NYU

VeriGuard AI-Driven Detection of Silent Verification Escapes

Cognichip Hackathon 2026

TEAM MEMBERS:

Makshi Baskaran

Tejas Attarde

Princy Doshi

Aviraj Dongare

GitHub - https://github.com/avirajdongare/VeriGuard-AI

Problem Statement & Motivation

Modern silicon failures are often not caused by missing testbenches, but by silent verification escapes. These escapes occur when RTL successfully passes simulation while still permitting unsafe, unintended, or untested behaviors. Even when coverage metrics report 100%, they only reflect what has been exercised, not what is possible. As a result, critical corner cases and legal yet unsafe design states may remain unverified, creating latent risks that surface only after silicon deployment.

Motivation

- Traditional verification focuses on stimulus generation and coverage metrics.
- However, coverage only reflects exercised behavior, not the complete legal behavior space of the RTL.
- Testbenches often embed assumptions about how the design should operate, but these assumptions are rarely validated.
- As design complexity grows, manually identifying untested legal states becomes increasingly impractical.
- An automated, intent-aware verification approach is required to systematically expose hidden gaps before tape-out.

Proposed Solution

- **VeriGuard Overview:** VeriGuard is an LLM powered verification gap analyzer built using the Cognichip platform.

- **Core Capability:** The system reads RTL and its corresponding testbench, infers design intent, identifies unverified behaviors, and generates targeted assertions and stress tests.

- **Key Focus:** The emphasis is on intent aware verification rather than stimulus only coverage.

Why Cognichip-Style AI is Necessary?

We need intent-aware intelligence that can:

* Infer design intent directly from RTL
* Detect hidden testbench assumptions
* Identify spec–RTL mismatches
* Expose untested legal behaviors
* Generate targeted verification artifacts

Key Insight:
Silent verification escapes are invisible to coverage metrics. Only intent-aware reasoning can systematically eliminate them.

