# Honors in Embedded & Intelligent Systems

## Department of Electronics and Communication Engineering

Tabel IV Honors in Embedded & Intelligent Systems

| S.No. | Semester | Course Code | Course Name | Category | Type | Credit | L-T-P |
|---|---|---|---|---|---|---|---|
| 1 | V | | CAD Algorithms for Synthesis of VLSI Systems | | Theory | 3 | 3-0-0 |
| 2 | V | | Advanced Microcomputer Systems & Interfacing | | Theory | 3 | 3-0-0 |
| 3 | VI | | Advanced Embedded software design | | Theory | 3 | 3-0-0 |
| 4 | VI | | Computer vision | | Theory | 3 | 3-0-0 |
| 5 | VII* | | Internet of Things & IIoT | | Theory | 3 | 3-0-0 |
| 6 | VII* | | Pattern Analysis & Machine Intelligence | | Theory | 3 | 3-0-0 |
| 7 | VII* | | Reduced order Modeling, Optimization & Machine Intelligence | | Theory | 3 | 3-0-0 |
| 8 | VIII* | | Embedded SoC Design | | Theory | 3 | 3-0-0 |
| 9 | VIII* | | Quantum Computing | | Theory | 3 | 3-0-0 |
| 10 | VIII* | | Formal Verification of Digital Hardware & Embedded Software | | Theory | 3 | 3-0-0 |
| 11 | VIII* | | Mini Project on Embedded Systems | | Practical | 3 | 0-0-6 |

Laxmi Bhargava        SJNanda

**Course Name : CAD ALGORITHMS FOR SYNTHESIS OF VLSI SYSTEMS**

**Course Code :**

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

**Syllabus:**

Unit 1: **Introduction to CAD Algorithms**

Role of CAD in digital system design, levels of design, modeling & description and support of languages, RTL, gate and system level synthesis; Technological alternatives and technology mapping

Unit 2: **CAD Tools for synthesis**

CAD tools for synthesis, optimization, simulation and verification of design at various levels as well as for special realizations and structures such as microprogrammes, PLAs, gate arrays etc. Technology mapping for FPGAs. Low power issues in high level synthesis and logic synthesis.

Unit 3: **Architectural-Level Synthesis and Optimization**

Architectural Synthesis, Scheduling, Data path synthesis and control unit synthesis, scheduling algorithm, Resource Sharing and Binding

Unit 4: **Logic-Level Synthesis and Optimization**

Two-Level Combinational Logic Optimization, Multiple-Level Combinational Logic Optimization, Sequential Logic Optimization

Unit 5: **CAD Algorithms for VLSI Physical Design**

Introduction to VLSI Physical Design flow. Circuit partitioning, placement and routing algorithms. Design Rule-verification, Circuit Compaction; Circuit Extraction and post layout simulation. FPGA design flow- partitioning, placement and routing algorithms. Deep sub-micron issues; interconnects modeling and synthesis

**Course Outcomes :**

CO1: Is able to grasp various operations on graphs, clique, coloring, partitioning etc & apply graph algorithms and its applications into Boolean function representation (Skills- Apply)
CO2: Is able to grasp graph models for architecture representation (Cognitive- understanding)
CO3: Is able to analyze & implement two level/Multilevel/ sequential logic synthesis algorithms (approximate & exact algorithms) (skills- Analyze)
CO4: Is able to analyze & implement library binding algorithms- FSM equivalence & optimization (skills- Evaluate)
CO5: To able to grasp core concept of VLSI Physical Design algorithms. (Cognitive- Apply)

**References:**

1. G. D. Micheli. Synthesis and optimization of digital systems.
2. Dutt, N. D. and Gajski, D. D. High level synthesis, Kluwer, 2000.
3. T. H. Cormen, C. E. Leiserson and R. L. Rivest, "Introduction to Algorithms," McGraw-Hill, 1990.
4. N. Deo, Graph Theory, PH India.
5. Sait, S. M. and Youssef, H. VLSI Physical design automation. IEEE press, 1995.
6. Sherwani, N. VLSI physical design automation. Kluwer, 1999.

