2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

USF-based FMEDA-driven Functional Safety Verification

Francesco Lertora, Software Engineering Group Director, SVG
Mangesh Mukundrao Pande, Solutions Architect, SVG
Pete Hardee, Group Director Product Management, SVG

04 March 2024

cādence®

Outline

- Session 1
  - Introduction
  - Functional Safety Analysis Overview
  - Deep Dive
    - Architectural FMEDA
    - Detailed FMEDA
    - Safety Metrics Verification
- [Break]
- Session 2
  - Fault Campaign Management
- Summary

EDA as an Ecosystem of International and Industry Standards

1800 - IEEE Standard for SystemVerilog Unified Hardware Design, Specification, and Verification Language

1364 - IEEE Standard for Verilog Hardware Description Language

1076 - IEEE Standard for VHDL Language Reference Manual

Library Exchange Format (LEF)/Design Exchange Format (DEF)

1801 - IEEE Standard for Design and Verification of Low-Power, Energy-Aware Electronic Systems

Timing Constraints – SDC

1497 IEEE Standard for Standard Delay Format (SDF) for the Electronic Design Process

Liberty™ library format

GDSII - Graphic Design System

OASIS® – Open Artwork System Interchange Standard

1685 - IEEE Standard for IP-XACT, Standard Structure for Packaging, Integrating, and Reusing IP within Tool Flows

Why not for safety?

- Describe safety features, targets (intent) and exchange safety-related information

Motivations & Mission

- Lack of formalism, standards ambiguity, differentiated assessors scenario, lead to customer-specific methodologies + widespread usage of Spreadsheets
  - «consulting-driven» market side-effects:
    - ‘keep it obscure’
    - ‘this is *my* (certified) methodology’
    - ‘(only) We will tell you what you have to do’...etc...

To develop a modular safety analysis platform to exchange safety-related information and to enable Design For Safety with Cadence® Tools

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES SAN JOSE, CA, USA MARCH 4-7, 2024

cadence®

Cadence Approach

- Modularity
  - A solution that can be adapted and scaled to different scenarios
- Defined scope
  - A set of kernel functionalities - Rooted by safety analysis capabilies
- Not enforcing a «methodology»
- EDA ‘friendly’

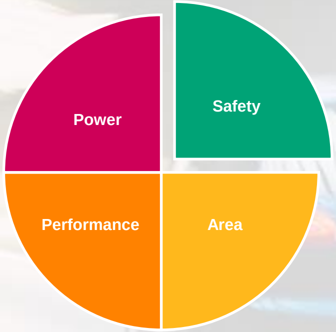
Current Status

- Accellera Functional Safety Working Group (FSWG)
  - Second White Paper Published - December 2023
  - Cadence was part of the WG formation and kick-off in 2019
  - Being the collaborative work of entities the final Accelera proposal will be different from USF

- IEEE Std 2851™- 2023 – “Standard for Functional Safety Data Format for Interoperability within the Dependability Lifecycle”
  - “dot standards” will follow
  - IEEE to adopt the Accelera FSWG work on FMEDA

Cadence is committed to adopt and support the IEEE 2851 family of standards

- Where we are going:
  - Safety Analysis: an international standard to share safety information
  - Safety Implementation: adding a new variable to PPA

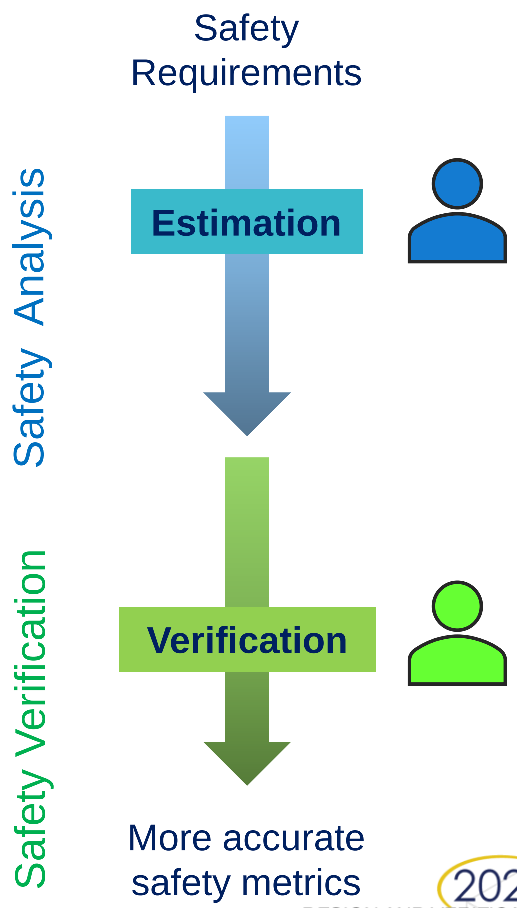


2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

cadence®

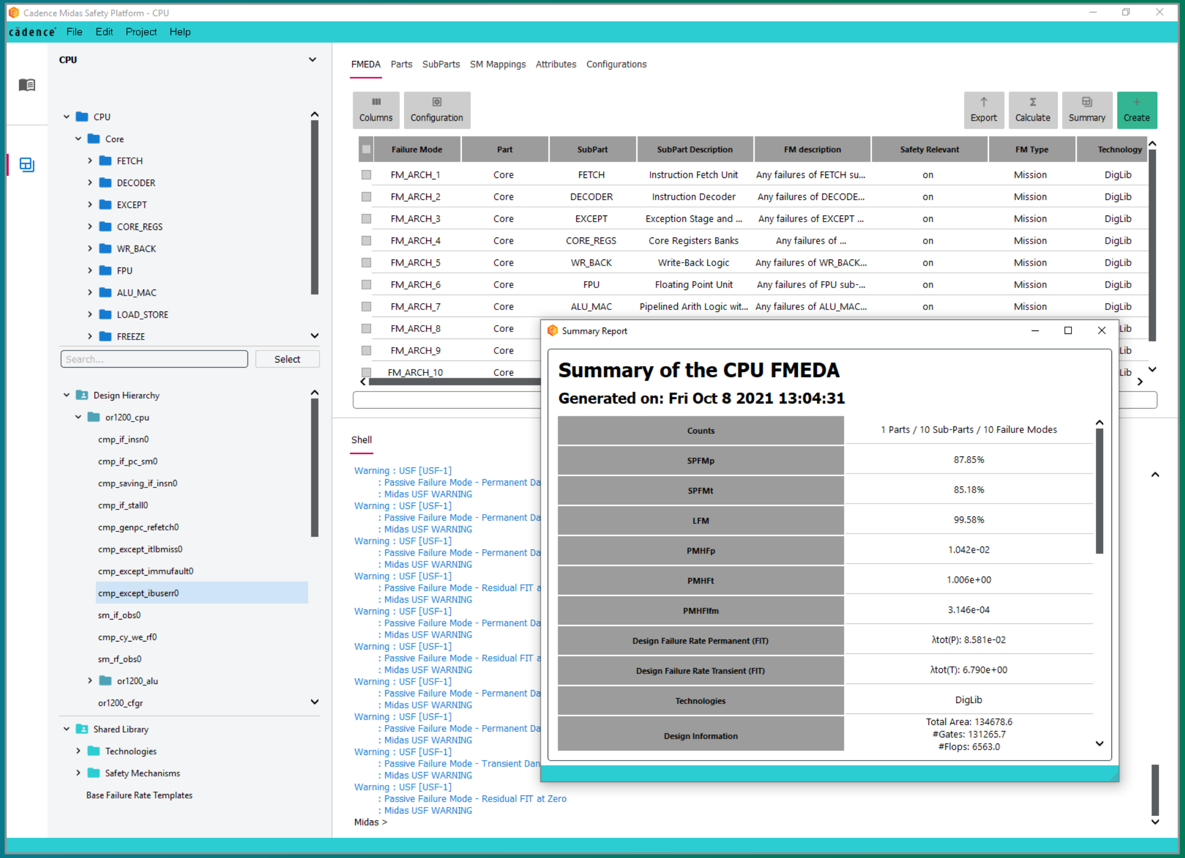
Closing the Gap between FMEDA and Safety Verification

	Abstraction	Safety Step	User
Functional Safety Concept	Functional	FMEA	Safety Architect (System level)
Technical Safety Concept SoC	Block Diagram	FMEDA (architectural)	Safety Architect (SoC level)
SoC Design	RTL/netlist	FMEDA (detailed)	Safety Engineer (RTL/gate level)
SoC Safety Verification	Netlist	Safety Verification (Formal/Fault Injection)	Safety Verification Engineer
Safety Metrics	Verification Result	FMEDA backannotation	Safety Verification Engineer



Midas Safety Platform for FMEDA-driven Functional Safety

- **Midas™ Safety Platform** driving analog and digital flows for FMEDA-based functional safety
- Early phase safety analysis and architecture exploration
- Automated safety mechanism insertion and verification
- Native chip design data for accuracy and detailed safety analysis
- Unified Safety Format (USF) support



□	Failure Mode	Part	SubPart	SubPart Description	FM description	Safety Relevant	FM Type	Technology
□	FM_ARCH_1	Core	FETCH	Instruction Fetch Unit	Any failures of FETCH su...	on	Mission	DigLib
□	FM_ARCH_2	Core	DECODER	Instruction Decoder	Any failures of DECODE...	on	Mission	DigLib
□	FM_ARCH_3	Core	EXCEPT	Exception Stage and ...	Any failures of EXCEPT ...	on	Mission	DigLib
□	FM_ARCH_4	Core	CORE_REGS	Core Registers Banks	Any failures of ...	on	Mission	DigLib
□	FM_ARCH_5	Core	WR_BACK	Write-Back Logic	Any failures of WR_BACK...	on	Mission	DigLib
□	FM_ARCH_6	Core	FPU	Floating Point Unit	Any failures of FPU sub-...	on	Mission	DigLib
□	FM_ARCH_7	Core	ALU_MAC	Pipelined Arith Logic wit...	Any failures of ALU_MAC...	on	Mission	DigLib
□	FM_ARCH_8	Core						Lib
□	FM_ARCH_9	Core						Lib
□	FM_ARCH_10	Core						Lib

```
Warning : USF [USF-1]
        : Passive Failure Mode - Permanent Da
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode - Permanent Da
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode - Permanent Da
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode - Residual FIT a
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode - Permanent Da
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode - Residual FIT a
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode - Permanent Da
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode - Permanent Da
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode - Transient Dan
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode - Residual FIT at Zero
        : Midas USF WARNING
Midas >
```

Summary of the CPU FMEDA

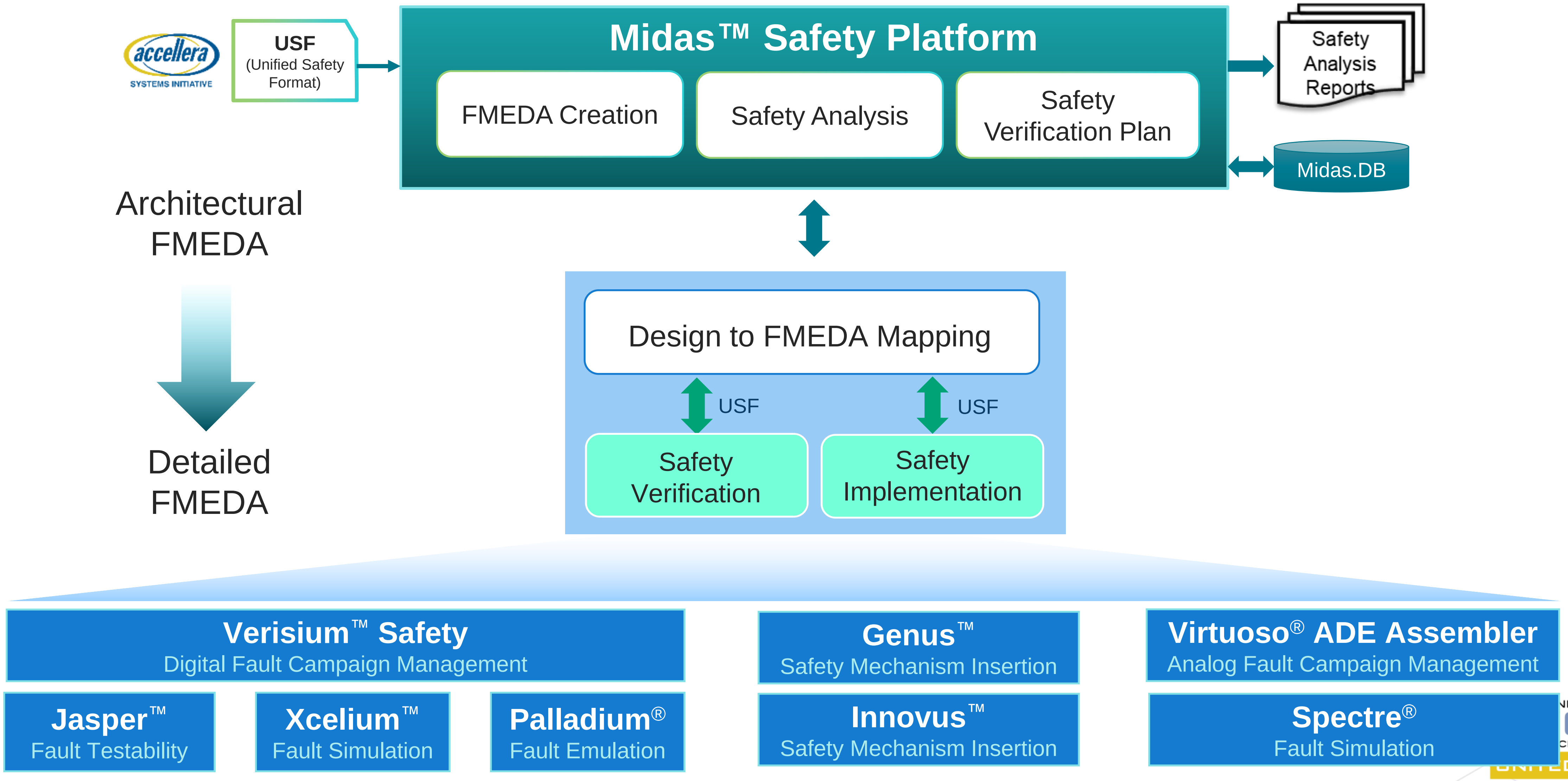
Generated on: Fri Oct 8 2021 13:04:31

Counts	1 Parts / 10 Sub-Parts / 10 Failure Modes
SPFMp	87.85%
SPFMt	85.18%
LFM	99.58%
PMHFp	1.042e-02
PMHFt	1.006e+00
PMHFlfm	3.146e-04
Design Failure Rate Permanent (FIT)	λtot(P): 8.581e-02
Design Failure Rate Transient (FIT)	λtot(T): 6.790e+00
Technologies	DigLib
Design Information	Total Area: 134678.6 #Gates: 131265.7 #Flops: 6563.0

UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

cādence®

Cadence Functional Safety Full Flow



accellera SYSTEMS INITIATIVE

USF (Unified Safety Format)

Midas™ Safety Platform

FMEDA Creation

Safety Analysis

Safety Verification Plan

Safety Analysis Reports

Midas.DB

Architectural FMEDA

Detailed FMEDA

Design to FMEDA Mapping

USF

USF

Safety Verification

Safety Implementation

Verisium™ Safety
Digital Fault Campaign Management

Jasper™
Fault Testability

Xcelium™
Fault Simulation

Palladium®
Fault Emulation

Genus™
Safety Mechanism Insertion

Innovus™
Safety Mechanism Insertion

Virtuoso® ADE Assembler
Analog Fault Campaign Management

Spectre®
Fault Simulation

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES SAN JOSE, CA, USA MARCH 4-7, 2024

cadence®

Digital Safety Verification

Fault campaign management, analysis, simulation and emulation

Fault Campaign Management – Verisium Safety

- Unified campaign management across all engines
- Backannotation of DC results into Midas FMEDA
- Provides requirements traceability and reporting

Fault Analysis – Jasper FSV App

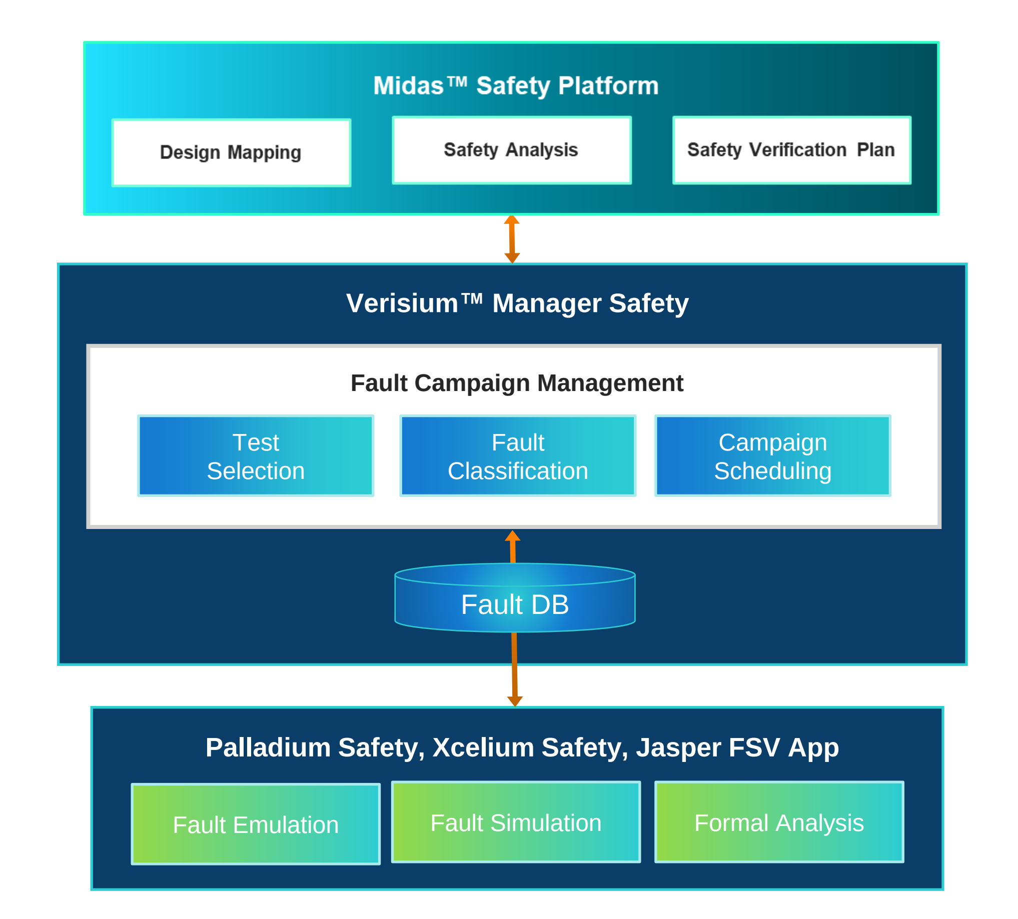
- Structural analysis to reduce the fault list
- Formal analysis for accurate fault classification

Fault Simulation – Xcelium Safety

- Native serial and concurrent fault verification
- Same simulator for functional verification (GOOD machine) and fault simulation (BAD machine)

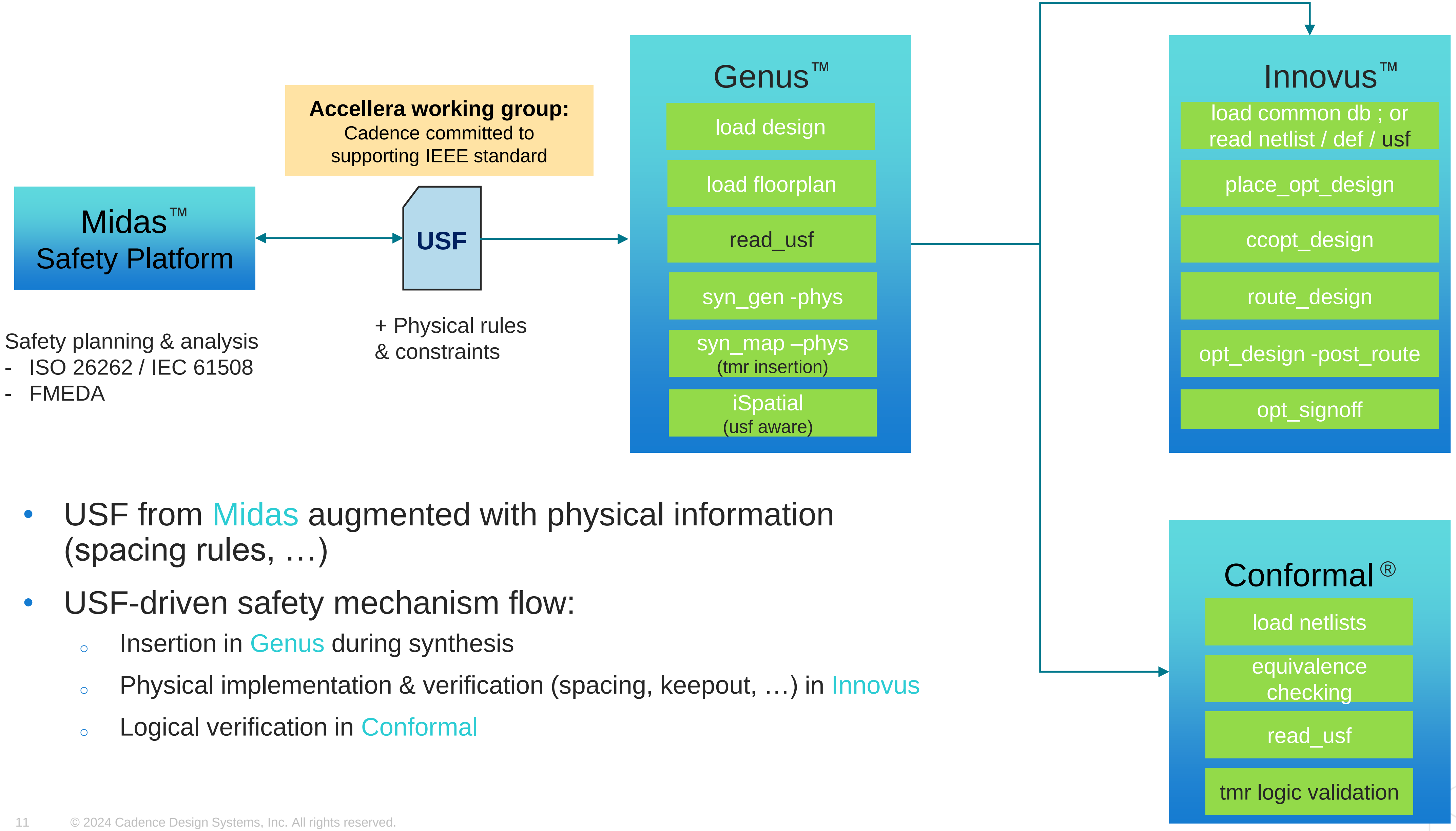
Fault Emulation – Palladium Safety

- Run full SoC with SW or STLs



SAFETY

Cadence Automotive Safety / USF-Driven Flow

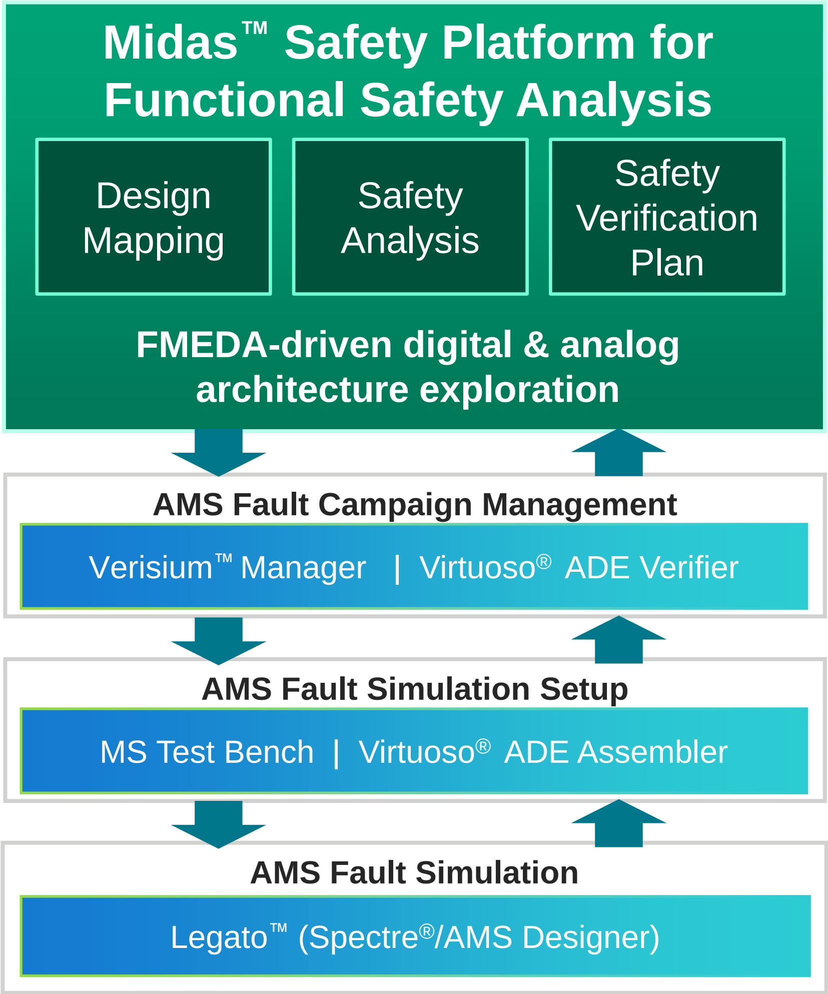


- USF from Midas augmented with physical information (spacing rules, …)
- USF-driven safety mechanism flow:
  - Insertion in Genus during synthesis
  - Physical implementation & verification (spacing, keepout, …) in Innovus
  - Logical verification in Conformal

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES SAN JOSE, CA, USA MARCH 4-7, 2024

cadence®

Midas AMS Functional Safety Flow Overview



Midas™ Safety Platform for Functional Safety Analysis

Design Mapping

Safety Analysis

Safety Verification Plan

FMEDA-driven digital & analog architecture exploration

AMS Fault Campaign Management

Verisium™ Manager | Virtuoso® ADE Verifier

AMS Fault Simulation Setup

MS Test Bench | Virtuoso® ADE Assembler

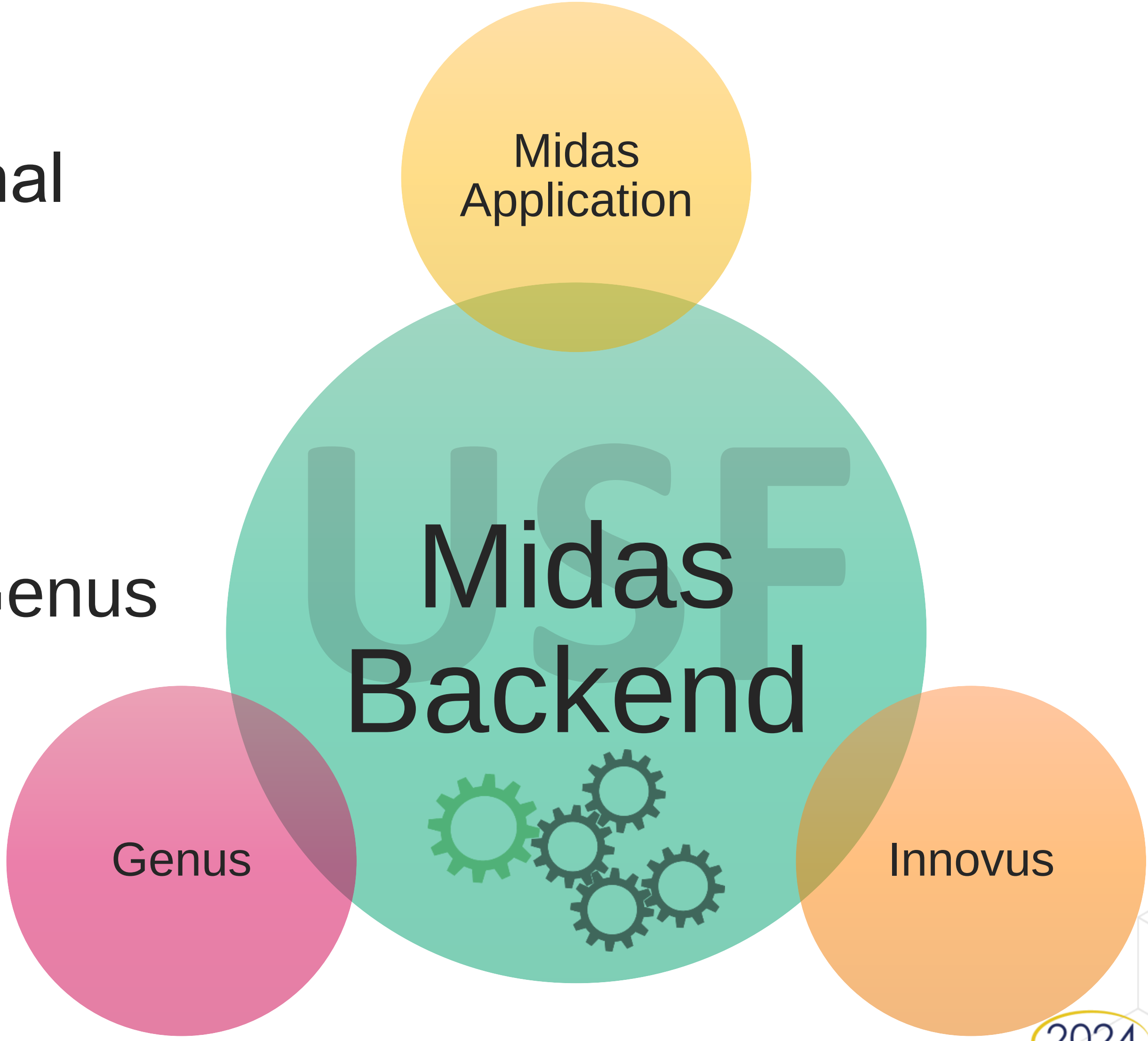
AMS Fault Simulation

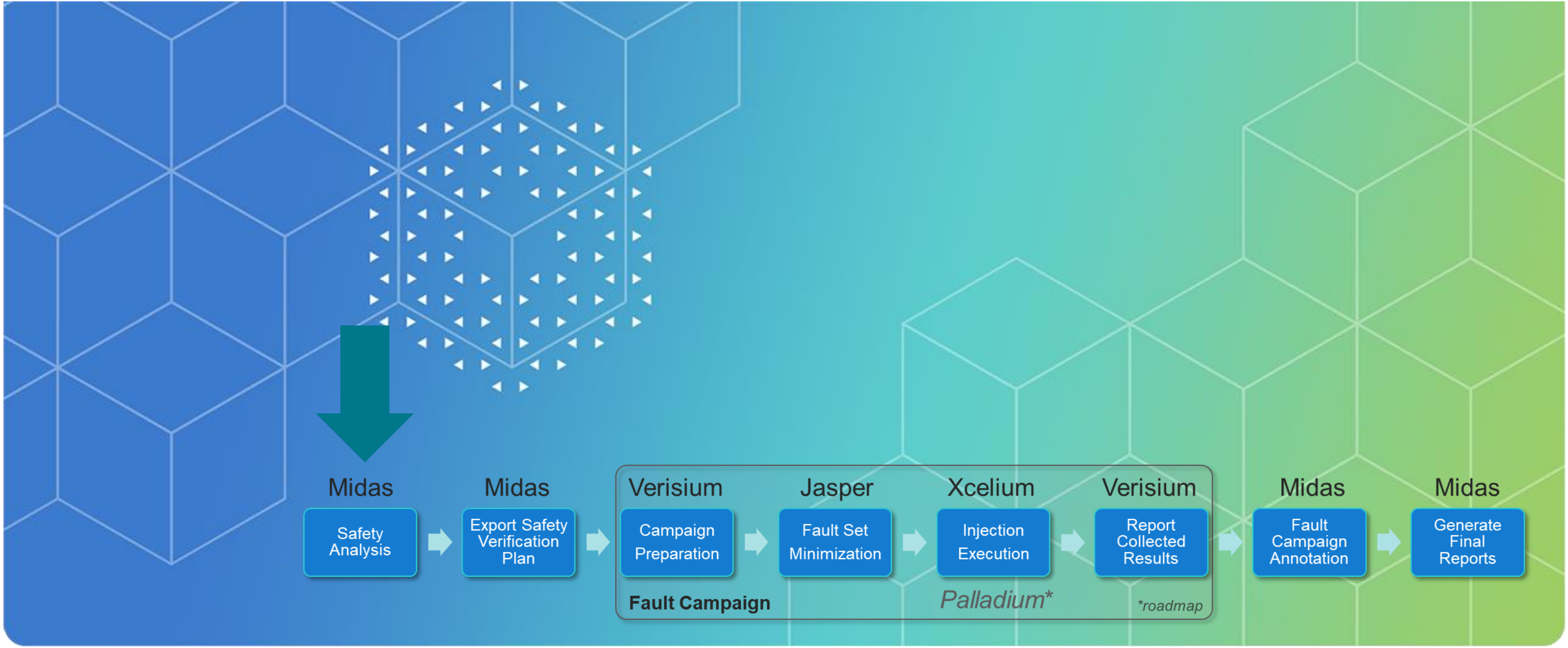
Legato™ (Spectre®/AMS Designer)

- Connect FMEDA data to design data
  - Import schematic/RTL hierarchy into Midas (DHE)
  - Direct mapping of Safety objects to Design objects
  - Generate fault campaigns inside Midas for various failure modes
- Clean hand-off from FuSa lead to IC design teams
  - Digital-centric or analog-centric AMS flows
  - Automated fault campaign management: ADE® Verifier and Assembler (analog-centric) or Verisium Manager (digital-centric)
  - Run fault campaigns with Virtuoso, Verisium Manager, and Legato
- Improve the accuracy & traceability of safety metrics
  - Back-annotate key safety metrics back to Midas

Midas Safety Platform Modularity

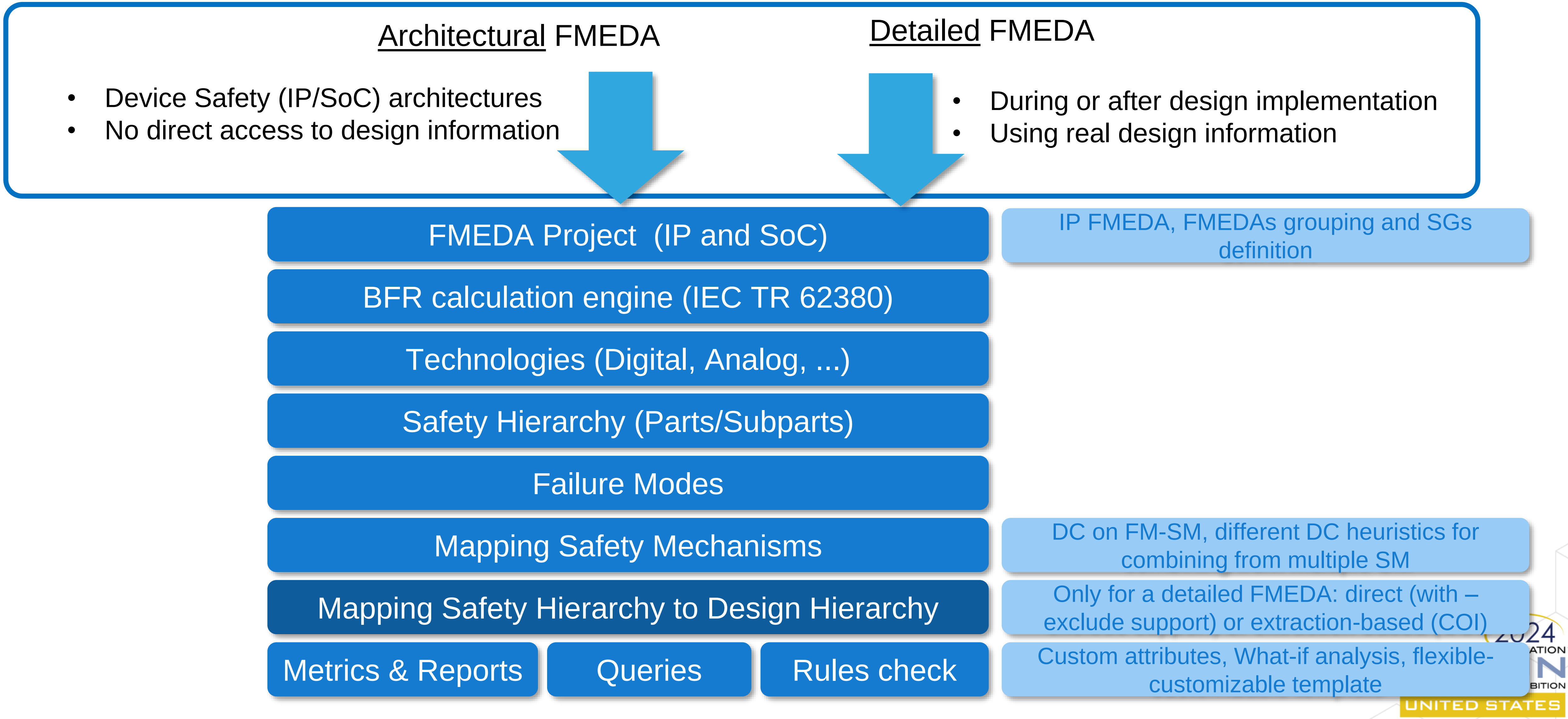
- The Midas backend is the ‘functional safety engine’
  - Support for Midas command line interface
  - ISO26262; IEC61508
  - BFR
- Same backend is integrated into Genus and Innovus
- Core features can be made easily available in different contexts





Functional Safety Analysis Overview

Functional Safety Analysis



Architectural FMEDA

- Device Safety (IP/SoC) architectures
- No direct access to design information

Detailed FMEDA

- During or after design implementation
- Using real design information

FMEDA Project (IP and SoC)

IP FMEDA, FMEDAs grouping and SGs definition

BFR calculation engine (IEC TR 62380)

Technologies (Digital, Analog, ...)

Safety Hierarchy (Parts/Subparts)

Failure Modes

Mapping Safety Mechanisms

DC on FM-SM, different DC heuristics for combining from multiple SM

Mapping Safety Hierarchy to Design Hierarchy

Only for a detailed FMEDA: direct (with – exclude support) or extraction-based (COI)

Metrics & Reports

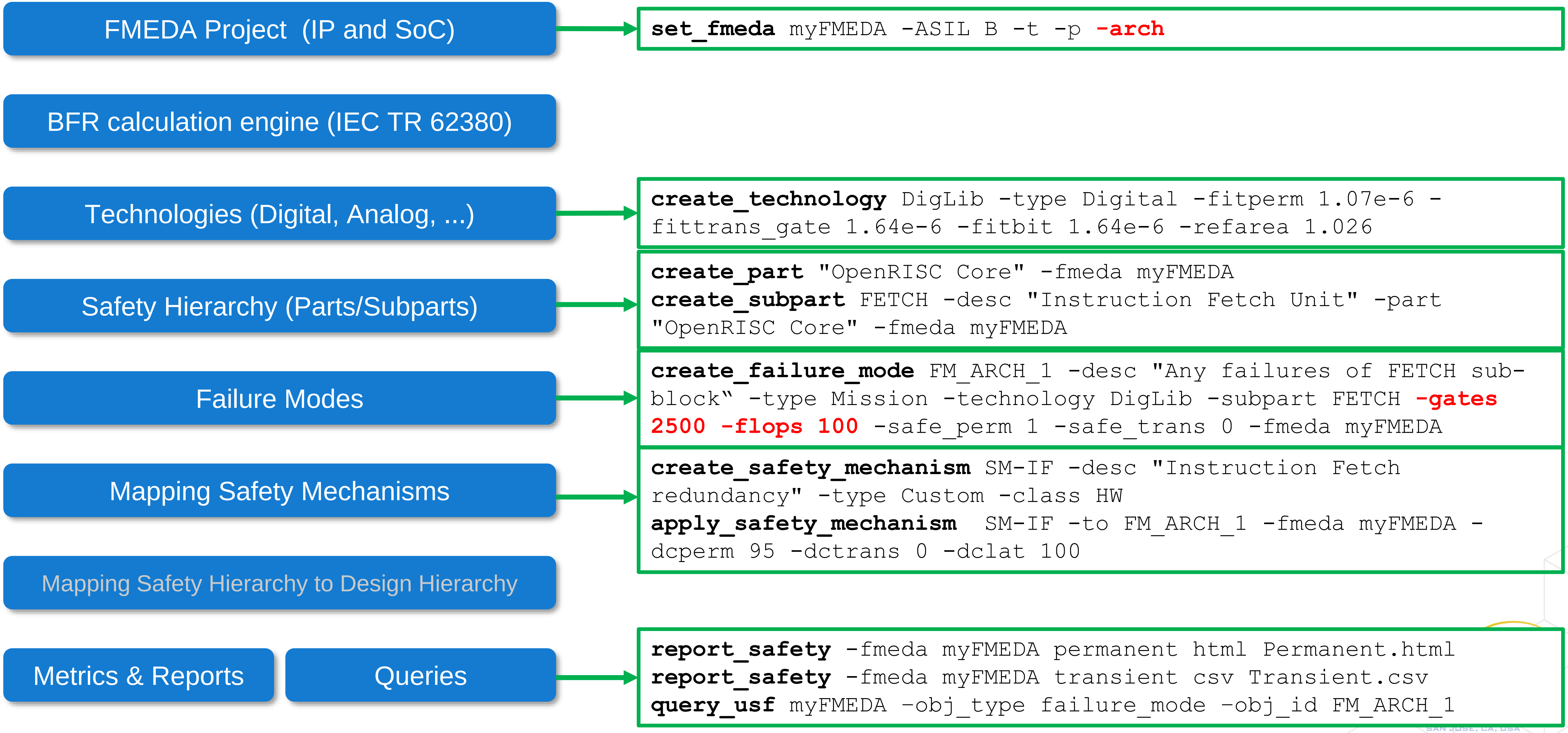
Queries

Rules check

Custom attributes, What-if analysis, flexible-customizable template

Architectural FMEDA

USF



FMEDA Project (IP and SoC)

```
set_fmeda myFMEDA -ASIL B -t -p -arch
```

BFR calculation engine (IEC TR 62380)

Technologies (Digital, Analog, ...)

```
create_technology DigLib -type Digital -fitperm 1.07e-6 -
fittrans_gate 1.64e-6 -fitbit 1.64e-6 -refarea 1.026
```

Safety Hierarchy (Parts/Subparts)

```
create_part "OpenRISC Core" -fmeda myFMEDA
create_subpart FETCH -desc "Instruction Fetch Unit" -part
"OpenRISC Core" -fmeda myFMEDA
```

Failure Modes

```
create_failure_mode FM_ARCH_1 -desc "Any failures of FETCH sub-
block‟ -type Mission -technology DigLib -subpart FETCH -gates
2500 -flops 100 -safe_perm 1 -safe_trans 0 -fmeda myFMEDA
```

Mapping Safety Mechanisms

```
create_safety_mechanism SM-IF -desc "Instruction Fetch
redundancy" -type Custom -class HW
apply_safety_mechanism  SM-IF -to FM_ARCH_1 -fmeda myFMEDA -
dcperm 95 -dctrans 0 -dclat 100
```

Mapping Safety Hierarchy to Design Hierarchy

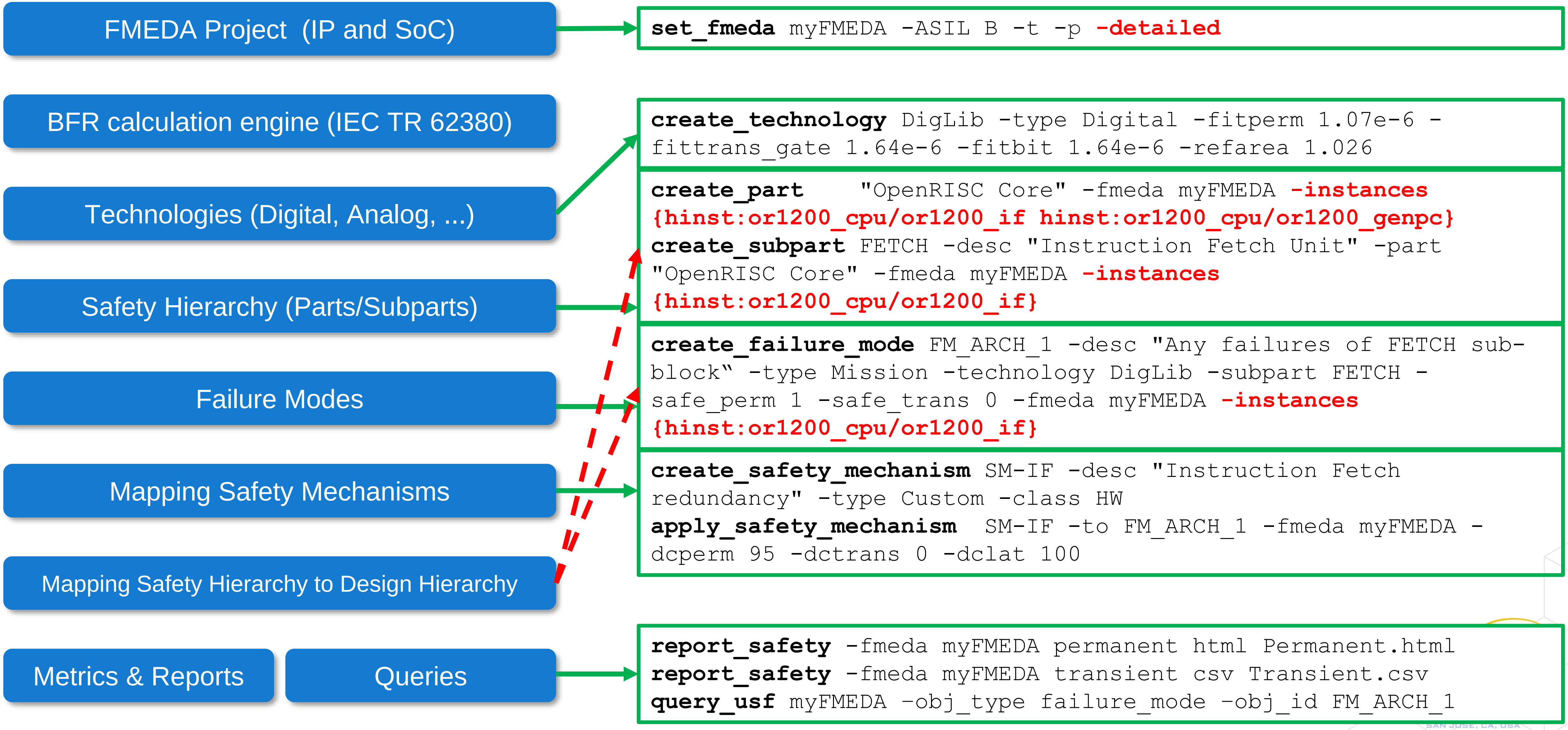
Metrics & Reports

Queries

```
report_safety -fmeda myFMEDA permanent html Permanent.html
report_safety -fmeda myFMEDA transient csv Transient.csv
query_usf myFMEDA -obj_type failure_mode -obj_id FM_ARCH_1
```

Detailed FMEDA

USF



FMEDA Project (IP and SoC)

```
set_fmeda myFMEDA -ASIL B -t -p -detailed
```

BFR calculation engine (IEC TR 62380)

Technologies (Digital, Analog, ...)

```
create_technology DigLib -type Digital -fitperm 1.07e-6 -
fittrans_gate 1.64e-6 -fitbit 1.64e-6 -refarea 1.026
```

Safety Hierarchy (Parts/Subparts)

Mapping Safety Hierarchy to Design Hierarchy

```
create_part    "OpenRISC Core" -fmeda myFMEDA -instances
{hinst:or1200_cpu/or1200_if hinst:or1200_cpu/or1200_genpc}
create_subpart FETCH -desc "Instruction Fetch Unit" -part
"OpenRISC Core" -fmeda myFMEDA -instances
{hinst:or1200_cpu/or1200_if}
```

Failure Modes

```
create_failure_mode FM_ARCH_1 -desc "Any failures of FETCH sub-
block‟ -type Mission -technology DigLib -subpart FETCH -
safe_perm 1 -safe_trans 0 -fmeda myFMEDA -instances
{hinst:or1200_cpu/or1200_if}
```

Mapping Safety Mechanisms

```
create_safety_mechanism SM-IF -desc "Instruction Fetch
redundancy" -type Custom -class HW
apply_safety_mechanism  SM-IF -to FM_ARCH_1 -fmeda myFMEDA -
dcperm 95 -dctrans 0 -dclat 100
```

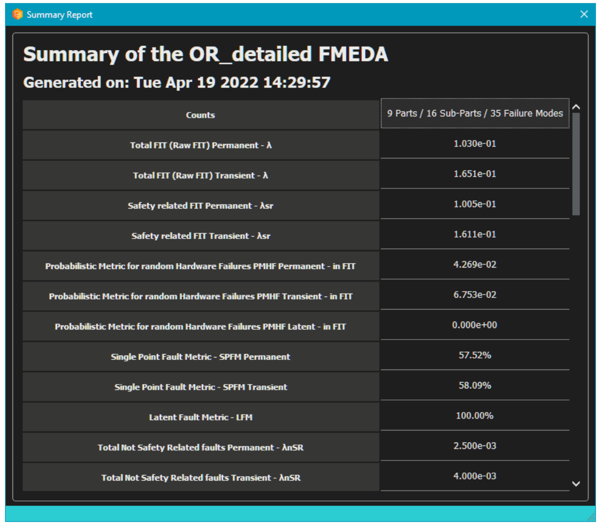
Metrics & Reports

Queries

```
report_safety -fmeda myFMEDA permanent html Permanent.html
report_safety -fmeda myFMEDA transient csv Transient.csv
query_usf myFMEDA -obj_type failure_mode -obj_id FM_ARCH_1
```

Refine FMEDA Data for Optimized Safety Design

Architectural FMEDA



Summary Report

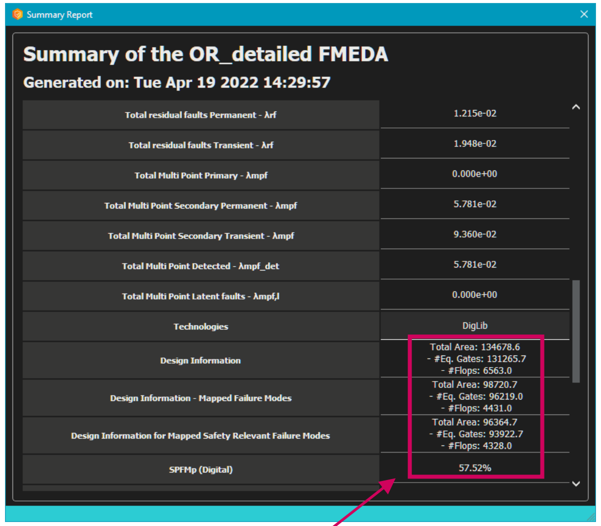
Summary of the OR_detailed FMEDA

Generated on: Tue Apr 19 2022 14:29:57

Counts	9 Parts / 16 Sub-Parts / 35 Failure Modes
Total FIT (Raw FIT) Permanent - λ	1.030e-01
Total FIT (Raw FIT) Transient - λ	1.651e-01
Safety related FIT Permanent - λsr	1.005e-01
Safety related FIT Transient - λsr	1.611e-01
Probabilistic Metric for random Hardware Failures PMHF Permanent - in FIT	4.269e-02
Probabilistic Metric for random Hardware Failures PMHF Transient - in FIT	6.753e-02
Probabilistic Metric for random Hardware Failures PMHF Latent - in FIT	0.000e+00
Single Point Fault Metric - SPFM Permanent	57.52%
Single Point Fault Metric - SPFM Transient	58.09%
Latent Fault Metric - LFM	100.00%
Total Not Safety Related faults Permanent - λnSR	2.500e-03
Total Not Safety Related faults Transient - λnSR	4.000e-03

- No design data available
- FMEDA hierarchy only
- Failure rates and distribution solely based on early estimations

Detailed FMEDA



Summary Report

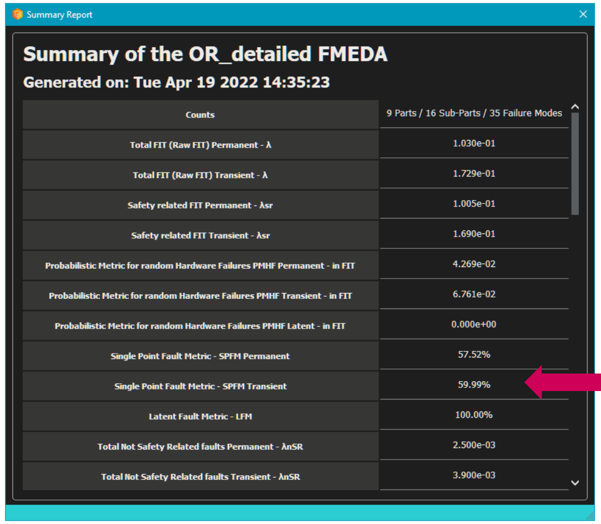
Summary of the OR_detailed FMEDA

Generated on: Tue Apr 19 2022 14:29:57

Total residual faults Permanent - λrf	1.215e-02
Total residual faults Transient - λrf	1.948e-02
Total Multi Point Primary - λmpf	0.000e+00
Total Multi Point Secondary Permanent - λmpf	5.781e-02
Total Multi Point Secondary Transient - λmpf	9.360e-02
Total Multi Point Detected - λmpf_det	5.781e-02
Total Multi Point Latent faults - λmpf,l	0.000e+00
Technologies	DigLib
Design Information	Total Area: 134678.6 - #Eq. Gates: 131265.7 - #Flops: 6563.0
Design Information - Mapped Failure Modes	Total Area: 98720.7 - #Eq. Gates: 96219.0 - #Flops: 4431.0
Design Information for Mapped Safety Relevant Failure Modes	Total Area: 96364.7 - #Eq. Gates: 93922.7 - #Flops: 4328.0
SPFMp (Digital)	57.52%

- With design data
- Design to FMEDA hierarchy mapping
- HW safety metric based on design data

Optimized Safety Design



Summary Report

Summary of the OR_detailed FMEDA

Generated on: Tue Apr 19 2022 14:35:23

Counts	9 Parts / 16 Sub-Parts / 35 Failure Modes
Total FIT (Raw FIT) Permanent - λ	1.030e-01
Total FIT (Raw FIT) Transient - λ	1.729e-01
Safety related FIT Permanent - λsr	1.005e-01
Safety related FIT Transient - λsr	1.690e-01
Probabilistic Metric for random Hardware Failures PMHF Permanent - in FIT	4.269e-02
Probabilistic Metric for random Hardware Failures PMHF Transient - in FIT	6.761e-02
Probabilistic Metric for random Hardware Failures PMHF Latent - in FIT	0.000e+00
Single Point Fault Metric - SPFM Permanent	57.52%
Single Point Fault Metric - SPFM Transient	59.99%
Latent Fault Metric - LFM	100.00%
Total Not Safety Related faults Permanent - λnSR	2.500e-03
Total Not Safety Related faults Transient - λnSR	3.900e-03

2% higher SPFM

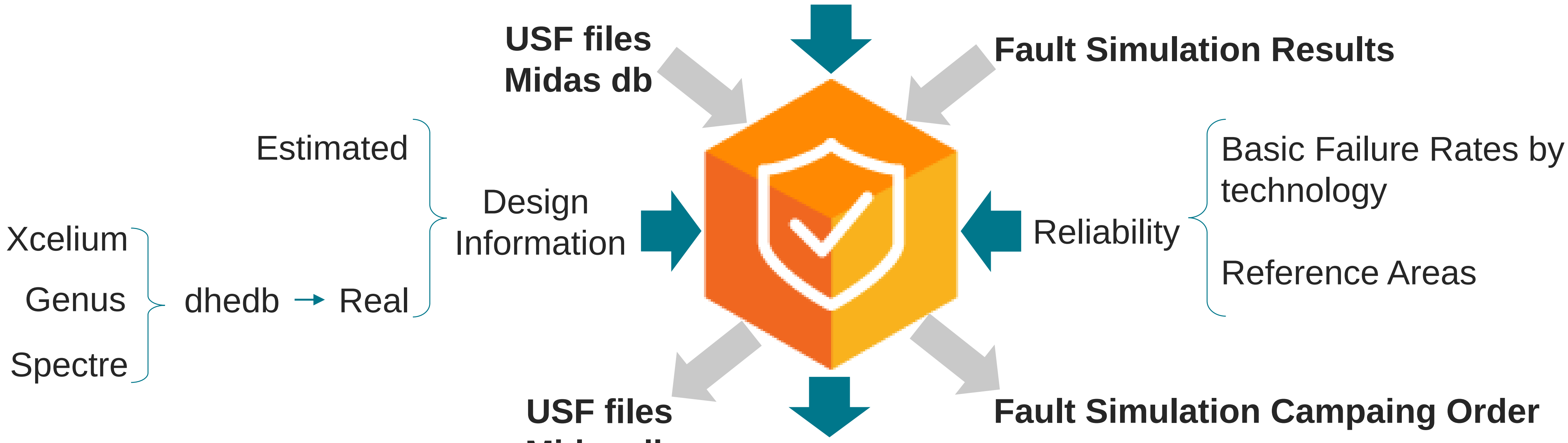
- With design & simulation data
- Design to FMEDA hierarchy mapping
- HW safety metric based on design & simulation data

Optimized FMEDA metric by using design & simulation-based data

Inputs / Outputs

- Definition of the FMEDA Project
- Parts, Subparts, Failure Modes, Safety Mechanism
- Design Mapping (for a Detailed FMEDA)
- Excel files

FMEDA Authoring



USF files
Midas db

Fault Simulation Results

Design Information

Estimated

Xcelium
Genus
Spectre

dhedb → Real

Reliability

Basic Failure Rates by technology

Reference Areas

USF files
Midas db

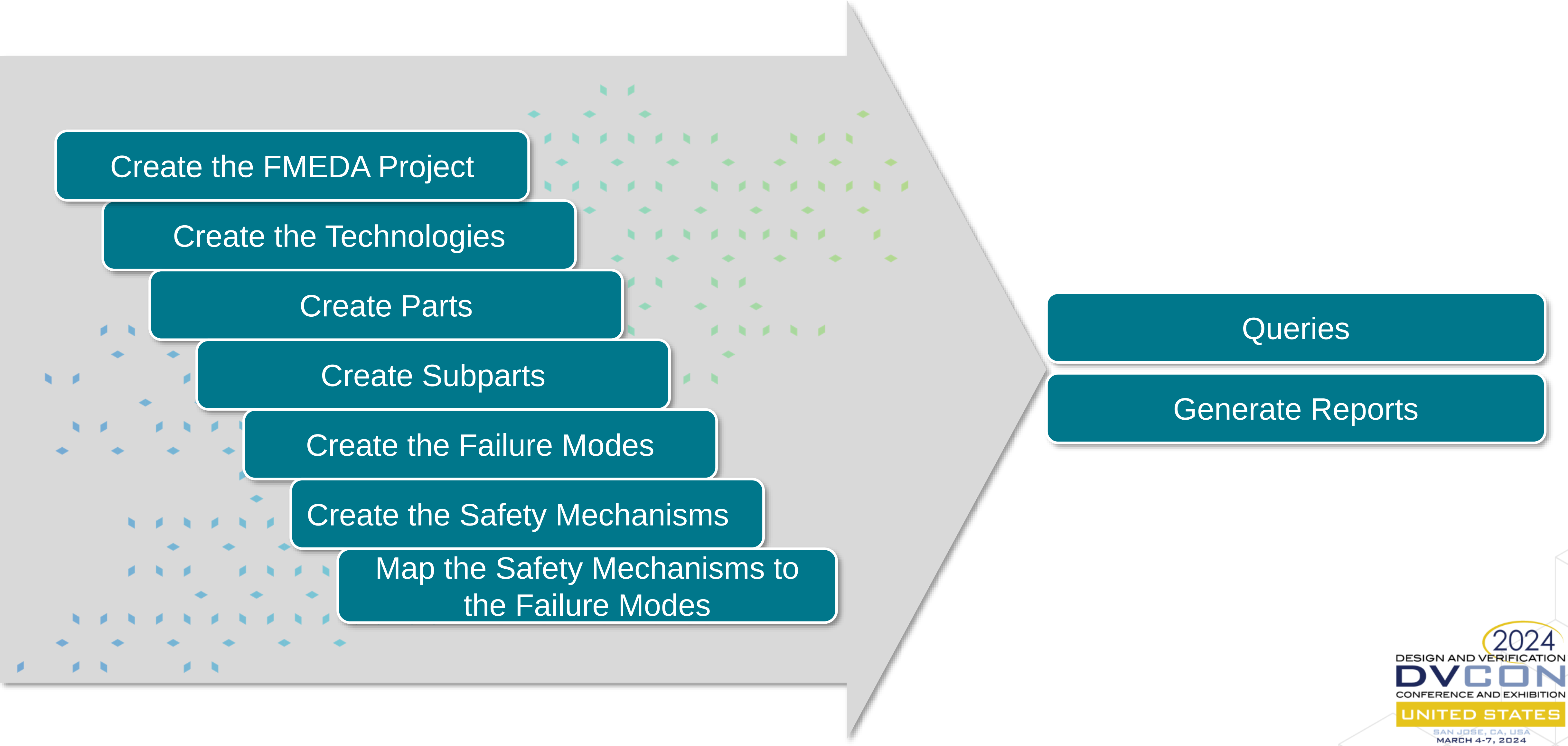
Fault Simulation Campaing Order

Reports

- FMEDA (Permanent+Transient)
- Summary
- SoC Summary
- Safety Goal Report

Architectural FMEDA

Architectural FMEDA Authoring Steps



Create the FMEDA Project

Create the Technologies

Create Parts

Create Subparts

Create the Failure Modes

Create the Safety Mechanisms

Map the Safety Mechanisms to the Failure Modes

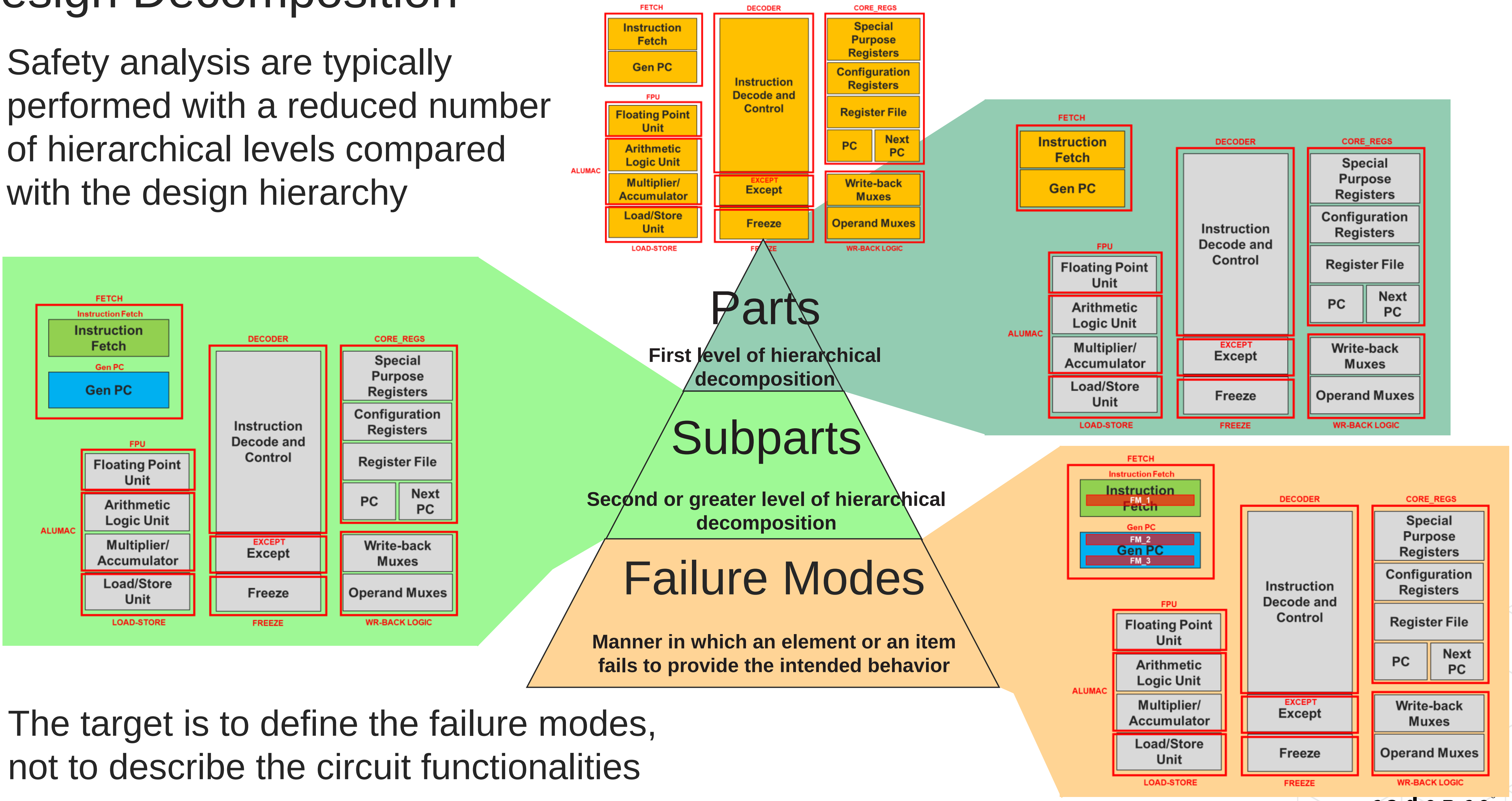
Queries

Generate Reports

Design Decomposition

- Safety analysis are typically performed with a reduced number of hierarchical levels compared with the design hierarchy

