2026 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

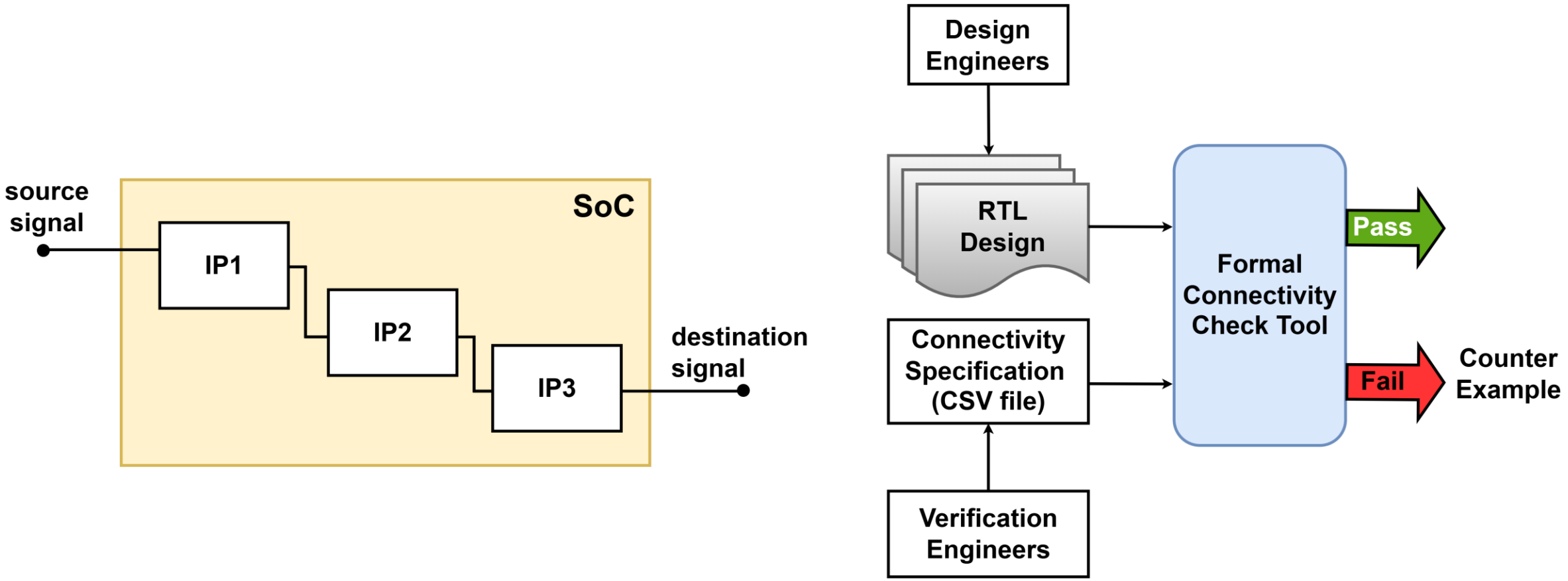
SANTA CLARA, CA, USA
MARCH 2 - 5, 2026

ConnChecker: Automated Root-Cause Analysis for Connectivity Check via Graph

Do Ngoc Tiep, Nguyen Linh Anh, Luu Danh Minh
Infineon Technologies

Table of content

Formal Connectivity Check In a Nutshell



