

AMBA AHB Assertion-based VIP

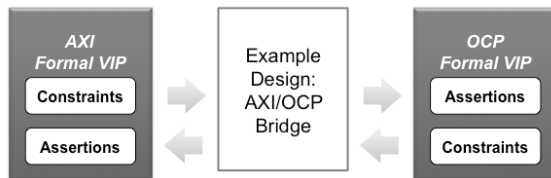
Overview

Formal analysis is a mathematical approach to verification that has the unique ability to prove that a design is 100% correct. This method is tremendously useful, but is limited in the size and types of designs that can be verified. Still, for IP blocks with bus-style interfaces, it is an ideal verification solution.

Cadence® Assertion-Based VIP simplifies formal verification through its plug-and-play approach. Just attach the VIP to your design and run – no need for complicated tests and coverage analysis.

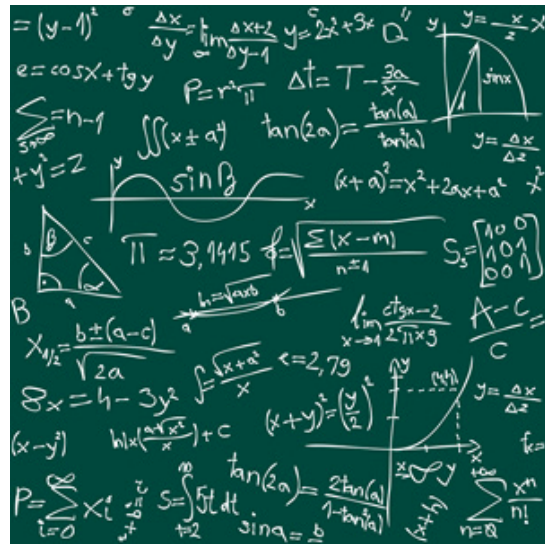
No Tests Required

With Cadence Assertion-Based VIP, no test creation is required. Unlike logic simulation that uses test sequences to stimulate a design, a pre-programmed set of “constraints” is supplied with the assertion-based VIP. These constraints describe the range of all possible stimulus for a given interface specification. Also, “assertions” in the VIP check the interface signaling to make sure your design is acting in accordance with the protocol specification. You can start verifying your design in minutes with our assertion-based VIP.



Debug Made Easy

Cadence Assertion-Based VIP works with the Cadence Incisive® Formal Verifier tool to make debug easy. When the VIP detects a design error, Incisive Formal Verifier displays a waveform trace, schematic view, and source code analysis of the bug. This makes it easy to find the root causes of bugs – and fix them!



Specification Support

The AHB ABVIP supports the AMBA 2, AMBA 3, and AMBA 5 AHB Interface Protocol Specifications.

The APB™ Assertion-Based VIP is included and supports the AMBA 2, AMBA 3, and AMBA 4 APB Interface Protocol Specifications

Product Highlights

- Allows AHB-Lite configuration
- Supports round-robin or user priority based arbitration
- Supports burst reconstruction after split or a retry response, in addition to early burst termination
- Allows master to continue after error response
- Controls maximum length of INCR burst transfer
- Supports monitoring and driving of locked transactions
- Controls the burst type supported by master and the response type supported by slave

Supported Design-Under-Test Configurations

- | | | |
|--|---|-------------------------------------|
| <input checked="" type="checkbox"/> Master | <input checked="" type="checkbox"/> Slave | <input type="checkbox"/> Hub/Switch |
| <input checked="" type="checkbox"/> Full Stack | <input type="checkbox"/> Controller-only | <input type="checkbox"/> PHY-only |