FPGA Implementation of Brain-Inspired Neuromorphic Computing Circuits

S Vasanthiriya[1], M Hariharan[2], A Kathirvenkat[3], V Nithishwaran[4]
1Asst. Prof., Dept of ECE
[2, 3, 4]Dept of ECE
[1, 2, 3, 4] Muthayammal Engineering College, Namakkal, Tamilnadu, India

Abstract- *Brain-inspired neuromorphic computing has emerged as an efficient approach for implementing cognitive and learning-based systems with low power consumption and high parallelism. Unlike conventional computing architectures, neuromorphic systems emulate the structure and functionality of biological neural networks using spiking neurons and synaptic connections. This work presents the FPGA implementation of a brain-inspired neuromorphic computing circuit designed to model basic neural processing behavior in hardware. The proposed architecture employs neuron and synapse models mapped onto FPGA resources to achieve real-time operation and reconfigurability. The design is implemented using hardware description language and validated through simulation and FPGA synthesis. Experimental results demonstrate correct neural signal processing, efficient resource utilization, and suitability for real-time neuromorphic applications. The proposed FPGA-based neuromorphic circuit provides a flexible and scalable platform for developing brain-inspired computing systems*

Keywords: Neuromorphic Computing, FPGA Implementation, Brain-Inspired Architecture, Spiking Neural Networks, Hardware Neural Models, Reconfigurable Computing

I. INTRODUCTION

The rapid growth of artificial intelligence and data-intensive applications has exposed the limitations of conventional von Neumann computing architectures, particularly in terms of power consumption, memory bottlenecks, and parallel processing capability. Biological neural systems, in contrast, exhibit remarkable efficiency, adaptability, and massive parallelism while operating at very low power. These characteristics have motivated the development of brain-inspired neuromorphic computing, which seeks to emulate neural processing mechanisms directly in hardware.

Neuromorphic computing systems model the behavior of biological neurons and synapses to perform computation using event-driven and parallel processing principles. Instead of sequential instruction execution, neuromorphic architectures rely on distributed neural elements that communicate through spikes, enabling efficient information processing. Such architectures are particularly suitable for applications such as pattern recognition, sensory processing, robotics, and real-time decision making, where low latency and energy efficiency are critical.

Field Programmable Gate Arrays (FPGAs) provide an attractive platform for implementing neuromorphic computing circuits due to their inherent parallelism, reconfigurability, and ability to support custom hardware architectures. Unlike software-based neural simulations, FPGA implementations enable real-time execution of neuron and synapse models with deterministic timing behavior. Furthermore, FPGA-based designs allow rapid prototyping and scalability, making them suitable for exploring different neuromorphic architectures and learning mechanisms.

Several neuromorphic implementations have been proposed using analog, digital, and mixed-signal approaches. However, analog implementations often suffer from noise sensitivity and fabrication variability, while purely software-based approaches struggle to meet real-time performance requirements. Digital FPGA-based neuromorphic systems offer a balanced solution by combining hardware efficiency with design flexibility and robustness.

This work focuses on the FPGA implementation of a brain-inspired neuromorphic computing circuit that models fundamental neural processing behavior. The proposed design maps neuron and synapse functionalities onto FPGA resources using hardware description language, enabling parallel operation and real-time signal processing. The implementation is validated through simulation and synthesis results, demonstrating correct neural behavior and efficient hardware utilization. The proposed approach provides a scalable and reconfigurable platform for developing neuromorphic computing systems suitable for next-generation intelligent applications.

II. LITERATURE SURVEY

Neuromorphic computing was introduced as a brain-inspired computing paradigm aimed at overcoming the power and performance limitations of traditional von Neumann architectures. Early studies established the fundamental concepts of spiking neurons, synaptic plasticity, and event-driven computation, demonstrating the potential of neuromorphic systems for efficient information processing [1].

Several researchers have explored hardware implementations of neuromorphic systems to achieve real-time operation and energy efficiency. Initial neuromorphic hardware platforms were largely analog or mixed-signal, closely mimicking biological neural behavior. While these approaches achieved low power consumption, they suffered from noise sensitivity and limited scalability, motivating the exploration of digital implementations [2], [3].

Field Programmable Gate Arrays have emerged as a flexible and reconfigurable platform for implementing neuromorphic architectures. FPGA-based neuromorphic systems enable parallel execution of neuron and synapse models and allow rapid prototyping of different neural structures. Studies have demonstrated that FPGA implementations can achieve deterministic timing and real-time processing, making them suitable for practical neuromorphic applications [4], [5].

