

Universal Verification Methodology (UVM)

Course Title

Universal Verification Methodology (UVM)

Course Description

This course presents Universal Verification Methodology (UVM), its various features and benefits for verification.

Chip designs have grown in complexity and more highly integrated. Many chip designs also require integration of design and verification IPs from external vendors. As a result, functional verification has become more challenging.

UVM has been developed to address issues of verification especially for large complex chips. UVM is an industry standard that defines methods to realize modular, scalable, configurable and reusable verification environments.

Target Audience

This course is intended for those who are involved in digital design verification. They include chip design engineers, verification engineers and technical managers.

Prerequisite

SystemVerilog language, in particular SystemVerilog classes

Duration

- 4 days (9 a.m. – 5 p.m.)

Course Presentation

- Each participant will receive training materials and source code files for lab exercises.
- Lecture is 70%, lab exercises is 30%
- The lab exercises are mostly coded and ready to run. Participants will study and understand the code, run them, and review the outputs and waveforms. In some lab exercises, there will be minor changes to run different scenarios
- Each participant will receive a Certificate of Attendance after completing at least 75% of the course

Course Instructor

Chang Chee Keng, Sys-ASIC Designs Pte Ltd

C.K. Chang, B.S.E.E., M.S.E.E., has 30 years of experience in chip design and verification, and has participated in a number of chip development projects. He was a technical staff of AMCC/JNI Corporation (Singapore) and participated in design, verification and architecture of 10Gbit Fibre Channel host adapter ASICs. Prior to that, he founded Digital Media Interconnect Pte Ltd to provide consulting and training services to companies in Singapore and the region. Previous to that, he was a Senior Staff Engineer at Silicon Systems (Singapore) Pte Ltd, a subsidiary of Texas Instruments Inc, and was involved in the design of Firewire storage integrated circuits. Prior to returning to Asia, he worked in Silicon Valley, California, USA, at ESS Technology, 3DO Company and Sun Microsystems in design, verification and architecture of various ASIC development projects.

Contact

Sys-ASIC Designs Pte Ltd
21 Bukit Batok Crescent
#09-79, WCEGA Tower
Singapore 658065
Tel: (65) 6022 1812
Email: ckchang@sys-asic.com
Website: www.sys-asic.com

Course Contents

Chapter 1: Introduction to UVM (Pages 19-36)

History of UVM
SystemVerilog and UVM
Main Goals of UVM
What is Verification Methodology
Methodologies Comparison
Verilog Verification Methodology
Pros and Cons of Verilog Verification
SystemVerilog Verification
Example UVM Environment
UVM Methodology
UVM as Industry Standard

Chapter 2: Brief Review of SystemVerilog Topics (Pages 37-58)

SystemVerilog Class Overview
Creating Class Instances
Multiple Class Instances
Assigning and Copying Class Objects
Class Inheritance
Variable Type and Object Handle
Downcasting with \$cast
Class Containing Other Classes
Virtual Methods
Virtual Classes
Parameterized Classes
Class Randomization Overview
Interfaces Overview

Chapter 3: Basics of UVM (Pages 59-98)

UVM Package
Example UVM Environment
UVM Class Hierarchy
UVM Base Classes

Traditional vs UVM Environment
Example Transactions
Transaction Class Hierarchy
Example User Sequence
Sequence Class Hierarchy
UVM Component Class
Component Class Hierarchy
Component Class Types
UVM Development Overview
User-Defined Tests
UVM Phases
First Look at Test Run
UVM Configuration Object
Printing UVM Topology
Bridging UVM to DUT
Connecting Components to Virtual Interface
Running UVM Test with run_test()
UVM Top-Level Hierarchy
Verification Roles and Partitioning
Chapter 4: UVM Reporting (Pages 99-108)
Reporting Message Overview
Reporting Macros
uvm_info, uvm_warning, uvm_error, uvm_fatal
Macros
Verbosity Settings
Configuring Verbosity Threshold
Actions with Reporting
Configuring Report Actions
Reporting Control on Command Line
Chapter 5: Components and Phases (Pages 109-151)
Component Hierarchy

Universal Verification Methodology (UVM)

Transaction and Config Hierarchy
Component Class Types: `uvm_test`, `uvm_env`,
`uvm_agent`, `uvm_driver`, etc
Usage of Component Types
Component Topology
Component Inheritance and Encapsulation
Components and Dynamic Transaction Objects
UVM Phase Concepts
Common Phases
Run-Time Phases
Top-down and Bottom-up Phases
`build_phase`, `connect_phase` and `run_phase`, etc
Run Phase and Run-Time Phases
Components and Phase Methods
Coding and Executing Phase Methods
`uvm_agent` is `_active` flag
Creating Different Test Scenarios
Chapter 6: TLM Connections (Pages 152-178)
Connecting UVM Components
TLM Connections
TLM Ports and Exports
Initiator and Target Roles
Compare Traditional and TLM
Types of TLM Connections: `get`, `put`, `put`
`broadcast`
Sequencer-Driver Connection: `seq_item_port`,
`seq_item_export`
Analysis Ports
Example Monitor-Scoreboard Connection:
`uvm_analysis_imp`
Broadcast to Multiple Components
UVM Subscriber and Usage
Blocking and Non-Blocking Calls
`get()`, `try_get()`, `put()` and `try_put()`
TLM FIFOs: `uvm_tlm_fifo` and
`uvm_tlm_analysis_fifo`
Chapter 7: UVM Scoreboards (Pages 179-196)
Scoreboard Overview
`uvm_scoreboard` Class
Single-Port Scoreboard
Multi-Port Scoreboard
`uvm_analysis_imp_decl` Macro
Coding Multi-Port Scoreboard

