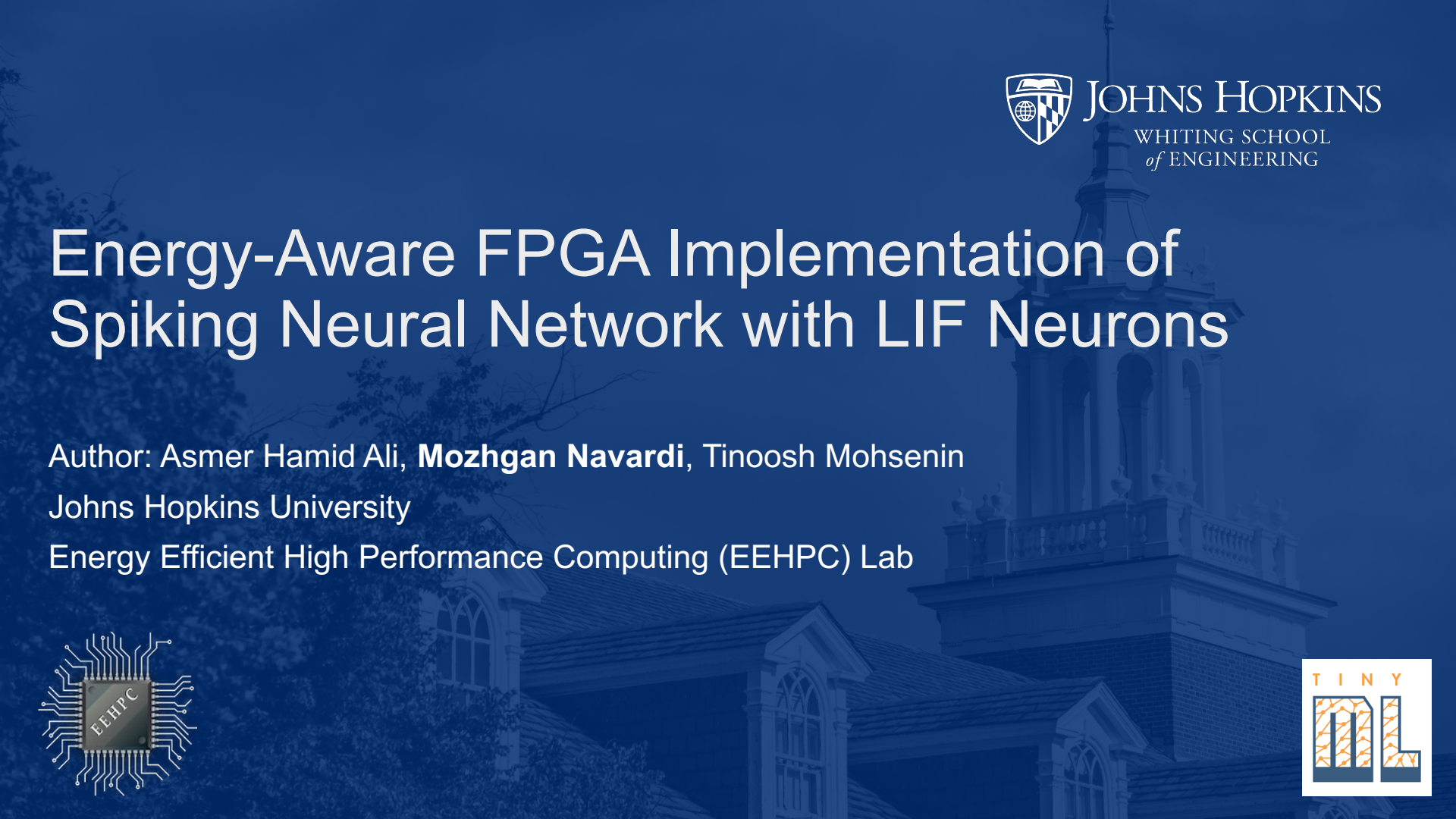
tinyML® Research Symposium

Enabling Ultra-low Power Machine Learning at the Edge

April 22, 2024

www.tinyML.org



Energy-Aware FPGA Implementation of Spiking Neural Network with LIF Neurons

Author: Asmer Hamid Ali, **Mozhgan Navardi**, Tinoosh Mohsenin

Johns Hopkins University

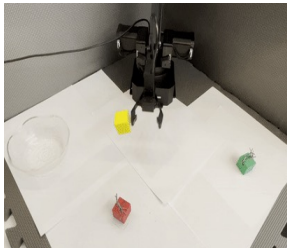
Energy Efficient High Performance Computing (EEHPC) Lab

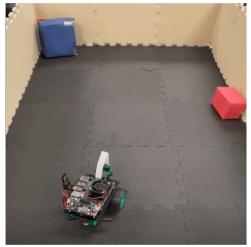
Multi Agent TinyML Systems

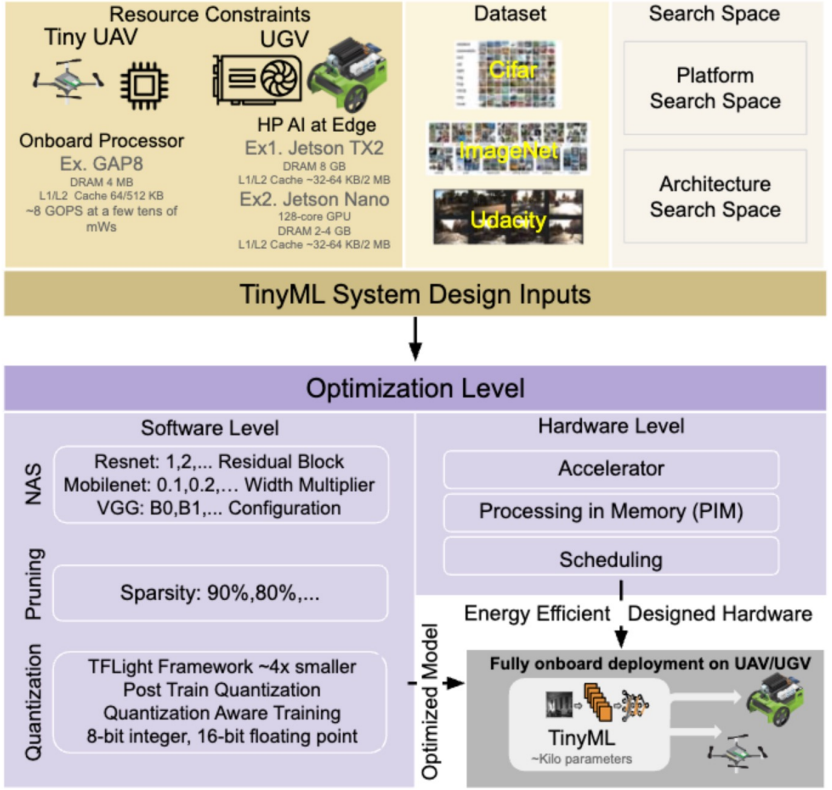
DARPA ArtIAMAS ARL DEVCOM ARMY RESEARCH LABORATORY

