



“Bridging the Gap:

A Practical Roadmap
To Adopt **Formal Verification**
for DV Engineers”

Vivek Raheja, Harsh Vardhan Gupta, **Gilberto Migliorin**

12th Nov 2025

Legal Disclosure

CONFIDENTIAL INFORMATION

The information contained in this presentation is the confidential and proprietary information of Synopsys. You are not permitted to disseminate or use any of the information provided to you in this presentation outside of Synopsys without prior written authorization.

IMPORTANT NOTICE

This presentation may include information related to Synopsys' future product or business plans. Such plans are as of the date of this presentation and subject to change. Synopsys is not obligated to update this presentation or develop the products with the features and/or functionality discussed in this presentation. Additionally, Synopsys' products and services may only be offered and purchased pursuant to an authorized quote and purchase order or a mutually agreed upon written contract.

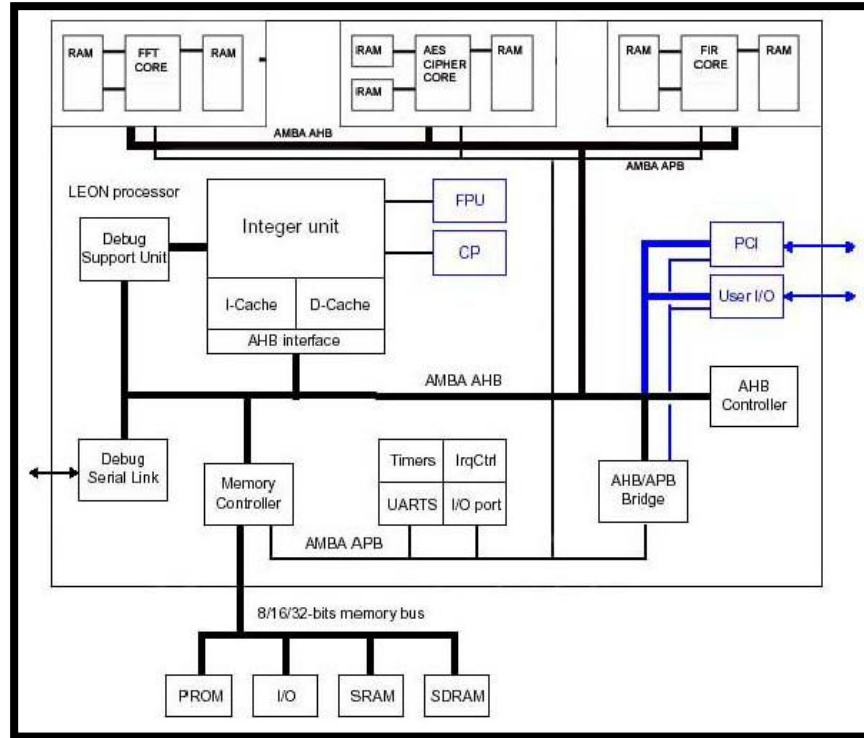
FORWARD LOOKING STATEMENTS

This presentation may include certain statements including, but not limited to, Synopsys' financial targets, expectations and objectives; business and market outlook, business opportunities, strategies and technological trends; and more. These statements are made only as of the date hereof and subject to change. Actual results or events could differ materially from those anticipated in such statements due to a number of factors. Synopsys undertakes no duty to, and does not intend to, update any statement in this presentation, whether as a result of new information, future events or otherwise, unless required by law.

Agenda

- Verification Challenges
 - 70% Challenges in SoC Verification
- Efficiency & Best Practices
 - Improving DV Efficiency & Complementing Simulation with Formal
- Formal Overview & Solutions
 - Formal in Design Verification & Synopsys VC Formal Overview
- Adoption Roadmap
 - Challenges & Solutions
- Measuring Success
 - ROI on Formal Verification & Summary/Takeaways

The 70% Challenges in SoC Verification



Ever increasing SoC design complexity –
design size, clock speed, number of clock and power domains,
leads to increasing challenges for Soc Verification

70%

Project cycle is spent in verification

70%

Verification time is spent in creating tests,
running simulation, and debugging failure

