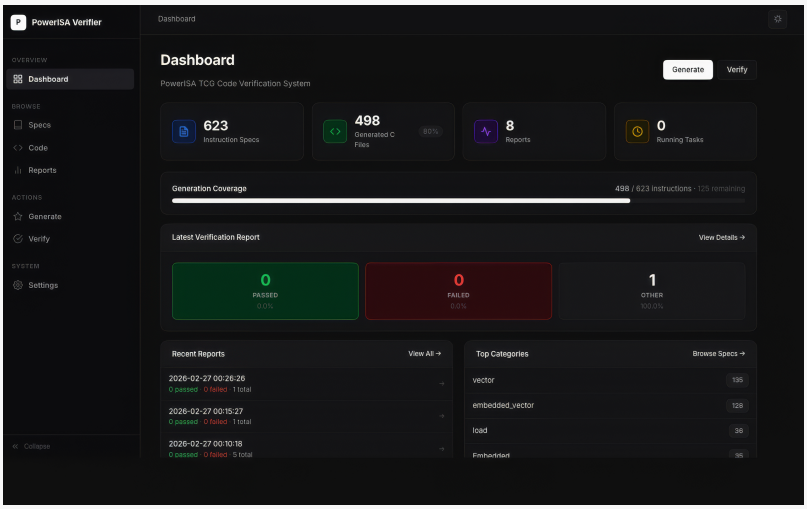
COONTEC and Soongsil University Are Developing LLM-Based Power ISA Verification Pipeline

Step-by-Step Integration from Structured Reference Data and LLM-Based Code Generation to Formal and Execution-Based Verification and Build Integration



PowerISA Verifier Dashboard for LLM-based Power ISA code generation and verification

SEONGNAM, GYEONGGI-DO, SOUTH KOREA, August 11, 2026 /EINPresswire.com/ -- COONTEC, a specialist in embedded virtualization and converged security, announced that it is jointly developing a Power Instruction Set Architecture (ISA) Verification System with Soongsil University. The system is designed to structure Power ISA specifications through an LLM-based, multi-stage generation and verification pipeline, generate candidate Instruction Set Simulator (ISS) C code for a virtual machine, and perform step-by-step verification.

Emulator development requires interpreting extensive ISA documentation and implementing the meaning and behavior of instructions in code. Errors introduced while interpreting specifications can propagate into code generation, compilation, and execution results. The research team is therefore focused on connecting source specifications, structured data, generated code, and verification results into a single traceable workflow.

Structured Ground Truth and LLM-Based Code Generation

The pipeline extracts instruction definitions and key information from the Power ISA manual and organizes them into a structured Ground Truth database. It includes mnemonics, opcodes, operands, bit fields, pseudocode, special registers, flags, and instruction descriptions needed for code generation and verification. The database serves as a reference layer linking the original specifications to the generated implementation.

Using retrieval-augmented generation (RAG), the LLM references both the structured database and source documents to generate candidate ISS C code for the virtual machine. Combining

structured information with the original context helps reduce generation errors and improve consistency between specification and implementation. The generated code then proceeds through separate review and verification stages.
A web-based dashboard supports specification lookup, generated source code review, verification tasks, and result checking, allowing users to trace how each instruction moves through the pipeline.

Formal and Execution-Based Verification and Build Integration

To confirm that the generated code matches the source specifications, the research team applies formal verification based on the Z3 SMT solver. For instructions that can be handled through symbolic reasoning, mathematical representations are constructed from both the specification and the generated code, and equivalence is compared across operation results, state information, conditional processing, and bit-field behavior.
Instructions that are difficult to represent directly in formal terms are supplemented with execution-based verification using representative inputs and trusted software references. Generated-code results are compared with reference results to determine whether behavior matches.
The research team is integrating the generated code into the virtual machine source tree and compiling it to identify syntax errors, ISS API compatibility issues, and type and dependency problems. PowerPC build integration is currently in progress, while full end-to-end verification is classified as a subsequent development phase.

Future Plans

The research team plans to improve the reliability of the structured reference data by cross-checking it against multiple independent machine-readable references, complete PowerPC build integration, and expand execution tracing so that register, memory, and state information can be compared at the instruction level.
Comparative verification using Golden Logs from physical reference hardware, trusted compilers and execution environments, and validated implementations will be pursued in a future phase. This is distinct from the formal verification and software-reference-based review methods currently in use. The team also plans to feed verification results back into code generation and support expert review when needed.
Joon Pang, CEO of COONTEC, said “This research is significant because it connects structured reference data, LLM-based virtual-machine ISS code generation, formal verification, and execution-based review into a single traceable pipeline. We are progressing with PowerPC build integration, and in later stages we plan to expand verification to physical hardware and Golden Logs to improve the efficiency of virtualization development and verification.”

About COONTEC Co., Ltd.

COONTEC is a cybersecurity company providing technologies and solutions across a wide range

of areas, including software supply chain security, AI security, infrastructure and OT security, embedded virtualization and security, and attack surface security. Based on technical expertise and project experience accumulated in the public, defense, financial, and manufacturing sectors, COONTEC supports customers in achieving secure digital transformation and business innovation. For more information, visit www.coontec.com.

Contact us
COONTEC
+82 31-751-9088
marketing@coontec.com
Visit us on social media:
LinkedIn
Instagram
Facebook
YouTube
X
