NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

COURSE STRUCTURE & SYLLABUS
M.Tech for
DIGITAL SYSTEMS & COMPUTER ELECTRONICS PROGRAMME

Academic Regulations (R25) for M.Tech. (Regular)

(Effective for the students admitted into I year from the Academic Year 2025-2026 onwards)

ACADEMIC REGULATIONS - R25 FOR M.Tech (REGULAR) DEGREE COURSE

Applicable for the students admitted to M.Tech (Regular) Course from the Academic Year 2025-26 and onwards. The M.Tech Degree of Narasaraopeta Engineering College(Autonomous) affiliated to Jawaharlal Nehru Technological University Kakinada shall be conferred on candidates who are admitted to the program and who fulfil all the requirements for the award of the Degree.

1.0 ELIGIBILITY FOR ADMISSIONS

Admission to the above program shall be made subject to eligibility, qualification and specialization as prescribed by the University from time to time.

Admissions shall be made on the basis of merit/rank obtained by the candidates (i) in national level qualifying Entrance Test (GATE), (ii) AP PGECET conducted by State Government and (iii) Few Sponsored seats notified by University on the basis of any order of merit as approved by the State Government /JNTUK University, Kakinada subject to reservations as laid down by the Government from time to time.

2.0 AWARD OF M.Tech DEGREE

2.1 A student shall be declared eligible for the award of the M.Tech Degree, if he pursues a course of study in not less than two and not more than four academic years. Under any circumstances, permission shall not be given to complete the course work beyond four years.

2.2 **The student shall register for all 80 credits and secure all the 80 credits.**

2.3 The minimum instruction period in each semester is 16 weeks.

3.0 PROGRAMME OF STUDY

The following specializations are offered at present for the M.Tech Programme of study.

M.Tech in

1. CSE/CS&E
2. Digital Electronics and Communication Systems
3. Digital Systems & Computer Electronics
4. Machine Design
5. Power and Industrial Drives
6. Structural Engineering
7. Thermal Engineering
8.VLSI & Embedded Systems

4.0 ATTENDANCE

4.1 Attendance is calculated separately for each course. Attendance in all classes (Theory/Laboratories) is compulsory. The minimum required attendance in each course is 75%. A student shall not be permitted to appear for the Semester End Examinations (SEE), if his/her attendance is less than 75%.

4.2 Condoning of shortage of attendance (between 65% and 75%) up to a maximum of 10% (*considering the days of attendance in sports, games, NSS activities and medical exigencies*) in each course (Theory/Lab/Seminar) is condoned on production of valid Certificates/documents in the stipulated time mentioned here

with:

4.2.1 Students who are admitted as in patients for treatment are only eligible to claim condonation of attendance. Such students under medical exigencies need to Produce (a) Doctor Medical Prescription, (ii) Medical bills duly signed by Doctor/Hospital authorities, (c) Diagnosis reports, if any, (d) Discharge summary issued at the time of discharge and any other supporting documents within two week(s) from the date of discharge.

Note: University at any point of time can inform the college to submit such list/proofs. *Hence, respective HOD shall verify and accord condonation privilege scrupulously*.

4.2.2 Students participation in Sports/Games and NSS activities shall also be permitted for condonation of attendance. In such cases, they need to produce (a) invitation letter from the organizing institute/agency, (ii) participation certificate and any supporting documents within two week(s) from the date of participation to the respective HOD.

4.3 A prescribed fee per course shall be payable for condoning shortage of attendance after getting the approval of College Academic Committee for the same. The College Academic Committee shall maintain all the relevant documents along with the request from the students, whose attendance is condoned.

4.4 Shortage of Attendance below 65% in any course shall in no case be condoned.

4.5 A Student, whose shortage of attendance is not condoned in any course(s) (Theory/Lab/Seminar) in any Semester, is considered as '**Detained in that course(s)**, and is not eligible to write Semester End Examination(s) of such Course(s), (in case of Seminar, his/her Seminar Report or Presentation are not eligible for evaluation) in that Semester; and he/she has to seek re-registration for those course(s) in subsequent Semesters, and attend the same as and when offered.

4.6 A student shall put in a minimum required attendance in at least FOUR courses in I semester for promotion to II Semester; and at least FOUR courses in II semester for promotion to III Semester.

Re-admission / re-registration

4.7 A student shall not be permitted to appear for the Semester End Examinations (SEE) in a course unless they meet the prescribed attendance requirements for that course. Such students may take readmission for the course in the subsequent semester when it is offered by paying the prescribed fee, *at least 30 days before the commencement of classwork*. The HOD concerned must obtain permission from the Principal by submitting the list of students eligible/applied for readmission before the commencement of classwork.

4.8 Students who fail due to **less internal marks (less than 50%)** may register for the course within the maximum permissible duration of the Program.

4.9 In such a case, the candidate must re-register for the course(s) and secure the required minimum attendance. The candidate's attendance in the re-registered course(s) shall be calculated separately to decide upon eligibility for writing the end examination in those course(s).

4.10 In a semester, students are permitted to re-register maximum of THREE courses.

4.11 Upon re-registration, the student's previous performance in the respective course(s) will be nullified. Re-registration must be completed by paying the prescribed fee at least 30 days prior to the commencement of classwork.

5.0 EVALUATION

The performance of the candidate in each semester shall be evaluated course-wise, with a maximum of 100 marks for theory and 100 marks for practical, on the basis of Internal Evaluation and End Semester Examination.

5.1 For the theory courses 60 marks shall be awarded based on the performance in the End Semester Examination and 40 marks shall be awarded based on the Internal Evaluation. The continuous / internal evaluation shall be made based on the **average** of the marks secured in the two CIE/Mid Term-Examinations conducted-one in the middle of the Semester and the other immediately after the completion of instruction. Each CIE/midterm examination shall be conducted for a total duration of 120 minutes with 4 questions (without choice) each question for 10 marks. End semester examination is conducted for 60 marks for all FIVE (5) questions (one question from one unit) to be answered (either or).

5.2 For practical courses, 60 marks shall be awarded based on the performance in the End Semester Examinations and 40 marks shall be awarded based on the day-to-day performance as Internal Marks. The internal evaluation based on the day to day work-10 marks, record- 10 marks and the remaining 20 marks to be awarded by conducting an internal laboratory test. The end examination shall be conducted by the examiners, with breakup marks of Procedure-**15,** Experimentation- **25,** Results-10, Viva-voce-10.

5.3 For Seminar, a student under the supervision of a faculty member, shall collect the literature on a topic and critically review the literature and submit it to thedepartment in a report form and shall make an oral presentation before the Project Review Committee consisting of Head of the Department, supervisor/mentor and two other senior faculty members of the department. For Seminar, there will be only internal evaluation of 100 marks. A candidate has to secure a minimum of 50% of marks to be declared successful.

5.4 A candidate shall be deemed to have secured the minimum academic requirement in a course if he secures a minimum of 40% of marks in the End semester Examination and a minimum aggregate of 50% of the total marks in the End Semester Examination and Internal Evaluation taken together.

5.5 Laboratory examination for M.Tech. courses must be conducted with two Examiners, one of them being the Laboratory Class Teacher and the second examiner shall be drawn from the other autonomous colleges.

5.6 Students shall undergo mandatory summer internship / industrial training (credit Course) for a minimum of **eight weeks duration** at the end of second semester of the Programme/Summer Break. A student will be required to submit a summer internship/industrial training report to the concerned department and appear for an oral presentation before the committee. The Committee comprises of a HoD / Professor of the department and two faculty. The report and the oral presentation shall carry 40% and 60% weightages respectively. For summer internship / industrial training, there will be only internal evaluation of 100 marks. A candidate has to secure a minimum of 50% of marks to be declared successful.

5.7 The objective of comprehensive viva-voce is to assess the overall knowledge of the student in the relevant field of Engineering/Specialization in the PG program. Viva will be conducted in 3rd semester. The examination committee will be constituted by the HoD and consist of Professor of the department and two faculty. For comprehensive viva-voce, there will be only internal evaluation of 100 marks. A candidate has to secure a minimum of 50% of marks to be declared successful.

6.0 EVALUATION OF SEMINAR/INTERNSHIP/DISSERTATION WORK

All the students admitted under these regulations have to mandatorily comply the requirements of (i) Seminar-I, (ii) Seminar-II, (iii) Comprehensive Viva, (iv) Dissertation Part-A and (v) Dissertation Part-B. Out of these, (i) to (iv) are evaluated by internally by Project Review Committee (PRC) and (v) External Evaluation.

6.1 A Project Review Committee (PRC) shall be constituted with Head of the Department and Two other senior faculty members in the department.

6.2 Students are required to appear for Seminar-I and Seminar-II in First and Second semester respectively. They shall present before PRC on the topic of their choice/interest preferably on the courses listed in respective semesters. PRC shall advise the students in advance to select topics which strengthen their Dissertation Part-A and Dissertation Part-B.

6.3 Registration of Dissertation/Project Work: A candidate is permitted to register for the project work after satisfying the attendance requirement of all the courses, both theory and practical and duly approved by PRC.

6.4 After satisfying 6.3, student has to submit, in consultation with his project supervisor, the title, objective and plan of action of his project work for approval

6.5 If a candidate wishes to change his/her supervisor or topic of the project, he/she can do so with the approval of PRC. However, the PRC shall examine whether or not the change of topic/supervisor leads to a major change of his initial plans of project proposal. If yes, his date of registration for the project work starts from the date of change of Supervisor or topic as the case may be.

6.6 Continuous assessment of Dissertation-Part A and Dissertation-Part B during the Semester(s) will be monitored by PRC. *Dissertation-Part A* will be only internal evaluation by PRC for 100 marks. A candidate has to secure a minimum of 50% of marks to be declared successful.

6.7 The candidate shall submit a status report to the PRC in two stages, each accompanied by an oral presentation, with a minimum interval of three months between the two.

6.8 The work on the project shall be initiated at the beginning of the III Sem and the duration of the project is two semesters. A candidate is permitted to submit Project Thesis *(Dissertation – Part A & Part B)* only with the approval of PRC not earlier than 40 weeks from the date of registration of the project work.

6.9 Three copies of the project thesis, certified by the supervisor, shall be submitted to the PRC along with the plagiarism report.

6.10 The thesis shall be adjudicated by one examiner selected by the Principal. For this, the HOD of the concerned department shall submit a panel of 3 examiners, eminent in that field, with the help of the guide concerned.

6.11 If the report of the examiner is *not favourable*, the candidate shall revise and

resubmit the Thesis, in the time frame as decided by the PRC. If the report of the examiner is *not favourable* again, the thesis shall be summarily rejected. The candidate has to reregister for the project and complete the project within the stipulated time after taking the approval from the Principal.

6.12 If the report of the examiner is favourable, Viva-Voce examination shall be conducted by a board consisting of the Supervisor, Head of the Department and the examiner who adjudicated the Thesis. The Head of the Department shall coordinate and make arrangements for the conduct of Viva-Voce examination. The Board shall jointly report the candidate's work for a maximum of 100 marks.

6.13 If the report of the Viva-Voce is unsatisfactory (i.e., < 50 marks), the candidate shall retake the Viva-Voce examination only after three months. If he fails to get a satisfactory report at the second Viva-Voce examination, the candidate has to re-register for the project and complete the project within the stipulated time after taking the approval from the Principal.

7.0 Cumulative Grade Point Average (CGPA)

Marks Range (Max – 100)	Letter Grade	Level	Grade Point
≥ 90	S	Outstanding	10
≥80 to <90	A	Excellent	9
≥70 to <80	B	Very Good	8
≥60 to <70	C	Good	7
≥50 to <60	D	Fair	6
<50	F	Fail	0
		Absent	0

Computation of SGPA

- The following procedure is to be adopted to compute the Semester Grade Point Average(SGPA) and Cumulative Grade Point Average(CGPA):
- The **SGPA** is the ratio of sum of the product of the number of credits with the grade points scored by a student in all the courses taken by a student and the sum of the number of credits of all the courses undergone by a student, i.e

  SGPA (Si) = ∑ (Ci X Gi) / ∑ Ci
- Where Ci is the number of credits of the i^{th} course and Gi is the grade point scored by the student in the i^{th} course.

