EDGE AI Academy

Summer School

7-8 July 2025, Pisa, Italy

Edge AI Academy Summer School

Lowering barriers for neuromorphic computing at the micro-edge.

Paolo Meloni, Università degli Studi di Cagliari, Sardinia

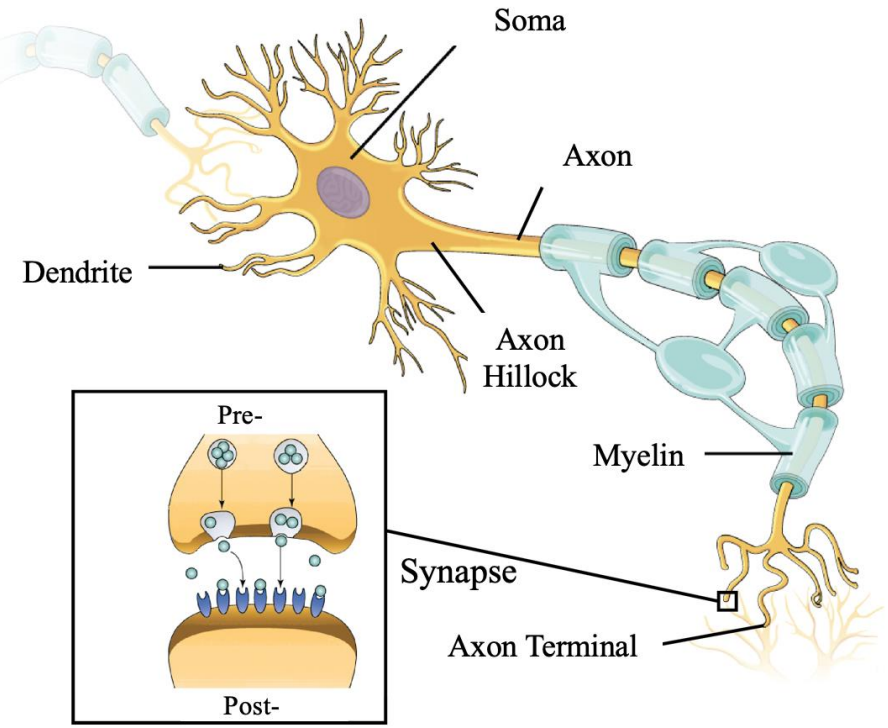
7-8 July 2025 Pisa, Italy

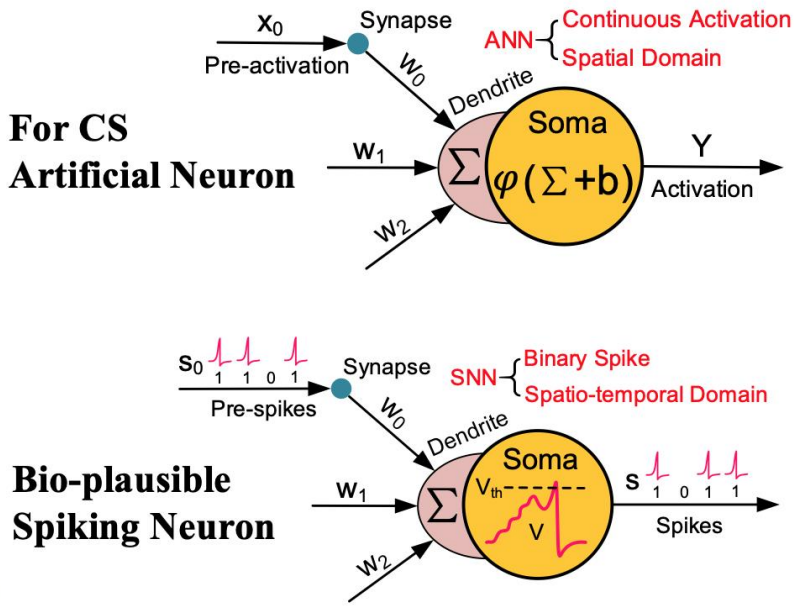
Outline

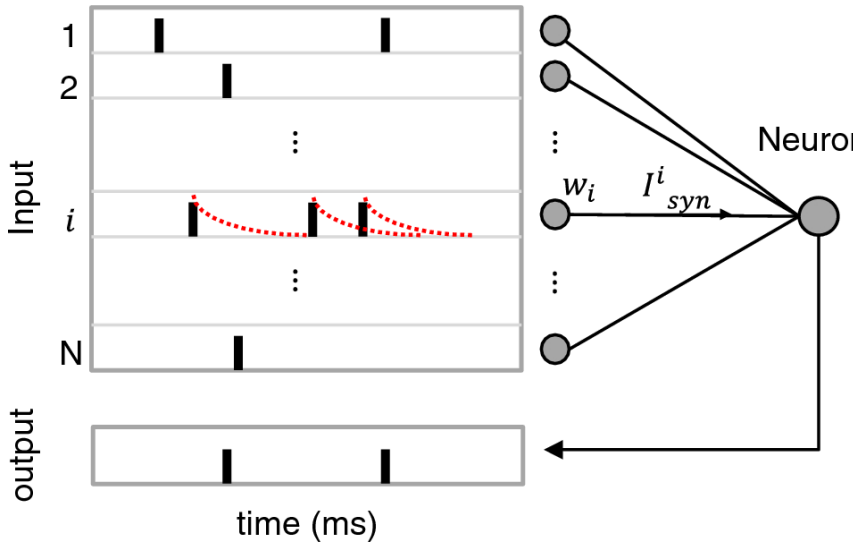
- Neuromoprhic computing and SNNs generalities
- Easy-access SNN processing
  - FPGA
    - Architecture and use-cases
  - Open-HW approach

Copyright © 2025

Spiking neural networks and spiking neurons







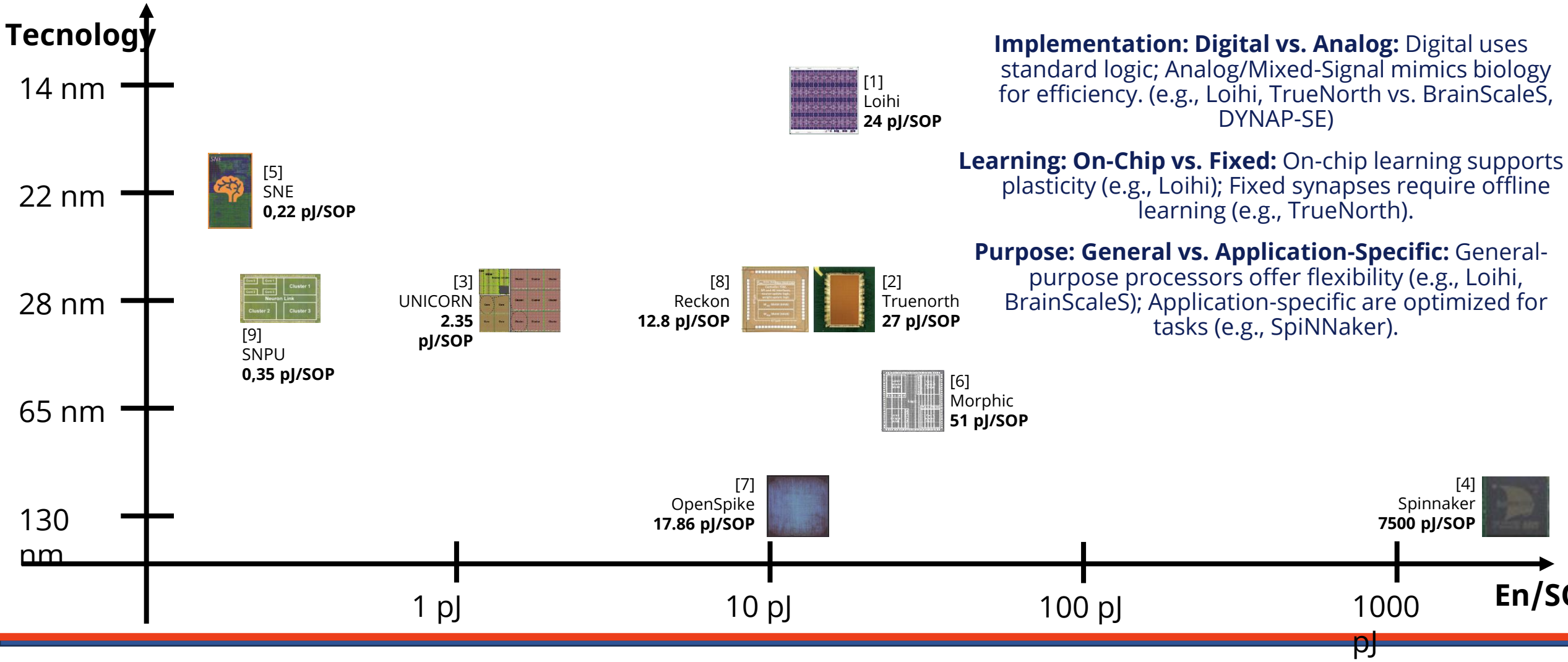
Advanced SNN = ANN + Neuronal Dynamics

- Sparse computations
- Energy efficiency

$$i(t) = \sum ws(t-1);$$
$$v(t) = \alpha v(t-1) + i(t);$$
$$s(t) = v(t) > \theta;$$
$$v(t) = v(t) \cdot (1 - s(t))$$

Copyright © 2025

SNN Hardware



Implementation: Digital vs. Analog: Digital uses standard logic; Analog/Mixed-Signal mimics biology for efficiency. (e.g., Loihi, TrueNorth vs. BrainScaleS, DYNAP-SE)

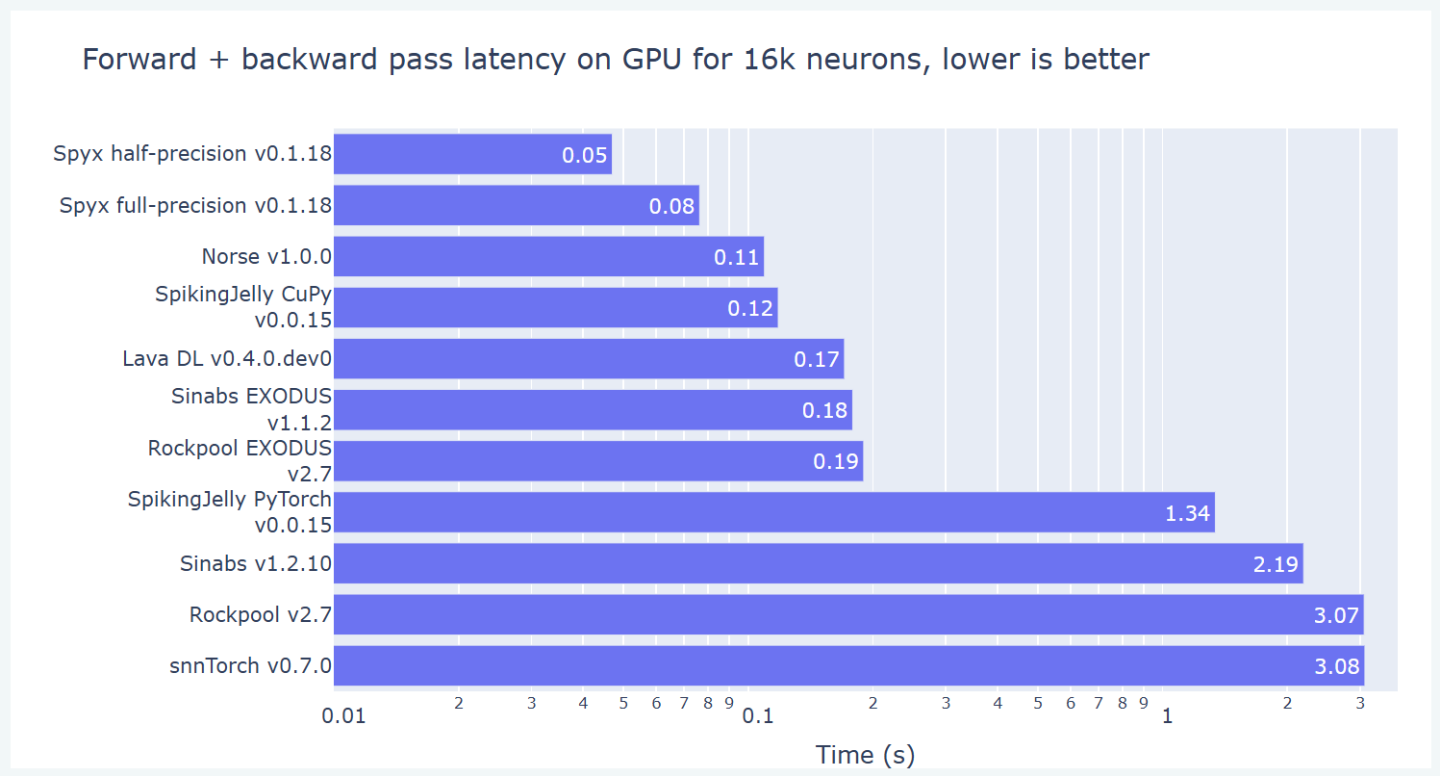
Learning: On-Chip vs. Fixed: On-chip learning supports plasticity (e.g., Loihi); Fixed synapses require offline learning (e.g., TrueNorth).

Purpose: General vs. Application-Specific: General-purpose processors offer flexibility (e.g., Loihi, BrainScaleS); Application-specific are optimized for tasks (e.g., SpiNNaker).

