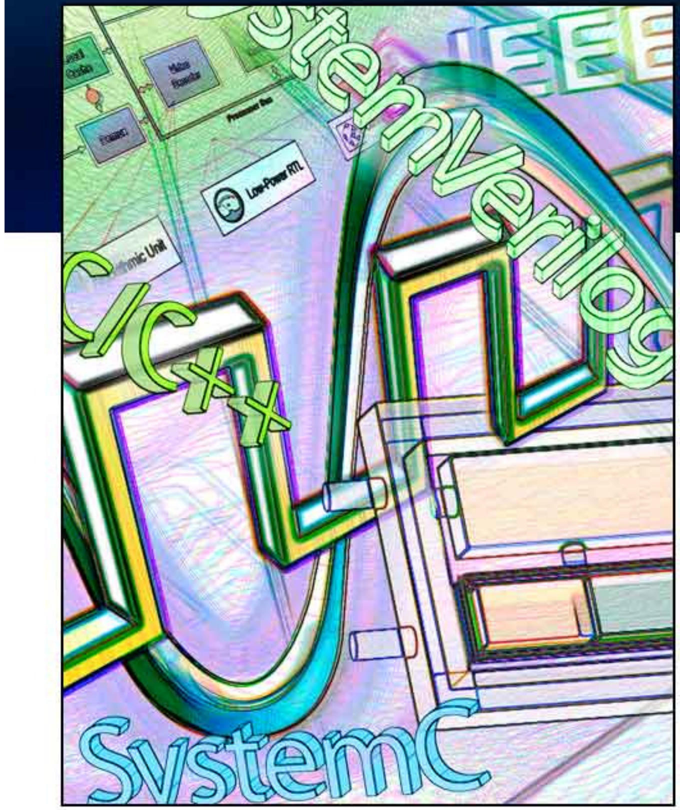


Formal Fault Injection for Functional Safety

Mark Handover

European Application Engineer
Digital Design & Verification Solutions

November 2016

Mentor Graphics®

Fault Injection – an ISO26262 Recommended Verification Method

- *Functional Safety*
  - Absence of unreasonable risk due to *hazards* caused by malfunction of Electrical/Electronic systems

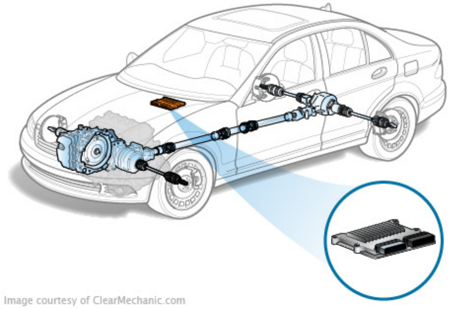


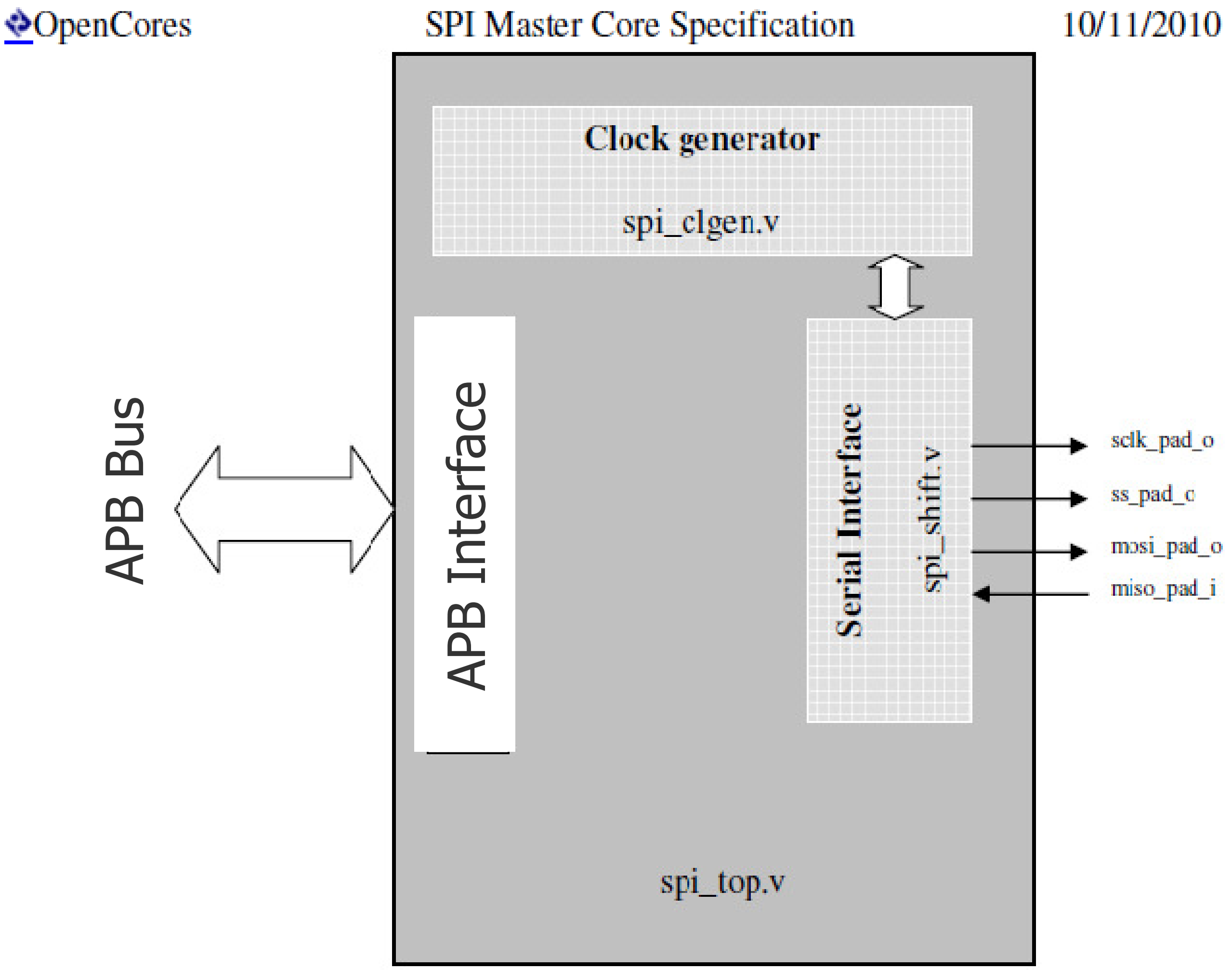
Image courtesy of ClearMechanic.com

- Fault tolerance the objective of the ISO26262 standard
  - Recovery or fail-safe – Safety Mechanism
- ISO26262 standard provides specific regulations and recommendations for automotive systems.
  - Fault Injection: is a method for hardware and software integration testing

Random Faults and Safety Mechanism (SM)

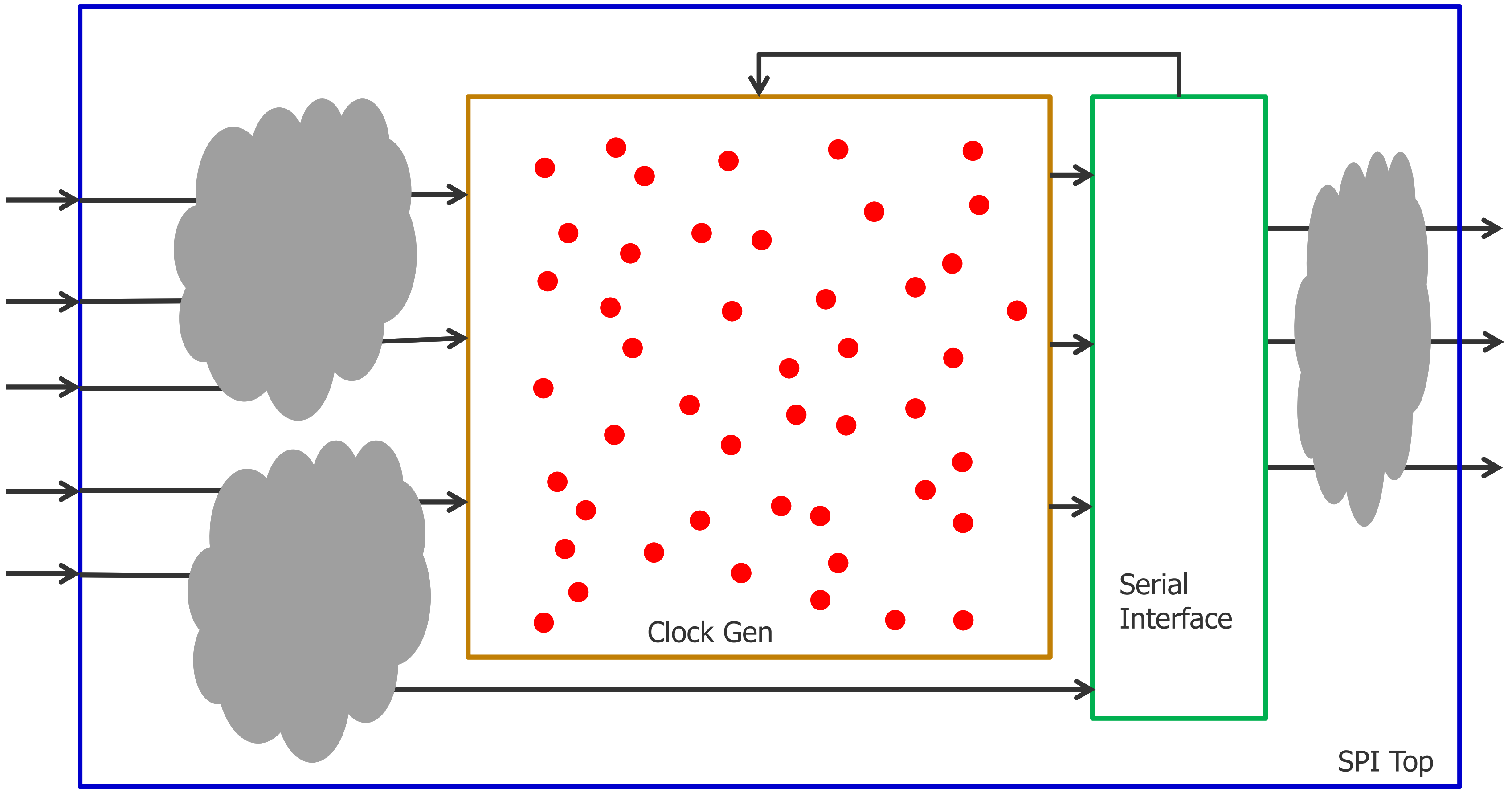
- Random Faults
  - Physical defects that can occur in system components during system operation
- Purpose of Safety Mechanism
  - Control random faults
    - Detect all faults
    - Provide a deterministic and correct reaction to faults
  - Guarantee safety operation of the system
    - Recover the system, or
    - Go to a safe system state
- Validation/Verification of Safety Mechanism
  - Completeness
    - Check the ability to detect and handle all possible faults
  - Correctness
    - Check that the safety mechanism specification/requirements are satisfied
    - For example:
      - Check design behaves as without presence of faults
      - Check design goes to a safe state

