AI & ML in Semiconductor Design Verification: The Next Big Disruption

SUMIT KUMAR

SENIOR ENGINEERING MANAGER
ANALOG DEVICES INDIA

ALUMNUS- D.S.C.E, BITS PILANI,
IIM BANGALORE

Introduction

The rapid growth in SoC complexity is challenging traditional verification techniques.

AI and ML are changing the game, enabling smarter and faster verification.

This session delves into the evolving role of AI in verification.

We'll look at practical applications, benefits, and future challenges.

Challenges in Traditional Verification

Increasing Complexity of SoCs and IPs: The proliferation of multi-core, high-performance, and heterogeneous computing architectures introduces significant verification challenges. Ensuring correctness across diverse processing elements and complex interconnects demands more robust verification methodologies.

Long Verification Cycles and High Costs: Traditional verification requires extensive simulation, emulation, and formal verification, leading to prolonged cycles and higher computational expenses.

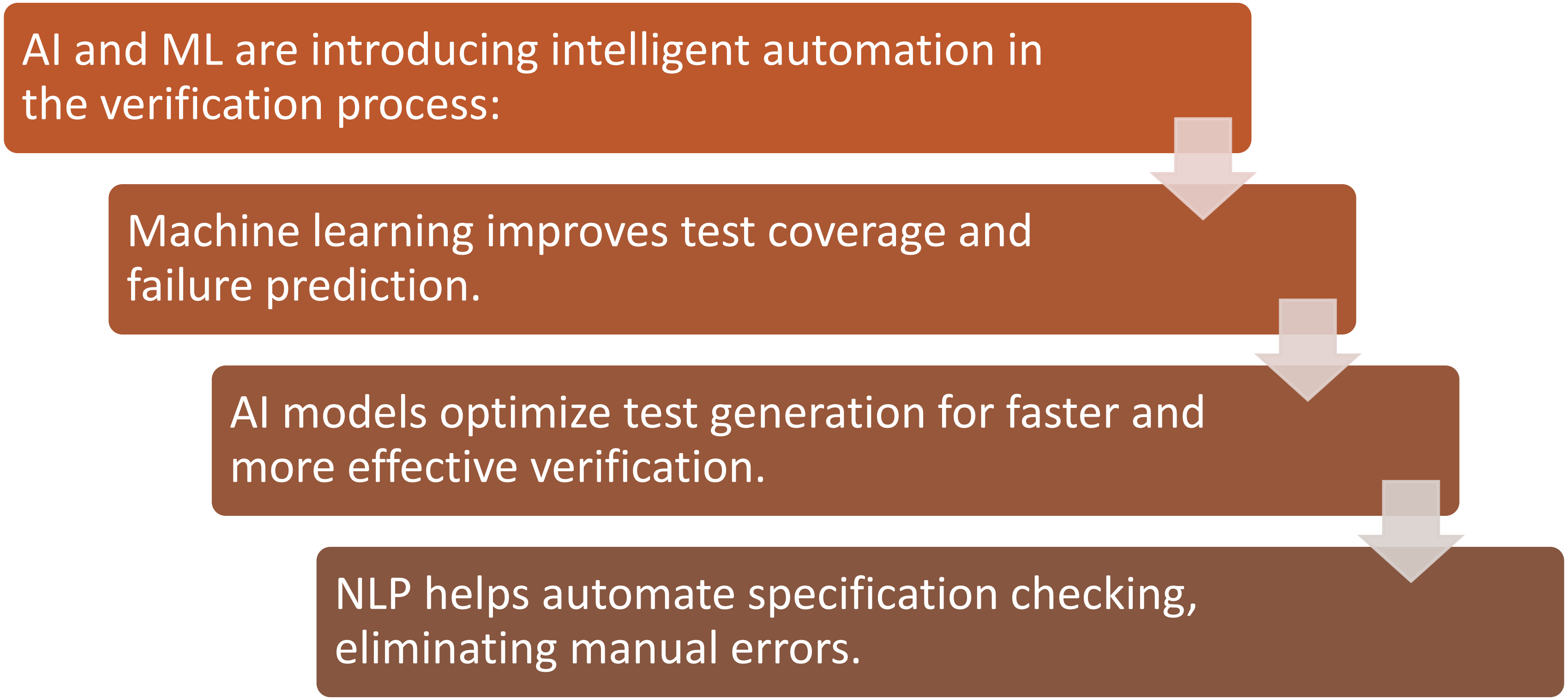
Limited Scalability of Conventional Methods: Existing verification approaches struggle with the exponential growth of design complexity, making it difficult to scale efficiently.

