JEPPIAAR INSTITUTE OF TECHNOLOGY

"Self-Belief | Self Discipline | Self Respect"

QUESTION BANK

II YEAR CSE-03RD SEMESTER

Academic year: 2019-2020

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 disciplinedhigh 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 liv**es.**

Department Vision

To produce Engineers with visionary knowledge in the field of Computer Science and Engineering through scientific and practical education in stance of inventive, modern and communal purpose for the improvement of society.

Department Mission

M1: Devise students for technical and operational excellence, upgrade them as competentengineers and entrepreneurs for country's development.

M2: Develop the standard for higher studies and perpetual learning through creative and critical thinking for the effective use of emerging technologies with a supportive infrastructure.

M3: Involve in a constructive, team oriented environment and transfer knowledge to balance the industry-institute interaction.

M4: Enrich students with professional integrity and ethical standards that will make them deal social challenges successfully in their life.

Program Educational Objectives (PEOs)

PEO 1: To support students with substantial knowledge for developing and resolving mathematical, scientific and engineering problems.

PEO 2: To provide students with adequate training and opportunities to work as a collaborator
with informative and administrative qualities.

PEO 3: To motivate students for extensive learning to prepare them for graduate studies, R&D
and competitive exams.

PEO 4: To cater students with industrial exposure in an endeavour to succeed in the emerging cutting edge technologies.

PEO 5: To shape students with principled values and to follow the code of ethics in social and professional life.

Program Specific Outcomes (PSOs)

PSO 1 : Students are able to analyse, design, implement and test any software with the programming and testing skills they have acquired.

PSO 2: Students are able to design and develop algorithms for real time problems, scientific and business applications through analytical, logical and problems solving skills.

PSO 3: Students are able to provide security solution for network components and data storage and management which will enable them to work efficiently in the industry.

BLOOM'S TAXONOMY

Definition:

- A theory to identify cognitive levels (Levels of thinking)
- Represents the full range of cognitive functions.

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 is able to recall, restate and remember learned information.
- **BTL 2** – **Understand -** The learner grasps the meaning of information by interpreting and translating what has been learned.
- **BTL 3** – **Apply -** The learner makes use of information in a context similar to the one in which it was learned.
- **BTL 4** – **Analyze -** The learner breaks learned information into its parts to best understand that information.
- **BTL 5** – **Evaluate -** The learner makes decisions based on in-depth reflection, criticism and assessment.
- **BTL 6** – **Create -** The learner creates new ideas and information using what has been previously learned.

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: 60OUTCOMES:

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 $P \to (Q \vee \neg R)$
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 $p \to q$ **CONTRAPOSITIVE:** $\neg q \to \neg p$ " If I do not buy an umbrella then there is no rain" **CONVERSE :** $q \to p$ "If I buy an umbrella then there is rain" **INVERSE:** $\neg p \to \neg q$ "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 $p \to q$ **CONTRAPOSITIVE:** $\neg q \to \neg p$ "If the home team does not win, then it is not raining" **CONVERSE :** $q \to p$ "If it is raining then the home team wins" **INVERSE:** $\neg p \to \neg q$ "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** $(P \to (Q \to R)) \to ((P \to Q) \to (P \to R)$ **is a tautology.(BTL3)**

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

P	Q	R	$Q \to R$	$P \to Q$	$P \to R$	$P \to (Q \to R)$	$(P \to Q) \to (P \to R)$	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** $(p \to q) \wedge (r \to q)$ **and** $(p \vee r) \to q$ **are logically equivalent. (BTL 5)**

P	q	R	$p \to q$	$r \to q$	$p \vee r$	$(p \to q) \wedge (r \to q)$	$(p \vee r) \to q$
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 $p \to (q \to p) \Leftrightarrow \neg p \to (p \to \neg q)$. (BTL3)**

L.H.S $\Leftrightarrow p \to (q \to p)$

$\Leftrightarrow \neg p \vee (q \to p)$ 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
$\Leftrightarrow \neg p \to (\neg p \vee \neg q)$ Implication law
$\Leftrightarrow \neg p \to (p \to \neg q)$ 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: $\forall x(P(x) \to Q(x))$
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) : $(x^2 \le x)$ **Given:** $\forall x(x^2 > x)$ Its negation is $\neg[\forall x(x^2 > x)] \Leftrightarrow \exists x(x^2 \le x)$ Let P(x) : $(x^2 = 2)$ $\neg$P(x) : $(x^2 \ne 2)$ **Given:** $\exists x(x^2 = 2)$ Its negation is $\forall x \neg(x^2 = 2) \Leftrightarrow \forall x(x^2 \ne 2)$
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 $\neg[(\exists x)(\forall y)\,p(x,y)] \Leftrightarrow (\forall x)(\exists y)\,p(x,y)$.
22.	**Given P={2,3,4,5} , state the truth value of the statement $(\exists x \in P)(x+3=10)$. (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 $\forall x(x^2 \geq x)$ 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 :** $(x^2 \geq x)$ $\Leftrightarrow x^2 - x = x(x-1) \geq 0$ Consequently $(x^2 \geq x)$ 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 $\forall x(x^2 \geq x)$ 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 $\forall x(x^2 \geq x)$ is true if the universe of discourse consists of all integers
24.	**Let P(x) denote the statement $x \leq 4$. Write the truth values of P(2) and P(6) . (BTL2)** P(x) : $x \leq 4$. When x=2 , P(2): $2 \leq 4$, which is true When x=6 , P(6): $6 \leq 4$, 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 $\rightarrow$ 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** $C \vee D, C \vee D \to \neg H, \neg H \to (A \wedge \neg B), (A \wedge \neg B) \to (R \vee S)$ **(BTL5) (8 Marks)** (Refer SKD, Pg.1.69) Keypoints: • $C \vee D \to H$ (2marks) • $C \vee D \to (A \wedge \neg B)$ (2marks) • $C \vee D \to (R \vee S)$(2marks) • Get the answer as $R \vee S$ (2marks)
5	**Show that the premises $P \to Q, Q \to R, R \to S, S \to \neg R$ and $P \wedge S$ are inconsistent. (Nov2015). (BTL5) (8 Marks)** (Refer SKD Pg. 1.81) Keypoints: • $P \to R$ (2marks) • $R \to \neg S$ (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 donot 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) • $\neg P \wedge \neg Q,\ R \to P,\ \neg R \to S,\ S \to T \Rightarrow T$ (2marks) • $\neg P, R \to P \Rightarrow \neg R$ (2marks) • $\neg R, \neg R \to S \Rightarrow S$ (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) • $(\neg P \vee \neg Q) \to (R \wedge S),\ R \to T, \neg T \Rightarrow P$ (2marks) • Use rules of inferences to the necessary premises(4marks) • $\neg R, \neg R \to P \Rightarrow P$. (1mark)
11	**Show that $R \to S$ is logically derived from the premises $P \to (Q \to S)$, $\neg R \vee P$ and Q. (Apr2017, Nov2015, May2016) (BTL3) (8 Marks)** Keypoints: • R is an additional premise(2marks) • $R, \neg R \vee P \Rightarrow P$ (2marks) • $P, P \to (Q \to S) \Rightarrow Q \to S$ (3marks) • Get the answer as S(1mark)
12	**Show that $(p \to q) \wedge (r \to s), (q \to t) \wedge (s \to u), \neg(t \wedge u), (p \to r) \Rightarrow \neg p$. (Apr 2015)(BTL3) (8 Marks)** Keypoints: • $(p \to q), (q \to m) \Rightarrow p \to m$ (2marks) • $(p \to r), (r \to n) \Rightarrow p \to n$ (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 $\forall x(P(x) \to Q(x)) \wedge (Q(x) \to R(x)) \Rightarrow \forall x(P(x) \to Q(x))$. (Nov2016) (BTL3) (8 Marks)** (Refer Balaji Pg.1.145) Keypoints: • $P(y) \to Q(y)$ (2marks) • $P(y) \to R(y)$ (4marks) • $\forall x(P(x) \to Q(x))$ (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: • $\exists x(P(x) \wedge Q(x)), \forall x(Q(x) \to R(x)) \Rightarrow \exists x(P(x) \wedge R(x))$ (2marks) • $P(a) \wedge Q(a)\ Q(a) \to R(a)$ (2marks) • $P(a) \wedge Q(a)$ (3marks) • $\exists x(P(x) \wedge R(x))$ (1mark)
17	**Prove that $\sqrt{2}$ is irrational by giving a proof by contradiction. (Nov 2013, May2016) (8 Marks)**

	(Refer SKD Pg. 1.78) **(BTL5)** Keypoints: • Use indirect method , Assume $\sqrt{2}$ is irrational (2marks) • $\sqrt{2} = \frac{p}{q}$ (2marks) • q = 2k (2marks) • a contradiction that $\sqrt{2}$ 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.) $P(k) = 1+2+3+.....+k = \frac{k(k+1)}{2}$ **Step 3:** To prove P(k+1) is true (i.e.) $P(k+1) = 1+2+3+.....+(k+1) = \frac{(k+1)(k+2)}{2}$ L.H.S = $1+2+3+.....+k+(k+1)$ $= \frac{k(k+1)}{2}+(k+1)$ $= \frac{(k+1)(k+2)}{2}$ 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 :** $y_n = A(3)^n + B(-4)^n$. $y_{n+1} = A(3)^{n+1} + B(-4)^{n+1} = 3A(3)^n - 4B(-4)^n$ $y_{n+2} = A(3)^{n+2} + B(-4)^{n+2} = 9A(3)^n + 16B(-4)^n$ $\begin{vmatrix} y_n & 1 & 1 \\ y_{n+1} & 3 & -4 \\ y_{n+2} & 9 & 16 \end{vmatrix} = 0$ $y_n(48+36) - 1(16y_{n+1} + 4y_{n+2}) + 1(9y_{n+1} - 3y_{n+2}) = 0$ $84y_n - 7y_{n+1} - 7y_{n+2} = 0$ $\Rightarrow 12y_n - y_{n+1} - y_{n+2} = 0$
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, $16 = 32A + 16B$ $2A + B = 1$ ----------(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 $1, a, a^2\ a^3, ...$ (BTL1)** The generating function for the sequence $1, a, a^2\ a^3, ...$ is the infinite series $G(x) = 1 + ax + a^2x^2 + a^3x^3 + ...$ $= \frac{1}{1-ax}$ $if\ \lvert ax \rvert < 1$
18	**Find the closed form generating function of the sequence 2,-2,2,-2,…. (BTL3)** $G(x) = \sum_{n=0}^{\infty} a_n x^n$ $= a_0 + a_1x + a_2x^2 + a_3x^3 + ...$ Generating function $= 2 + (-2)\ x + (2)x^2 + (-2)x^3 + ...$ $= 2[1 - x + x^2 - x^3 + ...]$ $= 2(1-x)^{-1} = \frac{2}{1+x}$
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 $\left(x^0+x^1+x^2+...\right)$ = coefficient of x^{100} in $\left(1+x^1+x^2+...\right)$ = 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 $52C_{13}$ = 635013559600
	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 $\sum_{r=0}^{n}3^r=\frac{3^{n+1}-1}{2}$ (8Marks) (May 2017, May 2016) (BTL5)** (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 by 3(4marks)
5	**Prove that $\frac{1}{\sqrt{1}}+\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{3}}+.....\frac{1}{\sqrt{n}}>\sqrt{n}$, $n\geq 2$, using principle of mathematical induction. (Nov 2016) (BTL5) (8 Marks)** (Refer SKD Pg. 2.42) Keypoints: • Prove for P(2) (i.e.,) $1+\frac{\sqrt{2}}{2}\geq\sqrt{2}$ is true (2marks) • Assume P(k) is true $\frac{1}{\sqrt{1}}+\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{3}}+.....\frac{1}{\sqrt{k}}>\sqrt{k}$, (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) • $G(x) = \dfrac{1+x}{(1-x)(1-3x)}$ (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:

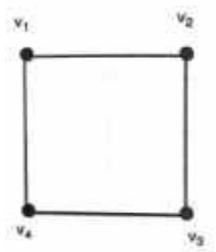
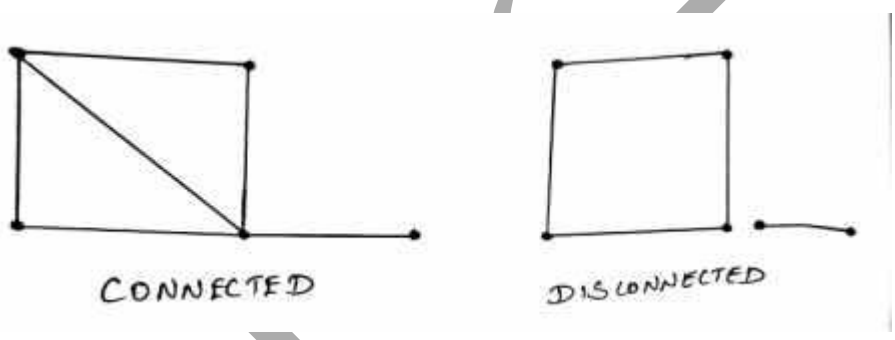
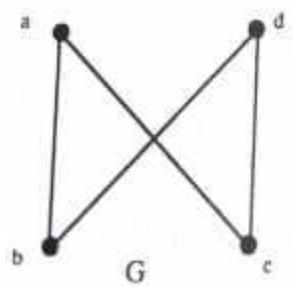
	• $a_{n+1}x^n - 2a_n x^n = 4^n x^n$ (1mark) • $\sum_{n=1}^{\infty} a_{n+1}x^n - 2\sum_{n=1}^{\infty} a_n x^n = \sum_{n=1}^{\infty} 4^n x^n$ (1mark) • Obtain G(x) (1mark) • Solve the obtained equation by partial fractions (3marks) • a_n= coeff of x^n in G(x) (2marks)
11	**Prove that in a group of six 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: • $\left[\frac{m-1}{n}+1\right]$ • 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**

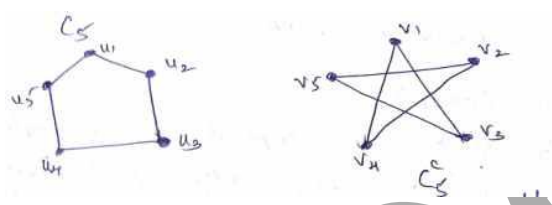
	and 7. (May 2015, Nov 2016,May 2016, May 2018) (BTL4) (8 Marks) (Refer SKD Pg. 2.94) Keypoints: • Use the formula $\frac{n}{P_1P_2}$ where P_1 and P_2 are distinct primes (2marks) • Substitute the values in $\lvert A\cup B\cup C\cup D\rvert$ (4marks) • Number of integers that are not divisible by 2,3,5 and 7 is got by $\overline{\lvert A\cup B\cup C\cup D\rvert}$ (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 $\lvert A\cup B\cup C\rvert=\lvert A\rvert+\lvert B\rvert+\lvert C\rvert-\lvert A\cap B\rvert-\lvert B\cap C\rvert-\lvert A\cap C\rvert+\lvert A\cap B\cap C\rvert$ (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.**

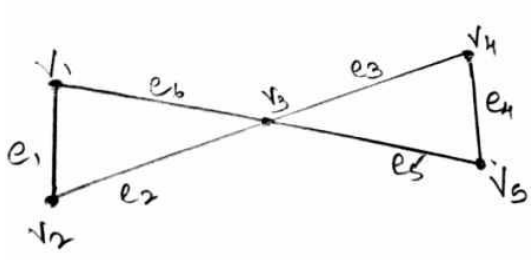
	(Nov 2015, May 2016, May 2017) (BTL4) (8 Marks) Keypoints: • $^{n}C_{r} = \frac{n!}{r!(n-r)!}$ (2marks) • (i) Answer =330ways (2marks) • Answer = 150 (2marks) • Answer = 20 (2marks)
17	**If H_n denote Harmonic numbers, then prove that $H_{2^n} \geq 1 + \frac{n}{2}$. (Nov 2017) (BTL5) (8 Marks)** (Refer 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)**

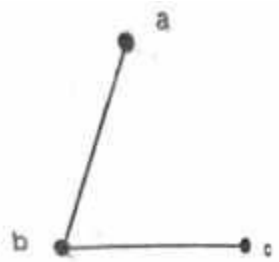
	The degree of a vertex in a graph G is the number of edges incident with it. A loop at a vertex contributes degree 2 to that vertex. Degree of a vertex v is denoted by deg(v).
3	**Show that the sum of the degree of all vertices in G is twice the number of degree in G. (Nov 2012) (BTL1)** Every non-loop 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)**  K_5
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* $K_{3,3}$**.** 

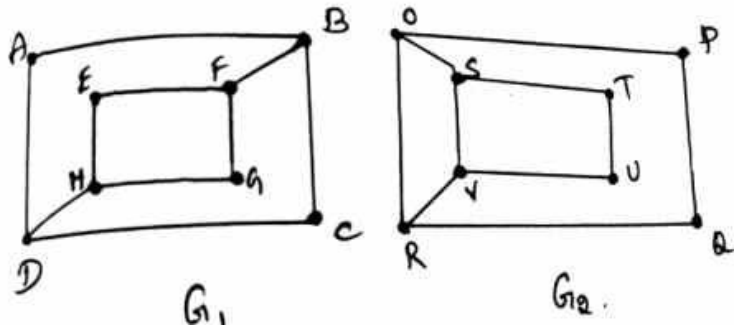
12	**Draw the graph represented by the given adjacency matrix** $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$ **(Nov 2013) (BTL6)** 
13	**Define isomorphism of directed graphs. (Nov 2014) (BTL1)** Two graphs $G_1 = (V_1 E_1)$, $G_2 = (V_2, E_2)$ are 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)** 

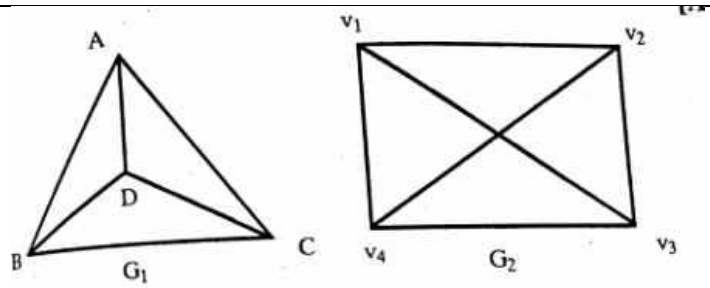
	The adjacency matrix of G is $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ Since 'a' is the first vertex and 'd' is the 4th vertex, the number of paths of length 4 from a to d is (1,4)th element A_4. $A_4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$ 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

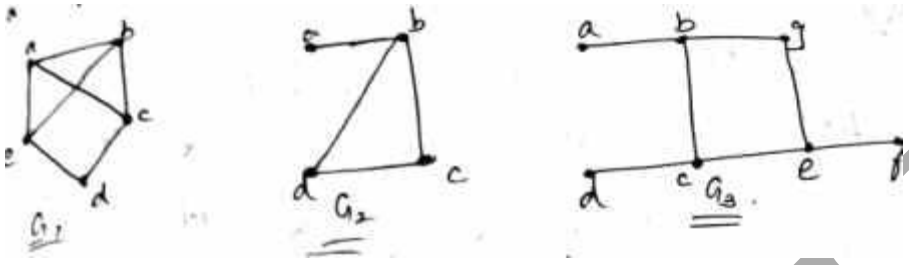
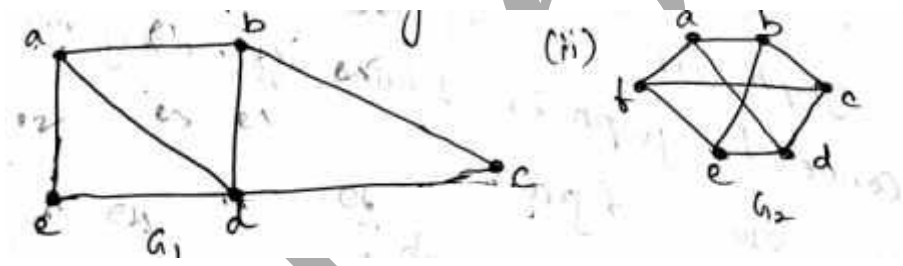
	vertices. If the initial and final vertices of a path are the same then the path is called a cycle or circuit.
25	**Draw the graph represented by the given adjacency matrix and** $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ **(Nov 2016) (BTL6)** 
	Part-B
1	**Prove that the number 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)

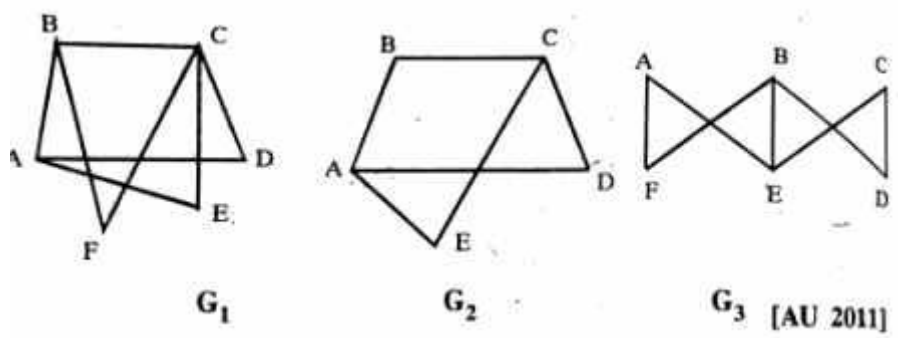
	• $e = \frac{n(n-1)}{2}$ (1mark)
3	**Prove that a simple graph with n vertices and k components cannot have more than $\frac{(n-k)(n-k+1)}{2}$ edges. (Nov 2013 , Nov 2015, May 2015, Nov 2017) (BTL5) (8 Marks)** (Refer Balaji Pg. 3.70) Keypoints: • Consider a simple graph (2marks) • Consider components with k vertices (2marks) • $\lvert E(G)\rvert \le \sum_{i=1}^{k} \frac{n_i(n_i-1)}{2}$ (2marks) • $\lvert E(G)\rvert \le \frac{(n-k)(n-k+1)}{2}$
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, $\lvert E(G)\rvert \le \frac{(n-k)(n-k+1)}{2}$ (2marks) • $\lvert E(G)\rvert > \frac{(n-1)(n-2)}{2}$ (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_2wewrite $G_1 \approx G_2$. (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 connectedcomponents 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: • $\lvert V(G)\rvert = \lvert V(\overline{G})\rvert, \lvert E(G)\rvert = \lvert E(\overline{G})\rvert$ (1mark) • $\lvert E(K_p)\rvert = {}^{p}C_2$ (2marks) • $\lvert E(G)\rvert = \frac{p(p-1)}{2}$ (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** $\langle Z,* \rangle$**. (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

	Now, $a = a*e$ $= a*(a*a^{-1})$ $= (a*a)*a^{-1}$ $= a*a^{-1}$ $= e$ 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$\in$H}**={**[0]+[0], [0]+[3]} = {[0], [3]} = H [1] +H ={1+h / h$\in$H}**={**[1]+[0], [1]+[3]} = {[1], [4]} [2] +H ={2+h / h$\in$H}**={**[2]+[0], [2]+[3]} = {[2], [5]} [3] +H ={3+h / h$\in$H}**={**[3]+[0], [3]+[3]} = {[3], [0]} = H [4] +H ={4+h / h$\in$H}**={**[4]+[0], [4]+[3]} = {[4], [1]} [5] +H ={5+h / h$\in$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) $\bullet$ 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 $\in$ 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 $a.0 = a.(0+0)$ $= a.0 + a.0$ $\Rightarrow a.0 + 0 = a.0 + a.0$ $\Rightarrow \quad 0 = a.0$ 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 $G=\langle a\rangle$ 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 $\lvert AB\rvert=\lvert A\rvert\lvert B\rvert$ (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**

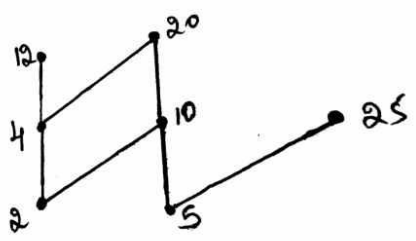
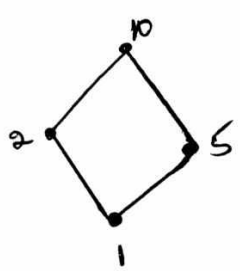
	2018) (BTL5)(8 Marks) (Refer SKD Pg.4.17) Keypoints: • Closure $a*b \in G$ (1mark) • Associative $a*(b*c)=(a*b)*c$ (2marks) • Identity e=2 (2marks) • Inverse $\frac{4}{a}$ (2marks) • Commutative a*b =b*a (1mark)
3	**Prove that** $\left\{\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix},\begin{bmatrix}-1 & 0\\0 & 1\end{bmatrix},\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix},\begin{bmatrix}-1 & 0\\0 & -1\end{bmatrix}\right\}$ **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 $H_1 \subseteq H_2 \; or \; H_2 \subseteq H_1$ (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 $\langle G,* \rangle$ 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)

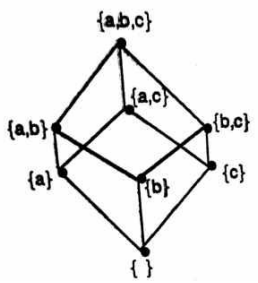
14	**State and prove Cayley's theorem. (May 2013) (BTL5) (8 Marks)** **(**Refer Balaji Pg. 4.59) Keypoints: • " Every finite group of order n is isomorphic to a permutation group of order n" (2marks) • Define a mapping $f : G \to G$ (1mark) • Find 1-1 $f_a(x)=f_a(y) \Rightarrow x=y$ (1 mark) • onto $if\ y \in G\ \ , y = f_a(a^{-1} * y)$ (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 $a \leq c \, then \, a \vee (b \wedge c) = (a \vee b) \wedge c, \forall a,b,c \in L$
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.) $0 \leq a \leq 1, \forall a \in L$

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)** $(i)\ (a+b)' = a'.b'$ $(ii)(a.b)' = a' + b' \quad , \forall a, b \in B$
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 $\langle x, y\rangle \in R$ 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. $\therefore R\{\langle 1,4\rangle, \langle 2,5\rangle, \langle 3,6\rangle\}$
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 $0 \leq x \leq 1$ 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) $a\vee(b\wedge c)=(a\vee b)\wedge(a\vee c)$ $=(a\vee b)\wedge c \qquad (by\,(2)$ 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)** $a \wedge \bar{b} = 0$ **(iii)** $\bar{a} \vee b = 1$ **(iv)** $\bar{b} \le \bar{a}$ **. (May2016, May2017 Nov 2017) (BTL5)(8 Marks)** (Refer Balaji 5.25) Keypoints:

	• $a \le b \Rightarrow a \wedge \bar{b} = 0$ (2marks) • $a \wedge \bar{b} = 0 \Rightarrow \bar{a} \vee b = 1$ (2marks) • $\bar{a} \vee b = 1 \Rightarrow \bar{b} \le \bar{a}$ (2marks) • $\bar{b} \le \bar{a} \Rightarrow a \le b$ (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) • Antisymmetric : 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 sublattices 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)

CS8351 DIGITAL PRINCIPLES & SYSTEM DESIGN L T P C

3 00 3

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 GatesTheorems 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:

On Completion of the course, the students 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 BOOK:

1. M. Morris R. Mano, Michael D. Ciletti, "Digital Design: With an Introduction to the
Verilog HDL, VHDL, and SystemVerilog", 6th Edition, Pearson Education, 2017.

