Advancing Analog Reservoir Computing through Temporal Attention and MLP Integration

1st Khalil Sedki
The Bradley Dept. of ECE
Virginia Tech
Blacksburg, USA
ksedki@vt.edu

2nd Yang Cindy Yi
The Bradley Dept. of ECE
Virginia Tech
Blacksburg, USA
yangyi8@vt.edu

2024 25th International Symposium on Quality Electronic Design (ISQED) | 979-8-3503-0927-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISQED60706.2024.10528762

***Abstract*—This paper presents a novel approach for Image classification, integrating analog Delay Feedback Reservoir (DFR), Temporal Attention Mechanism, Multi-Layer Perceptron (MLP), and backpropagation. The DFR system simplifies recurrent neural networks by focusing on the readout stage, offering enhanced performance and adaptability. The study details the design of an analog DFR system for low-power embedded applications, which utilizes a temporal encoder, Mackey-Glass nonlinear module, and dynamic delayed feedback loop to efficiently process sequential inputs with minimal power consumption. This system, implemented in standard GF 22nm CMOS FD-SOI technology, achieves high energy efficiency and a compact design area. It exhibits promise in emulating mammalian brain behavior, with only a remarkable 155μW power consumption and design area of 0.0044mm^2. In addition, this paper introduces a temporal attention mechanism that operates directly on continuous analog signals. The attention mechanism enhances the DFR system's ability to capture relevant temporal patterns. Furthermore, our approach incorporates the MLP for post-processing the DFR output. This comprehensive approach integrates DFR, Temporal Attention Mechanism and MLP via backpropagation, advancing the development of computationally-efficient Reservoir Computing (RC) systems for image classification with 98.96% accuracy.**

***Index Terms*—Delay-Feedback Reservoir (DFR), Mackey-Glass (MG) nonlinear function, temporal encoder, delay-feedback loop, Time to first spike encoding (TTFS), Interspike interval encoding (ISI), neuromorphic computing, attention mechanism, Multi-layer Perceptron (MLP), backpropagation.**

I. Introduction

A. Background and Motivation

The Modern computing architectures, based on the von Neumann paradigm, face inefficiencies in various applications such as speech recognition, sensor data processing, and time-series prediction [1]. The power consumption associated with data processing on supercomputers poses a significant challenge to global energy consumption. In contrast, the human brain exhibits remarkable cognitive abilities, such as learning, analyzing, and classifying vast amounts of information with a mere power consumption of 10 Watts [2]. This has led to the emergence of neuromorphic computing systems, which aim to break through the performance barriers of traditional von Neumann architectures by mimicking the functionality of mammalian brains.

B. Problem Statement

Liquid State Machines (LSMs), a specific type of recurrent neural network (RNN), closely emulate the functioning of biological nervous systems, displaying exceptional proficiency in processing temporal spiking information. However, training the recurrent connections in RNN can be computationally expensive. To address this, DFR systems have emerged as a novel machine learning concept, utilizing the dynamic behavior of RNN, as introduced by Jaeger [3] and Maass [4] in the early 2000s.

C. Research Objectives

The introduction of the high-performance reservoir computing (RC) system has been proposed. The DFR system employs a temporal encoder and a delay feedback loop to effectively process time-series input signals, utilizing feedback as a dynamic memory.

The motivation behind our work is to address specific challenges and achieve notable contributions. Firstly, we aim to improve power efficiency and reduce design area with the proposed DFR system. By doing so, we strive to overcome the limitations faced by conventional approaches and enhance the overall performance of DFR system. Secondly, we introduce a temporal attention mechanism that works with continuous analog signals, improving information processing efficiency and the system's ability to handle complex temporal patterns. Thirdly, we integrate a Multi-Layer Perceptron (MLP) with backpropagation to boost image recognition, enabling learning, prediction, and improved accuracy and reliability in image classification tasks.

D. Contribution of the Paper

• The novelty of our approach: lies in the incorporation of the temporal attention mechanism within our DFR system for image classification. To the best of our knowledge, this is the first time that an analog integrated circuit DFR computing system with a temporal attention mechanism has been implemented.

• Our DFR computing system: achieves of a notable power consumption of 155μW, showcasing a remarkable 25% improvement compared to the system in [11].

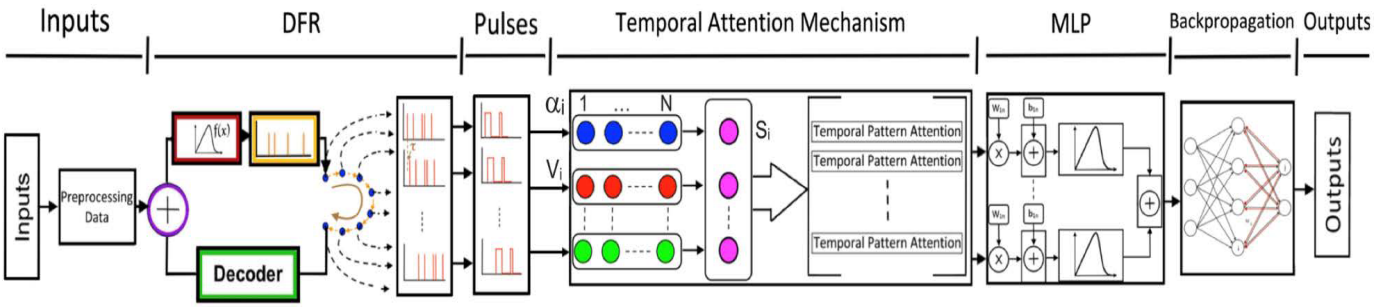

Fig. 1: DFR structure with temporal attention mechanism and MLP.

