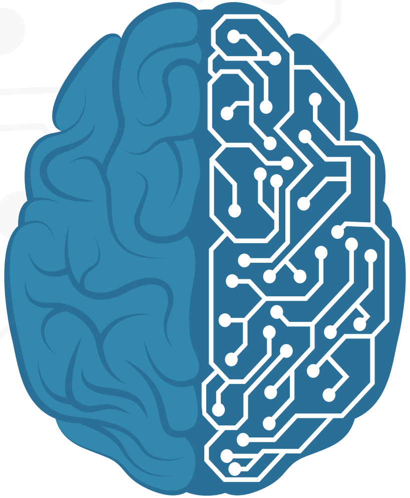


SpikeHard

Efficiency-Driven Neuromorphic Hardware for Heterogeneous Systems-on-Chip

Judicael Clair, *Guy Eichler, and Luca P. Carloni, Columbia University, New York, USA.*

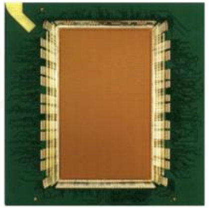
COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

CASES ’23

20th September 2023

Introduction

Popular Proprietary Neuromorphic Chips



TrueNorth [1]

Loihi 2 [2]

Akida [3]

- **Neuromorphic Computing** mimics biological brains.
- Human brain only consumes as much energy as a light bulb.
- Promising approach to energy-efficient embedded AI.

- We expect future **embedded neuromorphic apps** to depend on non-neuromorphic computations.
  - Sensory input pre-processing, e.g.:
    - Fast Fourier Transform (FFT) [4].
    - 2D Convolution (CONV2D) [5].

Heterogeneous Many-Accelerator Systems-on-Chip (SoCs)

[1] Akopyan et al. TrueNorth: Design and tool flow of a 65 mW 1 Million neuron programmable neurosynaptic chip. *IEEE TCAD* 34, 10 (2015), 1537–1557.
[2] Intel. https://download.intel.com/newsroom/2021/new-technologies/neuromorphic-computing-loihi-2-brief.pdf
[3] BrainChip. https://brainchip.com/wp-content/uploads/2023/03/BrainChip_second_generation_Platform_Brief.pdf
[4] Arsalan et al. 2022. RadarSNN: A resource efficient gesture sensing system based on mm-Wave radar. *T-MTT* 70, 4 (2022), 2451–2461.
[5] Chandarana et al. 2021. An adaptive sampling and edge detection approach for encoding static images for spiking neural networks. *In Proc. of IGSC*. 1–8.

Open-Source Alternatives

- **RANC: Reconfigurable Architecture for Neuromorphic Computing** [6]
  - Highly configurable at design time ➔ great for testing new architectural optimizations.
  - Similar architecture to TrueNorth and Loihi ➔ good algorithmic compatibility.
  - Deployable on FPGA ➔ fast prototyping.

- **SpikeHard (our work)** is based on RANC.

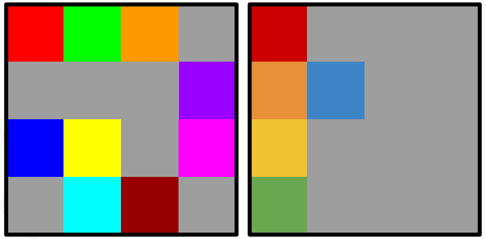
[6] Mack et al. 2021. RANC: Reconfigurable architecture for neuromorphic computing. *IEEE TCAD* 40, 11 (2021), 2265–2278.

Limitations of RANC that SpikeHard Solves

- **Lacks a standard interface**:
  - New interface required for integration in a heterogeneous many-accelerator SoC.

- **Not runtime-programmable**:
  - Model specified at design time ➔ immutable at run time.

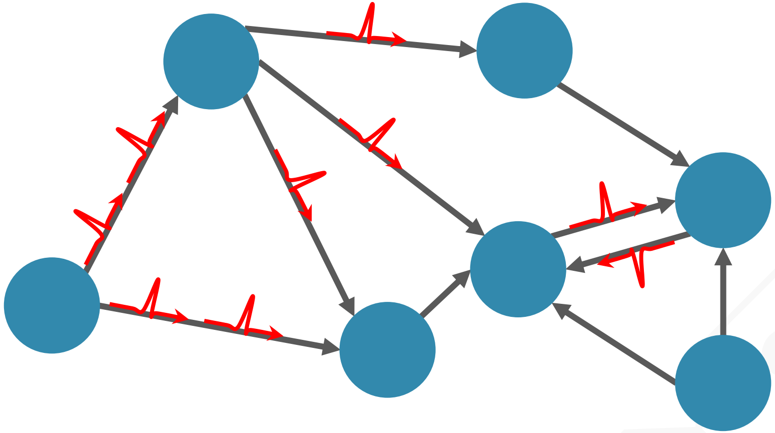
- **Model generation** tool includes **mapping model to hardware**. But,
  - Suboptimal resource utilization.
  - Performance, energy efficiency, and resource usage vary based on hardware architecture.
  - Hence, we created a new tool that:
    - Minimizes resource usage for a given architecture.
    - Enables architectural design-space exploration (DSE).



