

Fast-Tracking SVA Through Exposure

**Fast-Tracking SVA through Exposure:
Core Usage, Concepts, AI Integration
2025**

Ben Cohen



**Cohen Publishing
Los Angeles, California
<http://www.SystemVerilog.us/>**

Fast-Tracking SVA through Exposure: Core Usage, AI Integration, & Accelerated Learning 2025

Ben Cohen

Copyright © 2025 by Cohen Publishing

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without the prior written permission from the author, except for the inclusion of brief quotations in a review.

Printed on acid-free paper

Fast-Tracking SVA Through Exposure

1	SVA Concepts and Underlying Model	1
1.1	What is an assertion?	1
1.2	What is a property?	1
1.3	What is SVA?	2
1.4	Simple SVA Example	3
1.4.1	Requirements	3
1.4.2	Why the \$rose?	6
1.5	Understanding the SVA Modeling Machine	7
1.5.1	What is a thread?	7
1.6	points to remember	10
2	Applications requiring Static delays	11
2.1	sequence with many possible matches	11
2.2	Simple example with static range delays	11
2.2.1	Assertions with multiple matches	13
2.2.2	First successful match – YES or NO? []	17
2.3	points to remember	22
3	Applications requiring Static repeats	23
3.1	Applications with fixed repeats	23
3.2	Application with range repeats	25
3.3	Applications with occurrences and skips in repeats	26
3.3.1	Application example, multithreaded consequent	28
3.3.2	Application example, multithreaded antecedent	29
3.4	Applications with occurrences of events	31
3.4.1	Goto: <code>expr [->n]</code> , <code>expr [->m:n]</code> , <code>expr [->m:\$]</code>	31
3.4.2	Non-consecutive: <code>expr [=n]</code> , <code>expr [=m:n]</code> , <code>expr [=m:\$]</code>	32
3.4.3	Application: One and only one occurrence of signal ‘a’ within an occurrence of ‘b’	33
3.4.4	Application: No more than one occurrence of a sequence within an occurrence of ‘b’	34
3.4.5	Application of signal throughout and until another signal	37
3.5	points to remember	39
4	Applications requiring Dynamic Delays/repeats and Data	41
4.1	Applications with a simple delay defined in a register	41
4.1.1	The generate construct	41
4.1.2	Local variables as counters	42

4.2	Applications keeping track of Data	44
4.2.1	Requirements	44
4.2.2	Local Variables in SystemVerilog Assertions for Formal Verification	46
4.3	points to remember	47
5	Multiclocking.....	49
5.1	Verification Handshake Across Different Clock Domains	49
5.1.1	The assertion	50
5.2	The Multiclocking syntax.....	51
5.2.1	Property Syntax.....	53
5.3	points to remember	54
6	Applications Requiring Immediate checks	55
6.1.1	SystemVerilog Event Regions: Understanding Their Role.....	55
6.2	Immediate assertion in logic designs and verification.....	57
6.3	Where Immediate Assertions Are Evaluated	58
6.4	Immediate assertions restrictions, usage, and simulation.....	58
6.5	Application example.....	59
6.6	points to remember:.....	60
7	Disabling a property, abort an assertion	61
7.1	Disabling / killing / aborting Assertions.....	61
7.1.1	Disabling Assertions	61
7.1.2	Disabling through Assertion-Control System Tasks.....	63
7.1.3	Dynamic abort to vacuity or failure	64
7.2	In which region are assertions disabled ?	65
7.3	points to remember	66
8	SVA for Formal Verification	67
8.1	The process.....	67
8.2	Tuning for formal verification.....	68
8.2.1	DETERMINISM.....	68
8.2.2	Local Variable Support in Formal Verification	70
8.2.3	TEMPORAL OPERATOR INTERACTIONS	71
8.3	VACUITY RISKS AND MITIGATION	72
8.3.1	Detection and Mitigation of Vacuity in Formal Verification.....	72
8.4	SCALABILITY AND DESIGN SIZE	72
8.5	POINTS TO REMEMBER	73

