Report 05.11.20

05/11/2020

Matteo Perotti
Luca Bertaccini
Pasquale Davide Schiavone
Stefan Mach

Professor Luca Benini
Integrated Systems Laboratory
ETH Zürich

Summary

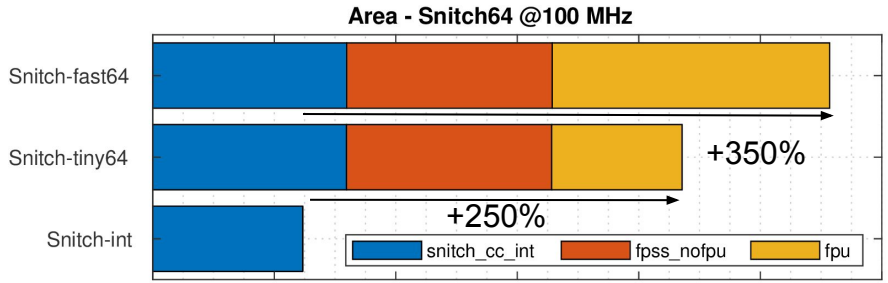
- TinyFPU - ISCAS paper
- Update on FP libraries

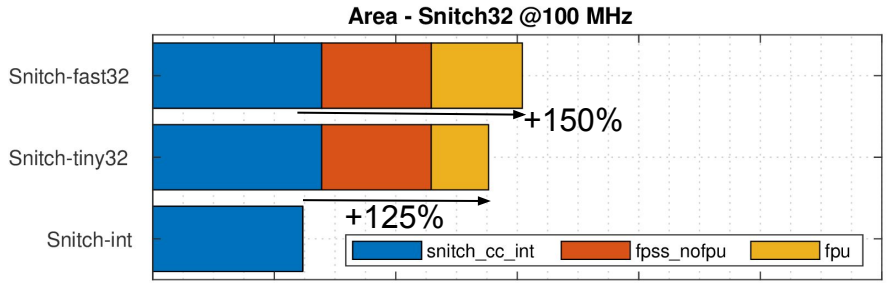
Benchmarks

Applications with different %FP instruction:

- (Artificial Benchmark) Parametric INT Matmul + FP Matmul:
  - 0.07% FP
  - 0.65% FP
  - 2.9%
  - 8.9%
  - 16%
  - 56%
- Other benchmarks:
  - fann (21%)
  - conv2d (20%)
  - knn (7%)
  - fixed_point fann (0%)

Area results





FPUs:

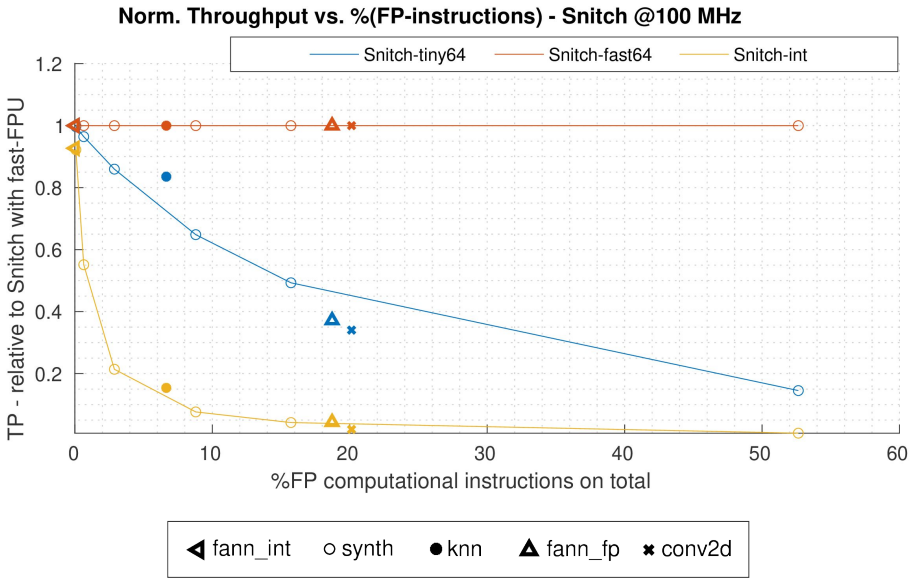
- Tiny32 is 37% smaller than Fast32
- Tiny64 is 53% smaller than Fast64

