ETH zürich

Report 17.09.20

17/09/2020

Matteo Perotti

Luca Bertaccini

Pasquale Davide Schiavone

Stefan Mach

Professor Luca Benini

Integrated Systems Laboratory

ETH Zürich

Tiny Floating-Point Unit

- **Microcode**
- **PAPER: FP comparison**

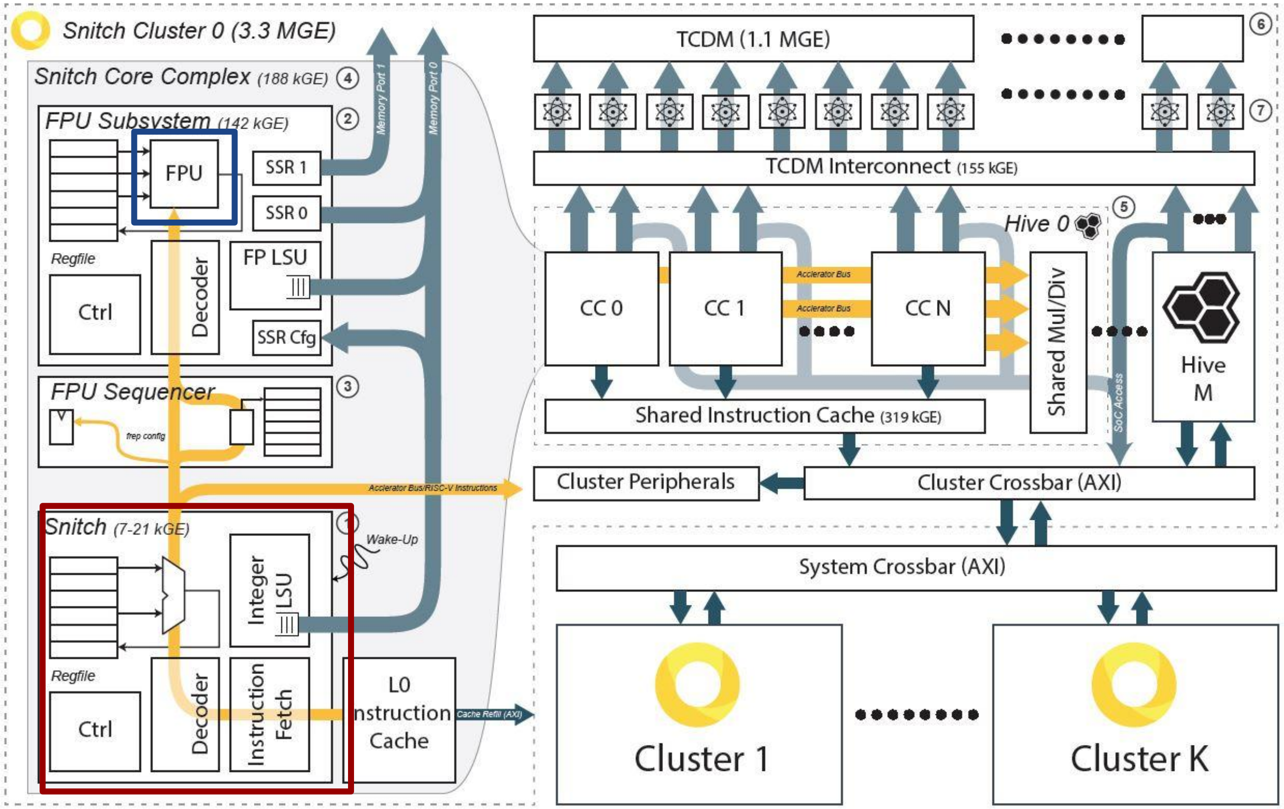
SNITCH

- **Snitch** RISC-V Processing Element with Streaming Registers (**RV32[I|E]MAFD**ZifenceiZicsrXssrXfrep)
  - Support for FP32 and FP64
  - **32-bit INT register file**
  - **64-bit FP register file**
- Snitch paper: https://arxiv.org/abs/2002.10143
- *“A general-purpose, **single-stage**, **single-issue core** tuned for utmost **energy efficiency**. Aiming to maximize the compute/control ratio (making the **FPU the dominant part of the design**) mitigating the effects of deep pipelines and dynamic scheduling”*

SNITCH

Tiny FPU

Snitch



Zaruba, F., Schuiki, F., Hoefler, T., & Benini, L. (2020). Snitch: A 10 kGE Pseudo Dual-Issue Processor for Area and Energy Efficient Execution of Floating-Point Intensive Workloads. arXiv preprint cs.AR/2002.10143.

Microcode

- Compile with **F/D extensions** enabled (e.g. fadd.s f1,f2,f3)
- Modify the decoder to forward the instructions to the Microcode module
- FSM generates the ALU operations needed to compute FP instructions on an INT datapath and forwards the inputs to the INT datapath
- The FSM has internal registers (“shadow register”) to store the operands and the partial results

ETH zürich

Microcode

Microcoded instructions:

- Comparison: FEQ.S, FLT.S, FLE.S
- FADD.S, FSUB.S
- FMUL.S

Microcode

Shadow registers:

- Three 32-bit registers: op_a/op_b/partial_result (rounding)
- FSM state
- Tag: 5 bits
- Flags to handle denormals: 3 bits
- 4 bits more for FADD.S

Microcode

	MICROCODE	SEGGER SW EMULATION (avg)	TINY FPU
FEQ_S	**2 cycles**	**10 cycles**	**2 cycles**
FLT_S	**5 cycles (max)**	**11 cycles**	**2 cycles**
FLE_S	**5 cycles (max)**	**10 cycles**	**2 cycles**
FMUL_S	**18 cycles**	**39.3 cycles**	**18 cycles**
FADD_S	**~30 cycles**	**49.5 cycles**	**10-13 cycles**
FSUB_S	**FADD_S latency + 1 cycle**	**62.2 cycles**	**10-13 cycles**

DATE21 Paper

Comparison:

1. **State-of-the-Art FPU**
2. **Tiny FPU**
3. **Optimized SW library**
4. **libgcc**

The microcode implementation is still ongoing, we will not include it in the paper

DATE21: Paper content

- Main focus of the paper: **Energy efficiency**
- Benchmark: 20x20 MATMUL + IDLE time
- I.e. we need to periodically compute a matmul. Changing the period will change the idle time

Next steps

- Complete the paper
- Complete the microcode FP32 implementations (FMADD.S)
- Find area cost of the microcode