Preface

Having been an end user of EDA tools for over 20 years, I have seen that many new technologies stay on way side, because either the engineers do not have time to learn of new technologies/languages or the available material is too complex to digest. A few years back I decided to tackle this problem by creating a very practical, application-oriented down-to-earth SystemVerilog Assertions (SVA) and Functional Coverage (FC) class for professional engineers. The class was well received and I received a lot of feedback on making the class even more useful. That culminated in over 500 slides of class material just on SVA and FC. Many suggested that I had collected enough material for a book. That is how I ended up on this project with the same goal that the reader should understand the concept clearly in an easy and intuitive manner and be able to apply the concepts to real-life applications right away.

The style of the book is such that the concepts are clarified directly in a slide style diagram with talking points. This will hopefully make it easy to use the book as a quick reference as well. Applications immediately following a topic will further clarify the subject matter and my hope is that once you understand the semantics and applications of a given topic, you are ready to apply that to your daily design work. These applications are modeled such that you should be able to use them in your design with minimal modifications.

This book is meant for both design and verification engineers. As a matter of fact, I have devoted a complete section on the reasons and practicality behind having micro-level assertions written by the design engineers and macro-level assertions written by verification engineers. Gone are the days when designers would write RTL and throw it over the wall for the verification engineer to quality check.

The book covers both IEEE 1800-2005 and IEEE 1800-2009 standard SVA language. Even though I have covered all the features of 1800-2009 standard SVA, please note that over 90 % of these features were not supported by EDA tools as of this writing. In other words, the examples belonging to this language subset are not simulated. I would greatly appreciate feedback on 1800-2009 SVA language examples for any errors or omissions.

Chapter 1 is Introduction to SVA and FC giving a brief history of SVA evolution. It also explains how SVA and FC fall under SystemVerilog umbrella to provide a complete assertions and functional coverage driven methodology.

Part I: System Verilog Assertions (SVA)

Chapter 2 goes in-depth on SVA-based methodology providing detail that you can right away use in your project execution. Questions like "How do I know I have added enough assertions?", "What type of assertions should I add", etc. are explained with clarity.

Chapter 3 describes Immediate Assertions. These are nontemporal assertions allowed in procedural code.

Chapter 4 goes into the fundamentals of Concurrent Assertions to set the stage for the rest of the book. How the concurrent multi-threaded semantics work, when and how assertions get evaluated in a simulation time tick, formal arguments, disabling, etc., are described here.

Chapter 5 describes the so-called sampled value functions such as $rose, $fell, $stable, $past etc.

Chapter 6 is the big one! This chapter describes all the operators offered by the language including Clock Delay with and without range, Consecutive repetition with and without range, nonconsecutive repetition with and without range, 'throughout', 'within', 'and', 'or', 'intersect', 'first_match', 'if..else', etc. Each of the operator description is immediately followed by examples and applications to solidify the concept.

Chapter 7 describes the System Functions and Tasks such as $isunknown, $onehot, etc.

Chapter 8 discusses a very important aspect of the language that being properties with multiple clocks. There is not a single design now a day that uses only a single clock. A simple asynchronous FIFO will have a Read Clock and a Write Clock which are asynchronous. Properties need to be written such that check in one clock domain triggers a check in another clock domain. The chapter goes in plenty detail to demystify semantics to write assertions that cross clock domains. The so-called CDC (Clock Domain Crossing) assertions are explained in this chapter.

Chapter 9 is probably the most useful one describing Local Variables. Without this multi-threaded feature many of the assertions would be impossible to write. There are plenty of examples to help you weed through the semantics.

Chapter 10 is on recursive properties. These are rarely used but are very handy when you want to know that a property holds until another becomes true or false.

Chapters 11–13 describe other useful features such as 'expect', 'assume', and detecting end point of a sequence. The ended and matched end-points of sequences are indeed very practical features.

Chapter 14 is entirely devoted to very powerful and practical features that do not quite fit elsewhere. Of main interest here are the examples/testbench for asynchronous fifo checks, concurrent assertions in procedural code, sequence in Verilog 'always' block sensitivity list, and the phenomenon of a 'vacuous pass'!

Chapter 15 is solely devoted to Asynchronous assertions. The example in this chapter shows why you need to be extremely careful in using such assertions.

Chapter 16 is entirely devoted to 1800-2009 features. There are many useful features added by the language designers. Now if only the EDA vendors would get on board and support them!

Chapter 17 describes six LABs for you to try out. The LABs start with simple example moving gradually onto complex ones.

Note The LABs are available on Springer download site extras.springer.com. All required Verilog files, test benches, and run scripts are included for both PC and Linux OS.

Chapter 18 provides answers to the LABs of Chap. 17.

Part II: System Verilog Functional Coverage (FC)

Chapter 19 provides introduction to Functional Coverage and explains differences with Code Coverage.

Chapter 20 is fully devoted to Functional Coverage including in-depth detail on Covergroups, Coverpoints, and Bins including transition and cross coverage.

Chapter 21 provides practical hints to performance implications of coverage methodology. Do not try to cover everything all the time.

Chapter 22 describes Coverage Options, which you may keep in your back pocket as reference material for a rainy day!

http://www.springer.com/978-1-4614-7323-7

SystemVerilog Assertions and Functional Coverage
Guide to Language, Methodology and Applications
Mehta, A.B.
2014, XXXIII, 356 p., Hardcover
ISBN: 978-1-4614-7323-7