Safety Mechanism:
Illustration Using SPI Master Core Example

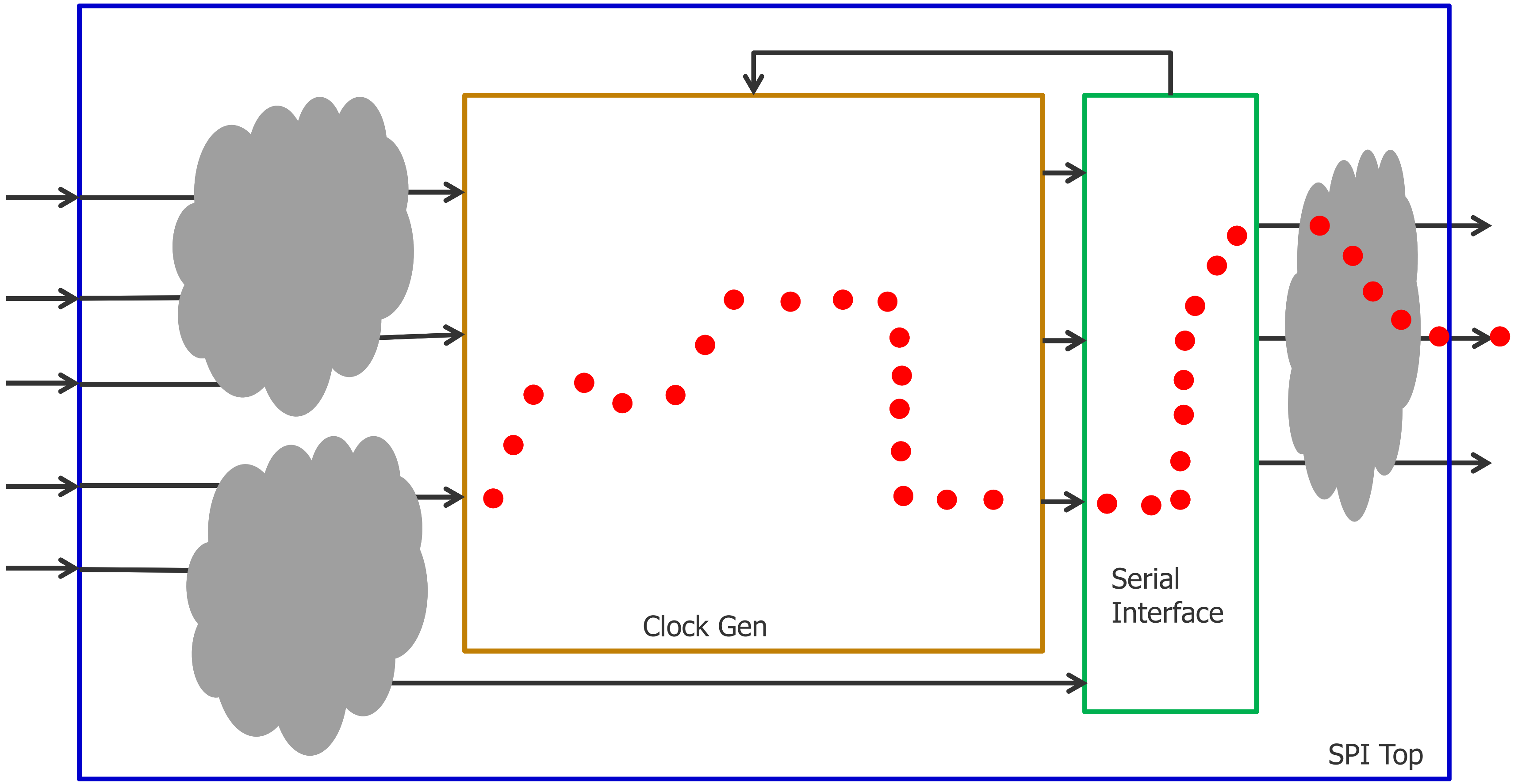


OpenCores SPI Master Core Specification 10/11/2010

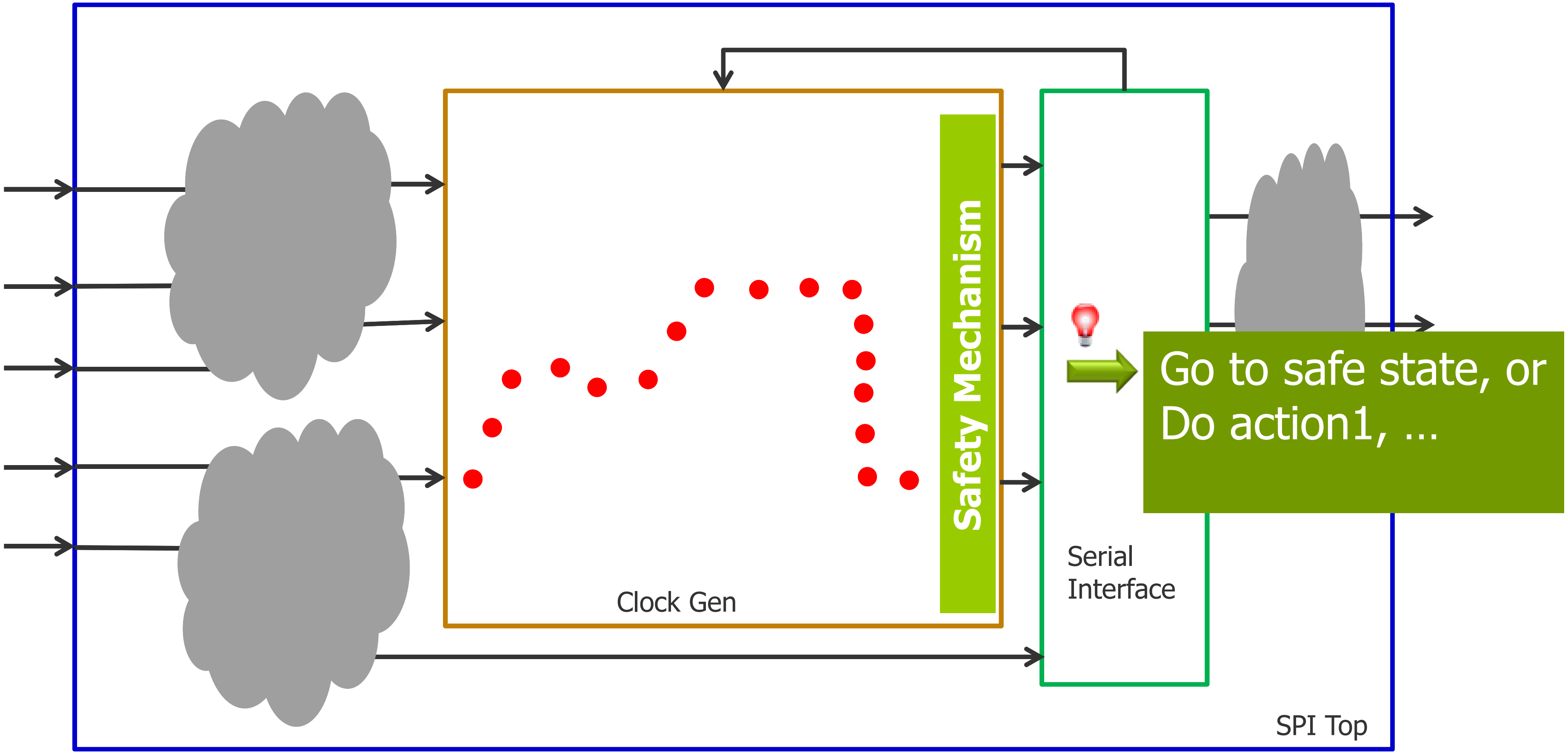
SPI Master Core:
Assume Faults Occurs in Clock Generator



