THE NATIONAL INSTITUTE OF ENGINEERING

Manandavadi Road, Mysuru

DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING

Curriculum Structure and Syllabus

2025-2027

M. Tech.

in

“Embedded System and Computing”

Department of Electronics and Communication Engineering

VISION

Department of Electronics and Communication will be globally recognized that imparts high quality education and enables innovation, research and teamwork capabilities to students, whose graduates serve diverse needs of society.

MISSION

M1: To design academic curricula and activities to produce competent Electronics graduates.

M2: To develop acumen to absorb emerging knowledge and to Life-Long Learning.

M3: To provide group activities in the area of Electronics and Communication Engineering that enable innovation and teamwork.

M4: To interact with professional bodies and corporates in Electronics, Communication and IT sectors.

PROGRAMME EDUCATIONAL OBJECTIVES

PEO1: Excel the in-depth knowledge in Embedded System and Computing to analyse, evaluate and design Embedded computing systems.

PEO2: Achieve Proficiency in Industry or Academia and Research & Development in Embedded System and Computing and its allied areas.

PEO3: Exhibit professional ethics, integrity, teamwork and leadership qualities through continuous learning.

PROGRAMME OUTCOMES

Students graduating from M. Tech. – Embedded System and Computing of department of Electronics and Communication Engineering shall have the ability to:

PO1: An ability to independently carry out research /investigation and development work to solve practical problems.

PO2: An ability to write and present a substantial technical report/document.

PO3: Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PROGRAMME SPECIFIC OUTCOMES

PSO1: Apply the advanced concepts of Embedded computing System Design with real-time constraints using advanced processors and FPGA based systems.

PSO2: Use the cutting-edge technologies in both hardware and software, to solve real world multi-disciplinary problems and arrive at a viable solution.

PSO3: Independently carry out research on diverse Embedded computing System strategies to address practical problems and present a substantial technical report.

I SEMESTER (Core Courses related to Embedded System and Computing)											
Sl. No.	**Course Type**	**Course Code**	**Course Title**	**Teaching Hours Per Week**			**Examination**				**Credits**
				Theory	**Tutorial / SDA**	**Practical / Seminar**	**Duration Hrs**	**CIE Marks**	**SEE Marks**	**Total Marks**	
				L	**T/SDA**	**P**					
1	PCC	MESC101	Embedded System Programming	3	0	0	03	50	50	100	3
2	IPCC	MESC102	Real-Time Operating Systems (RTOS)	3	0	2	03	50	50	100	4
3	PCC	MESC103	Advanced Computer architecture	3	0	0	03	50	50	100	3
4	PEC	MESC104x	**Professional Elective I**	3	0	0	03	50	50	100	3
5	PEC	MESC105x	**Professional Elective II**	3	0	0	03	50	50	100	3
6	PECL	MESCL106	Computing Laboratory – 1	0	0	4	04	50	50	100	2
7	NCMC	MRMI107	Research Methodology and IPR (Online)	Online Courses (online.vtu.ac.in)							PP
							Total	**300**	**300**	**600**	**18**

Professional Elective I		**Professional Elective II**	
MESC104A	ASIC Design	MESC105A	Embedded and Automation
MESC104B	Advanced Computer Networking	MESC105B	Adaptive Signal Processing
MESC104C	Advanced Wireless Communication networks	MESC105C	Multimedia & Applications
MESC104D	Data Analytics and Edge Computing	MESC105D	Process Control

Note: **MRMI107 – Research Methodology and IPR -** Non Credit Mandatory Course (NCMC) if students have not studied this course in their undergraduate program then he / she has to take this course at **http://online.vtu.ac.in** and to qualify for this course is compulsory before completion of the minimum duration of the program (Two years), however, this course will not be considered for vertical progression.

II SEMESTER Specialization in Embedded System and Computing (ESC)

Sl. No.	Course Type	Course Code	Course Title	Teaching Hours Per Week			Examination				Credits
				Theory	Tutorial / SDA	Practical / Seminar	Duration Hrs	CIE Marks	SEE Marks	Total Marks	
				L	T/SDA	P					
1	IPCC	MESC201	IoT Architecture and Protocol Systems	3	2	0	03	50	50	100	4
2	PCC	MESC202	Embedded Security	3	0	0	03	50	50	100	3
3	PCC	MESC203	Machine Learning and Deep Learning	3	0	0	03	50	50	100	3
4	PCC	MESC204	Hardware and Software Co-Design	3	0	0	03	50	50	100	3
5	PEC	MESC205x	**Professional Elective III**	3	0	0	03	50	50	100	3
6	PEC	MESC206x	**Professional Elective IV**	3	0	0	03	50	50	100	3
7	PCCL	MESCL207	Computing Laboratory – 2	0	0	4	04	50	50	100	2
8	AEC / SEC	MESC258x	Ability/Skill Enhancement Course (Offline/Online)	0	0	2	02	50	50	100	1
				1	0	0	01				
							Total	**400**	**400**	**800**	**22**

Ability / Skill Enhancement Courses

1	AEC / SEC	MESCL258A	Software Tools for Embedded system	0	0	2	
2		MESCL258B	Optimization Techniques	0	0	2	

3		MESCL258C	Python Programming	0	0	2	

For Professional Elective Course

Professional Elective Course-III (PEC)		**Professional Elective Course-IV (PEC)**	
MESC205A	Multicore and RISC - V Architecture	MESC206A	System on Chip
MESC205B	Cryptography and Network Security	MESC206B	Industrial IoT sensors and Interfacing
MESC205C	High Performance Computing	MESC206C	RTL – Simulation & Synthesis
MESC205D	FPGA Based System Design	MESC206D	Automotive Embedded Product Development

Note: **Ability Enhancement Courses (AEC):** These courses are designed to help students enhance their skills in communication, language, and personality development. They also promote a deeper understanding of subjects like social sciences and ethics, culture and human behaviour, human rights, and the law.

Skill Enhancement Course (SEC): Skill Enhancement Course means a course designed to provide value- based or skill-based knowledge and should contain both theory and lab/hands-on/training/fieldwork. The main purpose of these courses is to provide students with life skills in the hands-on mode to increase their employability.

If AEC/SEC courses are ONLINE (MOOCs) courses, suggested by the concerned board of studies. These courses will be made available on **www. online.vtu.ac.in.**

Subject Code: MESC101 **Course: Embedded System Programming**

Credits: 3 **L:T:P: 3-0-0**

CIE: 50% **SEE: 50%**

SEE Hrs.: 3 Hrs. **Max. Marks: 100**

Prerequisites if any	Programming Knowledge in C or Shell programming
Learning objectives	1. **Understand systems engineering principles** and their application to embedded system design and development. 2. **Analyze hardware–software co-design requirements** and select appropriate architectures for embedded solutions. 3. **Apply modelling, simulation, and verification techniques** for reliable and optimized embedded system performance. 4. **Evaluate design trade-offs** in terms of power, performance, cost, and scalability for embedded applications. 5. **Integrate systems engineering processes** with lifecycle management, testing, and deployment in embedded environments.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Explain the fundamentals of systems engineering and its relevance to embedded systems design.	L2
CO2	Apply requirement analysis and architectural design principles to create embedded system solutions.	L3
CO3	Develop and simulate embedded system models using appropriate tools and techniques.	L4
CO4	Evaluate and optimize embedded systems considering performance, cost, and power constraints.	L2
CO5	Demonstrate the ability to plan, execute, and document a complete embedded system development lifecycle.	L3

CO-PO Mapping

COs	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**	**PSO3**
CO1	2	2	2	1	2	1
CO2	2	2	2	1	2	1
CO3	2	2	2	1	2	1
CO4	2	2	2	1	2	1
CO5	2	2	2	1	2	1

Course Structure

Module-1 Embedded Systems & Low-Level Programming Foundations		***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	Overview of Embedded Systems & System Software	1	-	-
1.2	Hardware–software interface: memory-mapped I/O, registers	2	-	-
1.3	Cross-compilation tool chains, build systems (Make, C Make)	1	-	-
1.4	Bare-metal C & Assembly programming for ARM Cortex-M	2	-	-
1.5	Startup code, linker scripts, memory sections (.text, .data, .bss)	2	-	-
Module-2 Device Drivers & Peripheral Programming				
2.1	Types of device drivers (character, block, network)	2	-	-
2.2	Writing and loading kernel modules (Linux & RTOS drivers)	2	-	-
2.3	GPIO, UART, SPI, I²C programming from scratch	2	-	-
2.4	Interrupt handling and deferred work (task lets, work queues)	2	-	-
Module-3 Real-Time Operating Systems & Multitasking				
3.1	RTOS architecture: task scheduling, priorities, context switching	2	-	-
3.2	Task synchronization: semaphores, mutexes, message queues	2	-	-
3.3	ISR–task communication, tickles idle mode	2	-	-
3.4	Memory management in RTOS environments	2	-	-
Module-4 Embedded File Systems & Inter-Process Communication				
4.1	File system basics in embedded devices (FAT, Little FS, YAFFS)	2	-	-
4.2	Writing file system drivers in embedded Linux	2	-	-
4.3	IPC in embedded Linux: pipes, message queues, shared memory	2	-	-
4.4	Bootloaders and firmware upgrade mechanisms	2	-	-
Module-5 Embedded Networking & Security in Systems Programming				
5.1	Network stack basics in embedded Linux (TCP/IP, UDP)	2	-	-
5.2	Implementing socket programming in embedded systems	2	-	-
5.3	Secure firmware update process, cryptographic primitives in embedded code	2	-	-
5.4	Memory protection, stack overflow prevention, secure boot	2	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of No. of Practical Hours			**0**

Suggested Learning Resources:

Textbooks:

1. **Raj Kamal** – *Embedded Systems: Architecture, Programming and Design* (for fundamentals)
2. **Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman** – *Linux Device Drivers* (O'Reilly)
3. **Yiu, Joseph** – *The Definitive Guide to ARM Cortex-M3 and M4 Processors*

Reference Books:

1. **Jean Labrosse** – *MicroC/OS-II: The Real-Time Kernel*
2. **Chris Simmonds** – *Mastering Embedded Linux Programming*

Course Code: MESC102 **Course: Real Time Operating Systems**

Credits: 4 **L:T:P 3-0-2**

CIE: 50% **SEE: 50%**

SEE Hours: 3 Hrs **Max. Marks: 100**

Prerequisites if any	Operating systems and programming in C
Learning objectives	1. Understand the architecture and core concepts of the RTOS. 2. Learn to develop and debug applications using the Momentics IDE. 3. Gain knowledge of process and thread management, including synchronization techniques. 4. Explore inter-process communication (IPC) methods and their applications. 5. Understand hardware programming concepts, including interrupt handling and memory access. 6. Learn to build and configure boot/OS images for embedded systems.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Describe the OS architecture and its microkernel-based design.	L2
CO2	Apply process/thread management and synchronization techniques.	L3
CO3	Implement inter-process communication methods for real-time systems.	L4
CO4	Develop strong knowledge on the POSIX standards that help in System Application Development.	L2

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	2	1	2	1	1
CO2	3	2	3	3	2	1
CO3	3	2	3	3	2	2
CO4	2	2	2	2	1	1

3 – Strong 2 – Medium 1 – Low

Course Structure

Module – 1		*No. of Lecture Hours*	*No. of Tutorial Hours*	*No. of Practical Hours*
1.1	Introduction to QNX OS Architecture	2	-	-
1.2	Introduction to QNX OS Architecture: microkernel, process manager and standards	2	-	-
1.2	Introduction to inter-process communication IPC and synchronization.	2	-	-
1.3	Interrupt handling, scheduling	2	-	-
Module – 2				
2.1	Process manager, resource manager	3	-	-
2.2	Boot sequence, security, OS services	2	-	-
2.3	Process and threads management: creation, termination and cleanup	3	-	-
Module – 3				
3.1	Synchronization techniques: mutexes and condition variables	2	-	-
3.2	Overview of IPC methods in QNX: message passing	3	-	-
3.3	Pulses, shared memory, and event delivery	3	-	-
Module – 4				
4.1	Introduction to hardware programming: Hardware IO	2	-	-
4.2	Handling interrupts	3	-	-
4.3	Introduction to timing architecture: Getting and setting the system clock	3	-	-
Module – 5				
5.1	Introduction to timers, high resolution timers	2	-	-
5.2	Design considerations, kernel timeouts	2		
5.3	Components of a boot image: startup code, kernel, drivers, and scripts, Building and loading boot images onto target hardware,	2	-	-
5.4	Introduction to resource managers and their implementation	2	-	-
Total No. of Lecture Hours		**40**		
Total No. of Tutorial Hours			**0**	
Total No. of No. of Practical Hours				**0**

Suggested Learning Resources:

Text Books:

1. QNX Neutrino RTOS User's Guide, QNX Software Systems.
2. Programming for Embedded Systems, Michael Barr, O'Reilly Media.
3. Hands-on RTOS with Microcontrollers, Brian Amos, Packt Publishing, 2020

Reference Books:

1. Operating System Concepts, Abraham Silberschatz, et al., 9th Edition, Wiley, 2018.
2. Real-Time Embedded Components And Systems: With Linux and RTOS, Sam Siewert.

Online Resources:

- https://blackberry.qnx.com/en/products/qnx-everywhere
- https://gitlab.com/qnx
- https://www.reddit.com/r/QNX/
- https://stackoverflow.com/questions/tagged/qnx
- https://www.youtube.com/qnxcam
- https://gitlab.com/qnx/projects
- https://github.com/qnx-ports

List of Lab Experiments – 10 Hours

1. QNX configuration and application development using QNX Momentics IDE
2. Process and thread creation, management, and synchronization: Hands-on exercises: process/thread creation and synchronization.
3. Implementation of IPC methods: passing message and shared memory.
4. Hands-on exercises: message passing and shared memory.
5. Hands-on exercises: interrupt handling and timing mechanisms.
6. Building and deploying QNX boot/OS images.
7. Mini capstone project: Design and implement a QNX-based embedded system.

