Simulation Analog Fault Injection Flow for Mixed-Signal Designs

Pablo Cholbi Alenda, Analog Devices Inc., India

Dylan OConnor Desmond, Analog Devices Inc., Ireland

Raman K, Analog Devices Inc., India

accellera SYSTEMS INITIATIVE

2019 DESIGN AND VERIFICATION DVCON CONFERENCE AND EXHIBITION INDIA

What is Fault Injection?

- Verify how system reacts in presence of hardware failures
- According to ISO 26262 – Part 1 [1]:
  - **3.54 fault**: abnormal condition that can cause an element (3.41) or an item (3.84) to fail
  - **3.57 fault injection**: method to evaluate the effect of a fault (3.54) within an element (3.41) by inserting faults (3.54), errors (3.46), or failures (3.50) in order to observe the reaction by observation points (3.101)
- Goals: Verify…
  - Safety measure is implemented correctly
  - Safety measure detects required faults
  - Evidence of claimed diagnostic coverage
  - Enters safe state within fault tolerance time
  - Correlation on diagnostic coverage
  - No unforeseen dependent failures
  - No gaps in safety analysis

[1] International Standard Office (ISO), ISO 26262 Road Vehicles-Functional safety. Geneva, Switzerland: ISO, 2018.

What is Fault Injection?

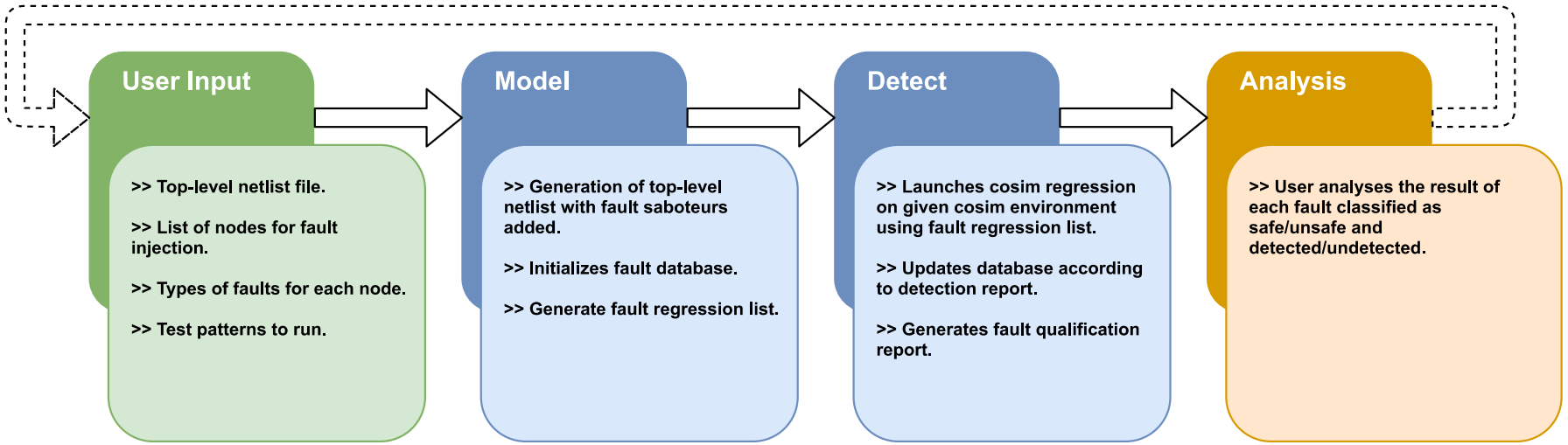
- Fault injection verification takes various forms:
  - According to project stage:
    - Pre-silicon (simulation)
    - Post-silicon (evaluation)
  - According to type of fault/block:
    - Analog
    - Digital
    - Firmware/software
- Functional safety verification should be approached holistically
- This work is focuses on **analog pre-silicon** analog fault injection **simulations** flow

