2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

Automation for Early Detection of X-propagation in Power-Aware Simulation Verification using UPF IEEE 1801

Tony Gladvin George, Ramesh Kumar, Kyuho Shim, Karan K,

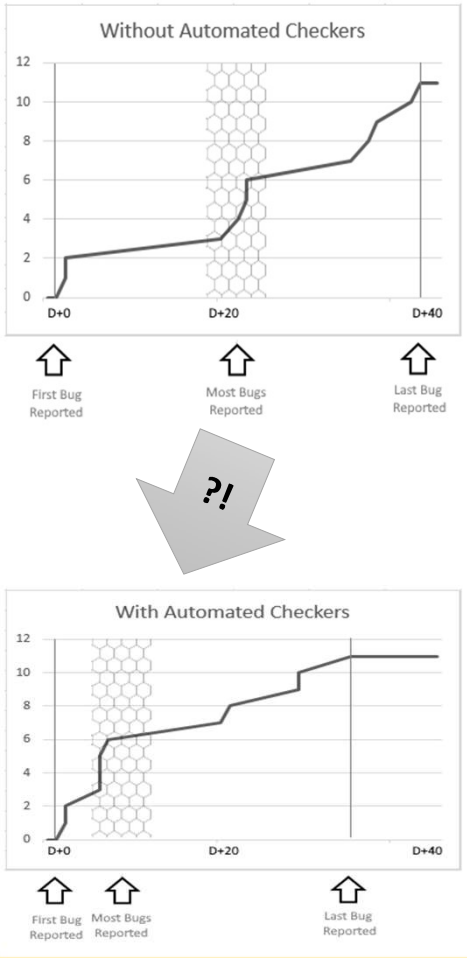
Wooseong Cheong, ByungChul Yoo

Memory Division, Samsung Electronics

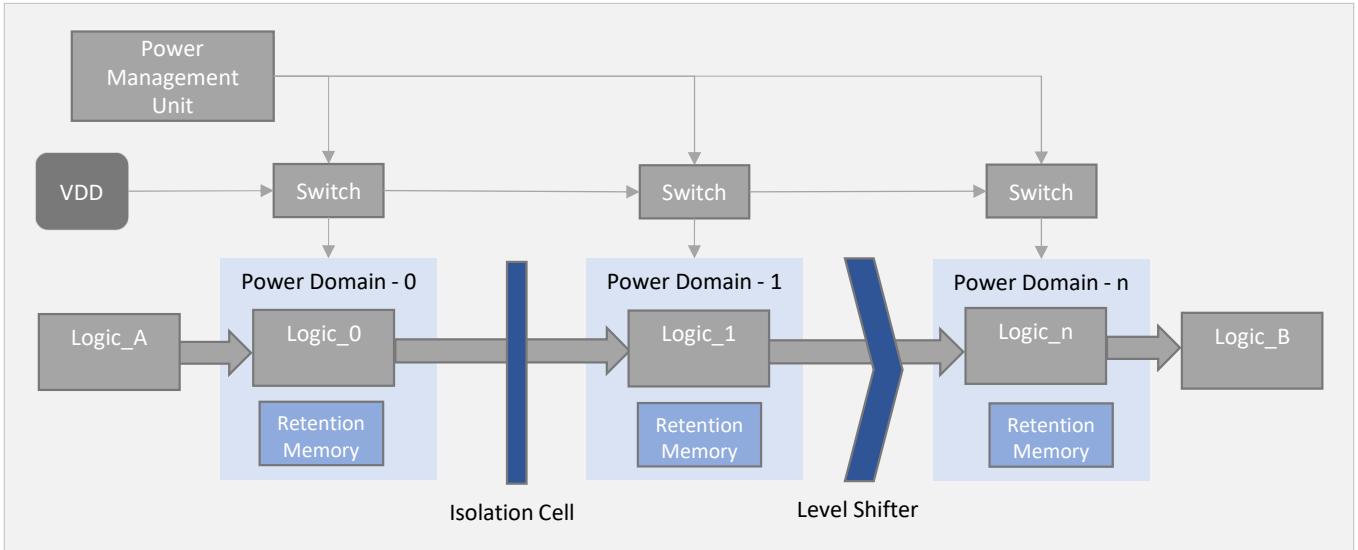
SAMSUNG

accellera SYSTEMS INITIATIVE

Table of Contents



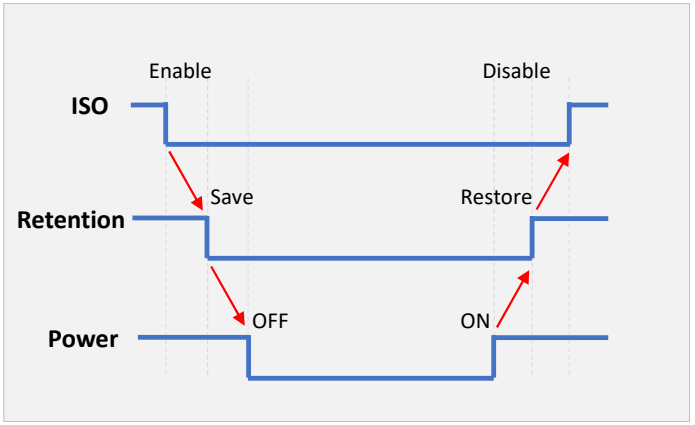
Verification of SoC with Power Design



- Power States
  - Power Off
  - Sleep
  - Power On
- During On/Off
  - Retention cell – save
  - Retention cell – restore
  - Manage Isolation cells

