

Design



## SystemVerilog VPI

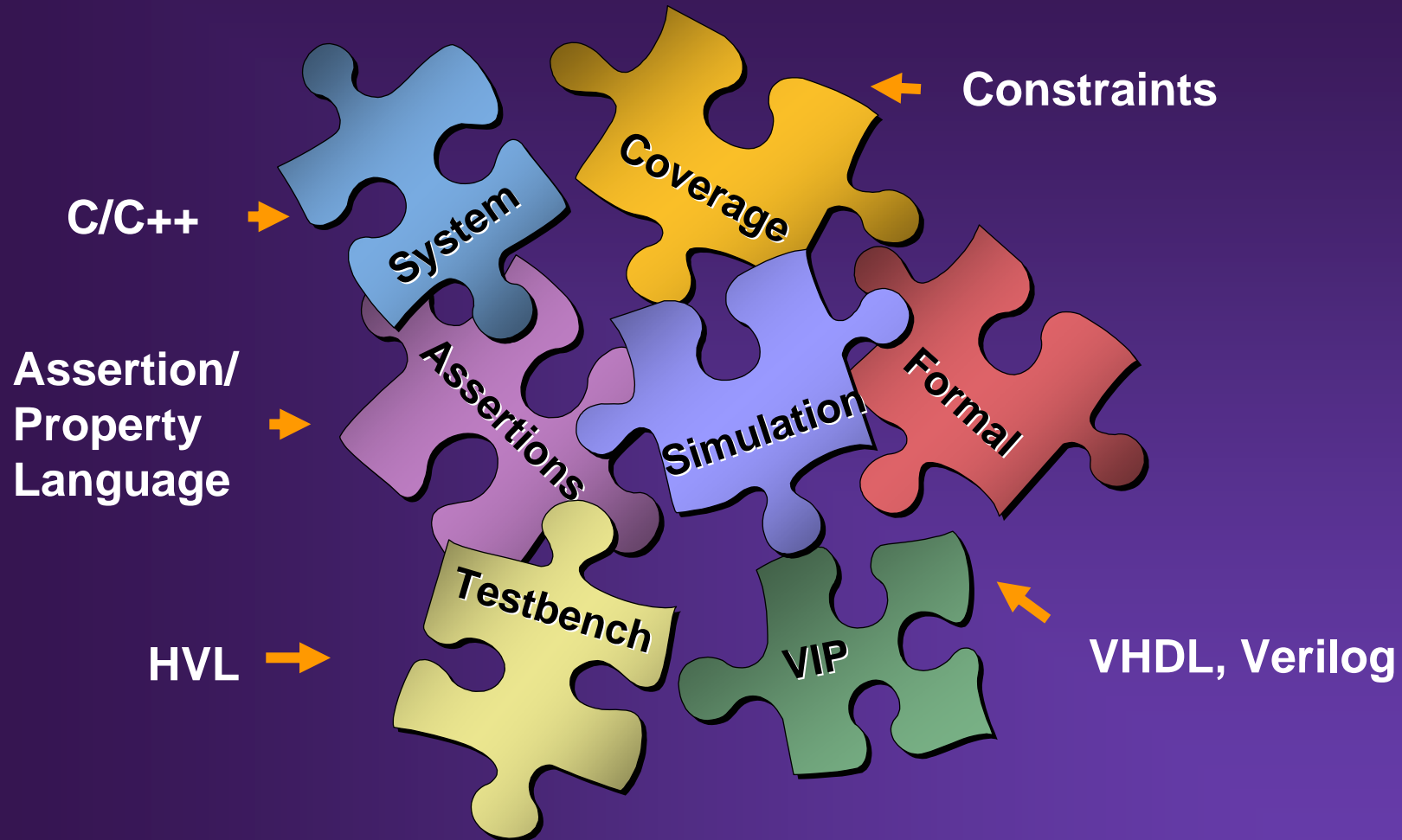
Tapati Basu  
VCS R&D Engineer, Synopsys Inc.