FMEDA Project



Parts

First level of hierarchical decomposition

Subparts

Second or greater level of hierarchical decomposition

Failure Modes

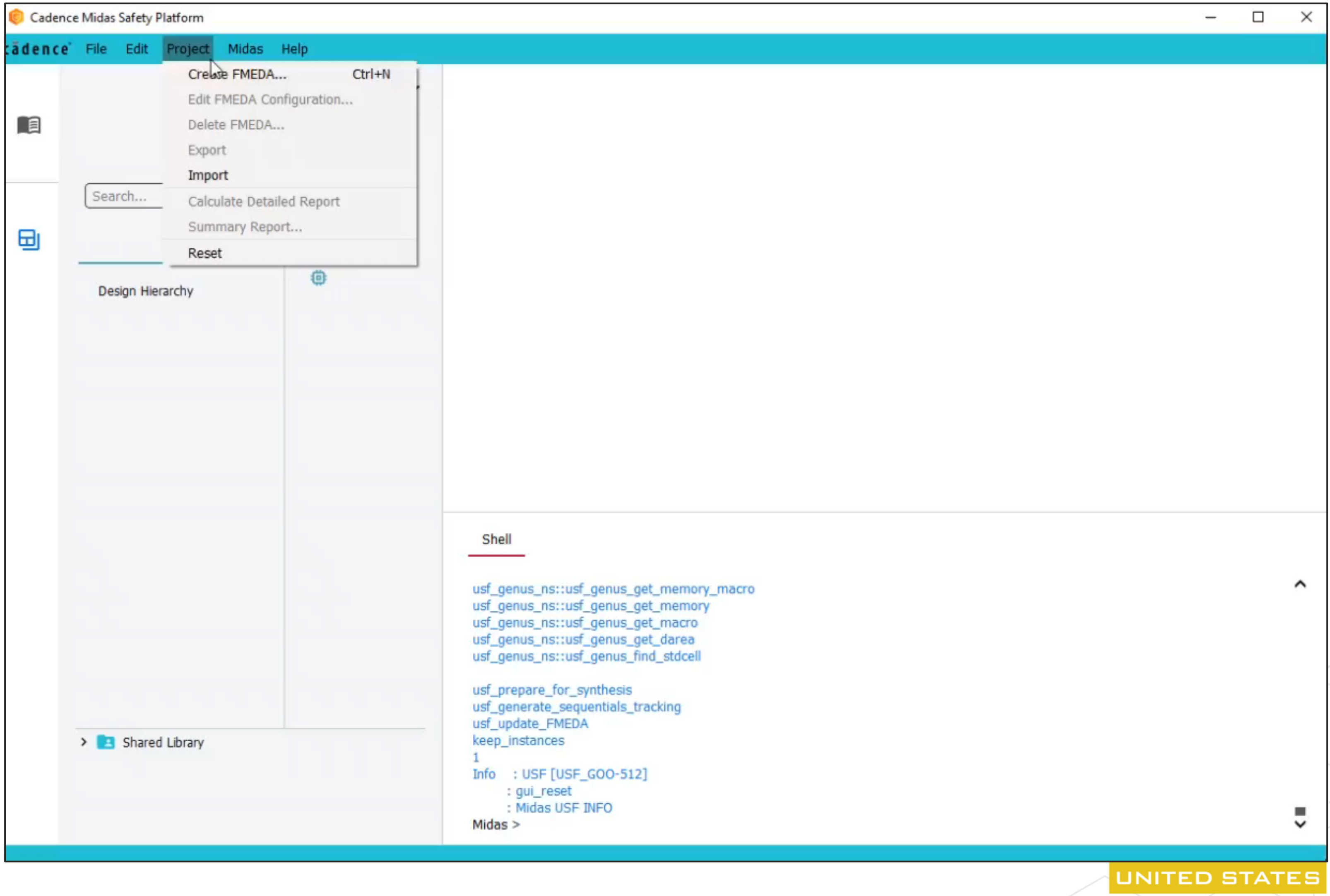
Manner in which an element or an item fails to provide the intended behavior

- The target is to define the failure modes, not to describe the circuit functionalities

cadence

Functional Safety Authoring

- The GUI provides an user-friendly FMEDA authoring enviroment
- Safety objects can also be created with USF commands

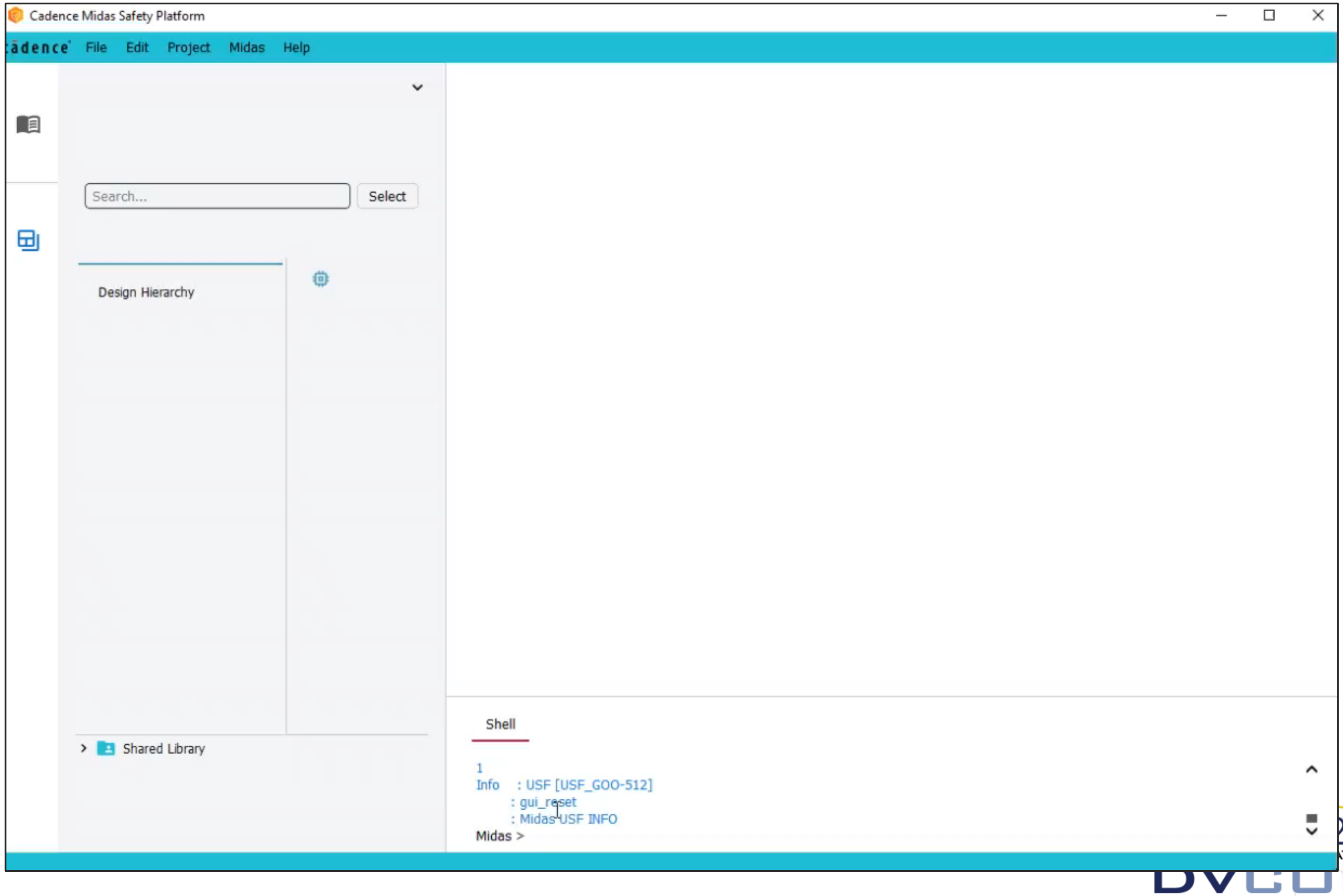


```
usf_genus_ns::usf_genus_get_memory_macro
usf_genus_ns::usf_genus_get_memory
usf_genus_ns::usf_genus_get_macro
usf_genus_ns::usf_genus_get_darea
usf_genus_ns::usf_genus_find_stdcell

usf_prepare_for_synthesis
usf_generate_sequentials_tracking
usf_update_FMEDA
keep_instances
1
Info    : USF [USF_GOO-512]
        : gui_reset
        : Midas USF INFO
Midas >
```

UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

Functional Safety Authoring

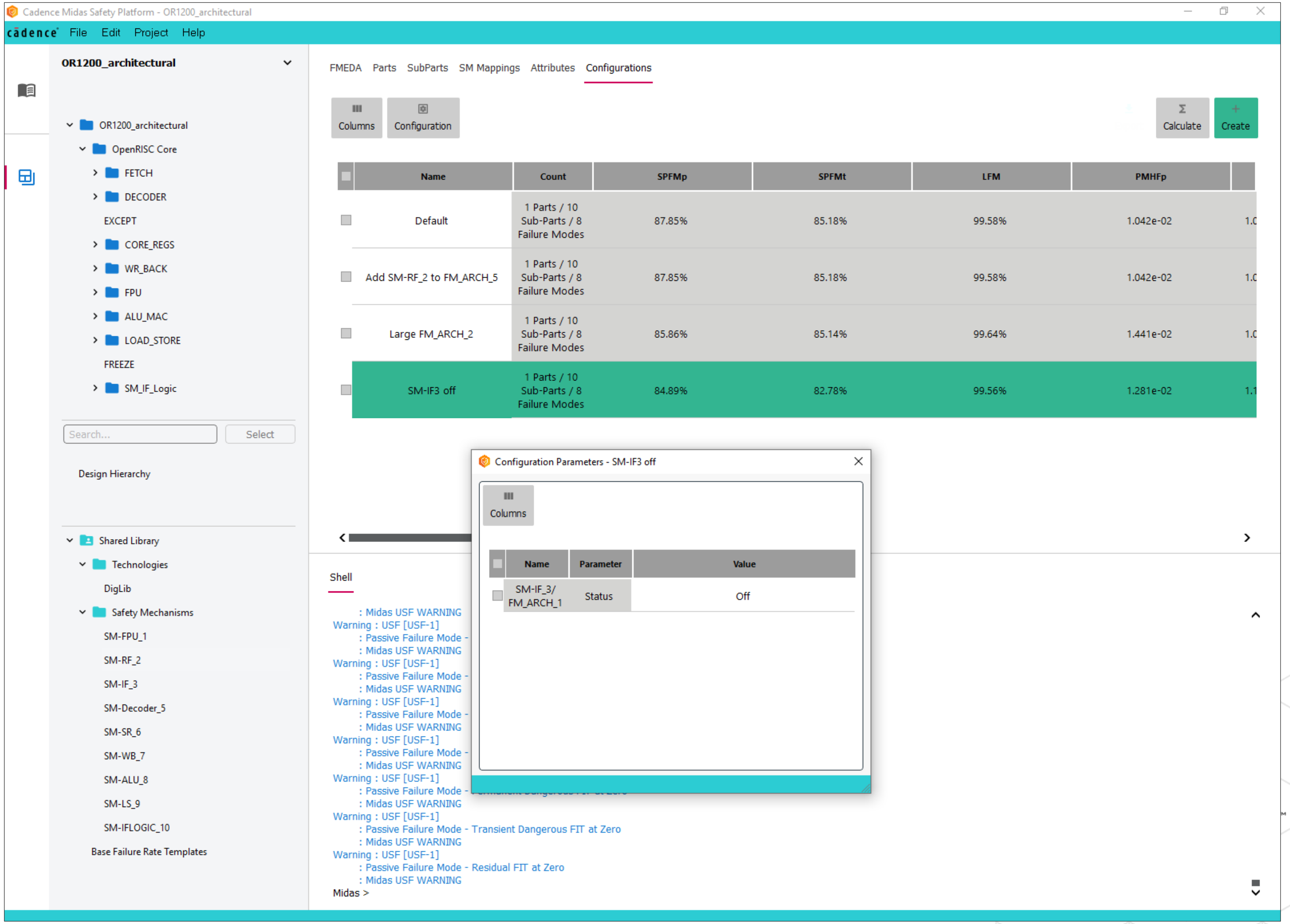
- The solution is fully scriptable
- Mixing GUI and scripted-automations is further possibile



```
1
Info    : USF [USF_GOO-512]
       : gui_reset
       : Midas USF INFO
Midas >
```

What-if Analysis: FMEDA Static Configurations

- Create configurations changing values in the FMEDA (e.g., design info., SM DCs)
- Each configuration generates safety metrics to be compared
- The configurations can be saved and restored



□	Name	Count	SPFMp	SPFMt	LFM	PMHFp	
□	Default	1 Parts / 10 Sub-Parts / 8 Failure Modes	87.85%	85.18%	99.58%	1.042e-02	1.0
□	Add SM-RF_2 to FM_ARCH_5	1 Parts / 10 Sub-Parts / 8 Failure Modes	87.85%	85.18%	99.58%	1.042e-02	1.0
□	Large FM_ARCH_2	1 Parts / 10 Sub-Parts / 8 Failure Modes	85.86%	85.14%	99.64%	1.441e-02	1.0
□	SM-IF3 off	1 Parts / 10 Sub-Parts / 8 Failure Modes	84.89%	82.78%	99.56%	1.281e-02	1.1

Configuration Parameters - SM-IF3 off

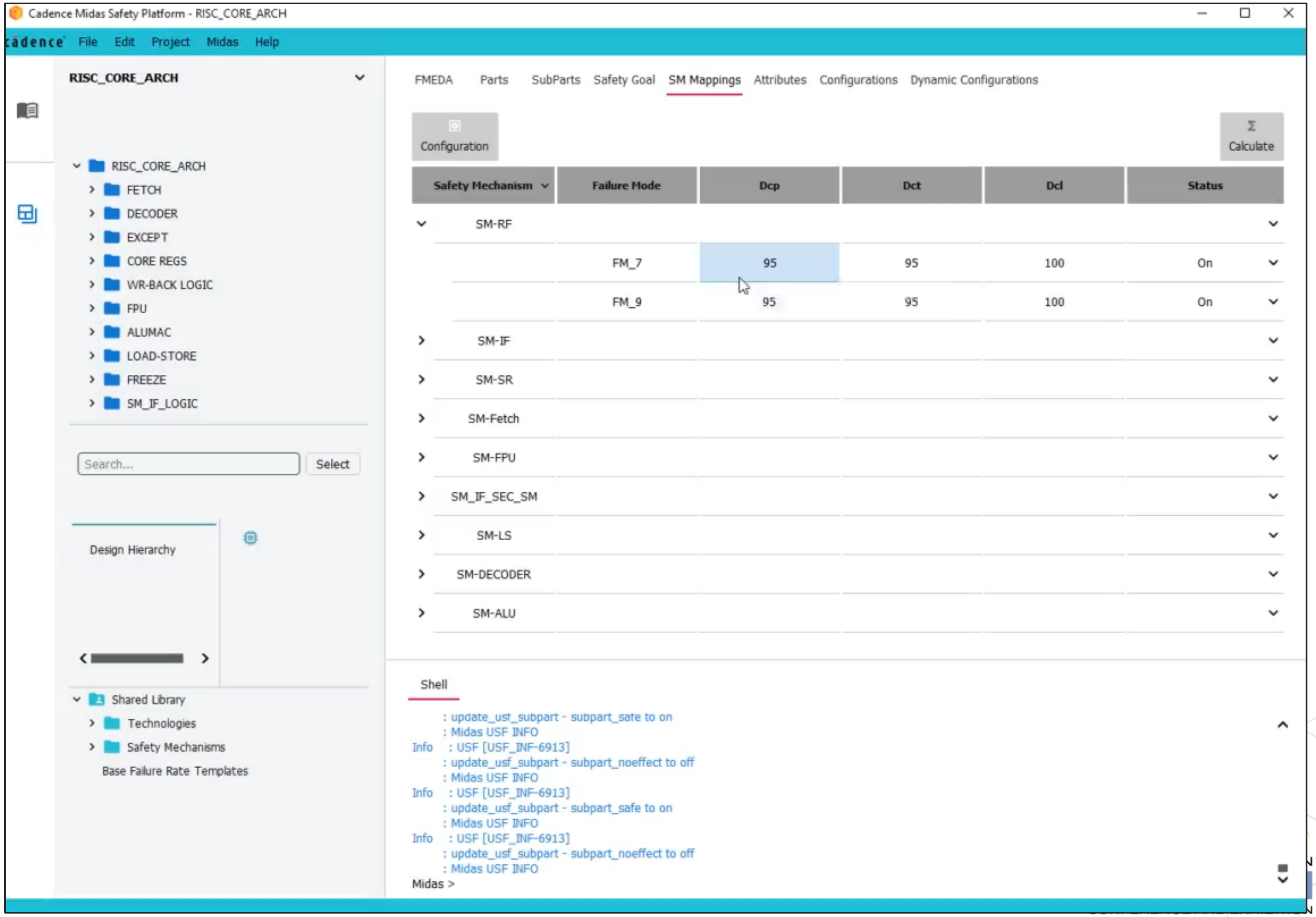
Columns

□	Name	Parameter	Value
□	SM-IF_3/ FM_ARCH_1	Status	Off

```
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode -
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode -
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode -
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode -
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode -
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode - Transient Dangerous FIT at Zero
        : Midas USF WARNING
Warning : USF [USF-1]
        : Passive Failure Mode - Residual FIT at Zero
        : Midas USF WARNING
Midas >
```

What-if Analysis: FMEDA Dynamic Configurations

- It is possibile to select one parameter (e.g, DC), define the interval and an output metric to be reported
- By leveraging the USF backend Midas provides the result of the simulation
- Graphs, and values, can be saved and restored



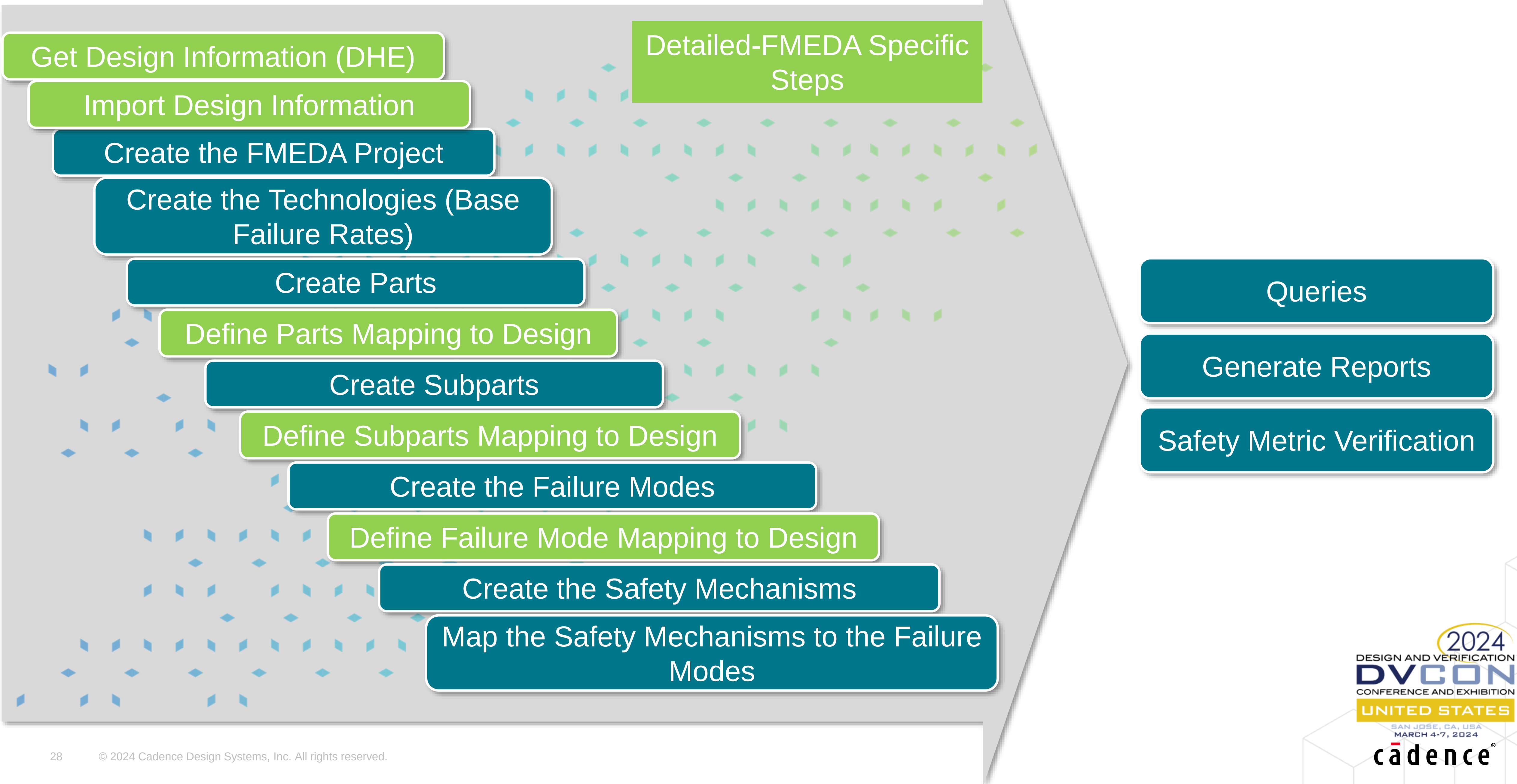
Safety Mechanism	Failure Mode	Dcp	Dct	Dcl	Status
SM-RF					
	FM_7	95	95	100	On
	FM_9	95	95	100	On
SM-IF					
SM-SR					
SM-Fetch					
SM-FPU					
SM_IF_SEC_SM					
SM-LS					
SM-DECODER					
SM-ALU					

```
      : update_usf_subpart - subpart_safe to on
      : Midas USF INFO
Info   : USF [USF_INF-6913]
      : update_usf_subpart - subpart_noeffect to off
      : Midas USF INFO
Info   : USF [USF_INF-6913]
      : update_usf_subpart - subpart_safe to on
      : Midas USF INFO
Info   : USF [USF_INF-6913]
      : update_usf_subpart - subpart_noeffect to off
      : Midas USF INFO
Midas >
```

UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

Detailed FMEDA

Detailed FMEDA Authoring Steps



Detailed-FMEDA Specific Steps

- Get Design Information (DHE)
- Import Design Information
- Create the FMEDA Project
- Create the Technologies (Base Failure Rates)
- Create Parts
- Define Parts Mapping to Design
- Create Subparts
- Define Subparts Mapping to Design
- Create the Failure Modes
- Define Failure Mode Mapping to Design
- Create the Safety Mechanisms
- Map the Safety Mechanisms to the Failure Modes

- Queries
- Generate Reports
- Safety Metric Verification

Design Hierarchy Extraction - Genus

```
usf_genus_ns::usf_genus_dhe
      [designInstance]
      {dheFileName}
      {ffFileName}
      [-bbox bboxFileName]
      [-seq_leaf  {instances_name_list}
       -comb_leaf {instances_name_list} [-stopathier]]
```

`designInstance`	Hierarchical design instance to collect design information. If no instance is passed, the current design is assumed to be extracted
`dheFileName`	Filename to store design Hierarchy Information
`ffFileName`	Filename to store the information for each hierarchical instance
`-bbox bboxFileName`	If this option is used, the command will try to find all the memories and the macros in the design and to generate automatically a description file

`[-seq_leaf {instances_name_list}`
`-comb_leaf {instances_name_list} [-stopathier]]`
	Support the extraction of leaf instances

- The generated database can be parsed with the **`usf_dhe_parser`** command
- A Midas database can be generated by using the **`save_usf`** `-db` command

Design Hierarchy Extraction - Xcelium

```
xrun -elaborate
      -fault_mdb_gen
     [-fault_top <top_instance | top_module>]
     [-fault_mdb_file          <dheDB_filename>]
     [-fault_mdb_ff]
     [-fault_lib_mfile         <lib_list_file>]
     [-fault_mdb_overwrite]
     [other_options]
     <source_files>
```

`-fault_top`	Specifies the top_instance or top_module for design information extraction.
`-fault_mdb_gen`	Enables design extraction and generates a Midas database file.
`-fault_mdb_file`	Name of the Midas database file
`-fault_mdb_ff`	Includes sequential element extraction (pinout and flip-flop information) in the generated Midas database file.
`-fault_lib_mfile`	Specifies a liberty file list for gate-level design.
`-fault_mdb_overwrite`	Overwrites a previously generated Midas database file, if it exists

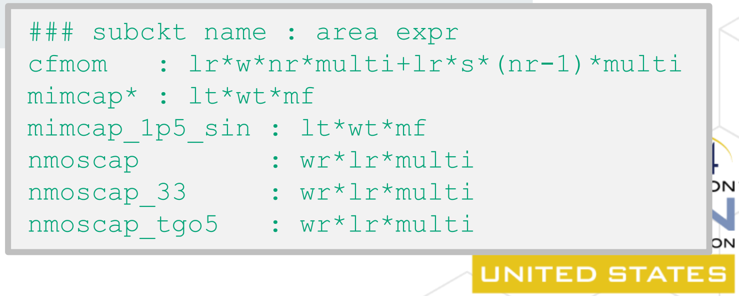
- For macros, read the liberty files into the Xcelium elaboration
  - Area is extracted if the `-macro_cell` option is used when reading the relevant `.lib` files and the macro are elaborated as a library using `-v`
- Import the generated database into Midas or parse using the **usf_dhe_parser** `-db` command

Design Hierarchy Extraction - Spectre

- Spectre Circuit Information (info)
    - New keyword: `what=dhe`

- DHE Options

Parameter	Description
`dheminarea`	Lower bound of area value for device to be considered during design hierarchy extraction
`dhesubckt`	Design hierarchy is generated for all instances of the specified sub-circuits
`dheinst`	Design hierarchy is generated for the specified sub-circuit instances
`dhexsubckt`	All instances of the specified sub-circuits are excluded from the design hierarchy
`dhexinst`	The specified sub-circuit instances are excluded from the design hierarchy
`dheparams`	Name of the file that provides the rules to calculate area for subcircuits when what=dhe. Area are calculated on instance parameters



```
### subckt name : area expr
cfmom   : lr*w*nr*multi+lr*s*(nr-1)*multi
mimcap* : lt*wt*mf
mimcap_1p5_sin : lt*wt*mf
nmoscap        : wr*lr*multi
nmoscap_33     : wr*lr*multi
nmoscap_tgo5   : wr*lr*multi
```

- Import the generated database into Midas or parse using the **`usf_dhe_parser`** `-db` command

Basic Failure Rate (BFR) Support

IEC TR 62380: USF Commands

MATHEMATICAL MODEL :

$$\lambda = \left(\underbrace{\left\{ \overbrace{\lambda_1 \times N \times e^{-0.35\times a} + \lambda_2}^{\texttt{set\_bfr}} \right\} \times \left\{ \frac{\sum_{i=1}^{y} (\pi_t)_i \times \tau_i}{\tau_{on} + \tau_{off}} \right\}}_{\lambda_{die}} + \left\{ \underbrace{2.75\times 10^{-3} \times \pi_\alpha \times \left(\sum_{i=1}^{z} (\pi_n)_i \times (\Delta T_i)^{0.68} \right) \times \lambda_3}_{\lambda_{package}} \right\} + \left\{ \underbrace{\pi_I \times \lambda_{EOS}}_{\lambda_{overstress}} \right\} \right) \times 10^{-9} / h$$

die — `set_IEC62380_lDIE`

package — `set_IEC62380_lPackage`

overstress — `set_IEC62380_loverstress`

- Customizations:
  - Mission Profile: `set_Mission_Profile`; `get_Mission_Profile`
  - Safe/Dangerous Ratio : `set_safeness; get_safeness`
  - Confidence Level: `set_Confidence; get_confidence`
  - Conservative (ISO26262-11) temperature derating
  - Package customizations: `set_IEC62380_cpackage; get_IEC62380_cpackage`

SN29500: USF Command

```
set_SN29500_lambda {lambda ref table} {technology} {size} {factors list} {power consumption}
                   {number of pins} {cooling method} {package} {mission profile}
                   {table 9 category} {voltage type} {operating voltage} {rated voltage}
                   {drift flag}
```

$$\lambda = \lambda_{ref} \times \pi_u \times \pi_T \times \pi_D$$

Analog integrated circuits with extended range of operating voltages

$$\lambda = \lambda_{ref} \times \pi_T \times \pi_D$$

Analog integrated circuits with fixed operating voltages

Drift sensitivity factor

$$\lambda = \lambda_{ref} \times \pi_U \times \pi_T$$

Digital CMOS-B

Temperature sensitivity factor

$$\lambda = \lambda_{ref} \times \pi_T$$

Voltage Dependence factor

For all other integrated circuits

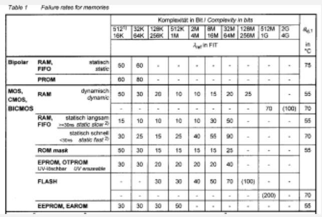


Table 1 *Failure rates for memories*

			512 [1] 16K	32K 64K	128K 256K	512K 1M	2M 4M	8M 16M	32M 64M	128M 256M	512M 1G	2G 4G	$\vartheta_{s,1}$ in °C
			Komplexität in Bit / *Complexity in bits*										
			λ_{ref} in FIT										
Bipolar	**RAM, FIFO**	statisch *static*	60	60	-	-	-	-	-	-	-	-	75
	PROM		60	80	-	-	-	-	-	-	-	-	
MOS, CMOS, BICMOS	**RAM**	dynamisch *dynamic*	50	30	20	10	10	15	20	25		-	55
			-	-	-	-	-	-	-	-	70	(100)	70
	RAM, FIFO	statisch langsam >=30ns *static slow* [2]	15	10	10	10	10	30	50	-	-	-	55
		statisch schnell <30ns *static fast* [2]	30	25	15	25	40	55	90	-	-	-	70
	ROM mask		50	30	15	15	15	15	25	-	-	-	55
	EPROM, OTPROM UV-löschbar *UV erasable*		30	30	20	20	20	20	40	-	-	-	
	FLASH		-	-	30	30	40	50	70	(100)	-	-	
			-	-	-	-	-	-	-	-	(200)	-	70
	EEPROM, EAROM		30	30	30	50	-	-	-	-	-	-	55

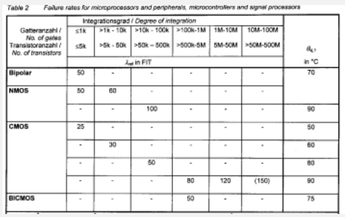


Table 2 *Failure rates for microprocessors and peripherals, microcontrollers and signal processors*

Gatteranzahl / *No. of gates* Transistoranzahl / *No. of transistors*	≤1k ≤5k	>1k - 10k >5k - 50k	>10k - 100k >50k – 500k	>100k-1M >500k-5M	1M-10M 5M-50M	10M-100M >50M-500M	$\vartheta_{s,1}$ in °C
	Integrationsgrad / *Degree of integration*						
	λ_{ref} in FIT						
Bipolar	50	-	-	-	-	-	70
NMOS	50	60	-	-	-	-	
	-	-	100	-	-	-	90
CMOS	25	-	-	-	-	-	50
	-	30	-	-	-	-	60
	-	-	50	-	-	-	80
	-	-	-	80	120	(150)	90
BICMOS	-	-	-	50	-	-	75

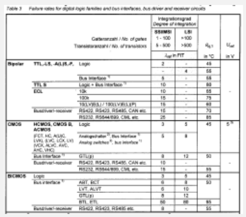


Table 3 *Failure rates for digital logic families and bus interfaces, bus driver and receiver circuits*

		Gatteranzahl / *No. of gates* Transistoranzahl / *No. of transistors*	**SSI/MSI** 1 - 100 5 - 500	**LSI** >100 >500	$\vartheta_{s,1}$ in °C	U_{ref} in V
			Integrationsgrad *Degree of integration*			
			λ_{ref} in FIT			
Bipolar	**TTL,-LS,-A(L)S,-F,**	Logic	2	-	45	-
			-	4	55	
		Bus interface [1]	5	-	55	
	TTL S	Logic + bus interface [1]	10	-	60	
	ECL	10k	10	-	55	
		100k	15	-	75	
		10(LV)E(L) / 100(LV)E(L)(P)	15	-	60	
	Busdriver/-receiver	RS422, RS423, RS485, CAN etc.	15	-	70	
		RS232, RS644/899, CML etc.	25	-	85	
CMOS	**HCMOS, CMOS B, ACMOS**	Logic	3	5	45	5 [3]
	(FCT, HC, A(L)C, LV(X), (L)VC, LCX, LV) (VCX, ALVC, AVC, AHC, VHC)	Analogschalter [2], Bus interface [1] *Analog switches* [2], *bus interface* [1]	5	8		
	Bus interface [1]	GTL(p)	8	12	50	-
	Busdriver/-receiver	RS422, RS423, RS485, CAN etc.	10	-		
		RS232, RS644/899, CML etc.	15	-	65	
BICMOS	Logic		3	5	45	-
	Bus interface [1]	ABT, BCT	6	8	50	
		LVT, ALVT	6	10		
		GTL(p)	8	12		
		RTL, FTL	60	60	85	
	Busdriver/-receiver	RS422, RS423, RS485 etc.	8	-	55	

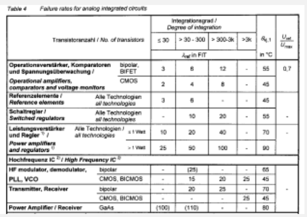


Table 4 *Failure rates for analog integrated circuits*

Transistoranzahl / *No. of transistors*			≤ 30	> 30 - 300	> 300-3k	>3k	$\vartheta_{s,1}$ in °C	$\frac{U_{ref}}{U_{max}}$
			Integrationsgrad / *Degree of integration*					
			λ_{ref} in FIT					
Operationsverstärker, Komparatoren und Spannungsüberwachung /	bipolar, BIFET		3	6	12	-	55	0,7
Operational amplifiers, comparators and voltage monitors	CMOS		2	4	8	-	45	
Referenzelemente / ***Reference elements***	Alle Technologien *all technologies*		3	6	-	-	45	-
Schaltregler / ***Switched regulators***	Alle Technologien *all technologies*		-	10	20	-	55	
Leistungsverstärker und Regler [1] **/**	Alle Technologien / *all technologies*	≤ 1 Watt	10	20	40	-	70	-
Power amplifiers and regulators [1]		> 1 Watt	25	50	100	-	90	
Hochfrequenz IC [2] **/** ***High Frequency IC*** [2]								
HF modulator, demodulator,	bipolar		-	(25)	-	-	65	-
PLL, VCO	CMOS, BICMOS		-	15	20	25	45	
Transmitter, Receiver	bipolar		-	20	25	-	70	
	CMOS, BICMOS		-	-	-	25	45	
Power Amplifier / Receiver	GaAs		(100)	(110)	-	-	80	

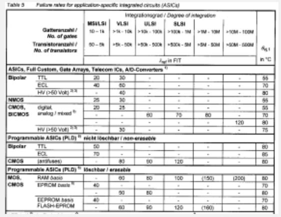
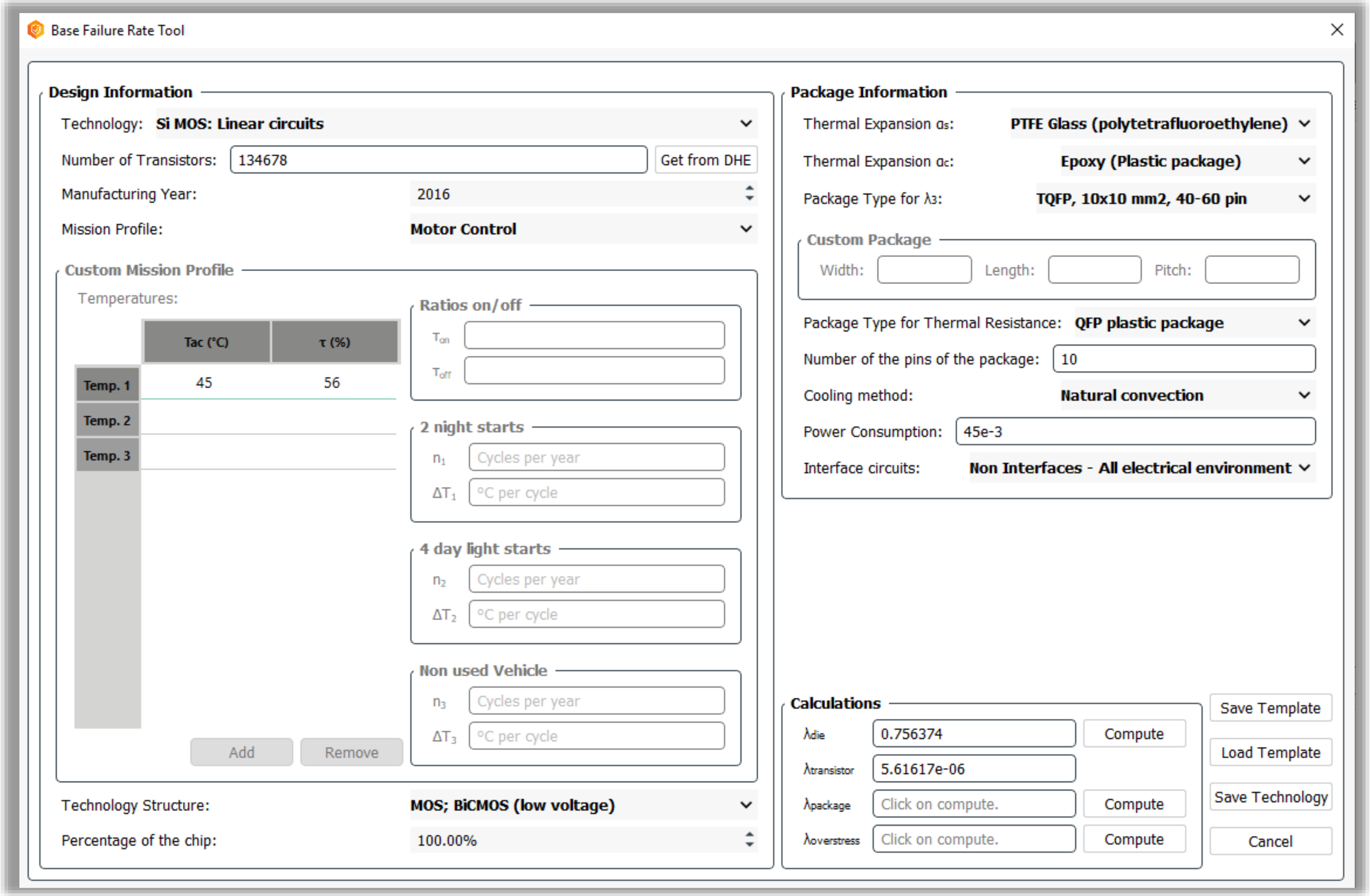


Table 5 *Failure rates for application-specific integrated circuits (ASICs)*

	Gatteranzahl / ***No. of gates*** **Transistoranzahl /** ***No. of transistors***	**MSI/LSI** 10 – 1k 50 – 5k	**VLSI** >1k - 10k >5k - 50k	**ULSI** >10k - 100k >50k - 500k	**SLSI** >100k - 1M >500k - 5M	 >1M - 10M >5M - 50M	 >10M - 100M >50M - 500M	$\vartheta_{s,1}$ in °C
		Integrationsgrad / Degree of integration						
		λ_{ref} in FIT						
ASICs, Full Custom, Gate Arrays, Telecom ICs, A/D-Converters [1]								
Bipolar	TTL	20	30	-	-	-	-	55
	ECL	40	60	-	-	-	-	70
	HV (>50 Volt) [2] [3]	-	40	-	-	-	-	80
NMOS		25	30	-	-	-	-	55
CMOS, BICMOS	digital, analog / mixed [4]	20	25	-	-	-	-	55
		-	-	60	70	80	-	70
		-	-	-	-	-	120	80
	HV (>50 Volt) [2] [3]	-	30	-	-	-	-	75
Programmable ASICs (PLD) [5] **nicht löschbar / non-erasable**								
Bipolar	TTL	50	-	-	-	-	-	80
	ECL	70	-	-	-	-	-	85
CMOS	(antifuses)	-	80	90	120	-	-	80
Programmable ASICs (PLD) [5] **löschbar / erasable**								
MOS, CMOS	RAM *basis*	-	60	80	100	(150)	(200)	80
	EPROM *basis* [6]	40	-	-	-	-	-	70
		-	50	80	-	-	-	80
	EEPROM *basis* FLASH-EPROM	40	-	-	-	-	-	70
		-	60	90	120	(160)	-	80

Midas GUI BFR Tools

IECTR 62380



Design Information

Technology: **Si MOS: Linear circuits**

Number of Transistors: 134678

Manufacturing Year: 2016

Mission Profile: **Motor Control**

Custom Mission Profile

Temperatures:

	Tac (°C)	τ (%)
Temp. 1	45	56
Temp. 2		
Temp. 3		

Ratios on/off

T_{on}:

T_{off}:

2 night starts

n_1: Cycles per year

ΔT_1: °C per cycle

4 day light starts

n_2: Cycles per year

ΔT_2: °C per cycle

Non used Vehicle

n_3: Cycles per year

ΔT_3: °C per cycle

Technology Structure: **MOS; BiCMOS (low voltage)**

Percentage of the chip: 100.00%

Package Information

Thermal Expansion αs: **PTFE Glass (polytetrafluoroethylene)**

Thermal Expansion αc: **Epoxy (Plastic package)**

Package Type for λ3: **TQFP, 10x10 mm2, 40-60 pin**

Custom Package

Width: Length: Pitch:

Package Type for Thermal Resistance: **QFP plastic package**

Number of the pins of the package: 10

Cooling method: **Natural convection**

Power Consumption: 45e-3

Interface circuits: **Non Interfaces - All electrical environment**

Calculations

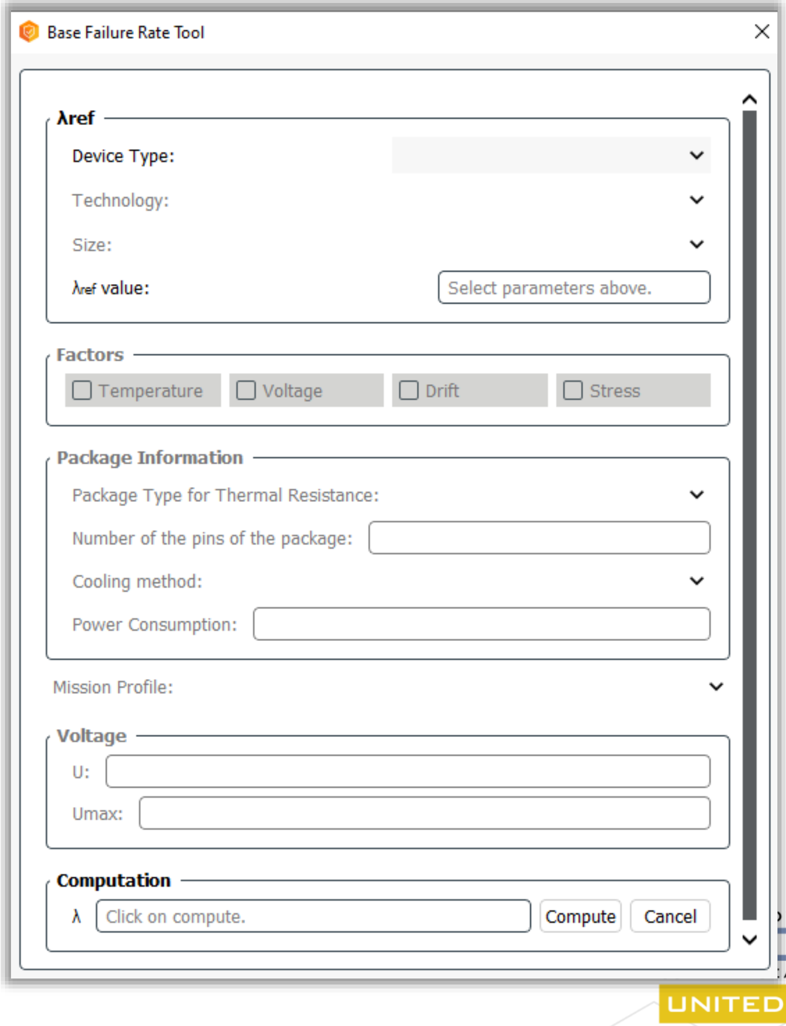
λ_{die}: 0.756374

$\lambda_{transistor}$: 5.61617e-06

$\lambda_{package}$: Click on compute.

$\lambda_{overstress}$: Click on compute.

SN29500



λref

Device Type:

Technology:

Size:

λ_{ref} value: Select parameters above.

Factors

☐ Temperature ☐ Voltage ☐ Drift ☐ Stress

Package Information

Package Type for Thermal Resistance:

Number of the pins of the package:

Cooling method:

Power Consumption:

Mission Profile:

Voltage

U:

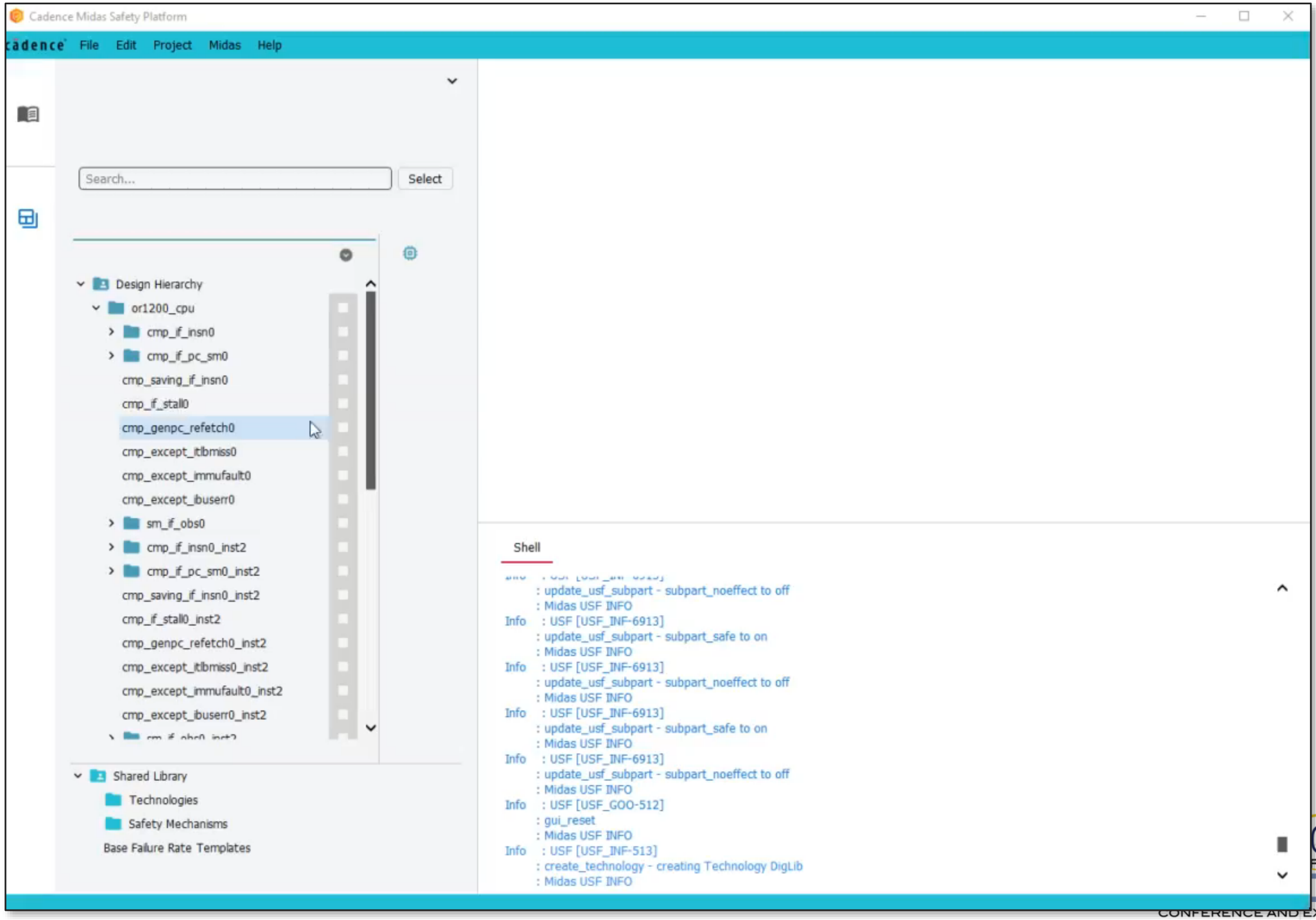
Umax:

Computation

λ: Click on compute.

Leverage Design Information in the BFR Computation

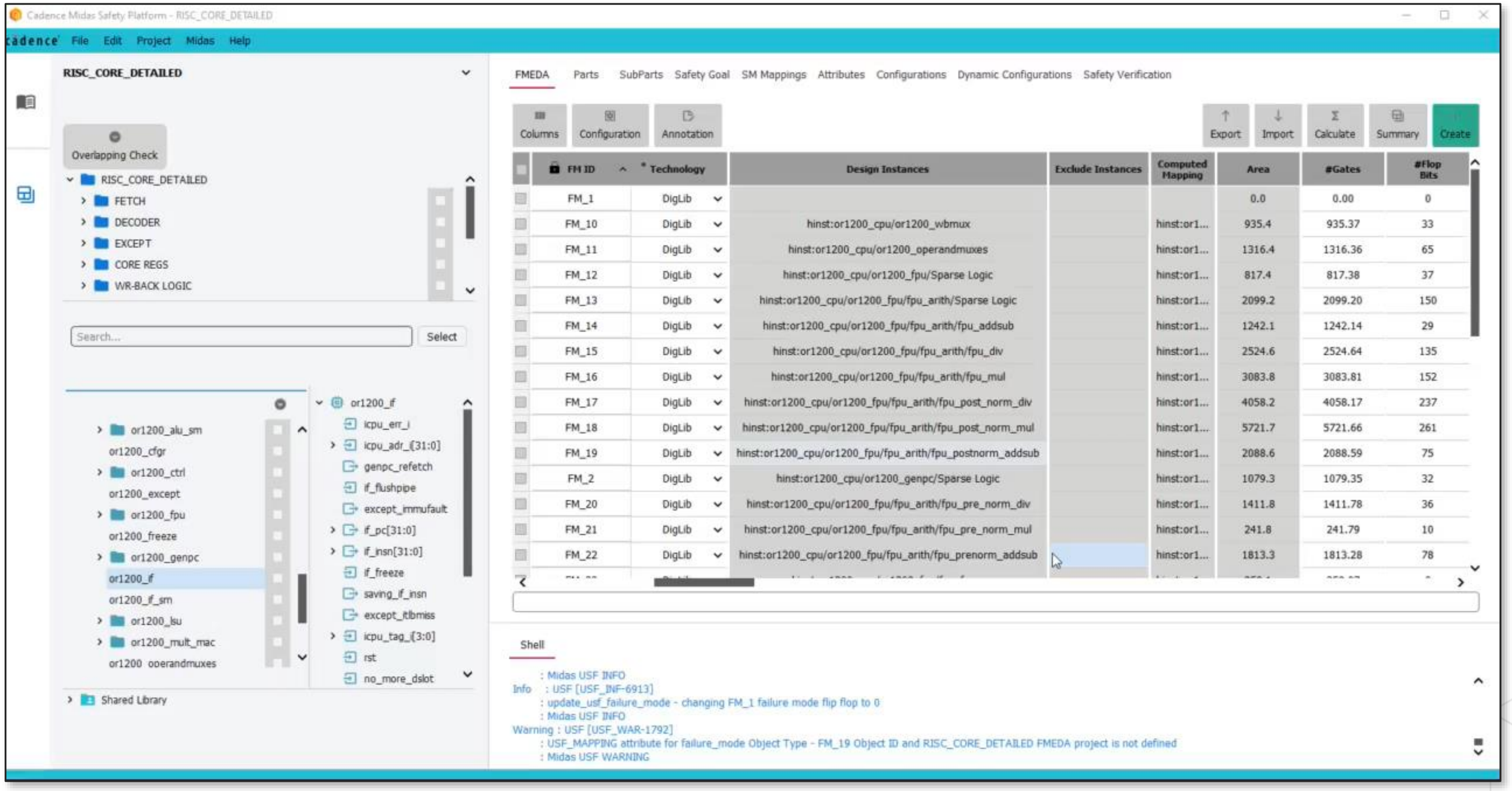
- Create a Technology by using the IEC 62380 BFR tool with automatic computation of the number of transistors
- The technology is saved in the shared library, available for all FMEDA projects



```
         : update_usf_subpart - subpart_noeffect to off
         : Midas USF INFO
Info     : USF [USF_INF-6913]
         : update_usf_subpart - subpart_safe to on
         : Midas USF INFO
Info     : USF [USF_INF-6913]
         : update_usf_subpart - subpart_noeffect to off
         : Midas USF INFO
Info     : USF [USF_INF-6913]
         : update_usf_subpart - subpart_safe to on
         : Midas USF INFO
Info     : USF [USF_INF-6913]
         : update_usf_subpart - subpart_noeffect to off
         : Midas USF INFO
Info     : USF [USF_GOO-512]
         : gui_reset
         : Midas USF INFO
Info     : USF [USF_INF-513]
         : create_technology - creating Technology DigLib
         : Midas USF INFO
```

Design Information Mapping

- Drag & drop Design information to Parts, Subparts and Failure Modes
- Area, equivalent number of gates and number of sequential elements are automatically computed



FM ID	Technology	Design Instances	Exclude Instances	Computed Mapping	Area	#Gates	#Flop Bits
FM_1	DigLib				0.0	0.00	0
FM_10	DigLib	hinst:or1200_cpu/or1200_wbmux		hinst:or1...	935.4	935.37	33
FM_11	DigLib	hinst:or1200_cpu/or1200_operandmuxes		hinst:or1...	1316.4	1316.36	65
FM_12	DigLib	hinst:or1200_cpu/or1200_fpu/Sparse Logic		hinst:or1...	817.4	817.38	37
FM_13	DigLib	hinst:or1200_cpu/or1200_fpu/fpu_arith/Sparse Logic		hinst:or1...	2099.2	2099.20	150
FM_14	DigLib	hinst:or1200_cpu/or1200_fpu/fpu_arith/fpu_addsub		hinst:or1...	1242.1	1242.14	29
FM_15	DigLib	hinst:or1200_cpu/or1200_fpu/fpu_arith/fpu_div		hinst:or1...	2524.6	2524.64	135
FM_16	DigLib	hinst:or1200_cpu/or1200_fpu/fpu_arith/fpu_mul		hinst:or1...	3083.8	3083.81	152
FM_17	DigLib	hinst:or1200_cpu/or1200_fpu/fpu_arith/fpu_post_norm_div		hinst:or1...	4058.2	4058.17	237
FM_18	DigLib	hinst:or1200_cpu/or1200_fpu/fpu_arith/fpu_post_norm_mul		hinst:or1...	5721.7	5721.66	261
FM_19	DigLib	hinst:or1200_cpu/or1200_fpu/fpu_arith/fpu_postnorm_addsub		hinst:or1...	2088.6	2088.59	75
FM_2	DigLib	hinst:or1200_cpu/or1200_genpc/Sparse Logic		hinst:or1...	1079.3	1079.35	32
FM_20	DigLib	hinst:or1200_cpu/or1200_fpu/fpu_arith/fpu_pre_norm_div		hinst:or1...	1411.8	1411.78	36
FM_21	DigLib	hinst:or1200_cpu/or1200_fpu/fpu_arith/fpu_pre_norm_mul		hinst:or1...	241.8	241.79	10
FM_22	DigLib	hinst:or1200_cpu/or1200_fpu/fpu_arith/fpu_prenorm_addsub		hinst:or1...	1813.3	1813.28	78

```
        : Midas USF INFO
Info    : USF [USF_INF-6913]
        : update_usf_failure_mode - changing FM_1 failure mode flip flop to 0
        : Midas USF INFO
Warning : USF [USF_WAR-1792]
        : USF_MAPPING attribute for failure_mode Object Type - FM_19 Object ID and RISC_CORE_DETAILED FMEDA project is not defined
        : Midas USF WARNING
```

Safety Checks

USF `check_usf` Command

- **check_usf** -fmeda FMEDA_OpenRisc
- Rule Examples:

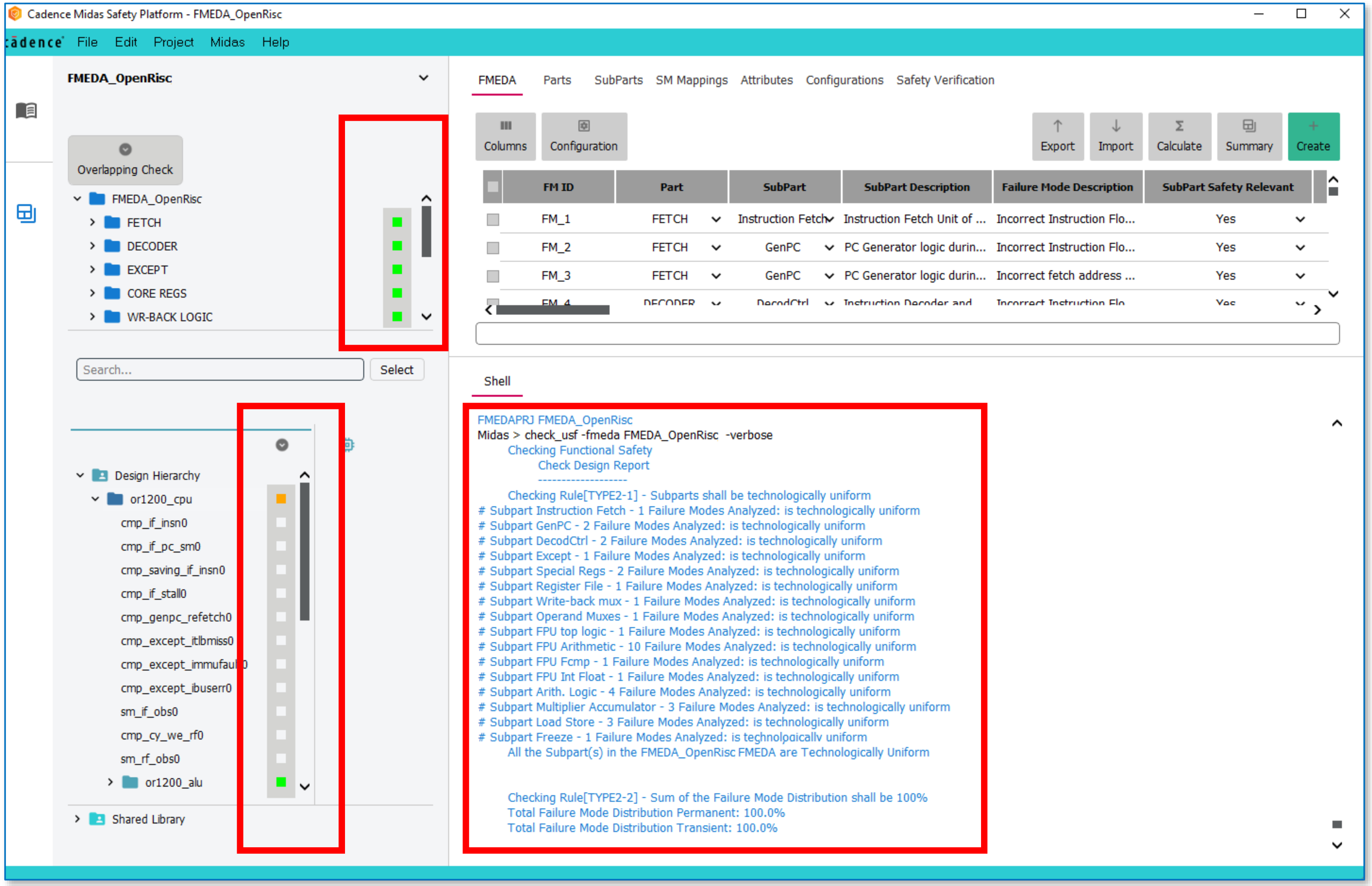
- **TYPE2-1:** *Subparts shall be technologically uniform*
- **TYPE2-2:** *Sum of the Failure Mode Distribution shall be 100%*
- **TYPE2-3:** *One safety mechanism should be defined for each failure mode*
- **TYPE2-5:** *All the design logic has been mapped to a Subpart*
- **TYPE2-6:** *All the design logic has been mapped to a Failure mode*
- **TYPE2-7:** *All the design logic has not been mapped to more than one Part*