Power Sequence for each power domain

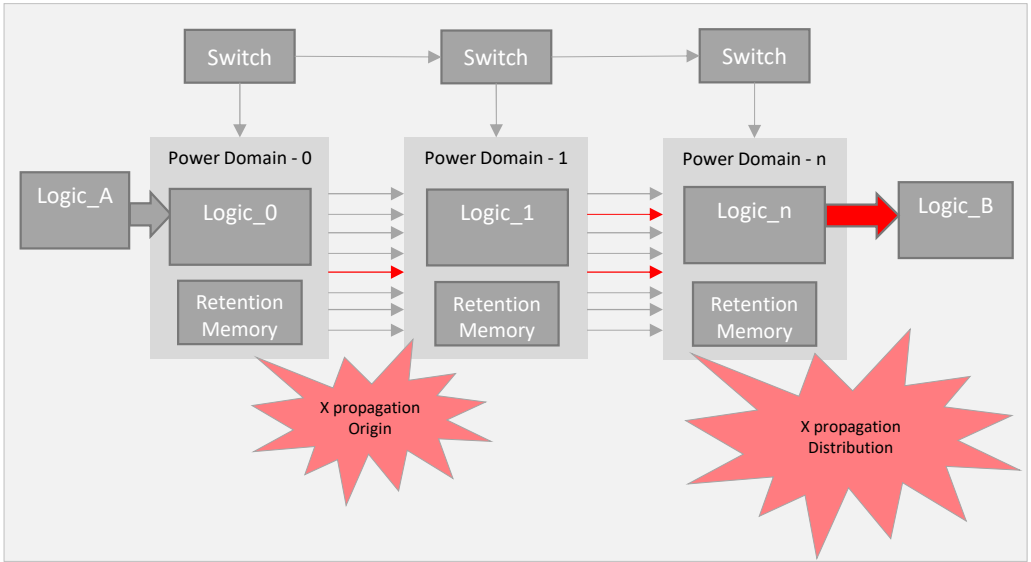
- Power Down
  - ISO enable → Retention Enable → Power off
- Power On
  - Power On → Retention Disable→ ISO disable



Abbreviations:

