2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
DECEMBER 6 - 7, 2022

Improving Simulation Regression Efficiency using a Machine Learning-based Method in Design Verification

Deepak Narayan Gadde, Infineon Technologies Dresden GmbH & Co. KG
Sebastian Simon, Infineon Technologies Dresden GmbH & Co. KG
Djones Lettnin, Infineon Technologies AG
Thomas Ziller, Cadence Design Systems GmbH

accellera SYSTEMS INITIATIVE

Agenda

- Motivation and Problem Statement
- Cadence Xcelium ML
- Application on Designs
- Xcelium ML vs Ranking
- Proposed Methodology
- Summary

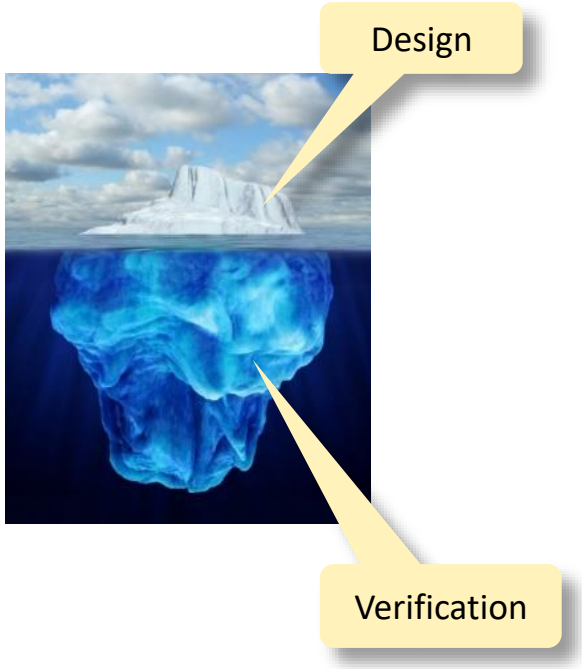
Agenda

- Motivation and Problem Statement
- Cadence Xcelium ML
- Application on Designs
- Xcelium ML vs Ranking
- Proposed Methodology
- Summary

Verification of Systems on Chip became a time-consuming task

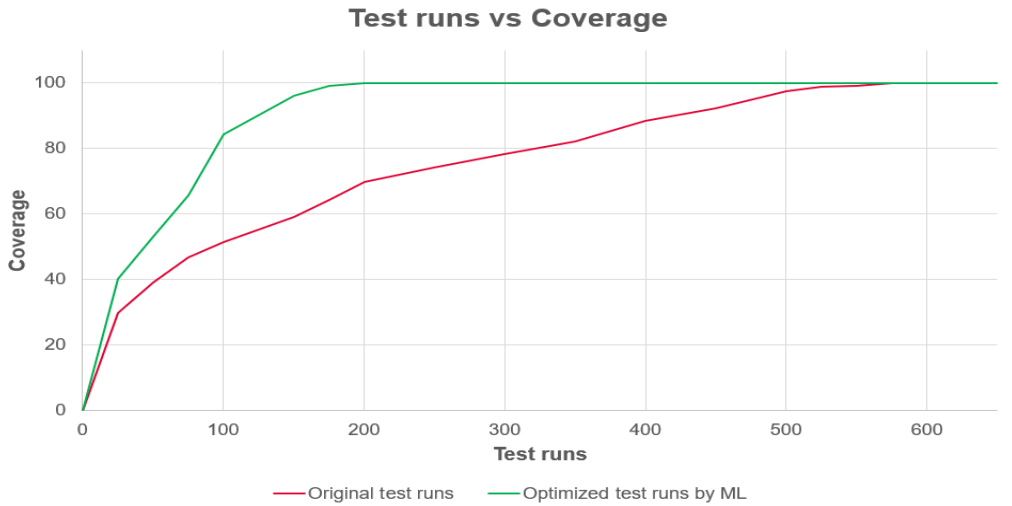
- Designs are becoming more complex, due to
  - Technology scaling: Ability to fabricate more
  - Consumer demands more from a single chip
  - Mixed-signal designs
  - Safety/Security critical devices and more…

- At present, pre-silicon verification typically consumes significantly more effort than design



Money and time are crucial...

