2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 15-16, 2024

Exploring the Next-Generation of Debugging with Verification Management System and Integrated Development Environment

Werner Kerscher, Ionut Cirjan, Noam Roth

SYNOPSYS®

accellera SYSTEMS INITIATIVE

Agenda

- Introduction to Next-Generation Verdi
- General Verdi Improvements
- Synopsys Verdi® Verification Management System with VC Execution Manger
- Verdi Integrated Design Environment (IDE) with Euclide
- Summary + Q&A

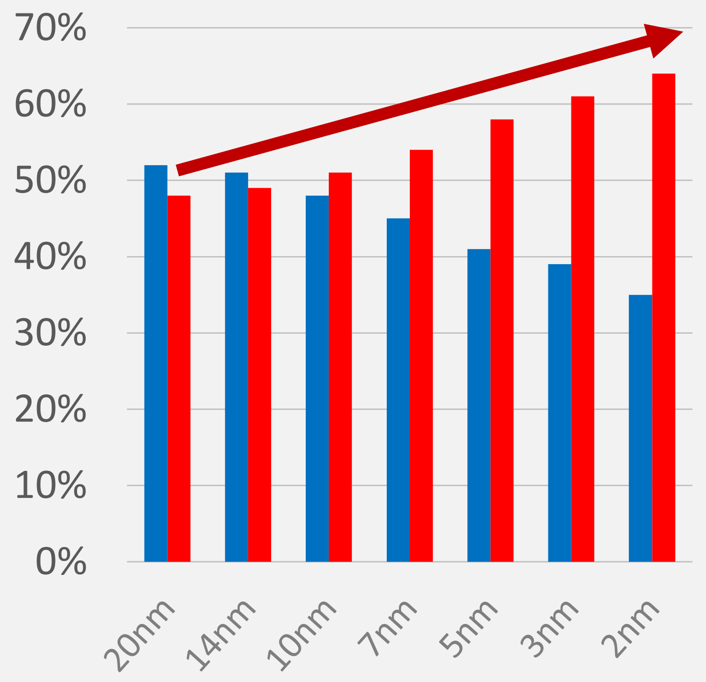
2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

Introduction to Next-Generation Verdi

accellera SYSTEMS INITIATIVE

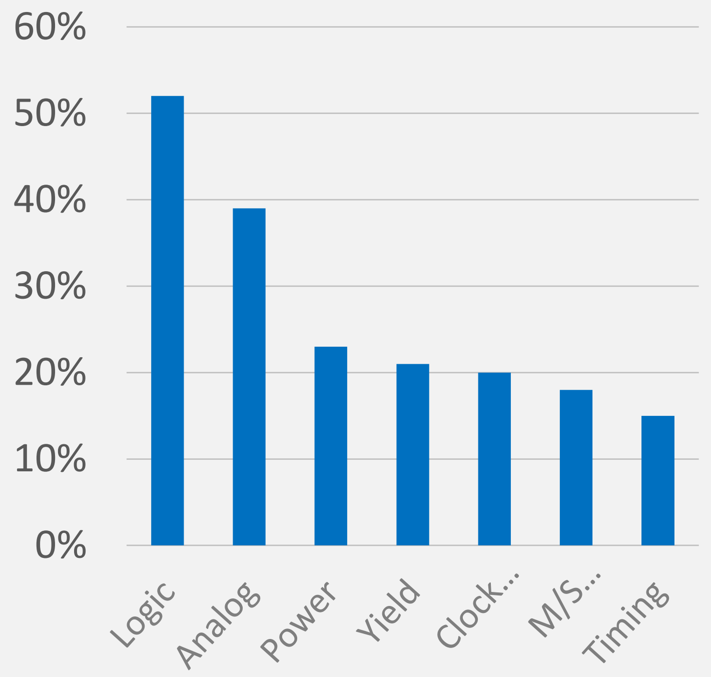
Impact on Right First-Time Silicon

Fewer First-Time Right



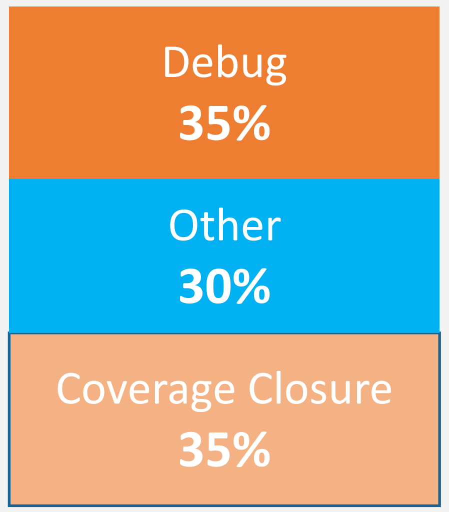
More Respins

Logic Bugs Dominating



Main Cause of Respins

Verification Time Spent

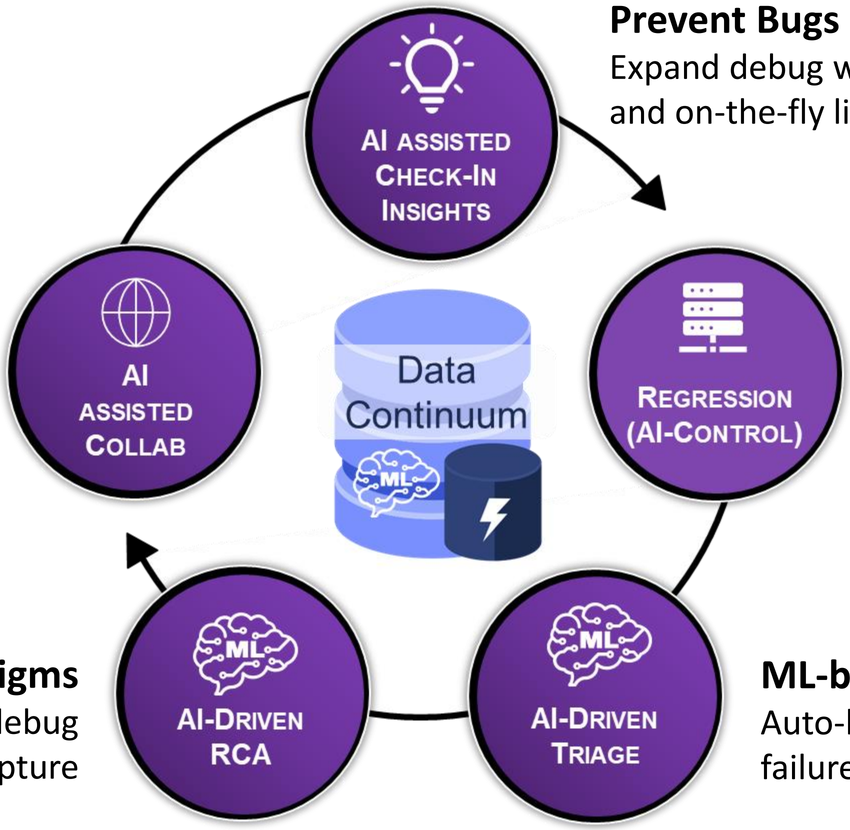


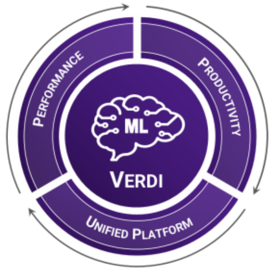
Need to Reduce Debug Time!

Source: Wilson Report 2022

AI-Assisted Debug Flow

Next-Generation Debug: Improves debug productivity up to 10X





Prevent Bugs
Expand debug with RTL & TB IDE, and on-the-fly linting

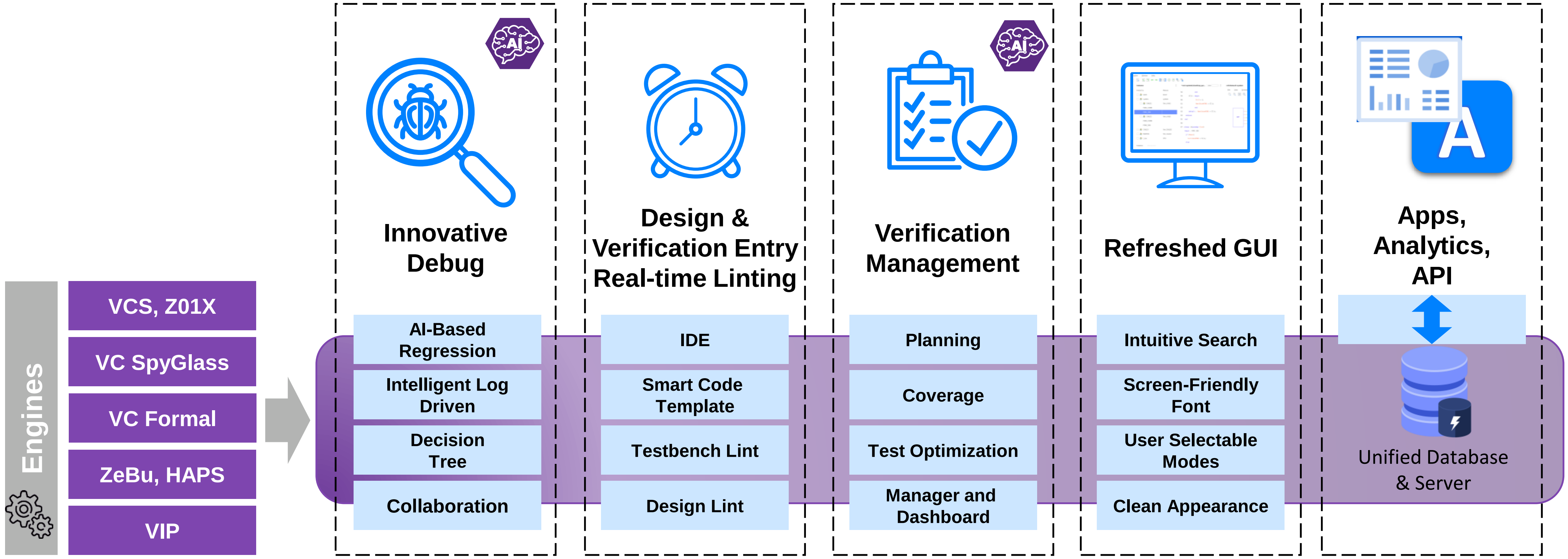
Regression Management
ML-optimized daily regression, reduce HW costs and increase efficiency

ML-based Triage
Auto-binning and triage, predict source of failures (RTL or engineer)

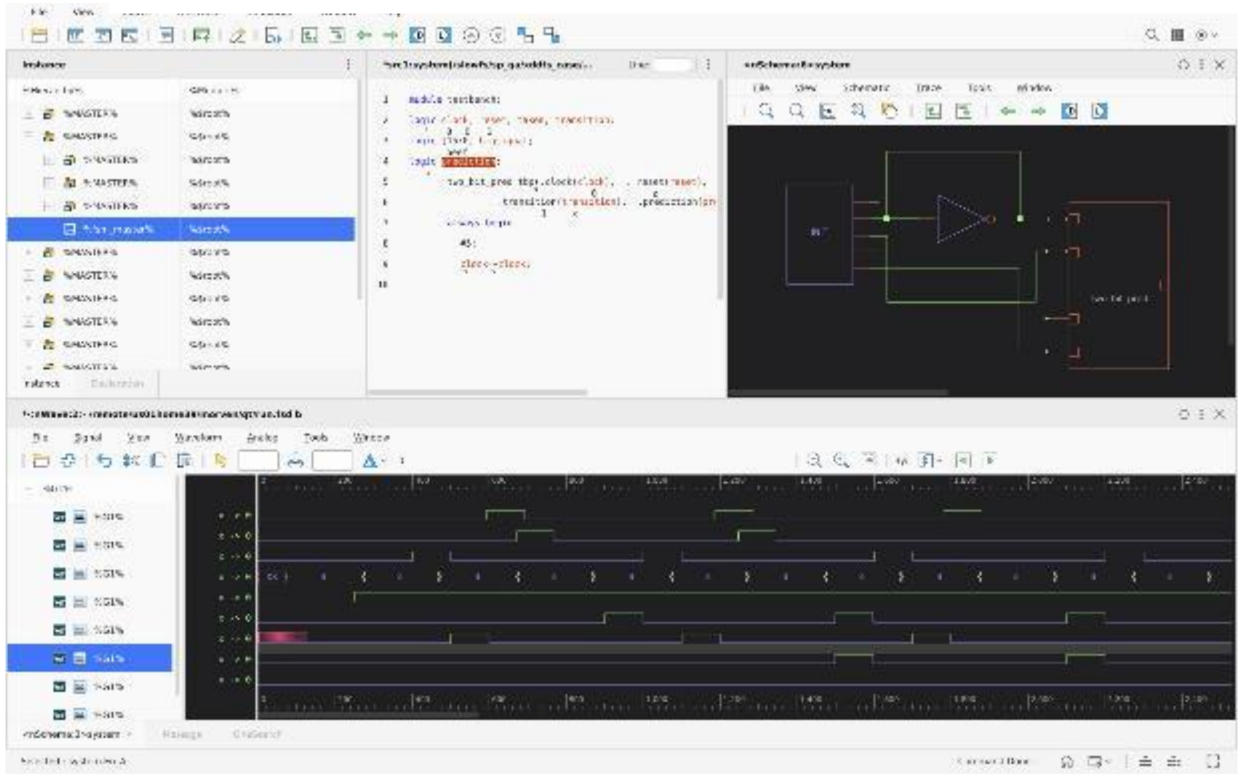
New Debug Paradigms
Auto RCA, Smart Log visualization, debug knowledge capture