Existing Analog Fault injection Simulation Solutions

- Digital Fault Injection Simulations (DFI)
  - Covered in more detail in standards
  - More mature EDA tools and flow
- Analog Fault Injection Simulation (AFI)
  - Not so well standardized
    - IEEE P2427 is trying to standardize
  - Cadence® Legato™
    - Requires Spectre® license
    - Earlier based on Verilog AMS
    - Now based at transistor-level
  - Mentor® DefectSim™
    - Faults injected at transistor-level
  - Synopsis®
    - No solution available at time of writing

- A need for an alternate was identified
  - Verilog AMS fault injection is too high level
  - Transistor-level fault injection is too low-level
  - Use Verilog AMS is not widespread at ADI
  - If possible, reduce number of licenses
- This work elaborated on the implemented/proposed solution

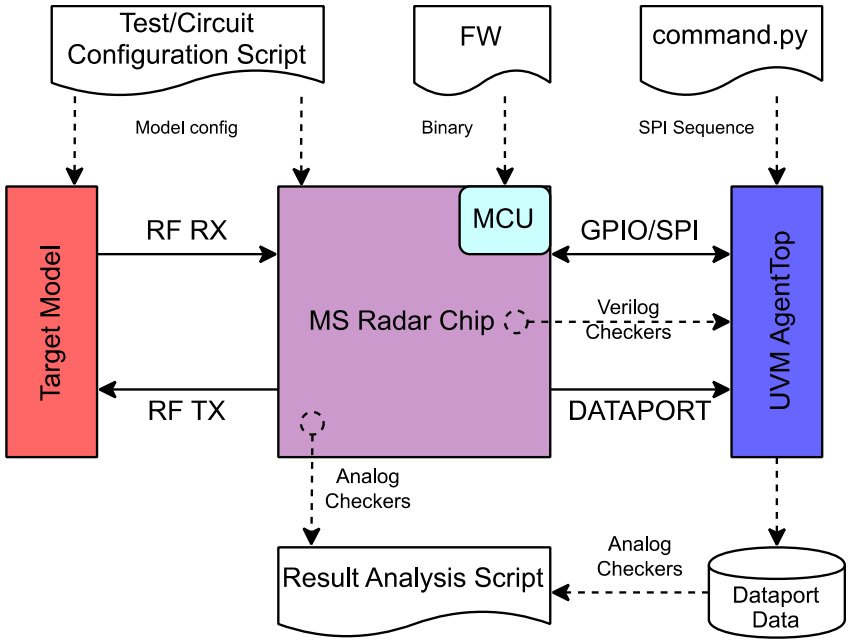
Analog Fault Injection Simulation Flow



User Inputs:

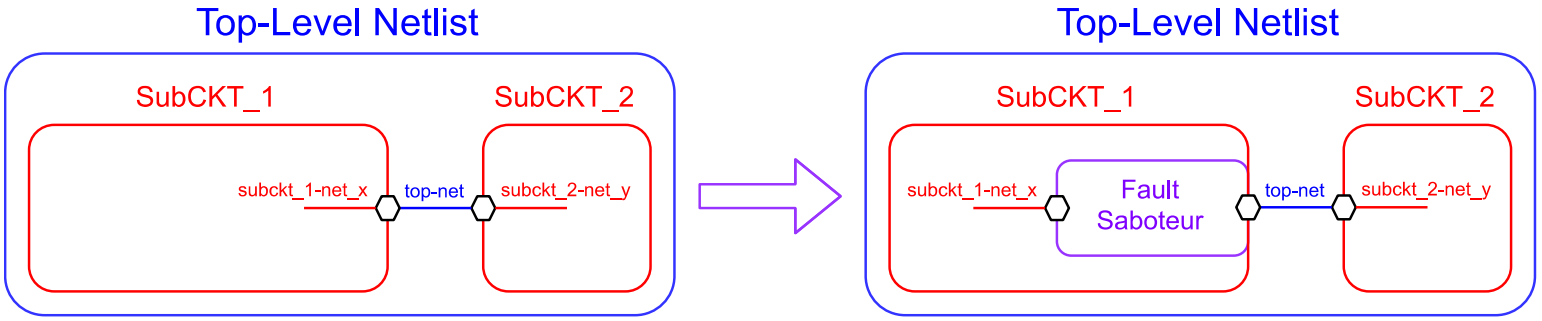
- Build on top-level functional co-simulation verification environment for AFI simulations

