HSLU Hochschule Luzern

Technik & Architektur
Master of Science in Engineering
Specialization: Computer Science

Master of Science in Engineering - Master Thesis

HW/SW-Codesign of a Tightly-Coupled Coprocessor for LTC

Design of a RISC-V coprocessor for the acceleration of LTC networks

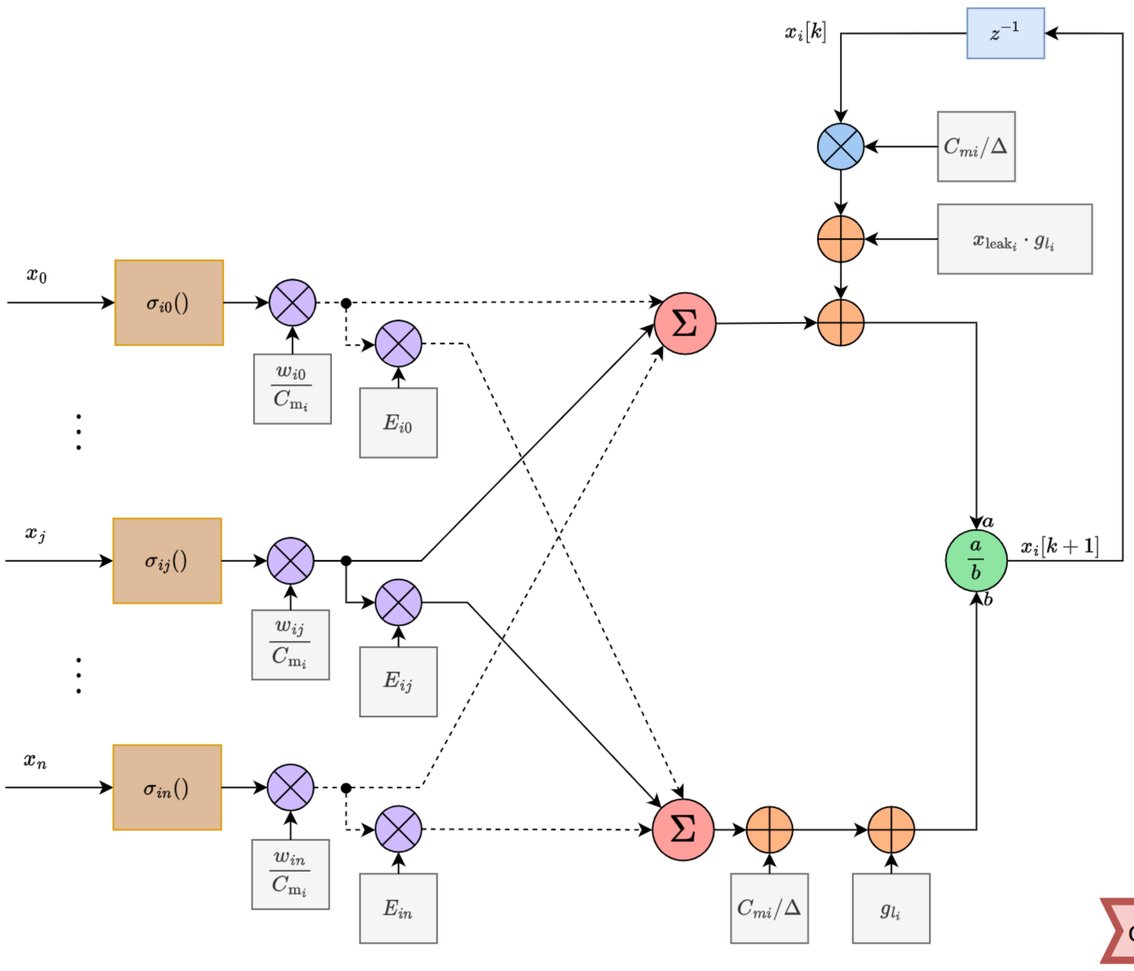

Figure 1: Signal flow graph of the LTC

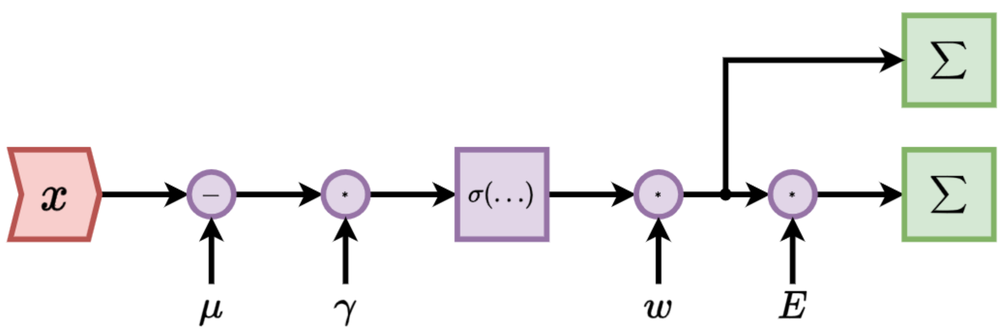

Figure 3: Datapath of the accelerated LTC part

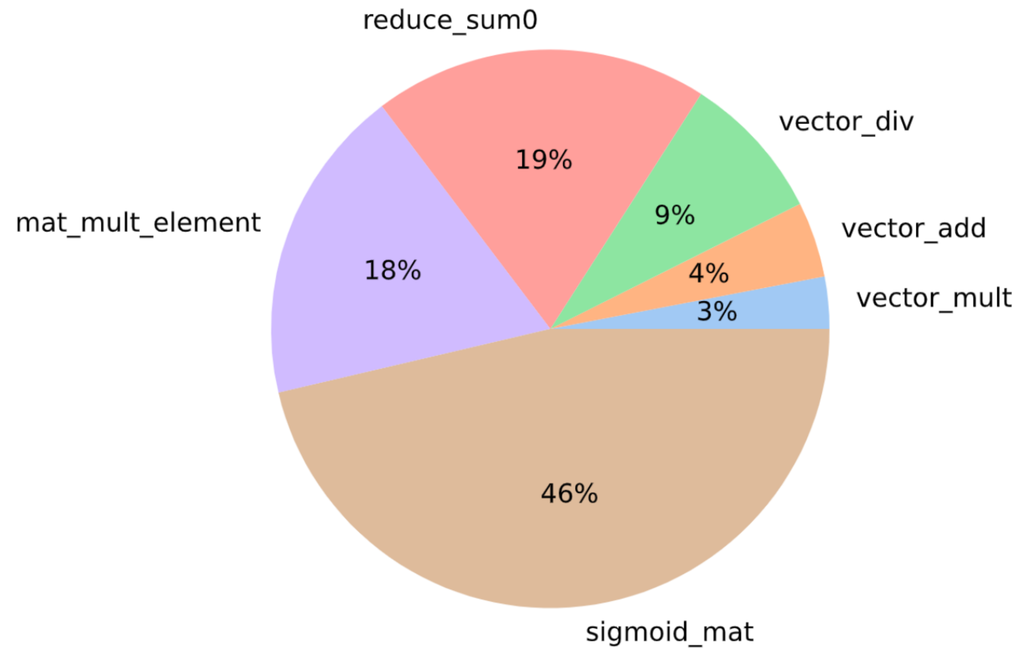

Figure 2: Profiling results of an LTC model in software

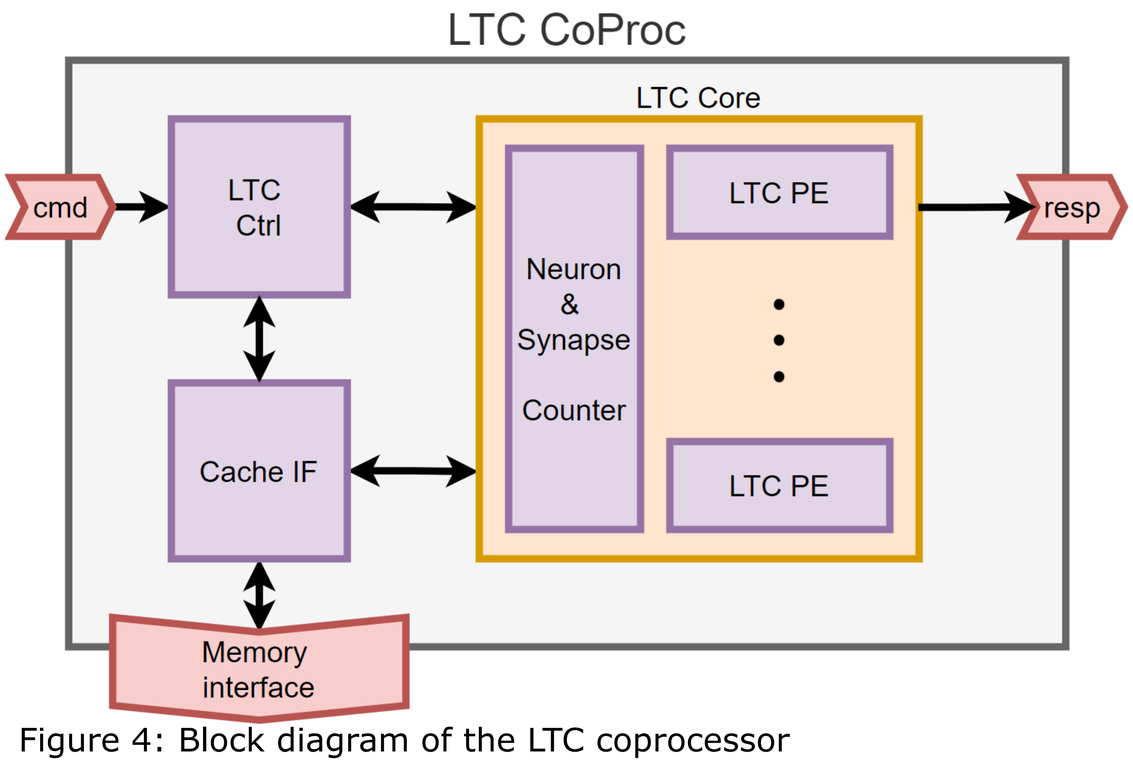

Figure 4: Block diagram of the LTC coprocessor

Project Task
Previous projects have shown that Liquid Time-Constant Networks (LTC) perform remarkably well in several machine learning tasks. Compared to other model types, they stand out by doing this with a very compact model architecture. However, estimations for low-power embedded systems showed, that the processing time still takes too long to fulfill certain real time requirements. Thus, the computations in software must be accelerated.
