Next Generation Verdi : Overview of New Debug and Verification Management

Prasad Chelur Thirumalaiah Siddeshwara and
Rohit Kumar Ohlayan

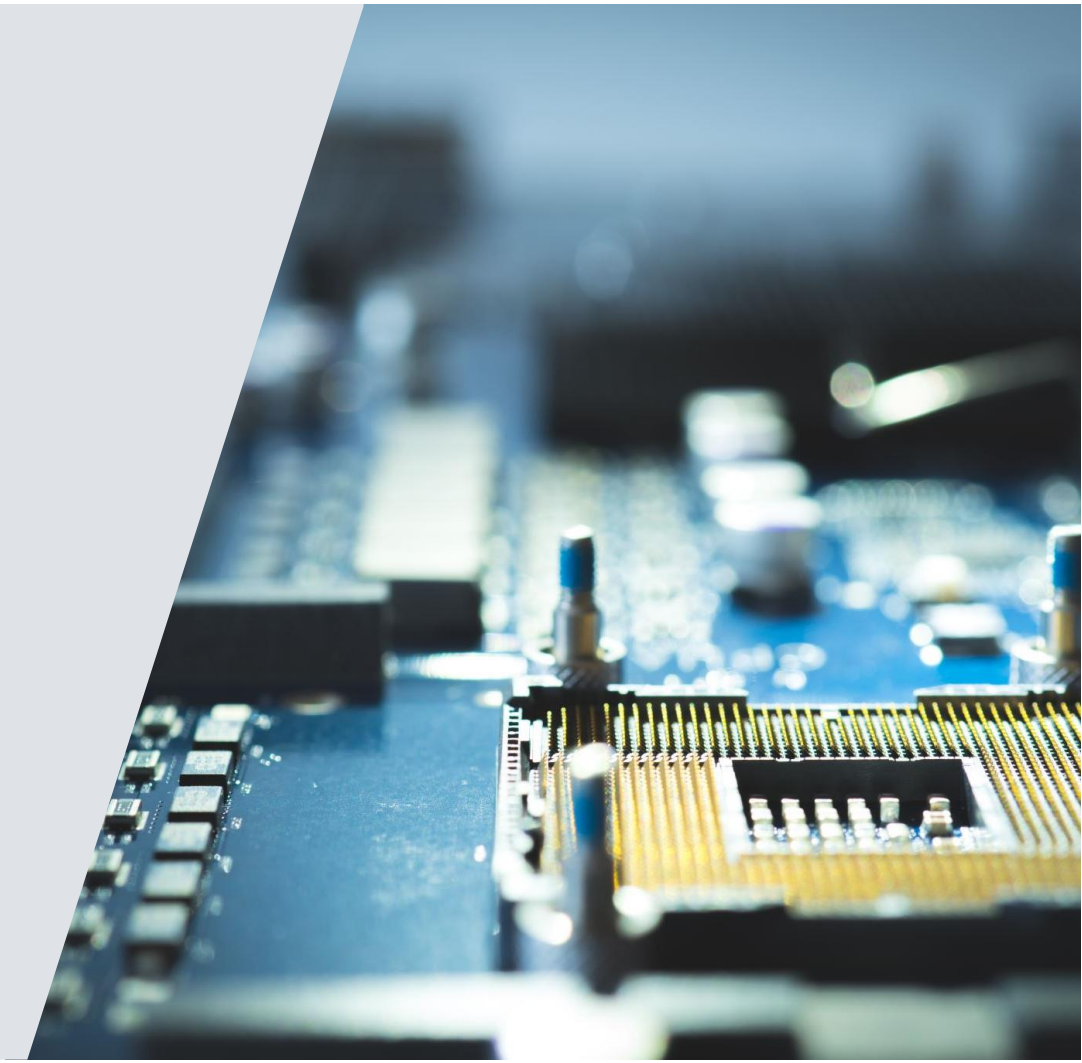
SYNOPSYS®

accellera SYSTEMS INITIATIVE™

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Agenda

- Motivation and Debug Flow
- AI Based debug automation
- Debug Decision Tree
- Verdi Verification Management System
- Refreshed GUI and IDE
- VC Replay
- Summary + Q&A



Prasad Chelur Thirumalaiah Siddeshwara - Presenter

R&D Engineering, Sr Manager, Synopsys Inc.

Prasad is Sr Manager, R&D having 12+ years of EDA experience with expertise in Testbench and Protocol debug part of Verdi. Prasad key Focus area includes general debug of Verdi, Debug and Performance analysis of industry standard and proprietary Protocol. Providing innovative solution for testbench debug like TBRCA, Protocol RCA and Log based debug.

Rohit K Ohlayan - Author

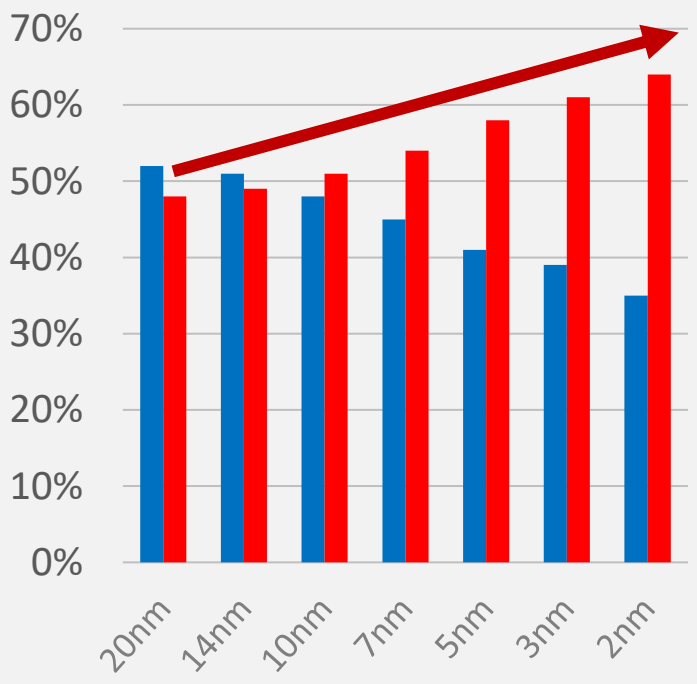
R&D Engineering Sr Director, Synopsys Inc.

Rohit is Sr Director, R&D and manages the Static Verification line of products in Synopsys. He has 25+ years of experience, working in SpyGlass - the industry leader in Static Verification - and its various apps like Lint, RDC, Functional Lint, Euclide which is IDE based Linting. Rohit has architected various solutions around Linting flows in VC SpyGlass, and holds multiple patents and publications to his name

Motivation and Debug Flow

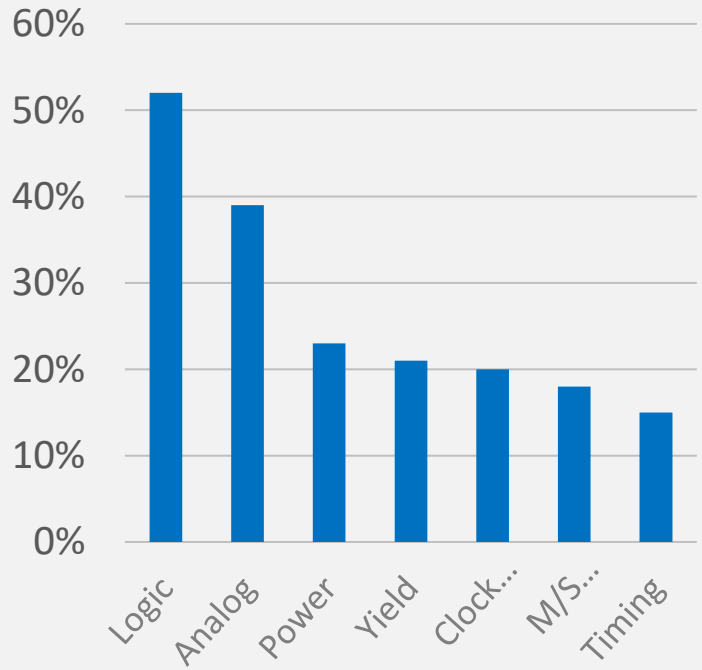
Impact on Right First-Time Silicon

Fewer First-Time Right



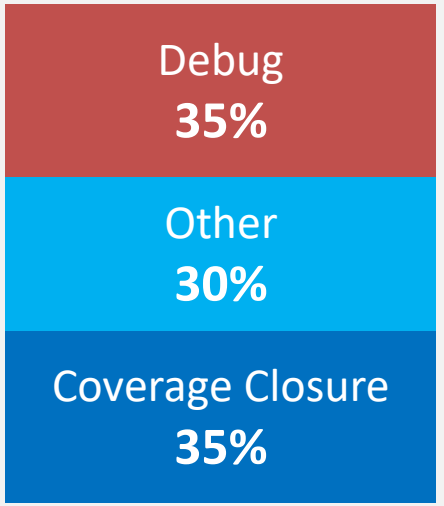
More Respins

Logic Bugs Dominating



Main Cause of Respins

Verification Time Spent

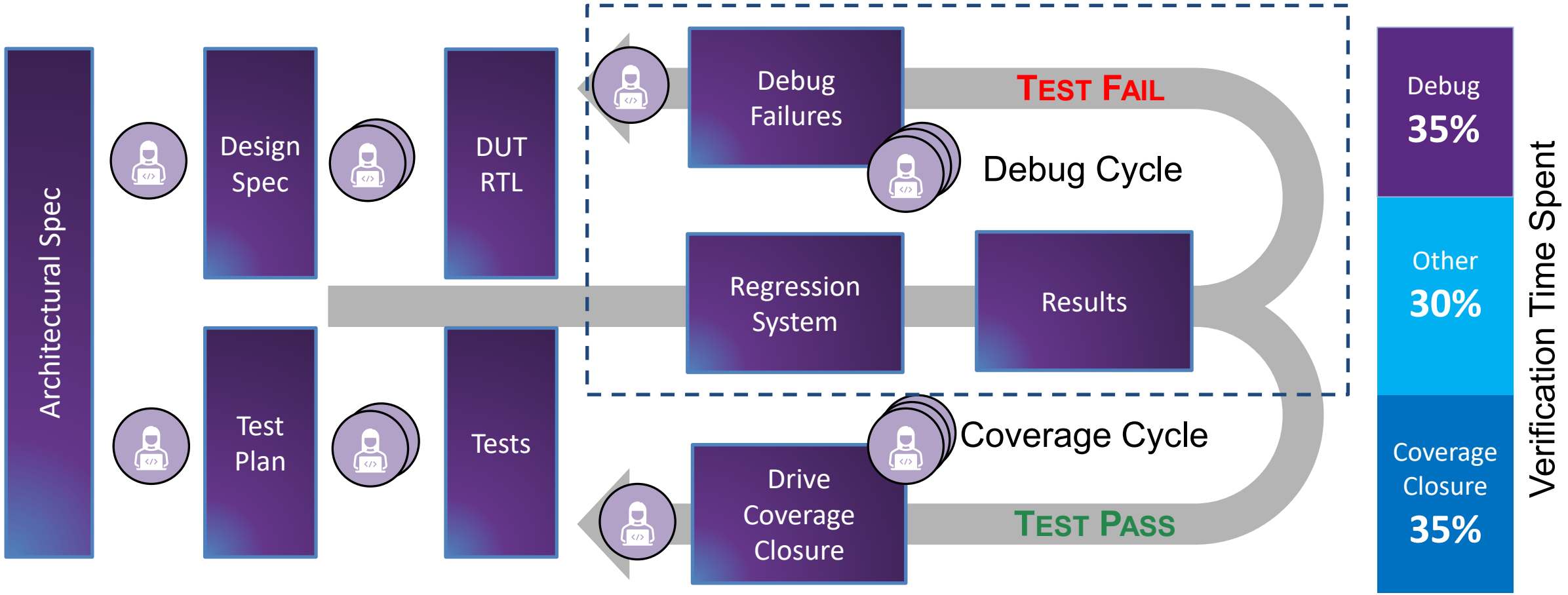


Need to Reduce Debug Time!