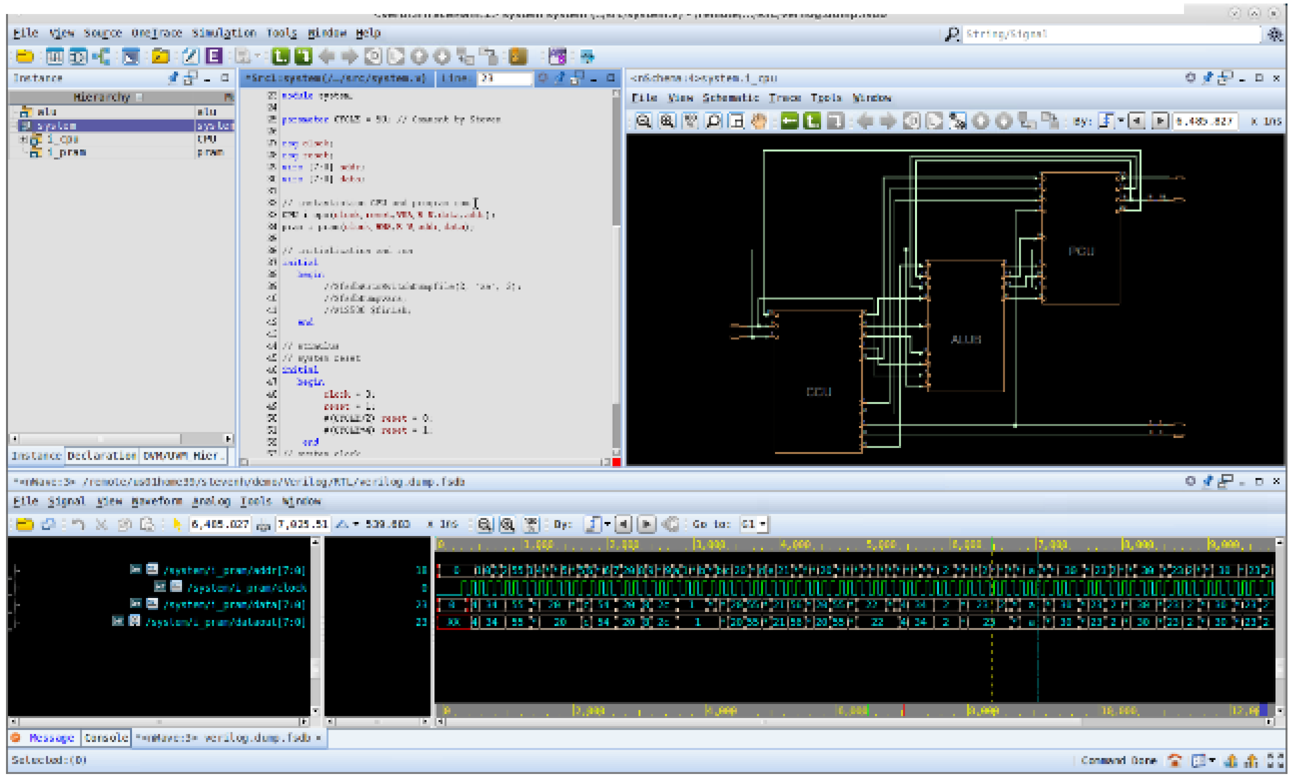
Global Collaboration
Multi-site, multi-user collaborative debug, knowledge reuse

Introducing Next-Generation Verdi Platform



Refreshed GUI

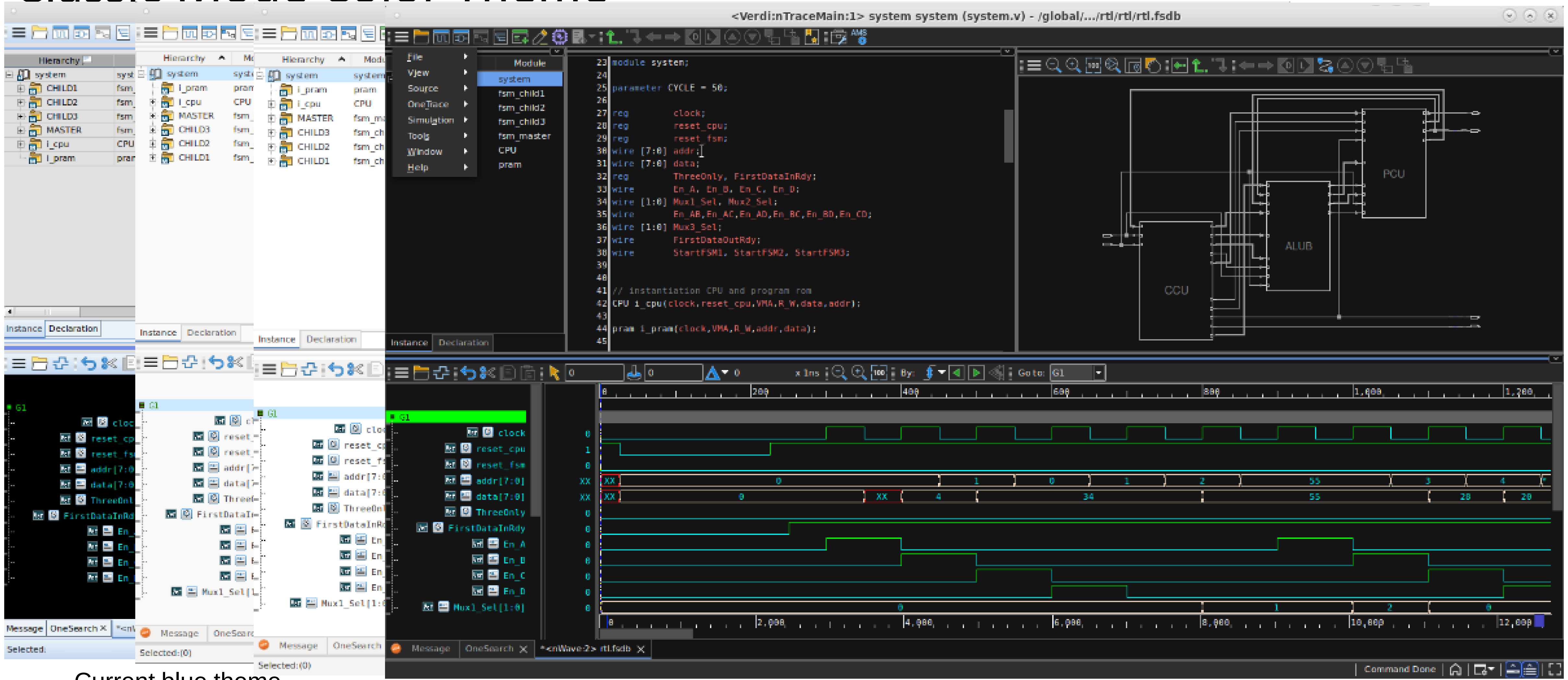




New GUI

Current GUI

Classic Mode Color Theme



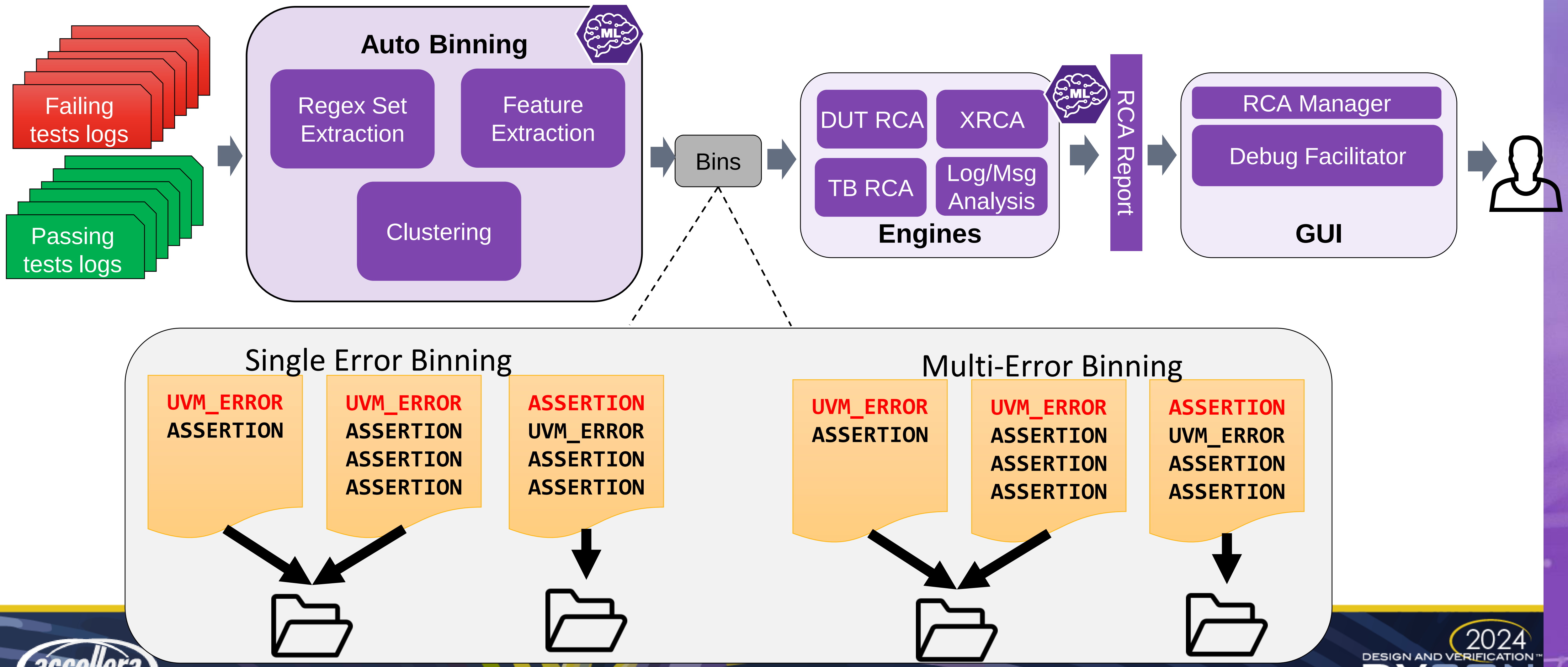
<Verdi:nTraceMain:1> system system (system.v) - /global/.../rtl/rtl/rtl.fsdb

```
module system;

parameter CYCLE = 50;

reg         clock;
reg         reset_cpu;
reg         reset_fsm;
wire [7:0]  addr;
wire [7:0]  data;
reg         ThreeOnly, FirstDataInRdy;
wire        En_A, En_B, En_C, En_D;
wire [1:0]  Mux1_Sel, Mux2_Sel;
wire        En_AB,En_AC,En_AD,En_BC,En_BD,En_CD;
wire [1:0]  Mux3_Sel;
wire        FirstDataOutRdy;
wire        StartFSM1, StartFSM2, StartFSM3;


// instantiation CPU and program rom
CPU i_cpu(clock,reset_cpu,VMA,R_W,data,addr);

pram i_pram(clock,VMA,R_W,addr,data);
```

Current blue theme

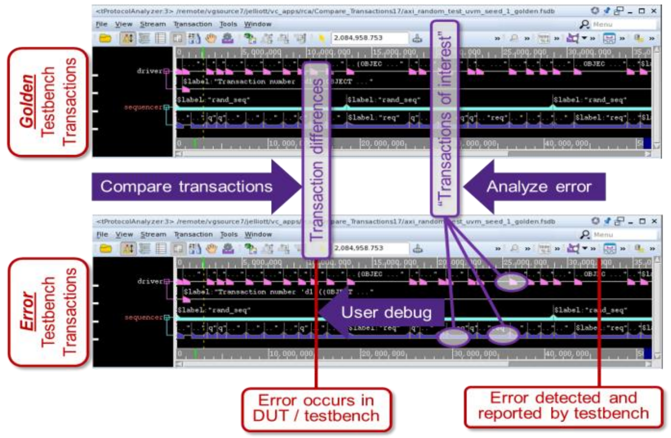
Classic mode (default) Natural mode Bright mode Dark Mode

