2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 14-15, 2025

Fully Automated Verification Framework for Configurable IPs: From Requirements to Results

Shuhang Zhang, Jelena Radulovic, Thorsten Dworzak

Infineon Technologies AG

infineon

accellera SYSTEMS INITIATIVE

Outline

- Development Challenges of Configurable IPs
- Automation Challenges
- Automated Development Framework
  - Automated Verification Framework
- Results
- Conclusion

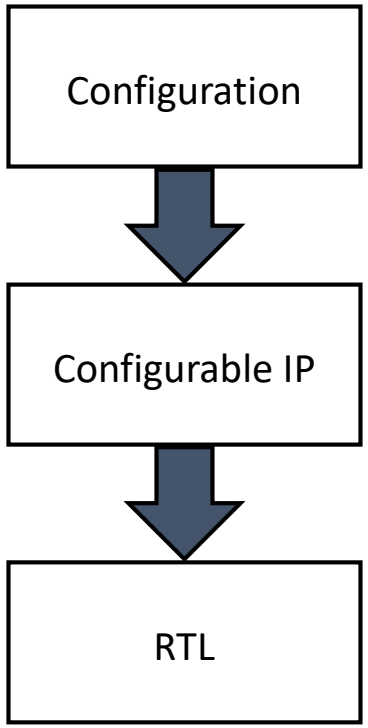
Development Challenges of Configurable IPs

Configurable IPs widely utilized in today's SoCs

- Design flexibility and scalability
- Reduced development time
- Optimized resource utilization

However,

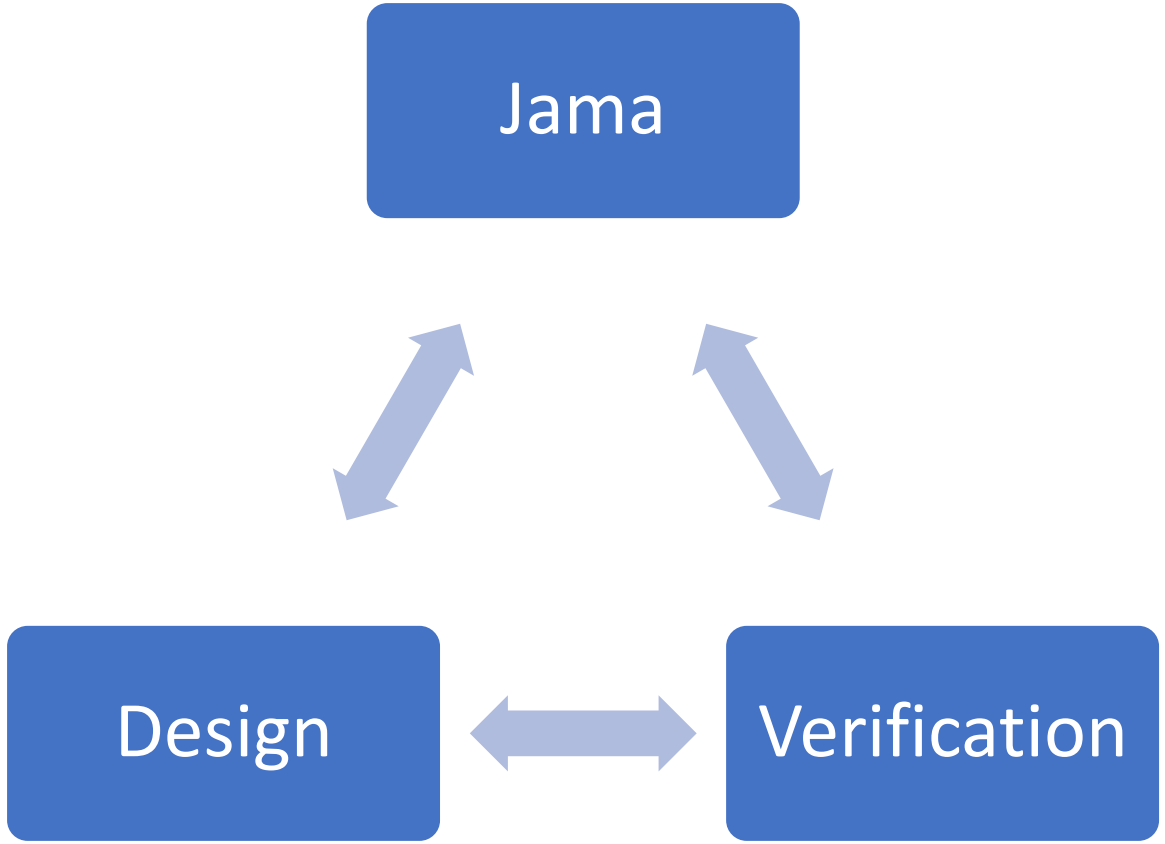
- Increased complexity in design and verification
- Time-consuming to ensure the correctness
- Hard to handle a large volume of configurations



