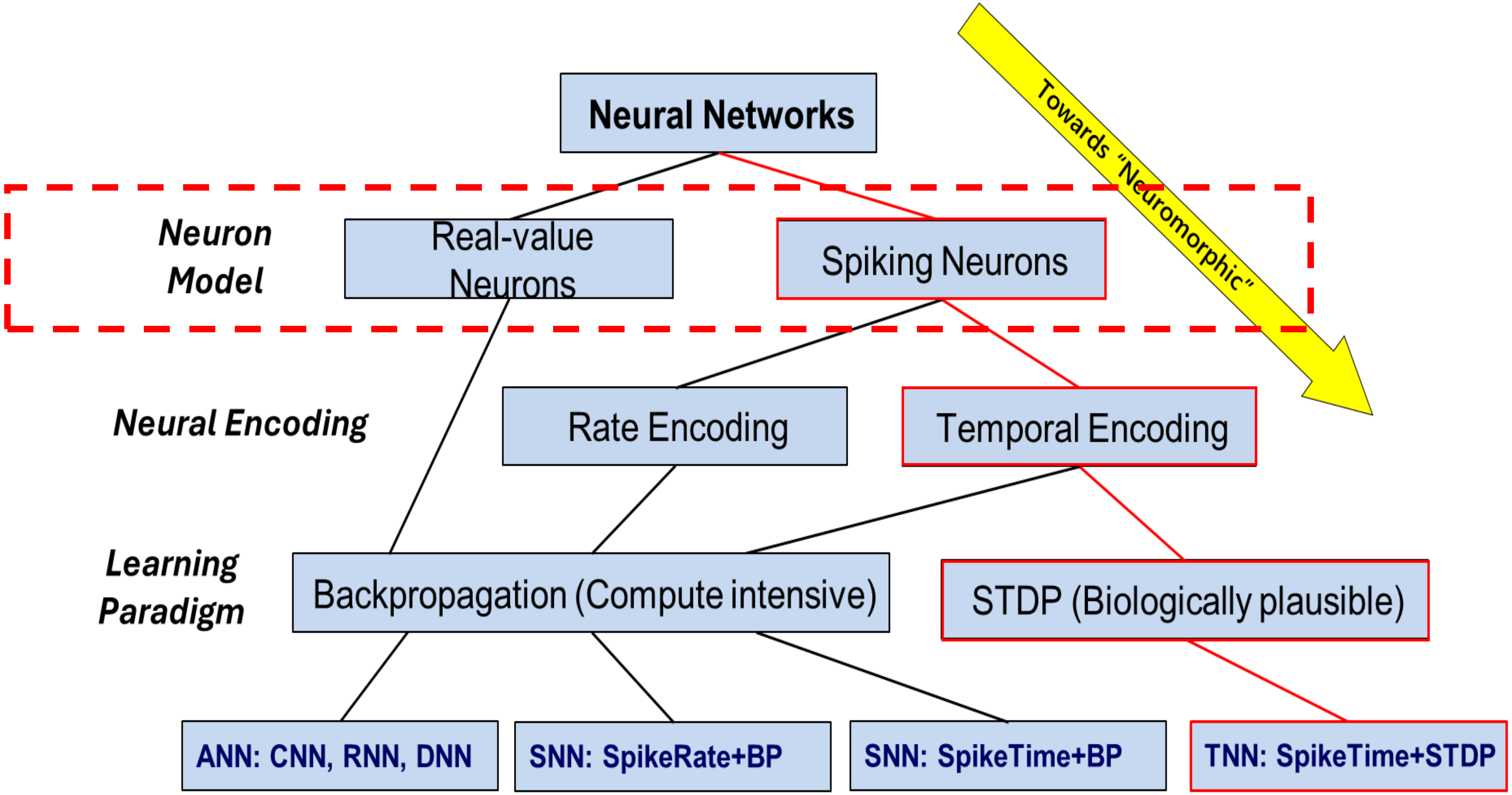
Catwalk: Unary Top-K for Efficient Ramp-No-Leak Neuron Design for Temporal Neural Networks

Devon Lister* devon.lister@ucf.edu
Prabhu Vellaisamy◇ pvellais@andrew.cmu.edu
John Paul Shen◇ jpshen@andrew.cmu.edu
Di Wu* di.wu@ucf.edu

*University of Central Florida
◇Carnegie Mellon University

ISVLSI 2025
Kalamata, Greece
20 May 2025

Temporal Neural Networks (TNNs) [1,2]

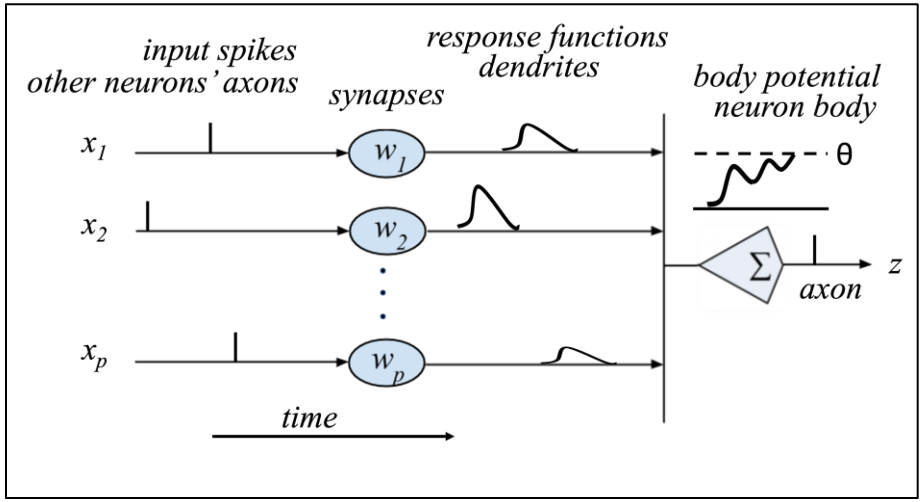


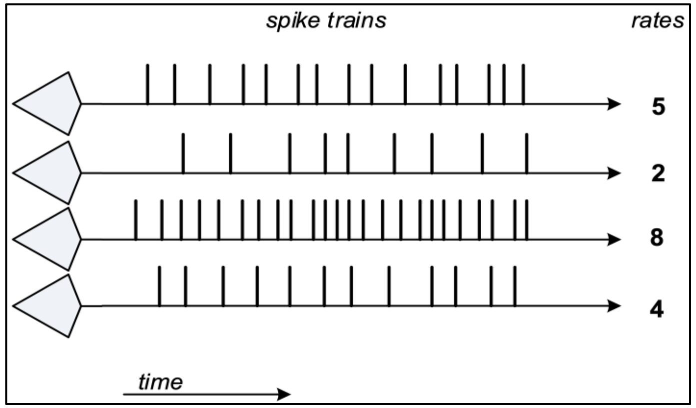
- TNNs are capable of continuous online learning and unsupervised clustering.
- This work focuses addresses the inefficiency of current spiking neuron implementations.

