# AI-Enabled Formal Verification Flow : From Spec to Sign-off

Rinu Mathew
SmartSoC Solutions Pvt. Ltd.

## Introduction

Traditional simulation-based verification is no longer sufficient to thoroughly verify increasingly complex designs especially when trying to shift-left the design cycle.

- Formal Verification could be a great aid!

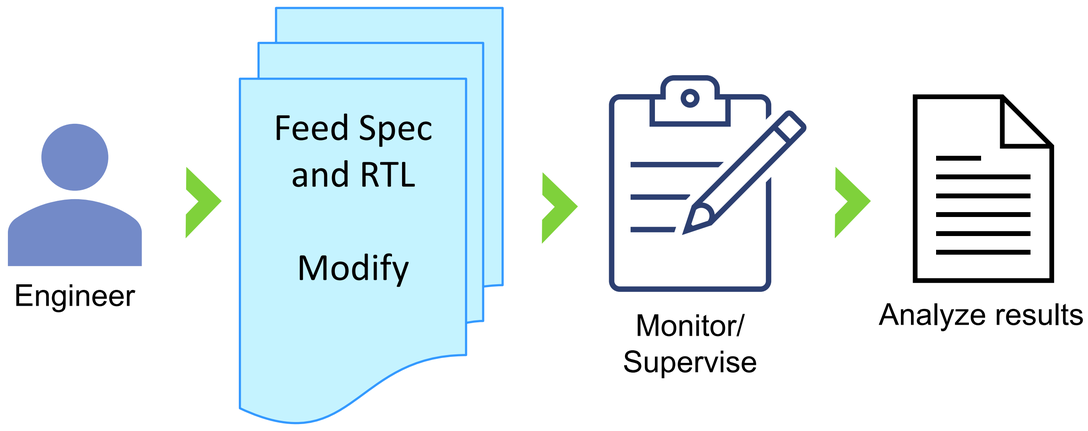
- However, formal verification's adoption is hindered by complexity and scalability issues, limiting its use to small design blocks.
- Also, fine tuning formal properties and tools require expertise.

- But every problem has a solution! (AI)