Regression Binning with ML



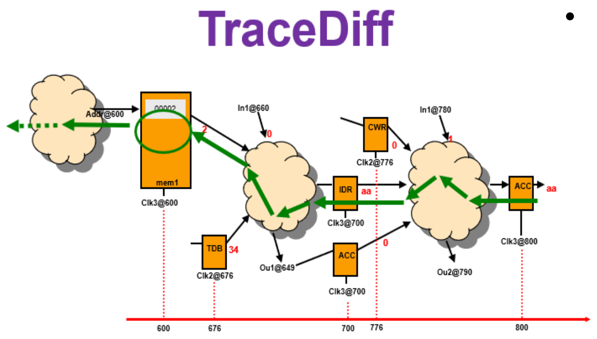
Verdi Root Cause Analysis (RCA) Engines

TBRCA


Golden Testbench Transactions
Error Testbench Transactions
Compare transactions
Transaction differences
"Transactions of interest"
Analyze error
User debug
Error occurs in DUT / testbench
Error detected and reported by testbench

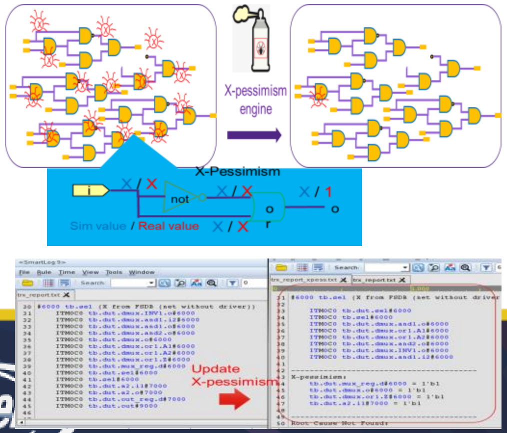
- Transaction Diff – Diff the transaction in the reference vs failing FSDB

DUTRCA



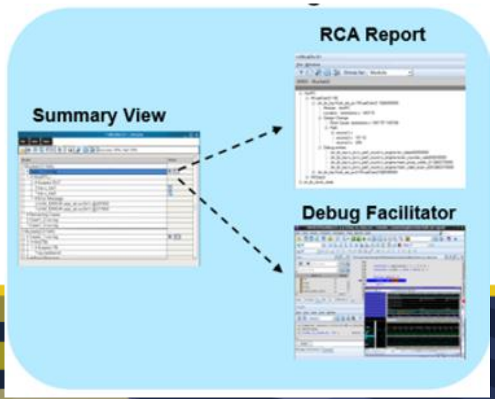
- Adopt roll back mechanism and TraceDiff technology to narrow down DUT problem

XRCA / with X-Pessimism



- Scan X signals in FSDB and trace the root cause of X.

Debug Facilitator



- Generates debug data nightly for each bin.

Verdi Next-Gen: Accelerate Debug Automation

Customer Examples

Automotive

Gate-level with Many Xs

Xx

XRCA Engine

Video

DUT Code Change

Xx

DUTRCA Engine

Server

Multiple Failing Assertions

Xx

DUTRCA Engine

Graphics

DFT with X Monitors

Xx

XRCA Engine

Large SoC

Massive Log Files

Xx

ML-Binning

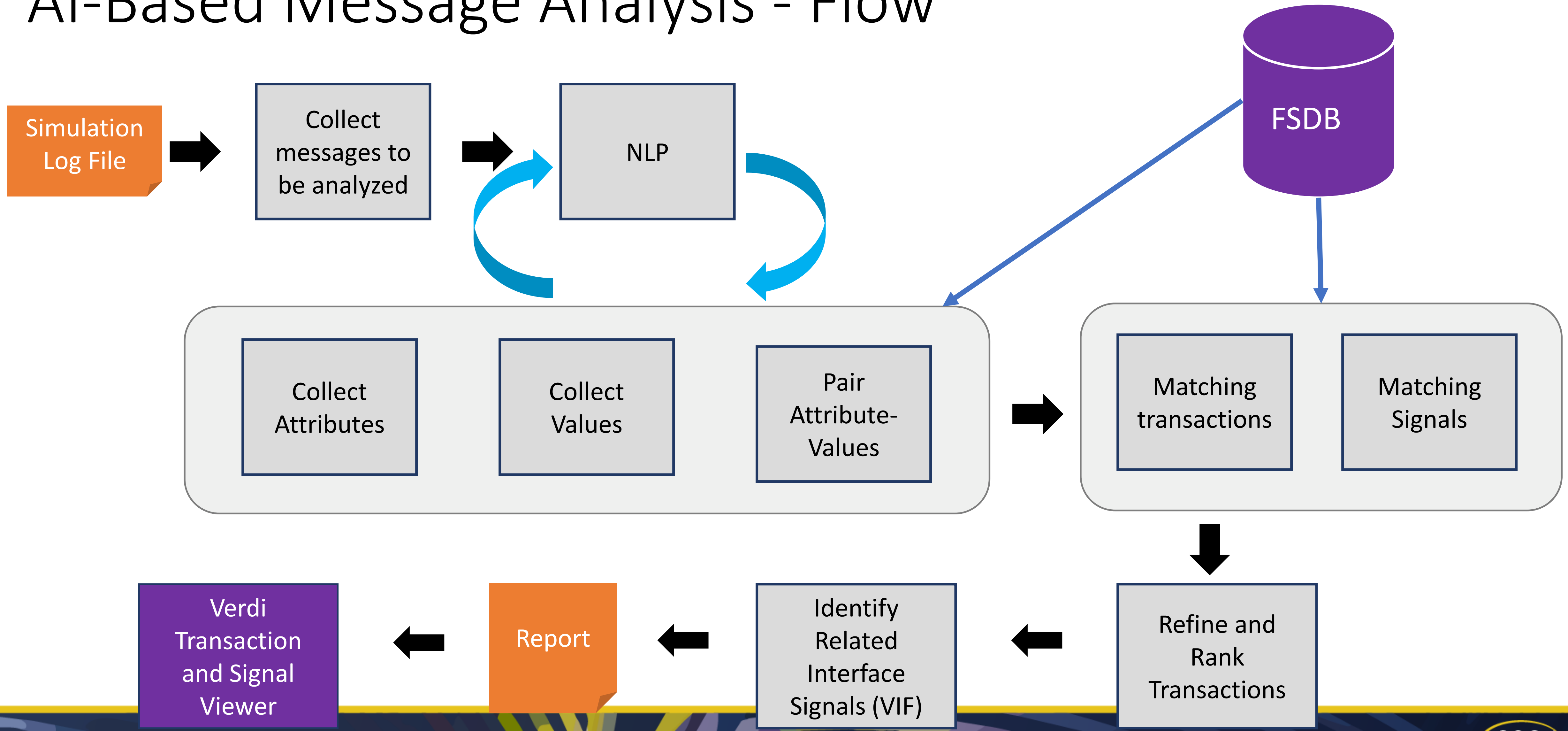
Video

DUT Hang

Xx

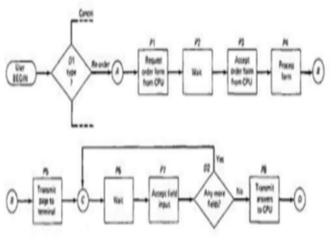
DUTRCA Engine

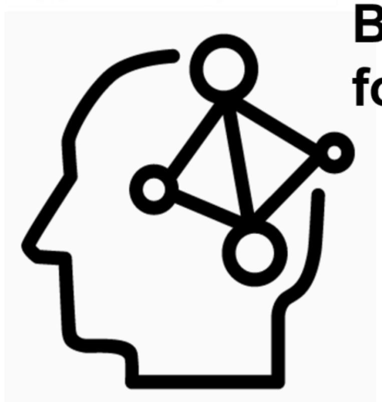
AI-Based Message Analysis - Flow


Simulation Log File
Collect messages to be analyzed
NLP
FSDB
Collect Attributes
Collect Values
Pair Attribute-Values
Matching transactions
Matching Signals
Refine and Rank Transactions
Identify Related Interface Signals (VIF)
Report
Verdi Transaction and Signal Viewer

Debug Decision Tree (DDT)

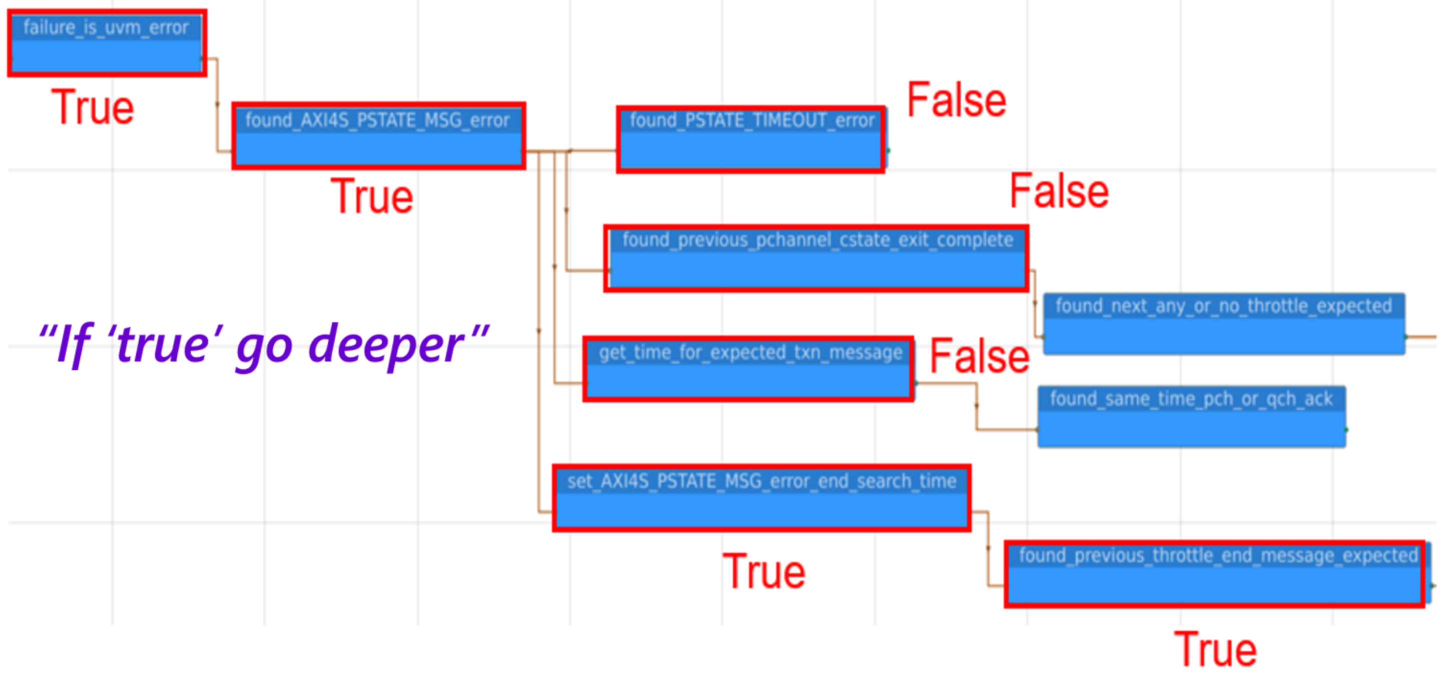
A tool for capturing, sharing, and executing debug knowledge across platforms