SPI Master Core:
Faults Could Affect Functional Safety

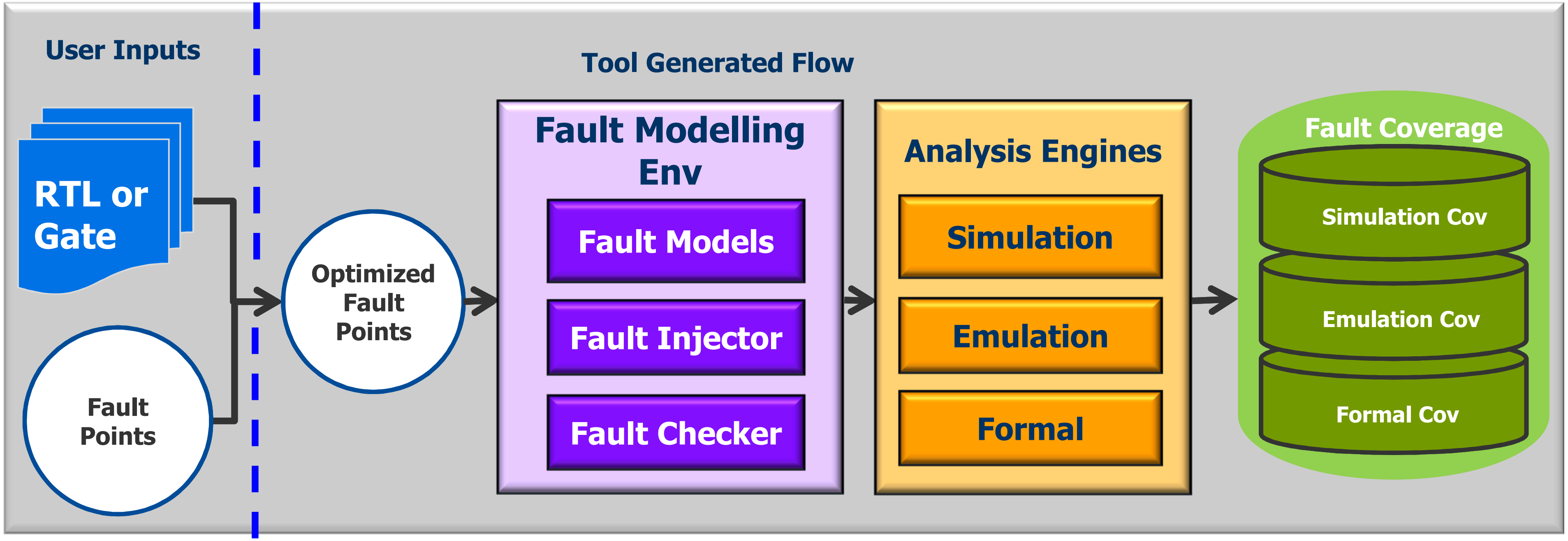


SPI Master Core: Fail-Operational Safety Mechanism Handles Faults



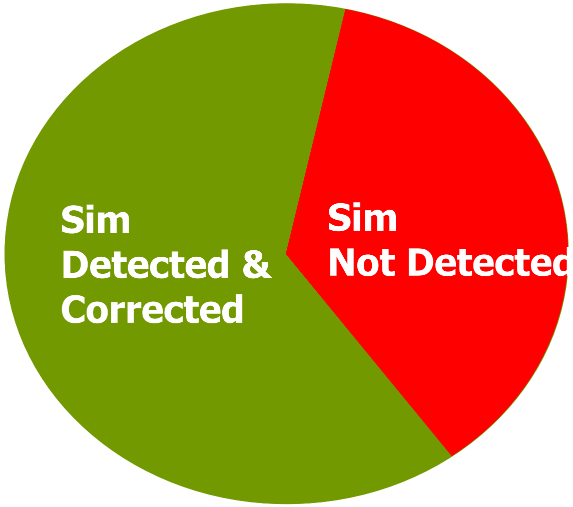
Go to safe state, or Do action1, …

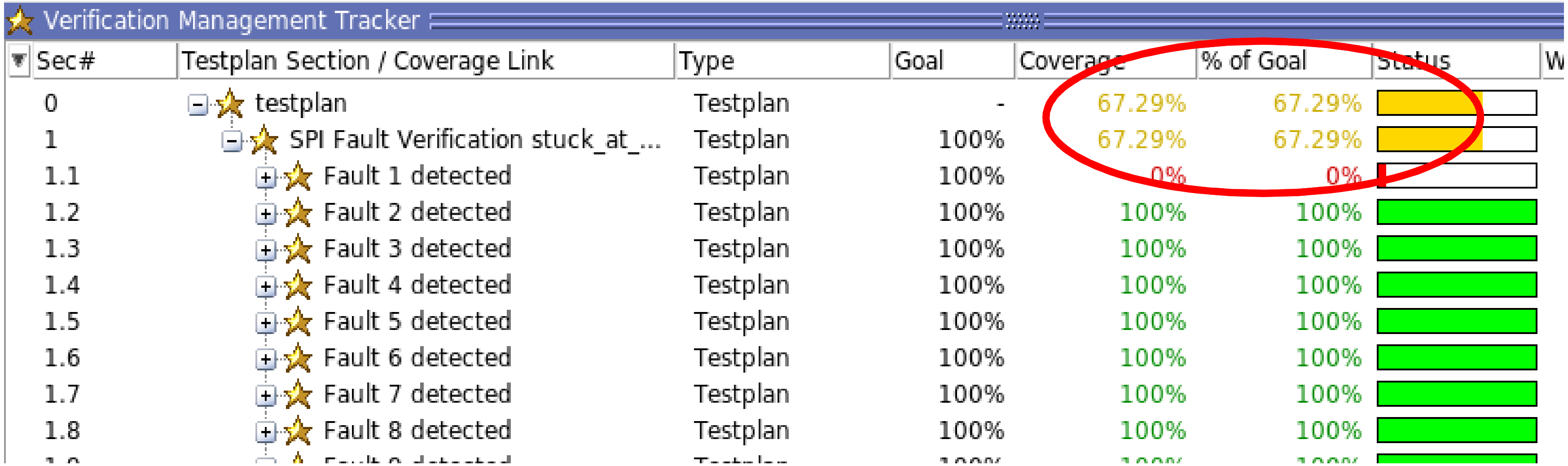
General Functional Safety Validation Flow



Fault Simulation Regression Results

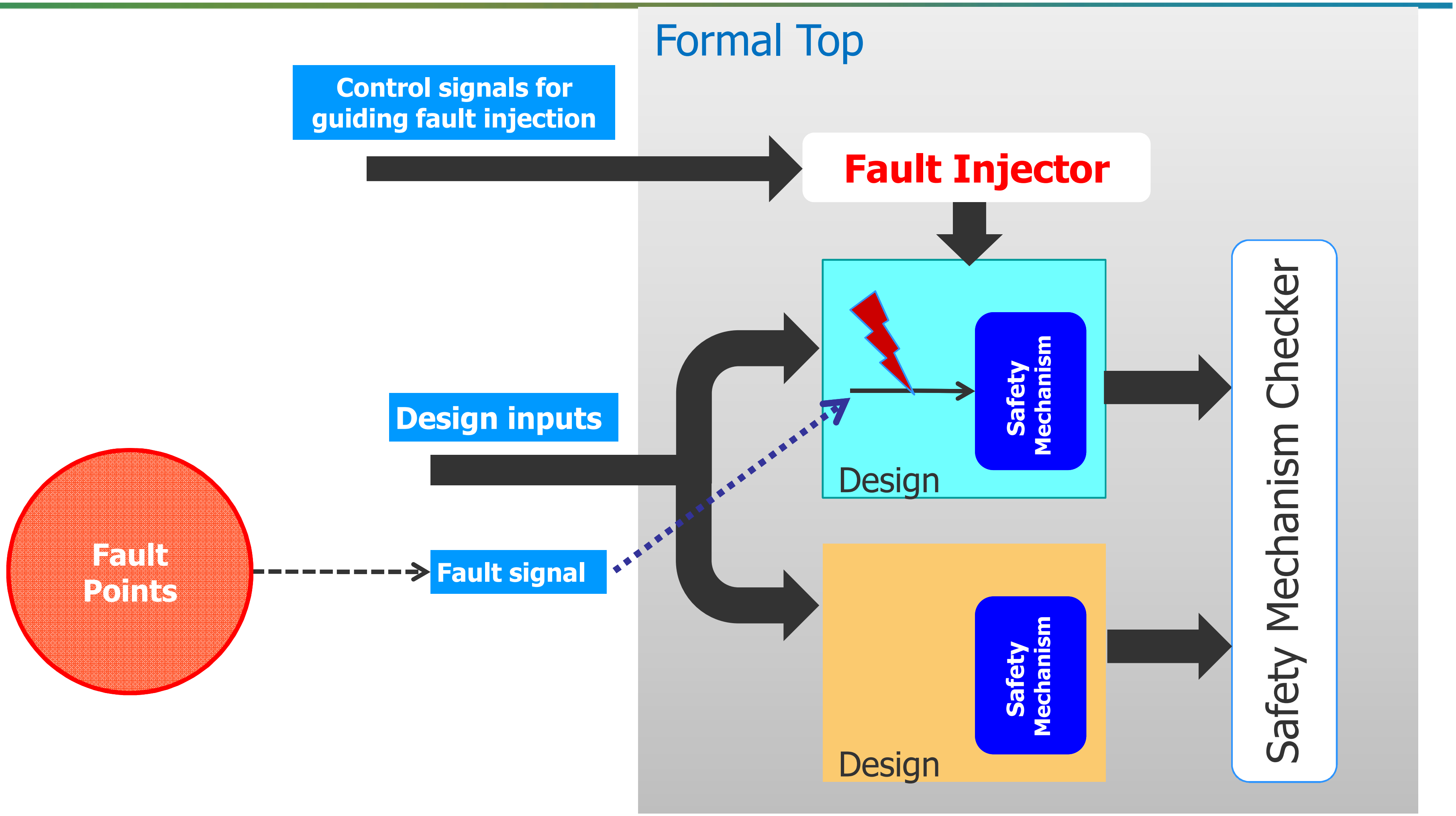
- Regression
  - Number of Tests **372**
    - Fault Models: Stuck-at-1
    - 372 Faults (Cell Outputs)
  - Results
    - Non-Propagatable **67%**
  - Run Time **2H 16min**





