PHOTONIC HARDWARE ACCELERATORS FOR RESERVOIR COMPUTING

A Dissertation

by

SYED ALI HASNAIN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Rabi Mahapatra
Committee Members,	Eun Jung Kim
	Dezhen Song
	Srinivas Shakkotti
Head of Department,	Scott Schaefer

December 2020

Major Subject: Computer Engineering

ABSTRACT

Machine Learning (ML) approaches like Deep Neural Networks (DNNs) have emerged as a powerful tool for big data classification and prediction problems. While feed-forward neural networks are good for non-temporal tasks, a lot of real-world problems like time series prediction (*e.g.* weather forecasting) and classification problems are temporal in nature. For such problems, Recurrent Neural Networks (RNNs) have been developed. However, the presence of recurrent connections coupled with iterative nature of training algorithms make RNN training extremely hard. Recently, it has been discovered that temporal problems can be solved by network of random recurrent connections coupled with a single trainable readout layer. This is called Reservoir Computing (RC). RC has emerged as a promising area but its implementation is challenging. In Software, RC provides limited performance, whereas hardware implementations have proved to be challenging due to many non-linear nodes present. To solve this problem, we propose to look towards the field of photonic computing to come up with high performance, power efficient photonic hardware accelerators for RC. We integrate ideas from ML, analog photonic computing, photonic device physics and hardware design to build architectures for photonic RC. We design a multi-layer photonic RC architecture to improve the performance of RC. We then integrate Time Division Multiplexing to exploit the inherent parallelism in reservoir layer and design a photonic architecture that is capable of running multiple tasks in parallel. To make photonic RC accelerators scalable, we design a first of its kind architecture that is completely on-chip.

Lastly, we study the limitations of the architectures design thus far and design a new kind of reconfigurable architecture that optimizes performance vs power consumption for any given task.

DEDICATION

To my late grandfather and teacher, Prof S.M. Asif, who taught us the value of education and to my parents who made it all possible.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Rabi Mahapatra for his guidance and support throughout the course of my research. His valuable advice at every step not only made me a better researcher but also a better human being. Without his support, this dissertation would not have come to its fruition. I would also like to thank my committee members, Dr. Eun Jung Kim, Dr. Dezhen Song and Dr. Srinivas Shakkotti for their valuable feedback and constructive criticism. I would also like to thank Dr. Duncan Walker and Karrie Bourquin for their helpful support as graduate advisors throughout the course of my graduate studies.

I would also like to thank my former lab mate and collaborator Dr. Dharnidhar Dang, for his timely feedback and ideas on my work. His contribution has always been very valuable. We have also shared an interest for cricket, and played many matches together. I will also like to thank my lab mates Jyotikrishna Dass, Karl Ott and Jerry Yiu, who have always made themselves available for brainstorming ideas.

I would not have been able to complete my Ph.D. without the support of my family. Without the love, support, encouragement and guidance of my parents, I would never have been able to reach where I am today. They have always inspired me to do better. I would also like to thank my brother, Umair, for his continuous support. My sister, Aleena, for making me believe I can do this and my youngest sister, Shiza, for providing all the entertainment throughout these years. I would also like to thank my grandmothers for their prayers, love and support. My uncles, who have been very supportive throughout, have

each been a role model, in their own way for me and for that I am grateful to them. Last but not the least, I would like to thank my beautiful wife, Maham, for first agreeing to marry me and then supporting me through the last phase of this dissertation.

Finally, I would also like to thank my friends Qasim, Hameed ul Haq, Basim, Sajjad, Umair, Ovais, Hamza, Taimoor, Maliha and Maheen for making my journey in USA throughout my Ph.D. a beautiful one.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Dr. Rabi Mahapatra (advisor), Dr. Eun Jung Kim and Dr. Dezhen Song of Department of Computer Science and Engineering and Dr. Srinivas Shakkotti of Department of Electrical and Computer Engineering.

Dr. Dharnidhar Dang of University of California, San Diego helped review results presented in Chapter 2.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported through graduate assistantships from the Department of Computer Science And Engineering.

NOMENCLATURE

ML	Machine Learning
RNN	Recurrent Neural Network
RC	Reservoir Computing
DNN	Deep Neural Network
DFR	Delayed Feedback Model
DL	Delay Line
MZI	Mach Zehnder Interferometer
MRR	Micro Ring Resonator
GPU	Graphic Processing Unit
NARMA	Non-linear Auto Regressive Moving Average
NMSE	Normalized Mean Square Error
BER	Bit Error Rate
WER	Word Error Rate
TDM	Time Division Multiplexing

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

1.1. The Reservoir Computing Model

In the advent of big data, Machine Learning (ML) has emerged as a promising area to solve prediction and classification problems. Loosely speaking, ML "gives the computer's ability to learn without being explicitly programmed" (Arthur Samuel, 1959). More formally, ML can be defined as computational learning using algorithms to learn from and make predictions on data. ML is different from traditional programming as no explicit programming related to data is required. Figure 1.1 depicts this difference between traditional learning and Machine Learning.

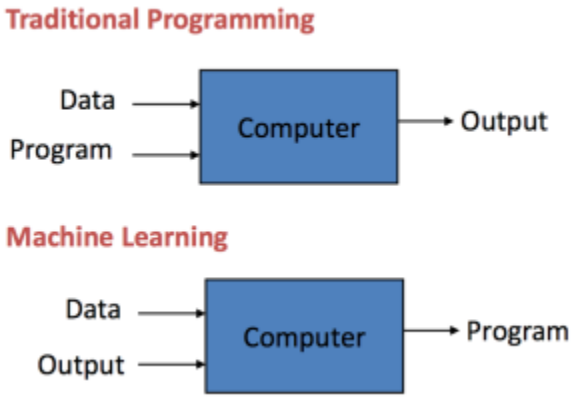


Figure 1-1 Traditional Programming Versus Machine Learning

While many different ML methods exist, Artificial Neural networks (ANN) are extremely popular [1, 2, 3, 4, 5, 6, 7]. Given data, these networks compute transformations of features or representation to make a final decision on it as represented in Figure 1.2. Feed forward networks comprising of multiple layers (DNNs) have been used extensively

in literature to study non-temporal problems. These networks are also well understood due

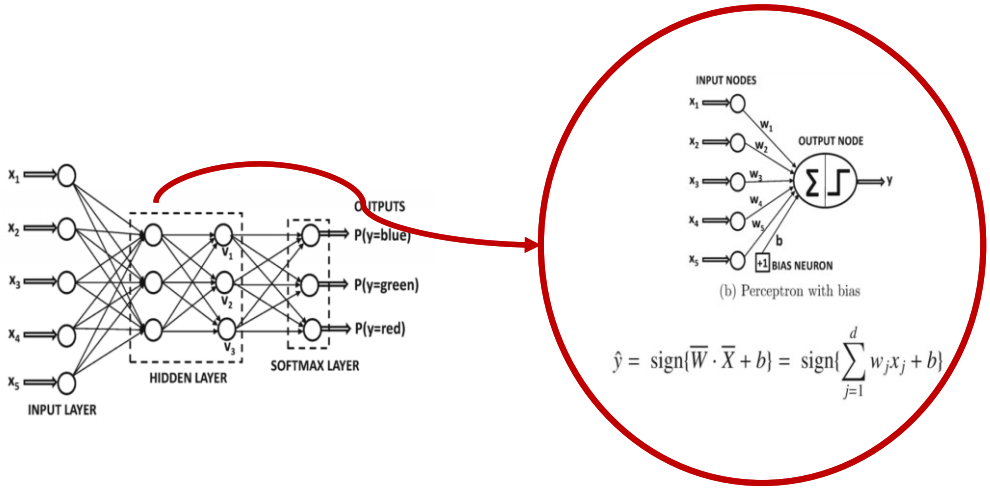


Figure 1-2 Deep Neural Network Architecture Comprising Of Multiple Neural Layers

to their non-dynamic nature. DNNs comprise of several layers, with each layer containing many neurons as shown in Figure 1.3. Mathematically, these networks are trained by solving an optimization problem to compute the appropriate weights [8]. The training process is computationally very expensive and requires a lot of computational power.