• Temporal attention mechanism: Allows the model to focus on relevant image features. It improves on-chip accuracy in image classification with average recognition rate of 98.96%.

II. Efficiency Enhancement of Delay-Feedback Reservoir Computing

DFR is a cutting-edge computing paradigm that utilizes neural networks to effectively process inputs that vary over time. The DFR system, depicted in Fig. 2, comprises two main components: the "reservoir," which is connected to the input, and the "readout function," responsible for analyzing reservoir states and generating the desired output.

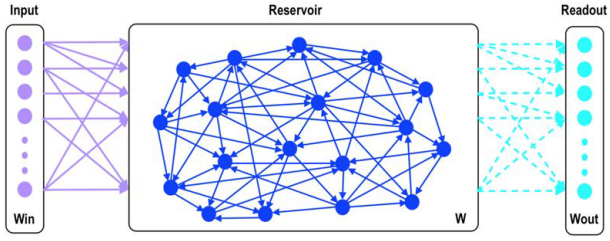

Fig. 2: conventional representation of reservoir computing based on RNN.

The reservoir, characterized by a fixed connectivity structure, does not require training. However, its neurons dynamically evolve with the temporal input signals. At a specific moment in time, denoted as t, the combined states of the reservoir neurons create the reservoir state x(t). By means of these dynamic evolutions, the reservoir non-linearly maps the input u(t) to a distinct space represented by x(t), allowing for a transformation of the input. Afterwards, the trained readout function examines the resultant reservoir states in order to generate the ultimate output y(t). One notable benefit of DFR is its lower training cost in comparison to traditional RNN methods. When training a DFR system, the main focus lies in modifying the connection weights (depicted as dashed arrows in the Fig. 1) between the reservoir and the output.

The Echo State Network (ESN) [3] and Liquid State Machine (LSM) [4] are widely used variants of DFR. ESN adopts a reservoir composed of artificial neurons that operate in discrete time, whereas LSM focuses on constructing biologically inspired learning models utilizing spiking neural networks (SNNs) with recurrent connections, resembling the configuration depicted in Fig. 1. LSM reservoir units typically incorporate both excitatory and inhibitory spiking neurons. Extensive research has demonstrated the universal approximation capability of these DFR systems.

III. Circuit Design of Delay-feedback reservoir Computing

The design of DFR represents a hardware implementation of the RC concept, aiming to harness the computational power and efficiency of RC in real-world applications [5][1]. This design leverages electronic components and circuits to create a physical DFR that can process and analyze complex temporal data. At its core, the design of DFR typically involves three main components: input nodes, a recurrent dynamic system, and output nodes [5]. The input nodes receive the time-varying input signals and transmit them to the recurrent dynamic system. This system, often implemented using analog or digital circuits, represents the DFR and consists of interconnected nodes that exhibit dynamic behavior [5]. The output nodes receive the processed information from the DFR and generate the desired output or perform further analysis. To implement our DFR, we utilize various electronic components and circuits including MG module, temporal encoder, decoder, and delay-feedback loop, as illustrated in Fig. 1.

A. Mackey-Glass Transfer Function

The MG nonlinear function, originally proposed by Mackey and Glass in their seminal work on physiological control systems [6], is a mathematical function that describes a dynamical system exhibiting chaotic behavior. It serves as a benchmark for studying the performance of time-delay systems and prediction models. The function is defined by the following equation:

$$\dot{x}(t) = \frac{\beta x(t-\tau)}{1 + x(t-\tau)^n} - \gamma x(t) \tag{1}$$

where $x(t)$ represents the system's state at time t, β controls the strength of the feedback, γ governs the dissipation rate, τ represents the time delay, and n determines the nonlinearity of the system. In the context of DFR, the MG nonlinear function is often used as the target or desired output for prediction tasks. The primary role of this function within DFR is to generate

complex temporal dynamics, which can be challenging to predict accurately. By feeding the time series generated by the MG function into the input layer of a reservoir, the reservoir's internal dynamics can learn to capture and exploit the temporal dependencies present in the data.