Achieving High Coverage: Ensuring functional coverage closure remains a significant challenge due to the vast state space of modern SoCs.

Debug Complexity: Debugging modern SoCs involves analyzing massive amounts of log data, requiring extensive manual effort to localize and fix issues.

Power and Performance Trade-offs: Traditional verification methods do not always effectively capture and analyze power consumption and performance bottlenecks in real-world workloads.

AI/ML: Revolutionizing the Verification Process



AI and ML are introducing intelligent automation in the verification process:

Machine learning improves test coverage and failure prediction.

AI models optimize test generation for faster and more effective verification.

NLP helps automate specification checking, eliminating manual errors.

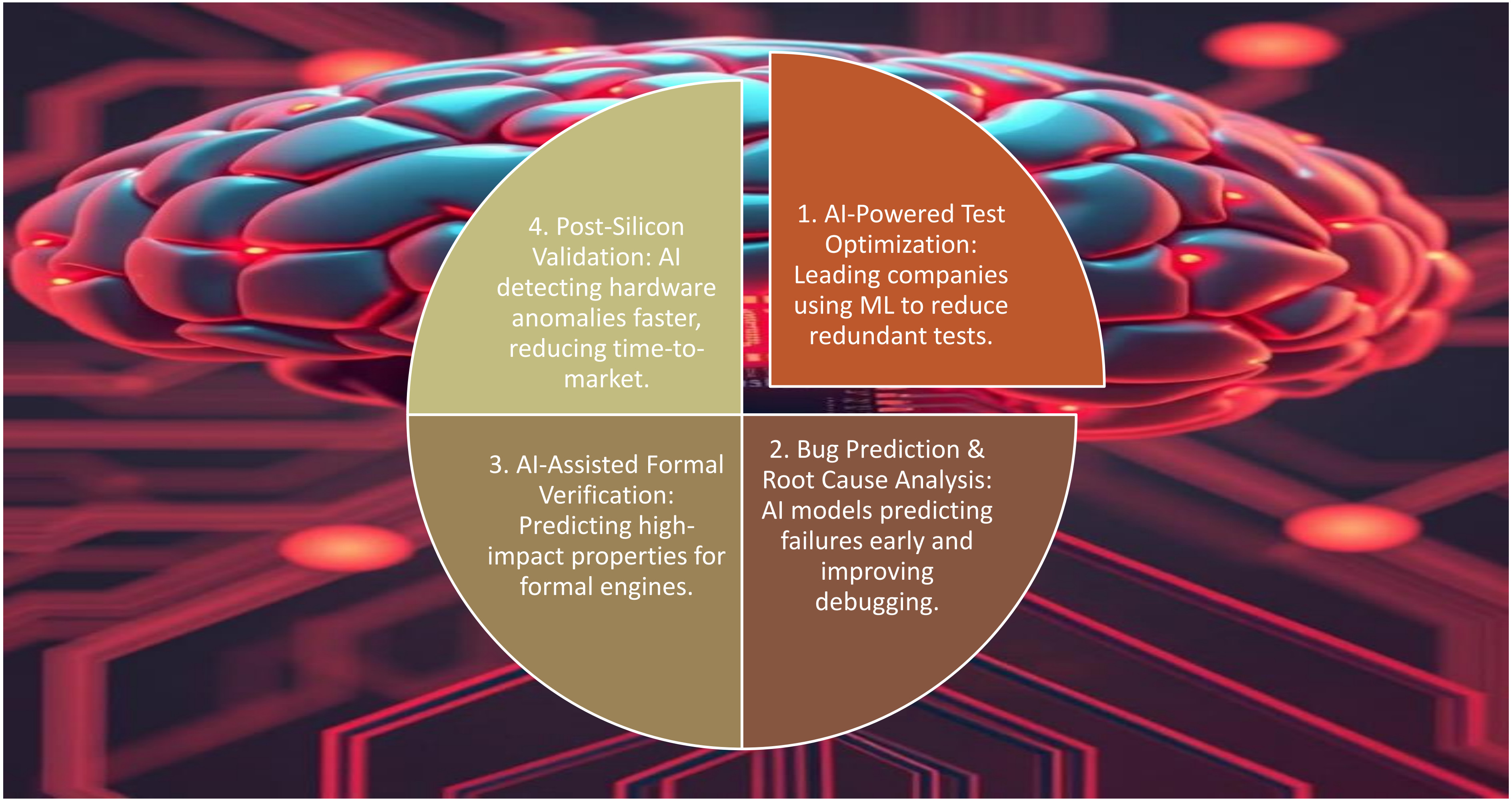
AI/ML Techniques in Verification

SUPERVISED LEARNING: IDENTIFYING FAILURE PATTERNS FROM PAST SIMULATIONS.

UNSUPERVISED LEARNING: DETECTING ANOMALIES IN TEST RESULTS.

REINFORCEMENT LEARNING: DYNAMIC TEST SEQUENCE OPTIMIZATION.

NATURAL LANGUAGE PROCESSING : AUTOMATING SPEC EXTRACTION AND ASSERTION CREATION.



1. AI-Powered Test Optimization: Leading companies using ML to reduce redundant tests.

2. Bug Prediction & Root Cause Analysis: AI models predicting failures early and improving debugging.

3. AI-Assisted Formal Verification: Predicting high-impact properties for formal engines.

4. Post-Silicon Validation: AI detecting hardware anomalies faster, reducing time-to-market.

Real-World Applications of AI in Verification

1. AI-Powered Test Optimization

Industry Problem:

Regression suites have grown exponentially. Random test generators produce huge test pools, leading to compute overuse and diminishing returns.