Source: Wilson Report 2022

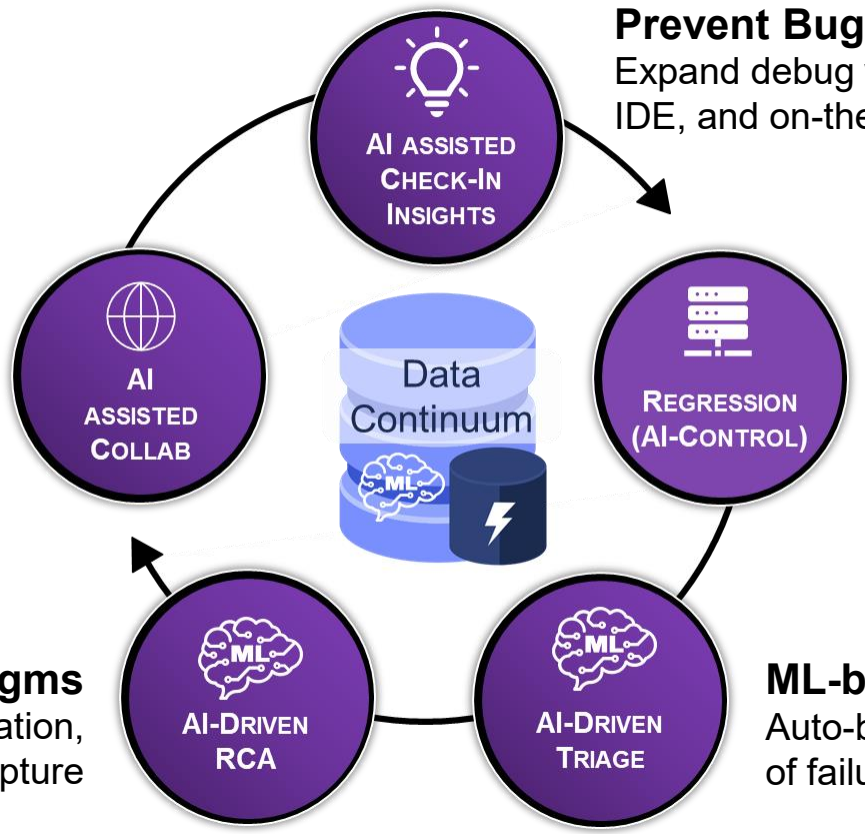
Manual Process

How is Debug Automated Today?



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

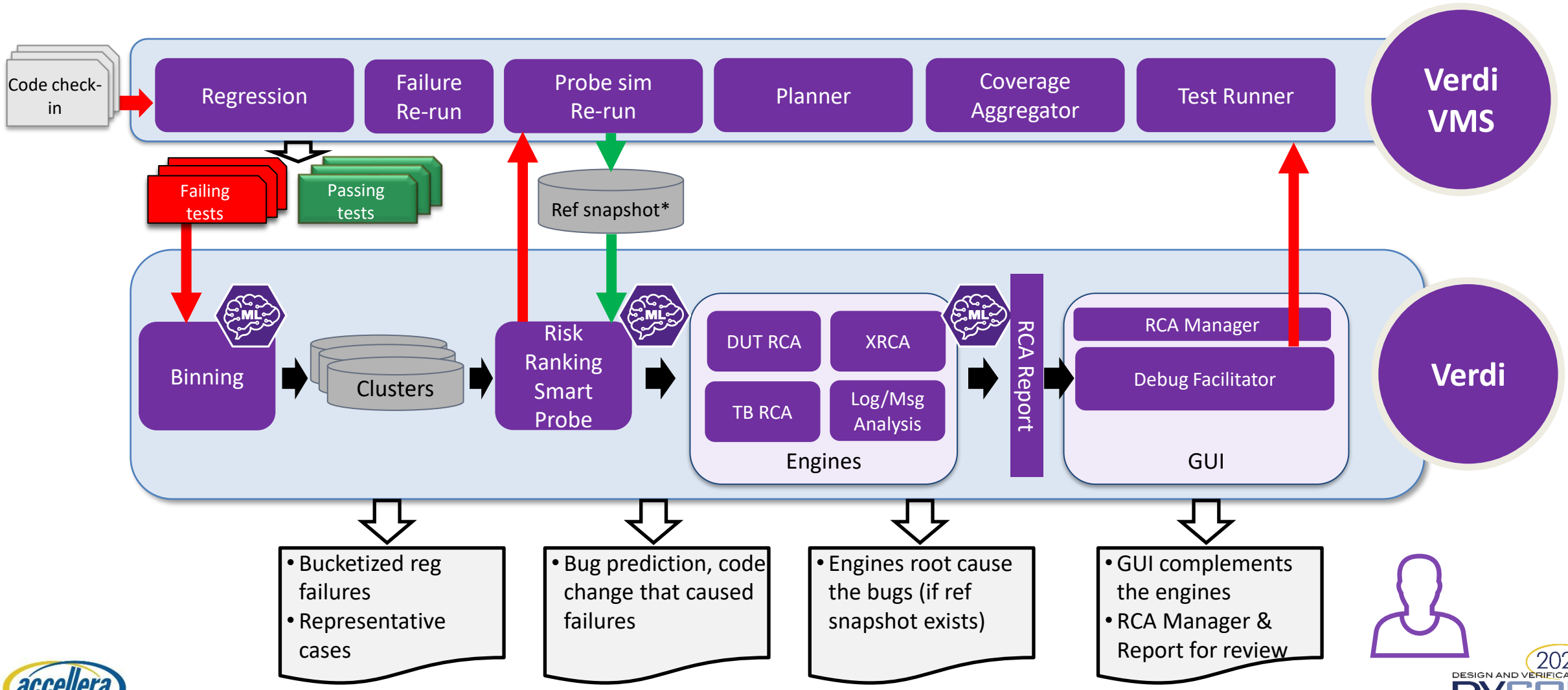
Regression Management
ML-optimized daily regression, reduce HW costs and increase efficiency

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

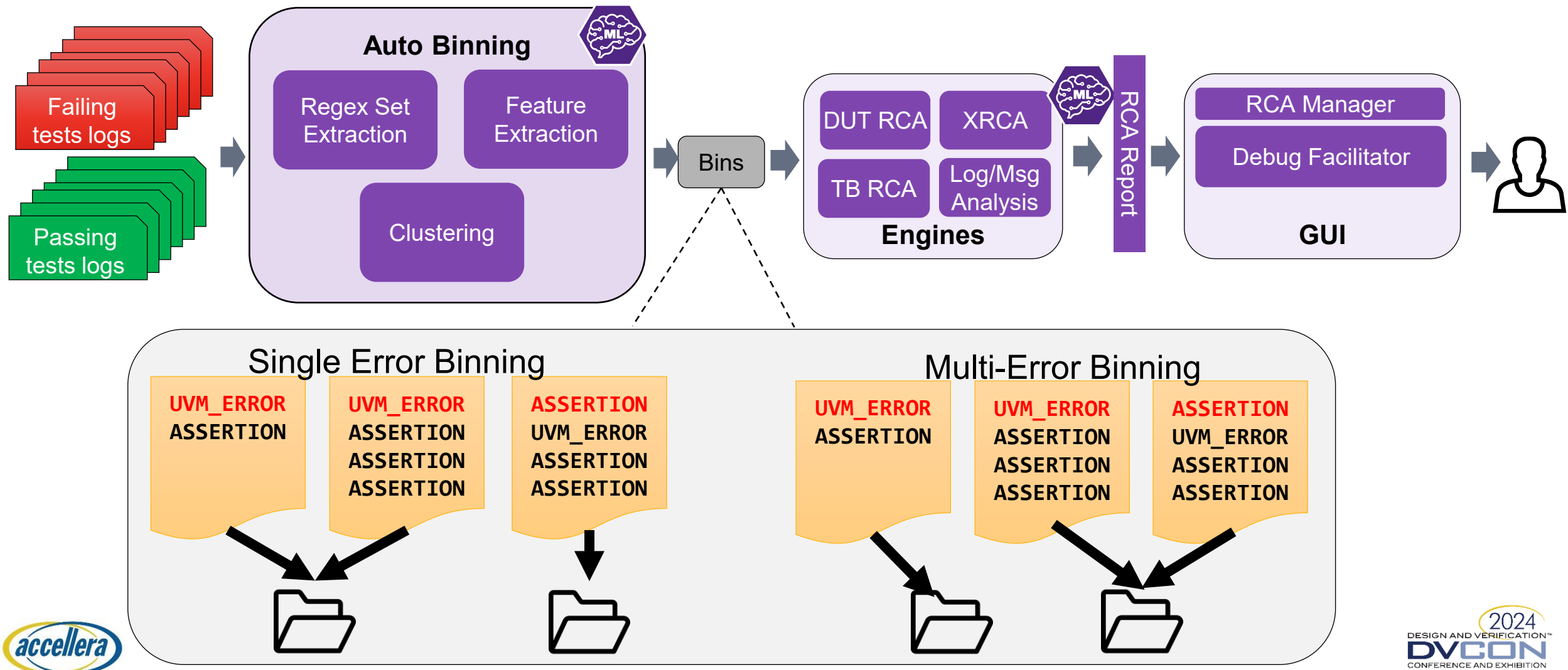
Global Collaboration
Multi-site, multi-user collaborative debug, knowledge reuse