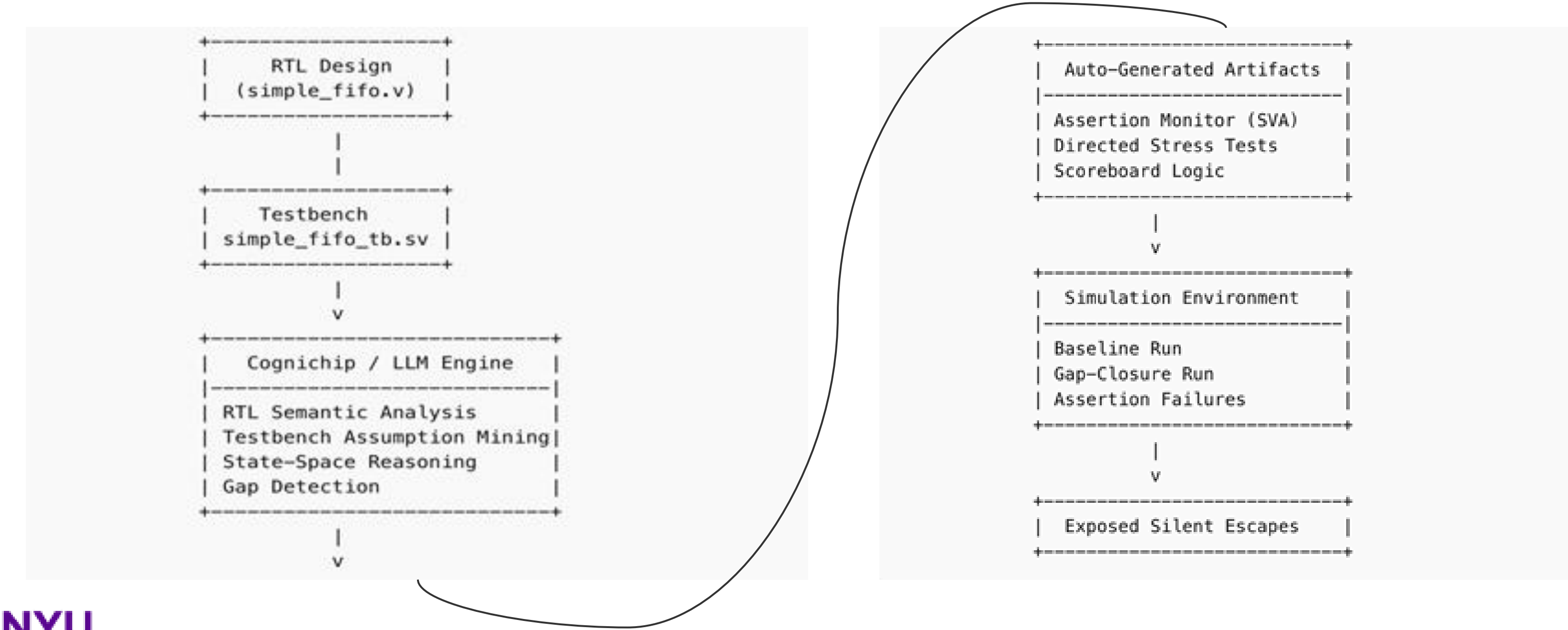
Design Methodology

Closed-Loop Refinement Flow

RTL + TB
→ Intent Analysis
→ Gap Detection
→ Assertion Generation
→ Simulation Validation

Hidden assumptions are converted into executable checks. The system reasons about what can happen, not just what did happen.

Design Methodology - System Workflow Diagram



```
+--------------------+
|     RTL Design     |
|  (simple_fifo.v)   |
+--------------------+
          |
          |
+--------------------+
|     Testbench      |
| simple_fifo_tb.sv  |
+--------------------+
          |
          v
+-----------------------------+
|   Cognichip / LLM Engine    |
|-----------------------------|
| RTL Semantic Analysis       |
| Testbench Assumption Mining|
| State-Space Reasoning       |
| Gap Detection               |
+-----------------------------+
          |
          v
```

```
+------------------------------+
|  Auto-Generated Artifacts    |
|------------------------------|
| Assertion Monitor (SVA)      |
| Directed Stress Tests        |
| Scoreboard Logic             |
+------------------------------+
             |
             v
+------------------------------+
|  Simulation Environment      |
|------------------------------|
| Baseline Run                 |
| Gap-Closure Run              |
| Assertion Failures           |
+------------------------------+
             |
             v
+------------------------------+
|  Exposed Silent Escapes      |
+------------------------------+
```

Architecture Under Test

Case Study: 4-Entry FIFO

- 4-Entry synchronous FIFO
- Memory array (depth = 4)
- Read / write pointers (3-bit)
- Occupancy counter
- FULL (Count == 4) and EMPTY flag (Count == 0)

Key Detail: No guard logic prevents overflow or underflow. The design is functionally correct ; but vulnerable if misused.

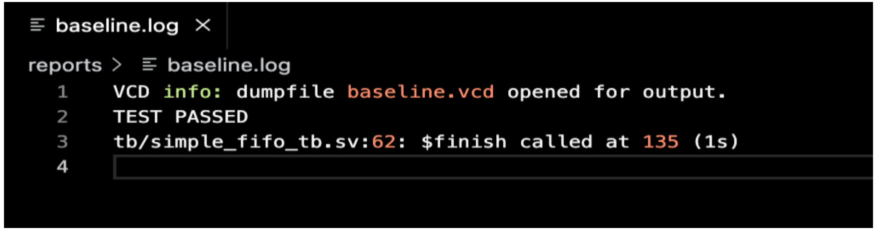
Baseline Simulation

- The original testbench performs a basic write sequence followed by a read sequence.
- It declares success once the expected flow completes.
- It does not verify data integrity or validate correctness of outputs.
- It does not stress overflow or underflow conditions.
- It does not validate status flag behavior.
- It does not test concurrent read and write scenarios.
- Simulation passes, but critical corner cases remain unverified.
- This results in a silent verification escape.

