

Optimizing Learning through Co-Design in Neuromorphic Computing

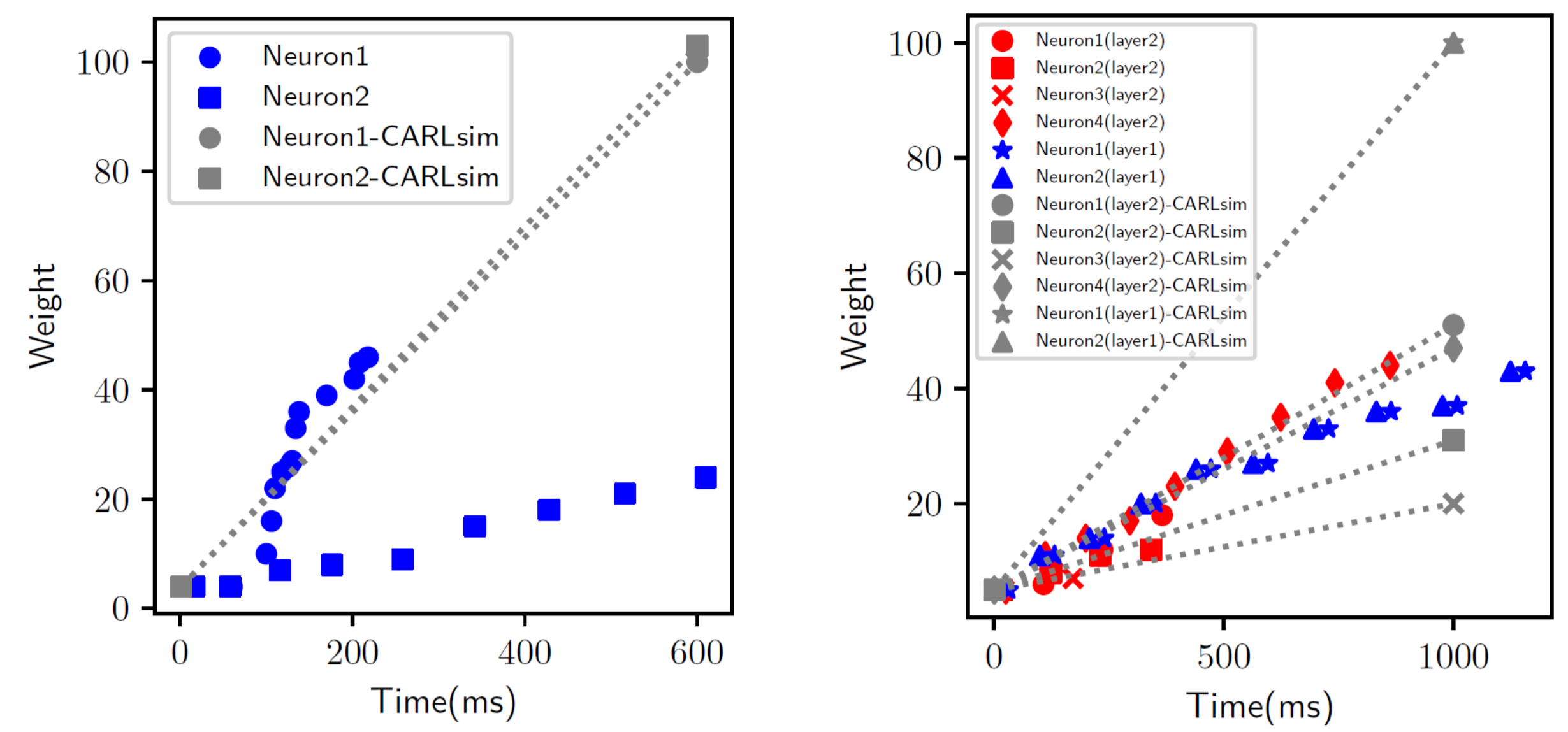
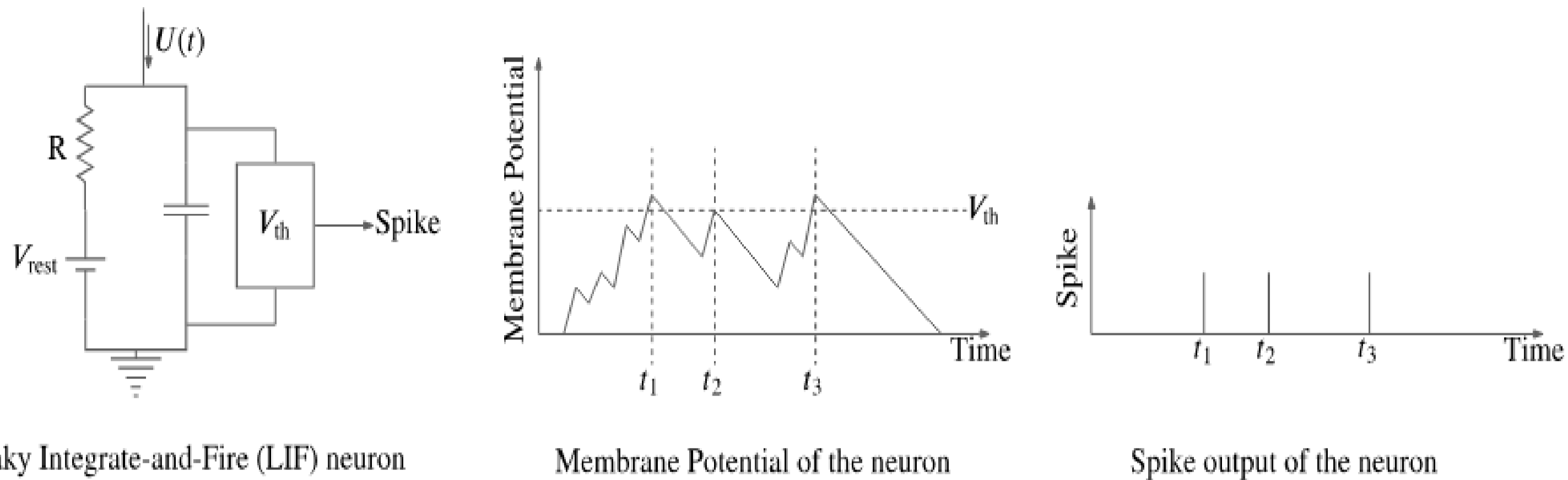


