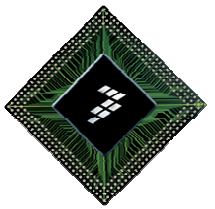
May 16th, 2007

Static Verification Flow



CDNLIVE! 2007, Session 5.9 Logic Design

A. Dabbagh, H. Luepken, N. Bossemeyer (Freescale)
K. Fotouhi, C. Komar (Cadence)

cadence™ freescale semiconductor ™

Introduction

► Recent advancements in EDA tools have put a new emphasis on static and automated formal verification to augment current functional verification flows.

► Rule set is defined as Reuse Standard or Company Standard to qualify the RTL.

► What is Static Verification ?

- Functional Formal Verification (FSM checks, Dead-Code,…).
- Static / Structural Checks (Coding Style, Synthesizability, Testability-checks…).
- Timing Constraints Checks (False-Path, Multi-Cycle-Path,…).
- Implementation Formal Verification (Clock Domain Crossing (CDC), Multi-Cycle-Path, False-Path,…).

► What is a Static Verification Flow ?

- Introduces complimentary technologies.
- Automated (less user time, more CPU time).
- Driven by a methodology:
  - **Q:** Who should use it ? **A:** Module Owner, Integrator, Verification Engineers,…
  - **Q:** when should it be used? **A:** Project Cycle,…

freescale semiconductor ™

Problems faced during traditional design flows

► Traditional Gate-Level simulations have often been used to detect a class of problems that have been impossible/difficult to catch in RTL simulation. Typically:

- Not very robust.
- Starts very late.
- Significant manual effort to develop (False-Path, Multi-Cycle-Path).

► Structural rule checkers can not completely identify all types of problems found late on the process (For example: asynchronous inputs).

► Tool limitations:

- Old Lint architecture.
- Limited Clock Domain Crossing (CDC) checks.
- Limited timing constraint check.

► Insufficient knowledge about rule set and the kind of checks applied.

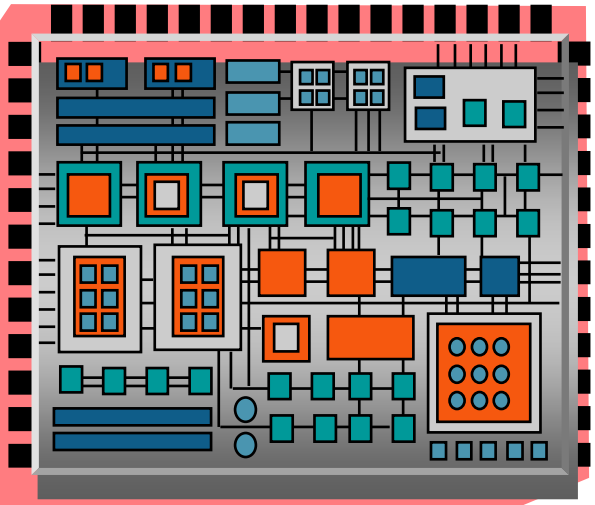
Cost and Pain of Fail

- Direct Costs
  - Foundry/Mask costs.
  - Costs for Engineering Change Order (ECO) implementation, verification and test.
    - 7 Persons/Day for one week effort (mid size ECO), several thousands **€**.
  - Cost for Application Engineering/Marketing/Bureaucracy effort.
  - Cost for re-qualification if the device requires qualification.

- Indirect Costs
  - Loss of Reputation/Business
    - Hard to measure but can be disastrous if actual/future products are denied by customer.

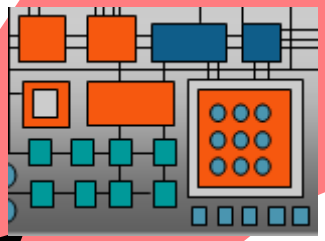
Problem Solution (Complementary Technologies For Different Stakeholder)

Chip Integrator/Verification Engineer



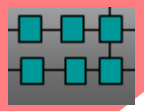
❑ **Full chip level**

- ➢ Consistency of constraint files ?
- ➢ False-Path, Multi-Cycle-Path ?
- ➢ Equivalent Check ?
- ➢ Clock Domain Crossing ?
- ➢ …



❑ **Integrated Cluster / blocks**

- ➢ Clock domain crossing correct ?
- ➢ Crossing Power isolation ?
- ➢ Power shut down and wakeup protocol ?
- ➢ State retention mapping after synthesis ?
- ➢ Equivalent check (diff. power domains) ?
- ➢ Equivalent check (RTL vs. Gate)?
- ➢ …..