Course Code: MESC103 **Course: Advanced Computer Architecture**

Credits: 3 **L:T:P: – 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks:100**

Prerequisites if any	Digital Electronics
Learning objectives	1. This course aims to develop understanding in the advanced design principles of a computer system by addressing key issues such as instruction set design, micro-architecture of scalar and superscalar processors along with the interaction of other hardware components in a computer system. 2. This course aids you to acquire necessary skills for analysing and estimating the performance of computing systems. 3. In this course, there is a strong emphasis on the study of various constraints in the design of single and multi-processor systems. Students will complete this course with an appreciation and understanding of processor design issues relating to simplicity of implementation, performance-enhancement techniques, and power-reduction methods.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Demonstrate concepts of parallelism in hardware/software.	L1
CO2	Discuss memory organization and mapping techniques.	L2
CO3	Describe architectural features and Interpret performance of advanced processors.	L2

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	1	2	3	2	3
CO2	3	1	2	3	2	3
CO3	3	1	2	3	2	3

3 – Strong 2 – Medium 1 – Low

Course Structure

	Module – 1 - PARALLEL COMPUTER MODELS	***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	Evolution of Computer architecture, system attributes to performance,	1	-	-
1.2	Multi processors and multi computers, Multi-vector and SIMD computers, PRAM and VLSI models-Parallelism in Programming,	2	-	-
1.3	conditions for Parallelism-Program Partitioning and	1	-	-
1.4	Scheduling-program flow	1	-	-
1.5	Mechanisms-Speed up performance laws-Amdahl‘s law, Gustafson‘s law-Memory bounded speedup Model.	3	-	-
	Module – 2 - MEMORY SYSTEMS AND BUSES			
2.1	Memory hierarchy-cache and shared memory concepts	1	-	-
2.2	Cache memory organization-cache addressing models,	1	-	-
2.3	Aliasing problem in cache, cache memory mapping techniques	2	-	-
2.4	Shared memory organization-Interleaved memory organization	1	-	-
2.5	Lower order interleaving, Higher order interleaving.	1	-	-
2.6	Back plane bus systems-Bus addressing, arbitration and transaction.	2	-	-
	Module – 3 – ADVANCED PROCESSORS			
3.1	Instruction set architectures	1	-	-
3.2	CISC and RISC scalar processors	1	-	-
3.3	Super scalar processors-VLIW architecture	1	-	-
3.4	Multivector and SIMD computers-	1	-	-
3.5	Vector processing principles-Cray Y-MP 816 system	1	-	-
3.6	Inter processor communication	1	-	-
3.7	RISC V architecture	2	-	-
	Module – 4 – MULTI PROCESSOR AND MULTI COMPUTERS			
4.1	Multiprocessor system interconnects- Cross bar switch,	1	-	-
4.2	Multiport memory-Hot spot problem,	2	-	-
4.3	Message passing mechanisms-	1	-	-
4.4	Pipelined processors-Linear pipeline, on linear pipeline Instruction pipeline design-Arithmetic pipeline design.	2	-	-
4.5	Introduction to Network on chip and its features	2	-	-
	Module – 5 – DATA FLOW COMPUTERS AND VLSI COMPUTATIONS			
5.1	Data flow computer architectures-Static, Dynamic	1	-	-
5.2	VLSI Computing Structures-Systolic array architecture, mapping algorithms into systolic arrays	2	-	-
5.3	Reconfigurable processor array-VLSI matrix arithmetic processors	1	-	-

5.4	VLSI arithmetic models, partitioned matrix algorithms, matrix arithmetic pipelines.	2	-	-
5.5	Neuromorphic computing architectures	2	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of Practical Hours			**0**

Suggested Learning Resources:

Textbooks:

1. Kai Hwang, Advanced Computer architecture Parallelism ,scalability ,Programmability ‖, Mc Graw Hill,N.Y, 2003
2. Kai Hwang and F.A. Briggs, Computer architecture and parallel processor‖' Mc Graw Hill, N.Y, 1999

Reference Books:

1. David A. Pearson and John L. Hennessey, Computer organization and design‖ Elsevier, Fifth edition, 2014.
2. David Culler, J.P. Singh and Anoop Gupta, Parallel Computer Architecture: A Hardware/Software Approach, Morgan Kaufmann, first edition, 1998.
3. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer Organization, McGraw Hill, 5th Ed, 2014

Online Resources:

1. Course Name: Advanced Computer Architecture Course Link: https://nptel.ac.in/courses/106/103/106103206/ Course Instructor: Prof. John Jose, IIT, Guwahati
2. Course Name: High Performance Computer Architecture Course Link: https://nptel.ac.in/courses/106/105/106105033/ Course Instructor: Prof. A. Pal, IIT, Kharagpur

PROFESSIONAL ELECTIVE – I

Course Code: MESC104A **Course: ASIC Design**

Credits: 3 **L:T:P: – 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks:100**

Prerequisites if any	Digital Electronics and Circuit Theory
Learning objectives	Students will be able to learn 1. ASIC Design Flow. 2. Physical design for the embedded computing system design

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Describe the concepts of ASIC design methodology, data path elements, logical effort and FPGA architectures.	L2
CO2	Analyze the design of FPGAs and ASICs suitable for specific tasks, perform design entry and explain the physical design flow.	L3
CO3	Design data path elements for ASIC cell libraries and compute optimum path delay.	L2

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	1	2	3	2	2
CO2	3	1	3	3	2	2
CO3	3	1	3	3	2	2

Mapping Strength: **Strong– 3** **Medium – 2** **Low – 1**

Course Structure

Module – 1- Introduction to ASICs		*No. of Lecture Hours*	*No. of Tutorial Hours*	*No. of Practical Hours*
1.1	Full custom, Semi-custom and Programmable ASICs,	1	-	-
1.2	ASIC Design flow, ASIC cell libraries, CMOS Logic, Adders: Carry skip, Carry bypass	3	-	-
1.3	Carry saves, Carry selects, Conditional sum, Multiplier (Booth encoding),	2	-	-
1.4	Data path Operators, I/O cells, Cell Compilers.	2	-	-
Module-2- ASIC Library Design				

2.1	Logical effort: Predicting Delay, Logical area and logical efficiency, Logical paths, cells	3	-	-
2.2	Optimum delay and number of stages, library cell design. Programmable ASIC Logic Cells: MUX as Boolean function generators,	2	-	-
2.3	Acted ACT: ACT 1, ACT 2 and ACT 3 Logic Modules, Xilinx LCA: XC3000 CLB, Altera FLEX and MAX, Programmable ASIC I/O Cells: Xilinx and Altera I/O Block.	3	-	-
Module-3- Low-level design entry				
3.1	Schematic entry: Hierarchical design, The cell library, Names, Schematic Icons & Symbols, Nets, Schematic Entry for ASICs, Connections	3	-	-
3.2	vectored instances & buses, Edit in place, attributes, Netlist screener. ASIC Construction: Physical Design, CAD Tools System partitioning, Estimating ASIC size.	2	-	-
3.3	Partitioning: Goals and objectives, Constructive Partitioning, Iterative Partitioning Improvement, KL, FM and Look Ahead algorithms.	3	-	-
Module 4: Floor planning and placement				
4.1	Goals and objectives, Measurement of delay in Floor planning, Floor planning tools, Channel definition, I/O and Power planning and Clock planning.	4	-	-
4.2	Placement: Goals and Objectives, Min-cut Placement algorithm, Iterative Placement Improvement, Time driven placement methods, Physical Design Flow.	4	-	-
Module 5: Routing				
5.1	Global Routing: Goals and objectives, Global Routing Methods, Global routing between blocks, Back-annotation.	3	-	-
5.2	Detailed Routing: Goals and objectives, Measurement of Channel Density, Left-Edge Algorithm, Area-Routing Algorithms, Multilevel routing, Timing –Driven detailed routing, Final routing steps, Special Routing, Circuit extraction and DRC.	5	-	-
	Total No. of Theory Hours	40		
	Total No. of Tutorial Hours		**0**	
	Total No. of Practical Hours			**0**

Textbooks:

1. "Application -Specific Integrated Circuits", Michael John Sebastian Smith Addison-Wesley Professional 2005

Reference books:

1. "VLSI Design: A Practical Guide for FPGA and ASIC Implementations" Vikram Arkalgud Chandrasetty, Springer, ISBN: 978-1-4614-1119-2. 2011
2. "An ASIC Low Power Primer" Rakesh Chadha, Bhasker J Springer, ISBN: 978-14614-4270-7

Online Resources:

1. nptel.ac.in/courses/106105161

Course Code: MESC104B **Course: Advanced Computer Networking**

Credits: 3 **L:T:P: – 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks:100**

Prerequisites if any	Communication Networks
Learning objectives	1. Describe how computer networks are organized with the concept of layered approach. 2. Analyze the contents in a given Data Link layer packet, based on the layer concept. 3. Describe classless addressing scheme. 4. Describe the working of routing protocols, resource allocation method and congestion control mechanism.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Comprehend the use of Computer Network by different applications along with the key metrics used to measure the performance of the network	L2
CO2	Compare Various Network architectures and apply fundamental protocols for networking.	L3
CO3	Discuss Resource Allocation techniques, its variations, and distinguish various congestion control mechanism	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	1	2	3	2	2
CO2	3	3	1	2	2	3
CO3	3	2	1	2	2	2

3 – Strong 2 – Medium 1 – Low

Course Structure

Module – 1: Foundation and Internetworking		***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	Building a Network, Requirements, Perspectives, Scalable Connectivity.	2	-	-
1.2	Cost-effective Resource sharing, Support for Common Services, Manageability, Protocol layering	2	-	-
1.3	Performance, Bandwidth and Latency, Delay X Bandwidth Product, Perspectives on Connecting	2	-	-
1.4	Classes of Links, Reliable Transmission, Stop-and-Wait, Sliding Window	2	-	-
Module – 2: Internetworking- I				
2.1	Switching and Bridging, Datagrams, Virtual Circuit Switching, Source Routing, Bridges and LAN Switches.	2	-	-
2.2	Basic Internetworking (IP), Service Model, Global Addresses.	2	-	-
2.3	Datagram Forwarding in IP, subnetting and classless addressing.	2	-	-
2.4	Address Translation (ARP), Host Configuration (DHCP), Error Reporting (ICMP).	2	-	-
Module – 3: Internetworking- II				
3.1	Network as a Graph, Distance Vector (RIP), Link State (OSPF), Metrics,	2	-	-
3.2	The Global Internet, Routing Areas, Routing among Autonomous systems (BGP), IP Version 6(IPv6).	3	-	-
3.3	Routing among Mobile Devices-Challenges for Mobile Networking, Routing to Mobile Hosts (Mobile IP), Mobility in IPv6	3	-	-
Module – 4: End-to-End Protocols				
4.1	Simple Demultiplexer (UDP), Reliable Byte Stream (TCP), End-to-End Issues.	2	-	-
4.2	Segment Format, Connecting Establishment and Termination, Sliding Window Revisited, Triggering Transmission.	2	-	-
4.3	Adaptive Retransmission, Record Boundaries, Queuing Disciplines, FIFO, Fair Queuing.	2	-	-
4.4	TCP Congestion Control, Additive Increase/ Multiplicative Decrease, Slow Start, Fast Retransmit.	2	-	-
Module – 5: Congestion Control and Resource Allocation				
5.1	Congestion-Avoidance Mechanisms, DEC bit.	2	-	-
5.2	Random Early Detection (RED), Source-Based Congestion Avoidance.	2	-	-
5.3	The Domain Name System (DNS).	2	-	-

5.4	Electronic Mail (SMTP, POP, IMAP, MIME), World Wide Web (HTTP).	2	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of Practical Hours			**0**

Textbooks:

1. Larry Peterson and Bruce S Davis "Computer Networks: A System Approach" 5th Edition, Elsevier - 2014.

Reference Books:

1. Douglas E Comer, "Internetworking with TCP/IP, Principles, Protocols and Architecture" 6th Edition, PHI – 2014.
2. Uyless Black "Computer Networks, Protocols, Standards and Interfaces" 2nd Edition – PHI
3. Behrouz A Forouzan "TCP/IP Protocol Suite" 4th Edition – McGraw-Hill.
4. Behrouz A Forouzan "Data Communication and Networking" 5th Edition – McGraw-Hill

Online resource link:

1. https://onlinecourses.nptel.ac.in/noc23_cs35/preview
2. https://archive.nptel.ac.in/courses/106/105/106105183/

Course Code: MESC104C **Course: Advanced Wireless Communication Networks**

Credits: 3 **L:T:P: – 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks:100**

Prerequisites if any	Communication Networks, Wireless communication, Embedded System
Learning objectives	After successful completion of this course, students will be able to: 1. Analyze PHY and MAC layer techniques used in modern wireless networks for embedded devices. 2. Evaluate different wireless standards (BLE, LoRa, NB-IoT, Wi-Fi 6, 5G) for power consumption, latency, and throughput in embedded scenarios. 3. Design wireless network architectures for IoT and real-time embedded applications considering constraints like range, cost, and energy efficiency. 4. Implement and test wireless connectivity on embedded platforms (e.g., ESP32, nRF52, STM32 with wireless modules).

