T	T	T

Since all the entries in the resulting column is true, the given proposition is a tautology.

13 **Give the truth value of $T \leftrightarrow T \wedge F$** **(BTL1)**

$T \leftrightarrow T \wedge F$
$\Leftrightarrow T \leftrightarrow F$
$\Leftrightarrow F$

14 **Show that** $(p \rightarrow q) \wedge (r \rightarrow q)$ **and** $(p \vee r) \rightarrow q$ **are logically equivalent. (BTL 5)**

P	q	R	$p \rightarrow q$	$r \rightarrow q$	$p \vee r$	$(p \rightarrow q) \wedge (r \rightarrow q)$	$(p \vee r) \rightarrow q$
T	T	T	T	T	T	T	T
T	T	F	T	T	T	T	T
T	F	T	F	F	T	F	F
T	F	F	F	T	T	F	F
F	T	T	T	T	T	T	T
F	T	F	T	T	F	T	T
F	F	T	T	F	T	F	F
F	F	F	T	T	F	T	T

The truth values are same in the give two statements.
Therefore the statements are logically equivalent.

15 **Using truth table show that** $p \vee (p \wedge q) \equiv p$ **(BTL5)**

P	Q	$p \wedge q$	$p \vee (p \wedge q)$
T	T	T	T
T	F	F	T
F	T	F	F
F	F	F	F

The truth values of p and $p \vee (p \wedge q)$ are same. Therefore the statements are logically equivalent .
That is $p \vee (p \wedge q) \equiv p$.

16 **Express** $A \leftrightarrow B$ **in terms of the connectives {** $\wedge, \neg$ **}. (BTL1)**

	The biconditional law is $A \leftrightarrow B \Leftrightarrow (A \wedge B) \vee (\neg A \wedge \neg B)$
17	**Without using truth table show that $p \rightarrow (q \rightarrow p) \Leftrightarrow \neg p \rightarrow (p \rightarrow \neg q)$. (BTL3)** L.H.S $\Leftrightarrow p \rightarrow (q \rightarrow p)$ $\Leftrightarrow \neg p \vee (q \rightarrow p)$ — Implication law $\Leftrightarrow \neg p \vee (\neg q \vee p)$ — Implication law $\Leftrightarrow \neg p \vee (p \vee \neg q)$ — commutative law $\Leftrightarrow (p \vee \neg p) \vee \neg q$ — Associative and commutative $\Leftrightarrow p \vee (\neg p \vee \neg q)$ — Associative law $\Leftrightarrow \neg p \rightarrow (\neg p \vee \neg q)$ — Implication law $\Leftrightarrow \neg p \rightarrow (p \rightarrow \neg q)$ — Implication and double negation law $\Leftrightarrow$ R.H.S
18	**Define rule of universal specification . (BTL1)** Universal specification or instantiation is the rule of inference which says that we conclude P(C) is true for a particular element C of the discourse if $\forall x\, P(x)$ is true.
19	**Give the symbolic form of "some men are giants" (BTL4)** P(x) : x is a man Q(x) : x is a gaint Symbolic form: $\forall x(P(x) \rightarrow Q(x))$
20	**What are the negations of the statements $\forall x(x^2 > x)$ and $\exists x(x^2 = 2)$? (BTL3)** Let P(x) : $(x^2 > x)$ $\neg$ P(x) : $(x^2 \leq x)$ **Given:** $\forall x(x^2 > x)$ Its negation is $\neg[\forall x(x^2 > x)] \Leftrightarrow \exists x(x^2 \leq x)$ Let P(x) : $(x^2 = 2)$

	$\neg$ P(x) : $\left(x^2 \neq 2\right)$ **Given:** $\exists x\left(x^2 = 2\right)$ Its negation is $\forall x \neg\left(x^2 = 2\right) \quad \Leftrightarrow \quad \forall x\left(x^2 \neq 2\right)$
21.	**Write the negation of the statement** $(\exists x)(\forall y)\, p(x, y)$**. (BTL2)** **Given :** $(\exists x)(\forall y)\, p(x, y)$. Its negation is $\neg[(\exists x)(\forall y)\, p(x, y)] \Leftrightarrow \quad (\forall x)(\exists y)\, p(x, y)$.
22.	**Given P={2,3,4,5} , state the truth value of the statement** $(\exists x \in P)\left(x + 3 = 10\right)$**. (BTL1)** The maximum value in P is 6 (6+3=9) There is no such „x" in P such that x+3=10 Therefore the truth value of the statement is FALSE
23.	**Find the truth value of** $\forall x\left(x^2 \geq x\right)$ **if the universe of discourse consists of all real numbers and what is its truth value if the universe of discourse consists of all integers? (BTL4)** **Given :** $\left(x^2 \geq x\right)$ $\Leftrightarrow x^2 - x = x(x-1) \geq 0$ Consequently $\left(x^2 \geq x\right)$ if and only if $x \leq 0$ or $x \geq 1$ The inequality is false for all real numbers x with 0<x<1 (For example if x=1/2 then x^2 = ¼ which is less than x) Therefore $\forall x\left(x^2 \geq x\right)$ is false if the universe of discourse consists of all real numbers. However if the universe of discourse consists of the integers, There are no integers x with 0<x<1 Therefore $\forall x\left(x^2 \geq x\right)$ is true if the universe of discourse consists of all integers
24.	**Let P(x) denote the statement x ≤ 4. Write the truth values of P(2) and P(6) . (BTL2)** P(x) : x ≤ 4. When x=2 , P(2): 2 ≤ 4, which is true When x=6 , P(6): 6 ≤ 4, which is false
25.	**Give an indirect proof of the theorem " If 3n+2 is odd, then n is odd". (BTL2)** **To Prove:** 3n+2 is odd → n is odd

	In indirect method, assume that the conclusion is false and come to a contradiction. That is assume that n is even. Let n=2k, where k is any integer. Then 3n+2 = 3(2k) + 2 = 6k+2 = 2(3k+1) Therefore 3n+2 is even, which contradicts the hypothesis 3n+2 is odd. Hence the assumption is wrong. Therefore n is odd and hence the given implication is true.
	PART * B
1	**Show that** $((p \vee q) \wedge \neg(\neg p \wedge (\neg q \vee \neg r))) \vee (\neg p \wedge \neg q) \vee (\neg p \wedge \neg r)$ **is a tautology. (Nov 2013, Apr 2015, Apr2017). (BTL5) (8 Marks)** (Refer Balaji Pg. 1.49) Keypoints: • $(\neg p \wedge \neg q) \vee (\neg p \wedge \neg r) \Leftrightarrow \neg((p \vee q) \wedge (p \vee r))$ (3marks) • $(\neg p \wedge (\neg q \vee \neg r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$ (3marks) • Get the answer as T (2marks)
2	**Show that** $(\neg p \wedge (\neg q \wedge r)) \vee (q \wedge r) \vee (p \wedge r) \Leftrightarrow r$ **without using truth table. (Nov2016, Apr 2018) . (BTL5) (8 Marks)** (Refer Balaji Pg. 1.44) Keypoints: • $(\neg p \wedge (\neg q \wedge r)) \Leftrightarrow \neg(p \vee q) \wedge r$ (2marks) • $(q \wedge r) \vee (p \wedge r) \Leftrightarrow (p \vee q) \wedge r$ (2marks) • $T \vee r$ (2marks) • Get the answer as r (2marks)
3	**Prove the conditional statement** $[(P \to Q) \wedge (Q \to R)] \to (P \to R)$ **is a tautology using logical equivalences. (Nov 2017). (BTL5) (8 Marks)** (Refer Balaji Pg. 1.49) Keypoints: • $[(P \to Q) \wedge (Q \to R)] \Leftrightarrow (P \to R)$ (3marks) • $P \vee \neg p \Leftrightarrow T$ (3marks) • Get the answer as T (2marks)
4	**Show that** $R \vee S$ **is a valid conclusion from the premises**

	$C \vee D, C \vee D \rightarrow \neg H, \neg H \rightarrow (A \wedge \neg B), (A \wedge \neg B) \rightarrow (R \vee S)$ **(BTL5) (8 Marks)** (Refer SKD, Pg.1.69) Keypoints: • $C \vee D \rightarrow H$ (2marks) • $C \vee D \rightarrow (A \wedge \neg B)$ (2marks) • $C \vee D \rightarrow (R \vee S)$ (2marks) • Get the answer as $R \vee S$ (2marks)
5	**Show that the premises $P \rightarrow Q, Q \rightarrow R, R \rightarrow S, S \rightarrow \neg R$ and $P \wedge S$ are inconsistent. (Nov2015). (BTL5) (8 Marks)** (Refer SKD Pg. 1.81) Keypoints: • $P \rightarrow R$ (2marks) • $R \rightarrow \neg S$ (2marks) • $(Q \wedge S) \wedge \neg(Q \wedge S)$ (2marks) • To prove inconsistency, derive a contradiction ((i.e.) Answer is F) (2marks)
6	**Using CP rule show that , $\neg P \vee Q$, $\neg Q \vee R$, $R \vee S \Rightarrow P \rightarrow S$. (Apr 2018) (BTL5) (8 Marks)** (Refer Classwork) Keypoints: • $\neg P \vee Q \Leftrightarrow P \rightarrow Q$ (2marks) • S is the additional premise (2 marks) • $\neg Q \vee R \Leftrightarrow Q \rightarrow R$ (2marks) • Get the answer as S (2marks)
7	**Obtain the PDNF AND PCNF of $(\neg P \rightarrow R) \wedge (Q \leftrightarrow P)$ by using equivalences. (Apr2017, May2016,, Nov2015). (BTL4) (8 Marks)** (Refer Balaji Pg. 1.83) Keypoints: • $(\neg P \rightarrow R) \wedge (Q \leftrightarrow P) \Leftrightarrow (P \vee R) \wedge [(\neg Q \vee P) \wedge (\neg P \vee Q)]$ (2marks) • $[(P \vee R) \vee F] \wedge [(\neg Q \vee P) \vee F] \wedge [(\neg P \vee Q) \vee F]$ (2marks) • $(P \vee Q \vee R) \wedge (P \vee \neg Q \vee R) \wedge (P \vee \neg Q \vee \neg R) \wedge (\neg P \vee Q \vee R) \wedge (\neg P \vee Q \vee \neg R)$ (2marks)

	• $\neg(\neg S)$ to obtain PCNF (2marks)
8	**Obtain the PDNF AND PCNF of** $(P \wedge Q) \vee (\neg P \wedge R)$ **. (Nov2016) (BTL4) (8 Marks)** (Refer SKD Pg. 1.45) Keypoints: • $((P \wedge Q) \vee \neg P) \wedge ((P \wedge Q) \vee R)$ (2marks) • $(Q \vee \neg P \vee F) \wedge (P \vee R \vee F) \wedge (Q \vee R \vee F)$ (2marks) • $(\neg P \vee Q \vee R) \wedge (\neg P \vee Q \vee \neg R) \wedge (P \vee Q \vee R) \wedge (P \vee \neg Q \vee R)$(2marks) • $\neg(\neg S)$ to obtain PDNF (2marks)
9	**Show that the hypothesis " It is not sunny this afternoon and it is colder than yesterday", " we will go swimming only if its sunny" , "If we do not go swimming then we will take a canoe trip" and "if we take a canoe trip, then we will be home by sunset" lead to the conclusion "we will be home by sunset". (Nov 2013) (BTL4) (8 Marks)** (Refer Classwork) Keypoints: • Denote the statements from the given sentences (1mark) • $\neg P \wedge \neg Q,\ R \rightarrow P,\ \neg R \rightarrow S,\ S \rightarrow T \Rightarrow T$ (2marks) • $\neg P,\ R \rightarrow P \Rightarrow \neg R$ (2marks) • $\neg R, \neg R \rightarrow S \Rightarrow S$ (2marks) • Answer is T. (1mark)
10	**Show that the following premises imply the following conclusion "It rained"** **"If it does not rain or if there is no traffic dislocation, then the sports day will be held and the cultural programme will go on"; "If the sports day is held, then the trophy will be awarded" and "The trophy was not awarded". (May2016) (BTL4) (8 Marks)** (Refer Classwork) Keypoints: • Denote the statements from the given sentences (1mark) • $(\neg P \vee \neg Q) \rightarrow (R \wedge S),\ R \rightarrow T, \neg T \Rightarrow P$ (2marks) • Use rules of inferences to the necessary premises(4marks) • $\neg R, \neg R \rightarrow P \Rightarrow P$.(1mark)
11	**Show that** $R \rightarrow S$ **is logically derived from the premises** $P \rightarrow (Q \rightarrow S)$ **,** $\neg R \vee P$ **and Q.**

	(Apr2017, Nov2015, May2016) (BTL3) (8 Marks) Keypoints: • R is an additional premise(2marks) • $R, \neg R \vee P \Rightarrow P$ (2marks) • $P, P \rightarrow (Q \rightarrow S) \Rightarrow Q \rightarrow S$ (3marks) • Get the answer as S(1mark)
12	**Show that** $(p \rightarrow q) \wedge (r \rightarrow s), (q \rightarrow t) \wedge (s \rightarrow u), \neg(t \wedge u), (p \rightarrow r) \Rightarrow \neg p$ **. (Apr 2015)(BTL3) (8 Marks)** Keypoints: • $(p \rightarrow q), (q \rightarrow m) \Rightarrow p \rightarrow m$ (2marks) • $(p \rightarrow r), (r \rightarrow n) \Rightarrow p \rightarrow n$ (4marks) • Get the answer as $\neg p$ (2marks)
13	**Show that** $\exists x(P(x) \wedge Q(x)) \Rightarrow \exists x\, P(x) \wedge \exists x\, Q(x)$ **. (Nov 2013)(BTL3) (8 Marks)** (Refer Balaji Pg. 1.146) Keypoints: • $p(y) \wedge Q(y)$ (2marks) • $\exists x P(x)$ (2marks) • $\exists x Q(x)$ (2marks) • $\exists x\, P(x) \wedge \exists x\, Q(x)$ (2marks)
14	**Show that** $\forall x(P(x) \vee Q(x)) \Rightarrow \forall x\, P(x) \vee \exists x\, Q(x)$ **. (Apr 2015, Apr2018) (BTL3) (8 Marks)** (Refer Balaji Pg. 1.147) Keypoints: • Using indirect method Assume $\neg(\forall x\, P(x) \vee \exists x\, Q(x))$ (2marks) • $\neg(P(y) \wedge Q(y))$ (4marks) • Answer F (2marks)
15	**Show that** $\forall x(P(x) \rightarrow Q(x)) \wedge (Q(x) \rightarrow R(x)) \Rightarrow \forall x\, (P(x) \rightarrow Q(x))$ **. (Nov2016) (BTL3) (8 Marks)**

	(Refer Balaji Pg.1.145) Keypoints: • $P(y) \rightarrow Q(y)$ (2marks) • $P(y) \rightarrow R(y)$ (4marks) • $\forall x\,(P(x) \rightarrow Q(x))$ (2marks)
16	**Use rules of inferences to obtain the conclusion of the following arguments: "one student in this class knows how to write a program in JAVA" and "Everyone who knows how to write programs in JAVA can get high paying job" imply the conclusion "someone in this class can get a high paying job". (Nov2015, Apr 2017) (BTL4) (8 Marks)** Keypoints: • $\exists x(P(x) \wedge Q(x)),\ \forall x(Q(x) \rightarrow R(x)) \Rightarrow \exists x(P(x) \wedge R(x))$ (2marks) • $P(a) \wedge Q(a)\ Q(a) \rightarrow R(a)$ (2marks) • $P(a) \wedge Q(a)$ (3marks) • $\exists x(P(x) \wedge R(x))$ (1mark)
17	**Prove that** 2 **is irrational by giving a proof by contradiction. (Nov 2013, May2016) (8 Marks)** (Refer SKD Pg. 1.78) **(BTL5)** Keypoints: • Use indirect method , Assume 2 is irrational (2marks) • $2 = \frac{p}{q}$ (2marks) • q = 2k (2marks) • a contradiction that 2 is rational (4m)

UNIT II – COMBINATORICS

Mathematical induction – Strong induction and well ordering – The basics of counting – The pigeonhole principle – Permutations and combinations – Recurrence relations – Solving linear recurrence relations – Generating functions – Inclusion and exclusion principle and its applications

PART A

Q.No.	Questions
1.	**State the first Principle of mathematical induction. (BTL1)** Let P(n) be a proposition corresponding to positive integers n. (i) If $P(n_0)$ is true for some integer n_0 (ii) If P(k) is true for an arbitrary integer k ($(>n_0)$ then P(k+1) is true Then P(n) is true, for all $n \geq n_0$.
2	**State the Principle of strong induction. (BTL1)** Let P(n) be a proposition corresponding to positive integer n. (i) If $P(n_0)$ is true for some integer n_0 and (ii) If the proposition is true for all integers upto k($>n_0$) then P(k+1) is true Then P(n) is true, for all $n \geq n_0$.
3	**Use mathematical induction to show that** $1 + 2 + 3 + ...+ n = \frac{n(n+1)}{2}$ **. (BTL5)** **Basic step:** To prove P(1) is true L.H.S = 1 R.H.S= $\frac{1(1+1)}{2}=1$ L.H.S = R.H.S Hence P(1) is true. **Inductive step:** Let us assume that P(k) is true for any positive integer k(>1) (i.e.) $P(k) = 1+2+3+\square+k = \frac{k(k+1)}{2}$ **Step 3:** To prove P(k+1) is true (i.e.) $P(k+1) = 1+2+3+\square+(k+1) = \frac{(k+1)(k+2)}{2}$ L.H.S = $1+2+3+\square+k+(k+1)$ $= \frac{k(k+1)}{2}+(k+1)$ $= \frac{(k+1)(k+2)}{2}$ P(k+1) is true when P(k) is true. Therefore by first principle of mathematical induction P(n) is true for all $n \geq 1$.
4	**State the Pigeonhole principle. (BTL1)** If n+1 pigeons are assigned to n pigeonholes, then there must be a pigeonhole containing atleast

	two pigeons.
5	**What is well ordering principle. (BTL1)** The well ordering principle states that every non-empty set of non-negative integers has a smallest element.
6	**How many bit strings are there of length seven? (BTL4)** Each position can be filled up with two choices 0‟s or 1‟s. Therefore number of different bit strings of length 7 = 2^7=128.
7	**What is the number of arrangements of all the six letters in the word PEPPER? (BTL5)** There are 6 letters in the word PEPPER, of which 3-P‟s, 2-E‟s are identical Therefore number of arrangements = $\frac{6!}{3! \times 2!} = 60$
8	**How many different words are there in the word MATHEMATICS. (BTL5)** There are 11 letters in the word MATHEMATICS of which 2-M‟s , 2-A‟s , 2- T‟s are identical. Therefore number of different permutations = $\frac{11!}{2! \times 2! \times 2!} = 4989600.$
9	**How many different words are there in the word ENGINEERING? (BTL5)** There are 11 letters in the word ENGINEERING of which 3-E‟s, 3-N‟s,2-I‟s, 2-G‟s are identical Therefore number of different words = $\frac{11!}{3! \times 3! \times 2! \times 2!} = 277200.$
10	**In how many ways can the letters of the word MISSISSIPPI be arranged? (BTL5)** There are 11 letters in the word MISSISSIPPI of which 4-I‟s, 4-S‟s, 2-P‟s are identical Therefore number of arrangements = $\frac{11!}{4! \times 4! \times 2!} = 34650.$
11	**How many permutations of {a,b,c,d,e,f,g) end with „a‟? (BTL3)**

	Here repeats are not allowed The last position must be an „a" So we have only 6 items in place. Therefore $6P_6 = 720$ permutations.
12	**Find the recurrence relation of the equation** $S(n)=a^n, n\geq 1$**. (BTL3)** **Given:** $S(n) = a^n$, $S(n-1) = a^{n-1} = a^n . a^{-1}$ $= S(n)a^{-1}$ $aS(n-1) = S(n)$ The recurrence relation is $S(n)\ \ a\,S(n-1)=0$.
13	**Write the particular solution of the recurrence relation** $a_n = 6a_{n-1} - 9a_{n-2} + 3^n$ **(BTL5)** The homogeneous equation is $a_n - 6a_{n-1} + 9a_{n-2} = 0$ Let $a_n = r^n$, $r^n - 6r^{n-1} + 9r^{n-2} = 0$ $r^{n-2}(r^2 - 6r + 9) = 0$ The characteristic equation is $r^2 - 6r + 9 = 0$ r=3,3 $a_n^{(h)} = (An + B)3^n$ $f(n) = 3^n$ and 3 is a double root of the characteristic equation Therefore the particular solution is $a_n = Cn^2 3^n$.
14	**Solve** $a_k = 3a_{k-1}, k \geq 1$ **with** $a_0 = 2$ **. (BTL5)** **Given :** $a_k - 3a_{k-1} = 0$ Let $a_n = r^n$ $r^n - 3r^{n-1} = 0$ $r^{n-1}(r-3) = 0$ The characteristic equation is r-3=0 Therefore $a_n = A3^n$ ---------- (1)

	Given: $a_0 = 2$ Sub n=0 in (1) $a_0 = A3_0 \Rightarrow A=2$ Therefore the solution is $a_n = 2(3)_n$.
15	**Find the recurrence relation for the equation** $y_n = A(3)_n + B(-4)_n$. **(BTL5)** **Given :** $y_n = A(3)_n + B(-4)_n$ $y_{n+1} = A(3)^{n+1} + B(-4)^{n+1} = 3A(3)^n - 4B(-4)^n$ $y_{n+2} = A(3)^{n+2} + B(-4)^{n+2} = 9A(3)^n + 16B(-4)^n$ $y_n \quad 1 \quad 1$ $y_{n+1} \quad 3 \quad -4 = 0$ $y_{n+2} \quad 9 \quad 16$ $y_n(48+36) - 1(16y_{n+1} + 4y_{n+2}) + 1(9y_{n+1} - 3y_{n+2}) = 0$ $84y_n - 7y_{n+1} - 7y_{n+2} = 0$ $\Rightarrow 12y_n - y_{n+1} - y_{n+2} = 0$
16	**Solve the recurrence relation** $y(k) - 8y(k-1) + 16y(k-2) = 0, k \geq 2$ **where y(2)=16, y(3)=80.** **(BTL5)** **Given :** $y(k) - 8y(k-1) + 16y(k-2) = 0,$ Let $y(k) = r^n$ $r^n - 8r^{n-1} + 16r^{n-2} = 0$ $r^{n-2}(r^2 - 8r + 16) = 0$ The characteristic equation is $r^2 - 8r + 16 = 0$ The roots are r = 4,4 Therefore $y_k = (Ak + B)4_k$ ---------- (1) **Given:** y(2)=16, y(3)=80 $y_2 = (A2 + B)4_2$ Put k=2, $16 = 32A + 16B$ $2A + B = 1 \quad ----------(2)$

	$y_3 = (A3 + B)3_2$ Put k=3, $80 = 27A + 9B$ $3A + B = \frac{5}{4}$ ----------(3) Solving (2) and (3), $A = \frac{1}{4}, B = \frac{1}{2}$ (1) Implies $y_k = (\frac{k}{4} + \frac{1}{2})4^k$
17	**Write the generating function for the sequence** $1, a, a^2 a^3, \ldots\ldots$ **(BTL1)** The generating function for the sequence $1, a, a^2 a^3, \ldots$ is the infinite series $G(x) = 1 + ax + a^2 x^2 + a^3 x^3 + \ldots\ldots$ $= \frac{1}{1 - ax}$ *if* $ax < 1$
18	**Find the closed form generating function of the sequence 2,-2,2,-2,…. (BTL3)** $G(x) = \sum_{n=0}^{\infty} a_n x^n$ $= a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots\ldots$ Generating function $= 2 + (-2)x + (2)x^2 + (-2)x^3 + \ldots\ldots$ $= 2[1 - x + x^2 - x^3 + \square]$ $= 2(1 - x)^{-1} = \square \frac{2}{1 + x}$
19	**What is the maximum number of students required in a mathematics class to be sure that at least six will receive the same grade, if there are five possible grades A,B,C,D and F? (Nov 2012) (BTL4)** The minimum number of students wanted to ensure that atleast six students receive the same grade is the smaller integer N such that $\frac{N}{5} = 6$. The smallest such integer is N=5(5)+1 =26 If you have only 25 students, it is possible for there to be five students who have received each grade so that no six students have received the same grade. Therefore 26 is the minimum number of students needed to ensure that atleast six students will receive the same grade.

20	**How many ways are there to select five players from a 10 member tennis team to make a trip to a match at another school? (BTL3)** Number of ways to select five players form 10 members $=10C_2 = 252$
21	**If seven colours are used to paint 50 bicycles, then show that atleast 8 bicycles will be the same colour.(BTL3)** Number of Pigeon = m = Number of bicycles=50 Number of Holes=n= Number of colours =7 By Generalised pigeon hole principle, we get $\left\lceil \frac{m-1}{n} \right\rceil + 1 = \left\lceil \frac{50-1}{7} \right\rceil + 1 = 8$
22	**Find the recurrence relation of the Fibonacci sequence. (BTL1)** The Fibonacci sequence is 0,1,1,2,3,5,8,13,….. (i.e.) $F_n = F_{n-1} + F_{n-2} \quad n \geq 2$ The recurrence relation is $F_n - F_{n-1} - F_{n-2} = 0 \quad n \geq 2$ with initial conditions $F_0 = 0$ *and* $F_1 = 1$.
23	**Define Permutation and combination. (BTL1)** A permutation is an arrangement of a given collection of objects in a definite order taking some of the objects or all at a time The number of r-permutations is denoted by nP_r and is defined as $nP_r = \frac{n!}{(n-r)!}$ A combination is a selection of objects from a given collection of objects taking some or all at a time. The order of selection is immaterial. The number of r-combinations from n things is denoted by nC_r $C(n, r)$ and is defined as $nC_r = \frac{nP_r}{r!}$.
24	**Find the number of solutions of the equation $x_1 + x_2 + x_3 = 100$, if x_1, x_2, x_3 are non-negative integers. (BTL5)** **Given:** The numbers are non-negative So the set of numbers are {0,1,2,3,...} Therefore the number of solutions = coefficient of x^{100} in $(x^0 + x^1 + x^2 + ...)$ = coefficient of x^{100} in $(1 + x^1 + x^2 + ...)$ = coefficient of x^{100} in $(1-x)^{-3}$ $= {}^{(3+100-1)}C_{100} = {}^{102}C_2$

	=5151
25	**Compute the number of 13 card hands that can be dealt from a deck of 52 cards?(Nov 2007) (BTL3)** The number of 13 card hands that can be dealt from a 52 cards is $52C_{13} = 6350135596\ 00$
	Part-B
1	**Prove by mathematical induction $6^{n+2} + 7^{2n+1}$ is divisible by 43. (Nov 2013) (BTL5) (8 Marks)** (Refer SKD Pg. 2.19) Keypoints: • Prove for P(1) (i.e.,) 559 is divisible by 43 (2marks) • Assume P(k) is true (i.e.,) $6^{k+2} + 7^{2k+1}$ (2marks) • Prove P(k+1) is true (i.e.,) $6^{k+3} + 7^{2k+3}$ (4marks)
2	**Prove by Mathematical induction $1^2 + 2^2 + 3^2 + ...+ n^2 = \frac{n(n+1)(2n+1)}{6}$. (May 2015) (BTL5) (8 Marks)** (Refer Balaji Pg. 2.2) Keypoints: • Prove for P(1) (i.e.,) 1 is divisible by 1 (2marks) • Assume P(k) is true (i.e.,) $1^2 + 2^2 + 3^2 + ...+ k^2 = \frac{k(k+1)(2k+1)}{6}$ (2marks) • Prove P(k+1) is true (i.e.,) $\frac{(k+1)(k+2)(2k+3)}{6}$ (4marks)
3	**Using Mathematical induction , show that $\sum_{r=0}^{n} 3^r = \frac{3^{n+1}-1}{2}$ (May 2017, May 2016) (BTL5) (8Marks)** (Refer Classwork) Keypoints: • Prove for P(1) (i.e.,) 1 is divisible by 1 (2marks) • Assume P(k) is true (i.e.,) $\sum_{r=0}^{k} 3^r = \frac{3^{k+1}-1}{2}$ (2marks) • Prove P(k+1) is true (i.e.,) $\frac{3^{k+2}-1}{2}$ (4marks)

4	**Using induction principle, prove that $n^3 + 2n$ is divisible by 3. (Nov 2015) (BTL5) (8 Marks)** (Refer SKD Pg. 2.41) Keypoints: • Prove for P(1) (i.e.,) 3 is divisible by 3 (2marks) • Assume P(k) is true (i.e.,) $k^3 + 2k = 3x$ (2marks) • Prove P(k+1) is true (i.e.,) $3(k^2 + k + x + 1)$ is divisible by3(4marks)
5	**Prove that $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + ... \frac{1}{n} > n$, $n \geq 2$, using principle of mathematical induction. (Nov 2016) (BTL5) (8 Marks)** (Refer SKD Pg. 2.42) Keypoints: • Prove for P(2) (i.e.,) $1 + \frac{2}{2} \geq 2$ is true (2marks) • Assume P(k) is true $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + ... \frac{1}{k} > k$, (2marks) • Prove P(k+1) is true (4marks)
6	**A factory makes custom sports car at an increasing rate. In the first month one car is made, in the second month two cars are made and so on, with n cars made in the nth month.** **(1) Set up recurrence relation for the number of cars produce in the first n months by this factory** **(2) How many cars are produced in the first year? (Nov 2013) (BTL4) (8 Marks)** Keypoints: • Form the recurrence relation as $P_n = P_{n-1} + n,\ n \geq 1\ \ P_0 = 0$ (3marks) • Find the number of cars in 12 months using the formula $\frac{n(n+1)}{2}$ (5marks)
7	**Use the method of Generating functions to solve the recurrence relation $a_n = 3a_{n-1} + 2,\ n \geq 1$, given that $a_0 = 1$ (May 2015) (BTL5) (8 Marks)** (Refer Balaji Pg. 2.85) Keypoints:

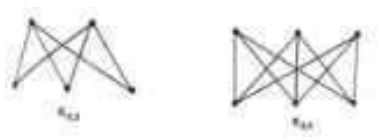
	• $a_n x^n = 3a_{n-1}x^{n-1} + 2x^n$ (1mark) • $\sum a_n x^n = \sum 3a_{n-1}x^{n-1} + \sum 2x^n$ (1mark) • $G(x) = \frac{1+x}{(1-x)(1-3x)}$ (1mark) • A=-1 , B=2 (4mark) • a_n= coeff of x^n in G(x) (1mark)
8	**Solve the recurrence relation** $a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$ **with** $a_0 = 5$, $a_1 = -9$, $a_2 = 15$ **. (Nov 2014) (BTL5) (8 Marks)** (Refer Balaji Pg. 2.74) Keypoints: • Put $a_n=r^n$ (1mark) • $r = -1,-1,-1$ (2marks) • $a_n = A(-1)^n + Bn(-1)^n + Cn^2(-1)^n$ (3marks) • A=1 , B= 0.5 , C=0.5 (2marks)
9	**Find the solution to the recurrence relation** $a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$ **with** $a_0 = 2$, $a_1 = 5$, $a_2 = 15$**.** **(Nov 2014) (BTL5) (8 Marks)** (Refer SKD Pg. 2.134) Keypoints: • Put $a_n=r^n$ (1mark) • r = 1,2,3 (2marks) • $a_n = A + B2^n + C3^n$ (3marks) • A=1, B=-1, C=2 (2marks)
10	**Solve using Generating function** $S(n+1) - 2S(n) = 4^n$; **, S(0)=1,** $n \geq 0$ **(May 2016) (BTL5) (8 Marks)** (Refer SKD Pg. 2.158) Keypoints: • $a_{n+1}x^{n} - 2a_n x^n = 4^n x^n$ (1mark) • $\sum_{n=1}^{\infty} a_{n+1}x^n - 2\sum_{n=1}^{\infty} a_n x^n = \sum_{n=1}^{\infty} 4^n x^n$ (1mark)

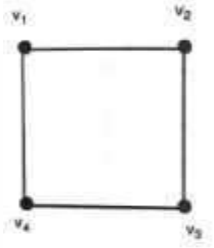
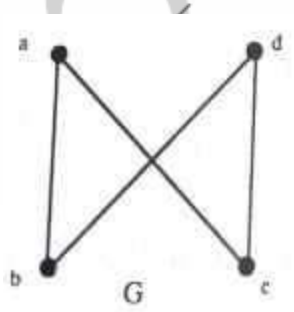
	• Obtain G(x) (1mark) • Solve the obtained equation by partial fractions (3marks) • a_n= coeff of x^n in G(x) (2marks)
11	**Prove that in a group of six people, atleast 3 must be mutual friends or atleast 3 must be mutual strangers. (Nov 2015) (BTL5) (8 Marks)** (Refer SKD Pg. 2.109) Keypoints: • $\left\lceil \begin{matrix} m-1 \\ n \end{matrix} +1 \right\rceil$ • Fix one of the friends, say A (2marks) • Form two groups as friends of A and strangers of A (3marks) • Check if the pigeon hole principle is satisfied in both the cases (3marks)
12	**How many bits of string of length 10 contain** **(1) Exactly four 1"s (2) Atleast four 1"s** **(3)Atleast 4 1"s (4) an equal number of 0"s and 1"s** **(Nov 2016) (BTL3) (8 Marks)** (Refer Balaji Pg. 2.47) Keypoints: • (1) Use the formula $\frac{n!}{n_1!.n_2!.....n_r!}$ answer = 210 (2marks) • (2) Use the formula $\frac{n!}{n_1!.n_2!.....n_r!}$ answer = 386 (2marks) • (3) Use the formula $\frac{n!}{n_1!.n_2!.....n_r!}$ answer = 848 (2marks) • (4) Use the formula $\frac{n!}{n_1!.n_2!.....n_r!}$ answer = 252 (2marks)
13	**Find the number of integers between 1 and 250 that are not divisible by any of the integers 2,3,5 and 7. (May 2015, Nov 2016,May 2016, May 2018) (BTL4) (8 Marks)** (Refer SKD Pg. 2.94)

	Keypoints: • Use the formula $\frac{n}{P_1P_2}$ where P_1 and P_2 are distinct primes (2marks) • Substitute the values in $\lvert A \cup B \cup C \cup D \rvert$ (4marks) • Number of integers that are not divisible by 2,3,5 and 7 is got by $\lvert \overline{A \cup B \cup C \cup D} \rvert$ (2marks)
14	**Find the Generating function of Fibonacci sequence. (Nov 2013) (BTL5) (8 Marks)** (Refer Balaji Pg. 2.91) Keypoints: • The Fibonacci sequence is 0,1,1,2,3,5,8,…. (1mark) • $G(x) = \sum_{k=0}^{\infty} f_k x^k$ (1mark) • $G(x) = \frac{x}{1 - x - x^2}$ (2mark) • $A = \frac{1+\sqrt{5}}{2}, B = \frac{1-\sqrt{5}}{2}$ (2marks) • a_n= coeff of x^n in G(x) (2mark)
15	**A total 1232 students have taken a course in Spanish, 879 have taken a course in French and 114 have taken a course in Russian. Further 103 have taken a course in both Spanish and French, 23 have taken a course in both Spanish and Russian and 14 have taken courses in both French and Russian. If 2092 students have atleast one of Spanish, French and Russian, how many students have taken a course in all 3 consequences? (Nov 2013, Nov 2017) (BTL4) (8 Marks)** (Refer Balaji Pg. 2.97) Keypoints: • Draw the venn diagram using the given data (4marks) • Substitute the necessary values in $A \cup B \cup C = A + B + C - A \cap B - B \cap C - A \cap C + A \cap B \cap C$ (4marks)
16	**There are 6 men and 5 women in a room. Find the number of ways 4 persons can be drawn from the room if (1) they can be male or female (2) two must be men and two women (3) they must all be of the same sex.** **(Nov 2015, May 2016, May 2017) (BTL4) (8 Marks)**

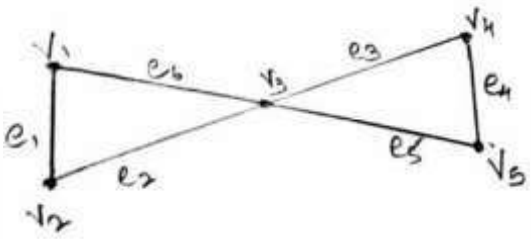
	Keypoints: • ${}^nC_r = \frac{n!}{r!(n-r)!}$ (2marks) • (i) Answer =330ways (2marks) • Answer = 150 (2marks) • Answer = 20 (2marks)
17	**If H_n denote Harmonic numbers, then prove that $H_{2^n} \geq 1 + \frac{n}{2}$. (Nov 2017) (BTL5) (8 Marks)** (Refer Balaji Pg. 2.10) Keypoints: • Prove for P(1) (i.e.,) H_1=1(2marks) • Assume P(k) is true (i.e.,) $H_{2^k} \geq 1 + \frac{k}{2}$ (2marks) • Prove P(k+1) is true (i.e.,) $H_{2^{k+1}} \geq 1 + \frac{k+1}{2}$ (4marks)
19	**Using induction principle , prove that $n^3 - n$ is divisible by 3. (May 2018) (BTL5) (8 Marks)** (Refer Balaji Pg. 2.12) Keypoints: • Prove for P(1) (i.e.,) 0 is divisible by 3 (2marks) • Assume P(k) is true (i.e.,) $k^3 - k$ is divisible by 3 (2marks) • Prove P(k+1) is true (i.e.,) $(k+1)^3 - (k+1)$ (2marks)
	UNIT III – Graphs
	Graphs and graph models – Graph terminology and special types of graphs – Matrix representation of graphs and graph isomorphism – Connectivity – Euler and Hamilton paths.
	PART A
Q.No.	**Questions**
1.	**Define a simple graph. (BTL1)** A graph G=(V,E) without loops and without parallel edges is called a simple graph.
2	**Define Degree of a vertex. (BTL1)**

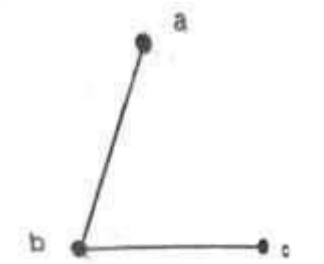
	The degree of a vertex in a graph G is the number of edges incident with it. A loop at a vertex contributes degree 2 to that vertex. Degree of a vertex v is denoted by deg(v).
3	**Show that the sum of the degree of all vertices in G is twice the number of degree in G. (Nov 2012) (BTL1)** Every non-loop edge is incident with two vertices and so contributes 2 to the degree. Every loop edge contributes 2 to the degree. Therefore edge contributes 2 to the sum of degrees of the vertices. So all the e edges contribute 2e degrees. Therefore sum of degrees of vertices = 2e $\Rightarrow \sum_{i=1}^{n} \deg(v_i) = 2e$
4	**Define complete graph (Nov 2011, May 2014 ,Nov 2016) (BTL1)** A simple graph is called a complete graph if there is exactly one edge between every pair of vertices. A complete graph on n vertices is denoted by Kn.
5	**Draw the complete graph K5. (Nov 2015) (BTL1)**
6	**How many edges are there in a graph with 10 vertices each of degree 5? (May 2017, May 2016) (BTL3)** Let „e" be the number of edges of the graph **Given:** 10 vertices each of degree 5 By Handshaking theorem, $\sum_{i=1}^{n} \deg(v_i) = 2e$ $10(5) = 2e$

	e = 25 Therefore number of edges = 25
7	**Show that there does not exist a graph with 5 vertices with degrees 1,3,4,2,3 respectively. (May 2018) (BTL3)** Sum of the degree of all the vertices = 1+3+4+2+3 = 13 Which is an odd number Hence no with the even degree.
8	**Define a Regular graph. Can a complete graph be a regular graph? (Apr 2006, Nov 2012) (BTL1)** A simple graph is called regular if every vertex of the graph has the same degree. If every vertex in a regular graph has a degree k, the graph is k-regular Any complete graph is regular, but the converse is not true.
9	**Define Pseudographs (Apr 2011) (BTL1)** A graph in which loops and parallel edges are not allowed is called pseudo graphs.
10	**Let G be a graph with 10 vertices. If 4 vertices have degree 4 and 6 vertices has degree 5, them find the number of edges of G? (Nov 2015) (BTL3)** Let e be the number of edges of the graph **Given:** 4 vertices have degree 4 6 vertices have degree 5 By Handshaking theorem, $\sum_{i=1}^{n} \deg(v_i) = 2e$ 4(4) +6(5) =2e 46 = 2e e = 23 Therefore number of edges = 23
11	**Draw the complete bipartite graph** $K_{2,3}$ *and* $K_{3,3}$**.** 

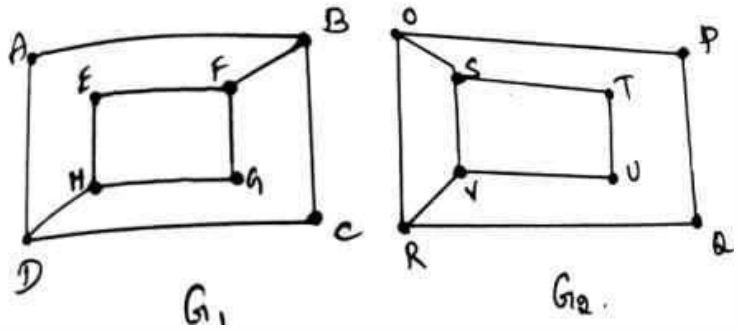
12	**Draw the graph represented by the given adjacency matrix** $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$ **(Nov 2013) (BTL6)** 
13	**Define isomorphism of directed graphs. (Nov 2014) (BTL1)** Two graphs $G_1 = (V_1 E_1)$, $G_2 = (V_2 , E_2)$ are said to be isomorphic if there is a one- to-one and onto function from V_1 to V_2 such that $(a, b) \in E_1$ iff $(f(a), f(b)) \in E_2$. We write $G_1 \approx G_2$.
14	**Define a connected graph and disconnected graph with examples. (May 2015) (BTL1)** A graph is connected if there is a path between every pair of distinct vertices of the graph. A graph which is not connected is disconnected.
15	**For the graph G given by the figure. Find the number of paths of length 4 from a to d. (Nov 2012) (BTL4)** 

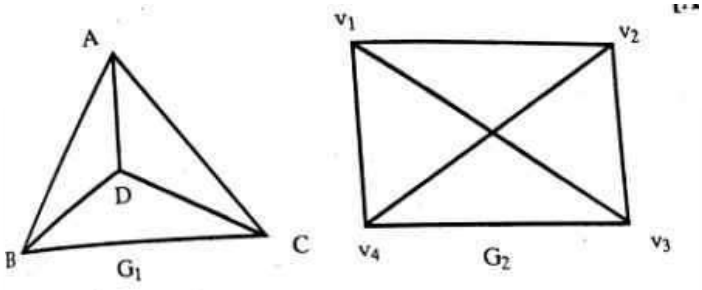
	The adjacency matrix of G is $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ Since „a" is the first vertex and „d" is the 4th vertex, the number of paths of length 4 from a to d is (1,4)th element A_4. $A_4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$ Therefore number of paths of length 4 from a to d is 8.
16	**Give an example of self-complementary graph. (Apr 2017, May 2016) (BTL3)** A graph G is said to be self-complimentary if G and G^c are isomorphic. Example Number of vertices, edges and degree sequences of C_5 *and* C_5^c are equal. Let $f: V_1 \to V_2$ $\therefore f(u_1) = v_1, f(u_2) = v_4, f(u_3) = v_2, f(u_4) = v_5, f(u_5) = v_3$ Clearly f is 1-1 and onto which preserves adjacency $\therefore C_5$ *and* C_5^c are isomorphic graphs.
17	**Define Euler path and Euler circuit . (BTL1)** A path of a graph G is called an Euler path if it contains each edge of the graph exactly once. An Euler circuit in a graph G is a simple circuit that includes every edge of G exactly once with same starting and ending vertex.
18	**Define Hamiltonian path and Hamilton circuit. (May 2018) (BTL1)** A path of a graph G is called a Hamilton path if it contains each vertex of G exactly once. A Hamiltonian cycle in a graph G is a simple circuit that includes each vertex of G exactly once

	except the starting and the ending vertex.
19	**Give an example of a graph which is Eulerian but not Hamiltonian. (Apr 2015, Nov 2017) (BTL3)**  All the vertices are of even degree $\therefore$ Eulerian Cycle is possible $\therefore v_1 - v_3 - v_4 - v_5 - v_3 - v_2 - v_1$ No edges are repeated and cover all the edges. But no Hamiltonian, because Hamiltonian circuit is not possible The vertices are repeated , so it is not Hamiltonian.
20	**Define strongly connected and weakly connected graph. (Nov 2010) (BTL1)** A directed graph G is said to be strongly connected if there is a path u to v and from v to u for any pair of vertices u and v in G. A directed graph is said to be weakly connected if there is a path between any two vertices of the underlying undirected graph((i.e.) without considering directions)
21	**Define complete bipartite graph. (BTL1)** Let G=(V,E) be a bipartite graph with bipartition (V_1, V_2) . If there is an edge of G connecting every vertex in V_1 and in V_2 then G is called a complete bipartite graph.
22	**What should be the degree of each vertex of a graph G if it has Hamiltonian? (BTL4)** Let G be a simple graph with n vertices where $n \geq 3$. If $\deg(v) \geq n/2$ for each vertex v, then G is Hamiltonian.
23	**Define cut vertex and cut edge. (BTL1)** A cut vertex of a connected graph G is a vertex whose removal increase the number of components. If v is a cut vertex of the connected graph G, then G-v is disconnected A cut edge or bridge of a graph is an edge whose removal increase the number of components. If e is an edge of a connected graph G, then G-e is disconnected.
24	**Define path and cycle. (BTL1)**

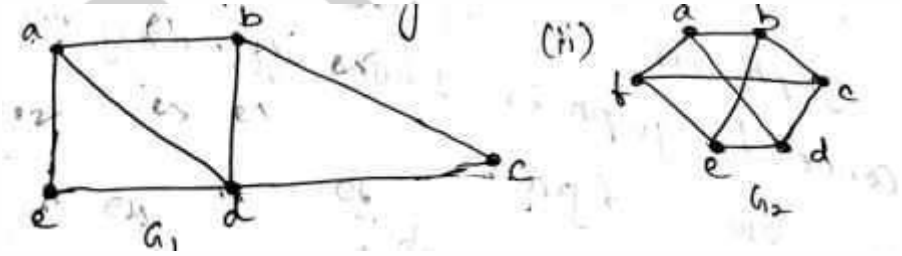
	A path in a graph G is a finite alternating sequence of vertices and edges beginning and ending with vertices. If the initial and final vertices of a path are the same then the path is called a cycle or circuit.
25	**Draw the graph represented by the given adjacency matrix and** $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ **(Nov 2016) (BTL6)** 
	Part-B
1	**Prove that the number of vertices of odd degree in any group is even. (May 2015, Nov 2015, May 2016, May 2017) (BTL5) (8 Marks)** (Refer Balaji Pg. 3.21) Keypoints: • $\sum_{i=1}^{n} d(v_i) + \sum_{j=1}^{m} d(v_j) = 2e$ (2marks) • Use Handshaking theorem , $\sum d\ (v) = 2e$ (4marks) • $\sum_{i=1}^{k} d(v_i) = evennumber$ (2marks)
2	**State and Prove Handshaking theorem. Hence prove that for any simple graph G with n vertices, the number of edges of G is less than or equal to** $\frac{n(n-1)}{2}$ **. (Nov 2016, May 2018) (BTL5) (8 Marks)** (Refer Balaji Pg. 3.20, 3.24) Keypoints: • Prove Handshaking theorem $\sum d\ (v) = 2e$ (3marks)

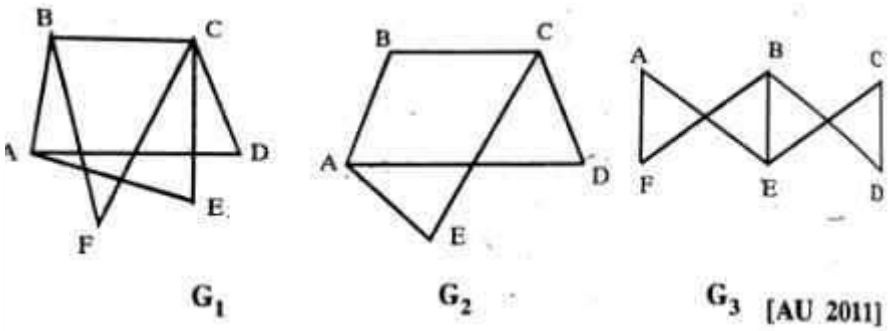
	• To prove the second part, Use handshaking theorem (2marks) • Use the result, Maximum degree of each vertex in G is (n-1). (2marks) • $e = \frac{n(n-1)}{2}$ (1mark)
3	**Prove that a simple graph with n vertices and k components cannot have more than $\frac{(n-k)(n-k+1)}{2}$ edges. (Nov 2013 , Nov 2015, May 2015, Nov 2017) (BTL5) (8 Marks)** (Refer Balaji Pg. 3.70) Keypoints: • Consider a simple graph (2marks) • Consider components with k vertices (2marks) • $E(G) \le \sum_{i=1}^{k} \frac{n_i(n_i-1)}{2}$ (2marks) • $E(G) \le \frac{(n-k)(n-k+1)}{2}$
4	**Show that a simple graph G with n vertices is connected if it has more than $\frac{(n-1)(n-2)}{2}$ edges.** **(Nov 2014) (BTL5) (8 Marks)** (Refer Balaji Pg. 3.76) Keypoints: • Proof by contradiction (i.e) Assume G has components (2marks) • Using previous theorem, $E(G) \le \frac{(n-k)(n-k+1)}{2}$ (2marks) • $E(G) > \frac{(n-1)(n-2)}{2}$ (4marks)
5	**Show that isomorphic of simple graphs is an equivalence relation. (Nov 2014) (8 Marks)** (Refer Balaji Pg. 3.61) Keypoints: • Reflexive: G is isomorphic to itself by the identity (3marks) • Symmetric : f^{-1} is a 1-1 correspondence from H to G that preserves adjacency and non-adjacency(3marks) • Transitive: If G is isomorphic to H and H is isomorphic to K, then there is a 1-1

	correspondence f and g from G to H and from H to K(2marks)
6	**Examine whether the following pair of graphs are isomorphic or not. Justify your answer. (My 2015 , Nov 2015) (BTL4) (8 Marks)** (Refer Classwork)  (Refer Balaji Pg. 3.57) Keypoints: • Number of vertices and degree sequences are equal in both the graphs. (3marks) • Incidences of both the graphs are not satisfied (3marks) • If necessary check for equal number of circuits in both the graphs. (1mark)
7	**Define isomorphism between two graphs. Are the simple graphs with the following adjacency matrices isomorphic (May 2016, May 2017) (BTL4) (8 Marks)** $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$ $\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$ (Refer Class work) Keypoints: • Draw the graph (2marks) • Check if number of vertices and degree sequences are equal in both the graphs. (2marks) • Check the incidences of both the graphs(3marks)

	• If necessary check for equal number of circuits in both the graphs. (1mark)
8	**Define Isomorphism. Establish an Isomorphism for the following graphs. (Nov 2011, Nov 2016) (8 Marks)**  (Refer SKD Pg. 3.49) Keypoints: • Two graphs $G_1 = (V_1 E_1)$, $G_2 = (V_2, E_2)$ are said to be isomorphic if there is a one- to-one and onto function from V_1 to V_2 such that (a, b) are adjacent in G1iff ($f(a)$, $f(b)$) are adjacent in G_2 we write $G_1 \approx G_2$. (2marks) • Check if number of vertices and degree sequences are equal in both the graphs. (2marks) • Check the incidences of both the graphs (3marks) • If necessary check for equal number of circuits in both the graphs.(1mark)
9	**Prove that a simple graph is bipartite if and only if it is possible to assign one of two different colours to each vertex of the graph so that no two adjacent vertices are assigned the same colour. (Nov 2017) (BTL5) (8 Marks)** (Refer Balaji Pg.3.40) Keypoints: • Explain Bipartition (2marks) • Assign colours to each vertex in the bipartitions (3marks) • Every edge connects a vertex V_1 and a vertex in V_2 since no two adjacent vertices are either both in V_1 or both in V_2. Consequently G is bipartite(3marks)
10	**Prove that the complement of a disconnected graph is connected. (May 2017) (BTL5) (8 Marks)** (Refer SKD Pg. 3.54) Keypoints: • G has two connected components G_1 and G_2(2marks) • Consider complement of a graph $\overline{G}$ (2marks)

	• Using connected components prove the theorem. (4marks)
11	**Prove that a given connected graph G is an Euler graph if and only if all the vertices of G are of even degree. (Nov 2013, Nov 2015, May 2018) (BTL5) (8 Marks)** (Refer Balaji Pg. 3.83) Keypoints: • Consider an Euler graph, then it has an Euler circuit (1mark) • Consider an Euler circuit (1mark) • Using definition of Euler circuit and prove that all the vertices are of even degree (2marks) • Conversely assume all the vertices are of even degree(2marks) • Construct an Euler circuit and prove if the graph is Euler. (2marks)
12	**If G is self complimentary graph, then prove that G has** $n \equiv 0$ *or* $1 \pmod 4$ **vertices. (May 2016) (BTL5) (8 Marks)** (Refer SKD Pg. 3.25) Keypoints: • $V(G) = V(G)$, $E(G) = E(G)$ (1mark) • $E(K_p) = {}_pC_2$ (2marks) • $E(G) = \frac{p(p-1)}{2}$ (2marks) • P=4n or p-1=4n . (3marks)
13	**If G is connected simple graph with n vertices with** $n \geq 3$ **, such that the degree of every vertex in G is atleast n/2, then prove that G has Hamilton cycle. (May 2017 , May 2016) (BTL5) (8 Marks)** (Refer Classwork) Keypoints: • Consider G cannot be complete (1mark) • Check if it is Hamiltonian if an edge is added (2marks) • Split the vertices (2marks) • Prove that the contradiction is false. ((3marks)
14	**Give an example of a graph which is** **(1) Eulerian but not Hamiltonian** **(2) Hamiltonian but not Eulerian**

	(3) Hamiltonian and Eulerian **(4) Neither Hamiltonian nor Eulerian (Nov 2016) (BTL3) (8 Marks)** (Refer Balaji Pg. 3.99) Keypoints: • (1)Give suitable examples and explain it (2marks) • (2) Give suitable examples and explain it (2marks) • (3) Give suitable examples and explain it (2marks) • (4) Give suitable examples and explain it (2marks)
15	**Which of the following simple graphs have a Hamilton circuit or if not a Hamilton path (Nov 2013) (BTL4) (8 Marks)** (Refer Balaji Pg. 3.95) Keypoints: • G_1 has Hamilton circuit (3marks) • G_2 has no Hamilton circuit (3marks) • G_3 has neither Hamilton circuit nor Hamilton path (2marks)
16	**Find an Euler path or an Euler circuit if it exists in the following graphs. If it does not exist, explain why? (Apr 2015) (BTL4) (8 Marks)**  (Refer Balaji Pg. 3.81) Keypoints: • G_1 has two vertices of odd degree so it does not have Euler circuit, but has an Euler path (4marks) • G_2 has all the vertices as odd, so neither Euler path nor Euler circuit is possible(4marks)

17	**Determine which of the following graphs are bipartite and which are not. If a graph is bipartite, state if it is completely bipartite. (Nov 2011) (BTL4) (8 Marks)**  G_1 G_2 G_3 [AU 2011] (Refer SKD Pg. 3.48) Keypoints: • Use definition of bipartition, G_1 is not a bipartite graph (2 marks) • G_2 is bipartite graph, since we can split the vertices into two groups and are not adjacent(3marks) • G_3 is bipartite graph, since we can split the vertices into two groups and are not adjacent(3marks)
	UNIT IV – ALGEBRAIC STRUCTURES
	Algebraic systems – Semi groups and monoids - Groups – Subgroups – Homomorphism‟s – Normal subgroup and cosets – Lagrange‟s theorem – Definitions and examples of Rings and Fields.
	PART A
Q.No.	**Questions**
1.	**Define Group. (BTL1)** A non-empty set G with a binary operation * defined on it is called a group if it satisfies the following: *(1)* **Closure:** Let $a,b \in G$ then $a*b \in G, \forall a,b \in G$ *(2)* **Associative:** Let $a,b,c \in G$ then $a*(b*c)=(a*b)*c \in G$ (3) **Identity:** There exists an element $e \in G$ such that $a*e=e*a=a, \forall a \in G$ where „e‟ is the identity element. (4) **Inverse:** For each $a \in G$ there exists an element a^{-1} such that $a * a^{-1} = a^{-1} * a = e$, where a^{-1} is the identity element.
2	**Define abelian group. (BTL1)**

	If a group (G,*) satisfies a*b=b*a $\forall a, b \in G$, then G is abelian group
3	**Define semigroup with an example (Nov 2014, Nov 2016, Apr 2018) (BTL1)** A non-empty set S together with a binary operation * satisfying *(1)* **Closure:** Let $a,b \in G$ then $a*b \in G, \forall a,b \in G$ *(2)* **Associative:** Let $a,b,c \in G$ then $a*(b*c)=(a*b)*c \in G$ then the set with binary operation is called a semi group. **Example :**"N" the set of all natural numbers is a group under addition.
4	**Define monoid with an example (Nov 2014) (BTL1)** A non-empty set „M" with a binary operation * satisfying *(1)* **Closure:** Let $a,b \in G$ then $a*b \in G, \forall a,b \in G$ *(2)* **Associative:** Let $a,b,c \in G$ then $a*(b*c)=(a*b)*c \in G$ (3) **Identity:** There exists an element $e \in G$ such that $a*e=e*a=a, \forall a \in G$ where „e" is the identity element. Then the set with binary operation is called a monoid. **Example: „**Z" set of all integers is a monoid under multiplication.
5	**Let Z be the group of integers with the binary operation * defined by a*b=a+b-2,** $\forall a, b \in Z$ **.** **Find the identity element of the group** $Z,*$ **. (Apr 2017) (BTL3)** Let e be the identity element Then $a * e = e * a = a$ Now, $a*e=a$ a+e-2 = a e-2 = 0 e=2 2 is the identity element.
6	**Prove that identity element of a group is unique. (Nov 2015) (BTL5)** **Given:** (G,*) is a group **To Prove:** identity element is uniqie Let e_1 and e_2 be two identity elements of G. Suppose e_1 is the identity element $e_1 * e_2 = e_2 * e_1 = e_2$ ---------- (1)

	Suppose e_2 is the identity element $e_2 * e_1 = e_1 * e_2 = e_1$ _____(2) From (1) and (2) $e_1 = e_2$ Therefore identity element is unique
7	**Prove that inverse element of a group is unique. (BTL5)** **Given:** (G,*) is a group **To Prove: identity element is unique** Let $a \in G$ and e is the identity element Let a_1^{-1} and a_2^{-1} be two inverse elements $a_1^{-1} * a = a * a_1^{-1} = e$ ----------(1) $a_2^{-1} * a = a * a_2^{-1} = e$ ----------(2) **To Prove:** $a_1^{-1} = a_2^{-1}$ L.H.S = $a_1^{-1} = a_1^{-1} * e$ $= a_1^{-1} * (a * a_2^{-1})$ (by (2)) $= (a_1^{-1} * a) * a_2^{-1}$ (by associative) $= e * a_2^{-1}$ (by (1)) $= a_2^{-1}$ Therefore inverse element is unique.
8	**For any group G, if $a^2 = e, \forall a \in G$ then G is abelian. (BTL2)** **Given:** $a^2 = e, \forall a \in G$ **To Prove:** G is abelian $a^{-1} * a^2 = a^{-1} * e$ $(a^{-1} * a) * a = a^{-1} * e$ $e * a = a^{-1}$ $a = a^{-1}, \forall a \in G$ (i.e.) Every element has its own inverse Therefore G is abelian
9	**Prove that in a group idempotent law is true for the identity element. (Apr 2018) (BTL5)**

	Given: (G,*) is a group Assume that $a \in G$ is an idempotent element Then, a * a = a $a = a^* e$ $= a^* (a^* a^{-1})$ Now, $= (a^* a)^* a^{-1}$ $= a^* a^{-1}$ $= e$ Therefore a=e Therefore the only idempotent element in a group is its identity element.
10	**State Lagrange"s theorem (May 2008, Nov 2015)** (BTL1) The order of a group H of a finite group G divides the order of the group. (i.e) O(H) divides O(G)
11	**Find the left cosets of {[0],[3]} in the group** $(Z_6, +_6)$ **(May 2016, May 2017) (BTL3)** Let Z_6={[0],[1],[2],[3],[4],[5]} be a group H= {[0],[3]}be subgroup The left cosets are, [0] +H ={0+h / h∈H}**={**[0]+[0], [0]+[3]} = {[0], [3]} = H [1] +H ={1+h / h∈H}**={**[1]+[0], [1]+[3]} = {[1], [4]} [2] +H ={2+h / h∈H}**={**[2]+[0], [2]+[3]} = {[2], [5]} [3] +H ={3+h / h∈H}**={**[3]+[0], [3]+[3]} = {[3], [0]} = H [4] +H ={4+h / h∈H}**={**[4]+[0], [4]+[3]} = {[4], [1]} [5] +H ={5+h / h∈H}**={**[5]+[0], [5]+[3]} = {[5], [2]} Therefore, H= [0] +H = [3] +H, [1] +H = [4] +H , [2] +H = [5] +H are the distinct left cosets of H in $(Z_6, +_6)$
12	**Find the idempotent elements of G= {1,I,-1,-i} under the multiplication operation. (BTL3)** We know that the identity element is the only idempotent element of a group. Here 1 is the identity element. Therefore 1 is the only idempotent element.
13	**Define Normal subgroup . (BTL1)**

	A group (H,*) of (G,*) is called normal subgroup of G if aH = Ha, $a \in G$
14	**Prove or disprove "Every subgroup of an abelian group is normal". (BTL5)** **(Nov 13)** **Given:** (G,*) is abelian. H is a subgroup of G **To Prove:** H is normal Let (G,*) be an abelian group and (H,*) be a subgroup of G. Let $a \in G$ be any element, then aH = {a *h / $h \in H$} ={h * a / $h \in H$} (since G is abelian) Ha, for all $a \in G$ Therefore H is a normal subgroup of G
15	**Prove that every cyclic group is abelian. (May 2016) (BTL5)** **Given:** G is cyclic group **To Prove:** G is abeliana Let $G = \{a^n / n \in Z\}$ Let $x,y \in G$ be any two elements Then x= a^m , y = a^k for some integers m and k x * y = $a^m * a^k = a^{m+k}$ $= a^{k+m}$ $= a^k * a^m$ = y * x Therefore x*y = y*x, for all $x,y \in G$ Therefore G is abelian.
16	**Define Group homomorphism with an example. (Nov 2014) (BTL1)** Let (G, *) and $(G', \bullet)$ be two groups. A mapping $f : G \to G'$ is called a group homomorphism if for all $a,b \in G$, f(a*b) = f(a) • f(b). **Example:** Consider the group (R, +) and $(R^*, \bullet)$ where R^* = R-{0}. Let $f : R \to R^*$ be defined by $f(a) = 2^a \forall a \in R$. Then f is a homomorphism.
17	**Define Kernal of a homomorphism in a group. (Nov 2017) (BTL1)** Let (G, *) and $(G', \bullet)$ be groups with e' as the identity element of G' . Let $f : G \to G'$ be a

	homomorphism. The ker $f = \{a \in G / f(a) = e'\}$
18	**Define Rings. (BTL1)** A non-empty set R with two binary operations denoted by „+" and **„."** is called a ring if (1) (R,+) is an abelian group with 0 as identity (2) (R,**.**) is a semigroup (3) The operation **„."** is distributive over „+" (i.e.) a**.** (b+c) = a**.**b + a**.**c and (b+c) **.**a = b**.**a + c**.**a , for all a,b,c∈R
19	**Define a field in an algebraic system. (Apr 2015) (BTL1)** A commutative ring (R,+, **.**) with identity in which every non-zero element has a multiplicative inverse is called a field.
20	**Give an example of a ring which is not a field. (Nov 2013) (BTL3)** (Z,+, **.**) is a ring but not a field because integers does not contain its multiplicative inverse.
21	**If (R,+, .) is a ring then prove that a.0=0,** $\forall a \in R$ **and 0 is the identity element in R under addition. (Nov 2017) (BTL2)** **Given: (R,+,** .) is a ring **To Prove:** $a.0 = 0, \forall a \in R$ If $a \in R$ then $a.0 = a.(0+0)$ $= a.0 + a.0$ $\Rightarrow a.0 + 0 = a.0 + a.0$ $\Rightarrow 0 = a.0$ Similarly 0.a = (0+0).a = 0.a +0.a 0.a = 0
22	**Prove that if G is abelian, then** $\forall a,b \in G, (a*b)^2 = a^2 * b^2$ **. (May 2011, Nov 2010, May 2013) (BTL5)** **Given:** G is abelian **To Prove:** $(a*b)^2 = a^2 * b^2$ L.H.S = $(a*b)^2$ = (a*b) * (a*b) = a * ((b*a) *b) (since associativity) = a * ((a*b) * b) (since abelian)