Mimic `check_design`, `check_timing` … EDA commands: *"Make FuSa EDA-Friendly"*

- To report more information:
  - **check_usf** -fmeda FMEDA_OpenRisc -verbose
- Adding custom specific rules:

```
usf_add_drcrule <drcRuleTAG> {-active {on|off}} {-severity number} {-proc tclproc} {-description description}

        <drcRuleTAG>:
                Tag ID for the USF Rule to be added
        {-active {on|off}:
                Rule Checking status. off = the rule will not be considered
        {-severity}:
                Rule Severity, a number from 0 to 10 (0 is an error message; 1 is a warning message; 2 is info message)
        {-proc tclproc}:
                TCL procedure that will manage the DRC check. The TCL procedure has to exists
        {-description description}:
                Textual description for the rule to be added
```

Safety Checks on GUI



FM ID	Part	SubPart	SubPart Description	Failure Mode Description	SubPart Safety Relevant
FM_1	FETCH	Instruction Fetch	Instruction Fetch Unit of ...	Incorrect Instruction Flo...	Yes
FM_2	FETCH	GenPC	PC Generator logic durin...	Incorrect Instruction Flo...	Yes
FM_3	FETCH	GenPC	PC Generator logic durin...	Incorrect fetch address ...	Yes
FM_4	DECODER	DecodCtrl	Instruction Decoder and	Incorrect Instruction Flo	Yes

```
FMEDAPRJ FMEDA_OpenRisc
Midas > check_usf -fmeda FMEDA_OpenRisc  -verbose
        Checking Functional Safety
                Check Design Report
                -------------------

        Checking Rule[TYPE2-1] - Subparts shall be technologically uniform
# Subpart Instruction Fetch - 1 Failure Modes Analyzed: is technologically uniform
# Subpart GenPC - 2 Failure Modes Analyzed: is technologically uniform
# Subpart DecodCtrl - 2 Failure Modes Analyzed: is technologically uniform
# Subpart Except - 1 Failure Modes Analyzed: is technologically uniform
# Subpart Special Regs - 2 Failure Modes Analyzed: is technologically uniform
# Subpart Register File - 1 Failure Modes Analyzed: is technologically uniform
# Subpart Write-back mux - 1 Failure Modes Analyzed: is technologically uniform
# Subpart Operand Muxes - 1 Failure Modes Analyzed: is technologically uniform
# Subpart FPU top logic - 1 Failure Modes Analyzed: is technologically uniform
# Subpart FPU Arithmetic - 10 Failure Modes Analyzed: is technologically uniform
# Subpart FPU Fcmp - 1 Failure Modes Analyzed: is technologically uniform
# Subpart FPU Int Float - 1 Failure Modes Analyzed: is technologically uniform
# Subpart Arith. Logic - 4 Failure Modes Analyzed: is technologically uniform
# Subpart Multiplier Accumulator - 3 Failure Modes Analyzed: is technologically uniform
# Subpart Load Store - 3 Failure Modes Analyzed: is technologically uniform
# Subpart Freeze - 1 Failure Modes Analyzed: is technologically uniform
        All the Subpart(s) in the FMEDA_OpenRisc FMEDA are Technologically Uniform


        Checking Rule[TYPE2-2] - Sum of the Failure Mode Distribution shall be 100%
        Total Failure Mode Distribution Permanent: 100.0%
        Total Failure Mode Distribution Transient: 100.0%
```

USF **check_usf** on command line interface

- Safety hierarchy overlapping checks

🟩 The instances mapped to the given safety object (part, subpart, or failure mode) do not have any hierarchical dependency with other safety objects of the same type (part, subpart, or failure modes)

🟧 The instances mapped to the given safety object (part, subpart, or failure mode) have one or more hierarchical dependency with other safety objects of the same type (part, subpart, or failure modes)

- Failure modes mapping checks

⬛ Design instance is not mapped to any failure mode

🟩 Design instance is mapped to one failure mode

🟥 Design instance is mapped to more than one failure mode

USF Query & Reporting

`query_usf` USF Relational Queries

The **`query_usf`** command reports in a ‘TCL friendly’ format the information to create safety automations

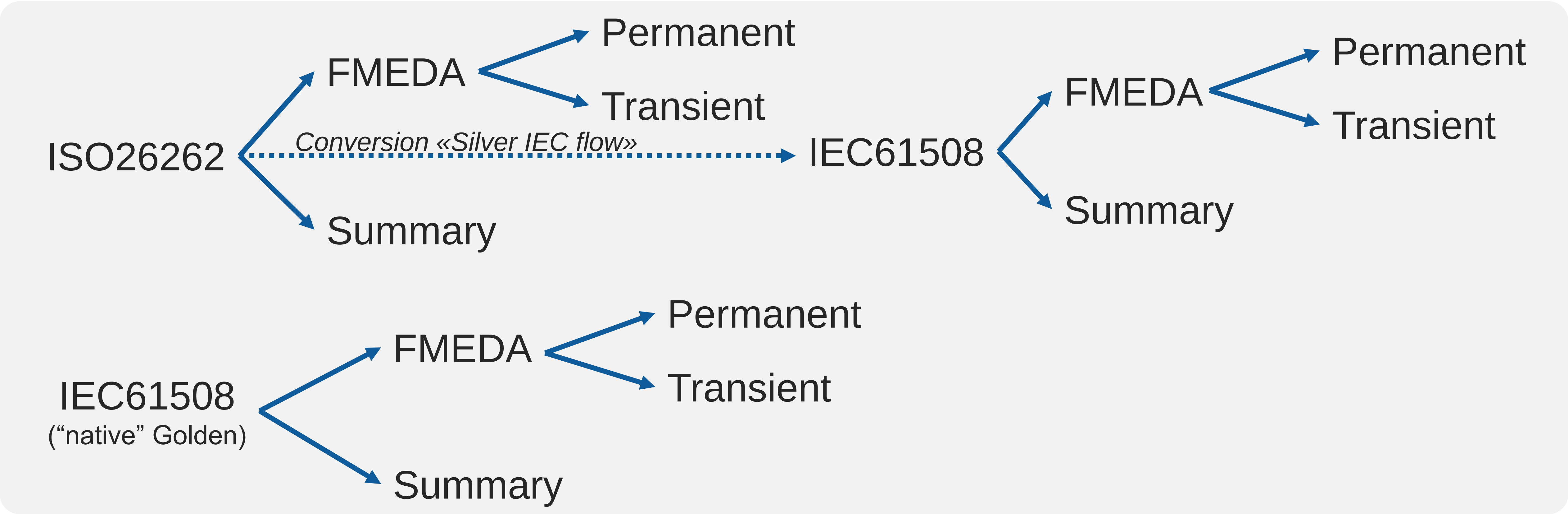
LEVEL 0	`query_usf *`	Listing available information
LEVEL 1	`query_usf {fmeda} {-obj_id id} {-obj_type type}`	Direct query
LEVEL 2	`query_usf {fmeda} {-obj_id id} {-obj_type type} [-ref_type RefType] [-ref_id refid]`	By referencing another object

- How many FMEDA projects do we have?
  - *`query_usf *`*
    - `FMEDAPRJ FMEDA_OpenRisc`
- How many Failure Modes have been defined for this project?
  - *`query_usf FMEDA_OpenRisc -obj_type failure_mode -obj_id *`*

    -
      ```
      FAILUREMODES FM_1 FM_2 FM_3 FM_4 FM_5 FM_6 FM_7 FM_8 FM_9 FM_10 FM_11 FM_12 FM_13 FM_14
      FM_15 FM_16 FM_17 FM_18 FM_19 FM_20 FM_21 FM_22 FM_23 FM_24 FM_25 FM_26 FM_27 FM_28
      FM_29 FM_30 FM_31 FM_32 FM_33 FM_34 FM_35
      ```
- Report the metrics for a specific FMEDA project
  - *`query_usf FMEDA_OpenRisc -obj_type fmeda -obj_id metrics`*

    -
      ```
      FMEDAPRJ FMEDA_OpenRisc off on on B off on {9 16 35} {57.5% 58.1% 100.0%} {4.269e-02
      6.753e-02 0.000e+00 1.005e-01 1.611e-01} DigLib {{134678.6 131265.7 6563.0} {98720.7
      96219.0 4431.0} {96364.7 93922.7 4328.0}} {57.52% -- -- -- --} {100.00% -- -- -- --}
      {58.09% -- -- --}
      ```

USF Reports: ISO26262 and IEC61508



ISO26262

FMEDA

Permanent

Transient

Summary

Conversion «Silver IEC flow»

IEC61508

FMEDA

Permanent

Transient

Summary

IEC61508

("native" Golden)

FMEDA

Permanent

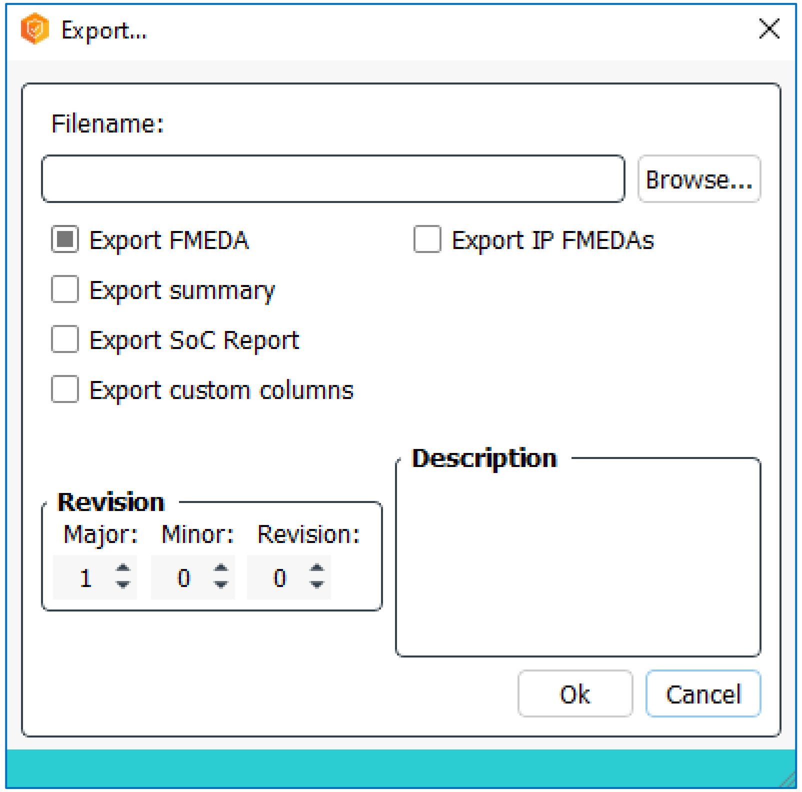
Transient

Summary

```
report_safety -standard iso26262 -fmeda myFMEDA permanent html "reports/ISO_PERMANENT.html"
report_safety -standard iso26262 -fmeda myFMEDA transient html "reports/ISO_TRANSIENT.html"
report_safety -standard iso26262 -fmeda myFMEDA report    html "reports/ISO_SUMMARY.html“

report_safety -standard iec61508 -fmeda myFMEDA permanent html "reports/IEC_PERMANENT.html"
report_safety -standard iec61508 -fmeda myFMEDA transient html "reports/IEC_TRANSIENT.html"
report_safety -standard iec61508 -fmeda myFMEDA report    html "reports/IEC_SUMMARY.html"
```

Midas Application Import-Export

- Microsoft Excel import/export is supported
- Rationales
  - Use USF (text file) for exchange/integration
  - Use MS Excel for final reporting and auditing

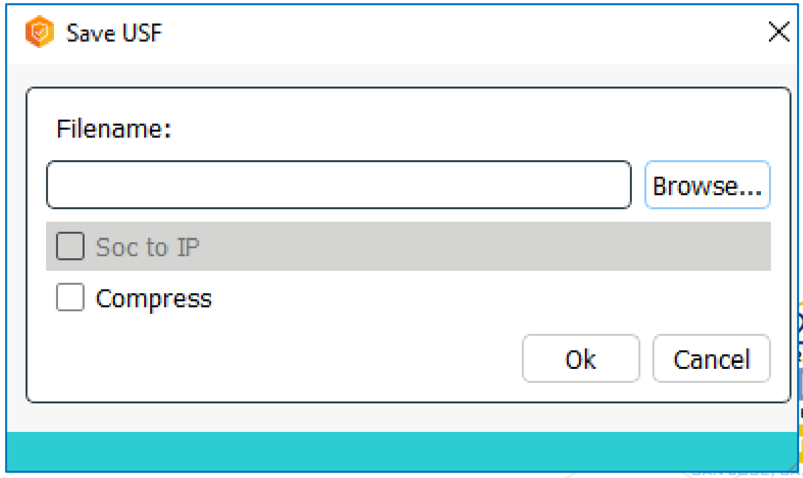


Filename:

■ Export FMEDA
□ Export IP FMEDAs
□ Export summary
□ Export SoC Report
□ Export custom columns

Revision
Major: 1
Minor: 0
Revision: 0

Description



Filename:

□ Soc to IP
□ Compress

FMEDA Compression

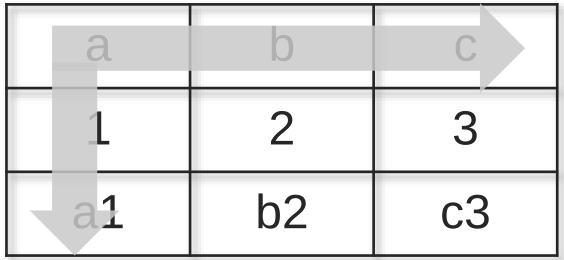
Reduce the number of safety objects, preserving the metrics

```
set_fmeda "IP1" -permanent -transient -ASIL B -architectural
create_technology "Tech1" -type Digital -fitperm 1.070e-006 -fittrans_gate 1.640e-006 -fitbit 1.640e-006 -refarea 1.026
…
create_technology "Tech5" -type Flash -fitperm 9.759e-004 -fittrans_gate 0.000e+000 -fitbit 9.759e-002 -refarea 1.026
create_part    "IP1/P1" -fmeda "IP1"
create_part    "IP1/P2" -fmeda "IP1"
create_subpart "IP1/P1/SP1" -part "IP1/P1" -fmeda "IP1"
…
create_subpart          "IP1/P2/SP2" -part "IP1/P2" -fmeda "IP1"
create_failure_mode     "IP1/P1/SP1+Tech1:FM1" -type Active -technology "Tech1" -subpart "IP1/P1/SP1" -gates 1234 -flops 567 -safe_perm 10 -fmeda "IP1"
create_failure_mode     "IP1/P1/SP1+Tech1:FM2" -type Passive -technology "Tech1" -subpart "IP1/P1/SP1" -gates 7654 -flops 321 -safe_trans 40 -fmeda "IP1"
…
create_failure_mode     "IP1/P2/SP2+Tech5:FM1" -type Mission -technology "Tech5" -subpart "IP1/P2/SP2" -membits 890 -safe_trans 70 -fmeda "IP1"
create_failure_mode     "IP1/P2/SP2+Tech5:FM2" -type Active -technology "Tech5" -subpart "IP1/P2/SP2" -membits 123 -safe_perm 5 -fmeda "IP1"
create_safety_mechanism "SM:IP1/P1" -type Custom -class HW
apply_safety_mechanism  "SM:IP1/P1" -to "IP1/P1/SP1+Tech1:FM1" -dcperm 80 -dctrans 90 -dclat 60 -fmeda "IP1"
…

save_usf saved_IPs_compress.usf -compress
```

```
set_fmeda "IP1" -permanent -transient -ASIL B -architectural
create_technology "Tech1" -type Digital -fitperm 1.070e-006 -fittrans_gate 1.640e-006 -fitbit 1.640e-006 -refarea 1.026
create_part part_IP1_Tech1 -fmeda IP1
create_subpart subpart_IP1_Tech1 -fmeda IP1 -part part_IP1_Tech1
create_failure_mode fm_IP1_Tech1_Active_on -type Active -technology Tech1 -fmeda IP1 -subpart subpart_IP1_Tech1 -gates 4936 -flops 2268 -safe_perm 10 -
safe_trans 0
create_safety_mechanism sm_IP1_Tech1_Active -type Custom -class HW
apply_safety_mechanism sm_IP1_Tech1_Active -to fm_IP1_Tech1_Active_on -fmeda IP1 -dcperm 80 -dctrans 90 -dclat 60
create_failure_mode fm_IP1_Tech1_Passive_on -type Passive -technology Tech1 -fmeda IP1 -subpart subpart_IP1_Tech1 -gates 30616 -flops 1284 -safe_perm 0 -
safe_trans 40
create_safety_mechanism sm_IP1_Tech1_Passive -type Custom -class HW
apply_safety_mechanism sm_IP1_Tech1_Passive -to fm_IP1_Tech1_Passive_on -fmeda IP1 -dclat 60
…
```

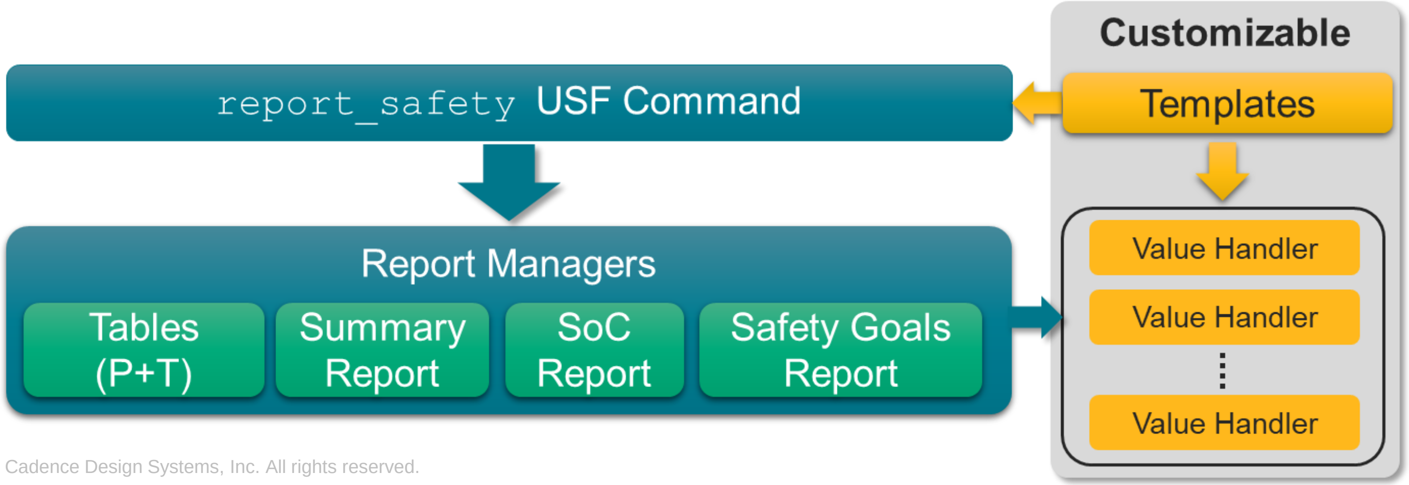
Report Customizations

- Report Managers
  - Organize the report information by rows and columns
  - Each report has its own template that defines the values handlers



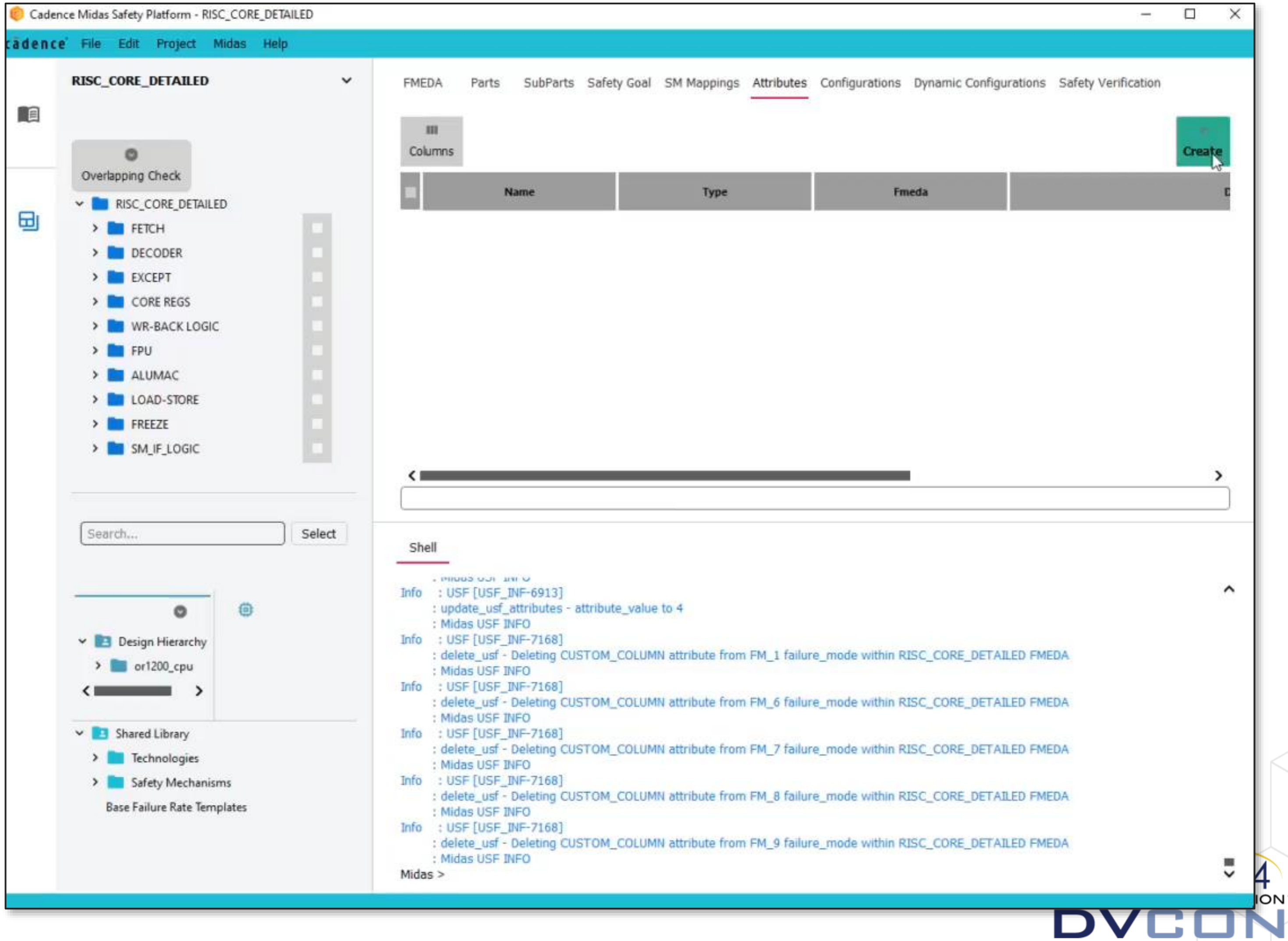
a	b	c
1	2	3
a1	b2	c3

- Templates
  - Stored in the `usf_report_safety_templates` directory
    - They can be replaced and customized by TCL procedures that have to follow a formalism defined in the USF command reference
- Customizations examples
  - Remove a column/row; Change the columns/row order
  - Adding a custom column/row by providing the TCL value handlers



Custom FMEDA Columns

- It is possibile to add custom columns to the FMEDA
- The custom FMEDA columns are leveraging USF attributes
- An attribute tagged to a failure mode can be a custom column
  - Select the «Create FMEDA custom column»
- Custom columns can be exported in the Excel reports



```
        : Midas USF INFO
Info    : USF [USF_INF-6913]
        : update_usf_attributes - attribute_value to 4
        : Midas USF INFO
Info    : USF [USF_INF-7168]
        : delete_usf - Deleting CUSTOM_COLUMN attribute from FM_1 failure_mode within RISC_CORE_DETAILED FMEDA
        : Midas USF INFO
Info    : USF [USF_INF-7168]
        : delete_usf - Deleting CUSTOM_COLUMN attribute from FM_6 failure_mode within RISC_CORE_DETAILED FMEDA
        : Midas USF INFO
Info    : USF [USF_INF-7168]
        : delete_usf - Deleting CUSTOM_COLUMN attribute from FM_7 failure_mode within RISC_CORE_DETAILED FMEDA
        : Midas USF INFO
Info    : USF [USF_INF-7168]
        : delete_usf - Deleting CUSTOM_COLUMN attribute from FM_8 failure_mode within RISC_CORE_DETAILED FMEDA
        : Midas USF INFO
Info    : USF [USF_INF-7168]
        : delete_usf - Deleting CUSTOM_COLUMN attribute from FM_9 failure_mode within RISC_CORE_DETAILED FMEDA
        : Midas USF INFO
Midas >
```

Failure Mode Distribution (FMD) Post-processing

- Post-process the failure mode distribution

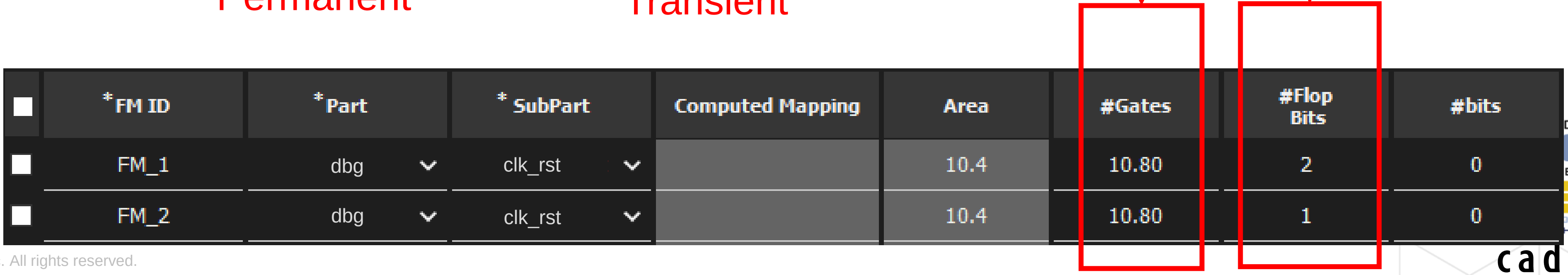
```
usf_set_fmd  {-fmeda fmendaprj}
             [-part part_name]
             [-subpart subpart_name]
             [-permanent]
             [-transient]
             [-strategy {area_uniform | fit_constant | custom} |
              -fm fm_name [-value {0-100}]
             [-distribution {distributions}]
             [-rounding_cost {default | cascade | sum_of_dist_diffs}]
```

- Example: custom redistribution

```
create_subpart clk_rst -part   dbg  -fmeda core  … -gates 21.61  -flops 3

create_failure_mode FM_1  -part dbg  -subpart clk_rst  -fmeda core …
create_failure_mode FM_2  -part dbg  -subpart clk_rst  -fmeda core …

usf_set_fmd -fmeda -part dbg -subpart etm_clk_rst \
            -strategy custom -distribution {FM_1 {50.0% 50.0%} FM_2 {50.0% 50.0%}}
```

Permanent

Transient

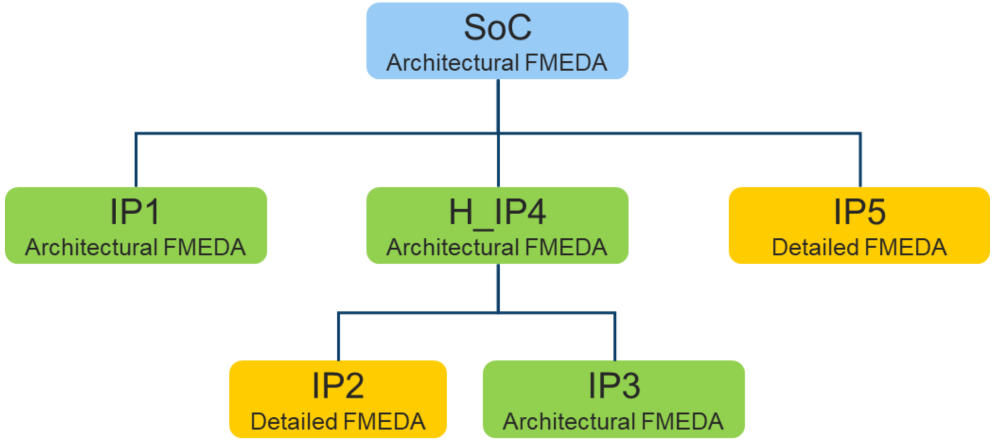


☐	*FM ID	*Part	*SubPart	Computed Mapping	Area	#Gates	#Flop Bits	#bits
☐	FM_1	dbg	clk_rst		10.4	10.80	2	0
☐	FM_2	dbg	clk_rst		10.4	10.80	1	0

SoC Safety Analysis

SoC Safety Analysis Integration

- SoC metrics are calculated combining (grouping) IP FMEDAs
- IP FMEDA work is partitioned, the owner of the overall safety analysis is grouping the IP FMEDAs into a SoC FMEDA
- Multiple levels of hierarchy are supported
- Combination of detailed and architectural FMEDA is possible
- Keep the details in the IP FMEDAs but keep SoC FMEDA as simple as possible
- Propagation and combination of Safety Goals (aka Failure Mode Effect)
- Ability to support weights of Failure Modes to different Safety Goals



```
# FMEDA 1
usf_reset
set troot1 {…}

load_usf [file join $troot1 "arm_cortex_m7_fmeda.usf"]
save_usf [file join $troot1 IP_USF "fmeda_1.usf"] -compress
set fmeda1 [lreplace [query_usf *] 0 0]

# FMEDA 2
usf_reset
set troot2 {..}
load_usf [file join $troot2 "dtmf.usf"]
save_usf [file join $troot2 IP_USF "fmeda_2.usf"] -compress
set fmeda2 [lreplace [query_usf *] 0 0]

# FMEDA ...

# Create SoC and group IP FMEDA
usf_reset
set_fmeda SOC -soc -ASIL B -permanent -transient -architectural
group_fmeda -fmeda_list [list $fmeda1 $fmeda2] \
            -fmeda_file [list [file join $troot1 IP_USF  "fmeda_1.usf"]\
                        [file join $troot2 IP_USF "fmeda_2.usf"]] -to SOC
```

Grouping IP FMEDAs into a SoC FMEDA: USF Command

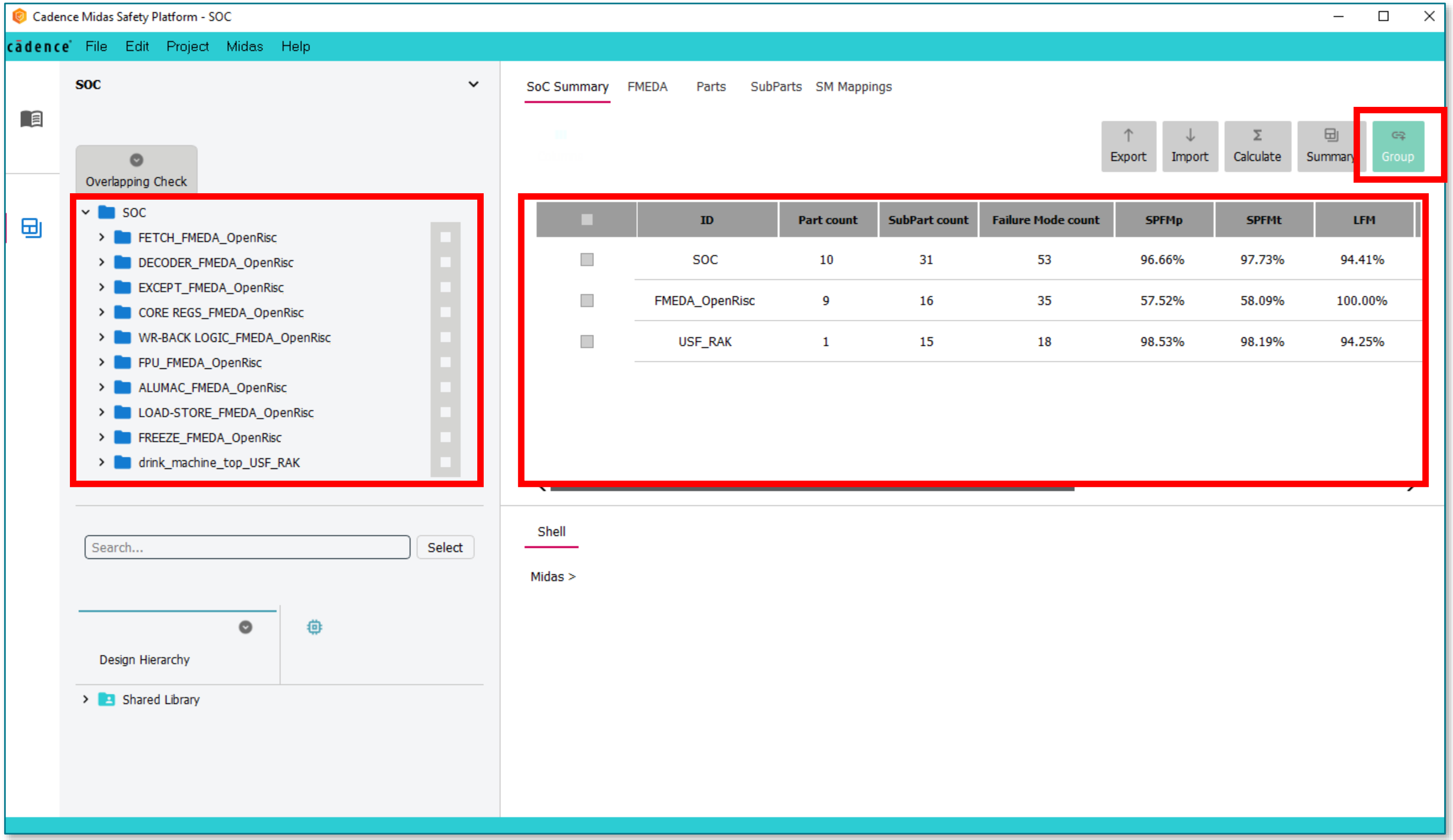
```
group_fmeda {-fmeda_list}
            [-fmeda_file]
            {-to fmeda_soc}
            [-linkonly]
```

`-fmeda_list FMEDA_tags_list`	Specify a list of FMEDA to link to a SoC FMEDA. With the format `FMEDAIP(`*`num_replica`*`),` automatically creates replicas of the same FMEDA
`-fmeda_file FMEDA_files_list`	Optional. Specify a list of FMEDA project files to link to an SoC FMEDA. The files are assumed to be generated using `save_usf` commands.
`-to fmeda_soc`	Specify that the SoC FMEDA is used as a reference for the FMEDA project. The SoC FMEDA must be previously created with the `set_fmeda` command using the `-soc` option.
`-linkonly`	Optional. Link an IP FMEDA to the SoC FMEDA without copying parts, subparts, and failure modes

Examples

- `group_fmeda -fmeda_list {myFMEDA1 myFMEDA2} -fmeda_file {myFMEDA1.usf myFMEDA2.usf} -to mySOCFMEDA`
- `group_fmeda -fmeda_list {myFMEDA1 myFMEDA2} -to mySOCFMEDA`

SoC FMEDA Project: Midas Application


Cadence Midas Safety Platform - SOC
cadence File Edit Project Midas Help
SOC
Overlapping Check
SoC Summary FMEDA Parts SubParts SM Mappings
Export Import Calculate Summary Group
Search... Select
Design Hierarchy
Shared Library
Shell
Midas >

- SOC
  - FETCH_FMEDA_OpenRisc
  - DECODER_FMEDA_OpenRisc
  - EXCEPT_FMEDA_OpenRisc
  - CORE REGS_FMEDA_OpenRisc
  - WR-BACK LOGIC_FMEDA_OpenRisc
  - FPU_FMEDA_OpenRisc
  - ALUMAC_FMEDA_OpenRisc
  - LOAD-STORE_FMEDA_OpenRisc
  - FREEZE_FMEDA_OpenRisc
  - drink_machine_top_USF_RAK

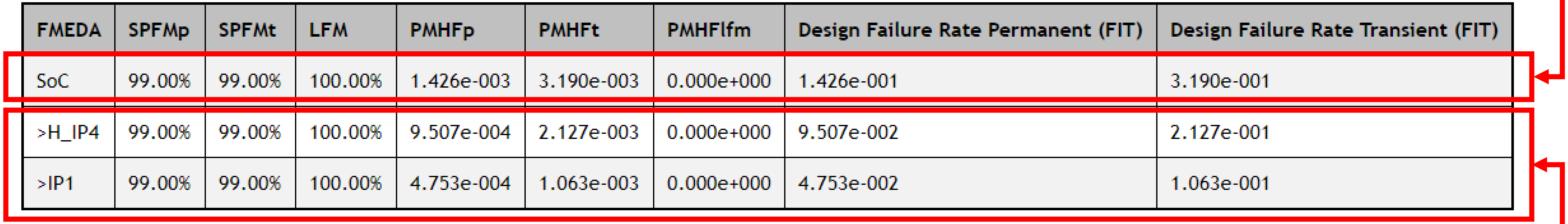
□	ID	Part count	SubPart count	Failure Mode count	SPFMp	SPFMt	LFM
□	SOC	10	31	53	96.66%	97.73%	94.41%
□	FMEDA_OpenRisc	9	16	35	57.52%	58.09%	100.00%
□	USF_RAK	1	15	18	98.53%	98.19%	94.25%

Grouping IP FMEDAs into a SoC FMEDA

- Safety Hierarchy
- SoC Summary

SoC Reports – USF Examples

- SoC Table
  - **report_safety** `-fmeda SoC soc html SoC_soc.html`

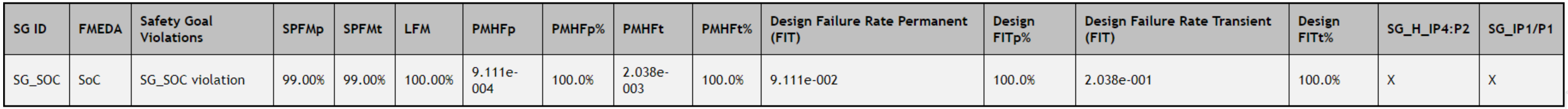


FMEDA	SPFMp	SPFMt	LFM	PMHFp	PMHFt	PMHFlfm	Design Failure Rate Permanent (FIT)	Design Failure Rate Transient (FIT)
SoC	99.00%	99.00%	100.00%	1.426e-003	3.190e-003	0.000e+000	1.426e-001	3.190e-001
>H_IP4	99.00%	99.00%	100.00%	9.507e-004	2.127e-003	0.000e+000	9.507e-002	2.127e-001
>IP1	99.00%	99.00%	100.00%	4.753e-004	1.063e-003	0.000e+000	4.753e-002	1.063e-001

SoC Summary

IPs Summary

- SoC Safety Goal table
  - **report_safety** `-fmeda SoC safety_goal html SoC_sg.html`

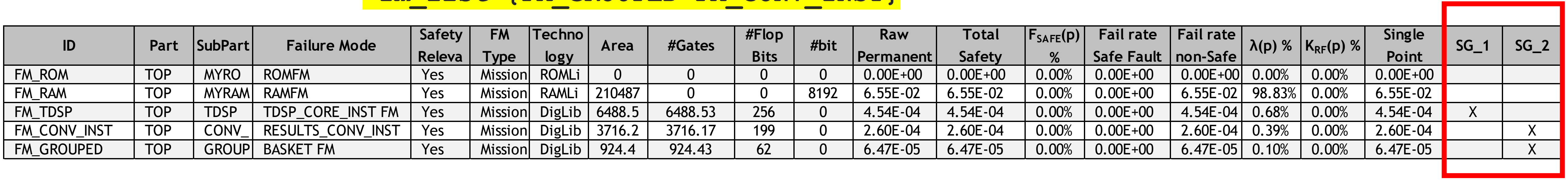


SG ID	FMEDA	Safety Goal Violations	SPFMp	SPFMt	LFM	PMHFp	PMHFp%	PMHFt	PMHFt%	Design Failure Rate Permanent (FIT)	Design FITp%	Design Failure Rate Transient (FIT)	Design FITt%	SG_H_IP4:P2	SG_IP1/P1
SG_SOC	SoC	SG_SOC violation	99.00%	99.00%	100.00%	9.111e-004	100.0%	2.038e-003	100.0%	9.111e-002	100.0%	2.038e-001	100.0%	X	X

Safety Goals (aka Failure Mode Effects, High Level Failure Modes)

- Can be used to track the metrics of a list of failure modes of a given IP FMEDA

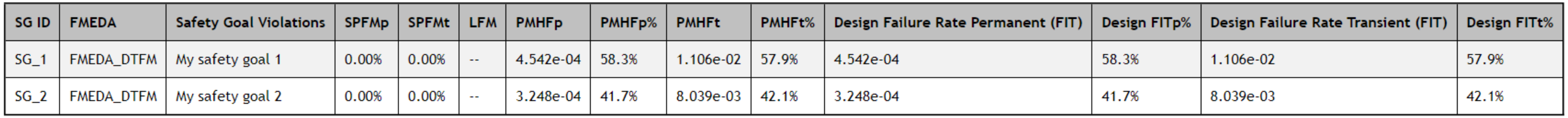
```
create_safety_goal SG_1 -description "My safety goal 1" -fmeda "FMEDA_DTFM" \
                   -fm_list {FM_TDSP}
create_safety_goal SG_2 -description "My safety goal 2" -fmeda "FMEDA_DTFM" \
                   -fm_list {FM_GROUPED FM_CONV_INST}
```



ID	Part	SubPart	Failure Mode	Safety Releva	FM Type	Technology	Area	#Gates	#Flop Bits	#bit	Raw Permanent	Total Safety	F_{SAFE}(p) %	Fail rate Safe Fault	Fail rate non-Safe	λ(p) %	K_{RF}(p) %	Single Point	SG_1	SG_2
FM_ROM	TOP	MYRO	ROMFM	Yes	Mission	ROMLi	0	0	0	0	0.00E+00	0.00E+00	0.00%	0.00E+00	0.00E+00	0.00%	0.00%	0.00E+00		
FM_RAM	TOP	MYRAM	RAMFM	Yes	Mission	RAMLi	210487	0	0	8192	6.55E-02	6.55E-02	0.00%	0.00E+00	6.55E-02	98.83%	0.00%	6.55E-02		
FM_TDSP	TOP	TDSP	TDSP_CORE_INST FM	Yes	Mission	DigLib	6488.5	6488.53	256	0	4.54E-04	4.54E-04	0.00%	0.00E+00	4.54E-04	0.68%	0.00%	4.54E-04	X	
FM_CONV_INST	TOP	CONV_	RESULTS_CONV_INST	Yes	Mission	DigLib	3716.2	3716.17	199	0	2.60E-04	2.60E-04	0.00%	0.00E+00	2.60E-04	0.39%	0.00%	2.60E-04		X
FM_GROUPED	TOP	GROUP	BASKET FM	Yes	Mission	DigLib	924.4	924.43	62	0	6.47E-05	6.47E-05	0.00%	0.00E+00	6.47E-05	0.10%	0.00%	6.47E-05		X

- It is possibile to export the Safety Goals metrics into a report

```
report_safety -fmeda FMEDA_DTFM safety_goal html "fmeda_sg.html"
```



SG ID	FMEDA	Safety Goal Violations	SPFMp	SPFMt	LFM	PMHFp	PMHFp%	PMHFt	PMHFt%	Design Failure Rate Permanent (FIT)	Design FITp%	Design Failure Rate Transient (FIT)	Design FITt%
SG_1	FMEDA_DTFM	My safety goal 1	0.00%	0.00%	--	4.542e-04	58.3%	1.106e-02	57.9%	4.542e-04	58.3%	1.106e-02	57.9%
SG_2	FMEDA_DTFM	My safety goal 2	0.00%	0.00%	--	3.248e-04	41.7%	8.039e-03	42.1%	3.248e-04	41.7%	8.039e-03	42.1%

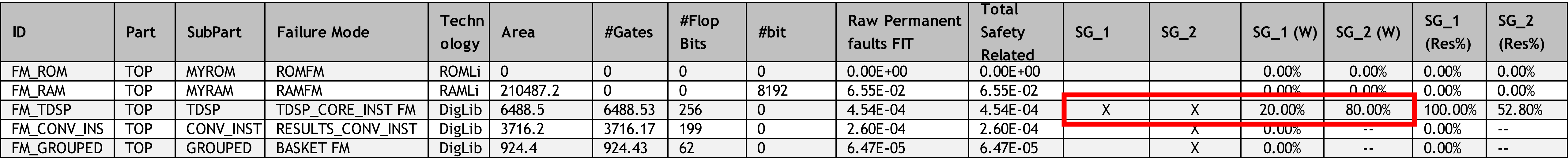
- It is possible to create SoC Safety Goals linked to IPs Safety Goals

```
create_safety_goal SGTOP -description "My new safety goal" -fmeda FMEDA_SOC \
                         -sg_list SG_1
```

Safety Goals (aka Failure Mode Effects, High Level Failure Modes)

- In case the same failure mode is distributed across different safety goals, it is possible to specify a list of weights (sum of the weights must be 100%)

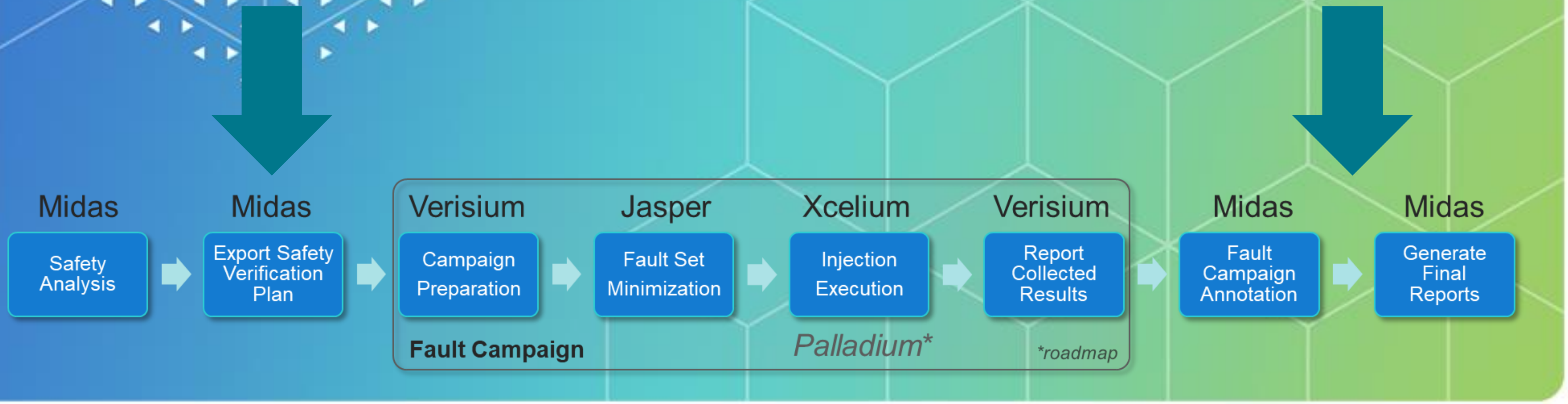
```
set_safety_goal_weights -fm_list {FM_TDSP} -fmeda "FMEDA_DTFM"  \
                        -list_weights {{SG_1 20} {SG_2 80}}
```



ID	Part	SubPart	Failure Mode	Technology	Area	#Gates	#Flop Bits	#bit	Raw Permanent faults FIT	Total Safety Related	SG_1	SG_2	SG_1 (W)	SG_2 (W)	SG_1 (Res%)	SG_2 (Res%)
FM_ROM	TOP	MYROM	ROMFM	ROMLi	0	0	0	0	0.00E+00	0.00E+00			0.00%	0.00%	0.00%	0.00%
FM_RAM	TOP	MYRAM	RAMFM	RAMLi	210487.2	0	0	8192	6.55E-02	6.55E-02			0.00%	0.00%	0.00%	0.00%
FM_TDSP	TOP	TDSP	TDSP_CORE_INST FM	DigLib	6488.5	6488.53	256	0	4.54E-04	4.54E-04	X	X	20.00%	80.00%	100.00%	52.80%
FM_CONV_INS	TOP	CONV_INST	RESULTS_CONV_INST	DigLib	3716.2	3716.17	199	0	2.60E-04	2.60E-04		X	0.00%	--	0.00%	--
FM_GROUPED	TOP	GROUPED	BASKET FM	DigLib	924.4	924.43	62	0	6.47E-05	6.47E-05		X	0.00%	--	0.00%	--

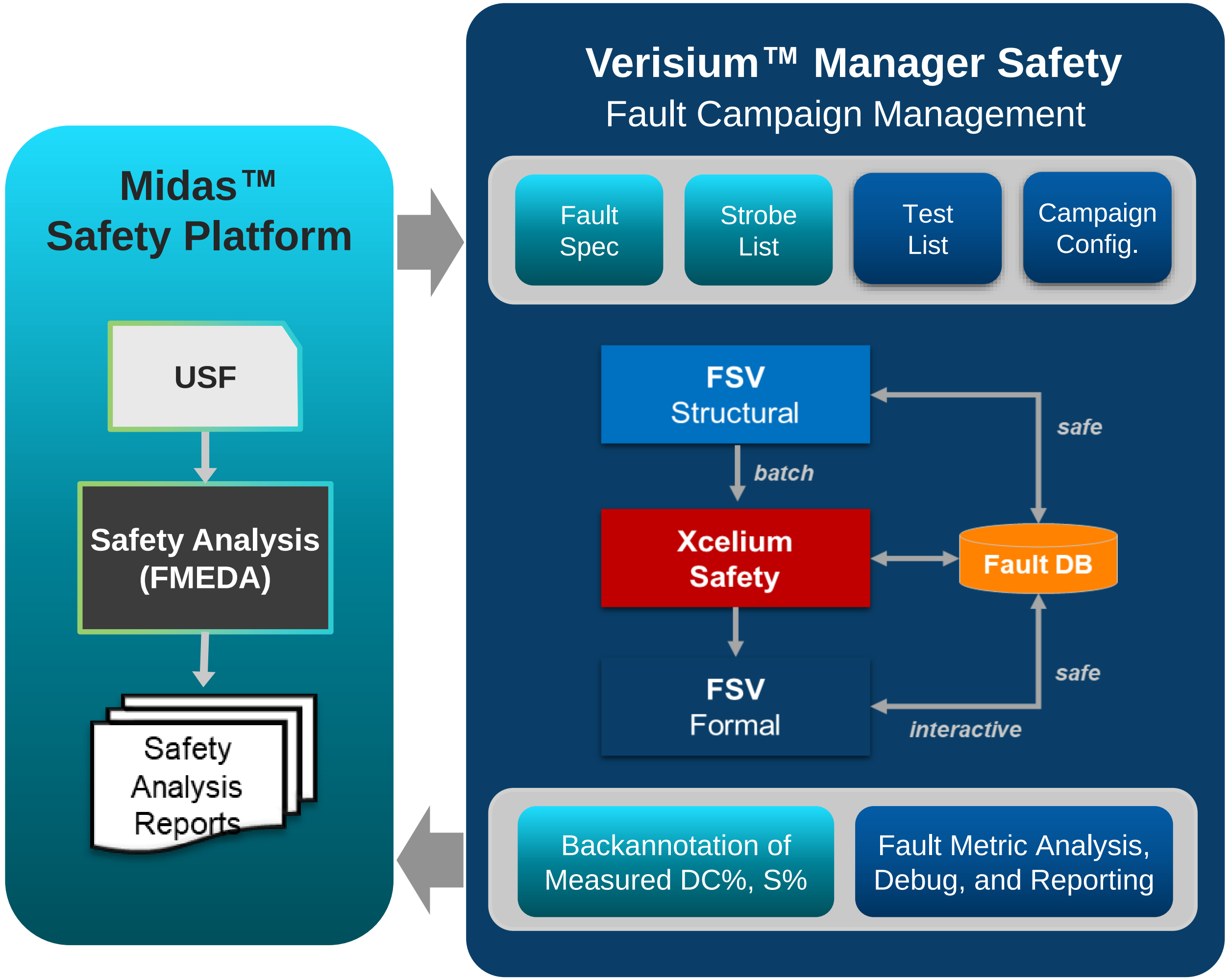
- Example use case

		Safety Goals			
		Deadlock	Data Corruption	Exceptions	Performance
FM_1	...	80%	20%	--	--
FM_2	...	--	100%	--	--
FM_3	...	--	--	50%	50%

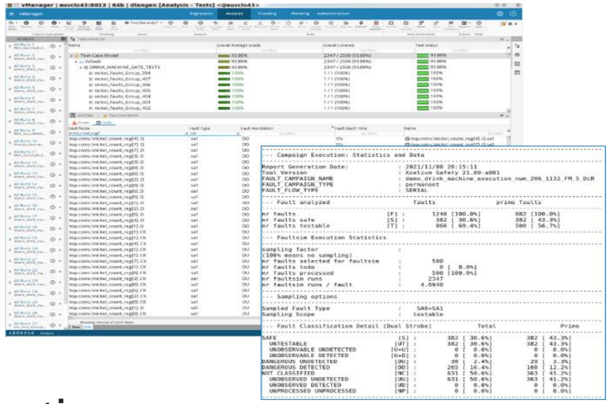


Safety Metrics Verification

Fault Campaign Management – Automation & Optimization



- Test selection and ranking
  - Coverage-based test selection
  - Customizable ranking criteria
- Fault list reduction
  - Fault sampling
  - Fault collapsing
  - Testability analysis
  - Test Dropping
- Fault campaigns execution
  - Measured Diagnostic Coverage and Safeness
  - Backannotation of results to FMEDA
  - Generate reports and analyze fault metric
  - FMEDA, fault classification, campaign summary,…

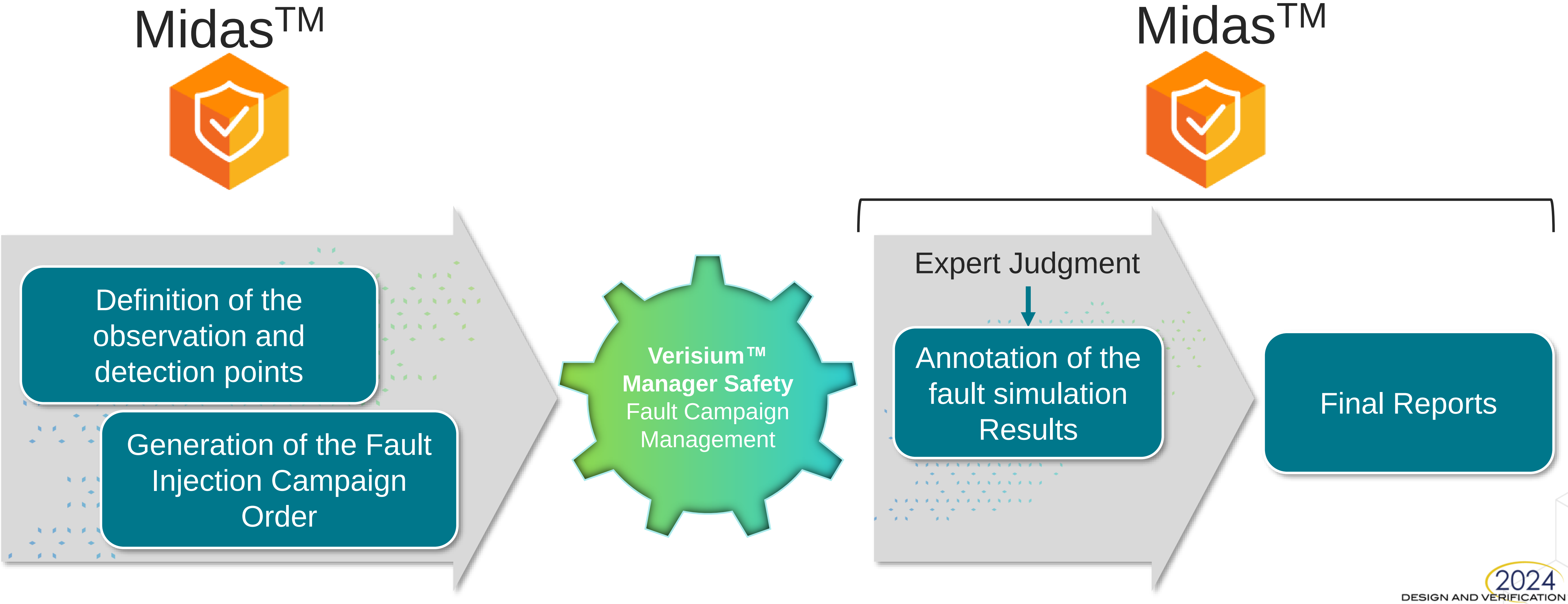


```
--- Campaign Execution: Statistics and Data
Report Generation Date:                 : 2021/11/08 20:15:11
Tool Version                            : Xcelium Safety 21.09-a001
FAULT_CAMPAIGN_NAME                     : demo_drink_machine_execution_num_206_1132_FM_5_DcR
FAULT_CAMPAIGN_TYPE                     : permanent
FAULT_FLOW_TYPE                         : SERIAL
--- Fault analyzed                           faults               prime faults
nr faults                        [F] :        1248 [100.0%]        882 [100.0%]
nr faults safe                   [S] :         382 [ 30.6%]        382 [ 43.3%]
nr faults testable               [T] :         866 [ 69.4%]        500 [ 56.7%]
--- Faultsim Execution Statistics
sampling factor                         :
(100% means no sampling)
nr faults selected for faultsim         :        500
nr faults todo                          :          0 [  0.0%]
nr faults processed                     :        500 [100.0%]
nr faultsim runs                        :       2347
nr faultsim runs / fault                :     4.6940
--- Sampling options
Sampled Fault Type                      :    SA0+SA1
Sampling Scope                          :   testable
--- Fault Classification Detail (Dual Strobe)          Total               Prime
SAFE                              [S] :          382 [ 30.6%]        382 [ 43.3%]
  UNTESTABLE                     [UT] :          382 [ 30.6%]        382 [ 43.3%]
  UNOBSERVABLE UNDETECTED       [U+U] :            0 [  0.0%]          0 [  0.0%]
  UNOBSERVABLE DETECTED         [U+D] :            0 [  0.0%]          0 [  0.0%]
DANGEROUS UNDETECTED             [DU] :           30 [  2.4%]         29 [  3.3%]
DANGEROUS DETECTED               [DD] :          205 [ 16.4%]        108 [ 12.2%]
NOT CLASSIFIED                   [NC] :          631 [ 50.6%]        363 [ 41.2%]
  UNOBSERVED UNDETECTED          [UU] :          631 [ 50.6%]        363 [ 41.2%]
  UNOBSERVED DETECTED            [UD] :            0 [  0.0%]          0 [  0.0%]
  UNPROCESSED UNPROCESSED        [NP] :            0 [  0.0%]          0 [  0.0%]
```

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES SAN JOSE, CA, USA MARCH 4-7, 2024

cādence®

Safety Metrics Verification



Midas™

Definition of the observation and detection points

Generation of the Fault Injection Campaign Order

Verisium™ Manager Safety Fault Campaign Management

Midas™

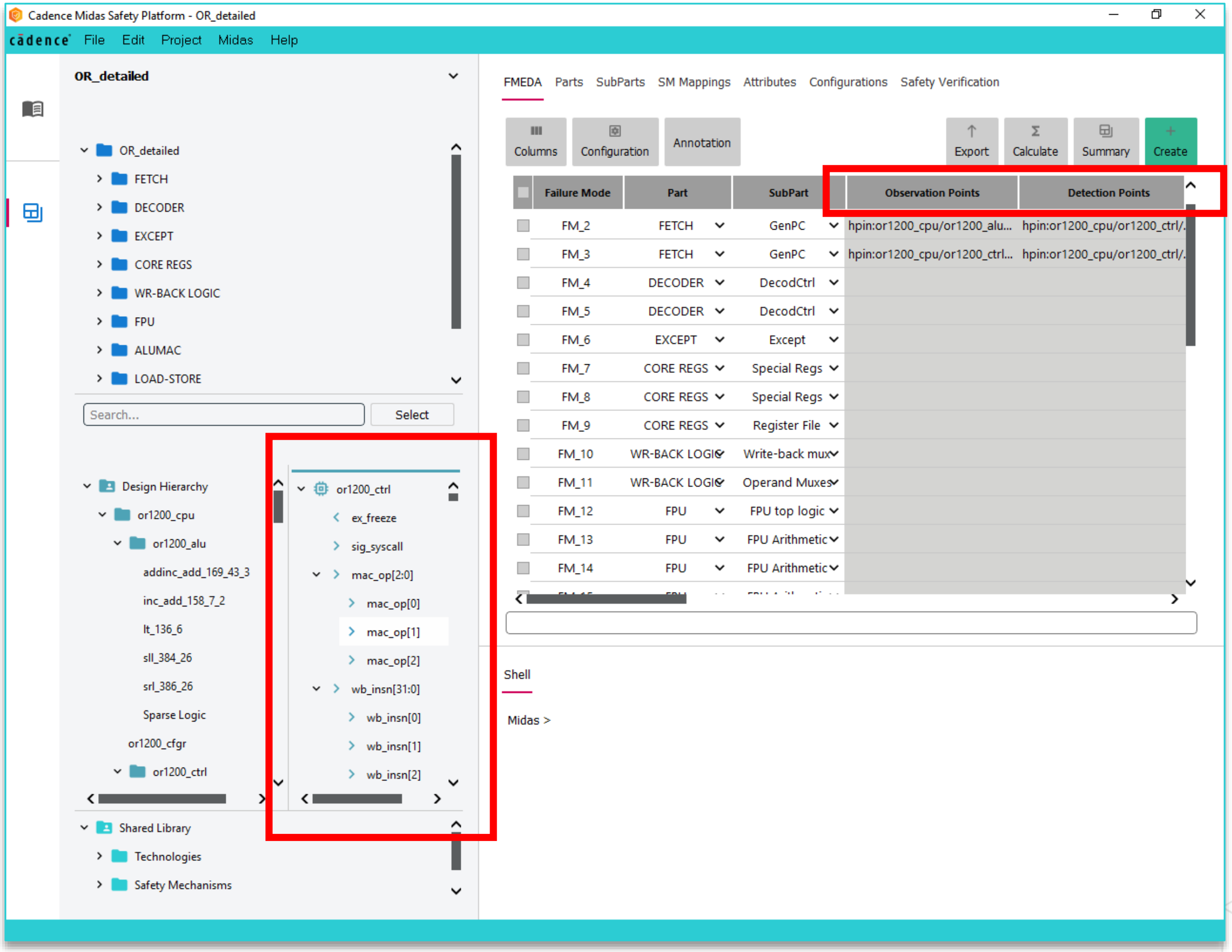
Expert Judgment

Annotation of the fault simulation Results

Final Reports

Strobing Points Definition

- Strobing points can be dragged & dropped from the design hierarchy into the related fields of the FMEDA
- The operation can be scripted



□	Failure Mode	Part	SubPart	Observation Points	Detection Points
□	FM_2	FETCH	GenPC	hpin:or1200_cpu/or1200_alu...	hpin:or1200_cpu/or1200_ctrl/...
□	FM_3	FETCH	GenPC	hpin:or1200_cpu/or1200_ctrl...	hpin:or1200_cpu/or1200_ctrl/...
□	FM_4	DECODER	DecodCtrl		
□	FM_5	DECODER	DecodCtrl		
□	FM_6	EXCEPT	Except		
□	FM_7	CORE REGS	Special Regs		
□	FM_8	CORE REGS	Special Regs		
□	FM_9	CORE REGS	Register File		
□	FM_10	WR-BACK LOGIC	Write-back mux		
□	FM_11	WR-BACK LOGIC	Operand Muxes		
□	FM_12	FPU	FPU top logic		
□	FM_13	FPU	FPU Arithmetic		
□	FM_14	FPU	FPU Arithmetic		