Fast-Tracking SVA Through Exposure

9	Support Logic	75
9.1	Why Support Logic	75
9.2	Uniqueness of Attempts Across Concurrent Threads.....	76
	Requirements for Request-Grant Protocol.....	76
9.2.1	Developing the Assertions	76
9.3	Existence of Past Occurrences within Specified Ranges.....	80
9.3.1	Past occurrences of variable expressions	81
9.3.2	Storage of past variable expressions	81
9.3.3	Bounded Time Windows (Finite Range)	82
9.3.4	Unbounded Time Windows (Infinite Range)	82
9.3.5	IF a then b's signature==value.....	83
9.4	Missing clocks.....	83
9.4.1	Gated clock	83
9.5	Memory write/read data integrity	85
9.5.1	Implementation Strategy	85
9.6	points to remember.....	86
10	AI Integration.....	87
10.1	AI and this book	87
10.2	AI experience	88
10.2.1	AI in Requirements.....	88
10.2.2	How I used AI.....	89
10.3	AI Use Process: requirements -> SVA/ RTL/TB	89
10.3.1	Why bother with AI?	89
10.3.2	AI, what is your process?.....	89
10.3.3	AI, How should I talk to you?	91
10.4	Parking Lot Case Study: with Perplexity and chatbotapp	93
10.5	AI Assessment in the Design Process	105
10.5.1	Ben's Case Study: Is AI Effective Today?	106
10.6	Assertion Generation.....	107
10.6.1	Domain-specific knowledge	107
10.6.2	Assertion quality.....	109
10.6.3	Instructions to AI for SVA	110
10.6.4	AI in the workplace	111
10.7	Points to remember	112

Fast-Tracking SVA Through Exposure

FOREWORD

SystemVerilog Assertions (SVA) presents a paradox in verification engineering: while it offers the potential to revolutionize workflows, its complexity hinders widespread adoption. This disconnect highlights the gap between SVA's potential and its actual implementation across the industry.

Ben Cohen's "Exposure2SVA" addresses this challenge through a unique and practical learning approach. Unlike conventional teaching methods, the book mirrors how engineers naturally learn—through hands-on exposure, practical application, and real-world problem-solving. SVA concepts are introduced contextually, aligning with specific verification challenges engineers encounter. Ben has helped me by actively participating in the Verification Academy forums and other platforms like LinkedIn, providing valuable responses to questions in areas I have trouble mastering.

The book takes a notable stance on artificial intelligence in verification. Instead of taking extreme positions, it presents a balanced and pragmatic perspective. AI is viewed as a valuable complementary tool for enhancing requirements development, specification clarity, and code generation. However, it emphasizes the crucial role of engineering judgment and verification rigor. Based on years of collaborative experience, the author's methodology effectively bridges the gap between advanced theoretical concepts and practical implementation. This makes it an invaluable resource for verification teams seeking to enhance their assertion-based verification capabilities.

May your assertions always catch bugs early, *(line generate by ChatGPT)*

Dave Rich
Principal Verification Engineer
Siemens EDA
Technical Chair SystemVerilog IEEE 1800