- **Problem:** Huge number of test simulations in the Regression
  - High demand for simulator licenses
  - Long turn-around times
  - Uncertainty in coverage regain
- **Goals:**

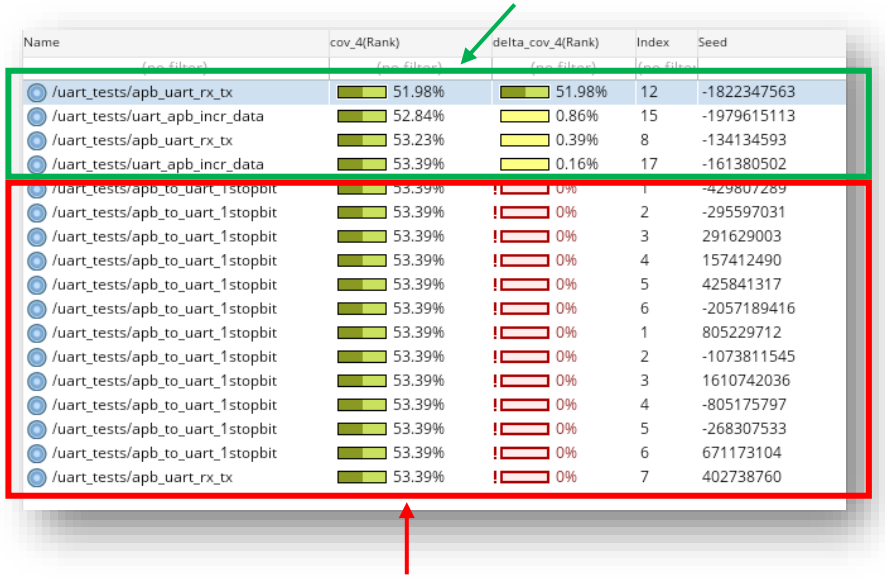


Common Solution used so far - Ranking

Definition: A collection of test-seed pairs of the original regression to reach its achieved coverage

- **Pro**:
  - Easy
  - built-in (e.g. vManager)
  - Proven technology
- **Con**:
  - Effectively re-simulation (like directed testing)
  - Exact same coverage as original regression
  - Inability to hit new coverage/ identify new bugs