REFERENCES:

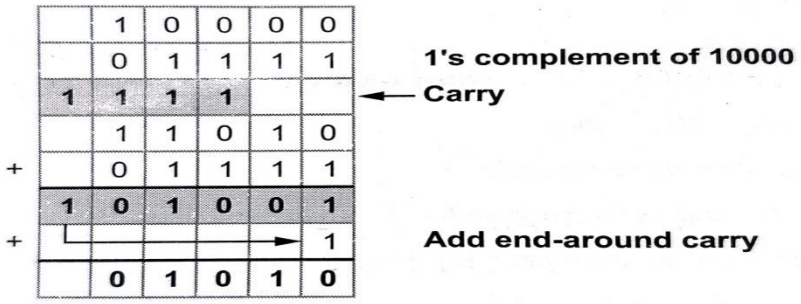
1. G. K. Kharate, Digital Electronics, Oxford University Press, 2010
2. John F. Wakerly, Digital Design Principles and Practices, Fifth Edition, Pearson Education, 2017.
3. Charles H. Roth Jr, Larry L. Kinney, Fundamentals of Logic Design, Sixth Edition, CENGAGE Learning, 2013
4. Donald D. Givone, Digital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

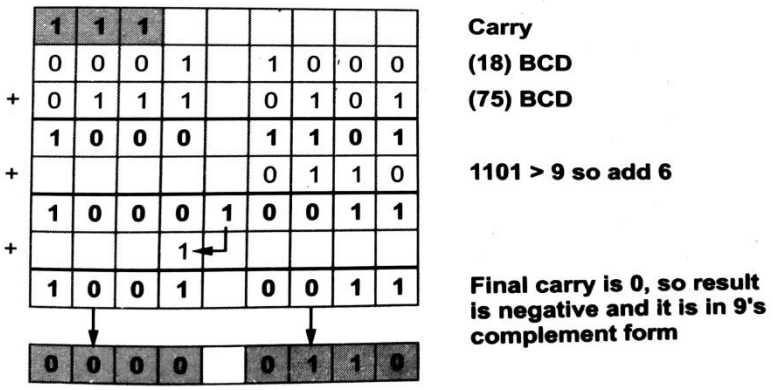
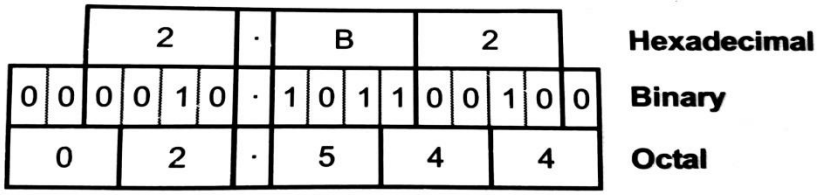
Subject Code:CS8351 **Year/Semester: II /03**
Subject Name: DPSD **Subject Handler: D.Joshua Jeyasekar**

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.	Questions
1.	**What is meant by weighted and non – weighted coding?** (BTL 1) **Weighted Codes:** In Weighted Codes, each digit position of the number represents a specific weight. For example, in decimal code, if number is 567 then weight of 5 is 100, weight of 6 is 10 and weight of 7 is one. In weighted binary codes each digit has a weight 8,4, 2 or 1. **Non – Weighted Codes**: Non – weighted codes are not assigned with any weight to each digit position within the number is not assigned fixed value. Excess – 3 and gray codes are non – weighted codes.
2	**What are the different ways to represent a negative number?** (BTL 1) 1. 1's complement representation **2.** 2's complement representation
3	**What is the advantage of gray code over the binary number sequence?** (BTL 1) **I**n Gray code there is only one bit change over the binary number sequence.
4	**Perform subtraction using 1's complement$(11010)_2 - (10000)_2$.** (BTL 2) 
5	**What are error detecting codes?** (BTL - 1) The data along with the extra bits/bits forms the code. C odes which allow only error detection are called error detecting codes.
6	**Perform 9's and 10's complement subtraction between 18 and -24.** (BTL 2)

	Find 9's complement of 24 9's complement of 24 = 99 − 24 = 75 Add 18 and 9's complement of 24 
7	**Convert the $(153.513)_{10}$ to octal** (BTL 5) **(Apr/May 2015)** Divide by 8 the value before decimal point. Multiply by 8 the value after decimal point. $(231.4065)_8$
8	**Find the octal equivalent of hexadecimal number AB.CD** (BTL 5) Convert the given value to its binary equivalent. Convert the binary value to octal. $(231.4065)_8$
9	**Represent the decimal numbers -200 and 200 using 2's complement binary form** (BTL 5) + 200 = 0 1 1 0 0 1 0 0 0 − 200 = 1 0 0 1 1 0 1 1 1 — 1's complement + 1 — Add 1 1 0 0 1 1 1 0 0 0 — 2's complement
10	**Perform the following code conversions** (BTL 2) $(1010.10)_{16} \Rightarrow (?)_2 \Rightarrow (?)_8 \Rightarrow (?)_{10}$ $(100000001)_2 \Rightarrow (10020.02)_8 \Rightarrow (4112.0625)_{10}$
11	**Subtract 11001 from 01101 using 2's complement.** (BTL 2) Find the 1s complement of 11001. Add it with 01101. 1100
12	**Convert $(2.B2)_{16}$ to binary and octal numbers.** (BTL 2) 
13	**State the two absorption properties of Boolean Algebra.** (BTL 1) 1. A+AB=A 2. A(A+B)=A
14	**State the Associative Law of Boolean Algebra.** (BTL 1)

	Law 1 (The Associative Law of Addition) In the ORing of the several variables, the result is the same regardless of the grouping of the variables. For three variables, A ORed with B OR C is the same as A OR B Ored with C. i.e., $A + (B + C) = (A + B) + C$
15	**State the Associative Law of Multiplication.** (BTL 1) *Law 2* (*The Associative Law of Multiplication)* It makes no difference in what order the variables are grouped when ANDing several variables. For three variables, A AND B ANDed with C is the same as A ANDed with B and C. $i.e., (AB)C = A(BC)$
16	**Explain the principle of duality with the help of example.** (BTL 1) The duality theorem states that, starting with a Boolean relation, you can derive another Boolean relation by, 1. Changing each OR sign to an AND sign 2. Changing each AND sign to an OR sign **3.** Complementing any 0 to 1 appearing in the expression Ex: A+0 =A. Using duality theorem, we can say that, A.1=A
17	**State and prove the consensus theorem in Boolean Algebra.** (BTL 1) In Simplification of Boolean expression, an expression of the form AB+A'C+BC the term BC is redundant and can be eliminated to form the equivalent expression AB+A'C. The theorem used for this simplification is known as consensus theorem.
18	**Explain the De Morgan's theorem in Boolean Algebra.** (BTL 1) **(Apr/May 2015)** $\overline{(\boldsymbol{A}+\boldsymbol{B})} = \overline{\boldsymbol{A}}.\overline{\boldsymbol{B}}$ $\overline{(\boldsymbol{A}.\boldsymbol{B})} = \overline{\boldsymbol{A}} + \overline{\boldsymbol{B}}$
19	**Name the two canonical forms for Boolean Algebra.** (BTL 1) 1. Standard SOP and 2. Standard POS forms.
20	**Express $\boldsymbol{F = BC' + AC}$ in a canonical SOP form.** (BTL 3) **F = BC'+AC** **= (A+A')BC'+AC(B+B')** **= ABC'+A'BC'+ABC+AB'C**
21	**Simplify the following Boolean expression to a minimum number of literals.** (BTL 3) **a) (X+Y) (X+Y')** **= XX+XY'+XY+0 = X+X(Y'+Y)** **= X+X = X** **b) XY+X'Z+YZ** **= XY(Z+Z')+X'Z(Y+Y')+YZ(X+X')** **= XYZ+XYZ'+X'YZ+X'Y'Z+XYZ+X'YZ** **= XYZ+XYZ'+X'YZ+X'Y'Z** **= XY(Z+Z')+X'Z(Y+Y')** **= XY+X'Z**
22	**Simplify the following Boolean expression.** (BTL 3) **ab'c'+ab'c+abc** **= ab'c'+ac(b'+b)** **= ab'c'+ac** **= a(b'c'+c)** **= a(b'+c)** **= ab'+ac**
23	**What code is used to label the row headings and column heading of K – map? Why?** (BTL

	1) 1. Gray Code is used to label the rows and columns of K – Map. **2.** In case of Gray code, only one variable changes between two consecutive numbers. This is useful in grouping pair, quad, or octets in K – Maps and thus eliminating variables in the final expression. Hence gray code is used to label the rows and columns of K – Map.
24	**What are Prime Implicants?** (BTL 1) All the implicants of a function determined using a Karnaugh Map is called Prime Implicants.
25	**What are the basic digital logic gates?** (BTL 1) The three basic logic gates are 1. AND gate 2. OR gate 3. NOT gate
	PART * B
1	**Simplify the following Boolean function using 4 variable map. (10M)** (BTL 5) **F(w,x,y,z) = Σ (2,3,10,11,12,13,14,15) (Nov/Dec 2014)** **Answer: Page :2-38 A.P.Godse** **Determining & Grouping**(5M) Check for minterms Determine 4 – variable k- map an essential. Write minterms, literals within, row wise, column wise on the k-map. Group 1's cells. **Identify the Boolean Expression** (5M) Group cells from higher order to lower order. Avoid redundant groups. Write final expression in sop form.
2	**Draw a NAND logic diagram that implements the complements of the following function. (8M)** (BTL 3) **F(A,B,C,D) = Σ (0,1,2,3,4,8,9,12) (Nov/Dec 2014)** **Answer: Page 2-83, A.P.Godse model problem** **Determining & Grouping**(3M) Check for maxterms Determine 4 – variable k- map an essential. Write the maxterms, literals within, row wise, column wise on the k-map. Group the 0's cells. **Identify the Boolean Expression** (2M) Group cells from higher order to lower order. Avoid redundant groups. Write final expression in POS. **Implementation using NAND gates.** (3M) STEP 1: Implement expression using basic logic gates. STEP 2: Introduce NOT, Invert OR, NAND logic gates. STEP 3: change all gates to NAND logic gates.
3	**Using QM method simplify the Boolean expression(13M)** (BTL 3) **f(v,w,x,y,z) = Σ (0,1,4,5,16,17,21,25,29)** **Answer: Page 2-57, A.P.Godse** List minterms in binary form (2M) Arrange minterms according to categories of 1's (2M) Compare each binary number with every term in the next higher category (3M)

	List prime implicants (3M) Select minimum number of prime implicants - must cover all minterms (3M)
4	**Π M(0,1,4,11,13,15) + Π d(5,7,8) and verify the result using K-map method. (13M) (Nov/Dec 2014)** (BTL 3) **Answer: Page 2-46, A.P.Godse** **Determining & Grouping**(6M) Check for maxterms Determine 4 – variable k- map an essential. Write maxterms, literals within, row wise, column wise on the k-map. Group 0's cells. **Identify the Boolean Expression** (7M) Group cells from the higher order to lower order. Avoid redundant groups. Write final expression in POS.
5	**f(A,B,C,D) = Σ m(1,3,4,5,9,10,11) + Σd(6,8) and realize using NAND gates. (13M)** (BTL 5) **Answer: Page No. 2-83, A.P.Godse** See question no 3 in part – c.
6	**F (A,B,C,D) = Σm (0,1,3,4,5,7,10,13,14,15) and realize using NAND gates. (13M)** (BTL 5) **Answer: Page 2-83 A.P.Godse** See question no 3 in part – c.
7	**Simplify the following Boolean expression F = x'y'z' + x'yz +xy'z'+xyz'. (13M)** (BTL 3) **(Apr/May 2015)(Nov/Dec 2018)** **Answer: Page 2-48, A.P.Godse** **Determining (**5M**)** Compare given expression with sum of minterms form. Write standard sop canonical form. **Grouping (**4M**)** Check for minterms Determine 3 – variable k- map an essential. Write minterms, literals within, row wise, column wise on the k-map. Group 1's cells. **Identify the Boolean Expression** (4M) Group cells from higher order to lower order. Avoid redundant groups. Write final expression in SOP.
8	**Using K-map simplifies the following expressions and implements using basic gates.(13M)** (BTL 3) **1. F = Σ (1,3,4,6)** **2. F = Σ (1,3,7,11,15) + d(0,2,5)** **Answer: Page 2-85 A.P.Godse** See question no 3 in part – c.
9	**Simplify using K-map to obtain a minimum POS expression. (7M)**(BTL 3) **(A'+B'+C+D)(A+B'+C+D)(A+B+C+D')(A+B+C'+D') (A'+B+C'+D')(A+B+C'+D)** **Answer: Page 2-44, A.P.Godse** **Determining (**5M**)** Compare given expression with product of maxterms form. Write standard pos canonical form. **Grouping (**4M**)**

	Check for maxterms Determine 4 – variable k- map an essential. Write maxterms literals within, row wise, column wise on the k-map. Group 0's cells. **Identify the Boolean Expression** (4M) Group cells from higher order to lower order. Avoid redundant groups. Write final expression in POS.
10	**Find a Min SOP for f = b'c'd + bcd + acd' + a'b'c + a'bc'd(7M)** (BTL 3) **(Apr/May 2015)(Nov/Dec 2018)** **Answer: Page 2-34,2-44, A.P.Godse** **Determining (**3M**)** Check given expression for all literals. Literal A missing in first, second variable. Literal B missing in third variable. Literal D missing in fourth variable. **Converting** (4M) Multiply (A + A'), (B + B'), (D + D') with respective variable. Write standard canonical form.
11	**Convert the following function into product of Max terms canonical form simplify and implement the same using NAND and NOR.(13M)** (BTL 3) **F (A, B, C) = (A+B') (B+C) (A+C'). (Apr/May 2015)** **Answer: Page 2- 44, A.P.Godse.** **Conversion (**1M**)** Convert given expression to standard canonical form. **Determining & Grouping**(4M) Check for maxterms Determine 4 – variable k- map an essential. Write maxterms, literals within, row wise, column wise on the k-map. Group 0's cells. **Identify the Boolean Expression** (4M) Group cells from higher order to lower order. Avoid redundant groups. Write final expression in POS. **Implementation using NAND & NOR gates.** (4M) STEP 1: Implement expression using basic logic gates. STEP 2: Introduce NOT, Invert OR, NAND logic gates. STEP 3: change all gates to NAND & NOR logic gates.
	PART* C
1	**Simplify the following Boolean function using Tabulation method. (15M)**(BTL 5) **F(w,x,y,z) = Σ (2,3,10,11,12,13,14,15)** **Answer page 2-55, Godse, DPSD notes**. **Tabulate** (8M) Convert given minterms to corresponding binary values. Arrange all minterm values starting from minimum to maximum terms. Arrange terms according to number of 1's from minimum to maximum. Check values bit by bit for grouping. Primitive table (4M) Write terms on left side in corresponding rows.

	Write minterm values in corresponding columns. Check for prime implicants and essential prime implicants. Function's expression. (3M) Write final expression in sum of products term. Avoid repeated terms.
2	**i). Implement Y = (A+C) (A+D') (A+B+C') using NOR gates only (15M)** (BTL 3) **ii) Find a network of AND and OR gate to realize f(a,b,c,d) = Σ m (1,5,6,10,13 ,14)** **Answer page :2-85, 2-74,Godse.** **i)Implementation using NOR gates.** (5M) STEP 1: Implement expression using basic logic gates. STEP 2: Introduce NOT, Invert AND, NOR logic gates. STEP 3: change all gates to NOR logic gates. **ii)Determining & Grouping**(5M) Check for minterms Determine 4 – variable k- map an essential. Write minterms, literals within, row wise, column wise on the k-map. Group 1's cells. **Identify the Boolean Expression** (5M) Group cells from higher order to lower order. Avoid redundant groups. Write final expression in SOP.
3	**Simplify the Boolean function F(A,B,C,D) = Σ m (1,3,5,7,11,12,14,15) + Σd (0,2,5) using K map and implement the same using universal logic gates.(15M)** (BTL 5) **Answer page :2-83, Godse**. **Determining & Grouping**(5M) Check for minterms Determine 4 – variable k- map an essential. Write minterms, literals within, row wise, column wise on k-map. Group 1's cells. **Identify the Boolean Expression** (5M) Group cells from higher order to lower order. Avoid redundant groups. Write final expression in SOP. **Implementation using NAND gates.** (5M) STEP 1: Implement expression using basic logic gates. STEP 2: Introduce NOT, Invert OR, NAND logic gates. STEP 3: change all gates to NAND logic gates.

UNIT II - COMBINATIONAL LOGIC

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.	Questions
1.	**Define Combinational Logic Circuits (May - June 16)** (BTL 1) When logic gates are connected together to produce a specified output for certain specified combinations of input variables, with no storage involved, the resulting circuit is called combinational logic.
2	**Distinguish between combinational logic and sequential logic**.(BTL 1)

Combinational logic circuit	Sequential logic circuit
It consists of input signal, gates and output signals	It consists of a combinational circuit to which memory elements are connected to form a feedback path.
The outputs at any instant of time are entirely dependent upon the inputs present at that time.	The outputs dependent not only on the present input variable but they also depend upon the past value of the input variable.
Combinational circuits are faster in speed	Sequential circuits are slower than the combinational circuits.
Combinational circuits are easy to design	Sequential circuits are comparatively harder to design
Example: Parallel adder, Code converter, Decoder	Example: Serial Adder, Counter, shift register

3	**Define Half Adder and Full Adder**.(BTL1) The logic circuit that performs the addition of two bits is a half adder. The circuitthat performs the addition of three bits is a full adder.
4	**Define Half Subtractor and Full Subtractor**. (BTL 1) The logic circuit that performs the subtraction of two bits is a half adder. The circuitthat performs the subtraction of three bits is a full adder.
5	**What do you mean by carry propagation delay?** (BTL 1) In parallel adders, sum and carry outputs at any stage cannot be produced until the input carry occurs. This time delay in the addition process is called carry propagation delay.
6	**Suggest a solution to overcome the limitation on the speed of an adder.** (BTL 1) It is possible to increase the speed of adder by eliminating the inter – stage carry delay. This method utilizes logic gates to look at the lower – order bits of the augend and addend to see if a higher – order carry is to be generated.
7	**Mention any two uses of HDL.** (BTL1) • When this HDL code is passed through initial synthesis tool, a lower – level description of the circuit is generated as an output. • With this process, a set of logic expressions which describes the logic functions required

	to realize the circuit is produced. • The logic expressions produced by the synthesis tool are not likely to be in an optimal form.
8	**What do you mean by encoder?** (BTL1) An encoder is a digital circuit that performs the inverse operation of a decoder. Encoder has 2^n input lines and n output lines. Encoder has enable inputs to activate encoded outputs.
9	**What is decoder?** (BTL1) Decoder is a multiple - input multiple output logic circuit that converts coded inputs into coded outputs where the input and output codes are different. In a binary decoder n – inputs produce 2^n outputs.
10	**What is data selector? Or What is multiplexer? Or Why is MUX is called as data detector?** (BTL1) A **multiplexer** (or **mux**) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2*n* inputs has *n* select lines, which are used to select which input line to send to the output
11	**Mention the difference between MUX and DEMUX**. (BTL 2) Multiplexer is a data selector, Demultiplexer is a data distributer.
12	**Give an application each for a multiplexer.**(BTL 1) 1. It can be used to realize a Boolean function 2. It can be used in communication systems e.g., time division multiplexing. 3. Data routing 4. Logic function generator 5. Control sequencer 6. Parallel-to-serial converter
13	**Distinguish between a decoder and a Demultiplexer** (BTL2)

Decoder	Demux
A decoder accepts a set of binary inputs and activates only the output that corresponds to that input number.	A Demultiplexer is a circuit that receives information on a single line and transmits this information on one of many output lines
Decoder with enable input is used as Demultiplexer.	Data Distributor

14	**What is a priority encoder?** (BTL1) A priority encoder is an encoder circuit that includes the priority function. In priority encoder, if 2 or more inputs are equal to 1 at the same time, the input having the highest priority will take precedence.
15	**Distinguish between decoder and Encoder**.(BTL2)

Decoder	Encoder
In decoder one of the output lines is activated corresponding to the binary input.	In encoder, the output lines generate the binary code, corresponding to the input value.
Input of the decoder is encoded information presented as n inputs	Input of the encoder is decoded information presented as 2^n inputs producing n possible

	producing 2^n possible outputs.	outputs.
	The input code generally has a fewer bits than the output code.	The input code generally has a more bits than the output code.

16	**Mention the three modeling techniques that can be used for describing a module.**(BTL 1) • Gate – level Modeling • Dataflow modeling • Behavioral modeling
17	**What is gate level modeling?** (BTL 1) In gate level modeling, Verilog uses components or gates to model the system.
18	**What is the difference between behavioral modeling and dataflow modeling?** (BTL1) *Dataflow Modeling:* It describes how the circuit signals flow from the inputs to the outputs. There are some concurrent statements which allow describing the circuit in terms of operations on signals and flow of signals in the circuit. *Behavioral Modeling*: It is possible to directly describe the behavior or the functionality of a circuit.
19	**Implement the Boolean function F = ∑m(1,2,3,7) using 3:8 decoder**. (BTL 2) **Answer: Page: 3 – 46 & 3 – 47 A.P.Godse** Connect function variables as input to the decoder. Logically OR the outputs correspond to present minterms to obtain the output.
20	**How addition and subtraction are done in a parallel adder/subtractor?** (BTL 2) If mode M = 0, then addition is performed. If mode M = 1, then subtraction is performed.
21	**Mention some applications of Decoders.** (BTL 1) 1. Code converters 2. Address decoding 3. BCD to 7-segment decoder
22	**How many 4:1 mux are needed to design 16:1 mux?** (BTL 2) Five 4:1 mux are needed to design 16:1 mux.
23	**What will be the maximum number of outputs for a decoder with a 6 bit data word?** (BTL 1) $2^6 = 64$. 64 outputs for a decoder.
24	**What is a magnitude comparator?** (BTL 1) It is a special combinational circuit designed primarily to compare the relative magnitude of two binary numbers.
25	**Mention the applications of Demultiplexer?** (BTL 1) 1. Used as a decoder. 2. As a data distributor. 3. In time division multiplexing as data separator.
	PART * B
1	**Verilog HDL Code in structural description of a full – adder.(13M)** (BTL 3) **Answer: Page 3 – 108 A.P.Godse** Module full_adder (A,B,Cin,Sum,Cout); (2M) Input A,B,Cin; Output Sum,Cout; (2M) Wire s0,c0,c1;

	Full adder (4M) HA H1 (A,B,S0,C0); HA H2 (S0,Cin,Sum,C1); Or (Cout,C0,C1); Endmodule Module HA (A,B,S,C); (5M) Input (A,B); Output (S,C); Xor (S,A,B); And (C,A,B); Endmodule
2	**Implement the following boolean function with 8:1mux. F = ∑m(0,2,6,10,11,12,13) + d(3,8,14) (Apr/May 2015)** **Answer: page 3-65, Godse.** Refer Que 9 in part B.
3	**Design half subtractor and full subtractor circuit and implement using NAND gates. (13M) (Nov – Dec 2015)** (BTL 4) **Answer: page 3-12 to 3-14, Godse.** • Half Subtractor (2M) • Truth table for Half Subtractor (1M) • K – Map Simplification (2M) • Full Subtractor (2M) • Truth table for Full Subtractor (2M) • K – Map Simplification (1M) • Implementation using NAND gates. (3M)
4	**Elaborate Hardware Description Language. (13M)** (BTL 2) **Answer: page 3-88, Godse.** Specify desired behavior of circuit. (2M) Synthesize the circuit. (2M) Implement the circuit (1M) Test the circuit (1M) HDL - Computer Aided Design tools to design such systems. (1M) 2 main applications – synthesis, simulation (1M) Boolean expressions, logic diagrams, digital circuits represented using HDL. (2M)
5	**Draw and explain the block diagram of 4 – bit parallel adder/subtractor. (8M)** (BTL 2) **Refer pg.no 3-16, Godse.** 4 – bit parallel adder/subtractor (8M) Determine input, output. Consider 4 adder blocks - implementing 4-bit parallel adder. Set cin to be 0 or 1. If cin = 0, addition If cin = 1, subtraction
6	**Construct the 4 – bit adder with look ahead carry adder. (8M) (April – May 2015)** (BTL 5) **Refer pg.no 3-17 to 3-20, Godse.** Full adder circuit (3M) Determine the input, output. Construct the full adder circuit. Logic Diagram - look ahead carry generator (3M) Construct look ahead carry generator by determining the Gi, Pi.

	4 – bit parallel adder with look ahead carry generator (2M) Construct 4 bit adder with A,B, Cin, output S , C0.
7	**Realize F(w, x, y, z) = ΠM(1, 4, 6, 7, 8, 9, 10, 11, 15) using MUX. (13M)** (BTL 4) **Answer: page 3-65, Godse.** Express Boolean functions in maxterm form.(3M) Convert into standard SOP form. (4M) Implement using Implementation table.(3M) Implementation using 8:1 mux.(3M)
8	**With a suitable block diagram explain the operation of BCD adder.(8M)** (BTL 1) **Refer pg.no 3-22, Godse.** Determine the inputs A0,A1,A2,A3 and B0,B1,B2,B3. (2M) Determine 2 4-bit binary adder.(1M) Output would be S0,S1,S2,S3. (2M) If sum greater than 9 then add 6. (1M) Construct K-map for carry.(2M)
9	**Realize F(w, x, y, z) = ∑ (1, 4, 6, 7, 8, 9, 10, 11, 15) using 4 to 1 MUX and 8:1 Mux. (13M)** (BTL 4) **(Nov/Dec 2014)** **Answer: page 3-65, Godse.** Express Boolean functions in minterm form.(3M) Convert into standard SOP form. (4M) Implement using Implementation table.(3M) Implementation using 4:1 mux.(3M)
10	**Implement the following boolean function using 8 to 1 Multiplexer F(A ,B, C, D)= A'BD'+ACD+B'CD+A'C'D. Also implement the function using 16 to 1 Multiplexer. (13M) (May – June 2014)** (BTL 4) **Answer: page 3-65, Godse.** Express Boolean functions in minterm form.(3M) Convert into standard SOP form. (4M) Implement using Implementation table.(3M) Implementation using 8:1 mux.(3M)
	PART* C
1	**Design full adder with inputs 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. (15M) (May – June 2016)** (BTL4) **i. using only Nor Gates** **ii. using two half adders** **Answer: page 3-11, Godse.** Full Adder using NOR gate implementation (8M) Determine the inputs, outputs. Realize using logic gates. Convert to nor, invert, gates. Complete the design - completely converting to Nor gates. Full Adder design using two half adders (7M) Determine the inputs, outputs. Realize using logic gates for half adder.
2	**Implement 1:16 Demultiplexer using 1:4 Demultiplexer and explain decoder (15M)** (BTL 4) **(Nov/Dec 2018)** **Answer page. 3-61, Godse.** Determine the inputs for multiplexers- 16 inputs. (3M)

	Four 4:1 multiplexers required.(3M) Four outputs again multiplexed.(1M) Connect select lines (S1, S0) of 4 mux in parallel.(3M) Connect most significant select lines (S3,S2) to mux 5. (2M) Connect the outputs Y0,Y1,Y2,Y3 to mux 5.(1M) Decoder circuit diagram (2M)
3	**Design a 4 – bit Magnitude comparator using Gates and write a Verilog code. (15M) (Nov – Dec 2014)** (BTL 2). Answer: **page 3-84, 3-86, 3-109, Godse.** • Truth Table. (4M) • If A=B then output A=B is 1. • If A>B then output A>B is 1. • If A<B then output A<B is 1. • Otherwise 0. • K – Map Simplification for A=B, A<B, A>B. (5M) • Logic Diagram (6M) • Implement the given expression using basic gates, universal gates.

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

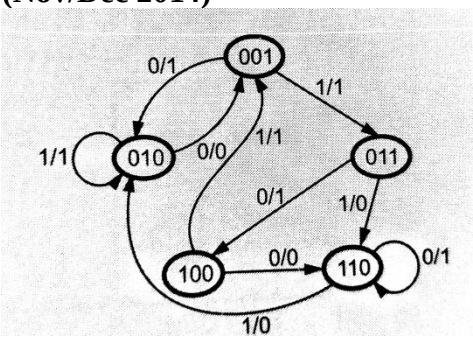
Q.No.	Questions
1.	**Differentiate between Latch and Flip – Flop.** (BTL 1) **LATCHES** / **FLIP – FLOP** A simple latch is the basis for flip flop building / Flip – Flop is built by connecting some additional components around a latch Latch level triggered either positive level or negative level triggered. / Flip – Flop is pulse or clock – edge triggered either positive edge or negative edge triggered. The latch output responds to inputs, until active level is maintained at the enable input. / Flip – Flop responds to inouts only at the specified (positive or negative) edges of clock pulse
2	**Define Flip flop.**(BTL 1) The basic unit for storage is flip flop. A flip-flop maintains its output state either at 1 or 0 until directed by an input signal to change its state.
3	**Give the excitation table for JK Flip – Flop.**(BTL 1) 1. _ 0_0 transition: This can happen when J=0 and K=1 or K=0. 2. _ 0_1 transition: This can happen either when J=1 and K=0 or when J=K=1. 3. _ 1_0 transition: This can happen either when J=0 and K=1 or when J=K=1. 4. _ 1_1 transition: This can happen when K=0 and J=0 or J=1.
4	**Define Shift Registers**. (BTL 1) The binary information in a register can be moved from stage to stage within the register or into or out of the register upon application of clock pulses. This type of bit movement or shifting is essential for certain arithmetic and logic operations used in microprocessors. This gives rise to group of registers called Shift Register.
5	**What are the different types of Shift Registers?**(BTL 1) 1. Serial In Serial Out Shift Register 2. Serial In Parallel Out Shift Register 3. Parallel In Serial Out Shift Register **4.** Parallel In Parallel out Shift Register
6	**Define universal Shift Registers.**(BTL 1) The register which has both shift and parallel load capabilities is referred to as Universal Shift Registers.
	Define a sequential logic circuit. Give an example.(BTL 1) In sequential circuits the output variables dependent not only on the present input variables but they also depend up on the past history of these input variables.
8	**Differentiate between combinational and sequential circuits.**(BTL 1) **<u>Combinational Circuits</u>** Memory unit is not required

	Parallel adder is a combinational Circuit **<u>Sequential circuits</u>** Memory unity is required Serial adder is a sequential circuit
9	**What is the operation of RS flip – flop?** (BTL1) When R input is low and S input is high the Q output of flip-flop is set. When R input is high and S input is low the Q output of flip-flop is reset. When both the inputs R and S are low the output does not change. When both the inputs R and S are high the output is unpredictable.
10	**What is the operation of JK flip – flop?**(BTL1) When K input is low and J input is high the Q output of flip-flop is set. When K input is high and J input is low the Q output of flip-flop is reset. When both the inputs K and J are low the output does not change When both the inputs K and J are high it is possible to set or reset the flip-flop (ie) the output toggle on the next positive clock edge.
11	**What is the operation of D flip – flop?**(BTL1) In D flip-flop during the occurrence of clock pulse if D=1, the output Q is set and if D=0, the output is reset.
12	**What is the operation of T flip – flop?**(BTL1) T flip-flop is also known as Toggle flip-flop. When T=0 there is no change in the output. When T=1 the output switch to the complement state (ie) the output toggles.
13	**What is a master – slave flip – flop?**(BTL1) **(Apr/May 2015)** A master-slave flip-flop consists of two flip-flops where one circuit serves as a master and the other as a slave.
14	**Define race around condition.**(BTL1) In JK flip-flop output is fed back to the input. Therefore change in the output results change in the input. Due to this in the positive half of the clock pulse if both J and K are high then output toggles continuously. This condition is called race around condition".
15	**Define synchronous sequential circuit.**(BTL1) In synchronous sequential circuits, signals can affect the memory elements only at discrete instant of time.
16	**Define State.**(BTL1) The information stored in the memory elements at any given time defines the state at that time of the corresponding sequential circuit.
17	**What do you mean by present state?**(BTL1) The information stored in the memory elements at any given time defines the present state of the sequential circuit.
18	**What do you mean by next state?**(BTL1) The present state and the external inputs determine the outputs and the next state of the sequential circuit.
19	**Define State Table.**(BTL1) For the design of Sequential counters we have to relate present states and next states. The table which represents the relationship between present states and next states is called state table.
20	**Explain about state reduction or Why is state reduction necessary?**(BTL1) State reduction is technique that reduces the number of states in the sequential circuit by keeping only one state for two or more redundant/equivalent states. This reduces the number of required flip – flops and logic gates, reducing the cost of the final circuit. Two states are said to be

	redundant or equivalent, if every possible set of inputs generate exactly same output and same next state.
21	**What is lockout? How it is avoided?**(BTL1) **(Nov/Dec 2014)** In a counter, if the next state of some unused state is again some unused state, it may happen that the counter remains in unused states never to arrive at a used state. Such a condition is called a lockout condition. To avoid lockout, the counter should be provided with an additional logic circuitry which will force the counter from an unused state to the next state as initial state.
22	**What is Mealy Machine?** (BTL1) When the output of the sequential circuit depends on both the present state of flip – flops and on the inputs, the sequential circuit is referred to as Mealy Model.
23	**What is Moore Machine?** (BTL1) When the output of the sequential circuit depends only on the present state of flip – flops the sequential circuit is referred to as Moore Model.
24	**Compare Moore and Mealy models.**(BTL1) **(Nov/Dec 2014)** **MOORE MODEL** / **MEALEY MODEL** Its output is a function of present state only. / Its output is a function of present state as well as present input. Input changes does not affect the output / Input changes may affect the ouput of the circuit. Moore model requires more number of states for implementing same function. / It requires less number of states for implementing same function.
25	**Define state assignment.**(BTL1) The state assignment is a one step in the design of sequential circuits which assign binary values to the states in such a way that it reduces the cost of the combinational circuit that drives the flip – flops.
	PART * B
1	**Explain in detail the operation of a 4 – bit BCD counter. (13M)** (BTL2) **(Nov/Dec 2014) (Apr/May 2015) (Nov/Dec 2018)** **Answer: Page 7-16 Godse.** ➢ Determinating number of flip flops. (3M) ➢ Choosing the flip flop type. (3M) ➢ Write the truth table. (3M) ➢ Derive the reset logic by K – map. (2M) ➢ Logic diagram. (2M)
2	**Explain the operation of 4 – bit Johnson counter. (13M)** (BTL2) **(Nov/Dec 2018)** **Answer: Page 6-17 Godse.** ➢ Determination of flip flop. (3M) ➢ Connect to a common clock. (3M) ➢ Connect the complement output of last flip flop to first flip flop. (4M) ➢ After 8 states the same sequence is repeated. (3M)
3	**Explain the operation of Universal shift registers. (7m)** (BTL2) **Answer: Page 6-10 Godse.**

- Both shifts & parallel load capabilities. (3M)
- 4 flip flops & 4 mux. (4M)
- 2 common selection inputs s1 & s0. (3M)
- Mode control with register operation. (3M)

4 **Design a Synchronous sequential circuit using JK for the given state diagram. (13m)** (BTL4) **(Nov/Dec 2014)**



Answer: Page 5-20 Godse.

- Excitation table (2M)
- K – Map Simplification (4M)
- Logic Diagram (2M)
- Derive Circuit Output and flip – flops considering unused states (2M)
- Logic Diagram (3M)

5 **Design a sequential circuit using RS Flip – Flop for the state table given below using minimum number of flip – flops. (13m)** (BTL4)

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**

Answer: Page 5-20 Godse.

