SEMESTER II

ESD 5201: Device Drivers	L	T	P	C	Total hours
	3	0	0	3	36

Course Outcome

1. Explain the broad concept of device drivers and build character drivers.
2. Describe design of kernel modules and debugging these modules.
3. Handle concurrency, race condition and understand the importance of time while designing a device driver.
4. Allocate dynamic memory and communicating with devices though I/O ports.
5. Demonstrate and design USB drivers on a kit.

Unit	Topics	No. of Hours
I	Introduction to Device Drivers	3
II	Building & Running Modules	3
III	Character Driver	3
IV	Debugging Techniques	3
V	Concurrency and Race Condition	3
VI	Advanced Character Driver Operations	4
VII	Time, Delay and Deferred Work	3
VIII	Allocating Memory	4

IX	Communicating with Hardware	4
X	Interrupt Handling	4
XI	PCI Drivers, USB Drivers	2

References

1. Alessandro Rubini, "Linux Device Drivers", (Nutshell Handbook), O'Reilly Publishers, 2009.
2. John Madieu, "Linux Device Drivers Development: Develop customized drivers for embedded Linux", Packt Publishing, 2017.
3. Robert Love, "Linux Kernel Development", Addison Wesley, Third Edition, 2010.
4. Daniel P. Bovet, Marco Cesati, "Understanding the Linux Kernel", O'Reilly Media, Third Edition, 2008.
5. Wolfgang Mauerer, "Professional Linux Kernel Architecture", Wrox, 2008.
6. Sreekrishnan Venkateswaran, "Essential Linux Device Drivers", Prentice Hall, 2008.
7. W. Richard Stevens, Stephen A. Rago, "Advanced Programming in the UNIX Environment", Addison Wesley, Third Edition, 2013.
8. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, "Unix Network Programming, Vol1: Sockets", Pearson Education India, Third Edition, 2015.
9. MOOC: https://in.coursera.org/learn/linux-kernel-programming-yocto-project#syllabus

ESD 5202: Embedded Systems	L	T	P	C	Total hours
	3	0	0	3	36

Course Outcome

1. Employ the knowledge of Microcontrollers to build Embedded systems.
2. Explain the concept of Programming ARM Microcontrollers using Assembly and Embedded C.
3. Design a Real time Embedded Systems by interfacing Sensors, Actuators and porting Real time operating systems.

Unit	Topics	No. of Hours
I	**Introduction to Embedded Systems:** Design Challenges, Processors Technology, Design Technology	2
II	**Introduction to ARM Cortex processor:** Variants of Cortex and ARM versions, Comparison of M-series processor, Architecture, Programmers Model, APSR register, Memory Model, Exception, Interrupts, Reset	4
III	**Instruction Set Architecture:** More on Memory System, Exceptions and Interrupts, NVIC, Memory Protection Unit, Assembly Programming, Embedded C programming, CMSIS, Startup Code	3
IV	**Introduction to LPC13/17xx Microcontroller:** Memory Mapping, Registers involved and programming with GPIO, PWM	3
V	**Data Acquisition System:** ADC, Types of ADC, Choosing the ADC, DAC	4
VI	**Serial Communication:** UART, I2C, SPI, Interfacing	4
VII	**USB BUS:** Speed Identification on the bus, States, Packets, Data flow types, Enumeration, Descriptors, USB Interface -C Programs	6
VIII	**CAN BUS:** Introduction, Frames, Bit stuffing, Types of errors, Nominal Bit Timing, A simple application with CAN	4
IX	**Introduction to Multitasking in Microcontrollers:** Variants of RTOS, FreeRTOS, UCOS, uCLinux, FreeRTOS on Cortex based Microcontrollers, TASK CREATION, QUEQUES, SEMAPHORE, MUTEX, Application development	4
X	**Designing a Digital Camera:** Introduction, Requirement, Specifications, Implementation, Testing	2

References

1. Joseph Yiu, “The definitive guide to the ARM Cortex-M3”, Elsevier, 2nd Edition, 2010.
2. Frank Vahid, Tony Givargis, “Embedded System Design: A Unified Hardware/Software Introduction”, Wiley India, ISBN:81-265-0837-X, 2007.
3. Richard Barry, “NXP Semiconductors, LPC13xx/17xx User Manual”, 2012.
4. NXP Semiconductors, “LPCzone Examples”, 2012.

