CS System On Chip Architecture and Design

Objective

- Model and specify embedded systems at high levels of abstraction.
- Understand Assembly and high level language support for ARM Architecture.
- Analyze hardware/software tradeoffs, algorithms, and architectures to optimize the system based on requirements and implementation constraints.
- Describe architectures for control-dominated and data-dominated systems and real-time systems.
- Understand hardware, software, and interface synthesis and describe examples of applications and systems developed using a co-design approach.
- Appreciate issues in system-on-a-chip design associated with co-design, such as intellectual property, reuse, and verification.

Unit-I

Introduction to Processor Design- Abstraction In Hardware Design. Muo A Simple Processor, Processor Design Trade Off, Design for Low Power Consumption. ARM Processor as System – on-Chip - Acorn RISC Machine- Architecture Inheritance – ARM Programming Model ARM Development Tools – 3 And 5 Stage Pipeline ARM Organization – ARM Instruction Execution and Implementation – ARM Co-Processor Interface.

Unit – II

ARM Assembly Language Programming - ARM instructions types- data transfer, data processing and control flow instructions- ARM co-processor interface. Architectural support for high level language - data types- abstraction in software design- expressions – loops – functions and procedures-conditional statements – use of memory. Memory Hierarchy - memory size and speed-on-chip memory – caches- cache design an example-memory management

Unit – III

Architectural Support for System Development - Advanced Microcontroller bus architecture- ARM memory interface – ARM reference peripheral specification – Hardware system prototyping tools – Armulator – Debug architecture Architectural Support for Operating System - An introduction to operating systems – ARM system control coprocessor- CP15 protection unit registers - ARM MMU Architecture – Synchronization – Context Switching input and output .

Unit -IV

System-level and SoC design methodologies and tools - HW/SW co-design: analysis, partitioning, real-time scheduling, hardware acceleration.Virtual platform models - co-simulation and FPGAs for prototyping of HW/SW systems.

Unit-V

Transaction-Level Modeling (TLM), Electronic System-Level (ESL) languages , SystemC - High-Level Synthesis (HLS) - allocation, scheduling, binding, resource sharing, pipelining - SoC and IP integration, verification and test.