Next-Generation Verdi: Overview of New Debug and Verification Management

Myles Glisson
Synopsys

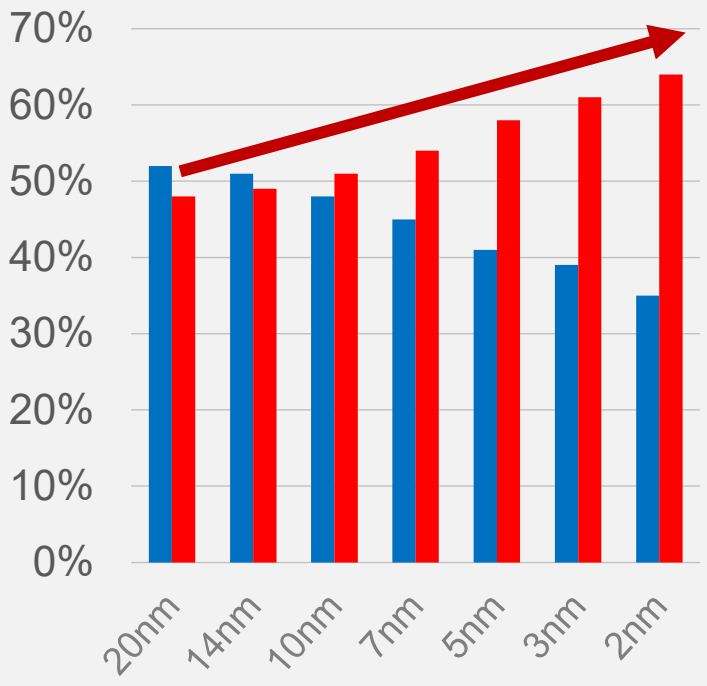
Agenda

- Motivation and Debug Flow
- Synopsys Verdi® Verification Management System
- Refreshed Verdi Graphical User Interface (GUI)
- Verdi Integrated Design Environment (IDE)
- AI-Based Regression Binning
- AI-Based Bug Prediction
- Root Cause Analysis Components
- AI-Based Message Analysis
- Debug Decision Tree
- Summary + Q&A