Sec#	Testplan Section / Coverage Link	Type	Goal	Coverage	% of Goal	Status	W
0	testplan	Testplan	-	67.29%	67.29%	🟨	
1	SPI Fault Verification stuck_at_...	Testplan	100%	67.29%	67.29%	🟨	
1.1	Fault 1 detected	Testplan	100%	0%	0%	🟥	
1.2	Fault 2 detected	Testplan	100%	100%	100%	🟩	
1.3	Fault 3 detected	Testplan	100%	100%	100%	🟩	
1.4	Fault 4 detected	Testplan	100%	100%	100%	🟩	
1.5	Fault 5 detected	Testplan	100%	100%	100%	🟩	
1.6	Fault 6 detected	Testplan	100%	100%	100%	🟩	
1.7	Fault 7 detected	Testplan	100%	100%	100%	🟩	
1.8	Fault 8 detected	Testplan	100%	100%	100%	🟩	
1.9	Fault 9 detected	Testplan	100%	100%	100%	🟩	

Questa Formal Model for Fault Injection



Fault Points

All design elements whose faults effect the functional safety

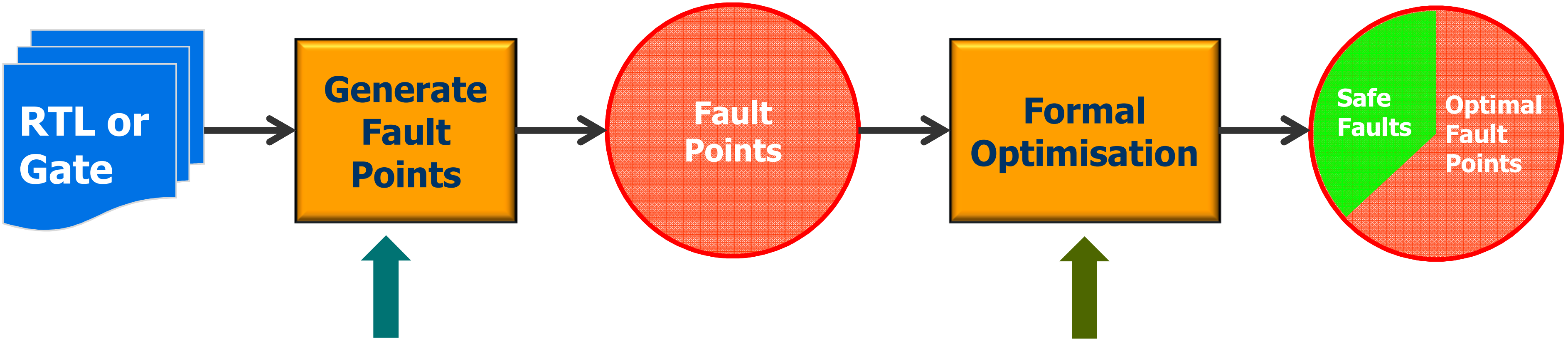
RTL

- Input/output ports
- Internal signals
- Registers
- Memories

Gate-Level

- Cell pins
- Internal nets

Generation of Optimized Fault Points



- Simple Rules
  1) Only Fan-in logic of "Safety Mechanism"
  2) All nodes
     - Exclude silicon-proven Cells/Modules
     - Exclude internal cell nets
     - Exclude not used nets
     - Exclude specific types of cells (Buffers, …)

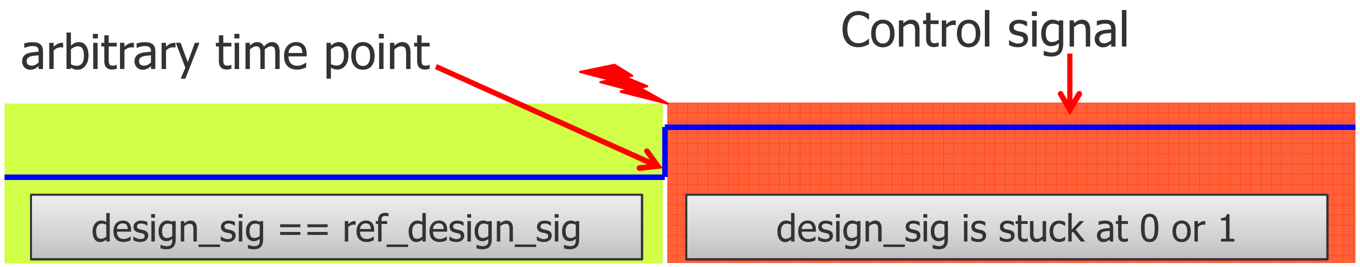
- Advanced Rules
  1) Remove equivalent (collapsible) faults
  2) Remove undetectable faults

Categories for Faults

- Permanent Faults (Stuck at 0, Stuck at 1)
  - Irreversible component damage
- Transient Faults (a.k.a. soft-errors, SEU and SET)
  - Environmental Conditions
  - Cause Erroneous States in the system
  - Do not cause permanent damage
  - Hardest to detect
- Intermittent Faults
  - Caused by unstable HW
  - Often become permanent faults after a period of time