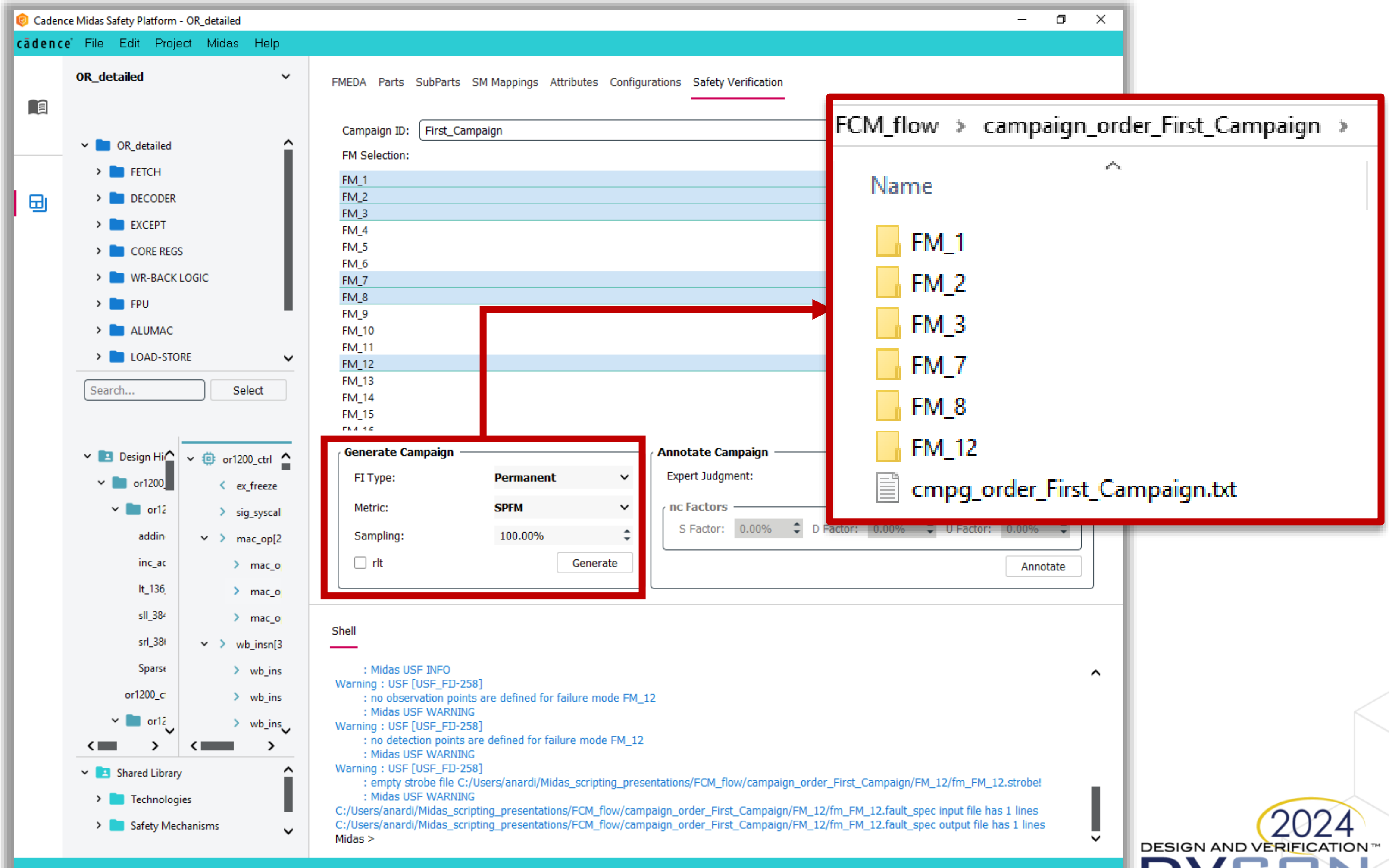
```
apply_safety_mechanism <SM_TAG> {-to FM_TAG} [-dcperm DCp_VAL] [-dctrans DCt_VAL] [-dclat DCl_VAL] \
                       {-fmeda FMEDA_TAG} [-diagnostic_points detpoints]

create_failure_mode ... [-observation_points obspoints] [-diagnostic_points detpoints]
```

SAN JOSE, CA, USA
MARCH 4-7, 2024

Driving Fault Simulation Campaign for DC Validation

Fault Injection Campaign Order Generation

- Generation of the campaign order
  - Summary of the Fault Injection Campaign
  - Fault specification file
  - Strobe specification
  - Verisium Manager configuration



Campaign ID: First_Campaign

FM Selection:

FM_1
FM_2
FM_3
FM_4
FM_5
FM_6
FM_7
FM_8
FM_9
FM_10
FM_11
FM_12
FM_13
FM_14
FM_15
FM_16

FI Type: Permanent

Metric: SPFM

Sampling: 100.00%

□ rlt

S Factor: 0.00% D Factor: 0.00% U Factor: 0.00%

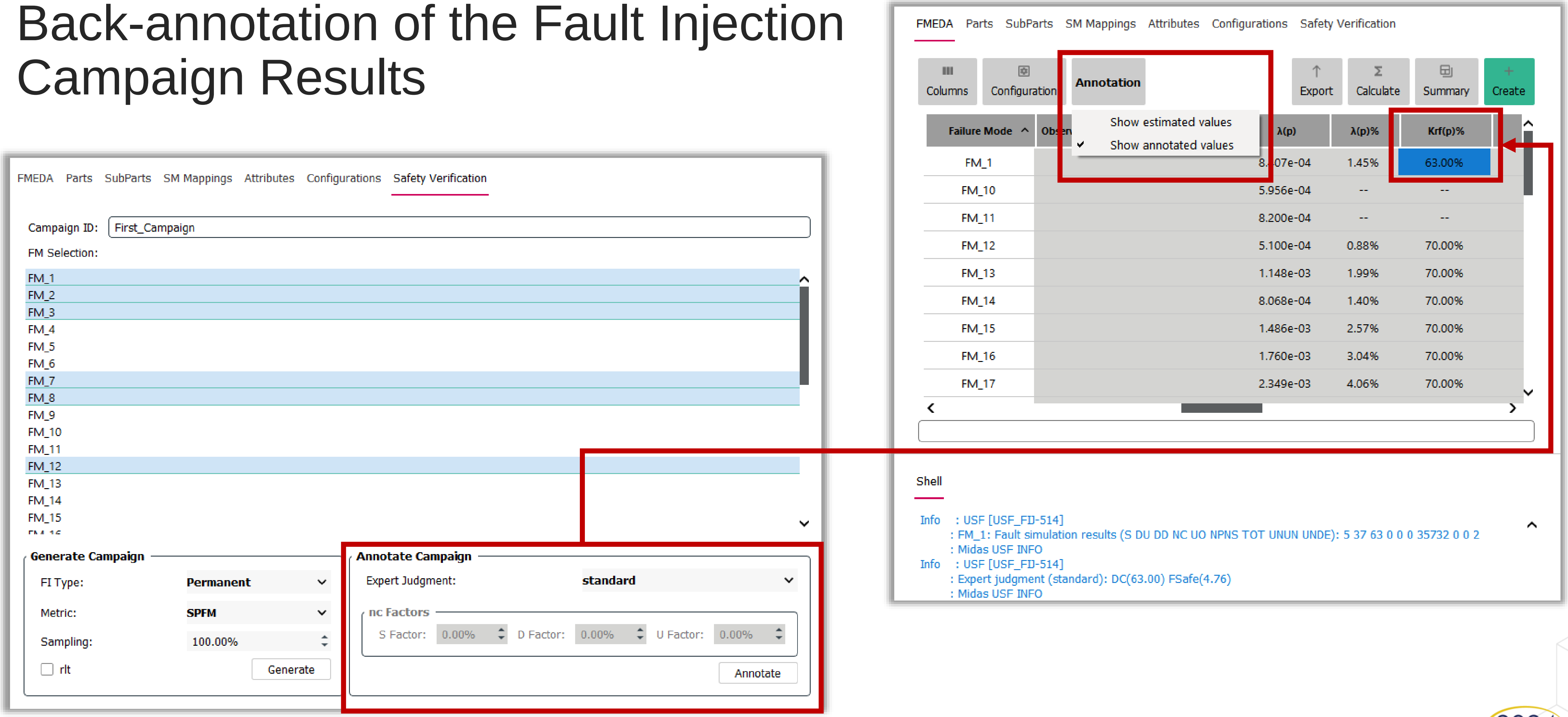
FM_1
FM_2
FM_3
FM_7
FM_8
FM_12
cmpg_order_First_Campaign.txt

```
        : Midas USF INFO
Warning : USF [USF_FI]-258]
        : no observation points are defined for failure mode FM_12
        : Midas USF WARNING
Warning : USF [USF_FI]-258]
        : no detection points are defined for failure mode FM_12
        : Midas USF WARNING
Warning : USF [USF_FI]-258]
        : empty strobe file C:/Users/anardi/Midas_scripting_presentations/FCM_flow/campaign_order_First_Campaign/FM_12/fm_FM_12.strobe!
        : Midas USF WARNING
C:/Users/anardi/Midas_scripting_presentations/FCM_flow/campaign_order_First_Campaign/FM_12/fm_FM_12.fault_spec input file has 1 lines
C:/Users/anardi/Midas_scripting_presentations/FCM_flow/campaign_order_First_Campaign/FM_12/fm_FM_12.fault_spec output file has 1 lines
Midas >
```

```
usf_campaign_order CAMPAIGN_FI1 -fmeda MYFMEDA -fm {I2C0_FM_1 I2C0_FM_2} \
                   -p -generate -sampling 100 -spfm
```

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES SAN JOSE, CA, USA MARCH 4-7, 2024

Back-annotation of the Fault Injection Campaign Results



Campaign ID: First_Campaign

FM Selection:

FM_1
FM_2
FM_3
FM_4
FM_5
FM_6
FM_7
FM_8
FM_9
FM_10
FM_11
FM_12
FM_13
FM_14
FM_15

Generate Campaign

FI Type: **Permanent**

Metric: **SPFM**

Sampling: 100.00%

☐ rlt

Annotate Campaign

Expert Judgment: **standard**

nc Factors

S Factor: 0.00% D Factor: 0.00% U Factor: 0.00%

Failure Mode	Obsen	$\lambda(p)$	$\lambda(p)$%	Krf(p)%
FM_1		8.[illegible]07e-04	1.45%	63.00%
FM_10		5.956e-04	--	--
FM_11		8.200e-04	--	--
FM_12		5.100e-04	0.88%	70.00%
FM_13		1.148e-03	1.99%	70.00%
FM_14		8.068e-04	1.40%	70.00%
FM_15		1.486e-03	2.57%	70.00%
FM_16		1.760e-03	3.04%	70.00%
FM_17		2.349e-03	4.06%	70.00%

Shell

```
Info   : USF [USF_FII-514]
      : FM_1: Fault simulation results (S DU DD NC UO NPNS TOT UNUN UNDE): 5 37 63 0 0 0 35732 0 0 2
      : Midas USF INFO
Info   : USF [USF_FII-514]
      : Expert judgment (standard): DC(63.00) FSafe(4.76)
      : Midas USF INFO
```

```
usf_campaign_order CAMPAIGN_FI1 -fmeda MYFMEDA -fm {I2C0_FM_1 I2C0_FM_2 I2C0_FM_3 I2C0_FM_4} -p -annotate \
                   -expert standard
```

Supported Expert Judgment Methods

standard (Default)

$$F_{safe} = \frac{S_{measured}}{S_{measured} + NC_{measured} + DD_{measured} + DU_{measured}}$$

$$DC = \frac{DD_{measured}}{DD_{measured} + DU_{measured}}$$

progressive

$$Rate = \frac{DD_{measured} + DU_{measured} + S_{measured}}{S_{measured} + NC_{measured} + DD_{measured} + DU_{measured}}$$

- F_{safe} is computed with $NC_{measured}$
- Rate is evaluating the % of $NC_{measured}$
  - Inversely: NC High → Rate Low

Automatically and conservatively moves a given percentage of NC faults to detected - The higher NC are, the less they are moved

Rate	DC calculation
> 75%	$DC_{75\%} = \frac{DD_{measured}}{DD_{measured} + DU_{measured}}$
>= 50% && <= 75%	$DD_{rate} = NC_{measured} * 0.5 * DC_{75\%}$ $DC = \frac{DD_{measured} + DD_{rate}}{DD_{measured} + NC_{measured} + DU_{measured}}$
< 50%	$DD_{rate} = NC_{measured} * 0.25 * DC_{75\%}$ $DC = \frac{DD_{measured} + DD_{rate}}{DD_{measured} + NC_{measured} + DU_{measured}}$

ncshare (two factors expert judgment)

Distribution of the Not Classified (NC) faults according to configurable percentages. DUassumed% + DDassumed% + Sassumed% = 100%

$NC_{measured}$

$S_{assumed} = \% \text{ Sassumed} * NC_{measured}$

$DD_{assumed} = \% \text{ DDassumed} * (NC_{measured} - S_{assumed})$

$DU_{assumed} = NC_{measured} - (S_{assumed} + DD_{assumed})$

$$F_{safe} = \frac{S_{measured} + S_{assumed}}{TOT = S_{measured} + NC_{measured} + DD_{measured} + DU_{measured}}$$

$$DC = \frac{(DD_{measured} + DD_{assumed})}{(DD_{measured} + DD_{assumed}) + (DU_{measured} + DU_{assumed})}$$

ncjudge

Redistribute percentages of faults to a given basket, with the only limitation that the total number of redistributed faults cannot be higher than the total NC.

```
NC = NOT CLASSIFIED
Sfactor% → Sassumed =  NC x Sfactor%
Dfactor% → DDassumed = NC x Dfactor%
Ufactor% → DUassumed = NC x Ufactor%

Sfactor% + Dfactor% + Ufactor% <= 100%
```

$$F_{safe} = \frac{S_{measured} + S_{assumed}}{TOT = S_{measured} + NC_{measured} + DD_{measured} + DU_{measured}}$$

$$DC = \frac{(DD_{measured} + DD_{assumed})}{(DD_{measured} + DD_{assumed}) + (DU_{measured} + DU_{assumed})}$$

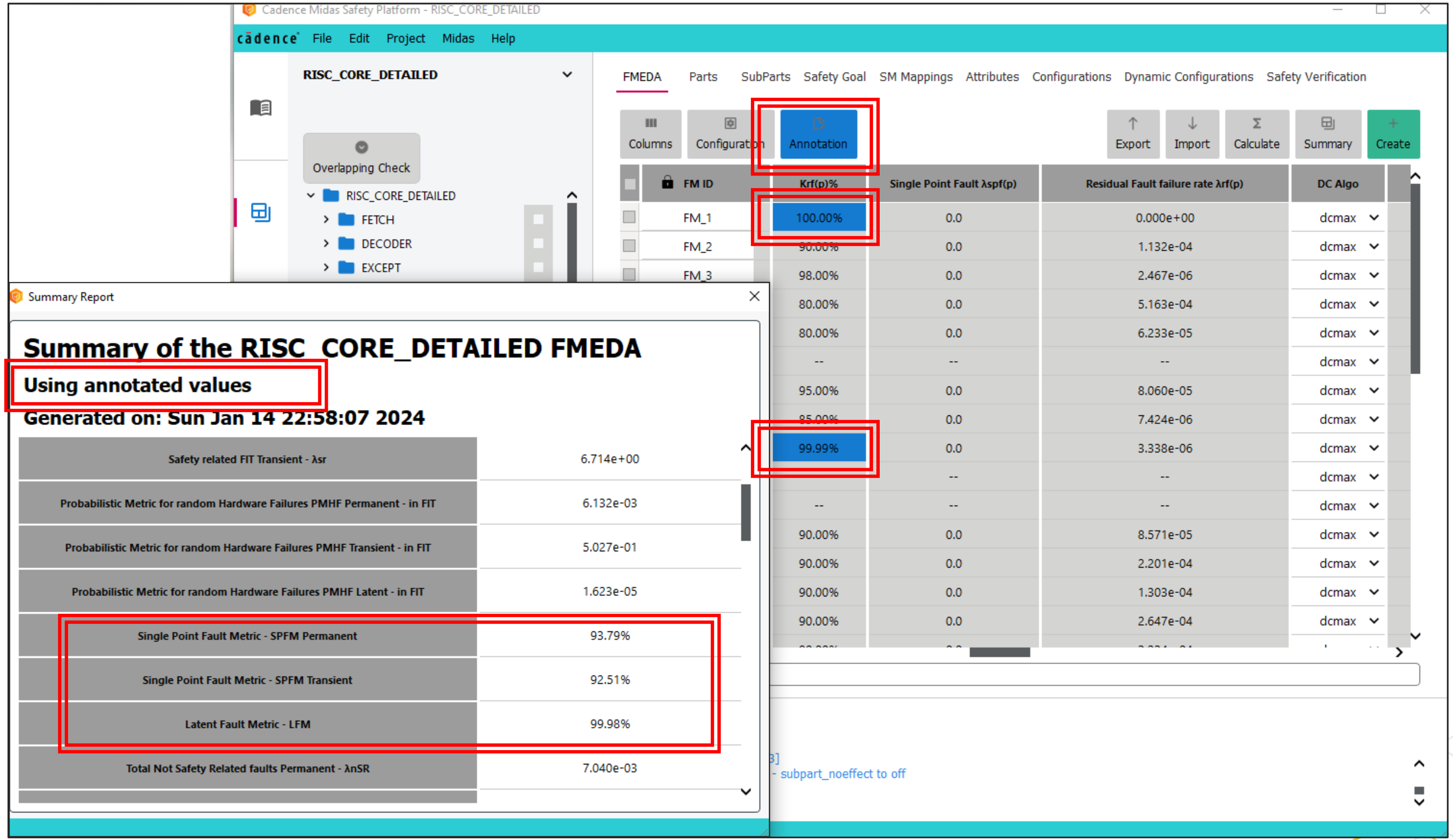
direct

2024

Enable external, not supported expert judgment algorithms. Use it to directly annotate DC or safe values based on users evaluation. The provided DC and safe values are annotated to the target failure modes.

Generate Final Reports

- Once annotated, both estimated and measured values are available
- Switch between the two modes and generate reports
- Save and restore


Cadence Midas Safety Platform - RISC_CORE_DETAILED
cādence File Edit Project Midas Help
RISC_CORE_DETAILED
Overlapping Check
RISC_CORE_DETAILED
FETCH
DECODER
EXCEPT
FMEDA Parts SubParts Safety Goal SM Mappings Attributes Configurations Dynamic Configurations Safety Verification
Columns
Configuration
Annotation
Export
Import
Calculate
Summary
Create
subpart_noeffect to off

FM ID	Krf(p)%	Single Point Fault λspf(p)	Residual Fault failure rate λrf(p)	DC Algo
FM_1	100.00%	0.0	0.000e+00	dcmax
FM_2	90.00%	0.0	1.132e-04	dcmax
FM_3	98.00%	0.0	2.467e-06	dcmax
	80.00%	0.0	5.163e-04	dcmax
	80.00%	0.0	6.233e-05	dcmax
	--	--	--	dcmax
	95.00%	0.0	8.060e-05	dcmax
	85.00%	0.0	7.424e-06	dcmax
	99.99%	0.0	3.338e-06	dcmax
		--	--	dcmax
	--	--	--	dcmax
	90.00%	0.0	8.571e-05	dcmax
	90.00%	0.0	2.201e-04	dcmax
	90.00%	0.0	1.303e-04	dcmax
	90.00%	0.0	2.647e-04	dcmax

Summary Report

Summary of the RISC_CORE_DETAILED FMEDA

Using annotated values

Generated on: Sun Jan 14 22:58:07 2024

Safety related FIT Transient - λsr	6.714e+00
Probabilistic Metric for random Hardware Failures PMHF Permanent - in FIT	6.132e-03
Probabilistic Metric for random Hardware Failures PMHF Transient - in FIT	5.027e-01
Probabilistic Metric for random Hardware Failures PMHF Latent - in FIT	1.623e-05
Single Point Fault Metric - SPFM Permanent	93.79%
Single Point Fault Metric - SPFM Transient	92.51%
Latent Fault Metric - LFM	99.98%
Total Not Safety Related faults Permanent - λnSR	7.040e-03

```
report_safety -fmeda RISC_CORE_DETAILED report html summary.html -annotate
```

CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

2024 DESIGN AND VERIFICATION DVCON CONFERENCE AND EXHIBITION UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

USF-based FMEDA-driven Functional Safety Verification

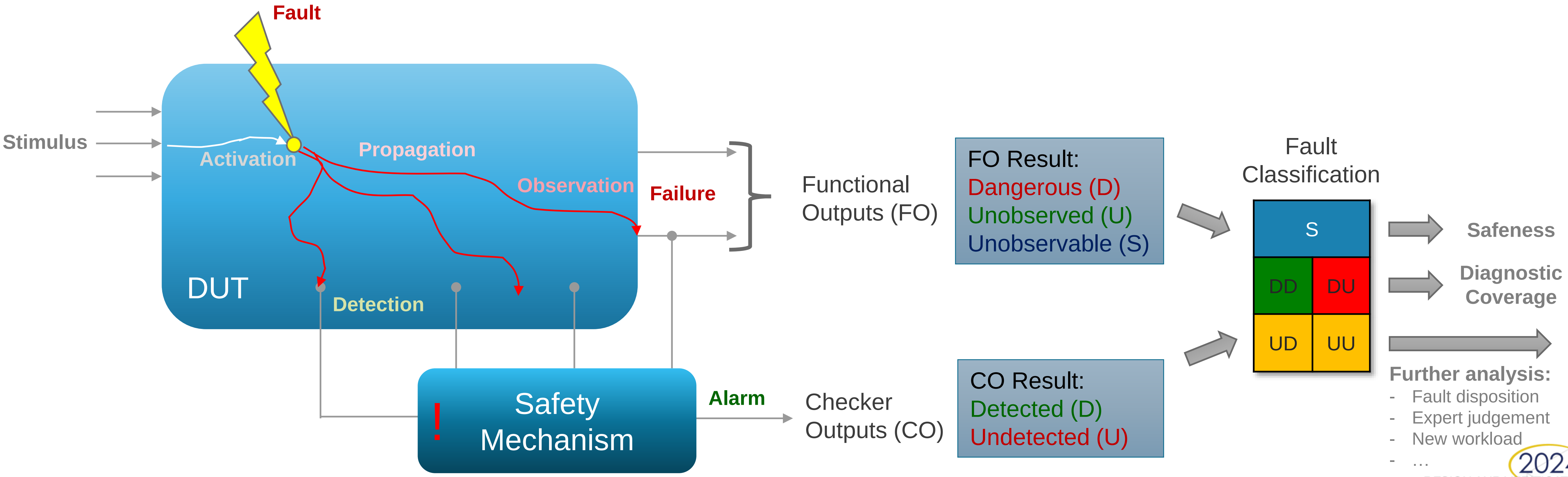
Fault Campaign Management (Verisium Manager Safety + Xcelium Safety + Jasper Safety)

Mangesh Mukundrao Pande

cadence

Automotive / Functional Safety / Random Faults / …

- Goal: *prevent or mitigate the effect of a hazardous event due to (operational)* ***random faults***
- Requirement: *deliver* ***diagnostic coverage*** *according to ASIL (Automotive Safety Integrity Level)*
- Method: *integrate* ***safety mechanisms*** *across the system architecture*
- Validation: *show evidence and assess robustness via* ***fault injection***



FO Result:
Dangerous (D)
Unobserved (U)
Unobservable (S)

CO Result:
Detected (D)
Undetected (U)

Further analysis:
- Fault disposition
- Expert judgement
- New workload
- …

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES SAN JOSE, CA, USA MARCH 4-7, 2024

cadence

Digital Safety Verification

FMEDA-driven safety verification

Campaign Automation – Verisium™ Manager Safety

- Unified front-end to manage all engines and analyze results
- Validation and FMEDA back-annotation of diagnostic coverage

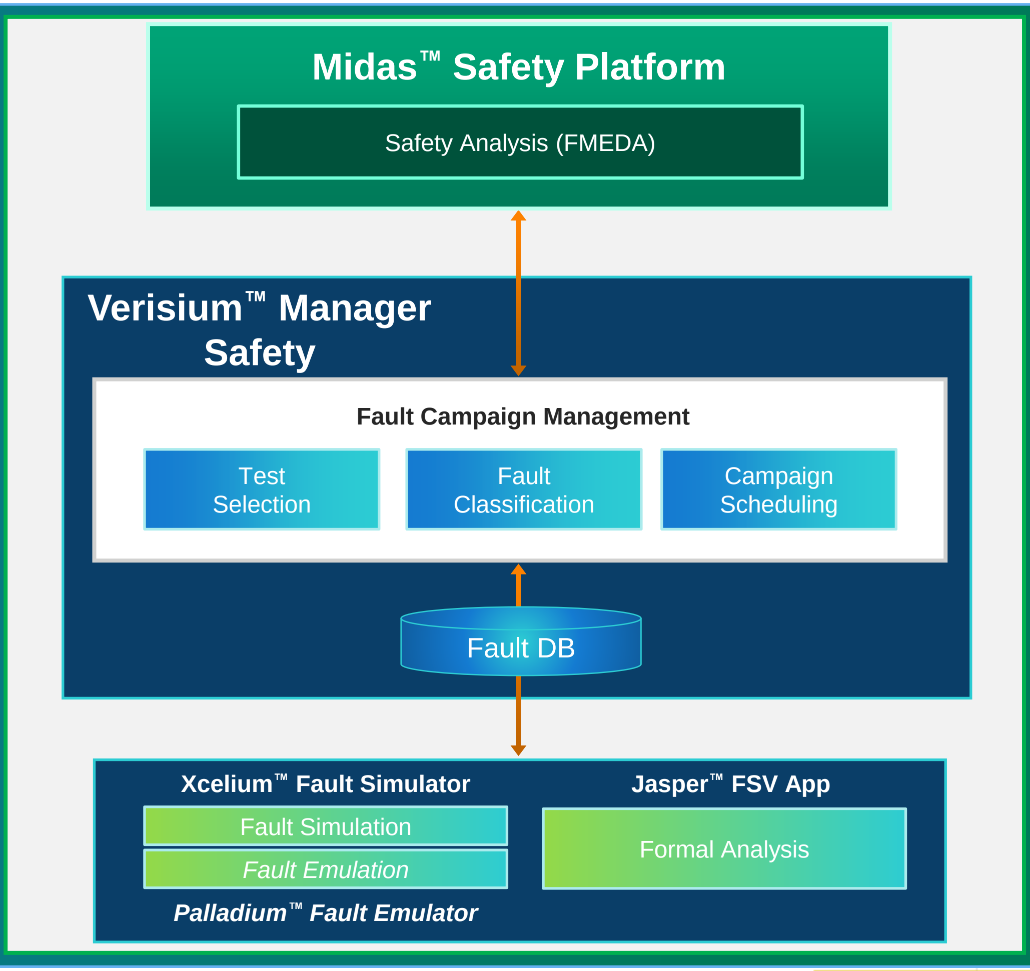
Complexity Reduction – Jasper™ FSV App

- Applies industry-leading formal techniques to fault analysis
- Increases safety verification performance

Injection Engine – Xcelium™ Fault Simulator

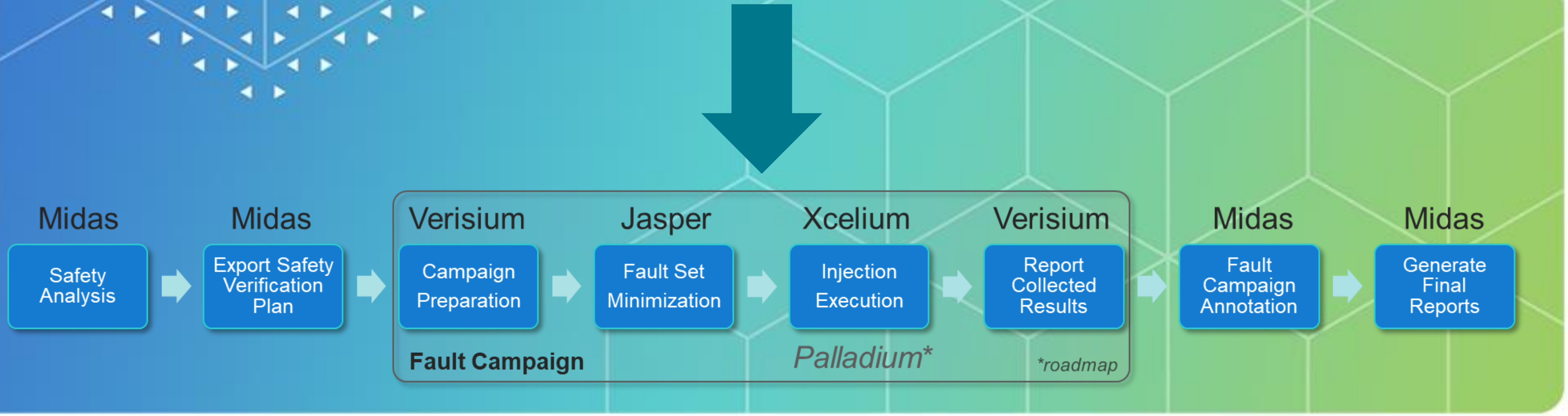
- Native serial and concurrent fault simulation engine

Acceleration – Palladium™ Fault Emulator



UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

cadence®



Verisium™ Manager Safety

Fault Campaign Manager – FCM

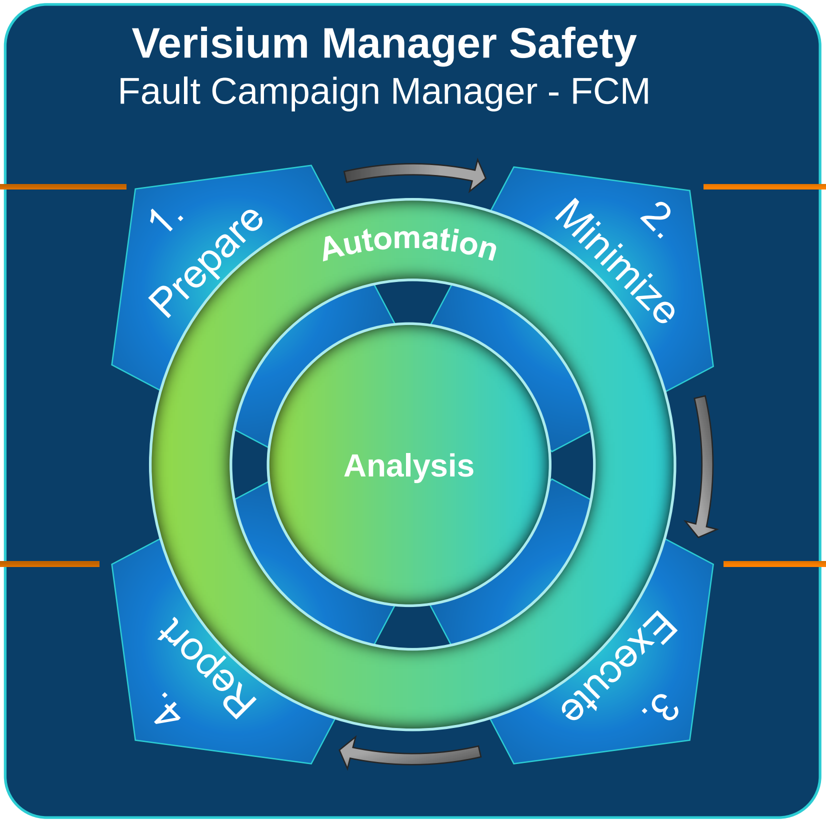
Fault Campaign Automation and Analysis

1. Prepare Data

- Single front-end campaign configuration
- Expand fault targets & instrument design
- Translate strobe definition

2. Minimize Fault Set

- Collapse redundant faults
- Identify unobservable/safe faults
- Test-based fault pruning



4. Generate Reports

- Campaign summary report
- Diagnostic Coverage / Safeness
- FMEDA validated results

3. Execute Campaign

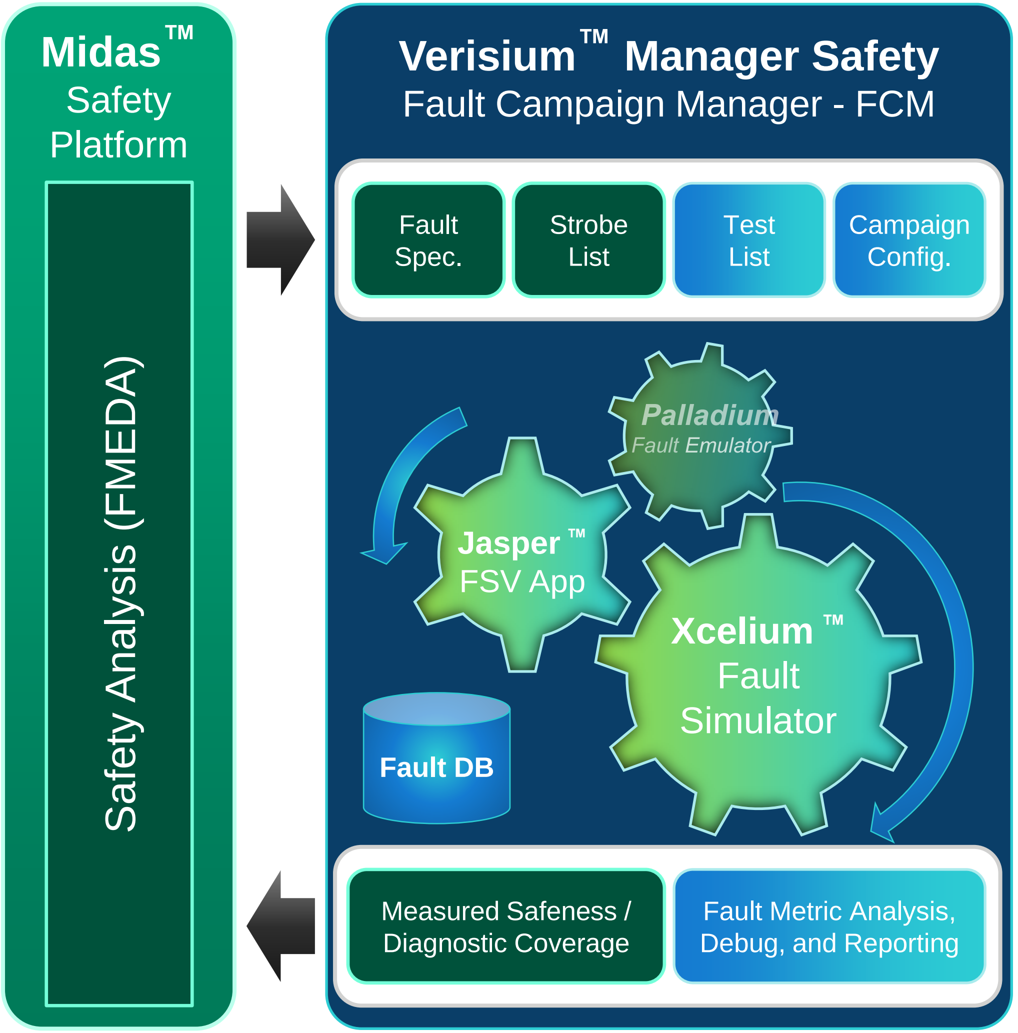
- Create runs per fault groups
- Verisium Manager state-of-the-art DRM
- Drop exhausted faults/tests

Fault Metric Analysis

- Merge fault results across different campaigns
- Disposition of not-classified faults
- Offer insights towards analysis closure

Prepare
Minimize
Execute
Report
Analyze

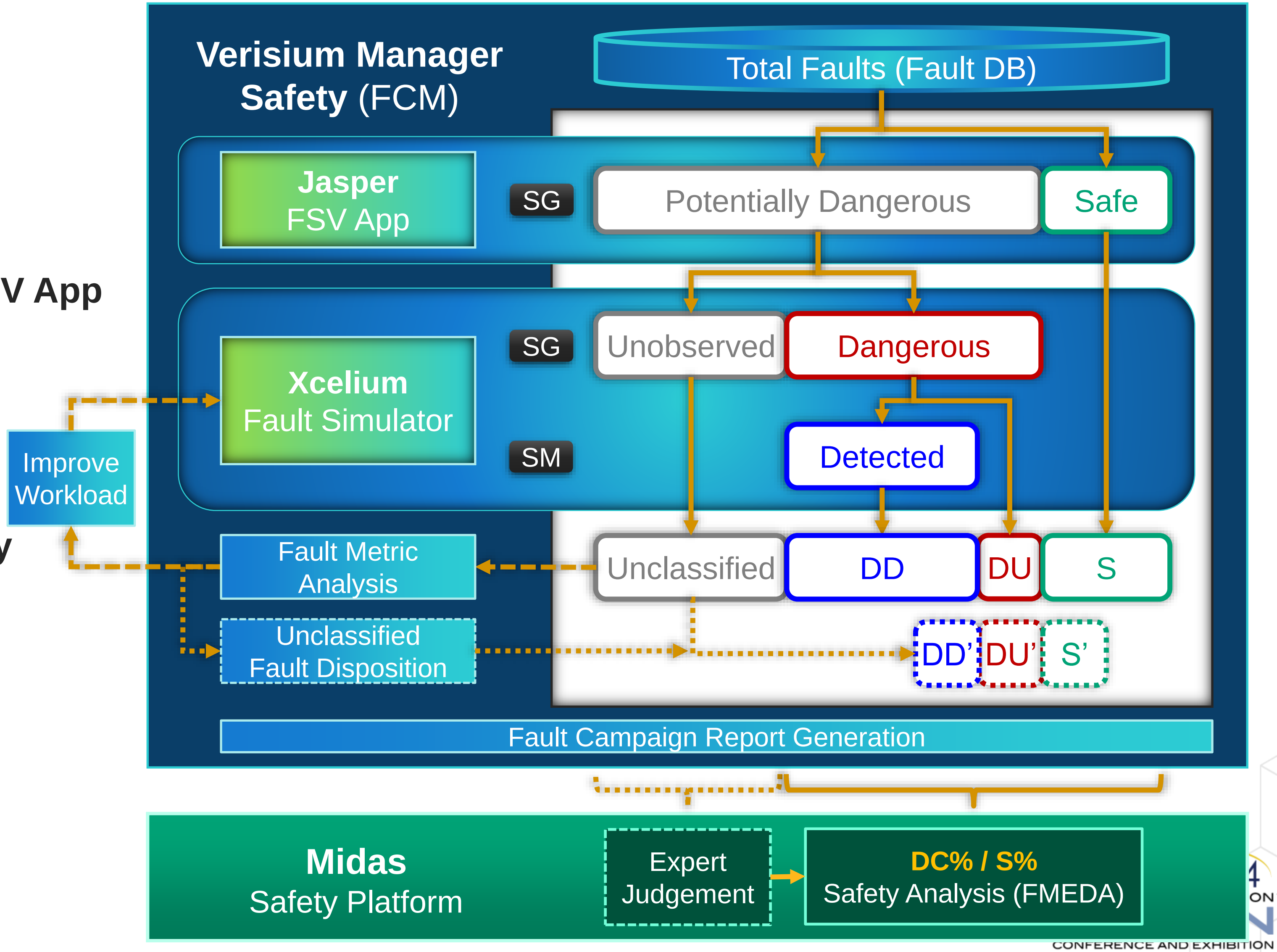
FMEDA-driven Fault Campaign



- **Inputs**
  - Safety Engineer
    - Fault Targets (derived from FMEDA ⇔ design mapping)
    - Strobe List (observation and detection points)
  - Verification Engineer
    - Test List (selected for fault analysis)
    - Campaign Configuration
      - Optimizations, runs distribution, customization, etc.
- **Outputs**
  - Summary Report
    - Measured Fault/Diagnostic Coverage, Safeness
  - Fault Annotation
    - Fault Metric Analysis, annotated fault list, …

Fault Classification

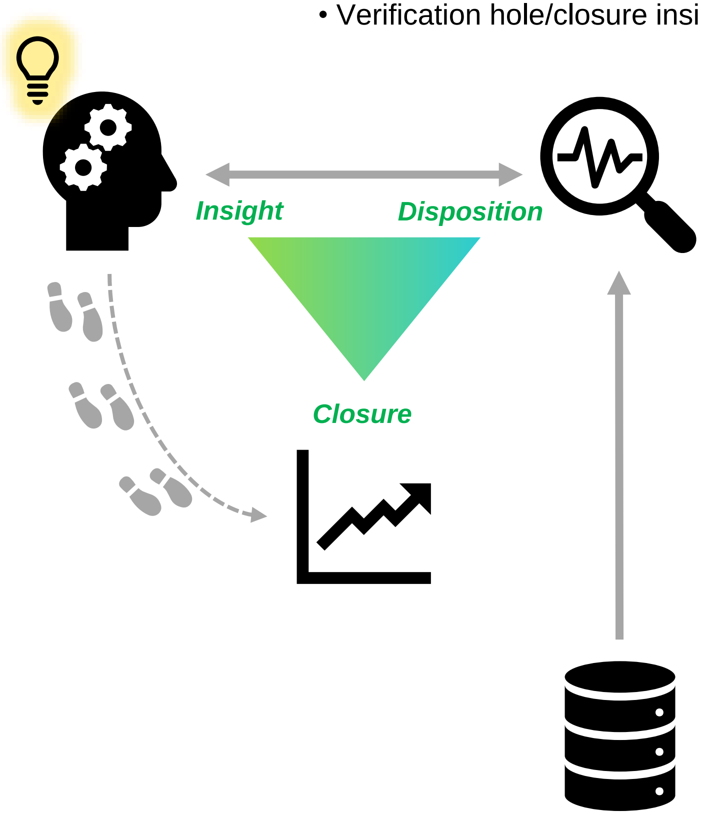
- **Safeness (S%)**
  - Unable to violate Safety Goal (SG)
  - Exhaustive fault analysis with **Jasper FSV App**
- **Diagnostic Coverage (DC%)**
  - Safety Mechanism (SM) performance
  - Simulation evidence with **Xcelium Safety**
    - Dangerous faults Detection (DD)
- Closure
  - Dedicated fault metric analysis
  - Insight for Workload/SM improvements
  - Disposition of the Not Classified faults



Features / Optimizations

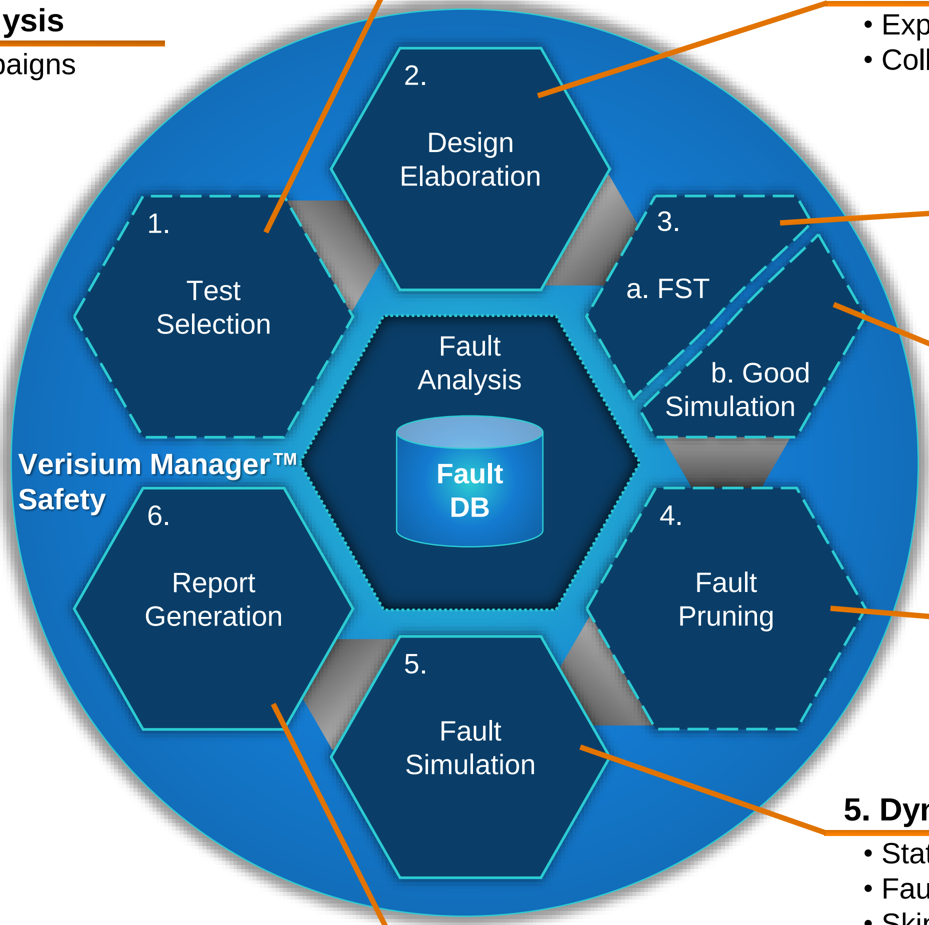
Dedicated Fault Metric Analysis

- Merge results across different campaigns
- Disposition of not-classified faults
- Verification hole/closure insight



Unified Fault Database

- Scalable to multi-millions of fault results
- Cross-engine data exchange



1. Test Ranking & Pruning

- Custom ranking criteria

2. Fault Instrumentation

- Expand user fault specification
- Collapsing and design-based testability

3.a. Structural Fault Reduction

- Strobe-based testability (safe faults)
- Advanced fault collapsing

3.b. Fault-free Simulation

- Required for serial engine and pruning
- Test-based activatability analysis

4. Test-based Fault Reduction

- Strobe-based propagability analysis
- Constant propagation analysis

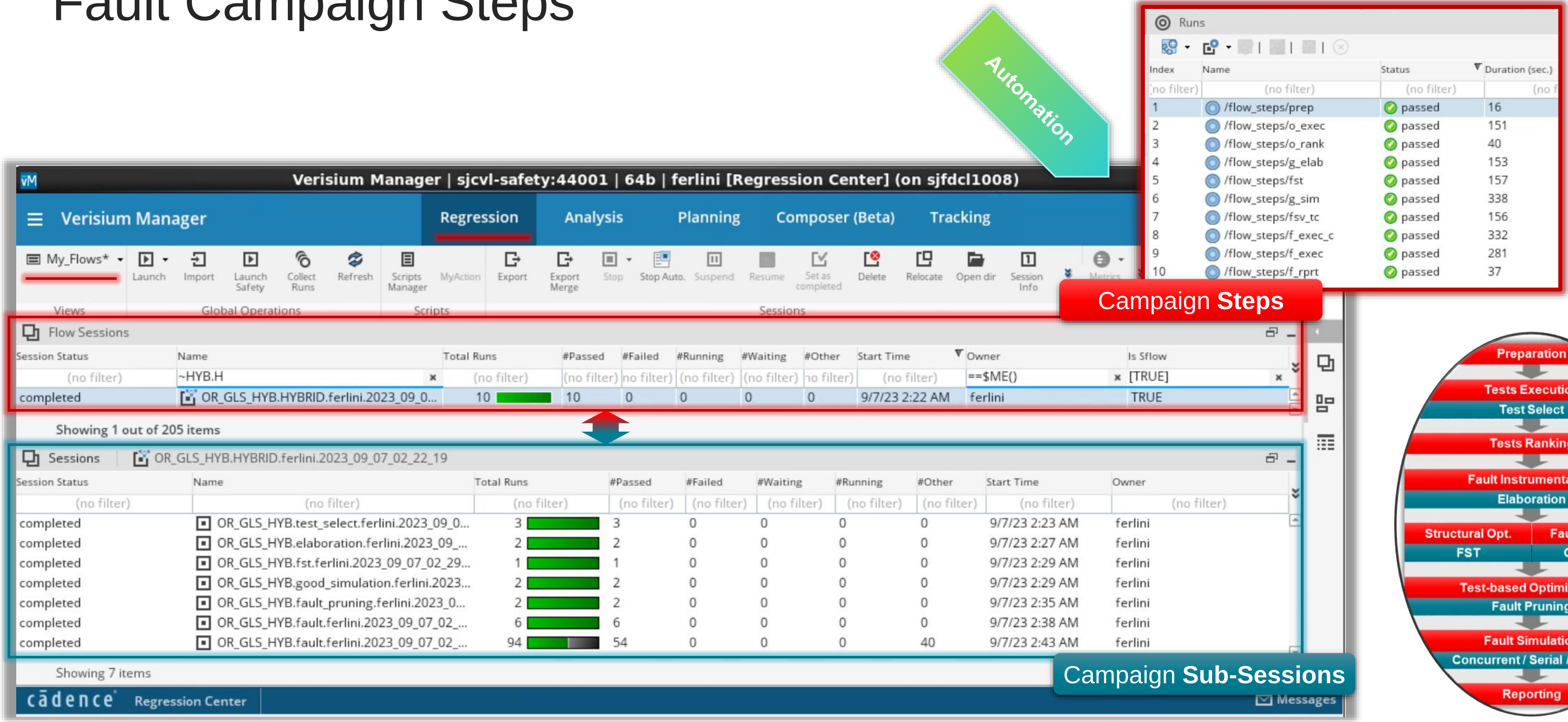
5. Dynamic Fault Simulation Control

- State-of-the-art run distribution manager
- Fault/Test Dropping, Timeout
- Skip pre-injection simulation

6. Reports Generation

- Campaign summary, annotated fault list
- FMEDA back-annotation results

Fault Campaign Steps



Runs

Index	Name	Status	Duration (sec.)
(no filter)	(no filter)	(no filter)	(no f
1	/flow_steps/prep	passed	16
2	/flow_steps/o_exec	passed	151
3	/flow_steps/o_rank	passed	40
4	/flow_steps/g_elab	passed	153
5	/flow_steps/fst	passed	157
6	/flow_steps/g_sim	passed	338
7	/flow_steps/fsv_tc	passed	156
8	/flow_steps/f_exec_c	passed	332
9	/flow_steps/f_exec	passed	281
10	/flow_steps/f_rprt	passed	37

Flow Sessions

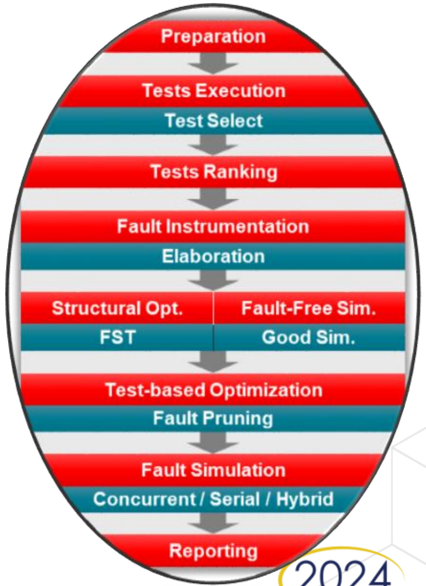
Session Status	Name	Total Runs	#Passed	#Failed	#Running	#Waiting	#Other	Start Time	Owner	Is Sflow
(no filter)	~HYB.H	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	==$ME()	[TRUE]
completed	OR_GLS_HYB.HYBRID.ferlini.2023_09_0...	10	10	0	0	0	0	9/7/23 2:22 AM	ferlini	TRUE

Showing 1 out of 205 items

Sessions OR_GLS_HYB.HYBRID.ferlini.2023_09_07_02_22_19

Session Status	Name	Total Runs	#Passed	#Failed	#Waiting	#Running	#Other	Start Time	Owner
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
completed	OR_GLS_HYB.test_select.ferlini.2023_09_0...	3	3	0	0	0	0	9/7/23 2:23 AM	ferlini
completed	OR_GLS_HYB.elaboration.ferlini.2023_09_...	2	2	0	0	0	0	9/7/23 2:27 AM	ferlini
completed	OR_GLS_HYB.fst.ferlini.2023_09_07_02_29...	1	1	0	0	0	0	9/7/23 2:29 AM	ferlini
completed	OR_GLS_HYB.good_simulation.ferlini.2023...	2	2	0	0	0	0	9/7/23 2:29 AM	ferlini
completed	OR_GLS_HYB.fault_pruning.ferlini.2023_0...	2	2	0	0	0	0	9/7/23 2:35 AM	ferlini
completed	OR_GLS_HYB.fault.ferlini.2023_09_07_02_...	6	6	0	0	0	0	9/7/23 2:38 AM	ferlini
completed	OR_GLS_HYB.fault.ferlini.2023_09_07_02_...	94	54	0	0	0	40	9/7/23 2:43 AM	ferlini

Showing 7 items

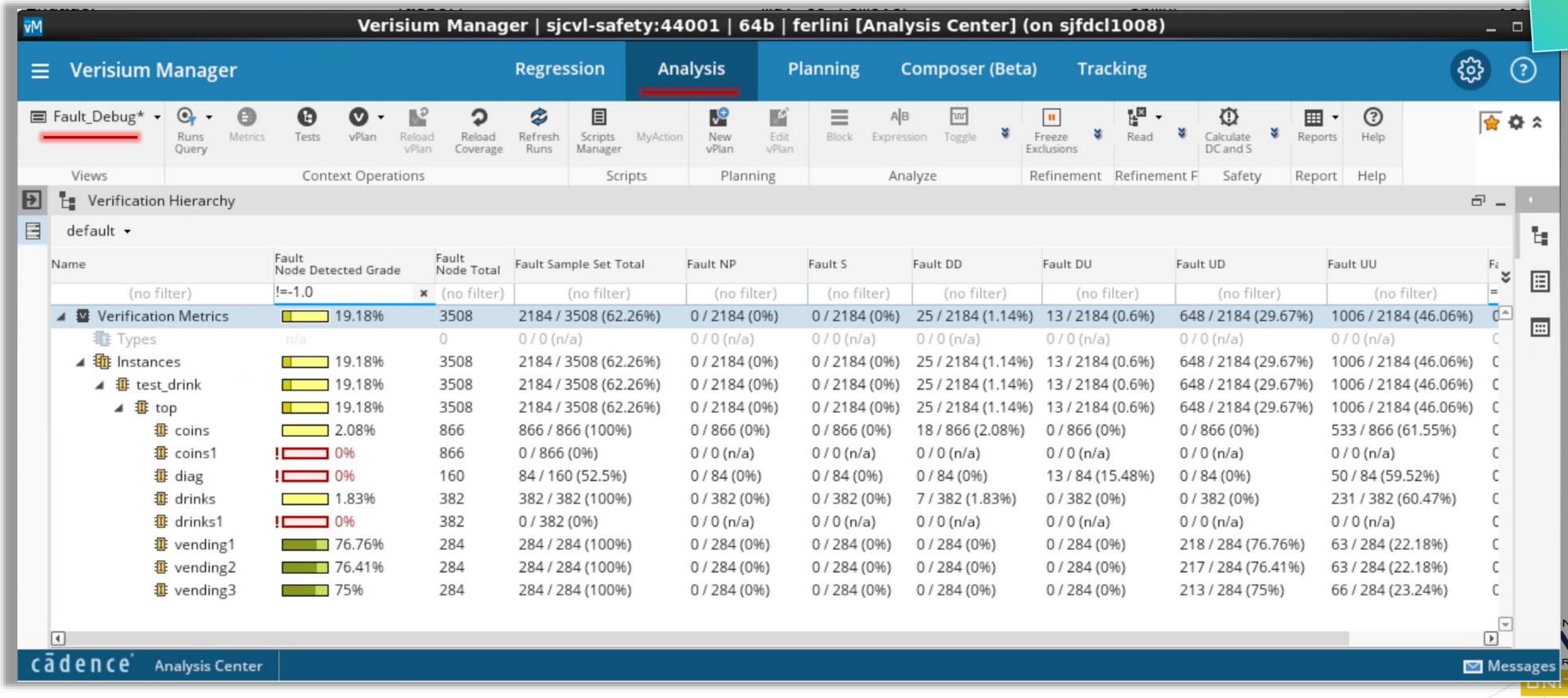


Preparation
Tests Execution
Test Select
Tests Ranking
Fault Instrumentation
Elaboration
Structural Opt.
FST
Fault-Free Sim.
Good Sim.
Test-based Optimization
Fault Pruning
Fault Simulation
Concurrent / Serial / Hybrid
Reporting

Dedicated Fault Analysis

Hierarchical Data

Analysis



Name	Fault Node Detected Grade	Fault Node Total	Fault Sample Set Total	Fault NP	Fault S	Fault DD	Fault DU	Fault UD	Fault UU	Fa
(no filter)	!=-1.0 ×	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	=
Verification Metrics	19.18%	3508	2184 / 3508 (62.26%)	0 / 2184 (0%)	0 / 2184 (0%)	25 / 2184 (1.14%)	13 / 2184 (0.6%)	648 / 2184 (29.67%)	1006 / 2184 (46.06%)	C
Types	n/a	0	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	C
Instances	19.18%	3508	2184 / 3508 (62.26%)	0 / 2184 (0%)	0 / 2184 (0%)	25 / 2184 (1.14%)	13 / 2184 (0.6%)	648 / 2184 (29.67%)	1006 / 2184 (46.06%)	C
test_drink	19.18%	3508	2184 / 3508 (62.26%)	0 / 2184 (0%)	0 / 2184 (0%)	25 / 2184 (1.14%)	13 / 2184 (0.6%)	648 / 2184 (29.67%)	1006 / 2184 (46.06%)	C
top	19.18%	3508	2184 / 3508 (62.26%)	0 / 2184 (0%)	0 / 2184 (0%)	25 / 2184 (1.14%)	13 / 2184 (0.6%)	648 / 2184 (29.67%)	1006 / 2184 (46.06%)	C
coins	2.08%	866	866 / 866 (100%)	0 / 866 (0%)	0 / 866 (0%)	18 / 866 (2.08%)	0 / 866 (0%)	0 / 866 (0%)	533 / 866 (61.55%)	C
coins1	! 0%	866	0 / 866 (0%)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	C
diag	! 0%	160	84 / 160 (52.5%)	0 / 84 (0%)	0 / 84 (0%)	0 / 84 (0%)	13 / 84 (15.48%)	0 / 84 (0%)	50 / 84 (59.52%)	C
drinks	1.83%	382	382 / 382 (100%)	0 / 382 (0%)	0 / 382 (0%)	7 / 382 (1.83%)	0 / 382 (0%)	0 / 382 (0%)	231 / 382 (60.47%)	C
drinks1	! 0%	382	0 / 382 (0%)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	C
vending1	76.76%	284	284 / 284 (100%)	0 / 284 (0%)	0 / 284 (0%)	0 / 284 (0%)	0 / 284 (0%)	218 / 284 (76.76%)	63 / 284 (22.18%)	C
vending2	76.41%	284	284 / 284 (100%)	0 / 284 (0%)	0 / 284 (0%)	0 / 284 (0%)	0 / 284 (0%)	217 / 284 (76.41%)	63 / 284 (22.18%)	C
vending3	75%	284	284 / 284 (100%)	0 / 284 (0%)	0 / 284 (0%)	0 / 284 (0%)	0 / 284 (0%)	213 / 284 (75%)	66 / 284 (23.24%)	C

Campaign Summary Report

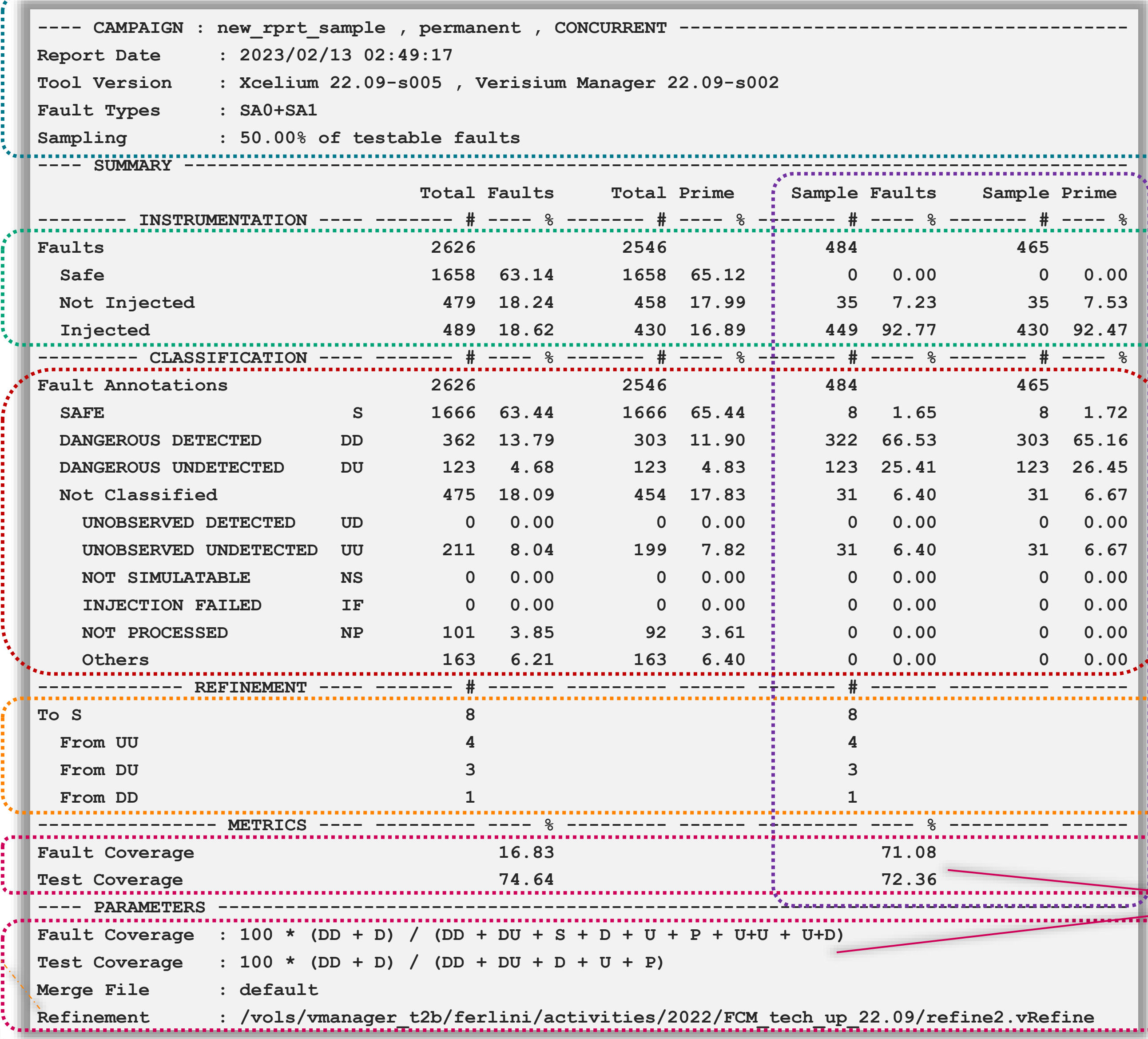
Date, tool version, fault types, sampling, …

Static instrumentation fault results

Overall campaign(s) merged results

Fault Disposition (user refinement)

