Do-it Right – Universal Verification Methodology UVM (DR-UVM)

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.

What is UVM?

Universal Verification Methodology (UVM) is the industry standard Verification methodology for Verification using SystemVerilog (SV). UVM 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. It is also supported by a standard set of base classes to help building structured verification environment faster. More details about the standard can be found at: http://www.go2uvm.org

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. UVM 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 UVM course gives you an in-depth introduction to the main enhancements that UVM offers, discussing the benefits and issues with the new features and demonstrating how design and verification is more efficient and effective when using SystemVerilog constructs. Basic UVM training gets the user up to speed on UVM usage with which one can start building IP level testbenches with UVM framework. Towards the end of Basic UVM course we touch upon some of the advanced features of UVM such as Virtual Sequencer, TLM port in-depth etc. Detailed usage of these components is dealt in a separate course on “Advanced UVM” from CVC.

**Duration**

Standard version is 3 days. We can also customize it on need basis. Location: CVC premises at HSR Layout, Bangalore or at the customer site, provided infrastructure exists.

**Objectives**

- To “Get Up To Speed” on UVM base classes and their usage
- To appreciate key concepts behind layered testbenches, Driver, Monitor, Agent, Env, basic TLM, Analysis Ports, Scoreboards, factory & configuration mechanism.
- At the end of the course the attendees should be able to create a verification environment for a given DUT from scratch
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, which is a 3-day course. If you have queries on these prerequisites, please contact CVC.

Enrolling for a class

Send an email to training@cvcblr.com with details such as:

Name:
Company Name:
Official Email ID:
Contact Number:
Preferred dates:

If you are coordinating for an entire team, kindly mention how many attendees are expected.

Verification using UVM

Table of Contents

• Introduction to SystemVerilog /UVM
  o What is Universal Verification Methodology? (UVM)
Why UVM?
- Transaction Based Verification
  - Overview
  - Is it language/tool dependent?
- Classes & OOP - Refresh
- DUT description
- Overall view of UVM base classes, structure (UML diagrams)
- UVM macros – need, basic usage
- Unified messaging in UVM
  - Reporting methods
  - Related macros
  - FILE & LINE number printing
  - Verbosity control from command line
- Modeling Transactions
  - Creating Data models – transactions
  - Modeling Transactions
    (uvm_transaction/uvm_sequence_item)
  - Related macros
- Modeling Components/transactors
  - Types of components
  - Typical component methods (a la UVM Phases)
  - Developing Driver BFM (uvm_driver)
- Transaction Sequencer/Generator (uvm_sequencerer)
- UVM Phases
  - 1.0 phasing (OVM style)
  - 1.1 new phasing
- UVM agents (uvm_agent)
  - Hierarchical grouping
  - ACTIVE/PASSIVE controls
- Modeling Environment (uvm_env)
- Creating test cases (uvm_test)
• Putting it all together
  o Singleton design pattern
  o uvm_top usage
  o print_topology()
  o Global methods in UVM
  o run_test()
• Analysis Ports in UVM
  o Publisher-Subscriber mechanism
  o write() method – single subscriber usage
• Building Monitor (uvm_monitor)
• Scoreboards in UVM
  ▪ uvm_scoreboard
  ▪ uvm_tlm_analysis_fifo usage
• Usage of assertions as checkers
• Functional Coverage integration to UVM
  ▪ uvm_subscriber
  ▪ uvm_tlm_analysis_imp
• Configurations in UVM
  o OVM style, simple set/get config usage
  o Benefits of simple usage
  o Potential issues with simple usage
  o Passing virtual interface via configuration mechanism
  o Preview of advanced uvm_config_db
• Advanced stimulus generation
  o More on Sequencer
  o Sequences – how to define new sequences
  o Means of choosing sequences
• Factory pattern
  ▪ Concept of factory
  ▪ Type based API
  ▪ Global overrides
Instance specific overrides
- TLM ports – theory behind TLM
  - Port, export
- Advanced Messaging features
- Summary, closing remarks
  - Advanced features: virtual sequencer
  - “Day-in-the-life” of a Sequence
  - Config DB usage etc.

Trainer Profiles

Srinivasan Venkataramanan, CTO

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

- Over 16 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, 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, UVM, E, ABV, CDV and OOP for Verification
CVC Pvt. Ltd.

• Holds M.Tech in VLSI Design from prestigious IIT, Delhi.

Ajeetha Kumari, CEO & MD

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

• Has 12+ 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.