ENHANCING VERIFICATION EFFICIENCY WITH AUTOFOCUS ON THE VERISIUM PLATFORM

2025.12.16
NGHIA NGUYEN
ENGINEER, HWVN
HPC
RENESAS ELECTRONICS CORPORATION

AGENDA

RENESAS ELECTRONIC

INTRODUCTION

WHO WE ARE

Renesas Electronics empowers a safer, smarter and more sustainable future where technology helps make our lives easier.

The leading global provider of microcontrollers, Renesas combines our expertise in embedded processing, analog, power and connectivity to deliver complete semiconductor solutions. These Winning Combinations accelerate time to market for automotive, industrial, infrastructure and IoT applications, enabling billions of connected, intelligent devices that enhance the way people work and live.

Headquarters
Tokyo, Japan

Approx. 21,000
employees *

Operating in
30+ countries

1,502.7 billion yen
revenue in 2022

Approx. 20,000
patents & pending applications

* Consolidated, as of December 31, 2022

SOLUTION OFFERING

Automotive

Highly reliable vehicle control, safe and secure autonomous driving, Eco-friendly electric vehicles

Industrial

Lean, flexible and smart industry

Infrastructure

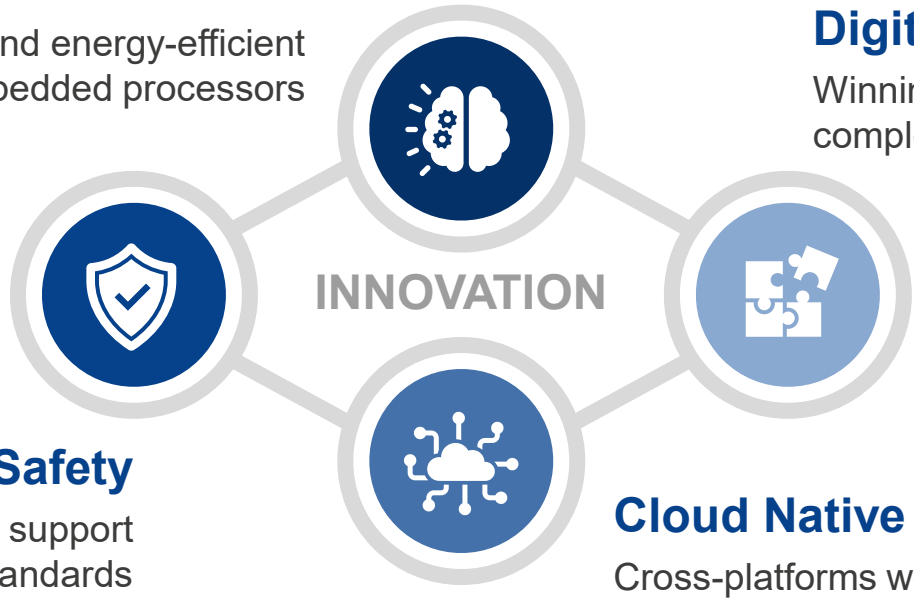
Robust infrastructure, enabling safety and efficiency

IoT

Comfortable, safe and healthy lifestyles through IoT

CORE TECHNOLOGIES

Renesas is poised to address the requirements of every industry with our four pillars of core technologies AI, Security & Safety, Digital & Analog & Power Solution and Cloud Native as the key to encouraging more innovation that is finely-tuned to achieve future sustainable growth.



AI

A broad set of power and energy-efficient embedded processors

Digital & Analog & Power Solution

Winning Combinations that combine our complementary product portfolios

Security & Safety

Comprehensive technology and support that meet the industry's stringent standards

Cloud Native

Cross-platforms working with partners in different verticals and organizations

RENESAS DESIGN VIETNAM

INTRODUCTION

Office HCMC
District 7
(Started 05 Oct 2004)

Office Da Nang
Thanh Khe District
(Started 21 Apr 2021)

Major operations: Design hardware (SoCs and MCUs) and software for automotive and industry applications mainly.