Simulation Logs

Outputs produced:

reports/baseline.log (text evidence)



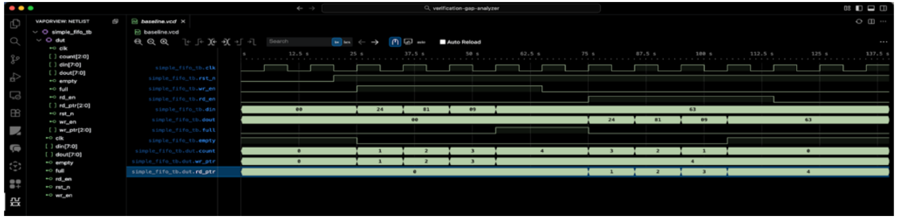
baseline.log ✕

reports > baseline.log

```
VCD info: dumpfile baseline.vcd opened for output.
TEST PASSED
tb/simple_fifo_tb.sv:62: $finish called at 135 (1s)

```

baseline.vcd

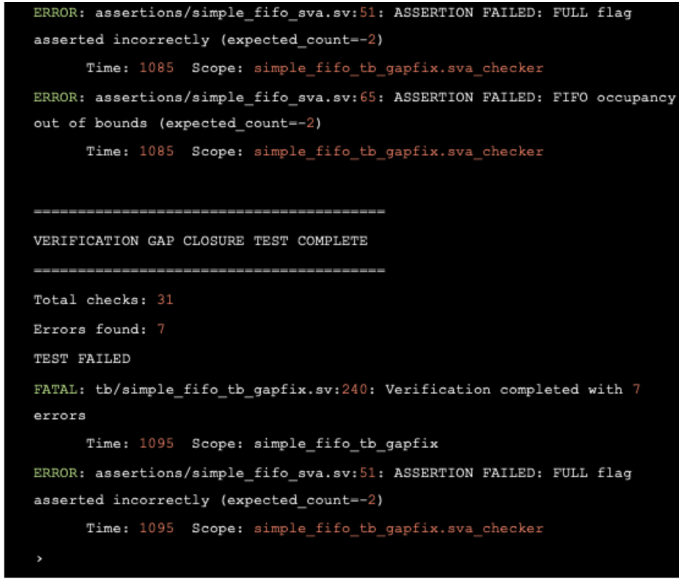
(waveform evidence)



Interpretation: Baseline TB passes, but it does not verify critical behaviors like overflow/underflow/data integrity and flag correctness—so bugs can escape silently.

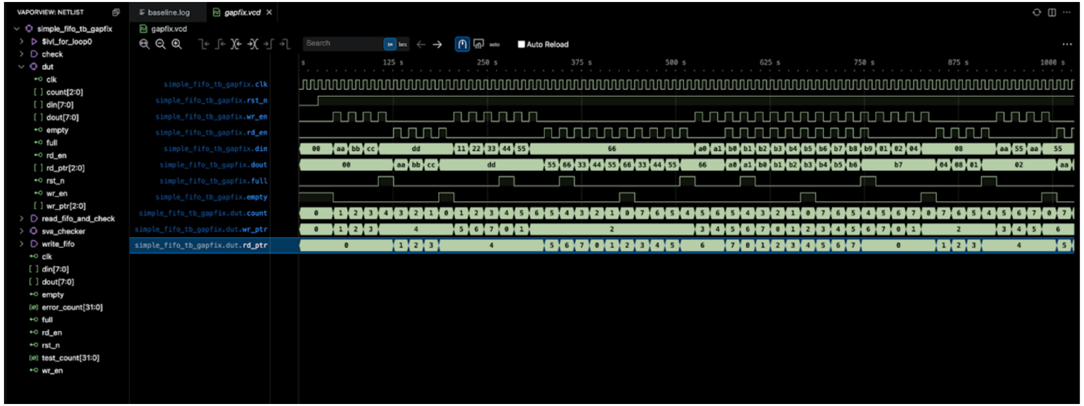
Gap-Closure Simulation (Exposes Bugs)