Copyright © 2025

SNN software

- backpropagation through time, time complexity scales with the number of time steps.
- slightly different extensions of deep learning frameworks



open-neuromorphic.org

Copyright © 2025

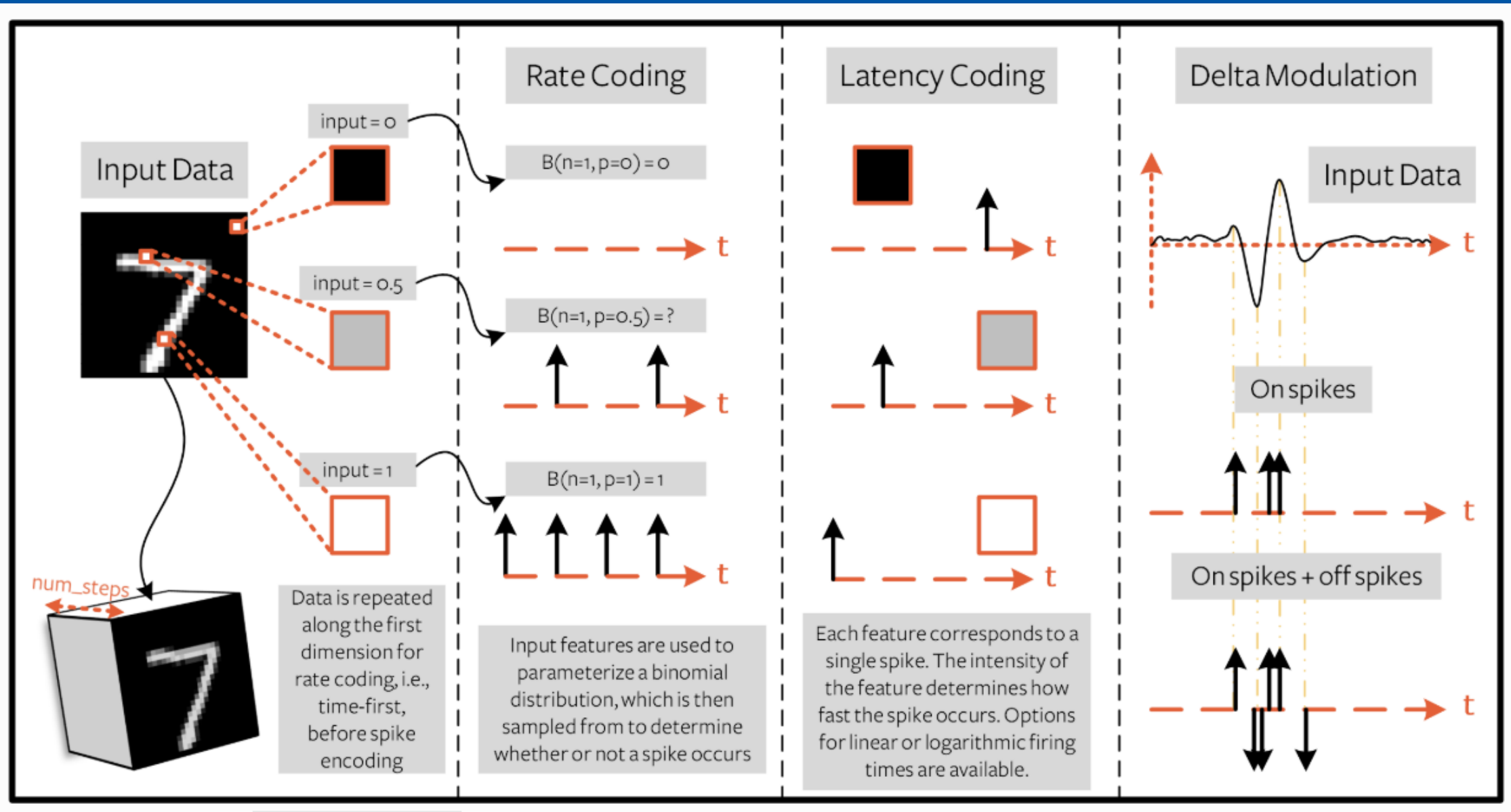
Training SNNs

```
############### simple SNN  #################

class SNN(nn.Module):
    def __init__(self, tau, input_channels=32, hidden_channels=64, n_classes=4):
        super().__init__()
        self.fc1 = nn.Linear(input_channels, hidden_channels, bias=False)
        self.bn1 = nn.BatchNorm1d(hidden_channels)
        self.lif1 = neuron.LIFNode(tau=tau, detach_reset=True)
        self.fc2 = nn.Linear(hidden_channels, n_classes, bias=False)
        self.bn2 = nn.BatchNorm1d(n_classes)
        self.lif2 = neuron.LIFNode(tau=tau, detach_reset=True)

        functional.set_step_mode(self, step_mode='m')
        if use_cupy:
            functional.set_backend(self, backend='cupy')

    def forward(self, x: torch.Tensor):
        B, _, _, T = x.shape
        x = torch.permute(x,(3,0,2,1)) # T,B,N,C
        x = self.fc1(x.flatten(0, 1)) # TB,N,C
        x = self.bn1(x. transpose(-1, -2)).transpose(-1, -2).reshape(T, B,x.shape[1],x.shape[2]).contiguous()
        x = self.lif1(x) # T,B,N,C
        x = self.fc2(x.flatten(0, 1)) # TB,N,C
        x = self.bn2(x. transpose(-1, -2)).transpose(-1, -2).reshape(T, B,x.shape[1],x.shape[2]).contiguous()
        x = self.lif2(x) # T,B,N,C
        return x.squeeze(2) # T,B,C

use_cupy=False
device = torch.device('cpu')
net = SNN(tau=2.0, input_channels=32, hidden_channels=64, n_classes =4)
net.to(device)
```

- Surrogate Gradients: snnTorch provides various surrogate gradient functions, such as fast_sigmoid, to approximate the derivative of the spiking
- Methods for handling temporal dynamics: e.g. Backpropagation Through Time (BPTT), Multi-step neurons
- Neuron Models: Supports computational neuron models like Leaky Integrate-and-Fire (LIF)
- Training Workflow: Users define networks using PyTorch modules, incorporate spiking neuron layers, and train using standard PyTorch optimizers and loss functions.

Copyright © 2025

Data to spikes - encoding

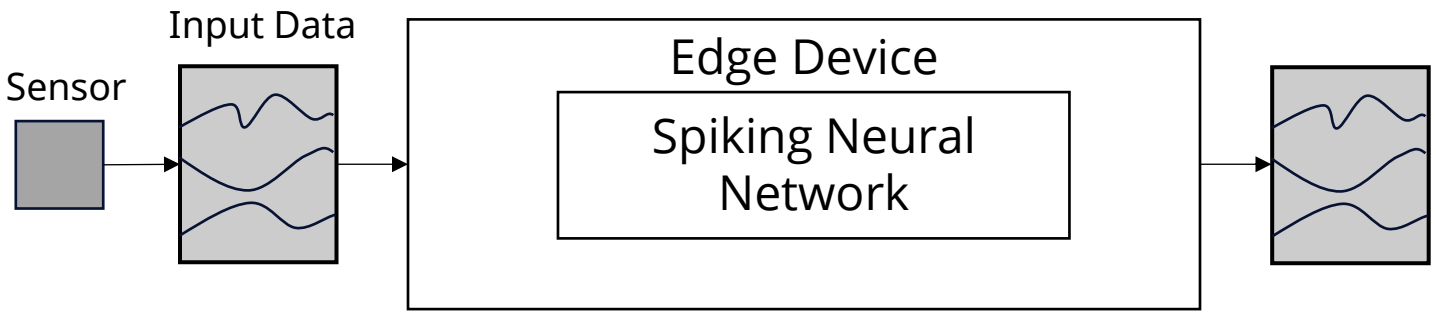


Copyright © 2025

Easy access to neuromorphic computing

Copyright © 2025

Easy-access approach to SNN processing



Target Platform

Low Power FPGAs

- Low-cost approach
- High static power consumption
- Scalability limited by hardware resources available

Open-Source ASICs

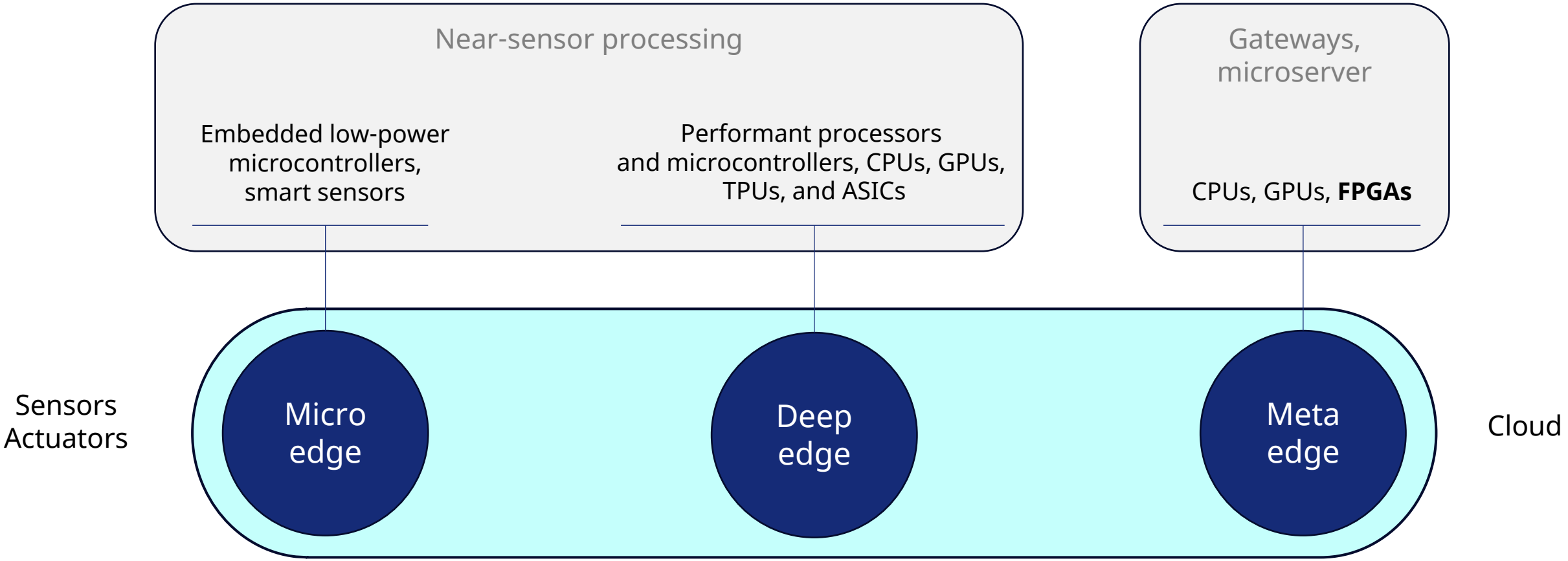
- Compromise between the accessibility of FPGAs and the high costs associated with proprietary EDA tools and PDKs for chip development
- Reliance on legacy technology nodes