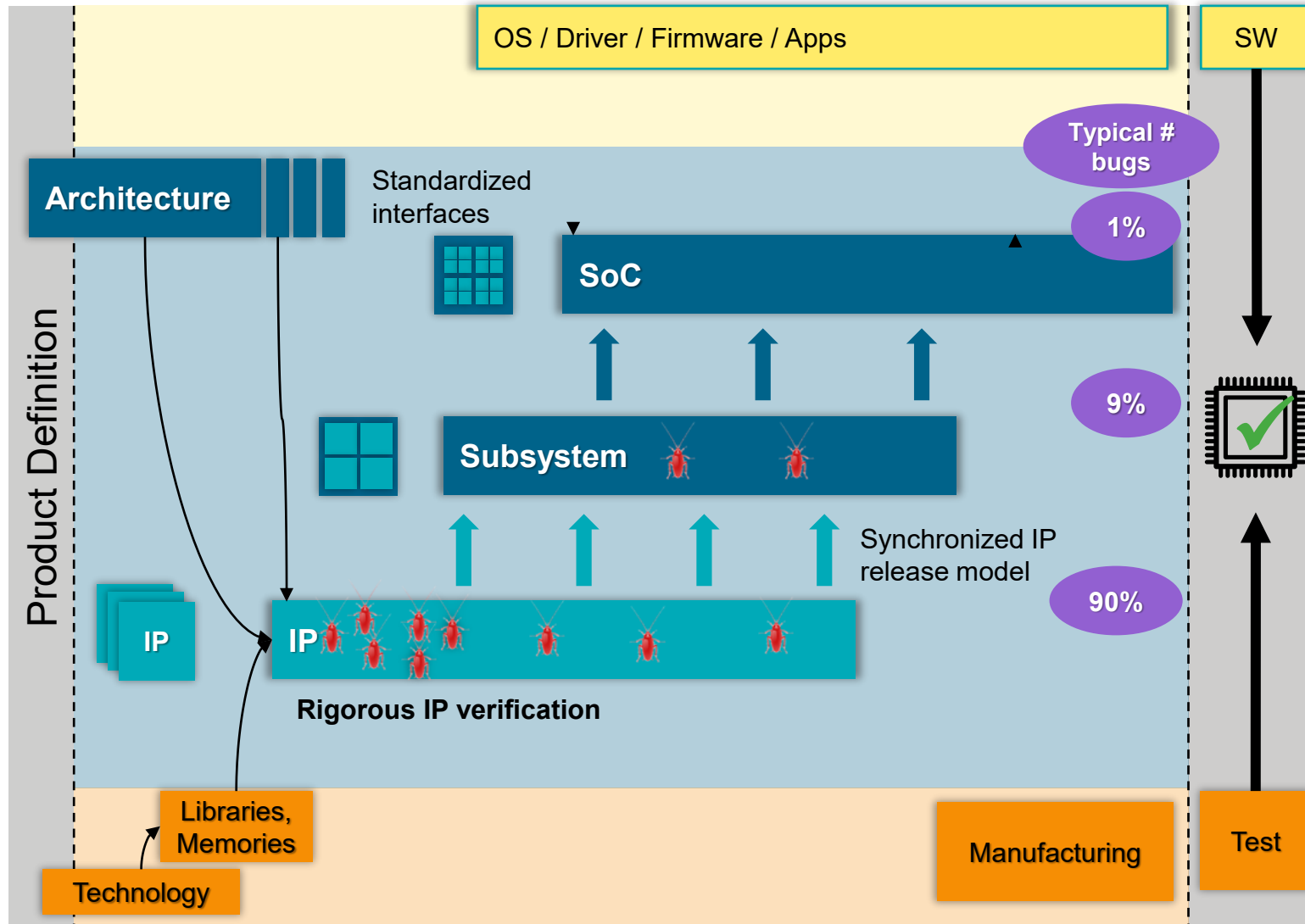
70%

Projects are behind schedule

70%

ASIC requires more than one respin

Best Practices Needed to Increase Verification Efficiency



Industry Best Practices

Rigorous IP verification

Find bugs earlier



Architecting for Verification

Prevent bugs with correct-by-construction design



Advanced Debug

Higher abstractions across time domains

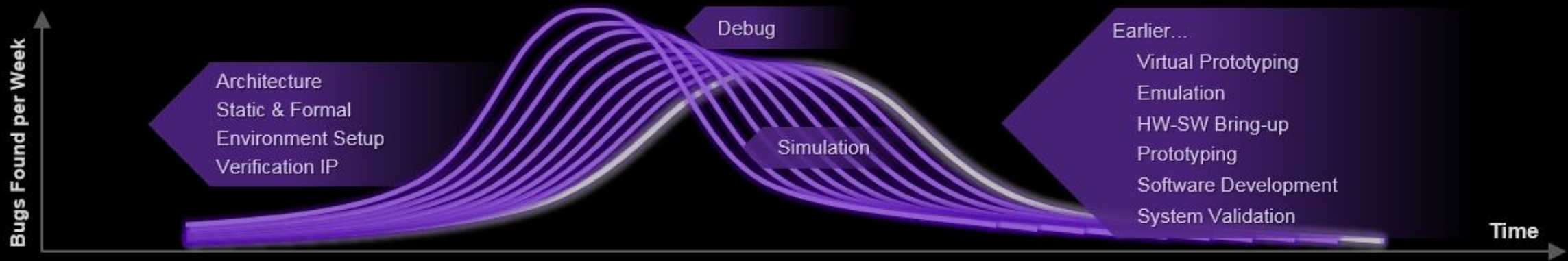


Improving Verification Efficiency



Enabling shift-left, reducing bug rate, preventing bug escapes

Formal Verification: Key Enabler for “Shift Left”

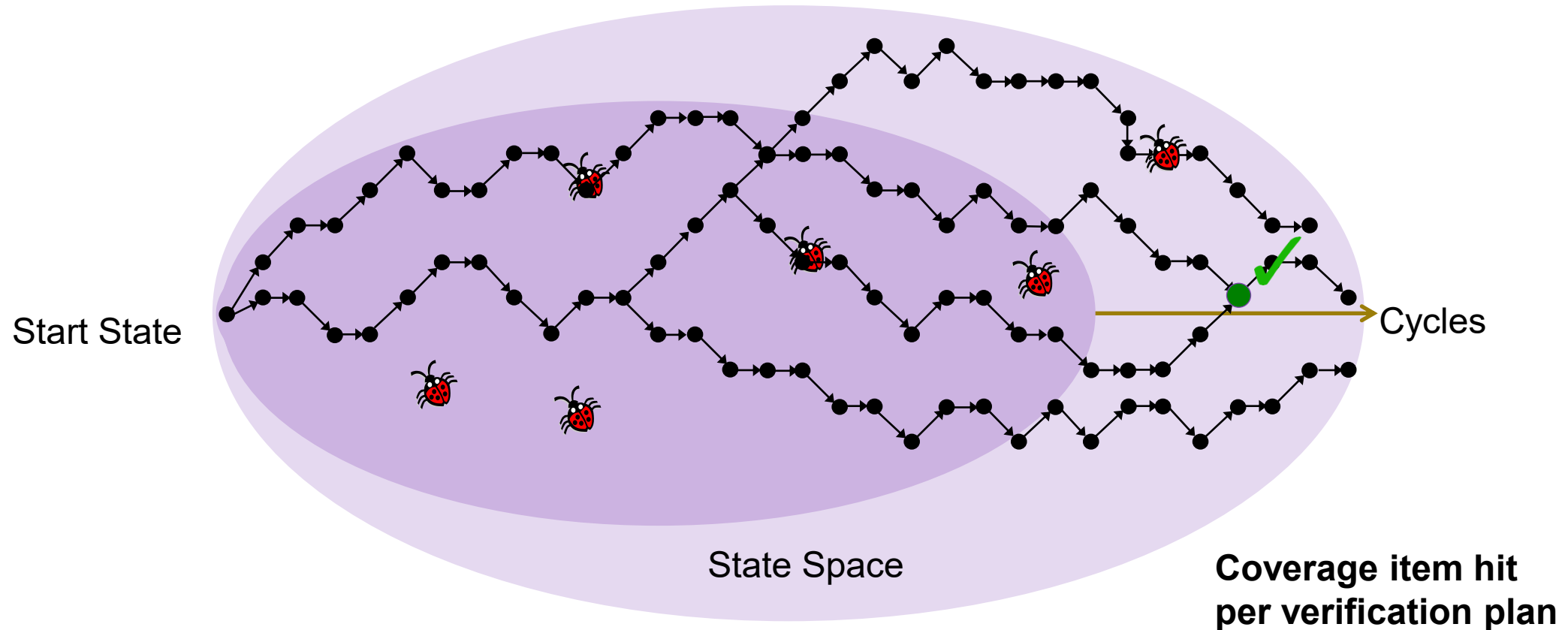


Find & fix bugs as early as possible

- Exhaustive verification
- Find hard bugs & corner-case bugs
- No testbench required
- Early-stage bug-hunting