Why TNNs? – Neuromorphic Traits

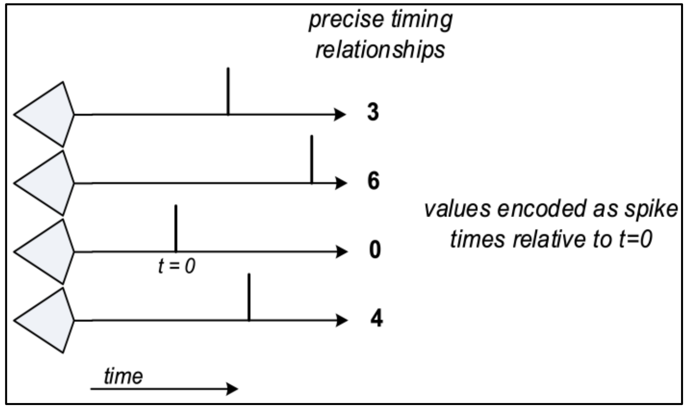
- **Spiking Neuron** - **Spike Response Model**
- **Temporal Encoding**[3] - **One spike per neuron**
- **STDP** - **Form of Hebbian Learning**

Condition	Action
o/p spike occurs after i/p spike arrives	Increase synaptic weight
o/p spike occurs before i/p spike arrives	Decrease synaptic weight





Rate Encoding

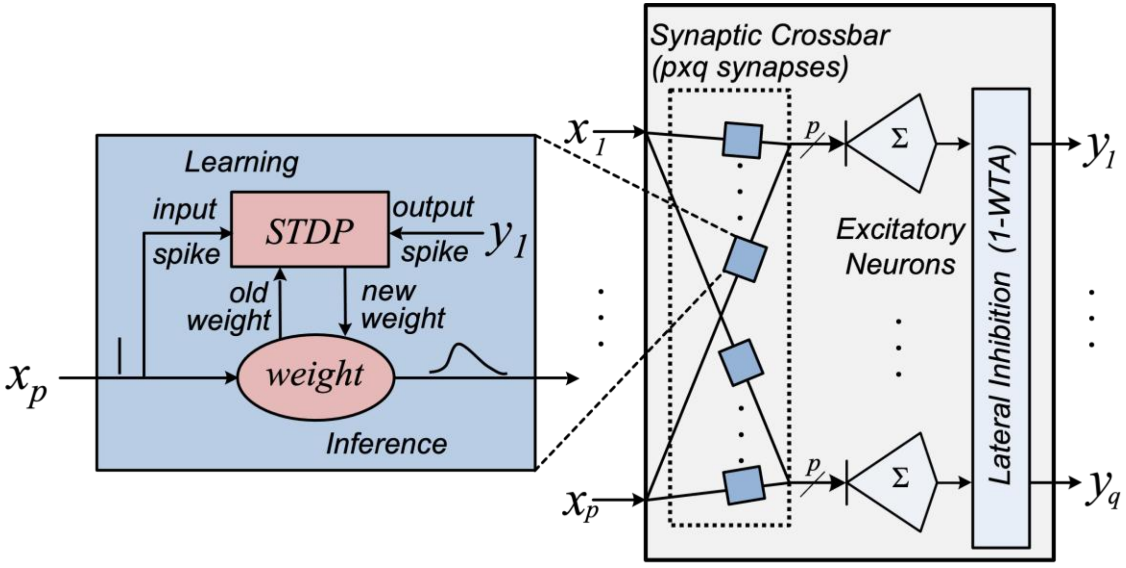


Temporal Encoding

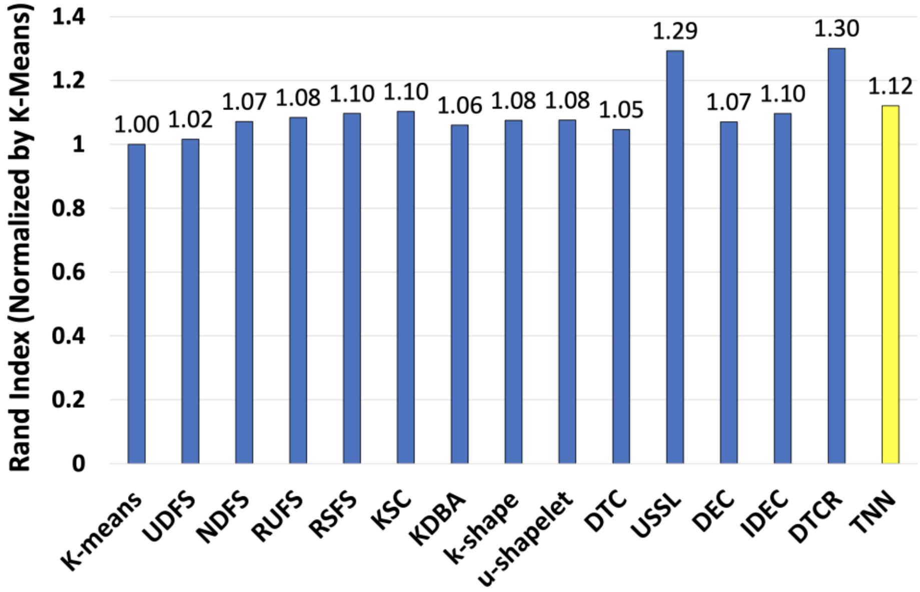
TNNs – Online Learning and Clustering

- **Excitatory neurons + Winner-Take-All inhibition**

- **A fully operational TNN building block!**
  - **Online Learning of MNIST digits[4]**
  - **Unsupervised time-series clustering[5]**