Introduction to TinyML

- Various techniques for accelerating Artificial Neural Networks (ANN) inference. These techniques include both:
  - Hardware-level optimization
    - Accelerator
    - Processing in Memory (PIM)
    - Scheduling
  - Software-level optimization
    - Neural Architecture Search (NAS)
    - Pruning
    - Quantization



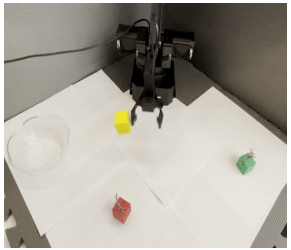


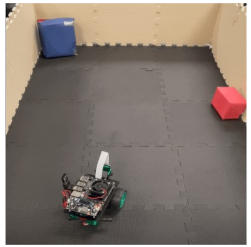


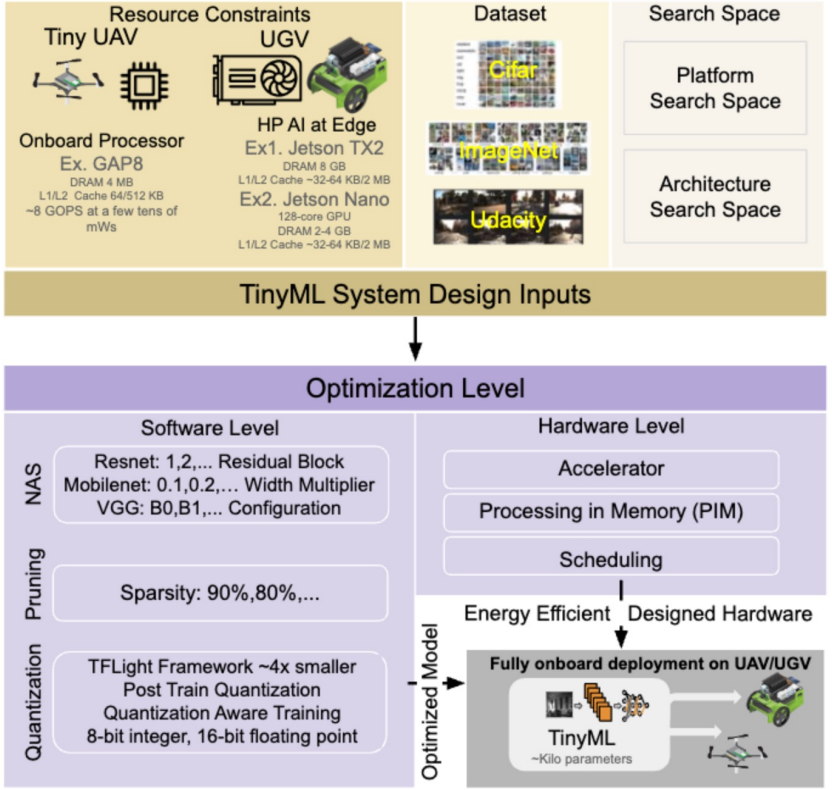
[Mohammad Shafique et al, "TinyML: Current Progress, Research Challenges, and Future Roadmap ", 2021 58th ACM/IEEE Design Automation Conference (DAC), 2021]

Introduction to TinyML

- Various techniques for accelerating Artificial Neural Networks (ANN) inference. These techniques include both:
  - Hardware-level optimization
    - **Accelerator**
    - Processing in Memory (PIM)
    - Scheduling
  - Software-level optimization
    - Neural Architecture Search (NAS)
    - Pruning
    - Quantization



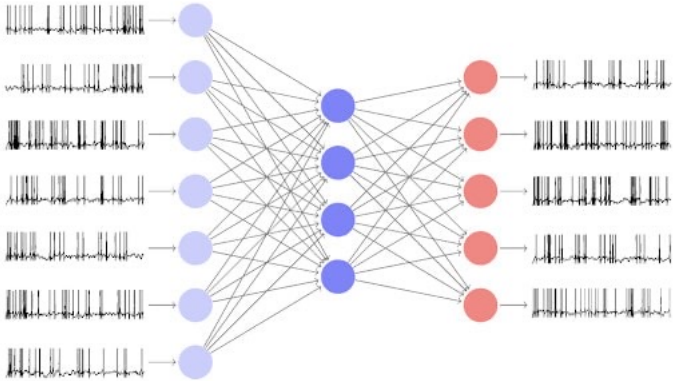




[Mohammad Shafique et al, "TinyML: Current Progress, Research Challenges, and Future Roadmap ", 2021 58th ACM/IEEE Design Automation Conference (DAC), 2021]

Introduction to Spiking Neural Networks (SNN)

- SNN tries to closely mimic the working of a human brain. This is why instead of working with continuously changing in time values used in ANN, SNN operates with discrete events which occur at certain points of time.
- SNN receives a series of spikes as input and produces a series of spikes as the output (a series of spikes is usually referred to as spike trains).
- Threshold models generate an impulse at a certain threshold:
  - Perfect Integrate-and-fire
  - Leaky Integrate-and-fire (LIF)
  - Adaptive Integrate-and-fire



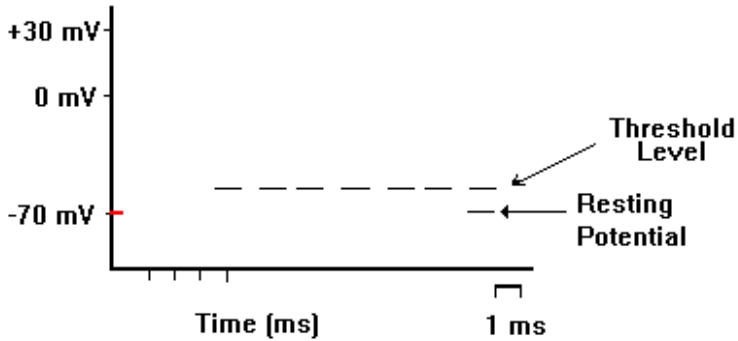
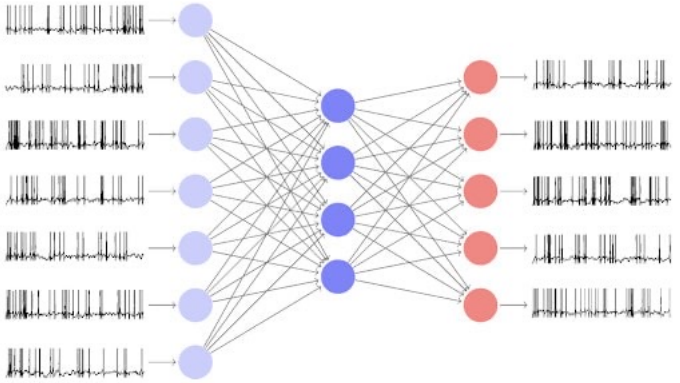


image reference: https://cnvrg.io/spiking-neural-networks/

Introduction to Spiking Neural Networks (SNN)