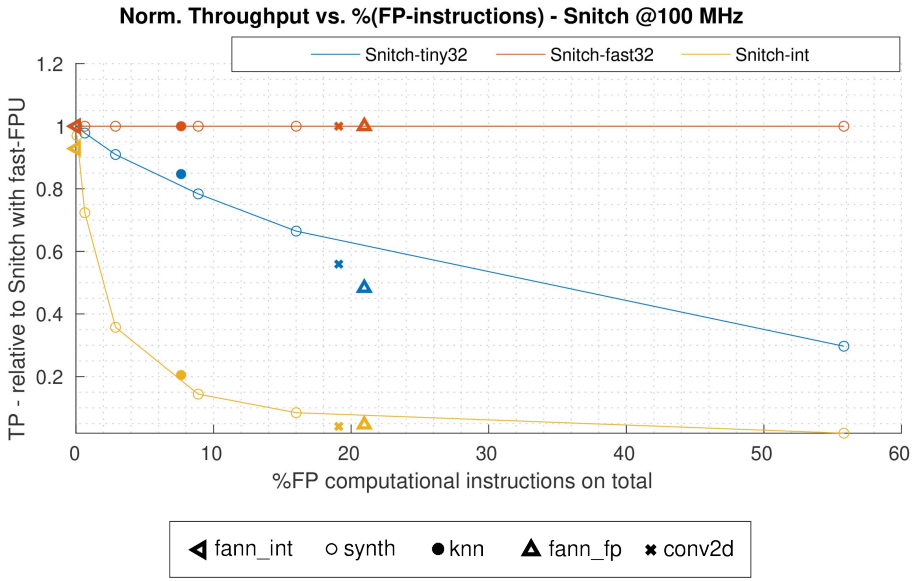
Core+FPU (at the moment):

- Snitch-tiny64 22% smaller than Snitch-fast64
- Snitch-tiny32 9% smaller than Snitch-fast32