Dr. Satyasai Jagannath Nanda
DUGC Convener, DEPT OF ECE

Prof. Lava Bhargava,
HOD , DEPT of ECE

**Course Name: Advanced Microcomputer Systems & Interfacing**
**Course Code :**
**Credits : 3 (L-T-P : 3-0-0)**

# Course contents:

Unit-1: Introduction; Processor-processor (Intel/ARM) and micro controller (Intel/ARM), assembly language programming,

Unit-2: Interfacing methods-protocols, synchronization, parallel I/O, serial I/O, Memory interfacing, Digital/Analog interfacing, high speed I/O interfacing, data acquisition systems, CAN, I2C, USB, ESSI (Enhanced Synchronous Serial Interface) protocols; General Purpose Input/Output (GPIO)

Unit-3: Interrupt Synchronization & Timing generation- Features of interrupts, interrupt vectors & priority, polling, priority algorithms; frequency measurement, frequency and period conversion.

Unit-4: Miscellaneous- Serial and parallel port interfaces; State machine & concurrent process models.

Unit-5: System examples- camera etc; Debugging: JTAG, ISP, BDM Port, BITP, and DB9 ports.

Course outcomes:

- CO1. To Understand the 16,32,64-bit processors ISA (CISC and RISC)
- CO2. To understand the language and use of micro controller (ARM/Atmega 328 )
- CO3. To understand different I/O interface protocols and write programs for ARM interfaces
- CO4. To understand memory and different transducers and interfacing
- CO5. To write assembly programmes interfacing and design issues of embedded system(analytically and design issues)

Textbooks:

1. Jonathan W. Valvano, Embedded Microcomputer Systems: Real-Time Interfacing, Brookes/Cole, Pacific Grove, 2000.
2. Douglas V. Hall , Microprocessors and interfacing, McGraw Hills,
3. K.Ayala, The 8051 Microcontroller, Thompsons, Mazidi,Naimi, Naimi , avr microcontroller and embedded system, pearsons.
4. David A. Patterson and John L. Hennessy, Computer Organization and design ARM ed.,Morgan Kaufmann,
5. F. Vahid & T. Givargis, Embedded System Design, Wiley.
6. Wolf, W., Computers as Components: Principles of Embedded Computing System Design, Morgan Kaufmann, San Francisco, 2001.
7. Furber, S., ARM: system-on-chip architecture, 2nd Edition, Addison-Wesley, London, 2000.
8. Hayes, J. P., Computer Architecture and Organization, 3rd Edition, McGraw-Hill
9. Manuals- Intel 32/64 Architectures , ARM manual - 32/64-bit architecture, Intel 8051 and ATmega328P datasheet

**Course Name: Advanced Embedded Software Design**
**Course Code :**
**Credits : 3 (L-T-P : 3-0-0)**

**SYLLABUS :**

Unit-1: Processor micro-architecture, application-specific architecture,

Unit-2: Embedded OS, middleware, graphics libraries,

Unit-3: Software Development Tools, graphics IPs, virtual prototyping solutions,

RTOS, Embedded linux, concurrency & concurrent programming languages;

Unit-4: Design automation of such systems including methodologies, techniques and tools for their design as well as novel designs of software components

Course Outcomes:

CO1. Is able to grasp core concepts, basic tenets of micro-architecture vis a vis embedded operating systems, Unix shell programming (Cognitive- understanding) PO1

CO2. Is able to grasp features, properties of Embedded OS(Cognitive- understanding) PO1

CO3. Is able to learn & apply scheduling, deadlock avoidance algorithms and its applications into process scheduling, deadlock avoidance related problem solving etc. (Skills- evaluate) PO2, PO4, PO5, PO11

CO4. Is able in long perspective, to appreciate the significance of virtual memory, file management, security & privacy in OS (Skills- Analyze) PO1, PO4, PO12

CO5. Is able to write programmes for RTOS- scheduling, concurrency, deadlock prevention, etc.; and the significance that it can be used for analysis, problem solving as well as design of OS kernels (Skills- Evaluate) PO5

CO6. Is able to use Embedded OS CAD tools & development environment- VxWorks Windriver, RTLinux, Micrivision (ARM) (Skills- Create) PO2, PO13

