Arbitrary and Variable Precision Floating-Point Arithmetic Support in Dynamic Binary Translation
Marie Badaroux Frédéric Pétrot
Univ. Grenoble Alpes, CNRS, Grenoble INP†, TIMA
†Institute of Engineering Univ. Grenoble Alpes

The big picture
• Fast processor centric digital system simulators → Instruction Set Architecture design choices evaluation → Playground for compiler backends for ISA extensions

The goal
• Arbitrary precision floating-point arithmetic support in DBT → Demonstrate feasibility → Propose a design and an implementation

Why?
• Limits of IEEE 754 Standard

Technical choices
• ISA: RISC-V, DBT engine: QEMU, AP library: MPFR

Overview

Background

Floating-point numbers: IEEE 754 Standard

1 bit e bits m bits

π ≈ 3.1415927410125732421875

DBT mechanism

Target code → TCG Intermediate Representation → Host code

Experiments

unit-tests
Test all AP instructions with random values e using Taylor expansion: \[ e = \sum_{k=0}^{n} \frac{1}{k!}. \]

cholesky
Cholesky decomposition of lower triangular matrices.

Arbitrary precision support in DBT

Our Extended Arbitrary Precision ISA Proposal
• 32 AP registers of virtually unlimited size
• 2 CSRs of 64-bit holding exponent size and precision
• 31 instructions largely inspired from F/D extensions

Simulation Strategy Design and Implementation
• Adding our new instructions as helpers into QEMU

Conclusion

• Operational support of arbitrary precision floating-point arithmetic in dynamic binary translation
• Some points are left to be addressed → Compiler does not target our AP extension → Use other libraries than MPFR
• https://github.com/fpetrot/qemu-ap

References


Figure 1. x86-64 and QEMU execution times for e

Table 1. Arbitrary Precision Instruction Encodings