Outputs produced:
reports/gapfix.log (shows assertion/monitor failures)



```
ERROR: assertions/simple_fifo_sva.sv:51: ASSERTION FAILED: FULL flag
asserted incorrectly (expected_count=-2)
      Time: 1085  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy
out of bounds (expected_count=-2)
      Time: 1085  Scope: simple_fifo_tb_gapfix.sva_checker


========================================
VERIFICATION GAP CLOSURE TEST COMPLETE
========================================
Total checks: 31
Errors found: 7
TEST FAILED
FATAL: tb/simple_fifo_tb_gapfix.sv:240: Verification completed with 7
errors
      Time: 1095  Scope: simple_fifo_tb_gapfix
ERROR: assertions/simple_fifo_sva.sv:51: ASSERTION FAILED: FULL flag
asserted incorrectly (expected_count=-2)
      Time: 1095  Scope: simple_fifo_tb_gapfix.sva_checker
>
```

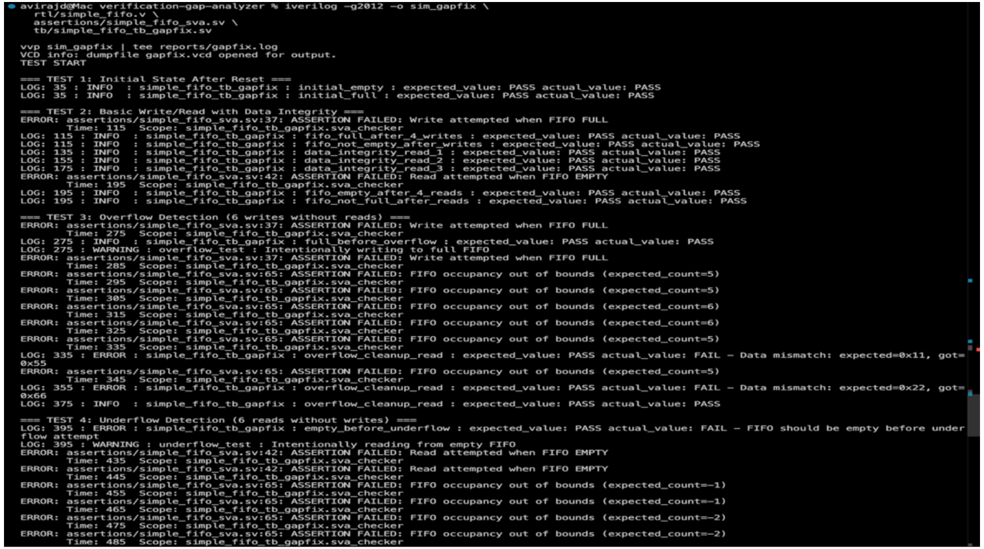
gapfix.vcd (waveform shows failing corner cases)



