**INSTITUTE OF ENGINEERING**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**PROGRAM: M.TECH. (VLSI & EMBEDDED SYSTEMS)**

**COURSE OUTCOMES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S No** | **Class & Semester** | **Course & Course Code** | **COs** | **Course Outcomes** |
|  | M.Tech. I Semester | MME6101T - RTL Simulation and Synthesis with PLDs | CO1 | Demonstrate a solid understanding of the fundamental concepts of digital logic design, including combinational and sequential circuits, and the role of PLDs in digital system design. |
| CO2 | Develop proficiency in writing and understanding hardware description languages (VHDL or Verilog) to describe the behavior and structure of digital circuits, both at the RTL (Register Transfer Level) and gate level. |
| CO3 | Perform RTL simulations using simulation tools like ModelSim, XSIM, or other simulation environments. Understand simulation testbenches, stimulus generation, and debugging techniques to verify and validate the functionality of designs. |
| CO4 | Design digital circuits (combinational, sequential, and state machines) using RTL and debug them efficiently using simulation tools. Learn to detect and correct logical errors, timing issues, and resource conflicts. |
| CO5 | Understand and apply the synthesis process to convert RTL designs into gate-level descriptions. Use synthesis tools (like Xilinx ISE, Vivado, or Altera Quartus) to generate optimized netlists for implementation in PLDs. |
|  | M.Tech I Semester | MME6102T - Microcontrollers and Programmable Digital Signal Processors | CO1 | Understand the basic architecture, features, and operation of microcontrollers and programmable DSPs. Learn about the different types of microcontrollers (e.g., 8051, ARM, PIC) and DSPs (e.g., Texas Instruments, Analog Devices), their instruction sets, and internal components (ALU, timers, memory). |
| CO2 | Develop proficiency in writing efficient assembly and high-level language (C) programs for microcontrollers and DSPs. Understand how to interface with peripheral devices such as sensors, actuators, and communication modules (e.g., UART, SPI, I2C). |
| CO3 | Design embedded systems using microcontrollers and DSPs for real-time signal processing applications. Implement systems like motor control, sensor data acquisition, and communication protocols. |
| CO4 | Implement digital signal processing algorithms on DSPs, such as filtering, Fourier Transform, convolution, and discrete-time signal processing. Learn how DSPs accelerate computation for real-time applications in audio, video, and communications. |
| CO5 | Interface microcontrollers with various external devices like sensors (e.g., temperature, pressure), displays (LCD, LEDs), actuators, and communication peripherals (e.g., Bluetooth, Wi-Fi, ZigBee) for embedded system applications. |
|  | M.Tech I Semester | MME6103T - Digital Signal and Image Processing  (Programme Elective-I) | CO1 | Understand the basic concepts of signals and systems, including continuous and discrete-time signals, linear time-invariant (LTI) systems, and their properties. Apply mathematical tools like Fourier Transform, Z-Transform, and Laplace Transform to analyze and manipulate digital signals. |
| CO2 | Analyze and process discrete-time signals using techniques like sampling, quantization, and filtering. Apply algorithms such as Fast Fourier Transform (FFT), Discrete Fourier Transform (DFT), and Digital Filtering for signal enhancement and manipulation. |
| CO3 | Design and implement digital filters (FIR and IIR filters) for various applications. Analyze filter characteristics (frequency response, stability, etc.) and design filters for tasks like noise reduction, signal enhancement, and data smoothing. |
| CO4 | Apply various transform techniques (Fourier Transform, Discrete Cosine Transform (DCT), Wavelet Transform) to analyze and process signals. Use these techniques in applications like signal compression and feature extraction. |
| CO5 | Understand the representation of images in digital form and perform basic image processing operations such as enhancement (contrast adjustment, filtering), transformation, and noise reduction using common algorithms and techniques. |
|  | M.Tech I Semester | MME6106T - Programming Languages for Embedded Software  (Programme Elective-I) | CO1 | Understand the fundamentals of embedded systems and their characteristics, including hardware-software interaction, resource constraints (memory, power), and real-time requirements. Develop basic knowledge of embedded system design and software requirements. |
| CO2 | Develop proficiency in programming embedded systems using C. This includes writing efficient and optimized code for microcontrollers, handling hardware interrupts, and utilizing low-level peripherals such as timers, GPIOs, ADCs, and communication interfaces. |