5. "FreeRTOS Reference Manual", Real Time Engineers Ltd., 2016.
6. MOOC: https://in.coursera.org/learn/introduction-embedded-systems#syllabus

ESD 5203: Embedded Software Design	L	T	P	C	Total hours
	3	0	0	3	36

Course Outcome

1. Analyze the OOP concepts for the embedded system applications.
2. Evaluate the applications using JAVA constructs for the general purpose and embedded systems.
3. Analyze the models for an embedded application using the concept of UML.
4. Interpret embedded application model using suitable diagrams using UML tool.

Unit	Topics	No. of Hours
I	**Object-oriented principles of composition:** Inheritance - Aggregation and containment - Delegation - Structural design patterns for composing objects.	3
II	**Specification of object-oriented systems:** Specification of object-oriented systems: UML for specifying functional requirements - Use cases and Scenarios - Subsystems, packages and deployment - Assigning responsibilities to objects in UML- Specifying quality attributes: Performance- Security - Privacy- Safety.	4
III	**Modelling object-oriented systems:** UML for modelling object-oriented systems-Class diagrams- Collaboration Diagrams-Sequence diagrams- State diagrams.	6
IV	**Modelling real-time embedded systems behaviours:** UML real-time profile.	2
V	**Developing object-oriented systems in Java:** Classes, interfaces, methods-Generics-Scope rules and access control. Inner classes-Functional programming	14

	constructs – lambdas-Threads, concurrency control and timers-I/O, Streams and network I/O-Security and Cryptography.	
VI	**Testing Java programs:** Challenges in testing object-oriented Program-Functional testing-Testing quality properties of the system- Java SE Embedded.	3
VII	**Compact Profile:** Overview and technical details-Compact1, Compact2, and Compact3 profiles and their capabilities-Designing systems using embedded profile.	2
VIII	**Realtime and Embedded Specification for Java:** Real-time threads – Asynchrony-Time. Clocks and Timers-System and Options – POSIX realtime signals-Examples of programs using realtime specifications for Java.	2

References

1. UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition), Addison-Wesley Professional. 2003.
2. The Java Programming Language. Ken Arnold, James Gosling and David Holmes. Addison-Wesley Professional; 4 edition (August 27, 2005)
3. http://www.oracle.com/technetwork/java/embedded/resources/tech/compact-profiles-overview-2157132.html
4. Realtime Specification for Java *2.0 https://java.net/projects/rtsj-2/pages/Home*
5. *MOOC:https://www.coursera.org/programs/manipal-education-tguaf?collectionId=&currentTab=CATALOG&productId=Xpv78UYjEeemlQrrzf9X-A&productType=course&showMiniModal=true*

ESD 5235: Multicore Program Optimization	L	T	P	C	Total hours
	3	0	0	3	36

Course Outcome

1. Distinguish between single core, multicore architectures, various architectures, trends, various levels of parallelisms.

2. Illustrate Various cache coherence, issues, memory consistency models, various protocols, working principles, performances.
3. Analyse Justification of primitives, optimizations, applications.

Unit	Topics	No. of Hours
I	**Introduction to parallel computers:** Introduction, why parallel architecture, application trends, technology trends, architectural trends	1
II	**Instruction Level Parallelism (ILP)**	1
III	**Cache memory**	5
IV	**Shared Memory Multiprocessors:** General architecture, Introduction to Interconnect, communication latency Problem of cache coherence; memory consistency models: SC, PC, TSO, PSO, WO/WC, RC; snoopy protocol: invalidate vs. update, MSI,MESI, MOESI, MOSI; performance trade-offs; synchronization primitives: atomic primitives; locks: TTS, tickets, array; barriers: central and tree; performance implications in shared memory programs; chip multiprocessors: why CMP (Moore's law, wire delay); shared L2 vs. tiled CMP; core complexity; power/performance	13
V	**Introduction to Basic optimization:** Hotspot, Faster Algorithms, ILP, Data Dependency, Branching, Memory, Loops, Slow Operations	2
VI	**Introduction to Performance Tools (Intel Software Tools):** Benchmark, Optimizing Compilers, Profilers, Performance Tools, Code Coverage Tools, Sampling vs Instrumentation	2
VII	**Introduction to Multicore Optimization:** ILP vs TLP, Data vs Task Parallelism, Parallel Application Case Studies, Parallelization Process	2
VIII	**Programming for Performance:** Partitioning for performance, Data Access and Communication, Orchestration, Performance factors, Case-Studies	2
IX	**MultiThreading with Open MP:** Threading, High Level vs Low Level Threading, Threading Goals and Issues, Introduction to OpenMP pragmas, Execution Model and Memory Model, Advanced OpenMP Topics	4