Modelling Faults in Formal

Permanent



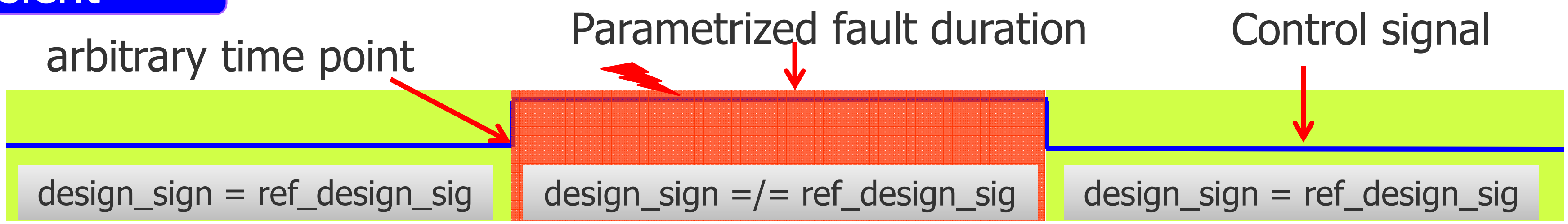
arbitrary time point

Control signal

design_sig == ref_design_sig

design_sig is stuck at 0 or 1

Transient



arbitrary time point

Parametrized fault duration

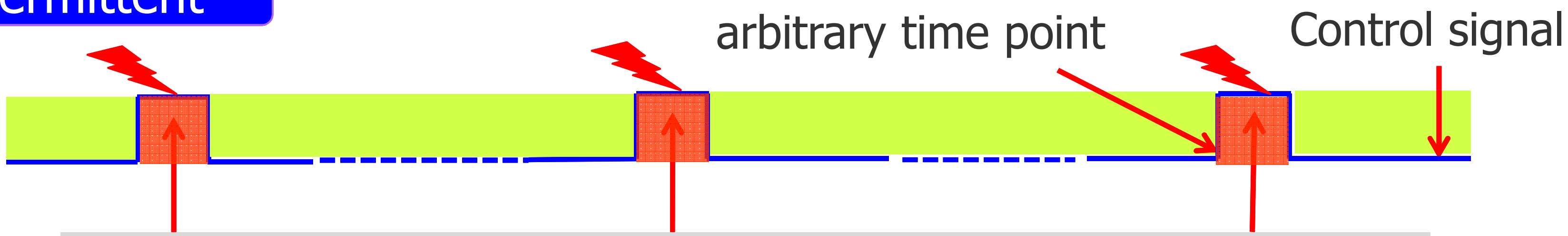
Control signal

design_sign = ref_design_sig

design_sign =/= ref_design_sig

design_sign = ref_design_sig

Intermittent

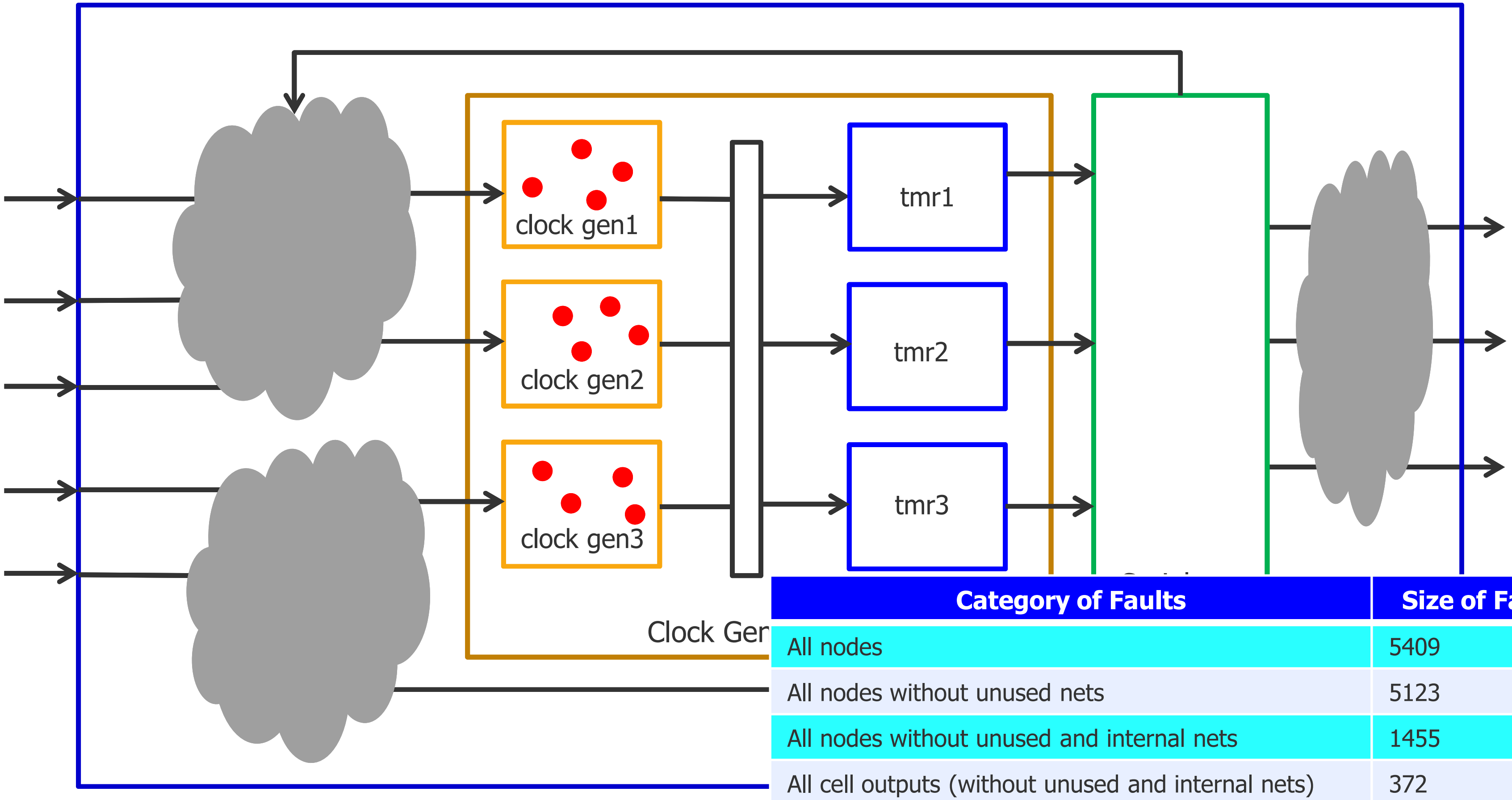


arbitrary time point

Control signal

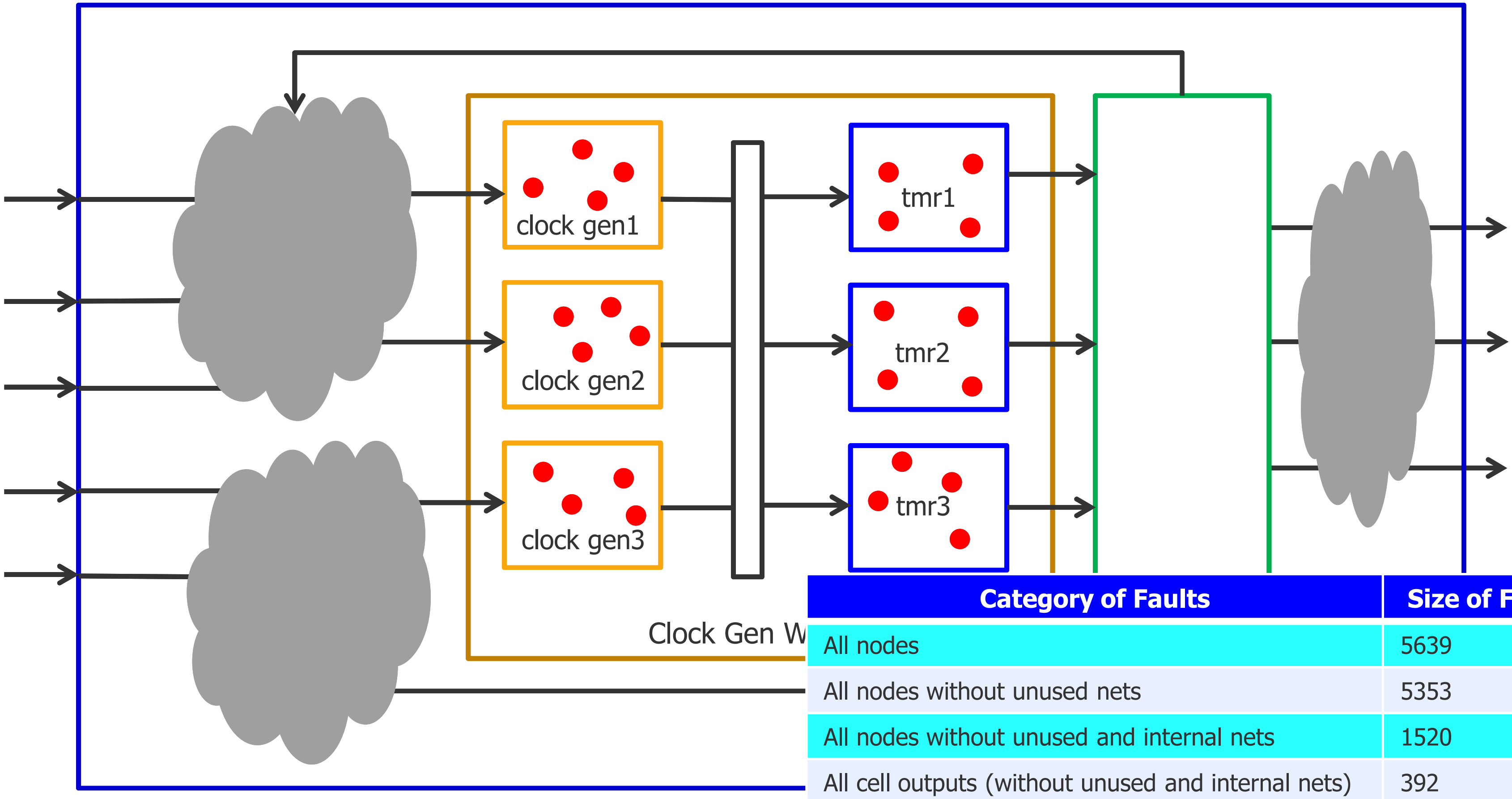
Only at these time points design_sign and ref_design_sig could be different

Size of Fault List:
Assume Faults in Clock Generator



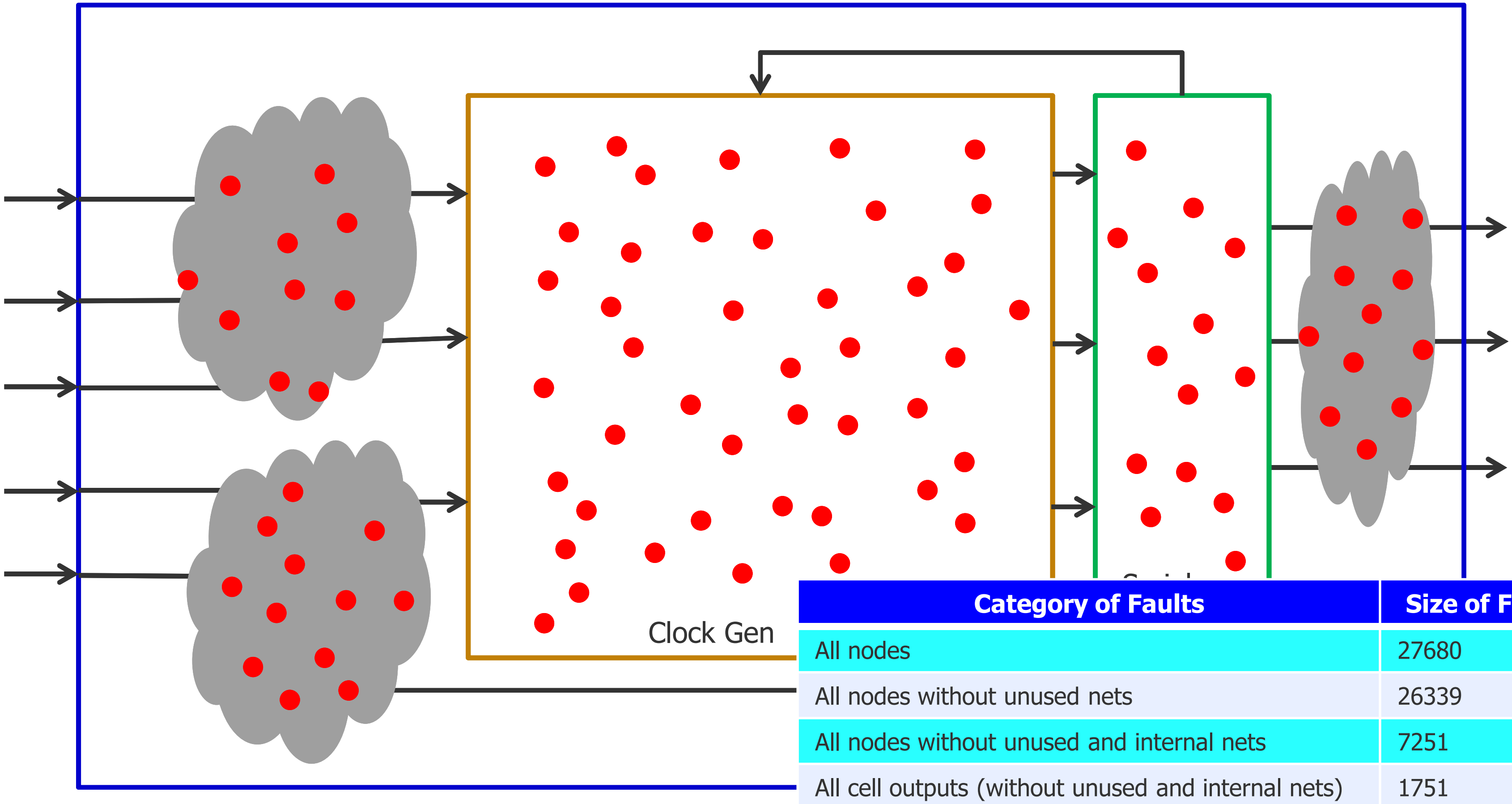
Category of Faults	Size of Fault List
All nodes	5409
All nodes without unused nets	5123
All nodes without unused and internal nets	1455
All cell outputs (without unused and internal nets)	372
All storage cells (without unused and internal nets)	57
All nodes optimized for Stuck-at-0	1287
All nodes optimized for Stuck-at-1	1188

