S R UNIVERSITY, WARANGAL

M.TECH (EMBEDDED SYSTEMS)

List of courses aligned with ARM University

1. Embedded Systems Design and Programming
2. Embedded Linux
3. Internet of Things
4. Computer architecture
5. Embedded Systems Design and Programming Lab
6. Embedded Linux Lab
7. VLSI Fundamentals: A Practical Approach
8. Advanced System- On- Chip Design
9. Advanced Digital Signal Processing
10. VLSI Fundamentals-A Practical Approach Lab
11. Advanced System- On- Chip Design Lab

S R UNIVERSITY, WARANGAL
M.TECH. I YEAR- I SEMESTER
EMBEDDED SYSTEMS

EMBEDDED SYSTEMS DESIGN AND PROGRAMMING

Pre-requisite: Basics of Microcontrollers

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

- Knowledge and understanding of Arm processor architectures as modern embedded computing platforms, Software design basics, software engineering principles.
- Ability to choose between different programing techniques for embedded system design.
- Ability to evaluate implementation results (e.g. speed, cost, power) and correlate them with the corresponding programing techniques.

UNIT I

Introduction to Embedded Systems Design

Introduction, Options for Building Embedded Systems, Example Embedded System: Microcontroller vs. Microprocessor, Attributes of Embedded Systems, MCU Hardware & Software for Concurrency, , Impact of Constraints, Target Board - FRDM-KL25Z, CPU Scheduling, Scheduling Approaches, Event-Triggered Scheduling using Interrupts, Static Schedule Example, Dynamic Schedule, Common Schedulers – (Cyclic executive - non-preemptive and static, Run-To-Completion Scheduler, Preemptive Scheduler) Task State and Scheduling Rules, What's an RTOS? Comparison of Timing Dependence, Comparison of RAM Requirements, Software Engineering FOR Embedded Systems

UNIT II

Microcontroller vs. Microprocessor, Cortex-M0+ Core, Architectures and Memory Speed, Instruction Set, Modes for Addressing Memory.

Cortex-M0+ Processor Core

Microcontroller vs. Microprocessor, Cortex-M0+ Core, Architectures and Memory Speed, Instruction Set, Modes for Addressing Memory

UNIT III

C Code as Implemented in Assembly Language

Programmer's World: The Land of Chocolate!, Processor's World, Program Translation Stages, Examining Assembly Code before Debugger, A Warning About Code Optimizations, Application Binary Interface, Using Registers - AAPCS Register Use Conventions, AAPCS Core Register, Memory requirements, accessing data in Memory

UNIT IV

Interrupts

Exception and Interrupt Concepts - Example System with Interrupt, Example Program Requirements & Design, Example Exception Handler, Types of interrupts, Interrupt service routine (ISR). Timing Analysis - Visualizing the Timing of Processor Activities, Big Picture Timing Behavior, Interrupt Response Latency, Maximum Interrupt Rate, and Program Design with Interrupts.

General Purpose Digital Interfacing
Basic Concepts, KL25Z GPIO Ports, GPIO Port Bit Circuitry in MCU, CMSIS - Accessing Hardware Registers in C, Coding Style and Bit Access, Using Masks, C Code, Clocking Logic, Connecting a GPIO Signal to a Pin, Pin Control Register

UNIT-V
Interfacing - Inputs: What's a One? A Zero? Outputs: What's a One? A Zero? Output Example: Driving LEDs.

Analog Interfacing
Analog to Digital conversion concepts, Digital to Analog Converter, Analog Comparator, Timers.

Serial Communication
Overview, Software Structure – Handling asynchronous Communication, Software Structure – Parsing Messages, KL25Z and Freedom Specifics, Asynchronous serial (UART) Communications, SPI Communications, I2C Communications, Protocol Comparison

TEXTBOOKS

1. Embedded Systems Fundamentals on Arm Cortex-M based Microcontrollers: A Practical Approach by Alexander G. Dean
2. The Designer's Guide to the Cortex-M Processor Family: A Tutorial Approach by Trevor Martin
3. The Definitive Guide to the ARM Cortex-M0 by Joseph Yiu
4. The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors, Third Edition by Joseph Yiu