Complementing Simulation with Formal (Co-exist)

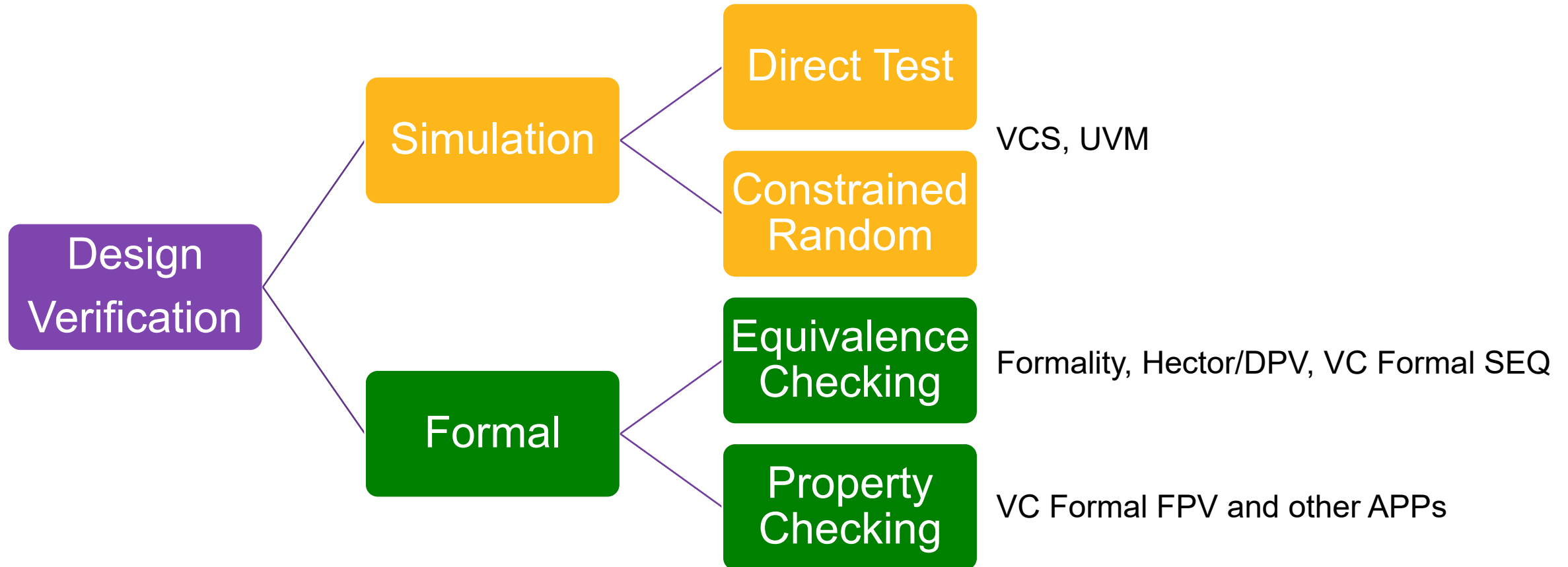
- Formal explores design state space exhaustively
- Simulation is not exhaustive but explores deep sequential behavior



Why Need Formal Today?

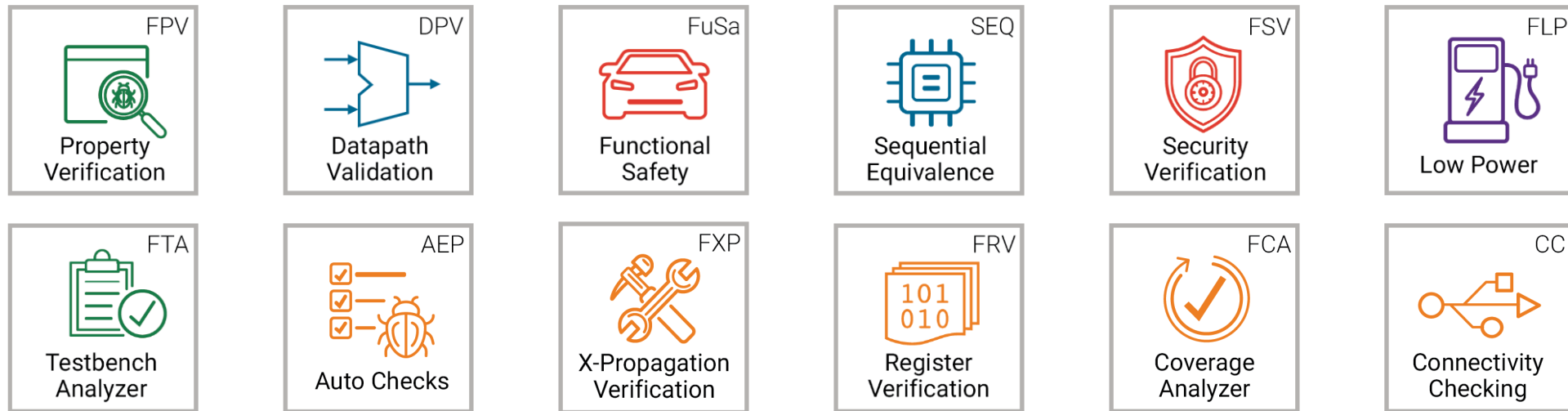
- Formal is not new! It has been part of signoff flows for a long time
 - VC Formal is Synopsys formal verification product
- Complexity is different compared to the old times
 - Advanced process nodes (How many times can you do ECO?)
 - IP's and SS's size growing over the years
 - Security, Safety, Automobile, Medical devices have zero bug tolerance