Computation of CGPA

- The **CGPA** is also calculated in the same manner taking into account all the courses undergone by a student over all the semester of a Programme, i.e.

  CGPA = ∑ (Ci X Si) / ∑ Ci
- Where Si is the SGPA of the i^{th} semester and Ci is the total number of credits in that semester.
- The SGPA and CGPA shall be rounded off to 2 decimal points and reported in the transcripts.
- Equivalent Percentage = (CGPA- 0.5) x 10

8.0 AWARD OF DEGREE AND CLASS

After a student has satisfied the requirements prescribed for the completion of the program and is eligible for the award of M.Tech. Degree he/she shall be placed in one of the following classes:

Class Awarded	CGPA to be secured	
First Division with Distinction	≥ 7.5 (without supplementary History)	From the CGPA secured from 80 credits
First Class	≥ 6.5	
Second Class	≥ 6.0 to < 6.5	

The secured grade, grade points, status and credits obtained will be shown separately in the memorandum of marks.

9.0 WITHHOLDING OF RESULTS

If the student is involved in indiscipline/malpractices/court cases, the result of the student will be withheld.

10.0 GENERAL

10.1 Wherever the words "he", "him", "his", occur in the regulations, they include "she", "her", "hers".

10.2 The academic regulation should be read as a whole for the purpose of any interpretation.

10.3 In the case of any doubt or ambiguity in the interpretation of the above rules, the decision of the Principal is final.

10.4 The College may change or amend the academic regulations or syllabi at any time and the changes or amendments made shall be applicable to all the students with effect from the dates notified by the College.

MALPRACTICES RULES

DISCIPLINARY ACTION FOR / IMPROPER CONDUCT IN EXAMINATIONS

	Nature of Malpractices/Improper conduct	**Punishment**
	If the candidate:	
1. (a)	Possesses or keeps accessible in examination hall, any paper, note book, programmable calculators, Cell phones, pager, palm computers or any other form of material concerned with or related to the subject of the examination (theory or practical) in which he is appearing but has not made use of (material shall include any marks on the body of the candidate which can be used as an aid in the subject of the examination)	Expulsion from the examination hall and cancellation of the performance in that subject only.
(b)	Gives assistance or guidance or receives it from any other candidate orally or by any other body language methods or communicates through cell phones with any candidate or persons in or outside the exam hall in respect of any matter.	Expulsion from the examination hall and cancellation of the performance in that subject only of all the candidates involved. In case of an outsider, he will be handed over to the police and a case is registered against him.
2.	Has copied in the examination hall from any paper, book, programmable calculators, palm computers or any other form of material relevant to the subject of the examination (theory or practical) in which the candidate is appearing.	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted to appear for the remaining examinations of the subjects of that Semester/year. The Hall Ticket of the candidate is to be cancelled and sent to the University.

3.	Impersonates any other candidate in connection with the examination.	Both the candidates involved in the malpractice will forefeit their seat. If the imposter is an outsider, he will be handed over to the police and a case is registered against him.
4.	Smuggles in the Answer book or additional sheet or takes out or arranges to send out the question paper during the examination or answer book or additional sheet, during or after the examination.	Expulsion from the examination hall and cancellation of performance in that subject and all the other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The candidate is also debarred for two consecutive semesters from class work and all University examinations. The continuation of the course by the candidate is subject to the academic regulations in connection with forfeiture of seat.
5.	Uses objectionable, abusive or offensive language in the answer paper or in letters to the examiners or writes to the examiner requesting him to award pass marks.	Cancellation of the performance in that subject.

6.	Refuses to obey the orders of the Chief Superintendent/Assistant – Superintendent / any officer on duty or misbehaves or creates disturbance of any kind in and around the examination hall or organizes a walk out or instigates others to walk out, or threatens the officer-in charge or any person on duty in or outside the examination hall of any injury to his person or to any of his relations whether by words, either spoken or written or by signs or by visible representation, assaults the officer in-charge, or any person on duty in or outside the examination hall or any of his relations, or indulges in any other act of misconduct or mischief which result in damage to or destruction of property in the examination hall or any part of the College campus or engages in any other act which in the opinion of the officer on duty amounts to use of unfair means or misconduct or has the tendency to disrupt the orderly conduct of the examination.	In case of students of the college, they shall be expelled from examination halls and cancellation of their performance in that subject and all other subjects the candidate(s) has (have) already appeared and shall not be permitted to appear for the remaining examinations of the subjects of that semester/year. The candidates also are debarred and forfeit their seats. In case of outsiders, they will be handed over to the police and a police case is registered against them.
7.	Leaves the exam hall taking away answer script or intentionally tears of the script or any part thereof inside or outside the examination hall.	Expulsion from the examination hall and cancellation of performance in that subject and all the other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The candidate is also debarred for two consecutive semesters from class work and all University examinations. The continuation of the course by the candidate is subject to the academic regulations in connection with forfeiture of seat

8.	Possess any lethal weapon or firearm in the examination hall.	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The candidate is also debarred and forfeits the seat.
9.	If student of the college, who is not a candidate for the particular examination or any person not connected with the college indulges in any malpractice or improper conduct mentioned in clause 6 to 8.	Student of the colleges expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The candidate is also debarred and forfeits the seat. Person(s) who do not belong to the College will be handed over to police and, a police case will be registered against them.
10.	Comes in a drunken condition to the examination hall.	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the candidate has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year.
11.	Copying detected on the basis of internal evidence, such as, during valuation or during special scrutiny.	Cancellation of the performance in that subject and all other subjects the candidate has appeared including practical examinations and project work of that semester/year examinations.
12.	If any malpractice is detected which is not covered in the above clauses 1 to 11 shall be reported to the University for further action to award suitable punishment.	

Malpractices identified by squad or invigilators: Punishments to the candidates as per the above guidelines.

Programme Structure

R25 M.Tech (DS&CE) Structure

M.Tech (DS&CE) I – Semester

S. No.	Course Code	Course Title	L	T	P	C
1	PC	Mathematical Foundation for Communication Engineering	3	1	0	4
2	PC	Digital System Design	3	1	0	4
3	PC	Embedded System Design	3	1	0	4
4	PE-I	Program Elective-I	3	0	0	3
5	PE-II	Program Elective-II	3	0	0	3
6		Digital System Design Lab	0	1	2	2
7		Embedded System Design Lab	0	1	2	1
8		Seminar-1	0	0	2	1
		Total	**15**	**5**	**6**	**23**

List of Professional Elective Courses in I Semester (Electives – I & II)

S.No.	Course Code	Course Title
1	PE -I	MEMS
2	PE -I	Network Security and Cryptography
3	PE -I	Data Acquisition Systems
4	PE -II	FPGA and ASIC Design
5	PE -II	Big Data Analytics
6	PE -II	System Modeling and Simulation

@ Minimum 2/3 themes per elective

MTech (DS&CE) II – Semester

Sl. No.	Course Code	Course Title	L	T	P	C
1	PC	System Design with RTOS & Embedded LINUX	3	1	0	4
2	PC	IoT Architecture & Applications	3	1	0	4
3	PC	Computer Networks	3	1	0	4
4	PE-III	Program Elective-III	3	0	0	3
5	PE-IV	Program Elective-IV	3	0	0	3
6		Internet of Things (IoT) Laboratory	0	1	2	2
7		System Design with RTOS & Embedded LINUX Lab	0	1	2	2
8		Seminar-II	0	0	2	1
		Total	**15**	**5**	**6**	**23**

List of Professional Elective Courses in II Semester (Electives III & IV)

S.No.	Course Code	Course Title
1	PE -III	System On Chip Design
2	PE -III	Advanced Computer Architecture and Organization
3	PE -III	Cyber Security
4	PE -IV	Quantum Science and Technology
5	PE -IV	ARM Controllers and Embedded C
6	PE -IV	Robotics

@ Minimum 2/3 themes per elective

MTech (DS&CE) - III Semester

Sl. No.	Course Title	L	T	P	C
1	Research Methodology and IPR/Swayam 12 Week MOOC course- RM & IPR	3	0	0	3
2	Summer Internship/Industrial training (8- 10) Weeks*	-	-	-	3
3	Comprehensive Viva#	-	-	-	2
4	Dissertation part -A$	-	-	20	10
	Total	**3**	**-**	**20**	**18**

* Student attended during summer / year break and assessment will be done in 3rd Sem.
# Comprehensive viva can be conducted courses completed upto second sem.
$ Dissertation – Part A, internal assessment

M.Tech (DS&CE) – IV Semester

Sl. No.	Course Title	L	T	P	C
1	Dissertation Part – B%	-	-	32	16
	Total	**-**	**-**	**32**	**16**

% External Assessment

Revised Bloom's Taxonomy Action Verbs

Definitions	I. Remembering	II. Understanding	III. Applying	IV. Analyzing	V. Evaluating	VI. Creating
Bloom's Definition	Exhibit memory of previously learned material by recalling facts, terms, basic concepts, and answers.	Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions, and stating main ideas.	Solve problems to new situations by applying acquired knowledge, facts, techniques and rules in a different way.	Examine and break information into parts by identifying motives or causes. Make inferences and find evidence to support generalizations.	Present and defend opinions by making judgments about information, validity of ideas, or quality of work based on a set of criteria.	Compile information together in a different way by combining elements in a new pattern or proposing alternative solutions.
Verbs	• Choose • Define • Find • How • Label • List • Match • Name • Omit • Recall • Relate • Select • Show • Spell • Tell • What • When • Where • Which • Who • Why	• Classify • Compare • Contrast • Demonstrate • Explain • Extend • Illustrate • Infer • Interpret • Outline • Relate • Rephrase • Show • Summarize • Translate	• Apply • Build • Choose • Construct • Develop • Experiment with • Identify • Interview • Make use of • Model • Organize • Plan • Select • Solve • Utilize	• Analyze • Assume • Categorize • Classify • Compare • Conclusion • Contrast • Discover • Dissect • Distinguish • Divide • Examine • Function • Inference • Inspect • List • Motive • Relationships • Simplify • Survey • Take part in • Test for • Theme	• Agree • Appraise • Assess • Award • Choose • Compare • Conclude • Criteria • Criticize • Decide • Deduct • Defend • Determine • Disprove • Estimate • Evaluate • Explain • Importance • Influence • Interpret • Judge • Justify • Mark • Measure • Opinion • Perceive • Prioritize • Prove • Rate • Recommend • Rule on • Select • Support • Value	• Adapt • Build • Change • Choose • Combine • Compile • Compose • Construct • Create • Delete • Design • Develop • Discuss • Elaborate • Estimate • Formulate • Happen • Imagine • Improve • Invent • Make up • Maximize • Minimize • Modify • Original • Originate • Plan • Predict • Propose • Solution • Solve • Suppose • Test • Theory

I Semester	MATHEMATICAL FOUNDATION FOR COMMUNICATION ENGINEERING	L	T	P	C
		3	1	0	4

COURSE OBJECTIVES:

1. To develop a strong foundation in probability and statistics, including parameter estimation, hypothesis testing, and regression analysis.
2. To learn stochastic processes with emphasis on random walks and Markov chains relevant to communication systems.
3. To apply numerical methods for solving equations, differential equations, and matrix eigenvalue problems in engineering contexts.
4. To explore optimization techniques for multivariable functions, including constrained optimization using Lagrange multipliers and gradient-based methods.
5. To introduce wavelet transforms and multi-resolution analysis for signal processing applications in communication engineering.