Grey ➔ unused resource
Colored ➔ used resource

Spiking Neural Network (SNN)

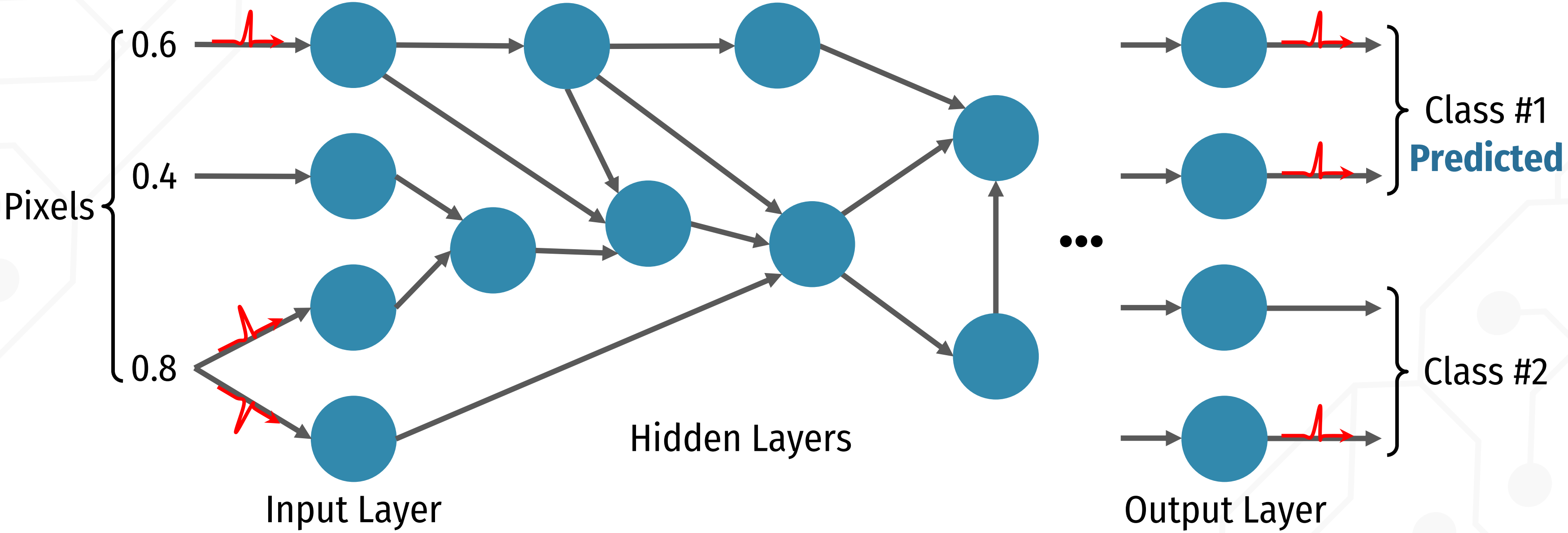
Learning model used in neuromorphic computing is the **Spiking Neural Network (SNN)**.



→	**Axon**	Communication channel
●	**Neuron**	Computational element
⎍→	**Spike**	Time-sensitive pulse (i.e. event) sent between neurons

Example SNN – MNIST Image Classification [7]

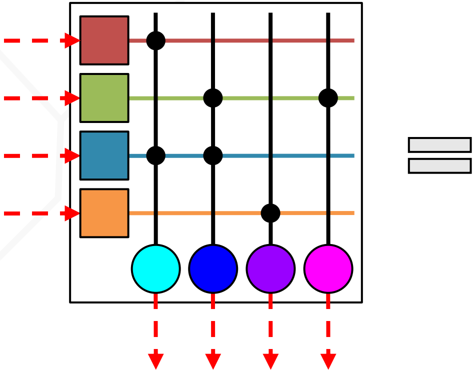
- For each greyscale pixel ∊ [0, 1], if rounded = 1, then spike generated for that pixel.
- Multiple neurons can receive spikes from same pixel.
- Each class assigned a set of neurons. Predicted class is the one with the most output spikes.



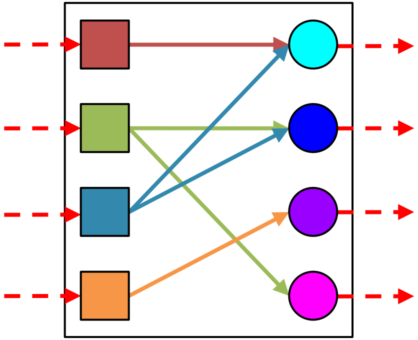
[7] Yepes et al. 2017. Improving Classification Accuracy of Feedforward Neural Networks for Spiking Neuromorphic Chips. *In Proc. of IJCAI.* 1973–1979.

