AFRL-RI-RS-TR-2020-103

ENABLING BRAIN-INSPIRED PROCESSORS THROUGH ENERGY-EFFICIENT DELAYED FEEDBACK RESERVOIR COMPUTING INTEGRATED CIRCUITS

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

JUNE 2020

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2020-103 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /	**/ S /**
CLARE D. THIEM	GREGORY J. HADYNSKI
Work Unit Manager	Assistant Technical Advisor
	Computing & Communications Division
	Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government’s approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*
JUNE 2020

2. REPORT TYPE
FINAL TECHNICAL REPORT

3. DATES COVERED *(From - To)*
JUN 2018 – DEC 2019

4. TITLE AND SUBTITLE

ENABLING BRAIN-INSPIRED PROCESSORS THROUGH ENERGY-EFFICIENT DELAYED FEEDBACK RESERVOIR COMPUTING INTEGRATED CIRCUITS

5a. CONTRACT NUMBER
FA8750-18-2-0009

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62788F

6. AUTHOR(S)

Yang Yi

5d. PROJECT NUMBER
T2NH

5e. TASK NUMBER
VA

5f. WORK UNIT NUMBER
ST

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Virginia Polytechnic Institute and State University
300 Turner St., NW, Suite 4200
Blacksburg VA 24061-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RITB
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER

AFRL-RI-RS-TR-2020-103

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Reservoir computing (RC), an emerging machine learning paradigm, is considered a simplification of a conventional recurrent neural network. RC offers a unique learning mechanism at the readout stage that accelerates learning and computing operations. The objective of this project was to build a new class of computationally-efficient delayed feedback reservoir systems. The final outcome of this project was a Delayed Feedback Reservoir processor designed to exploit recent advancements in machine learning, integrated circuits, and nanotechnology.

15. SUBJECT TERMS

Reservoir computing, machine learning, recurrent neural network, nanotechnology

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** **CLARE D. THIEM**
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	83	**19b. TELEPHONE NUMBER (*Include area code*)** **N/A**

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1.0 SUMMARY

Reservoir computing, an emerging machine learning paradigm, is considered a simplification of conventional recurrent neural network (RNN), offering a unique learning mechanism only at the readout stage to accelerate learning and computing operations. In general, the role of the reservoir layer is to nonlinearly transform sequential inputs onto a high-dimensional space, such that features of inputs can be efficiently read out by a simple learning algorithm. As such, any nonlinear dynamical systems can be used as reservoirs. The objective of this project was to build a new class of computationally-efficient delayed feedback reservoir (DFR) systems. The research takes on a cross-layer approach to develop a circuit- and architectural-level design of the DFR system. A comprehensive investigation of nonlinear transfer functions, delay systems, chaotic systems, and temporal decoder were conducted; moreover, a DFR system was designed and simulated with the low-power complementary metal-oxide-semiconductor (CMOS) technology. The circuitry was laid out in the Cadence Virtuoso Platform while statistical data on circuits' performance was captured and examined. The research deliverables include the electronic circuit design, Simulation Program with Integrated Circuit Emphasis (SPICE) circuit models, layout, simulation results, and measurement data for the DFR system. The final outcome of this project was a DFR processor designed to exploit recent advancements in machine learning, integrated circuits, and nanotechnology.

2.0 INTRODUCTION

The rapid evolvement of computing systems was perfectly predicted by Moore's law in the past several decades. However, it has been observed that the rate of enhancement is starting to saturate and slow down, indicating as the end of Moore's prediction due to fundamental physical limits of the complementary metal-oxide-semiconductor (CMOS) process [1]. Inspired by the biology and proposed by Dr. Carver Mead in the 1980s, the neuromorphic computing has matured to provided intelligent systems that able to imitate natural neuro-biological processes through highly parallelized computing architectures; such systems typically model the function of neural networks through very-large-scaled-integrated (VLSI) circuits [2].

Two well-known artificial neural networks (ANNs), feedforward neural networks (FNNs) and recurrent neural networks (RNNs) are powerful ANNs that are capable of learning. Learning is not probable in classic computing systems reliant on preprogrammed instructions. RNNs, which are constructed with random recurrent connections, closely mimic feedback operations in neurological systems and have the capability to process temporal information. Although RNNs are more powerful in performing temporal tasks than FNNs, the training complexity of recurrent connections are, however, computationally expensive.

In recent years, reservoir computing [3, 4], as shown in Figure 1, has emerged exploiting the dynamic behavior of conventional RNNs while drastically reduces its computational cost of learning. The main characteristic of the reservoir computing is that input weights, W_{in}, and weights of recurrent connections, W_{res}, within the reservoir layer are fixed at all-times whereas only readout weights, W_{out}, are trained with a simple learning algorithm. In general, the role of the reservoir layer is to nonlinearly transform sequential inputs onto a high-dimensional space, such that features of inputs can be efficiently read out by a simple learning algorithm through output weighted elements. As such, any nonlinear dynamical systems can be used as reservoirs.

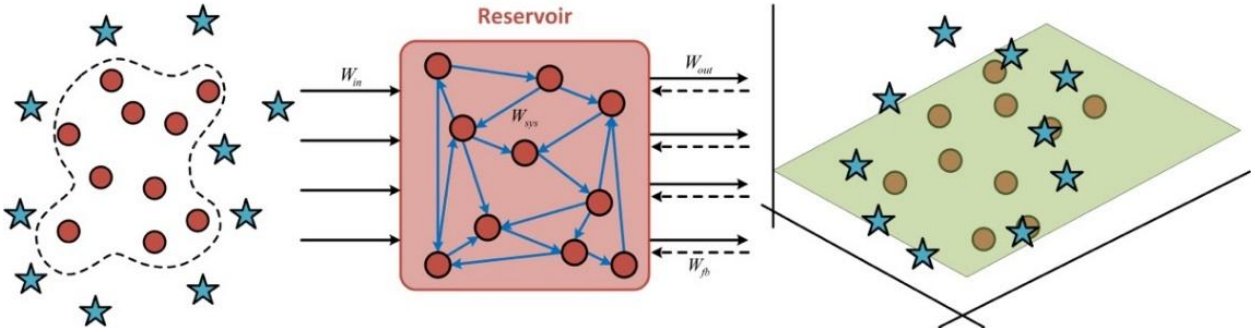


Figure 1. General architecture of reservoir computing.

Two well-studied representations of reservoir computing models, the echo state network (ESN) [3] and the liquid state machine (LSM) [4], employ the strength of conventional RNNs without the need for synaptic connections within the reservoir layer to be trained. The major difference that sets these two models of reservoir computing apart is the format of the signal. In LSM, spiking signals are used as pre- and post-neural signals, while analog signals are examined in ESN. The general node state, $s(t)$, at the current time step of the reservoir layer can be expressed as

$$s(t) = f\big(u(t) \cdot W_{in} + s(t-1) \cdot W_{res} + y(t-1) \cdot W_{fb}\big), \tag{1}$$

where $f(\quad)$ is the nonlinear activation function; $u(t)$ is the input at the current time step; $s(t-1)$ and $y(t-1)$ are the internal state and output state of the network, respectively, at the previous time step; W_{in}, W_{res}, and W_{fb} denote input weights, internal weights within the reservoir layer, and feedback weights from the output to the reservoir layer, respectively. The output state of the network can be then expressed as

$$y(t) = s(t) \cdot W_{out}, \tag{2}$$

where W_{out} is output weights. These neuron-like nodes within the reservoir layer achieved functionalities of (a) high dimensional projection and (b) fading memory, similar to the biological neuron's behavior. These advantages make the reservoir computing especially suitable for neuromorphic computing paradigms.

The straightforward hardware realization of neural networks usually consumes a large volume of memory and computing resources, as well as requires high design complexity and hardware cost. For example, a text-recognition software is typically designed to run on a high-performance computing (HPC) cluster, consisting of ~70,000 processor cores that provides a massive peak computing power of 500 trillion floating-point operations per second (FLOPS) [5]. Algorithm enhancement and conventional hardware implementation can mitigate the computational cost issue to some degree, but not fundamentally resolve it. It is therefore essential to design a new class of hardware optimized for conducting the crucial operations of neuromorphic computing, instead of relying on system implementations built upon traditional computer structures.

In general, the hardware implementation of reservoir computing can be either digital or analog. The current technology mainly focuses on digital implementation. The reservoir computing based on digital implementation offers high computational precision, high reliability, and high programmability [6-10]. Synaptic weights can be stored on- or off-chip. However, disadvantages of digital implementation are that it requires a relatively large circuit size and consumes higher power compared to analog [11-15]. By contrast, analog implementation takes advantage of electronic and physical laws to implement basic functions. For instance, operational amplifiers can perform neuron-like functions, such as sigmoid transfer. Likewise, a temporal integration can be achieved through capacitive integration and spatial summation through Kirchhoff's law. In analog, the computationally intensive calculations are automatically performed by physical processes, such as summing of currents or charges. Furthermore, the analog implementation of reservoir computing systems offers significantly higher speed, less design area, and less energy dissipation than digital design.

Compared to the current technology, this project sought to build a brain-like computing system with analog integrated circuits (IC), which could offer potentially disruptive capabilities in real-time signature analysis, time series predictions, and environmental perception for autonomous operations and dynamic control systems. This project included circuit design, analysis, fabrication, and testing of the dynamic delayed feedback reservoir (DFR) system that adopts sensory encoding and processing methodologies employed in biological brains. The resulting dynamic time-series data was processed using reservoir computing processors. Current predictive control strategies typically take the form of black-box systems. Such systems are based on process models built from physical concepts and data-driven simulations that cannot cope with problems that have strong temporal aspects. By contrast, control systems built on the nonlinear dynamics of reservoir computing are capable of addressing these issues and have the potential to form the foundation for a new generation of deterministic adaptive processors. The underlying inspiration for reservoir computing is the insight that the brain processes information by generating patterns of transient neuronal activity excited by input sensory information.

This effort's research on nanotechnology-based neuromorphic computing system design could impact the society of HPC, energy-efficient computing, information technology, and nanotechnology. The project focused on the analog implementation of reservoir computing. The implementation of real-time neuromorphic systems is ideal for pattern and signature recognition in mobile platforms with severe Size, Weight, and Power (SWAP) constraints. As a practical matter, such resource restrictions rule out traditional software approaches, which often require high-performance processing or run too slowly due to the inherent serial nature of von Neumann architectures.

The research project holds great promise for many important engineering and scientific applications. Systems that exploit a type of non-traditional architecture that encompasses evolutionary systems hold great promise for leveraging these behaviors to address specific classes of mission-critical problems that have not been solved by the current state-of-the-art CMOS digital computing.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

This overall project was divided into four interconnected research thrusts: (1) Mackey-Glass nonlinear electronic design with chaotic characteristic; (2) delay-feedback loop design with an Integrate-and-Fire (I&F) neuron; (3) short-term memory achievement with temporal decoder; (4) delayed feedback reservoir system integration and optimization.

Task 1.1 Comprehensive Investigation of Mackey-Glass Nonlinear Equation

The biological neural system within mammal brains has highly-nonlinear recurrent connections with chaotic characteristics. In the hardware realization, sigmoid and hyperbolic tangent functions are commonly used to emulate the nonlinear behavior. However, both sigmoid and hyperbolic tangent functions do not have the chaotic property to mimic the biological behavior as in the neural system. On the other hand, the Mackey-Glass equation, a delay differential equation (DDE), falls into the category of chaotic system.

In this task, we comprehensively investigated three types of Mackey-Glass circuit models. Design aspects such as nonlinearity, energy consumption, and design area were examined. Based on this comprehensive study of these three circuit models, we developed novel and fundamental methodologies to represent the nonlinear information based on neuronal activity.

Task 1.2 Design and Optimization of Mackey-Glass Nonlinear Electronic Circuit with Low-power CMOS Technology

In this project, we first designed and optimized nonlinear circuitry by utilizing the working mechanism of single-ended charge pump (CP) using the standard 180nm CMOS technology. The successful design was laid out through the Cadence Virtuoso platform. Design characteristics, including energy consumption, design area, robustness to the CMOS process and operating temperature variability were documented.

Task 1.3 Design of dynamic Signal Conversion Circuit with Trans-Impedance Amplifier and Embedded Loop Filter

Most electronic circuits use voltage as a trigging reference signal; however, to enable the function of spiking information processing through our previously introduced temporal encoder [16], an analog current signal was required to trigger the spiking neuron within the temporal encoder. As such, the analog voltage signal needs to be converted into current by the trans-conductance amplifier. It should be noted that analog circuit design also has, as one of its key aspects, the capability conduct noise analysis.

In this task, we first designed and optimized a trans-conductance amplifier with an embedded loop filter. The circuit was then integrated with the CP-based Mackey-Glass nonlinear electronic circuitry. Design characteristics, including energy consumption, design area, robustness to CMOS process and operating temperature were documented.

Task 2.1 Comprehensive Investigation of Delay Elements

In this task, we comprehensively investigated three types of delay elements. Design aspects such as energy consumption, design area, and system robustness were examined. Based on the comprehensive study of various delay elements, we developed a practical approach to represent the delay-feedback loop based on the spiking neuronal activity.

Task 2.2 Design and Optimization of Delay-feedback Loop with Integrate-and-fire Neurons

Analog delay line design is essential for building computing-processor based dynamic reservoirs. Charge-coupled device and switched-capacitor solutions are well known. A drawback of these discrete-time circuits is the necessity of clocking and the occurrence of aliasing effects. A continuous-time approach may, therefore, be attractive, particularly if the delay per section can be controlled electronically.

In this task, we designed a delay-line section with two important properties: (a) a modulus of the transfer function equal to unity over a broad frequency range, and (b) phase shift that depends linearly on frequency to provide a frequency-independent group delay.

Task 2.3 Robustness Enhancement for Static Delay Loop Design

The delay plays an important role in the reservoir layer, which determines the dynamic behavior of the system. The robustness of the I&F delay neuron is proportional to transistors' channel length in the input stage of the firing threshold. Increase the channel length of transistors significantly improve the system robustness, which often results in a larger design area.

In this task, the system robustness in terms of CMOS process and operating temperature variation are preliminarily studied and analyzed through Monte-Carlo simulations in the Cadence Virtuoso platform. Results were evaluated and used as a reference to enhance the robustness of the system without dramatically increase its design area.

Task 3.1 Comprehensive Investigation of Analog and Digital Signal Integrators

In this task, we comprehensively worked on different hardware implementation schemes on the temporal code to digitize the pulse decoder and signal integrator. For both analog and digital design schemes, we evaluated the performance in terms of implementation complexity, energy consumption, and computational accuracy.

Task 3.2 Design and Analysis of the Temporal Code to Digitized Pulse Decoder

Unlike the rate encoding scheme, the information of the temporal code is encoded into the time interval between spikes; in other words, the total number of spike is fixed, and time intervals between spikes alter varied based on the given input data. Therefore, two different temporal codes cannot be merged directly. This project tentatively utilizes the spike-timing-dependent plasticity (STDP) methodology to design the decoder for extracting timing information from a temporal spike train.

Task 3.3 Design and Optimization of Signal Integrator with Both Analog and Digital Model

Both analog and digital implementations of the signal integrator were designed, and their performances evaluated in this task. This resulted in the evaluation of complexity, energy consumption, and computational accuracy were evaluated. Based on these evaluation results, we examined the best available design scheme for our application.

Task 4.1 Delayed Feedback Reservoir System Design and Integration

In this task, our DFR system together with the inter-spike-interval (ISI) temporal encoder, Mackey-Glass nonlinear node, and static delay-feedback loop were implemented and optimized through the Cadence Virtuoso Platform with the standard 180nm CMOS technology.

Task 4.2: Circuit Fabrication and Testing with Advanced CMOS Technology

The optimized circuit and generate the layout of our designed DFR system was completed under this task. We fabricated our chip using the standard Global Foundries (GF) 180nm CMOS technology. The fabricated chip was then tested at VT's Multifunctional Integrated Circuit and System (MICS) Laboratory using its state-of-the-art specialized lab facilities for integrated circuit measurement.

Task 4.3: Exploration of Multi-layer DFR by Utilizing the Introduced DFR System

The field of deep learning has attracted worldwide attention due to its hierarchical architecture that allows more efficient performance than a shallow structure, not only on accuracy but also on the processing speed. This superior performance is a result of its intrinsic deep structure. Deep neural networks (DNNs) were constructed by multiple layers working in the form of a processing pipeline. Deep learning architecture has been proven to have the exceptional performance in high-dimensional data that is applicable to many fields, ranging from business to science. Many performance records are broken by deep learning architecture in the application of image recognition, handwritten recognition. The depth is generally defined as stacking multiple hidden layers in between the input and output layers that could either be defined in time or space.

In the reservoir computing, recurrent connections are adapted as in the reservoir, so-called the hidden layer. Hence, traditional reservoir computing systems rely on a depth-in-time computing structure. For DFR systems, the depth-in-time arises from the delayed signal that combines with the new input. However, for both traditional reservoir computing systems and DFR systems, a single reservoir layer does not create the depth-in-space computing structure. Similar to stacked FNNs in the deep learning field, the depth-in-space could also be achieved by stacking multiple reservoir layers between input and output layers. In this task, we explored the possibility of merging the deep learning and our introduced DFR system.

Task 4.4: Proof of Concept on the Development of Hybrid Photonic + ASIC Platform

Both photonic and analog IC implementations of reservoir processors have advantages and disadvantages. The photonic implementation offers high-speed optical processing and high bandwidth, but it requires a large design size and results in high power consumption. Moreover, the photonic implementation of reservoir processors requires expensive peripheral devices such as the digitizer, the waveform generator, and the mach-zehnder modulator, which are difficult to scale. Analog IC implementation, on the other hand, offers compact design size and low power dissipation, but is susceptible to noise, which makes it difficult to design. When this report was written, analog IC implementations of reservoir processors have not yet been reported in the literature. In this task, we focused on the proof of concept development of a hybrid photonic and ASIC platform. We also explored the possibility of combining the advantages of both photonic and IC implementations of reservoir computing.

3.1 Mackey-Glass Nonlinear Node Design with Chaotic Behavior

3.1.1 Comprehensive investigation of Mackey-Glass nonlinear equation

The input and output relationships are mathematically described by transfer functions. The step function was one of the earliest transfer functions to express the input-output correlation [17]. However, for the reservoir computing, a nonlinear mapping of input is required. Hence, in order to achieve such functionality, nonlinear transfer functions are employed. Sigmoid and hyperbolic tangent functions are the most commonly used nonlinear equation that have been adopted as the activation functions. The slope of the sigmoid function can be tuned by varying its coefficients. As the slope becomes steeper, the shape of the sigmoid function becomes more like the step function. However, different from the step function, the sigmoid function is a continuous function ranging from 0 to 1. This function has a mix of linear and nonlinear behavior. Another nonlinear function that can be used is the hyperbolic tangent function. The hyperbolic tangent function, antisymmetric with respect to the origin, converges faster compared to the sigmoid function in neuron network designs.

In general, sigmoid and hyperbolic tangent functions are not the only nonlinear functions that can be utilized as the activation function. Different nonlinear functions that could serve as the activation function for the reservoir computing were explored. Initially designed to deal with diseases that exhibit symptoms with oscillatory instabilities, the Mackey-Glass function was found to be a potential candidate to serve as a transfer function for delay-feedback systems. Described by a DDE, the Mackey-Glass function falls into the category of delayed systems. Dynamics of the Mackey-Glass function depends on both current and previous states. The Mackey-Glass function is mathematically expressed in the following form

$$\frac{dP}{dt} = \frac{\alpha \cdot P_\tau}{1 + \beta \cdot p_\tau^n} - P, \tag{3}$$

where α and β are arbitrary design parameters, n is the nonlinearity coefficient, P denotes the input at the current time step, and $P_\tau = P(t - \tau)$ represents the information from the previous state of the system. By varying the nonlinearity coefficient, the nonlinearity of the function can be changed accordingly. As plotted in Figure 2, the shape of the Mackey-Glass equation changes with the increasing nonlinearity coefficient. This property of the Mackey-Glass equation enables discovering the optimal regime.

The dynamic of a system can be transformed from stable to chaotic regimes by tuning the delay, which can be either stabilizing or destabilizing. It was found that the best computational performance occurs in the transition region from stable to chaotic regime, which is called the edge of chaos [18-20]. Hence, with delay embedded in the system, Mackey-Glass equation, as plotted in Figure 2, possesses the potential of operating at the edge of chaos region. Compared to the hyperbolic tangent function, the Mackey-Glass equation exhibits higher nonlinearity.

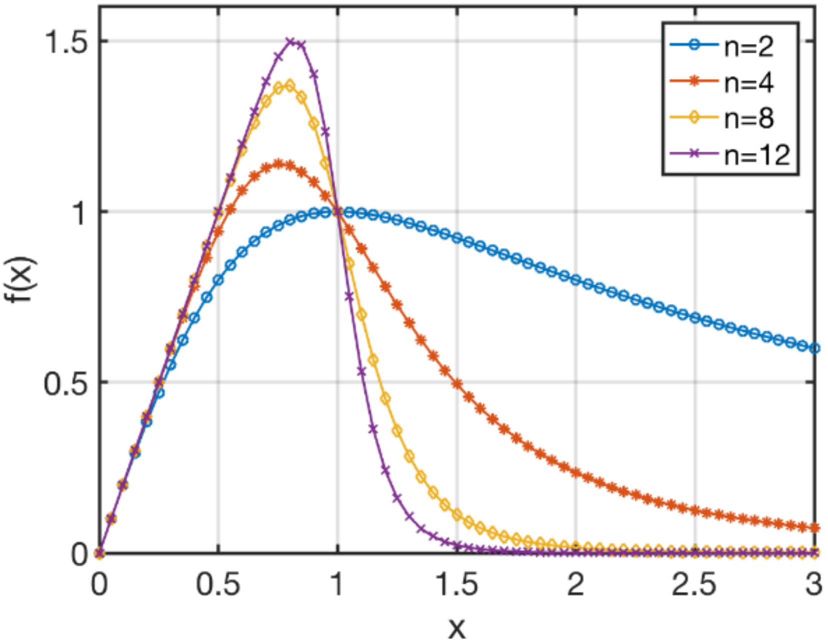
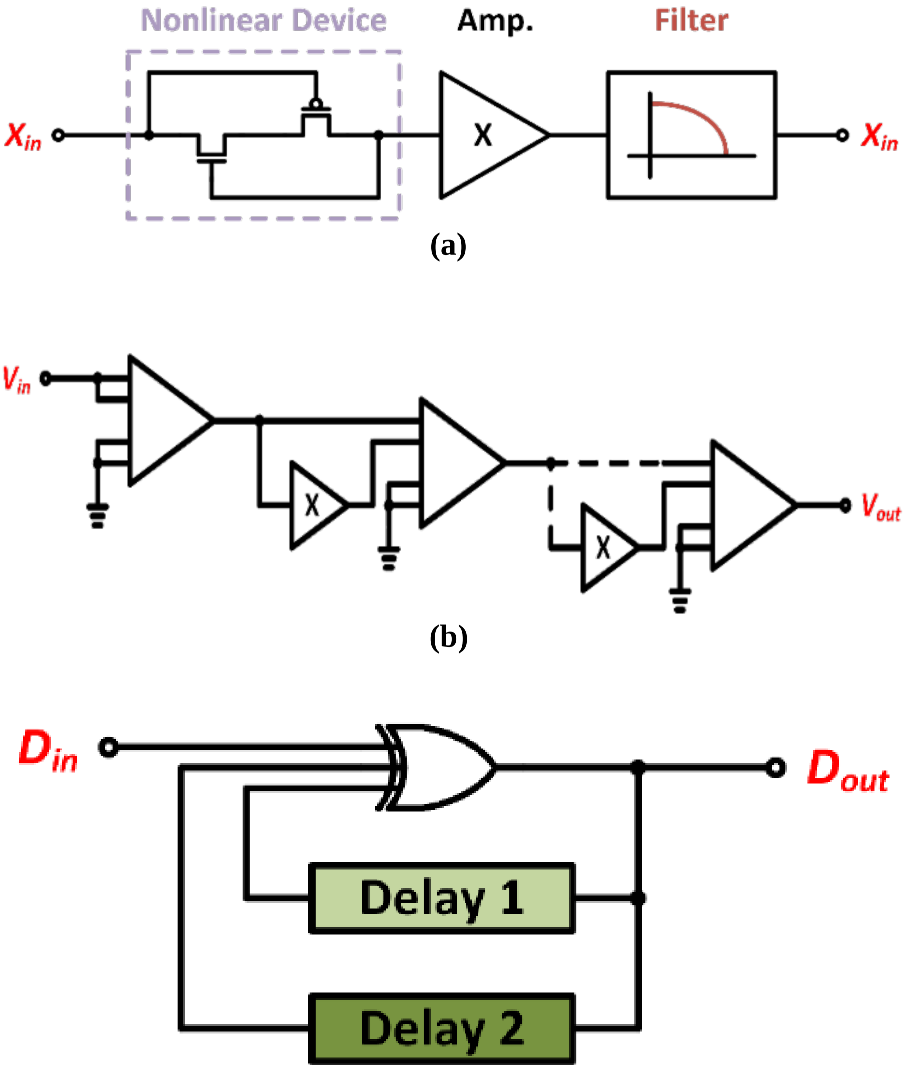


Figure 2. Mackey-Glass equation with different nonlinearity coefficients.

Implementing the Mackey-Glass function into the hardware is a promising area that needs widespread attention and exploration. Only a few research efforts on the implementation of Mackey-Glass model in electronic circuits were discovered in the literature [21-23]. Results from the literature demonstrate its capability of generating high-dimensional data and its potential for tuning dynamical behavior by varying the delay. However, all of the work focused on the concept of approval with discrete components. To the best of our knowledge, real-world applications with on-chip computing capability have not been discovered yet.



(a)

(b)

(c)

Figure 3. Simplified electronic circuit models of the Mackey-Glass nonlinear equation in (a) JFET model; (b) analog multiplier model; (c) autonomous Boolean model.

The junction gate field-effect transistor (JFET) model is one of the most iconic electronic circuit schemes of Mackey-Glass function, as shown in Figure 3(a). In the JFET model, the nonlinear characteristic is mimicked by coupling a pair of n-type and p-type JFETs. The transfer function can be expressed as

$$X_{out} = \frac{C \cdot X_{in}}{1 + b^p (X_{in})^p}, \tag{4}$$

where C is the equivalent capacitance within the low-pass filter, b^p is the total finite gain within the amplifier stage. Among all these Mackey-Glass electronic circuit models, the JFET model has been widely used to implement the Mackey-Glass function due to its simple structure. To recover the coupling loss in the nonlinear device and reduce the noise interference of the analog signals, an amplifier and low-pass filter are implemented.

In the analog multiplier model, the nonlinear characteristic is modeled by coupling multiple four-quadrant-multiplication analog multipliers, as shown in Figure 3(b). The transfer function can be simplified as

$$f(v) = \beta \frac{v}{\Theta^n + v^n}, \tag{5}$$

where Θ is the scaling factor of the analog multiplier, and β is the finite gain of the operational amplifier. Although results demonstrate its capability of nonlinear mapping, the circuit implementation is extremely complicated.

Autonomous Boolean model, a purely digital implementation of the nonlinear function, overcomes design drawbacks in the analog implementation of the Mackey-Glass equation, as shown in Figure 3(c). The nonlinear characteristic is emulated by an XOR logic gate and two digital delay loops with a lookup table scheme. The nonlinear function in digital format can be described as

$$d_{i,j}(t) = \frac{1}{\tau} \int_t^{t+\tau} x_i(t') \oplus x_j(t') dt', \tag{6}$$

where $d_{i,j}(t)$ represents output signals as a 2-bit digital code. The autonomous Boolean mode is one of the most accurate Mackey-Glass electronic circuit models due to its advantages of noise immunity and the mature manufacturing process in digital implementation.