Unit – I: Probability and Statistics:

Sampling distributions, Estimation of parameters (point estimation – unbiasedness & minimum variance, basics of interval estimation – confidence interval for mean), Testing of hypotheses (one and two sample tests for mean), Linear regression, Introduction to non-linear regression.

Unit II: Stochastic process:

Random processes, Random walk, Markov process with special emphasis on Markov chain

Unit – III: Numerical Analysis:

Introduction to Interpolation formulae [Bessel's & Sterling's], Roots of transcendental equations [Bisection, Regula-Falsi & Newton-Raphson] Solutions of simultaneous non-linear equations [Newton's method], Numerical solution of Ordinary Differential equation [Modified Euler's method, fourth order Runga-Kutta method], Matrix Eigen value and Eigen vector problems.

Unit IV: Optimization Technique:

Calculus of several variables, Implicit function theorem, Nature of singular points, Necessary and sufficient conditions for optimization, Constrained Optimization, Lagrange multipliers, Gradient method – steepest descent method.

Unit V: Wavelet Transform:

Resolution problems, Multi-resolution analysis, Continuous & discrete wavelet transform.

TEXT BOOKS:

1. A. Papoulis and S. Unnikrishnan Pillai, ``Probability, Random Variables and StochasticProcesses," Fourth Edition, McGraw Hill. (Indian Edition is available).
2. Gibert Strang," Linear Algebra and its applications", Thomson Learning Inc, 4th Edition.

REFERENCE BOOKS:

1. 1.H.Stark and J. Woods, 'Probability and Random Processes with Applications to Signal Processing," Third Edition, Pearson Education. (Indian Edition is available).
2. Steven M. Kay, " Intuitive Probability and Random Process using Matlab", Springer Publications.
3. 3.Todd K Moon, Wynn C. Stirling" Mathematical Methods and Algorithms for Signal Processing, Prentice Hall.

COURSE OUTCOMES:

After completion of the course students will be able to:

1. Analyze stochastic processes and model systems using Markov chains and random walks.
2. Solve mathematical problems using numerical techniques for interpolation, equation roots, and differential equations.
3. Employ optimization techniques including calculus-based methods and constrained optimization strategies.
4. Familiarize and utilize wavelet transforms for multi-resolution analysis and signal processing applications.

I Semester	DIGITAL SYSTEM DESIGN	L	T	P	C
		3	1	0	4

COURSE OBJECTIVES:

1. To introduce Verilog HDL and its role in digital system design, covering syntax, simulation, synthesis, and design hierarchy.
2. To develop gate-level modelling skills using Verilog primitives, including flip-flop design, delay modelling, and net types.
3. To explore dataflow-level modelling techniques with continuous assignments, vector operations, and parameterized designs.
4. To learn behavioural modelling constructs such as initial blocks, always blocks, and delay-based assignments for functional simulation.
5. To master procedural and control constructs in Verilog including loops, conditional statements, and event-driven simulation for complex digital systems.

UNIT - I:

Introduction to VerilLog HDL: Verilog as HDL, Levels of Design Description, Concurrency, Simulation and Synthesis, Function Verification, System Tasks, Programming Language Interface, Module, Simulation and Synthesis Tools Language Constructs and Conventions: Introduction, Keywords, Identifiers, White Space, Characters, Comments, Numbers, Strings, Logic Values, Strengths, Data Types, Scalars and Vectors, Parameters, Operators.

UNIT - II:

Gate Level Modelling: Introduction, AND Gate Primitive, Module Structure, Other Gate Primitives, Illustrative Examples, Tristate Gates, Array of Instances of Primitives, Design of Flip-Flops with Gate Primitives, Delay, Strengths and Construction Resolution, Net Types, Design of Basic Circuit.

UNIT -III:

Modelling at Dataflow Level: Introduction, Continuous Assignment Structure, Delays and Continuous Assignments, Assignment to Vector, Operators, Design at dataflow level, Parameter and constant usage in dataflow.

UNIT - IV:

Behavioural Modelling: Introduction, Operations and Assignments, Functional Bifurcation, 'Initial' Construct, Assignments with Delays, 'Wait Construct, Multiple Always Block, Designs at Behavioural Level

UNIT -V:

Verilog Procedural and Control Constructs: Blocking and Non-Blocking Assignments, The 'Case' Statement, Simulation Flow, 'If an 'if-Else' Constructs, 'Assign- De-Assign' Constructs, 'Repeat' Construct, for loop, 'The Disable' Construct, 'While Loop', Forever Loop, Parallel Blocks, Force-Release, Construct, Event.

TEXT BOOKS:

1. T.R. Padmanabhan, B Bala Tripura Sundari, Design Through Verilog HDL, Wiley 2009.
2. Zainalabdien Navabi, Verliog Digital System Design, TMH, 2nd Edition.

REFERENCE BOOKS:

1. Fundamentals of Digital Logic with Verilog Design - Stephen Brown, Zvonkoc Vranesic, TMH, 2nd Edition.
2. Advanced Digital Logic Design using Verilog, State Machines & Synthesis for FPGA Sunggu Lee, Cengage Learning, 2012.
3. Verilog HDL - Samir Palnitkar, 2nd Edition, Pearson Education, 2009.
4. Advanced Digital Design with Verilog HDL -Michel D. Ciletti, PHI,2009.

COURSE OUTCOMES:

After completion of the course students will be able to:

1. Develop digital circuits using gate-level modeling techniques.
2. Design and simulate systems at the dataflow level using continuous assignments.
3. Model system behavior using constructs like initial and always blocks.
4. Apply procedural and control constructs to implement and verify digital designs.

I Semester	EMBEDDED SYSTEM DESIGN	L	T	P	C
		3	1	0	4

COURSE OBJECTIVES:

1. To learn the fundamentals of embedded systems, including their classification, applications, characteristics, and differences from general-purpose computing systems.
2. To explore the architecture of typical embedded systems, covering processors, memory types, sensors, actuators, and communication interfaces.
3. To learn embedded firmware components and design approaches, including reset circuits, watchdog timers, and development tools.
4. To gain in-depth knowledge of ARM architecture, instruction sets, operating modes, and interrupt handling for embedded applications.
5. To develop practical skills in programming Raspberry Pi using Python and interfacing sensors and actuators through standard communication protocols.

UNIT I

Introduction to Embedded Systems

Definition of Embedded System, Embedded Systems Vs General Computing Systems, History of Embedded Systems, Classification, Major Application Areas, Purpose of Embedded Systems, Characteristics and Quality Attributes of Embedded Systems.

UNIT II

Typical Embedded System

Core of the Embedded System: General Purpose and Domain Specific Processors, ASICs, PLDs, Commercial Off-The-Shelf Components (COTS), Memory: ROM, RAM, Memory according to the type of Interface, Memory Shadowing, Memory selection for Embedded Systems, Sensors and Actuators, Communication Interface: Onboard and External Communication Interfaces.

UNIT III

Embedded Firmware

Reset Circuit, Brown-out Protection Circuit, Oscillator Unit, Real Time Clock, Watchdog Timer, Embedded Firmware Design Approaches and Development Languages.

UNIT -IV

ARM:

ARM design philosophy, data flow model and core architecture, registers, program status register, instruction pipeline, interrupts and vector table, operating modes and ARM processor families.

Instruction Sets: Data processing instructions, addressing modes, branch, load, store instructions, PSR instructions, and conditional instructions.

UNIT -V

Raspberry Pi: Raspberry Pi board and its processor, Programming the Raspberry Pi using Python, Communication facilities on Raspberry Pi (I2C, SPI, UART), Interfacing of sensors and actuators.

TEXT BOOKS:

1. Introduction to Embedded Systems - Shibu K.V, Mc Graw Hill.
2. A. N. Sloss, D. Symes, and C. Wright, "ARM System Developer's Guide: Designing andOptimizing System Software", Elsevier, 2008.
3. S. Monk, “Programming the Raspberry Pi” McGraw-Hill Education, 2013.

REFERENCE BOOKS:

1. Steave Furber, “ARM system-on-chip architecture”, Addison Wesley, 2000.
2. Embedded Systems - Raj Kamal, TMH.
3. Embedded System Design - Frank Vahid, Tony Givargis, John Wiley.

COURSE OUTCOMES:

After completion of the course students will be able to:

1. Identify and evaluate core components and communication interfaces in embedded systems.
2. Design embedded firmware with appropriate hardware support circuits.
3. Apply ARM architecture and instruction sets in embedded programming tasks.
4. Utilize Raspberry Pi for embedded applications using Python and peripheral interfacing.

I Semester	MEMS	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To introduce the fundamentals of MEMS and microsystems, including smart materials, microfabrication techniques, and their diverse applications.
2. To explore various micro sensors, actuators, and integrated systems, with emphasis on real-world devices like accelerometers, pressure sensors, and micro-mirrors.
3. To learn micro-fabrication processes such as thin-film deposition, lithography, etching, and specialized materials used in MEMS technology.
4. To learn mechanical modelling of micro scale structures using beam theory, energy methods, and stress analysis for accurate system design.
5. To apply the Finite Element Method (FEM) to analyze MEMS structures, particularly those involving piezoelectric sensors and actuators.

UNIT I

INTRODUCTION TO MEMS: Microsystems versus MEMS, Micro fabrication, Smart Materials, Structures and Systems, Integrated Microsystems, Applications of Smart Materials and Microsystems

UNIT II

MICRO SENSORS, ACTUATORS, SYSTEMS AND SMART MATERIALS: Silicon Capacitive Accelerometer, Piezo-resistive Pressure Sensor, Conductometric Gas Sensor, An Electrostatic Comb-Drive, A Magnetic Micro relay, Portable Blood Analyzer, Piezoelectric Inkjet Print Head, Micro-mirror Array for Video ProjectionSmart Materials and Systems.

UNIT III

MICRO FABRICATION TECHNIQUE: Silicon as a Material for Micromachining, Thin-Film Deposition, Lithography, Etching, Silicon Micromachining Specialized Materials for Microsystems, Advanced Processes for Micro fabrication

UNIT IV

MODELING OF SOLIDS IN MICROSYSTEMS: The Simplest Deformable Element: A Bar, Transversely Deformable Element: A beam, Energy Methods for Elastic Bodies, Heterogeneous Layered Beams, Bimorph Effect, Residual Stresses and Stress Gradients, Poisson Effect and the Anticlastic Curvature of Beams, Torsion of Beams and Shear Stresses, Dealing with Large Displacements, In-Plane Stresses.

UNIT V

FINITE ELEMENT METHOD: Need for Numerical Methods for Solution of Equations - Variational Principles, Finite Element Method, Finite Element Model for Structures with Piezoelectric Sensors and Actuators, Analysis of a Piezoelectric Bimorph Cantilever Beam.

TEXT BOOKS:

1. Fundamentals of Microfabrication — *Marc J. Madou* (CRC Press, 1997/2002).
2. An Introduction to Microelectromechanical Systems Engineering — N. Maluf(Artech House, 1999).

REFERENCE BOOKS:

1. Micro and Smart Systems by G.K. Ananthasuresh, K.J. Vinoy, S.Gopalakrishnan, K.N.Bhat,V.K.Aatre : Wiley, India (2016).
2. Smart Material Systems and MEMS: Design and Development Methodologies: Vijay K., 2017.
3. The MEMS Handbook: Edited by Mohamed Gad-el-Hak, University of NotreDame, CRCPress LLC, 2015

COURSE OUTCOMES:

After completion of the course students will be able to:

1. Analyze the design and operation of micro sensors, actuators, and integrated microsystems.
2. Evaluate various microfabrication techniques and specialized materials used in MEMS production.
3. Apply solid mechanics principles to model deformable structures in microsystem design.
4. Utilize finite element methods to analyze MEMS structures incorporating smart sensing and actuation.