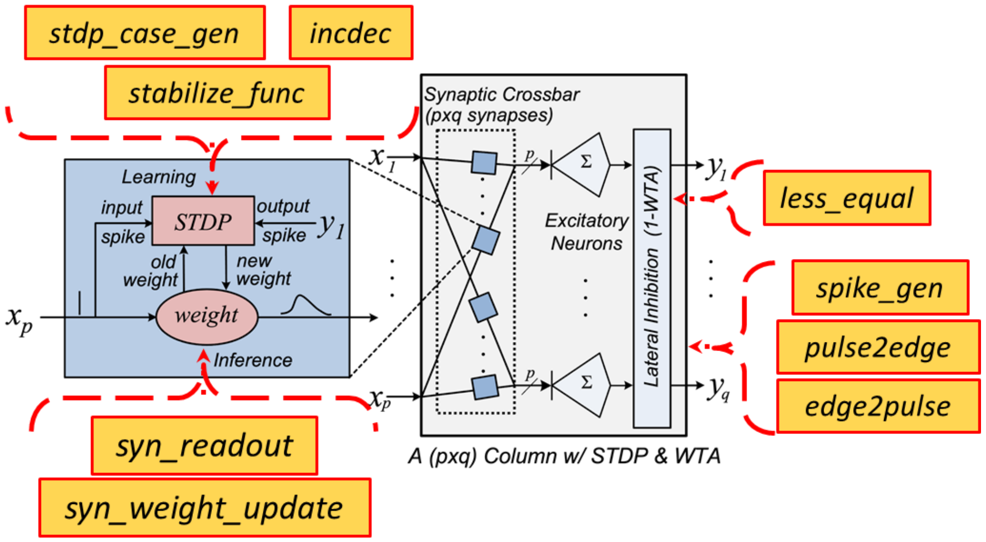
A (pxq) Column w/ STDP & WTA

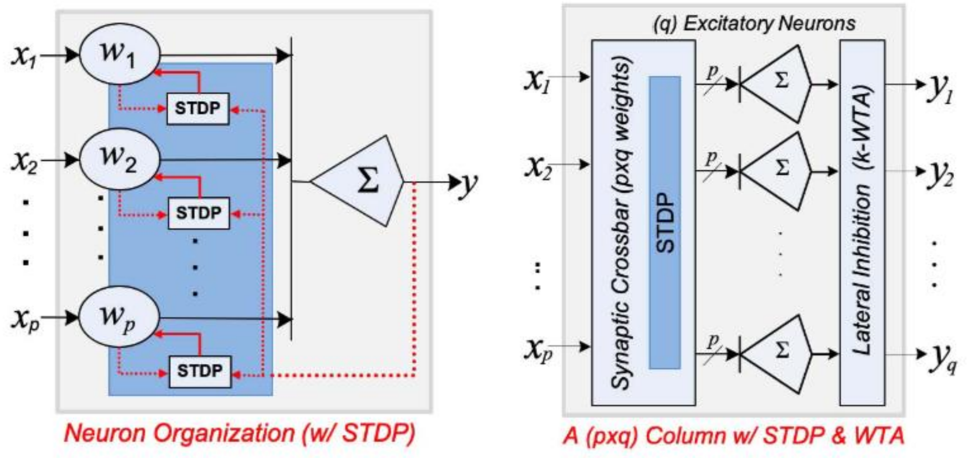


Previous TNN Implementations

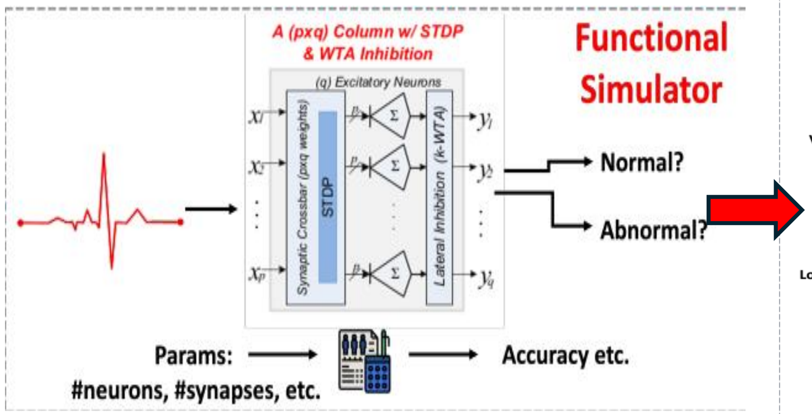
❑ TNN hardware developments

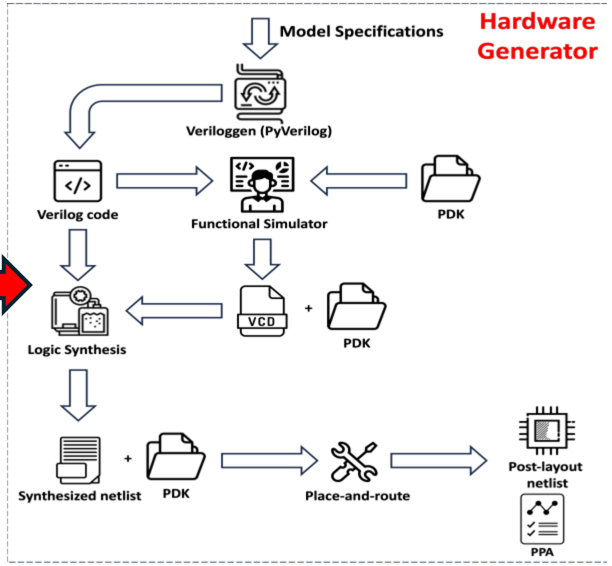
- Microarch implementation of TNNs in 45nm CMOS[4]
- TNN7: custom cell library for TNNs in 7nm[6]
- TNNGen: Automated SW-HW design flow[7]



