Course Profile

SystemC (SysC)

I. CONTENTS

Course Profile

1. Verification using SystemVerilog (VSV)

CVC's *Verification Using SystemVerilog* course gives you an in-depth introduction to the main enhancements that SystemVerilog offers for testbench development, discussing the benefits and issues with the new features. It also demonstrates how verification is more efficiently and effectively done using SystemVerilog constructs. The course explores in depth verification enhancements such as object-oriented design, constraint random generation, and functional coverage.

2. Class Details:

- Duration: 3-days full time (Can be extended up-to 5 days on need basis)
- Prerequisites: Attendees must be familiar with Verilog and ideally, but not essentially, Verilog-2001. No prior knowledge of SystemVerilog is required. If you have queries on these prerequisites, please contact CVC.
- Enrolling for a class: Please refer to Registration section.

3. Trainers Profiles

Srinivasan Venkataramanan, CTO

http://www.linkedin.com/in/svenka3

- Over 18+ years of experience in VLSI Design & Verification
- Designed, verified and lead several multi-million ASICs in image processing, networking and communication domain
- Worked at **Philips, Intel,** and **Synopsys** in various capacities. Co-authored leading books in the Verification domain.
- Presented papers, tutorials in various conferences, publications and avenues.
- Conducted workshops and trainings on PSL, SVA, SV, VMM, E, ABV, CDV and OOP for Verification
- Holds M.Tech in VLSI Design from prestigious IIT, Delhi.

Ajeetha Kumari, CEO & MD

http://www.linkedin.com/in/ajeetha

- Has 17+ years of experience in Verification
- Implemented, architected several verification environments for block & subsystems
- Co-authored leading books in the Verification domain.
- Presented papers, tutorials in various conferences, publications and avenues.
- Has worked with all leading edge simulators and formal verification (Model Checking) tools.
- Conducted workshops and trainings on PSL, SVA, SV, OVM, E, ABV, CDV and OOP for Verification
- Holds M.S.E.E. from prestigious IIT, Madras.

4. Why CVC?

Factor \ Vendor	CVC	XYZ training company	EDA Vendor
Training Delivery	World renowned experts	Part timers, in bet'n job engineers	Tool support Engineer
Focus	Verification	Language	EDA tools
Topics covered	User/Verification perspective	Language perspective	Based on the tools strength
How Recently Updated	Last week	Months Back	As old as language was standardized
Verification Expertise	Yes	Depends on the trainer	No
Can I run labs across tools	Yes	Yes	No
Is Content Tool independent	Yes	No/Yes (Typically only one tool)	No
Global Footprint	Yes	No	Yes
Publications	Yes	No	No
Post training support	Yes	No	No
Online Technical Evaluation	Yes	No	No
Customization	Yes	No	No
Online Blogs	Yes	No	No
Extended Hands on	Yes	No	No
Code review	Yes	No	No
Architecture Review	Yes	No	No
Productivity Tools	Yes	No	No
Cost	Low	<Unknown>	Expensive

5. Our Global Footprint

- Headquartered in Bangalore, India
- India
    - Bangalore, Manipal, Cochin, Chennai
    - Pune, Hyderabad
    - Mumbai, Noida
- USA
    - San Jose/Santa Clara
    - Boston
    - Austin
- Europe
    - Poland
    - Munich (partner)
- Asia-Pac
    - Vietnam
    - Singapore
    - China
    - Taiwan
- Middle East
    - Turkey (partner, in the near future)

6. Other Relevant Courses

- Verification using SystemVerilog (VSV)
- UVM Level 1 (Basic)
- UVM Level 2 (Intermediate)
- UVM Level 3 (Expert)
- UVM RAL
- Art of Debugging with UVM
- ABV-UVM
- Go2UVM
- Graph Based Verification
- Formal Verification

7. Customer list (sub-set)

ALDEC
THE DESIGN VERIFICATION COMPANY
intel
NVIDIA
ANALOG DEVICES
TEXAS INSTRUMENTS
applied micro
Infosys
NOKIA
Infotech
CYIENT
BROADCOM
KARMIC LABS
NXP
TATA
TATA ELXSI LIMITED
Canon
KPIT
Quartics
TATA
TATA CONSULTANCY SERVICES
CISCO
LSI
SAMSUNG
Continental
maxim integrated
SYNOPSYS
Silicon to Software
WIPRO
Applying Thought
EATON
Powering Business Worldwide
MICROCHIP
Synplicity
Simply Better Results