> *Your Design Partner*

**SYNOPSYS®**

# Language And Tools Fragmentation Have Led To Verification Inefficiencies



**Wasted Time and Productivity**

# SystemVerilog: The First HDVL

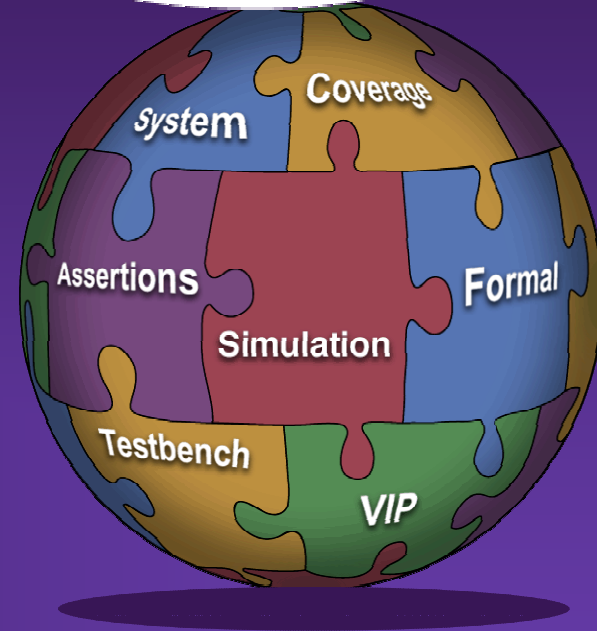
## Hardware Description & Verification Language



Verification (Simulation)

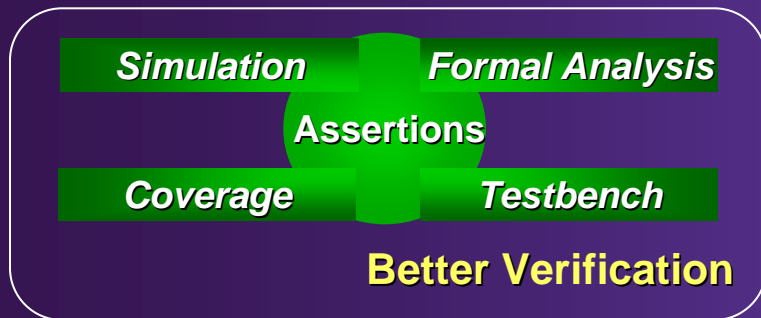
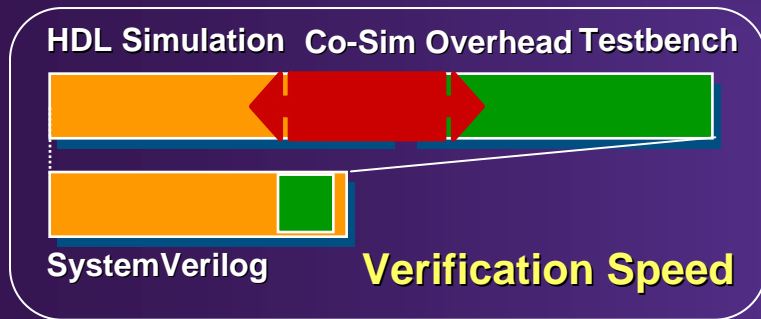
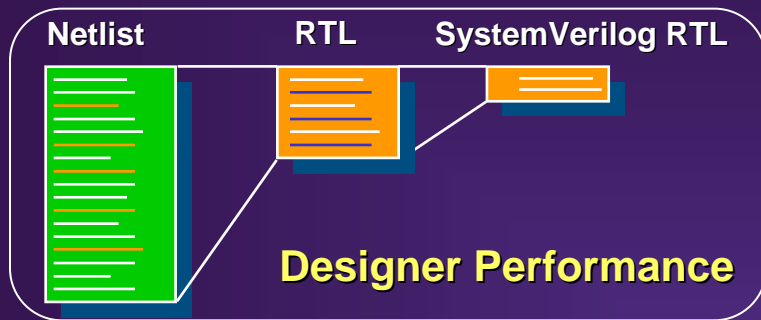


SystemVerilog



Single, Unified Language

# SystemVerilog Increases Productivity



- 2-5x less code
- Reduces synthesis/simulation mismatches
- 100% compatible with Verilog!
- No change in synthesis flow
  
- Full native testbench
- 2-5X faster verification
  
- Built-in assertions
- Capture design intent with single point of specification
- Pinpoint design errors quickly

# Verilog and Interoperability

- **The Verilog language specified the Programming Language Interface (PLI)**
  - **Provided a consistent framework for accessing design information**
  - **Critical interoperability feature**
  - **One of the main reasons for Verilog's success in the industry**
- **PLIs allowed to access design information, and to interact dynamically with simulators**
  - **Creation of waveform viewers, debuggers, and testbench automation tools**
- **Three generations of PLI**
  - **TF - operations involving user defined Task/Fns**
  - **ACC - access Verilog HDL structural objects**
  - **VPI - access Verilog HDL structural and behavioral objects**