Debugging: The most Painful Step

- Time-consuming
- Stressful
- A lot of recurring steps

Jasper Connectivity App

Pass

Fail

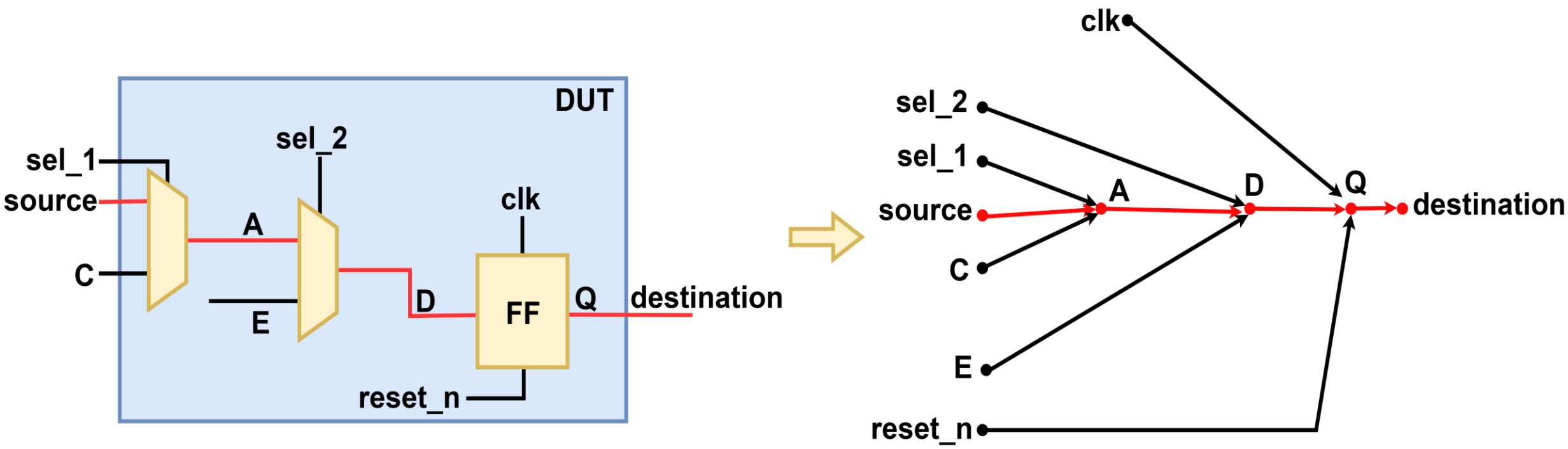
Counter Example

Manually Debug by Waveform, Schematic, Source code

- Full Path Analysis
- Why's the connection failed ?
- How to fix it ?