I Semester	NETWORK SECURITY AND CRYPTOGRAPHY	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To learn the fundamental concepts of network security including types of attacks, security services, and classical and modern encryption techniques.
2. To explore advanced symmetric encryption algorithms such as Triple DES, Blowfish, RC5, and key distribution methods for secure communication.
3. To learn public key cryptographic principles and algorithms including RSA, Diffie-Hellman, and Elliptic Curve Cryptography, supported by essential number theory.
4. To examine message authentication techniques, hash functions, digital signatures, and authentication protocols including Kerberos and X.509.
5. To analyze security mechanisms for IP and web communication, study malware threats, and understand firewall design and trusted systems

UNIT I:

Introduction: Attacks, Services and Mechanisms, Security attacks, Security services, A Model for Internetwork security. Classical Techniques: Conventional Encryption model, Steganography, Classical Encryption Techniques.

Modern Techniques: Simplified DES, Block Cipher Principles, Data Encryption standard, Strength of DES, Differential and Linear Cryptanalysis, Block Cipher Design Principles and Modes of operations.

UNIT II:

Triple DES, International Data Encryption algorithm, Blowfish, RC5, CAST-128, RC2, Characteristics of Advanced Symmetric block cifers.

Conventional Encryption: Placement of Encryption function, Traffic confidentiality, Key distribution, Random Number Generation.

UNIT III:

Public Key Cryptography: Principles, RSA Algorithm, Key Management, Diffie-Hellman Key exchange, Elliptic Curve Cryptography.

Number Theory: Prime and relatively prime numbers, Modular arithmetic, Fermat's and Euler's theorems, Testing for primality, Euclid's Algorithm, the Chinese remainder theorem, Discrete logarithms.

UNIT IV:

Message Authentication and Hash Functions: Authentication requirements and functions, Message Authentication, Hash functions, Security of Hash functions and MACs. Hash and Mac Algorithms. MD File, Message digest Algorithm, Secure Hash Algorithm, RIPEMD-160, HMAC. Digital signatures and Authentication protocols: Digital signatures, Authentication Protocols, Digital signature standards.

Authentication Applications: Kerberos, X.509 directory Authentication service. Electronic Mail, Security: Pretty Good Privacy, S/MIME.

UNIT V:

IP Security: Overview, Architecture, Authentication, Encapsulating Security Payload, Combining security Associations, Key Management. Web Security: Web Security requirements, Secure sockets layer and Transport layer security, Secure Electronic Transaction.Intruders, Viruses and WormsIntruders, Viruses and Related threats. Fire Walls: Fire wall Design Principles, Trusted systems.

TEXT BOOKS:

1. Cryptography and Network Security: Principles and Practice - William Stallings,Pearson Education.
2. Network Security Essentials (Applications and Standards) by William StallingsPearson Education.

REFERENCE BOOKS:

1. Fundamentals of Network Security by Eric Maiwald (Dreamtech press)
2. Network Security - Private Communication in a Public World by Charlie Kaufman, Radia Perlman and Mike Speciner, Pearson/PHI.

COURSE OUTCOMES:

After completion of the course students will be able to:

1. Analyze symmetric and public key cryptographic algorithms for secure communication.
2. Apply number theory and key management principles in cryptographic systems.
3. Evaluate authentication mechanisms, hash functions, and digital signature standards.
4. Demonstrate knowledge of IP and web security protocols, firewall architecture, and threat prevention techniques.

I Semester	DATA ACQUISITION SYSTEMS	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To learn the architecture and objectives of Data Acquisition Systems (DAS), including converter characteristics and performance metrics.
2. To explore various types of Analog-to-Digital Converters (ADCs) and Non-Linear Data Converters (NDCs), along with their applications in real-world systems.
3. To learn the principles and design techniques of Digital-to-Analog Converters (DACs) and their use in programmable analog systems.
4. To study monolithic data converters and their interfacing with microprocessors for embedded applications.
5. To analyze error sources in data converters and apply noise reduction and error budgeting techniques through practical case studies.

UNIT-I

INTRODUCTION: Objective of a DAS, single channel DAS, Multi-channel DAS, Componentsused in DAS– Converter Characteristics-Resolution-Non-linearity,settling time, Monotonicity.

UNIT-II

ANALOG TO DIGITAL CONVERTERS (ADCS): Classification of A/Dconverters.Parallelfeedback – Successive approximation – Ramp comparison – Dual slopeintegration – Voltage to frequency – Voltage to Time – Logarithmic types of ADCS.

NON-LINEAR DATA CONVERTERS (NDC): Basic NDC configurations – Some commonNDACS and NADCS – Programmable non-linear ADCS – NADC using optimal sized ROMHigh speed hybrid NADC – PLS based NADC – Switched capacitor NDCS.

ADC APPLICATIONS: Data Acquisition systems – Digital signal processing systems – PCMvoice communication systems – Test and measurement instruments – Electronic weighingmachines.

UNIT-III

DIGITAL TO ANALOG CONVERTERS (DACS): Principles and design of – Parallel R–2R, Weighted resistor, inverted ladder, D/A decoding – Codes other than ordinary binary.

DATA CONVERTER APPLICATIONS: DAC applications – Digitally programmable V/Isources – Arbitrary waveform generators – Digitally programmable gain amplifiers – Analogmultipliers/ dividers – Analog delay lines.

UNIT-IV

Monolithic data converters: typical study of monolithic DACS and ADCS. Interfacing of DACS and ADCS to a μP.

UNIT-V

Error budget of DACS and ADCS: Error sources, error reduction and noise reductiontechniques in DAS. Error budget analysis of DAS, case study of a DAC and an ADC.

TEXT BOOKS:

1. Electronic data converters fundamentals and applications – Dinesh K. Anvekar, B.S. Sonde –Tata McGraw Hill.

REFERENCES:

1. Electronic Analog/ Digital conversions – Hermann Schmid – Tata McGraw Hill.
2. E.R. Hanateck, User's Handbook of D/A and A/D converters - Wiley
3. Electronic instrumentation by HS Kalsi- TMH 2 ndEdition, 2004.

COURSE OUTCOME:

After completion of the course students will be able to:

1. Analyze various analog-to-digital and non-linear data converters along with their applications.
2. Design and evaluate digital-to-analog converters and their use in signal generation and control systems.
3. Explore monolithic data converters and microprocessor interfacing techniques.
4. Assess error sources and apply correction strategies to enhance data conversion accuracy and performance.

I Semester	FPGA AND ASIC DESIGN	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To introduce the evolution, design flow, and applications of FPGAs and programmable logic devices through practical design examples.
2. To familiarize students with FPGA/CPLD programming technologies and commercially available platforms such as Xilinx, Actel, and Altera.
3. To learn the internal architecture of FPGAs/CPLDs including configurable logic blocks, routing structures, and I/O blocks, and their impact on performance.
4. To explore various FPGA routing architectures and strategies, including segmented and symmetrical routing techniques.
5. To analyze FPGA architectural components and apply knowledge through case studies involving Kintex-7, Virtex-7, and Artix-7 devices.

UNIT-I:

INTRODUCTION TO FPGAs: Evolution of programmable devices, FPGA Design flow, Applications of FPGA.

DESIGN EXAMPLES USING PLDs: Design of Universal block, Memory, Floating pointmultiplier, Barrel shifter

UNIT-II:

FPGAs/CPLDs: Programming Technologies, commercially available FPGAs, Xilinx's Vertex and Spartan, Actel's FPGA, Altera's FPGA/CPLD.

UNIT-III:

BUILDING BLOCKS OF FPGAS/CPLDS: Configurable Logic block functionality, Routing structures, Input/output Block, Impact of logic block functionality on FPGA performance, Model for measuring delay.

UNIT-IV:

ROUTING ARCHITECTURES: Routing terminology, general strategy for routing in FPGAs, routing for row – based FPGAs, introduction to segmented channel routing, routing for symmetrical FPGAs, example of routing in a symmetrical FPGA, general approach to routing in symmetrical FPGAs, independence from FPGA routing architectures, FPGA routing structures

UNIT-V:

FPGA architectural assumptions, the logic block, the connection block, connection block topology, the switch block, switch block topology, architectural assumptions for the FPGA CASE STUDY – Applications using Kintex-7, Virtex-7, Artix-7.

TEXTBOOKS:

1. John V. Old Field, Richrad C. Dorf, Field Programmable Gate Arrays, Wiley, 2008.
2. VLSI Design: A Practical Guide for FPGA and ASIC Implementations by Vikram A. Chandra setty (Springer Briefs, 2011).

REFERENCE BOOKS:

1. Data sheets of Artix-7, Kintex-7, Virtex-7
2. Stephen D. Brown, Robert J. Francis, Jonathan Rose, Zvonko G. Vranesic, Field
3. Programmable Gate Arrays, 2nd Edition, Springer, 1992.

COURSE OUTCOMES:

After completion of the course students will be able to:

1. Analyze OFDM transmission over wideband channels and evaluate system performanceunder various impairments.
2. Apply synchronization techniques to mitigate timing and frequency errors in OFDM systems.
3. Explore the information-theoretic aspects and capacity limits of MIMO communication channels.
4. Examine space-time coding techniques and multi-user detection methods for advanced MIMO-OFDM systems.

I Semester	BIG DATA ANALYTICS	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To develop proficiency in Java data structures and generics, including serialization and wrapper classes for scalable data handling.
2. To learn the architecture and components of Hadoop and configure clusters in local, pseudo-distributed, and fully distributed modes.
3. To learn to write and execute MapReduce programs using Hadoop APIs, including driver, mapper, reducer, and auxiliary components.
4. To explore Hadoop I/O mechanisms through Writable interfaces, custom comparators, and efficient data serialization techniques.
5. To gain hands-on experience with Pig scripting, its architecture, and execution modes to simplify Hadoop-based data processing.

UNIT-I

Data structures in Java: Linked List, Stacks, Queues, Sets, Maps; Generics: Generic classes and Type parameters, Implementing Generic Types, Generic Methods, Wrapper Classes, Concept ofSerialization

UNIT-II

Working with Big Data: Google File System, Hadoop Distributed File System (HDFS) – Building blocks of Hadoop (Namenode, Datanode, Secondary Namenode, JobTracker, TaskTracker), Introducing and Configuring Hadoop cluster (Local, Pseudo-distributed mode,Fully Distributed mode), Configuring XML files.

UNIT-III

Writing MapReduce Programs: A Weather Dataset, Understanding Hadoop API forMapReduce Framework (Old and New), Basic programs of HadoopMapReduce: Driver code, Mapper code, Reducer code, RecordReader, Combiner, Partitioner

UNIT-IV

Hadoop I/O: The Writable Interface, Writable Comparable and comparators, Writable Classes: Writable wrappers for Java primitives, Text, Bytes Writable, NullWritable, ObjectWritable and GenericWritable, Writable collections, implementing a Custom Writable: Implementing a Raw Comparator for speed, Custom comparators.

UNIT-V

Pig: Hadoop Programming Made Easier: Admiring the Pig Architecture, going with the Pig Latin Application Flow, working through the ABCs of Pig Latin, Evaluating Local and Distributed Modes of Running Pig Scripts, Checking out the Pig Script Interfaces, Scripting with Pig Latin

TEXT BOOKS:

1. Big Java 4th Edition, Cay Horstmann, Wiley John Wiley & Sons, INC
2. Hadoop: The Definitive Guide by Tom White, 3rd Edition, O'reilly
3. Hadoop in Action by Chuck Lam, MANNING Publ.
4. Hadoop for Dummies by Dirk deRoos, Paul C. Zikopoulos, Roman B.Melnyk, Bruce, Brown, Rafael Coss

REFERENCE BOOKS:

1. Hadoop in Practice by Alex Holmes, MANNING Publ.
2. Hadoop MapReduce Cookbook, SrinathPerera, ThilinaGunarathne

COURSE OUTCOMES:

After completion of the course students will be able to:

1. Demonstrate understanding of Hadoop architecture and configure clusters for big data processing.
2. Develop and execute MapReduce programs using the Hadoop API.
3. Implement efficient data I/O handling in Hadoop using Writable interfaces and custom comparators.
4. Create and manage big data workflows using Pig Latin scripting and Pig architecture.
5. Apply Pig scripting and its architecture to efficiently process and analyze data in Hadoop environments.