- SNN tries to closely mimic the working of a human brain. This is why instead of working with continuously changing in time values used in ANN, SNN operates with discrete events which occur at certain points of time.
- SNN receives a series of spikes as input and produces a series of spikes as the output (a series of spikes is usually referred to as spike trains).
- Threshold models generate an impulse at a certain threshold:
  - Perfect Integrate-and-fire
  - **Leaky Integrate-and-fire (LIF)**
  - Adaptive Integrate-and-fire



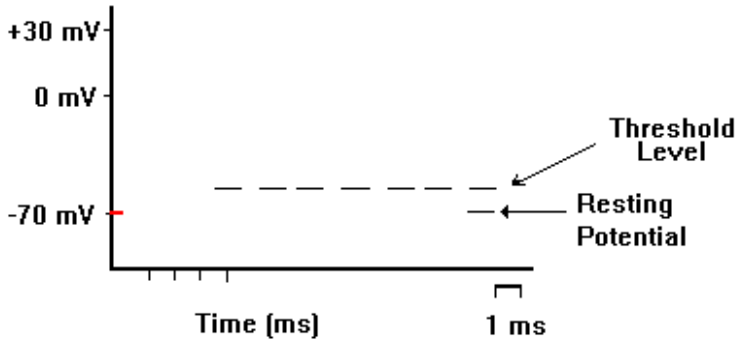


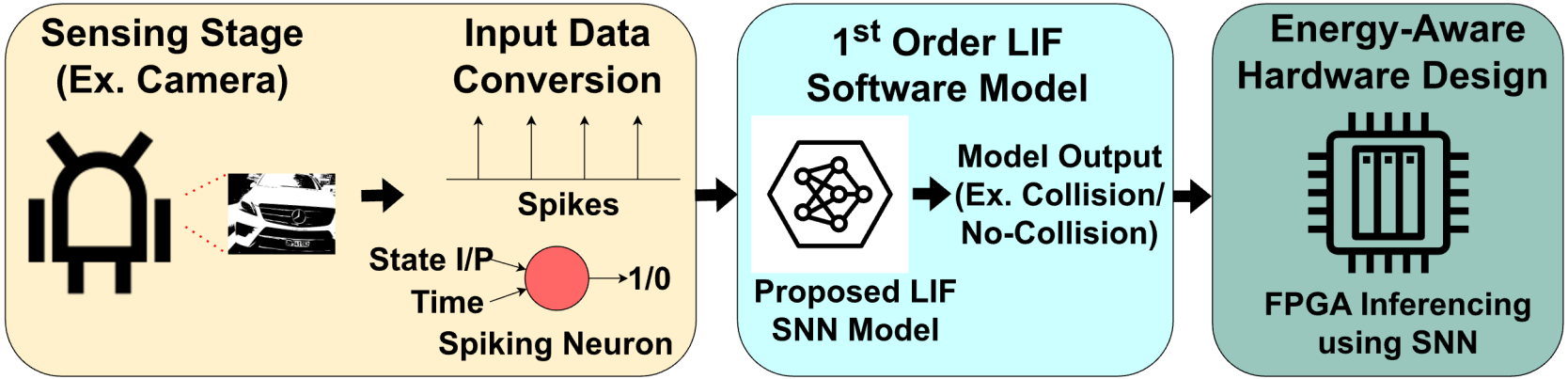
image reference: https://cnvrg.io/spiking-neural-networks/

Problem Definition and Research Objective

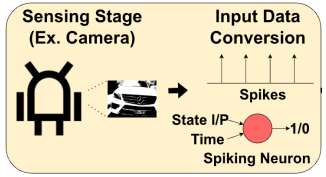
- **Problem Definition**
  - Edge devices are increasingly being deployed for on-device AI applications, which demand low-latency and privacy-preserving computations.
  - Traditional ML algorithms are not well-suited for implementation on low-power, low-resource devices due to their high computational and energy requirements.
- **How to solve it?**
  - By developing an SNN architecture that mimics the biological processes of neurons, we can significantly reduce power consumption since these networks are event-driven and only process signals when necessary.
- **Research Objective**
  - To introduce a novel SNN architecture optimized for energy efficiency and low-power edge devices.
  - Implementation of the 1st Order Leaky Integrate-and-Fire (LIF) neuron model for vision-based ML algorithms on TinyML systems.

Proposed Approach

- The image depicts a three-stage energy-aware framework for TinyML systems:
  - A camera captures the scene (Sensing Stage), and the data is translated into spikes by a neuron model (Input Data Conversion)
  - SNN software model predicts collision events (1st Order LIF Software Model)
  - SNN model is implemented on an FPGA (Energy-Aware Hardware Design)



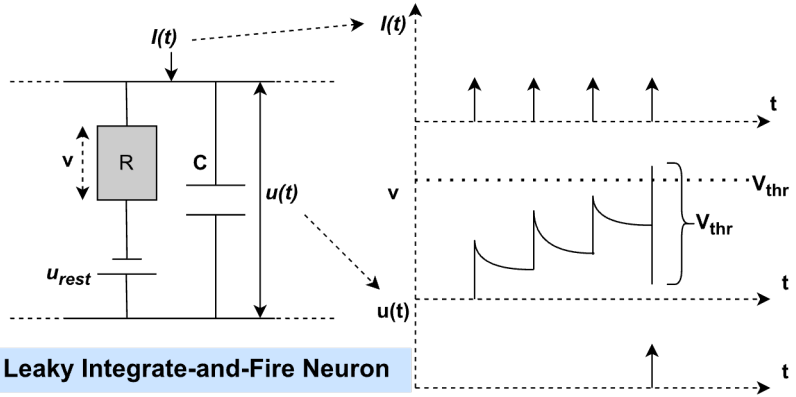
Sensing Stage and Input Data Conversion



- In SNN coding technique is essential to convert static input data, such as images, into a dynamic, time-varying format that SNNs can process effectively.
- LIF model
  - Neuron is represented by a resistor (R) and a capacitor (C) in parallel, analogous to the cell membrane's leak resistance and capacitance, respectively.

- The dynamics of the membrane potential are governed by the differential equation, which describes how $u\ (t)$ changes over time-based on the decay towards resting potential and the input current, with the membrane time constant $\tau_m = RC$ characterizing the rate of decay.