| CO3 | Understand and apply Assembly language programming for embedded systems, including writing assembly code for performance-critical sections, direct memory access (DMA), and low-level hardware interactions. |
| CO4 | Develop embedded software using C++, including object-oriented concepts such as classes, inheritance, and polymorphism. Learn to apply these concepts to manage system complexity and improve the maintainability of embedded applications. |
| CO5 | Design and develop real-time embedded software for systems with stringent timing constraints. Understand real-time operating systems (RTOS) principles and use them in programming languages for tasks like scheduling, task prioritization, and inter-task communication. |
|  | M.Tech I Semester | MLC6101T - VLSI signal processing  (Programme Elective-I) | CO1 | Understand the basic concepts of VLSI technology and its relevance in digital signal processing. Learn about different types of VLSI technologies (CMOS, BiCMOS), scaling, fabrication processes, and their impact on the performance of signal processing systems. |
| CO2 | Develop a strong foundation in digital signal processing algorithms, including FIR and IIR filtering, FFT, and other signal transformation techniques. Understand how these algorithms are the basis for many VLSI signal processing applications. |
| CO3 | Learn how to implement digital signal processing algorithms on hardware using VLSI techniques. Focus on techniques for mapping algorithms to hardware (e.g., FPGA, ASIC) and consider hardware/software co-design approaches. |
| CO4 | Design and implement efficient VLSI-based systems for signal processing tasks such as filtering, transformation, and compression. Apply techniques such as pipelining, parallelism, and resource sharing to optimize designs. |
| CO5 | Apply optimization techniques in the design of VLSI signal processing systems. These include optimizing for speed, area, power, and cost while ensuring the system meets real-time performance requirements. |
|  | M.Tech I Semester | MME6101P - Parallel Processing (Programme Elective-II) | CO1 | Understand the fundamental concepts of parallel processing, including parallel architectures, parallel algorithms, and the distinction between parallelism and concurrency. Learn about Amdahl’s Law, speedup, and how parallelism affects performance. |
| CO2 | Study different parallel computing models such as shared memory, distributed memory, and hybrid models. Understand the communication mechanisms involved in parallel processing, including message passing and shared memory access. |
| CO3 | Analyze and evaluate different parallel computing architectures, including multi-core processors, distributed systems, and graphics processing units (GPUs). Understand the hardware and software components that enable parallel execution of tasks. |
| CO4 | Gain proficiency in parallel programming using languages and tools such as OpenMP, MPI (Message Passing Interface), CUDA, and OpenCL. Learn how to write parallel programs for multi-core and multi-processor systems. |
| CO5 | Apply techniques for data decomposition, task partitioning, and parallelization of algorithms to optimize performance. Learn how to divide computational tasks into smaller sub-tasks and efficiently distribute them across multiple processors or cores. |
|  | M.Tech I Semester | MME6102P - System Design with Embedded Linux  (Programme Elective-II) | CO1 | Understand the fundamentals of Embedded Linux, including its architecture, components, and how it differs from desktop Linux. Learn about the kernel, bootloaders, root filesystem, and various Linux components that are relevant to embedded systems. |
| CO2 | Gain practical skills in setting up an embedded Linux development environment, including installing and configuring cross-compilers, debugging tools, and using virtual machines or real embedded hardware platforms like Raspberry Pi or BeagleBone. |
| CO3 | Learn how to configure and customize the Linux kernel for embedded applications. Understand how to modify kernel configurations, compile the kernel for embedded platforms, and optimize it for specific hardware requirements and performance. |
| CO4 | Understand the creation, management, and optimization of root filesystems for embedded Linux systems. Learn how to use tools like BusyBox, U-Boot, and Yocto to build and configure embedded Linux root filesystems efficiently. |
| CO5 | Develop skills to write device drivers in Linux for embedded systems, such as drivers for GPIOs, UARTs, SPI, I2C, and other peripherals. Understand hardware abstraction layers (HAL) and how to interface with hardware devices using Linux. |