Formal in Design Verification



Synopsys VC Formal: Innovative Formal Verification Solutions

VC Formal Apps Can Be Used Throughout the SoC Flow



Block/IP

Subsystem

SoC

Challenges for Adoption of Formal

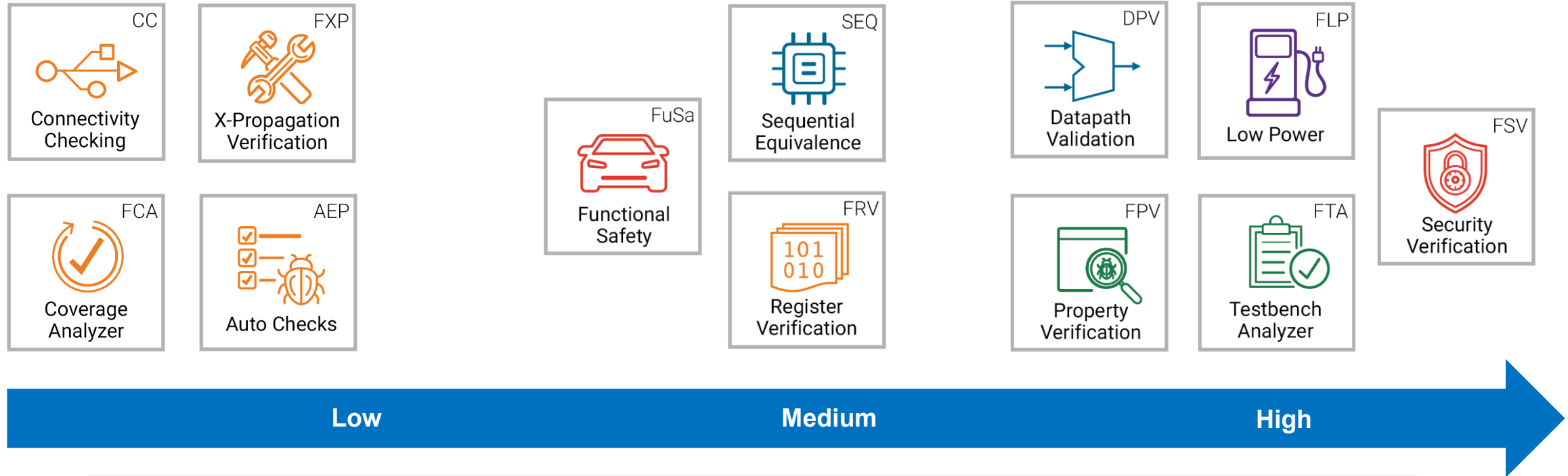
- Demands **strong expertise** in formal methods, assertion writing, and tool usage
- **Lack of support**, as complexity grows, causes engineers to quit formal
- **Starting with advanced Formal** apps requires dedicated expertise and resources
- **Debug & resolving complexity issues leads to** slow adoption

Some Solutions for Adoption of Formal

- Use Formal as a **complement to simulation**, not a replacement
- Begin with **low-effort, high-impact apps**: Unreachability, Connectivity, Auto-checks, X-Prop, Assertion-based IPs
- Progress to medium and **advanced methodologies** (Formal Property Verification, Datapath Verification, Sequence Equivalence Check etc) for comprehensive formal adoption
- **Use AI enabled** tool features to generate Properties/Constraints
- Establish **Formal Champions with SIGs** to drive expertise, culture, adoption, quality, and formal success
- Build **expertise and methodology** around **tackling complexity** in Formal Verification

Synopsys VC Formal: Innovative Formal Verification Solutions

VC Formal Apps Adoption Effort – Formal Expertise Not Always Required

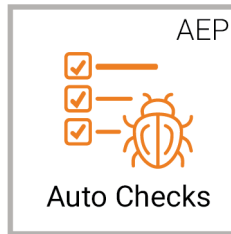
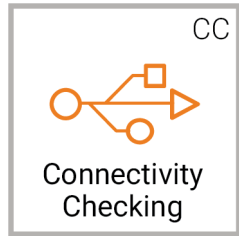


Adoption Effort: In terms of formal expertise and testbench required to apply the specific APP

AIP (Assertion based IP) :- We have AIPs used for protocol compliance and bus certification.

Synopsys VC Formal: Innovative Formal Verification Solutions

VC Formal Apps Adoption Effort – Formal Expertise Not Always Required



Low

Medium

High

LOW EFFORTS HIGH GAINS/RETURNS

AIP (Assertion based IP) :- We have AIPs used for protocol compliance and bus certification.