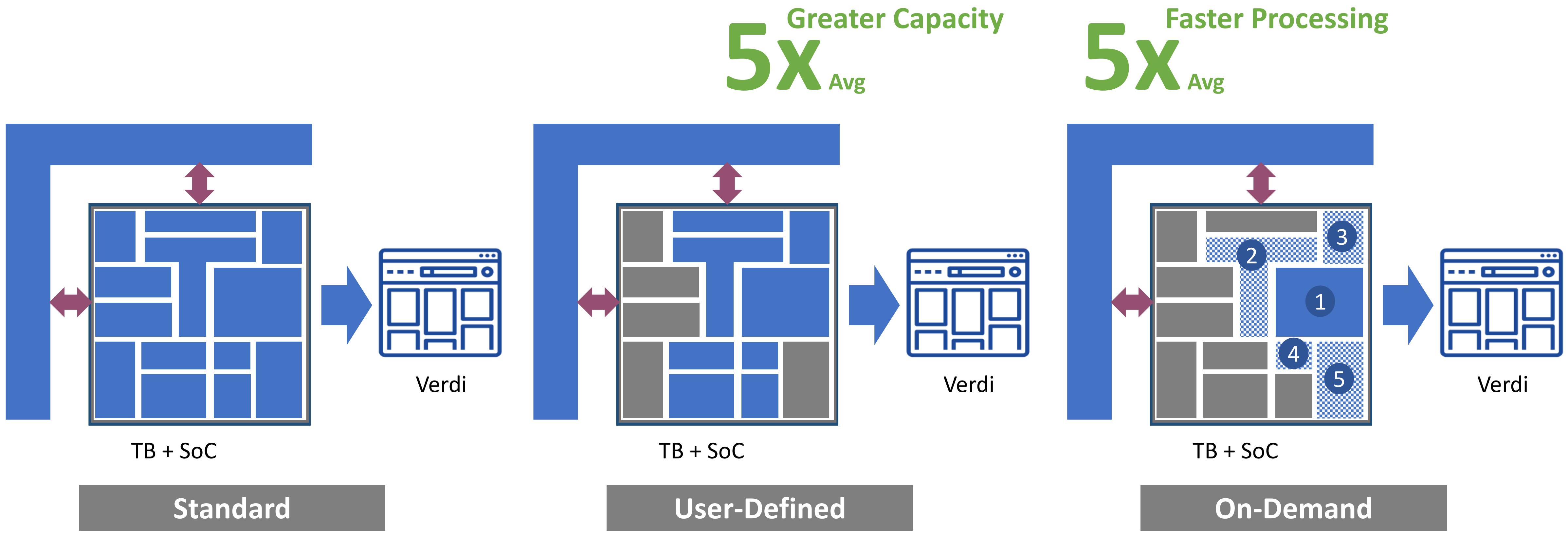
Bug found !

Collected algorithms in heads
Found the data required to feed those algorithms

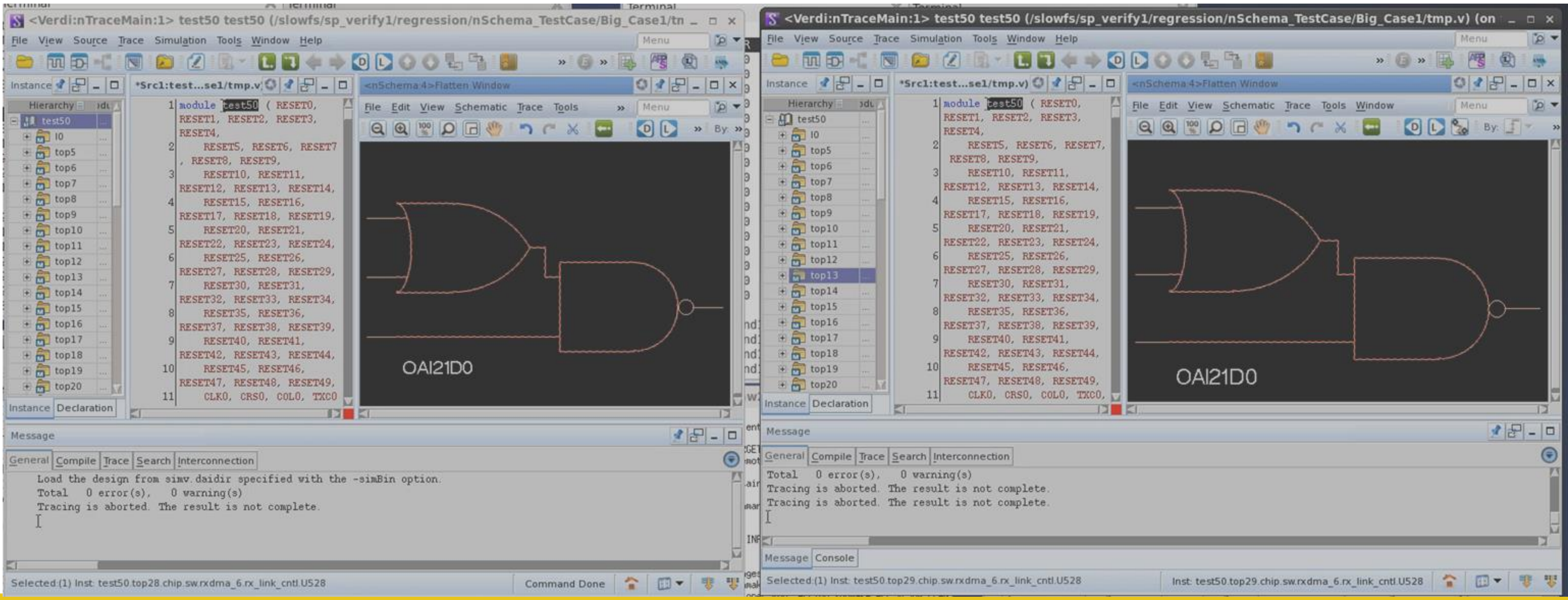

failure_is_uvm_error
True
found_AXI4S_PSTATE_MSG_error
True
found_PSTATE_TIMEOUT_error
False
found_previous_pchannel_cstate_exit_complete
False
found_next_any_or_no_throttle_expected
“If ‘true’ go deeper”
get_time_for_expected_txn_message
False
found_same_time_pch_or_qch_ack
set_AXI4S_PSTATE_MSG_error_end_search_time
True
found_previous_throttle_end_message_expected
True

Improved Debug Capacity and Performance

Large Designs, Fast Processing



Faster Debug with Background Multitasking



<Verdi:nTraceMain:1> test50 test50 (/slowfs/sp_verify1/regression/nSchema_TestCase/Big_Case1/tmp.v)
File View Source Trace Simulation Tools Window Help
Instance
*Src1:test...sel/tmp.v
<nSchema:4>Flatten Window
File Edit View Schematic Trace Tools Window
Hierarchy
test50
I0
top5
top6
top7
top8
top9
top10
top11
top12
top13
top14
top15
top16
top17
top18
top19
top20
Instance Declaration
OAI21D0
Message
General Compile Trace Search Interconnection
Message Console

```
module test50 ( RESET0,
RESET1, RESET2, RESET3,
RESET4,
    RESET5, RESET6, RESET7
, RESET8, RESET9,
    RESET10, RESET11,
RESET12, RESET13, RESET14,
    RESET15, RESET16,
RESET17, RESET18, RESET19,
    RESET20, RESET21,
RESET22, RESET23, RESET24,
    RESET25, RESET26,
RESET27, RESET28, RESET29,
    RESET30, RESET31,
RESET32, RESET33, RESET34,
    RESET35, RESET36,
RESET37, RESET38, RESET39,
    RESET40, RESET41,
RESET42, RESET43, RESET44,
    RESET45, RESET46,
RESET47, RESET48, RESET49,
    CLK0, CRS0, COL0, TXC0,
```

```
Load the design from simv.daidir specified with the -simBin option.
Total    0 error(s),    0 warning(s)
Tracing is aborted. The result is not complete.
```

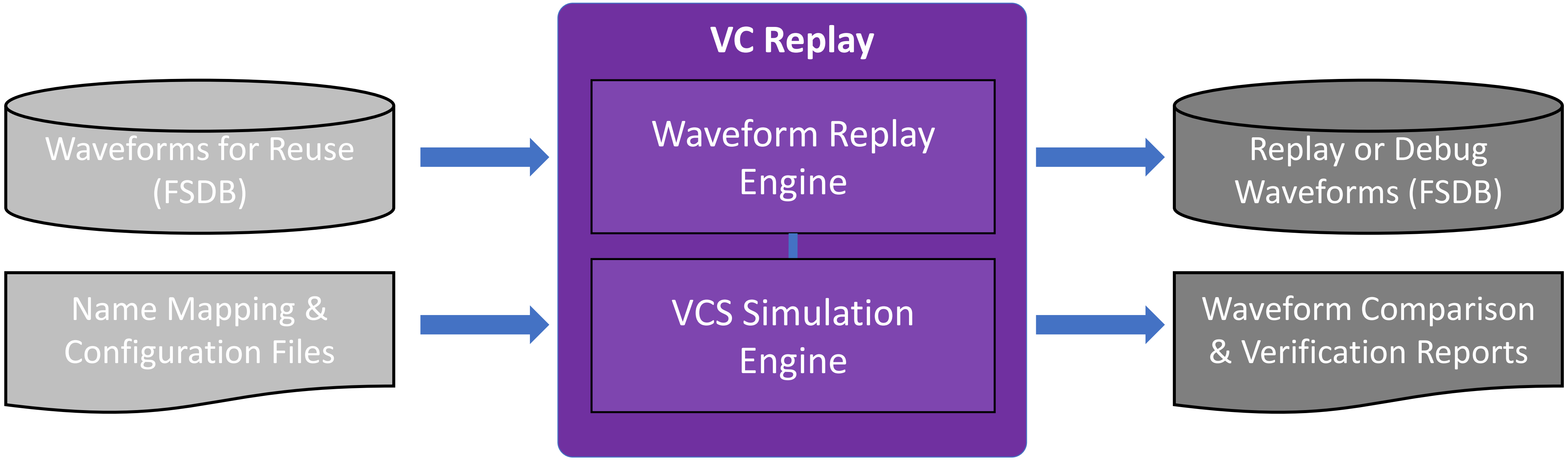
```
Selected:(1) Inst: test50.top28.chip.sw.rxdma_6.rx_link_cntl.U528    Command Done
```

```
Total    0 error(s),    0 warning(s)
Tracing is aborted. The result is not complete.
Tracing is aborted. The result is not complete.
```

```
Selected:(1) Inst: test50.top29.chip.sw.rxdma_6.rx_link_cntl.U528    Inst: test50.top29.chip.sw.rxdma_6.rx_link_cntl.U528
```

VC Replay Technology Overview

For Speed Up of Functional Verification


Waveforms for Reuse (FSDB)
Name Mapping & Configuration Files
VC Replay
Waveform Replay Engine
VCS Simulation Engine
Replay or Debug Waveforms (FSDB)
Waveform Comparison & Verification Reports

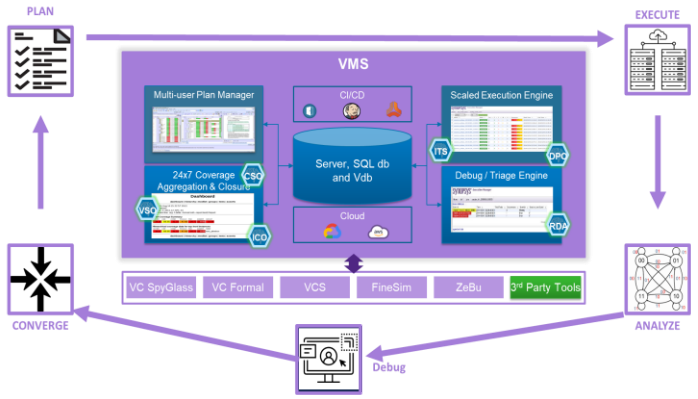
Opportunity for 10X reduction in debug TAT

[This slide intentionally blank]

Verdi Verification Management System

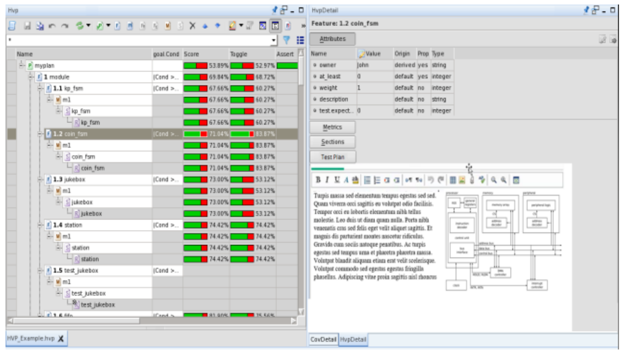
Verification Management System

Manager and Dashboard



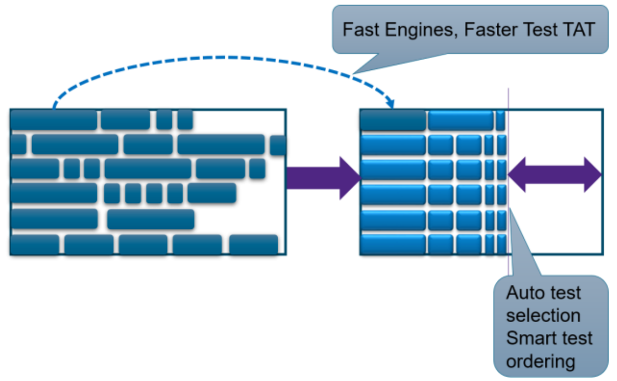
- Test planning, execution & debug, coverage merge and annotation
- Enables verification data-over-time to be mined for analytics