Example Scoreboard Code
Connecting Scoreboard to Monitor
Using Transaction Methods and Queue
Scoreboard with Different Transaction Types
Connecting Scoreboard with Multiple Agents
Using Reference Models
Environment with Scoreboard and Reference
Model
Chapter 8: Transactions (Pages 197-224)
Transactions Overview
Example Transactions
`uvm_sequence_item` Class
Transactions passed between Components
Transactions Decoded by Driver
Generating Interface Signals
Transaction Control Knobs
Methods: `copy()`, `compare()`, `sprint()`, etc
Implementing Transaction Methods
UVM Field Macros
Field Macro Types and Flags
User-defined `do_methods()`
Implementing `do_copy()`, `do_compare()`, etc
Comparing Field Macros with `do_methods()`
Printing Transaction Objects
Example `sprint()` and `convert2string()`
Chapter 9: Sequences (Pages 225-262)
UVM Sequences and Sequencer
UVM Sequence Class Hierarchy
Generating Transactions
`start_item()/finish_item()`
Using Sequence Macros
Comparing Transaction Generation Methods
Sequencer-Driver Interaction
Starting a Sequence
Starting sequence using `start()`
Starting sequence using `uvm_config_db()`
Using `p_sequencer` Variable
Basic Sequence
Nested Sequence
Concurrent Sequence with `fork-join`
Alternative Nested Sequence

Alternative Concurrent Sequence
Sequence with Multiple Sequencers
Variations of Starting Sequences
Virtual Sequence without Sequencer
Virtual Sequence with Sequencer
Returning Response to Sequence
Sequencer Arbitration Modes
Sequence Priority Settings
Sequence Arbitration Examples
lock(), unlock(), grab() and ungrab()

Chapter 10: UVM Tests (Pages 263-278)

UVM Tests and Hierarchy
Printing UVM Topology
Running UVM Tests
Test Component Usages
Command Line Interface
Phase Objections
raise_objection() and drop_objection()
Objection by Component
Objection by Sequence
Drain Time and Timeout

Chapter 11: UVM Factory (Pages 279-305)

UVM Factory Overview
Limitations without Factory
Factory Usage Overview
Registering Components
Registering Sequences and Transactions
Factory Registration Macros
Creating Components
Creating Sequences and Transactions
Factory Override Overview
Factory Override Methods
Override by Type of Transactions
Override by Type of Components
Override by Instance of Transactions
Override by Instance of Component
Factory Override Ordering

Chapter 12: Configuration Database (Pages 306-334)

Configuration Database Overview
uvm_config_db
Methods set() and get()

Configuring Component Counts
Configuring Virtual Interface
Configuration Object Overview
Example Configuration Object
Configuring Testbench
Configuration Object Hierarchy
Example Configuration Hierarchy
Configuring Sequences
Configuration and Resource Database
uvm_config_db and uvm_resource_db
Choosing Access Methods
Sharing and Accessing Same Database

Chapter 13: Register Layer (Pages 335-382)

DUT Registers
UVM Register Model and Usages
Register Model Representation
Frontdoor and Backdoor Access
Structure of Register Layer
uvm_reg_block uvm_reg, uvm_reg_field, etc.
Register Layer Class Hierarchy
RAL Auto-Generation
Field Access Policies
Instantiating Register Model
Adapter uvm_reg_adapter
Generic Register Transaction Type:
uvm_reg_bus_op
Predictor uvm_reg_predictor
Explicit and Implicit Prediction
HDL Backdoor Access Path
Register Access API Methods: write(), read(),
set(), update(), etc
Desired and Mirrored Values
Updating Models and Generating Bus
Transactions
Using Register Model in Sequences
Creating Reusable Regmodel Sequences
Concurrent Sequence with Regmodel
Pre-defined Regmodel Sequences

Appendix A: Further Sequences (Pages 383-396)

Sequencer Specified Per Item
Mixed Sequences and Data Items
Return Response to Sequence

Universal Verification Methodology (UVM)

item_done(), set_id_info(), get_response(),
put_response()

Sequencer Response Queue

Sequencer-Driver Lock-Step Interaction

Sequencer-Driver Pipelined Interaction

Variations of Sequencer-Driver Interactions

Appendix B: Policy Classes (Pages 397-406)

Policy Classes Overview

Printer Policy Class

uvm_default_printer and Other Printing Policies

Compare Policy Class

Pack and Unpack Policy Class

Recorder Policy Class

Appendix C: Synchronization and Container Classes (Pages 407-412)

uvm_event, uvm_barrier, uvm_heartbeat

uvm_queue, uvm_pool

Course Outline

Day 1:

- Chapter 1: Introduction to UVM
- Chapter 2: Brief Review of SystemVerilog Topics
- Chapter 3: Basics of UVM
- Chapter 4: UVM Reporting
- Chapter 5: Components and Phases (partial)
- Lab Exercises: Labs 1-5

Day 2:

- Chapter 5: Components and Phases (remaining)
- Chapter 6: TLM Connections
- Chapter 7: UVM Scoreboards
- Chapter 8: Transactions
- Chapter 9: Sequences (partial)
- Lab Exercises: Labs 6-10

Day 3:

- Chapter 9: Sequences (remaining)
- Chapter 10: UVM Tests
- Chapter 11: UVM Factory
- Lab Exercises: Labs 11-17, Lab 21
- Appendices A-C

Day 4:

- Chapter 12: Configuration Database
- Chapter 13: Register Layer
- Lab Exercises: Labs 18-20, Labs 22-27