ConnChecker

Divide & Conquer via Dependency Graph

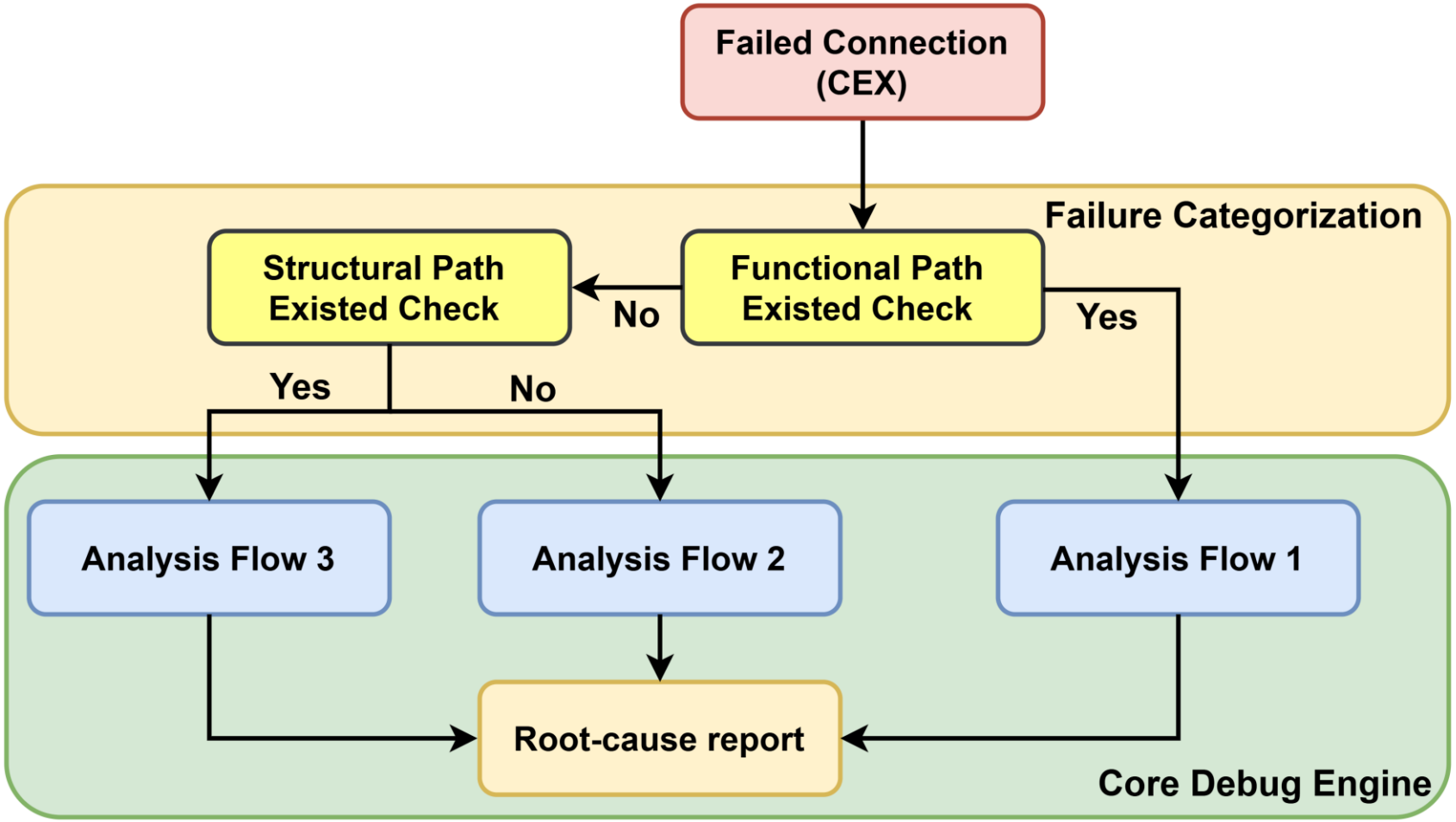


- Dependency graph shows how signals depend on each other
- Formal tool (Jasper) can help to generate dependency graph