Automation is the solution

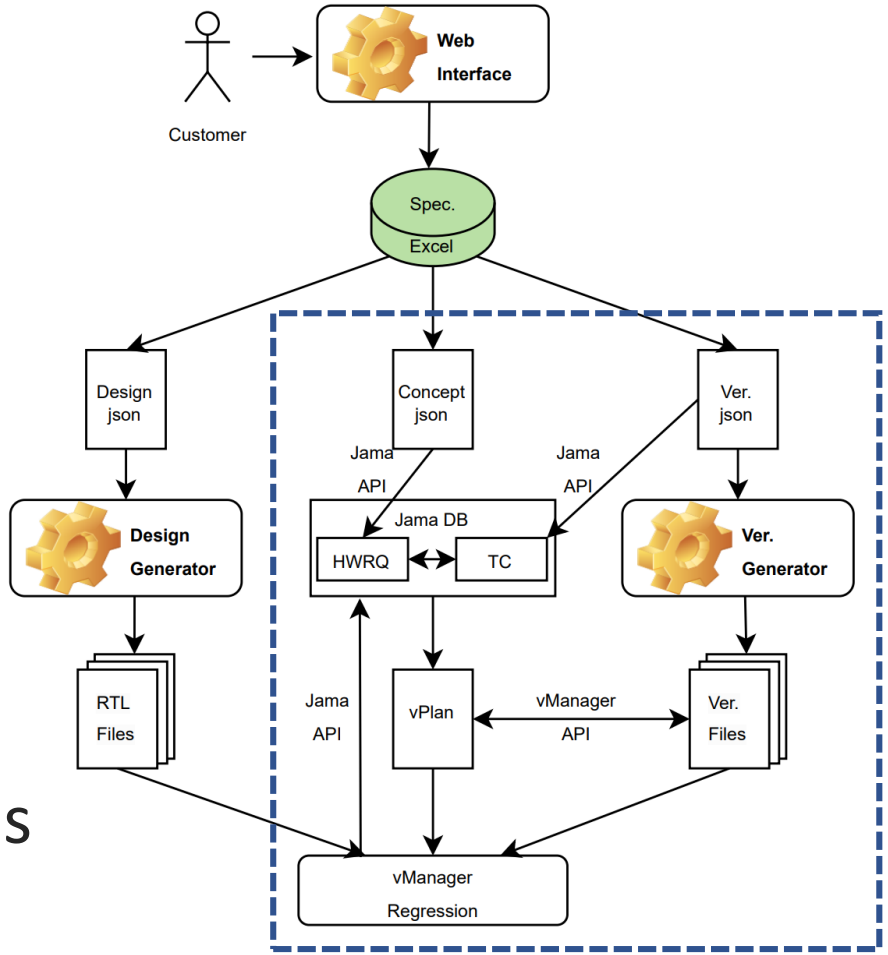
Automation Challenges

- Concept
  - Construction of the Jama database
- Design
  - Jama to Design
- Verification
  - Jama to Verification
  - Regression
  - Verification to Jama

