Report 19.11.20

19/11/2020

Matteo Perotti
Luca Bertaccini
Pasquale Davide Schiavone
Stefan Mach

Professor Luca Benini
Integrated Systems Laboratory
ETH Zürich

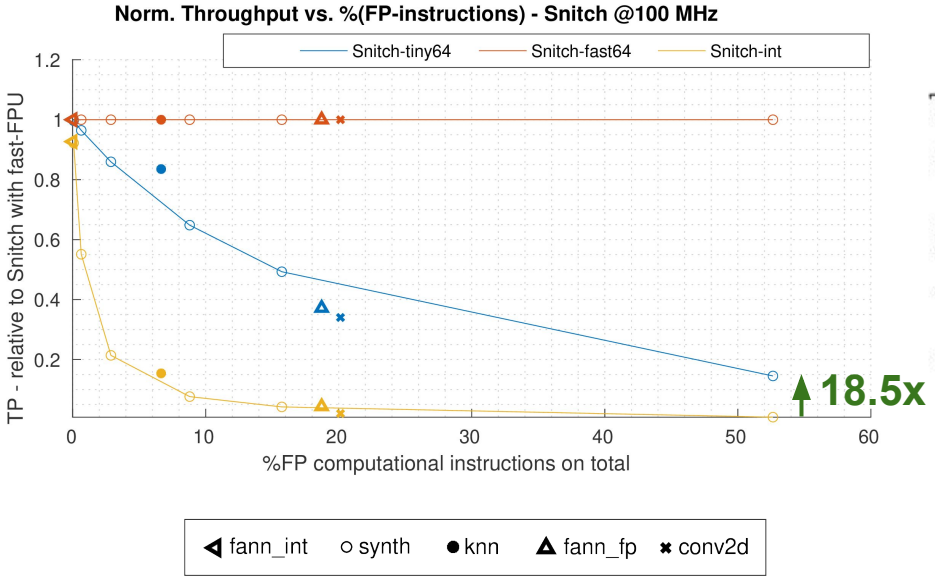
Summary

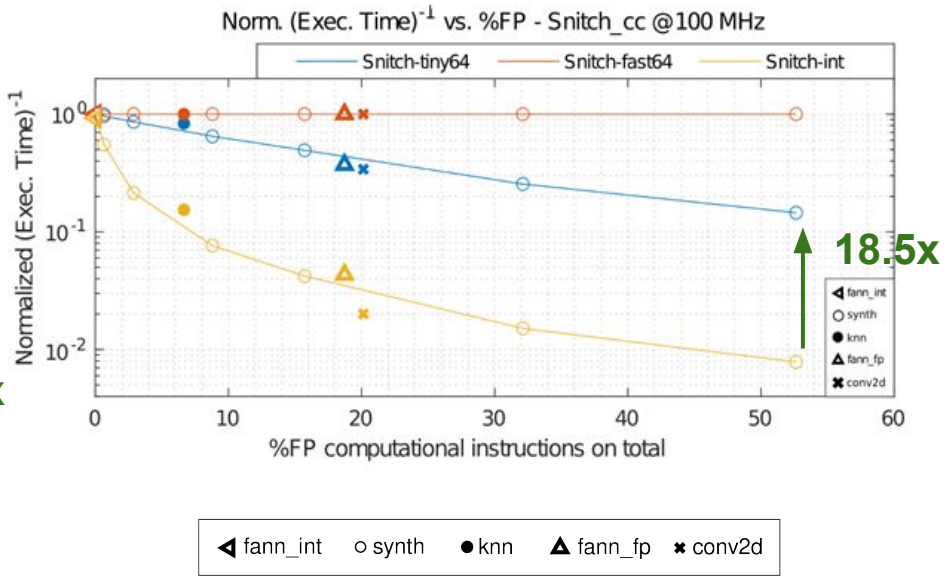
- TinyFPU - ISCAS paper
- Update on FP libraries