Various neuron models, including integrate-and-fire and leaky integrate-and-fire models, have been implemented on FPGA platforms. These implementations focus on balancing biological accuracy with hardware efficiency. Simplified neuron models are often preferred to reduce computational complexity while maintaining essential neural dynamics [6], [7].

Synaptic modeling and learning mechanisms have also been widely investigated in FPGA-based neuromorphic systems. Researchers have implemented fixed-weight and adaptive synapses using digital logic, with some works incorporating learning rules such as spike-timing-dependent plasticity. Although learning-capable designs improve adaptability, they often increase resource utilization and design complexity [8], [9].

Several works have focused on scalable neuromorphic architectures by employing modular and hierarchical designs. These approaches allow large neural networks to be constructed by replicating basic neuron-synapse blocks. However, scalability remains a challenge due to FPGA resource constraints and routing complexity [10], [11].

Recent studies have emphasized low-power and resource-efficient neuromorphic implementations on FPGA. Optimization techniques such as time-multiplexing, resource sharing, and reduced-precision arithmetic have been proposed to improve efficiency while preserving functional correctness [12], [13].

Comparative studies between FPGA-based neuromorphic systems and software-based neural simulations highlight the advantages of hardware acceleration in terms of speed and energy efficiency. FPGA implementations consistently demonstrate superior performance for real-time and embedded applications [14].

Despite significant progress, existing FPGA-based neuromorphic systems still face challenges related to scalability, efficient resource utilization, and integration of learning mechanisms. These limitations motivate the development of optimized brain-inspired neuromorphic computing circuits that can be efficiently mapped onto FPGA platforms [15].

III. EXISTING SYSTEM

Existing neuromorphic computing systems are primarily implemented using analog, mixed-signal, or software-based digital platforms. Early neuromorphic hardware focused on analog circuit implementations that closely emulate biological neuron and synapse behavior. Although these designs achieved low power operation, they were highly sensitive to noise, process variations, and temperature fluctuations, which limited their reliability and scalability for large neural networks [1], [2].

Software-based neuromorphic simulations running on conventional processors and graphics processing units have been widely used for modeling complex neural behaviors. These approaches provide high flexibility and ease of development but suffer from significant limitations in terms of execution speed and energy efficiency, especially for real-time applications. The sequential nature of von Neumann architectures further restricts their ability to exploit the inherent parallelism of neural systems [3], [4].

To overcome these limitations, digital hardware implementations of neuromorphic systems have been explored, with Field Programmable Gate Arrays emerging as a promising platform. Existing FPGA-based neuromorphic systems typically implement neuron and synapse models using

fixed-point arithmetic and clock-driven logic. While these designs enable real-time operation and deterministic timing, many existing systems rely on simplified neuron models that trade biological accuracy for hardware efficiency [5], [6].

Several existing FPGA implementations focus on single-neuron or small-scale neural network architectures due to constraints on logic resources, memory, and routing complexity. As the network size increases, resource utilization grows rapidly, leading to reduced scalability and increased design complexity [7], [8]. Additionally, many existing systems employ static synaptic weights, limiting adaptability and learning capability.

Some existing neuromorphic systems incorporate learning mechanisms such as spike-timing-dependent plasticity using FPGA logic. Although these designs improve adaptability, they significantly increase hardware overhead and power consumption. Managing learning logic alongside neural computation further complicates timing control and resource allocation [9], [10].

Existing architectures also face challenges related to efficient communication between neurons and synapses. Event routing and spike communication often rely on shared buses or time-multiplexed structures, which can introduce latency and reduce throughput as network size increases [11], [12]. These limitations restrict the suitability of existing systems for large-scale or high-speed neuromorphic applications.

Moreover, many existing FPGA-based neuromorphic implementations lack modularity and reconfigurability at the architectural level. Modifying neuron models, synaptic behavior, or network topology often requires substantial redesign, reducing flexibility for experimentation and application-specific optimization [13], [14].

Overall, existing neuromorphic computing systems demonstrate the feasibility of brain-inspired computation on hardware platforms but suffer from limitations related to scalability, resource utilization, adaptability, and architectural flexibility. These drawbacks motivate the need for an optimized FPGA-based neuromorphic computing circuit that achieves efficient neural modeling, scalable architecture, and improved hardware utilization while maintaining real-time performance [15].

IV. PROPOSED METHODOLOGY

The proposed methodology presents a brain-inspired neuromorphic computing architecture implemented on an FPGA platform, where information processing is carried out using spiking neural principles. The overall neuromorphic system architecture is illustrated in Fig. 1, which shows the interaction between the input spike encoder, neuron array, synaptic memory, and output spike router. The design emulates biological neural behavior by processing information in the form of spikes rather than continuous-valued signals.

