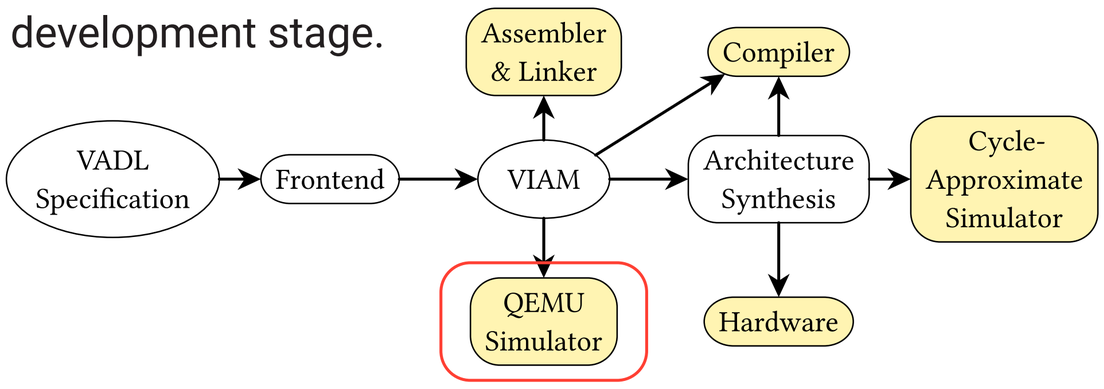
Generation of a QEMU-Based Instruction Set Simulator from a Processor Description in OpenVADL

Johannes Zottele, Matthias Raschhofer, Benedikt Huber, and Andreas Krall
Technische Universität Wien, Vienna, Austria

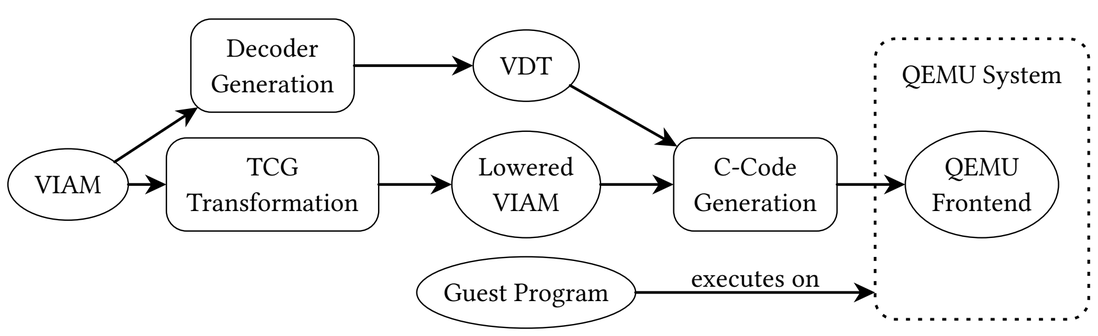
OpenVADL

OpenVADL is an open-source implementation of the **Vienna Architecture Description Language (VADL)**. From a VADL specification, it can generate various tools and artifacts, including an assembler, linker, compiler, and **instruction set simulator (ISS)**. Artifact generation based on architecture synthesis is still in an early development stage.

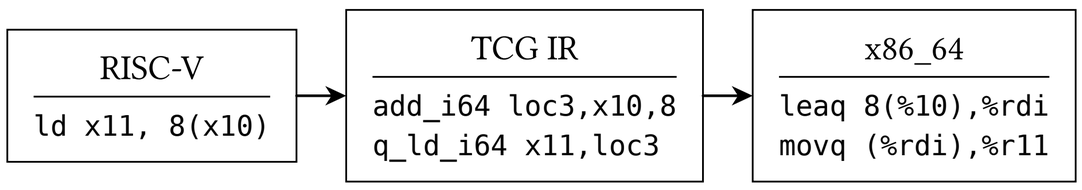


QEMU Generation

The ISS generator builds on QEMU, an open-source machine emulator that uses **Dynamic Binary Translation (DBT)** to emulate guest architectures.



OpenVADL generates the guest frontend, which translates guest instructions into **TCG operations** —QEMU's architecture-independent intermediate representation (IR). These are then compiled by the host backend into native host instructions.

