

## Open Verification Session Interoperability Developers' Forum

**Kevin Kranen**  
**Director, Strategic Programs**  
**Strategic Market Development**  
**Synopsys, Inc.**



October 16, 2003

 *Your Interoperability Partner*

## Open Verification Agenda

- Introduction
- Leveraging OpenVera into SystemVerilog, Sashi Obilisetty, VeriEZ Solutions
- SystemVerilog: A Synthesis Perspective, Karen Pieper, Synopsys
- EDA & SystemVerilog, Dennis Brophy, MTI
- EDA & SystemVerilog, Steve Wang, Axis Systems
- Assertions Update for SystemVerilog, Bruce Greene, Synopsys

## Two things to remember:

- Full Speed ahead with SystemVerilog !
- You are safe and secure with your other previous standards investments !

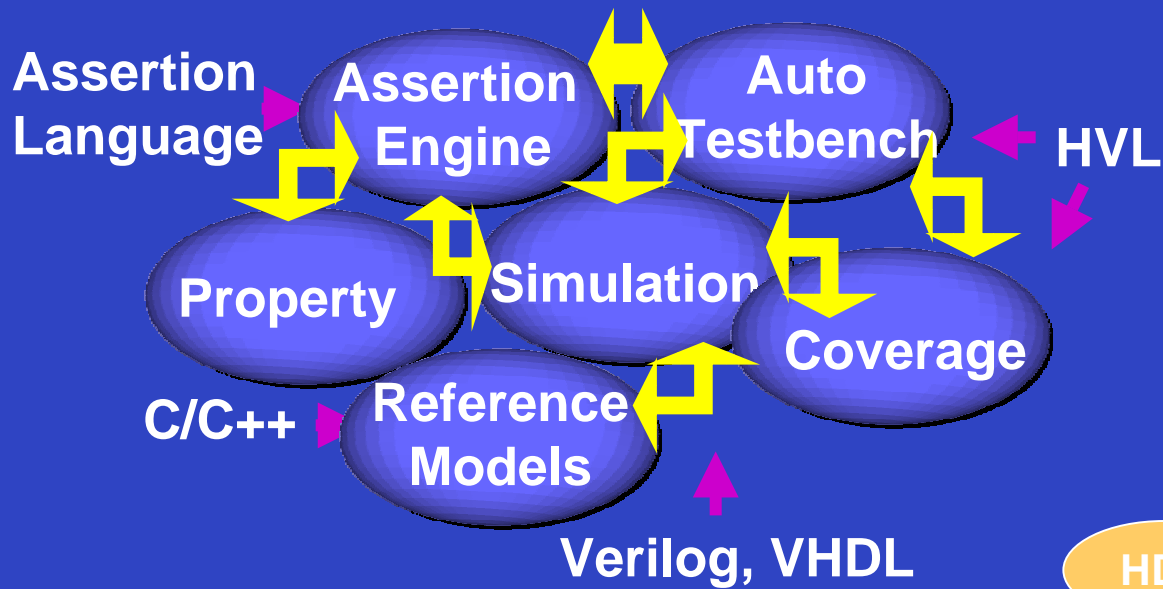


➤ *Your Interoperability Partner*

# Synopsys Delivering SystemVerilog to Designers Today!

- SystemVerilog is THE HDVL – Hardware Definition and Verification Language
- Synopsys Products support SystemVerilog
  - VCS, HDL Compiler, Design Compiler, Leda
  - Formality and additional products coming soon
  - Accelerated assertion support driven by market needs
- SystemVerilog Catalyst – Over 40 companies
  - Enabling all EDA and IP providers to adopt faster
- SystemVerilog NOW! – Market Education
  - Sponsored by 5 EDA Verification Leaders
  - Over 1,000 user registered to attend