8. Course Content

About SystemC

SystemC addresses the need for a system design and verification language that spans hardware and software. It is a language built in standard C++ by extending the language with the use of class libraries. The language is particularly suited to model system's partitioning, to evaluate and verify the assignment of blocks to either hardware or software implementations, and to architect and measure the interactions between and among functional blocks. Leading companies in the intellectual property (IP), electronic design automation (EDA), semiconductor, electronic systems, and embedded software industries currently use SystemC for architectural exploration, to deliver high-performance hardware blocks at various levels of abstraction and to develop virtual platforms for hardware/software co-design. SystemC was defined by the Open SystemC Initiative (OSCI) and ratified as IEEE Std. 1666™-2011

Why SystemC?

A typical system on a chip (SoC), consists of both silicon and embedded software. Its design involves complex algorithm and architecture development and analysis similar to that performed in system design – a trade-off process that determines critical metrics, such as SOC performance, functionality, and power consumption.

SystemC is a single, unified design and verification language that expresses architectural and other system-level attributes in the form of open-source C++ classes. It enables design and verification at the system level, independent of any detailed hardware and software implementation, as well as enabling co-verification with RTL design. This higher level of abstraction enables considerably faster, more productive architectural trade-off analysis, design, and redesign than is possible at the more detailed RT level. Furthermore, verification of system architecture and other system-level attributes is orders of magnitude faster than that at the pin-accurate, timing-accurate RT level.

The SystemC community consists of a large and growing number of system design companies, semiconductor companies, intellectual property providers, and EDA tool vendors who have joined together to support and promote the standard.

SystemC TLM

In July 2012, transaction-level modeling (TLM) was integrated into SystemC. TLM standard interfaces for SystemC provide an essential framework needed for model exchange within companies and across the IP supply chain for architecture analysis, software development and performance analysis, and hardware verification. It explicitly addresses virtual prototyping in which SystemC models can easily be exchanged and arranged within a system, enabling the optimal reuse of models and modeling effort across different use cases

SystemC Verification

The SystemC Verification (SCV) library provides a common set of APIs that are used as a basis to verification activities with SystemC (generation of values under constraints, transaction recording, etc.). These APIs are implemented in all major SystemC simulators available on the market.

Agenda

Session 1: SystemC Basics

- Introduction
  - ♦ Level of Abstraction.
  - ♦ System level design flow.
  - ♦ Why C++?
  - ♦ Benefits of C/C++ based design flow.
  - ♦ Why not just use C/C++? Drawback of C/C++ based design flow

- What is SystemC?
- Why SystemC for system design?
- How does it work?
- Benefits of SystemC – based design flow.
- What are the SystemC approaches?
- SystemC history
- SystemC environment
- SystemC objectives
- HDL based flow
- SystemC features
- SystemC design Flow:
  - ♦ Developing SystemC – An Overview
  - ♦ Model structure - system level
  - ♦ Model structure – implementation level
  - ♦ Basic system structure
- SystemC language Architecture

Session 2: Class & OOP

- Class & OOP: key features
  - o Encapsulation
  - o Inheritance
  - o Polymorphism
- Module
- Ports & signals
- Clocks

Session 3: Process & Design Activity

- Processes

- Waiting & watching
- Cycle –Accurate Simulation Scheduler
- Design Activities
    - Modeling
        - Module for “zigzag” computation
    - *Simulation*
        - Generation and run of an executable specification
    - Debugging
        - Techniques for checking the functionality of the system

Session 4: SystemC Data Types, Statement & Flow control

- SystemC supports all C++ data types. In addition SystemC provides additional data types for describing hardware.
- Data types explain in two parts:
    - C++ Data types
    - SystemC data Types
- C++ Data types:
    - Fundamental Data Types : Character types, Floating-point m Types, Integer Types, Boolean Types, Void Type.
    - Additional Data types: String Data Types, Enumerated data Types, Typedef data Types.
- SystemC Int bit logic:
    - Fixed-Precision, Arbitrary-Precision, Arbitrary Width Bit Vectors, logic Types
- SystemC Operators:
    - Arithmetic Operator, Increment Decrement Operator, Bitwise Operator, Assignment Operator, Equality Operator
- SystemC statement and flow control:
    - if-else, for-loop, while and do-while loop.
    - jump statement break, continue, goto.