Subset - same coverage as original regression

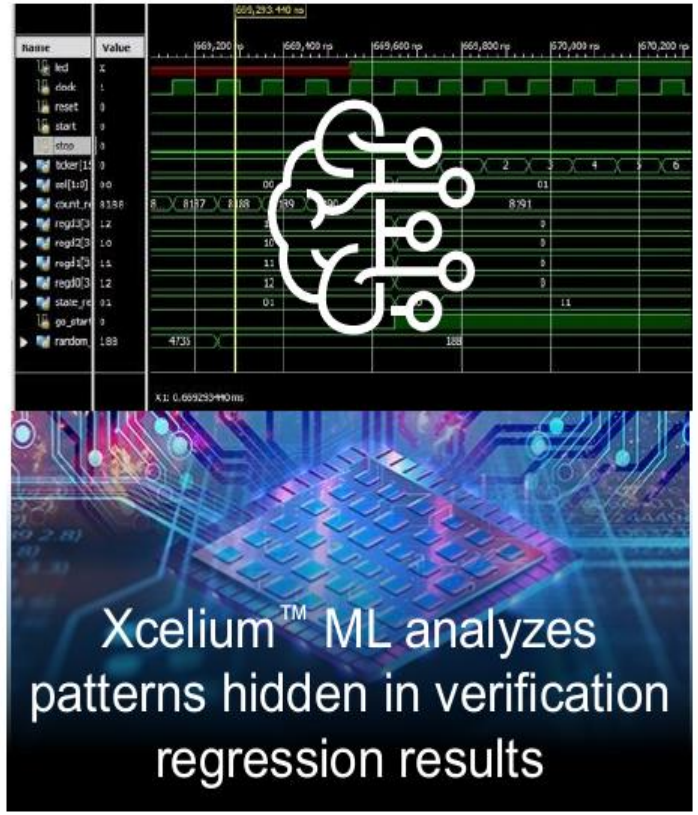


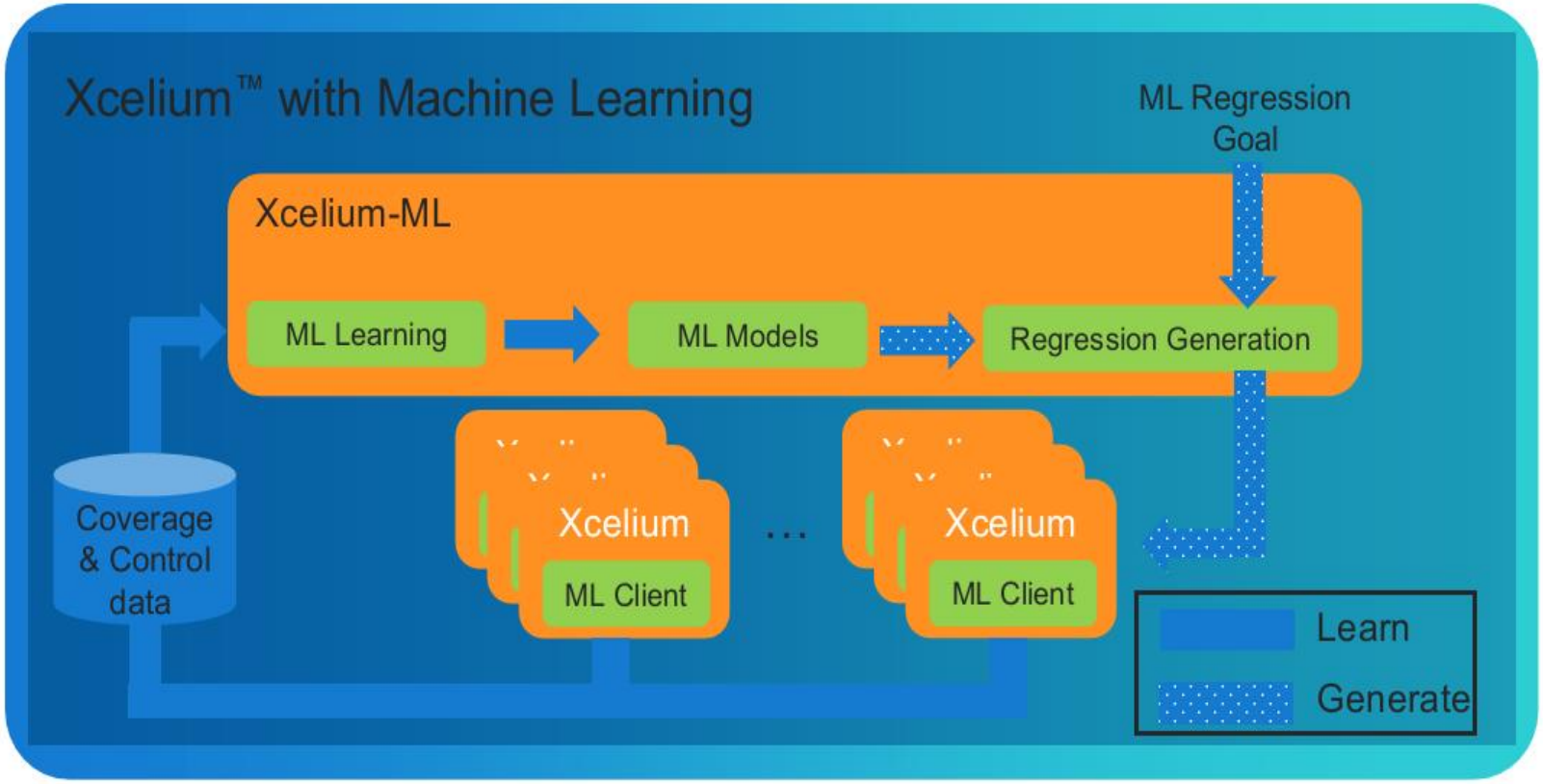
No additional coverage contribution, can be neglected for ranked regression

Agenda

- Motivation and Problem Statement
- Cadence Xcelium ML
- Application on Designs
- Xcelium ML vs Ranking
- Proposed Methodology
- Summary