Copyright © 2025

SYNtzulu: SNNs for near-sensor processing on FPGA

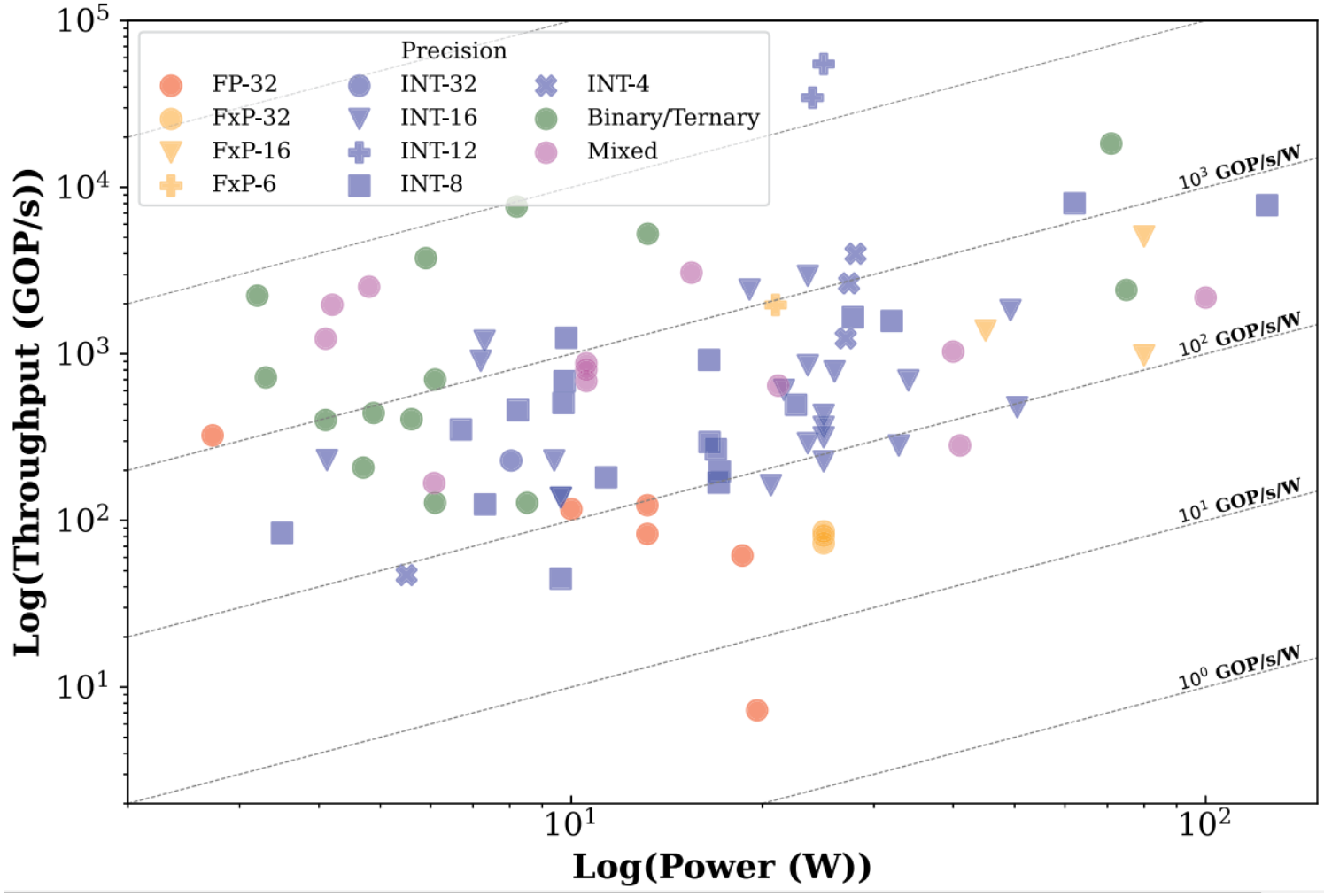
Copyright © 2025

Using FPGAs in the edge-to-cloud continuum



Copyright © 2025

FPGA-based AI Inference Accelerators

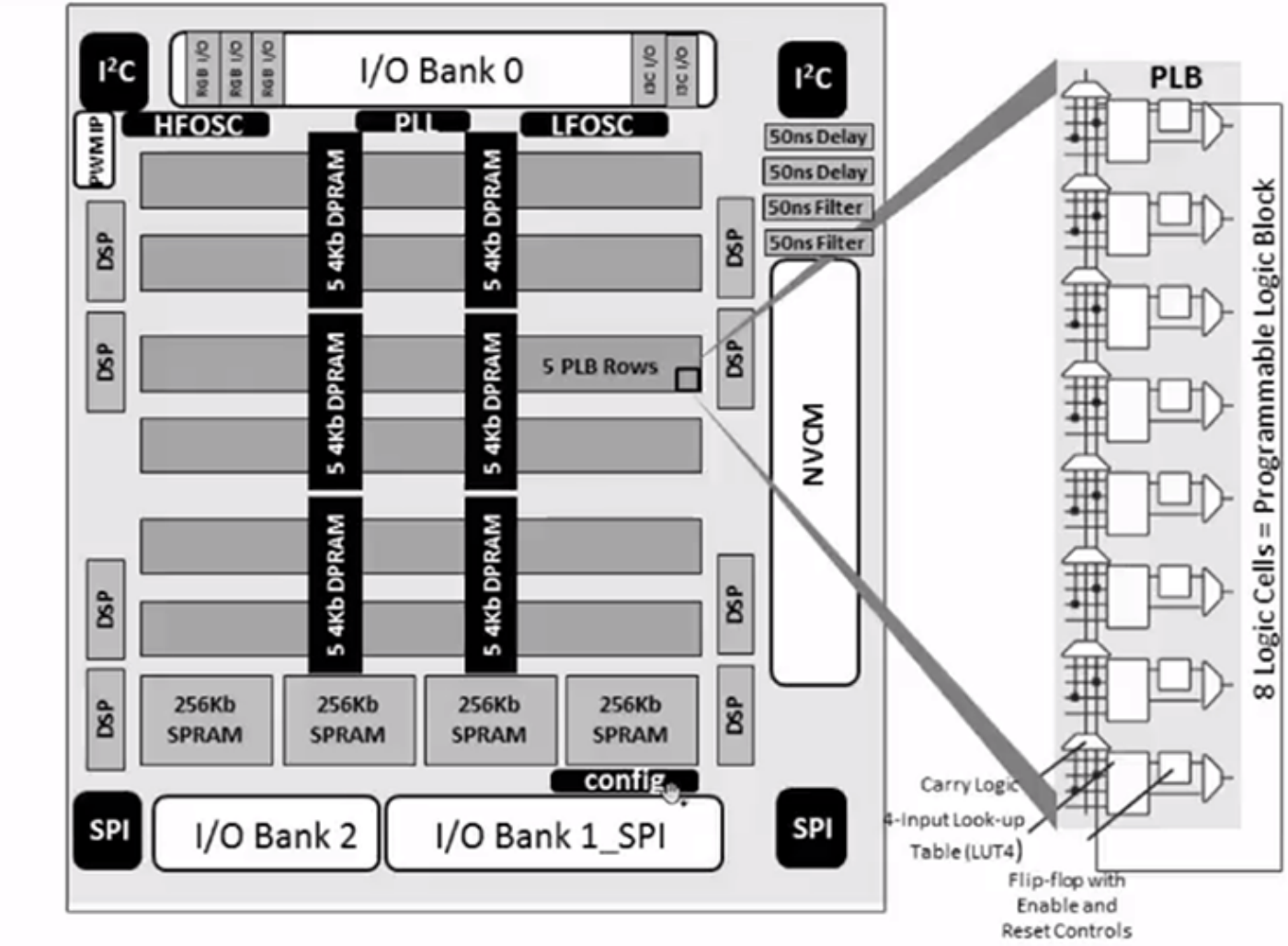


Anouar Nechi, Lukas Groth, Saleh Mulhem, Farhad Merchant, Rainer Buchty, and Mladen Berekovic. 2023. FPGA-based Deep Learning Inference Accelerators: Where Are We Standing?. ACM Trans. Reconfig. Technol. Syst. 16, 4, Article 60 (October 2023)

Copyright © 2025

SNNs on FPGA

- FPGA as target
  - Control modules near data-crunching logic
  - **Flexibility** to implement custom encoding and signal processing
  - **Affordable cost** compared to neuromorphic processors



Copyright © 2025

State-of-the-art: FPGA-based SNN Accelerators

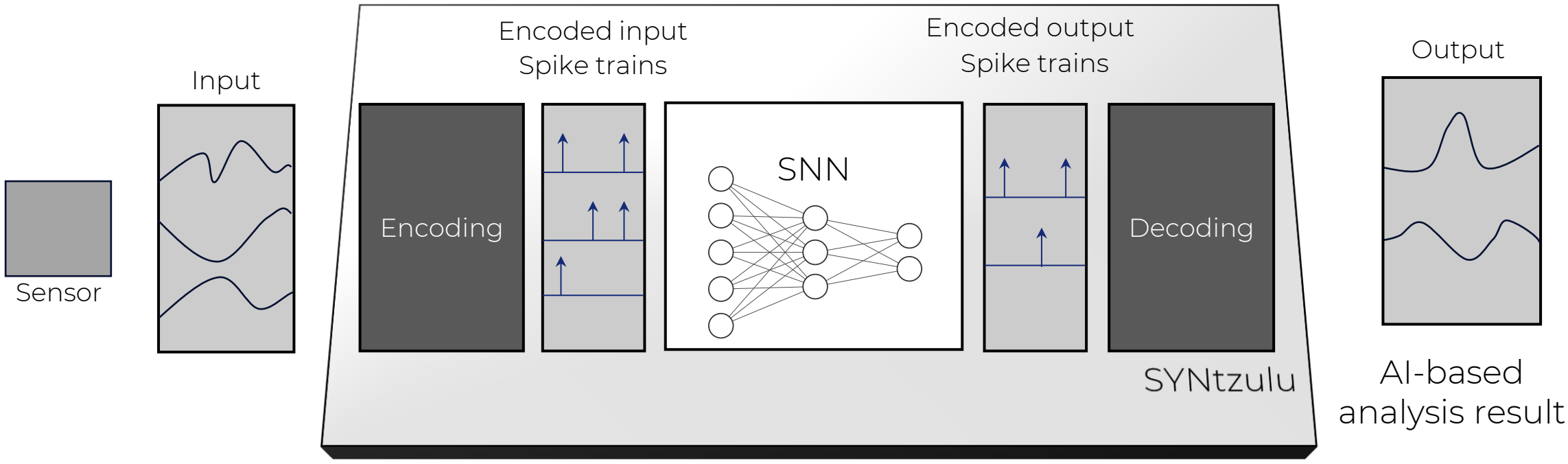
- Two main directions:
  - NN emulation for neuroscience
  - AI inference

	Optimized for	Frequency	FPGA target	Power
Li et al.	MNIST	100 MHz	Virtex7	1.6W
Liu et al.	MNIST	100 MHz	XA7Z020	280 mW
SyncNN	MNIST	200 MHz	Zynq (ZCU102)	400 mW
Spiker	MNIST	100 MHz	Artix7	-