Motivation and Debug Flow

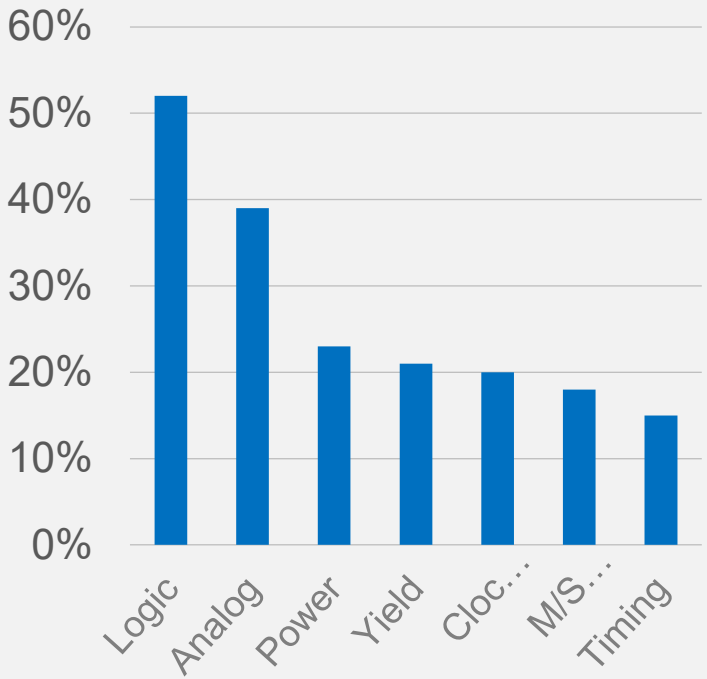
Impact on Right First-Time Silicon

Fewer First-Time Right



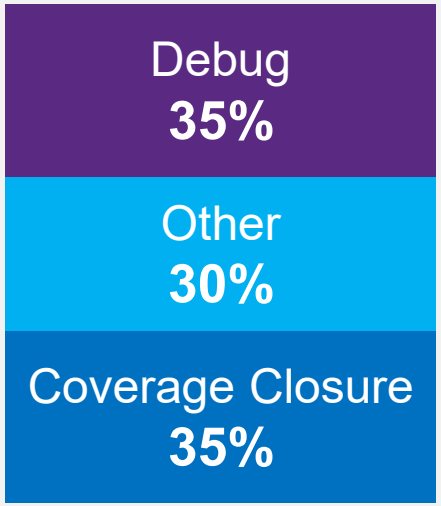
More Respins

Logic Bugs Dominating



Main Cause of Respins

Verification Time Spent

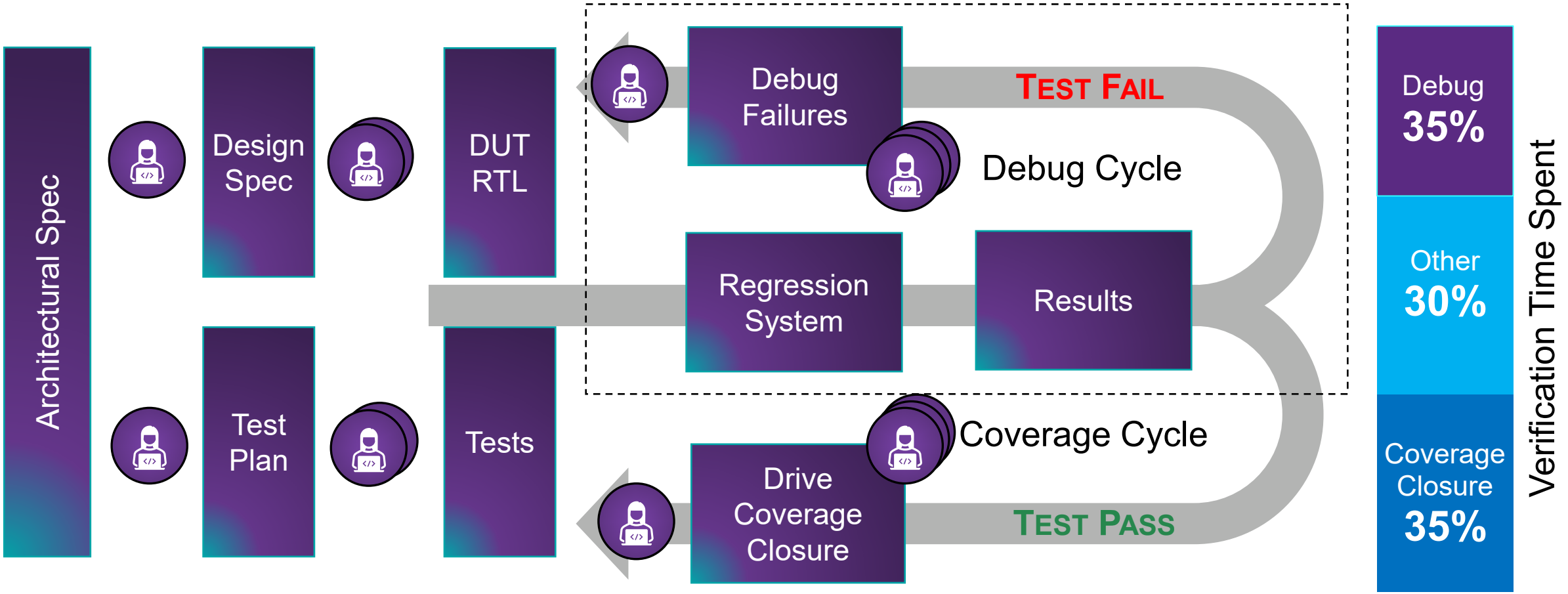


Need to Reduce Debug Time!