Cadence Xcelium ML High Level View





Source: Cadence Design Systems, Inc

Faster Regressions with Xcelium ML

Original Regression

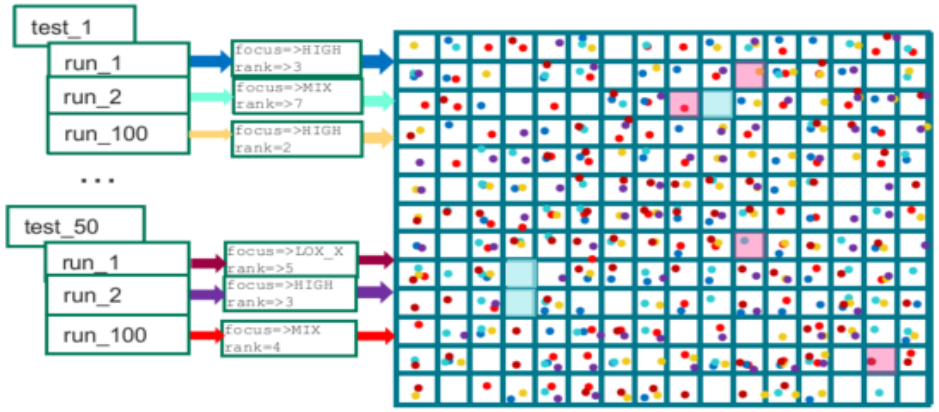
- 50 tests, 100 seeds per test (5,000 runs)

Random control

```
class cfg_c extends uvm_sequence_item;
  rand focus_e focus;
  rand [2:0] rank;
  ...
endclass
```

```
function void test::setup();
  cfg_c cfg = get_config();
  cfg.randomize();
  set_config_info(cfg);
endfunction
```

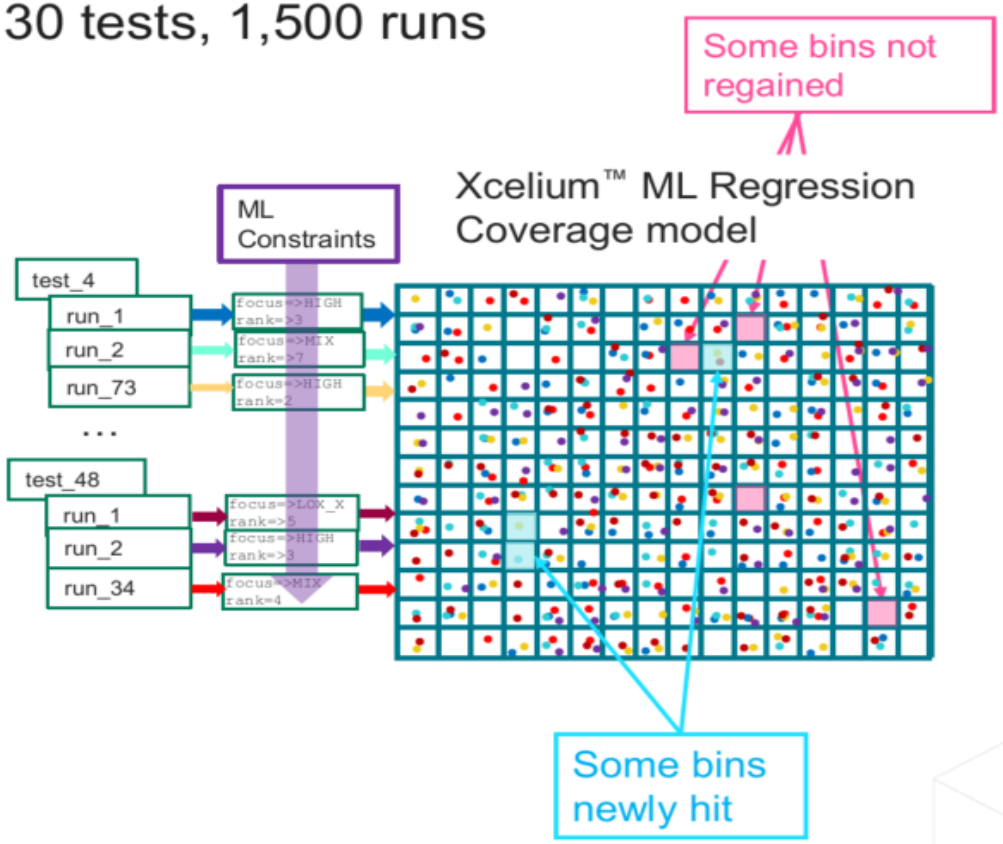
Original Regression
Coverage model



Generate new regression runs

Generated Regression

30 tests, 1,500 runs



Agenda

- Motivation and Problem Statement
- Cadence Xcelium ML
- Application on Designs
- Xcelium ML vs Ranking
- Proposed Methodology
- Summary