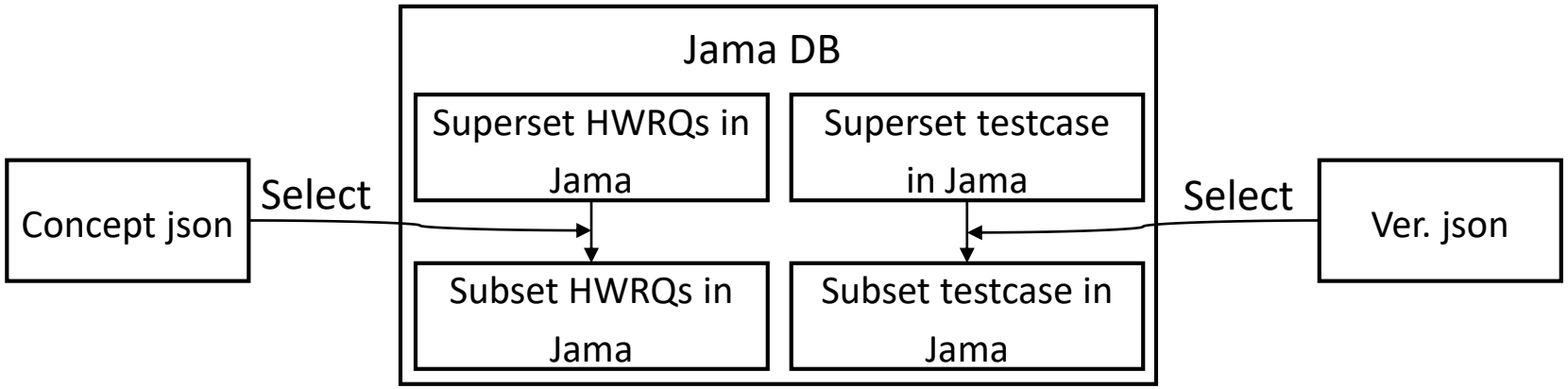


Automated Development Framework

- Concept
  - Jama database is constructed using configurations through the web interface
- Design
  - Design is generated automatically based on the configurations
- Verification
  - Automated Jama-to-Jama verification flow is implemented, using the same configuration

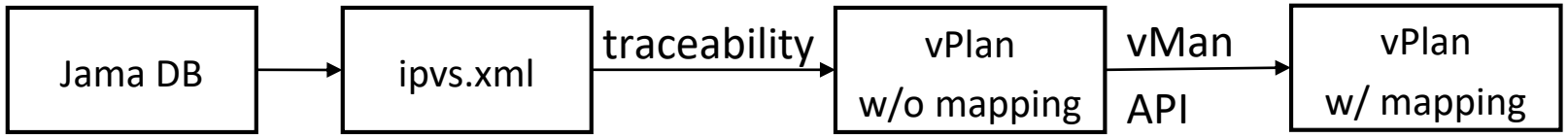


Jama Database Generation



- A predefined superset of Hardware Requirements (HWRQs) and test cases is designed to cover all possible configurable options
- Based on the configurations provided by the customer, a tailored subset of HWRQs and corresponding test cases is dynamically selected

