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

Overview

The big picture

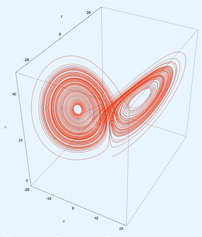
- Fast processor centric digital system simulators
  → Instruction Set Architecture design choices evaluation
  → Playground for compiler backends for ISA extensions

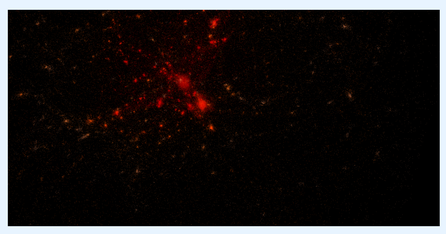
The goal

- 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**

Background

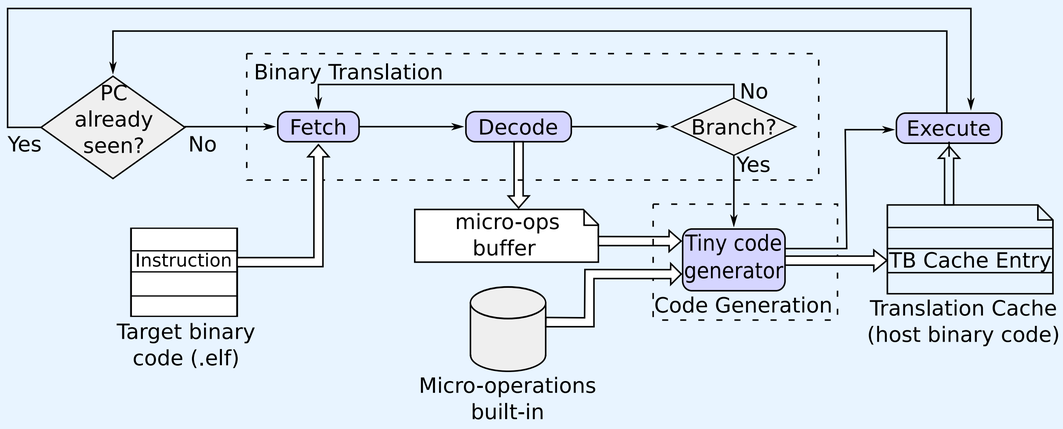
Floating-point numbers: IEEE 754 Standard

Sign	Exponent	Mantissa
1 bit	e bits	m bits

$$\pi \approx 3.14159274101257324218750$$

0 10000000 10010010000111111011011

DBT mechanism



Target code → TCG Intermediate Representation → Host code

Experiments

unit-tests

Test all AP instructions with random values

e

e using Taylor expansion: $e = \sum_{k=0}^{n} \frac{1}{k!}$.

cholesky

Cholesky decomposition of lower triangular matrices.

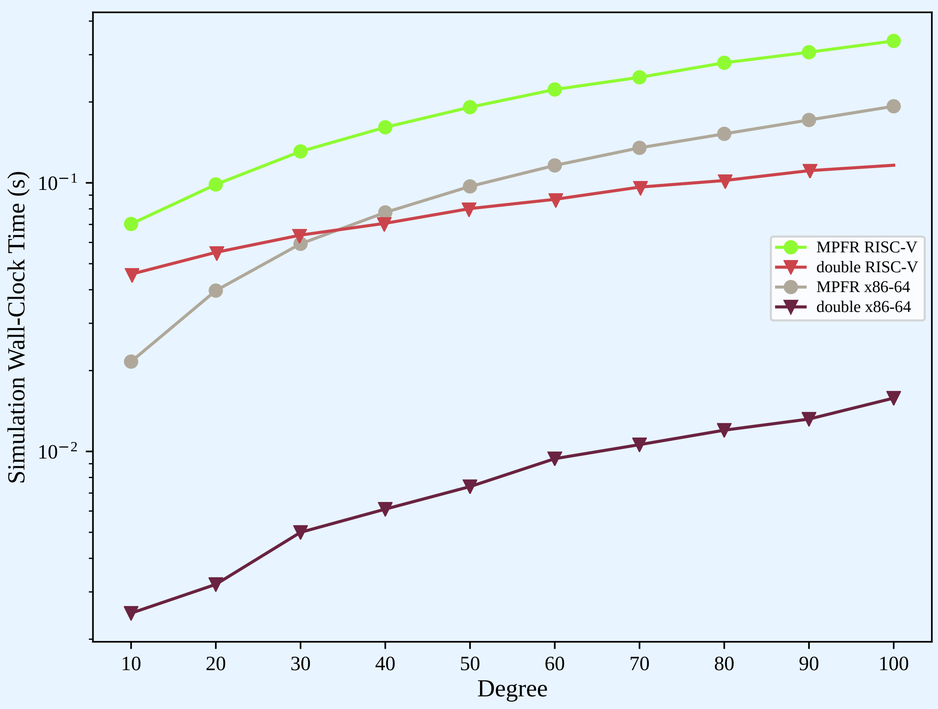


Figure 1. x86-64 and QEMU execution times for **e**

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

31 27	26 25	24 20	19 15	14 12	11 7	6 0	
imm[11:0]			rs1	101	rd	0000111	FLP
imm[11:5]		rs2	rs1	101	imm[4:0]	0100111	FSP
rs3	10	rs2	rs1	rm	rd	1000011	FMADD.P
rs3	10	rs2	rs1	rm	rd	1000111	FMSUB.P
rs3	10	rs2	rs1	rm	rd	1001011	FNMSUB.P
rs3	10	rs2	rs1	rm	rd	1001111	FNMADD.P
0000011		rs2	rs1	rm	rd	0001011	FADD.P
0000111		rs2	rs1	rm	rd	0001011	FSUB.P
0001011		rs2	rs1	rm	rd	0001011	FMUL.P
0001111		rs2	rs1	rm	rd	0001011	FDIV.P
0101111		00000	rs1	rm	rd	0001011	FSQRT.P
0010011		rs2	rs1	000	rd	0001011	FSGNJ.P
0010011		rs2	rs1	001	rd	0001011	FSGNJN.P
0010011		rs2	rs1	010	rd	0001011	FSGNJX.P
0010111		rs2	rs1	000	rd	0001011	FMIN.P
0010111		rs2	rs1	001	rd	0001011	FMAX.P
1010011		rs2	rs1	010	rd	0001011	FEQ.P
1010011		rs2	rs1	001	rd	0001011	FLT.P
1010011		rs2	rs1	000	rd	0001011	FLE.P
1100011		0----	rs1	rm	rd	0001011	FCVT.x.y

Table 1. Arbitrary Precision Instruction Encodings

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

M. Badaroux and F. Pétrot. "Arbitrary and Variable Precision Floating-Point Arithmetic Support in Dynamic Binary Translation". In: *2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC)*. 2021, pp. 325–330.