However, many real-world problems are temporal in nature. For example, prediction problems like financial data forecasting or weather forecasting and classification problems like speech recognition [9]. The feedforward networks have been

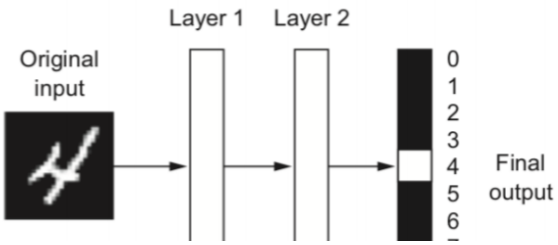


Figure 1-3 Deep Neural Network For Digit Classification

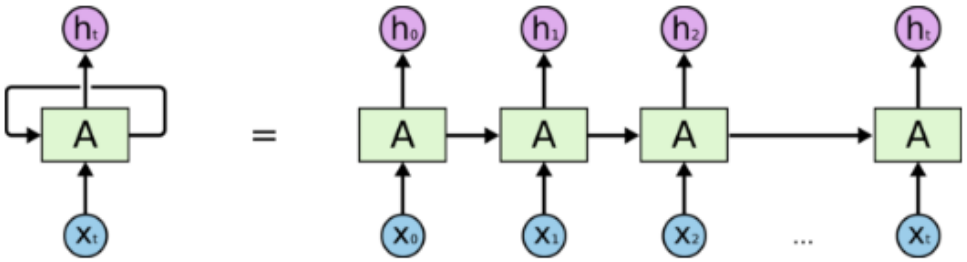


Figure 1-4 An Unrolled Recurrent Neural Network

used to solve these temporal problems by means of converting temporal problems to spatial problems by delayed embeddings. However, this is not a natural way of representing time [10]. Furthermore, such methods give rise to problems like artificially introduced time horizons [11]. To solve temporal problems Recurrent Neural Networks (RNNs) have been proposed. The networks consist of many recurrent connections as shown in Figure 1.4. The recurrent connections give rise to even larger number of parameters to be trained and hence training an RNN is even more compute intensive.

1.2. Reservoir Computing

Recently, it has been discovered that temporal problems can also be solved by a random network of recurrent connections coupled with a single trainable readout layer. This idea was proposed by Rosenblatt in 1962[12]. Fairly recent works have explored this paradigm and have independently come up with two main models, the echo state networks (ESNs) and the liquid state machines (LSMs)[13, 14]. The ESNs are based on non-spiking artificial neural networks whereas the LSMs were proposed in the context of spiking neural networks. The two concepts combined together have been named as Reservoir Computing (RC)[15]. Reservoir Computing (RC) is a subset of RNN and is a promising

approach to solve large scale classification and prediction problems[16, 17, 18, 19, 20]. In an RC system, there are three main layers: input layer, reservoir layer and output layer. The basic idea behind RC can be summed up as follows:

1. The reservoir layer is used for feature extraction and comprises of several hidden layers. The readout layer is trained for prediction or classification problem.
2. Contrary to RNN, the weights of input layer, W_{in}, as well as the hidden layers inside the reservoir layer, W_{random}, are not trained but are randomly initialized.
3. Only the weights of the output layer, W_{out}, are trained. This allows for a far reduced training time compared to RNN. Figure 1.5 shows a representation of a RC network.

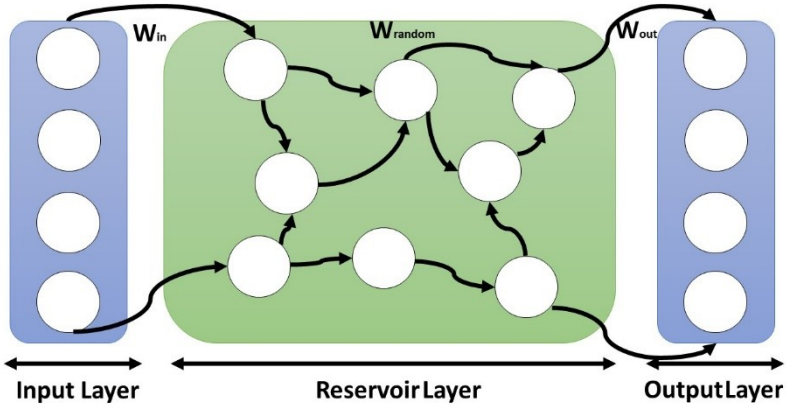


Figure 1-5 Reservoir Computing Architecture

RC has emerged as a promising field, with various benefits. The reduced training time makes the RC network very attractive. As with most neural networks this concept is inspired by the workings of biological brain. Several studies have argued that RC concept is similar to the workings of a brain, where an external stimulus excites the internal states

of network and processes information[21, 22, 14, 23, 24, 25]. Furthermore, the reservoir section of RC is independent of task being performed. This means that multiple tasks can use the same reservoir with a different readout layer for each task. A reservoir can be considered as a generic computation tool that is task independent. As a result, several tasks can be performed in parallel. As an example, recently, it has been shown that the same reservoir can be used for speech recognition as well as speaker recognition [26].

While RC is a promising approach, there are still many open research questions. These questions can be categorized into three main research areas in the field:

1. Computational Aspect
2. Design of Reservoir
3. Implementation of RC

In our work, we focus on implementation of RC with the goals of improving accuracy as well as making the implementation high performing and energy efficient. While RC can be implemented on software using conventional processors, some applications require hardware accelerators. In an era where Internet of Things (IoT) and edge computing ideas are fast taking shape, compact, fast and energy efficient devices are required for computation and data processing. A hardware accelerator is designed with these needs in mind.

1.3. Hardware Implementations Of Reservoir Computing

While RC has performance comparable to RNNs [27], its implementation has been challenging. In the literature, software implementations of RC provide limited performance [10] while hardware implementations of RC prove to be difficult due to many

non-linear nodes that exist in reservoir layer [28]. To solve this challenge, we first study the different topologies of RC that have been proposed. RC networks have three stages: Input layer, Reservoir and Output layer, as shown in Figure 1.6.

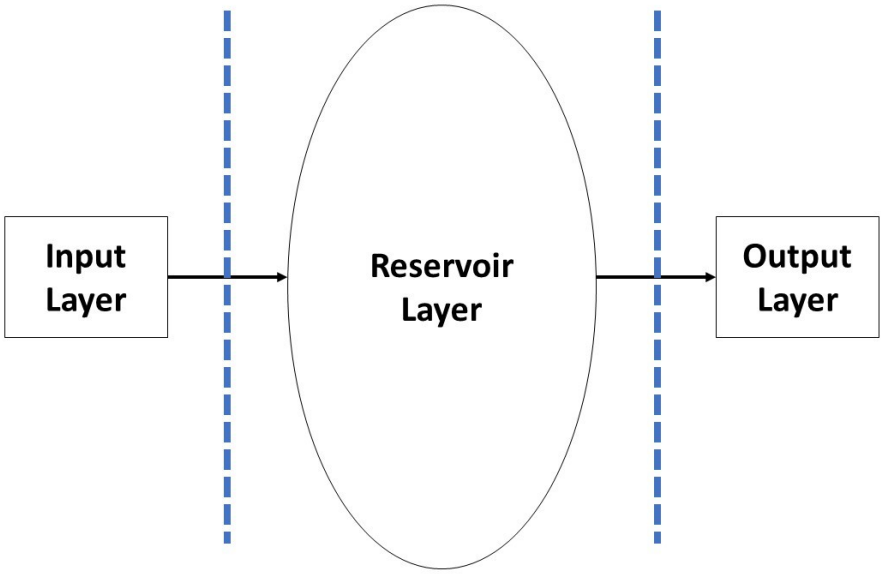
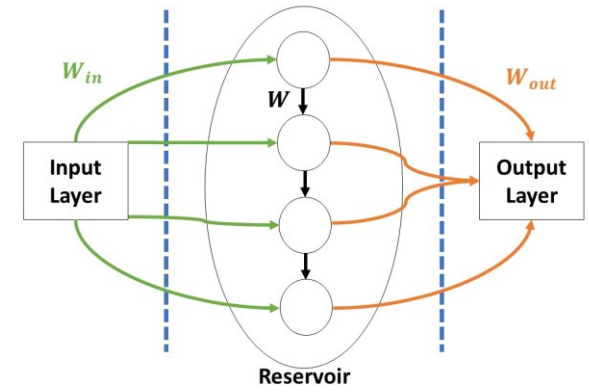