REFERENCES

1. Joseph Yiu, "The definitive guide to ARM Cortex-M3", Elsevier, 2nd Edition
2. VenkatramaniB. and Bhaskar M. "Digital Signal Processors: Architecture, Programming and Applications" , TMH , 2nd Edition
3. Sloss Andrew N, Symes Dominic, Wright Chris, "ARM System Developer's Guide: Designing and Optimizing", Morgan KaufmanPublication.
4. Steve furber, "ARM System-on-Chip Architecture", PearsonEducation
5. White Paper: Cortex-M for Beginners - An overview of the Arm Cortex-M processor family and comparison
6. Technical references and user manuals on www.arm.com, NXP Semiconductor www.nxp.com and Texas Instruments www.ti.com

S R UNIVERSITY, WARANGAL
M.TECH. I YEAR- I SEMESTER
EMBEDDED SYSTEMS

EMBEDDED LINUX

Pre-requisite: Basics of operating systems, basics of software programing in C

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

- Knowledge and understanding of Embedded Linux Operating System architecture
- Ability to choose between different software tools for the development of an embedded Linux system
- Ability to evaluate implementation results (e.g. speed, cost, power) and correlate them with the corresponding system

UNIT I

Linux and Embedded Systems: An Introduction

An Embedded System, Embedded system components, Basic software, Operating systems for embedded systems, Why Linux-based embedded systems? Linux evolution, Linux-based embedded system: example1, example 2, example 3.

Linux-based Embedded System Component Stack

Linux-based embedded system components, Reference hardware model, Reference hardware model implementations, CPU memory map, The role of the bootloader, Possible scenarios, An example of bootloader operations, Linux kernel, Device tree, System programs, Application, Typical layout of the root filesystem.

UNIT II

Anatomy of a Linux-based System

Linux architecture, Conceptual view of the kernel, Process scheduler, Memory manager, external interfaces, Memory manager architecture, Virtual file system, i-node, i-node interface, File interface, Virtual file system architecture, Inter-process communication, Inter-process communication architecture, Network, Device trees, Device tree example for the UDOO NEO, Device tree syntax, Device tree content, Device tree addressing, The U-Boot bootloader, The U-Boot bootloader, UDOO NEO boot process, An example: NXP i.MX6 System-on-Chip, UDOO NEO boot process.

UNIT III

Configuration & Build Process of an Embedded Linux System

Introduction, The workflow, Build systems, Buildroot vs Yocto – general aspects, configuration, purpose, The Yocto Project, The Yocto build system, The build system workflow-configuration files, user configuration, Metadata, Machine (BSP) configuration, Distribution policy, source fetching, patching, configure/compile/install, output analysis/packaging, image generation, SDK generation.

UNIT IV

Introduction to Linux kernel modules

Introduction, CPU – I/O interface, I/O interface with polling, I/O interface with interrupt, I/O interface, I/O interface latency, Direct memory access (DMA) architecture - transfer modes, I/O taxonomy, Typical operations, Linux devices, The Virtual File System (VFS) abstraction, VFS– an example, VFS functions – include/linux/fs.h, The device file concept, Linux kernel modules – the initialization function, the cdev data structure, the initialization function, the clean-up function, custom VFS functions.

UNIT V

Communication Between Kernel and User Space

Introduction, The reference use case, The CPU/Device interface, The module level – file operations, ioctl() implementation, open()/release() implementation, read() implementation, Passing data to/from the kernel, write() implementation, communication with the device, Memory mapped I/O – initialization, clean-up, read, write, GPIO-based I/O – initialization, clean-up, read, write, Interrupts, Requesting the interrupt line, Freeing the interrupt line, The interrupt handler, Interrupt handling, Top-half and bottom-half, Needed support, Work queue, The user level, The user level – the application

Application Demo: Building a Ranging Sensor Kernel Module

Introduction, The sysfs file system - controlling GPIOs, Adding entries to the sysfs file system, Using sysfs and virtual file system API, The HC-SR04 ultrasonic ranging sensor, The HC-SR04 ultrasonic ranging sensor, Building Linux support for the HC-SR04 sensor, Module data structure definition, Module initialization function, Module clean-up function, Module open function, Module close function, Module write function, Module read function, Module show and store function, Module test application.

TEXTBOOKS

1. Embedded Linux Systems with the Yocto Project by Rudolf K. Sterif
2. Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux by Derek Molloy

REFERENCES

1. Chris Simmonds “Mastering Embedded Linux Programming” - Second Edition, PACKT Publications Limited.
2. Karim Yaghmour, “Building Imbedded Linux Systems”, O'Reilly & Associates
3. P Raghvan, Amol Lad, Sriram Neelakandan, “Embedded Linux System Design and Development”, Auerbach Publications

S R UNIVERSITY, WARANGAL
M.TECH. I YEAR- I SEMESTER
EMBEDDED SYSTEMS

INTERNET OF THINGS (PE- I)

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

- Understand the concept of IOT and M2M
- Study IOT architecture and applications in various fields
- Study the security and privacy issues in IOT.