- Image classification
- Mid-to-high end FPGA devices

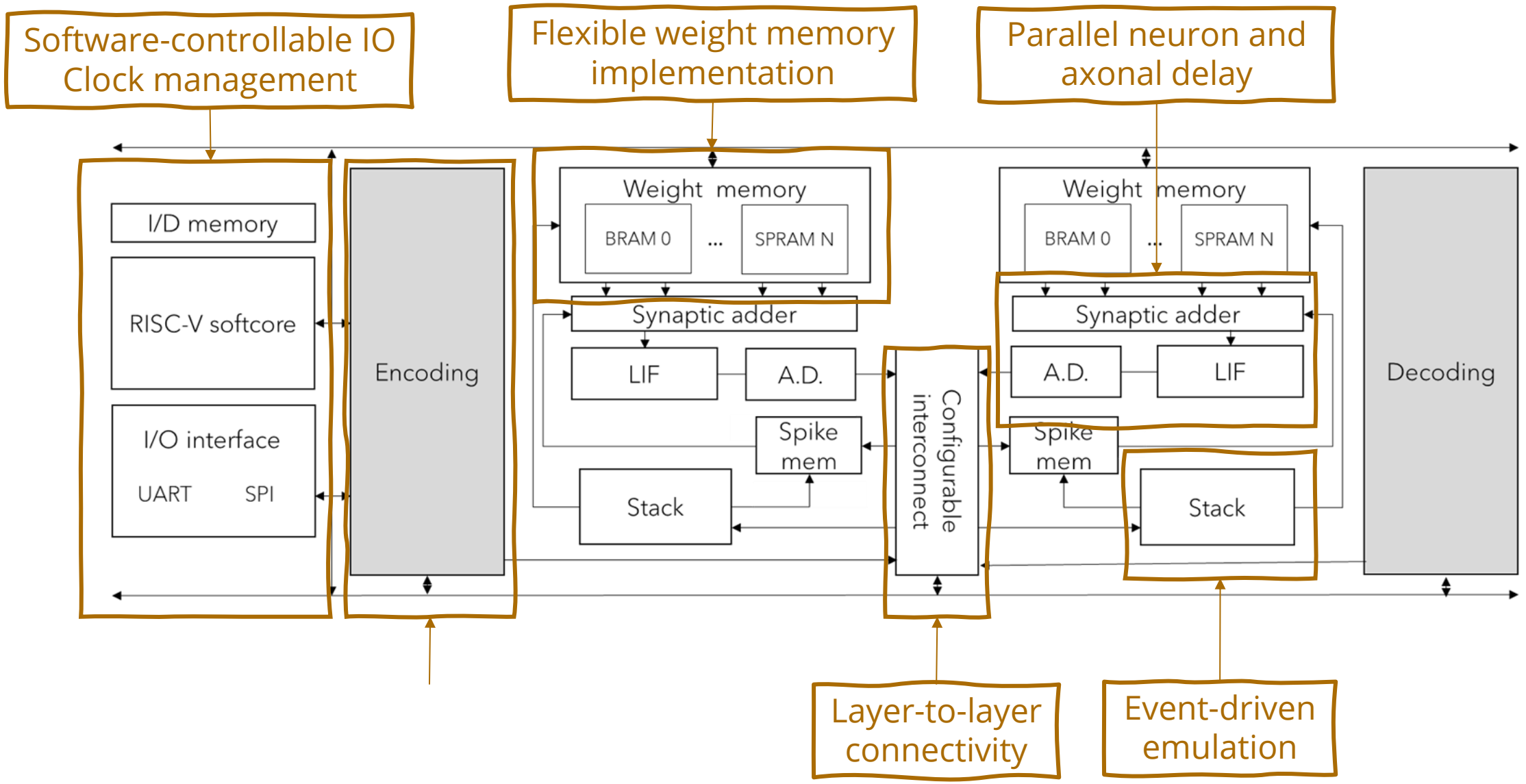
Copyright © 2025

Approach overview



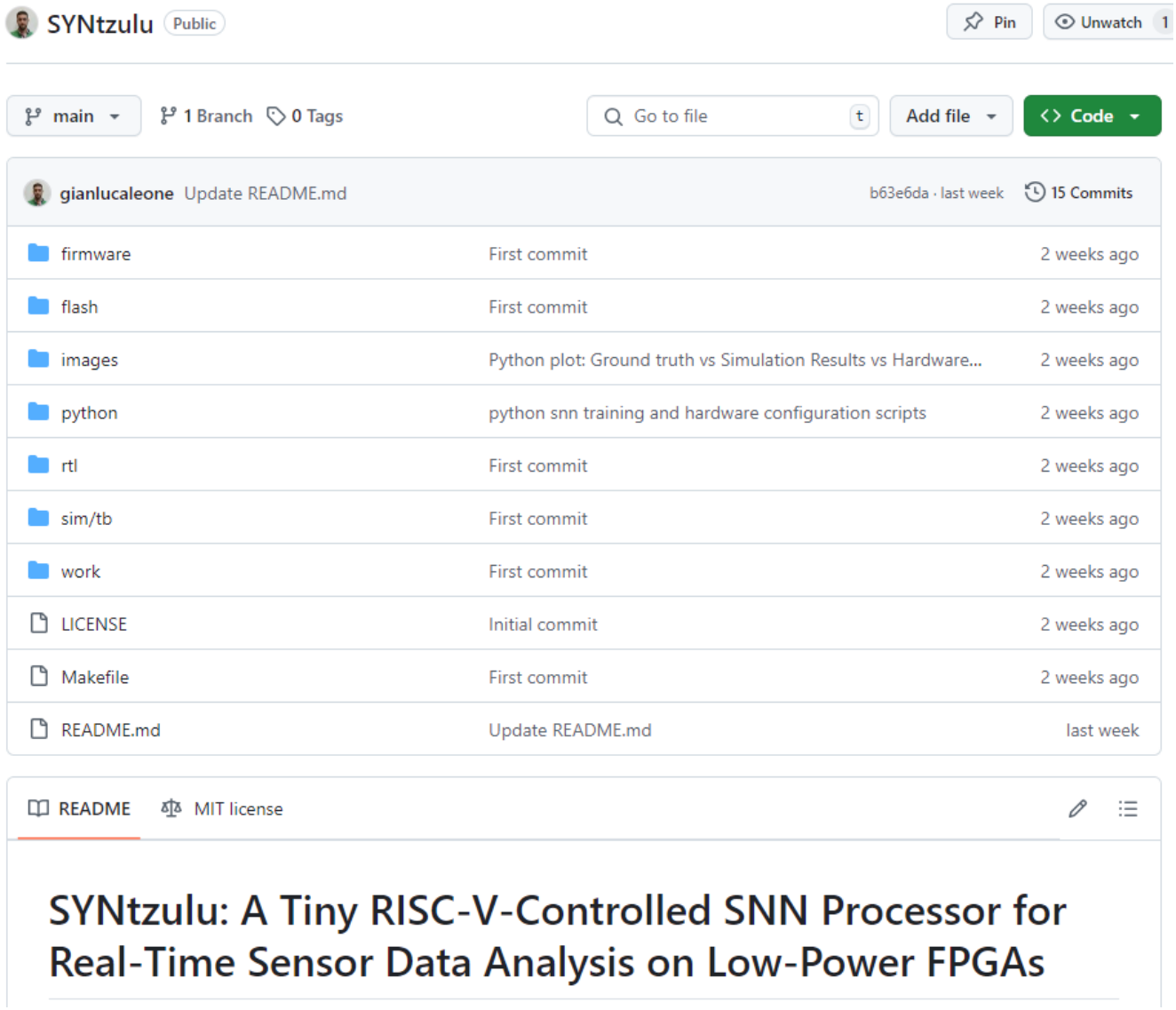
Copyright © 2025

SYNtzulu: architecture



Copyright © 2025

SYNtzulu is now free for you to download!



Try me!

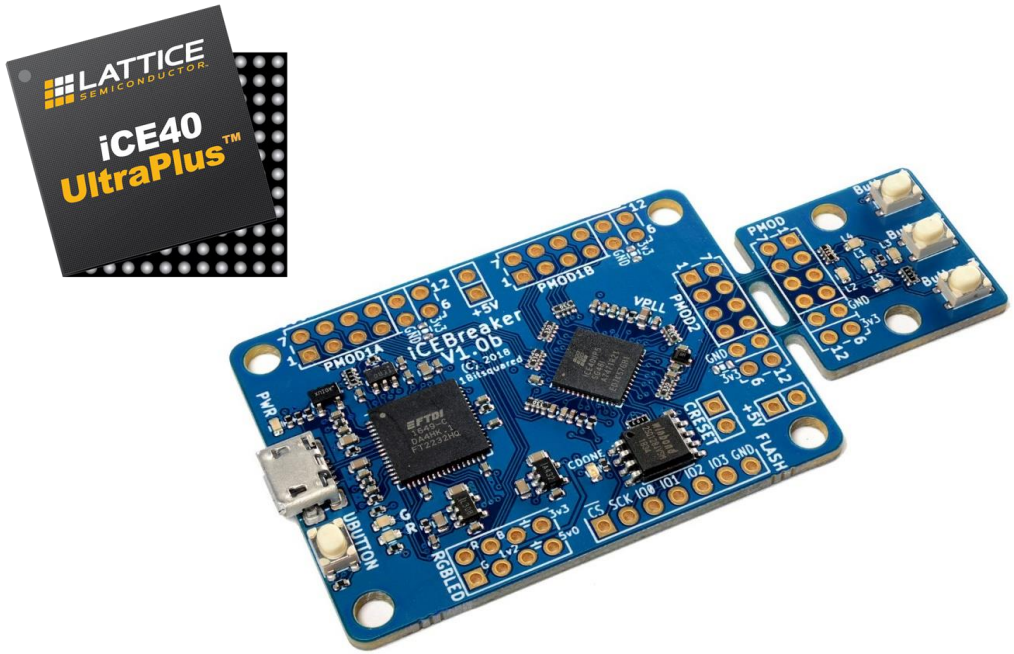
https://github.com/gianlucaleone/SYNtzulu

Copyright © 2025

Assessment: low-end implementation

Lattice iCE40UP5k FPGA

- **5280 logic cells** (4-LUT + Carry + FF)
- **8 DSP** multiplier blocks
- Around **1 Mbit** on-chip **SRAM**
- **PLL**, 2 x SPI, 2 x I2C hard IPs
- Two internal **oscillators** (10 kHz and 48 MHz)



	LUT	DSP	BRAM	SPRAM
UC1	4484 (84%)	4 (50%)	22 (73%)	4 (100%)
UC2	4690 (89%)	3 (37%)	28 (93%)	4 (100%)
UC3	4262 (80%)	2 (25%)	29 (90%)	4 (100%)

Copyright © 2025

USE CASES

iEEG

ECG

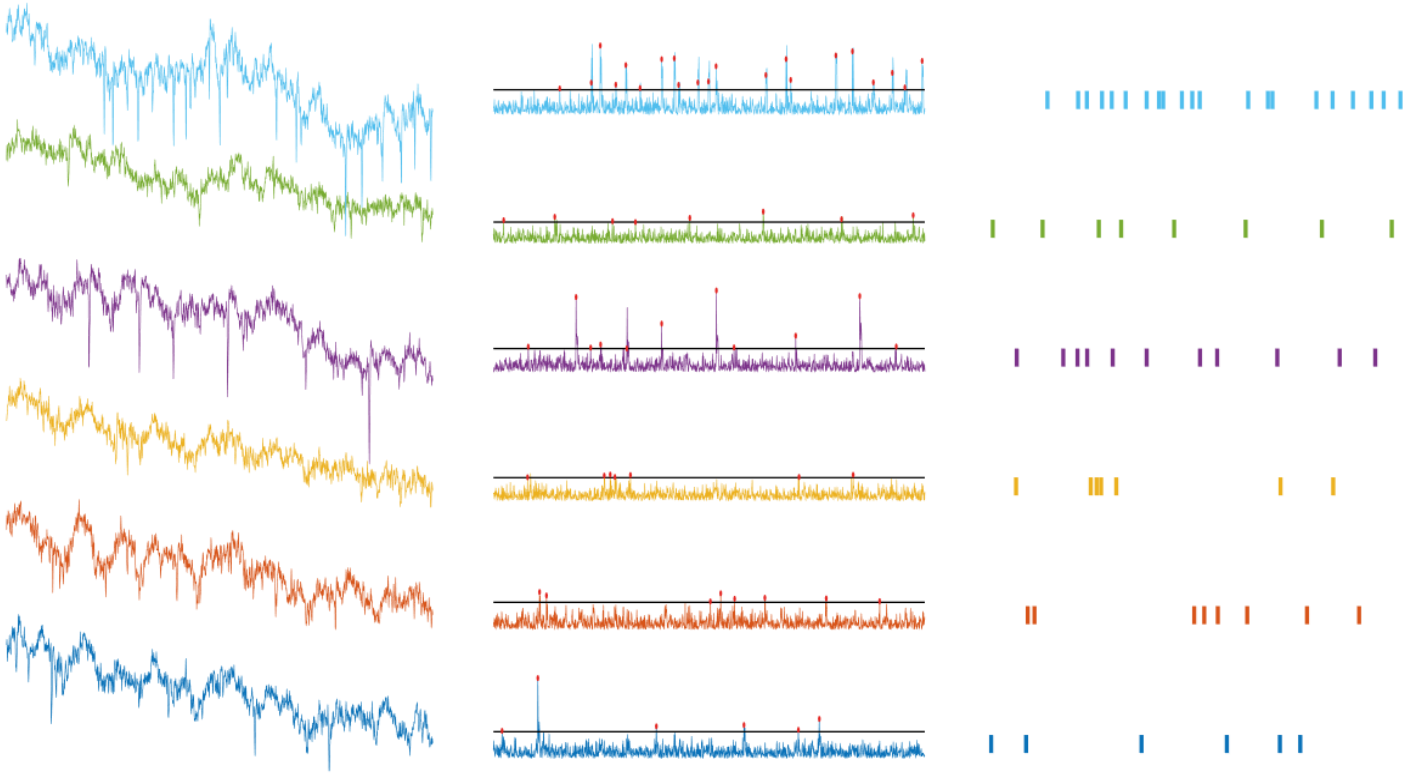
sEMG

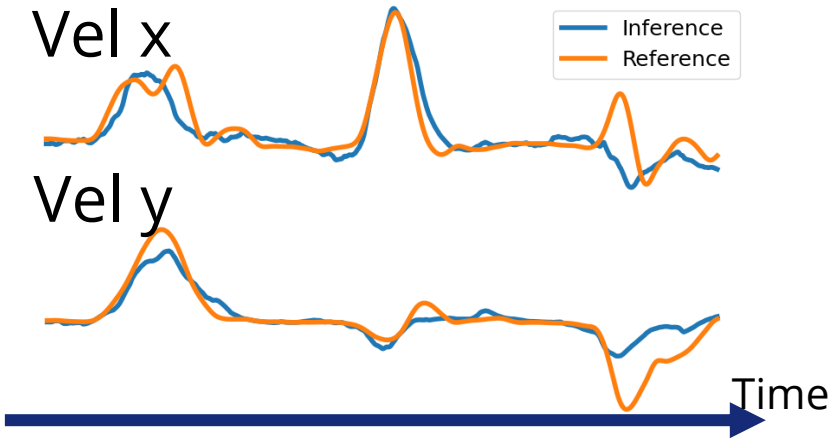
Copyright © 2025

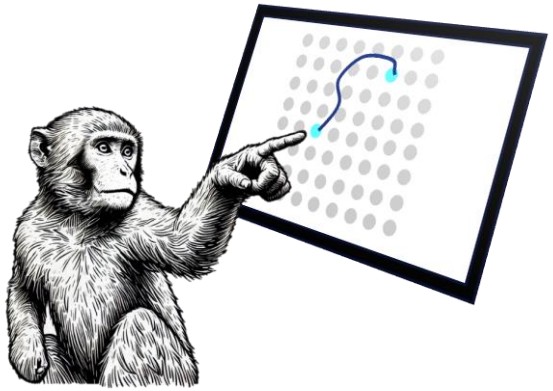
Assessment Use Case 1: iEEG

i-EEG case: finger motion tracking with intracranial EEG decoding

Raw signal → Encoding: denoising – spike emphasis – thresholding - binning → SNN → Regression results







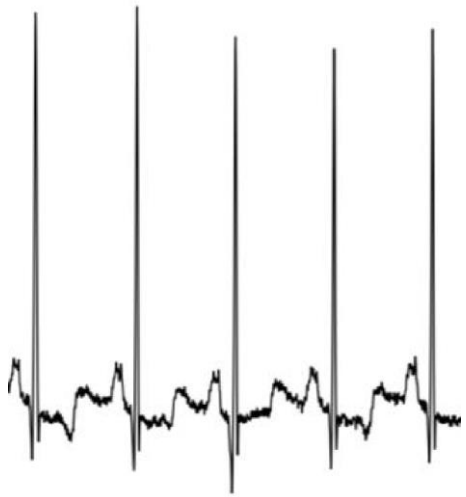
Reference dataset: J. E. O'Doherty, M. M. B. Cardoso, J. G. Makin, and P. N.Sabes, "Nonhuman Primate Reaching with Multichannel SensorimotorCortex Electrophysiology," May 2020