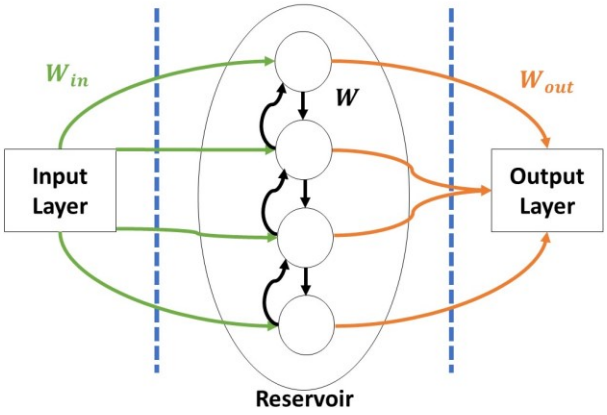
Figure 1-6 Three Stages Of The Reservoir Computing

The input layer and output layer are consistent in all topologies. However, due to the random nature of reservoir layer, we can arrange the connections in several ways [29]. Researchers in [29] have investigated various reservoir topologies and have come up with three main topologies: Delay Line, Delay line with feedback and Single Cycle Reservoir (SCR). The topologies differ in the way feedback connections are arranged in the reservoir, as shown in figure 1.7.

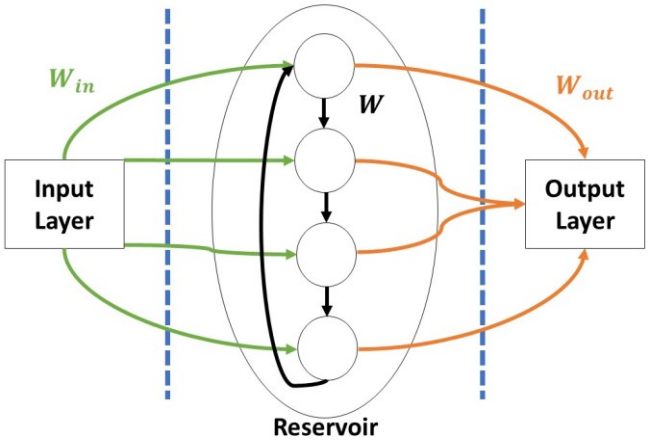
The delayed feedback reservoir (DFR) model of RC[30, 31, 32], inspired by SCR has emerged as a very attractive model to implement Reservoir Computing. It uses only one single neuron along with a delay line in a ring topology to create the reservoir.



(a)



(b)



(c)

Figure 1-7 (A) Delay Line Reservoir Topology (B) Delay Line Feedback Topology (C) Single Cycle Reservoir Topology

especially in the photonics domain can be used for its implementation. Following this model, several silicon photonics based single layer RC implementations have been presented in the literature [33, 34, 35]. These single-layer photonic implementations demonstrate 100x improvements in speedup compared to previous RC implementations. However, the accuracy of these implementations is lower than software RNN implementations. Recently, authors in [36, 37] have demonstrated software implementation of multi-layer RC to be of identical accuracy as compared to RNN implementations at the expense of large execution time. These systems also employ fiber optics as delay line and contain off-chip elements. While the proposed architectures are very promising, they still lack performance, use only one layer of reservoir and employ off chip components. Therefore, there is a need for more accurate, high performing accelerators for RC that addresses these issues.

1.4. The Emergence Of Silicon Photonic

Silicon photonics has emerged not only as an exciting prospect for on chip interconnects but also for computation in the analog domain. The computation and communication in silicon photonics domain has proved to be high speed, high bandwidth and less power hungry compared to traditional electronic counterparts. Moreover, several commercial photonics CAD tool like IPKISS [38] are available now that enables working with photonic components easier. All of this serves as a motivation to design high speed and high bandwidth computing systems that can process large scale data, while consuming less power than their electronic counterparts.

1.4.1. Basics Of Silicon Photonics

Silicon photonics-based computing systems employ several optoelectronic components. Before discussing photonic architectures for RC, it is important to review these components.

1.4.1.1. Waveguides And Fiber Spools

Silicon photonic waveguide is one of the building blocks for not only photonic components but also many photonic computing and communication systems. Waveguides are used as building blocks for Micro Ring Resonators, Mach Zehnder Modulators, Couplers as well as feedback mechanisms in photonic components. A silicon core with cladding is used to make a waveguide. Using the principle of total internal reflection and difference in refractive index of core and cladding, light is contained inside the waveguide and travels through it.

A fiber spool is a fiber optic cable, that again uses the principle of total internal reflection to transmit light from one point to the other. These are often employed to provide a delay in photonic computing systems as well as communication over large distances.

1.4.1.2. Mach Zehnder Modulator

A Mach Zehnder Modulator (MZM) or Mach Zehnder Interferometer (MZI) is another important photonic component. This component is used to modulate data over laser light. A MZM is basically two waveguides connected at either end by a Y junction. The length of the waveguides is normally unequal which causes light traveling through both arms to have different phase. The constructive and destructive interference helps

create an output. Researchers have also come up with clever methods to use electrical field to control light speed in one of the arms. This enables modulation of electrical data over the laser light. Figure 1.8 below shows the basic working of MZM.

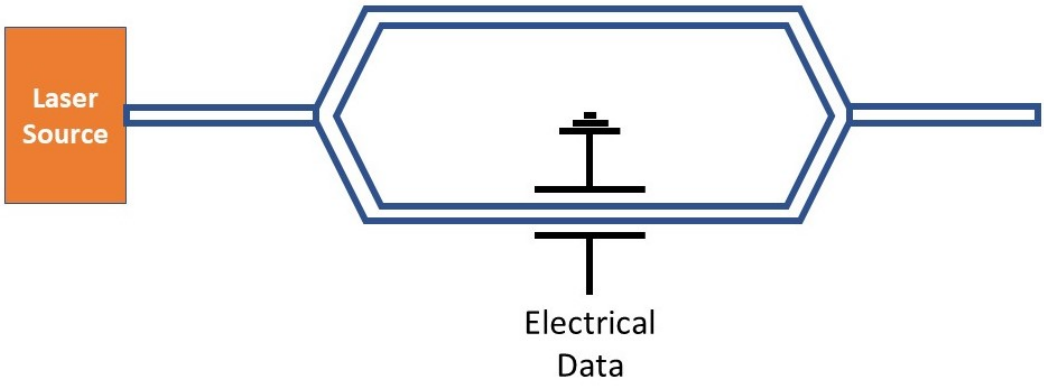


Figure 1-8 MZI Structure With Laser Light As Input And Electric Field On One Arm To Modulate Data

Laser light enter the modulator from one end and is split into the two arms. In one arm light propagates without any change. However, on the other arm, an electric field is applied which changes the refractive index of that arm and slows down the propagation of light in this arm. Due to the applied electric field, light in lower arm goes out of phase with light in upper arm. At the output, constructive and destructive interference cause a modulated version of laser light. This phenomenon can be used as the activation function of a neuron.

1.4.1.3. Micro Ring Resonators

Optical Ring Resonators have been extensively discussed in literature [39, 40]. They are essential for the success of silicon photonics. Using resonance, they can enable

the control of photonic path. In literature several MRR designs have been proposed [41, 42]. The main focus of these design is speed, application and size. Recently a 2x2 MRR was proposed which can be employed as reconfigurable DEMUX/MUX[43]. Such a component is ideally suited for selection of a delay line in a multi delay line-based architecture.

1.4.1.4. Photodiodes

Photodiodes are used for detection of light traveling through a component or system. It is also used for optical to electrical conversion of data. Photodiodes are often designed for a specific band and a rise/fall time.