Hardware Implementation

□	**Axon**	Communication channel
○	**Neuron**	Computational element
⇢	**Spike**	Time-sensitive event



Axon-Neuron Crossbar



Graph Representation

Neuromorphic Processor Architecture

- Multiple interconnected cores.
- Each core implements a crossbar.

Spike Timing

- Spike is received at a particular **tick**.
- **Tick Period**: Time elapsed between ticks.
  - Mainly depends on core architecture.
  - Minimize to maximize performance.

Model Restructuring

Model is already mapped to a hardware architecture (e.g. model generated by RANC).

Goal

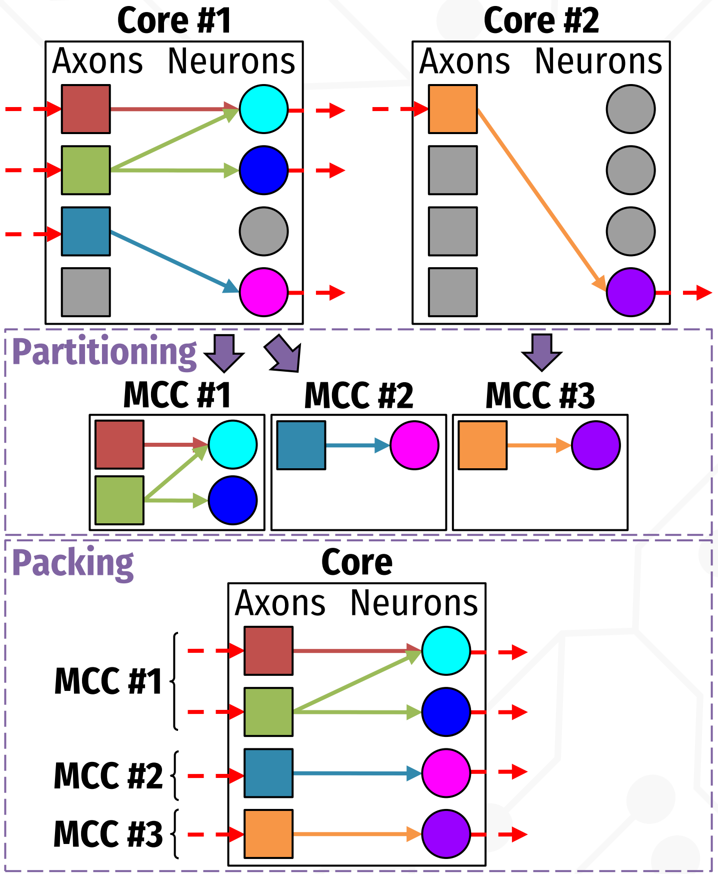
Use minimum number of cores.

Minimal Connected Components (MCCs)

Smallest disjoint subsets of connected neurons and axons in a core.

Integer Linear Program (ILP)

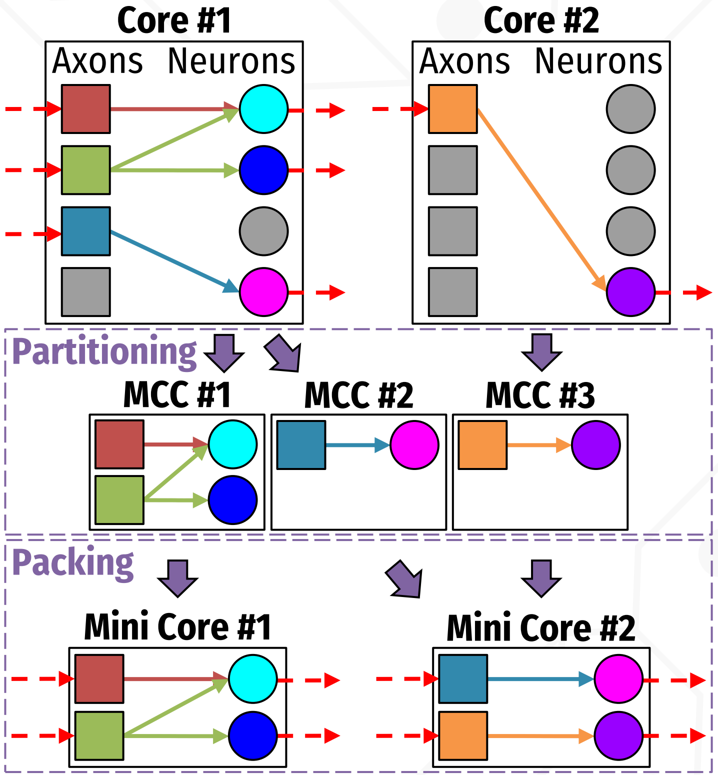
Problem similar to **bin packing**. Objective and constraints can be described as an ILP. Use standard ILP solver to find optimal solution.