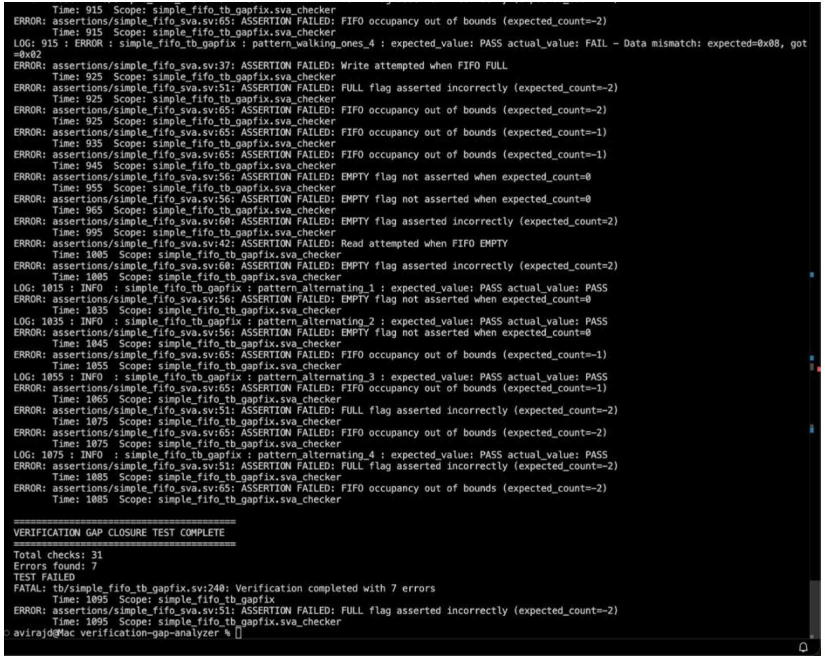
Interpretation: The gap-fix TB + checker actively tests/monitors missing behaviors. The simulation now fails with explicit errors, demonstrating that previously “passing” verification was incomplete.

Gap-Closure Simulation (Fix)