3.1.2 Mackey-Glass nonlinear electronic circuit with low-power CMOS technology

Generally, the structure of our introduced Mackey-Glass nonlinear electronic circuit is comprised of a single-ended charge pump (I_p, I_n, SW_p and SW_n), a loop filter, an operational amplifier, and an output current mirror (M_1 ~ M_3), as illustrate in Figure 4.

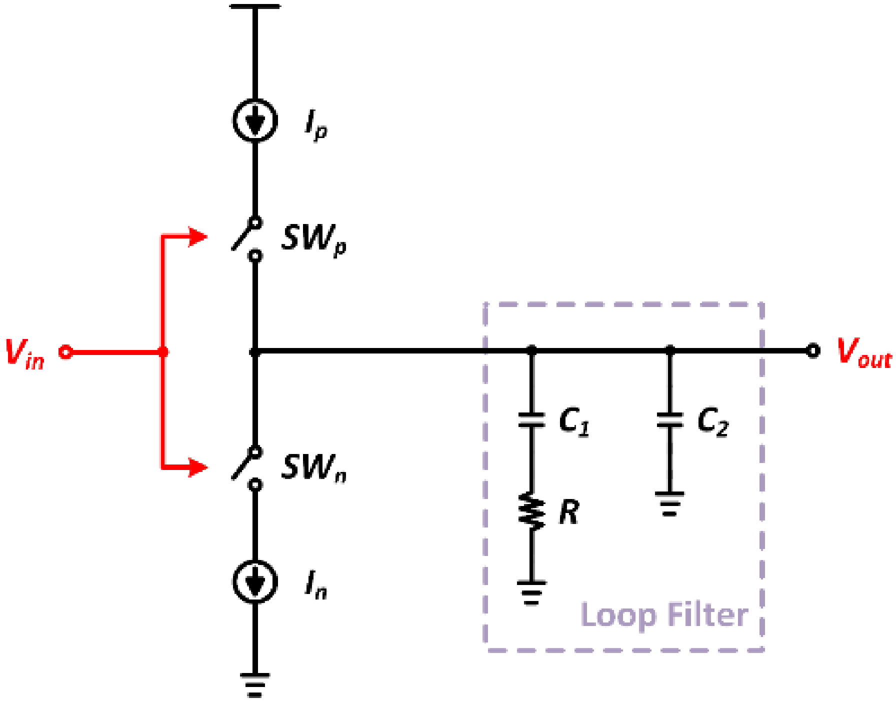


Figure 4. Simplified design scheme of charge pump (CP)-based Mackey-Glass nonlinear electronic circuit.

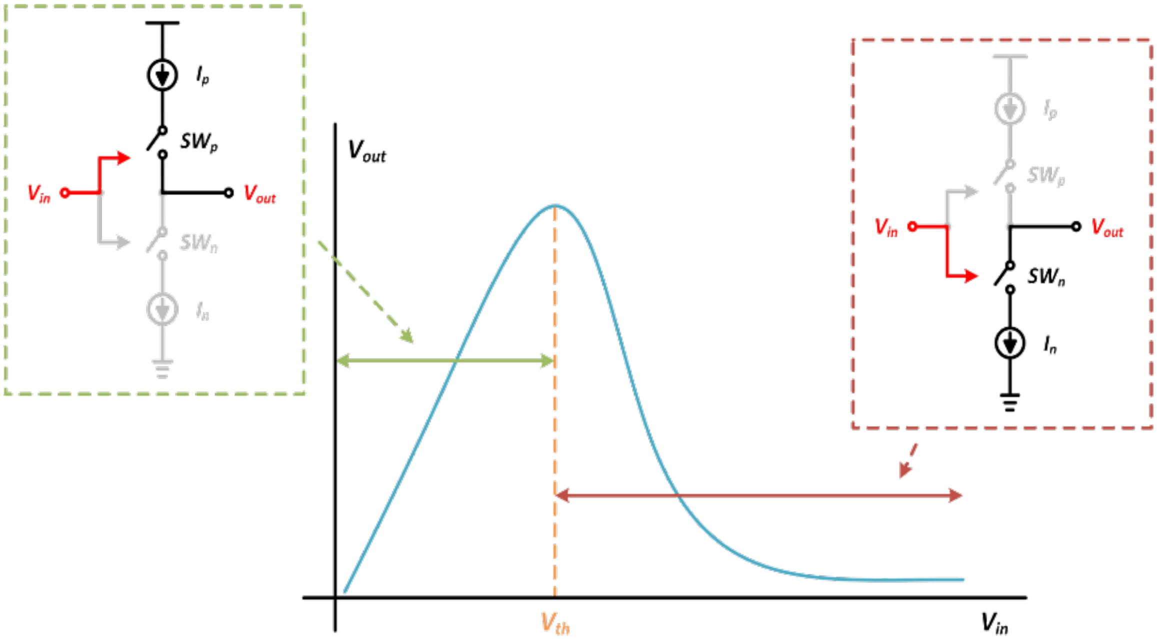


Figure 5. Operating principle of Mackey-Glass nonlinear electronic circuit through the physical charging and discharging behavior of CMOS transistor.

The nonlinearity of the Mackey-Glass equation can be emulated by coupling the physical charging and discharging behavior of CMOS transistors, as depicted in Figure 5. Charging and discharging operations of the system are achieved by comparing the potential level of the input signal and the threshold potential of triggering switches (SW_p and SW_n). When the input voltage is less than the threshold potential, the circuit is operated at the charging mode. Similar to the control mechanism of the water valve, the charging current, I_p, through the loop filter is regulated by the SW_p. During the charging operation, the voltage across the loop filter, V_L, starts to increase; as the input voltage increases, the p-type switch, SW_p, starts reducing charges through the loop filter, which reduces the charging rate; eventually, V_L saturates at its threshold potential level. On the other hand, when the input voltage is higher than the threshold potential, the circuit is operated at the discharging mode. As the input voltage keeps increasing, the n-type switch, SW_n, starts increasing the discharging rate, which drains charges from the loop filter to GND; eventually, V_L saturates at its minimum potential level.

3.1.3 Dynamic signal conversion circuit design with trans-impedance amplifier and embedded loop filter

Noise analysis is one of the key aspects of analog circuit design. A loop filter is often required to be embedded into the charge pump to eliminate noise, where the filter can be implemented with either passive or active filter. The active filter has embedded power gain and smaller design area but with limited bandwidth; while the passive filter has better stability and lower energy consumption but with larger design area. Considering the tradeoff between the stability, signal bandwidth, design area as well as energy consumption, the passive filter was implemented in our Mackey-Glass nonlinear electronic circuit.

Most electronic circuitries use voltage as a trigging signal; however, to enable the functionality of the spiking computation via our temporal encoder, the analog current is needed to trigger the LIF neuron within our temporal encoder. Therefore, the analog voltage signal, V_{out}, needs to be converted into current by the trans-conductance amplifier. The system-level implementation of our Mackey-Glass nonlinear electronic circuit with the trans-impedance amplifier is illustrated in Figure 6.

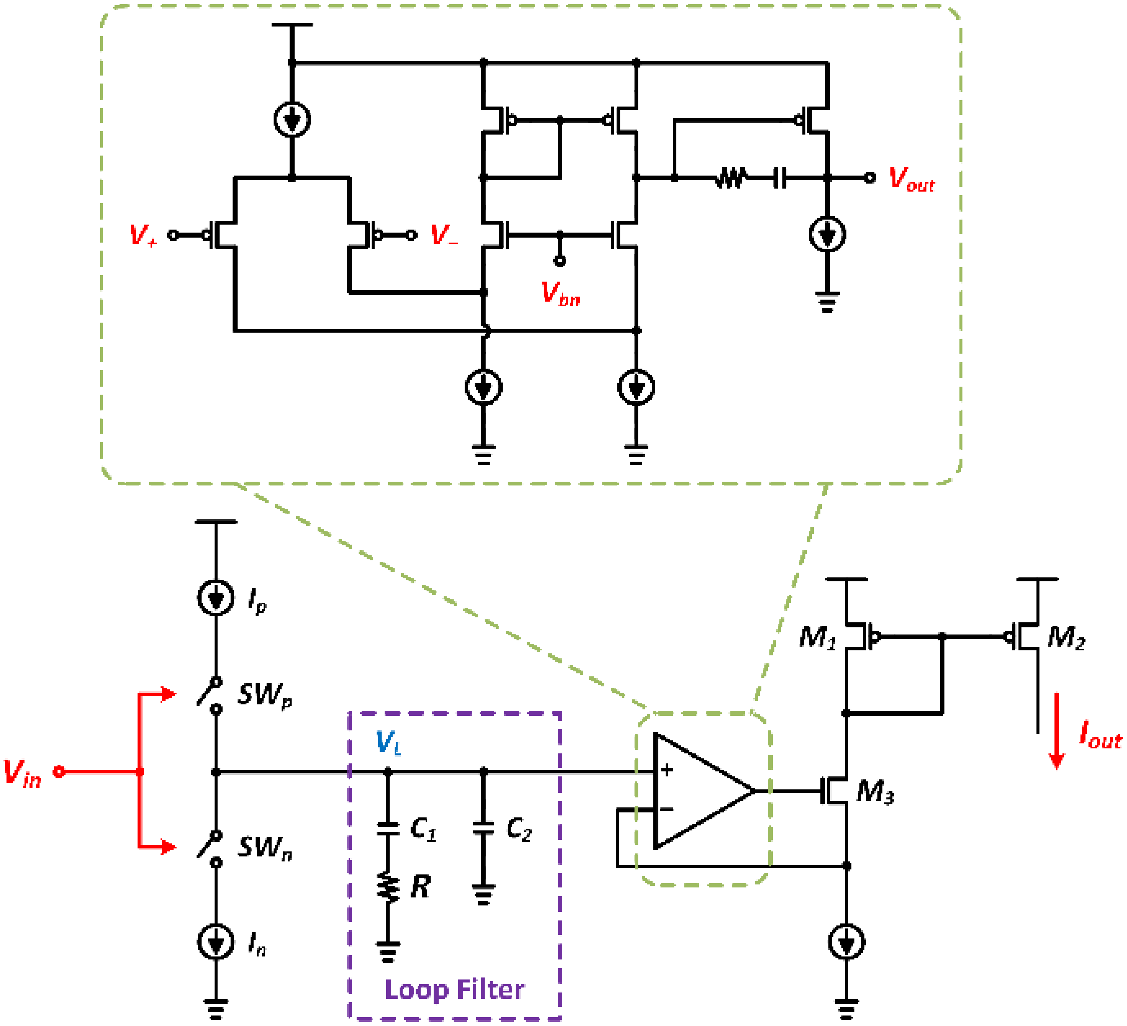


Figure 6. Simplified design scheme of charge pump (CP)-based Mackey-Glass nonlinear electronic circuit with trans-impedance amplifier.

During the operation, the output current mirror keeps tracking the variation of V_L and linearly generates the current to the temporal encoder. The operational amplifier, which is used to create the negative feedback loop to maintain the stability of the conversion process, is optimized such that it can operate in the sub-threshold region to achieve the minimum power consumption without losing the computational accuracy. The layout of our Mackey-Glass nonlinear electronic circuit was implemented through the standard GF 180nm CMOS technology as shown in Figure 7 and Figure 8.

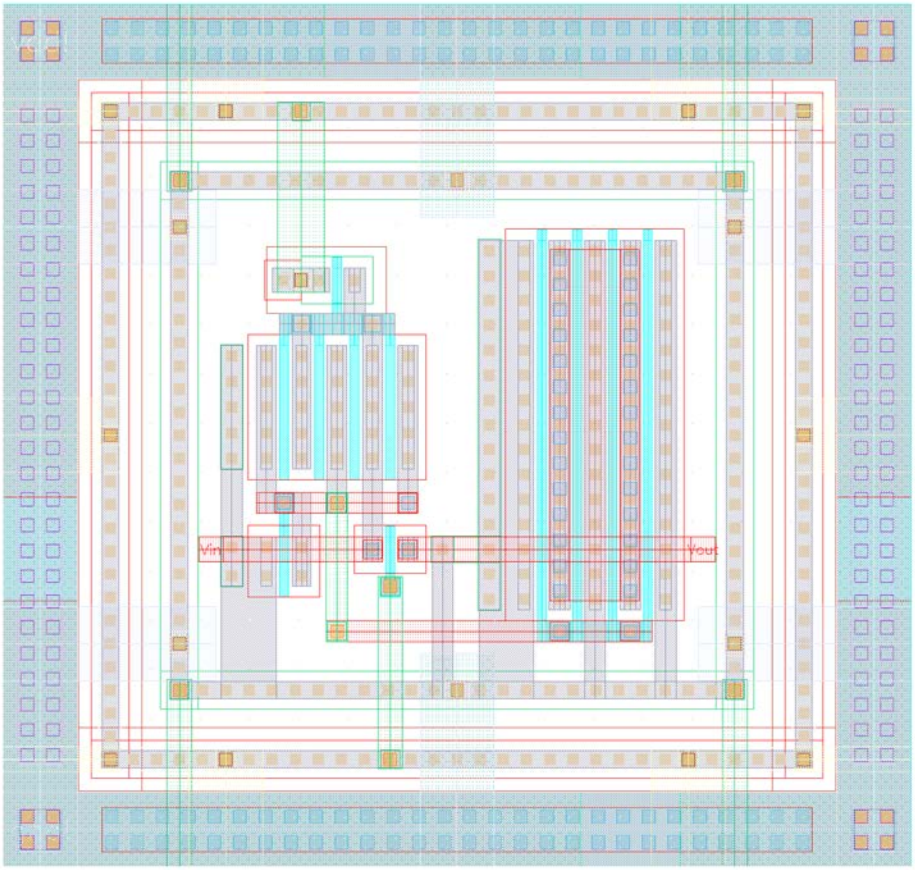


Figure 7. Layout of the proposed Mackey-Glass nonlinear module.

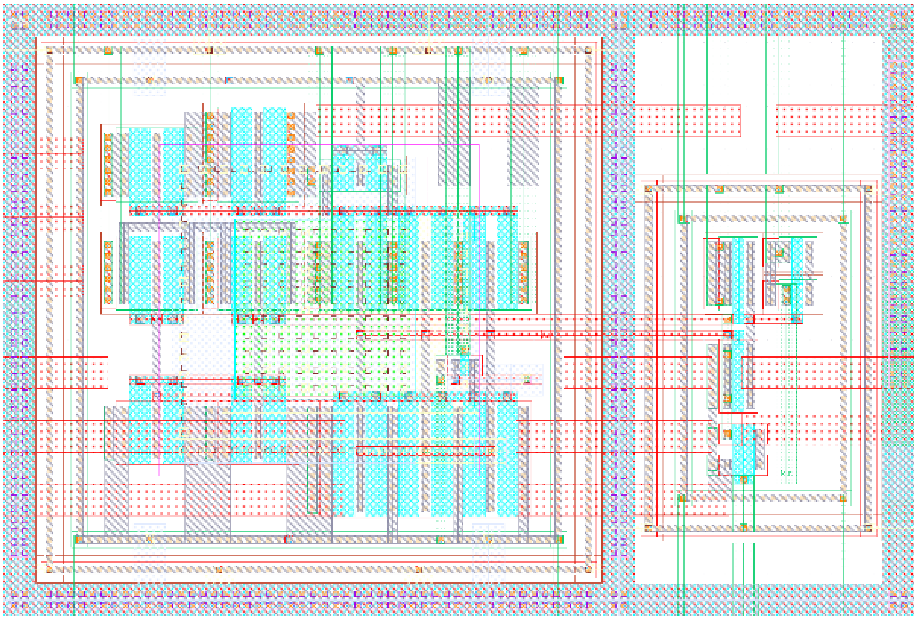
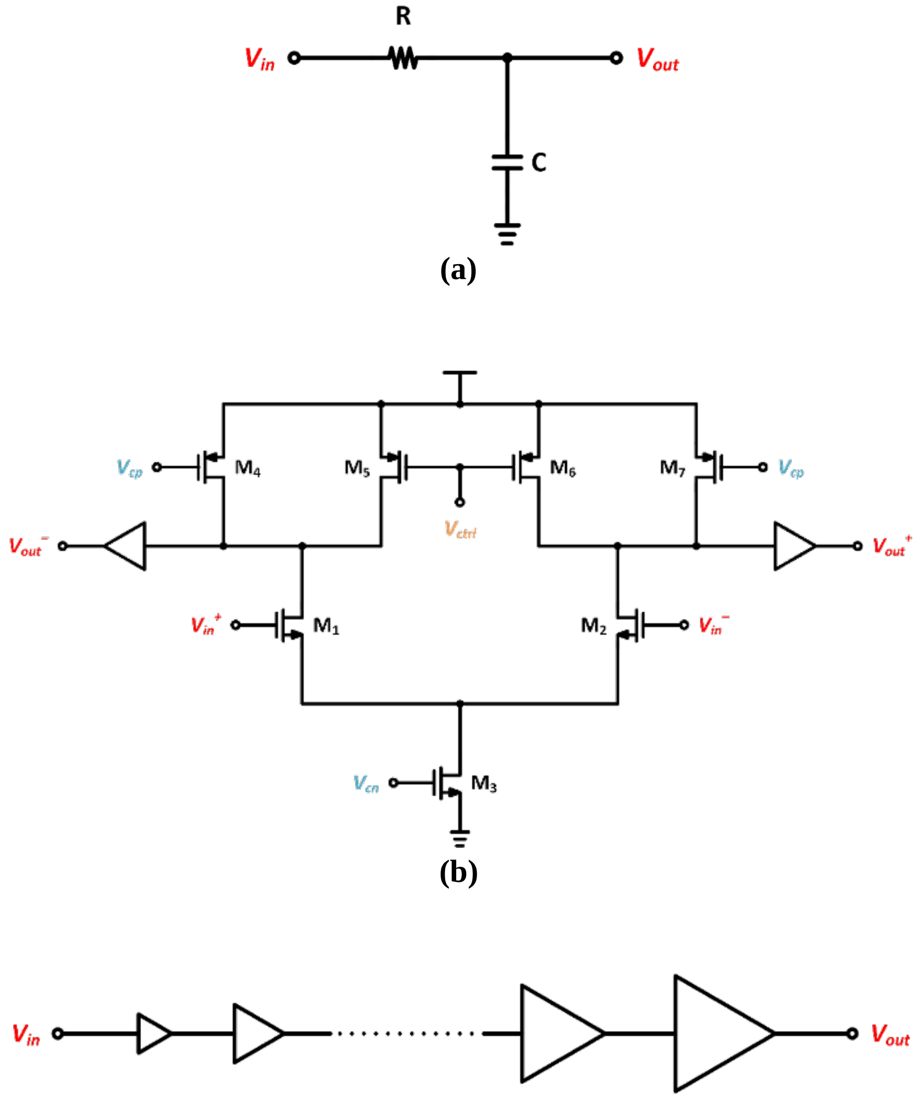


Figure 8. Layout of voltage-to-current Converter.

3.2 Static Delay Loop Design with Integrate-and-Fire Neuron

3.2.1 Comprehensive investigation on delay models

Inevitably, delay is ubiquitous in almost every system. For instance, the diffusion of substances (oxygen and carbon dioxide in the blood), and intrinsic time for transportation between neurons [24]. With the delay embedded, the system exhibits the near chaotic regime behavior. Delay elements in the reservoir layer compute the functionality of nonlinear mapping with delay whereby the biological behavior of the neural system is represented. Various electronic circuit implementations for delay elements have been studied including the resistor-capacitor model, digital delay line (DDL) model, and differential delay element model, as summarized in Figure 9.



(a)

(b)

(c)

Figure 9. Simplified electronic circuit models of delay element in (a) Resistor-Capacitor model; (b) differential delay element model; (c) digital delay line model.

The traditional Resistor-Capacitor delay model, as shown in Figure 9(a), is commonly implemented into analog or digital circuits to form a desired delay time due to its simple structure and wide controllable dynamic range. The delay time is generally known as the one time constant (1τ) of the Resistor-Capacitor circuit, which can be expressed as $\tau = R \cdot C$. The delay time is formed by the charging and discharging process across the capacitor, as shown in Figure 10. As the capacitor starts charging up, the voltage potential across the capacitor slowly increases. The one time constant (1τ) is triggered when the capacitor's potential reaches 63% of its maximum possible voltage.

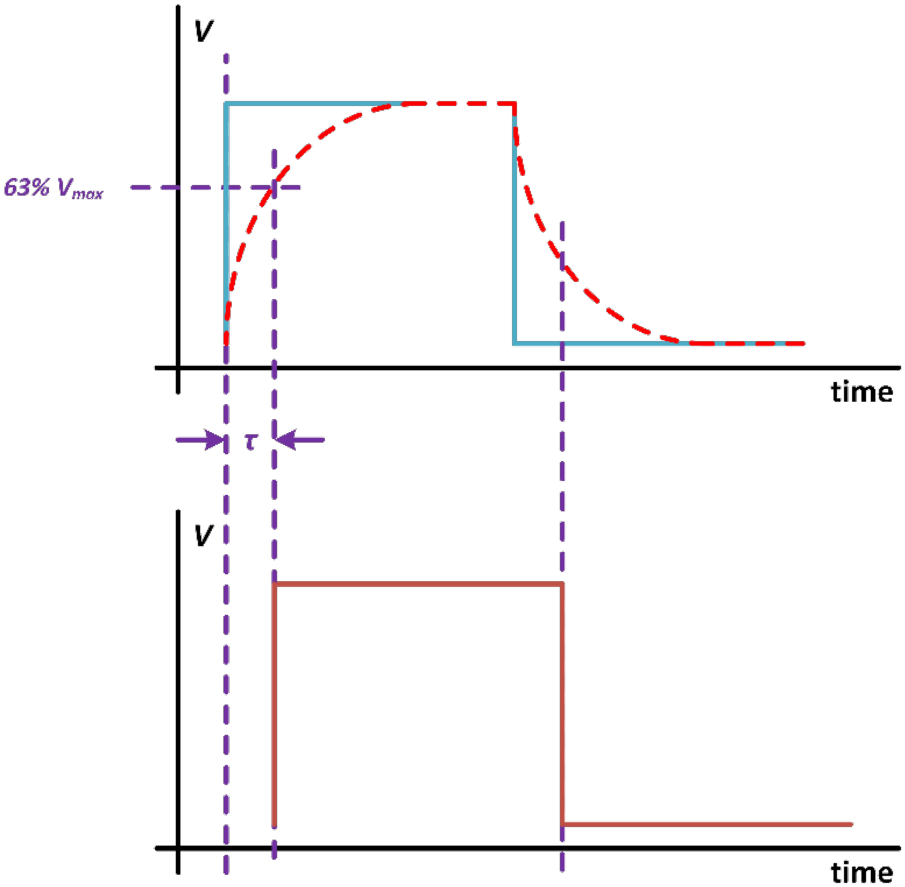


Figure 10. Operating principle of the Resistor-Capacitor delay model.

In general, the delay time can be controlled by regulating the resistor and capacitor. However, since the delay time is proportional to the resistance and capacitance values, longer delay time often requires a larger resistor and capacitor, which dramatically increase the die area in the IC implementation.

The DDL model, as shown in Figure 9(c), is widely used in the IC implementation for the time alignment. The delay circuit is constructed by a cascading digital buffer (event number of inverters); in order to maintain the load driving capability, the size of each inverter is two times larger compared to its previous stage. The delay is formed by the response time of each inverter, which is determined by the parasitic resistance and capacitance of each inverter. Unlike the traditional Resistor-Capacitor delay circuit, the delay that is formed by the parasitic resistance and capacitance via the inverter is usually in the pico-second range. A longer delay time with DDL often results higher power consumption and larger die area, as a large number of inverters are needed with scaling parameters.

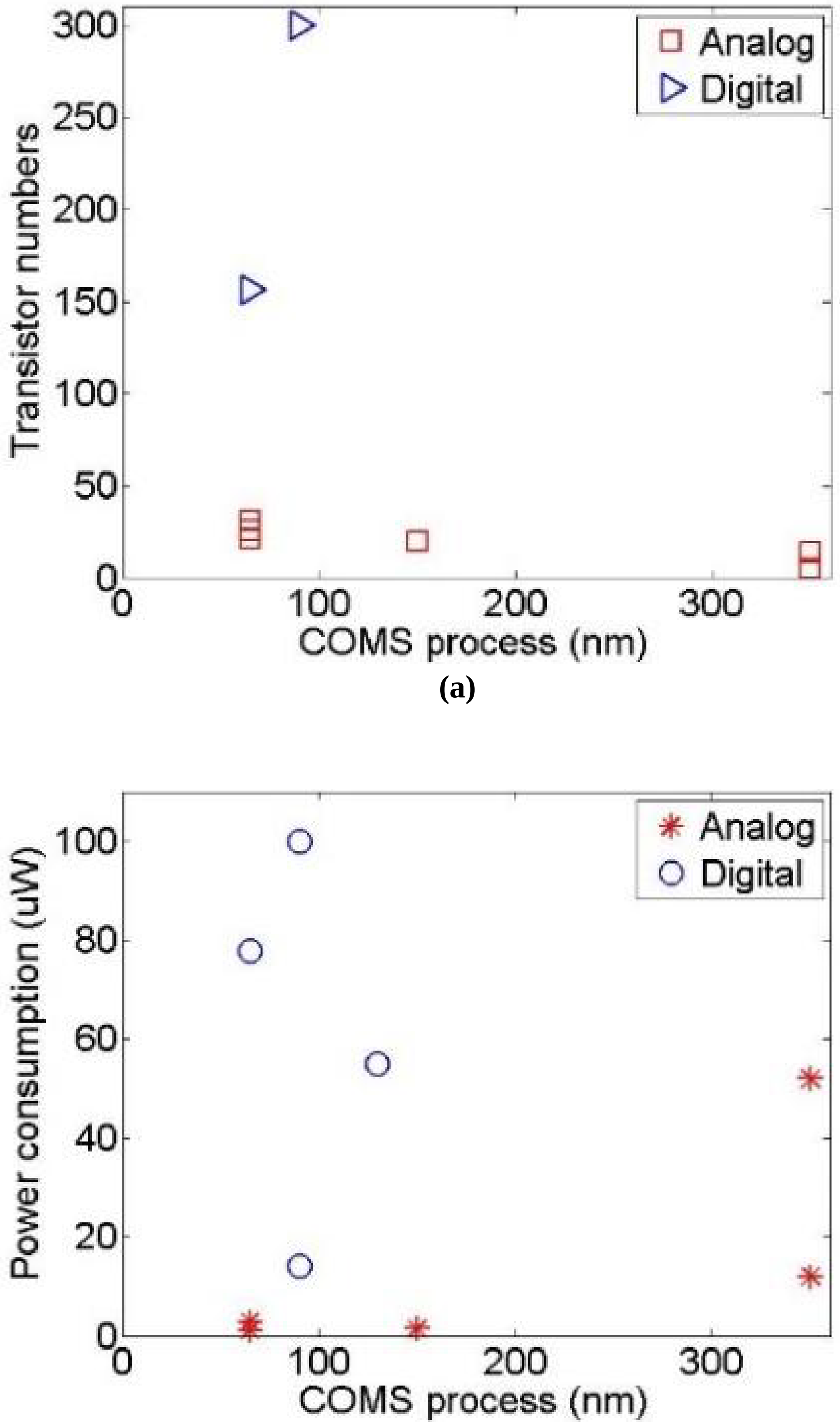
Similar to the DDL model, the delay time within the differential delay element model, as shown in Figure 9(b), is proportional to the response time of the circuit itself. Compared to the single inverter; the comparator requires longer response time to generate the output signal, and thus, the differential delay element model has the potential to generate a larger dynamic range of delay time with small design area.

Since the neuron is the fundamental component in a neuromorphic system, power consumption and die area play significant roles in the neuron system design. In general, there are two implementations available: digital and analog. As discussed in Section 2.0, digital implementation is usually preferred due to its advantages in ease of implementation and noise immunity, while analog implementation closely mimics the physical characteristic of the neurological system.

Table 1 and Figure 11 compare the digital implementation and the analog implementation. Note that we adopted the same normalization method to estimate power consumption, die areas, and transistor size.