FP register file occupies ~70% of the red area → Zfinx for minimum area

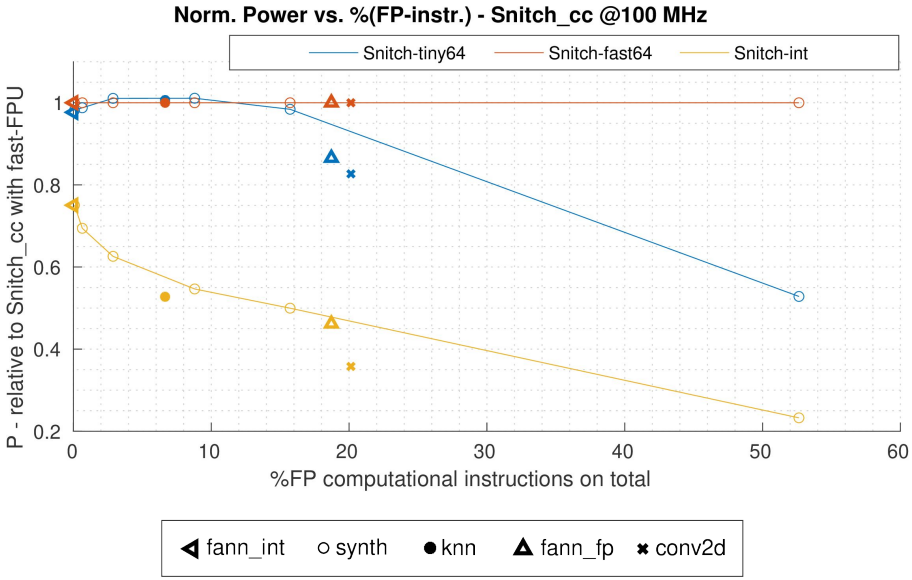
Performance

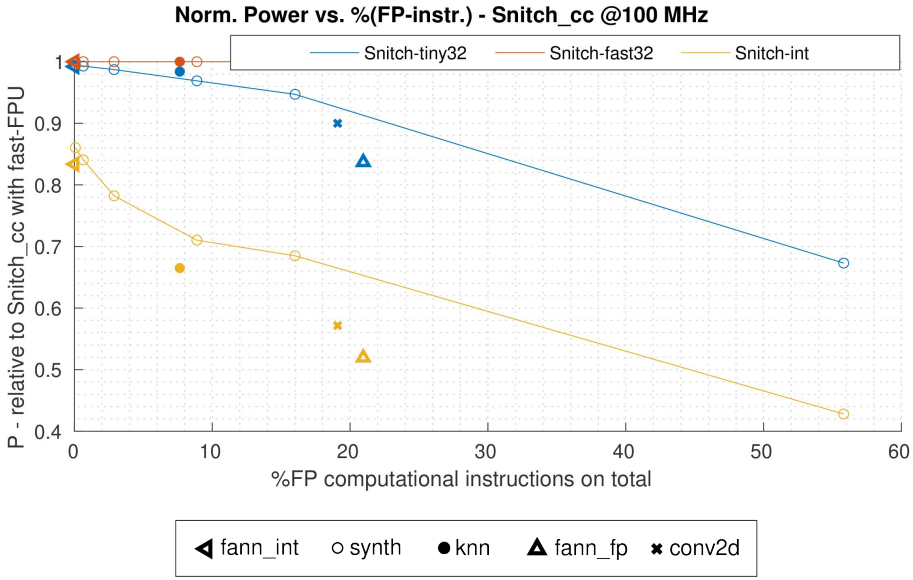




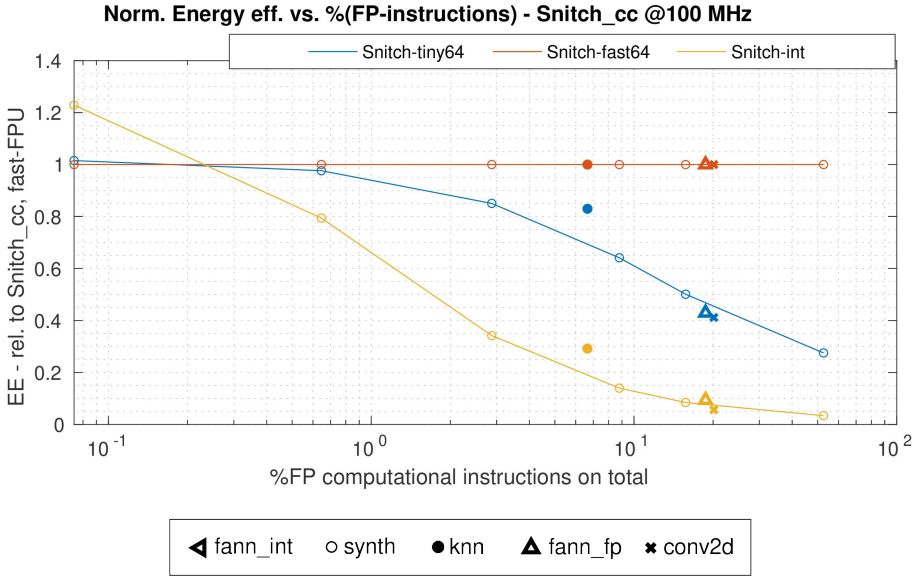
- $TP_x^{norm} = \frac{T_{fastFPU}}{T_x}$

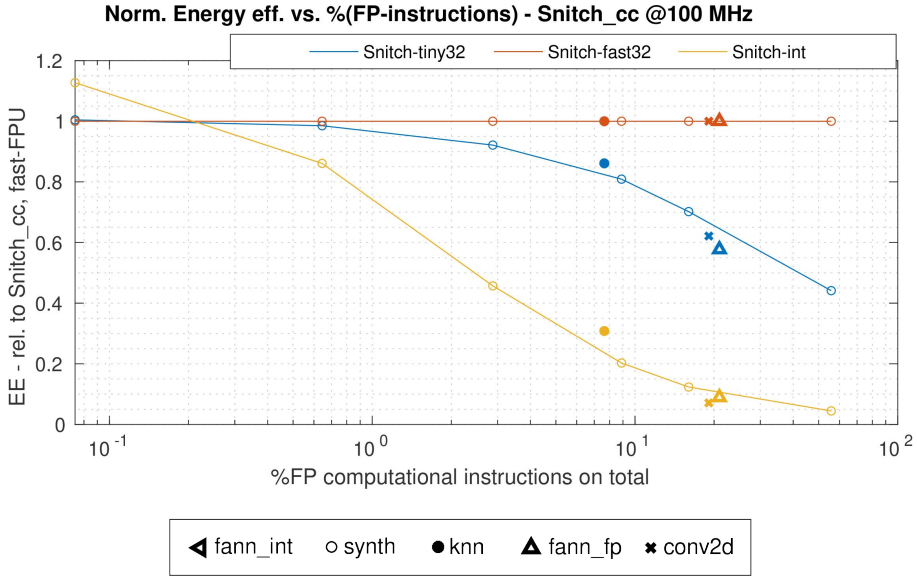
Power


Norm. Power vs. %(FP-instr.) - Snitch_cc @100 MHz
Snitch-tiny64
Snitch-fast64
Snitch-int
P - relative to Snitch_cc with fast-FPU
%FP computational instructions on total
fann_int
synth
knn
fann_fp
conv2d