1.5. Research Focus

This dissertation focuses on design of high speed and power efficient photonic Reservoir Computing hardware accelerators. We first review the working of a single node photonic computing model that is based on delayed feedback reservoir (DFR) computing model. We begin by extending the single node computing model, to incorporate multiple layers for Reservoir for Deep RC. Deep RC improves the performance of the system. We then study the ability of a reservoir to process multiple tasks in parallel and propose an improvement in our architecture to exploit parallel processing. To make the systems compact and on chip, we also make an effort to design a completely on chip system. Furthermore, we study the performance and power consumption of a system with different reservoir configurations and investigate reconfigurable photonic RC accelerators.

1.6. Contributions

The contributions of this dissertation are as follows:

MReC- A Multilayer Photonic Reservoir Computing Architecture: We propose MReC, a novel scalable and energy-efficient multilayer photonic reservoir computing architecture for large-scale classification and prediction. To the best of our knowledge, the proposed work is first of its kind using photonic components for multilayer reservoir computing. We introduce a multi-layer pipeline approach to enhance MReC's overall throughput by introducing multiple delay lines in series in the microarchitecture. We synthesized the proposed MReC architecture using commercial photonic CAD tools and ran system-level simulation on the architecture using four well-known classification and prediction benchmarks. The results demonstrate: (1) up to 26.8% reduction in prediction error rate in 1-layer MReC and up to 50% reduction in prediction error in 4-layer MReC compared to state-of-the-art design; (2) at least 132x improvement in speedup compared to best reported result; and (3) up to 34.21% improvement in power consumption compared to the best hardware implementation in literature. These improvements come at a cost of 12% area overhead.

Multilayer Photonic Reservoir Computing Architecture using Time Division Multiplexing for Parallel Computation: We review the RC computing principles and explain the single node photonic computing paradigm. We propose a new time-shared multi-layer photonic architecture for RC to perform tasks in parallel. Through experiments we show that our architecture can outperform some of the leading single layer architectures by up to 90% for NARMA task while performing analog speech recognition in parallel. We also show that our proposed architecture closely matches the performance of leading multi-layer photonic RC architecture with an increased error of 8% only due to

parallel processing. It is also shown that the proposed high-speed architecture has a power consumption of ~50W for a 4-layer network.

On-Chip Parallel Photonic Reservoir Computing using Multiple Delay lines: We propose a new architecture for on-chip parallel photonic reservoir computing employing multiple electronically tunable delay lines along with an MRR switch for delay line selection. Through simulations we show that the proposed architecture is up to 84% more accurate compared to a leading architecture while executing NARMA task alone and 80% more accurate when executing two tasks in parallel. It outperforms other architectures presented in literature. We also show that the proposed architecture performs 46% more accurate compared to an RC architecture employing Time Division Multiplexing (TDM) at input layer to execute tasks in parallel. It is shown that the architecture removes the off-chip fiber optics-based delay line at the cost of 0.0184 mm2 of on chip area. The power overhead is just 26mW.

Towards reconfigurable optoelectronic hardware accelerator for reservoir computing: we propose a new reconfigurable optoelectronic architecture for multi-layer RC. Our proposed architecture, is based on DFR model implemented by the use of Mach Zehnder Modulator (MZM) and on chip low loss delay lines for improved performance. It integrates photonic switches based on Micro Ring Resonators (MRR) to enable reconfigurability. The architecture enables layer selection and layer gating to select the number of layers required for a task. Selection of number of layers can optimize the architecture for a specific application, resulting in huge power savings, while maintaining the overall accuracy. Our experiments with NARMA task and analog speech recognition

task show that by optimally configuring an up-to 4-layer architecture, power savings up to 40% can be achieved compared to state-of-the-art architectures while gaining up to 80% more accuracy. Our scalable architecture has an on-chip area overhead of 0.0184mm^2 for a single delay line and MRR switch.

Each chapter of the dissertation focuses on key elements to improve for design. The contribution of each of the works can be summed in Table 1-1

Table 1-1 Summary Of Design Features Of Each Of The Proposed Photonic Architectures

	Scalability	**Performance**	**Parallelism**	**Energy Efficiency**
Chapter 2		✔		✔
Chapter 3		✔	✔	
Chapter 4	✔	✔	✔	
Chapter 5	✔	✔		✔

1.7. Organization

The rest of the dissertation is organized as follows: Chapter 2 discusses the design of a proposed multilayer photonic reservoir computing architecture. This is followed by Chapter 3 which discusses a modified multilayer photonic computing architecture that integrates time division multiplexing in its reservoir layer to enable parallel processing. Chapter 4 investigates the use of multiple on chip delay lines and proposes an architecture for completely on chip parallel reservoir computing system. Chapter 5 investigates the effect of reservoir configuration on performance and power, and proposes an architecture for reconfigurable reservoir computing system. This is followed by Chapter 6 which discusses some of the potential applications in which RC and our proposed hardware accelerators can be employed. Chapter 7 concludes this dissertation.

2. A MULTILAYER PHOTONIC RESERVOIR COMPUTING SYSTEM[1]

Photonic reservoir computing is a promising paradigm for large-scale classification and prediction problems. However, its single-layer nature is a bottleneck for higher performance and accuracy. Therefore, in this chapter we investigate the design of a multilayer RC system that can outperform other state of the art architectures.

2.1. Motivation

Recurrent Neural Networks (RNN) are often used in sequential classification and prediction problems. Deep RNN is a promising approach to enhance the accuracy of large-scale machine learning applications [44, 45, 46] . However, training a deep RNN is very compute intensive as weights of all the layers are determined in a sequential fashion. A subset of RNN is Reservoir computing (RC). It has emerged as a promising candidate to provide reduced training time with similar accuracy[27]. RNN and RC both consist of two stages: feature extractor and feature classifier. Feature extractor in RNN comprises of several hidden layers with weights that are trained extensively using available datasets to fine tune desired features. However, in RC, randomly generated fixed weights are used in the hidden layers to generate such features. For the feature classification, weights are used both in RNN and RC to fine tune the output. It may be noted that the training of weights in RC takes place only in the classifier stage. This makes the training and

[1]adapted with permission from Copyright © 2019, IEEE Dhang, D., Hasnain, S. A., & Mahapatra, R. (2019, March). MReC: A multilayer photonic reservoir computing architecture. In *20th International Symposium on Quality Electronic Design (ISQED)* (pp. 170-175). IEEE.

classification/prediction of an RC less time consuming compared to RNN. The feature extractor and the feature classifier of an RC are otherwise known as reservoir layer and output layer respectively as shown in Figure 2.1. While RC is a promising solution to reduced training time, its implementation has been a challenge.

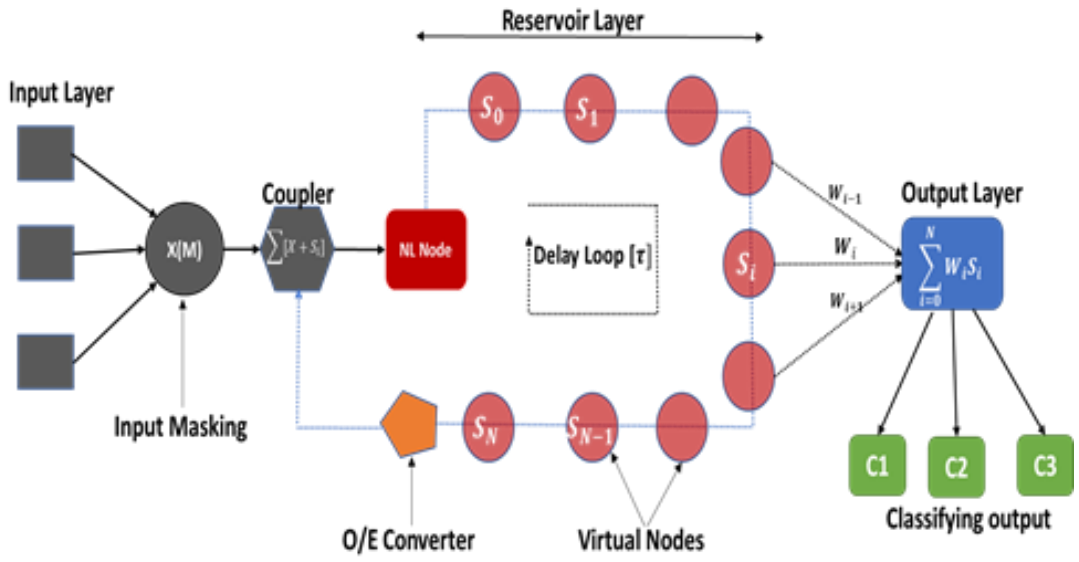


Figure 2-1 Single Node Photonic Computing Model

In the literature, software implementations of RC provide limited performance [27, 10] while hardware implementations of RC prove to be cumbersome due to many non-linear nodes that exist in reservoir layer [28]. The delayed feedback reservoir (DFR) model of RC has emerged as a potential solution to this problem. It uses only one single neuron in a ring topology to create the reservoir. Following this model, several silicon photonics based single layer RC implementations have been presented in the literature [35, 34, 33]. These single-layer photonic implementations demonstrate 100x improvements in speedup

compared to previous RC implementations. However, the accuracy of these implementations is lower than software RNN implementations. Intuitively, a multi-layer RC approach would improve the accuracy of RC system similar to deep neural network. Recently, authors in [37, 36] have demonstrated software implementation of multi-layer RC to be of identical accuracy as compared to RNN implementations at the expense of large execution time.

