



SANTA CLARA, CA, USA
MARCH 2 - 5, 2026

A Platform-Based Approach to Reduce Verification Turn-Around Time in Memory Designs

Junho An, Seonyong Lee, Minseon Kim,
Gyehyun Na, Dongeun Lee

Memory Design Technology, Samsung

SAMSUNG

Jaehyun Park, Kiuk Kim

Synopsys Korea

SYNOPSYS®

A Unified Platform to Reduce Debug TAT

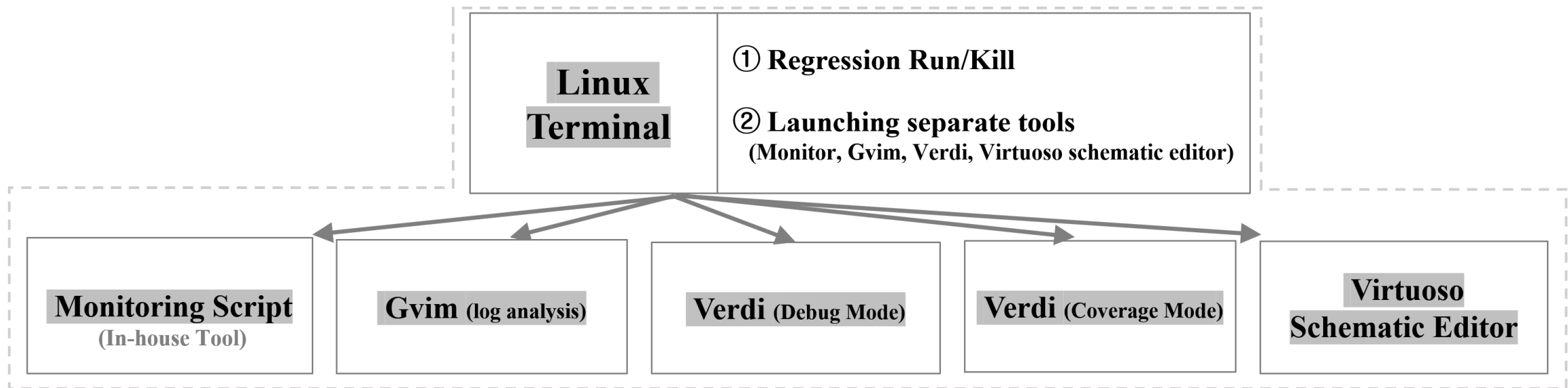
- A platform for large-scale memory regression debug
- Automated bucketing + Log $\leftrightarrow$ wave correlation + Incremental debug
- Result: 3 $\rightarrow$ 2 regressions, TAT 24.6 $\rightarrow$ 16.4 days, manual 24.6 $\rightarrow$ 12.3 days

Motivation

- Debug/triage dominates TAT in large memory regressions
- Manual monitoring delays time to first waveform (FSDB)
- Need a reproducible, platform-level debug workflow