B. Neural Encoder

The Neural Encoder is an important component in the DFR that transforms input signals into appropriate representations for further processing. It plays a crucial role in capturing the relevant features of the input data and mapping them onto the reservoir. There are different schemes or methodologies for implementing the Neural Encoder, as depicted in Fig. 3, including rate-based encoding (Fig. 3(a)), time-to-first-spike (TTFS) encoding (Fig. 3(b)), and interspike interval (ISI) encoding (Fig. 3(c)):

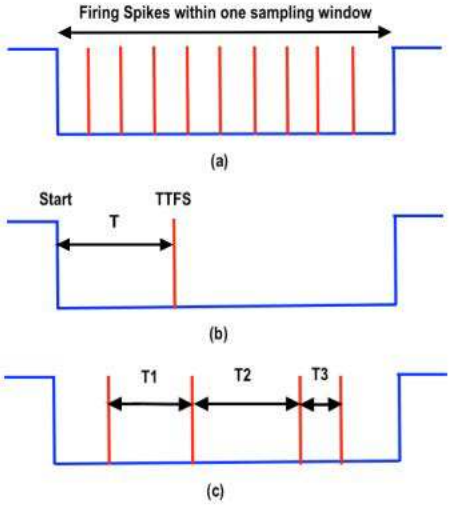

Fig. 3: Encoding schemes within one sampling window: (a) rate encoding, (b) Time to first spike encoding, (c) Interspike interval encoding.

1. Rate-Based Encoding: In this scheme, the information is encoded based on the firing rate of neurons. The input signal is typically represented by the average firing rate of a group of neurons over a given time interval. The higher the firing rate, the stronger the representation of the input signal [4][7].

2. Time-to-First-Spike (TTFS) Encoding: In TTFS encoding, the timing information of the first spike fired by a neuron is used to represent the input signal. The relative time at which the first spike occurs after the stimulus onset carries the encoded information [8][9].

3. Interspike Interval (ISI) Encoding: ISI encoding utilizes the time intervals between successive spikes of a neuron to represent the input signal. The pattern of the intervals can convey specific information about the input.

In DFR, the Neural Encoder acts as the interface between the input data and the DFR. Its role is to transform the input signals into a suitable format that can be effectively processed by the DFR. By converting the input data into a neural representation, the Neural Encoder enables the DFR to capture the relevant information contained within the input signals.

C. Delay-Neuron and Delay-Loop

In DFR architecture, the delay loop is a crucial component that contributes to the system's ability to process and capture temporal information. It is a feedback loop that introduces a time delay between the input and the output of the reservoir nodes. The delay loop works by feeding back the previous outputs of the reservoir nodes into the system after a certain time delay. This delayed feedback mechanism allows the system to retain and utilize past information when processing new input data. By incorporating the delayed feedback, the system can capture and exploit temporal dependencies, patterns, and dynamics present in the input signals.

Fig. 4 illustrates the implementation of a neuron within a feedback delay-loop, where a set of neurons is utilized to store and retrieve the previous outputs of the reservoir nodes.

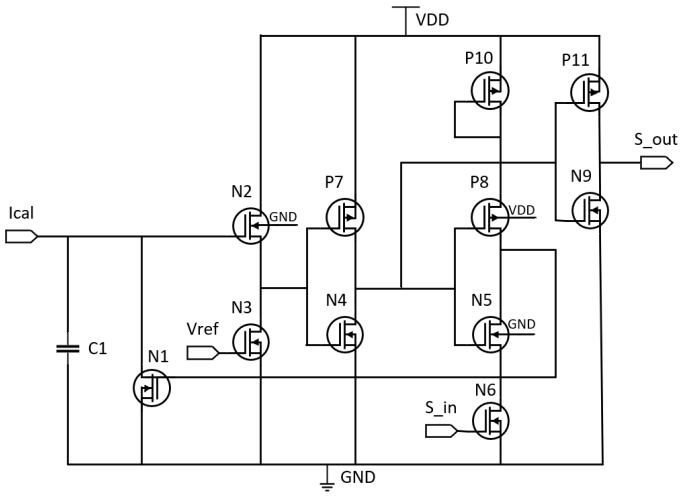

Fig. 4: Integrate-and-Fire (IF) neuron scheme.

The length of the time delay is typically adjustable and can be expressed as:

$$\tau = C_m . \frac{V_{th}(in)}{I_{cal}} \tag{2}$$

Where C_m is the membrane capacitor, $V_{th}(in)$ is the threshold voltage at the input of the delay neuron, and I_{cal} is the adjustable calibration current. The delayed feedback provided by the delay loop contributes to the reservoir's computational power and memory capacity. It enables the system to exhibit a rich temporal behavior and effectively handle time-dependent tasks such as time-series prediction [10], temporal pattern recognition, and signal processing.

One advantage of the delay loop in reservoir computing is that it allows for the separation of the input processing and the learning of the readout layer. The input data is processed by the reservoir nodes and transformed into a higher-dimensional space, while the readout layer, typically a linear model, can

be trained separately to map the reservoir states to the desired outputs. This separation of tasks simplifies the learning process and enhances the system's flexibility.

IV. Circuit Integration of Temporal Attention Mechanism and Multi-layer Perceptron

A. Temporal Attention Mechanism for Delay-Feedback Reservoir Enhancement

The temporal attention mechanism is an invaluable addition to the DFR system, especially when dealing with the nuances of continuous analog signals. Its components work cohesively to enhance the system's capacity to process complex temporal data, introduce non-linearity in attention weight calculations, and allows our model to focus on relevant image features and improve its accuracy. This represents a significant advancement in the capabilities of the DFR system. To the best of our knowledge, this is the first time of its application in this context, particularly in domains like MNIST digit classification and similar image recognition tasks. Its introduction is driven by the particular challenges and requirements presented by the DFR system in processing continuous analog signals. The key components of this innovative mechanism effectively tackle these challenges:

• Exponential Approximation Circuit depicted in Fig. 5: Designed to approximate the exponential function for each continuous analog signal V_i. This circuit generates values proportional to $\exp(approximated\, V_i)$, effectively translating the non-linearity of attention weight calculations into the analog domain. The significance of this component lies in its ability to capture the non-linear relationships within the analog signals, enabling more precise attention weight calculations.