	= a * (a * (b*b)) (since associavity) = (a*a) * (b*b) = $a^2 * b^2$
23	**Give an example of semi group but not a monoid. (BTL3)** The set of all positive integers over addition form a semi group but it is not a monoid because identity axiom is not satisfied.
24	**If „a‟ is a generator of a cyclic group G, then show that „a^{-1}‟ is also a generator of G. (BTL4)** **Given:** „a‟ is a generator of G **To prove:** a^{-1} is also a generator Let $G = a$ be a cyclic group generated by „a‟ If $x \in G$, then $x=a^n$ for some $n \in Z$ $\therefore x = a^n = (a^{-1})^{-n}, (-n \in Z)$ $\therefore$ a^{-1} is also a generator of G.
25	**Give an example to show that union of two subgroups need not be a subgroup. (BTL3)** We know that (Z,+) is a group Let $H_1=2z$ and $H_2=3z$ $\therefore$ $(H_1,+)$ and $(H_2,+)$ are subgroups of Z Now $2 \in H_1$ and $3 \in H_2$, $\therefore$ $2,3 \in H_1 \cup H_2$ But $2,3 \in H_1 \cup H_2$ $\because 5 \notin H_1$ *and* $5 \notin H_2$ So $H_1 \cup H_2$ is not a subgroup of G
	Part-B
1	**Show that M_2, the set of all 2x2 non-singualar matrices over R is a group under usual matrix multiplication. Is it abelian? (Apr 2015) (BTL5) (8 Marks)** (Refer SKD pg.4.38) Keypoints: • Assume a 2x2 matrix (1mark) • closure $\|AB\| = \|A\|\|B\|$ (1mark) • Associative A(BC)=(AB)C (2mark)

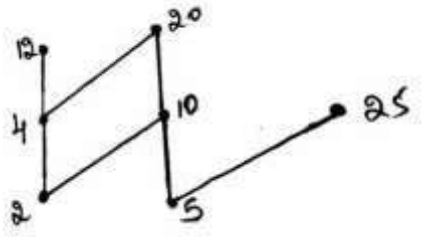
	• Identity $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ (2mark) • Inverse $A^{-1} = \frac{1}{\lvert A \rvert} adjA$ (1mark) • Commutative is not satisfied. (1mark)
2	**Show that (Q^+,*) is an abelian group where * is defined by $a * b = \frac{ab}{2}, \forall a,b \in Q^+$. (Nov 2016, Apr 2018) (BTL5) (8 Marks)** (Refer SKD Pg.4.17) Keypoints: • Closure $a*b \in G$ (1mark) • Associative $a*(b*c)=(a*b)*c$ (2marks) • Identity e=2 (2marks) • Inverse $\frac{4}{a}$ (2marks) • Commutative a*b =b*a (1mark)
3	**Prove that** $\left\{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}\right\}$ **forms an abelian group under matrix multiplication. (Nov 2015) (BTL5) (8 Marks)** (Refer SKD Pg. 4.15) Keypoints: • Closure : all the elements of G are closed under multiplication (1 mark) • Associative : Matrix multiplication is always associative (2marks) • Identity: I is the identity element (1mark) • Inverse : Inverse of I is I, Inverse of A is A, Inverse of B is B, inverse of C is C (2marks) • Prove commutative.(2marks)
4	**Prove that every cyclic group is an abelian group. (Nov 2013) (BTL5) (8 Marks)** (Refer Balaji Pg. 4.54) Keypoints: • Consider a cyclic group generated by a. (2marks) • Take $x = a^n\ y = a^m$ (2marks)

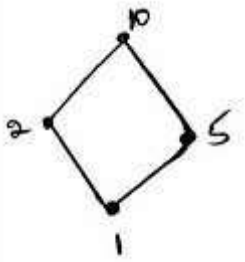
	• prove its abelian : x*y = y*x (4marks)
5	**Prove that intersection of any two subgroups of a group (G,*) is again a subgroup of (G,*). (May 2013,Nov 2013, Nov 2015) (BTL5) (8 Marks)** (Refer Balaji Pg. 4.56) Keypoints: • Consider two subgroups H_1 and H_2 with same elements in both the groups. (2marks) • $a*b^{-1} \in H$, $a*b^{-1} \in k$ (2marks) • $a * b^{-1} \in H \cap K$ (4marks)
6	**Show that union of two subgroups of a group G is a subgroup of G iff one is contained in the other. (Apr 2015, Nov 2014) (BTL5) (8 Marks)** (Refer Balaji 4.56) Keypoints: • Consider union of two subgroups (2marks) • Prove by contrary (3marks) • Prove the converse by considering $H_1 \subseteq H_2 \text{ or } H_2 \subseteq H_1$ (3marks)
7	**State and Prove Lagrange"s theorem for groups. Is the converse true? (May 2015, May 2016, Nov 2016, May 2018, May 2017) (BTL5) (16 Marks)** (Refer Balaji 4.68) Keypoints: • Prove the theorem " Let (H , *) be a subgroup of (G, *). Then the set of all left cosets of H in G form a partition of G. That is every element of G belongs to only one left coset of H in G". (4marks) • Prove the theorem " There is a 1-1 correspondence between any two left coset of H in G".(4marks) • Using the above two theorems prove order of H divides order of G (4marks) • Check if the converse is true. (4marks)
8	**If S=NxN, the set of ordered pairs of positive integers with the operation * defined by (a,b)*(c,d)=(ad+bc,bd) and if $f : (S,*) \to (Q,+)$ is defined by $f(a,b) = \frac{a}{b}$, show that f is a semigroup homomorphism. (May 2008, Nov 2014) (BTL5) (8 Marks)** (Refer SKD Pg.4.109)

	Keypoints: • Check closure : $a * b \in G$ (3marks) • Associative a*(b*c)=(a*b)*c (3marks) • Check f(x*y) = f(x) + f(y) (2marks)
9	**Show that a semigroup with more than one idempotent element cannot be a group. Give an example of a semigroup which is not a group. (Nov 2014) (BTL5) (8 Marks)** (Refer Balaji 4.17) Keypoints: • Consider two idempotent elements a*a=a , b*b=b(2marks) • Prove by contradiction (4marks) • Give an example (2marks)
10	**Prove that every subgroup of a cyclic group is cyclic. (May 2016, May 2017) (BTL5) (8 Marks)** (Refer SKD Pg.4.56) Keypoints: • Consider a cyclic group generated by a. (2marks) • Consider a subgroup H of G (2marks) • Prove that H is a cyclic group generated by a^m , $x=(a^m)^q$ (4marks)
11	**In any group** $G,*$ **show that** $(a * b)^{-1} = b^{-1} * a^{-1}, \forall a,b \in G$ **. (May 2016) (BTL5) (8 Marks)** (Refer Balaji Pg. 4.35) Keypoints: • Consider two elements in the group G (2marks) • Its inverse also exists in G (2marks) • $(a*b)*(b^{-1}*a^{-1}) = (b^{-1}*a^{-1})*(a*b) = e$ (4marks)
12	**Prove that kernel of a group homomorphism is a normal subgroup of the group. (May2017, May 2016, May 2018) (BTL5) (8 Marks)** (Refer Balaji Pg.4.69**)** Keypoints: • Consider a kernel of the homomorphism (1mark) • Consider two elements in kerf (2marks) • Prove that kerf is a subgroup of G (i.e.,) $x * y^{-1} \in Kerf$ (3marks)

	• Prove that kerf is normal (i.e.,) $f * x * f^{-1} \in Kerf$ (2marks)
13	**Prove that intersection of two normal subgroups of a group G is again a normal subgroup of G. (Nov 2016, Apr 2018) (BTL5) (8 Marks)** (Refer Balaji Pg. 4.71) Keypoints: • Consider two normal subgroups N_1 and N_2 (2marks) • $ab^{-1} \in N_1 \cap N_2$ • Prove that $ana^{-1} \in N_1 \cap N_2$ (6marks)
14	**State and prove Cayley"s theorem. (May 2013) (BTL5) (8 Marks)** **(**Refer Balaji Pg. 4.59) Keypoints: • " Every finite group of order n is isomorphic to a permutation group of order n" (2marks) • Define a mapping $f : G \to G$ (1mark) • Find 1-1 $f_a(x)=f_a(y) \Rightarrow x=y$ (1mark) • onto $if\ y \in G\ , y = f_a(a_{-1} * y)$(1mark) • Consider a set G' , prove that it"s a group (2marks) • Prove G is isomorphic to G' . (1mark)
15	**Let $f : (G,*) \to (G',\bullet)$ be a group homomorphism then prove that** **(1)** $[f(a)]^{-1} = f(a^{-1}), \forall a \in G$ **(2) f(e) is an identity of G' , when e is an identity element of G. (Nov 2015) (BTL1) (8 Marks)** (Refer SKD Pg. 4.80) Keypoints: • (i) f(a).f(e)=f(a).e" (4marks) • (ii) $f(a^{-1} * a) = f(e) \Rightarrow f(a^{-1}).\ f(a) = e'$ (4marks)
16	**State and prove fundamental theorem on group homomorphism of groups. (May 2011, Nov 2013) (8 Marks)** (Refer Balaji Pg. 4.70) Keypoints: • "Let (G,*) and $(G',\bullet)$ be two groups. Let $f : G \to G'$ be a homomorphism of groups with

	kernel K. Then G/K is isomorphic to $f(G) \subseteq G'$ (2marks) • Consider a mapping (1mark) • Prove that it is well defined : If ak=bk then f(a)= f(b) (1mark) • 1-1 and onto (2marks) • Prove that it is a homomorphism : $\phi(ak \oplus bk) = \phi(ak) \bullet \phi(bk)$ (2marks)
17	**Prove that Z_4= {0,1,2,3} is a commutative ring with respect to the binary operation $+_4$ and x_4. (Nov 2015). (BTL5) (8 Marks)** Keypoints: • Check if Z_4 is an abelian group over + (2marks) • Check if Z_4 is a semigroup over x (2marks) • Prove that x is distributive over + (2marks) • Check if Z_4 is commutative (2marks)
	UNIT V – LATTICES AND BOOLEAN ALGEBRA
	Partial ordering – Posets – Lattices as posets – Properties of lattices - Lattices as algebraic systems – Sub lattices – Direct product and homomorphism – Some special lattices – Boolean algebra.
Q.No.	**PART-A**
1.	**Define Partial order relation and give an example . (BTL1)** A relation R on A is called partial order relation if R is reflexive, antisymmetric and transitive **Example:** set of positive integers
2	**Define a lattice. Give suitable example. (Nov 2014, Nov 2015, Nov 2016) (BTL1)** A lattice is a poset $(L, \leq)$ in which every pair of elements $a,b \in L$ has a greatest lower bound and least upper bound. **Example:** $(Z^+, \leq)$ where $\leq$ denotes divisibility is a lattice. The poset N with the usual $\leq$ is a lattice if $a,b \in N$ then $a \vee b = Max\{a,b\}$ and $a \wedge b = Min\{a,b\}$
3	**Define distributive lattice. (BTL1)** A lattice $(L,\wedge,\vee)$ is said to be distributive if $\wedge$ and $\vee$ satisfies the following conditions, $\forall a,b,c \in L$ $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$

4	**State modular lattice. (BTL1)** A lattice $(L,\wedge,\vee)$ is said to be modular lattice if it satisfies the following condition If $a \le c$ then $a \vee (b \wedge c) = (a \vee b) \wedge c$, $\forall a,b,c \in L$
5	**Define Complete lattice. (BTL1)** A lattice $(L,\wedge,\vee)$ is said to be complete if every non-empty subset has a least upper bound and greatest lower bound **Example:** Every finite lattice L is complete
6	**Define bounded lattice. (BTL1)** A lattice $(L,\wedge,\vee)$ is said to be bounded if it has a greatest element 1 and a least element 0. (i.e.) $0 \le a \le 1, \forall a \in L$
7	**Define complemented lattice. (BTL1)** A bounded lattice $(L,\wedge,\vee,0,1)$ is said to be complemented , if every element of L has atleast one complement.
8	**Define lattice homomorphism. (Apr 2015) (BTL1)** Let $(L,*,\oplus)$ and $(M,\wedge,\vee)$ be two lattices. A mapping $f: L \to M$ is called a lattice homomorphism from the lattice $(L,*,\oplus)$ to the lattice $(M,\wedge,\vee)$ if $f(a * b) = f(a) \wedge f(b)$ and $f(a \oplus b) = f(a) \vee f(b)$.
9	**State modular inequality in lattices. (Nov 2017)** (BTL1) If $(L,\wedge,\vee)$ is a lattice , then $a \le c \Leftrightarrow a \vee (b \wedge c) = (a \vee b) \wedge c$, $\forall a,b,c \in L$.
10	**Draw the Hasse diagram of** $(X, \le)$ **where X={2,4,5,10,12,20,25} and the relation** $\le$ **be such that** $x \le y$ **if x divides y. (Nov 2013)(BTL4)** **Hasse Diagram:** 
11	**Let A={1,2,5,10} with the relation divide. Draw the Hasse diagram. (Nov 2015) (BTL4)**

	
12	**Define Boolean algebra. (Nov 2007, May 2010) (BTL1)** A boolean algebra is a complemented distributive lattice. A non-empty set B together with two binary operations „+" , „." on B , a unary operation on B ' called complementation and two distinct elements 0 and 1 is called a Boolean algebra if the following axioms are satisfied for all $a,b,c \in B$. **Commutative Law:** a+b = b+a and a.b=b.a **Associative Law:** a + (b + c) = (a + b) + c and a . (b . c) = (a . b) . c **Distributive Law:** a + (b . c) = (a + b) . (a + c) and a . (b + c) = (a . b) + (a . c) **Identity Law:** There exists $0,1 \in B$ such that a + 0 = a and a . 1 = a **Complement Law:** For each $a \in B$ there exists an element $a' \in B$ such that $a + a' = 1$ and $a.a' = 0$ The Boolean algebra is usually denoted as 6-tuple $(B, +, .,', 0, 1)$.
13	**State the De Morgan"s law in a Boolean algebra. (Nov 2016)** $(i)\ (a + b)' = a'.b'$ $(ii)(a.b)' = a' + b'\ , \forall a,b \in B$
14	**Show that Absorbtion laws are valid in a Boolean algebra. (May 2016, May 2017) (BTL5)** **The absorbtion laws are** **(i) a . (a + b) = a** **(ii) a + a . b = a** $\forall a,b \in B$ (i) L.H.S=a. (a + b) = (a + 0) . (a + b) (by identity law) = a + (0 . b) (by distributive law) = a + (b . 0) (by commutative law) = a + 0 (by boundedness law) = a (by identity law) =R.H.S (ii) L.H.S = a + (a . b) = (a . 1) + (a . b) (by identity law)

	= a . (1 + b) (by distributive law) = a . (b + 1) (by commutative law) = a . 1 (by bounded law) = a (by identity law) =R.H.S
15	**Prove the Boolean identity** $a.b + a.b' = a$ **(May 2015) (BTL5)** L.H.S = $a.b+a.b'=a.(b+b')$ (by distributive law) $=a.1$ $(b+b'=1)$ = a = R.H.S
16	**Is there a Boolean algebra with 5 elements? Jsutify your answer. (Nov 2013) (BTL4)** Since each Boolean algebra must have 2^n elements for some integer n. Here $5 \neq 2^n$ for some integer n Hence there is no Boolean algebra having 5 elements.
17	**Let X={1,2,3,4,5} and R be a relation defined as** $x, y \in R$ **if and only if x-y is divisible by 3. Find the elements of the relation R. (Apr2016, May 2017) (BTL3)** **Given:** X={1,2,3,4,5} The relation R is defined as x-y divisible by 3. $\therefore R\{1,4\ ,\ 2,5\ ,\ 3,6\}$
18	**Does Boolean algebra contain 6 elements? Justify. (Nov 2015) (BTL1)** Since each Boolean algebra must have 2^n elements for some integer n. Here $6 = 2^n$ for some integer n Hence there is Boolean algebra having 6 elements.
19	**Define sublattice . (BTL1)** Let $(L,\wedge,\vee)$ be a lattice and $S \subseteq L$, be a subset of L, then $(S,\wedge,\vee)$ is a sublattice of $(L,\wedge,\vee)$ if S is closed under the operation $\wedge$ *and* $\vee$.
20	**Show that a chain of three or four elements is not complemented. (BTL4)** Let $(L,\wedge,\vee)$ be a given chain We know that, in a chain any two elements are comparable.

	Let 0,x,1 be any three elements of $(L,\wedge,\vee)$ with 0 is the least element and 1 is the greatest element We have $0 \le x \le 1$ Now $0 \wedge x = 0$ *and* $0 \vee x = x$ $1 \wedge x = x$ *and* $1 \vee x = 1$ In both cases, x does not have any complement. Hence any chain with 3 or more elements is not complemented.
21	**In a Boolean algebra, show that** $ab' + a'b = 0$ *if* $a = b$ **. (BTL4)** Let $(B, +, .,', 0, 1)$ be a Boolean algebra Let $a,b \in B$ be any two elements $ab' + a'b = aa' + a'a$ Let a=b then $= 0 + 0$ $= 0$
22	**Let A={a,b,c} and P(A) is a Poset, Draw a Hasse diagram of** $(P(A),\subseteq)$ **. (BTL2)** **Given:** A = {a,b,c} P(A) is the set of all subsets of A P(A) = { ϕ,{a}, {b},{c}, {a,b}, {a,c}, {b,c}, {a,b,c}} Since empty set is a subset of every set in P(A), ϕ is the least of P(A) Similarly A={a,b,c} contains all elements of P(A). Therefore A is the greatest element in P(A) Hence every pair of elements of P(A) has L.U.B and G.L.B. Therefore $(P(A),\subseteq)$ is a lattice.
23	**Show that every distributive lattice is modular. Is the converse true? Justify. (BTL4)** Let $(L,\wedge,\vee)$ be the given distributive lattice $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ holds good for all $a,b,c \in L$ ------ (1)

	Now if $a \le c$ *then* $a \vee c = c$ ----------(2) From (1) $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ $= (a \vee b) \wedge c$ $(by\,(2)$ Therefore every distributive lattice is modular But the converse is not true. (i.e.) Every modular lattice need not be distributive. For example diamond lattice M_5 is modular but not distributive.
24	**Is a chain a modular lattice? Justify. (BTL5)** Since any chain is a distributive lattice By theorem , Every distributive lattice is modular Hence every chain is a modular lattice.
25	**In any Boolean algebra show that if** $a = 0$ *then* $ab' + a'b = b$ **(BTL5)** Let $(B, +, .,', 0, 1)$ be a Boolean algebra Let $a, b \in B$ be any two elements If a=0 then $ab' + a'b = 0 + a'b$ $= 0 + 1.b$ $= 0 + b$ $= b$
	PART-B
1	**Prove that every chain is a distributive lattice. (Nov2013, Apr2015, May2016, Apr2017, Nov2017, Nov 2016) (BTL5) (8 Marks)** (Refer Balaji Pg. 5.22) Keypoints: • Consider a chain with two elements (2marks) • Consider two cases $a \le b and\ b \le a$ (3marks) • Prove that GLB and LUB exists which proves that a chain is a lattice(3marks) • Prove that a chain is distributive: $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
2	**State and prove De Morgan‟s law in a complemented distributed lattice. (Apr2015) (BTL5)** (Refer Balaji 5.27) Keypoints:

	• $(a+b)' = a'.b'$ (2marks) • Prove that : $(a \wedge b) \wedge (a' \vee b') = 0$, $(a \wedge b) \vee (a' \vee b') = 1$ (2marks) • $(a.b)' = a' + b'$ (2marks) • Prove that : $(a \vee b) \wedge (a' \wedge b') = 0$, $(a \vee b) \vee (a' \wedge b') = 1$ (2marks)
3	**In a distributive complemented lattice , show that the following are equivalent** **(i)** $a \le b$ **(ii)** $a \wedge b = 0$ **(iii)** $a \vee b = 1$ **(iv)** $b \le a$ **. (May2016, May2017 Nov 2017) (BTL5) (8 Marks)** (Refer Balaji 5.25) Keypoints: • $a \le b \Rightarrow a \wedge b = 0$ (2marks) • $a \wedge b = 0 \Rightarrow a \vee b = 1$ (2marks) • $a \vee b = 1 \Rightarrow b \le a$ (2marks) • $b \le a \Rightarrow a \le b$ (2marks)
4	**Show that every ordered lattice $(L, \le)$ satisfies the following properties of the algebraic lattice , (i) idempotent (ii) commutative (iii) Associative** **(iii) Absorption. (Apr 2017) (BTL5) (8 Marks)** (Refer Balaji 5.13) Keypoints: • To prove idempotent : $a \vee a = a$ & $a \wedge a = a$ (2marks) • Prove associative: $a \vee (b \vee c) = (a \vee b) \vee c$ & $a \wedge (b \wedge c) = (a \wedge b) \wedge c$ (3marks) • Prove absorption : $a \vee (a \wedge b) = a$ & $a \wedge (a \vee b) = a$ (3marks)
5	**Show that $(N, \le)$ is a partially ordered set, where N is the set of all positive integers and $\le$ is a relation defined by $m \subseteq n$ iff n-m is a non-negative integer. (Apr 2018) (BTL5) (8 Marks)** (Refer SKD Pg. 5.9) Keypoints: • Prove the $\le$ is reflexive: $\forall x \in N$, xRx (2marks)

	• Anti symmetric : xRy , $yRx \Rightarrow x = y$ (3marks) • Transitive: xRy & $yRz \Rightarrow xRz$ (3marks)
6	**In a complemented distributive lattice, prove that complement of each element is unique. (Nov 2015, Apr 2018) (BTL5) (8 Marks)** (Refer Balaji Pg. 5.32) Keypoints: • Consider a distributive lattice $(L,\vee,\wedge,0,1)$, then $a \wedge x = 0$, $a \vee x = 1$, if x is a compliment of „a". similarly for y(2marks) • Prove : $x = x \vee y$ (2marks) • Prove : $y = x \vee y$ (4marks)
7	**Show that every chain is modular. (May 2016) (BTL5) (8 Marks)** (Refer SKD Pg.5.52) Keypoints: • Prove every chain is a distributive lattice(Check problem 1) (4marks) • Prove every distributive lattice is modular: If $a \le c \Rightarrow a \vee (b \wedge c) = (a \vee b) \wedge c$ (4marks)
8	**Let $(L, \le)$ be a lattice, in which * and $\oplus$ denote the operation of meet and join respectively. For any $a, b \in L$, $a \le b \Leftrightarrow a * b = a \Leftrightarrow a \oplus b = b$. (Nov 2017) (8 Marks)** (Refer Balaji Pg. 5.14) **(BTL4)** Keypoints: • Prove $a \le b \Leftrightarrow a * b = a$ (3marks) • Prove $a * b = a \Leftrightarrow a \oplus b = b$ (3marks) • Prove $a \oplus b = b \Leftrightarrow a \le b$ (2marks)
9	**Let $(L,\wedge,\vee, \le)$ be a distributive lattice and $a,b \in L$ if $a \wedge b = a \wedge c$ and $a \vee b = a \vee c$ then show that b=c. (Apr 2018) (BTL5) (8 Marks)** (Refer Balaji Pg. 5.23) Keypoints: • $a \vee (b \wedge c)$ = c (4marks) • $a \wedge (b \vee c)$ =b (4marks)
10	**Prove that the diamond lattice is distributive or not. (Nov 2015) (BTL5) (8 Marks)**

	(Refer Balaji Pg. 5.24) Keypoints: • Draw the diamond lattice (2marks) • Consider case (i) as (0,b,a) get the answer as 0 (1mark) • Consider case (ii) as (0,1,a) get the answer as a (1mark) • Consider case (iii) as (0,a,1) get the answer as a (1mark) • Consider case (iv) as (a,0,1) get the answer as a (1mark) • Consider case (v) as (a,b,1) get the answer as 1 (1mark) • Conclude with the following cases (1mark)
11	**Let D_{30} = {1,2,3,5,6,10,15,30} with a relation $x \le y$ iff x divides y. Find** **(i) All lower bounds of 10 and 15** **(ii) All G.L.B of 10 and 15** **(iii) All upper bounds of 10 and 15** **(iv) All L.U.B of 10 and 15** **(v) Hasse diagram of D_{30} (Nov2015, Apr 2018) (BTL5) (8 Marks)** Keypoints: • Draw the hasse diagram (4marks) • Find the GLB and LUB (4marks)
12	**Show that in a lattice if $a \le b \le c$ then** **(1)** $a \oplus b = b^* c$ **(or)** $a \vee b = b \wedge c$ **(2)** $(a^* b) \oplus (b^* c) = b = (a \oplus b)^* (a \oplus c)$ **(or)** $(a \wedge b) \vee (b \wedge c) = b = (a \vee b) \wedge (a \vee c)$ **. (Nov 2013) (BTL5) (8 Marks)** (Refer Balaji Pg. 5.18) Keypoints: • Using $a \le b \le c$ prove (1) (4marks) • Using necessary laws prove (2) , $(a^* b) \oplus (b^* c) = b = (a \oplus b)^* (a \oplus c)$ (4marks)
13	**If S_n is the set of all divisors of the positive integers n and D is the relation of division, prove that $\{S_{30}, D\}$ is a lattice. Find also all the sublattices of $\{S_{30}, D\}$ that contains six or more elements. (Apr 2015) (BTL5) (8 Marks)** (Refer Balaji Pg. 5.30) Keypoints:

	• Draw the Hasse diagram (3marks) • Find GLB and LUB (2marks) • Find all the sub lattices that contain 6 or more elements(3marks)
14	**Show that the De Morgan"s law holds in a Boolean algebra. (Nov 2014, May 2016) (BTL5) (8 Marks)** (Refer Balaji Pg. 5.39) Keypoints: • $(a+b)' = a'.b'$ (2marks) • Prove: $\begin{matrix}(a+b)+(a'.b')=1\\(a+b).(a'.b')=0\end{matrix}$ (2marks) • $(a.b)' = a'+b'$ (2marks) • Prove: $\begin{matrix}(a.b)+(a'+b')=1\\(a.b).(a'+b')=0\end{matrix}$ (2marks)
15	**In any Boolean algebra show that** $(a+b')\,(b+c')\,(c+a') = (a'+b)(b'+c)\,(c'+a)$ **. (Nov 2013) (BTL5) (8 Marks)** (Refer Balaji Pg. 5.50) Keypoints: • Consider LHS = $(a+b')(b+c')(c+a')$ (4marks) • prove the RHS = $(a'+b)(b'+c)(c'+a)$ (4marks)
16	**If P(S) is the power set of a non-empty set S, prove that** $\{P(S), \cup, \cap, /, \phi, S\}$ **is a Boolean algebra. (Nov 2015) (BTL2) (8 Marks)** (Refer Balaji Pg. 5.41) Keypoints: • Consider elements from P(A) (2marks) • prove that the given set is a Boolean algebra (6marks)
17	**If** $a,b \in S = \{1,2,3,6\}$ **and a+b = LCM(a,b) , a*b = GCD(a,b) and** $a' = \frac{6}{a}$ **, show that** $(B, +, .,', 1, 6)$ **is a Boolean algebra. (BTL3) (8 Marks)** Keypoints: • Prove Commutative, Associative, (3marks) • Distributive, Identity (3marks)

	• Complement. (2marks)

CS8351DIGITAL PRINCIPLES AND SYSTEM DESIGNL T P C4 0 04

OBJECTIVES:

- To design digital circuits using simplified Boolean functions
- To analyze and design combinational circuits
- To analyze and design synchronous and asynchronous sequential circuits
- To understand Programmable Logic Devices
- To write HDL code for combinational and sequential circuits

UNIT I - BOOLEAN ALGEBRA AND LOGIC GATES **12**

Number Systems - Arithmetic Operations - Binary Codes- Boolean Algebra and Logic Gates - Theorems and Properties of Boolean Algebra - Boolean Functions - Canonical and Standard Forms - Simplification of Boolean Functions using Karnaugh Map - Logic Gates – NAND and NOR Implementations.

UNIT II -COMBINATIONAL LOGIC **12**

Combinational Circuits – Analysis and Design Procedures - Binary Adder-Subtractor - Decimal Adder - Binary Multiplier - Magnitude Comparator - Decoders – Encoders – Multiplexers - Introduction to HDL – HDL Models of Combinational circuits.

UNIT III - SYNCHRONOUS SEQUENTIAL LOGIC **12**

Sequential Circuits - Storage Elements: Latches , Flip-Flops - Analysis of Clocked Sequential Circuits - State Reduction and Assignment - Design Procedure - Registers and Counters - HDL Models of Sequential Circuits.

UNIT IV ASYNCHRONOUS SEQUENTIAL LOGIC **12**

Analysis and Design of Asynchronous Sequential Circuits – Reduction of State and Flow Tables – Race-free State Assignment – Hazards.

UNIT V MEMORY AND PROGRAMMABLE LOGIC **12**

RAM – Memory Decoding – Error Detection and Correction - ROM - Programmable Logic Array – Programmable Array Logic – Sequential Programmable Devices.

TOTAL: 60 PERIODS

OUTCOMES:

After studying this course, the student should be able to:

- Simplify Boolean functions using KMap
- Design and Analyze Combinational and Sequential Circuits
- Implement designs using Programmable Logic Devices
- Write HDL code for combinational and Sequential Circuits

TEXT BOOKS:

M. Morris R. Mano, Michael D. Ciletti, ―Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog‖, 6th Edition, Pearson Education, 2017. (All five units)

REFERENCES

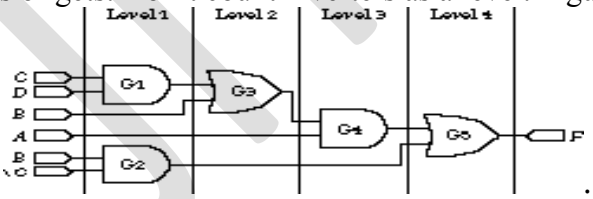
1. G. K. Kharate, Digital Electronics, Oxford University Press, 2010
2. John F. Wakerly, Digital Design Principles and Practices, Fifth Edition, Pearson Education, 2017.
3. Charles H. Roth Jr, Larry L. Kinney, Fundamentals of Logic Design, Sixth Edition, CENGAGE Learning, 2013
4. Donald D. Givone, Digital Principles and Design‖, Tata McGraw Hill, 2003.

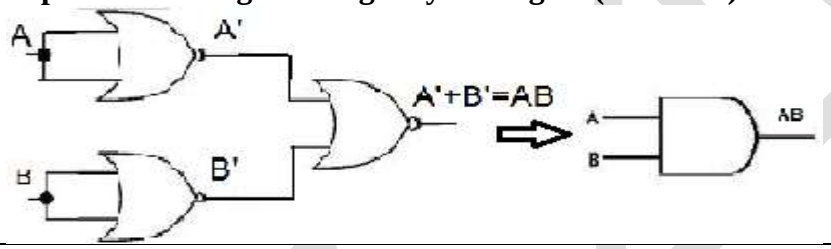
Subject Code: CS8351 **Year/Semester: II /04**
Subject Name: DIGITAL PRINCIPLES AND SYSTEM DESIGN **Subject Handler: R.R.Uma**

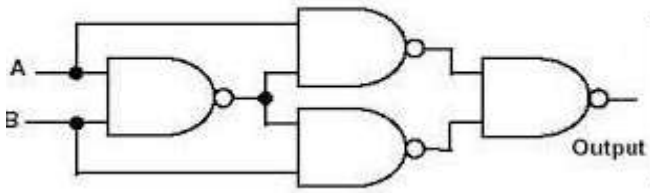
UNIT I-BOOLEAN ALGEBRA AND LOGIC GATES

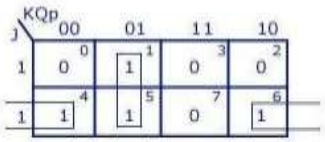
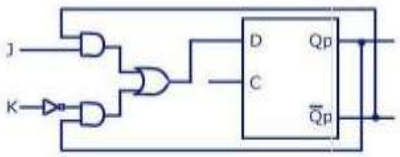
Number Systems - Arithmetic Operations - Binary Codes- Boolean Algebra and Logic Gates - Theorems and Properties of Boolean Algebra - Boolean Functions - Canonical and Standard Forms - Simplification of Boolean Functions using Karnaugh Map - Logic Gates – NAND and NOR Implementations.

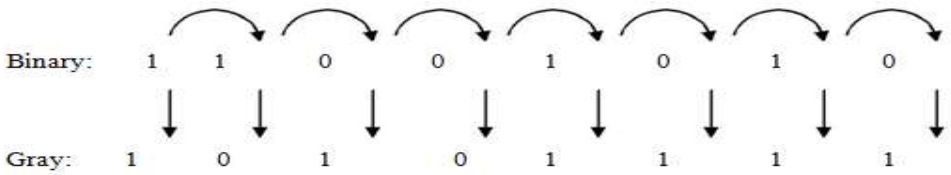
PART A

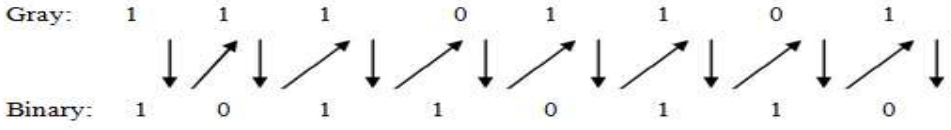
Q. No	Question& Answer
1	**Simplify the following Boolean function (Apr/May 2019)** BTL 1 F=x'y'+xy+x'y =x'(y'+y)+xy =x'+xy =(x'+x)(x'+y) =x'+y
2	**Find the Octal equivalent of the hexadecimal number DC.BA. (Apr/May 2019)** BTL 1 DC. BA_{16} = 11011100.10111010_2 = 334.564_8
3	**State the classification of binary codes. (Nov/Dec 2018)** BTL 1 • Weighted Codes • Non-Weighted Codes • Binary Coded Decimal Code • Alphanumeric Codes • Error Detecting Codes • Error Correcting Codes
4	**Define associative law. (Nov/Dec 2018)** BTL 1 a+(b+c)=(a+b)+c a.(b.c)=(a.b).c
5	**What is meant by multilevel gates networks? (May/June 2016)** BTL 1 A number of gates cascaded in series between a network input and output is referred to as the number of levels of gets. Don't count inverters as a level. Figure shows 4 level networks. 
6	**Discuss the NOR operation with a truth table. (Nov/Dec 2015)** BTL 1 This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if any of the inputs are high. **2 Input NOR gate** A | B | $\overline{A+B}$ 0 | 0 | 1 0 | 1 | 0 1 | 0 | 0 1 | 1 | 0

7	**Write short notes on weighted binary codes. (Nov/Dec 2015)** BTL 1 Weighted binary codes are those binary codes which obey the positional weight principle. Each position of the number represents a specific weight. Several systems of the codes are used to express the decimal digits 0 through 9.
8	**Convert $(126)_{10}$ to Octal number and binary number. (Nov/Dec 2015)** BTL 1 $126_{10} = 1111110_2$ & 176_8
9	**Prove the following using Demorgan' theorem [(X+Y)'+(X+Y)']'= X+Y (May 2015)** BTL 1 = [(X+Y)'+(X+Y)']' = X+Y''. X+Y'' = (X+Y). (X+Y) = X+Y
10	**Convert $(0.6875)_{10}$ to binary. (May 2015)** BTL 1 INTEGER FRACTION COEFFICIENTS – 0.6875 x 2 = 1 + 0.3750 $a_1 = 1$ – 0.3750 x 2 = 0 + 0.7500 $a_2 = 0$ – 0.7500 x 2 = 1 + 0.5000 $a_3 = 1$ – 0.5000 x 2 = 1 + 0.0000 $a_4 = 1$ Answer: $(0.6875)_{10} = (0.a_1a_2a_3a_4)_2 = (0.1011)_2$
11	**Implement AND gate using only NOR gate (Dec 2014)** BTL 1 
12	**State the principle of duality (Dec 2014)** BTL 1 The duality theorem states that starting with a Boolean relation we can drive another Boolean relation by changing OR operation i.e., + sign to an and operation i.e., dot and vice versa. Complement any 0 and 1appearing in the expression i.e., replacing contains 0 and 1 by 1 and o respectively
13	**State and prove the consensus theorem. (June 2014)** BTL 1 Theorem: AB+A'C+BC = AB+A'C **Proof:** AB+A'C+BC = AB+A'C+BC.1 =AB+A'C+BC(A+A) = AB+A'C+ABC+A'BC = AB(1+C) + A'C(1+B) = AB+A'C
14	**Find the octal equivalent of hexadecimal numbers AB.CD. (June 2014)** BTL 1 (i) Convert the hexadecimal to binary equivalent $(AB.CD)_{16}=(1010\ 1011.1100\ 1101)_2$ (ii) Then convert binary equivalent to octal number $(10101.1100\ 1101)_2=(\mathbf{253.315})_8$
15	**Realize XOR gate using only 4 NAND gates. (Dec 2013)** BTL 1

<table>
<tr><td></td><td>
Exclusive OR (XOR)


<table>
<tr><th>A</th><th>B</th><th>Output</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>16</td><td>
Realize JK flip flop using D flip flop. (Dec 2013) BTL 1
In this conversion, D is the actual input to the flip flop and J and K are the external inputs. J, K and Qp make eight possible combinations, as shown in the conversion table below. D is expressed in terms of J, K and Qp. The conversion table, the K-map for D in terms of J, K and Qp and the logic diagram showing the conversion from D to JK are given in the figure below.
D Flip Flop to J-K Flip Flop
Conversion Table
<table>
<tr><th colspan="2">J-K Input</th><th colspan="2">Outputs</th><th>D Input</th></tr>
<tr><th>J</th><th>K</th><th>Qp</th><th>Qp+1</th><th></th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
</table>
K-map

$D = J\overline{Q}p + \overline{K}Qp$
Logic Diagram


</td></tr>
<tr><td>17</td><td>
Convert the following hexadecimal numbers into decimal numbers: (Dec 2012) BTL 1
(a) 263 (b)1C3.
263H= 2x16x16+6x16+3x1=611
1C3H=1x16x16+12x16+3x1=451
</td></tr>
<tr><td>18</td><td>
What is the significance of BCD code. (Dec 2012) BTL 1
(i) Any large decimal number can be easily converted into corresponding binary number
(i) A person needs to remember only the binary equivalents of decimal number from 0 to 9.
Conversion from BCD into decimal is also very easy.
</td></tr>
<tr><td>19</td><td>
Convert $(11001010)_2$ into gray code. BTL 1
b) Convert a Gray code 11101101 into binary code.
(a) Binary to Gray code conversion:


(b) Gray to Binary code conversion:


</td></tr>
<tr><td>20</td><td>State & prove De-Morgan’s theorem.BTL 1</td></tr>
</table>

De-Morgan's theorem 1: The complement of product of any number of variables is equivalent to sum of the individual complements.

De-Morgan's theorem 2: The complement of sum of any number of variables is equivalent to product of the individual complements.

Proof:

a) (AB) '= A'+ B'

A	B	AB	**(AB)'**		A'	B'	**A'+B'**
0	0	0	**1**		1	1	**1**
0	1	0	**1**		1	0	**1**
1	0	0	**1**		0	1	**1**
1	1	1	**0**		0	0	**0**

b) (A+B)' = A'B'

A	B	A+B	**(A+B)'**		A'	B'	**A'B'**
0	0	0	**1**		1	1	**1**
0	1	1	**0**		1	0	**0**
1	0	1	**0**		0	1	**0**
1	1	1	**0**		0	0	**0**

21 **Describe the canonical forms of the Boolean function.** BTL 1

a) Sum of minterms: Combination of minterms using OR operation.

Minterm (standard product) is a combination of n variables using AND operation for the function of n variables.

Example for function of two variables A & B: $F = A'B + AB = m_1 + m_3$ $F = \sum m(1,3)$

b) Product of maxterms: Combination of maxterms using AND operation.

Maxterm (standard sum) is a combination of n variables using OR operation for the function of n variables. $F = (A+B)(A'+B) = M_0 M_2$ $F = \prod M(0,2)$

22 **Describe the importance of don't care conditions.** BTL 1

(i) Functions that have unspecified outputs for some input combinations are called incompletely specified functions. We simply don't care what value is assumed by the function for the unspecified minterms. (ii) The unspecified minterms are called don't care conditions. These don't care conditions can be used on a map to provide further simplification of the Boolean expression.

23 **Define the following: minterm and maxterm.** BTL 1

(i) Minterm(standard product) is a combination of n variables using AND operation for the function of n variables. Possible minterms for a function of two variables A & B: A'B', A'B, AB', AB

(ii) Maxterm(standard sum) is a combination of n variables using OR operation for the function of n variables. Possible maxterms for a function of two variables A & B: A+B, A+B', A'+B, A'+B'

24 **Minimize the function using K-map: F=∑m(1,2,3,5,6,7).** BTL 1

A \ BC	00	01	11	10
0	0 **0**	1 **1**	1 **3**	1 **2**
1	0 **4**	1 **5**	1 **7**	1 **6**

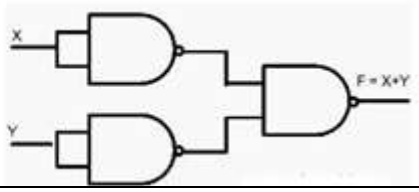
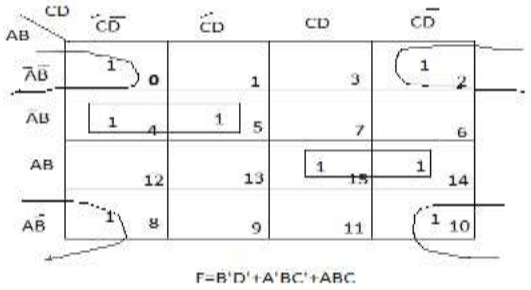
Quad (2,3,6,7) = B

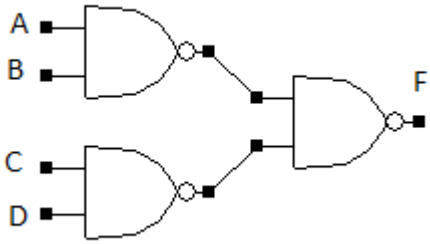
Quad (1,3,5,7) = C

F = B + C

25 **What are Universal Gates? Why are they called so?** BTL 1

A Universal gates are NAND and NOR, they are called so because using these codes any logical gate

	or logical expression can be derived.
26	**Implement OR using NAND only.** BTL 1 **Input** **Output** **Rule** ((XX)'(YY)')' = (X')' Idempotent = X''+Y'' DeMorgan = X+Y Involution 
	PART B
1	**(i) covert the following numbers to decimal (2M)** **(a) $(127.4)_8$, $(B65F)_{16}$** **(ii) Perform the following arithmetic operation using 2's complement 1010100 – 1000011 (4M)** **(iii) Express the following functions in sum of minterms and product of maxterm F(A,B,C,D)= AB+BD+AC' (6M) (Apr/May 2019)** BTL 5 **Ans: Refer Morris Mano, PG.NO: 21,31** (a) $(127.4)_8 = 1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} = (87.5)_{10}$ (3M) **(b)** $(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15 \times 16^0 = (46,687)_{10}$ $X = 1010100$ 2's complement of $Y = + 0111101$ Sum $= 10010001$ Discard end carry $2^7 = -10000000$ *Answer*: $X - Y = 0010001$ **(i)** (4M) **(ii)** F(A,B,C,D)= AB(C+C')(D+D')+(A+A')(C+C')BD+A(B+B')(D+D')C' (6M)
2.	**b) (i) Demonstrate by means of truth tables the Validity of the Demorgan's theorem for three variables:(XYZ)'=X'+Y'+Z' (4M)** **(ii) Simplify the following Boolean functions by means of a 4-variable K-map F(A,B,C,D)=∑m (0,2,4,5,8,10,14,15) (5M)** **(iii) Implement the following Boolean function only with NAND gate using a minimum number of gate inputs: F(A,B,C,D)=AB+CD (4M) (Apr/May 2019)** BTL 5 **Ans: Refer Salivahnan, PG.NO: 21,31** **(i) F(A,B,C,D)=∑m (0,2,4,5,8,10,14,15)**  (5M) (III) F(A,B,C,D)=AB+CD The minterms are ∑m(3,7,11,12,13,14,15)



(4M)

3 **Write short notes on Demorgan's theorem, Absorption law and consensus law (13M) (Nov/Dec 2018)** BTL 5

Ans: Refer Salivahnan, PG.NO: 42-43

Absorption Law

A+AB=A

A(A+B)=A

A+A'B=A+B

A.(A'+B)=AB (5M)

consensus law

AB+A'C+BC=AB+A'C

(A+B)(A'+C)(B+C)= (A+B)(A'+C) (3M)

Demorgan's theorem

(AB)'=A'+B'

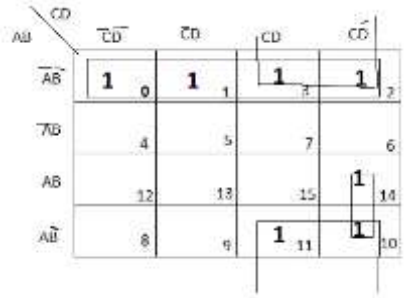
(A+B)'=A'.B' (3M)

4 **Convert the following boolean expression into standard SOP form**

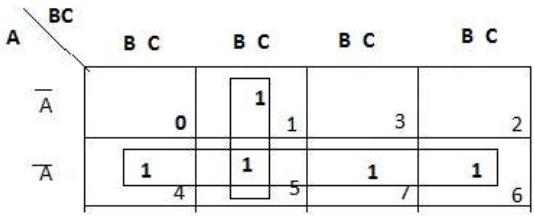
(a) AB'C+A'B'+ABC'D (6M)

(b) F=A+B'C (7M) (Nov/Dec 2018) BTL 5

(a) F(A,B,C,D)=∑(0,1,2,3,10,11,14)

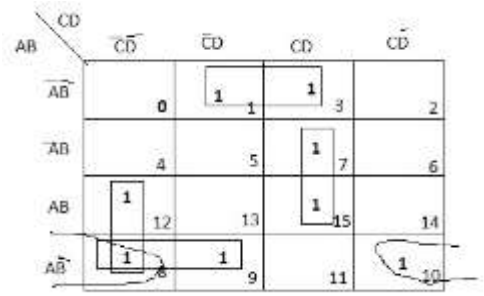


(b) F(A,B,C)=∑1,4,5,6,7) (3M)



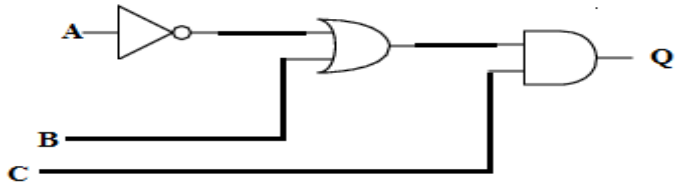
(5M)

5 **Simplify the following switching functions using Karnaugh map methodand realize expression using gates F(A,B,C,D) = Σ(0,3,5,7,8,9,10,12,15).(10M) (Nov/Dec 2015)** BTL 5

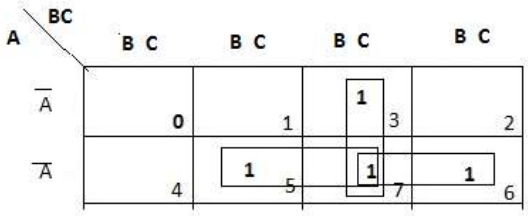


(10M)

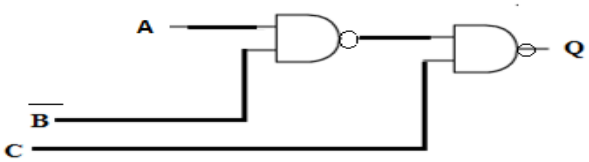
6 **(a) Express the following function in sum of min-terms and product of max-terms**
F(X,Y,Z)=X+YZ (May 2015) (5M)
(b) convert the following logic system into NAND gates only. (5M) (May 2015) BTL 5



(a) F(X,Y,Z)= ∑3,5,6,7 (2M)



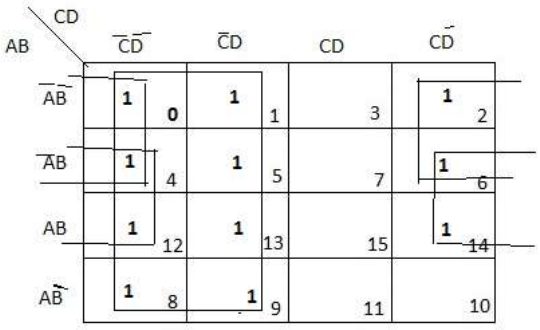
(3M)



(5M)

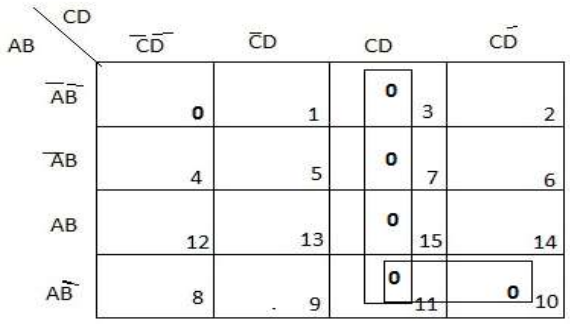
7 **Simply the following Boolean expression in (i) sum of product (ii) product of sum using k-map ∑(0,1,2,4,5,6,8,9,12,13,14) (10M) (May 2015)** BTL 5

(i) SOP



(5M)

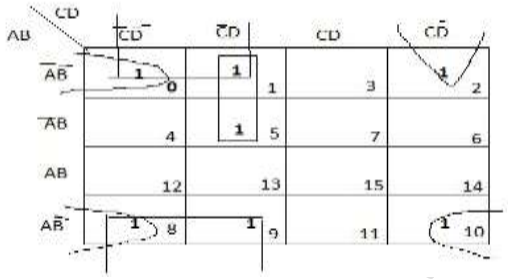
(ii) POS



(5M)

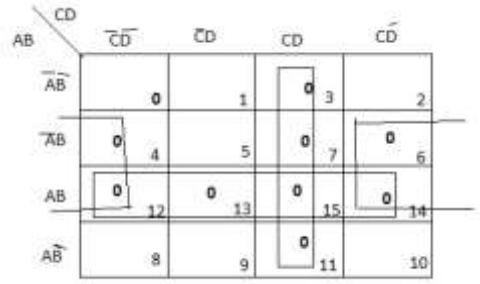
8 **Simplify the Boolean function in SOP and POS F(A,B,C,D)=∑m(0,1,2,5,8,9,10) (10M) (Dec2014) (ii) plot the following Boolean function in k-map and simplify it. F(w,x,y,z) = ∑m(0,1,2,4,5,6,8,9,12,13,14). (5M) (Dec2014)** BTL 5

(i) SOP



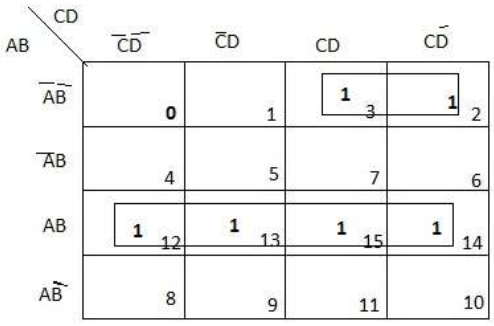
(5M)

(ii) POS



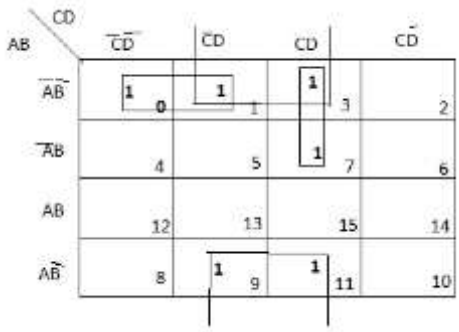
(5M)

9 **Simply the function F(w,x,y,z)= ∑m(2,3,12,13,14,15). Implement the simplified using gates. (10M) (Dec2014)** BTL 5

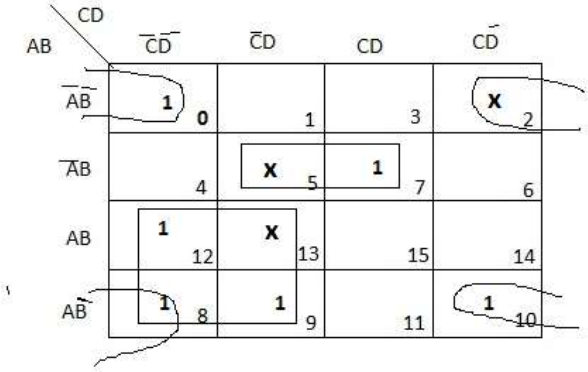


(10M)

10 **Simplify the following functions using K-map technique (10M) (June 2014)** BTL 5
G=∑m (0,1,3,7,9,11) (ii) f(w,x;y,z)=∑m(0,7,8,9,10,12)+∑d(2,5,13).



(5M)



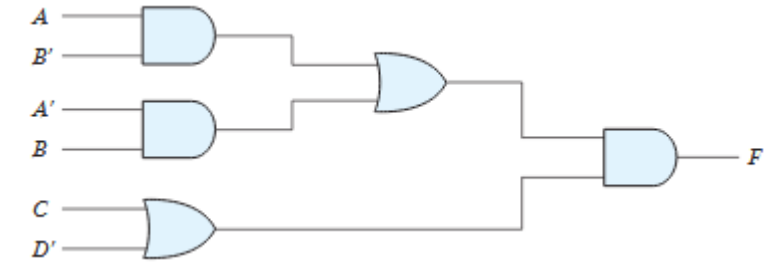
(5M)

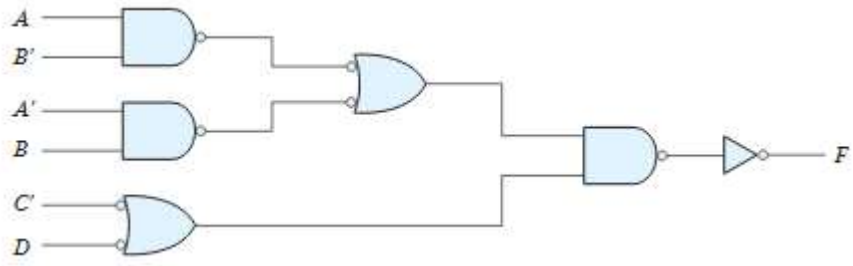
PART C

1 **Convert the multilevel function (15M)** BTL 6

$$F = (AB' + A'B)(C + D')$$

(i)

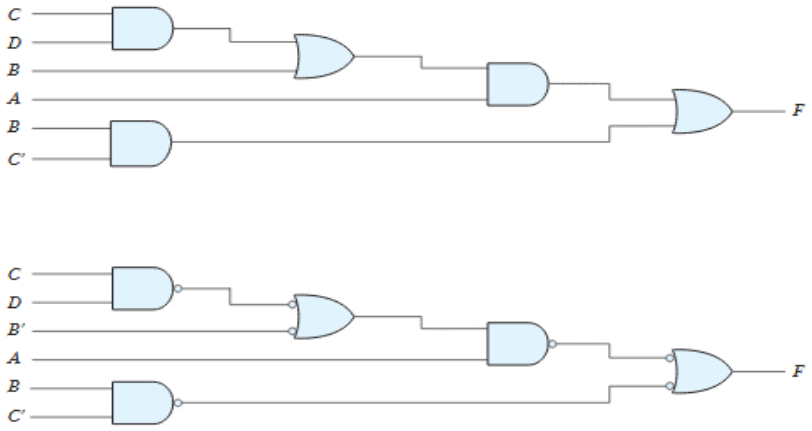




(5M)

$$F = A\,(CD + B) + BC'$$

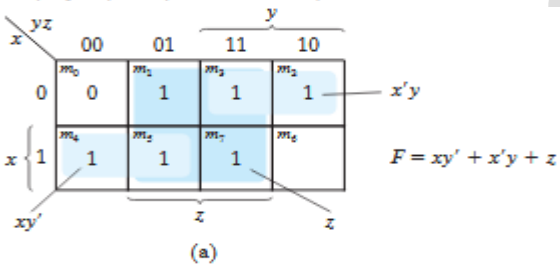
(i)



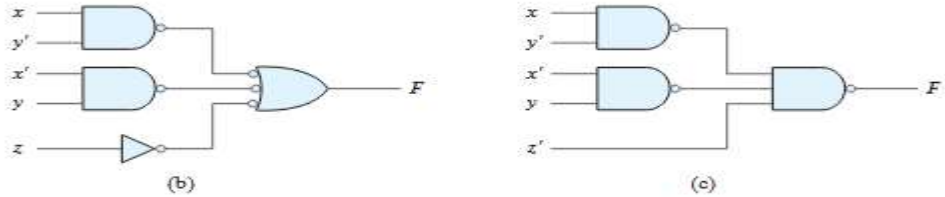
(10M)

2 **Implement the following Boolean function with NAND gates: (15M)** BTL 6

$F(x, y, z) = (1, 2, 3, 4, 5, 7)$

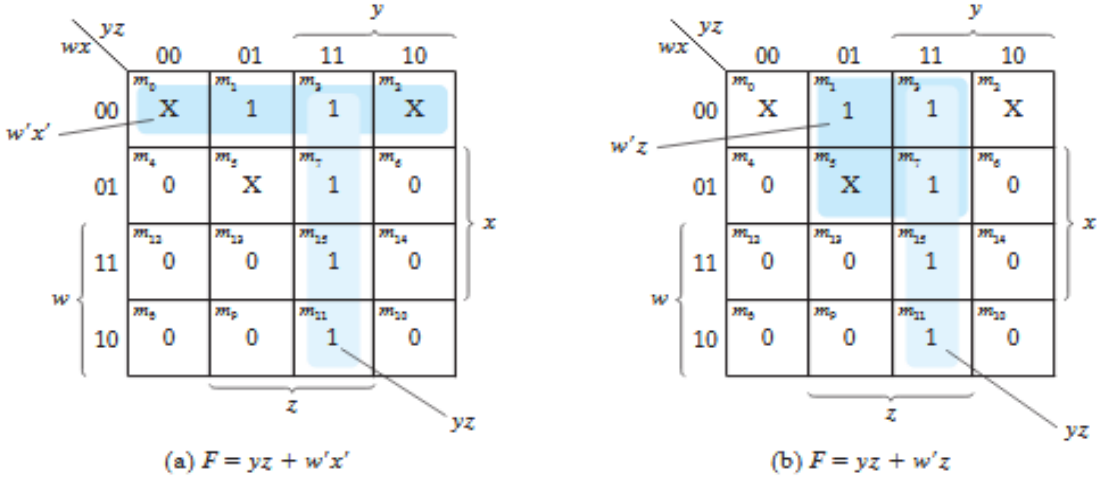


(5M)



(10M)

3 **Simplify the Boolean function F (w, x, y, z)=∑(1, 3, 7, 11, 15) which has the don't-care conditions d (w, x, y, z)=∑(0, 2, 5) (15M)** BTL 5

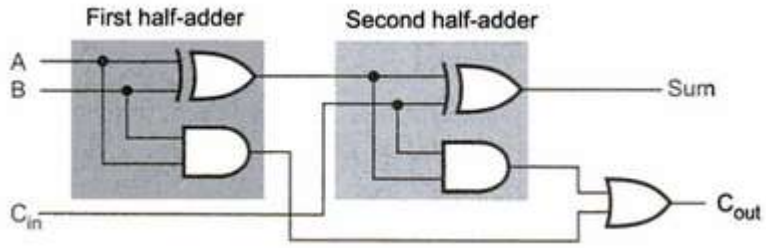
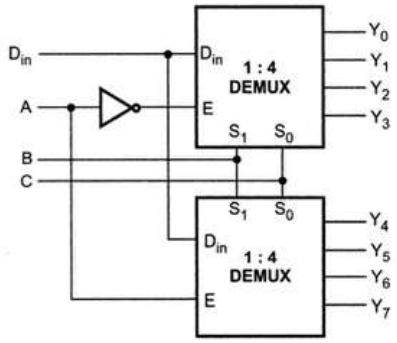
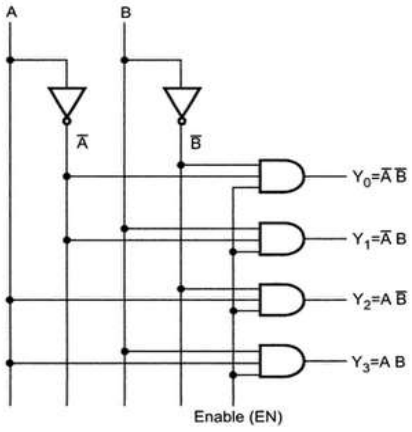


(15M)

UNIT – II

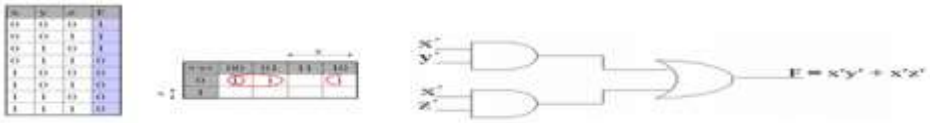
Combinational Circuits – Analysis and Design Procedures - Binary Adder-Subtractor - Decimal Adder - Binary Multiplier - Magnitude Comparator - Decoders – Encoders – Multiplexers - Introduction to HDL – HDL Models of Combinational circuits.