Source: Wilson Report 2022

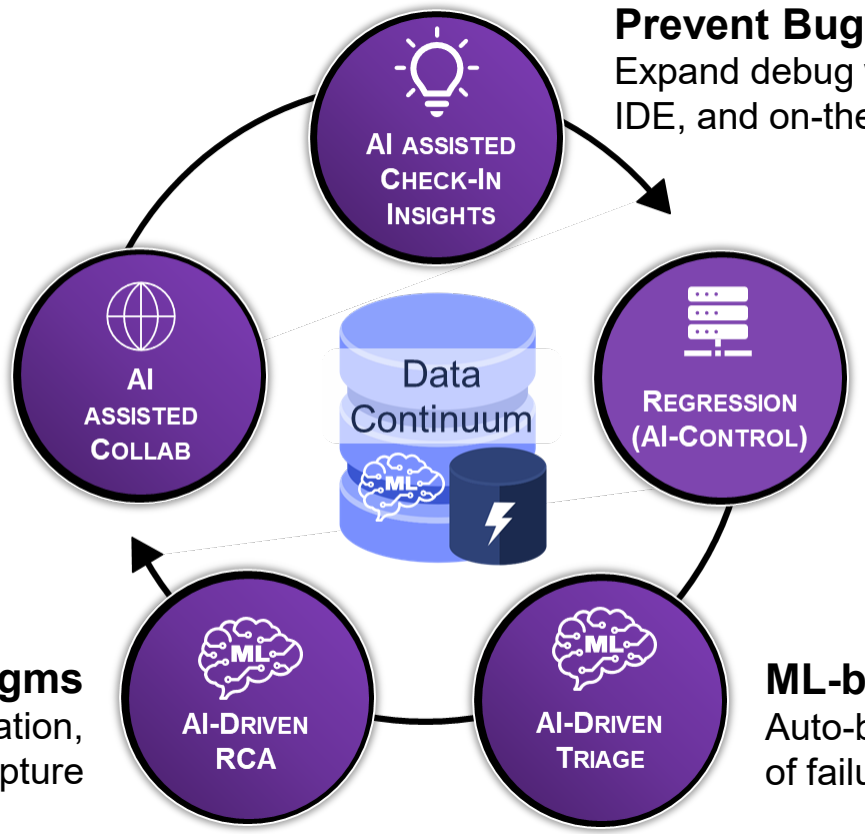
Manual Process

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

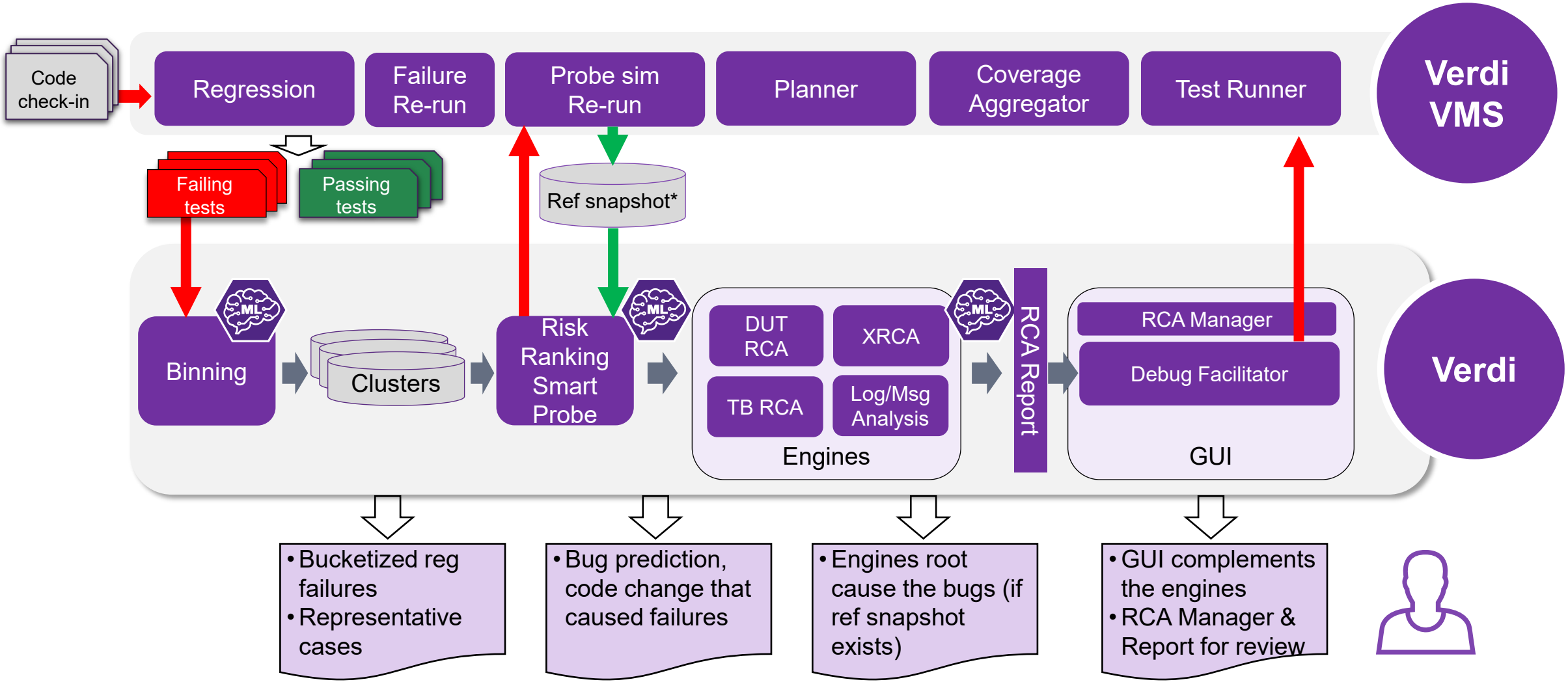
Regression Management
ML-optimized daily regression, reduce HW costs and increase efficiency

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

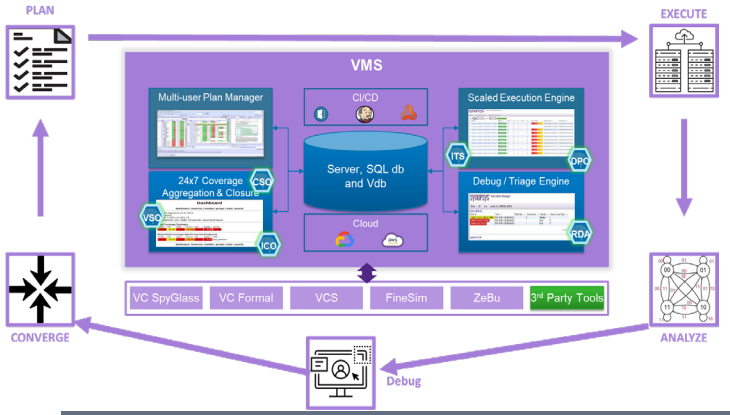
ML-Based, Automated Regression Debug



Verdi Verification Management System

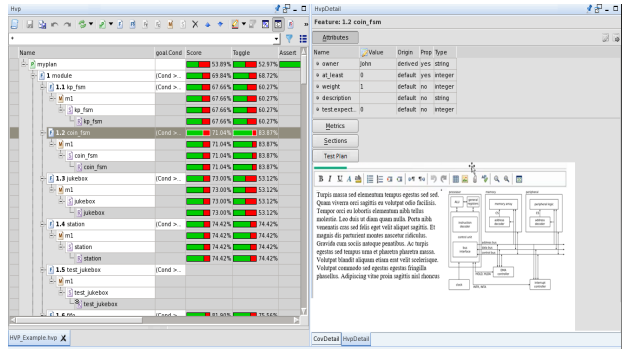
Verification Management System

Manager and Dashboard



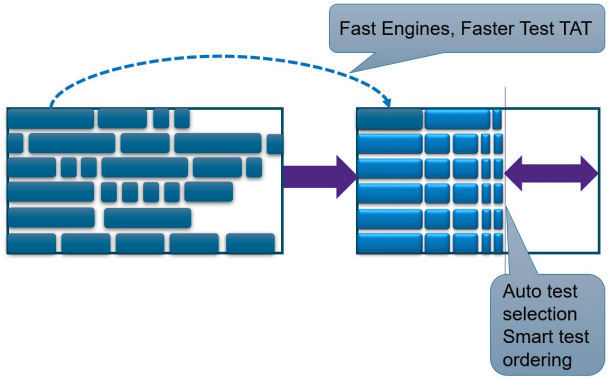
- Test planning, execution & debug, coverage merge and annotation
- Enables verification data-over-time to be mined for analytics