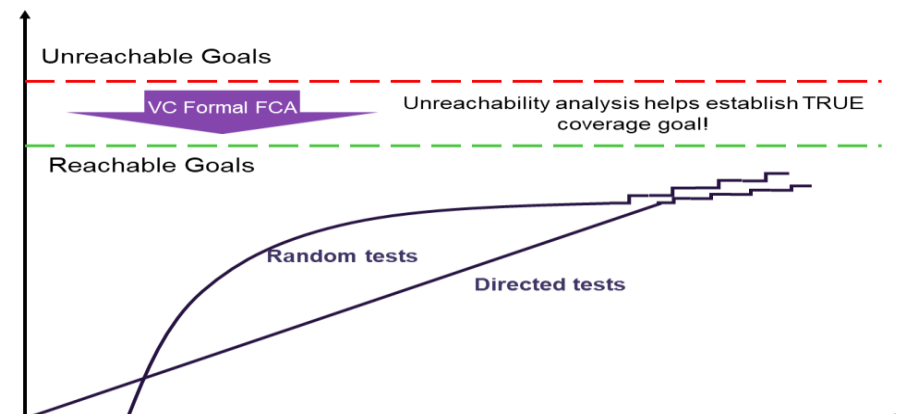
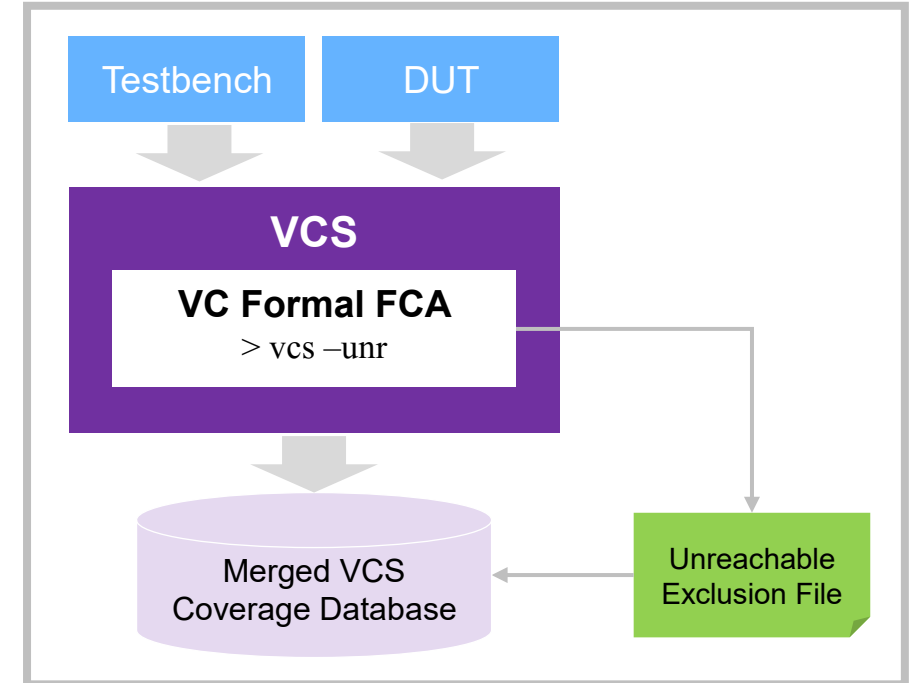
VC Formal FCA: Formal Coverage Analyzer - UNR

FCA BENEFITS

- Identifies unreachable targets early to achieve faster coverage closure
- Eliminates unnecessary coverage goals, saves simulation time and effort
- Exposes hidden design bugs
- No formal knowledge required

FCA FEATURES

- Supports VCS coverage metrics, such as line, condition, branch, FSM, toggle, and cover group
- Use model supports early to late design & verification phases
- Ability to read in VCS compile and coverage databases



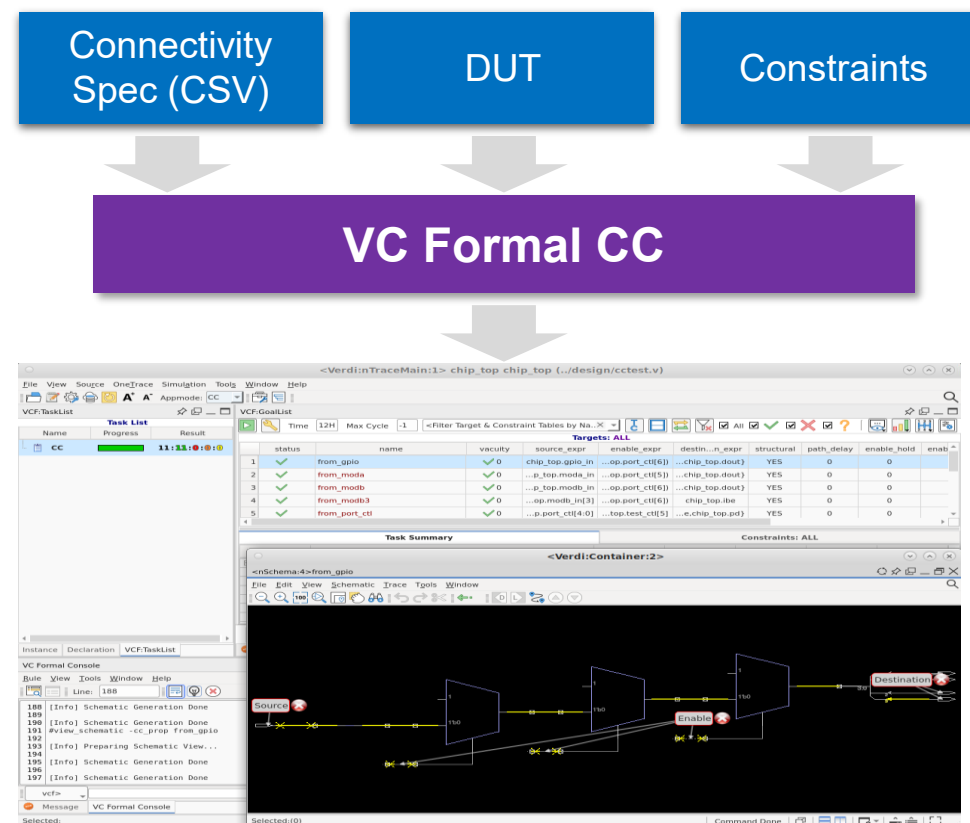
VC Formal CC: Connectivity Checking

CC BENEFITS

- Verifies connectivity in many applications, such as SoC I/O, scan mode, register to debug bus, pin muxing & demuxing, configurable interconnects, etc.
- No formal expertise and SVA knowledge required
- Fast convergence, applicable to SoC level

CC FEATURES

- Supports encrypted IP modules
- Allows single direction connectivity checking
- Automatic black-boxing of unrelated logic for fast performance
- Measures toggle coverage in the path of proven connectivity