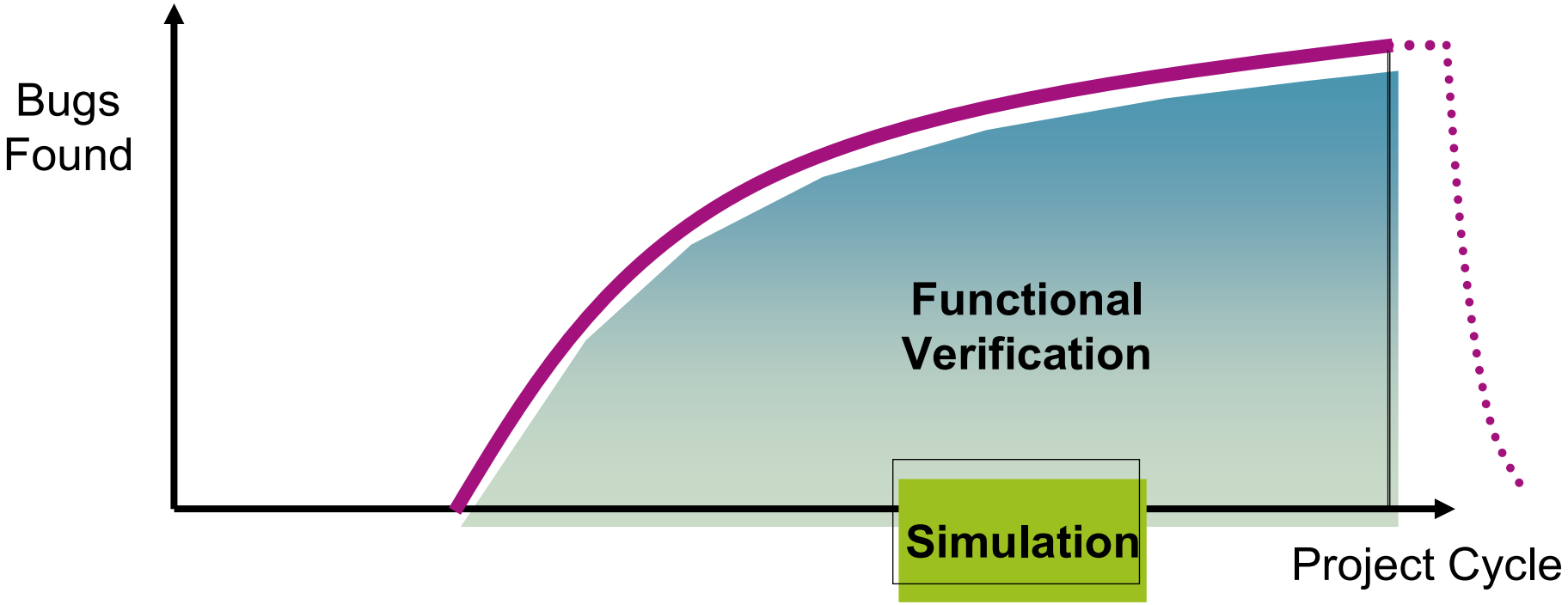
❑ **RTL block level**

- ➢ Coding style ? Semantic correctness ?
- ➢ Synthesizability ? Testability ?
- ➢ Simulation and Synthesis mismatch ?
- ➢ FSM reachability ? FSM transitions ? Dead-Code ?
- ➢ ….