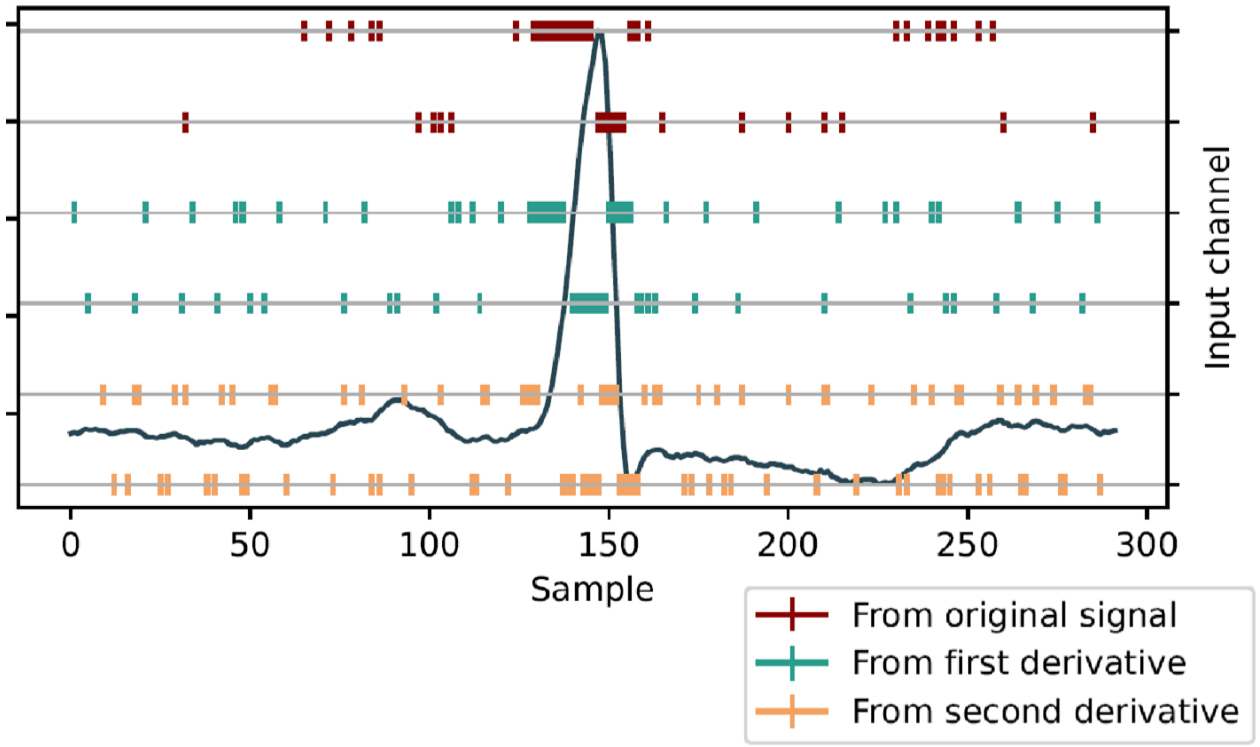
Copyright © 2025

Assessment Use Case 2: ECG

ECG case: classifying arrythmic cardiac beats from single-lead ECG signal

Raw signal → Encoding: denoising – peak detection - delta modulation → **SNN** → Classification results





- Normal beat
- S
- V
- F
- Q

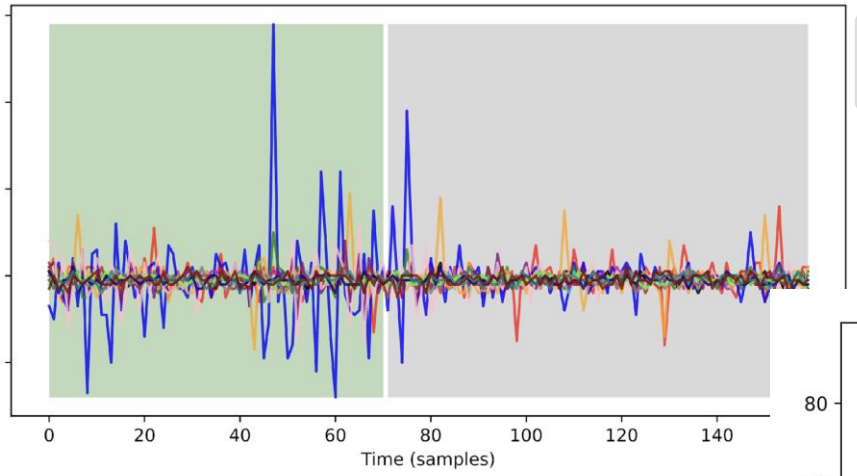
Reference dataset: A. L. Goldbergeret al. "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals,"

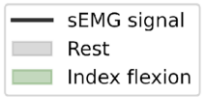
Copyright © 2025

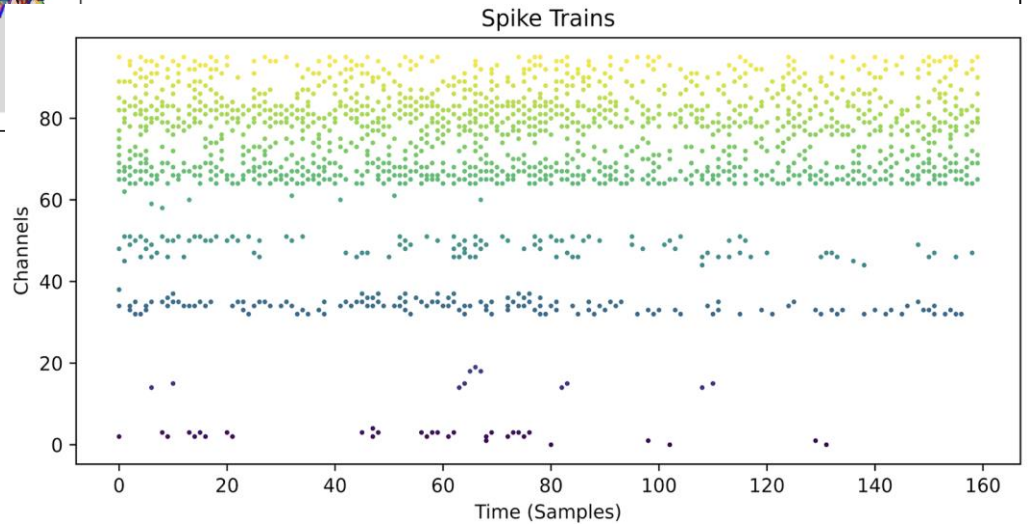
Assessment Use Case 3: sEMG

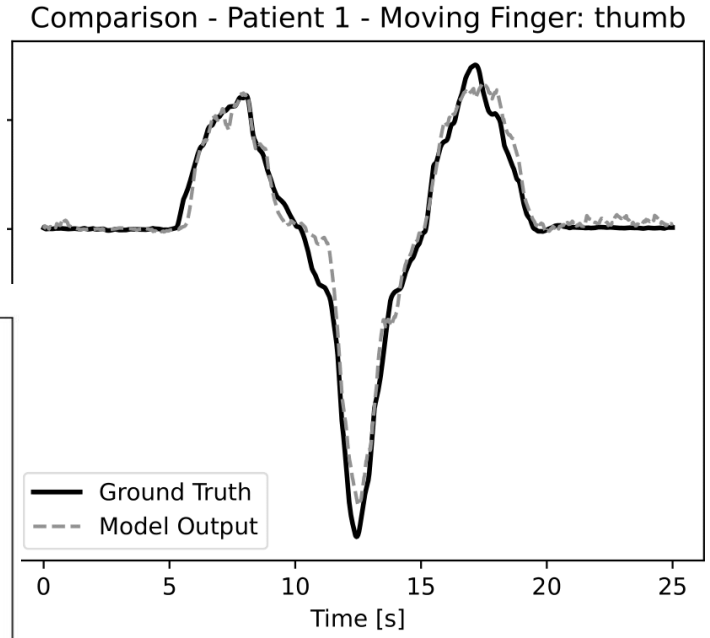
sEMG case: fingers force modelling from HD-sEMG

Raw signal → Encoding: delta modulation → **SNN** → Regression results









X. Jiang et al., "Open Access Dataset, Toolbox and Benchmark Processing Results of High-Density Surface Electromyogram Recordings," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 29, pp. 1035-1046, 2021, doi: 10.1109/TNSRE.2021.3082551.

Copyright © 2025

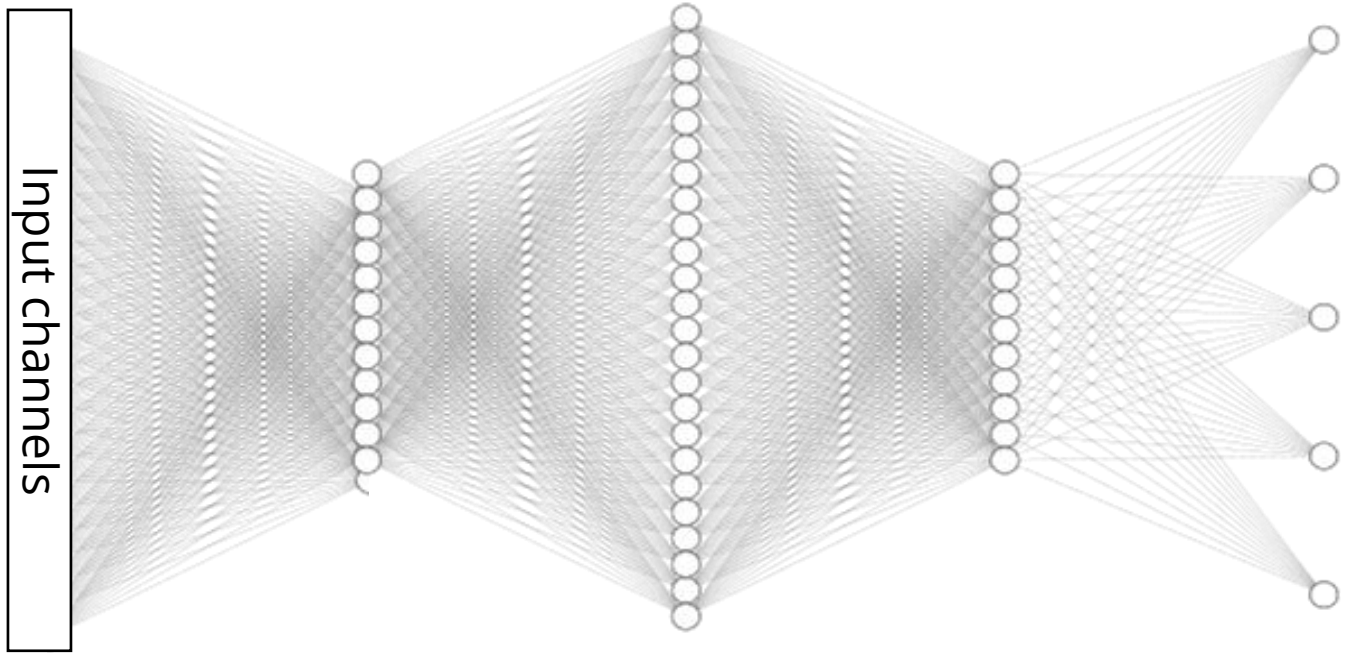
Assessment Use Cases - Flexibility

	Task	Num. channels	Samp. Freq.	Bandwidth	Encoding	Decoding	Output
iEEG	Neural signal decoder	96	12 KHz	18 Mbps	Spike detection	None (output neuron potential)	2 velocity at 1KHz
ECG	ECG arrhythmia classification	1	360 Hz	6 Kbps	Peak detection and heart-beat delta encoding	Spike rate ranking	Five-class classification task
sEMG	sEMG finger force modelling	128	1 kHz	2 Mbps	Continuous raw sEMG delta modulation	None (output neuron potential)	5 fingers force modelling

Copyright © 2025

SNN and training

- Feed-forward SNN
- 4 FC layers
- I/O match use case dataset
- Training library:
  - SnnTorch for regression
  - Slayer for classification



USE CASE	INPUT CHANNELS	L1	L2	L3	L4	LOSS	LEARNING RATE	PATIENCE	TRAINING PLATFORM
iEEG	96	64	128	64	2	MSE	1e-3	50	snnTorch
ECG	8				5	Spike Rate	1e-3, 1e-4, 1e-5	20	Slayer
sEMG	128				5	MSE	3e-4	30	snnTorch

Copyright © 2025

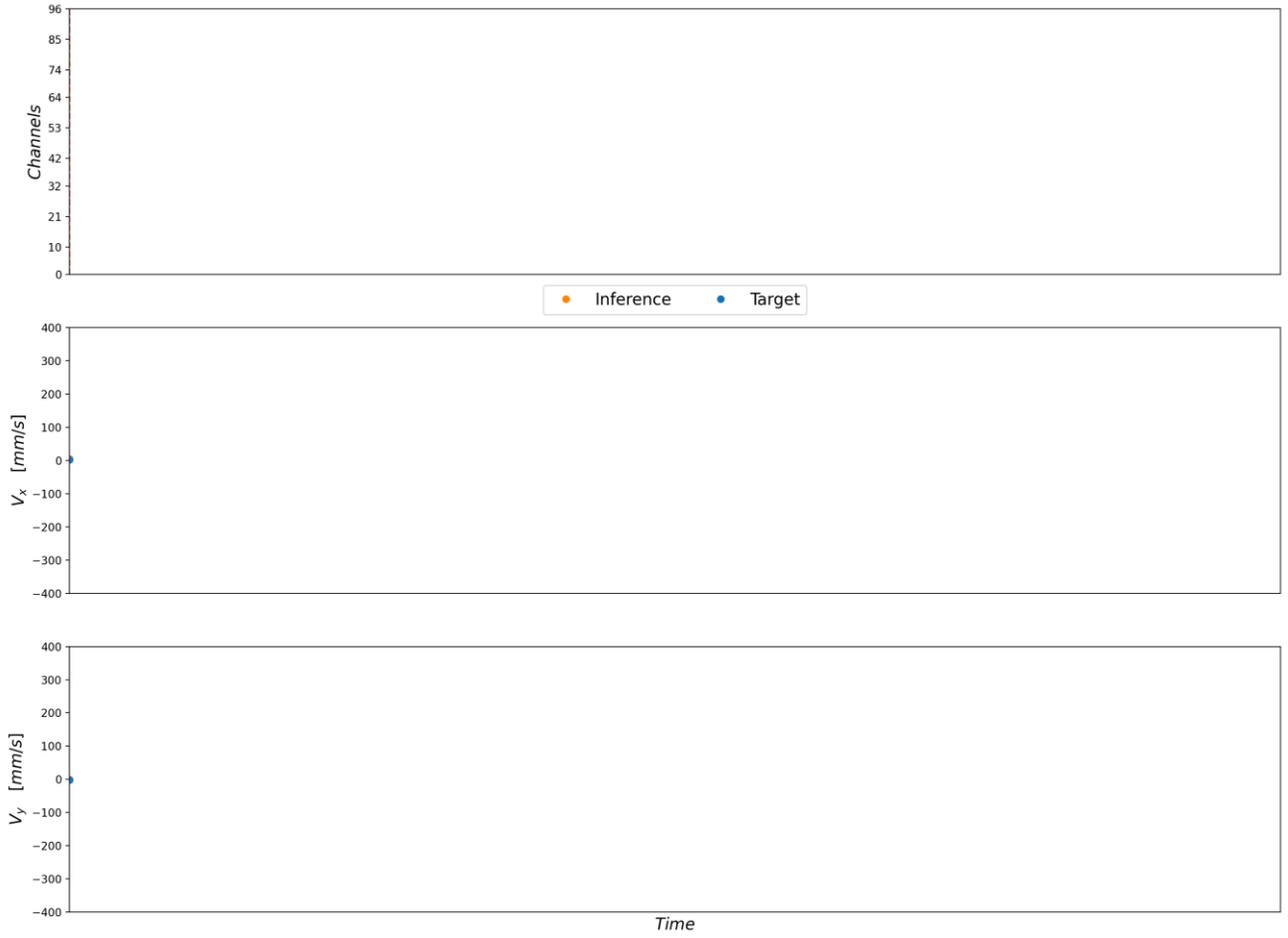
Accuracy

	Reference	Accuracy metric	Ref. accuracy	Ours
	iEEG	[1]	CC	0.77	0.82
	ECG	[2]	Interclass	98.3%	98.4%
	sEMG	[3]	R^2	0.87	0.87