|  | M.Tech I Semester | MTE6103T - CAD of Digital System (Programme Elective-II) | CO1 | Understand the fundamental concepts of digital system design, including combinational and sequential circuits, and how these concepts are applied to real-world systems. Learn how to represent digital systems using Boolean algebra, logic gates, and finite state machines (FSMs). |
| CO2 | Develop proficiency in using CAD tools for the design and simulation of digital systems. Learn how to use popular design software like Xilinx ISE, Cadence, ModelSim, Quartus, and other tools for synthesizing and verifying digital circuits. |
| CO3 | Gain expertise in using Hardware Description Languages (HDL) such as VHDL and Verilog to design digital systems. Learn how to model, simulate, and synthesize digital circuits using these languages for both combinational and sequential logic designs. |
| CO4 | Learn to simulate and verify digital designs using CAD tools. Understand how to use testbenches, waveform viewers, and simulation platforms to ensure that digital circuits function correctly before hardware implementation. |
| CO5 | Understand the process of synthesis in digital system design, where high-level HDL descriptions are transformed into gate-level circuits. Learn to optimize designs for performance, area, and power using synthesis tools in CAD environments. |
|  | M.Tech I Semester | MTE6104T - RTL Simulation and Synthesis with PLDs (Lab) | CO1 | Gain a strong understanding of RTL design concepts for digital circuits, including register-transfer level abstraction, data-path design, and control-path design. Learn how RTL represents digital systems and how it serves as an intermediate step between high-level algorithm specifications and hardware implementation. |
| CO2 | Develop proficiency in using HDLs like VHDL or Verilog for modeling and describing digital circuits at the RTL level. Learn to write code to represent combinational and sequential circuits, finite state machines (FSMs), and data flow. |
| CO3 | Learn how to simulate and verify RTL designs using simulation tools such as ModelSim or Vivado Simulator. Perform functional verification of the designs to ensure correctness before synthesis. Understand how to create testbenches for simulating various scenarios and checking timing and functionality. |
| CO4 | Understand the synthesis process for digital designs targeted at PLDs such as FPGAs and CPLDs. Learn how to use synthesis tools like Xilinx ISE, Quartus, or Vivado to convert RTL descriptions into gate-level representations suitable for PLD implementation. |
| CO5 | Gain practical experience in implementing digital designs on FPGAs/CPLDs. Learn how to map synthesized designs onto the PLD hardware, configure the device, and perform hardware verification to ensure that the design functions as expected on actual hardware. |
|  | M.Tech I Semester | MTE6105T - Microcontrollers and Programmable Digital Signal Processors (Lab) | CO1 | Understand the architecture and functionality of microcontrollers and programmable DSPs. Learn about the core components such as ALU, timers, I/O ports, interrupt handling, memory (RAM, Flash), and their application in embedded system designs. |
| CO2 | Develop proficiency in programming microcontrollers using C and Assembly language. Understand the use of integrated development environments (IDEs) such as Keil for microcontroller programming and debugging, as well as hardware interfaces like serial communication, GPIOs, and PWM. |
| CO3 | Implement real-time digital signal processing (DSP) algorithms on programmable DSP processors. Learn how to apply DSP techniques like filtering, FFT (Fast Fourier Transform), convolution, and decimation on processors like TI C6000 series, Analog Devices SHARC, or ARM-based DSP systems. |
| CO4 | Gain hands-on experience in interfacing microcontrollers with external peripherals such as sensors, actuators, displays, and keypads. Learn to handle analog-to-digital (ADC) and digital-to-analog (DAC) conversions, PWM generation, and serial communication (UART, SPI, I2C). |
| CO5 | Learn how to configure and use interrupts in microcontrollers for real-time event handling and control applications. Develop the ability to write interrupt-driven programs for precise timing and system responses to external stimuli. |
|  | M.Tech I Semester | MTE6106T - Research Methodology and IPR | CO1 | Define research problems and formulate research objectives using appropriate methodologies. |
| CO2 | Apply ethical principles and intellectual property rights (IPR) regulations in research practices. |
| CO3 | Analyze research literature and synthesize findings to support research hypotheses. |