Logic Designer & Verification Engineers

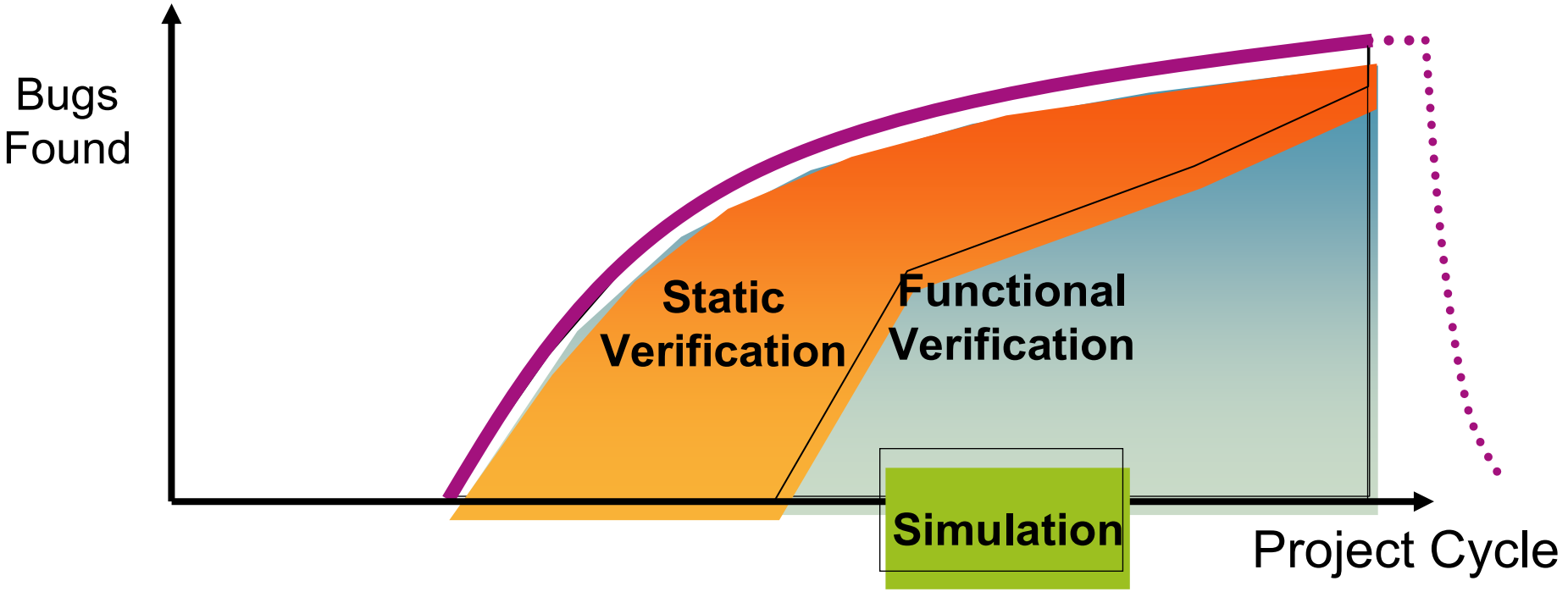
Time
RTL

Traditional Verification



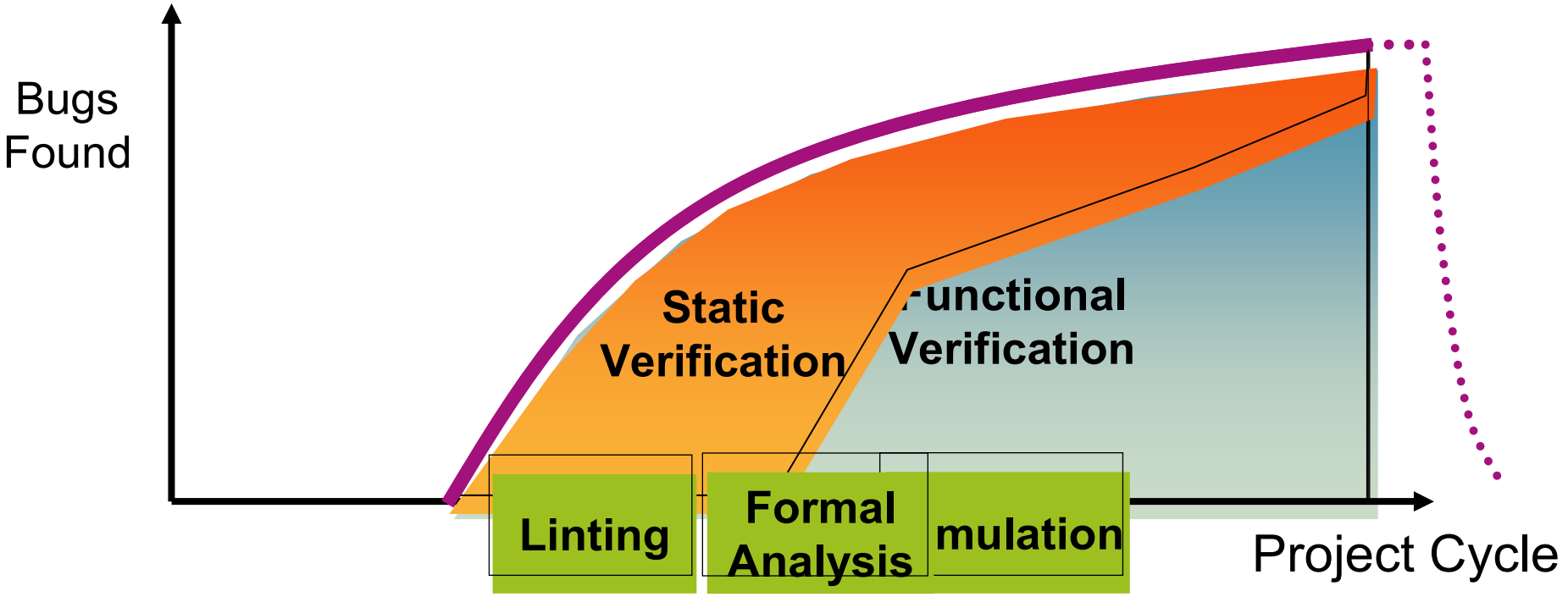
- Traditional functional verification is focused on simulation
  - Starts late, after significant manual testbench development