ISO –> Isolation

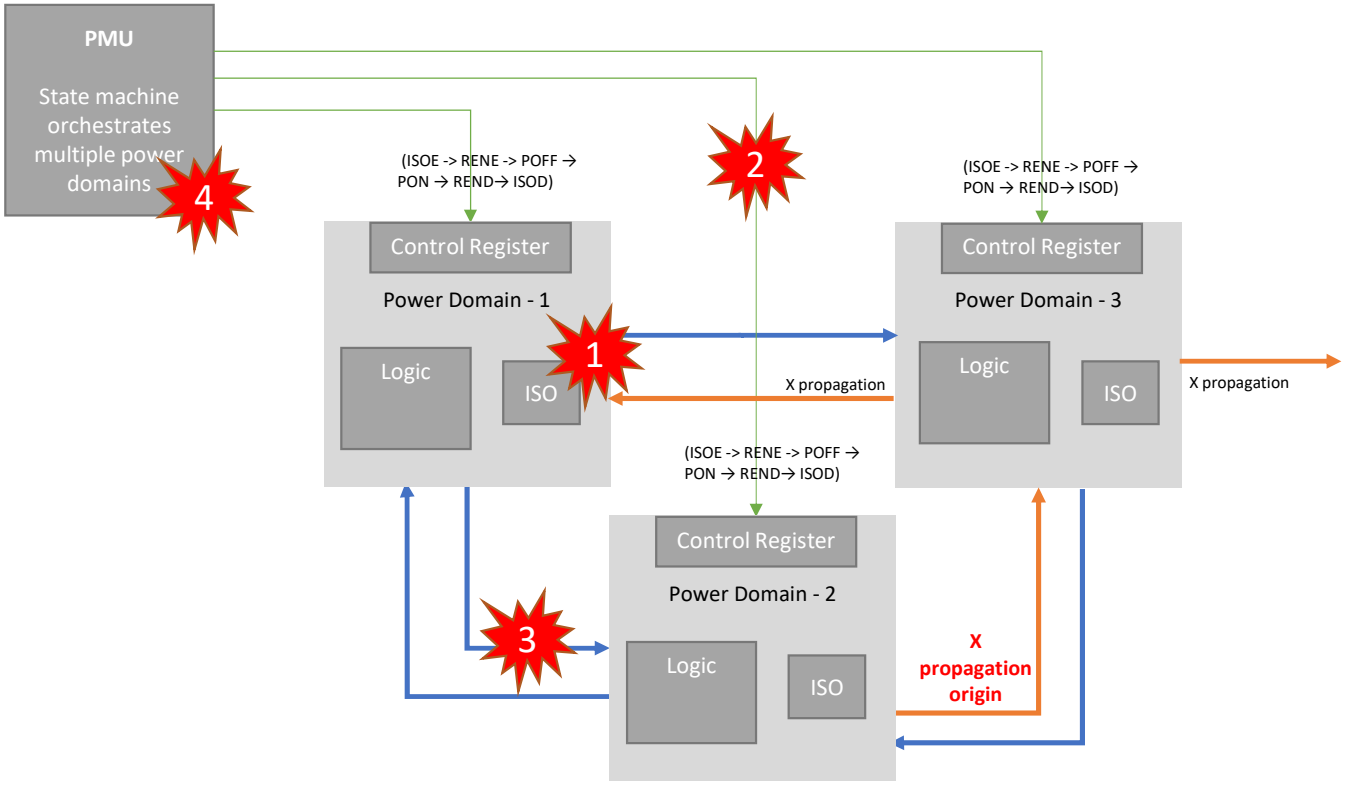
The Challenge of X-Propagation



1. X Propagation and Amplification
   - Manually tracing back the X-value from hundreds of signal is slow process.

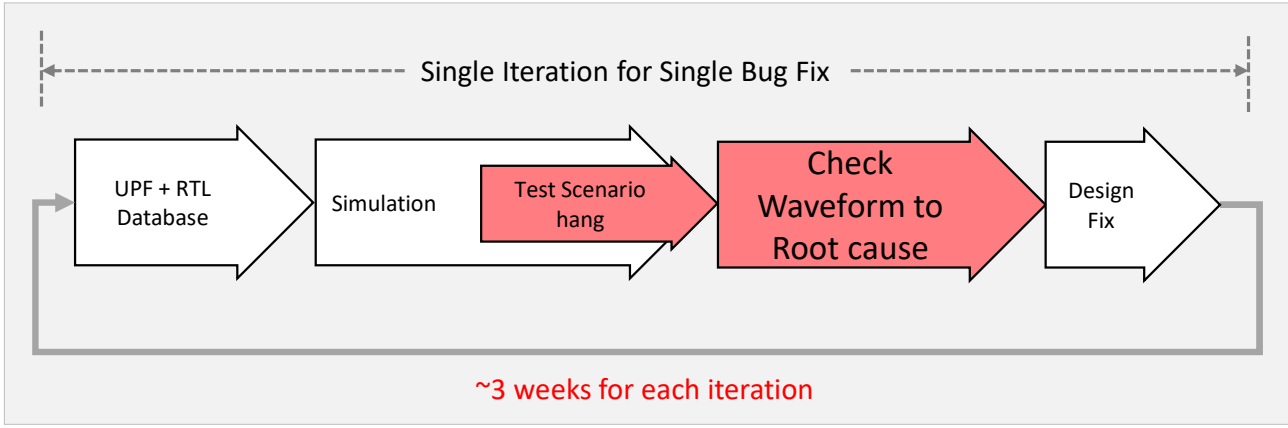
2. Execution of correct power sequence in independently operated power domains.
   - Validating the power sequence is executed correctly.
   - The wrong power sequence can cause X-propagation.
   - Number of Power domains in our project was 38, which increases the scale of problem
   - Non-Scalable verification effort when the power domains scales up.

How does X-propagation happen?



1) Missing isolation cell
   - Design Issue
2) Missing control signal
   - Connection Issue
3) Missing connection in the data signal
   - Connection Issue
4) Wrong power sequence control
   - Design Issue

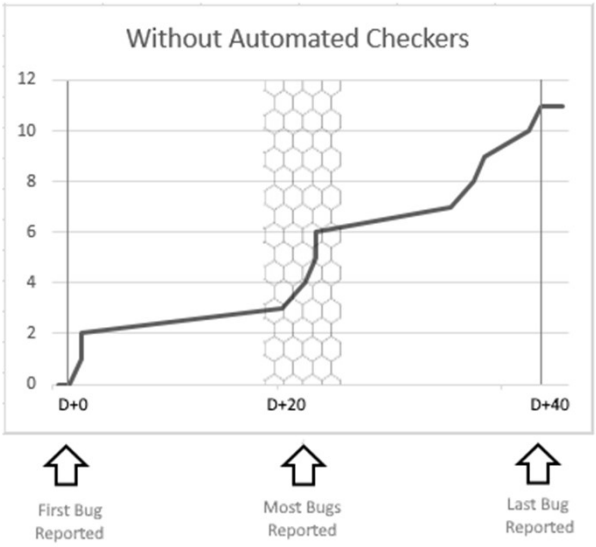
Limitations of Traditional Approach of UPF simulation verification