UNIT I

IoT& Web Technology The Internet of Things Today, Time for Convergence, Towards the IoT Universe, Internet of Things Vision, IoT Strategic Research and Innovation Directions, IoT Applications, Future Internet Technologies, Infrastructure, Networks and Communication, Processes, Data Management, Security, Privacy & Trust, Device Level Energy Issues, IoT Related Standardization, Recommendations on Research Topics.

UNIT II

M2M to IoT – A Basic Perspective– Introduction, Some Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT, The international driven global value chain and global information monopolies. M2M to IoT-An Architectural Overview– Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.

UNIT III

IoT Architecture -State of the Art – Introduction, State of the art, Architecture Reference Model- Introduction, Reference Model and architecture, IoT reference Model, IoT Reference Architecture- Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views.

UNIT IV

IoT Applications for Value Creations Introduction, IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications, Four Aspects in your Business to Master IoT, Value Creation from Big Data and Serialization, IoT for Retailing Industry, IoT For Oil and Gas Industry, Opinions on IoT Application and Value for Industry, Home Management, eHealth.

UNIT V

Internet of Things Privacy, Security and Governance Introduction, Overview of Governance, Privacy and Security Issues,

TEXTBOOKS

1. Vijay Madisetti and Arshdeep Bahga, "Internet of Things (A Hands-on-Approach)", 1st Edition, VPT, 2014.
2. Francis daCosta, "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything", 1st Edition, Apress Publications, 2013.

REFERENCES

1. Cuno Pfister, "Getting Started with the Internet of Things", O Reilly Media, 2011.

S R UNIVERSITY, WARANGAL
M.TECH. I YEAR- I SEMESTER
EMBEDDED SYSTEMS

COMPUTER ARCHITECTURE (PE – II)

UNIT I

Fundamentals of Computer Design: Fundamentals of Computer design, Changing faces of computing and task of computer designer, Technology trends, Cost price and their trends, measuring and reporting performance, quantitative principles of computer design, Amdahl's law.
Instruction set principles and examples- Introduction, classifying instruction set- memory addressing- type and size of operands, operations in the instruction set.

UNIT II

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

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

UNIT III

Instruction Level Parallelism the Hardware Approach: Instruction-Level parallelism, Dynamic scheduling, Dynamic scheduling using Tomasulo's approach, Branch prediction, high performance instruction delivery- hardware based speculation.

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

UNIT IV

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

UNIT V

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

TEXT BOOK

1. John L. Hennessy, David A. Patterson, "Computer Architecture: A Quantitative Approach", 3rd Edition, Elsevier.
2. John P. Shen and Miikko H. Lipasti, "Modern Processor Design: Fundamentals of Super Scalar Processors", 2002, Beta Edition, McGraw-Hill

REFERENCE BOOKS

1. Kai Hwang, Faye A.Brigs., "Computer Architecture and Parallel Processing", Mc Graw Hill.
2. Dezso Sima, Terence Fountain, Peter Kacsuk, "Advanced Computer Architecture -A Design Space Approach", Pearson Education.

S R UNIVERSITY, WARANGAL
M.TECH. I YEAR- I SEMESTER
EMBEDDED SYSTEMS

EMBEDDED SYSTEMS DESIGN AND PROGRAMMING LAB

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

- Knowledge and understanding of Arm processor architectures
- Ability to use commercial tools to develop Arm-based embedded systems
- Ability to build an Arm-based embedded system and program to satisfy given user specifications

List of Experiments

1. CPU ASM Lab Exercise: Processing Text in Assembly Language
2. C as Implemented in Assembly Lab Exercise
3. Interrupt Lab Exercise: Stack use and Timing Behavior
4. General Purpose I/O Lab Exercise: Basic User Interface
5. ADC Lab Exercise: Voltage Monitor
6. Comparator Lab Exercise: Voltage Monitor
7. DAC Lab Exercise: Signal Generator
8. Timer Lab Exercise: Signal Generator with Precision Timing and Buffering
9. Serial Communications Lab Exercise: Performance Analysis
10. DMA Lab Exercise: Copy Speed Analysis

