

JUNE 23-27, 2024

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

Leveraging Formal Verification to Create, Reproduce, Verify Design Scenarios from Simulation Wave-dump

Authors:

Ujjwal Talati	(ujjwal.talati@synopsys.com ,	Synopsys India Pvt. Ltd, Pune, Maharashtra, India)
Saikiran Ashwini	(ashwini.saikiran@synopsys.com ,	Synopsys India Pvt. Ltd, Pune, Maharashtra, India)
Aman Chauhan	(aman.chauhan@synopsys.com ,	Synopsys India Pvt. Ltd, Pune, Maharashtra, India)
Viraj Rawal	(viraj.rawal@synopsys.com ,	Synopsys India Pvt. Ltd, Bengaluru, Karnataka, India)
Nivin George	(nivinninan.george@synopsys.com ,	Synopsys India Pvt. Ltd, Bengaluru, Karnataka, India)

SYNOPSYS



Motivation

- **Reproducing Corner Case Bugs or a Scenario:**

- Design Engineers often face missing corner case bugs in today's complex Digital Designs.
- These corner case bugs will be found by the Verification team and the Simulation waveform that hits these scenarios is provided in terms of FSDB or VPD to the Design Engineers.
- Design Engineers analyze the scenario, add a design fix, and re-run the Simulation to verify it. This is a time-consuming as well as multi-iterative process.

- **Challenges:**

- It is very challenging and even time-consuming to reproduce the same scenario on the Designer side found in the verification environment independently.
- Typically, the Simulation Verification Environment is used to verify the design behavior, but it is not exhaustive for verifying corner cases.
- Sometimes it takes days to reproduce a corner case scenario for Verification Engineers normally done by fine-tuning the testbench.
- Formal Verification provides an exhaustive Verification but requires knowledge of writing System Verilog Property.

- **Implemented Solution:**

- With this Python-based utility, Design Engineers can reproduce the intended scenario directly from the Simulation dump, use automatically generated cover properties, and verify the fix in a Formal Verification Environment (Assurance of fix in Formal)
- The turnaround time would be reduced for reproducing the scenario since only a timing sample of interest is generated (Time/Productivity)
- Design Engineers can do this as a spot verification step (Agnostic of Verification/Formal Knowledge)
- Generated Properties can be directly added to the Simulation Verification Environment which can help pinpoint the issue or trace the source if it re-occurs in the future (Observability)

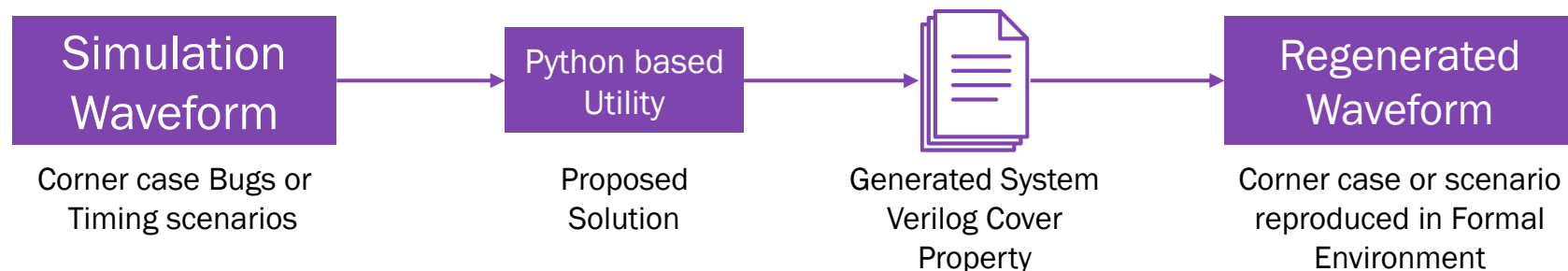


Figure 1: Flow Diagram

Main Idea

...(1)

- Utility can be used to:
 - Regenerate failing scenarios – Input Simulation Wave-dump (F1)
 - By analyzing the waveform dump in a failing scenario, the Design Engineers identify the precise failing timing window that they want to reproduce.
 - Using the FSDB report, the timing information of various signals for that window is generated in a CSV format.
 - The CSV file is fed to the Sim2SVA Utility along with the clock information.
 - Hit a user scenario – Input existing waveform as JSON (F2) [In absence of F1]
 - User can create their timing diagram in JSON format and provide that to the utility.
- Once input is provided, the utility generates the System Verilog Cover Property Sequence.
- RTL Design (without the fix) along with generated Cover Property is fed to Formal Property Verification Tool (Synopsys VC Formal Tool)
- The Formal tool will regenerate the scenario using the given properties and provide the status as Coverable.
- Design Engineers can analyze the reproduced scenario and can re-run the Formal with fixed RTL Design to verify the fix. The System Verilog Cover Property will be un-coverable after the fix which proves the failure scenario is not seen anymore.

