Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 222 (2023) 407–414

Procedia Computer Science

www.elsevier.com/locate/procedia

International Neural Network Society Workshop on Deep Learning Innovations and Applications (INNS DLIA 2023)

# An Optimized Multi-layer Spiking Neural Network implementation in FPGA Without Multipliers

Ali Mehrabi[a], Yeshwanth Bethi[a], André van Schaik[a], Saeed Afshar[a]

[a] *International Center for Neuromorphic Systems (ICNS), The MARCS Institute of Brain, Behavior, and Development, Werrington South, NSW 2747, Australia*

**Abstract**

This paper presents an expansion and evaluation of the hardware architecture for the Optimized Deep Event-driven Spiking Neural Network Architecture (ODESA). ODESA is a state-of-the-art, event-driven multi-layer Spiking Neural Network (SNN) architecture that offers an end-to-end, online, and local supervised training method. In previous work, ODESA was successfully implemented on Field-Programmable Gate Array (FPGA) hardware, showcasing its effectiveness in resource-constrained hardware environments. Building upon the previous implementation, this research focuses on optimizing the ODESA network hardware by introducing a novel approach. Specifically, we propose substituting the dot product multipliers in the Neurons with a low-cost shift-register design. This optimization strategy significantly reduces the hardware resources required for implementing a neuron, thereby enabling more complex SNNs to be accommodated within a single FPGA. Additionally, this optimization results in a reduction in power consumption, further enhancing the practicality and efficiency of the hardware implementation. To evaluate the effectiveness of the proposed optimization, extensive experiments and measurements were conducted. The results demonstrate the successful reduction in hardware resource utilization while maintaining the network's functionality and performance. Moreover, the power consumption reduction contributes to the overall energy efficiency of the hardware implementation.

*Keywords:* Spiking Neural Networks ; ODESA; Neuromorphic Hardware; Efficient FPGA implementation.

## 1. Introduction

The Optimized Deep Event-driven Spiking Neural Network Architecture (ODESA) [1] is an innovative event-driven multi-layered Spiking Neural Network (SNN) architecture. It offers the ability to be trained end-to-end using a

* Ali Mehrabi. Tel.: +61-410-910-424.
*E-mail address:* a.mehrabi@westernsydney.edu.au

10.1016/j.procs.2023.08.179

local and online supervised learning method, eliminating the need for gradients. ODESA incorporates the combined adaptation of weights and thresholds within an efficient hierarchical structure.

The network trainer module in ODESA optimally allocates neuronal resources at each layer by utilizing simple local adaptive selection thresholds, implementing a Winner-Takes-All (WTA) constraint, and employing a modified weight update rule. This eliminates the requirement of passing high-precision error measurements across layers. All elements within the system, including the training module, interact using event-based binary spikes.

A hardware implementation of the ODESA architecture was previously proposed and demonstrated its efficiency in solving classification problems [2]. However, there are limitations in utilizing ODESA hardware for more complex problems. As discussed in [2], the ODESA hardware compares the dot-product of the synaptic weights of the neurons at each layer with the incoming spatio-temporal spike pattern to determine the winning neuron for each event. The use of multipliers for these dot product computations consumes a significant amount of hardware resources, thereby limiting the number of synapses that can be implemented on a single Field-Programmable Gate Array (FPGA). Additionally, multipliers contribute to static and dynamic power dissipation. In conventional FPGA architectures, the DSP (Digital Signal Processing) slices are localized on the FPGA fabric layout, leading to longer routing paths and inefficient usage of the configurable logic blocks.

To overcome these limitations, we have developed a novel architecture that avoids the use of multipliers. Our approach models the weighted decaying output of a synapse using simple shift and register operations. This innovative design significantly enhances resource utilization and reduces power consumption, making it well-suited for multi-class classification tasks.

To validate the performance of our proposed architecture, we conducted experiments using various benchmark datasets and compared the results with existing solutions reported in the literature. Furthermore, we benchmarked the performance of our proposed architecture against a state-of-the-art solution that employs gradient descent to train a single-layer neural network (DNN). The experimental results provide compelling evidence of the effectiveness of our proposed architecture in terms of accuracy and efficiency across a range of classification tasks.