ML-Based, Automated Regression Debug



AI-Based Debug Automation

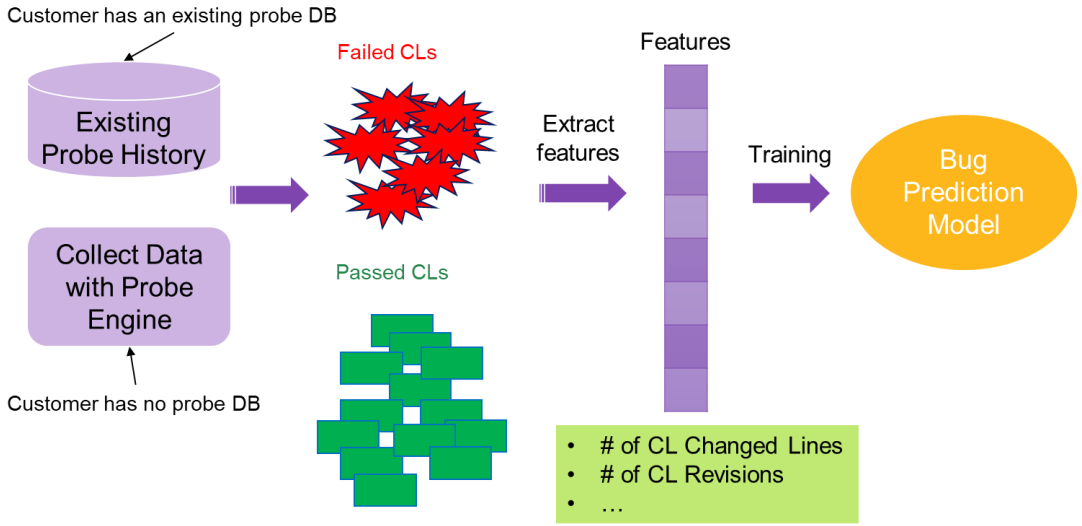
Regression Binning with ML



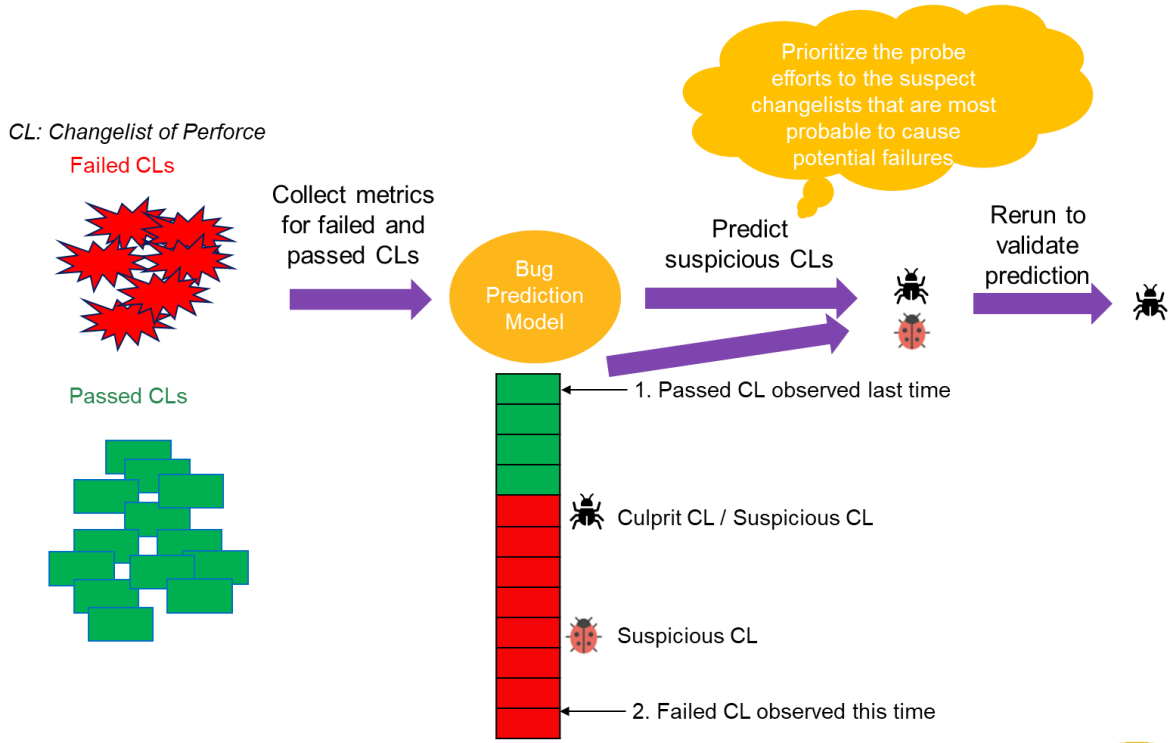
Smart Probe

- ML-based ranking of change lists to root cause regression failures

Train the bug prediction model

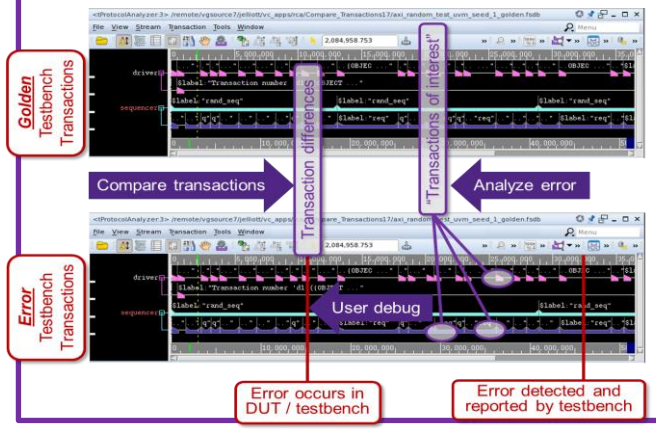


Bug prediction use model



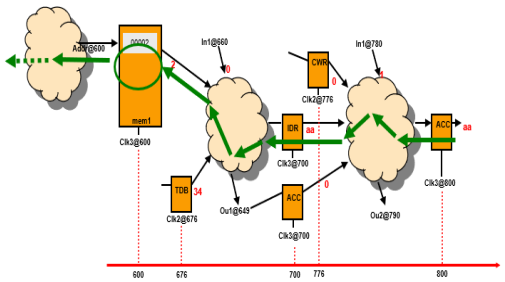
RDA Component Technology

TBRCA



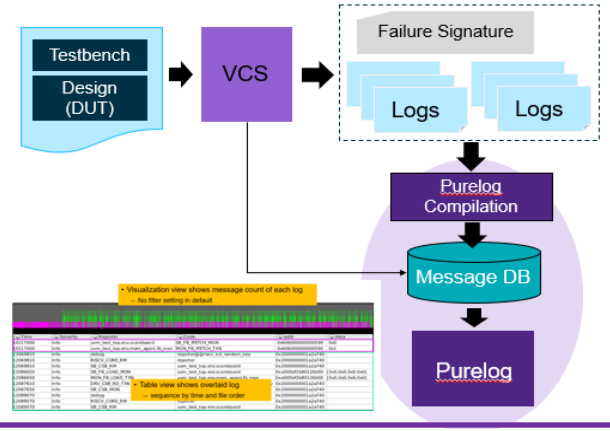
- Transaction Diff – Diff the transaction in the reference vs failing FSDB
- Message Analysis – No ref FSDB required. Analyze info from the “error” message.
- Report transaction of interest linked to the error

DUTRCA



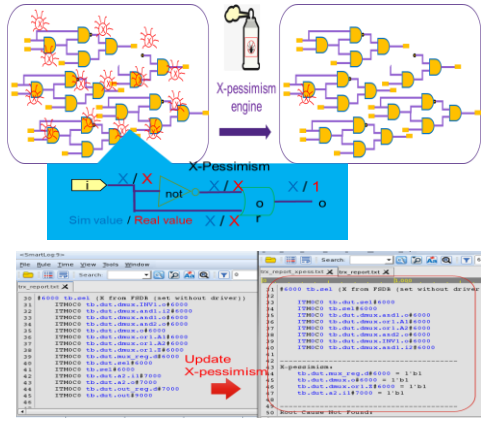
- Adopt roll back mechanism and TraceDiff technology to narrow down DUT problem
- Temporal Flow View to analyze root cause path

PureLog



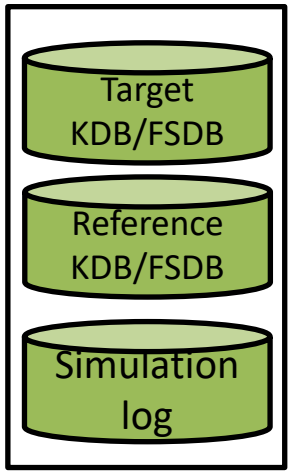
- Display messages in table and message chart view
- Intuitive filter, search, data mining operations
- API to access database
- ML technology to detect abnormal message

XRCA / with X-Pessimism



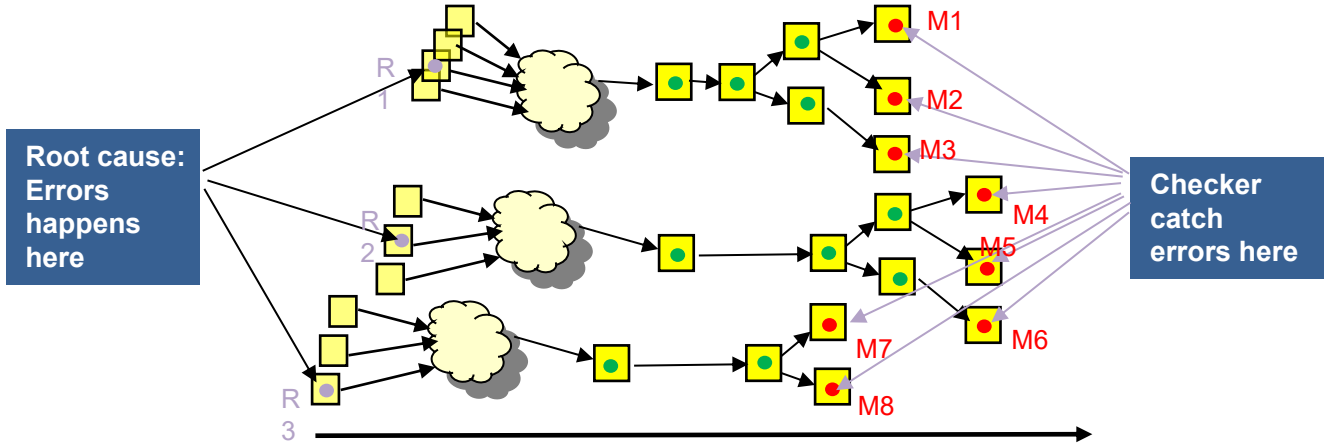
- Scan X signals in FSDB and trace the root cause of X.
- Handle large amount of X signals in batch mode
- Formal engine to identify X pessimism to remove the noise

