Chapter 5

ASSERTION-BASED VERIFICATION

Property Specification

Claudionor Nunes Coelho Jr.
Computer Science Department, Universidade Federal de Minas Gerais, Belo Horizonte, MG - Brazil
celho@dcc.ufmg.br

Harry D. Foster
Jasper Design Automation, Inc, Mountain View, CA - USA
harry@jasper-da.com

Abstract  Assertion-based verification – that is, user-specified properties combined with simulation, formal techniques, and even synthesis – is likely to be the next revolution in hardware design and verification. This chapter explores a verification break-through prompted by multi-level specification and assertion verification techniques. The emerging Accellera PSL formal property language, as well as the Accellera Open Verification Library standards and the important roles they will play in future assertion-based verification flows are discussed.

Keywords: assertion, assumption, constraint, dynamic verification, formal verification, restriction, static verification, property, specification, synthesis

1. Introduction

As formal research matures and approaches a level of sophistication required by industry (beyond the bounds of research and early adopters), we must take steps to ensure a successful transfer (scaling) to this more demanding level. One step is to fundamentally change design methodologies such that we move from ambiguous natural language forms of specification to forms that are mathematically precise and verifiable. Furthermore, these languages must lend themselves to automation. For-
mal property specification is the key ingredient in this methodological change. The end result is higher design quality through:

- **improved understanding of the design space** – resulting from the engineer’s intimate analysis of the requirements, which often uncovers design deficiencies prior to RTL implementation

- **improved communication of design intent** among multiple stakeholders in the design process

- **improved verification quality** through the adoption of assertion-based verification techniques

Although the need for methodological change is clear, transitioning formal verification technology into an industry design environment has been limited by a lack of methodology guidelines for effective use.

Property specification (that is, assertions, constraints, and functional coverage) is fundamental to an assertion-based verification platform. Once specified, properties enable the following components, which may be included in an assertion-based verification platform:

- **verifiable testplans** through property specification (for example, executable functional coverage models, which help answer the question "what functionality has not been exercised?")

- **exhaustive** and **semi-exhaustive** static formal property checking technology (for example, model checking and bounded-model checking)

- **dynamic property checking** technology (for example, monitoring assertions in simulation) for improved observability that reduces the time involved in debug

- **hardware verification languages** (HVLs) for testbench generation that leverage property specification to define expected input (constraints) and output (assertions) behavior

- **constraint-driven stimulus generation** based on interface properties that target block-level designs

- **assertion property synthesis** to address silicon observability challenges during chip bring-up in the lab, as well as operational error detection required for high availability (HA) class systems

Dynamic verification, for the foreseeable future, will remain a critical component of an assertion-based methodology for two reasons: (1) the