Textbooks:

1. Unix Shell Programming, Kernighan & Pike, PHI
2. Lex & Yacc
3. Linux for Embedded and Real-time Applications, Doug Abbott, Newnes, Elsevier, 2003.
4. An Embedded Software Primer, David E. Simon, Addison Wesley, 1999.
5. Embedded Linux, Pearson.
6. Operating systems principles, Silberchatz, Galvin, Wiley
7. Short, K, Embedded Microprocessor System Design, Prentice Hall, 1998.
8. Embedded Linux, Pearson
9. Edward A. Lee, "Embedded Software", Advances in Computers (M. Zelkowitz, editor) 56, Academic Press, London, 2002.

10. Testing Embedded Software, by Bart Broekman and Edwin Notenboom, Pearson/Addison-Wesley (UK), ISBN: 0-321-15986-1

References:

11. Design Patterns for Embedded Systems in C: An Embedded Software Engineering, Bruce Powel Douglass, Newnes, Elsevier.
12. Software Engineering for Embedded Systems: Methods, Practical Techniques, Robert Oshana, Mark Kraeling, Newnes, Elsevier.
13. Embedded Software, Newnes know it all series, Jean J. Labrosse, ISSN 1879-8683, Elsevier.
14. Programming Embedded Systems in C and C++, O'Reilly Series, Michael Barr.
15. Real-Time Concepts for Embedded Systems, CMP books, R and D Developer Series, Qing Li, Caroline Yao, CRC press.
16. Testing Embedded Software, Broekman Bart, Pearson Education India, ISBN 813172509X, 9788131725092.
17. Performance Analysis of Real-Time Embedded Software, Yau-Tsun Steven Li, Sharad Malik, Kluwer.
18. Embedded Software for SoC, Ahmed Amine Jerraya, Springer.

**Course Name: Computer Vision**
**Course Code :**
**Credits : 3 (L-T-P : 3-0-0)**

## SYLLABUS :

Unit 1: Introduction and Overview: Overview of image processing systems, image formation and perception, continuous and digital image representation, image contrast enhancement, histogram equalization, affine transformations, model of image degradation/restoration process, Image Filtering

Unit 2: Feature Detection and Matching: Interest point detection, Edge, Blob, Corner detection; SIFT, SURF, HoG descriptors, Local Image Features and Feature Matching, RANSAC, Bag-of-words

Unit 3: Machine Learning and Deep Learning Quick course: Supervised & Unsupervised Machine Learning, Clustering , Classification, Review of Neural Networks, Convolutional Neural Network, CNN Architectures: AlexNet, VGG, InceptionNets, ResNets, DenseNets, Transfer Learning, Recurrent Neural Network, Long Short Term Memory(LSTM) , Visualization with CAM, Grad-CAM

Unit 4: CNNs for Computer Vision Tasks: Image Classification: CIFAR, MNIST, ImageNet Datasets, Object Detection: R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD; Segmentation: FCN, U-Net, Mask-RCNN

Unit 5: Recent Trends and Applications: Deep Generative Models, Generative Adversarial Networks (GANs), Attention Models, Graph Convolutional Networks, Zero-shot, One-shot, Few-shot Learning, Visual Question Answering, Image Captioning

Course Outcomes:

At the end of the course students should be able to:

CO1: Describe different image representation, their mathematical representation and different their data structures used. (Cognitive- Remembering, Understanding)

CO2: Implement feature extraction techniques for developing computer vision applications (Skills - Apply, create)

CO3: Recognize the object using the concepts of machine vision (Cognitive + Skill- Analyze)

CO4: Grasp the principles of state-of-the-art deep neural networks (Skills- Apply, Evaluate)

CO5: Develop the practical skills necessary to build computer vision applications (Skills- Apply, Evaluate)

References:

1. Computer Vision: Algorithms and Applications, by Richard Szeliski
2. Computer Vision: A Modern Approach, Forsyth and Ponce, Pearson Education.
3. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2008
4. Concise Computer Vision by Reinhard Klette
5. Deep Learning, by Goodfellow, Bengio, and Courville.
6. NPTEL Course Deep Learning for Computer Vision By Prof. Vineeth N Balasubramanian (IIT Hyderabad)