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	To understand advanced principles of wireless communication networks relevant to embedded and IoT devices.	L2
CO2	To analyse PHY, MAC, and network layer interactions for efficient wireless connectivity.	L3
CO3	To explore emerging wireless technologies (5G, 6G, LPWAN, TSN over wireless) for embedded applications.	L2
CO4	To develop hands-on skills in designing, simulating, and prototyping wireless networked embedded systems.	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	2
CO2	3	1	2	2	1	2
CO3	3	1	2	2	1	2
CO4	3	3	2	2	3	2

3 – Strong 2 – Medium 1 – Low

Course Structure

Module – 1 - Fundamentals of Wireless Networks for Embedded Systems		***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	Review of digital & wireless communication principles (modulation, coding, channel models)	2	-	-
1.2	Embedded device constraints in wireless networks: energy, memory, latency, range	2	-	-
1.3	Link budget & coverage estimation for IoT devices	2	-	-
1.4	Spectrum allocation & ISM band regulations	2	-	-
Module – 2 - Physical Layer Technologies				
2.1	Advanced modulation (OFDM, SC-FDMA, spread spectrum)	2	-	-
2.2	MIMO, beamforming, and diversity techniques for small devices	2	-	-
2.3	Channel estimation & equalization in constrained environments	2	-	-
2.4	Error control coding (LDPC, Polar, convolutional codes) in IoT links	2	-	-
Module – 3 – MAC Layer Protocols and Scheduling				
3.1	MAC for low-power embedded devices (CSMA/CA, TDMA, hybrid)	2	-	-
3.2	QoS and latency control for time-critical embedded applications.	2	-	-
3.3	Synchronization and scheduling in wireless sensor/actuator networks	2	-	-
3.4	Cross-layer optimization for energy and reliability.	2	-	-
Module – 4 – Wireless Network Protocols & Standards, Network Layer and Routing in Wireless Embedded Networks				
4.1	**Short range:** BLE, ZigBee, UWB, Thread, NFC	1	-	-
4.2	**Medium range:** Wi-Fi 6/HaLow, IEEE 802.15.4	1	-	-
4.3	**Long range:** LoRaWAN, Sigfox, NB-IoT, LTE-M	1	-	-
4.4	**High performance:** 5G URLLC/mMTC, emerging 6G	1	-	-
4.5	IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN)	2	-	-
4.6	Routing in Wireless Sensor Networks (WSN) & IoT (RPL, AODV, DSR)	1	-	-
4.7	Mobile Ad Hoc Networks (MANET) and Vehicular Ad Hoc Networks (VANET)	1	-	-
Module – 5 – Security and Emerging Trends				
5.1	Secure communication protocols for embedded wireless networks	2	-	-
5.2	Key management and lightweight encryption for constrained devices	1	-	-
5.3	AI/ML-driven wireless resource optimization	1	-	-
5.4	Ambient backscatter, energy harvesting wireless networks	2	-	-
5.5	Mm Wave & THz wireless for embedded systems	2	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of Practical Hours			**0**

Suggested Learning Resources:

Textbooks:

1. Andreas F. Molisch – *Wireless Communications* (2nd Edition, Wiley)

Reference Books:

1. William Stallings – *Wireless Communications & Networks* (2nd Edition, Pearson)
2. Matthew N.O. Sadiku & Sarhan M. Musa – *Wireless Sensor Networks* (CRC Press)
3. Afif Osseiran, José F. Monserrat, Patrick Marsch – *5G Mobile and Wireless Communications Technology* (Cambridge University Press)

Online Resources:

- https://nptel.ac.in/courses/117104099
- MIT OpenCourseWare – Principles of Wireless Communications https://ocw.mit.edu/courses/6-450-principles-of-digital-communication-i-fall-2006/

Course Code: MESC104D **Course: Data Analytics and Edge Computing**

Credits: 3 **L:T:P: – 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks:100**

Prerequisites if any	Nil
Learning objectives	1. Understand the fundamentals of data analytics and edge computing concepts, architectures, and their integration in real-world applications. 2. Apply data preprocessing, modeling, and visualization techniques for analyzing data at the edge. 3. Explore edge computing platforms and frameworks to enable distributed, low-latency, and resource-efficient computation. 4. Evaluate challenges in security, privacy, and scalability when deploying analytics on edge devices. 5. Investigate emerging applications and research trends in edge AI, IoT data processing, and cloud–edge collaboration.

Course Outcome (Course Skill Set)

At the end of the course the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Explain data analytics workflows and edge computing architectures for real-time data processing.	L2
CO2	Apply data cleaning, feature engineering, and visualization techniques to edge data sources.	L3
CO3	Design and deploy machine learning/inference models on edge platforms and IoT devices.	L3
CO4	Evaluate system performance in terms of latency, scalability, security, and energy efficiency at the edge.	L3
CO5	Demonstrate the ability to integrate cloud–edge analytics for applications such as smart cities, healthcare, and industrial IoT.	L4

CO-PO Mapping

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	2	2	1	2	1
CO2	2	2	2	1	2	1
CO3	2	2	2	1	2	1
CO4	2	2	2	1	2	1
CO5	2	2	2	1	2	1

Course Structure

Module-1 Fundamentals of Data Analytics & Edge Computing		***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	Introduction to Data Analytics: descriptive, diagnostic, predictive, prescriptive	1	-	-
1.2	Basics of Edge Computing: architecture, edge vs. cloud vs. fog	2	-	-
1.3	Edge hardware overview: gateways, micro data centers, IoT edge devices	1	-	-
1.4	Data lifecycle at the edge: acquisition, preprocessing, transmission	2	-	-
1.5	Role of AI/ML at the edge	2	-	-
Module-2 Data Acquisition, Processing & Storage at the Edge				
2.1	Edge data ingestion pipelines	1	-	-
2.2	Streaming vs. batch data processing	2	-	-
2.3	Data cleaning, transformation, and feature extraction at the edge	2	-	-
2.4	Edge storage: time-series databases (Influx DB), embedded databases (SQLite, Rocks DB)	2	-	-
2.5	Data compression & serialization formats (JSON, Protobuf, Avro)	1	-	-
Module-3 Machine Learning and AI for Edge Devices				
3.1	Lightweight ML frameworks: TensorFlow Lite, PyTorch Mobile, Edge Impulse	2	-	-
3.2	Model compression: quantization, pruning, knowledge distillation	2	-	-
3.3	On-device inference optimization	1	-	-
3.4	Real-time prediction and decision-making at the edge	2	-	-
3.5	Case studies: predictive maintenance, anomaly detection	1	-	-
Module-4 Edge Networking, Security & Privacy				
4.1	Edge network protocols: MQTT, CoAP, gRPC, 5G integration	1	-	-
4.2	Distributed data processing and federated learning at the edge	2	-	-
4.3	Security for edge devices: encryption, secure boot, trusted execution	2	-	-
4.4	Privacy-preserving analytics: differential privacy, homomorphic encryption	2	-	-
4.5	Regulatory compliance for edge analytics (GDPR, HIPAA)	1	-	-
Module-5 Applications & Emerging Trends in Data Analytics and Edge Computing				
5.1	Smart cities, Industry 4.0, autonomous vehicles, healthcare IoT	1	-	-
5.2	Integration with cloud and hybrid architectures	2	-	-
5.3	Edge-to-cloud orchestration and workload distribution	2	-	-
5.4	Edge AI accelerators: Google Coral, NVIDIA Jetson, Intel Movidius	2	-	-

5.5	Trends: 6G edge, quantum edge analytics	1	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of Practical Hours			**0**

Suggested Learning Resources:

Textbooks:

1. **Andrew Ng et al.** – *Machine Learning Yearning* (free online PDF)
2. **Rajkumar Buyya, Satish Narayana Srirama** – *Fog and Edge Computing: Principles and Paradigms*
3. **Jules S. Damji et al.** – *Learning Spark: Lightning-Fast Big Data Analysis*

References:

1. **Pete Warden, Daniel Situnayake** – *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*
2. **Ian Goodfellow, Yoshua Bengio, Aaron Courville** – *Deep Learning*

Web links and Video Lectures (e-Resources):

YouTube: "Introduction to Edge Computing" – IBM Technology

Coursera: *Introduction to Data Analytics* – IBM (link)

Coursera: *Introduction to Edge Computing* – University at Buffalo

YouTube: "IoT Data Processing at the Edge" – Microsoft Azure IoT

Coursera: *Data Wrangling, Analysis and AB Testing with SQL* – University of California Davis

Coursera: *Big Data Analysis with Scala and Spark* – École Polytechnique Fédérale de Lausanne

Professional Elective - II

Course Code: MESC105A **Course: Embedded System and Automation**

Credits: 3 **L:T:P: 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 Hrs **Max. Marks: 100**

Prerequisites if any	Basic Concept of Microprocessor
Learning objectives	1. To provide knowledge on building blocks of embedded system, input/output interfacing & Bus communication with processors. 2. To teach automation using scheduling algorithms and real time operating systems. 3. To discuss different phases & modeling of a new embedded product. 4. To involve discussions/ practice/exercise onto revising & familiarizing the concepts

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Understand the functionalities of processor internal blocks, with their requirement.	L3
CO2	Understand the role and features of RT operating system, that makes multitask execution possible by processors.	L3
CO3	Understand multiple CPU based on either hardcore or soft core helps data overhead management with processing- speed reduction for µC execution.	L3
CO4	To design embedded systems to perform dedicated function.	L4

Mapping with POs and PSOs:

COs	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	2	2	2	2
CO2	3	3	2	2	2	2
CO3	3	3	2	2	2	2
CO4	3	3	2	2	2	2

Mapping strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

	Module -1: Embedded System, Memory, Processor	***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	Introduction to Embedded Systems-built in features for embedded Target Architecture.	1	-	-
1.2	Selection of Embedded processor-DMA, memory devices.	1	-	-
1.3	Memory management methods-memory mapping, cache replacement policies-Timer and Counting devices, Watchdog Timer, Graphics.	2	-	-
1.4	Processing Unit (GPU)- multiprocessing and parallel processing, Real Time Clock	2	-	-
1.5	Software Development tools-IDE, assembler, compiler, linker, simulator, debugger, In circuit emulator, Target Hardware Debugging.	2	-	-
	Module-2: Embedded Networking			
2.1	Embedded Networking: Introduction, I/O Device Ports & Buses - multiple interrupts and interrupt service mechanism.	2	-	-
2.2	Serial Bus communication protocols - RS232 standard, RS485-USB-Inter Integrated Circuits (I2C).	2	-	-
2.3	CAN Bus-Wireless protocol based on Wifi, Bluetooth, Zigbee.	2	-	-
2.4	Introduction to Device Drivers-interfacing peripherals with processors.	1	-	-
2.5	Configuration of networking modules-timing and control signals.	1	-	-
	Module-3: Introduction to RTOS			
3.1	Introduction to basic concepts of RTOS-Need, Task, process & threads, interrupt routines in RTOS, Multiprocessing and Multitasking, Preemptive and non-preemptive scheduling.	2	-	-
3.2	Task communication-context switching, interrupt latency and deadline shared memory, message passing, Interposes Communication –synchronization between processes-semaphores.	2	-	-
3.3	Introduction, Edge Operating system Virtual machine, selection of operating system for edge computing.	2	-	-
3.4	Comparison of Windows and Linux based Edge operating systems.	1	-	-
3.5	Edge devices, Edge devices in IoT applications.	1	-	-
	Module-4: Modelling of Embedded System			
4.1	Modelling embedded systems, embedded software development approach, Overview of UML modeling with UML.	2	-	-

4.2	UML Diagrams, Hardware/Software Partitioning.	2	-	-
4.3	Co-Design Approaches for System Specification and modeling.	2	-	-
4.4	Synthesis-features comparing Single processor Architectures and Multi-Processor Architectures.	1	-	-
4.5	Design approach on parallelism in uniprocessors and Multiprocessors.	1	-	-
Module-5: Application Embedded for Automation				
5.1	Application development: Objective, Need, different Phases &Modelling of the EDLC.	2	-	-
5.2	Choice of Target Architectures for Embedded Application Development for for Control Dominated-Data Dominated Systems.	2	-	-
5.3	Case studies on Digital Camera, Adaptive Cruise control in a Car, Mobile Phone, automated robonoids.	2	-	-
5.4	Interface to sensors, GPS, GSM, Actuators.	2	-	-
	Total No of Teaching Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of Practical Hours			**0**

Text Books:

1. Tammy Noergaard, "Embedded System Architecture, A comprehensive Guide for Engineers and Programmers", 2nd Edition, Elsevier, 2013.
2. Peckol, "Embedded system Design", JohnWiley&Sons,2010.
3. Lyla B Das," Embedded Systems-An Integrated Approach", Pearson 2013.

Reference Books:

1. Elicia White, "Making Embedded Systems", O'Reilly Series, SPD, 2011.
2. Wolf Wayne Hendrix, Computers as Components: Principles of Embedded Computing System Design, 3rd Edition, Morgan Kaufmann, 2012.
3. Shibu K.V, "Introduction to Embedded Systems", Tata Mcgraw Hill,2009.
4. Raj Kamal, "Embedded System-Architecture, Programming, Design", Mc Graw Hill, 2013.