The main focus of this work is to design a tightly-coupled coprocessor to accelerate the inference of LTC. This includes verifying the functional correctness by simulation. Additionally to the design, an appropriate number format for the computation shall be selected and analyzed. Finally, the runtime performance shall be analyzed, and the improvement evaluated. The last step is to deploy the CPU and coprocessor to the Field Programmable Gate Array (FPGA) on the target platform. The coprocessor shall be designed as an extension of a RISC-V core within the Rocket-Chip generator.

Concept
A thorough analysis of the software implementation shows what part of the LTC contribute the most to the overall runtime. Hardware components for a coprocessor are designed to accelerate these computations. The coprocessor is highly configurable to suit different model sizes and a variable resources – throughput tradeoff.

Results
The profiling clearly showed that the operations on the left of the summation sign in Figure 1 are the major contributors for the computational effort. Figure 2 shows the percentages of the individual operations.
The datapath representing these operations is shown in Figure 3. This is implemented as a processing element (PE), which can be instantiated multiple times to enable parallel processing. The implementation is generic for all LTCs. The weights and architecture for a specific model can be loaded at runtime. The overall concept of the coprocessor is shown in the Figure 4.

The final evaluation shows that the accelerated operations can be computed up to 1000 times faster with the coprocessor, compared to a pure software implementation. For some models, the overall acceleration is still limited by parts implemented in software. Small adaptions to the coprocessor however could enable more acceleration. This would also bring a very significant overall speedup for all model architectures.

Student:
Andreas Rebsamen

Advisor:
Prof. Dr. Jürgen Wassner

Expert:
Dr. David Perels

Research / Development project:
ESA Study Object Tracking