| CO4 | Design research experiments and methodologies suitable for mechanical engineering applications. |
| CO5 | Communicate research findings effectively through technical reports and presentations. |
|  | M.Tech I/ II Semester | AUD6101T - Disaster Management (Audit Course – I & II) | CO1 | Explain the fundamental concepts and phases of disaster management. |
| CO2 | Analyze the causes and impacts of natural and man-made disasters. |
| CO3 | Evaluate disaster risk reduction strategies and emergency response plans. |
| CO4 | Design community-based disaster management frameworks for resilience. |
| CO5 | Demonstrate the ability to develop disaster preparedness and mitigation plans. |
|  | M.Tech I/ II Semester | AUD6102T - English for Research Paper Writing (Audit Course – I & II) | CO1 | Identify the structure and components of a high-quality research paper. |
| CO2 | Apply academic writing techniques to draft clear and concise research papers. |
| CO3 | Evaluate research articles to identify strengths and areas for improvement. |
| CO4 | Demonstrate proficiency in referencing, citation styles, and avoiding plagiarism. |
| CO5 | Develop effective communication skills for presenting research findings. |
|  | M.Tech I/ II Semester | AUD6103T - Sanskrit for Technical Knowledge (Audit Course – I & II) | CO1 | Understand the basics of Sanskrit grammar and technical terminologies. |
| CO2 | Apply Sanskrit knowledge to comprehend ancient scientific texts. |
| CO3 | Analyze the contributions of Sanskrit literature to modern science and technology. |
| CO4 | Interpret technical terms and phrases in classical Sanskrit for academic purposes. |
| CO5 | Develop the ability to integrate Sanskrit knowledge into contemporary research contexts. |
|  | M.Tech I/ II Semester | AUD6104T - Value Education (Audit Course – I & II) | CO1 | Explain the fundamental concepts of value-based education and ethics. |
| CO2 | Analyze the role of values in personal and professional life. |
| CO3 | Apply ethical principles to real-life scenarios and decision-making processes. |
| CO4 | Evaluate the impact of cultural, social, and environmental values on society. |
| CO5 | Develop strategies for promoting ethical practices in personal and organizational contexts. |
|  | M.Tech I/ II Semester | AUD6105T - Constitution of India (Audit Course – I & II) | CO1 | Understand the key features, principles, and structure of the Indian Constitution. |
| CO2 | Analyze the roles and powers of the legislative, executive, and judiciary. |
| CO3 | Evaluate fundamental rights, duties, and directive principles in the context of civic responsibilities. |
| CO4 | Discuss the evolution and amendments of the Constitution in response to societal changes. |
| CO5 | Apply constitutional principles to real-life legal and ethical issues. |
|  | M.Tech I/ II Semester | AUD6106T - Pedagogy Studies (Audit Course – I & II) | CO1 | Explain the fundamental concepts and theories of pedagogy and learning. |
| CO2 | Analyze various teaching methods and their effectiveness in diverse learning environments. |
| CO3 | Apply pedagogical strategies to design effective learning experiences. |
| CO4 | Evaluate the impact of educational policies and practices on student learning outcomes. |
| CO5 | Develop critical thinking and reflective skills for continuous improvement in teaching practices. |
|  | M.Tech I/ II Semester | AUD6107T - Stress Management by Yoga (Audit Course – I & II) | CO1 | Understand the principles and practices of yoga for stress management. |
| CO2 | Analyze the physiological and psychological effects of stress and relaxation techniques. |
| CO3 | Apply yogic techniques to manage stress and promote mental well-being. |
| CO4 | Evaluate the effectiveness of yoga-based interventions in different contexts. |
| CO5 | Develop personal stress management plans integrating yoga practices. |
|  | M.Tech I/ II Semester | AUD6108T - Personality Development through Life Enlightenment Skills (Audit Course – I & II) | CO1 | Explain the principles of personality development and self-awareness. |
| CO2 | Analyze the role of emotional intelligence and communication skills in personal growth. |
| CO3 | Apply techniques for building confidence, leadership, and interpersonal skills. |
| CO4 | Evaluate personal strengths and areas for development through reflective practices. |
| CO5 | Develop strategies for continuous self-improvement and life-long learning. |
|  | M.Tech II Semester | Analog and Digital CMOS VLSI Design (MME6201T) | CO1 | Understand the basics of CMOS technology, including its fabrication process, materials, and process flow. Gain knowledge about NMOS and PMOS devices, their characteristics, and their role in digital and analog circuit design. |