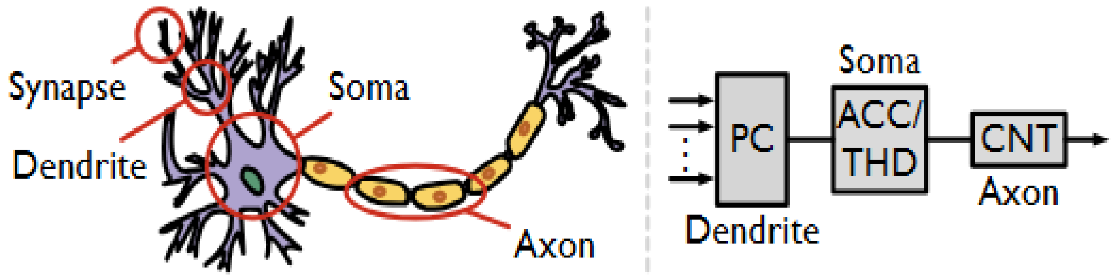


(a) Neuron: p Synapses, STDP (b) Column: q Neurons & WTA

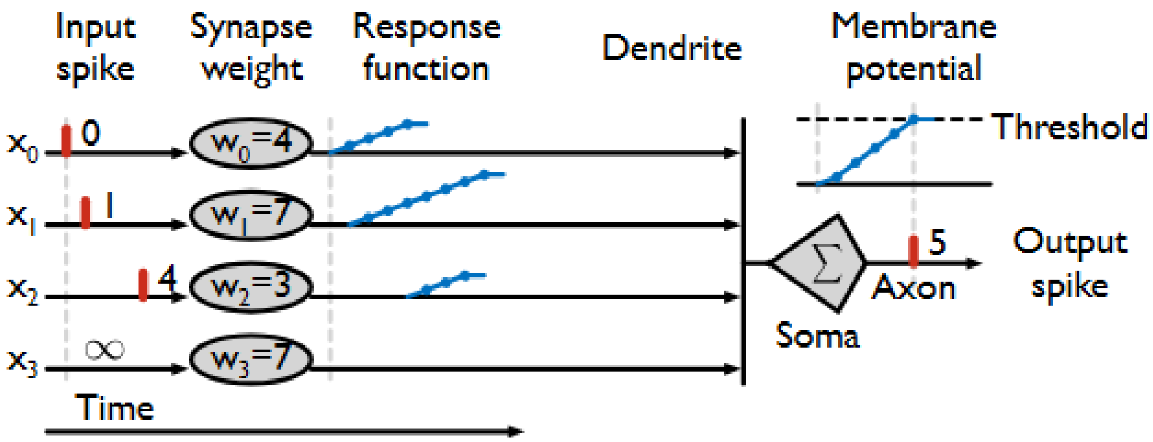




SRM0-RNL Neuron Model



Biological neuron and its RNL circuit representation for RNL response function.



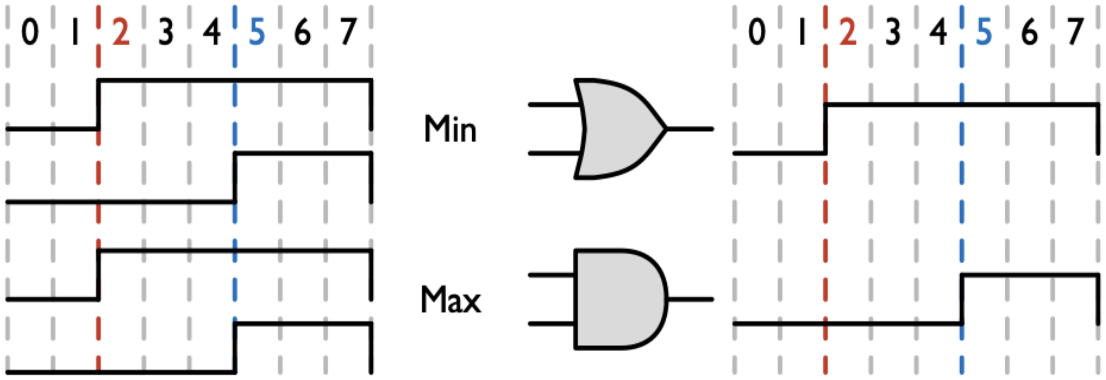
Existing SRM0-RNL neuron model with input spikes temporally-coded (red pulses).

$$\rho(w,t) = \begin{cases} 0, & t < 0 \\ t+1, & 0 \le t < w \\ w, & t \ge w \end{cases}$$

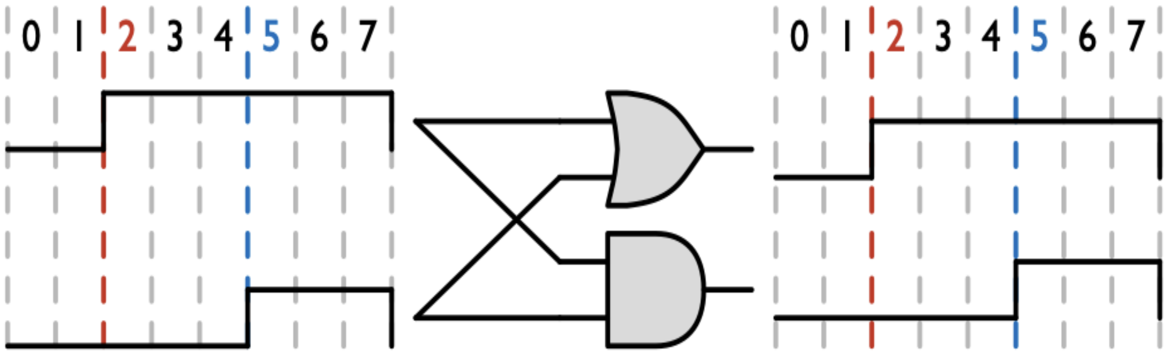
RNL response function equation.

- **Existing SRM0-RNL neuron design assumes worst-case scenarios and are suboptimal.**
  - **For *n*-input neuron, PC must accumulate *n* inputs even when absence of temporal spikes.**
  - **However, neuron spikes are sparse (only 0.1% - 10% of total neurons are spiking actively).**

Unary Sorting



Min and max operations using temporal coding.

