FPGA Based Emulation Environment for Neuromorphic Architectures

Spencer Valancius[*], Edward Richter[*], Ruben Purdy[*], Kris Rockowitz[*], Michael Inouye[*], Joshua Mack[*], Nirmal Kumbhare[*], Kaitlin Fair[†], John Mixter[‡], Ali Akoglu[*]

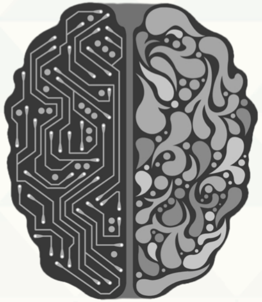
[*]Department of Electrical and Computer Engineering, University of Arizona
{svalancius12, edwardrichter, rubenpurdy, rockowitzks, mikesinouye, jmack2545, nirmalk, akoglu}@arizona.edu

†Air Force Research Labs
kaitlin.fair@us.af.mil

‡: Raytheon Missile Systems
John_E_Mixter@raytheon.com

COLLEGE OF ENGINEERING
Electrical & Computer Engineering

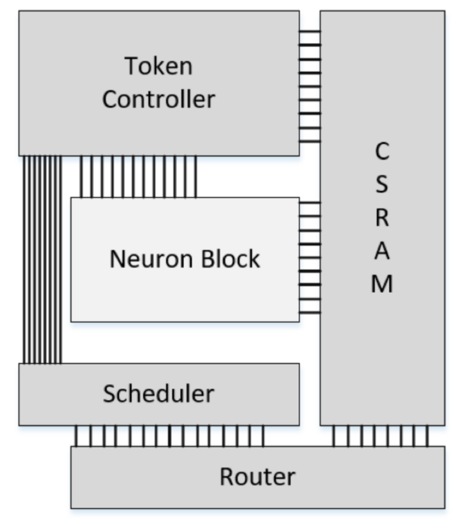
Motivation



- There is a need for an open source and easy to use ecosystem in neuromorphic hardware research
- In this work, we present such an environment
  - A highly flexible environment that enables rapid experimentation with neuromorphic architectures in both software via C++ simulation and hardware via FPGA emulation
  - This enables hardware architects and application engineers to investigate and tune parameters of their neuromorphic architecture that would otherwise be unavailable on a purely prefabricated ASIC.

Reference Architecture

- MxN grid of cores with each core consisting of
  - Neuron block: implements fundamental leaky-integrate-and-fire (LIF) neuron model
  - CSRAM: stores configuration parameters for the neuron block and router
  - Token controller: Coordinates data movement and computation between the CSRAM and neuron block. Additionally, receives incoming spikes from the scheduler
  - Router: routes outgoing spikes to their destination cores
  - Scheduler: schedules received spikes for processing by this core's neuron block



Reference Architecture

- Able to recreate a majority of the features of IBM's TrueNorth platform through parameter choices
- Verified functional equivalence through a number of case studies using MNIST and Vector Matrix Multiplication experiments

	TrueNorth	This Work
Global Tick Rate	1 kHz	1 kHz
Axon-Dendrite Crossbar	256 x 256	Parameterized
Weights per Neuron	4	Parameterized
Neuron Potential	9 bit signed	Parameterized
Reset Potential	9 bit signed	Parameterized
Weight Value	9 bit signed	Parameterized
Leak Value	9 bit signed	Parameterized
Pos./Neg. Threshold	18/18 bit signed	Parameterized
LIF Neuron Model	YES	YES
Linear Reset	YES	YES
Stochastic Behaviors	YES	NO

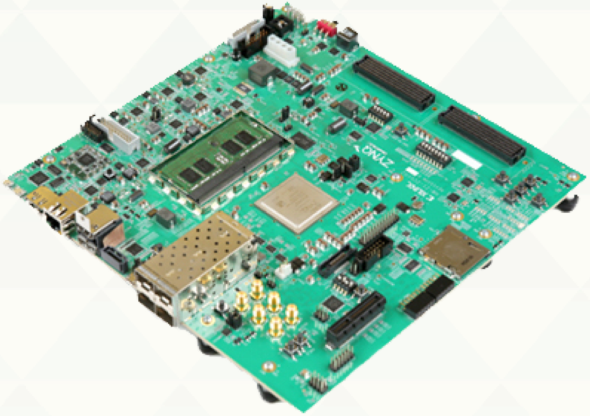
Hardware Implementation Results

- Network Size: number of neuromorphic cores present in the design
- Resource utilization results collected from Xilinx Zynq Ultrascale+ MPSoC ZCU102 development board
- Limited by LUT exhaustion