| CO2 | Design and analyze the operation of a CMOS inverter as the basic building block of both analog and digital CMOS circuits. Understand the static and dynamic characteristics of the inverter, including noise margins, propagation delay, and power consumption. |
| CO3 | Develop skills in designing CMOS digital circuits, including logic gates (AND, OR, NAND, NOR), flip-flops, and sequential circuits. Understand the trade-offs between speed, area, and power in CMOS digital design. |
| CO4 | Analyze and characterize CMOS devices in terms of their IV characteristics, threshold voltage, on-state resistance, and sub-threshold behavior. Understand the effects of process variations, temperature variations, and scaling on device performance. |
| CO5 | Design key analog CMOS circuits, such as op-amps, voltage followers, current mirrors, differential amplifiers, and voltage-controlled oscillators (VCOs). Learn to evaluate circuit performance in terms of gain, linearity, input/output impedance, and frequency response. |
|  | M.Tech II Semester | VLSI Design Verification and Testing (MME6202T) | CO1 | Understand the overall VLSI design flow, including stages such as RTL design, synthesis, place and route, and verification. Gain an appreciation for the importance of verification and testing within this flow to ensure functional correctness and design reliability. |
| CO2 | Gain proficiency in modern verification methodologies like functional verification, formal verification, and equivalence checking. Understand the role of these methods in validating the behavior and correctness of VLSI designs at different stages of the design flow. |
| CO3 | Develop skills in writing and using testbenches for VHDL or Verilog-based designs. Learn to create testbenches that simulate the design's functionality and perform simulation-based verification to detect errors and ensure design correctness. |
| CO4 | Learn to perform static timing analysis (STA) to check for timing violations in digital designs. Understand how to set up and analyze timing constraints like setup, hold, clock skew, and propagation delays to ensure the design meets its performance requirements. |
| CO5 | Understand and apply Design for Testability (DFT) techniques, including the insertion of scan chains, boundary scan, and BIST (Built-In Self-Test). Learn how DFT enables efficient testing of complex VLSI circuits and reduces the cost of production testing. |
|  | M.Tech II Semester | Analog and Digital CMOS VLSI Design Lab (MME6201P) | CO1 | Gain practical experience in the design and simulation of CMOS-based analog and digital circuits using software tools like Cadence, Mentor Graphics, or Synopsys. Learn to design simple to complex circuits, simulate their behavior, and analyze results. |
| CO2 | Design, simulate, and test CMOS inverters and basic digital logic gates (AND, OR, NOT, XOR). Learn how to characterize the performance of CMOS logic gates in terms of parameters such as propagation delay, power consumption, and noise margins. |
| CO3 | Design and simulate analog CMOS amplifiers, such as common-source amplifiers, differential amplifiers, and operational amplifiers (op-amps). Analyze their frequency response, gain, and input/output impedance using simulation tools. |
| CO4 | Gain hands-on experience in layout design for CMOS circuits. Learn to design and verify layout structures, including the creation of schematics and physical layouts for CMOS inverters, logic gates, and amplifiers, ensuring compliance with design rules (DRC). |
| CO5 | Learn the process of digital circuit synthesis and optimization, focusing on the use of CMOS technology for efficient logic design. Perform area, power, and delay optimization for combinational and sequential circuits. |
|  | M.Tech II Semester | VLSI Design Verification and Testing Lab (MME6202P) | CO1 | Gain a comprehensive understanding of the VLSI design flow and the critical role of verification and testing in the overall process. Learn the stages of verification and how they integrate with synthesis, layout, and fabrication. |
| CO2 | Develop hands-on experience in using popular verification tools such as ModelSim, Cadence, Synopsys, and Mentor Graphics. Learn how to write testbenches, simulate digital designs, and verify their correctness at different abstraction levels (e.g., RTL, gate-level, post-layout). |
| CO3 | Learn how to design and develop testbenches to validate the functionality of RTL designs in HDL (Hardware Description Languages) such as VHDL or Verilog. Understand how to apply different simulation techniques (e.g., functional, timing, and race condition checking) to ensure the correctness of digital circuits. |