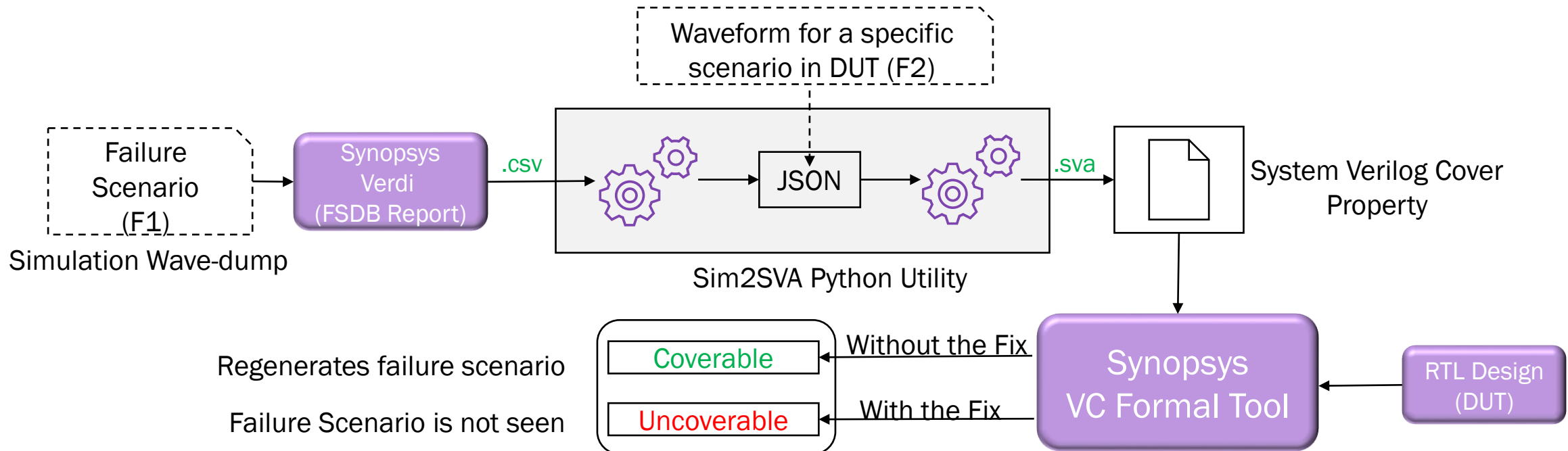


Figure 2: Sim2SVA Block Diagram

...(2)

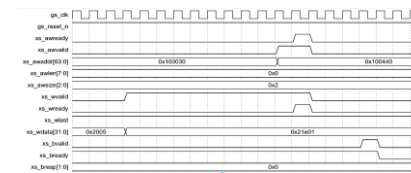


Figure 3: User Flow

Evidence

Table1: Results

Exhibit	Time Window (cycle)	Simulation Test RUNTIME	Proposed Utility RUNTIME	Time Reduction
Protocol 1	19	~180 min	~1 min	~180x
Protocol 2	75	~150 min	~3 min	~50x
Protocol 3	151	~720 min	~14 min	~51x

- It is evident from Table 1 that to reproduce the scenario by simply re-running the test in Simulation, it takes 180 mins, 150 mins, and 720 mins respectively for the three protocol scenarios.
- After identifying the precise failing timing window (19, 75, and 159 cycles), reproducing the same reference scenarios using the proposed utility and running it with Formal Verification takes ~1 min, ~3 min, and ~14 min, respectively.
- Thus, by using the proposed utility, 180x, 50x, and 51x time reduction in reproducing the scenario and verifying it using Formal Verification can be achieved in a completely autonomous way.

Protocol 1 : Covered in ~1min

Targets: ALL Filter by name					
	status	depth	safe_depth	name (C)	elapsed_time
1	✓	20	19protocol_1_wavefrom2sva__cover_property	00:00:01
Run Time: 0:00:36					

Protocol 2 : Covered in ~3min

Targets: ALL Filter by name					
	status	depth	safe_depth	name (C)	elapsed_time
1	✓	79	78protocol_2_wave2sva__cover_property	00:02:51
Run Time: 0:03:09					

Protocol 2 : With fix is Uncoverable

Targets: ALL Filter by name					
	status	depth	safe_depth	name (C)	elapsed_time
1	✗		200protocol_2_wave2sva__cover_property	00:01:32
Run Time: 0:01:55					

Protocol 3 : Covered in ~14min

Targets: ALL Filter by name							
	status	depth	name (C)	vacuity	witness	engine	elapsed_time
1	✓	159wavefrom2sva__cover_property			b1	00:13:53
Run Time: 0:14:12							

Figure 4: Results

Summary

- The utility helps **Design Engineers** to independently recreate scenarios and verify the fix for the following applications:
 - Reproduce corner cases directly from the generated Simulation waveform.
 - Define a scenario at a high level and generate them at an early development cycle.
- This approach provides **~50x to 180x reduction** in time to regenerate scenarios.
- Generated timing diagram can also be used for Architecture or Change Order Specification documents to clearly define and expand upon the design intent.
- **Future Scope:**
 - The generated **System Verilog Cover Property** can later be modified to enhance the coverage scope.
 - Functional coverage can be integrated with the **System Verilog Cover Property** to reach uncovered design logic.
- **Limitations:**
 - The time taken by the Formal tool to reproduce the **System Verilog Cover Property** Sequence is limited by the depth of the time sample.



SHAPING THE NEXT GENERATION OF ELECTRONICS

JUNE 23-27, 2024

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA



Thank you

Email: ujjwal.talati@synopsys.com

SYNOPSYS