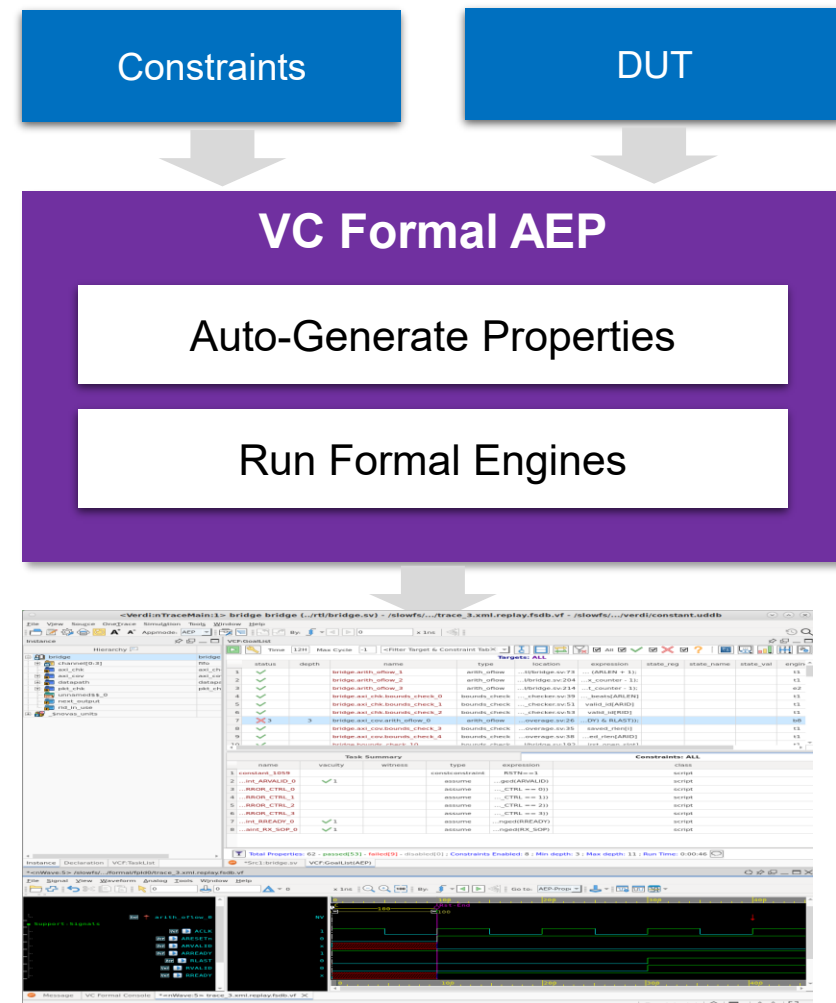
VC Formal AEP: Automatic Extracted Property Checks

AEP BENEFITS

- Easiest to adopt, no formal expertise required
- Automatic property generation, no formal testbench required
- Enable finding functional bugs early in the design cycle

AEP FEATURES

- Automatic generation of comprehensive check types:
 - Arithmetic Overflow
 - Bus Contention
 - FSM Livelock/Deadlock
- Integrated with Verdi debugging platform





VC Formal FXP: Formal X-Propagation Verification

FXP BENEFITS

- Exhaustively analyze the impact of X propagation without testbench
- Ensure unknown values can not propagate to important design points
- Reduce effort to debug RTL/gate-level simulation mismatch due to x-optimism

FXP FEATURES

- Automatically generates checks for observation points
- Easy configurability of X sources and destinations
- Shortest path to failure and root-cause tracing reduce debugging time

Injection &
Observation Config

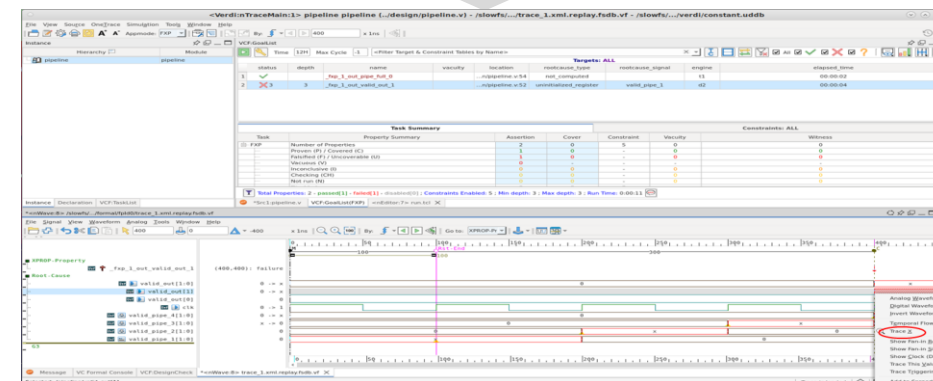
DUT

VC Formal FXP

Auto-Generate Properties

Run Formal Engines

Trace X Root Cause Analysis



AIP: Assertion based IP

AIP BENEFITS

- Faster protocol compliance with minimal manual effort.
- Detects corner-case bugs missed by simulation.
- Quick deployment—only configuration needed.

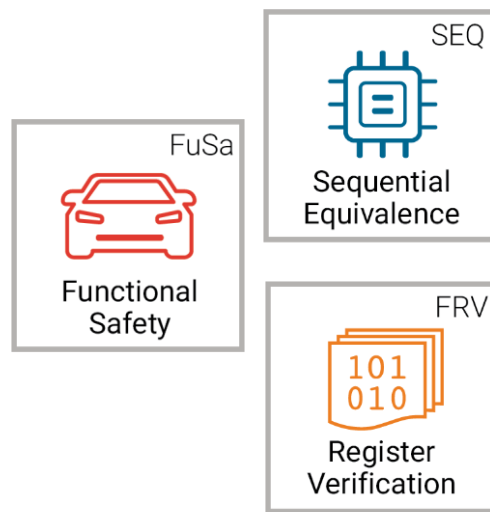
AIP FEATURES

- Pre-verified, configurable assertion modules.
- Works across Formal, Simulation, and Emulation.
- Simple setup with plug-and-play integration.
- Available in installation

	Title	Version
AMBA	APB	APB2, APB3, APB4
	AHB	AHB-Lite, AHB (Full), AHB5
	AXI3	
	AXI4	AXI4, AXI4-Lite
	AXI5	AXI5, AXI5-Lite
	AXI_STREAM	
	ACE	ACE, ACE5, ACE-Lite
	CHI	CHI-A, CHI-B, CHI-C, CHI-D, CHI-E
	GIC	GICv3, GICv4,
Others	I2C	
	SPI	
	SPMI	