Subject code: MESC105B **Course: ADAPTIVE SIGNAL PROCESSING**

Credits: 03 **L: T: P: 3: 0 : 0**

CIE: 50% Marks **SEE: 50% Marks**

SEE Hrs: 3 Hrs. **Max. Marks: 100**

Prerequisites if any	Basic Concept of Linear Algebra, Probability and Statistics, Digital Signal Processing (DSP)
Learning objectives	1. Understanding the Need for Adaptivity 2. Model Random Signals and Environments 3. Design Adaptive Filters 4. Analyze Convergence and Stability

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Convert a sampled signal to a different sampling rate.	L3
CO2	Design an adaptive filter.	L3
CO3	Estimate spectral characteristics of signals.	L3
CO4	Explain the use of Wavelets and multiresolution.	L4
CO5	Apply ML algorithm for signal processing.	L3

Mapping with POs and PSOs:

COs	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	2	2	2	2
CO2	3	3	2	2	2	2
CO3	3	3	2	2	2	2
CO4	3	3	2	2	2	2
CO5	3	3	2	2	2	2

Mapping strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

	Module-1 Multirate signal processing	***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	Review of sampling, Nyquist rate,	2	-	-
1.2	aliasing, reconstruction, anti-aliasing filters.	2	-	-
1.3	Decimation and interpolation, Sampling rate conversion and implementation,	2	-	-
1.4	Digital filter banks, Sampling rate conversion for multistage and bandpass signals.	2	-	-
	Module-2 Adaptive filters			
2.1	Adaptive filter applications – System identification	2	-	-
2.2	Adaptive channel equalization, Echo cancellation,	2	-	-
2.3	Adaptive line enhancer, Adaptive noise cancellation,	2	-	-
2.4	LPC of speech signals, LMS and RLS algorithms	2	-	-
	Module-3 Power spectral estimation			
3.1	Estimation of power spectra, Non-parametric estimation – Barlet	2	-	-
3.2	Welch methods, characteristics of nonparametric power spectrum estimators	2	-	-
3.3	Parametric power spectrum estimation – Auto correlation model parameters,	2	-	-
3.4	Yule-walker method, unconstrained least squares, AR model parameters.	2	-	-
	Module -4 Wavelets			
4.1	Continuous time wavelets, CWT as operator,	2	-	-
4.2	Inverse CWT, DWT and vector subspaces,	2	-	-
4.3	MRA, Formal definition, scaling functions	2	-	-
4.4	Subspaces Wavelet basis for MRA.	2	-	-
	Module -5 Machine learning for Signal processing			
5.1	Supervised learning,	2	-	-
5.2	Classification models to predict class models,	2	-	-
5.3	regression models,	2	-	-
5.4	classifying iris species	2	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of No. of Practical Hours			0

Textbooks:

1. John G Proakis, Dimitris G Manolakis,**"Digital Signal Processing, Principles, Algorithms and applications"**,4th Edition, Pearson Education, 2007.
2. Raghuveer Rao and AjitBopardikar,**"Wavelet transforms – introduction to theory and applications"**, Pearson education/ Addison-Wesley, Delhi,2000.
3. Michael Beyeler, **"Machine learning for OpenCV"**, Packt, 2017.

Reference Books/Links:

1. Tarun Kumar Rawat **"Digital Signal Processing"**, Oxford university press, New Delhi, 2015.
2. Jaideva C Goswami and Andres K Chan, **"Fundamentals of Wavelets theory, algorithms and applications"**, Wiley India 2006.
3. Christopher M Bishop **"Pattern Recognition and Machine Learning"**, Springer, 2011.
4. https://www.deeplearningbook.org/
5. NPTEL Course: https://www.youtube.com/watch?v=HVGW85eGPQQ&list=PLyqSpQzTE6M_h5UgZWpybzBVDGmHGhQQb
6. https://www.microsoft.com/en-us/research/people/cmbishop/prml-book/
7. https://www.sp4comm.org/

Course Code: MESC105C **Course: Multimedia and Applications**

Credits: 3 **L:T:P – 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 Hrs **Max. Marks: 100**

Prerequisites if any	Signal Processing and communication
Learning objectives	1. Understand Multimedia Fundamentals 2. Explore Multimedia Systems and Architecture 3. Understand Multimedia Data Standards and Formats

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Develop an understanding of fundamental concepts in multimedia	L2
CO2	Learn about image data representation and various color models.	L3
CO3	Explore different types of video signals and the principles of digital audio.	L2
CO4	Understand multimedia data compression methods and audio compression standards.	L2
CO5	Gain knowledge of essential video compression techniques.	L2

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	2	1	3	1	1
CO2	3	2	2	3	1	1
CO3	3	2	1	3	1	1
CO4	3	2	1	3	1	1
CO5	3	2	1	3	1	1

Mapping strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

		No. of Lecture Hours	*No. of Tutorial Hours*	*No. of Practical Hours*
Module – 1 Introduction to multimedia				
1.1	What is Multimedia, Components of Multimedia System, Multimedia and Hypermedia, Multimedia Authoring metaphors.	4	-	-
1.2	Multimedia Production, Multimedia Presentation, Automatic Authoring.	4	-	-
Module – 2 Fundamental concepts in video				
2.1	Types of Video Signals, Analog Video, Digital Video, Basics of Digital Audio: Sound, Digitization of Sound.	4	-	-
2.2	Quantization and Transmission of Audio, Pulse code modulation, Differential coding of audio, Predictive coding.	4	-	-
Module – 3 Multimedia Data Compression				
3.1	Introduction, Basics of Information Theory, Lossless Compression Algorithms, Fix-Length Coding, Run- length coding, Dictionary- based coding.	4	-	-
3.2	Variable Length Coding, Huffman Coding Algorithm, Audio Compression standards: Introduction, Psychoacoustics model, MPEG Audio.	4	-	-
Module – 4 Basic Video Compression Techniques				
4.1	Introduction to Video compression, Video compression standard H.261	4	-	-
4.2	Video compression, standard MPEG-1, MPEG-4, MPEG-7.	4	-	-
Module – 5 Cloud Computing for Multimedia Services				
5.1	Cloud computing Overview, Multimedia Cloud Computing	4	-	-
5.2	Cloud Assisted Areas, Interactive Cloud gaming.	4	-	-
	Total No. of Lecture Hours	***40***		
	Total No. of Tutorial Hours		***0***	
	Total No. of Practical Hours			***0***

Textbooks:

1. Fundamentals of Multimedia by Ze-Nian Li & Mark S. Drew. Publisher: Prentice Hall

Reference Books:

1. An introduction to digital multimedia by Savage, T. M. and Vogel, K. E. 2008.
2. Digital Multimedia by Nigel Chapman & Jenny Chapman. 2009.

Online Resources: https://www.tutorialspoint.com/multimedia

Course Code: MESC105D **Course: Process Control**

Credits: 03 **L:T:P: 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 03 **Max. Marks: 100**

Prerequisites if any	Control systems
Learning objectives	1. To understand the basics of process control and modelling of various process systems. 2. To identify various process elements and understand the dynamic behavior of first and second order systems. 3. To understand and implement various PID controllers. 4. To implement tuning of the controllers using various methods and study about digital controllers. 5. To select advanced control strategy to enhance performance.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Understand process control principles, manufacturing process dynamics, and apply engineering knowledge for process analysis and modeling.	L2
CO2	Select and tune appropriate controllers, apply tuning rules, and implement digital controller techniques for optimum performance.	L3
CO3	Apply advanced control strategies to enhance system performance in industrial applications.	L3

Mapping with POs and PSOs:

COs	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	2	1	1	1
CO2	3	-	2	1	1	1
CO3	3	2	1	3	1	1

3 – Strong 2 – Medium 1 – Low

Course Structure

Module – 1		***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	Introduction: Introduction to Process Control. Control objectives, servo regulatory control, and classification of process variables.	2	-	-

1.2	Modeling of some Chemical Process Systems: Modeling basics, Degree of Freedom, Mass Balance, Energy Balance equations, linearization of nonlinear systems	3	-	-
1.3	Modeling of Level Tank System, Continuous Stirred Tank Heater, Continuous Stirred Tank Reactor, Transfer function.	3	-	-
Module – 2				
2.1	Elements of Process Control: Dead time, Interacting and non-interacting systems	2	-	-
2.2	self-regulation, inverse response, capacity of process, integrating systems, multi-capacity process.	2	-	-
2.3	Process Identification: Dynamic behavior of first and second order processes	2	-	-
2.4	Obtaining First Order Plus Time Delay (FOPTD) model with Process Reaction curve. Obtaining second order model of processes.	2	-	-
Module – 3				
3.1	Common Controller Modes: Controller Modes, ON OFF, Multi position, time proportional controller	2	-	-
3.2	Theory Proportional, Integral and Derivative modes, PI, PD, PID Controller	3	-	-
3.3	Electronics Controller implementation, Dynamic Behavior of closed loop systems with P, I, D, PI , PID modes.	3	-	-
Module – 4				
4.1	Discretization and Implementation Issues: Discrete time control mode realization.	2	-	-
4.2	Velocity and Position algorithm of PID control. Integral windup, anti-windup systems, controller bias, bumps less transfer	3	-	-
4.3	Tuning of Controllers: Application and tuning, ZN Tuning (Open loop and Closed loop), Performance criteria, Integral criteria.	3	-	-
Module – 5				
5.1	Some Advance Control Techniques: Cascade Control, Feed forward Control, ratio Control, Air Fuel Ratio Control for Drum Boilers.	4	-	-
5.2	Level Control in Drum Boiler, Shrinking and Swelling, Inverse response of Drum Boiler.	4	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of No. of Practical Hours			**0**

Suggested Learning Resources:

Text Books:

1. G. Stephanopolous, “Chemical Process Control An Introduction to Theory and Practice”, Prentice Hall India, August 2000.
2. Surekha Bhanot, “Process Control Principles and Applications”, Oxford, 2008

Reference Books:

1. C.D. Johnson, “Process Control Instrumentation Technology”, Prentice Hall India.
2. Thomas Marlin, “Process Control Designing Processes and Control for Dynamic Performance”, Tata MC Graw Hill, 2012.
3. F.G. Shinskey, “Process Control Systems Application Design and Adjustment” 3rd edition, McGraw Hill International.
4. D. E. Seborg, T.F. Edgar, D. A. Mellichamp, “Process Dynamics and Control”, Wiley, 2004.

Online Resources:

• https://nptel.ac.in/courses/103106148

Course Code: MESCL106A **Course: Computing Laboratory-1**

Credits: 2 **L:T:P: – 0:0:4**

CIE: 50% **SEE: 50%**

SEE Hours: 2 **Max. Marks:50**

Prerequisites if any	Digital Electronics, Computer Organization, Operating systems
Learning objectives	1. Able to implement the software on an hardware 2. Able to show the booting and configuring the OS for embedded systems 3. Analyse the computing architecture design constraints,

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	To apply the concepts of embedded systems architecture and its implementation.	L2
CO2	To show the effect of hardware and software for embedded systems.	L2
CO3	To analyze the pipeline effect in processor and cache effects	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	2	2	3	1	1
CO2	3	2	2	3	1	1
CO3	3	2	2	3	1	1

3 – Strong 2 – Medium 1 – Low

Course Structure

		No. of Lecture Hours	***No. of Tutorial Hours***	***No. of Practical Hours***
1	Examining the Effect of Cache Parameters and Program Factors on Cache Hit Rate using Gem 5	-	-	2
2.	Simulating an X86 Architecture in gem5	-	-	2
3.	Implementing the MIPS Processor with Pipelined Microarchitecture	-	-	2
4.	Implement the MIPS Single-Cycle Datapath and Controller	-	-	2
5.	Developing RISC V architecture using Gem5	-	-	2
6	Write a simple Linux character device driver (simulated via QEMU)	-	-	2
7	Develop an interrupt-driven UART echo application	-	-	2
8	Implement multitasking in FreeRTOS: LED blink task, sensor read task, UART logging task	-	-	2
9	Simulate IPC with named pipes and shared memory in QEMU Linux	-	-	2
10	Implement TCP client/server communication between two embedded boards (or QEMU instances)	-	-	2
	No. of Lecture Hours	**0**		
	No. of Tutorial Hours		***0***	
	No. of Practical Hours			**20**

Suggested Learning Resources:

Text Books:

1. gem5: Getting Started with gem5
2. https://www.cs.sfu.ca/~alaa/courses/cmpt450/fall2024/tutorials/gem5/index.html#analyze-trace-4-ooo--cache
3. GitHub - esattok/computer-organization: This repository contains laboratory assignments for CS 224 (Computer Organization) course taken at Bilkent University [2020-2021 Spring]
4. Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman – *Linux Device Drivers* (O'Reilly)
5. Chris Simmonds – *Mastering Embedded Linux Programming*

Course Code: MESC201 **Course: IoT Architecture and Protocol Systems**

Credits: 04 **L:T:P: – 3:0:2**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	Computer architecture and Computer Networks
Learning objectives	Knowledge on concepts of IoT applications and IoT architectures, Event driven analysis and security testing IoT systems

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Understand the overview of IoT and its applications.	L2
CO2	Describe the IoT reference architecture and different types of architecture	L2
CO3	Understand the various protocols and IoT based systems	L2

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	1	2	3	1	2
CO2	2	1	2	3	1	2
CO3	2	1	2	3	1	2

3 – Strong 2 – Medium 1 – Low

Course Structure

Module – 1		*No. of Lecture Hours*	*No. of Tutorial Hours*	*No. of Practical Hours*
1.1	Introduction to IOT,	2	-	-
1.2	Applications of IOT,	3	-	-
1.3	Use cases of IOT	3	-	-
Module – 2				
2.1	**The IoT Architectural Reference Model as Enabler, IoT in Practice:** Examples: IoT in Logistics and Health	2	-	-
2.2	**IoT Reference Model:** Domain, information, functional and communication models.	2	-	-
2.3	**Amazon Web Services for IoT:** Amazon EC2, Amazon Auto Scaling, Amazon S3, Amazon RDS,	2	-	-

2.4	Amazon DynamoDB, Amazon Kinesis, Amazon SQS, Amazon EMR, SkyNet IoT Messaging Platform.	2	-	-
Module – 3				
3.1	**IoT Reference Architecture:** Architecture, Functional, information,	2	-	-
3.2	deployment and operation views,	2	-	-
3.3	SOA based Architecture, API-based Architecture	2	-	-
3.4	OPEN IoT Architecture for IoT/Cloud Convergence	2	-	-
Module – 4				
4.1	**Application Protocols for IoT:** UPnP, CoAP,	2	-	-
4.2	MQTT, XMPP. SCADA, Web Socket, IP-based	2	-	-
4.3	Protocols: 6LoWPAN, RPL	2	-	-
4.4	Authentication Protocols; IEEE 802.15.4.	2	-	-
Module – 5				
5.1	**Case study:** Cloud-Based Smart-Facilities Management,	2	-	-
5.2	Healthcare,	3	-	-
5.3	Environment Monitoring System	3	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of Practical Hours			**0**

Suggested Learning Resources:

Text Books:

1. Bassi, Alessandro, et al, “Enabling things to talk”, Springer-Verlag Berlin An, 2016.
2. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, “IoT Fundamentals:Networking Technologies, Protocols, and Use Cases for the Internet of Things”, CISCO Press, 2017
3. Vijay Madisetti and Arshdeep Bagha, “Internet of Things (A Hands-on- Approach)”, 1st Edition, VPT, 2014.