- Only additional inputs needed are:
  - list of fault nodes
  - fault types/models
- Assuming that:
  - A test pattern for the AFI campaign is ready
  - Fault detection and safety goal violation monitors in place

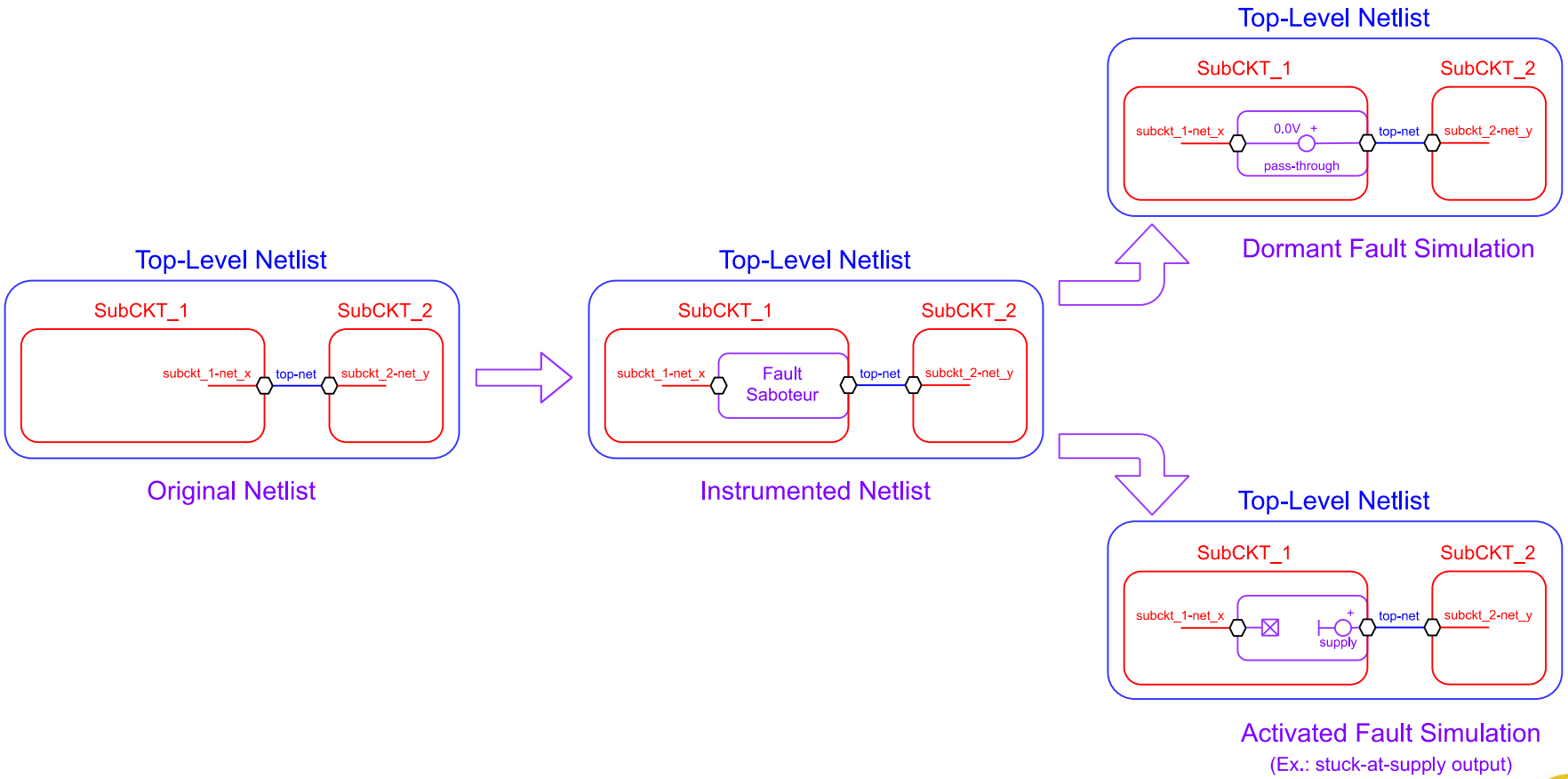


Model Phase: Inserting Fault Saboteur

- Faults can be inserted on any sub-circuit port
- 'Fault saboteur' sub-circuit inserted on fault nodes
- Fault saboteur is placeholder for fault model