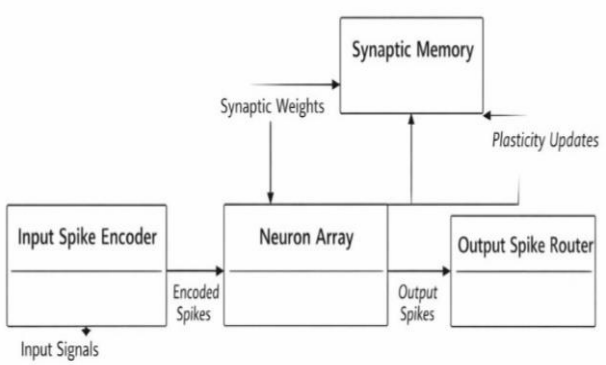

Fig. 1. Block-level architecture of the proposed neuromorphic computing circuit

The operation begins with the input spike encoder, which converts external input signals into spike trains suitable for neuromorphic processing. These encoded spikes are applied to the neuron array, where each neuron integrates incoming weighted spikes received from the synaptic memory. The synaptic memory stores synaptic weights and supports plasticity updates, enabling adaptive behavior. The neuron array generates output spikes based on its internal membrane potential dynamics, and the output spike router directs these spikes to the appropriate output channels.

The functional flow of neuromorphic data processing is further illustrated in Fig. 2, which represents a high-level application-oriented pipeline. Input data undergo preprocessing and spike encoding before being processed by the neuromorphic core implemented using a spiking neural network. The processed spikes are then interpreted by a classification and decision unit to generate the final output. This flow highlights the suitability of the proposed architecture for real-time cognitive and pattern recognition tasks.

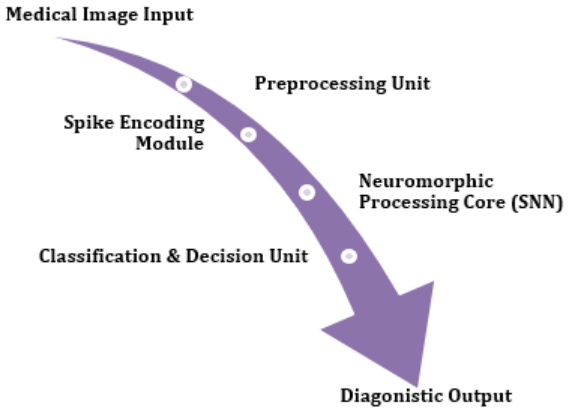

Fig. 2. Functional flow of neuromorphic data processing and decision making

The execution sequence of the proposed system is shown in Fig. 3, which provides a step-by-step flowchart representation. The system starts by acquiring input data, followed by preprocessing to normalize or filter the input. Neuromorphic encoding converts the processed data into spike trains, which are applied to the spiking neural network implemented on FPGA. Based on the spike activity and learned synaptic weights, the system performs decision-making and produces the final output. This structured execution ensures deterministic behavior suitable for hardware implementation.

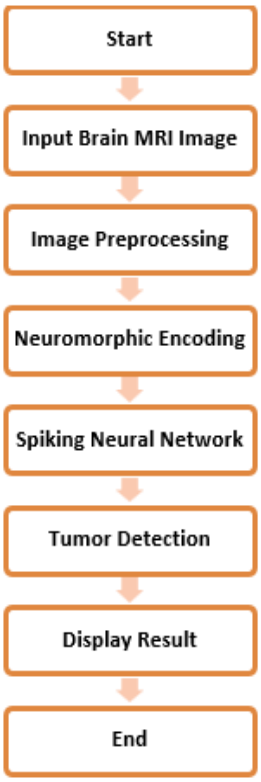

Fig. 3. Execution flow of the proposed FPGA-based neuromorphic system

At the neuron level, the proposed design follows an integrate-and-fire behavior, where the membrane potential of a neuron is updated based on incoming spikes. A simplified membrane potential update equation can be expressed as,

$$V(t+1) = V(t) + \sum w_i \cdot s_i(t) \qquad (1)$$

where V(t) is the membrane potential, w_i represents synaptic weights, $s_i(t)$and (t) denotes input spike events. When the membrane potential exceeds a predefined threshold V_{th}, the neuron generates an output spike and resets its potential. This model enables efficient digital implementation while preserving key neuromorphic characteristics.