- Minimized state table – 4m
- K – Map Simplification – 6m
- Logic Diagram – 3m

6 **A Synchronous Counter with four JK flip – flops has the following connections: (7M)** (BTL4)

$$J_A = K_A = 1, J_B = Q_A Q_D, K_B = Q_A$$
$$J_C = K_C = Q_A Q_B$$

	$J_D = Q_A Q_B Q_C$ *and* $K_D = Q_A$ **Determine the modulus n of the counter and the output waveforms of the same.** **Answer: Page 5-23 Godse.** ➢ Next state map for JK Flip – Flop (2M) ➢ Transition table (3M) ➢ Output Waveform's (2M)
7	**A sequential circuit with 2D FFs A and B and input X and output Y is specified by the following next state and output equations. (13M)** (BTL4) **(Apr/May 2015)** **A(t+1) = AX + BX** **B(t+1) = A'X** **Y = (A+B)X'** **i) Draw the logic diagram** **ii) Derive the state table** **iii) Derive the state diagram** **Answer: Page 5-13 Godse.** ➢ Logic Diagram (3M) ➢ State table (3M) ➢ Transition Table (3M) ➢ State Diagram (4M)
8	**Convert D Flip – flop to T Flip – Flop. (7m)** (BTL3) **Answer: Page 4-33 Godse.** ➢ Excitation table (3M) ➢ K – Map simplification (2M) ➢ Logic Diagram (2M)
9	**How will you convert a D Flip – flop into JK Flip – Flop? (7M)** (BTL1) **Answer: Page 4-37 Godse.** ➢ Excitation table (3M) ➢ K – Map simplification (2M) ➢ Logic Diagram (2M)
10	**Design a 3 bit synchronous counter using T flip flop. (13M)** (BTL 4) **Answer: Page 7-36, Godse.** ➢ Determine the flip flops. (3M) ➢ Determine the excitation table. (4M) ➢ K map simplification. (3M) ➢ Logic diagram (3M)
	PART* C
1	**Explain the working of 4 – bit synchronous binary up counter. (15M)** (BTL2) **Answer: Page 7-19 Godse.** ➢ Determine the flip flops. (3M) ➢ Determine the excitation table. (4M) ➢ K map simplification. (4M) ➢ Logic diagram (4M)
2	**Design and explain the working of an up – down ripple counter. (15M)** (BTL4) **Answer: Page 7-16 Godse.** ➢ Determinating number of flip flops. (3M) ➢ Choosing the flip flop type. (3M) ➢ Write the truth table. (4M) ➢ Derive the reset logic by K – map. (3M)

	➢ Logic diagram. (2M)
3	**Design a synchronous 3 bit up/down counter using T flip flop. (13M)** (BTL4) **Answer: Page 7-19 Godse.** ➢ Determine the flip flops. (3M) ➢ Determine the excitation table. (4M) ➢ K map simplification. (4M) ➢ Logic diagram (4M)

UNIT IV - ASYNCHRONOUS SEQUENTIAL LOGIC	
Analysis and Design of Asynchronous Sequential Circuits – Reduction of State and Flow Tables – Race-free State Assignment – Hazards.	
PART * A	
Q.No.	**Questions**
1.	**What is an Asynchronous sequential circuit?**(BTL1) The Sequential circuits in which change in input signals can affect memory element at any instant of time are called asynchronous sequential circuits.
2	**How does the operation an asynchronous input differ from that of a synchronous input?**(BTL1) **(Apr/May 2015)** InSynchronous sequential circuit, memory elements are clocked flip – flops. Hence input signals can affect the memory elements only at discrete instants of time. In asynchronous sequential circuits, the memory elements are either unclocked flip – flops or time delay elements. Therefore asynchronous sequential circuits change in input signals can affect the memory at any instant of time.
3	**What are the types of asynchronous circuits?**(BTL1) • Fundamental Mode Circuits • Pulse Mode Circuits
4	**What is a fundamental mode asynchronous sequential circuit?**(BTL1) It assumes that: • The input variables change only when the circuit is stable. • Only one input variable can change at a given time • Inputs are levels and not pulses.
5	**What is pulse mode circuit?**(BTL1) It assumes that • Input variables are pulses instead of levels • The width of the pulse is long enough for the circuit to respond to the input • The pulse width must not be so long that it is still present after the new state is reached. • Pulses should not occur simultaneously on two or more input lines.
6	**Define secondary variable and excitation variables.**(BTL1) The present state and next state variables in asynchronous sequential circuit are called secondary variables and excitation variables respectively.
7	**Define flow table in asynchronous sequential circuit.**(BTL1) Inasynchronous sequential circuit state table is known as flow table because of the behavior of the asynchronous sequential circuit. The stage changes occur in independent of a clock, based on the logic propagation delay, and cause the states to flow from one to another.
8	**Define primitive flow table.**(BTL1) It is defines as a flow table which has exactly one stable state for each row in the table. The design process begins with the construction of primitive flow table.
9	**Define Merger graph.**(BTL1) The merger graph is defines as follows. It contains the same number of vertices as the state table contains states. A line drawn between the two state vertices indicates each compatible state pair. It two states are incompatible no connecting line is drawn.
10	**What is a cycle? Or When does a cycle occur?**(BTL1) A cycle occurs when anasynchronous sequential circuit makes a transition through a series of unstable states. The cycle does not contain a stable state, the circuit will go from one unstable

	state to another, until the inputs are changed.
11	**What are races?**(BTL1) When two or more binary state variable change their value in response to a change in 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.
12	**Define non critical race.**(BTL1) If the final stable state that the circuit reaches does not depend on the order in which the state variable changes, the race condition is not harmful and it is called a non-critical race.
13	**Define critical race.**(BTL1) **(Apr/May 2015)** If the final stable statedepends on the order in which the state variable changes, the race condition is harmful and it is called a critical race.
14	**What are the significant of state assignment?**(BTL1) Synchronous circuits: State assignments are made with the objective of circuit reduction Asynchronous Circuits: Objective is to avoid critical races
15	**What are the different techniques used in state assignments?**(BTL1) There are two techniques used in state assignments. They are • Shared row state assignment • One hot state assignment
16	**What are Hazards?**(BTL1) The unwanted switching transients that may appear at the output of a circuit are called hazards.
17	**Name the types of Hazards.**(BTL1) There are two types of hazards. They are • Static Hazard • Dynamic Hazard
18	**What is static Hazard?**(BTL1) Static Hazard exits if a signal is supposed to remain at particular logic value when an input variable changes its value, but instead the signal undergoes a momentary change in its required value.
19	**What are static – 0 and static – 1 hazard?**(BTL1) In a combinational circuit, if output goes momentarily 0 when it should remain a 1, the hazard is known as static – 1 hazard. On the other hand, if output goes momentarily 1 when it should remain a 0, the hazard known as static – 0 hazard.
20	**Explain dynamic hazard.**(BTL1) The hazard in which output changes three or more times when it should change from 1 to 0 or from 0 to 1 is called dynamic hazard.
21	**What is the cause of essential hazard?**(BTL1) An essential hazard is caused by unequal delays along two or more paths that originate from the same input. Such hazards can be eliminated by adjusting the amount of delays in the affected path.
22	**What are the basic building blocks of an algorithmic state machine chart?** (BTL1) • State Box • Decision Box • Conditional Box
23	**Define state assignment. (**BTL1**)** The state assignment is a one step in the design of sequential circuits which assign binary values to the states in such a way that it reduces the cost of the combinational circuit that drives the flip – flops.

24	**Compare the ASM chart with a conventional flow chart. (**BTL2**)** The ASM chart resembles a conventional flow chart, but is interpreted somewhat differently. A conventional flow chart describes the sequence of procedural steps and decision path for an algorithm without for their time relationship. An ASM chart describes the sequence of events as well as the timing relationship between the states of a sequential controller and the events that occur while going from one state to the next.
25	**Explain about state reduction or Why is state reduction necessary? (**BTL2**)** State reduction is technique that reduces the number of states in the sequential circuit by keeping only one state for two or more redundant/equivalent states. This reduces the number of required flip – flops and logic gates, reducing the cost of the final circuit. Two states are said to be redundant or equivalent, if every possible set of inputs generate exactly same output and same next state.
	PART * B
1	**Illustrate the Types of Asynchronous Sequential Circuits.(7M)** (BTL2) **Answer: page: 9-2 Godse** • Fundamental Mode Circuits (3M) • Only one input change • Inputs levels not pulses • Delay lines as memory elements • Pulse Mode Circuits (4M) • Inputs pulses not levels • Pulse width long • Either complemented or uncomplemented
2	**An asynchronous sequential circuit is described by the following excitation and output function.(13M) (Nov/Dec 14)** (BTL4) $$Y = X_1X_2 + (X_1 + X_2)Y, Z = Y$$ **i) Draw the logic diagram of the circuit** **ii) Derive the transition table and output map** **iii) Describe the behavior of the circuit** **Answer: page: 9-6 Godse** • Logic Diagram (3M) • State table (4M) • Transition Table (3M) • Output Map (3M)
3	**An asynchronous sequential circuit has two internal states and one output. The excitation and output function describing the circuit are as follows. (13M)** (BTL 4) $Y_1 = x_1x_2 + x_1y_2 + x_2y_1$ $Y_2 = x_2 + x_1y_1y_2 + x_1y_1$ $Z = x_2 + y_1$ **Answer: page: 9-7 Godse** • Logic Diagram (3M) • State table (4M) • Transition Table (3M) • Output Map (3M)
4	**Design an asynchronous sequential circuit with two inputs X and Y and with one output Z. Whenever Y is 1, input X is transferred to Z. When Y is 0, the output does not change in X. (13M) (June 16)** (BTL4)

	Answer: page: 9-21 Godse • Draw the state Diagram (3M) • Derive the Primitive Flow Table (3M) • State Assignment (3M) • Realization of circuit using logic elements (2M) • Realization of circuit using SR latch (2M)
5	**Design a two – input (x_1, x_2), two – output (z_1, z_2) fundamental – mode circuit that has the following specifications. When x_1x_2 =00, z_1z_2=00.The output 10 will be produced following the occurrence of the input sequence 00 – 01 – 11. The output will remain at 10 until the input returns to 00 at which it becomes 00. An output of 01 will be produced following the receipt of the input sequence 00 – 10 – 11. And once again, the output will remain at 01 until a 00 input occurs, which returns the output to 00. (13M)** (BTL4) **Answer: page: 9-27 Godse** • Draw the state Diagram (3M) • Derive the Primitive Flow Table (3M) • State Assignment (3M) • K – Map Simplification (2M) • Logic Diagram (2M)
6	**Design a T Flip – flop from logic gates. (7m)** (BTL4) **Answer: page: 9-31 Godse** • Draw the state Diagram (2M) • Derive the Primitive Flow Table (1M) • State Assignment (1M) • K – Map Simplification (2M) • Logic Diagram (1M)
7	**Design a asynchronous D – type latch with two inputs G and D and output Q. Assume fundamental mode of operation. (13M)** (BTL4) **Answer: page: 9-35 Godse** • Draw the state Diagram (3M) • Derive the Primitive Flow Table (3M) • State Assignment (3M) • K – Map Simplification (2M) • Logic Diagram (2M)
8	**What is a hazard? Explain the different types of hazards. What is an essential hazard? Discuss in detail how hazards can be eliminated. (13M)** (BTL1) **(Apr/May 2015)(Nov/Dec 2018)Answer: page: 9-40 Godse** • Hazard – Definition (2M) • Types of Hazards (3M) • Static 0 hazard • Static 1 hazard • Dynamic hazard • Essential Hazard (3M) • Hazard elimination (5M) • Designing a hazard free circuit
9	**Give the hazard – free realization for the Boolean function.** **f(A, B, C, D) = ∑m (0, 2, 6, 7, 8, 10, 12) (8M)** (BTL4) **(Apr/May 2015)** **Answer: page: 9-40 Godse**

	• K – Map Simplification (4M) • Logic Diagram (4M)
	PART* C
1	**Find a static and dynamic hazard free realization for the following function using i) NAND gates ii) NOR gates F(a, b, c, d) = ∑m(1, 5, 7, 14, 15). (15M)** (BTL4) **(Nov/Dec 2018)** **Answer: page: 9-44 Godse** • Circuit realization using NOR gates (7M) • Determination of K map • Grouping of minterms • Implementation of logic gates • Conversion using invert gate • Conversion to NOR universal gate • Circuit Realization gates using NAND gates (8M) • Determination of K map • Grouping of minterms • Implementation of logic gates • Conversion using invert gate • Conversion to NAND universal gate
2	**Write the analysis procedure for an asynchronous fundamental sequential circuit with example. (15M)** (BTL4) **Answer: page: 9-4 Godse** • Determination of next secondary state (3M) • Determination of output equations (3M) • Construct state table (3M) • Construct transition table (3M) • Construct output map (3M)
3	**Write the design procedure for an asynchronous fundamental sequential circuit with example. (15M)** (BTL4) **Answer: page: 9-11 Godse** • Construction of primitive flow table (4M) • Reduction of primitive flow table (4M) • State assignment (4M) • Realization of primitive flow table (3M)

UNIT V – MEMORY DEVICES AND DIGITAL INTEGRATED CIRCUITS	
Basic memory structure – ROM -PROM – EPROM – EEPROM –EAPROM, RAM – Static and dynamic RAM - Programmable Logic Devices – Programmable Logic Array (PLA) - Programmable Array Logic (PAL) – Field Programmable Gate Arrays (FPGA) - Implementation of combinational logic circuits using PLA, PAL.	
PART* A	
Q.No.	**Questions**
1	**Define Memory Cell. Give an example.** BTL1 Memories are made up of registers. Each register consists of storage elements each of which stores one bit of data. Such a storage element us called Memory Cell.
2	**Define Memory Location.** BTL1 Memories are made up of registers. Each register in the memory is one storage location also called memory location. Each memory location is identified by an address.
3	**What is volatile memory? Give example.** BTL1**(Nov/Dec 2014)** The memory which cannot hold data when power is turned off is known as volatile memory. The Static RAM is a volatile memory.
4	**Name the types of ROM.**BTL1 • PROM • EPROM • EEPROM
5	**Define Address and Word.**BTL1 In ROM, each bit combination of the input variable is called an address. Each bit combination that comes out of the output lines is called a word.
6	**Explain ROM.**BTL1 A read only memory (ROM) is a device that includes both the decoder and the OR gates within a single IC package. It consists of n input lines and m output lines. Each bit combination of the input variables is called an address. Each bit combination that comes out of the output lines is called a word. The number of distinct addresses possible with n input variables is 2n.
7	**Define PROM.** BTL1 PROM (Programmable Read Only Memory) It allows user to store data or program. PROMs use the fuses with material like nichrome and polycrystalline. The user can blow these fuses by passing around 20 to 50 mA of current for the period 5 to 20 µs. The blowing of fuses is called programming of ROM. The PROMs are one time programmable. Once programmed, the information is stored permanent.
8	**What is mask – programmable?**BTL1 With a mask programmable PLA, the user must submit a PLA program table to the manufacturer.
9	**Define PLD.**BTL1 Programmable Logic Devices consist of a large array of AND gates and OR gates that can be programmed to achieve specific logic functions.
10	**Give the classification of PLDs.** BTL1 • Programmable Read Only Memory (PROM) • Programmable Logic Array (PLA) • Programmable Array Logic (PAL) • Generic Array Logic (GAL)

11	**What is PLA?**BTL1 PLA stands for Programmable Logic Array, which is LSI component. In PLA both AND and OR gates have fuses at the inputs, therefore in PLA both AND and OR gates are programmable. The outputs from OR gates through fuses as inputs to output inverters so that final output can be programmed as either AND – OR or AND – OR – INVERT
12	**What is the advantage of PLA over PAL?**BTL1**(Nov/Dec 2014)** • Both AND and OR gates are programmable • AND array can be programmed to get desired minterms • Any Boolean functions in SOP can be implemented using PLA.
13	**Why was PAL developed?**BTL1 PAL is PLD that was developed to overcome certain disadvantages of PLA, such as longer delays due to additional fusible links that result from using two programmable arrays and more circuit complexity.
14	**Why the input variables to a PAL are buffered?**BTL1 The input variables to a PAL are buffered to prevent loading by the large number of AND gate inputs to which available output can be connected.
15	**How is individual location in a EEPROM programmed or erased?**BTL1**(Apr/May 2015)** Since it is electrically erasable memory, by activating the particular row and column it is possible that individual can be programmed or erased.
16	**What is write cycle time?**BTL1 It is the minimum time for which an address must be held stable on the address bus, in write cycle.
17	**What is memory cycle?**BTL1 In read/write memory the operation that allows data to be retrieved (Read) and Stored (written) is called the memory cycle.
18	**Compare and contrast static RAM and dynamic RAM.**BTL2

Static RAM	Dynamic RAM
Static RAM contains less memory cells per unit area.	Dynamic RAM contains more cells as compared to static RAM per unit area.
It has less access time hence faster memories.	Its access time is greater than static RAMs.
Static RAM consists of number of flip – flops. Each flip flop stores one bit.	Dynamic RAM stores the data as a charge on the capacitor. It consists of MOSFET and capacitor for each cell.
Cost is more.	Cost is less.

19	**What is access time of a memory?**BTL2 It is the maximum specified time within which a valid new data is put on the data bus after an address is applied.
20	**What is a combinational PLD?** BTL1 PROM is a combinational programmable logic device. A combinational PLD is an integrated circuit with programmable gates divided into an AND array and an OR array to provide an AND – OR sum of product implementation.

21	**Determine the number of address lines for 512 bytes of memory and for a 2kB memory.** BTL1 $2^9 = 512$. 9 address lines required for 512 bytes of memory $2^{11} = 2kB$. 11 address lines required for 2kB memory
22	**What are CPLDs?** BTL1 CPLD is a collection of multiple PLDs and interconnection structure on a single chip. In CPLD, along with individual PLDs the interconnection structure is also programmable.
23	**What is FPGA?** BTL1 In Field Programmable Gate Arrays (FPGA), the word field refers to the ability of gate arrays to be programmed for a specific function by the user instead of by the manufacturer of the device and the word array indicated a series of columns and rows of gates that can be programmed by the end user.
24	**What is LUT?** BTL1 It is the look – up table, used in FPGAs which is a memory device that can be programmed to perform the logic functions.
25	**How the interconnections between logic blocks are done?** BTL1 FPGAs use either SRAM or antifuse methods. Antifuse normally open and shorted when programmed. SRAM method is transistor controlled by the state of on – chip SRAM Cell.
	PART* B
1	**Write a descriptive note on memories.(7M)** BTL1 **Answer Page:5.1-5.9 - D.EdwinDhas** Random Access Memory – Description (1M) Steps required for Read and Write Operations (2M) Timing Waveforms (2M) Memory Classification (2M)
2	**Give the classification of semiconductor memories. (7M)** BTL2 **(Nov/Dec 2014)** **Answer Page:5.2-5.9 - D.EdwinDhas** **Non Volatile Memory** (3M) ROMRead/Write Memory (NVRAM) Mask – programmable ROM EPROM Programmable ROM EEPROM Flash **Volatile Memory** (4M) Read/Write Memory (RWM) Random Access Non – random Access SRAM FIFO DRAM LIFO
3	**Explain in detail about the types and internal diagram of Random Access Memories (RAM). (7M)** BTL2 **(Apr/May 2015)** **Answer Page:5.6-5.9 - D.EdwinDhas** Static RAM (SRAM) (4M) • Logic Diagram • Block Diagram Dynamic RAM (DRAM) (2M) Comparison (1M)
4	**Define ROM Cell. Describe in detail about the types of ROM. (7M)** BTL2 **Answer Page:5.1-5.6 - D.EdwinDhas** Masked ROM (2M)

	PROM (1M) EPROM (2M) EEPROM (2M)
5	**Using ROM realize the following expressions. (7M)** BTL4 $\boldsymbol{F_1(a,b,c) = \sum m(0,1,3,5,7)}$ $\boldsymbol{F_2 = \sum m(1,2,5,6)}$ **Answer Page:5.43-5.44 - D.EdwinDhas** • Block Diagram (2M) • ROM truth Table (3M) • Logic Diagram (2M)
6	**Design a combinational circuit using ROM. The circuit accepts 3 – bit number and generates an output binary number equal to square of input number. (7M)** BTL4 **Answer Page:5.45-5.46 - D.EdwinDhas** • Block Diagram (2M) • ROM truth Table (3M) • Logic Diagram (2M)
7	**Designing a switching circuit that converts a 4 – bit binary code into a 4 – bit gray code using ROM array. (7M)** BTL4 **Answer Page:5.45-5.46 - D.EdwinDhas** • ROM truth Table (3M) • Implementation (4M)
8	**A combinational circuit is defined by the functions:** $\boldsymbol{F_1 = \sum m(3,5,7)}$ $\boldsymbol{F_2 = \sum m(4,5,7)}$ **Implement the circuit with a PLA having 3 inputs, 3 product terms and two outputs. (7M)** BTL4 **Answer Page:5.20-5.21 - D.EdwinDhas** • Simplify the Boolean functions using K – Map (2M) • Write PLA Program table (2M) • Implementation (3M)
9	**Draw a PLA circuit to implement the logic functions** **A'BC+AB'C+AC' and A'B'C'+BC (7M)** BTL3 **Answer Page:5.27-5.28 - D.EdwinDhas** • Simplify the Boolean functions using K – Map (3M) • Implementation (4M)
10	**Implement the following multiboolean function using 3*4*2 PLA and PLD.** $\boldsymbol{f_1(a_2,a_1,a_0) = \sum m(0,1,3,5) and f_2(a_2,a_1,a_0) = \sum m(3,5,7)}$ **(7M)** BTL3 **Answer Page:5.26-5.27 - D.EdwinDhas** • Simplify the Boolean functions using K – Map (3M) • Implementation (4M)
11	**Design a BCD to Excess – 3 code converter and implement using suitable PLA. (13M)** BTL4 **Answer Page:5.30-5.32 - D.EdwinDhas** • Truth table of BCD to Excess – 3 Converter (3M) • Simplify the Boolean functions using K – Map (3M) • Write PLA Programmable Table (3M) • Implementation (4M)
12	**Design and implement 3 – bit binary to gray code converter using PLA. (13M)** BTL4

	(Nov/Dec 2014) **Answer Page:5.51- D.EdwinDhas** • Truth table of BCD to Excess – 3 Converter (4M) • Simplify the Boolean functions using K – Map (4M) • Implementation (5M)
13	**A combinational circuit is defined by the functions,** $\boldsymbol{F_1(A,B,C) = \sum(3,5,6,7)}$ $\boldsymbol{F_2(A,B,C) = \sum(0,2,4,7)}$ **Implement the circuit with a PLA having three inputs, four product terms and two outputs. (13M)** BTL4 **(Apr/May 2015)** **Answer Page:5.45-5.49 - D.EdwinDhas** • Simplify the Boolean functions using K – Map (6M) • Implementation (7M)
14	**Implement the Boolean function with a PLA.** $\boldsymbol{F_1(A,B,C) = \sum(0,1,2,4)}$ $\boldsymbol{F_2(A,B,C) = \sum(0,5,6,7)}$ $\boldsymbol{F_3(A,B,C) = \sum(0,3,5,7)}$ **(10M)** BTL4**(Nov/Dec 2018)** **Answer Page:5.43-5.45 - D.EdwinDhas** • Simplify the Boolean functions using K – Map (4M) • Implementation (6M)
15	**Implement the switching functions:** $\boldsymbol{Z_1 = ab'd'e + a'b'c'd'e' + bc + de}$ $\boldsymbol{Z_2 = a'c'e}$ $\boldsymbol{Z_3 = bc + de + c'd'e' + bd}$ $\boldsymbol{Z_4 = a'c'e + ce}$ **Using a 5*8*4 PLA** **(7M)** BTL4 **Answer Page:5.45-5.49 - D.EdwinDhas** • Implementation (7M) • Plot the PLA table. • Write the product term. • Write 1 in respective inputs. • Write 1 in respective outputs.
	PART* C
1	**Write short notes on sequential programmable devices & FPGA. (15M)** BTL1 **(Apr/May 2015)** **Answer Page:5.51-5.54 - D.EdwinDhas** Basic Architecture of FPGA (4M) An LUT programmed to produce the SOP function (3M) Basic block in an FPGA (4M) A simplified typical FPGA logic element (4M)
2	**Generate the following Boolean functions with a PAL with 4 inputs and 4 outputs** $\boldsymbol{Y_3 = A'BC'D' + A'BCD' + ABC'D}$ $\boldsymbol{Y_2 = A'BCD' + A'BCD + ABCD}$ $\boldsymbol{Y_1 = A'BC' + A'BC + AB'C + ABC'}$ $\boldsymbol{Y_0 = ABCD}$ **(15M)** BTL4 **Answer Page:5.40-5.43 - D.EdwinDhas** • Simplify the Boolean functions using K – Map (6M)

	• Implementation (7M)
3	**Implement the following Boolean functions using PAL.** $\boldsymbol{w(A,B,C,D) = \sum m(0,2,6,7,8,9,12,13)}$ $\boldsymbol{x(A,B,C,D) = \sum m(0,2,6,7,8,9,12,13,14)}$ $\boldsymbol{y(A,B,C,D) = \sum m(2,3,8,9,10,12,13)}$ $\boldsymbol{z(A,B,C,D) = \sum m(1,3,4,6,9,12,14)}$**(Nov – Dec 2015)(Nov/Dec 2018) (15M)** BTL4 **Answer Page:5.40-5.43 - D.EdwinDhas** • Simplify the Boolean functions using K – Map (3M) • Array Logic for PAL (3M) • PAL Program table (3M) • Implementation (3M)

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

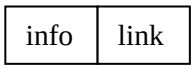
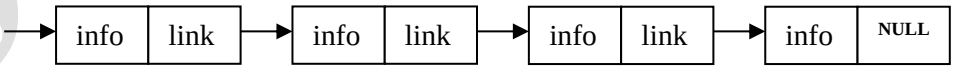
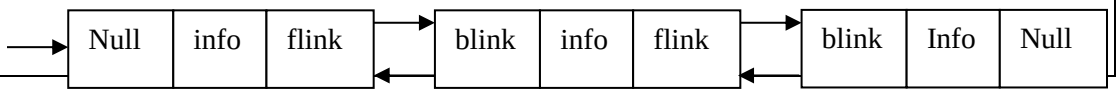
REFERENCES:

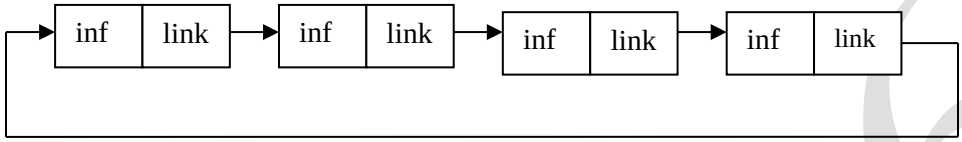
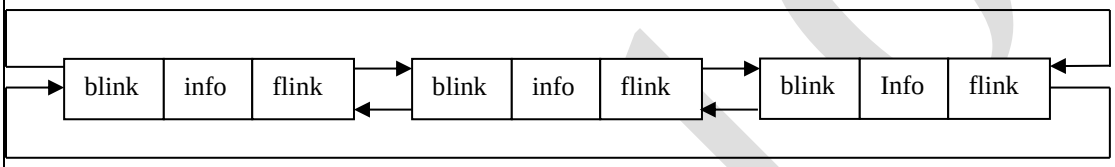
1. Thomas H. Cormen, Charles E. Leiserson, Ronald L.Rivest, Clifford Stein, -Introduction to Algorithms", Second Edition, Mcgraw Hill, 2002.
2. Aho, Hopcroft and Ullman, -Data Structures and Algorithms|, Pearson Education,1983.
3. Stephen G. Kochan, -Programming in C|, 3rd edition, Pearson Education.
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.

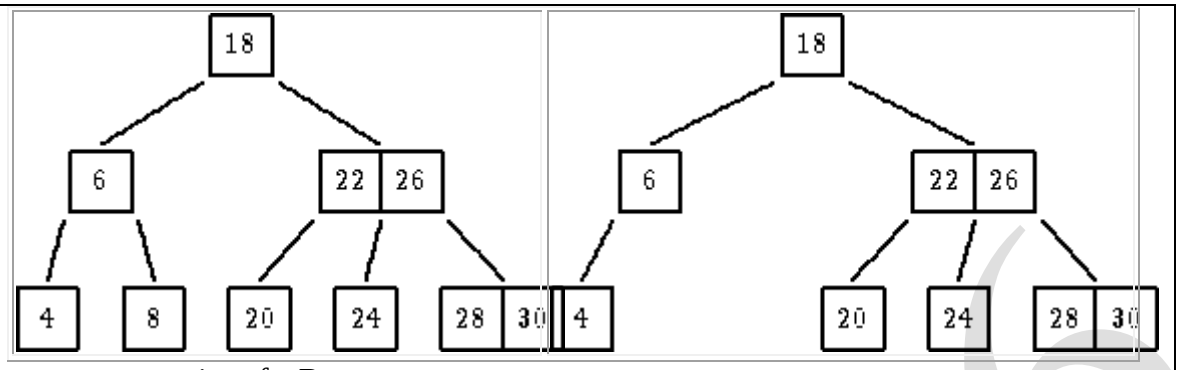
5.	**What is a DeQueue? Explain its operation. (13M)** BTL2 **Answer pg no:264-268 Reema Theraja** **Definition(2M)** Deque or Double Ended Queue is a generalized version of Queue data structure that allows insert and delete at both ends.

	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

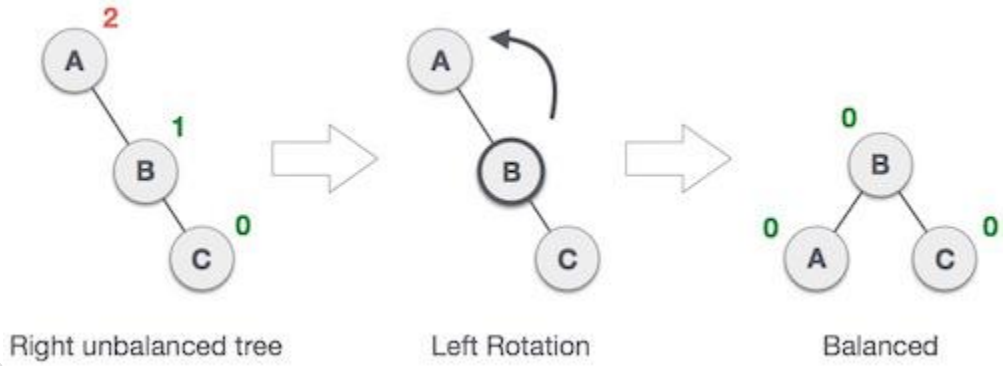
AVL Rotations

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

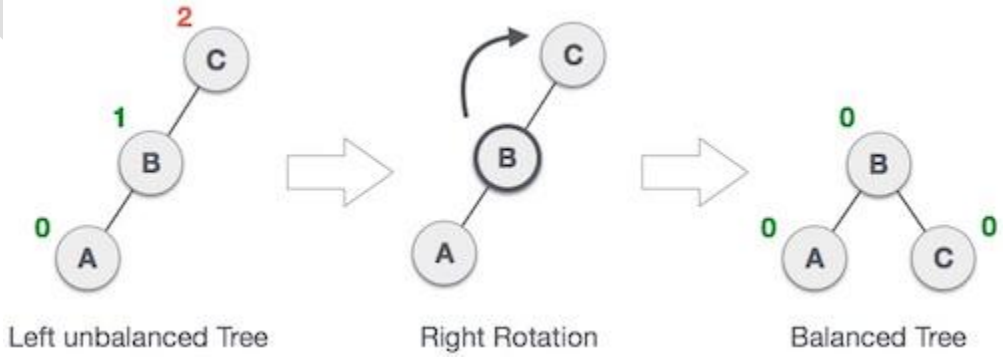
- Left rotation
- Right rotation
- Left-Right rotation
- Right-Left rotation

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 kth 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

EC8395 COMMUNICATION ENGINEERING L T P C
3 0 0 3

OBJECTIVES:

- To introduce the relevance of this course to the existing technology through demonstrations, case studies, simulations, contributions of scientist, national/international policies with a futuristic vision along with socio-economic impact and issues
- To study the various analog and digital modulation techniques
- To study the principles behind information theory and coding
- To study the various digital communication techniques

UNIT I ANALOG MODULATION 9

Amplitude Modulation – AM, DSBSC, SSBSC, VSB – PSD, modulators and demodulators – Angle modulation – PM and FM – PSD, modulators and demodulators – Superheterodyne receivers

UNITII PULSE MODULATION 9

Low pass sampling theorem – Quantization – PAM – Line coding – PCM, DPCM, DM, and ADPCM And ADM, Channel Vocoder - Time Division Multiplexing, Frequency Division Multiplexing

UNIT III DIGITAL MODULATION AND TRANSMISSION 9

Phase shift keying – BPSK, DPSK, QPSK – Principles of M-ary signaling M-ary PSK & QAM – Comparison, ISI – Pulse shaping – Duo binary encoding – Cosine filters – Eye pattern, equalizers

UNIT IV INFORMATION THEORY AND CODING 9

Measure of information – Entropy – Source coding theorem – Shannon–Fano coding, Huffman Coding, LZ Coding – Channel capacity – Shannon-Hartley law – Shannon's limit – Error control codes – Cyclic codes, Syndrome calculation – Convolution Coding, Sequential and Viterbi decoding

UNIT V SPREAD SPECTRUM AND MULTIPLE ACCESS 9

PN sequences – properties – m-sequence – DSSS – Processing gain, Jamming – FHSS – Synchronisation and tracking – Multiple Access – FDMA, TDMA, CDMA,

TOTAL: 45 PERIODS

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

- Ability to comprehend and appreciate the significance and role of this course in the present contemporary world
- Apply analog and digital communication techniques.
- Use data and pulse communication techniques.
- Analyze Source and Error control coding.