| CO4 | Perform RTL design verification using simulation tools, focusing on functional correctness. Test designs for corner cases, edge cases, and ensure that they meet timing requirements. |
| CO5 | Conduct static timing analysis (STA) to check the timing of digital circuits. Learn to use STA tools to verify the timing closure of designs, ensuring they meet performance requirements in terms of setup and hold times, clock skew, and propagation delays. |
|  | M.Tech II Semester | Mini Project (MME6203S) | CO1 | Apply interdisciplinary knowledge to solve complex engineering problems. |
| CO2 | Analyze project requirements to develop effective solutions. |
| CO3 | Evaluate project outcomes based on defined objectives and constraints. |
| CO4 | Design innovative engineering solutions through project-based learning. |
| CO5 | Understand project management principles in engineering contexts. |
|  | M.Tech II Semester | Memory Technologies (MME6203T) (Programme Elective-III) | CO1 | Understand the fundamental concepts and classification of memory technologies, including volatile (e.g., SRAM, DRAM) and non-volatile memories (e.g., EEPROM, Flash, FRAM). Understand their structure, operation, and basic principles of data storage. |
| CO2 | Comprehend the architecture and working of memory arrays. Learn the working principles of memory cells, address decoders, sense amplifiers, and read/write circuitry. Understand how these components interact to perform memory operations in various technologies. |
| CO3 | Analyze and compare Static RAM (SRAM) and Dynamic RAM (DRAM) technologies in terms of speed, power consumption, cost, and density. Understand the operation of flip-flop-based SRAM cells and the need for refreshing in DRAM. |
| CO4 | Learn how to design and optimize memory systems, considering factors such as access time, latency, bandwidth, power, and area. Apply this knowledge to system-level memory design for high-performance applications like cache memory, video memory, and system-on-chip (SoC) designs. |
| CO5 | Understand the principles and operation of non-volatile memory (NVM) technologies such as Flash Memory, EEPROM, and Ferroelectric RAM (FeRAM). Learn about NAND and NOR Flash memory, their structure, advantages, disadvantages, and applications in storage devices. |
|  | M.Tech II Semester | SoC Design (MME6204T) (Programme Elective-III) | CO1 | Understand the overall SoC design flow, including the stages of specification, design, synthesis, verification, and implementation. Learn how to integrate different system components (e.g., processors, memories, peripherals) into a single chip. |
| CO2 | Gain a comprehensive understanding of various SoC architectures, including the role of different processing units (e.g., CPU, DSP, FPGA, GPU) and memory subsystems. Learn about multiprocessor systems and heterogeneous computing in SoCs. |
| CO3 | Understand the concepts and techniques of hardware/software co-design in the context of SoC development. Learn how to partition tasks between hardware (e.g., custom logic, hardware accelerators) and software (e.g., application software, firmware) to optimize performance, power, and cost. |
| CO4 | Develop skills in designing, integrating, and customizing IP (Intellectual Property) cores such as processors, memory controllers, peripherals, and communication interfaces. Understand the role of IP reuse in speeding up the SoC design process and improving time-to-market. |
| CO5 | Learn about on-chip communication networks (e.g., AMBA, NoC (Network-on-Chip)) used to connect different components of the SoC. Understand the trade-offs between bus-based and network-based communication approaches and how to optimize the data flow across different modules in the SoC. |
|  | M.Tech II Semester | Low power VLSI Design (MME6205T) (Programme Elective-III) | CO1 | Understand the sources of power consumption in VLSI circuits, including dynamic power, static power, and short-circuit power. Learn how these components contribute to the overall power dissipation and the challenges in designing low-power systems. |
| CO2 | Gain skills in performing power analysis of VLSI circuits using tools and techniques for measuring dynamic and static power dissipation. Learn how to evaluate the power-performance trade-off and optimize circuit designs accordingly. |
| CO3 | Understand the principles of designing low-power CMOS circuits. Study techniques such as threshold voltage scaling, transistor sizing, and multi-threshold CMOS (MTCMOS) to reduce dynamic and static power consumption in CMOS-based designs. |
| CO4 | Apply low-power design techniques at the circuit level, including clock gating, power gating, sleep transistors, voltage scaling, and transistor sizing. Learn how these techniques can be used to minimize power dissipation while maintaining circuit performance. |
