



CVC Pvt. Ltd.

<http://www.cvcblr.com>

Certificate course on Functional Verification

...basics to ASIC verification using SystemVerilog

Dear candidate,

Thanks for your interest in our 5-day certificate course on Functional Verification covering SystemVerilog in depth. Broadly it covers the following topics:

- SystemVerilog basics (SVB)
- Verification Using SystemVerilog (VSV)
- Verification Methodology (VMM)

Duration

Here is a detailed breakdown of the course with duration. Note that we have several “mini projects” tightly embedded in the course that helps in mastering topics learned so far in the course. This is **on top of the regular labs that are part of the training.**

<u>Topic</u>	<u>Duration</u>
SystemVerilog Basics	0.5 day
Verification using SystemVerilog	1.5 days
Project I	0.5 day
Verification Methodology	1.5 days
Project II	1.0 day

Fees

Course	Fees Per Candidate
Certificate course on Functional Verification with SystemVerilog	Rs. 25,000* * for minimum of 10 attendees Plus Service Tax as applicable

#422, VIBHU Complex, 2nd Floor, 27th Main, Sector-I, HSR Layout,
Bangalore – 560102. Ph. No/Fax: 080-42134156, +91-9916176014
<http://www.cvcblr.com>, info@cvcblr.com



CVC Pvt. Ltd.

<http://www.cvcblr.com>

Mode of Payment:

The fee has to be paid in full via either DD or local cheque in favor of “**Contemporary Verification Consultants**” before the training.

We sincerely hope that this quotation fits your requirements and are looking forward to this engagement. Please do not hesitate to call me for any further clarifications. Thanks for your time and interest in our offerings.

Thanks and Regards,
CVC Team

Trainer Profiles

Srinivasan Venkataramanan, CTO

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

- Over 12 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.

**#422, VIBHU Complex, 2nd Floor, 27th Main, Sector-I, HSR Layout,
Bangalore – 560102. Ph. No/Fax: 080-42134156, +91-9916176014
<http://www.cvcblr.com>, info@cvcblr.com**



CVC Pvt. Ltd.

<http://www.cvcblr.com>

Ajeetha Kumari, CEO & MD

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

- Has 8+ 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.

SystemVerilog Basics (SVB)

What is SystemVerilog?

SystemVerilog is a major extension to Verilog-2001, adding significant new features to Verilog for verification, design and synthesis. Enhancements range from simple enhancements to existing constructs, addition of new language constructs to the inclusion of a complete Object-Oriented paradigm features. There are also considerable improvements in the usability of Verilog for RTL design.

**#422, VIBHU Complex, 2nd Floor, 27th Main, Sector-I, HSR Layout,
Bangalore – 560102. Ph. No/Fax: 080-42134156, +91-9916176014
<http://www.cvcblr.com>, info@cvcblr.com**



CVC Pvt. Ltd.

<http://www.cvcblr.com>

Objectives

- To explore the new features of SystemVerilog for design & verification and demonstrate the improvements in RTL design and verification environment efficiency from their use.
- To build the basics for advanced topics on SystemVerilog such as Assertions, testbench etc.

Table of Contents

Session 1: Introduction to SystemVerilog

- Language evolution
- SV Design
- SV Testbench
- SV Assertions
- SV DPI
- SV API

Session 2: Abstract modeling constructs

- Data types, type checking, type cast
- Structure and union
- Enhanced always, case/if... else, loop, flow
- Operators
- Packages
- Arrays and its Operators
- task and function
 - automatic and static
 - void
 - extern
 - Argument pass by value/reference



CVC Pvt. Ltd.

<http://www.cvcblr.com>

Verification Using SystemVerilog

Table of Contents

Interface

- SV Interface
 - Grouping of signals
 - Modport
 - Clocking block
 - Tasks, functions

Classes & OOP

- Class
 - OOP basics
 - *new()* constructor
 - Inheritance
 - Polymorphism
- Virtual Interface

Threads

- Enhanced Concurrency modeling
- Threads – variants of fork .. join
 - ❖ Disable fork, terminate
- Inter process communication
 - ❖ Semaphore
 - ❖ Mailboxes
 - ❖ Queues
- Program block
- Final block



CVC Pvt. Ltd.