X	**Multithreaded Applications:** Some applications in Integer Programming, Digital Signal Processing (Video Codec)	4

References

1. The Software Optimization Cookbook High Performance Recipes for IA-32 Platforms, Richard Gerber,Aart J. C Bik, Kevin B. Smith, and Xinmin Tian, 2nd Edition, Intel Press
2. JComputer Architecture: A Quantitative Approach, Morgan Kaufmann Publishers,. L. Hennesey andD. A. Patterson. 3rd Edition
3. Parallel Computer Architecture: A Hardware/Software Approach. Morgan Kaufmann Publishers, D.E. Culler, J. P. Singh, with A. Gupta, 2nd Edition
4. Introduction to Concurrent Programming with GPUs | Coursera

ELECTIVE II

ESD 5001: Digital Signal Processing	L	T	P	C	Total hours
	3	0	0	3	36

Course Outcome

1. Analyse Fast Fourier Transform (FFT) algorithms on computational complexity.
2. Describe the structures for IIR and FIR filters.
3. Interpret Multirate Signal Processing and Adaptive Filters.
4. Explain architecture, memory management and pipelining concepts of General and TMS320C67XX Digital Signal Processor.

Unit	Topics	No. of Hours
I	Review: (Self Study) Introduction, Classification of signals and systems, brief discussions on z-transform, inverse z-transform & Fourier transform, DFT, linear convolution using circular convolution & DFT.	

3. Develop a Web application using ASP.NET.
4. Recognize the need for Client-Server Architecture.
5. Construct a web page using HTML and CSS.

Unit	Topics	No. of Hours
I	**Database Concepts:** Introduction to Transact SQL, Database optimization Techniques (Normalization), Creating database, Queries, sub queries, Joins, Stored Procedures, Triggers Tools: SQL server 2008 R2	10
II	**Programming:** ASP.NET: Introduction to ASP.NET, Clint server and web application design of presentation, business logic and storage functionality. WIN form and WEB form Designs using ASP.NET, Silverlight, Windows Client, HTML basics, CSS, AJAX, Java Scripts, Styling with Themes, Componentization (Code behind, Data Layers, User Controls), Roles and Profiles Tool: Visual Studio 2010	20
III	**Web Services:** The life Cycle of a Web Service, Structure of Web Service, Creating a Web Service, SOAP, Data Contracts, Binding, Security, Discovery, Publishing, WSDL	6

References

1. Beginning ASP.NET 4.0 with C# byChris Hart, John Kauffman, David Sussman, Chris Ullman
2. Professional C# - Simon Robinson, Christian Nagel- Wiley Publishing, Inc.

ESD 5238: Virtual Prototyping	L	T	P	C	Total hours
	3	0	0	3	36

Course Outcome

1. Understand different modeling abstractions and usecases of Virtual Prototype
2. Illustrate SystemC and TLM2.0 concepts through examples

3. Describe evolving modelling standards and FPGA/Emulation based prototyping

Unit	Topics	No. of Hours
I	**Introduction:** Overview of Virtual Prototyping methodology, purpose, VP in SoC development flow, Abstractions - Functional, Programmers View, Programmers View and Time, Cycle Approximate, Cycle Accurate, Usecases - SW shift left, architectural exploration and performance analysis, RTL co-verification, high level synthesis, hybrid prototyping	5
II	**Tools, languages and standards:** Tools Overview - QEMU, GEM5, Accellera, Commercial tools. Languages - C/C++ Standards and Methodologies - SystemC, TLM2.0, CCI, UVM-SC	3
III	**SystemC:** Overview, Simulation elaboration and semantics (sc_main, sc_start, end_of_elaboration, etc), core language class definitions (sc_module, sc_event, sc_port, etc), predefined channel class definitions (sc_signal, sc_buffer, sc_fifo, etc), SystemC data types and SystemC utilities (sc_report, sc_exception, etc)	12
IV	**Transaction Level Modelling 2.0:** Overview (Initiators, targets, bridges, coding style, etc), core interface (transport, DMI, debug), interface and sockets, generic payload, base protocol and phases, utilities, message passing interface and analysis ports	8
V	**Evolving methodologies:** Overview of CCI, Serial TLM and UVM-SC, Configuration, Usage guidelines	3
VI	**FPGA Prototyping and Emulation:** Introduction to FPGA, Real Number Modelling using SV synthesizable subset	5