Size of Fault List:
Assume Faults in Clock Generator + TMRs



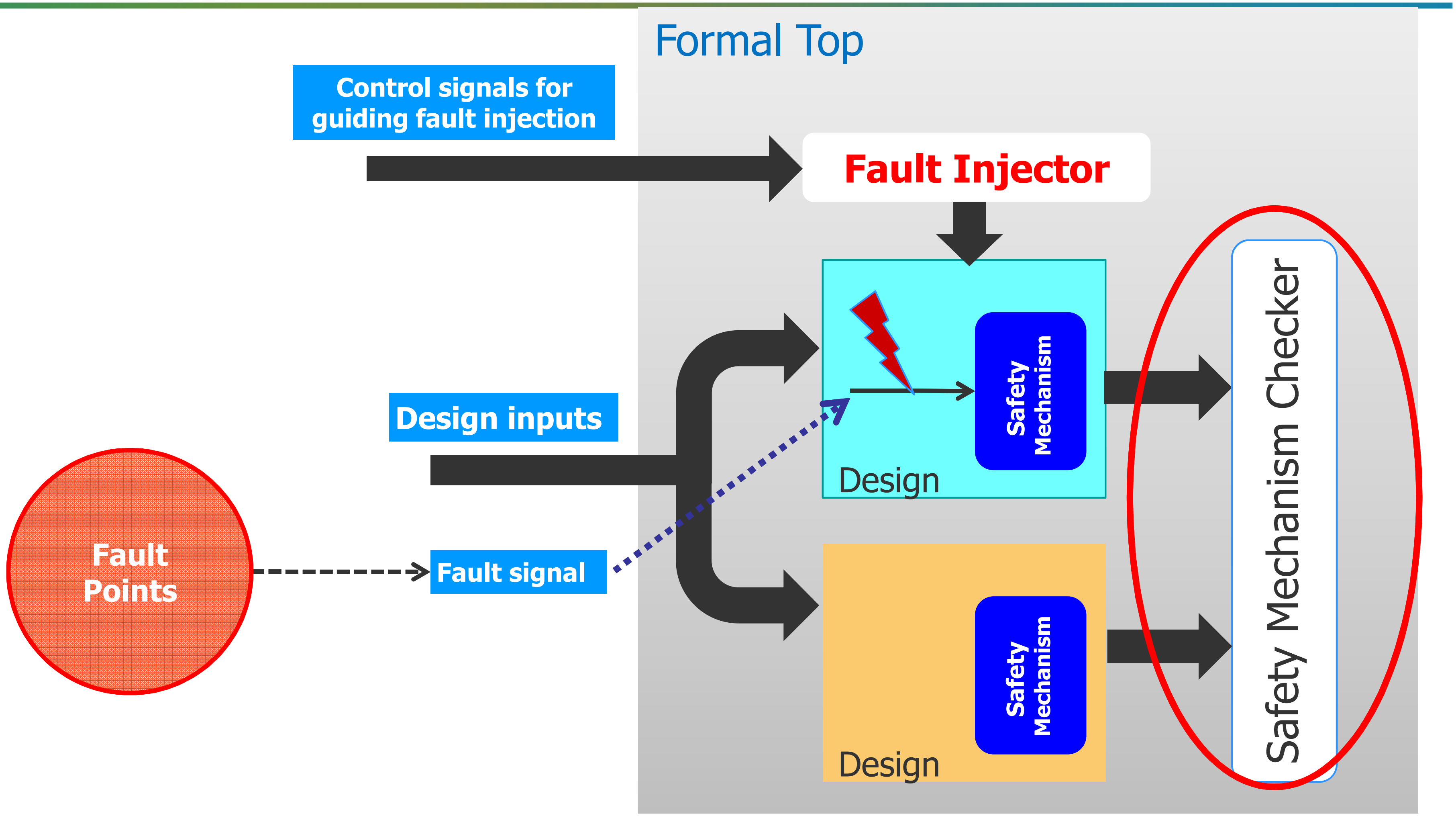
Category of Faults	Size of Fault List
All nodes	5639
All nodes without unused nets	5353
All nodes without unused and internal nets	1520
All cell outputs (without unused and internal nets)	392
All storage cells (without unused and internal nets)	57
All nodes optimized for Stuck-at-0	1341
All nodes optimized for Stuck-at-1	1241

Size of Fault List:
Assume Faults in SPI Top



Category of Faults	Size of Fault List
All nodes	27680
All nodes without unused nets	26339
All nodes without unused and internal nets	7251
All cell outputs (without unused and internal nets)	1751
All storage cells (without unused and internal nets)	267
All nodes optimized for Stuck-at-0	6639
All nodes optimized for Stuck-at-1	6460

Questa Formal Model for Fault Injection



Formal Results

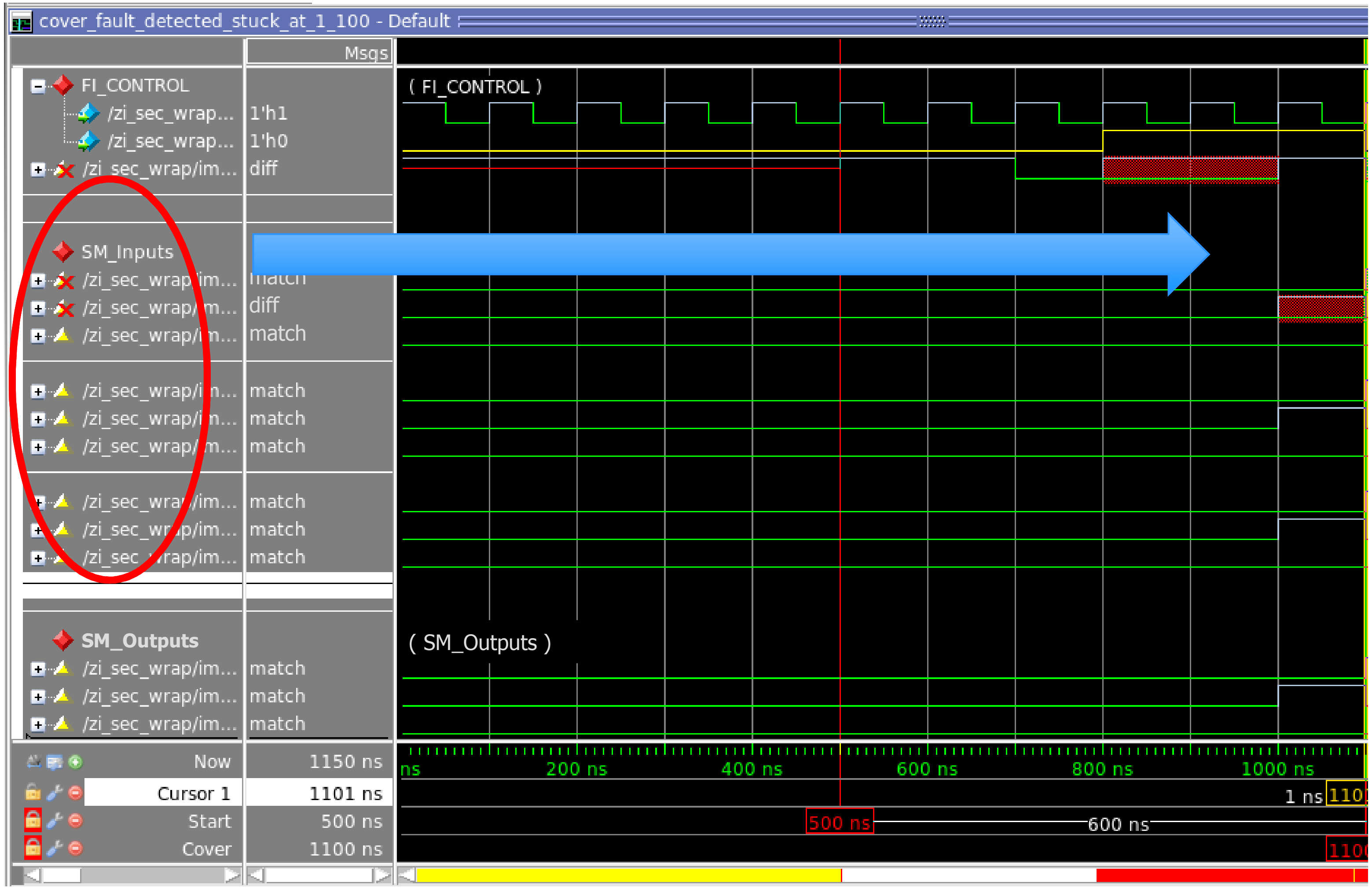
P Fault injected and SM behaves correctly