Auto Debug Design Change Problem



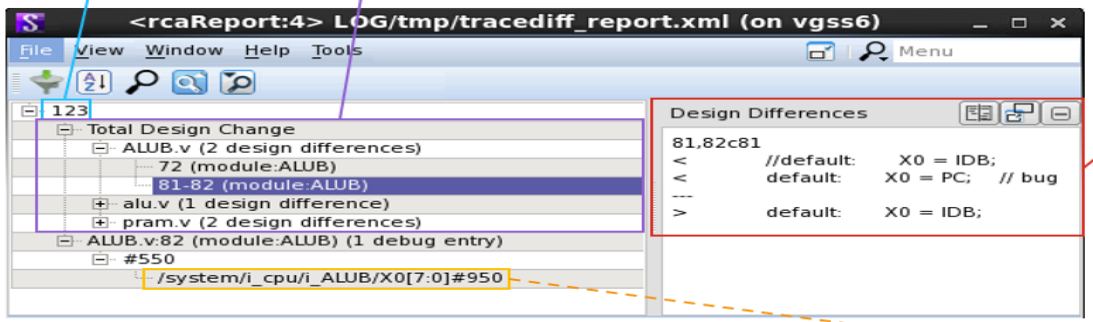
autorca –cfg run.yaml

DUTRCA



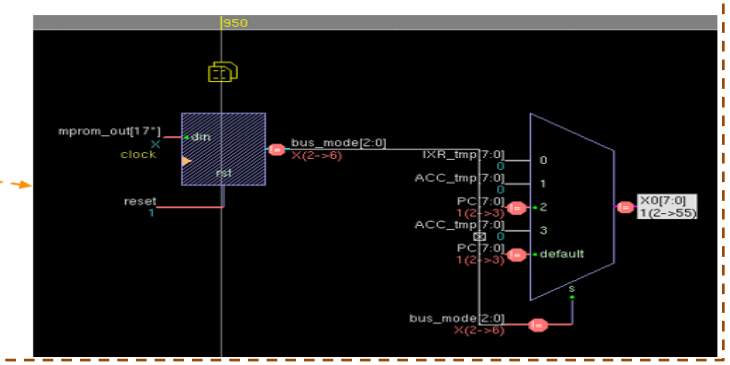
Changelist id that make regression fail

Total design changes of the changelist id

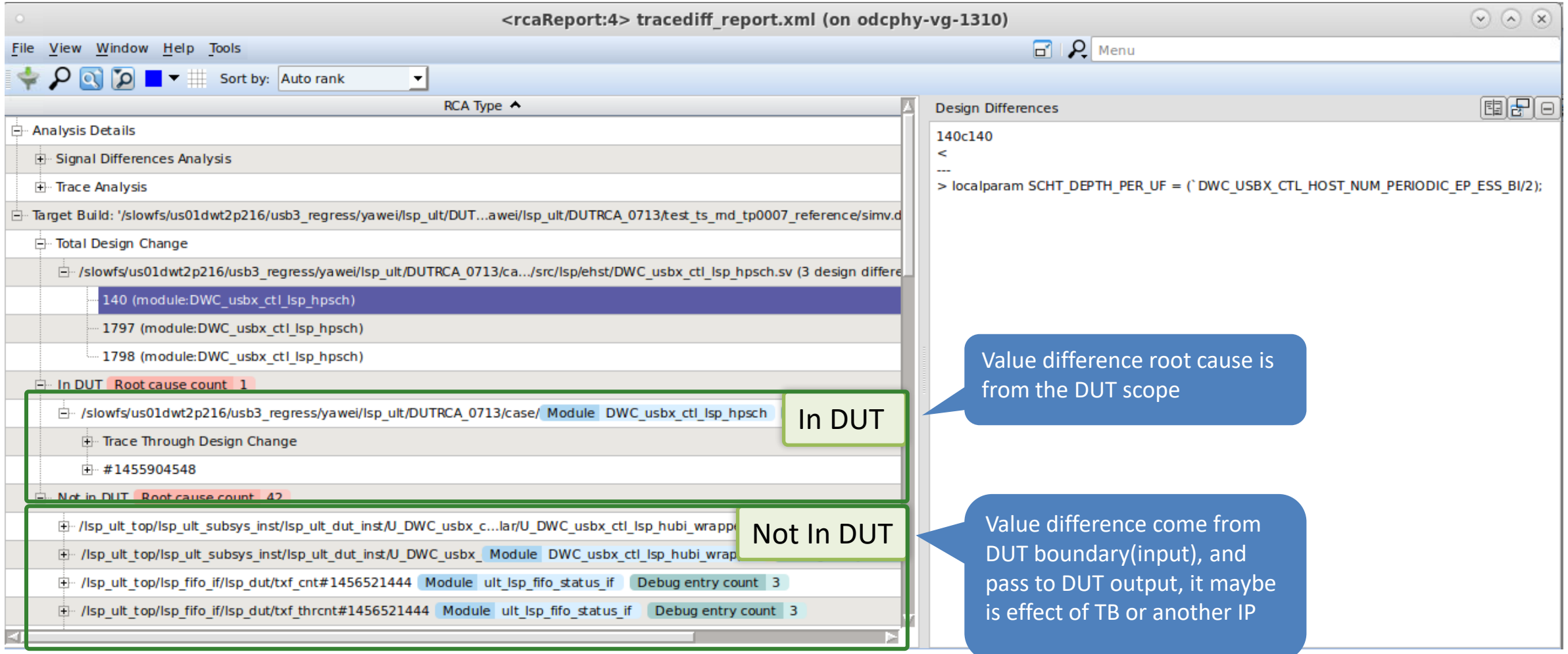


Show design difference

Debug entry point: double click will show tracing path from debug entry point to root cause

