SystemVerilog Assertions

Language and Methodology

Training Agenda

(1 Day Class with 6 LABs)

Ashok B. Mehta

DefineView Consulting

ashok_mehta@yahoo.com

(408) 309-1556

Training :: Abstract

- **What is SVA (SystemVerilog Assertions)?**
  - SystemVerilog Assertions (SVA) is a powerful subset of the IEEE 1800 SystemVerilog standard.
  - Its hardware oriented concurrent semantics allow for intuitive development of complex multi-clock domain assertions to catch those elusive bugs at the source.
  - SVA works both with VHDL and Verilog. Clean separation between RTL design and SVA assertions.
  - Parameterized reusability.

- **Course Highlights**
  - Each operator/feature is explained in detail using comprehensive examples, timing diagrams and simulation logs.
  - Real life applications are discussed to put it all in perspective.
  - A reference grade handout book is provided to the class. It has comprehensive detail on each page that can serve as excellent reference material for future.
  - Labs are geared to solidify understanding of key concepts using application oriented designs.

Training :: Agenda

- **Introduction to Assertions**
  - What's an assertion? Why can't I just use Verilog?
  - Advantages of Assertion Based Verification (ABV) .
  - Assertion Based Verification (ABV) Methodology components
- **System Verilog Assertions :: Syntax and Semantics (with applications)**
  - Immediate assertions
  - Concurrent assertions - Basics
    - clocking basics; formal arguments; severity levels; threads
    - Sequence introduction
    - Property introduction (with/without an implication)
    - Vacuous pass?
    - Binding properties.
    - Threading (*what are the performance implications*?)
  - Sampled value functions (in property/sequence and procedural)
    - Functions that return Boolean pass/fail: $rose, $fell, $stable
    - Function that return sampled value; $past (with/without gating expr.)
  - Sequence Operators
    - ##m and ##[m:n] clock delay (*SVA allows only fixed delays. So what if you want variable delays??)*
    - [*] and [*m:n] – Consecutive repetition operator
    - [=] and [=m:n] – Non-consecutive repetition operator
    - [->] and [-> m:n] – Goto (non-consecutive) repetition operator
    - Pros/Cons of infinite ($) range
    - 'throughout', 'within', 'intersect', 'first_match'
    - 'and' and 'or' of sequences with/without delay range
    - 'intersect' vs. 'and'

Training :: Agenda (contd.)

- Property operators
  - ‘not’ operator; If … else ; ‘disable iff’
- System functions
  - $onehot, $onehot0, $isunknown, $countones
- Multiple Clocks / Multiply clocked properties and sequences
- Local variables (*one of the most powerful features*...)
  - Pipelined behavior (multiple threads)
- Detecting and using endpoint of a sequence
  - .ended, .matched, .triggered
- ‘expect’, ‘assume’ (for formal verification)
- Asynchronous Assertions
- 2012 features: Strong and Weak sequences, ‘followed by’, ‘always’, ‘eventually’, ‘until’, ‘until_with’, ‘nexttime’, ‘case’, inferred clock and disable, ‘accept_on’, ‘reject_on’
- ‘let’ declaration
- ‘checker’

SVA LABS

- LAB 1: Learn how to 'bind' property module with design module.
  - *Understand vacuous pass and properties with/without implication*
- LAB 2: Enforces how pipelined threads of a property work.
- LAB 3: Synchronous FIFO
  - *A synchronous FIFO design is presented. You will write assertions to check for various FIFO fail conditions.*
  - *FIFO assertions are some of the most useful assertions to write for any design. The assertions developed in this LAB will be directly applicable to your design.*
- LAB 4: Up/Down Counter
  - *A simple UP/DOWN COUNTER design is presented. Counter assertions deployed directly at the source can greatly reduce the time to debug since these assertions will point to the exact cause of a Counter error without the need for extensive back-tracing debug when design fails.*
- LAB 5: Generic Bus Interface Protocol
  - *A simple bus interface is presented. Assertions are written to find bugs on burst mode data and other protocol violations*
- LAB 6: PCI Read Protocol
  - *A simple system with a PCI Master and PCI Target modules designed to do a simple basic PCI Read operation. The LAB shows how to derive and write simple but effective assertions for a PCI type bus.*

CUSTOMER TESTIMONIALS …

"Ashok is a very good instructor – very impressive."

John Reykjalin, President, Grizzly Peak Engineering, Inc.