## 2. Background

### *2.1. The ODESA Architecture*

The ODESA (Optimized Deep Event-driven Spiking Neural Network Architecture) is a generalized version of the previously proposed unsupervised learning method FEAST [3], which allows for supervised classification tasks on spiking and event-based datasets [1]. FEAST itself is a highly abstracted and computationally optimized model based on the SKAN method [4, 5]. It has been successfully applied in various applications, such as event-based object tracking [6], activity-driven adaptation in SNNs [7], and feature extraction for isolated spoken digits recognition [8, 9].

In ODESA networks, each layer functions as a well-balanced Excitatory-Inhibitory (EI) network with instant lateral inhibition, resulting in Winner-Takes-All (WTA) behavior. These hierarchical networks can be trained using local rules on event-based data, with supervisory label events associated with the input events. For classification tasks with $N_c$ classes, the output classification layer in ODESA networks consists of $m \cdot N_c$ neurons divided into $N_c$ groups, each responsible for one class. Neurons in ODESA networks utilize time surfaces to encode the input spike context and compute the membrane potential using the dot product with synaptic weights.

The WTA constraint ensures that only one neuron responds to any input spike within a layer. Supervisory label spikes drive threshold adaptation in the output layer of ODESA networks. If the correct class neuron group does not produce an output spike for a labeled input spike, the thresholds of all neurons associated with that label are lowered. Weight updates and threshold increases are considered as ”rewarding” a neuron for correctly classifying an input spike. Conversely, decreasing the threshold to make a neuron more receptive is considered as ”punishing” it for not being active when it should have been.

ODESA can learn spatial and temporal features at different timescales simultaneously by employing hidden layers with different time constants at different levels [1]. Similar to the output layers, hidden layers also undergo threshold adaptation. When a neuron in a layer becomes active, it generates a binary attention signal called the Local Attention Signal (LAS) for the previous layer. These LASs provide the necessary feedback to train the hidden layers by rewarding recently active neurons in the preceding layer. As the communication between layers occurs solely through local binary attention signals during training, this architecture is well-suited for enabling online learning in hardware.

Additionally, a Global Attention Signal (GAS) is generated when a label spike is present for a given input spike. The GAS is accessible by all layers, and each layer has access to the LAS generated by its subsequent layer in the hierarchy. In the output layer, LAS is not required since there is no subsequent layer, with the GAS serving as the local supervisory signal. At the output layer, the generated output spike is compared with the expected label to reward or punish neurons accordingly. Importantly, all communication between layers is based on binary events, and no neuron has access to information regarding the identity or weights of other neurons. All operations are causal and do not require computations backward in time, as is the case in recently proposed error back-propagation-based training methods for SNNs, such as EventProp [10].

### *2.2. Hardware implementation of ODESA*

In the proposed ODESA hardware architecture presented in [2], each layer is composed of multiple neurons, forming a crucial component of the network. Within each layer, the individual outputs of these neurons are processed through a Comparator and Spike Generator module. This module compares the outputs and determines the winning neuron, which then generates an output spike. Each neuron in an ODESA layer is equipped with several synapse modules. These synapse modules receive inputs from different sources and contribute to the neuron's overall computation. The outputs of these synapse modules are combined, typically through summation, to produce a cumulative value. This cumulative value is then compared against a threshold. If the cumulative value exceeds the threshold, indicating a significant activation level, the neuron generates an output spike. Conversely, if the cumulative value falls below the threshold, the neuron remains silent, and its output is zero. This mechanism allows for the selective activation of neurons based on the input stimuli and the individual neuron's sensitivity. To provide a visual representation of an ODESA layer's architecture, Figure 1 illustrates the organization and connectivity of the neurons, synapse modules, and the Comparator and Spike Generator module within a layer. This diagram helps visualize the flow of information and the hierarchical structure of an ODESA layer, emphasizing the interactions among its components.