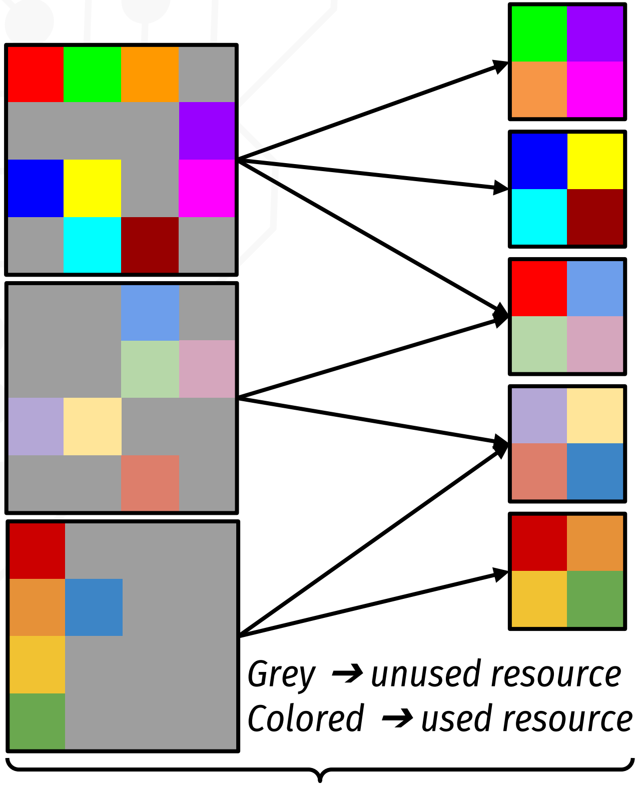
Model Restructuring

Additional Goal

Optimally remap to a different hardware architecture (e.g. smaller cores).