Lakshmi Varshika Mirtinti



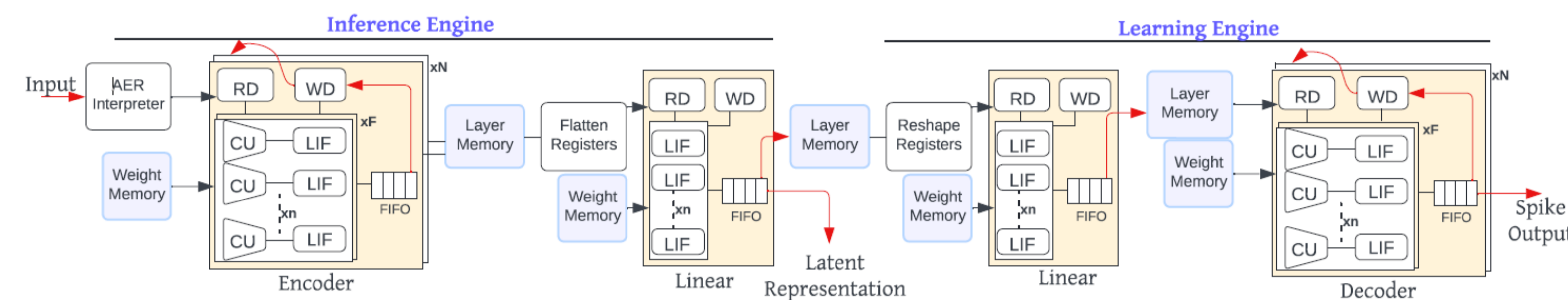
Abstract

- Neuromorphic systems, inspired by **spiking neural networks (SNNs)**, mimic biological neural systems to achieve superior **computational efficiency** and **lower power** consumption.
- **Unsupervised learning** enables pattern and feature extraction from unstructured, unlabeled data prevalent in real-world applications.
- **Co-design** aligns hardware and software from the start, optimizing the interaction between the two.