Reference Books:

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

Online Resources:

1. **Internet of Things (IoT) – Batch 2 – NPTEL+**
2. **Practical IoT Concepts-Devices, IoT Protocols & Servers | Udemy**

Course Code: MESC202 **Course: Embedded Security**

Credits: 3 **L:T:P: 3-0-0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	1. Basics of Embedded Systems 2. C/C++ Programming for Microcontrollers 3. Fundamentals of Networking
Learning objectives	1. Understand the fundamentals of security in embedded and IoT systems. 2. Learn cryptographic methods suited for constrained devices. 3. Implement secure boot, firmware protection, and hardware security measures. 4. Design secure communication protocols and embedded OS security features. 5. Apply penetration testing and forensics techniques for embedded devices.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Describe embedded security threats, models, and mitigation techniques.	L2
CO2	Implement cryptographic algorithms optimized for embedded platforms.	L3
CO3	Configure secure boot, firmware encryption, and hardware protection features.	L4
CO4	Develop secure communication protocols and RTOS-based secure applications.	L3
CO5	Perform penetration testing, firmware analysis, and apply IoT security standards.	L4

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	2	1	2	1	1
CO2	3	2	3	3	2	1
CO3	3	2	3	3	2	2
CO4	2	2	2	2	1	1
CO5	3	3	3	3	2	2

3 – Strong 2 – Medium 1 – Low

Course Structure

	Module – 1 - Introduction to Embedded Security	***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	Introduction to embedded systems & IoT ecosystem	2	-	-
1.2	Threat models, attack surfaces, and security-by-design principles	3	-	-
1.2	Case studies: Stuxnet, Jeep Hack, Mirai Botnet.	3	-	-
	Module – 2 Cryptography for Embedded Systems			
2.1	Symmetric cryptography	4	-	-
2.2	Public key cryptography	2	-	-
2.3	Lightweight cryptography	1	-	-
2.4	Hash functions, MAC, RNG in constrained devices	1	-	-
	Module – 3 Secure Boot, Firmware & Hardware Security			
3.1	Secure boot principles & chain of trust	2	-	-
3.2	Bootloader security, firmware signing & OTA update security	2	-	-
3.3	Hardware root of trust, secure elements, TPM	2	-	-
3.4	JTAG/SWD protection, tamper detection, side-channel attack prevention	2	-	-
	Module – 4 Secure Communication & Embedded OS Security			
4.1	Secure communication protocols	2	-	-
4.2	Authentication mechanisms (OAuth 2.0, mutual TLS)	2	-	-
4.3	Replay attack prevention & sequence number techniques	1	-	-
4.4	RTOS security: MPU/MMU protection, secure IPC	1	-	-
4.5	Secure coding practices & compiler hardening	2	-	-
	Module – 5 Penetration Testing, Forensics & Standards			
5.1	IoT penetration testing methodology	2	-	-
5.2	Firmware extraction & reverse engineering	2	-	-
5.3	IoT security standards (ETSI EN 303 645, NIST 8259, ISO 27001)	2	-	-
5.4	Secure Development Lifecycle (SDL) & legal compliance (GDPR, HIPAA)	2	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of Practical Hours			**0**

Suggested Learning Resources:

Text Books:

1. Practical IoT Hacking: The Definitive Guide to Attacking the Internet of Things – Fotios Chantzis et al., Wiley, 2021.
2. Engineering Secure Devices: A Practical Guide for Embedded System Architects and Developers, Dominik Merli.
3. Hardware Hacking Handbook, Breaking Embedded security with hardware attacks, Jasper Van

Woundenberg and colin.

4. Embedded Systems Security David Kleidermacher and Mike Kleidermacher, Elsevier
5. Security in Embedded Devices, Gebotys, Catherine H., Springer
6. Practical Embedded Security, Stapko T., Elsevier/Newnes

References:

1. ETSI EN 303 645 IoT Security Standard
2. NIST SP 8259 IoT Cybersecurity Guidelines
3. FreeRTOS and Zephyr security documentation

Online Resources:

1. https://www.etsi.org/technologies/consumer-iot-security
2. https://csrc.nist.gov/publications/detail/sp/8259/final
3. https://freertos.org/security
4. https://zephyrproject.org/security

Course Code: MESC203 **Course: Machine Learning and Deep Learning**

Credits: 3 **L:T:P 3-0-0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 Hrs **Max. Marks: 100**

Prerequisites if any	nil
Learning objectives	1. To understand the fundamental concepts of machine learning and its applications 2. To master the concepts of classification and clustering techniques. 3. To develop a deep understanding of convolutional neural networks (CNNs) and their architecture. 4. To apply deep learning techniques to large-scale datasets and real-world problems

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	**Demonstrate** a comprehensive understanding of machine learning and deep learning fundamentals and their applications.	L2
CO2	**Apply** various machine learning algorithms and deep learning architectures to solve complex problems.	L3
CO3	**Develop** and implement machine learning models using appropriate programming languages and tools.	L4

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	2	2	1	2	1
CO2	2	2	2	1	2	1
CO3	2	2	2	1	2	1

3 – Strong 2 – Medium 1 – Low

Course Structure

Module 1 – Introduction to Machine Learning		*No. of Lecture Hours*	*No. of Tutorial Hours*	*No. of Practical Hours*
1.1	Introduction, Training, Rote Learning, Learning Concepts,.	1	-	-
1.2	General-to-Specific Ordering, Version Spaces, Candidate	1	-	-

	Elimination, Inductive Bias,			
1.3	Decision-Tree Induction, The Problem of Overfitting,	1	-	-
1.4	The Nearest Neighbor Algorithm,	2	-	-
1.5	Learning Neural Networks, Supervised Learning,	2	-	-
1.6	Unsupervised Learning, Reinforcement Learning	2	-	-
Module 2 – Neural Networks				
2.1	Introduction, Neurons, Perceptrons,	2	-	-
2.2	Multilayer Neural Networks,	2	-	-
2.3	Recurrent Networks,.	2	-	-
2.4	Unsupervised Learning Networks, Evolving Neural Networks	2	-	-
Module 3 – Convolution Neural Networks				
3.1	The operation, Pooling, Convolution and Pooling as an infinitely strong prior,	2	-	-
3.2	Variants of the basic functions, efficient algorithms,	2	-	-
3.3	Random or Unsupervised Features,	2	-	-
3.4	Neuroscientific Basis for Convolutional Networks.	2	-	-
Module 4 – Recurrent Neural Networks				
4.1	RNN, Bidirectional RNN, Encoder-Decoder Sequence to sequence architecture	2	-	-
4.2	Deep Recurrent Networks, Recursive Neural Networks,	2	-	-
4.3	The Long Short Term Memory and other Gated RNNs,	2	-	-
4.4	Optimization for Long Term Dependencies	2	-	-
Module 5 - Applications				
5.1	Large-Scale Deep Learning,	2	-	-
5.2	Computer Vision,	2	-	-
5.3	Speech Recognition,	2	-	-
5.4	Natural Language Processing, Other Applications.	2	-	-
	Total No. of Teaching Hours	40		
	Total No. of Tutorial Hours		0	
	Total No. of Practical Hours			0

Suggested Learning Resources:

Text Books:

1. Artificial Intelligence Illuminated - Ben Coppin
2. Deep Learning - Ian Goodfellow, Yoshua Bengio, Aaron Courville
3. Fundamentals of Deep Learning – Nikhil Budama

Reference books:

1. Neural Networks and Deep Learning – Charu Aggarwal
2. Hands-on Deep Learning Algorithms with Python – Sudharsan Ravichandran

Skill Development Activities Suggested

- Individual or group projects to apply learned concepts to real-world problems.
- Regular coding assignments to reinforce theoretical concepts.
- Experimentation with different libraries and frameworks (e.g., TensorFlow, PyTorch, Scikit-learn).
- Guest lectures from industry experts to provide insights into current trends.

Course Code: MESC204 **Course: Hardware and Software Co-Design**

Credits: 3 **L:T:P: – 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks:100**

Prerequisites if any	Digital Electronics
Learning objectives	1. Understanding the relationship between hardware and software in digital systems design. 2. Learning to design systems that meet system-level objectives through concurrent design. 3. Gaining knowledge on modeling and architecture design in hardware-software co-design. 4. Developing skills in partitioning system requirements into hardware and software components. 5. Applying principles of dependability and real-time constraints in embedded systems design.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Acquire knowledge on various models of Co-design.	L2
CO2	Explore the interrelationship between Hardware and software in a embedded system	L3
CO3	Acquire the knowledge of firmware development process and tools during Co design.	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	1	2	3	2	1
CO2	2	1	2	3	2	1
CO3	2	1	1	2	2	1

3 – Strong 2 – Medium 1 – Low

Course Structure

Module – 1 - Co-Design Issues		*No. of Lecture Hours*	*No. of Tutorial Hours*	*No. of Practical Hours*
1.1	Co- Design Models, Architectures, Languages, A Generic Co-design Methodology.	3	-	-
1.2	Co-Synthesis Algorithms: Hardware software synthesis algorithms: hardware – software partitioning distributed system co-synthesis.	5	-	-
Module – 2 - Prototyping and Emulation				
2.1	Prototyping and emulation techniques, prototyping and emulation environments,	1	-	-

2.2	future developments in emulation and prototyping architecture specialization techniques, system communication infrastructure.	2	-	-
2.3	Target Architectures: Architecture Specialization techniques, System Communication infrastructure,	1	-	-
2.4	Target Architecture and Application System classes, Architecture for control dominated systems (8051- Architectures for High performance control),	2	-	-
2.5	Architecture for Data dominated systems (ADSP21060, TMS320C60), Mixed Systems.	2	-	-
Module – 3 – Compilation Techniques and Tools for Embedded Processor Architectures:				
3.1	Modern embedded architectures,	2	-	-
3.2	embedded software development needs,	2	-	-
3.3	compilation technologies, practical consideration in a compiler development environment.	4	-	-
Module – 4 – Design Specification and Verification:				
4.1	Design, co-design, the co-design computational model,	1	-	-
4.2	concurrency coordinating concurrent computations	2	-	-
4.3	interfacing components, design verification,	2	-	-
4.4	implementation verification, verification tools, and interface verification.	3	-	-
Module – 5 – Languages for System – Level Specification and Design				
5.1	Languages for System – Level Specification and Design-I: System – level specification, design representation for system level synthesis,	2	-	-
5.2	system level specification languages.	2	-	-
5.3	Languages for System – Level Specification and Design-II: Heterogeneous Specifications and multi- language co-simulation,	2	-	-
5.4	the cosyma system and lycos system.	2	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of Practical Hours			**0**

Suggested Learning Resources:

Text Books:

1. Hardware / Software Co- Design Principles and Practice – Jorgen Staunstrup, Wayne Wolf –Springer, 2009.

2. Kai Hwang and F.A.Briggs,‖Computer architecture and parallel processor‖‘ Mc Graw Hill, N.Y, 1999

Reference Books:

1. Hardware / Software Co- Design - Giovanni De Micheli, Mariagiovanna Sami, Kluwer Academic Publishers, 2002.
2. A Practical Introduction to Hardware/Software Co-design -Patrick R. Schaumont, Springer, 2010

Online resources:

https://www.bing.com/videos/riverview/relatedvideo?&q=hardware+software+codesign+for+embedded+ai&&mid=F2AAC179162B663

Professional Elective III

Course Code: MESC205A **Course: Multicore and RISC - V Architecture**

Credits: 3 **L:T:P: 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks:100**

Prerequisites if any	1. C/Assembly programming 2. Computer architecture/OS fundamentals
Learning objectives	Students will learn 1. Limitation of Single core and CISC architecture 2. Port threaded code to multicore and RISC architecture

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Identify performance related parameters in the field of Computer Architecture and need for multicore architecture	L2
CO2	Apply the concept of multithreading and OPENMP to parallelize a process	L3
CO3	Develop programs and simulation tools for the RISC architecture	L3
C04	Utilize RISC-V simulation tools to write, debug, and test RISC-V assembly code	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	1	2	1	2	3
CO2	2	3	1	1	3	2
CO3	2	3	1	1	3	2
CO4	3	1	2	1	2	3

3 – Strong 2 – Medium 1 – Low

Course Structure

Module – 1 – Introduction to Multicore		*No. of Lecture Hours*	*No. of Tutorial Hours*	*No. of Practical Hours*
1.1	Motivation For Concurrency in Software, Parallel Computing Platforms, Parallel Computing In Microprocessors	2	-	-
1.2	Differentiating Multi-Core Architectures from Hyper Threading	2	-	-