CONVENTIONAL FLOW AND PROPOSED FLOW

CONVENTIONAL FLOW

1. Complex and large design
2. Involvement of many sites

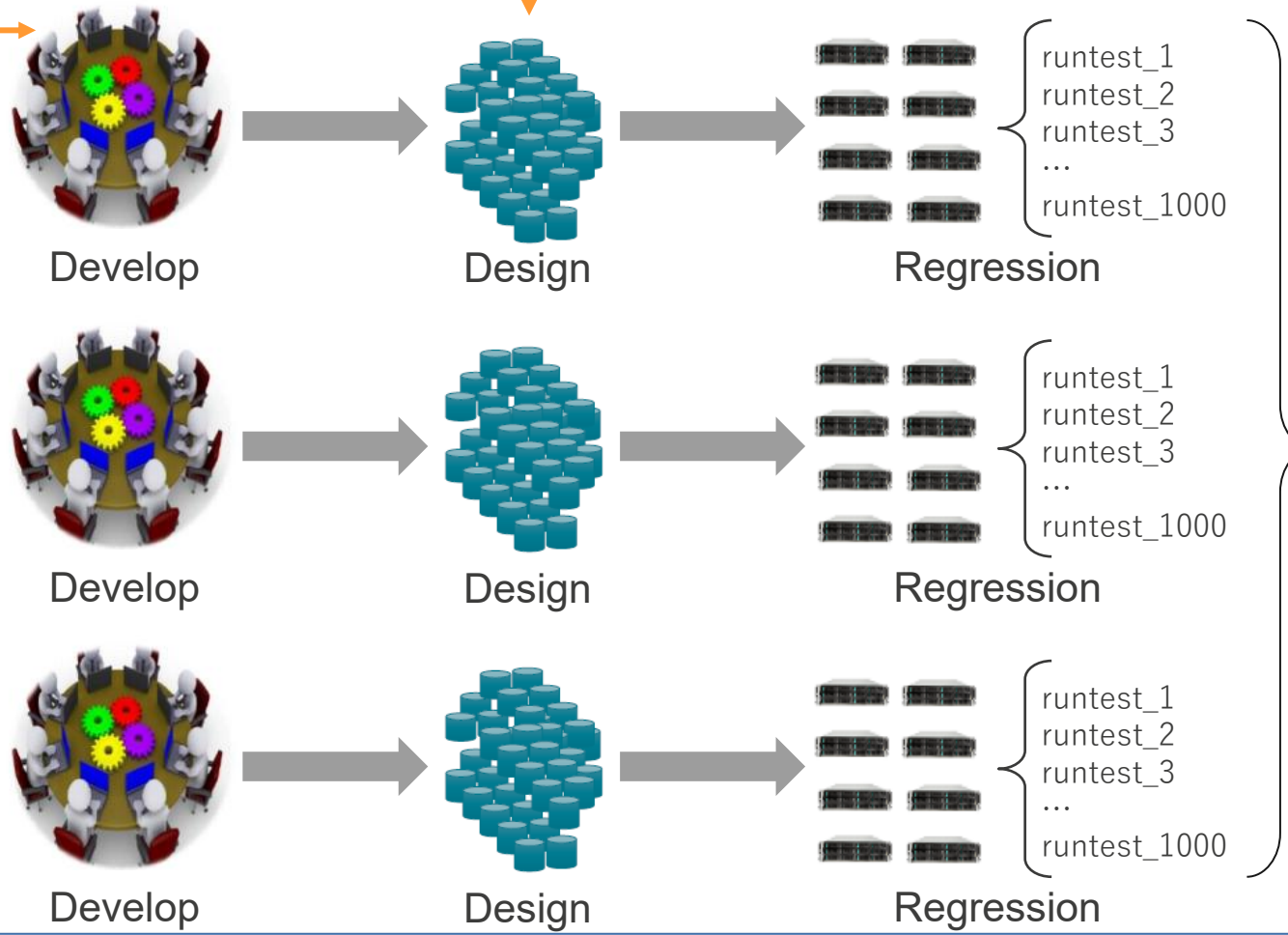
Difficult understanding of design or test-run

+

Design updated frequently

Inefficient regression after each design update

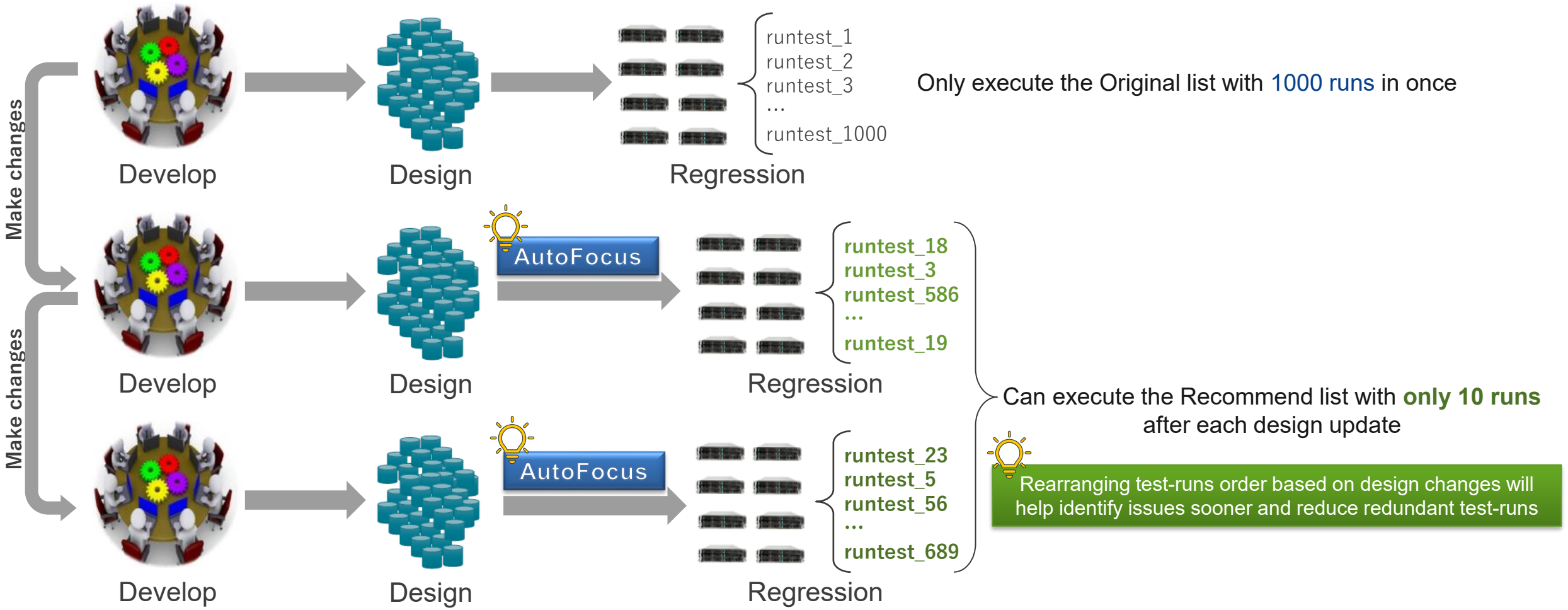
Need to perform more efficient regression