[1] Yik, Jason, et al. "NeuroBench: A Framework for Benchmarking Neuromorphic Computing Algorithms and Systems." arXiv preprint arXiv:2304.04640 (2023).
[2] Y. Xing, et al. "Accurate ecg classification based on spiking neural network and attentional mechanism for real-time implementation on personal portable devices," Electronics, vol. 11, no. 12, 2022.
[3] A. Vitale et al. "Neuromorphic edge computing for biomedical applications: Gesture classification using emg signals". IEEE Sensors Journal 22(20)

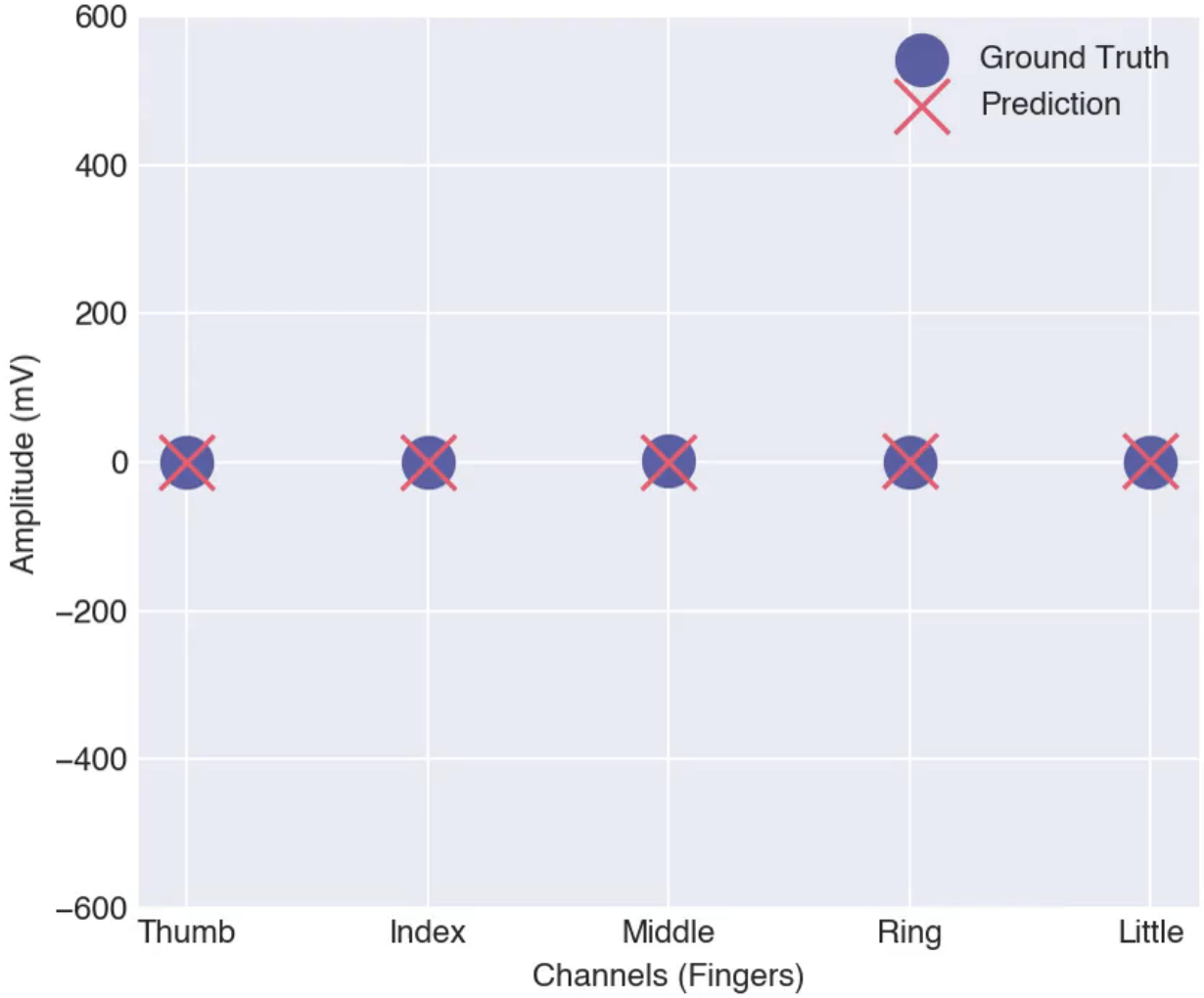
Copyright © 2025

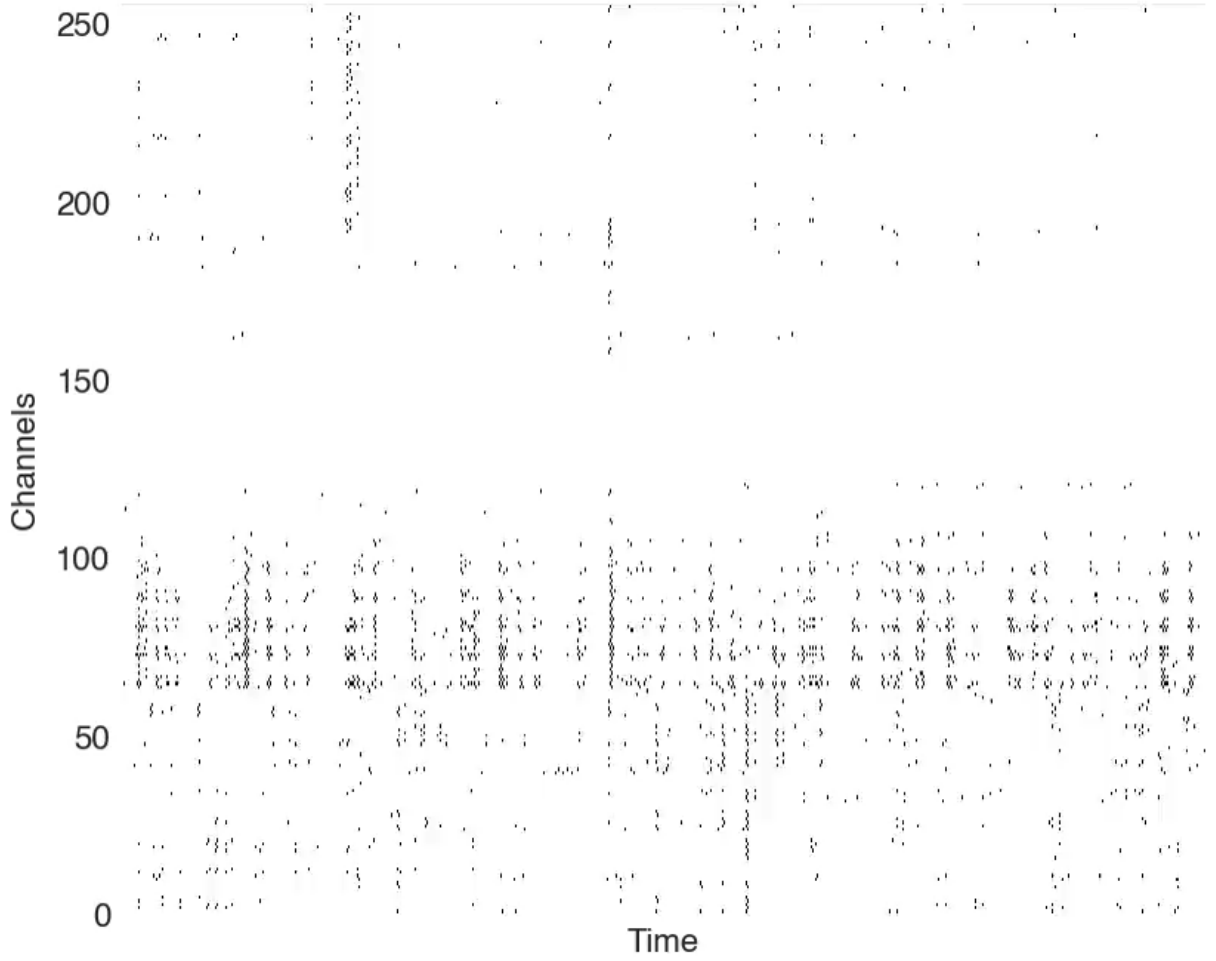
Motion intention tracking with SNN -iEEG