Applicable client configuration



```
---- CAMPAIGN : new_rprt_sample , permanent , CONCURRENT ----------------------------------------
Report Date     : 2023/02/13 02:49:17
Tool Version    : Xcelium 22.09-s005 , Verisium Manager 22.09-s002
Fault Types     : SA0+SA1
Sampling        : 50.00% of testable faults
---- SUMMARY ------------------------------------------------------------------------------------
                                  Total Faults     Total Prime     Sample Faults    Sample Prime
-------- INSTRUMENTATION ---- ------- # ---- % ------- # ---- % ------- # ---- % ------- # ---- %
Faults                                 2626             2546              484              465
  Safe                                 1658  63.14      1658  65.12         0   0.00         0   0.00
  Not Injected                          479  18.24       458  17.99        35   7.23        35   7.53
  Injected                              489  18.62       430  16.89       449  92.77       430  92.47
--------- CLASSIFICATION ---- ------- # ---- % ------- # ---- % ------- # ---- % ------- # ---- %
Fault Annotations                      2626             2546              484              465
  SAFE                      S          1666  63.44      1666  65.44         8   1.65         8   1.72
  DANGEROUS DETECTED       DD           362  13.79       303  11.90       322  66.53       303  65.16
  DANGEROUS UNDETECTED     DU           123   4.68       123   4.83       123  25.41       123  26.45
  Not Classified                        475  18.09       454  17.83        31   6.40        31   6.67
    UNOBSERVED DETECTED    UD             0   0.00         0   0.00         0   0.00         0   0.00
    UNOBSERVED UNDETECTED  UU           211   8.04       199   7.82        31   6.40        31   6.67
    NOT SIMULATABLE        NS             0   0.00         0   0.00         0   0.00         0   0.00
    INJECTION FAILED       IF             0   0.00         0   0.00         0   0.00         0   0.00
    NOT PROCESSED          NP           101   3.85        92   3.61         0   0.00         0   0.00
    Others                              163   6.21       163   6.40         0   0.00         0   0.00
------------- REFINEMENT ---- ------- # ------ --------- ------ ------- # ------ --------- ------
To S                                  8                                  8
  From UU                             4                                  4
  From DU                             3                                  3
  From DD                             1                                  1
---------------- METRICS ---- --------- ---- % --------- ------ --------- ---- % --------- ------
Fault Coverage                           16.83                             71.08
Test Coverage                            74.64                             72.36
---- PARAMETERS ---------------------------------------------------------------------------------
Fault Coverage  : 100 * (DD + D) / (DD + DU + S + D + U + P + U+U + U+D)
Test Coverage   : 100 * (DD + D) / (DD + DU + D + U + P)
Merge File      : default
Refinement      : /vols/vmanager_t2b/ferlini/activities/2022/FCM_tech_up_22.09/refine2.vRefine
```

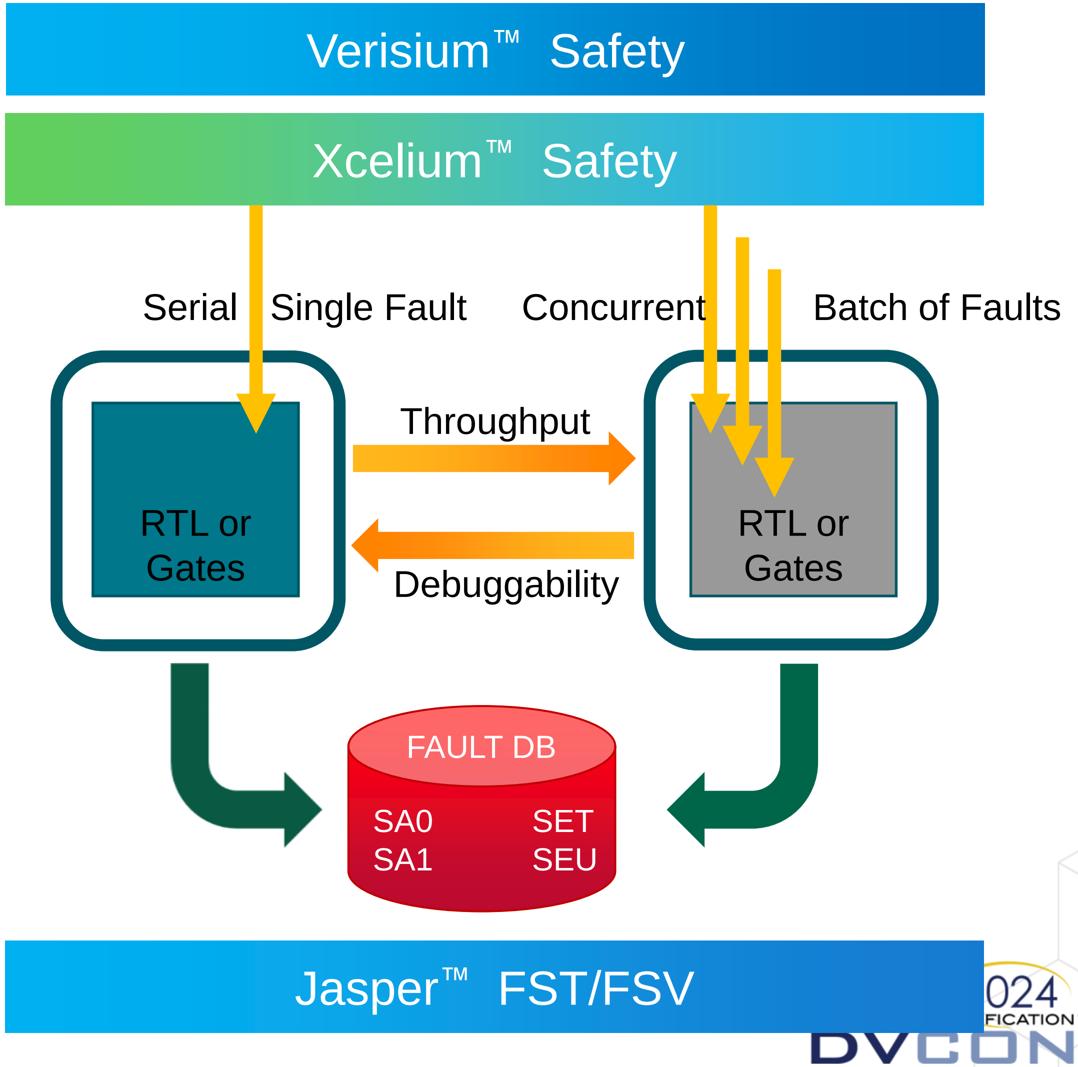
Sampled fault scope

Fault and Test Coverage results and formulas

Fault Campaign Management – Safety Engines

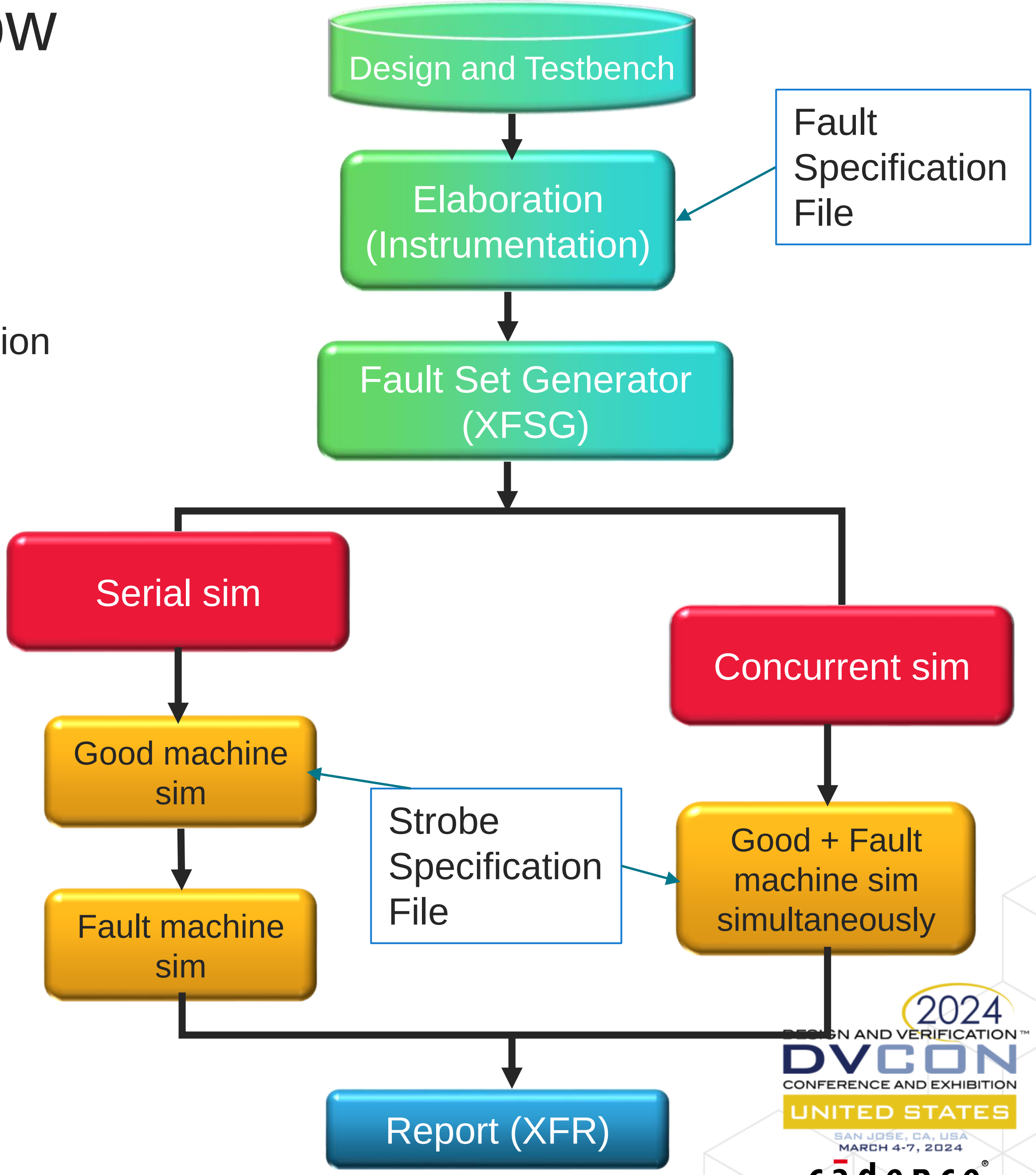
Xcelium Safety App

- The Xcelium Safety App provides native fault simulation by integrating Functional & Safety Engines
- Supports existing Xcelium Methodologies
  - Capture Replay, DSS (Dual Snapshot), Save Restore
- The Xcelium Safety App operates in 2 modes:
  - Serial mode: Flow setup and Debug
  - Concurrent mode:
    - Higher throughput
    - 5-100x faster than serial
    - Handles 2K to 20K faults in a single run (Single CPU Core)
- Supports Random Sampling as Sampling Percentage, Sampling Number
- Support Dual Strobe, Single Strobe Fault Classification
- Interoperable serial and concurrent fault simulation engines
- Both modes have identical flow and can easily switch back and forth
- The Xcelium Safety App simulates & annotates all faults in the fault DB
- Supports Fault Boundary to limit CoPF (Cone of Fault Propagation)



Xcelium Safety Overview – User Flow

- Easy to migrate from Functional verification flows to Fault Injection
  - Additional file (Fault File) and option to be added to Elaboration
  - Elaboration has added steps for fault Instrumentation
- Fault Simulations
  - Serial Engine or
  - Concurrent Engine
- Hybrid Mode Support
  - Xcelium Safety Simulation allows for users to run the hybrid flow where Concurrent followed by Serial
- Reporting
  - Standalone Support Available


Design and Testbench
Elaboration (Instrumentation)
Fault Specification File
Fault Set Generator (XFSG)
Serial sim
Concurrent sim
Good machine sim
Fault machine sim
Strobe Specification File
Good + Fault machine sim simultaneously
Report (XFR)

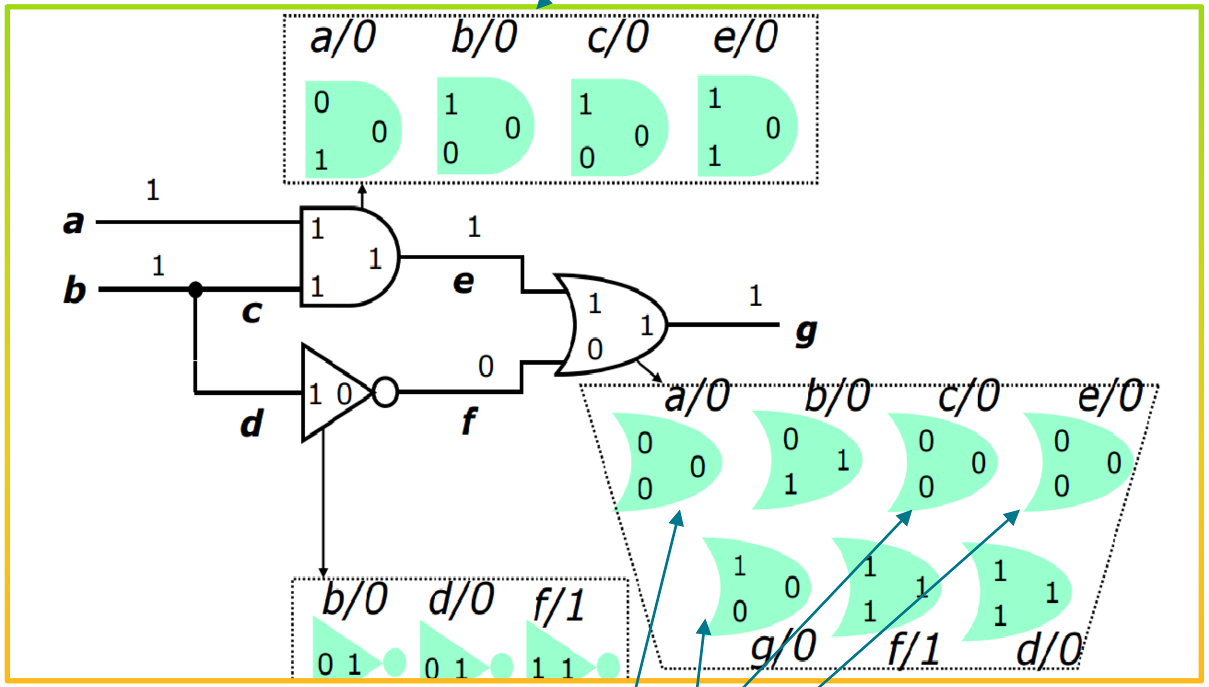
2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES SAN JOSE, CA, USA MARCH 4-7, 2024

cadence®

Xcelium Safety – Concurrent

- Inject and simulate multiple faults together
- Concurrent fault simulation is a throughput solution
  - Allows injection of multiple faults during simulation in a single run.
  - Better throughput than the serial engine.
  - New simulation kernel
    - new scheduler
    - fault management
- Native Integration with Xcelium Engine
- Good Simulation runs along with Fault Simulation
  - Fault Value diverges then simulation continues, or fault is dropped
  - Multiple fault runs concurrently in a single simulation (corresponding test vector)
  - Single CPU core per simulation, no multi core multi thread support

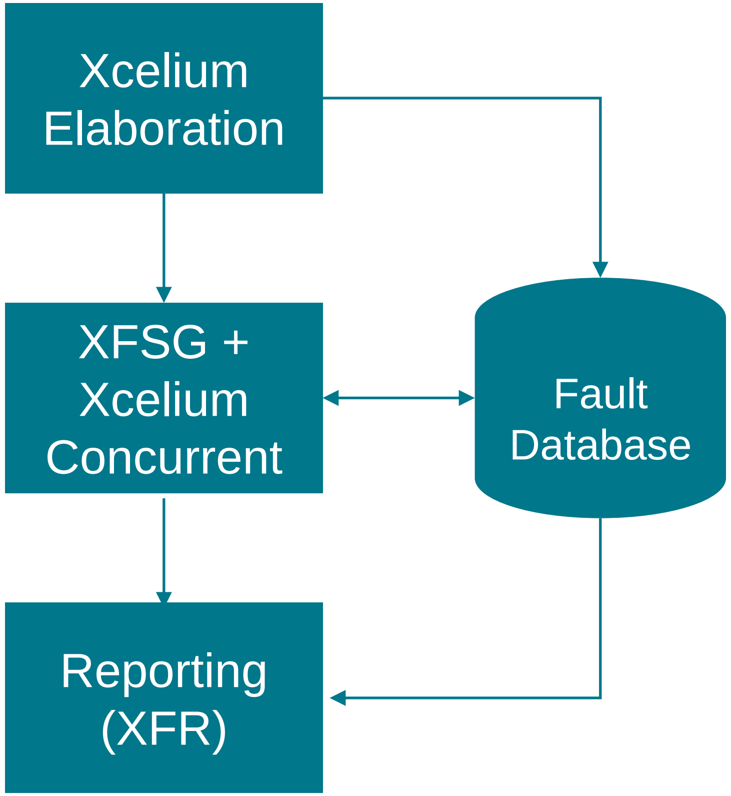
At least one value differs Good Value

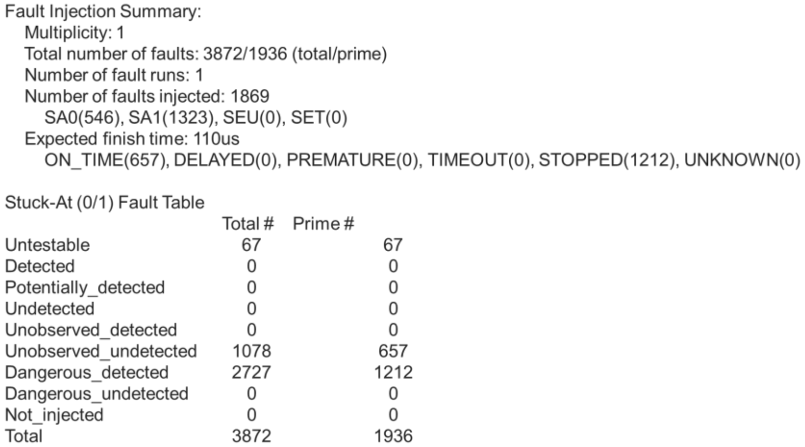


Incorrect Output Values in this list indicate detectable faults i.e., a/0, c/0, e/0, g/0

Concurrent Fault Simulation - Overview

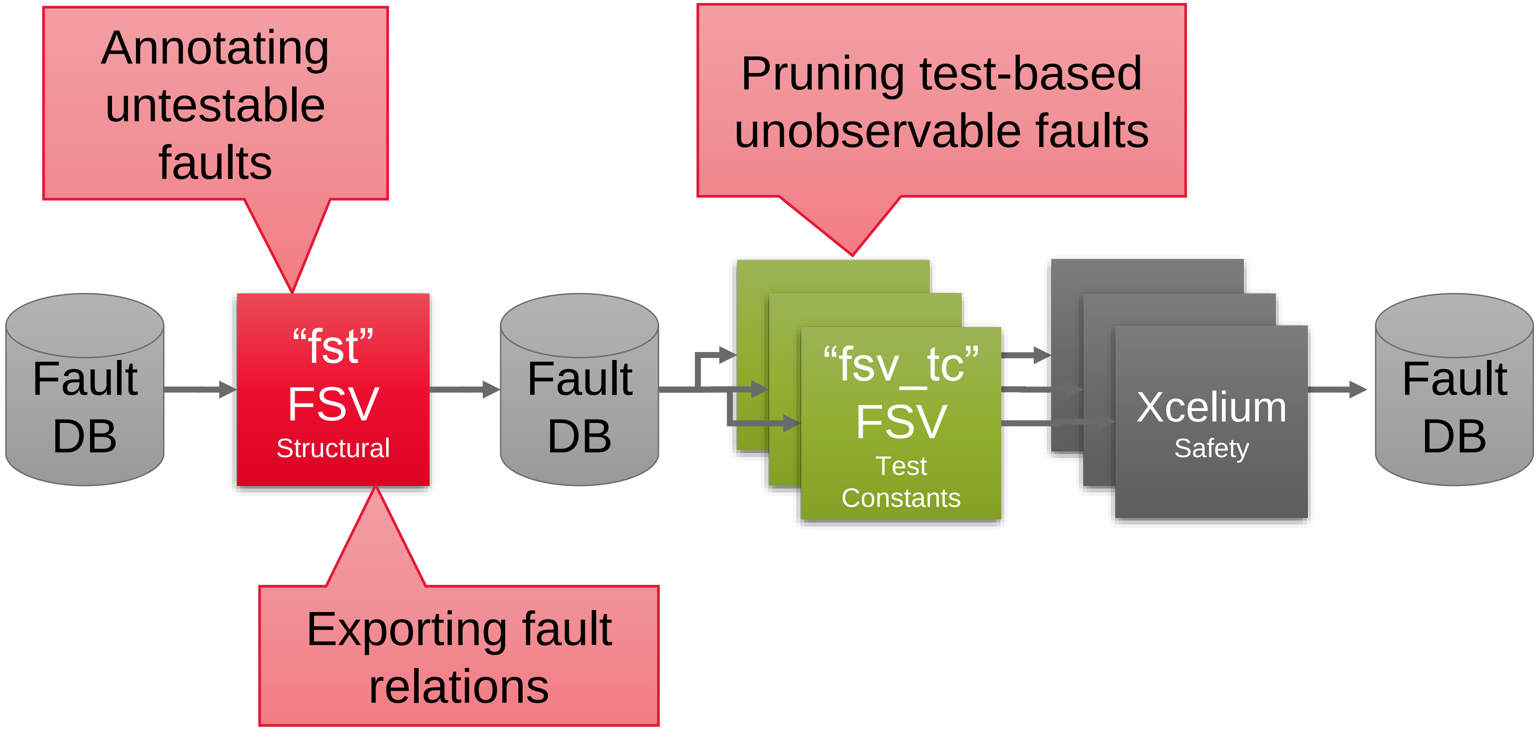
- Xcelium Elaboration
  - Used for Fault Instrumentation
  - Extra analysis done for Concurrent
  - Example:
    - xrun –fault_file <input_fault_file> -fault_rtl
- XFSG
  - xfsg -fault_work ./fault_db/ -fault_type sa0+sa1 –fault_list foutput –fault_spilt_size <number of Faults>
- Xcelium Concurrent Run
  - Run the injected faults in concurrent mode
  - Each fault simulated independent of the others
  - Example:
    - xrun –fault_concurrent –input <injected_fault_list>
- Reporting
  - Separate utility to generate fault report (Xfr)
  - Example:
    - xfr –fault_work <path_to_fault_database>

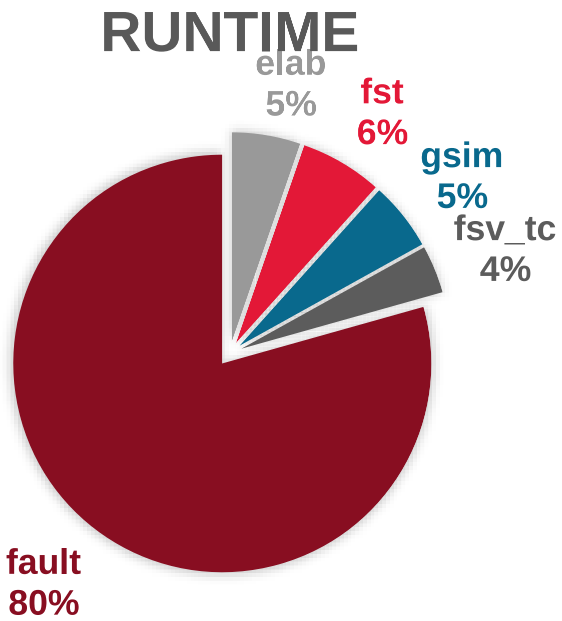




```
Fault Injection Summary:
   Multiplicity: 1
   Total number of faults: 3872/1936 (total/prime)
   Number of fault runs: 1
   Number of faults injected: 1869
      SA0(546), SA1(1323), SEU(0), SET(0)
   Expected finish time: 110us
      ON_TIME(657), DELAYED(0), PREMATURE(0), TIMEOUT(0), STOPPED(1212), UNKNOWN(0)

Stuck-At (0/1) Fault Table
                          Total #   Prime #
Untestable                  67         67
Detected                     0          0
Potentially_detected         0          0
Undetected                   0          0
Unobserved_detected          0          0
Unobserved_undetected     1078        657
Dangerous_detected        2727       1212
Dangerous_undetected         0          0
Not_injected                 0          0
Total                     3872       1936
```

FCM – Optimizations from Jasper Safety (FSV)


Annotating untestable faults
Pruning test-based unobservable faults
Fault DB
"fst"
FSV
Structural
Fault DB
"fsv_tc"
FSV
Test Constants
Xcelium
Safety
Fault DB
Exporting fault relations



A few minutes of optimization can save hours of simulation

- FSV exports fault relations → equivalent faults will be skipped
- FSV annotates untestable faults → Safe faults will be ignored
- FSV annotates faults as unobservable by test → Pruned faults will be dropped

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES SAN JOSE, CA, USA MARCH 4-7, 2024

cadence®

Jasper Functional Safety Verification App (FSV)

FSV Structural Fault Analysis

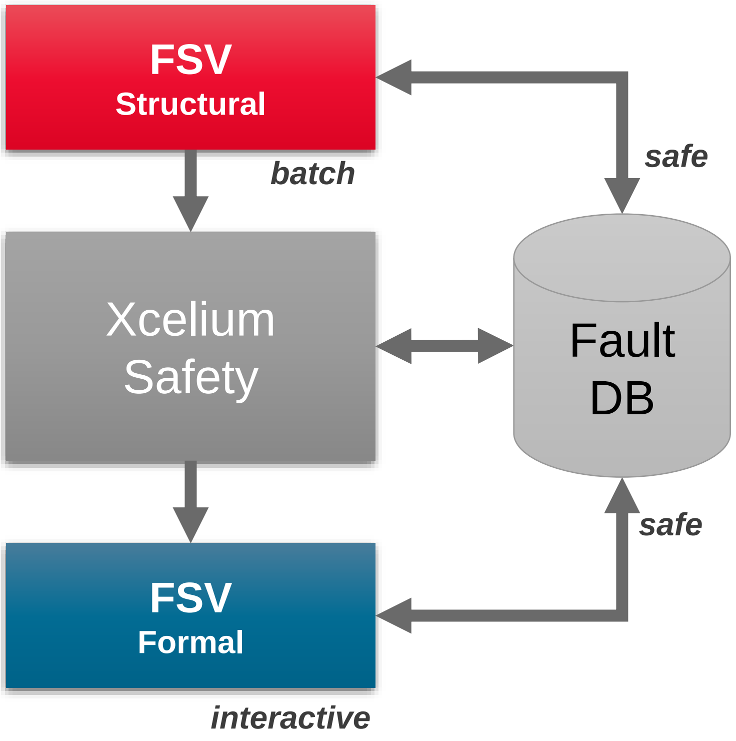
- Structural fault connectivity, activatability and relation analysis
- Highly automated pre-qualification flow for Xcelium Safety
- Reduces number and runtime of fault simulations

FSV Formal Fault Analysis

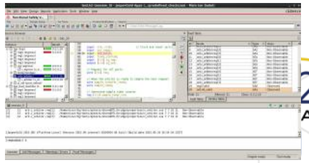
- Formal activatability and propagatability analysis
- Interactive debug, schematics and visualization of propagation
- Assists fault analysis sign-off with Xcelium Safety

FSV Custom Safety and Security Analysis

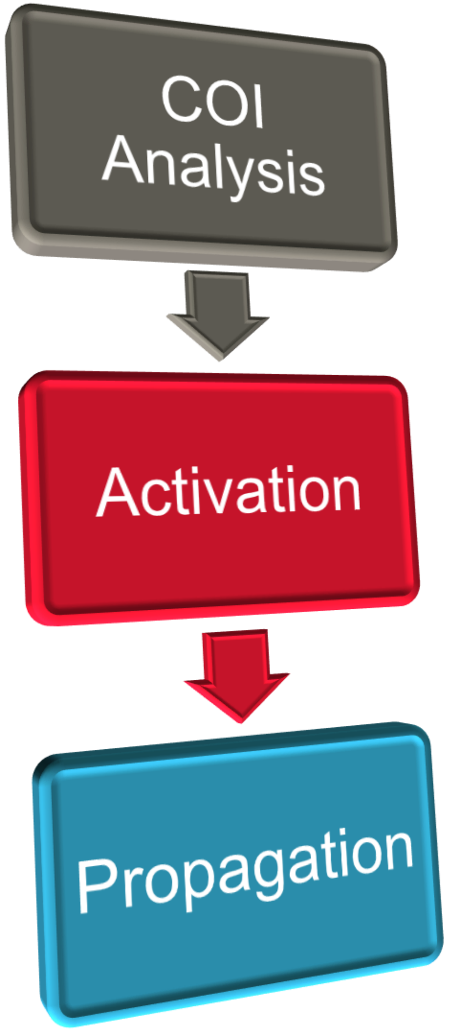
- Custom strobes and faults specification to model hacker attacks
- Advanced formal checks, barriers and multiplicity of faults
- Addresses safety and security hardware qualification



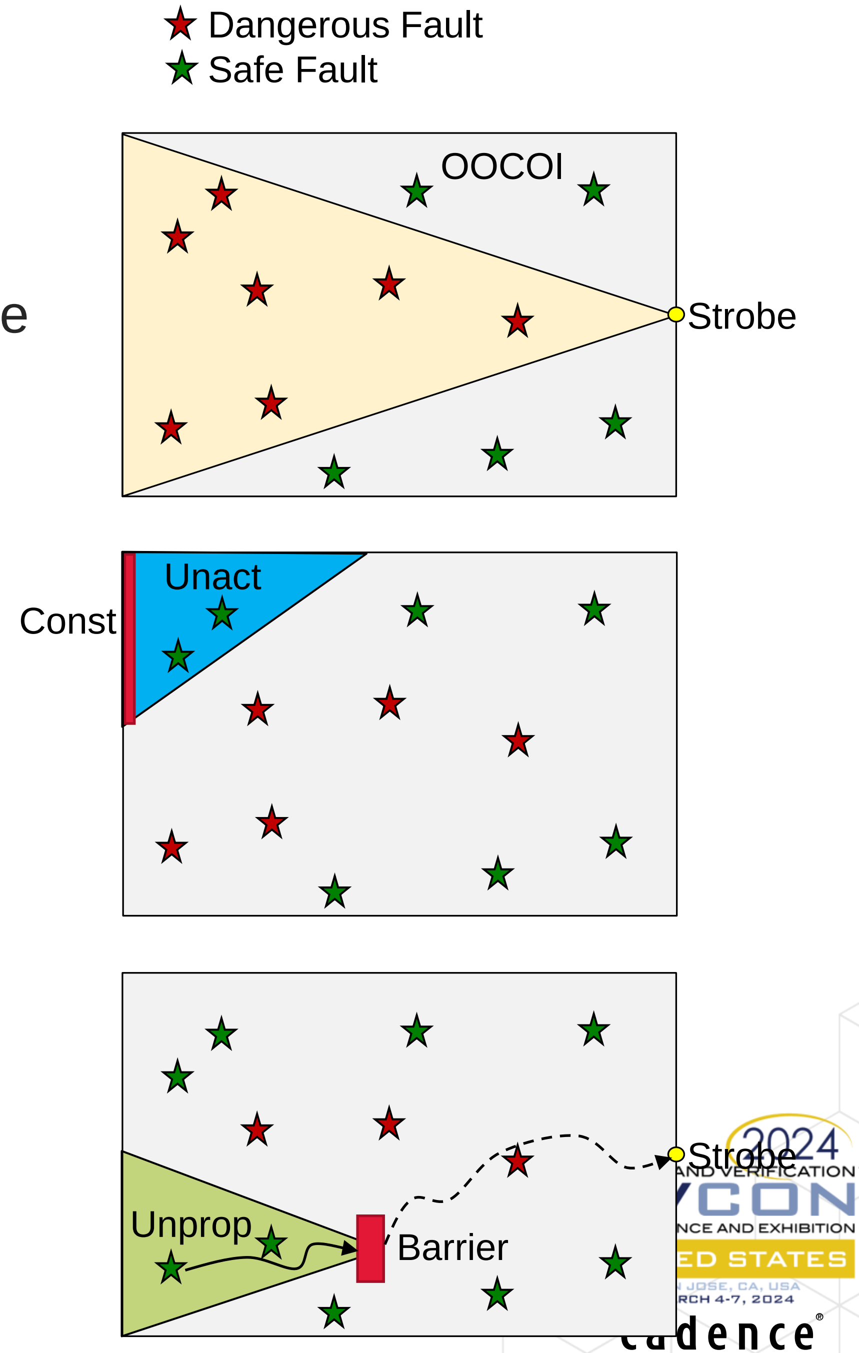
FSV
Custom



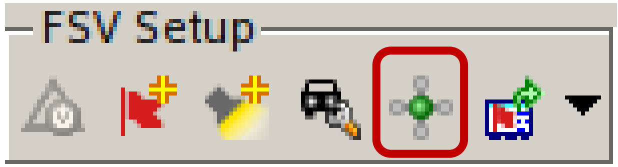
FSV Structural Analysis Check Types

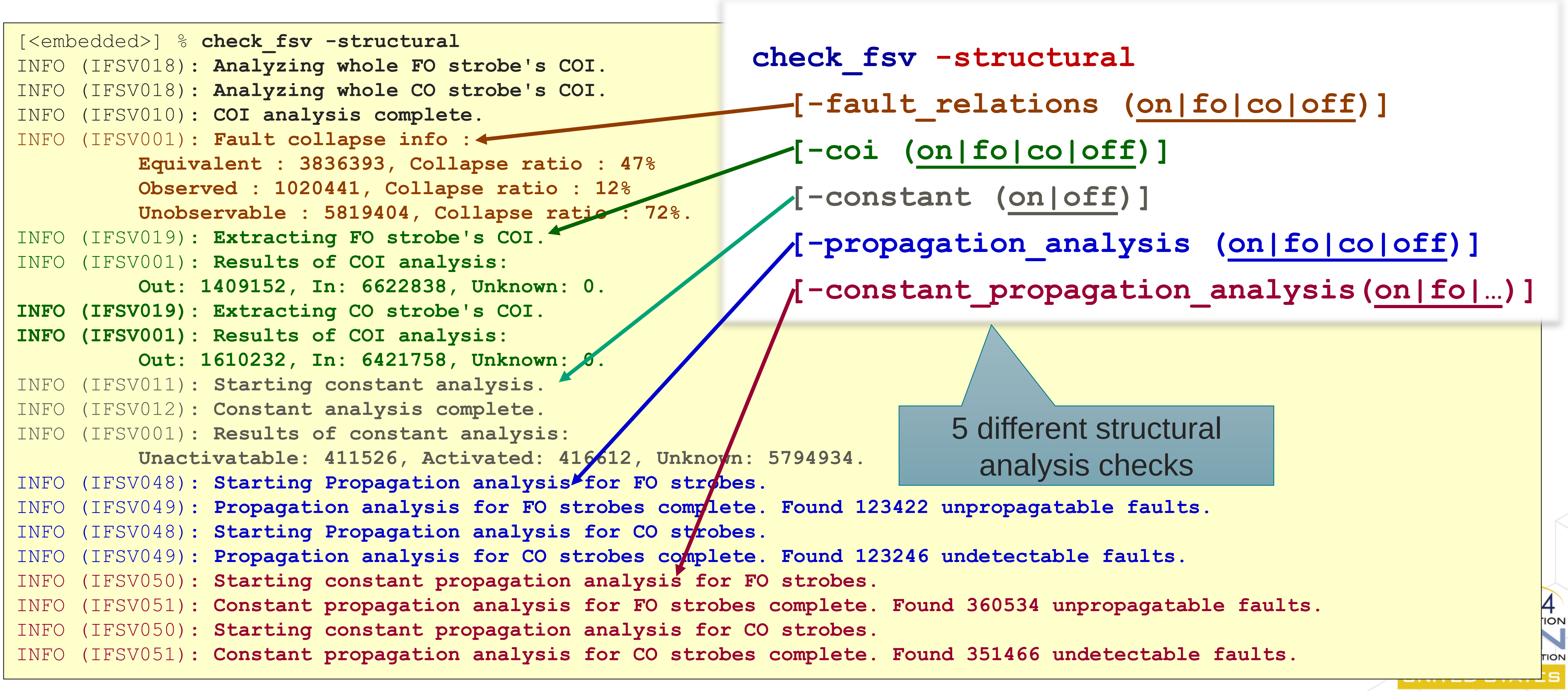


- Out-of-COI Analysis
  - A fault node outside the Cone-of-Influence (COI) has no physical connection to the functional strobe(s)
  - Fault is Out-of-COI = Safe
- Activatability Analysis
  - A SA0/1 fault injected on a node which is constant 0/1 cannot be activated
  - Fault is Unactivatable = Safe
- Propagatability Analysis
  - A fault that is activated and in COI, but cannot propagate to the functional strobe
  - Fault is Unpropagatable = Safe



FSV – Structural Analysis





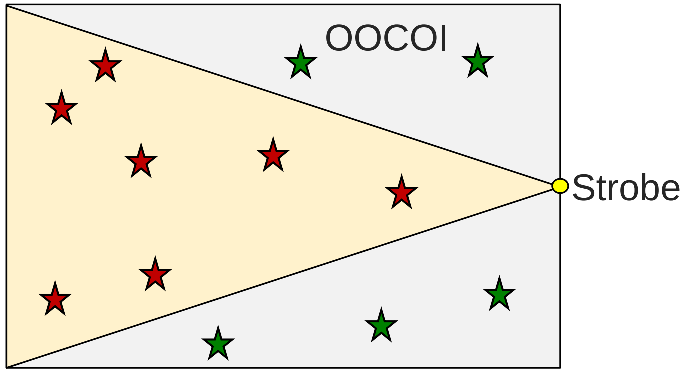
```
[<embedded>] % check_fsv -structural
INFO (IFSV018): Analyzing whole FO strobe's COI.
INFO (IFSV018): Analyzing whole CO strobe's COI.
INFO (IFSV010): COI analysis complete.
INFO (IFSV001): Fault collapse info :
          Equivalent : 3836393, Collapse ratio : 47%
          Observed : 1020441, Collapse ratio : 12%
          Unobservable : 5819404, Collapse ratio : 72%.
INFO (IFSV019): Extracting FO strobe's COI.
INFO (IFSV001): Results of COI analysis:
          Out: 1409152, In: 6622838, Unknown: 0.
INFO (IFSV019): Extracting CO strobe's COI.
INFO (IFSV001): Results of COI analysis:
          Out: 1610232, In: 6421758, Unknown: 0.
INFO (IFSV011): Starting constant analysis.
INFO (IFSV012): Constant analysis complete.
INFO (IFSV001): Results of constant analysis:
          Unactivatable: 411526, Activated: 416612, Unknown: 5794934.
INFO (IFSV048): Starting Propagation analysis for FO strobes.
INFO (IFSV049): Propagation analysis for FO strobes complete. Found 123422 unpropagatable faults.
INFO (IFSV048): Starting Propagation analysis for CO strobes.
INFO (IFSV049): Propagation analysis for CO strobes complete. Found 123246 undetectable faults.
INFO (IFSV050): Starting constant propagation analysis for FO strobes.
INFO (IFSV051): Constant propagation analysis for FO strobes complete. Found 360534 unpropagatable faults.
INFO (IFSV050): Starting constant propagation analysis for CO strobes.
INFO (IFSV051): Constant propagation analysis for CO strobes complete. Found 351466 undetectable faults.
```

```
check_fsv -structural
  [-fault_relations (on|fo|co|off)]
  [-coi (on|fo|co|off)]
  [-constant (on|off)]
  [-propagation_analysis (on|fo|co|off)]
  [-constant_propagation_analysis(on|fo|…)]
```

5 different structural analysis checks

FSV – Structural Analysis – COI Analysis

- COI Analysis
  - Goal is to produce „safe“ fault results
  - Sensitive to strobes and design
  - Scales up to multi-million gates



```
INFO (IFSV019): Extracting FO strobe's COI.
INFO (IFSV001): Results of COI analysis:
        Out: 1409152, In: 6622838, Unknown: 27.

INFO (IFSV019): Extracting CO strobe's COI.
INFO (IFSV001): Results of COI analysis:
        Out: 1610232, In: 6421758, Unknown: 27.
```

Out of COI FO means “safe”!

Out of COI CO means nothing for safeness. But it could be used for qualifying the diagnostic safety mechanism.

Unknown: Black boxes prevent deterministic results

FSV – Structural Analysis – Sequential Constant Propagation

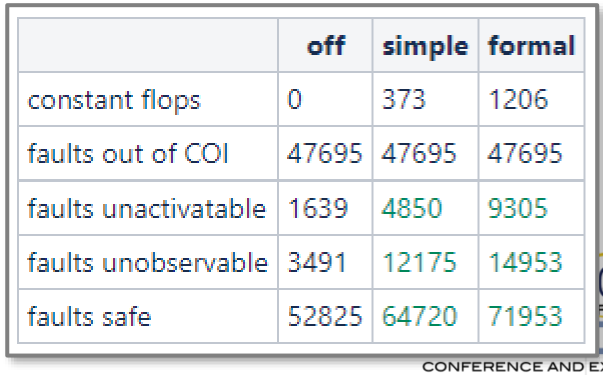
```
assume scan_en==0 -env
```

- FSV Structural Analysis benefits from constants in the design
  - But regular constant propagation stops at flops/latches!
- FSV runs sequential constant propagation in the beginning of structural analysis
  - Design constants are propagated through sequential elements using reset, clock and design constraints
  - **`set_fsv_structural_seq_constants_propagation ( off | simple | formal )`**
    - **`off`**: no analysis
    - **`simple`**: using fast proof simplification only (default)
    - **`formal`**: using regular prove engines* and associated time limits

* formal requires FSV license

- Benefit
  - More propagated constants, more structurally safe faults
- Note: Only environment constraints (-env) are respected!
  - Task based constants are ignored in structural analysis

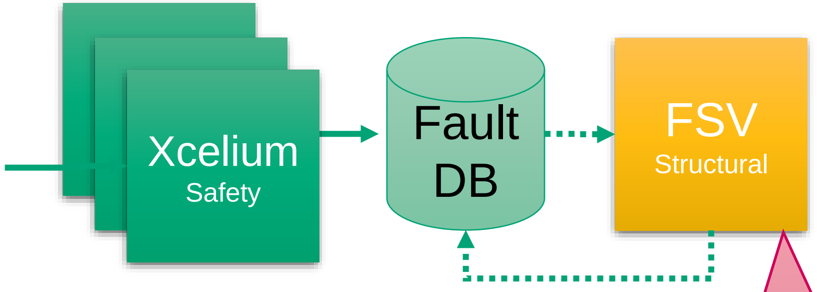
Example customer testcase



	off	simple	formal
constant flops	0	373	1206
faults out of COI	47695	47695	47695
faults unactivatable	1639	4850	9305
faults unobservable	3491	12175	14953
faults safe	52825	64720	71953

FSV – Structural UU Disposition Post-Fault Simulation

- Try hard to find more SAFE faults in a fault injection campaign with many Unclassified faults (UU, UD)
  - Reduces the % of UU/UD fault
- By adding custom constraints and barriers
  - Declare UU/UD as SAFE!
  - Analysis of remaining UU/UD proposes additional test sequences in XFS to turn UU into DD or DU



Annotating unobserved faults as SAFE

UU/UD faults = unclassified
DU/DD/S faults = classified
Confidence of SPFM/ASIL:

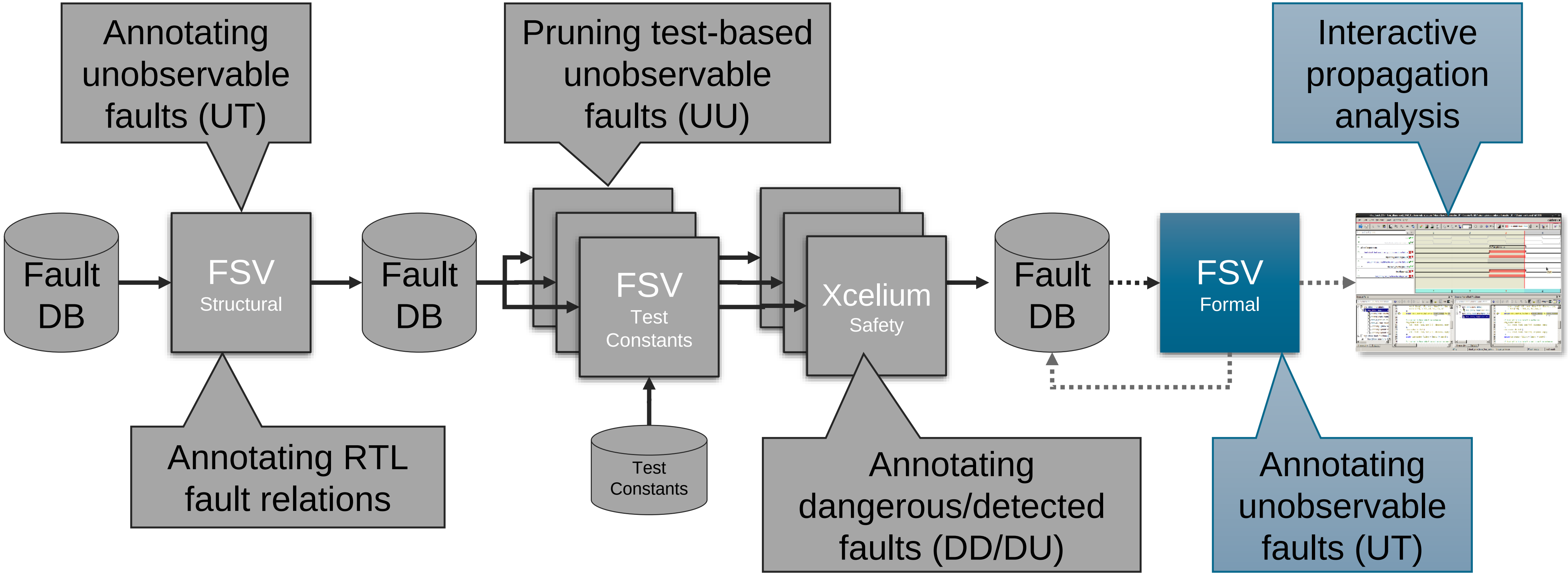
SPFM 99.3% with 50% UU – low
SPFM 99.1% with 2% UU – high

$$DC = \frac{\lambda_{DD}}{\lambda_{DD} + \lambda_{DU}}$$

$$SPFM = \frac{\lambda_{DD} + \lambda_{S}}{\lambda_{DD} + \lambda_{DU} + \lambda_{S}}$$

SPFM	ASIL
>= 99%	D
>= 97%	C
>= 90%	B
< 90%	A

FSV Integration with Xcelium Safety Simulator


Annotating unobservable faults (UT)
Pruning test-based unobservable faults (UU)
Interactive propagation analysis
Fault DB
FSV Structural
Fault DB
FSV Test Constants
Xcelium Safety
Fault DB
FSV Formal
Test Constants
Annotating RTL fault relations
Annotating dangerous/detected faults (DD/DU)
Annotating unobservable faults (UT)

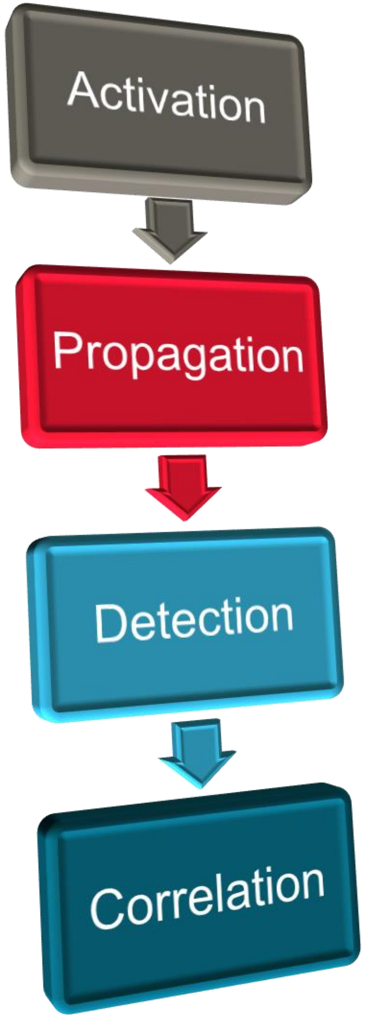
- FSV Structural automatically annotates unobservable faults and RTL fault relations in database
- FSV TC prunes faults not exercisable by particular simulation test
- Xcelium Safety simulates and annotates all remaining faults in database
- FSV Formal annotates unobservable faults and provides interactive propagation analysis

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES SAN JOSE, CA, USA MARCH 4-7, 2024

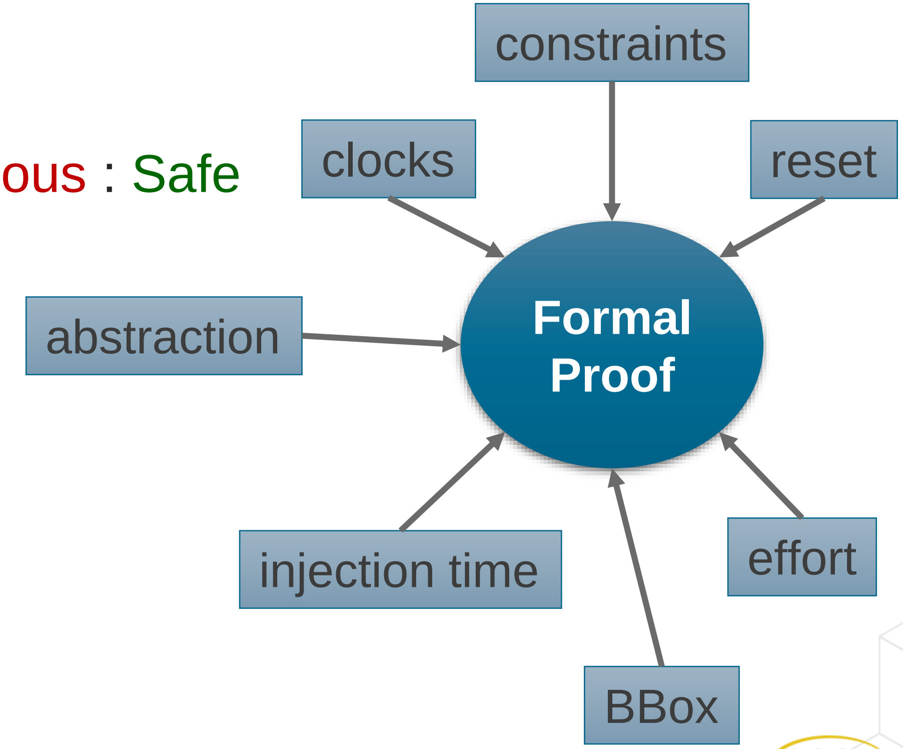
cādence®

FSV Formal Analysis Check Types

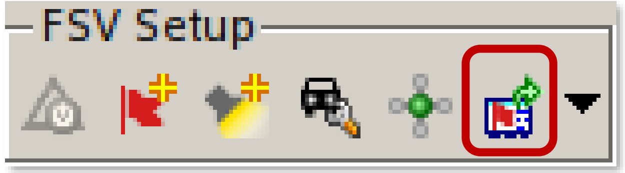
FSV
Formal



- Activation Analysis
  - Can the fault be functionally activated from the inputs? No = Safe
- Propagation Analysis
  - Can the fault propagate to FO? Dangerous : Safe
  - Will it always propagate to FO?
- Detection Analysis
  - Can the fault be detected at the CO?
  - Will it always be detected at the CO?
- Correlation Analysis
  - Will a propagated fault always be detected?



Generating Properties



- FSV can generate 6 types of formal properties

```
check_fsv -generate [-id <tcl_list>] [-task <name>]
      [-activatability (on|off)]
      [-propagatability (on|off)]
      [-detectability (on|off)]
      [-always_propagated (on|off)]
      [-always_detected (on|off)]
      [-propagated_always_detected (on|off)]
```

6 different formal analysis checks

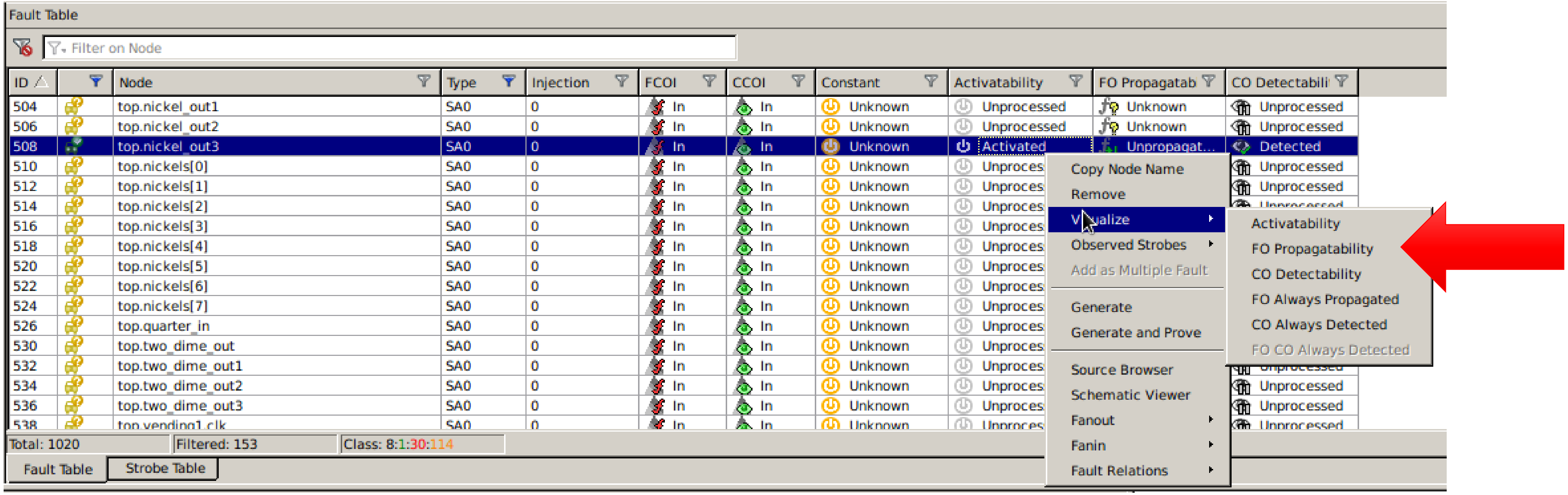
Easiest, runs in an optimized region

Most difficult, yet most meaningful check

```
INFO (IFSV007): Starting generate.
INFO (IFSV002): Mapping X and undriven.
INFO (IFSV004): Analyzing non-resettable regs.
INFO (IFSV005): Completed analyzing non-resettable regs.
INFO (IFSV013): Creating <fsv_task_0>.
INFO (IFSV016): Modeling task <fsv_task_0> faults.
INFO (IFSV003): Mapping non-resettable regs.
…
INFO (IFSV014): Task <fsv_task_0> created.
INFO (IFSV009): Generate completed.
```

FSV Formal – Debugging Visualize Waveforms

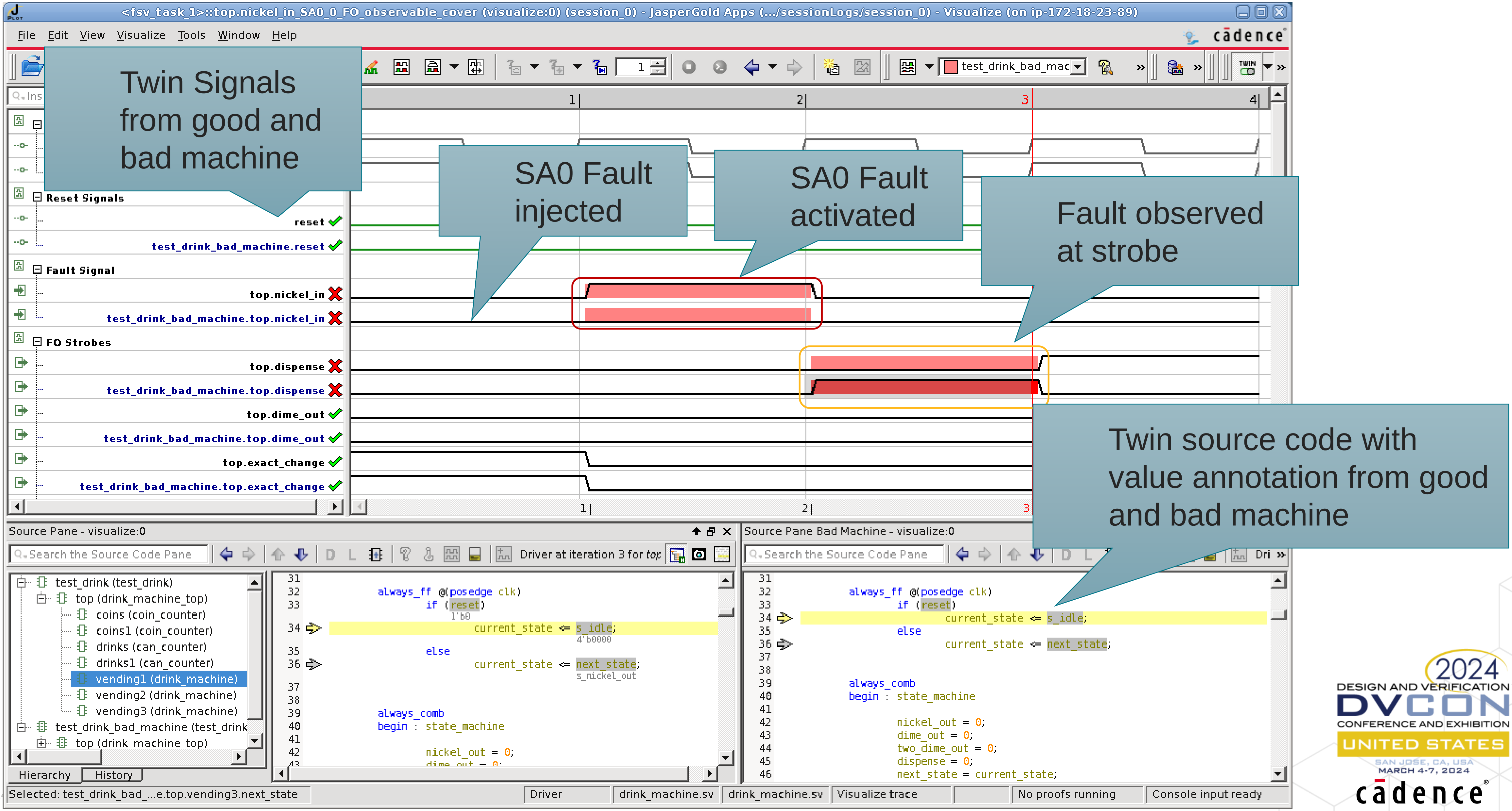
- Visualize for detection traces and unobservable analysis
  - Use Right-Mouse-Button Menu over an item in the Fault Table



ID		Node	Type	Injection	FCOI	CCOI	Constant	Activatability	FO Propagatab	CO Detectabili
504		top.nickel_out1	SA0	0	In	In	Unknown	Unprocessed	Unknown	Unprocessed
506		top.nickel_out2	SA0	0	In	In	Unknown	Unprocessed	Unknown	Unprocessed
508		top.nickel_out3	SA0	0	In	In	Unknown	Activated	Unpropagat...	Detected
510		top.nickels[0]	SA0	0	In	In	Unknown	Unproces		Unprocessed
512		top.nickels[1]	SA0	0	In	In	Unknown	Unproces		Unprocessed
514		top.nickels[2]	SA0	0	In	In	Unknown	Unproces		Unprocessed
516		top.nickels[3]	SA0	0	In	In	Unknown	Unproces		
518		top.nickels[4]	SA0	0	In	In	Unknown	Unproces		
520		top.nickels[5]	SA0	0	In	In	Unknown	Unproces		
522		top.nickels[6]	SA0	0	In	In	Unknown	Unproces		
524		top.nickels[7]	SA0	0	In	In	Unknown	Unproces		
526		top.quarter_in	SA0	0	In	In	Unknown	Unproces		
530		top.two_dime_out	SA0	0	In	In	Unknown	Unproces		
532		top.two_dime_out1	SA0	0	In	In	Unknown	Unproces		Unprocessed
534		top.two_dime_out2	SA0	0	In	In	Unknown	Unproces		Unprocessed
536		top.two_dime_out3	SA0	0	In	In	Unknown	Unproces		Unprocessed
538		top.vending1.clk	SA0	0	In	In	Unknown	Unproces		Unprocessed

Total: 1020 Filtered: 153 Class: 8:1:30:114

FSV Formal – Visualize Fault Detection Traces



<fsv_task_1>::top.nickel_in_SA0_0_FO_observable_cover (visualize:0) (session_0) - JasperGold Apps (.../sessionLogs/session_0) - Visualize (on ip-172-18-23-89)

File Edit View Visualize Tools Window Help
cadence
test_drink_bad_mac
1
1
2
3
4
Reset Signals
reset ✓
test_drink_bad_machine.reset ✓
Fault Signal
top.nickel_in ✗
test_drink_bad_machine.top.nickel_in ✗
FO Strobes
top.dispense ✗
test_drink_bad_machine.top.dispense ✗
top.dime_out ✓
test_drink_bad_machine.top.dime_out ✓
top.exact_change ✓
test_drink_bad_machine.top.exact_change ✓
Source Pane - visualize:0
Search the Source Code Pane
Driver at iteration 3 for top
Source Pane Bad Machine - visualize:0
Search the Source Code Pane
Dri
Hierarchy
History

Twin Signals from good and bad machine

SA0 Fault injected

SA0 Fault activated

Fault observed at strobe

Twin source code with value annotation from good and bad machine

```
test_drink (test_drink)
  top (drink_machine_top)
    coins (coin_counter)
    coins1 (coin_counter)
    drinks (can_counter)
    drinks1 (can_counter)
    vending1 (drink_machine)
    vending2 (drink_machine)
    vending3 (drink_machine)
test_drink_bad_machine (test_drink
  top (drink_machine_top)
```

```
        always_ff @(posedge clk)
           if (reset)
              1'b0
              current_state <= s_idle;
                               4'b0000

           else
              current_state <= next_state;
                               s_nickel_out


        always_comb
        begin : state_machine

           nickel_out = 0;
           dime_out = 0;
```

