

# An Assertion Studio for SystemVerilog Users

Tarak Parikh

VP of Product Engineering

[www.atHDL.com](http://www.atHDL.com)

# @HDL Version 4.0 Product Families



Finds the toughest design problems by innovative application of Formal Model Checking and Clock Domain

Analysis, Supporting **PSL**, **OVA** and **SystemVerilog** Assertions



Delivers a next generation graphical debugging and design analysis environment to quickly isolate functional errors during creation, model checking, simulation, assertion development and test bench debugging

# Verification Flow using an ABV Methodology

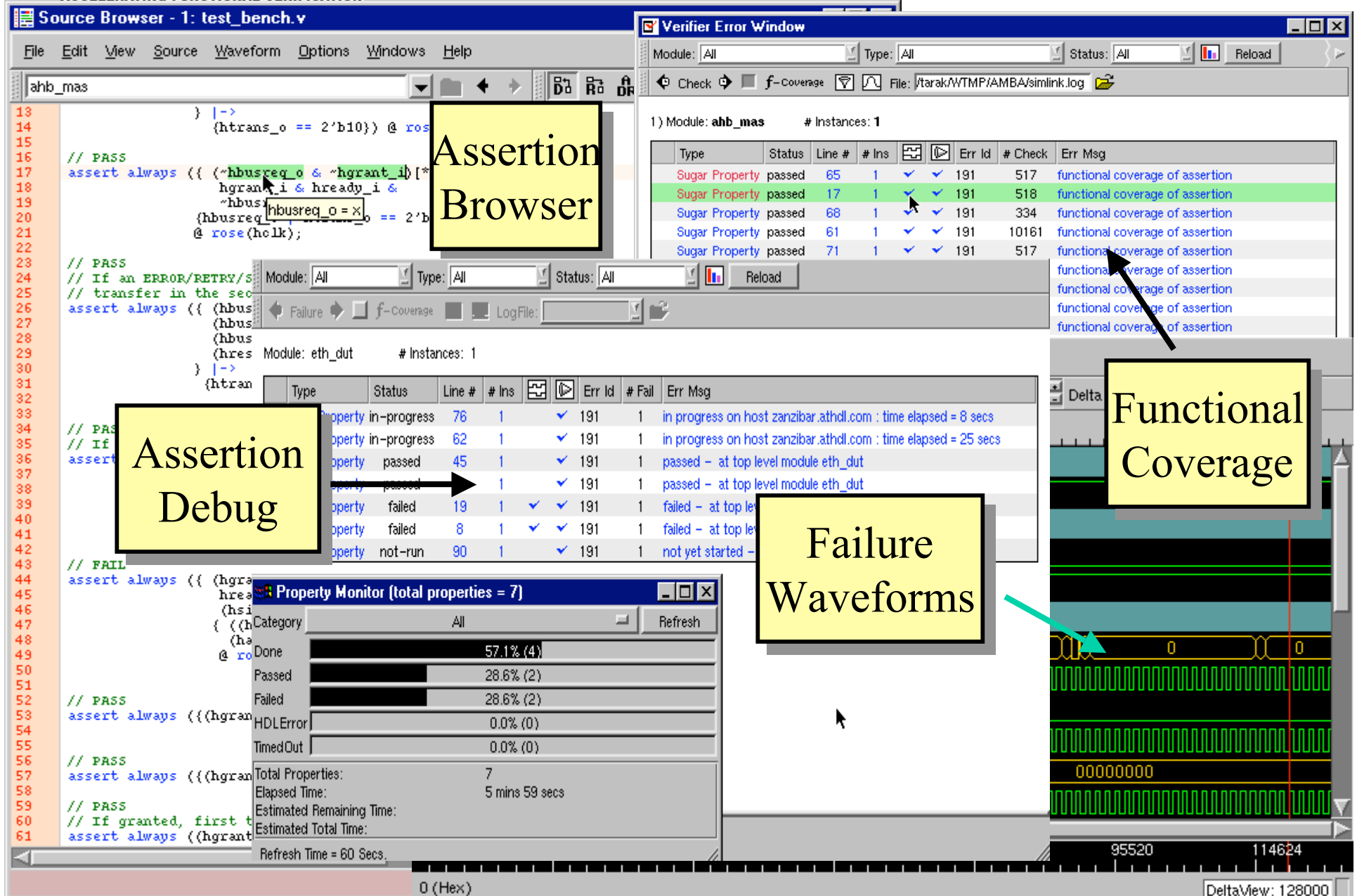
*Specification – Protocols, Interfacing, etc.*



Assertion



ACCELERATING FUNCTIONAL VERIFICATION



The screenshot displays the HDL Verifier software interface with several key components:

- Source Browser - 1: test\_bench.v**: Shows Verilog code for the `ahb_mas` module. A yellow box labeled **Assertion Browser** highlights the assertion logic:
 

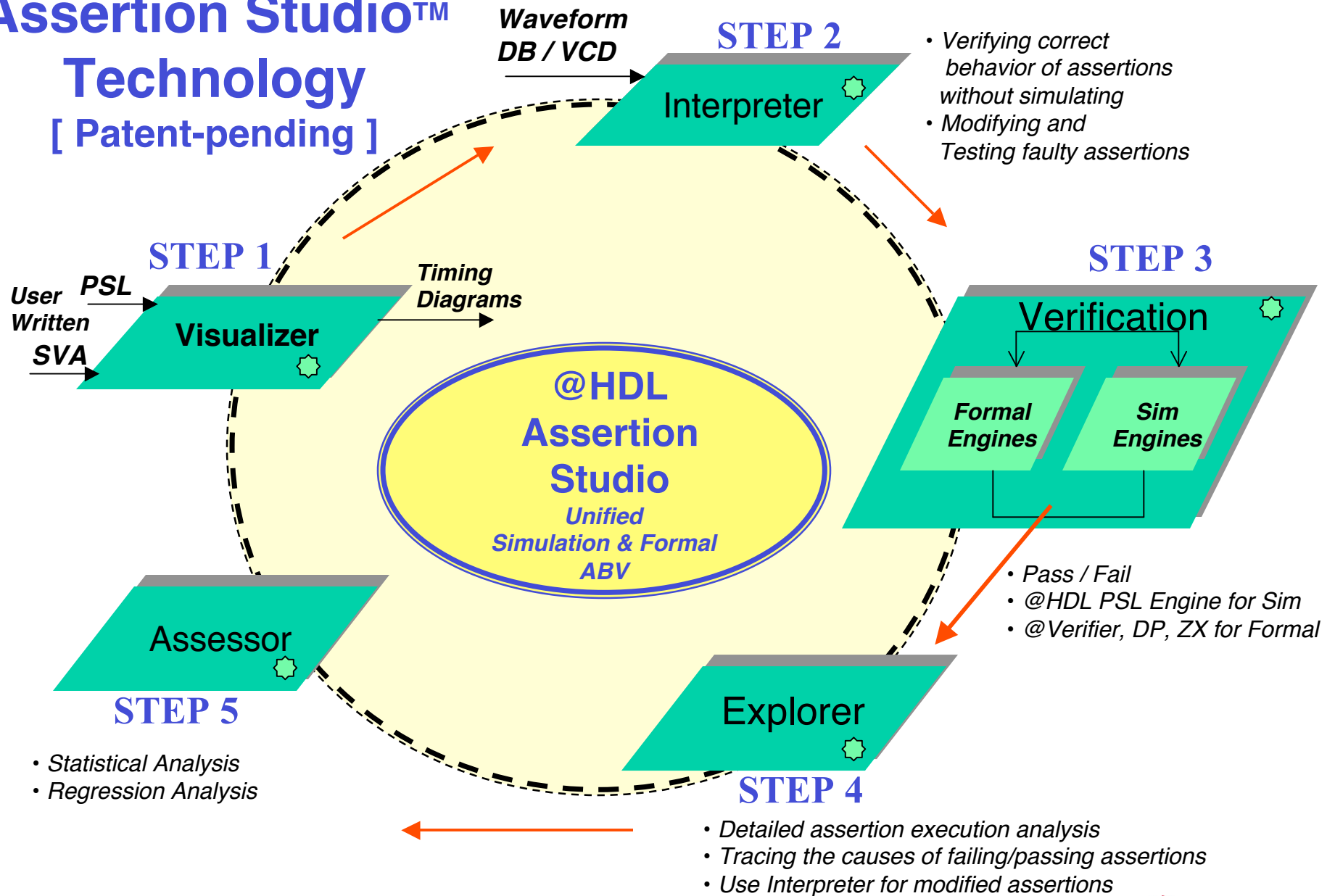
```

      17 assert always (( (~hbusreq_o & ~hgrant_ib) [*
      18 hgrant_i & hready_i &
      19 ~hbusreq_o] == 2'b
      20 @ rose(hclk);
      21
      22
      23 // PASS
      24 // If an ERROR/RETRY/s
      25 // transfer in the sec
      26 assert always (( (hbus
      27 (hbus
      28 (hbus
      29 (hres
      30 ) |->
      31 (htran
      32
      33
      34 // PAS
      35 // If
      36 assert
      37
      38
      39
      40
      41
      42
      43 // FAIL
      44 assert always (( (hgra
      45 hrea
      46 (hsi
      47 { (~h
      48 (ha
      49 @ ro
      50
      51
      52 // PASS
      53 assert always ((hgran
      54
      55
      56 // PASS
      57 assert always ((hgran
      58
      59 // PASS
      60 // If granted, first t
      61 assert always ((hgran
      
```
- Verifier Error Window**: Lists verification results for module `ahb_mas`.
 