"The class was excellent, well distributed between the fundamentals and the practical examples. With a little tweak, we can use those example assertions presented right now in our design development!
In addition I would like to thank you for seminar associated text material. The handout (book) is a few levels above anything I have previously seen. The rules and example descriptions cover the associated topic completely."

Thomas Slee, Sr. Electrical Engineer, ASIC Verification Lead, Space System Loral

"The seminar on SVA was very educative and informative. The material was in-depth, was from a hardware design/verification person's perspective and it was vendor neutral. The information was very good and I hope to have a chance to use it in the future."

Shubha Umesh, Senior Logic Engineer, LeCroy Corporation

"Ashok, I take this opportunity to personally thank you for your dedication. You have very good knowledge on the subject. I intend to now use assertions heavily on my next verification project and will use your examples extensively. This class helped me strengthen my knowledge on SVA."

Mohammad Ashraf, Sr. Electrical Engineer, Space System LORAL

Ashok Mehta – 19 Issued US Patents

Subject matter of the patents is Design Verification of SoC, 2.5D IC and 3DIC (i.e. verifying stacked dies). Also, progressive reusable refinement of Verification from Algorithm to RTL level. RTL ⇔TLM2.0 ESL Models cosimulation.

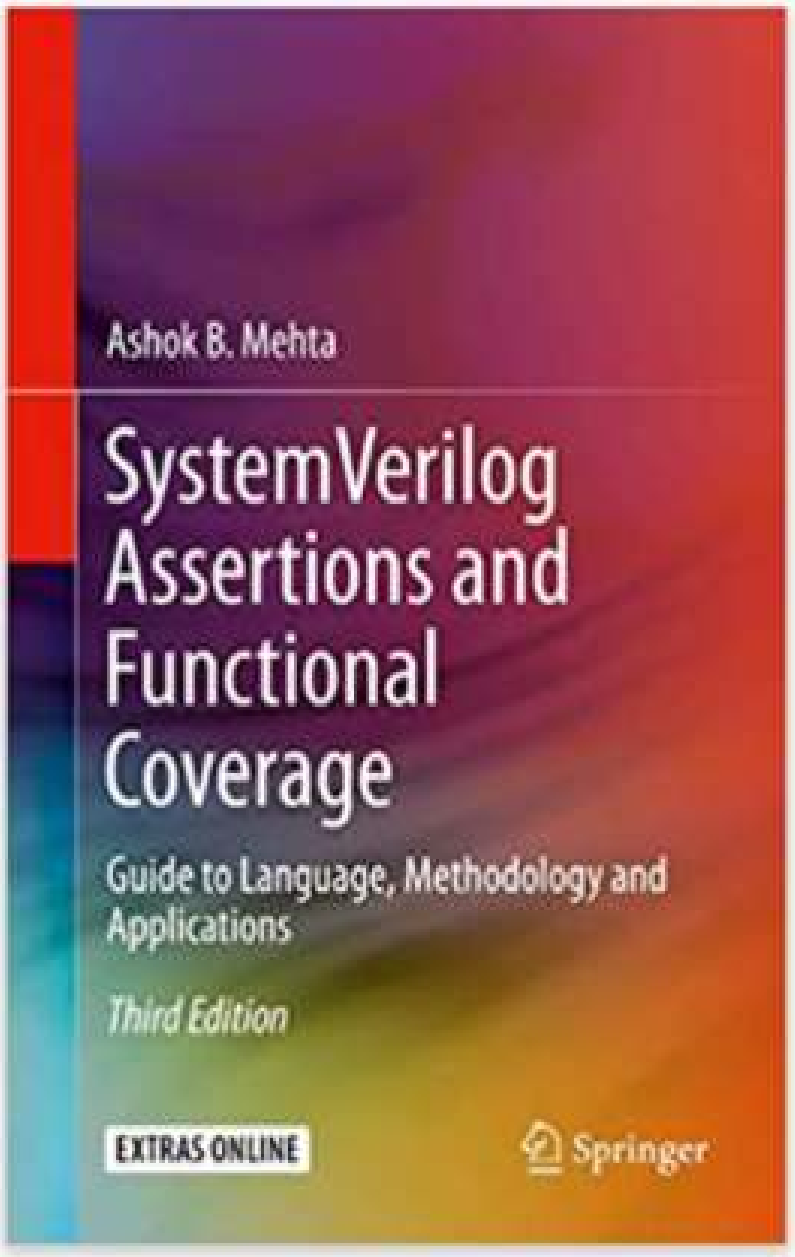
	PAT. NO.		Title
1	10,440,281	T	Image processing apparatus on integrated circuit and method thereof
2	10,061,374	T	Dynamic frequency scaling
3	9,646,128	T	System and method for validating stacked dies by comparing connections
4	9,625,971	T	System and method of adaptive voltage frequency scaling
5	9,612,277	T	System and method for functional verification of multi-die 3D ICs
6	9,552,448	T	Method and apparatus for electronic system model generation
7	9,514,268	T	Interposer defect coverage metric and method to maximize the same
8	9,404,971	T	Circuit and method for monolithic stacked integrated circuit testing
9	9,158,881	T	Interposer defect coverage metric and method to maximize the same
10	9,110,136	T	Circuit and method for monolithic stacked integrated circuit testing
11	9,047,432	T	System and method for validating stacked dies by comparing connections
12	9,015,649	T	Method and apparatus for electronic system model generation
13	8,972,918	T	System and method for functional verification of multi-die 3D ICs
14	8,966,419	T	System and method for testing stacked dies
15	8,826,202	T	Reducing design verification time while maximizing system functional coverage
16	8,578,309	T	Format conversion from value change dump (VCD) to universal verification methodology (UVM)
17	8,522,177	T	Method and apparatus for electronic system function verification at two levels
18	8,402,404	T	Stacked die interconnect validation
19	8,336,009	T	Method and apparatus for electronic system function verification at two levels