Always execute the Original list with 1000 runs after each design update

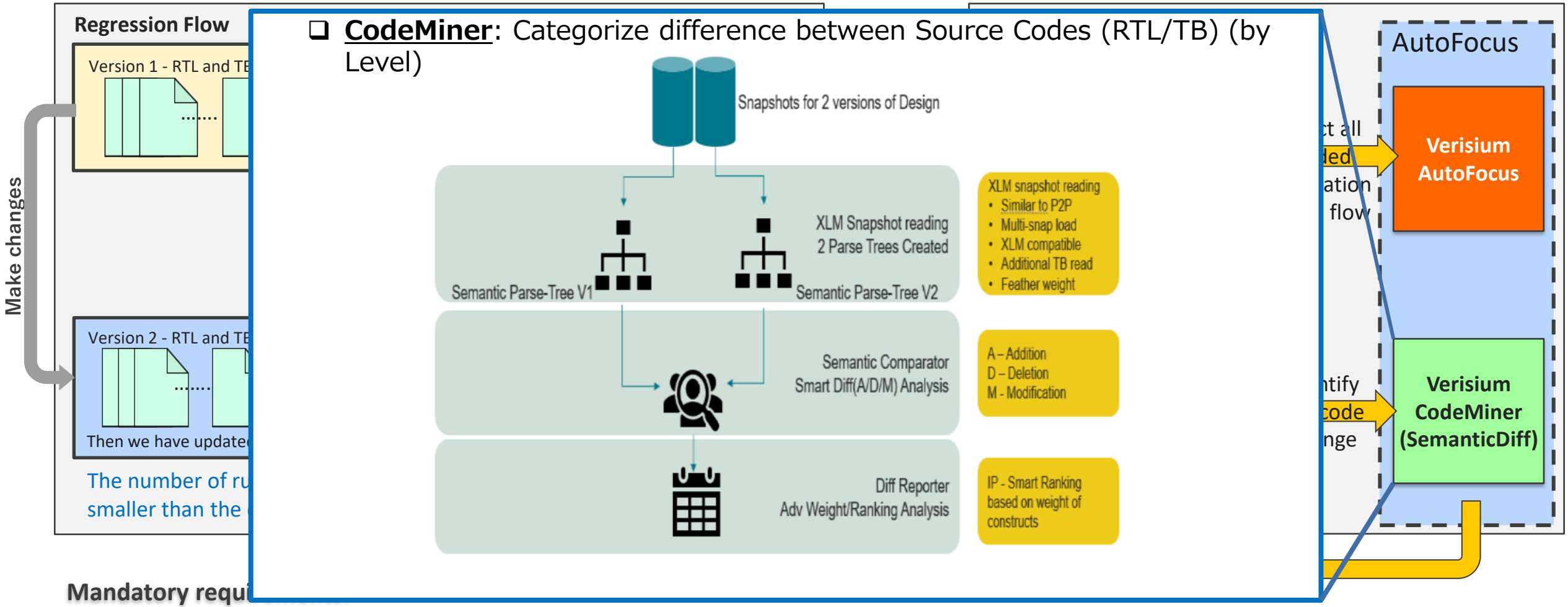
After each design update, we must **rerun all tests**, which is costly in terms of time and resources (engineers, licenses, servers, …)

PROPOSED FLOW WITH AUTOFOCUS



Significant reduction from the original number of runs after each design update makes the regression more efficient

FLOWS IN AUTOFOCUS



Mandatory requi

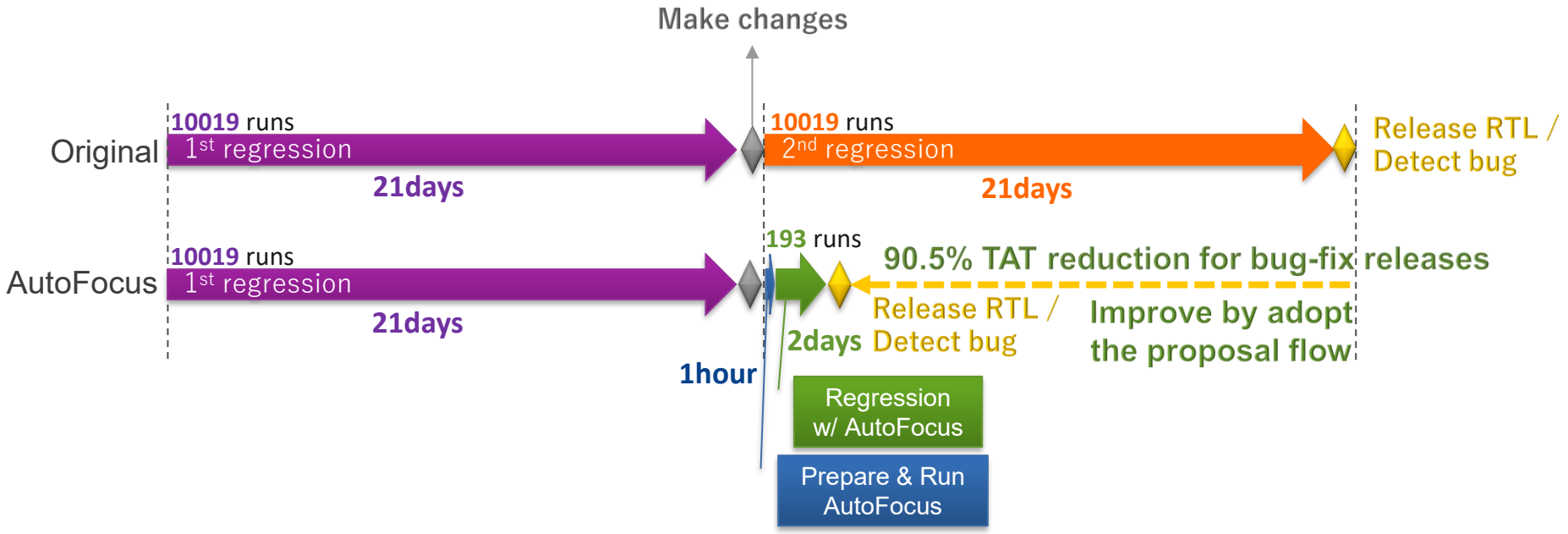
- Generate a shorter regression list which must have tests that can detect the issue of code change.
- Hit the coverage threshold of the design