Type	Status	Line #	# Ins	Err Id	# Check	Err Msg
Sugar Property	passed	65	1	191	517	functional coverage of assertion
Sugar Property	passed	17	1	191	518	functional coverage of assertion
Sugar Property	passed	68	1	191	334	functional coverage of assertion
Sugar Property	passed	61	1	191	10161	functional coverage of assertion
Sugar Property	passed	71	1	191	517	functional coverage of assertion
- Assertion Debug**: A yellow box pointing to the assertion logic in the source browser.
- Property Monitor (total properties = 7)**: Shows the overall verification statistics:
 

Category	Count	Percentage
Done	4	57.1%
Passed	2	28.6%
Failed	2	28.6%
HDL Error	0	0.0%
Timed Out	0	0.0%
<b>Total Properties</b>	<b>7</b>	
Elapsed Time	5 mins 59 secs	
Estimated Remaining Time		
Estimated Total Time		
- Failure Waveforms**: A yellow box pointing to a waveform showing a failure event. The waveform displays digital signals over time, with a red vertical line indicating the failure point.
- Functional Coverage**: A yellow box pointing to the coverage data in the Verifier Error Window.

# Assertion Studio™ Technology [ Patent-pending ]

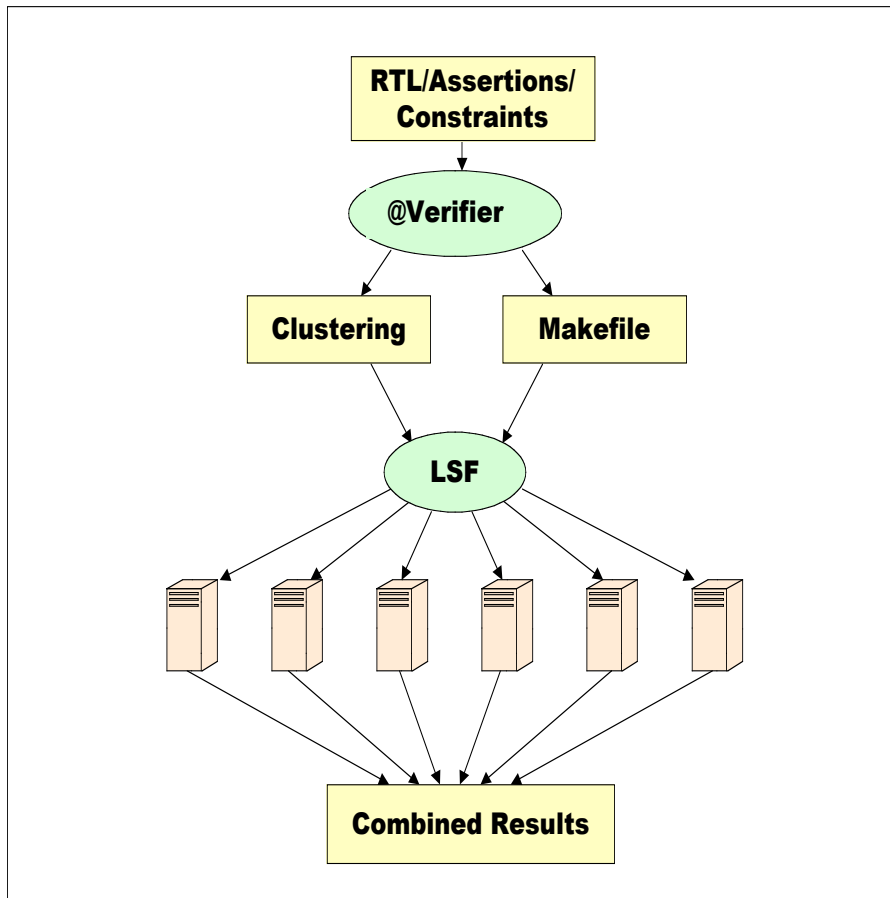


# @HDL Assertion Studio™

## Key Components

1. Timing Diagrams from PSL & SV Assertions
2. Interactive Verification of PSL and SVA
3. PSL Engine for use with non-PSL simulators
4. Assertion Debugging
5. Transaction Display and Debugging
6. Assertion Coverage Reporting from Simulation