	Technology, Multi-Threading on Single-Core Versus Multi-Core Platforms, Understanding Performance, Amdahl's Law			
1.3	Defining Threads, System View of Threads, Threading Above The Operating System, Threads Inside The OS, Threads Inside The Hardware, What Happens When A Thread Is Created	2	-	-
1.4	Application Programming Models and Threading, Virtual Environment: VMs And Platforms, Runtime Virtualization	2	-	-
Module – 2 - Threads and Synchronization				
2.1	Challenges, Parallel Programming Patterns, Motivating Problem	2	-	-
2.2	Synchronization, Critical Sections, Deadlock, Synchronization Primitives, Semaphores, Locks, Messages, Flow Control- Based Concepts, Fence, Barrier, Implementation-Dependent Threading Features	3	-	-
2.3	Creating Threads, Managing Threads, Thread Pools, Thread Synchronization, POSIX Threads	2	-	-
2.4	Signaling, Compilation and Linking	1	-	-
Module – 3 – OpenMP				
3.1	Challenges In Threading a Loop, Loop-Carried Dependence, Data-Race Conditions, Managing Shared and Private Data, Loop Scheduling And Portioning	2	-	-
3.2	Effective Use of Reductions, Minimizing Threading Overhead, Work-Sharing Sections, Performance-Oriented Programming.	2	-	-
3.3	Using Barrier and No Wait, Interleaving Single-Thread And Multi-Thread Execution, Data Copy-In and Copy-Out, Protecting Updates Of Shared Variables	2	-	-
3.4	Intel Task Queuing Extension to OpenMP, OpenMP Library Functions, OpenMP Compilation, Debugging, Performance	2	-	-
Module – 4 – RISC-V Introduction				
4.1	Motivation, Proprietary and open Instruction set	1	-	-
4.2	RISC-V Architecture - Register set, modes, formats of instructions, extensions	1	-	-
4.3	Types of Instructions – Encoding of immediate and register instruction, Instruction specifications, decoding instructions	2	-	-
4.4	Instruction types – Arithmetic, logic, Signed, unsigned operations	2	-	-
4.5	Flow control instructions – Conditional and un-conditional branches, branch penalty, building conditional logic ang branches	2	-	-
Module – 5 – Programming RISC-V				
5.1	Instructions for memory operations – load, store, array operations, byte, word, double word etc. Memory alignment and addressing techniques, Accessing arrays	2	-	-

5.2	RISC Assembly and high-level programming – Programming techniques, simulation, debugging techniques	2	-	-
5.3	Calling functions – calling conventions, Exploring Stack frame, Stack for JAL and JALR, Recursive function	2	-	-
5.4	RISC – V simulation tools (like - gcc tool chain, QEMU) building, cross compiling and simulation	2	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of Practical Hours			**0**

Suggested Learning Resources:

Text Books:

1. Shameem Akhter and Jason Roberts, "Multicore Programming, Increased Performance through Software Multithreading", Intel Press, 2006
2. David Patterson And Andrew Waterman, "The RISC-V Reader: An Open Architecture Atlas", Strawberry Canyon 2017, ISBN-13 : 978-0999249116

Reference Books:

1. Michael J. Quinn, "Parallel Programming In C With Mpi And Openmp", McGrawHill
2. David Harris and Sarah L. Harris, "Digital Design and Computer Architecture, RISC-V", Morgan Kaufmann, 2021, □ ISBN-13 : 978-0128200643
3. Sarful Hassan, "RISC-V Programming Guide: Getting Started with the RISC-V Microcontroller and C/C++", 2024, ISBN: 979-8300419486

Online Resources:

1. https://www.youtube.com/playlist?list=PLwdnzlV3ogoU0TR333JyxG8T3HDg52S0h
2. https://github.com/TheThirdOne/rars
3. https://kvakil.github.io/venus/
4. http://www.riscvbook.com/greencard-20181213.pdf

Course Code: MESC205B **Course: Cryptography and Network Security**

Credits: 3 **L:T:P: 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks:100**

Prerequisites if any	Nil
Learning objectives	1. Study the network security model, security attacks, mechanisms and services and demonstrate use of various symmetric key ciphers and their principles. 2. Understand the concept of Modular Arithmetic and its application in public key cryptography and apply the knowledge to solve security related problems. 3. Understand the design principles of public key cryptosystems for encryption, key exchange and authentication. 4. Comprehend the concept of secured electronic transaction with web security considerations. 5. Study the security threats to networks and their counter measures

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Identify and describe different techniques in modern cryptography	L2
CO2	Employ modular arithmetic fundamentals to cryptography	L4
CO3	Describe, recognize and use the principles of public key cryptosystems for various applications.	L4
C04	Recognize the use of cryptography in Data Networks	L4
CO5	Analyse the security issues related to internet and networks	L5

Mapping with POs and PSOs:

COs	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**	**PSO3**
CO1	2	2	2	1	2	1
CO2	2	2	2	1	2	1
CO3	2	2	2	1	2	1
CO4	2	2	2	1	2	1
CO5	2	2	2	1	2	1

3 – Strong 2 – Medium 1 – Low

Course Structure

Module-1 Information and Network Security Concepts		***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	Cybersecurity, Information security,	1	-	-
1.2	Network Security, OSI Survey Architecture,	2	-	-
1.3	Security Attacks & Security Services,	2	-	-
1.4	Cryptography,	1	-	-
1.5	Trust and Trustworthiness, Standards	2	-	-
Module-2 Number Theory				
2.1	Divisibility and Division Algorithms,	2	-	-
2.2	The Euclidean Algorithms, Modular Arithmetic,	2	-	-
2.3	Prime Numbers, Fermat's and Euler's theorems,.	2	-	-
2.4	Testing for Primality, The Chinese Remainder theorem, Discrete Logarithms	2	-	-
Module-3 Symmetric Ciphers				
3.1	Classical Encryption Techniques,	2	-	-
3.2	Block Ciphers and Data Encryption Standard,	2	-	-
3.3	Finite Fields, Advanced Encryption Standards,	2	-	-
3.4	Block Cipher Operation, Random Bit Generation and stream Ciphers	2	-	-
Module-4 Cryptographic Data Integrity Algorithms				
4.1	Cryptographic Hash Function,	2	-	-
4.2	Message Authentication Codes, Digital Signatures,	2	-	-
4.3	Lightweight Cryptography and Post-Quantum Cryptography.	2	-	-
4.4	Mutual Trust: Cryptography Key Management and Distribution, User Authentication	2	-	-
Module-5 Network and Internet Security				
5.1	Transport-Level Security, Wireless Network Security,	2	-	-
5.2	Electronic Mail Security, IP Security,	2	-	-
5.3	Network Endpoint Security, Cloud Security,	2	-	-
5.4	Internet of Things (IoT) Security	2	-	-
Total No. of Lecture Hours		**40**		
Total No. of Tutorial Hours			**0**	
Total No. of No. of Practical Hours				**0**

Suggested Learning Resources:

Text Books

1. William Stallings, "Cryptography and Network Security Principles and Practice", Pearson Education Inc., eBook ISBN 13: 978-1-292-43749-1, 8th Edition, 2023 and onwards.
2. Behrouz A. Fourouzan, "Introduction to Cryptography and Network Security" Tata McGraw-Hill, ISBN 978–0– 07–287022–0, 1st Edition, 2008 and onwards.
3. Atul Kahate," Cryptography and Network security", Tata McGraw-Hill, ISBN-13: 978-0-07-064823-4, 2nd Edition, 2008 and onwards.
4. H. Yang et al., Security in Mobile Ad Hoc Networks: Challenges and Solution, IEEE Wireless Communications, 2004, https://escholarship.org/uc/item/5p89k583

Web links and Video Lectures (e-Resources):

1. https://nptel.ac.in/

Course Code: MESC205C **Course: High Performance Computing**

Credits: 3 **L:T:P: – 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 Hr **Max. Marks:100**

Prerequisites if any	Digital Electronics and Computer Architecture
Learning objectives	1. To Study various computing technology architecture. 2. To know Emerging trends in computing technology. 3. To highlight the advantage of deploying computing technology.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Illustrate the key factors affecting performance of CSE applications	L3
CO2	Illustrate mapping of applications to high-performance computing systems	L3
CO3	Apply hardware/software co-design for achieving performance on real-world applications	L2

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	2	1	3	2	1
CO2	2	2	1	3	1	1
CO3	2	2	1	3	1	1

3 – Strong 2 – Medium 1 – Low

Course Structure

	Module – 1 - Introduction to Parallel Computing	*No. of Lecture Hours*	*No. of Tutorial Hours*	*No. of Practical Hours*
1.1	**Introduction to Parallel Computing:** Motivating Parallelism, Scope of Parallel Computing	2	-	-
1.2	**Parallel Programming Platforms:** Implicit Parallelism: Trends in Microprocessor Architectures	2	-	-
1.3	Limitations of Memory System Performance, Dichotomy of Parallel Computing Platforms, Physical Organization of Parallel Platforms,	2	-	-
1.4	Communication Costs in Parallel Machines, Routing Mechanisms for Interconnection Networks, Impact of Process-Processor Mapping and Mapping Techniques.	2	-	-

Module – 2 - Principles of Parallel Algorithm Design				
2.1	**Principles of Parallel Algorithm Design:** Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing	2	-	-
2.2	Methods for Containing Interaction Overheads, Parallel Algorithm Models.	2	-	-
2.3	**Basic Communication Operations:** One-to-All Broadcast and All-to-One Reduction, Allto-All Broadcast and Reduction, All-Reduce and Prefix-Sum Operations	2	-	-
2.4	Scatter and Gather, All-to-All Personalized Communication, Circular Shift, Improving the Speed of Some Communication Operations	2	-	-
Module – 3 – Analytical Modelling of Parallel Programs				
3.1	**Analytical Modelling of Parallel Programs:** Sources of Overhead in Parallel Programs, Performance Metrics for Parallel Systems, The Effect of Granularity on Performance, Scalability of Parallel Systems.	2	-	-
3.2	Minimum Execution Time and Minimum Cost-Optimal Execution Time, Asymptotic Analysis of Parallel Programs. Other Scalability Metrics,	2	-	-
3.3	**Programming Using the Message-Passing Paradigm:** Principles of Message-Passing Programming, The Building Blocks: Send and Receive Operations, MPI: the Message Passing Interface	2	-	-
3.4	Topologies and Embedding, Overlapping Communication with Computation, Collective Communication and Computation Operations, Groups and Communicators	2	-	-
Module – 4 – Programming Shared Address Space Platforms				
4.1	**Programming Shared Address Space Platforms:** Thread Basics, Why Threads?, The POSIX Thread API, Thread Basics: Creation and Termination, Synchronization Primitives in Pthreads, Controlling Thread and Synchronization Attributes,	2	-	-
4.2	Thread Cancellation, Composite Synchronization Constructs, Tips for Designing Asynchronous Programs,	2	-	-
4.3	**OpenMP:** a Standard for Directive Based Parallel Programming Dense Matrix Algorithms: Matrix-Vector Multiplication, Matrix-Matrix Multiplication, Solving a System of Linear Equations	2	-	-
4.4	**Sorting:** Issues in Sorting on Parallel Computers, Sorting Networks, Bubble Sort and its Variants, Quicksort, Bucket and Sample Sort.	2	-	-
Module – 5 – Graph Algorithms				
5.1	Definitions and Representation, Minimum Spanning Tree: Prim's Algorithm, Single-Source Shortest Paths: Dijkstra's Algorithm	2	-	-

5.2	All-Pairs Shortest Paths, Transitive Closure, Connected Components, Algorithms for Sparse Graphs,	2	-	-
5.3	Search Algorithms for Discrete Optimization Problems: Definitions and Examples, Sequential Search Algorithms,	2	-	-
5.4	Search Overhead Factor, Parallel Depth-First Search, Parallel Best-First Search, Speedup, Anomalies in Parallel Search Algorithms	2	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of Practical Hours			**0**

Suggested Learning Resources:

Text Books:

1. Introduction to Parallel Computing, AnanthGrama, Anshul Gupta, George Karypis, and Vipin Kumar, 2nd edition, Addison-Welsey, 2003.

Reference Books:

1. Grama, A. Gupta, G. Karypis, V. Kumar, An Introduction to Parallel Computing, Design and Analysis of Algorithms: 2/e, Addison-Wesley, 2003.
2. G.E. Karniadakis, R.M. Kirby II, Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and their Implementation, Cambridge University Press,2003.
3. Wilkinson and M. Allen, Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, 2/E, Prentice Hall, 2005.
4. M.J. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill, 2004.
5. G.S. Almasi and A. Gottlieb, Highly Parallel Computing, 2/E, Addison-Wesley, 1994.
6. David Culler Jaswinder Pal Singh,"Parallel Computer Architecture: A hardware/Software Approach", Morgan Kaufmann, 1999.
7. Kai Hwang, "Scalable Parallel Computing", McGraw Hill 1998.