Introducing Static Verification flow



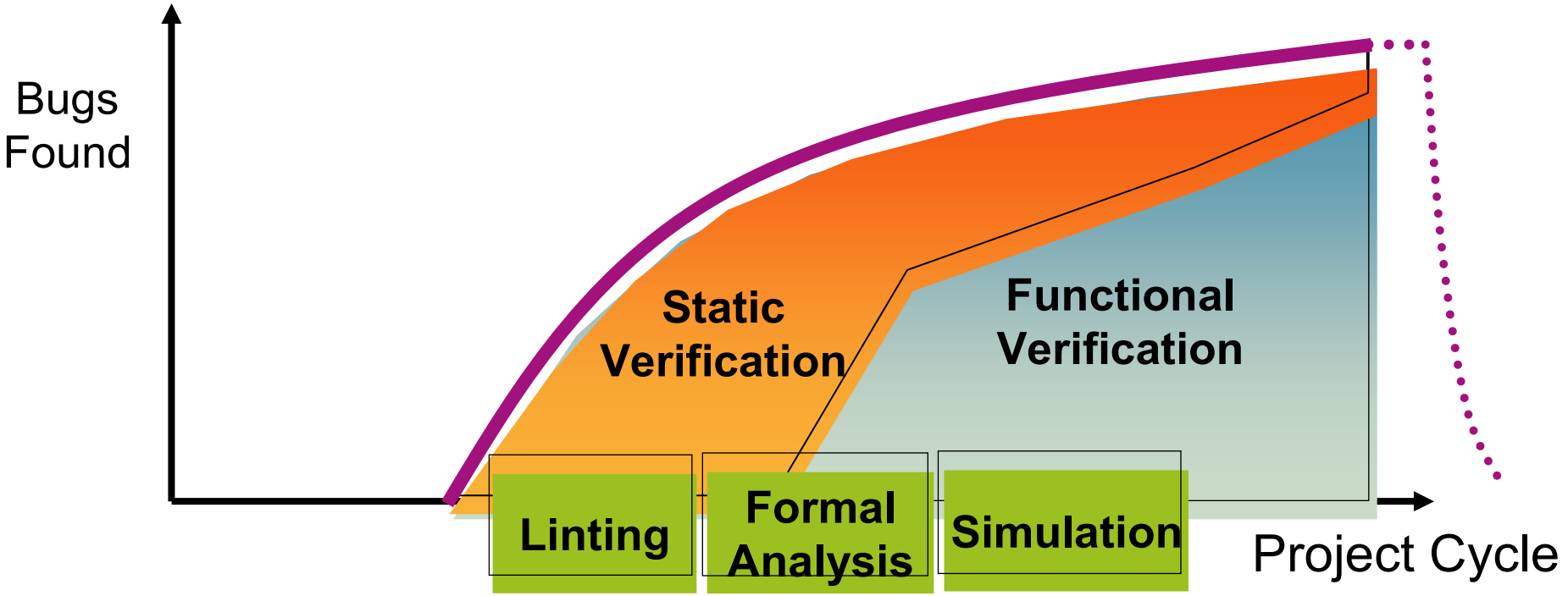
cadence™

Static Verification Technologies



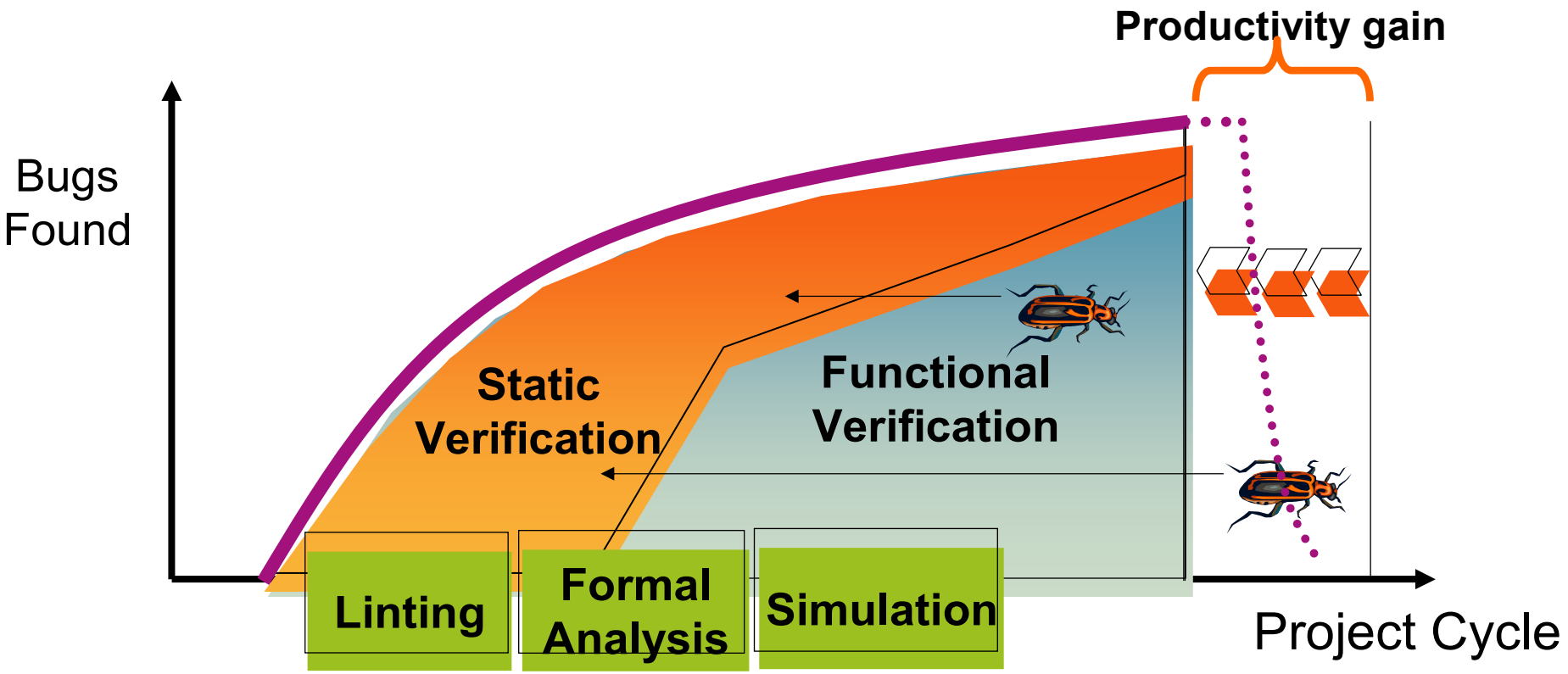
cadence™

Introducing Static Verification technologies



cadence™

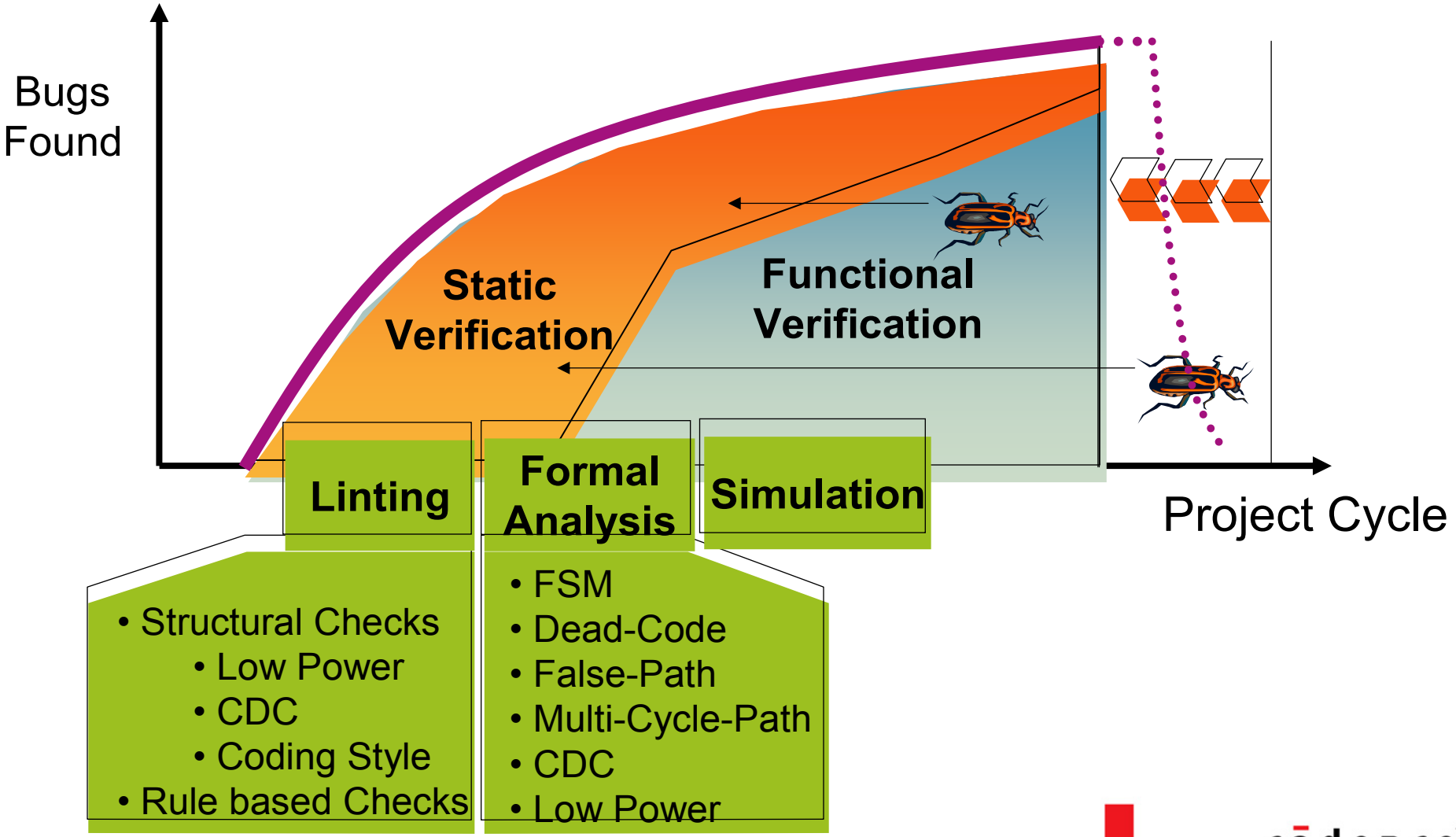
Introducing Static Verification



- Static verification using linting and automated formal analysis technologies can be applied earlier in the development cycle to help find bugs typically found late in the project cycle

cadence™

Static Verification Checking



cadence™

Challenges you face running static checks !!

► Do not overwhelm designer with huge number of rules (***Example: > 1000 rules, > 1000 warnings and errors***).

► Lack of knowledge about rule set and the kind of checks that should be applied at the different project cycles (***No design-friendly categories***).