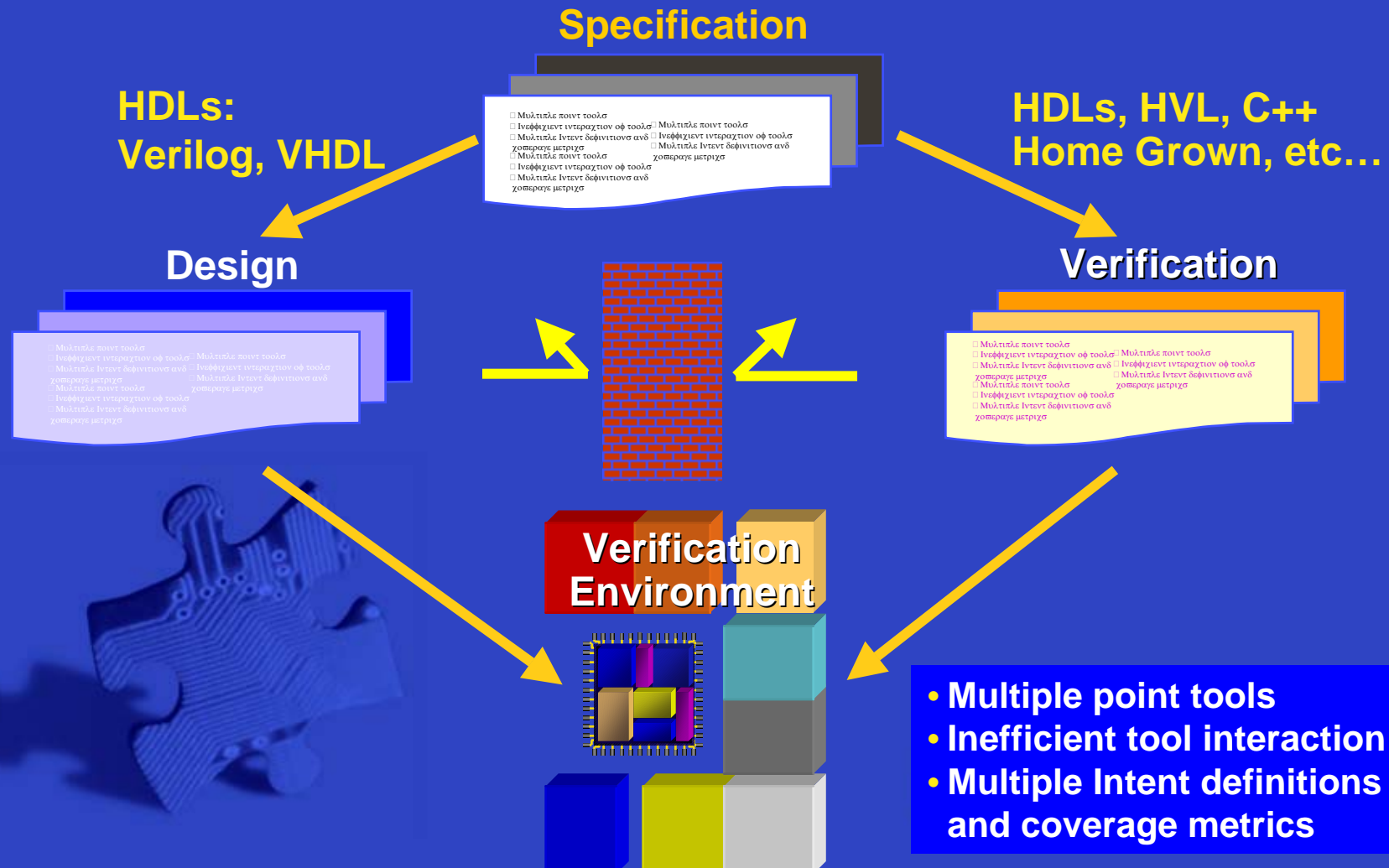
## Language Fragmentation Has Led to Pain