S R UNIVERSITY, WARANGAL
M.TECH. I YEAR- I SEMESTER
EMBEDDED SYSTEMS

EMBEDDED LINUX LAB

Course Outcomes: At the end of this course, students will be able to
1. Knowledge and understanding of Embedded Linux Operating System architecture
2. Ability to use industry standard tools to configure and build an embedded Linux system stack
3. Ability to develop kernel modules for customer peripherals

List of Experiments
1. Introduction to the Board and Workspace Set-Up
2. Custom Embedded Linux Build Using the Manual Approach
3. Introduction to Linux Kernel Modules under Yocto
4. Handling General Purpose I/O Using Linux Kernel Modules
5. Handling Hc-Sr04 Ranging Sensor Using Linux Kernel Modules
6. Introduction to Code Development and Debugging Using Yocto
7. Introduction to Linux Kernel and Application Profiling

S R UNIVERSITY, WARANGAL
M.TECH. I YEAR- II SEMESTER
EMBEDDED SYSTEMS

VLSI Fundamentals: A Practical Approach

Pre-requisites: Analog electronics& Digital electronics

Course Outcomes:
- Describe the characteristics of CMOS circuits.
- Optimization/ estimation of delay and power dissipation in the circuits.
- Develop the logic circuit for best critical path/ delays.
- Design and describe the data path circuits with proper clock distribution and wiring.

UNIT I
Introduction: CMOS Logic, Fabrication & layout, Design partitioning, Ex. Microprocessor, CMOS transistor theory, CMOS characteristics, Non-ideal effects, DC transfer characteristics.

UNIT II
Delay and power: Introduction, Transient response, RC delay model, linear delay model, logical efforts of paths, timing analysis of delay models. Dynamic power, static power, delay optimization.

UNIT III
Scaling and simulation: Reliability, scaling, simulation introduction, spice models, device models, device characterization, circuit characterization.

UNIT IV
Combinational and sequential circuit design: Circuit families, silicon on insulator circuit design, sub threshold circuit design. Sequencing static circuits, flip flops, synchronizers.

UNIT V
Data path subsystem and testing: Adders, subtractors, counters, multipliers, SRAM, clocking, testing, packaging, I/O & power distribution.

TEXT BOOKS
1. CMOS VLSI Design: A Circuits and Systems Perspective-Book by David Harris and Neil Weste
2. Digital Design and Computer Architecture-Book by David Harris.

REFERENCES
1. Digital VLSI Chip Design with Cadence and Synopsys CAD Tools-Book by Erik Brunvand

S R UNIVERSITY, WARANGAL
M.TECH. I YEAR- II SEMESTER
EMBEDDED SYSTEMS

ADVANCED SYSTEM- ON- CHIP DESIGN

Prerequisite: None

- Basics of hardware description language (Verilog or VHDL)
- Separate purchase of hardware and/or software tools, in order to replicate the course labs

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

- Knowledge and understanding of Arm Cortex-A processor architectures and Arm Cortex-A based SoCs
- Capture the design of Arm Cortex-A based SoCs in a standard hardware description language
- Apply low-level software design for Arm Cortex-A based SoCs and high-level application

UNIT I

Introduction to Programmable SoCs

SoC Design Concept, Moore's Law, Importance of Scaling, The Design Productivity Gap, Bridging the Design Productivity Gap, SoC - Inside an SoC, Example Arm-based SoC, Advantages of SoCs, Limitations of SoCs, SoC v Microcontroller v Processor, SoC Design Flow, SoC Examples: NVIDIA Tegra 2, Apple SoC Families.

Arm and Arm Processors

Arm Processors and Applications, Arm Processor Families, Arm Processors vs Arm Architectures, Arm and Thumb Instruction Sets, AAPCS, Processor modes, Vector table, Memory model, Memory types example Cached Arm Macrobell, Data Alignment, Endianness, Coprocessors, PMU, Trust Zone, Virtualization, Arm Cortex-A Series Processors, Arm Cortex-A9 Processor, Cortex-A9 MP Core, NEON-NEON Registers

UNIT II

Arm DS-5 Development Studio

Arm DS-5 Development Studio Overview, ARM DS-5- Code, Build, Debug, Debug Hardware, Virtual Debug Interface – VSTREAM, ARM DS-5 Analyser – Streamline, Energy Probe, Simulation, Device Configuration Database.