**Course Name: Internet of Things & IIoT**
**Course Code :**
**Credits : 3 (L-T-P : 3-0-0)**

**SYLLABUS :**

i) Introduction to IoT;
IoT sensors, devices, networks & protocols; Cyber physical Systems

ii) IoT programming & big data
Machine-to-Machine Communications; Interoperability in IoT, Introduction to Arduino Programming; Integration of Sensors and Actuators with Arduino; Introduction to Python programming, Introduction to Raspberry Pi, Implementation of IoT with Raspberry Pi; Data Handling and Analytics, Cloud Computing;

iii) Cybersecurity & privacy in IoT (optional)

iv) Low energy, secure hardware for IoT/sensors

v) Global applications (selected only)
Smart Cities and Smart Homes; Connected Vehicles, Smart transportation; Smart Grid, Industrial IoT; Case Study: Agriculture, healthcare including Smart monitoring of critical diseases & point of care; Activity Monitoring, supply chain & semiconductor manufacturing;

vi) Industrial IoT (IIoT): (selected only)
Enabling Factors; CPS, Energy Market; Example Deployment: Building Automation; Automotive and Transportation; Industrial (Manufacturing); building automation, agriculture, Oil & Gas; RTOS; Network Functions Virtualization; Long-range Wireless Protocols; LoRa WAN; Satellite Communications; ANT+, WiFi, ZigBee, WHART, EnOcean, Z-Wave, NFC; SECURITY: Encryption algorithms- Diffie-Hellman, Encryption Algorithms; Threat Vectors, Attacks: Man-in-the-middle, Replay, Protection Methods, Side-Channel Attacks, Chain of Trust; Hash and MAC Functions; Secure Firmware Updates, Random Number Generation; Predictive and Preventive Maintenance, IIOT deployment and Industrial Internet

Course Outcomes:

1. Able to grasp the concept of IoT and embedded systems, cyber physical systems
2. Exposing for the end-to-end design of Internet-of-Things applications from sensors to cloud, as well as hardware design/security aspects
3. Building in confidence and capability regarding electronics, sensors, and software through hands-on labs.
4. Providing exposure to practical problems and their solutions, through case studies using EDA Tools (Electronic Design Automation tools).
5. Enhancing the knowledge to Security and privacy needs, and the analysis required to address these needs.

Books:

1. "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", by Pethuru Raj and Anupama C. Raman (CRC Press)
2. "Internet of Things: A Hands-on Approach", by Arshdeep Bahga and Vijay Madisetti (Universities Press)

**Course Name: Pattern Analysis & Machine Intelligence**

**Course Code :**

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

**SYLLABUS :**

UNIT I

Introduction to Pattern Recognition and Machine Learning, ML design cycle, types of learning: supervised, unsupervised and reinforced, introduction to feature extraction and classification, density and discriminant functions, decision surfaces.

Review of probability theory: conditional probability, Bayes theorem, random variables, density and mass functions, expectation and variance, joint distribution function of multiple random variables, multivariate normal distribution.

UNIT II

Bayesian decision theory, Bayes Classifier, Naïve Bayes classifier, Euclidian distance and Mahalanobis distance-based classifiers, minimum-error-rate classification. Parameter estimation methods, Maximum-Likelihood estimation, Gaussian mixture models, Expectation maximization method, Bayesian estimation,

Hidden Markov models for sequential pattern classification: discrete hidden Markov models, continuous density hidden Markov models.

UNIT III

K-nearest neighbour classification: Simple and distance weighed voting approach. Support vector machines: linear SVM, soft-margin approach for non-separable data, kernel trick to learn non-linear SVM, radial basis function, polynomial, and sigmoidal kernel. Decision tree classifier: set of questions, splitting criterion, stop-splitting rule, and class assignment rule in decision tree. Introduction to neural networks: perceptron as linear classifier and multi-layer perceptron. Regression Analysis.

UNIT IV

Feature extraction methods: statistical features, Fourier and wavelet transforms for feature extraction, Data transformation and dimension reduction: Fisher's linear discriminant analysis, Bayesian LDA, step-wise LDA, principal component analysis, kernel-PCA. Optimization in feature selection. Feature visualization.