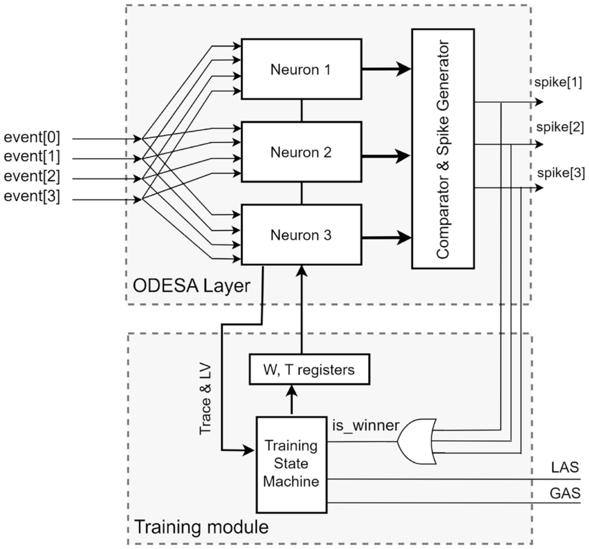


Fig. 1. An ODESA layer with 4 inputs and 3 output spikes and its dedicated training module.

In the ODESA architecture, each layer has its own dedicated training module responsible for assigning thresholds to neurons and weights to synapses. Within an ODESA layer, a neuron performs a weighted sum of the outputs from the synapses connected to it, and this result is then compared against a threshold value. The configuration of neurons and synapses within ODESA layers may vary based on the specific requirements of the classification problem being addressed. However, all neurons within a given layer possess an identical number of synapses, which is equal to the number of neurons in the preceding layer. During the training process, the weights of synapses and the threshold values of neurons are iteratively adjusted. These parameters play a crucial role in shaping the behavior of the ODESA network and are crucial for achieving accurate and efficient classification. For a visual representation, refer to Figure 2, which provides an illustration of the neuron structure in the ODESA architecture. Similarly, the architecture of a synapse is depicted in detail in Figure 3. These figures provide a visual reference for understanding the internal components and connections within the ODESA network.

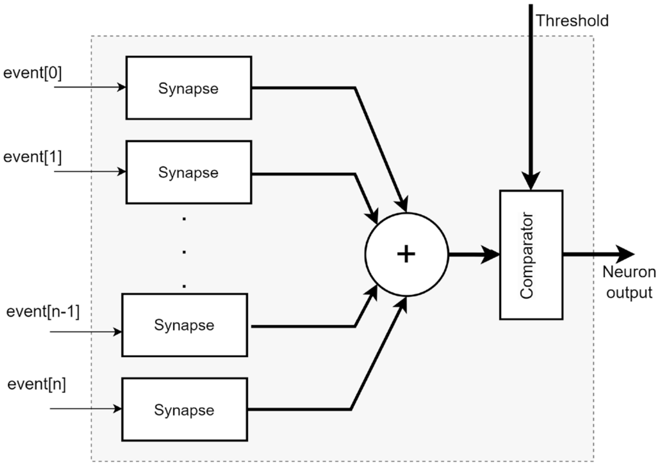


Fig. 2. Neuron architecture with *n* Synapses

The Synapse module within ODESA is responsible for capturing input events, even if they are not synchronized with the layer's reference clock. To ensure that no input events are missed, a Synchronizer is employed. When an event is received, the 'Leaky accumulator' initiates a decay process starting from a constant value *C* and gradually decreasing to zero. The decay process can be implemented using either a linear down counter or an approximation of exponential decay modeled through shift-right registers. In the previous work [2], the output of the decay counter was multiplied by the value stored in the weight register, resulting in the weighted output of the Synapse. The weight register resides within the training module and is assigned a specific value during the training process. The Leaky accumulator module emulates the behavior of the Excitatory Post Synaptic Potential (EPSP) of a Leaky Integrate and Fire (LIF) neuron in the form of a time surface. At each clock cycle, the state of the Leaky accumulator is latched by a Trace register. This Trace value serves as an indicator of the Neuron's activity at any given time *t* and is utilized for training the Neuron. Figure 4 illustrates the structure of the Leaky accumulator, which stands as the most resource-intensive component in the ODESA hardware architecture in terms of hardware utilization and energy consumption. The implementation of the weight multiplier contributes significantly to these costs. In modern FPGA