PREFACE

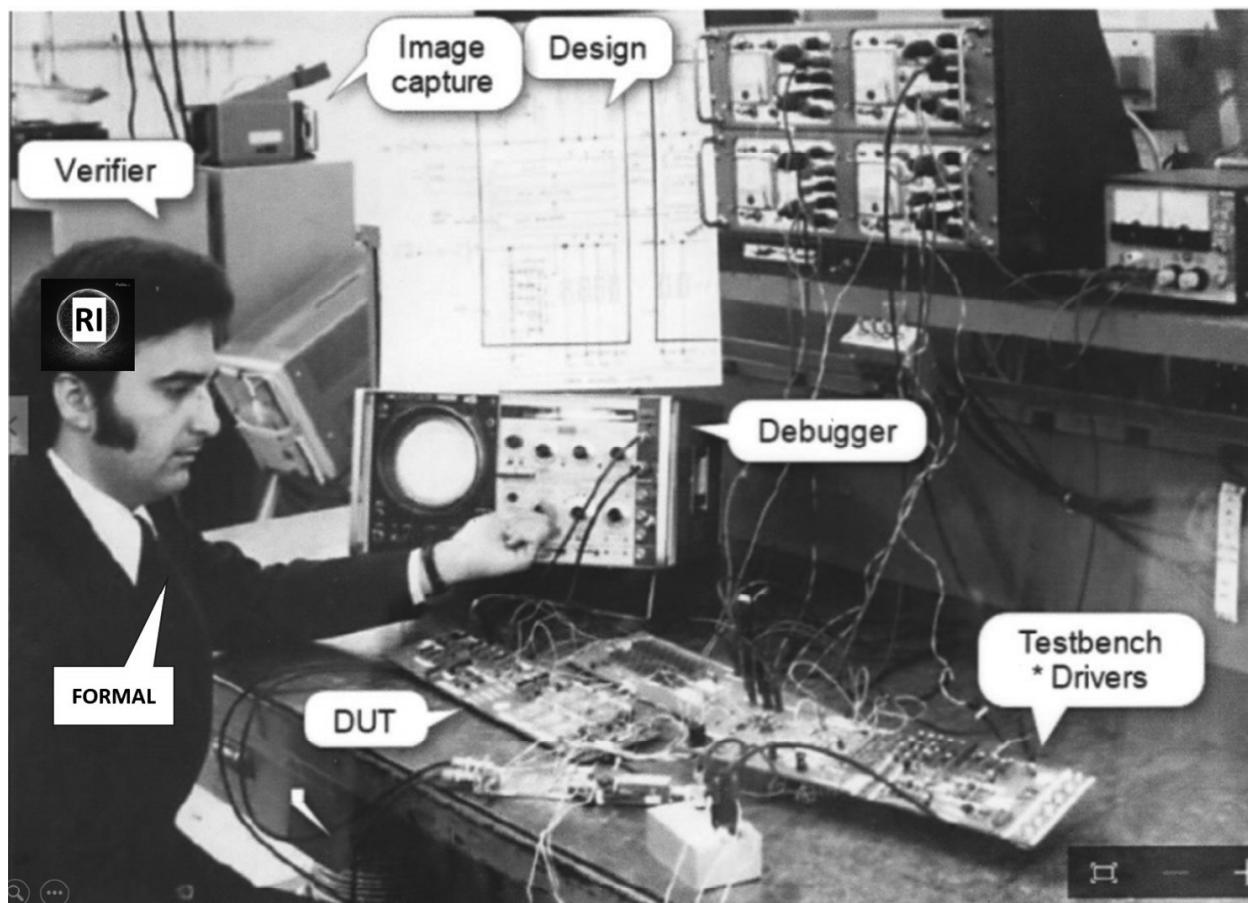
This book presents a revolutionary, application-first approach to mastering SystemVerilog Assertions (SVA), designed for rapid learning and immediate impact in real-world engineering environments. The emphasis is on the essential aspects of SVA: the practical, most frequently used structures necessary for both simulation and formal verification. Additionally, I highlight the growing role of formal verification and the integration of AI throughout the chip design and verification process, particularly in defining requirements and writing code optimized for formal methods.

Unlike traditional texts, this book reverses the usual learning order: it guides you through practical solutions, introducing only the most relevant SVA concepts as they become necessary. My approach is inspired by the natural way children learn: through direct experience and problem-solving, rather than grammar first and then language. Exposure to new concepts and code is how we, as people, truly learn. At first, these concepts, words, or code snippets may seem strange, confusing, or even overwhelming, but with repeated exposure and further reading, they gradually become clear and intuitive. That's why exposure, exposure, exposure is essential. By grounding each chapter in real-world applications, you will see exactly how SVA is used to address the verification problems you encounter every day.

My motivation for writing this book stems from years of experience in chip design and verification, as well as from my interactions with user groups such as those at <https://verificationacademy.com/forums/>. Time and again, I have witnessed engineers struggle with applying SVA to real verification challenges. These experiences even inspired me to write several papers, available at https://systemverilog.us/vf/Cohen_Links_to_papers_books.pdf

Each chapter focuses on real verification challenges. Readers begin with fundamental SVA concepts, such as assertions and the underlying model. Chapters then move to more complex applications, including SV regions, immediate assertions, static and dynamic delays, and repeats. Topics also cover multiclock domains, immediate checks, and SVA for formal verification. The book addresses support logic and strategies for overcoming tool limitations. It also explains how AI is integrated to accelerate defining requirements, specifications, design, and the verification process.