Network Size	LUT (%)	LUT-RAM (%)	FF (%)	BRAM (%)	Delay (ns)
1x1	8.40	0.31	3.15	0.60	9.143
2x2	9.99	0.73	3.68	1.81	7.975
3x3	14.03	1.79	5.05	4.82	9.727
4x4	19.75	3.27	7.01	9.05	8.480
5x5	27.15	5.17	9.55	14.47	9.360
6x6	36.24	7.49	12.68	21.11	9.397
7x7	47.01	10.23	16.40	28.95	10.907
8x8	59.47	13.40	20.70	37.99	9.015
9x9	73.62	16.99	25.59	48.25	9.498
10x10	89.45	21.00	31.07	59.70	8.305
10x11	97.78	23.11	33.95	65.73	9.926

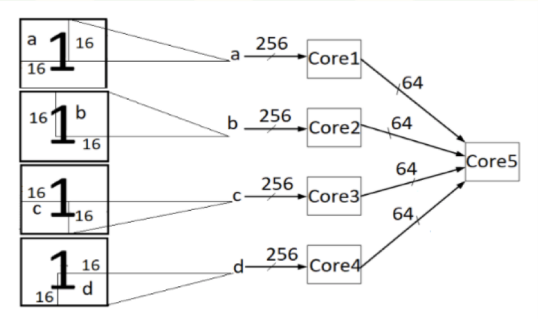
Streaming Framework

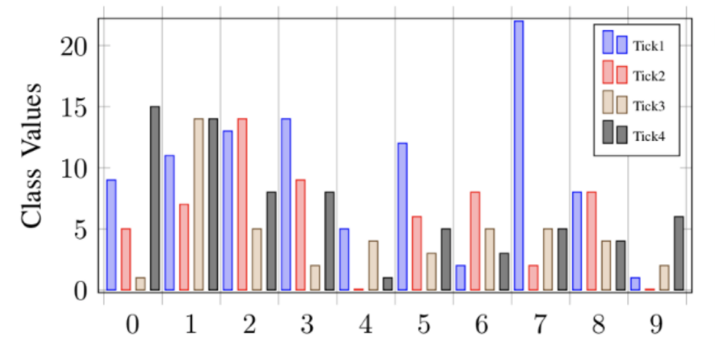
- “Bare metal” C application utilizing two threads
  - “TX” thread that reads input from SD card and sends data into the network via DMA transfer
  - “RX” thread that receives and writes output to the SD card
- Supports deploying neural networks that have been trained with the “constrain-then-train” methodology from Esser et al. [1].



Verification Approaches

- Using the streaming framework, verified with a mixture of traditional (MNIST) and non-traditional (VMM) neural network applications
- MNIST: Validated against 5 core network presented by Yepes et al. [2]
- VMM: Validated 100 random VMM executions from 2x3 to 8x8 against IBM's Compass [3] environment
- Both matched with TrueNorth across both experiments, giving evidence towards functional equivalence





Neuron Modifications for Efficient VMM Mapping

- Traditional signed (pos./neg.) VMM on TrueNorth requires a large amount of resource duplication due to architectural limitations [4]
    - Negative threshold uses a "<" comparison while the positive uses "≥"
- Architectural reconfigurability allows for correcting this asymmetry
    - Yields a massive savings in neurons required and consequently FPGA resources while maintaining VMM functionality

$$y = x^T M$$

Design	**LUT**	**LUT-RAM**	**FF**	**BRAM**	**Delay (ns)**
Reference	1700	192	1210	4	10.266
Proposed	1165	48	1056	2	8.026
Reduction (%)	**31.5**	**75.0**	**12.7**	**50.0**	**21.8**

Conclusion

- We are looking forward to exploring opportunities available in this environment for architectural optimizations
- Future work:
  - CNN execution and optimizations
  - On-chip learning
  - Multicast routing
- Website + GitHub + Contact information: https://ua-rcl.github.io/RANC

References

[1] S. K. Esser, P. A. Merolla, J. V. Arthur, A. S. Cassidy, R. Appuswamy, A. Andreopoulos, D. J. Berg, J. L. McKinstry, T. Melano, D. R. Barch, C. di Nolfo, P. Datta, A. Amir, B. Taba, M. D. Flickner, and D. S. Modha. Convolutional networks for fast, energy-efficient neuromorphic computing. *Proceedings of the National Academy of Sciences*, 113(41):11441–11446, 2016.

[2] A. Jimeno-Yepes, J. Tang, and B. S. Mashford. Improving classification accuracy of feedforward neural networks for spiking neuromorphic chips. In *2017 International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[3] R. Preissl, T. M. Wong, P. Datta, M. Flickner, R. Singh, S. K. Esser, W. P. Risk, H. D. Simon, and D. S. Modha. Compass: A scalable simulator for an architecture for cognitive computing. In *2012 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 1–11, 2012.

[4] K. L. Fair, D. R. Mendat, A. G. Andreou, C. J. Rozell, J. Romberg, and D. V. Anderson. Sparse coding using the locally competitive algorithm on the TrueNorth neurosynaptic system. *Frontiers in Neuroscience*, 13:754, 2019.