- Fault saboteur sub-circuit has ports A and B
  - 'A' connected to port; 'B' connected to internal circuitry
  - CAD provides maintains core set of fault models
  - Projects/teams can create own custom fault models



Model Phase: Netlist Instrumentation

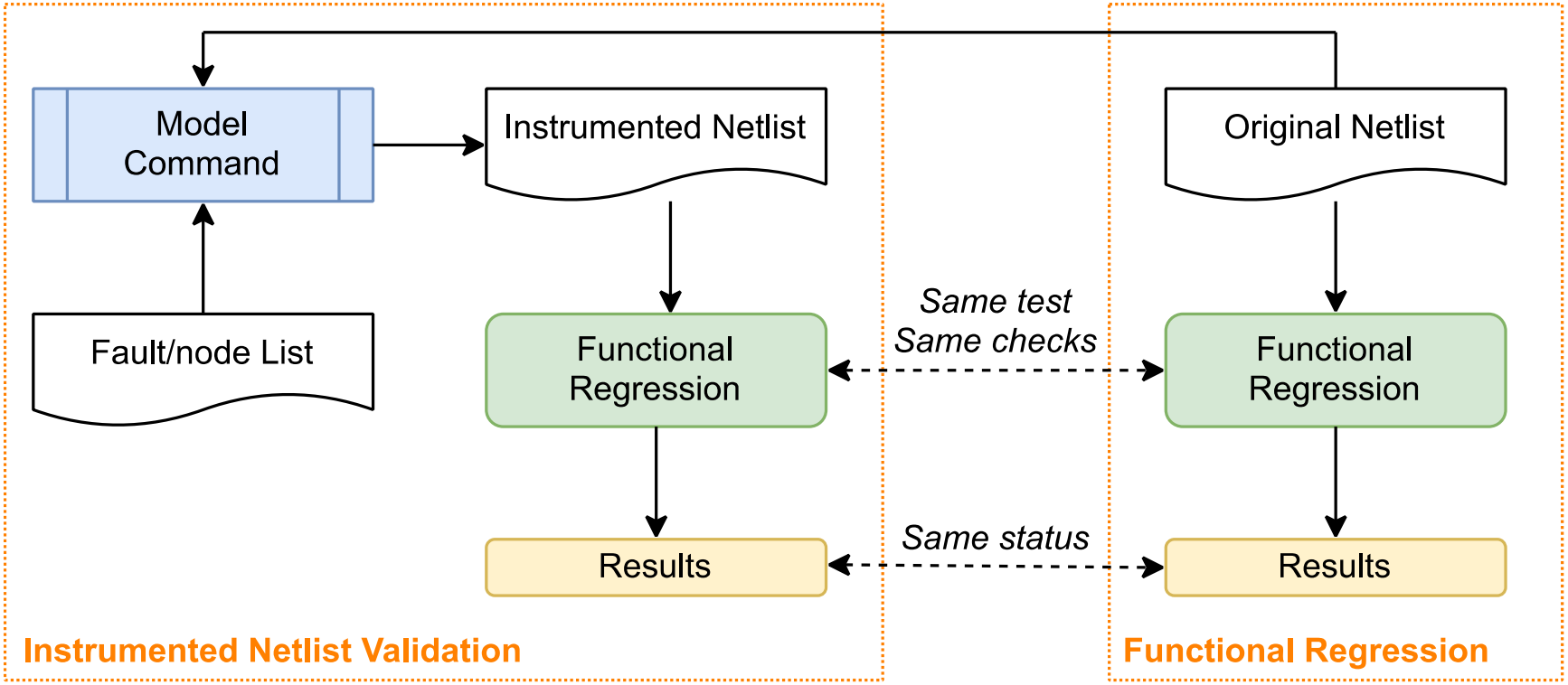


Model Phase:
Instrumented Netlist Equivalence

- How to claim functional equivalence of the instrumented netlist?