Therefore, in this chapter we propose a hardware implementation of multi-layer Reservoir computing using photonic components. Our proposed architecture improves the overall performance of such system. A hardware based multi- layer RC not only increases the accuracy of such systems but is also going to be faster than software-based implementations.

2.2. Related Works

In literature, multiple single layer photonic RC architectures have been proposed[33, 34, 35]. These architectures are based on DFR model and employ fiber optics and Mach Zehnder Interferometer as the non-linear node. The delay line length varies from 20m to 1.7Km. However, the use of fiber optics of large lengths makes these architectures non-scalable. These implementations have power consumption ranging from 76W to 200W. Alternative to optoelectronic architectures for RC, memristor based models have also been proposed in literature [47]. These architectures employ memristor meshes to implement neurons[48, 49]. While they also show promising results for specific applications, the use of memristors raises reliability issues. The designs are also less fault tolerant.

SW based implementations of RC have also been presented which demonstrate performance at par with RNNs [36, 37]. However, the increased performance comes at the expense of a larger execution time. Another proposed model that employs multiple reservoirs was proposed in [50]. However, there are no implementation details.

2.3. Contribution

In this chapter the following contributions are made:

1. We propose MReC, a novel scalable and energy-efficient multilayer photonic reservoir computing architecture for large-scale classification and prediction. To the best of our knowledge, the proposed work is first of its kind using photonic components for multilayer reservoir computing.
2. We introduce a multi-layer pipeline approach to enhance MReC's overall throughput by introducing multiple delay lines in series in the microarchitecture.
3. We synthesized the proposed MReC architecture using commercial photonic CAD tools and ran system-level simulation on the architecture using four well-known classification and prediction benchmarks. The results demonstrate: (1) up to 26.8% reduction in prediction error rate in 1-layer MReC and up to 50% reduction in prediction error in 4-layer MReC compared to state-of-the-art design [34]; (2) at least 132x improvement in speedup compared to best reported result [34]; and (3) up to 34.21% improvement in power consumption compared to the best hardware implementation in [33].
4. These improvements come at a cost of 12% area overhead.

2.4. Overview Of Photonic Reservoir Computing

RC's popularity is growing rapidly in the field of data science due to its high-speed prediction mechanism. The working principle of RC is as follows.

2.4.1. Principle Of Reservoir Computing

A RC comprises of an input layer, a reservoir layer, and an output layer as shown in Figure 2-1. The input layer of a RC distributes input data to its reservoir layer in discrete time through fixed connection weights. The reservoir layer is a dynamical system whose state at discrete time step n can be described as a set of N scalar variables $S_i(n)$ ($i = 1,2, \dots N$) called neurons. All the neurons in a reservoir layer are randomly interconnected with fixed random weights, constituting a recurrent network (i.e. a network of neurons having feedback loops). Under the influence of input data, the reservoir layer exhibits transient responses. The transient behavior of a reservoir is governed by an evolution equation as depicted in Equation 1:

$$S_i(n) = f[\propto C_i x(n) + \beta \sum_1^N w_{ij} s_j(n-1)] \tag{2.1}$$

Here, $S_i(n)$ is the state of i^{th} neuron at discrete time n, f is a non-linear function, $x(n)$ is input to RC at discrete time n, C_i & w_{ij} are the connection coefficients that define the topology of a reservoir layer, and $\propto$ & β are tuning parameters to regulate the dynamics of a reservoir. The transient states $S_i(n)$ are fed to the output layer through readout weights W_i to determine the output $O(n)$.

$$O(n) = \sum_1^N W_i s_i(n) + W_{bias} \tag{2.2}$$

Here, W_{bias} is the bias value required for the training of RC. During the training phase, the readout-layer weights W_i and bias weight W_{bias} are optimized to minimize error between the expected output $O'(n)$ and the actual output $O(n)$.

The performance of a RC is directly proportional to the number of neurons in its reservoir layer [44]. There are several attempts to design hardware implementation of RC which involves multiple photonic neurons [45]. However, such an approach is not feasible for designing RCs with thousands of neurons as it would cost significant power and area overhead [46]. A solution to this problem is a single node photonic RC with delay dynamics, which is explained in the following section.

2.4.2. Single Node Photonic RC

A photonic implementation of RC fully exploits the advantages of optical properties of photonic hardware (low-power, high-bandwidth, and inherent parallelism). Figure 2-1 shows a conceptual diagram of a single node photonic RC. A single node photonic RC can be treated as a single-layer photonic RC architecture. It comprises of an input layer, a reservoir layer, and an output layer. In the input layer, input signal multiplied with a masking function is fed to an optical coupler. The random masking input serves the same purpose in this system, as weights of interconnects do in a standard electrical or software-based RC system. The reservoir layer comprises of a Mach Zehnder interferometer based non-linear node (NL node) and optical fiber spool-based delay loop. MZI along with the optical fiber spool is considered as neurons in RC system. In the output layer, data from the reservoir layer are trained offline. The details of each layer are presented in next section. The working principle of a single-layer RC is as follows.

The coupler receives discrete masked input $x_i(n)$ from the input layer and feedback signal from the O/E converter. The O/E converter converts analog optical data of duration T to N discrete electrical samples $s_i(n)$ which is fed to the coupler one by one. Each $s_i(n)$ represents the state of a neuron in the reservoir layer at time step n. At time step n+1, The NL node receives sum of $s_i(n)$ and masked input $x_i(n+1)$ and then transforms it to $s_i(n+1)$ as depicted in Equation.1. After that, the state value $s_i(n+1)$ is stored in the delay loop to be used in the next time step, i.e. $n+2$. The ratio of delay τ to the O/E conversion time t determines the number of neurons N in this kind of design.

2.5. MReC Architecture

A multilayer RC is realized by simply including multiple reservoir layers (NL + delay) in between architecture output of optical couplers and input of output layer as shown in Figure 2-2. Each reservoir layer stores multiple reservoir states. As shown, the output from 1st layer enters the NL Node of 2nd layer as input and so on. Each reservoir state from the last layer (Mth layer) is fed to the readout layer for training. The classification/prediction of output follows Equation.2. The weights and bias value are trained using a linear regression technique in an offline computer to determine the final output.