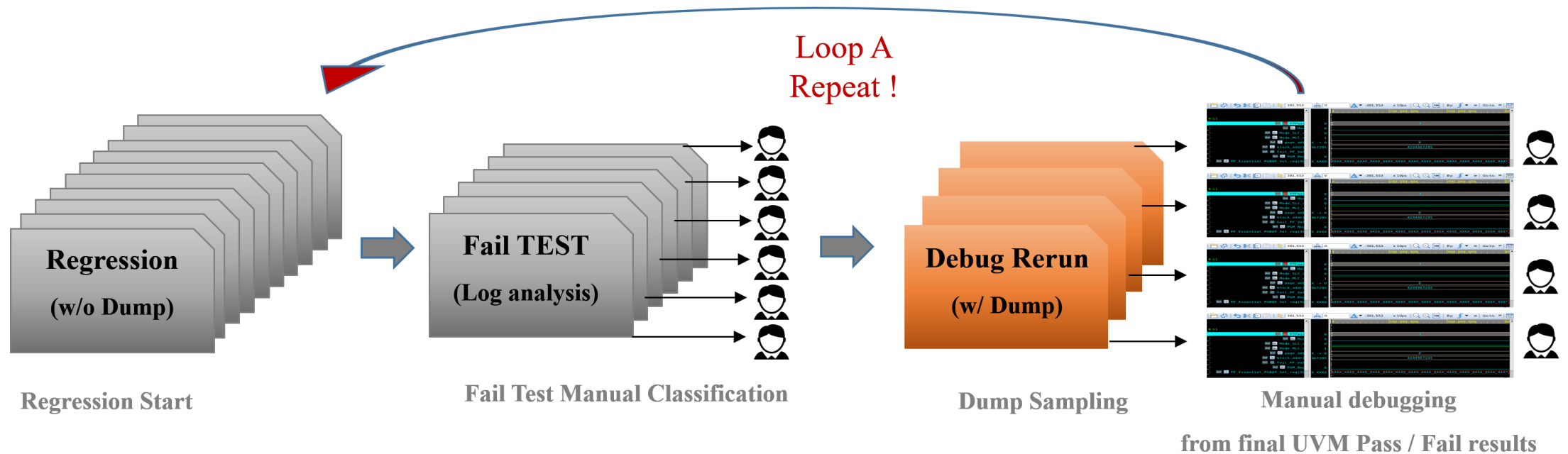
Conventional Pain

- Tool/script fragmentation across
Monitor → Log → Rerun → Wave → Coverage



Conventional Pain

- Manual fail triage and manual debug reruns
- Repeated regression iterations (Loop-A)



Goal & Contributions

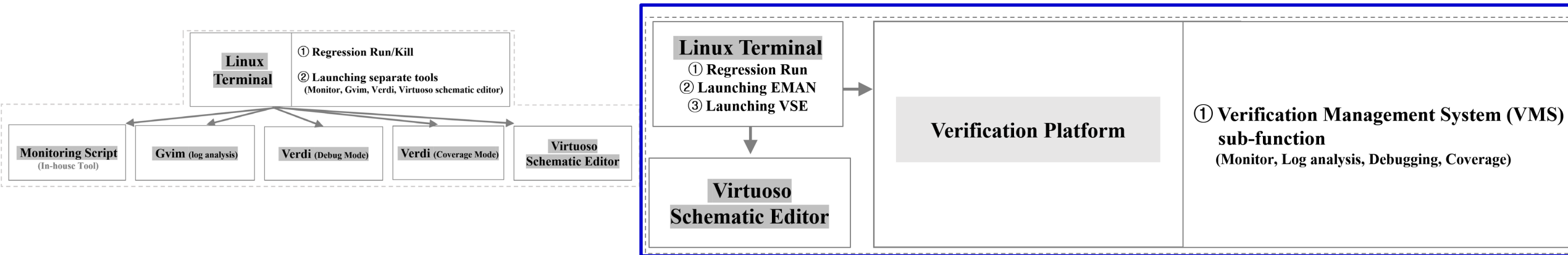
- Goal: Reduce debug TAT by minimizing manual triage/reruns
- Contributions:
 - A) Unified platform
 - B) Auto bucketing
 - C) Log ↔ Wave
 - D) Incremental debug
 - E) GUI customization
- Reusable framework on commercial EDA tools

Proposed Platform Overview

- Single integrated verification platform:
Monitor → Bucket(triage) → Rerun → Waveform → Report
- Buckets drive representative-case debugging
- Automation reduces repeated human-in-the-loop steps

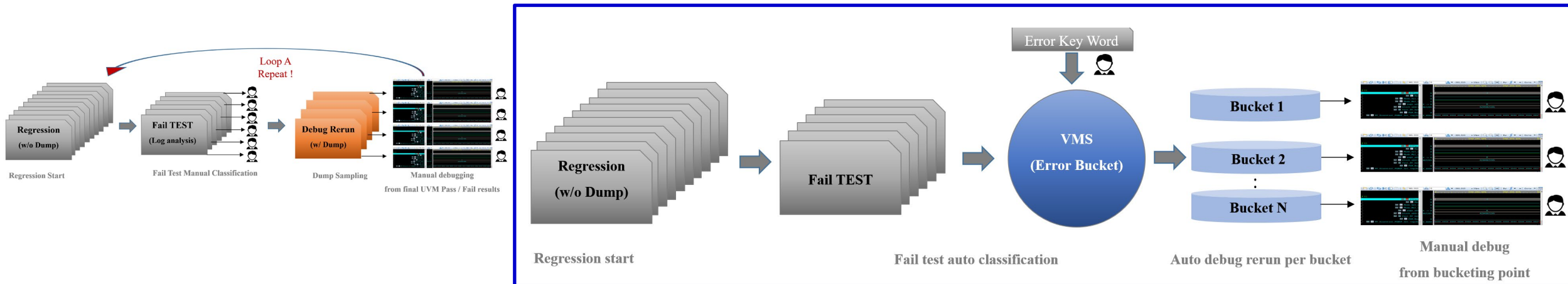
A. Unified Platform

- Consolidate 5 verification functions into one platform
- Eliminate tool switching overhead
- Improve maintainability vs. ad-hoc internal scripts