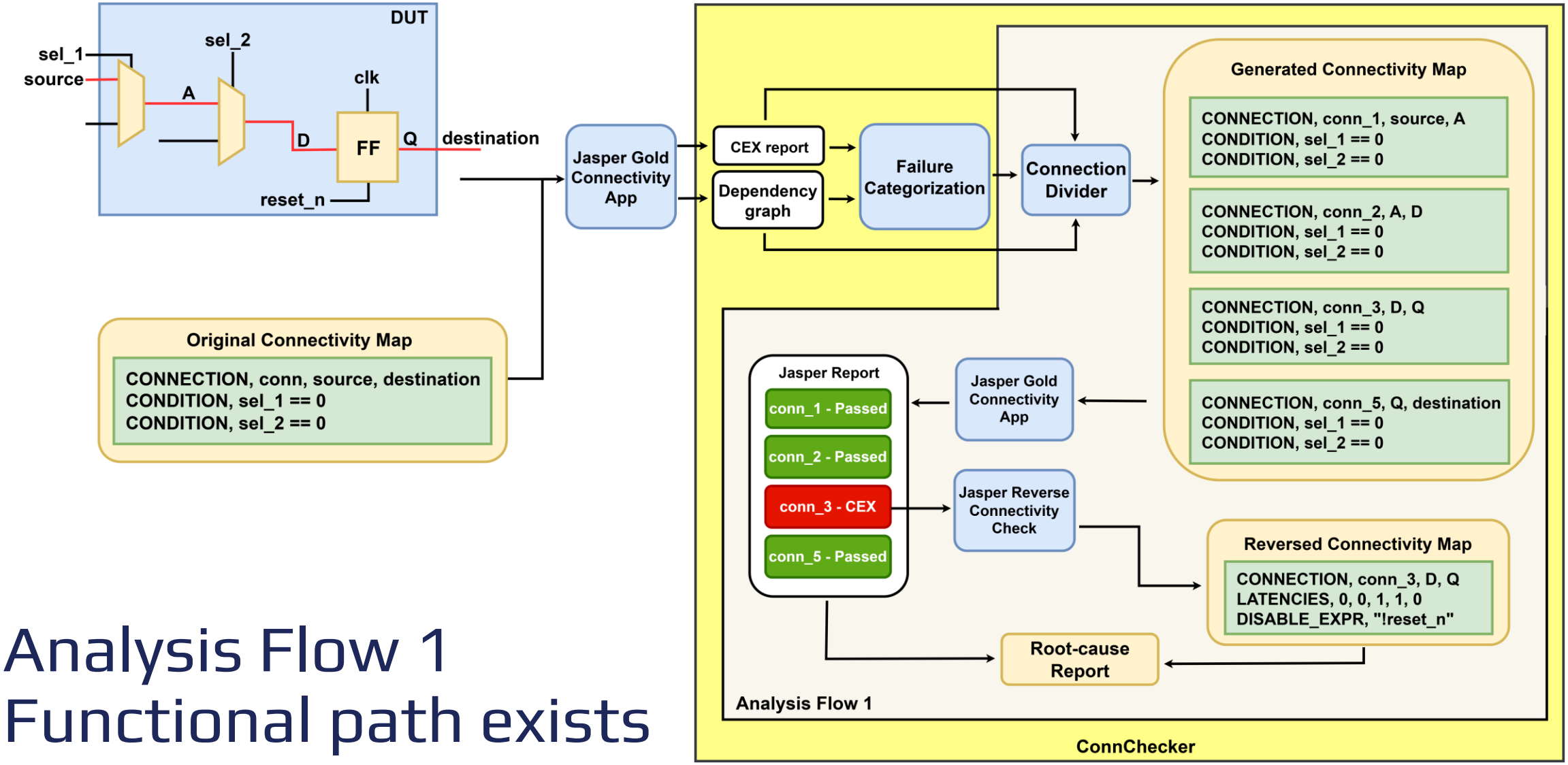
Functional Path & Structural Path

- Structural path:
  - An RTL-defined path between two signals
  - Visible in the cone of influence
  - Regardless of whether signal propagation is possible
- Functional path:
  - A subset of structural paths
  - The source signal can logically influence the destination signal
- Three types of bug:
  - Both structural & functional path exists (missing conditions, RTL bug,…)
  - Only structural path exists (over-constrained, RTL bug,…)
  - No structural path exists (not connected yet,…)