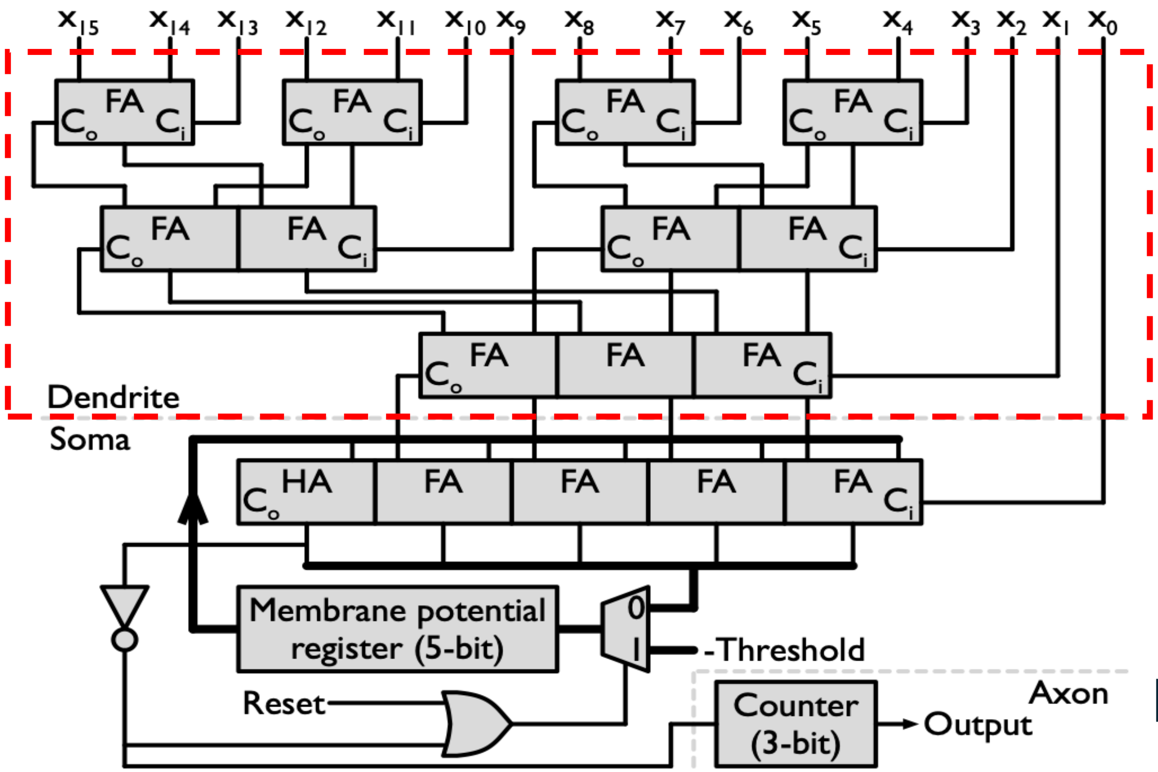


Compare-and-swap unit using min and max implementing a 2-input bitonic sorter.

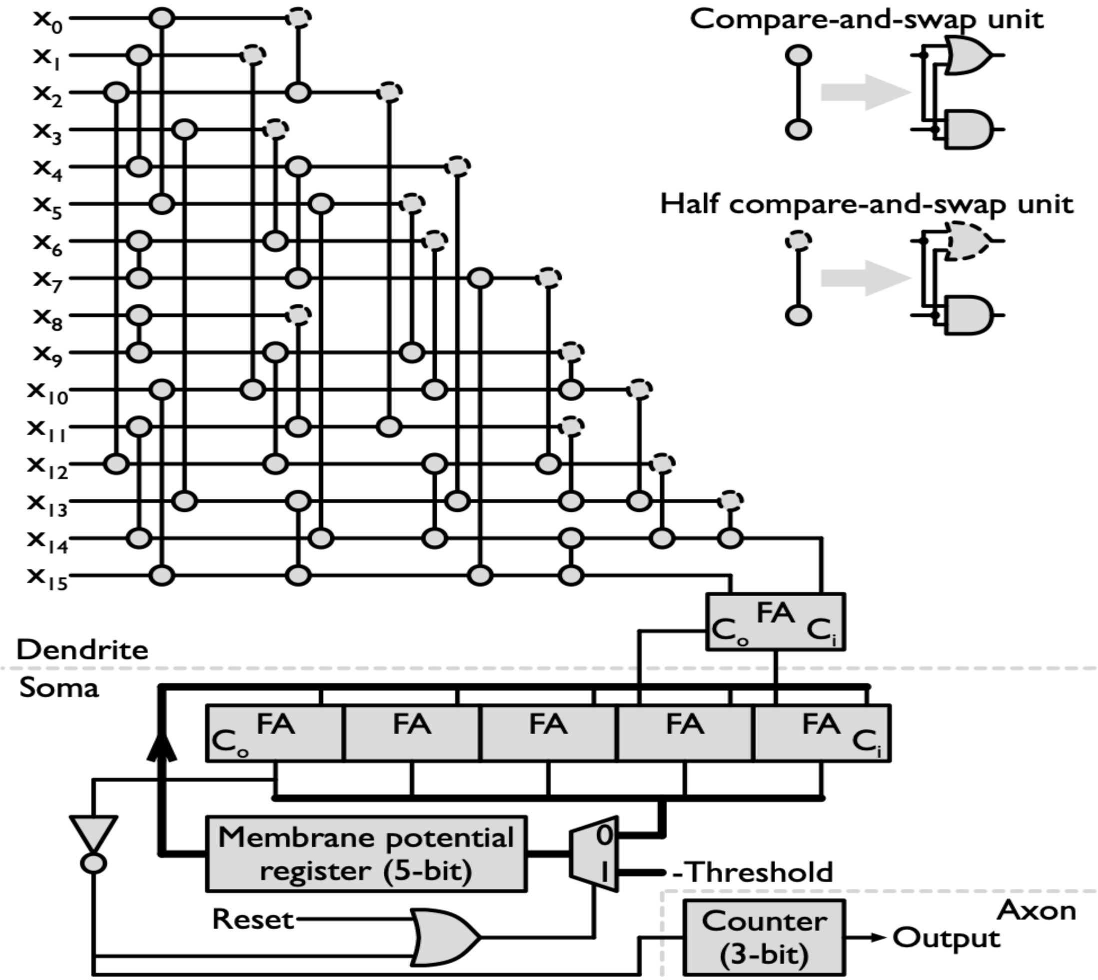
- **Bitonic sorting can be implemented via simple AND (min) and OR (max) gates.**
  - **Temporal spikes to SRM0-RNL neuron can be ranked, with larger values clustered at bottom.**
  - **Finding inputs with effective spikes allows implementation of more lightweight parallel counter (PC).**

Propose Catwalk neuron model leveraging optimized spike aggregation with lightweight PC design!!