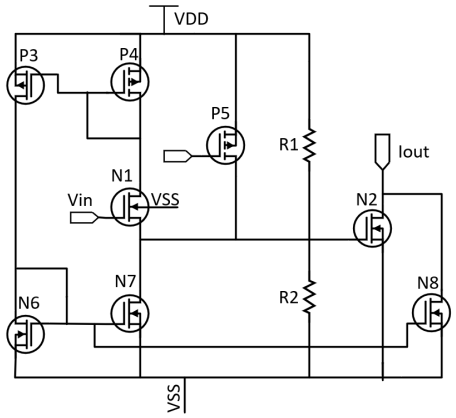

Fig. 5: Novel exponential V-I converter for our Attention mechanism.

• Attention Weight Calculation: Another crucial aspect of the temporal attention mechanism is the Normalization Circuit, which employs a dedicated circuit to normalize the $\exp(approximated\, V_i)$ values. It achieves this by dividing each $\exp(approximated\, V_i)$ value by the sum of all $\exp(approximated\, V_i)$ values across all time steps. This step is essential as it effectively normalizes attention weights based on analog signal values, ensuring that the system can adapt to varying levels of input significance. The modification of the traditional attention weight calculation process to operate in the analog domain is a significant breakthrough. Each continuous analog signal V_i now has its attention weight α_i calculated as:

$$\alpha_i = \frac{\exp(approximated\, V_i)}{\sum \exp(approximated\, V_i)} \tag{3}$$

This adaptation enables the system to perform attention-weighted operations directly on analog signals, enhancing its ability to focus on relevant information within the continuous data stream.

• Temporal integration: Incorporate attention into the DFR process. We multiply each continuous analog signal V_i by its corresponding normalized attention weight α_i and sum up these products, resulting in an integrated signal S_i:

$$S_i = \sum(\alpha_i.V_i) \tag{4}$$

The temporal integration not only optimizes the processing of input data but also facilitates the system's ability to handle complex temporal patterns efficiently.

B. Multi-Layer Perceptron for Image Recognition

To further enhance the capabilities of the DFR system, we introduce a MLP designed specifically for image classification tasks. Circuit Block1 (CB1), depicted in Fig. 6, plays a vital role in the forward propagation process, responsible for computing the activation function.

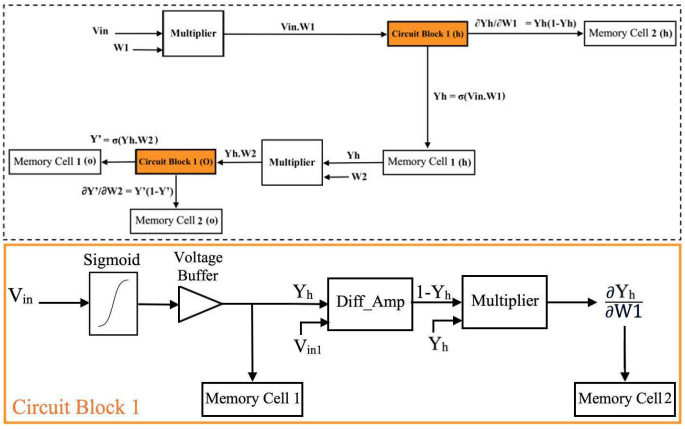

Fig. 6: Forward propagation architecture.

Within CB1, denoted as CB1 (h) for the hidden layer and CB1 (o) for the output layer, standard backpropagation procedures involving gradient descent are implemented. This includes the computation of both the sigmoid function and its derivative. The results generated by CB1 find their place in Memory Cell1 (h), which, in turn, serves as the input for the multiplier illustrated in Fig. 7, employed in the forward propagation system.

To ensure scalability and adaptability, the outputs of Memory Cell1 (h) can undergo a normalization process through

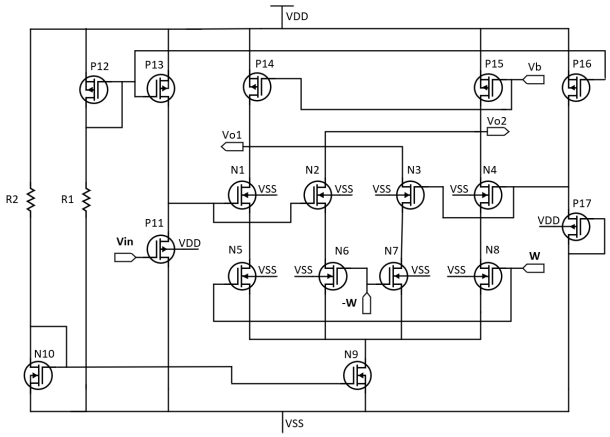

Fig. 7: Novel multiplier circuit for MLP.

a dedicated circuit. The voltages generated by the multiplier continue their path to a second CB1, ultimately concluding in the final output of the feedforward propagation, which is stored in Memory Cell1 (o). These final outputs rely on the activation function. Moreover, CB1 delivers not only the outputs of the activation function but also stores them in Memory Cell1 (h) and Memory Cell1 (o) for subsequent utilization in the backpropagation process. Additionally, CB1 retains the derivatives of the activation function, keeping them in Memory Cell2 (h) and Memory Cell2 (o). These derivatives prove to be invaluable during the backpropagation phase.