EVALUATION RESULT

EVALUATION RESULT

USING AUTOFOCUS TO HELP OPTIMIZE THE REGRESSION PROCESS FOR BUG-FIX RELEASES

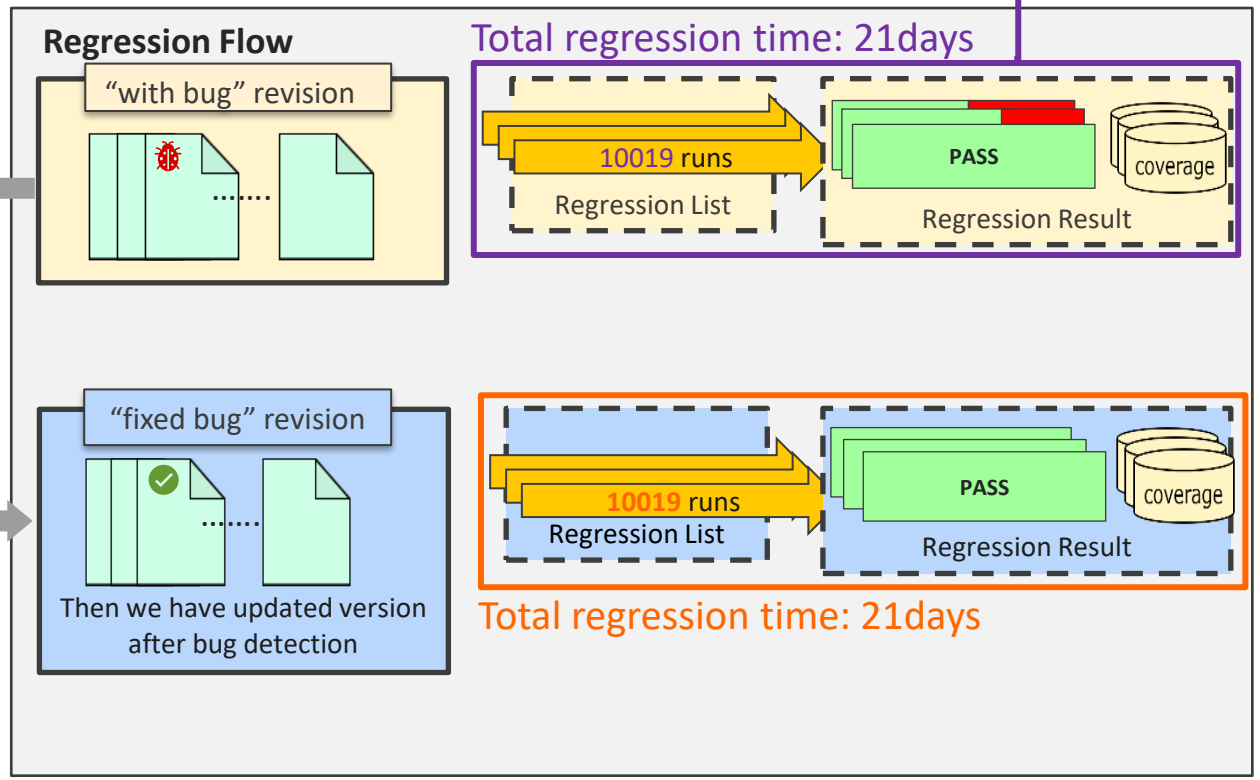
Original: Execute all run-tests of original list in the first time regression, and start regression again after code changes.
AutoFocus: Also execute all run-tests of original list in the first time regression, and start regression again based on recommend list of AutoFocus after code change
Condition: Release RTL or Detect bug from code changes.



EVALUATION RESULT

USING AUTOFOCUS TO HELP OPTIMIZE THE REGRESSION PROCESS FOR BUG-FIX RELEASES

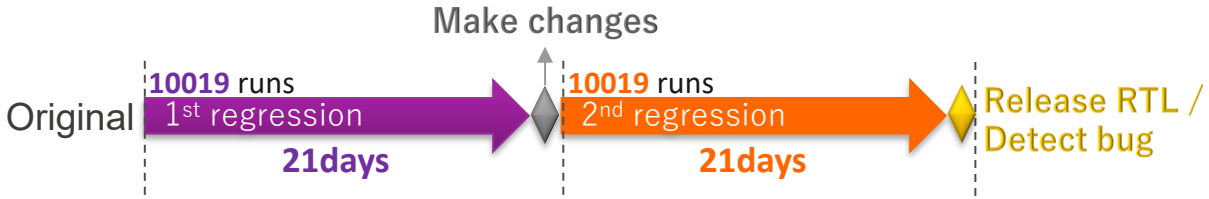
The input data and associated information for **Original Flow**



Bugcan be detected following test patterns:

- runtest_234919239
- runtest_309942689
- runtest_241449698
- runtest_253362130
- runtest_136807909
- runtest_335165086
- runtest_315524471
- runtest_239738026