Ensemble of classifiers. Evaluation the performance of a classifier: holdout, random sampling, and cross validation methods, sensitivity, specificity, confusion matrix and ROC curve. Multi-class classification. Statistical analysis for comparison of significance of multiple classifiers over multiple dataset: Template matching and context dependent classification. The curse of dimensionality.

UNIT V

Unsupervised learning and clustering. Criterion functions for clustering, Prototype-based, Graph-based, Density-based clusters. Algorithms for clustering: K means, DBSCAN, Hierarchical clustering, Cluster validation.

Reference Text-Books

1: R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, John Wiley, 2001

2: S. Theodoridis and K. Koutroumbas, Pattern Recognition, 4th Ed., Academic Press, 2009

3: C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

4: T. Pang-Ningm, V. Kumar, M. Steinbach. Introduction to data mining. Pearson Education India, 2018.

Programming Books

1: S. Theodoridis *et. al.*, 'Introduction to Pattern Recognition, A MATLAB Approach'. Academic Press, 2010

**Course Name: Reduced order modeling, Optimization and Machine Intelligence**
**Course Code :**
**Credits : 3 (L-T-P : 3-0-0)**

**SYLLABUS :**

A. Reduced order modelling & large Eigen value methods-

(i) (a) Large Matrix analysis and large Eigen value problem– Groups, fields and rings; vector spaces; basis & dimensions; canonical forms; inner product spaces- orthogonalization, Gram-Schmidt orthogonalization, unitary operators, change of orthonormal basis, diagonalization; (b) Eigenvalues & eigen vectors- Gerschghorin theorem, iterative method, Sturm sequence, QR method, introduction to large eigen value problems.

(ii) Reduced order modelling of systems- Taylor's polynomial, least square approximation, Chebyshev series/polynomial, curve fitting & splines, Pade & rational approximation

B. Discrete Structures, algorithms & Combinatorial optimization- counting methods, algorithm analysis, graph algorithms, dynamic algorithms, randomized algorithms, probabilistic algorithms, combinatorial optimization

C. Digital arithmetic & machine intelligence-

(i) Number theory & computer arithmetic- unconventional number systems, residue number system, logarithmic number system, Chinese remainder theorem; fast evaluation of elementary & transcendental arithmetic functions.

(ii) Preface to AI- first order logic & inferencing, uncertainty, probabilistic reasoning systems, making decisions under uncertainty.

Suggested references (not limited to)-

1. Schaum's outline on Linear Algebra, McGraw Hill
2. Topics in Algebra, I. N. Herstein, Wiley.
3. Advanced Model Order Reduction Techniques in VLSI Design, Sheldon Tan, Lei He, Cambridge Univ. Press, 2007.
4. Model Order Reduction: Theory, Research Aspects and Applications edited by W. H. A. Schilders, Henk A. Van Der Vorst, Joost Rommes, Springer.
5. Gerald, C F; Wheatley P O; Applied Numerical Analysis, Pearson, 2017
6. Theory and Applications of Numerical Analysis, G. M. Phillips, Peter J. Taylor, Academic press
7. Discrete Structures, Schaum outline
8. Cormen, Rivest, Leiserson, Introduction to Algorithms, PHI
9. Combinatorial optimization, Papadimitriou and Steiglitz, PHI (I)
10. Israel Koren, Computer Arithmetic- Academic Press
11. Russel and Norvig- Artificial Intelligence: A Modern Approach, Pearson, 3rd Ed. 2017

Further references

1. Luigi FORTUNA, Guiseppe NUNNARI, Antonio GALLO, MODEL ORDER REDUCTION TECHNIQUES WITH APPLICATIONS IN ELECTRICAL ENGINEERING, Springer, 1992.
2. Y. Saad, Numerical methods for large Eigenvalue problems, www.umn.edu
3. Matrix Analysis & linear algebra, Meyer, SIAM
4. H. A. van der Vorst, Iterative methods for large linear systems, citeseerx.ist.psu.edu
5. Cheng et al, Symbolic analysis and reductions of VLSI circuits, Springer, 2005