Planner



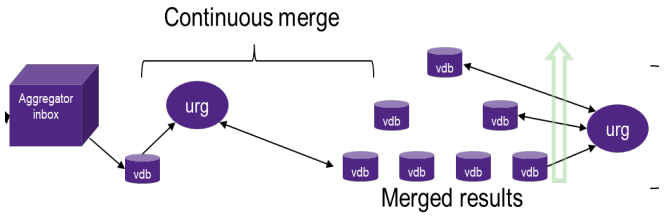
- Multi-user test scheduling/planning
- Supports change history and restore
- API for automated report generation and updates

Runner



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

Coverage



24/7 Merge

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

Refreshed Verdi Graphical User Interface (GUI)

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

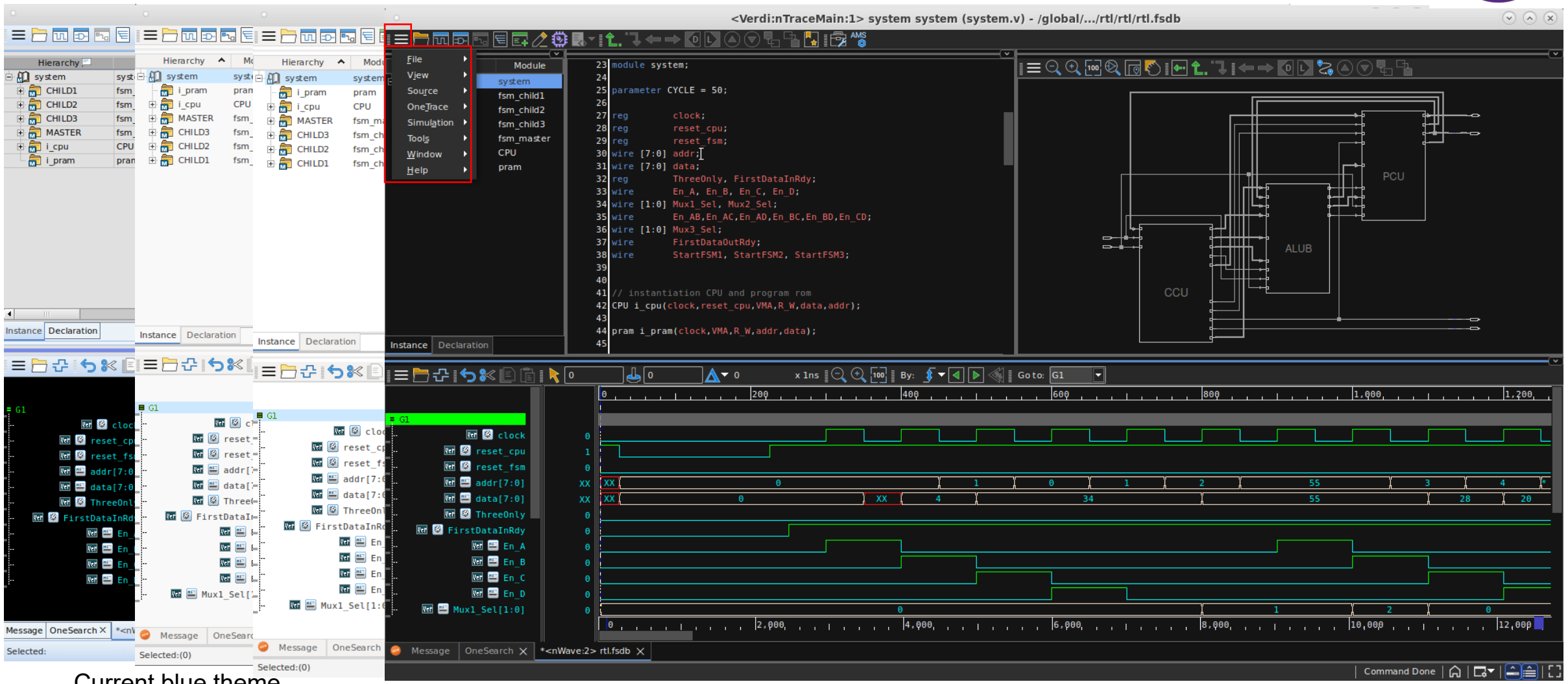
- Hamburger menu
- App launcher
- Maximized debug spacing with Auto ‘hide’

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



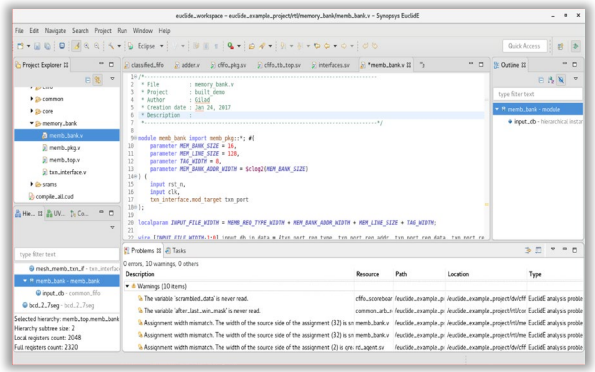
Current blue theme

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

Verdi Integrated Design Environment (IDE)

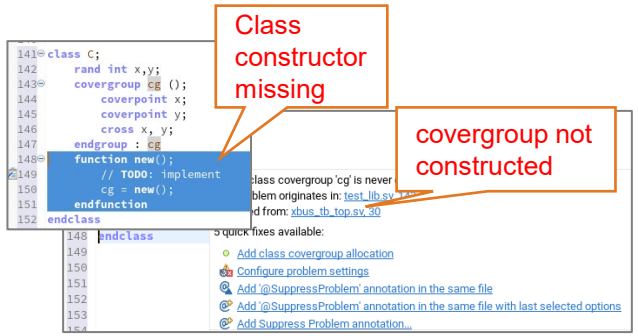
Euclide-Design & Verification Entry & Real-Time Linting