F Fault injected and SM behaves incorrectly

C Fault injected and detected/observed

U Fault is undetectable

Fault Detected

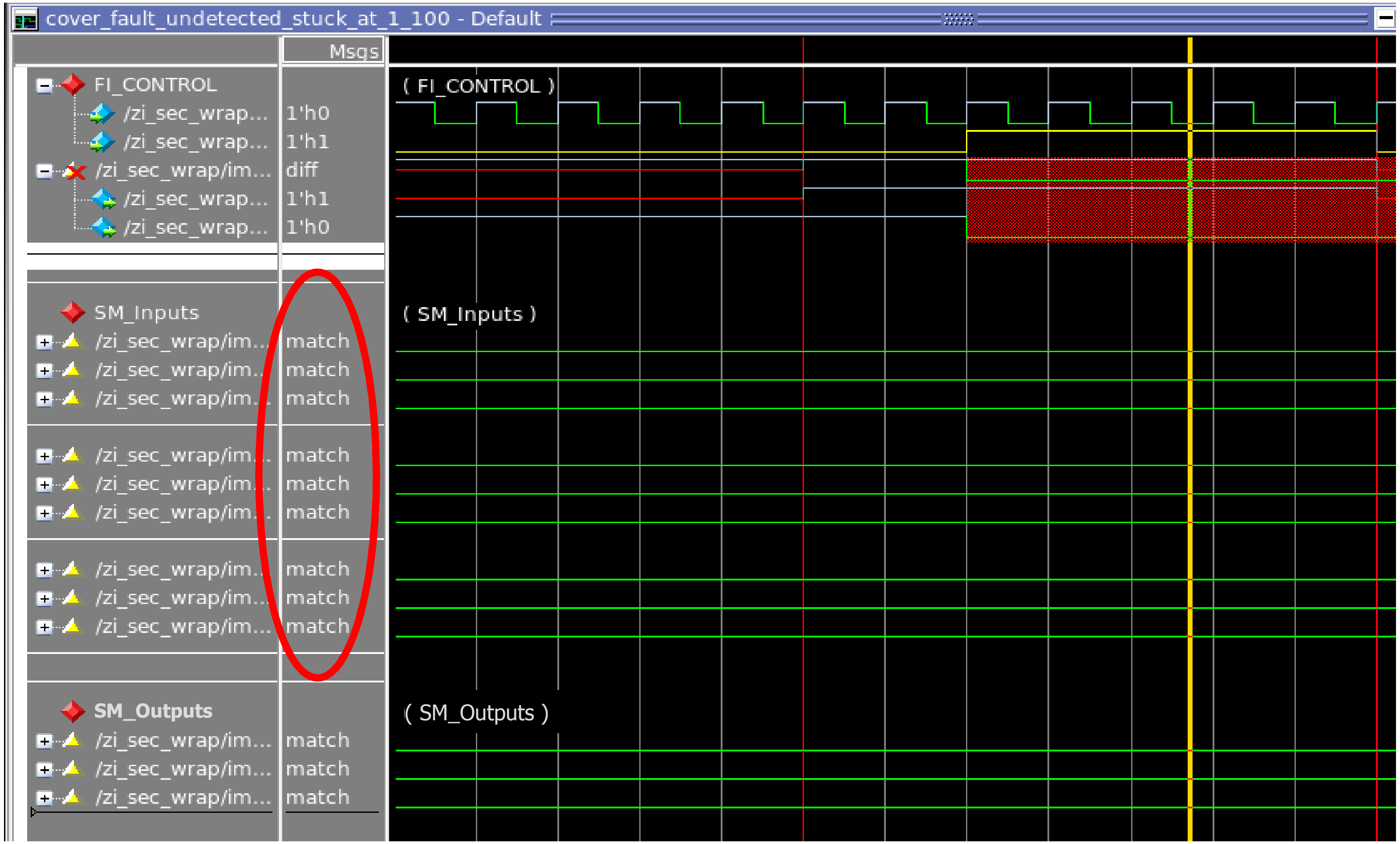


cover_fault_detected_stuck_at_1_100 - Default

	Msgs
FI_CONTROL	
/zi_sec_wrap...	1'h1
/zi_sec_wrap...	1'h0
/zi_sec_wrap/im...	diff
SM_Inputs	
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	diff
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
SM_Outputs	
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
Now	1150 ns
Cursor 1	1101 ns
Start	500 ns
Cover	1100 ns

Questa Formal-based fault injection, 2016

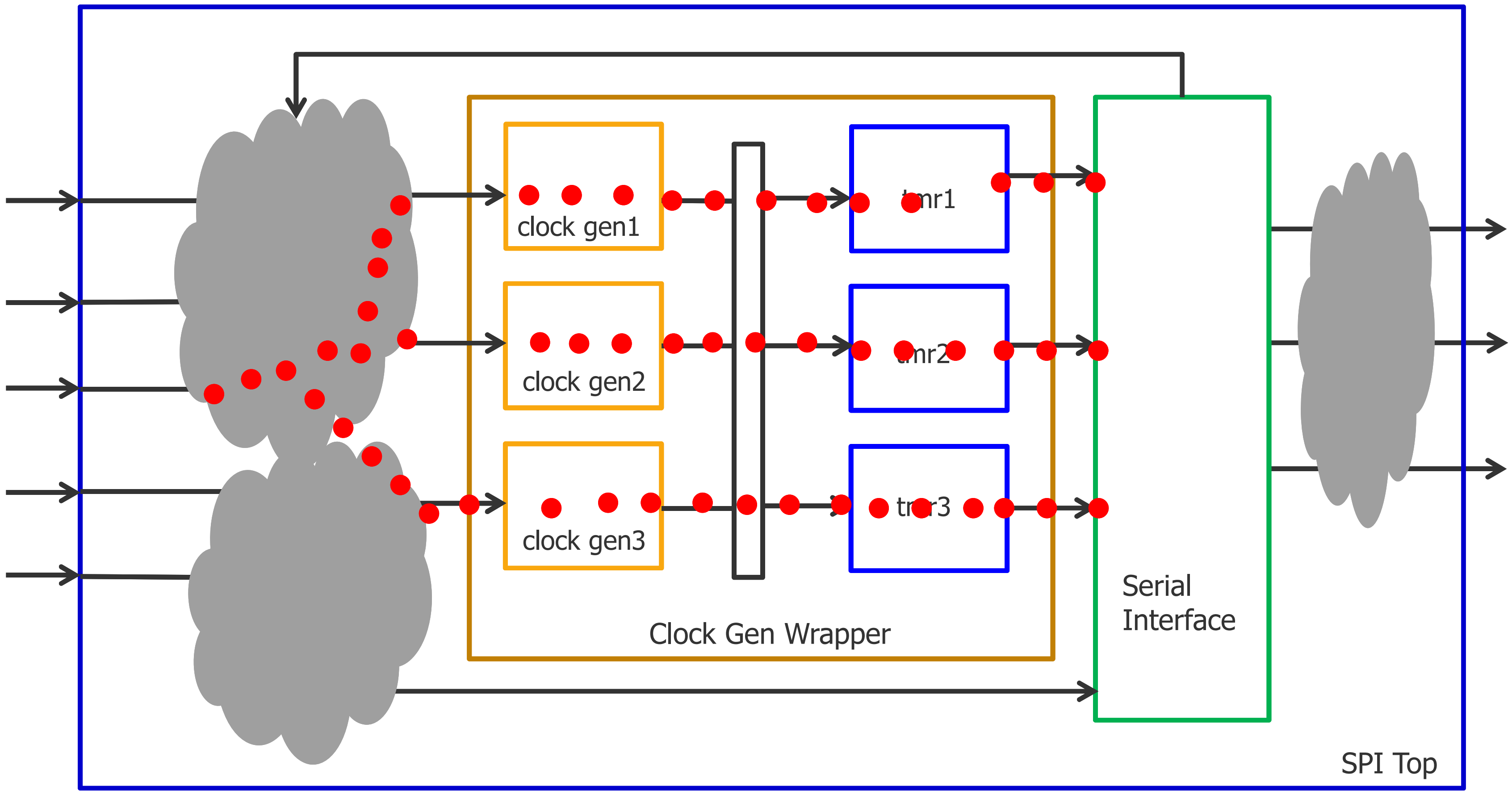
Undetected/Masked Fault



cover_fault_undetected_stuck_at_1_100 - Default

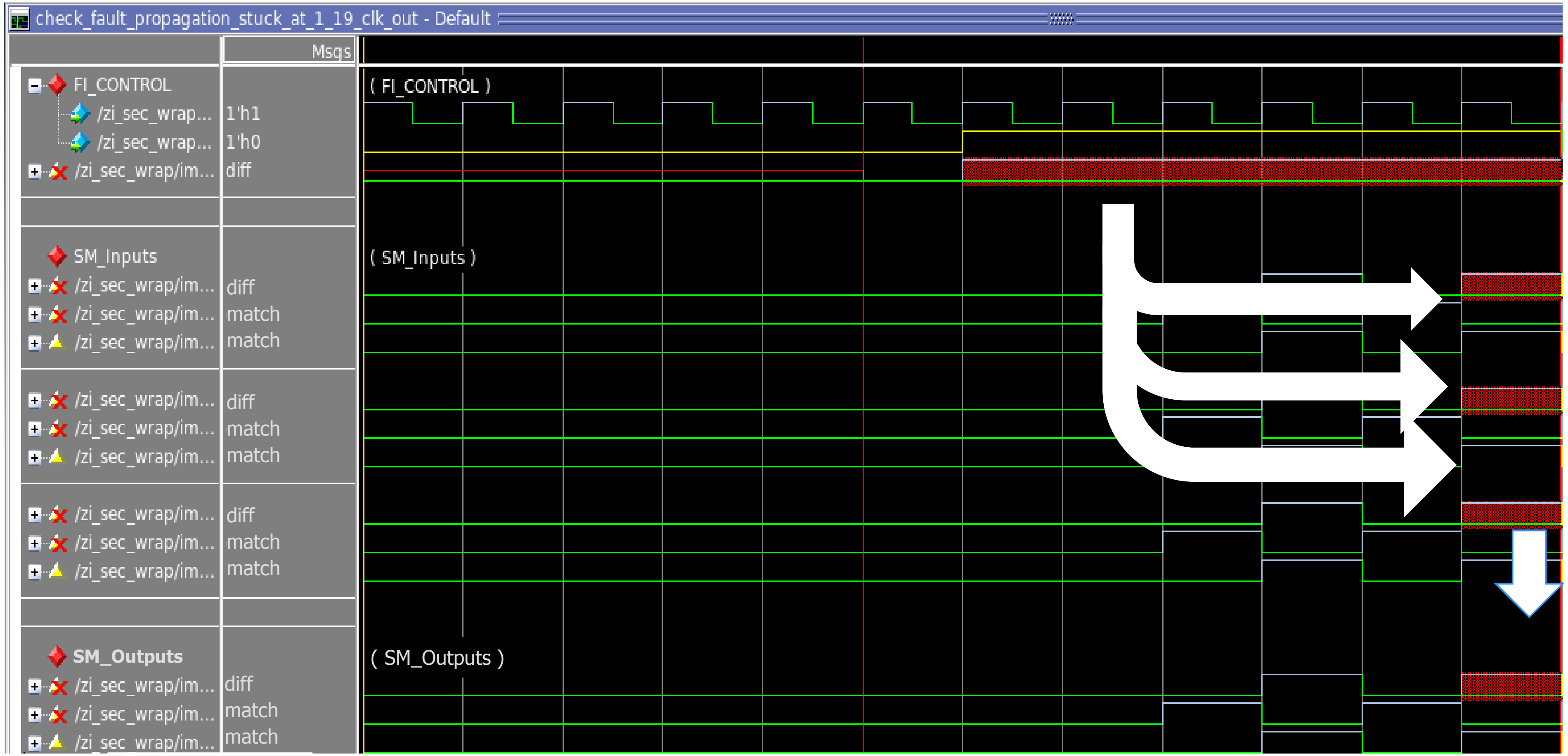
	Msgs
FI_CONTROL	
/zi_sec_wrap...	1'h0
/zi_sec_wrap...	1'h1
/zi_sec_wrap/im...	diff
/zi_sec_wrap...	1'h1
/zi_sec_wrap...	1'h0
SM_Inputs	
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
SM_Outputs	
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match
/zi_sec_wrap/im...	match