ISCAS Paper

- Deadline has been extended again
- The paper is ready
- New plots

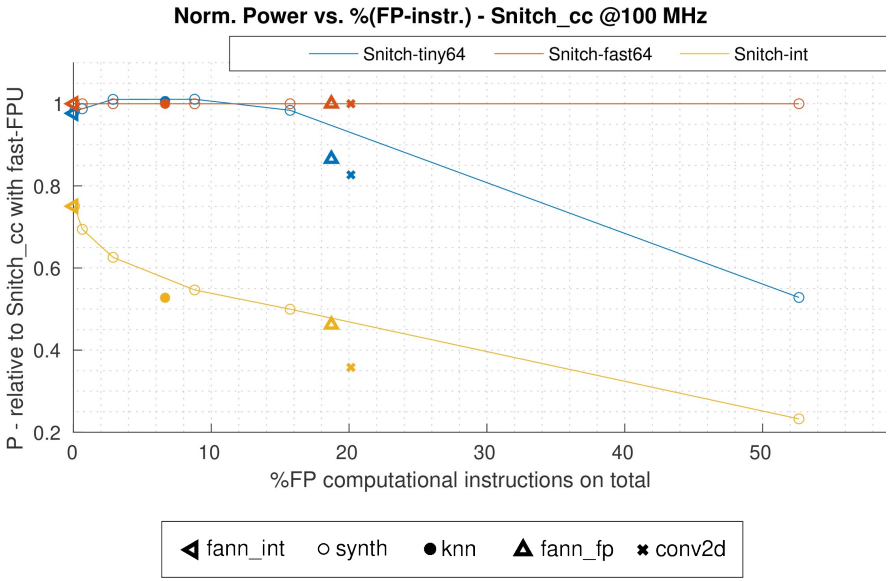
Performance - FP64

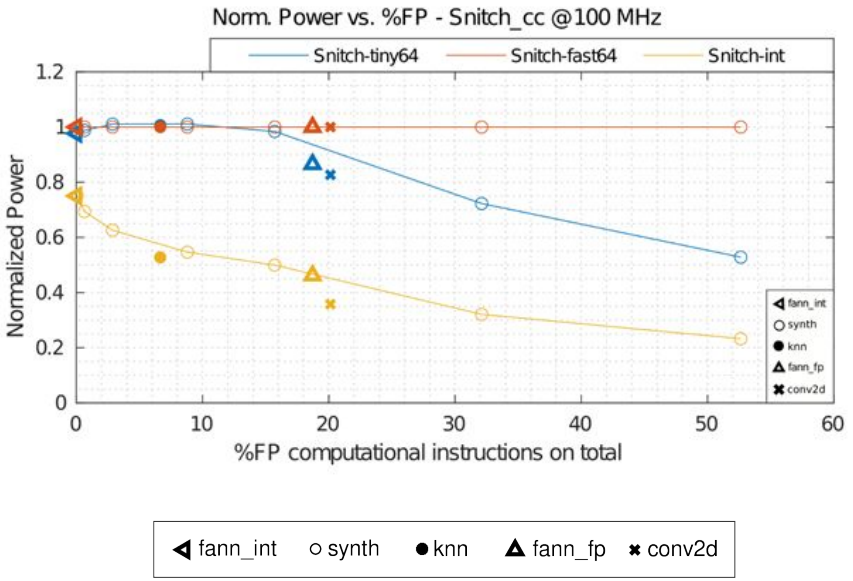




- Linear scale (y-axis)
- Logarithmic scale (y-axis)