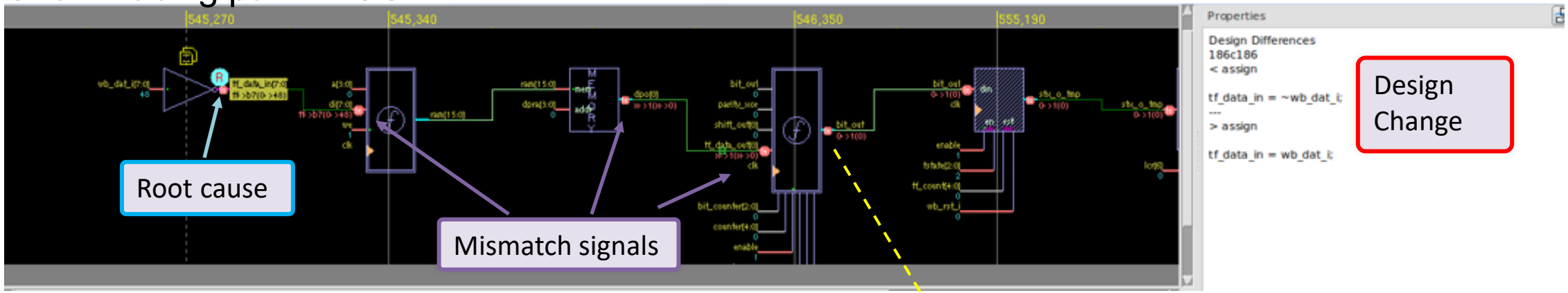


DUTRCA Report

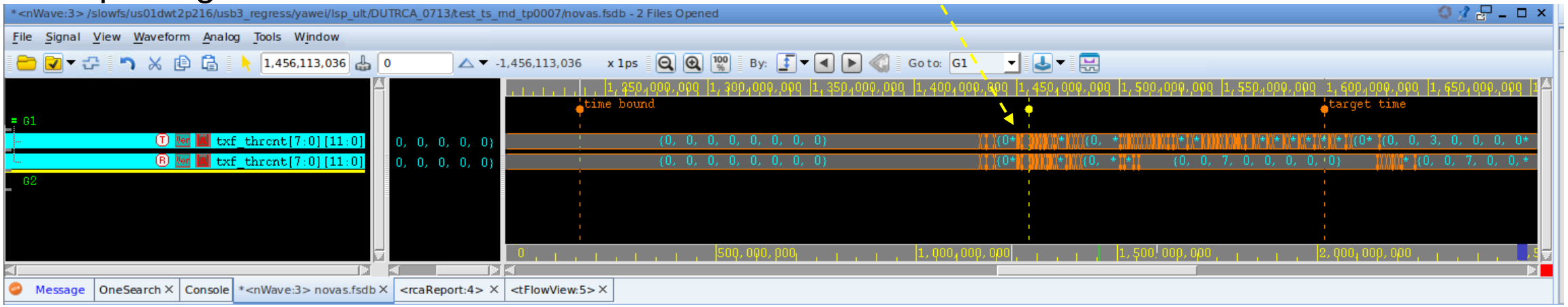


DUTRCA Report: Show Tracing Path

Show tracing path in GUI



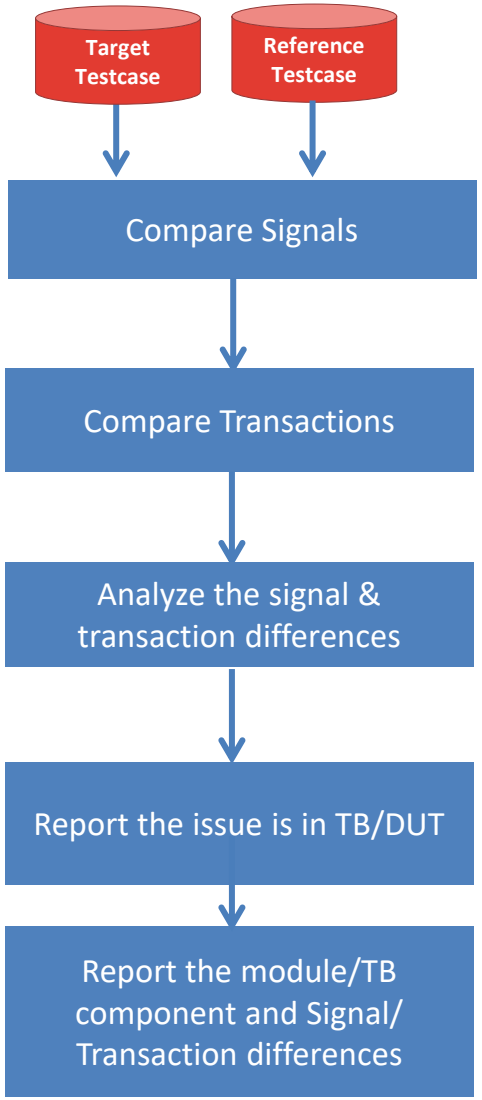
Show pair signals in waveform



accellera SYSTEMS INITIATIVE

2024 DESIGN AND VERIFICATION DVCON CONFERENCE AND EXHIBITION INDIA

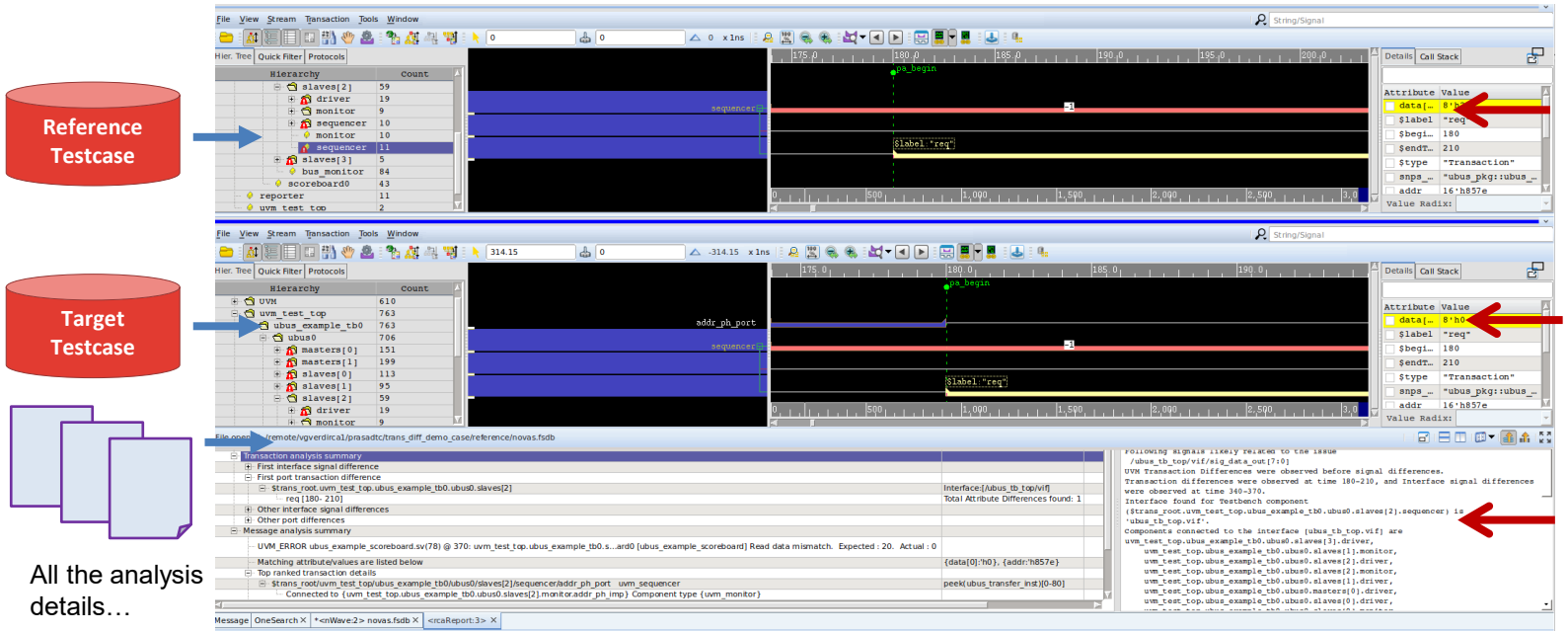
Transaction Diff : High Level Flow



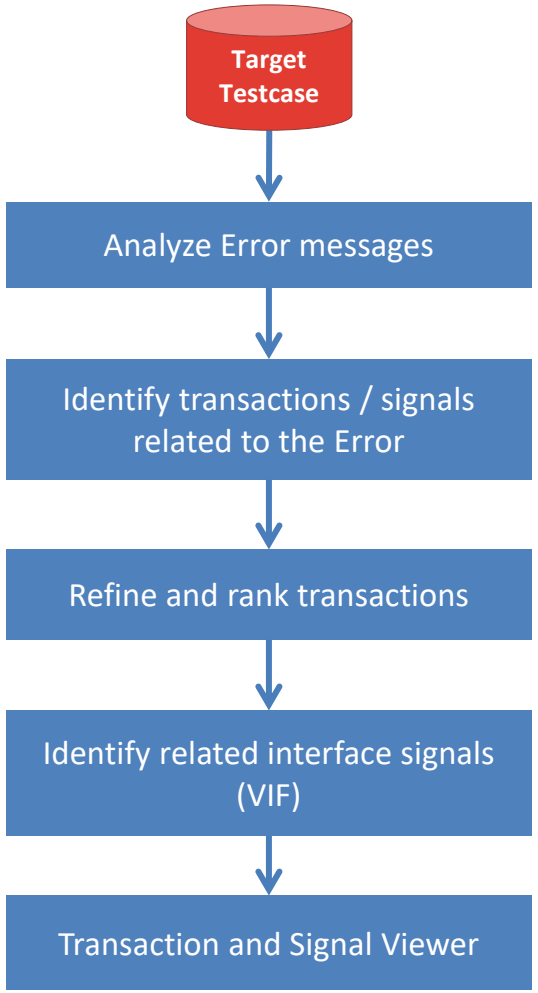
Report

```
TBAR_INFO:  Initializing....
TBAR_INFO:  Analyzing Signals ...
TBAR_INFO:  Comparing the interface signals...
TBAR_INFO:  Differences found in VIF interfaces.
TBAR_INFO:  Running trace diff....
TBAR_INFO:  No signals read from trace diff report.
TBAR_INFO:  Differences found during signal analysis.
TBAR_INFO:  Compare UVM/OVM characteristics,  Hierarchy, Connections, Objection, Trace etc..
TBAR_INFO:  No differences found in UVM/OVM characteristics
TBAR_INFO:  Comparing UVM/OVM transaction data...
TBAR_INFO:  Differences found in UVM/OVM agents
TBAR_INFO:  Comparing VIP transaction data...
TBAR_INFO:  Differences found in VIP data

TBAR_INFO:  Analyzing.....
TBAR_INFO:  ***** The issue likely in Testbench *****
TBAR_INFO:  Following signals likely related to the issue
 /test_top/chi_if/rn_if[0]/debug_modport/rx_rsp_obj_num[31:0]
TBAR_INFO:  Components connected to the interface {test_top.chi_if.rn_if[0]} are uvm_test_top.env.amba5_system_env.chi_system[0].rn[0].link_mon,
        uvm_test_top.env.amba5_system_env.chi_system[0].rn[0].tx_dat_flit_seqr,
        uvm_test_top.env.amba5_system_env.chi_system[0].rn[0].tx_req_flit_seqr,
        uvm_test_top.env.amba5_system_env.chi_system[0].rn[0].link,
        uvm_test_top.env.amba5_system_env.chi_system[0].rn[0].tx_rsp_flit_seqr,
        uvm_test_top.env.amba5_system_env.chi_system[0].rn[0].rn_snp_xact_seqr,
        uvm_test_top.env.amba5_system_env.chi_system[0].rn[0].prot_svc_seqr,
        uvm_test_top.env.amba5_system_env.chi_system[0].rn[0].link_svc_seqr,
        uvm_test_top.env.amba5_system_env.chi_system[0].rn[0].prot,
        uvm_test_top.env.amba5_system_env.chi_system[0].rn[0].prot_mon,
        uvm_test_top.env.amba5_system_env.chi_system[0].rn[0].virt_seqr,
        uvm_test_top.env.amba5_system_env.chi_system[0].rn[0],
        uvm_test_top.env.amba5_system_env.chi_system[0].rn[0].rn_xact_seqr.
TBAR_INFO:  ***** First transaction difference found in Monitor component {$trans_root.uvm_test_top.env.chi_ic_env.rn[0].prot_mon.item_observed_port} is
 'write(xact_factory)' *****.
TBAR_INFO:  Monitor is connected to {$trans_root.uvm_test_top.env.chi_protocol_layer_scoreboard.passive_rn_item_observed_response_export} component type {Scoreboard}.
TBAR_INFO:  The issue likely in Monitor Testbench component, however if the Scoreboard is passing the transaction back then issue likely in Scoreboard.
TBAR_INFO:  ***** First Issue found in VIP transaction {svt_chi_request_wr_writenosnpfull_cleansharedpersistsep_transaction} in stream {$trans_root_VIP.env.chi_ic_env.rn_0.svt_chi_request_wr_writenosnpfull_cleansharedpersistsep_transaction} *****
TBAR_INFO:  Please run the { /remote/vgrndn2/prasadtc/vip_tbrca_eng/amba/verdiLog/tbAutoRCA/reports/streamReports/run_verdi } script to view the transaction differences inside Verdi.
```



Message Analysis: High Level Flow

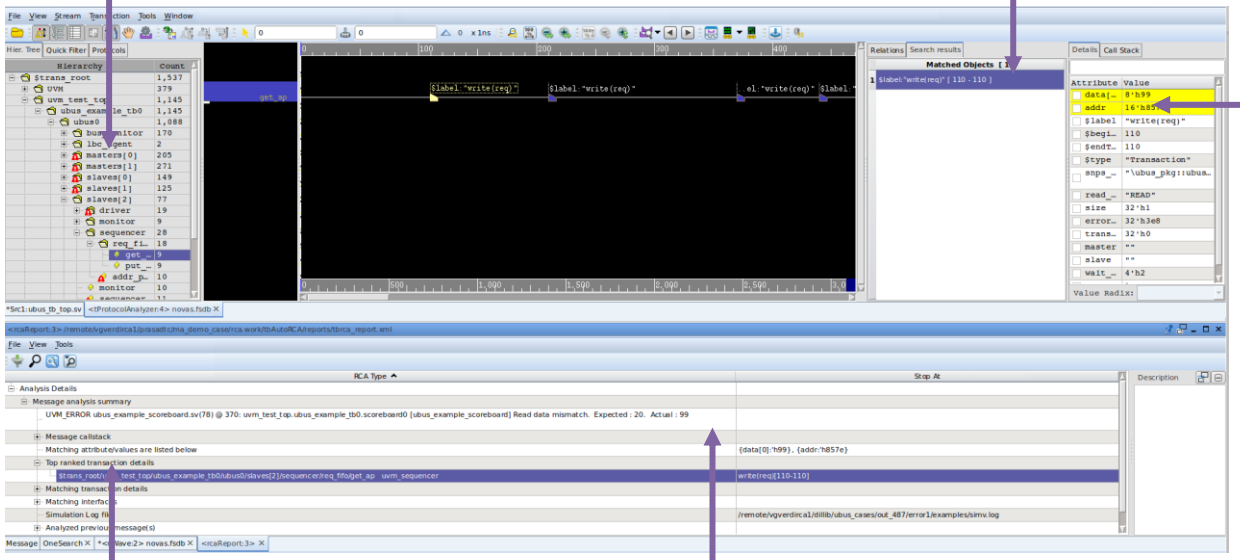


Report

```
TBAR_INFO: Started analyzing debug message(s) from simulation log file {/remote/vgsource12/ikshvaku/TD.VERDI_REG/unit_VERDI/unittest/evProds/tbAutoRCA/testData/swayCases/TB_ERRROR/out_230/examples/simv.log}
TBAR_INFO: Simulation log file contains ERRORS(14) debug messages.
TBAR_INFO: Analyzing the first ERROR log message.
TBAR_INFO: Analyzing the below log message.
    {UVM_ERROR ubus_example_master_seq_lib.sv(206) @ 80: uvm_test_top.ubus_example_tb0.ubus0.masters[1].sequencer@@loop_read_modify_write_seq.rmw_seq [read_modify_write_seq]
loop_read_modify_write_seq.rmw_seq Read Modify Write Read error!}
TBAR_INFO: Completed analyzing debug message(s) from simulation log file.
TBAR_INFO: Matching attribute/values are listed below.
TBAR_INFO:   {addr:020a}, {data[0]:e9}
TBAR_INFO: Matching transactions after analyzing actual, next and previous messages are listed below
TBAR_INFO:   Stream: {$trans_root/uvm_test_top/ubus_example_tb0/ubus0/masters[1]/driver/seq_item_port} Matching transactions: {2}
TBAR_INFO:   Stream: {$trans_root/uvm_test_top/ubus_example_tb0/ubus0/masters[1]/sequencer} Matching transactions: {1}
TBAR_INFO: Total matching transactions {3}
TBAR_INFO: Matching virtual interface paths are listed below
TBAR_INFO:   /ubus_tb_top/vif  Signals: {sig_read,sig_write,sig_addr}
TBAR_INFO: Top ranked transaction details are listed below
TBAR_INFO:   Transaction {get_next_item(req)[80-80]} Stream {$trans_root/uvm_test_top/ubus_example_tb0/ubus0/masters[1]/driver/seq_item_port} Component type {uvm_driver} Port type {uvm_seq_item_pull_port}
TBAR_INFO:     Connected to {uvm_test_top.ubus_example_tb0.ubus0.masters[1].sequencer.seq_item_export} Component type {uvm_sequencer} Port type {uvm_seq_item_pull_imp}
TBAR_INFO: Please run the {/slowfs/vgvips19/prasadtc/verdiLog/tbAutoRCA/reports//simvErrorAnalysisLog/run_verdi} script to view the simv log file analysis results in Verdi.
```