References

1. Grotker, Thorsten, Stan Liao, Grant Martin, and Stuart Swan., System Design with SystemC, Springer Science & Business Media, 2007
2. Bhasker, Jayaram. A SystemC primer. Vol. 2. Allentown: Star Galaxy Publishing, 2002.
3. Black, David C., and Jack Donovan, eds. SystemC: From the ground up. Boston, MA: Springer US, 2004.
4. Doulos, SystemC Golden Reference Guides, available at https://www.doulos.com/reference-guides/
5. Online resource from accellera.org
6. MOOC: https://www.coursera.org/learn/c-plus-plus-a?specialization=coding-for-everyone#syllabus

II	**Entrepreneurial Traits:** Personality of an entrepreneur, Types of Entrepreneurs	5
III	**Process of Entrepreneurship:** Factors affecting Entrepreneurship process	6
IV	**Business Start-up Process:** Idea Generation, Scanning the Environment, Macro and Micro analysis	7
V	**Business Plan writing:** Points to be considered, Model Business plan	6
VI	**Case studies:** Indian and International Entrepreneurship	6

References

1. NVR Naidu and T. Krishna Rao, "Management and Entrepreneurship", IK International Publishing House Pvt. Ltd 2008.
2. Mohanthy Sangram Keshari, "Fundamentals of Entrepreneurship", PHI Publications, 2005

Semester II Lab

ESD 5251: Device Drivers Lab	L	T	P	C	Total hours
	0	0	3	1	36

Course Outcome

1. Explain the broad concept of device drivers and build character drivers.
2. Describe design of kernel modules and debugging these modules.
3. Handle concurrency, race condition and understand the importance of time while designing a device driver.
4. Allocate dynamic memory and communicating with devices though I/O ports.
5. Demonstrate and design USB drivers on a kit.

Unit	Topics	No. of Hours
I	Introduction to Device Drivers	6
II	Building & Running Module	6
III	Character Driver	6
IV	Debugging Techniques, Concurrency and Race Condition	6
V	Advanced Character Driver Operations, Time, Delay and Deferred Work, Allocating Memory	6
VI	Communicating with Hardware, Interrupt Handling, PCI Drivers, USB Drivers	6

References

1. Alessandro Rubini, "**Linux Device Drivers**", (Nutshell Handbook), O'Reilly Publishers, 2009.
2. John Madieu, "**Linux Device Drivers Development: Develop customized drivers for embedded Linux**", Packt Publishing, 2017.
3. Robert Love, "**Linux Kernel Development**", Addison Wesley, Third Edition, 2010.
4. Daniel P. Bovet, Marco Cesati, "**Understanding the Linux Kernel**", O'Reilly Media, Third Edition, 2008.
5. Wolfgang Mauerer, "**Professional Linux Kernel Architecture**", Wrox, 2008.
6. Sreekrishnan Venkateswaran, "**Essential Linux Device Drivers**", Prentice Hall, 2008.
7. W. Richard Stevens, Stephen A. Rago, "**Advanced Programming in the UNIX Environment**", Addison Wesley, Third Edition, 2013.
8. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, "**Unix Network Programming, Vol1: Sockets**", Pearson Education India, Third Edition, 2015.

ESD 5252: Embedded Systems Lab	L	T	P	C	Total hours
	0	0	3	1	36

Course Outcome

1. Employ the knowledge of Microcontrollers to build Embedded systems.
2. Explain the concept of Programming ARM Microcontrollers using Assembly and Embedded C.
3. Design a Real time Embedded Systems by interfacing Sensors, Actuators and porting Real time operating systems.

Unit	Topics	No. of Hours
I	Discover the usage of keil Microvision IDE and LPCXpresso IDE.	3
II	Develop ASM programs for data transfer, code conversation, arithmetic, logical, sorting applications.	9
III	Identify the peripheral of LPC13/17xx Microcontroller, its usage and summarize the CMSIS, Startup Code in LPCXpresso IDE.	3
IV	Design and development of data acquisition system.	6
VI	Illustrate usage of timer, counter and interrupt applications.	6
VII	Experiment with the usage of USB and CAN bus.	3
VIII	Demonstrate Multitasking in Microcontrollers using FreeRTOS.	6

References

1. Joseph Yiu, “The definitive guide to the ARM Cortex-M3”, Elsevier, 2nd Edition, 2010.
2. Frank Vahid, Tony Givargis, “Embedded System Design: A Unified Hardware/Software Introduction”, Wiley India, ISBN:81-265-0837-X, 2007.
3. Richard Barry, “NXP Semiconductors, LPC13xx/17xx User Manual”, 2012.
4. NXP Semiconductors, “LPCzone Examples”, 2012.
5. “FreeRTOS Reference Manual”, Real Time Engineers Ltd., 2016.