TEXT BOOKS:

1. H Taub, D L Schilling, G Saha, -Principles of Communication Systems‖ 3/e, TMH 2007
2. S. Haykin -Digital Communications‖ John Wiley 2005

REFERENCES:

1. B.P.Lathi, -Modern Digital and Analog Communication Systems‖, 3rd edition, Oxford University Press, 2007
2. H P Hsu, Schaum Outline Series – -Analog and Digital Communications‖ TMH 2006
3. B.Sklar, Digital Communications Fundamentals and Applications‖ 2/e Pearson Education 2007.

UNIT I- ANALOG MODULATION	
Amplitude Modulation – AM, DSBSC, SSBSC, VSB – PSD, modulators and demodulators – Angle modulation – PM and FM – PSD, modulators and demodulators – Superheterodyne receivers	
PART * A	
Q.No.	**Questions**
1.	**Define modulation? What is the need formodulation?** BTL1 Modulation is a process by which some characteristics of high frequency carrier Signal is varied in accordance with the instantaneous value of the modulating signal. **Needs for modulation:** • Ease of transmission • Multiplexing • Reduced noise • Narrow bandwidth • Frequency assignment • Reduce the equipments limitations.
2	**Give the Classification of Modulation.**BTL1 There are two types of modulation. They are a)Analog modulation, b)Digital modulation Analog modulation is classified as follows (i)Continuous wave modulation (ii)Pulse modulation Continuous wave modulation is classified as follows (i)Amplitude modulation(ii)Double side band suppressed carrier (iii)Single side band suppressed carrier (iv)Vestigial side band suppressed carrier Angle modulation Frequency modulation Phase modulation Pulse modulation is classified as follows (i)Pulse amplitude modulation (ii)Pulse position modulation (iii)Pulse duration modulation (iv)Pulse code modulation Digital modulation is classified as follows Amplitude shift keying (ii)Phase shift keying Frequency shift keying.
3	**Define the term modulation index for AM. (May/June 2015)** BTL1 Modulation index is the ratio of amplitude of modulating signal (Em) to amplitude of carrier (Ec) i.e.m = Em/Ec
4	**What are the degrees of modulation?**BTL1 a) Under modulation (m < 1) b) Critical modulation (m=1) c) Over modulation(m>1)

5

Compare linear and non-linear modulators?BTL4

S.No	Linear Modulators	Non Linear Modulators
1.	Heavy filtering is not required.	Heavy filtering is required.
2.	These modulators are used in high level modulation.	These modulators are used in low level modulation.
3.	The carrier voltage is very much greater than modulating signal	The modulating signal voltage is very much greater than the carrier signal voltage.

6

Define Amplitude Modulation.BTL1

In amplitude modulation, the amplitude of a carrier signal is varied according to variations in amplitude of modulating signal.

The AM signal can be represented mathematically as, eAM = (Ec + Em sinωmt) sinωct

and the modulation index is given as,m = Em /EC

7

What is Super Heterodyne Receiver?BTL1

The super heterodyne receiver converts all incoming RF frequencies to a fixed lower frequency, called intermediate frequency (IF). This IF is then amplitude and detected to get the original signal.

8

Difference between high level modulation and lowlevel modulation?BTL4

Low level AM modulator	High level AM modulator
1. Modulation takes place prior to the final stage of the transmitter	1.Modulation takes place in the final element of final stage.
2. Less modulating signal power is required	2.More modulating signal power is required.

9

Compare AM with DSB-SC and SSB-SC.(MAY/JUNE 2013)BTL4

S.No	AM signal	DSB-SC	SSB-SC
1	Bandwidth = 2fm	Bandwidth = 2fm	Bandwidth = fm
2	Contains USB,LSB,Carrier	Contains USB,LSB	USB,LSB
3	More Power is required for transmission	Power required is less than that of AM.	Power required is less than AM &DSB-SC

10

What are the advantages of VSB-AM?BTL1

1. It has bandwidth greater than SSB but less than DSB system.
2. Power transmission greater than DSB but less than SSB system.
3. No low frequency component lost. Hence it avoids phase distortion.

11

How will you generating DSBSC-AM?BTL6

There are two ways of generating DSBSC-AM such as

a).Balanced modulator,

b).Ring modulators

12

Define demodulation BTL1

Demodulation or detection is the process by which modulating voltage is recovered from the modulated signal. It is the reverse process of modulation. The devices used for demodulation or detection are called demodulators or detectors. For amplitude modulation, detectors or demodulators are categorized as,

2. Square-law detectors
3. Envelope detectors

13

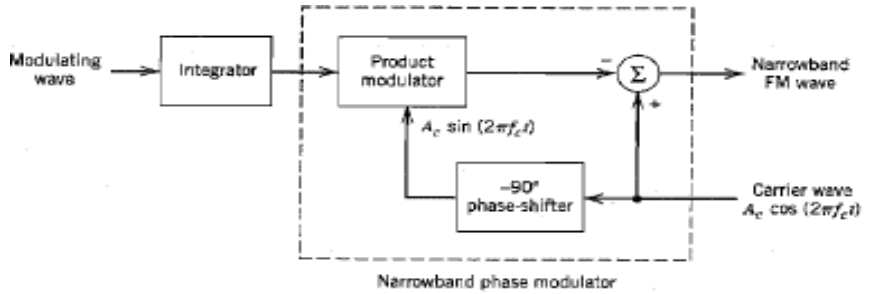
Define DSB-SC. BTL 1

After modulation, the process of transmitting the sidebands (USB, LSB) alone and suppressing the carrier is called as Double Side Band-Suppressed Carrier.

14

Define SSB-SC. BTL1

	(i) SSB-SC stands for Single Side Band Suppressed Carrier (ii) When only one sideband is transmitted, the modulation is referred to as Single side band modulation. It is also called as SSB or SSB-SC.
15	**What is Vestigial Side Band Modulation?**BTL1 Vestigial Sideband Modulation is defined as a modulation in which one of the sideband is partially suppressed and the vestige of the other sideband is transmitted to compensate for that suppression.
16	**Define depth modulation in AM. (Nov 2017)** BTL1 It is defined as the ratio between message amplitude to that of carrier amplitude. m=Em/Ec. It is also known as co efficient of modulation
17	**What are the advantages of signal sideband transmission? What are the disadvantages of single side band transmission?** BTL1 **Advantage:** a) Power consumption b) Bandwidth conservation c) Noise reduction **Disadvantages** : A) Complex receivers: Single side band systems require more complex and expensive receivers thn conventiaonal AM transmission. B) Tuning difficulties: Single side band receivers require more complex and precise tunig than conventional AM receivers.
18	**Define frequency modulation. Define modulation index of frequency modulation. (May/June 2015)** BTL1 Frequency modulation is defined as the process by which the frequency of the carrier wave is varied in accordance with the instantaneous amplitude of the modulating or message signal. Modulation index of frequency modulation. It is defined as the ratio of maximum frequency deviation to the modulating $\beta = \delta f / f_m$
19	**Define phase modulation. (MAY/JUNE 2014)**BTL1 Phase modulation is defined as the process of changing the phase of the carrier signal in accordance with the instantaneous amplitude of the message signal.
20	**What are the types of Frequency Modulation? (MAY/JUNE 2015)**BTL1 Based on the modulation index FM can be divided into types. They are Narrow band FM and Wide band FM. If the modulation index is greater than one then it is wide band FM and if the modulation index is less than one then it is Narrow band FM
21	**What are the two methods of producing an FM wave?** BTL1 Basically there are two methods of producing an FM wave. They are, i) Direct method: In this method the transmitter originates a wave whose frequency varies as function of the modulating source. It is used for the generation of NBFM ii) Indirect method: In this method the transmitter originates a wave whose phase is a function of the modulation. Normally it is used for the generation of WBFM where WBFM is generated from NBFM
22	**A 80 MHz carrier is frequency modulated by a sinusoidal signal of 1V amplitude and the frequency sensitivity is 100 Hz/V. Find the approximate bandwidth of the FM waveform if the modulating signal has a frequency of 10 kHz.**BTL5 Frequency Sensitivity = 100 Hz/ volt. Amplitude of modulating signal = 1V Hence maximum frequency deviation, δ =100 Hz / volt ×1V= 100 kHz Frequency of modulating signal, fm =10khz BW = 2 [δ + f (max)] = 2 [100 +10×10^3] BW = 20.2 kHz
23	**Draw the block diagram of a method for generating a narrow band FM.**BTL6



Block diagram of a method for generating a narrowband FM signal.

24

Distinguish between Narrow band FM and Wide Band FM. (April 2018, Nov 2016, May 2013) BTL 5

S.No	Narrow band FM	Wide band FM
1	Modulation index is < 1	Modulation index > 10
2	$s(t) = A_c \cos(2\pi f_c t) - mA_c \sin(2\pi f_c t)\sin(2\pi f_m t)$	$s(t) = A_c \sum_{n=-\infty}^{\infty} J_n(m)\cos[2\pi(f_c + nf_m)t]$
3	Spectrum contains 2 sidebands and carrier	Spectrum cont6ains infinite number of sidebands and carrier
4	BW=2fm	Bandwidth: $BW = 2(\delta + f_m(\max))$
5	It is used for mobile communication	It is used for broadcasting and Entertainment
6	Maximum deviation =75Hz	Maximum deviation = 5 Hz
7	Range of modulating frequency 30Hz to 15 Kz	Range of modulating frequency 30Hz to 3 Kz

25

Obtain the bandwidth of the FM signal.

c (t) =10 × cos [2 ×10 7 ×π t + 8 cos (1000 ×π t)] BTL 6

Compare the given FM signal equation with standard FM signal equation, c (t) = Ec cos (ωc t + m cos ωm t)

Here,

m = 8 ,ωm =1000 π , Hence f m =1000 π or f m = 500 Hz 2π

δ = m f m=8× 500 Hz = 4000 Hz

BW = 2 (δ + fm)= 2 (4000 + 500) = 9000 Hz or 9 kHz

PART B

1

Explain the operation of super heterodyne receiver with neat block diagram. (Nov 2017) BTL 1

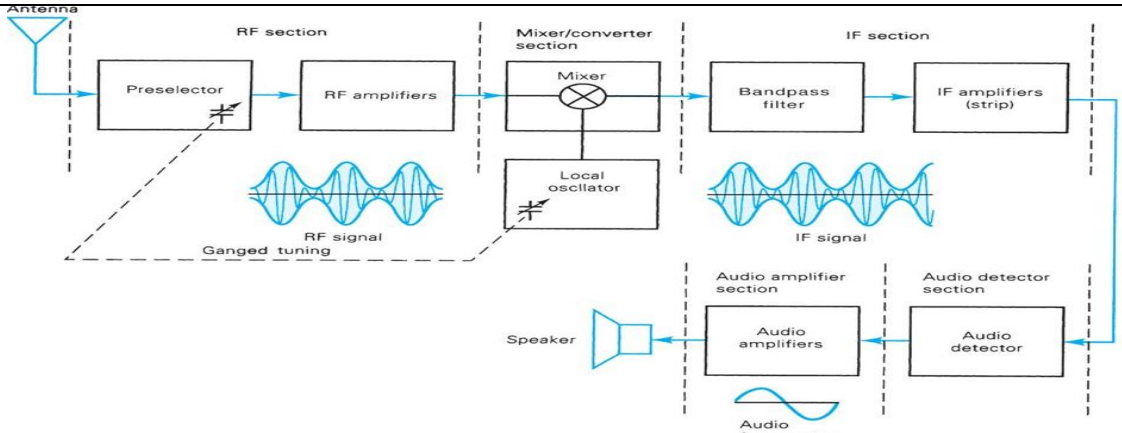
Discuss the principle of AM Super heterodyne receiver with a block diagram. (13M) **(April 2016)** BTL 1

Answer: Page 5.11 - K.Muralibabu

Heterodyning: Mixing two frequencies in a non linear device. (2M)

Translate one frequency to another

Diagram: (5M)



Explanation: (6M)

RF selector-consists of preselector and amplifier-band limiting- prevent image frequency.
Mixer-includes local oscillator, detector-heterodyning occurs.
Intermediate frequency-455 KHz.
IF section-Achieved receiver gain and selectivity
Detector-convert IF signals to original signal.

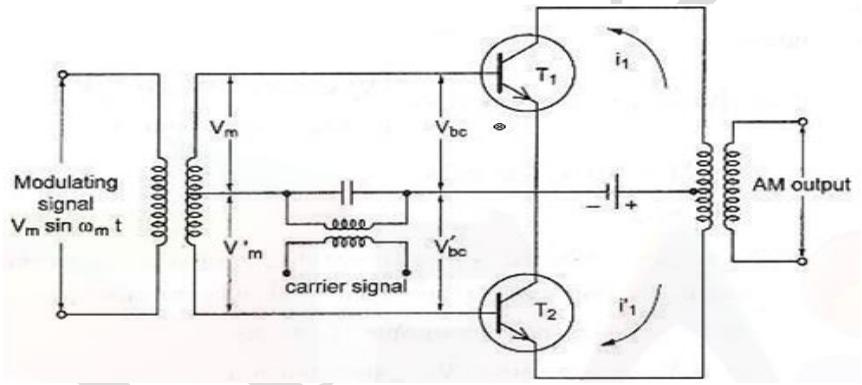
2 **With The Help Of Neat Diagram Explain The Generation Of DSB-SC Using Balanced Modulator And Ring Modulator. (Nov/Dec 2010, MAY/JUNE 2016,NOV/DEC 20160 (13M)**BTL1

Answer:Pg:197-H Taub,D L Schilling

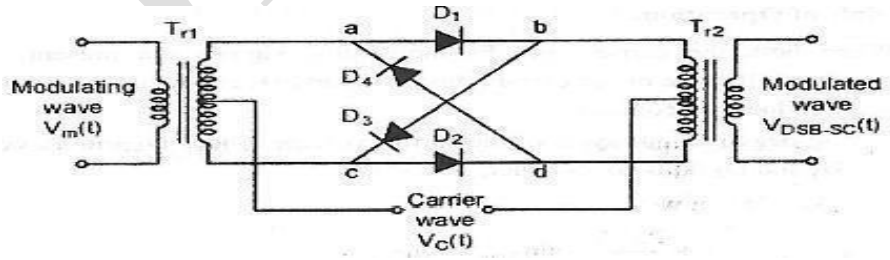
Definition **(2M)**

Generation of DSB – SC – AM **(7M)**

There are two ways of generating DSB – SC – AM such as ,
Balanced modulator,
Ring modulator.
Balanced modulator



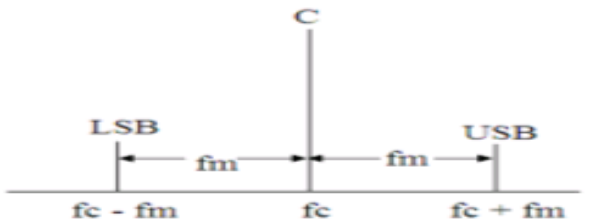
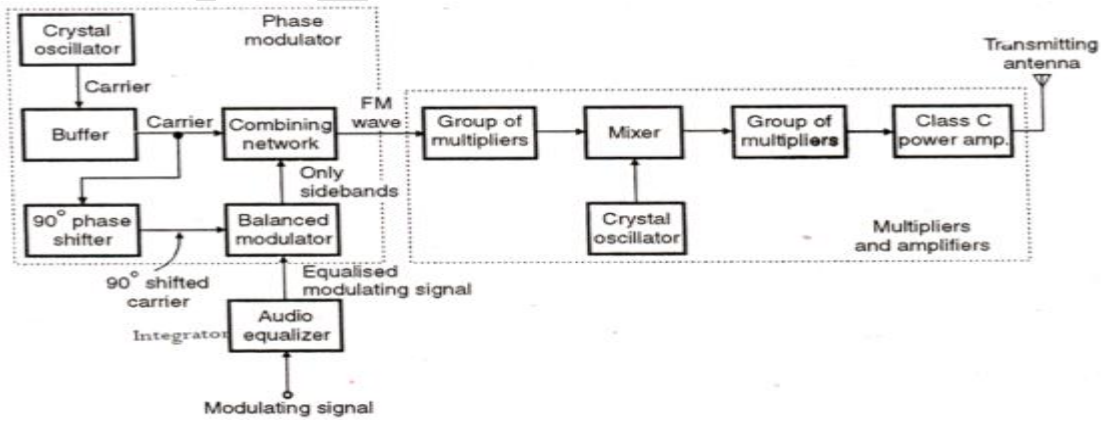
Ring modulator or diode balanced modulator:



(4M)

3 **Explain the operatio of DSB -FC-AM circuit. (13M)**BTL1

Answer : Page : 150- H Taub, D L Schilling

	Introduction of non linear property (1M) **Types** 1. Square law modulator (2M) 2. balanced modulator (2M) Circuit diagram and Vout expression (5M) Modulation index =ma= 2a2Vm/a1 (1M) **Advantages** - The undesired nonlinear terms are eliminated by a BPF.hence the BPF must be carefully designed (1M)
4	**Derive the frequency spectrum of AM signal. What is the power distribution in the AM signal?** (13M) **(April 2017)** BTL 1 **With a neat diagram derive the expressions for frequency spectrum of AM wave.** (13M) **(Nov 2015)** BTL 1 **Answer: Page 193 - K.Muralibabu** **AM:** Changing high frequency carrier amplitude (2M) **Voltage distribution of AM:** (4M) $e_{AM} = E_c \sin \omega_c t + m E_c /2 \cos(\omega_c - \omega_m)t - m E_c /2\cos(\omega_c + \omega_m)t$ **Power distribution**: (3M) P_t= Carrier power + Power in USB + Power in LSB $= E_{carr}^2 / R + E_{USB}^2 / R + E_{LSB}^2 /R$ **Frequency Spectrum:** (3M)  Frequency spectrum of AM wave
5	**Discuss the Armstrong method for the generation of FM signal. (13M) (Nov 2017, April 2017, Nov 2014)** **Explain in detail about indirect method of FM generation. (13M) (Nov 2016)** BTL **2** **Answer: Page 193 - K.Muralibabu** Armstrong Method: Indirect method- using PM for generating FM. (2M) Diagram: (5M)  Explanation: Modulating signal integrated-phase modulated with carrier signal. (6M) Frequency multipliers-get desired wideband FM. Crystal oscillator-used to generate PM-from which narrowband FM obtained. Multipliers- to increase carrier frequency and frequency deviation.
6	**Write The Comparison Of Amplitude Modulation Systems.(MAY/JUNE 2012)**BTL4

Answer: Page 4.24 - K.Muralibabu (13M)

Description on	AM with carrier	DSB – SC – AM	SSB – SC - AM	VSB - AM
Band width	2fm	2fm	fm	fm <BW<2fm
Power Saving for Sinusoid al	33.33%	66.66%	83.3%	75%
Power Saving for non - Sinusoidal	33.33%	50%	75%	75%
Generation methods	Easier to generate	Not difficult	More difficult to generate	Difficult. But easier to generate than SSB-SC
Detection methods	Simple & Inexpensive	Difficult	More difficult	Difficult

7 **Write The Comparison Of Wideband And Narrowband FM (NOV/DEC 2011, MAY/JUNE 2013)**BTL4

Answer: Page 6.15 - K.Muralibabu (13M)

SI. No	Parameter Characteristics	Wideband	Narrowband FM
1.	Modulation index	Greater than	Less than (or) slightly greater than 1
2.	Maximum deviation	75 KHz	5 KHz
3.	Range of modulating Frequency	30 Hz to 15	30 Hz to 3 KHz
4.	Bandwidth	Large, about 15 times higher than BW of narrow band FM. BM=2(ΔF+F	Small. Approximately same as that of AM. BW=2fm
5.	Maximum modulation	5 to 2500	Slightly greater than 1
6.	Pre-emphasis and De-emphasis	Needed	Needed
7.	Noise	Noise is more suppressed	Less suppressing of noise
8.	Applications	Entertainment broadcasting	FM mobile communication

9.	Side bands	Spectrum contains infinite number of side bands and carrier	Spectrum contains two sidebands and carrier.	

8

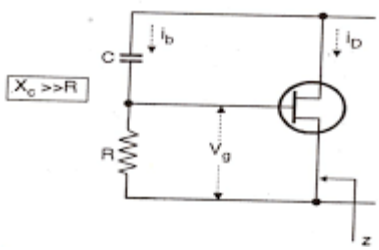
Explain With Diagram The Generation Of FM Using Direct Method. (APRIL/MAY 2015, NOV/DEC 2016) (13M)BTL1

Answer: Page 7.1 - K.Muralibabu

Direct FM (2M)

Direct FM can be obtained by using FET and varactor diode. These methods are discussed next.

(i).FET Reactance Modulator (5M)



(ii).Frequency Modulation using varactor Diode (6M)

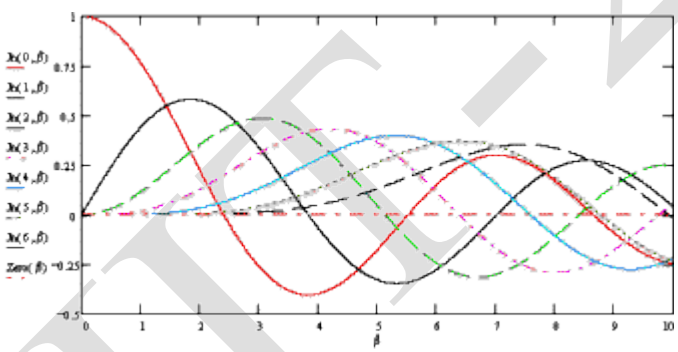
9

Using a suitable mathematical analysis, show that FM modulation produces infinite sidebands. Also derive an expression for the frequency modulated output and its spectrum. (13M) (Nov 2015) BTL 1

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.

$$V_{pm}(t) = V_c \sum_{n=-\infty}^{\infty} J_n(m) cos(\omega_c t + n\,\omega_m t + \frac{n\pi}{2})$$ (8M)



(3M)

10

Explain the generation of DSB-SC wave using Balanced Modulator. Derive the power of DSB-SC signal. (13M) **(Nov 2016)** BTL 1

Answer: Page 4.6 - K.Muralibabu

DSB-SC: Transmitted wave-contains-only upper and lower sidebands. (2M)

Explanation: (3M)

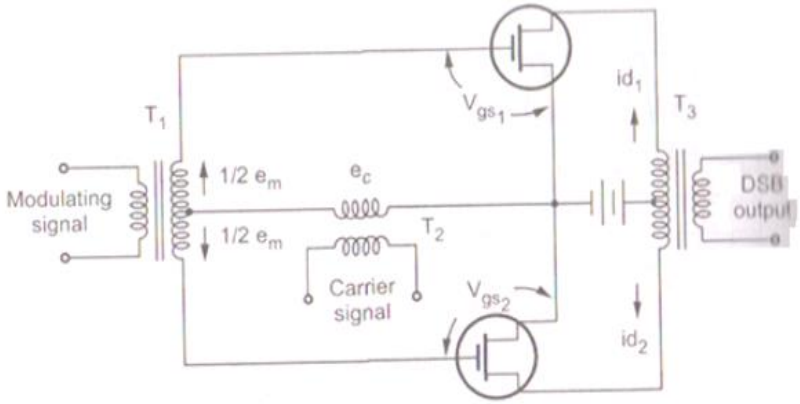
- Also called ring modulator-produce DSB-SC.
- Two non linear devices-connected-in balanced mode-to suppress carrier.
- Symmetric circuit.

Equation: (2M)

$V_0 = 2Ka_1 v_m[1 + m_a cos\omega_c t]cos\omega_m t$

Power: $\boldsymbol{P_t = [1 + \frac{m^2}{2}]}$ (3M)

Circuit Diagram: (3M)



11

Illustarte the operation of VSB transmission .(13M) BTL1

Answer: Page: 171 - H Taub, D L Schilling

Definition: (2M)

The design of the sideband filter is simplified since the need for sharp cutoff at the carrier frequency is eliminated. In addition, a VSB system has improved low-frequency response and can even have dc response

VSB modulator and demodulator diagram: (4M)

Explanation: The product modulator generates DSB-SC signal from the message and carrier signals, VSB signal can be generated by passing a DSB-SC signal through as appropriate BPF having a transfer function (3M)

Magnitude response of VSB filter diagram & $|H(fc)| = \frac{1}{2}$ (3M**)**

Applications : VSB is used in television for transmission of picture signal

Advantages: required BW is less compared to DSB, the filter required need not have a sharp cut off (1M)

PART * C

1

Compare Am and FM in detail.BTL4 (15M)

Answer: Page 6.13 - K.Muralibabu

	AM	**FM**
Stands for	AM stands for Amplitude Modulation	FM stands for Frequency Modulation
Origin	AM method of audio transmission was first successfully carried out in the mid 1870s.	FM radio was developed in the United states in the 1930s, mainly by Edwin Armstrong.
Modulating differences	In AM, a radio wave known as the "carrier" or "carrier wave" is modulated in amplitude by the signal that is to be transmitted. The frequency and phase remain the same.	In FM, a radio wave known as the "carrier" or "carrier wave" is modulated in frequency by the signal that is to be transmitted. The amplitude and phase remain the same.
Pros and cons	AM has poorer sound quality compared with FM, but is cheaper and can be transmitted over long distances. It has a lower bandwidth so it can have more stations available in any frequency range.	FM is less prone to interference than AM. However, FM signals are impacted by physical barriers. FM has better sound quality due to higher bandwidth.

Frequency Range	AM radio ranges from 535 to 1705 KHz (OR) Up to 1200 bits per second.	FM radio ranges in a higher spectrum from 88 to 108 MHz. (OR) 1200 to 2400 bits per second.
Bandwidth Requirements	Twice the highest modulating frequency. In AM radio broadcasting, the modulating signal has bandwidth of 15kHz, and hence the bandwidth of an amplitude-modulated signal is 30kHz.	Twice the sum of the modulating signal frequency and the frequency deviation. If the frequency deviation is 75kHz and the modulating signal frequency is 15kHz, the bandwidth required is 180kHz.
Zero crossing in modulated signal	Equidistant	Not equidistant
Complexity	Transmitter and receiver are simple but syncronization is needed in case of SSBSC AM carrier.	Transmitter and receiver are more complex as variation of modulating signal has to beconverted and detected from corresponding variation in frequencies.(i.e. voltage to frequency and frequency to voltage conversion has to be done).
Noise	AM is more susceptible to noise because noise affects amplitude, which is where information is "stored" in an AM signal.	FM is less susceptible to noise because information in an FM signal is transmitted through varying the frequency, and not the amplitude.

2

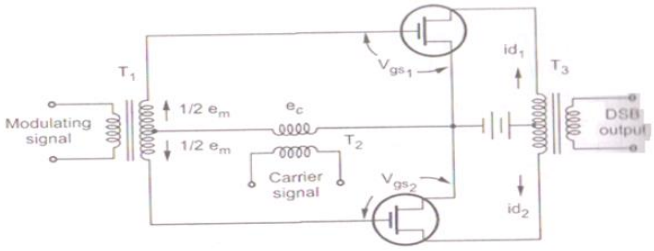
Explain the generation of DSB-SC wave using Balanced Modulator. Derive the power of DSB-SC Signal (15M)
BTL1

Answer: Page: 147 - H Taub, D L Schilling

Introduction of DSB-SC –the modulated wave consists of only upper and lower sidebands Transmitted power is saved here through the suppression of the carrier wave because it does not contain any useful information ,but the channel bandwidth required is the same as AM

BW=2fm (only upper and lower sidebands),66.6% (2M)

DSB-SC generation diagram



(3M)

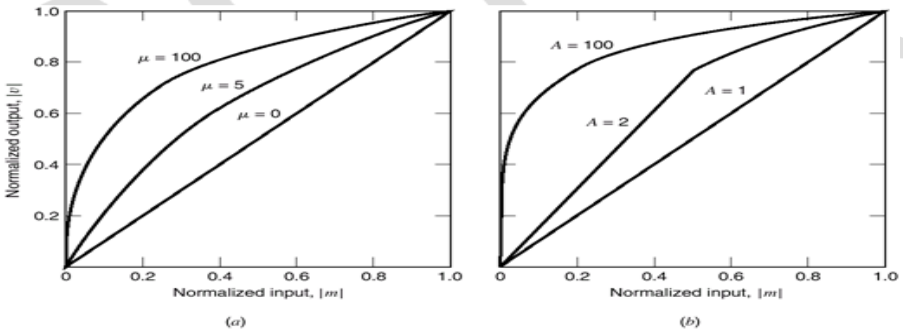
V(AM) equation ,phosar diagram,waveform (1M)

Types – balanced modulator(nonlinear device) ,ring modulator (didoes, positive halfcycle-D1,D2-Forward device and D3,D4- reverse biased -magnetic field cancel each other and negative half cycle

D1,D2-reverse biased and D3,D4- forward biased -opposite magnetic field) (8M)

Advantages-more efficient is use of transmitted power (66.6%),better signal to noise ratio compared to SSB

	Disadvantages -Even though the carrier is suppressed ,the BW of DSB-SC remains same as AM and more than SSB (2M)
3	**A,(i)Determine the peak frequency deviation (Δf) and modulation index(mf) for an FM modulation with a deviation sensitivity Kf=5khz and modulating signal Vm(t)=2cos(2π*2000t)** BTL6 **(7M)** **(ii)The peak phase deviation (mp) for a pm modulator with a deviation sensitivity Kp=2.5rad/v and a modulating signal Vm(t)=2cos (2π 2000t)** **(8M)** Kf=5khz/v Vm(t)= 2cos (2π 2000t) Kp=2.5 rad/v Vm=2v,fm=2000hz Peak freq deviation Δf=Kf Vm =5khz/v *2v=10khz Modulation index mf=Δf/fm=10khz/2khz=5 Modulation index of PM Mp=KpV=2.5 rad/v * 2=5rad **B,In a FM system,if the max value of deviation is 75khz and the max modulating freq is 10khz.Calculate the deviation ratio and bandwidth of the system using carsons rule.** BTL3 Deviation ratio D= Δfm Max/Fm max =75/10 =7.5 System bandwidth B=2[=Δfmax + fm(max)] =2[75+10] =170khz

UNIT II – PULSE MODULATION	
Low pass sampling theorem – Quantization – PAM – Line coding – PCM, DPCM, DM, and ADPCM And ADM, Channel Vocoder – Time Division Multiplexing, Frequency Division Multiplexing	
PART * A	
Q.No.	**Questions**
1.	**What is aliasing or foldover? (MAY/JUNE2016& Nov/Dec 2016)** BTL1 Aliasing effect takes place when sampling frequency is less than Nyquist rate. Under such condition, the spectrum of the sampled signal overlaps with itself. Hence higher frequencies take the form of lower frequencies. This interference of the frequency components is called as aliasingeffect A band limited signal of finite energy, which has no frequency components higher than W Hz, may be completely recovered from the knowledge of its samples taken at the rate of 2W samples persecond A band limited signal of finite energy, which has no frequency components higher than W Hz, may be completely recovered from the knowledge of its samples taken at the rate of 2W samples persecond
2.	**What is companding? Sketch the input and output characteristics of expander and compressor and also write A-law and μ-law for compression. (MAY/JUNE2016& Nov/Dec 2016& April/May 2017)**BTL1 The signal is compressed at the transmitter and expanded at the receiver. This is called as companding. The combination of a compressor and expander is called a compander.  $\|v\|=\frac{log(1+\mu\|m\|)}{log(1+\mu)}$ $\|v\| = \begin{cases} \frac{A\|m\|}{1+logA} \\ \frac{1+logA\|m\|}{1+logA} \end{cases}$ $0 \le m \le \frac{1}{A}$
3	**State sampling theorem for band limited signals and filters to avoid aliasing(OR)State Low passsamplingtheorem (NOV/DEC2015)** BTL3 ➢ If a finite –energy signal g(t) contains no frequencies higher than W hertz ,it is completely determined by specifying its co=ordinates at a sequence of points spaced ½W seconds apart. ➢ If a finite energy signal g(t) contains no frequencies higher than W hertz, it may be completely recovered from its co=ordinates at a sequence of points spaced ½W secondsapart. ➢ Filters to avoidaliasing o Before sampling a low pass pre alias filter is used to attenuate those high frequency components which do not contribute the information content of thesignal. o The filtered signal is sampled at a rate slightly higher than Nyquist rate 2W.
4	**Define quantizing process and Write the two fold effects of quantization process. (NOV/DEC2015)** BTL1 The conversion of analog sample of the signal into digital form is called quantizingprocess. The peak-to-peak range of input sample values subdivided into a finite set of decision levels or decision thresholds The output is assigned a discrete value selected from a finite set of representation levels are reconstruction values that are aligned with the treads of the staircase.
5	**Define Nyquist rate, Nyquist interval, Dirac comb and crestfactor.** BTL1 Nyquist rate-Let the signal be band limited to „W Hz. Then Nyquist rate is given as, Nyquist rate = 2W samples/sec

	Aliasing will not take place if sampling rate is greater than Nyquist rate. Nyquist interval- Its reciprocal ½W is called Nyquist interval. Dirac comb or ideal sampling function- It is nothing but a periodic impulse train in which the impulses are spaced by a time interval of Ts seconds. Crest Factor – It defines how strong the peak value is with respect to its rms value.
6	**What is meant byquantization, Quantization noise power andquantization error?** BTL1 While converting the signal value from analog to digital, quantization is performed. The analog value is assigned to nearest digital value. This is called quantization.The quantized value is then converted into equivalent binary value. .The quantization levels are fixed depending upon the number of bits. Quantization is performed in every Analog to DigitalConversion. Quantization noise power is the quantizer error instance and is given byvariance(ie) , where q is stepsize. Quantization error is the difference between the output and input values ofquantizer
7	**What is meant by PCM and what are the noises present in PCM system what is the SNR of PCM system if the number of quantization level is2^8?** BTL1 Pulse code modulation (PCM) is a method of signal coding in which the message signal is sampled, the amplitude of each sample is rounded off to the nearest one of a finite set of discrete levels and encoded so that both time and amplitude are represented in discrete form.. This allows the message to be transmitted bymeans of a digitalwaveform. Aliasing noise, quantization noise, Channel noise, Intersymbol interference Signal to noiseRatio $\frac{S}{N} indB \leq (4.8 + 6\vartheta)dB$ $\leq (4.8 + 6 \times 8)dB$ $\leq 52.2\, dB$
8	**What you meant by non- uniformquantization?** BTL1 Step size is not uniform. Non uniform quantizer is characterized by a step size that increases as the separation from the origin of the transfer characteristics is increased. Non- uniform quantization is otherwise called as robust quantization.
9	**What is the disadvantage of uniform quantization over the non-uniform quantization?** BTL1 SNR decreases with decrease in input power level at the uniform quantizer but non-uniform quantization maintain a constant SNR for wide range of input power levels. This type of quantization is called as robust quantization.
10	**What are the advantages and disadvantages of PCMSystem?** BTL1 Advantages: Very Efficient power bandwidthexchange. Highly robust as it is immune to channel noise andinterference. As regenerative repeaters are used, PCM used in long haul communication. As it is digital coding techniques are available for compression, encryption and errorcorrection. Disadvantages PCM signal generation and reception involve complexprocesses. PCM requires much larger transmission bandwidth than analog modulation.
11	**What is TDM? Write its advantages anddisadvantages**BTL1 The system which enables joint utilization of a common channel by many independent message signals without mutual interference is called Time Division Multiplexing system. **Advantages:** Only one carrier in the medium at any giventime High throughput even for manyusers Common TX component design, only one poweramplifier Flexible allocation of resources (multiple timeslots). **Disadvantages** Synchronization Requires terminal to support a much higher data rate than the user information rate therefore possible problems with intersymbol- interference.
12	**What is meant by forward and backward estimation?(NOV/DEC 2015)** BTL1 AQF: Adaptive quantization with forward estimation. Unquantized samples of the input signal are used to derive the forwardestimates. AQB: Adaptive quantization with backward estimation. Samples of the quantizer output are used to derive the backward estimates.