Copyright © 2025

Motion intention tracking with SNN -sEMG

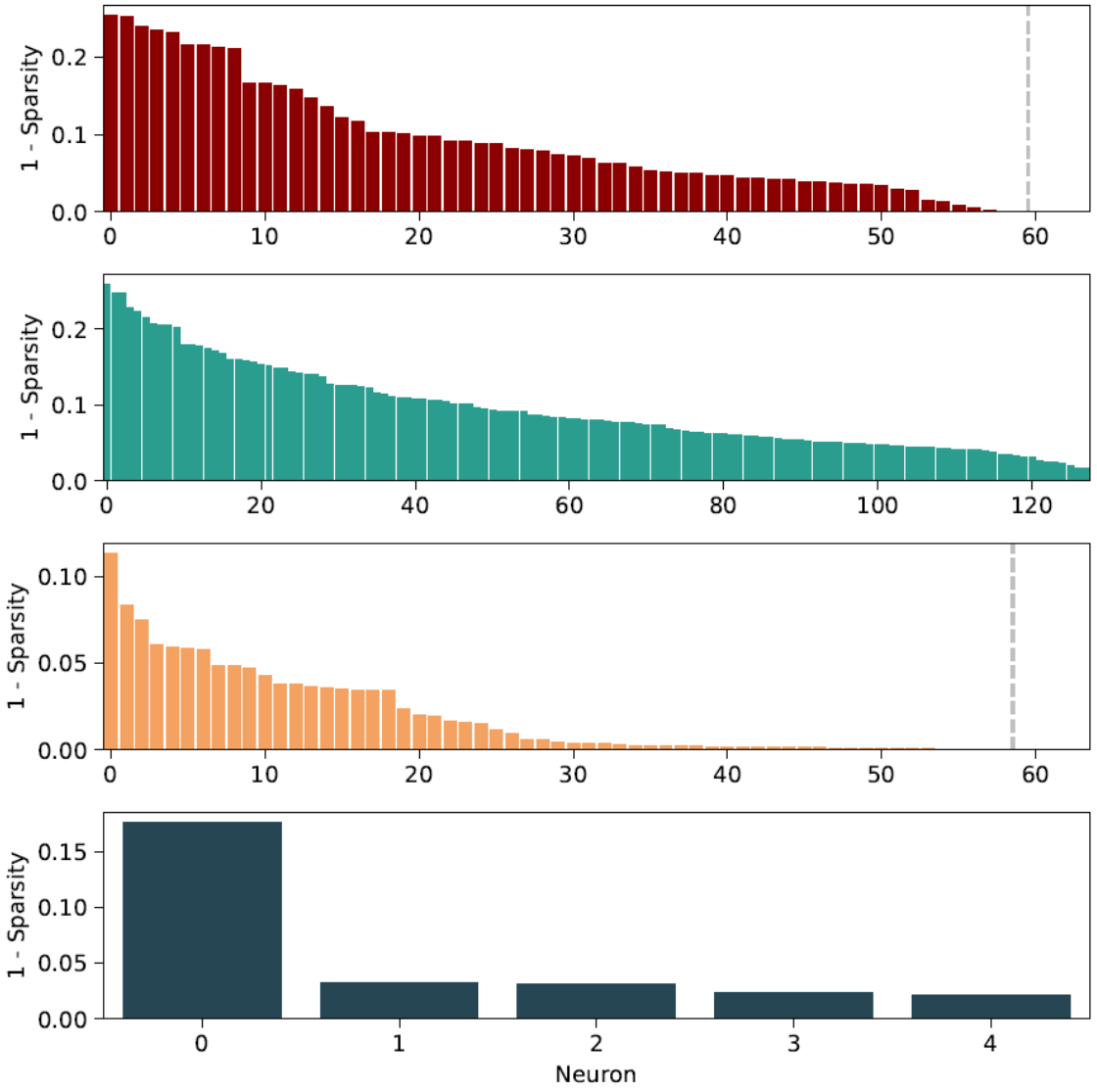




Copyright © 2025

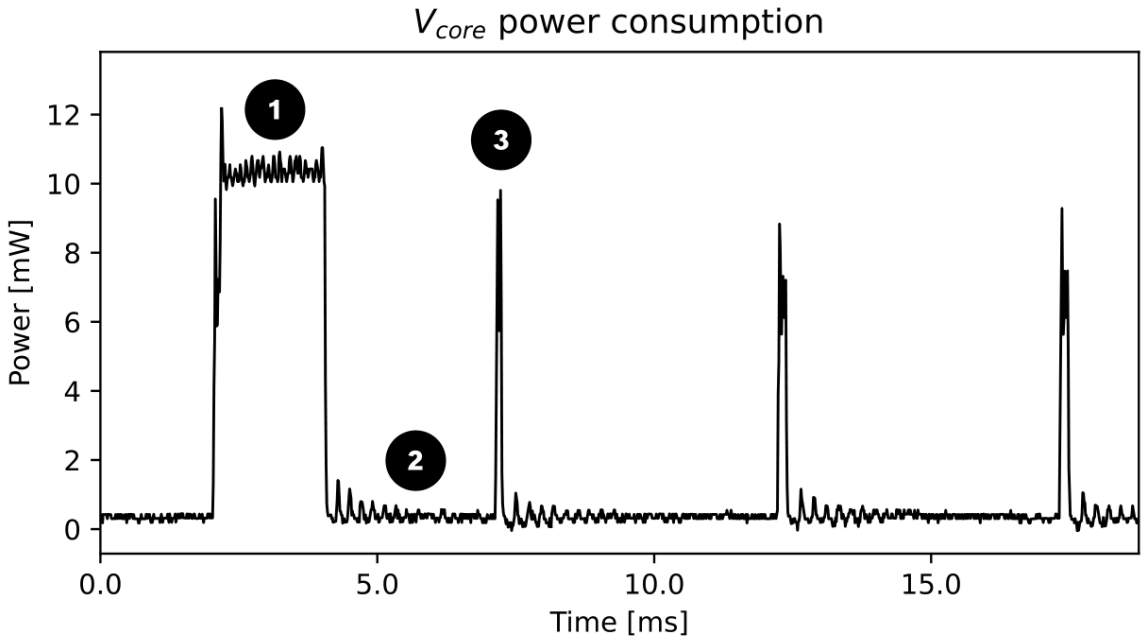
Inference time and sparsity

	I-EEG	ECG	S-EMG
Sparsity	86%	84,7%	92%
Inference time	19 µs	4 ms	2.9 ms
RT requirement	0.1 ms	500 ms (@120bpm)	100 ms



Copyright © 2025

Power Consumption



1. Inference + I/O + encoding
2. Idle
3. I/O encoding

iEEG	ECG	sEMG
9.83 mW	1.45 mW	3.11 mW

Average power consumption per use case

Copyright © 2025

SYNtzulu vs other FPGA-based SNN Accelerators

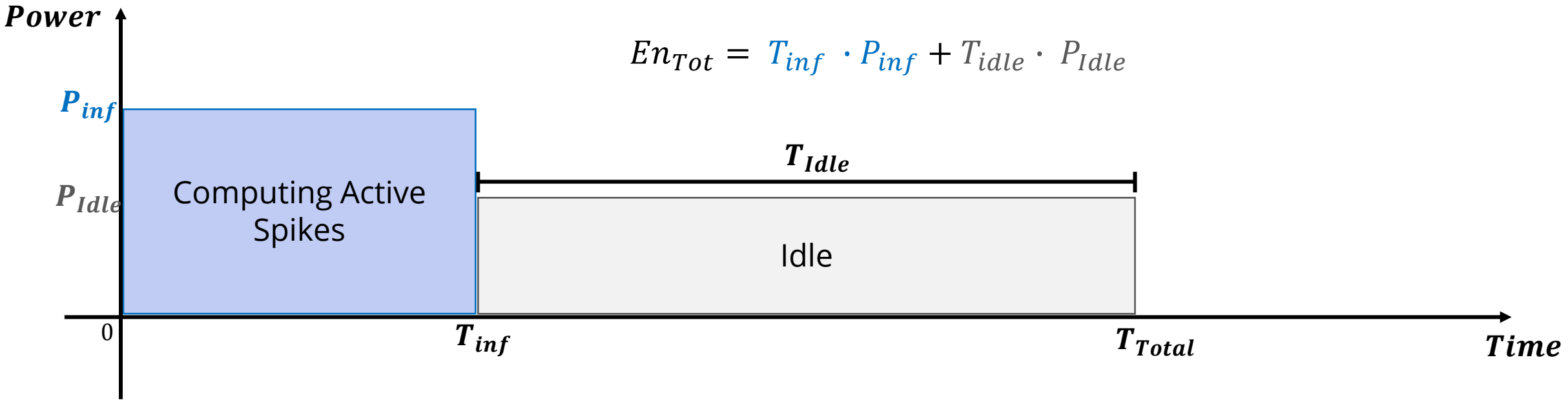
	Optimized for	Frequency	FPGA target	Inference power	Energy/synapse
Li et al.	MNIST	100 MHz	Virtex7	1.6W	28nJ
Liu et al.	MNIST	100 MHz	XA7Z020	280 mW	-
SyncNN	MNIST	200 MHz	Zynq (ZCU102)	400 mW	-
Spiker	MNIST	100 MHz	Artix7	-	41nJ
SYNtzulu	sensor data	22.5 MHz	Lattice iCE405K	12.05 mW	62 pJ

Copyright © 2025

SYNtzulu flying to OPEN-HW ASIC

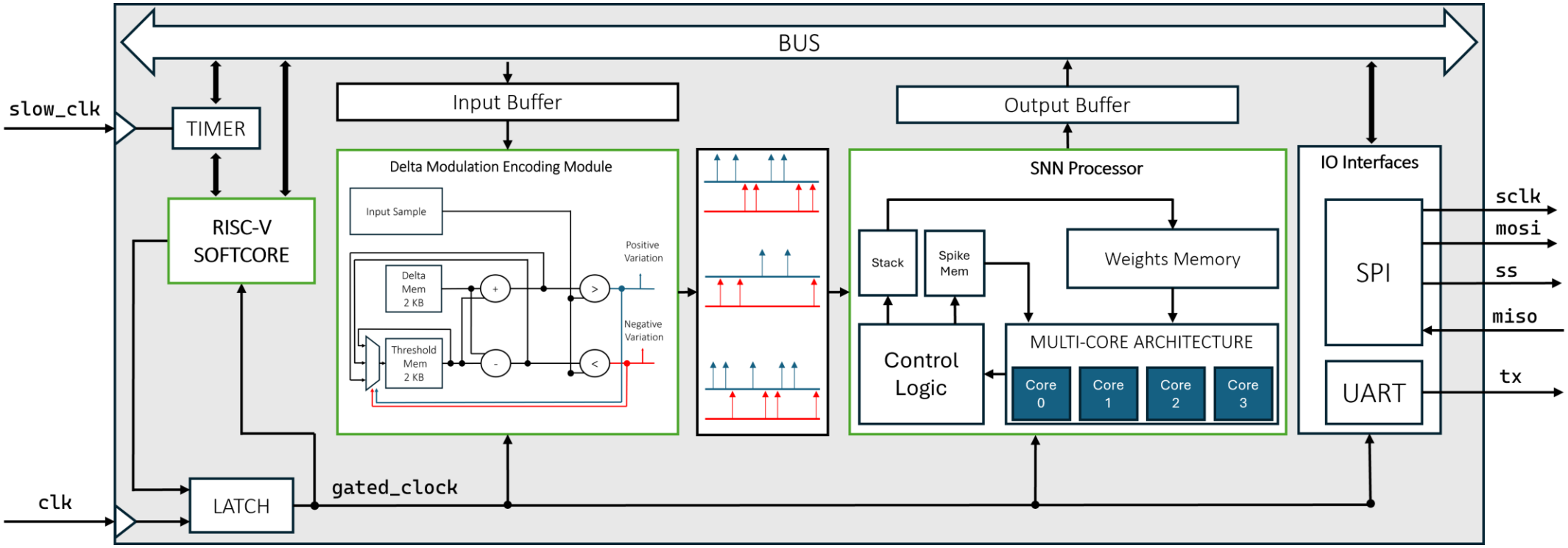
Copyright © 2025

SYNtzulA (SYNtzulu on ASIC)



- Reduce *PIdle* wrt FPGA (High static power consumption, due to routing infrastructure and unused logic)
- Exploit resource allocation to increase performance (Allocate JUST enough resource to increase throughput)
- Tolerate legacy technology nodes

Copyright © 2025



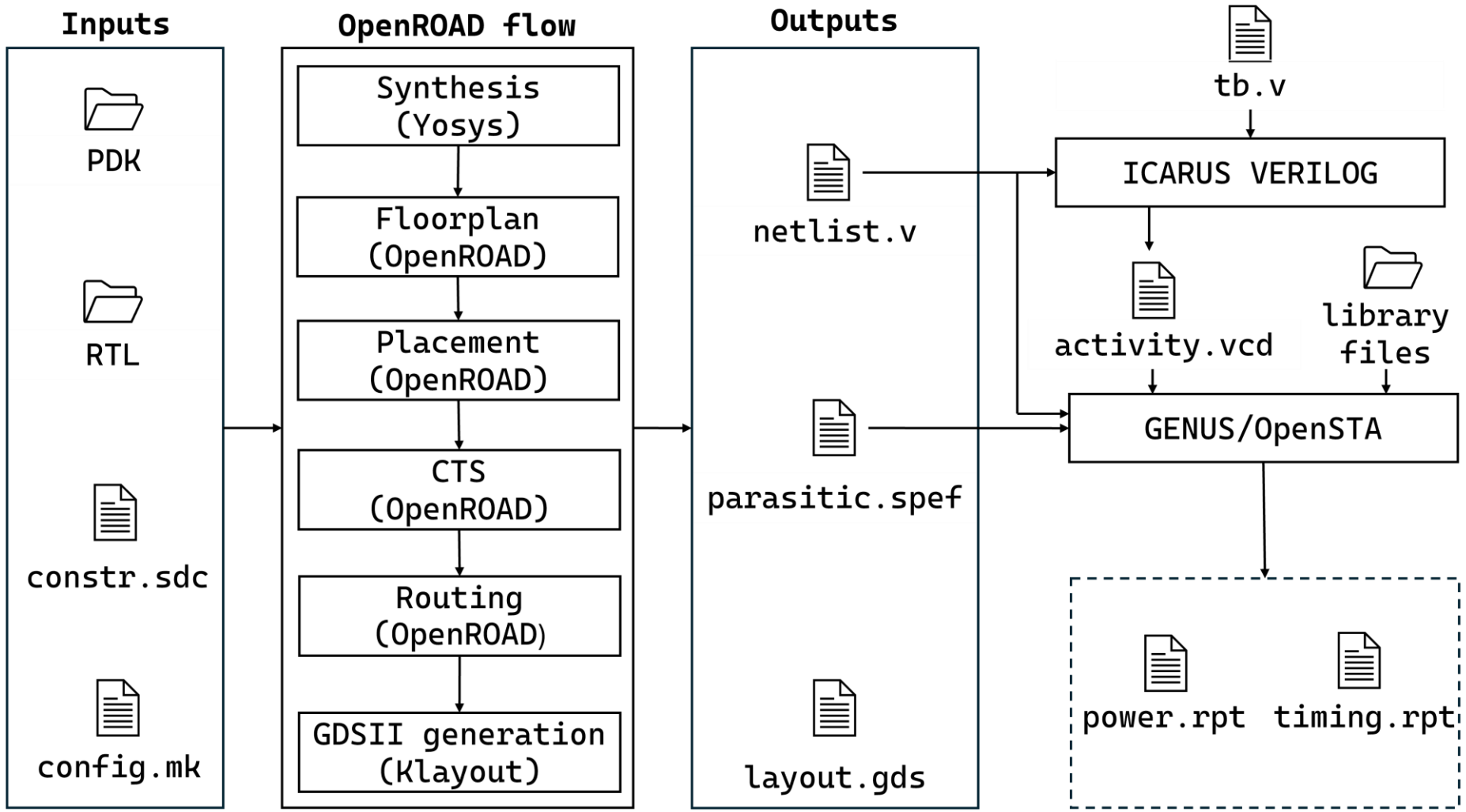
adapt to the new PDK :

- Replace FPGA macros (e.g., BRAM, SPRAM) with flip-flops and SRAM macros available in the PDK.
- encoding scheme is fixed
- quad-core version with doubled throughput and improved energy efficiency.