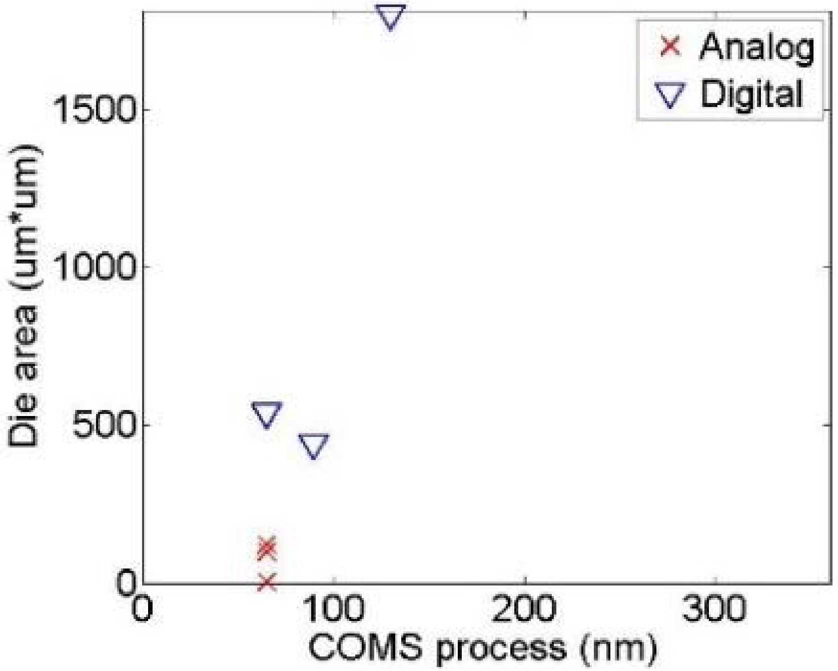
Table 1. Comparison between analog and digital implementations.

		CMOS Process	# of Transistors	Design Area	Power Consumption
Analog	[25]	$350nm$	14	N/A	$52\mu W$
	[26]	$350nm$	5	N/A	$12\mu W$
	[14]	$65nm$	31	$120\mu m^2$	$3.8\mu W$
	[27]	$150nm$	20	N/A	$1.5\mu W$
	[28]	$65nm$	26	$100\mu m^2$	N/A
Digital	[28]	$65nm$	N/A	N/A	$100\mu W$
	[29]	$65nm$	156	$538\mu m^2$	$77.9\mu W$
	[7]	$130nm$	N/A	$1800\mu m^2$	$55\mu W$
	[8]	$90nm$	300	$440\mu m^2$	$14.3\mu W$


Transistor numbers
300
250
200
150
100
50
0
0
100
200
300
COMS process (nm)
Analog
Digital
Power consumption (uW)
100
80
60
40
20
0
0
100
200
300
COMS process (nm)
Analog
Digital

(a)

(b)



(c)

Figure 11. Comparison of analog and digital neuron implementations using various CMOS technologies based upon (a) transistor numbers, (b) power consumption, and (c) die size.

We normalized the data presented in Table 1 and summarized the comparison of power consumption between analog and digital implementations in Table 2.

Table 2. Normalized Comparisons.

	Analog	Digital
# of Transistors	19.5	225
Design Area	$74.2\mu m^2$	$1119.7\mu m^2$
Power Consumption	$1.15\sim52\mu W$	$14.3\sim100\mu W$

3.2.2 Design and optimization of integrate-and-fire delay neurons

The design scheme of our introduced I&F delay neuron is depicted in Figure 12. During the operation, the sensing capacitor, C_S, continuously tracks the current that is generated from the delay calibration module and charges up its potential. When the voltage potential across the sensing capacitor, $V_{th(c)}$, exceeds the firing threshold voltage of input transistors, $M_{1,2}$, two cascading inverters, $M_{3,4}$ and $M_{7,8}$, fire a spike as output. Meanwhile, the positive feedback loop, $M_{5,6}$, induces a high voltage at V_{reset}, such that the sensing capacitor can be fully discharged by the resetting switch, SW_2. As such, the firing process for one output spike is accomplished.

In the I&F delay neuron, the delay time can be regulated by the integrating time of the membrane capacitor, C_m. The delay time constant, τ, can be expressed as

$$\tau = C_m \cdot \frac{V_{th}}{I_{ex}}, \tag{7}$$

where the V_{th} is the threshold voltage of the I&F neuron, and I_{ex} is the controllable excitation current. Theoretically, the mathematical analysis of the I&F delay neuron is similar to the traditional Resistor-Capacitor delay model; since the input impedance, R_{in}, of the I&F delay neuron is equivalent to $\frac{V_{th}}{I_{ex}}$. Thus, the delay time constant can be then rewritten as

$$\tau = C_m \cdot R_{in}. \tag{8}$$

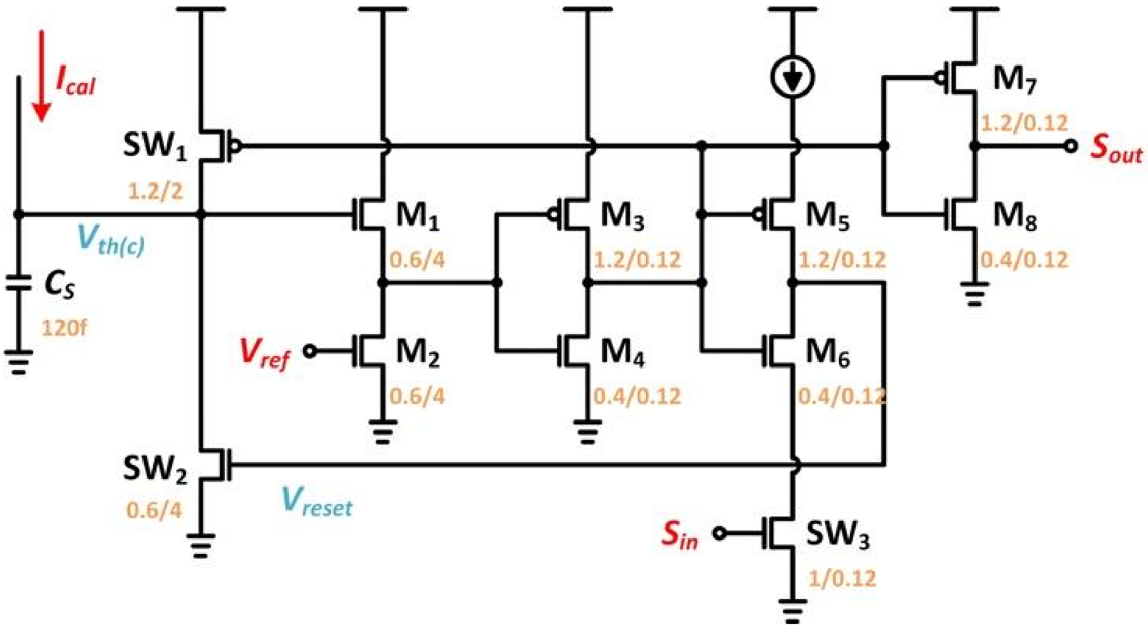


Figure 12. Simplified design scheme of integrate-and-fire delay neuron.

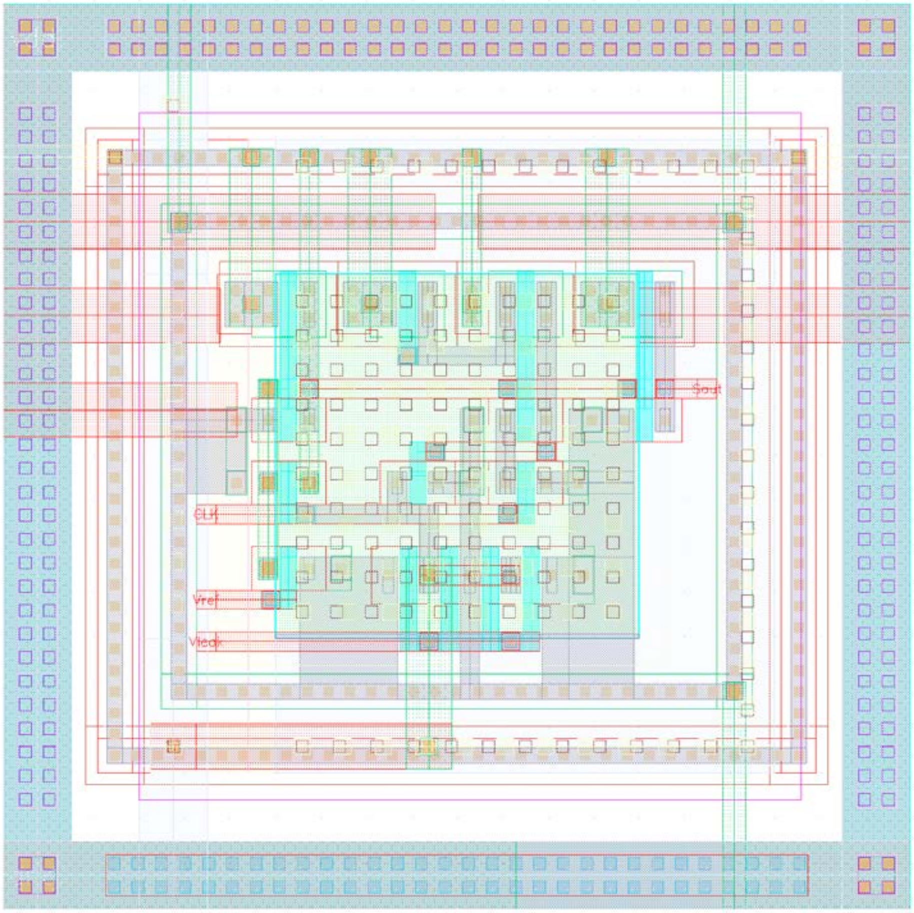


Figure 13. Layout of the integrate-and-fire delay neuron.

However, unlike the traditional Resistor-Capacitor delay model that is formed by a large capacitor, the delay time of the I&F delay neuron can be regulated by the input impedance as described in Equation (8). Consequently, a large delay time can be achieved by increasing the equivalent input impedance, which can be formed by reducing the I_{ex}.

To facilitate the functionality of dynamic behavior regulation, from periodic to chaotic or vice versa, the delay calibration module, as depicted in Figure 14, is implemented.

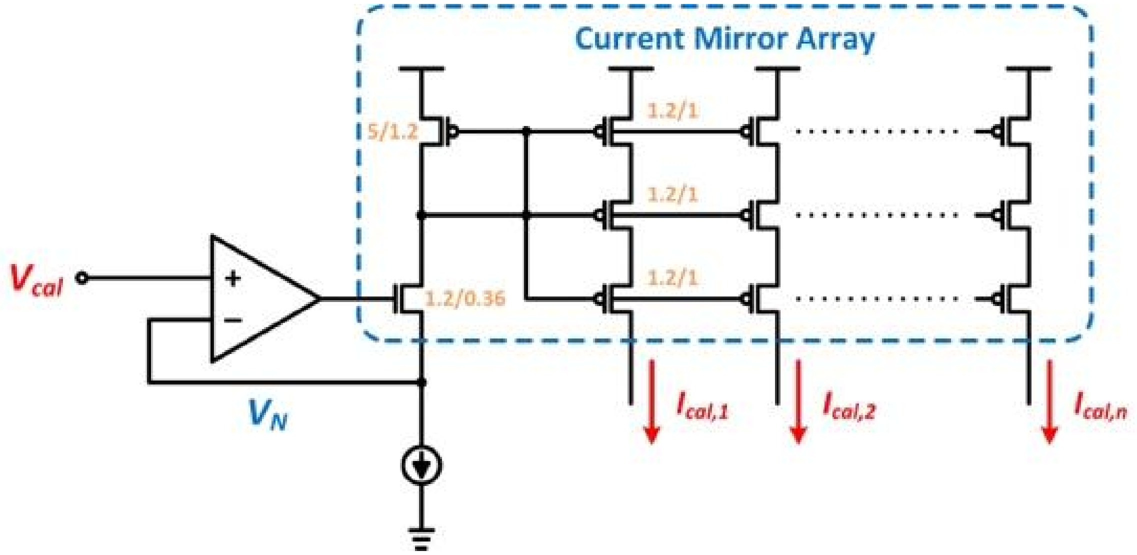


Figure 14. Design scheme of delay calibration module.

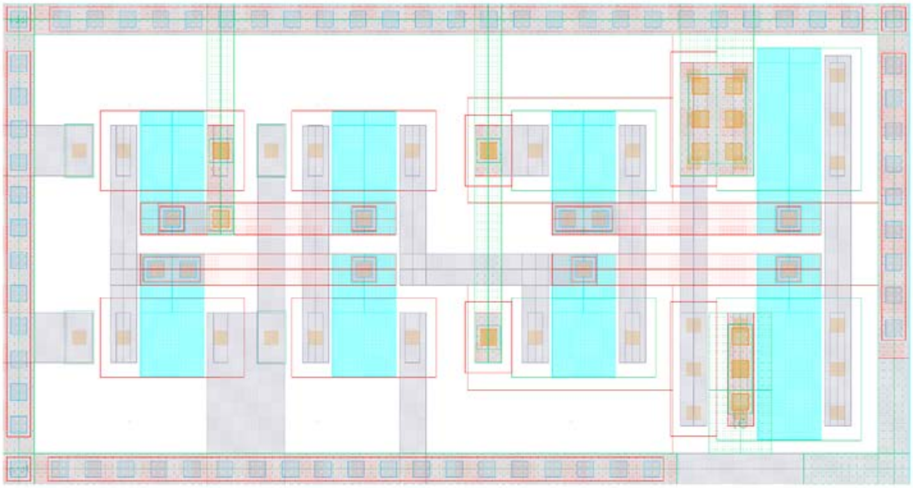


Figure 15. Layout of delay calibration module (single-cell), where multiple cells are needed based on the number of delay neurons.

The delay calibration module utilizes the voltage-to-current conversion technique from the conventional trans-impedance amplifier. The current mirror array, as shown in the dashed box in Figure 14, keeps sensing the variation of the input signal, V_{cal}, and linearly generates the corresponding calibration current, I_{cal}, which could be expressed as

$$I_{cal} = (V_{cal} - V_N) \cdot G_m, \tag{9}$$

where V_N is the feedback signal from the current mirror, and G_m is the trans-conductance of the operational amplifier. Within the dynamic delay-feedback loop, each I&F delay neuron has equidistant delay time constant, and thus, the current mirror array within the delay calibration module is designed to achieve identical calibration currents.

3.2.3 Design and optimization of delay-feedback loop

The delay-feedback loop, which is constructed with multiple I&F delay neurons, as illustrated in Figure 16, is implemented by using the output spike train from the previous neuron as the clock trigging signal to its following. For instant, when the temporal spike train is generated from the first delay neuron, n_1, within the delay loop, it resets the following delay neuron, n_2; meanwhile, the voltage potential across the sensing capacitor of n_2 starts to charge up. Over the time period of τ_{delay}, n_2 fires a spike as output, which results in the input spike train at a given delay time.

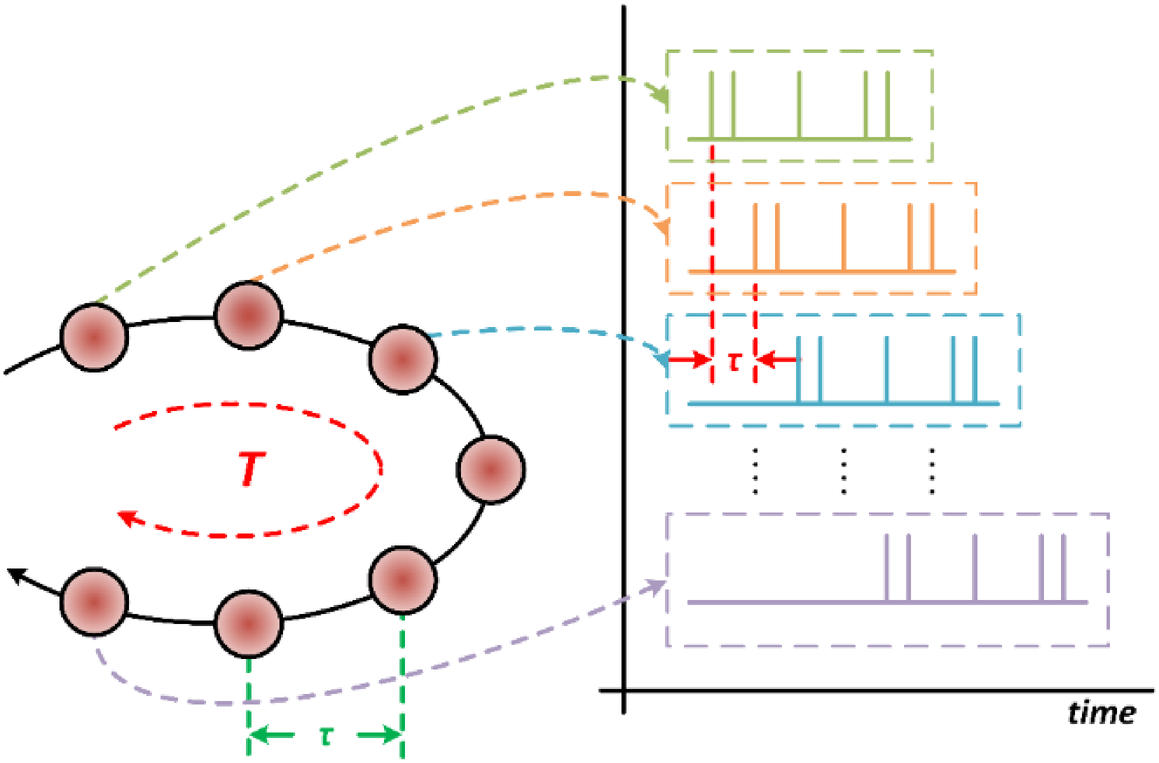


Figure 16. Simplified design scheme of the delay-feedback loop.

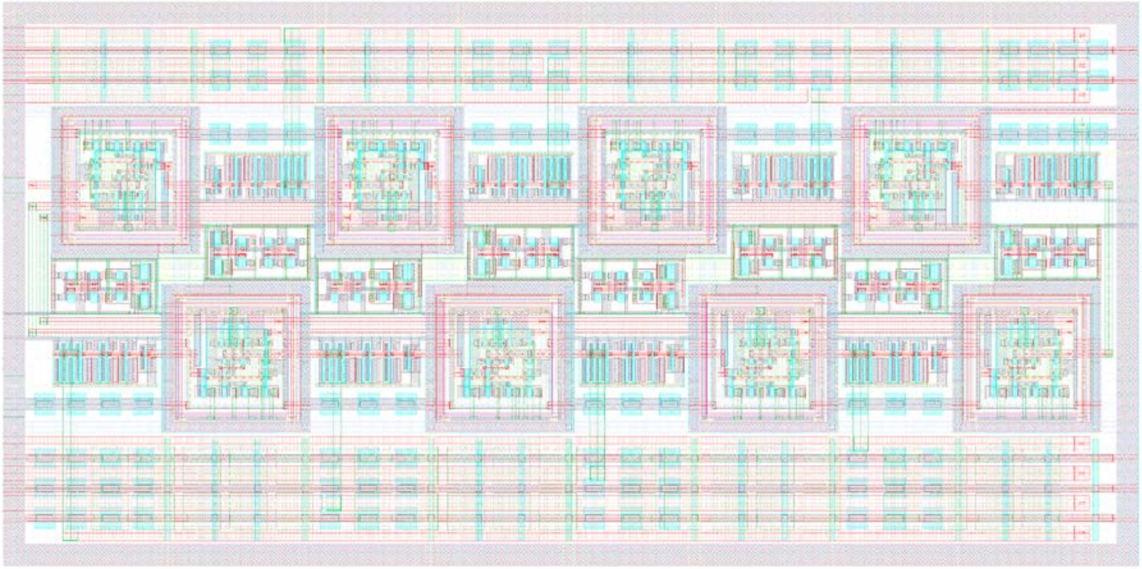


Figure 17. Layout of the delay-feedback loop.

The delay plays an important role in the reservoir layer, which determines the dynamic behavior of the system. The robustness of the I&F delay neuron is proportional to transistors' channel length in the input stage of the firing threshold. Increase the channel length of the transistor significantly improve the system robustness, which often results in a larger design area.

3.3 Short-term Memory Achievement with Signal Integrator

3.3.1 Comprehensive investigation on signal integration

By integrating the feedback signal with the new incoming input data, the DFR utilizes the delayed feedback loop to formulate a short-term dynamic memory, as depicted in Figure 18, such that the new incoming data carries the information from its previous state(s). To improve the computation accuracy, the feedback signal is decoded into an analog voltage followed by a gain regulator to scale down the potential level such that the new incoming input data is dominant.

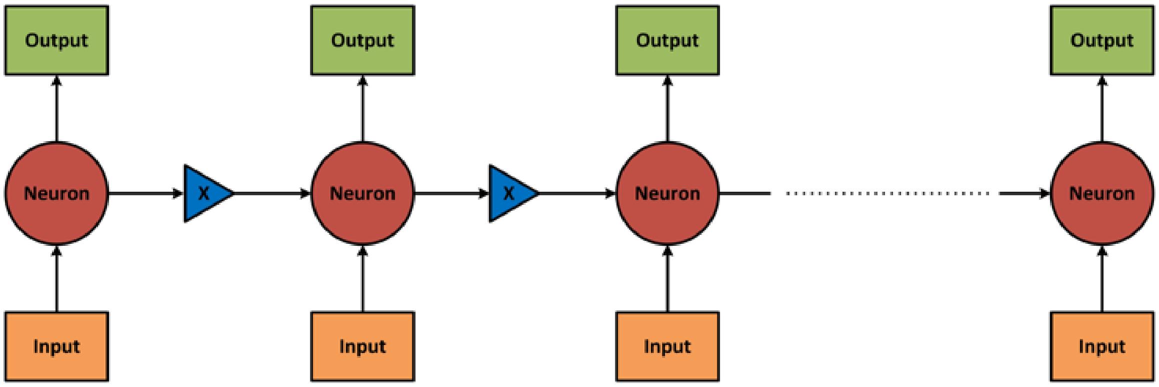


Figure 18. Simplified block diagram of short-term dynamic memory integration.

Unlike the rate encoding scheme, the information of the temporal code is encoded into the time interval between spikes; in other words, the total number of spike is fixed, and the time intervals between spikes alter based on the given input data. Therefore, two different temporal codes cannot be merged directly, which is illustrated in Figure 19.

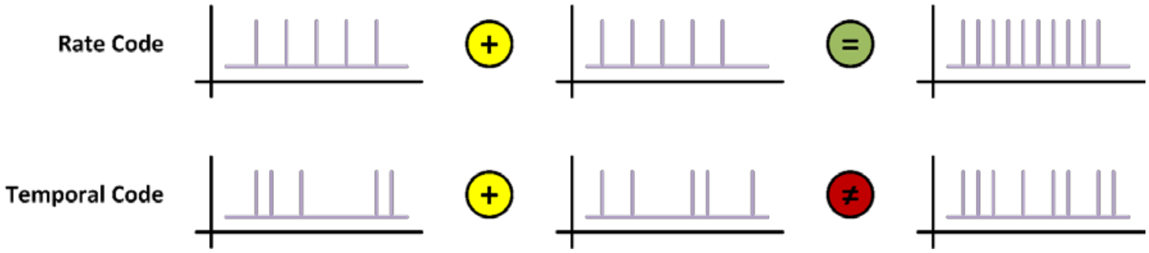


Figure 19. Integration of rate code vs. temporal code.

To integrate two different temporal codes together, all-time intervals between spikes need to be extracted for computation. The time intervals in the output spike train can be express as

$$D_{sN} = f(D_{aN}, D_{bN}) , \tag{10}$$

where D_{sN} defines the N-th time interval in the output spike train, D_{aN} and D_{bN} represent the N-th time interval in the input pattern A and B, respectively. Figure 20 illustrates the computing principle of the temporal code.

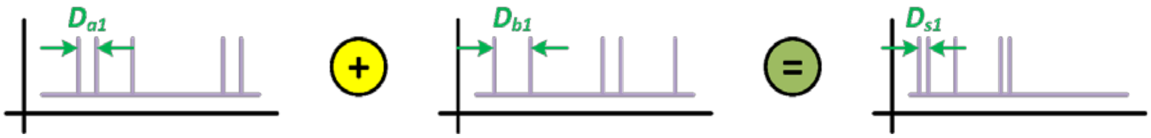


Figure 20. Computing principle of temporal code.

Since two different temporal codes cannot be merged directly, the time intervals between spikes need to be extracted. However, the spike-representative data cannot be used for data computation directly due to the narrowed spike width. This task tentatively utilizes the DDL with the XOR logic gate to extend the spike width, as demonstrated in Figure 21.

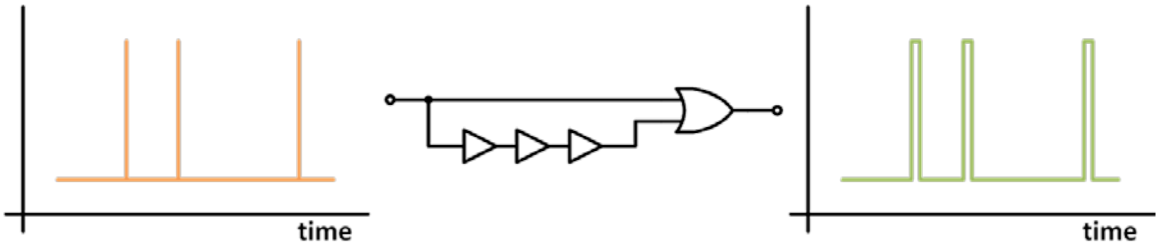


Figure 21. Spike width extender.

Figure 22 depicts the interval extraction. The sampling window is used as a reference to extract the time intervals between spikes, and thus, time intervals are represented as digital pulses. Depending on the design scheme of the signal integrator, either time of arrival or lookup table design scheme is used.

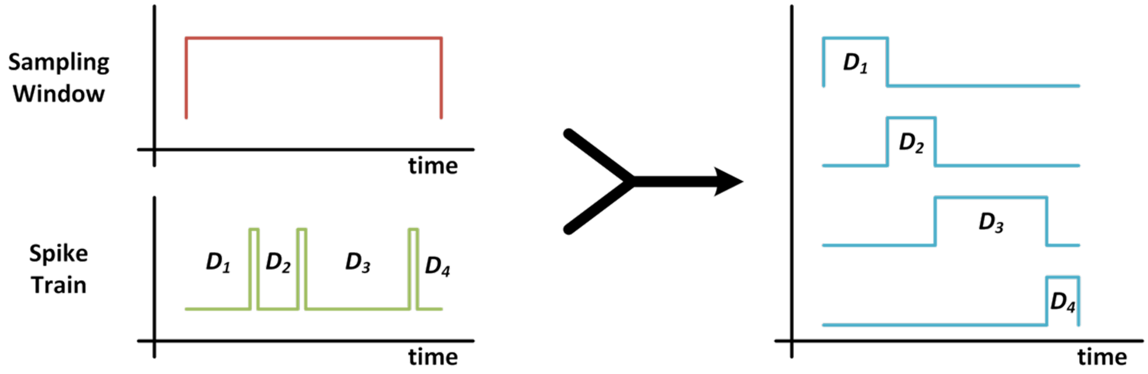
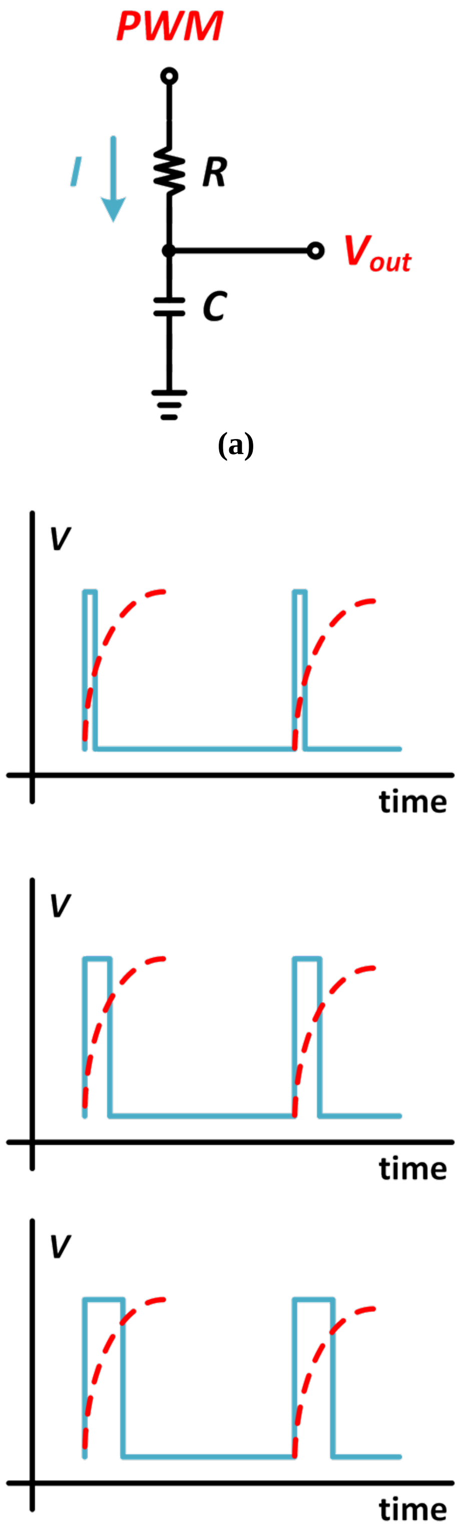


Figure 22. Time-interval extraction.