Planner



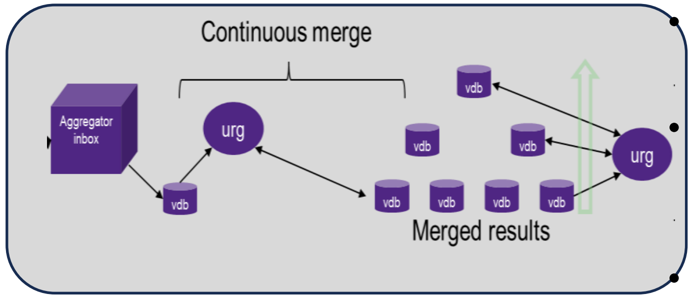
- Multi-user test scheduling/planning
- Supports change history and restore
- API for automated report generation and updates

Runner



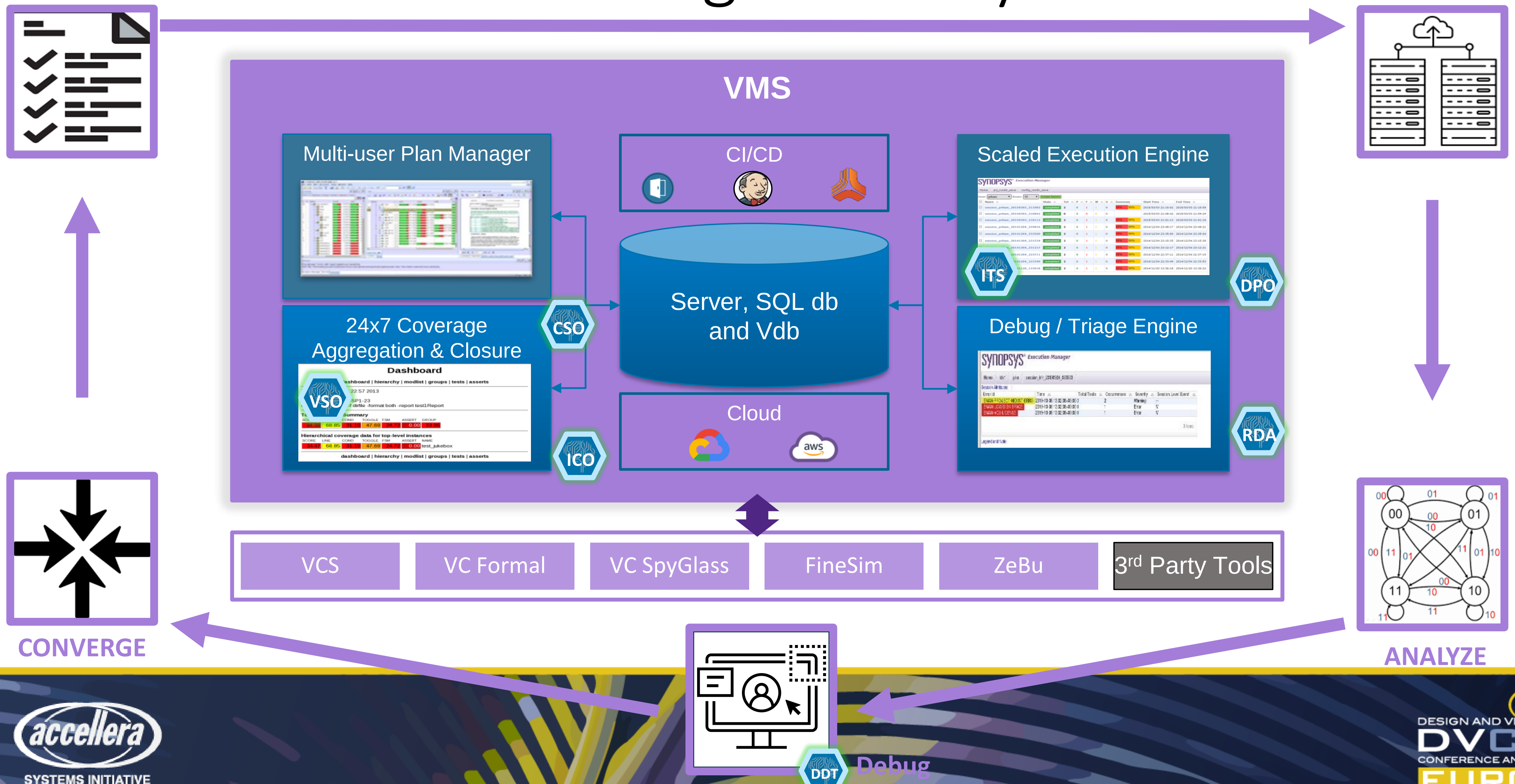
- Runs regressions
- Order tests to eliminate long tail
- Synopsys VCS® engine performance enhancement

Coverage



- Continuously merges incoming coverage
- Integrated tagged VDB from ad hoc regression runs
- Can generate moving window merge VDB

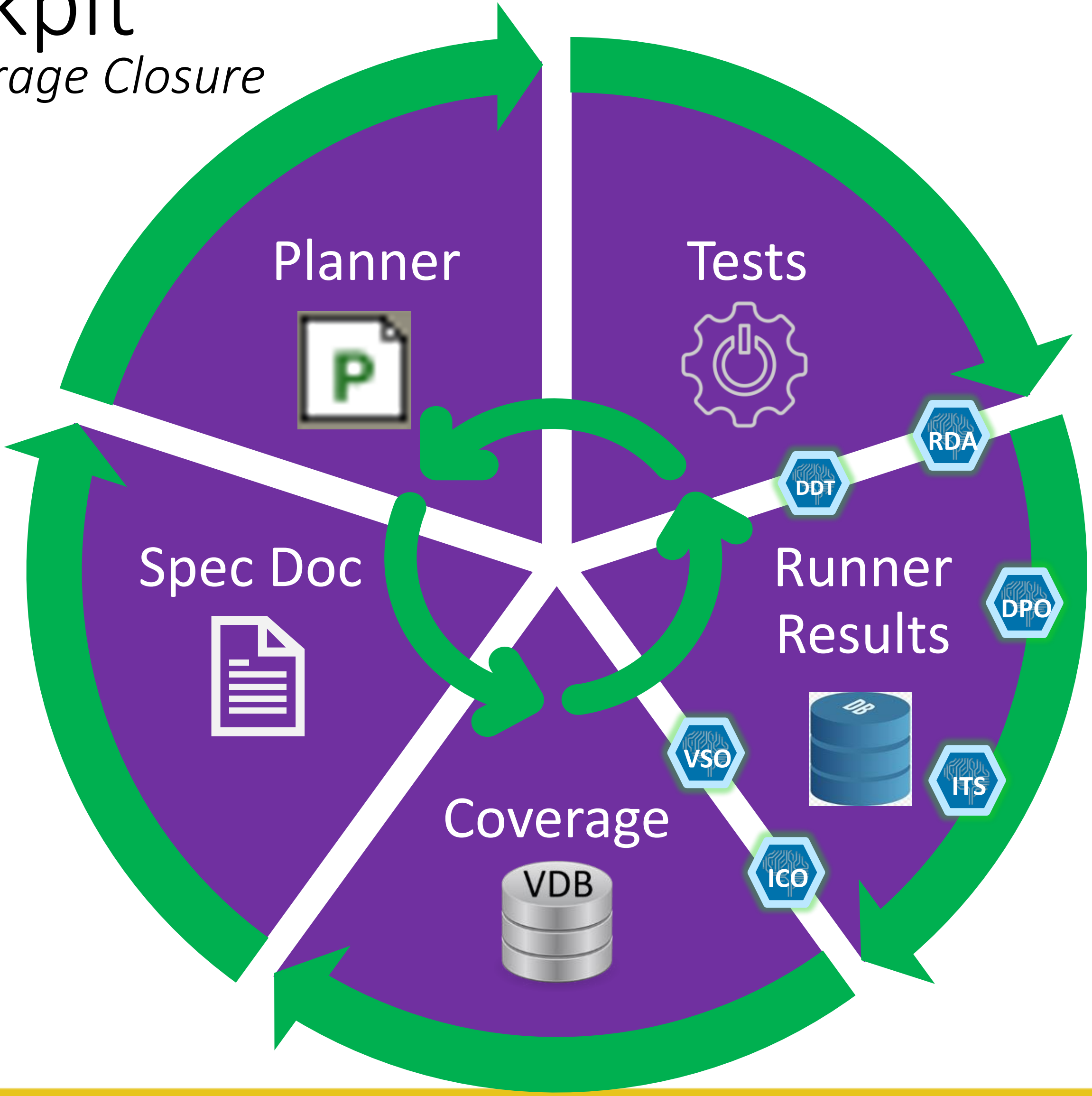
Verification Management System

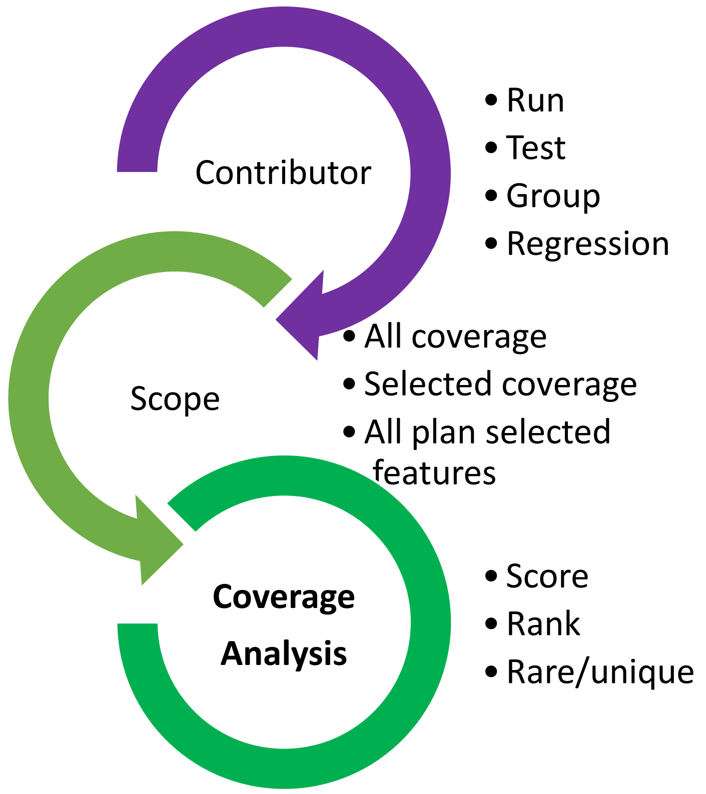


Unified Cockpit

Driving towards Coverage Closure

- All in one place GUI
- GUI layouts
- UI Customization
- Interconnections
- Coverage Results
- Coverage Analysis