Copyright © 2025

RTL top GDSII flow - ORFS

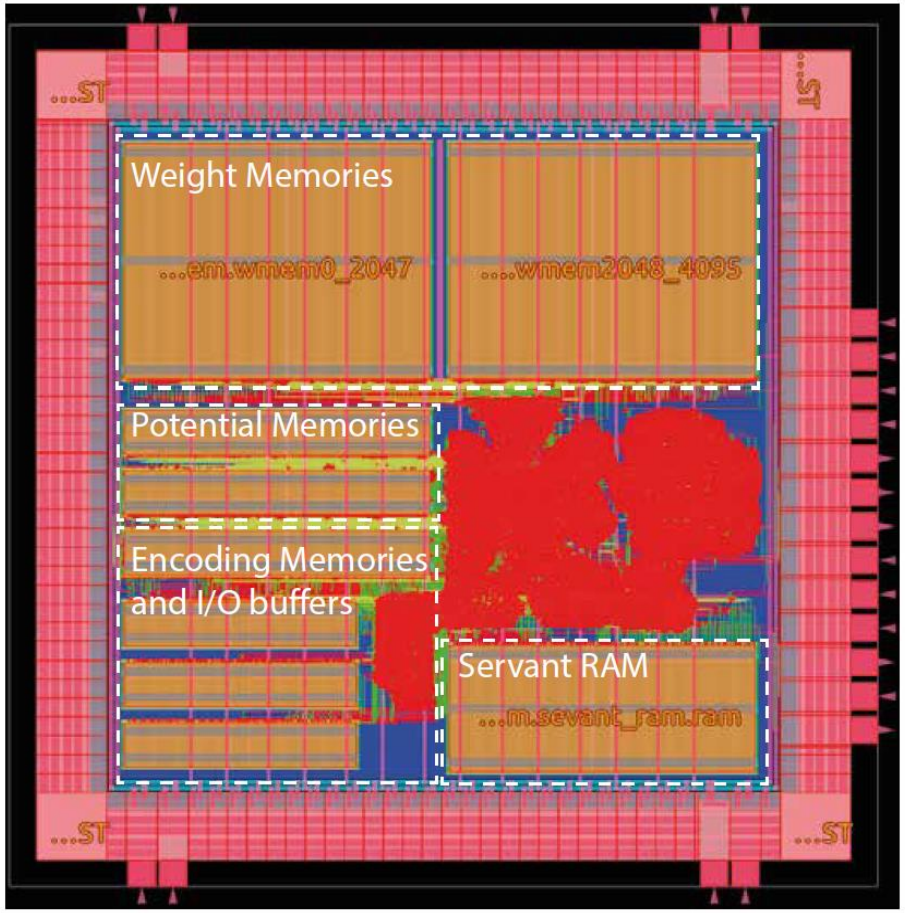
130nm BiCMOS Open Source PDK

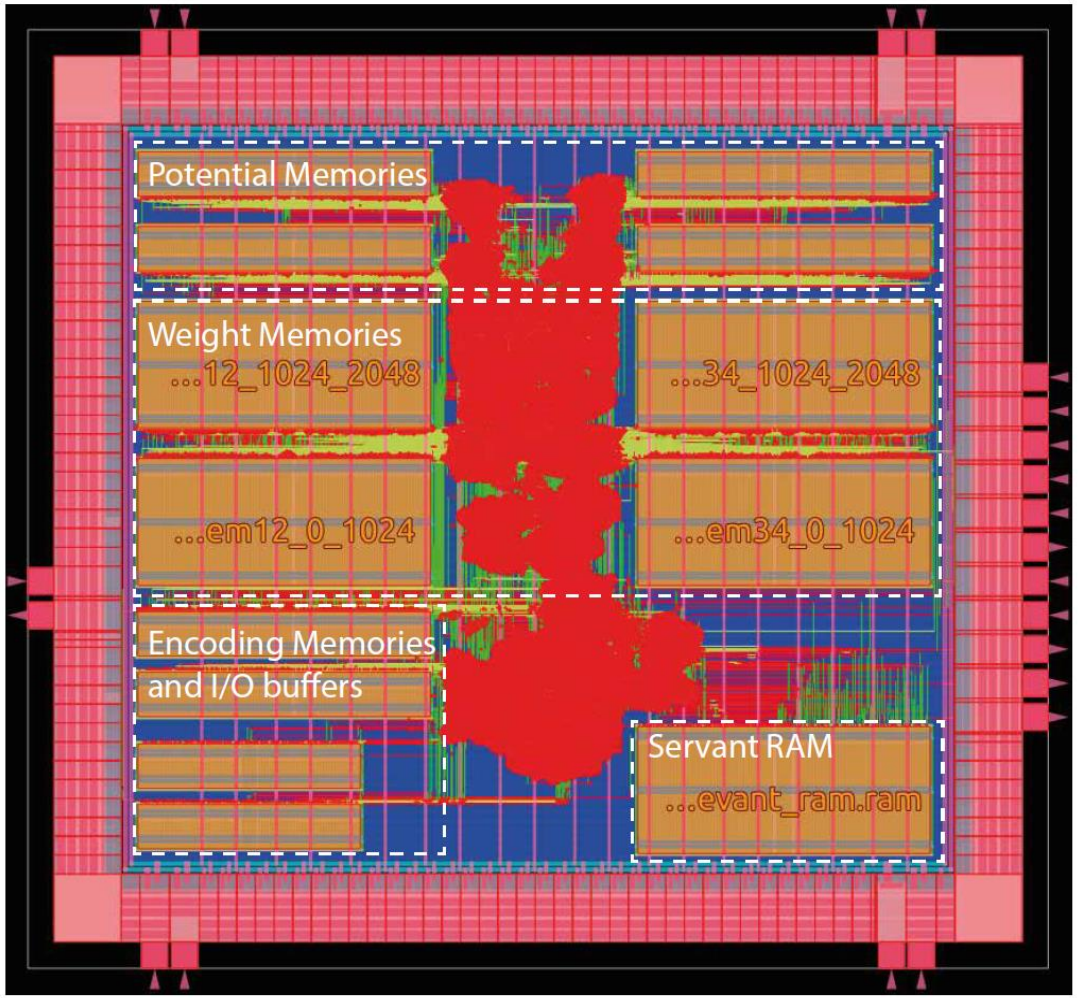


https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts
https://github.com/IHP-GmbH/IHP-Open-PDK

Copyright © 2025

Dual/Quad core Layout

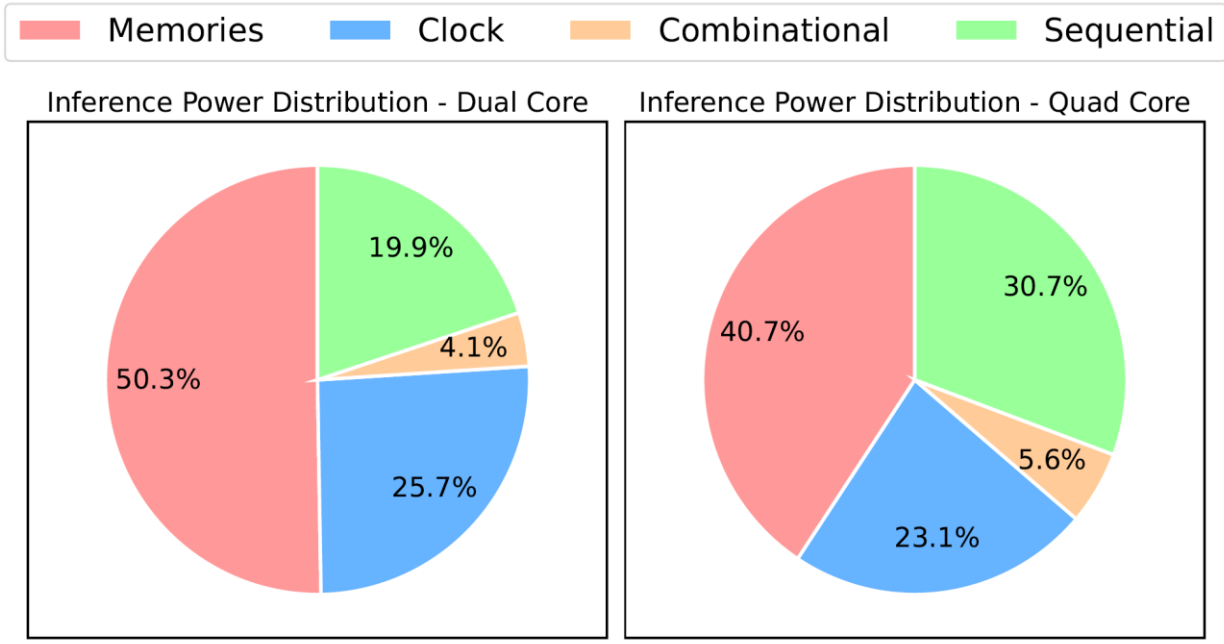




Copyright © 2025

SYNtzula - results

	Dual Core	Quad Core
Die Area	2230 × 2285 µm	2735 × 2485 µm
Core Area	1660 × 1715 µm	2165 × 1915 µm
FMAX	125 MHz	125 MHz
Performance	1 GSOP/s	2 GSOP/s
PInference	0.517 mW/MHz	0.584 mW/MHz
PLeakage	0.046 mW	0.097 mW
Energy/SOP	64.66 pJ	36.50 pJ



Copyright © 2025

SYNtzula vs SYNtzulu

1) sEMG-based Gesture Classification

SNN Topology	Synapses	Sparsity	Accuracy
96x64x128x64x13	23360	77%	82%

2) iEEG-based Neural Decoding

SNN Topology	Synapses	Sparsity	Accuracy
96x64x128x64x5	22848	73%	0.84

3) ECG-based Arrhytmia detection

SNN Topology	Synapses	Sparsity	Accuracy
6x64x128x64x5	17088	85%	98.4%

Use case	Platform	Cores	T_{inf} [µs]	T_{idle} [µs]	En_{inf} [nJ]	En_{Idle} [nJ]	En_{Tot} [nJ]	Improvements
sEMG Gesture Classification	FPGA	2	30	102	368	132	500	
	ASIC	4	15	51	196	6	**202**	**- 60%**
iEEG Neural Decoding	FPGA	2	34	95	418	122	540	
	ASIC	4	17	47	223	5	**228**	**- 58%**
ECG Arrhythmia detection	FPGA	2	14	82	175	107	282	
	ASIC	4	7	41	94	4	**98**	**- 65%**

Copyright © 2025

Conclusion

- Design and optimization of **Spiking Neural Networks** for power efficient and accurate sensor data processing at the micro-edge is feasible.
- Hardware and software instruments are solid and continuously improving.
- Demonstration of **accuracy** comparable with the state of the art, and suitable **efficiency** for low-power execution on constrained hardware platforms.
- Adoption of FPGAs and OpenHW are suitable easy-access development instruments for ubiquitous SNN-based AI

Copyright © 2025

Paola, PhD
Optimization tools and emerging models

Gianluca, PhD
Spiking neural networks and FPGA design

Matteo, PhD
Multi-agent DL and on-device learning

Luca, PhD student
EDA tools, IC design and tapeout

Cecilia, management support

Paolo, interrupts, noise, micro-management and other bad practices

6 scolarships
X many bachelor and master thesis

Copyright © 2025

Publications

- Leone G., Antonio Scrugli M., Badas L., Martis L., Raffo L., Meloni P. - SYNtzulu: A Tiny RISC-V-Controlled SNN Processor for Real-Time Sensor Data Analysis on Low-Power FPGAs(2025)
- Leone G., Martis L., Raffo L., Meloni P. - Enabling SNN-Based Near-MEA Neural Decoding with Channel Selection: An Open-HW Approach(2025)
- Scrugli M.A., Leone G., Busia P., Raffo L., Meloni P. - Real-Time sEMG Processing With Spiking Neural Networks on a Low-Power 5K-LUT FPGA(2025)
- Leone G., Raffo L., Meloni P. - On-FPGA Spiking Neural Networks for End-to-End Neural Decoding(2023)Leone G., Martis L., Raffo L., Meloni P.Spiking Neural Networks for Integrated Reach-to-Grasp Decoding on FPGAs(2023)
- Scrugli M.A., Busia P., Leone G., Meloni P. - On-FPGA Spiking Neural Networks for Integrated Near-Sensor ECG Analysis(2024)
- Martis L., Leone G., Raffo L., Meloni P. - Spiking Neural Networks for Near-Sensor Processing: An Open-Hardware Experience(2024)
- Busia P., Leone G., Matticola A., Raffo L., Meloni P. - Wearable Epilepsy Seizure Detection on FPGA with Spiking Neural Networks(2025)
- Meloni P., Busia P., Leone G., Martis L., Scrugli M.A. - Exploiting FPGAs and Spiking Neural Networks at the Micro-Edge: The EdgeAI Approach(2024)
- Leone G., Martis L., Raffo L., Meloni P. - On-FPGA Spiking Neural Networks for Multi-variable End-to-End Neural Decoding(2023)
- Scrugli M.A., Leone G., Busia P., Meloni P. - sEMG-Based Gesture Recognition with Spiking Neural Networks on Low-Power FPGA(2024)
- Martis L., Leone G., Raffo L., Meloni P. - Low-Power FPGA-Based Spiking Neural Networks for Real-Time Decoding of Intracortical Neural Activity(2024)

Copyright © 2025

Thank You

@ Name.Name@name.eu

EAI

https://edge-ai-tech.eu/

The two-day innovative educational initiative is designed for students, young researchers, and professionals eager to expand their understanding and skills in these cutting-edge technologies.

Participants will engage in interactive discussions, hands-on sessions, and lectures covering foundational concepts and innovative applications of edge AI across various domains with leading experts, gain insights into real-world applications, and work on team-based projects to foster collaboration and innovation.

Copyright © 2025

EDGE AI Academy 2025

Summer School Objectives

- The EDGE AI Academy Summer School aims to empower participants with advanced knowledge, hands-on experience and an in-depth understanding of edge AI, which combines Artificial Intelligence (AI) with edge computing and the Internet of Things (IoT) to process data closer to its source.
- Edge AI technology is gaining prominence due to its ability to enhance real-time decision-making, reduce latency, improve data privacy, and diminish reliance on continuous internet connectivity.
- The topics will extend to RISC-V architectures as ultra-low-power high-performance RISC-V-based processors and ultra-low-power vector-based accelerators are advancing smart IoT and edge AI computing capabilities.

Copyright © 2025