Ashok B. Mehta

- **30+ years of experience in SoC, CPU design and verification at DEC, Data General, Intel, Applied Micro, TSMC**

- **Author of three books**
  - SystemVerilog Assertions and Functional Coverage (3rd edition – Springer 2020)
    - A comprehensive guide to languages, methodology and applications
  - Introduction to SystemVerilog :
    - The book covers entire SystemVerilog Language, excluding PLI/DPI, Gate Level and Specify Block
  - ASIC/SoC Functional Design Verification
    - A comprehensive guide to technologies and methodologies

- **19 US Patents on 3DIC and SoC verification**



This book provides a hands-on, application-oriented guide to the language and methodology of both SystemVerilog Assertions and SystemVerilog Functional Coverage.

Readers will benefit from step-by-step approach to functional hardware verification using SystemVerilog Assertions and Functional Coverage.

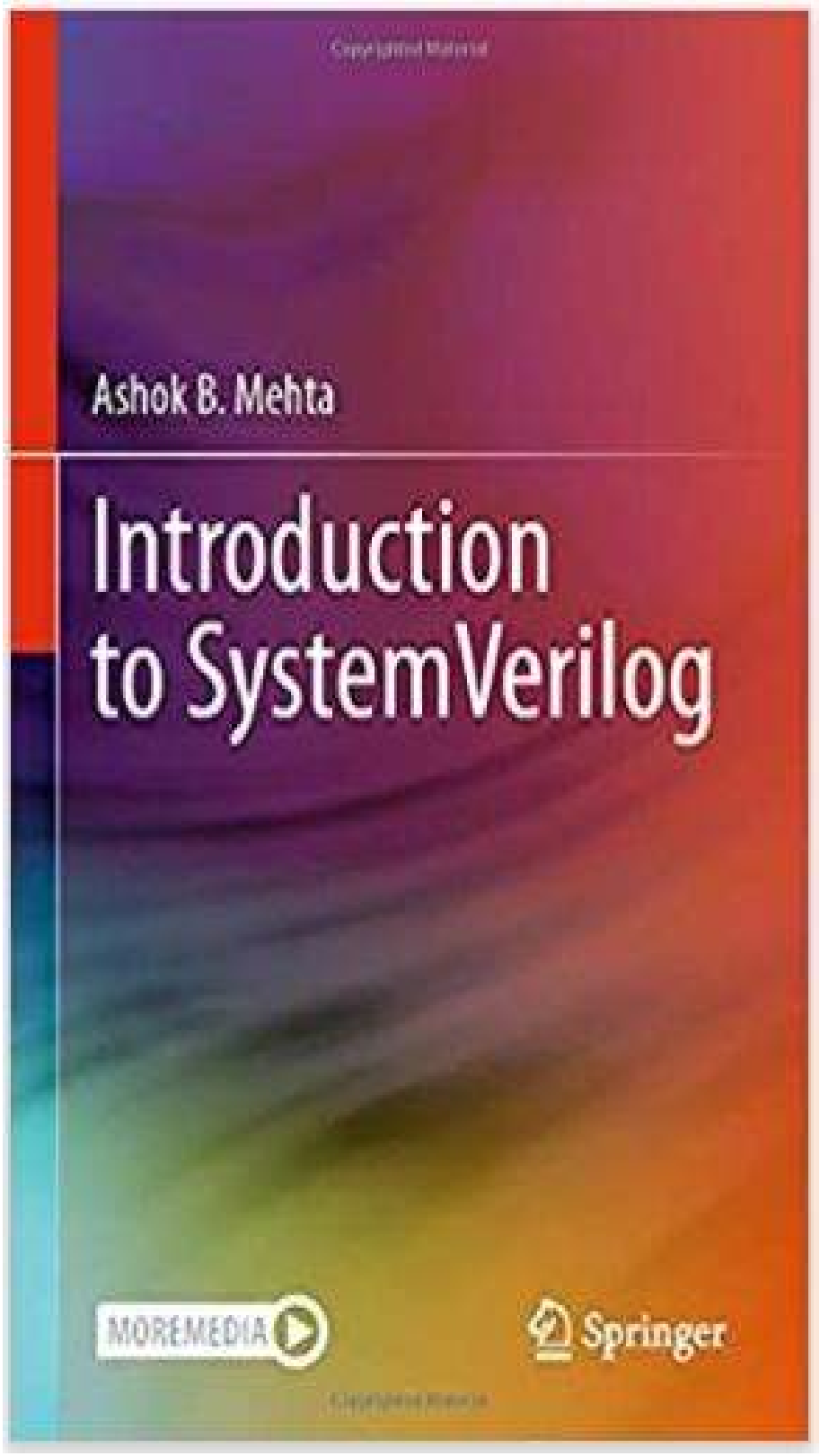
The book has a strong end-user perspective, which makes it plenty easy to digest complex features and apply them with ease to a design.