Online resources:

High Performance Computing for Scientists and Engineers - Course

Course Code: MESC205D **Course: FPGA based System Design**

Credits: 3 **L:T:P: – 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks:100**

Prerequisites if any	Digital Electronics and Circuit Theory
Learning objectives	1. To learn the different types of programming elements, programmable logic blocks, programmable input-output blocks and programmable interconnects of various types of FPGAs 2. To understand the steps involved in synthesis, simulation, and testing of systems *3.* To design and implement circuits, subsystem and system using FPGA and I/O boards

Course Outcomes:

On the successful completion of the course, the student will be able to

COs	Course Outcomes	Bloom's level
CO1	Understand the fundamental principles and practices in reconfigurable architecture.	L2
CO2	Simulate and synthesize the reconfigurable computing architectures.	L3
CO3	Understand the FPGA design principles, applications and logic synthesis	L2

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	3	2	3	2	2
CO2	3	2	3	3	2	2
CO3	3	2	3	3	2	2

Mapping Strength: **Strong– 3** **Medium – 2** **Low – 1**

Course Structure

Module – 1- Introduction		***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	History, Reconfigurable vs Processor based system, RC Architecture. Reconfigurable Logic Devices: Field Programmable Gate Array, Coarse Grained Reconfigurable Arrays.	3	-	-
1.2	Reconfigurable Computing System: Parallel Processing on Reconfigurable Computers, A survey of Reconfigurable Computing System.	5	-	-
Module-2- Languages, Compilation and Implementation				
2.1	Design Cycle, Languages, HDL, High Level Compilation,	4	-	-

2.2	Low level Design flow, Integration, FPGA Design flow, Logic Synthesis	4	-	-
Module-3- Synthesis, Placement and on-line communication				
3.1	High Level Synthesis for Reconfigurable Devices: Modelling, Temporal Partitioning Algorithms,	4	-	-
3.2	TEMPORAL PLACEMENT: Offline and Online Temporal Placement, Managing the Device's Free Space with Empty Rectangles, Managing the Device's Occupied Space, NoC, Dynamic NoC.	4	-	-
Module 4: Partial Reconfiguration Design				
4.1	Partial Reconfiguration Design, Bitstream Manipulation with JBits, The modular Design flow,	4	-	-
4.2	The Early Access Design Flow, Creating Partially Reconfigurable Designs, Partial Reconfiguration using Hansel-C Designs, Platform Design.	4	-	-
Module 5: FPGA Platform				
5.1	Components, Adding to platform FPGA systems, assembling custom compute cores. Software Design-System Software Options, Root File system, Cross-Development Tools, Monitors and Boot-loader.	4	-	-
5.2	Components, Adding to platform FPGA systems, assembling custom compute cores. Software Design-System Software Options, Root File system, Cross-Development Tools, Monitors and Boot-loader.	4	-	-
Total No. of Theory Ho		40		
Total No. of Tutorial Hours			**0**	**-**
Total No. of Practical Hours				**0**

Text Books:

1. Reconfigurable Computing: Accelerating Computation with Field-Programmable Gate Arrays, M. Gokhale and P. Graham, Springer, ISBN: 978-0-387-26105-8, 2005.
2. Introduction to Reconfigurable Computing: Architectures, Algorithms and Applications, C. Bobda, Springer, ISBN: 978-1-4020-6088-5, 2007
3. Ron Sass, Andrew G Schmidt Embedded Systems Design with Platform FPGAs Principles and Practices, 2011, First Edition, Tata McGraw Hill, India.

Reference Books:

1. Practical FPGA Programming in C D. Pellerin and S. Thibault Prentice-Hall 2005
2. FPGA Based System Design, W. Wolf, Prentice-Hall, 2004
3. Rapid System Prototyping with FPGAs: Accelerating the Design Process, R. Cofer and B. Harding, Newnes, 2005

Online Resources:

nptel.ac.in/courses/117108040

PROFESSIONAL ELECTIVE - IV

Course Code: MESC206A **Course: System on Chip**

Credits: 3 **L:T:P: – 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 hrs **Max. Marks: 100**

Prerequisites if any	Digital Electronics and Computer Architecture
Learning objectives	1. Understand SoC fundamentals, packaging approaches (SoC, SoB, SiP), IP reuse, and the main architectural components such as processors, memories, and interfaces. 2. Demonstrate the SoC design flow, including system modeling, hardware–software partitioning, RTL design, synthesis, physical design, and analyze interconnect challenges with NoC. 3. Apply low-power and thermal-aware design techniques, and validate SoC functionality using SystemVerilog simulation and FPGA prototyping.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Explain SoC fundamentals, compare SoC with SoB and SiP, and describe IP reuse, 3D packaging, chiplets, processors, memories, and interfaces.	Understand
CO2	Demonstrate SoC design methodology, including modeling, partitioning, RTL, synthesis, physical design, and analyze interconnect issues with NoC.	Apply
CO3	Apply low-power and thermal-aware design, and validate SoCs using SystemVerilog simulation and FPGA prototyping.	Analyze

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	1	2	3	2	1
CO2	2	1	2	3	2	1
CO3	2	1	2	3	2	1

3 – Strong 2 – Medium 1 – Low

Course Structure

	Module – 1	***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	SoC overview, and how is it different from SoB and SiP? Historical perspective of ICs. Advantages and challenges of SoC, applications, Concept of IP, IP reuse/integration. SoC Platforms, 3-D Packaging, Chiplets	2	-	-
1.2	Hardware Accelerators. I/O - interfaces, communication protocols (UART, SPI, I2C, USB)	2	-	-
1.3	Hardware Accelerators. I/O - interfaces, communication protocols (Ethernet, AMBA, AHB, AXI, and OCP).	2	-	-
1.4	Clock distribution, Multiple Clock Domains, CDC, RDC	2	-	-
	Module – 2			
2.1	High-Level Design - System modeling, functional description.	3	-	-
2.2	SoC Life Cycle - Specification, Hardware-Software Partition (brief),	2	-	-
2.3	RTL Design and Verification, synthesis, layout, validation, manufacturing, packaging and test.	3	-	-
	Module – 3			
3.1	Floor planning of IP blocks on an SoC, Placement and Routing.	2	-	-
3.2	Interconnect problems in SoC - routing difficulties, delay and power issues.	2	-	-
3.3	Network on Chip, Signal Integrity issues - crosstalk and IR-drop.	2	-	-
	Module – 4			
4.1	Power density and thermal issues in SoC. Power Estimation and power reduction techniques in SoCs.	3	-	-
4.2	Multiple clock domains and power domains.	2	-	-
4.3	Clock gating, power gating, and dynamic voltage/frequency scaling (DVFS). Thermal modeling and management in multi-core SoCs.	3	-	-
	Module – 5			
5.1	System Verilog for Functional Verification.	2	-	-
5.2	Test benches, simulation, and assertion-based verification. Static Timing Analysis	2	-	-
5.3	System-level Testing.	2	-	-
5.4	FPGA based SoC Design and Implementation with an example such as Image Processing, Signal Processing	2	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of No. of Practical Hours			**0**

Suggested Learning Resources:

Text Books:

1. Fundamentals of System-on-Chip Design on Arm Cortex-M Microcontrollers, by René Beuchat, Florian Depraz, Sahand Kashani, Andrea Guerrieri, August 2021, ISBN 978-1-911531-33-3 T2.
2. Joseph Yiu. System on Chip Design with ARM Cortex-M Processors, ISBN – 978-1-911531-18-0

Reference Books:

1. SystemVerilog for Verification: A guide to learning the testbench language features , Chris Spear Synopsys, inc. springer, Hardcover ISBN 978-1-4614-0714-0, 14 February 2012.\
2. Architecting and Building High-Speed SoCs: Design, develop, and debug complex FPGA-based systems-on-chip, 9 December 2022, Mounir Maaref.
3. Vivado Design Suite User Guide: Getting Started (UG910) available online

Online Resources:

Introduction to SoC Design Course - SoC Architecture – Arm®

Course Code: MESC206B **Course: Industrial IOT Sensor And Interfacing**

Credits: 3 **L:T:P: – 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks:100**

Prerequisites if any	Prerequisite Analog Circuit
Learning objectives	1. To provide in depth knowledge in physical principles applied in sensing, measurement and a comprehensive understanding on how measurement systems are designed, calibrated, characterized, and analysed. 2. To introduce the students to various kind of sensor such as thermal sensor, radiation sensors and smart sensor and their sensing mechanisms and provide in depth understanding of the principle of measurement, and theory of instruments and sensors for measuring parameter such as radiation intensity, variation in temperature. 3. To give a fundamental knowledge on the how radiation and thermal parameter converted into electrical parameter. 4. Also to give fundamental knowledge of actuator and how it converts sensor parameter such as radiation intensity into electrical parameter such as current and voltage.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Illustrate the working principles of different types of sensors and actuators.	L3
CO2	Analyze the phenomena that define behaviour of various sensors and actuators.	L3
CO3	Apply the concepts in common methods for converting a physical parameter into an electrical quantity.	L3
CO4	Identify suitable sensors and actuator for real time applications.	L4

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	3	2	3	2	1
CO2	3	3	2	3	2	1
CO3	3	3	2	3	2	1
CO4	3	3	2	3	2	1

Mapping strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

	Module -1: Sensor and Transducer	***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	Sensors/Transducers, Principles, Classification, Characterization.,	3	-	-
1.2	Mechanical and Electromechanical Sensors: Introduction, Resistive Potentiometer, Inductive Sensors.	2	-	-
1.3	Capacitive Sensors-Parallel plate & serrated plate types, Ultrasonic Sensors.	3	-	-
	Module-2: Thermal Sensor			
2.1	Thermal Sensors: Introduction, Helium Low Temperature Thermometer, Nuclear Thermometer	3	-	-
2.2	Magnetic Thermometer, Junction Semiconductor Types, Magnetic Sensors	3	-	-
2.3	Introduction, Sensors and the Principles Behind, Force & displacement Sensors.	2	-	-
	Module-3: Radiation Sensor			
3.1	Radiation Sensors: Introduction-Basic Characteristics	2	-	-
3.2	Types of Photoresistor /Photo detectors	2	-	-
3.3	X-ray and Nuclear Radiation Sensors	2	-	-
3.4	Fiber Optic Sensors	2	-	-
	Module-4: Smart Sensor			
4.1	Smart Sensors: Introduction, Primary Sensors, Excitation, Amplification, Filters, Converters.	2	-	-
4.2	Information Coding/Processing, Data Communication.	2	-	-
4.3	Standards for Smart Sensor Interface and Automation.	2	-	-
4.4	Sensors Applications: Introduction, On-board Automobile Sensors (Automotive Sensors), Home Appliance Sensors.	1	-	-
4.5	Medical Diagnostic Sensors, Sensors for Manufacturing, Sensors for environmental Monitoring.	1	-	-
	Module 5			
5.1	Actuators: Pneumatic and Hydraulic Actuation Systems, Valves, Rotary actuators.	2	-	-
5.2	Choice of Target Architectures for Embedded Application Development for for Control Dominated-Data Dominated Systems.	2	-	-
5.3	Mechanical Actuation Systems	2	-	-
5.4	Electrical Actuation Systems	2	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of Practical Hours			**0**

Text Books:

1. D. Patranabis-Sensors and Transducers, PHI Learning Private Limited.
2. W. Bolton-Mechatronics, Pearson Education Limited.

Reference Books:

1. Patranabis-Sensors and Actuators- 2nd Ed., PHI, 2013.
2. Robert H. Bishop-The Mechatronics Handbook, 2nd Ed., Mechatronic Systems, Sensors and Actuators, fundamentals and modelling.

Online resource

1. https://nptel.ac.in/content/syllabus_pdf/108108147.pdf

Course Code: MESC206C **Course: RTL Synthesis and Simulation**

Credits: 3 **L:T:P: – 3:0:0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks:100**

Prerequisites if any	Digital Electronics, Verilog
Learning objectives	Provide an overview of Design Principles of Embedded Systems. Understanding about the role of ASIC design flow and RTL source codes.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Understand proven design methodologies based on standard EDA tools	L2
CO2	Design combinational devices with full set of EDA tools.	L3
CO3	Design, Analyze and Verify the synchronous digital design on FPGAs	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	1	1	3	2	1
CO2	2	1	2	2	1	1
CO3	2	1	1	3	2	1

3 – Strong 2 – Medium 1 – Low

Course Structure

Module – 1 -		*No. of Lecture Hours*	*No. of Tutorial Hours*	*No. of Practical Hours*
1.1	Top down approach	1	-	-
1.2	Hardware modelling of combinational and sequential circuits with verilog HDL	3	-	-
1.3	writing a test bench	4	-	-
Module – 2				
2.1	Design of finite state machines (Synchronous and asynchronous),	2	-	-
2.2	system design using ASM chart,	2	-	-
2.3	Static Timing analysis, Meta-stability, clock issues,	2	-	-

2.4	Need and design strategies for Multi-clock Domain designs.	2	-	-
Module – 3				
3.1	Data path and Control path design	2	-	-
3.2	Arithmetic implementation strategies for data path design,	2	-	-
3.3	Processor Design,	2	-	-
3.4	Micro-programmed control design, Single cycle MMIPS	2	-	-
Module – 4				
4.1	Programmable Logic Devices: Fine grained and coarse-grained FPGA , Xilinx series	2	-	-
4.2	IP and Prototyping.	2	-	-
4.3	Simulation. Testbenches and debugging.	2	-	-
4.4	Synthesis flow. Synthesis to Standard cells and FPGA.	2	-	-
Module – 5				
5.1	ASIC Design flow- Introduction to ASIC Design Flow, SOC,	1	-	-
5.2	Floor planning, Placement,	2	-	-
5.3	Clock tree synthesis, Routing, Physical verification,	2	-	-
5.4	Power analysis, Design for performance, Low power VLSI design techniques, Technology Challenges.	3	-	-
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**0**	
	Total No. of No. of Practical Hours			**0**

Suggested Learning Resources:

Textbooks:

1. Stephen Brown and ZvonkoVranesic, “Fundamentals of Digital logic with Verilog Design”, Mc-GrawHill, 3rd edition
2. . David Harris and Sarah Harris, “Digital Design and Computer Architecture”, Morgan Kaufmann, 2nd edition, 2012
3. Douglas Smith, “HDL Chip Design: A Practical Guide for Designing, Synthesizing & Simulating ASICs & FPGAs Using VHDL or Verilog”, Doone Publications.