Following the completion of forward propagation, the system seamlessly shifts into the backpropagation phase, as depicted in Figs. 8 and 9. In this stage, the output of Circuit Block 2 (CB2) is stored within Memory Cell 3, aligning with the reverse direction of the signal propagation. The backpropagation process within the output layer is executed through the utilization of CB2 and Circuit Block 4 (CB4). Meanwhile, backpropagation through the hidden layer is effectively managed by Circuit Block 3 (CB3).

The final phase of the backpropagation algorithm involves weight updates. Memory Cells 4 and 5 store the update values before the update process begins. This weight update process involves applying a voltage signal with a specific amplitude to each multiplier. The amplitude of the update voltage signal is determined based on the desired weight adjustments, which are calculated from the gradient of the error. The precise amplitude of the update voltage signal is determined through the weight update circuit, taking into account the outputs of CB2 and CB4.

C. Backpropagation Training Process

The architecture of the forward propagation, as illustrated in Fig. 6, is as follows:

• Hidden layer: In the forward propagation step [12], the hidden layer computes activation using the sigmoid function σ depicted in Fig. 10 with input data X and weight matrix W_1:

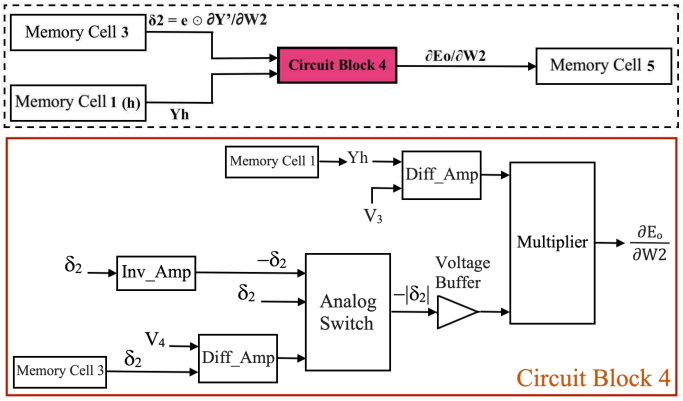

Fig. 8: Backpropagation to the output layer architecture.

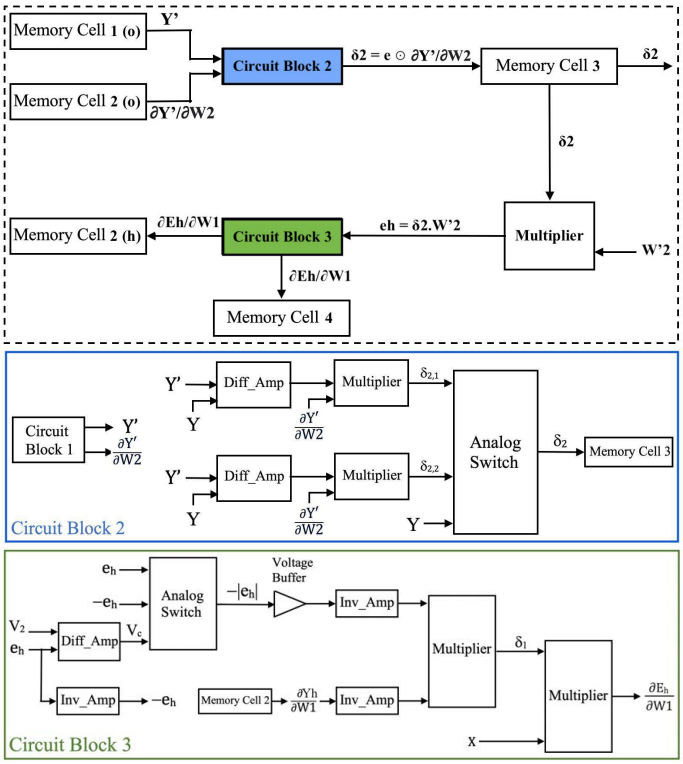

Fig. 9: Backpropagation to the hidden layer architecture.

$$Y_h = \sigma(X.W_1) \tag{5}$$

• Output layer: The output layer processes the hidden layer's activation Y_h to produce the final output Y' using the sigmoid function and a second weight matrix W_2:

$$Y' = \sigma(Y_h.W_2) \tag{6}$$

• Cost function: To guide the training process, we employ a cost function E that quantifies the difference between the target values Y_{target} and the actual predicted values Y_{actual}:

$$E = \frac{1}{2}\sum(Y_{target} - Y_{actual})^2 \tag{7}$$

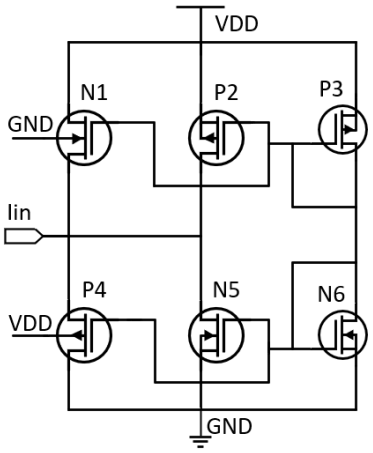

Fig. 10: Sigmoid activation circuit.