Application on an SoC Design - Overview

- Mixed Signal SoC
- Designed in VHDL, Verilog, SystemVerilog
- UVM (SV) testbench
- 356 Distinct Tests
- ≈ 5124 runs in the Regression

Configuration	CPU	Verification IP	Flash
P1	Yes	No	Simple
P2	Yes	Yes	Complex
P3	Yes	Yes	Simple
P4	No	No	Complex
P5	No	No	Simple
P6	No	Yes	Complex
P7	No	Yes	Simple

Application on an SoC Design – Detailed Results

Configuration	Original Runs	Original Runtime CPU hours	Optimization Runs	Opt. Runtime CPU hours	Compression in Runs	Coverage Regain	Compression in Runtime
P1	297	0.200	110	0.151	2.70	99.86%	1.32
P2	415	0.562	188	0.337	2.20	101.2%	1.66
P3	2959	2.813	786	0.785	3.76	97.96%	3.58
P4	140	0.213	116	0.133	1.20	98.07%	1.60
P5	1193	1.336	318	0.356	3.75	99.48%	3.75
P6	10	0.007	3	0.003	3.33	98.19%	2.33
P7	110	0.074	84	0.070	1.30	101.3%	1.05
Total	5124	5.205	1605	1.835	3.19	99.42%	2.83

Application on other Designs

- **Microprocessor Digital IP**
  - VHDL design
  - UVM-SV testbench
  - Already taped out
  - 260 runs with 26 distinct tests
  - Results (XceliumML)
    - 6x - 17x speed-up factor
    - ≈ 99.9% coverage regain

- **Radar based SoC**
  - System Verilog design
  - UVM-SV testbench
  - Ongoing project
  - 25 tests/ 271 runs @ Stage I
  - 35 tests/ 301 runs @ Stage II
  - Results (XceliumML):
    - 3x speed-up factor
    - Over 100% coverage regain
    - Able to discover more functional bugs

Agenda

- Motivation and Problem Statement
- Cadence Xcelium ML
- Application on Designs
- Xcelium ML vs Ranking
- Proposed Methodology
- Summary