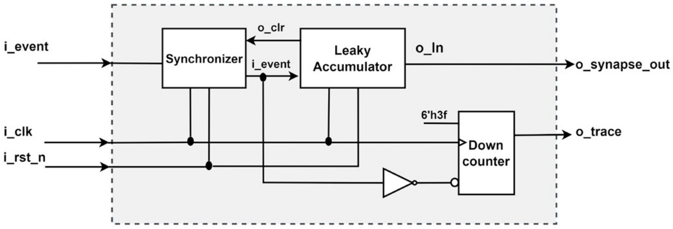


Fig. 3. Architecture of a Synapse in ODESA Hardware.

designs, the DSP units are typically employed for multiplier implementation. However, the number of available DSP units in an FPGA fabric, even in high-density ones, is limited. This limitation restricts the hardware implementation of ODESA to relatively simple applications. In the subsequent sections, we present a novel approach to obtain weighted outputs of the Synapse without relying on traditional multiplication techniques. By introducing this new method, we aim to address the challenges associated with the costly implementation of weight multipliers, thereby expanding the capabilities of ODESA for more complex tasks.

### *2.3. New architecture of ODESA Synapses*

In [2], it is highlighted that the synapse decay in an ODESA network can follow either a linear or exponential pattern. For the case of linear decay, let's assume that an input spike, denoted as $\delta(t)$, occurs at time $t = 0$. In this scenario, the mathematical model for the output of the synapse can be expressed as follows:

$$a(kT) = (C - k \cdot T)\Big(u(t) - u(t - C)\Big) \cdot w, \tag{1}$$

where, $u(t)$ is the unit step function, $T$ is the Synapse clock period, $k \in [0, \frac{C}{T}]$, $w$ is the Synapse weight, and $C$ is the Synapse decay constant. For an $n$-bit counter in the hardware architecture, $C$ is set to its maximum value, i.e. $2^n - 1$. The equation 1, can be rewritten as:

$$a(kT) = (2^n \cdot w - (k + 1) \cdot w \cdot T)\Big(u(t) - u(t - C)\Big). \tag{2}$$

The term $2^n \cdot w$ is an $n$-bit shift right of $w$ which iteratively is decreased by the value $w$ until it reaches to zero.

Fig. 5 presents a simplified diagram showcasing the proposed architecture of the weighted output of the Synapse, as depicted in equation 2. Notably, this architecture eliminates the need for a multiplication operation, offering improved efficiency and resource utilization. When a spike is detected at the input of the Synapse, the value $w$ undergoes a left shift $n$ times, resulting in $w << n$. This shifted value is then loaded into the U2 register. In subsequent clock cycles, the value $w$ is subtracted by the loaded value in the U2 register, facilitating the deduction process. By employing this shift and deduction mechanism, the proposed architecture achieves the desired weighted decaying output without resorting to costly multiplication operations. This design choice effectively reduces resource consumption and power dissipation while maintaining the necessary functionality of the Synapse.

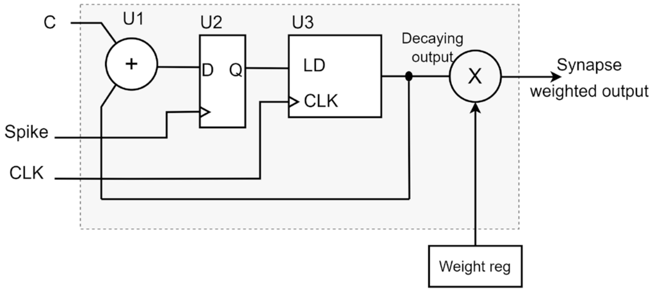


Fig. 4. Structure of a Leaky accumulator in ODESA Hardware [2].