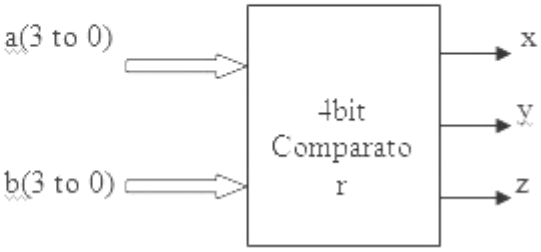
PART – A

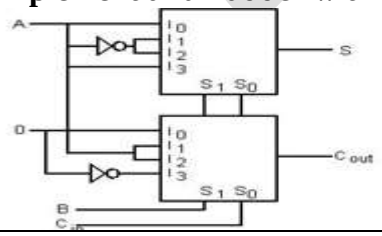
Q. No	Question& Answer
1.	**Construct a full adder using two half adders and OR gate. (Apr/May 2019)** BTL 1 
2	**Draw 1:8 Demultiplexer using two 1:4 Demultiplexer(Apr/May 2019)** BTL 1 
3	**Write the truth table of 2 to 4 decoder and draw its logic diagram.(Nov/Dec 2018)** BTL 1 

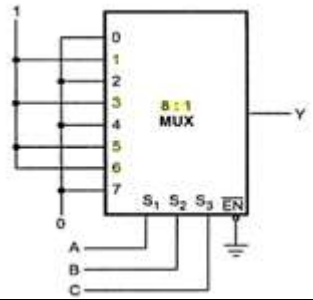
Inputs			Outputs			
EN	A	B	Y_3	Y_2	Y_1	Y_0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

4	**What is the propagation delay? (Nov/Dec 2018)** BTL 1 The time taken by the signal from the input transition to output is called propagation delay.
5	**What is priority encoder? (Dec 2014) (Apr/May 2017)** BTL 1 A priority encoder is an encoder circuit that includes the priority function. The operation of the priority encoder is such that if two or more inputs are activated at the same time, the output binary code will be generated to the highest-numbered input

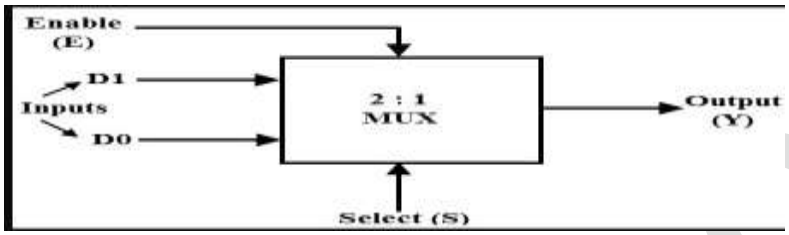
2	**Design the combinational circuit with 3 inputs and 1 output. The output is 1 when the binary value of the input is less than 3. The output is 0 otherwise. (May/June 2016)** BTL 1 
6	**Define Combinational circuits. (May/June 2016)** BTL 1 A combinational logic circuit consists of logic gates whose output is determined by the combination of current inputs.
7	**Draw the truth table of half adder. (Nov./Dec. 2015)** BTL 1 INPUTS: A, B; OUTPUTS: SUM, CARRY 0 0 0 0 0 1 1 0 1 0 1 0 1 1 0 1

8	**Write the Data flow description of a 4-bit Comparator. (Apr/May 2015)** BTL 1
	module comp(a,b,aeqb,agtb,altb);
	input [3:0] a,b;
	output aeqb,agtb,altb;
	reg aeqb,agtb,altb;
	always @(a or b)
	begin
	aeqb=0; agtb=0; altb=0;
	if(a==b)
	aeqb=1;
	else if (a>b)
	agtb=1;
	else
	altb=1;
	end
	end module
	 

9	**Write the data flow description of a 4-bit comparator. (May 2015)** BTL 1 module mag_comp(A,B,ALTB,AGTB,AETB); input [3:0] A,B; output ALTB,AGTB,AETB; assign ALTB=(A<B), AGTB=(A>B), AETB=(A= =B); end module
10	**Implement a full adder with 4×1 multiplexer. (May 2015)** BTL 1 
11	**Implement the following Boolean function using 8:1 multiplexer F(A,B,C)= ∑m(1,3,5,6) (Dec 2014)** BTL 1



12 **Draw a 2 to 1 multiplexer circuit. (June 2014)** BTL 1

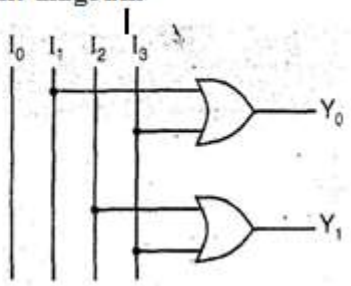


13 **Draw the truth table and circuit diagram of 4 to 2 encoder. (Dec 2013)** BTL 1

The Truth-table is as:

Input		Output	
		Y_0	Y_1
I_0	0	0	0
I_1	1	0	1
I_2	2	1	0
I_3	3	1	1

Circuit diagram



14 **Obtain the truth table for BCD to Excess-3 code converter. (Dec 2013)** BTL 1

Decimal	BCD				Excess – 3 code			
	A	B	C	D	E_3	E_2	E_1	E_0
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	1	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	0	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

15 **Write the stimulus for 2 to 1 line MUX. (June 2012)** BTL 1

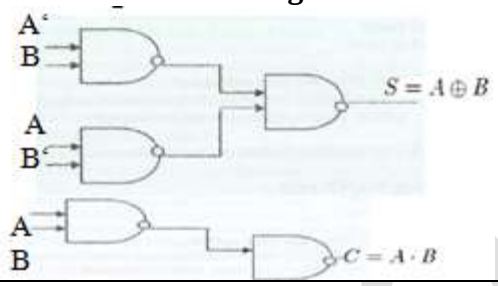
```
module exm4_6(A,B,S,O);
input A,B,S;
output O;
assign O=S ? A:B;
end module
```

16 **Distinguish between a decoder and a demultiplexer. (June 2012)** BTL 1

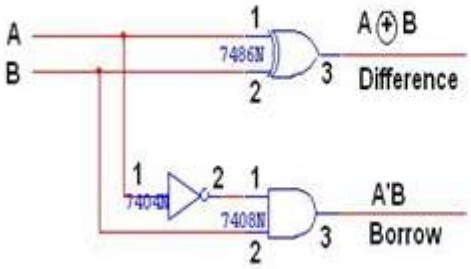
S.NO	DEMUX	DECODER

	1.	DEMUX Block diagram: Single input 1 X 2^n 2^n Outputs n selection inputs	DECODER n Intputs n X 2^n 2^n Outputs
	2.	The demultiplexer is the circuit that receives information on a single line and transmits this information on one of many output lines.	The decoder accepts a set of binary inputs and activates only the output that corresponds to that input number.
	3.	Data distributor	Decoder with enable input is used as Demultiplexer

17 **Implement half Adder using NAND Gates.** BTL 1

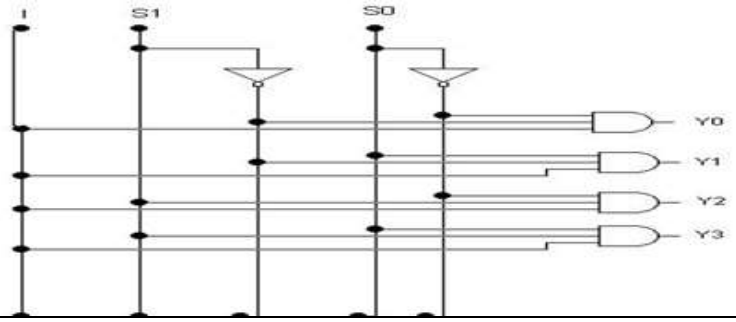


18 **Design a half subtractor.** BTL 1

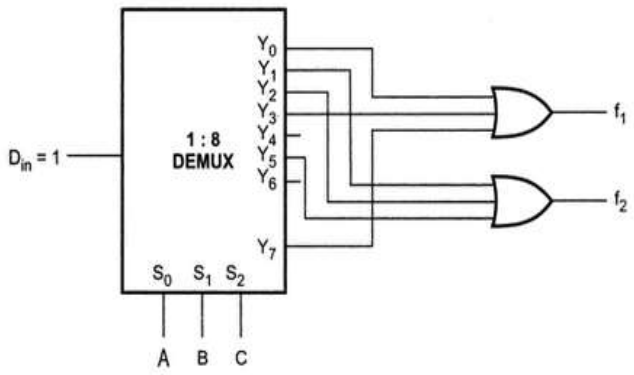
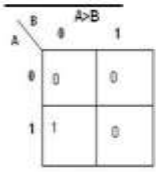
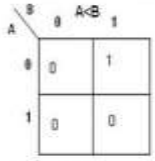
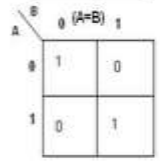
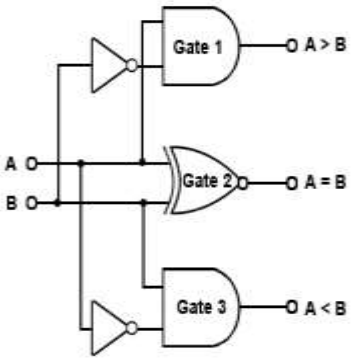


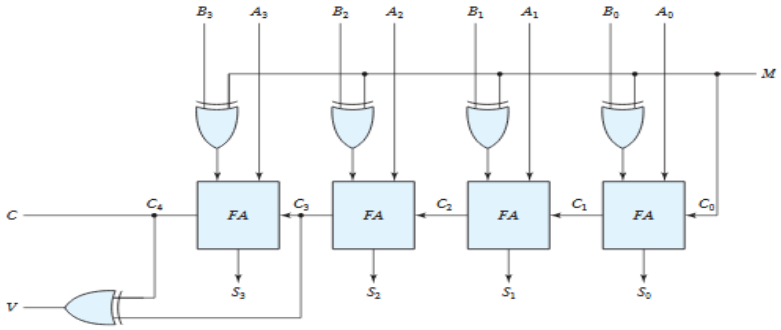
X	Y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

19 **Draw the logic diagram of a one to four line de-multiplexer.** BTL 1



20 **Implement the following functions using Demultiplixer.** BTL 1
f1(A,B,C)= ∑(0,3,7)
f2(A,B,C)=∑(1,2,3)

	Implementation using 1 : 8 demultiplexer 
21	**Design single bit comparator circuit.** BTL 1  A>B  Equation is A>B = $A\overline{B}$ A<B  Equation is A<B = $\overline{A}B$ (A=B)  The equation is f(A=B) = $\overline{A}\overline{B} + AB$ = A XNOR B  A_0 B_0 L E G 0 0 0 1 0 0 1 1 0 0 1 0 0 0 1 1 1 0 1 0
22	**Define Encoder.** BTL 1 An encoder is a digital circuit that performs the inverse operation of a decoder. An encoder has 2^n (or fewer) input lines and n output lines. The output lines, as an aggregate, generate the binary code corresponding to the input value
23	**Define Multiplexer.** BTL 1 A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally, there are 2^n input lines and n selection lines whose bit combinations determine which input is selected.
24	**Define Decoder.** BTL 1 A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines. If the n -bit coded information has unused combinations, the decoder may have fewer than 2^n outputs.
25	**Define Demultiplexer.** BTL 1 A demultiplexer (or demux) is a device that takes a single input line and routes it to one of several digital output lines. A demultiplexer of 2^n outputs has n select lines, which are used to select which output line to send the input. A demultiplexer is also called a data distributor.
	PART – B
1	**Design of 4 bit binary adder-subtractor circuit. (5M) (April/May/ 2019)** BTL 5 **Ans: Refer Morris Mano, PG.NO: 182**



(5M)

2 **Design a combinational circuit that accepts a 3-bit number and generates a 6-bit binary number output equal to the square of the input number. Write a high-levelbehavioural VHDL description for the circuit. (8M) (Apr/May/ 2019)** BTL 6

Ans: Refer Morris Mano, PG.NO: 414

Inputs			Outputs						
A_2	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	Decimal
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

(5M)

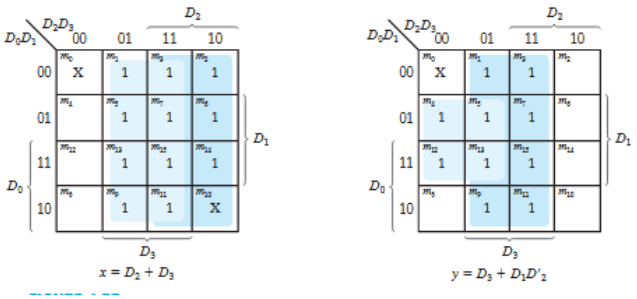
HDL code (3M)

3 **Explain the logic diagram of a 4-input priority encoder. (8M) (Apr/May/ 2019)** BTL 6

Ans: Refer Morris Mano, PG.NO: 196-198

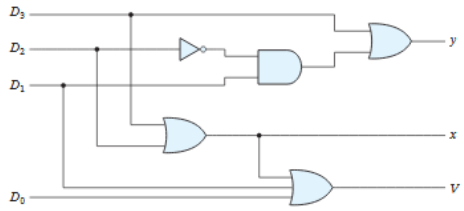
Truth Table of a Priority Encoder

Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1



(4M)

$x = D_2 + D_3$
$y = D_3 + D_1 D_2'$
$V = D_0 + D_1 + D_2 + D_3$

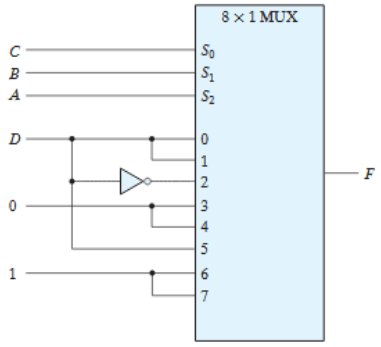


(4M)

4 **Implement the following Boolean with an 8 to 1 multiplexer and an inverter (A,B,C,D)=∑(1, 3, 4, 11, 12, 13, 14, 15) (8M) (Apr/May 2019)** BTL 5

Ans: Refer Morris Mano, PG.NO: 203

A	B	C	D	F	
0	0	0	0	0	F = D
0	0	0	1	1	
0	0	1	0	0	F = D
0	0	1	1	1	
0	1	0	0	1	F = D′
0	1	0	1	0	
0	1	1	0	0	F = 0
0	1	1	1	0	
1	0	0	0	0	F = 0
1	0	0	1	0	
1	0	1	0	0	F = D
1	0	1	1	1	
1	1	0	0	1	F = 1
1	1	0	1	1	
1	1	1	0	1	F = 1
1	1	1	1	1	



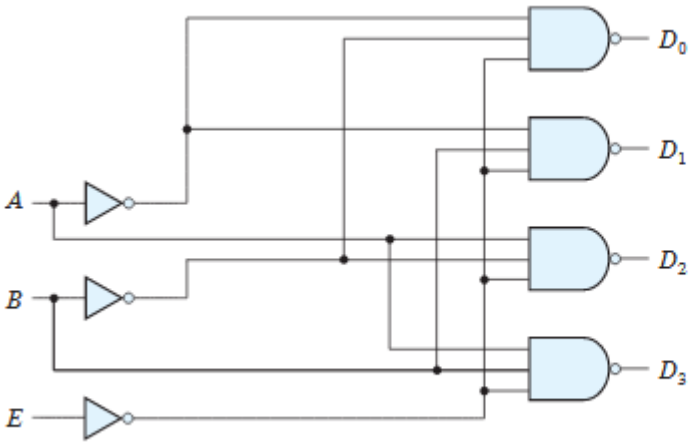
(5M)

(3M)

5 **Explain in detail about encoders and decoders. (13M) (Nov/Dec 2018)** BTL 5

Ans: Refer Morris Mano, PG.NO: 191-198

An encoder is a digital circuit that performs the inverse operation of a decoder. An encoder has 2^n (or fewer) input lines and n output lines. The output lines, as an aggregate, generate the binary code corresponding to the input value



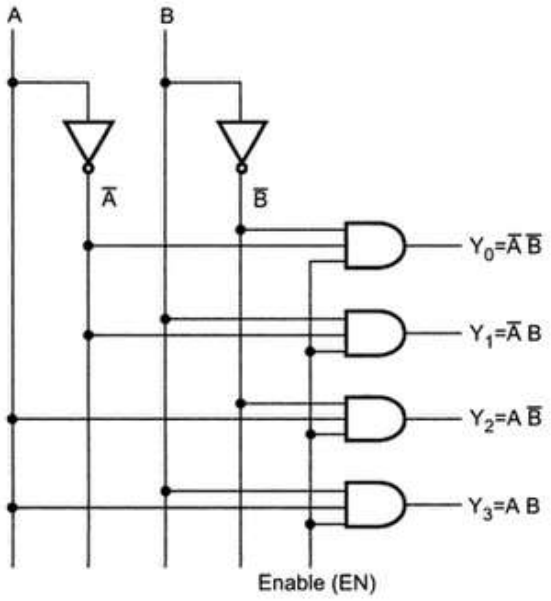
(a) Logic diagram

E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	1

(b) Truth table

(7M)

A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines. If the n -bit coded information has unused combinations, the decoder may have fewer than 2^n outputs.



(8M)

6 **Design 32 to 1 multiplexer using four 8 to 1 multiplexer and 2 to 4 decoder. (13M) (Nov/Dec2018)** BTL 5

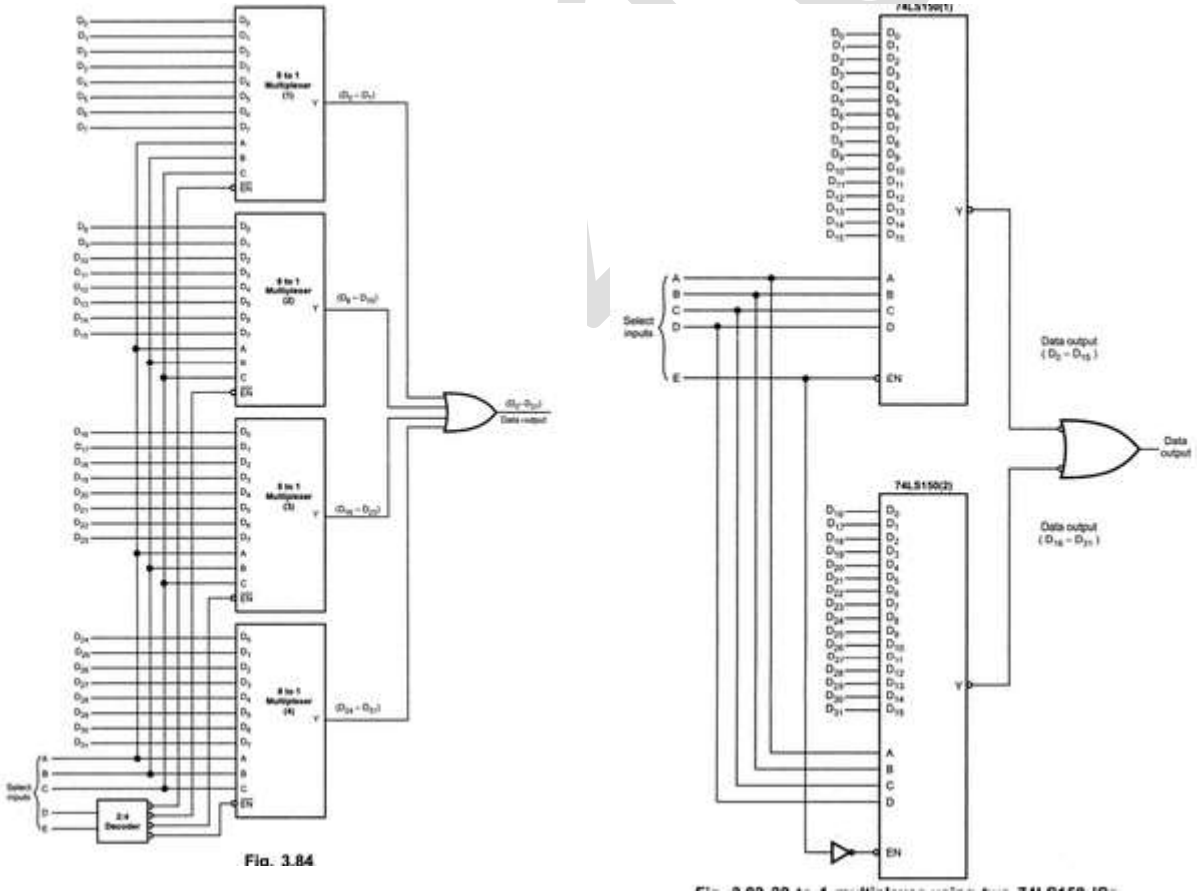


Fig. 3.84

(15M)

7 **Design a full adder with x, y, z and two outputs S and C. The circuits performs x+y+z, z is the input carry, C is the output carry and S is the Sum. (10M) (May/June 2016)** BTL 5

Ans: Refer Morris Mano, PG.NO: 174-176

A full adder is a combinational circuit that forms the arithmetic sum of three bits. It consists of three inputs and two outputs.

Two of the input variables, denoted by x and y , represent the two significant bits to be added. The third input, z , represents the carry from the previous lower significant position.

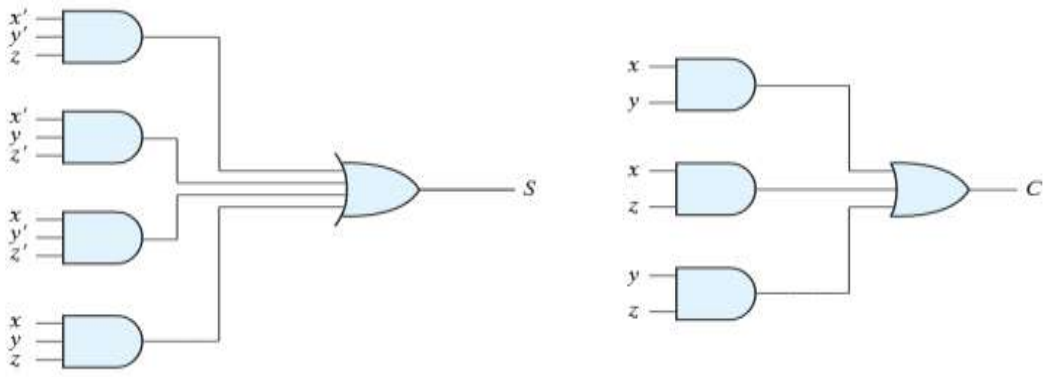
Two outputs are necessary because the arithmetic sum of three binary digits ranges in value from 0 to 3, and binary representation of 2 or 3 needs two bits.

The two outputs are designated by the symbols S for sum and C for carry. The binary variable S gives the value of the least significant bit of the sum. The binary variable C gives the output carry formed by adding the input carry and the bits of the words.

Full Adder

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

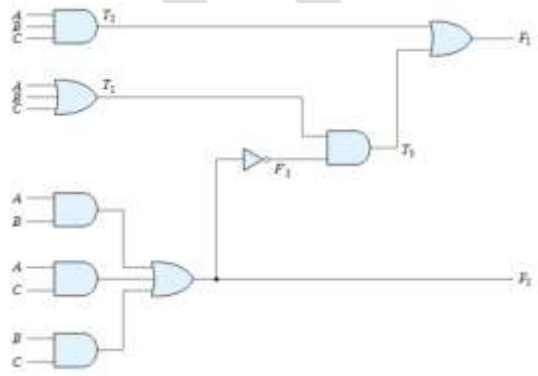
(7M)



(3M)

8 **(i) Explain the Analysis procedure. Analyze the following logic diagram. (10M) (Apr/May 2015)** BTL 6

Ans: Refer Morris Mano, PG.NO: 166-168



(5M)

(ii) With neat diagram explain the 4-bit adder with carry lookahead. (10M) BTL 6

Ans: Refer Morris Mano, PG.NO: 178-180

(i) Analysis procedure

The outputs of gates that are a function only of input variables are T_1 and T_2. Output F_2 can easily be derived from the input variables. The Boolean functions for these three outputs are

$$F_2 = AB + AC + BC$$
$$T_1 = A + B + C$$
$$T_2 = ABC$$

Next, we consider outputs of gates that are a function of already defined symbols:

$$T_3 = F'_2 T_1$$
$$F_1 = T_3 + T_2$$

To obtain F_1 as a function of A, B, and C, we form a series of substitutions as follows:

$$\begin{aligned} F_1 = T_3 + T_2 = F'_2 T_1 + ABC &= (AB + AC + BC)'(A + B + C) + ABC \\ &= (A' + B')(A' + C')(B' + C')(A + B + C) + ABC \\ &= (A' + B'C')(AB' + AC' + BC' + B'C) + ABC \\ &= A'BC' + A'B'C + AB'C' + ABC \end{aligned}$$

(5M)

A	B	C	F_2	F'_2	T_1	T_2	T_3	F_1
0	0	0	0	1	0	0	0	0
0	0	1	0	1	1	0	1	1
0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	0	0	0
1	0	0	0	1	1	0	1	1
1	0	1	1	0	1	0	0	0
1	1	0	1	0	1	0	0	0
1	1	1	1	0	1	1	0	1

(ii) Carry lookahead adder

Consider the circuit of the full adder shown in Fig. . If we define two new binary variables

$$P_i = A_i \oplus B_i$$
$$G_i = A_i B_i$$

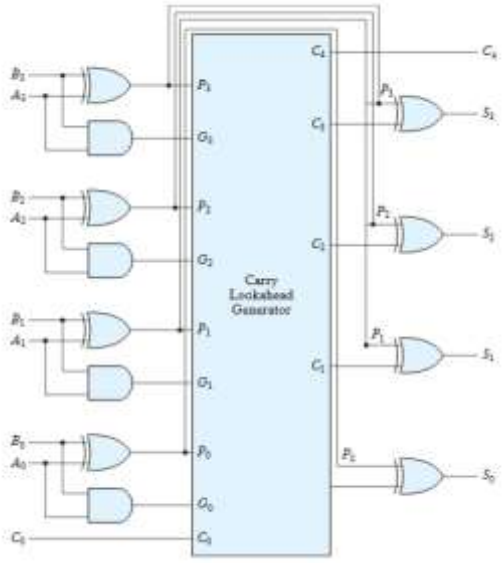
the output sum and carry can respectively be expressed as

$$S_i = P_i \oplus C_i$$
$$C_{i+1} = G_i + P_i C_i$$

G_i is called a *carry generate*, and it produces a carry of 1 when both A_i and B_i are 1, regardless of the input carry C_i. P_i is called a *carry propagate*, because it determines whether a carry into stage i will propagate into stage $i + 1$ (i.e., whether an assertion of C_i will propagate to an assertion of C_{i+1}).

We now write the Boolean functions for the carry outputs of each stage and substitute the value of each C_i from the previous equations:

$$C_0 = \text{input carry}$$
$$C_1 = G_0 + P_0 C_0$$



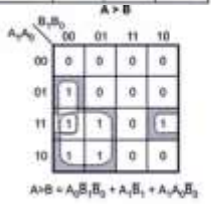
(5M)

9 **Design 2-bit magnitude comparator and write a verilog HDL code. (10M) (Dec 2015)** BTL 5

Ans: Refer Morris Mano, PG.NO: 230

2—bit comparator

Inputs				Outputs		
A_1	A_0	B_1	B_0	A > B	A = B	A < B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0



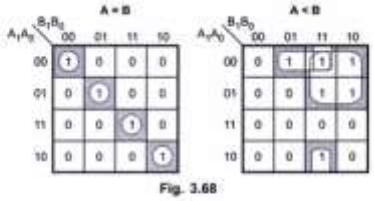
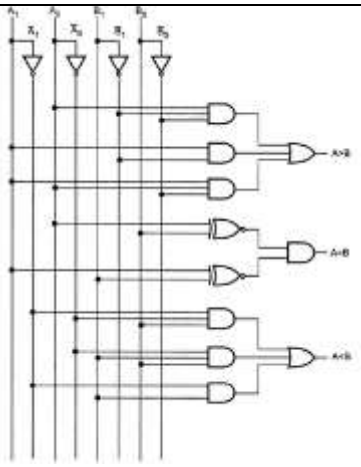


Fig. 3.68

$$(A = B) = \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1B_0 + A_1A_0B_1B_0 + A_1\bar{A}_0B_1\bar{B}_0$$
$$= \bar{A}_1\bar{B}_1(\bar{A}_0\bar{B}_0 + A_0B_0) + A_1B_1(A_0B_0 + \bar{A}_0\bar{B}_0)$$
$$= (A_0 \odot B_0)(A_1 \odot B_1)$$
$$(A < B) = \bar{A}_1\bar{A}_0B_0 + \bar{A}_1B_1B_0 + \bar{A}_1B_1$$

(5M)



(5M)

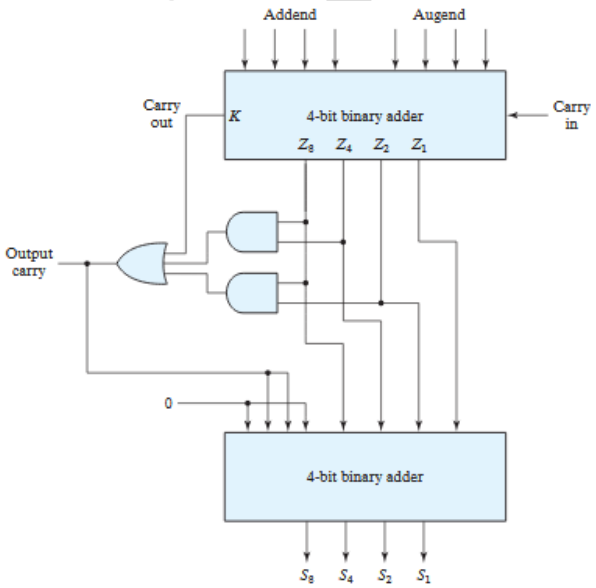
10 **Design a combinational circuit to perform BCD addition. (10M) (Dec 2013)** BTL 5

Ans: Refer Morris Mano, PG.NO: 184-188

Derivation of BCD Adder

Binary Sum					BCD Sum					Decimal
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

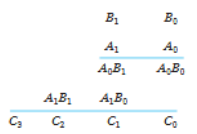
(5M)



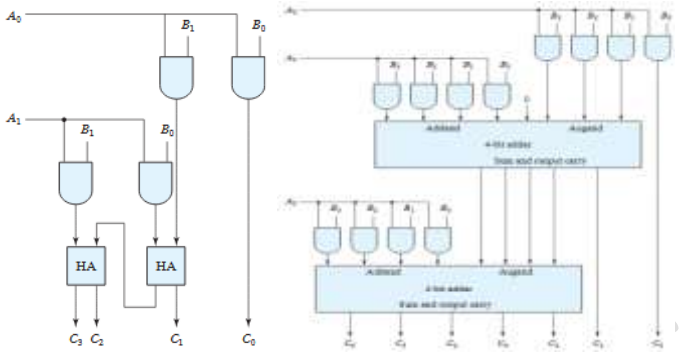
(5M)

11 **Design 4-bit parallel multiplier. (13M)** BTL 6

Ans: Refer Morris Mano, PG.NO: 186-188



(7M)



(8M)

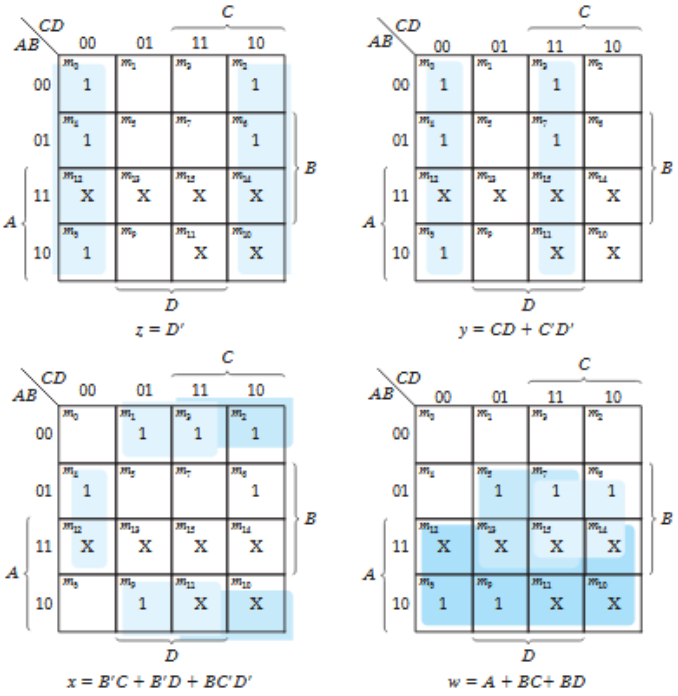
PART C

1 **Design a BCD to excess-3 code converter and explain. (15M) (Apr/May 2019)** BTL 5

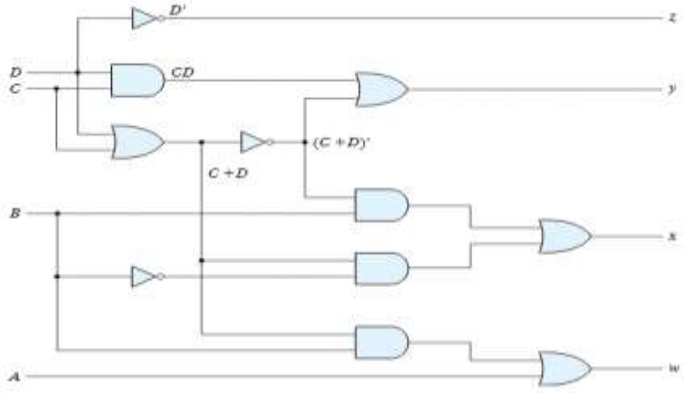
Ans: Refer Morris Mano, PG.NO: 169-172

Truth Table for Code Conversion Example

Input BCD				Output Excess-3 Code			
A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0



(10M)



(5M)

2 **Draw and explain the logic circuit of a 4bit magnitude comparator.(15M) (Apr/May 2019)** BTL 5

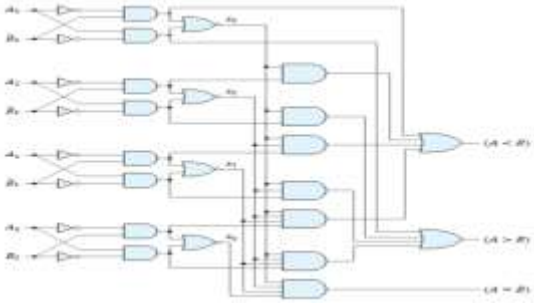
Ans: Refer Morris Mano, PG.NO: 188-190

The comparison of two numbers is an operation that determines whether one number is greater than, less than, or equal to the other number. A *magnitude comparator* is a combinational circuit that compares two numbers A and B and determines their relative magnitudes. The outcome of the comparison is specified by three binary variables that indicate whether $A > B$, $A = B$, or $A < B$.

$$(A > B) = A_3B_3' + x_3A_2B_2' + x_3x_2A_1B_1' + x_3x_2x_1A_0B_0'$$

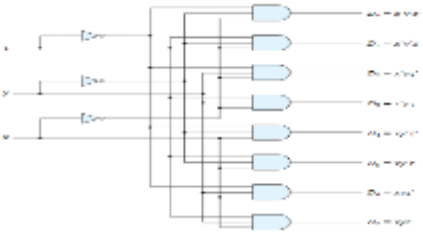
$$(A < B) = A_3'B_3 + x_3A_2'B_2 + x_3x_2A_1'B_1' + x_3x_2x_1A'n_0B_0'$$

(10M)



(5M)

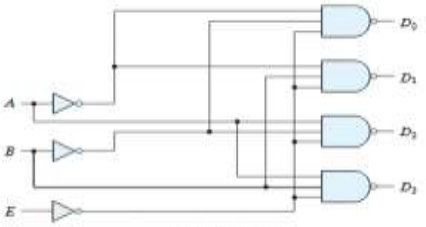
3 **Implement 3 to 8, 2 to 4, 4 to 16 using 3 to decoder(15M) BTL 5 Morris mano page:191-193**



Truth Table of a Three-to-Eight-Line Decoder

Inputs			Outputs							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

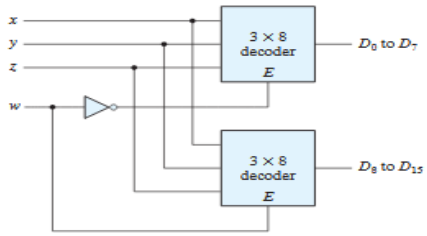
(8M)



(a) Logic diagram

E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	1

(b) Truth table



(7M)

UNIT – III SYNCHRONOUS SEQUENTIAL LOGIC

Sequential Circuits - Storage Elements: Latches , Flip-Flops - Analysis of Clocked Sequential Circuits - State Reduction and Assignment - Design Procedure - Registers and Counters - HDL Models of

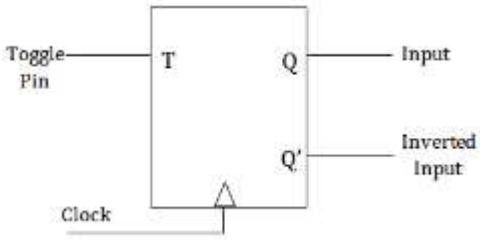
Sequential Circuits.

PART – A

Q. No	Question& Answer
1	**State the difference between latches and flipflops. (Apr/May 2019)** BTL 1

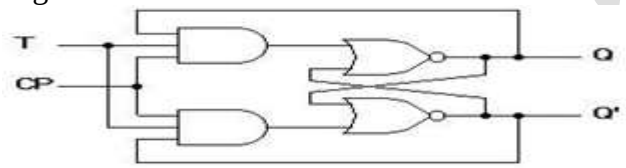
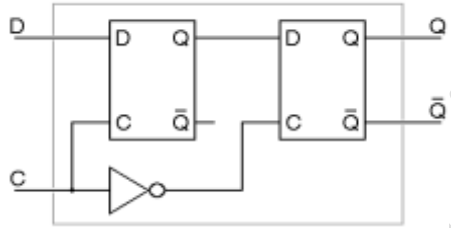
Latches	Flip Flops
Latches are building blocks of sequential circuits and these can be built from logic gates	Flip flops are also building blocks oof sequential circuits. But, these can be built from the latches.
Latch continuously checks its inputs and changes its output correspondingly.	Flip flop continuously checks its inputs and changes its output correspondingly only at times determined by clocking signal
The latch is sensitive to the duration of the pulse and can send or receive the data when the switch is on	Flipflop is sensitive to a signal change. They can transfer data only at the single instant and data cannot be changed until next signal change. Flip flops are used as a register.
It is based on the enable function input	It works on the basis of clock pulses
It is a level triggered, it means that the output of the present state and input of the next state depends on the level that is binary input 1 or 0.	It is an edge triggered, it means that the output and the next state input changes when there is a change in clock pulse whether it may a +ve or -ve clock pulse.

2	**What is meant by edge triggered flip flops? (Apr/May 2019)** BTL 1 An edge-triggered flip-flop changes states either at the positive edge (rising edge) or at the negative edge (falling edge) of the clock pulse on the control input.
3	**State the operation of T FF. (Nov/Dec 2018)** BTL 1 T flip flop is modified form of JK flip-flop **making** it to operate in toggling region. Whenever the clock signal is LOW, the input is never going to affect the output state**.** The clock has to be high for the inputs to get active. Thus, T flip-flop is a controlled Bi-stable latch where the clock signal is the control signal. Thus, the output has two stable states based on the inputs which have been discussed below.



Input	Outputs	
	Present State	Next State
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

4	**Mention the different types of shift register. (Nov/Dec 2018)** BTL 1 There are 4 types of shift registers: serial in-serial out (SISO) shift registers, serial in-parallel out (SIPO) shift registers,

	parallel in-serial out (PISO) shift registers parallel in-parallel out (PIPO) shift registers.
5	**State the excitation table of JK Flip Flop. (May/June 2016)** BTL 1 Q_n Q_{n+1} J K 0 0 0 X 0 1 1 X 1 0 X 1 1 1 X 0
6	**What is the minimum number of flip flops needed to build a counter of modulus 8? (May 2016)** BTL 1 3 Flip Flops
7	**Write short notes on propagation delay. (Nov/Dec 2015)** BTL 1 Propagation delay is the amount of time it takes for the head of the signal to travel from the sender to the receiver.
8	**Draw the diagram of T flip flop and discuss its working. (Nov/Dec 2015)** BTL 1 The T flip flop has two possible values. When T = 0, the flip flop does a hold. A hold means that the output, Q is kept the same as it was before the clock edge. When T = 1, the flip flop does a toggle, which means the output Q is negated after the clock edge, compared to the value before the clock edge. 
9	**Give the block diagram of master-slave D flip- flop. (May 2015)** BTL 1 
10	**What is ring counter? (May 2015)** BTL 1 A ring counter is a type of counter composed of a type circular shift register. The output of the last shift register is fed to the input of the first register.
11	**With reference to a JK flip-flop, what is racing? (June/Dec 2014)** BTL 1 (i) Because of the feedback connection in the JK flip-flop,when both J & K are equal to 1 at the same time, the output will be complemented while activating the clock pulse. (ii) the output is complemented again and again if the pulse duration of the clock signal is greater than the signal propagation delay of the JK flipflop for this particular input combination (J=K=1).(iii) there is a race between 0 and 1 within a single clock pulse.this condition of the JK FF is called race-around condition or racing.
12	**What are Mealy and Moor machines? 1(Dec 2014)** BTL 1 Mealy and Moor machines are two models of clocked or synchronous sequential circuit. **Mealy machine:** The output depends on both the present state of the flip-flops and on the inputs. **Moore machine:** The output depends only on the present state of the flip-flops.
13	**Write the characteristics table and equation of JK flip flop. (June 2014)** BTL 1

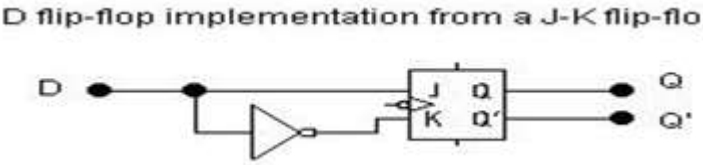
("X" is "don't care")

	Present state	J	K
0	0	0	x
0	1	0	x
1	0	x	1
1	1	x	0

Characteristic equation Q(next) = JQ' + K'Q

14 **Write any two applications of shift registers. (June 2014)** BTL 1
(i)Parallel to serial conversion for signal transmission
(ii)Pattern recognition

15 **Show D flip-flop implementation from a J-K flip-flop. (Dec 2013)** BTL 1



16 **What is meant by triggering of Flip flop?** BTL 1
The state of a flip-flop is switched by a momentary change in the input signal. This momentary change is called a trigger and the transition it causes is said to trigger the flip-flop.

17 **Why D FF is known as Delay FF?** BTL 1
The binary information present at the data i/p of the D FF is transferred to the Q o/p when the cp input is enabled. The o/p follows the data i/p as long as the pulse remains in its 1 state. When the pulse goes to 0, the binary information that was present at the data i/p at the time the pulse transition occurred is retained at the Q o/p until the pulse i/p is enabled again. So D FF is known as Delay FF.

18 **What is the minimum number of flip-flops needed to build a counter of modulus 60?**BTL 1
Modulus N $\leq 2^k$, where k is the number of flip-flops
Modulus $60 < 2^6 = 64$, k = 6

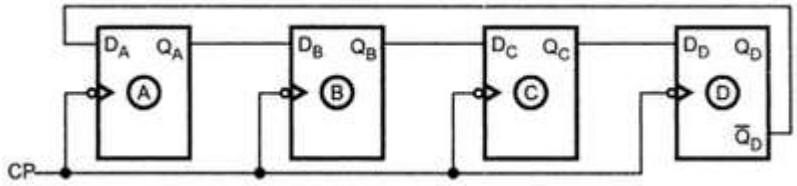
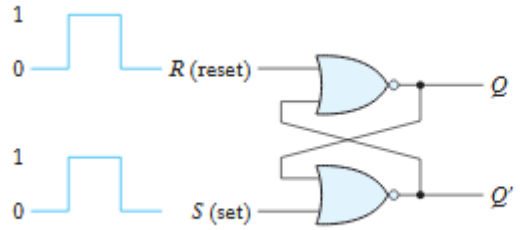
19 **What is a universal shift register?**BTL 1
(i) A register may operate in any of the following five modes
1. SISO 2. SIPO 3. PIPO 4.PISO 5. Bidirectional
(ii)If a register can be operated in all the five possible ways, it is known as Universal Shift Register

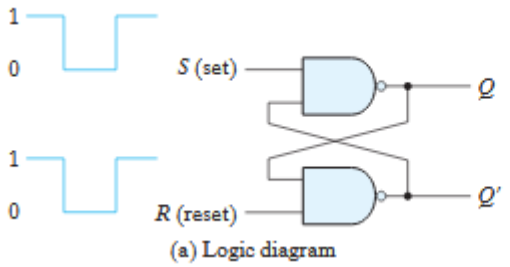
20 **Differentiate between sequential and combinational circuits.**BTL 1

Combinational circuits	**Sequential circuits**
Output depends only on the past values of input.	Output depends on the present and past values of input.
Feedback path is not used in combinational circuits.	Feedback path is used for sequential circuits.
Memory element is not present	Memory element is present.
Clock is not used in this circuit.	Clock is used in sequential circuits.
Examples: adder, subtractors, code converters, comparators, Mux,etc	Examples: flip-flops,counters,registers,etc

21 **What is a Mealy circuit?**BTL 1
Mealy circuit is a clocked or synchronous sequential circuit.
The output depends on both the present state of the flip-flops and on the inputs.

22 **What is a state diagram?**BTL 1

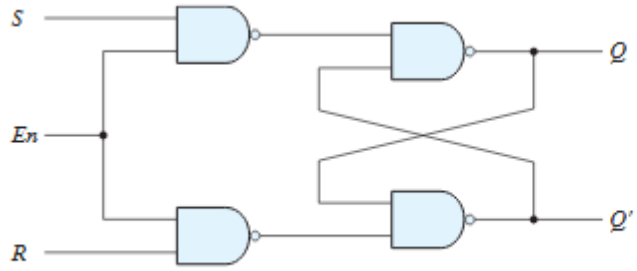
	(i) State diagram is the graphical representation of state table of sequential logic circuits. (ii)In the state diagram, a state is represented by a circle and the transition between states is indicated by directed lines connecting the circles. (iii)The directed lines are labeled with two binary numbers separated by a slash. The input value during the present state is labeled first and the number after the slash gives the output during the present state.
23	**What is finite state machine?**BTL 1 A finite state machine (or finite automation) is an abstract model describing the synchronous sequential machine and its spatial counter, part, the iterative network
24	**What do you meant by the term state reduction problem?**BTL 1 The reduction of the number of flip-flops in a sequential circuit is referred to as the state – reduction problem. State – reduction algorithms are concerned with procedures for reducing the number of states in a state table while keeping the external input – output requirements unchanged.
25	**What is Johnson counter?** BTL 1 In Johnson counter the Q output of each stage of flip-flop isconnected to the D-input of the next stage. The single exception is that the complement output of the last flip-flop is connectedback to the D-input of the first flip-flop. 
	PART – B
1	**(i) Describe the operations of R-S latch with a neat sketch. (5M)** BTL 5 **Ans: Refer Morris Mano, PG.NO: 265-267** **(ii)Design a sequential circuit with two D flip-flops A and B and one input X. When X=0, the state of the circuit remains the same. When X=1, the circuit goes through the state transition from 00 to 10 to 11 to 01 back to 00 and then repeats. (8M) (Apr/May 2019)** BTL 6 The latch has two useful states. When output Q=1 and Q=0, the latch is said to be in the set state . When Q=0 and Q=1, it is in the reset state . Outputs Q and Q are normally the complement of each other. However, when both inputs are equal to 1 at the same time, a condition in which both outputs are equal to 0 (rather than be mutually complementary) occurs. If both inputs are then switched to 0 simultaneously, the device will enter an unpredictable or undefined state or a meta-stable state.  (a) Logic diagram S R | Q Q' 1 0 | 1 0 0 0 | 1 0 (after $S = 1, R = 0$) 0 1 | 0 1 0 0 | 0 1 (after $S = 0, R = 1$) 1 1 | 0 0 (forbidden) (b) Function table



(a) Logic diagram

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

(b) Function table



(a) Logic diagram

En	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table

(5M)

2 **(i) Construct a clocked Master Slave J-K flip flop and explain. (5M)** BTL 5

Ans: Refer Morris Mano, PG.NO: 270-272

(ii) A sequential circuit with two D flip flops A and B, two inputs X and Y and one output Z is specified by the following input equations

A(t+1) = x'y+xA

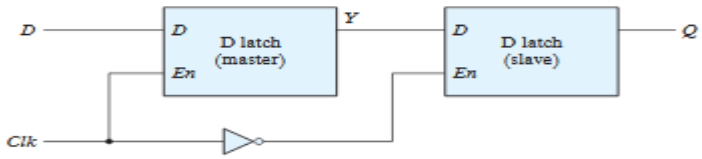
B(t+1)=x'b+xA

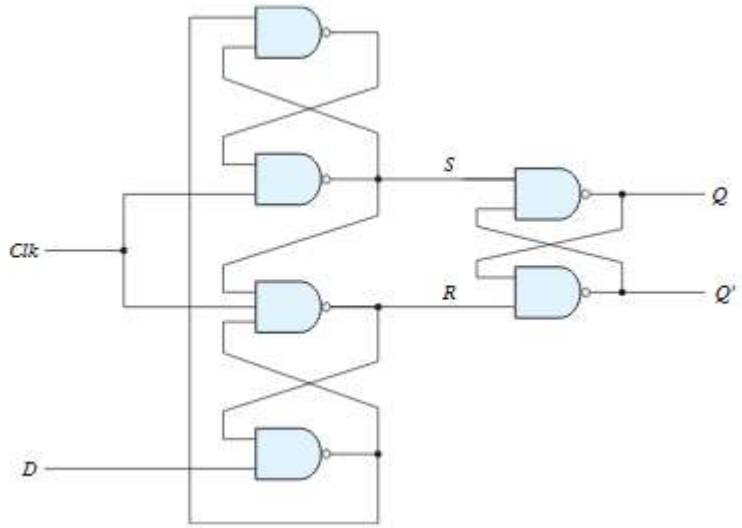
z=B

Draw the logic diagram of the circuit. derive the state table and state diagram and state whether it is a Mealy or a Moore machine (8M) (April/May 2019) BTL 6

The construction of a D flip-flop with two D latches and an inverter is shown in Fig. 5.9 . The first latch is called the master and the second the slave.

The circuit samples the Dinput and changes its output Q only at the negative edge of the synchronizing or control-ling clock (designated as Clk). When the clock is 0, the output of the inverter is 1. The slave latch is enabled, and its output Q is equal to the master output Y .





(5M)

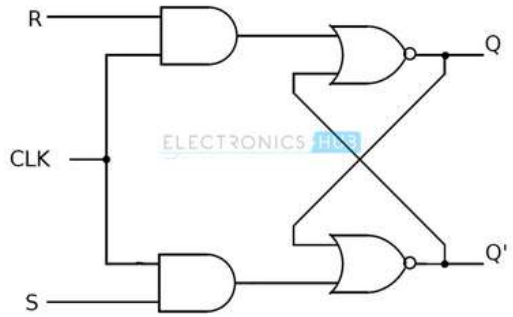
3 **Design and implementation of SR flip-flop using NOR gate (13M) (Nov/Dec 2018)** BTL 5

The SR flip – flop is one of the fundamental parts of the sequential circuit logic. SR flip – flop is a memory device and a binary data of 1 – bit can be stored in it. SR flip – flop has two stable states in which it can store data in the form of either binary zero or binary one. Like all flip – flops, an SR flip – flop is also an edge sensitive device.

(5M)

Working

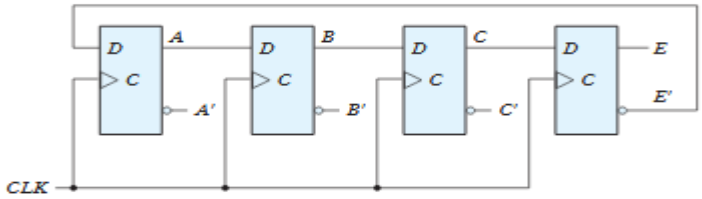
Case 1:
When both the SET and RESET inputs are low, then the output remains in previous state i.e. it holds the previous data
Case 2:
When SET input is low and RESET input is high, then the flip flop will be in RESET state.
Case 3:
When SET input is high and RESET input is low, then the flip flop will be in SET state
Case 4:
When both the SET and RESET inputs are high, then the flip flop will be undefined state.



S	R	Q	State
0	0	Previous State	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Forbidden

(8M)

4 **Explain in detail about 4-bit Johson counter (13M)) (Nov/Dec 2018)** BTL 5
Ans: Refer Morris Mano, PG.NO: 371-372



(8M)

Sequence number	Flip-flop outputs A	B	C	E	AND gate required for output
1	0	0	0	0	$A'E'$
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	$A'B$
7	0	0	1	1	$B'C$
8	0	0	0	1	$C'E$

(5M)

5 **Design a BCD synchronous counter using T Flip Flop and implement it. Construct its timing diagram. (10M) (May/June 2016)** BTL 6

Ans: Refer Morris Mano, PG.NO: 363-364

The simplified function:

$$T_{Q1} = 1$$
$$T_{Q2} = Q'_8 Q_1$$
$$T_{Q4} = Q_2 Q_1$$
$$T_{Q8} = Q_8 Q_1 + Q_4 Q_2 Q_1$$
$$y = Q_8 Q_1$$

(5M)

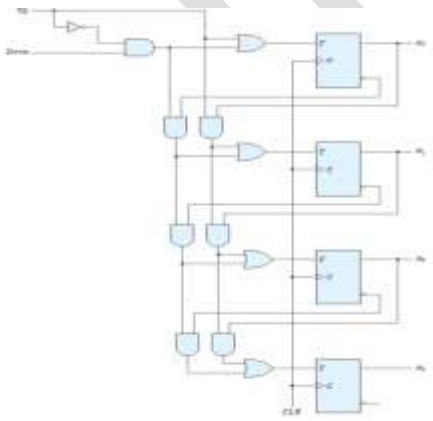
State Table for BCD Counter

Present State Q_8	Q_4	Q_2	Q_1	Next State Q_8	Q_4	Q_2	Q_1	Output y	Flip-Flop Inputs TQ_8	TQ_4	TQ_2	TQ_1
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	1	1	0	0	1

(5M)

6 **Design 4-bit synchronous up/down counter (10M) (May/June 2016)** BTL5

Ans: Refer Morris Mano, PG.NO: 360-362

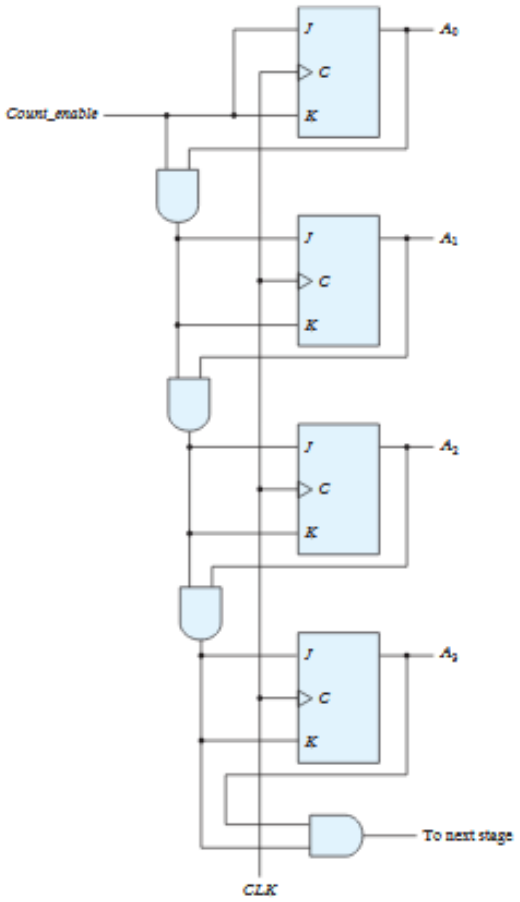


(10M)

7 **Design four bit synchronous counter with JK flip flop and draw the diagram. (10M) (Nov/ Dec**

2015) BTL 6

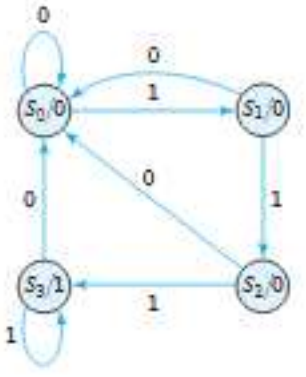
Ans: Refer Morris Mano, PG.NO: 360-361



(10M)

8 **Design a sequence detector that detects a sequence of three or more consecutive 1's in a string of bits coming through an input line and produces an output whenever this sequence is detected. (10M) (Nov/ Dec 2015)** BTL 5

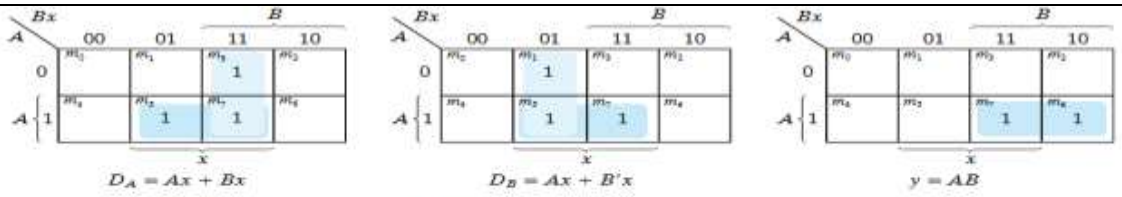
Ans: Refer Morris Mano, PG.NO: 323-325



State Table for Sequence Detector

Present State		Input	Next State		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

(5M)



K-Maps for sequence detector

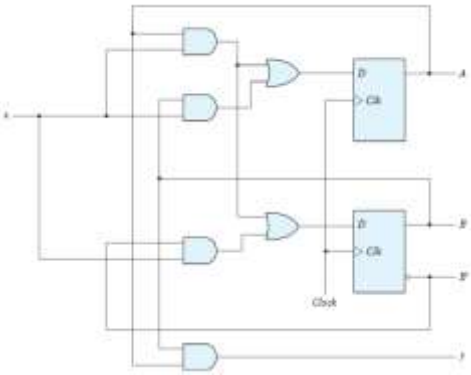
can be obtained directly from the next-state columns of A and B and expressed in sum-of-minterms form as

$$A(t+1) = D_A(A,B,x) = \Sigma(3,5,7)$$
$$B(t+1) = D_B(A,B,x) = \Sigma(1,5,7)$$
$$y(A,B,x) = \Sigma(6,7)$$

where A and B are the present-state values of flip-flops A and B, x is the input, and D_A and D_B are the input equations. The minterms for output y are obtained from the output column in the state table.

The Boolean equations are simplified by means of the maps plotted in Fig. 1. The simplified equations are

$$D_A = Ax + Bx$$
$$D_B = Ax + B'x$$
$$y = AB$$



(5M)

9 **Write the HDL code for up-down counter using behavioral model. (10M) (Dec 2015)** BTL 5

```
module behav_counter( d, clk, clear, load, up_down, qd);
input [7:0] d;
input clk;
input clear;
input load;
input up_down;
output [7:0] qd;
reg [7:0] cnt;
assign qd=cnt;
always @ (posedgeclk) begin
 if (!clear)
        cnt = 8'h00;

else if (load)
        cnt = d;
else if (up_down)
        cnt = cnt + 1;
else
        cnt = cnt - 1;
end
endmodule
```

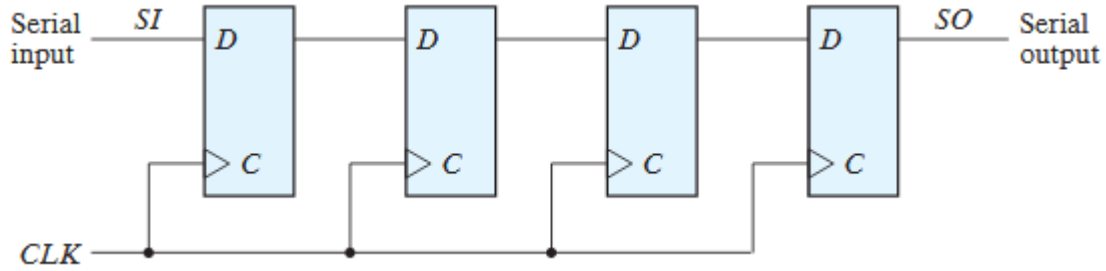
(10M)

10 **Explain shift register with parallel load . (13M)** BTL 6

Ans: Refer Morris Mano, PG.NO: 346-349

A register capable of shifting the binary information held in each cell to its neighboring cell, in a selected direction, is called a *shift register*. The logical configuration of a shift register consists of a chain of flip-flops in cascade, with the output of one flip-flop connected to the input of the next flip-flop. All flip-flops receive common clock pulses, which activate the shift of data from one stage to the next.

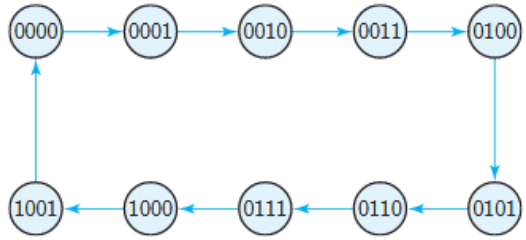
(8M)



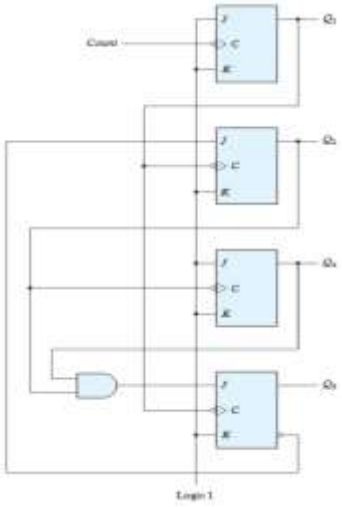
(5M)

11 **Design of 4-bit BCD ripple counter (13M)** BTL 5

Ans: Refer Morris Mano, PG.NO: 357-359



(10M)



(3M)

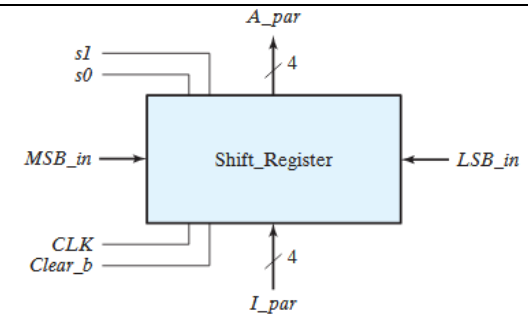
PART C

1 **Explain the operations of a 4-bit bidirectional shift register. (15M)** BTL 5

Ans: Refer Morris Mano, PG.NO: 352-354

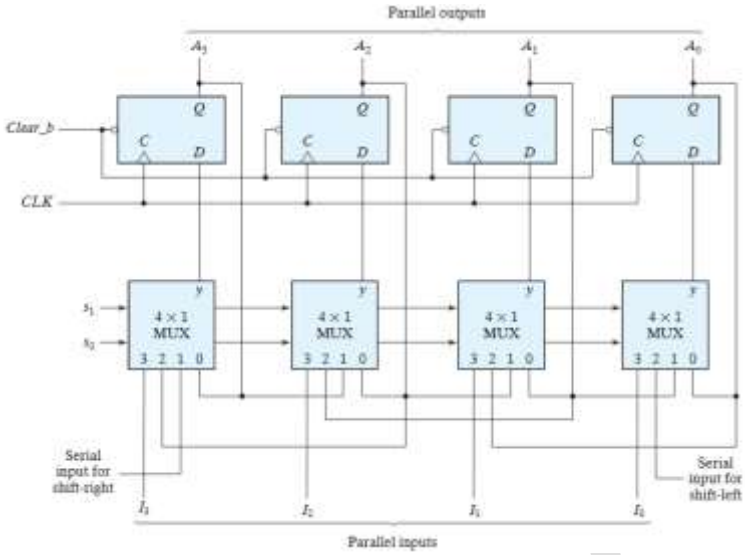
4-bit bidirectional shift register

Description: (3M)



(a)

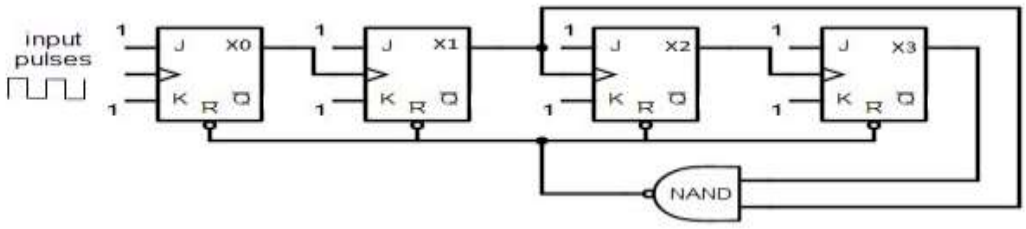
(7M)



(5M)

2 **Design a decade counter using JK flipflop. (15M) (Nov/Dec 2018)** BTL 6

A binary coded decimal (BCD) is a serial digital counter that counts ten digits and it resets for every new clock input. As it can go through 10 unique combinations of output, it is also called as "Decade counter". A BCD counter can count 0000, 0001, 0010, 1000, 1001, 1010, 1011, 1110, 1111, 0000, and 0001 and so on.



(10M)

Decade Counter Operation

When the Decade counter is at REST, the count is equal to 0000. This is first stage of the counter cycle. When we connect a clock signal input to the counter circuit, then the circuit will count the binary sequence. The first clock pulse can make the circuit to count up to 9 (1001). (5M)

3 **Design 3-bit ring counter. (15M)** BTL 5

Ans: Refer Morris Mano, PG.NO: 368-370

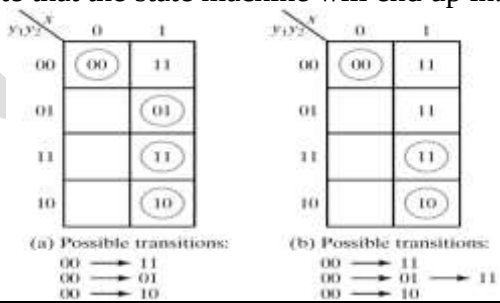
Description of ring counter design (10M)

Circuit design (5M)

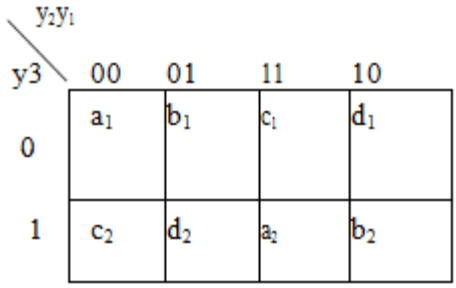
UNIT – IV

Analysis and Design of Asynchronous Sequential Circuits – Reduction of State and Flow Tables – Race-free State Assignment – Hazards.

PART – A

Q. No	Question& Answer
1	**What is meant by race free condition in sequential circuits? (Apr/May 2019)** BTL 1 When the binary assignment of state variable is done properly to avoid in proper transition in sequential circuit is called race free condition in sequential circuits
2	**Define state table. (Nov/Dec 2018)** BTL 1 Table presenting the present state, inputs and next state with or without output of a sequential circuit is called state table.
3	**What is race around condition? (Nov/Dec 2018)** BTL 1 When 2 or more binary state variables change their value in response to a change in an inputvariable, race condition occurs in an asynchronous sequential circuit. In case of unequal delays, a race condition may cause the state variables to change in an unpredictable manner.
4	**Define the critical race and non critical race. (May/June 2016)** BTL 1 Critical race in asynchronous circuits occur between two signals that are required to change at the same time when the next stable state is dependent on the delay paths in the circuit. Non Critical race The final stable state does not depend on the change order of state variables.
5	**What is lockout? How is avoided? (May/June 2016)** BTL 1 Lockout condition is that condition wherein a counter gets onto a forbidden state and rather than coming out of it to another acceptable state or initial state, the counter switches to another forbidden state and gets stuck up in the cycle of forbidden states only. The counter should be provided with an additional circuit. This will force the counter from an unused state to the next state as initial state. It is not always necessary to force all unused states into an initial state. This frees the circuit from the Lock out condition
6	**What is critical race condition? Give example. (Apr/May 2015)** BTL 1 A *critical race condition* occurs when the order in which internal variables are changed determines the eventual state that the state machine will end up in.  (a) Possible transitions: 00 → 11, 00 → 01, 00 → 10 (b) Possible transitions: 00 → 11, 00 → 01 → 11, 00 → 10
7	**Define critical race in asynchronous sequential circuits. (May 2015)** BTL 1 Critical race in asynchronous circuits occur between two signals that are required to change at the same time when the next stable state is dependent on the delay paths in the circuit
8	**What are the types of hazards? (June 2014) (May/June 2014)** BTL 1 (i) Static hazards (ii) Dynamic hazards
9	**What is a Hazard? (June 2012/Dec 2014)** BTL 1 Hazards are unwanted switching transients that may appear at the output of a circuit because different paths exhibit different propagation delays. Hazards occur in combinational circuits, where they may cause a temporary false output value. When this condition occurs in asynchronous sequential circuits, it

	may result in a transition to a wrong stable state. Steps must be taken to eliminate this effect.
10	**Difference between fundamental mode circuits and pulse-mode circuits. (Dec 2013)** BTL 1 **Fundamental Mode Circuit** (i) The input variables change only when the circuit is stable. (ii) Only one input variable can change at a given time (iii) Inputs are levels and not pulses. **Pulse Mode Circuits** (i) The input variables are pulses instead of levels. (ii) The width of the pulses is long enough for the circuit to respond to the input. (iii) The pulse width must not be so long that it is still present after the new state is reached and cause a faulty change of state.(iv) No two pulses should arrive at the input lines simultaneously.
11	**What is Primitive Flow table? (Dec 2013)** BTL 1 A primitive flow table is a flow table with only one stable total state in each row.
12	**What are cycles and races? (June 2012)** BTL 1 A cycle occurs when an asynchronous circuit makes a transition through a series of unstable states. If a cycle does not contain a stable state, the circuit will go from one unstable to stable to another, until the inputs are changed. When 2 or more binary state variables change their value in response to a change in an input variable, race condition occurs in an asynchronous sequential circuit. In case of unequal delays, a race condition may cause the state variables to change in an unpredictable manner.
13	**Why is the pulse mode operation of asynchronous sequential circuits not very popular?** BTL 1 Because of the input variable pulse width restrictions, pulse mode circuits are difficult to design. For this reason the pulse mode operation of asynchronous sequential circuits is not very popular.
14	**Differentiae Static & Dynamic Hazard.** BTL 1 **Static 1-hazard:** The output may momentarily go to 0 when it should remain. **Static 0-hazard:** The output may momentarily go to 1 when it should remain 0. **Dynamic hazard** causes the output to change three or more times when it should change from 1 to 0 or from 0 to 1
15	**What is State Assignment?**BTL 1 (i)Assigning binary values to each state that is represented by letter symbol in the flow table of sequential circuit is called state assignment. (ii)The primary objective in choosing a proper binary state assignment in asynchronous circuit is the prevention of critical races
16	**Define Essential Hazard.** BTL 1 An essential Hazard is caused by unequal delays along two or more paths that originate from the same input. An excessive delay through an inverter circuit in comparison to the delay associated with the feedback path may cause such a hazard. Essential hazards cannot be corrected by adding redundant gates as in static hazards. To avoid essential hazard, each feedback loop must be handled with individual care to ensure that the delay in the feedback path is long enough compared to delays of other signals that originate from the input terminals.
17	**Define Flow table.** BTL 1 During the design of asynchronous sequential circuits, it is more convenient to name the states by letter symbols without making specific reference to their binary values, such a table is called a Flow table.
18	**Define Merger diagram**.BTL 1 The merger diagram is a graph in which each state is represented by a dot placed along the circumference of a circle. Lines are drawn between any two corresponding dots that form a compatible pair. All possible compatibles can be obtained from the merger diagram by observing the geometrical patterns in which states are connected to each other.