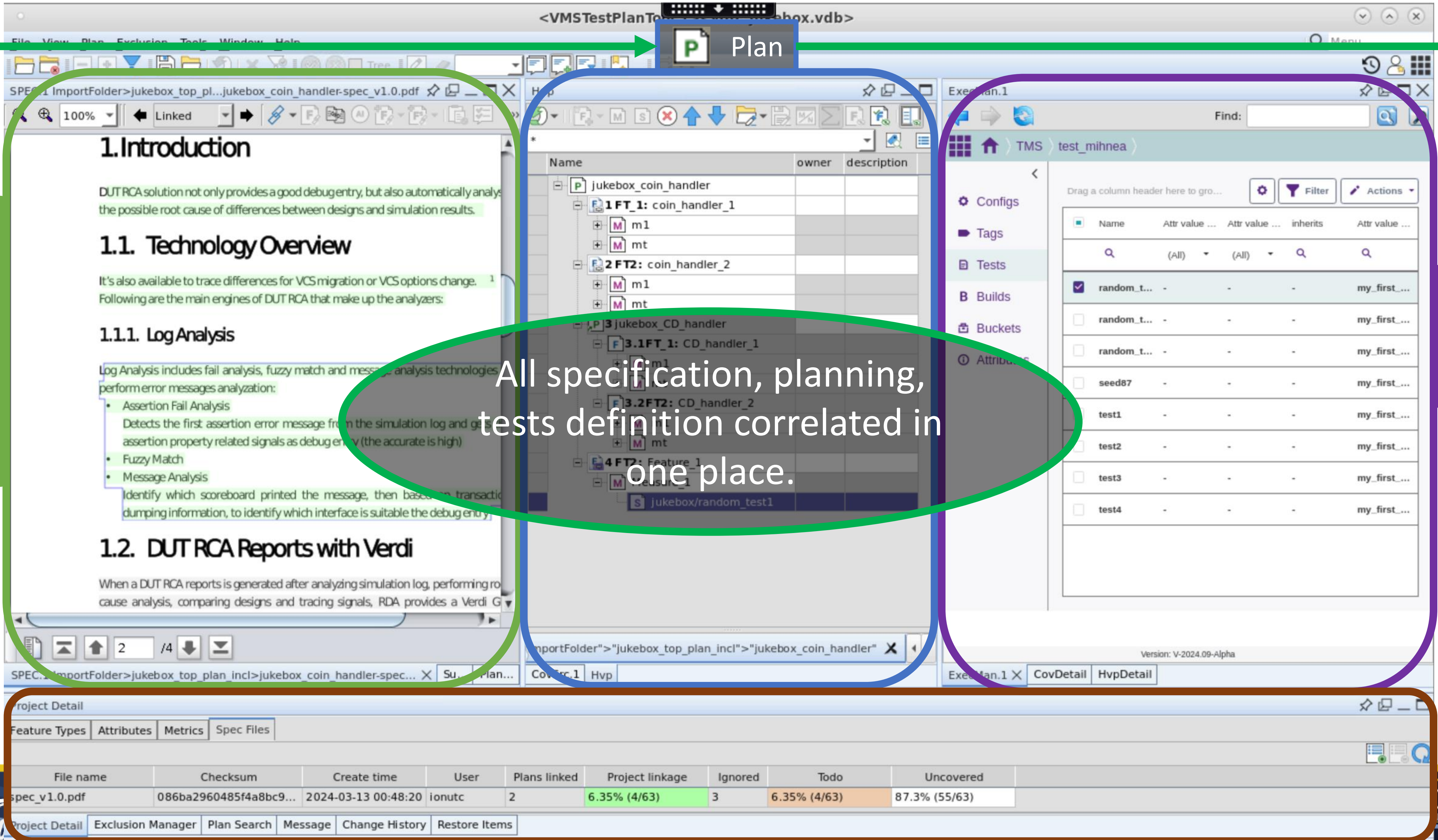




Plan

Spec Document

Tests



SPEC...ImportFolder>jukebox_top_pl...jukebox_coin_handler-spec_v1.0.pdf

1. Introduction

DUT RCA solution not only provides a good debug entry, but also automatically analy... the possible root cause of differences between designs and simulation results.

1.1. Technology Overview

It's also available to trace differences for VCS migration or VCS options change. [1] Following are the main engines of DUT RCA that make up the analyzers:

1.1.1. Log Analysis

Log Analysis includes fail analysis, fuzzy match and message analysis technologies ... perform error messages analyzation:

- Assertion Fail Analysis
  Detects the first assertion error message from the simulation log and ... assertion property related signals as debug entry (the accurate is high)
- Fuzzy Match
- Message Analysis
  Identify which scoreboard printed the message, then based on transactio... dumping information, to identify which interface is suitable the debug entry

1.2. DUT RCA Reports with Verdi

When a DUT RCA reports is generated after analyzing simulation log, performing ro... cause analysis, comparing designs and tracing signals, RDA provides a Verdi G...

```
jukebox_coin_handler
  1 FT_1: coin_handler_1
    m1
    mt
  2 FT2: coin_handler_2
    m1
    mt
  3 jukebox_CD_handler
    3.1FT_1: CD_handler_1
      m1
    3.2FT2: CD_handler_2
      mt
  4 FT2: Feature_1
    Measure_1
      jukebox/random_test1
```

...portFolder">"jukebox_top_plan_incl">"jukebox_coin_handler"

Name	Attr value ...	Attr value ...	inherits	Attr value ...
✓ random_t...	-	-	-	my_first_...
random_t...	-	-	-	my_first_...
random_t...	-	-	-	my_first_...
seed87	-	-	-	my_first_...
test1	-	-	-	my_first_...
test2	-	-	-	my_first_...
test3	-	-	-	my_first_...
test4	-	-	-	my_first_...

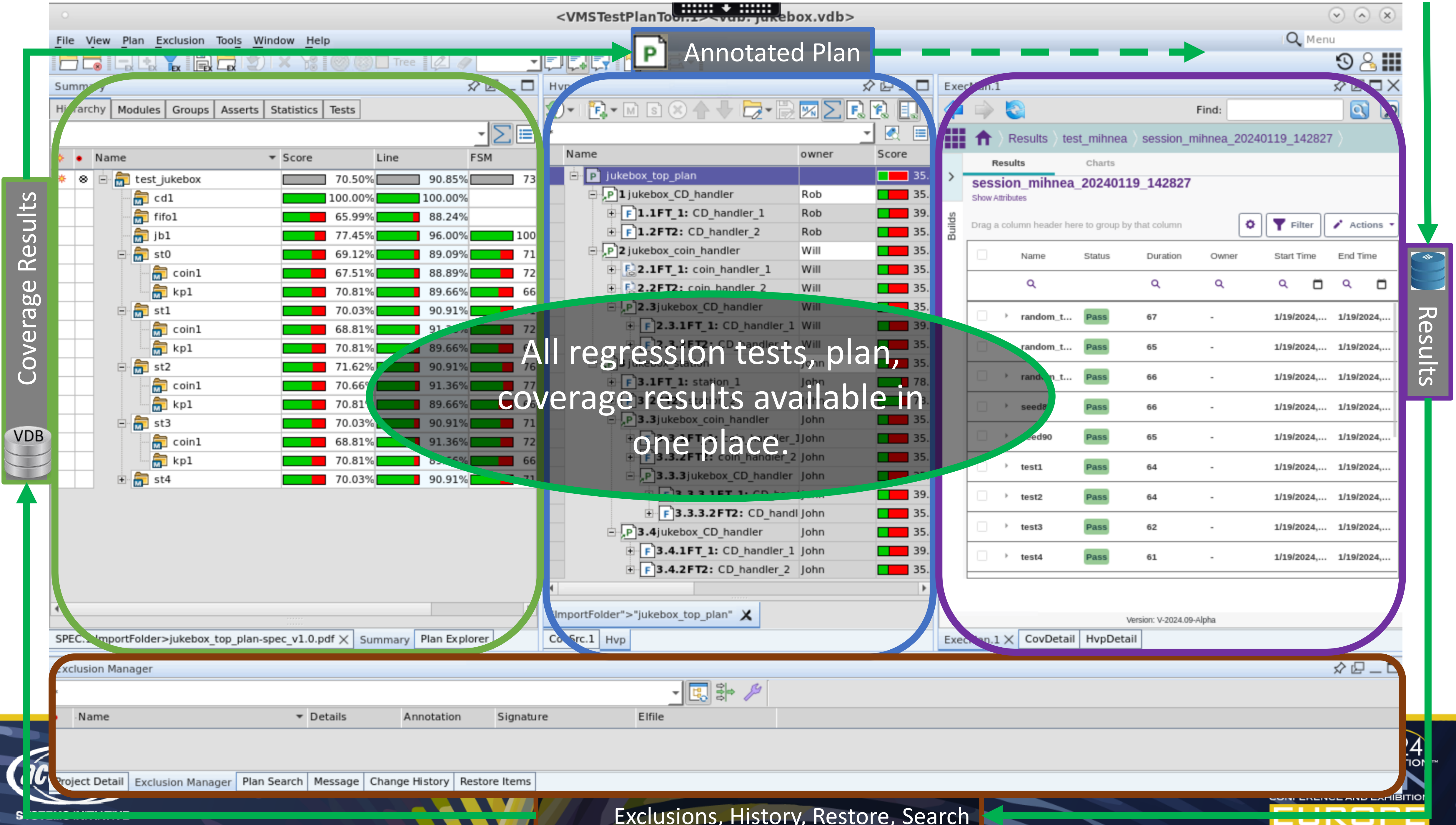
File name	Checksum	Create time	User	Plans linked	Project linkage	Ignored	Todo	Uncovered
...spec_v1.0.pdf	086ba2960485f4a8bc9...	2024-03-13 00:48:20	ionutc	2	6.35% (4/63)	3	6.35% (4/63)	87.3% (55/63)

All specification, planning, tests definition correlated in one place.

Project Details, Spec Summaries, Admin

SYSTEMS INITIATIVE

2024 DESIGN AND VERIFICATION™ CONFERENCE AND EXHIBITION EUROPE



All regression tests, plan, coverage results available in one place.

<VMSTestPlanTool.1><vdb: jukebox.vdb>

File View Plan Exclusion Tools Window Help

Summary — Hierarchy | Modules | Groups | Asserts | Statistics | Tests

Name	Score	Line	FSM
test_jukebox	70.50%	90.85%	73
cd1	100.00%	100.00%	
fifo1	65.99%	88.24%	
jb1	77.45%	96.00%	100
st0	69.12%	89.09%	71
coin1	67.51%	88.89%	72
kp1	70.81%	89.66%	66
st1	70.03%	90.91%	
coin1	68.81%		72
kp1	70.81%	89.66%	
st2	71.62%	90.91%	76
coin1		91.36%	77
kp1		89.66%	
st3	70.03%	90.91%	71
coin1	68.81%	91.36%	72
kp1	70.81%		66
st4	70.03%	90.91%	

SPEC...ImportFolder>jukebox_top_plan-spec_v1.0.pdf | Summary | Plan Explorer

Hvp

Name	owner	Score
jukebox_top_plan		35.
1 jukebox_CD_handler	Rob	35.
1.1FT_1: CD_handler_1	Rob	39.
1.2FT2: CD_handler_2	Rob	35.
2 jukebox_coin_handler	Will	35.
2.1FT_1: coin_handler_1	Will	35.
2.2FT2: coin_handler_2	Will	35.
2.3jukebox_CD_handler	Will	35.
2.3.1FT_1: CD_handler_1	Will	39.
	Will	35.
	John	35.
3.1FT_1: station_1	John	78.
3.3jukebox_coin_handler	John	35.
	John	35.
3.3.2FT2: coin_handler_2	John	35.
3.3.3jukebox_CD_handler	John	
		39.
3.3.3.2FT2: CD_handl	John	35.
3.4jukebox_CD_handler	John	35.
3.4.1FT_1: CD_handler_1	John	39.
3.4.2FT2: CD_handler_2	John	35.

ImportFolder">"jukebox_top_plan" | CovSrc.1 | Hvp

Execution.1 — Find:

Results › test_mihnea › session_mihnea_20240119_142827

Results | Charts

session_mihnea_20240119_142827
Show Attributes

Drag a column header here to group by that column — Filter — Actions

Name	Status	Duration	Owner	Start Time	End Time
random_t...	Pass	67	-	1/19/2024,...	1/19/2024,...
random_t...	Pass	65	-	1/19/2024,...	1/19/2024,...
random_t...	Pass	66	-	1/19/2024,...	1/19/2024,...
seed8	Pass	66	-	1/19/2024,...	1/19/2024,...
seed90	Pass	65	-	1/19/2024,...	1/19/2024,...
test1	Pass	64	-	1/19/2024,...	1/19/2024,...
test2	Pass	64	-	1/19/2024,...	1/19/2024,...
test3	Pass	62	-	1/19/2024,...	1/19/2024,...
test4	Pass	61	-	1/19/2024,...	1/19/2024,...