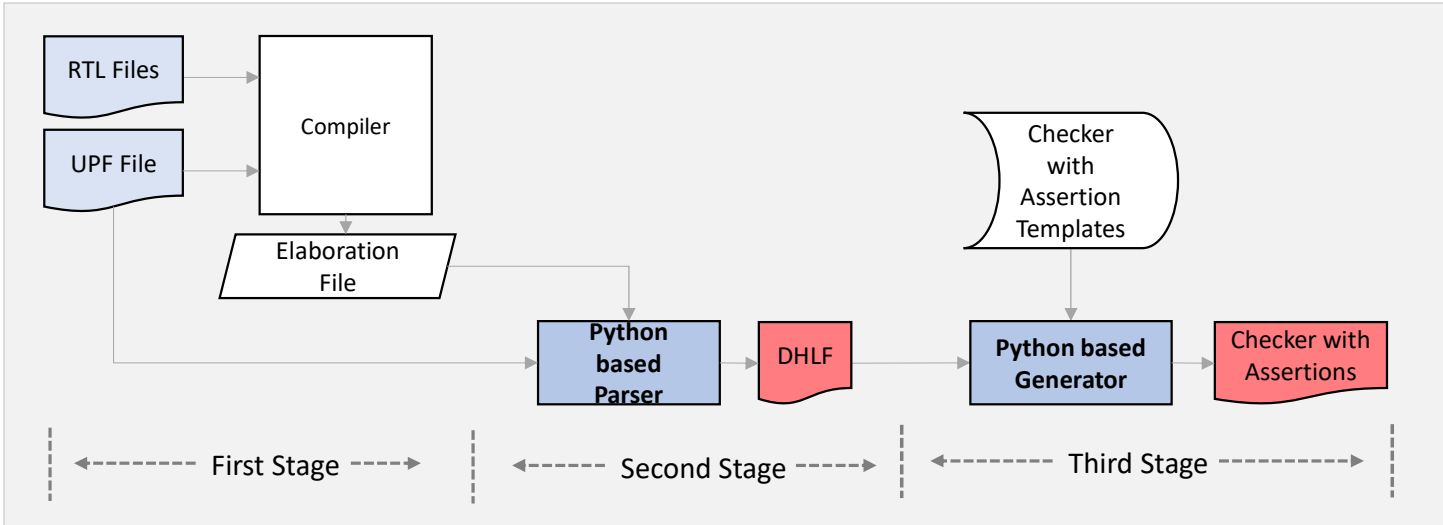
Case Study on a finished project

Issues faced during UPF verification

1. Bugs found were late in the verification cycle
2. Root causing of the X propagation is time consuming
3. Waveform based analysis of power sequence is time consuming