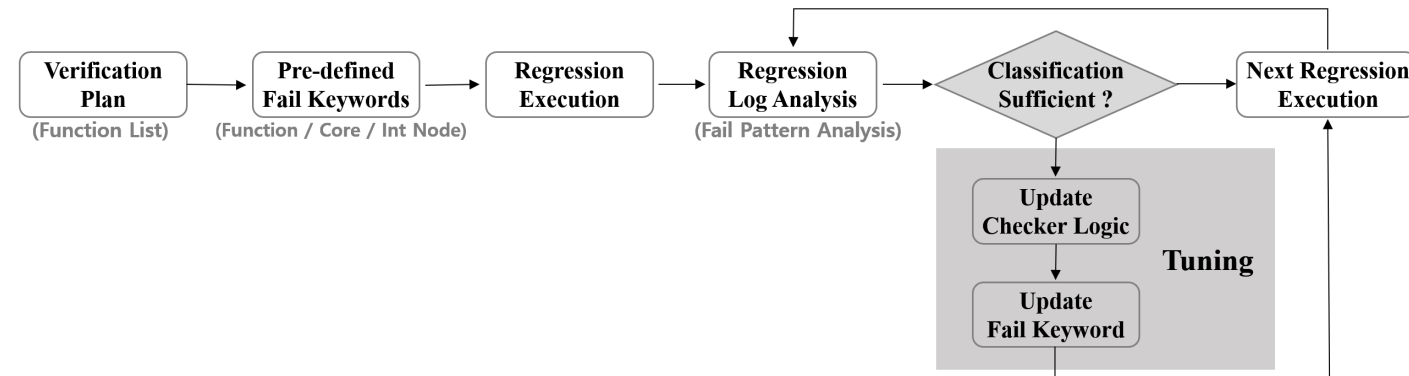
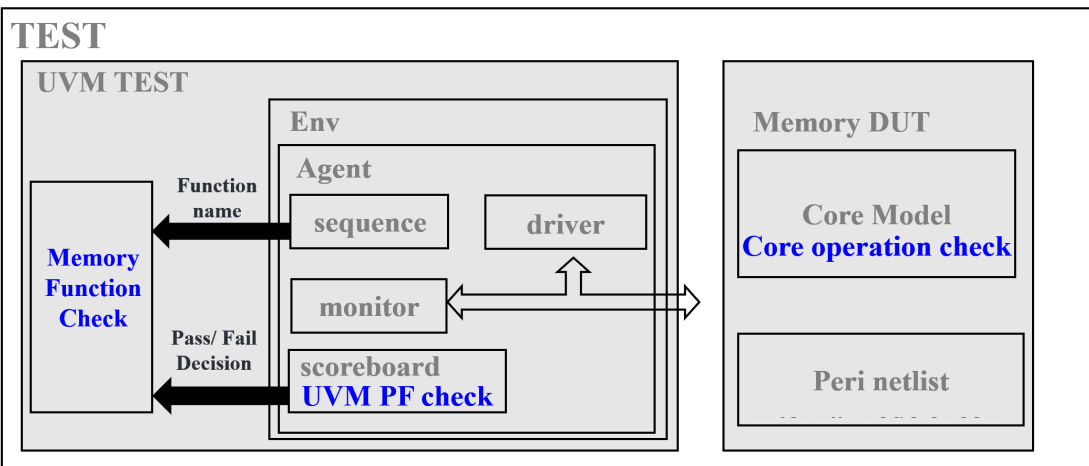
B-1. Auto Bucketing

- Predefined fail keywords → automatic bucket generation
- Representative testbench rerun per bucket (FSDB on demand)
- Remove repeated Loop-A regressions