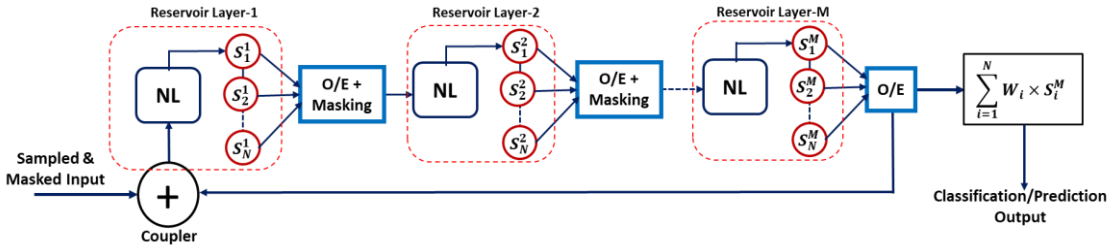


Figure 2-2 Logical Schematic Of MReC Architecture

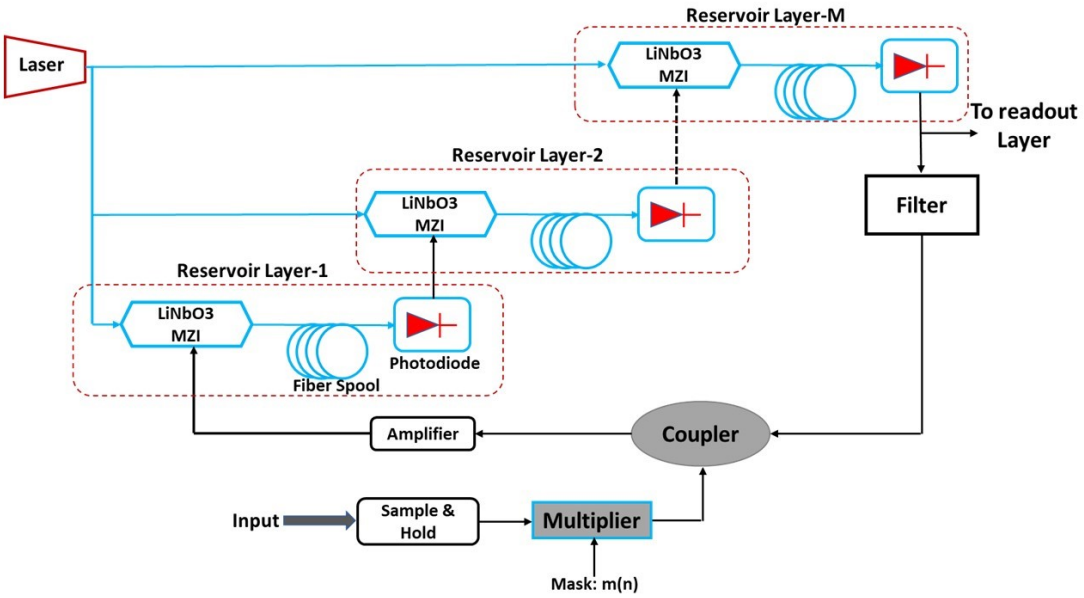


Figure 2-3 MReC Microarchitecture

Figure 2-3 presents the microarchitecture of proposed MReC architecture. The entire microarchitecture can be divided into three parts: input layer, reservoir layer, and readout layer. These layers work in a pipeline fashion to process an input. The details are as follows:

2.5.1.1. Input Layer

As shown in Figure 2-3, the input electronic signal, x(t) is sampled with a period of T_S using a 'sample & hold' circuit. This in turn converts each continuous-time task x(t) to a discretized piecewise constant function p(n) where $p(t) = p(n)$, $nT_S \leq t < (n+1)\,T_S$, n is a time step. Each discrete input $p(n)$ is multiplied with a periodic mask input $m(n)$ of period T_S. Here $m(n) = m_i(n)$ for $(i-1)\,(\frac{T_S}{N}) < n \leq (i+1)\,(\frac{T_S}{N})$; $i = 1, 2 \ldots N$; $m_i(n)$ is randomly

chosen from [−1, +1]. The result of this multiplication is a masked input $p_i(n)$ which drives the 1st reservoir layer in the 'reservoir computing segment'.

2.5.1.2. Reservoir Layer

Reservoir computing segment is the heart of our proposed MReC architecture. It comprises of 'M' reservoir layer as shown in Figure 2-3. We consider up to four reservoir layers as an example for this work. Each reservoir layer consists of LiNbO3 Mach Zehnder Interferometer (MZI) as NL node and an optical fiber spool to provide delay. We consider sinusoidal nonlinearity in our design. The laser source provides optical carrier to the MZI of each reservoir layer. The output from the coupler is fed to the MZI of first reservoir layer through an electronic amplifier. For a timestep 'n', masked input $p_i(n)$ from the coupler is fed to the MZI of first reservoir layer. The MZI converts $p_i(n)$ into a reservoir state S_i^1 where 1 stands for 1st reservoir layer and i =1, 2…N (as we consider *N* reservoir states). The optical fiber spool provides a delay of T_S which is same as the sample time of 'sample & hold' circuit. For each time step n, the photodiode in the reservoir layer converts each reservoir state S_i^1 from optical form to electronic form. The photodiode of each reservoir layer has an operating period of h second. Analytically, we can write $N = \frac{T_S}{h}$. The state of reservoir *i* in the first layer can be written as,

$$S_i^1(n) = Sin(\alpha S_i^1(n-1) + \beta m_i(n)x(n) + \emptyset) \tag{2.3}$$

The electronic output from the photodiode of one reservoir layer becomes an input to the next reservoir layer. This way, at any timestep n, i^{th} reservoir state of j^{th} reservoir layer can be written as,

$$S_i^j(n) = Sin(\alpha S_i^j(n-1) + \beta m_i(n)x(n) + \emptyset) \quad (2.4)$$

Here $\propto$ and β are feedback gains; $\emptyset$ is a bias value; and $m_i(n)$ represents the mask input. $\propto$, β, and $\emptyset$ are adjustable parameters. The MZI used in our design has sinusoidal non-linearity; hence the above equation is based on a *sin* function. One cycle of MReC architecture is defined as the time taken by a masked input $p_i(n)$ to travel from the 1st reservoir layer until the readout layer.

2.5.1.3. Output Layer

Using the photodiode from the last reservoir layer (here the M^{th} layer), all the N reservoir states $S_i^M(n)$ are fed to an offline computer. The predicted output is determined using the following equation.

$$O(n) = \sum_{i=1}^{N} W_i S_i^M(n) + W_{bias} \quad (2.5)$$

Here Wi is calculated using linear regression training by comparing $O(n)$ with target output $O'(n)$.

2.5.1.4. Pipeline Operation In MReC

Each reservoir layer in MReC amounts to a processing time of T_s . One masked input $p(n)$ can be processed by one reservoir layer at a time, keeping the rest of the reservoir layers idle. This leads to underutilization of MReC architecture. We introduce a pipeline approach to fully utilize all the reservoir layers in MReC. T_d unit of time after feeding $p(n)$ ($T_d \ll T_S$), MReC's 1st layer is fed with a masked input q(n) from another task. This will keep two reservoir layers busy at a time. We can process M number of tasks in MReC at a time where M is the total number of reservoir layers. As the time interval T_d

between two signals is negligible, one can assume that the overall throughput of MReC is Mx compared to single-layer photonic RC if M tasks are processed in a pipeline.

2.6. Experimental Methodology

We designed and synthesized optoelectronic components such as optical fiber spool, photodiode, coupler, MZI, and sampler using a commercial photonic design tool called IPKISS [38]. The synthesized components are used to design and simulate the proposed MReC microarchitecture using a CAD tool called Caphe. We integrate Caphe with Caffe [51], a C++ & Python based deep learning toolbox to rapidly build, train and evaluate machine learning models. Table 2-1 illustrates details of components used in our design. For power and area models, we use DSENT [52].

Table 2-1 Parametric Details Of Components Used In The Proposed Architecture

COMPONENTS	PARAMETERS	VALUES
LASER	Wavelength	1550nm
	Power	10W
MZI	Power	5W
FIBER SPOOL	Length	20m
	Delay	0.1us
	Power	Negligible
PHOTODIODE	Power	5watt
	Rise Time	15ps

Four widely used machine learning benchmark tasks namely, spoken digit recognition, Santa Fe time series prediction, non-linear channel equalization, and NARMA task are run on MReC using Caffe. For the purpose of simulation, we modify the Caffe toolbox to integrate the errors introduced due to photonic components. The system is then tested with the benchmarks mentioned. The details of the benchmarks are as follows.

