init

Cognitive Smoke Testing

Google

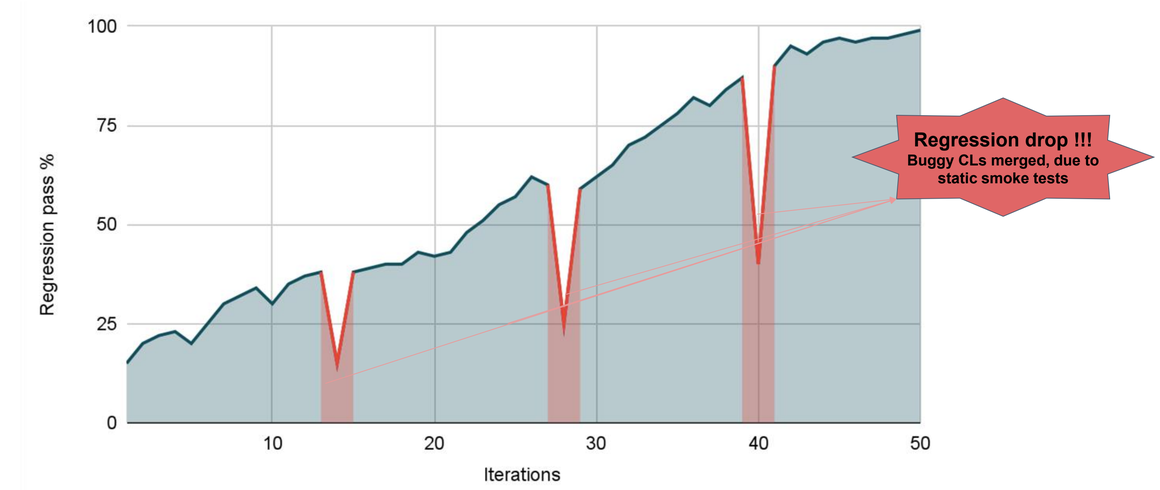
Nikhil Singla, Rohit Jindal, Pandithurai Sangaiyah

Debarati Banerjee, Krish Desai

accellera SYSTEMS INITIATIVE™

2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Motivation



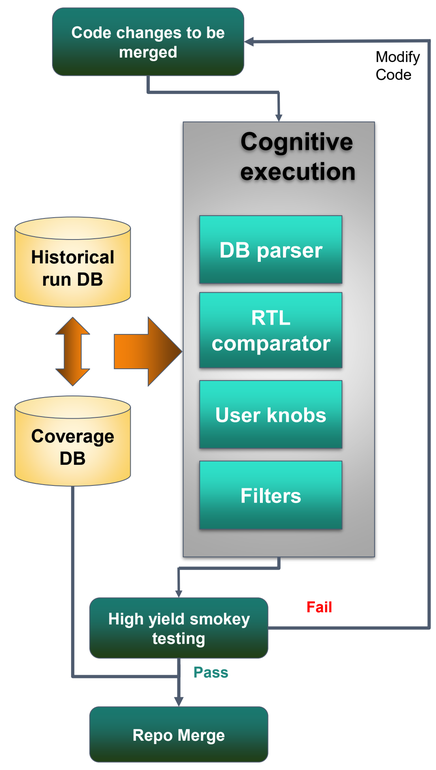
Pain points

- Broken repo top
- Manual debug and revert
- Delayed milestones
- Stalled development
- Restrict pipeline CLs
- Bug escapees

Proposed Methodology

Features :

- RTL change cognitive smoke testing to maximise bug hunting
- Simulator and vendor independent
  - Adapter layer provided to to handle simulator specific switches
- User knobs to handle the control the execution
  - Runtime budget
  - Machine resource /license bottleneck
- Self learning algorithm
- Fast access database to keep the decoding overhead less than couple of mins



Advantages

Early Bug Detection
Catches critical issues at the earliest stage, saving time and effort. It significantly left shifts detection of bugs on critical design paths.