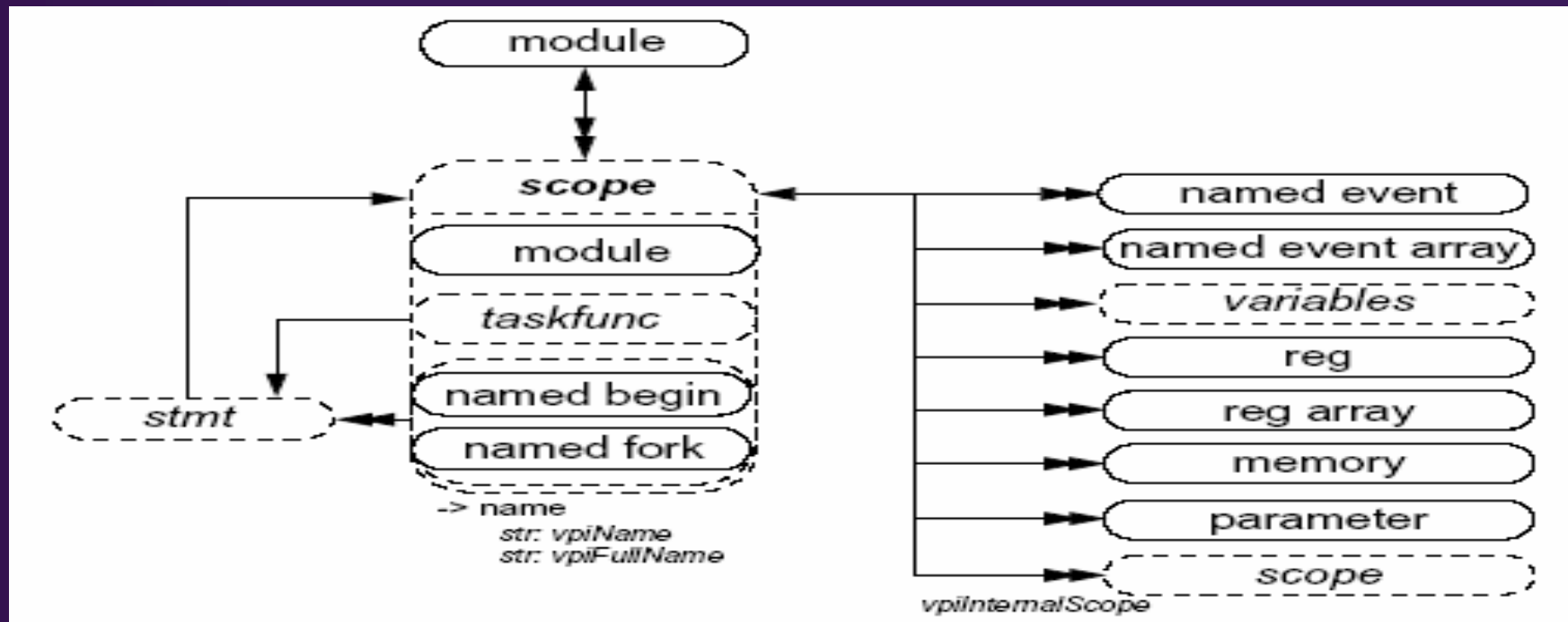
# The Acc interface (PLI1.0)

- **Second generation of PLI routines**
  - Allowed more access than just Task/Function arg
- **Retrieve structural information in the design**
  - Example: `acc_next_topmod` (retrieve all top modules in a design)
- **Modify and retrieve logic and delay values of simulation objects**
  - Application: helps debugging any possible design error

# The VPI interface (PLI2.0)

- Enhanced access to behavioral objects
  - Access to processes, statements, etc.
  - Makes it possible to regenerate the complete design
- Additional callback mechanism
  - Enables line tracing, time related callbacks, enhanced synchronization with other applications
- Extensive file I/O
- Easier to use
  - Organized error reporting
  - Fewer APIs to remember
  - Object oriented
  - Object Diagrams easily map VPI functionality

# Sample VPI Object Diagram



→ one-to-one relationship (vpi\_handle)

→→ one-to-many relationship (vpi\_iterate)

Type of target object defines the query

Properties are written in plain text (vpi\_get)

# VPI Applications

- Debuggers
- Delay calculators
- Power estimation
- Stimulus generation
- RTL Design Rule Checking
- Co-simulation (emulators, mixed mode)
- Assertion managers

# Synopsys SystemVerilog VPI Support

- **VCS support for VPI**
  - **VCS 7.2 supports VPIs for SV design and SV assertions**
- **Support for complete SystemVerilog by H1 2006**

# Conclusion

- **SystemVerilog VPI is key for tool interoperability**
- **VPI standardization is in progress (IEEE)**
- **VCS provides VPI support for 3<sup>rd</sup> party interoperability**