# Towards a High-Performance RISC-V Emulator

Dążąc do wydajnego emulatora RISC-V

Leandro Lupori Vanderson Martins do Rosario Edson Borin

March 1, 2021

## Contents

https://ieeexplore.ieee.org/document/8748908

# 1 Dążąc do wydajnego emulatora RISC-V

RISC-V jest otwartą architekturą procesora która zwróciła uwage całego swiata poprzez swój szybki rozwój i adaptacje. Jest ona juz wspierana przez GCC, Clang, i Linux. Ponadto niedawno pojawiło się wiele emulatorów i symulatorów RISC-V, niestety żaden z nich nie jest wystarczająco wydajny. Zazwyczaj, najczęściej stosowaną i najszybszą metodą implementacji takich emulatorów jest DBT - dynamiczna translacja maszynowa (*z ang.Dynamic Binary Translation*), która wymaga dobrej jakości translacji by uzyskać wysoką wydajność. W tej pracy zbadamy czy wysokiej jakości translacja kodu maszynowego RISC-V jest możliwa. W tym celu użylyliśmy SBT - statycznej translacji binarnej (*z ang. Static Binary Translation*) by sprawdzić jakość jaka może być uzyskana przy tłumaczeniu RISC-V na x86 i ARM. Nasze wyniki experymentalne pokazują że nasz SBT potrafi stworzyć wysokiej jakości kod, otrzymując tylko 12%/35% nizsza wydajnośc w porównaniu do kompilowanego kodu x86/ARM. Jest to lepszy wynik niż inne bardziej znane dynamiczne translatory RISC-V takie jak RV8 i QEMU. Ponieważ dynamiczna translacja jest silnie związana z jakościa translacji, nasz statyczny translator pokazuje że istnieją możliwości stworzenia bardziej wydajnych dynamicznych translatorów RISC-V niż obecnie istniejące

# 2 Spis treści

(b) GCC vs Clang i opcje operacji zmiennoprzecinkowych

(c) Konfiguracja RISC-V

7. Wyniki eksperymentów

   (a) GCC vs Clang

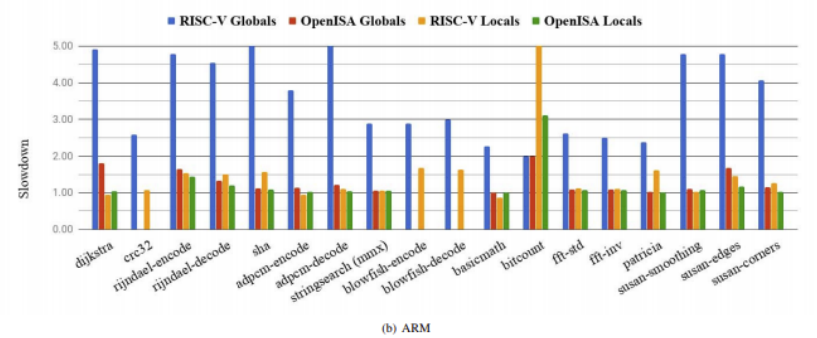
   (b) RISC-V vs openISA

   (c) Nasz SBT vs dostępne DBTv

8. Wnioski

# 3 Przedstawienie wyników

Wyniki sa przedstawione w postaci tabeli porównujących programy kompilowane na x86/ARM do programów statycznie przetłumaczonych za pomocą nowego SBT oraz istniejących translatorów (RV8 i QEMU). Porównywane są różne ustawienia kompilatorów i ich wpływ na wyniki oraz porównanie do translacji openISA




(b) ARM

Figure 4. Slowdown for our RISC-V SBT and for the OpenISA SBT. All binaries were compiled using the hard-float ABI.

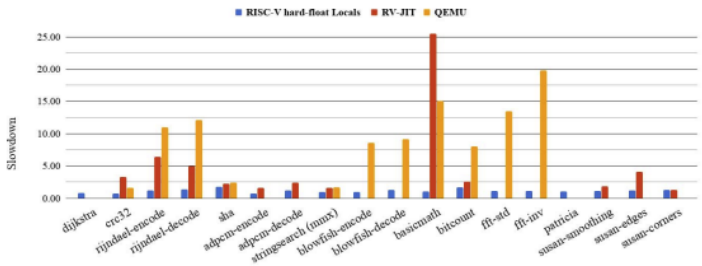



Figure 5. Slowdown comparison between our RISC-V SBT, RV8 DBT and QEMU-RISCV DBT. All tests emulating RISC-V binaries on a x86 processor.

# 4 Interpretacja wyników

Po porównaniu stworzonego silnika translacji statycznej stwierdzili że zbudowanie wydajnego emulatora RISC-V na sprzęcie dostępnym dzisiaj (W wiekszosci x86 i ARM). Oczekują rownież używanie tej otwartej architektury jako IR (kod binarny gotowy do translacji wynikowej)