Norm. Power vs. %(FP-instr.) - Snitch_cc @100 MHz
Snitch-tiny32
Snitch-fast32
Snitch-int
P - relative to Snitch_cc with fast-FPU
%FP computational instructions on total
fann_int
synth
knn
fann_fp
conv2d

Energy Efficiency





Benchmarks

TABLE II
RESULTS - DOUBLE-PRECISION BENCHMARKS (TCK = 100 MHz)

	Execution Time [cycles]			Power [µW]			Energy [nJ]		
Benchmark	**snitch-fast64**	**snitch-tiny64**	**snitch-int**	**snitch-fast64**	**snitch-tiny64**	**snitch-int**	**snitch-fast64**	**snitch-tiny64**	**snitch-int**
fann (21.0%)	43.74k (1.0)	117.67k (2.7)	1009.31k (23.0)	227.30 (1.0)	196.70 (0.9)	104.95 (0.5)	99.43 (1.0)	231.46 (2.3)	1059.27 (10.7)
conv2d (20.2%)	7.76k (1.0)	22.80k (2.9)	385.94k (49.7)	328.00 (1.0)	271.20 (0.8)	117.36 (0.4)	25.45 (1.0)	61.84 (2.4)	452.94 (17.8)
knn (6.6%)	11.07k (1.0)	13.26k (1.2)	71.91k (6.5)	210.90 (1.0)	212.30 (1.0)	111.25 (0.5)	23.35 (1.0)	28.14 (1.2)	80.00 (3.4)
fann fixed-point (0%)	63.96k (1.0)	63.96k (1.0)	69.00k (1.08)	146.40 (1.0)	143.00 (1.0)	109.87 (0.8)	93.64 (1.0)	91.47 (1.0)	75.81 (0.8)

TABLE III
RESULTS - SINGLE-PRECISION BENCHMARKS (TCK = 100 MHz)

	Execution Time [cycles]			Power [µW]			Energy [nJ]		
Benchmark	**snitch-fast32**	**snitch-tiny32**	**snitch-int**	**snitch-fast32**	**snitch-tiny32**	**snitch-int**	**snitch-fast32**	**snitch-tiny32**	**snitch-int**
fann (21.0%)	35.53k (1.0)	73.55k (2.1)	766.29k (21.6)	205.9 (1.0)	172.29 (0.8)	106.96 (0.5)	73.15 (1.0)	126.73 (1.7)	819.62 (11.2)
conv2d (19.1%)	7.79k (1.0)	13.93k (1.8)	208.54k (26.8)	222 (1.0)	199.79 (0.9)	126.88 (0.6)	17.29 (1.0)	27.82 (1.6)	264.60 (15.3)
knn (7.6%)	7.79k (1.0)	9.19k (1.2)	38.01k (4.9)	171.7 (1.0)	168.93 (1.0)	114.16 (0.7)	13.37 (1.0)	15.53 (1.2)	43.39 (3.3)
fann fixed-point (0%)	64.08k (1.0)	64.08k (1.0)	69.00k (1.1)	131.8 (1.0)	130.77 (1.0)	109.87 (0.8)	84.45 (1.0)	83.79 (1.0)	75.81 (0.9)

Results

Less than 5% FP instructions, Snich-tiny is:

- Up to 5x and 3x faster than Snitch-int (SW emulation)
- Only up to 25% and 15% of energy efficiency penalty (wrt Snitch-fast)
- Reduced code size (wrt to SW emul): 3.5kB and 2kB (for fadd e fmul), up to 16kB
- TinyFPU smaller than 53% and 37% wrt to FastFPU

Area constrained-systems:

- Snitch-tiny’s execution times up to 18.5x and 15.5x better than snitch-int (sw emul)

RISC-V Summit

- Tiny-FPU has been **accepted** to RISC-V Summit (online - Dec 8-10)
- **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

- Deadline presentation submission: November 13

FP library

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