To optimize the MLP for image classification, we implement a backpropagation training algorithm that updates the network's weights. The training process involves several key steps:

• Backpropagation to the output layer depicted in Fig. 8: We calculate the derivative of the output layer error E_o with respect to the weight matrix W_2 [12]. This derivative, denoted as $\frac{\partial E_o}{\partial W_2}$, is computed as follows:

$$\frac{\partial E_o}{\partial W_2} = Y_h.\delta_2 \tag{8}$$

where δ_2 represents the rate of change of the error with respect to the weight W_2. The error for the output layer e is determined as the difference between the expected neural network output Y and the actual output Y': e = Y - Y'. Additionally, we utilize the derivative of the sigmoid function for this layer [12]:

$$\frac{\partial Y'}{\partial W_2} = Y'(1 - Y'). \tag{9}$$

• Backpropagation to the hidden layer shown in Fig. 9: We calculate the derivative of the error for the hidden layer E_h with respect to the weight matrix W_1. This derivative, denoted as $\frac{\partial E_h}{\partial W_1}$, is computed as follows:

$$\frac{\partial E_h}{\partial W_1} = X'.\delta_1 \tag{10}$$

where X' is an inverted input matrix, and δ_1 is the error of the hidden layer. The error δ_1 is calculated by propagating back δ_2 as follows:

$$\delta_1 = \delta_2.W_2' \tag{11}$$

The derivative of the hidden layer output E_h with respect to W_1 is the same as that used for the output layer [12]:

$$\frac{\partial Y_h}{\partial W_1} = Y_h(1 - Y_h) \tag{12}$$

• Weight updating process: Finally, we update the weight matrices (W_1 and W_2) using the calculated changes in weights:

$$\Delta W_1 = \frac{\partial E_h}{\partial W_1}.\eta \tag{13}$$

$$\Delta W_2 = \frac{\partial E_o}{\partial W_2}.\eta \tag{14}$$

where η represents the learning rate responsible for controlling the speed of convergence.

• The weight matrices are updated accordingly:

$$(W_1)_{(new)} = W_1 + \Delta W_1 \tag{15}$$

$$(W_2)_{(new)} = W_2 + \Delta W_2 \tag{16}$$

Fig. 11 provides circuit-level realizations of components found in the primary backpropagation modules. In Fig. 11(a), we present the circuit implementation of the analog switch circuit. Fig. 11(b) showcases the realization of the inverting amplifier. Finally, Fig. 11(c) provides an illustration of the difference amplifier circuit.

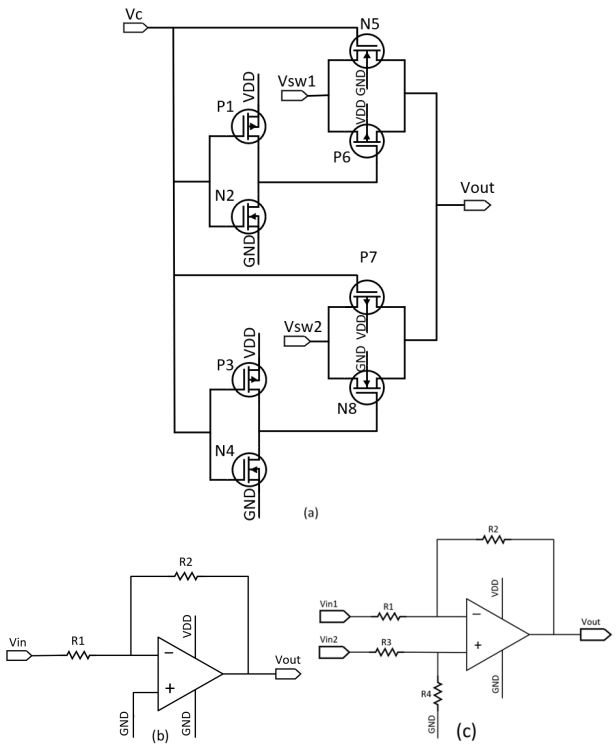

Fig. 11: Circuit components used in Circuit Blocks: (a) Analog switch. (b) Inverting amplifier. (c) Difference amplifier.

The integration of the temporal attention mechanism and the MLP via backpropagation into the DFR system equips it with the ability to process complex temporal patterns and perform image recognition efficiently. This synergy enhances the DFR's performance in handling dynamic and real-world data.

V. Results and Discussions

Our DFR system is implemented using Global Foundries 22nm CMOS FD-SOI Technology. In this section, we analyze and evaluate the experimental performance of our system and its constituent components.

A. Analysis of Mackey-Glass Module

In Fig. 12, the successful attainment of a nonlinear correlation between input and output signals in the Mackey-Glass function is demonstrated through simulation.