2.6.1. Spoken Digit Recognition

Spoken digit recognition task is a widely-used classification task. The objective of the task is to classify ten spoken digits (0-9), each recorded ten times by five different persons. The dataset is obtained from [53]. The input $p_i(n)$ to the reservoir comprises of an 86-dimensional (i = 1, 2, ... 86) state vector with up to 130 time-steps. The number of variables N <= 350. This requires an input mask m_{ij} of matrix size $N \times 86$, where each element is chosen randomly from {-1, +1} with equal probabilities. $\sum_{i=1}^{86} m_{ji} p_i(n)$ (product of input and mask) is used to drive the reservoir. The metric to measure the performance of a system executing spoken digit recognition task is word-error-rate (WER). WER is the fraction of misclassified digits resulting from this task.

2.6.2. Santa Fe Time Series

A time-series is a sequence of periodic data points over a continuous time interval. In our experiment, we use Santa Fe financial time-series recorded from a far-infrared laser operating in chaotic state [54]. The goal of this experiment is to predict a data point one time-step ahead in the future. The dataset contains 10000 points and we use 4000 points

out of it. Prediction performance is evaluated based on the normalized mean square error (NMSE) defined as:

$$NMSE = \frac{1}{n}\sum_{i=1}^{n}\frac{(O_i' - O_i)}{\sigma O_i^2} \tag{2.6}$$

where O_i' and O_i are predicted and expected values at time step i, n is total number of time step, and σ is the standard deviation. Here, NMSE = 0 implies perfect prediction and NMSE = 1 indicates no prediction.

2.6.3. Non-Linear Channel Equalization

Equalization of communication channel is a way to facilitate reliable wireless communication. Whenever a signal is transmitted across a wireless communication channel, it encounters noise, channel effects (e.g. distortion, dispersion), and inter-symbol interference. Equalization of a wireless communication channel has been widely used as a benchmark task for RC simulation. The following two equations represent the relationship of the output s(n) a non-linear wireless channel to its input g(n).

$$z(n) = 0.08g(n+2) - 0.12g(n+1) + g(n) + 0.18g(n-1) - 0.1g(n-2) + 0.091g(n-3) - 0.05g(n-4) + 0.04gn(n-5) + 0.03g(n-6) + 0.01g(n-7) \tag{2.7}$$

$$s(n) = z(n) + 0.36z(n)^2 - 0.011z(n)^3 + d(n) \tag{2.8}$$

As depicted in Equation 2.7 and 2.8, s(n) encounters second-order $z(n)^2$ and the third-order $z(n)^3$ nonlinear distortions, and also additive Gaussian white noise d(n), which may result from the channel. s(n) is used as the final input to reservoir system.

2.6.4. NARMA Task

The NARMA task is one of the most widely used benchmarks in RC[55]. The input u(k) for this task consists of scalar random numbers, drawn from a uniform distribution in the interval [0, 0.5] and the target y(k + 1) is given by the following recursive formula:

$$y_{k+1} = 0.3y_k + 0.05y_k\left[\sum_{i=0}^{9} y_{k-i}\right] + 1.5u_k u_{k-9} + 0.1 \tag{2.9}$$

2.7. Results And Analysis

Using Caphe, we simulate the proposed design to perform photonic feature extraction of above-mentioned benchmarks. These benchmarks are input for the proposed MReC architecture. The output from photonic simulation is fed to Caffe [51] which acts as the readout layer. In the readout layer, feature classification/prediction takes place. We determine classification/prediction error rate, power consumption, area overhead, and throughput for all the tasks. Since this is the first attempt to a multilayer RC system, we compare our proposed multilayer system with our single layer system first. We then compare our single layer results with three state-of-the-art single layer photonic RC systems [35, 34, 33] to show that our system performs better.

2.7.1. Prediction Error Rate Comparison

2.7.1.1. Single Layer Vs Multilayer Proposed Architecture

We first evaluate prediction error rate for four benchmarks such as NARMA task, non-linear channel equalization task, isolated spoken-digit recognition task, and Santa Fe time series prediction task using a single-layer proposed RC as well as multilayer proposed

RC. The metric to determine prediction error rate differs from one task to other. For NARMA task, we consider normalized mean square error (NMSE) whereas for non-linear channel equalization task, we consider symbol error rate (SER) which is the fraction of misclassified symbols. For isolated spoken- digit recognition task, we determine word error rate (WER) which is the percentage of misclassified digits and for Santa Fe time series prediction, we evaluate NMSE which is the percentage of prediction error. For multilayer photonic RC, we consider 2-layer, 3-layer, and 4-layer of the proposed MReC architecture. Each layer of the multilayer RCs comprises of same number of reservoir nodes as that of a single-layer proposed RC. We consider number of reservoir nodes/layer, N=50 as most of the single-layer RC in literature use 50 nodes per layer for comparison. Table 2-2 presents prediction error rate of 2-layer, 3-layer, and 4-layer proposed RC normalized to a single-layer proposed RC for NARMA task, non-linear channel equalization task, isolated spoken-digit recognition task, and Santa Fe time series prediction task. For NARMA task, as we move from 1-layer to 4-layer, there is gradual reduction in the NMSE value from 0.082±0.0075 to 0.052±0.0045.

It is evident from the table that all the tasks show gradual reduction in prediction error rate with the rise in number layer of the proposed RC. NCQ task has a very small change in SER from 1-layer photonic RC to 4-layer photonic RC. This is due to the fact that NCQ task is prone to the noise introduced by a RC layer. This affects the overall reduction in SER value by the multilayer approach. In all the four cases, the prediction error rate saturates beyond the 3-layer architecture.

Table 2-2 Prediction Error Rate Comparison Of 1-Layer, 2-Layer, 3-Layer, And 4-Layer Proposed RC

	1-Layer	*2-Layer*	*3-Layer*	*4-Layer*
NARMA	NMSE= 0.082± 0.0075	NMSE= 0.071± 0.0075	NMSE= 0.058± 0.0065	NMSE= 0.052± 0.0045
NCQ	SER=0.002	SER=0.0019	SER=0.0017	SER=0.0015
ISDR	WER=0.9	WER=0.75	WER=0.62	WER=0.55
Santa Fe	NMSE= 0.092 ± 0.0075	NMSE= 0.08 ± 0.0065	NMSE= 0.067 ± 0.0045	NMSE=0.06 ± 0.0075

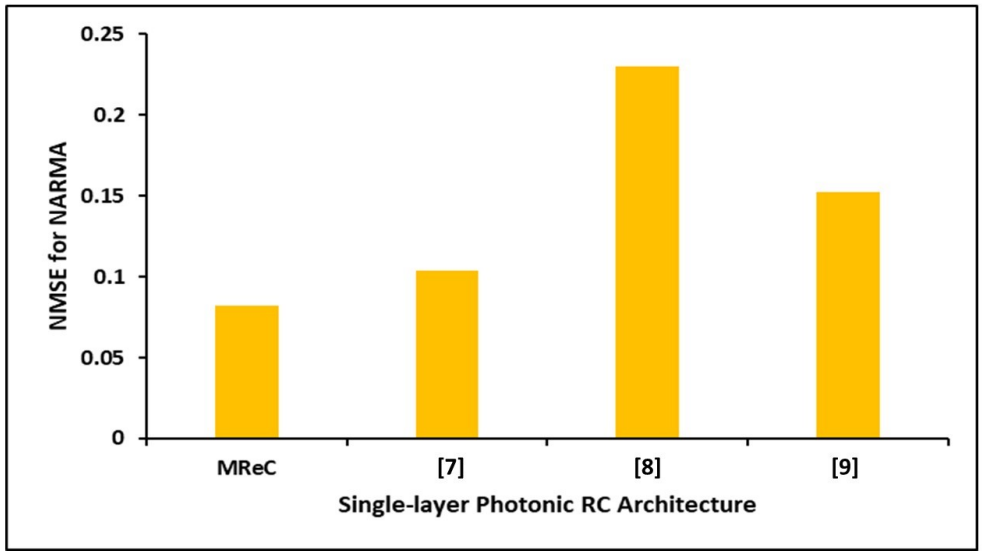
2.7.2. MReC Vs State Of The Art Architectures

As multilayer RC architecture is yet to be demonstrated in literature, we compare single-layer MReC with state-of-the-art architectures in [35, 34, 33]. The metric to compare prediction error rate differs from one benchmark task to another. For NARMA