ESD 5253: Embedded Software Design Lab	L	T	P	C	Total hours
	0	0	3	1	36

Course Outcome

1. Apply the OOP concepts for the embedded system applications.
2. Implement the applications using JAVA constructs for general purposes and embedded systems.
3. Use of UML tools to represent an embedded application model using suitable diagrams.
4. Apply the UML concepts for the embedded applications

Unit	Topics	No. of Hours
I	**Object-oriented principles of composition:** Inheritance - Aggregation and containment - Delegation - Structural design patterns for composing objects.	3
II	**Specification of object-oriented systems:** Specification of object-oriented systems: UML for specifying functional requirements - Use cases and Scenarios - Subsystems, packages and deployment - Assigning responsibilities to objects in UML- Specifying quality attributes: Performance- Security - Privacy- Safety.	4
III	**Modelling object-oriented systems:** UML for modelling object-oriented systems- Class diagrams- Collaboration Diagrams-Sequence diagrams- State diagrams.	6
IV	**Modelling real-time embedded systems behaviours:** UML real-time profile.	2
V	**Developing object-oriented systems in Java:** Classes, interfaces, methods- Generics-Scope rules and access control. Inner classes-Functional programming constructs - lambdas-Threads, concurrency control and timers-I/O, Streams and network I/O-Security and Cryptography.	14
VI	**Testing Java programs:** Challenges in testing object-oriented Program-Functional testing-Testing quality properties of the system- Java SE Embedded.	3

VII	**Compact Profile:** Overview and technical details-Compact1, Compact2, and Compact3 profiles and their capabilities-Designing systems using embedded profile.	2
VIII	**Realtime and Embedded Specification for Java:** Real-time threads - Asynchrony-Time. Clocks and Timers-System and Options - POSIX realtime signals-Examples of programs using realtime specifications for Java.	2

References

1. UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition), Addison-Wesley Professional. 2003.
2. The Java Programming Language. Ken Arnold, James Gosling and David Holmes. Addison-Wesley Professional; 4 edition (August 27, 2005)
3. http://www.oracle.com/technetwork/java/embedded/resources/tech/compact-profiles-overview-2157132.html
4. Realtime Specification for Java 2.0 https://java.net/projects/rtsj-2/pages/Home

ESD 5285: Multicore Program Optimization Lab	L	T	P	C	Total hours
	0	0	3	1	36

Course Outcome

1. Understand the performance variation between single core and multicore computing.
2. Practice parallelism using POSIX thread and openMP.
3. Use the performance tools to understand the application benchmarking, optimization.

Unit	Topics	No. of Hours
I	Writing the function for large dataset to understand the performance variation between single core and multicore computing.	3
II	Introduction to POSIX threads and openMP libraries.	6

III	Practice parallelism using POSIX thread and openMP.	18
IV	Understanding the performance tools.	9

References

1. The Software Optimization Cookbook High Performance Recipes for IA-32 Platforms, Richard Gerber,Aart J. C Bik, Kevin B. Smith, and Xinmin Tian, 2nd Edition, Intel Press
2. JComputer Architecture: A Quantitative Approach, Morgan Kaufmann Publishers,. L. Hennesey andD. A. Patterson. 3rd Edition
3. Parallel Computer Architecture: A Hardware/Software Approach. Morgan Kaufmann Publishers, D.E. Culler, J. P. Singh, with A. Gupta, 2nd Edition

ELECTIVE II Lab

ESD 5051: Digital Signal Processing Lab	L	T	P	C	Total hours
	0	0	3	1	36

Course Outcome

1. Analyse Fast Fourier Transform (FFT) algorithms on computational complexity.
2. Describe the structures for IIR and FIR filters.
3. Interpret Multirate Signal Processing and Adaptive Filters.