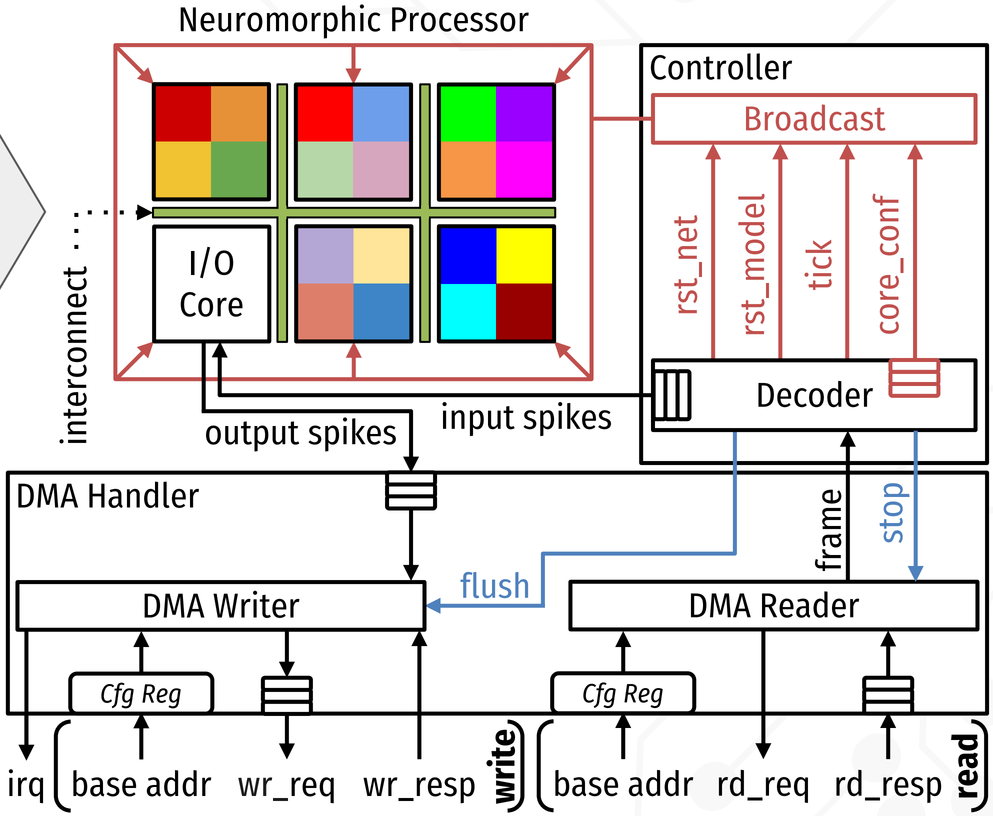
SpikeHard Accelerator



Grey → unused resource

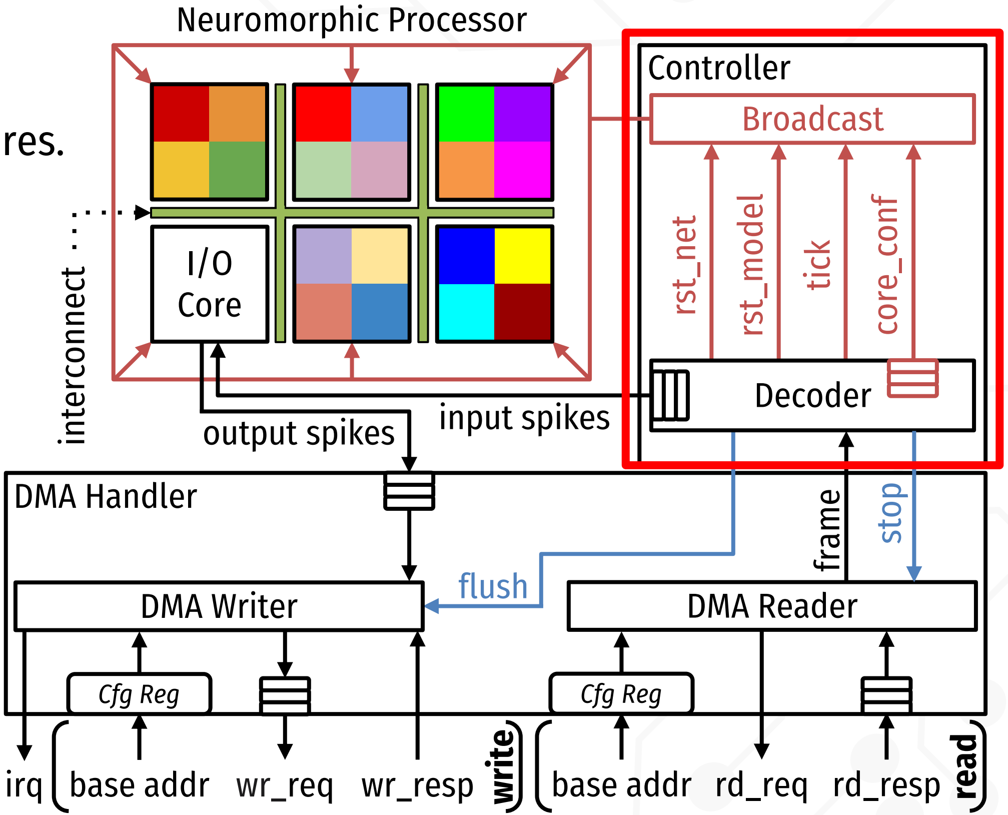
Colored → used resource

Model Restructuring

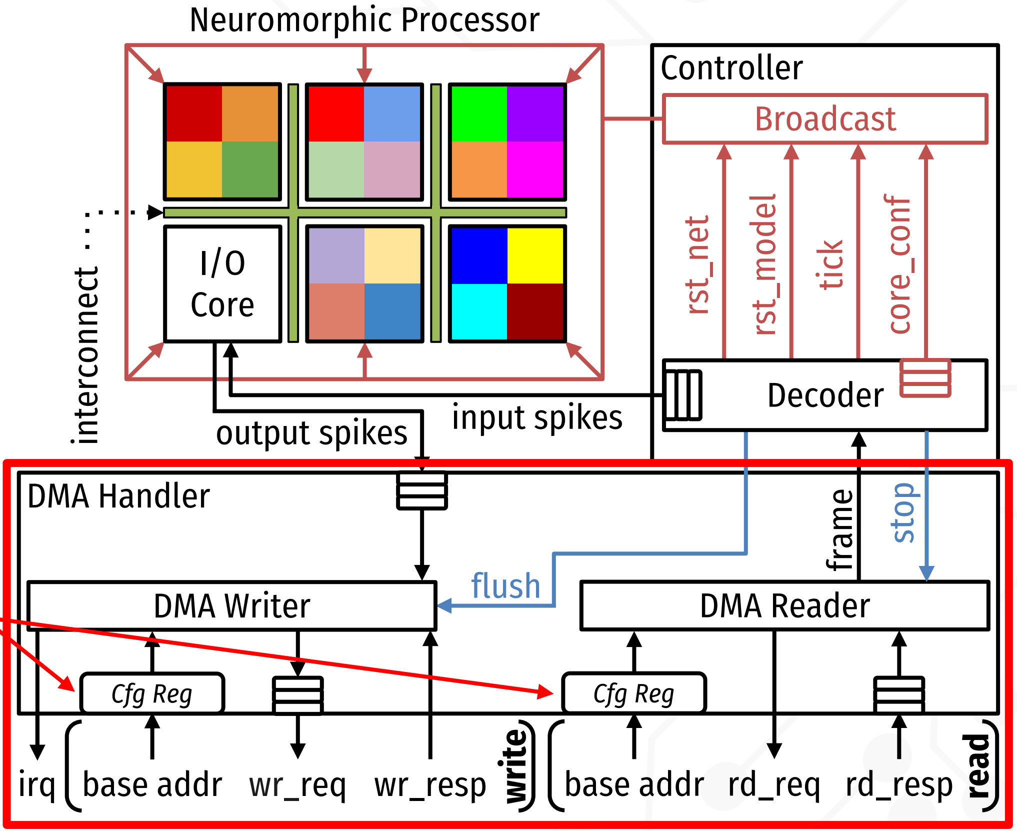


SpikeHard Accelerator

- Controller decodes input data.
- Commands are broadcasted to all cores.



SpikeHard Accelerator



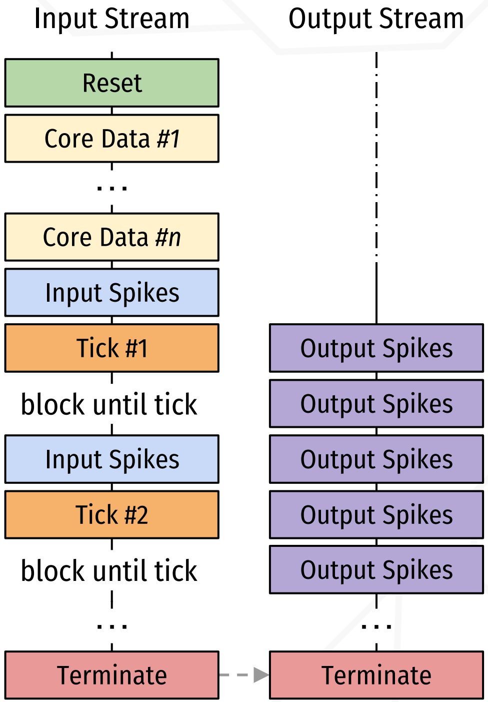
- Main memory accessed via DMA.
- Base addresses provided at the start.
- irq asserted at the end to notify CPU.

Accelerator Interface

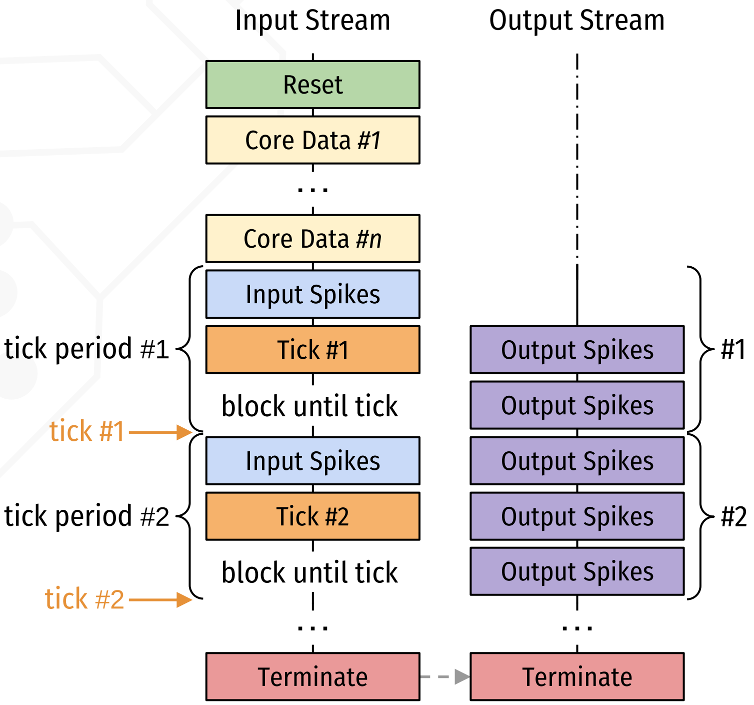
- **Input Stream of Frames**
  - Input spikes (e.g. input image to classify).
  - Commands (e.g. for model loading).

- **Output Stream of Frames**
  - Output spikes (e.g. predicted image class).

- **Frame**
  - 128-bit header:
    - 3 bits encode frame type.
    - Remaining bits frame-type specific.
  - Optional payload adjacent to header.



Accelerator Interface



Input Stream

Output Stream

Reset

Core Data *#1*

...

Core Data *#n*

Input Spikes

Tick #1

block until tick

Input Spikes

Tick #2

block until tick

...

Terminate

Output Spikes

Output Spikes

Output Spikes

Output Spikes

Output Spikes

...

