

Development of a TC39x TriCore Instruction Set Simulator for AUTOSAR OS Conformance ELF Verification using AI-Assisted Methodology

Sunghoon Woo¹⁾ Daeki Kim¹⁾ Youngmo Lee¹⁾ Kyungnam Kim¹⁾ Jaekyung Kim¹⁾

*1) Hyundai Autoever, Test Engineering Team 1
12 Teheran-ro 113-gil, Gangnam-gu, Seoul, 06171, Republic of Korea(sunghoon_woo@hyundai-autoever.com)*

KEY WORDS: software and its underlying technologies; HILS, SILS; virtualization [E3]

1. Background

Verifying AUTOSAR software on the Infineon AURIX TC39x before physical hardware is available requires an Instruction Set Simulator (ISS) that accurately models both the processor ISA and AUTOSAR OS/BSW-dependent peripheral behavior. When an LLM builds such an ISS from scratch, it tends to generate forced code on its own — directly overwriting a test-result memory field rather than letting the OS compute it through genuine execution — to paper over gaps left by incomplete peripheral models or unimplemented OS APIs. Left uncorrected, this masks scheduling and interrupt-priority faults that would surface on real hardware, making the result unusable for safety-critical development under ISO 26262. This work investigates whether an AI-generated first-pass ISS can be evolved into a practical version with genuine, unforced conformance.

2. Method

We present a SystemC TLM-2.0 ISS for the AURIX TC39x six-core MCU, developed through an AI-assisted methodology in which an LLM served as a collaborative engineering partner. The ISS implements the TriCore v1.6.2P instruction set for six CPU cores with CSA chain management, a TLM-2.0 bus router, and 32 peripheral stubs. Starting from the AI's own first-pass output — reliant on forced code to satisfy a fixed conformance test suite — a 42-stage binary-search protocol systematically eliminated forced code, replacing it with genuine OS-driven paths such as a synthesized Boot ROM dispatch table.

3. Key Results

The ISS achieves 896 of 1,282 genuine PASS (69.9%) without any forced code — a 967% improvement over the initial 7.2% baseline. A complete 28-module AUTOSAR BSW stack is verified through 34 integration phases (ALL PASS), and a production ECU ROM runs crash-free for 3,600 s at a 1.07% real-time ratio, independently reproduced crash-free on a second host (RT=4.94%, confirming host-independent stability). TRACE32 JTAG cross-comparison confirms ~100% OS scheduling fidelity to physical hardware. Table 1 and Fig. 1 summarize key results.

Summarize Key Results:						
Metric	Result					
Genuine PASS (strict mode)	896 / 1,282 (69.9%)					
BSW integration phases	34 (ALL PASS, 28 modules)					
Aging test (production ECU ROM)	Host	PC Spec	Duration	Crashes	RT Ratio	Status
	PC#1 (original)	Intel i7-1255U 16GB RAM	3,600 s	0	1.07%	PASS ✓
	PC#2 (reproduction)	Intel Ultra7 265K 64GB RAM	3,600 s	0	4.94%	PASS ✓

Table 1. Summary of key verification results.

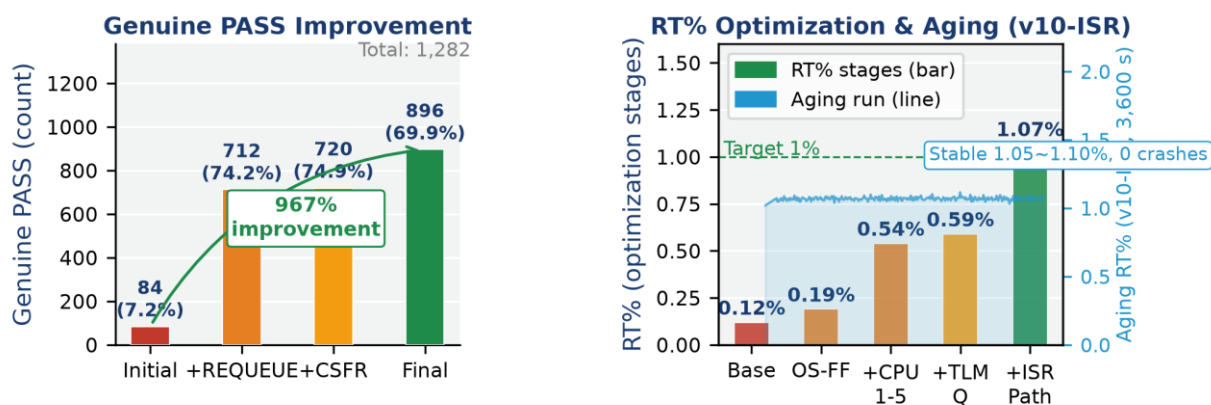


Fig. 1. (Left) Genuine PASS improvement across development stages. (Right) RT% optimization stages and 3,600 s aging stability (v10-ISR).