Xcelium ML vs Ranking

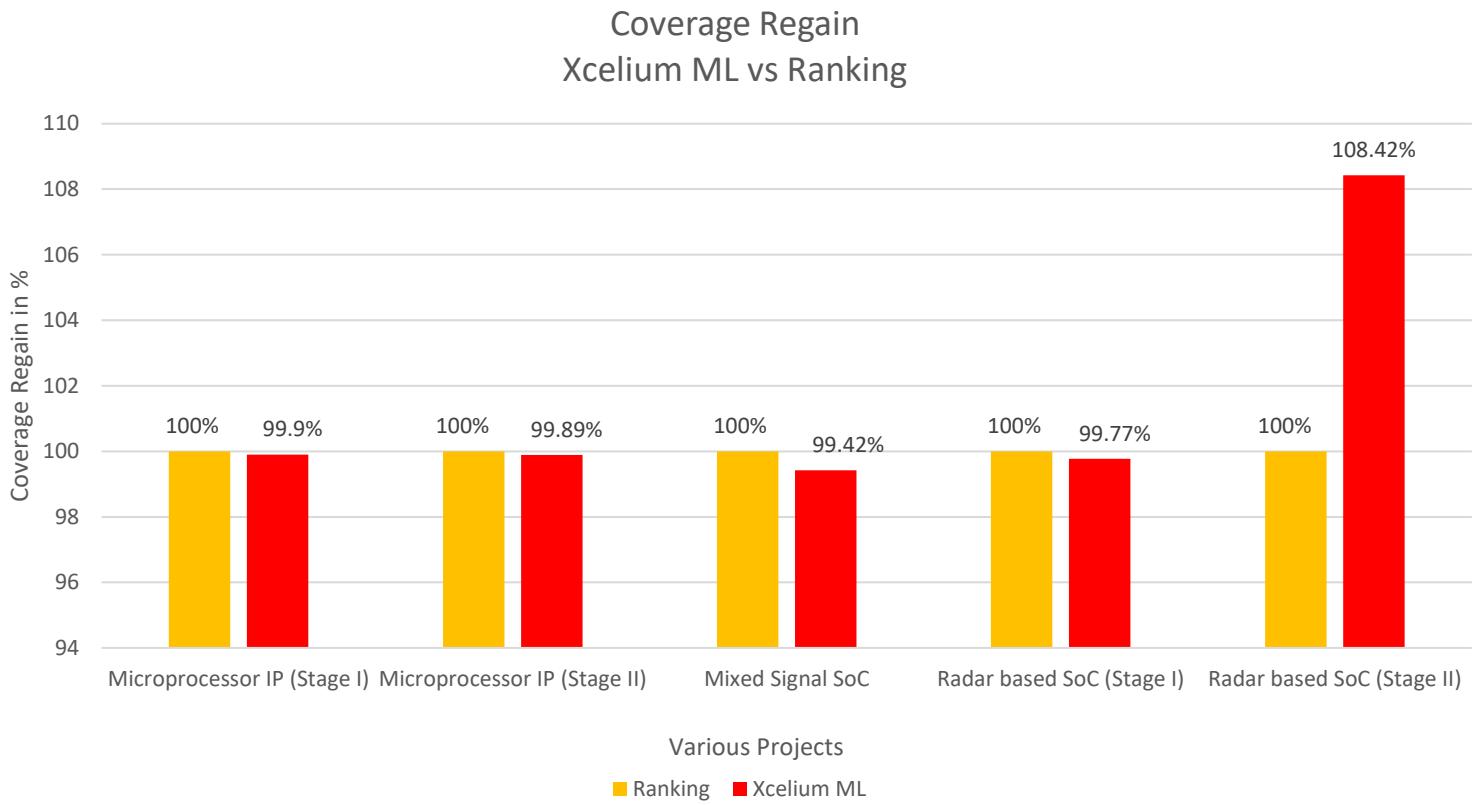
Xcelium ML

- ML tries to find the correlations between the actions inside the testbench and the coverage results
- The produced regression by ML is randomized in nature
- It can improve/ increase the original coverage
- It can discover new functional bugs that the orig regr did not find