B-2. Modularized Checker

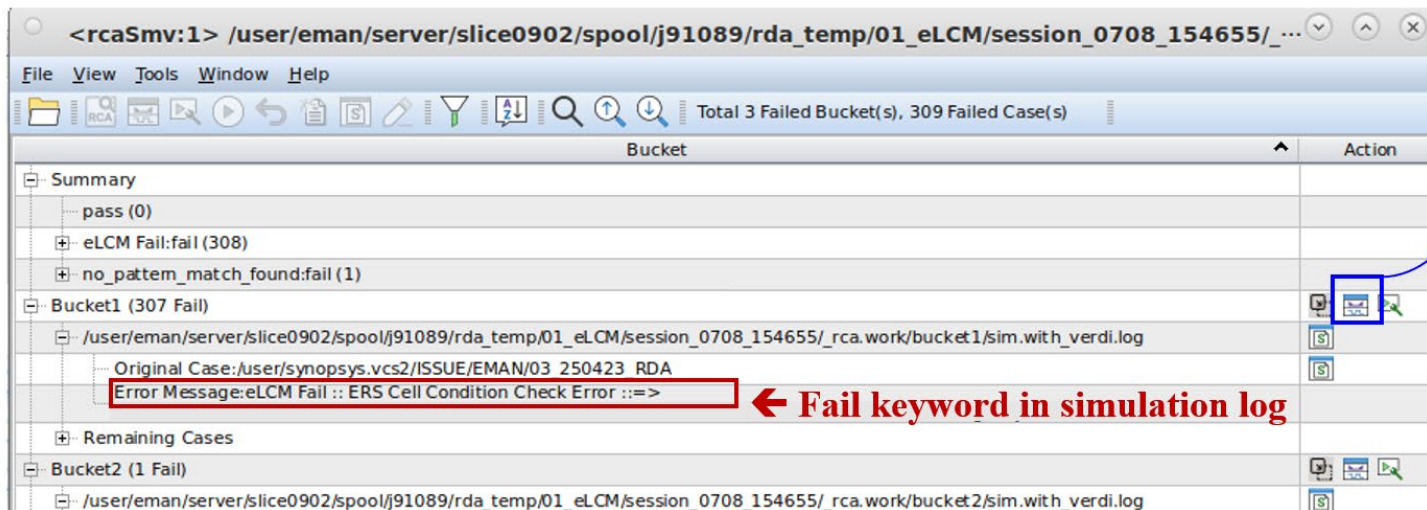
- 3 sources: UVM Pass/Fail, Function Checker, Core Op. Checker
- Logs include function/keyword + instance location
- Faster triage and shorter debug navigation



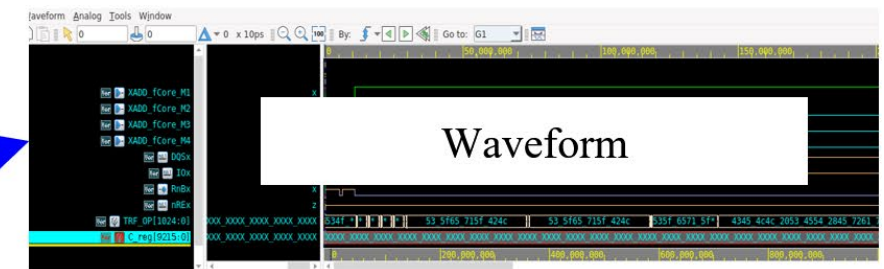
C. Log → Wave Correlation

- Parse log: keyword + instance + time window
- Open waveform at failure point with curated signal lists
- Effective for deep memory hierarchy (high-density)