Outputs produced:



```
avirajd@Mac verification-gap-analyzer % iverilog -g2012 -o sim_gapfix \
  rtl/simple_fifo.v \
  assertions/simple_fifo_sva.sv \
  tb/simple_fifo_tb_gapfix.sv

vvp sim_gapfix | tee reports/gapfix.log
VCD info: dumpfile gapfix.vcd opened for output.
TEST START

=== TEST 1: Initial State After Reset ===
LOG: 35 : INFO  : simple_fifo_tb_gapfix : initial_empty : expected_value: PASS actual_value: PASS
LOG: 35 : INFO  : simple_fifo_tb_gapfix : initial_full : expected_value: PASS actual_value: PASS

=== TEST 2: Basic Write/Read with Data Integrity ===
ERROR: assertions/simple_fifo_sva.sv:37: ASSERTION FAILED: Write attempted when FIFO FULL
       Time: 115  Scope: simple_fifo_tb_gapfix.sva_checker
LOG: 115 : INFO  : simple_fifo_tb_gapfix : fifo_full_after_4_writes : expected_value: PASS actual_value: PASS
LOG: 115 : INFO  : simple_fifo_tb_gapfix : fifo_not_empty_after_writes : expected_value: PASS actual_value: PASS
LOG: 135 : INFO  : simple_fifo_tb_gapfix : data_integrity_read_1 : expected_value: PASS actual_value: PASS
LOG: 155 : INFO  : simple_fifo_tb_gapfix : data_integrity_read_2 : expected_value: PASS actual_value: PASS
LOG: 175 : INFO  : simple_fifo_tb_gapfix : data_integrity_read_3 : expected_value: PASS actual_value: PASS
ERROR: assertions/simple_fifo_sva.sv:42: ASSERTION FAILED: Read attempted when FIFO EMPTY
       Time: 195  Scope: simple_fifo_tb_gapfix.sva_checker
LOG: 195 : INFO  : simple_fifo_tb_gapfix : fifo_empty_after_4_reads : expected_value: PASS actual_value: PASS
LOG: 195 : INFO  : simple_fifo_tb_gapfix : fifo_not_full_after_reads : expected_value: PASS actual_value: PASS

=== TEST 3: Overflow Detection (6 writes without reads) ===
ERROR: assertions/simple_fifo_sva.sv:37: ASSERTION FAILED: Write attempted when FIFO FULL
       Time: 275  Scope: simple_fifo_tb_gapfix.sva_checker
LOG: 275 : INFO  : simple_fifo_tb_gapfix : full_before_overflow : expected_value: PASS actual_value: PASS
LOG: 275 : WARNING : overflow_test : Intentionally writing to full FIFO
ERROR: assertions/simple_fifo_sva.sv:37: ASSERTION FAILED: Write attempted when FIFO FULL
       Time: 285  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=5)
       Time: 295  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=5)
       Time: 305  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=6)
       Time: 315  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=6)
       Time: 325  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=5)
       Time: 335  Scope: simple_fifo_tb_gapfix.sva_checker
LOG: 335 : ERROR : simple_fifo_tb_gapfix : overflow_cleanup_read : expected_value: PASS actual_value: FAIL - Data mismatch: expected=0x11, got=
0x55
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=5)
       Time: 345  Scope: simple_fifo_tb_gapfix.sva_checker
LOG: 355 : ERROR : simple_fifo_tb_gapfix : overflow_cleanup_read : expected_value: PASS actual_value: FAIL - Data mismatch: expected=0x22, got=
0x66
LOG: 375 : INFO  : simple_fifo_tb_gapfix : overflow_cleanup_read : expected_value: PASS actual_value: PASS

=== TEST 4: Underflow Detection (6 reads without writes) ===
LOG: 395 : ERROR : simple_fifo_tb_gapfix : empty_before_underflow : expected_value: PASS actual_value: FAIL - FIFO should be empty before under
flow attempt
LOG: 395 : WARNING : underflow_test : Intentionally reading from empty FIFO
ERROR: assertions/simple_fifo_sva.sv:42: ASSERTION FAILED: Read attempted when FIFO EMPTY
       Time: 435  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:42: ASSERTION FAILED: Read attempted when FIFO EMPTY
       Time: 445  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=-1)
       Time: 455  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=-1)
       Time: 465  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=-2)
       Time: 475  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=-2)
       Time: 485  Scope: simple_fifo_tb_gapfix.sva_checker
```



```
       Time: 915  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=-2)
       Time: 915  Scope: simple_fifo_tb_gapfix.sva_checker
LOG: 915 : ERROR : simple_fifo_tb_gapfix : pattern_walking_ones_4 : expected_value: PASS actual_value: FAIL - Data mismatch: expected=0x08, got
=0x02
ERROR: assertions/simple_fifo_sva.sv:37: ASSERTION FAILED: Write attempted when FIFO FULL
       Time: 925  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:51: ASSERTION FAILED: FULL flag asserted incorrectly (expected_count=-2)
       Time: 925  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=-2)
       Time: 925  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=-1)
       Time: 935  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=-1)
       Time: 945  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:56: ASSERTION FAILED: EMPTY flag not asserted when expected_count=0
       Time: 955  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:56: ASSERTION FAILED: EMPTY flag not asserted when expected_count=0
       Time: 965  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:60: ASSERTION FAILED: EMPTY flag asserted incorrectly (expected_count=2)
       Time: 995  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:42: ASSERTION FAILED: Read attempted when FIFO EMPTY
       Time: 1005  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:60: ASSERTION FAILED: EMPTY flag asserted incorrectly (expected_count=2)
       Time: 1005  Scope: simple_fifo_tb_gapfix.sva_checker
LOG: 1015 : INFO  : simple_fifo_tb_gapfix : pattern_alternating_1 : expected_value: PASS actual_value: PASS
ERROR: assertions/simple_fifo_sva.sv:56: ASSERTION FAILED: EMPTY flag not asserted when expected_count=0
       Time: 1035  Scope: simple_fifo_tb_gapfix.sva_checker
LOG: 1035 : INFO  : simple_fifo_tb_gapfix : pattern_alternating_2 : expected_value: PASS actual_value: PASS
ERROR: assertions/simple_fifo_sva.sv:56: ASSERTION FAILED: EMPTY flag not asserted when expected_count=0
       Time: 1045  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=-1)
       Time: 1055  Scope: simple_fifo_tb_gapfix.sva_checker
LOG: 1055 : INFO  : simple_fifo_tb_gapfix : pattern_alternating_3 : expected_value: PASS actual_value: PASS
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=-1)
       Time: 1065  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:51: ASSERTION FAILED: FULL flag asserted incorrectly (expected_count=-2)
       Time: 1075  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=-2)
       Time: 1075  Scope: simple_fifo_tb_gapfix.sva_checker
LOG: 1075 : INFO  : simple_fifo_tb_gapfix : pattern_alternating_4 : expected_value: PASS actual_value: PASS
ERROR: assertions/simple_fifo_sva.sv:51: ASSERTION FAILED: FULL flag asserted incorrectly (expected_count=-2)
       Time: 1085  Scope: simple_fifo_tb_gapfix.sva_checker
ERROR: assertions/simple_fifo_sva.sv:65: ASSERTION FAILED: FIFO occupancy out of bounds (expected_count=-2)
       Time: 1085  Scope: simple_fifo_tb_gapfix.sva_checker

========================================
VERIFICATION GAP CLOSURE TEST COMPLETE
========================================
Total checks: 31
Errors found: 7
TEST FAILED
FATAL: tb/simple_fifo_tb_gapfix.sv:240: Verification completed with 7 errors
       Time: 1095  Scope: simple_fifo_tb_gapfix
ERROR: assertions/simple_fifo_sva.sv:51: ASSERTION FAILED: FULL flag asserted incorrectly (expected_count=-2)
       Time: 1095  Scope: simple_fifo_tb_gapfix.sva_checker
avirajd@Mac verification-gap-analyzer % 
```