Session 5: SystemC Function, Module & TLM

- Functions, Void functions, function call as Expression, Functions declaration

- Function argument passing
  - Pass by value
  - Pass by reference
  - Const
  - Default argument values
- Modules
- Introduction to TLM-2.0
  - What is TLM?
  - Why is TLM interesting?
  - How is TLM being adopted?
  - Transaction Level Modeling Key Concept.
  - The architecture of TLM-2.0
  - Coding guideline of TLM2.0.
  - How to use in real environment

9. Registration

Send us the following details:

- Name, Email, Contact number of all attendees
- A coordinator name (In case of multiple attendees)
- Training module you are looking for
- Onsite or at CVC premises
- Tentative schedule – month & week (Indicate when your team is available to attend the training)

You may email the details to training@cvcblr.comor

Visit Us: http://www.cvcblr.com or

Call Us: +91- 42134156, +91-9620209223

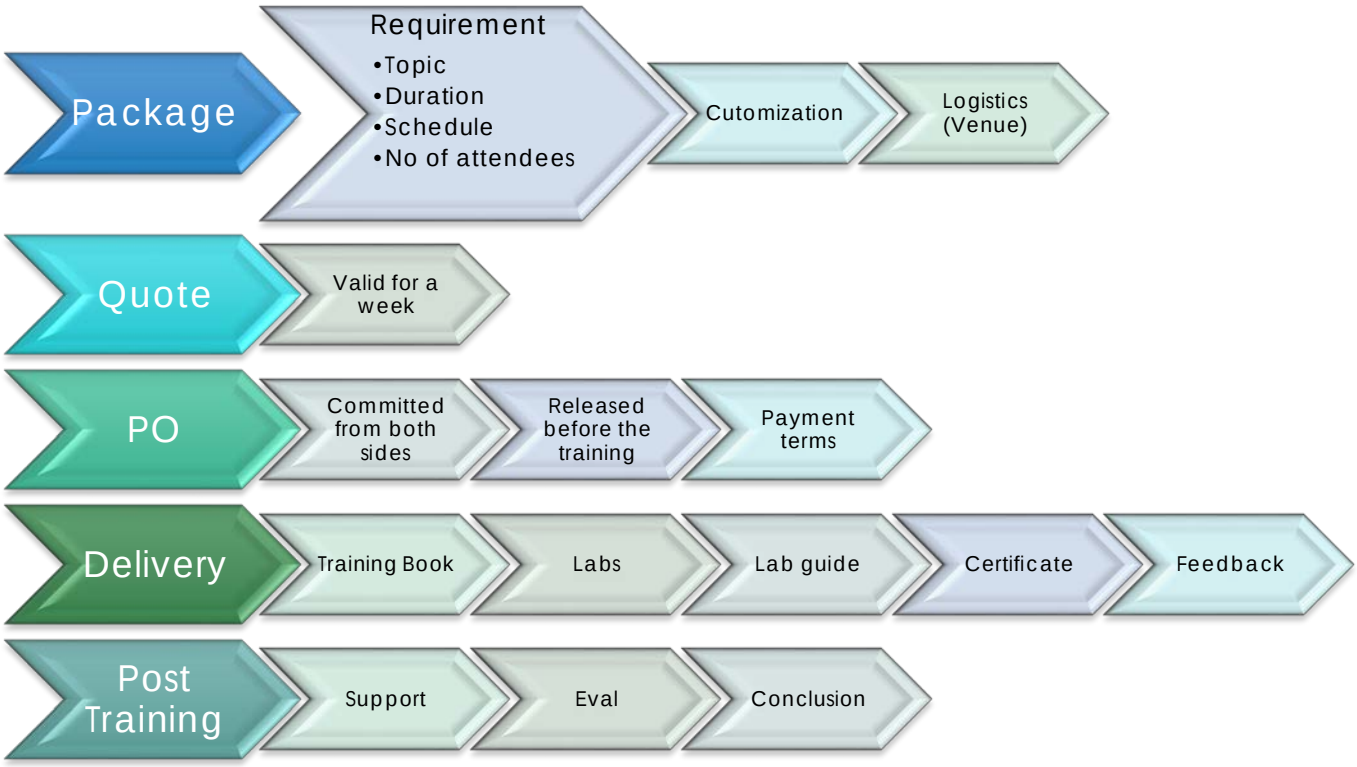
+91-9620209223

https://www.linkedin.com/company/cvc-pvt-ltd


http://www.fb.com/cvc.uvm


http://www.twitter.com/cvcblr

10. Engagement Process



11. More Questions?

Please contact us via email/phone: training@cvcblr.comCall Us: +91-42134156, +91-9620209223