EMBEDDED SYSTEM DESIGN

MODULE – I (13 hours)

Introduction to Embedded Computing: Terms and scope, Application areas, Growing importance of embedded systems. Specifications: Requirements, Models of computation, State Charts: Modeling of hierarchy, Timers, Edge labels and StateCharts semantics, Evaluation and extensions, General language characteristics: Synchronous and asynchronous languages, Process concepts, Synchronization and communication, Specifying timing, Using non-standard I/O devices, SDL, Petri nets: Introduction, Condition/event nets, Place/transition nets, Predicate/transition nets, Evaluation, Message Sequence Charts, UML, Process networks: Task graphs, Asynchronous message passing, Synchronous message passing, Java, VHDL: Introduction, Entities and architectures, Multi-valued logic and IEEE 1164, VHDL processes and simulation semantics, System C, Verilog and System Verilog, Spec C, Additional languages, Levels of hardware modelling, Language comparison, Dependability requirements.

MODULE – II (13 hours)

Embedded System Hardware: Introduction, Input: Sensors, Sample-and-hold circuits, A/D-converters, Communication: Requirements, Electrical robustness, Guaranteeing real-time behaviour, Examples, Processing units: Application-Specific Circuits (ASICs), Processors, Reconfigurable Logic, Memories, Output: D/A-converters, Actuators.Standard Software: Embedded Operating Systems, Middleware, and Scheduling: Prediction of execution times, Scheduling in real-time systems: Classification of scheduling algorithms, Aperiodic scheduling, Periodic scheduling, Resource access protocols, Embedded operating systems: General requirements, Real-time operating systems, Middleware: Real-time data bases, Access to remote objects

MODULE – III (14 hours)

Implementing Embedded Systems: Hardware/Software Co-design: Task level concurrency management, High-level optimizations: Floating-point to fixed-point conversion, Simple loop transformations, Loop tiling/blocking, Loop splitting, Array folding, Hardware/software partitioning: Introduction, COOL, Compilers for embedded systems: Introduction, Energy-aware compilation, Compilation for digital signal processors, Compilation for multimedia processors, Compilation for VLIW processors, Compilation for network processors Compiler generation, retargetable compilers and design space exploration, Voltage Scaling and Power Management: Dynamic Voltage Scaling, Dynamic power management (DPM), Actual design flows and tools: SpecC methodology, IMEC tool flow, The COSYMA design flow, Ptolemy II, he OCTOPUS design flow.Validation: Introduction, Simulation, Rapid prototyping and emulation, Test: Scope, Design for testability and Self-test programs, Fault simulation, Fault injection, Risk- and dependability analysis, Formal verification.

Textbooks:

1.Peter Marwedel, Embedded System Design, Springer, 2006 http://ls12-www.cs.uni-dortmund.de/~marwedel/kluwer-es-book/

Recommended Reading:

1.Wayne Wolf, Computers as Components, Morgan Kaufmann, 2001 http://www.ee.princeton.edu/~wolf/embedded-book

2. G. De Micheli, Rolf Ernst and Wayne Wolf, eds, Readings in Hardware/Software Co-Design, Morgan Kaufmann, Systems-on-Silicon Series Embedded

3. Frank Vahid and Tony D. Givargis, System Design: A Unified Hardware/Software Introduction, Addison Wesley, 2002.

4. Michael Barr, Programming Embedded Systems in C and C++, O'Reilly, 1999.

5. David E. Simon, An Embedded Software Primer, Addison Wesley, 1999.

6. Jack Ganssle, The Art of Designing Embedded Systems, Newnes, 2000.

7. K. Short, Embedded Microprocessor System Design, Prentice Hall, 1998. C. Baron, J. Geffroy and G. Motet, Embedded System Applications, Kluwer, 1997.