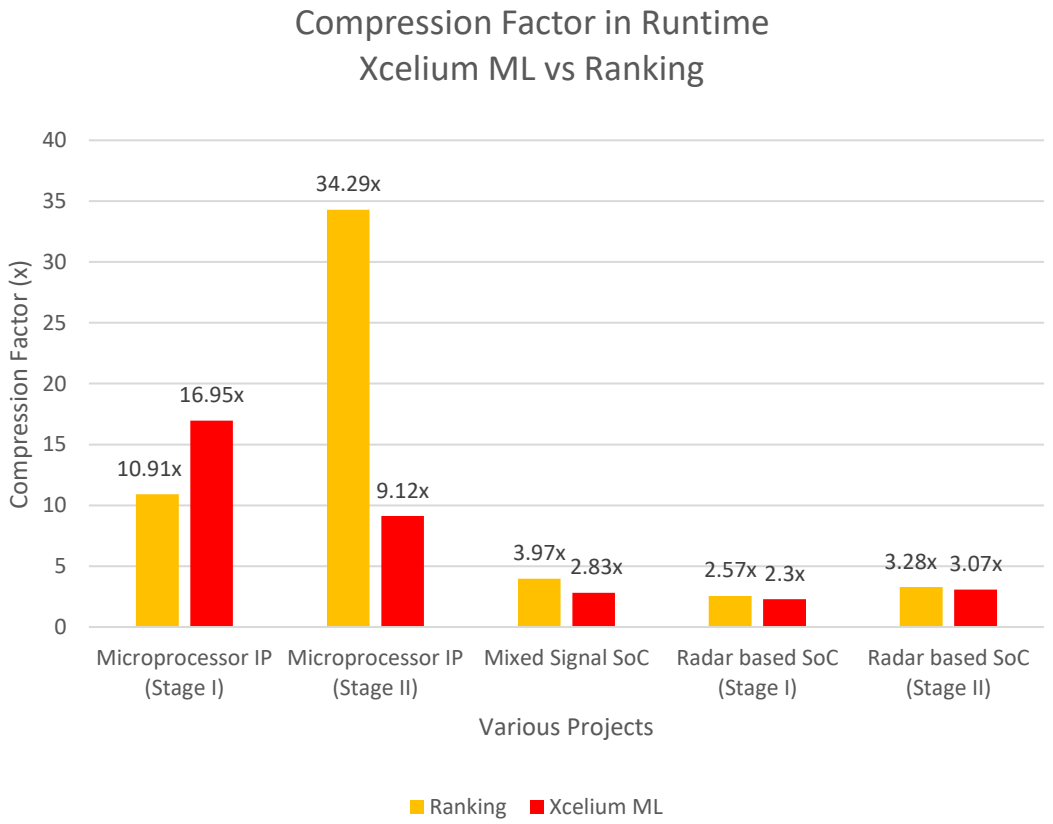
Ranking

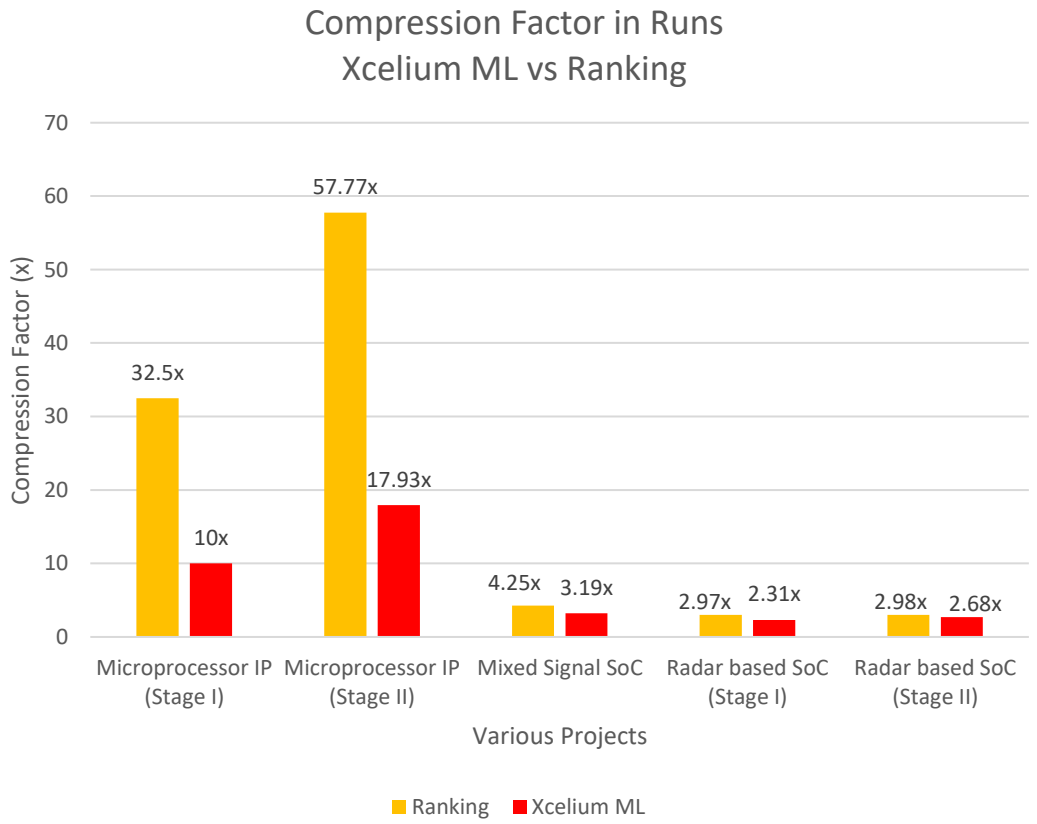
- Ranking finds the effective test and seed pairs to reach the same coverage
- Re-simulation of tests → Directed testing
- Inability to exercise new scenarios
  - No new coverage
  - Can not identify the bugs

