REAL-TIME, VERSATILE, EVENT-DRIVEN, NEUROMORPHIC INFERENCE PROCESSOR (RAVEN)

Mr. Luke Vassallo, Prof. Nima Taherinejad
luke.vassallo@ziti.uni-heidelberg.de
ECLECTX Team, Institute for Computer Engineering, Heidelberg University

Motivation

- Temporal delays have recently emerged as a powerful computational primitive. When combined with sparse parameter sets, they enable extremely compact model architectures — unlocking new potential for ultra-efficient neural computation.
- From a computational standpoint, parameter sparsity and synaptic delays share a fundamental property: both rely on *indexed calculation*.
- RAVEN leverages a tiny RISC processor as a programmable state machine, achieving programmability, and precise, per-cycle control.
- RAVEN supports programmable neuron models, and arbitrary sparse connectivity allowing recurrence, convolutions, and delayed calculations.

Background

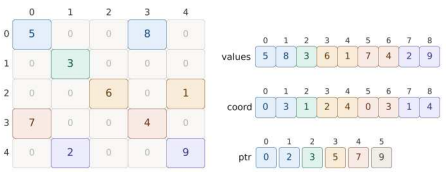

Figure 1: *Left:* Dense 5×5 matrix A — coloured cells are non-zeros (nnz = 9), grey cells are zeros (36 % fill). *Right:* CSR encoding of A using three arrays: `values[nnz]` (non-zero entries row by row), `col_indices[nnz]` (column of each entry), and `row_ptr[rows+1]` delimiting each row's slice. Colour coding matches the originating row in A.

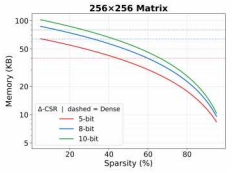

Figure 2: Absolute memory usage of Δ-CSR versus sparsity for 5-, 8-, and 10-bit quantisation (256-hidden-layer model).

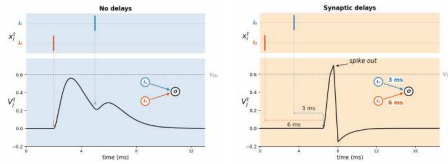

Figure 3: Two pre-synaptic neurons I1 and I2 project to output neuron O with an identical 3 ms firing. *Left:* Without synaptic delays no spike is produced. *Right:* Delays compensate for the firing offset, causing coincident arrival at the post-synaptic neuron and subsequent firing.

Delays as a computational primitive. Delays offer a dedicated parameter set for pattern recognition across the temporal dimension. Introducing delays shifts part of the representational burden from *weight magnitude* to *timing*, enabling the network to exploit temporal structure in the input stream.

Sparse weight and delay parameter sets achieve classification accuracy levels that would otherwise require significantly larger dense models. This synergy between sparsity and delay is central to RAVEN's design: both features map naturally onto the same indexed-access hardware, imposing no additional control overhead.

Organization and Architecture

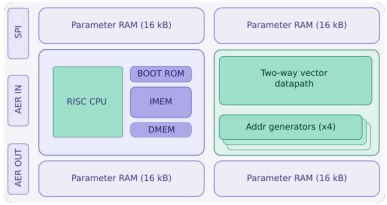

Figure 4: RAVEN block diagram - An SPI interface is used for configuring the device, and the Address Event Representation interface is used to transfer spiking data into and out of the chip. The PSM performs intricate control, while the computation fabric performs the SNN calculations using the parameter memories for both *model storage* and *model implementation*.

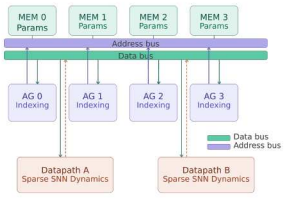

Figure 5: Organisation of the computation fabric. Address generators (one per memory bank) compute parameter memory addresses using a combination of parameters and coordinates. The datapath consists of a two-way vector unit that implements the full dynamics.

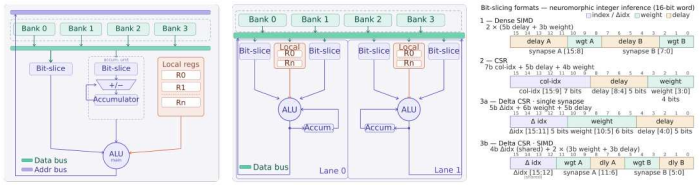

Figure 6: (a) Address generator block diagram — the destination memory address can be computed as a function of matrix metadata (index), model parameter (delay) and several local iterators, supporting structural sparsity and temporal delays simultaneously. (b) Organization of the two-way vector 16-bit integer datapath — Each lane interfaces with the parameter memory pools and operates on data indexed by the address generators. (c) Illustrative example of how bit-slicing hardware facilitates tight packing of model parameters into a 16-bit word.