1. Generate an instrumented netlist from the node/fault list
2. Run top-level functional co-simulation regression on original netlist and instrumented netlist
   - Fault saboteurs dormant by default
   - Assuming functional verification test plan is comprehensive
   - Equivalent results indicates equivalent functionality
3. Compare results

Model Phase:
Instrumented Netlist Equivalence

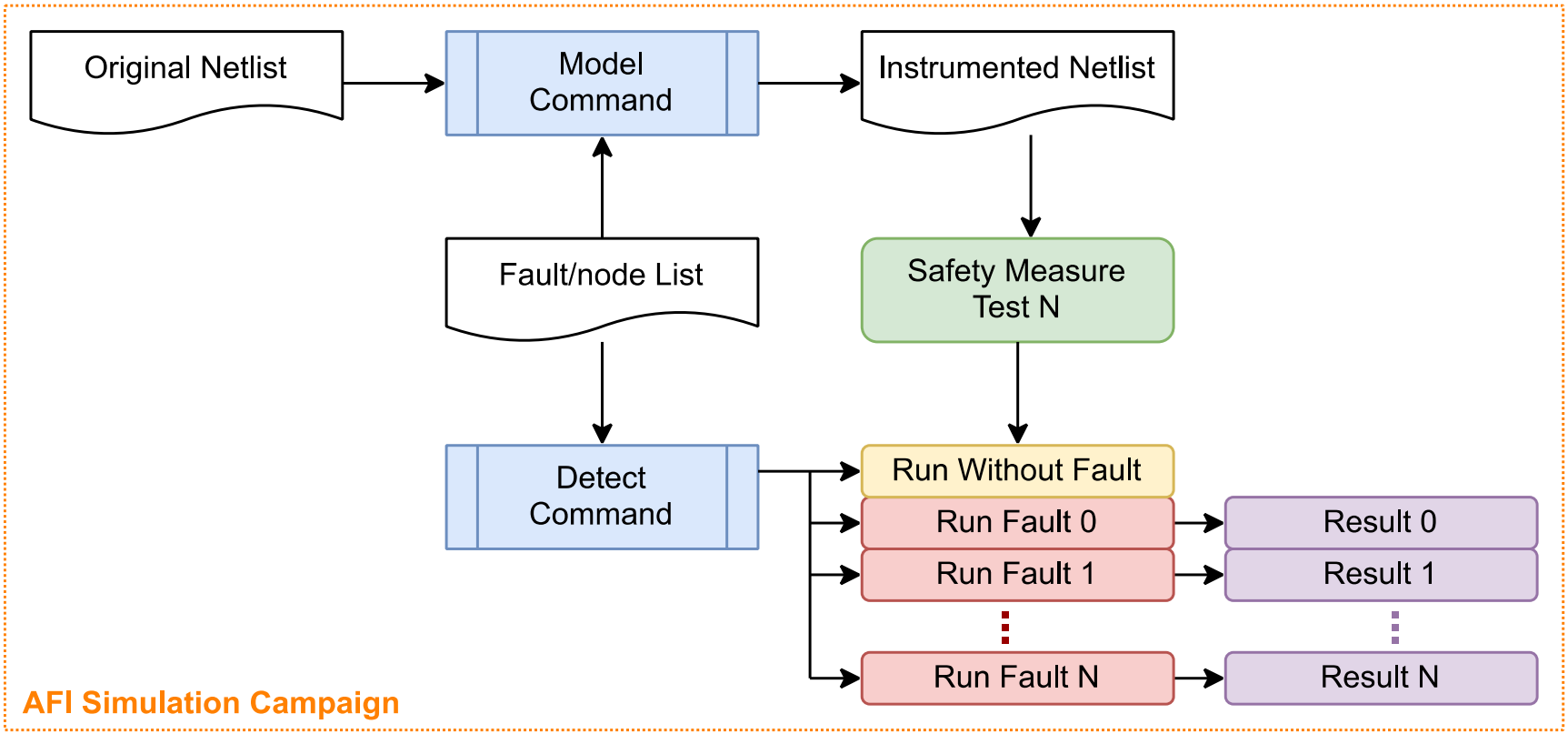


Detect Phase:
AFI Simulation Regression

- Fault injection regression
  - 1 ‘good run’ without fault injected
  - N fault runs for N fault/node combinations
- Testbench checker
  - Print message if fault is detected
  - Print message if fault violated the safety requirement
- Output/result
  - Classification of faults

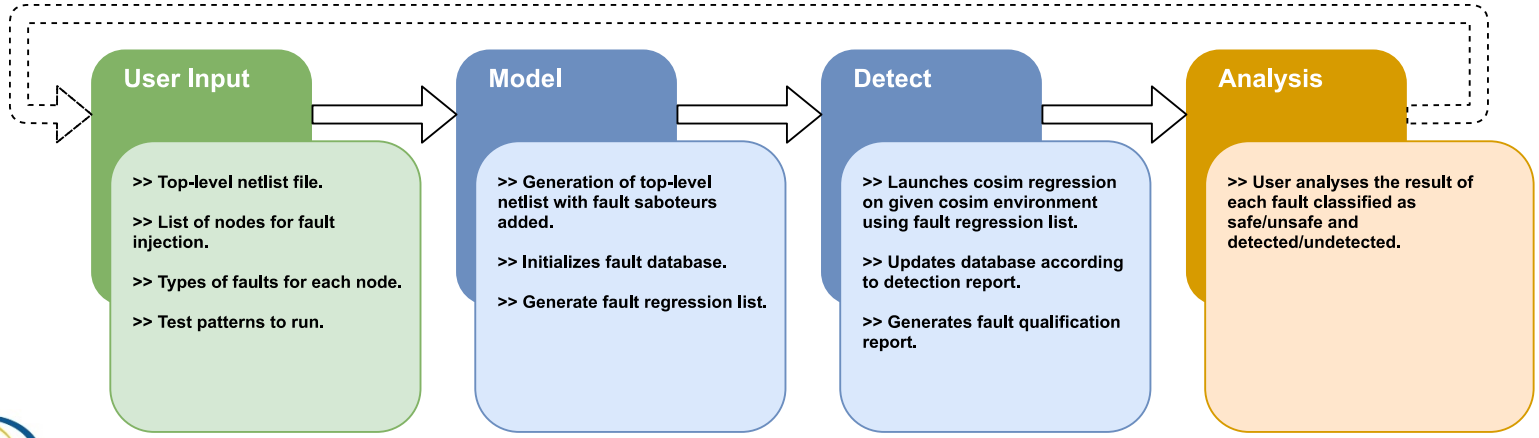
	Detected	Undetected
Safe	Safe Detected	Safe Undetected
Unsafe	Unsafe Detected	Unsafe Undetected

Detect Phase:
AFI Simulation Regression



Analysis Phase

- **Review** analog fault injection simulation regression results
  - Does the fault distribution/classification match expectations?
- If results **do not match** expectations:
  - Modify one or more inputs and re-iterate
- Remember: flow is iterative



Analysis Phase

- What inputs might need to be modified?

1. **Safety analysis**
   - Modify the assumptions of the safety analysis
2. **Design**
   - Modify hardware design, configuration, and/or firmware
3. **Environment**
   - Modify the simulation environment to overcome possible limitations
4. **Test & criteria**
   - Modify tests/checkers which are not consistent with the requirements

Future Work

1. Integration with ANSYS® medini™ analyze in investigation
2. Spectre™ in the FI tool planned for evaluation
3. Verilog AMS based models for fault planned for evaluation
4. GUI under development

- ADI to remain aligned with evolving AFI flows and standards
- Industry developments continuously monitored
  - Regular updates from AFI tool vendors

Questions