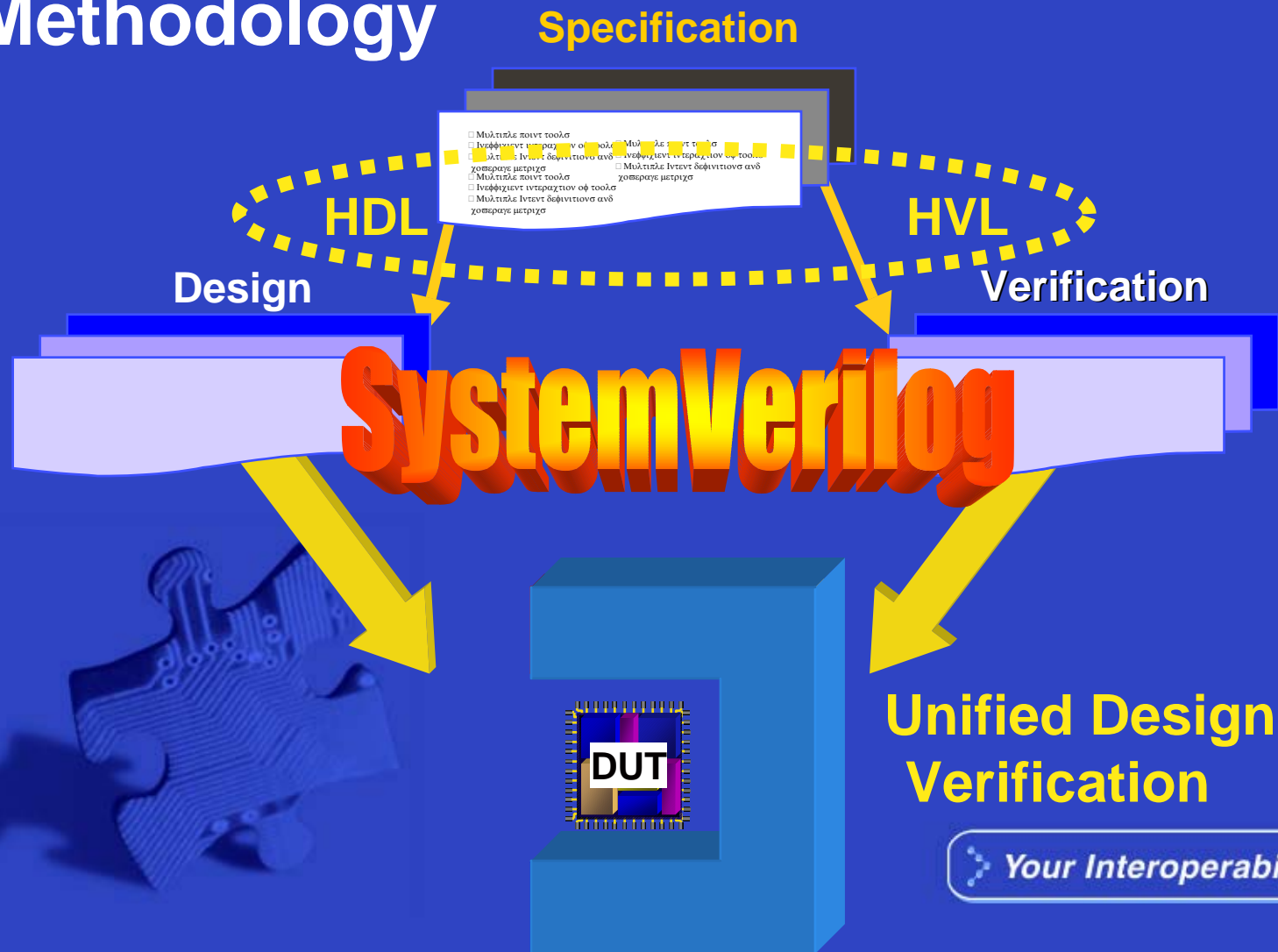
- Painful for design and verification engineers
- Painful for EDA and IP developers