I Semester	SYSTEM MODELLING AND SIMULATION	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To learn the mathematical foundations of dynamic systems through transfer functions, state-space models, and numerical simulation of differential equations.
2. To apply statistical techniques for data analysis, including regression, variance analysis, and model fitting to support system identification.
3. To explore probabilistic models and stochastic processes such as Monte Carlo methods, Markov chains, and time series for simulating random behaviour.
4. To learn discrete-event simulation methodologies, including event scheduling algorithms and model verification and validation techniques.
5. To develop continuous simulation models using differential equations, bond graphs, and system dynamics for real-world applications.

UNIT I:

Introduction Circuits as dynamic systems, Transfer functions, poles and zeroes, State space, Deterministic Systems, Difference and Differential Equations, Solution of Linear Difference and Differential Equations, Numerical Simulation Methods for ODEs, System Identification, Stability and Sensitivity Analysis.

UNIT II: **Statistical methods, Descriptionofdata, Data-fitting methods, Regression analysis, LeastSquares Method, Analysis of Variance, Goodness of fit.**

UNIT III:

Probability and Random Processes, Discrete and Continuous Distribution, Central Limit theorem, Measure of Randomness, Monte Carlo Methods. Stochastic Processes and Markov Chains, Time Series Models.

UNIT IV:

Modeling and simulation concepts, Discrete-event simulation, Event scheduling/Time advance algorithms, Verification and validation of simulation models.

UNIT V:

Continuous simulation: Modeling with differential equations, Example models, Bond Graph Modeling, Population Dynamics Modeling, System dynamics

TEXT BOOKS:

1. R.L.Woods and K.L.Lawrence, “Modelling and Simulation of Dynamic Systems”, Prentice- Hall,1997.

REFERENCES:

1. Z.Navalih,“VHDLAnalysisandModellingofDigitalSystems”,McGraw-Hill,1993.

2. J.Banks,JS.CarsonandB.Nelson,"Discrete-eventSystemSimulation",2nd Edition,Prentice-Hall of India,1996

COURSE OUTCOMES:

After completion of the course students will be able to:

1. Analyse dynamic systems using transfer functions, state-space models, and solve linear differential/difference equations.
2. Apply statistical methods for data analysis, regression, and goodness-of-fit evaluation.
3. Learn and utilize probability theory, stochastic processes, and Monte Carlo methods for modelling uncertainty.
4. Develop and validate discrete-event simulation models using appropriate algorithms.
5. Model and simulate continuous systems using differential equations, bond graphs, and system dynamics approaches.

I Semester	DIGITAL SYSTEM DESIGN LAB	L	T	P	C
		0	1	2	2

SYSTEMS DESIGN EXPERIMENTS:

- The students are required to design the logic to perform the following experiments using necessary Industry standard simulator to verify the logical /functional operation, perform the analysis with appropriate synthesizer and to verify the implemented logic with different hardware modules/kits (CPLD/FPGA kits).
- Consider the suitable switching function and data to implement the required logic if required.

LIST OF EXPERIMENTS:

A student has to do at least 10 Experiments.

1. Determination of EPCs using CAMP-I Algorithm.
2. Determination of SPCs using CAMP-I Algorithm.
3. Determination of SCs using CAMP-II Algorithm.
4. PLA minimization algorithm (IISc algorithm)
5. PLA folding algorithm(COMPACT algorithm)
6. ROM design.
7. Control unit and data processor logic design
8. Digital system design using FPGA.
9. Kohavi algorithm.
10. Hamming experiments.

COURSE OUTCOMES:

After completion of the course students will be able to:

1. Apply advanced logic design techniques to implement and verify digital systems using industry-standard simulation tools.
2. Analyse and optimize programmable logic structures such as PLAs and ROMs using specialized algorithms.
3. Design and synthesize control units and data processors for complex digital applications.
4. Implement and test digital systems on hardware platforms like CPLD and FPGA kits.
5. Evaluate the performance and correctness of switching functions and error-correcting codes through experimental methods.

I Semester	EMBEDDED SYSTEM DESIGN LAB	L	T	P	C
		0	1	2	2

COURSE OBJECTIVES:

1. To gain practical experience in configuring GPIOs, timers, interrupts, and serial communication using ARM Cortex-M microcontrollers.
2. To interface and control peripheral devices such as LCDs, ADCs, and PWM modules for real-time embedded applications.
3. To implement data storage solutions using micro-SD cards for logging and retrieval in embedded systems.
4. To establish wireless communication using BLE and Wi-Fi for remote monitoring and smart control functionalities.
5. To develop and debug embedded programs using IDE tools and terminal interfaces to ensure reliable system behaviour.

EXPERIMENTS USING ARM CORTEX-M MICROCONTROLLER

(NUCLEO board -F429ZI):

1. Program to configure and control General Purpose Input / Output (GPIO) port pins.
2. Program to demonstrate Serial communication. Transmission from Kit and reception from PC using Serial Port on IDE environment use debug terminal to trace the program.
3. Program to demonstrate Time delay program using built in Timer / Counter feature onIDE environment.
4. Program to demonstrate a simple interrupt handler and setting up a timer.
5. Program to Displaying a message in a 2-line x 16 Characters LCD display and verify the result in debug terminal.
6. Program to demonstrate ADC interfacing.
7. Generation of PWM Signal with the objective of introducing the practical application of timers and fundamental principles of control theory.
8. To integrate a micro-SD card with the computing system for the purpose of storing event logs conveniently on the SD card.
9. To establish a connection between the two computing systems using Bluetooth Low Energy(BLE), with the objective of monitoring pertinent information from one system and facilitating gate control through the other system.
10. To enhance the smart home system by enabling it to host a web page through Wi-Ficonnectivity, thereby allowing users to access information using a smartphone or PC.

COURSE OUTCOMES:

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

1. Develop and debug embedded applications using GPIO, timers, interrupts, and serial communication on ARM Cortex-M microcontrollers.
2. Interface and control peripheral devices such as LCDs, ADCs, PWM modules, and SD cards in embedded systems.
3. Implement wireless communication protocols like BLE and Wi-Fi for real-time data exchange and remote control.
4. Design and test embedded systems for smart applications using industry-standard development environments and tools.
5. Analyse and optimize embedded system performance through hands-on experimentation and system-level integration.

I Semester	SEMINAR – I	L	T	P	C
		0	0	2	1

I Semester	SYSTEM DESIGN WITH RTOS & EMBEDDED LINUX	L	T	P	C
		3	1	0	4

COURSE OBJECTIVES:

1. To introduce the fundamentals of Real-Time Operating Systems (RTOS), including task management, scheduling, system calls, and concurrency control.
2. To explore synchronization and communication mechanisms such as semaphores, message queues, and I/O subsystems, with emphasis on design challenges like deadlocks and priority inversion.
3. To examine exception handling, interrupt management, and timer services essential for real-time system responsiveness.
4. To develop proficiency in Linux kernel operations and shell scripting, including command-line utilities, script control structures, and signal handling.
5. To familiarize students with embedded Linux architecture, device drivers, board support packages, and techniques for porting and debugging real-time applications.

UNIT I: Introduction to RTOS and Task Management

Introduction to Real-Time Operating Systems (RTOS): Key characteristics, scheduler, kernel objects and services, system calls, static and dynamic libraries, cross tool chains, Task management: Defining tasks, task states, scheduling, task operations, synchronization, communication, concurrency.

UNIT II: Synchronization, Communication, and I/O Systems

Semaphores: Operations, use cases, Message Queues: Types, operations, use cases (including pipes, event registers, signals, condition variables), I/O Subsystems: I/O concepts, subsystems, Synchronization and Communication: Resource synchronization methods, critical section, design patterns, priority inversion, common design problems (deadlocks, priority inversion).

UNIT III: Exceptions, Interrupts, and Timer Services

Exceptions and Interrupts: Definitions, applications, spurious interrupts, Timer Services: Real-time clocks, system clocks, programmable interval timers, timer interrupt service routines.

UNIT IV: Linux Kernel and Shell Scripting

Introduction to Linux Kernels: Linux basics, GNU utilities, distributions, access methods (CLI, graphical terminal emulators), Bash Shell Commands: Navigation, file handling, system monitoring, environment variables, user-defined variables, Shell Scripting: Script creation, control structures (if-else, loops, case commands), output redirection, practical examples, handling signals, background scripts, basic script functions, alternative shells (dash, zsh).

UNIT V: Embedded Linux Architecture and Application Porting

Embedded Linux Architecture: Kernel architecture, memory manager, scheduler, file system, I/O and networking subsystems, IPC, user space, startup sequence, Board Support Package: Embedded storage (MTD), embedded file system, embedded device drivers (communication between user space and kernel, character/block drivers, interrupt handling, kernel modules), Porting Applications: Real-time Linux, hard real-time programming, building and debugging (bootloaders, kernel, root file system, device tree).

TEXT BOOKS:

1. Qing Li, Caroline Yao (2020), "Real-Time Concepts for Embedded Systems", CMP Books
2. Chris Simmonds, "Mastering Embedded Linux Programming" - Second Edition, PACKT Publications Limited.
3. Karim Yaghmour, "Building Embedded Linux Systems", O'Reilly & Associates
4. P Raghvan, Amol Lad, Sriram Neelakandan, "Embedded Linux System Design and Development", Auerbach Publications

REFERENCE BOOKS:

1. Albert Cheng, (2022), "Real-Time Systems: Scheduling, Analysis and Verification", Wiley Interscience.
2. Insup Lee, Joseph Leung, and Sang Son, (2018) "Handbook of Real-Time Systems", Chapman and Hall.
3. Krishna and Kang G Shin, (2021), "Real-Time Systems", McGraw Hill
4. Christopher Hallinan, "Embedded Linux Primer: A Practical Real-World Approach", Prentice Hall, 2nd Edition, 2010.
5. Derek Molloy, "Exploring Beagle Bone: Tools and Techniques for Building with Embedded Linux", Wiley, 1st Edition, 2014

COURSE OUTCOMES:

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

1. Apply synchronization and communication techniques to address resource sharing and concurrency issues.
2. Utilize interrupt handling and timer services for responsive embedded systems.
3. Develop shell scripts and manage Linux systems using kernel utilities and scripting methods.
4. Implement embedded Linux architecture and port applications with appropriate device driver integration.

I Semester	IOT ARCHITECTURE & APPLICATIONS	L	T	P	C
		3	1	0	4

COURSE OBJECTIVES:

1. To learn the concepts of IoT Architecture Reference model.
2. To learn the different protocols employed for IoT.
3. To design IoT-based systems for real-world problems.

UNIT-I:

Introduction to IOT, Applications of IOT, Use cases of IOT Introduction- The IOT Today & Progression to Tomorrow – Success. Factors –Strategic Research & Innovation Directions.

UNIT-II:

IOT and Related Issues - IOT & Related Future Internet Technologies – Networks & Communication – Processes & Data Management - Security, Privacy & Trust - Protocol Convergence

UNIT-III:

An Architectural Overview, Reference Model and IOT Architecture - Architecture Reference Model – IOT Reference Architecture: Architecture, Functional, information, deployment and operation views; SOA based Architecture, API-based Architecture, OPEN IoT Architecture for IoT/Cloud Convergence.