Version: V-2024.09-Alpha

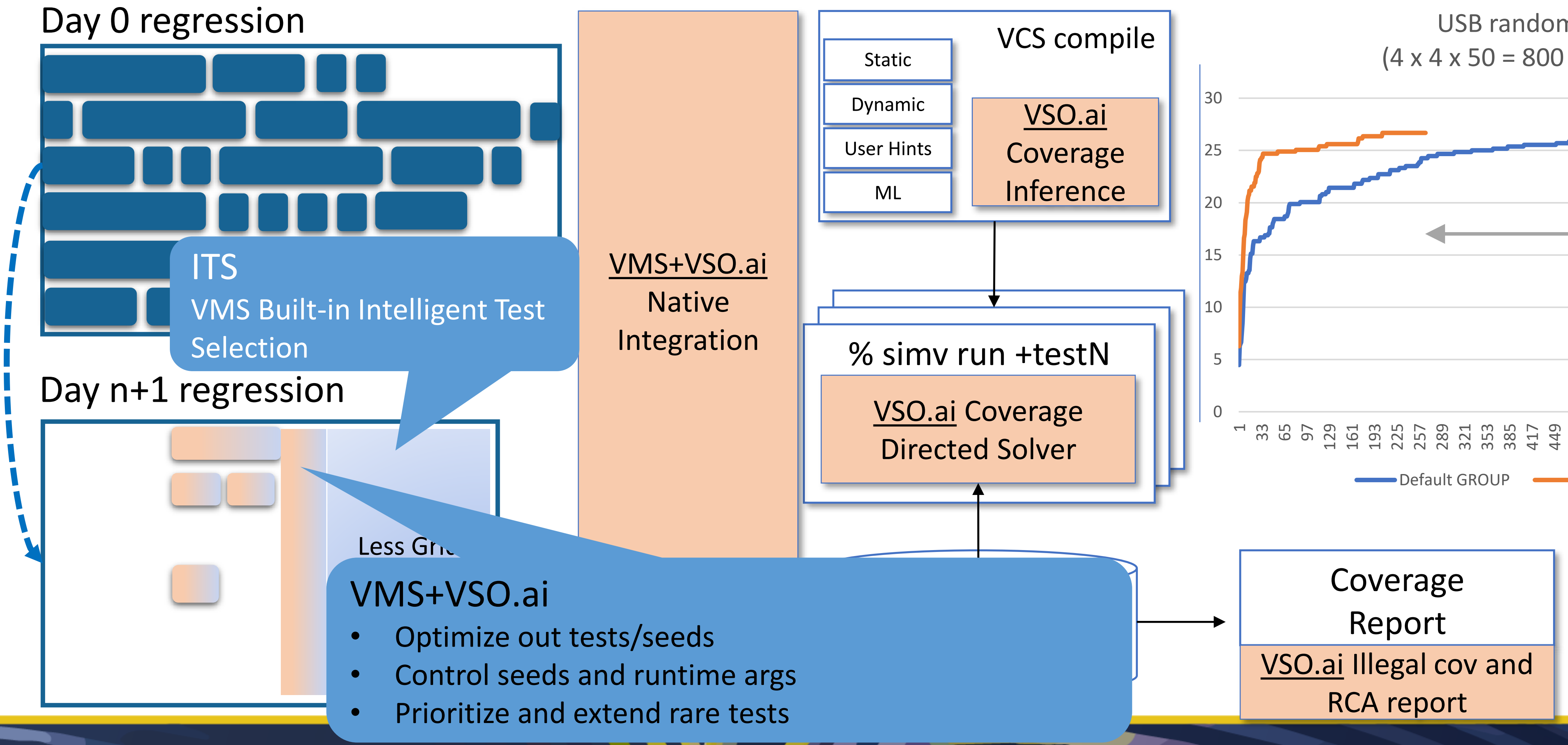
Execution.1 | CovDetail | HvpDetail

Exclusion Manager

Name	Details	Annotation	Signature	Elfile

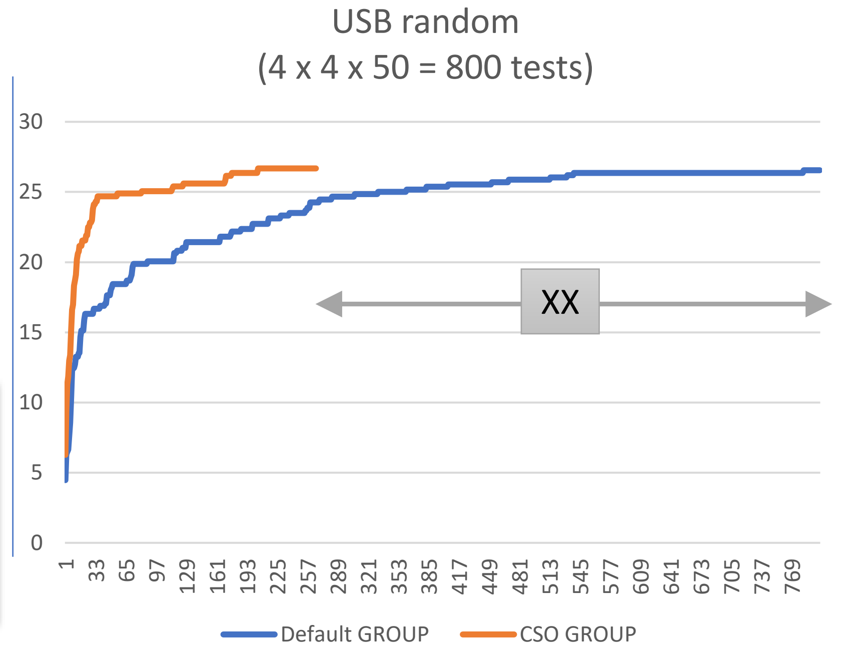
Project Detail | Exclusion Manager | Plan Search | Message | Change History | Restore Items

Optimized Regression with VMS+VSO.ai


Day 0 regression
Day n+1 regression
Less Gr
VMS+VSO.ai Native Integration
VCS compile
Static
Dynamic
User Hints
ML
VSO.ai Coverage Inference
% simv run +testN
VSO.ai Coverage Directed Solver
Coverage Report
VSO.ai Illegal cov and RCA report

ITS
VMS Built-in Intelligent Test Selection

VMS+VSO.ai
- Optimize out tests/seeds
- Control seeds and runtime args
- Prioritize and extend rare tests



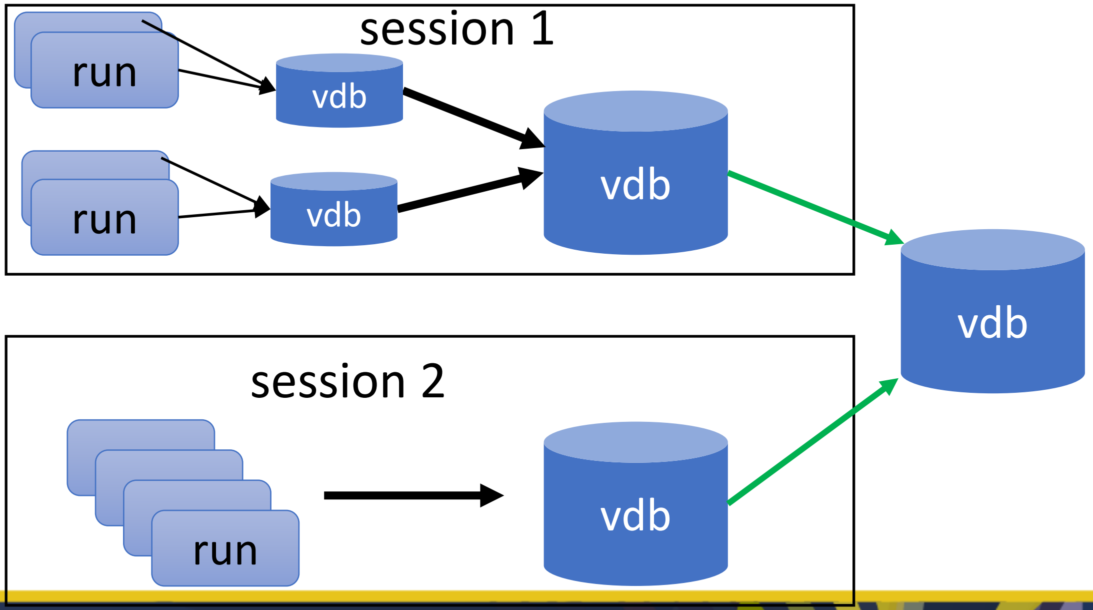
Standalone Aggregation or Integrated Inside Runner

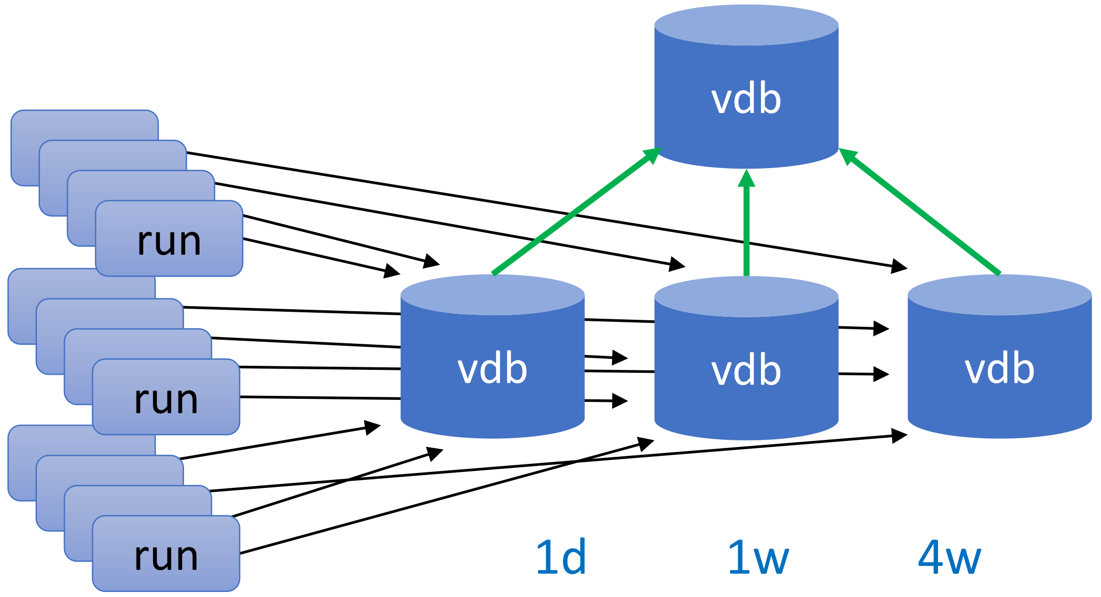
Runner Coverage Merging

- Merges are tied to specific build and runs
- Final merged VDB and session report
- Session VDBs can be combined separately

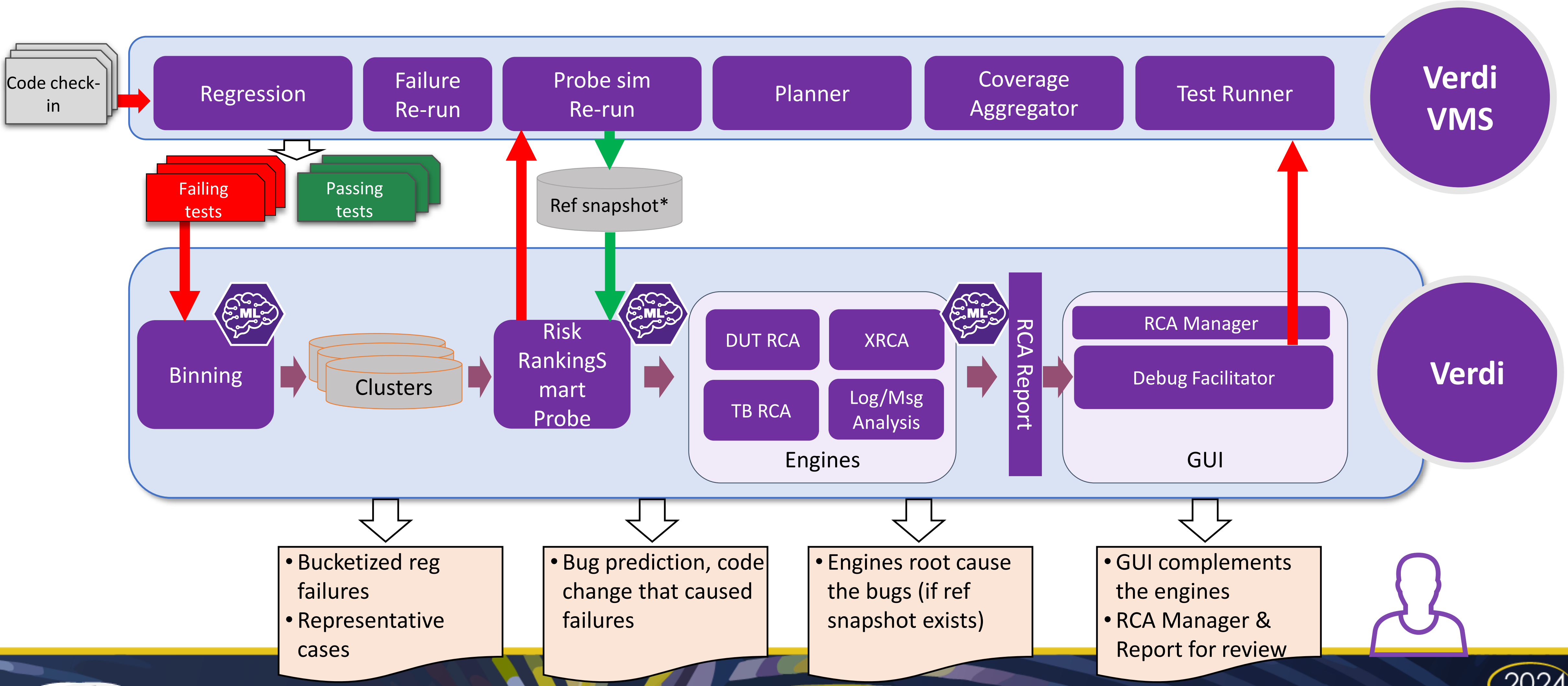
Standalone Coverage Merging

- VDBs come from multiple builds and runs
- Continuous merge incoming VDBs
- Tree combines results at higher levels