RDA report



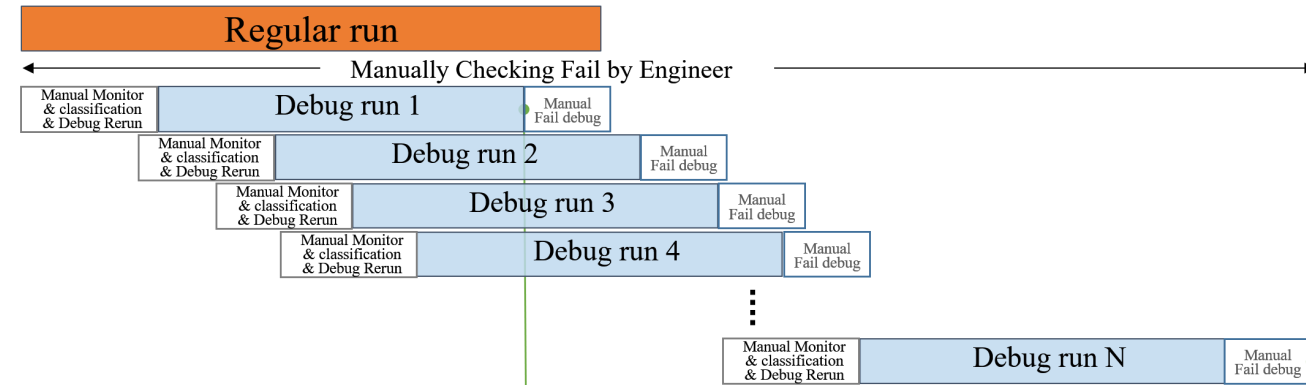
← Fail keyword in simulation log



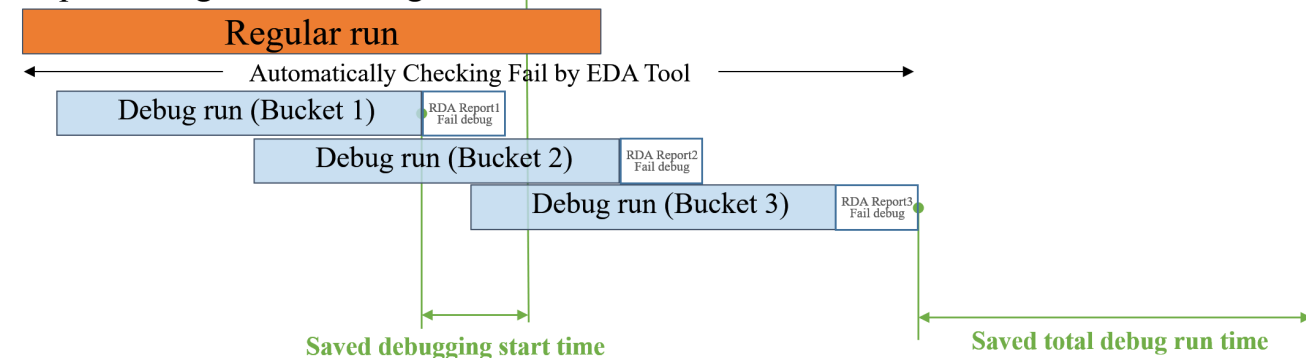
D. Incremental Debugging

- Don't wait for full regression completion
- Fail detected → bucketed → representative rerun launched immediately
- Reduce time-to-signal and fix/verify loop time

< Conventional Regression Debug >

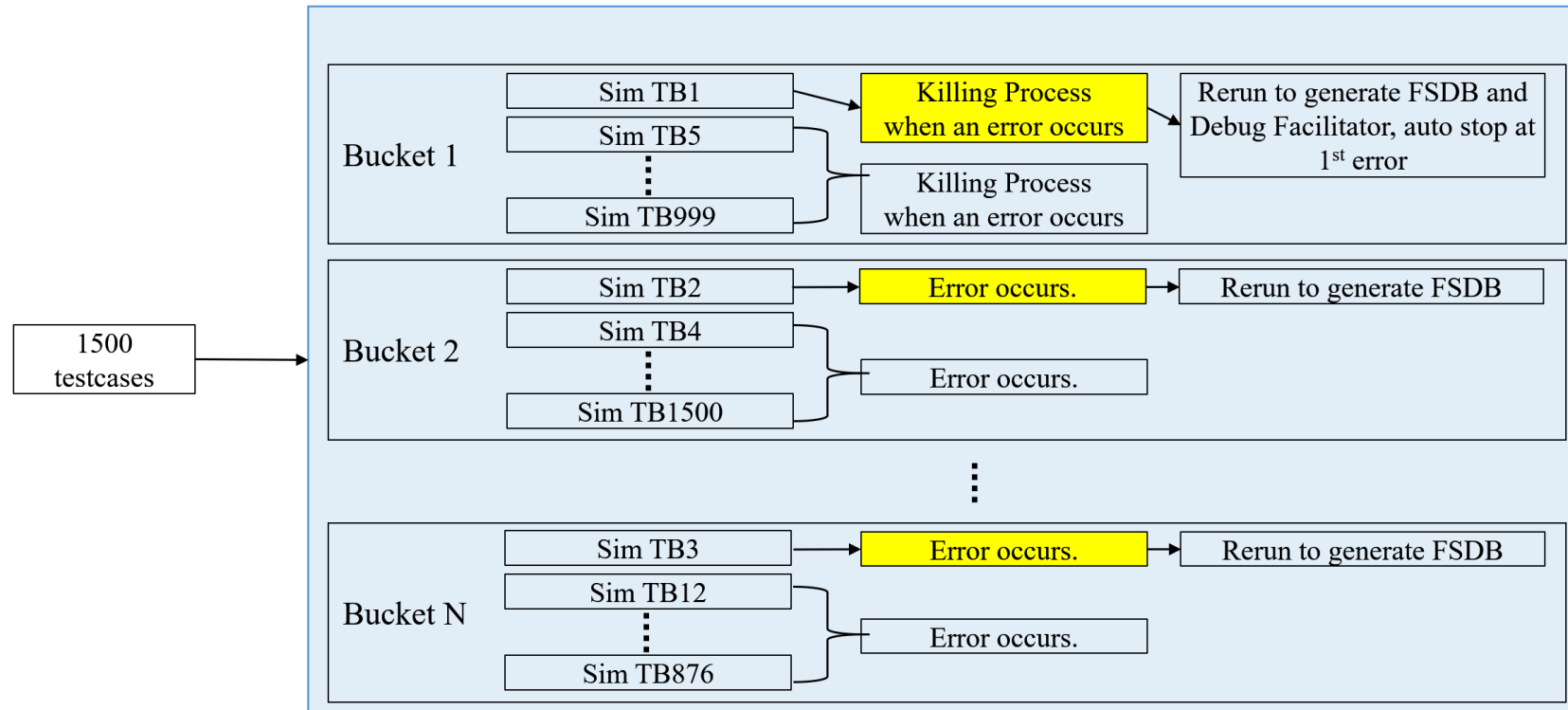


< Proposed Regression Debug >



D. Incremental Debugging

- Incremental binning



E. GUI Customization

- Custom status beyond Pass/Fail (e.g., stuck, runtime fail)
- One-click RDA report + capacity visibility
- Reduce triage-to-debug friction in daily ops