The final weight update of the **hardware** is similar to **software**.

Learning from Encoded Representation



- The **Variational Autoencoders (VAE)** encodes input data x in a high-dimensional space N into a lower-dimensional latent space z .
- **Encoder** outputs a distribution over latent space z and **Decoder** maps the latent variable back to reconstruct x .
- A novel framework combining **VAEs for capturing essential features** in low-dimensional latent spaces and **SNNs for energy-efficient, real-time data processing**.

TABLE I. Hardware utilization on a Xilinx Virtex-7 (XC7VX485T-2FFG1761C) FPGA.

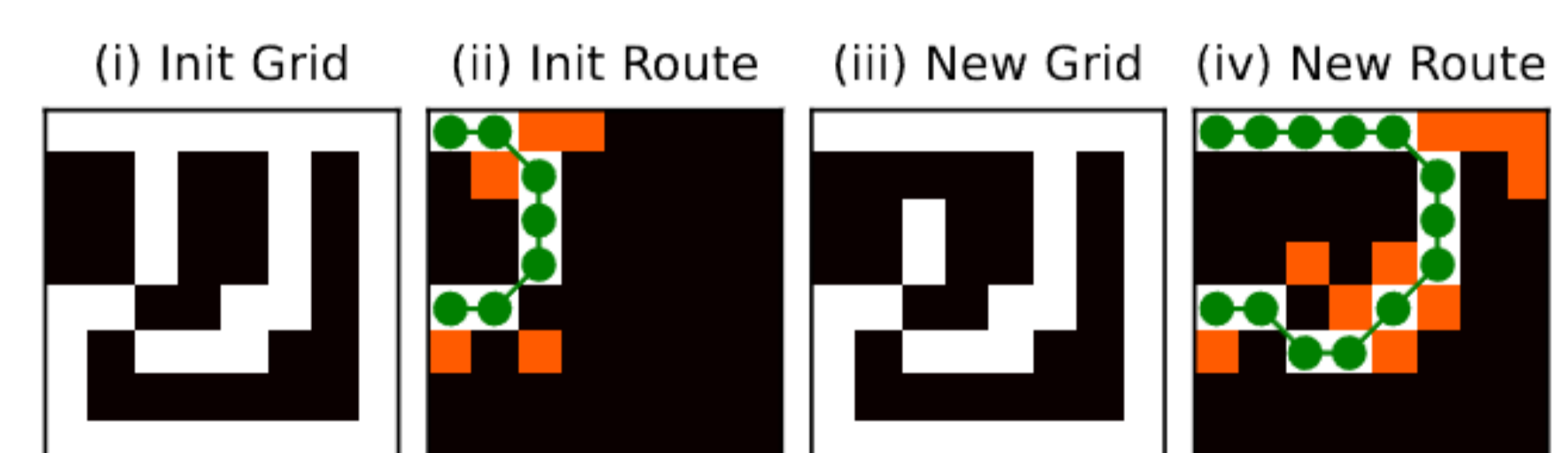
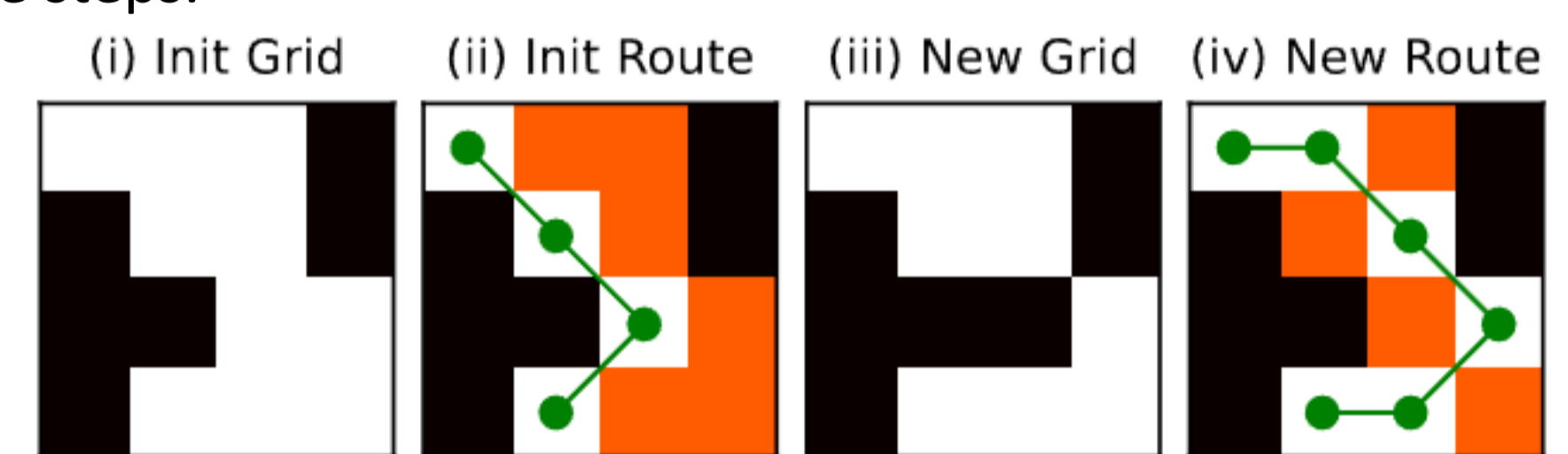
	Performance		Resource Utilization			
	Latency	Power	FF	LUT	DSP	BRAM
Baseline	131.3sec	2.29W	2288	13227	30	75
Hybrid	1.4sec	0.51W	2233	16138	2	28
SNN-VAE	5.2sec	2.20W	2802	5091	10	33