Catwalk Neuron: Microarchitecture



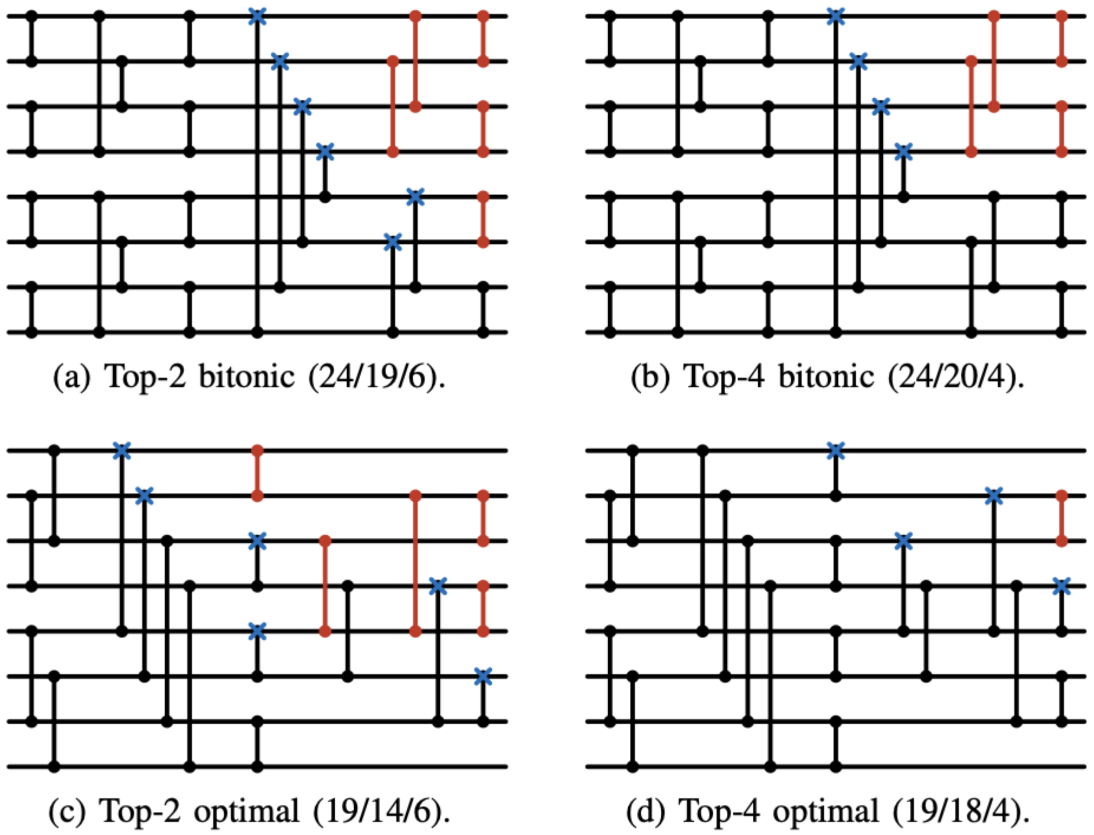
Previous SRM0-RNL neuron body microarchitecture (PC Compact[4]), utilizing a 16-input PC.



Catwalk SRM0-RNL neuron body microarchitecture, taking in 16-inputs and selecting top-2 outputs.

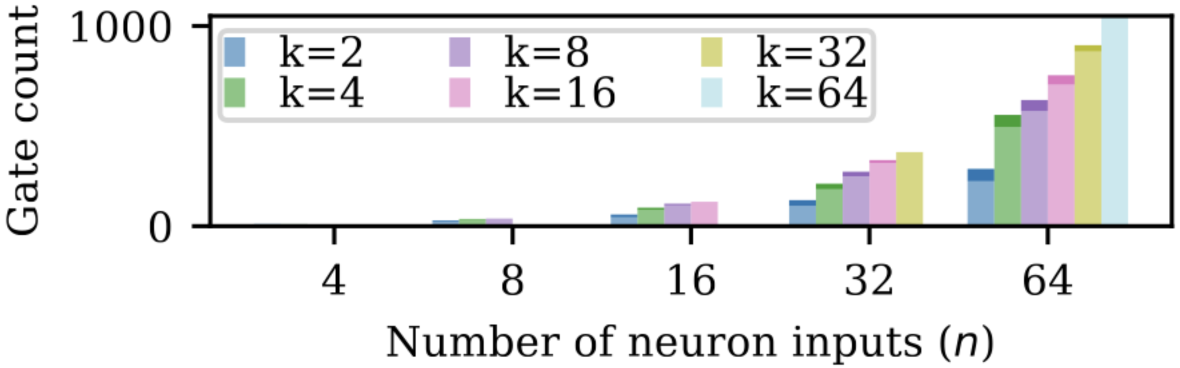
Catwalk Neuron: Top-*k* Selection Designs

- Two types of unary sorters - (i) bitonic and (ii) optimal[7].
- Different unary sorters produce identical results with different cost reduction.
  - For top-2, bitonic and optimal sorters prunes identical compare-and-swap units.
  - For top-4, bitonic sorters prunes more.
- Final cost of unary top-k is independent of the cost reduction in compare-and-swap unit.
- Higher the *k*, the higher the hardware cost.
- Catwalk neuron model incorporates the optimal sorters.

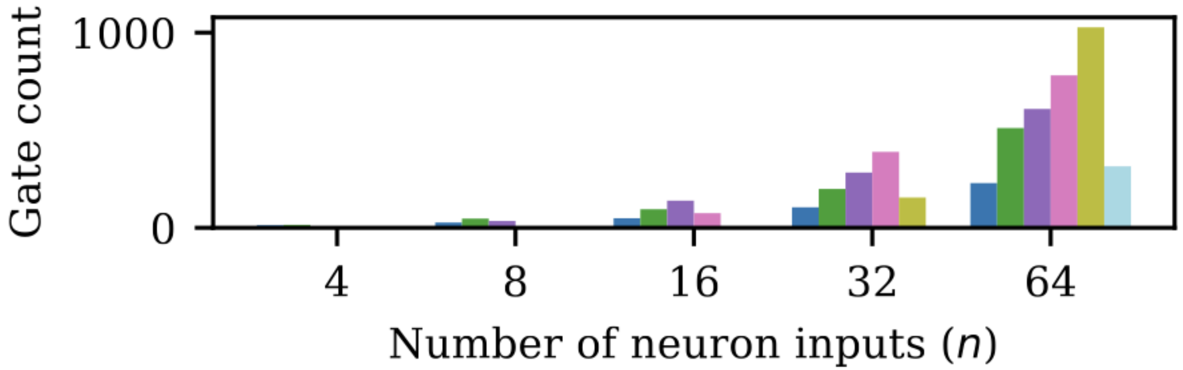


Comparison of unary top-k selector derived from different unary sorters with 8 inputs. (a) and (b) are pruning bitonic sorters, while (c) and (d) are pruning optimal,

Gate Count Analysis



Gate count of unary top-k. Solid color at top is for the removed gates in half compare-and-swap units.



Gate count of dendrite adopting unary top-k and compact PC.