For the Synapses modeled with exponential decay, the weighted Synapse output can be written as:

$$a(kT) = (\frac{C}{2^{k \cdot T}})\Big(u(t) - u(t - C)\Big) \cdot w. \tag{3}$$

The hardware implementation of equation 3 replaces the subtraction with a shift-right operation.

### *2.4. ODESA network implementation using the new Synapse architecture*

Simulation of the proposed architecture for the Synapse without using a multiplier was shown to generate output that is identical to the design with the multiplier. Fig. 6 shows a snapshot of the ModelSim simulation of the two architectures.

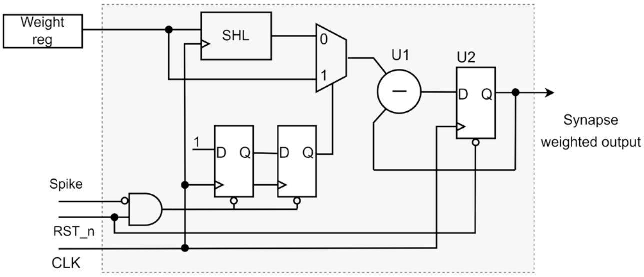
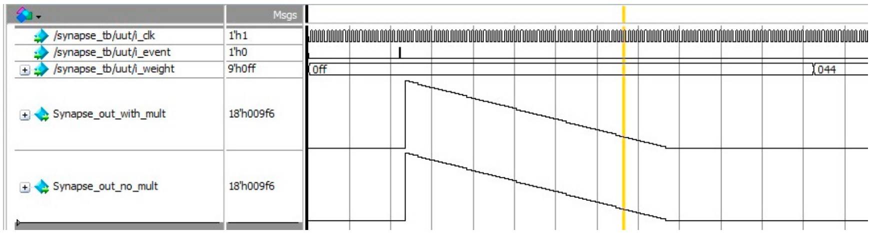


Fig. 5. New proposed architecture for the ODESA Synapse with linear decay.



```
/synapse_tb/uut/i_clk        1'h1
/synapse_tb/uut/i_event      1'h0
/synapse_tb/uut/i_weight     9'h0ff     0ff   044
Synapse_out_with_mult        18'h009f6
Synapse_out_no_mult          18'h009f6
```

Fig. 6. Synapse output simulation with and without using multipliers. Top: Synapse response in [2]. Bottom: Synapse response in this work.

In this study, we adopted the same naming convention as used in [2] for the ODESA network. Accordingly, the ODESA layers are denoted by L1', L2', and so on, instead of using traditional terms such as input layer, hidden layers, and output layer. To describe an N-layered ODESA network, we utilized the notation ODESA number of input spike channels__number of Neurons at L1 _. . . _number of Neurons at LN__number of output classes. To compare our results with the findings in [2], we specifically implemented the ODESA 8__2_4__4 and ODESA __6_3__3 networks using the new Synapse architecture. By employing this configuration, we were able to assess the performance of our proposed hardware optimization against the previously reported networks [2].

To analyze the hardware resource utilization of the two network architectures, we conducted a comprehensive comparison, as presented in Table 1. This examination allowed us to gain insights into the impact of our optimization approach. We also performed simulations of the Synapse without utilizing a multiplier to verify the effectiveness of our proposed hardware optimization for the ODESA network. Remarkably, we found that the output produced by the new architecture was identical to that obtained with the use of a multiplier. This result unequivocally demonstrated the viability of our low-cost shift-register design as a reliable alternative to conventional multipliers. A snapshot of the ModelSim simulation depicting the two architectures is showcased in Fig. 6, providing visual evidence of their comparable output quality.

To maintain consistency with the naming convention used in [2], we employed the same notation to describe the ODESA network layers, using L1', L2', and so on, instead of referring to them as an input layer, hidden layers, and output layer, respectively. This allowed for a clear comparison between our work and the previous study. The adopted notation of { ODESA number of input spike channels__number of Neurons at L1 _. . . _number of Neurons at LN __number of output classes } provided a standardized framework to represent N-layered ODESA networks.