ML-Based Solution:

Use classification models (Random Forests, SVMs) or clustering (K-means, DBSCAN) to:

Identify redundant tests that don’t contribute new coverage.

Prioritize test sequences that hit new or uncovered functionality.

Some teams leverage Graph Neural Networks (GNNs) to model dependencies between tests and coverage points.

ROI: Reduce simulation cycles by 30–50%.

Retain the same functional and code coverage with 40–60% fewer tests.

2. Bug Prediction & Root Cause Analysis

Challenge:

Debug cycles dominate verification effort. Engineers often sift through terabytes of logs and traces.

AI Implementation:

Use supervised learning on waveform features, coverage metrics, and log signatures to:

Predict the likelihood of a test failing.

Classify failure types (timing, protocol violation, deadlock).

Use NLP models (BERT, GPT-style summarizers) to analyze UVM logs or simulation output and provide root cause insights.

Benefits: Bug triage becomes automated and scalable.

Enhances first-time-right debug triaging during nightly regressions.

3. AI-Assisted Formal Verification

Problem Area:
Formal engines are powerful but highly compute-intensive. Selecting the right properties or cutpoints is critical.
ML Intervention:
Use reinforcement learning to dynamically guide property selection.
Apply unsupervised techniques to identify meaningful observability and controllability regions in the design.
ML can predict assertion impact using structural design metrics (cone size, toggle rates, fan-in/fan-out).
Outcome:
Reduce convergence time by 20–30%.
Focus proof engines on semantically rich assertions, not just syntactic ones.
In Use: Formal teams at Qualcomm, Intel, and EDA vendors like Cadence and Siemens EDA are integrating these ML techniques.