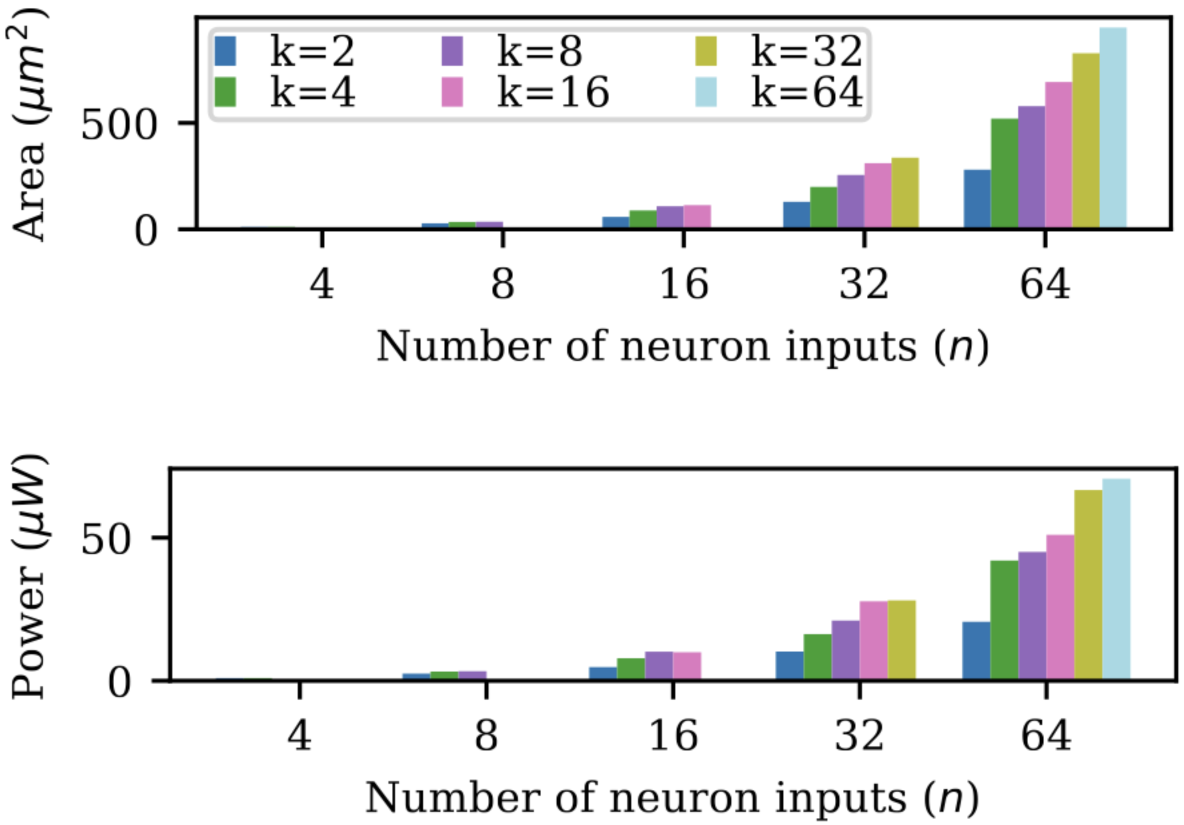
- **Significant hardware savings from:**
  - **Pruned swap-and-compare units.**
  - **Removed gates from half compare-and-swap units**.
- **Increased cost savings with scaled inputs – demonstrating potential of unary top-k.**
- **For dendrite designs, unary top-2 provides gains in gate count. Larger k values do not.**

Evaluation Setup

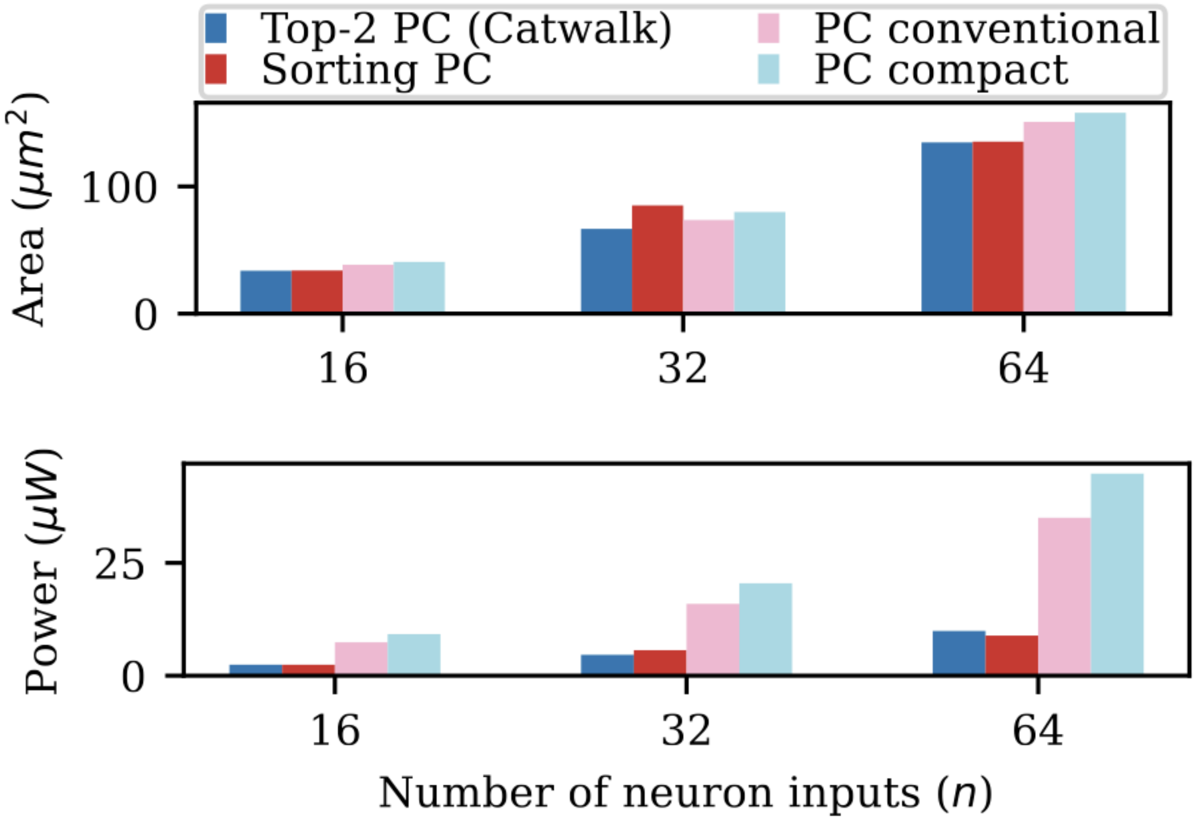
- Hardware evaluations performed using NanGate45 standard cell library for 45nm CMOS results.
- Design configurations:
  - A stand-alone sorting/top-k stage, including unary bitonic sorters and optimal unary top-k.
  - A sorting/top-k stage interfaced with a PC (a conventional design and a compact design).
  - Full SRM0-RNL neuron (bitonic sorting/optimal top-2 stage interfaced with a PC and augmented with a thresholding and firing unit).
- Synthesis and place-and-route performed using Synopsys Design Compiler and Cadence Innovus at 400MHz clock frequency.
  - Square floorplan with 70% utilization for each neuron input size (16, 32, and 64).
- Only n={4, 8, 16, 32, 64} are publicly available. Exploration of larger n is for future work.