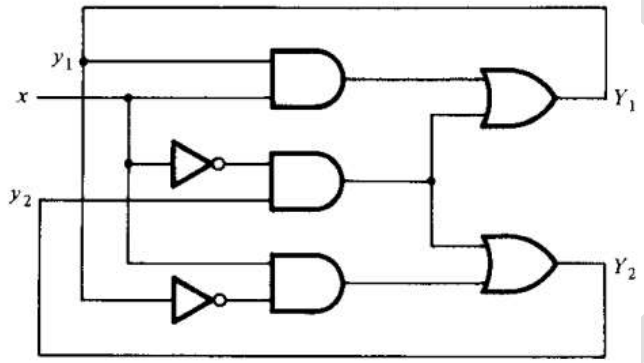
19	**Define Multiple row method.** BTL 1 In the multiple row assignment each state in the original flow table is replaced by two or more combinations of state variables. The state assignment map shows the multiple row assignment that can be used with any four- row flow table. 
20	**Define the term Maximal compatible.** BTL 1 The maximal compatible is a group of compatibles that contains all the possible combinations of compatible states. The maximal compatible can be obtained from a merger diagram.
21	**Define closed covering.** BTL 1 The condition that must be satisfied for row merging is that the set of chosen compatibles must cover all the states that must be closed. The set will cover all the states if it includes all the states of the original state table. The closure condition is satisfied if there are no implied states or if the implied states are included within the set. A closed set of compatibles that covers all the states is called a closed covering.
22	**Explain Shared Row method**. BTL 1 The method of making race free assignment by adding extra rows in the flow table is sometimes referred to as Shared Row method
23	**What is the need of state reduction in sequential circuit design?**BTL 1 (i)To reduce the number of flip-flops (ii)To reduce the number of gates in the combinational circuit that drives the flip-flop inputs
24	**What is the use of flip-flop excitation table?**BTL 1 If the transition from present state to next state is known in the design of sequential circuit, the flip-flop excitation table is used to find the flip-flop input conditions that will cause the required transition.
25	**List any two drawbacks of asynchronous circuits.**BTL 1 Race condition and Hazards. Circuit design is complicated.
	PART - B
1	**(i) Write the difference between synchronous and asynchronous sequential circuit. (5M)** BTL 5 **(ii) Outline the procedure for analyzing asynchronous sequential circuit. (8M) (April/May 2019)** BTL 5 **Ans: Refer Morris Mano, PG.NO: 573-576**

Sr. No.	Synchronous sequential circuits	Asynchronous sequential circuits
1.	In synchronous circuits, memory elements are clocked flip-flops.	In asynchronous circuits, memory elements are either unclocked flip-flops or time delay elements.
2.	In synchronous circuits, the change in input signals can affect memory element upon activation of clock signal.	In asynchronous circuits change in input signals can affect memory element at any instant of time.
3.	The maximum operating speed of clock depends on time delays involved.	Because of absence of clock, asynchronous circuits can operate faster than synchronous circuits.
4.	Easier to design.	More difficult to design.

Comparison between synchronous and asynchronous sequential circuits

(5M)

(ii) Analysis of asynchronous circuit



y_1y_2 \ x	0	1
00	0	0
01	1	0
11	1	1
10	0	1

(a) Map for $Y_1 = xy_1 + x'y_2$

y_1y_2 \ x	0	1
00	0	1
01	1	1
11	1	0
10	0	0

(b) Map for $Y_2 = xy_1' + x'y_2$

y_1y_2 \ x	0	1
00	(00)	01
01	11	(01)
11	(11)	10
10	00	(10)

(c) Transition table

(8M)

2 **(i) Discuss about the possible hazards and methods to avoid them in combinational circuits (7M)** BTL 5

Ans: Refer Morris Mano, PG.NO: 610-611

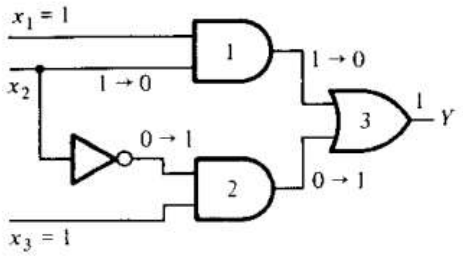
(ii) Discuss about the possible hazards in sequential circuits. (6M) BTL 5

Ans: Refer Morris Mano, PG.NO: 611-612

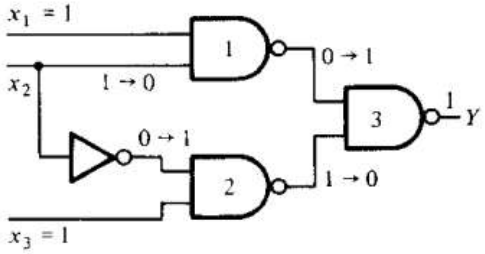
(i) Hazards in combinational circuit

This type of implementation may cause the output to go to 0 when it should remain a 1.
If the circuit is implemented in product of sums

$$Y = (x_1 + x_2')(x_2 + x_3)$$



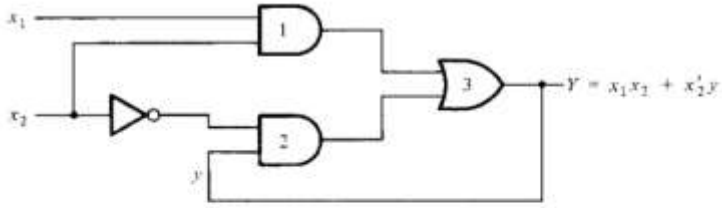
(a) AND-OR circuit



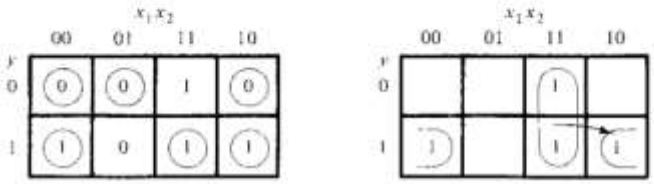
(b) NAND circuit

(7M)

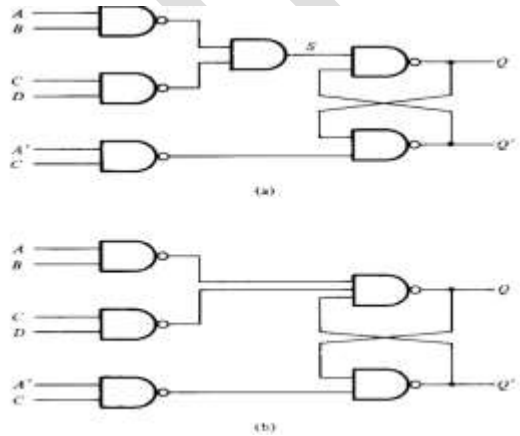
(ii) **Hazards in sequential circuits**



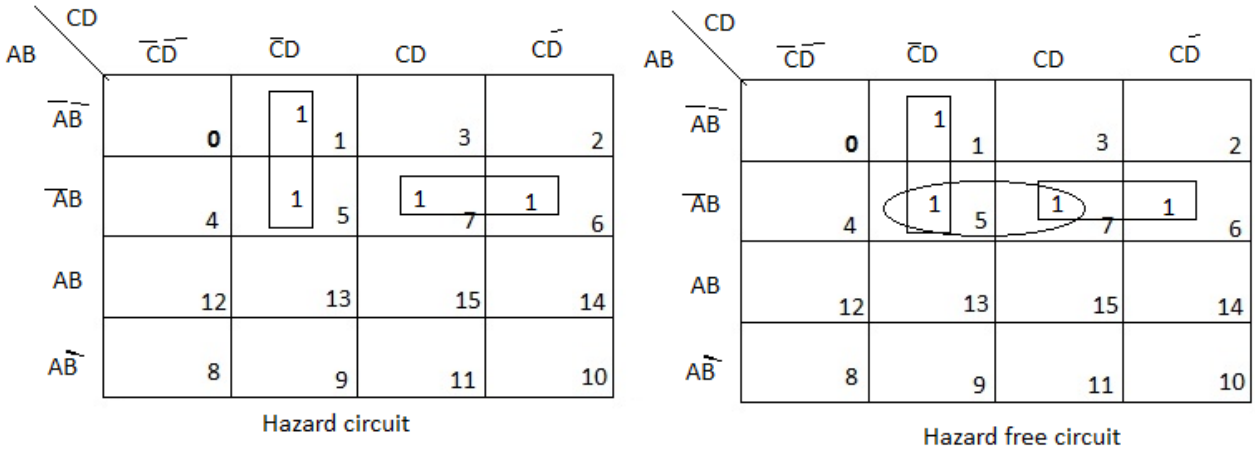
(a) Logic diagram



(6M)

3	**What are called as essential hazards? How does the hazard occurin sequential circuits? How can the same be eliminated usingSR latches? Give example. (13M) (Nov/Dec 2018)** BTL 5 **Ans: Refer Morris Mano, PG.NO: 613-614** Description of essential hazards (7M)  (6M)
4	**Find the circuit that has no static hazards and implement the Boolean function**

F(A,B,C,D)=∑m(1,5,6,7) (13M) (Nov/Dec 2018) BTL 6



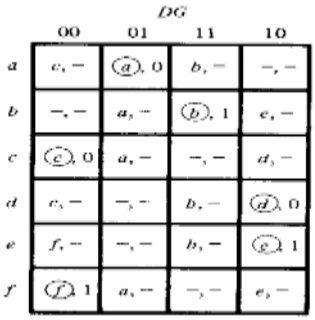
(13M)

5 **Discuss in detail the procedure for reducing the flow table with an example. (10M) (May/June 2016)** BTL6

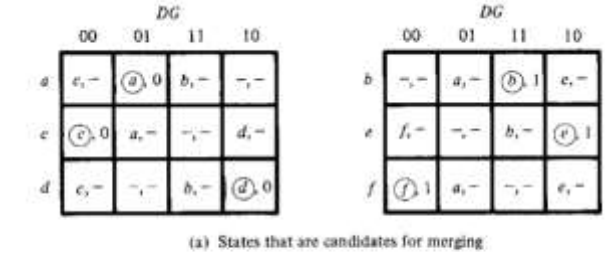
Ans: Refer Morris Mano, PG.NO: 590-593

Gated-Latch Total States

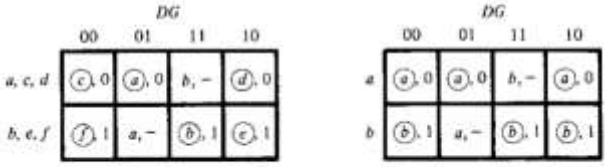
State	Inputs D	Inputs G	Output Q	Comments
a	0	1	0	D = Q because G = 1
b	1	1	1	D = Q because G = 1
c	0	0	0	After state a or d
d	1	0	0	After state c
e	1	0	1	After state b or f
f	0	0	1	After state e



(5M)



(a) States that are candidates for merging



(b) Reduced table (two alternatives)

Reduction of the primitive flow table

(5M)

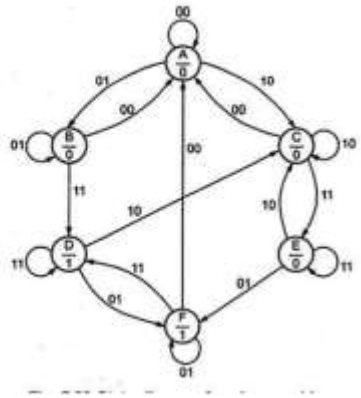
6 **Design an asynchronous sequential circuit that has two inputs X1and X2 and one output Z. When X1=0, the output Z is 0.The first change in X2 that occurs while is X1 will cause output Z to be 1.**

The output Z will remain X1 until returns to 0. (16M) (May/June 2016) BTL 6

Primitive flow table constructed from state diagram

Present state	Next state, Output Z for X_2X_1 inputs			
	00	01	11	10
A	(A), 0	B, –	–, –	C, –
B	A, –	(B), 0	D, –	–, –
C	A, –	–, –	E, –	(C), 0
D	–, –	F, –	(D), 1	C, –
E	–, –	F, –	(E), 0	C, –
F	A, –	(F), 1	D, –	–, –

State diagram

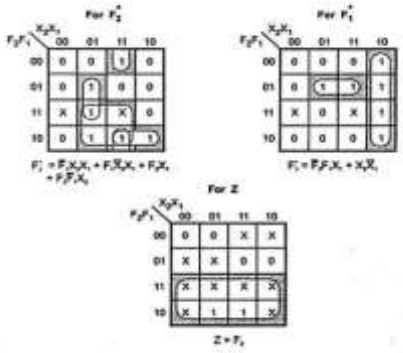


Flow table with state assignment

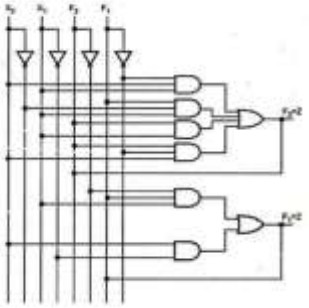
Present state Y_2 Y_1	Next state, Output Z for X_2X_1 inputs			
	00	01	11	10
S_0 0 0	(S_0), 0	(S_3), 0	S_2, –	S_1, –
S_1 0 1	S_0, –	S_3, –	(S_1), 0	(S_1), 0
S_2 1 0	S_0, –	(S_2), 1	(S_2), 1	S_3, –
S_3 1 1	–, –	S_2, –	–, –	S_1, –

(10M)

K-map simplification



Logic diagram



(6M)

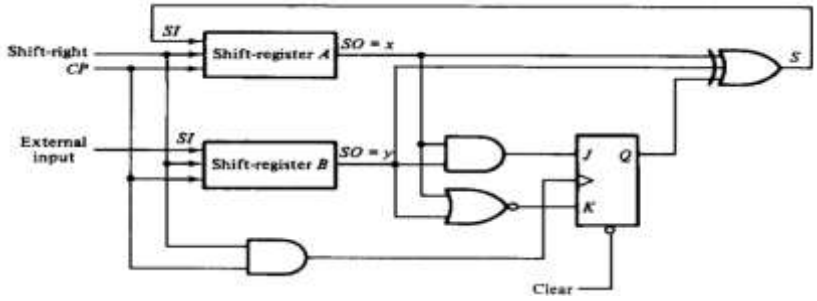
7 **Design a serial adder using a full adder and a flip flop. (15M) (Nov/Dec 2015)** BTL 6

Ans: Refer Morris Mano, PG.NO: 349-351

Excitation Table for a Serial Adder

Present State	Inputs		Next State	Output	Flip-Flop Inputs	
Q	x	y	Q	S	JQ	KQ
0	0	0	0	0	0	X
0	0	1	0	1	0	X
0	1	0	0	1	0	X
0	1	1	1	0	1	X
1	0	0	0	1	X	1
1	0	1	1	0	X	0
1	1	0	1	0	X	0
1	1	1	1	1	X	0

(10M)



(5M)

8 **(i) Explain the Race- free state assignment procedure. (5M)** BTL 5

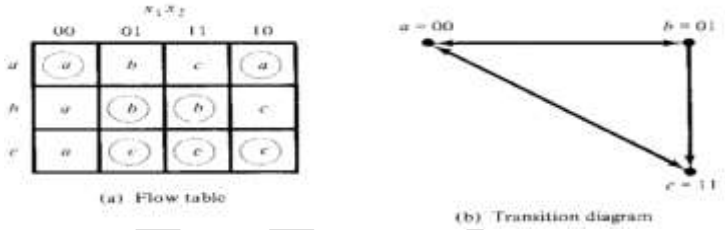
Ans: Refer Morris Mano, PG.NO: 604-609

(ii) Reduce the number of states in the following state diagram. Tabulated the reduced state table and Draw the reduced state diagram. (5M) (May 2015) BTL 5

Ans: Refer Morris Mano, PG.NO: 318-319

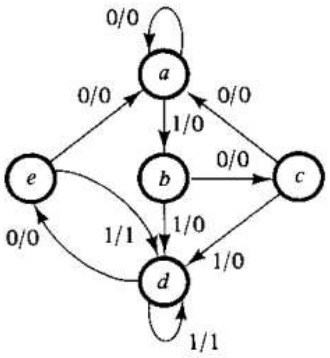
Present state	Next state	output
	x=0 x=1	x=0,x=1
a	a b	0 0
b	c d	0 0
c	a d	0 0
d	e f	0 1
e	a f	0 1
f	g f	0 1
g	a f	0 1

(i) Race- free state assignment procedure



(5M)

(ii) State reduction



(5M)

9

It is necessary to design a gated latch circuit with two inputs, G (gate) and D (data), and one output, Q. Binary information present at the D input is transferred to the Q output when G is equal to 1. The Q output will follow the D input as long as $G = 1$. When G goes to 0, the information that was present at the D input at the time the transition occurred is retained at the Q output. The gated latch is a memory element that accepts the value of D when $G = 1$ and retains this value after G goes to 0. Once $G = 0$, a change in D does not change the value of the output Q.

(13M) BTL 6

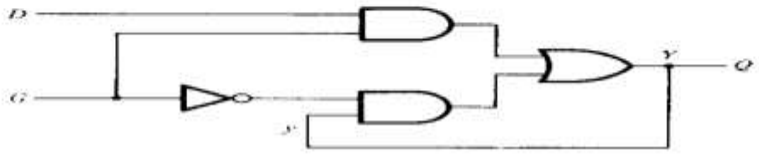
Ans: Refer Morris Mano, PG.NO: 590-594

Gated-Latch Total States

State	Inputs D	Inputs G	Output Q	Comments
a	0	1	0	$D = Q$ because $G = 1$
b	1	1	1	$D = Q$ because $G = 1$
c	0	0	0	After state a or d
d	1	0	0	After state c
e	1	0	1	After state b or f
f	0	0	1	After state e

(10M)

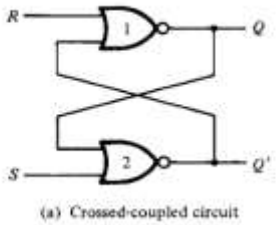
Transition table and output map for gated latch



(3M)

10 **Analyze SR asynchronous latch (13M)** BTL 5

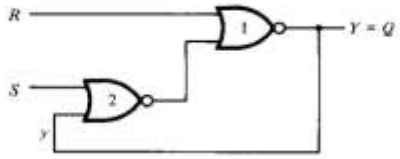
Ans: Refer Morris Mano, PG.NO: 582-586

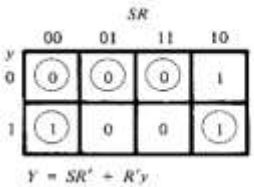


(a) Crossed-coupled circuit

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(After $SR = 10$)
0	1	0	1	
0	0	0	1	(After $SR = 01$)
1	1	0	0	

(b) Truth table





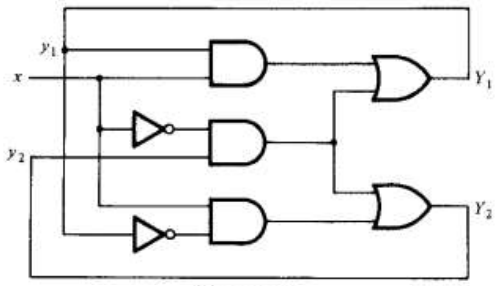
$Y = SR' + R'y$

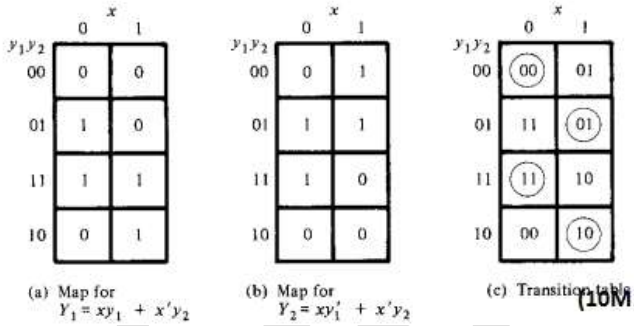
$Y = S + R'y$ when $SR = 0$

(13M)

11 **A synchronous sequential circuit is described by the following excitation and output function**

	Y=X1X2+(X1+X2)Y, Z=Y. (i) Draw the logic diagram of the circuit. (ii) derive the transition table and output map.(iii) describe the behavior of the circuit. (Dec 2014) BTL 5
	PART C
1	**Analyze the circuit shown below, draw transition diagram, flow table of the circuit. (15M)** BTL 6 **Ans: Refer Morris Mano, PG.NO: 574-576**





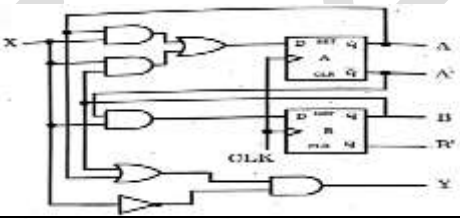
(15M

Present State		Next State $x=0$		Next State $x=1$	
0	0	0	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	1	1	1	1	0

$$Y_1 = xy_1 + x'y_2$$
$$Y_2 = xy_1' + x'y_2$$

(15M)

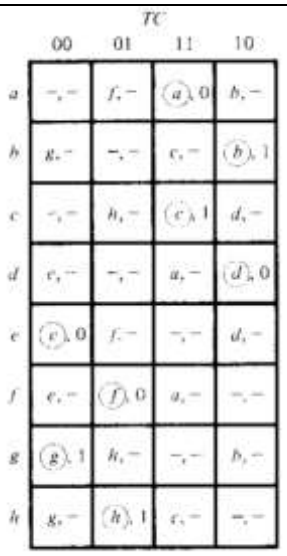
2	**Analyze the following clocked sequential circuit and obtain the state equations and state diagram. (15M) (Nov/Dec 2015)** BTL 6



3	It is necessary to design a negative-edge-triggered T flip-flop. The circuit has two inputs, T (toggle) and C (clock), and one output, Q. The output state is complemented if $T = 1$ and the clock C changes from 1 to 0 (negative-edge triggering). Otherwise, under any other input condition, the output Q remains unchanged. Although this circuit can be used as a flip-flop in clocked sequential circuits, the internal design of the flip-flop (as is the case with all other flip-flops) is an asynchronous problem. **(15M)** BTL 6 **Ans: Refer Morris Mano, PG.NO: 614-620**

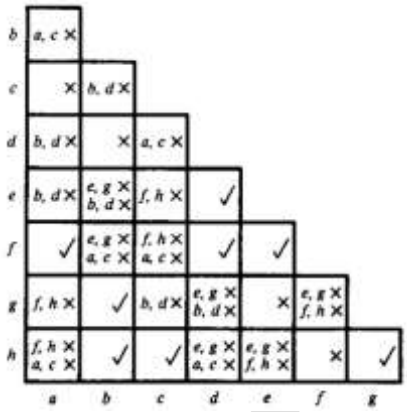
Specification of Total States

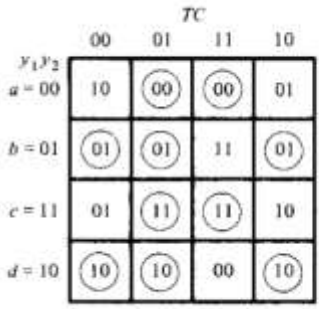
State	Inputs T	Inputs C	Output Q	Comments
a	1	1	0	Initial output is 0
b	1	0	1	After state *a*
c	1	1	1	Initial output is 1
d	1	0	0	After state *c*
e	0	0	0	After state *d* or *f*
f	0	1	0	After state *e* or *a*
g	0	0	1	After states *b* or *h*
h	0	1	1	After states *g* or *c*



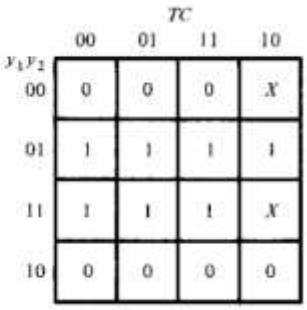
Primitive flow table

(10M)

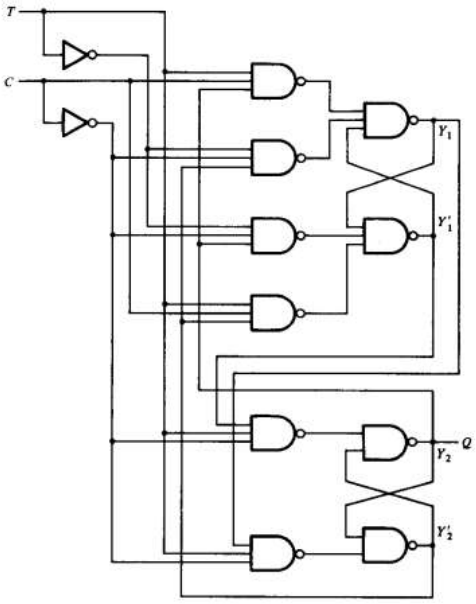




(a) Transition table



(b) Output map $Q = y_2$



(5M)

RAM – Memory Decoding – Error Detection and Correction - ROM - Programmable Logic Array – Programmable Array Logic – Sequential Programmable Devices

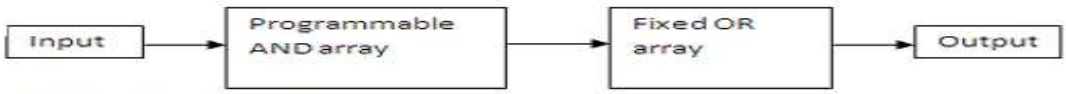
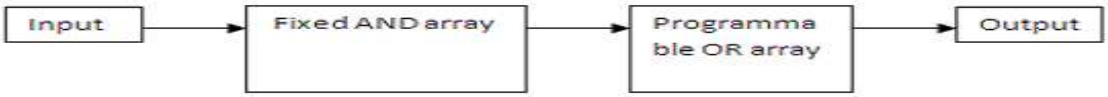
PART - A

Q. No	Question& Answer
1	**What are error detecting codes? Give examples. (Apr/May 2019)** BTL 1 Error-detecting codes are a sequence of numbers generated by specific procedures for detecting errors in data that has been transmitted over computer networks. Eg Parity Check,Checksum
2	**List the advantages of using sequential programmable devices. (Apr/May 2019)** BTL 1 Less board space, faster, lower power requirements (i.e., smaller power supplies), less costly assembly processes, higher reliability (fewer ICs and circuit connections means easier troubleshooting), and availability of design software.
3	**List the major difference between PLA and PAL. (Nov/Dec 2018)** BTL 1 The main difference among these two is that PAL can be designed with a collection of AND gates and fixed collection of OR gates whereas PLA can be designed with a programmable array of AND although a fixed collection of OR gate.
4	**What is field programmable logic array? (Nov/Dec 2018)** BTL 1 A field-programmable gate array (*FPGA*) is an integrated circuit (IC) that can be programmed in the field after manufacture. *FPGAs* are similar in principle to, but have vastly wider potential application than, programmable read-only memory (PROM) chips.
5	**Write short notes on PLA. (Nov/Dec 2015)** BTL 1 Programmable Logic Array (PLA) is a programmable logic device with a Programmable AND array and a programmable OR array. PLA can be used to implement complex logic circuits. It uses conventional symbol. It is more flexible than PAL
6	**What is memory address register? (Nov/Dec 2015)** BTL 1 MAR holds the memory location of data that needs to be accessed.
7	**How to detect double error and correct single error? (May 2015)** BTL 1 Single Bit Error Correction using parity bits. Double Bit Error Detection, which is somehow related to the even or odd parity of the bit sequence.
8	**Differentiate between EEPROM and PROM. (May 2015)** BTL 1

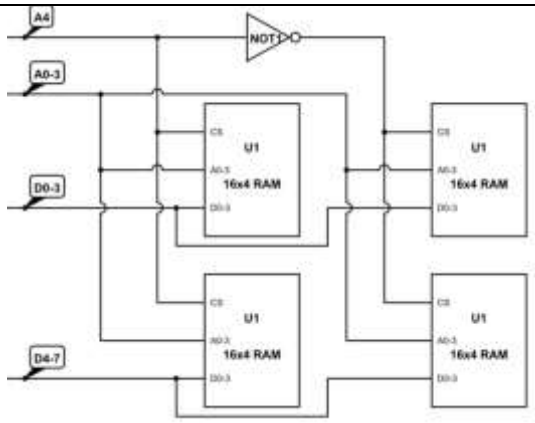
EEPROM	PROM
Reusable the programmable	One time programmable
Electrically erasable	Not erasable
Programmed in place (no need to remove from circuit board)	Using external for programming device

9	**What is a volatile memory? Give example. (Dec 2014)** BTL 1 Volatile memory means that any storage memory location can be accessed to read or write operation. RAM is volatile memory, so data will lost if power is switched off.
10	**What is memory decoding? (June 2014)** BTL 1 The memory IC used in a digital system is selected or enabled only for the range of addresses Assigned to it and this process is called memory decoding
11	**Define ASIC. (June 2014)** BTL 1 An ASIC (application-specific integrated circuit) is a microchip designed for a special application, such as a particular kind of transmission protocol or a hand-held computer.
12	**Distinguish between PAL and PLA. (June 2012/Dec 2014)** BTL 1

PLA	PAL

		In programmable logic array both AND and OR arrays are programmable.	In PAL OR arrays are fixed and AND arrays are programmable.
		It is costlier as compared to PAL	It is cheaper.
		It is complex than PAL	It is simple
		It can't easily be programmed	It is easy to program a PAL
13	**What is the difference between PROM and PLA?** BTL 1 The programmable array logic(PAL) is a programmable logic device with a fixed OR array and a programmable AND array. The programmable Read only Memory(PROM) is a programmable logic device with a fixed AND array and programmable OR array. Architecture: PAL  Architecture: PROM 		
14	**What is PLA and Its uses?** BTL 1 (i) PLA (Programmable Logic Array) is a Programmable Logic device with a programmable AND array and Programmable OR array.(ii) PLA can be used to implement complex logic circuits.(iii) It is more economical to use PLA rather than PROM to implement logic circuits that have more number of don't care conditions in order to reduce number of gates.		
15	**What are the major drawbacks of the EEPROM?**BTL 1 (i)COST: In EEPROM, the erasing and programming of an EEPROM can be done in circuit.(Without using separate UV light source and special PROM programmer unit). Because of this on-chip support circuitry the EEPROM is available with more cost. (ii) DENSITY: The high level integration of the EEPROM occupies more space. For example, 1-Mbit EEPROM requires about twice as much silicon as a 1-Mbit EPROM.		
16	**How many data inputs, data outputs and address inputs are needed for a 1024 *4 ROM?** BTL 1 No. of data inputs and outputs = 4*1024 =2^{10} No of address inputs = 10		
17	**Describe the basic functions of ROM and RAM.** BTL 1 **ROM:** Read only memory is used to store information permanently. The information cannot be altered. **RAM:** Random Access Memory is used to store information. The information can be read form it and the new information can be written into the memory.		
18	**What is Configurable Logic Block?** BTL 1 The programmable logic blocks in the Xilinx family of FPGAs are called configurable logic blocks (CLBs). The CLB of Xilinx 3000 series can be configured to perform any logic function of up to a maximum of seven variables.		
19	**Give the different types of RAM.** BTL 1 RAM can be classified into two types: (i)Static RAM: The storage elements used in this type RAM are latches (unclocked FFs).		

	(ii) Dynamic RAM: A dynamic RAM is one in which data are stored on capacitors which require periodic recharging (refreshing) to retain the data. RAMs are manufactured with either bipolar or MOS technologies. Bipolar RAMs are all static RAM. MOS RAM are available in both static and dynamic types .
20	**What is dynamic RAM cell? Draw its basic structure.** BTL 1 A dynamic RAM is one in which data are stored on capacitors which require periodic recharging (refreshing) to retain the data.
21	**What is Memory refresh?** BTL 1 Dynamic RAMs are fabricated using MOS technology. They store 1s and 0s as charges on a small MOS capacitor (typically a few Pico farads). Because if the tendency for these charges to leak of after a period of time, dynamics require periodic recharging of the memory cells This is called refreshing the dynamic RAM or memory refresh.
22	**What do you mean by PLD's?** BTL 1 PLDs: Programmable logic devices are the special type of IC's used by the USE and are programmed before use Different type of logic functions can be implemented using a single programmed IC chip of PLD's. PLD s can be reprogrammed because these are based on re-.writable memory technologies fuse links are used to programmed the PLD b the user according to the type of PLD to be manufactured.
23	**Compare SRAM and DRAM.** BTL 1 SRAM: Static RAM uses the flip-flop for its basic storage element. It is possible to store data as long as power is applied to the chip. It make use of cross coupled TTL multiemitter bipolar transistors or cross coupled MOSFETs for its construction. DRAM: Dynamic RAM make use of capacitive element for storing the data bit. Binary information is stored as charge. If charge is present at a capacitive element it represents a logic 1 and in the absence of the charge a logic 0 is stored. DRAM's consumes less power as compared to SRAM's..
24	**List out the different types of ROM.** BTL 1 ROM, PROM, EPROM, EEPROM
25	**A seven bit Hamming code is received as 1111110. What is the correct code?** BTL 1 C_1=1 C_2=1 C_4=1 The corrected code 1111111
	PART – B
1	**(i) Discuss briefly about RAM and its types. (6M)** BTL 5 **Ans: Refer Morris Mano, PG.NO: 400-402** **(ii) Explain the logical construction of a 256x8 RAM using 64x8 RAM chips. (7M) (April/May 2019)** BTL 6 **(i) Types of RAM** RAM is of two types•Static RAM (SRAM)•Dynamic RAM (DRAM) Static RAM (SRAM)The word staticindicates that the memory retains its contents as long as power remains applied. However, data is lost when the power gets down due to volatile nature. Dynamic RAM (DRAM)DRAM, unlike SRAM, must be continually refreshing order for it to maintain the data. (6M) **(ii) Logical construction of Memory**



(7M)

2 **(i) Given the 8 bit data word 11000100 generate the 13 bit word for the Hamming code that correctssingle errors and detect double errors. (7M) (Apr/May 2019)** BTL 5

Ans: Refer Morris Mano, PG.NO: 408-409

(ii) Implement the following two Boolean functions with a PLA. (6M) BTL 5

F1(A,B,C)=AB'+AC+A'BC'

F2(A,B,C)=(AC+BC)'

(i) Hamming code

Bit position:	1	2	3	4	5	6	7	8	9	10	11	12
	P_1	P_2	1	P_4	1	0	0	P_8	0	1	0	0

The 4 parity bits, P_1, P_2, P_4, and P_8, are in positions 1, 2, 4, and 8, respectively. The 8 bits of the data word are in the remaining positions. Each parity bit is calculated as follows:

$$P_1 = \text{XOR of bits } (3, 5, 7, 9, 11) = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$$

$$P_2 = \text{XOR of bits } (3, 5, 7, 10, 11) = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$P_4 = \text{XOR of bits } (5, 6, 7, 12) = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$P_8 = \text{XOR of bits } (9, 10, 11, 12) = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

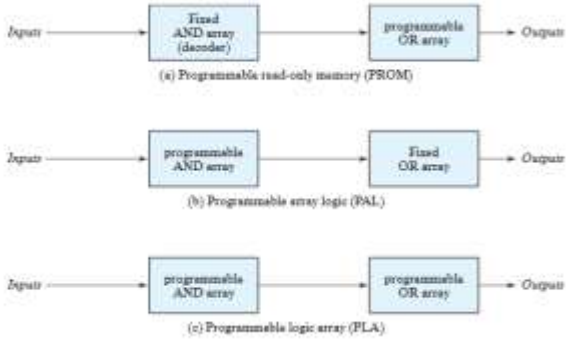
(7M)

(ii) PLA implementation (6M)

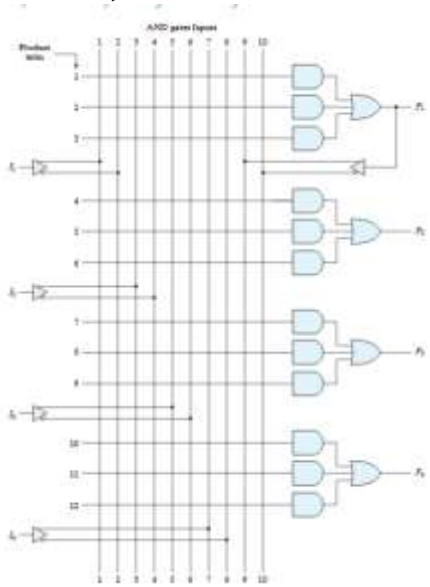
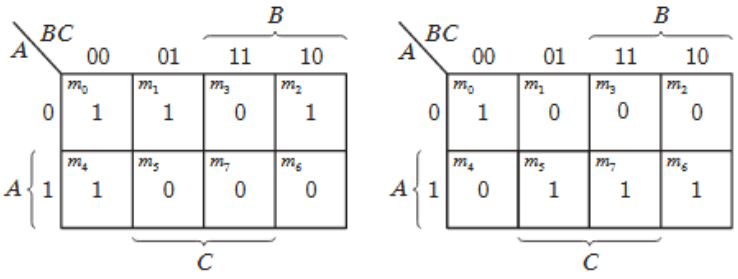
3 **Illustrate with neat sketch and describe the categories of ROM. (13M) (Nov/Dec 2018)** BTL 5

Ans: Refer Morris Mano, PG.NO: 415-416

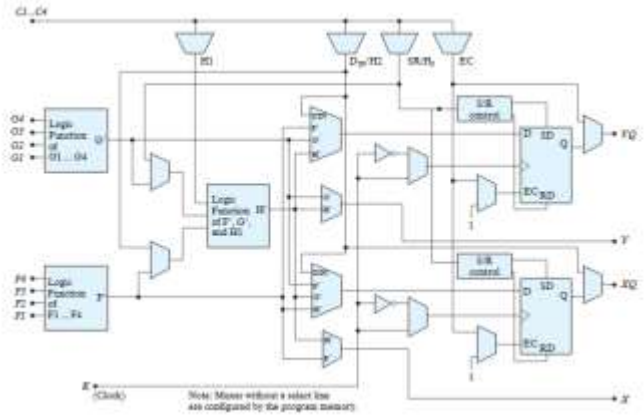
For small quantities, it is more economical to use a second type of ROM called pro-grammable read only memory, or PROM. (10M)



(3M)

4	**With neat diagrams describe the working principle of programmable array logic. (13M) (Nov/Dec 2018)** BTL 5 **Ans: Refer Morris Mano, PG.NO: 420-421** The PAL is a programmable logic device with a fixed OR array and a programmable AND array. Because only the AND gates are programmable, the PAL is easier to pro-gram than, but is not as flexible as, the PLA. (10M)  (3M)
5	**Implement the switching functions. (13M) (May/June 2016)** BTL 5 **F1(A,B,C)=∑(0,1,2,4)** **F2(A,B,C)=∑(0,5,6,7)** **Ans: Refer Morris Mano, PG.NO: 419-420** $F_1 = (AB + AC + BC)'$ $F_2 = AB + AC + A'B'C'$ (3M) PLA programming table Product term / Inputs A B C / Outputs (C) F_1 (T) F_2 AB 1 1 1 – 1 1 AC 2 1 – 1 1 1 BC 3 – 1 1 1 – $A'B'C'$ 4 0 0 0 – 1  (10M)
6	**Write short notes on FPGA. (15M) (April/May 2015)** BTL 5 **Ans: Refer Morris Mano, PG.NO: 428-430** The basic architecture of Spartan and earlier device families consists of an array of configurable logic blocks (CLBs), a variety of local and global routing resources, and input–output (I/O) blocks (IOBs), programmable I/O buffers, and an SRAM based configuration memory. (7M)

.



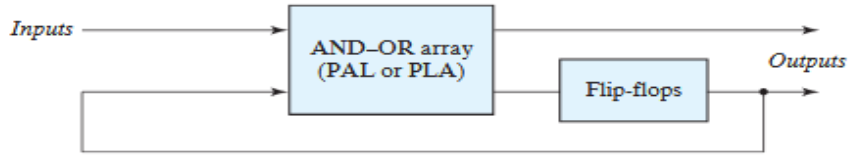
(8M)

7 **Briefly discuss the sequential programmable devices. (15M) (Apr/May 2015)** BTL 5

Ans: Refer Morris Mano, PG.NO: 424-427

1. Sequential (or simple) programmable logic device (SPLD)
2. Complex programmable logic device (CPLD)
3. Field-programmable gate array (FPGA)

The sequential PLD is sometimes referred to as a simple PLD to differentiate it from the complex PLD. The SPLD includes flip-flops, in addition to the AND–OR array, within the integrated circuit chip. The result is a sequential circuit as shown in Fig. 7.18. A PAL or PLA is modified by including a number of flip-flops connected to form a register. The circuit outputs can be taken from the OR gates or from the outputs of the

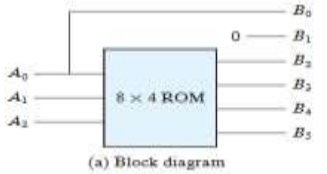


(15M)

8 **Design a combinational circuit using ROM that accepts a three bit binary number and outputs a binary number and outputs a binary number equal to the square of the input number. (15M) (Nov/Dec 2015)** BTL 5

Ans: Refer Morris Mano, PG.NO: 414

Inputs			Outputs						
A_2	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	Decimal
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49



(a) Block diagram

A_2	A_1	A_0	B_5	B_4	B_3	B_2
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	1	0
0	1	1	0	1	0	0
1	0	0	1	0	0	0
1	0	1	1	0	0	1
1	1	0	1	1	0	0
1	1	1	1	1	0	0

(b) ROM truth table

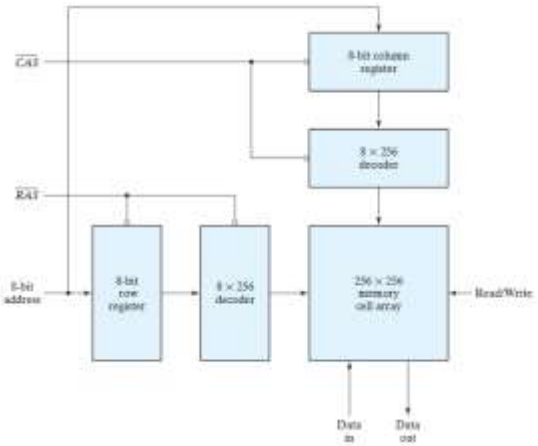
(15M)

9 **Explain address multiplexing in detail. (15M) (May/June 2014)** BTL 5

Ans: Refer Morris Mano, PG.NO: 405-406

The 16-bit address is applied to the DRAM in two steps using RAS and CAS. Initially, both strobes are in the 1 state. The 8-bit row address is applied to the address inputs and RAS is changed to 0. This loads the row address into the row address register. RAS also enables the row decoder so that it can decode the row address and select one row of the array. After a time equivalent to the settling time of the row selection, RAS goes back to the 1 level. The 8-bit column address is then applied to the address inputs, and CAS is driven to the 0 state. This transfers the column address into the column register and enables the column decoder. Now the two parts of the address are in their respective registers, the decoders have decoded them to select the one cell corresponding to the row and column address, and a read or write operation can be performed on that cell. CAS must go back to the 1 level before initiating another memory operation.

(10M)



(5M)

10 **Explain error detecting and correcting codes in detail. (13M)** BTL 5

Ans: Refer Morris Mano, PG.NO: 407-410

One of the most common error-correcting codes used in RAMs was devised by R. W. Hamming. In the Hamming code, k parity bits are added to an n-bit data word, forming a new word of $n + k$ bits. The bit positions are numbered in sequence from 1 to $n + k$. Those positions numbered as a power of 2 are reserved for the parity bits. The remaining bits are the data bits. The code can be used with words of any length. Before giving the general characteristics of the code, we will illustrate its operation with a data word of eight bits.

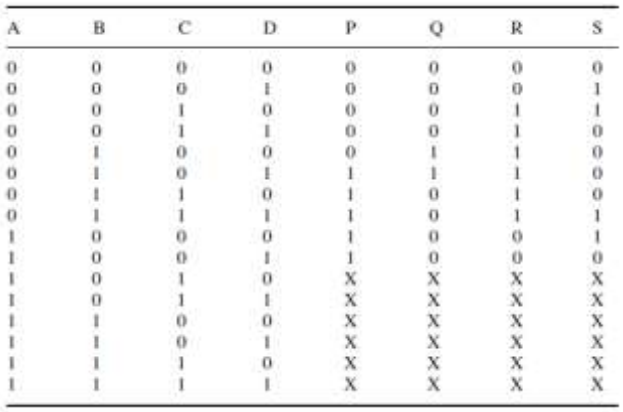
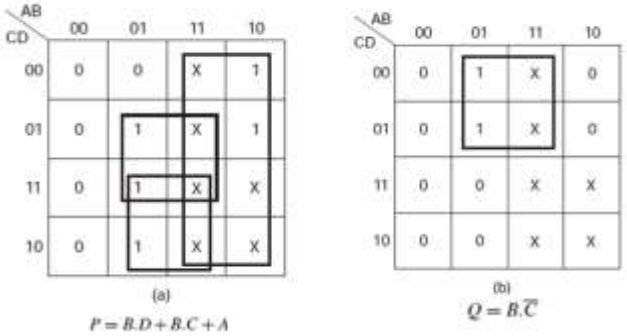
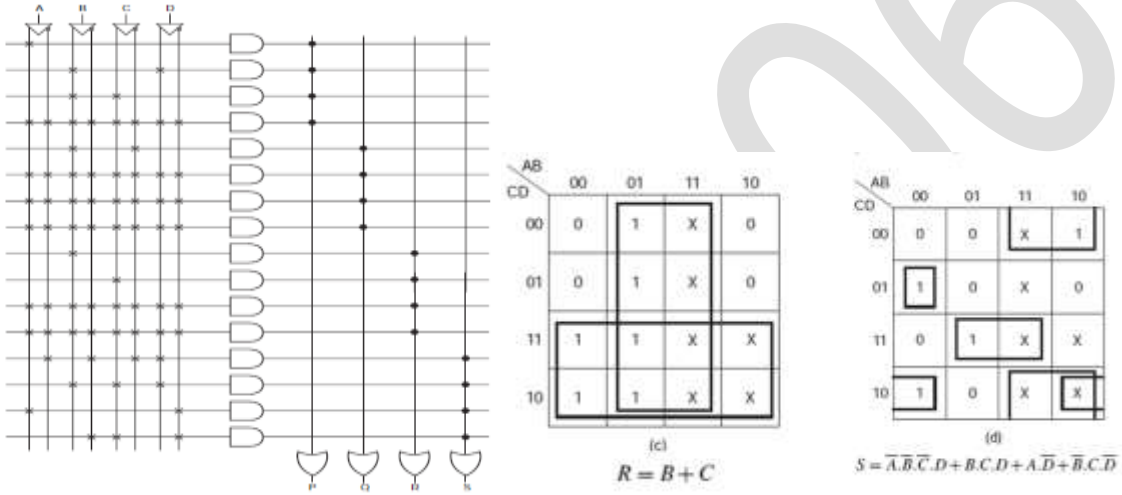
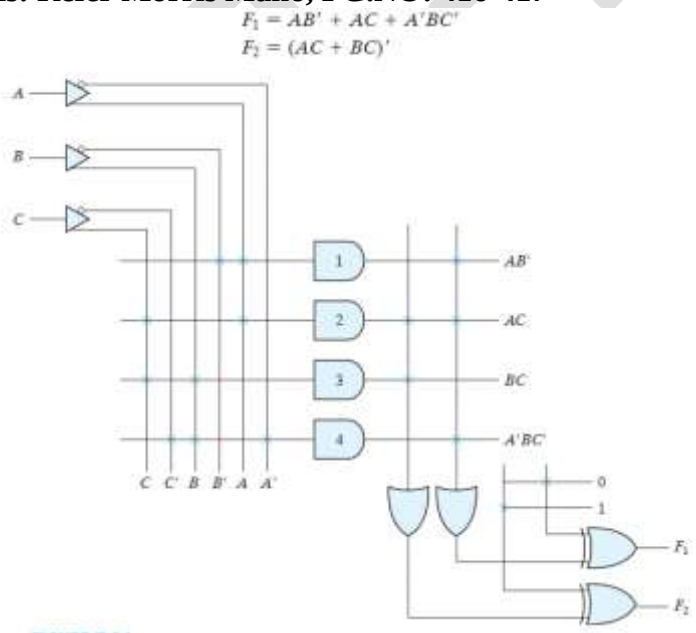
(13M)

11 **The following messages have been coded in the even parity hamming code and transmitted through a noisy Channel. Decode the messages, assuming that at most a single error has occurred in each code word. (13M)** BTL 5

(i)1001001 (ii)0111001 **(**13M)

PART C

1 **Design a BCD to gray code converter and implement using suitable PLA. (15M)** BTL5

	  (7M)  (8M)
2	**Implement the given Boolean function using PLA. (15M)** BTL5 **Ans: Refer Morris Mano, PG.NO: 416-417**  (15M)
3	**Implement the given Boolean function using PAL. (15M)** BTL5

Ans: Refer Morris Mano, PG.NO: 422-423

$w(A, B, C, D) = \Sigma(2, 12, 13)$

$x(A, B, C, D) = \Sigma(7, 8, 9, 10, 11, 12, 13, 14, 15)$

$y(A, B, C, D) = \Sigma(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$

$z(A, B, C, D) = \Sigma(1, 2, 8, 12, 13)$

Simplifying the four functions to a minimum number of terms results in the following Boolean functions:

$$w = ABC' + A'B'CD'$$

$$x = A + BCD$$

$$y = A'B + CD + B'D'$$

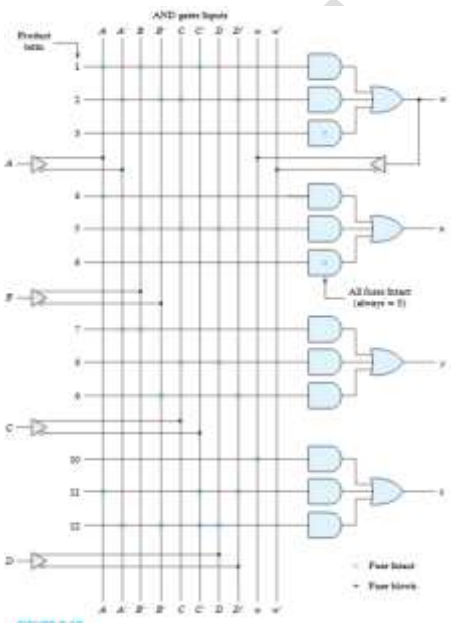
$$z = ABC' + A'B'CD' + AC'D' + A'B'C'D$$

$$= w + AC'D' + A'B'C'D$$

PAL Programming Table

Product Term	AND Inputs A	B	C	D	w	Outputs
1	1	1	0	—	—	$w = ABC' + A'B'CD'$
2	0	0	1	0	—	
3	—	—	—	—	—	
4	1	—	—	—	—	$x = A + BCD$
5	—	1	1	1	—	
6	—	—	—	—	—	
7	0	1	—	—	—	$y = A'B + CD + B'D'$
8	—	—	1	1	—	
9	—	0	—	0	—	
10	—	—	—	—	1	$z = w + AC'D' + A'B'C'D$
11	1	—	0	0	—	
12	0	0	0	1	—	

(10M)



(5M)

CS8391 DATA STRUCTURES **L T P C**

3 0 0 3

OBJECTIVES:

- To understand and apply the algorithm analysis techniques.
- To critically analyze the efficiency of alternative algorithmic solutions for the same problem
- To understand different algorithm design techniques.
- To understand the limitations of Algorithmic power.

UNIT I LINEAR DATA STRUCTURES – LIST 9

Abstract Data Types (ADTs) – List ADT – array-based implementation – linked list implementation —singly linked lists- circularly linked lists- doubly-linked lists – applications of lists –Polynomial Manipulation – All operations (Insertion, Deletion, Merge, Traversal).

UNIT II LINEAR DATA STRUCTURES – STACKS, QUEUES 9

Stack ADT – Operations - Applications - Evaluating arithmetic expressions- Conversion of Infix to postfix expression - Queue ADT – Operations - Circular Queue – Priority Queue – de Queue – applications of queues.

UNIT III NON LINEAR DATA STRUCTURES – TREES 9

Tree ADT – tree traversals - Binary Tree ADT – expression trees – applications of trees – binary search tree ADT –Threaded Binary Trees- AVL Trees – B-Tree - B+ Tree - Heap – Applications of heap.

UNIT IV NON LINEAR DATA STRUCTURES - GRAPHS 9

Definition – Representation of Graph – Types of graph - Breadth-first traversal - Depth-first traversal – Topological Sort – Bi-connectivity – Cut vertex – Euler circuits – Applications of graphs.

UNIT V SEARCHING, SORTING AND HASHING TECHNIQUES 9

Searching- Linear Search - Binary Search. Sorting - Bubble sort - Selection sort - Insertion sort - Shell sort – Radix sort. Hashing- Hash Functions – Separate Chaining – Open Addressing – Rehashing – Extendible Hashing.

TOTAL: 45 PERIODS

OUTCOMES:

At the end of the course, the student should be able to:

- Implement abstract data types for linear data structures.
- Apply the different linear and non-linear data structures to problem solutions.
- Critically analyze the various sorting algorithms.

TEXT BOOKS:

1. Mark Allen Weiss, -Data Structures and Algorithm Analysis in C|, 2nd Edition, Pearson Education,1997.
2. Reema Thareja, -Data Structures Using C‖, Second Edition , Oxford University Press, 2011

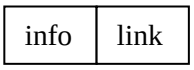
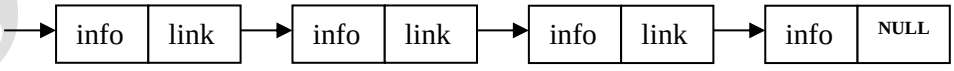
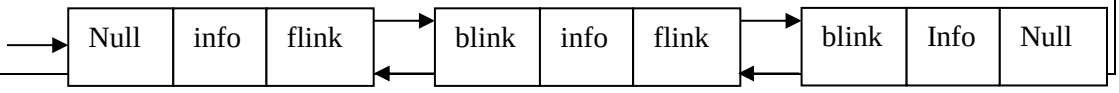
REFERENCES:

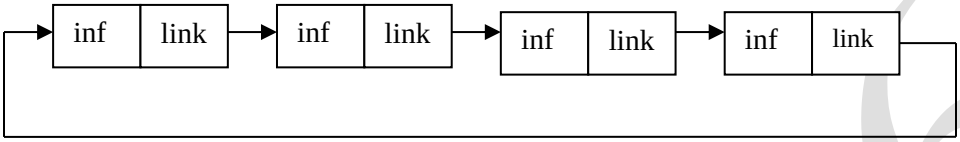
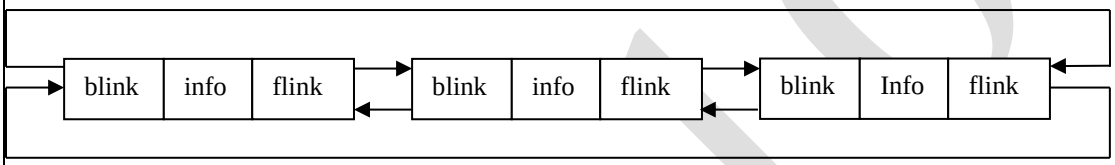
1. Thomas H. Cormen, Charles E. Leiserson, Ronald L.Rivest, Clifford Stein, -Introduction to Algorithms", Second Edition, Mcgraw Hill, 2002.
2. Aho, Hopcroft and Ullman, -Data Structures and Algorithms|, Pearson Education,1983.
3. Stephen G. Kochan, -Programming in C|, 3rd edition, Pearson Education.
4. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, -Fundamentals of Data Structures in C|, Second Edition, University Press, 2008

Subject Code: CS8391 **Subject Name – Data Structures**

Subject Handler: S. Sudha Mercy **Sem / Year: III/Second Year**

UNIT I -INTRODUCTION	
Abstract Data Types (ADTs) – List ADT – array-based implementation – linked list implementation—singly linked lists- circularly linked lists- doubly-linked lists – applications of lists –Polynomial Manipulation – All operations (Insertion, Deletion, Merge, Traversal).	
Q.NO	**PART* A**
1.	**Define: data structure.** BTL1 A data structure is a way of storing and organizing data in the memory for efficient usage. The way information is organized in the memory of a computer
2.	**Give few examples for data structures.** BTL1 Arrays, stacks, queue, list, tree, graph, set, map, table and deque.
3.	**What are the different types of data structures?** BTL1 i) Primitive ii) Composite iii) Abstract
4.	**What are primitive data types?** BTL1 The basic building blocks for all data structures are called primitive data types. (e.g) int, float, char, double, Boolean
5.	**What are composite data types?** BTL1 Composite data types are composed of more than one primitive data type. (e.g) array,structure,union
6.	**What is meant by an abstract data type?(April/May 2017)** BTL1 An ADT is a mathematical model for a certain class of data structures that have similar behavior. (e.g) list, stack, queue
7.	**How data structures can be categorized based on data access?** BTL1 **Linear –** list, stack, queue **Non-linear-** heap, tree, graph
8.	**State the difference between linear and non-linear data structures. (Nov/Dec 2018)** BTL2 The main difference between linear and nonlinear data structures lie in the way they organize data elements. In linear data structures, data elements are organized sequentially and therefore they are easy to implement in the computer's memory. In nonlinear data structures, a data element can be attached to several other data elements to represent specific relationships that exist among them. Due to this it might be difficult to be implemented in computer's linear memory.
9.	**List a few real-time applications of data structures.** BTL1

	• Undo and redo feature - stack • Decision making - graph • Printer (printing jobs) – queue • Directory structure- trees • Communication networks- graphs
10.	**Define List.** BTL1 The general form of the list is a_1, a_2, a_3… a_n. The size of the list is 'n'. Any element in the list at the position i is defined to be at a_i, a_{i+1} the successor of a_i , and a_{i-1} is the predecessor of a_i. a_1 doesn't have predecessor and a_n doesn't have successor.
11.	**What are the various operations done on List ADT?(April/May 2016)** BTL1 The operations done under List ADT are Print list, Insert, Delete, FindPrevious, Find k^{th}, Find, MakeEmpty, IsLast and IsEmpty.
12	**What are the different ways to implement list?** BTL1 • Array implementation of list • Linked list implementation of list • Cursor implementation of list
13	**Arrays are not used to implement lists. Why?** BTL2 • Requires that the list size to be known in advance • Running time for insertions and deletions is slow
14	**What are the advantages in the array implementation of list?(April/May2017)** BTL1 • Print list operation can be carried out at linear time • Finding K^{th} element takes a constant time
15	**What are the disadvantages in the array implementation of list?** BTL1 The running time for insertions and deletions is so slow and the list size must be known in advance.
16	**Define node.** BTL1 A node consists of two fields namely an information field called INFO and a pointer field called LINK. The INFO field is used to store the data and the LINK field is used to store the address of the next field. 
17	**What is a linked list?** BTL1 Linked list is series of nodes, which are not necessarily adjacent in memory. Each node contains a data element and a pointer to the next node. 
18	**What is a doubly linked list?** BTL1 In a doubly linked list, along with the data field there will be two pointers one pointing the next node(flink) and the other pointing the previous node(blink). 

19	**Define circularly linked list. (April/May 2017)** BTL1 In a singly circular linked list the last node's link points to the first node of the list. 
20	**Define double circularly linked list?** BTL1 In a circular doubly linked list the last node's forward link points to the first ode of the list, and the first node's back link points to the last node of the list. 
21	**Mention the disadvantages of circular list.** BTL2 The disadvantage of using circular list is • It is possible to get into an infinite loop. • It is not possible to detect the end of the list.
22	**What are the advantages of doubly linked list over singly linked list?(April/May 2019)** BTL1 The doubly linked list has two pointer fields. One field is previous link field and another is next link field. Because of these two pointer fields we can access any node efficiently whereas in singly linked list only one pointer field is there which stores forward pointer.
23	**Why is the linked list used for polynomial arithmetic?** BTL1 We can have separate coefficient and exponent fields for representing each term of polynomial. Hence there is no limit for exponent. We can have any number as an exponent.
24	**What is the advantage of linked list over arrays? (NOV/DEC 2018)** BTL1 The linked list makes use of the dynamic memory allocation. Hence the user can allocate or de allocate the memory as per his requirements. On the other hand, the array makes use of the static memory location. Hence there are chances of wastage of the memory or shortage of memory for allocation.
25	**What is the basic purpose of header of the linked list?** BTL1 The header node is the very first node of the linked list. Sometimes a dummy value such - 999 is stored in the data field of header node.This node is useful for getting

	the starting address of the linked list.
26	**State the advantage of an ADT? (NOV/DEC 2018)** BTL2 **Change:** the implementation of the ADT can be changed without making changes in the client program that uses the ADT. **Understandability:** ADT specifies what is to be done and does not specify the implementation details. Hence code becomes easy to understand due to ADT. **Reusability:** the ADT can be reused by some program in future
27	**State the properties of LIST abstract data type with suitable example.** BTL2 Various properties of LIST abstract data type are • It is linear data structure in which the elements are arranged adjacent to each other. • It allows to store single variable polynomial. • If the LIST is implemented using dynamic memory, then it is called linked list. Example of LIST are- stacks, queues, linked list.
28	**What is static linked list? State any two applications of it.** BTL1 • The linked list structure which can be represented using arrays is called static linked list. • It is easy to implement, hence for creation of small databases, it is useful. • The searching of any record is efficient, hence the applications in which the record need to be searched quickly, the static linked list are used.

	PART B
1	**Derive an ADT to perform insertion and deletion in a singly linked list.(13) (Nov 10) (NOV/DEC 2018)** BTL2 **Answer Pg no:171-175 in Reema Theraja** **Definition of Linked List(2M)** • Linked List can be defined as collection of objects called **nodes** that are randomly stored in the memory. • A node contains two fields i.e. data stored at that particular address and the pointer which contains the address of the next node in the memory. **Insertion(6M)** The insertion into a singly linked list can be performed at different positions. Based on the position of the new node being inserted, the insertion is categorized into the following categories. A node can be added in three ways • At the front of the linked list • After a given node • At the end of the linked list. **Deletion(5M)** To delete a node from linked list, do following steps. • Find previous node of the node to be deleted • Change the next of previous node. • Free memory for the node to be deleted.

2.	**Explain the steps involved to reverse the linked list. (13M)** BTL3
	Answer Pg no:171-175 in Reema Theraja
	Steps involved to reverse the elements in the linked list(7M)
	• Count the number of nodes in the linked list.
	• Declare an array with the number of nodes as its size.
	• Start storing the value of nodes of the linked list from the end of the array i.e. reverse manner.
	• Print k values from starting of the array.
	Algorithm(6M)
	// Structure of a node
	struct Node {
	int data;
	Node* next;
	};
	
	// Function to get a new node
	Node* getNode(int data){
	// allocate space

```
    Node* newNode = new Node;

    // put in data
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

// Function to print the last k nodes
// of linked list in reverse order
void printLastKRev(Node* head,
                   int& count, int k) {
    struct Node* cur = head;

    while(cur != NULL){
        count++;
        cur = cur->next;
    }

    int arr[count], temp = count;
    cur = head;

    while(cur != NULL){
        arr[--temp] = cur->data;
        cur = cur->next;
    }

    for(int i = 0; i < k; i++)
        cout << arr[i] << " ";
}
  //
// Driver code
int main()
{
    // Create list: 1->2->3->4->5
    Node* head = getNode(1);
    head->next = getNode(2);
    head->next->next = getNode(3);
    head->next->next->next = getNode(4);
    head->next->next->next->next = getNode(5);
    head->next->next->next->next->next = getNode(10);

    int k = 4, count = 0;

    // print the last k nodes
    printLastKRev(head, count, k);

    return 0;
```

	} Example: **Input :** list: 1->2->3->4->5, K = 2 **Output :** 5 4 3 2 1
3.	**Write an algorithm for inserting and deleting an element from Circular linked list. (13M)(NOV/DEC 2018)** BTL2 **Answer Pg no:187-195 Reema Theraja** **Definition(2M)** In a singly linked list, for accessing any node of linked list, we start traversing from the first node. If we are at any node in the middle of the list, then it is not possible to access nodes that precede the given node. This problem can be solved by slightly altering the structure of singly linked list. **Insertion(6M)** A node can be added in three ways: • Insertion in an empty list • Insertion at the beginning of the list • Insertion at the end of the list • Insertion in between the nodes **Algorithm for Inserting an element from circularly linked list:** **Insertion in an empty List:** Initially when the list is empty, *last* pointer will be NULL. **Insertion at the beginning of the list:** To Insert a node at the beginning of the list, follow these step: Step 1:Create a node, say T. Step 2:Make T -> next = last -> next. Step 3:last -> next = T. **Insertion at the end of the list:** To Insert a node at the end of the list, follow these step: Step 1: Create a node, say T. Step 2: Make T -> next = last -> next; Step 3: last -> next = T. Step 4: last = T. **Insertion in between the nodes**: To Insert a node at the end of the list, follow these step: Step 1: Create a node, say T. Step 2: Search the node after which T need to be insert, say that node be P. Step 3: Make T -> next = P -> next; Step 4: P -> next = T. **Algorithm for deleting an element from circularly linked list(5M)** **Case 1**: List is empty. • If the list is empty we will simply return. **Case 2**:List is not empty • If the list is not empty then we define two pointers **curr** and **prev** and initialize the pointer **curr** with the **head** node. • Traverse the list using **curr** to find the node to be deleted and before moving curr to next node, everytime set prev = curr. • If the node is found, check if it is the only node in the list. If yes, set head = NULL and free(curr). • If the list has more than one node, check if it is the first node of the list. Condition to check this(curr == head). If yes, then move prev until it reaches the last node.