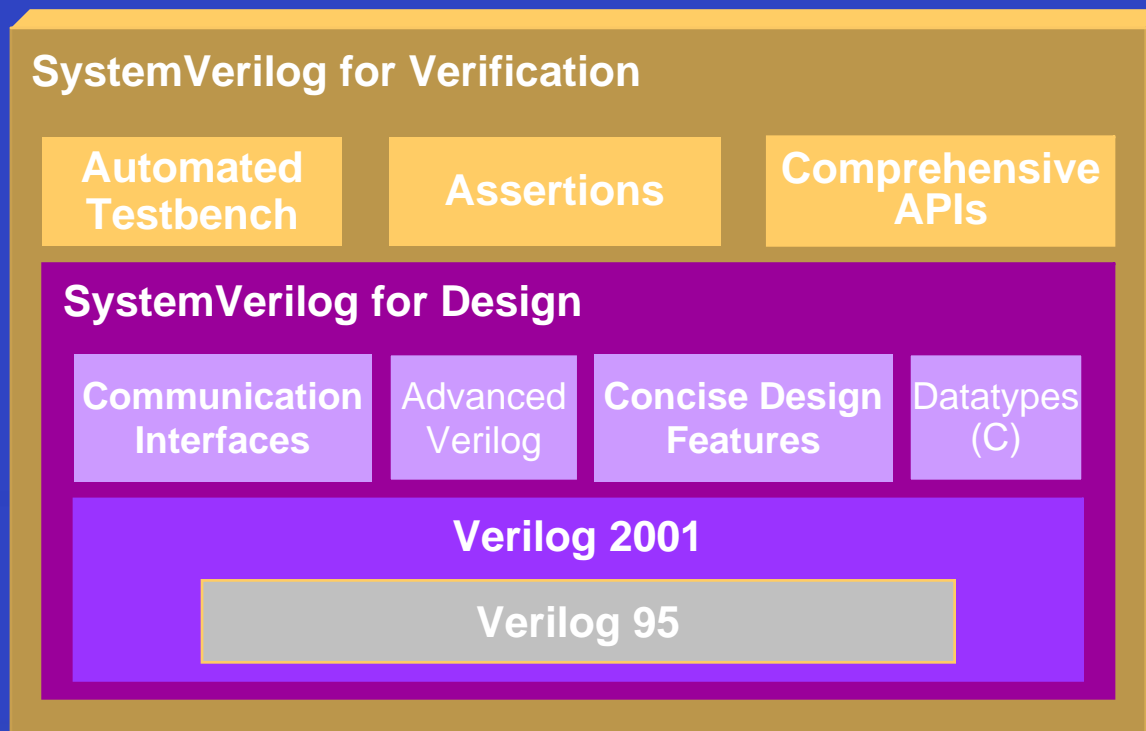
# RTL Verification Today



# Unified Design & Verification Methodology



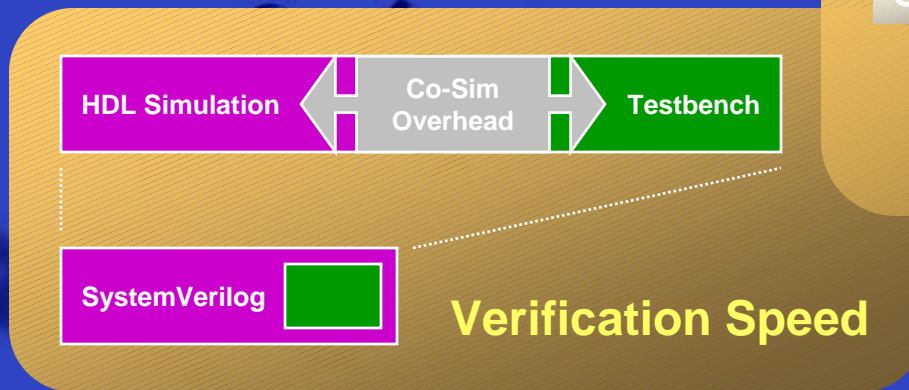
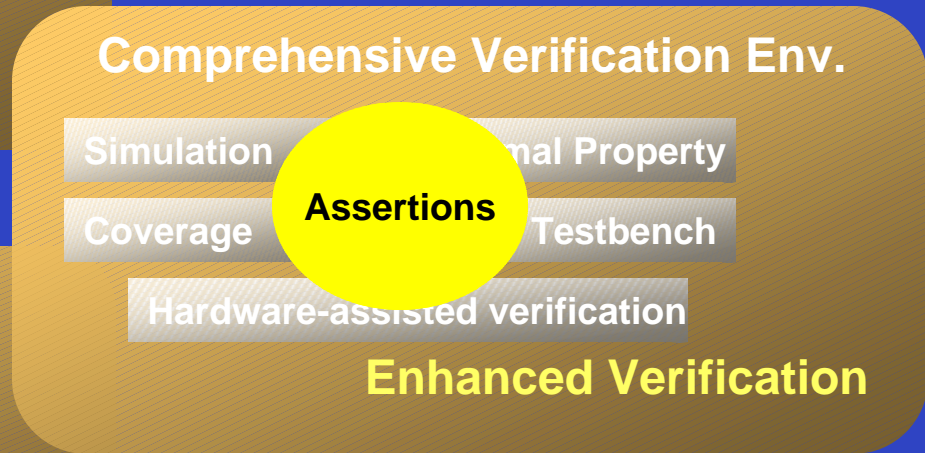
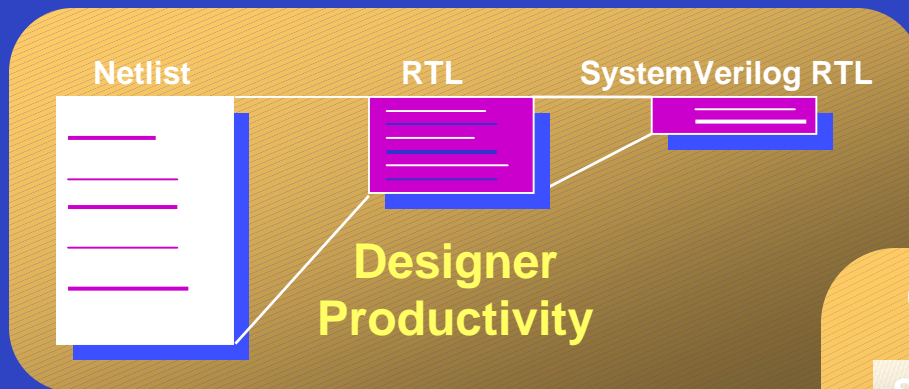
# SystemVerilog: Everything needed for Design and Verification