In the analog implementation, the time-of-arrival (only the time interval between the rising edge of the sample window and the first spike) and current sensing scheme are tentatively utilized.

With the current sensing design scheme, as shown in Figure 23, the voltage is depended on the integration over time through the sensing capacitor, which can be expressed as:

$$V = \frac{I}{C} \cdot \Delta t, \tag{11}$$



(a)

(b)

Figure 23. (a) Design scheme and (b) operating principle of current sensing.

where I and C represent the sensing current and sensing capacitor, respectively; Δt defines the integration time, which is regulated by the pulse width modulation (PWM) of the first-time interval in the temporal code. In the proposed circuit implementation, the sensing capacitor is fixed; moreover, since the amplitude of the PWM is fixed at V_{DD}, the sensing current maintains a constant. Thus, the voltage is only proportional to the PWM signal.

In the digital implementation, all-time intervals are utilized for digital computation using the lookup table design scheme. Figure 24 demonstrates the design scheme with a digital integrator.

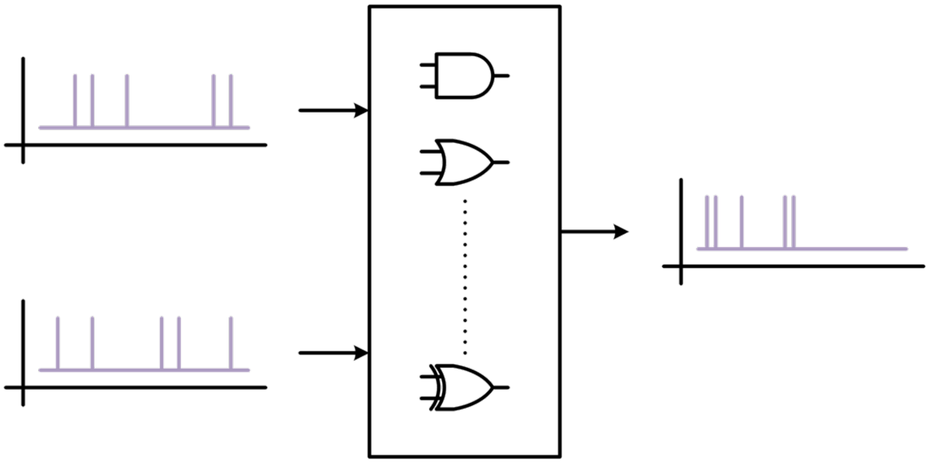
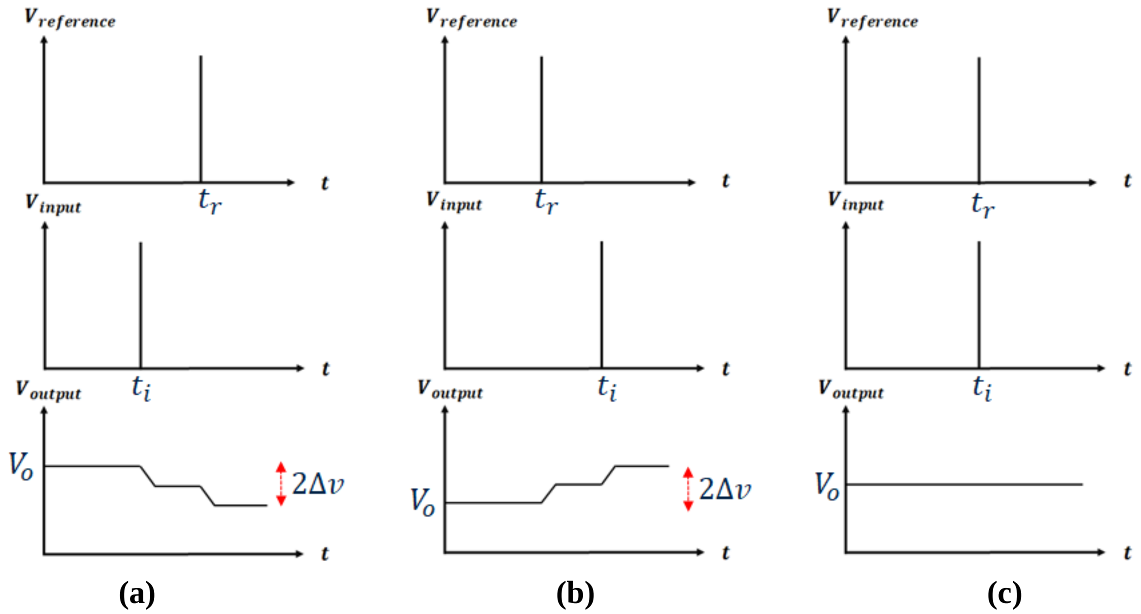


Figure 24. Design scheme of digital integrator.

3.3.2 Spike-timing-dependent plasticity-based inter-spike-interval decoder

In our first-generation decoder, the decoding scheme is based on two techniques including capacitor charging/discharging and interval extracting[16]. In the standard CMOS process technology, the most accurate capacitor could be made through the metal-insulator-metal (MIM) capacitor, which is restricted by the technique parameter offset. This kind of parameter offset would introduce a great number of errors if just one capacitor is adopted and no compensation is applied. Furthermore, due to the narrow pulse width of a spike train, it is difficult to charge the capacitor into a higher value (e.g., the 100mV level). Consequently, the final output voltage would be in the several-mV range, resulting in very low noise immunity. In order to resolve this issue, we connect two spikes with STDP scheme, as demonstrated in Figure 25.



(a) (b) (c)

Figure 25. Spike-timing-dependent plasticity in (a) t_i<t_r; (b) t_i>t_r; (c) t_i=t_r.

As shown in Figure 25, each column has three signals, which are reference spike, input spike, and output voltage, respectively. As shown in Figure 25(a), the input spike appears earlier than the reference spike, and thus, the output voltage increases by $2 \cdot \Delta v$. As shown in Figure 25(b), the input spike appears later than the reference spike, and thus, the output voltage decreases by $2 \cdot \Delta v$. The final case is described in Figure 25(c), where the input spike and reference spike appear at the same time, and thus, the output voltage remains unchanged.

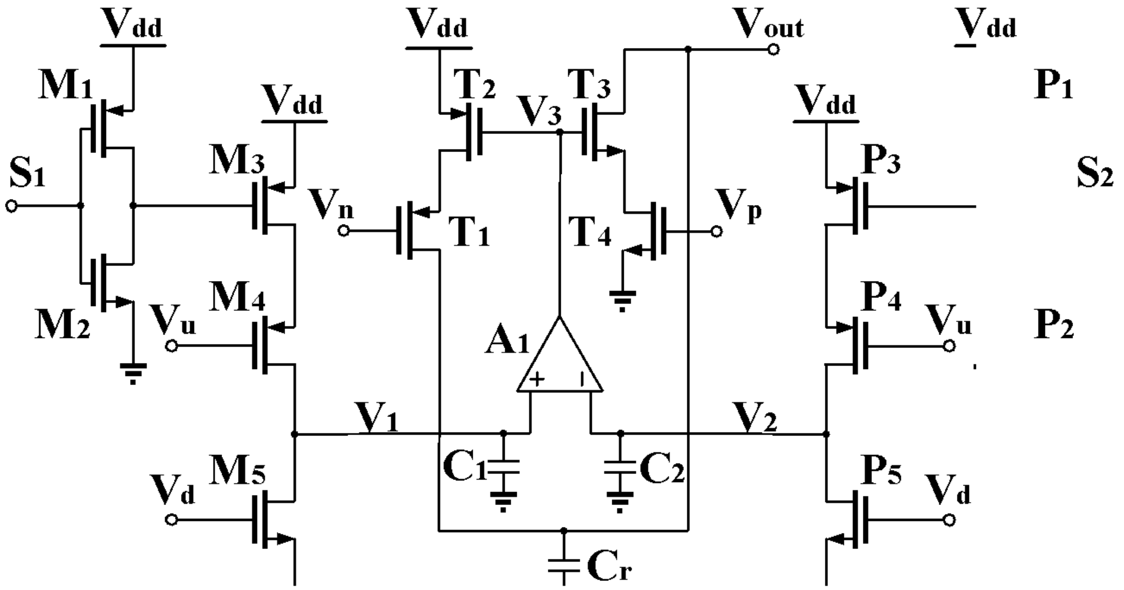


Figure 26. Design scheme of the STDP-based inter-spike-interval decoder.

As shown in Figure 26, there are three major portions in the decoder design, namely, the input module for the ISI spike train (constructed with transistors labeled in M_i), the input module for the reference spike train (constructed with transistors labeled in P_i), and the process module (constructed with transistors labeled in T_i). In the input module for the ISI spike train, the input spike is applied on the gate terminal of M_1 and M_2, which will be transformed into a spike train with the inverse voltage level. M_3 and M_4 are used to transform the inverse spike train into a current to charge C_1 capacitor. V_u is used to control the current intensity, and V_d is used to ensure M_5 working in the sub-threshold region. The input module for the reference spike train has the same specification as the input module for an ISI spike train. These voltages across C_1 and C_2, which are represented as V_1 and V_2, will be then applied to the comparator, A_1. Thereby, the output of A_1 can be expressed as

$$V_3 = \left\{\begin{matrix} V_{DD}, & V_1 - V_2 > 0 \\ \frac{1}{2}V_{DD}, & V_1 - V_2 = 0 \\ 0, & V_1 - V_2 < 0 \end{matrix}\right\}. \tag{12}$$

Once the potential difference of V_1 and V_2 is compared by A_1, T_1 and T_2 are used to charge C_r, while T_3 and T_4 are used to discharge C_r. Therefore, the output voltage of the decoder can be then determined as

$$V_{out} = \left\{ V_{ex} \cdot \left(1 - e^{-\frac{t_n}{\tau_n}}\right) \; V_{ex} \cdot e^{-\frac{D_{ref} - t_n}{\tau_p}} \right\}, \tag{13}$$

where V_{ex} represents the transition voltage between the charging and discharging, D_{ref} is the ISI period of the reference spike train, and t_n is determined by V_3, τ_n, and τ_p, which can be expressed as

$$\tau_n = \left\{\begin{matrix} C_r \cdot \left(\frac{W_{T4}}{L_{T4}}\right) \cdot I_{oN} \cdot e^{\frac{V_{pq}}{nkt}} \\ C_r \cdot \left(\frac{W_{T1}}{L_{T1}}\right) \cdot I_{oP} \cdot e^{\frac{V_{pq}}{nkt}} \end{matrix}\right\}, \tag{14}$$

where I_{oN} and I_{oP} are determined by the physical process of NMOSs and the PMOSs, respectively. By regulating control voltages of V_p and V_n, the same charging and discharging speeds could be achieved. In other words, the output voltage of the decoder is only determined by $V_1 - V_2$.

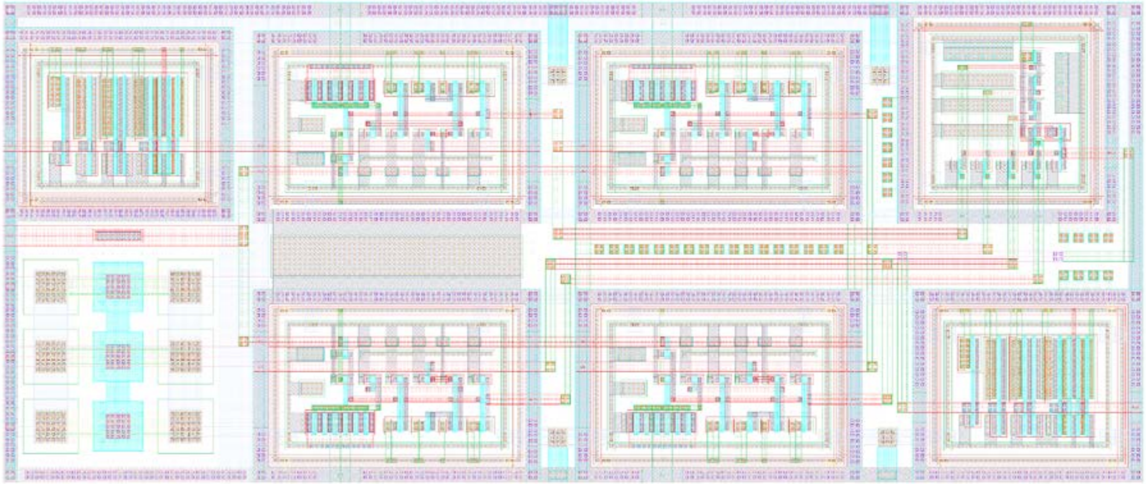


Figure 27. Layout of inter-spike-interval decoder with spike-time-dependent plasticity methodology.

3.4 Delayed Feedback Reservoir System

3.4.1 Delayed feedback reservoir system design and integration

Applications of the reservoir computing include chaotic dynamic predictions [30], character recognition [31, 32], speech recognition [33, 34], and the generation and prediction of chaotic time series [29]. In order to more closely mimic the mammalian brains, delay should be taken into consideration, which has been successfully implemented in the DFR system. In this context, the reservoirs function as time-delayed recursive networks that use feedback as a short-term dynamic memory for processing time-series input signals. Delay systems exhibit two prerequisites for reservoir computing: (1) the high dimensionality, and (2) the short-term memory. In such delay systems, the dynamic is influenced by its own output at the previous time step [35].

The photonic implementation of the delay-feedback system has attracted worldwide attention [23, 36, 37]. DFR networks with photonic implementation introduce the phenomenon of optical chaos where complex dynamics could be beneficial for different applications [38]. However, to the best of our knowledge, there is no analog IC implementation for the spike-time-dependent DFR system in the literature.

One of the simplest possible delay systems consists of a single nonlinear node whose dynamics are influenced by its own output at the previous time step. Such a system is easy to implement because it comprises only two elements, a nonlinear node, and a delay-feedback loop. The delay-feedback loop goes through a number of virtual nodes. Each virtual node is separated by an equidistant delay time, τ. Each virtual node holds the delayed version of the previous node's output in time $\tau = \frac{\theta}{N}$, where θ is the total time constant of the delay-feedback loop, and N represents the number of virtual nodes. The dynamic characteristic of the delay system can be influenced by simply changing the feedback strength or the delay interval θ and τ. Numerical results show that the DFR system has an approximately identical performance compared to the traditional reservoir computing design.

In general, our DFR system was constructed with a single nonlinear neuron, a temporal encoder, and a delay-feedback loop, as depicted in Figure 28. During the operation, analog input signals are first nonlinearly projected onto a higher dimensional space through the nonlinear neuron, followed by accumulating in the sensing capacitor within the temporal encoder, such that post-neuron signals are represented by a temporal spike train, enabling the spiking information processing capability. The encoded temporal spike train travels along the dynamic feedback loop and eventually integrates with the next incoming input data. Such a feedback network creates a short-term memory, establishing connections within the context of data.

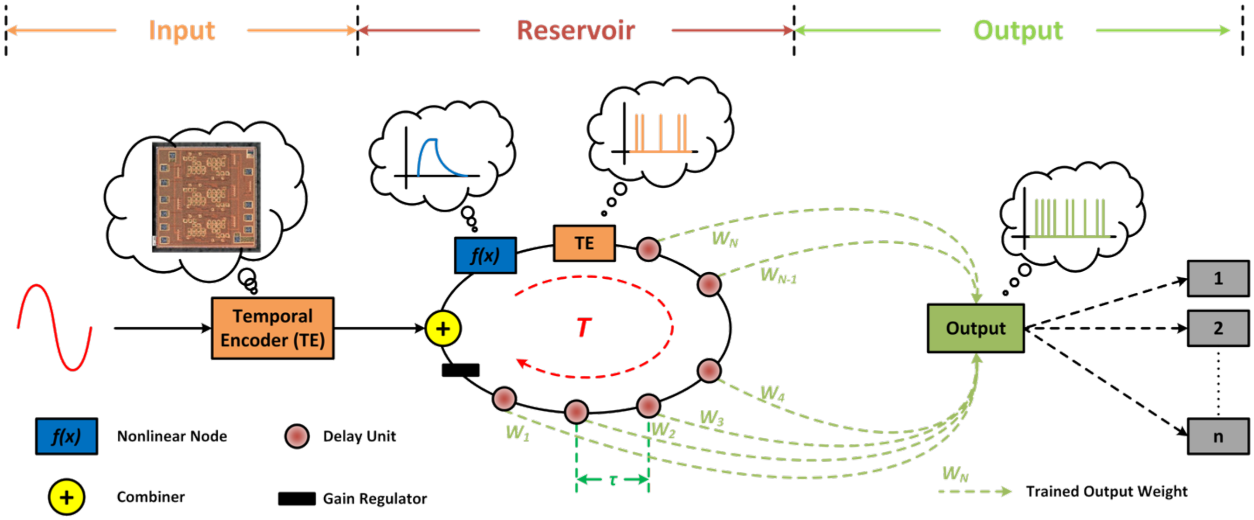


Figure 28. System architecture of delay-feedback reservoir (DFR) system.

3.4.2 Circuit fabrication and testing with advanced CMOS technology

Our first-generation prototype of the spike-based DFR system was submitted for fabrication through MOSIS using the standard GF 130nm CMOS technology in May 22, 2017. This chip contained 12 DFR system modules and was separated into two sections. Each section can be biased and measured individually with 2 separated groups of input/output (I/O) pins. Moreover, since device mismatch might occurred during the fabrication process, 4 different design floor plans of the DFR system were implemented in this chip. In order to reduce the usage of bonding pads within the limited chip area, outputs from all DFR systems within the same section are shared with 1 output pin. To prevent the interference between each output signal, a 6-to-1 multiplex was implemented to select the specific DFR system module that needs to be measured. In addition to DFR system modules, an individual nonlinear neuron, as well as the temporal encoder, were also included in this chip for individual performance testing. Figure 29 demonstrates the layout of our first-generation prototype of the spike-based DFR system.

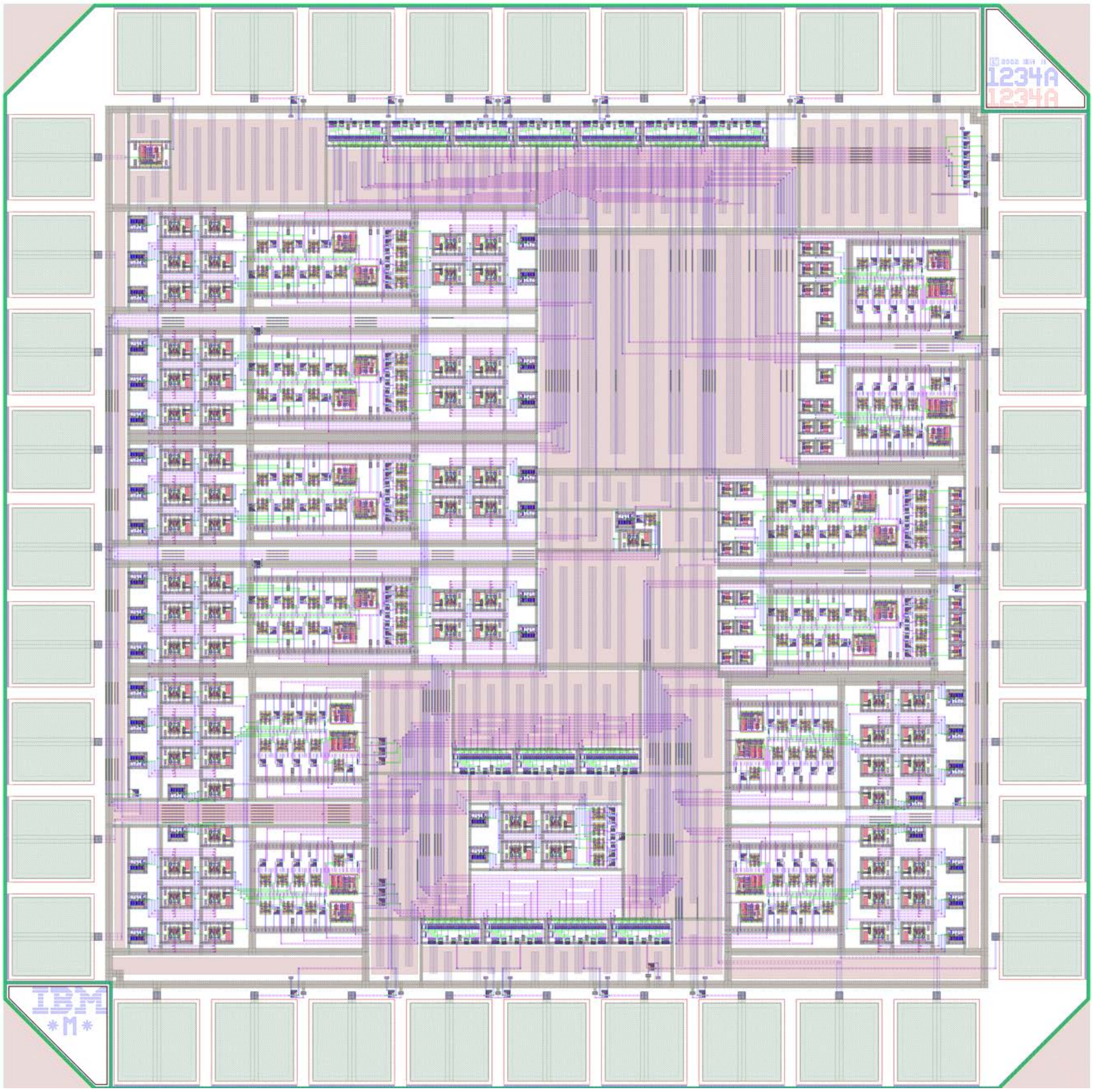


Figure 29. Layout of the first-generation spike-based DFR system.

The improved second-generation spike-based DFR was submitted for fabrication through MOSIS using the standard GF 180nm CMOS technology in December 8, 2019. This chip contains 2 hybrid neural network (HNN) cores, 4 single-layer DFR system modules, and a global control system. For individual performance testing, an individual nonlinear neuron, a temporal encoder, an ISI decoder, and a spiking neuron are implemented in this chip. The layout of our second-generation of spike-based DFR system is shown in Figure 30.

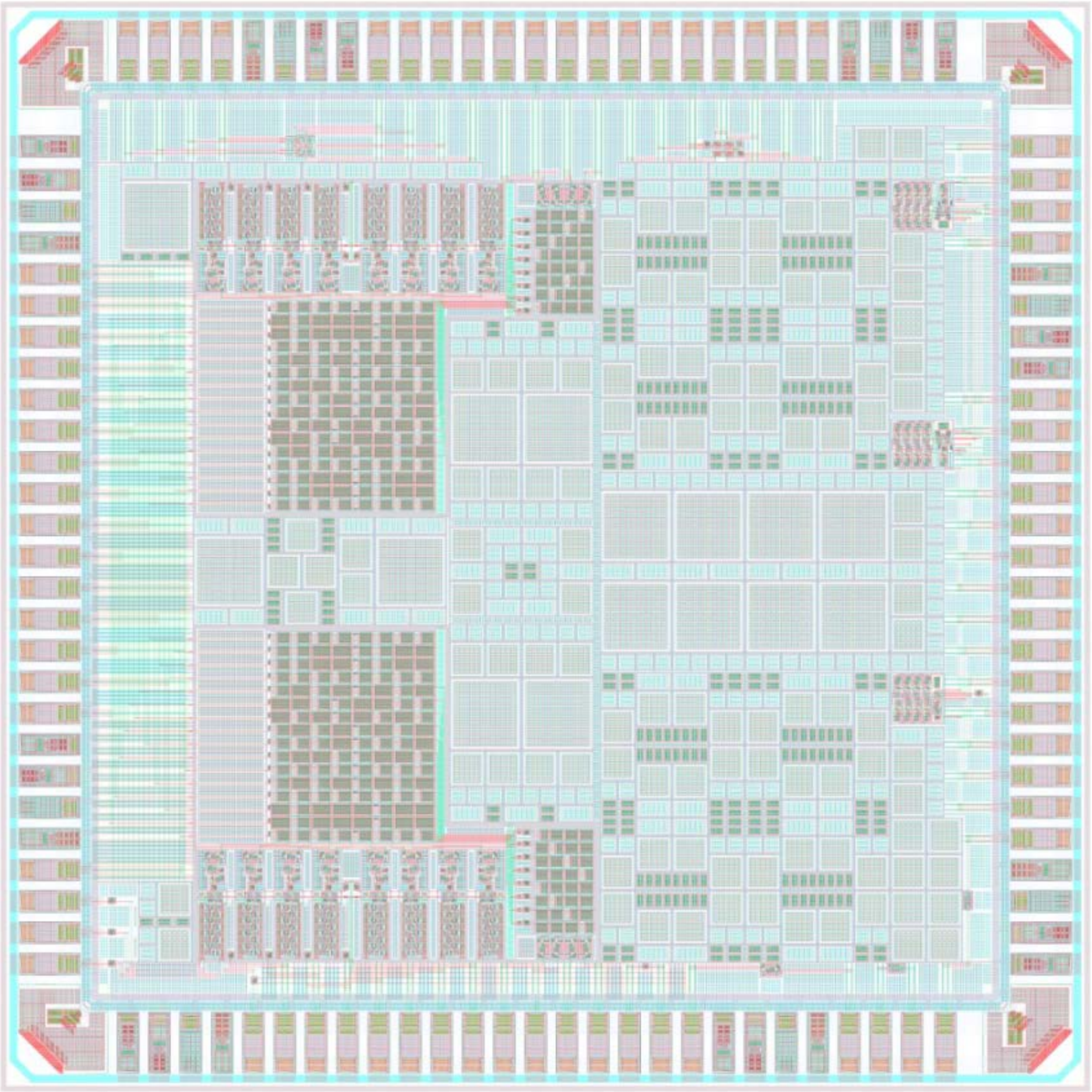
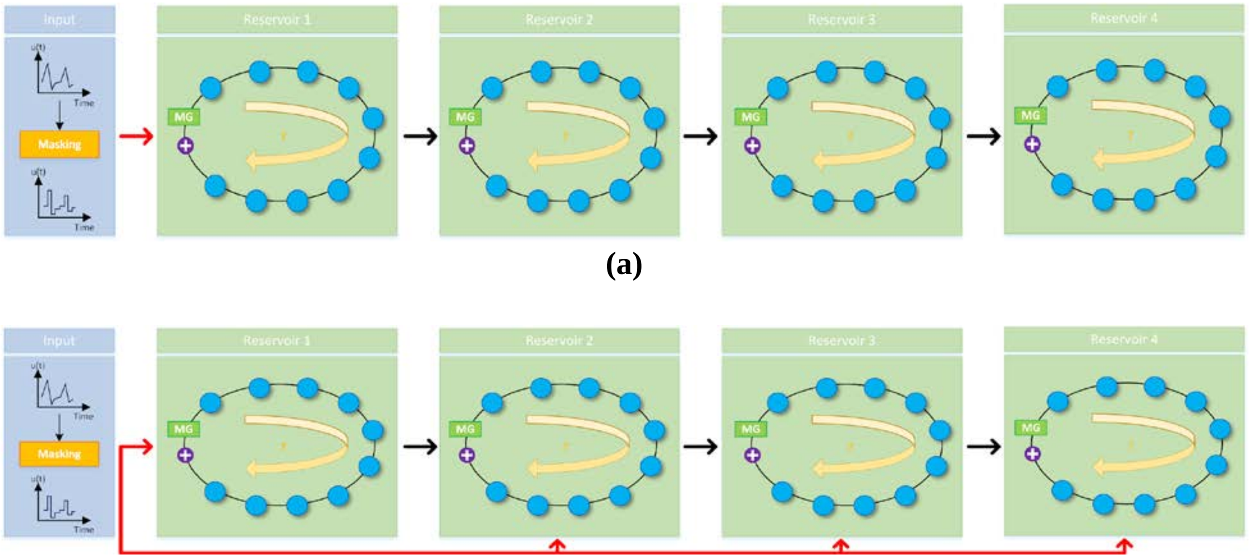


Figure 30. Layout of the second-generation spike-based DFR system.