The discrete-time LIF neuron with arbitrary connectivity and delays (axonal, synaptic, or dendritic) evolves as follows:

$$V_j^{t+1} = \alpha V_j^t + \underbrace{\sum_i W_{ji}\, x_i^{t-D_{ji}}}_{\text{feedforward}} + \underbrace{\sum_k W_{jk}\, z_k^{t-D_{jk}}}_{\text{recurrent}} - z_j^t V_r$$

$$z_j^t = \begin{cases} 1 & \text{if } V_j^t > \vartheta \\ 0 & \text{otherwise} \end{cases}$$

where V_j^t is the membrane potential of neuron j, α the leak factor, x_i^t the input spike train, z_j^t the output spike train, W_{ji} feedforward weights, W_{jk} recurrent weights, D_{ji}, D_{jk} the corresponding synaptic delays, V_r the reset voltage, and ϑ the firing threshold.

Design, Validation, Tapeout

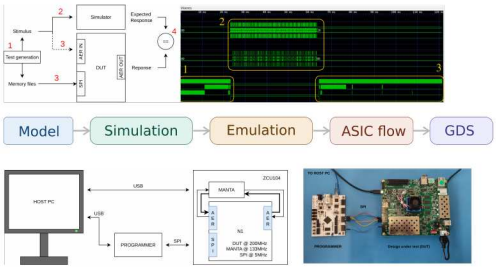

Figure 7: Design validation at various steps of the flow, namely, RTL simulation, FPGA emulation, and ASIC testing.

- A Python simulator models all aspects of RAVEN, and was used to generate representative test cases for all core features supported by the hardware.
- Prior to tapeout RAVEN was validated end-to-end in RTL functional simulation, FPGA emulation, and post place-and-route back-annotated simulation at all relevant corners.
- Bare dies were wire-bonded in-house to an FPGA-Mezzanine-Card and validated in our lab at all operational voltage-frequency points.

Results

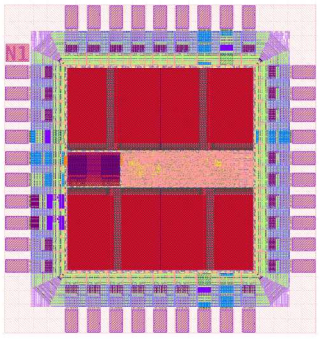

Figure 8: RAVEN's layout - Four 16kB single port parameter memories enclose the programmable state machine with 1kB of dual port memory, and processing elements.

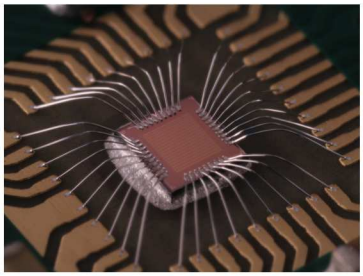

Figure 9: Photo of the bare die glued and wire-bonded to a PCB.

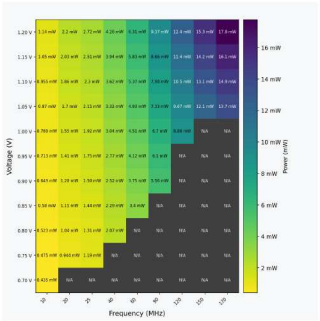

Figure 10: Voltage-frequency sweep

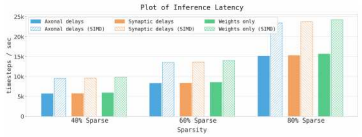

Figure 11: Inference latency for a 140 x 128 x 20 model

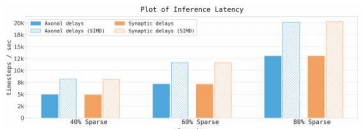

Figure 12: Inference latency for a 140 x 128 x 128 x 20 model

Conclusions and Future Outlook

- Preliminary power and latency data shows that RAVEN can perform real-time operation in the $500\mu W$ regime while achieving SoTA task accuracy.
- Optimizing for the highest accuracy at the lowest power requires leveraging the versatility of SNNs, as well as factoring in microarchitecture constraints during model training.
- RAVEN is being integrated into wearable healthcare devices for real-time classification of EEG, ECG and time-series data.

Figure 13: Modular wearable biosignal measurement platform

Acknowledgements

We would like to thank Ms. Didem Duman, Mr. Sören Jost, and Mr. Chibueze Elesulam Peter for contributing to the development of the test infrastructure; The ASIC Team at Heidelberg University, namely Mr. Markus Dorn for assistance with tools, technologies, and design submission, and Mr. Ralf Achenbach for attaching and wire bonding the dice. We also extend our gratitude to Prof. Dr. Johannes Schemmel, Dr. Andreas Grübl, Yannik Stradmann, Joscha Ilmberger, and the Electronic Visions Group for their insightful discussions.