ML-Based, Automated Regression Debug



- Bucketized reg failures
- Representative cases

- Bug prediction, code change that caused failures

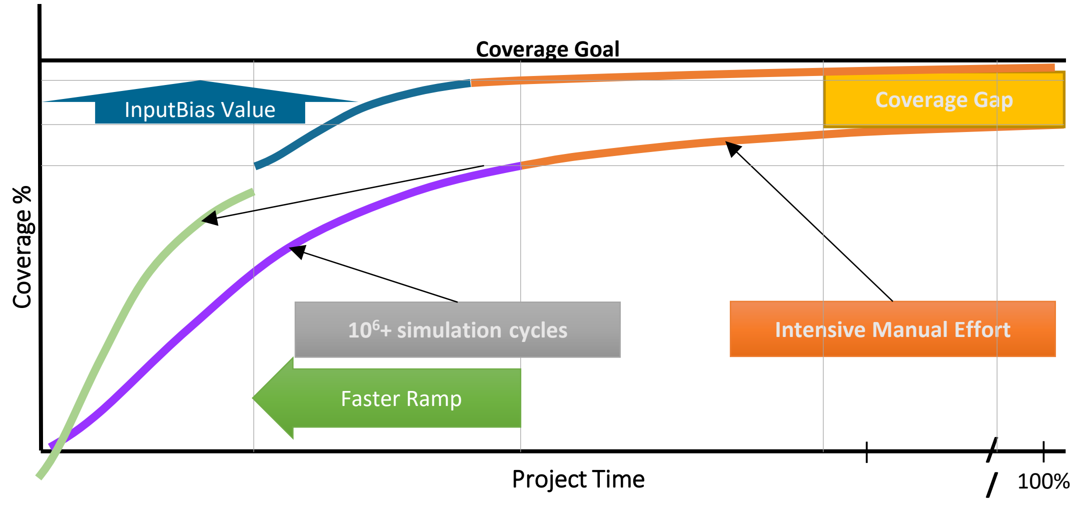
- Engines root cause the bugs (if ref snapshot exists)

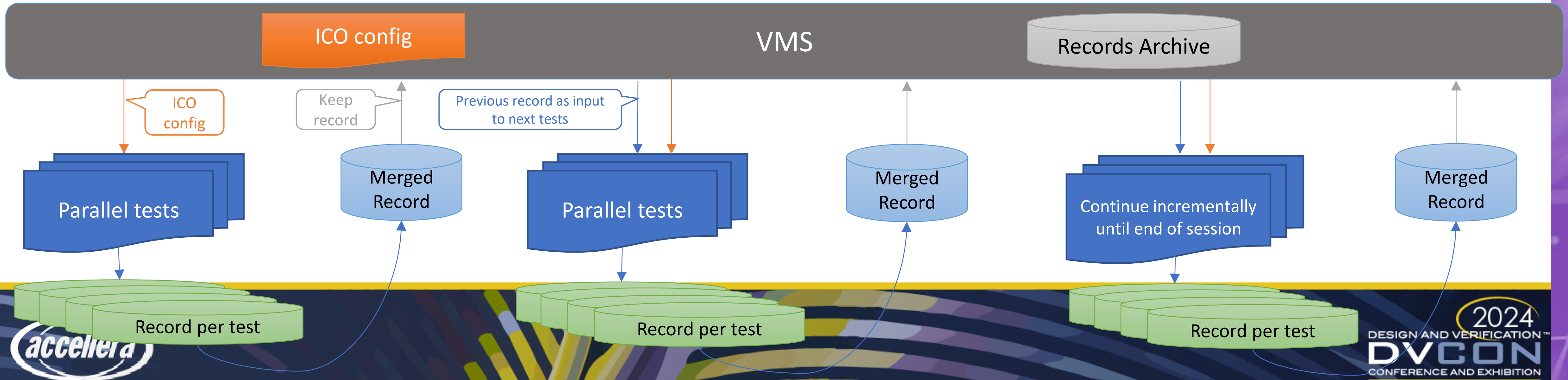
- GUI complements the engines
- RCA Manager & Report for review

Integration of Intelligent Coverage Optimization (ICO)

Enabling constraints biasing on full regression

- Achieve higher, faster coverage
- Catch more bugs
- Leverage VMS managed infrastructure
- Use simple interface
- Merge server scalable technology
- Run parent session
- Provides consistent debug and retry
- Reports
- Grade results

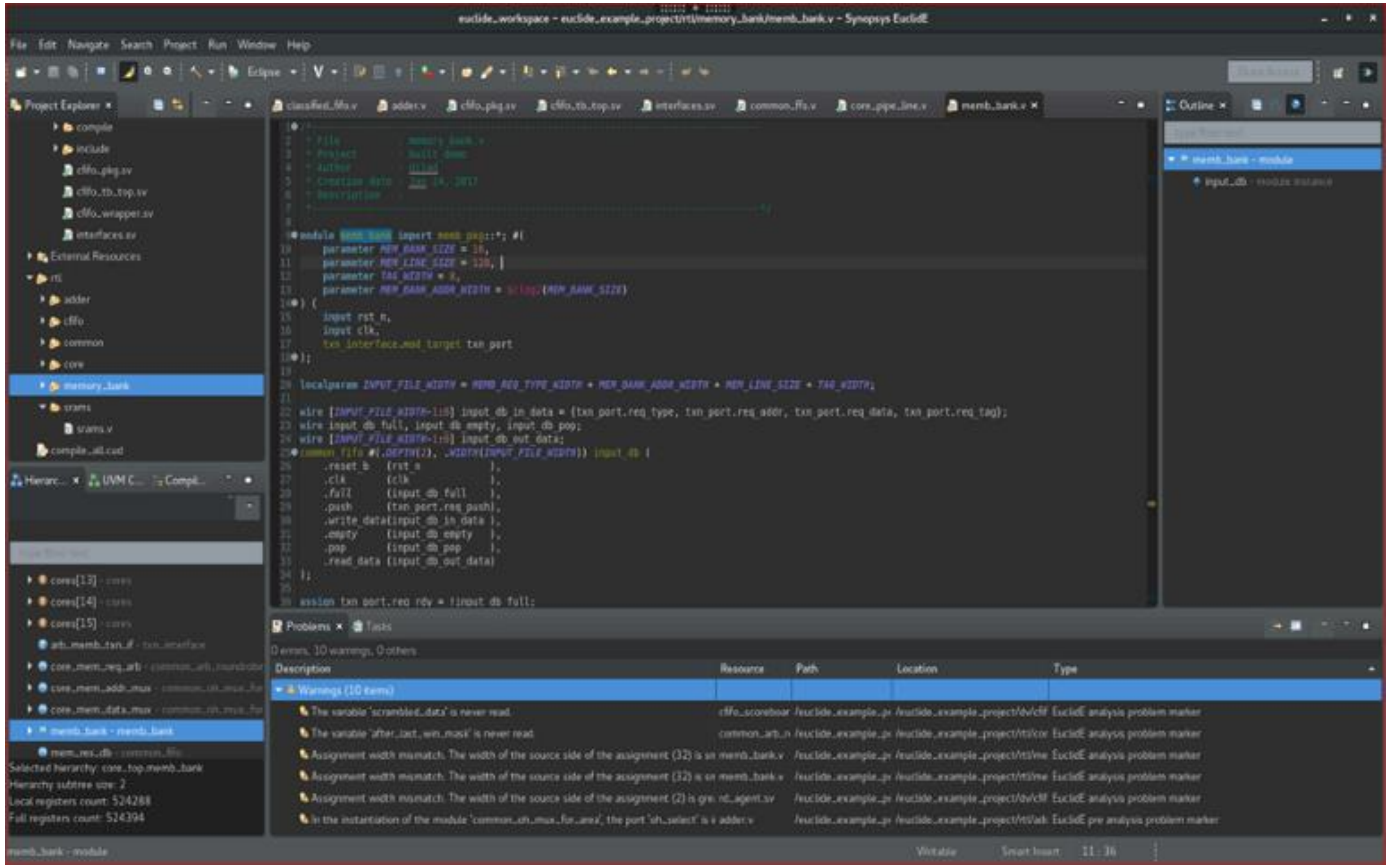




Euclide – Verdi IDE

Euclide IDE

- On-the-fly SVTB linter and design checks
- Content assistance
- Hover information
- Hyperlinks
- Design hierarchy tree
- UVM component hierarchy and more
- Task management



```
module memb_bank import memb_pkg::*; #(
    parameter MEM_BANK_SIZE = 16,
    parameter MEM_LINE_SIZE = 128,
    parameter TAG_WIDTH = 8,
    parameter MEM_BANK_ADDR_WIDTH = $clog2(MEM_BANK_SIZE)
) (
    input rst_n,
    input clk,
    txn_interface.mod_target txn_port
);

localparam INPUT_FILE_WIDTH = MEMB_REQ_TYPE_WIDTH + MEM_BANK_ADDR_WIDTH + MEM_LINE_SIZE + TAG_WIDTH;

wire [INPUT_FILE_WIDTH-1:0] input_db_in_data = {txn_port.req_type, txn_port.req_addr, txn_port.req_data, txn_port.req_tag};
wire input_db_full, input_db_empty, input_db_pop;
wire [INPUT_FILE_WIDTH-1:0] input_db_out_data;
common_fifo #(.DEPTH(2), .WIDTH(INPUT_FILE_WIDTH)) input_db (
    .reset_b    (rst_n             ),
    .clk        (clk               ),
    .full       (input_db_full     ),
    .push       (txn_port.req_push),
    .write_data(input_db_in_data ),
    .empty      (input_db_empty    ),
    .pop        (input_db_pop      ),
    .read_data (input_db_out_data)
);

assign txn_port.req_rdy = !input_db_full;
```

0 errors, 10 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (10 items)				
The variable 'scrambled_data' is never read.	cfifo_scoreboar	/euclide_example_pr	/euclide_example_project/dv/cfif	EuclidE analysis problem marker
The variable 'after_last_win_mask' is never read.	common_arb_n	/euclide_example_pr	/euclide_example_project/rtl/cor	EuclidE analysis problem marker
Assignment width mismatch. The width of the source side of the assignment (32) is sn	memb_bank.v	/euclide_example_pr	/euclide_example_project/rtl/me	EuclidE analysis problem marker
Assignment width mismatch. The width of the source side of the assignment (32) is sn	memb_bank.v	/euclide_example_pr	/euclide_example_project/rtl/me	EuclidE analysis problem marker
Assignment width mismatch. The width of the source side of the assignment (2) is gre	rd_agent.sv	/euclide_example_pr	/euclide_example_project/dv/cfif	EuclidE analysis problem marker
In the instantiation of the module 'common_oh_mux_for_area', the port 'oh_select' is i	adder.v	/euclide_example_pr	/euclide_example_project/rtl/adc	EuclidE pre analysis problem marker

Selected hierarchy: core_top.memb_bank
Hierarchy subtree size: 2
Local registers count: 524288
Full registers count: 524394

Euclide

Design & Testbench Checks

Advanced Rule Checking as You Type

RTL &

- **High-performance, Highly incremental engine for interactive design and testbench checks**
  - Complete elaboration, design resolution and pseudo-synthesis even on incomplete code
- **Advanced linter with deeper checks**
  - Over 1000 rules for RTL and testbench
- **Real-time design checks for down-stream EDA tools**
  - Simulation performance, emulation compliance and synthesis

[This slide intentionally blank]

[This slide intentionally blank]

Summary

- **Next-Gen Verdi** lets you avoid manual regression debug is tedious
  - Automate it with AI and advanced RCA technologies to debug any failing simulations
- Use **VC Execution Manager** (VMS) as a Unified Cockpit to create verification plans, manage simulation, gather coverage data and analyze date, optimize regression TAT and achieve faster coverage closure
  - Available with Verdi
- Create correct code by construction by using **Euclide**
  - Available with Verdi

2023
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
EUROPE
10 YEAR ANNIVERSARY

Thank You

accellera
SYSTEMS INITIATIVE