- Single language for design, assertions and testbench
- Supports increase code conciseness
- Evolves Verilog into a HDVL

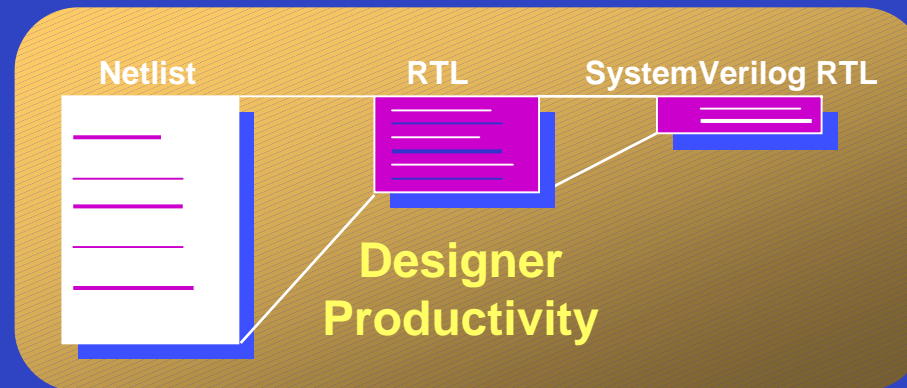
**SystemVerilog – 100% compatible with Verilog**

## SystemVerilog Means Productivity



*Your Interoperability Partner*

## Increasing Designer Productivity



- 2X – 5X less code to capture the same functionality
  - Less code → Fewer bugs
  - Easier to interpret and communicate among teams
- Evolutionary: Reduces learning curve