Home Proj_250807232501_1925 show_runtime_status session_kerwin_odclegacy0130

Session Attributes |

☐ Show Filter Fields Filters: --

Select Layout: runtime_status_layout ☐ Quick Filters

Trend Chart Select All Deselect All ☐ Bulk Edit Default view: ☒ Show: standard

☐	Name ▲	Owner ▲	Status ▲	Runtime Status	Stop Rule ▲	Smoke ▲	RTime(s)
☐	should_runtime_fail	—	Running	RunTimeFail	—	—	—
☐	should_stuck	—	Running	Stuck	—	—	—

Original status column

Add a custom-column to show “Runtime Status”

>> Home rda_demo_eman_jaehyunp a.b.c.alu_uvm

User: jaehyunp Rows: 10 View: flat Compare Sessions

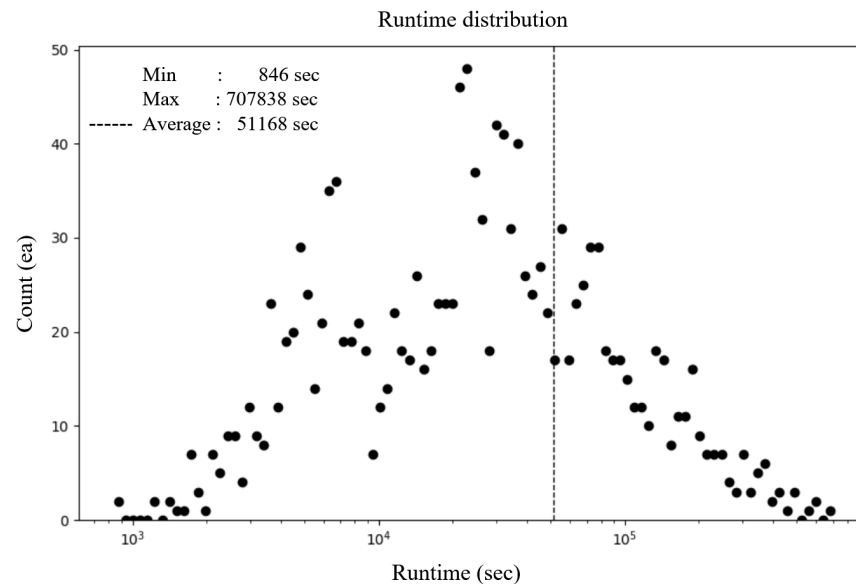
☐ Show Filter Fields Filters: --

Open RDA report Total: 28T, Used: 28T, Avail: 311G (/remote/gts_corpac) 99%

Customize...

Experimental Setup

- Workload: **1,500 regression testbenches**
- Runtime distribution : 1 regression = max 8.2 days
- Compare conventional vs. proposed under same workload



Conventional Results

- Manual log inspection and manual rerun queueing
- Multiple iterations needed to cover remaining fail classes
- **Regression count: 3 (Debug Rerun 2)**

< Testbench (TB) 1500 ea / Fail TB 100 ea / Bug Fix 4ea >

< TB 100 ea / Fail TB 40 ea / Bug Fix 4ea >

< TB 40 ea / Fail TB 20 ea / Bug Fix 2ea >

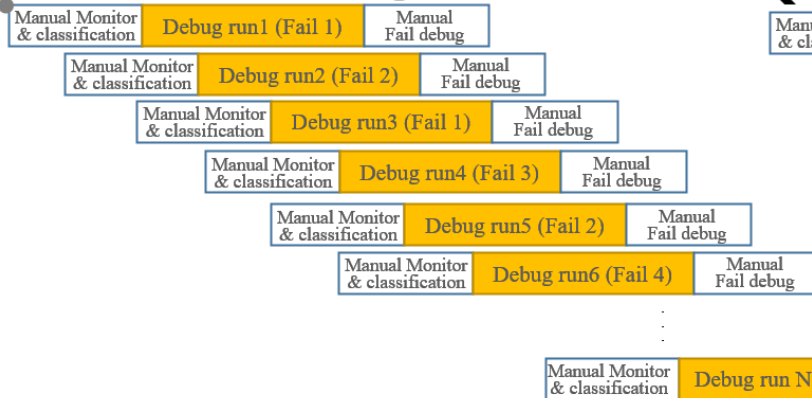
1st Regression run (without dump)