	APF: Adaptive prediction with forward estimation, in which unquantized samples of the input signal are used to derive the forward estimates of the predictor coefficients. APB: Adaptive prediction with backward estimation, in which Samples of the quantizer output and the prediction error are used to derive estimates of the predictor coefficients. Limitations of forward estimation with backward estimation: Sideinformation Buffering Delay
13	**What are the advantages & disadvantage of delta modulator? (MAY/JUNE2016)** BTL1 Advantage: Delta modulation transmits only one bit for one sample. Thus the signaling rate and transmission channel bandwidth is quite small for delta modulation. The transmitter and receiver implementation is very much simple for delta modulation. There is no analog to digital converter involved in deltamodulation. Disadvantage: Poor SNR due to quantisation noise It requires more bit rate than PCM for the same S/N performance
14	**Mention the merits of DPCM.** BTL1 Bandwidth requirement of DPCM is less compared toPCM. Quantization error is reduced because of predictionfilter Numbers of bits used to represent .one sample .value are also reduced compared to PCM.
15	**Define delta modulation and adaptive delta modulation(NOV/DEC 2015)** BTL1 Delta modulation is the one-bit version of differential pulse code modulation. Types of noise in DM or Drawbacks of DM 1.Slope Overload noise(Startup error) 2.Granular noise(Hunting) Adaptive delta modulation-The performance of a delta modulator can be improved significantly by making the step size of the modulator assume a time- varying form. In particular, during a steep segment of the input signal the step size is increased. Conversely, when the input signal is varying slowly, the step is reduced, In this way, the step size is adapting to the level of the signal. The resulting method is called adaptive delta modulation(ADM).
16	**Define ADPCM.** BTL1 It means adaptive differential pulse code modulation, a combination of adaptive quantization and adaptive prediction. Adaptive quantization refers to a quantizer that operates with a time varying step size. The autocorrelation function and power spectral density of speech signals are time varying functions of the respective variables. Predictors for such input should be time varying. So adaptive predictors are used.
17	**What is the main difference in DPCM and DM?**BTL4 DM encodes the input sample by one bit. It sends the information about + δ or -δ, ie step rise or fall. DPCM can have more than one bit of encoding the sample. It sends the information .about difference between actual sample value and the predicted sample value
18	**What are the two basic types of multiplexing?** BTL1 ○ Frequency division multiplexing ○ Time division multiplexing In FDM many signals are transmitted simultaneously where each signal occupies a different frequency slot within a common bandwidth. In TDM the signals are transmitted at a time, instead they are transmitted in different time slots.
19	**Advantages of FDM?** BTL1 ➢ A large number of signals (Channels) can be transmitted simultaneously ➢ FDM does not need synchronization between its transmitter and receiver for proper operation Demodulation of FDM is easy ➢ Due to slow narrow band fading only a single channel gets affected
20	**Applications of FDM?** BTL1 Telephone systems AM (amplitude modulation) and FM(frequency modulation) radio broadcasting TV broadcasting and also first generation of cellular phones used FDM.
21	**Compare uniform and non uniform quantisation** BTL4

S.No	**Uniform Quantisation**	**Non-uniform quantisation**

	1	The signal to noise ratio varies with input signal amplitude	The signal to noise ratio can be maintained constant with non uniform quantisation
	2	The quantisation step size remains same throughout the dynamic range of the signal	The quantisation step size varies according to specific law depending upon amplitude of the input signal

22

It is required to transmit speech over PCM channel with 8-bit accuracy.Assume the speech in baseband limited to 3.6KHZ.Determine the bit rate. BTL4

The signaling rate in PCM is

r=v fs

v=8bits

w=3.6khz

Sampling freq fs=2W

=2*3.6=7.2 KHZ

$r=v*fs=8*7.2*10^3=57.6$ bits/sec

23

Compare PCM and DPCM BTL4

S.NO	PCM	DPCM
1	Large BW required	BW is less than PCM
2	Poor SNR	Good SNR
3	Suitable for video,audio and telephony	Suitable for video and speech signal

24

Comparison between DM and ADM BTL

S.NO	DM	ADM
1	It uses only one bit for one sample	It uses only one bit for one sample
2	Step size is fixed	According to the signal variation step size varies
3	Lowest bandwidth is required	Highest bandwidth is required

25

What is ADM?What are the advantage and disadvantages of ADM BTL1

ADM: To overcome the quantisation error ,step size Δ is varied in accordance with changes in the input signal.ADM tracks changes in the sinusoidal input signal

Advantage:

- One bit per sample is taken
- Because of the variable step size,the dynamic range is wide
- SNR is better than DM

Disadvantage:

- Quantisation noise granular noise is present
- Adaptive algorithm uses complex circuits for modulation and demodulation

26

List the properties of linecodes (April/May 2017) BTL1

- Transmission Bandwidth : as small aspossible
- Power Efficiency : As small as possible for given BW and probability
- oferror
- Error detection and correction capability :ExBipolar
- Favorable power spectral density :dc=0

27

What are line codes? Name some popular line codes. (MAY/JUNE2016) BTL1

Line coding refers to the process of representing the bit stream (1"s and 0"s) in the form of voltage or current variations optimally tuned for the specific properties of the physical channel beingused.

- Unipolar (Unipolar NRZ and UnipolarRZ)
- Polar (Polar NRZ and PolarRZ)
- Non-Return-to-Zero, Inverted(NRZI)
- Manchesterencoding

28

Define the followingterms BTL1

2) **NRZ unipolar format**

ii) NRZ polar format

iii) NRZ bipolar format

iv) Manchesterformat

NRZ unipolar format-In this format binary 0 is represent by no pulse and binary

1 is represented by the positivepulse.
NRZ polar format-Binary 1 is represented by a positive pulse and binary 0 is represented by a Negative pulse.
NRZ bipolar format- Binary 0 is represented by no pulse and binary one is represented by the alternative positive and negative pulse.
Manchester format-Binary 0: The first half bit duration negative pulse and the second half Bit duration positive pulse. Binary 1: first half bit duration positive pulse and the second half Bit duration negativepulse.

PART *B

1

Summarise the process of quantization and obtain the expression for signal to quantization ratio in the case of uniform quantizer(13M) BTL2 (**Nov/Dec2016,April/May17)**
Answer: Page 8.11 – K.Muralibabu, Signal to quantisation ratio Page 8.18 – K.Muralibabu
DEFINITION: (2M)
The conversion of an analog sample of the signal into a digital form is called the quantizing process. The quantizing process has a twofold effect.
2) The peak to peak range of input sample is subdivided into a finite set of decision levels (or) decision thresholds that are aligned with the „risers‟ of the staircase and
2) The output is assigned a discrete value selected from a finite set representation levels (or) reconstruction levels that are aligned with „treads‟ of the staircase.
2 types of quantizers. (4M)
1) Uniformquantizer
2) Non uinifomquantizer
1) Uniform quantizer:
In uniform quantization.as in figure 1(a), the separation between the I thresholds and the separation between the representation levels of the quantizer have a common value called the step size.
2 types of uniform quantizers
1) Symmetric quantizer of midtreadtype.
2) Symmetric quantizer of midrisetype.
1. Mid treadtype::
According to the staircase-like transfer characteristics of figure 1a, the decision thresholds of the quantizer are located at ±Δ/2, ±3Δ/2, ±5Δ/2..... and representation levels are located at 0, ±Δ, ±2Δ, where Δ is a step size. Since the origin lies in the middle of tread of the staircase, it is referred to as symmetric quantizer of midtread type.
2. Mid risertype:
Figure 2(a) shows staircase-like transfer characteristics in which the decision thresholds of the quantizer are located at 0, ±Δ, ±2Δ, and the representation levels are located at ±Δ/2, ±3Δ/2, ±5Δ/2.....where Δ is a stepsize..

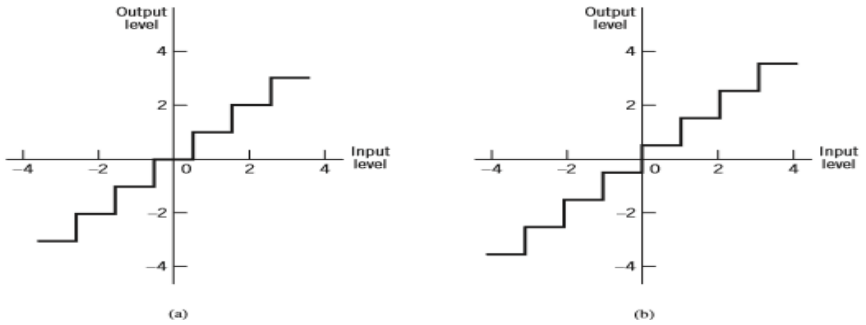


Fig Two types of quantization: (a) mid tread and (b) midrise.

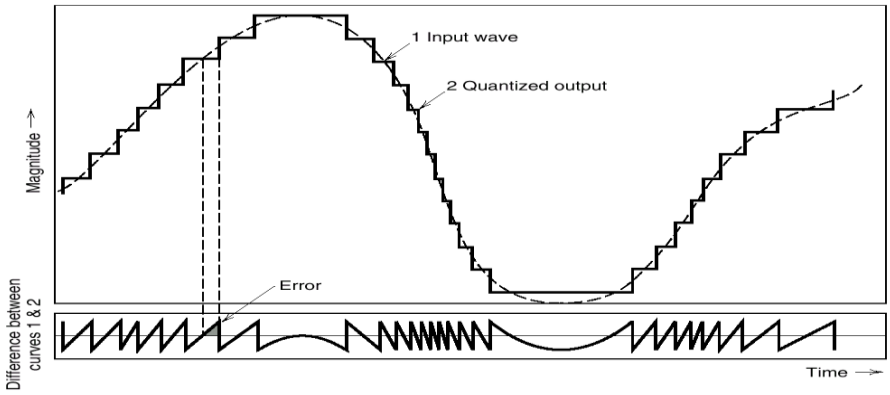
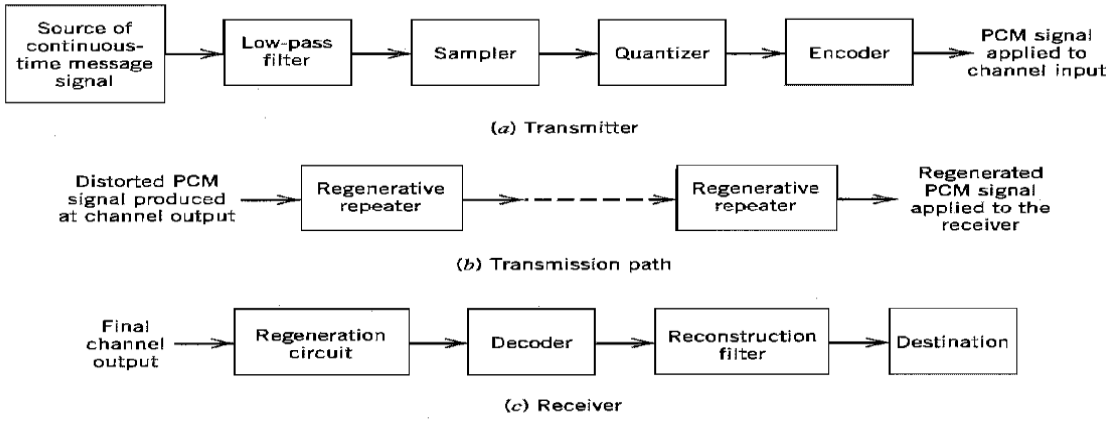
<table>
<tr><td></td><td>


Fig: Illustration of the quantization process
Diagrams (3M)
Signal to quantisation ratio (4M)
</td></tr>
<tr><td>2</td><td>
Using block diagram, explain the functioning of each block present in a PCM transmitter and receiver setup.(Nov/Dec 2017) (13M))BTL 4
Answer : Page : 267 – H Taub, D L Schilling
• Definition : (ADC process -transmitted as a serial bit stream) (2M)


(4M)
• Sampler : $f_s \geq 2f_m$ (2M)
• Quantizer : process of making signal discrete in amplitude - sampled signal to the nearest predefined or representation level (2M)
• Encoder : encode the discrete set of samples (1M)
Companding : Combination of compression of signal in transmitter – expansion at the receiver – companding (2M)
</td></tr>
<tr><td>3</td><td>
Describe the process of sampling & how the message is reconstructed from its samples. Also illustrate the effect of aliasing with neat sketch.(13M) BTL3
(Nov/Dec2015)
Answer: Page 8.2 – K.Muralibabu
Sampling Theory (2M)
The sampling process is usually described in the time domain, it is an operation
that is basic to digital signal processing and digital communications. Through use of
the sampling process, an analog signal is converted into a corresponding sequence of samples that are usually spaced uniformly in time. The sampling rate is properly
chosen in relation to the bandwidth of the message signal, so that the sequence of
</td></tr>
</table>

samples uniquely defines the original analog signal.

Frequency-Domain Description of Sampling **(6M)**

Consider an arbitrary signal g(t) of finite energy, which is specified for all time t. A segment of the signal g(t) is shown in Figure 6a. Suppose that we sample the signal g(t) instantaneously and at a uniform rate, once every Ts seconds. Consequently, we obtainaninfinitesequenceofsamplesspacedTssecondsapartanddenotedby {g(nTs)}, where n takes on all possible integer values, positive as well as negative. We refer to Ts as the sampling period, and to its reciprocal fs = 1/Ts as the sampling rate. For obvious reasons, this ideal form of sampling is called instantaneous sampling.

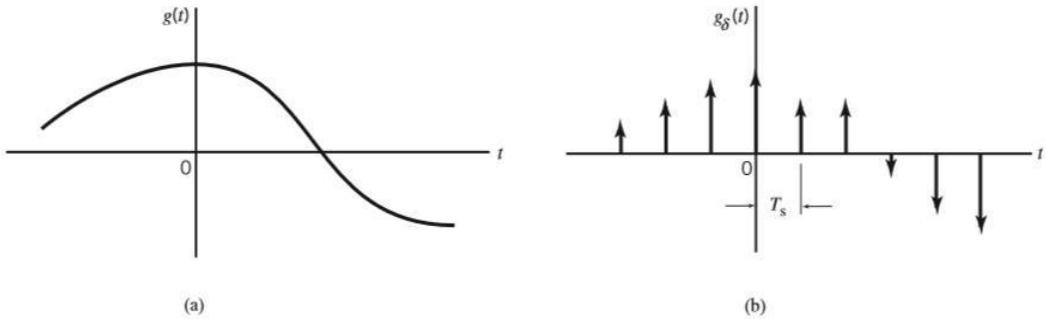


Figure. The sampling process. (a) Analog signal. (b) Instantaneously sampled version of the analog signal.

$$g_\delta(t) = \sum_{n=-\infty}^{\infty} g(nT_s)\delta(t - nT_s)$$

.

$$g_\delta(t) \rightleftharpoons f_s \sum_{m=-\infty}^{\infty} G(f - mf_s)$$

(5M)

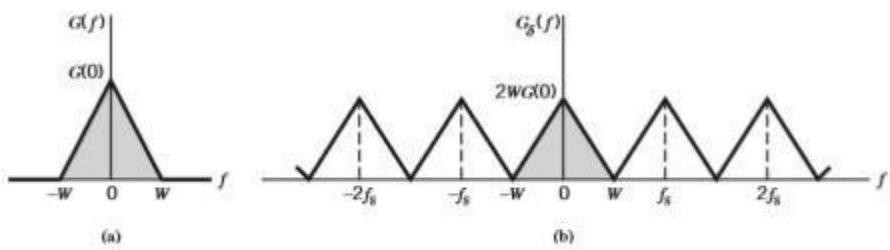


Fig: (a) Spectrum of a strictly band-limited signal g(t). (b) Spectrum of the sampled version of g(t) for a sampling period Ts =1/2W.

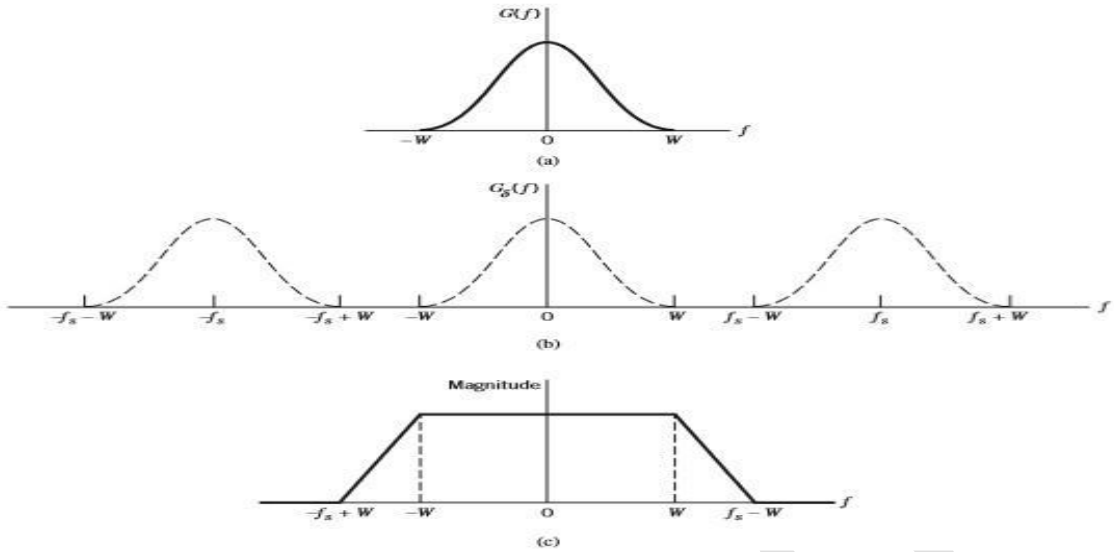


Figure 9 (a) Anti-alias filtered spectrum of an information-bearing signal. (b) Spectrum of instantaneously sampled version of the signal

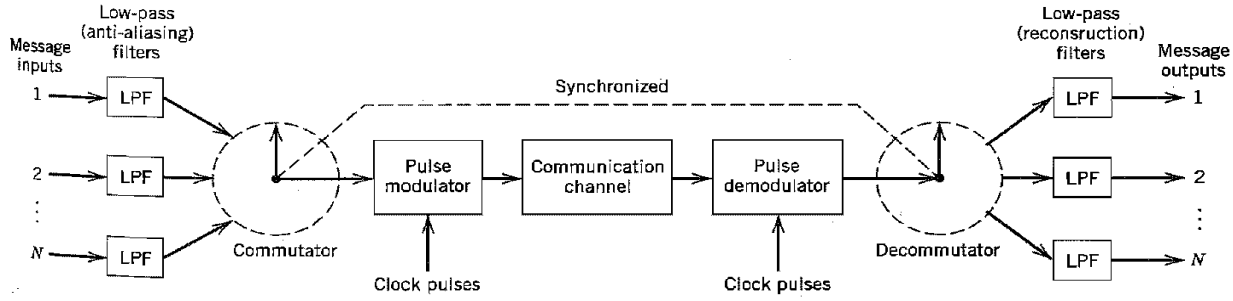
4

What is TDM? Explain the difference between analog TDM and digital TDM. (13M) (May 2016) BTL 4

Answer: Page 162 – S. Haykin

Time Division Multiplexing: Transmission of number of signals – single channel – time slots (2M)

Block diagram : (3M)



Low Pass Filter : Removes noise components (6M)

Commutator : Sampling – Interleaving

Pulse Modulator : Pulse amplitude Modulation

Channel : Wired – Wireless channel

Pulse Demodulator : Pulse Amplitude Demodulation

Decommutator : Disinter leaving – Reconstruction

Applications : RF Communication – Military Applications

Difference between analog / digital TDM :

Analog TDM -sampling and quantizer block – included.

Digital TDM – sampling and quantizer block – included. (2M)

5

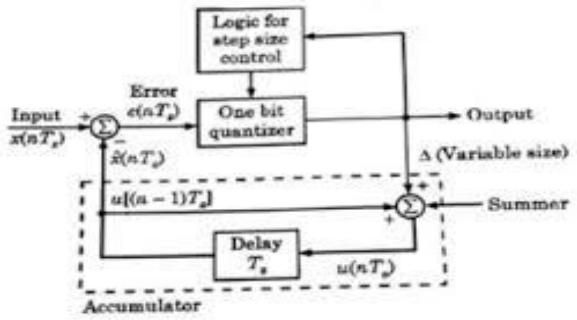
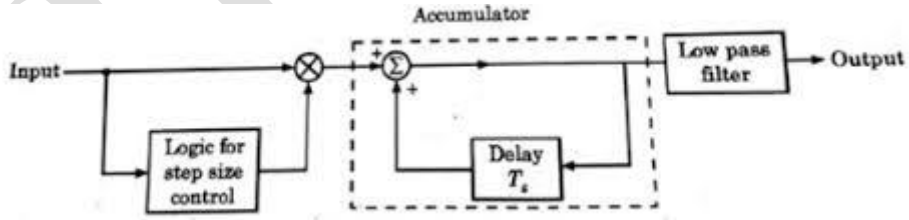
Explain about different types of delta modulation schemes. (MAY/JUNE 2014) (13M) BTL2

Answer : Page : 289-295 – H Taub, D L Schilling

Adaptive delta modulation :

Definition : The performance of delta modulator – improved significantly- making the step size modultor – time varying form i.e the step size of the modulator increased, input – varying rapidly – step size of the modulator – decreased, input varying slowly- adaptive modulation – reducing the quantization errors -ADM

(2M)

	ADM transmitter – receiver block diagram (2M+2M) **Transmitter** : logic for step control (1M) • Step size doubled – one bit quantizer output – high • Step size reduced – one bit quantizer output – low (2M) • Receiver : 1.for each incoming bit step size – produced 2. The step size -decided- both previous, present inputs (2M) • Advantages : SNR better than DM -reduction in slope overload distortion – granular noise • Better utilization – BW compared DM • Low signalling rate (2M)
6	**Explain how Adaptive Delta Modulation performs better and gains more SNR than delta modulation.[Nov/Dec 2016,April/May 2017] (13M)**BTL1 **Answer: Page 9.10 – K.Muralibabu** **Definition** **(2M)** The performance of the delta modulator can be improved significantly by making the step size of the modulator (assume time-varyingform). • During a steep segment of the input signal the step size isincreased. • Conversely when the input signal is varying slowly, the step size isreduced. • In this way, the step size is adapted to the level of the input signal is called adaptive delta modulation(ADM). • Several ADM schemes to adjust stepsize 1) Discrete set of values is provided for the stepsize. 2) Continuous range for step size variation isprovided. **TRANSMITTER** **(4M)** **BLOCK DIAGRAM**  **RECEIVER** **(4M)**  **Advantage and disadvantage** **(3M)** **Adv:** low signaling Less complicated **Disadv:**Slope overload error Granular noise
7	**Explain DPCM transmitter andreceiver.(April/May 2017) (13M)**BTL1 **Answer: Page 9.1 – K.Muralibabu** **Definition** **(2M)**

The difference in the amp of two successive samples is transmitted rsther than actual sample
Transmitter **(4M)**

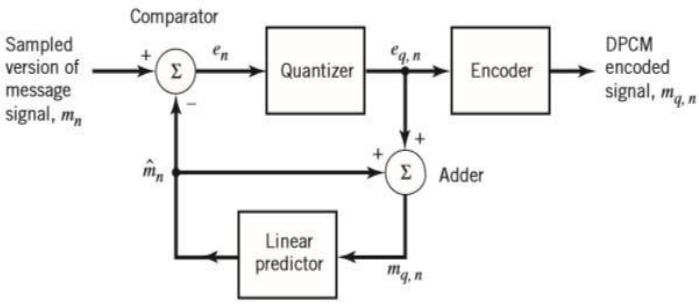


FIG:Block diagram of transmitter
Reciever **(4M)**

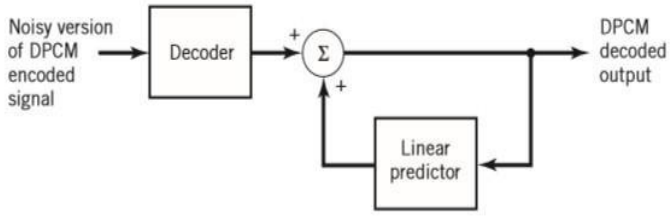


Fig:block diagram of receiver
Advantage and disadvantage **(3M)**
Adv:less number of quantisation
Disadv: Practical usage is limited

8

Explain ADPCM and illustrate adaptive quantization with forward estimation (AQF) and adaptive quantization with backward estimation(AQB)
(or)
Illustrate how the adaptive time domain coder codes the speech at low bit rate and compare it with frequency domain coder. [NOV/DEC 2015] (13M) BTL3
Answer: Page 9.13 – K.Muralibabu
Definition **(2M)**
ADPCM is a variant of DPCM that varies the size of the quantisation step to allow further reduction of the required data bw for a given SNR

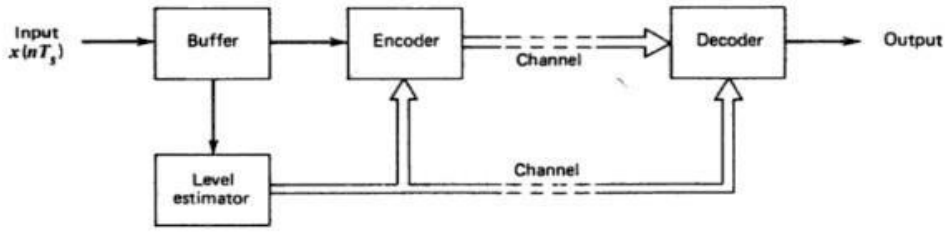


Fig: adaptive quantization with forward estimation (AQF) **(3M)**

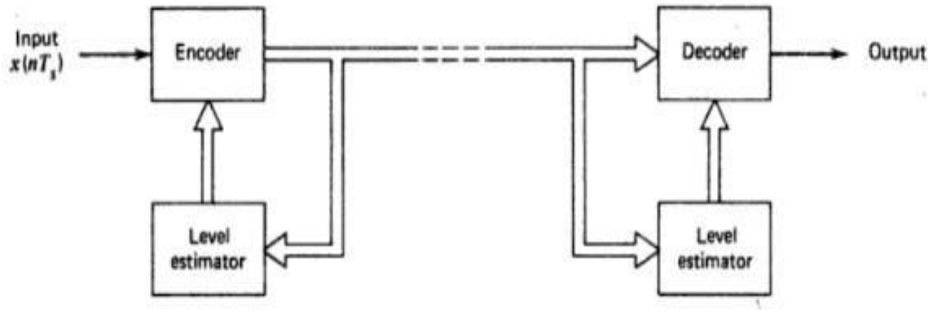


Fig: adaptive quantization with backward estimation (AQB) **(3M)**

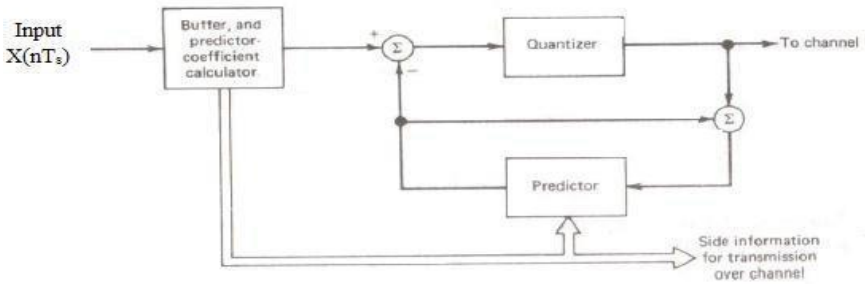


Fig:Adaptive prediction with forward estimation(APF) **(3M)**

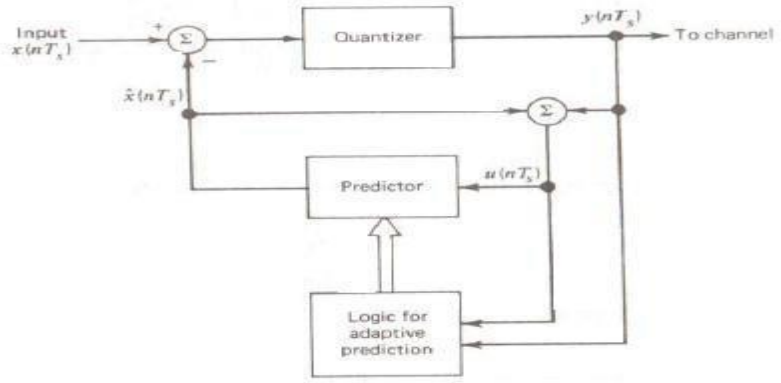


Fig:Adaptive prediction with backward estimation(APB) **(2M)**

9	**Determine the power spectral density of NRZ polar and unipolar data formats.[NOV/DEC2015,April/May 2017](13M)** BTL1 **Answer: Page 10.1 – K.Muralibabu** **Unipolar format (on-off signaling)** **(7M)**

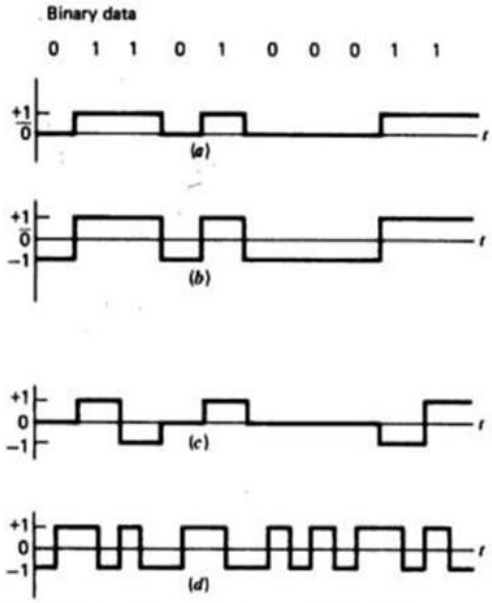


Figure Binary data formats. (*a*) Nonreturn-to-zero unipolar format. (*b*) Nonreturn-to-zero polar format. (*c*) Nonreturn-to-zero bipolar format. (*d*) Manchester format.