Power - FP64

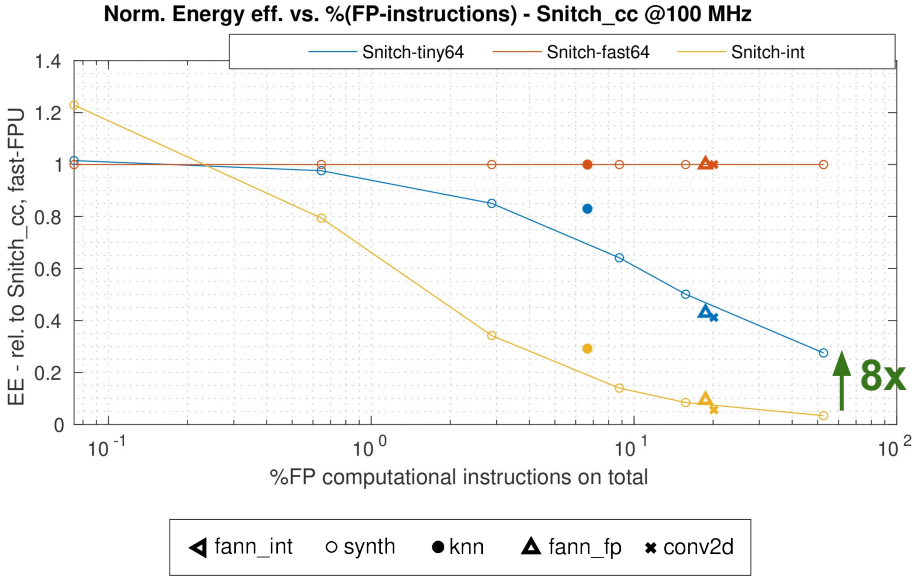


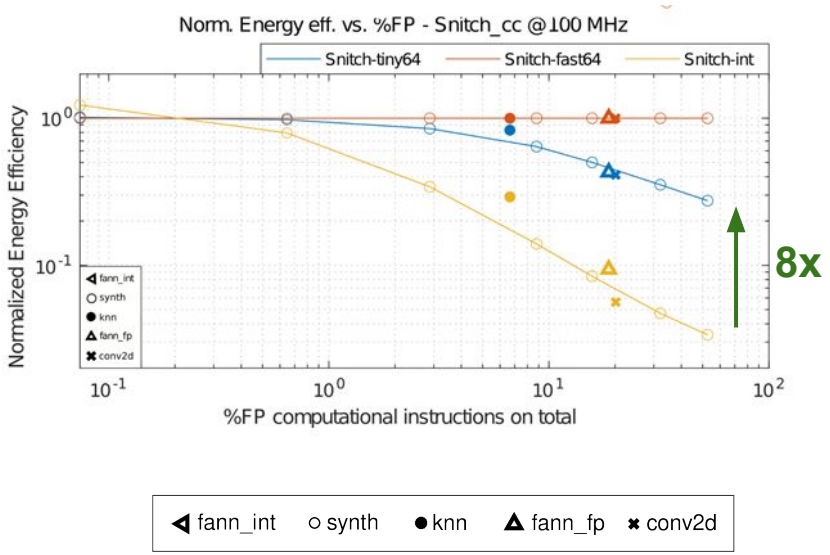


- Linear scale (y-axis)

- Linear scale (y-axis)

Energy Efficiency - FP64





- Linear scale (y-axis)
- Logarithmic scale (y-axis)

RISC-V Summit

- **Presentation**: Tuesday, 8 December 2020 11:00am - 11:20am PST (Pacific Standard Time, GMT-8)

https://tmt.knect365.com/risc-v-summit/agenda/1/#system-architectures_a-tiny-risc-v-floating-point-unit_11-00am

- Already **recorded**

Next Steps (1)

- **Zfinx** implementation of **Snitch + 32-bit TinyFPU** to achieve the **lowest area overhead** to support **FP** in **HW**
- Snitch has already some **external integer functional units** that need data contained in the **INT register file**
- There is already an infrastructure that we can adapt to our case (just two operands so far)

Next Steps (2)

- Measure libgcc's single functions performance on Snitch
- So far we measured the performance using entire algorithms

