Codesign of Embedded Systems

Codesign refers to the simultaneous consideration of hardware and software in the design of an embedded system, rather than the traditional approach of specifying the hardware and constraining the software to fit.

This course presents codesign as the codevelopment and coverification of hardware and software through the use of simulation and emulation. This codesign begins with a functional design and ends with a detailed Design. Detailed design includes software generation but not hardware fabrication.

Traditional embedded system design views the design of software as a separate task from the design of hardware. The purpose of hardware/software codesign is to provide an integrated method. The design work starts from a system description that is not biased. This system description is then refined into software and hardware components. Typically the refinement step is done automatically by synthesizing software and VHDL code from the system description.

The decision about which parts go into hardware and which into software (a process called partitioning) is done by estimating the performance of the system on the system model level. The advantage of this is that it makes it possible to fully explore the design space and find an optimal solution to the design problem.

The design process is based on hardware / software codesign and is no longer partitioned by hardware and software disciplines but rather by the levels of abstraction represented in the system, architecture, and detailed design processes.

Curriculum

Day 1

Overview Of Embedded System Design

Unified approach to codesign
Designtasks in hardware/ software
Starting with incomplete specifications
Primary bottlenecks in the design process
How complete is designcapture?

System Architecture

Processors, Coprocessors, ASIPs, DSPs
Memory: Standard,Flash, Cache,Shared
Buses and Communication Devices
I/O and Peripheral Components
Preselection of architecture

- **Assignment 1 -System Behaviour** - Study the given system. Translate the specification into a state transition diagram. What are the essential control functions?

Day 2

Design Process

Requirements definition
Specification
System architecture development
Software development
Hardware development
Interface design
Integration andtest

Embedded Design Case Study 1

Designof a Priority-Queue

- **Assignment 2 -Partitioning Exercise** – In this assignment, you are given a system design. Provide a partition for the hardware, software and interface glue logic. State any assumptions you have made. Give a

rationale for the partition.

Day 3

Cosimulation Techniques

Abstract executionmodels Models of computation Processor, bus and memory models Tools for cosimulation Levels of abstraction: Specificationand implementation

Communication

Communicating processes
Required communication actions
Dataflow analysis
Communication protocol selection
Integration with hardware / software partitioning

Embedded Design Case Study 2

Designof a 32-bit Embedded Computer

- **Assignment 3 - Case Study A** -Read the given case study. Identify the bottlenecks and sources of error. protocols will you use and why?

Day 4

Issues In System Design

Potentialsources of error
Conceptual models for capturing system behaviour
Design estimates
Reuse of components
Techniques for system partitioning and estimation
Synthesis of software and hardware
Earlydetection of design faults

What communication

Embedded Design Case Study 3

Designof a Java-Based Web Camera

- **Assignment 4 -Case Study B** - Read the given case study. Develop a codesign on paper. Identify the critical factors for a successful design.

Training Duration

4 days.

Level

Core.

Who Should Attend?

This course is designed for project managers who wish to get updated on the latest trends and techniques in embedded system codesign. Software and firmware engineers about to embark on embedded systems design may also attend.

Prerequisites

Core Subject: ES101 Systems Thinking is preferred, although not necessary.

Skills Gained

After completing this training, you will be able to:

- **Use the codesign process to design an embedded system** of medium complexity.
- **Determine the mutual influence** of both hardware and software early in the design cycle. Software performance is a key criterion for selecting an architecture.
- **Evaluate the enlarged design space.** The interoperability of tools and automation of codesign early in the architecture process significantly improves the ability to consider designs that may otherwise be ignored.
- **Perform system integration and testing.** This reduces effort and resources devoted to integration and testing because hardware and software come from reuse libraries and are coverified.