Plenty of real-life applications

This is an excellent Reference Book

The book will enable design verification engineers to uncover hidden and hard to find bugs, point directly to the source of the bug, provide for a clean and easy way to model complex timing checks and objectively answer the question 'have we functionally verified everything'.

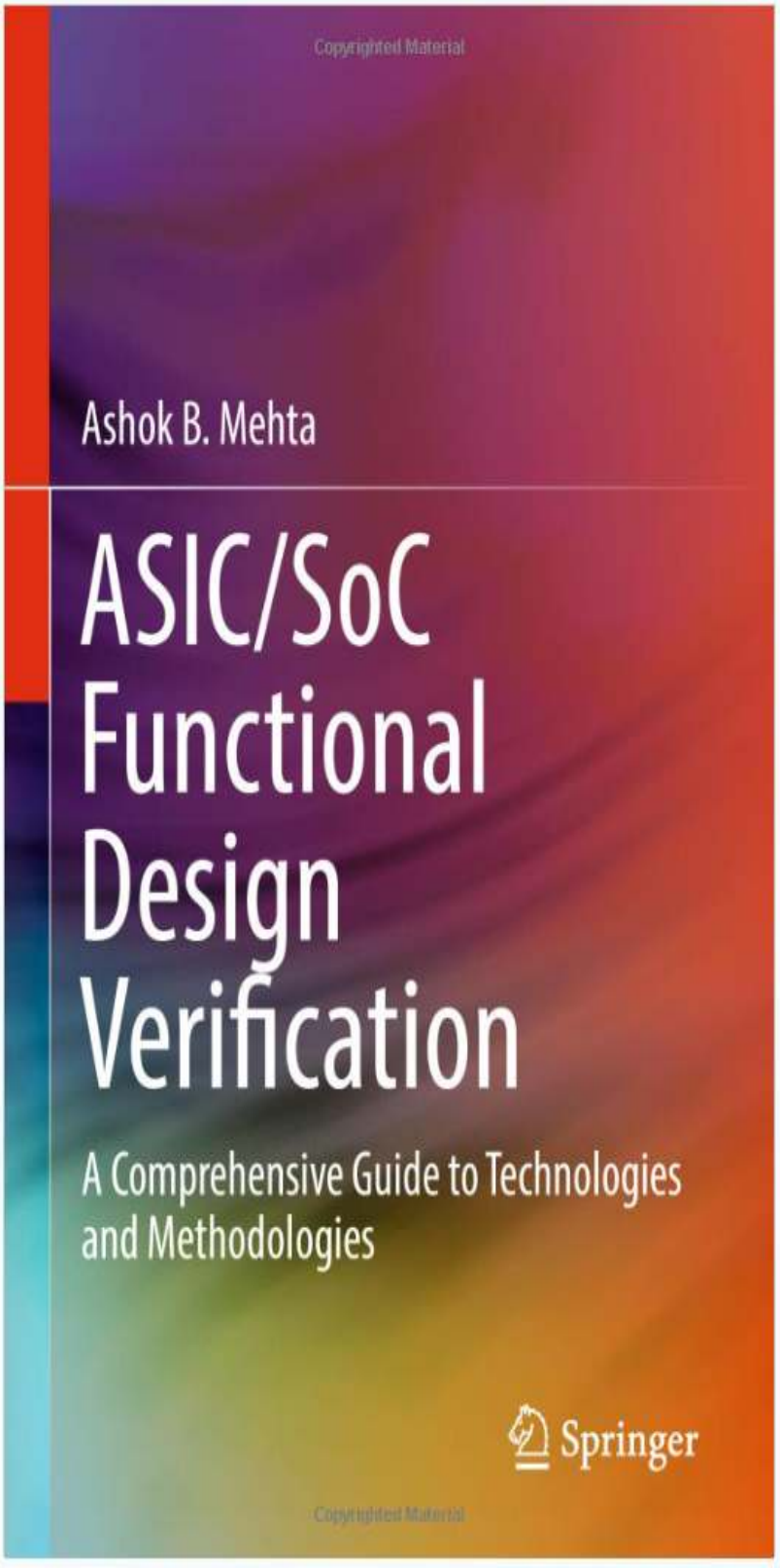
Available on Amazon



A comprehensive book on SystemVerilog

Covers entire language at introductory level. Does not cover PLI/Specify Block.

Available on Amazon.



This book describes in detail all required technologies and methodologies needed to create a comprehensive, functional design verification strategy and environment.

The author describes industry standard technologies such as