FP library

- **Completed** and **tested**: 18 different conversion functions
- To **test**: fast-addition32, div32
- To **write**: fast-addition64, div64

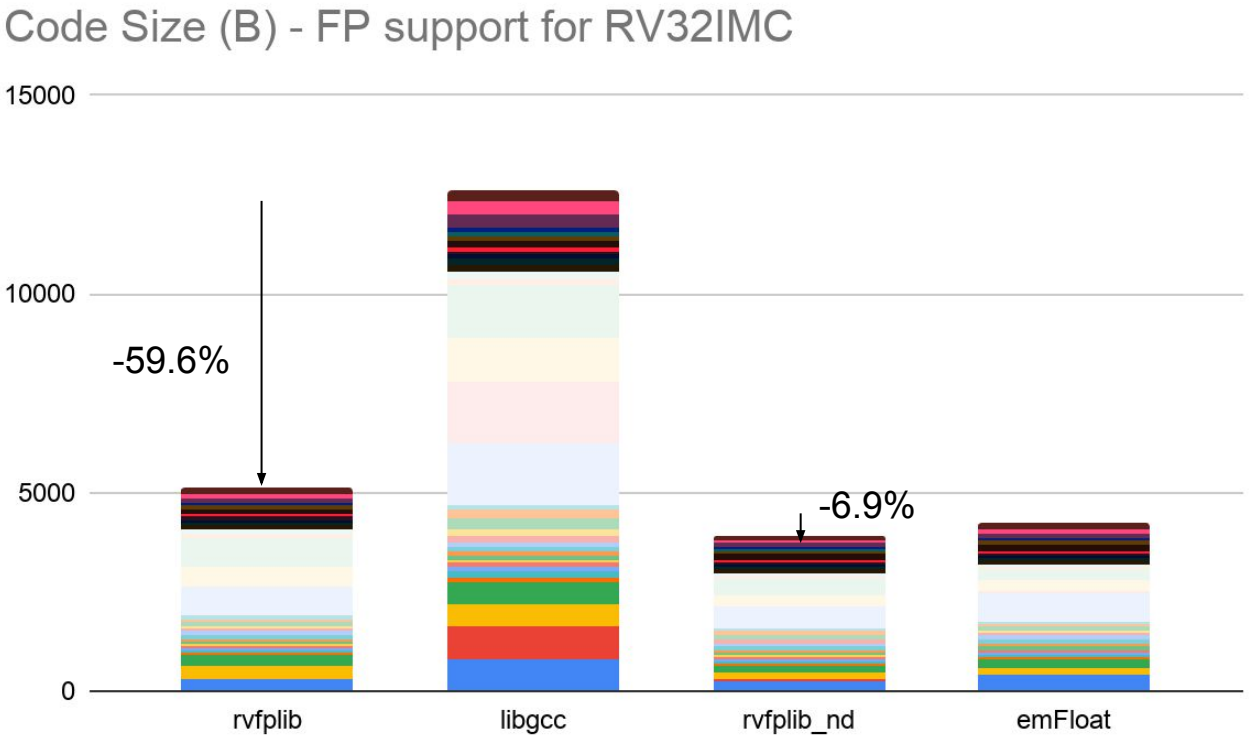
DAC paper

- **RVfplib**

- RVfplib **code size** vs libgcc and SEGGER
  - Single functions (RVfplib vs. libgcc && RVfplib_nd vs. SEGGER)
  - Benchmarks ".text+.rodata" (RVfplib vs. libgcc)