ConnChecker Flow



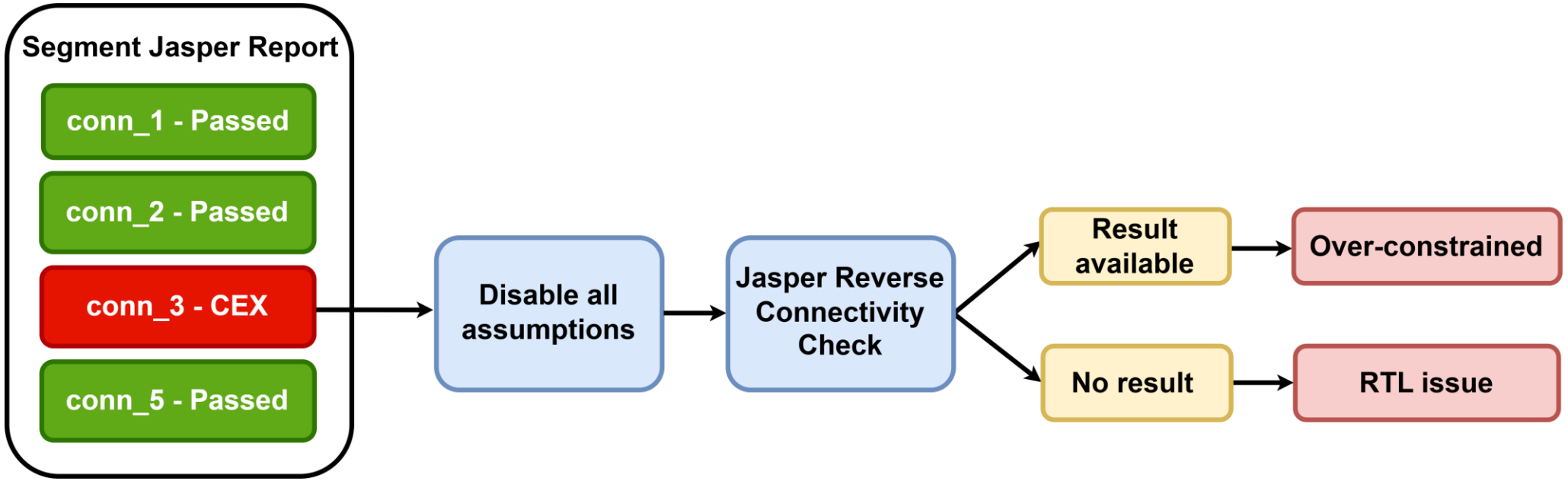
Fully automated by Jasper TCL and Python



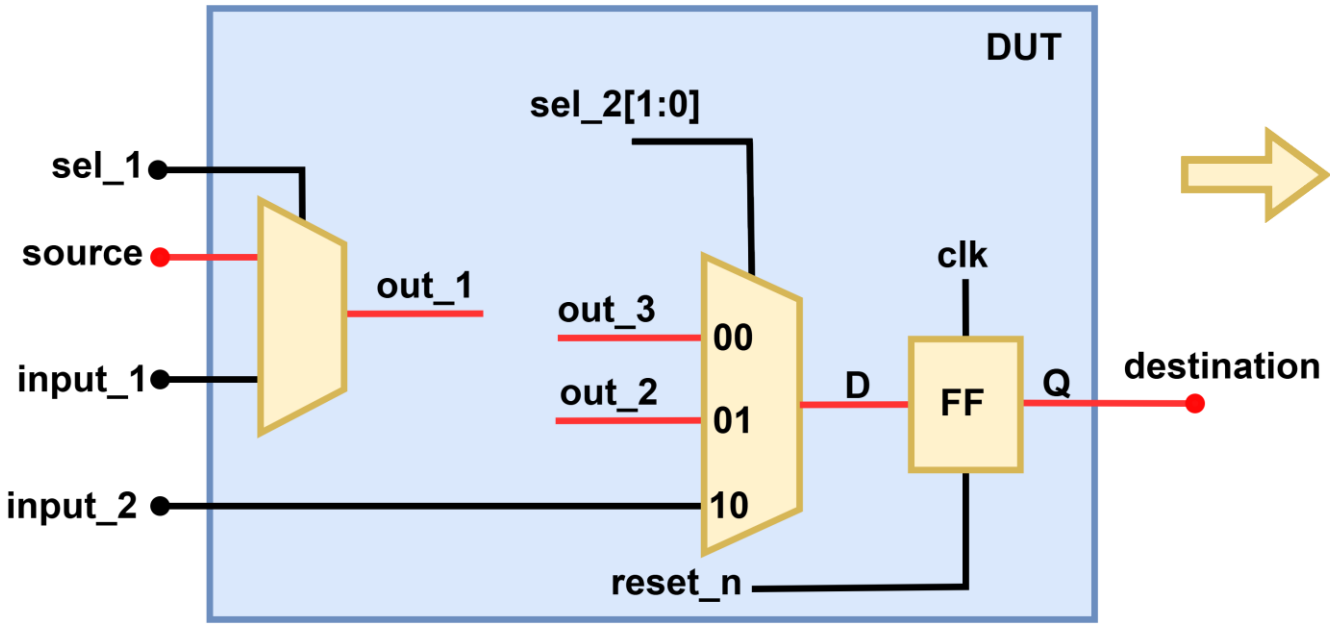
Analysis Flow 1
Functional path exists

Analysis Flow 3 – Only structural path exists

- Mostly similar to Analysis Flow 1
- Reverse connectivity check can not be applied on failed segments if they are not functional (over-constrained, tied signal,…)