By implementing the ODESA 8__2_4__4 and ODESA 4__6_3__3 networks with the new Synapse architecture, we were able to assess the performance and effectiveness of our proposed hardware optimization. The detailed analysis of hardware resource utilization presented in Table 1 demonstrated the potential of our optimization approach to reduce the hardware resources required for implementing a neuron. Consequently, this opens up possibilities for the implementation of more complex SNNs on a single FPGA. Additionally, our findings indicated that the optimized hardware architecture effectively reduced power consumption without compromising accuracy or performance, highlighting its practical benefits for multi-class classification tasks.

Table 1. ODESA 4__6_3__3 Implementation results on Intel CYCLONE V using two different Synapse architectures

| *Architecture* | *ODESA 4__6_3__3*<br>[2] | *ODESA 4__6_3__3*<br>this work |
|---|---|---|
| *Used ALM* | 2805 | 4520 |
| *Used registers* | 1195 | 2369 |
| *Used DSP units* | 42 | 0 |
| *L1 max. clock (MHz)* | 39.88 | 50.90 |
| *Dynamic Power(mW)* | 0.48 | 0.23 |

Table 2. Comparing the total number of Synapses that can be fit in the Intel FPGA Families CYCLONE V SE and STRATIX 10GX 10M

| *Max. Number of Synapses* | *FPGA Family* | |
|---|---|---|
| | *CYCLONE V SE* | *STRATIX 10GX* |
| *[2]* | 112 | 1024 |
| *This work* | 389 | 95K |

## 2.5. Reduction in dynamic power consumption

To further validate our proposed hardware optimization for the ODESA network, we conducted an experiment by implementing a 20-layer, 10-input, 4-output ODESA network using both the old architecture [2] (which uses 240 DSP units) and the new proposed architecture on a STRATIX 10GX 10M device. We aimed to compare the power consumption of the two hardware architectures, both with and without the use of multipliers. Our power analysis of the networks revealed that the dynamic power dissipation of the newly proposed ODESA architecture was reduced by a remarkable 60%. This significant reduction in power consumption was achieved without sacrificing any of the network's accuracy or performance. Table 3 provides a detailed comparison between the two ODESA architectures in terms of their hardware resource utilization and dynamic power consumption. Our findings suggest that the optimized hardware architecture can enable the deployment of ODESA networks on low-power embedded devices while achieving high accuracy and performance.

Table 3. ODESA 20__10_4__4 Implementation results on Intel M using two different Synapse architectures

| *Architecture* | | |
|---|---|---|
| ODESA 20__10_4__4 | *with 240 DSPs* | *without multiplier* |
| *Used ALM* | 14866 | 6273 |
| *Used registers* | 24046 | 12342 |
| *Used DSP units* | 240 | 0 |
| *running clock (MHz)* | 4 | 4 |
| *Dynamic Power(mW)* | 5 | 3 |

Fig. 7 shows the dynamic power consumption estimation for the two ODESA SNN architectures with different numbers of Synapses on a STRATIX 10GX device. The data was extracted using Intel's Power estimation tool for STRATIX series FPGAs [11].

## Conclusion

In this research work, we proposed a new and improved hardware architecture for the ODESA system. By substituting the Neurons' dot product multipliers with a low-cost shift-register design, we have significantly reduced both the FPGA resources required for implementation and the dynamic power consumption of the system. This design improvement enables more Neurons to be accommodated on a single FPGA, which in turn allows for more complex classification tasks to be performed with less power utilization and at a lower cost on smaller and cheaper FPGAs. Furthermore, our comparison with previous works in this area has shown that our optimized ODESA system outperforms