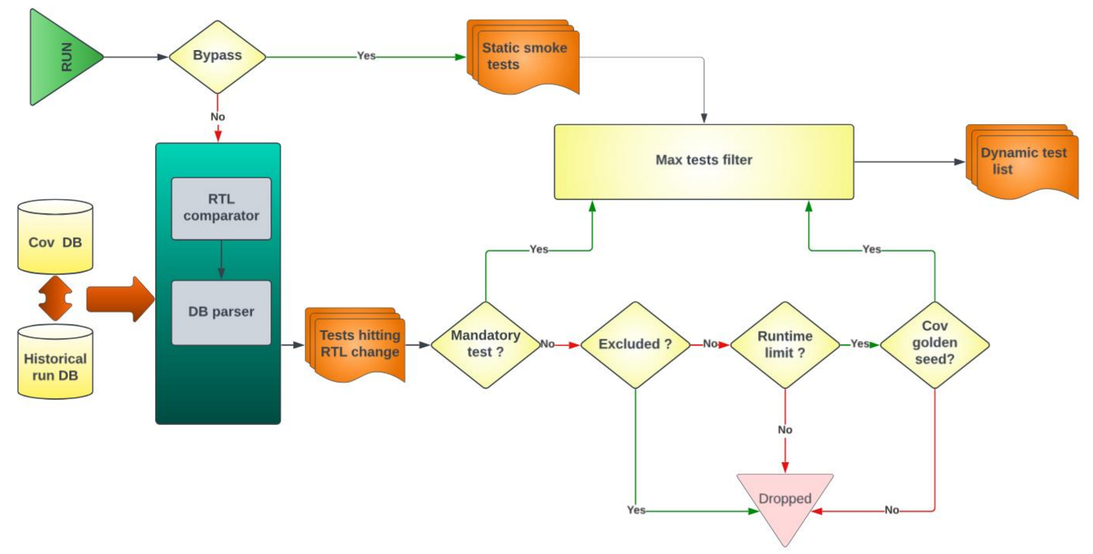
Code Quality Improvement
Prevents broken code from being merged into the main codebase.

Fast Feedback Loops
Developers get immediate feedback on their code changes.

Increases Stability
Ensures that the build is stable for more extensive testing (e.g., regression, performance).

Compatibility
Compatible with all version control systems like git, SV. Suitable for any code base not limited to just HDL languages.

Algorithm

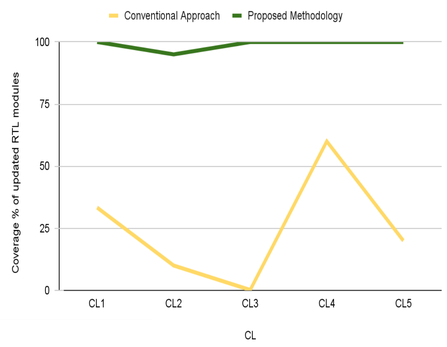


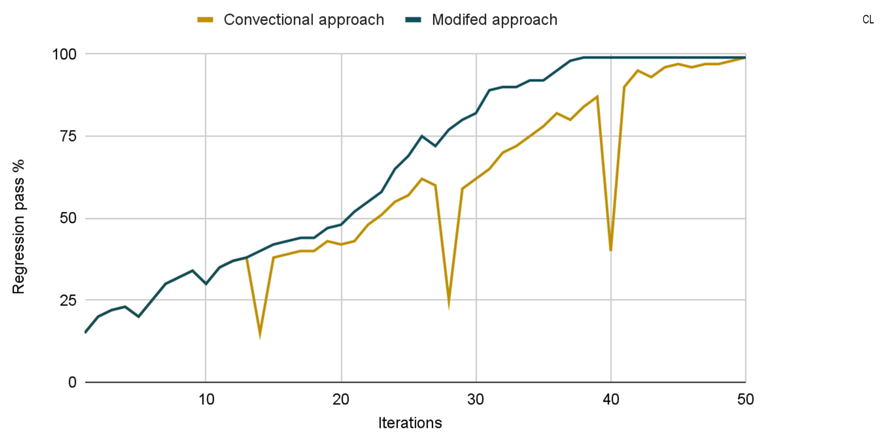
Decision points

01	Runtime budget	• Selecting tests within a defined time limit. • Longer running tests can be excluded.
02	No of licenses/ tests	• The number of smoke tests can be capped by available licenses
03	Golden seeds	• Choosing the best test from the set of equal-priority options
04	Historical runs	• Smoke tests are filtered by regression status • Known failures are excluded.
05	Exclusions	• User specific test exclusions
06	Mandatory tests	• User can mark the mandatory tests

Results/Conclusion

- Deployed on multiple projects having a huge code base (1K+ files)
- Reduced bad CLs merge by 98%
- Regression closure left shifted by 4 weeks





REFERENCES

- https://www.geeksforgeeks.org/software-engineering/smoke-testing-software-testing/
- https://developers.google.com/edu/python/utilities

Google