Armv7-A/R ISA

ARM Assembler, ARM assembler file syntax, Single/ Double register data transfer, Addressing Memory, Pre- and Post -Indexed Addressing, Multiple Register Data Transfer, Data Processing Instructions, Shift/Rotate Operations, Instructions for loading constants, Multiply/Divide, Bit Manipulation Instructions, Byte Reversal, Flow control, Branch instructions, Interworking, Compare and Branch if zero, Conditional Instructions, If Then, Coprocessor instructions, PSR access, DSP instructions overview, Saturated Maths and CLZ, Saturation, SIMD

UNIT III

ARM Cortex-A9 Processor

Cortex- A9 - MPCore, MPE Configuration, Media Processing Engine, Register Renaming, Virtual Flags Registers, Small Loop Mode, Program Flow Prediction, Performance Monitoring Unit (PMU), Cortex A9 supports ARMv7-A Architecture, caches, Data Cache, Memory Management Unit, ARM v7 Architecture Effects

AMBA AXI4 Bus Architecture

Bus -Types, Terminology, Operation, Communication Architecture Standards, ARM AMBA System Bus, AMBA 3 AXI Interface, AMBA 4 Specifications, AXI Components and Topology, Transaction Channels, Basic Signals, Clock and Reset, Channel Timing Example, Relationship Between the Channels

UNIT IV

AXI4-Lite GPIO Peripheral and DDR Memory Controller

AMBA AX14-Lite, AXI Low Power Interface, GPIO Overview, AX14-Lite GPIO, Computer Memory, Memory Accessing, Volatile vs Non-Volatile Memory, Types of Memory, Static RAM, Dynamic RAM, Non -Volatile Memory, Memory Controller, The Roles of a Memory Controller, Single Description example Timing

AXI UART and AXI4-Stream Peripherals

Serial Communication, Serial Communication vs Parallel Communication, Types of Serial Communication, UART Overview, UART Protocol, Character- Encoding Scheme, ASCII Encoded Characters, AXI UART Implementation, UART Control, UART Register Block, First In First Out(FIFO), UART FIFOs, Stream Data Transmission, AX-14 Stream Protocol, Data Streams, Global Signals, Master Signals, Slave Signals, Clock and Reset, Handshake, Packet Boundaries

UNIT V

AXI4-Stream with VGA Output Peripheral

VGA Overview, VGA Timing, VGA Interface, Utilization of FIFO, Hardware Implementation

AXI4-Stream with HDMI Input Peripheral

HDMI Overview, HDMI Interface, HDMI Signals: TMDS Channels, TMDS Timing, Data Display Channels, Consumer Electronics Control, Hot Plug Detect, AX14- Stream HDMI Input Peripheral, TMDS Deserialization and Decoding in Xilinx FPGA, Utilization of FIFO, TVALID / TUSER / TLAST Logic. Final Application: Image Processing

Final Application: Image Processing

Edge Detection, Image Scaling, Gray Scale, Intensity Gradient Magnitude, Software Programming: Edge Detection Algorithm.

TEXT BOOKS

1. ARM System-on-Chip Architecture by Steve B. Furber
2. ARM Assembly Language: Fundamentals and Techniques by William Hohl

REFERENCES

1. Cortex-A Series Programmer's Guide for ARMv7-A by Arm http://infocenter.arm.com/help/topic/com.arm.doc.den0013d/index.html
2. The Zynq Book Tutorials for Zybo and ZedBoard by Louise H Crockett (Author), Ross A Elliot (Author), Martin A Enderwitz

S R UNIVERSITY, WARANGAL
M.TECH. I YEAR- II SEMESTER
EMBEDDED SYSTEMS

ADVANCED DIGITAL SIGNAL PROCESSING (PE-III)

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

- To understand theory of different filters and algorithms
- To understand theory of multirate DSP, solve numerical problems and write algorithms
- To understand theory of prediction and solution of normal equations
- To know applications of DSP at block level.

UNIT I

Overview of DSP, Characterization in time and frequency, FFT Algorithms, Digital filter design and structures: Basic FIR/IIR filter design &structures, design techniques of linear phase FIR filters,IIR filters by impulse invariance, bilinear transformation, FIR/IIR Cascaded lattice structures, parallel realization of IIR.

UNIT II

Multi rate DSP, Decimators and Interpolators, Sampling rate conversion, multistage decimator & interpolator, poly phase filters, QMF, digital filter banks, Applications in subband coding.

UNIT III

Linear prediction & optimum linear filters, stationary random process, forward-backward linear prediction filters, solution of normal equations, AR Lattice and ARMA Lattice-Ladder Filters, Wiener Filters for Filtering and Prediction.