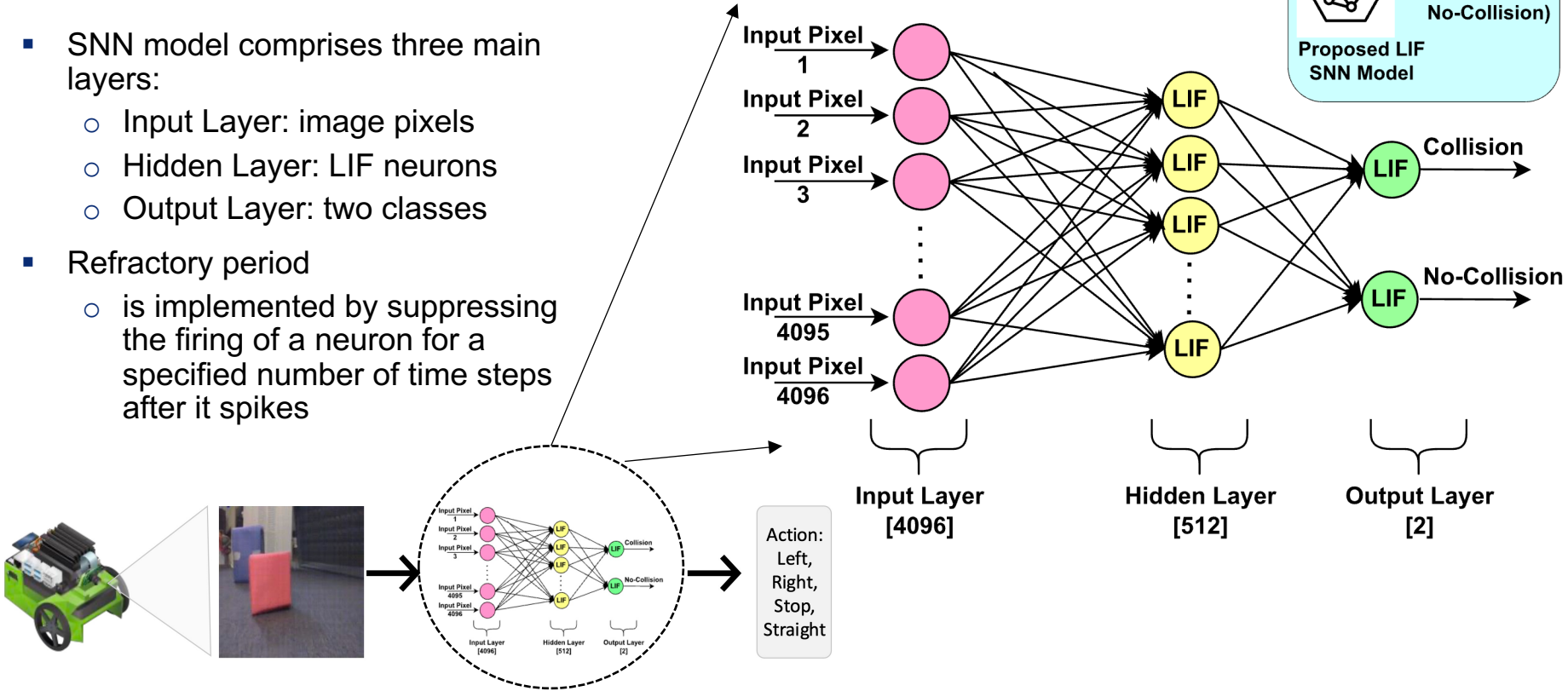
$$\tau_m \frac{d_u}{d_t} = -[\mu(t) - \mu_{rest}] + RI$$



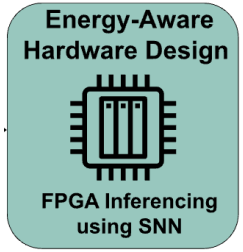
Leaky Integrate-and-Fire Neuron

Proposed 1st Order LIF Software Model

- SNN model comprises three main layers:
  - Input Layer: image pixels
  - Hidden Layer: LIF neurons
  - Output Layer: two classes
- Refractory period
  - is implemented by suppressing the firing of a neuron for a specified number of time steps after it spikes

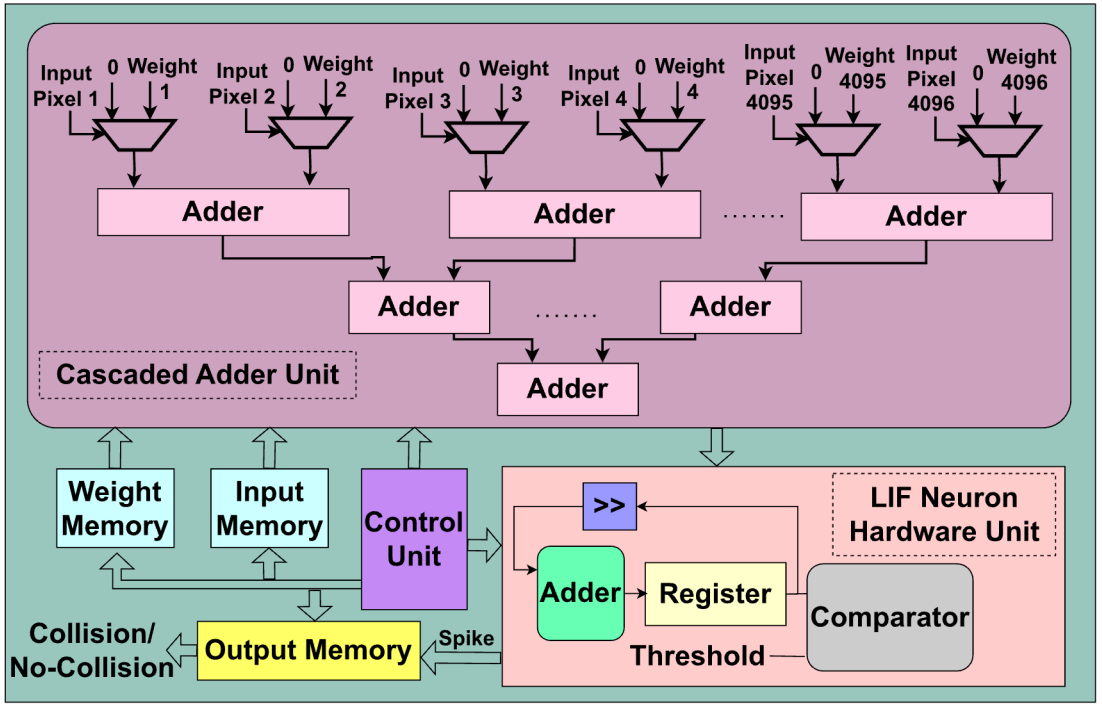


Proposed Energy-Aware Hardware Design

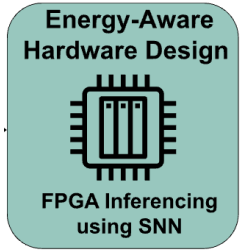


- Cascaded Adder: given that the input pixel values are binary (0 or 1), the standard matrix multiplication operation common in neural networks is simplified.
- A cascaded adder replaces the need for multipliers, which are traditionally more resource-intensive on silicon.
- LIF Neuron Hardware: its primary functionality revolves around processing input signals, accumulating them over time, and generating an output spike under certain conditions, akin to a neuron firing in response to stimuli