25.2x, 93.7x latency reduction for SNN-VAE, Co-design

1.04x, 4.5x energy savings for SNN-VAE, Co-design

Trace-based Learning

- The **Eligibility Propagation (E-Prop) algorithm** to continuously learn an environment and take a **detour** when necessary **in recurrent models** to process data sequentially.
- To create a spiking wavefront, we introduce a novel scoreboard mechanism that **continuously records synaptic activities** and combines them to generate input for subsequent time steps.



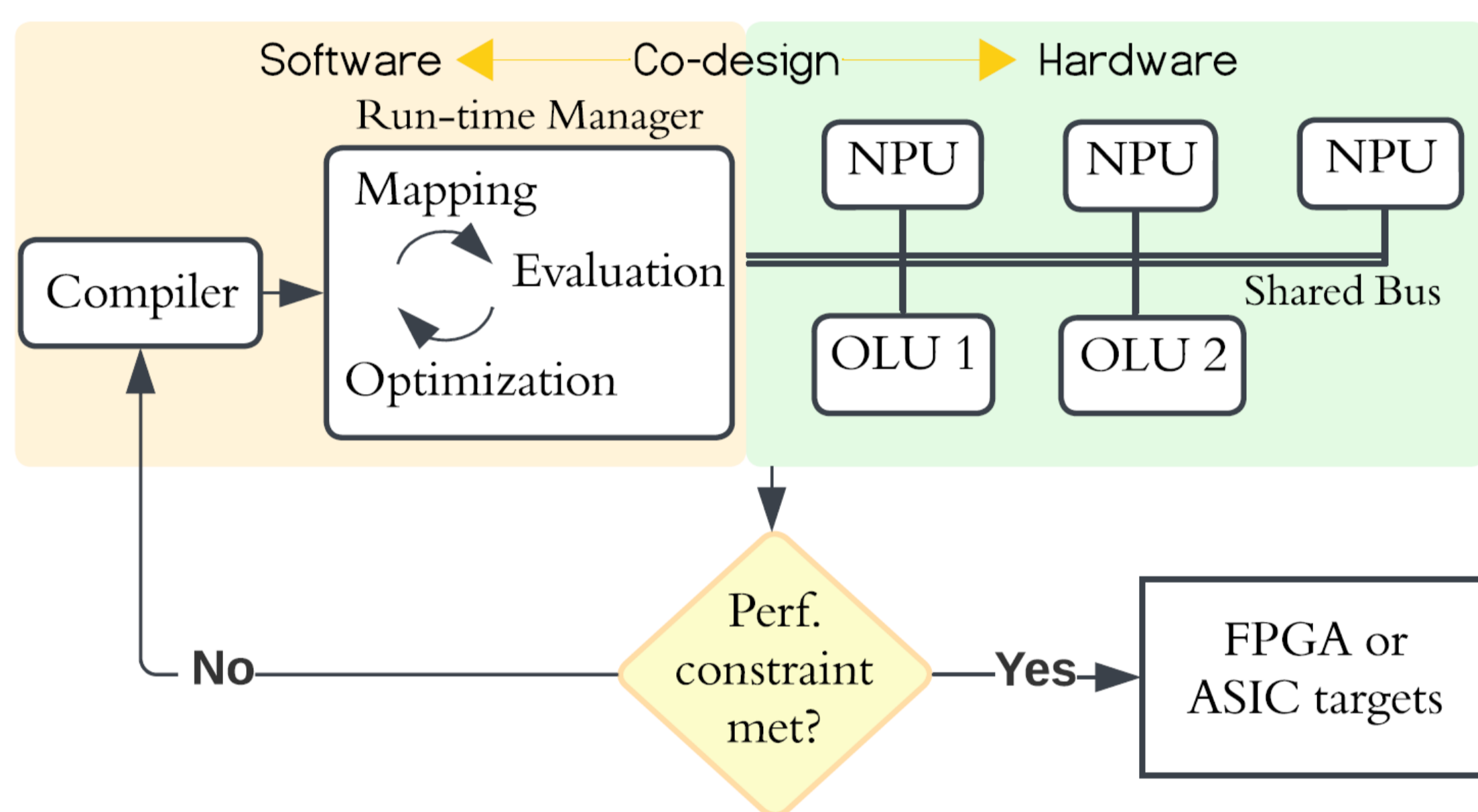
- The **Eligibility Propagation (E-Prop) algorithm** to continuously learn an environment and take a **detour** when necessary **in recurrent models** to process data sequentially.