| CO5 | Study how low-power techniques can be applied at the architectural level. Learn to design low-power datapaths, multiplexers, and adders by using energy-efficient components, and consider techniques like bus encoding and functional unit optimization for power savings. |
|  | M.Tech II Semester | Communication Buses and Interfaces (MME6206T) (Programme Elective-IV) | CO1 | Understand the concept and significance of communication buses in embedded systems and digital systems. Learn about the role of buses in facilitating communication between processors, memories, and peripheral devices. |
| CO2 | Learn about various bus architectures, including single bus, multiplexed buses, and parallel buses. Understand the differences in performance, complexity, and data transfer speeds associated with different bus designs. |
| CO3 | Gain a strong foundation in standard communication protocols used in communication buses, such as I2C, SPI, UART, USB, CAN, and PCI. Learn about their features, advantages, and use cases in embedded systems and communication networks. |
| CO4 | Understand the differences between serial and parallel communication systems. Learn the advantages, limitations, and applications of serial communication (e.g., SPI, I2C) and parallel communication (e.g., PCI, ISA). |
| CO5 | Develop the skills to design and implement bus systems in digital systems, focusing on factors like data width, speed, and protocol compatibility. Learn how to select appropriate buses for specific applications in embedded systems, consumer electronics, and industrial automation. |
| 29 | M.Tech II Semester | Network Security and Cryptography (MME6206T) (Programme Elective-IV) | CO1 | Understand the basic concepts and principles of network security, including the confidentiality, integrity, availability, and authentication of data transmitted over networks. Learn about different types of security threats (e.g., attacks, malware, spoofing) and vulnerabilities in network systems. |
| CO2 | Gain an in-depth understanding of cryptographic algorithms used to secure data. This includes both symmetric key cryptography (e.g., AES, DES) and asymmetric key cryptography (e.g., RSA, ECC), as well as hash functions (e.g., SHA, MD5) and digital signatures. Learn how these algorithms are applied to protect data confidentiality and integrity. |
| CO3 | Learn about cryptographic protocols such as SSL/TLS, IPSec, Kerberos, and SSH, which are widely used to secure communication over the Internet. Understand how these protocols provide encryption, authentication, key management, and integrity. |
| CO4 | Understand the importance of key management in cryptographic systems, including key generation, key distribution, key exchange protocols (e.g., Diffie-Hellman, RSA), and techniques for ensuring the secure storage and usage of keys. |
|  |  |  | CO5 | Develop the skills to design and implement secure communication systems that incorporate cryptographic techniques to ensure the confidentiality, integrity, and authentication of data. Understand the trade-offs between security and system performance. |
| 30 | M.Tech II Semester | Physical design automation (MME6206T) (Programme Elective-IV) | CO1 | Gain a comprehensive understanding of the physical design flow of integrated circuits, including placement, routing, timing analysis, power optimization, clock distribution, and verification. Understand the stages of physical design and their significance in the overall IC design process. |
| CO2 | Learn how to create layout designs for digital circuits, with an understanding of the physical constraints imposed by the manufacturing process. Understand how to incorporate design rules, layout guidelines, and technology files in the physical design process to ensure manufacturability and functionality. |
| CO3 | Understand the principles and techniques of placement and floorplanning in IC design. Learn how to optimally place logic cells, standard cells, and IP blocks on a chip to minimize area, power consumption, and routing complexity. Understand how floorplanning impacts the design’s performance and manufacturability. |
| CO4 | Learn routing techniques and algorithms used to connect different components in the layout design. Study concepts like global routing, detailed routing, routing congestion, and design rule checking (DRC). Understand how to minimize wirelength, crosstalk, and interference in the routing phase. |
|  |  |  | CO5 | Understand how to perform timing analysis to ensure that the digital design meets performance requirements. Learn how to optimize for setup time, hold time, clock skew, and signal propagation delays. Study static timing analysis (STA) and its application in ensuring the timing closure of a design. |