Post-Synthesis Results: Top-K and Dendrite

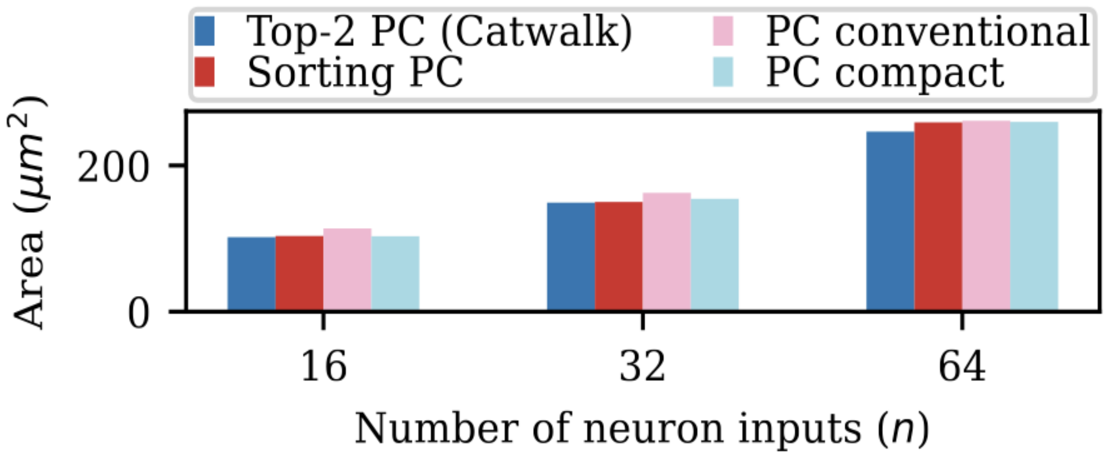
Unary Top-k

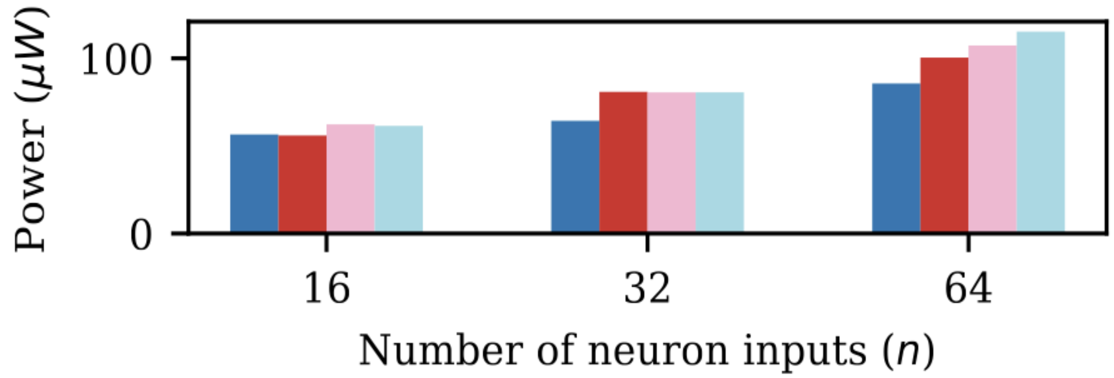


Dendrite



Post-Synthesis Results: Neuron





Gate count of dendrite adopting unary top-k and compact PC.

- **Neurons apply identical 5-bit accumulation and threshold implementation.**
- **Top-k uses optimal sorters, while sorting use bitonic sorters.**
- **Catwalk (Top-2 PC) improves area and power efficiencies by:**
  - **1.05x and 1.35x over the neuron with compact PC.**
  - **1.05x and 1.17x over the sorting-based neuron.**

Post-PnR Results: Neuron

Neuron design	Power (μW)			Area
	Leakage	Dynamic	Total	(μm^2)
$n = 16, k = 2$				
PC conventional	5.11	94.65	99.76	245.25
PC compact [7]	4.84	96.95	101.80	239.13
Sorting PC	4.28	70.11	74.39	197.64
Top-k PC (Catwalk)	4.22	69.40	73.62	194.98
$n = 32, k = 2$				
PC conventional	6.73	138.08	144.81	338.62
PC compact [7]	6.59	147.57	154.16	333.56
Sorting PC	5.73	88.24	93.97	256.42
Top-k PC (Catwalk)	5.66	86.79	92.45	252.97
$n = 64, k = 2$				
PC conventional	9.39	210.79	220.19	500.88
PC compact [7]	9.29	236.20	245.50	495.03
Sorting PC	8.12	129.59	137.71	364.15
Top-k PC (Catwalk)	7.85	124.21	132.06	355.38

- **Catwalk efficiency compared to PC Compact:**
  - **Area efficiency: 1.23x, 1.32x and, 1.39x for *n* = 16, 32 and, 64, resp.**
  - **Power efficiency by 1.38x, 1.67x and 1.86x for *n* = 16, 32 and, 64, resp.**
- **Generally, area-power efficiency scales with inputs *n*.**
- **Demonstrates importance of opting for top-k over sorting, despite the identical functionality.**

References

[1] Smith, James E. "Space-time algebra: A model for neocortical computation." 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2018.

[2] Smith, James E. "A temporal neural network architecture for online learning." arXiv preprint arXiv:2011.13844 (2020).

[3] Thorpe, Simon J., and Michel Imbert. “Biological constraints on connectionist modelling.” Connectionism in perspective (1989): 63-92.

[4] Nair, Harideep, John Paul Shen, and James E. Smith. "A microarchitecture implementation framework for online learning with temporal neural networks." 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). IEEE, 2021

[5] Chaudhari, Shreyas, Harideep Nair, José MF Moura, and John Paul Shen. "Unsupervised Clustering of Time Series Signals Using Neuromorphic Energy-Efficient Temporal Neural Networks." International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021

[6] Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. Orca: A distributed serving system for Transformer-Based generative models. In 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22), pages 521–538, 2022.

[7] Bert Dobbelaere. Smallest and Fastest Sorting Networks for A Given Number of Inputs, 2017. Accessed: 2025-03-06.

Thank you!

Any Questions?

Email: pvellais@andrew.cmu.edu