Terminate

tick period #1

tick #1

tick period #2

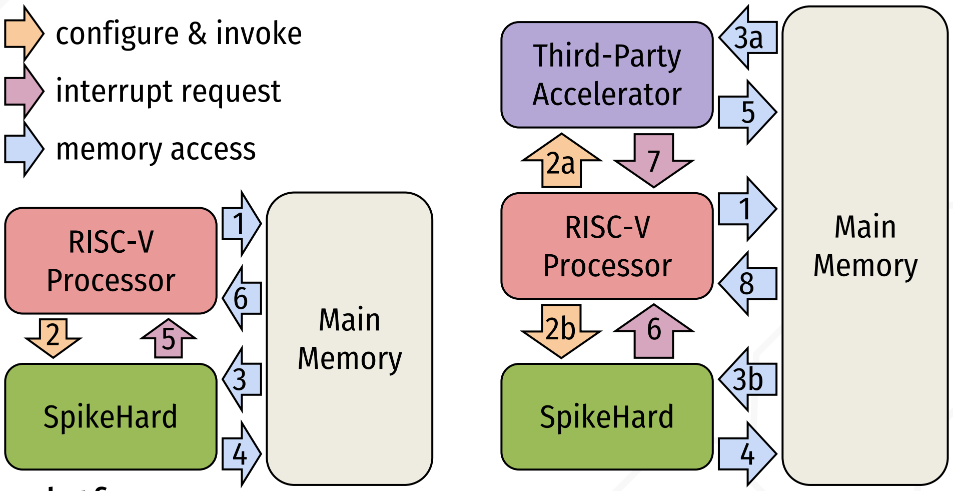
tick #2

#1

#2

Frame Type	Description
Reset	Reset dynamic state (`rst_net`) or unload model (`rst_model`).
Core Data	Configure model parameters for a given core (`core_conf`).
Input Spikes	Send spikes to specific axons (e.g. input image to classify).
Output Spikes	Output spikes for a given tick (e.g. predicted image class).
Tick	Proceed to next algorithmic time step (`tick`).
Terminate	End-of-File token.

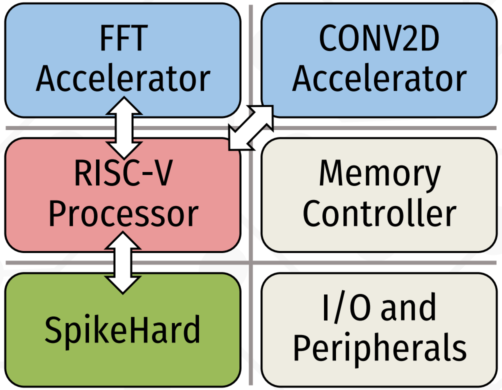
SoC Integration



We used **ESP** [8], an open-source SoC design platform.

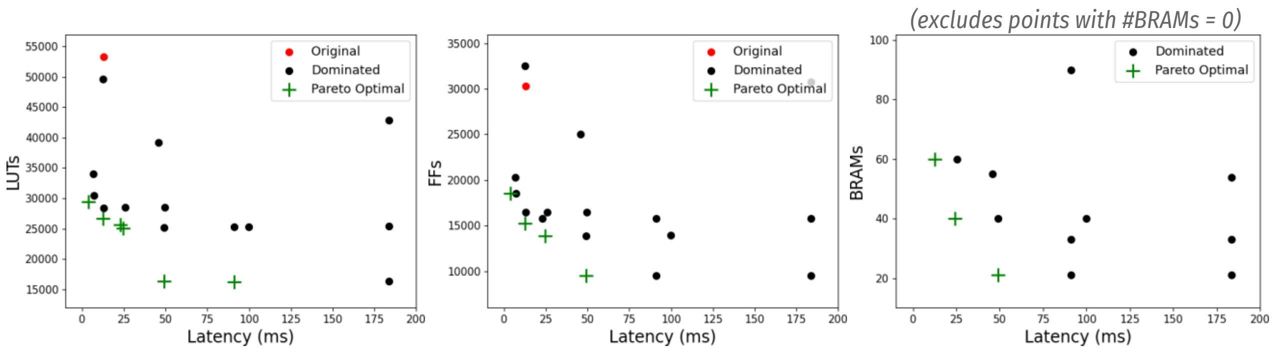
SpikeHard was integrated as part of a standalone SoC containing:

- **Third-party accelerators**: FFT and CONV2D.
- **General-purpose 64-bit RISC-V CVA6 processor**:
  - Orchestrates accelerator execution.
  - Runs Linux from which app invokes the accelerators.
  - We tested parallel execution of SpikeHard, FFT, and CONV2D.
  - SpikeHard no longer a performance bottleneck after DSE.