$$U[t+1] = \beta U[t] + I[t+1] - U_{rest}$$

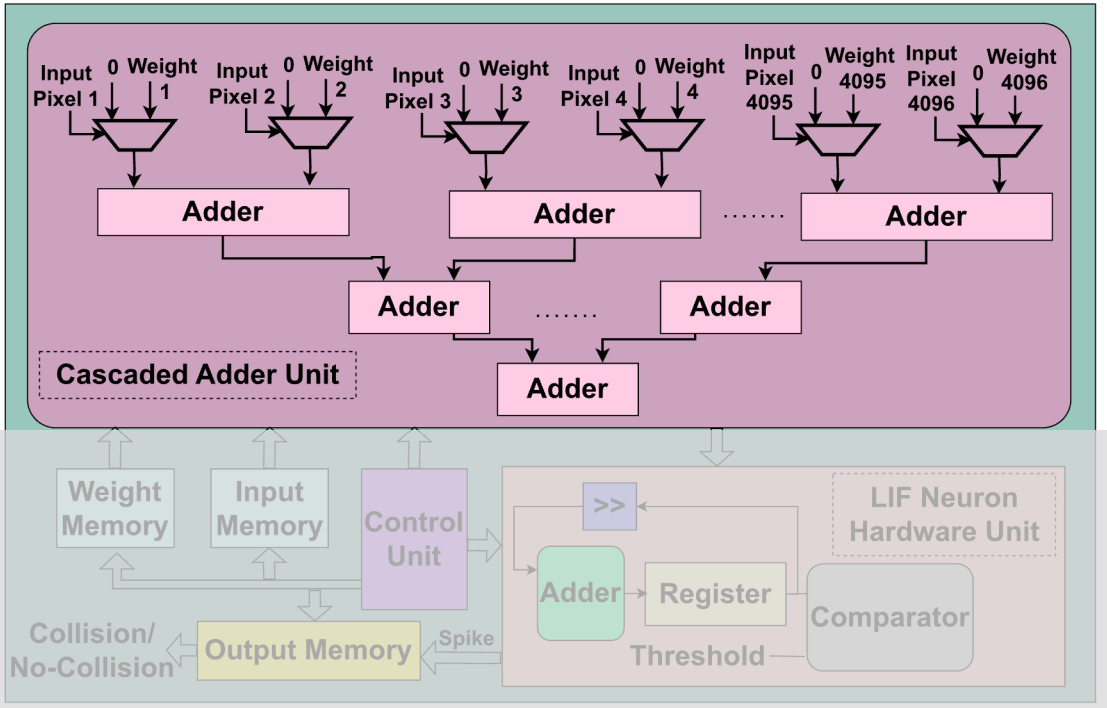


Proposed Energy-Aware Hardware Design

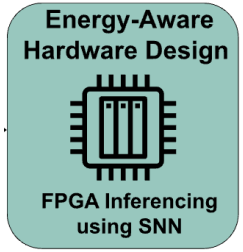


- Cascaded Adder: given that the input pixel values are binary (0 or 1), the standard matrix multiplication operation common in neural networks is simplified.
- A cascaded adder replaces the need for multipliers, which are traditionally more resource-intensive on silicon.
- LIF Neuron Hardware: its primary functionality revolves around processing input signals, accumulating them over time, and generating an output spike under certain conditions, akin to a neuron firing in response to stimuli

$$U[t+1] = \beta U[t] + I[t+1] - U_{rest}$$

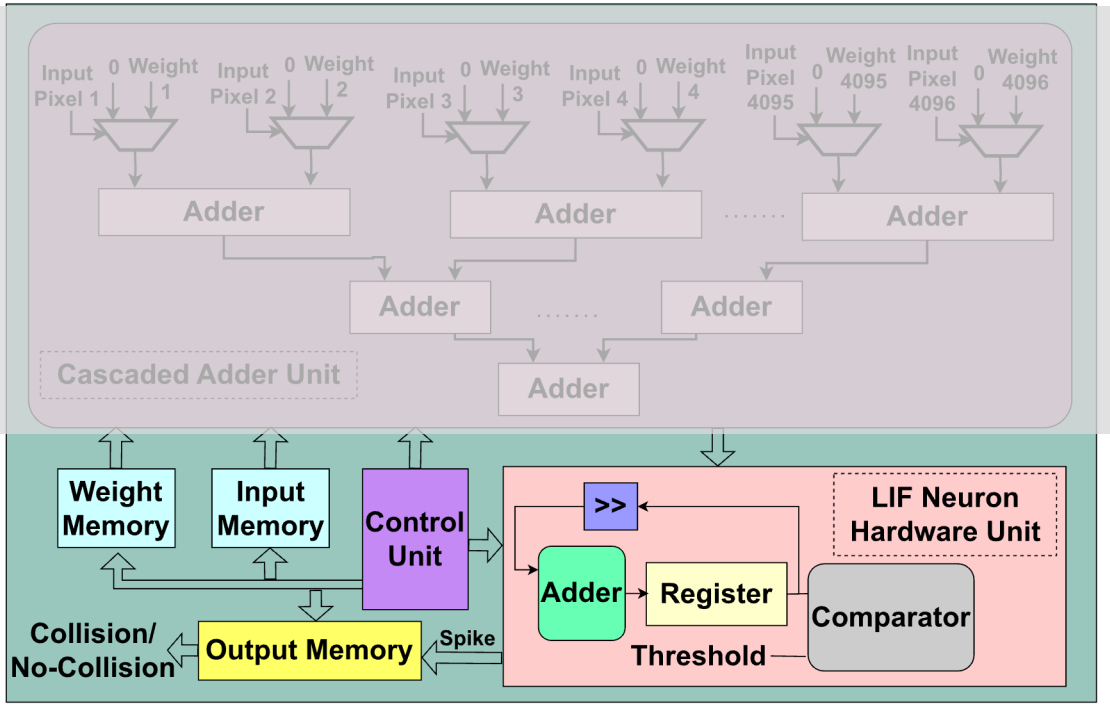


Proposed Energy-Aware Hardware Design



- Cascaded Adder: given that the input pixel values are binary (0 or 1), the standard matrix multiplication operation common in neural networks is simplified.
- A cascaded adder replaces the need for multipliers, which are traditionally more resource-intensive on silicon.
- LIF Neuron Hardware: its primary functionality revolves around processing input signals, accumulating them over time, and generating an output spike under certain conditions, akin to a neuron firing in response to stimuli