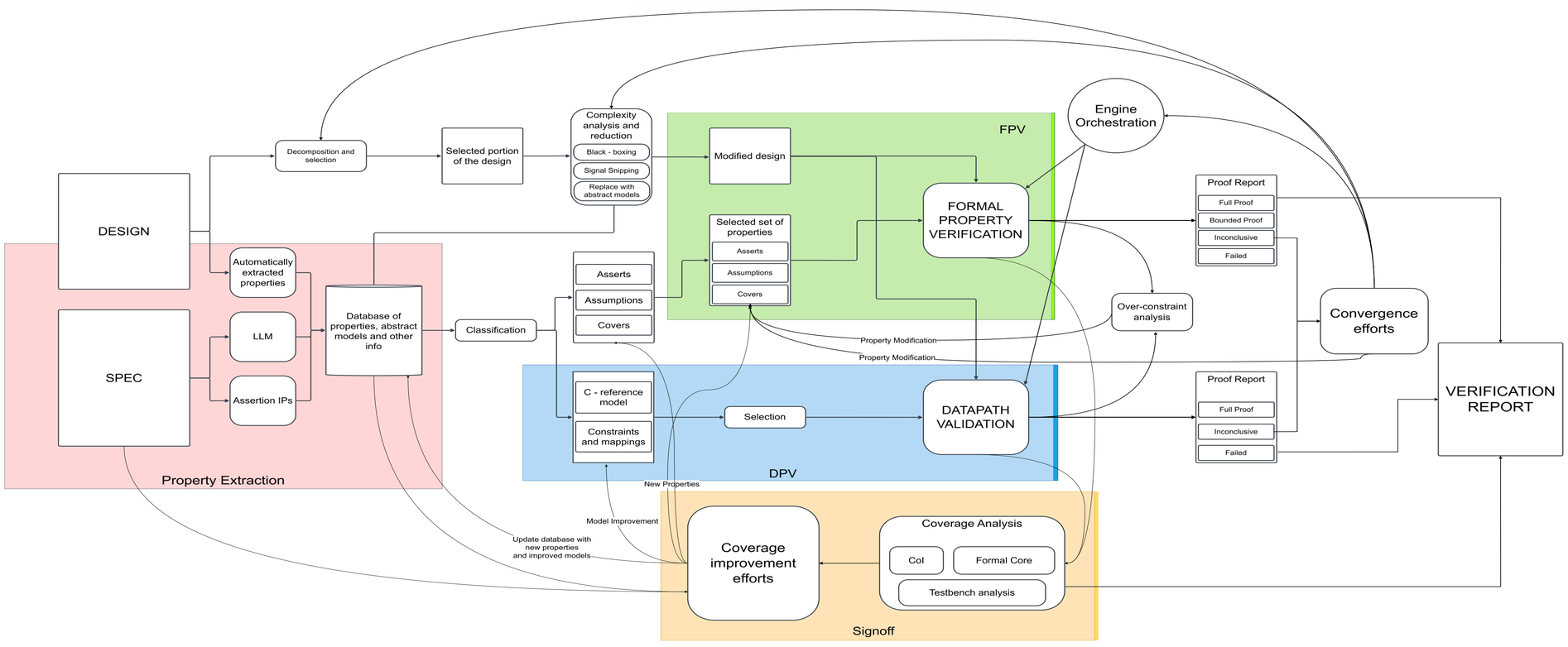
## Proposed Methodology

### Automate

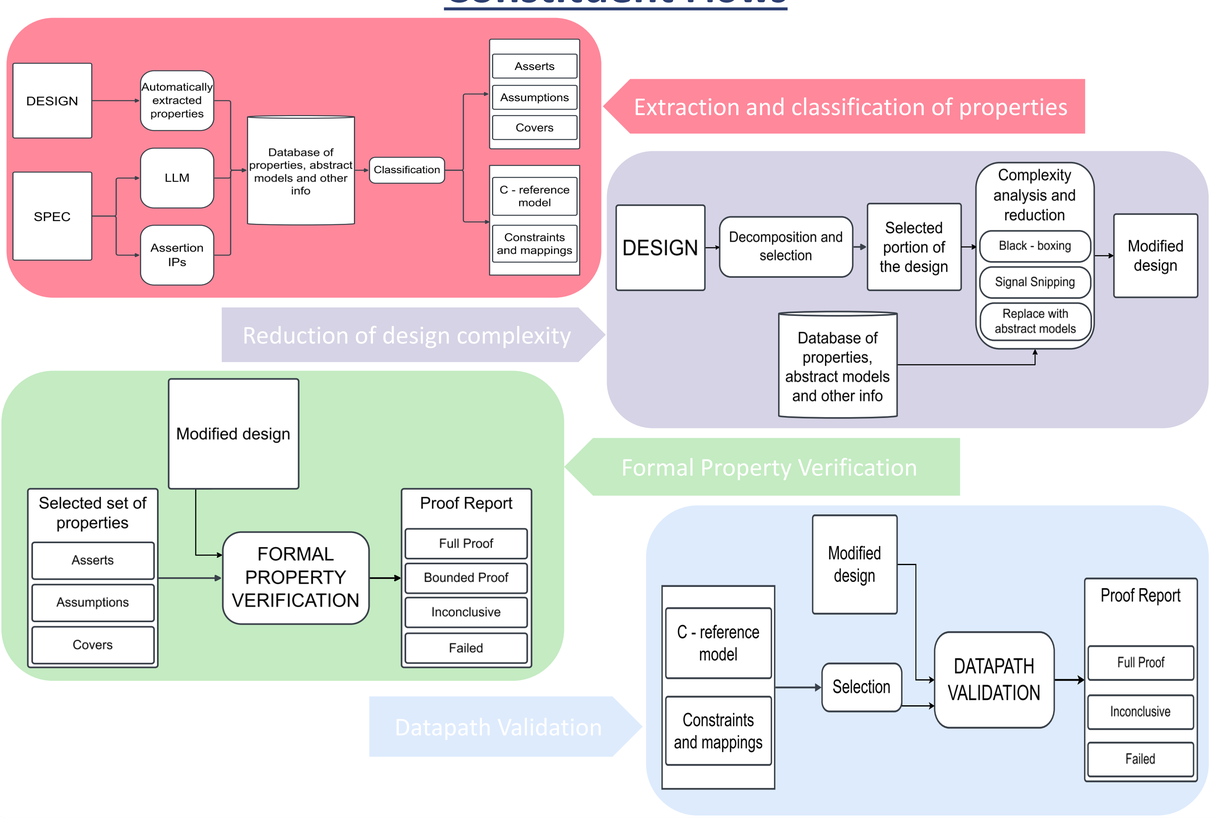
- Property generation and usage
- Tool use and tuning




## Implementation Diagram




## Constituent Flows




## Summary & Conclusion

This paper proposes a formal verification flow, integrating AI to minimise human efforts and need for skill in every step, focusing mainly on two subsets of formal verification, viz. FPV and DPV. It can be extended to other formal verification applications as well in the future. Since use of AI for generation of RTL code is already being extensively researched, it could be integrated with the formal verification flow for extreme left-shift in the front-end of the chip development cycle. AI can also be leveraged to extend the flow to include netlist creation and formal sequential equivalence comparison with RTL, lint checks, connectivity checks, low power verification etc. This could mean, in the future, that the design rolls out to the backend in just a matter of days.

Although the implementation of this flow might require extensive human intervention in the early days, especially in areas such as property generation, design decomposition and selection and proof checking, we expect it to mature relatively quickly. Anyway, we believe this step would be a milestone in removing the fear of formal verification and enable its widespread adoption by verification teams.

## References

[1] E. Seligman, T. Schubert, and A. Kiran, Formal Verification. Elsevier, 2023.

[2] "Formal Verification vs. Functional Verification," VLSI Worlds, Dec. 26, 2024. https://vlsiworlds.com/2024/12/26/formal-verification-vs-functional-verification/ (accessed Jan. 18, 2025).

[3] M. Orenes-Vera, M. Martonosi, and D. Wentzlaff, "Using LLMs to Facilitate Formal Verification of RTL," arXiv (Cornell University), Jan. 2023, doi: https://doi.org/10.48550/arxiv.2309.09437.

[4] S. Ganesh, "A Blueprint for Formal Verification - systemverilog.io," Systemverilog.io, 2015. https://www.systemverilog.io/verification/blueprint-for-formal-verification/

[5] P. Yu, "Understanding and Managing Complexity in Formal Verification," Forum for Electronics, Jul. 18, 2024. https://www.edaboard.com/blog/understanding-and-managing-complexity-in-formal-verification.2291/

We thank all our friends and colleagues at SmartSoC solutions for their help and support!