Smaller grids take fewer iteration with 17% latency reduction

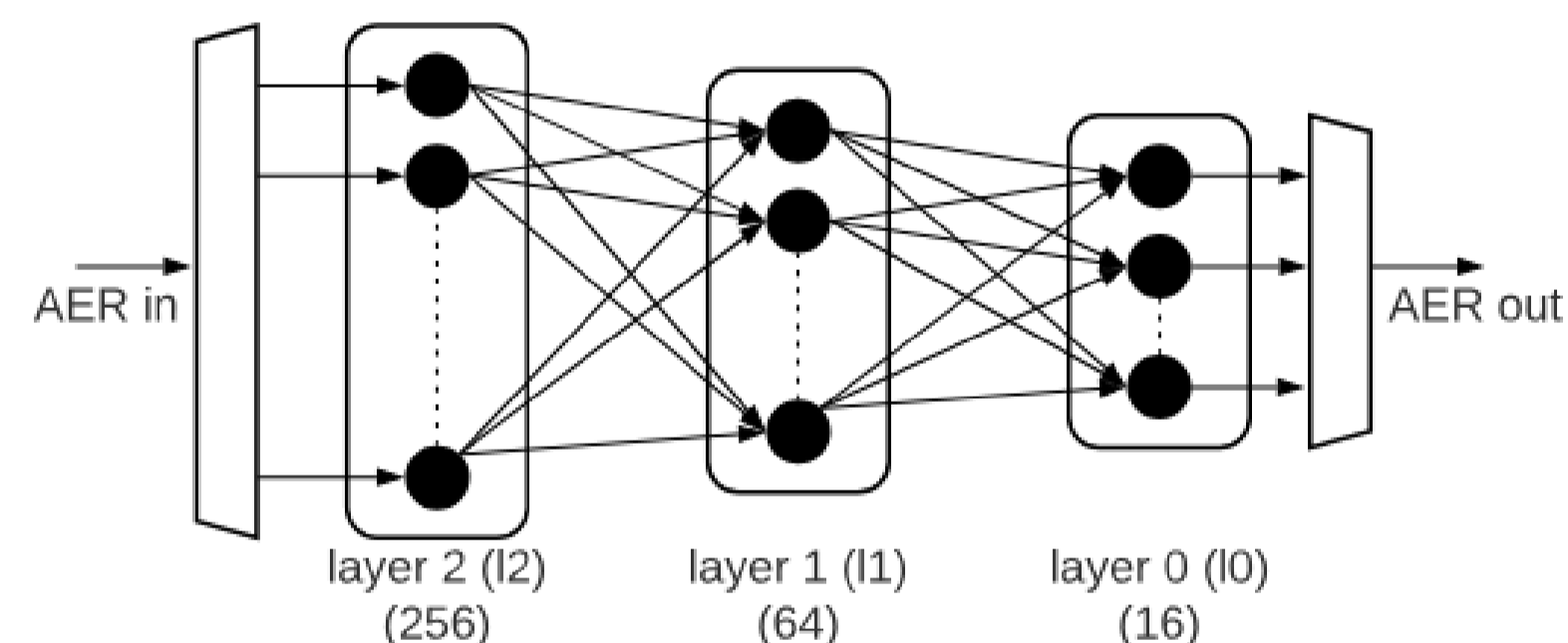
References

- [1] P. K. Huynh, M. L. Varshika, A. Paul, M. Isik, A. Balaji, and A. Das. "Implementing spiking neural networks on neuromorphic architectures: A review". In: arXiv preprint arXiv:2202.08897 (2022).
- [2] S. Johari, J. Dubey, and A. Das. "Design of a tunable astrocyte neuro-morphic circuitry with adaptable fault tolerance". In: 2023 IEEE 66th International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, 2023, pp. 904–908.
- [3] S. Song, M. L. Varshika, A. Das, and N. Kandasamy. "A designflow for mapping spiking neural networks to many-core neuromorphic hardware". In: 2021 IEEE/ACM International Conference On Computer-Aided Design (ICCAD). IEEE, 2021, pp. 1–9.
- [4] M. L. Varshika, A. Balaji, F. Corradi, A. Das, J. Stuijt, and F. Catthoor. "Design of many-core big little μ Brains for energy-efficient embedded neuromorphic computing". In: 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2022, pp. 1011–1016.
- [5] M. L. Varshika, F. Corradi, and A. Das. "Nonvolatile memories in spiking neural network architectures: Current and emerging trends". In: Electronics 11.10 (2022), p. 1610.
- [6] M. L. Varshika, A. K. Mishra, N. Kandasamy, and A. Das. "Hardware-software co-design for on-chip learning in AI systems". In: Proceedings of the 28th Asia and South Pacific Design Automation Conference. 2023, pp. 624–631.
- [7] M. Varshika and A. Das. "Platform-Based Design of Embedded Neuromorphic Systems". In: Embedded Machine Learning for Cyber-Physical, IoT, and Edge Computing: Hardware Architectures. Springer, 2023, pp. 337–358.