POLAR FORMAT **(6M)**

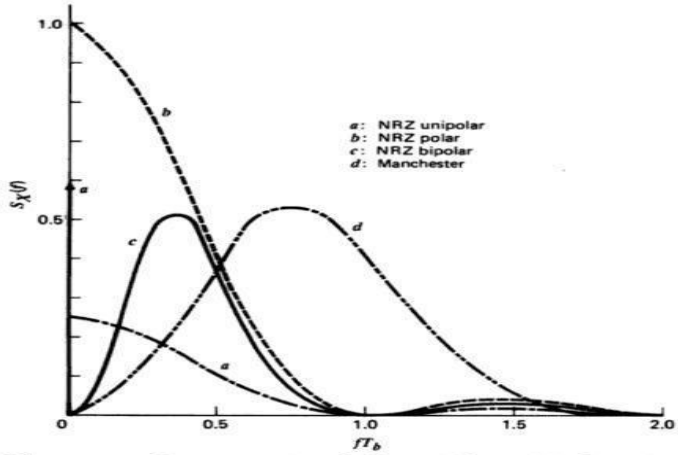


Figure Power spectra of different binary data formats.

UNIT III PULSE MODULATION

Phase shift keying – BPSK, DPSK, QPSK – Principles of M-ary signaling M-ary PSK & QAM – Comparison, ISI – Pulse shaping – Duo binary encoding – Cosine filters – Eye pattern, equalizers

PART*A

1

Distinguish BPSK, QAM and QPSK techniques. Write the expression for the signal set of QPSK (MAY/JUNE 2016),(NOV/DEC2015),(April/May 2017) BTL4

BPSK-Phase of the carrier is shifted between two values according to input bit sequence(1,0).

QAM-The information carried is contained in both amplitude and phase of the transmitted carrier. Signals from two separate information sources modulate the same carrier frequency at the same time. It conserves theB.W.

QPSK-The information carried by the transmitted wave is carried in the phase. Phase of carrier takes place on one of the four values[π/4,3π/4,5π/4,7π/4].Two successive bits of data sequence are groupedtogether.

2

Distinguish coherent and non-coherent reception.(May/June16,Nov/Dec16) BTL4

S.NO	Coherent detection	Non-coherent detection
1	Local carrier generated at the receiver is phase locked with the carrier at the transmitter.	Local carrier generated at the receiver not be phase locked with the carrier at the transmitter.
2	Synchronous detection	Synchronous detection is not Possible
3	Low probability of error	High probability of error
4	Complex in design	Simple in design

3

Explain how QPSK differs from PSK in term of transmission bandwidth and bit Information it carries? BTL4

For a given bit rate 1/Tb, a QPSK wave requires half the transmission bandwidth of the corresponding binary PSK wave. Equivalently for a given transmission bandwidth, a QPSK wave carries twice as many bits of information as the corresponding binary PSKwave.

4

List out the applications ofQAM. BTL1

Stereo broadcasting of AMsignals

Encoding color signals in analog TV broadcastingsystem.

Used inmodems

Used in digital communicationsystem.

5

Give the two basic operation of DPSKtransmitter. BTL4

Differential encoding of the input binarywave.

Phase –shift keying hence, the name differential phase shiftkeying.

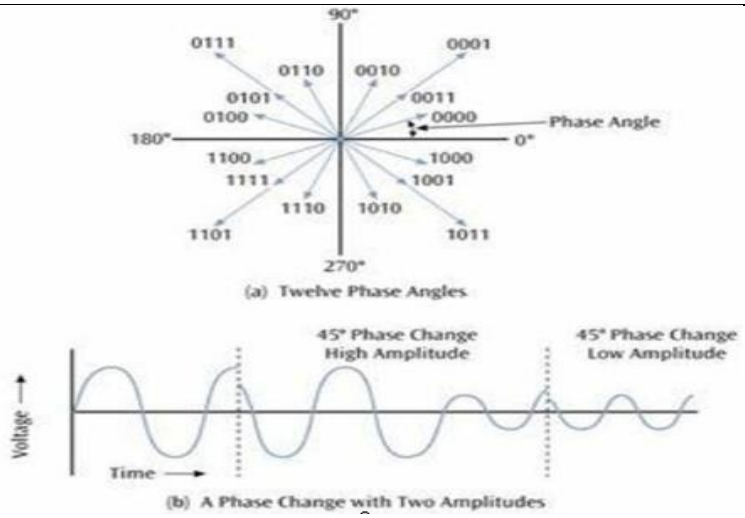
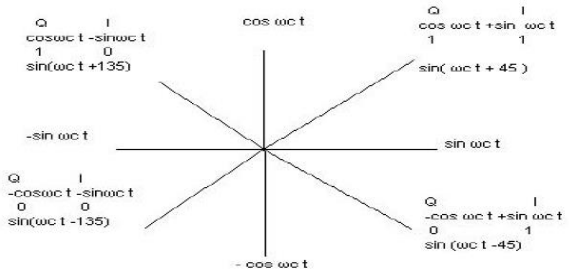
6

DefineBER [MAY14] BTL1

The signal gets contaminated by several undesired waveforms in channel. The net effect of all these degradations causes error in detection. The performance measure of this error is called Bit error rate.

7

Draw the constellation diagram of QAM. [NOV 10, MAY 13, NOV14] BTL2

	
8	**What is ISI and give its causes.** BTL1 The ringing tails of several pulses have overlapped, thus interfering with major pulse lobe. This interference is commonly called as intersymbol interference or ISI. The four primary causes of ISI are i.Timing inaccuracies ii.Insufficient bandwidth iii.Amplitude distortion iv.Phase distortion
9	**What is an eye pattern?** BTL1 The performance of a digital transmission system can be measured by displaying the received signal on an oscilloscope and triggering the time base at data rate. Thus, all waveform combinations are superimposed over adjacent signaling intervals. Such a display is called eye pattern or eye diagram.
10	**List the significance of eye pattern.** BTL2 It used to determine the effects of degradations introduced into pulses as they travel to the regenerator. It discloses any noise or errors in line equalization. It also gives the amount of ISI present
11	**What is binary phase shift keying? NOV/DEC 2011 NOV/DEC2012** BTL1 With Binary phase shift keying two phases are possible for the carrier. One phase represents a logic 1 and the other phase represents a logic 0 As the input digital signal changes state (from 1 to 180)
12	**What are the advantages of QPSK? APRIL?MAY 2010,** APRIL?MAY 2011 **NOV/DEC 2012** BTL1 a.All signal points placed on circumference of circle b.Circuit is simple c.Noise immunity is high. **d.Error probability is less then AQSK**
13	**Draw the phasor diagram of QPSK NOV/DEC 2009** BTL2 
14	**Compare binary PSK with QPSK.** BTL4

S.NO	BPSK	QPSK
1	One bit forms a symbol.	Two bits form a symbol.
2	Two possible symbols	Four possible symbols.
3	Minimum bandwidth is	Minimum bandwidth is

		twice of f b	equal to f b .
	4	Symbol duration = Tb.	Symbol duration = 2Tb.

15 **Bring out the difference between DPSK and BPSK.** BTL4

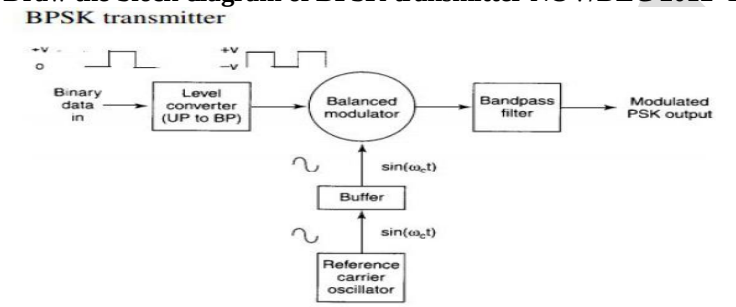
S.NO	DPSK	BPSK
1	It does not need a carrier at its receiver	It needs a carrier at receiver
2	Bandwidth reduced compared to BPSK	More bandwidth.
3	Probability of error or bit error rate more than BPSK	Comparatively low
4	Error propagation more, since it uses two bits for its reception	Comparatively low, since it uses only single bit
5	Noise interference more	Comparatively low

16 **What are the advantages of M-ary signaling scheme?** BTL1

i. M-ary signaling schemes transmit bits at a time.

Bandwidth requirement of M-ary signaling schemes is reduced

17 **Draw the block diagram of BPSK transmitter NOV/DEC 2012** BTL1



18 **Define QAM.** BTL1

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.

19 **Define inter symbol interference (ISI). NOV/DEC 2011** BTL1

The ringing tails of several pulses have overlapped, thus interfering with major pulse lobe. This interference is commonly called as intersymbol interference or ISI. The four primary causes of ISI are

i. Timing inaccuracies

ii. Insufficient bandwidth

iii. Amplitude distortion

iv. Phase distortion

20 **Define duo binary system?what are the drawbacks of it?** BTL1

DUO implies doubling of the transmission capacity of straight binary system.It is a practical means of achieving the maximum signaling rate.

Drawbacks:

- Tails decay rate is faster than Nyquist channel
- Once errors are made,they tend to propagate through the output

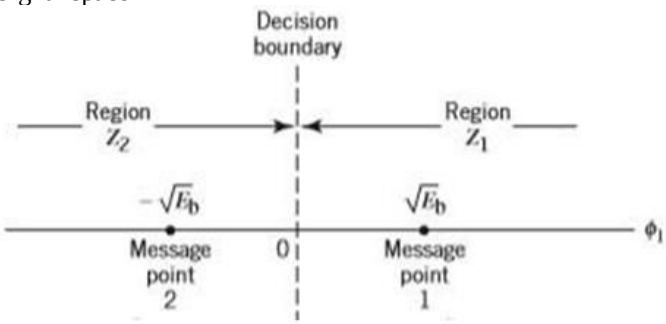
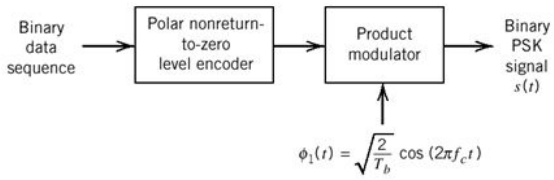
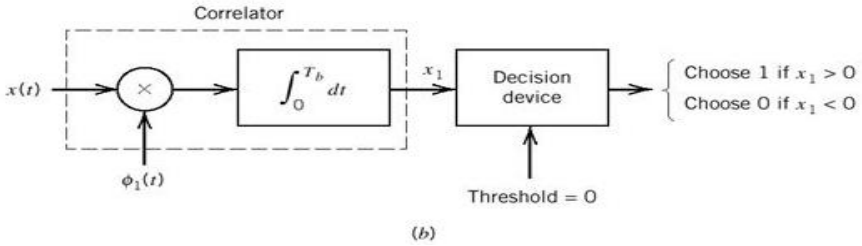
21 **Why do we needadaptive equalization in switched telephone network** BTL4

The fixed pair of transmitting and rreceiving filters used in an average channel characteristics may not adequately reduce,ISI.To realize the full transmission capability of a telephone channel,there is a need for adaptive equalization

22 **Define raised cosine filter** BTL1

It is a filter frequently used for pulse-shaping in digital modulationdue to its ability to minimize intersymbol interference. Its name stems from the fact that the non-zero portion of the frequency spectrum of its simplest form is a cosine function, 'raised' up to sit above the axis

23 **What is pulse shaping:** BTL1

	Pulse shaping is the process of changing the waveform of transmitted**pulses**. Its purpose is to make the transmitted signal better suited to its purpose or the communication channel, typically by limiting the effective bandwidth of the transmission.
24	**What M-ary signaling** BTL1 An *M*-ary transmission is a type of digital modulation where instead of transmitting one bit at a time, two or more bits are transmitted simultaneously. This type of transmission results in reduced channel bandwidth.
25	**What is M ary PSK?** BTL1 **M-ary** Phase Shift Keying - or MPSK - is a modulation where data bits select one of **M** phase shifted versions of the carrier to transmit the data. Thus, the **M** possible waveforms all have the same amplitude and frequency but different phases. The signal constellations consist of **M** equally spaced points on a circle.
	PART*B
1	**Explain the transmitter, receiver and signal space diagram of BPSK [may/June 2016,April /May 2017] (13M)**BTL1 **Answer: Page 11.3 - K.Muralibabu** **Definition** **(2M)** With Binary phase shift keying two phases are possible for the carrier. One phase represents a logic 1 and the other phase represents a logic 0 As the input digital signal changes state (from 1 to 180) Signal space (3M)  Fig: Signal space diagram for coherent binary PSK system Transmitter (4M)  **Fig:** Binary PSK transmitter Receiver (4M)  Fig: Coherent PSK receiver
2	**Explain the transmitter, receiver and signal space diagram of QPSK [nov/dec 2015,2016] (13M)**BTL1 **Answer: Page 11.14 - K.Muralibabu** **Definition** **(2M)** **Two successive bits in a input bit stream are combined together to form a message and each message is represented by a distinct value of phase shift of the carrier.So the signaling rate and BW are reduced.**

Signal space (3M)

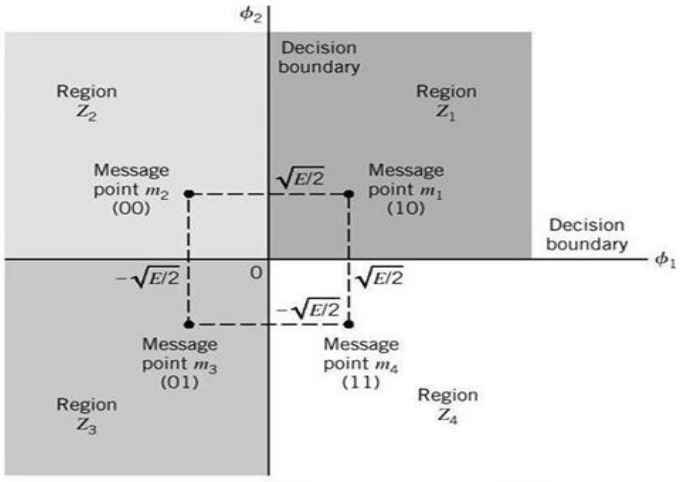


Fig:Signal space diagram for coherent QPSK system

Transmitter (4M)

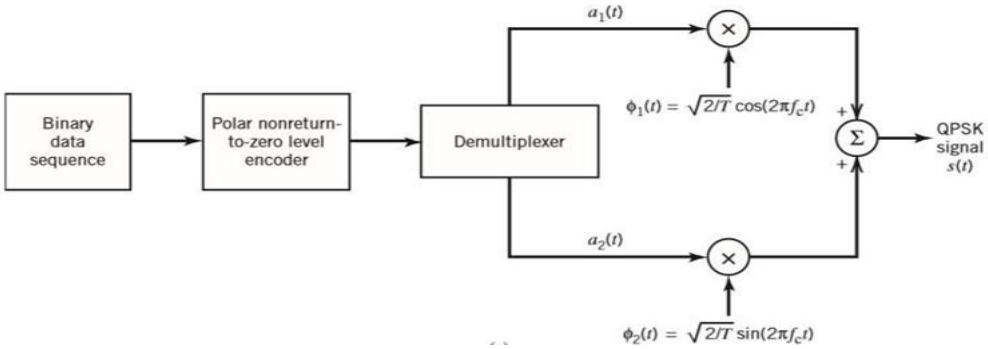


Fig:QPSK Transmitter

Receiver **(4M)**

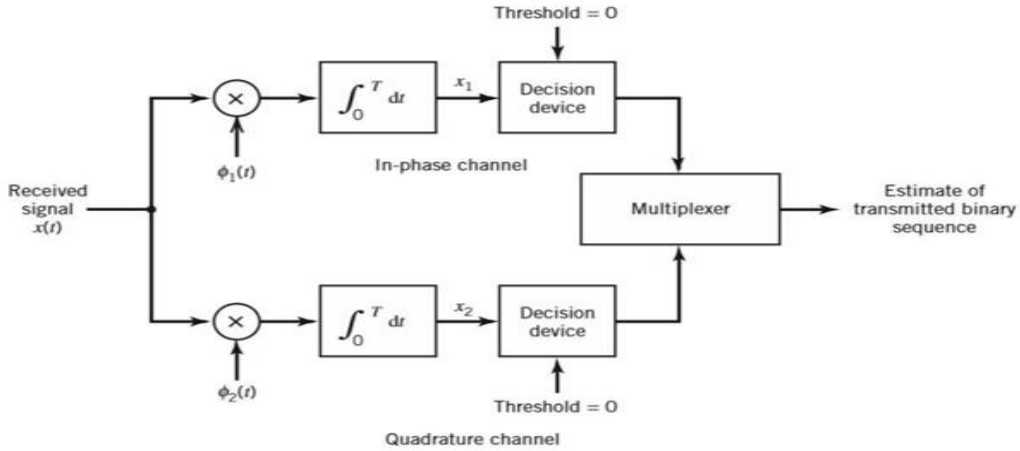


Fig:QPSK receiver

3 **Explain the transmitter, receiver and signal space diagram of DPSK (13M)** BTL1

Answer: Page 11.8 - K.Muralibabu

Signal space (2M)

Differential Phase Shift Keying is the non-coherent version of the PSK. It eliminates the need for coherent reference signal at the receiver by combining two basic operations at thetransmitter

(1) Differential encoding of the input binary waveand (3M)

Phase shiftkeying

$\{b_k\}$		1	0	0	1	0	0	1	1
$\{\bar{b}_k\}$		0	1	1	0	1	1	0	0
$\{d_{k-1}\}$		1	1	0	1	1	0	1	1
$\{\bar{d}_{k-1}\}$		0	0	1	0	0	1	0	0
$\{b_k d_{k-1}\}$		1	0	0	1	0	0	1	1
$\{\bar{b}_k \bar{d}_{k-1}\}$		0	0	1	0	0	1	0	0
Differentially encoded sequence $\{d_k\}$	1	1	0	1	1	0	1	1	1
Transmitted phase (*radians*)	0	0	π	0	0	π	0	0	0

Fig:Illustrating the generation of DPSK signal

Transmitter (4M)

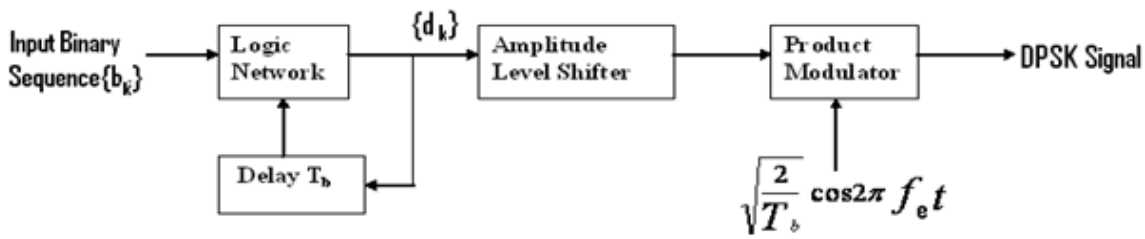


Fig: Block diagram for DPSK Transmitter

Receiver (4M)

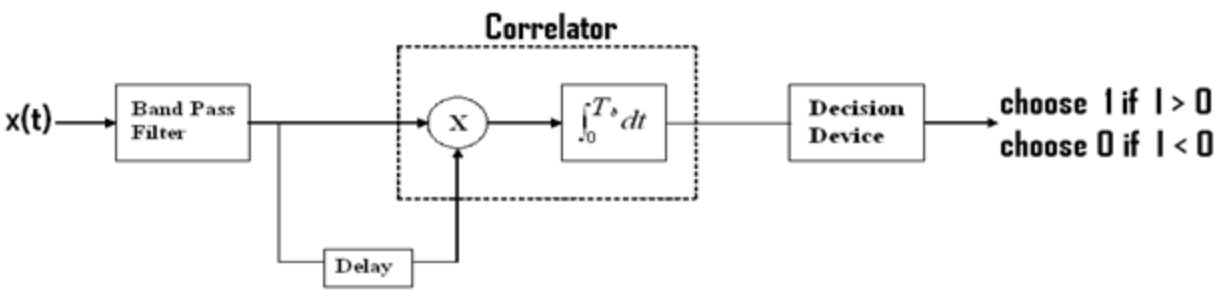


Fig: Block diagram for DPSK Receiver

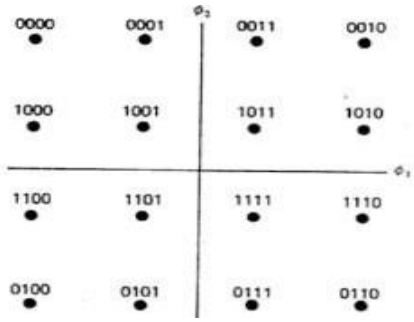
4

Explain the transmitter, receiver ofQAM. (13M)BTL1

Answer: Page 11.24 - K.Muralibabu

Signal space (2M)

In a M-ary PSK system, in phase and quadrature components of the modulated signal are interrelated in such a way that the envelope is constrained to the main constant. This constrained manifests itself in a circular constellation for the message points. However if this constraint is removed, and the in phase and quadrature components are thereby permitted to be independent. We get a new modulation scheme called M-ary quadrature modulation (QAM)scheme.



(3M)

FIG: Signal constellation of M-ary QAM for M=16

Transmitter **(4M)**

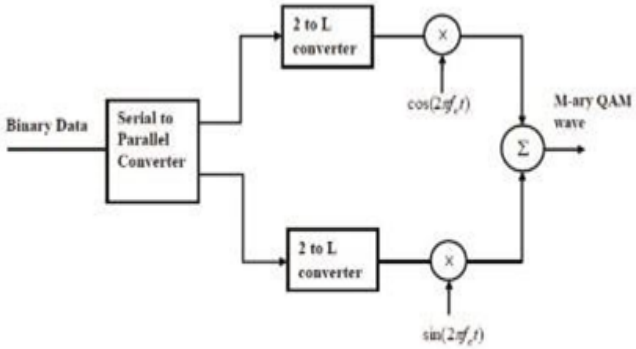


Fig: Block diagram of M-ary QAM Transmitter

Receiver **(4M)**

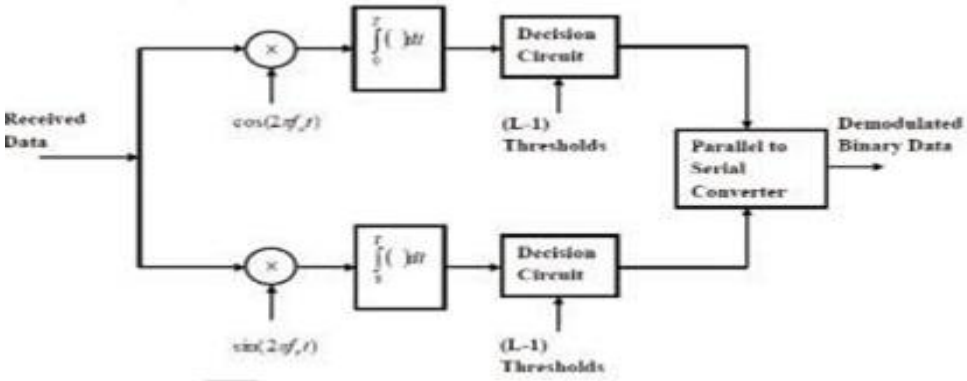


Fig: Block diagram of M-ary QAM Receiver

5 **Write short notes on Eye pattern & Inter symbol interference [Nov/Dec 2015,2016](13M)** BTL1

Answer: Page 12.16 - K.Muralibabu

Eye pattern: **(2M)**

Eye patterns can be observed using an oscilloscope. The received wave is applied to the vertical deflection plates of an oscilloscope and the saw tooth wave at a rate equal to transmitted symbol rate is applied to the horizontal deflection plates, resulting display is eye pattern as it resembles human eye. The interior region of eye pattern is called eye opening.

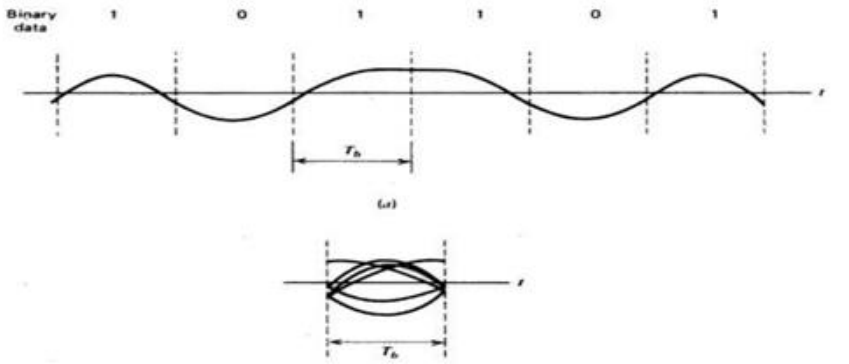
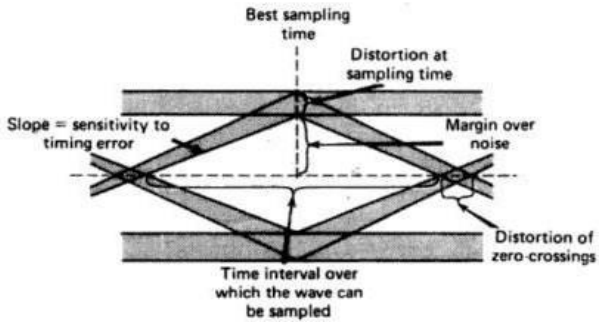


Fig : (a) Distorted binary wave (b) Eye pattern



(5M)

Fig: Interpretation of Eye pattern

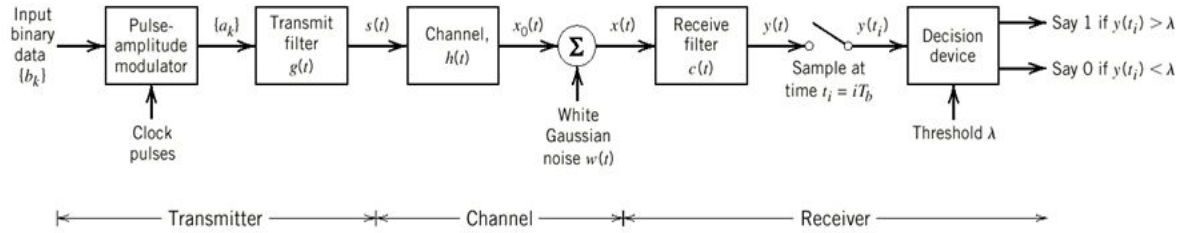
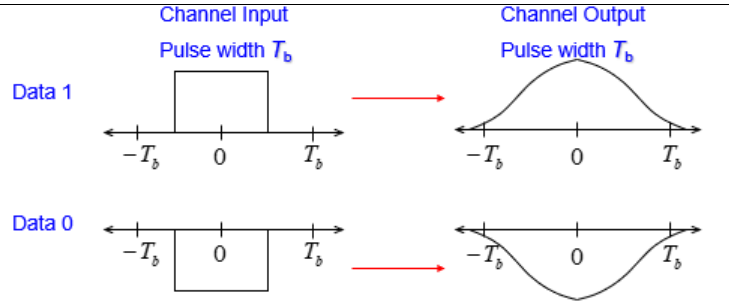


Fig: Base band binary data transmission system (6M)

6	**Write short notes on (13M)**BTL **3** • **ISI (4M)** • **Duo binary coding (5M)** • **Eye pattern (4M)** **Answer : Page. : 252,243,261 H Taub, D L Schilling** **Definition :** • Distortion -one symbol interferes-subsequent symbols-previous symbols-noise-reliable-presence -ISI-errors -decision device-receiver output • Adapative equalization -error correcting codes- reduces -ISI (2M) **Waveform :**



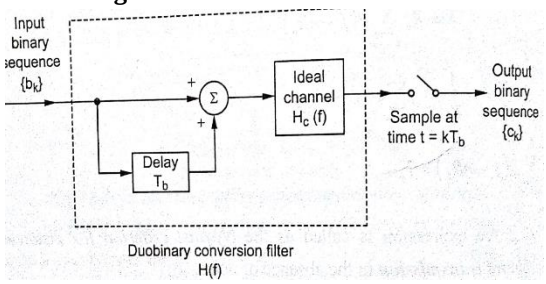
(2M)

Duo binary coding

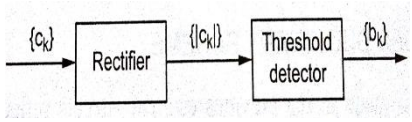
Duo binary -transmission capacity -simple binary transmission system (2M)

Binary input sequence- uncorrelated -duration Tb- 0 –(-1) – 1 –(+1)

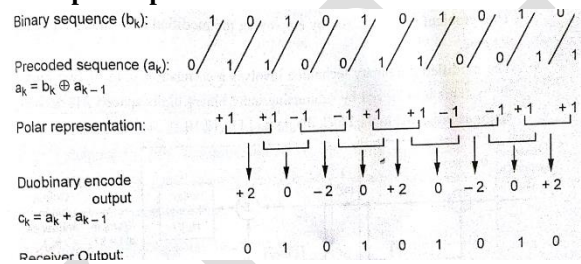
Block diagram : (1M)



Decoded diagram : (1M)

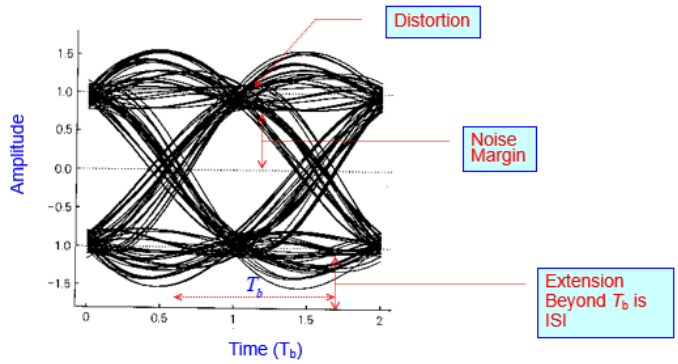
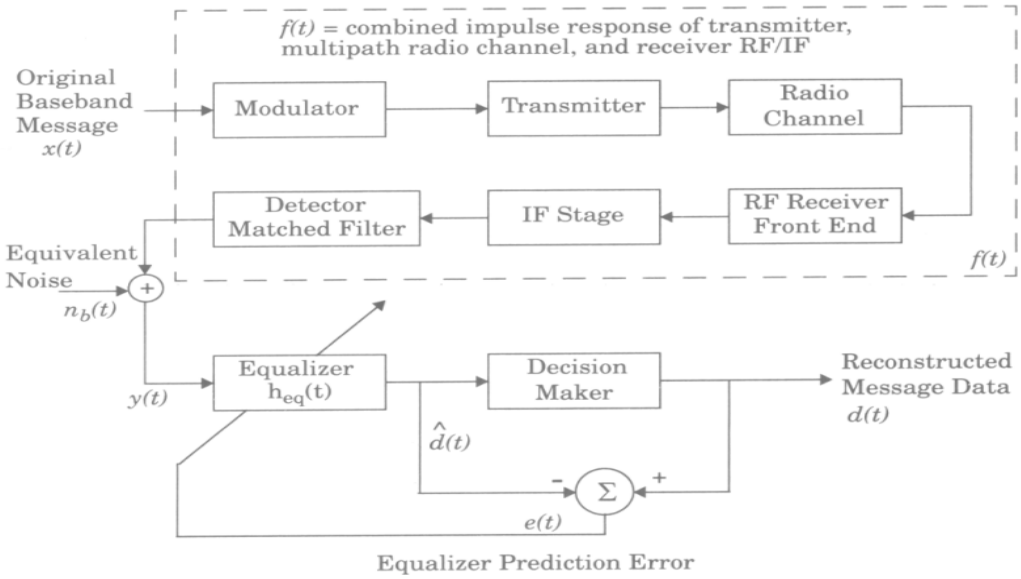


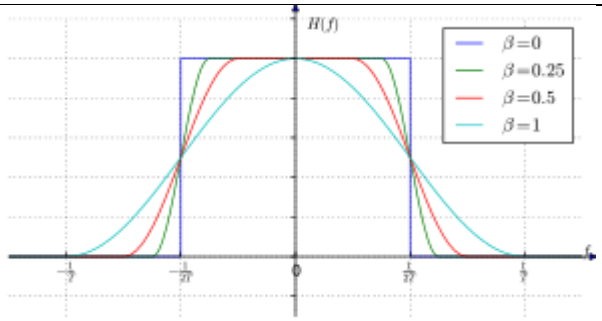
Example output waveform: **(2M)**



Eye pattern :

Displayed – screen-CRO-study ISI-Effect of PCM-shape -pattern resembles-human eye-eye pattern (2M)

	 (2M)
7	**Explain adaptive equalization and also explain about two operation modes of adaptive equalization(13M)**BTL1 **Answer: Page 12.21 - K.Muralibabu** **Definition** **(2M)** **In a mobile cellular environment the characcteristics of the wireless dispersive fading channel change randomly with time.In order for an equalizer to effectively combat ISI,the equalizer coefficients should change according to the channel status so as to track the channel variations**  **(4M)** **Fig:ADAPTIVE EQUALIZER** **Adaptive Equalizer operation:** **(7M)** • **Training mode** • **Decision directed mode**
8	**Summarize cosine filter in detail** BTL2 **Answer: Page 12.13- K.Muralibabu** **Definition** **(2M)** The **raised-cosine filter** is a filter frequently used for pulse-shaping in digital modulation due to its ability to minimise intersymbol interference (ISI). Its name stems from the fact that the non-zero portion of the frequency spectrum of its simplest form (β=1) is a cosine function, 'raised' up to sit above the f axis.