3.4.3 Multi-layer delayed feedback reservoir system

Recently, the field of deep learning has attracted worldwide attention due to its hierarchical architecture that allows more efficient performance than a shallow structure, not only on accuracy but also the processing speed [39]. The superior performance is a result of its intrinsic deep structure. DNNs are constructed by multiple layers working in a fashion of processing pipeline [40]. The deep learning architecture has proven to have exceptional performance in high-dimensional data that is applicable to many fields, ranging from business to science [41]. Many performance records are broken by deep learning architectures in applications of image classification and handwritten character recognition [40-42]. The depth is generally defined as stacking multiple hidden layers in between input and output layers. This could either be defined in the time-domain or in the space-domain.

RNNs could be defined as a variant of DNNs. For RNNs, the depth arises from inherent recurrent connections, which lead to depth-in-time. However, the training process of such neural networks is considered complex and time-consuming. In an endeavor to reduce the complexity of RNNs, the reservoir computing architecture was proposed in the field of machine learning. In DFR systems, the depth-in-time computing structure arises from the delayed signal that combines with the new input. However, for both RNNs and DFRs, a single reservoir does not create any depth-in-space computing structure. Similar to stacked FNNs in the deep learning field, the depth-in-space computing structure can also be achieved by stacking multiple reservoir layers between input and output layers.



(a)

(b)

Figure 31. Illustration of deep DFR models in (a) deep DFR and (b)MI-deep DFR.

Along with the analog implementation of the DFR system, we investigated the possibility of merging the deep learning and the DFR system. Two deep DFR structures, deepDFR, and multiple-input (MI-deepDFR), are introduced. In the deepDFR model, the output from the previous layer was injected into the successive reservoir layers. The governing equation can be expressed as

$$\dot{x}_1^l = -x_1^l(t) + f\left(x_1^l(t-\tau), I_1^l(t), \theta\right), \tag{15}$$

where $x_1^l(t)$ is the state at l-th layer, θ is the time interval between each virtual node, and $f(\quad)$ is the nonlinear mapping function by using the Mackey-Glass nonlinear activation function as shown below

$$f\left(x_1^l, I_1^l\right) = \frac{a\left(x_1^l + I_1^l\right)}{1 + \left(x_1^l + I_1^l\right)^n}, \tag{16}$$

where I_1^l is the input signal that injects to each layer for deepDFR model, where the input signal is organized as

$$I_1^l = \left\{\begin{matrix} m \cdot u_1(t) & l = 1 \\ x_1^{l-1}(t) & l > 1 \end{matrix}\right\}, \tag{17}$$

where $u_1(t)$ is the original input signal, m is the masking operation, $x_1^{l-1}(t)$ is the output state from the previous layer. This topology of the deepDFR model is illustrated in Figure 31(a).

The other deep structure of the DFR system is similar to the deepDFR, but the input is injected into each layer together with the output state from the previous layer. By adding external input to each reservoir layer, each layer would have a more recent memory of the input signal. This might be useful when carrying out prediction tasks. The governing equation for the MI-deepDFR can be expressed as

$$\dot{x}_2^l = -x_2^l(t) + f\left(x_2^l(t-\tau), I_2^l(t), \theta\right), \tag{18}$$

where $x_2^l(t)$ is the state at l-th layer, θ is the time interval between each virtual node, and $f(\quad)$ is the nonlinear mapping function by using the MG nonlinear function as shown below

$$f\left(x_2^l, I_2^l\right) = \frac{a\left(x_2^l + I_2^l\right)}{1 + \left(x_2^l + I_2^l\right)^n}, \tag{19}$$

where I_2^l is the input signal that injects to each layer for MI-deepDFR model, where the input signal is organized as

$$I_1^l = \left\{\begin{matrix} m \cdot u_2(t) & l = 1 \\ [m \cdot u_2(t) \quad x_2^{l-1}(t)]^T & l > 1 \end{matrix}\right\}, \tag{20}$$

where $u_2(t)$ is the original input signal, m is the masking operation, $x_2^{l-1}(t)$ is the output state from the previous layer. This topology of the MI-deepDFR model is illustrated in Figure 31(b).

4.0 RESULTS AND DISCUSSIONS

4.1 Performance Analysis of Mackey-Glass Nonlinear Node

As depicted in Figure 32, it can be observed that the nonlinear correlation between input and output signals was successfully achieved. Similar to the nonlinear characteristic of the ideal Mackey-Glass function, it can be observed that the nonlinearity of the transfer function in the circuit implementation can be regulated by controlling the CP current, I_{cp}. To demonstrate such feature, the CP current, I_{cp}, is altered to achieve various nonlinearity of the transfer function. From Equation (1), as n increases, the nonlinearity of the transfer function rises accordingly. The same characteristic can be observed in the circuit implementation. With increasing the CP current, the nonlinearity of the circuit's function can be regulated, as plotted in Figure 33.

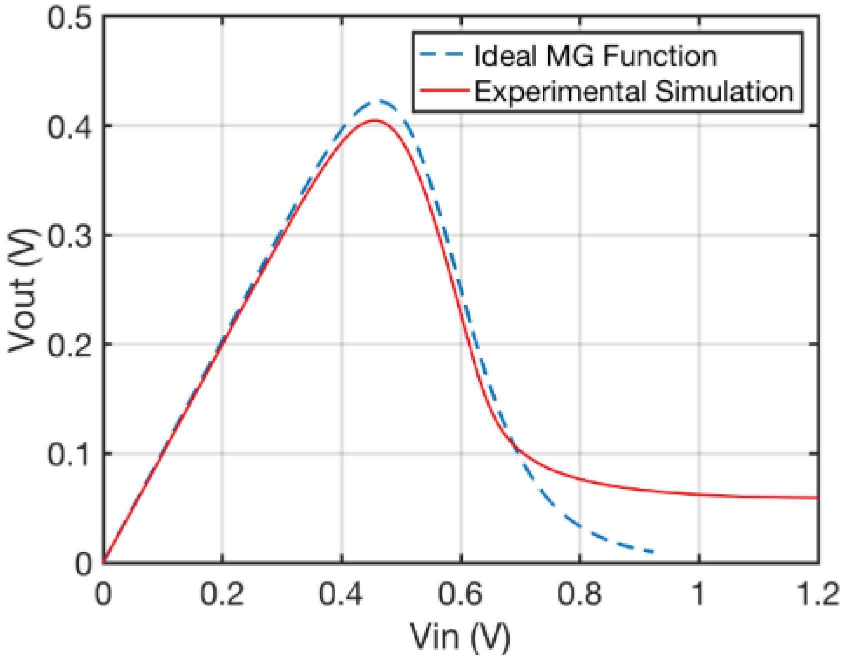


Figure 32. Nonlinear regime of the ideal Mackey-Glass equation and electronic circuit implementation.

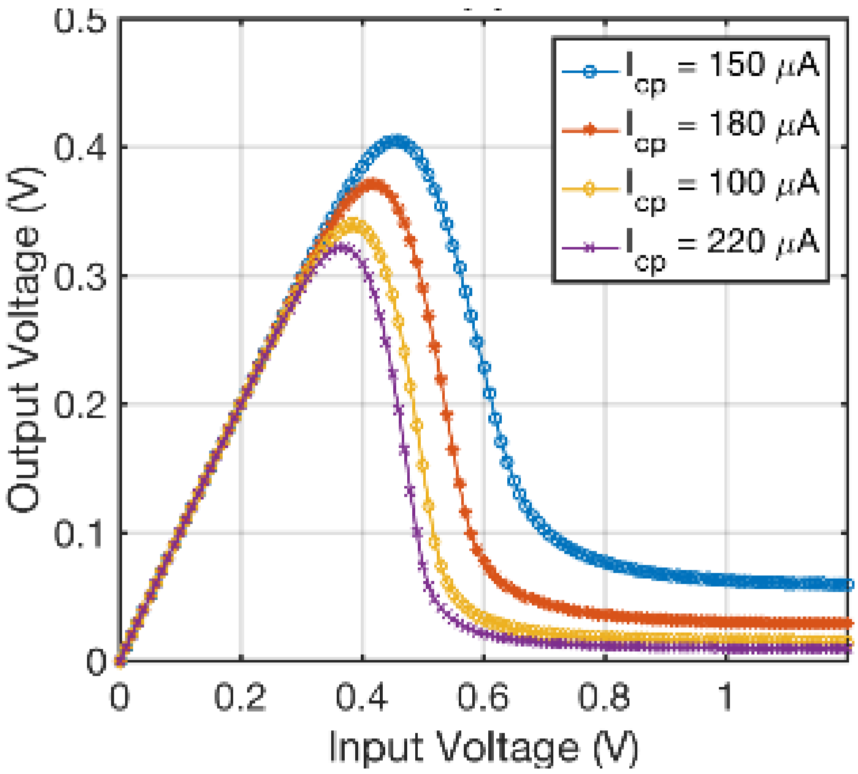


Figure 33. Nonlinearity of the Mackey-Glass equation on the electronic circuit.

To closely examine dynamic behaviors, the solution to the DDE equation carried out. The dynamic behavior of the nonlinear function is modeled by the DDE with varied delay time, as demonstrated in Figure 34.

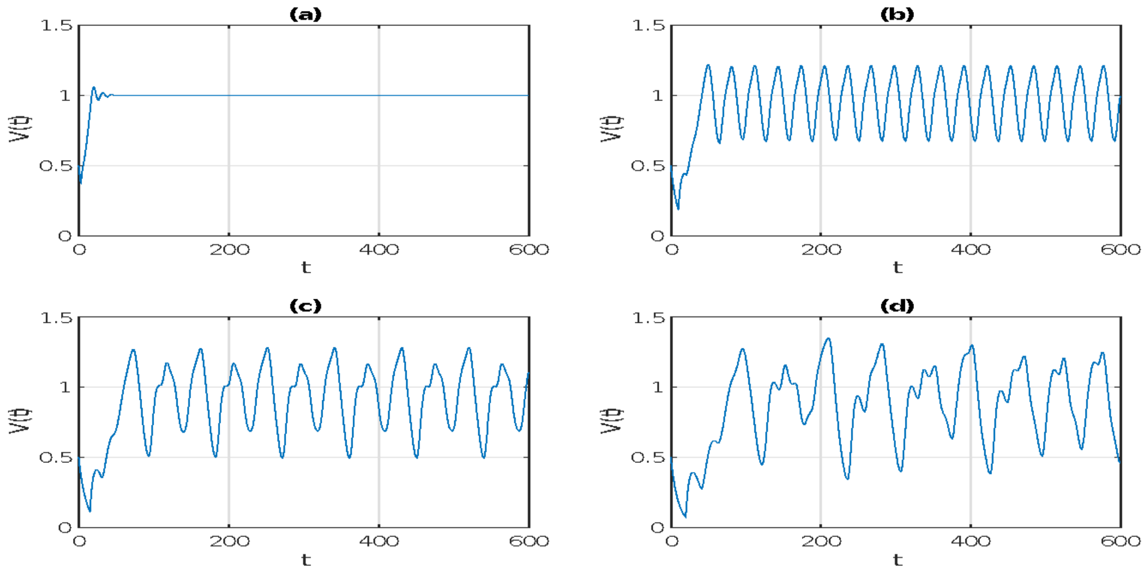


Figure 34. Dynamic behavior of the Mackey-Glass nonlinear equation when (a) τ=3; (b) τ=12; (c) τ=16; (d) τ=20.

As plotted in Figure 34, the solution converges to an equilibrium state when the delay is small. The dynamic behavior varies accordingly as the delay starts to increase. With the increasing time delay, the dynamic alters from periodic to chaotic as shown in Figure 34(a) to (d).

The phase portrait is a representation of solutions, tracing the path of each particular solution. It is a graphical tool to visualize how the solution of a given system of differential equations would behave in the long run. In other words, the phase portrait is a tool to track the dynamic behavior of a system's solutions. By varying the time delay, the phase portraits are illustrated in Figure 35. It can be observed that as the delay increases, the dynamic behavior varies from order to the edge of chaos and even further to completely chaotic.

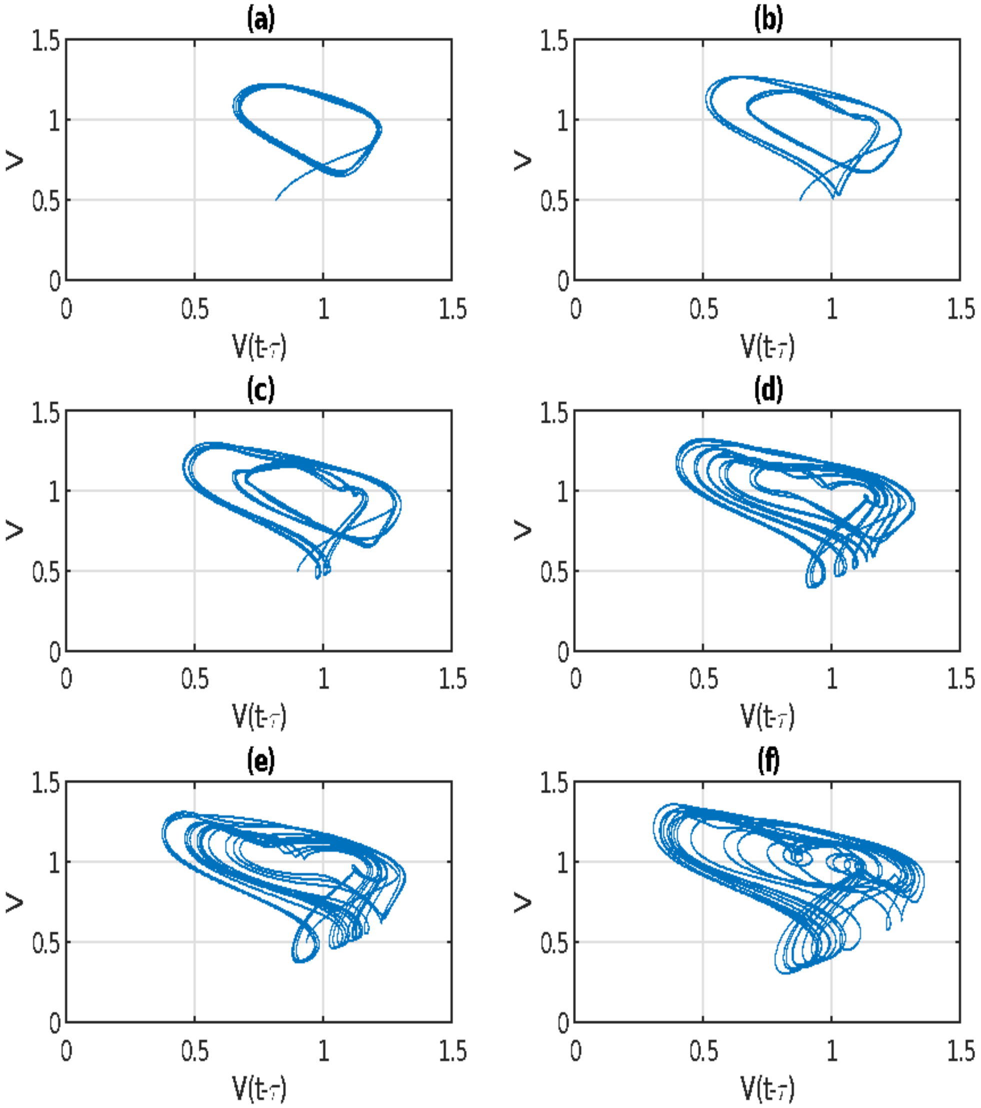


Figure 35. Phase portrait of the dynamic system in (a) τ=12; (b) τ=14; (c) τ=16; (d) τ=18; (e) τ=20; (f) τ=22.

The system robustness of the nonlinear transformation is evaluated through the Monte-Carlo simulation by introducing the process variation with 500 sampling points at the room temperature. In this task, the input signal of the nonlinear node was set to be 0.8V. As depicted in Figure 36, the average offset for the nonlinear node was 6mV with a standard deviation of 8mV. Simulated results indicate that 100% of data points lie within a band of 3σ region.

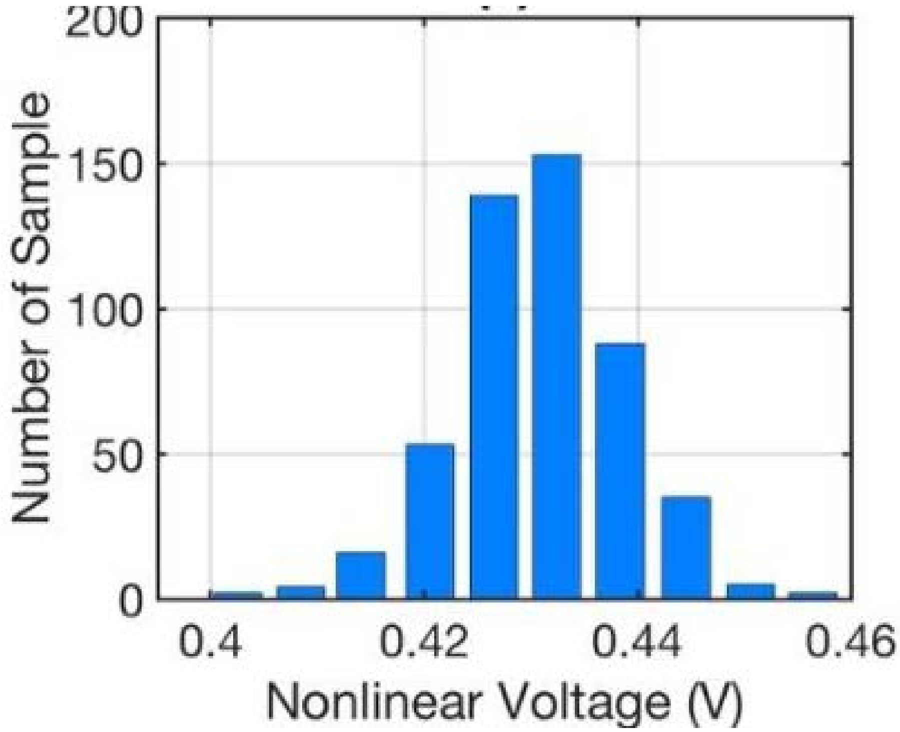


Figure 36. Simulated system robustness with process variation in nonlinear transformation.

The temperature variation was analyzed by simulating the temperature from 0°C to 60°C in the Cadence Virtuoso platform, as plotted in Figure 37. With simulated results in temperature variation, the average error rates remain below 3.5% for the nonlinear transformation if the temperature is below 32°C. As the temperature increases, error rates increase by up to 15.5%.

Measured normalized errors of the nonlinear transformation were evaluated through 20 sampling points at room temperature, as depicted in Figure 38. In this task, the input signal of the nonlinear node was set to be 0.8V. Normalized errors were evaluated by examining the difference between the predicted value from the simulation and the actual value from the measurement. With the testing results in the measured system robustness, the average normalized error for the nonlinear transformation was 1.97%.

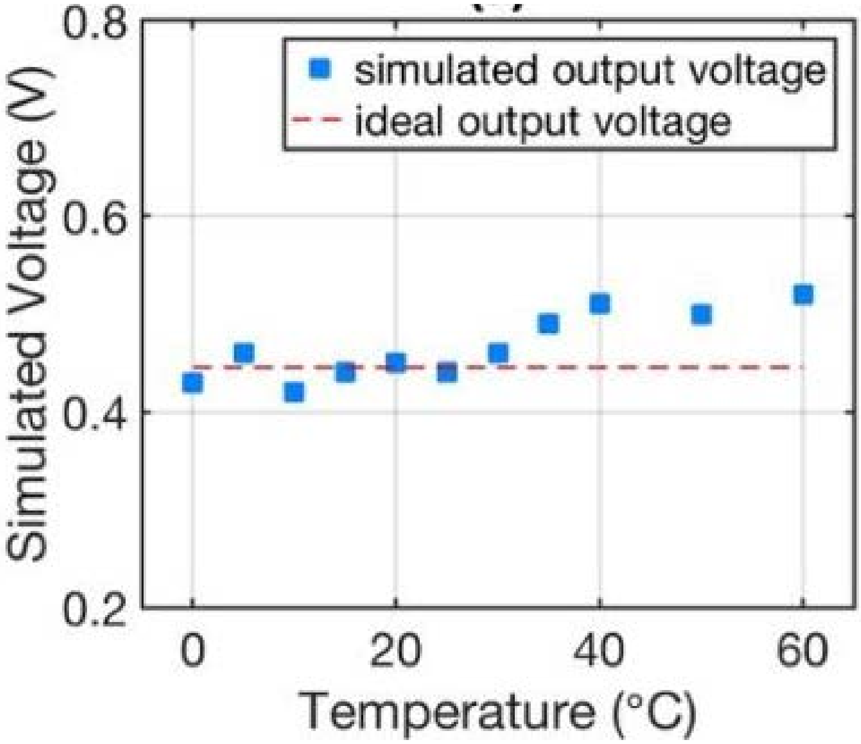


Figure 37. Simulated system robustness with temperature variation in nonlinear transformation.

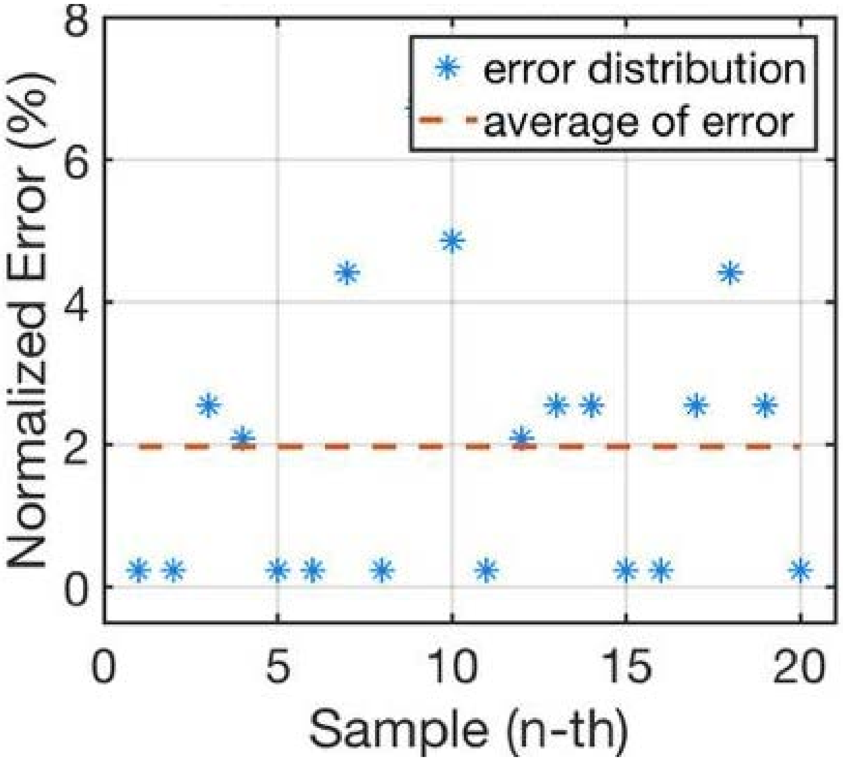
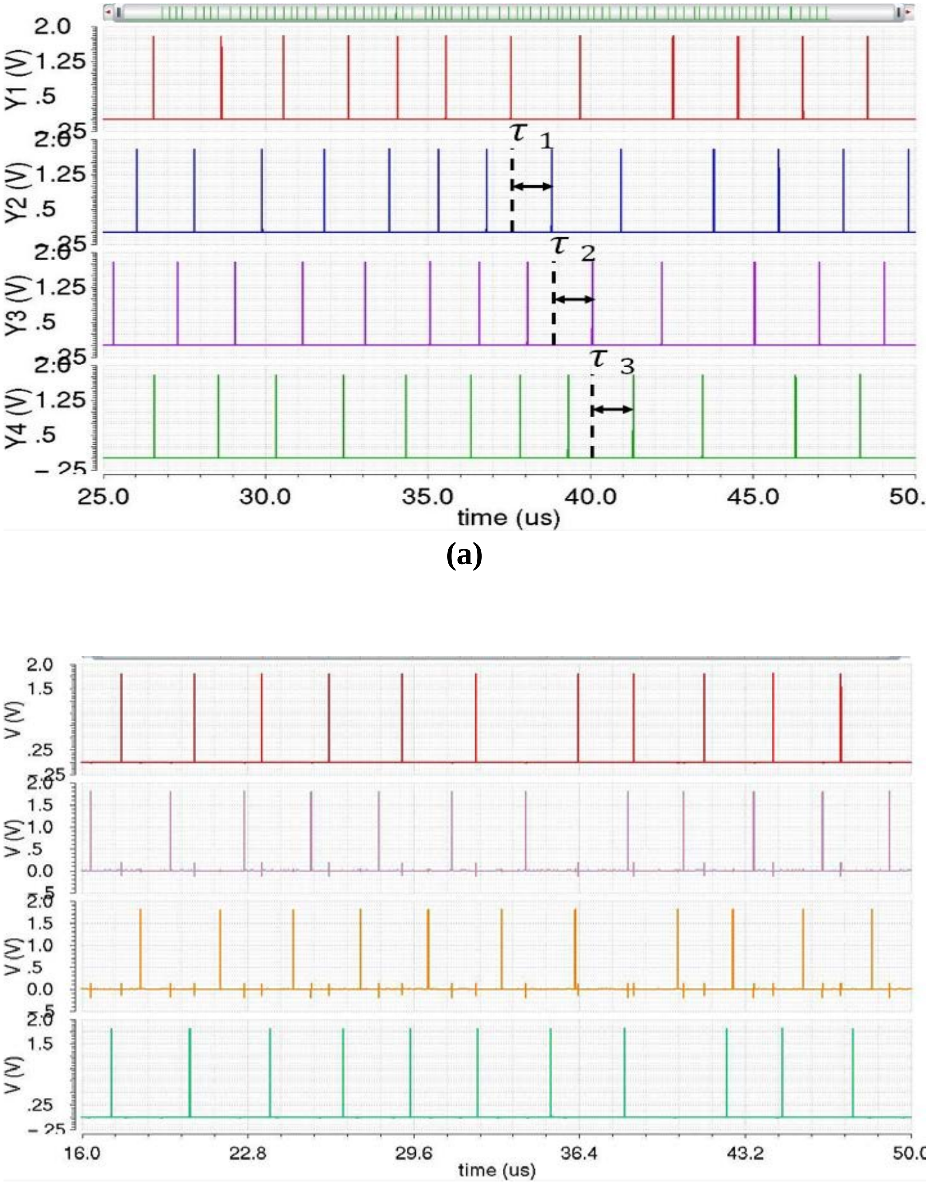


Figure 38. Normalized error in nonlinear transformation.