After prev reaches the last node, set head = head -> next and prev -> next = head. Delete curr.

- If curr is not first node, we check if it is the last node in the list. Condition to check this is (curr -> next == head).
- If curr is the last node. Set prev -> next = head and delete the node curr by free(curr).
- If the node to be deleted is neither the first node nor the last node, then set prev -> next = temp -> next and delete curr.

4. **Explain the algorithm for the reverse operations on doubly linked list. (13M) (April/May 2019)(Nov 09)**

Answer Pg no:180-187 Reema Theraja

Explanation(5M)

swap prev and next pointers for all nodes, change prev of the head (or start) and change the head pointer in the end.

Algorithm for reversing doubly linked list:(8M)

```
/* Function to reverse a Doubly Linked List */
void reverse(struct Node **head_ref)
{
    struct Node *temp = NULL;
    struct Node *current = *head_ref;

    /* swap next and prev for all nodes of
      doubly linked list */
    while (current !=  NULL)
    {
      temp = current->prev;
      current->prev = current->next;
      current->next = temp;
      current = current->prev;
    }

    /* Before changing head, check for the cases like empty
       list and list with only one node */
    if(temp != NULL )
       *head_ref = temp->prev;
}



/* UTILITY FUNCTIONS */
/* Function to insert a node at the beginging of the Doubly Linked List */
void push(struct Node** head_ref, int new_data)
{
   /* allocate node */
   struct Node* new_node =
          (struct Node*) malloc(sizeof(struct Node));

   /* put in the data  */
   new_node->data  = new_data;
```

```
  /* since we are adding at the begining,
    prev is always NULL */
  new_node->prev = NULL;

  /* link the old list off the new node */
  new_node->next = (*head_ref);

  /* change prev of head node to new node */
  if((*head_ref) !=  NULL)
    (*head_ref)->prev = new_node ;

  /* move the head to point to the new node */
  (*head_ref)    = new_node;
}

/* Function to print nodes in a given doubly linked list
  This function is same as printList() of singly linked lsit */
void printList(struct Node *node)
{
 while(node!=NULL)
 {
  printf("%d ", node->data);
  node = node->next;
 }
}

/* Drier program to test above functions*/
int main()
{
 /* Start with the empty list */
 struct Node* head = NULL;

 /* Let us create a sorted linked list to test the functions
  Created linked list will be 10->8->4->2 */
 push(&head, 2);
 push(&head, 4);
 push(&head, 8);
 push(&head, 10);

 printf("\n Original Linked list ");
 printList(head);

 /* Reverse doubly linked list */
 reverse(&head);

 printf("\n Reversed Linked list ");
 printList(head);

 getchar();
}
```

	PART C

1	**Explain with algorithms to perform the insertion and deletion in doubly linked list (13M)(May 10)** BTL2
	Answer Pg no:180-187 Reema Theraja
	Definition(2M)
	A **D**oubly **L**inked **L**ist (DLL) contains an extra pointer, typically called *previous pointer*, together with next pointer and data which are there in singly linked list.
	Insertion(6M)
	A node can be added in four way:
	• At the front of the DLL
	• After a given node.
	• At the end of the DLL
	• Before a given node.
	Add a node at the front:
	void push(struct Node** head_ref, int new_data)
	{
	/* 1. allocate node */
	struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
	
	/* 2. put in the data */
	new_node->data = new_data;
	
	/* 3. Make next of new node as head and previous as NULL */
	new_node->next = (*head_ref);
	new_node->prev = NULL;
	
	/* 4. change prev of head node to new node */
	if ((*head_ref) != NULL)
	(*head_ref)->prev = new_node;
	
	/* 5. move the head to point to the new node */
	(*head_ref) = new_node;
	}
	Add a node after a given node
	void insertAfter(struct Node* prev_node, int new_data)
	{
	/*1. check if the given prev_node is NULL */
	if (prev_node == NULL) {
	printf("the given previous node cannot be NULL");
	return;
	}
	
	/* 2. allocate new node */
	struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));

```
   /* 3. put in the data  */
   new_node->data = new_data;

   /* 4. Make next of new node as next of prev_node */
   new_node->next = prev_node->next;

   /* 5. Make the next of prev_node as new_node */
   prev_node->next = new_node;

   /* 6. Make prev_node as previous of new_node */
   new_node->prev = prev_node;

   /* 7. Change previous of new_node's next node */
   if (new_node->next != NULL)
      new_node->next->prev = new_node;
}
```

Add a node at the end

```
void append(struct Node** head_ref, int new_data)
{
   /* 1. allocate node */
   struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));

   struct Node* last = *head_ref; /* used in step 5*/

   /* 2. put in the data  */
   new_node->data = new_data;

   /* 3. This new node is going to be the last node, so
         make next of it as NULL*/
   new_node->next = NULL;

   /* 4. If the Linked List is empty, then make the new
         node as head */
   if (*head_ref == NULL) {
      new_node->prev = NULL;
      *head_ref = new_node;
      return;
   }

   /* 5. Else traverse till the last node */
   while (last->next != NULL)
      last = last->next;

   /* 6. Change the next of last node */
   last->next = new_node;

   /* 7. Make last node as previous of new node */
   new_node->prev = last;
```

return;

Add a node before a given node:

- Check if the next_node is NULL or not. If it's NULL, return from the function because any new node can not be added before a NULL
- Allocate memory for the new node, let it be called new_node
- Set new_node->data = new_data
- Set the previous pointer of this new_node as the previous node of the next_node, new_node->prev = next_node->prev
- Set the previous pointer of the next_node as the new_node, next_node->prev = new_node
- Set the next pointer of this new_node as the next_node, new_node->next = next_node;
- If the previous node of the new_node is not NULL, then set the next pointer of this previous node as new_node, new_node->prev->next = new_node
- Else, if the prev of new_node is NULL, it will be the new head node. So, make (*head_ref) = new_node.

Algorithm for deleting an element from the node(5M)

Let the node to be deleted is del.

- If node to be deleted is head node, then change the head pointer to next current head
- Set next of previous to del, if previous to del exists.
- Set prev of next to del, if next to del exists.

2. **Explain with an algorithm to perform the polynomial manipulation using linked list representation(13M) (NOV/DEC 2018)** BTL2

Answer Pg no:211-215 Reema Theraja

Definition(2M)

A polynomial p(x) is the expression in variable x which is in the form ($ax^n + bx^{n-1} + \ldots. + jx + k$), where a, b, c …., k fall in the category of real numbers and 'n' is non negative integer, which is called the degree of polynomial.

A polynomial can be thought of as an ordered list of non zero terms. Each non zero term is a two-tuple which holds two pieces of information:

- The exponent part
- The coefficient part

Algorithm AddTwoPolynomials(11M)

```
struct DoublyLinkedList{
    Element *element;
   DoublyLinkedList *left;
   DoublyLinkedList *right;
}
while DLL1 != NULL and DLL2 != NULL do
      DoubleyLinkedList *dll =  new DoublyLInkedList // C++ syntax
      dll ->right = NULL
      dll->element = new Element
      dll->element->coefficient =  DLL1->element->coefficient + DLL2->element-
>coefficient
      dll->element->exponent =  DLL1->element->exponent
      addAtTail( DLL3, dll ) // This will add DoublyLinkedList(dll) at the tail of
DLL3 and adjust point as well
   DLL1 = DLL1->right
   DLL2 = DLL2 ->right
End return DLL3
```

UNIT II LINEAR DATA STRUCTURES – STACKS, QUEUES	
Stack ADT – Operations - Applications - Evaluating Arithmetic Expressions- Conversion of Infix to postfix expression - Queue ADT – Operations - Circular Queue – Priority Queue - dequeue – applications of queues.	
PART A	
1	**Define Stack.** BTL1 A Stack is an ordered list in which all insertions (Push operation) and deletion (Pop operation) are made at one end, called the top. The topmost element is pointed by top. The top is initialized to -1 when the stack is created that is when the stack is empty. In a stack S = (a1,…an), a1 is the bottom most element and element ai is on top of element ai-1. Stack is also referred as Last In First Out (LIFO) list.
2	**What are the various Operations performed on the Stack?** BTL1 The various operations that are performed on the stack are • CREATE(S) – Creates S as an empty stack. • PUSH(S,X) – Adds the element X to the top of the stack. • POP(S) – Deletes the top most elements from the stack. • TOP(S) – returns the value of top element from the stack. • ISEMTPTY(S) – returns true if Stack is empty else false. • ISFULL(S) - returns true if Stack is full else false.
3	**How do you test for an empty stack?** BTL1 The condition for testing an empty stack is top =-1, where top is the pointer pointing to the topmost element of the stack, in the array implementation of stack.In linked list implementation of stack the condition for an empty stack is the header node link field is NULL.
4	**Name two applications of stack. (NOV/DEC 2018)** BTL2 Nested and Recursive functions can be implemented using stack. Conversion of Infix to Postfix expression can be implemented using stack. Evaluation of Postfix expression can be implemented using stack.
5	**Define a suffix expression.** BTL2 The notation used to write the operator at the end of the operands is called suffix notation. Suffix notation format : operand operand operator Example: ab+, where a & b are operands and '+' is addition operator.
6	**What do you meant by fully parenthesized expression? Give eg.** BTL1 A pair of parentheses has the same parenthetical level as that of the operator to which it corresponds. Such an expression is called fully parenthesized expression. Ex: (a+((b*c) + (d * e))
7	**Write the postfix form for the expression -A+B-C+D?** BTL1 A-B+C-D+
8	**What are the postfix and prefix forms of the expression?(April/May 2019)** BTL1 A+B*(C-D)/(P-R) Postfix form: ABCD-*PR-/+

	Prefix form: +A/*B-CD-PR
9	**Mention the usage of stack in recursive algorithm implementation.** BTL2 In recursive algorithms, stack data structures is used to store the return address when a recursive call is encountered and also to store the values of all the parameters essential to the current state of the function.
10	**Define Queues.**BTL1 A Queue is an ordered list in which all insertions take place at one end called the rear, while all deletions take place at the other end called the front. Rear is initialized to -1 and front is initialized to 0. Queue is also referred as First In First Out (FIFO) list.
11	**What are the various operations performed on the Queue? (April/May 2018)** BTL1 • The various operations performed on the queue are • CREATE(Q) – Creates Q as an empty Queue. • Enqueue(Q,X) – Adds the element X to the Queue. • Dequeue(Q) – Deletes a element from the Queue. • ISEMTPTY(Q) – returns true if Queue is empty else false. • ISFULL(Q) - returns true if Queue is full else false.
12	**What are the various types of queue? (May 2008)** BTL1 The following are the types of queue: • Linear Queue • Double ended queue • Circular queue • Priority queue
13	**How do you test for an empty Queue?** BTL2 The condition for testing an empty queue is rear=front-1. In linked list implementation of queue the condition for an empty queue is the header node link field is NULL.

14	**Write down the function to insert an element into a queue, in which the queue is implemented as an array. (May 10)** BTL1
	Q – Queue
	X – element to added to the queue Q
	IsFull(Q) – Checks and true if Queue Q is full
	Q->Size - Number of elements in the queue Q
	Q->Rear – Points to last element of the queue Q
	Q->Array – array used to store queue elements
	void enqueue (int X, Queue Q) {
	if(IsFull(Q))
	Error (“Full queue”);
	else {
	Q->Size++;
	Q->Rear = Q->Rear+1;
	Q->Array[Q->Rear]=X;
	}}

15	**Define Deque.** BTL1 Deque stands for Double ended queue. It is a linear list in which insertions and deletion are made from either end of the queue structure
16	**Define Circular Queue.(Nov/Dec 2017)**BTL1 Another representation of a queue, which prevents an excessive use of memory by arranging elements/ nodes $Q_1,Q_2,\ldots Q_n$ in a circular fashion. That is, it is the queue, which wraps around upon reaching the end of the queue
17	**Define Priority queue. (Nov/Dec 2018) (May 2006)** BTL2 Priority queue is a collection of elements, each containing a key referred as the priority for that element can be inserted in any order (i.e., of alternating priority), but are arranged in order of their priority value in the queue. The elements are deleted from the queue in the order of their priority (i.e., the elements with the highest priority is deleted first). The elements with the same priority are given equal importance and processed accordingly.
18	**Write any four applications of Queue. (Nov 2008)** BTL2 The following are the areas in which queues are applicable • Batch processing in an operating system • Multiprogramming platform systems • Queuing theory • Printer server routines • Scheduling algorithms like disk scheduling , CPU scheduling
19	**State the difference between queues and linked lists.** BTL2 The difference between queues and linked lists is that insertions and deletions may occur anywhere in the linked list, but in queues insertions can be made only in the rear end and deletions can be made only in the front end.
20	**State different ways of representing expressions.** BTL2 The different ways of representing expressions are • Infix Notation • Prefix Notation • Postfix Notation
	PART B
1	**Explain the algorithm for Push and Pop operations on Stack using Linked list. (13M)(April/May 2019)** BTL2 **Answer Pg no:224-225 Reema Theraja** **Implement a stack using singly linked list:** A stack can be easily implemented through the linked list. In stack Implementation, a stack contains a top pointer. which is "head" of the stack where pushing and popping items happens at the head of the list. first node have null in link field and second node link have first node address in link field and so on and last node address in "top" pointer. **Stack Operations:** 1. **Push() :** Insert the element into linked list nothing but which is the top node of Stack.

2. **Pop() :** Return top element from the Stack and move the top pointer to the second node of linked list or Stack.

```
#include <stdio.h>
#include <stdlib.h>

// Declare linked list node

struct Node {
   int data;
   struct Node* link;
};
struct Node* top;

// Utility function to add an element data in the stack
 // insert at the beginning
void push(int data)
{
   // create new node temp and allocate memory
   struct Node* temp;
   temp = (struct Node*)malloc(sizeof(struct Node));

   // check if stack (heap) is full. Then inserting an element would
   // lead to stack overflow
   if (!temp) {
      printf("\nHeap Overflow");
      exit(1);
   }

   // initialize data into temp data field
   temp->data = data;

   // put top pointer reference into temp link
   temp->link = top;

   // make temp as top of Stack
   top = temp;
}

// Utility function to check if the stack is empty or not
int isEmpty()
{
   return top == NULL;
}

// Utility function to return top element in a stack
int peek()
{
   // check for empty stack
```

```
    if (!isEmpty(top))
        return top->data;
    else
        exit(EXIT_FAILURE);
}

// Utility function to pop top element from the stack

void pop()
{
    struct Node* temp;

    // check for stack underflow
    if (top == NULL) {
        printf("\nStack Underflow");
        exit(1);
    }
    else {
        // top assign into temp
        temp = top;

        // assign second node to top
        top = top->link;

        // destroy connection between first and second
        temp->link = NULL;

        // release memory of top node
        free(temp);
    }
}

void display() // remove at the beginning
{
    struct Node* temp;

    // check for stack underflow
    if (top == NULL) {
        printf("\nStack Underflow");
        exit(1);
    }
    else {
        temp = top;
        while (temp != NULL) {

            // print node data
            printf("%d->", temp->data);
```

```
            // assign temp link to temp
            temp = temp->link;
        }
    }
}

// main function

int main(void)
{
    // push the elements of stack
    push(11);
    push(22);
    push(33);
    push(44);

    // display stack elements
    display();

    // print top elementof stack
    printf("\nTop element is %d\n", peek());

    // delete top elements of stack
    pop();
    pop();

    // display stack element
display();

    // print top elementof stack
    printf("\nTop element is %d\n", peek());
    return 0;
}
```

2 **Explain linear linked implementation of Stack and Queue(13M)** BTL2

Answer Pg no:224-230 Reema Theraja

Explanation(6M)

In a Queue data structure, we maintain two pointers, front and rear. The front points the first item of queue and rear points to last item.

enQueue() This operation adds a new node after rear and moves rear to the next node.

deQueue() This operation removes the front node and moves front to the next node.

Algorithm(7M)

```
void enQueue(Queue *q, int k)
{
    // Create a new LL node
    QNode *temp = newNode(k);

    // If queue is empty, then
    // new node is front and rear both
```

```
    if (q->rear == NULL)
    {
    q->front = q->rear = temp;
    return;
    }

    // Add the new node at
    // the end of queue and change rear
    q->rear->next = temp;
    q->rear = temp;
}
// Function to remove
// a key from given queue q
QNode *deQueue(Queue *q)
{
    // If queue is empty, return NULL.
    if (q->front == NULL)
    return NULL;

    // Store previous front and
    // move front one node ahead
    QNode *temp = q->front;
    q->front = q->front->next;

    // If front becomes NULL, then
    // change rear also as NULL
    if (q->front == NULL)
    q->rear = NULL;
    return temp;
}
```

3 **Explain the algorithm for converting infix expression to postfix expression in detail.(13M) (Nov/Dec 2018)(April/May 2019)** BTL2

Answer Pg no:232-237 Reema Theraja

Explanation(5M)

Infix expression: The expression of the form a op b. When an operator is in-between every pair of operands.

Postfix expression: The expression of the form a b op. When an operator is followed for every pair of operands.

Algorithm(8M)

Step1: Scan the infix expression from left to right.

Step 2: If the scanned character is an operand, output it.

Step 3: Else,

Step 3.1: If the precedence of the scanned operator is greater than the precedence of the operator in the stack(or the stack is empty or the stack contains a '('), push it.

Step 3.2: Else, Pop all the operators from the stack which are greater than or equal to in precedence than that of the scanned operator. After doing that Push the scanned

	operator to the stack. (If you encounter parenthesis while popping then stop there and push the scanned operator in the stack.) **Step 4:**If the scanned character is an '(', push it to the stack. **Step 5:** If the scanned character is an ')', pop the stack and and output it until a '(' is encountered, and discard both the parenthesis. **Step 6:** Repeat steps 2-6 until infix expression is scanned. **Step 7:** Print the output **Step 8:** Pop and output from the stack until it is not empty.

4	**Explain in detail about priority queue ADT. (13M)** BTL2
	Answer Pg no:257-259 Reema Theraja
	Explanation(5M)
	Priority Queue is an extension of queue with following properties.
	• Every item has a priority associated with it.
	• An element with high priority is dequeued before an element with low priority.
	• If two elements have the same priority, they are served according to their order in the queue.
	Operations(8M)
	A typical priority queue supports following operations.
	insert(item, priority): Inserts an item with given priority.
	getHighestPriority(): Returns the highest priority item.
	deleteHighestPriority(): Removes the highest priority item.
	How to implement priority queue?
	Using Array: A simple implementation is to use array of following structure.
	struct item {
	int item;
	int priority;
	}
	insert() operation can be implemented by adding an item at end of array in O(1) time.
	getHighestPriority() operation can be implemented by linearly searching the highest priority item in array. This operation takes O(n) time.
	deleteHighestPriority() operation can be implemented by first linearly searching an item, then removing the item by moving all subsequent items one position back.

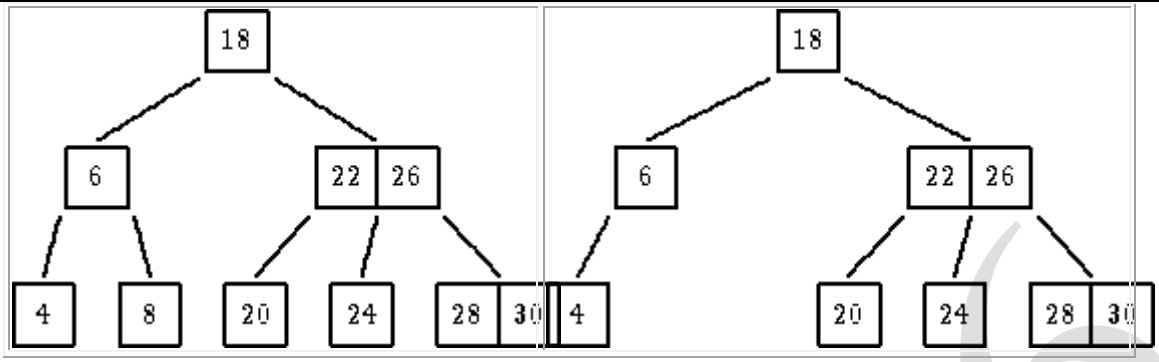
5.	**What is a DeQueue? Explain its operation. (13M)** BTL2 **Answer pg no:264-268 Reema Theraja** **Definition(2M)** Deque or Double Ended Queue is a generalized version of Queue data structure that allows insert and delete at both ends.

	Operations on Deque(11M) Mainly the following four basic operations are performed on queue: **insetFront()**: Adds an item at the front of Deque. **insertRear()**: Adds an item at the rear of Deque. **deleteFront()**: Deletes an item from front of Deque. **deleteRear()**: Deletes an item from rear of Deque. In addition to above operations, following operations are also supported **getFront()**: Gets the front item from queue. **getRear()**: Gets the last item from queue. **isEmpty()**: Checks whether Deque is empty or not. **isFull()**: Checks whether Deque is full or not.
	PART C
1	**Explain the array implementation of queue ADT in detail.(13M)** BTL2 **Answer pg no:252-256 Reema Theraja** To implement a queue using array, create an array arr of size *n* and take two variables front and rear both of which will be initialized to 0 which means the queue is currently empty. Element rear is the index upto which the elements are stored in the array and front is the index of the first element of the array. Now, some of the implementation of queue operations are as follows: • **Enqueue:** Addition of an element to the queue. Adding an element will be performed after checking whether the queue is full or not. If rear < n which indicates that the array is not full then store the element at arr[rear] and increment rear by *1* but if rear == n then it is said to be an Overflow condition as the array is full. • **Dequeue:** Removal of an element from the queue. An element can only be deleted when there is at least an element to delete i.e. rear > 0. Now, element at arr[front] can be deleted but all the remaining elements have to shifted to the left by one position in order for the dequeue operation to delete the second element from the left on another dequeue operation. • **Front:** Get the front element from the queue i.e. arr[front] if queue is not empty. • **Display:** Print all element of the queue. If the queue is non-empty, traverse and print all the elements from index front *to* rear.
2	**Explain the addition and deletion operations performed on a circular queue in detail.(13M)(Nov/Dec 2018) (April/May 2019)** BTL2 **Answer pg no:260-265 Reema Theraja** **Defintion(2M)** Circular Queue is a linear data structure in which the operations are performed based on FIFO (First In First Out) principle and the last position is connected back to the first position to make a circle. It is also called **'**Ring Buffer'**.** **Operations on Circular Queue(11M)** • **Front:** Get the front item from queue. • **Rear:** Get the last item from queue.

	• **enQueue(value)** This function is used to insert an element into the circular queue. In a circular queue, the new element is always inserted at Rear position. **Steps:** 1. Check whether queue is Full – Check ((rear == SIZE-1 && front == 0) || (rear == front-1)). 2. If it is full then display Queue is full. If queue is not full then, check if (rear == SIZE – 1 && front != 0) if it is true then set rear=0 and insert element. • **deQueue()** This function is used to delete an element from the circular queue. In a circular queue, the element is always deleted from front position. **Steps:** 1. Check whether queue is Empty means check (front==-1). 2. If it is empty then display Queue is empty. If queue is not empty then step 3 3. Check if (front==rear) if it is true then set front=rear= -1 else check if (front==size-1), if it is true then set front=0 and return the element.

UNIT III NON LINEAR DATA STRUCTURES – TREES	
Tree ADT – tree traversals - Binary Tree ADT – expression trees – applications of trees – binary search tree ADT –Threaded Binary Trees- AVL Trees – B-Tree - B+ Tree - Heap – Applications of heap.	
PART A	
1	**Define tree.** BTL1 Trees are non-liner data structure, which is used to store data items in a shorted sequence. It represents any hierarchical relationship between any data Item. It is a collection of nodes, which has a distinguish node called the root and zero or more non-empty sub trees T1, T2,….Tk. each of which are connected by a directed edge from the root.
2	**Define Height of tree(May/June 2014).** BTL1 The height of n is the length of the longest path from root to a leaf. Thus all leaves have height zero. The height of a tree is equal to a height of a root.
3	**What are the drawbacks of dynamic programming?** BTL1 • Time and space requirements are high, since storage is needed for all level. • Optimality should be checked at all levels.
4	**Define Depth of tree. (April/May 2018)** BTL1 For any node n, the depth of n is the length of the unique path from the root to node n. Thus for a root the depth is always zero.
5	**What is the length of the path in a tree?** BTL1 The length of the path is the number of edges on the path. In a tree there is exactly one path form the root to each node.
6	**Define sibling (May/June 2012).** BTL2 Nodes with the same parent are called siblings.
7	**Define binary tree** BTL1 A Binary tree is a finite set of data items which is either empty or consists of a single item called root and two disjoin binary trees called left sub tree max degree of any node is two.
8	**What are the two methods of binary tree implementation?** BTL1 Binary tree is used in data processing. a. File index schemes b. Hierarchical database management system
9	**List out few of the Application of tree data-structure?(April/May 2018)** BTL2 • The manipulation of Arithmetic expression • Used for Searching Operation • Used to implement the file system of several popular operating systems • Symbol Table construction • Syntax analysis
10	**Define expression tree.** BTL1 Expression tree is also a binary tree in which the leafs terminal nodes or operands and non-terminal intermediate nodes are operators used for traversal.

12	**Define tree traversal and mention the type of traversals** BTL1 Visiting of each and every node in the tree exactly is called as tree traversal. Three types of tree traversal • Inorder traversal • Preoder traversal • Postorder traversal.
13	**Define in -order traversal** BTL1 In-order traversal entails the following steps; a. Traverse the left subtree b. Visit the root node c. Traverse the right subtree
14	**Define threaded binary tree. (April/May 2018)** BTL2 A binary tree is threaded by making all right child pointers that would normally be null point to the inorder successor of the node, and all left child pointers that would normally be null point to the inorder predecessor of the node.
15	**What are the types of threaded binary tree?** BTL1 • Right-in threaded binary tree • Left-in threaded binary tree • Fully-in threaded binary tree
16	**Define Binary Search Tree. (April/May 2017)** BTL1 Binary search tree is a binary tree in which for every node X in the tree, the values of all the keys in its left subtree are smaller than the key value in X and the values of all the keys in its right subtree are larger than the key value in X.
17	**What is AVL Tree? (Nov/Dec 2016)**BTL1 AVL stands for Adelson-Velskii and Landis.An AVL tree is a binary search tree which has the following properties: 1. The sub-trees of every node differ in height by at most one. 2. Every sub-tree is an AVL tree. Search time is O(logn). Addition and deletion operations also take O(logn) time.
17	**What is 'B' Tree?. (April/May 2015)**BTL1 A B-tree is a tree data structure that keeps data sorted and allows searches, insertions, and deletions in logarithmic amortized time. Unlike self-balancing binary search trees, it is optimized for systems that read and write large blocks of data. It is most commonly used in database and file systems. B-tree of order 3 | not a B-tree



Important properties of a B-tree:

- B-tree nodes have many more than two children.
- A B-tree node may contain more than just a single element.

18. **What is binomial heaps?** BTL2

A binomial heap is a collection of binomial trees that satisfies the following binomial-heap properties:

1. No two binomial trees in the collection have the same size.
2. Each node in each tree has a key.
3. Each binomial tree in the collection is heap-ordered in the sense that each non-root has a key strictly less than the key of its parent

.

The number of trees in a binomial heap is O(log n).

19. **Define complete binary tree.** BTL2

If all its levels, possible except the last, have maximum number of nodes and if all the nodes in the last level appear as far left as possible.

PART B

1 **Explain the AVL tree insertion and deletion with suitable example. (13M)** BTL2

Answer pg no:318-320 Reema Theraja

Definition(2M)

AVL tree is a self-balancing Binary Search Tree (BST) where the difference between heights of left and right subtrees cannot be more than one for all nodes.

Steps to follow for insertion(6M)

Let the newly inserted node be w

- Perform standard BST insert for w.
- Starting from w, travel up and find the first unbalanced node. Let z be the first unbalanced node, y be the child of z that comes on the path from w to z and x be the grandchild of z that comes on the path from w to z.
- Re-balance the tree by performing appropriate rotations on the subtree rooted with z. There can be 4 possible cases that needs to be handled as x, y and z can be arranged in 4 ways. Following are the possible 4 arrangements.
  - y is left child of z and x is left child of y (Left Left Case
  - y is left child of z and x is right child of y (Left Right Case)

- y is right child of z and x is right child of y (Right Right Case)
- y is right child of z and x is left child of y (Right Left Case)

Steps to follow for deletion(5M)

To make sure that the given tree remains AVL after every deletion, we must augment the standard BST delete operation to perform some re-balancing. Following are two basic operations that can be performed to re-balance a BST without violating the BST property (keys(left) < key(root) < keys(right)).

- Left Rotation
- Right Rotation

2 **Explain single and double rotation on AVL tree in detail. (13M)** BTL2

Answer pg no:320-324 Reema Theraja

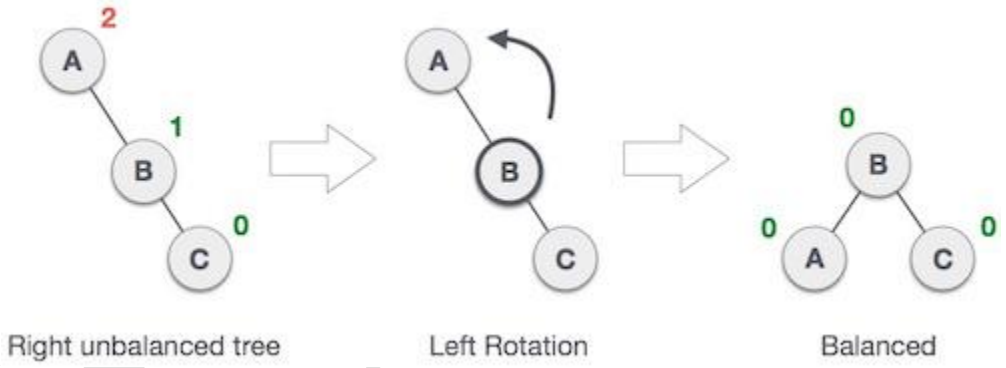
AVL Rotations

To balance itself, an AVL tree may perform the following four kinds of rotations −

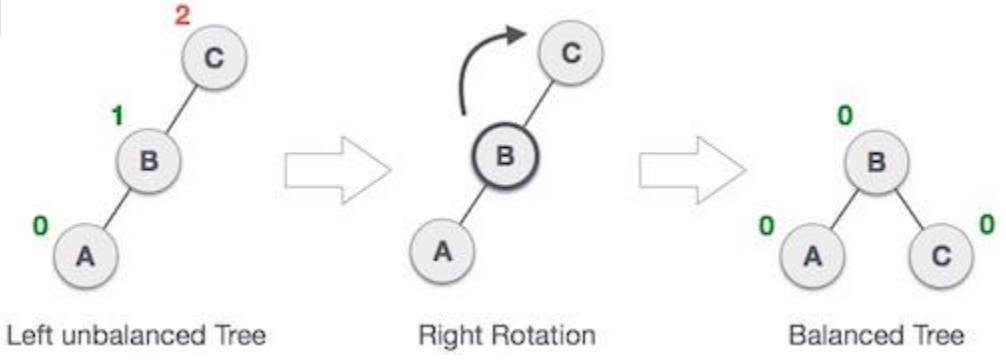
- Left rotation
- Right rotation
- Left-Right rotation
- Right-Left rotation

Left Rotation(2M)

The first two rotations are single rotations and the next two rotations are double rotations.



Right Rotation(3M)



	Left-Right Rotation(4M) A left-right rotation is a combination of left rotation followed by right rotation. **Right-Left Rotation(4M)** The second type of double rotation is Right-Left Rotation. It is a combination of right rotation followed by left rotation.

3	**Explain about B-Tree with suitable example(13M) (Nov/Dec 2018)** BTL2
	Answer pg no:325-330 Reema Theraja
	Definition(2M)
	B Tree is a specialized m-way tree that can be widely used for disk access. A B-Tree of order m can have at most m-1 keys and m children.
	
	Operations(11M)
	
	Insertion
	
	Insertions are done at the leaf node level. The following algorithm needs to be followed in order to insert an item into B Tree.
	
	• Traverse the B Tree in order to find the appropriate leaf node at which the node can be inserted.
	• If the leaf node contain less than m-1 keys then insert the element in the increasing order.
	• Else, if the leaf node contains m-1 keys, then follow the following steps.
	  • Insert the new element in the increasing order of elements.
	  • Split the node into the two nodes at the median.
	  • Push the median element upto its parent node.
	  • If the parent node also contain m-1 number of keys, then split it too by following the same steps.
	
	Deletion
	
	Deletion is also performed at the leaf nodes. The node which is to be deleted can either be a leaf node or an internal node. Following algorithm needs to be followed in order to delete a node from a B tree.
	• Locate the leaf node.
	• If there are more than m/2 keys in the leaf node then delete the desired key from the node.
	• If the leaf node doesn't contain m/2 keys then complete the keys by taking the element from eight or left sibling.

	• If the left sibling contains more than m/2 elements then push its largest element up to its parent and move the intervening element down to the node where the key is deleted. • If the right sibling contains more than m/2 elements then push its smallest element up to the parent and move intervening element down to the node where the key is deleted. • If neither of the sibling contain more than m/2 elements then create a new leaf node by joining two leaf nodes and the intervening element of the parent node. • If parent is left with less than m/2 nodes then, apply the above process on the parent too.
4	**Explain the following in detail:** **1.Binomial heaps(6M)** **2. Fibonacci heaps(7M)** BTL1 **1.Binomial Heap(2M)** A Binomial Tree of order 0 has 1 node. A Binomial Tree of order k can be constructed by taking two binomial trees of order k-1 and making one as leftmost child or other. A Binomial Tree of order k has following properties. • It has exactly 2^k nodes. • It has depth as k. • There are exactly ${}^kC_i$ nodes at depth i for i = 0, 1, . . . , k. • The root has degree k and children of root are themselves Binomial Trees with order k-1, k-2,.. 0 from left to right. **Operations of Binomial Heap(4M)** The main operation in Binomial Heap is union(), all other operations mainly use this operation. The union() operation is to combine two Binomial Heaps into one. Let us first discuss other operations, we will discuss union later. • insert(H, k): Inserts a key 'k' to Binomial Heap 'H'. This operation first creates a Binomial Heap with single key 'k', then calls union on H and the new Binomial heap. • getMin(H): A simple way to getMin() is to traverse the list of root of Binomial Trees and return the minimum key. This implementation requires O(Logn) time. It can be optimized to O(1) by maintaining a pointer to minimum key root. • extractMin(H): This operation also uses union(). We first call getMin() to find the minimum key Binomial Tree, then we remove the node and create a new Binomial Heap by connecting all subtrees of the removed minimum node. Finally, we call union() on H and the newly created Binomial Heap. This operation requires O(Logn) time. • delete(H): Like Binary Heap, delete operation first reduces the key to minus infinite, then calls extractMin(). • decreaseKey(H): decreaseKey() is also similar to Binary Heap. We compare the decreases key with it parent and if parent's key is more, we swap keys and recur for the parent. We stop when we either reach a node whose parent has a smaller key or we hit the root node. Time complexity of decreaseKey() is O(Logn).

2.Fibonacci heaps(2M)

Fibonacci Heap is a collection of trees with min-heap or max-heap property. In Fibonacci Heap, trees can can have any shape even all trees can be single nodes

Insertion(3M)

- Create a new node 'x'.
- Check whether heap H is empty or not.
- If H is empty then:
  - Make x as the only node in the root list.
  - Set H(min) pointer to x.
- Else:
  - Insert x into root list and update H(min).

Union(2M)

Union of two Fibonacci heaps H1 and H2 can be accomplished as follows:

- Unoin root lists of Fibonacci heaps H1 and H2 and make a single Fibonacci heap H.
- If H1(min) < H2(min) then:
  - H(min) = H1(min).
- Else:
  - H(min) = H2(min).

PART C

1 **Explain the tree traversal techniques with an example. (13M)** BTL2

Answer pg no:287-289 Reema Theraja

Traversal is a process to visit all the nodes of a tree and may print their values too.

There are three ways which we use to traverse a tree −

- In-order Traversal
- Pre-order Traversal
- Post-order Traversal

In-order Traversal(4M)

Algorithm

Until all nodes are traversed −
Step 1 − Recursively traverse left subtree.
Step 2 − Visit root node.
Step 3 − Recursively traverse right subtree.

Pre-order Traversal(4M)

Algorithm

Until all nodes are traversed −
Step 1 − Visit root node.
Step 2 − Recursively traverse left subtree.
Step 3 − Recursively traverse right subtree.

Post-order Traversal(5M)

Algorithm

Until all nodes are traversed −
Step 1 − Recursively traverse left subtree.
Step 2 − Recursively traverse right subtree.
Step 3 − Visit root node.

2 **Explain insertion and search of an element into a binary search tree(13M) Answer Nov/Dec 2018 Reema Theraja** BTL2
pg no:298-303

Definition(2M)

A Binary Search Tree (BST) is a tree in which all the nodes follow the below-mentioned properties −

- The left sub-tree of a node has a key less than or equal to its parent node's key.
- The right sub-tree of a node has a key greater than to its parent node's key.

Insert Operation(6M)

Algorithm

```
void insert(int data) {
  struct node *tempNode = (struct node*) malloc(sizeof(struct node));
  struct node *current;
  struct node *parent;

  tempNode->data = data;
  tempNode->leftChild = NULL;
  tempNode->rightChild = NULL;

  //if tree is empty
  if(root == NULL) {
    root = tempNode;
  } else {
    current = root;
```

```
    parent = NULL;

    while(1) {
      parent = current;

      //go to left of the tree
      if(data < parent->data) {
        current = current->leftChild;
        //insert to the left

        if(current == NULL) {
          parent->leftChild = tempNode;
          return;
        }
      }  //go to right of the tree
      else {
        current = current->rightChild;

        //insert to the right
        if(current == NULL) {
          parent->rightChild = tempNode;
          return;
        }
      }
    }
  }
```

```
}
```

Search Operation(5M)

Algorithm

```
struct node* search(int data){
   struct node *current = root;
   printf("Visiting elements: ");

   while(current->data != data){

      if(current != NULL) {
         printf("%d ",current->data);

         //go to left tree
         if(current->data > data){
            current = current->leftChild;
         }  //else go to right tree
         else {
            current = current->rightChild;
         }

         //not found
         if(current == NULL){
            return NULL;
         }
      }
   }
```

4

What are threaded binary tree? Explain the algorithm for inserting a node in a threaded binary tree.(13M) BTL2

Answer pg no:311-315 Reema Theraja

Threaded Binary Tree(2M)

The idea of threaded binary trees is to make inorder traversal faster and do it without stack and without recursion. A binary tree is made threaded by making all right child pointers that would normally be NULL point to the inorder successor of the node here are two types of threaded binary trees.

Single Threaded: Where a NULL right pointers is made to point to the inorder successor (if successor exists)

Double Threaded*:* Where both left and right NULL pointers are made to point to inorder predecessor and inorder successor respectively. The predecessor threads are useful for reverse inorder traversal and postorder traversal.

The threads are also useful for fast accessing ancestors of a node.

Algorithm to do inorder traversal in a threaded binary tree (11M)

```
void inOrder(struct Node *root)
{
   struct Node *cur = leftmost(root);
   while (cur != NULL)
   {
      printf("%d ", cur->data);

      // If this node is a thread node, then go to
      // inorder successor
      if (cur->rightThread)
         cur = cur->right;
      else // Else go to the leftmost child in right subtree
         cur = leftmost(cur->right);
   }
}
```

UNIT IV NON LINEAR DATA STRUCTURES - GRAPHS	
Definition – Representation of Graph – Types of graph - Breadth-first traversal - Depth-first traversal – Topological Sort – Bi-connectivity – Cut vertex – Euler circuits – Applications of graphs.	
PART A	
1	**Write the definition of weighted graph**BTL1 A graph in which weights are assigned to every edge is called a weighted graph.
2	**Define Graph** BTL1 A graph G consist of a nonempty set V which is a set of nodes of the graph, a set E which is the set of edges of the graph, and a mapping from the set of edges E to set of pairs of elements of V. It can also be represented as G=(V, E).
3	**Define adjacency matrix (April/May 2016)** BTL1 The adjacency matrix is an n x n matrix A whose elements aij are given by aij = 1 if (vi, vj) Exists =0 otherwise
4	**Define adjacent nodes** BTL1 Any two nodes, which are connected by an edge in a graph, are called adjacent nodes. For example, if an edge x□E is associated with a pair of nodes (u,v) where u, v □V, then we say that the edge x connects the nodes u and v.
5	**What is a directed graph?**BTL1 A graph in which every edge is directed is called a directed graph.
6	**What is an undirected graph?**BTL2 A graph in which every edge is undirected is called an undirected graph.
7	**What is a loop?**BTL2 An edge of a graph, which connects to itself, is called a loop or sling.
8	**What is a simple graph?**BTL2 A graph in which weights are assigned to every edge is called a weighted graph.
9	**Define indegree and out degree of a graph (April/May 2018)** BTL2 In a directed graph, for any node v, the number of edges, which have v as their initial node, is called the out degree of the node v. Outdegree: Number of edges having the node v as root node is the outdegree of the node v.
10	**Define path in a graph.** BTL1 The path in a graph is the route taken to reach terminal node from a starting node.
11	**What is a simple path?** BTL1 A path in a diagram in which the edges are distinct is called a simple path. It is also called as edge simple.
12	**What is a cycle or a circuit?** BTL1 A path which originates and ends in the same node is called a cycle or circuit.
13	**What is an acyclic graph?** BTL1 A simple diagram, which does not have any cycles, is called an acyclic graph.
14	**What is meant by strongly connected and weakly connected in a graph?** BTL1 An undirected graph is connected, if there is a path from every vertex to every other

	vertex. A directed graph with this property is called strongly connected. When a directed graph is not strongly connected but the underlying graph is connected, then the graph is said to be weakly connected.
15	**Name the different ways of representing a graph. Give examples (Nov/Dec 2018)** BTL2**(Nov 10)** a. Adjacency matrix b. Adjacency list
17	**What is an undirected acyclic graph?** BTL1 When every edge in an acyclic graph is undirected, it is called an undirected acyclic graph. It is also called as undirected forest.
18	**What is meant by depth?** BTL1 The depth of a list is the maximum level attributed to any element with in the list or with in any sub list in the list.
19	**What is the use of BFS?** BTL1 BFS can be used to find the shortest distance between some starting node and the remaining nodes of the graph. The shortest distance is the minimum number of edges traversed in order to travel from the start node the specific node being examined.
20.	**What is topological sort? (April/May 2017)** BTL1 It is an ordering of the vertices in a directed acyclic graph, such that: If there is a path from u to v, then v appears after u in the ordering.
21.	**Write the steps involved in BFS algorithm.** BTL1 1.Initialize the first node's dist number and place in queue 2. Repeat until all nodes have been examined 3. Remove current node to be examined from queue 4. Find all unlabeled nodes adjacent to current node 5. If this is an unvisited node label it and add it to the queue 6. Finished.
22	**Define biconnected graph.** BTL1 A graph is called biconnected if there is no single node whose removal causes the graph to break into two or more pieces. A node whose removal causes the graph to become disconnected is called a cut vertex.
23.	**What are the two traversal strategies used in traversing a graph?(April/May 2016)** BTL1 a. Breadth first search b. Depth first search
24	**What is a Euler path? (Nov/Dec 2018)** BTL1 An Euler path is a path that uses every edge of a graph exactly once. An Euler circuit is a circuit that uses every edge of a graph exactly once. An Euler path starts and ends at different vertices. An Euler circuit starts and ends at the same vertex.
	PART B
1	**Explain the various representation of graph with example in detail.(13 M)** BTL3 **Answer pg no:385-390 Reema Theraja** **Definition(2M):** Graph is a data structure that consists of following two components: **1.** A finite set of vertices also called as nodes. **2.** A finite set of ordered pair of the form (u, v) called as edge. The pair is ordered

	because (u, v) is not same as (v, u) in case of a directed graph(di-graph). The pair of the form (u, v) indicates that there is an edge from vertex u to vertex v. The edges may contain weight/value/cost. **Graph and its representations(11M):** Following two are the most commonly used representations of a graph. • Adjacency Matrix • Adjacency List • There are other representations also like, Incidence Matrix and Incidence List. The choice of the graph representation is situation specific. It totally depends on the type of operations to be performed and ease of use. **AdjacencyMatrix:** Adjacency Matrix is a 2D array of size V x V where V is the number of vertices in a graph. Let the 2D array be adj[][], a slot adj[i][j] = 1 indicates that there is an edge from vertex i to vertex j. Adjacency matrix for undirected graph is always symmetric. Adjacency Matrix is also used to represent weighted graphs. If adj[i][j] = w, then there is an edge from vertex i to vertex j with weight w. **AdjacencyList:** An array of lists is used. Size of the array is equal to the number of vertices. Let the array be array[]. An entry array[i] represents the list of vertices adjacent to the ***i***th vertex. This representation can also be used to represent a weighted graph. The weights of edges can be represented as lists of pairs. Following is adjacency list representation of the above graph.
2	**Explain Breadth First Search algorithm in detail. (13M) (Nov/Dec 2018)** BTL3 **Answer pg no:394-397 Reema Theraja** **Definition(2M):** Breadth First Search (BFS) algorithm traverses a graph in a breadthward motion and uses a queue to remember to get the next vertex to start a search, when a dead end occurs in any iteration. **Rules for BFS(11M):** • **Rule 1** − Visit the adjacent unvisited vertex. Mark it as visited. Display it. Insert it in a queue. • **Rule 2** − If no adjacent vertex is found, remove the first vertex from the queue. • **Rule 3** − Repeat Rule 1 and Rule 2 until the queue is empty.
3.	**Explain Depth First Traversal in detail. (13M) (Nov/Dec 2018)** BTL3 **Answer Pg no:397-400 Reema Theraja** **Definition(2M):** Depth First Search (DFS) algorithm traverses a graph in a depth ward motion and uses a stack to remember to get the next vertex to start a search, when a dead end occurs in any iteration. **Rules for DFS(11M):**

	It employs the following rules: • **Rule 1** − Visit the adjacent unvisited vertex. Mark it as visited. Display it. Push it in a stack. • **Rule 2** − If no adjacent vertex is found, pop up a vertex from the stack. (It will pop up all the vertices from the stack, which do not have adjacent vertices.) • **Rule 3** − Repeat Rule 1 and Rule 2 until the stack is empty.

4	**What is topological sort? Write an algorithm to perform topological sort? (13M) (Nov/Dec 2018) (Nov 09)**
	Answer Pg no:400-405 Reema Theraja
	Definition(2M):
	The topological sorting for a directed acyclic graph is the linear ordering of vertices. For every edge U-V of a directed graph, the vertex u will come before vertex v in the ordering.
	Algorithm for Topological Sorting(11M):
	topoSort(u, visited, stack)
	Input: The start vertex u, An array to keep track of which node is visited or not. A stack to store nodes.
	Output: Sorting the vertices in topological sequence in the stack.
	Begin
	mark u as visited
	for all vertices v which is adjacent with u, do
	if v is not visited, then
	topoSort(c, visited, stack)
	done
	push u into a stack
	End
	performTopologicalSorting(Graph)
	Input: The given directed acyclic graph.
	Output: Sequence of nodes.

Begin

initially mark all nodes as unvisited

for all nodes v of the graph, do

if v is not visited, then

topoSort(i, visited, stack)

done

pop and print all elements from the stack

End.

PART C

1. **Explain with an algorithm to determine the bi connected components in the given graph. (15M)** BTL2

Definition(2M)

- It is connected, i.e. it is possible to reach every vertex from every other vertex, by a simple path.
- Even after removing any vertex the graph remains connected.

Algorithm for Bi connected Graph(13M):

```
time = 0
function isBiconnected(vertex, adj[][], low[], disc[], parent[], visited[], V)
   disc[vertex]=low[vertex]=time+1
   time = time + 1
   visited[vertex]=true
   child = 0
   for i = 0 to V
      if adj[vertex][i] == true
         if visited[i] == false
            child = child + 1
            parent[i] = vertex
            result = isBiconnected(i, adj, low, disc, visited, V, time)
            if result == false
               return false
            low[vertex] = minimum(low[vertex], low[i])
            if parent[vertex] == nil AND child > 1
               return false
            if parent[vertex] != nil AND low[i] >= disc[vertex]
               return false
         else if parent[vertex] != i
            low[vertex] = minimum(disc[i], low[vertex])
```

UNIT V SEARCHING, SORTING AND HASHING TECHNIQUES	
Searching- Linear Search - Binary Search. Sorting - Bubble sort - Selection sort - Insertion sort - Shell sort – Radix sort. Hashing- Hash Functions – Separate Chaining – Open Addressing – Rehashing – Extendible Hashing.	
PART A	
1	**What is meant by Sorting?**BTL1 Sorting is ordering of data in an increasing or decreasing fashion according to some linear relationship among the data items.
2	**List the different sorting algorithms.** BTL2 • Bubble sort • Selection sort • Insertion sort • Shell sort • Quick sort • Radix sort • Heap sort • Merge sort
3	**State the logic of bubble sort algorithm.(Nov/Dec 2017)** BTL2 The bubble sort repeatedly compares adjacent elements of an array. The first and second elements are compared and swapped if out of order. Then the second and third elements are compared and swapped if out of order. This sorting process continues until the last two elements of the array are compared and swapped if out of order.
4	**What number is always sorted to the top of the list by each pass of the Bubble sort algorithm?** BTL1 Each pass through the list places the next largest value in its proper place. In essence, each item "bubbles" up to the location where it belongs.
5	**When does the Bubble Sort Algorithm stop?** BTL1 The bubble sort stops when it examines the entire array and finds that no "swaps" are needed. The bubble sort keeps track of the occurring swaps by the use of a flag.
6	**State the logic of selection sort algorithm.** BTL2 It finds the lowest value from the collection and moves it to the left. This is repeated until the complete collection is sorted.
7	**How does insertion sort algorithm work?(April/May 2017)** BTL2 In every iteration an element is compared with all the elements before it. While comparing if it is found that the element can be inserted at a suitable position, then space is created for it by shifting the other elements one position up and inserts the desired element at the suitable position. This procedure is repeated for all the elements in the list until we get the sorted elements.
8	**What operation does the insertion sort use to move numbers from the unsorted section to the sorted section of the list?** BTL1 The Insertion Sort uses the swap operation since it is ordering numbers within a single list.

9	**How many key comparisons and assignments an insertion sort makes in its worst case?** BTL2 The worst case performance in insertion sort occurs when the elements of the input array are in descending order. In that case, the first pass requires one comparison, the second pass requires two comparisons, third pass three comparisons, kth pass requires (k-1), and finally the last pass requires (n-1) comparisons. Therefore, total numbers of comparisons are: f(n) = 1+2+3+……+(n-k) +…. +(n-2) +(n-1) = n(n-1)/2 = O(n2)
10	**Which sorting algorithm is best if the list is already sorted? Why?** BTL1 Insertion sort as there is no movement of data if the list is already sorted and complexity is of the order O(N).
11	**Which sorting algorithm is easily adaptable to singly linked lists? Why?** BTL1 Insertion sort is easily adaptable to singly linked list. In this method there is an array link of pointers, one for each of the original array elements. Thus the array can be thought of as a linear link list pointed to by an external pointer first initialized to 0. To insert the k^{th} element the linked list is traversed until the proper position for x[k] is found, or until the end of the list is reached. At that point x[k] can be inserted into the list by merely adjusting the pointers without shifting any elements in the array which reduces insertion time.
12	**Why Shell Sort is known diminishing increment sort?** BTL1 The distance between comparisons decreases as the sorting algorithm runs until the last phase in which adjacent elements are compared. In each step, the sortedness of the sequence is increased, until in the last step it is completely sorted.
13	**What is the key idea of radix sort?** BTL1 Sort the keys digit by digit, starting with the least significant digit to the most significant digit.
14	**Define Searching.(April/May 2019)** BTL1 Searching for data is one of the fundamental fields of computing. Often, the difference between a fast program and a slow one is the use of a good algorithm for the data set. Naturally, the use of a hash table or binary search tree will result in more efficient searching, but more often than not an array or linked list will be used. It is necessary to understand good ways of searching data structures not designed to support efficient search.
15	**What is linear search?** BTL1 In Linear Search the list is searched sequentially and the position is returned if the key element to be searched is available in the list, otherwise -1 is returned. The search in Linear Search starts at the beginning of an array and move to the end, testing for a match at each item.
16	**Define hash function?** BTL1 Hash function takes an identifier and computes the address of that identifier in the hash table using some function.
17	**Why do we need a Hash function as a data structure as compared to any other data structure?** BTL2**(may 10)** Hashing is a technique used for performing insertions, deletions, and finds in constant

	average time.
18	**What are the important factors to be considered in designing the hash function? (Nov 10)** BTL1 • To avoid lot of collision the table size should be prime • For string data if keys are very long, the hash function will take long to compute.
19	**What are the problems in hashing?** BTL1 a. Collision b. Overflow
20	**What do you mean by hash table?** BTL1 The hash table data structure is merely an array of some fixed size, containing the keys. A key is a string with an associated value. Each key is mapped into some number in the range 0 to tablesize-1 and placed in the appropriate cell.
21.	**What do you mean by hash function?(April/May 2019)** BTL1 A hash function is a key to address transformation which acts upon a given key to compute the relative position of the key in an array. The choice of hash function should be simple and it must distribute the data evenly. A simple hash function is hash_key=key mod table size.
22.	**What do you mean by separate chaining?** BTL1 Separate chaining is a collision resolution technique to keep the list of all elements that hash to the same value. This is called separate chaining because each hash table element is a separate chain (linked list). Each linked list contains all the elements whose keys hash to the same index.

PART B	

1	**Write an algorithm to implement Bubble sort with suitable example. (13M)** BTL3
	Answer Pg no:434-437 Reema Theraja
	Definition for Bubble sort(2M):
	Bubble sort is a simple sorting algorithm. This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large data sets as its average and worst case complexity are of $O(n^2)$ where **n** is the number of items.
	Algorithm for Bubble sort(11M):
	begin BubbleSort(list)
	for all elements of list
	if list[i] > list[i+1]
	swap(list[i], list[i+1])
	end if
	end for
	return list

	end BubbleSort
	procedure bubbleSort(list : array of items
	loop = list.count;
	for i = 0 to loop-1 do:
	swapped = false
	for j = 0 to loop-1 do:
	/* compare the adjacent elements */
	if list[j] > list[j+1] then
	/* swap them */
	swap(list[j], list[j+1])
	swapped = true
	end if
	end for
	/*if no number was swapped that means
	array is sorted now, break the loop.*/
	if(not swapped) then
	break
	end if
	end for

2	**Explain insertion sort in detail with suitable example. (13M)** BTL2 **Answer pg no:438-440 Reema Theraja** **Definition for insertion sort(2M):** This is an in-place comparison-based sorting algorithm. Here, a sub-list is maintained which is always sorted. For example, the lower part of an array is maintained to be sorted. An element which is to be inserted in this sorted sub-list, has to find its appropriate place and then it has to be inserted there. Hence the name insertion sort. **Algorithm for insertion sort(11M):** **Step 1** − If it is the first element, it is already sorted. return 1; **Step 2** − Pick next element **Step 3** − Compare with all elements in the sorted sub-list **Step 4** − Shift all the elements in the sorted sub-list that is greater than the value to be sorted

	Step 5 − Insert the value **Step 6** − Repeat until list is sorted
3	**Explain selection sort in detail with suitable example. (13M)** BTL2 **Answer Pg no:441-442 Reema Theraja** **Definition for Selection sort(2M):** Selection sort is a simple sorting algorithm. This sorting algorithm is an in-place comparison-based algorithm in which the list is divided into two parts, the sorted part at the left end and the unsorted part at the right end. Initially, the sorted part is empty and the unsorted part is the entire list. **Algorithm for Selection sort(11M):** **Step 1** − Set MIN to location 0 **Step 2** − Search the minimum element in the list **Step 3** − Swap with value at location MIN **Step 4** − Increment MIN to point to next element **Step 5** − Repeat until list is sorted

	Explain radix sort algorithm with suitable example. (13M) BTL1
	Answer pg no:450-452 Reema Theraja
	Definition of radix sort(2M)
	On the first pass, all the numbers are sorted on the least significant digit and combined in an array. Then on the second pass, the entire numbers are sorted again on the second least significant digits and combined in an array and so on
	Algorithm: Radix-Sort (list, n) (11M)
	shift = 1
	for loop = 1 to keysize do
	for entry = 1 to n do
	bucketnumber = (list[entry].key / shift) mod 10
	append (bucket[bucketnumber], list[entry])
	list = combinebuckets()
	shift = shift * 10

	PART C
1	**Explain binary search algorithm in detail with suitable example. (15M) (April/May 2019)** BTL3` **Answer Pg no:421-425 Reema Theraja** **Definition(2M)** Binary search is a fast search algorithm with run-time complexity of O(log n). This search algorithm works on the principle of divide and conquer. For this algorithm to work properly, the data collection should be in the sorted form. **Algorithm for Binary search(13M)** Procedure binary_search A ← sorted array n ← size of array

	x ← value to be searched Set lowerBound = 1 Set upperBound = n while x not found if upperBound < lowerBound EXIT: x does not exists. set midPoint = lowerBound + (upperBound - lowerBound) / 2 if A[midPoint] < x set lowerBound = midPoint + 1 if A[midPoint] > x set upperBound = midPoint - 1 if A[midPoint] = x EXIT: x found at location midPoint end while end procedure
2	**Explain Re-hashing and Extendible hashing.(15M)(April/May 2019)** BTL1 **Answer Pg no:473-481 Reema Theraja** **Definition of Rehashing(2M):** As the name suggests, **rehashing means hashing again**. Rehashing is done because whenever key value pairs are inserted into the map, the load factor increases, which implies that the time complexity also increases as explained above. This might not give the required time complexity of O(1). Hence, rehash must be done, increasing the size of the bucketArray so as to reduce the load factor and the time complexity. **Steps involved in Rehashing(5M):** Rehashing can be done as follows: • For each addition of a new entry to the map, check the load factor. • If it's greater than its pre-defined value (or default value of 0.75 if not given), then Rehash.

- For Rehash, make a new array of double the previous size and make it the new bucketarray.
- Then traverse to each element in the old bucketArray and call the insert() for each so as to insert it into the new larger bucket array.

Definition of Extended Hashing(2M):

The problem with static hashing is that it does not expand or shrink dynamically as the size of the database grows or shrinks. Dynamic hashing provides a mechanism in which data buckets are added and removed dynamically and on-demand. Dynamic hashing is also known as extended hashing.

Hash function, in dynamic hashing, is made to produce a large number of values and only a few are used initially.

Operation(8M)

- **Querying** − Look at the depth value of the hash index and use those bits to compute the bucket address.
- **Update** − Perform a query as above and update the data.
- **Deletion** − Perform a query to locate the desired data and delete the same.
- **Insertion** − Compute the address of the bucket

  If the bucket is already full.

  - Add more buckets.
  - Add additional bits to the hash value.
  - Re-compute the hash function.

  Else

  - Add data to the bucket,

  If all the buckets are full, perform the remedies of static hashing.

CS8392 OBJECT ORIENTED PROGRAMMING L T P C

3 0 0 3

OBJECTIVES:

- To understand Object Oriented Programming concepts and basic characteristics of Java
- To know the principles of packages, inheritance and interfaces
- To define exceptions and use I/O streams
- To develop a java application with threads and generics classes
- To design and build simple Graphical User Interfaces

UNIT I INTRODUCTION TO OOP AND JAVA FUNDAMENTALS 10

Object Oriented Programming – Abstraction – objects and classes – Encapsulation- Inheritance – Polymorphism- OOP in Java – Characteristics of Java – The Java Environment – Java Source File -Structure – Compilation. Fundamental Programming Structures in Java – Defining classes in Java – constructors, methods -access specifiers – static members -Comments, Data Types, Variables, Operators, Control Flow, Arrays , Packages – JavaDoc comments.

UNIT II INHERITANCE AND INTERFACES 9

Inheritance – Super classes- sub classes –Protected members – constructors in sub classes- the Object class – abstract classes and methods- final methods and classes – Interfaces – defining an interface, implementing interface, differences between classes and interfaces and extending interfaces – Object cloning -inner classes, Array Lists – Strings

UNIT III EXCEPTION HANDLING AND I/O 9

Exceptions – exception hierarchy – throwing and catching exceptions – built-in exceptions, creating own exceptions, Stack Trace Elements. Input / Output Basics – Streams – Byte streams and Character streams – Reading and Writing Console – Reading and Writing Files

UNIT IV MULTITHREADING AND GENERIC PROGRAMMING 8

Differences between multi-threading and multitasking, thread life cycle, creating threads, synchronizing threads, Inter-thread communication, daemon threads, thread groups. Generic Programming – Generic classes – generic methods – Bounded Types – Restrictions and Limitations.

UNIT V EVENT DRIVEN PROGRAMMING 9

Graphics programming – Frame – Components – working with 2D shapes – Using color, fonts, and images – Basics of event handling – event handlers – adapter classes – actions – mouse events – AWT event hierarchy – Introduction to Swing – layout management – Swing Components – Text Fields , Text Areas – Buttons- Check Boxes – Radio Buttons – Lists- choices- Scrollbars – Windows –Menus – Dialog Boxes.

TOTAL: 45 PERIODS

OUTCOMES:

Upon completion of the course, students will be able to:

- Develop Java programs using OOP principles
- Develop Java programs with the concepts inheritance and interfaces
- Build Java applications using exceptions and I/O streams
- Develop Java applications with threads and generics classes
- Develop interactive Java programs using swings

TEXT BOOKS:

1. Herbert Schildt, “Java The complete reference”, 8th Edition, McGraw Hill Education, 2011.
2. Cay S. Horstmann, Gary cornell, “Core Java Volume –I Fundamentals”, 9th Edition, Prentice Hall, 2013.

REFERENCES:

1. Paul Deitel, Harvey Deitel, "Java SE 8 for programmers", 3rd Edition, Pearson, 2015.

2. Steven Holzner, "Java 2 Black book", Dreamtech press, 2011.

3. Timothy Budd, "Understanding Object-oriented programming with Java", Updated Edition, Pearson Education, 2000.

Subject Code: CS8392 **Year/Semester: IV /08**
Subject Name: OBJECT ORIENTED PROGRAMMING **Subject Handler: M.SUGANYA**

UNIT 1 - INTRODUCTION TO OOP AND JAVA FUNDAMENTALS	
Object Oriented Programming - Abstraction – objects and classes - Encapsulation- Inheritance - Polymorphism- OOP in Java – Characteristics of Java – The Java Environment - Java Source File -Structure – Compilation. Fundamental Programming Structures in Java – Defining classes in Java – constructors, methods -access specifiers - static members -Comments, Data Types, Variables, Operators, Control Flow, Arrays , Packages - JavaDoc comments.	
PART * A	
Q.NO	**QUESTIONS**
1.	**What is meant by Object Oriented Programming?** **BTL 1** OOP is a method of programming in which programs are organised as cooperative collections of objects. Each object is an instance of a class and each class belong to a hierarchy.
2.	**What is a Class?** **BTL 1** Class is a template for a set of objects that share a common structure and a common behaviour.
3.	**What is an Object?** **BTL 2** Object is an instance of a class. It has state,behaviour and identity. It is also called as an instance of a class.
4.	**What is an Instance?** **BTL 1** An instance has state, behaviour and identity. The structure and behaviour of similar classes are defined in their common class. An instance is also called as an object.

5.	**What are the core OOP's concepts? BTL 2** ➢ Abstraction, Encapsulation,Inheritance and Polymorphism are the core OOP's concepts.
6.	**What is meant by abstraction? NOV/DEC 2018 BTL 5** Abstraction defines the essential characteristics of an object that distinguish it from all other kinds of objects. Abstraction provides crisply-defined conceptual boundaries relative to the perspective of the viewer. It's the process of focussing on the essential characteristics of an object. Abstraction is one of the fundamental elements of the object model.
7.	**What is meant by Encapsulation? APR/MAY 2019 BTL 1** Encapsulation is the process of compartmentalising the elements of an abtraction that defines the structure and behaviour. Encapsulation helps to separate the contractual interface of an abstraction and implementation.
8.	**What are Encapsulation, Inheritance and Polymorphism? BTL 2** Encapsulation is the mechanism that binds together code and data it manipulates and keeps both safe from outside interference and misuse. Inheritance is the process by which one object acquires the properties of another object. Polymorphism is the feature that allows one interface to be used for general class actions.
9.	**What are methods and how are they defined? BTL 2** Methods are functions that operate on instances of classes in which they are defined. Objects can communicate with each other using methods and can call methods in other classes. Method definition has four parts. They are name of the method, type of object or primitive type the method returns, a list of parameters and the body of the method. A method's signature is a combination of the first three parts mentioned above.