Larger set of transactions

Smaller set of Related transactions



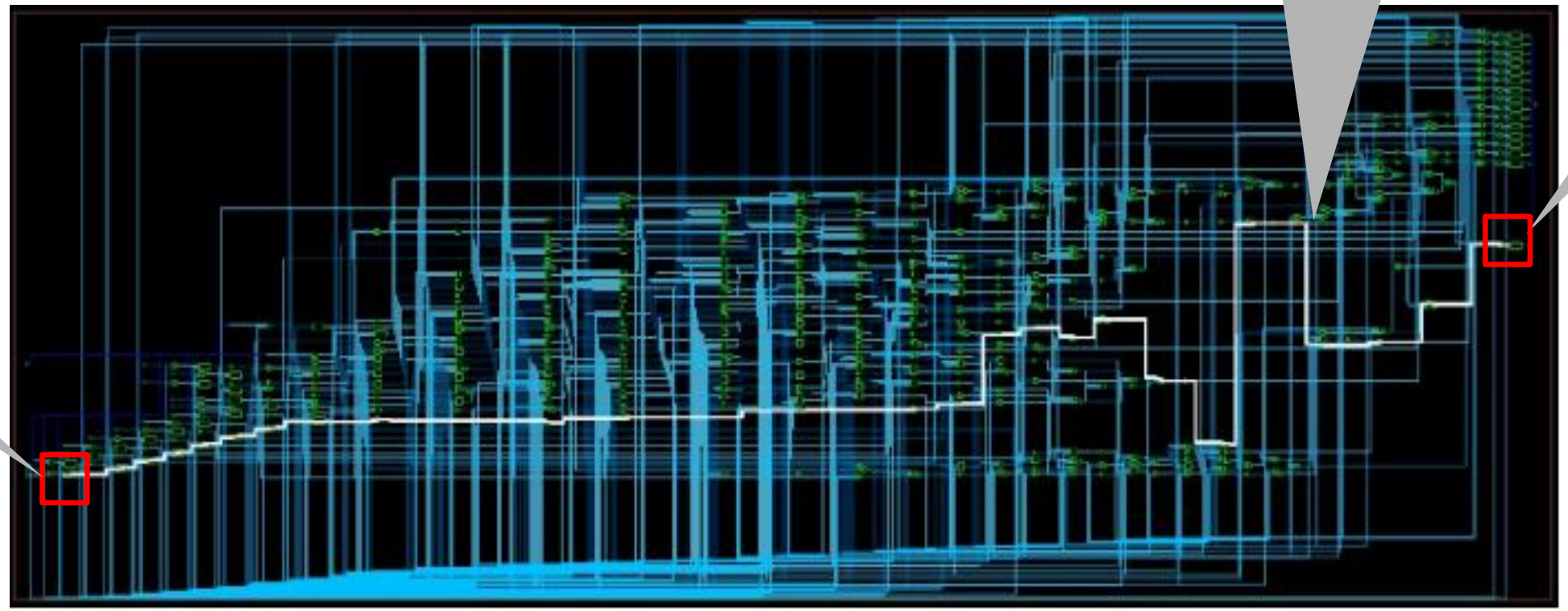
Interested transaction attributes

Matching transaction details

Simulation log message

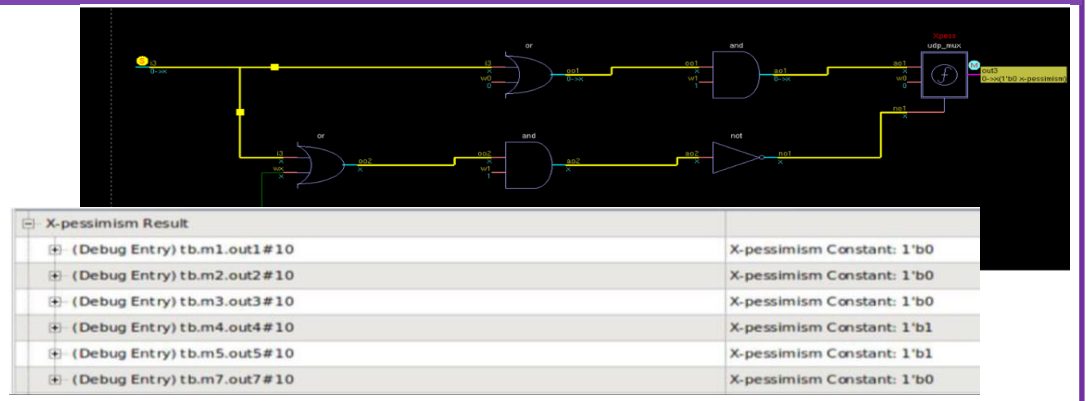
Backward Tracing to X Root Cause

Can Work on RTL Level Design or Gate Level Design

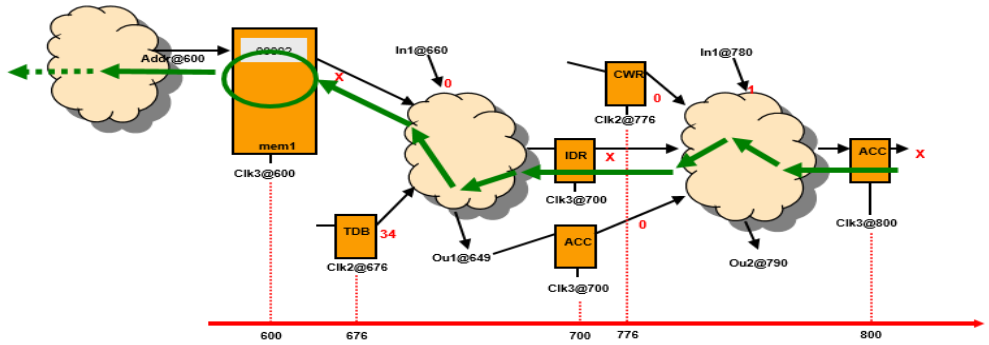


XRCA Technology -- Value Tracing

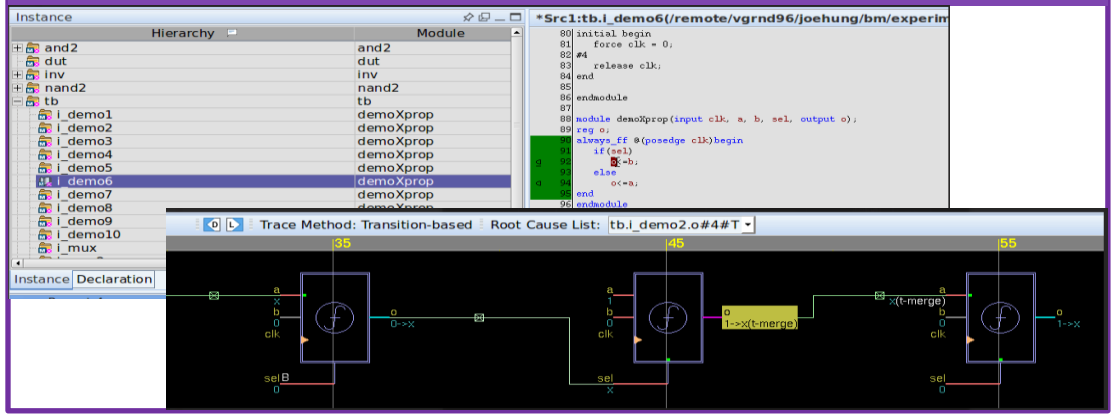
XPESS



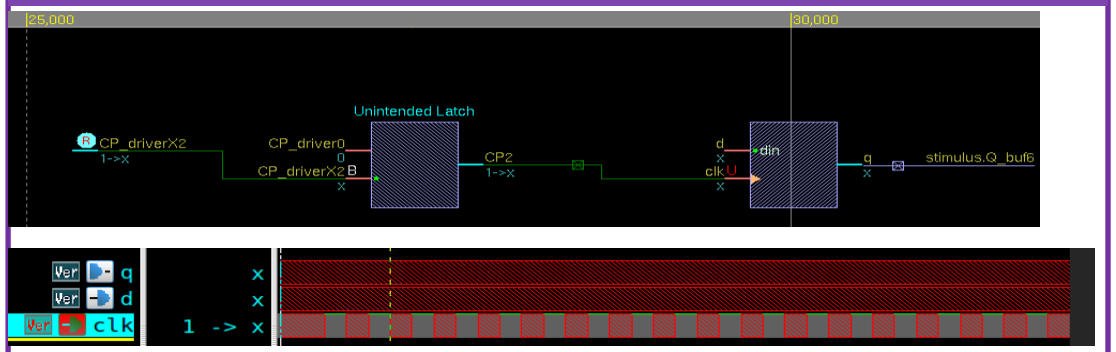
TraceX



Support Low Power / Xprop



X value on Register / Detect Reset Problem



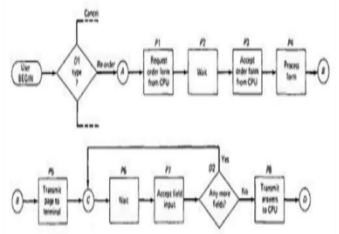
accellera SYSTEMS INITIATIVE™

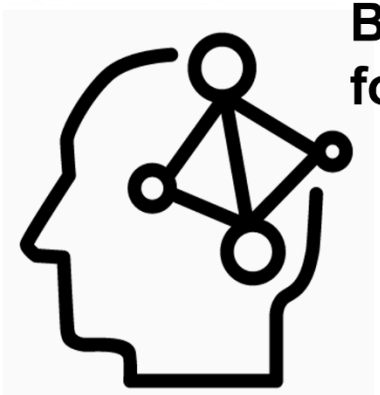
2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Debug Decision Tree

Debug Decision Tree (DDT)