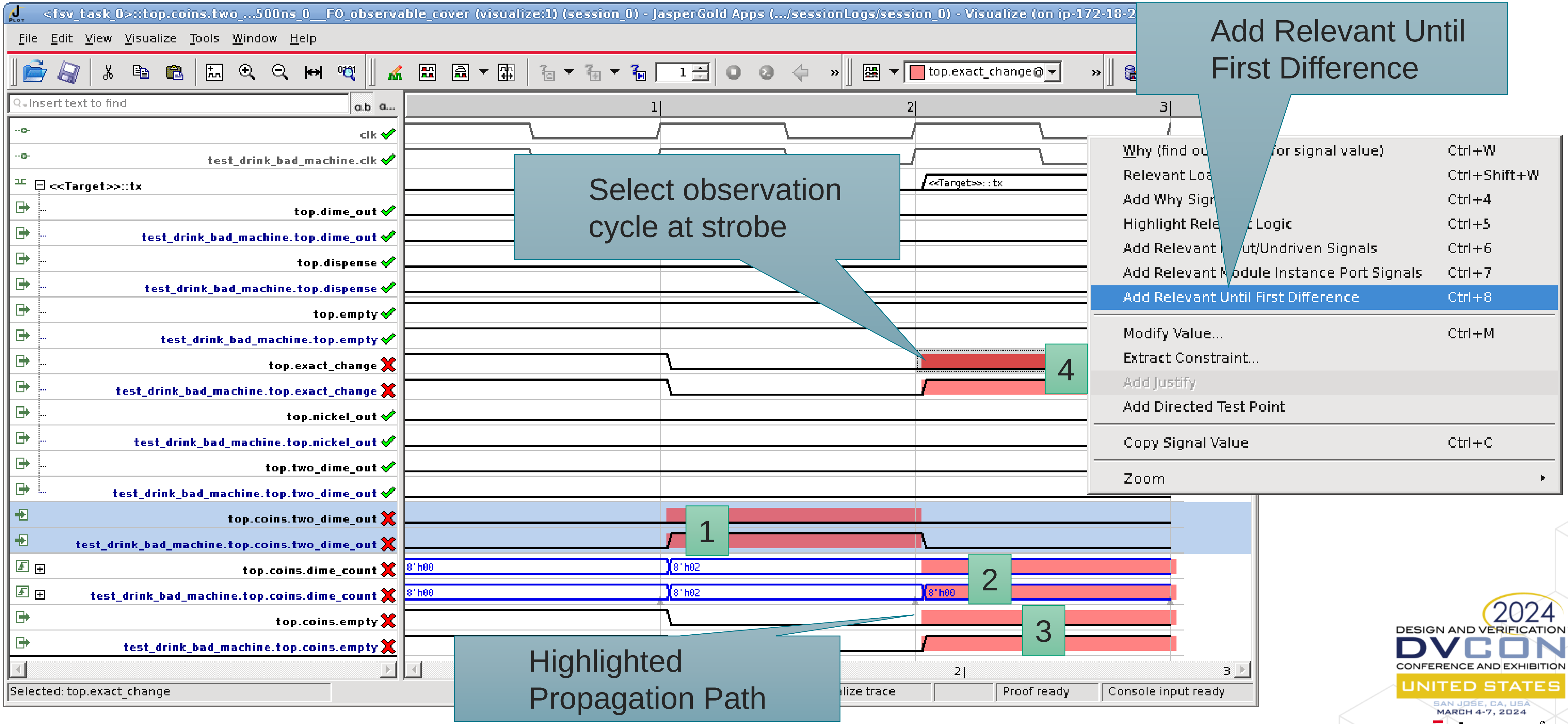
```
        always_ff @(posedge clk)
           if (reset)
              current_state <= s_idle;
           else
              current_state <= next_state;


        always_comb
        begin : state_machine

           nickel_out = 0;
           dime_out = 0;
           two_dime_out = 0;
           dispense = 0;
           next_state = current_state;
```

Selected: test_drink_bad_...e.top.vending3.next_state | Driver | drink_machine.sv | drink_machine.sv | Visualize trace | No proofs running | Console input ready

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES SAN JOSE, CA, USA MARCH 4-7, 2024

cadence®

FSV Formal – Visualize Highlight Propagation Path


<fsv_task_0>::top.coins.two_...500ns_0__FO_observable_cover (visualize:1) (session_0) - JasperGold Apps (.../sessionLogs/session_0) - Visualize (on ip-172-18-2
File Edit View Visualize Tools Window Help
top.exact_change@
Insert text to find
clk
test_drink_bad_machine.clk
<<Target>>::tx
top.dime_out
test_drink_bad_machine.top.dime_out
top.dispense
test_drink_bad_machine.top.dispense
top.empty
test_drink_bad_machine.top.empty
top.exact_change
test_drink_bad_machine.top.exact_change
top.nickel_out
test_drink_bad_machine.top.nickel_out
top.two_dime_out
test_drink_bad_machine.top.two_dime_out
top.coins.two_dime_out
test_drink_bad_machine.top.coins.two_dime_out
top.coins.dime_count
test_drink_bad_machine.top.coins.dime_count
top.coins.empty
test_drink_bad_machine.top.coins.empty
8'h00
8'h02
Why (find ou... for signal value) Ctrl+W
Relevant Loa... Ctrl+Shift+W
Add Why Sign... Ctrl+4
Highlight Rele...t Logic Ctrl+5
Add Relevant I...ut/Undriven Signals Ctrl+6
Add Relevant Module Instance Port Signals Ctrl+7
Add Relevant Until First Difference Ctrl+8
Modify Value... Ctrl+M
Extract Constraint...
Add Justify
Add Directed Test Point
Copy Signal Value Ctrl+C
Zoom
Selected: top.exact_change
...lize trace
Proof ready
Console input ready
1
2
3
4

Add Relevant Until First Difference

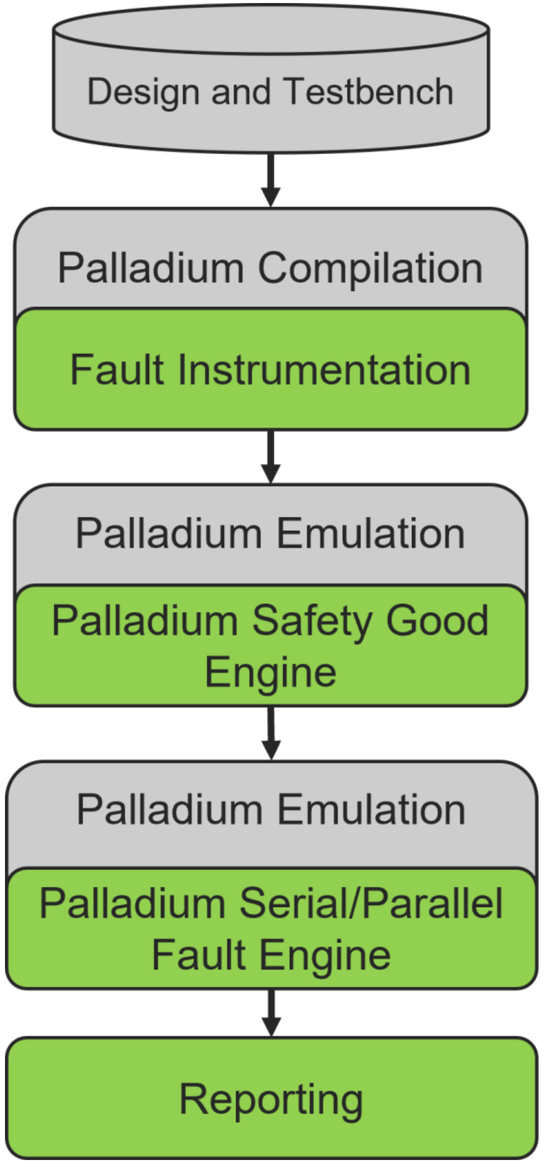
Select observation cycle at strobe

Highlighted Propagation Path

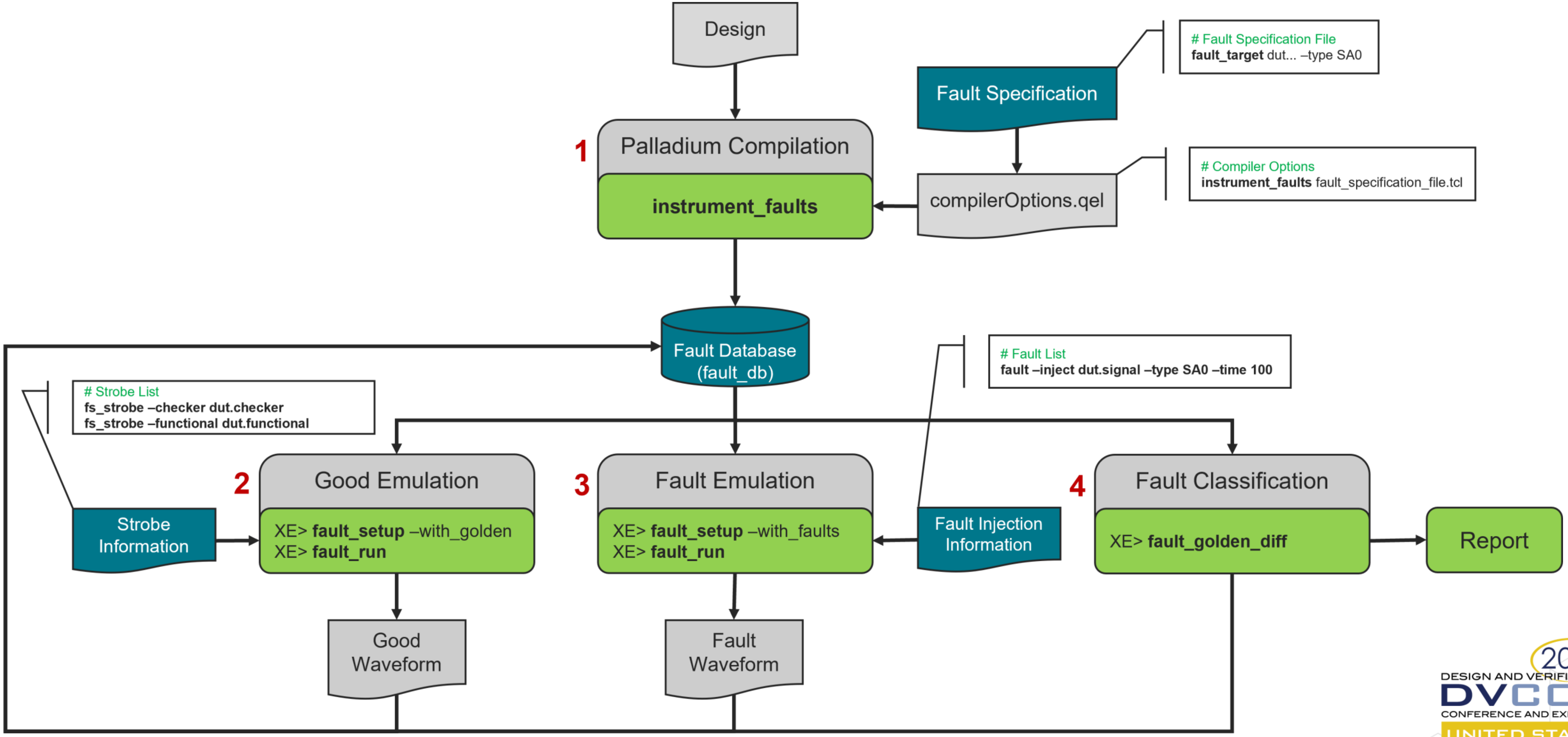
Palladium Safety

User Flow

- Easy to migrate from Functional verification flows to Fault Injection
  - Some files and option to be added to Palladium compilation
  - Faults are identified and instrumented during compilation
- Fault-free circuit emulation (Good Emulation)
  - Process strobe points and capture good waveform
- Fault Emulation Flows
  - Serial Fault Injection
  - Parallel Fault Injection
  - Interactive Fault Injection
- Fault Detection
  - Post-processing
    - Compares good and fault waveforms after each run
  - Inline
    - Detects the fault during the run using detection system
- Reporting
  - Standalone or using Xcelium utility (xfr)



Palladium Safety Flow Overview



Design

1 Palladium Compilation

instrument_faults

Fault Specification

```
# Fault Specification File
fault_target dut... –type SA0
```

compilerOptions.qel

```
# Compiler Options
instrument_faults fault_specification_file.tcl
```

Fault Database (fault_db)

```
# Strobe List
fs_strobe –checker dut.checker
fs_strobe –functional dut.functional
```

```
# Fault List
fault –inject dut.signal –type SA0 –time 100
```

Strobe Information

2 Good Emulation

```
XE> fault_setup –with_golden
XE> fault_run
```

3 Fault Emulation

```
XE> fault_setup –with_faults
XE> fault_run
```

Fault Injection Information

4 Fault Classification

```
XE> fault_golden_diff
```

Report

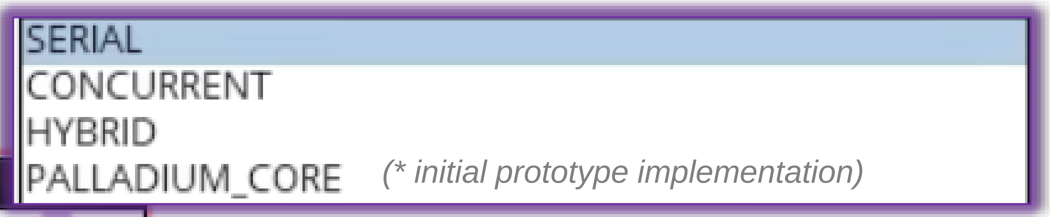
Good Waveform

Fault Waveform

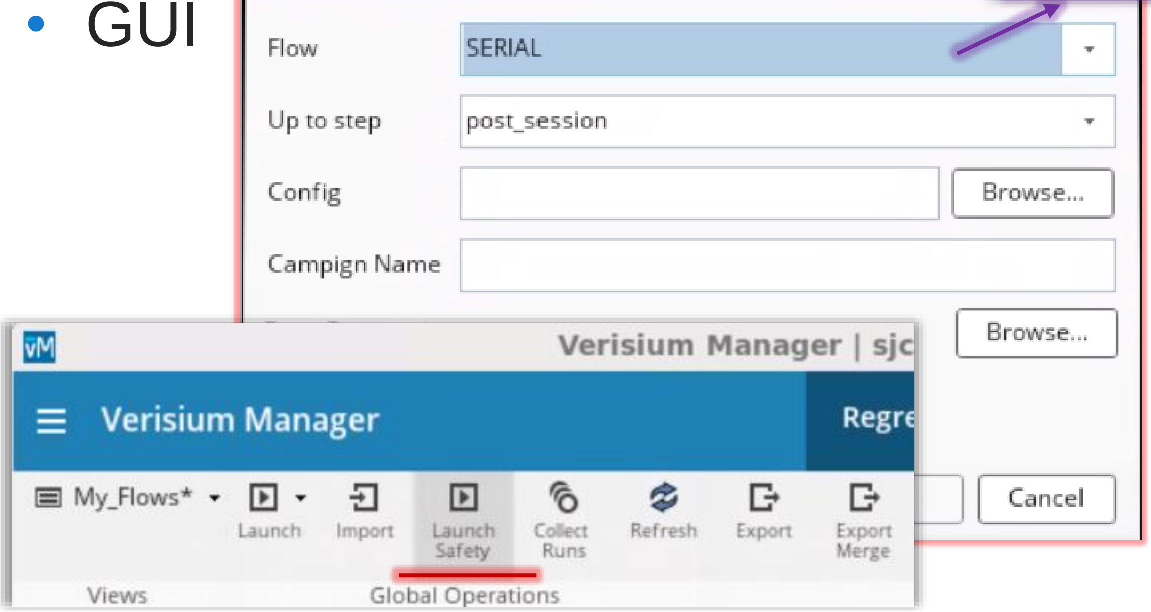
Fault Campaign Automation

Campaign Invocation

- GUI

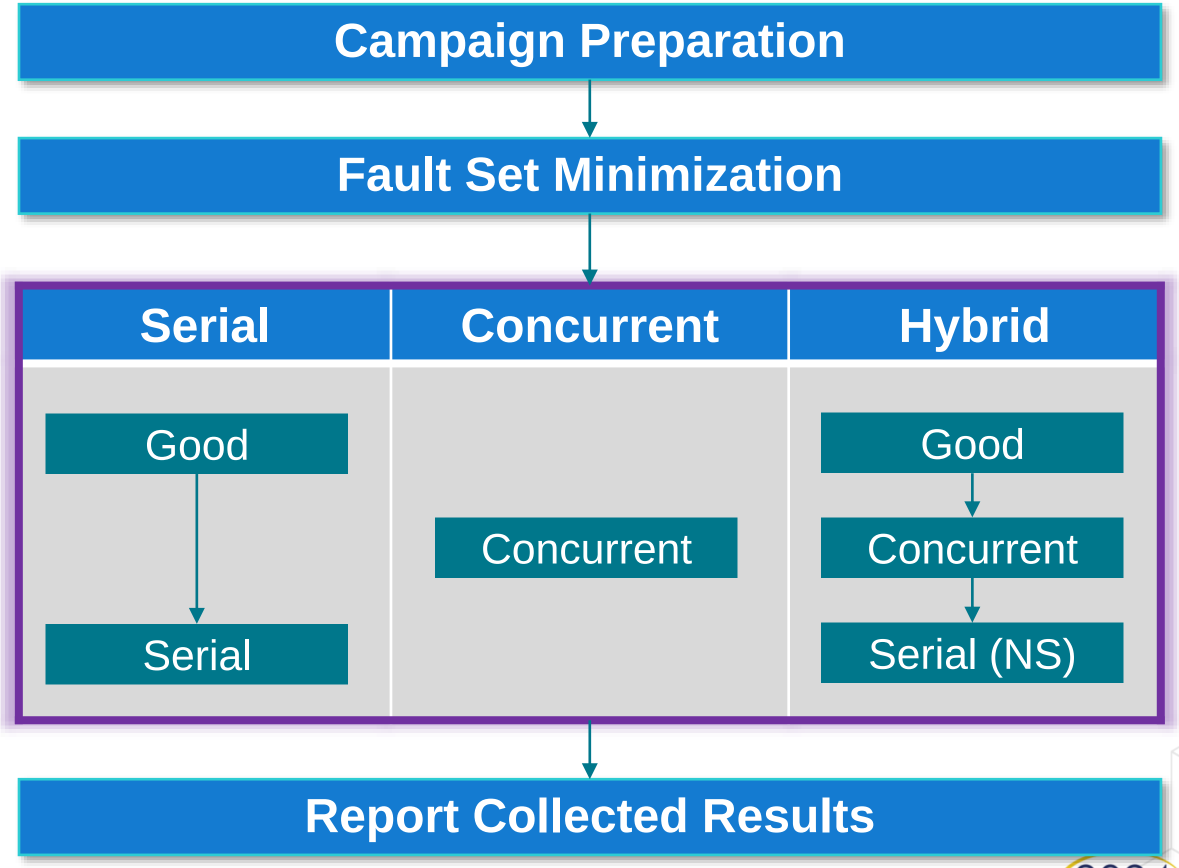


SERIAL
CONCURRENT
HYBRID
PALLADIUM_CORE *(* initial prototype implementation)*



Launch Safety Flow (on sjfdcl1008)

Flow	SERIAL
Up to step	post_session
Config	
Campign Name	



Campaign Preparation

Fault Set Minimization

Serial	Concurrent	Hybrid
Good → Serial	Concurrent	Good → Concurrent → Serial (NS)

Report Collected Results

- CLI

```
vmanager -safety \
  -execcmd "fi_campaign -launch <…> -flow_type <…> -cfg <…>"
```

Campaign Preparation

Organize Data

- Campaign directory

```
fs_exec_myc.HYBRID.usr…
├── flowData
├── input
├── report
└── sessions
    ├── myc.HYBRID.usr…
    ├── myc.test_select.usr…
    ├── myc.elaboration.usr…
    ├── myc.fst.usr…
    ├── myc.good_simulation.usr…
    ├── myc.fault_pruning.usr…
    ├── myc.fault.usr…
    └── myc.fault.usr…
```

Translate Inputs

- User-input (e.g., strobes)

```
strobe functional top.dut.o
strobe checker top.sm.alarm
```

- Xcelium syntax

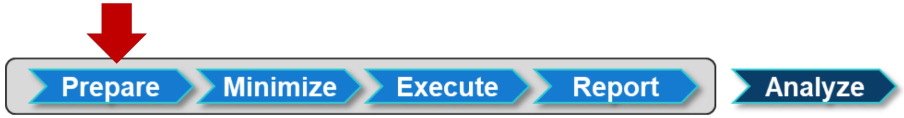
```
fs_strobe -functional top.dut.o
fs_strobe -checker top.sm.alarm
```

- Jasper syntax

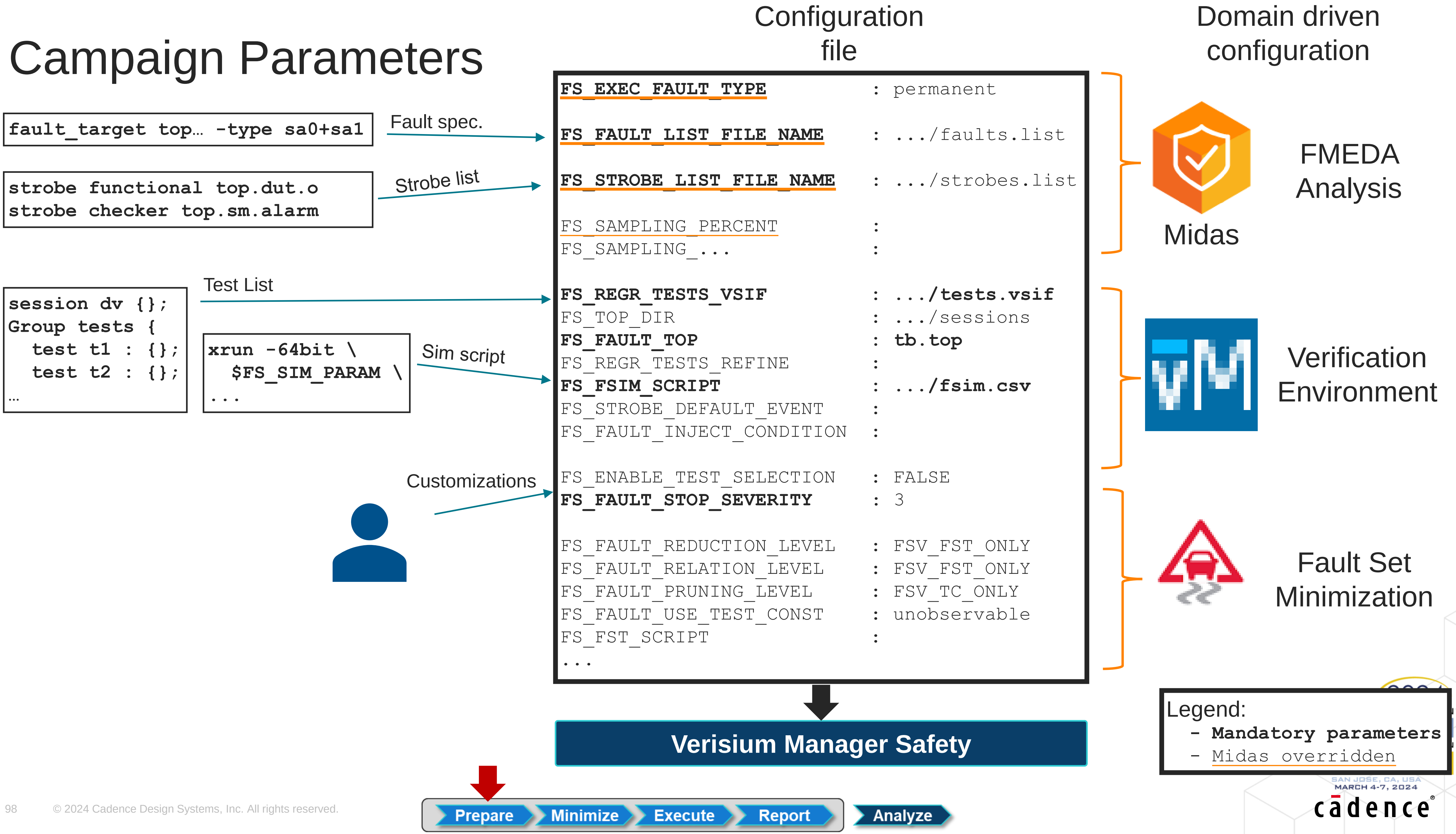
```
strobe functional dut.out
strobe checker sm.alarm
```

Prune Tests (optional)

- Remove redundant tests
  - 0% additional coverage
- Order per cov/time
- Customizable heuristic
  - Coverage type and contribution threshold
- Permanent campaigns
  - Select functional tests



Campaign Parameters



Configuration file

Domain driven configuration

```
fault_target top… -type sa0+sa1
```

Fault spec.

```
strobe functional top.dut.o
strobe checker top.sm.alarm
```

Strobe list

```
session dv {};
Group tests {
  test t1 : {};
  test t2 : {};
…
```

Test List

```
xrun -64bit \
  $FS_SIM_PARAM \
...
```

Sim script

Customizations

```
FS_EXEC_FAULT_TYPE             : permanent

FS_FAULT_LIST_FILE_NAME        : .../faults.list

FS_STROBE_LIST_FILE_NAME       : .../strobes.list

FS_SAMPLING_PERCENT            :
FS_SAMPLING_...                :

FS_REGR_TESTS_VSIF             : .../tests.vsif
FS_TOP_DIR                     : .../sessions
FS_FAULT_TOP                   : tb.top
FS_REGR_TESTS_REFINE           :
FS_FSIM_SCRIPT                 : .../fsim.csv
FS_STROBE_DEFAULT_EVENT        :
FS_FAULT_INJECT_CONDITION      :

FS_ENABLE_TEST_SELECTION       : FALSE
FS_FAULT_STOP_SEVERITY         : 3

FS_FAULT_REDUCTION_LEVEL       : FSV_FST_ONLY
FS_FAULT_RELATION_LEVEL        : FSV_FST_ONLY
FS_FAULT_PRUNING_LEVEL         : FSV_TC_ONLY
FS_FAULT_USE_TEST_CONST        : unobservable
FS_FST_SCRIPT                  :
...
```

Midas

FMEDA Analysis

Verification Environment

Fault Set Minimization

Verisium Manager Safety

Legend:
- **Mandatory parameters**
- Midas overridden

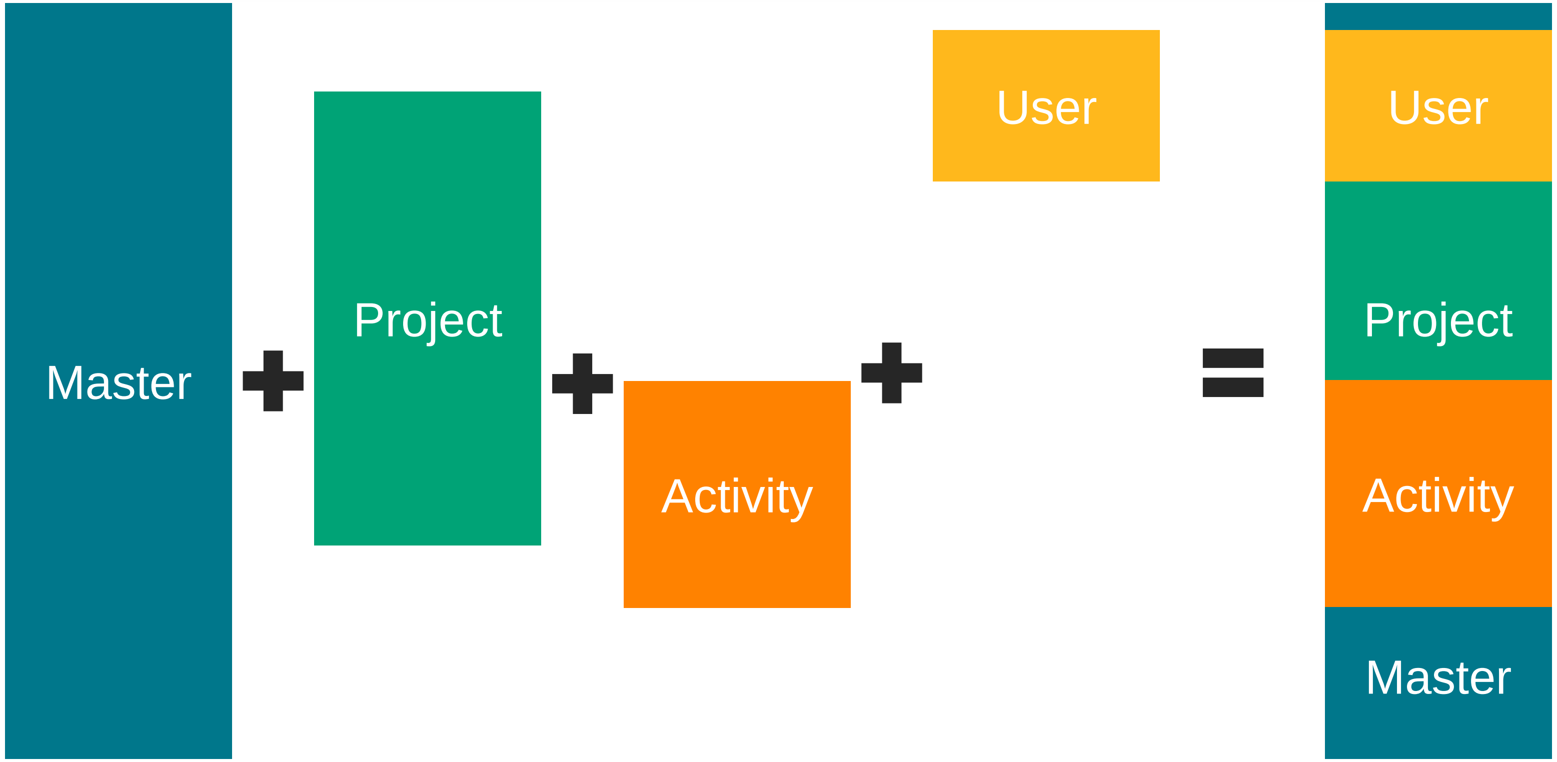
SAN JOSE, CA, USA
MARCH 4-7, 2024

cadence

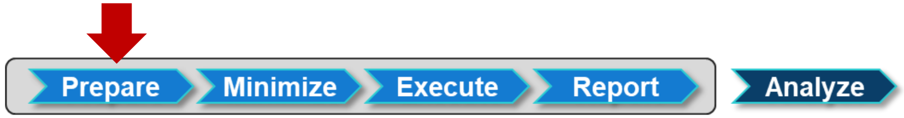
Campaign Configuration

- Parameters override and traceability

```
fi_campaign -launch fi -flow_type CONCURRENT \
  -cfg master.cfg,project.cfg,activity.cfg,user.cfg
```

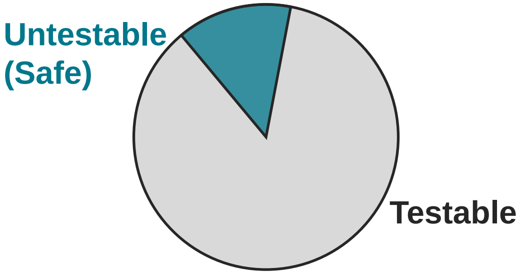


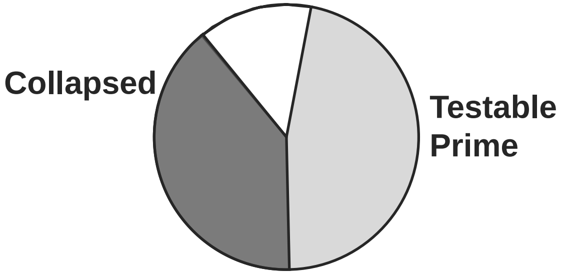
```
fs_exec_myc.HYBRID.usr…
├── flowData
├── input
│   ├── fs_exec1.cfg
│   ├── fs_exec2.cfg
│   ├── fs_exec3.cfg
│   └── fs_exec4.cfg
├── report
└── sessions
```

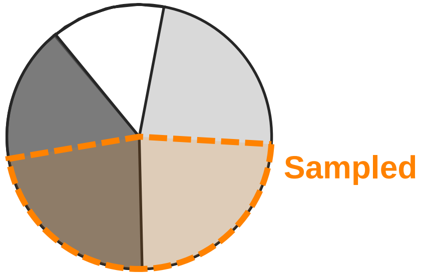


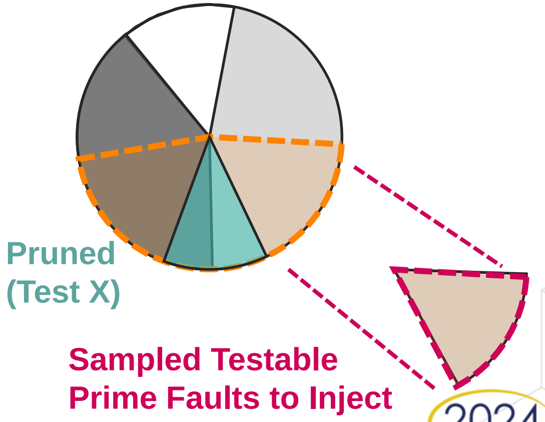
Fault Set Minimization

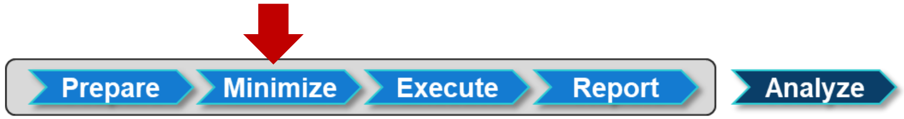
Design Structure		Statistics	Test Stimulus
Testability Analysis	Fault Collapsing	Random Sampling	Fault Pruning
Identify faults: - Uncontrollable - Unobservable	Group equivalent faults and consider only their prime representative	Estimate the overall results based on a representative sample	Find extra untestable faults by constraining testability based on stimulus patterns











Statistical Sampling - Sample Size Calculation

Leveugle, R., Calvez, A., Maistri, P., & Vanhauwaert, P. (2009). Statistical fault injection: Quantified error and confidence. *2009 Design, Automation & Test in Europe Conference & Exhibition*, 502–506. https://doi.org/10.1109/DATE.2009.5090716

- ISO26262-11:2018 – 4.8.1 General Fault Injection
  - *“NOTE 4 A sampling factor can be used to reduce the fault list if justified with respect to the specified purpose, confidence level, type/nature of the safety mechanism, selection criteria etc.”*
- Statistical Sampling
  - It allows selecting subset(s) to estimate properties of the population set
    - i.e., to estimate the “***proportion***” of faults that are covered (the campaign result)
  - The required precision of the estimated result defines the calculated sample size
    - i.e., the “***confidence level***” that the estimated “***proportion***” (result) is within the “***error margin***”
      - Note: current implementation assumes infinite population size (conservative) – i.e., larger the “population”, greater the sample size.
        Note: infinite vs. finite population size shows insignificant impact on the calculation of samples representing less than 5% of the population

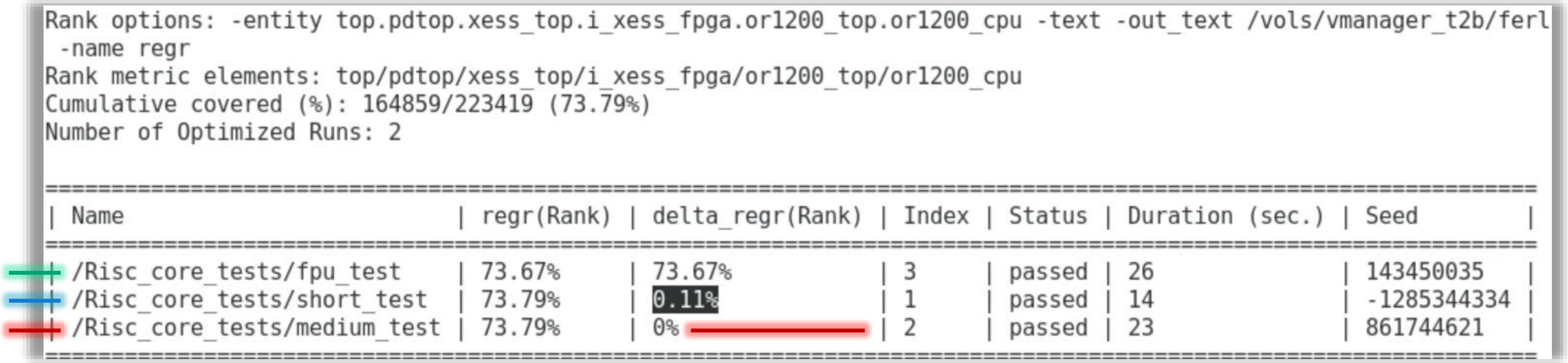
Campaign Parameters	Conservative Recommendation	Typical values			
FS_SAMPLING_ERROR : 0.5 *“error margin” percentage*	Conservative/Tighter error margins (near 0%) are used when the estimated proportion gets closer to 100%	1	1	0.5	0.5
FS_SAMPLING_CONFIDENCE : 95 *“confidence level” percentage*	Conservative/High confidence levels (near 100%) increase sample size, but not as much as reducing error margin	95	99	95	99
FS_SAMPLING_PROPORTION : 50 *“population proportion” or estimation*	Conservatively use 50% when no rational estimation exists	50	50	50	50
	Sample Size	***9,604***	***16,588***	***38,415***	***66,349***

Prepare Minimize Execute Report Analyze

Test Pruning & Ordering

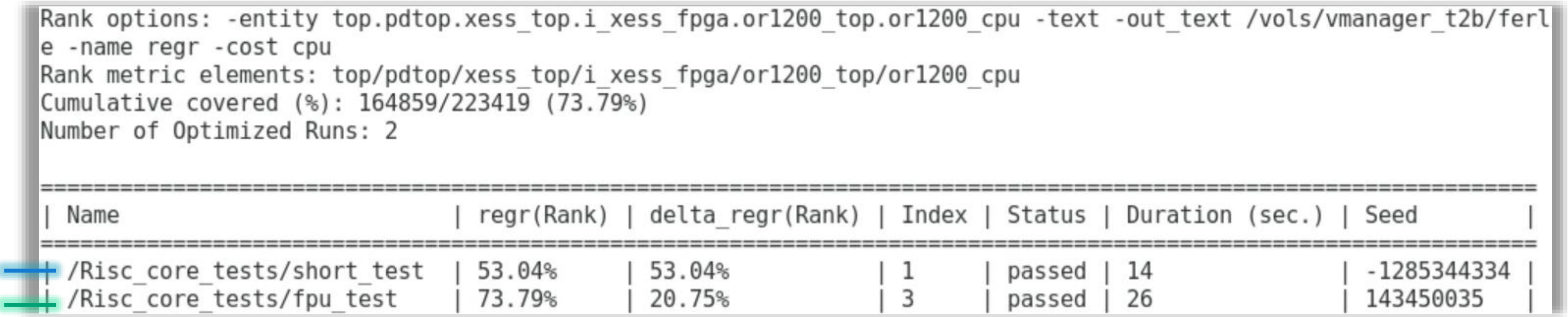
- Optional selection of functional tests using toggle-coverage based heuristics
  - Suitable for permanent faults. Configurable coverage type and pruning cutpoint

- 1st Pruning

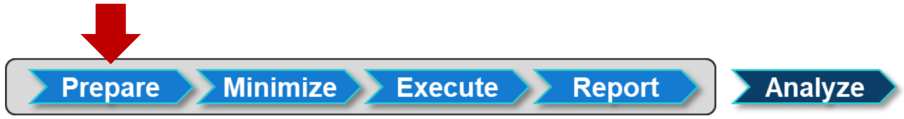


```
Rank options: -entity top.pdtop.xess_top.i_xess_fpga.or1200_top.or1200_cpu -text -out_text /vols/vmanager_t2b/ferl
 -name regr
Rank metric elements: top/pdtop/xess_top/i_xess_fpga/or1200_top/or1200_cpu
Cumulative covered (%): 164859/223419 (73.79%)
Number of Optimized Runs: 2

===================================================================================================================
| Name                         | regr(Rank) | delta_regr(Rank) | Index | Status | Duration (sec.) | Seed          |
===================================================================================================================
| /Risc_core_tests/fpu_test    | 73.67%     | 73.67%           | 3     | passed | 26              | 143450035     |
| /Risc_core_tests/short_test  | 73.79%     | 0.11%            | 1     | passed | 14              | -1285344334   |
| /Risc_core_tests/medium_test | 73.79%     | 0%               | 2     | passed | 23              | 861744621     |
===================================================================================================================
```

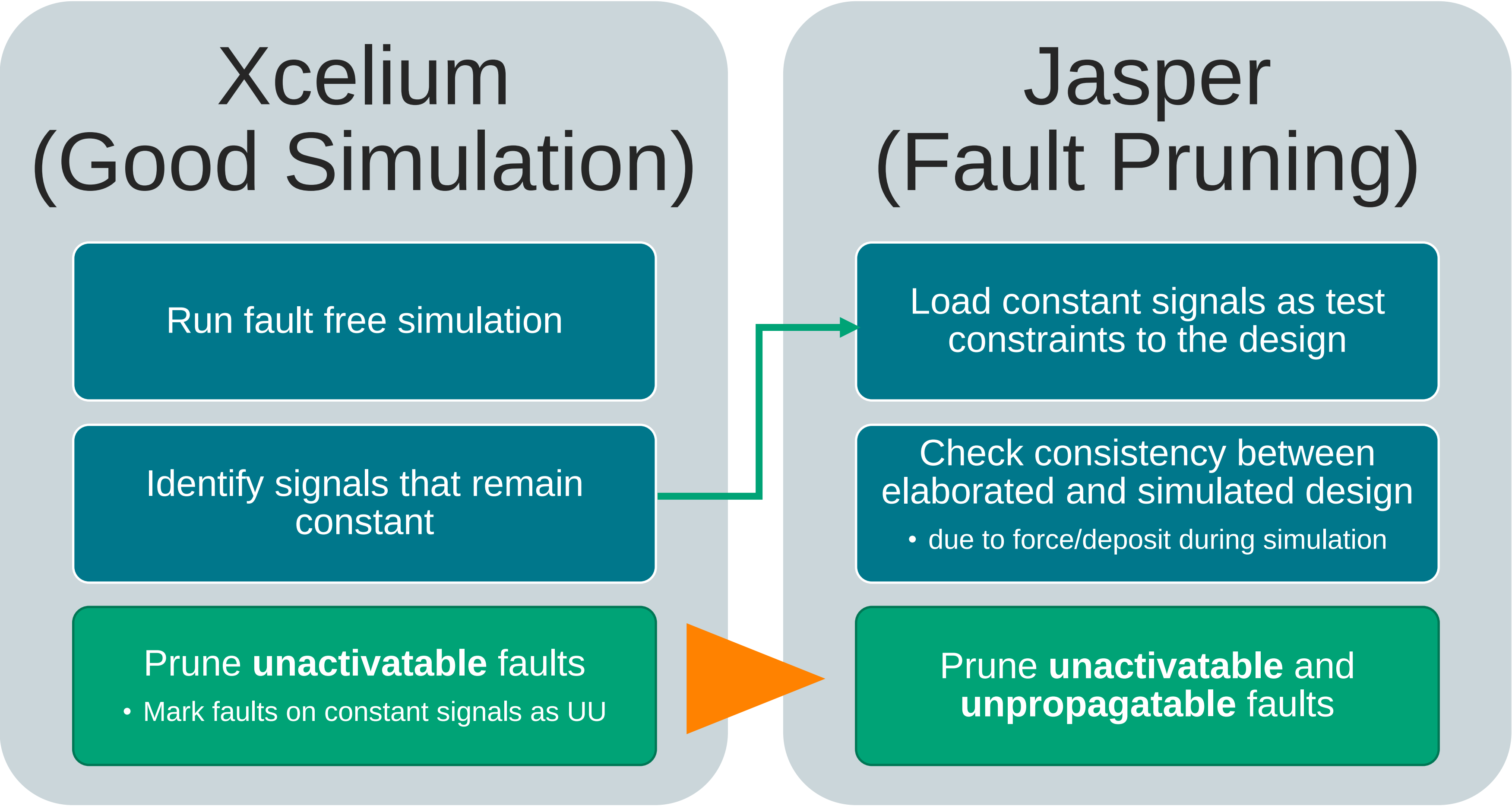
- 2nd Ordering



```
Rank options: -entity top.pdtop.xess_top.i_xess_fpga.or1200_top.or1200_cpu -text -out_text /vols/vmanager_t2b/ferl
e -name regr -cost cpu
Rank metric elements: top/pdtop/xess_top/i_xess_fpga/or1200_top/or1200_cpu
Cumulative covered (%): 164859/223419 (73.79%)
Number of Optimized Runs: 2

===================================================================================================================
| Name                         | regr(Rank) | delta_regr(Rank) | Index | Status | Duration (sec.) | Seed          |
===================================================================================================================
| /Risc_core_tests/short_test  | 53.04%     | 53.04%           | 1     | passed | 14              | -1285344334   |
| /Risc_core_tests/fpu_test    | 73.79%     | 20.75%           | 3     | passed | 26              | 143450035     |
```



Fault Pruning



Xcelium (Good Simulation)

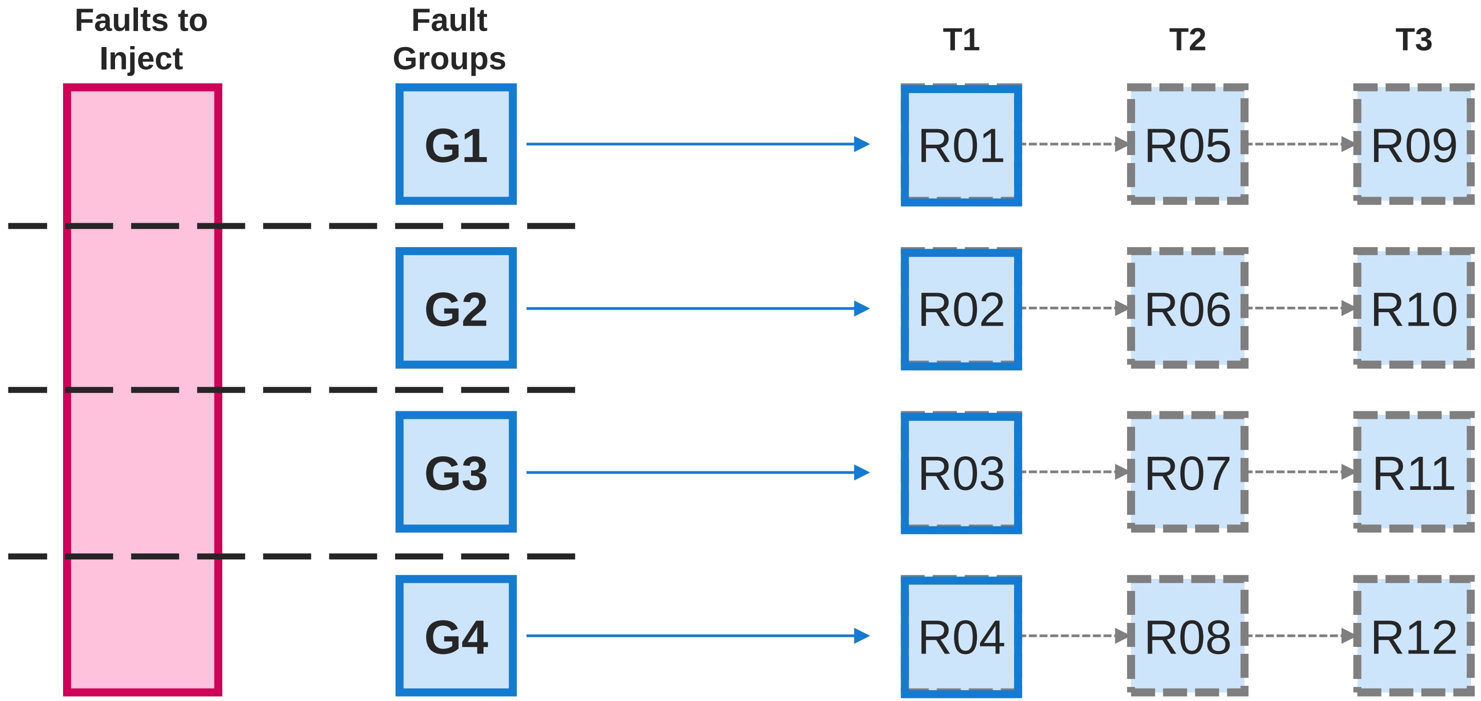
- Run fault free simulation
- Identify signals that remain constant
- Prune **unactivatable** faults
  - Mark faults on constant signals as UU

Jasper (Fault Pruning)

- Load constant signals as test constraints to the design
- Check consistency between elaborated and simulated design
  - due to force/deposit during simulation
- Prune **unactivatable** and **unpropagatable** faults

Prepare › Minimize › Execute › Report › Analyze

Fault Injection Execution



Fault Groups	T1	T2	T3
G1	R01	R05	R09
G2	R02	R06	R10
G3	R03	R07	R11
G4	R04	R08	R12

RUN XX

1. Filter Pruned Faults
2. Invoke Serial/Concurrent
   1. Inject fault/s
   2. Drop detected
   3. Stop simulation
3. Scan simulator logs
4. Remove dropped faults
   1. Skip next test if no left

Fault Grouping

- Configurable
- Max F per G (Serial = 1)

Fault Session Build

- Test dependency
- Submit runs to the computer farm

Fault Run Execution

- Filter faults
- Check for errors
- Optimize runs

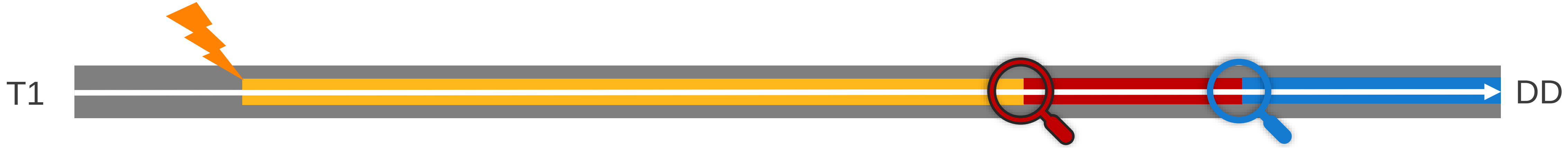
Prepare → Minimize → Execute → Report → Analyze

Fault Dropping

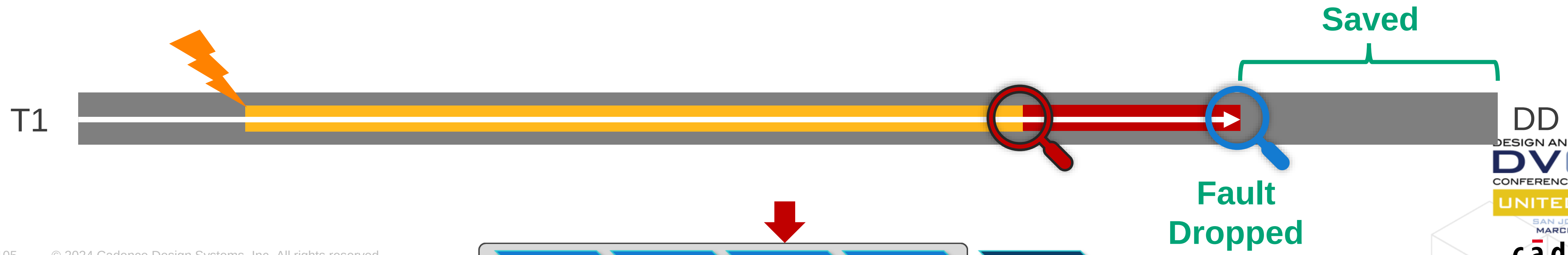
Fault injection
Checker Strobe
Functional Strobe
Test / Simulation

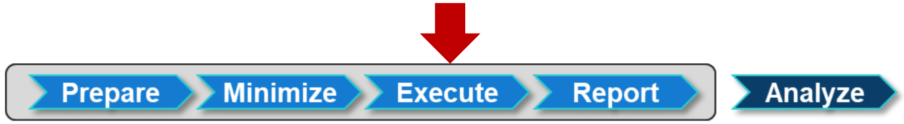
- Stopping simulating covered faults

- Without dropping



- **Optimized**



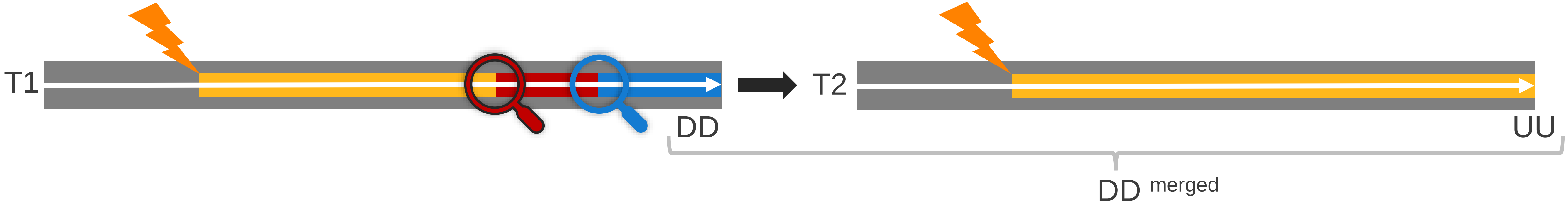


Fault injection
Checker Strobe
Functional Strobe
Test / Simulation

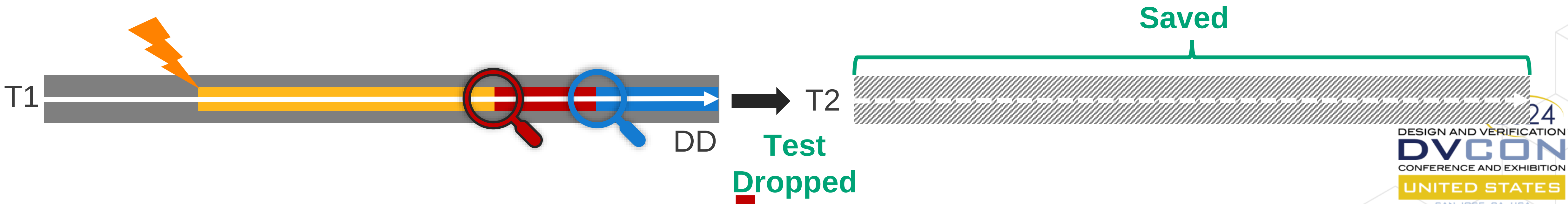
Test Dropping

- Skip running already covered faults with subsequent tests

- Without dropping

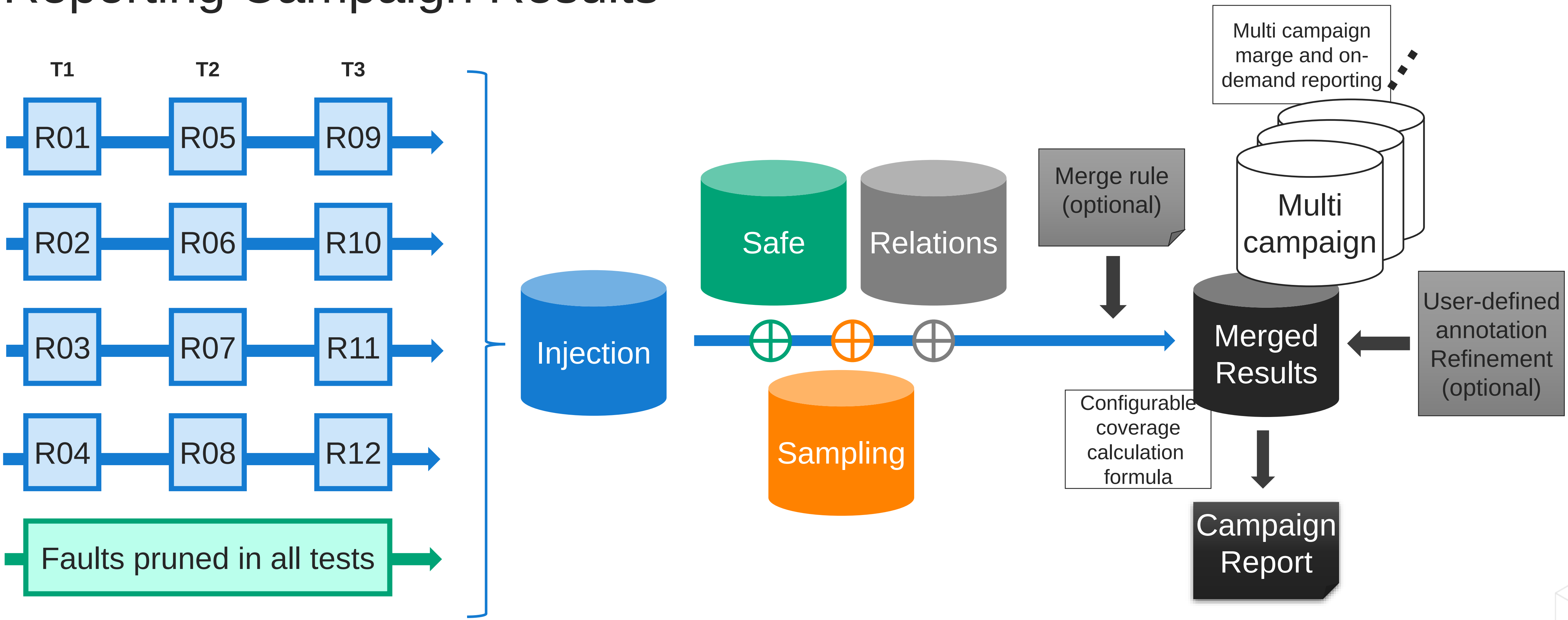


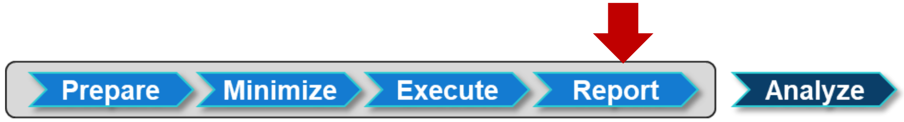
- **Optimized**



Prepare → Minimize → Execute → Report → Analyze

Reporting Campaign Results

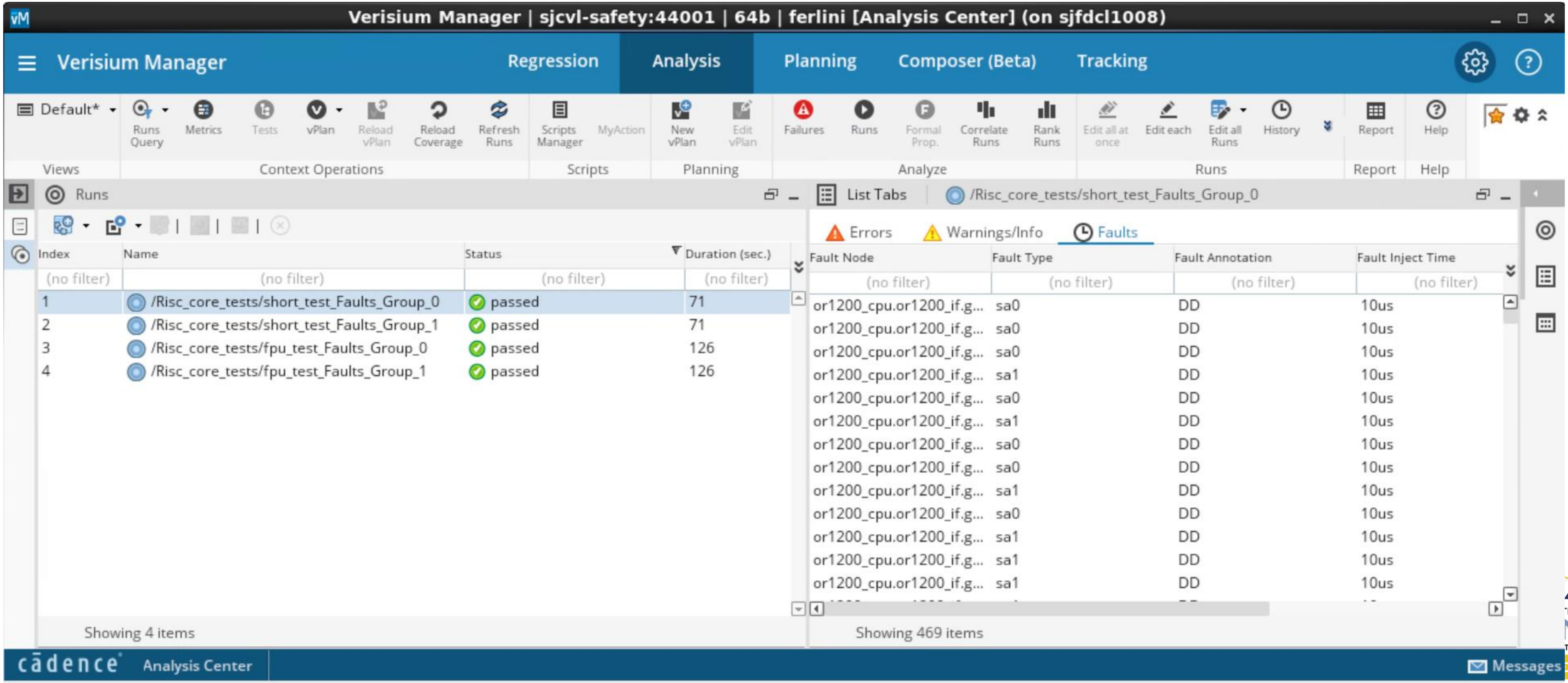

T1
T2
T3
R01
R05
R09
R02
R06
R10
R03
R07
R11
R04
R08
R12
Faults pruned in all tests
Injection
Safe
Relations
Sampling
Merge rule (optional)
Multi campaign marge and on-demand reporting
Multi campaign
Merged Results
User-defined annotation Refinement (optional)
Configurable coverage calculation formula
Campaign Report



Fault Campaign Analysis

Fault Simulation Results

Run generated fault annotation



Index	Name	Status	Duration (sec.)
(no filter)	(no filter)	(no filter)	(no filter)
1	/Risc_core_tests/short_test_Faults_Group_0	✓ passed	71
2	/Risc_core_tests/short_test_Faults_Group_1	✓ passed	71
3	/Risc_core_tests/fpu_test_Faults_Group_0	✓ passed	126
4	/Risc_core_tests/fpu_test_Faults_Group_1	✓ passed	126

Showing 4 items

Fault Node	Fault Type	Fault Annotation	Fault Inject Time
(no filter)	(no filter)	(no filter)	(no filter)
or1200_cpu.or1200_if.g...	sa0	DD	10us
or1200_cpu.or1200_if.g...	sa0	DD	10us
or1200_cpu.or1200_if.g...	sa0	DD	10us
or1200_cpu.or1200_if.g...	sa1	DD	10us
or1200_cpu.or1200_if.g...	sa0	DD	10us
or1200_cpu.or1200_if.g...	sa1	DD	10us
or1200_cpu.or1200_if.g...	sa0	DD	10us
or1200_cpu.or1200_if.g...	sa0	DD	10us
or1200_cpu.or1200_if.g...	sa1	DD	10us
or1200_cpu.or1200_if.g...	sa0	DD	10us
or1200_cpu.or1200_if.g...	sa1	DD	10us
or1200_cpu.or1200_if.g...	sa1	DD	10us
or1200_cpu.or1200_if.g...	sa1	DD	10us