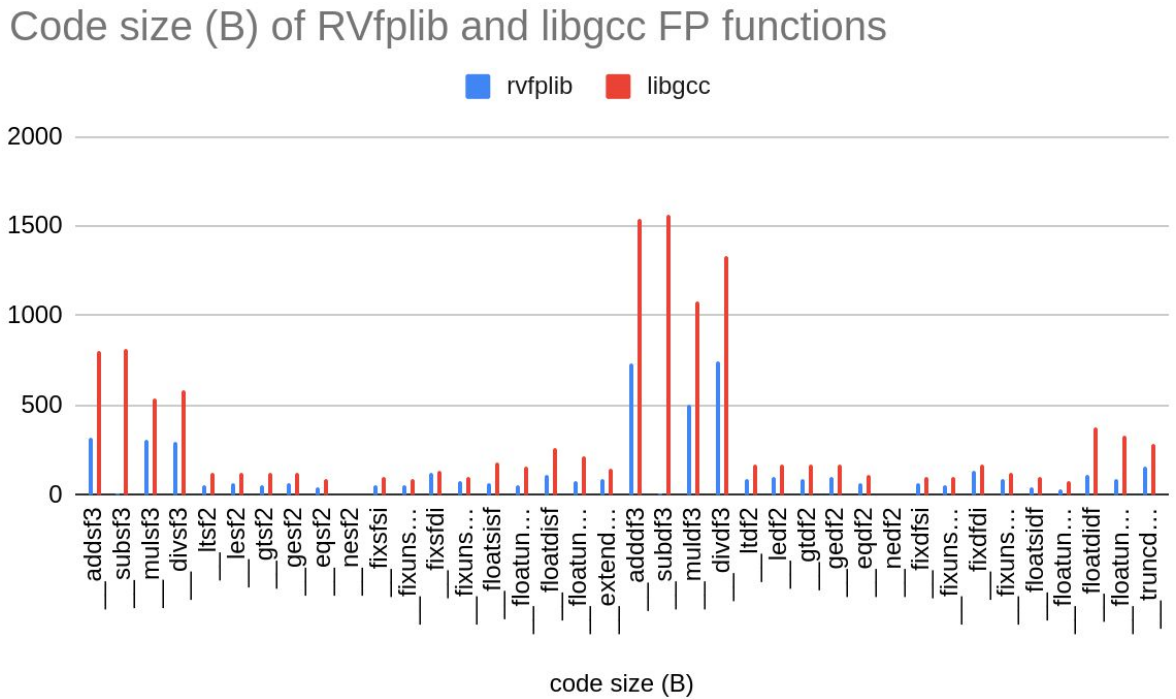
- RVfplib **execution latency** with SPIKE
  - Single latencies with CV32E40P (real processor)
  - Benchmarks with SPIKE (fast, CPI == 1)