This book is not intended to be an exhaustive reference; that role is best served by resources like my *[SystemVerilog Assertions Handbook, 4th Edition](#)*. Instead, my goal is to equip you with the essential tools and mindset to become an effective verification engineer quickly and confidently.



**Conceptual Methodologies Remain the Same,
But with AI Supplementing RI**
Photo taken at Hughes Aircraft, a long time ago!

ABOUT THE BOOK

Fast-Tracking SVA through Exposure is a modern, application-driven guide to mastering SystemVerilog Assertions (SVA) for digital design verification. The book stands out for its practical, exposure-based teaching philosophy and its forward-looking integration of Artificial Intelligence (AI) into the verification flow.

Approach and Pedagogy:

The book departs from traditional, theory-first texts by adopting an application-first, exposure-driven methodology. Instead of overwhelming readers with exhaustive syntax and grammar, it introduces SVA concepts as they become necessary within real-world verification challenges. This mirrors the way people naturally learn -- through repeated exposure and hands-on problem-solving. The author's goal is to enable engineers to master the core 90% of SVA constructs most relevant to industry practice quickly and confidently, rather than providing an encyclopedic reference.

Content Structure and Core Concepts

Chapters are organized around actual verification problems, systematically building from foundational SVA elements (assertions, properties, sequences) to more advanced applications. Key SVA topics include:

- **Assertion Modeling:** How to represent in raw SV (e.g., *automatic tasks and launches*) cause-effect relationships of SVA's concise syntax, with clear explanations of antecedents and consequents.
- **Temporal Logic:** Deep dives into SVA's temporal operators, static and range delays, repeats, and the importance of determinism versus non-determinism in sequences and assertion design.
- **Advanced Techniques:** Handling vacuity, empty matches, multiclock verification, and dynamic delays using both standard SVA and creative workarounds (e.g., **generate** constructs, local variables).

AI Integration: A Distinctive Feature

A hallmark of this book is its explicit focus on AI's role in the verification process:

- **AI for Requirements and Code Generation:** The author shares concrete experiences using AI tools, including Perplexity PRO, to clarify requirements, translate English descriptions into SVA code, and accelerate the definition of both specifications and verification environments. AI is positioned not just as a futuristic add-on, but as a practical assistant that already streamlines the translation of design intent into formal, verifiable properties.
- **AI in the Design Flow:** The book discusses how AI can automate assertion generation, improve verification coverage, and enhance productivity, especially in formal verification contexts. It reflects on the current effectiveness of AI and anticipates its growing influence in automating routine verification tasks and even in advanced applications such as requirement mining and property synthesis.
- **AI as a Collaborative Partner:** Ben emphasizes the importance of communicating clearly with AI tools, framing questions precisely and iterating on AI-generated outputs-to maximize their utility and reliability in real design environments.

Practicality, Determinism, and Industry Relevance

Throughout, the book stresses:

- **Practical SVA Usage:** Focusing on constructs and techniques that are most commonly used and robust in both simulation and formal verification.
- **Determinism and Clarity:** Encouraging deterministic sequences and assertion writing and explicit constraints to ensure predictable, meaningful verification-crucial for both simulation debugging and formal proof convergence.
- **Integration with Modern Flows:** Recognizing the increasing role of formal verification and AI, the book offers strategies for blending these technologies with traditional simulation to create efficient, future-proof verification methodologies.

Acknowledgements

My extensive interactions with users on forums such as Verification Academy and LinkedIn inspired me to write a book focused on the core knowledge required to use SystemVerilog assertions effectively in real-world environments. By "core knowledge," I refer to the most frequently used techniques and the fundamental information necessary for writing assertions for both design and verification, whether through simulation or formal verification.