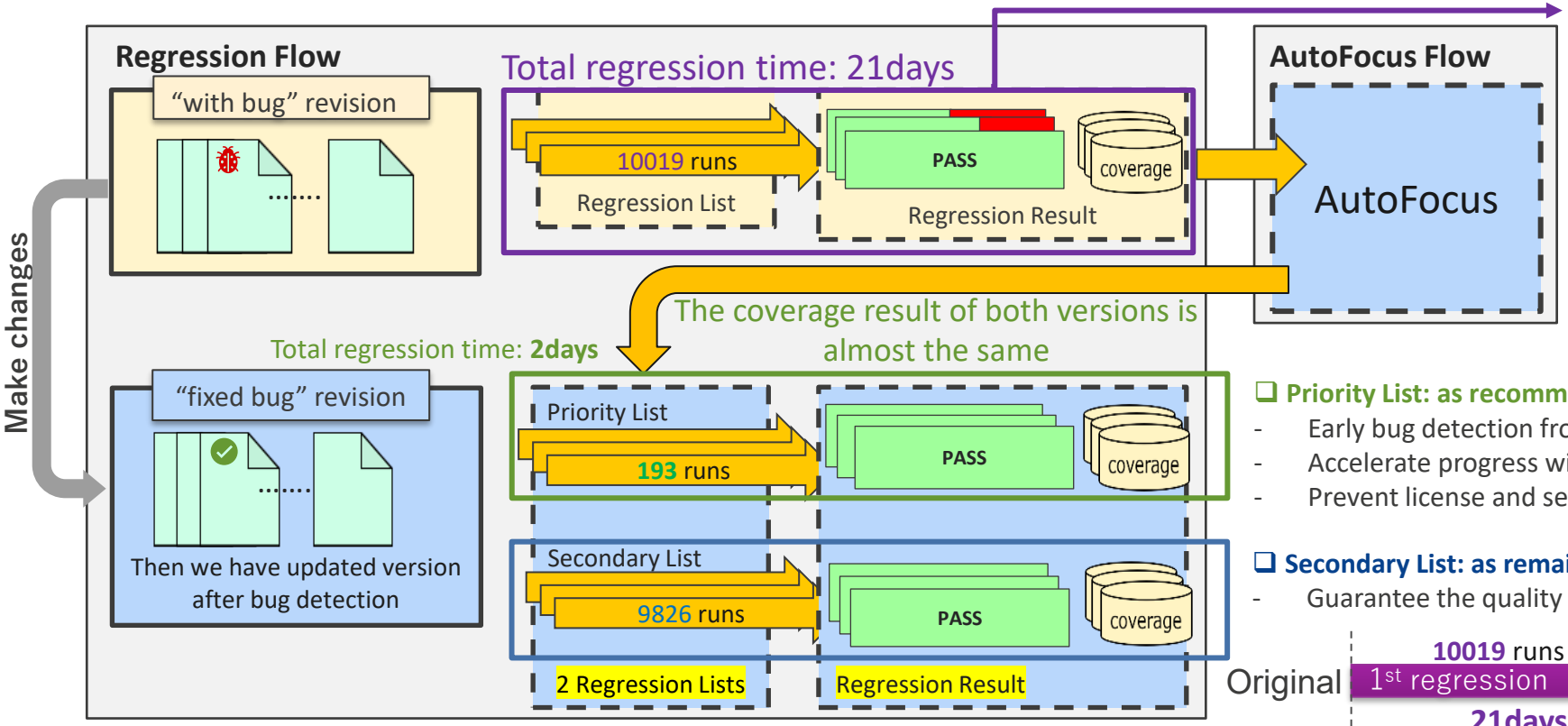
Too many runs leads to a lot of time spent re-executing regression!!



EVALUATION RESULT

USING AUTOFOCUS TO HELP OPTIMIZE THE REGRESSION PROCESS FOR BUG-FIX RELEASES

The input data and associated information for **AutoFocus Flow**



Bug can be detected following test patterns:

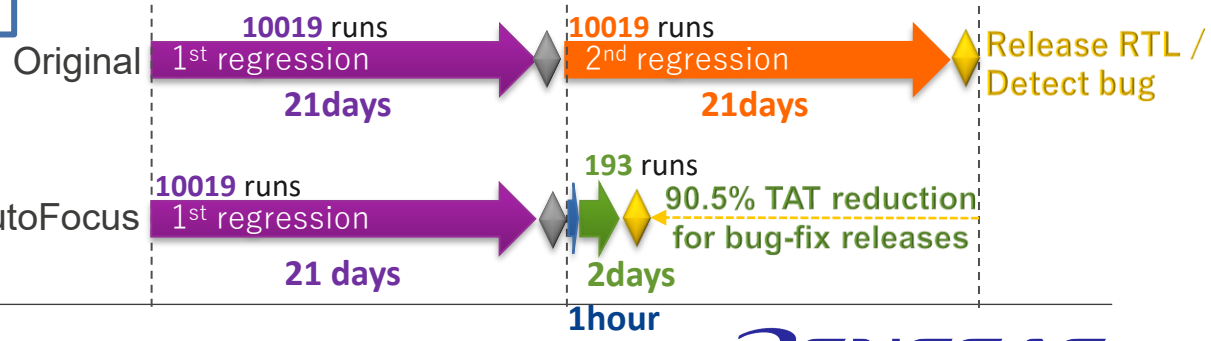
- runtest_234919239
- runtest_309942689
- runtest_241449698
- runtest_253362130
- runtest_136807909
- runtest_335165086
- runtest_315524471
- runtest_239738026

❑ **Priority List: as recommend runs list**

- Early bug detection from code changes
- Accelerate progress with early RTL release upon hitting design coverage threshold
- Prevent license and server overload during peak times