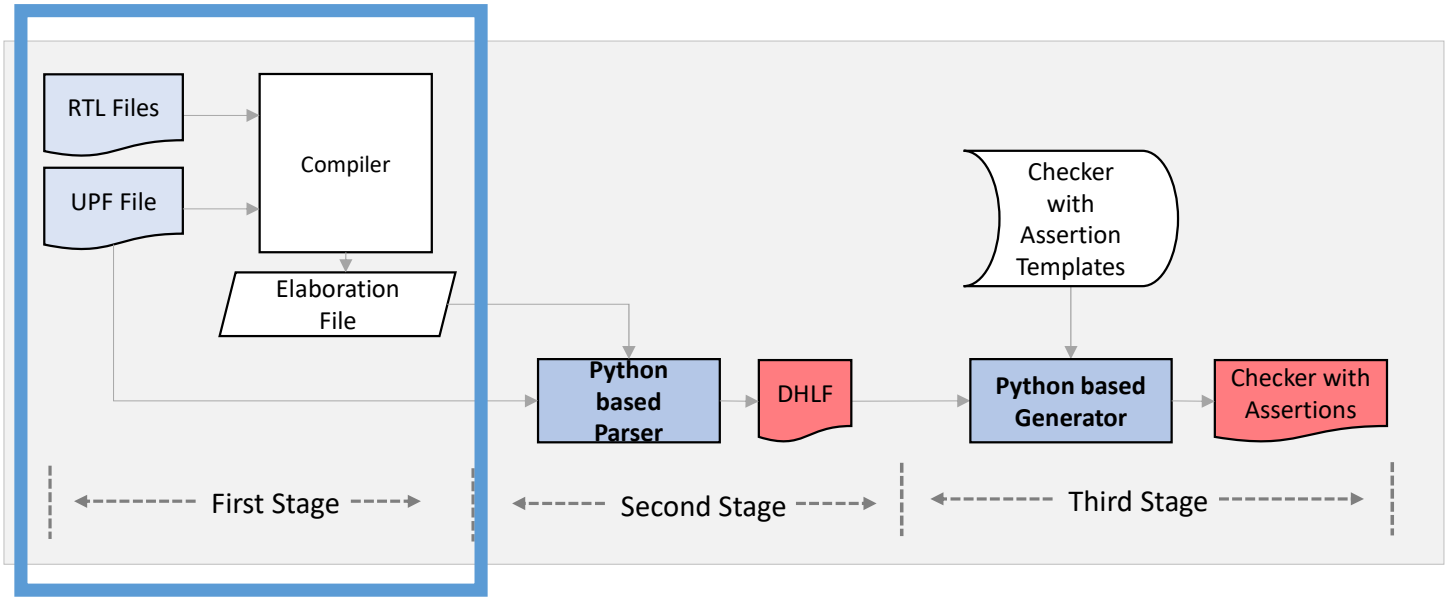


Generation of checkers with assertions in Python



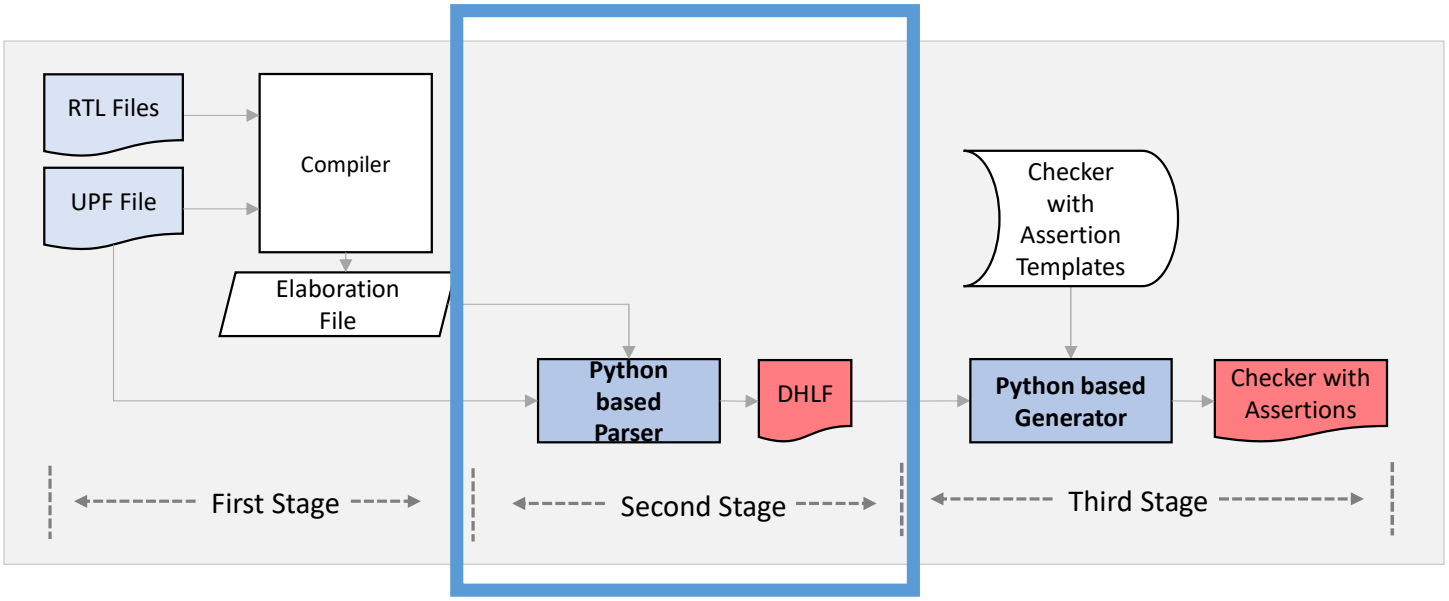
- Three stages

Generation of checkers with assertions in Python



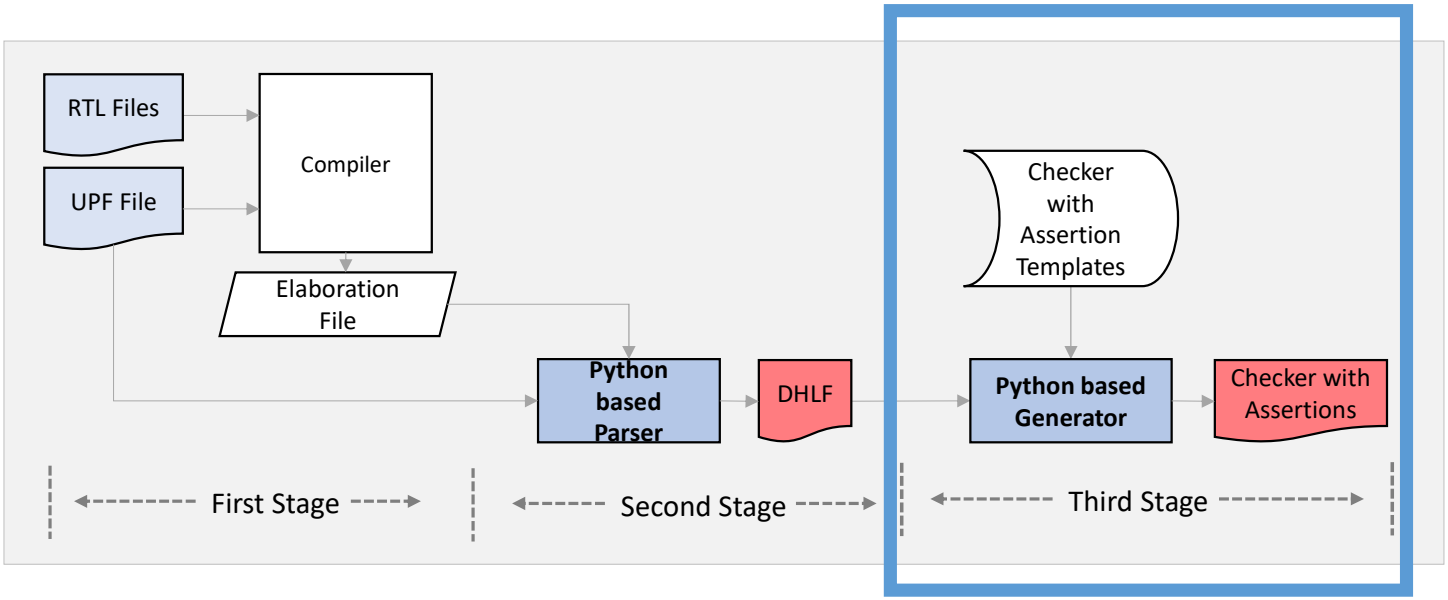
- First Stage
  - Challenge – Need for RTL hierarchy
  - Use Elaboration file to gather RTL hierarchy information.