Synopsys VC Formal: Innovative Formal Verification Solutions

VC Formal Apps Adoption Effort – Formal Expertise Not Always Required



Low

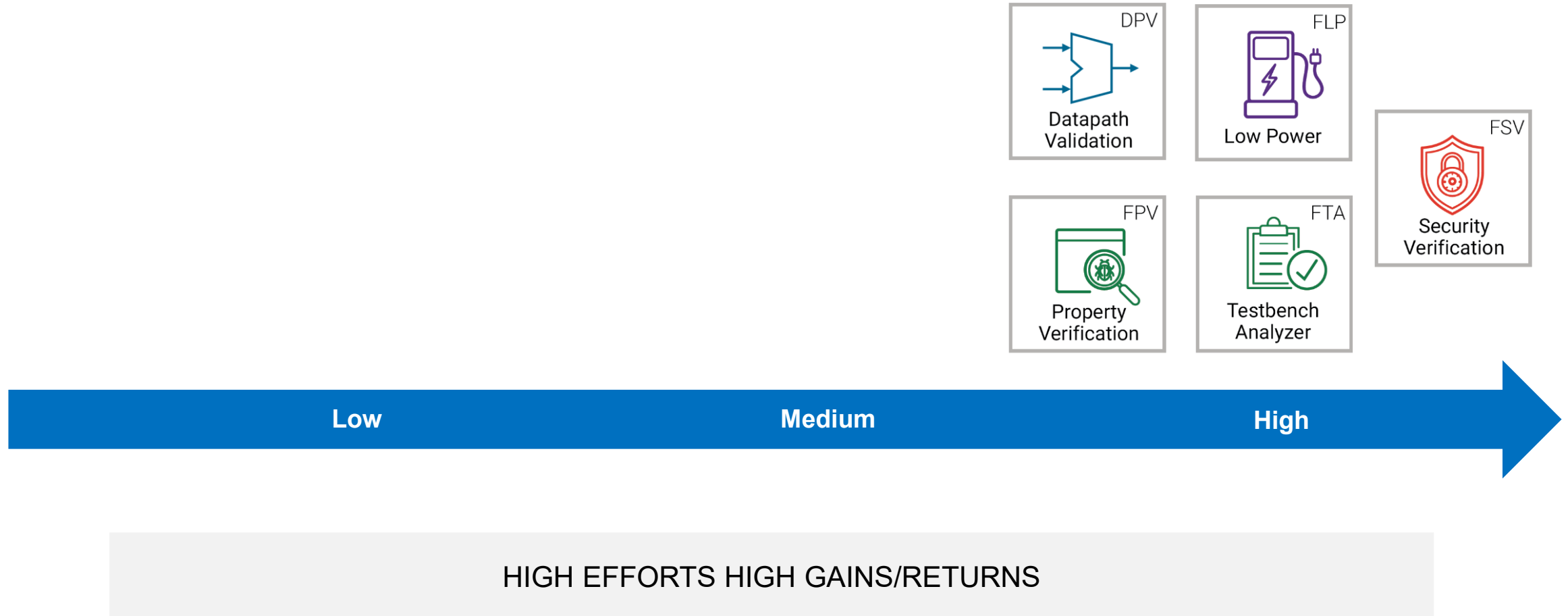
Medium

High

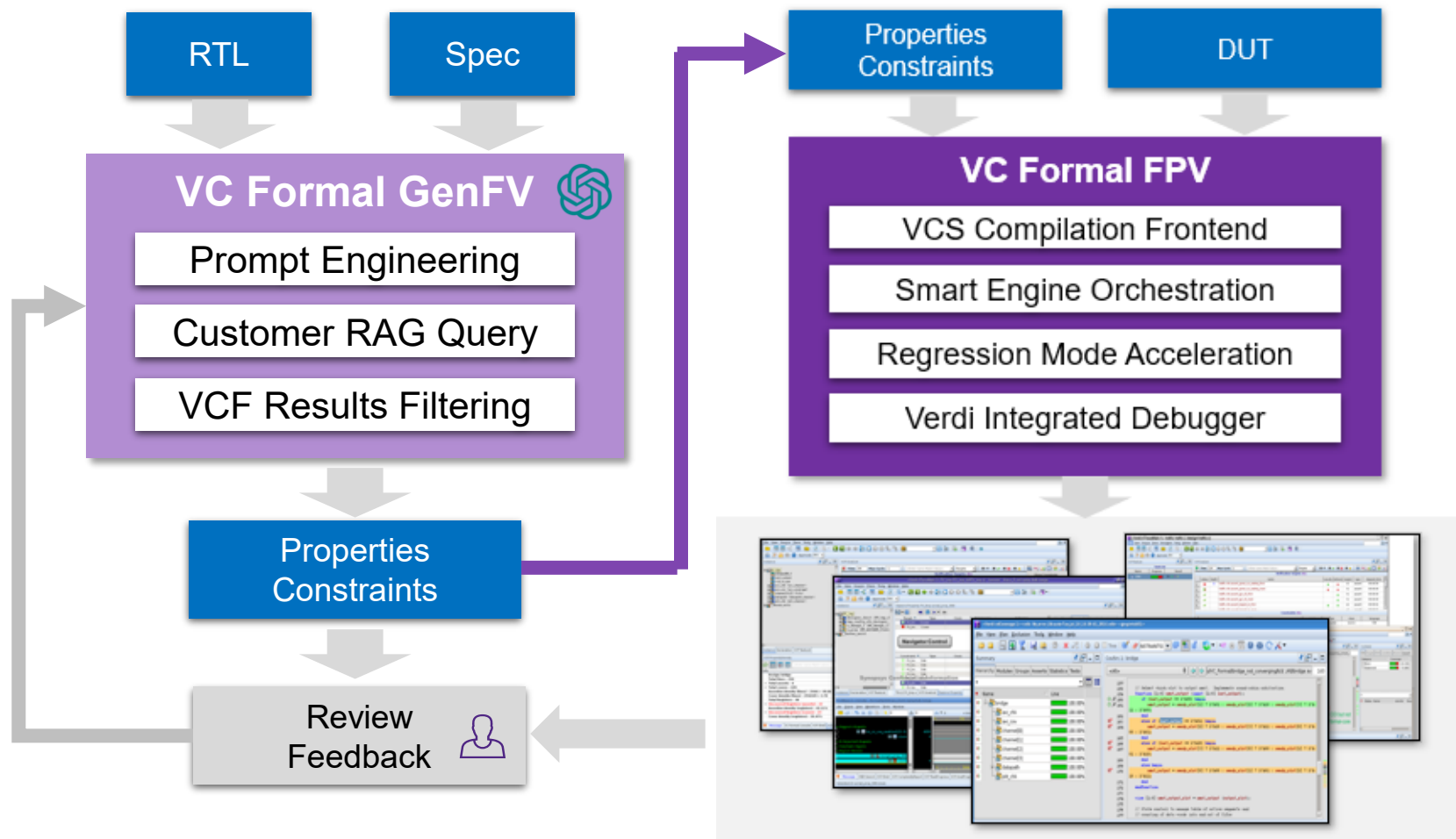
MEDIUM EFFORTS HIGH GAINS/RETURNS

Synopsys VC Formal: Innovative Formal Verification Solutions

VC Formal Apps Adoption Effort – Formal Expertise Not Always Required



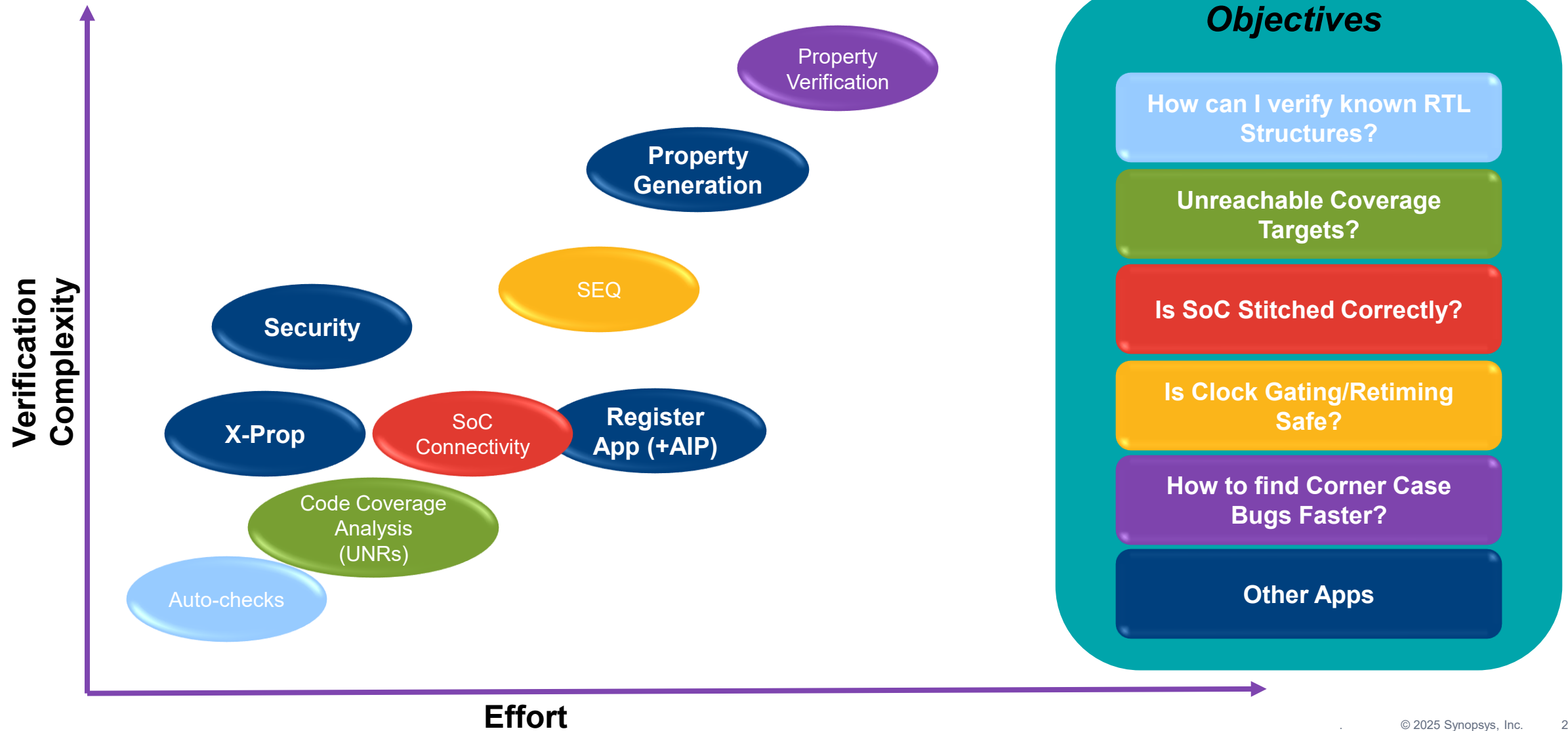
VC Formal GenFV User Flow



Key Capabilities:

- Input spec:
 - Natural Language Description on What to Verify
 - No Specific Format Requirement
 - No Need to Focus on Specifics
- Property & Formal Testbench Generation
 - Prompt Engineering, Customer RAG Query, and Under-the-Hood Results Filtering Improve QoR
- Interactive User Interface
 - Accept, Reject, Modify, Delete, and Regenerate Properties
- Seamless flow with VC Formal/Verdi
 - Property Verification and Debugging
- Chat Interface
 - Additional Context and Support

ROI on Formal Verification





Summary

- **Complementing Simulation with Formal** – A Key Enabler for ‘Shift Left’ and for Improved Verification Efficiency
- EDA is **easing FV adoption** by integrating **AI** tools – leverage it.
- **Formal Pathway Methodology:**
 - Step1: Low-Hanging Fruits – Quick Wins with Minimal Effort
 - Unreachability, Connectivity, Auto-checks, Assertion IPs, X-Prop
 - Step2: Intermediate Formal – Expanding Scope and Depth
 - Register Validation, Sequence Equivalence Checks, Functional Safety
 - Step3: Advanced Formal – Deep Design Analysis
 - Formal Property Verification, Datapath Verification, Testbench Analyzer, Low-power
- Teams to build **expertise and methodology** around tackling complexity in Formal Verification
- Establish **Formal Champions** to drive expertise, adoption, quality, and formal success

SYNOPSYS®

Thank you