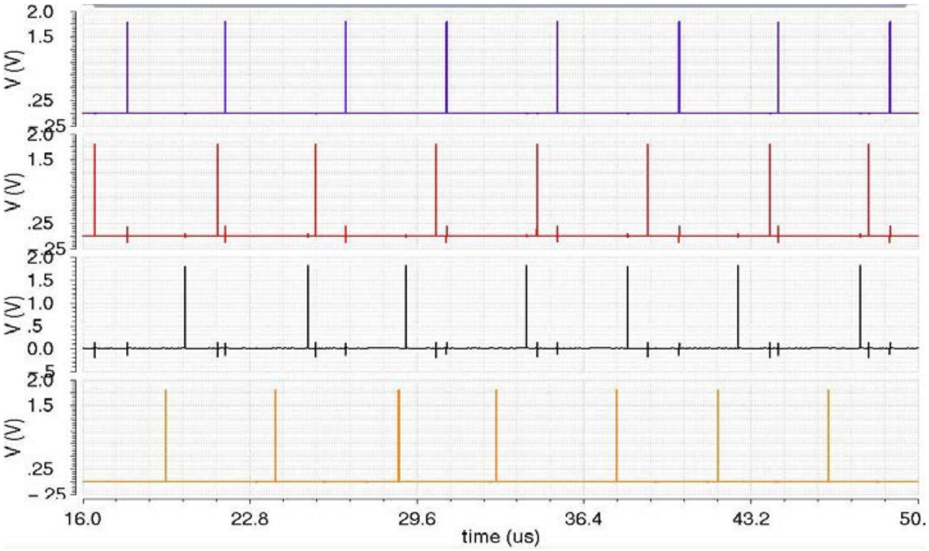
4.2 Performance Analysis of Static Feedback Loop

Figure 39 demonstrates a four-stage delay loop whereby the output spike trains are illustrated. The V_{th} was fixed at 1V and the $I_{ex}-I_{leak}$ is fixed in 0.1μA, which is equivalent to 10MΩ resistance. Therefore, the proposed delay unit could achieve large delay time with a very small capacitor. Hence, the dynamic of the system can be varied from order to edge of chaos by tuning the delay constant with very small capacitance and resistance value.



(a)

(b)



(c)

Figure 39. Output spike train with different delay time in (a)1.27µA, (b) 2.03µA, and (c) 3.69µA.

In the DFR system, the dynamic of the system can be varied from the order to the edge of chaos by controlling the total delay time along the delay loop. To demonstrate the delay behavior of the system, the delay time, τ_{delay}, of the I&F delay neuron is altered to achieve a large dynamic range of controllable delay time. As plotted in Figure 40, the delay time can be regulated from 180ns to 1.5µs by controlling the excitation current from 50nA to 300nA.

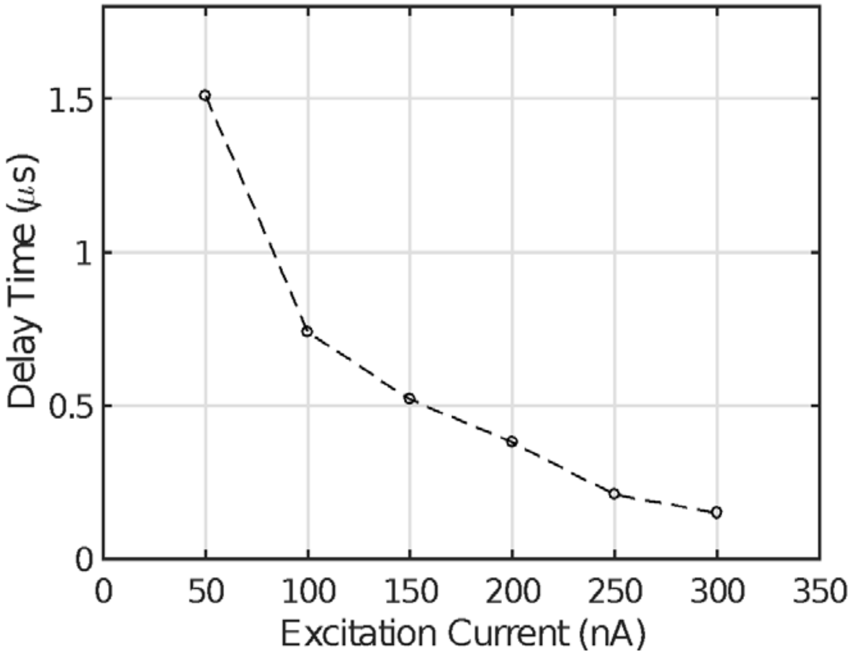


Figure 40. Controllable delay time.

To acquire 1.5μs delay via the traditional Resistor-Capacitor delay element, such a system requires a 100kΩ resistor and a 15pF capacitor, resulting in a large design area. The introduced I&F delay neuron overcomes this drawback by regulating the equivalent input impedance of the circuit. By injecting 50nA excitation current into the delay unit, the equivalent input impedance reaches 25MΩ, thus, a large delay time can be achieved with an extremely small capacitor. Compared to the traditional Resistor-Capacitor delay element, as illustrated in Figure 41, that is built upon a large design area of resistors and capacitors, our I&F delay neuron has the capability to process the spiking information directly with a superior dynamic range of controllable delay time and a small design area.

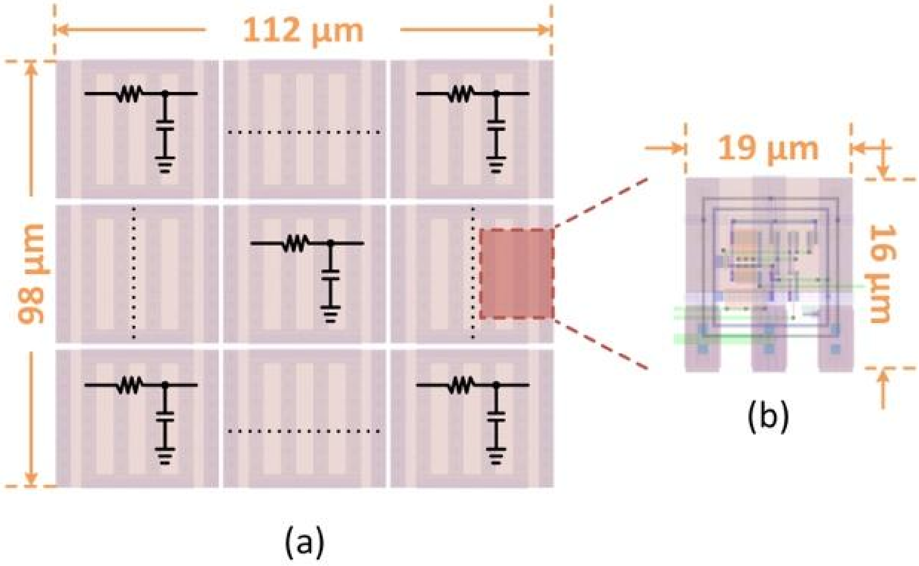


Figure 41. Design area illustration in (a) Resistor-Capacitor delay element, and (b) introduced I&F delay neuron.

The system robustness of the delay regulation was evaluated through the Monte-Carlo simulation by introducing the process variation with 500 sampling points at the room temperature. In this task, the input signal of the I&F delay neuron was set to be 250nA. As depicted in Figure 42, the average offset for the I&F delay neuron was 1.99ns with a standard deviation of 2.83ns. Simulated results indicate that 100% of data points lie within a band of 3σ region.

The temperature variation was analyzed by simulating the temperature from 0°C to 60°C in the Cadence Virtuoso platform, as plotted in Figure 43. With simulated results in temperature variation, the average error rates remain below 3.5% for the delay regulation if the temperature is below 32°C. As the temperature increases, error rates increase by up to 8.6%.

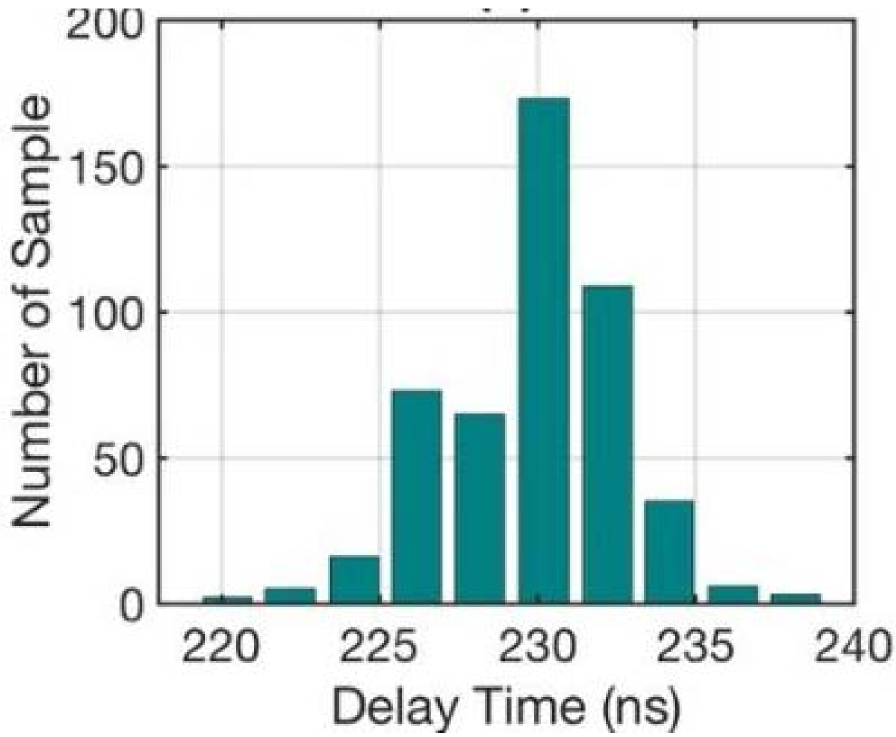


Figure 42. Simulated system robustness with process variation in delay regulation.

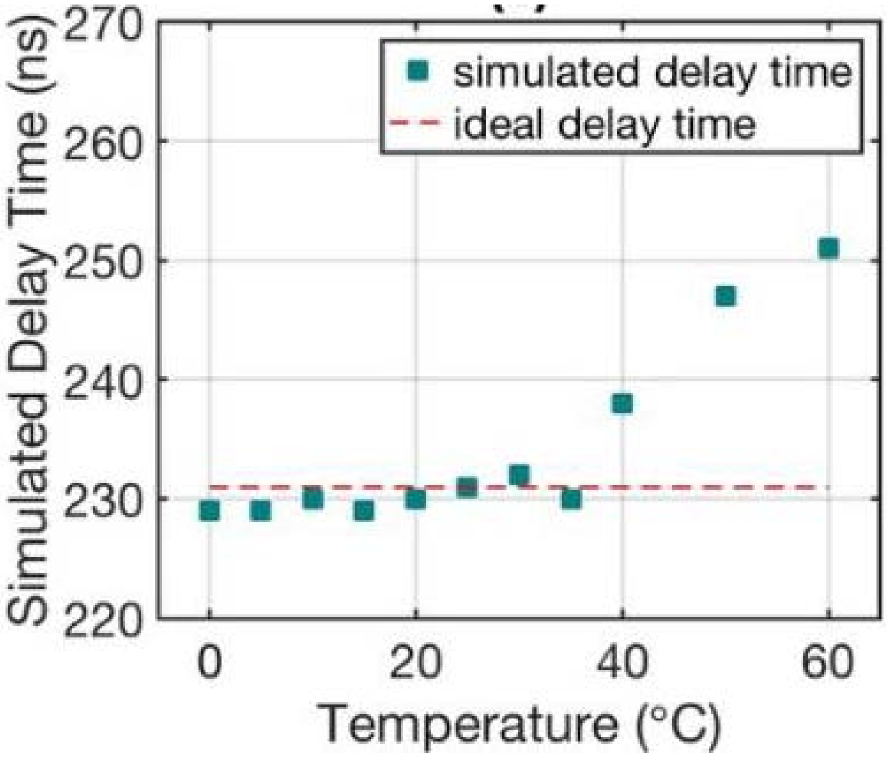


Figure 43. Simulated system robustness with temperature variation in delay regulation.

Measured normalized errors of the I&F delay neuron are evaluated through 20 sampling points at room temperature, as depicted in Figure 44. In this task, the input signal of the I&F delay neuron was set to be 250nA. Normalized errors were evaluated by examining the difference between the predicted value from the simulation and the actual value from the measurement. With the testing results in the measured system robustness, the average normalized error for the delay regulation was 3.48%.

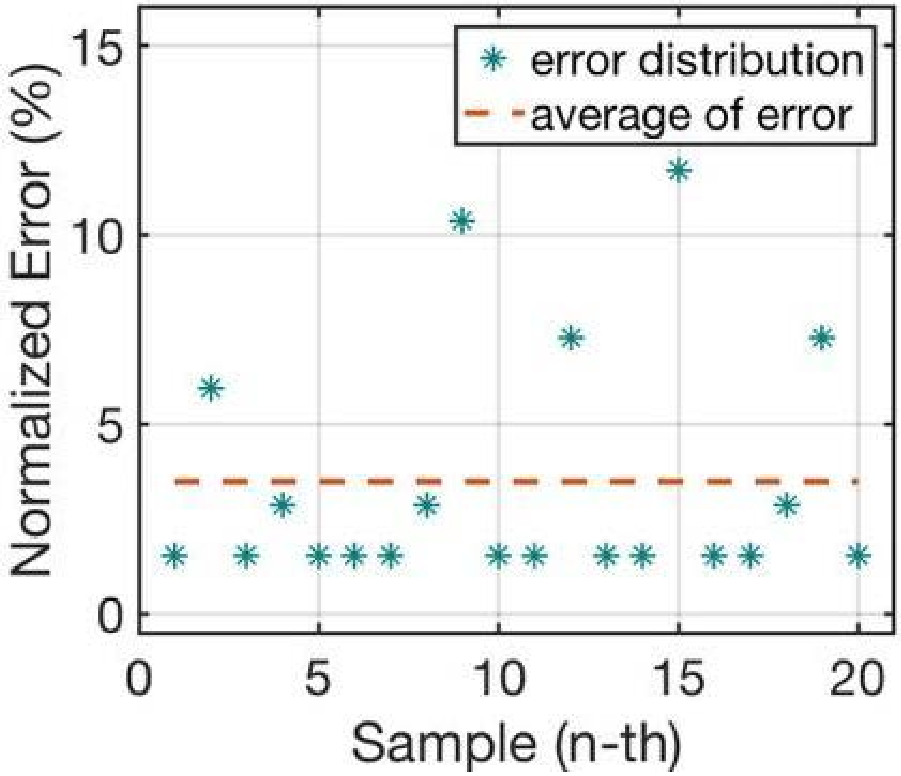


Figure 44. Normalized error in delay regulation.

4.3 Performance Analysis of Inter-spike-interval Decoder with Spike-timing-dependent Plasticity Methodology

For an ISI spike processing module, one of the challenges was to design a robust and efficient decoder. There are two main steps in the encoding process: 1) convert the signal's amplitude into a latency spike, and 2) integrate latency spikes into an ISI spike train. Once the encoding process is accomplished, the numerical dimension (amplitude) of an input signal is transformed into a temporal dimension (time interval). Thereby, the purpose of the ISI decoding scheme is to transform the encoded temporal dimension-based ISI spike train back to a numerical dimension-based analog value, or a level-based signal.

In [43], a direct decoder design was discussed in which the key idea was to measure time intervals directly and transform these intervals into a level-based signals. However, this decoder adopts Sample/Hold sub-circuits to detect the position of spikes, which occupy large active design areas. Most importantly, as the number of spikes of an ISI temporal code increases, more Sample/Hold sub-circuits are needed. In other words, the reliability of the decoding operation has become a critical challenge when scaling up the information density of an ISI spike train. In order to overcome these issues, a decoding strategy, which includes the capability to handle any information density of an ISI spike train with a constant circuitry scaling factor, is needed. Inspired by the synapse updating technique, the STDP principle can be adopted to implement such a decoder for an ISI spike train.

One of the differences between our ISI decoding scheme and traditional STDP decoding scheme is the output scale, also known as the amplitude level. The traditional STDP decoding could only generate bi-stable status, representing 1 and 0. However, our ISI decoder could map the ISI spike train into output with multiple scales.

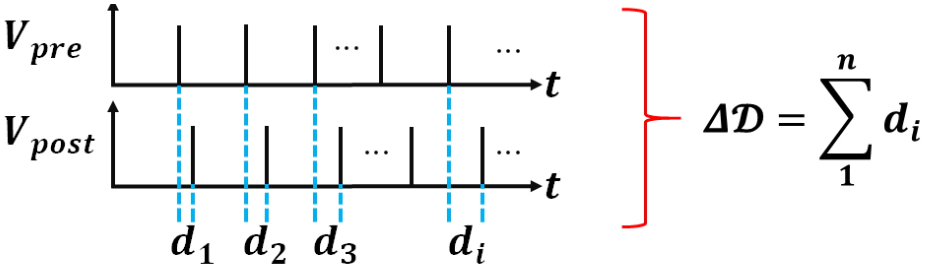


Figure 45. Operating principle of inter-spike-interval decoder.

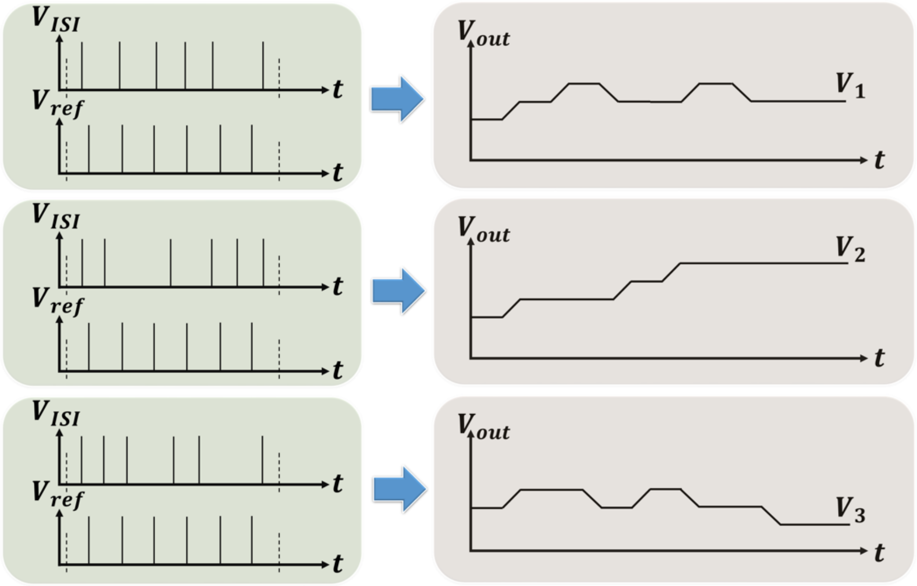


Figure 46. Signal flow of the inter-spike-interval decoder.

As shown in Figure 46, by applying a uniform reference spike train on the post-spike port and information carrier spike train on the pre-spike port, the final output from the ISI decoder would generate different voltages (i.e., V_1, V_2, V_3. Under such configuration, there is no need to adopt additional Sample/Hold modules after the ISI decoder. Furthermore, the decoding speed increases significantly compared with the decoder in [43]. This is mainly because the length of an ISI spike train is constant and much shorter compared to the rate encoded spike train. Moreover, compared to the latency spike train, multiple spike trains have much higher error tolerance when the spike-missing is considered.

Unlike the bi-stable STDP application, our introduced ISI decoder could achieve multiple scales by applying an ISI spike code. The transient simulation results are illustrated in Figure 47.

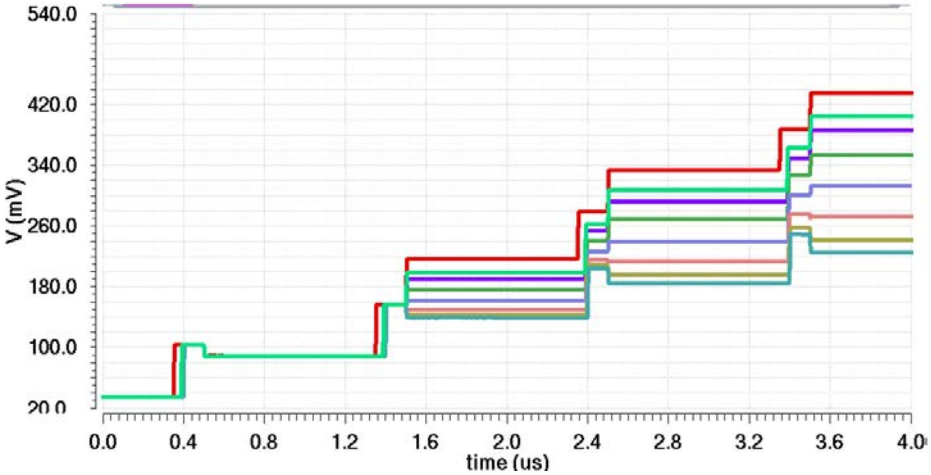
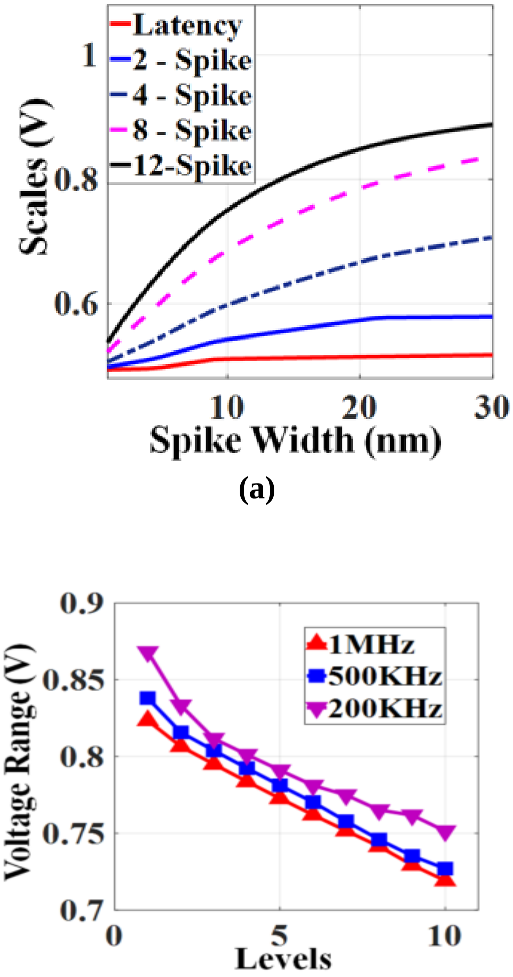


Figure 47. Transient response of inter-spike-interval decoder.



(a)

(b)

Figure 48. Simulation results of (a) the relationship between spike width and output signal’s scales; (b) the linearity of the output signal.

In order to ensure that the output signal can be easily processed by other discrete systems, the linearity and the signal range are two important specifications that need to be considered. Furthermore, the tolerance on the pulse width of spikes significantly impacts the performance of the decoding operation. Simulated results of our introduced ISI decoder is illustrated in Figure 48.

In this evaluation, a latency spike code together with four other different ISI spike codes were used as testing signals. As shown in Figure 48(a), there are two important properties: 1) the larger ISI spike train could achieve higher output range; 2) for the same scale of ISI spike train, the range of the output signal is influenced by the spike width (i.e., wider width leads to a larger range). For the latency spike train, it has only a 59mV output range even if a 30ns-wide spike is applied. On the other hand, a 12-spike ISI spike train with a 30ns-wide spike train has a 1V output range, which could be detected directly by other discrete systems without the usage of level shifters or amplifiers.

In Figure 48(b), ISI spike trains with 200KHz, 500KHz, and 1MHz were evaluated. Without generality, the 10-level case is adopted. Each level is related to one output voltage. Distributions of the output range are all in linear relationship, especially in higher frequency regime (e.g., 1MHz). Such kind of output signal needs no extra normalization process by the following processor (e.g., training module).

4.4 Performance Analysis of Delayed Feedback Reservoir System

Figure 49 demonstrates the die photo of our fabricated first-generation DFR system. The design area of the whole DFR chip occupies 1.5mm×1.5mm while each DFR module takes up to 175µm×56µm.

From solutions of DDE, the Lyapunov stability analysis shows that the dynamic behavior of a given system could be achieved with a simple delay, and it has been proven that the best performance of the reservoir computing is found when operating at the edge of chaos regime [19, 44, 45]. In [46-49], phase portraits were used to demonstrate the transition of intrinsic dynamic behavior from a theoretical point of view. A phase portrait is a graphics tool to visualize how the solutions of a given system of DDE would behave in the long run.

As demonstrated in Figure 50, plotted phase portraits were obtained from measurements using two signals within the reservoir layer where one of them is collected with time delay. By varying the total delay time within the delay-feedback loop, the dynamic behavior of the system changes from order to chaotic as the delay increases. When the total delay time within the system is maintained around 1µs, the delayed signal, $I(t-\tau)$, repeated traces its initial path even in a long run, resulting in the periodic behavior as plotted in Figure 50(b). As the total delay time within the system increases to 1.4µs, the delayed signal diverges from its initial path but without off-tracking from the equilibrium point even in a long run, resulting in the edge of chaotic behavior as plotted in Figure 50(c).