vPlan Generation

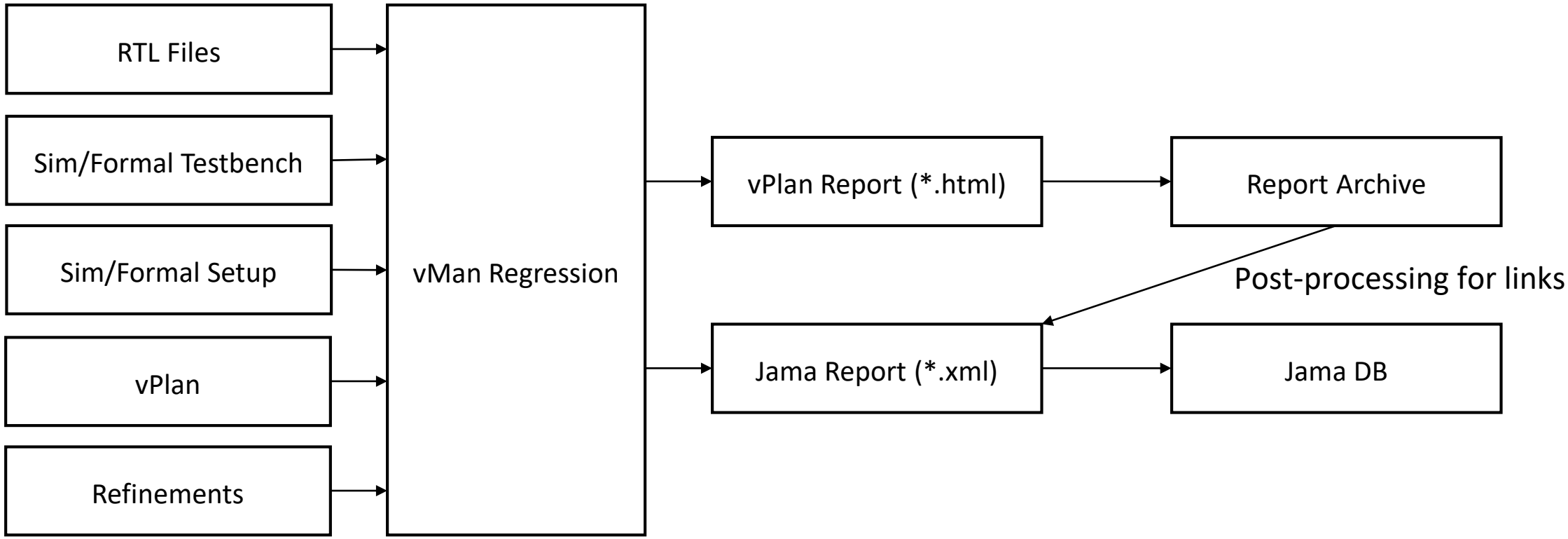


- The Jama API is utilized to automatically generate the required XML file
- An in-house tool is employed to generate the vPlan without mapping patterns
- The vManager API is then used to programmatically add and finalize the mapping patterns to the vPlan

Verification Environment Generation

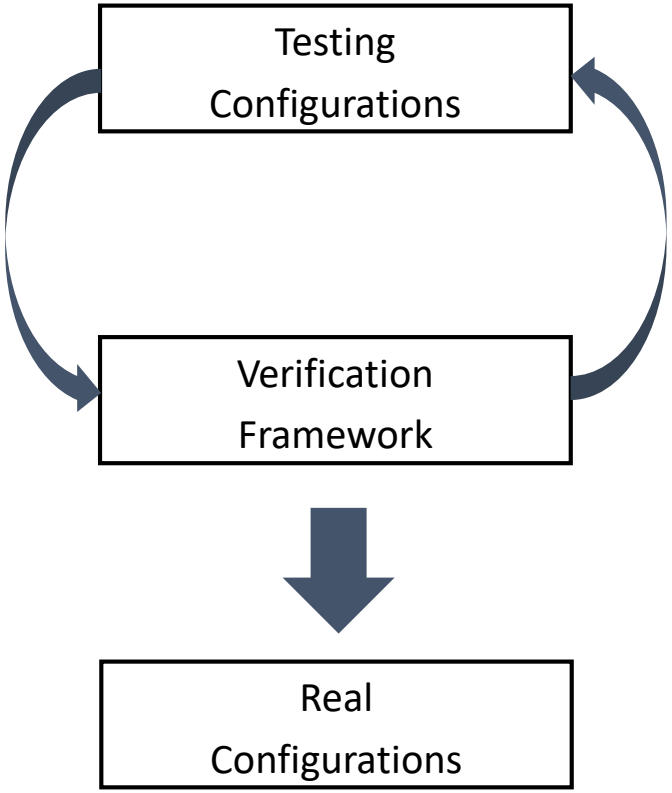
- Simulation-Based Verification
  - Utilizes an in-house testbench generator to create the skeleton of the testbench
  - The framework dynamically generates its own files to replace pre-defined placeholder files, such as: Test sequence and Scoreboard files
- Formal Verification
  - Primarily focuses on the automated generation of SVA files
  - Additionally, generates the necessary TCL scripts to ensure seamless execution of the formal verification process