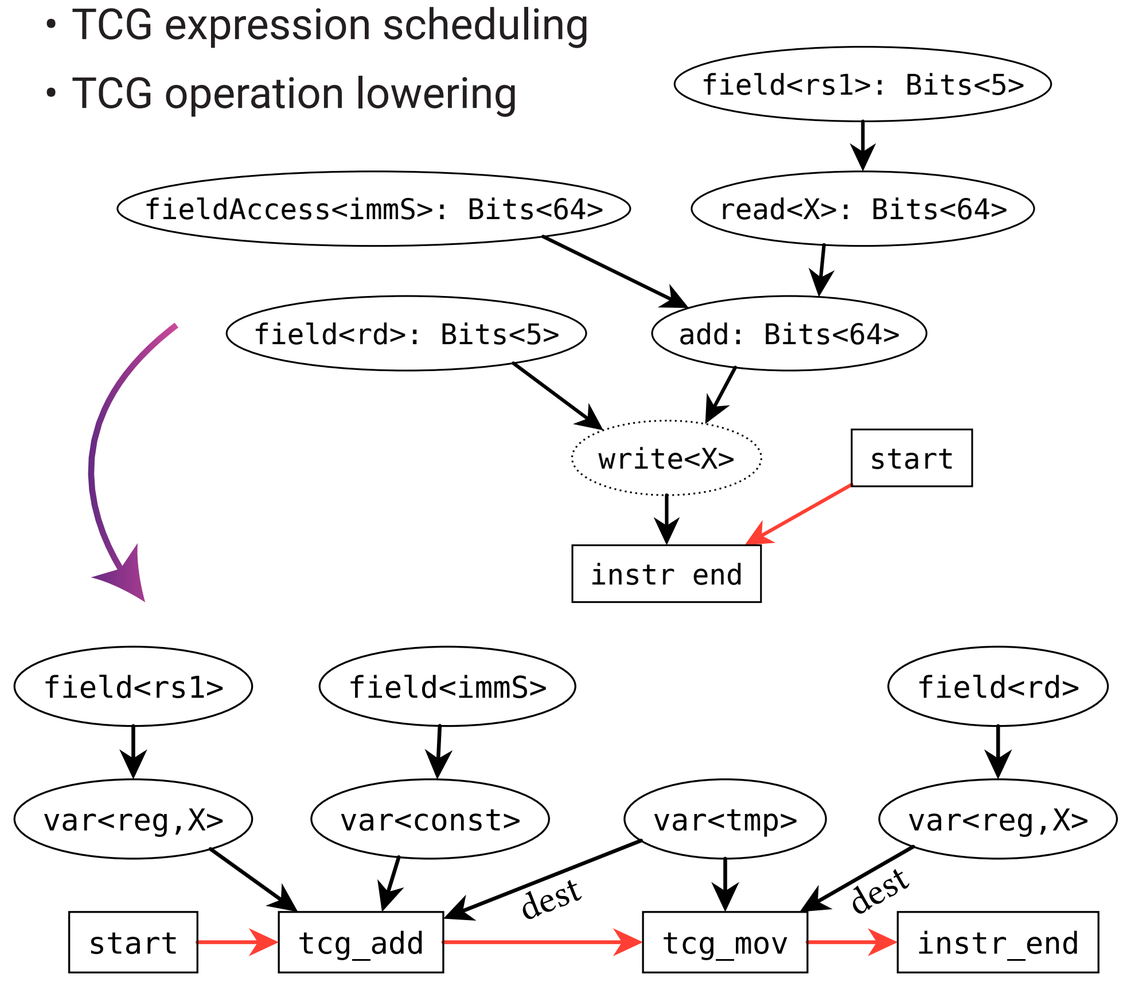


VIAM Transformation and Optimization

The VIAM is OpenVADL's intermediate representation, modeling instruction behavior as a multigraph that integrates a **control flow and dependency graph**. To convert a VADL instruction into TCG operations, this graph is transformed and optimized into a TCG-specific control flow graph suitable for direct C code generation.

Key transformations include:

- Normalization to the QEMU target size
- Side-effect scheduling
- TCG expression scheduling
- TCG operation lowering



Runtime Performance Evaluation

The **Embench benchmark suite** was used to evaluate the runtime performance of the generated QEMU against the upstream version **(baseline 1)**. Results show a runtime reduction of up to **44%** (RISC-V crc32). Further optimizations are expected to yield similar performance gains for AArch64.