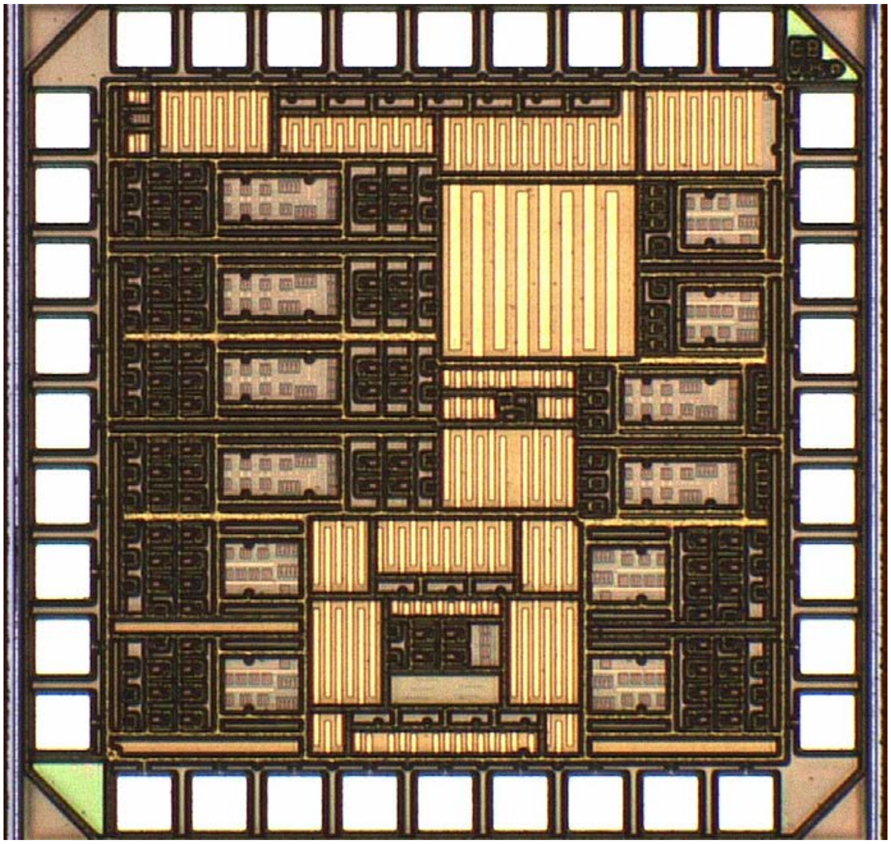
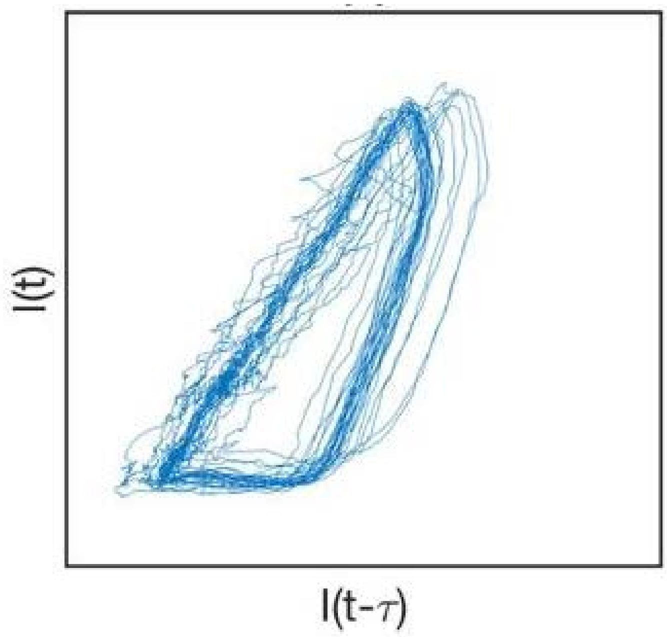
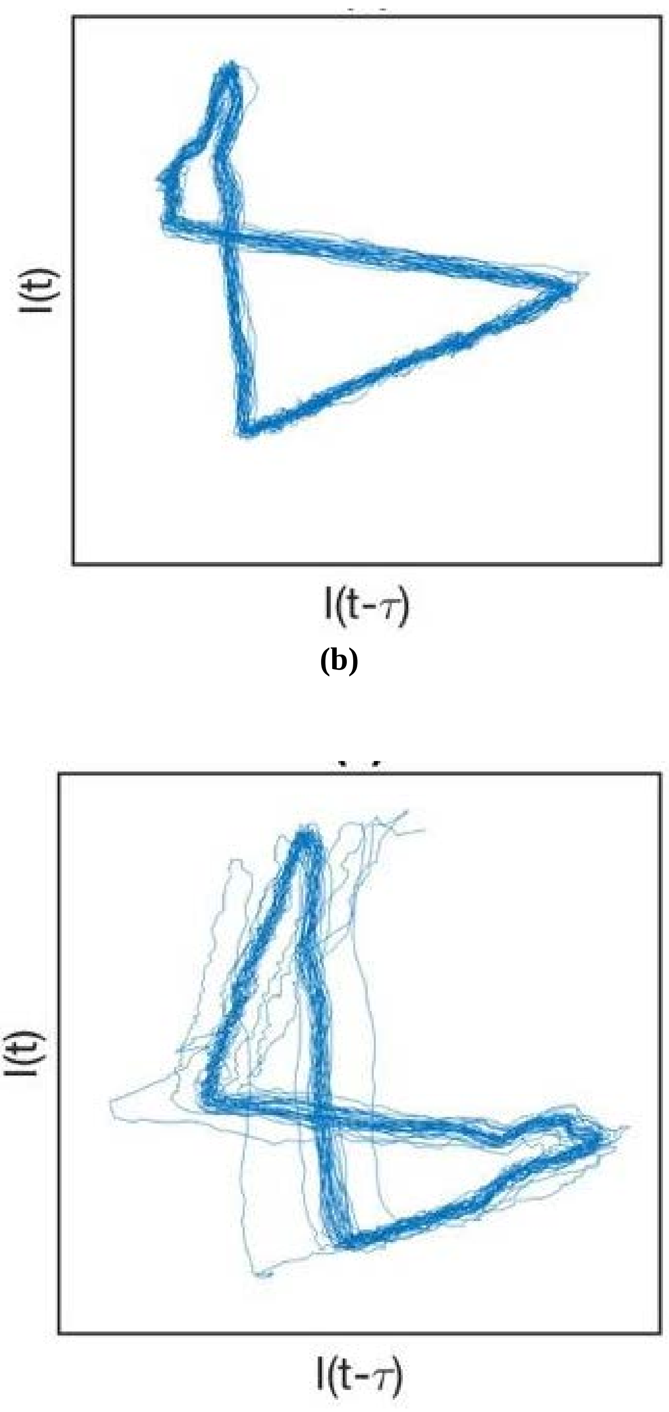


Figure 49. Die photo of the first-generation spike-based DFR system.



(a)



(b)

(c)

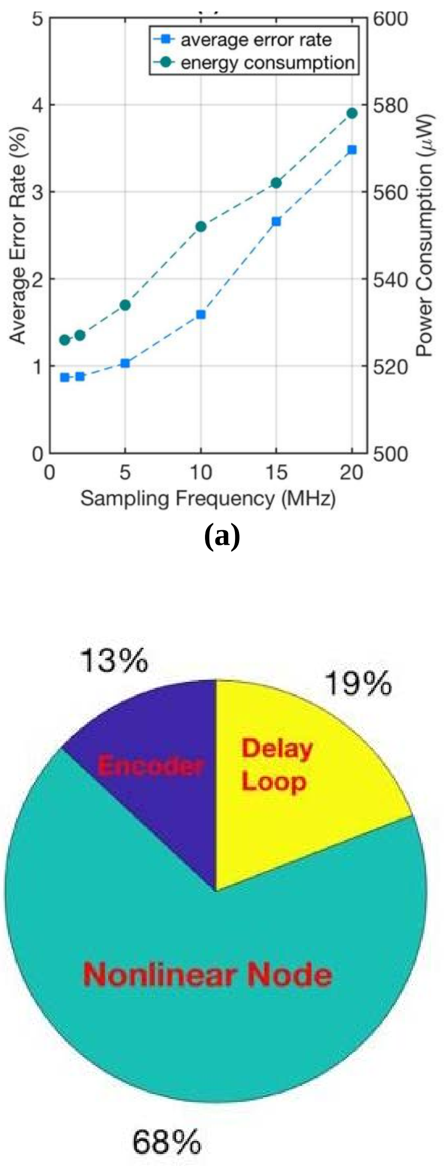
Figure 50. Measured phase portrait of dynamic system in (a) T = 0.64µs, (b) T = 1µs, and (c) T = 1.4µs.

From testing results on dynamic behaviors, one can see that our DFR system does go through a range of dynamic behaviors. It is reasonable to conclude that our fabricated DFR chip successfully implemented the desired functionality of delay and richness of dynamic behavior. This indicates that our analog DFR system closely mimics neurological systems with conduction time delay.

In general, the total power used by the CMOS IC consists of two parts, namely, the static power and the dynamic power [50]. The static power defines the power used when transistors are not in the switching process. The static power is independent to the speed, since input signals remain unchanged. On the other hand, the change of input signals charges or discharges the parasitic capacitor of a transistor and its corresponding loading capacitor, thereby, the dynamic power changes accordingly. Since the rate of charging and discharging processes is proportional to the sampling frequency, the dynamic power is dependent on the frequency, which can be expressed as

$$P_{dynamic} = \left(C_p + C_l\right) \cdot f \cdot V_{DD}^2 \, , \tag{21}$$

where C_p is the parasitic capacitance of a transistor, C_l is loading capacitance, and f is the sampling frequency. As depicted in Figure 51(a), the measured average error rate and simulated power consumption versus the sampling frequency were plotted. When the sampling frequency was less than 5MHz, the average error rate was below 1% with the average power consumption of 529µW. However, as the sampling frequency increases, the average error rate and power consumption increase up to 3.5% and 578uW, respectively. The power distribution of our DFR system with the sampling frequency at 1MHz is illustrated in Figure 51(b). The overall power consumption reaches 526µW; the nonlinear node requires 68% of the total power consumption, while the temporal encoder and the dynamic delayed feedback loop take up to 13% and 19% of the total power consumption, respectively.



(a)

(b)

Figure 51. (a) Error and power analysis in terms of sampling frequency; (b) power distribution of the DFR system with the sampling frequency at 1MHz.

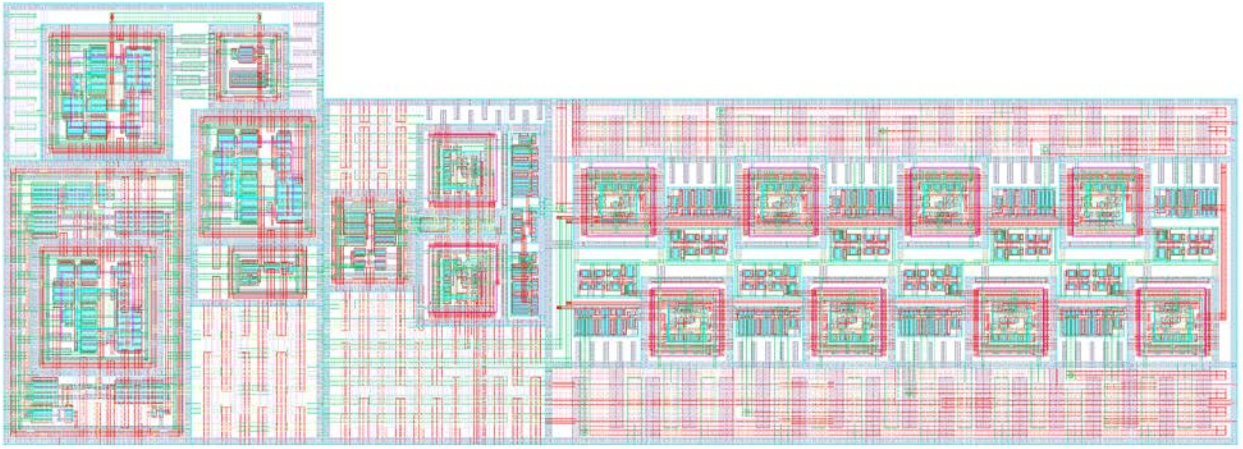


Figure 52. Layout of second-generation DFR system.

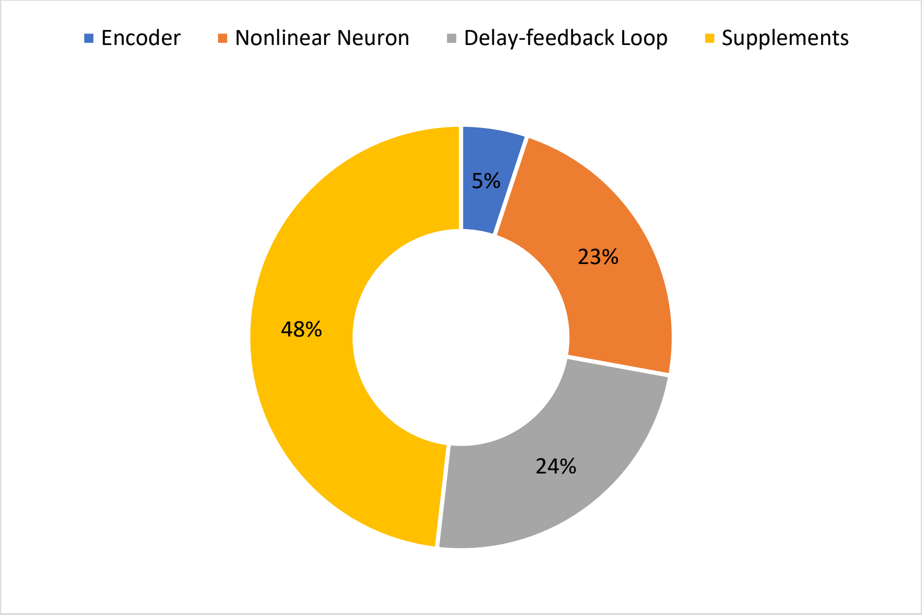


Figure 53. Power distribution of the second-generation DFR system.

The power distribution of the newly designed DFR system was simulated with the sampling frequency at 1MHz and a supply voltage of 1.8V. As plotted in Figure 53, the DFR system consumes 390μW of power, where the nonlinear neuron requires 23% of the total power consumption from the DFR system, the temporal encoder, and the dynamic delay-feedback loop occupy 5% and 24% of the total power consumption from the DFR system, respectively, and the rest were consumed by the supplemental circuitries. Design specifications of the first-generation and second-generation of DFR system are summarized in Table 3.

Table 3. Design Specification of first-generation and second-generation of Delay-feedback Reservoir System.

	1st Generation [46]	2nd Generation
Process	130nm	180nm
Input Dynamic Range	0 ~ 300nA	0 ~ 5µA
Activation Function	sigmoid	MG
Types of Spiking Signal	temporal	Temporal, ISI
Number of Neurons	6	8
Energy Metric	16.69pJ/spike	9.63pJ/spike
Supply Voltage	1.2V	1.8V
Power Consumption (@1MHz)	529µW	390µW

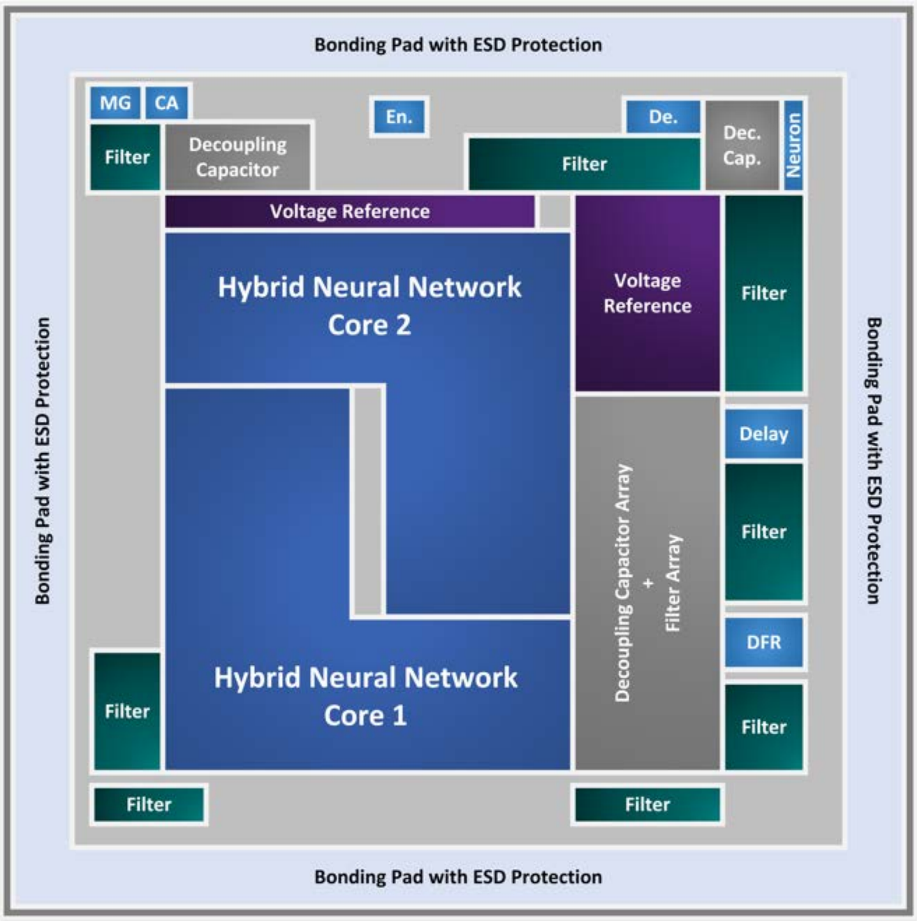

Bonding Pad with ESD Protection
MG
CA
En.
De.
Dec. Cap.
Neuron
Filter
Decoupling Capacitor
Filter
Voltage Reference
Hybrid Neural Network Core 2
Voltage Reference
Filter
Delay
Filter
Decoupling Capacitor Array + Filter Array
DFR
Filter
Filter
Hybrid Neural Network Core 1
Filter
Filter
Bonding Pad with ESD Protection
Bonding Pad with ESD Protection
Bonding Pad with ESD Protection

Figure 54. Floor plan of second-generation DFR chip.

The newly improved DFR system and the hybrid neural network (HNN) was fabricated through the standard GF 180nm CMOS technology in December 2019. The floor plan of the silicon chip is shown in Figure 54. The fabricated chip will be tested at VT's MICS Group using their state-of-the-art specialized lab facilities for integrated circuit testing.

4.5 Chaotic Times Series Prediction Benchmark

To evaluate the precision of the DFR system, a chaotic time series prediction benchmark, the tenth-order nonlinear autoregressive moving average system (NARMA10), was carried out, which can be governed by

$$o(t) = \alpha \cdot o(t-1) + \beta \cdot o(t-1) \cdot \sum_{i=0}^{9} o(t-i) + \gamma \cdot d(t-9) \cdot d(t) + \delta \,, \qquad (22)$$

where $d(t)$ is the random input signal at time t; $o(t-1)$ is the output at the previous time step; α, β, γ, and δ are random design parameters that would be replaced with a new random values taken from a ±50% interval around the respective original constants for every 2000 steps. During the simulation, the initial condition of design parameters were set to be $\alpha = 0.3$, $\beta = 0.05$, $\gamma = 1.5$, and $\delta = 0.1$.

In this task, a total of 10 thousand sampling points were generated for training and testing phases. 100 samples were used for the initialization, 5900 samples were used for the training and 4000 samples were used for the testing. The prediction error was examined using the normalized root mean square error (NRMSE), and compared to state-of-the-art reservoir computing models. During the training process, output weights were trained by minimizing the deviation between predicted and target outputs. Both training and testing errors were achieved by the NRMSE, which can be defined as

$$NRMSE = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{n\sigma_{\hat{y}}^2}}, \qquad (23)$$

where y_i is the predicted output, $\hat{y}$ is the target output, n is the total number of samples, and $\sigma_{\hat{y}}^2$ is the output variance.

Table 4. Performance comparison in different models.

	Model	NRMSE		Error Rate Reduction
		Training	Testing	
[51]	ESN	/	0.1075	36.5%
[22]	DFR	/	0.15	54.5%
[31]	DFR	0.065	0.464	85.3%
[52]	DFR	/	0.17	59.8%
This Work	DFR	0.0849	0.0683	/

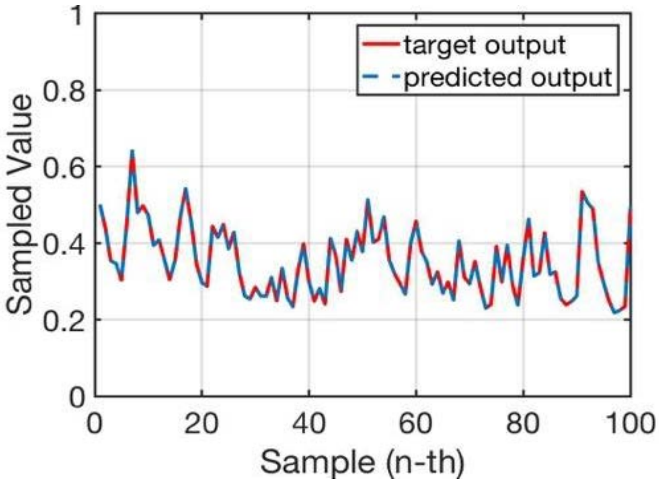


Figure 55. Target signals versus predicted signals for NARMA10 benchmark (demonstrated with the first 100 samples in the testing phase).

The NRMSE of training and testing operations was then evaluated through Equation (23). NRMSE results along with the comparison to state-of-the-are DFR designs were tabulated in Table 4. It can be observed that the NRMSE from our introduced DFR computing system exhibits a 36%-85% reduction on the error rate compared to state-of-the-art reservoir computing modules. The experimental result of predicted output signals against target outputs with our introduced DFR system is plotted in Figure 55.

4.6 Face Recognition

In this experiment, we evaluated the performance and reliability of our DFR system by using the application of face recognition. The learning system was built by using the multi-layer perceptron (MLP) training model as the readout layer of our introduced DFR system. Weights were then trained by the backpropagation training algorithm. Since the training time increases dramatically with the design scale, the hidden layer of the MLP training model was built with two neuron layers that contain 80 and 40 neurons, respectively.

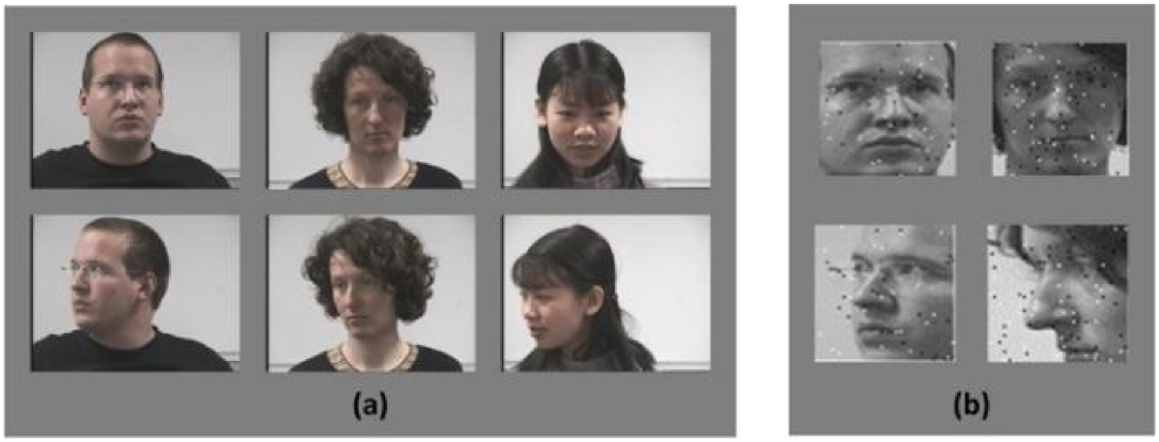


Figure 56. (a) Standard training database with three subjects; (b) down-sampled testing dataset with various salt-and-pepper noise levels.

Six images corresponding to different training subjects from the Head Pose Image database [53] are demonstrated in Figure 56(a). A total of 48 images, which contain three different subjects with multiple rotation angles, were drawn. Among those images, 24 were used during the training stage, while the rest of the images were used for the testing phase. For the training dataset, the horizontal angle of face alters from 0° to 75° with an increment of 15° per rotation, while the vertical angle of face maintains at 0°. In the testing dataset, the alteration of horizontal angles of face follow the training dataset but with an additional 15° applied to the vertical angle of the face. The reliability of the system was investigated by introducing various levels of salt-and-pepper noise to the down-sampled testing dataset, as depicted in Figure 56(b).

Figure 57 illustrates the test bench of our hybrid training model. During the operation, each input image was first down-sampled to 64 × 64 pixels, and each pixel of the image was mapped onto higher-dimensional spaces for linear separation by our DFR system. As its output, a total of 4096 spike trains, which represent the information of each pixel of an image, were generated from the reservoir layer. These spike data patterns were then recorded and converted into analog signals, x_n. Within the hidden layer of the MLP training model, weighted inputs were mapped to each neuron in the following layer. Initially, weights were generated randomly. During the training stage, as each piece of data was processed, weights were calibrated based on the corrections that minimizing the error between actual and target outputs. The output layer, which contains three neurons, represents 001, 010, 100 in a digital format, such that the system can classify input patterns up to three different categories. In order to prevent overfitting, the cross-validation technique was applied during the training process.

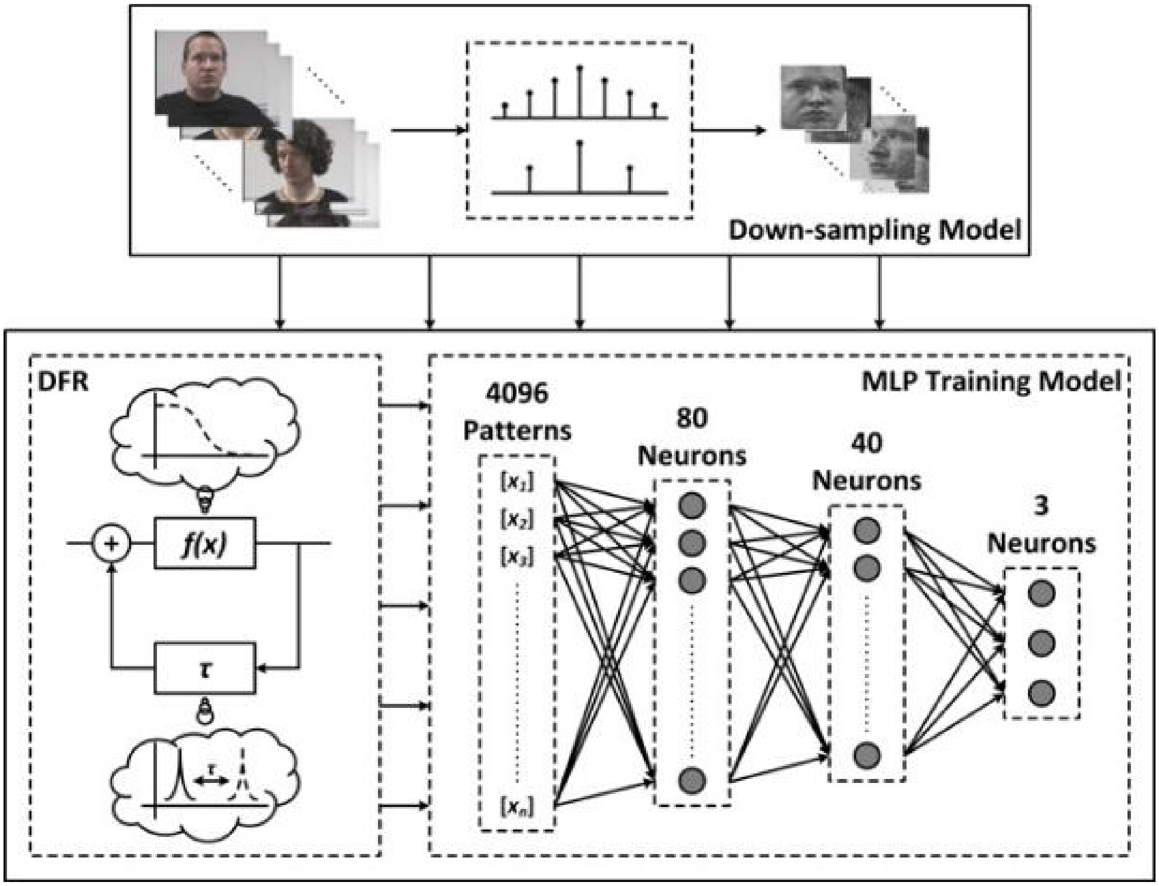


Figure 57. Hybrid training model for application evaluation.

The recognition rate was evaluated against different levels of salt-and-pepper noise with two different training models: (1) hybrid and (2) MLP-only. As depicted in Figure 58, the recognition rate maintains at 98% if the salt-and-pepper noise level was less than 10% by using the hybrid training model. The same trend can be found with the MLP-only training model, which was at a constant recognition rate of 78%. At low noise level, our hybrid training model exhibits a much higher recognition rate than that of the MLP-only training model. This trend can also be found at higher noise levels. As the noise level reaches 50%, the recognition rate of the hybrid training model was decreased to 93%. However, with the MLP-only training model at 50% of the noise level, the recognition rate has dropped to 67%, which was 26% more than that of the hybrid training model. Hence, the hybrid training model is more robust against noise than the MLP-only training model does.

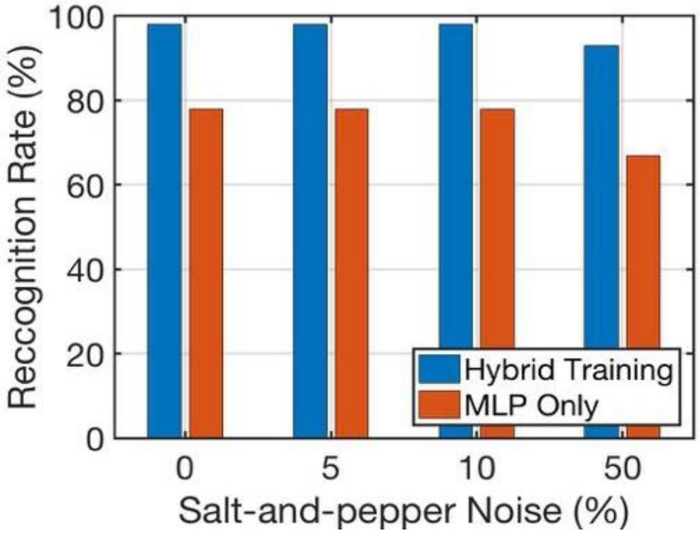


Figure 58. Recognition rate with respect to various levels of salt-to-pepper noise.