Regression and Reporting



Framework Robustness

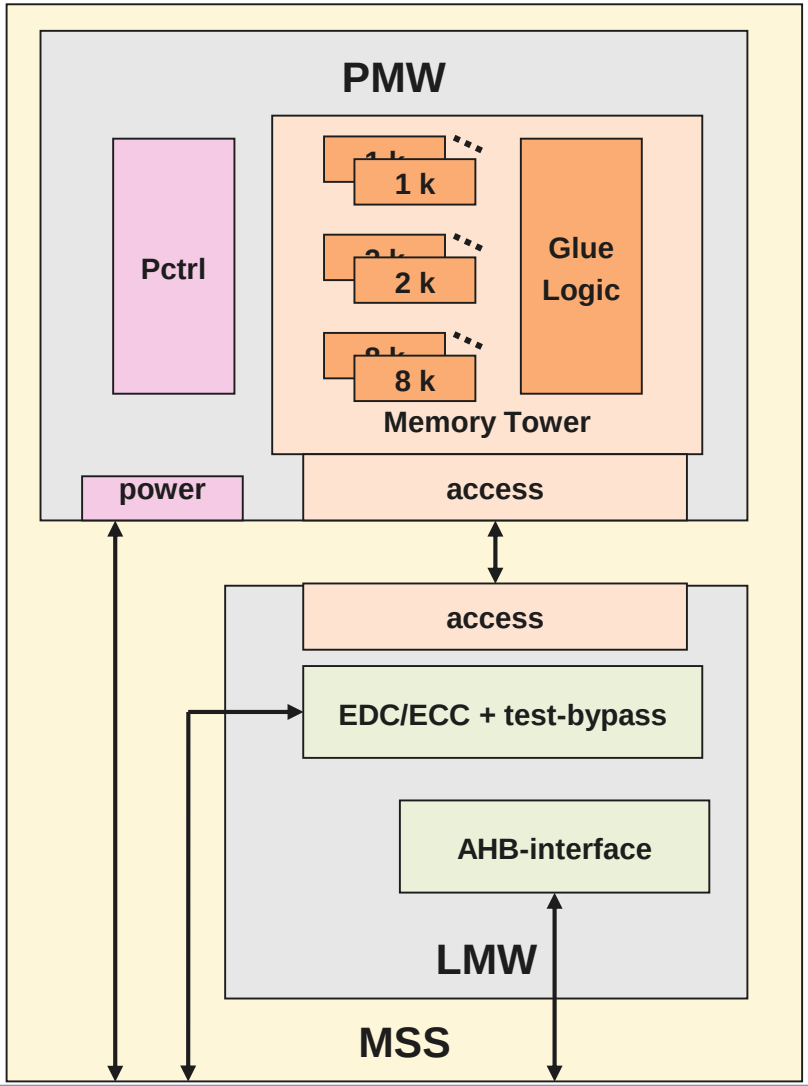
- Robustness is crucial for fully automated flows
- Extensive testing configurations enhance robustness
- Thorough review of superset HWRQs and TCs
- Detailed waiver reviews



Results

Configurable Memory Subsystem IP

- Initial Setup: Requires a large one-time effort to establish the automation framework
- Effort Reduction: Achieves a significant reduction in effort for subsequent configurations
- Impact: The effort per configuration is reduced dramatically, from 40 person-days (PD) to just 2 person-days (PD)



Conclusion

- A push-button solution can be achieved for configurable IPs, streamlining the process from requirements to results.
- The framework serves as a wrapper, integrating multiple tools such as Jama/vMan-API and in-house tools.
- Although the initial effort for creating generators and supersets is substantial, it provides long-term efficiency and benefits for future configurations.

Questions