2nd Regression run (without dump)

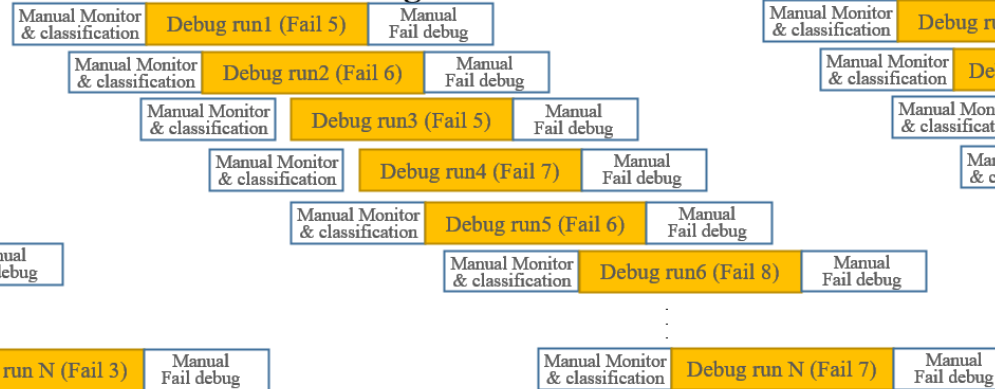
3rd Regression run (without dump)