Course outcomes

- CO1. Is able to grasp core concepts, basic tenets of linear algebraic structures- groups, fields and rings;vector spaces (knowledge)
- CO2. Is able to grasp features, properties and operations on vector spaces- orthogonalization, change of basis, diagonalization (knowledge)
- CO3. Is able to learn & apply problem solving for computing eigen values and eigen vectors etc. (Thinking, skills)
- CO4. Is able to demonstrate application of algorithms (Gerschgorin, Sturm sequence method, QR method) for eigen value computation/estimation and MATLAB/SCILAB validation(skills)
- CO5. Is able to describe algorithms for function approximation, fitting (rational, Chebychev, Pade etc.) using MATLAB (skills)
- CO6. Develops appreciation for combinatorial optimization algorithms, AI probabilistic approaches & implements through MATLAB/C++/SCILAB (skills)

**Course Name: Embedded SoC Design**
**Course Code :**
**Credits : 3 (L-T-P : 3-0-0)**

## Course contents:

Unit-1: Embedded computing- Microprocessors, embedded design process, system description formalisms. Instruction sets- CISC and RISC; CPU fundamentals- programming I/Os, co-processors, supervisor mode, exceptions, memory management units and address translation, pipelining, super scalar execution, caching, CPU power consumption.

Unit-2: Embedded computing platform- CPU bus, memory devices, I/O devices, interfacing, designing with microprocessors, debugging techniques.

Unit-3: Program design and analysis- models of program, assembly and linking, compilation techniques, analysis and optimization of execution time, energy, power and size.

Unit-4: Processes and operating systems- multiple tasks and multiple processes, context switching, scheduling policies, inter-process communication mechanisms.

Unit-5: Hardware accelerators- CPUs and accelerators, accelerator system design.

Unit-6: Networks- distributed embedded architectures, networks for embedded systems, network-based design, Internet-enabled systems.

Unit-7: System design techniques- design methodologies, requirements analysis, system analysis and architecture design, quality assurance.

Course Outcomes:

CO1: knowledge of embedded processors, RISC and CISC (Cognitive- Understand)

CO2: Basic concepts in CPU operation (Cognitive- Understand)

CO3: Understanding of I/O devices and their interfacing (Cognitive- Understand)

CO4: To learn program and system design and analysis methodologies (Skills- analyze and design)

Textbooks:

1. Wolf, W. Computers as components- Principles of embedded computing system design. Academic Press (Indian edition available from Harcourt India Pvt. Ltd., 27M Block market, Greater Kailash II, New Delhi-110 048.)

**Course Name : Quantum Computing**
**Course Code :**
**Credits : 3 (L-T-P : 3-0-0)**

**Syllabus:**

**Introduction to Quantum Computing**
Basics of quantum mechanics, quantum postulates, superposition, entanglement, No cloning theorem, Qubits, measurement on single and multiple qubits, Hilbert space, state vector Bra ket notation, Bloch sphere,

**Quantum Computing and Networking Components**
Single and two Qubit quantum gates, unitary transformation, composite state tensor product, cascade and parallel quantum circuits, Quantum diode, quantum router, quantum memory

**Potential Quantum Algorithms**
Quantum key distributions, quantum teleportation, Grover's algorithm (data base search) and Shor's algorithm (integer factorization), Deutsch–Jozsa algorithm, quantum Fourier transform (QFT), overview of open-source software for working with quantum computers at the level of circuits, and algorithms

**Physical Realization of Quantum Circuits**
Physical representation of Qubit, Nuclear magnetic resonance (NMR), Trapped ion, linear optics quantum computing (LOQC), Quantum electrodynamics (QED), Superconducting quantum computer

**Application of Quantum Computing**
Quantum communication and Cryptography, Quantum error correction (QEC), Quantum Machine learning, Big data search, Drug simulation