UNIT IV

Adaptive Filters, Applications, Gradient Adaptive Lattice, Minimum mean square criterion, LMS algorithm, Recursive Least Square algorithm

UNIT V

Estimation of Spectra from Finite-Duration Observations of Signals. Nonparametric Methods for Power Spectrum Estimation, Parametric Methods for Power Spectrum Estimation, Minimum-Variance Spectral Estimation, Eigen analysis Algorithms for Spectrum Estimation.

TEXTBOOKS

1. J. G. Proakis and D.G. Manolakis, "Digital signal processing: Principles, Algorithm and Applications", 4th Edition, Prentice Hall, 2007.
2. N. J. Fliege, "Multirate Digital Signal Processing: Multirate Systems -Filter Banks Wavelets", 1st Edition, John Wiley and Sons Ltd, 1999.

REFERENCES

1. Bruce W. Suter, "Multirate and Wavelet Signal Processing",1st Edition, Academic Press, 1997.
2. M. H. Hayes, "Statistical Digital Signal Processing and Modeling", John Wiley & Sons Inc., 2002.
3. S. Haykin, "Adaptive Filter Theory", 4th Edition, Prentice Hall, 2001.
4. D. G. Manolakis, V. K. Ingle and S. M. Kogon, "Statistical and Adaptive Signal Processing", McGraw Hill, 2000

S R UNIVERSITY, WARANGAL
M.TECH. I YEAR- II SEMESTER
EMBEDDED SYSTEMS

VLSI FUNDAMENTALS-A PRACTICAL APPROACH LAB

Prerequisites: Candence tools (Front end and back end)

List of experiments: (Cycle I)

1. NMOS and PMOS characteristics;
2. Common source amplifiers; layout of resistors, capacitors, transistors;
3. Differential amplifier;
4. Cascode amplifier;
5. Current mirror
6. Push pull CS amplifier;
7. Negative feedback amplifier;
8. Multistage amplifiers;
9. Operational amplifiers and
10. Comparators

List of Experiments: (Cycle II)

1. Cell Design and Verification
2. Datapath Design and Verification
3. Controller Design and Verification
4. Full Chip Assembly

REFERENCE BOOKS

1. Neil H.E Weste and Kamran Eshraghian, "Principles of CMOS VLSI Design", 2nd Edition, Addition ,Wesley, 1998.
2. Kiat Seng Yeo, Samir R.Rofail, Wang-Ling Gob, “CMOS/BiCMOS VLSI-Low Voltage, Low Power”, Pearson Education, Low price edition, 2003.
3. James D Plummer, Michael D. Deal, Peter B.Griffin, “Silicon VLSI Technology: fundamentals practice and Modeling”, Prentice Hall India, 2009.

S R UNIVERSITY, WARANGAL
M.TECH. I YEAR- II SEMESTER
EMBEDDED SYSTEMS

ADVANCED SYSTEM- ON- CHIP DESIGN LAB

Pre-requisites: Computer Architecture, VHDL Programming skills

Course Outcomes:

- Realize efficient instruction-processing of a computer processor
- Implement forwarding, different styles of cache-based memory hierarchies, and examine advanced strategies
- Enables students to design, verify and manage mid-size hardware projects by coding synthesizable VHDL and writing test benches or using on-FPGA logic analyzers

List of Experiments:

1. MIPS VHDL model and with the used design tools, refresh VHDL skills by coding small units

2. Pipelining.
   A. Hazard detection,
   B. stalling,
   C. forwarding

3. Caching
   A. Direct-mapped I-cache,
   B. 2-way set-associative D-cache,
   C. replacement strategies

4. Advanced topics (any one)
   A. Branch prediction,
   B. Multi-threading,
   C. SIMD instruction processing,
   D. Co-processors, busses,
   E. Multi-/many-core processing,
   F. Out-of-order execution

References:

1. John L. Hennessy, David A. Patterson. Computer Organization and Design - The Hardware/Software Interface. 4th Edition, Morgan Kaufmann, 2009.
2. Antonio Gonzalez, Fernando Latorre, Grigorios Magklis. Processor Microarchitecture - An Implementation Perspective, Synthesis Lectures on Computer Architecture, 2011.
3. John L. Hennessy, David A. Patterson. Computer Architecture - A Quantitative Approach. 5th Edition, Morgan Kaufmann, 2012.
4. Peter J. Ashenden. The Designer's Guide to VHDL. 3rd Edition, Elsevier, 2008.
5. The Zynq Book: http://www.zynqbook.com