$$U[t+1] = \beta U[t] + I[t+1] - U_{rest}$$

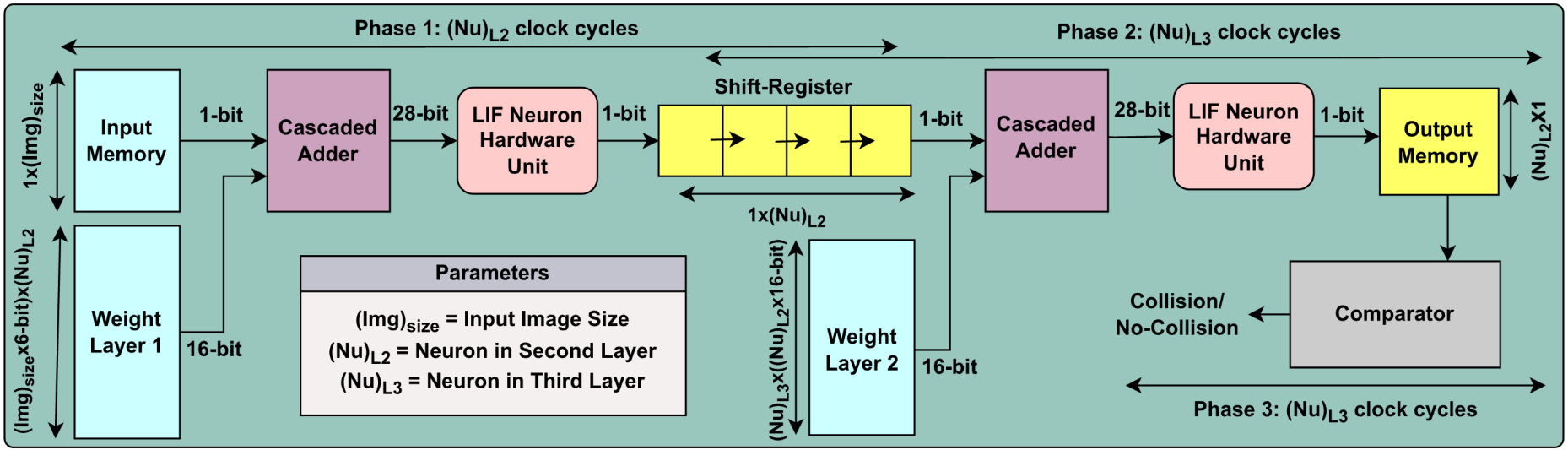


Proposed Energy-Aware Hardware Design

Energy-Aware Hardware Design

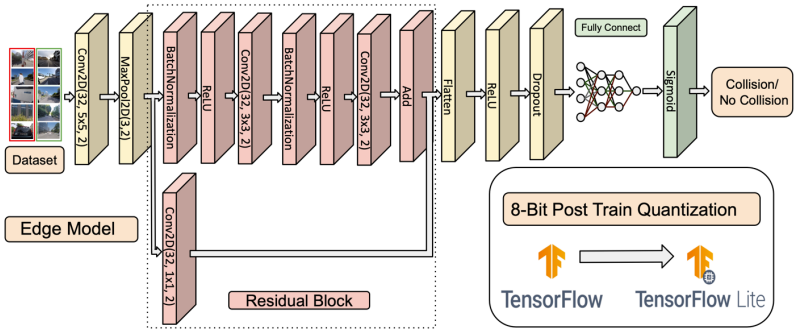
FPGA Inferencing using SNN

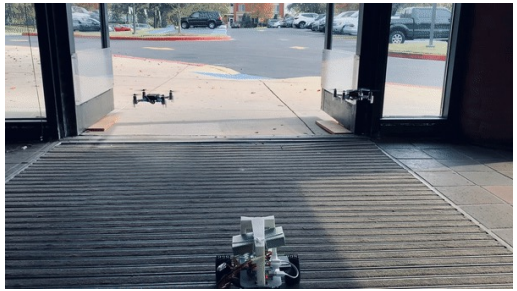
- The figure illustrates how data moves through weight layers, is processed by cascaded adders, undergoes transformation by Leaky Integrate-and-Fire (LIF) neurons, and finally, how classification is determined by a comparator.
- Each segment of the process is timed with clock cycles, showing a sequential processing order and synchronization within the hardware system.



Energy-Efficient TinyML Systems Deployment

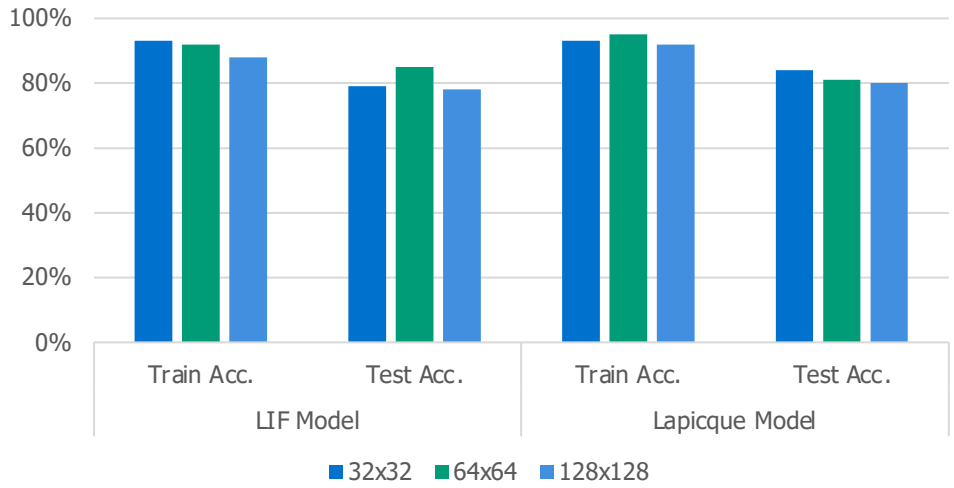
- Autonomous edge devices with AI-based visual navigation: Accuracy, Latency and Power consumption
- UAV: 320x320 RGB camera, 512 KB memory, octa-core GAP8 low-power processor
- UGV: 1 GB memory, Raspberry Pi
- HoloLens: Microsoft HoloLens 2
- Model Training Process:
  - Obstacle detection and Steering: Resnet trained on 33,000 images from car driving
  - Person detection: Yolo-v4, COCO dataset





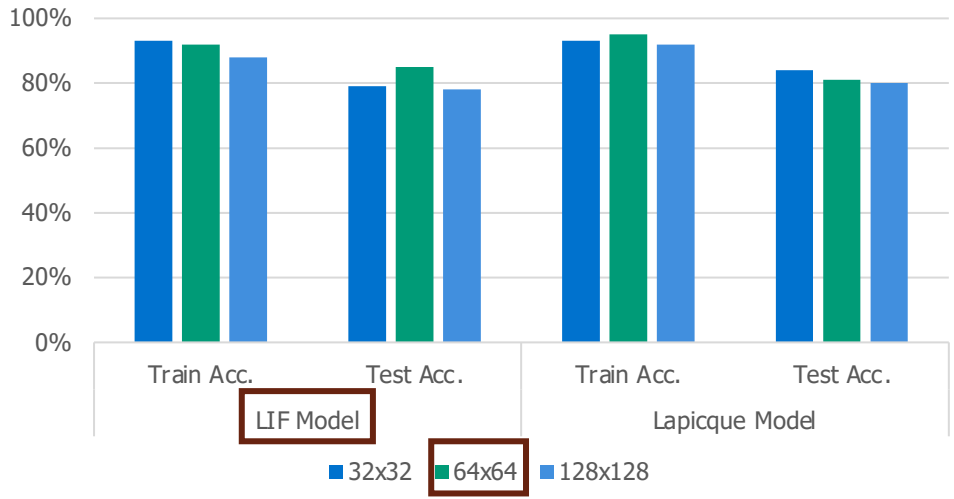
Software Results

- The LIF model demonstrated a training accuracy that slightly declined with increasing image size, starting at 93% for 32×32 images and decreasing to 88% for 128x128 images.
- Testing accuracy for the LIF model showed a similar trend, with the highest accuracy of 85% for 64×64 image size.
- The LIF model is preferred for hardware implementations where mimicking biological neuron behavior is critical, such as in real-time collision avoidance systems.