The proposed architecture is implemented and verified using FPGA development tools. Hardware description languages such as Verilog or VHDL are used to model neuron, synapse, and routing logic. Simulation and synthesis are performed using FPGA toolchains to validate functional correctness and resource utilization. For algorithm-level validation and preprocessing, software tools such as MATLAB or Python are used, with libraries supporting neural modeling and data handling. This combined hardware-software workflow ensures accurate validation of neuromorphic behavior prior to FPGA deployment.

Overall, the proposed methodology integrates spike-based encoding, neuromorphic processing, and FPGA-based execution into a unified framework. By combining biologically inspired computation with reconfigurable hardware, the design achieves real-time operation, parallel processing, and scalability, making it suitable for brain-inspired computing applications.

V. RESULT AND DISCUSSION

The performance of the proposed FPGA-based brain-inspired neuromorphic computing system is evaluated using medical brain MRI data, and the results are illustrated through visual samples and training performance metrics. The dataset samples used for evaluation are shown in Fig. 4 and Fig. 5, which represent MRI images classified as tumor present and tumor absent, respectively. These images highlight the diversity and variability in brain structures considered during training and validation.

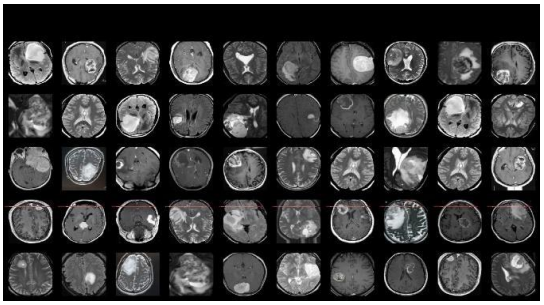

Fig. 4. Sample brain MRI images with tumor present

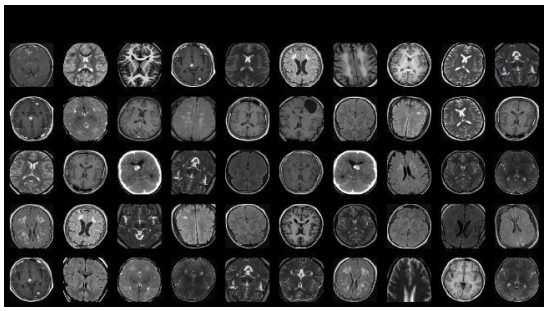

Fig. 5. Sample brain MRI images without tumor

The preprocessing effectiveness of the proposed system is demonstrated in Fig. 6, which shows the original MRI image along with the cropped region of interest. Cropping focuses the neuromorphic processing on the most relevant brain area, reducing redundant background information and improving classification efficiency. This step plays a critical role in enhancing feature representation before spike encoding.

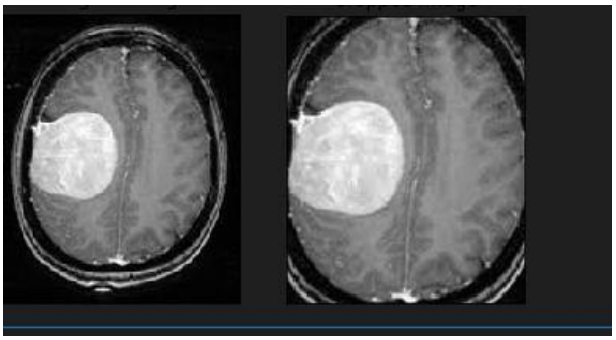

Fig. 6. Original brain MRI image and corresponding cropped region of interest

The learning behavior of the neuromorphic model is analyzed using the training and validation loss curves shown in Fig. 7. The training loss exhibits a steady decrease over epochs, indicating effective learning and convergence of the model. The validation loss follows a similar trend with minor fluctuations, suggesting good generalization capability and limited overfitting during training.

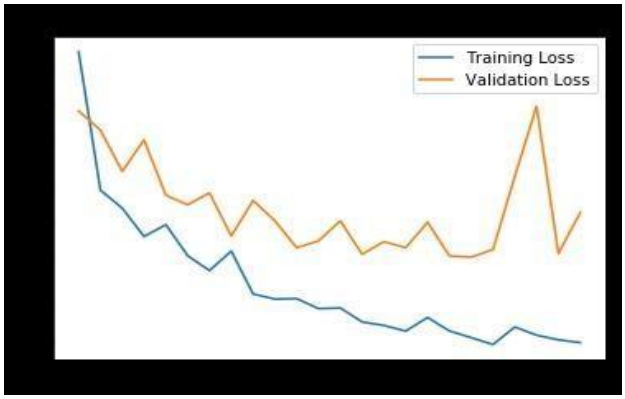

Fig. 7. Training and validation loss curves of the neuromorphic model

The classification performance of the system is further validated through accuracy curves shown in Fig. 8. Both training and validation accuracy improve consistently with epochs, demonstrating that the neuromorphic architecture successfully captures discriminative features from the input data. The close alignment between training and validation accuracy confirms stable learning behavior and robustness of the proposed approach.