## Examples of Concise Coding Constructs

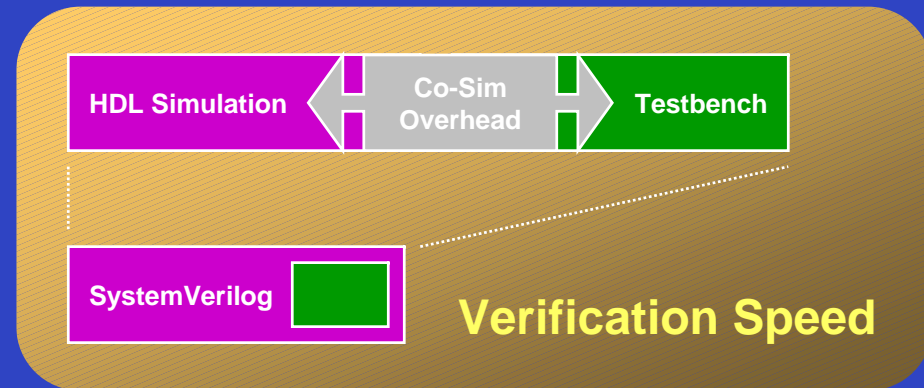
- Structures and user defined data types
  - Improves data modeling, simplifies port lists
- Interfaces – Define once, use many times
  - Encapsulates communication between modules
  - Can be designed/verified separately
  - Supports multiple levels of abstraction – signal level, transaction

**Less code – still synthesizable**

## Other Improvements To Benefit Designers

- Strong type checking
  - Global type system – earlier detection of errors
- Many language improvements
  - “Same semantics” interpretation by all tools
  - Enhanced programming statements and operators
  - Enhanced tasks and functions
- Eliminate simulation and synthesis mismatches
  - 2-state types reduces overall memory requirements
  - `always_comb`, `always_latch`, `always_ff`

# SystemVerilog Enables Greater Verification Speed

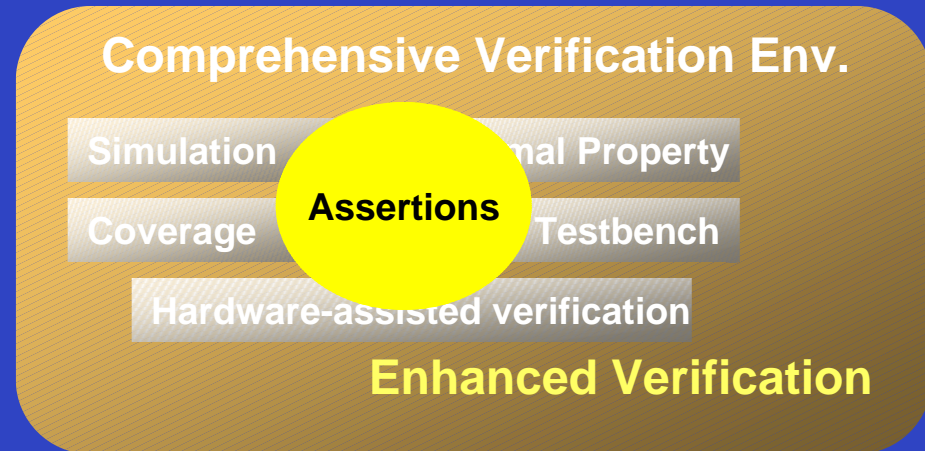


- Unified language for design and verification improves effectiveness
  - Advanced constrained-random test generation
  - Fully integrated, complete assertion technology
  - Easy integration of C Models

## Benefits of Single Language for Design AND Testbench

- Easy learning curve
  - Facilitates quicker adoption
- Improved communication between design and verification teams
- Reduces design and verification complexity with advanced constructs
- Enables faster tools and acceleration