10.	**What are different types of access modifiers (Access specifiers)? BTL 2** Access specifiers are keywords that determine the type of access to the member of a class. These keywords are for allowingprivileges to parts of a program such as functions and variables. These are: public: Any thing declared as public can be accessed from anywhere. private: Any thing declared as private can't be seen outside of its class. protected: Any thing declared as protected can be accessed by classes in the same package and subclasses in the other packages. default modifier : Can be accessed only to classes in the same package.
11.	**What is an Object and how do you allocate memory to it? BTL 3** Object is an instance of a class and it is a software unit that combines a structured set of data with a set of operations for inspecting and manipulating that data. When an object is created using new operator, memory is allocated to it.
12.	**Explain the usage of Java packages. BTL 1** This is a way to organize files when a project consists of multiple modules. It also helps resolve naming conflicts when different packages have classes with the same names. Packages access level also allows you to protect data from being used by the non-authorized classes.
13.	**What is method overloading and method overriding? NOV/DEC 2016 BTL 4** Method overloading: When a method in a class having the same method name with different arguments is said to be method overloading. Method overriding : When a method in a class having the same method name with same arguments is said to be method overriding
14.	**What gives java it's "write once and run anywhere" nature? BTL 4** All Java programs are compiled into class files that contain bytecodes. These byte codes can be run in any platform and hence java is said to be platform independent.
15.	**What is a constructor? What is a destructor? BTL 2** Constructor is an operation that creates an object and/or initialises its state. Destructor is an operation that frees the state of an object and/or destroys the object itself. In Java, there is no concept of destructors. It's taken care by the JVM.
16.	**What is the difference between constructor and method? BTL 2** Constructor will be automatically invoked when an object is created whereas method has to be called explicitly

17.	**What is Static member classes?** **BTL 1** ➢ A static member class is a static member of a class. Like any other static method, a static member class has access to all static methods of the parent, or top-level, class.
18.	**What is Garbage Collection and how to call it explicitly?** **BTL 1** When an object is no longer referred to by any variable, java automatically reclaims memory used by that object. This is known as garbage collection. System. gc() method may be used to call it explicitly.
19.	**In Java, How to make an object completely encapsulated?** **BTL 2** All the instance variables should be declared as private and public getter and setter methods should be provided for accessing the instance variables
20	**What is static variable and static method?** **BTL 2** Static variable is a class variable which value remains constant for the entire class. Static method is the one which can be called with the class itself and can hold only the staic variables
21	**What is finalize( ) method in Java?** **APR/MAY 2015** **BTL 1** finalize () method is used just before an object is destroyed and can be called just prior to garbage collection.
22	**What is the difference between String and String Buffer?** **BTL 2** a) String objects are constants and immutable whereas StringBuffer objects are not. b) String class supports constant strings whereas StringBuffer class supports growable and modifiable strings.
23	**What is a package?** **BTL 1** A package is a collection of classes and interfaces that provides a high-level layer of access protection and name space management.
24	**What is the difference between this() and super()?** **BTL 2** this() can be used to invoke a constructor of the same class whereas super() can be used to invoke a super class constructor.

25	**Explain working of Java Virtual Machine (JVM)? BTL 2** JVM is an abstract computing machine like any other real computing machine which first converts .java file into .class file by using Compiler (.class is nothing but byte code file.) and Interpreter reads byte codes.
	PART * B
1	**How Strings are handled in java? Explain with code, the creation of** **Substring, Concatenation and testing for equality. (13) NOV/DEC 2018 BTL 3** **Answer: Page No. 389 Herbert Schildt** Key Points: 1. Introduction to Strings (3) – Strings is the collection of characters. 2. Various Operations on Strings [Strcat,Strcpy,strlen,strrev](6) 3. Sample code explaining substring, concatenation and equality. (2) 4. Output with explanation (2)
2	**Explain with an example the following features of Constructors: (13)** **(i). Overloaded Constructors** **(ii). A Call to another constructor with this operator** **(iii). An object initialization block** **(iv). A static initialization block BTL 2** **Answer: Page No. 124 Herbert Schildt** Key Points: **1.** Introduction to constructor with sample code (3) [Whenever an object is created ,it will be automatically called] Sample code : Class student { Student() {

	}
	
	};
	
	2. Concept of overloading, constructor overloading with code (8) – [Multiple constructors inside the class is called overloading]
	3. Explanation of Object Initialization block (1)
	4. Explanation about static Initialization block (1)

3	**Write a java program to sort ten names in descending order. (13) BTL 5** **Answer: Page No. 153 Herbert Schildt** Key Points: **1.** Coding (include necessary comments) (11) **2.** Output explanation (2)
4	**Explain string handling classes in Java with examples. (13) APR/MAY 2016 BTL 3** **Answer: Page No. 389 Herbert Schildt** Key Points: **1.** String Concatenation (3) [strCat()] **2.** Character Extraction (3) [charAt()] **3.** String Comparison(3) [strCmp()] **4.** Modifying a string(3) **5.** valueOf( ) (1)
5	**Explain briefly the object oriented concepts. (13) BTL 1** **Answer: Page No. 18 Herbert Schildt** Key Points: 1. Abstraction (3) –gathering essential details and removing background details 2. Encapsulation (3) – binding of data members and member functions 3. Inheritance (3) – Deriving a sub class from super class 4. Polymorphism (3) – Ability to take more than one form **5.** Dynamic Binding and Message Passing.(1)

6	**How objects are constructed? Explain constructor overloading with an example. (13) MAY/JUNE 2017 BTL 3** **Answer: Page No. 124 Herbert Schildt** Key Points: **1.** Introduction to Constructors (3) – Whenever an object is created ,constructor will be called. **2.** Overloading Concept with example (4) **3.** Constructor overloading with code (6) – Multiple constructors in a class
7	**Write short notes on access specifiers in java. (13) BTL 2** **Answer: Page No. 190 Herbert Schildt** **Key Points:** **1.** Introduction to access specifiers. (1) **2.** Public (3) – can be accessed anywhere **3.** Private (3) – accessed only within the class **4.** Protected (2) – accessed only the inherited class **5.** Private protected (2) **6.** Default(1) **7.** Code snippet for each type with an example.(1)
8	**Explain arrays in java. (13) MAY/JUNE 2016 BTL 2** **Answer: Page No. 51 Herbert Schildt** Key Points: 1. Introduction to arrays (2) – Collection of similar data types which is stored under a common name. 2. Diagram representation with an example. (6) 3. Declaration, Creation and initialization of array (4) syntax: int arrayname[]=new datatype; **4.** Sample code with explanation (1)
9	**What is a Package? How does a complier locate packages? (13) NOV/DEC 2018 BTL 2** **Answer: Page No. 187 Herbert Schildt** Key points: **1.** Definition of Package. (3) – Collection of classes,interfaces and sub packages **2.** Diagram representation (7) **3.** Sample path of directory with explanation. (3)

	PART C

1	**Write a java program for push and pop operations in stack using arrays in classes and object. (15)** **BTL 4**
	
	Answer: Page No. 126 Herbert Schildt
	
	Key Points:
	
	1. Introduction to Stack (2) – LIFO [Last In First Out]
	2. Separate method for push and pop (11) –
	Push : public void push(int x)
	{
	if(top>maxsize)
	{
	System.out.println("Overflow");
	}
	else
	{
	Top++;
	Stack[top]=x;
	}
	}
	3. Output with explanation (2)

2	**Explain the usage of command line parameter. (15)** **BTL 4** **Answer: Page No. 328 Herbert Schildt** Key Points: 1. Definition about Command line arguments (3) –Used to get input at run time. 2. Sample code with explanation (9) 3. Output of the code through command line arguments. (3)

3	**Describe the static fields and methods used in java. (15) APR/MAY 2015 BTL 5** **Answer: Page No. 366 Herbert Schildt** Key Points: **1.** Definition of static data member (6) – Static is declared as datamember **2.** Definition of static member function(5) – Static is declared as memberfunction. **3.** Sample code with static field and method (4)

UNIT 2 – INHERITANCE AND INTERFACES	
Inheritance – Super classes- sub classes –Protected members – constructors in sub classes- the Object class – abstract classes and methods- final methods and classes – Interfaces – defining an interface, implementing interface, differences between classes and interfaces and extending interfaces - Object cloning -inner classes, Array Lists – Strings	
PART A	
1	**What is meant by Inheritance? BTL 1** Inheritance is a relationship among classes, wherein one class shares the structure or behaviour defined in another class. This is called Single Inheritance. If a class shares the structure or behaviour from multiple classes, then it is called Multiple Inheritance. Inheritance defines "is-a" hierarchy among classes in which one subclass inherits from one or more generalised superclasses.
2	**What is meant by Inheritance and what are its advantages? BTL 1** Inheritance is the process of inheriting all the features from a class. The advantages of inheritance are reusability of code and accessibility of variables and methods of the super class by subclasses.
3	**What is the difference between superclass and subclass? APR/MAY2018 BTL 4** A super class is a class that is inherited whereas sub class is a class that does the inheriting.

4	**Differentiate between a Class and an Object? NOV/DEC 2017 BTL 4** The Object class is the highest-level class in the Java class hierarchy. The Class class is used to represent the classes and interfaces that are loaded by a Java program. The Class class is used to obtain information about an object's design. A Class is only a definition or prototype of real life object. Whereas an object is an instance or living representation of real life object. Every object belongs to a class and every class contains one or more related objects.
5.	**What is meant by Binding? BTL 1** Binding denotes association of a name with a class
6.	**What is meant by Polymorphism? BTL 1** Polymorphism literally means taking more than one form. Polymorphism is a characteristic of being able to assign a different behavior or value in a subclass, to something that was declared in a parent class.
7	**What is Dynamic Binding? APR/MAY 2017 BTL 1** Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding (also known as late binding) means that the code associated with a given procedure call is not known until the time of the call at run-time. It is associated with polymorphism and inheritance.
8	**What is final modifier? BTL 1** The final modifier keyword makes that the programmer cannot change the value anymore. The actual meaning depends on whether it is applied to a class, a variable, or a method. • final Classes- A final class cannot have subclasses. • final Variables- A final variable cannot be changed once it is initialized. • final Methods- A final method cannot be overridden by subclasses.
9	**What is an Abstract Class? BTL 1** Abstract class is a class that has no instances. An abstract class is written with the expectation that its concrete subclasses will add to its structure and behaviour, typically by implementing its abstract operations.
10	**What are inner class and anonymous class? BTL 2** Inner class: classes defined in other classes, including those defined in methods are called inner classes. An inner class can have any accessibility including private. Anonymous

	class: Anonymous class is a class defined inside a method without a name and is instantiated and declared in the same place and cannot have explicit constructors
11	**What is an Interface? BTL 2** Interface is an outside view of a class or object which emphaizes its abstraction while hiding its structure and secrets of its behaviour.
12	**What is a base class? BTL 1** Base class is the most generalised class in a class structure. Most applications have such root classes. In Java, Object is the base class for all classes.
13	**What is reflection in java? BTL 2** Reflection allows Java code to discover information about the fields, methods and constructors of loaded classes and to dynamically invoke them.
14	**Define superclass and subclass. BTL 2** Superclass is a class from which another class inherits. Subclass is a class that inherits from one or more classes.
15	**What is meant by Binding, Static binding, Dynamic binding? BTL 1** Binding: Binding denotes association of a name with a class. Static binding: Static binding is a binding in which the class association is made during compile time. This is also called as Early binding. Dynamic binding: Dynamic binding is a binding in which the class association is not made until the object is created at execution time. It is also called as Late binding.
16	**What is reflection API? How are they implemented? BTL 1** Reflection is the process of introspecting the features and state of a class at runtime and dynamically manipulate at run time. This is supported using Reflection API with built-in classes like Class, Method, Fields, Constructors etc. Example: Using Java Reflection API we can get the class name, by using the getName method.
17	**What is the difference between a static and a non-static inner class?NOV/DEC 2019 BTL 2** A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.

18	**What is the difference between abstract class and interface? BTL 2** a) All the methods declared inside an interface are abstract whereas abstract class must have at least one abstract method and others may be concrete or abstract. b) In abstract class, key word abstract must be used for the methods whereas interface we need not use that keyword for the methods. c) Abstract class must have subclasses whereas interface can't have subclasses.
19	**Can you have an inner class inside a method and what variables can you access? BTL 4** Yes, we can have an inner class inside a method and final variables can be accessed.
20	**What is interface and its use? BTL 2** Interface is similar to a class which may contain method's signature only but not bodies and it is a formal set of method and constant declarations that must be defined by the class that implements it. Interfaces are useful for: a) Declaring methods that one or more classes are expected to implement. b) Capturing similarities between unrelated classes without forcing a class relation. c) Determining an object's programming interface without revealing the actual body of the class.
21	**How is polymorphism achieved in java? BTL 2** Inheritance, Overloading and Overriding are used to achieve Polymorphism in java.
22	**What modifiers may be used with top-level class? BTL 2** public, abstract and final can be used for top-level class.
23	**What is a cloneable interface and how many methods does it contain? BTL 1** It is not having any method because it is a TAGGED or MARKER interface.
24	**What are the methods provided by the object class? BTL 1** The Object class provides five methods that are critical when writing multithreaded Java programs: • notify

	• notifyAll • wait (three versions)
25	**What is object cloning? NOV/DEC 2017 BTL 1** It is the process of duplicating an object so that two identical objects will exist in the memory at the same time.
	PART B
1	**Explain about inheritance in java. (13) NOV/DEC 2017 BTL 2** Key Points: **1.** Introduction about inheritance (2) – Process of deriving a sub class from super class. **2.** Diagram(5) **3.** Usage of 'extends" keyword (2) – Inheriting super class. **4.** Superclass and subclass code (2) – Syntax of super class : Class Superclassname { } Syntax of Sub class : Class Subclassname extends Superclassname { } **5.** Sample code with output (2) Answer: Page No. 161 in Herbert Schildt
2	**State the properties of inheritance. (13) BTL 3** Key Points: **1.** Introduction about inheritance (5) –Process of deriving a sub class from super class **2.** Diagram (4) **3.** Usage of 'extends" keyword (1) – Inheriting super class **4.** Advantages of inheritance (1) -Reusability **5.** Rules to be followed in inheritance (2) Answer: Page No. 145 in Herbert Schildt
3	**What is dynamic binding? How it is achieved? (13) APR/MAY 2018 BTL 1** Key Points: **1.** Definition of Dynamic Binding (2) – Binding happens at run time. **2.** Difference between early and late binding (6) –In early binding ,binding happens at compile time whereas in late binding, happens at run time . Early binding is achieved through overloading and late binding achieved through overriding. **3.** Sample code with output.(5) Answer: Page No. 198 in Herbert Schildt

4	**Explain interfaces with example. (13) BTL 3** Key Points: **1.** Definition of interfaces (2) – Collection of final variables and abstract methods **2.** Usage of keyword "implements" (2) **3.** Diagrammatic explanation (4) **4.** Sample code illustrates the inheritance concept. (5) Answer: Page No. 196 in Herbert Schildt
5	**Explain briefly about multilevel inheritance with neat example. BTL 4** Key Points: **1.** Introduction to multilevel inheritance (3)- deriving a sub class from another sub class. **2.** Explanation with diagram (flowchart) (6) **3.** Sample code for multilevel inheritance. (4) Answer: Page No. 171 in Herbert Schildt
6	**Explain how inner classes and anonymous classes work in java program. (13) BTL 4** Key Points: **1.** Introduction to Inner classes (2) **2.** Sample code snippet (7) **3.** Anonymous class – Description (2) **4.** Sample code with output. (2) Answer: Page No. 731 in Herbert Schildt
	PART C
1	**Write a note on class hierarchy. How do you create hierarchical classes in Java? (15) BTL 4** Key Points: **1.** Introduction about inheritance (3) –Process of deriving a class from super class **2.** Diagram (5) **3.** Usage of 'extends" keyword (2) –Inheriting the super class. **4.** Superclass and subclass code (3) **5.** Sample code with output (2) Answer: Page No. 161 in Herbert Schildt
2	**What is a Package? What are the benefits of using packages? Write down the steps in creating a package and using it in a java program with an example. (15) NOV/DEC 2016 BTL 5** Key points: **1.** Definition of Package. (3) **2.** Diagram representation (4) **3.** Sample path of directory with explanation (4) **4.** Advantageous of Packages (4)

	Answer: Page No. 187 in Herbert Schildt
3	**Differentiate method overloading and method overriding. Explain both with an example program. (15) MAY/JUNE 2017 BTL 1** Key Points: 1. Concept of Overloading (3) – Function which has the same name but differs with different arguments or different types 2. Concept of Overriding (3) – Function which has the same name with same no of arguments. 3. Difference between Overloading and overriding (3) **4.** Explanation with an example**. (**6) Answer: Page No. 158, 286 in Herbert Schildt

UNIT -3: EXCEPTION HANDLING AND I/O	
Exceptions - exception hierarchy - throwing and catching exceptions – built-in exceptions, creating own exceptions, Stack Trace Elements. Input / Output Basics – Streams – Byte streams and Character streams – Reading and Writing Console – Reading and Writing Files	
PART A	
1	**What is an exception? NOV/DEC 2019 BTL 2** An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.
2	**What is error? BTL 1** An Error indicates that a non-recoverable condition has occurred that should not be caught. Error, a subclass of Throwable, is intended for drastic problems, such as OutOfMemoryError, which would be reported by the JVM itself.
3	**Which is superclass of Exception? BTL 1** "Throwable", the parent class of all exception related classes.
4	**What are the advantages of using exception handling? APR/MAY 2018 BTL 2** Exception handling provides the following advantages over "traditional" error management techniques: Separating Error Handling Code from "Regular" Code. Propagating Errors Up the Call Stack. Grouping Error Types and Error Differentiation.

5	**What are the types of Exceptions in Java? NOV/DEC 2019 BTL 1** There are two types of exceptions in Java, unchecked exceptions and checked exceptions. **Checked exceptions**: A checked exception is some subclass of Exception (or Exception itself), excluding class RuntimeException and its subclasses. Each method must either handle all checked exceptions by supplying a catch clause or list each unhandled checked exception as a thrown exception. **Unchecked exceptions**: All Exceptions that extend the RuntimeException class are unchecked exceptions. Class Error and its subclasses also are unchecked.
6	**Why Errors are Not Checked? BTL 4** A unchecked exception classes which are the error classes (Error and its subclasses) are exempted from compile-time checking because they can occur at many points in the program and recovery from them is difficult or impossible. A program declaring such exceptions would be pointlessly.
7	**How does a try statement determine which catch clause should be used to handle an exception? BTL 1** When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exception is executed. The remaining catch clauses are ignored.
8	**What is the purpose of the finally clause of a try-catch-finally statement? BTL 2** The finally clause is used to provide the capability to execute code no matter whether or not an exception is thrown or caught.
9	**What is the difference between checked and Unchecked Exceptions in Java? BTL 4** All predefined exceptions in Java are either a checked exception or an unchecked exception. Checked exceptions must be caught using try.. catch () block or we should throw the exception using throws clause. If you don't, compilation of program will fail.
10	**What is the difference between exception and error? BTL 2** The exception class defines mild error conditions that your program encounters. Exceptions can occur when trying to open the file, which does not exist, the network connection is disrupted, operands being manipulated are out of prescribed ranges, the class file you are interested in loading is missing. The error class defines serious error conditions that you should not attempt to recover from. In most cases it is advisable to let the program

	terminate when such an error is encountered.
11	**What is the catch or declare rule for method declarations? BTL 2** If a checked exception may be thrown within the body of a method, the method must either catch the exception or declare it in its throws clause.
12	**When is the finally clause of a try-catch-finally statement executed? BTL 2** The finally clause of the try-catch-finally statement is always executed unless the thread of execution terminates or an exception occurs within the execution of the finally clause.
13	**What if there is a break or return statement in try block followed by finally block? BTL 2** If there is a return statement in the try block, the finally block executes right after the return statement encountered, and before the return executes.
14	**What are the different ways to handle exceptions? BTL 2** There are two ways to handle exceptions: Wrapping the desired code in a try block followed by a catch block to catch the exceptions. List the desired exceptions in the throws clause of the method and let the caller of the method handle those exceptions.

15	**How to create custom exceptions? BTL 1**
	By extending the Exception class or one of its subclasses.
	Example:
	class MyException extends Exception {
	public MyException() { super(); }
	public MyException(String s) { super(s); }
	}

16	**Can we have the try block without catch block? BTL 2** Yes, we can have the try block without catch block, but finally block should follow the try block.

	Note: It is not valid to use a try clause without either a catch clause or a finally clause.
17	**What is the difference between swing and applet? BTL 4** Swing is a light weight component whereas Applet is a heavy weight Component. Applet does not require main method, instead it needs init method.
18	**What is the use of assert keyword? BTL 2** Assert keyword validates certain expressions. It replaces the if block effectively and throws an AssertionError on failure. The assert keyword should be used only for critical arguments (means without that the method does nothing).
19	**How does finally block differ from finalize() method? NOV/DEC 2016 BTL 2** Finally block will be executed whether or not an exception is thrown. So it is used to free resoources. finalize() is a protected method in the Object class which is called by the JVM just before an object is garbage collected.
20	**What is the difference between throw and throws clause? APR/MAY 2017 BTL 2** throw is used to throw an exception manually, where as throws is used in the case of checked exceptions, to tell the compiler that we haven't handled the exception, so that the exception will be handled by the calling function.
21	**What are the different ways to generate and Exception? BTL 2** There are two different ways to generate an Exception. 1. Exceptions can be generated by the Java run-time system. Exceptions thrown by Java relate to fundamental errors that violate the rules of the Java language or the constraints of the Java execution environment. 2. Exceptions can be manually generated by your code. Manually generated exceptions are typically used to report some error condition to the caller of a method.
22	**Where does Exception stand in the Java tree hierarchy? BTL 2** • java.lang.Object • java.lang.Throwable • java.lang.Exception • java.lang.Error

23	**What is StackOverflowError? NOV/DEC 2018 BTL 1**
	
	The StackOverFlowError is an Error Object thorwn by the Runtime System when it Encounters that your application/code has ran out of the memory. It may occur in case of recursive methods or a large amount of data is fetched from the server and stored in some object. This error is generated by JVM.
	
	e.g. void swap(){
	
	swap();
	
	}

24	**Brief about the exception hierarchy in java. BTL 2** The hierarchy is as follows: Throwable is a parent class off all Exception classes. They are two types of Exceptions: Checked exceptions and UncheckedExceptions. Both type of exceptions extends Exception class
25	**How do you get the descriptive information about the Exception occurred during the program execution? BTL 2** All the exceptions inherit a method printStackTrace() from the Throwable class. This method prints the stack trace from where the exception occurred. It prints the most recently entered method first and continues down, printing the name of each method as it works its way down the call stack from the top.
	PART B

1	**Discuss on Exception handling in detail. (13) NOV/DEC 2018 BTL 2**
	Key points:
	1. Creation of Exception class (4)
	2. Sample code includes try and catch block (4)
	Try {
	}
	catch(Exception e)
	{
	}
	Finally
	{
	}
	3. Catching exceptions (2)
	4. Sample code for different exceptions (3)
	Answer: Page No. 299 in Herbert Schildt

2	**Explain briefly about user defined exceptions and stack trace elements in exception handling mechanisms. (13) BTL 1** Key Points: 1. Concept of Exception and exception handling (4) 2. Predefined and userdefined exceptions (5) Predefined – ArithmeticException,ArrayOutOfBoundException,SQLException,IOException. **3.** Sample code for userdefined exception with output (4) Answer: Page No. 221 in Herbert Schildt
3	**Explain the task of catching exceptions with example. (13) APR/MAY 2018 BTL 2** Key points: **1.** Creation of Exception class (4) **2.** Sample code includes try and catch block (4) **3.** Catching exceptions (2) **4.** Sample code for different exceptions (3) Answer: Page No. 207 in Herbert Schildt
4	**Explain in detail about reading and writing files in JAVA. (13) BTL 1** Key Points: 1. FileInputStream class (4) 2. FileOutputStream class (4) **3.** Sample code with explanation (5) Answer: Page No. 661 in Herbert Schildt
5	**Brief about the following classes: (13) BTL 1** **(a). Byte Stream** **(b). Character Stream** **(c). PrintWriter class** Key points: **1.** Explanation about the above-said class (3+3+3) **2.** Sample code for each class with the concept.(2+1+1) Answer: Page No. 582 in Herbert Schildt
	PART C
1	**Explain the task of catching exceptions with example. (15) NOV/DEC 2017 BTL 2** Key points: **1.** Creation of Exception class (5)

	2. Sample code includes try and catch block (4) **3.** Catching exceptions (3) **4.** Sample code for different exceptions (3) Answer: Page No. 207 in Herbert Schildt
2	**Describe about how JAVA handles overflows and underflows. (15) BTL 4** Key Points: 1. Overflow or underflow conditions never throw a run time exception (5) 2. Flowed output is predictable and reproducible. That is, its behaviour is the same every time you run the program.(5) 3. Sample code (5) Answer: Page No. 223 in Herbert Schildt
3	**Discuss the concept of exception handling with an application of your choice. Write necessary code snippets. (15) MAY/JUNE 2017 BTL 6** Key points: **1.** Creation of Exception class (5) **2.** Sample code includes try and catch block (6) **3.** Catching exceptions (2) **4.** Sample code for different exceptions(2) Answer: Page No. 299 in Herbert Schildt

UNIT 4 - MULTITHREADING AND GENERIC PROGRAMMING

Differences between multi-threading and multitasking, thread life cycle, creating threads, synchronizing threads, Inter-thread communication, daemon threads, thread groups. Generic Programming – Generic classes – generic methods – Bounded Types – Restrictions and Limitations.

PART A

1	**Explain different way of using thread? BTL 1** The thread could be implemented by using runnable interface or by inheriting from the Thread class. The former is more advantageous, 'cause when you are going for multiple inheritance, the only interface can help.
2	**What are the different states of a thread ? BTL 1** The different thread states are ready, running, waiting and dead.
3	**Why are there separate wait and sleep methods? BTL 1** The static Thread.sleep(long) method maintains control of thread execution but delays the next action until the sleep time expires. The wait method gives up control over thread

	execution indefinitely so that other threads can run.
4	**What is multithreading and what are the methods for inter-thread communication and what is the class in which these methods are defined? BTL 2** Multithreading is the mechanism in which more than one thread run independent of each other within the process. wait (), notify () and notifyAll() methods can be used for inter-thread communication and these methods are in Object class. wait() : When a thread executes a call to wait() method, it surrenders the object lock and enters into a waiting state. notify() or notifyAll() : To remove a thread from the waiting state, some other thread must make a call to notify() or notifyAll() method on the same object.
5	**What is synchronization and why is it important? BTL 2** With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources. Without synchronization, it is possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often leads to significant errors.
6	**How does multithreading take place on a computer with a single CPU? BTL1** The operating system's task scheduler allocates execution time to multiple tasks. By quickly switching between executing tasks, it creates the impression that tasks execute sequentially.
7	**What is the difference between process and thread? NOV/DEC2018 BTL1** Process is a program in execution whereas thread is a separate path of execution in a program.
8	**What happens when you invoke a thread's interrupt method while it is sleeping or waitIng? BTL 1** When a task's interrupt() method is executed, the task enters the ready state. The next time the task enters the running state, an InterruptedException is thrown.
9	**How can we create a thread? BTL 2** A thread can be created by extending Thread class or by implementing Runnable interface. Then we need to override the method public void run().
10	**What are three ways in which a thread can enter the waiting state? BTL 2** A thread can enter the waiting state by invoking its sleep() method, by blocking on I/O, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's wait()

	method. It can also enter the waiting state by invoking its (deprecated) suspend() method.
11	**How can i tell what state a thread is in ? BTL 2** Prior to Java 5, isAlive() was commonly used to test a threads state. If isAlive() returned false the thread was either new or terminated but there was simply no way to differentiate between the two.
12	**What is synchronized keyword? In what situations you will Use it? BTL 1** Synchronization is the act of serializing access to critical sections of code. We will use this keyword when we expect multiple threads to access/modify the same data. To understand synchronization we need to look into thread execution manner.
13	**What is serialization? BTL 1** Serialization is the process of writing complete state of java object into output stream, that stream can be file or byte array or stream associated with TCP/IP socket.
14	**What does the Serializable interface do? APR/MAY 2016 BTL 1** Serializable is a tagging interface; it prescribes no methods. It serves to assign the Serializable data type to the tagged class and to identify the class as one which the developer has designed for persistence. ObjectOutputStream serializes only those objects which implement this interface.
15	**When you will synchronize a piece of your code? BTL 2** When you expect your code will be accessed by different threads and these threads may change a particular data causing data corruption.
16	**What is daemon thread and which method is used to create the daemon thread? BTL 4** Daemon thread is a low priority thread which runs intermittently in the back ground doing the garbage collection operation for the java runtime system. setDaemon method is used to create a daemon thread.
17	**What is the difference between yielding and sleeping? BTL 4** When a task invokes its yield() method, it returns to the ready state. When a task invokes its sleep() method, it returns to the waiting state.
18	**What is casting? NOV/DEC 2018 BTL 2**

	There are two types of casting, casting between primitive numeric types and casting between object references. Casting between numeric types is used to convert larger values, such as double values, to smaller values, such as byte values. Casting between object references is used to refer to an object by a compatible class, interface, or array type reference.
19	**What classes of exceptions may be thrown by a throw statement? BTL 4** A throw statement may throw any expression that may be assigned to the Throwable type.
20	**A Thread is runnable, how does that work? BTL 4** The Thread class' run method normally invokes the run method of the Runnable type it is passed in its constructor. However, it is possible to override the thread's run method with your own.
21	**Can I implement my own start() method? BTL 4** The Thread start() method is not marked final, but should not be overridden. This method contains the code that creates a new executable thread and is very specialised. Your threaded application should either pass a Runnable type to a new Thread, or extend Thread and override the run() method.
22	**Do I need to use synchronized on setValue(int)? BTL 1** It depends whether the method affects method local variables, class static or instance variables. If only method local variables are changed, the value is said to be confined by the method and is not prone to threading issues.
23	**What is thread priority? BTL 2** Thread Priority is an integer value that identifies the relative order in which it should be executed with respect to others. The thread priority values ranging from 1- 10 and the default value is 5. But if a thread have higher priority doesn't means that it will execute first. The thread scheduling depends on the OS.
24	**What are the different ways in which a thread can enter into waiting state?** **BTL 2** There are three ways for a thread to enter into waiting state. By invoking its sleep() method, by blocking on I/O, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's wait() method.
25	**How would you implement a thread pool? BTL 2** The ThreadPool class is a generic implementation of a thread pool, which takes the

	following input Size of the pool to be constructed and name of the class which implements Runnable (which has a visible default constructor) and constructs a thread pool with active threads that are waiting for activation. once the threads have finished processing they come back and wait once again in the pool.
	PART B
1	**How generic methods and generic expressions are translated? (13) BTL 3** Key Points: **1.** Concept of Generic methods (4) **2.** Generic code (4) **3.** Virtual machine (5) Answer: Page No. 366 in Herbert Schildt
2	**Explain in detail, the inheritance rules for generic types. (13) BTL 1** Key Points**:** **1.** Introduction about Inheritance (5) **2.** How generics can be used in inheritance? (4) **3.** Sample code with explanation (2) **4.** Output of the sample code(2) Answer: Page No. 359 in Herbert Schildt
3	**What are interrupting threads? Explain thread states and synchronization? (13) BTL 4** Key Points: 1. Concept of interrupting thread (5) 2. Different kinds of thread states with an example. (4) – newborn state,running state,runnable state,dead state,blocked state 3. Explanation about synchronization (2) [One thread finishes its execution, then only the next thread starts] **4.** Sample code with output(2) Answer: Page No. 437 in Herbert Schildt
4	**Explain the various state of thread. (13) BTL 5** Key Points: **1.** Explanation about threads (3) – Each and every part of a program **2.** New state (2) **3.** Runnable state(2) **4.** Blocked state (2) **5.** Ready state (2) **6.** Sample code with explanation(2) Answer: Page No. 231 in Herbert Schildt

5	**Explain the process of synchronization in detail with suitable example. (13) BTL 3** Key Points: 1. Concept of Synchronization (5) 2. Usage of keyword "synchronized" and "volatile" (5) 3. Sample code with explanation (3) Answer: Page No. 241 in Herbert Schildt
	PART C
1	**Explain the procedure for running a task in a separate thread and running multiple threads. (15) MAY/JUNE 2017 BTL 5** Key Points: **1.** Explanation about threads (5) **2.** New state(3) **3.** Runnable state (2) **4.** Blocked state (2) **5.** Ready state(2) **6.** Sample code with explanation(1) Answer: Page No. 237 in Herbert Schildt
2	**Explain the States of a thread with a neat diagram. (15) APR/MAY 2018 BTL 4** Key Points: **1.** Explanation about threads (5) **2.** New state (2) **3.** Runnable state (2) **4.** Blocked state (2) **5.** Ready state(2) 6.Sample code with explanation (2) Answer: Page No. 231 in Herbert Schildt
3	**What is Generic programming and why is it needed? List the limitations and restrictions of generic programming. (15) NOV/DEC 2019 BTL 2** Key Points: **1.** Conept of Generic methods (4) **2.** Generic code (6) **3.** Virtual machine (3) **4.** Limitations of Generic Programming (2) Answer: Page No. 361 in Herbert Schildt

UNIT-5 EVENT DRIVEN PROGRAMMING

Graphics programming - Frame – Components - working with 2D shapes - Using color, fonts, and images -Basics of event handling - event handlers - adapter classes - actions - mouse events- AWT event hierarchy -Introduction to Swing – layout management - Swing Components – Text Fields, Text Areas – Buttons-Check Boxes – Radio Buttons – Lists- choices- Scrollbars – Windows – Menus – Dialog Boxes

PART A

1	**What is the relationship between the Canvas class and the Graphics class? (BTL 4)** A Canvas object provides access to a Graphics object via its paint() method.
2	**How would you create a button with rounded edges? (BTL 3)** There's 2 ways. The first thing is to know that a JButton's edges are drawn by a Border. so you can override the Button's paintComponent(Graphics) method and draw a circle or rounded rectangle (whatever), and turn off the border. Or you can create a custom border that draws a circle or rounded rectangle around any component and set the button's border to it.
3	**What is the difference between the 'Font' and 'FontMetrics' class? (BTL 2)** The Font Class is used to render 'glyphs' - the characters you see on the screen. FontMetrics encapsulates information about a specific font on a specific Graphics object. (width of the characters, ascent, descent)
4	**What is the difference between the paint() and repaint() methods? (BTL 2)** The paint() method supports painting via a Graphics object. The repaint() method is used to cause paint() to be invoked by the AWT painting thread.
5	**Which containers use a border Layout as their default layout?NOV/DEC 2018 (BTL 1)** The window, Frame and Dialog classes use a border layout as their default layout.
6	**What is the difference between applications and applets? BTL 2** a). Application must be run on local machine whereas applet needs no explicit installation on local machine. b). Application must be run explicitly within a java-compatible virtual machine whereas applet loads and runs itself automatically in a java-enabled browser.

	c). Application starts execution with its main method whereas applet starts execution with its init method. d). Application can run with or without graphical user interface whereas applet must run within a graphical user interface.
7	**Difference between Swing and Awt? BTL 2** AWT are heavy-weight componenets. Swings are light-weight components. Hence swing works faster than AWT.
8	**What is a layout manager and what are different types of layout managers available in java AWT? BTL 1** A layout manager is an object that is used to organize components in a container. The different layouts are available are FlowLayout, BorderLayout, CardLayout, GridLayout and GridBagLayout.
9	**How are the elements of different layouts organized? BTL 2** FlowLayout: The elements of a FlowLayout are organized in a top to bottom, left to right fashion. BorderLayout: The elements of a BorderLayout are organized at the borders (North, South, East and West) and the center of a container. CardLayout: The elements of a CardLayout are stacked, on top of the other, like a deck of cards. GridLayout: The elements of a GridLayout are of equal size and are laid out using the square of a grid. GridBagLayout: The elements of a GridBagLayout are organized according to a grid. However, the elements are of different size and may occupy more than one row or column of the grid. In addition, the rows and columns may have different sizes. The default Layout Manager of Panel and Panel sub classes is FlowLayout.
10	**Why would you use SwingUtilities.invokeAndWait or SwingUtilities.invokeLater? (BTL 4)** I want to update a Swing component but I'm not in a callback. If I want the update to happen immediately (perhaps for a progress bar component) then I'd use invokeAndWait. If I don't care when the update occurs, I'd use invokeLater.
11	**What is an event and what are the models available for event handling? BTL 1**

	An event is an event object that describes a state of change in a source. In other words, event occurs when an action is generated, like pressing button, clicking mouse, selecting a list, etc. There are two types of models for handling events and they are: a) event-inheritance model and b) event-delegation model
12	**What is the difference between scrollbar and scrollpane? BTL 1** A Scrollbar is a Component, but not a Container whereas Scrollpane is a Conatiner and handles its own events and perform its own scrolling.
13	**Why won't the JVM terminate when I close all the application windows? BTL 4** The AWT event dispatcher thread is not a daemon thread. You must explicitly call System.exit to terminate the JVM.
14	**What is meant by controls and what are different types of controls in AWT? (BTL 1)** Controls are components that allow a user to interact with your application and the AWT supports the following types of controls: Labels, Push Buttons, Check Boxes, Choice Lists, Lists, Scrollbars, and Text Components. These controls are subclasses of Component.
15	**What is the difference between a Choice and a List? BTL 1** A Choice is displayed in a compact form that requires you to pull it down to see the list of available choices. Only one item may be selected from a Choice. A List may be displayed in such a way that several List items are visible. A List supports the selection of one or more List items.
16	**What is the purpose of the enableEvents() method? BTL 2** The enableEvents() method is used to enable an event for a particular object. Normally,an event is enabled when a listener is added to an object for a particular event. The enableEvents() method is used by objects that handle events by overriding their eventdispatch methods.
17	**What is the difference between the File and RandomAccessFile classes? BTL 2** The File class encapsulates the files and directories of the local file system. The RandomAccessFile class provides the methods needed to directly access data contained in any part of a file.
18	**What is the lifecycle of an applet? BTL 2** init() method - Can be called when an applet is first loaded start() method - Can be called

	each time an applet is started. paint() method - Can be called when the applet is minimized or maximized. stop() method - Can be used when the browser moves off the applet's page. destroy() method - Can be called when the browser is finished with the applet.
19	**What is the difference between a MenuItem and a CheckboxMenuItem? BTL 2** The CheckboxMenuItem class extends the MenuItem class to support a menu item that may be checked or unchecked.
20	**What class is the top of the AWT event hierarchy? BTL 1** The java.awt.AWTEvent class is the highest-level class in the AWT event-class hierarchy.
21	**What is source and listener? NOV/DEC 2017 BTL 1** source : A source is an object that generates an event. This occurs when the internal state of that object changes in some way. listener : A listener is an object that is notified when an event occurs. It has two major requirements. First, it must have been registered with one or more sources to receive notifications about specific types of events. Second, it must implement methods to receive and process these notifications.
22	**Explain how to render an HTML page using only Swing. BTL 1** Use a JEditorPane or JTextPane and set it with an HTMLEditorKit, then load the text into the pane.
23	**How would you detect a keypress in a JComboBox? BTL 1** This is a trick, most people would say 'add a KeyListener to the JComboBox' - but the right answer is 'add a KeyListener to the JComboBox's editor component.'
24	**What is an I/O filter? NOV/DEC 2018 BTL 1** An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another.
25	**How can I create my own GUI components? BTL 1** Custom graphical components can be created by producing a class that inherits from java.awt.Canvas. Your component should override the paint method, just like an applet does, to provide the graphical features of the component.
	PART B

1	**Describe the sophisticated layout management in user interface component with example. (13) BTL 2** Key Points: 1. Explanation of layout manager class (5) – How content should appear in output 2. FlowLayout (3) 3. BorderLayout (2) 4. GridLayout (2) **5.** Explanation with sample code and output (1) Answer: Page No. 796 in Herbert Schildt
2	**State and explain in detail the basic of event handling. (13) APR/MAY 2018 BTL 1** Key Points: 1. Event Sources (4) 2. Event Classes (3) 3. Event Listeners(3) **4.** Event Adapters(3) Answer: Page No. 707 in Herbert Schildt
3	**Write a short notes on** **(i). JLabel** **(ii). JButton** **(iii). Layout Managers (13) BTL 1** Key Points: 1. Concepts of JLabel, JButton, Layout managers (3+3+3) 2. Sample code with output.(2+2) Answer: Page No. 950,949,707 in Herbert Schildt
4	**Write short notes on the following : (13) NOV/DEC 2017 BTL 1** **(i) Graphics programming** (ii) **Frame** Key Points: **1.** Concept of graphics Context (5) **2.** Graphics class drawing methods (4) –Lines,Rectangke,Circle,Polygon **3.** Explanation with sample code.(4) Answer: Page No. 307, 736 in Herbert Schildt

5	**List the methods available to draw shapes. (13) BTL 1** Key points: **1.** Shape Operations (6) **2.** Text Operations (4) **3.** Image Operations (3) Answer: Page No. 749 in Herbert Schildt
	PART C
1	**Explain the AWT Event handling in detail. (15) NOV/DEC 2019 BTL 1** Key Points: 1. Event Sources (4) 2. Event Classes (3) 3. Event Listeners(4) 4. Event Adapters (4) Answer: Page No. 736 in Herbert Schildt
2	**How is a Frame created? Write a java program that creates a product enquirer form using frames. (15) MAY/JUNE 2017 BTL 3** Key Points: 1. Concept of Frame (5) 2. Usage of JFrame (6) **3.** Sample code with explanation (4) Answer: Page No. 736 in Herbert Schildt
3	**Explain any five swing components with an example program. (15) APR/MAY 2016 BTL 2** Key Points: **1.** JPanel (3) **2.** JFrame (3) **3.** JInternalframe (2) **4.** JWindow (2) **5.** JDialog (2) **6.** JLabel (3) Answer: Page No. 949, 950, 949 in Herbert Schildt

REGULATION: 2017 **ACADEMIC YEAR: 2019-2020**

EC8394 ANALOG AND DIGITAL COMMUNICATION L T P C 3 0 0 3

OBJECTIVES:

Understand analog and digital communication techniques.

- Learn data and pulse communication techniques.
- Be familiarized with source and Error control coding.
- Gain knowledge on multi-user radio communication.

UNIT I - ANALOG COMMUNICATION 9

Introduction to Communication Systems - Modulation – Types - Need for Modulation. Theory of Amplitude Modulation - Evolution and Description of SSB Techniques - Theory of Frequency and Phase Modulation – Comparison of Analog Communication Systems (AM – FM – PM).

UNIT II - PULSE AND DATA COMMUNICATION 9

Pulse Communication: Pulse Amplitude Modulation (PAM) – Pulse Time Modulation (PTM) – Pulse code Modulation (PCM) - Comparison of various Pulse Communication System (PAM – PTM – PCM). **Data Communication:** History of Data Communication - Standards Organizations for Data Communication- Data Communication Circuits - Data Communication Codes - Data communication Hardware - serial and parallel interfaces.

UNIT III - DIGITAL COMMUNICATION 9

Amplitude Shift Keying (ASK) – Frequency Shift Keying (FSK)–Phase Shift Keying (PSK) – BPSK – QPSK – Quadrature Amplitude Modulation (QAM) – 8 QAM – 16 QAM – Bandwidth Efficiency– Comparison of various Digital Communication System (ASK – FSK – PSK – QAM).

UNIT IV - SOURCE AND ERROR CONTROL CODING 9

Entropy, Source encoding theorem, Shannon fano coding, Huffman coding, mutual information, channel capacity, Error Control Coding, linear block codes, cyclic codes - ARQ Techniques.

UNIT V MULTI-USER RADIO COMMUNICATION 9

Global System for Mobile Communications (GSM) - Code division multiple access (CDMA) – Cellular Concept and Frequency Reuse - Channel Assignment and Handover Techniques - Overview of Multiple Access Schemes - Satellite Communication - Bluetooth.

TOTAL: 45 PERIODS

OUTCOMES:

At the end of the course, the student should be able to:

- Apply analog and digital communication techniques.
- Use data and pulse communication techniques.
- Analyze Source and Error control coding.
- Utilize multi-user radio communication.

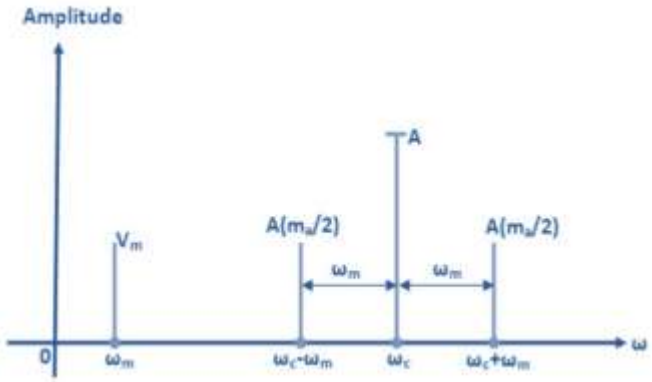
TEXT BOOK:

1. Wayne Tomasi, ―Advanced Electronic Communication Systems‖, 6th Edition, Pearson Education, 2009.

REFERENCES:

1. Simon Haykin, ―Communication Systems‖, 4th Edition, John Wiley & Sons, 2004
2. Rappaport T.S, "Wireless Communications: Principles and Practice", 2nd Edition, Pearson Education, 2007
3. H.Taub, D L Schilling and G Saha, ―Principles of Communication‖, 3rd Edition, Pearson Education, 2007.
4. B. P.Lathi, ―Modern Analog and Digital Communication Systems‖, 3rd Edition, Oxford University Press, 2007.
5. Blake, ―Electronic Communication Systems‖, Thomson Delmar Publications, 2002.
6. Martin S.Roden, ―Analog and Digital Communication System‖, 3rd Edition, Prentice Hall of India, 2002.
7. B.Sklar, ―Digital Communication Fundamentals and Applications‖ 2nd Edition Pearson Education 2007.

Subject Code: EC8394 **Year/Semester: II /03**
Subject Name: ANALOG & DIGITAL COMMUNICATION **Subject Handler: Mrs.R.Ramakala**

UNIT I - ANALOG COMMUNICATION	
Introduction to Communication Systems - Modulation – Types - Need for Modulation. Theory of Amplitude Modulation - Evolution and Description of SSB Techniques - Theory of Frequency and Phase Modulation – Comparison of Analog Communication Systems (AM – FM – PM).	
PART * A	
Q.No.	**Questions**
1	**Define Modulation? BTL1** Modulation is the process of changing any one parameter (amplitude,frequency or phase) of a relatively high frequency carrier signal in proportion with the instantaneous value of the modulating signal or message signal.
2	**What is the need for modulation? (Nov/Dec 2014)** BTL1 In order to carry the low frequency message signal to a longer distance, the high frequency carrier signal is combined with it. a) Reduction in antenna height b) Long distance communication c) Ease of radiation d) Multiplexing e) Improve the quality of reception f) Avoid mixing up of other signals
3	**Draw the spectrum of AM signal? (April/ May 2015)** BTL3 
4	**Define AM (Nov/Dec 2015)(APR/MAY-17)** BTL1 Amplitude Modulation is defined as changing the amplitude of the carrier signal with respect to the instantaneous change in message signal.
5	**What is Modulation Index and Percentage modulation?((May/June 2014)** BTL1 Modulation Index (or) Coefficient of Modulation (or) Depth of modulation It is defined as ratio of amplitude of the message signal to the amplitude of the carrier signal **ma=Em/Ec**

	Percentage Modulation It is defined as the percentage change in the amplitude of the output wave when the carrier is acted on by a modulating signal. **% ma=(Em/Ec)*100**
6	**Define Angle modulation (May/June 2014)** BTL1 Angle Modulation can be classified into **(a) Frequency Modulation:** Frequency Modulation is defined as changing the frequency of the carrier signal with respect to the instantaneous change in message signal. **(b) Phase modulation:** Phase Modulation is defined as changing the phase of the carrier signal with respect to the instantaneous change in message signal. Hence changing the time parameters such as frequency or the phase of the carrier signal with respect to instantaneous change in message signal is called Angle Modulation **Advantages:** (i) Noise reduction. (ii) Improved system fidelity. (iii) More efficient use of power.
7	**What is demodulation? (May/June 2013)** BTL1 The process of recovering the original modulating signal from a modulated signal at the receiver is termed as demodulation (or) detection process.
8	**Define frequency deviation and phase deviation (April /May 2015)** BTL1 Frequency deviation is the change in frequency that occurs in the carrier when it is acted on by a modulating signal frequency. Frequency deviation is typically given as a peak frequency shift in Hertz (Δf). The peak-to-peak frequency deviation (2Δf) is sometimes called carrier swing. The peak frequency deviation is simply the product of the deviation sensitivity and the peak modulating signal voltage and is expressed Mathematically as **Δf = KfVm Hz.** The relative angular displacement (shift) of the carrier phase (radians) in respect to reference phase is called phase deviation **(ΔΘ)**
9	**Define repetition rate of the AM Envelope? (April/May 2015)** BTL1 The repetition rate of the envelope is equal to the frequency of the modulating signal, and the shape of the envelope is identical to the shape of the modulating signal.
10	**Differentiate between narrow band and wide band FM? (Nov/Dec 2013)** BTL2

Sl.No	Wide Band FM (WBFM)	Narrow Band FM (NBFM)
1	Modulation index is greater than 1	Modulation index is less than 1
2	Frequency deviation = 75 kHz	Frequency deviation = 5 kHz
3	Modulating frequency ranges from 30Hz to 15 kHz	Modulating frequency = 3 kHz
4	Bandwidth is 15 times that of narrow band FM	Bandwidth = $2f_m$
5	Noise is more suppressed	Less suppression of noise
6	Used in broadcasting	Used in mobile communication

11 **Define the bandwidth of AM** BTL1

Bandwidth (B) of AM (DSBFC) is the difference between highest upper side band frequency and lowest lower side band frequency.

B = FUSB - FLSB = (fc+fm)-(fc-fm) = 2fm

where fm(max)– maximum modulating signal frequency.

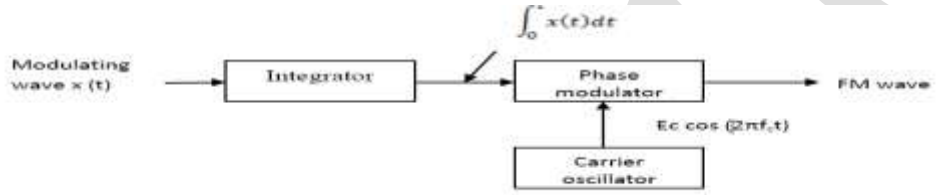
12 **A broadcast radio transmitter radiates 5 kW power when the modulation ercentage is 60%. How much is the carrier power?** BTL1

$P_t = P_c(1+m^2/2) = 5000/(1+0.6^2/2) = 4237.28w$

13 **What is the relationship between frequency and phase modulation? (May 2016)** BTL2

PM demodulator is obtained from FM demodulator by placing an integrator followed by FM modulator.

FM modulator is obtained from PM modulator by placing an integrator followed by a PM modulator.



14 **Compare AM with DSB-SC and SSB-SC.(NOV/DEC-15)** BTL4

Description	AM with carrier	DSB-SC AM	SSB-SC-AM
Bandwidth	2fm	2fm	Fm
Power Saving	33.33%	66.66%	83.3%

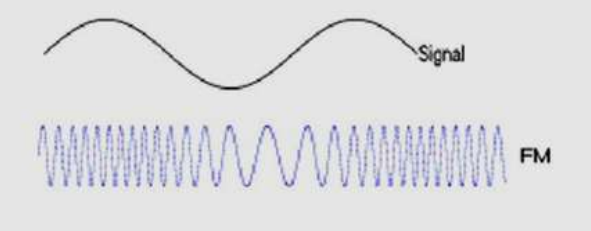
15 **List the major segments of electromagnetic spectrum and give their frequency ranges (NOV/DEC-15)** BTL1

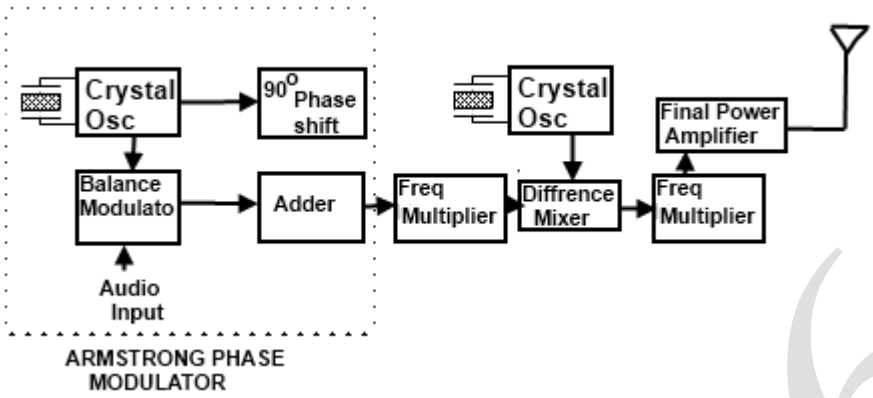
Spectrum of Electromagnetic Radiation

Region	Wavelength (Angstroms)	Wavelength (centimeters)	Frequency (Hz)	Energy (eV)
Radio	$> 10^9$	> 10	$< 3 \times 10^9$	$< 10^{-5}$
Microwave	$10^9 - 10^6$	10 - 0.01	$3 \times 10^9 - 3 \times 10^{12}$	$10^{-5} - 0.01$
Infrared	10^6 - 7000	0.01 - 7×10^{-5}	$3 \times 10^{12} - 4.3 \times 10^{14}$	0.01 - 2
Visible	7000 - 4000	$7 \times 10^{-5} - 4 \times 10^{-5}$	$4.3 \times 10^{14} - 7.5 \times 10^{14}$	2 - 3
Ultraviolet	4000 - 10	$4 \times 10^{-5} - 10^{-7}$	$7.5 \times 10^{14} - 3 \times 10^{17}$	$3 - 10^3$
X-Rays	10 - 0.1	$10^{-7} - 10^{-9}$	$3 \times 10^{17} - 3 \times 10^{19}$	$10^3 - 10^5$
Gamma Rays	< 0.1	$< 10^{-9}$	$> 3 \times 10^{19}$	$> 10^5$

16	**Design the bandwidth of FM signal if the frequency deviation of the modulator is 25kHz per Volt? (APR/MAY-15)** BTL6 mf= Δf /fm Δf= fm* mf B.W=2fm=50kHz
17	**Summarize the advantages of SSBSC modulation.** BTL2 **The advantages of SSB-SC over DSB-FC signal are :** 1. Less bandwidth requirements . This allow more number of signals to be transmitted in the same frequency range . 2. Lots of power saving . This is due to the transmission of only one sideband component .At 100% modulation, the percent power saving is 83.33% . 3. Reduced interference of noise . This is due to the reduced bandwidth . As the bandwidth increases, the amount of noise added to the signal will increase . **Disadvantages of SSB-SC Modulation** The disadvantage of SSB modulation are : 1. The generation and reception of SSB signal is complicated (which we will discuss later in our following post). 2. The SSB transmitter and receiver need to have an excellent frequency stability . A slight change in frequency will hamper the quality of transmitted and received signal . Therefore, SSB is not generally used for the transmission of good quality music. It is used for speech transmission . **Application of SSB** 1. SSB transmission is used in the applications where the power saving and low bandwidth requirements are important . 2. The application areas are land and air mobile communication, telemetry, military communications, navigation and amateur radio . Many of these applications are point to point communication application .
18	**Consider an AM signal x(t)=2cos(2πfct) +0.5cos(2πfct).cos(2πfmt). Find the modulation index used to generate the signal.** BTL5 x(t)=2cos(2πfct) +0.5cos(2πfct).cos(2πfmt). ma=Em/Ec or mf= Δf /fm
19	**Plan the bandwidth which is needed to transmit voice signal of 4kHz, use AM.** BTL6 B.W=2fm=8 kHz
20	**What is the purpose of limiter in FM receiver?** BTL4 Usually FM receivers have at least one amplification stage with a limiter. The purpose of the limiter is to provide a constant level of signal to the FMdemodulator, thus reducing the effect of signal level changes in the output.
21	**What is over, under, critical modulation?**BTL2 If m >1, has severe distortion.This condition is Over modulation. If m=1, has greatest output and condition is Critical modulation. If m< 1 , has no distortion and condition is Under modulation**.**
22	**Define deviation ratio.** It is the worst-case modulation index which is the ratio of maximum permitted frequency deviation and maximum modulating signal frequency.

	Deviation ratio = Δf(max) / fm(max)
23	**State Carson's rule for determining approximate Band Width of FM signal**. BTL2 Carson rule states that the bandwidth required to transmit an angle modulated wave as twice the sum of the peak frequency deviation and the highest modulating signal frequency**.** **Band Width = 2 [Δf+ fm(max)]Hz** **Δf = frequency deviation in Hz** **fm(max) = highest modulating signal frequency in Hz**
24	**A carrier is frequency is frequency modulated with a sinusoidal signal of 2 KHz resulting in a maximum frequency deviation of 5 KHz. Find the approximate band width of the modulated signal.** BTL5 Δf = frequencdeviation in Hz = 5 KHz fm(max) = highest modulating signal frequin Hz = 2 KHz Band Width = 2 [Δf+ fm(max)]Hz = 14 KHz
25	**In amplitude modulation system the carrier frequency is FC=100KHz. Maximum frequency of the signal is 5KHz.Determine the lower and upper side band and bandwidth of AM (April/May 2010)** BTL5 FUSB = fc+fm =105 KHz FLSB = fc-fm = 95 KHz BW=2fm = 2*5KHz = 10 KHz
26	**What are the advantages of FM over AM?** BTL2 **i)** The amplitude of FM is constant. Hence transmitter power remains constant in FM where as it varies in AM. **ii)** Since amplitude of FM is constant, the noise interference is minimum in FM. **iii)** Any noise superimposing on modulated carrier can be removed with the help of amplitude limiter. **iv)** The depth of modulation have limitation in AM. But in FM, the depth of modulation can be increased to any value. **v)** Since guard bands are provided in FM, there is less possibility of adjacent channel interference **vi)** Since space waves are used for FM, the radius of propagation is limited to line of sight (LOS). Hence it is possible to operate several independent transmitters on same frequency with minimum interference. **vii)** Since FM uses UHF and VHF ranges, the noise interference is minimum compared to AM which uses MF and HF ranges.
27	**Illustrate the Degree of modulation in AM** BTL2 In the AM the degree of modulation m is defined as the ratio of the amplitude modulation signal to said carrier signal. The volume level of an audio signal corresponding to the signal amplitude.
28	**What is the advantage and disadvantage of Angle modulation?** BTL2 **Advantages:** 1. Noise Reduction 2. Improved system fidelity 3. More effective use of power **Disadvantage:** 1. Require more Bandwidth 2. Use more complex circuits in both transmitter and receiver

29	**Draw the FM waveform? (June'13)** BTL3 
30	**Determine the modulation depth of FM system with a maximum frequency deviation of 75 KHz and the maximum modulating frequency of 10 KHz** BTL5 $mf = \Delta f / fm = 75 \times 10^3 / 10 \times 10^3 = 7.5$
31	**Define instantaneous frequency deviation.** BTL2 The instantaneous frequency deviation is the instantaneous change in the frequency of the carrier and is defined as the first derivative of the instantaneous phase deviation.
	PART * B
1	**For an AM DSBFC transmitter with an unmodulated carrier with power Pc = 100W that is modulated simultaneously by three modulating signals with coefficients of modulation m_1= 0.2, m_2= 0.4 and m_3= 0.5, Determine :** **1) Total coefficient of modulation** **2) Upper and lower sideband power** **3) Total transmitted power. (6M) (Nov 2017)** BTL 3 **Answer: Page 256-Notes** **Total Coefficient of modulation:** $M=\sqrt{m1^2 + m2^2 + m3^2}=\sqrt{0.2^2 + 0.4^2 + 0.5^2} = 0.67$ (2M) **Upper and lower side band power:** $\frac{m^2 Ec^2}{8R}$ = 22.445W (2M) **Total power**: $P_t = P_c\sqrt{1 + \frac{m^2}{2}}$= 122.445W (2M)
2	**Draw the block diagram of Armstrong indirect FM transmitter and describe its operation. (7M) (Nov 2017)** BTL 1 **Answer: Page 2.10-Chitode** **FM:** Changing carrier signal frequency(2M) **Block Diagram function:** Converts narrowband signal to wideband signal-using phase shifters and multipliers. (5M)



3 **Discuss the advantages and disadvantages of angle modulation. (6M) (Nov 2017)** BTL 2

Answer: Page-2.2-Chitode

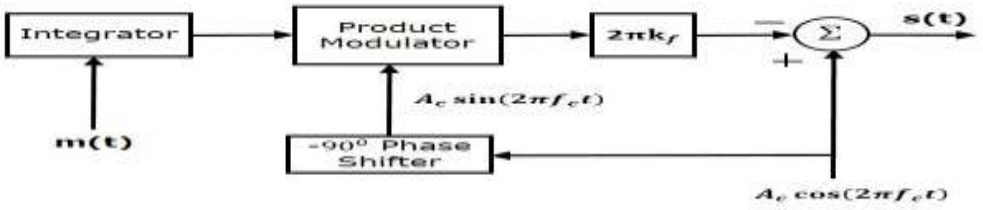
FM and PM: Changing carrier signal frequency (1M)
PM: Changing carrier signal phase(1M)
Advantages and disadvantages: (4M)

S. No	PM	FM
1	Poor fidelity.	Large bandwidth, fidelity better.
2	Infinite sidebands	Infinite sidebands
3	Noise interference more than FM	Minimum noise interference

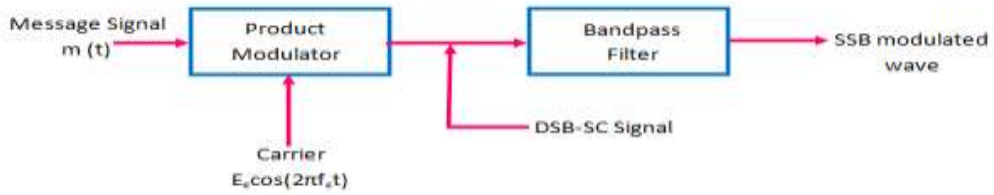
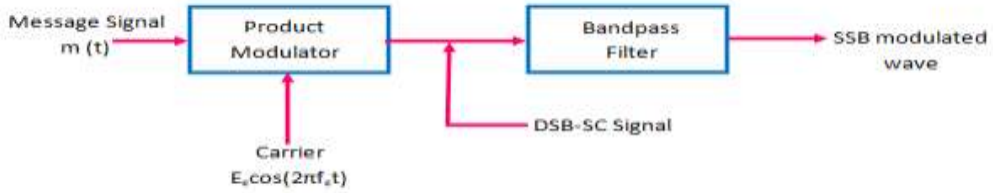
4 **Discuss about the indirect method of generating wideband FM signal. (May 2017)(13M)** BTL 2

Answer: Page-2.8-Chitode

FM : Carrier frequency varied.(2M)
Wideband FM : Better signal quality (1M)
Greater spectrum usage. (1M)
Modulation index larger than 10 (2M)
Product modulator: One input -90◦ Phase shifted carrier signal- other input integrated modulating signal. (3M)
Summer: Output added -original carrier generated - carrier generator. (2M)

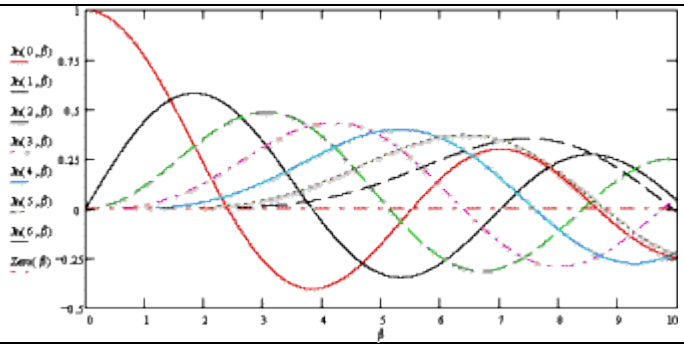
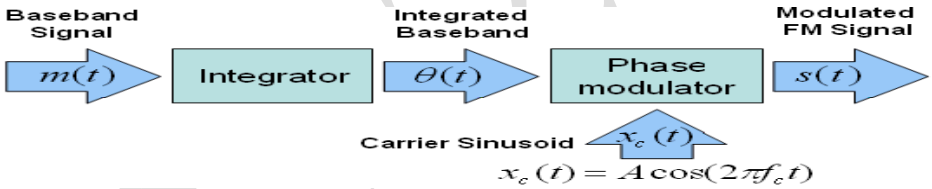


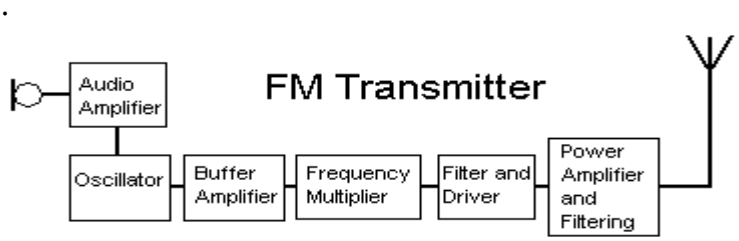
Bandwidth: BW = $2(\delta+f_m)$.(2M)

5	**Describe frequency discrimination method of generating SSB modulated wave and a method to demodulate it. What are the design issues involved in this method of generation? What is the cause and effect of phase error in demodulated signal? (13M)** **(May 2017)** BTL 4 **Answer: Page-1.74-Chitode** **SSB modulation**:One sideband transmitted- no carrier- transmit information like audio signal. (2M) **Design**: Generated by modulator - filtered usingBandpass filter. (7M)  **Issues**: Costly,Complicate, Special purpose applications. (4M)
6	**Derive the expression for the instantaneous voltage of SSB wave. (13M) (Nov 2016)** BTL 2 **Answer: Page-1.76-Chitode** **SSB modulation**: One sideband transmitted - no carrier- transmit information like audio signal (2M) **Design:**Generated by modulator -filtered using Bandpass filter. (8M) **Expression:** $e_{SSb}(t)= e_m(t)cos\omega_c t \pm sin\omega_m t sin\omega_c t$(3M) 
7	**Derive the expression for instantaneous voltage of AM wave. (7M) (May 2016)** BTL 2 **Answer: Page-1.55-Chitode** **AM** : Changing high frequency carrier amplitude (2M) **Voltage distribution :**(5M) $e_{AM} = E_c\sin \omega_c t + mE_c/2 \cos(\omega_c - \omega_m)t - mE_c/2\cos(\omega_c + \omega_m)t$
8	**Describe the relationship between the instantaneous carrier frequency and the modulating signal for FM. (7M) (May 2016)** BTL 4 **Answer: Page -2.3-Chitode** **FM:** Changing carrier frequency.(2M) **Equations of FM**: $f_i(t) = f_c + k_f E_m \cos(2\pi f_m t)$.(3M) **Frequency deviation:** $\Delta f = k_f E_m$.(2M)