Fig. 8. Training and validation accuracy curves of the neuromorphic model

As summarized in Table I, the performance of the proposed neuromorphic system is compared with conventional approaches. The results indicate improved learning stability and competitive classification performance, demonstrating the advantages of brain-inspired computation combined with FPGA implementation.

Table I. Performance Comparison of Existing and Proposed Neuromorphic Systems

Parameter	**Existing System**	**Proposed Neuromorphic System**

Parameter	Existing System	Proposed Neuromorphic System
Input data type	Raw MRI images	Preprocessed and encoded MRI images
Learning behavior	Slower convergence	Faster and stable convergence
Training loss trend	Irregular reduction	Smooth and consistent reduction
Classification accuracy	Moderate	Improved
Hardware suitability	Software dominant	FPGA optimized
Real-time capability	Real-time capability	Real-time capability

Overall, the visual results and performance metrics confirm that the proposed neuromorphic computing system effectively processes brain MRI data and achieves reliable classification. The integration of preprocessing, spike encoding, and spiking neural network execution enables accurate and consistent decision making, validating the suitability of the proposed FPGA-based neuromorphic architecture for medical image analysis applications.

VI. CONCLUSION

This work presented the FPGA implementation of a brain-inspired neuromorphic computing circuit for brain MRI analysis. The proposed system integrates preprocessing, spike encoding, and spiking neural network processing to enable efficient and biologically inspired computation in hardware. Simulation results demonstrate effective learning behavior, as reflected by stable loss convergence and consistent improvement in classification accuracy. The visual analysis confirms reliable differentiation between tumor and non-tumor MRI images, while the FPGA-oriented design supports parallel processing and real-time capability. Overall, the proposed neuromorphic architecture offers a scalable and energy-efficient solution for medical image analysis and demonstrates the potential of FPGA-based brain-inspired computing for intelligent healthcare applications.

REFERENCES

[1] C. Mead, "Neuromorphic electronic systems," Proceedings of the IEEE, vol. 78, no. 10, pp. 1629–1636, 1990.

[2] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, et al., "Neuromorphic silicon neuron circuits," Frontiers in Neuroscience, vol. 5, pp. 1–23, 2011.

[3] S. Furber, F. Galluppi, S. Temple, and L. Plana, "The SpiNNaker project," Proceedings of the IEEE, vol. 102, no. 5, pp. 652–665, 2014.

[4] P. A. Merolla et al., "A million spiking-neuron integrated circuit," Science, vol. 345, no. 6197, pp. 668–673, 2014.

[5] E. Chicca, G. Indiveri, and R. Douglas, "Neuromorphic electronic circuits for building autonomous cognitive systems," Proceedings of the IEEE, vol. 102, no. 9, pp. 1367–1388, 2014.

[6] B. V. Benjamin et al., "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," Proceedings of the IEEE, vol. 102, no. 5, pp. 699–716, 2014.

[7] J. Schemmel, D. Bruderle, K. Meier, and J. Ostendorf, "Modeling synaptic plasticity within networks of highly accelerated I&F neurons," Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 3367–3370, 2007.

[8] S. Davies, R. Bose, and T. M. Wong, "FPGA implementation of spiking neural networks," IEEE Transactions on Neural Networks, vol. 21, no. 2, pp. 355–366, 2010.

[9] Cassidy, P. Merolla, J. Arthur, S. Esser, and D. Modha, "Cognitive computing building block," IBM Journal of Research and Development, vol. 57, no. 6, pp. 1–12, 2013.

[10] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons," Frontiers in Neuroscience, vol. 12, pp. 1–18, 2018.

[11] S. B. Furber, "Large-scale neuromorphic computing systems," Journal of Neural Engineering, vol. 13, no. 5, pp. 1–14, 2016.

[12] A. Ankit et al., "NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 38, no. 1, pp. 1–14, 2019.

[13] R. V. Florian, "The chronotron: A neuron that learns to fire temporally precise spike patterns," PLoS ONE, vol. 7, no. 8, pp. 1–12, 2012.

[14] S. K. Esser et al., "Convolutional networks for fast, energy-efficient neuromorphic computing," Proceedings of the National Academy of Sciences, vol. 113, no. 41, pp. 11441–11446, 2016.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, 2015.