Unit	Topics	No. of Hours
I	Discover the usage of signal processing tool kit in Matlab	6
II	Illustrate FFT Algorithms: Radix-2 DIT-FFT Algorithm, DIF-FFT Algorithm.	15

ESD 5288: Virtual Prototyping Lab	L	T	P	C	Total hours
	0	0	3	1	36

Course Outcome

1. Employ various SystemC and TLM 2.0 semantics and constructs
2. Experiment HW model development with optimum simulation speed
3. Practice various debugging, tracing and analysis features

Unit	Topics	No. of Hours
I	Develop and verify basic building block models: Delay element, Counter, Register, Memory, Finite State Machine, FIFO	12
II	Develop and verify complete HW IP model: Timer, Serial Protocol IP(I2C or SPI or UART), Data processing IPs (Encryption/Decryption or Encoder/Decoder)	18
III	Simulate and Verify System Verilog based design and emulate using FPGA	6

References

1. Doulos, SystemC Golden Reference Guides, available at https://www.doulos.com/reference-guides/
2. Online resource from accellera.org

BDA 5182: Principles of Data Visualization Lab	L	T	P	C	Total hours
	0	0	3	1	36

Course Outcome

1. Experiment web scrapping techniques to extract data from websites.
2. Implement NumPy and Pandas for data science operations with examples.
3. Organize data for visualization using data manipulation techniques.
4. Experiment different visualization techniques.
5. Use power BI for analytics and to manage workspace.

Program Outcome and Course Outcome Mapping

Sl. No.	Course Code	Course Name	Credits	PO1	PO2	PO3	PO4	PO5
1	ESD 5101	Advanced Computer Architecture	3	*		*	*	*
2	ESD 5102	Data Structures and Algorithms	3	*		*	*	*
3	ESD 5103	Microcontrollers and its Applications	3	*		*	*	*
4	ESD 5104	Real Time Operating Systems	3	*		*	*	*
5	ESD 5131	Computer Networks	3	*		*	*	
	ESD 5132	Internet of Things	3	*		*		*
	ESD 5133	Database Programming in Java	3	*		*	*	*
	VLS 5132	System on Chip Design	3			*	*	
6	ESD 5151	Advanced Computer Architecture Lab	1	*		*	*	*
7	ESD 5152	Data Structures and Algorithms Lab	1	*		*	*	*
8	ESD 5153	Microcontrollers and its Applications Lab	1	*		*	*	
9	ESD 5154	Real Time Operating Systems Lab	1	*		*	*	*
10	ESD 5181	Computer Networks Lab	1	*		*	*	
	ESD 5182	Internet of Things Lab	1	*		*		*
	ESD 5183	Database Programming in Java Lab	1	*		*	*	*
	VLS 5182	System on Chip Design Lab	1				*	*
11	MPT 5100	Mini Project - I	4	*	*	*	*	*
12	PSD 5100	Professional Skill Development - I	1	*	*			
13	ESD 5201	Device Drivers	3	*		*		*

14	ESD 5202	Embedded Systems	3	*		*	*	*
15	ESD -5203	Embedded Software Design	3	*		*	*	*
16	ESD 5235	Multicore Program Optimization	3	*		*	*	*
17	BDA 5132	Principles of Data Visualization	3			*		*
	ESD 5232	IT Project Management	3	*		*		*
	ESD 5233	Mobile Application Development using Android	3	*		*	*	*
	ESD 5001	Digital Signal Processing	3	*		*	*	
	ESD 5237	Web Application Development	3	*		*		
	ESD 5238	Virtual Prototyping	3	*		*	*	*
	ENP 5230	Entrepreneurship	3			*	*	
	VLS 5001	High Level Digital Design	3			*	*	*
18	ESD 5251	Device Drivers Lab	1	*		*		*
19	ESD 5252	Embedded Systems Lab	1	*		*	*	
20	ESD 5253	Embedded Software Design Lab	1	*		*	*	
21	ESD 5285	Multicore Program Optimization Lab	1	*		*	*	*
22	BDA 5182	Principles of Data Visualization Lab	1			*		*
	ESD 5282	IT Project Management Lab	1	*		*		*
	ESD 5283	Mobile Application Development using Android Lab	1	*		*	*	*
	ESD 5051	Digital Signal Processing Lab	1	*		*	*	
	ESD 5288	Virtual Prototyping Lab	1	*		*	*	*
	ESD 5287	Web Application Development Lab	1	*		*		
	ENP 5280	Entrepreneurship Lab	1			*	*	
	VLS 5051	High Level Digital Design Lab	1			*	*	*
23	MPT 5200	Mini Project - II	4	*	*	*	*	*
24	PSD 5200	Professional Skill Development - II	1			*	*	*
25	ESD 6098	Project Work	25	*	*	*	*	*