# SystemVerilog Powers Comprehensive Verification

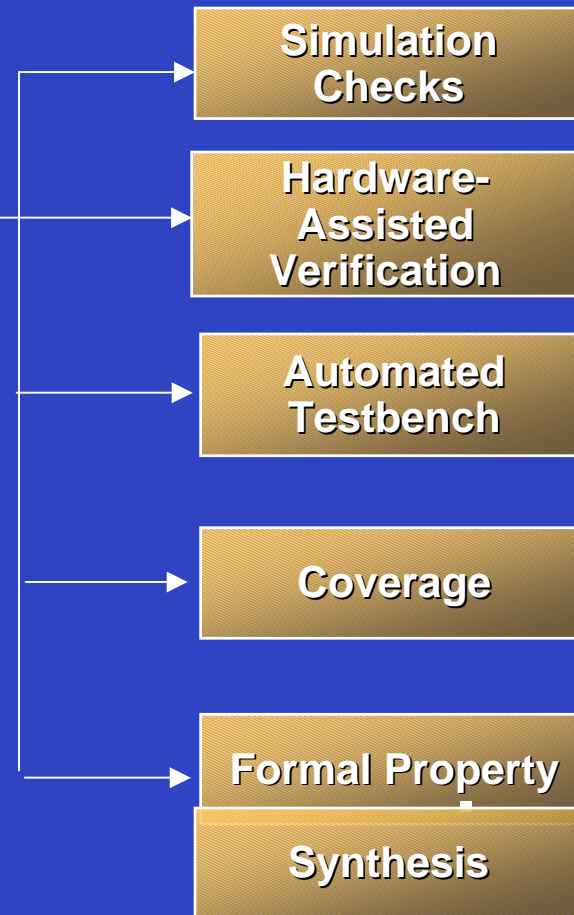


- Integrated assertions expand effectiveness of verification methodology
  - Assertion-based verification
  - Formal property verification
  - Across the board verification acceleration

## SystemVerilog Assertions: A Single Point of Specification



- Capture design intent
- Accessible to every design / verification eng.
- Same familiar language
- Fast learning curve



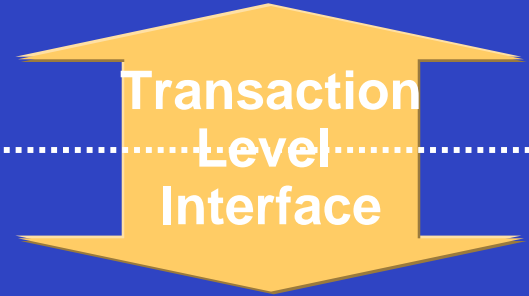
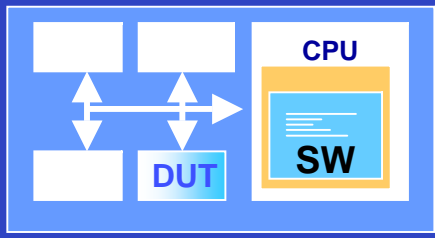
 *Your Interoperability Partner*

## We're Committed to Supporting existing Open Standards

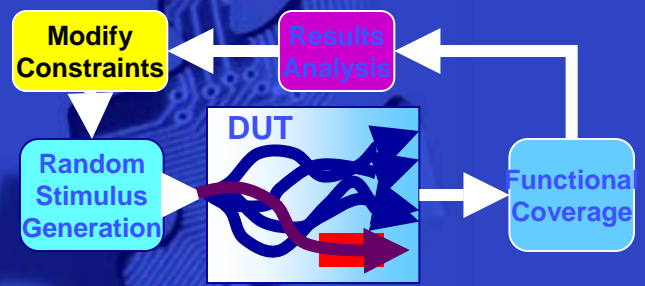
- Continued and expanding product support for
  - VHDL – VCS MX
  - OpenVera – VCS, Vera, OVASim
  - SystemC – System Studio, VCS, SystemC Compiler
- Offering SystemVerilog Catalyst Program to existing OpenVera Catalyst Program members



# System Level



# RTL



SystemVerilog  
(Verilog + OpenVera)  
VHDL

Your Interoperability Partner

# OpenVera – Today and Tomorrow

- OpenVera with Vera and VCS today
  - Vera - Simulator independent
  - VCS – High performance
- Easy migration from OpenVera to SystemVerilog
  - Performance and portability
  - SystemVerilog testbench semantics based on OpenVera
- OpenVera support will continue in Vera and VCS
  - Migration to SystemVerilog is optional



## Two things to remember:

- Full Speed ahead with SystemVerilog !
- You are safe and secure with your other previous standards investments !



➤ *Your Interoperability Partner*

## Next Sessions / Next Steps

- Leveraging OpenVera into SystemVerilog
  - Transitioning OpenVera testbenches to SystemVerilog
- SystemVerilog: A Synthesis Perspective
  - Leverage synthesizable SystemVerilog
- Learn what other EDA companies are doing with SystemVerilog
- Assertions Update for SystemVerilog
  - Transitioning from OpenVera assertions to SystemVerilog