9	**With the help of mathematical expressions explain about Amplitude modulation and its generation. (13M) (May 2015) (May 2011)** BTL 2 **Answer: Page-1.47-Chitode** **AM:** Changing high frequency carrier amplitude(2M) **Voltage distribution of AM:**(7M) $e_{AM} = E_c \sin \omega_c t + mE_c/2 \cos(\omega_c - \omega_m)t - mE_c/2\cos(\omega_c + \omega_m)t$ **Power distribution**: (4M) P_t= Carrier power + Power in USB + Power in LSB $= E_{carr}^2 / R + E_{USB}^2 / R + E_{LSB}^2 / R$
10	**Draw the phasor diagram of a wideband FM and explain about the bandwidth of FM signal. (7M) (May 2015)** BTL 4 **Answer: Page-2.10-Chitode** **FM** : Carrier frequency varied (2M) **Wideband FM** : Better signal quality (1M) Greater spectrum usage. Modulation index larger than 10 (1M) **Phasor diagram**(1M) **Bandwidth** : BW = 2 [$\Delta f + f_m$] (2M)
11	**Explain the difference between phase modulation and frequency modulation. (7M) (May 2015) (Nov 2011) (May 2011)** BTL 4 **Answer: Page -2.18-Chitode** **FM and PM**:Carrierfrequency and phase varied –according to- amplitude - modulating signal. (2M) **Noise:** PM-Noise immunity better than AM but less than FM. (1M) FM- Better Noise immunity (1M) **Equations :** PM-$E_{PM}(t) = E_c \sin[\omega_c t + M_{PM} \cos\omega_m t]$ (1M) FM- $f_i(t) = f_c + k_f E_m \cos(2\pi f_m t)$ (1M) **Phase and frequency deviation:** (1M) PM-$\Delta\varphi$: phase deviation FM- Δf : frequency deviation
12	**In modulation by several sine waves simultaneously, the bandwidth of AM requires twice the highest modulation frequency. Prove this concept using appropriate expressions. (7M) (Nov 2014)**BTL 4 **Answer: Page-1.49-Chitode** **AM:** Changing high frequency carrier amplitude (2M) **Voltage distribution of AM:** (3M) $e_{AM} = E_c \sin \omega_c t + mE_c/2 \cos(\omega_c - \omega_m)t - mE_c/2\cos(\omega_c + \omega_m)t$ **Bandwidth:** B=2f_m (2M)

13	**Calculate the percentage power saving when the carrier and one of the sidebands are suppressed in an AM wave modulated to a depth of 100 percent and 50 percent. (6M) (Nov 2014)** BTL 3 **Answer: Page-251-Notes** **Total power:** $P_t = (1 + \frac{m^2}{2})$ = 5001 W (M=100%) (3M) = 1251 W (M=50%) (3M)
14	**Describe frequency modulation and phase modulation and their inter-relationship. (7M) (Nov 2014)** **Answer: Page-2.18-Chitode** BTL 2 **FM and PM** :Carrier frequency and phase varied – inpropotion- amplitude - modulating signal. (2M) **Noise:** PM-Noise immunity better than AM but less than FM. (1M) FM- Better Noise immunity (1M) **Equations :** PM-$E_{PM}(t) = E_c \sin[\omega_c t + M_{PM} \cos\omega_m t]$ (1M) FM- $f_i(t) = f_c + k_f E_m \cos(2\pi f_m t)$ (1M) **Phase and frequency deviation:** (1M) PM-$\Delta\varphi$: phase deviation FM- Δf : frequency deviation
15	**Derive equations for AM voltage and power distribution. (13M) (May 2014)** BTL 2 **Answer: Page -1.47-Chitode** **AM:** Changing high frequency carrier amplitude (2M) **Voltage distribution of AM:** (5M) $e_{AM} = E_c \sin \omega_c t + mE_c/2 \cos(\omega_c - \omega_m)t - mE_c/2 \cos(\omega_c + \omega_m)t$ **Power distribution**: (6M) P_t= Carrier power + Power in USB + Power in LSB $= E_{carr}^2 / R + E_{USB}^2 / R + E_{LSB}^2 / R$
16	**Explain the frequency analysis of angle modulated waves. (7M) (May 2014) (Nov 2011)** BTL 2 **Answer: Page-2.6-Chitode** **FM:**Carrier frequency varied – inproportion – amplitude - modulating signal.(2M) **Frequency analysis of angle modulated waves:** Single frequency modulating signal produces infinite sidebands**.** Bandwidth infinity. Frequency components complexly related than AM. $V_{pm}(t) = V_c \sum_{n=-\infty}^{\infty} J_n(m) \cos(\omega_c t + n\,\omega_m t + \frac{n\pi}{2})$(5M)

	
17	**Write a note on frequency deviation of FM wave. (7M) (May 2014)** BTL 2 **Answer: Page-2.4-Chitode** **FM**: Changing carrier frequency - proportion- message signal (2M) **Equations:** $f_i(t) = f_c + k_f E_m \cos(2\pi f_m t)$ (3M) **Frequency deviation:** $\Delta f = k_f E_m$(2M)
18	**Describe the generation of FM. (7M) (May 2014)** BTL 2 **Answer: Page-2.3-Chitode** **FM** :Changing carrier frequency - proportion - message signal (2M) **Equation:** $f_i(t) = f_c + k_f E_m \cos(2\pi f_m t)$ (2M) **Generation:** Obtained from PM through Integrator (3M) 
19	**What is the need for modulation? (6M) (May 2013)** BTL 2 **Answer: Page-1.4-Chitode** **Reduction in height of antennae:** $h=\frac{\lambda}{4}$(1M) **Multiplexing:** Transmitting more signals -single channel(1M) **Avoids radiation:** Design Antenna - high frequency (1M) **Improves Quality of Reception:** Reduced noise effect (1M) **Range of Communication:** Long distance (1M) **Mixing of signals:** Avoided- modulating with different frequency (1M)
20	**Explain with block diagram of a FM transmitter with direct modulation. (7M) (May 2013)**BTL 2 **Answer: Page -1.24-Chitode** **FM:** Changing -carrier signal frequency.(1M) **Equations:** $f_i(t) = f_c + k_f E_m \cos(2\pi f_m t)$(2M) **Generation**: Message signal multiplied -carrier generated-oscillator- filtered -amplified. (4M)



21 **Discuss about spectral characteristics of FM signal. (7M) (May 2013)** BTL 2
Answer: Page-2.6-Chitode

FM: Changing -carrier signal frequency –inproportion - message signal.(2M)
Equations: $f_i(t) = f_c + k_f E_m \cos(2\pi f_m t)$ (1M)
Spectral characteristics: Single frequency - modulating signal - infinite sidebands**.** Infinite bandwidth. Frequency components complexly related than AM(4M)

22 **Derive for carrier power and transmitter power in AM in terms of modulation index. (7M) (Nov 2012)** BTL 2
Answer: Page-1.52-Chitode

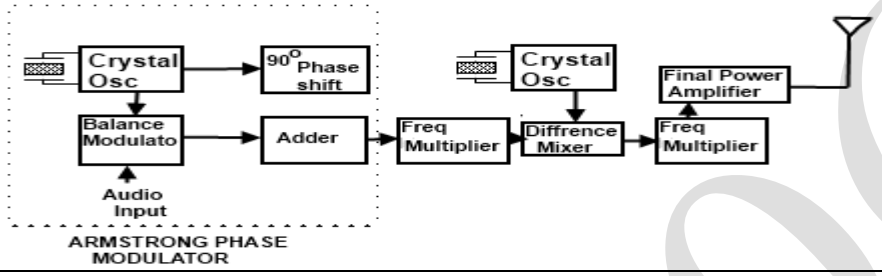
AM: Changing - amplitude - high frequency carrier signal. (2M)
Power distribution: P_t= Carrier power + Power in USB + Power in LSB
$= E_{carr}^2 / R + E_{USB}^2 / R + E_{LSB}^2 / R$ (3M)
Modulation Index: $m = E_m / E_c$(2M)

23 **Differentiate between AM and FM. (6M) (Nov 2012)** BTL 4
Answer: Page 1.47 and 2.3-Chitode

AM	FM
Carrier Amplitude varied (1M)	**Carrier frequency varied (1M)**
Amplitude varies. (1M)	**Amplitude remains constant.**
Very much affected by noise. (1M)	**High noise immunity**
Limited Bandwidth (1M)	**Large bandwidth**
535 – 1705 KHz range. (1M)	**88 – 108 MHz range**

24 **Define FM and PM modulation with their equations. Describe the generation of FM wave using Armstrong method. (7M) (May 2012)** BTL 2
Answer: Page-2.2-Chitode

FM and PM: Carrier frequency and phase varied - according - modulating signal amplitude. (1M)
Noise:(1M)
PM-Noise immunity better than AM - less than FM.
FM- Better Noise immunity

	Equations : (1M) PM-$E_{PM}(t) = E_c \sin[\omega_c t + M_{PM} \cos\omega_m t]$ FM- $f_i(t) = f_c + k_f E_m \cos(2\pi f_m t)$ **Block diagram of amstrong FM**: Converts narrowband to wideband signal - phase shifters and multipliers(4M) 
25	**Explain the principles of amplitude modulation. (7M) (Nov 2011)** BTL 2 **Answer: Page-1.47-Chitode** **AM**: Changing high frequency carrier signal amplitude (2M) **Power distribution:** P_t= Carrier power + Power in USB + Power in LSB $= E_{carr}^2 / R + E_{USB}^2 / R + E_{LSB}^2 / R$ (4M) **Modulation Index:** $m = E_m / E_c$ (1M)
26	**Explain the bandwidth requirement for FM and define carson's rule. (6M) (May 2011)** BTL 2 **Answer: Page-2.7-Chitode** **FM** - Changing carrier signal frequency -inproportion - message signal (4M) **Bandwidth**: $BW = 2[\Delta f + f_m]$ (2M)
	PART*C
1	**For an envelope with +Vmax=30Vp and +Vmin=+10Vp, determine.** **1) Unmodulated carrier amplitude. (3M)** **2) Modulated carrier amplitude. (3M)** **3) Peak change in the amplitude of the envelope. (3M)** **4) Modulation coefficient. (3M)** **5) Percent modulation. (3M)** BTL 3 **Answer: Page-2.72-Sakthidasan** **Unmodulated carrier:** $E_C = \frac{1}{2}$(Vmax +Vmin)=20V (3M) **Modulated carrier amplitude:** $E_C(Modulated) = E_C(Unmodulated)$=20V(3M) **Change in output:** $E_m = \frac{1}{2}$(Vmax -Vmin)=10V (3M) **Modulation Coefficient:** m=$\frac{E_m}{E_c}$=0.5(3M) **Percent Modulation:** M=m*100%=50%(3M)
2	**An audio frequency signal 10 sin(2π500t) is used to amplitude modulate a carrier of 50sin($2\pi 10^5 t$). *calculate*** **1) Modulation index (3M)**

	2) Sideband frequencies (3M) **3) Amplitude of each sideband frequencies (3M)** **4) Bandwidth (3M)** **5) Total power dissipated to load of 600Ω .(3M)** BTL 3 **Answer: Page-255-Notes** Vm=10 sin (2π500t)=Vm sin(2πf_mt)(1M) Vm=10V, f_m =500Hz Vc= 50sin(2π$10^5 t$)= Vc sin(2πf_ct) (1M) Vc=50V, $f_c = 10^5$ **Modulation Index:** m=$\frac{E_m}{E_c}$=0.2(1M) **Sideband frequencies:** $f_{lsb} = f_c - f_m$= 99.5 KHz (3M) $f_{usb} = f_c + f_m$= 100.5 KHz **Amplitude of sidebands:** $E_{usf} = E_{lsf} = \frac{V_{max}-V_{min}}{4} = \frac{2E_m}{4}$ = 5V (3M) **Bandwidth:** $B = 2f_m$ = 1KHz (3M) **Total Power:** Pt = Pc(1+$\frac{m^2}{2}$)= 2.125W. (3M)
3	**Describe frequency modulation and phase modulation and their inter-relationship. (15M) (Nov 2014)** BTL 2 **Answer: Page 2.18-Chitode** **FM and PM**: Frequency and phase of the carrier varied according to amplitude of modulating signal. (3M) **Noise:** PM-Noise immunity better than AM but less than FM. (2M) FM- Better Noise immunity. (2M) **Equations :** PM-E_{PM} (t) = E_c sin[ω_c t + M_{PM} cosω_mt] (2M) FM- f_i (t) = f_c +$k_f E_m$cos (2π f_mt) (2M) **Phase and frequency deviation:** PM-Δφ : phase deviation.(2M) FM- Δf : frequency deviation. (2M)

UNIT II - PULSE AND DATA COMMUNICATION

Pulse Communication: Pulse Amplitude Modulation (PAM) – Pulse Time Modulation (PTM) – Pulse code Modulation (PCM) - Comparison of various Pulse Communication System (PAM – PTM – PCM). Data Communication: History of Data Communication - Standards Organizations for Data Communication- Data Communication Circuits - Data Communication Codes - Data communication Hardware - serial and parallel interfaces.

PART * A

Q.1. Name the standards organizations for data communication? MAY/JUNE-16 BTL2

- International Standard Organization (ISO) ...
- International Telecommunications Union-Telecommunication Sector (ITU-T) ...
- Institute of Electrical and Electronics Engineers (IEEE) ...
- American National Standards Institute (ANSI) ...
- Electronics Industry Association (EIA)

Q.2. List out all data communication codes BTL1

- ASCII (American Standard Code for Information Interchange)
- The Baudot Code

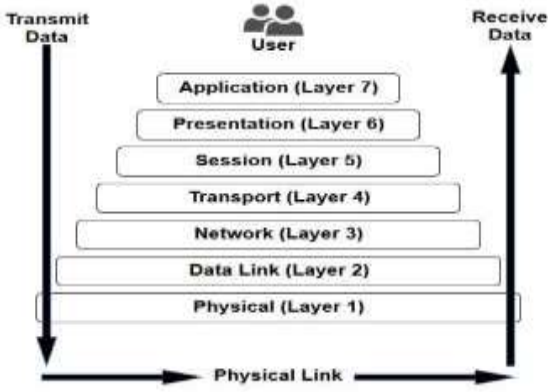
Q.3 Distinguish between half duplex and full duplex transmission. BTL4

Basis for Comparison	Simplex	Half Duplex	Full Duplex
Direction of Communication	Unidirectional	Two-directional, one at a time	Two-directional, simultaneously
Send / Receive	Sender can only send data.	Sender can send and receive data, but one a time.	Sender can send and receive data simultaneously.
Performance	Least performing mode of transmission.	Better than Simplex	Most performing mode of transmission.
Example	Keyboard and monitor	Walkie-talkie	Telephone

Q.4 What is data modem? BTL2

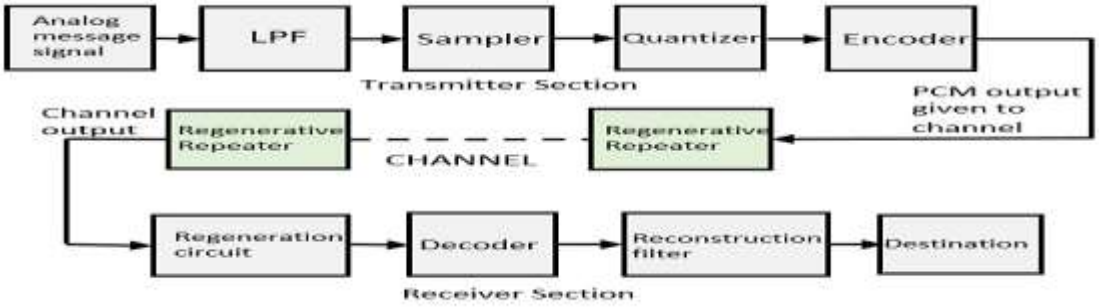
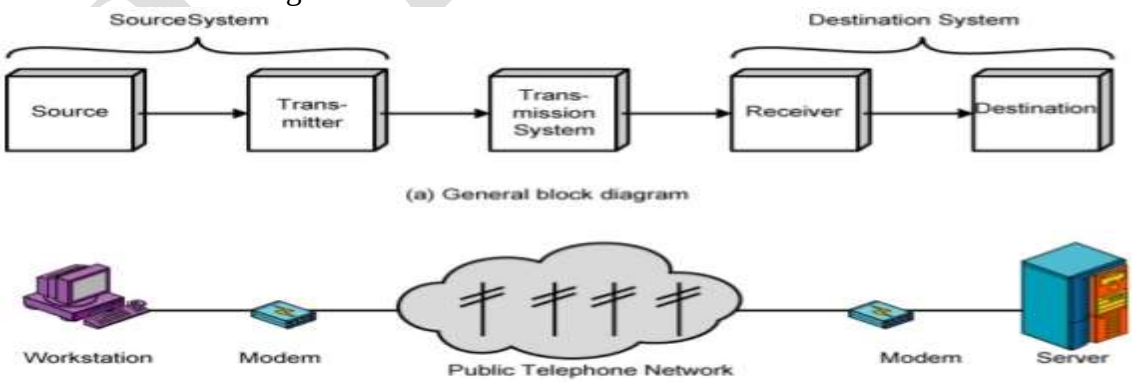
A modem is a device or program that enables a computer to transmit data over, for example, telephone or cable lines. Computer information is stored digitally, whereas information transmitted over telephone lines is transmitted in the form of analog waves. A modem converts between these two forms.

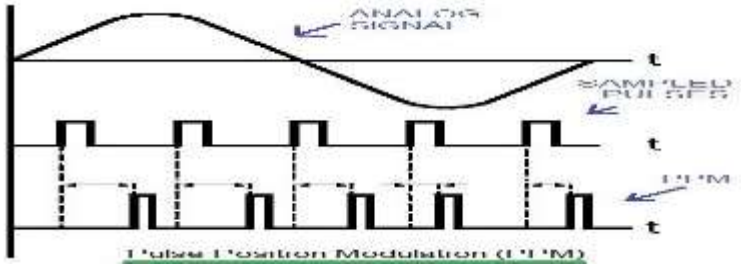
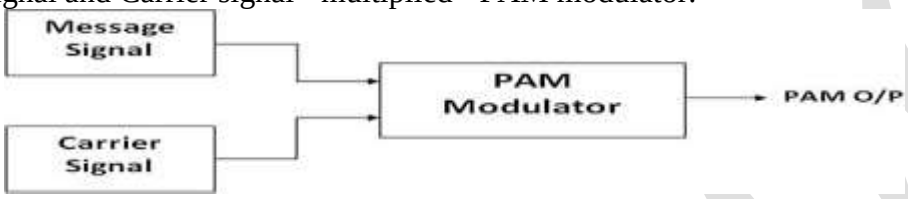
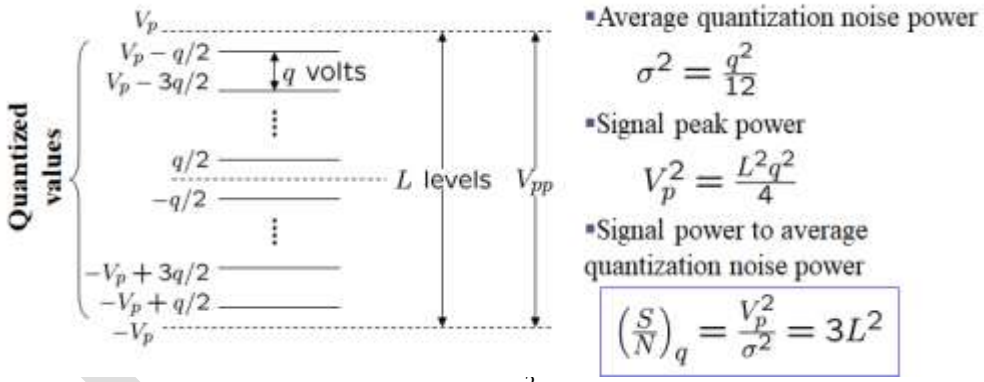
Q.5 List out the layer presented in ISO-OSI reference model. BTL2 **-sevenlayers of OSI**

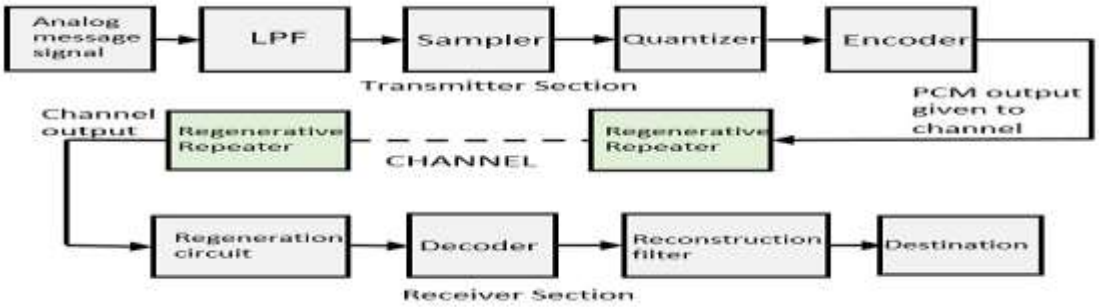


6	**Define USRT, USART.** BTL1 USART vs UART: Know the difference. ... A USART -- a Universal Synchronous/Asynchronous Receiver/Transmitter -- is a microcontroller peripheral that converts incoming and outgoing bytes of data into a serial bit stream. Hmm. Thedefinition of a USART is identical to that of a UART, but with "synchronous" added to the term.
7	**Determine the odd and even parity bits for the ASCII character R whose Hex code is 52.** BTL5 The hex code for the ASCII character R is 52, which is P1010010 in binary, where P designates the parity bit.•For odd parity, the parity bit is a 0 because 52 hex contains three logic 1s, which is an odd number. Therefore, the odd-parity bit sequence for the ASCII character R is 01010010.•For even parity, the parity bit is 1, making the total number of logic 1s in the eight-bit sequence four, which is an even number. Therefore, the even-parity bit sequence for the ASCII character R is 11010010.
8	**State the sampling theorem for band limited signal of finite energy. (NOV/DEC-15)** BTL3 **1)** A band limited signal of finite energy , which has no frequency components higher than W hertz , is completely described by specifying the values of the signal at instants of time separated by 1/2W seconds and **2**) A band limited signal of finite energy, which has no frequency components higher than W hertz , may be completely recovered from the knowledge of its samples taken at the rate of 2W samples per second.The first part of above statement tells about sampling of the signal and second part tells about reconstruction of the signal. Above statement can be combined and stated alternately as follows :Acontinuous time signal can be completely represented ints samples and recovered back ifthe sampling frequency is twice of the highest frequency content of the signal i.e., **fs≥2W** Here fs is the sampling frequency
9	**Prepare the Nyquist rate for analog input frequency of a) 4kHz b) 10kHz** BTL6 **a)** $2f_m = 2*4=8$ kHz **b)** $2f_m = 2*10=20$ kHz
10	**Define Aliasing and Aperture effect.** BTL1 **Effects of aliasing:** i)since high and low frequencies interfere with each other , distortion is generated. ii)The data is lost and it cannot be recovered. **Different ways to avoid aliasing :** Aliasing can be avoided by two methods i)sampling rate fs≥2Wfs≥2W ii)Strictly bandlimit the signal to 'W' **Aperture effect** Well, this aperture effect takes place in Flat Top Sampling. Aperture error is the difference between the actual value of the input signal, and the flat-topped sample value. ... It would appear that reducing the sampling aperture to near zero would eliminate this form of distortion.
11	**Infer about Quantization process.** BTL5 Quantization, in mathematics and digital signal processing, is the process of mapping input values from a large set (often a continuous set) to output values in a (countable) smaller set, often with a finite number of elements. Rounding and truncation are typical examples of quantization processes.
12	**Define DTE, DCE.** BTL1 A data circuit-terminating equipment (DCE) is a device that sits between the data terminal equipment (DTE) and a data transmission circuit. It is also called data communication(s) equipment and data carrier equipment. Usually, the DTE device is the terminal (or computer), and the DCE is a modem.

13	**Give any two function of UART.** BTL2 A universal asynchronous receiver/transmitter (UART) is a block of circuitry responsible for implementing serial communication. Essentially, the UART acts as an intermediary between parallel and serial interfaces. ... UARTs do exist as stand-alone ICs, but they're more commonly found inside microcontrollers
14	**List the advantages of PCM. (APR/MAY-17)** BTL1 • The PCM (pulse code modulation) convenient for long distance communication. • It has a higher transmitter efficiency. • It has a higher noise immunity.
15	**Define Pulse time modulation. (MAY/JUNE-16)** BTL1 Definition of pulse time modulation. : modulation of the time intervals between successive pulses of constant duration and amplitude in accordance with a signal specifically : a system of multiplex high-frequency transmission using this method ofmodulation**.**
16	**State the need for companding in a PCM system. (APR/MAY-15)** BTL3 Techopedia explains Companding. ... For digital audio signals, companding is used in pulse code modulation (PCM). The process involves decreasing the number of bits used to record the strongest (loudest) signals. In the digital file format,companding improves the signal-to-noise ratio at reduced bit rates.
17	**Illustrate the regenerative repeaters** BTL3 Regenerative Repeater. in telegraphy, a device that receives current pulses and corrects their shape and duration and further transmits them. Regenerative repeaters are used during many pulse retransmissions in order to improve the quality of data transmission.
18	**Mention how PPM is derived from PWM (APR/MAY-15)** BTL4 • To generate pulse position modulation, the PWM pulses obtained at the output of the comparator are used as the trigger input as the trigger input to as a monostablemultivibrator. • The Monostable is triggered on negative falling edge of PWM.The output of monostable goes high.The voltage remains high for the fixed period then goes low.
19	**Why do we encounter aperture effect in PAM? How will you rectify it? (NOV/DEC-15)** BTL4 The distortion caused by the use of PAM to transmit an analog signal is called theaperture affect. ... The equalizer has the effect of decreasing the in-band loss of the filter as the frequency increases in such a manner to compensate for the aperture effect.
20	**What are the advantages of PWM?** BTL1 So the advantage is greater efficiency, less heat dissipation needed and higher power output for the same type of regulator component. Here , Application of Pulse Width Modulation (PWM) to control speed of motor . You can do similar things like dimming LED etc .
	PART*B
1	**Describe the following data communications codes: Baudot, ASCII and EBCDIC. (6M) (Nov 2017) (May 2016)** BTL 2 **Answer: Page-150-W.Tomasi** **Data communication codes:**Represent characters, symbols- such as letters, digits,punctuation marks. (1M) **Baudot**: Five character code - teletype equipment (1M)

	ASCII: Uses seven bits- encode each character. (2M) **EBCDIC**: 8 bit code used - IBM equipment (2M)
2	**Explain the generation of PCM signal with a block diagram. (7M) (Nov 2017)** BTL 2 **Answer: Page-279-W.Tomasi** **PCM**: Digital representation - analog signals – takes analog signal samples - regular intervals. (1M) **Block diagram:** **Sampler**: Converting message signal to discrete signal.(2M) **Quantizer**: Rounding off the amplitudes - nearest level.(2M) **Encoder**: Coding quantized signal - efficient transmission.(2M) **Regenerative Repeaters**: Long distance communication.(1M) 
3	**Explain the working of a two station data communication circuit with ablock diagram. (7M) (Nov 2017)** BTL 2 **Answer: Page-171-W.Tomasi** **Data communication**: Using computing and communication technologies - transfer data from one place to another - vice versa.(2M) **Block diagram**: (5M) **Source:** Generate data **Transmitter**:Efficient transmission. **Channel:**Medium - transfer signal. **Destination:** Receives signal. 
4	**Describe the generation and demodulation of PPM signalwaveforms. (6M) (Nov 2017)** BTL 2 **Answer: Page-279-W.Tomasi** **PPM with waveforms:**Signal modulation- *M* message bits - encoded - transmitting single pulse -

	one possible required time shifts. (2M)  **Block diagram**: (4M) Message signal and Carrier signal - multiplied - PAM modulator. 
5	**Explain quantization process in detail and derive the expression output signal to noise ratio in uniform quantizer. (13M) (May 2017)** BTL 2 **Answer: Page-293-W.Tomasi** **Quantization**: Discretization in amplitude domain. (2M) **Types:** Uniform- Uniform step size, Non uniform = unequal step size. (3M) **Diagram**:(3M)  **PCM SNR**: Derivation with Uniform Quantizer =$3L^2$.(5M)
6	**Explain about various operations performed in transmitter and receiver of PCM system. (7M) (May 2017)** BTL 2 **Answer: Page-279-W.Tomasi** **PCM:** Digital representation - analog signals - samples – regular intervals - sampled signal – quantization - encoding. (2M) **Block diagram**: (5M) **Sampler**: Converting message signal to discrete signal. **Quantizer**: Rounding off amplitudes - nearest level. **Encoder**: Coding quantized signal - efficient transmission. **Regenerative Repeaters**: Long distance communication.



.

7 **Explain the working of two station data communication circuit with a block diagram. (7M) (Nov 2016) (May 2016) (Nov 2014)** BTL 2

Answer: Page-173-W.Tomasi

Data communication: Using computing and communication technologies - transfer data from one place to another - vice versa.(2M)

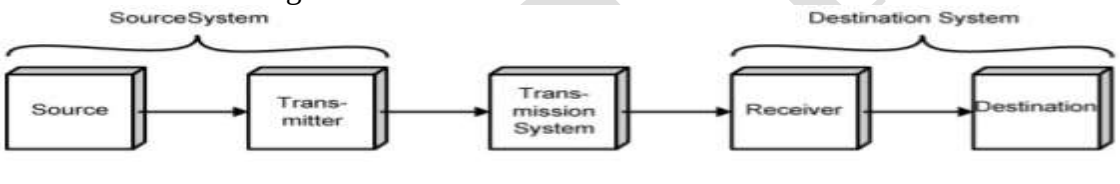
Block diagram: (5M)

Source: Generate data

Transmitter:Efficient transmission.

Channel: Medium - transfer signal.

Destination: Receives signal



(a) General block diagram

8 **Discuss the various data communication codes and its significance. (7M) (Nov 2016) (Nov 2014)** BTL 4

Answer: Page-150-W.Tomasi

Data communication codes: Represent characters, symbols - letters, digits and punctuation marks. (1M)

Baudot: Five character code - teletype equipment (1M)

ASCII: Uses seven bits - encode each character. (2M)

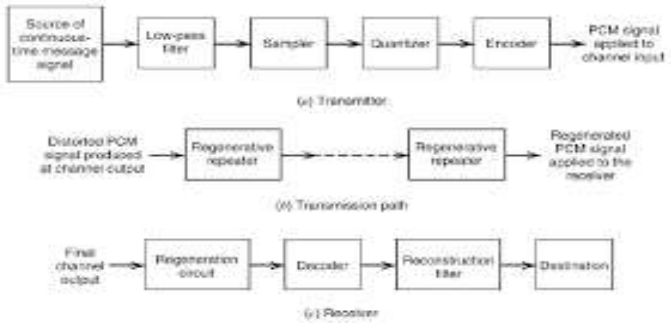
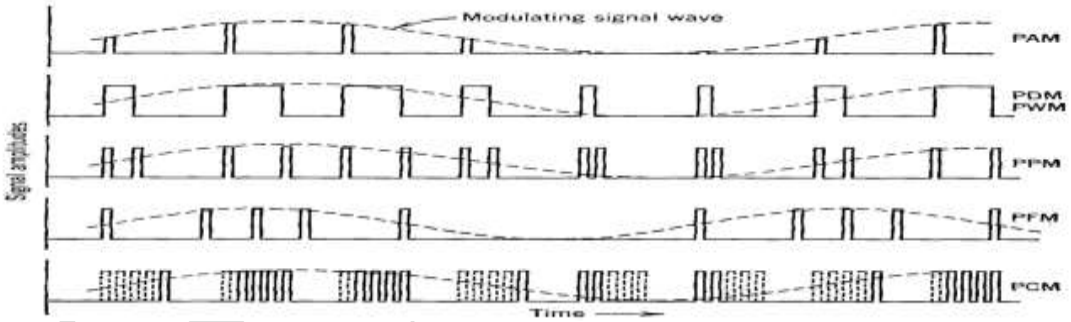
EBCDIC: 8 bit code used - IBM equipment.(1M)

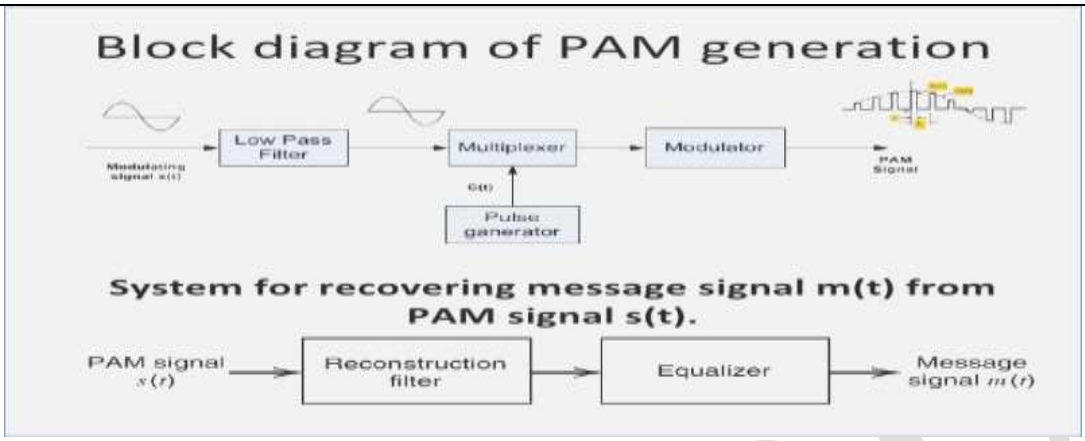
Bar code: Series of vertical bars - separated - vertical white spaces.(2M)

9 **Explain the generationofPCM signal with block diagram. (7M) (Nov 2016)(May 2013)(Nov 2012) (Nov 2011)** BTL 2

Answer: Page-279-W.Tomasi

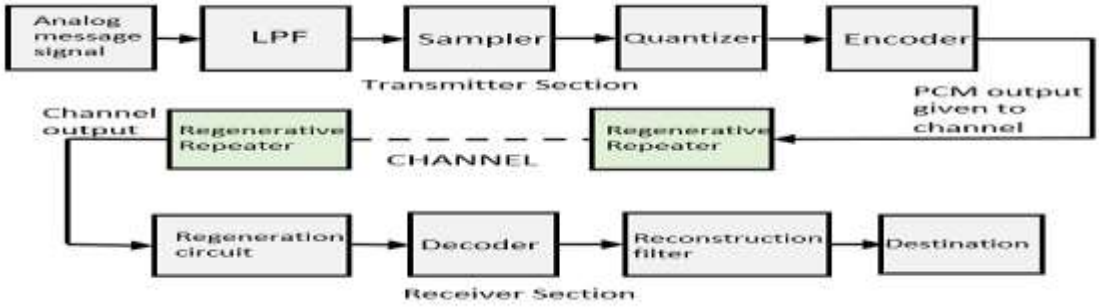
PCM: Digital representation - analog signal takes samples - analog signal at regular intervals. (1M)

	Block diagram: (6M) **Sampler**: Converting message signal to discrete signal. **Quantizer**: Rounding off amplitudes - nearest level. **Encoder**: Coding quantized signal - efficient transmission. **Regenerative Repeaters**: Long distance communication. 
11	**Compare the various pulse analog modulation techniques. (7M) (May 2016)** BTL 4 **Answer: Page-278-W.Tomasi** **Types of pulse modulation**: PAM, PPM, PWM, PTM (2M) **Explanation:**(4M) **PAM**: amplitudes of carrier pulses varied -according to - message signal sample. **PPM**: position of carrier pulses varied - according to - message signal sample. **PWM**: width of carrier pulses varied - according to- message signal sample. **PTM**: duration of carrier pulses varied - according to - message signal sample. **Diagram:**(1M) 
12	**Discuss the generation of PAM and its demodulation. (7M) (May 2015)** BTL 2 **Answer: Page-277-W.Tomasi** **PAM:**Amplitudes of carrier pulses varied - according to - message signal sample.(2M) **Block diagram**: (1M) **Explanation**: (4M) **Modulator**: Message signal and Carrier signal gets multiplied. Output- PAM. **Reconstruction filter**: PAM signal - passed through - reconstruction filter – original signal recovered.



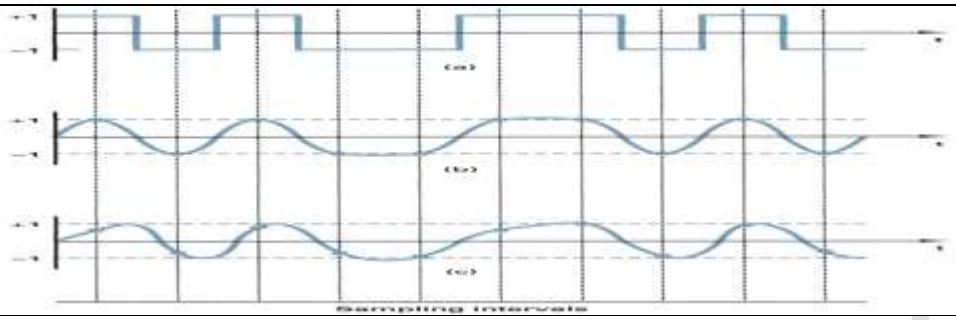
15 **Explain in detail the concept of PCM. (13M) (Nov 2014) (May 2015) (Nov 2011)** BTL 2
Answer: Page-279-W.Tomasi

PCM: Digital representation - analog signals - samples – regular intervals - sampled signal – quantization - encoding. (2M)
Block diagram: (4M)
Explanation:(7M)
Sampler: Converting message signal to discrete signal
Quantizer: Rounding off amplitudes - nearest level
Encoder: Coding quantized signal - efficient transmission
Regenerative Repeaters: Long distance communication
Decoder: Decoding PCM data to PAM
Reconstruction Filter: PAM signal – original signal



16 **What is inter symbol interference? Explain. (6M)(May 2014) (May 2013) (Nov 2012) (May 2011)** BTL 2
Answer: Page-314-W.Tomasi

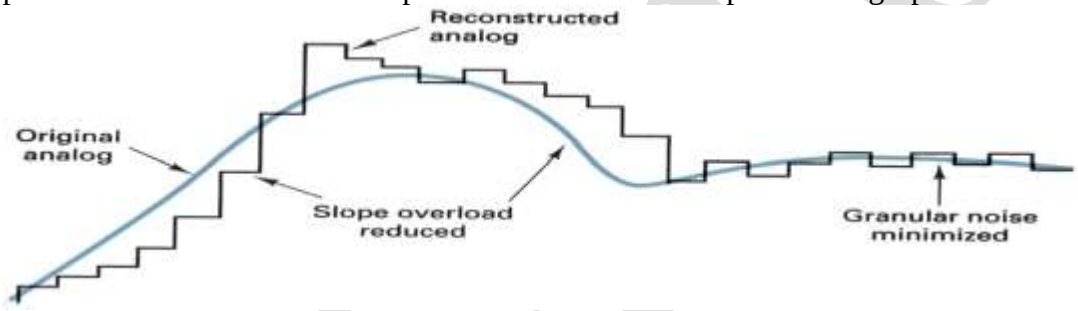
ISI: Maximum value cannot be attained at center. Overlapping of several pulses- Ringing tails - interfering with major pulse lobe. (2M)
Explanation:(4M)
Error introduced - decision making device - receiver output. Causes crosstalk between channels.



17 **Explain the principle operation of adaptive delta modulation. (13M) (May 2014) (Nov 2011)** BTL 2

Answer: Page-311-W.Tomasi

ADPCM:DAC step size - automatically varied- depending on amplitude of analog input. (2M)
Waveform: (2M)
Explanations: (9M)
ADPCM introduced - To overcome slope overload distortion and granular noise.
Output of transmitter indicates -slope of DAC - less than slope of analog - positive or negative direction.



18 **Explain in detail about quantization error. (7M) (Nov 2013) (May 2011)** BTL 2

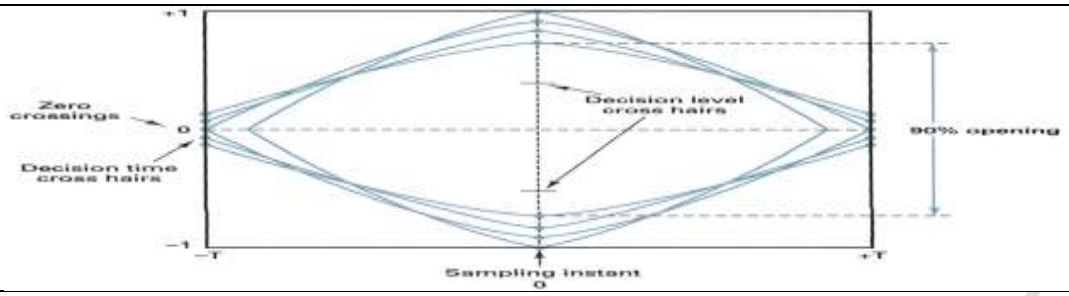
Answer: Page-293-W.Tomasi

Quantization:Converting infinite number of possibilities to finite number of conditions. Rounding off amplitudes. (2M)
Quantization error or distortion: Average signal power-to-average noise power ratio - caused by digitizing analog sample. (5M)

19 **Explain in detail about ISI and eye diagram. (7M) (Nov 2013) (May 2013) (May 2011)** BTL 2

Answer: Page-314-W.Tomasi

ISI: Maximum value cannot be attained at centre. Overlapping of several pulses- Ringing tails - interfering with major pulse lobe.
Error introduced - decision making device - receiver output. Causes crosstalk between channels.(4M)
Eye pattern: Superimposing all waveform combinations over adjacent signalling intervals. (3M)



20 **What is meant by companding? (Nov 2011) Explain the concept of analog companding.(13M) (Nov 2013) (May 2011)** BTL 2

Answer: Page-302-W.Tomasi

Companding: Combination of compression andexpansion. Compression at transmitter- expansion at receiver. (3M)

Block diagram: (2M)

Explanation:(8M)

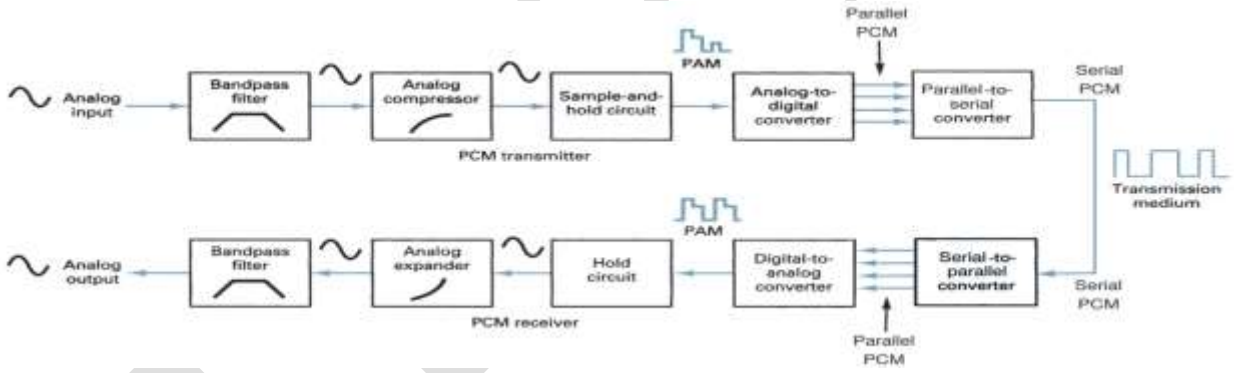
AnalogCompanding: Compression occurs before sampling. Expansion takes place after PAM signal is obtained.

BPF: Filters unwanted signal.

Sample and hold circuit: Analog signal converted to PAM.

ADC: Convert PAM signal to PCM.

Parallel to serial Converter: Parallel data converted to serial data.



PART*C

1 **Explain the types of companding in detail. (15M)** BTL 2

Answer: Page-302-W.Tomasi

Companding:Combination of compressionand expansion.
Compression of higher amplitude analog signals prior to transmission.
Expansion at receiver.
Improving dynamic range of communicationssystem. (3M)

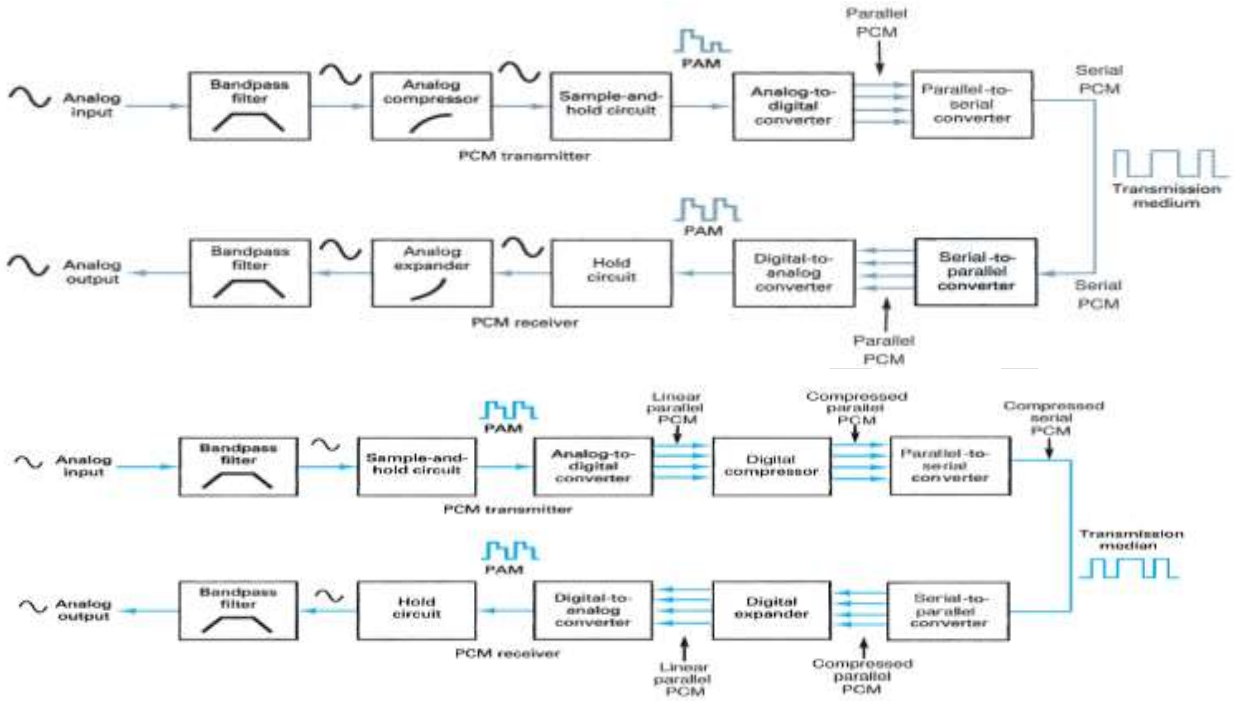
Block diagram: (2M)

AnalogCompanding: Compression occurs before sampling. Expansion takes place after PAM signal is obtained. (5M)

BPF: Filters unwanted signal.

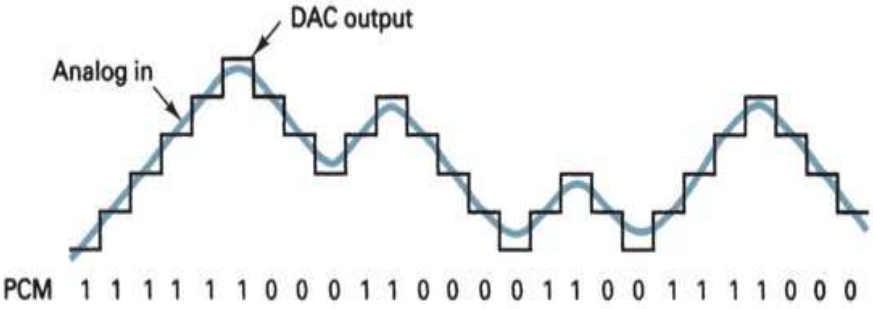
Sample and hold circuit: Analog signal converted to PAM.

ADC: Convert PAM signal to PCM.
Parallel to serial Converter: Parallel data converted to serial data.
Digital Companding: Compression takes place after sampling. Expansion takes place before the signal converted into PAM. (5M)



2 **Describe delta modulation systems. What are the limitations? How can they be overcome? (15M) (Nov 2013)(May 2013) (May 2012)** BTL 4
Answer: Page-309-W.Tomasi

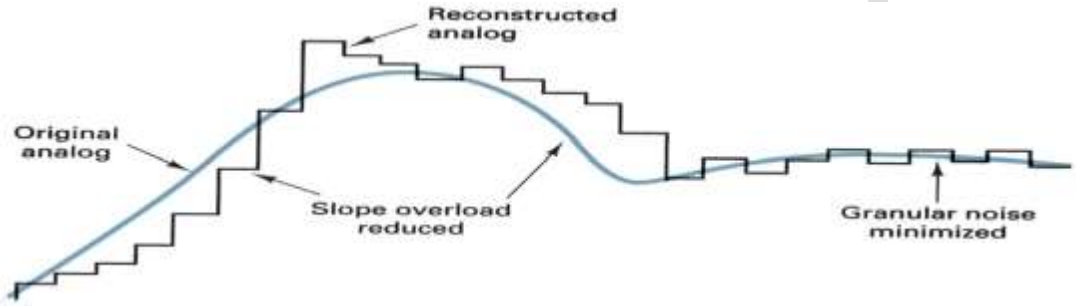
Delta modulation: up–down counter - incremented or decremented –based upon 1 or 0. (2M)



Limitations: Slope oveload distrortion and granular noise. (7M)
Slope oveload distortion: Analog input changes at faster rate than DAC can maintain.
Granular noise: Analog input - constant amplitude, the reconstructed signal has variations- were not in original signal.
Methods to overcome DM: ADPCM(6M)
Output of transmitter indicates slope of DAC - less than slope of analog - positive or negative direction.

3 **Explain the principle operation of adaptive delta modulation. (15M) (May 2014)** BTL 2
Answer: Page-311-W.Tomasi

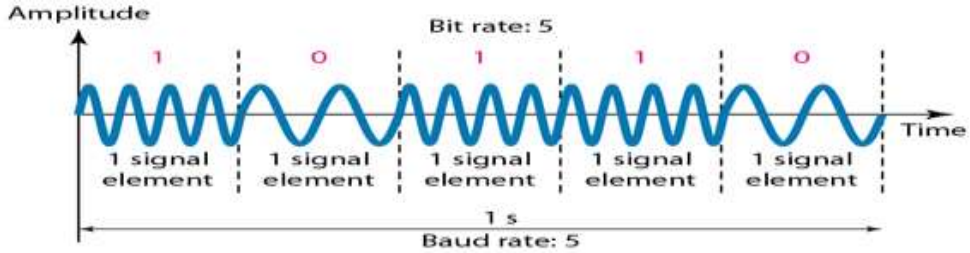
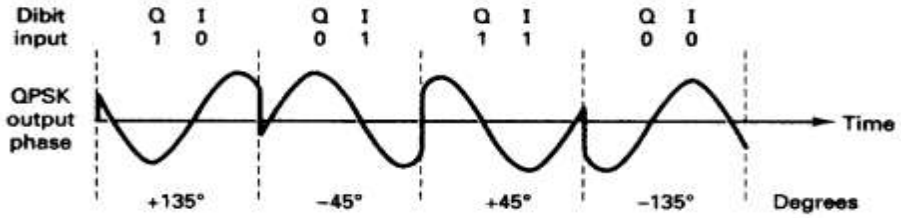
ADPCM:DAC step size - automatically varied- depending on amplitude of analog input. (3M)
Waveform: (2M)
Explanations: (10M)
ADPCM introduced - To overcome slope overload distortion and granular noise.
Output of transmitter indicates slope of DAC - less than slope of analog - positive or negative direction.

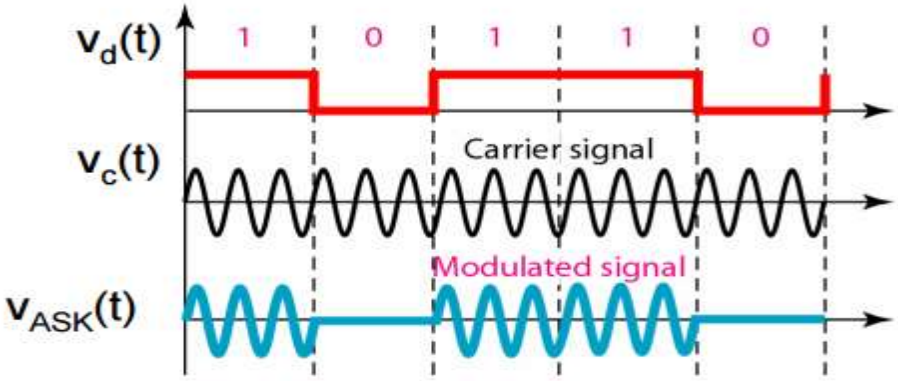
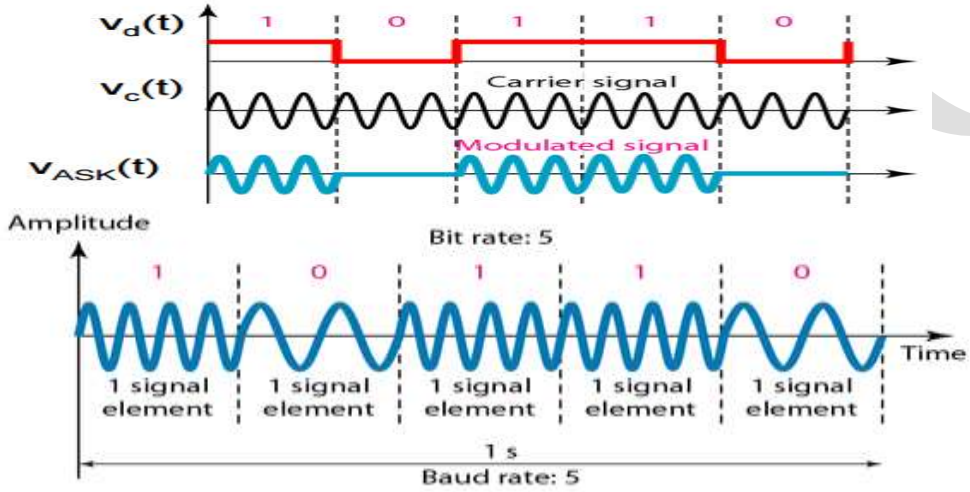
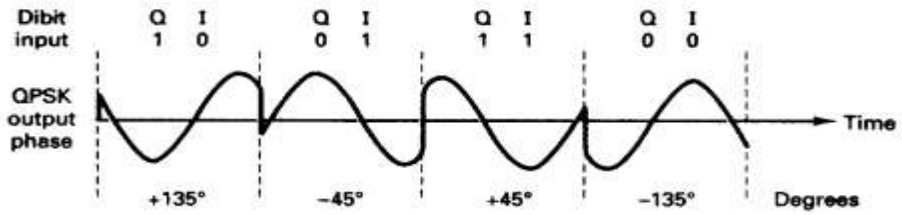


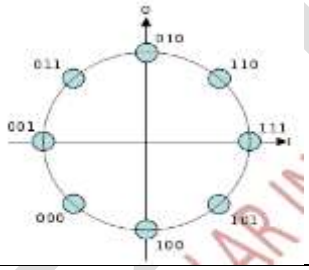
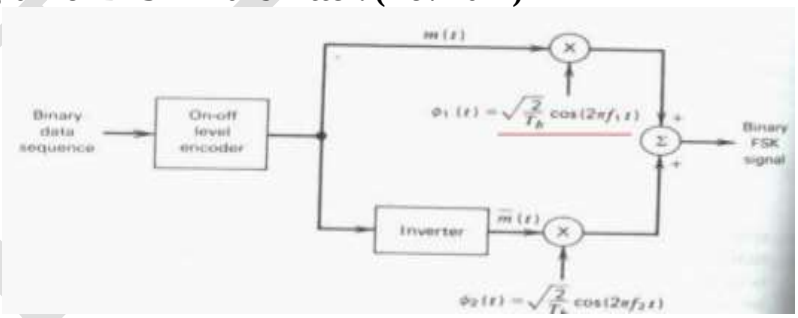
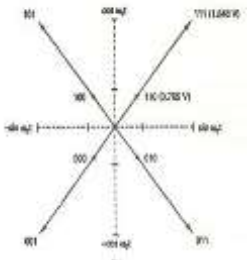
UNIT III DIGITAL COMMUNICATION

Amplitude Shift Keying (ASK) – Frequency Shift Keying (FSK)–Phase Shift Keying (PSK) – BPSK – QPSK – Quadrature Amplitude Modulation (QAM) – 8 QAM – 16 QAM – Bandwidth Efficiency– Comparison of various Digital Communication System (ASK – FSK – PSK – QAM).

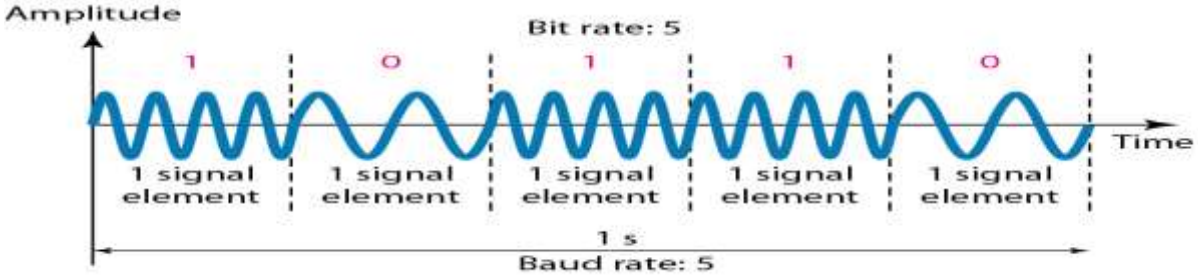
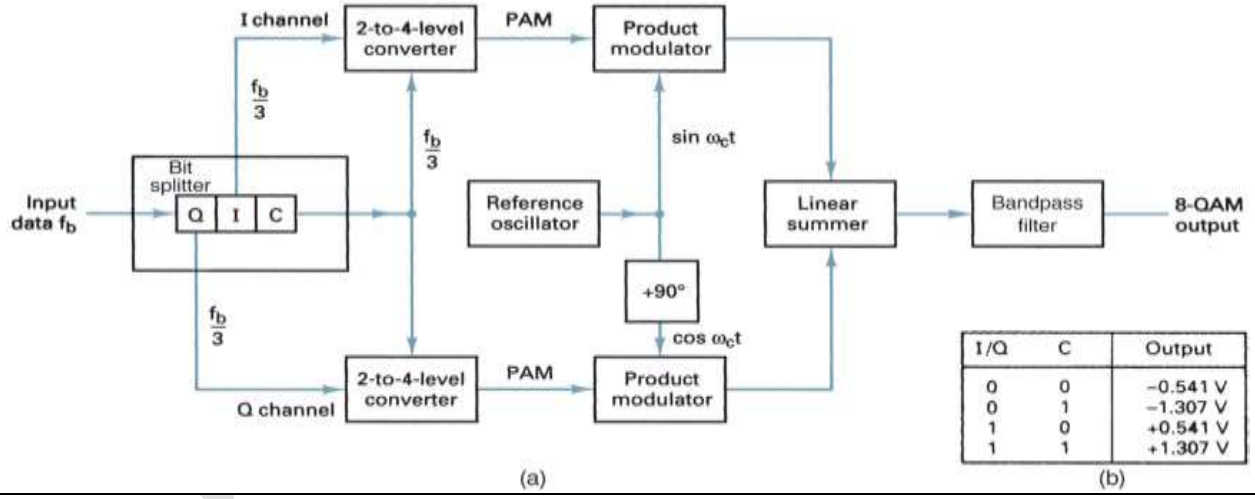
PART*A

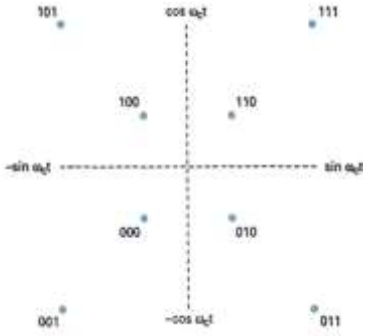
Q.No.	Questions
1.	**Draw the FSK signal for the binary message 1011001. (Nov 2017)** BTL 1 
2	**Define bandwidth efficiency. (Nov 2017) (May 2016) (Nov 2012)** BTL 1 The ratio of data rate to transmission bandwidth is referred as bandwidth efficiency. It is denoted as RBR. Bandwidth efficiency = R / B Where, R – Data rate B – Bandwidth
3	**Given the input binary sequence 1100100010, sketch the waveforms of the in-phase and quadrature components of a modulated wave obtained by using QPSK. (May 2017)** BTL 2 
4	**Determine bandwidth efficiency for a 8 PSK system operation with an information bit rate of 24 Kbps. (Nov 2016) (Nov 2014)** BTL 3 Baud is determined by baud = 24 kbps / 3 = 8000 Bandwidth is determined by B = 24 kbps / 3 = 8000 Bandwidth efficiency is calculated from $B\eta$ = 24, 000 / 8000 = 3 bits per second per cycle of bandwidth

5	**Draw the ASK signal for the given message 101101.(May 2016)** BTL 2 
6	**Sketch the digitally modulated waveform for the binary data 110101 using ASK, FSK. (Nov 2015) (May 2011)** BTL 2 
7	**Why FSK and PSK signals are preferred over ASK signals?(Nov 2015)** BTL 4 FSK is less Susceptible to errors than ASK. PSK is less susceptible to errors than ASK, while it requires/occupies the same bandwidth as ASK and more efficient use of bandwidth (higher data rate) are possible compared to FSK.
8	**Sketch the QPSK signal for the binary sequence 11001100. (May 2015)** BTL 2 
9	**Compare QPSK and 16 PSK signal in terms of bandwidth. (May 2015)** BTL 2 **16 PSK** i) 16 phases are used to represent two binary values. ii) Each signal element represents four bit. **QPSK** i) Four different phases are used to represent two binary values. ii) 2. Each signal element represents two bits.

10	**Define Shannon limit for information capacity.(May 2014) (May 2013) (Nov 2011)** BTL 1 I α B*t Information capacity is direct proportional to Bandwidth and transmission line. I = B log2(1+S/N) Or I =3.32 log1010(1+S/N) Where B—Band width S/N—Signal to noise ratio.
11	**How are bit rate and baud rate related? (May 2014)** BTL 2 The rate at which data (bits) are transmitted is called **bit rate**. That is number of bits transmitted per second. Unit is bps (bits per second). The rate at which signal elements (pulses) are transmitted is called baudrate (modulation rate). The number of signal elements (pulses) transmitted per second. Unit is bauds.
12	**What do you mean by FSK? (Nov 2013)** BTL 1 FSK (Frequency Shift Keying) also a modulation technique which converts digital data to analog signal. In FSK, the two binary values are represented by two different frequencies near the carrier frequency.
13	**What is M-ary encoding? (Nov 2013)** BTL 1 i) M-arysignaling schemes transmits M bits at a time. ii) Bandwidth requirement of M-arysignaling schemes is reduced.
14	**Draw the constellation diagram of QPSK signal.(May 2013)** BTL 1 
15	**Draw the block diagram of BFSK Transmitter. (Nov 2012)** BTL 1 
16	**Draw 8-QAM phasor diagram.(May 2012)** BTL 1 

17	**Determine the peak frequency deviation and minimum bandwidth for a BFSK signal with a mark frequency of 49 KHz and a space frequency of 51 KHz. (May 2012)**BTL 3 Bandwidth = fm-fs = 51- 49=2KHz
18	**What is binary phase shift keying? (Nov 2011)** BTL 1 BPSK (Binary Phase Shift Keying) also a modulation technique which converts digital data to analog signal. In PSK, the two binary values are represented by two different phases (0^o and 180^o).
19	**What are the advantages of QPSK? (May 2011)** BTL 1 The advantage of the Quadrature Phase Shift Keying (QPSK) modulation versus the Binary Phase Shift Keying (BPSK) one is well known. It is the possibility to transmit in the same frequency band twice more information, while the number of errors and the Eb/No relation are the same. It required minimum Bandwidth as in BPSK.
20	**Compare: Coherent and non-coherent detection.** BTL 2 **Coherent (synchronous) detection:** In coherent detection, the local carrier generated at the receiver is phase locked with the carrier at the Transmitter. The detection is done by correlating received noisy signal and locally generated carrier. The coherent detection is a synchronous detection. **Non-coherent (envelope) detection:** This type of detection does not need receiver carrier to be phase locked with Transmitter carrier. The advantage of such a system is that the system becomes simple, but the drawback is that error probability increases.
21	**Define information capacity.** BTL 1 Information Capacity represents the number of symbols that can be carried through a system that is called information capacity. It is a measure of how much information can be propagated through communication system and is function of bandwidth and transmission time. Information Capacity, I=B * t B= Bandwidth (Hz) t=Transmission time(sec) C (or) I = Channel capacity or Information capacity
22	**Mention any four advantage of digital modulation over analog modulation.** BTL 1 i) Maximum data rate ii) Minimum probability of symbol error iii) Minimum transmitted power. iv) Minimum channel bandwidth. v) Minimum circuit complexity vi) Maximum resistance to interfering signals
23	**What is a constellation diagram?** BTL 1 It is also called as signal state-space diagram, similar to phasor diagram where, the relative position of peaks of phasors is shown.
24	**Define QAM.** BTL 1 Quadrature amplitude modulation is a form of digital modulation where the digital information is contained in both the amplitude and phase of the transmitted carrier
25	**Give the Nyquist formulation for channel capacity.** BTL 1 fb =2B log2 M Where, fb –channel capacity (bps) B-minimum Nyquist bandwidth (Hz) M- number of discrete level or voltage levels

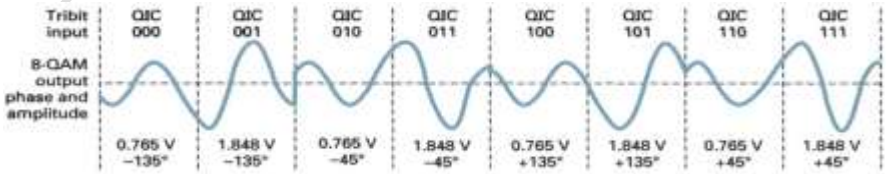
	PART*B
1	**With a block diagram explain the working of coherent binary FSK transmitter and receiver. (7M) (Nov 2017) (Nov 2016)** BTL 2 **Answer: Page-55-W.Tomasi** **BFSK:** Analog carrier frequency - varied -according - binary information signal. (2M) **Block diagram**: Transmitter , receiver (2M) **Explanation:**Coherent- carrier present. (3M) Receiver carrier - should be -same - carrier at transmitter. Same Transmitter - both coherent, non-coherent FSK. **Waveforms:** 
2	**Draw the block diagram of 8-QAM Transmitter and explain its working. (7M) (Nov 2017)** BTL 2 **Answer: Page-81-W.Tomasi** **8-QAM**:Combination of ASK , PSK(1M) **Block diagram of 8-QAM Transmitter:**(2M) 



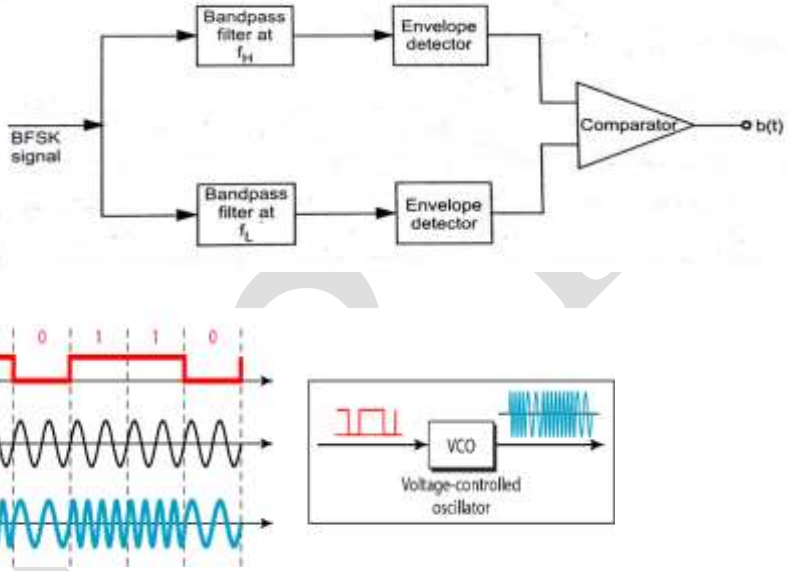
Explanation: Incoming data : divided- three bits (I,Q,C)(4M)
2-4 level converter: Digital to analog converter
Product modulator: Multiplies PAM -Carrier.
I and Q: Determines polarity
C: Determines magnitude
I channel: PAM multiplied - reference carrier
Q channel: PAM multiplied with- 90◦ phase shifted carrier.