IDE



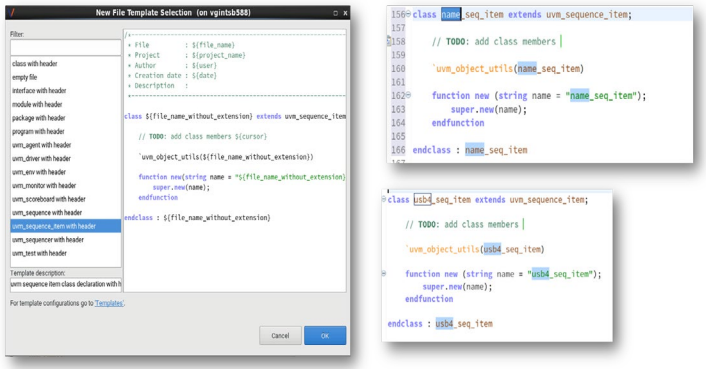
- Synopsys Euclide-based design environment
- Customizable look (ex dark mode)
- Supports design and testbench creation

Testbench Lint



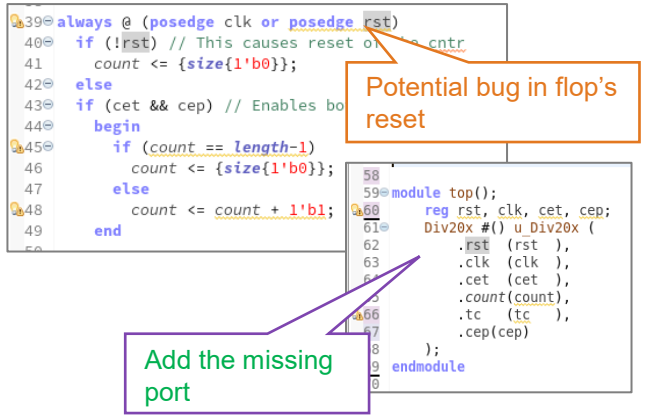
- Checks on-the-fly
- Supports UVM and SystemVerilog
- Addresses potential sim compile issues