<http://www.cvcblr.com>

Constrained random generation

- Random vs. directed testing
- Need for random testing
 - ❖ Constraints in SVTB
 - ❖ Class constraint
 - ❖ Randomize success / fail
 - ❖ Inheritance
 - ❖ Randomize.with()
 - ❖ Distribution
 - ❖ Array constraints
 - ❖ Pre / post randomize

Functional coverage

- Functional coverage
 - ❖ Motivation
 - ❖ Introduction
 - ❖ Types of coverage
 - ❖ Functional coverage process
 - ❖ Covergroup
 - ❖ Coverpoint
 - ❖ Concept of binning
 - ❖ Cross
 - ❖ Sampling event

DPI

- DPI
 - ❖ Import
 - ❖ Export
 - ❖ DPI vs. VPI/PLI



CVC Pvt. Ltd.

<http://www.cvcblr.com>

Do-it Right – Verification using SystemVerilog with VMM (DR-VMM)

What is VMM?

Verification Methodology Manual (VMM) is a framework for Verification using SystemVerilog (SV). VMM provides a mean of doing verification in a well defined and structured way. It is a culmination of well known ideas, thoughts and best practices all bundled as a good text book. It is also supported by a standard set of base classes to help building structured verification environment faster.

What's a Do-it Right course?

Do-it Right is a series of methodology trainings from CVC for those who are familiar with the basics and want to do the verification the “right” way. For instance, SystemVerilog offers a wide variety of features that can be used in many contexts. VMM helps to focus on end goal of achieving structured, reusable verification using these various language features. CVC's Do-it Right courses are intended for engineers with good verification background and familiarity with SystemVerilog as a Verification language and are looking at deploying it in their next project.

Overview

CVC's VMM course gives you an in-depth introduction to the main features that VMM offers to effectively utilize SystemVerilog for Verification. The course breaks down into two modules. Basic VMM module gets the user upto speed on VMM usage. In the advanced session, we delve into details of some of the advanced features of VMM such as various design patterns such as factory, callback etc. Towards the end of the course we touch up on other powerful VMM components such as VMM Scenario generator, broadcaster, scheduler and notification. Detailed usage of these components is dealt in a separate course on “Advanced VMM” from CVC.

Objectives

- To “Get Up To Speed” on VMM base classes and their usage
- To appreciate key concepts behind factory, callback and other design patterns as it applies to verification
- At the end of the course the attendees should be able to create a verification environment for a given DUT from scratch using VMM.

**#422, VIBHU Complex, 2nd Floor, 27th Main, Sector-I, HSR Layout,
Bangalore – 560102. Ph. No/Fax: 080-42134156, +91-9916176014
<http://www.cvcblr.com>, info@cvcblr.com**



CVC Pvt. Ltd.

<http://www.cvcblr.com>

Prerequisites

Delegates must be familiar with Verification features of SystemVerilog. If you need to get started, we suggest you look at CVC's VSV course. If you have queries on these prerequisites, please contact CVC.

Verification using VMM

Table of Contents

Session 1: Basic VMM

- Introduction to SystemVerilog /VMM
 - What is Verification Methodology (VMM)?
 - Why VMM?
- Transaction Based Verification
 - Overview
 - Is it language/tool dependent?
- Classes & OOP - Refresh
- Modeling Transactions
 - Creating Data models – transactions
 - Basic messaging using *VMM_log*
 - Methods
- Modeling Transactors
 - Types of transactors
 - Methods
 - Implementing a BFM using *VMM_xactor*
- Modeling Communication across transactors
 - Creating and using *VMM_channel*
 - Hook –up with BFM
 - Channel methods
- Modeling Environment
 - Structural segment

#422, VIBHU Complex, 2nd Floor, 27th Main, Sector-I, HSR Layout,
Bangalore – 560102. Ph. No/Fax: 080-42134156, +91-9916176014
<http://www.cvcblr.com>, info@cvcblr.com



CVC Pvt. Ltd.

<http://www.cvcblr.com>

- Test Flow segment
- Program and Top level Testbench
 - Hook –up dut
 - Sample verification environment
- Transaction Generator
 - Types of generators
 - Methods
 - Using *VMM_atomic_gen*
 - *Case Study 1*

Session 2: Advanced VMM

- Messaging
 - Log
 - Severity Level
 - Various log macros
- Factory
 - Introduction, requirements
 - Methods
 - Using factory to change the generator output
- Callbacks
 - Introduction, requirements
 - Façade class declaration
 - Adding the callback hook in transactor
 - Populating the callback method
 - Registering callbacks
 - Error Injection example
- Other powerful VMM components
 - Scenario Generator
 - Notifications in VMM
 - Case Study 2