Fault Outside of SM defined Scope



** extended with for faulty injection*

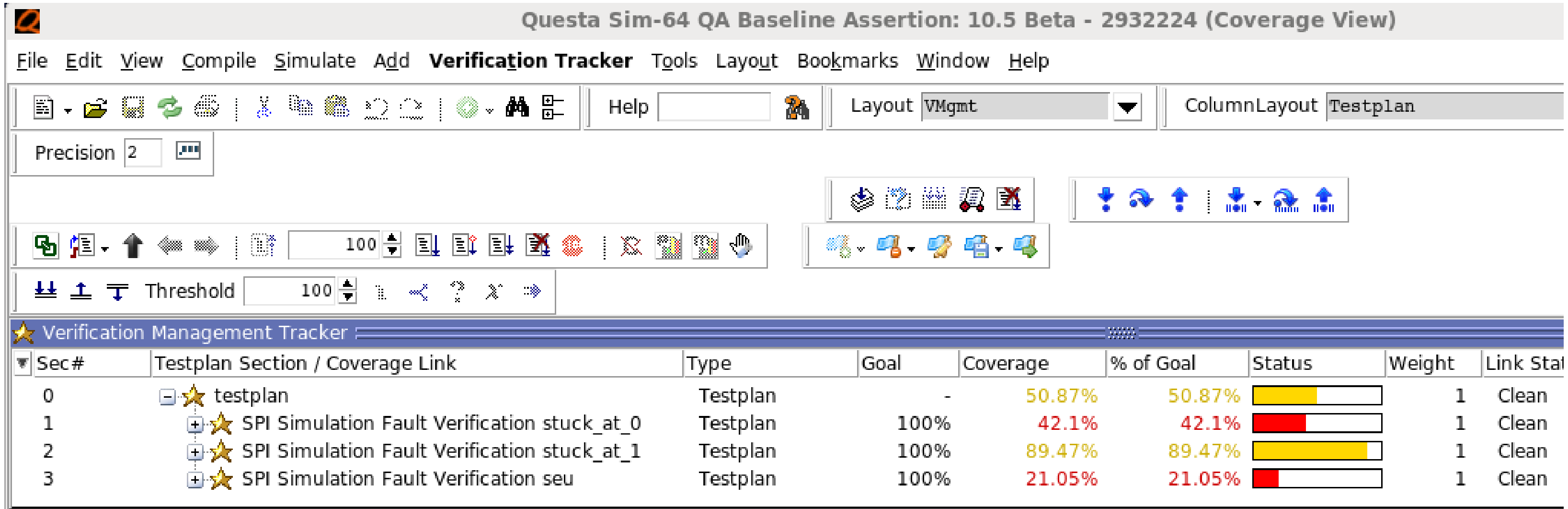
Fault Propagated



check_fault_propagation_stuck_at_1_19_clk_out - Default

```
                      Msgs
FI_CONTROL
  /zi_sec_wrap...     1'h1
  /zi_sec_wrap...     1'h0
  /zi_sec_wrap/im...  diff

SM_Inputs
  /zi_sec_wrap/im...  diff
  /zi_sec_wrap/im...  match
  /zi_sec_wrap/im...  match

  /zi_sec_wrap/im...  diff
  /zi_sec_wrap/im...  match
  /zi_sec_wrap/im...  match

  /zi_sec_wrap/im...  diff
  /zi_sec_wrap/im...  match
  /zi_sec_wrap/im...  match

SM_Outputs
  /zi_sec_wrap/im...  diff
  /zi_sec_wrap/im...  match
  /zi_sec_wrap/im...  match
```

SPI Master Core
Fault Coverage Results



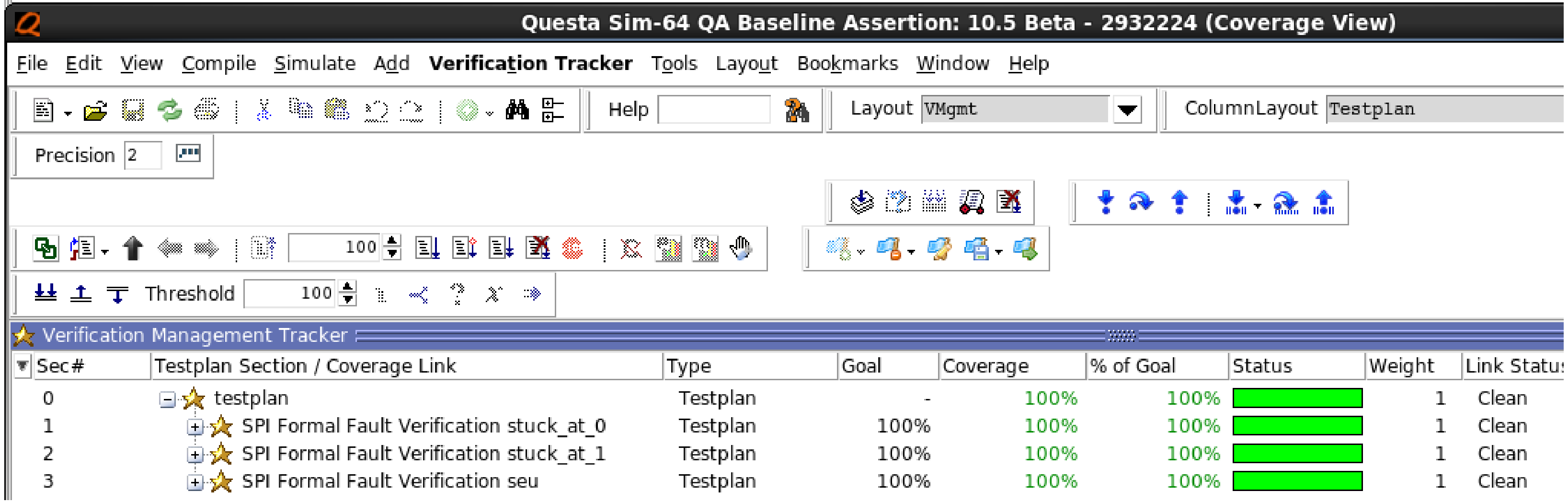
Questa Sim-64 QA Baseline Assertion: 10.5 Beta - 2932224 (Coverage View)
File Edit View Compile Simulate Add Verification Tracker Tools Layout Bookmarks Window Help
Help
Verification Management Tracker

Layout VMgmt ColumnLayout Testplan

Precision 2

Threshold 100

Sec#	Testplan Section / Coverage Link	Type	Goal	Coverage	% of Goal	Status	Weight	Link Stat
0	testplan	Testplan	-	50.87%	50.87%	🟨	1	Clean
1	SPI Simulation Fault Verification stuck_at_0	Testplan	100%	42.1%	42.1%	🟥	1	Clean
2	SPI Simulation Fault Verification stuck_at_1	Testplan	100%	89.47%	89.47%	🟨	1	Clean
3	SPI Simulation Fault Verification seu	Testplan	100%	21.05%	21.05%	🟥	1	Clean



Questa Sim-64 QA Baseline Assertion: 10.5 Beta - 2932224 (Coverage View)
File Edit View Compile Simulate Add Verification Tracker Tools Layout Bookmarks Window Help
Help
Verification Management Tracker

Layout VMgmt ColumnLayout Testplan

Precision 2

Threshold 100

Sec#	Testplan Section / Coverage Link	Type	Goal	Coverage	% of Goal	Status	Weight	Link Statu
0	testplan	Testplan	-	100%	100%	🟩	1	Clean
1	SPI Formal Fault Verification stuck_at_0	Testplan	100%	100%	100%	🟩	1	Clean
2	SPI Formal Fault Verification stuck_at_1	Testplan	100%	100%	100%	🟩	1	Clean
3	SPI Formal Fault Verification seu	Testplan	100%	100%	100%	🟩	1	Clean

Summary

- Functional safety critical components are often small enough to be analyzed using formal techniques
- Formal fault injection is exhaustive regarding legal design input pattern AND failure time points
- Questa Formal Fault Injection can enable you to reach your safety verification targets

Mentor Graphics®

www.mentor.com