Xcelium ML vs Ranking



Xcelium ML vs Ranking

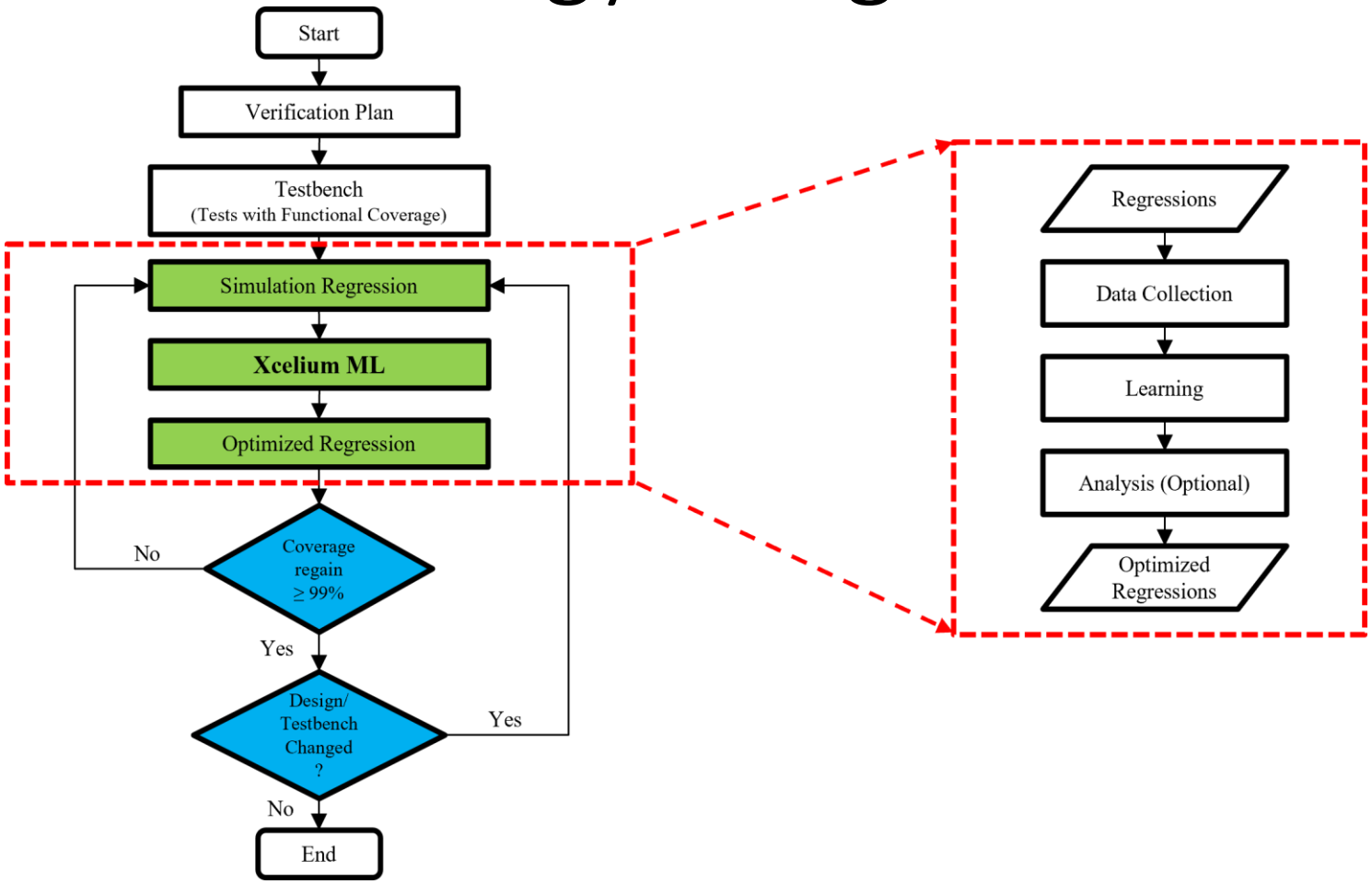




Agenda

- Motivation and Problem Statement
- Cadence Xcelium ML
- Application on Designs
- Xcelium ML vs Ranking
- Proposed Methodology
- Summary

Proposed Methodology using Xcelium ML



Agenda

- Motivation and Problem Statement
- Cadence Xcelium ML
- Application on Designs
- Xcelium ML vs Ranking
- Proposed Methodology
- Summary

Summary

- Out of the box solution with minimal setup
- Time & cost savings during simulation regression
- 3x to 15x speed-up factor
- Generates a compressed regression with random seeds (unlike Ranking)
  - Uncover new functional bugs
  - Might exceed original coverage
- ML learning/ training consumes extra 20% (on average) of the original regression time to produce an optimized regression
- Re-learning required after significant RTL/ TB changes (typically every week)

Thank you.

accellera
SYSTEMS INITIATIVE