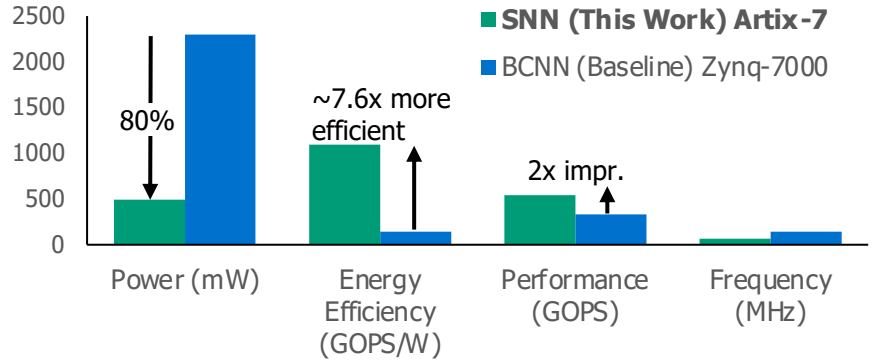
Software Results

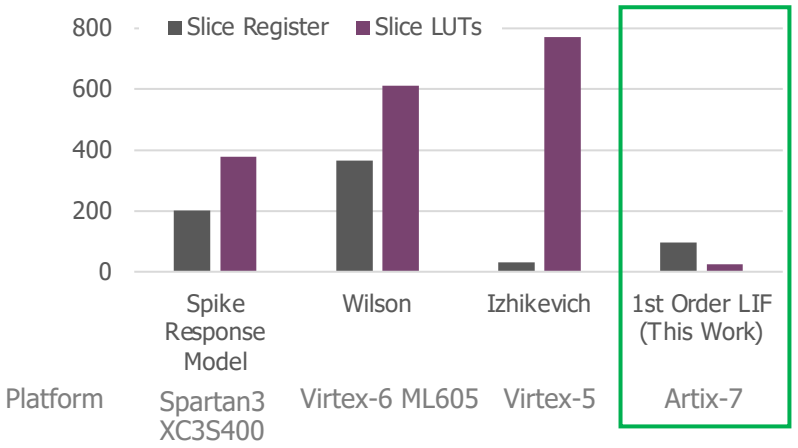
- The LIF model demonstrated a training accuracy that slightly declined with increasing image size, starting at 93% for 32×32 images and decreasing to 88% for 128x128 images.
- Testing accuracy for the LIF model showed a similar trend, with the highest accuracy of 85% for 64×64 image size.
- The LIF model is preferred for hardware implementations where mimicking biological neuron behavior is critical, such as in real-time collision avoidance systems.



FPGA Deployment Results

- To evaluate this work, we compare it with SOTA
  - Binarized CNN (BCNN): Nakahara'17
  - Spike Neural Mode: Walravens'20
  - Wilson: Karimi'18
  - Izhikevich: Murali'16
- Model deployment on FPGA for SNN and BCNN comparison
  - Power Consumption: 80% improvement
  - Energy Efficiency: ~7.6x more efficient
  - Performance: 2x faster
- Proposed Neuron resource utilization comparison with SOTA
  - The strategic balance between:
    - Resource utilization
    - Operational frequency
    - Power consumption
  - The proposed 1st Order LIF model
    - Operates at a frequency of 100 MHz
    - Exhibits lower power consumption at 85 mW





FPGA Deployment Results

- The operating frequency of the proposed SNN is 67 MHz, which is moderately placed among the referenced works.
- It is notably lower than the highest frequency presented, but this is a strategic choice to balance power consumption and processing speed, which is often a critical consideration in embedded and real-time applications.
- The comparison shows that while the proposed SNN requires more resources than some of the simpler models, it still operates effectively within the constraints of the Artix-7 device.

Table 4: Comparision of Proposed SNN with Previous Works

Reference	Slice Register	Slice LUT	Frequency (MHz)	Architecture	Device
[6]	1023	11339	189	25-5-1	Virtex-6 ML605
[25]	33	19	717	25-10 (interconnection)	Terasic DE1-SoC
[2]	119	587	25	25-5-2	Artix-7
This Work	**1780**	**6521**	**67**	4096-512-2	Artix-7

Conclusion

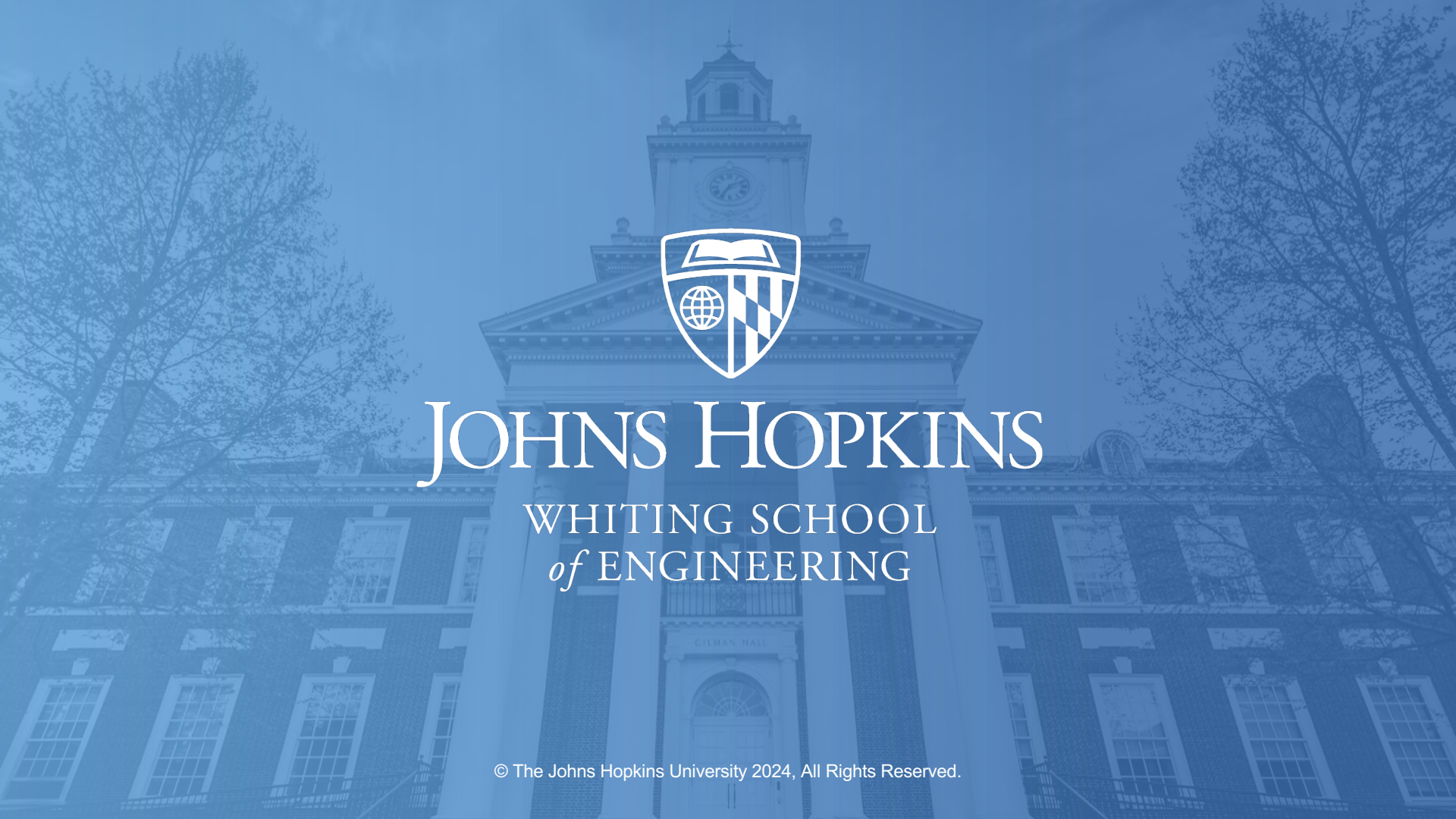
- Successfully introduced a SNN architecture optimized for energy-efficiency, ideal for deployment in edge devices.
- Demonstrated significant energy savings with the implementation of 1st order LIF model, paving way for sustainable AI and remote applications.
- Enabled vision-based machine learning algorithm to TinyML systems, opening new possibilities for advanced applications in compact devices in limited power.