Engineer manual debugging period

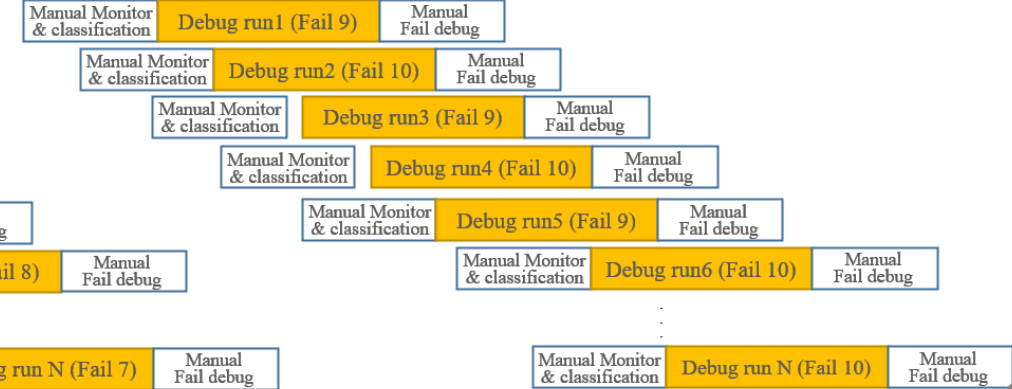
← 1st Fail Manual Debug



← 2nd Fail Manual Debug

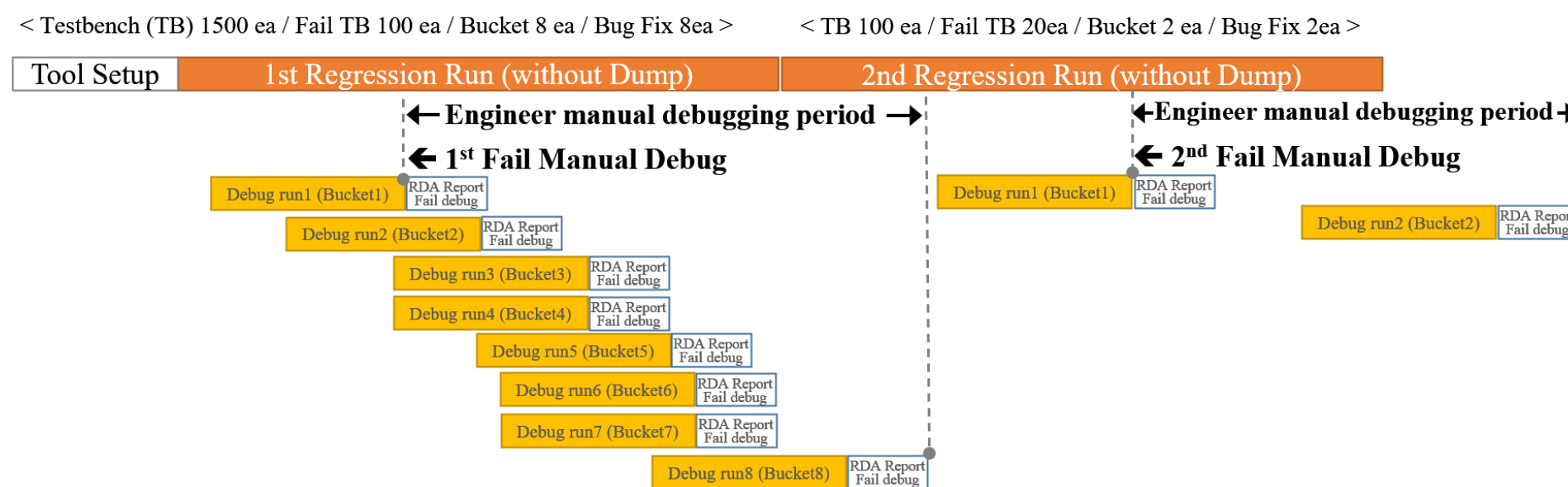


← 3rd Fail Manual Debug



Proposed Results

- Auto-generated buckets during execution
- Representative reruns + early waveform availability
- Regression count: 2 (Debug Rerun 1)



Category	Conventional	Proposed
UVM Pass/Fail buckets	1	1
Function buckets	-	136
Core operation buckets	-	20
Total buckets	1	157

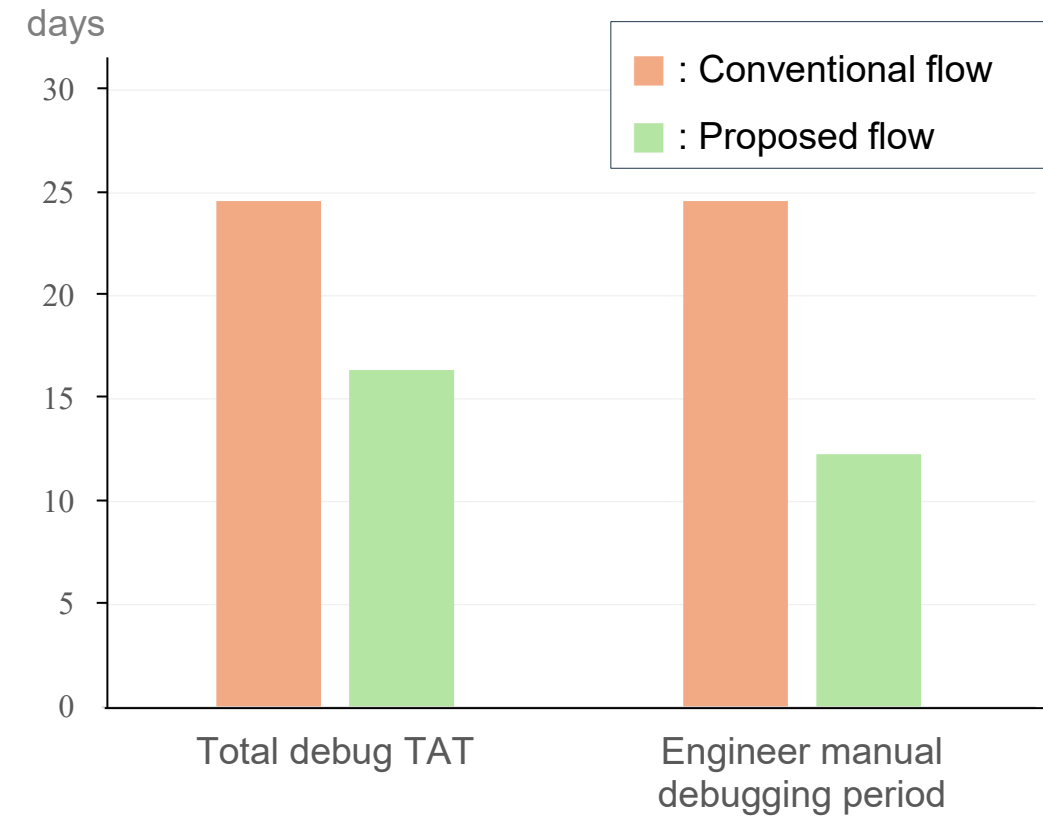
Key KPI Table

Comparison of Verification Flow
in terms of normalized turnaround time and efficiency

	Conventional flow	Proposed flow	Improvement
Regression count	3 turns	2 turns	Reduced failure iterations
Total debug TAT	100% (Max 24.6 days)	~ ^a 66% (~16.4 days)	~1.5x faster
Engineer manual debugging period	100%	~ ^a 50%	Significant reduction

a. Estimated results based on internal pilot project;
actual values may vary depending on design complexity and regression configuration.

Regression and Debug TAT Comparison



Conclusion & Reuse

- Platform-level Automation:
Bucketing + Log $\leftrightarrow$ wave + Incremental reruns
- Quantitative TAT and manual-effort reduction demonstrated
- Extensible to other large regression domains (DRAM, Controller IP)