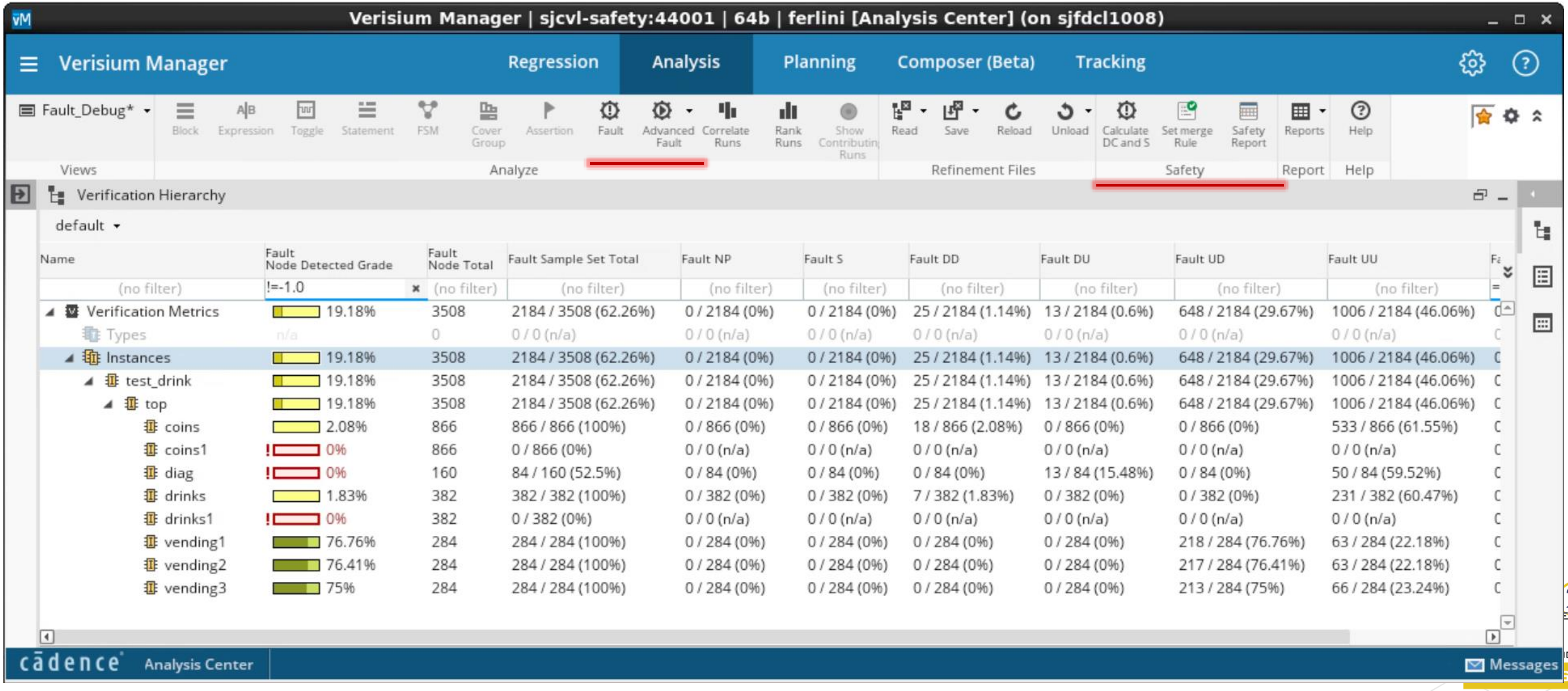
Showing 469 items

Prepare → Minimize → Execute → Report → Analyze

MARCH 4-7, 2024

Fault Campaign Results – Hierarchical View

Merged annotation

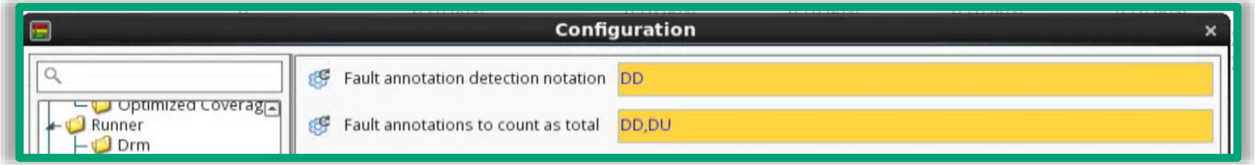


Name	Fault Node Detected Grade	Fault Node Total	Fault Sample Set Total	Fault NP	Fault S	Fault DD	Fault DU	Fault UD	Fault UU	Fa
(no filter)	!=-1.0	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	=
Verification Metrics	19.18%	3508	2184 / 3508 (62.26%)	0 / 2184 (0%)	0 / 2184 (0%)	25 / 2184 (1.14%)	13 / 2184 (0.6%)	648 / 2184 (29.67%)	1006 / 2184 (46.06%)	0
Types	n/a	0	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	C
Instances	19.18%	3508	2184 / 3508 (62.26%)	0 / 2184 (0%)	0 / 2184 (0%)	25 / 2184 (1.14%)	13 / 2184 (0.6%)	648 / 2184 (29.67%)	1006 / 2184 (46.06%)	C
test_drink	19.18%	3508	2184 / 3508 (62.26%)	0 / 2184 (0%)	0 / 2184 (0%)	25 / 2184 (1.14%)	13 / 2184 (0.6%)	648 / 2184 (29.67%)	1006 / 2184 (46.06%)	C
top	19.18%	3508	2184 / 3508 (62.26%)	0 / 2184 (0%)	0 / 2184 (0%)	25 / 2184 (1.14%)	13 / 2184 (0.6%)	648 / 2184 (29.67%)	1006 / 2184 (46.06%)	C
coins	2.08%	866	866 / 866 (100%)	0 / 866 (0%)	0 / 866 (0%)	18 / 866 (2.08%)	0 / 866 (0%)	0 / 866 (0%)	533 / 866 (61.55%)	C
coins1	! 0%	866	0 / 866 (0%)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	C
diag	! 0%	160	84 / 160 (52.5%)	0 / 84 (0%)	0 / 84 (0%)	0 / 84 (0%)	13 / 84 (15.48%)	0 / 84 (0%)	50 / 84 (59.52%)	C
drinks	1.83%	382	382 / 382 (100%)	0 / 382 (0%)	0 / 382 (0%)	7 / 382 (1.83%)	0 / 382 (0%)	0 / 382 (0%)	231 / 382 (60.47%)	C
drinks1	! 0%	382	0 / 382 (0%)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	C
vending1	76.76%	284	284 / 284 (100%)	0 / 284 (0%)	0 / 284 (0%)	0 / 284 (0%)	0 / 284 (0%)	218 / 284 (76.76%)	63 / 284 (22.18%)	C
vending2	76.41%	284	284 / 284 (100%)	0 / 284 (0%)	0 / 284 (0%)	0 / 284 (0%)	0 / 284 (0%)	217 / 284 (76.41%)	63 / 284 (22.18%)	C
vending3	75%	284	284 / 284 (100%)	0 / 284 (0%)	0 / 284 (0%)	0 / 284 (0%)	0 / 284 (0%)	213 / 284 (75%)	66 / 284 (23.24%)	C

Prepare → Minimize → Execute → Report → Analyze

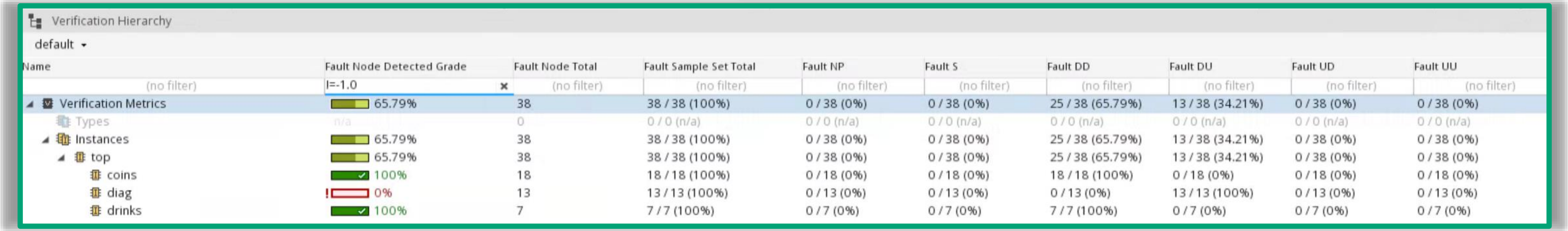
Fault Campaign Analysis

$$\frac{\#DD}{\#DD + \#DU} = Diag.Cov.$$

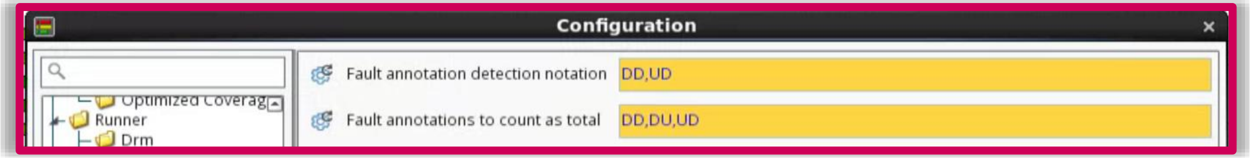


Fault annotation detection notation: DD

Fault annotations to count as total: DD,DU



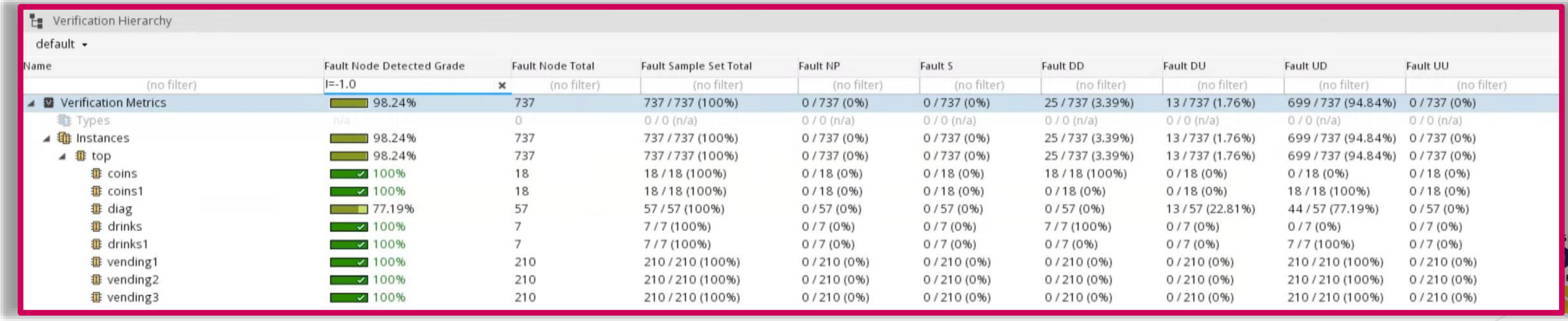
Name	Fault Node Detected Grade	Fault Node Total	Fault Sample Set Total	Fault NP	Fault S	Fault DD	Fault DU	Fault UD	Fault UU
(no filter)	!=-1.0	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
Verification Metrics	65.79%	38	38 / 38 (100%)	0 / 38 (0%)	0 / 38 (0%)	25 / 38 (65.79%)	13 / 38 (34.21%)	0 / 38 (0%)	0 / 38 (0%)
Types	n/a	0	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)
Instances	65.79%	38	38 / 38 (100%)	0 / 38 (0%)	0 / 38 (0%)	25 / 38 (65.79%)	13 / 38 (34.21%)	0 / 38 (0%)	0 / 38 (0%)
top	65.79%	38	38 / 38 (100%)	0 / 38 (0%)	0 / 38 (0%)	25 / 38 (65.79%)	13 / 38 (34.21%)	0 / 38 (0%)	0 / 38 (0%)
coins	✓ 100%	18	18 / 18 (100%)	0 / 18 (0%)	0 / 18 (0%)	18 / 18 (100%)	0 / 18 (0%)	0 / 18 (0%)	0 / 18 (0%)
diag	! 0%	13	13 / 13 (100%)	0 / 13 (0%)	0 / 13 (0%)	0 / 13 (0%)	13 / 13 (100%)	0 / 13 (0%)	0 / 13 (0%)
drinks	✓ 100%	7	7 / 7 (100%)	0 / 7 (0%)	0 / 7 (0%)	7 / 7 (100%)	0 / 7 (0%)	0 / 7 (0%)	0 / 7 (0%)



Fault annotation detection notation: DD,UD

Fault annotations to count as total: DD,DU,UD

$$\frac{\#DD + \#UD}{\#DD + \#UD + \#DU} = Custom\ Diag.Cov.$$

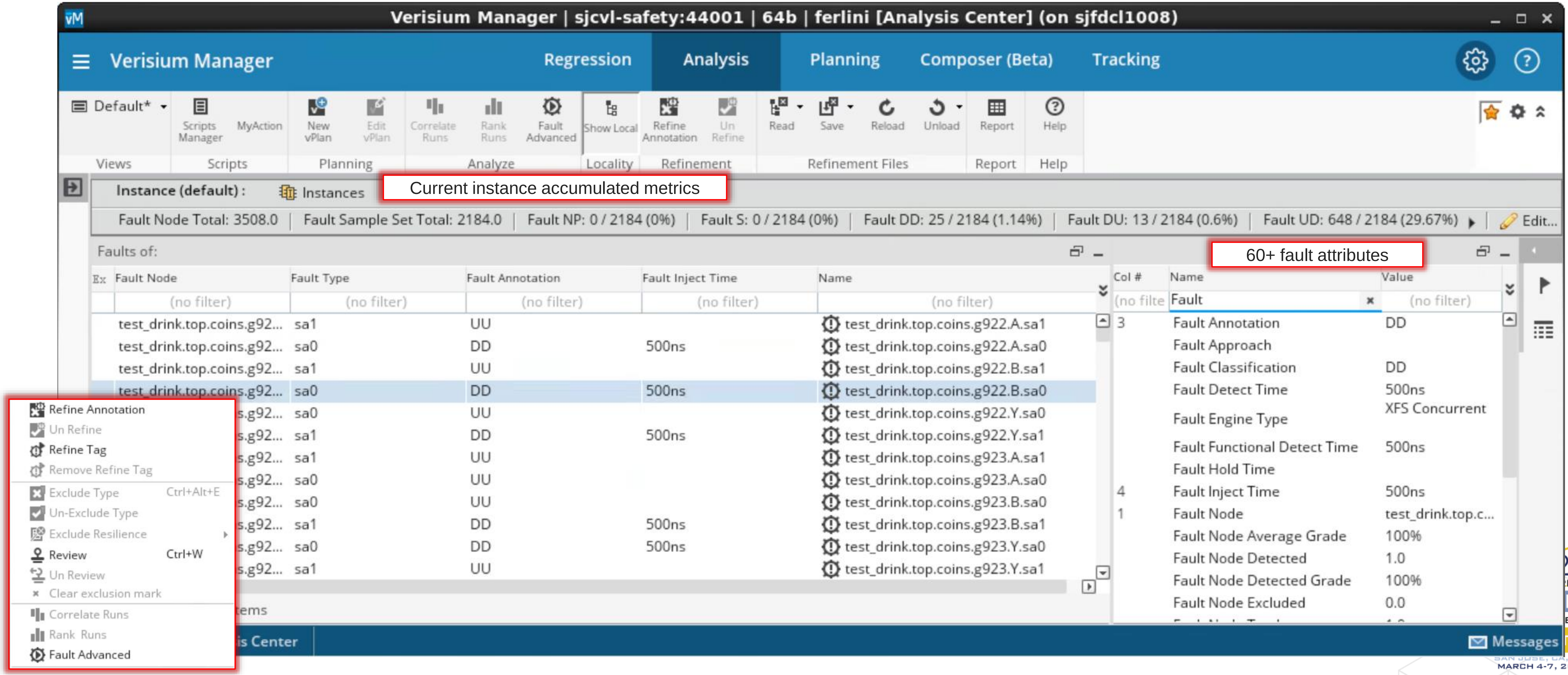


Name	Fault Node Detected Grade	Fault Node Total	Fault Sample Set Total	Fault NP	Fault S	Fault DD	Fault DU	Fault UD	Fault UU
(no filter)	!=-1.0	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
Verification Metrics	98.24%	737	737 / 737 (100%)	0 / 737 (0%)	0 / 737 (0%)	25 / 737 (3.39%)	13 / 737 (1.76%)	699 / 737 (94.84%)	0 / 737 (0%)
Types	n/a	0	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)
Instances	98.24%	737	737 / 737 (100%)	0 / 737 (0%)	0 / 737 (0%)	25 / 737 (3.39%)	13 / 737 (1.76%)	699 / 737 (94.84%)	0 / 737 (0%)
top	98.24%	737	737 / 737 (100%)	0 / 737 (0%)	0 / 737 (0%)	25 / 737 (3.39%)	13 / 737 (1.76%)	699 / 737 (94.84%)	0 / 737 (0%)
coins	✓ 100%	18	18 / 18 (100%)	0 / 18 (0%)	0 / 18 (0%)	18 / 18 (100%)	0 / 18 (0%)	0 / 18 (0%)	0 / 18 (0%)
coins1	✓ 100%	18	18 / 18 (100%)	0 / 18 (0%)	0 / 18 (0%)	0 / 18 (0%)	0 / 18 (0%)	18 / 18 (100%)	0 / 18 (0%)
diag	77.19%	57	57 / 57 (100%)	0 / 57 (0%)	0 / 57 (0%)	0 / 57 (0%)	13 / 57 (22.81%)	44 / 57 (77.19%)	0 / 57 (0%)
drinks	✓ 100%	7	7 / 7 (100%)	0 / 7 (0%)	0 / 7 (0%)	7 / 7 (100%)	0 / 7 (0%)	0 / 7 (0%)	0 / 7 (0%)
drinks1	✓ 100%	7	7 / 7 (100%)	0 / 7 (0%)	0 / 7 (0%)	0 / 7 (0%)	0 / 7 (0%)	7 / 7 (100%)	0 / 7 (0%)
vending1	✓ 100%	210	210 / 210 (100%)	0 / 210 (0%)	0 / 210 (0%)	0 / 210 (0%)	0 / 210 (0%)	210 / 210 (100%)	0 / 210 (0%)
vending2	✓ 100%	210	210 / 210 (100%)	0 / 210 (0%)	0 / 210 (0%)	0 / 210 (0%)	0 / 210 (0%)	210 / 210 (100%)	0 / 210 (0%)
vending3	✓ 100%	210	210 / 210 (100%)	0 / 210 (0%)	0 / 210 (0%)	0 / 210 (0%)	0 / 210 (0%)	210 / 210 (100%)	0 / 210 (0%)

$$\frac{\#DD + \#UD}{\#DD + \#UD + \#DU} = Custom\ Diag.Cov.$$

Fault Campaign Results – Annotated Fault List

Merged annotation



Current instance accumulated metrics

Fault Node Total: 3508.0 | Fault Sample Set Total: 2184.0 | Fault NP: 0 / 2184 (0%) | Fault S: 0 / 2184 (0%) | Fault DD: 25 / 2184 (1.14%) | Fault DU: 13 / 2184 (0.6%) | Fault UD: 648 / 2184 (29.67%)

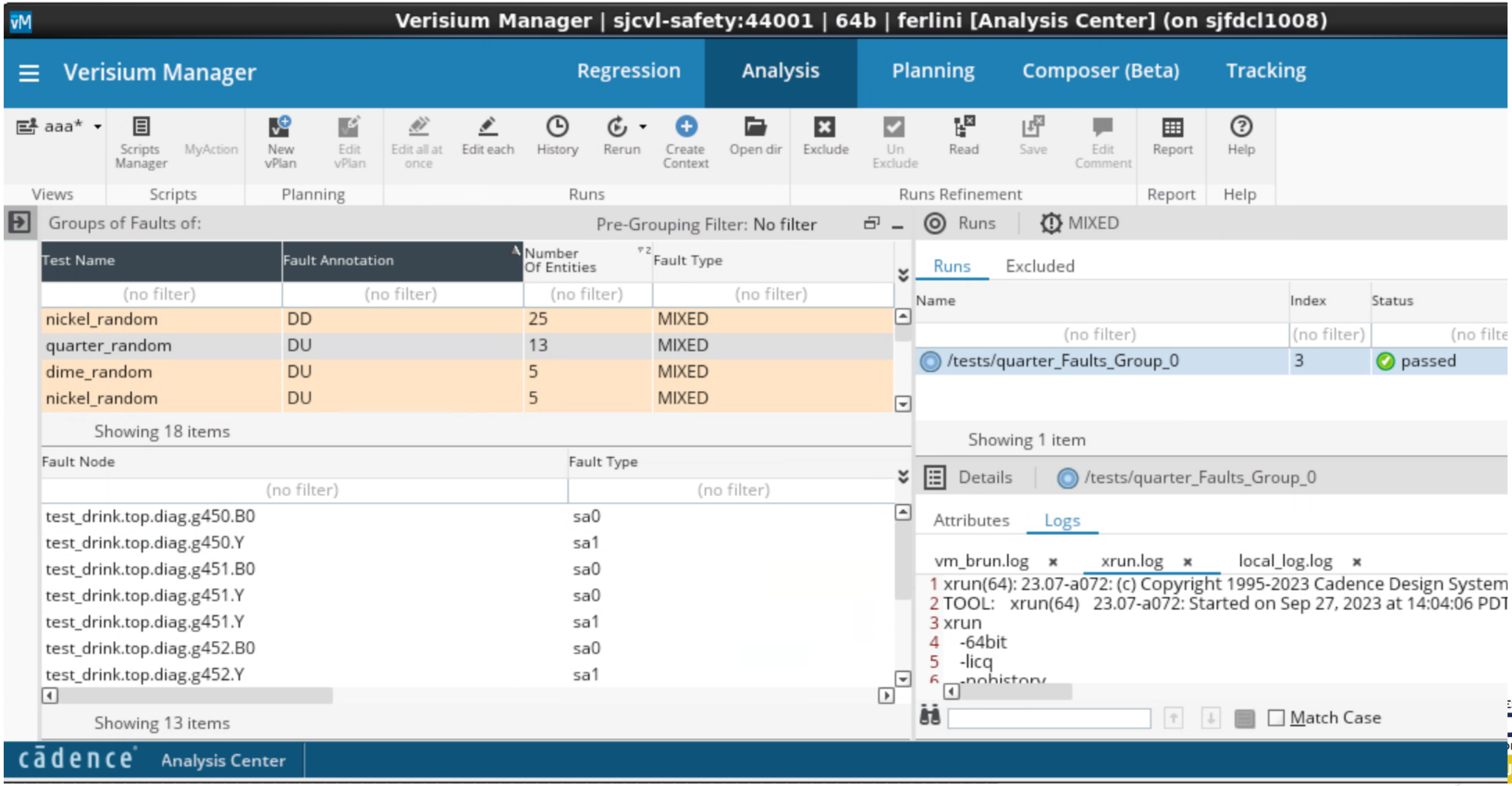
Ex	Fault Node	Fault Type	Fault Annotation	Fault Inject Time	Name
	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
	test_drink.top.coins.g92...	sa1	UU		test_drink.top.coins.g922.A.sa1
	test_drink.top.coins.g92...	sa0	DD	500ns	test_drink.top.coins.g922.A.sa0
	test_drink.top.coins.g92...	sa1	UU		test_drink.top.coins.g922.B.sa1
	test_drink.top.coins.g92...	sa0	DD	500ns	test_drink.top.coins.g922.B.sa0
	s.g92...	sa0	UU		test_drink.top.coins.g922.Y.sa0
	s.g92...	sa1	DD	500ns	test_drink.top.coins.g922.Y.sa1
	s.g92...	sa1	UU		test_drink.top.coins.g923.A.sa1
	s.g92...	sa0	UU		test_drink.top.coins.g923.A.sa0
	s.g92...	sa0	UU		test_drink.top.coins.g923.B.sa0
	s.g92...	sa1	DD	500ns	test_drink.top.coins.g923.B.sa1
	s.g92...	sa0	DD	500ns	test_drink.top.coins.g923.Y.sa0
	s.g92...	sa1	UU		test_drink.top.coins.g923.Y.sa1

60+ fault attributes

Col #	Name	Value
(no filte	Fault	(no filter)
3	Fault Annotation	DD
	Fault Approach	
	Fault Classification	DD
	Fault Detect Time	500ns
	Fault Engine Type	XFS Concurrent
	Fault Functional Detect Time	500ns
	Fault Hold Time	
4	Fault Inject Time	500ns
1	Fault Node	test_drink.top.c...
	Fault Node Average Grade	100%
	Fault Node Detected	1.0
	Fault Node Detected Grade	100%
	Fault Node Excluded	0.0

Fault Annotation Distribution per Test

Annotation per each test



Test Name	Fault Annotation	Number Of Entities	Fault Type
(no filter)	(no filter)	(no filter)	(no filter)
nickel_random	DD	25	MIXED
quarter_random	DU	13	MIXED
dime_random	DU	5	MIXED
nickel_random	DU	5	MIXED

Showing 18 items

Fault Node	Fault Type
(no filter)	(no filter)
test_drink.top.diag.g450.B0	sa0
test_drink.top.diag.g450.Y	sa1
test_drink.top.diag.g451.B0	sa0
test_drink.top.diag.g451.Y	sa0
test_drink.top.diag.g451.Y	sa1
test_drink.top.diag.g452.B0	sa0
test_drink.top.diag.g452.Y	sa1

Showing 13 items

Name	Index	Status
(no filter)	(no filter)	(no filte
/tests/quarter_Faults_Group_0	3	passed

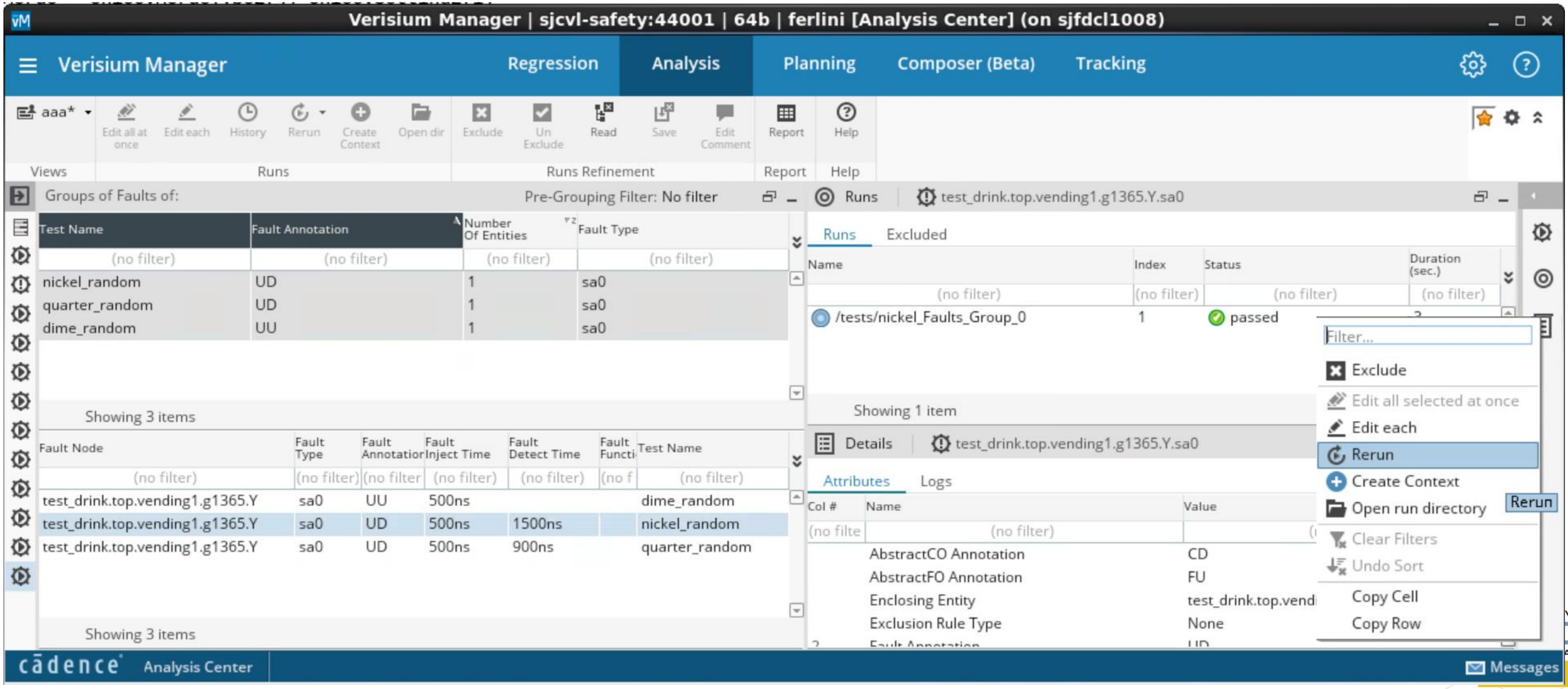
Showing 1 item

Details /tests/quarter_Faults_Group_0

```
xrun(64): 23.07-a072: (c) Copyright 1995-2023 Cadence Design System
TOOL:    xrun(64)    23.07-a072: Started on Sep 27, 2023 at 14:04:06 PDT
xrun
   -64bit
   -licq
   -nohistory
```

Fault Annotation Traceability

Result per each test



Test Name	Fault Annotation	Number Of Entities	Fault Type
(no filter)	(no filter)	(no filter)	(no filter)
nickel_random	UD	1	sa0
quarter_random	UD	1	sa0
dime_random	UU	1	sa0

Showing 3 items

Fault Node	Fault Type	Fault Annotatior	Fault Inject Time	Fault Detect Time	Fault Functi	Test Name
(no filter)	(no filter)	(no filter	(no filter)	(no filter)	(no f	(no filter)
test_drink.top.vending1.g1365.Y	sa0	UU	500ns			dime_random
test_drink.top.vending1.g1365.Y	sa0	UD	500ns	1500ns		nickel_random
test_drink.top.vending1.g1365.Y	sa0	UD	500ns	900ns		quarter_random

Showing 3 items

Name	Index	Status	Duration (sec.)
(no filter)	(no filter)	(no filter)	(no filter)
/tests/nickel_Faults_Group_0	1	passed	

Showing 1 item

Col #	Name	Value
(no filte	(no filter)	
	AbstractCO Annotation	CD
	AbstractFO Annotation	FU
	Enclosing Entity	test_drink.top.vendi
	Exclusion Rule Type	None
2	Fault Annotation	UD

Prepare › Minimize › Execute › Report › Analyze

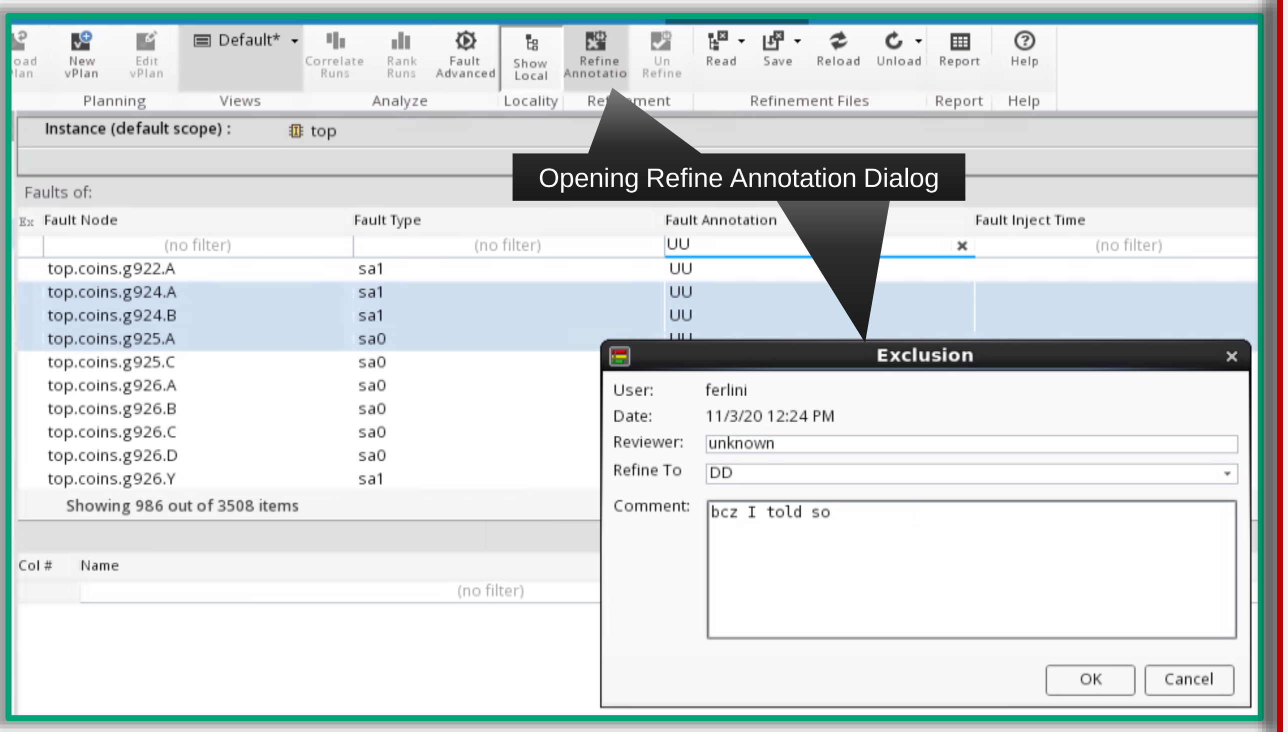
Fault Annotation Refinement

Dispositioning unclassified faults

```
vmanager> refine_annotation -faults {top.vending1.\\current_state_reg\[3\] } \
             -fault_type SEU -refineTo S -comment {bcz I want}
...
A total of 8 faults were refined to S
vmanager> save -refinement fcm_refinement.vRefine
vmanager> fi_campaign -report -overwrite -output fcm_refined_report
Writing report to: fcm_refined_report/faultsim_stat_summary.report
```

- GUI and CLI →

↓



Opening Refine Annotation Dialog

Instance (default scope) : top

Faults of:

Fault Node	Fault Type	Fault Annotation	Fault Inject Time
(no filter)	(no filter)	UU	(no filter)
top.coins.g922.A	sa1	UU	
top.coins.g924.A	sa1	UU	
top.coins.g924.B	sa1	UU	
top.coins.g925.A	sa0	UU	
top.coins.g925.C	sa0		
top.coins.g926.A	sa0		
top.coins.g926.B	sa0		
top.coins.g926.C	sa0		
top.coins.g926.D	sa0		
top.coins.g926.Y	sa1		

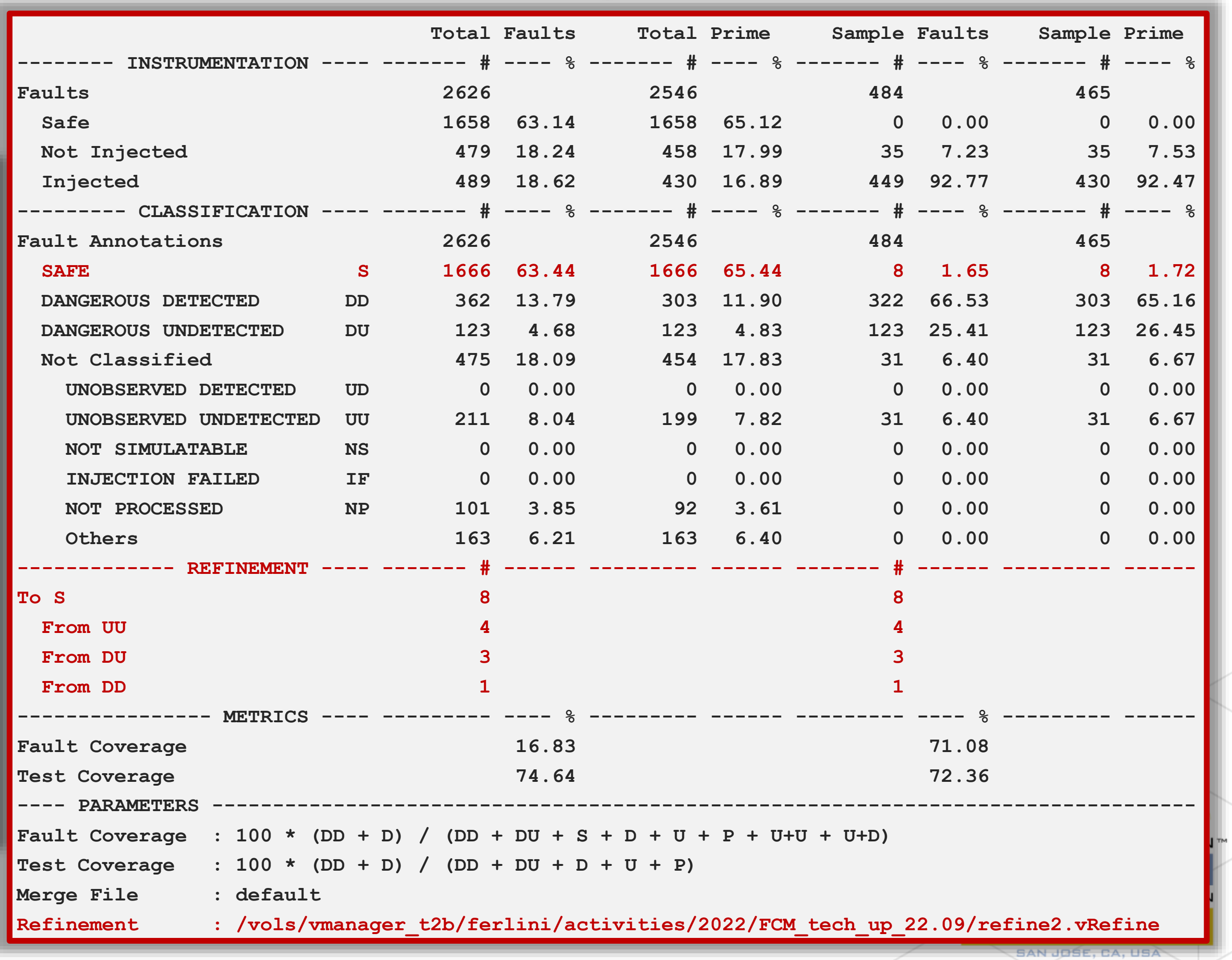
Showing 986 out of 3508 items

Col #	Name
	(no filter)

Exclusion

User: ferlini
Date: 11/3/20 12:24 PM
Reviewer: unknown
Refine To: DD
Comment: bcz I told so

OK Cancel



```
                                          Total Faults      Total Prime      Sample Faults     Sample Prime
-------- INSTRUMENTATION ---- -------- # ---- % -------- # ---- % -------- # ---- % -------- # ---- %
Faults                                    2626              2546              484               465
  Safe                                    1658  63.14       1658  65.12         0   0.00          0   0.00
  Not Injected                             479  18.24        458  17.99        35   7.23         35   7.53
  Injected                                 489  18.62        430  16.89       449  92.77        430  92.47
--------- CLASSIFICATION ---- -------- # ---- % -------- # ---- % -------- # ---- % -------- # ---- %
Fault Annotations                         2626              2546              484               465
  SAFE                          S         1666  63.44       1666  65.44         8   1.65          8   1.72
  DANGEROUS DETECTED            DD         362  13.79        303  11.90       322  66.53        303  65.16
  DANGEROUS UNDETECTED          DU         123   4.68        123   4.83       123  25.41        123  26.45
  Not Classified                           475  18.09        454  17.83        31   6.40         31   6.67
    UNOBSERVED DETECTED         UD           0   0.00          0   0.00         0   0.00          0   0.00
    UNOBSERVED UNDETECTED       UU         211   8.04        199   7.82        31   6.40         31   6.67
    NOT SIMULATABLE             NS           0   0.00          0   0.00         0   0.00          0   0.00
    INJECTION FAILED            IF           0   0.00          0   0.00         0   0.00          0   0.00
    NOT PROCESSED               NP         101   3.85         92   3.61         0   0.00          0   0.00
    Others                                 163   6.21        163   6.40         0   0.00          0   0.00
------------- REFINEMENT ---- -------- # ------ --------- ------ ------- # ------ --------- ------
To S                                         8                                  8
  From UU                                    4                                  4
  From DU                                    3                                  3
  From DD                                    1                                  1
--------------- METRICS ---- --------- ---- % --------- ------ --------- ---- % --------- ------
Fault Coverage                                 16.83                              71.08
Test Coverage                                  74.64                              72.36
---- PARAMETERS ---------------------------------------------------------------------------------
Fault Coverage   : 100 * (DD + D) / (DD + DU + S + D + U + P + U+U + U+D)
Test Coverage    : 100 * (DD + D) / (DD + DU + D + U + P)
Merge File       : default
Refinement       : /vols/vmanager_t2b/ferlini/activities/2022/FCM_tech_up_22.09/refine2.vRefine
```

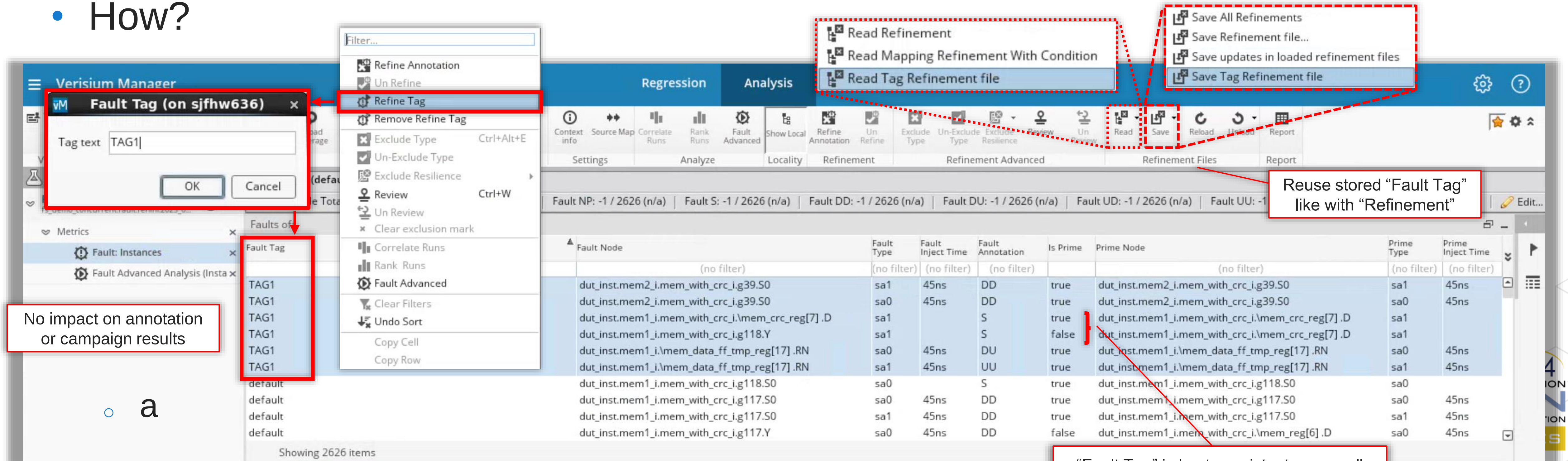
Prepare → Minimize → Execute → Report → Analyze

SAN JOSE, CA, USA
MARCH 4-7, 2024

cadence

Fault Tagging

- What?
  - User-editable (string) attribute per fault metric element
- Why?
  - Support post-campaign analysis (debug, refinement, etc.) by tagging relevant faults
    - Logically gather faults even if they do not share a common attribute value (e.g., hierarchy, annotation)
- How?
  - a



Fault NP: -1 / 2626 (n/a) | Fault S: -1 / 2626 (n/a) | Fault DD: -1 / 2626 (n/a) | Fault DU: -1 / 2626 (n/a) | Fault UD: -1 / 2626 (n/a) | Fault UU: -1

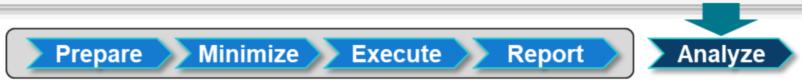
Fault Tag	Fault Node	Fault Type	Fault Inject Time	Fault Annotation	Is Prime	Prime Node	Prime Type	Prime Inject Time
	(no filter)	(no filter)	(no filter)	(no filter)		(no filter)	(no filter)	(no filter)
TAG1	dut_inst.mem2_i.mem_with_crc_i.g39.S0	sa1	45ns	DD	true	dut_inst.mem2_i.mem_with_crc_i.g39.S0	sa1	45ns
TAG1	dut_inst.mem2_i.mem_with_crc_i.g39.S0	sa0	45ns	DD	true	dut_inst.mem2_i.mem_with_crc_i.g39.S0	sa0	45ns
TAG1	dut_inst.mem1_i.mem_with_crc_i.\mem_crc_reg[7] .D	sa1		S	true	dut_inst.mem1_i.mem_with_crc_i.\mem_crc_reg[7] .D	sa1	
TAG1	dut_inst.mem1_i.mem_with_crc_i.g118.Y	sa1		S	false	dut_inst.mem1_i.mem_with_crc_i.\mem_crc_reg[7] .D	sa1	
TAG1	dut_inst.mem1_i.\mem_data_ff_tmp_reg[17] .RN	sa0	45ns	DU	true	dut_inst.mem1_i.\mem_data_ff_tmp_reg[17] .RN	sa0	45ns
TAG1	dut_inst.mem1_i.\mem_data_ff_tmp_reg[17] .RN	sa1	45ns	UU	true	dut_inst.mem1_i.\mem_data_ff_tmp_reg[17] .RN	sa1	45ns
default	dut_inst.mem1_i.mem_with_crc_i.g118.S0	sa0		S	true	dut_inst.mem1_i.mem_with_crc_i.g118.S0	sa0	
default	dut_inst.mem1_i.mem_with_crc_i.g117.S0	sa0	45ns	DD	true	dut_inst.mem1_i.mem_with_crc_i.g117.S0	sa0	45ns
default	dut_inst.mem1_i.mem_with_crc_i.g117.S0	sa1	45ns	DD	true	dut_inst.mem1_i.mem_with_crc_i.g117.S0	sa1	45ns
default	dut_inst.mem1_i.mem_with_crc_i.g117.Y	sa0	45ns	DD	false	dut_inst.mem1_i.mem_with_crc_i.\mem_reg[6] .D	sa0	45ns

Showing 2626 items

No impact on annotation or campaign results

Reuse stored “Fault Tag” like with “Refinement”

“Fault Tag” is kept consistent across all equivalent faults automatically



Scripted Annotation Refinement Leveraging Fault Tagging

- Load fault session and apply tags

```
vmanager> load fs_demo_concurrent.fault.ferlini.2023_01_20_09_04_38

vmanager> refine_tag -faults {dut_inst.mem2_i.mem_with_crc_i.g39.S0} -refineTo TAG1
vmanager> refine_tag -faults {dut_inst.mem1_i.\\mem_data_ff_tmp_reg\[17] .RN} -refineTo TAG1
vmanager> refine_tag -faults {dut_inst.mem1_i.mem_with_crc_i.\\mem_crc_reg\[7] .D} -fault_type sa1 -refineTo TAG1
```

```
Note: Wildcard '*' is supported in -faults <value>
```

- Save tags (e.g., open in GUI)
  - Optional – export filtered CSV

```
vmanager> save -fault_tag -refinement tech_up_cli.vRefineTag

vmanager> csv_export -metrics -fault -filter {fault_tag:==TAG1} -view MY -inst ... -out red.csv
```

```
Fault Tag,Fault Annotation,Fault Type,Fault Node,Fault Inject Time
TAG1,S,sa1,dut_inst.mem1_i.mem_with_crc_i.\mem_crc_reg[7] .D,
TAG1,S,sa1,dut_inst.mem1_i.mem_with_crc_i.g118.Y,
TAG1,DU,sa0,dut_inst.mem1_i.\mem_data_ff_tmp_reg[17] .RN,45ns
TAG1,UU,sa1,dut_inst.mem1_i.\mem_data_ff_tmp_reg[17] .RN,45ns
TAG1,DD,sa0,dut_inst.mem2_i.mem_with_crc_i.g39.S0,45ns
TAG1,DD,sa1,dut_inst.mem2_i.mem_with_crc_i.g39.S0,45ns
```

- Tag-based annotation refinement
  - Optional – export filtered CSV

```
vmanager> refine_annotation -tag_name TAG1 -refineTo S -comment {bcz...}

vmanager> csv_export -metrics -fault -filter {fault_tag:==TAG1} -view CLI -inst ... -out blue.csv
```

```
Fault Tag,Fault Annotation,Fault Type,Fault Node,Fault Inject Time
TAG1,S,sa1,dut_inst.mem1_i.mem_with_crc_i.\mem_crc_reg[7] .D,
TAG1,S,sa1,dut_inst.mem1_i.mem_with_crc_i.g118.Y,
TAG1,S,sa0,dut_inst.mem1_i.\mem_data_ff_tmp_reg[17] .RN,45ns
TAG1,S,sa1,dut_inst.mem1_i.\mem_data_ff_tmp_reg[17] .RN,45ns
TAG1,S,sa0,dut_inst.mem2_i.mem_with_crc_i.g39.S0,45ns
TAG1,S,sa1,dut_inst.mem2_i.mem_with_crc_i.g39.S0,45ns
```

- Generate updated summary
  - Must save refinement

```
vmanager> save -refinement tag_based.vRefine

vmanager> fi_campaign -report -summary -output refined_summary_rpt
```

```
...
------------- REFINEMENT ---- ------- # ------ --------- ------ ------- # ------ --------- ------
To S                                  4                                 4
  From DD                             2                                 2
  From UU                             1                                 1
  From DU                             1                                 1
---------------- METRICS ---- --------- ---- % --------- ------ --------- ---- % --------- ------
Fault Coverage                           24.97                             71.13
Test Coverage                            71.44                             71.44
---- PARAMETERS ---------------------------------------------------------------------------------
Fault Coverage   : 100 * (DD + D) / (DD + DU + S + D + U + P + U+U + U+D)
Test Coverage    : 100 * (DD + D) / (DD + DU + D + U + P)
Merge File       : default
Refinement       : tag_based.vRefine
```

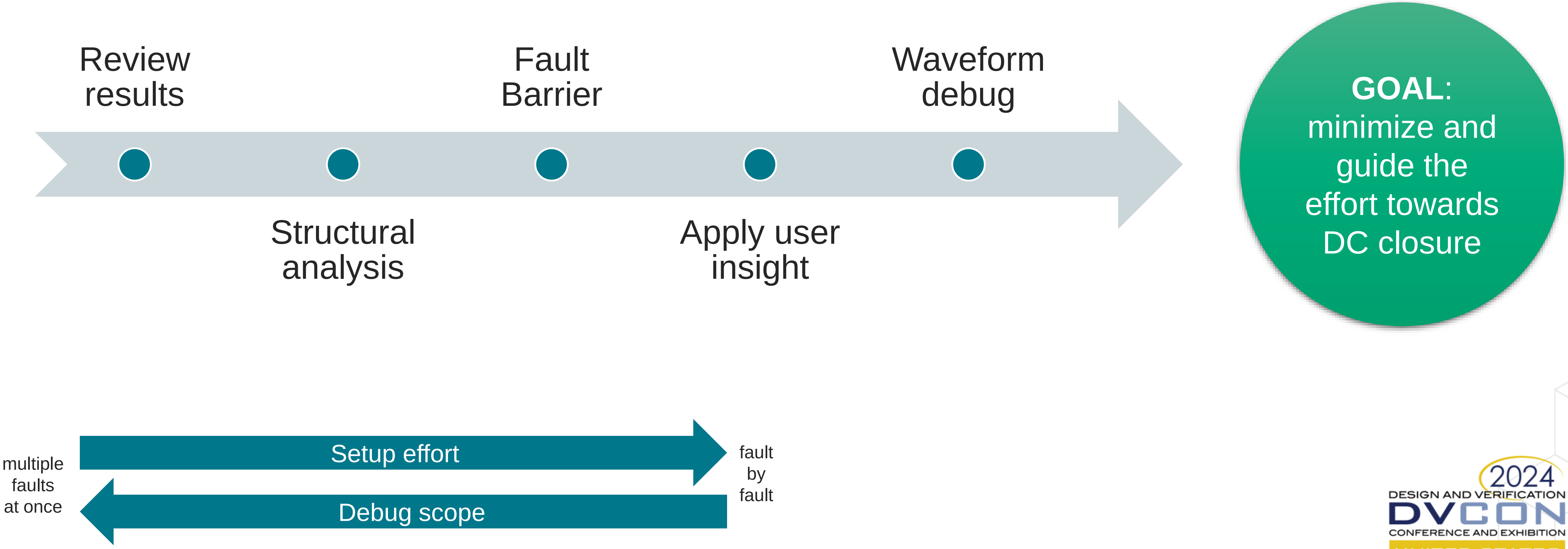
Prepare → Minimize → Execute → Report → Analyze

MARCH 4-7, 2024

cādence®

Fault Campaign Debug

Fault Campaign Closure



Review results

Structural analysis

Fault Barrier

Apply user insight

Waveform debug

GOAL: minimize and guide the effort towards DC closure

multiple faults at once

Setup effort

Debug scope

fault by fault

Approach 1 – Fault Analysis to Improve Tests

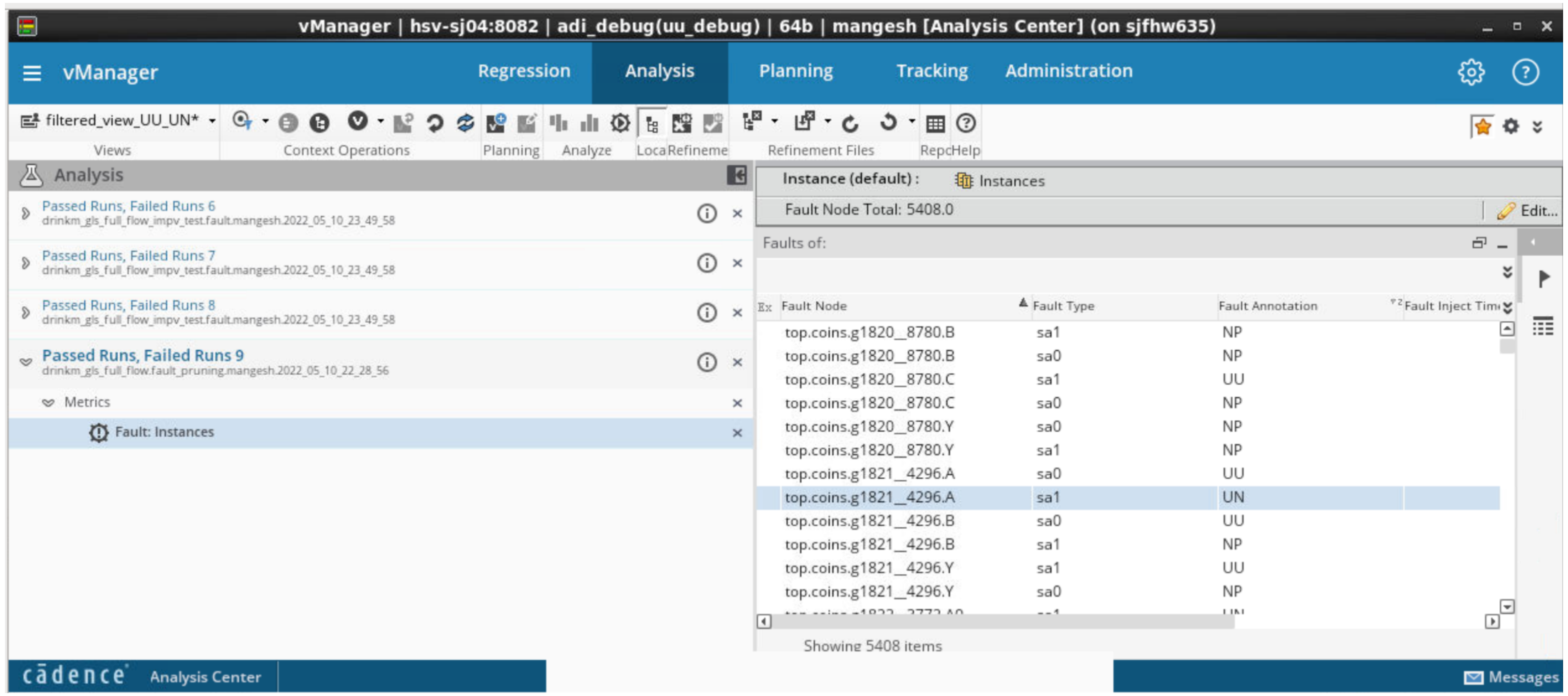
Configuration Superset

Reference Fault Campaign with FST,FSV TC Enabled in Verisium^{TM} Manager Safety

Reference Fault Campaign Results

Metrics, Report Analysis

Hierarchical, Filtering, Test Analysis, Report Reviews



Passed Runs, Failed Runs 6
drinkm_gls_full_flow_impv_test.fault.mangesh.2022_05_10_23_49_58

Passed Runs, Failed Runs 7
drinkm_gls_full_flow_impv_test.fault.mangesh.2022_05_10_23_49_58

Passed Runs, Failed Runs 8
drinkm_gls_full_flow_impv_test.fault.mangesh.2022_05_10_23_49_58

Passed Runs, Failed Runs 9
drinkm_gls_full_flow.fault_pruning.mangesh.2022_05_10_22_28_56

Fault Node Total: 5408.0

Ex	Fault Node	Fault Type	Fault Annotation	Fault Inject Tim
	top.coins.g1820__8780.B	sa1	NP	
	top.coins.g1820__8780.B	sa0	NP	
	top.coins.g1820__8780.C	sa1	UU	
	top.coins.g1820__8780.C	sa0	NP	
	top.coins.g1820__8780.Y	sa0	NP	
	top.coins.g1820__8780.Y	sa1	NP	
	top.coins.g1821__4296.A	sa0	UU	
	top.coins.g1821__4296.A	sa1	UN	
	top.coins.g1821__4296.B	sa0	UU	
	top.coins.g1821__4296.B	sa1	NP	
	top.coins.g1821__4296.Y	sa1	UU	
	top.coins.g1821__4296.Y	sa0	NP	

Showing 5408 items

Configuration With new tests

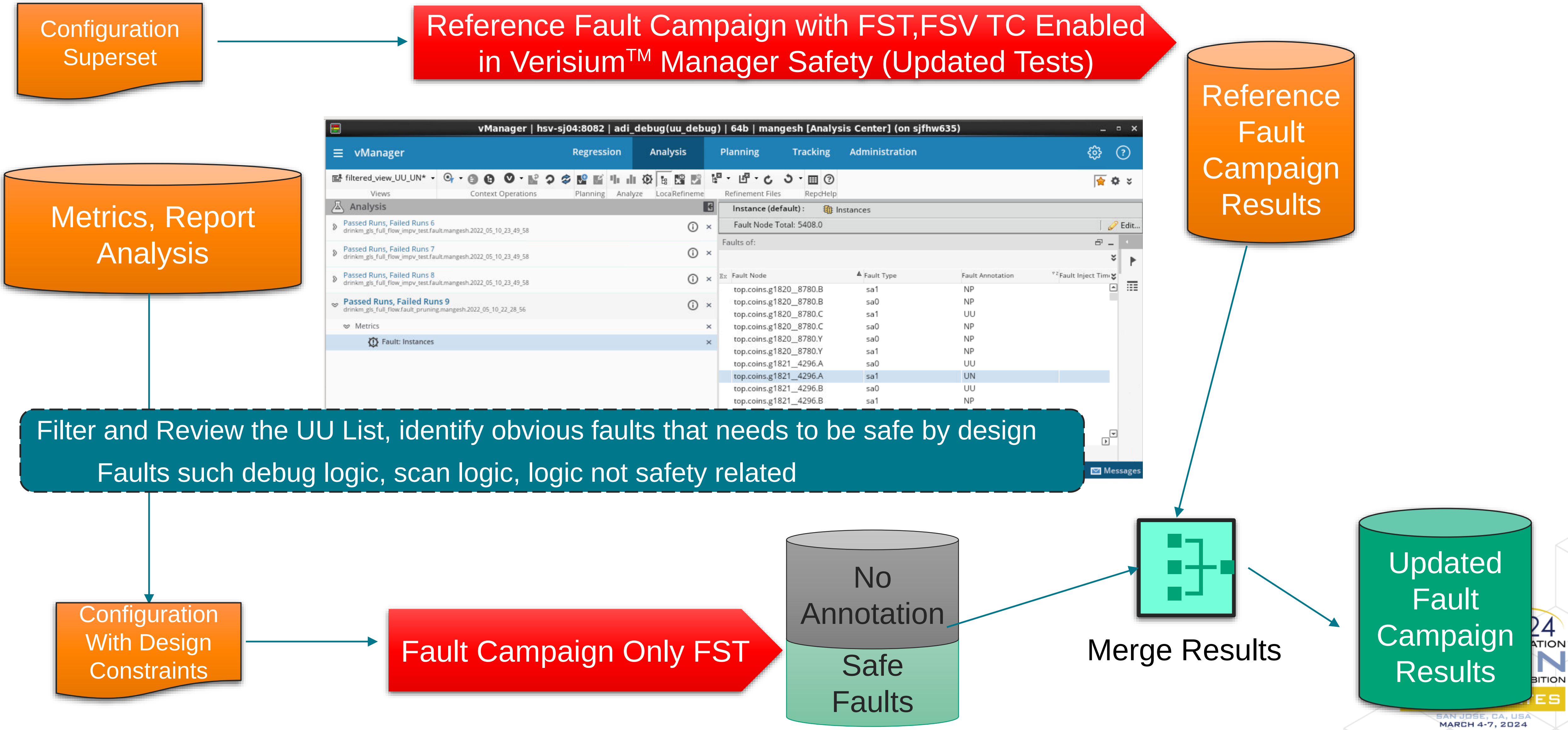
Updated tests based Fault Campaign with FST,FSV TC Enabled in vManager Safety

Updated Fault Campaign Results

Add additional tests, modify tests to cover the UU Faults

SAN JOSE, CA, USA
MARCH 4-7, 2024

Approach 2 – Design Constraints to Jasper FSV



Configuration Superset
Reference Fault Campaign with FST,FSV TC Enabled in Verisium™ Manager Safety (Updated Tests)
Reference Fault Campaign Results
Metrics, Report Analysis
Configuration With Design Constraints
Fault Campaign Only FST
No Annotation
Safe Faults
Merge Results
Updated Fault Campaign Results

Filter and Review the UU List, identify obvious faults that needs to be safe by design

Faults such debug logic, scan logic, logic not safety related

vManager | hsv-sj04:8082 | adi_debug(uu_debug) | 64b | mangesh [Analysis Center] (on sjfhw635)

Passed Runs, Failed Runs 6
drinkm_gls_full_flow_impv_test.fault.mangesh.2022_05_10_23_49_58

Passed Runs, Failed Runs 7
drinkm_gls_full_flow_impv_test.fault.mangesh.2022_05_10_23_49_58

Passed Runs, Failed Runs 8
drinkm_gls_full_flow_impv_test.fault.mangesh.2022_05_10_23_49_58

Passed Runs, Failed Runs 9
drinkm_gls_full_flow.fault_pruning.mangesh.2022_05_10_22_28_56

Metrics

Fault: Instances

Instance (default) : Instances

Fault Node Total: 5408.0

Edit...