References

- Jason K. Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu "Training Spiking Neural Networks Using Lessons From Deep Learning". Proceedings of the IEEE, 111(9) September 2023.
- Asmer Hamid Ali, Mohammad Zain, Syed Mehdi Kazim, and Mohd Hasan. 2022. Energy Efficient FPGA Implementation of a Spiking Neural Network. In 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT). 1–6. https://doi.org/10.1109/GCAT55367.2022.9971948
- Hiroki Nakahara, Tomoya Fujii, and Shimpei Sato. 2017. A fully connected layer elimination for a binarizec convolutional neural network on an FPGA. In 2017 27th International Conference on Field Programmable Logic and Applications (FPL). 1–4. https://doi.org/10.23919/FPL.2017.8056771
- Maximiliaan Walravens, Erik Verreycken, and Jan Steckel. 2020. Spiking Neural Network Implementation on FPGA for Robotic Behaviour. 694–703. https://doi. org/10.1007/978-3-030-33509-0_65
- Gholamreza Karimi, Morteza Gholami, and Edris Zaman Farsa. 2018. Digital implementation of biologically inspired Wilson model, population behavior, and learning. International Journal of Circuit Theory and Applications 46, 4 (2018), 965–977. https://doi.org/10.1002/cta.2457 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cta.2457
- Shanmukha Murali, Juneeth Kumar, Jayanth Kumar, and Ramesh Bhakthavatchalu. 2016. Design and implementation of Izhikevich spiking neuron model on FPGA. 946–951. https://doi.org/10.1109/RTEICT.2016.7807968
- Corey Lammie, Tara Hamilton, and Mostafa Rahimi Azghadi. 2018. Unsupervised Character Recognition with a Simplified FPGA Neuromorphic System. In 2018 IEEE International Symposium on Circuits and Systems (ISCAS). 1–5. https://doi.org/10.1109/ISCAS.2018.8351532

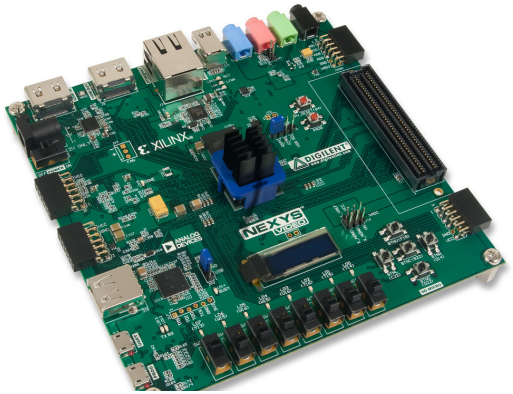
Thank You!
Any Questions?

Contact Information
Mozhgan Navardi: mnavard1@jhu.eu
Asmer Hamid Ali: aali56@jh.edu



Experimental Setup

- **Implementation Platform**
  - Selection of FPGA as a device for implantation is driven by considerations including FPGAs offer a unique blend of flexibility and performance, allowing for rapid prototyping and iterative design that is crucial for the evolving field of SNNs.
  - Selection of Xilinx Artix-7 FPGA for its balance between performance, cost, and energy efficiency.



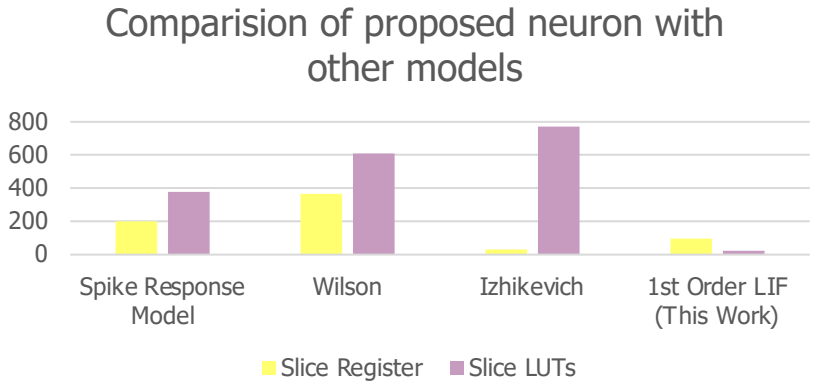
(Image source: AMD)

FPGA Deployment Results

- The proposed LIF neuron model gives strategic balance between resource utilization, operational frequency, and power efficiency.
- The proposed 1^{st} Order LIF model operates at a frequency of 100 MHz and exhibits lower power consumption at 85 mW.
- Moreover, the reduced power usage of the proposed model extends the potential for deployment in power sensitive contexts, such as battery-operated embedded systems or portable devices requiring efficient neural network computation.

Table 3: Comparison of Proposed Neuron Model With Previous Works

Neuron Model	Slice Registers	Slice LUTs	Frequency (MHz)	Power (mW)	Device
Spike Response Model [54]	202	378	100	-	Spartan3 XC3S400
Wilson [22]	365	611	98	-	Virtex-6 ML605
Izhikevich [36]	31	771	-	1043	Virtex-5
Simplified LIF [13]	29	70	100	-	Virtex 6
LIF [11]	297	196	100	0.48	Virtex-7 VX690T
LIF [2]	16	27	50	95	Artix-7
1^{st} Order LIF (This Work)	**96**	**25**	**100**	**85**	Artix-7



Copyright Notice

www.tinyml.org