4. Post-Silicon Validation

Verification Gap:
Post-silicon debug deals with incomplete observability, non-reproducible failures, and massive trace data.
AI Techniques:
Anomaly detection using Isolation Forests, Autoencoders, and PCA to identify outlier behavior in telemetry or trace data.
Use AI to correlate multi-dimensional sensor data (voltage, thermal, logs) to failure patterns.
Results:
AI finds corner-case bugs faster than traditional manual methods.
Reduces silicon bring-up cycles and accelerates system-level validation.
Example: AI detects subtle coherency bugs during multi-core interactions that weren’t caught pre-silicon.

AI & ML enables smart prioritization, early bug detection, test reduction, and predictive insights that scale with SoC complexity.

AI-Driven Verification Framework

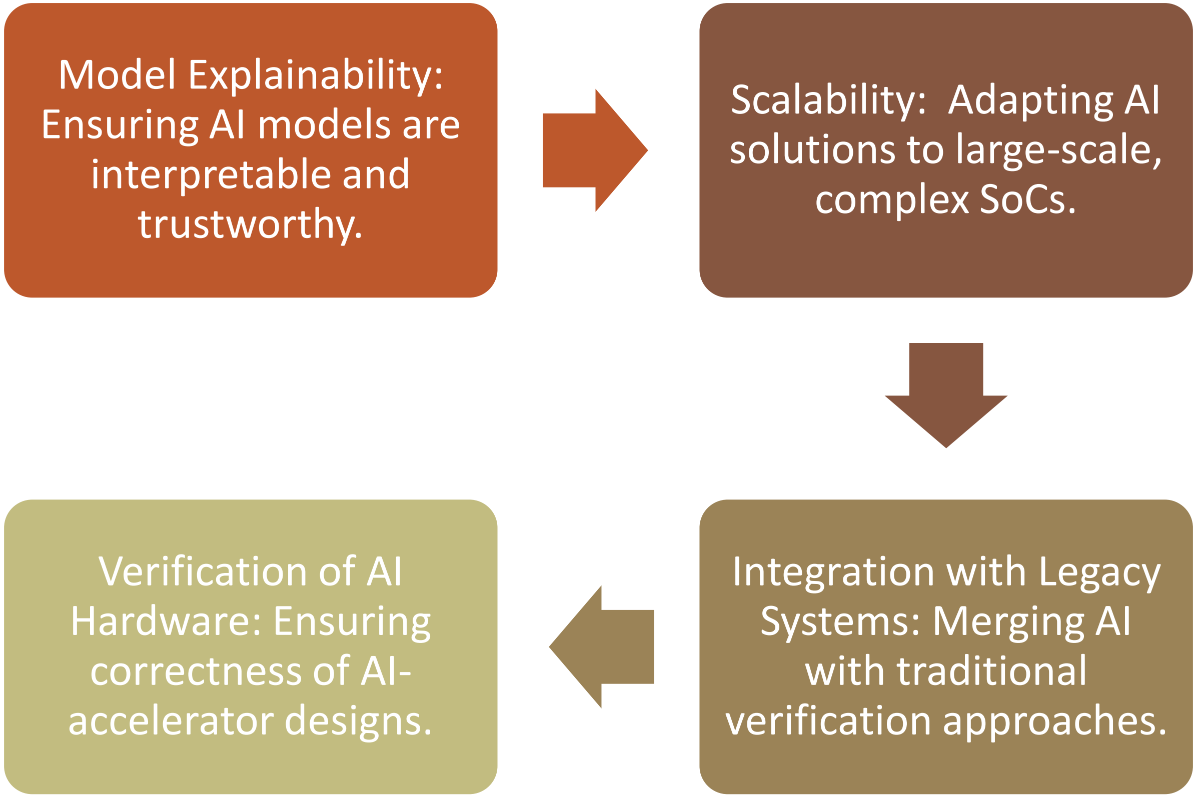
Data Collection: Gathering simulation data, logs, and test results.