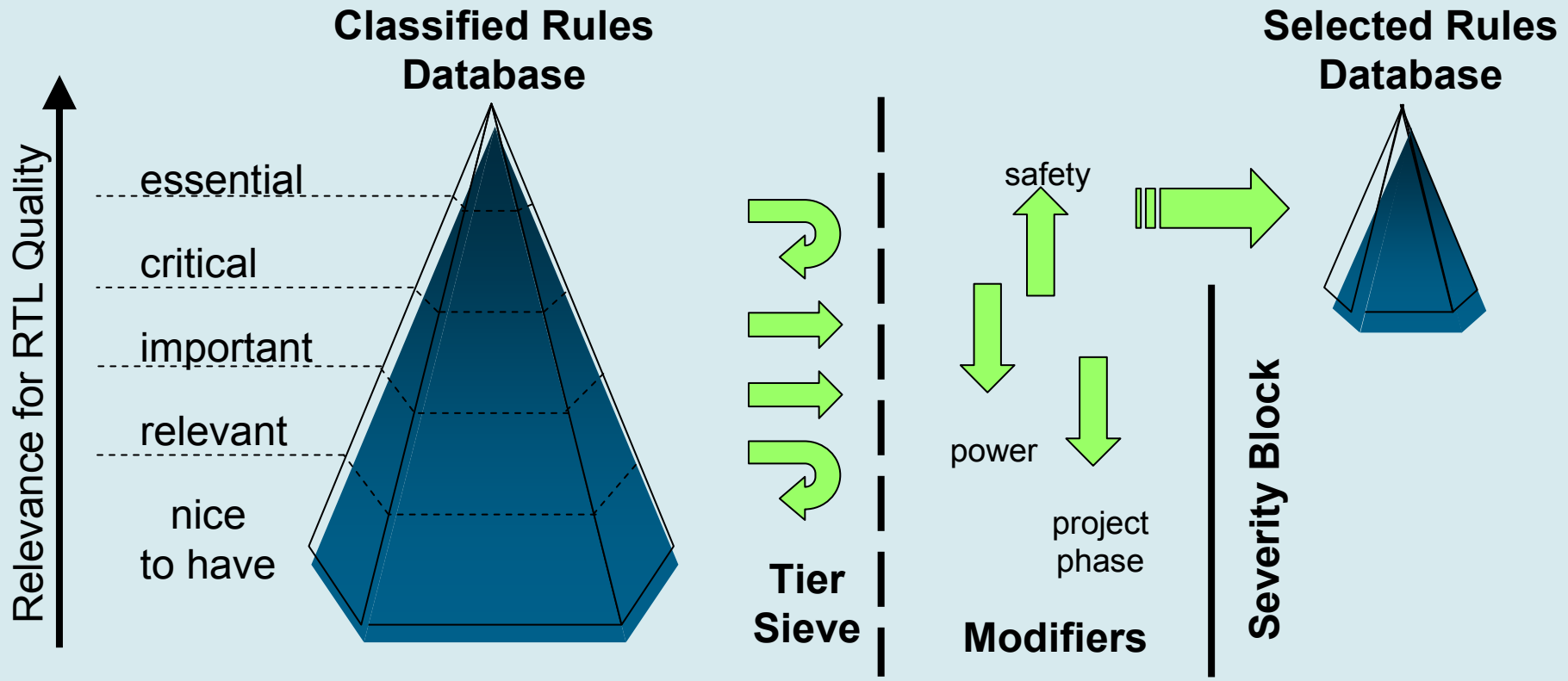
► Static verification is not established as a part of the design flow (***Time allocation not part of the project schedule, underestimate the debugging effort***).

► Reusable methodology is missing (***Rule sets are not applied to the sign-off criteria***).

freescale semiconductor ™

Our Vision (Freescale) -1-

- Effective rule set (Rule Architecture)
  - Categorization & Prioritization of rules applied in different static checks