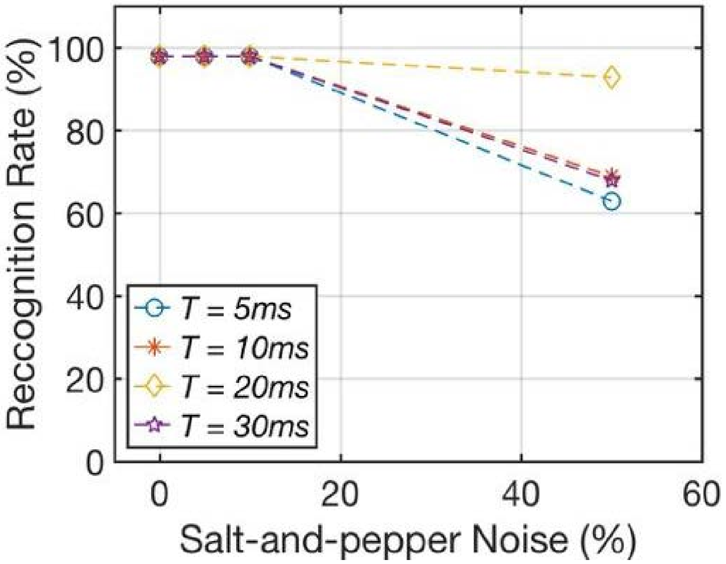


Figure 59. Recognition rate with respect to various dynamic behavior under different noise levels.

In this task, the recognition rate was evaluated with various delay times. As demonstrated in Figure 59, it can be observed that the recognition rate varies with respect to dynamic behaviors. When our DFR system operates at the edge of chaos regime (T = 20ms), the recognition rate was maintained at 98% with a noise level below 10%. As the noise level approaches 50%, the recognition rate at the edge of chaos regime maintains above 93%. However, if the delay deviates from 20ms, the recognition rate was drastically affected by the change in the dynamic behavior of the system, which resulted in approximately 25% decrease in the performance.

4.7 Performance Analysis of Multilayer Delayed Feedback Reservoir Systems

To evaluate the performance of two deep DFR models, two-time series prediction tasks were carried out to study the performance of deep DFR systems. Time series prediction tasks are important in real-world applications not only in the engineering field, but also in medical care [54, 55]. Computational abilities of our deep DFR models were examined using the NRMSE which was then compared to a baseline comparison model. Each deep DFR model contained four reservoir layers with 10 virtual nodes. In this task, the baseline comparison model was constructed by a leaky ESN model [56]. The total number of neurons used in the leaky ESN model was 40, which is equivalent to the total number of virtual nodes in the deep structures of DFR. The governing state equation for the leaky ESN is given as in [56]

$$x(t) = (1 - a) \cdot x(t-1) + a \cdot tanh\ tanh\ (u(t) \cdot W_{in} + x(t-1) \cdot W_{res})\ , \tag{24}$$

$$y(t) = x(t) \cdot W_{out} \tag{25}$$

where a is the leakage term, W_{in} is the input weight, W_{res} is the weight in the reservoir.

The first prediction task was the Santa Fe time series which is a typical benchmark test in the field of machine learning [57]. The Santa Fe dataset utilized in the task contained a total of 6600 values, which were generated by a laser working in the chaotic region. The dataset was divided into three portions, 100 values were used for initialization, 4000 samples were used for training, and 2500 samples were used for testing.

The other prediction task carried out was the prediction of the ElectroCardioGram (ECG) signal. The dataset consists of 7100 points whereby 100 samples were used for initialization, 5000 samples were used for training, and the rest samples were used for testing. Figure60 shows a portion of the testing results for both Santa Fe and ECG time series prediction tasks. The outputs from the virtual nodes were linearly combined through output weights. During the training process, the output weights were trained by minimizing the deviation of the predicted output to the target signal or correct output signal. After the training, testing was carried out using the trained output weight. Both training and testing errors were obtained by computing the NRMSE as in Equation (23).

Table 5. Training results comparison for different models.

Model	NRMSE		Normalized Training Time
	Santa Fe	ECG	
Leaky ESN	0.0896	0.0831	
deepDFR	0.0213	0.0418	0.45X
MI-deepDFR	0.0167	0.0319	0.57X

Table 6. Testing results comparison for different models.

Model	NRMSE	
	Santa Fe	ECG
Leaky ESN	0.0914	0.0917
deepDFR	0.0508	0.0561
MI-deepDFR	0.0395	0.0328

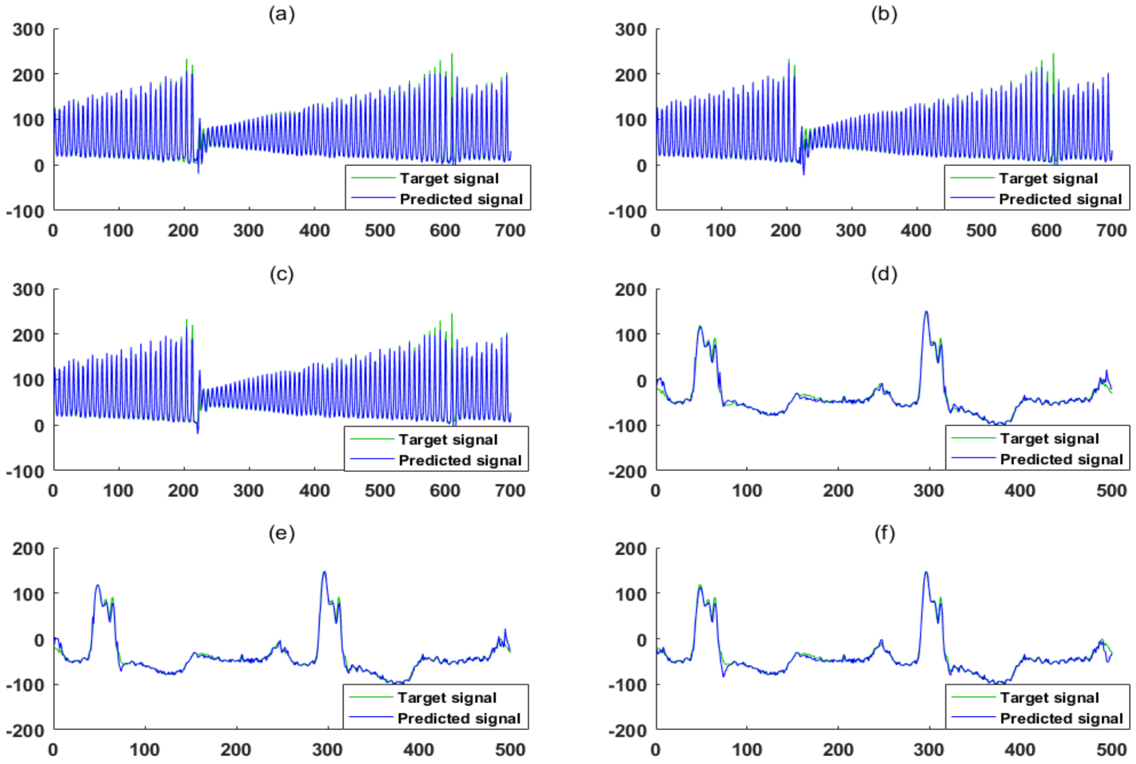


Figure 60. Target signal vs. predicted signal for (a) Santa Fe time series using leaky ESN; (b) Santa Fe time series using deep DFR; (c) Santa Fe time series using MI-deep DFR; (d) ECG using leaky ESN; (e) ECG using deep DFR; (f) ECG using MI-deep DFR.

Training and testing results for each deep DFR model were tabulated in Tables 5 and 6. As can be seen in Table 5, MI-deepDFR exhibits the lowest NRMSE for both prediction tasks among these three models during training. It is clear that in both prediction tasks, training NRMSEs for deep DFR systems was lower than that of leaky ESN. By evaluating training results, deep DFR systems show 76%-81% better performance than the shallow leaky ESN model in the Santa Fe time series prediction task. Whereas in the ECG prediction task, deep DFR systems exhibit 50%-62% performance improvement. Although MI-deepDFR illustrates better computational ability than that of the deepDFR, the training time of MI-deepDFR requires approximately 21% longer than that of deepDFR. Due to the difference in architecture, there was a tradeoff between accuracy and training time.

In Table 6, testing NRMSEs were listed for different models. During the testing stage, deep DFR systems exhibited 44%-57% better performance in the Santa Fe time series prediction task compared to that of the shallow leaky ESN model. In the ECG prediction task, the testing performance of deep DFR systems shows a 39%-64% improvement than the shallow model.

4.8 Potential of Hybrid Photonic-ASIC Platform Development

Traditional reservoir computing implementations are generally composed of three distinct parts: an input layer, the reservoir layer, and an output layer. The recurrent part of the network, named as the reservoir, acts as a kernel of dynamical features. If the reservoir has rich enough dynamics, it is possible to perform a wide range of tasks using only linear readout neurons to extract the relevant information from the reservoir. While most implementations of reservoir computing are embodied in software, efficient hardware implementations of these concepts would provide numerous advantages. Hardware implementations would be capable of exploiting the full potential of the intrinsic parallelism of neural networks. The dedicated hardware implementation for specific tasks also offers advantages over software implementations where low power consumption or high processing speeds are a priority.

In [22], authors exploited rich dynamics of delayed feedback information processing systems by using the system's transient response to an external input to show that a single nonlinear node with delayed feedback could replace a large network of nonlinear nodes. Their results demonstrated that this new information processing architecture performs well in a variety of tasks, such as time-series prediction and speech recognition. [58] and [59] introduce a photonic implementation of the reservoir with coupled semiconductor optical amplifiers (SOA). Table 7 shows the performance comparison between the photonic design and the IC implementation. Given the potential advantages, it is essential to explore the production of hybrid photonic-ASIC, which could offer an outstanding platform for hardware implementations of reservoir computing.

Both photonic and IC implementations of reservoir processors have advantages and disadvantages. The photonic implementation offers high-speed optical processing and high bandwidth, but it requires a large design size and has high power consumption. Moreover, the photonic implementation of reservoir processors requires expensive peripheral devices (digitizer, waveform generator, mach-zehnder modulator, etc.) and is difficult to scale. The IC implementation offers compact design size and low power dissipation but is susceptible to noise, which makes it difficult to design. At the time this report was prepared, IC implementations of reservoir processors have not yet been reported in the literature.

Table 7. Performance comparisons between photonic and ASIC implementations.

	Photonic	ASIC
Advantages	● Devices are nonlinear in nature ● High-speed optical processing ● High bandwidth	● Compact in size ● Low power dissipation ● Synaptic weight can be stored online or offline ● High computational precision, high reliability, and high programmability
Disadvantages	● Large size ● High power consumption ● Requires expensive peripheral devices ● Scaling is not easy	● Analog IC implementation is susceptible to noise ● Analog IC implementation is not available yet

5.0 CONCLUSIONS

The recurrent part of the network for the concept we explored, called the reservoir, acts as a kernel of dynamical features. If the reservoir has rich enough dynamics, it is possible to perform a wide range of tasks using only linear readout neurons to extract the relevant information from the reservoir. While most implementations of reservoir computing are embodied in software, efficient hardware implementations of these concepts would provide numerous advantages. Hardware implementations would be capable of exploiting the full potential of the intrinsic parallelism of neural networks. Dedicated hardware implementation for specific tasks also offers advantages over software implementations where low power consumption or high processing speeds are a priority. In this effort, we designed a delay-feedback reservoir system. An advanced neuron circuit design for Mackey-Glass function was introduced, which bears a much closer resemblance to the behavior of neural networks than that of the hyperbolic tangent and sigmoid functions. In order to ensure the real-time operation, the digital signals were required to interface with the analog world, which leads to the addition of digital-to-analog and analog-to-digital converters. However, our analog implementation has the advantage of implicit real-time operation, resulting in a small design area and lower power consumption. Furthermore, our dynamic delay-based Mackey-Glass neuron could perform nonlinear transformation and map input signals to higher dimensional state, which makes it well suited for reservoir computing.

Our design has five main contributions which were:

- The introduction of a Mackey-Glass nonlinear electronic circuit that facilitates the high dimensional projection operation as in the neural network design;
- A delay-feedback loop was presented with the capability to process the spiking information directly with a superior dynamic range of controllable delay time and a small design area;
- An inter-spike-interval temporal decoder with the spike-timing-dependent plasticity methodology was revealed that could achieve multiple scales by applying an ISI spike train;
- The power consumption and design area for our DFR system design were greatly reduced since power-hungry peripheral components, such as analog-to-digital and digital-to-analog converters, were not included;
- Demonstrated the potential of deep DFR systems providing better computational ability than a shallow neural network.

The significant technical contribution was the tape-out of the newly improved second-generation DFR system in December 2019 that is a spike-based DFR system. Analysis of the spike-based DFR with Mackey-Glass nonlinear transfer function revealed that it had a higher energy efficiency with less design area compared to state-of-the-art Mackey-Glass hardware implementations. More importantly, experimental results showed the DFR’s capability of operating at the edge of chaos regime with constant spike-based delay.

In summary, the specific objective of the project was to build a new class of computationally efficient delay-based reservoir computing systems that meet the requirements of high dimensionality and finite memory. This project pursued an agile analog IC implementation of spike-time encoding circuit as a signal conditioner and electronic reservoir as a dynamic processor for the reservoir computing system. This multidisciplinary effort bridged high-performance computing, nanotechnology, as well as integrated circuits and systems. The resulting DFR circuits and architectures could serve as the foundation for unprecedented capabilities in signature analysis and time-series classification with applications that fall within Air Force Research Laboratory (AFRL) neuromorphic computing consolidated programs.

6.0 REFERENCE

[1] Schaller, R.R., Moore's law: past, present and future. IEEE spectrum, 1997. 34(6): p. 52-59.

[2] Mead, C., Neuromorphic electronic systems. Proceedings of the IEEE, 1990. 78(10): p. 1629-1636.

[3] Jaeger, H., 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, 2001. 148(34): p. 13.

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

[5] Wu, Q., et al. A multi-answer character recognition method and its implementation on a high-performance computing cluster. in FUTURE COMPUTING 2011, The Third International Conference on Future Computational Technologies and Applications. 2011. Citeseer.

[6] Eryilmaz, S.B., et al. Neuromorphic architectures with electronic synapses. in 2016 17th International Symposium on Quality Electronic Design (ISQED). 2016. IEEE.

[7] Ebong, I.E. and P. Mazumder, CMOS and memristor-based neural network design for position detection. Proceedings of the IEEE, 2011. 100(6): p. 2050-2060.

[8] Kim, Y., Y. Zhang, and P. Li. A digital neuromorphic VLSI architecture with memristor crossbar synaptic array for machine learning. in 2012 IEEE International SOC Conference. 2012. IEEE.

[9] Seo, J.-s. and M. Seok. Digital CMOS neuromorphic processor design featuring unsupervised online learning. in 2015 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC). 2015. IEEE.

[10] Piguet, C., Low-power CMOS circuits: technology, logic design and CAD tools. 2005: CRC press.

[11] Smith, L.S., Neuromorphic systems: past, present and future, in Brain Inspired Cognitive Systems 2008. 2010, Springer. p. 167-182.

[12] Zhao, C., et al., Spike-time-dependent encoding for neuromorphic processors. ACM Journal on Emerging Technologies in Computing Systems (JETC), 2015. 12(3): p. 23.

[13] Ramanaiah, K. and S. Sridhar, Hardware implementation of artificial neural networks. i-Manager's Journal on Embedded Systems, 2014. 3(4): p. 31.

[14] Joubert, A., et al. Hardware spiking neurons design: Analog or digital? in The 2012 International Joint Conference on Neural Networks (IJCNN). 2012. IEEE.

[15] Basu, A. and P.E. Hasler, Nullcline-based design of a silicon neuron. IEEE Transactions on Circuits and Systems I: Regular Papers, 2010. 57(11): p. 2938-2947.

[16] Zhao, C., et al., Interspike-interval-based analog spike-time-dependent encoder for neuromorphic processors. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2017. 25(8): p. 2193-2205.

[17] Myers, D. and R. Hutchinson, Efficient implementation of piecewise linear activation function for digital VLSI neural networks. Electronics Letters, 1989. 25: p. 1662.

[18] Schrauwen, B., D. Verstraeten, and J. Van Campenhout. An overview of reservoir computing: theory, applications and implementations. in Proceedings of the 15th european symposium on artificial neural networks. p. 471-482 2007. 2007.

[19] Legenstein, R. and W. Maass, Edge of chaos and prediction of computational performance for neural circuit models. Neural Networks, 2007. 20(3): p. 323-334.

[20] Legenstein, R. and W. Maass, What makes a dynamical system computationally powerful. New directions in statistical signal processing: From systems to brain, 2007: p. 127-154.

[21] Amil, P., C. Cabeza, and A.C. Marti, Exact discrete-time implementation of the Mackey–Glass delayed model. IEEE Transactions on Circuits and Systems II: Express Briefs, 2015. 62(7): p. 681-685.

[22] Appeltant, L., et al., Information processing using a single dynamical node as complex system. Nature communications, 2011. 2: p. 468.

[23] Soriano, M.C., et al., Delay-based reservoir computing: noise effects in a combined analog and digital implementation. IEEE transactions on neural networks and learning systems, 2014. 26(2): p. 388-393.

[24] Milton, J.G., Time delays and the control of biological systems: An overview. IFAC-PapersOnLine, 2015. 48(12): p. 87-92.

[25] Wijekoon, J.H. and P. Dudek. Integrated circuit implementation of a cortical neuron. in 2008 IEEE International Symposium on Circuits and Systems. 2008. IEEE.

[26] Wojtyna, R. and T. Talaśka, Transresistance CMOS neuron for adaptive neural networks implemented in hardware. Bulletin of the Polish Academy of Sciences: Technical Sciences, 2006.

[27] Indiveri, G. A low-power adaptive integrate-and-fire neuron circuit. in Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS'03. 2003. IEEE.

[28] Joubert, A., B. Belhadj, and R. Héliot. A robust and compact 65 nm LIF analog neuron for computational purposes. in 2011 IEEE 9th International New Circuits and systems conference. 2011. IEEE.

[29] Schrauwen, B. and D. Stroobandt. Using reservoir computing in a decomposition approach for time series prediction. in ESTSP 2008 European Symposium on Time Series Prediction. 2008. Multiprint Oy/Otamedia.

[30] Jaeger, H. and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. science, 2004. 304(5667): p. 78-80.

[31] Goudarzi, A., M.R. Lakin, and D. Stefanovic. Reservoir computing approach to robust computation using unreliable nanoscale networks. in International Conference on Unconventional Computation and Natural Computation. 2014. Springer.

[32] Jin, Y., et al. Handwritten numeral recognition utilizing reservoir computing subject to optoelectronic feedback. in 2015 11th International Conference on Natural Computation (ICNC). 2015. IEEE.

[33] Hinaut, X. and P.F. Dominey. On-line processing of grammatical structure using reservoir computing. in International Conference on Artificial Neural Networks. 2012. Springer.

[34] Ghani, A., et al., Neuro-inspired speech recognition based on reservoir computing. Advances in Speech Recognition, 2010: p. 164.

[35] Namajūnas, A., K. Pyragas, and A. Tamaševičius, An electronic analog of the Mackey-Glass system. Physics Letters A, 1995. 201(1): p. 42-46.

[36] Koch, C. and I. Segev, The role of single neurons in information processing. Nature neuroscience, 2000. 3(11s): p. 1171.

[37] Giles, C.L. and T. Maxwell, Learning, invariance, and generalization in high-order neural networks. Applied optics, 1987. 26(23): p. 4972-4978.

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

[39] Pascanu, R., et al., How to construct deep recurrent neural networks. arXiv preprint arXiv:1312.6026, 2013.

[40] Hermans, M. and B. Schrauwen. Training and analysing deep recurrent neural networks. in Advances in neural information processing systems. 2013.

[41] LeCun, Y., Y. Bengio, and G. Hinton, Deep learning. nature, 2015. 521(7553): p. 436-444.

[42] Bueno, J., et al., Conditions for reservoir computing performance using semiconductor lasers with delayed optical feedback. Optics express, 2017. 25(3): p. 2401-2412.

[43] Zhao, C., et al., Analog Spike-Timing-Dependent Resistive Crossbar Design for Brain Inspired Computing. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2017. 8(1): p. 38-50.

[44] Hegger, R., et al., Identifying and modeling delay feedback systems. Physical review letters, 1998. 81(3): p. 558.

[45] Ikeda, K. and K. Matsumoto, High-dimensional chaotic behavior in systems with time-delayed feedback. Physica D: Nonlinear Phenomena, 1987. 29(1-2): p. 223-235.

[46] Bai, K., et al. Enabling An New Era of Brain-inspired Computing: Energy-efficient Spiking Neural Network with Ring Topology. in 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC). 2018. IEEE.

[47] Bai, K. and Y.Y. Bradley. 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). 2018. IEEE.

[48] Bai, K. and Y. Yi, DFR: An Energy-efficient Analog Delay Feedback Reservoir Computing System for Brain-inspired Computing. ACM Journal on Emerging Technologies in Computing Systems (JETC), 2018. 14(4): p. 45.

[49] Junges, L. and J.A. Gallas, Intricate routes to chaos in the Mackey–Glass delayed feedback system. Physics letters A, 2012. 376(30-31): p. 2109-2116.

[50] Sah, C.-T., Fundamentals of solid state electronics. 1991: World Scientific Publishing Company.

[51] Rodan, A. and P. Tino, Minimum complexity echo state network. IEEE transactions on neural networks, 2010. 22(1): p. 131-144.

[52] Ortín, S. and L. Pesquera, Reservoir computing with an ensemble of time-delay reservoirs. Cognitive Computation, 2017. 9(3): p. 327-336.

[53] Gourier, N., D. Hall, and J.L. Crowley. Estimating face orientation from robust detection of salient facial structures. in FG Net workshop on visual observation of deictic gestures. 2004. FGnet (IST–2000–26434) Cambridge, UK.

[54] Gutiérrez, J.M., et al. Simple reservoirs with chain topology based on a single time-delay nonlinear node. in ESANN. 2012.

[55] Kantz, H. and T. Schreiber, Nonlinear time series analysis. Vol. 7. 2004: Cambridge university press.

[56] Jaeger, H., et al., Optimization and applications of echo state networks with leaky-integrator neurons. Neural networks, 2007. 20(3): p. 335-352.

[57] Weigend, A.S., Time series prediction: forecasting the future and understanding the past. 2018: Routledge.

[58] Larger, L., et al., Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing. Optics express, 2012. 20(3): p. 3241-3249.

[59] Paquot, Y., et al., Optoelectronic reservoir computing. Scientific reports, 2012. 2: p. 287.

APPENDIX I – PUBLICATIONS

[1] Bai, K., and Y. Yi. "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, 2018.

[2] Li, J., K. Bai, L. Liu, and Yang Yi. "A deep learning based approach for analog hardware implementation of delayed feedback reservoir computing system." In 2018 19th International Symposium on Quality Electronic Design (ISQED), pp. 308-313. IEEE, 2018.

[3] Bai, K., Li, J., Hamedani, K. and Yi, Y. "Enabling a new era of brain-inspired computing: energy-efficient spiking neural network with ring topology." In Proceedings of the 55th Annual Design Automation Conference (DAC). ACM, 2018

[4] Bai, K., and Y. Yi. "DFR: An Energy-efficient Analog Delay Feedback Reservoir Computing System for Brain-inspired Computing." ACM Journal on Emerging Technologies in Computing Systems (JETC) 14, no. 4 (2018): 45.

[5] Bai, K., and Y. Yi. "Opening the "Black Box" of Silicon Chip Design in Neuromorphic Computing." In Bio-Inspired Technology. IntechOpen, 2019.

[6] Bai, K., Q. An, and Y. Yi. "Deep-DFR: A Memristive Deep Delayed Feedback Reservoir Computing System with Hybrid Neural Network Topology." In Proceedings of the 56th Annual Design Automation Conference 2019, p. 54. ACM, 2019.

[7] Zhao, C., Liu, L., & Yi, Y. "Design and Analysis of Real Time Spiking Neural Network Decoder for Neuromorphic Chips." In Proceedings of the International Conference on Neuromorphic Systems (p. 15). ACM, 2019

[8] Zhao, C., K. Bai, Q. An, B. Wysocki, C. Thiem, L. Liu, and Y. Yi. "Energy Efficient Temporal Spatial Information Processing Circuits Based on STDP and Spike Iteration." IEEE Transaction on Circuits and Systems II: Express Briefs. IEEE, 2019

[9] Bai, K., Q. An, L. Liu, Y. Yi. "A Training-efficient Hybrid Structured Deep Neural Network with Reconfigurable Memristive Synapses." IEEE Transaction on Very Large Scale Integration (VLSI) Systems. IEEE, 2019

[10] Bai, K., Liu, S. and Yi, Y. "November. High speed and energy efficient deep neural network for edge computing." In Proceedings of the 4th ACM/IEEE Symposium on Edge Computing (pp. 347-349). ACM, 2019

APPENDIX II – PRESENTATIONS AND MEETINGS

Presentations:

- “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), Santa Clara, CA, March 2018.
- “A deep learning based approach for analog hardware implementation of delayed feedback reservoir computing system,” in 2018 19th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, March 2018.
- “Enabling a new era of brain-inspired computing: energy-efficient spiking neural network with ring topology,” in 55th Annual Design Automation Conference (DAC), San Francisco, CA, July 2018.
- “Deep-DFR: A Memristive Deep Delayed Feedback Reservoir Computing System with Hybrid Neural Network Topology,” in 56th Annual Design Automation Conference (DAC), Las Vegas, NV, June 2019.
- “Design and Analysis of Real Time Spiking Neural Network Decoder for Neuromorphic Chips,” in International Conference on Neuromorphic Systems (ICONS), Oak Ridge, TN, July 2019.

Meetings:

- Air Force Research Laboratory kick off meeting (@VT) in Aug. 8th, 2018
- Air Force Research Laboratory kick off meeting (@AFRL) in July 15th, 2019
- Air Force Research Laboratory kick off meeting (@VT) in Dec. 10th, 2019

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

AFRL	Air Force Research Laboratory
ANN	Artificial Neural Network
ASIC	Application Specific Integrated Circuit
CMOS	Complementary Metal–Oxide–Semiconductor
CP	Charge Pump
DDE	Delay differential Equation
DDL	Digital Delay Line
DFR	Delayed Feedback Reservoir
DNN	Deep Neural Network
ESN	Echo State Network
FLOPS	Floating Point Operations Per Second
FNN	Feedforward Neural Network
GF	Global Foundries
HNN	Hybrid Neural Network
HPC	High-Performance Computing
IC	Integrated Circuit
I/O	Input/Output
ISI	Inter-Spike-Interval
I&F	Integrate-and-Fire
JFET	Junction Gate Field-Effect Transistor
LIF	Leaky Integrate-and-Fire
LSM	Liquid State Machine
MI	Multiple Input
MICS	Multifunctional Integrated Circuits and Systems
MIM	Metal-Insulator-Metal
MLP	Multilayer Perceptron
MOSIS	Metal Oxide Semiconductor Implementation Service
NARMA	Nonlinear Autoregressive Moving Average System
NMOS	N-type Metal-Oxide-Semiconductor
NRMSE	Normalized Root Mean Square Error
PMOS	P-type Metal-Oxide-Semiconductor
PWM	Pulse Width Modulation
RNN	Recurrent Neural Network
SOA	Semiconductor Optical Amplifier
SPICE	Simulation Program with Integrated Circuit Emphasis
STDP	Spike-Timing-Dependent Plasticity
SWAP	Size, Weight, and Power
VLSI	Very Large Scaled Integration
VT	Virginia Tech