Generation of checkers with assertions in Python



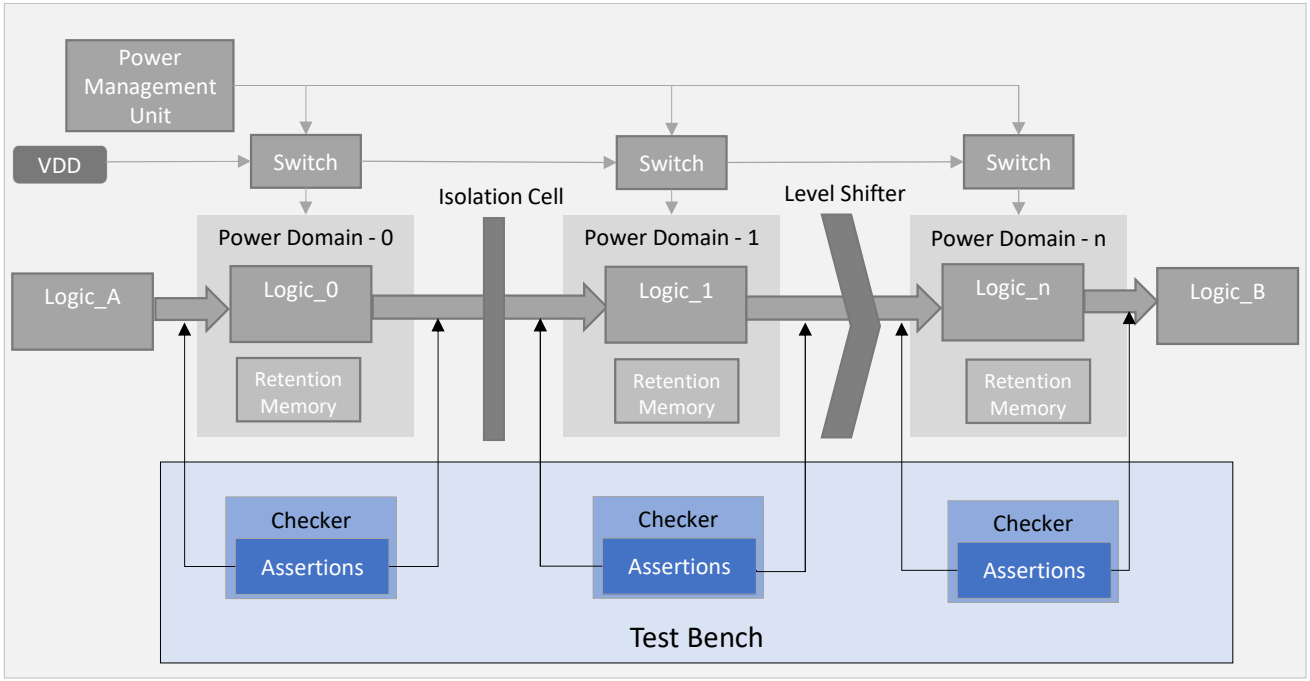
- Second Stage
  - Parse elaboration file.
  - Get the RTL hierarchy information.
  - Parse the UPF syntax.
  - Get power domains, control signals and in/out signals.

Generation of checkers with assertions in Python



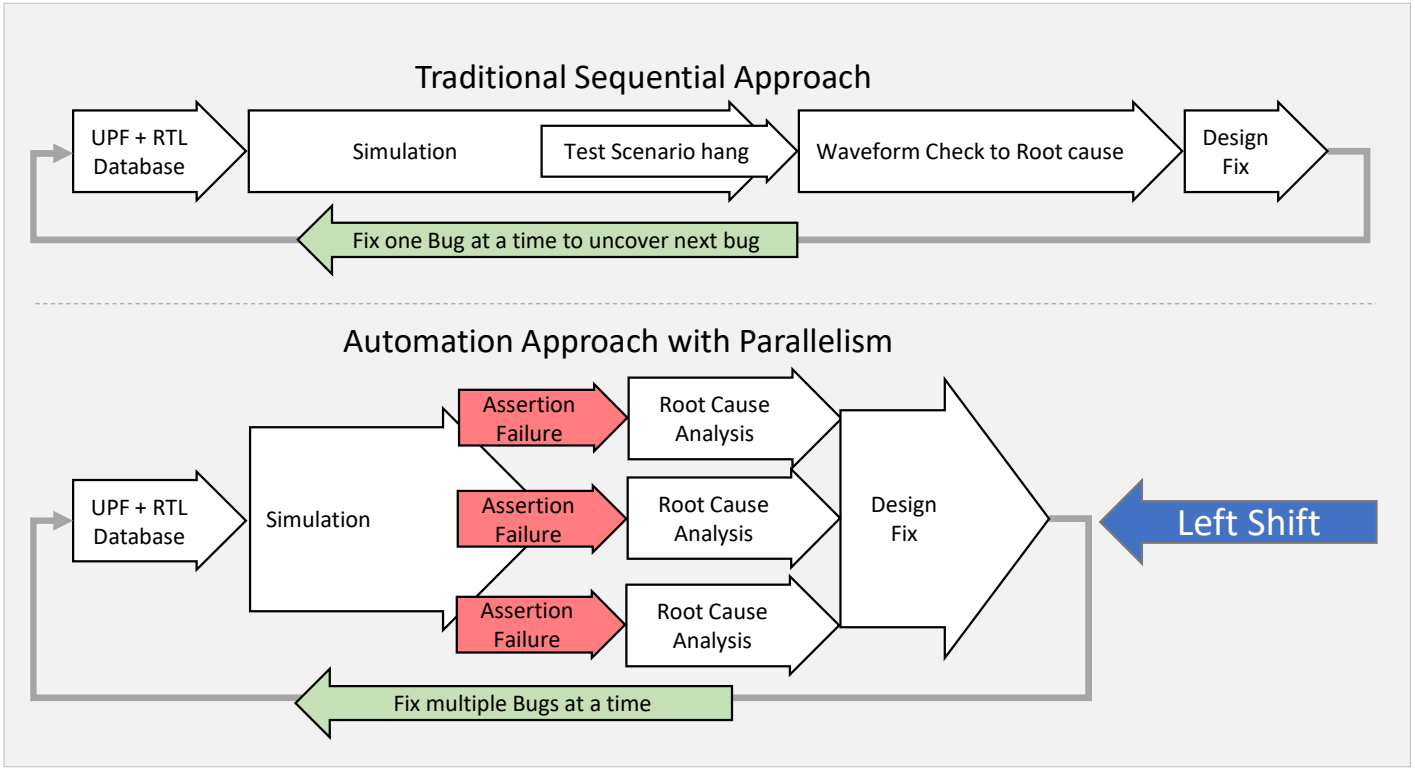
- Third Stage
  - Use the Assertion Templates
  - Generate Checkers with Assertions included.