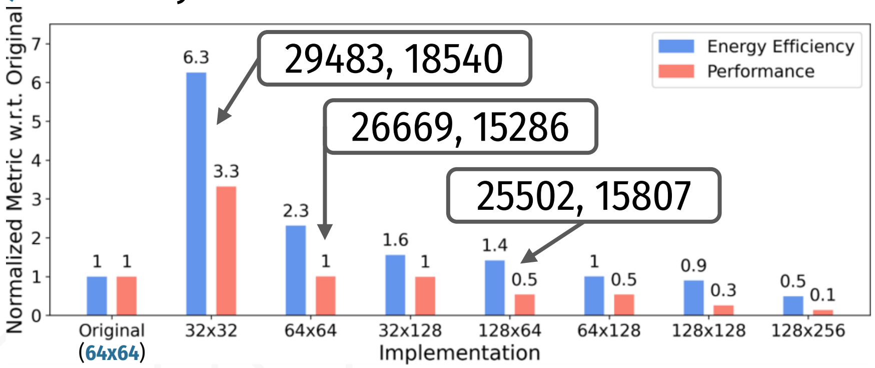


[8] SLD Group at Columbia University. https://www.esp.cs.columbia.edu/

Evaluation



- Deployed SoCs on the **Xilinx VCU128 FPGA** with 75 MHz clock frequency.
- Model estimates **Vector-Matrix Multiplication (VMM)** for a 6 by 6 matrix.



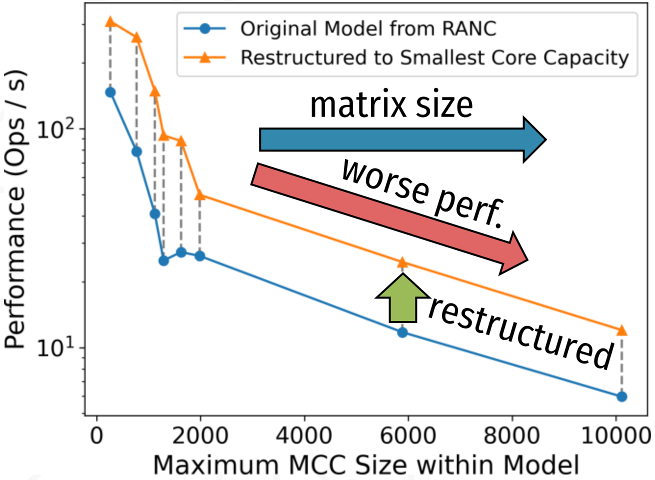
#LUTs, #FFs (, #BRAMs = 0)

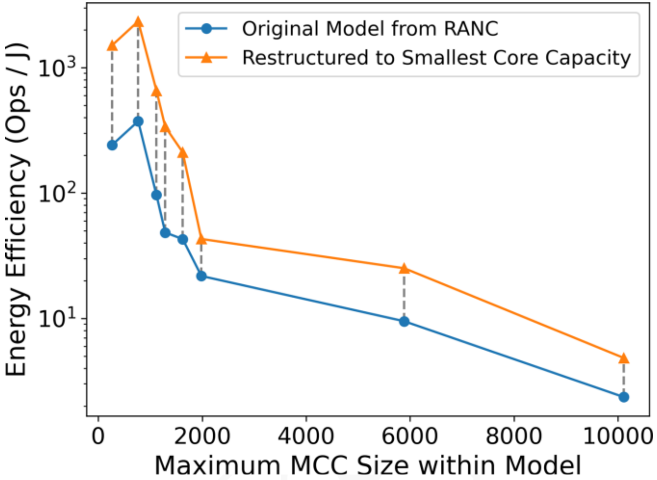
- **Original mapping from RANC**: 64x64.
- **AxN**: each core has **A** axons and **N** neurons.
- Varied **A** & **N** via model restructuring.
- Tested: ∀**A**, **N** ∊ {32, 64, 128, 256, 512}.

- Larger cores ➔ better resource usage.
- Smaller cores ➔ better performance and energy-efficiency.
- Restructuring to smallest core capacity (32x32) improved:
  - Performance by **3.3x** (**89x**)
  - Energy efficiency by **6.3x** (**170x**)

} w.r.t. original with(out) tuned tick period.

Larger VMMs





- Restructuring to smallest core capacity improved:
  - Performance by **1.90** – **3.73x**.
  - Energy efficiency by **1.97** – **6.96x**.

- Larger matrix ➔ larger and more numerous MCCs (bigger model).
- Larger MCCs ➔ worse performance and energy-efficiency.

Conclusion

- We developed SpikeHard, a neuromorphic hardware accelerator that is:
  - Programmable at runtime.
  - Easy to integrate into a heterogeneous many-accelerator SoC.
    - Suitable for embedded apps with both neuromorphic and non-neuromorphic kernels.

- We devised an optimization algorithm (model restructuring) that:
  - Minimizes resource utilization for a particular neuromorphic architecture.
  - Enables a model to be optimally remapped to different architectures.

- We performed broad DSE on FPGA:
  - Significant improvements in performance and energy-efficiency over baseline.

- We have released the contributions of this work in the public domain:
  - https://github.com/sld-columbia/spikehard

Thank You!