During the writing process, I consulted with several expert colleagues who also contributed to reviewing the material. I am grateful to Ed Cerny, Erik Seligman, and Dave Rich for their valuable insights and feedback.

As a Siemens partner, I sincerely thank Siemens for their years of support and for providing simulation tool licenses, including Questa, which have been instrumental in my work.

I would also like to thank DOULOS (<https://www.doulos.com/>) for granting me access to a wide range of simulation tools from various vendors through the EDA Playground platforms (<https://www.edaplayground.com/>). This access has been invaluable for testing SystemVerilog Assertion language interpretations and compatibility across different tools. It is worth noting that, even today, subtle compatibility issues between tools still persist, making such resources especially helpful for thorough verification and validation.

Strangely enough, I also thank Perplexity (perplexity.ai) who, (*treating it as a human*), was an impressive helper: it assisted me in research, brainstorming ideas, rephrased my writing, and created bullet points when needed. Some of the links Perplexity used are listed below. It did not provide direct English quotes from the references. Instead, it summarized, paraphrased, or rewrote the information to address the specific requests, improve clarity, and fit the formatting or style requirements.

Typical links to the references used by Perplexity:

1. ASIC World SystemVerilog Tutorial
<https://www.asic-world.com/systemverilog/tutorial.html>
2. Ben Cohen, *SystemVerilog Assertions Handbook, 4th Edition* (PDF)
https://systemverilog.us/sva4_preface.pdf
3. IBM Research, "Formal Verification at IBM" (PDF)
<https://www.cmi.ac.in/activities/conferences/fm-workshop-jan2013/slides/jason-presentation.pdf>
4. DVCon, "Deadlock Verification for Dummies: The Easy Way of Using SVA and Formal" (PDF)
https://dvcon-proceedings.org/wp-content/uploads/deadlock-verification-for-dummies-the-easy-way-of-using-sva-and-formal_paper.pdf
5. DVCon, "Practical Asynchronous SystemVerilog Assertions"
<https://dvcon-proceedings.org/document/practical-asynchronous-systemverilog-assertions/>
6. Verification Academy Forums
<https://verificationacademy.com/forums/>
7. Ben Cohen, Links to Papers and Books (PDF)
https://systemverilog.us/vf/Cohen_Links_to_papers_books.pdf
8. ASIC World SystemVerilog Assertions Part-I
<https://www.asic-world.com/systemverilog/assertions1.html>
9. DVCon, "Practical Asynchronous SystemVerilog Assertions" (PDF)
<https://dvcon-proceedings.org/wp-content/uploads/1035-1.pdf>
10. ASIC World SystemVerilog Assertions
<https://www.asic-world.com/systemverilog/assertions.html>
11. Doulos, "Practical Asynchronous SystemVerilog Assertions"
<https://www.doulos.com/knowhow/systemverilog/practical-asynchronous-systemverilog-assertions/>

About the Author

My career has spanned digital and analog hardware design and verification, computer architecture, ASIC development, and the application of hardware description languages. I have been an active contributor to the evolution of industry standards, representing Accellera in the IEEE 1800-2012 working group and participating in the development of several editions of the SystemVerilog Language Reference Manual, including the 2005, 2009, 2012, and 2017 versions.

Over the years, I participated in user forums and authored numerous books covering VHDL applications, chip design processes from requirements to verification, RTL design and verification techniques, pragmatic approaches to using VMM, PSL applications, and the “*SystemVerilog Assertions Handbook*,” now in its fourth edition. In addition to these books, I wrote several papers focused on understanding and applying SystemVerilog Assertions (SVA).

These books and papers have become foundational references for engineers and verification professionals worldwide, bridging the gap between theoretical concepts and practical application. My commitment to education is reflected in my extensive teaching of VHDL, PSL, and SVA, as well as in my frequent presentations and tutorials at leading industry conferences such as DVCon and SNUG.

This book, “*Fast-Tracking SVA through Exposure: Core Usage, Concepts, AI Integration*,” draws on my decades of experience to guide readers through the essential principles of SVA, enriched with insights into AI integration for today’s verification challenges.

A comprehensive list of my papers and books, many of which are now freely available, can be found at <https://systemverilog.us/>