Checkers insertion into SoC Simulation Verification



- Instantiate into the RTL hierarchy
- Bind the Checkers to Testbench
- Run Simulation.

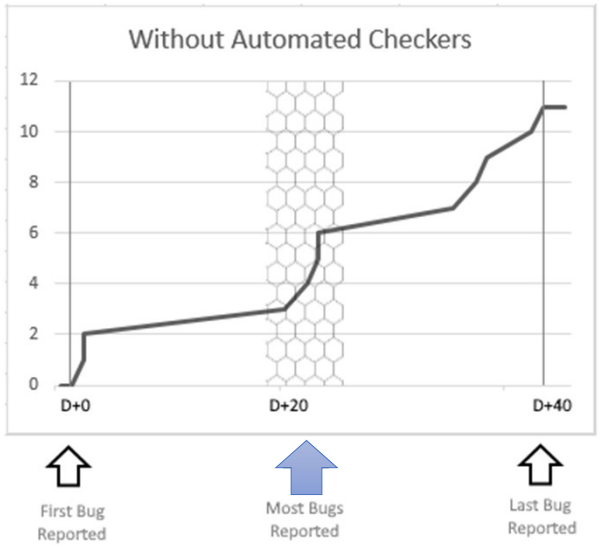
Observations with improved approach

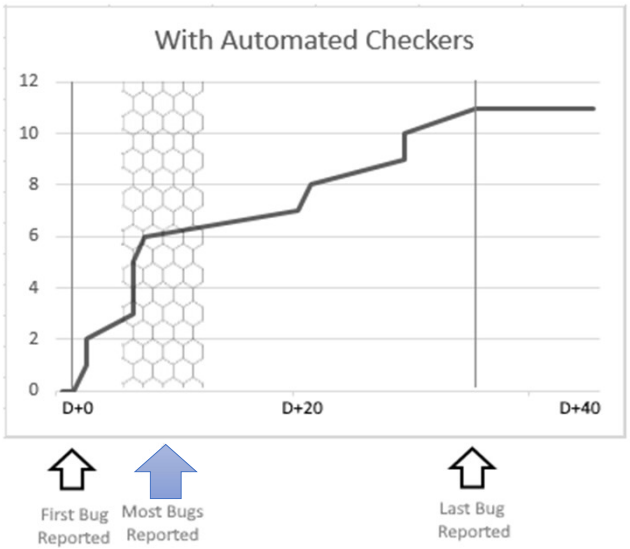


- Executed the old project again and observed time taken for RCA of each bugs.
- Re-run the same project again (pretend execution)
- Multiple checkers failures, which enabled parallel debugging

Conclusion

- Most number of the bugs was identified at the early stage of verification
- The bug trend is stabilized before the tape out.
- The introduction of automation can left-shift the verification process.





Additional Benefits

- Reusability.
  - The checker and assertion templates are prepared with reusability in mind.
  - Hence the effort necessary to setup this environment is same for future project as well
- Scalability
  - Number of Power domains in our project is 38
  - Scalability of verification effort

Questions

Back up

Tool Based Checkers

Datasheet

SYNOPSYS®
Accelerating Innovation

VCS Native Low Power with MVSIM

Accurate, Comprehensive Low Power Simulation

- Built-in automated assertions, based on years of low power verification expertise, is transparently available in VCS native low power flows. The rich set of low power assertions is generated based on the design and power intent and helps pin-point complex bugs. This feature augments

VCS Xprop Debug

When a bug is found in VCS Xpropenabled simulations, the user may still dump waveforms. Debugging an X-related simulation mismatch actually becomes easier at RTL rather than at gate level because RTL descriptions are closer to the actual functional intent of a circuit. There are different methods to debug RTL simulation failures, but typically, when VCS Xprop is enabled, the regression is run, a regression or test failure is identified, the test is rerun with waveform dumping enabled, the user goes to the point of test failure (usually identified by an assertion or monitor failure), and the user leverages signal tracing to identify the origin of the X and root cause the problem.