UNIT-IV:

IOT Smart Applications, Cloud Service Management and IOT - Connecting IOT to cloud – Cloud Storage for Iot – Data Analytics for IoT – Software & Management Tools for IOT. Application Protocols for IoT: UPnP, CoAP, MQTT, XMPP. SCADA, Web Socket; IP-based protocols: 6LoWPAN, RPL; Authentication Protocols; IEEE 802.15.4.

UNIT-V:

Case study 1: M2M To IOT -M2M Vs IOT – A vision from M2M to IOT – Case Study 2: Cloud-Based Smart-Facilities Management, Healthcare, Environment Monitoring System.

TEXT BOOKS:

1. Arshdeep Bahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, 2015.
2. Manoel Carlos Ramon, "Intel® Galileo and Intel® Galileo Gen 2: API Features and Arduino Projects for Linux Programmers", Apress, 2014.

REFERENCE BOOKS:

1. Hersent, Olivier, David Boswarthick, and Omar Elloumi. The internet of things: Keyapplications and protocols. John Wiley & Sons, 2011.
2. Buyya, Rajkumar, and Amir Vahid Dastjerdi, eds. Internet of Things: Principles and paradigms. Elsevier, 2016.

COURSE OUTCOMES:

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

1. Comprehend the essentials of IoT and its applications
2. Learn the concepts of IoT Architecture Reference model
3. Analyze various IoT Application layer Protocols.
4. Apply IP based protocols and Authentication Protocols for IoT
5. Design IoT-based systems for real-world problems

II Semester	COMPUTER NETWORKS	L	T	P	C
		3	1	0	4

COURSE OBJECTIVES:

1. To introduce the architecture and foundational concepts of the Internet, including access networks, switching techniques, and network delays and congestion.
2. To explore network protocols and addressing schemes such as NAT, ICMP, SNMP, CIDR, IPv6, and routing protocols including RIP and IGRP.
3. To examine transport layer protocols including TCP variants, UDP, and application-layer services like HTTP, FTP, DNS, and socket programming.
4. To learn access methods, IEEE standards, routing algorithms, congestion control, and IP addressing with emphasis on IPv4/IPv6 transition.
5. To familiarize students with advanced networking topics including IP multicasting, VoIP, virtual and optical networking, cellular networks, and network security protocols like SSL, IPsec, TLS, and Kerberos.

UNIT-I:

The internet architecture, Access Networks, the network Core, Peer-to-Peer Networks, Content Distribution Networks, Delay Tolerant Networks, Circuit Switching vs. Packet switching, Packet switching Delays and congestion, Client/Server and Peer-to-Peer Architectures, MAC and LLC, Virtual LAN, Asynchronous Transfer Mode (ATM)

UNIT-II:

Network Address Translator, Internet Control Message Protocol, SNMP, CIDR, IPv6, Routing Protocol Basics in advanced networks, Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), Switching Services, Spanning Tree Protocol (STP), Standard Network Management Protocol.

UNIT-III:

TCP and Mobile TCP, TCP Tahoe and TCP Reno, High speed TCP, Coexistence of UDP and TCP flows, HTTP and HTTPS, FTP and SFTP, Domain Name Service, TCP and UDP sockets

UNIT IV:

Access Methods - Pure ALOHA - Slotted ALOHA. Carrier Sense Multiple Access (CSMA) - Carrier Sense Multiple Access with Collision Detection - Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) - Idle Signal Casting Multiple Access - Packet Reservation Multiple Access – IEEE 802.3 – IEEE 802.4 – IEEE 802.5 – IEEE 802.11 – FDDI – SONET. Routing: Network–Layer Routing, Least-Cost-Path algorithms, Non-Least-Cost-Path algorithms,

Intradomain Routing Protocols, Interdomain Routing Protocols, Congestion Control at Network Layer. Logical Addressing: IPv4 Addresses, IPv6 Addresses - Internet Protocol: Internetworking, IPv4, IPv6, Transition from IPv4 to IPv6. Transport and End-to-End Protocols: Transport Layer, Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Mobile Transport Protocols, performance evaluation of TCP protocol.

UNIT-V:

Explaining IP Multicasting, VOIP, Unified Communication, Virtual Networking, Data center Networking, Introduction to Optical Networking, SONET /SDH Standard, Next generation cellular networks, Secure Socket Layer, IP Sec, TLS, Kerberos, Domain name system Protection

TEXT BOOKS:

1. Computer Networking: A Top-Down Approach, 6/e, James F. Kurose and Keith W. Ross, Pearson Education, 2012.
2. Larry L. Peterson and Bruce S. Davie, Computer Networks: A systems approach, Morgan Kaufman, 5th Edition, 2012
3. Data Communications and Networking, Behrouz A. Forouzan, Fourth Edition, Tata McGraw Hill

REFERENCE:

1. High Speed Networks and Internets – Performance and Quality of Service, William Stallings, Second Edition, Pearson Education.
2. Top-Down Network Design, Priscilla Oppenheimer, Second Edition, Pearson Education
3. William Stallings, Data and Computer Communication, Prentice Hall of India.
4. S.Keshav, An Engineering Approach to Computer Networking , Pearson Education.

COURSE OUTCOMES:

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

1. Apply synchronization and communication techniques to address resource sharing and concurrency issues.
2. Utilize interrupt handling and timer services for responsive embedded systems.
3. Develop shell scripts and manage Linux systems using kernel utilities and scripting methods.
4. Implement embedded Linux architecture and port applications with appropriate device driver integration.
5. Analyze advanced networking concepts such as IP multicasting, VoIP, virtual and optical networking, cellular networks, and security protocols including SSL, IPsec, TLS, and Kerberos

II Semester	SYSTEM ON CHIP DESIGN	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To introduce the fundamentals of System-on-Chip (SoC) design, including CPU/IP cores, co-processors, memory controllers, and physical design methodologies.
2. To explore hardware-software co-synthesis techniques with emphasis on partitioning, performance trade-offs, power optimization, and real-time scheduling.
3. To develop proficiency in virtual prototyping and high-level synthesis, including C-to-RTL translation, system-level design, and source-level optimizations.
4. To examine SoC interconnection structures and protocols such as AMBA AXI, IBM CoreConnect, and Network-on-Chip (NoC) architectures.
5. To enable students to perform system-level modelling, simulation, and performance/power analysis of SoCs and MPSoCs through practical use cases.

UNIT-I:

SoC Design Approach: Basics of Chips and SoC ICs, SoC Design: SoC CPU/IP Cores, Co-processor, Cache, DRAM Controller, SoC Synthesis, Static Timing Analysis (STA), Design for Testability, Verification, Physical Design.

UNIT-II:

Processors: Hardware-Software Co-Synthesis: Partitioning, Cycle Time, Die Area and Cost, Power, Area-Time-Power Trade-offs and Chip Reliability, Real-Time Scheduling, Hardware Acceleration.

UNIT-III:

Virtual Prototyping and High-Level Synthesis (HLS): Mapping High-Level Language Applications to Hardware, Transaction-Level Modeling and Electronic System-Level Languages, Hardware Accelerators, Media Instructions, Coprocessors, System-Level Design Methodology, High-Level Synthesis (C-to-RTL), Hardware Synthesis and Architecture Techniques, Source-Level Optimizations.

UNIT-IV:

SoC Interconnection Structures: Bus-Based Interconnection, Bus Protocols: AMBA AXI Bus, AXI4-Stream, IBM Core Connect, Avalon. Interconnection Structures, Network on Chip (NoC) Interconnection and NoC Systems, IP Interfacing.

UNIT-V:

Performance/Power Analysis of SoCs: System-Level Modeling and Integration, Simulation Platform for Performance Analysis of SoC/MPSoC, Use Cases and Examples.

TEXT BOOKS:

1. Veena Chakravarthi, A Practical Approach to VLSI System on Chip (SoC) Design – A Comprehensive Guide, Springer, 2020
2. S. Pasricha and N. Dutt, On-Chip Communication Architectures: System on Chip Interconnect, Morgan Kaufmann–Elsevier Publishers, 2008.

REFERENCE BOOKS:

1. Keating, M., The Simple Art of SoC Design, Springer, 2011

COURSE OUTCOMES:

At the end of the course, students will be able to:

1. Learn and estimate key design metrics and requirements including area, latency, throughput, energy, and power
2. Implement both hardware and software solutions, formulate hardware/software trade-offs, and perform hardware/software co-design
3. Analyse issues in system-on-chip design associated with interconnection structures, performance, and power consumption
4. Use System C programming and high-level synthesis (HLS) for design and modelling
5. Design and optimize a modern System-on-a-Chip

I Semester	ADVANCED COMPUTER ARCHITECTURE AND ORGANIZATION	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To provide a comprehensive understanding of computer design fundamentals, instruction set architectures, and performance evaluation techniques including Amdahl's Law.
2. To explore pipelining concepts, RISC architectures, and memory hierarchy design to enhance processor performance and efficiency.
3. To examine instruction-level parallelism through both hardware and software approaches, including dynamic scheduling and compiler-level optimizations.
4. To introduce multiprocessor systems and thread-level parallelism, focusing on shared memory architectures and synchronization mechanisms.
5. To familiarize students with interconnection networks, cluster design, and Intel IA-64 architecture, highlighting practical challenges and design pitfalls.

UNIT-I:

Fundamentals of Computer Design: Fundamentals of Computer design, Changing faces of computing and task of computer designer, Technology trends, Cost price and their trends, measuring and reporting performance, Quantitative principles of computer design, Amdahl's law.

Instruction set principles and examples- Introduction, classifying instruction set- memory addressing- type and size of operands, Operations in the instruction set.

UNIT-II:

Pipelines: Introduction, basic RISC instruction set, Simple implementation of RISC instruction set, Classic five stage pipelined RISC processor, Basic performance issues in pipelining, Pipeline hazards, Reducing pipeline branch penalties.

Memory Hierarchy Design: Introduction, review of ABC of cache, Cache performance, Reducing cache miss penalty, Virtual memory.

UNIT-III:

Instruction Level Parallelism (ILP)-The Hardware Approach: Instruction-Level parallelism, Dynamic scheduling, Dynamic scheduling using Tomasulo's approach, Branch prediction, High performance instruction delivery- Hardware based speculation.

ILP Software Approach: Basic compiler level techniques, Static branch prediction, VLIW approach, Exploiting ILP, Parallelism at compile time, Cross cutting issues - Hardware verses Software.

UNIT-IV:

Multi Processors and Thread Level Parallelism: Multi Processors and Thread level Parallelism- Introduction, Characteristics of application domain, Systematic shared memory architecture, Distributed shared – Memory architecture, Synchronization.

UNIT-V:

Inter Connection and Networks: Introduction, Interconnection network media, Practical issues in interconnecting networks, Examples of inter connection, Cluster, Designing of clusters.
Intel Architecture: Intel IA-64 ILP in embedded and mobile markets Fallacies and pit falls.

TEXT BOOKS:

1. John L. Hennessy, David A. Patterson - Computer Architecture: A Quantitative Approach, 3rd Edition, an Imprint of Elsevier.

REFERENCE BOOKS:

1. John P. Shen and Miikko H. Lipasti -, Modern Processor Design: Fundamentals of Super Scalar Processors
2. Computer Architecture and Parallel Processing - Kai Hwang, Faye A.Brigs., MC Graw Hill.
3. Advanced Computer Architecture - A Design Space Approach, DezsoSima, Terence Fountain, Peter Kacsuk, Pearson Ed.