Course outcomes:
CO1: Understand basic concepts of quantum computing
CO2: Understand and analyze quantum switching, storage and computing devices
CO3: Apply and analyze quantum algorithms
CO4: Understand various quantum circuit implementation techniques
CO5: Design various applications of quantum computing
Reference Books:
1. Quantum Computation and Quantum Information" by Michael Nielsen and Isaac Chuang
2. Quantum computing : a gentle introduction / Eleanor Rieffel andWolfgang Polak.
3. Quantum Computing for Everyone" by Chris Bernhardt
4. Quantum Computing: An Applied Approach" by John Preskill
5. Quantum Computing: From Linear Algebra to Physical Realizations" by Masahiro Hotta and Keiji Matsumoto

SJNanda
Dr. Satyasai Jagannath Nanda
DUGC Convener, DEPT OF ECE

Lava Bhargava
Prof. Lava Bhargava,
HOD , DEPT of ECE

**Course Name: Formal Verification of Digital Hardware & Embedded Software**
**Course Code :**
**Credits : 3 (L-T-P : 3-0-0)**

**SYLLABUS :**

UNIT 1. Introduction to Design Verification, OVM and UVM methodology, case studies using Verilog and System Verilog, Static verification,

UNIT 2. Formal Verification of digital hardware systems- BDD based approaches, functional equivalence, finite state automata, FSM verification, Model checking, Various industry & academia CAD tools for formal verification.

UNIT 3. Verification, validation & testing - Debugging techniques for embedded software, instruction set simulators, clear box technique, black box testing, evaluating function test

UNIT 4. Recent trends in Design verification, case study

Course Outcomes:

CO1: To understand features of System Verilog (Cognitive- Understanding)

CO2: To study Assertion Based Verification and also be aware of functional coverage. (Cognitive-Analyze/Evaluate)

CO3: To apply language constructs of Bluespec for high level design/synthesis. (Skills- Apply)

CO4: To understand the necessity of the verification methodology. (Affective- understanding)

CO5: Ability to develop the test bench for DUT with verification methodology for scheduling, resource sharing and binding.(Skills- Creativity)

Additonal/optional Outcomes:

6) Understand significance of formal verification methodologies vis-à-vis simulation/ABV (Knowledge)

7) To perform equivalence check for combinational as well as sequential digital circuits (Thinking)

8) To develop Kripe structure based Model for sequential circuits and write PROPERTIES for Model Checking (Thinking)

9) To be able to use Model checking CAD tool- SMV or Cadence/Synopsys tool (SMV or Formality) (Skills)

Textbooks:

1. Discrete Structures, Logic and Computability- James L. Hein, Jones & Barlett India.
2. Logic- Schaum Series
3. [Chapter 2, Micheli, Synthesis of Digital Systems, McGrawHill]
4. Articles by Bryant, Eap, Akers on BDDs.
5. Advanced Formal Verification, R. Drechsler, Kluwer.
6. Algorithms & Data structures in VLSI Design, C. Meinel and T. Theobald, Springer.

References:

1. SystemVerilog IEEE standard;
2. BlueSpec user guide/standard;
3. Embedded systems Design- Artist Roadmap for Research & Development, LNCS-3436, Springer.
4. J. W. Valvano, Embedded microcomputer systems- Real Time Interfacing, , Thomson press (Cengage India)
5. Computers as components- Principles of embedded computing system design. Wolf, W., Academic Press (Indian edition available from Harcourt India Pvt. Ltd., 27M Block market, Greater Kailash II, New Delhi-110 048.)
6. Verification, validation & testing in software engineering, A. Dasso and A. Funes, Idea Group Inc.
7. Hardware-Software codesign for data flow dominated embedded systems, R. Niemann, Springer.
8. Readings in Hardware/Software codesign, Micheli, Ernst, Wolf, Morgan Kaufmann.
9. Advanced Formal Verification, R. Drechsler, Kluwer.
10. Readings in Hardware/Software codesign, Micheli, Ernst, Wolf, Morgan Kaufmann.

**Course Name: Mini Project on Embedded Systems**

**Course Code :**
**Credits : 3 (L-T-P : 0-0-6)**

**List of Experiments/ activities**

Design, verification, prototyping and implementation of hardware.

Circuits and systems based on software, hardware, algorithms, concepts in emerging areas such as design of digital logic (combinational and sequencial circuit) on FPGA Boards, design of test bench using System C, System verilog for functional verification, modeling of sensors, microcontroller .