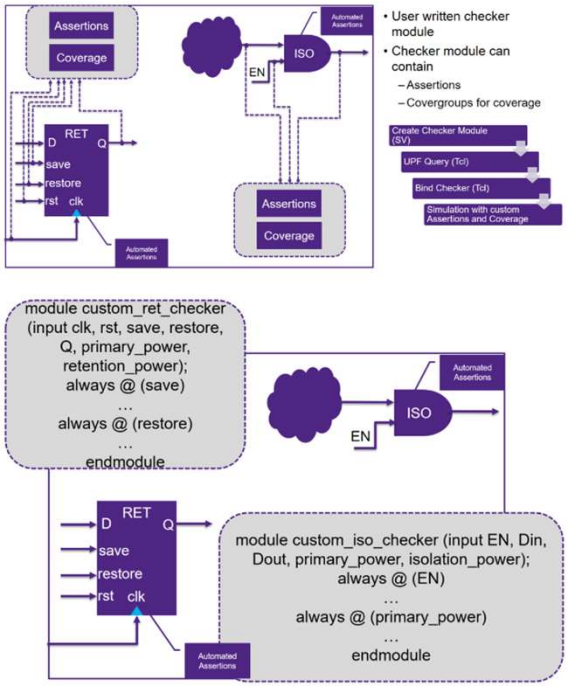


Figure 2: PAVE infrastructure

Tool Based Checkers

Mixed-signal verification of advanced SoCs using VCS AMS

By Helene Thibieroz and Pierluigi Daglio | No Comments | Posted: January 30, 2015
Topics/Categories: IP Topics, EDA - Verification | Tags: CustomSim, IP, mixed-signal, SPICE, VCS, VCS AMS | Organizations: ST Microelectronics, Synopsys

AMS Testbench

Synopsys' AMS Testbench extends digital verification strategies based on UVM to mixed-signal design. It includes predefined automated simulation generators, assertion-based checking logic, observation points defined by functional coverage, and verification planning.

Features include:

- checking connectivity between electrical and real conversion, for example, to check for asynchronous analog events within a digital testbench
- assertions and checkers for analog
- specific sources to mimic some analog characteristics

Tool Based Checkers

Confidently Sign-off any Low-Power Designs without Consequences

Madhur Bhargava (Madhur_bhargava@mentor.com), Siemens EDA
Jitesh Bansal (Jitesh_bansal@mentor.com), Siemens EDA
Progyna Khondkar (Progyna_khondkar@mentor.com), Seimens EDA

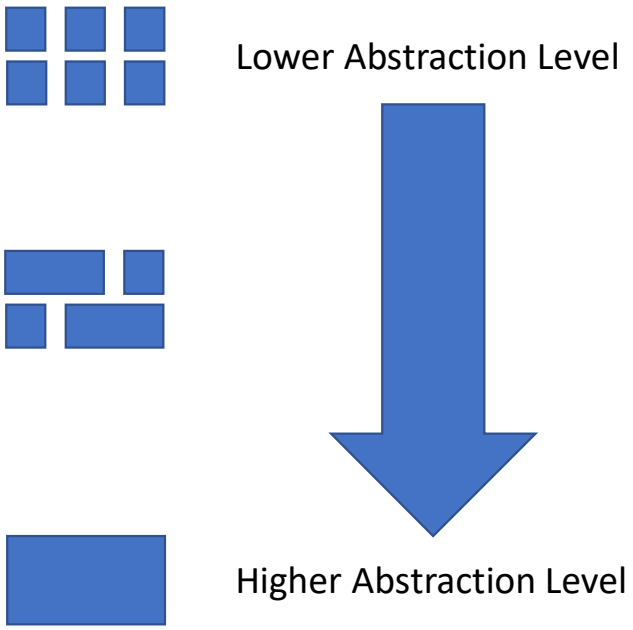
Tool generated assertions (or low-power checks) are used widely in low-power verification. However they may not be exhaustive in all the designs, as highlighted by the reasons below:

- A design can have a very specific requirement which is not being provided by the tool-generated low-power checks

- The low-power technology is still evolving and hence a new set of protocol appears every now and then, which may require a different set of checks which is not yet supported by the verification tool.

Abstraction Levels of Checkers for UPF simulation

- Static Check – Tool Based
  - Synopsys VC-LP
  - Cadence Jasper
- Dynamic Checkers - Tool Generated Checkers
  - ISO cell checkers
  - Retention Cell Checkers
- Power Use case Checkers
  - Checkers based on the intent of Power
  - (ISO AND is deployed instead of ISO OR) -> this bug will escape from above simple Checkers



References

[1] B. Wile, J. Goss, and W. Roesner, Comprehensive Functional Verification the Complete Industry Cycle. Elsevier/Morgan Kaufmann, 2005.

[2] Foster H. , 2020 Functional Verification Study, Wilson Research Group and Mentor, A Siemens Business, 2020

[3] F. Bembaron, S. Kakkar, R. Mukherjee, and A. Srivastava, 2009. “Low Power Verification Methodology Using UPF,” in Conference on Electronic SoCs Design and Verification Solutions, DVCON, pp. 228–233.

[4] Himanshu Bhatt, Kiran Vittal. Four Steps for Static Verification of Low Power Designs Using UPF with VC LP, Synopsys white paper.

[5] Madhur Bhargava, Jitesh Bansal, and Progyna Khondkar, 2022. “Confidently Sign-off any Low-Power Designs without Consequences,” DVCON2022.

[6] John Decker, Neyaz Khan, and Richard Goering, Power-Aware Verification Spans IC Design Cycle A Plan-To-Closure Approach Helps Ensure Silicon Success, Cadence Design Systems

[7] Christoph Trummer, Simulation-based Verification of Power Aware SoC-on-Chip Designs Using UPF IEEE 1801, 2010.

[8] IEEE Std 1801™-2015 for Design and Verification of Low Power Integrated Circuits. IEEE Computer Society, 05 Dec 2015.

[9] Tong Zhang, 2017. Automatic Assertion Generation for Simulation, Formal Verification and Emulation" IEEE Computer Society Annual Symposium on VLSI https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7987564

[10] A. Crone and G. Chidolue, 2007. “Functional Verification of Low Power Designs at RTL,” Lecture Notes in Computer Science, vol. 4644, pp. 288–299.