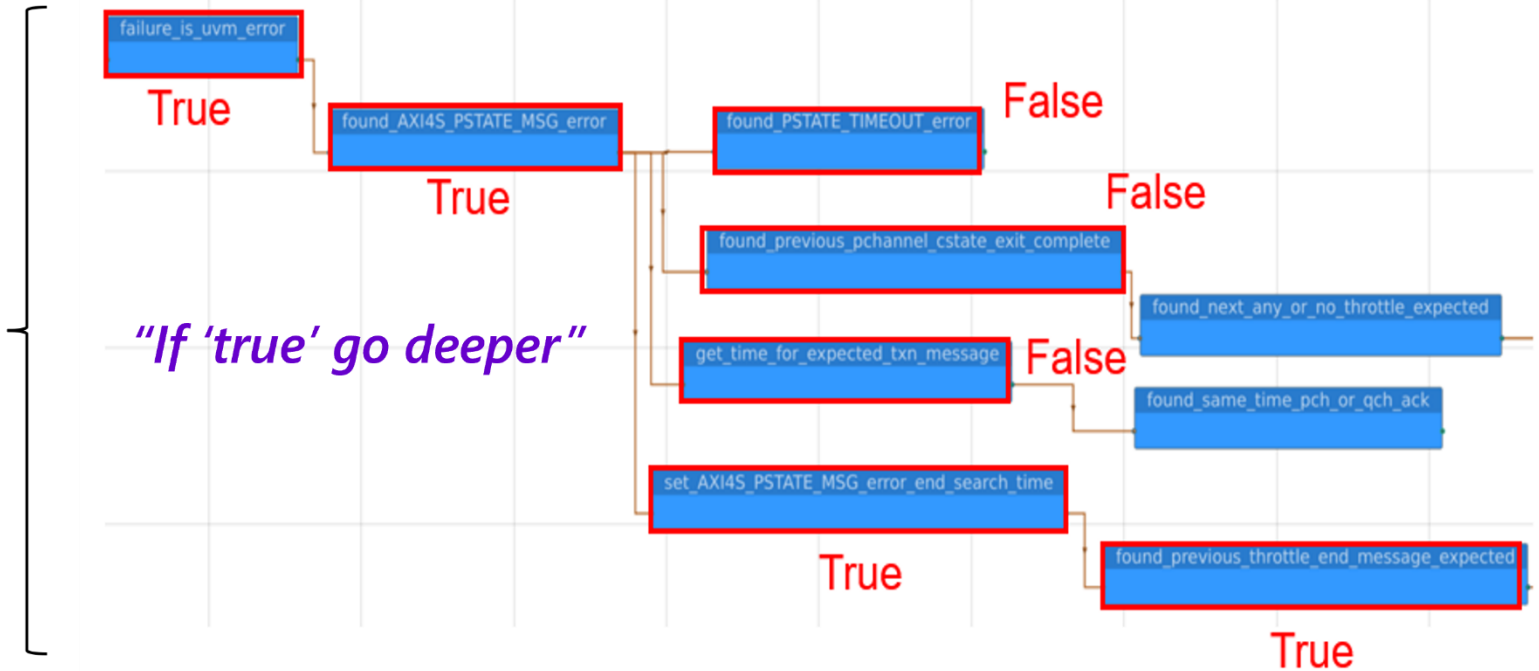
A tool for capturing, sharing, and executing debug knowledge across platforms





Bug found !

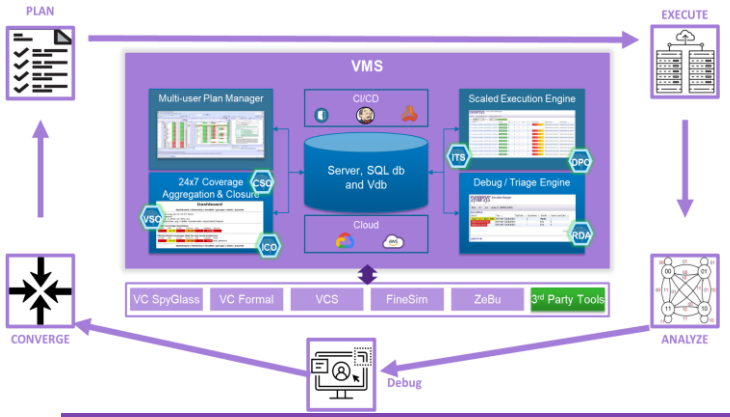
Collected algorithms in heads
Found the data required to feed those algorithms



Verdi Verification Management System

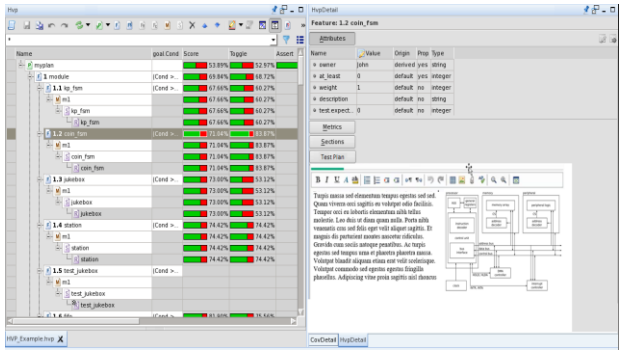
Verification Management System

Manager and Dashboard



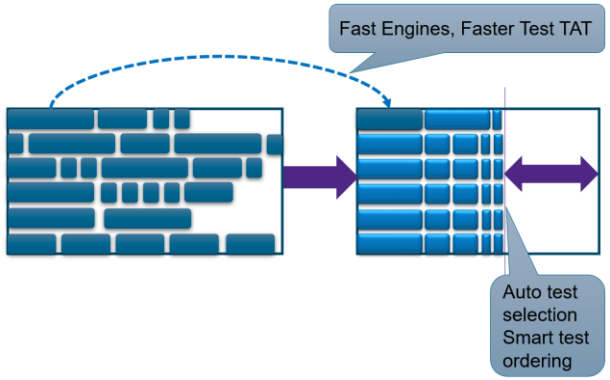
- Test planning, execution & debug, coverage merge and annotation
- Enables verification data-over-time to be mined for analytics

Planner



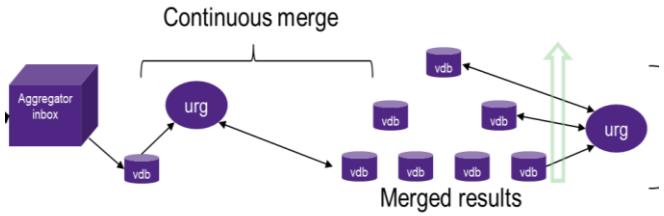
- Multi-user test scheduling/planning
- Supports change history and restore
- API for automated report generation and updates

Runner



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

Coverage



24/7 Merge

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

Refreshed Verdi GUI and IDE

New Verdi GUI Highlights and Key Benefits

Modern design style for comfortable view

- 4 color themes and better color system
- Flat icon design
- Consistent fonts
- Provide Bright/ Dark/ Classic modes

Intuitive tools access

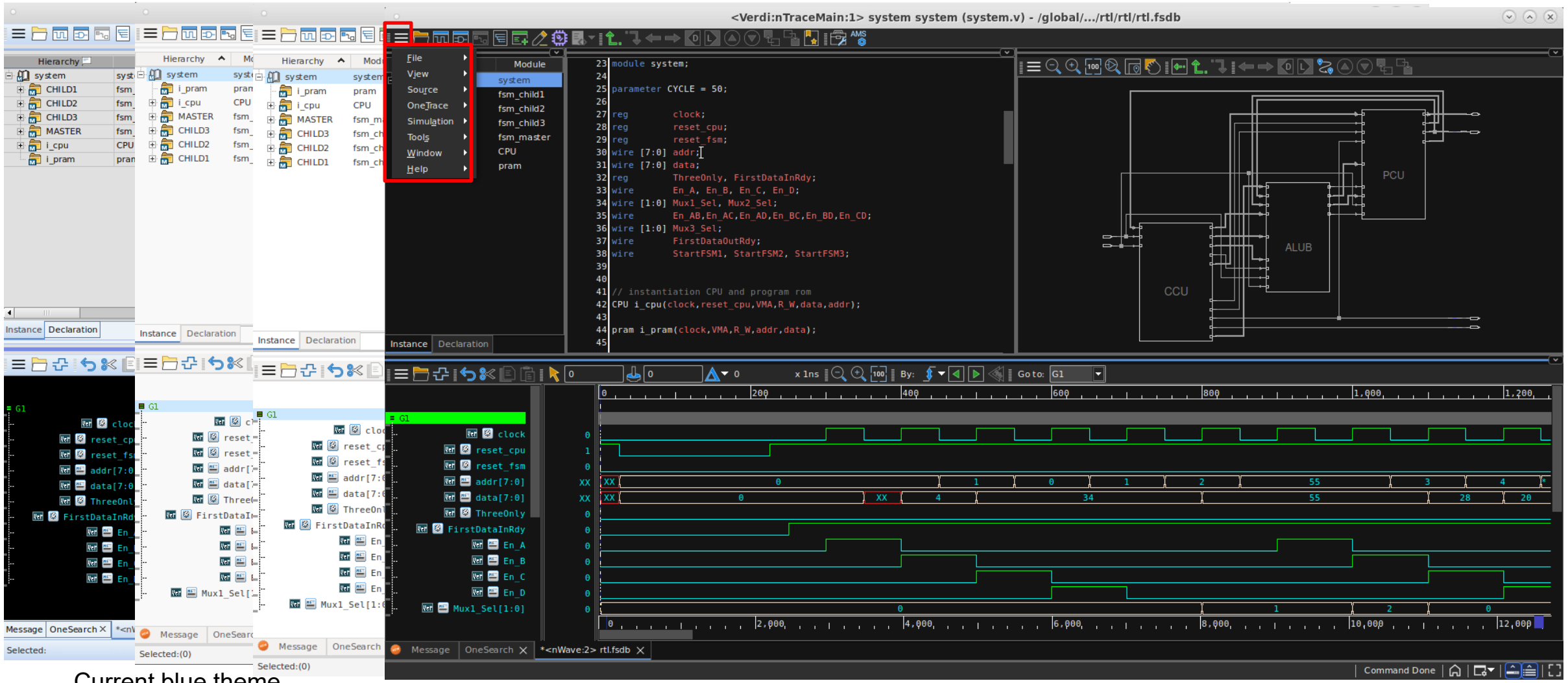
- Hamburger menu
- App launcher
- Maximized debug spacing with Auto ‘hide’

Updated search and find

- Efficiently find targets with new search pane
- Unified search for string/signal/command
- Dedicated floating find bar in each window

Don’t worry! - Menu commands remain in the same location

Classic Mode Color Theme

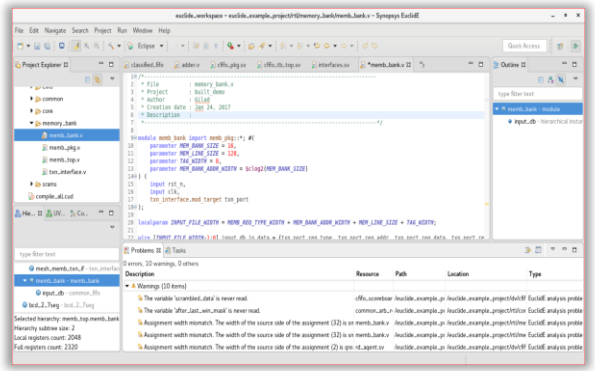


Current blue theme

Classic mode (default) Natural mode Bright mode Dark Mode and Hamburger menu

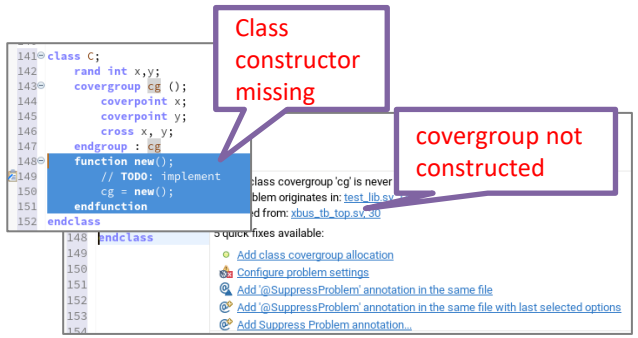
Euclide - Design & Verification Entry + Real-Time Linting