(5M)

Fig:frequency responce

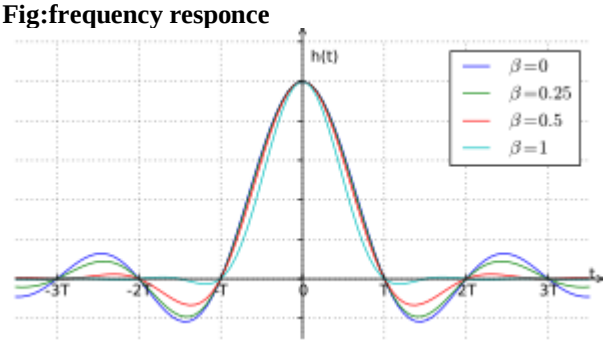
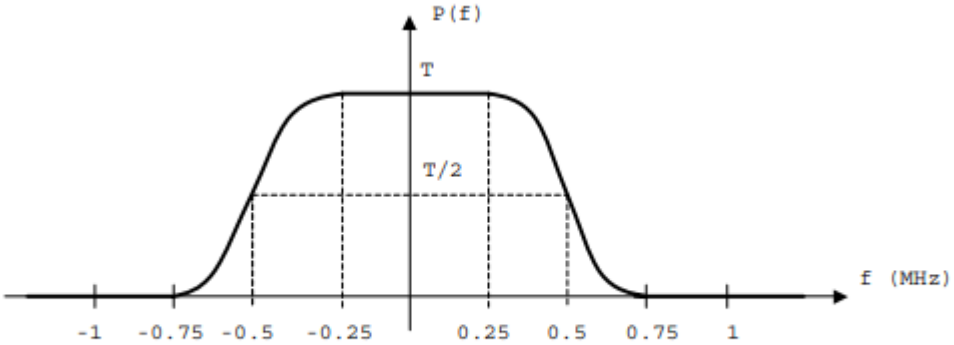


Fig:impulse response **(6M)**

PART*C

1 **Binary data at a rate of 4×10^6 bits/s are transmitted using M-PAM over a bandlimited channel, using the raised cosine roll-off characteristic shown in the figure below:**



Raised cosine roll-off characteristic.

(a) **Determine the roll-off factor α and the value of M of this system.** BTL1 **(8M)**

(b) **(b) Signals are transmitted over a bandlimited AWGN channel. Determine the minimum required bit energy-to-noise ratio, Eb/N0, to achieve a probability of a bit error less than 10−3. Express your result in decibels (dB).** BTL1 **(7M)**

Solution:

(a) $R_b = 4$ Mbits/s. Note that $1/2T = 0.5$ MHz, the point of symmetry of the raised cosine. As a result, $1/T = 1$ MHz. Since $R_b = 1/T_b = \log_2 M\ (1/T)$, it follows that $\log_2 M = 4$ and therefore $M = 16$.

To find the roll-off factor, note that $2\alpha(1/2T) = 0.75 - 0.25 = 0.5$ MHz (the region of the cosine characteristic) and therefore, with $1/2T = 0.5$ MHz, we obtain $\alpha = 0.5$.

(b) The probability of a bit error for M-PAM is

$$P[\epsilon] = \frac{M-1}{M}\, Q\left(\sqrt{\frac{6\log_2 M}{M^2-1}\frac{E_b}{N_0}}\right).$$

For $M = 16$,

$$P[\epsilon] = \frac{15}{16}\, Q\left(\sqrt{\frac{24}{255}\left(\frac{E_b}{N_0}\right)_{\min}}\right) < 1\times 10^{-3}$$

From the table of the Q-function, we have $Q(3.07) = 0.0011$. Therefore,

$$\sqrt{\frac{8}{85}\left(\frac{E_b}{N_0}\right)_{\min}} = 3.07 \quad\longrightarrow\quad \left(\frac{E_b}{N_0}\right)_{\min} = \frac{85}{8}\,(3.07)^2 = 100.14 = 20\ \text{dB}$$

2

In a digital communication system, the bit rate of a bipolar NRZ data sequence is 1 Mbps and carrier frequency is 100 MHz. Design by determining the symbol rate of transmission and the bandwidth requirement of the communications channel for BPSK,QPSK & DPSK System. (15M)BTL6

SOLUTION:

Given **(3M)**

fb=1MBPS=1*106

Tb=1/fb=1/2*106

BPSK

BW=$2f_b$

Symbol rate=$1/T_b$ (4M)

QPSK

BW=f_b

Symbol rate=$1/2T_b$ (4M)

DPSK

BW=2fb

Symbol rate=$1/T_b$ (4M)

3

Describe in detail about the operation of M ary- PSK(any one type) with neat diagram (15M) BTL1

Answer : Page: (11.20-Murali babu. K)

Definition : (2M)

Signal modulation technique - eight levels - phase shifts -supports three bits per symbol- 8-PSK, (n=3, M=8) to produce eight different phases- incoming bits - encoded in group -three- called tribits- producing eight different output phases.

Types : 8 PSK ,16PSK (1M)

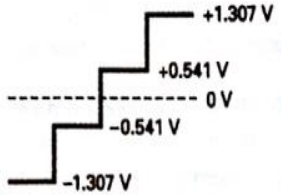
8PSK :

Transmitter and receiver block diagram : (4M)

TRUTHTABLE : (1M)

I	C	Output
0	0	−0.541 V
0	1	−1.307 V
1	0	+0.541 V
1	1	+1.307 V

Q	$\bar{C}$	Output
0	1	−1.307 V
0	0	−0.541 V
1	1	+1.307 V
1	0	+0.541 V

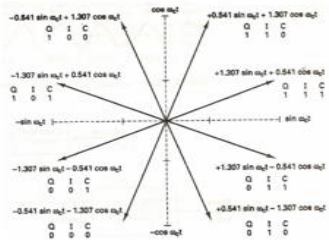


Phasor And Constellation Diagram : (2M)

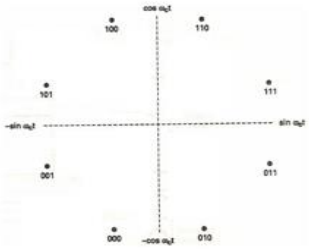
Truth Table

Binary input Q I C	8-PSK output phase
0 0 0	−112.5°
0 0 1	−157.5°
0 1 0	−67.5°
0 1 1	−22.5°
1 0 0	+112.5°
1 0 1	+157.5°
1 1 0	+67.5°
1 1 1	+22.5°

Phase diagram



constellation diagram



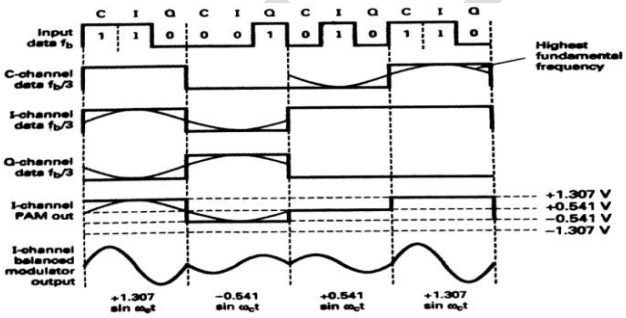
BANDWIDTH : (2M)

Binary input highest fundamental frequency is $f_a = \frac{f_b}{6}$

Output of balanced modulator is given by

$$8-PSK_{output} = \sin(2\pi f_a t)\sin(2\pi f_c t) = \sin(2\pi\left(\frac{f_b}{6}\right)t)\sin(2\pi f_c t)$$

Output Waveform : (1M)



16 PSK

Definition: (2M)

M-ary encoding- M=16- 16 different output phases-four bits (quadbits)- combined, producing 16 different output phases. The angular separation between adjacent output phases - 22.50. Baud rate - 1/4thof bit rate & Bandwidth - input bit rate

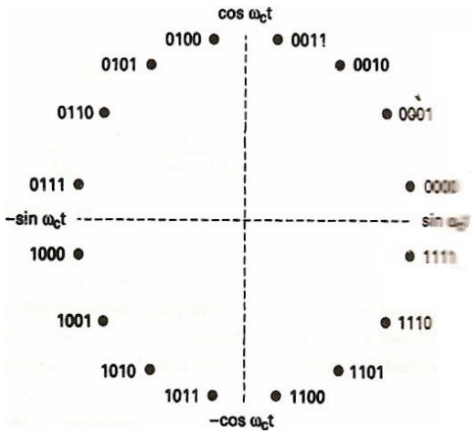
Types : 8 PSK ,16PSK (1M)

Transmitter and receiver block diagram : (4M)

Truth table : (2M)

Bit code	Phase	Bit code	Phase
0000	11.25°	1000	191.25°
0001	33.75°	1001	213.75°
0010	56.25°	1010	236.25°
0011	78.75°	1011	258.75°
0100	101.25°	1100	281.25°
0101	123.75°	1101	303.75°
0110	146.25°	1110	326.25°
0111	168.75°	1111	348.75°

Constellation diagram : (2M)



Merits: (2M)

PSK scheme provides reasonable data rate within minimum bandwidth.

Demerits:

More error – receiver side

UNIT IV INFORMATION THEORY AND CODING	**9**
Measure of information – Entropy – Source coding theorem – Shannon–Fano coding, Huffman Coding, LZ Coding – Channel capacity – Shannon-Hartley law – Shannon's limit – Error control codes – Cyclic codes, Syndrome calculation – Convolution Coding, Sequential and Viterbi decoding	
PART*A	
Q.No	Questions
1	**Define entropy(Nov-Dec 2014)** BTL1 The entropy of a source is a measure of the average amount of information per source symbol in a long message. $H=\sum_{i=1}^{\infty} pilog2(\frac{1}{pi})$
2	**State the properties of entropy.** BTL1 a) For sure event or impossible event entropy is zero. b) For M number of equally H max = log 2 M likely symbols, entropy is log2 M c) Entropy is lower bound on average number of bits per symbol.
3	**Define information rate** BTL1 If the message generated from source at the rate of r message per second, the information rate is defined as R=rH Average no of bits of information per second
4	**Define coding efficiency** BTL1 Coding efficiency is defined as the ratio of minimum of average code word length and average code word length H=Lmin/L’
5	**What is mutual information?(Apr-May 2015)** BTL1 It is defined as the amount of information transferred when Xi is transmitted and Yi is received. It is represented by I(Xi,Yi) The average mutual information is defined as the amount of source information gain per received symbol
6	**What is channel capacity?** BTL1 Channel capacity is defined as the maximum of the mutual information that may be transmitted through the channel. C=max I(X;Y)
7	**Define code rate** BTL1 The ratio k/n is called code rate. It is denoted as r R=k/n where r<1.
8	**What are the errors controls coding?** BTL1 • Linear block codes • Cyclic codes • Convolution codes
9	**Find out the hamming distance and hamming weight of a given code C1= 1001, C2= 1010** BTL4 Hamming distance: D(C1,C2)= 1 0 0 1 | 1 0 1 0 | =2 Hamming weight: C1=2 C2=2
10	**Define hamming distance and hamming weight.** BTL1 The Hamming distance d(C1,C2) between such a pair of code vectors is defined as the number of locations in which their respective elements differ.The hamming weight of a code vector C is defined as the number of non zero elements

	in the code vector.
11	**What is information theory?** BTL1 Information theory deals with the mathematical modelling and analysis of a communication system rather than with physical sources and physical channels.
12	**Define bandwidth efficiency. (MAY/JUNE-2010) (NOV/DEC-2010)** BTL1 The ratio of channel capacity to bandwidth is called bandwidth efficiency. i.e,Bandwidth
13	**What is channel redundancy?** BTL1 Redundancy (γ) = 1 – code efficiency Redundancy should be as low as possible
14	**Name the two source coding techniques.** BTL2 **The source coding techniques are,** a) Prefix coding b) Shannon-fano coding c) Huffman coding
15	**Give the expressions for channel capacity of a Gaussian channel.(MAY/JUNE 2016)** BTL2 Channel capacity of a Gaussian channel is given as, $C = B \log 1+S^2$ bits / sec N Here B is Channel bandwidth S is signal power N is total noise power within the channel bandwidth.
16	**What is prefix code?** BTL1 In prefix code, no code word is the prefix of any other code word. It is variable length code. The binary digits (code words) are assigned to the messages as per their probabilities of occurrence.
17	**State Shannon Hartley theorem of channel capacity.(MAY/JUNE 2013,2014) (NOV/DEC 2011. 2013,2014, NOV/DEC 2016)** BTL3 The channel capacity of a additive Gaussian noise channel. C=Blog2(1+S/N) Where B-channel bandwidth S/N -Signal to Noise ratio
18	**State source coding theorem.(nov/dec2015)** BTL3 The source coding theorem shows that it is impossible to compress the data such that the code rate (average number of bits per symbol) is less than the Shannon entropy of the source, without it being virtually certain that information will be lost. However it is possible to get the code rate arbitrarily close to the Shannon entropy, with negligible probability of loss.
19	**What is the entropy of a binary memory-less source?** BTL1 The entropy of a binary memory-less source $H(X)=-p_0 \log_2 p_0-(1-p_0)\log_2(1-p_0)$ p_0 probability of symbol„0‟,$p_1=(1-p_0)$ =probability of transmitting symbol „1‟.
20	**List the properties of Hamming distance. (NOV/DEC 2014)** BTL1 The Hamming distance is a metric on the set of the words of length n (also known as a Hamming space), as it fulfills the conditions of non-negativity, identity of indiscernibles and symmetry, and it can be shown by complete induction that it satisfies the triangle inequality as well.
21	**State the property of entropy.(May/June 2015)** BTL3 1. $\text{Log}M \geq H(x) \geq 0$ 2. $H(X) = 0$ if all probabilities are zero 3. $H(X) = \log_2 M$ if all probabilities are equal
22	**Give the relation between the different entropies.** BTL4 H(X,Y)=H(X)+H(Y/X)=H(Y)+H(X/Y) H(X)- entropy of the source(Y/X),H(X/Y)-conditional entropy H(Y)-entropy of destination H(X,Y)- Joint entropy of the source and destination

23	**Define viterbi coding** BTL1 The Viterbi-Algorithm is a basic part of the coding and modulation method of thedigital data transmission. With the help of the Viterbi-algorithm it is. possible to recognise data errors and. correct them at the receiver
24	**Define L-Z coding** BTL1 Lempel-Ziv Source Coding The basic idea is that if we have a dictionary of 2A source phrases (Available at both the encoder and the decoder) in order to encode on of these phrases one needs only A binary digits. • Normally, a computer stores each symbol as an ASCII character of 8 binary digits. (Actually only 7 are needed) • Using L-Z encoding, far less than 7 binary digits per symbol are needed. Typically the compression is about 2:1 or 3:1. • Variants of L-Z codes was the algorithm of the widely used Unix file compression utility compress as well as gzip. Several other popular compression utilities also used L-Z, or closely related encoding. • LZ became very widely used when it became part of the GIF image format in 1987. It may also (optionally) be used in TIFF and PDF files. • There are two versions of L-Z codes. We will only discuss the "window" version.
25	**Define syndrome calculation** BTL1 A parity and/or syndrome generator generates a block parity check for the detection and/or correction of errors in a multi-channel digital data communication system using a linear code or a coset of such code in which data and parity bytes are intended to be digitally encoded in n by m bit data blocks to form a respective codeword in n parallel bytes of m bits in serial order of significance in the form of a codeword having n elements represented by respective bytes in the Galois field GF(2^m), such Galois field being defined by an m-order field generator polynomial in integral powers of z between z^0 and z^m, where z is the inverse of the delay operator z^{-1} of such Galois field.
26	**What is meant by convolution code?** BTL1 For every binary digit that enters the encoder, two *code* digits are outputs. Hence *code* rate = 1/2. In general, the *code* rate R_{code} = k / n. The constraint length K of a *convolutional code* is *defined* as the number of shifts over which a single message bit can influence the encoder output
	PART*B

1 **Five source messages are probable to appear as m1=0.4 m2=0.15 m3=0.15 m4=0.15 m5=0.15 find coding efficiency for (a)Shannon fano coding (b) Huffman coding.(Nov/Dec 2014) (13M)**BTL4

Refer model: Answer: Page 13.16- K.Muralibabu

$$H=\sum_{i=1}^{\infty} pi\log2\left(\frac{1}{pi}\right)$$

$$=0.4\log_2(1/0.4)+4\times 0.15\log_2(1/0.15)$$

$$=2.171 \text{ bits}$$

(2M)

(a)Shannon fano coding (4M)

Message	Probability	I	II	III	No of bits	Code Word
m1	0.4	0	0		2	00
m 2	0.15	0	1		2	01
m 3	0.15	1	0	0	3	100
m 4	0.15	1	0	1	3	101
m 5	0.15	1	1		2	11

Average code word length =0.48 ×2 +2×0.15+ 3×0.15+3×0.15+2×0.15
=2.3 bits. (2M)

Coding efficiency= $\frac{Average\ information/message}{Average\ code\ length} = \frac{2.17bits}{2.3bits}$ = 0.9439=94.39% (2M)

b)Huffman coding (2M)

Message	Probability	I	II	III	IV	Code word
m1	0.4	0.4	0.4	0.4	0.6 0	1
m 2	0.15	0.15	0.3	0.3 0	0.4 1	000
m 3	0.15	0.15	0.15 0	0.3 1		00
m 4	0.15	0.15 0	0.15 1			010
m 5	0.15	0.15 1				011

$\bar{L}=\sum_{k=0}^{k-1} lkpk$

$=0.4\times1+4\times0.15+3$

$=2.2$ bits.

(2M)

Coding efficiency= $\frac{2.17bits}{2.2bits}$ = 0.9868=98.68% (1M)

2

State and prove Shannon's noiseless coding theorem. (13M) (April 2018) (April 2014) BTL 2

Answer: Page 2.12 - K.Muralibabu

Shannon's second theorem or channel coding theorem

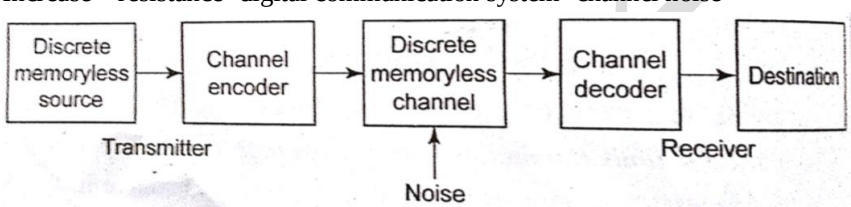
Need for channel coding : (3M)

Presence -noise- discrepancies – output and input data sequence

Error probability- 9 out of 10- transmitted bits are received- level of reliability

Probability error= 10^{-6} – even lower- channel coding

Increase – resistance -digital communication system -channel noise (3M)



Code rate :

$r=\frac{k}{n}=\frac{No\ of\ messages\ bits\ in\ a\ block}{no.of\ bits\ in\ code\ word}<1$ (1M)

Average information rate : (1M)

$=\frac{H(S)}{T_S}$ bits/sec

Channel capacity per units : (1M)

$=\frac{C}{T_C}$ bits/sec

Statement : (2M)

$\frac{H(S)}{T_S}\leq\frac{C}{T_C}$ (source output -transmitted – over -channel-reconstructed-small probability error)

$\frac{H(S)}{T_S}>\frac{C}{T_C}$ (no possibility -transmission- reconstructed-small probability error) (2M)

3

Explain how viterbi's decoding procedure is used for decoding convolution codes. (Apr/may 2015)(13M) BTL1

Answer: Page 15.28- K.Muralibabu

VITERBI ALGORITHM. **(2M)**

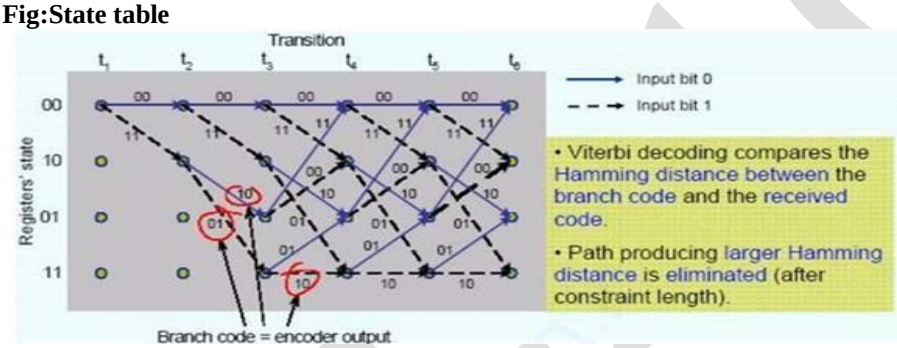
- ML algorithm is too complex to search all available paths.

- End to end calculation.
- Viterbi algorithm performs ML decoding by reducing its complexity.
- Eliminate least likely trellis path at each transmission stage.
- Reduce decoding complexity with early rejection of unlike paths.
- Viterbi algorithm gets its efficiency via concentrating on survival paths of the trellis.

Current state K1 k2	Message Bits (m)	C1=m1 ⊕k2	C2=m ⊕k1 ⊕k2	Next state K1 k2
a =0 0	0 1	0 1	0 1	0 0 = a 1 0 =b
b =1 0	0 1	0 1	1 0	0 1 = c 1 1 = d
c =0 1	0 1	1 0	1 0	0 0 =a 1 0 =b
d =1 1	0 1	1 0	0 1	0 1=c 1 1=d

(5M)

Fig:State table



(6M)

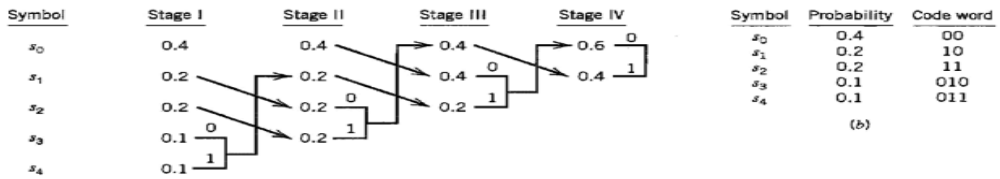
4 **Consider a linear block code with generator matrix**
G=[1 1 0 1 0 0 0; 0 1 1 0 1 0 0; 1 1 1 0 0 1 0; 1 0 1 0 0 0 1] (Dec-2016) (13m)BTL3

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. (4M)
iv) Calculate the syndrome for the received vector
r = [1 1 0 1 0 1 0] (3M)

Answer: Page: 4.6-K.Muralibabu

Parity Check matrix: H=$[P^T : I]$
Length: t $\leq d_{min} - 1$
Syndrome: S=rH^T
Original codeword: r ex-or e.

5 **Five symbols of the alphabet of discrete memory less source and their probabilities are given below.S={S0,S1,S2,S3,S4} P (S)=(0.4,0.2,0.2,0,1,0.1)Code the symbols using Huffman coding. (12) (AUC NOV/DEC 2010) Using Huffman code I, encode the following symbols.S = [0.3, 0.2, 0.25, 0.12, 0.05, 0.08,]Calculate average code length, entropy of the source, code efficiency, redundancy(13M)** BTL4
Answer: Page 13.31- K.Muralibabu



(4M)

The average code-word length is therefore

$$\bar{L} = 0.4(2) + 0.2(2) + 0.2(2) + 0.1(3) + 0.1(3)$$
$$= 2.2$$

(4M)

$$H(\mathcal{S}) = 0.4 \log_2\left(\frac{1}{0.4}\right) + 0.2 \log_2\left(\frac{1}{0.2}\right) + 0.2 \log_2\left(\frac{1}{0.2}\right) + 0.1 \log_2\left(\frac{1}{0.1}\right) + 0.1 \log_2\left(\frac{1}{0.1}\right)$$
$$= 0.52877 + 0.46439 + 0.46439 + 0.33219 + 0.33219$$
$$= 2.12193 \text{ bits}$$

Efficieny =96%

(5M)

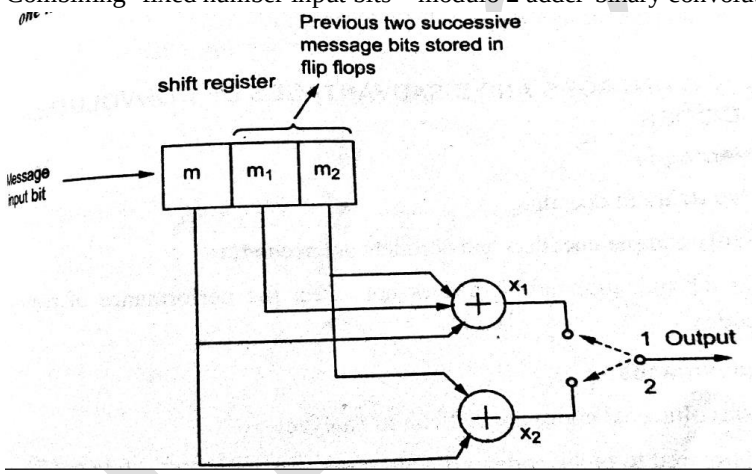
6 **Briefly discuss on various types of error control codes and explain in detail of convolutional codes with an example.** (13M) **(Nov 2016)** BTL 2

Explain briefly about linear and convolutional block codes with neat diagram with an example. (13M) **(April 2016)** BTL 1

Answer: Page 4.1 - K.Muralibabu

Define convolutional code :

Combining -fixed number input bits – modulo 2 adder-binary convolution – convolution coding (3M)

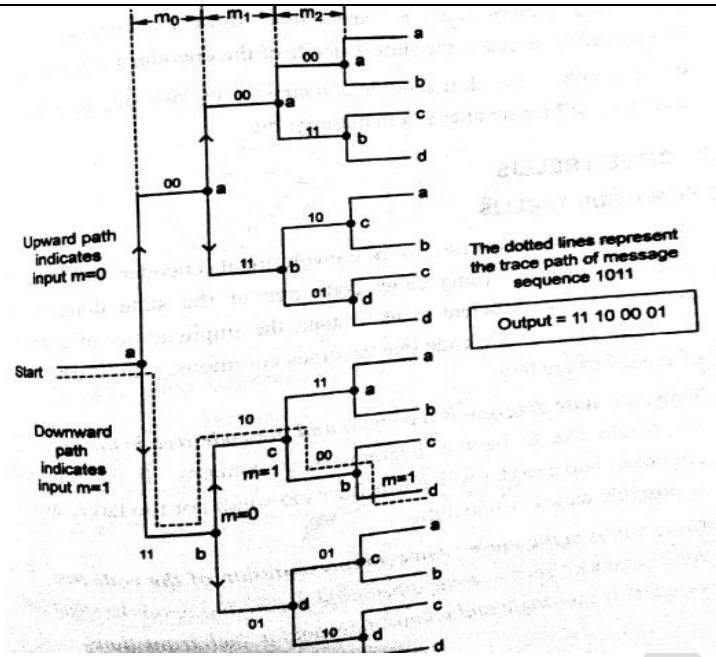


State table :

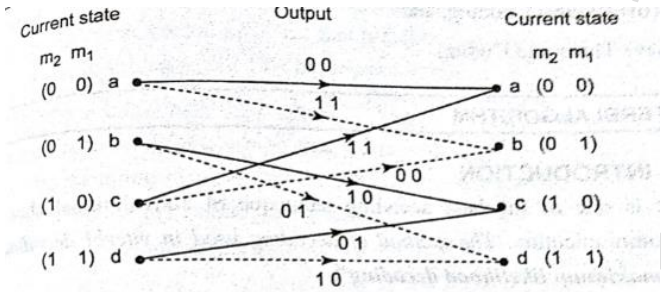
m_2	m_1	States of the encoder
0	0	a
0	1	b
1	0	c
1	1	d

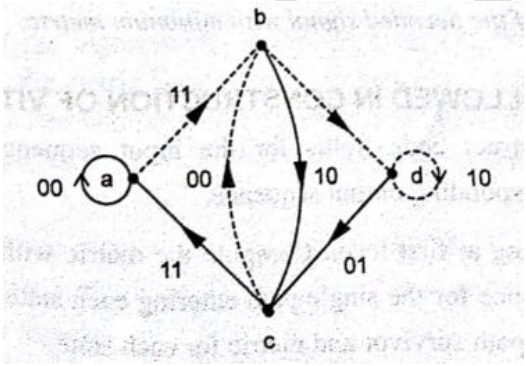
(2M)

Draw the code tree and state diagram :



TREE DIAGRAM : (6M)





(2M)

7	**Determine the generator polynomial g(X) FOR A (7, 4) cyclic code and fine the code vector for the following data vector 1010, 1111 and 1000 (13M)** BTL4 **Answer: Page 15.13- K.Muralibabu** n=7 k=4 q=n-k=3 To obtain the generator polynomial (p7+1) = (p+1)(p3+p2+1)(p3+p+1) Let G (p)= (p3+p+1) (3M) To obtain the generator matrix in systematic form

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & : & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & : & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & : & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & : & 0 & 1 & 1 \end{bmatrix}$$

To determine the code vector

1. code vector for M=1010

X=MG

X=[1010011] **(3M)**

2. code vector for M=1111

X=MG

X=[1111111] **(3M)**

3. code vector for M=1000

X=MG

X=[1000101] **(4M)**

8

What are cyclic codes. Explain the merits and demerits Give the properties of cyclic codes. (AU- May/June 2010) (13M)BTL1

Answer: Page 15.10- K.Muralibabu

Definition **(2M)**

Cyclic codes forms as subclass of linear block code.

