

1. Design Verification at a Crossroads

The Shrinking Scope of Design Verification

- As SoC functionality grows explosively, verifying all functional conditions with manually written tests has become impractical.
- As a result, DV increasingly narrows its scope from **functional verification** to **integration verification**.
- Verification turns into repetitive execution, not functional exploration.
- While shift-left verification requires coverage expansion, DV coverage is shrinking instead.**

Coverage Expansion Is the Only Way Forward

- This requires a new methodology for automatic, scalable test generation
- And, overcoming long-standing DV limitations:
 - **Temporal limitation** : inability to precisely control when effects occur
 - **Spatial limitation** : inability to reason about where and how effects propagate

2. PSS + IP-XACT for SoC Verification

Key Methodologies to Overcome DV Limitations

- PSS**

 - Portable Stimulus Standard
 - Modeling functional intent using actions, activities, and constraints
 - Supports reuse across simulation, emulation, and post-silicon environments
 - PSS provides "how the design behaves"

IP-XACT

 - Machine-readable SoC design description standard (IEEE 1685)
 - Captures hierarchy, interfaces, connectivity, and design metadata
 - Used to automate SoC RTL integration
 - IP-XACT provides "how the design is structured"

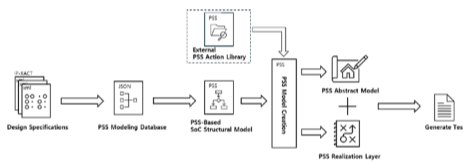
PSS + IP-XACT enable :

- Structure- and behavior-aware test generation
- Automatic alignment between RTL integration and verification
- Scalable shift-left SoC verification

3. IP-XACT Based PSS Modeling

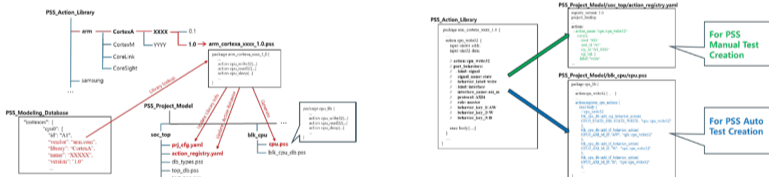
Behavior and Structure Abstraction using PSS and IP-XACT

- Design Specification**
 - Design metadata in IP-XACT xml format
- PSS modeling Database**
 - A JSON based database derived from IP-XACT
- SoC Structural Model**
 - A PSS ready database of SoC design structure
- PSS Action Library**
 - A reusable library of abstract PSS actions



PSS Project Model Generation Flow

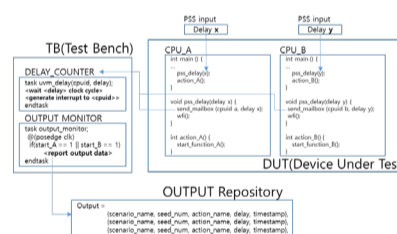
- A project-specific PSS model is created by combining the SoC structural model with the PSS action library.



4. IP-XACT Based PSS Modeling in Practice

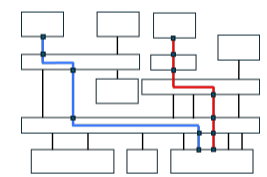
Advance Sequence modeling using IP-XACT

- Overcomes temporal controllability limitations in conventional DV
- Reproduces corner-case conditions using a delay counter and output monitor
- Generates multiple tests by sweeping cycle delays
- Monitors target condition occurrence across generated scenarios



SoC structural scenario modeling using IP-XACT

- Overcomes spatial limitations in conventional DV
- Builds a Connection Map database capturing full SoC structure and connectivity
- Enables test generation using instance hierarchy and interconnect paths
- Supports modeling and generation of tests for specific instances and end-to-end transaction paths



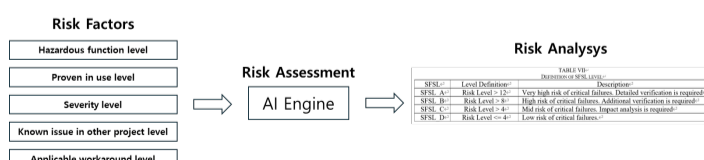
5. Enabling AI-Driven Verification with PSS

PSS: A Structural Backbone for AI-Assisted Verification

- Formalizes verification intent as an executable model
- Removes the need for AI to learn hardware semantics from scratch
- Integrates IP-XACT design metadata into a unified reasoning framework
- Enables structure-aware, constraint-driven AI test generation

Case Study: AI Assisted RISK-Aware Verification

- Risk scores for SoC instances and end-to-end transaction paths
- AI-guide test prioritization and scenario weighting through PSS
- High-risk instances and connectivity paths exercised first
- Enables scalable AI-assisted verification with reduced modeling overhead



6. Conclusion

Enabling Systematic Structural Exploration for True Shift-Left Verification

- Our goal is to verify **all functional conditions** that can arise in an SoC during pre-silicon verification
- Using IP-XACT design metadata, we systematically applied **Port Scan** and **Behavior Search** to enumerate SoC level functional conditions
- This IP-XACT driven functional enumeration enabled substantial coverage expansion and practical pre-silicon corner-case detection, as shown in the table below

Functional Cover group	Previous #of function bins	Latest #of function bins	Difference (%)
Accessibility	224	249	+11.1%
Block functions	126	131	+3.9%
Bus interface	80	107	+33.7%
Clock	147	183	+24.4%
Power	104	143	+37.5%
Reset	19	22	+15.7%
Interrupt	295	345	+16.9%
Security	160	178	+11.2%
Cross Functions	38	203	+434.2%
Total:	1193	1558	+30.5%

"Verification must evolve from random exploration to structural determinism." That is how we unlock its true potential and redefine the boundaries of design verification.