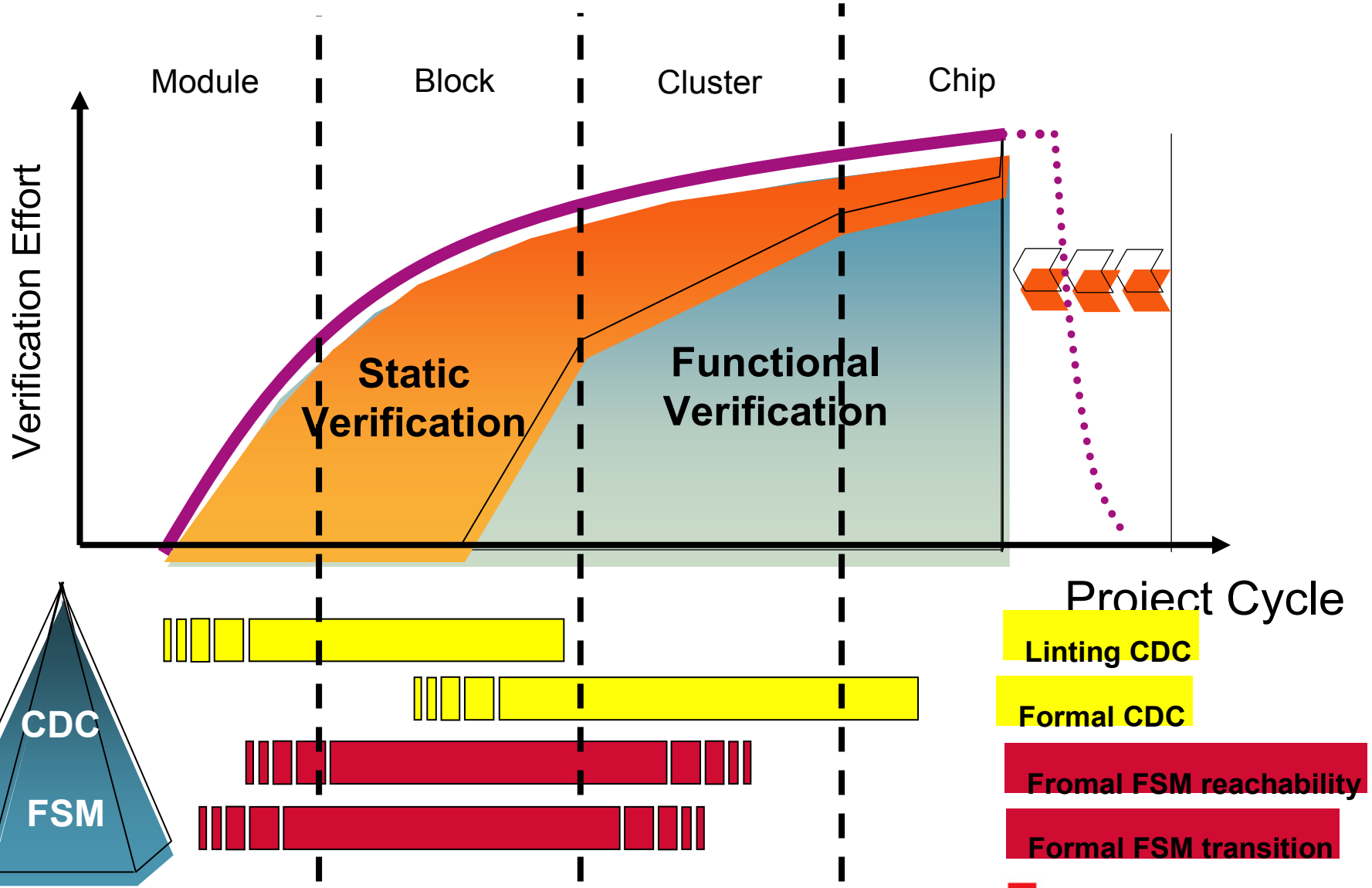
Slide is reused from a contribution of some of the authors to the forthcoming book:
Brian Bailey, Kathy Werner: "Intellectual Property for Electronic Systems: An essential introduction", IEC 2007

freescale semiconductor ™

Our Vision (Freescale) -2-

- Implement Structural Checks
  - Extend usage of Structural Checks on Block and SoC level by providing a cookbook and rules to identify vendor/costumer synchronizers.
- Implement Formal Checks
  - Usage of Formal Checks on Block and SoC level if complexity allows.
- Apply Timing Constraint Checks
  - Usage of Timing Constraints checker rules eliminates expensive and time consuming Gate-Level simulation efforts to verify that the intent and the implementation are consistent.
  - Avoid additional respins and to reduce time to Tape-Out (no Gate-Level simulation, ease ATPG path-delay pattern generation)

Practical Approach



cadence

Benefit of the Approach (Freescale)

- Cost Reduction
  - ***Significant reduction*** of the Gate-Level simulation effort by a usage of Timing Constraint Rule Checker earlier in the design flow.
  - IT infrastructure cost reduction by usage of predictive rule and timing checkers.
  - Reduce feedback re-work between Back-End and Front-End Teams.

- Methodology and Flow Improvement
  - Improving our Tape-Out flow by introduction of Static Verification Tools.
  - Reach completeness of static verification checks (e.g. Lint, Automatic Checks, CDC, Timing Constraints Checks).
  - Improve productivity.
  - High Quality IP delivery.
  - Reducing SoC malfunctions.

Conclusion

- Methodology based Rules Checkers Improve design quality
- Reducing cost and improving productivity.
  - Reducing Gate-Level simulation by static checks.
- Structural check of analog models
  - Regarding propagation of input properties
- Improve IP Reuse methodology
- Reduced risk of Re-Spin, ECO

Outlook

►Identify wrong Multi-Cycle/False Path formally in order to check it in early stage of the design flow.

► For complex or missing synchronizers:
- Identify and verify automatically.
- Interface to incorporate customized synchronizer structures.

► Identify asynchronous inputs used on internal Combinatorial-Logic during structural analysis.

► Extend the usage of Clock Domain Crossing checkers on SoC level.

Q&A