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

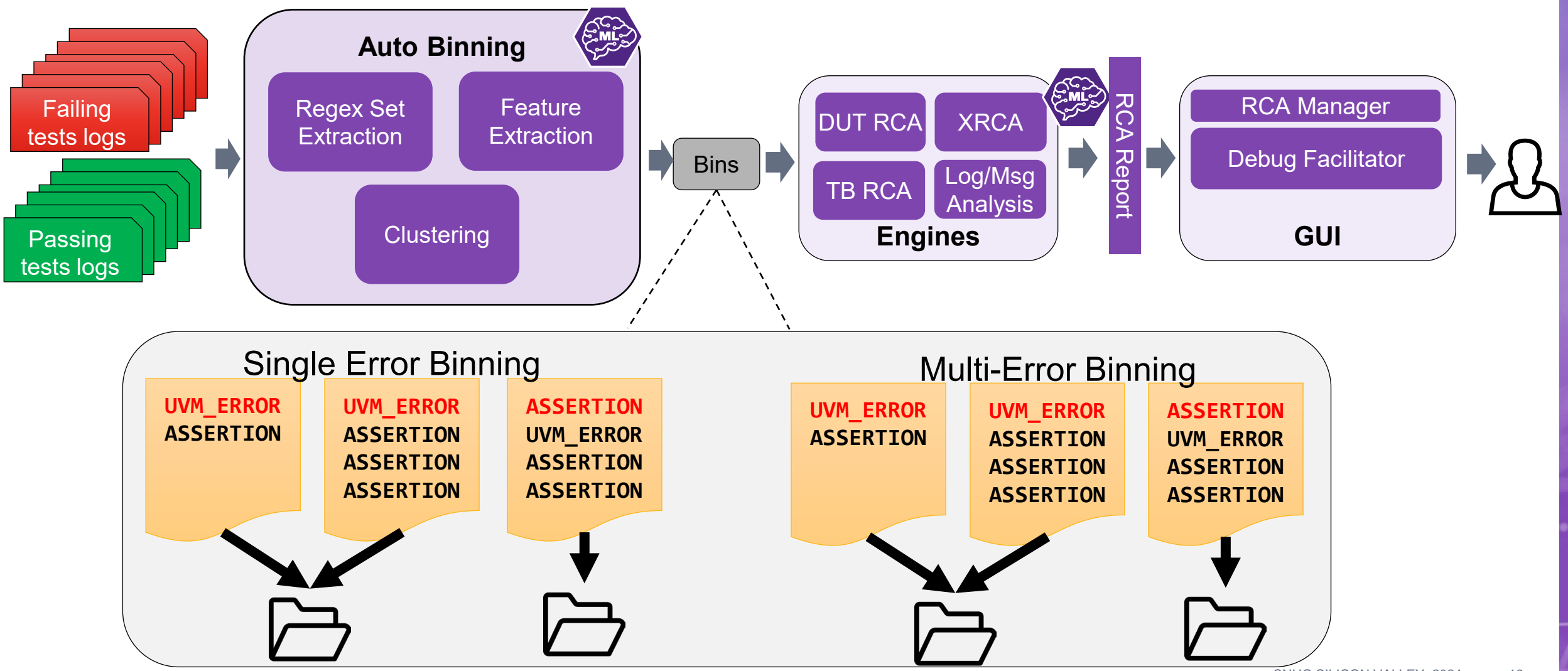
Design Lint



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

AI-Based Regression Binning

Regression Binning with ML

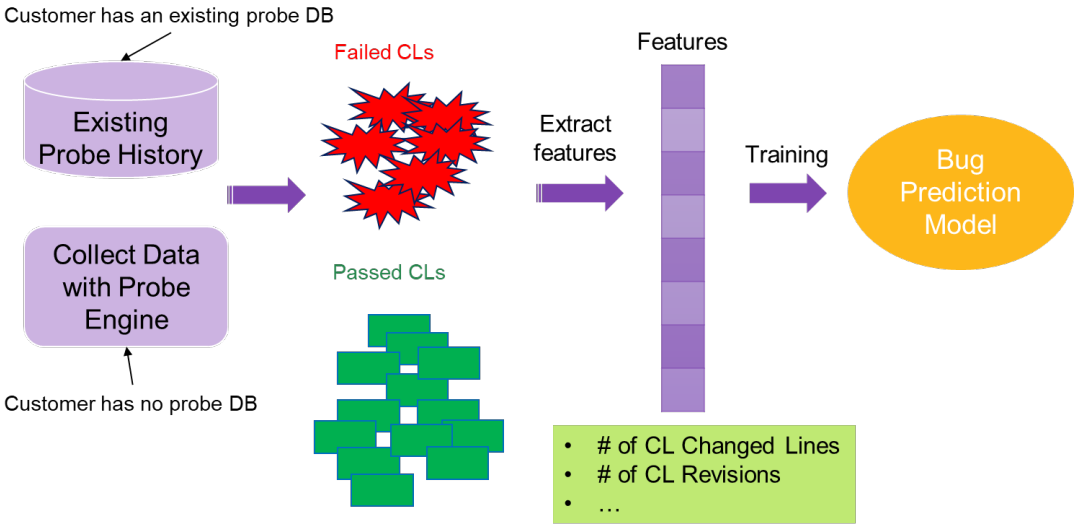


AI-Based Bug Prediction

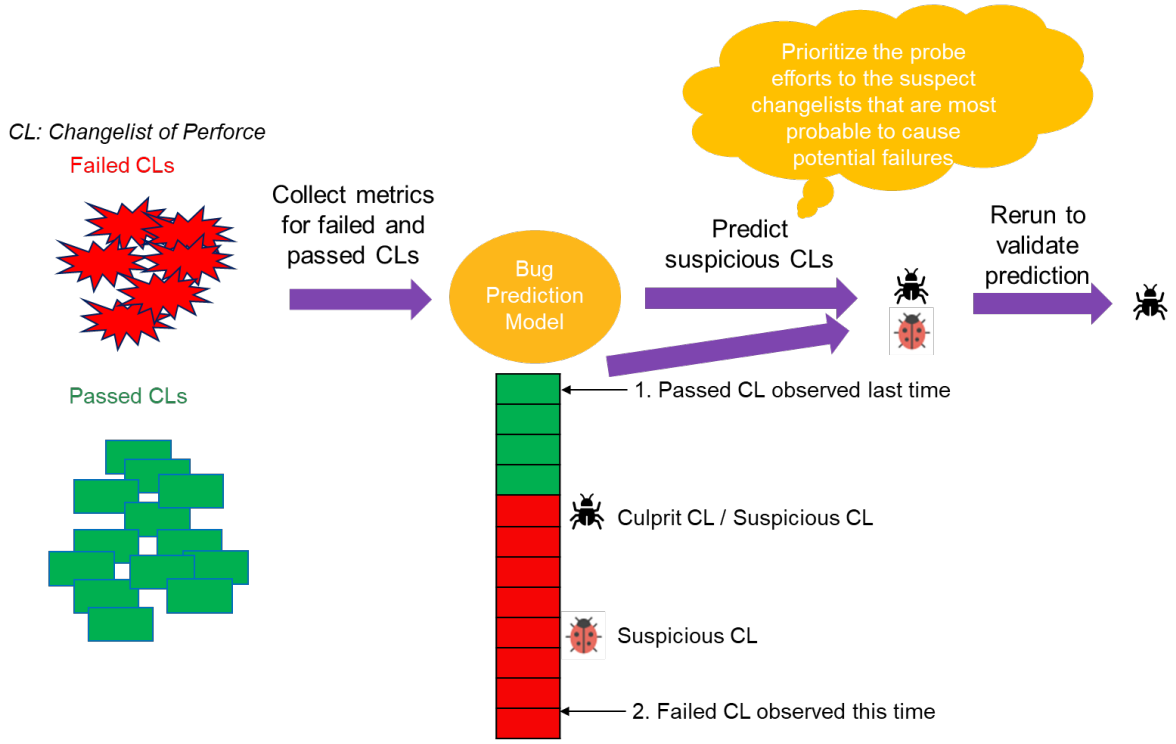
Smart Probe

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

Train the bug prediction model



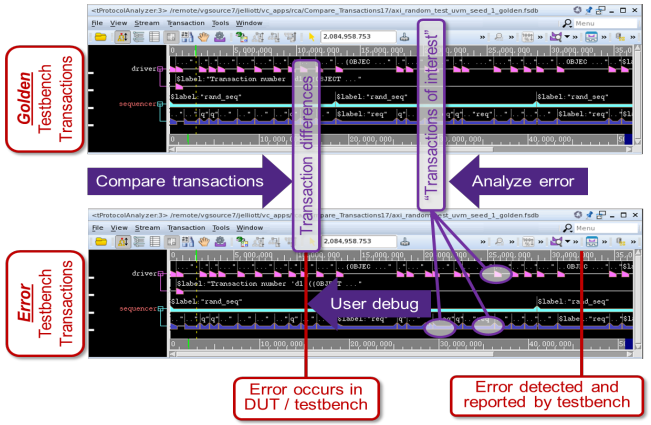
Bug prediction use model



Root Cause Analysis Components