μ BRAIN Digital Hardware Framework



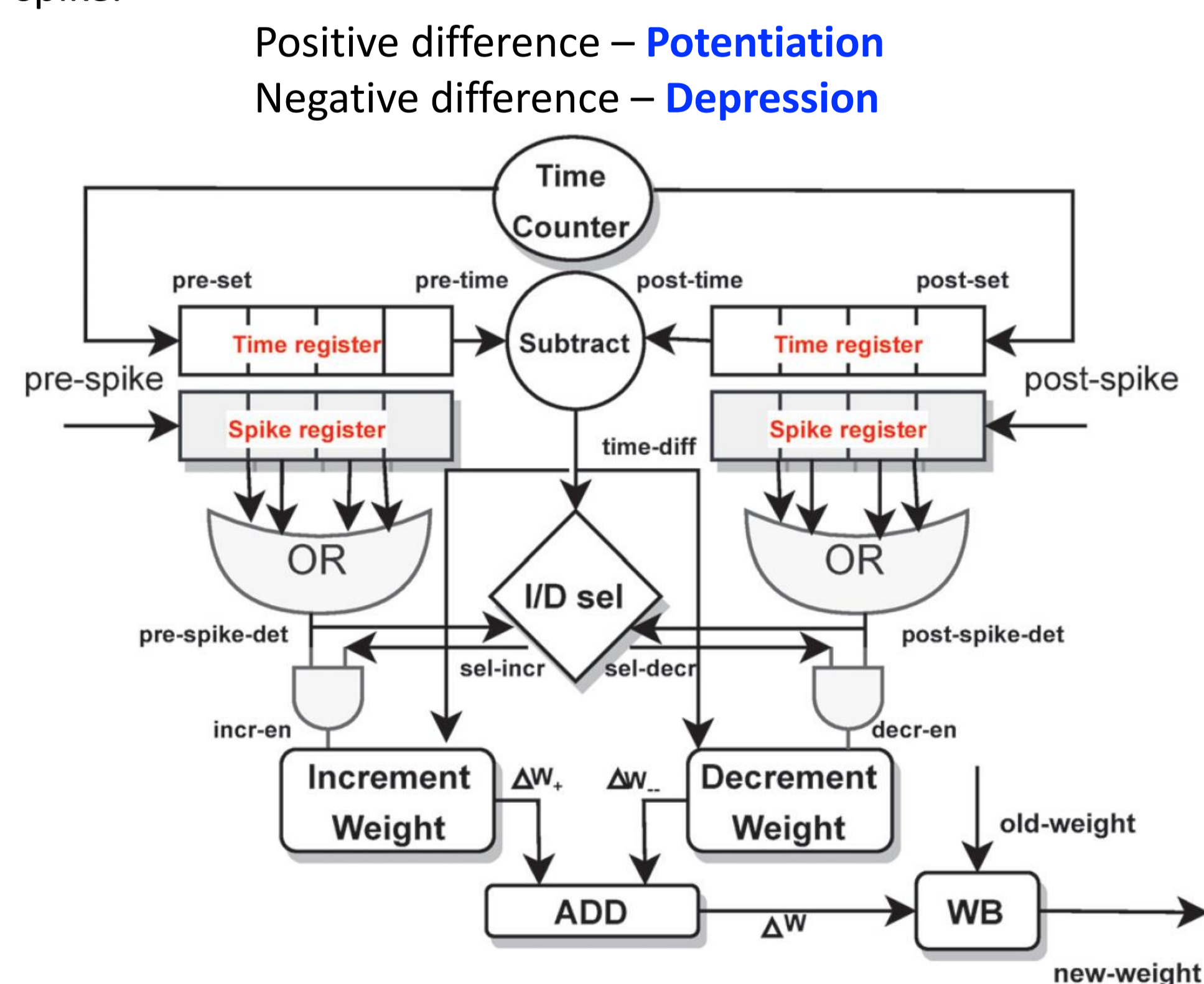
- An adaptive **co-design framework** that integrates **compiler-driven** partitioning and a dynamic **runtime management** system.
- The hardware architecture consists of **multi-core Neural Processing Units (NPUs)** and **On-Chip Learning Units (OLUs)** tailored to specific learning mechanisms, interconnected via a **shared bus** for efficient communication.



- NPU hardware is based on μ Brain's three-layered architecture.
- The **programmable synaptic connections** across its three layers.
- μ Brain operates asynchronously, with each neuron independently accumulating weighted synaptic spikes and generating an output spike.

STDP Learning in MLP

- The OLU is designed as a separate computation unit to generate weight updates for an NPU according to the **STDP learning rule**.
- It updates the weight of a connection depending on the time difference between a pre-spike and a post-spike.



- Four **shift registers** to record pre-and post-spikes at a time represented by the **time counter**.

$$\Delta W = \begin{cases} A_+ \exp(-\Delta t / \tau_+) & \text{if } \Delta t > 0 \\ A_- \exp(-\Delta t / \tau_-) & \text{if } \Delta t \leq 0 \end{cases}$$