Code Size - Libraries

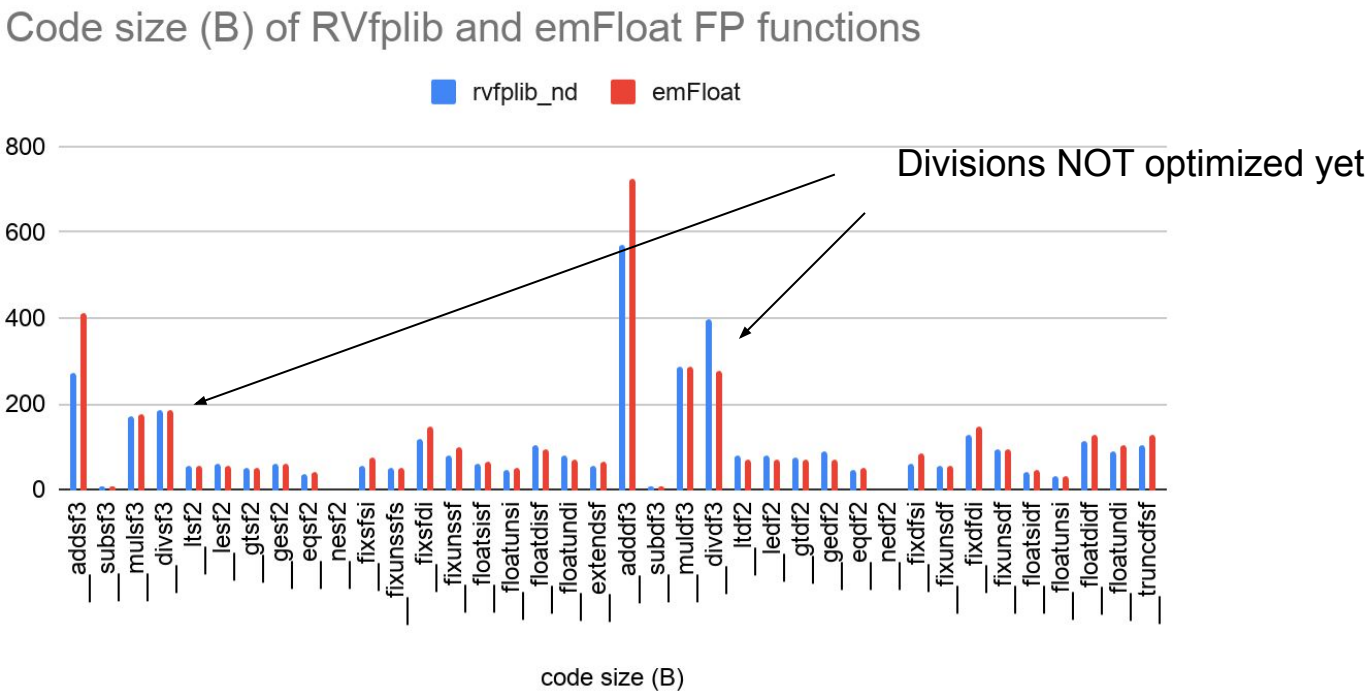


emFloat (SEGGER) code size data from
https://www.segger.com/products/development-tools/runtime-library/technology/floating-point-library/

Code Size - Libraries

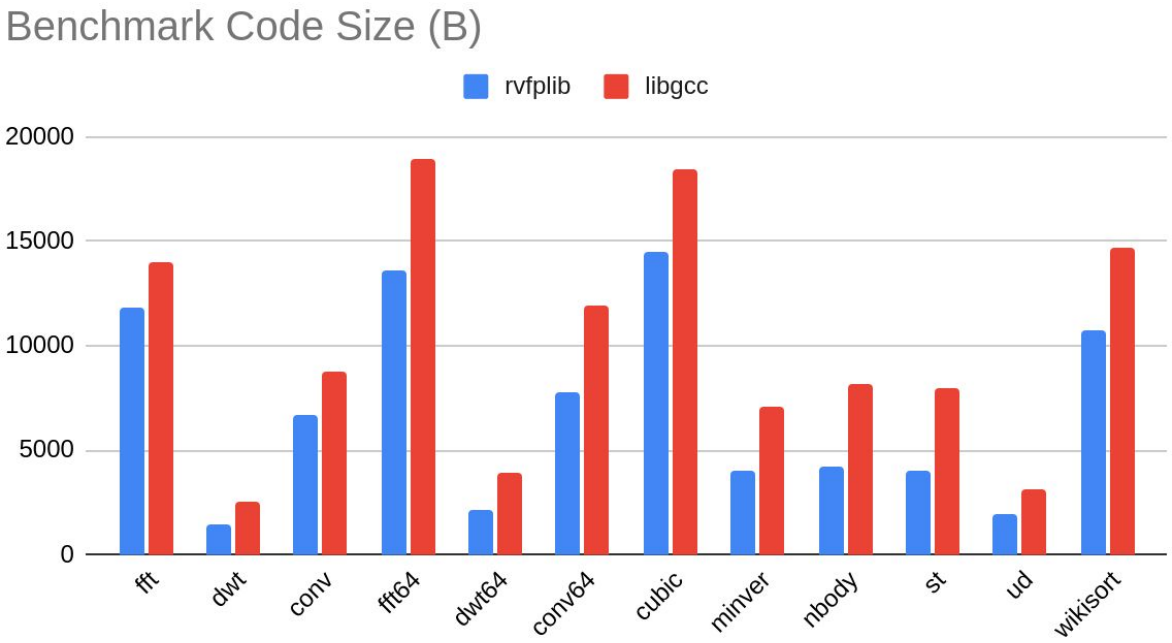


Code Size - Libraries



emFloat (SEGGER) code size data from
https://www.segger.com/products/development-tools/runtime-library/technology/floating-point-library/