Properties of cyclic code (5M)

1. Linearity property.

The sum of two code word in the code is also a code word.

2.Cyclic property

$$(c_{n-1}, c_0, \ldots, c_{n-2}),$$
$$(c_{n-2}, c_{n-1}, \ldots, c_{n-3}),$$
$$\vdots$$
$$(c_1, c_2, \ldots, c_{n-1}, c_0)$$

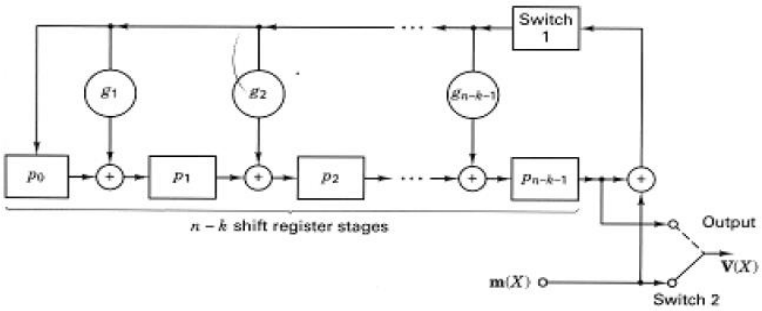
A cyclic shift of a code word in the code is also a code word.

GENERATOR POLYNOMIAL

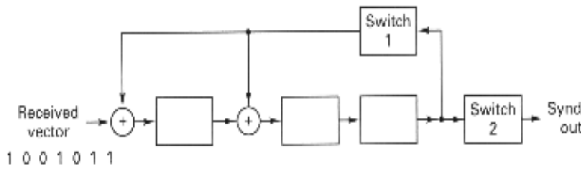
C(X) = a (X) g (X)

Encoding procedure **(4M)**

Encoder of cyclic code



Syndrome Calculator:



Adv and disadv **(2M)**

PART*C

1 **A discrete memory less source has an alphabet of seven symbols whose probabilities of occurrence are as described here**

Symbol	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
Probability	**0.25**	**0.25**	**0.125**	**0.125**	**0.125**	**0.0625**	**0.0625**

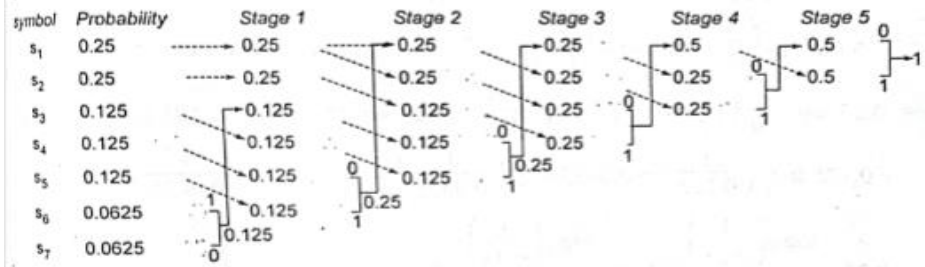
Compute i) shannaon fano coding ii) huffaman coding iii) calculate the efficiency.
(May 2016) (15M) BTL3
Answer : Page.2.3 - K.Murali Babu

Shannon fano coding & Huffaman coding : (6M)

Shannon Fano Code

Symbols	Probability	Stage 1	Stage 2	Stage 3	Stage 4	CW	Length
s_1	0.25 (0.5)	0 (0.25)	0			00	2
s_2	0.25	0 (0.25)	1			01	2
s_3	0.125 (0.5)	1 (0.25)	0 (0.125)	0		100	3
s_4	0.125	1	0 (0.125)	1		101	3
s_5	0.125	1 (0.25)	1 (0.125)	0		110	3
s_6	0.0625	1	1 (0.125)	1 (0.0625)	0	1110	4
s_7	0.0625	1	1	1 (0.0625)	1	1111	4

(iii) Huffman code



Symbol	Probability	Codeword	Length
s_1	0.25	10	2
s_2	0.25	11	2
s_3	0.125	001	3
s_4	0.125	010	3
s_5	0.125	011	3
s_6	0.0625	0000	4
s_7	0.0625	0001	4

$\bar{L} = \sum_{i=1}^{N} p_i L_i$ **(p1L1+ ---p7L7)** (3M)

$\bar{L}$ **=2.625bits /symbols**

Entropy :

$H(S) = \sum_{i=1}^{N} p_i \log_2\left(\frac{1}{p_i}\right)$ (3M)

$H(S)$ **= 2.625 bits/ sec**

Efficiency : (3M)

$\eta = \frac{H(S)}{\bar{L}}$

$\eta = 100\%$

2 **For a systematic (6,3) linear block code with parity matrix**

$$\begin{matrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{matrix}$$

Find all the possible code vectors. Construct the syndrome decoding table and decode the received code word 110001. (15M) BTL 3

Answer: Page: 4.6-K.Muralibabu

Generator matrix: G=[P:I_k] (3M)
Parity Check matrix: H=[$P^T : I$] (2M)
Length: t $\leq d_{min} - 1$ (2M)
Syndrome: S=rH^T (4M)
Original codeword: r ex-or e. (4M)

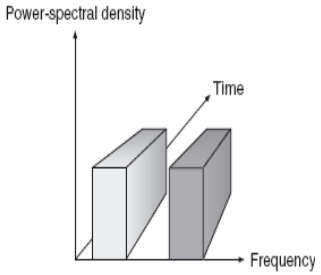
3	**The generator polynomial of a (7, 4) cyclic code is G (p) = p3+p+1. Find all the Code vectors for the code in non-systematic form. (15M)** BTL3 **Answer: Page: 4.12-K.Muralibabu** **n=7 k=4 q= n-k =3** **consider message M = (m3 m2 m1 m0)** **M(p) =m_3p^3 +$m_2p^2 + m_1p^1$+ m_0** **X(p)=M(p) G(p)= $p^5 + p^2 + p^1$+1**

UNIT V SPREAD SPECTRUM AND MULTIPLE ACCESS	
PN sequences – properties – m-sequence – DSSS – Processing gain, Jamming – FHSS – Synchronisation and tracking – Multiple Access – FDMA, TDMA, CDMA	
PART*A	
1	**List the spread spectrum techniques. NOV/DEC 2011** BTL1 ➢ Direct sequence spread spectrum with coherent Binary phase shift keying ➢ Frequency hop spread spectrum
2	**What is CDMA? NOV/DEC 2011** BTL1 In code division multiple access ,each subscriber is assigned a distinct spreading code(PN sequence),thereby permitting the subscriber full access to the channel all of the time.
3	**What are the applications of spread spectrum modulation?APRIL/MAY 2010, APRIL?MAY 2011** BTL1 1.Millitary applications. 2.secured communication
4	**Define processing gain in spread spectrum modulation. APRIL/MAY 2010, APRIL?MAY 2011** BTL1 Processing gain is defined as the gain in Signal to noise Ratio obtained by the use of spread spectrum . It is defined as the gain achieved by the processing a spread spectrum signal over an unspread signal .
5	**Define effective jamming power and processing. NOV/DEC 2009** BTL1 (Jamming margin) db = (processing gain) db -10log 10 (E b /N o) min Where (E b /N o) min is minimum value needed to support a prescribed Average probability of error .
6	**What is the principle of frequency hopping spread spectrum?** BTL2 **NOV/DEC 2009** The type of spread spectrum in which the carrier hops randomly from one frequency to another is called frequency hop spread spectrum
7	**Compare slow and fast frequency hopping.** BTL4

S.NO	**Slow Frequency Hopping**	**Fast Frequency Hopping**
1	More than one symbols are transmitted per frequency hop.	More than one frequency hops are required to transmit one symbol.
2	Chip rate is equal to symbol rate.	Chip rate is equal to hop rate.
3	Symbol rate higher than hop rate.	Hop rate higher than symbol rate.

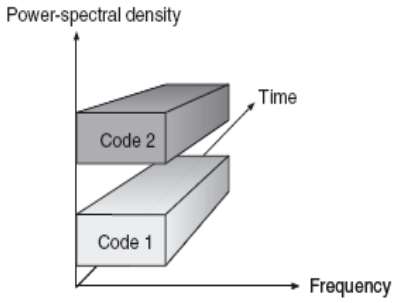
8	**What are the two different techniques used in speech coding for wireless communication?** BTL1 i. Multi-pulse excited Linear Predictive Coding (LPC). ii. Code-excited LPC
9	**What are the two function of fast frequency hopping?** BTL 1 1. Spread Jammer over the entire measure of the spectrum of transmitted signal. 2. Retuning the Jamming signal over the frequency band of transmitted signal
10	**What are the features of code Division multiple Accesses?** BTL1 1. It does not require external synchronization networks. 2. CDMA offers gradual degradation in performance when the no. of users is increased But it is easy to add new user to the system.
11	**Write some features of TDMA?** BTL1 *In TDMA , no. of time slots depends upon modulation technique,available bandwidth *Data transmission occurs in bursts It uses different time slots for transmission and reception, then duplexers are not required *Adaptive equalization is necessary *****Guard time should be minimized
12	**Write some features of CDMA?** BTL1 *In CDMA system, many users share the same frequency either TDD or FDD may be used *Channel data rate is high

	*Multipath fading may be substantially reduced *CDMA uses co –channel cells, it can use macroscopic spatial diversity to provide soft hand Off.
13	**What is near far effect in a CDMA system ? APRIL/MAY 2015** BTL1 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. The near-far problem is particularly difficult in CDMA systems, where transmitters share transmission frequencies and transmission time.
14	**What are Walsh codes.** BTL1 Walsh codes are orthogonal codes obtained from Hadamard matrices They are used in CDMA to separate the users
15	**Write some advantages of TDMA?** BTL1 Data transmission occurs in burst It uses different time slots for transmission and reception, then duplexers **are not**
16	**Write some advantages and disadvantages of CDMA** BTL1 **Advantages:** Multipath fading may be substantially reduced *CDMA uses co –channel cells, it can use macroscopic spatial diversity to provide soft hand Off. **Disadvantage:** Implementation complexity, need for power control, to avoid capture need for a large contiguous frequency band (for direct sequence) &problems installing in the field.
17	**Explain frequency hop spread spectrum(NOV/DEC 2015)** BTL1 In this technique, changing the carrier frequency in pseudo-random manner widens the spectrum of data modulated carrier.
18	**How many stages of flip-flops are required to generate PN sequence of length 31?** BTL3 N=2 $31 = 2^m - 1$ m = 5 stages
19	**Define code division multiple access.** BTL1 In code division multiple access ,each subscriber is assigned a distinct spreading code(PN sequence),thereby permitting the subscriber full access to the channel all of the time.
20	**What is effective jamming power .** BTL1 Jamming margin) db = (processing gain) db -10log 10 (E b /N o) min
21	**What is processing gain?** BTL1 Processing gain is defined as the gain in Signal to noise Ratio obtained by the use of spread spectrum . It is defined as the gain achieved by the processing a spread spectrum signal over an unspread signal **.**
22	**Is spread spectrum a modulation technique?** BTL4 Sometimes people call spread spectrum modulation. But that does not carry conventional meaning of modulation. Rather it includes conventional digital modulation techniques to generate spread spectrum modulated signals.
23	**Why pseudo-random code is used as special code for spreading the spectrum?** BTL2 Unintended receiver should not receive the signal. If the spreading code is not random, then unintended receiver can obtain the code by observing the signal over certain period of time. But if the code is random, then it is very difficult to identify it.
24	**What are the popular coding sequences of CDMA system. (NOV/DEC 2014)** BTL1 Popular code sequences used in spread-spectrum transmission are -Maximum Length sequences -Walsh Hadamard sequences -Gold codes, and -Kasami codes.

25	**Define slow frequency hopping. (NOV/DEC 2015)** BTL1 Several symbols of data are transmitted in one frequency hop. This means symbol rate is higher than hop-rate.
	PART*B
1	**What is a Pseudo noise sequence?How it is generated? What are the properties of Pseudo noise sequence? (8 Marks) APRIL/MAY 2010, APRIL?MAY2011, NOV/DEC2010 (13M)**BTL1 **Answer: Page 16.3- K.Muralibabu** ➢ Pseudo noise sequences – definition. (2M) A periodic binary sequence with a noise like waveform Maximum length sequence=N=2^m -1 Shift register diagram and explanation Rc=$\frac{1}{Tc}$ ➢ Generation: Feedback Shift registers – Block diagram with a sequence generation. (6M) ➢ Properties of PN sequences: (5M) - Balance property - Run property - Correlation property
2	**Explain about the multiple access scheme.(Apr/May 2010)(13M)** BTL1 **Answer: Page 17.2- K.Muralibabu** Three major techniques: (2M) ➢ Frequency division multiple access (FDMA) ➢ Time division multiple access (TDMA) ➢ Code division multiple access (CDMA) Others: ➢ Packet radio (PR) ➢ Space division multiple access (SDMA) **(4M)** **TDMA** Single carrier frequency - several users Not continuous - bursts. Handoff process - simpler Duplexers - not required. Transmission rates - High. Synchronization overhead - high **FDMA** **(4M)** Channel - only one phone circuit If Channel not in use - other users can't use Continuous transmission scheme Narrowband systems. Intersymbol interference - low. Mobile unit - duplexers. Requires RF filter - adjacent channel interference 

CDMA **(3M)**

Share the same frequency.
Soft capacity limit.
Frequency - dependent transmission impairments.
Multipath fading – reduced.
Channel data rates – high.
Macroscopic spatial diversity - soft handoff.



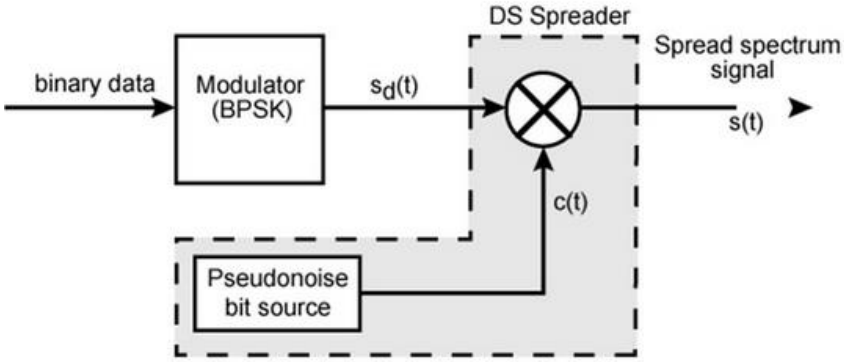
3

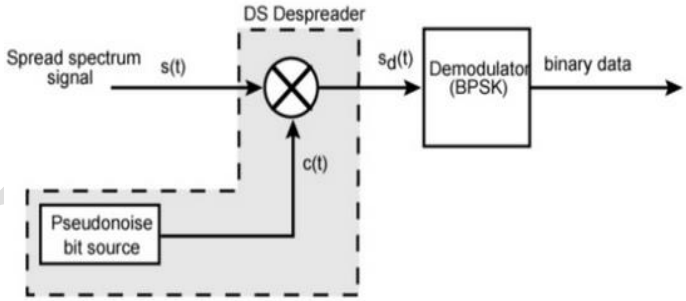
Explain the direct sequence spread spectrum techniques with neat block diagram. (13M) **(Nov 2017) (April 2016)** BTL2

Answer: Page: 1.6-K.Muralibabu

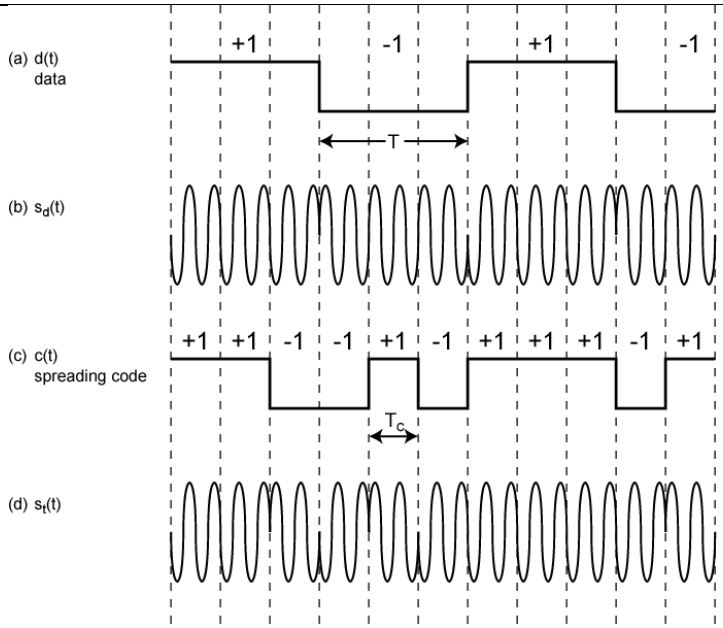
Definition : Spreading in transmitter and dispreading in receiver side (2M)

Model DS-bpsk transmitter and receiver diagram: (5M)





Transmitter and receiver output waveform: (3M)



Explanation: narrow band signal is converted into wideband with help of PN sequence and BPSK modulation technique. **(3M)**

4 **Describe the frequency hopping spread spectrum technique in detail. NOV/DEC 2011, NOV/DEC 2009 (13M)**BTL1

Answer: Page 16.14- K.Muralibabu

Definition **(2M)**

Process- randomly hopping modulated- carrier -one frequency -other – data -modulate- carrier.

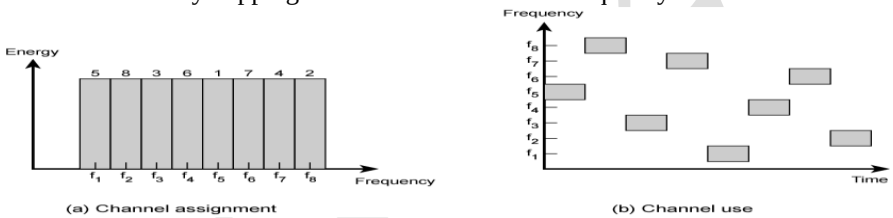


Fig:Example of FHSS

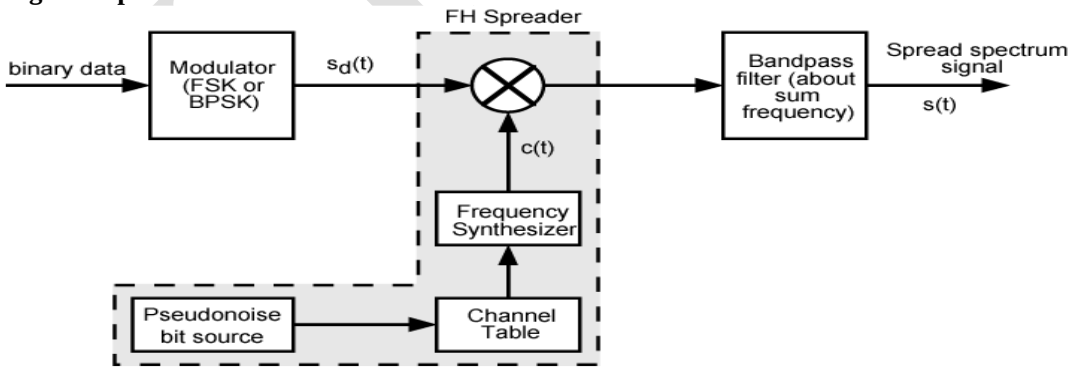


Fig: Frequency Hopping Spread Spectrum System (Transmitter) **(5M)**

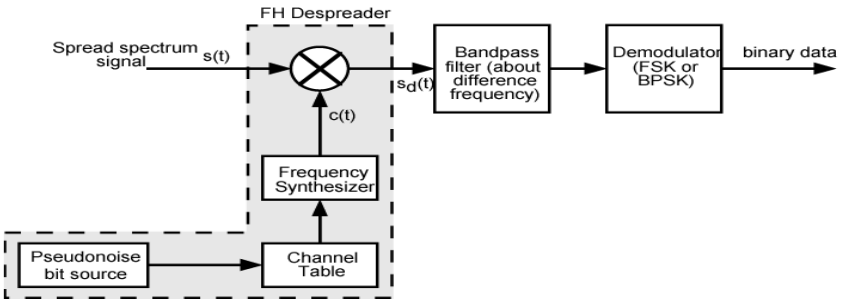
<table>
<tr><td></td><td>


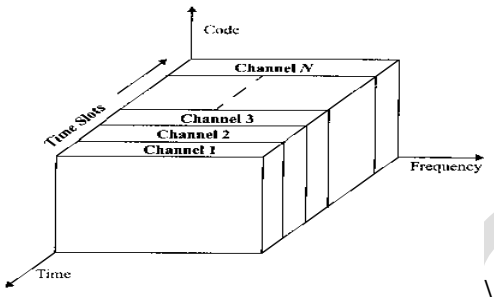
Fig: Frequency Hopping Spread Spectrum System (Receiver) **(4M)**
Processing gain : $\frac{bandwidth\ of\ spreaded\ signal}{bandwidth\ of\ unspreaded\ signal}$ **(2M)**
</td></tr>
<tr><td>5</td><td>
Explain the basic principle of TDMA. NOV/DEC 2011 (13M)BTL1
Answer: Page 17.7- K.Muralibabu
TDMA systems divide the channel time into frames. Each frame is further partitioned into time slots. In each slot only one user is allowed to either transmit or receive. (2M)


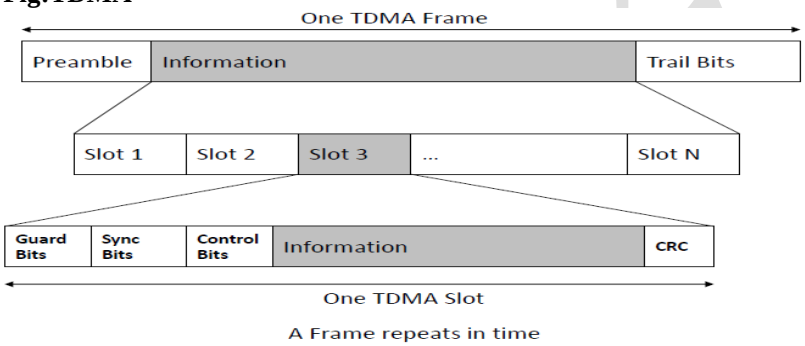
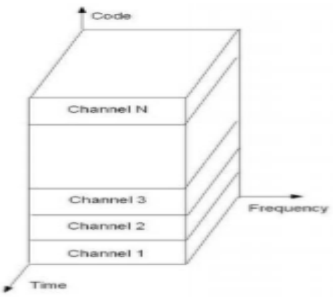
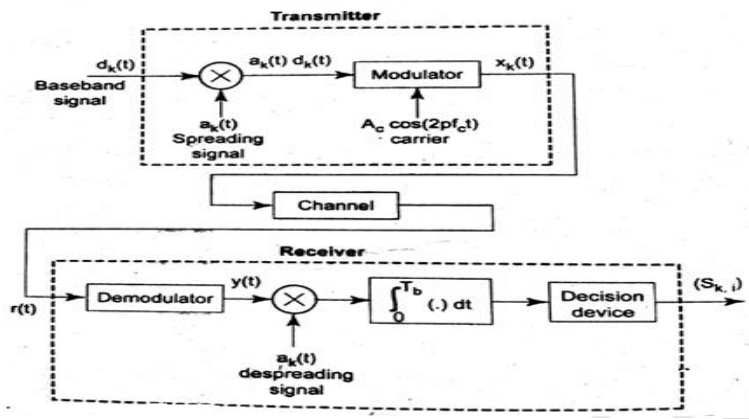
Fig:TDMA **(4M)**


Fig:Frame of TDMA **(4M)**
Advantage and disadvantage **(3M)**
Advantages- it shares the carrier frequency ,handoff process is much simpler for a subscriber unit .power efficiency of TDMA is better than FDMA.
Disadvantages—High synchronization, guard band should be minimized for higher frequency
</td></tr>
<tr><td>6</td><td>
Describe the application of CDMA in wireless communication systemList the advantages of TDMA over CDMA.APRIL/MAY2010 , NOV/DEC2010, APRIL?MAY 2011(13M) BTL1
Answer: Page 17.11- K.Muralibabu
In CDMA, the narrowband message signal is *multiplied* by a very large bandwidth signal called spreading signal (code) before modulation and transmission over the air. This is called spreading. (2M)
Diagram (3M)
</td></tr>
</table>



(5M)



ADVANTAGES: **(3M)**

Low power spectral density.
– Signal is spread over a larger frequency band
– Other systems suffer less from the transmitter
• Interference limited operation
– All frequency spectrum is used
• Privacy
– The codeword is known only between the sender and receiver. Hence other users can not decode the messages that are in transit
• Reduction of multipath affects by using a larger Spectrum

7 **Compare different types of multiple access technique. (13M)**BTL1

Answer : Page : 17.14- K.MURALIBABU

Approach	SDMA	TDMA	FDMA	CDMA
Idea	segment space into cells/sectors	segment sending time into disjoint time-slots, demand driven or fixed patterns	segment the frequency band into disjoint sub-bands	spread the spectrum using orthogonal codes
Terminals	only one terminal can be active in one cell/one sector	all terminals are active for short periods of time on the same frequency	every terminal has its own frequency, uninterrupted	all terminals can be active at the same place at the same moment, uninterrupted
Signal separation	cell structure, directed antennas	synchronization in the time domain	filtering in the frequency domain	code plus special receivers
Advantages	very simple, increases capacity per km²	established, fully digital, flexible	simple, established, robust	flexible, less frequency planning needed, soft handover
Dis-advantages	inflexible, antennas typically fixed	guard space needed (multipath propagation), synchronization difficult	inflexible, frequencies are a scarce resource	complex receivers, needs more complicated power control for senders
Comment	only in combination with TDMA, FDMA or CDMA useful	standard in fixed networks, together with FDMA/SDMA used in many mobile networks	typically combined with TDMA (frequency hopping patterns) and SDMA (frequency reuse)	still faces some problems, higher complexity, lowered expectations; will be integrated with TDMA/FDMA

(8M)

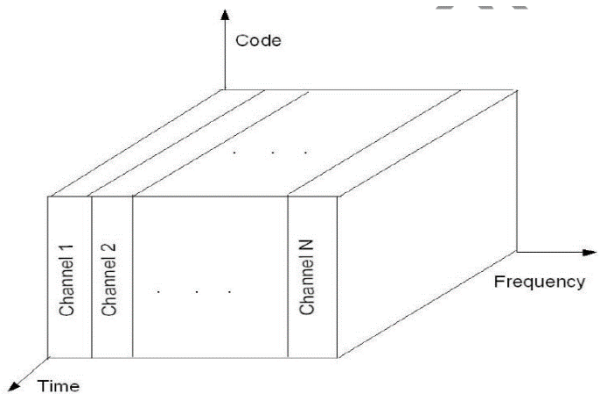
8

Describe the frequency division multiple access techniques. (13M) **(April 2018) (April 2016) (April 2017) (Nov 2017)**BTL2

Draw a typical TDMA system. Explain the operation with the time pattern. (13M) **(Nov 2014)**

Answer: Page: 2.6-K.Muralibabu

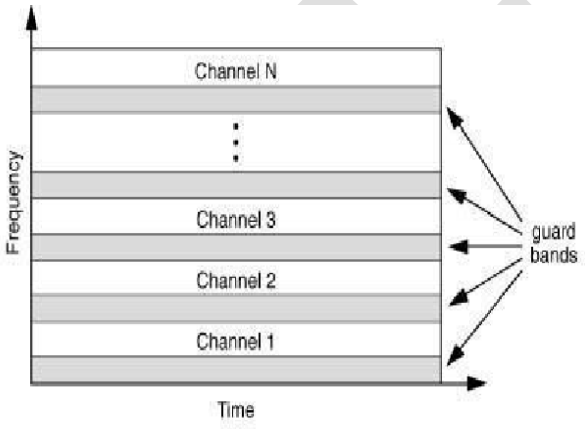
Definition : simplest multiple access- voice – data transmission -total bandwidth – divided- non overlapping frequency sub bands. (2M)



Guard bands: Guard band- adjacent frequency bands (2M)

$$N = \frac{B_t - B_{guard}}{B_c}$$

Diagram : (2M)



Non -linear effect in FDMA : many channels -same antenna – Base station -power amplifier – intermodulation frequencies- intermodulation distortion-minimize- stringent RF filter -reject intermodulation distortion -RF filter heavy- costly . (4M)

Advantages : simple- narrow band system – complexity-low compare -TDMA- absence synchronization- provide interference. (2M)

Dis advantages : not suitable -multimedia communication- wasted resources- intermodulation distortion – stringent RF-lack of flexibility –configuration. (1M)

PART*C

1

Explain the spread spectrum techniques and its types (15M) BTL2

Answer : Page : 1.6- K.Muralibabu (refer part b 4,5)

Definition : spreading in transmitter and dispreading in receiver side (2M)

Model DS-BPSK transmitter and receiver diagram (3M)

Transmitter and receiver output waveform

Explanation: narrow band signal is converted into wideband with help of PN sequence and BPSK modulation technique (3M)

Definition-one carrier frequency hopped one to another (1M)

Types : slow and fast

Block diagram (FH/BFSK) (3M)

Explanation : the process of randomly hopping the modulated data carrier form one frequency to other , the data is used to modulate a carrier. (3M)

2

Explain the Different types of multiple access technique. (15M) BTL1

Discuss in detail the multiple access techniques that are used in wireless communications. What differences is taken into account here as the channel is now wireless?

Answer : Page : 2.5- K.Muralibabu

Approach	SDMA	TDMA	FDMA	CDMA
Idea	segment space into cells/sectors	segment sending time into disjoint time-slots, demand driven or fixed patterns	segment the frequency band into disjoint sub-bands	spread the spectrum using orthogonal codes
Terminals	only one terminal can be active in one cell/one sector	all terminals are active for short periods of time on the same frequency	every terminal has its own frequency, uninterrupted	all terminals can be active at the same place at the same moment, uninterrupted
Signal separation	cell structure, directed antennas	synchronization in the time domain	filtering in the frequency domain	code plus special receivers
Advantages	very simple, increases capacity per km²	established, fully digital, flexible	simple, established, robust	flexible, less frequency planning needed, soft handover
Dis-advantages	inflexible, antennas typically fixed	guard space needed (multipath propagation), synchronization difficult	inflexible, frequencies are a scarce resource	complex receivers, needs more complicated power control for senders
Comment	only in combination with TDMA, FDMA or CDMA useful	standard in fixed networks, together with FDMA/SDMA used in many mobile networks	typically combined with TDMA (frequency hopping patterns) and SDMA (frequency reuse)	still faces some problems, higher complexity, lowered expectations; will be integrated with TDMA/FDMA

3

Summarise about the multiple access scheme.(Apr/May 2010) (15M)BTL2

Answer: Page 17.2- K.Muralibabu

Three major techniques: (2M)

- Frequency division multiple access (FDMA)
- Time division multiple access (TDMA)
- Code division multiple access (CDMA)

Others:

- Packet radio (PR)
- Space division multiple access (SDMA)

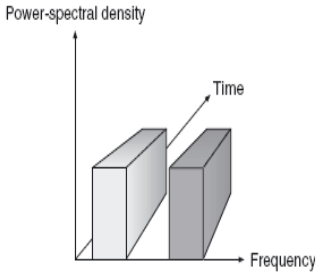
(5M)

TDMA

Single carrier frequency - several users
Not continuous - bursts.
Handoff process - simpler
Duplexers - not required.
Transmission rates - High.
Synchronization overhead - high

FDMA **(4M)**

Channel - only one phone circuit
If Channel not in use - other users can't use
Continuous transmission scheme
Narrowband systems.
Intersymbol interference - low.
Mobile unit - duplexers.
Requires RF filter - adjacent channel interference



CDMA **(4M)**

Share the same frequency.
Soft capacity limit.
Frequency - dependent transmission impairments.
Multipath fading – reduced.
Channel data rates – high.
Macroscopic spatial diversity - soft handoff.