COURSE OUTCOMES:

At the end of this course, students will be able to

1. Analyze fundamental computer design principles, performance metrics, and instruction set architectures using quantitative methods like Amdahl's Law.
2. Explain pipelining concepts, RISC architecture implementation, and memory hierarchy including cache and virtual memory performance.
3. Evaluate instruction-level parallelism using hardware and software approaches such as Tomasulo's algorithm, branch prediction, and VLIW.
4. Analyze multiprocessor systems and thread-level parallelism, including shared memory architectures and synchronization mechanisms.
5. Interconnect networks, cluster design, and examine real-world processor architectures like Intel IA-64, addressing design pitfalls.

I Semester	CYBER SECURITY	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To introduce fundamental security concepts including types of attacks, security services, mechanisms, and models for internetwork security.
2. To explore conventional encryption techniques, cipher modes, and key distribution methods along with secure hashing and message authentication.
3. To develop understanding of number theory and its application in public key cryptography, digital signatures, and certificate-based authentication systems.
4. To examine network-level security protocols such as IPsec, SSL/TLS, and secure email technologies like PGP and S/MIME.
5. To familiarize students with intrusion detection systems, malware threats, password management, and firewall design for robust system protection

UNIT I:

Introduction:

Security Attacks (Interruption, Interception, Modification and Fabrication), Security Services (Confidentiality, Authentication, Integrity, Non-repudiation, access Control and Availability) and Mechanisms, A model for Internetwork security, Internet Standards and RFCs, Buffer overflow & format string vulnerabilities, TCP session hijacking, ARP attacks, route table modification, UDP hijacking, and man-in-the-middle attacks.

UNIT II:

Conventional Encryption:

Conventional Encryption Principles, Conventional encryption algorithms, cipher block modes of operation, location of encryption devices, key distribution Approaches of Message Authentication, Secure Hash Functions and HMAC

UNIT III:

Number Theory: Prime and Relatively Prime Numbers, Modular Arithmetic, Fermat's and Euler'sTheorems, The Chinese Remainder theorem, Discrete logarithms.

Public key: Public key cryptography principles, public key cryptography algorithms, digitalsignatures, digital Certificates, Certificate Authority and key management Kerberos, X.509 Directory Authentication Service

UNIT IV:

IP Security: IP Security Overview, IP Security Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations and Key Management

Transport Level Security: Web Security Requirements, Secure Socket Layer (SSL) andTransport Layer Security (TLS), Secure Electronic Transaction (SET) **Email Privacy:** Pretty Good Privacy (PGP) and S/MIME.

UNIT V:

Intrusion Detection: Intruders, Intrusion Detection systems, Password Management.
Malicious Software: Viruses and related threats & Countermeasures.
Fire walls: Firewall Design principles, Trusted Systems.

TEXT BOOKS:

1. Network Security & Cryptography: Principles and Practices, William Stallings, PEA,Sixth edition.
2. Hack Proofing your Network, Russell, Kaminsky, Forest Puppy, Wiley Dreamtech.

REFERENCE BOOKS:

1. Network Security & Cryptography, Bernard Menezes, Cengage,2010

COURSE OUTCOMES:

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

1. Identify various security attacks, services, and mechanisms, and analyze network vulnerabilities such as buffer overflows and session hijacking.
2. Apply conventional encryption techniques, cipher modes, and message authentication methods including HMAC and secure hash functions.
3. Utilize number theory concepts in public key cryptography and explain digital signatures, certificates, and authentication protocols like Kerberos.
4. Analyze and implement security protocols at the IP and transport layers, including IPsec, SSL/TLS, and secure email mechanisms like PGP and S/MIME.
5. Evaluate intrusion detection systems, malware threats, and firewall architectures for securing network infrastructures.

II Semester	QUANTUM SCIENCE AND TECHNOLOGY	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To provide a foundational understanding of quantum mechanics and its relevance to quantum computation, including qubit representation and quantum evolution.
2. To explore quantum implementation technologies such as atomic, nuclear, superconducting, and photonic systems, with emphasis on entangled photons and quantum channels.
3. To introduce multi-qubit systems and quantum protocols like teleportation, superdense coding, and quantum logic gates.
4. To examine quantum measurement techniques, density operators, and composite systems to understand entanglement and purification.
5. To develop knowledge of quantum algorithms, circuits, and error correction, with applications in cryptography, search, and quantum information processing.

UNIT-I:

Review of Quantum Mechanics and Motivation for Quantum Computation Qubit: The qubit state - matrix and Bloch sphere representation - computational basis unitary evolution.

Quantum implementation: various technologies and limitations. Atomic states, nuclear states, superconductivity, optical qubits. Photonic Integrated circuits. Polarization, modes of waveguides. Single photon sources, high speed detectors. Entangled photons. Quantum channels, free space and fiber optic links.

Photonics and recent developments: photonics and Electronics, fiber optic sensors, bio photonics, optical computers, photonics for quantum communications.

UNIT-II:

Multi-qubitstates-No-cloningtheorem-Superdensecoding-PurestatestoBellstates– Bellinequalities. Protocols with multi-qubits: Swapping - Teleportation - gates: CNOT - Toffoli gate - NAND - FANOUT - Walsh Hadamard.

UNIT-III:

Measurement: Projective operators - General, Projective and POVM measure, Ensemble: Density operators - pure and mixed ensemble - time evolution – post measurement density operator. Composite systems: Partial trace - Reduced density operator - Schmidt decomposition, Purification bipartite entanglement.

UNIT-IV:

Quantum computing: Classical computing using qubits - Quantum parallelism - Deutsch's algorithm - Deutsch Josza algorithm.

UNIT-V:

Quantum circuits:

Basicgates-ABCdecomposition-Graycodes-Universalgates-Principleofdeferred and implicit measurements - Quantum Fourier transform - applications: phase estimation, order finding - factoring, discrete logarithm and hidden subgroup problems - Role of prime factoring in classical cryptography - search algorithms. Quantum error correcting codes, Physical realization of qubits.

TEXT BOOKS:

1. M.A.NielsenandI.L.Chuang,QuantumComputationandQuantumInformation,Cambridge University Press, 2010, 10th Anniversary Edition
2. ChrisBernhardt,QuantumComputingforEveryone,TheMITPress,2019.
3. RayLaPierre,IntroductiontoQuantumComputing,Springer,2021.

REFERENCE BOOKS:

1. Venkateswaran Kasirajan, Fundamentals of Quantum Computing: Theory and Practice, Springer, 2021.Other Suggested Readings:
2. QuantumTheory:ConceptsandMethods, Asher Peres, Kluwer Academic Publishers, 1993

COURSE OUTCOMES:

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

1. Analyze multi-qubit protocols and quantum gates to enable quantum communication and computation.
2. Apply quantum measurement principles and model composite quantum systems using density operators and entanglement.
3. Implement basic quantum algorithms and evaluate their advantages over classical computing paradigms.
4. Design quantum circuits and study their applications in cryptography, error correction, and quantum information processing.
5. Apply knowledge of quantum algorithms, circuits, and error correction in solving problems related to cryptography, search, and quantum information processing.

II Semester	ARM CONTROLLERS AND EMBEDDED C	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To introduce the architecture, instruction sets, and programming techniques of ARM processors, including efficient C and assembly coding practices.
2. To explore exception and interrupt handling mechanisms, memory hierarchy, and management units for robust embedded system design.
3. To develop proficiency in ARM Cortex-M programming, including GPIO, timers, and control structures for real-time applications.
4. To enable students to configure and interface peripherals such as UART, ADC/DAC, keypad, LCD, and seven-segment displays.
5. To familiarize learners with communication protocols like I²C and SPI through practical case studies in smart embedded applications.

UNIT I:

ARM Processor Fundamentals: ARM Design Philosophy, Registers, CPSR, Pipeline, Exceptions, Interrupts and Vector Table, Core Extensions**.**

Introduction to the ARM Instruction Set: Data Processing Instructions, Branch Instructions, Load –Store Instructions, Software Interrupt Instruction, PSR Instructions.

Introduction to the Thumb Instruction Set: Thumb Register Usage, Branch Instructions, Data Processing Instructions, Load-Store Instructions, Stack instructions, Software Interrupt Instruction**.**

Efficient C Programming: Basic C Data Types, C Looping Structures, Register Allocation, Function Calls, Structure Arrangement.

Writing and Optimizing ARM Assembly Code: Writing Assembly Code, Profiling and Cycle Counting, Instruction Scheduling, Register Allocation, Conditional Execution, Looping Constructs.

UNIT II:

Exception and Interrupt Handling: Exception Handling, Interrupts, Interrupt Handling Schemes

Caches: The Memory Hierarchy and Cache Memory, Cache Architecture, Cache Policy, Flushing and Cleaning Cache Memory**.**

Memory Protection Units: Protected Regions, Initializing the MPU, Caches and Write Buffer.

Memory Management Units: Moving from an MPU to an MMU, How Virtual Memory Works, Details of the ARM MMU, Page Tables, Translation Lookaside Buffer, Domains And Memory Access Permission, The Fast Context Switch Extension

UNIT III:

Introduction: Definition of Embedded Systems, Real life examples of embedded systems, Basics of Developing for Embedded Systems

ARM Instruction set Architecture: ARM Cortex-M Organization, Arithmetic, Logical and Shift instructions, Data Movement Instructions, Branch instructions, Program Status register, Bitwise logic operations, Sign and Zero extension, Data Comparison, Memory addressing, Branch and conditional execution, Control structures, Subroutines, 64-bit data processing.

GPIO: GPIO Input Modes, GPIO Output Modes, Memory-mapped I/O, Push button, Programming exercises on GPIO and Push-button

General-purpose Timers: Clock Configuration, Timer Organization, and Counting Modes, Timer Update Events, PWM Registers, Configuration and initialization of PWM block, Programming exercises on the selection of clock source, Timer's concept, and PWM.

UNIT IV:

UART: UART Block, UART Registers, UART baud rate calculation, Configuration and initialization of UART

ADC/DAC: ADC & DAC registers, pin configuration, ADC modes, Configuring ADC and DAC module, Programming exercises on ADC and DAC

Interfacing: Keypad, LCD, and Seven segment display interfacing with ARM Cortex-M3 Microcontroller.

UNIT-V:

Inter-Integrated Circuit (1^2C): I^2C operating modes, Configuration of I^2C, Interface a sensor using I^2C protocol.

Serial Peripheral Interface (SPI): SPI Modes, Master operation, Slave operation, Configuration of SPI

Case Study: Smart Home-Smart Door Locks and Interface a temperature sensor with an I^2C Module to measure the room temperature.

TEXT BOOKS:

1. A.Sloss, D.Symes, C.Wright, "ARM system Developers Guide: Designing and Optimizing System Software", Morgan Kaufmann publishers, 2012.
2. Dr.Yifeng Zhu "Embedded Systems with ARM Cortex-M Microcontrollers in Assembly and C" Third edition, 2018

REFERENCE BOOKS:

1. Steve Furber, "ARM System on Chip Architecture", 2nd ed., Addison Wesley Professional,2000
2. Valvano, J,"Embedded microcomputer systems: real time interfacing", 3rd Edition, Cengage Learning, 2011
3. Frank Vahid, TonyGivargis, "Embedded System Design", J Wiley India,2005
4. Ariel Lutenberg, Pablo Gomez, Eric Pernia "A Beginner's Guide to Designing Embedded System Applications on Arm Cortex-M Microcontrollers

COURSE OUTCOMES:

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

1. Learn and analyze the design aspects, Architecture, and instruction set associated withARM processors. Analyze the C programming optimization methods for ARM processor.
2. Examines various cache-technologies that surround the ARM cores. Analyze the Memory
   Protection Units and Memory Management Units around ARM
3. Learn and analyze the design aspects, Architecture, and instruction set associated with ARM processors and Analyze the GPIO Pins, Interrupt handling, and Timers concepts in task execution
4. Apply the concepts of UART, ADC, and DAC to Peripherals
5. Design and implementation of embedded case study with serial protocols

II Semester	ROBOTICS	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To introduce the fundamental concepts, anatomy, and classifications of robotic systems along with various drive mechanisms.
2. To explore the design and control of end effectors, robot joints, and feedback systems for precise motion and task execution.
3. To learn robot kinematics, transformations, and sensor technologies essential for perception and interaction.
4. To familiarize students with robot cell design, mobile robotics, and emerging micro/nano robotic systems and their applications.
5. To develop proficiency in robot programming languages and techniques for industrial tasks such as pick-and-place, welding, and automation.