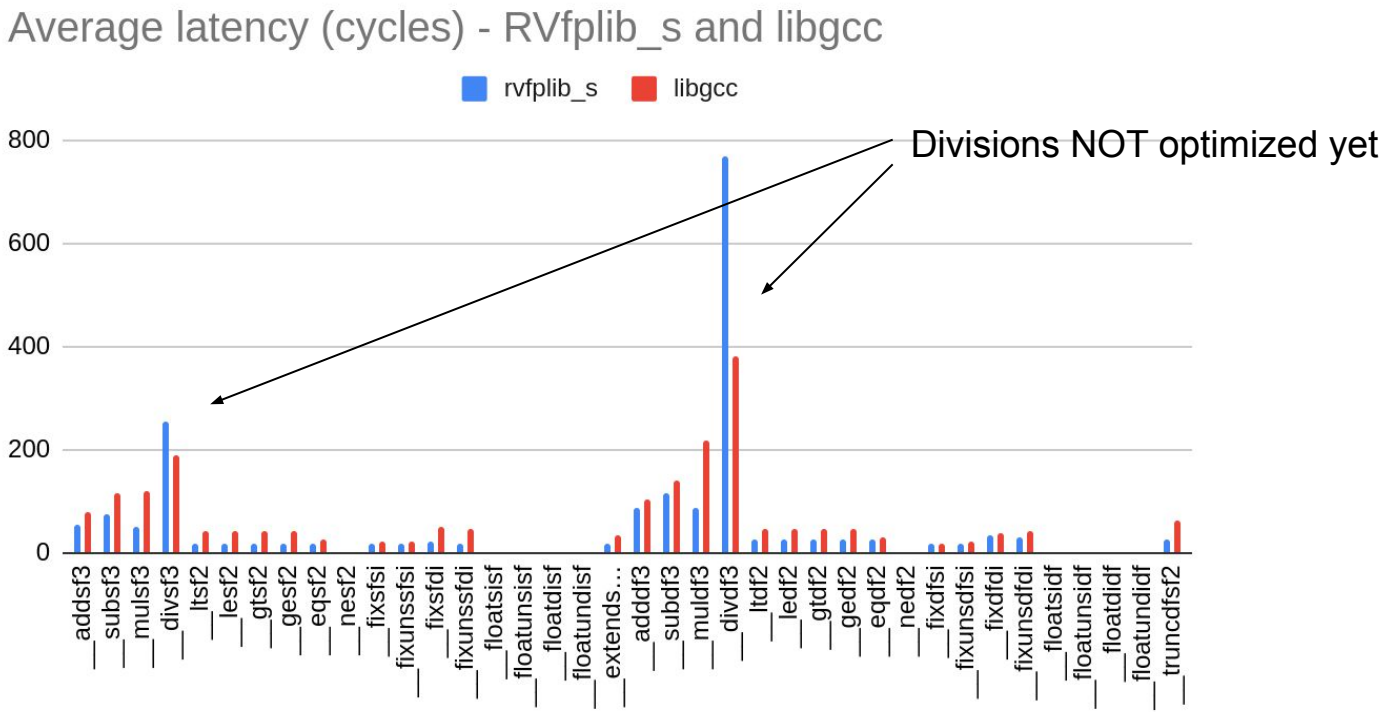
Code Size - Benchmarks



Avg. 35% lower code size (.text+.rodata)

Average Latency - Libraries

Avg speedup: 1.75x



Code Size - Benchmarks

Avg speedup: 1.51x