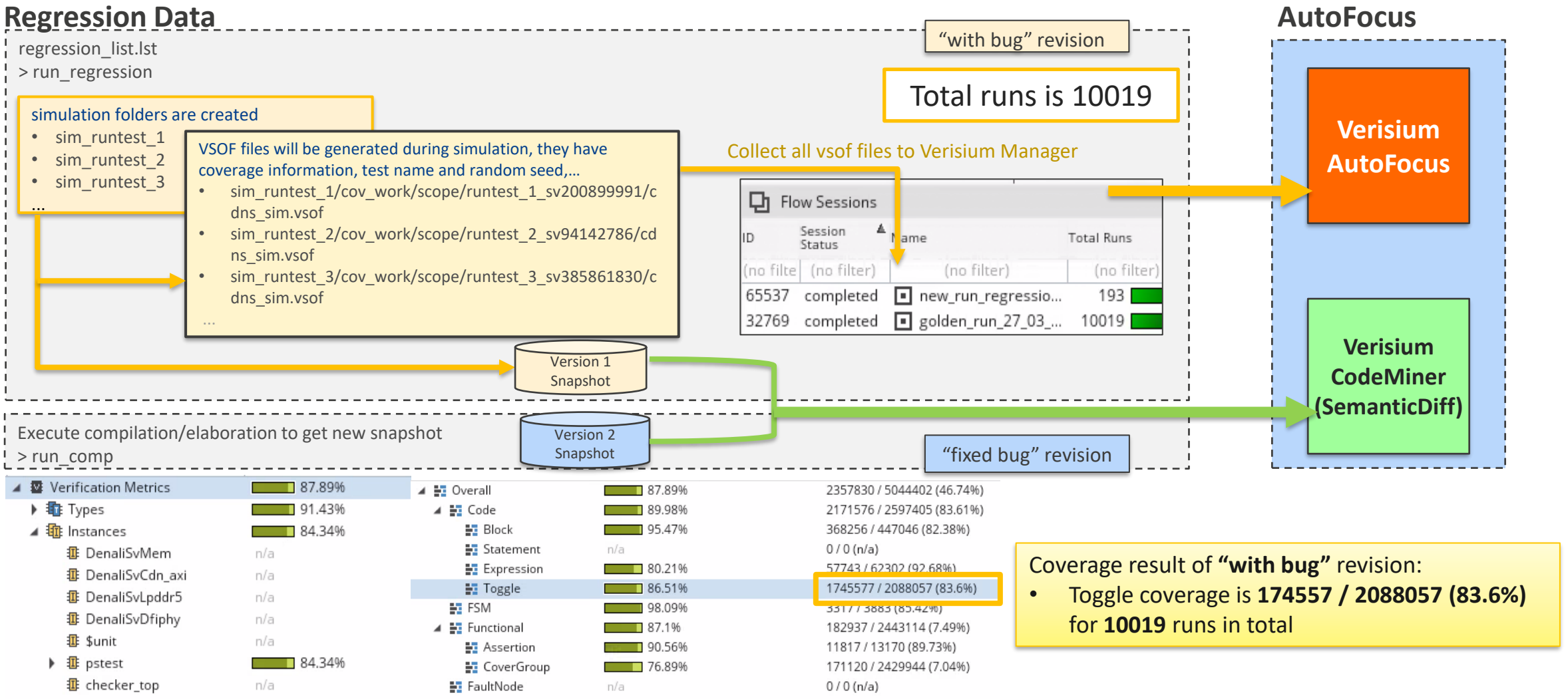
❑ **Secondary List: as remain runs list**

- Guarantee the quality of verification



EVALUATION RESULT

USING AUTOFOCUS TO HELP OPTIMIZE THE REGRESSION PROCESS FOR BUG-FIX RELEASES



Verification Metrics	87.89%
Types	91.43%
Instances	84.34%
DenaliSvMem	n/a
DenaliSvCdn_axi	n/a
DenaliSvLpddr5	n/a
DenaliSvDfiphy	n/a
$unit	n/a
pstest	84.34%
checker_top	n/a

Overall	87.89%	2357830 / 5044402 (46.74%)
Code	89.98%	2171576 / 2597405 (83.61%)
Block	95.47%	368256 / 447046 (82.38%)
Statement	n/a	0 / 0 (n/a)
Expression	80.21%	57743 / 62302 (92.68%)
Toggle	86.51%	1745577 / 2088057 (83.6%)
FSM	98.09%	3317 / 3383 (85.42%)
Functional	87.1%	182937 / 2443114 (7.49%)
Assertion	90.56%	11817 / 13170 (89.73%)
CoverGroup	76.89%	171120 / 2429944 (7.04%)
FaultNode	n/a	0 / 0 (n/a)

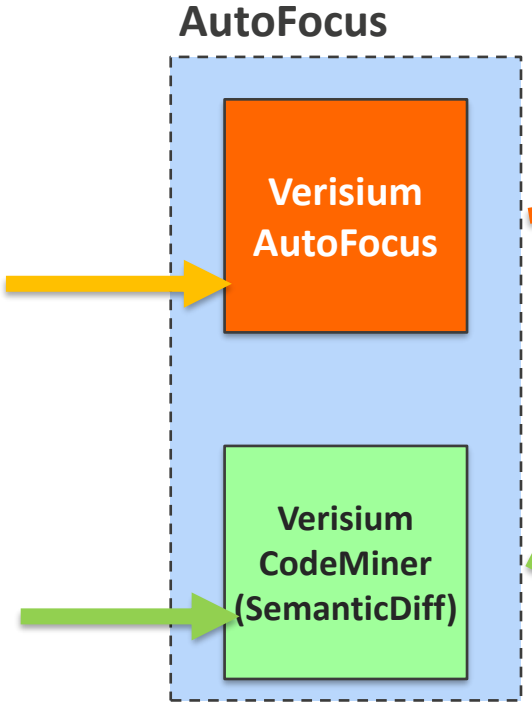
Coverage result of **"with bug"** revision:

- Toggle coverage is **174557 / 2088057 (83.6%)** for **10019** runs in total

EVALUATION RESULT

USING AUTOFOCUS TO HELP OPTIMIZE THE REGRESSION PROCESS FOR BUG-FIX RELEASES

AutoFocus processing…



Creates a recommended list based on the type of coverage from the user (we chose toggle coverage for this evaluation).


```
[{"name": "runtest_339886883", "sv_seed": "339886883", "accumulate coverage": "81.09%", "duration": 5416},
{"name": "runtest_269832203", "sv_seed": "269832203", "accumulate coverage": "89.28%", "duration": 6524},
{"name": "runtest_102172081", "sv_seed": "102172081", "accumulate coverage": "92
{"name": "runtest_188338954", "sv_seed": "188338954", "accumulate coverage": "94
{"name": "runtest_62858669", "sv_seed": "62858669", "accumulate coverage": "95.5
{"name": "runtest_89347332", "sv_seed": "89347332", "accumulate coverage": "96.6
{"name": "runtest_243681185", "sv_seed": "243681185", "accumulate coverage": "97.26%", "duration": 11546},
{"name": "runtest_21669424", "sv_seed": "21669424", "accumulate coverage": "97.81%", "duration": 4078},
{"name": "runtest_287486549", "sv_seed": "287486549", "accumulate coverage": "98.27%", "duration": 4354},
{"name": "runtest_234919239", "sv_seed": "234919239", "accumulate coverage": "98.48%", "duration": 1859},
{"name": "runtest_309942689", "sv_seed": "309942689", "accumulate coverage": "98.64%", "duration": 4082},
{"name": "runtest_241449698", "sv_seed": "241449698", "accumulate coverage": "98.75%", "duration": 11061},
{"name": "runtest_253362130", "sv_seed": "253362130", "accumulate coverage": "98.84%", "duration": 6751},
{"name": "runtest_136807909", "sv_seed": "136807909", "accumulate coverage": "98.93%", "duration": 4438},
{"name": "runtest_335165086", "sv_seed": "335165086", "accumulate coverage": "99.0%", "duration": 3910},
{"name": "runtest_315524471", "sv_seed": "315524471", "accumulate coverage": "99.07%", "duration": 2312},
{"name": "runtest_239738026", "sv_seed": "239738026", "accumulate coverage": "99.13%", "duration": 16021},
{"name": "runtest_229537123", "sv_seed": "229537123", "accumulate coverage": "99.18%", "duration": 2410},
{"name": "runtest_115387366", "sv_seed": "115387366", "accumulate coverage": "99.23%", "duration": 6878},
{"name": "runtest_265320195", "sv_seed": "265320195", "accumulate coverage": "99.26%", "duration": 4076},
{"name": "runtest_228844912", "sv_seed": "228844912", "accumulate coverage": "99.3%", "duration": 3343},
{"name": "runtest_36087782", "sv_seed": "36087782", "accumulate coverage": "99.34%", "duration": 5329},
{"name": "runtest_148564376", "sv_seed": "148564376", "accumulate coverage": "99.38%", "duration": 3789},
{"name": "runtest_87266945", "sv_seed": "87266945", "accumulate coverage": "99.4%", "duration": 1721},
```


Total runs is 193 / 10019

Detailed list of design changes with impact ranking

```
# ----------------------------------------
# tool         : semanticdiff 23.09-a071
# build date   : 20231020
# build config : 2023.09--2b49aa2e072e--/depot/tool/
# ----------------------------------------

Module analyzed: 518     diff found in : 5
Interface analyzed: 2    diff found in : 0
LibItem analyzed: 5      diff found in : 0
Program analyzed: 0      diff found in : 0
Global analyzed: 0       diff found in : 0
Primitive analyzed: 4    diff found in : 0
Return status 0
```

Entity Type	Entity Name	Entity golden File/Line	Entity new File/Line	Mode	Modified lines	Added lines	Deleted lines	Total lines	Diff count	Total count	Rank
Module	module_test0	/test/rtl/module_test.v(102654)	/test/rtl/module_test_fix.v(102658)	Modified	1	2	0	707	21	4989	1
Module	module_test1	/test/rtl/module_test.v(9400)	/test/rtl/module_test_fix.v(94001)	Modified	1	0	0	2453	12	10191	2
Module	module_test2	/test/rtl/module_test.v(101932)	/test/rtl/module_test_fix.v(101934)	Modified	1	0	0	710	10	3964	3
Module	module_test3	/test/rtl/module_test.v(22)	/test/rtl/module_test_fix.v(22)	Modified	1	0	0	3682	3	5597	4

EVALUATION RESULT

EVALUATE THE OUTPUT QUALITY OF AUTOFOCUS – THE OPTIMIZED REGRESSION LIST

The new recommended regression list:

Total runs is 193 / 10019


```
[{"name": "runtest_339886883", "sv_seed"
{"name": "runtest_269832203", "sv_seed":
{"name": "runtest_102172081", "sv_seed":
{"name": "runtest_188338954", "sv_seed": "188338954", "accumulate coverage": "94.31%", "duration": 14249},
{"name": "runtest_62858669", "sv_seed": "62858669", "accumulate coverage": "95.54%", "duration": 6086},
{"name": "runtest_89347332", "sv_seed": "89347332", "accumulate coverage": "96.64%", "duration": 2700},
{"name": "runtest_243681185", "sv_seed": "243681185", "accumulate coverage": "97.26%", "duration": 11546},
{"name": "runtest_21669424", "sv_seed": "21669424", "accumulate coverage": "97.81%", "duration": 4078},
{"name": "runtest_287486549", "sv_seed": "287486549", "accumulate coverage": "98.27%", "duration": 4354},
{"name": "runtest_234919239", "sv_seed": "234919239", "accumulate coverage": "98.48%", "duration": 1859},
{"name": "runtest_309942689", "sv_seed": "309942689", "accumulate coverage": "98.64%", "duration": 4082},
{"name": "runtest_241449698", "sv_seed": "241449698", "accumulate coverage": "98.75%", "duration": 11061},
{"name": "runtest_253362130", "sv_seed": "253362130", "accumulate coverage": "98.84%", "duration": 6751},
{"name": "runtest_136807909", "sv_seed": "136807909", "accumulate coverage": "98.93%", "duration": 4438},
{"name": "runtest_335165086", "sv_seed": "335165086", "accumulate coverage": "99.0%", "duration": 3910},
{"name": "runtest_315524471", "sv_seed": "315524471", "accumulate coverage": "99.07%", "duration": 2312},
{"name": "runtest_239738026", "sv_seed": "239738026", "accumulate coverage": "99.13%", "duration": 16021},
{"name": "runtest_229537123", "sv_seed": "229537123", "accumulate coverage": "99.18%", "duration": 2410},
{"name": "runtest_115387366", "sv_seed": "115387366", "accumulate coverage": "99.23%", "duration": 6878},
{"name": "runtest_265320195", "sv_seed": "265320195", "accumulate coverage": "99.26%", "duration": 4076},
{"name": "runtest_228844912", "sv_seed": "228844912", "accumulate coverage": "99.3%", "duration": 3343},
{"name": "runtest_36087782", "sv_seed": "36087782", "accumulate coverage": "99.34%", "duration": 5329},
{"name": "runtest_148564376", "sv_seed": "148564376", "accumulate coverage": "99.38%", "duration": 3789},
{"name": "runtest_87266945", "sv_seed": "87266945", "accumulate coverage": "99.4%", "duration": 1721},
```


The results after testing are totally matched with our expectations for this evaluation.