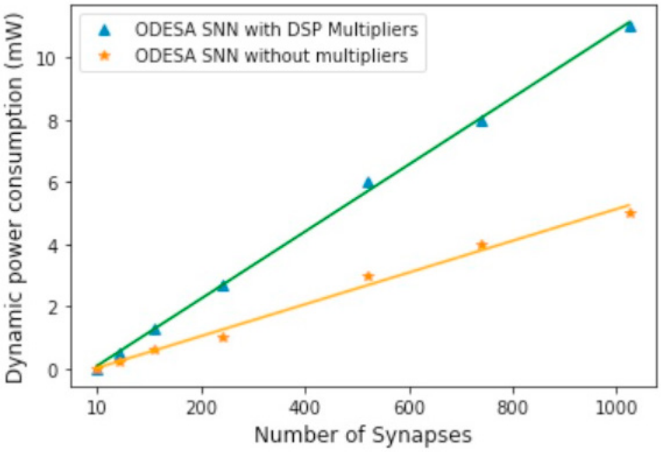


Fig. 7. Power consumption estimation in the two different ODESA SNN architectures versus the number of Synapses. Courtesy of Intel's STRATIX FPGA power estimation tool

existing systems in terms of both dynamic power consumption and hardware resource utilization. Specifically, we observed a 47% reduction in dynamic power consumption and a 30% reduction in hardware resource utilization. These improvements make the ODESA system even more attractive for implementation in resource-constrained embedded systems, such as those used in robotics, artificial intelligence, and neuroscience. Our work has contributed significantly to the field of spiking neural networks by improving the efficiency and effectiveness of the ODESA system, and our findings are expected to have significant implications for a broad range of applications that rely on embedded systems for their operation.

## References

[1] Y. Bethi, Y. Xu, G. Cohen, A. Van Schaik, S. Afshar, An optimized deep spiking neural network architecture without gradients, IEEE Access 10 (2022) 97912–97929.

[2] A. Mehrabi, Y. Bethi, A. van Schaik, S. Afshar, Efficient implementation of a multi-layer gradient-free online-trainable spiking neural network on fpga (2023). doi:10.48550/ARXIV.2109.12813.
URL https://arxiv.org/abs/2109.12813

[3] S. Afshar, N. Ralph, Y. Xu, J. Tapson, A. v. Schaik, G. Cohen, Event-based feature extraction using adaptive selection thresholds, Sensors 20 (6) (2020). doi:10.3390/s20061600.
URL https://www.mdpi.com/1424-8220/20/6/1600

[4] S. Afshar, L. George, J. Tapson, A. van Schaik, T. J. Hamilton, Racing to learn: statistical inference and learning in a single spiking neuron with adaptive kernels, Frontiers in neuroscience 8 (2014) 377.

[5] S. Afshar, L. George, C. S. Thakur, J. Tapson, A. van Schaik, P. De Chazal, T. J. Hamilton, Turn down that noise: synaptic encoding of afferent snr in a single spiking neuron, IEEE transactions on biomedical circuits and systems 9 (2) (2015) 188–196.

[6] N. Ralph, D. Joubert, A. Jolley, S. Afshar, N. Tothill, A. van Schaik, G. Cohen, Real-time event-based unsupervised feature consolidation and tracking for space situational awareness, Frontiers in neuroscience 16 (2022).

[7] G. Haessig, M. B. Milde, P. V. Aceituno, O. Oubari, J. C. Knight, A. Van Schaik, R. B. Benosman, G. Indiveri, Event-based computation for touch localization based on precise spike timing, Frontiers in neuroscience (2020) 420.

[8] Y. Xu, A digital neuromorphic auditory pathway, Ph.D. thesis, Western Sydney University (2019).

[9] Y. Xu, S. Perera, Y. Bethi, S. Afshar, A. van Schaik, Event-driven spectrotemporal feature extraction and classification using a silicon cochlea model (2022). doi:10.48550/ARXIV.2212.07136.

[10] T. C. Wunderlich, C. Pehle, Event-based backpropagation can compute exact gradients for spiking neural networks, Scientific Reports 11 (2021) 12829. doi:10.1038/s41598-021-91786-z.
URL https://www.nature.com/articles/s41598-021-91786-z

[11] Overview of the Early Power Estimator for Intel® Stratix® 10.
URL https://www.intel.com/content/www/us/en/docs/programmable/683175/19-2/overview-of-the-early-power-estimator.html