Fan-in Analysis - Driver & Root Driver



Destination's drivers

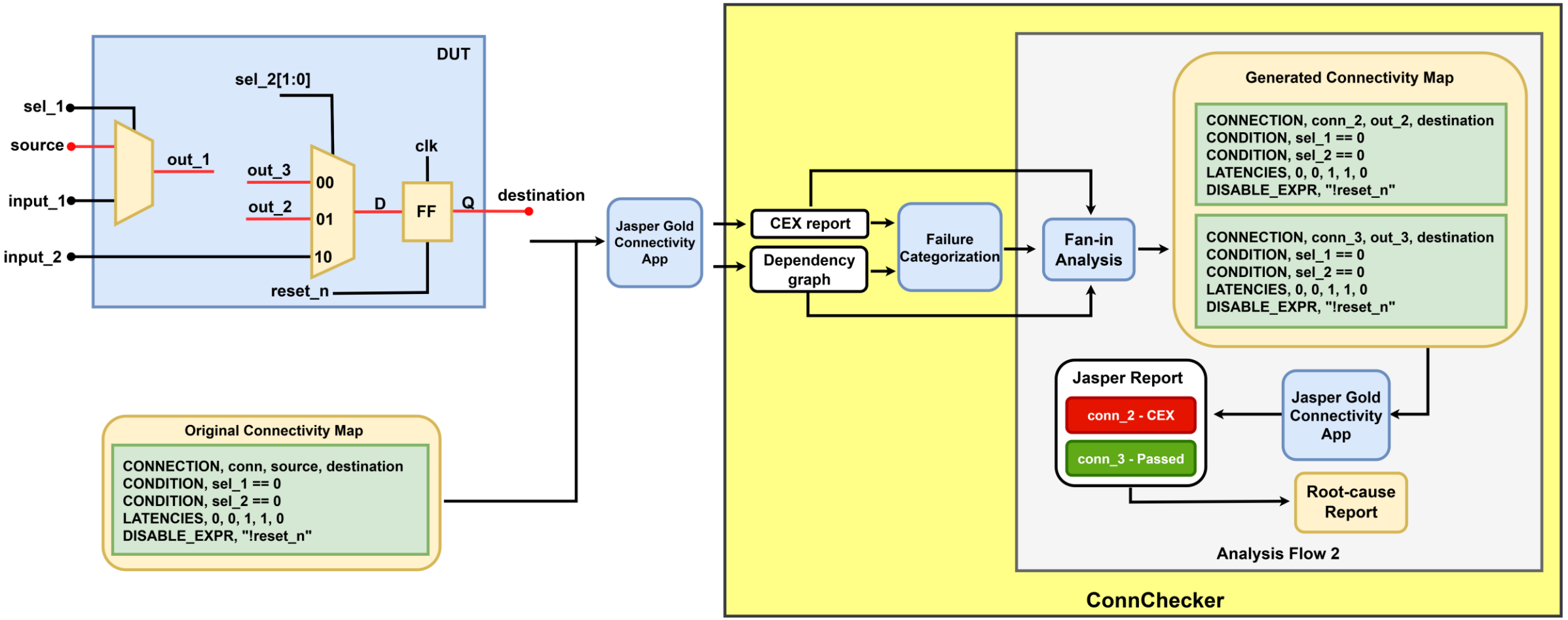
- Q
- D
- out_2
- out_3
- input_2

Destination's root drivers

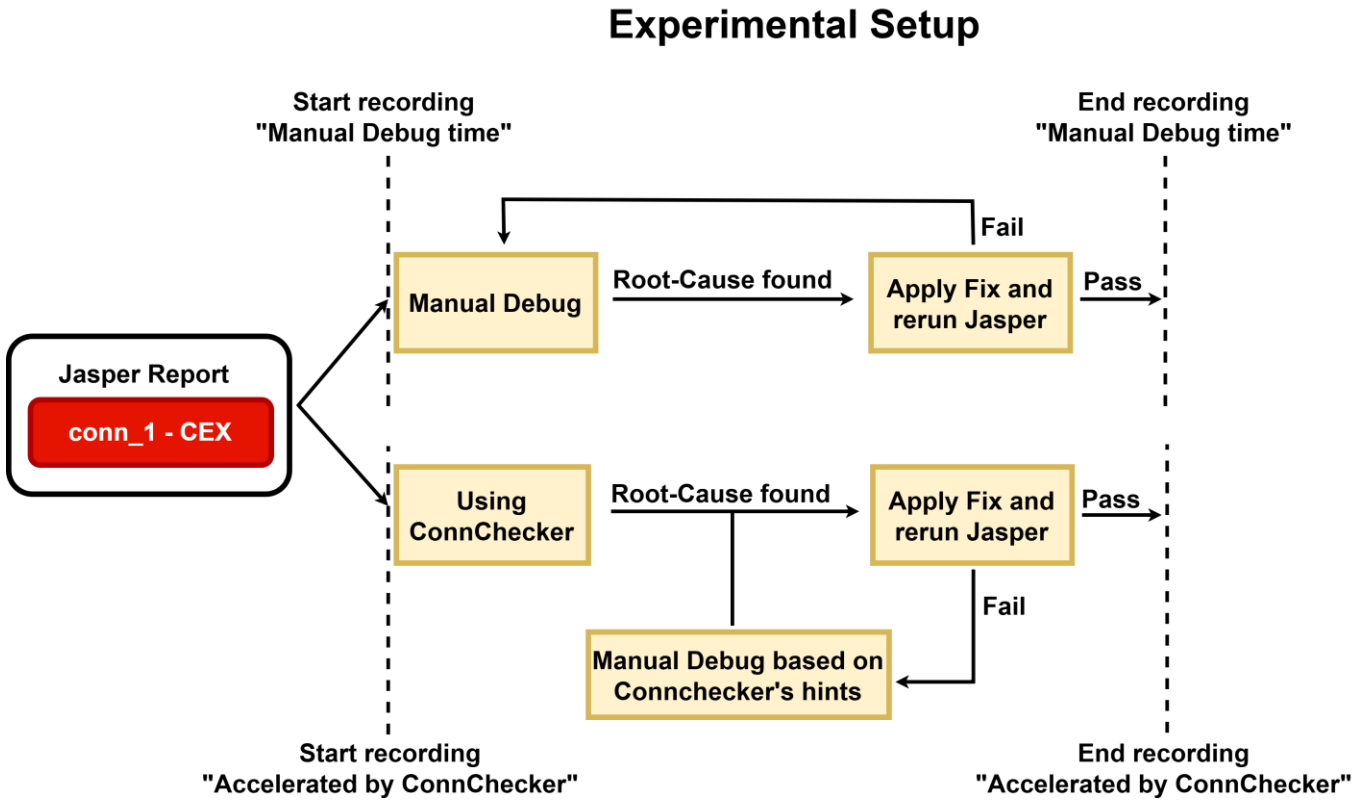
- out_3
- out_2

- Find all potential breaking points
- Root Driver: Drivers which not be driven

Analysis Flow 2 – No structural path exists

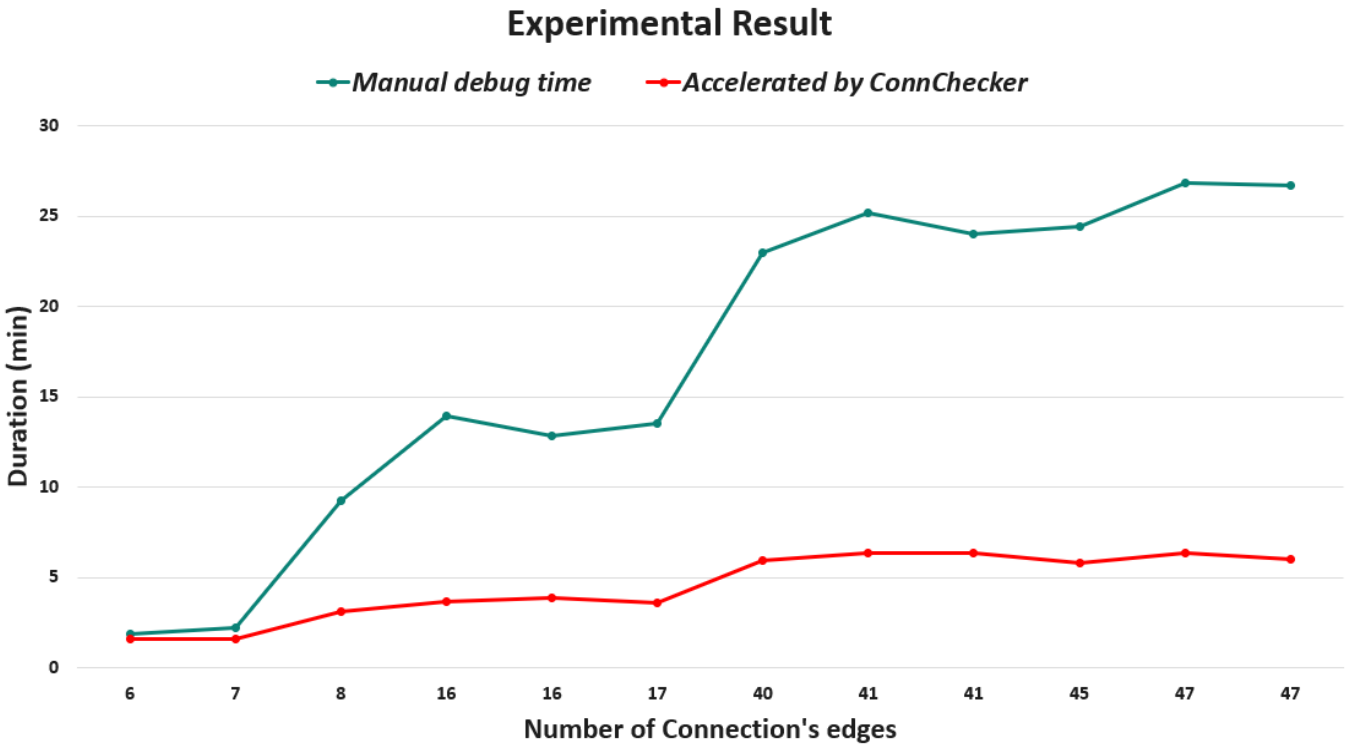


Experimental Setup



- Evaluated on 2 industrial SoC:
  - Microcontroller
  - Radar sensor
- Connection complexity metrics:
  - Number of connection's edge
  - Connection types:
    - Combinational
    - Sequential
    - Mixed-signal
    - Multi-clock domain

Experimental Result



- Efficiency: ~4x faster debugging time
- Scalability: red curve flattens as connection complexity grows
- Accuracy:
  - Strong on digital, multi-clock domain connections
  - Needs improvement for mixed signal connections

Summary

- ConnChecker
  - Scalable, graph-based framework for automated root-cause analysis
  - Converts tool (Jasper) outputs into structured debug workflows
  - Integrates easily through Python and TCL as an extension for formal tool (Jasper)
- Achievements
  - Observed an average 4x reduction in debugging time
- Future work
  - Leverage the graph-based foundation to enable innovation in formal debug

2026
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SANTA CLARA, CA, USA
MARCH 2 - 5, 2026

Q&A