UVM (Universal Verification Methodology)

SVA (SystemVerilog Assertions)

SFC (SystemVerilog Functional Coverage)

CDV (Coverage Driven Verification)

Low Power Verification (Unified Power Format UPF)

AMS (Analog Mixed Signal) verification

Virtual Platform TLM2.0/ESL (Electronic System Level) methodology

Static Formal Verification

LEC (Logic Equivalency Check)

Hardware Acceleration

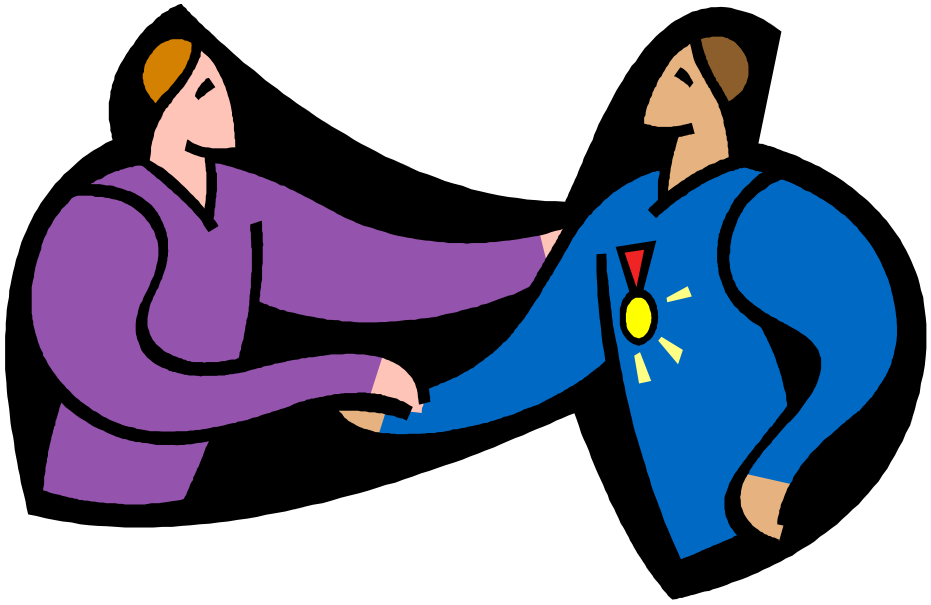
Hardware Emulation, Hardware/Software Co-verification

PPA (Power Performance Area) analysis on a virtual platform

Reuse Methodology from Algorithm/ESL to RTL, and other overall methodologies

Available on Amazon

Watch out Design Bugs



Please visit our web site for further detail

DefineView Consulting
(www.defineview.com)