# Distributed Processing Support



- Main Allocation Algorithms based on clustering of properties for the same cone of influence
- Automatically Allocate property checking across network resources using load sharing software
- Results in close to linear speed-up

**Assertion Studio™**

**Model  
Checking Debug**

**SV Testbench  
Language  
Debugging**

**Clock Analysis,  
Visualization**



***Find more bugs, faster.***

**SystemVerilog  
Source Code  
Browser**

**Static  
Analysis**

**Waveform Debugging**

**Timing and Hotspot  
Analysis**

# SystemVerilog Support with @HDL

Design	Testbench	Assertions
--------	-----------	------------

 **VERIFIER™**

Q2/04

*n / a*

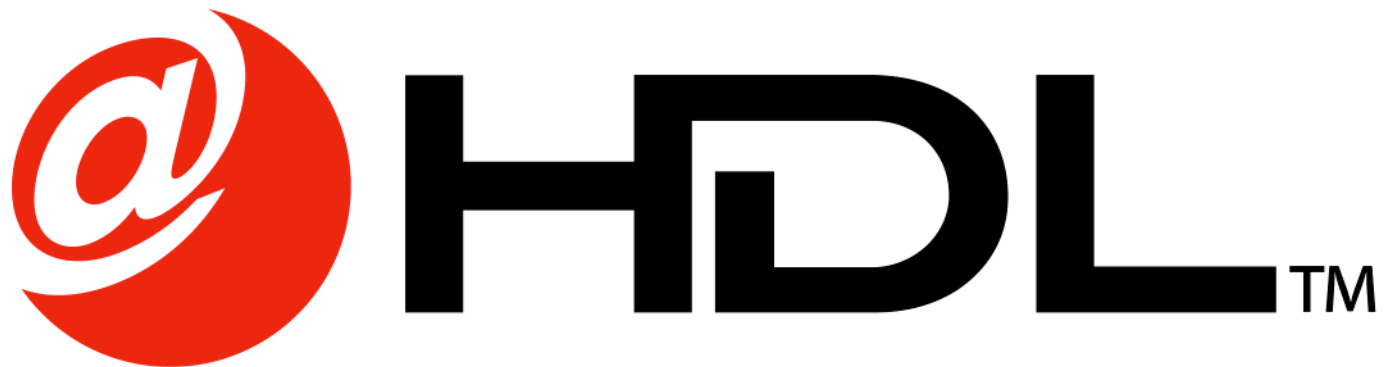
Q1/04

 **DESIGNER™** *PRO*

Q2/04

Q3/04

Q1/04



***ACCELERATING FUNCTIONAL VERIFICATION***

**[www.atHDL.com](http://www.atHDL.com)**

**Come See**

**@Verifier and @Designer-PRO**

**At the Vendor Fair Booth**

## Visualizer

*Automatically convert user-written assertions into  
Timing Diagrams*

### Key Benefits:

- Rapid learning of assertion language
- Documentation for RTL blocks with Assertions
- Ability to look at 3<sup>RD</sup> party IP
- Avoid multiple iterations of costly formal / simulation cycle

# ASSERTION STUDIO

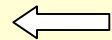
## Interpreter

*Verifies the assertions based on waveform dump (VCD) without running simulation or formal...*

### Key Benefits:

- Early verification of assertions before actual deployment in regression test suites
- Users can add assertions and verify / debug against the VCD
- Avoid multiple iterations of time-consuming formal & simulation
- Users can rapidly iterate on assertion code and validate

**ASSERTION STUDIO**



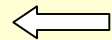


*Delivering industry-leading ABV products,  
addressing the key areas of  
**usability, performance and capacity***

### Key Benefits – Formal Engine:

- Best in class formal solvers (@HDL and IBM Rulebase)
- Distributed Processing
- Incremental and Hierarchical
- Selective Deep-Formal
- PSL and OVA/SVA Support
- Automatic Assertion Extraction
- Multiple Clock Domain Verification

**ASSERTION STUDIO**



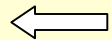
# Explorer

*Temporal decomposition of the assertion into sub-expressions to assist in precisely isolating the failing portion of the assertion*

Key Benefits:

- Rapid isolation of the failure
- Easy to use and debug
- Common Debugging Environment covering Formal and Simulation ABV

**ASSERTION STUDIO**



# Assessor

*Allows users to perform coverage analysis for assertions, as well as transactions*

## Key Benefits:

- Assertion coverage
- Utilize same language to determine transaction coverage
- Provides detailed view when the assertion was covered during simulation
- Creates a functional coverage matrix
- Same language used to do temporal search while debugging

**ASSERTION STUDIO**