Feature Engineering: Extracting useful patterns from verification data.

Model Training: Training ML models to predict failures.

Deployment: Integrating AI-driven insights for automated and intelligent verification.

Future Challenges in AI-Driven Verification



Model Explainability: Ensuring AI models are interpretable and trustworthy.

Scalability: Adapting AI solutions to large-scale, complex SoCs.

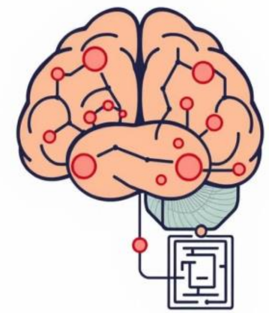
Integration with Legacy Systems: Merging AI with traditional verification approaches.

Verification of AI Hardware: Ensuring correctness of AI-accelerator designs.

Opportunities in AI-Driven Verification

- AI-assisted debugging and anomaly detection for faster validation cycles.
- Formal verification for AI-based chips (e.g., machine learning accelerators).
- Enhanced pre-silicon and post-silicon verification workflows.
- Predictive analytics to guide design teams in early-stage testing.

Conclusion



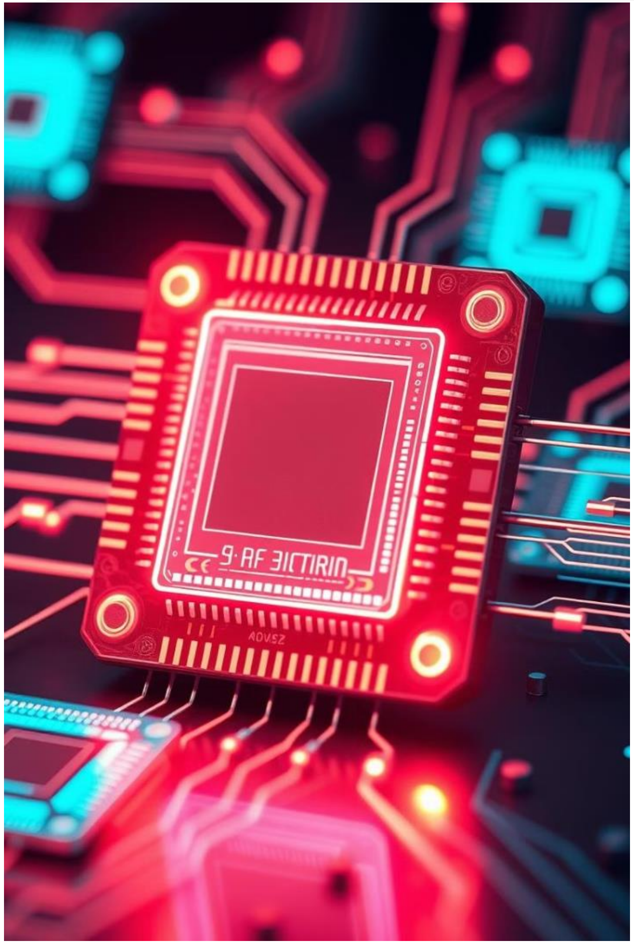
AI/ML is radically transforming semiconductor design verification, enabling:

- Smarter, faster, and more automated verification processes.
- Advanced tools that enhance test generation and coverage.
- Continued evolution of AI/ML in verification frameworks.
- Addressing future challenges in scalability, integration, and verification of AI hardware.

Q&A

THANK YOU!

FEEL FREE TO ASK QUESTIONS OR DISCUSS ANY THOUGHTS.



☎ Contact - Sumit Kumar

https://www.linkedin.com/in/sumitkumar7108/

Email- sumit7108@gmail.com