Reference Books:

1. Doug Amos, Austin Lesea, Rene Richter, “FPGA based prototyping methodology manual”, Xilinx
2. . IEEE standard HDL based on Verilog HDL, published by IEEE.
3. Ben Cohen, “Real Chip design and Verification using Verilog and VHDL”, Vhdl Cohen Publishing, 2002

Online resources:

VSD - RTL Synthesis Q&A Webinar | Udemy

http://www.bing.com/videos/riverview/relatedvideo?q=online+courses+on+RTL+Synthesis+and+Simulation&mid=B3077D6F4C9F117F14

Course Code: MESC206D **Course: Automotive embedded product development**

Credits: 3 **L:T:P: 3-0-0**

CIE: 50% **SEE: 50%**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	nil
Learning objectives	1. To introduce the functional domains, standards, and protocols in automotive embedded systems. 2. To familiarize students with AUTOSAR architecture, communication systems, and safety certification. 3. To explore networking technologies like CAN, Flex Ray, and middleware for automotive applications. 4. To analyse dependability issues and trends toward autonomous vehicle systems.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Explain the functional domains, standards, and certification issues of automotive embedded systems.	L2
CO2	Analyze AUTOSAR architecture, methodology, and conformance processes.	L3
CO3	Evaluate in-vehicle network protocols such as CAN, Flex Ray, and middleware.	L4
CO4	Apply concepts of dependability, diagnostics, and safety-critical system design.	L3
CO5	Assess emerging technologies including sensor fusion, ADAS, and autonomous systems.	L4

Mapping with POs and PSOs:

COs	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	1	2	1	1
CO2	3	2	3	3	2	1
CO3	2	2	2	3	2	2
CO4	2	2	2	2	1	1
CO5	3	3	3	3	2	2

3 – Strong 2 – Medium 1 – Low

Course Structure

Module – 1		***No. of Lecture Hours***	***No. of Tutorial Hours***	***No. of Practical Hours***
1.1	Introduction to automotive embedded systems, functional domains	2	-	-
1.2	Standardized components, models, and processes (In-vehicle networks, OS, middleware, architecture description languages)	3	-	-
1.2	Certification issues of safety-critical in-vehicle systems	3	-	-
Module – 2				
2.1	Setting up AUTOSAR: objectives, shortcomings in former software structures, working methods	3	-	-
2.2	AUTOSAR architecture and layered software approach	2	-	-
2.3	Standardization areas: BSW, RTE, conformance classes	2	-	-
2.4	AUTOSAR methodology, templates, ECU configuration, and integration	1	-	-
Module – 3				
3.1	Sensor technologies, sensor fusion, wireless networks	2	-	-
3.2	Intelligent control applications and driver assistance systems	2	-	-
3.3	Dependability issues: fail-safe systems, auto-diagnostics	2	-	-
3.4	Fully autonomous car concepts and deployment	2	-	-
Module – 4				
4.1	Event vs. time-triggered communication	2	-	-
4.2	FlexRay protocol, architecture, and applications	1	-	-
4.3	CAN bus fundamentals and limitations	1	-	-
4.4	Dependable CAN architectures: TTCAN, FlexCAN, FTT-CAN	2	-	-
4.5	Middleware and open issues in automotive networking	2	-	-
Module – 5				
5.1	Characteristics of automotive product lines	2	-	-
5.2	Feature modeling and variability management	2	-	-
5.3	Global coordination of product-line variability	2	-	-
5.4	Future networking trends: V2X, optimized architectures	2	-	-
Total No. of Lecture Hours		**40**		
Total No. of Tutorial Hours			**0**	
Total No. of No. of Practical Hours				**0**

Suggested Learning Resources:

Text Books:

1. Nicolas Navet, Francoise Simonot-Lion, Automotive Embedded Systems Handbook, CRC Press, 2008.
2. Oliver Scheickl et al., AUTOSAR – A Worldwide Standard is on the Road, SAE International, 2019.

References:

1. William Ribbens, Understanding Automotive Electronics, Elsevier, 2017.

Online Resources:

1. AUTOSAR Official Website
2. ISO 26262 Functional Safety Overview

Course Code: MESCL207 **Course: Computing Laboratory-2**

Credits: 2 **L:T:P: – 0:0:4**

CIE: 50% **SEE: 50%**

SEE Hours: 04 **Max. Marks:50**

Prerequisites if any	Digital Electronics, Computer Organization
Learning objectives	1. To know the implementation of different learning techniques on hardware 2. To know the implementation on Zynq-7 programming techniques

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Implement and apply machine learning techniques in prediction problems.	L2
CO2	Implement suitable learning algorithms to solve a given problem.	L3
CO3	Implement the available SDK on Zynq-7	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	1	1	2	1	1
CO2	2	1	1	2	1	1
CO3	3	1	2	3	2	2

3 – Strong 2 – Medium 1 – Low

Course Structure

		No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
1.	Implement multivariate linear regression.	-	-	2
2.	Implementing Decision tree Classification.	-	-	2
3.	Implement K-means clustering algorithm.	-	-	2
4.	Implementation of Natural Language Processing.	-	-	2
5.	Implementation of Speech Recognition.	-	-	2
6	Implement the combinational circuits on FPGA	-	-	2
7	Implement the Sequential circuits on FPGA	-	-	2

8	Implement the available SDK on Zynq-7 and carry out the cross compilation in FPGA environment	-	-	2
9	Analyse the hardware security algorithm	-	-	2
10	Implement the hardware security on ZYNQ-7	-	-	2
	No. of Lecture Hours	**0**		
	No. of Tutorial Hours		**0**	
	No. of Practical Hours			**20**

Reference Materials:

1. FPGA Prototyping by VHDL Examples: Xilinx MicroBlaze MCS SoC by Pong P.Chu, 2nd Edition, Willey
2. https://docs.amd.com/r/en-US/ug585-zynq-7000-SoC-TRM

ABILITY / SKILL ENHANCEMENT COURSES

Course Code: MESCL258A **Course: Software Tools for Embedded system**

Credits: 1 **L:T:P 0-0-2**

CIE: 50% **SEE: 0%**

SEE Hours: 0 **Max. Marks:50**

Prerequisites if any	NIL
Learning objectives	1. Introduce embedded development toolchains and IDEs. 2. Develop skills in hardware and software debugging. 3. Utilize simulation, profiling, and optimization tools for embedded firmware. 4. Apply scripting languages (Python, Bash) to automate embedded workflows. 5. Manage embedded projects using version control and CI/CD pipelines.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Explain various embedded software tools and their purposes.	L2
CO2	Configure and use compilers, linkers, and build systems.	L3
CO3	Perform debugging, simulation, and optimization of embedded firmware.	L4
CO4	Write and integrate automation scripts for testing and deployment.	L4
CO5	Apply project management and version control techniques for embedded development.	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	2	3	2	1	1
CO2	3	2	3	3	2	1
CO3	3	3	3	3	2	2
CO4	2	2	3	2	2	1
CO5	3	3	3	3	2	2

3 – Strong 2 – Medium 1 – Low

Course Structure

		No. of Lecture Hours	**No. of Tutorial Hours**	**No. of Practical Hours**
1.	Configure cross-compilation for ARM Cortex-M and build sample firmware.	-	-	4
2.	Debug firmware using GDB and OpenOCD.	-	-	4
3.	Simulate embedded board using QEMU and test I/O.	-	-	3
4.	Write Python script to automate firmware flashing via serial interface.	-	-	3
5.	Use Bash script for automated build and deployment.	-	-	3
6	Integrate Git-based CI pipeline for embedded firmware testing.	-	-	3
No. of Lecture Hours		**0**		
No. of Tutorial Hours			**0**	
No. of Practical Hours				**20**

Course Code: MESCL258B **Course: Optimization Techniques**

Credits: 1 **L: T:P: 0-0-2**

CIE: 50% **SEE: 0%**

SEE Hours: 2 **Max. Marks: 50**

Prerequisites if any	Communication Systems and Digital Signal Processing
Learning objectives	1. Understand different optimization techniques and their applications in communication systems. 2. Use MATLAB to model and solve optimization problems. 3. Connect and operate an SDR platform with MATLAB for experiments. 4. Apply optimization methods to improve SDR system performance. 5. Evaluate results using performance metrics like BER, SNR, and throughput.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Configure and operate SDR hardware with MATLAB	L3
CO2	Implement single-objective and multi-objective optimization methods	L4
CO3	Acquire and preprocess real-time SDR data, then analyse it using optimization-driven models	L4
CO4	Apply optimization techniques for different wireless applications	L3
CO5	Evaluate and compare the performance of optimized SDR systems under various constraints	L5

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	1	3	1	3	1
CO2	3	1	3	1	3	2
CO3	3	1	3	1	2	2
CO4	3	1	3	1	3	2
CO5	3	1	3	1	2	2

3 – Strong 2 – Medium 1 – Low

Course Structure

		No. of Lecture Hours	**No. of Tutorial Hours**	**No. of Practical Hours**
1.	Familiarization with MATLAB optimization toolbox and SDR hardware setup.	-	-	2
2.	Configure Adalm Pluto for BPSK/QPSK transmission, receive, and visualize spectrum. Tune parameters (gain, bandwidth) to minimize BER.	-	-	2
3.	Implement a modulation switching scheme based on channel SNR and perform transmission and reception using adalm pluto SDR. Use optimization to select the best modulation scheme in real time.	-	-	2
4.	Linear Programming for Resource Allocation: Optimize subcarrier power allocation in an OFDM simulation model	-	-	2
5.	Constrained Nonlinear Optimization: Minimize BER in a QAM system by adjusting modulation parameters with constraints	-	-	2
6	Genetic Algorithms for Parameter Estimation: Optimize antenna array beamforming weights to maximize SNR for a received signal.	-	-	2
7	Spectrum Sensing: Use PSO to optimize detection threshold in an energy detector for cognitive radio spectrum sensing.	-	-	2
8	Convex Optimization for Power Control: Optimize transmit power allocation for multi-user SDR downlink to maximize throughput under interference limits.	-	-	2
9	Adaptive Beamforming via Optimization: Combine adaptive algorithms with constrained optimization to steer nulls towards interference sources.	-	-	2
10	Multi-Objective Optimization: Trade-off between power consumption and BER in a simulated SDR link, generate 5G NR signals (OFDM, MIMO) in MATLAB, transmit using Mantiswave private 5G Network box. Optimize resource block allocation to maximize throughput.	-	-	2
	No. of Lecture Hours	**0**		
	No. of Tutorial Hours		**0**	
	No. of Practical Hours			**20**

Suggested Learning Resources:

Text Books:

1. D.P. Bertsekas, Nonlinear Programming, Athena Scientific, 3rd ed., 2016.
2. Matlab and Simulink for Engineers, Agam Kumar Tyagi, Oxford University Press, 2020.
3. Alexander M. Wyglinski, Di Pu, Travis F. Collins, David M. Whalen, Practical Software-Defined Radio with MATLAB and Simulink and the RTL-SDR, Artech House, 2017

Reference Books:

1. S. Boyd et al., Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares, Cambridge University Press, 2018.

Online Resources:

1. **https://www.mathworks.com/help/optim/**
2. **https://www.mathworks.com/help/gads/**
3. **https://www.mathworks.com/hardware-support.html**
4. **https://www.mathworks.com/matlabcentral/fileexchange/**

Course Code: MLCSL258C **Course: Python Programming**

Credits: 1 **L:T:P: – 0:0:2**

CIE: 50% **SEE: 0%**

SEE Hours: 0 **Max. Marks:50**

Prerequisites if any	Communication Systems and Digital Signal Processing
Learning objectives	1. **Develop and apply Python programming skills** including syntax, control structures, data types, and modular coding techniques. 2. **Manipulate and process data** using Python's built-in structures, file handling, regular expressions, and libraries such as NumPy, Pandas, and Matplotlib. 3. **Implement robust, reusable, and structured code** using functions, modules, and object-oriented programming principles. 4. **Design and test small-scale applications** that solve real-world problems in academic, research, or industry contexts.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	**Develop proficiency in Python syntax, data structures, and core programming constructs** to enable writing clean, efficient, and reusable code for solving computational problems.	L2
CO2	**Apply Python's standard libraries and third-party packages** (such as NumPy, Pandas, and Matplotlib) to perform data manipulation, analysis, visualization, and automation tasks.	L3
CO3	**Implement structured, modular, and object-oriented programming techniques** to design, develop, and document small- to medium-scale applications suitable for academic, research, and industrial contexts.	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	1	1	2	1	1	1
CO2	1	1	2	1	1	1
CO3	1	1	2	1	1	1

3 – Strong 2 – Medium 1 – Low

Course Structure

		No. of Lecture Hours	**No. of Tutorial Hours**	**No. of Practical Hours**
1.	Write a Python program which accepts the radius of a circle from the user and compute the area.	-	-	2
2.	Write a Python program which accepts the user's first and last name and print them in reverse order with a space between them.	-	-	2
3.	Write a Python program to solve (x + y) * (x + y). Take values of X and Y from user	-	-	2
4.	Write a Python program to sum of three given integers. However, if two values are equal sum will be zero.	-	-	2
5.	Write a Python program to convert seconds today, hour, minutes and seconds.	-	-	2
6	Python program to check whether the given number is even or not.	-	-	2
7.	Develop a program to read the student details like Name, USN, and Marks in three subjects. Display the student details, total marks and percentage with suitable messages.	-	-	2
8.	Develop a program to read the name and year of birth of a person. Display whether the person is a senior citizen or not.	-	-	2
9.	Method Overriding (Polymorphism). i. Define a base class Shape with a method area(), ii. Derive classes Circle, Rectangle, and Triangle. iii. Override the area() method in each class to compute the correct area. iv. Create a list of shapes and iterate through them, calling area() (demonstrating polymorphism).	-	-	2
10.	Multilevel Inheritance i. Create a base class Person with attributes name and age. ii. Create a derived class Employee that inherits from Person and adds employee_id. iii. Create another derived class Manager that inherits from Employee and adds department. iv. Write a method display() in Manager that prints all details.	-	-	2
	No. of Lecture Hours	**0**		
	No. of Tutorial Hours		**0**	
	No. of Practical Hours			**20**

Suggested Learning Resources:

Text Books

1. Python Crash Course: A Hand: A Hands-On, Project-Based Introduction to Programming by Eric Matthes, 2nd Edition
2. Learning Python – Mark Lutz

Reference Books

1. Fluent Python – Luciano Ramalho
2. Automate the Boring Stuff with Python – Al Sweigart
3. Python Cookbook – David Beazley & Brian K. Jones

Web links and Video Lectures (e-Resources):

YouTube

1. Corey Schafer – Python Programming Tutorials (playlist)
2. Programming with Mosh – Python for Beginners (link)
3. FreeCodeCamp.org – Python Full Course for Beginners (link)