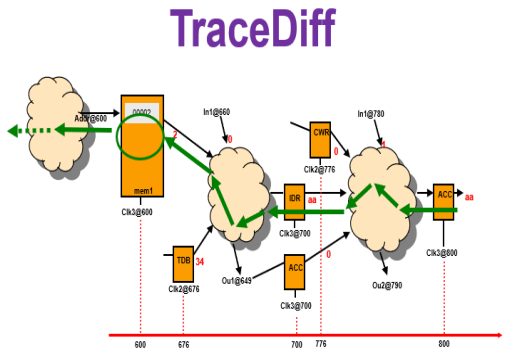
RDA Component Technology

TBRCA



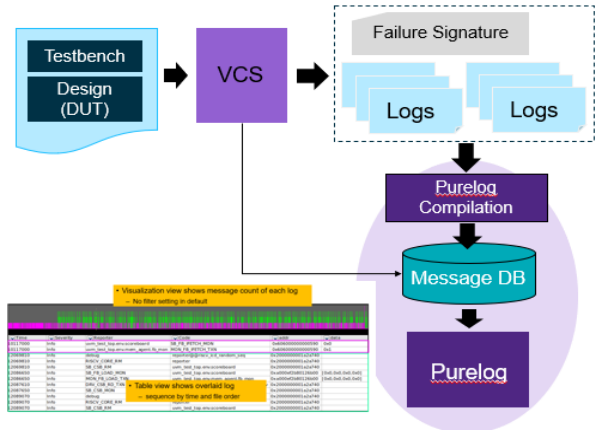
- Transaction Diff – Diff the transaction in the reference vs failing FSDB
- Message Analysis – No ref FSDB required. Analyze info from the “error” message.
- Report transaction of interest linked to the error

DUTRCA



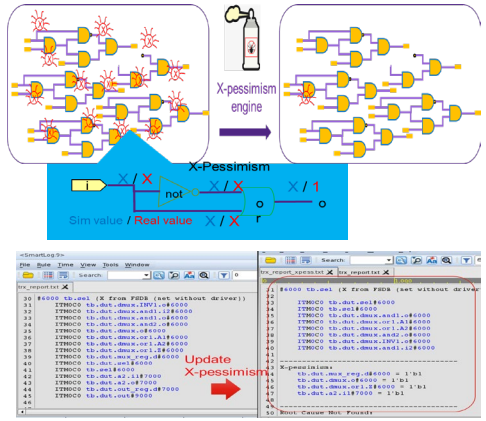
- Adopt roll back mechanism and TraceDiff technology to narrow down DUT problem
- Temporal Flow View to analyze root cause path