Gap Detection

Cognichip-guided analysis identified:

- Overflow never exercised
- Underflow never exercised
- FULL/EMPTY flag inconsistency unverified
- Occupancy bounds unchecked
- Simultaneous R/W corruption possible
- Data ordering never validated

High-risk blind spots detected automatically.

Gap Closure

The system generated:

- ❖ Runtime assertion monitor
- ❖ Mirrored occupancy tracking
- ❖ Scoreboard-based data validation
- ❖ Directed overflow and underflow tests
- ❖ Simultaneous R/W stress scenarios

Intent gaps were transformed into executable verification logic.

Simulation Results

Baseline Run:
Simulation reports “TEST PASSED.”

Gap-Closure Run:

- ❖ Write attempted when FIFO full
- ❖ Read attempted when FIFO empty
- ❖ Occupancy counter out of bounds
- ❖ FULL/EMPTY flag inconsistencies
- ❖ Data mismatches detected

Previously hidden RTL vulnerabilities were exposed.

Industry Impact

- Silent verification escapes are a key cause of post-silicon failures.
- A single undetected RTL bug can lead to costly respins, schedule delays, and significant debug effort.
- Intent-aware AI-assisted verification improves pre-silicon confidence and reduces verification blind spots.
- The approach scales to complex SoCs, protocol verification, and advanced microarchitectural designs.

Performance Discussion

The objective was verification robustness rather than throughput optimization.
Additional overhead introduced by assertion monitoring and scoreboarding was minimal.
The trade-off strongly favors early detection compared to:

- ❖ Silicon respin costs
- ❖ Debug time
- ❖ Product schedule delays

Intent-aware verification improves tape-out confidence.

Challenges & Lessons Learned

Challenges

- ❖ Passing simulations can create false confidence
- ❖ RTL bugs may cascade across cycles
- ❖ Automated intent inference requires semantic reasoning

Lessons Learned:

- ❖ Passing tests do not guarantee correctness
- ❖ Verification must validate legal state space
- ❖ Directed stress testing is essential
- ❖ AI-assisted reasoning strengthens verification depth

Future Work

- Extend to multi-module SoC subsystems
- Integrate formal property synthesis
- Enable CI/CD-driven continuous verification
- Direct integration with Cognichip API

Thank You

NYU