Faults of:

Fault Node	Fault Type	Fault Annotation	Fault Inject Time
top.coins.g1820__8780.B	sa1	NP	
top.coins.g1820__8780.B	sa0	NP	
top.coins.g1820__8780.C	sa1	UU	
top.coins.g1820__8780.C	sa0	NP	
top.coins.g1820__8780.Y	sa0	NP	
top.coins.g1820__8780.Y	sa1	NP	
top.coins.g1821__4296.A	sa0	UU	
top.coins.g1821__4296.A	sa1	UN	
top.coins.g1821__4296.B	sa0	UU	
top.coins.g1821__4296.B	sa1	NP	

SAN JOSE, CA, USA
MARCH 4-7, 2024

cadence®

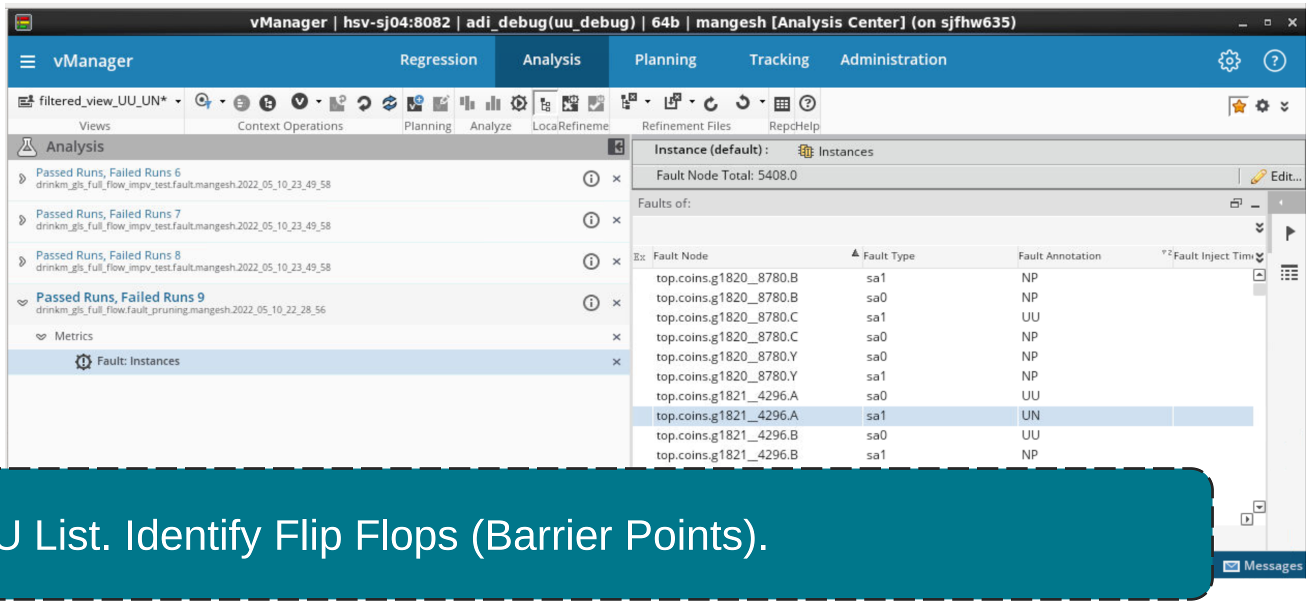
Approach 3 – Fault Refinement

Configuration Superset

Reference Fault Campaign with FST,FSV TC Enabled in Verisium™ Manager Safety (Updated Tests, barrier constraints)

Reference Fault Campaign Results

Metrics, Report Analysis



vManager | hsv-sj04:8082 | adi_debug(uu_debug) | 64b | mangesh [Analysis Center] (on sjfhw635)

- Passed Runs, Failed Runs 6 — drinkm_gls_full_flow_impv_test.fault.mangesh.2022_05_10_23_49_58
- Passed Runs, Failed Runs 7 — drinkm_gls_full_flow_impv_test.fault.mangesh.2022_05_10_23_49_58
- Passed Runs, Failed Runs 8 — drinkm_gls_full_flow_impv_test.fault.mangesh.2022_05_10_23_49_58
- Passed Runs, Failed Runs 9 — drinkm_gls_full_flow.fault_pruning.mangesh.2022_05_10_22_28_56
  - Metrics
    - Fault: Instances

Instance (default) : Instances

Fault Node Total: 5408.0

Faults of:

Ex	Fault Node	Fault Type	Fault Annotation	Fault Inject Tim
	top.coins.g1820__8780.B	sa1	NP	
	top.coins.g1820__8780.B	sa0	NP	
	top.coins.g1820__8780.C	sa1	UU	
	top.coins.g1820__8780.C	sa0	NP	
	top.coins.g1820__8780.Y	sa0	NP	
	top.coins.g1820__8780.Y	sa1	NP	
	top.coins.g1821__4296.A	sa0	UU	
	top.coins.g1821__4296.A	sa1	UN	
	top.coins.g1821__4296.B	sa0	UU	
	top.coins.g1821__4296.B	sa1	NP	

Filter and Review the UU List. Identify Flip Flops (Barrier Points).

Configuration With Barrier Tcl and UU Fault List from Reference Campaign

Identify faults belonging exclusively to the fan-in of given FFs

Fault Campaign Only FST

No Annotation

Safe Faults

Refinement TCL

Updated Fault Campaign Results

SAN JOSE, CA, USA
MARCH 4-7, 2024

Approach 4 – Enable Formal

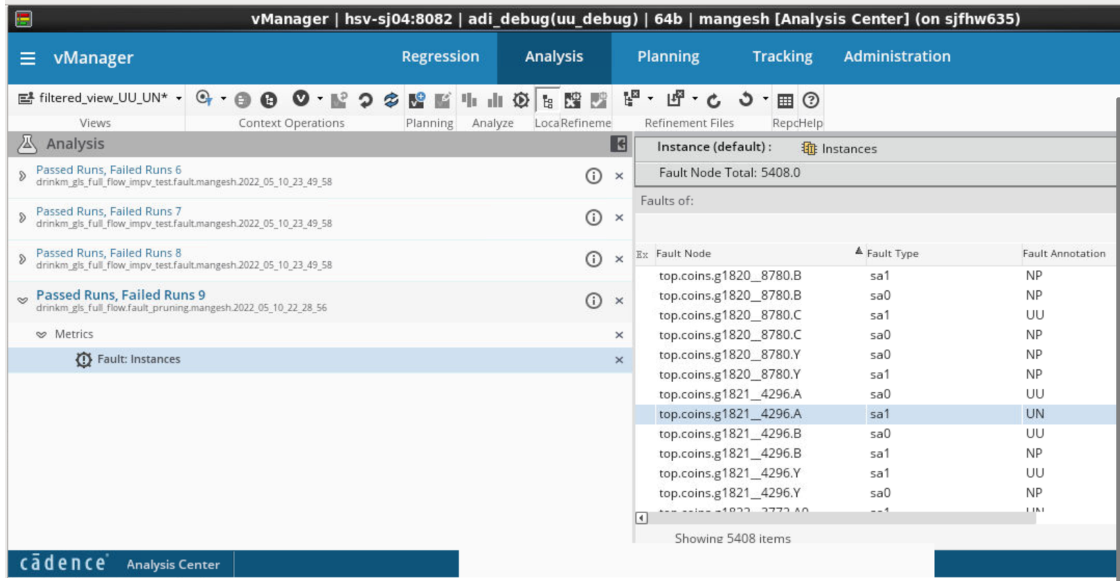
Configuration Superset

Available Reference Fault Campaign till FST in Verisium^{TM} Manager Safety (Updated Tests, Constraints)

Reference Fault Campaign Results

Metrics, Report Analysis

Filter and Review the UU List


vManager | hsv-sj04:8082 | adi_debug(uu_debug) | 64b | mangesh [Analysis Center] (on sjfhw635)

Passed Runs, Failed Runs 6
drinkm_gls_full_flow_impv_test.fault.mangesh.2022_05_10_23_49_58

Passed Runs, Failed Runs 7
drinkm_gls_full_flow_impv_test.fault.mangesh.2022_05_10_23_49_58

Passed Runs, Failed Runs 8
drinkm_gls_full_flow_impv_test.fault.mangesh.2022_05_10_23_49_58

Passed Runs, Failed Runs 9
drinkm_gls_full_flow.fault_pruning.mangesh.2022_05_10_22_28_56

Metrics

Fault: Instances

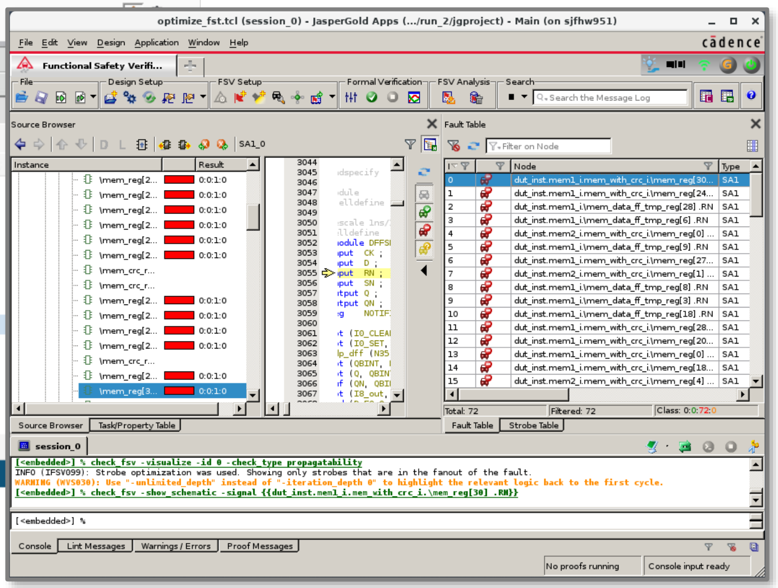
Instance (default) : Instances

Fault Node Total: 5408.0

Faults of:

Ex	Fault Node	Fault Type	Fault Annotation
	top.coins.g1820__8780.B	sa1	NP
	top.coins.g1820__8780.B	sa0	NP
	top.coins.g1820__8780.C	sa1	UU
	top.coins.g1820__8780.C	sa0	NP
	top.coins.g1820__8780.Y	sa0	NP
	top.coins.g1820__8780.Y	sa1	NP
	top.coins.g1821__4296.A	sa0	UU
	top.coins.g1821__4296.A	sa1	UN
	top.coins.g1821__4296.B	sa0	UU
	top.coins.g1821__4296.B	sa1	NP
	top.coins.g1821__4296.Y	sa1	UU
	top.coins.g1821__4296.Y	sa0	NP

Showing 5408 items


optimize_fst.tcl (session_0) - JasperGold Apps (.../run_2/jgproject) - Main (on sjfhw951)

I		Node	Type
0		dut_inst.mem1_i.mem_with_crc_i.\mem_reg[30...	SA1
1		dut_inst.mem1_i.mem_with_crc_i.\mem_reg[24...	SA1
2		dut_inst.mem1_i.\mem_data_ff_tmp_reg[28] .RN	SA1
3		dut_inst.mem1_i.\mem_data_ff_tmp_reg[6] .RN	SA1
4		dut_inst.mem2_i.mem_with_crc_i.\mem_reg[0] ...	SA1
5		dut_inst.mem1_i.\mem_data_ff_tmp_reg[9] .RN	SA1
6		dut_inst.mem1_i.mem_with_crc_i.\mem_reg[27...	SA1
7		dut_inst.mem2_i.mem_with_crc_i.\mem_reg[1] ...	SA1
8		dut_inst.mem1_i.\mem_data_ff_tmp_reg[8] .RN	SA1
9		dut_inst.mem1_i.\mem_data_ff_tmp_reg[3] .RN	SA1
10		dut_inst.mem1_i.\mem_data_ff_tmp_reg[18] .RN	SA1
11		dut_inst.mem1_i.mem_with_crc_i.\mem_reg[28...	SA1
12		dut_inst.mem1_i.mem_with_crc_i.\mem_reg[20...	SA1
13		dut_inst.mem1_i.mem_with_crc_i.\mem_reg[0] ...	SA1
14		dut_inst.mem1_i.mem_with_crc_i.\mem_reg[18...	SA1
15		dut_inst.mem2_i.mem_with_crc_i.\mem_reg[4] ...	SA1

Total: 72 Filtered: 72 Class: 0:0:72:0

session_0

```
[<embedded>] % check_fsv -visualize -id 0 -check_type propagatability
INFO (IFSV099): Strobe optimization was used. Showing only strobes that are in the fanout of the fault.
WARNING (WVS030): Use "-unlimited_depth" instead of "-iteration_depth 0" to highlight the relevant logic back to the first cycle.
[<embedded>] % check_fsv -show_schematic -signal {{dut_inst.mem1_i.mem_with_crc_i.\mem_reg[30] .RN}}

[<embedded>] %
```

Configuration

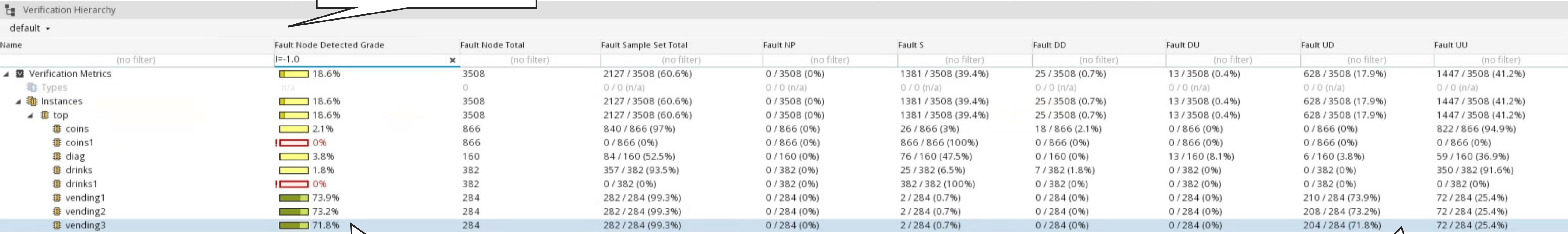
Rerun FST or Incremental Campaign with updated Fault List to invoke JG-FSV GUI

Debug Analysis, Safeness Annotation Refinement File

Fault Metric Analysis

- Customizable grade calculation
- Hierarchical results

Customizable Metric



Verification Hierarchy

default ▾

Name	Fault Node Detected Grade	Fault Node Total	Fault Sample Set Total	Fault NP	Fault S	Fault DD	Fault DU	Fault UD	Fault UU
(no filter)	I=-1.0 ×	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
Verification Metrics	18.6%	3508	2127 / 3508 (60.6%)	0 / 3508 (0%)	1381 / 3508 (39.4%)	25 / 3508 (0.7%)	13 / 3508 (0.4%)	628 / 3508 (17.9%)	1447 / 3508 (41.2%)
Types	n/a	0	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)
Instances	18.6%	3508	2127 / 3508 (60.6%)	0 / 3508 (0%)	1381 / 3508 (39.4%)	25 / 3508 (0.7%)	13 / 3508 (0.4%)	628 / 3508 (17.9%)	1447 / 3508 (41.2%)
top	18.6%	3508	2127 / 3508 (60.6%)	0 / 3508 (0%)	1381 / 3508 (39.4%)	25 / 3508 (0.7%)	13 / 3508 (0.4%)	628 / 3508 (17.9%)	1447 / 3508 (41.2%)
coins	2.1%	866	840 / 866 (97%)	0 / 866 (0%)	26 / 866 (3%)	18 / 866 (2.1%)	0 / 866 (0%)	0 / 866 (0%)	822 / 866 (94.9%)
coins1	! 0%	866	0 / 866 (0%)	0 / 866 (0%)	866 / 866 (100%)	0 / 866 (0%)	0 / 866 (0%)	0 / 866 (0%)	0 / 866 (0%)
diag	3.8%	160	84 / 160 (52.5%)	0 / 160 (0%)	76 / 160 (47.5%)	0 / 160 (0%)	13 / 160 (8.1%)	6 / 160 (3.8%)	59 / 160 (36.9%)
drinks	1.8%	382	357 / 382 (93.5%)	0 / 382 (0%)	25 / 382 (6.5%)	7 / 382 (1.8%)	0 / 382 (0%)	0 / 382 (0%)	350 / 382 (91.6%)
drinks1	! 0%	382	0 / 382 (0%)	0 / 382 (0%)	382 / 382 (100%)	0 / 382 (0%)	0 / 382 (0%)	0 / 382 (0%)	0 / 382 (0%)
vending1	73.9%	284	282 / 284 (99.3%)	0 / 284 (0%)	2 / 284 (0.7%)	0 / 284 (0%)	0 / 284 (0%)	210 / 284 (73.9%)	72 / 284 (25.4%)
vending2	73.2%	284	282 / 284 (99.3%)	0 / 284 (0%)	2 / 284 (0.7%)	0 / 284 (0%)	0 / 284 (0%)	208 / 284 (73.2%)	72 / 284 (25.4%)
vending3	71.8%	284	282 / 284 (99.3%)	0 / 284 (0%)	2 / 284 (0.7%)	0 / 284 (0%)	0 / 284 (0%)	204 / 284 (71.8%)	72 / 284 (25.4%)

leaf instances and fault nodes can be added for deep analysis

leaf instances and fault nodes can be added for deep analysis

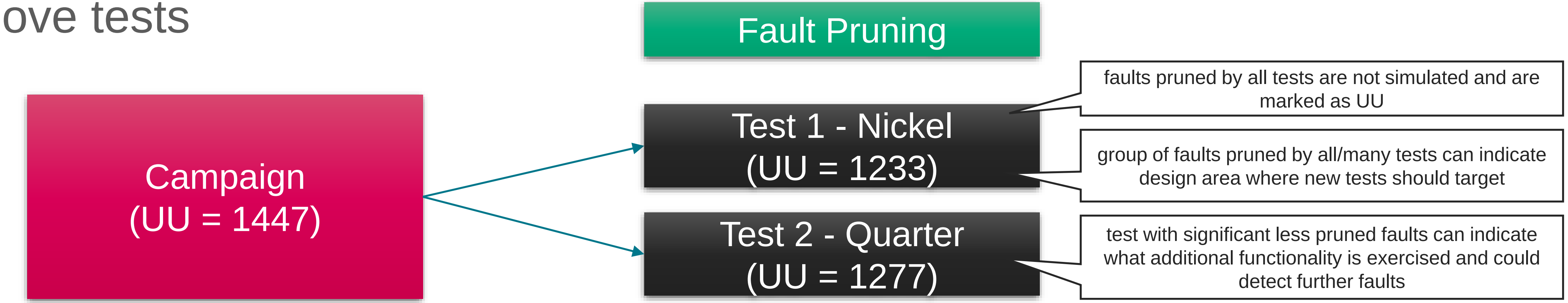
CSV dump support for post-processing

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES SAN JOSE, CA, USA MARCH 4-7, 2024

cadence®

Fault Pruning Results Per Test

Add / Remove tests


Fault Pruning

Campaign (UU = 1447)

Test 1 - Nickel (UU = 1233)

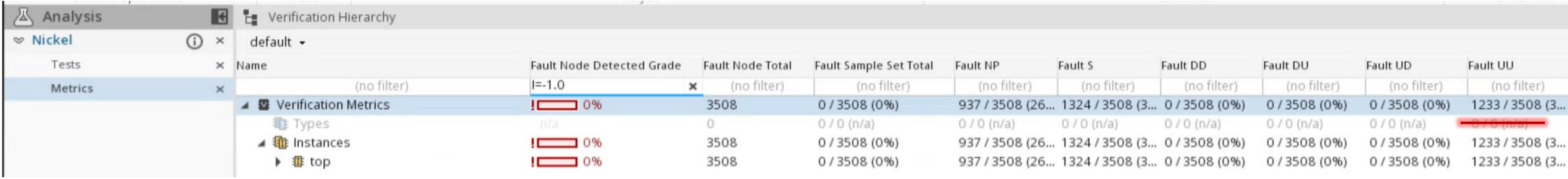
Test 2 - Quarter (UU = 1277)

faults pruned by all tests are not simulated and are marked as UU

group of faults pruned by all/many tests can indicate design area where new tests should target

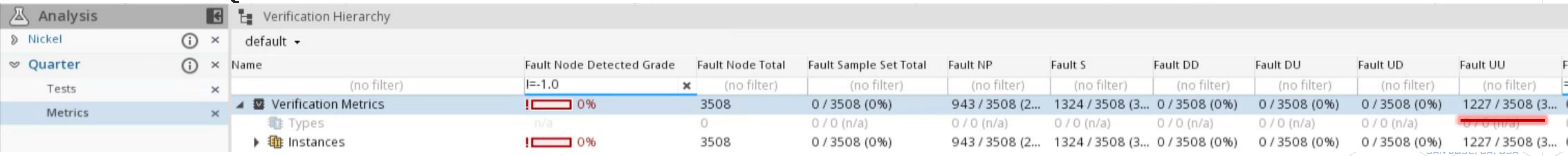
test with significant less pruned faults can indicate what additional functionality is exercised and could detect further faults

- Test 1 - Nickel



Name	Fault Node Detected Grade	Fault Node Total	Fault Sample Set Total	Fault NP	Fault S	Fault DD	Fault DU	Fault UD	Fault UU
(no filter)	!=-1.0	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
Verification Metrics	0%	3508	0 / 3508 (0%)	937 / 3508 (26...	1324 / 3508 (3...	0 / 3508 (0%)	0 / 3508 (0%)	0 / 3508 (0%)	1233 / 3508 (3...
Types	n/a	0	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	~~0 / 0 (n/a)~~
Instances	0%	3508	0 / 3508 (0%)	937 / 3508 (26...	1324 / 3508 (3...	0 / 3508 (0%)	0 / 3508 (0%)	0 / 3508 (0%)	1233 / 3508 (3...
top	0%	3508	0 / 3508 (0%)	937 / 3508 (26...	1324 / 3508 (3...	0 / 3508 (0%)	0 / 3508 (0%)	0 / 3508 (0%)	1233 / 3508 (3...

- Test 2 - Quarter



Name	Fault Node Detected Grade	Fault Node Total	Fault Sample Set Total	Fault NP	Fault S	Fault DD	Fault DU	Fault UD	Fault UU
(no filter)	!=-1.0	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
Verification Metrics	0%	3508	0 / 3508 (0%)	943 / 3508 (2...	1324 / 3508 (3...	0 / 3508 (0%)	0 / 3508 (0%)	0 / 3508 (0%)	1227 / 3508 (3...
Types	n/a	0	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	~~0 / 0 (n/a)~~
Instances	0%	3508	0 / 3508 (0%)	943 / 3508 (2...	1324 / 3508 (3...	0 / 3508 (0%)	0 / 3508 (0%)	0 / 3508 (0%)	1227 / 3508 (3...

MARCH 4-7, 2024

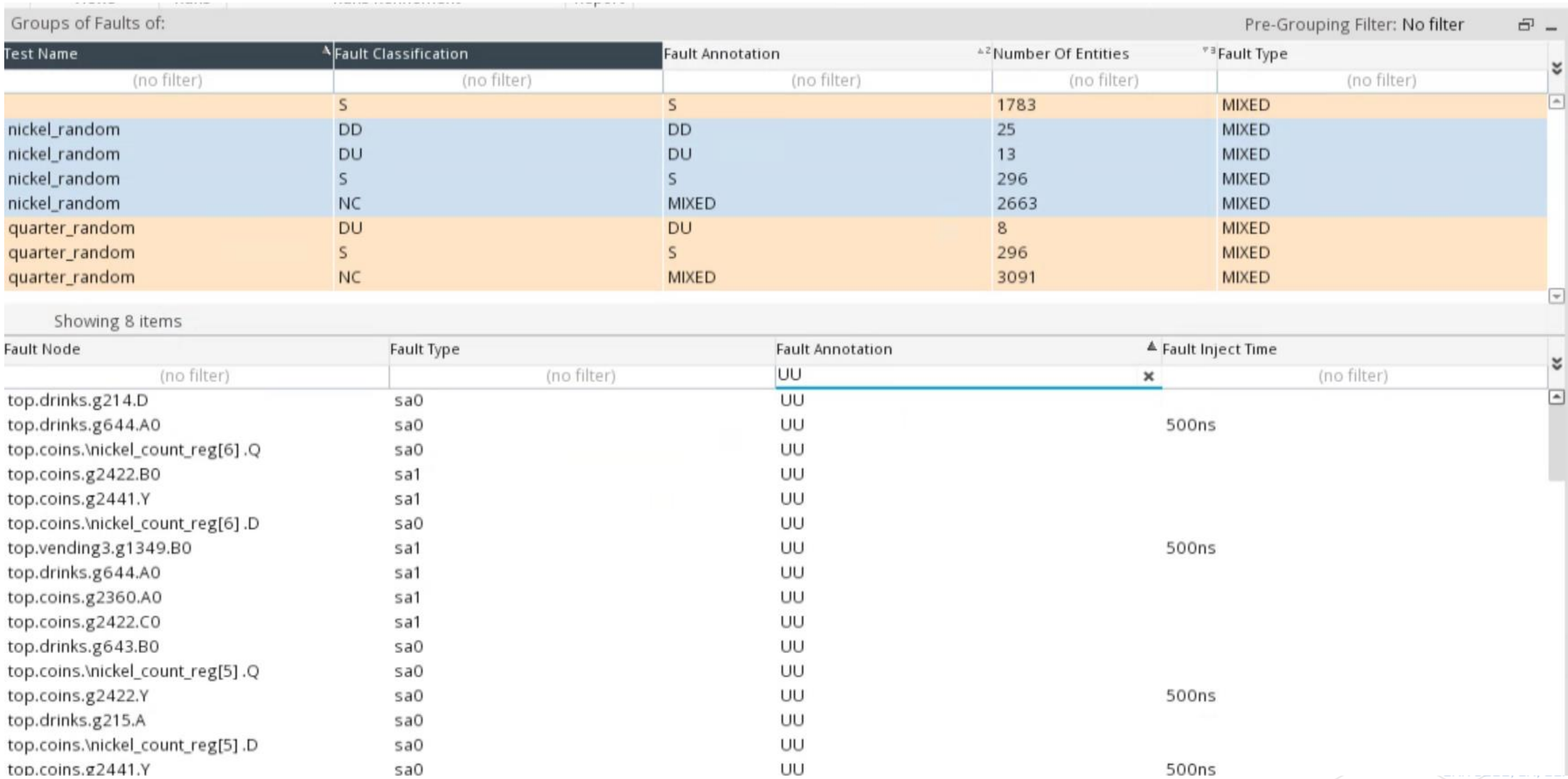
Individual Annotation Contribution of Each Test

Advanced Fault Analysis

- Results grouped by Test and Fault Classification

Test 1 - Nickel
DD : 25

Test 2 - Quarter
DD : 0 ??



Groups of Faults of:

Pre-Grouping Filter: No filter

Test Name	Fault Classification	Fault Annotation	Number Of Entities	Fault Type
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
	S	S	1783	MIXED
nickel_random	DD	DD	25	MIXED
nickel_random	DU	DU	13	MIXED
nickel_random	S	S	296	MIXED
nickel_random	NC	MIXED	2663	MIXED
quarter_random	DU	DU	8	MIXED
quarter_random	S	S	296	MIXED
quarter_random	NC	MIXED	3091	MIXED

Showing 8 items

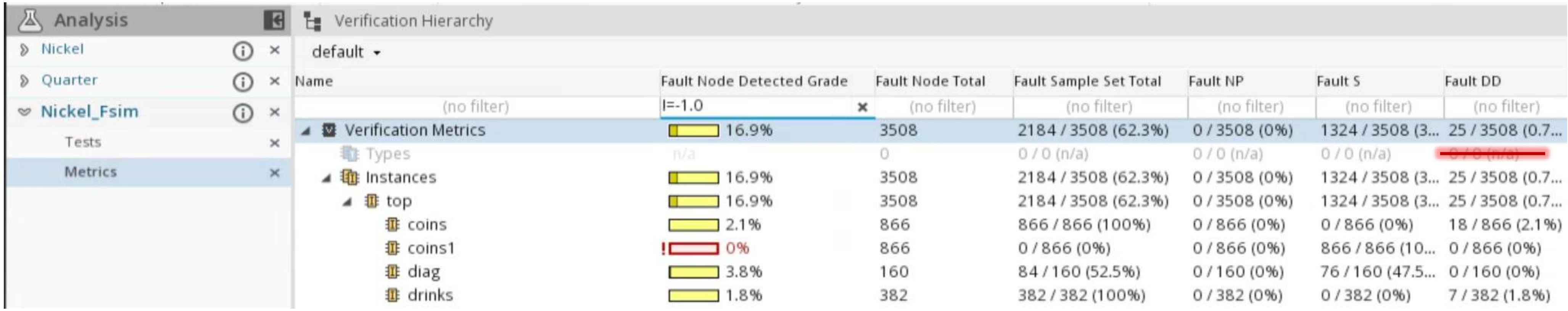
Fault Node	Fault Type	Fault Annotation	Fault Inject Time
(no filter)	(no filter)	UU	(no filter)
top.drinks.g214.D	sa0	UU	
top.drinks.g644.A0	sa0	UU	500ns
top.coins.\nickel_count_reg[6] .Q	sa0	UU	
top.coins.g2422.B0	sa1	UU	
top.coins.g2441.Y	sa1	UU	
top.coins.\nickel_count_reg[6] .D	sa0	UU	
top.vending3.g1349.B0	sa1	UU	500ns
top.drinks.g644.A0	sa1	UU	
top.coins.g2360.A0	sa1	UU	
top.coins.g2422.C0	sa1	UU	
top.drinks.g643.B0	sa0	UU	
top.coins.\nickel_count_reg[5] .Q	sa0	UU	
top.coins.g2422.Y	sa0	UU	500ns
top.drinks.g215.A	sa0	UU	
top.coins.\nickel_count_reg[5] .D	sa0	UU	
top.coins.g2441.Y	sa0	UU	500ns

MARCH 4-7, 2024

Fault Merged Annotation Per Each Test

Add / Remove tests

- Test 1 – Nickel



Analysis

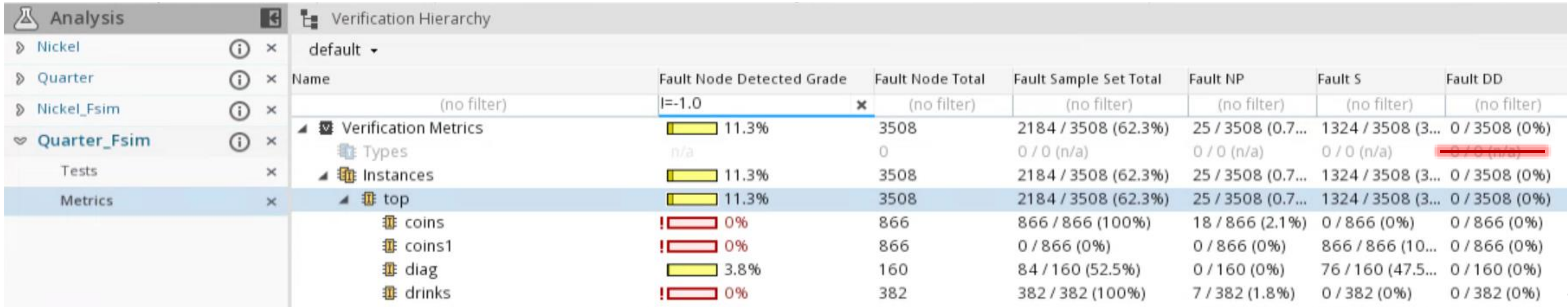
- Nickel
- Quarter
- Nickel_Fsim
  - Tests
  - Metrics

Verification Hierarchy

default

Name	Fault Node Detected Grade	Fault Node Total	Fault Sample Set Total	Fault NP	Fault S	Fault DD
(no filter)	!=-1.0	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
Verification Metrics	16.9%	3508	2184 / 3508 (62.3%)	0 / 3508 (0%)	1324 / 3508 (3...	25 / 3508 (0.7...
Types	n/a	0	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	~~0 / 0 (n/a)~~
Instances	16.9%	3508	2184 / 3508 (62.3%)	0 / 3508 (0%)	1324 / 3508 (3...	25 / 3508 (0.7...
top	16.9%	3508	2184 / 3508 (62.3%)	0 / 3508 (0%)	1324 / 3508 (3...	25 / 3508 (0.7...
coins	2.1%	866	866 / 866 (100%)	0 / 866 (0%)	0 / 866 (0%)	18 / 866 (2.1%)
coins1	! 0%	866	0 / 866 (0%)	0 / 866 (0%)	866 / 866 (10...	0 / 866 (0%)
diag	3.8%	160	84 / 160 (52.5%)	0 / 160 (0%)	76 / 160 (47.5...	0 / 160 (0%)
drinks	1.8%	382	382 / 382 (100%)	0 / 382 (0%)	0 / 382 (0%)	7 / 382 (1.8%)

- Test 2 - Quarter



Analysis

- Nickel
- Quarter
- Nickel_Fsim
- Quarter_Fsim
  - Tests
  - Metrics

Verification Hierarchy

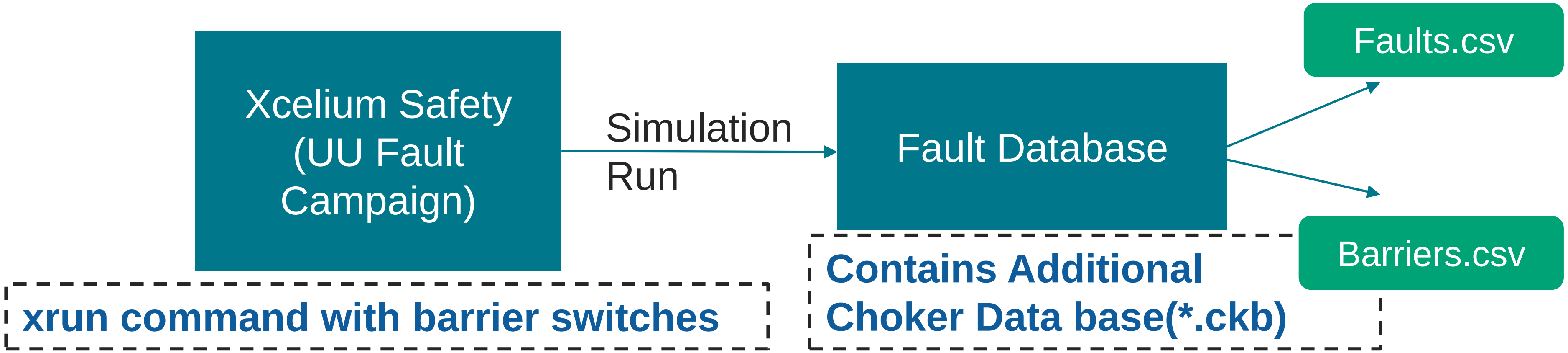
default

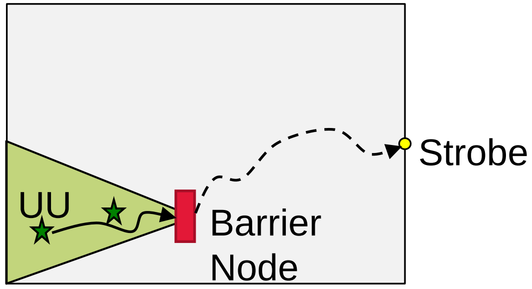
Name	Fault Node Detected Grade	Fault Node Total	Fault Sample Set Total	Fault NP	Fault S	Fault DD
(no filter)	!=-1.0	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
Verification Metrics	11.3%	3508	2184 / 3508 (62.3%)	25 / 3508 (0.7...	1324 / 3508 (3...	0 / 3508 (0%)
Types	n/a	0	0 / 0 (n/a)	0 / 0 (n/a)	0 / 0 (n/a)	~~0 / 0 (n/a)~~
Instances	11.3%	3508	2184 / 3508 (62.3%)	25 / 3508 (0.7...	1324 / 3508 (3...	0 / 3508 (0%)
top	11.3%	3508	2184 / 3508 (62.3%)	25 / 3508 (0.7...	1324 / 3508 (3...	0 / 3508 (0%)
coins	! 0%	866	866 / 866 (100%)	18 / 866 (2.1%)	0 / 866 (0%)	0 / 866 (0%)
coins1	! 0%	866	0 / 866 (0%)	0 / 866 (0%)	866 / 866 (10...	0 / 866 (0%)
diag	3.8%	160	84 / 160 (52.5%)	0 / 160 (0%)	76 / 160 (47.5...	0 / 160 (0%)
drinks	! 0%	382	382 / 382 (100%)	7 / 382 (1.8%)	0 / 382 (0%)	0 / 382 (0%)

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES SAN JOSE, CA, USA MARCH 4-7, 2024

cadence®

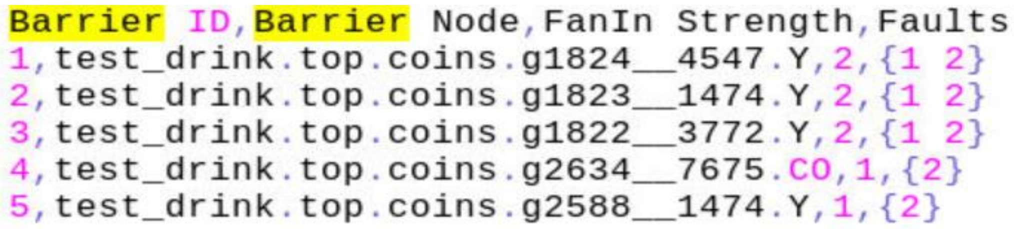
Functional Safety Flow: Barrier Analysis Details


Xcelium Safety (UU Fault Campaign)
Simulation Run
Fault Database
Faults.csv
Barriers.csv
xrun command with barrier switches
Contains Additional Choker Data base(*.ckb)



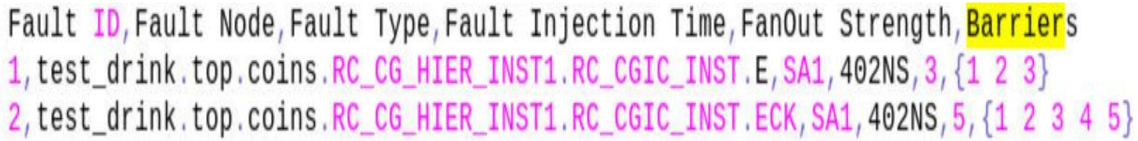
- Barrier Analysis executed on UU Faults to debug/identify block points
- Xcelium Safety supports barrier Analysis "–fault_barrier" switch to dump the data in Fault DB for every Fault Simulation
- Cadence developed Python utility is executed on fault_db to generate two files faults.csv and barrier.csv
    - barrier.csv -> captures the barriers and the associated blocked faults
        - contains the instance ; file name and line number of the code which block the fault propagation
    - faults.csv -> contains fault set and associated barriers for each of the fault nodes

- **Snippet of barriers.csv (Barrier to Fault Relation)**



```
Barrier ID,Barrier Node,FanIn Strength,Faults
1,test_drink.top.coins.g1824__4547.Y,2,{1 2}
2,test_drink.top.coins.g1823__1474.Y,2,{1 2}
3,test_drink.top.coins.g1822__3772.Y,2,{1 2}
4,test_drink.top.coins.g2634__7675.CO,1,{2}
5,test_drink.top.coins.g2588__1474.Y,1,{2}
```

- **Snippet of faults.csv (Faults to Barrier Mapping)**



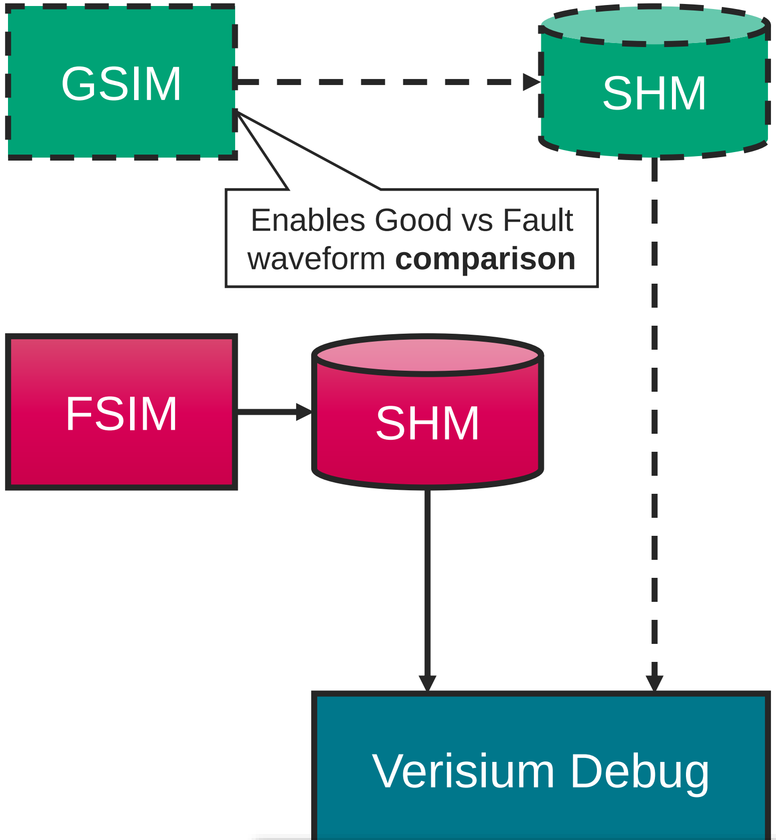
```
Fault ID,Fault Node,Fault Type,Fault Injection Time,FanOut Strength,Barriers
1,test_drink.top.coins.RC_CG_HIER_INST1.RC_CGIC_INST.E,SA1,402NS,3,{1 2 3}
2,test_drink.top.coins.RC_CG_HIER_INST1.RC_CGIC_INST.ECK,SA1,402NS,5,{1 2 3 4 5}
```

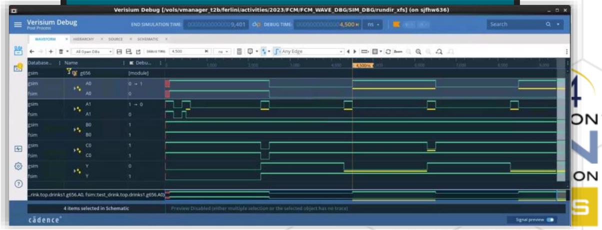
Waveform Generation

- 1. Good Simulation Waveform Generation (Xcelium)
  - Optional – Allows good vs fault waveform comparison
  - Concurrent
    - Note: probing signals is applicable to the good simulation by default

- 2. Fault Simulation Waveform Generation (Xcelium)
  - Serial
    - standard Xcelium probing mechanism
  - Concurrent
    - Enabled by -fault_dump_shm <id>
      - <id> is the fault id according to the injection order of the given run
      - i.e., 1st injected faults has id = 1, 2nd injected fault has id = 2…
      - Recommendation: only keep the fault being debug

- 3. Waveform Visualization (Verisium Debug)
  - Good vs Fault Waveform comparison





Rerun vs Incremental

- Rerun
  - Rerun (e.g., debug/exploration) data is mixed with original campaign data (
  - Risk: override valid results (unaffordable rerun to recover valid data)
  - Recommended when original data is invalid/unavailable

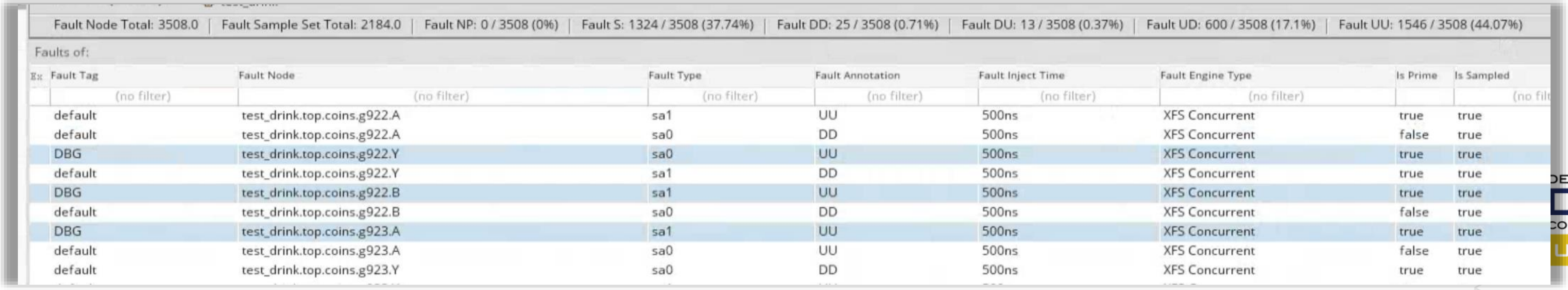
- Incremental
  - Separate set of data. Independent original and incremental analysis/results
  - Original and Incremental results can be analysed/reported independently or merged
  - Native support of fault sub-set selection based on its metrics (e.g., annotation)
    - Avoid reruns by skip already available optimisations results (e.g., analyzing UUs)
  - Flow type change support (e.g., concurrent → serial)
  - Support analysing faults sub-set with different stimulus

Fault Sub-Set Selection

Incremental campaign

- -fault_filter “attribute:value”
  - Select faults based on their attribute (e.g., annotation)
- -fault_limit <number>
  - Used to limit the number of selected faults for incremental campaign
- -runs_filter “attribute:value”
  - Selecting faults from specified runs of source campaign
  - E.g. –runs_filter “test_name:<run(s)_to_debug>”
- -fault_view / -runs_view
  - Instead of specifying the filters in batch, user can create filters via GUI and save the view
  - E.g. –fault_view “my_view_with_filters”
- -refinement_file “path/to/file.vRefine”
  - Used to apply user refinement on source campaign results, before applying the filters

```
fi_campaign \
  -launch <name> \
  -incremental <session(s)> \
  -flow_type <type> \
  [-cfg <cfg_path>] \
  [-refinement_file <file_path>] \
  [-fault_view <view_name>] \
  [-fault_filter <filter>] \
  [-runs_view <view_name>] \
  [-runs_filter <filter>] \
  [-fault_limit <num>]
  [-new_tests [-force_elab]]
```



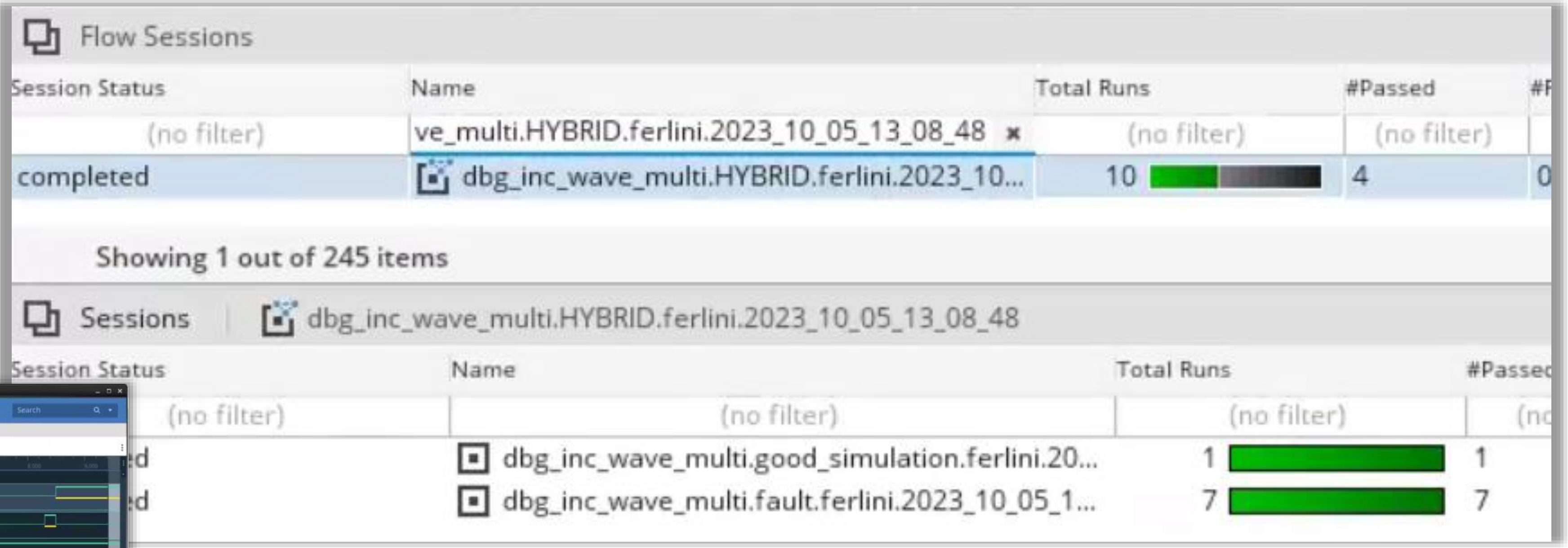
Fault Node Total: 3508.0 | Fault Sample Set Total: 2184.0 | Fault NP: 0 / 3508 (0%) | Fault S: 1324 / 3508 (37.74%) | Fault DD: 25 / 3508 (0.71%) | Fault DU: 13 / 3508 (0.37%) | Fault UD: 600 / 3508 (17.1%) | Fault UU: 1546 / 3508 (44.07%)

Faults of:

Ex	Fault Tag	Fault Node	Fault Type	Fault Annotation	Fault Inject Time	Fault Engine Type	Is Prime	Is Sampled
	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)	(no filter)		(no filt
	default	test_drink.top.coins.g922.A	sa1	UU	500ns	XFS Concurrent	true	true
	default	test_drink.top.coins.g922.A	sa0	DD	500ns	XFS Concurrent	false	true
	DBG	test_drink.top.coins.g922.Y	sa0	UU	500ns	XFS Concurrent	true	true
	default	test_drink.top.coins.g922.Y	sa1	DD	500ns	XFS Concurrent	true	true
	DBG	test_drink.top.coins.g922.B	sa1	UU	500ns	XFS Concurrent	true	true
	default	test_drink.top.coins.g922.B	sa0	DD	500ns	XFS Concurrent	false	true
	DBG	test_drink.top.coins.g923.A	sa1	UU	500ns	XFS Concurrent	true	true
	default	test_drink.top.coins.g923.A	sa0	UU	500ns	XFS Concurrent	false	true
	default	test_drink.top.coins.g923.Y	sa0	DD	500ns	XFS Concurrent	true	true

Waveform Debug

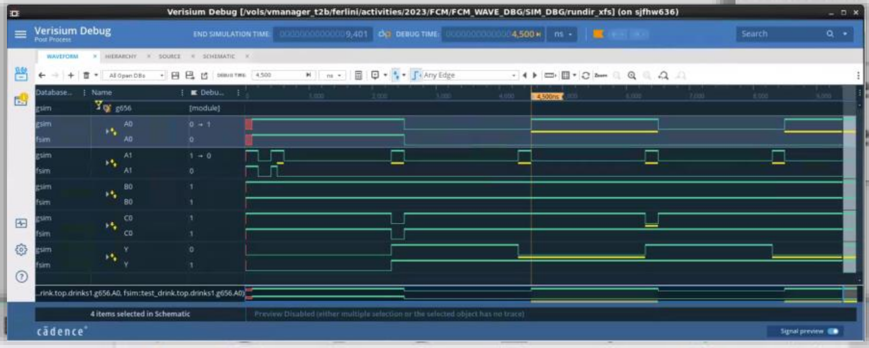
Incremental Campaign FCM integration



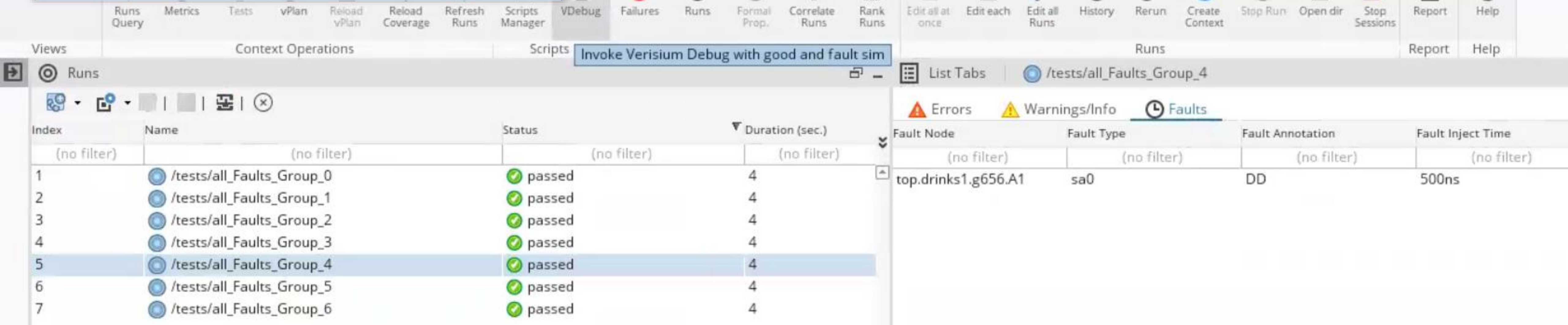
Session Status	Name	Total Runs	#Passed	#F
(no filter)	ve_multi.HYBRID.ferlini.2023_10_05_13_08_48	(no filter)	(no filter)	
completed	dbg_inc_wave_multi.HYBRID.ferlini.2023_10...	10	4	0

Showing 1 out of 245 items

Session Status	Name	Total Runs	#Passed
(no filter)	(no filter)	(no filter)	(nc
ed	dbg_inc_wave_multi.good_simulation.ferlini.20...	1	1
ed	dbg_inc_wave_multi.fault.ferlini.2023_10_05_1...	7	7



Verisium Debug [/vols/vmanager_t2b/ferlini/activities/2023/FCM/FCM_WAVE_DBG/SIM_DBG/rundir_xfs] (on sjfhw636)



Invoke Verisium Debug with good and fault sim

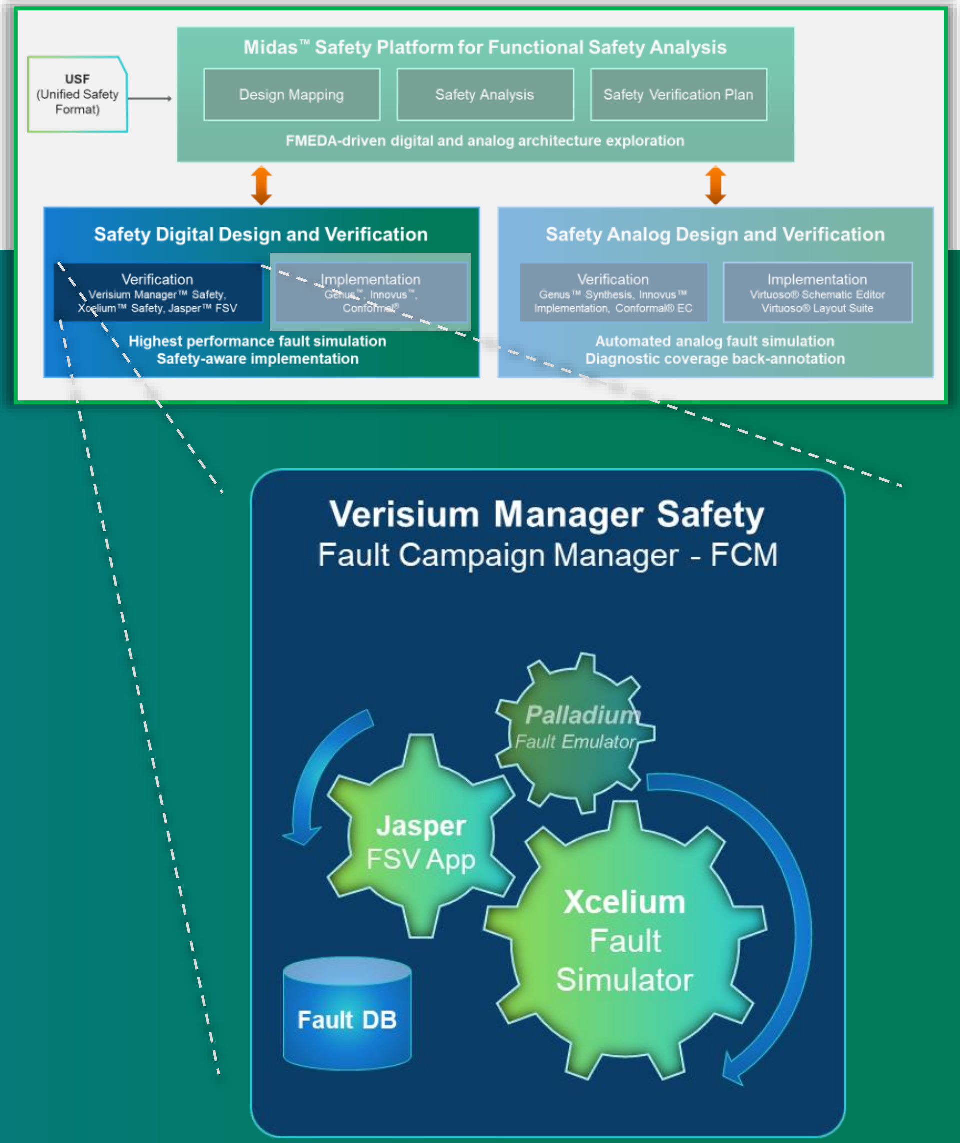
Index	Name	Status	Duration (sec.)
(no filter)	(no filter)	(no filter)	(no filter)
1	/tests/all_Faults_Group_0	passed	4
2	/tests/all_Faults_Group_1	passed	4
3	/tests/all_Faults_Group_2	passed	4
4	/tests/all_Faults_Group_3	passed	4
5	/tests/all_Faults_Group_4	passed	4
6	/tests/all_Faults_Group_5	passed	4
7	/tests/all_Faults_Group_6	passed	4

Fault Node	Fault Type	Fault Annotation	Fault Inject Time
(no filter)	(no filter)	(no filter)	(no filter)
top.drinks1.g656.A1	sa0	DD	500ns

Digital Safety Verification

Summary

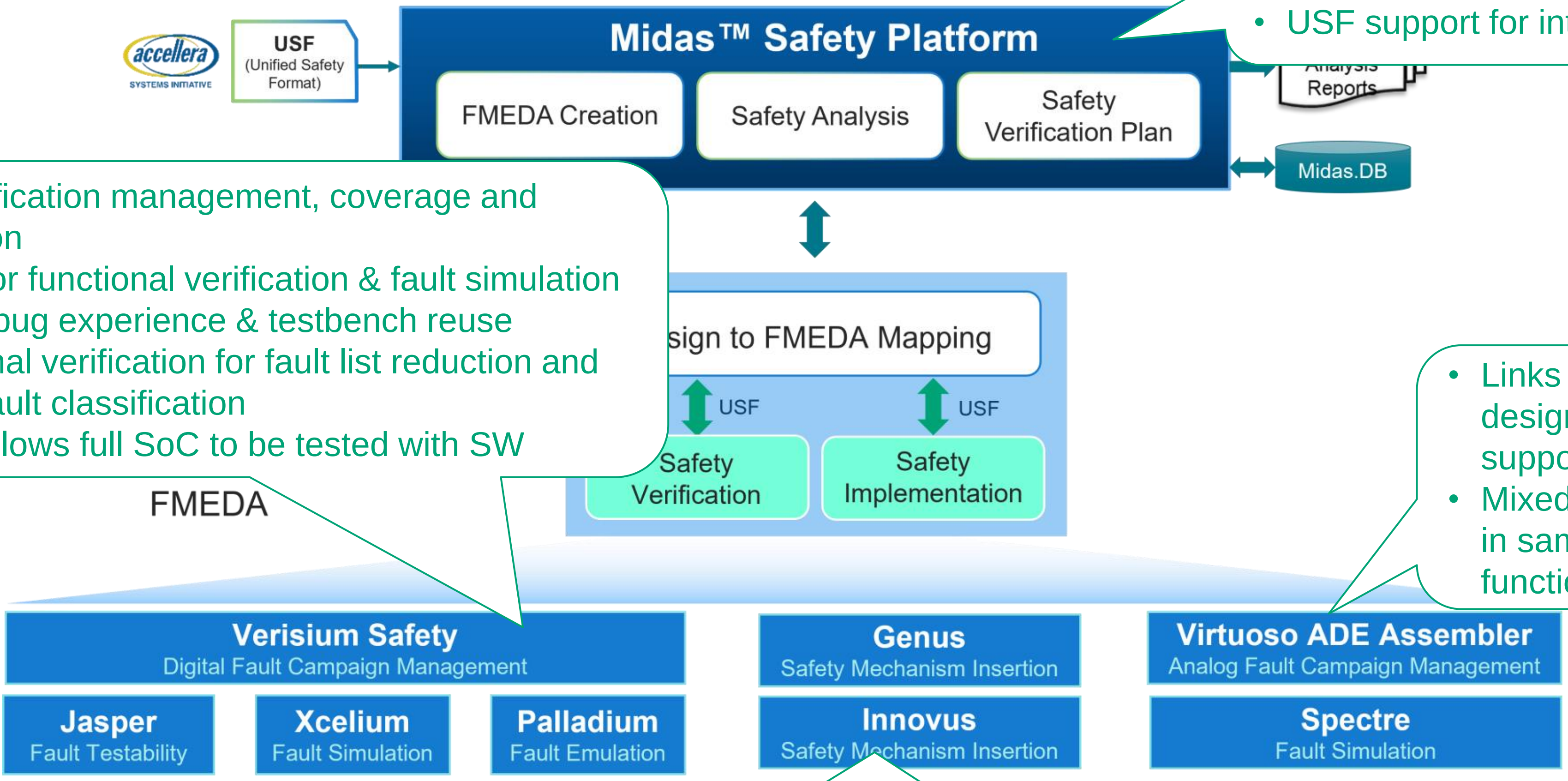
- ✓ Fault Campaign Automation
  - ✓ Same verification environment (Verisium Manager add-on)
  - ✓ Single front-end campaign configuration
  - ✓ Jasper and both Xcelium fault engines orchestration
    - ✓ Data exchange via the proprietary unified fault database
  - ✓ Dedicated fault coverage analysis (GUI and reports)
- ✓ Multi-Domain Fault Analysis support
  - ✓ Permanent and Transient fault campaigns
  - ✓ Diagnostic Coverage and Safeness
    - ✓ Software-based Self-Test Library (STL) assessment
    - ✓ Safety Mechanism (integration) Verification (+Detection Time Interval)
  - ✓ Fault / Test Grading (DFT) + Architectural Vulnerability (RadHard)
- ✓ ISO26262 tool qualification – up to ASIL D


USF
(Unified Safety Format)
Midas™ Safety Platform for Functional Safety Analysis
Design Mapping
Safety Analysis
Safety Verification Plan
FMEDA-driven digital and analog architecture exploration
Safety Digital Design and Verification
Verification
Verisium Manager™ Safety, Xcelium™ Safety, Jasper™ FSV
Implementation
Genus™, Innovus™, Conformal®
Highest performance fault simulation
Safety-aware implementation
Safety Analog Design and Verification
Verification
Genus™ Synthesis, Innovus™ Implementation, Conformal® EC
Implementation
Virtuoso® Schematic Editor
Virtuoso® Layout Suite
Automated analog fault simulation
Diagnostic coverage back-annotation
Verisium Manager Safety
Fault Campaign Manager - FCM
Palladium
Fault Emulator
Jasper
FSV App
Xcelium
Fault
Simulator
Fault DB

UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

Summary

Advantages of the Cadence Functional Safety Solution



accellera SYSTEMS INITIATIVE

USF (Unified Safety Format)

Midas™ Safety Platform

FMEDA Creation

Safety Analysis

Safety Verification Plan

Analysis Reports

Midas.DB

sign to FMEDA Mapping

USF

USF

Safety Verification

Safety Implementation

FMEDA

Verisium Safety Digital Fault Campaign Management

Jasper Fault Testability

Xcelium Fault Simulation

Palladium Fault Emulation

Genus Safety Mechanism Insertion

Innovus Safety Mechanism Insertion

Virtuoso ADE Assembler Analog Fault Campaign Management

Spectre Fault Simulation

- Comprehensive architectural FMEDA
- Links to best-in-class digital, analog and library tools for detailed FMEDA
- USF support for interoperability

- Best-in-class verification management, coverage and traceability solution
- Same simulator for functional verification & fault simulation gives superior debug experience & testbench reuse
- Best-in-class formal verification for fault list reduction and formally-proven fault classification
- Fault emulation allows full SoC to be tested with SW

- Links to analog/mixed signal design tools with USF support
- Mixed signal fault simulation in same environment as functional verification

- Links to digital synthesis and implementation tools with USF support
- Best-in-class RTL/gate/netlist equivalence checking

cadence®