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

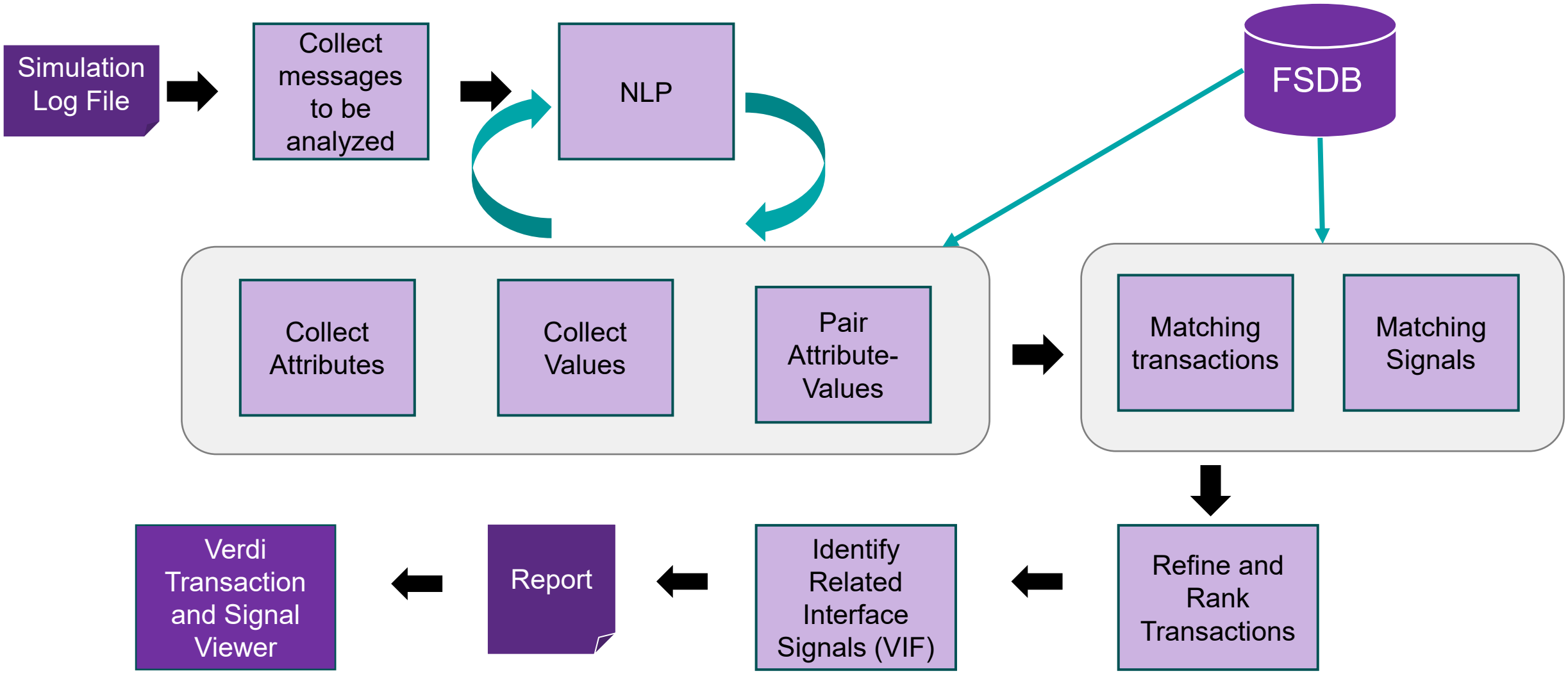
XRCA / with X-Pessimism



- 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

AI-Based Message Analysis

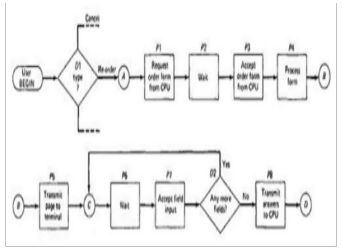
Message Analysis – High Level Flow



Debug Decision Tree

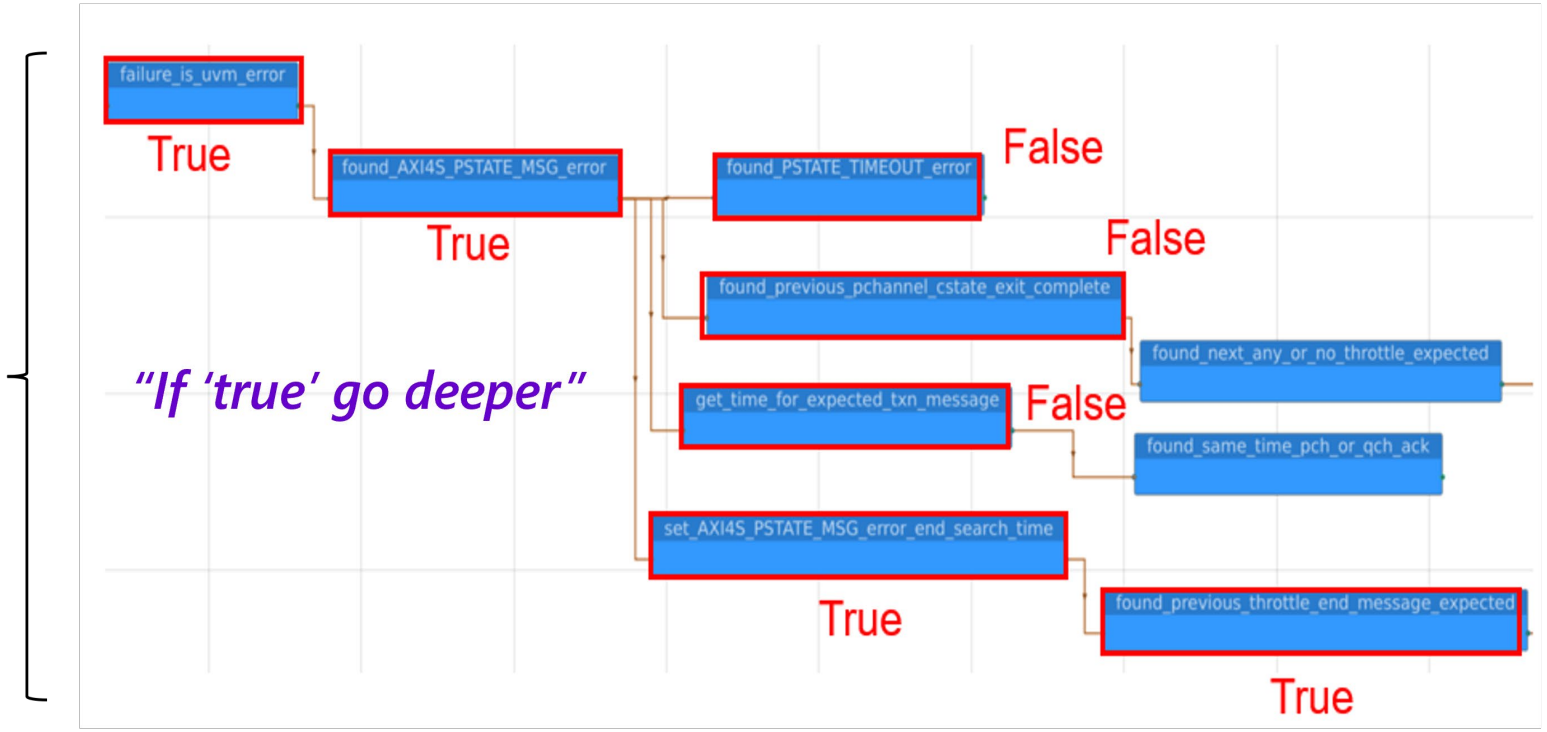
Debug Decision Tree (DDT)

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



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**.
- 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.
- 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.
- Message analysis identifies transactions and VIFs that are related to the error message.
- Debug Decision Tree allows the user to train as set of conditions that can then be checked in new debug cycles.


THANK YOU
YOUR
INNOVATION
YOUR
COMMUNITY