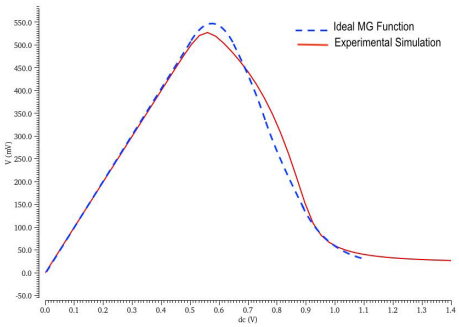

Fig. 12: Simulation of Mackey-Glass nonlinear Function.

The transfer function of the circuit, which resembles the nonlinearity of the ideal Mackey-Glass function, can be adjusted by controlling the parameters of the NMOS transistors responsible for shaping the Mackey-Glass node. Moreover, Fig. 13 illustrates the layout of the Mackey-Glass function, providing insight into the physical implementation. By referring to (1), it becomes evident that an increase in the value of n corresponds to an increased nonlinearity in the transfer function.

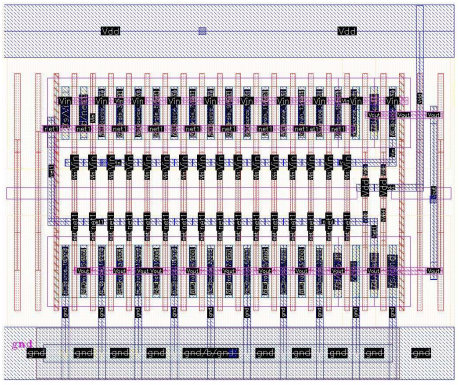

Fig. 13: The Layout of Mackey-Glass nonlinear Function.

B. Analysis of Delay-Loop

The delay loop consisting of multiple stages, where the output spike trains are generated, as depicted in Fig. 14. The voltage threshold V_{th} was set at 260 mV, and the current I_{cal} was fixed at 0.5 μA, equivalent to a resistance of 520 KΩ. As a result, our delay unit can achieve substantial delay times using a very small capacitor. By adjusting the delay constant with low capacitance and resistance values, the system's dynamics can be tuned from an ordered state to the edge of chaos.

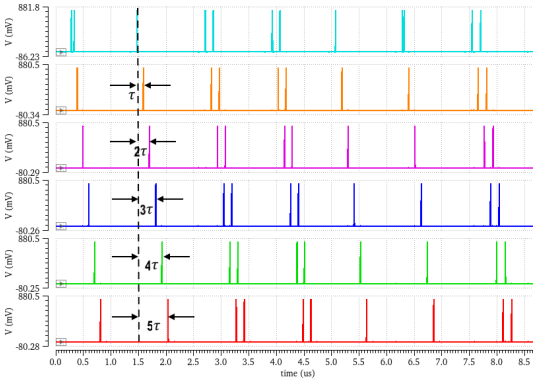

Fig. 14: Output spike trains of the delay-loop.

C. Analysis of the Delay-Feedback Reservoir Computing System

The dynamic behavior of our DFR system can range from periodic to chaotic by fine-tuning the total delay time in the delay loop. As mentioned in the previous section, Fig. 14 shows the output spike trains of the first six delay neurons in the delay loop. Experimental findings reveal that the delay time between each delay neuron is nearly the same.

The layout of our DFR system, illustrated in Fig. 15, is implemented using the Global Foundries 22nm CMOS FD-SOI Technology. It occupies a design area of 63.377μm×69.294μm. The design specification of our system and a comparison with the other state-of-the-art neuromorphic systems are summarized in Table I.

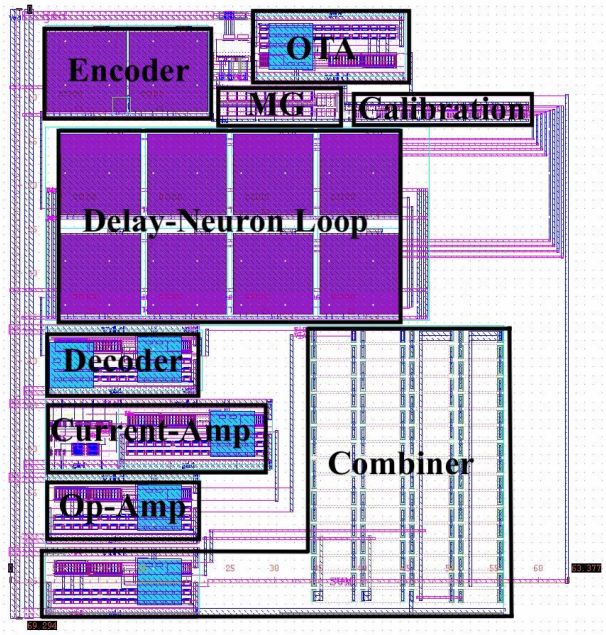

Fig. 15: The layout of the DFR system.