IDE



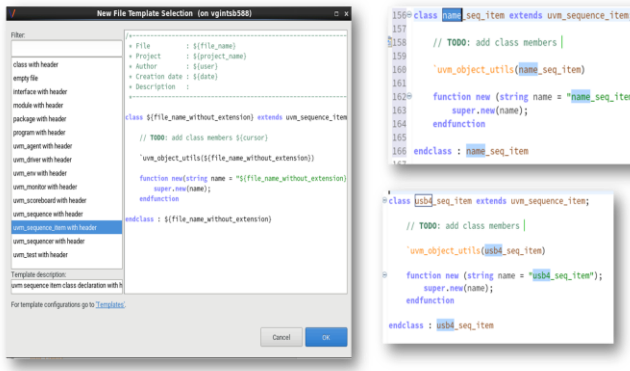
- Euclide-based design environment
- Customizable look (ex dark mode)
- Supports design and testbench creation

Testbench Lint



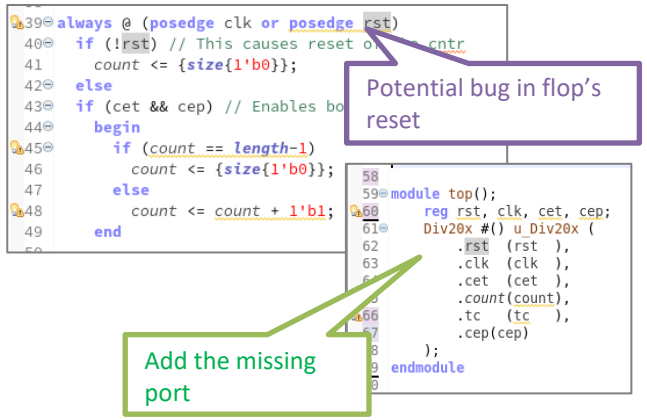
- Checks on-the-fly
- Supports UVM and SystemVerilog
- Addresses potential sim compile issues

Smart Code Template



- A selection of pre-defined and user-extensible templates for UVM classes
- Once actual class name is given, the generated code adapts automatically

Design Lint



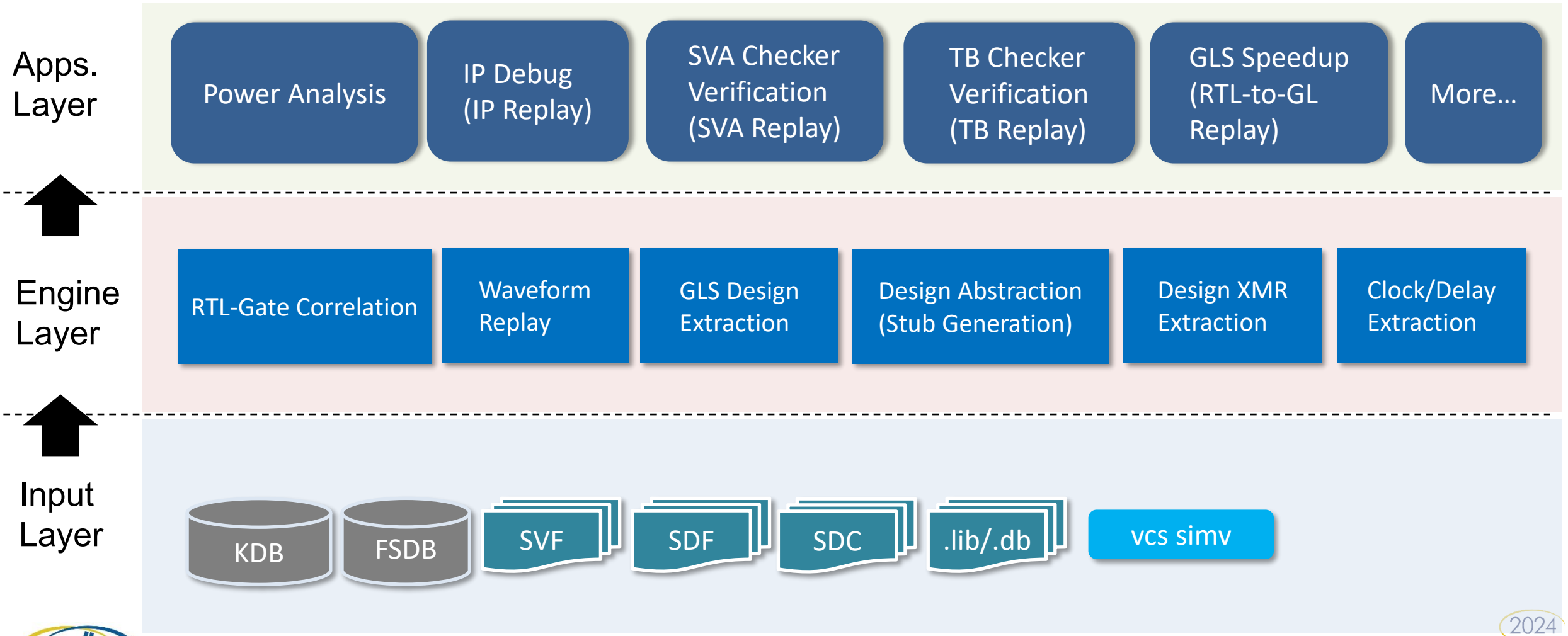
- Checks on-the-fly
- Errors and warnings noted on relevant code
- Rules and severities are easily configurable

accellera SYSTEMS INITIATIVE™

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

VC Replay

Applications Using Efficient Replay Engines



Waveform Replay Concept

LHS : Design Signals

RHS : FSDB Signals

```
TOP.u_dut.u_inst1.cen_reg1.n0 => Testbench.system.top.u_dut.u_inst1.cen[1]
TOP.u_dut.u_inst1.cen_reg2.n0 => Testbench.system.top.u_dut.u_inst1.cen[2]
TOP.u_dut.u_inst1.cen_reg3.n0 => Testbench.system.top.u_dut.u_inst1.cen[3]
TOP.u_dut.u_inst1.cen_reg4.n0 => Testbench.system.top.u_dut.u_inst1.cen[4]
TOP.u_dut.u_inst1.cen_reg5.n0 => Testbench.system.top.u_dut.u_inst1.cen[5]
```

Mapping File

Design

(w/ or w/o testbench)

FSDB

Scope
Begin Time
End Time
...

Configuration

Waveform Replay Engine

For interesting scope only

0

VC Replay Sim

Begin time

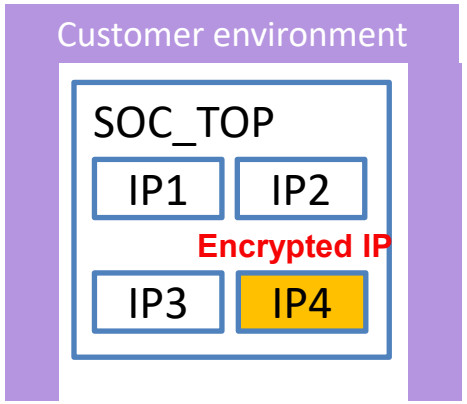
End time

Skip simulating from time zero

VC Replay App - Use Cases and Benefits

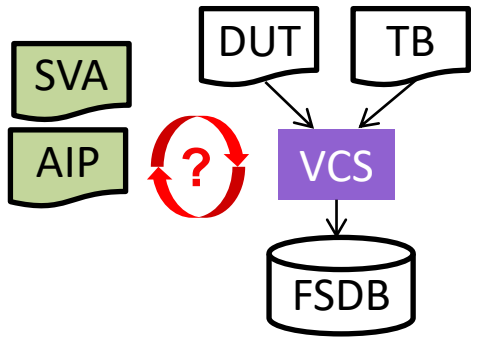
Enables reusing waveforms to avoid continually re-creating and re-running stimulus

Enable Efficient Debug of Encrypted IP



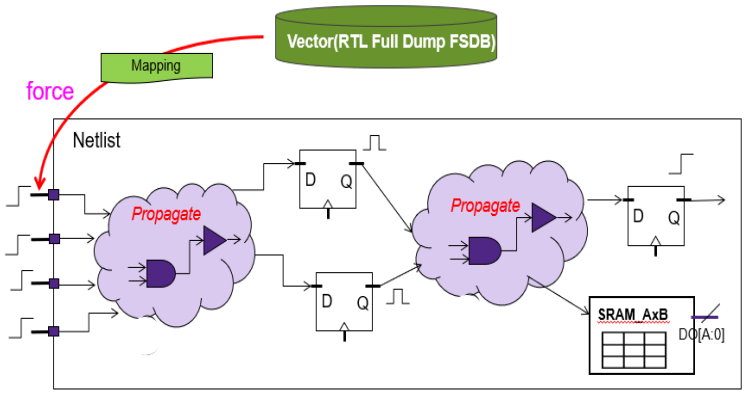
- Deliver reusable FSDB to IP vendor instead of full SOC-level customer environment
- Encrypted IP vendor can quickly and easily debug issues

Accelerate Assertion (SVA/AIP) Validation Flows



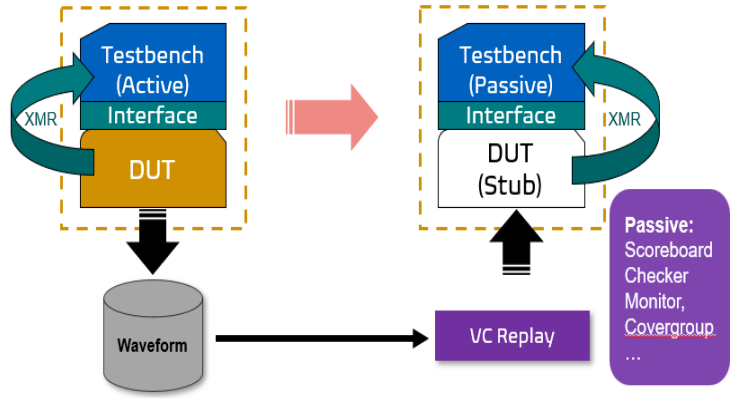
- Incorporate AIP into existing simulation FSDBs
- Decouples AIP development from simulation development
- Enables rapid AIP validation flows

Netlist Verification From RTL



- Run waveform replay at gate-level for functional verification
- Given the RTL SoC simulation, run many GLS IPs in parallel

Passive TB/Checker Debug



- Replace DUT with a Stub
- Run Replay simulation on the TB by driving the interface
- Quick TAT on testing the TB checker

accellera SYSTEMS INITIATIVE™

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Demo Videos

DDT Demo

XRCA Demo

Summary

Summary

- Manual regression debug is tedious, but you can automate it with AI and advanced RCA technologies to debug any failing simulations in **Next-Gen Verdi.**
- Regression binning classifies many failed tests into a few bins of with similar errors.
- Bug prediction reduces the time spent locating reference snapshots for debug engines.
- RCA helps in debugging the DUT and TB issues faster
- Debug Decision Tree allows the user to train as set of conditions that can then be checked in new debug cycles.
- Use the VMS to create a verification plan, then have it manage your simulation env, and gather all the coverage data for analysis in that plan.
- Create correct code by construction by using Euclide in Verdi's redesigned GUI which will allow for quicker and more accurate code generation for your testbenches and design.
- VC Replay Apps, to expediate the debug process in different scenarios

Q&A

SYNOPSYS®

Thank You

For more information visit www.synopsys.com/debug