What is the significant of QAM? Explain the operation of 8 QAM Transmitter and receiver using a block diagram and truth table. (13M) (May 2016) (Nov 2013) BTL 2
Answer: Page-81-W.Tomasi

8-QAM:Combination of ASK and PSK-3 bits-8 combinations. (1M)
Block diagram of 8-QAM Transmitter and Receiver: (2M)
Truth Table:(1M)
Explanation:
Transmitter: Incoming data : divided- three bits (I,Q,C) (5M)
2-4 level converter: Digital to analog converter
Product modulator: Multiplies PAM -Carrier.
I and Q: Determines polarity
C: Determines magnitude
I channel: multiply PAM with reference carrier
Q channel: multiply PAM with 90◦ phase shifted carrier
Receiver: Power splitter: directs input- I and Q product detectors-carrier recovery circuit (3M)
Product Detector: Incoming signal-mixed-recovered carrier. Output-4 level PAM
ADC: PAM to 3 bits.
Waveform:(1M)



3	**Define binary FSK and explain about the generation and detection of binary FSK signals using block diagram. (13M) (May 2017) (May 2016) (Nov 2015) (May 2014) (May 2013) (Nov 2011)** BTL 2 **Answer: Page-55-W.Tomasi**

	BFSK: Frequency of the analog carrier - varied according - binary information signal. (2M) **Block diagram**: BFSK-Transmitter and receiver. (2M) **Explanation:**Coherent- carrier present.(6M) Non-Coherent: carrier absent Same Transmitter -both coherent, non-coherent FSK. FSK modulator: Voltage Controlled Oscillator Logic 1: shifts VCO output-fm. Logic 0: shifts VCO output - fs. Multiplier: Multiplies - FSK ,Carrier Comparator: Compares two input- responds - largest output. **Transmitter, receiver, waveforms:**(3M) 
4	**Compare the various digital modulation schemes. (6M) (Nov 2016) (Nov 2014)** BTL 4 **Answer: Page-106-W.Tomasi** i) Definition of ASK,PSK,FSK,QPSK,8-QAM,16-QAM (1M) ii) Waveforms (1M) iii) Phasor diagram (1M) iv) Constellation diagram (1M) v) Truth table (1M) vi) Performance (1M)
7	**Explain in detail about the operation of QPSK Transmitter with necessary diagrams. (7M) (Nov 2015)** BTL 2 **Answer: Page-67-W.Tomasi** **QPSK**:M-ary encoding scheme.N = 2 and M =4 (2M) **Block diagram**: (1M)

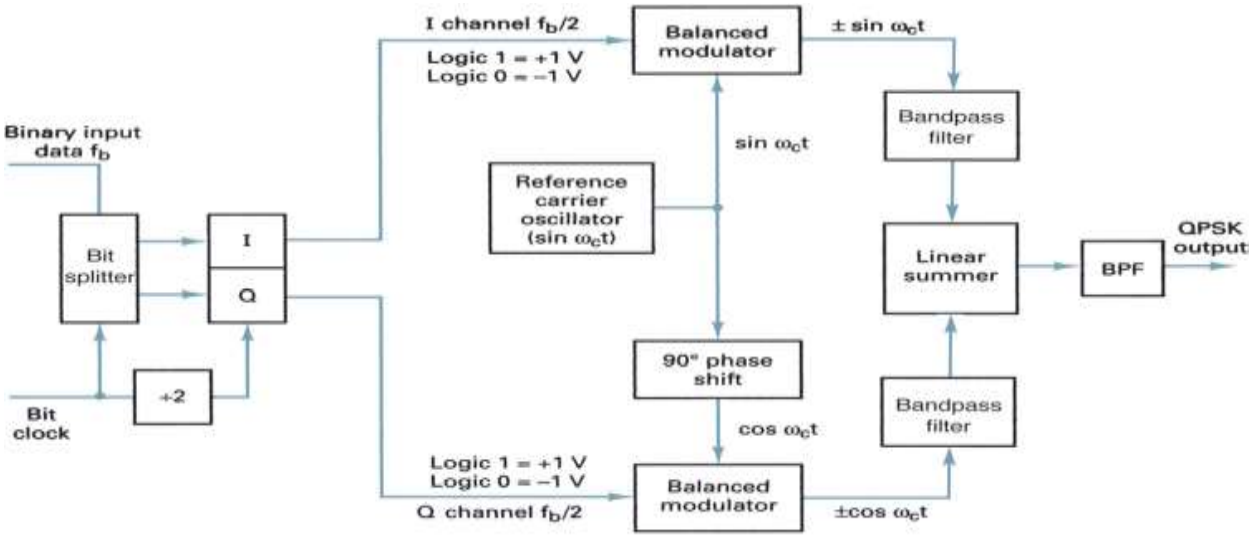
Signal Constellation: (1M)

Explanation:2 input bits (3M)

Incoming data - divided - two bits (I,Q)

Product modulator- multiplies PAM - Carrier.

Linear summer- Sums up - signal generated from- I, Q channel



8 **Compare QPSK and BPSK. (6M) (Nov 2015)** BTL 4

Answer: Page-67-W.Tomasi

Sl.No	QPSK	BPSK
1	Type of Modulation- binary i.e. two level	Type of Modulation- four level.
2	Group- two binary bits – represented- one phase state.	One bit- represented - one phase
3	Bit rate - twice -Baud Rate.	Bit Rate - equal- Baud Rate
4	Very high bit rate applications.	High bit rate applications
5	Complex.	Less complex.
6	Less error probability.	More error probability.

9 **Draw the constellation diagram of QPSK modulation and explain the QPSK modulation and demodulation of QPSK. (13M) (May 2015) (Nov 2014) (May 2014) (Nov 2012) (Nov 2011) (May 2011)** BTL 2

Answer: Page-67-W.Tomasi

QPSK: M-ary encoding scheme -N = 2 -M =4 (1M)

Block diagram: Transmitter and Receiver (2M)

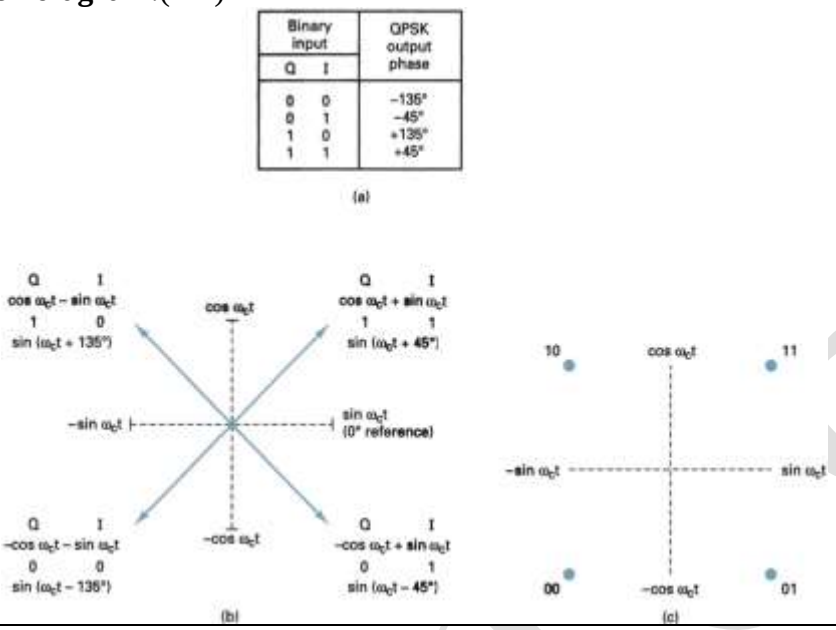
Explanation of modulator: 2 input bits - possible (5M)

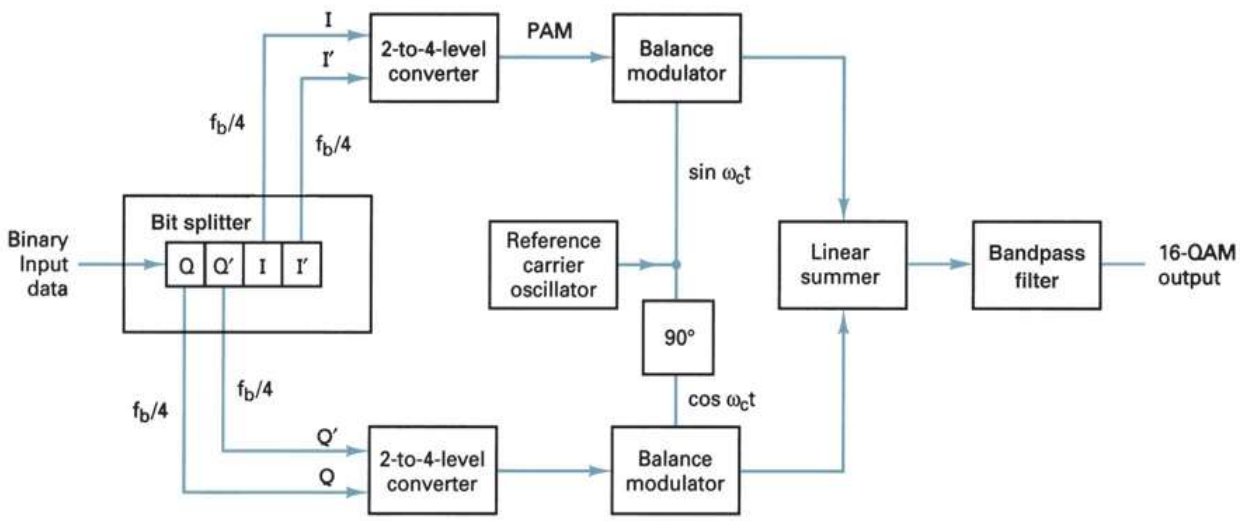
Incoming data - divided - two bits (I,Q)

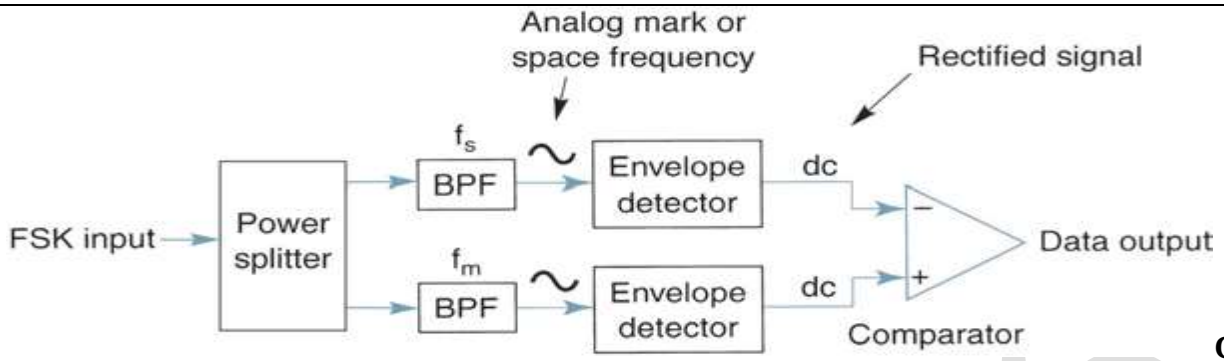
Product modulator: multiplies PAM - Carrier.

Linear summer: Sums up - signal generated – I,Q channel

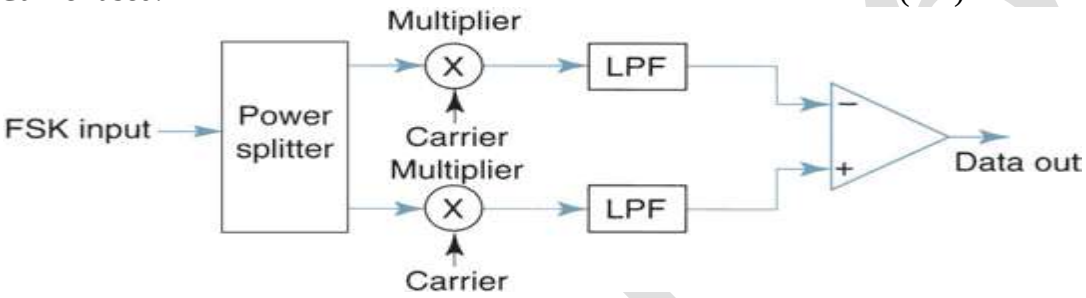
Explanation of demodulator: I, Q channel signals filtered separately - message signal - obtained. (4M)

	Signal constellation diagram:(1M) 
10	**Explain the method of generation of QAM and the demodulation of the same. (13M) (May 2015) (May 2011)** BTL 2 **Answer: Page-81-W.Tomasi** **QAM**:Combination - ASK and PSK. (1M) 8-QAM: 3 bits- 8 combinations 16 QAM: 4 bits- 16 combinations **Block diagram of 8-QAM or 16-QAM Transmitter and Receiver:** (2M) **Truth Table:**(1M) **Explanation:** **16 QAM Transmitter:** Incoming data : Divided- four bits (I,Q,I',Q') (5M) 2-4 level converter: Digital to analog converter Product modulator: Multiplies PAM -Carrier. I,Q: Determines polarity I',Q': Determines magnitude I channel: PAM multiplied-reference carrier Q channel: PAM multiplied - 90◦ phase shifted carrier **16 QAM Receiver:**(4M) Power splitter: directs input- I, Q product detectors-carrier recovery circuit. Product Detector: Incoming signal-mixed-recovered carrier. Output-4 level PAM. ADC: PAM to 4 bits.

	
11	**Write a note on bandwidth consideration in FSK. (6M) (May 2014)** BTL 1 **Answer: Page-55-W.Tomasi** **FSK**: Analog carrier Frequency – varied - according - binary information signal.(2M) **Bandwidth**: B= $\lvert(f_s - f_b) - (f_m - f_b)\rvert$ (4M) $\lvert f_s - f_m \rvert$=2Δf B=2(Δf+f_b) B-Minimum nyquist Bandwidth Δf-frequency deviation f_b-input bit rate (bps) H= $\lvert f_s - f_m \rvert$/2
	PART*C
1	**Explain in detail about the BFSK with coherent and non coherent receivers. Draw the block diagram with frequency domain representation. (15M)** BTL 2 **Answer: Page-55-W.Tomasi** **BFSK:** Frequency of the analog carrier - varied according - binary information signal. (1M) **Block diagram**: BFSK-Transmitter and receiver (4M) **Explanation:**Coherent- carrier present (6M) Non-Coherent: carrier absent Carrier at receiver -should -same - carrier at transmitter. Same Transmitter – both- coherent - non-coherent FSK. FSK modulator: Voltage Controlled Oscillator Logic 1: shifts VCO output-fm-mark frequency Logic 0: shifts VCO output – fs-space frequency Multiplier: Multiplies FSK - Carrier Comparator: Compares two input- responds - largest output.



Coherent receiver: Carrier used. (2M)



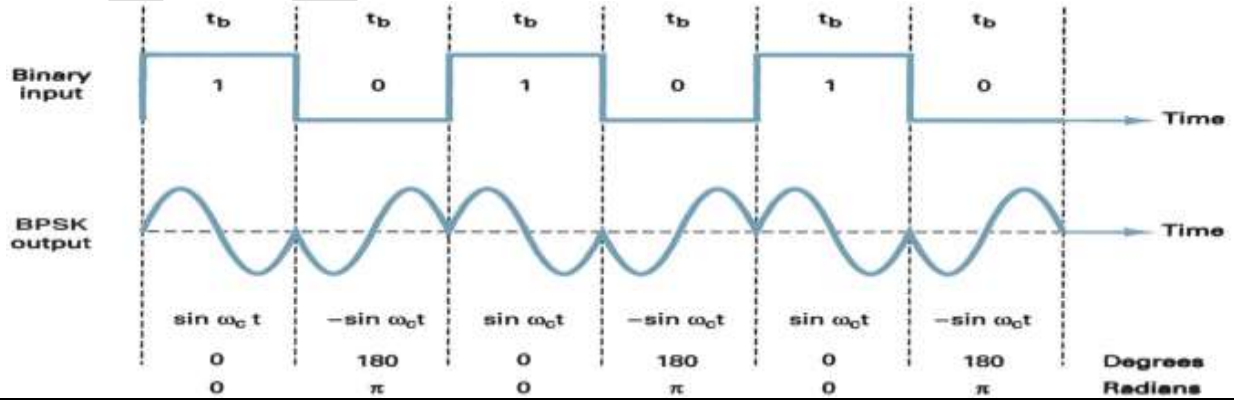
Non coherent receivers:No carrier (2M)

2 **Compare different digital modulation techniques. (15M)** BTL 4
Answer: Page-106-W.Tomasi

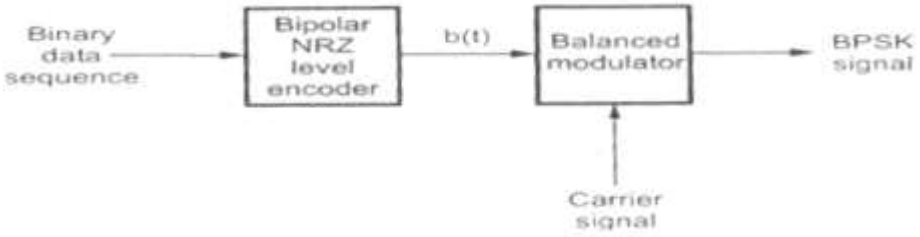
i) Definition of ASK,PSK,FSK,QPSK,8-QAM,16-QAM (3M)
ii) Waveforms (2M)
iii) Phasor diagram (3M)
iv) Constellation diagram (2M)
v) Truth table (2M)
vi) Performance (3M)

3 **Explain about binary phase shift keying. (15M) (May 2014) (Nov 2013) (Nov 2012) (May 2012) (May 2011)** BTL 2
Answer: Page-62-W.Tomasi

BPSK: Analog carrier phase - varied - according - binary information signal. (1M)
Waveform:(1M)



Block diagram: Transmitter, receiver (2M)
Constellation Diagram:(1M)
Explanation:Two phases (4M)
Symbol 0-180^o
Symbol 1- 0^o
Output carrier shifts-between-two phases-seperated-180^o
Other name-phase reversal keying, biphase modulation.
Transmitter:(3M)
Carrier oscillator: high frequency sine wave-carrier signal.
Level Converter: Converts 1-(+1V), 0-(-1V).
Balanced Modulator: Product Modulator-multiplies-input-carrier.



Receiver: (3M)
Input signal: sin$\omega_c t$ or –sin $\omega_c t$
Output signal: logic 1-logic 0.

UNIT IV-SOURCE AND ERROR CONTROL CODING

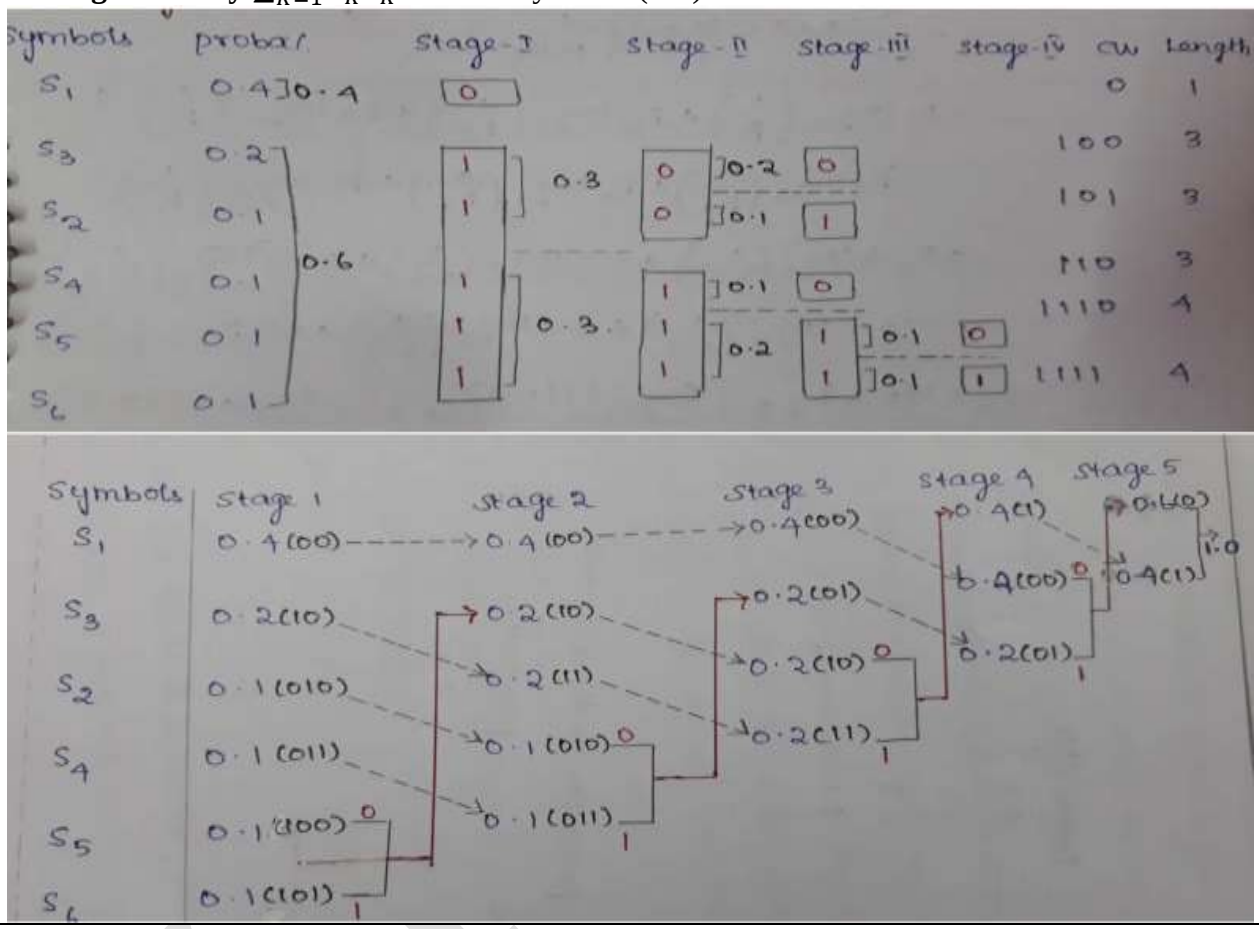
Entropy, Source encoding theorem, Shannon fano coding, Huffman coding, mutual information, channel capacity, Error Control Coding, linear block codes, cyclic codes – ARQ Techniques.

PART* A

Q.No.	Questions
1.	**Calculate the entropy of four possible messages {Q1, Q2, Q3, Q4} which is transmitted with probabilities {1/8, 3/8, 3/8, 1/8}.(Nov 2017)** BTL 3 $$H(s) = \sum_{i=1}^{3} p_i \log_2 \frac{1}{p_i}$$ $$= \frac{1}{2}\log_2(2) + \frac{1}{4}\log_2(4) + \frac{1}{4}\log_2(4) = 1.5 \text{ bits/symbol}$$
2	**Define entropy. (May 2017) (May 2016) (Nov 2014) (May 2011)** BTL 1 Entropy is the measure of the average information content per second. It is given by the expression H(X)=ΣI P(xi)log2P(xi) bits/sample.
3	**Consider a discrete memory less source with source alphabet (S_0, S_1, S_2) and with their respective probabilities ($P_0 = \frac{1}{4}, P_1 = \frac{1}{4}, P_2 = 1/2$) entropy of the source. (May 2017) (Nov 2016)** BTL 3 $$H(s) = \sum_{i=1}^{3} p_i \log_2 \frac{1}{p_i}$$ $$= \frac{1}{2}\log_2(2) + \frac{1}{4}\log_2(4) + \frac{1}{4}\log_2(4) = 1.5 \text{ bits/symbol}$$
4	**Define mutual information and mention its properties. (May 2017) (May 2015)** BTL 1 Mutual information I(X,Y) of a channel is defined by I(X,Y)=H(X)-H(X/Y) bits/symbol H(X)- entropy of the source, H(X/Y)- conditional entropy of Y. **Properties:** i) I(X,Y)=I(Y,X) ii) I(X,Y)>=0 iii) I(X,Y)=H(Y)-H(Y/X) iv) I(X,Y)=H(X)+H(Y)-H(X,Y).
5	**When a binary code is said to be cyclic code? (Nov 2016)** BTL 4 An (n, k) linear block code C is said to be cyclic if for every code word c = (c0, c1, . . . , cn−1) in C, there is also a code word c 0 = (cn−1, c0, . . . , cn−2) that is also in C. (c 0 is a cyclic shift of c.)
6	**List out the properties of cyclic codes. (May 2016)** BTL 2 Linearity property: The sum of any two code words in the code is also a code word. Cyclic property: Any cyclic shift of a code word in the code is also a code word.

7	**Show that if $C_i$and C_j are two code vectors in the (n,k) linear block code, then their sum is also a code vector with an example. (Nov 2015)** BTL 2 $$C_i \oplus C_j = m_i G \oplus m_j G = (m_i \oplus m_j)G$$ This implies that the modulo-2 sum of any two codewords ($(C_i and\ C_j)$ is another code word. Similarly mod-2 sum of any two message words $(m_i and\ m_j)$ represents a new message word. All zero code vector is always a code word since $C_i \oplus C_j = 0$.
8	**Define channel capacity of a discrete memory less channel. (Nov 2015)** BTL 1 The channel capacity of the discrete memory less channel is given as maximum averagemutual information. The maximization is taken with respect to input probabilities. **C=max I(X;Y), over all (p(x_1), $p(x_2)$, ……, $p(x_m)$)**
9	**An event has six possible outcomes with probabilities ½, ¼,1/8, 1/16, 1/32. Find the entropy of the system. (May 2015)** BTL 3 H = $\Sigma p_k log_2 {}^1/_{p_k}$ = (½)$log_2 2$ + (¼)$log_2 4$ + (1/16) $log_2 16$ + (1/32) $log_2 32$ + (1/32) $log_2 32$ = 1.5625.
10	**What are the two primary methods used for error correction?(Nov 2014)(Nov 2012)** BTL 1 i) Retransmission. ii) Forward error correction.
11	**What are linear block codes?(Nov 2014)** BTL 1 • A block code is a code in which k bits (or, more generally, symbols) are input and n bits (or, more generally symbols) are output. We designate the code as an (n, k) code.If we input k bits, then there are 2k distinct messages. Each message of n symbols associated with each input block is called a codeword. • A Linear Code has the following properties: (i) The sum of two codewords belonging to the code is also a codeword belonging to the code. (ii) The all-zero codeword is always a codeword. (iii) The minimum Hamming distance between two codewords of a linear code is equal to the minimum weight of any non-zero codeword, i.e., d* = w*.
12	**What are the advantages and disadvantages of error detection? (May 2014)**BTL 1 Error detection is defined as the process of monitoring the transmission of data and find when an error has occurred. Disadvantages: It won't correct the errors.
13	**Differentiate between error detection and error correction. (Nov 2013)**BTL 2 Errors introduced by communications faults, noise or other failures into valid data, especially compressed data were redundancy has been removed as much as possible, can be detected and/or corrected by introducing redundancy into the data stream. Error Detection is the process of monitoring data transmission and determining when errors have occurred. Purpose of error detection is not to prevent errors from occurring but to prevent undetected errors from occurring. This technique neither corrects nor identifies which bits are in error and they indicate when an error has occurred.
14	**List the methods of error correction. (May 2013)**BTL 1 i) Retransmission. ii) Forward error correction.
15	**What is the need for error control coding? (May 2011)**BTL 2 The main use of error control coding is to reduce the overall probability of error, which is also known as channel coding.

16	**Explain Shannon-Fano coding.** BTL 1 An efficient code can be obtained by the following simple procedure, known as Shannon-Fanoalgorthim. i) List the source symbols in order of decreasing probability. ii) Partition the set into two sets that are as close to equi probable as possible, and sign 0 to the upper set and 1 to the lower set. iii) Continue this process, each time partitioning the sets with as nearly equal probabilities as possible until further partitioning is not possible.
17	**When would be the average information delivered by a source of alphabet size 2 is maximum?**BTL 2 Average information is maximum, when the two messages are equally likely i.e., p1 = p2 = 1/2. Then the maximum average information is given as, Hmax = 1/2 $log_2 2$ + 1/2 $log_2 2$ = 1 bit / message.
18	**What is the channel capacity of a BSC and BEC?** BTL 1 For BSC the channel capacity C=1+plog_2 p +(1-p) log_2 (1-p). For BEC the channel capacity C=(1-p).
19	**Give the properties of syndrome in linear block code.**BTL 1 i) The syndrome depends only on the error patterns and not on the transmitted code word. ii) All error patterns that differ by a code word have the same syndrome.
20	**Give the difference between linear block code and cyclic code.**BTL 2 i) Linear block code can be simply represented in matrix form. ii) Cyclic code can be represented by polynomial form.
21	**Define Hamming distance (HD).** BTL 1 The number of bit position in which two adjacent code vectors differs is known as Hamming distance. (e.g) if c1 = 1 0 0 1 0 1 1 0 and c2 = 1 1 0 0 1 1 0 1 then HD=5.
22	**Write the syndrome properties of linear block codes.**BTL 1 Syndrome is obtained by S= YH^T. If Y=X, then S=0 ie no error in output. If y ≠x, then S≠ 0 ie there is error in output. Syndrome depends upon the error pattern only, S=EH^T.
23	**State the channel coding theorem for a discrete memory less channel.**BTL 1 Given a source of Mequally likely messages, with M>>1, which is generating information at rate R. Given channel with capacity C. Then if, R ≤ C
24	**Why cyclic codes are extremely well suited for error detection?**BTL 2 Cyclic codes are well suited for error detection because of the following reasons: They are easy to encode. They have well defined mathematical structure.
25	**What is error detection?**BTL 1 The decoder accepts the received sequence and checks whether it matches a valid message sequence. If not, the decoder discards the received sequence and notifies the transmitter (over the reverse channel from the receiver to the transmitter) that errors have occurred and the received message must be retransmitted. This method of error control is called error detection.
	PART*B
1	**Five source messages are probable to appear as m1 =0.4, m2= 0.1, m3 = 0.2, m4 = 0.1, and m5 = 0.1, m6 =0.1. Determine the coding efficiency for** **1) Shannon-Fano coding**

	2) Huffman coding (13M) (Nov 2017)BTL 3 **Answer: Page:187&193- Notes** **Explanation**: Arrange - given probabilities - decreasing order. (1M) Perform calculation. (8M) **Coding efficiency**:$\sum_{K=1}^{K} P_K L_K$=2.4 bits/symbol . (4M) 
2	**Derive the expression for mutual information and channel capacity. (Nov 2017) (7M)** BTL 2 **Answer: Page:5.87- Chitode** **Mutual information:** Information gain amount-per- received signal. I(X;Y)= H(X)-H(X/Y) (1M) **Channel capacity**: C= max I(X;Y) (1M) **Expression and properties of mutual information**:(2M) Symmetric Non negative Related to joint entropy -channel input - channel output **Types of channel capacity :**(3M) Noise free channel Symmetric channel Binary symmetric channel Cascaded channel

	Binary erasure channel.
3	**Discuss the types of error control coding. (6M)(Nov 2017)**BTL 2 **Answer: Page:5.37- Chitode** **Error control coding**:developing methods - coding - check correctness - bit stream transmitted. (1M) **Types of error control codes:**(2M) Linear block codes Cyclic codes **Error detection and error correction:**(3M) **Error detection**: detecting errors. **Error correction**: detecting - correcting errors.
4	**Draw and explain the generalized** **i) (n,k) cyclic encoder to implement an encoding procedure for an (n,k) cyclic code in systematic form** ii) **Syndrome calculator and properties of syndrome polynomial. (May 2017) (13M)** BTL 2 **Answer: Page:169.W.Tomasi** **Cyclic code**:Any cyclic shift – code word - code -also codeword. (2M) K=M+1; where K- Constraint Length M- No of Memory elements **Properties of cyclic code**: (2M) Linearity property: sum - two code words - also code word. Cyclic property: Any cyclic shift - code word - in code -also code word. **Explanation:**(5M) Find$x^{n-k}m(x)$ Divide$x^{n-k}m(x)$by g(x) Add r(x) - first step result - givescodeword. **Syndrome calculation**:(4M) Dividing error polynomial - g(x) - remainder - error syndrome polynomial.
5	**Five source messages are probable to appear as m1=0.4, m2=0.15, m3=0.15, m4=0.15 and m5=0.15. Find coding efficiency for** **i) Shannon-Fano coding** **ii) Huffman coding. (13M)(Nov 2016)**BTL 3 **Answer: Page:187&193- Notes** **Explanation**: Arrange - given probabilities - decreasing order.(1M) Perform calculation. (8M) **Coding efficiency**:$\sum_{K=1}^{K} P_K L_K$=2.4 bits/symbol (4M) **Ref Q.1 – Part B**
6	**The generator polynomial of a (7,4) cyclic code is given by G(D)=$x^3 + x + 1$. Compute all the non-systematic codewords for the message 100. (13M)(May 2016)**BTL 3 **Answer: Page:5.37- Chitode** **Cyclic code**: (2M)

	K=M+1; where K- Constraint Length M- No of Memory elements **Find**: $x^{n-k}m(x)$ $=x^6+x^3$ (4M) **Divide**: $x^{n-k}m(x)$by g(x) - gives r(x)= x^2+x (4M) Add r(x) - first step result - givescodeword= $x^6+x^3+x^2+x$ =1001110.(3M)
7	**Consider a systematic block code whose parity check equation are** **P1=m1+m2+m4** **P2=m1+m3+m4** **P3=m1+m2+m3** **P4=m2+m3+m4** **Where Mi is the message digits and Pi are the parity digits?** **i) Find the generator matrix and the parity check matrix for this code.** **ii) How many errors can be detected and corrected?** **i) If the received code word is 10101010, find the syndrome. (13M) (May 2016)** BTL 3 **Answer: Page:167-W.Tomasi** **Generator matrix:**G=[P:I_k] (3M) **Parity Check matrix**: H=[P^T: I](2M) **Length**: t ≤ $d_{min}-1$(1M) **Syndrome**: S=rH^T(3M) **Original codeword**:r ex-or e.(4M)
8	**A source generates five messages m0, m1, m2, m3, m4 with probabilities 0.55, 0.15, 0.15, 0.10, 0.05 respectively. The successive message emitted by the source are statistically independent. Determine code words for the messages and efficiency using shannonFano algorithm. (13M) (Nov 2015)**BTL 3 **Answer: Page:5.37 Chitode** **Explanation**: Arrange - given probabilities - decreasing order.(1M) Perform calculation. (8M) **Coding efficiency**:$\sum_{K=1}^{K} P_K L_K$=2.4 bits/symbol .(4M) **Ref Q.1 – Part B**
9	**Design a cyclic encoder for the same (7,4) cyclic code and obtain code vector for the message vector 1110. (7M) (Nov 2015)**BTL 3 **Answer: Page:266- Notes** **Cyclic code:** Any cyclic shift – code word - code -also codeword. K=M+1; where K- Constraint Length M- No of Memory elements **Multiply**:$x^{n-k}m(x)$ $=x^6+x^5+x^4$(2M) **Divide:**$x^{n-k}m(x)$by g(x) to give r(x) = x^2(3M) Add r(x) - first step result - gives codeword ==1110100. (2M) **Syndrome:** R(x)=q(x)g(x)+s(x).
10	**Find out the Huffman code for a discrete memoryless source with probability statistics {0.1, 0.1, 0.2, 0.2, 0.4}. (7M) (Nov 2014)**BTL 3 **Answer: Page:193- Notes**

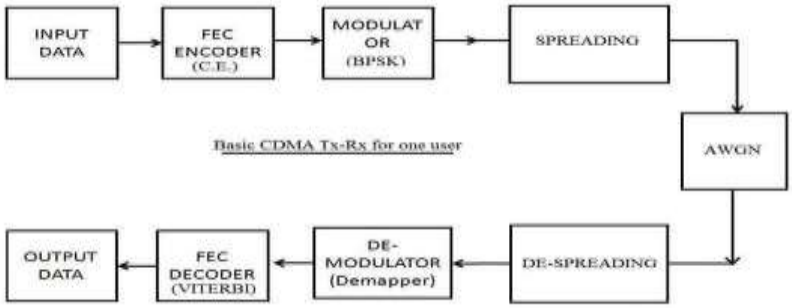
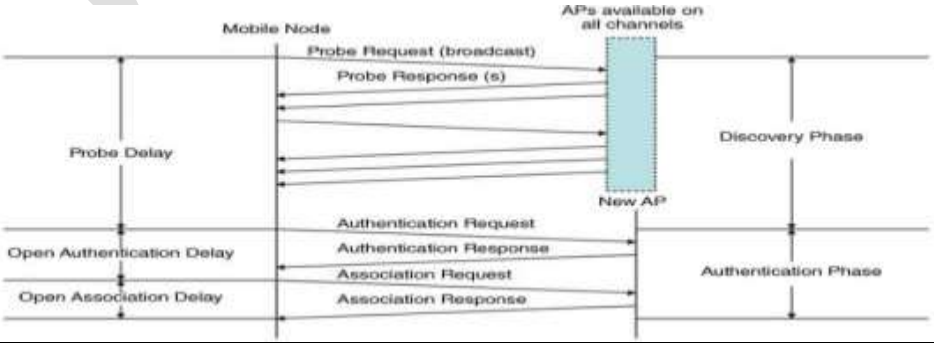
	Explanation: Arrange - given probabilities - decreasing order.(1M) Perform calculation. (4M) **Coding efficiency**:$\sum_{K=1}^{K} P_K L_K$=2.4 bits/symbol.(2M) Ref Q.1 – Part B
11	**Describe the concept of channel capacity. (7M) (Nov 2014)**BTL 2 **Answer: Page:5.120-chitode** **Channel capacity**: Represents uncertainity-about channel input - resolved by observing -channel output. (2M) C=max I(X;Y), over all (p(x_1), $p(x_2), \ldots\ldots, p(x_m)$) **Explanation**:Channel capacity per second Capacities of Special channel Loseless Channel Deterministic Channel Noiseless Channel Binary Symmetric Channel(5M)
12	**Write short notes on linear block codes. (7M) (Nov 2014)**BTL 2 **Answer: Page:167.W.Tomasi** **Generator matrix:**G=[P:I_k](1M) **Parity Check matrix**: H=[P^T: I](1M) **Length**: t ≤ $d_{min} - 1$(1M) **Syndrome**: S=rH^T (2M) **Original codeword**:r ex-or e.(2M)
13	**Devise a single-bit error correction code for 8-bit data and illustrate with an example. (7M) (May 2014)**BTL 2 **Answer: Page:167.W.Tomasi** **Hamming code:** Single bit error correction.(1M) **Explanation:** example –error correction (1M) Block length: n=$2^m - 1$(2M) No of message bits: k= $2^m - m - 1$ (1M) No of parity bits: n-k=m. (2M)
14	**Explain source coding theorem. Consider five messages So, S1, S2, S3, S4 given by the probabilities ½, ¼, 1/8, 1/16, 1/16. Use Shannon Fano algorithm to develop an efficient code. (13M) (May 2012)** BTL 3 **Answer:Page:188-Notes** **Explanation**: Arrange - given probabilities - decreasing order.(1M) Perform calculation. (8M) **Coding efficiency**:$\sum_{K=1}^{K} P_K L_K$=2.4 bits/symbol (4M) **Ref Q.1 – Part B**
	PART * C
1	**Consider a linear block code with generator matrix** **G=[1 1 0 1 0 0 0; 0 1 1 0 1 0 0; 1 1 1 0 0 1 0; 1 0 1 0 0 0 1] (Dec-2016)**

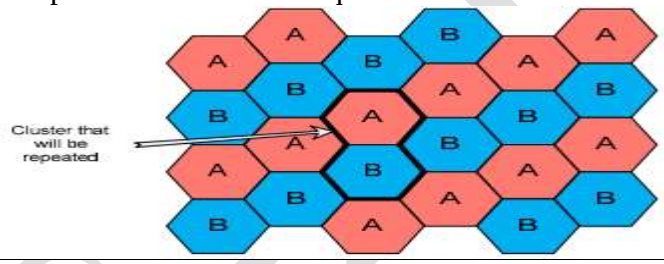
	i) Determine the parity check matrix (3M) **ii) Determine the error detecting and capability of the code (3M)** **iii) Draw the encoder and syndrome calculation circuits. (6M)** **iv) Calculate the syndrome for the received vector** **r = [1 1 0 1 0 1 0] (3M)** BTL 3 **Answer: Page:217- Notes** **Parity Check matrix**: H=[P^T: I] **Length**: t ≤ $d_{min} - 1$ **Syndrome**: S=rH^T **Original codeword**:r ex-or e.
2	**For a systematic (6,3) linear block code with parity matrix** **1 1 0** **1 0 1** **0 1 1** **Find all the possible code vectors. Construct the syndrome decoding table and decode the received code word 110001. (15M)** BTL 3 **Answer: Page:217- Notes** **Generator matrix:**G=[P:I_k](3M) **Parity Check matrix**: H=[P^T: I](2M) **Length**: t ≤ $d_{min} - 1$(2M) **Syndrome**: S=rH^T(4M) **Original codeword**: r ex-or e. (4M)
3	**Explain in detail about error detection and correction. (15M)** BTL 2 **Answer: Page:165.W.Tomasi** **Error control coding**:developing methods - coding - check correctness - bit stream transmitted. (2M) **Types of error control codes:**(3M) Linear block codes Cyclic codes **Error detection and error correction:**(5M) **Error detection**: detecting errors Forward error correction, Retransmission. **Error correction**: detecting - correcting errors.(5M)

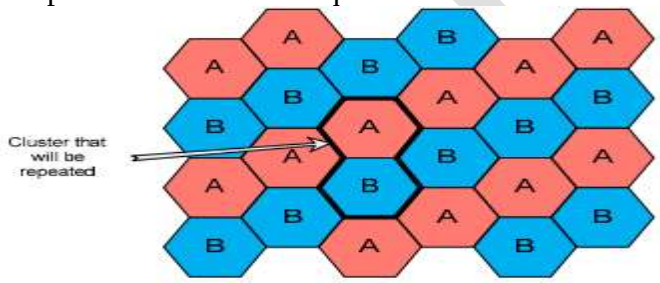
UNIT V MULTI-USER RADIO COMMUNICATION

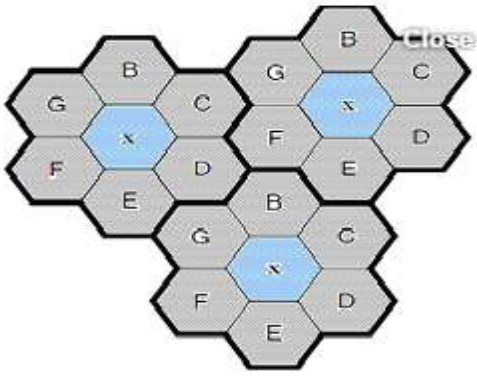
Global System for Mobile Communications (GSM) – Code division multiple access (CDMA) – Cellular Concept and Frequency Reuse – Channel Assignment and Handover Techniques – Overview of Multiple Access Schemes – Satellite Communication – Bluetooth.

PART-A

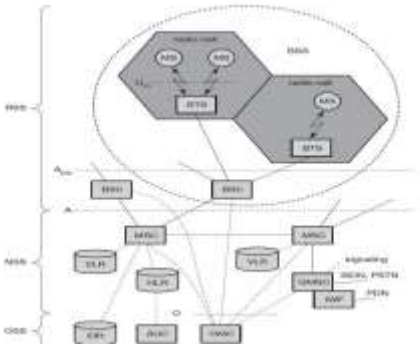
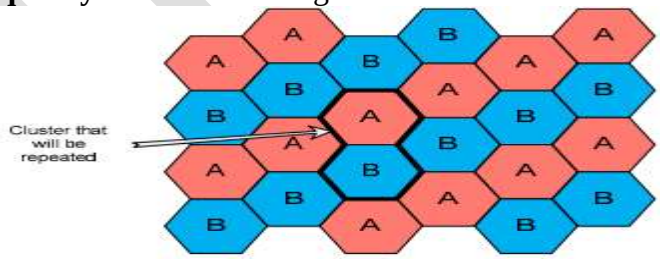
Q.No	Questions
1.	**Differentiate between GSM and CDMA. (Nov 2017)** BTL 2 Global System for Mobile Communication. It was developed to provide a common second generation technology for Europe so that the same subscriber units could be used throughout the continent. Transmission is in the form of Direct Sequence Spread Spectrum (DSSS) which uses a chipping code to increase the data rate of the transmission, resulting in an increased signal bandwidth. Multiple access is provided by assigning orthogonal chipping code to multiple users, so that the receiver can recover the transmission of an individual unit from multiple transmissions.
2	**What is Bluetooth technology? And mention its application. (Nov 2017)** BTL 1 i) Data and voice access points ii) Ad-hoc networking iii) Cable replacement
3	**What are all the essential components of GSM? (May 2017)** BTL 1 i) Mobile station ii) Base station subsystem iii) Network switching subsystem iv) Operation support subsystem
4	**Draw the block diagram of CDMA transmitter and receiver. (May 2017)** BTL 1  Basic CDMA Tx-Rx for one user
5	**What are the steps involved in Handoff process? (Nov 2016)** BTL 1 A handoff refers to the process of transferring an active call or data session from one cell in a cellular network to another or from one channel in a cell to another. 

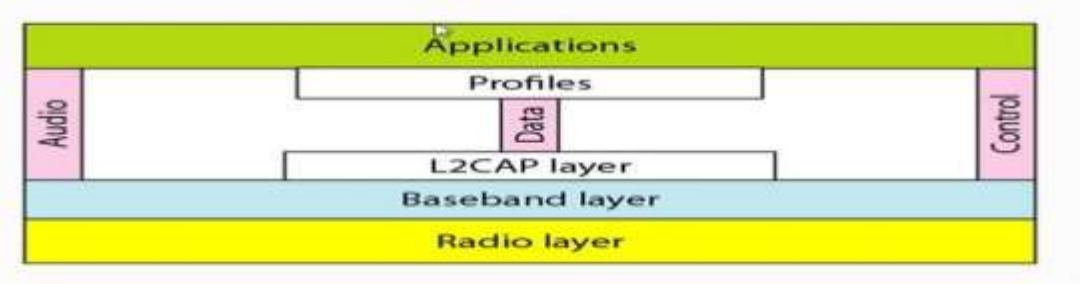
6	**Mention the three most commonly used multiple access technique. (Nov 2016)** BTL 1 In any cellular system it is necessary for it to be able have a scheme whereby it can handle multiple users at any given time. The multiple access schemes are known as FDMA, TDMA, CDMA and OFDMA. There are a number of requirements that any multiple access scheme must be able to meet: i) Ability to handle several users without mutual interference. ii) Ability to be able to maximise the spectrum efficiency iii) Must be robust, enabling ease of handover between cells.
7	**What is meant by frequency reuse? (May 2016)** BTL 1 Frequency reuse is the process of using the same radio frequencies on radio transmitter sites within a geographic area that are separated by sufficient distance to cause minimal interference with each other. Frequency reuse allows for a dramatic increase in the number of customers that can be served (capacity) within a geographic area on a limited amount of radio spectrum (limited number of radio channels). Cell Frequency Reuse Cluster [Invalid]: In this cluster, the allocated band is divided into 2 bands and the two sub-bands are reused in an alternating fashion somehow. Clearly, only two cells with the same frequency as a particular cell are at equal distance from it. 
8	**What is the coverage range of bluetooth? (May 2016)** BTL 1 Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz) from fixed and mobile devices, and building personal area networks (PANs). Invented by telecom vendor Ericsson in 1994, it was originally conceived as a wireless alternative to RS-232 data cables.
9	**What is Bluetooth? (Nov 2015)** BTL 1 Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz) from fixed and mobile devices, and building personal area networks (PANs). Invented by telecom vendor Ericsson in 1994, it was originally conceived as a wireless alternative to RS-232 data cables.
10	**What are the various handovers carried out in GSM? (Nov 2015)** BTL 2 Within the GSM system there are four types of handover that can be performed for GSM only systems: i) Intra-BTS handover: This form of GSM handover occurs if it is required to change the frequency or slot being used by a mobile because of interference, or other reasons ii) Inter-BTS Intra BSC handover: This form of GSM handover or GSM handoff occurs when the mobile moves out of the coverage area of one BTS but into another controlled by the same BSC iii) Inter-BSC handover: When the mobile moves out of the range of cells controlled by one BSC, a more involved form of handover has to be performed, handing over not only from

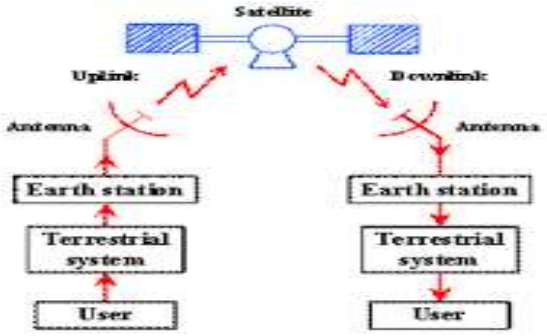
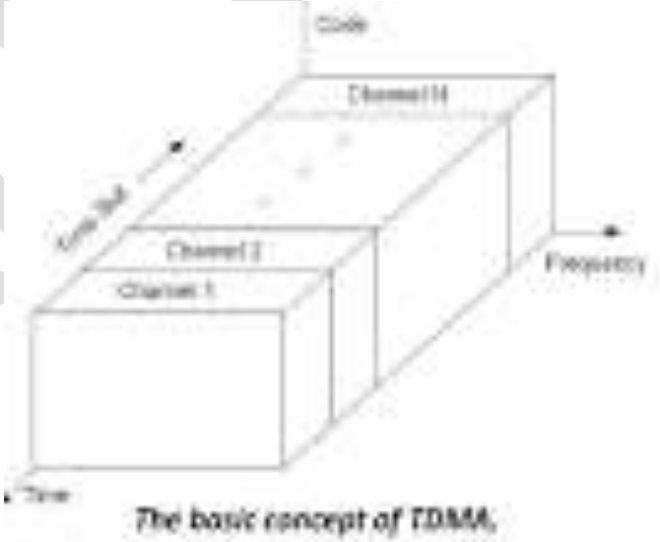
	one BTS to another but one BSC to another. iv) Inter-MSC handover: This form of handover occurs when changing between networks. The two MSCs involved negotiate to control the handover.
11	**What is near far effect in a CDMA system? (May 2015)** BTL 2 The near–far problem or hearability problem is a situation that is common in wireless communication systems, in particular, CDMA. The near–far problem is a condition in which a receiver captures a strong signal and thereby makes it impossible for the receiver to detect a weaker signal.
12	**Define the term frequency reuse factor in a cellular communication system. (May 2015)** BTL 1 Frequency reuse is the process of using the same radio frequencies on radio transmitter sites within a geographic area that are separated by sufficient distance to cause minimal interference with each other. Frequency reuse allows for a dramatic increase in the number of customers that can be served (capacity) within a geographic area on a limited amount of radio spectrum (limited number of radio channels). Cell Frequency Reuse Cluster [Invalid]: In this cluster, the allocated band is divided into 2 bands and the two sub-bands are reused in an alternating fashion somehow. Clearly, only two cells with the same frequency as a particular cell are at equal distance from it. 
13	**What is handoff? (Nov 2014)** BTL 1 Handoff is the procedure for changing the assignment of a mobile unit from one BS to another as the mobile unit moves from one cell to another. Handoff is the principle used to continue that call established in mobile communication. When subscriber is towards the cell boundary the signal strength reduces by which BTS of that cell handover the call to the net BTS of another cell where the subscriber enters.
14	**Why are hexagons employed to model coverage areas of mobile communication? (Nov 2014)** BTL 4 Hexagonal shapes are preferred than square or circle in cellular architecture because it covers an entire area without overlapping. We can argue that even square shaped system does not overlap. It is because it requires fewer cells to represent a hexagon than triangle or square. Other advantages of hexagonal cellular system: i) The frequency reuse become possible using this shape. ii) The radiation pattern of the antenna is 60 degree which means 6 are required for the full 360 degree coverage which is the same no. of sides as the hexagon(equidistant antennae). iii) Minimum interference.

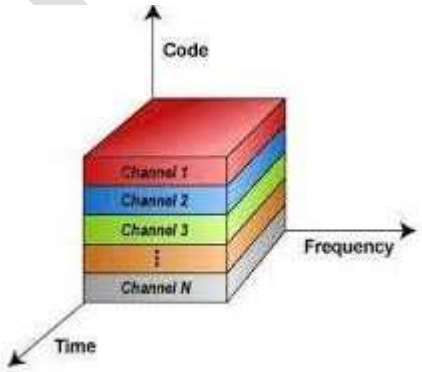
	
15	**What is a chip code in CDMA system? (May 2013)** BTL 1 The chip rate of a code is the number of pulses per second (chips per second) at which the code is transmitted (or received). The chip rate is larger than the symbol rate, meaning that one symbol is represented by multiple chips. The ratio is known as the spreading factor (SF) or processing gain: $SF=\frac{chip\ rate}{symbol\ rate}$
16	**Distinguish between FDMA and TDMA. (May 2013)** BTL 2 FDMA is a channel access method used in multiple-access protocols as a channelization protocol. FDMA gives users an individual allocation of one or several frequency bands, or channels. It is particularly commonplace in satellite communication. FDMA, like other multiple access systems, coordinates access between multiple users. Alternatives include TDMA, CDMA, or SDMA. Time-division multiple access (TDMA) is a channel access method for shared-medium networks. It allows several users to share the same frequency channel by dividing the signal into different time slots. The users transmit in rapid succession, one after the other, each using its own time slot. This allows multiple stations to share the same transmission medium (e.g. radio frequency channel) while using only a part of its channel capacity.
17	**What is CDMA? (Nov 2011)** BTL 1 Code Division Multiple Access systems use codes with certain characteristics to separate different users. To enable access to the shared medium without interference, the users use the same frequency and time to transmit data. The main problem is to find good codes and to separate this signal from noise. The good code can be found the following 2 characteristic i) Orthogonal & ii) Autocorrelation.
18	**Specify the security services offered by GSM.** BTL 1 i) Access control and authentication ii) Confidentiality iii) Anonymity
19	**What is GSM?** BTL 1 The primary goal of GSM (Global System for Mobile communication) was provide a mobile phone system that allows users to roam and provides voice services compatible to ISDN and other PSTN systems.

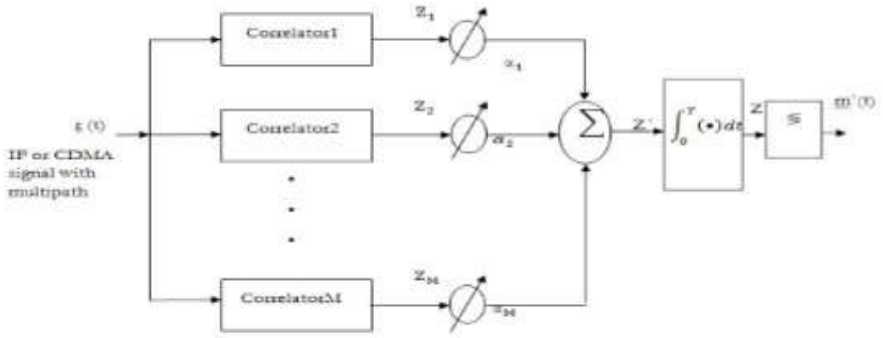
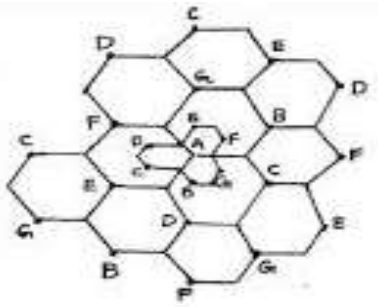
20	**What is the advantage of cell splitting concept?** BTL 2 In this technique the total number of users at a time will be increased so that traffic congestion in the area can be avoided. Also a set of lowest power antennas can be used in place of single antenna for the entire region so that the cellular coverage efficiency will be definitely high.
21	**What are the advantages of cellular systems?** BTL 2 i) Higher Capacity ii) Less transmission Power iii) Local interference only **iv)** Robustness
22	**State the laws of planetary motion.** BTL 1 Kepler's law may be simply stated as i) The planets move in ellipses with the sun at one focus, ii) The line joining the sun and a planet sweeps out equal areas in equal intervals of time, iii) The square of the time of revolution of a planet divided by the cube of its mean distance from the sun gives a number that is the same for all planets.
23	**Define geostationary orbit.** BTL 1 The circular equatorial orbit is exactly in the plane of equator on earth. All the points in this orbit are at equal distance from earth surface, and a satellite in this orbit appears to be stationary to the point of earth. Therefore this orbit is called geostationary orbit.
24	**Define angle of inclination and angle of elevation.** BTL 1 **Angle of inclination:** It is the angle between the earth's equatorial plane and the orbital plane of a satellite measured counter clockwise at the point in the orbit where it crosses the equatorial plane traveling from south to north. **Angle of elevation:** It is the vertical angle formed between the direction of travel of an electromagnetic wave radiated from an earth station antenna pointing directly toward a satellite and the horizontal plane.
25	**List the applications of a satellite.** BTL 1 i) Some of the applications of a satellite are: ii) Surveillance or observation iii) Navigation iv) TV broadcast v) Satellite telephones
	PART*B
1	**Discuss in detail about the architecture of GSM with necessary diagram. (7M) (Nov 2017) (May 2017) (May 2016) (Nov 2015) (May 2015)** BTL 2 **Describe the working of global system for mobile communication.** **Answer: Page:520-W.Tomasi** **GSM**: Standard developed - European Telecommunications Standards Institute (ETSI) - describe - protocols - second-generation digital cellular networks -used by mobile devices. (1M) **Services:** GSM Services (2M) Bearer data service Tele service

	Supplementary service **GSM architecture with diagram**: (2M)  **Important stations:** (2M) BSS- Base station subsystem BTS- Base transceiver station MS- Mobile station BSC-Base station controller
2	**Describe the concept of frequency reuse, channel assignment and hand-off in a cellular system. (7M) (Nov 2017) (Nov 2014)** BTL 2 **Briefly explain the concept of frequency reuse and channel assignment in CDMA. (7M) (Nov 2016)** **Explain the concept of cellular topology and cell fundamentals with examples. (7M) (Nov 2015)** **Answer: Page:515-W.Tomasi** **Cell sectorization**:Cells - divided into sectors - provides smaller coverage area-causes more frequency reuse. (1M) **Channel assignment**: (2M) i) Fixed channel assignment: each cell – allocated predetermined set-various channels. ii) Dynamic channel assignment: each time - when cell request made- serving base station request - channel- from MSC. Switch - allocates channel - to requested cell. **Frequency reuse**: Allocating channel- all base station. (2M)  **Handoff**: Changing - mobile unit assignment - from one BS to another -as mobile unit moves- from one cell to another. (2M)
3	**Briefly discuss about the different multiple access schemes. (6M) (Nov 2017) (May 2016)**BTL 4 **Write a note on multiple access technique. (6M) (May 2014)**

	Answer: Page:522.W.Tomasi **Multiple access technique**: Many subscribers - share- communication channel - same time.(1M) **Types** : (2M) TDMA CDMA FDMA **TDMA**: Each user - allocated - full spectral occupancy- only for short duration of time. (1M) **FDMA**: Sub band of frequencies - allocated - different users. (1M) **CDMA**: Separates users - assigning - digital codes - broad range - radio frequency. (1M)
4	**Explain in detail about the function of each layer in Bluetooth system. (13M) (May 2017)** BTL 2 **Briefly explain about the Bluetooth technology. (7M) (Nov 2016)** **Answer: Page-245-Notes** **Bluetooth**: Global standard - wireless connectivity. (2M) **Diagram**: (2M) **Explanation:** (6M) **Radio layer:** corresponds - physical layer - OSI model. **Baseband layer**: equivalent - MAC sub layer - LAN'S. **Logical Link, control adaptation layer (L2CAP):** equivalent - logical link control sub layer - LAN.  **Applications**: (3M) Hands-free voice communication -with headset. Dial up networking. Providing communication- between peripheral devices- like wireless mouse or keyboard.
5	**Explain the principle of working of satellite communication with block diagram. (7M) (Nov 2016) (May 2015)** BTL 2 **Describe the concepts of satellite communication.** **Answer: Page:522-W.Tomasi** **Satellite communication**: Satellites- relay stations -earth sources. (2M) **Block diagram:** (1M) **Explanation**: (4M)

	Uplink model: signal transmitted -from earth station -to satellite **Transponder**: transmitter-receiver combination **Downlink model**: retransmitted signal -from satellite -to receiving station  Figure 1. Block diagram of satellite communication
6	**Explain the use of TDMA in wireless communication system. (7M) (May 2014)** BTL 2 **Answer: Page:515.W.Tomasi** **Multiple access**: Many subscribers - share -communication channel - same time. (1M) **TDMA**: Each user – allocated - full spectral occupancy- short duration. (1M) **Diagram**: (1M) **Explanation:** (3M) Guard times inserted-between-assigned time slots Reduce interference -between users Requires- time synchronization Interchannel interference-negligible **Features:** (1M) Shares single frequency-several users Duplexers not required Data transmission-not continuous  The basic concept of TDMA.

	PART*C
1	**Compare different multiple access techniques with its advantages and disadvantages.(15M)** BTL4 **Answer: Page:516.W.Tomasi** **Multiple access technique**: Many subscribers - share- communication channel - same time. (2M) **Types** : (3M) TDMA CDMA FDMA **TDMA**: Each user - allocated - full spectral occupancy- only for short duration of time. (2M) **FDMA**: Sub band of frequencies - allocated - different users. (2M) **CDMA**: Separates users - assigning - digital codes - broad range - radio frequency. (2M) **Advantages:** (2M) FDMA: Uses low bit rates TDMA: Efficient use - bandwidth CDMA: Support many users-same channel-high capacity **Disadvantages:**(2M) FDMA: Bit rate- per channel-fixed-small TDMA: Requires -Synchronization CDMA: Careful selection - code length
2	**Explain in detail about the techniques of CDMA with its features. Write down the problems faced by CDMA. (15M)** BTL2 **Answer: Page:514-W.Tomasi** **CDMA**: Separates users - assigning digital codes -within - broad range- radio frequency. (2M)  **Features of CDMA** : (2M)

	Near far problem: Receiver captures - strong signal -makes - impossible for receiver- to detect weaker signal. (3M) **Rake receiver**: counter - effects - multipath fading. (4M)  Fig: RAKE Receiver **Advantages**: (2M) i) Can support many users ii)Low transmit power **Disadvantages:** (2M) i) Careful selection - code length ii)More handoff possibility
3	**Briefly discuss the process of channel assignment in cellular system. (15M)** BTL 2 **Answer: Page:518.W.Tomasi** **Cell sectorization**:Cells - divided into sectors - provides smaller coverage area-causes more frequency reuse. (4M) **Channel assignment**: (3M) i) Fixed channel assignment: each cell – allocated predetermined set-various channels. ii) Dynamic channel assignment: each time - when cell request made- serving base station request - channel- from MSC. Switch - allocates channel - to requested cell. **Frequency reuse**: Allocating channel- all base station. (5M)  **Handoff**: Changing - mobile unit assignment - from one BS to another -as mobile unit moves- from one cell to another. (3M)