TABLE I: Design Specifications and Performance Comparison with the State-of-the-Art Neuromorphic Systems

	[11]	*[13]*	**[14]**	**This work**
Design Architecture	DFR	Deep-DFR	ISI Encoding	DFR
Implementation	Analog IC Design			
Technology	130 nm		180nm	22nm
Supply Voltage	1.2V		1.8V	0.8V
Frequency	20MHz	1 MHz	—	250KHz
Activation Function	Mackey-Glass		—	Mackey-Glass
Neuron Type	IF	LIF		IF
Design Area	0.0098mm^2	—	—	0.0044mm^2
Power Consumption	206μW	526μW	2.9mW	155μW
Algorithm	—	FCNet	SNN	Attention+MLP
Dataset	—	MNIST		
Accuracy	—	98.7%	90%	98.96%

D. Application evaluation

In this experiment, we evaluated the performance of our DFR in a well-known image classification task using the MNIST dataset. To ensure efficient training, we initially normalized the image dataset in the training phase. We assessed the recognition rate by varying the number of neurons within hidden layers and training epochs in our neural network model, which includes the proposed DFR, pattern attention mechanism, and MLP, utilizing the backpropagation training process.

For on-chip training, we utilized images depicting digits 0 to 9 as inputs for our system, featuring a three-layer neural network tailored for image processing. Employing a compact neural network with an output layer represented by two CB1 (o), our model is trained to identify digits 0 to 9. In each clock cycle, a row of pixels is processed, undergoing transformation through five multipliers to acquire the necessary trained weights. The input signals undergo preprocessing and interacting with the weighted input multiplier, introducing variability to the signal. Applying voltage signals from the input stage to the multiplier, the input signals undergo multiplication with the weight values stored in Memory Cells 4 and 5, resulting in output voltages. The output of the multipliers is designed for classifying nine digits. Operating at a clock frequency of 250KHz, each image is processed within one clock cycle (4μs per image) with a supply voltage of 0.8V. The average recognition rate achieved by our DFR consistently stands at 98.96%, exceeding the accuracy of [13] and outperforming [14]. This demonstrates a classification improvement of 8.96% compared to [14], as shown in Table I.

VI. Conclusion

This paper presents a cutting-edge analog delay-feedback reservoir (DFR) system designed for low-power embedded applications. It incorporates various components such as a temporal encoder, Mackey-Glass nonlinear module, and delay-feedback loop. These components work together to efficiently process sequential inputs with outstanding energy efficiency of only 155 μW, ensuring a compact and energy-efficient circuit design using standard GF 22nm CMOS FD-SOI technology. A unique temporal attention mechanism tailored for continuous analog signals enhances system performance, and the integration of an MLP with backpropagation training further improves the system's capabilities, especially in image classification. In MNIST image classification, our DFR achieves an impressive 98.96% accuracy, a significant improvement of 8.96% compared to [14].

Acknowledgment

This work was supported in part by the U.S. National Science Foundation (NSF) under Grant CCF-1750450, Grant ECCS-1731928, Grant ECCS-2128594, Grant ECCS-2314813, and Grant CCF-1937487.

References

[1] Verstraeten, D., Schrauwen, B., D'Haene, M., & Stroobandt, D. (2007). An experimental unification of reservoir computing methods. Neural Networks, 20(3), 391-403.

[2] Yu, S., Wu, Y., Jeyasingh, R., Kuzum, D., & Wong, H. S. P. (2011). An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. IEEE Transactions on Electron Devices, 58(8), 2729-2737.

[3] Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report, 148(34), 13.

[4] Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural computation, 14(11), 2531-2560.

[5] Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. Computer science review, 3(3), 127-149.

[6] Mackey, M. C., & Glass, L. (1977). Oscillation and chaos in physiological control systems. Science, 197(4300), 287-289.

[7] Buesing, L., Bill, J., Nessler, B., & Maass, W. (2011). Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. PLoS computational biology, 7(11), e1002211.

[8] Gerstner, W., & Kistler, W. M. (2002). Spiking neuron models: Single neurons, populations, plasticity. Cambridge University Press.

[9] Thorpe, S., Delorme, A., & Van Rullen, R. (2001). Spike-based strategies for rapid processing. Neural Networks, 14(6-7), 715-725.

[10] Wierstra, D., Gomez, F. J., & Schmidhuber, J. (2005, June). Modeling systems with internal state using evolino. In Proceedings of the 7th annual conference on Genetic and evolutionary computation (pp. 1795-1802).

[11] Bai, K., & Bradley, Y. Y. (2018, March). A path to energy-efficient spiking delayed feedback reservoir computing system for brain-inspired neuromorphic processors. In 2018 19th International Symposium on Quality Electronic Design (ISQED) (pp. 322-328). IEEE.

[12] Krestinskaya, Olga, Khaled Nabil Salama, and Alex Pappachen James. "Learning in memristive neural network architectures using analog backpropagation circuits." IEEE Transactions on Circuits and Systems I: Regular Papers 66.2 (2018): 719-732.

[13] Bai, Kangjun, Qiyuan An, and Yang Yi. "Deep-DFR: A memristive deep delayed feedback reservoir computing system with hybrid neural network topology." Proceedings of the 56th Annual Design Automation Conference 2019. 2019.

[14] Nowshin, Fabiha, and Yang Yi. "Memristor-based deep spiking neural network with a computing-in-memory architecture." 2022 23rd International Symposium on Quality Electronic Design (ISQED). IEEE, 2022.