UNIT – I:

Introduction: Robot Anatomy-Definition, law of robotics, History and Terminology of Robotics-Accuracy and repeatability of Robotics-Simple problems- Specifications of Robot-Speed of Robot-Robot joints and Links-Robot classifications -Architecture of robotic Systems-Robot Drive systems- Hydraulic, Pneumatic and Electric system.

UNIT - II:

End Effectors and Robot Controls: Mechanical Grippers-Slider crank mechanism, Screw type, Rotary actuators, cam Type-Magnetic grippers -Vacuum grippers -Air operated Grippers-Gripper force Analysis-Gripper design-Simple problems-Robot controls-Point to point control, Continuous path control, Intelligent robot-Control system for robot joint-Control actions-Feedback devices-Encoder, Resolver, LVDT-Motion Interpolations-Adaptive control.

UNIT - III:

Robot Transformations and Sensors: Robot Kinematics- Types – 2D & 3D Transformation-Scaling, Rotation, Translation- Homogeneous coordinates, multiple Transformation Simple problems. Sensors in robot – Touch sensors -Tactile sensor – Proximity and range sensors – Robotic vision Sensor-Force sensor -Light sensors, Pressure sensors.

UNIT - IV:

Robot Cell Design and Micro/Nano Robotics System: Robot work cell design and Control-Sequence control, Operator interface, Safety monitoring devices in Robot-Mobile robot working principle, actuation using MATLAB, NXT Software Introductions- Robot applications- Material

handling, Machine loading and unloading, assembly, Inspection, Welding, Spray painting and undersea robot. Micro/Nanorobotics system Overview-Scaling effect- Top down and bottom up approach- Actuators of Micro/Nano robotics system-Nanorobot communication Techniques-Fabrication of micro/nano Grippers-Wall climbing micro robot working Principles-Biomimetic Robot-Swarm Robot-Nano robot in targeted drug delivery system.

UNIT - V:

Robot Programming-Introduction-Types- Flex Pendant- Lead through programming, Coordinate systems of Robot, Robot controller- major components, Functions-Wrist Mechanism-Interpolation- Interlock commands- Operating mode of robot, Jogging-Types, Robot specifications- Motion commands, end effectors and sensors commands. Robot Languages-Classifications, Structures- VAL- language commands motion control, hand control, program control, pick and place applications, palletizing applications using VAL, Robot welding application using VAL program-WAIT, SIGNAL and DELAY command for communications using simple applications. RAPID- language basic commands- Motion Instructions- Pick and place operation using Industrial robot- manual mode, automatic mode, subroutine command based programming. Move master command language- Introduction, syntax, simple problems. VAL-II programming-basic commands, applications- Simple problem using conditional Statements-Simple Pick and Place applications.

TEXT BOOKS:

1. Mikell P Groover& Nicholas Godrey, Mitchel Weiss, Roger N Nagel, Ashish Dutta,Industrial Robotics, Technology Programming and Applications, McGraw Hill, 2nd Edition.
2. J John Craig, “Introduction to Robotics, Pearson Education, 2018, 4th Edition.

REFERENCE BOOKS:

1. Klafter. R.D, Chmielewski. T.A. and Noggin‟s., “Robot Engineering: An IntegratedApproach”, Prentice Hall of India Pvt. Ltd., 2009
2. S.R. Deb, Robotics Technology and flexible automation, Tata McGraw-Hill Education., 2009

COURSE OUTCOMES:

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

1. Analyze various types of end effectors, control strategies, and robot joint control systems.
2. Apply kinematic transformations and integrate sensors into robotic systems for enhanced functionality.
3. Design robotic work cells and explore applications of micro and nanorobotics across industries.
4. Develop programming solutions for industrial robots using VAL, RAPID, and Move Master languages.

II Semester	INTERNET OF THINGS (IoT) LABORATORY	L	T	P	C
		0	1	2	2

COURSE OBJECTIVES:

1. To introduce students to the foundational concepts of IoT, including architecture, physical layer, and core components like sensors and microcontrollers.
2. To develop hands-on skills in programming and interfacing using Arduino, Scratch, and simulation tools like Tinkercad.
3. To enable learners to build and test IoT applications involving analog/digital I/O, displays, keypads, and embedded sensors.
4. To familiarize students with cloud integration and mobile app development for remote monitoring and control of IoT devices.
5. To explore real-world IoT applications such as smart home automation using platforms like Thing Speak and Blynk.

LIST OF EXPERIMENTS:

1. Introduction to IoT, IoT Architecture, introduction to Physical layer
2. Introduction to sensors, actuators, and transducers. Introduction to microcontrollers and microprocessors
3. Introduction to Arduino. Introduction to Scratch programming, S4A tool, and ArduinoIDE.
4. Introduction to Tinker cad and some practical examples
5. Working with analog, digital inputs & outputs
6. Interfacing Arduino with Embedded sensors and Actuators
7. Interfacing Arduino with additional sensors
8. Working on Displays and interfacing with Arduino
9. Arduino & LCD Based Projects
10. Arduino interfacing with Keypad and its operation
11. Creating the app (app designing using MIT) and controlling your hardware with your app. 12. Introduction to NodeMCU and basic tasks
12. Introduction to Cloud, some IoT Cloud Platforms publishing sensor data to a cloud usingThing speak.
13. Controlling your sensor data using Thing speak and MIT APP Inventor
14. Email notifications, app alerts using Blynk cloud
15. Home Automation Using Blynk app

COURSE OUTCOMES:

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

1. Learn the architecture of IoT systems and the role of sensors, actuators, and microcontrollers in physical computing.
2. Develop and test embedded applications using Arduino, Scratch programming, and simulation tools like Tinkercad.
3. Interface various input/output devices and sensors with microcontrollers for real-time data acquisition and control.
4. Design and implement IoT projects integrating cloud platforms such as ThingSpeak and Blynk for data monitoring and automation.
5. Create mobile applications to interact with IoT hardware and enable smart control features for home automation.

II Semester	SYSTEM DESIGN WITH RTOS & EMBEDDED LINUX LAB	L	T	P	C
		0	0	2	2

COURSE OBJECTIVES:

1. To introduce students to the fundamental concepts and operations of real-time operating systems through hands-on programming.
2. To enable learners to simulate and analyze task scheduling, context switching, and synchronization mechanisms.
3. To develop proficiency in managing multitasking environments using semaphores, mutexes, and interrupt handling techniques.
4. To familiarize students with inter-process communication methods such as queues, pipes, and signal handling.
5. To explore memory management strategies and system-level design principles within RTOS and embedded Linux environments.

Experiments:

1. Develop a program that involves various operations related to real-time tasks, such as creation, deletion, and synchronization.
2. Implement a program to understand and simulate task scheduling algorithms in a real-time operating system.
3. Create a program to explore the concept and mechanisms of context switching in the context of real-time operating systems.
4. Develop a program to illustrate the use of semaphores for synchronization and communication between different tasks.
5. Implement a program that demonstrates the usage of mutexes for managing critical sections in a multitasking environment.
6. Write a program to handle interrupts effectively, exploring the ways an operating system manages and responds to external interrupts.
7. Develop a program to manage and manipulate queues within the real-time operating system environment.
8. Implement a program that showcases the effective management of pipes for inter-process communication.
9. Write a program to understand and implement signal handling mechanisms within the real-time operating system.
10. Develop a program that explores memory management techniques in the context of real-time operating systems.

COURSE OUTCOMES:

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

1. Design and implement real-time task operations including creation, deletion, and synchronization using RTOS principles.
2. Simulate and analyze task scheduling algorithms to evaluate performance in real-time systems.
3. Explore context switching and its impact on multitasking through practical programming exercises.
4. Apply synchronization techniques using semaphores and mutexes to manage concurrent tasks effectively.
5. Develop programs for interrupt handling, inter-process communication, and memory management in RTOS environments.

II Semester	SEMINAR – II	L	T	P	C
		0	0	2	2

III Semester	RESEARCH METHODOLOGY AND IPR	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

1. To learn the fundamentals of identifying and formulating a research problem, including its scope, objectives, and investigative approaches.
2. To develop skills for conducting effective literature reviews, ensuring ethical research practices, and avoiding plagiarism.
3. To acquire proficiency in technical writing, report preparation, and research proposal development and presentation.
4. To gain knowledge of intellectual property rights, including patents, trademarks, copyrights, and international patenting procedures.
5. To explore emerging trends in IPR, including its application to biological systems, software, and traditional knowledge through case studies.

UNIT-I:

Meaning of research problem, Sources of research problem, Criteria Characteristics of a good researchproblem, Errors in selecting a research problem, Scope and objectives of researchproblem. Approachesof investigation of solutions for research problem, data collection, analysis, interpretation, Necessaryinstrumentations

UNIT-II:

Effective literature studies approaches, analysis Plagiarism, Research ethics,

Effective technical writing, how to write report, Paper Developing a Research Proposal, Format ofresearch proposal, a presentation and assessment by a review committee.

UNIT -III:

Nature of Intellectual Property: Patents, Designs, Trademarks and Copyright. Process of Patenting andDevelopment: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.

UNIT -IV:

Patent Rights: Scope of Patent Rights, Licensing and transfer of technology, Patent information anddatabases, Geographical Indications.

UNIT -V:

New Developments in IPR: Administration of Patent System. New developments in IPR, IPR of Biological Systems, Computer Software etc. Traditional knowledge Case Studies, IPR and IITs.

TEXT BOOKS

1. Stuart Melville and Wayne Goddard, "Research methodology: an introduction for science&engineering students'".
2. Wayne Goddard and Stuart Melville, "Research Methodology: An Introduction"

REFERENCES

1. Ranjit Kumar, 2nd Edition, "Research Methodology: A Step-by-Step Guide for beginners"
2. Halbert, "Resisting Intellectual Property", Taylor & Francis Ltd,2007.
3. Mayall, "Industrial Design", McGraw Hill,1992.
4. Niebel "Product Design", McGraw Hill,1974.
5. Asimov, "Introduction to Design", Prentice Hall,1962.
6. Robert P. Merges, Peter S. Menell, Mark A. Lemley, "Intellectual Property in New TechnologicalAge",2016.
7. T. Ramappa, "Intellectual Property Rights Under WTO", S. Chand,2008

COURSE OUTCOMES:

At the end of this course, students will be able to

1. Learn research problem formulation.
2. Analyze research related information
3. Follow research ethics
4. Learn that today's world is controlled by Computer, Information Technology, but tomorrow world will be ruled by ideas, concept, and creativity.
5. Familiarize that when IPR would take such important place in growth of individuals & nation, it is needless to emphasis the need of information about Intellectual Property Right to bepromoted among students in general & engineering in particular.
6. Learn that IPR protection provides an incentive to inventors for further research workand investment in R & D, which leads to creation of new and better products, and in turn brings about, economic growth and social benefits.

III Semester	SUMMER INTERNSHIP/INDUSTRIAL TRAINING	L	T	P	C
		0	0	0	2

III Semester	COMPREHENSIVE VIVA#	L	T	P	C
		0	0	0	2

III Semester	DISSERTATION PART– A$	L	T	P	C
		0	0	20	10

IV Semester	DISSERTATION PART– B%	L	T	P	C
		0	0	32	16