The new recommended list has the tests that potential to detect the bug.


```
{"name": "runtest_234919239", "sv_seed": "234919239", "accumulate coverage": "98.48%", "duration": 1859},
{"name": "runtest_309942689", "sv_seed": "309942689", "accumulate coverage": "98.64%", "duration": 4082},
{"name": "runtest_241449698", "sv_seed": "241449698", "accumulate coverage": "98.75%", "duration": 11061},
{"name": "runtest_253362130", "sv_seed": "253362130", "accumulate coverage": "98.84%", "duration": 6751},
{"name": "runtest_136807909", "sv_seed": "136807909", "accumulate coverage": "98.93%", "duration": 4438},
{"name": "runtest_335165086", "sv_seed": "335165086", "accumulate coverage": "99.0%", "duration": 3910},
{"name": "runtest_315524471", "sv_seed": "315524471", "accumulate coverage": "99.07%", "duration": 2312},
{"name": "runtest_239738026", "sv_seed": "239738026", "accumulate coverage": "99.13%", "duration": 16021},
```


Simulation result of 193 runs:

- "new toggle coverage" / "original toggle coverage" = 83.18 / 83.6 ~ **99.5%**
- "new total runs" / "original total runs" = 193 / 10019 ~ **1.9%**
  (this is for reference because the original runs included other coverage to achieve).

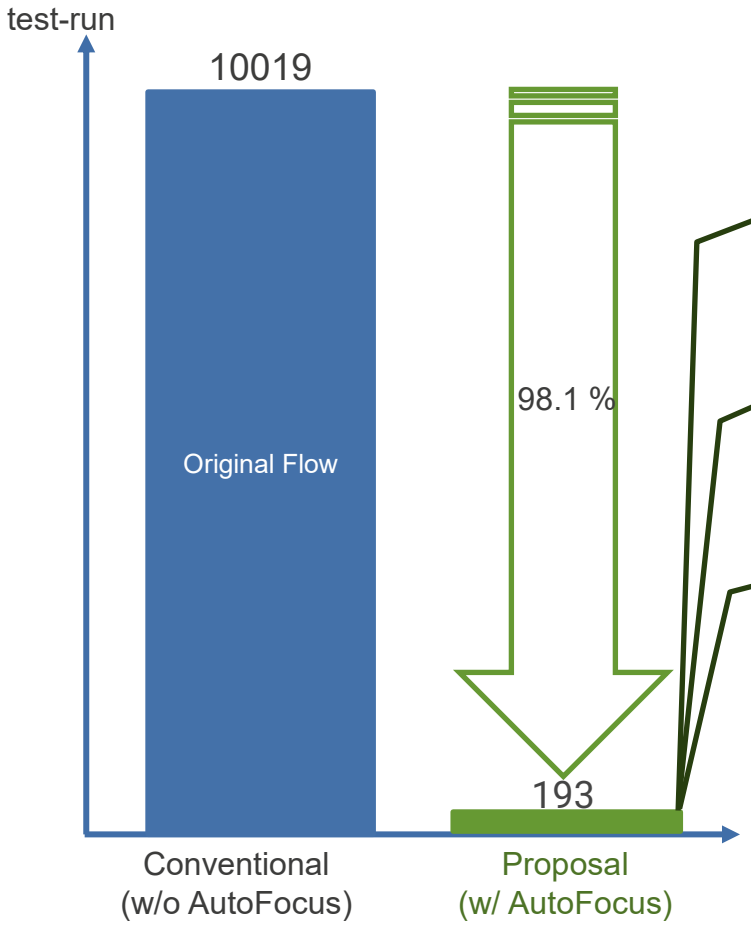
Verification Metrics	86.49%	
Types	89.31%	
Instances	83.66%	
DenaliSvMem	n/a	
DenaliSvCdn_axi	n/a	
DenaliSvLpddr5	n/a	
DenaliSvDfiphy	n/a	
$unit	n/a	
pstest	83.66%	
checker_top	n/a	

Overall	86.49%	2311026 / 3059955 (75.52%)
Code	89.49%	2157042 / 2597405 (83.05%)
Block	94.92%	363143 / 447046 (81.2
Statement	n/a	0 / 0 (n/a)
Expression	77.38%	57056 / 62302 (91.58%)
Toggle	86.3%	1736843 / 2088057 (83.18%)
FSM	93.66%	3226 / 3883 (83.08%)
Functional	84.19%	150758 / 458667 (32.87%)
Assertion	86.69%	11191 / 13164 (85.01%)
CoverGroup	73.01%	139567 / 445503 (31.33%)
FaultNode	n/a	0 / 0 (n/a)

Significantly reduce the total regression runs up to 98.1% while achieving up to 99.5% of toggle coverage

SUMMARY

SUMMARY



Evaluation results: We evaluated feature by measuring human resource

- ❑ **AutoFocus**
  - ➢ Result analysis effort of this case: -98.1% compared to the original workflow

193 recommend run-tests focused on the code change points

➥ There were the tests that potential to detect the bug -> Early bug detection from code changes

Hit the user coverage threshold up to 99.5% with 193 run-tests

➥ Accelerate project progress with early RTL release upon hitting design coverage threshold

A new regression list where the number of items is smaller than the original list

➥ Prevent license and server overload during peak times

Efficiency of function verification is as expected.

SUGGESTION FOR USER

- AutoFocus select the run-tests that trigger the code changes and satisfy the user requirement (coverage threshold, execution time, ...) by analyzing regression data and code changes
  - The recommend run-tests list from AutoFocus as Priority List to accelerate the project progress.
  - The remain run-tests list from the original regression list as Secondary List to guarantee the quality of verification.

FEEDBACK TO CADENCE

- It is more convenience if be able to use AutoFocus to select most appropriate tests for regression of derivative products
- AutoFocus can run with Sim AI to be able to create the new tests/seeds to cover the code changes

Cadence is already working on the improvements based on our feedback

We appreciate Cadence Team for your valuable support during the evaluation

Renesas.com