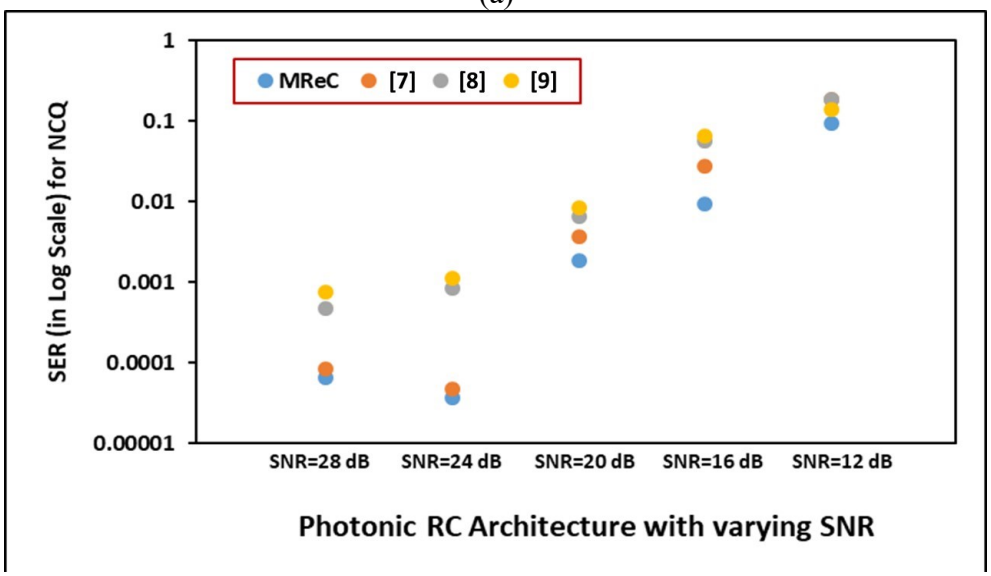
task, we compare normalized mean square error (NMSE) of the MReC architecture with [35, 34, 33]. Figure 2-4a illustrates comparison of NMSE of single-layer MReC architecture with [35, 34, 33]. For a fair comparison, the number of nodes in all the cases are chosen to be 50 (N=50). State-of-the-art photonic RC [4] performs with NMSE = 0.104 ± 0.012. The proposed MReC architecture with a single-layer shows improved performance with an NMSE = 0.082 ± 0.0075.

For "Non-linear channel equalization task", we determine symbol-error rate (SER) which is the percentage of misclassified symbols. As non-linear channel task is prone to channel noise, we also study the effect of a Gaussian noise (with zero mean) ranging from Signal-to-Noise (SNR) of 12 to 28dB. Figure 2-4b illustrates comparison of SER of single-layer MReC architectures with single-layer photonic RC in [35, 34, 33]. As evident from Figure 2-4b, MReC architecture with a single layer outperforms other photonic RC architectures irrespective of noise. With the rise in noise level in the reservoir layers, performances of photonic RCs in [35, 34, 33] degrade. When the noise is low (i.e. SNR is high) single-layer MReC demonstrates a low SER=$7x10^{-5}$ as opposed to a SER=$8x10^{-5}$ in [33] , a SER=$9x10^{-5}$ in [35] and a SER=$5x10^{-4}$ in [34] . When the noise is high (SNR=12 dB), proposed RC shows a SER = SER=$1x10^{-1}$, which is still better than SER=$1.5x10^{-1}$ in [33] , and SER=$2x10^{-1}$ in [35] and [34]. We evaluate word-error-rate (WER) which is the

fraction of misclassified digits when executing the isolated spoken-digit recognition task.



(a)

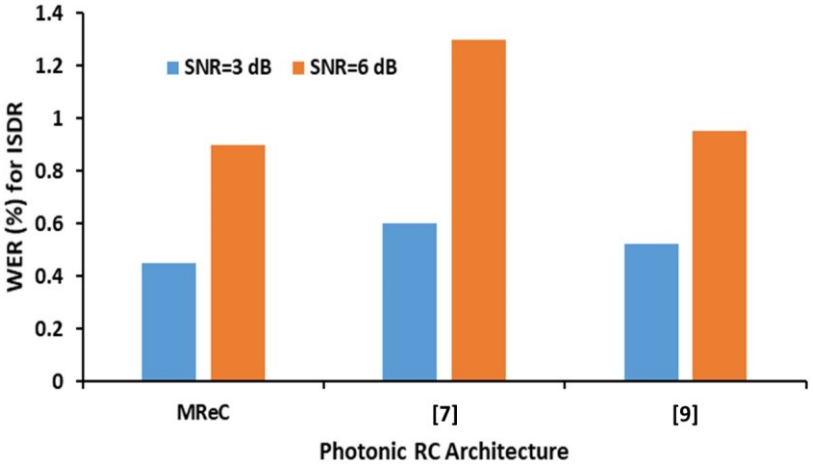


(b)

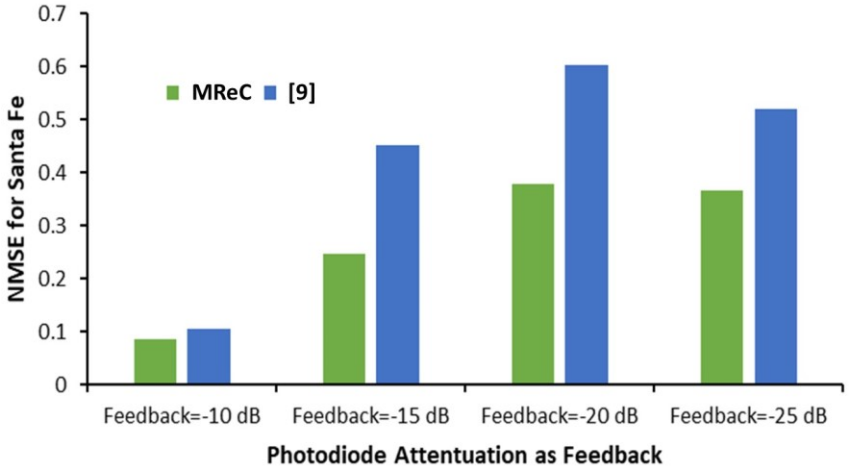
Figure 2-4 (A) NMSE Comparison For NARMA Task When N=50 In All Rcs, (B) SER Comparison For NCQ For Photonic Rcs With N=50 When SNR Is Varied From 12 To 28 Db

Figure 2-5a illustrates WER of single-layer MReC architecture as compared to [35] and [33] when number of reservoir nodes in each RC, N=50 while SNR = 3 dB and SNR = 6 dB. Photonic RC in [34] does not demonstrate results for isolated spoken digit recognition task. Therefore, we do not compare with [34] in this case. 1-layer MReC architecture shows lower WER compared to [35] and [33] . When SNR = 3 dB, proposed MReC with 1-layer has WER = 0.45% while [33] has WER = 0.52% and [35] has WER = 0.6%.

For Santa Fe task, we compared the NMSE result of MReC with that of in [33] as shown in Figure 2-5b. In Santa Fe task, a single data point needs to be predicted and hence, feedback from photodiode has a profound effect on photonic RC's accuracy. Therefore, we show NMSE results for a varying feedback parameter in terms of attenuation of photodiode. It is clearly evident that the MReC with one-layer outperforms the state-of-the-art design in [33]. Also, we obtain the best results for a feedback value of -10dB. The state-of-the-art silicon photonic diode does not produce an attenuation below -10dB. With the advancements in silicon photonic technology, if the attenuation is reduced, one may be able to predict task like Santa Fe with much better accuracy. From the above comparisons, we can conclude that MReC architecture with 1-layer outperforms other photonic RC architectures.



(a)



(B)

Figure 2-5 (A) WER Comparison Of 1-Layer MReC With [8] And [10] For Isolated Spoken-Digit Recognition Task (ISDR), (B) NMSE Comparison Of 1-Layer MReC With [10] For Santa Fe Task, W.R.T. Photodiode Attenuation As Feedback.

2.7.3. Energy Consumption Comparison

Table 2-1 illustrates all the optoelectronic components used in the proposed MReC architecture along with their parametric details. We use a standard optoelectronic power model called DSENT [52] to obtain the total power consumption in the MReC architecture. The total power consumption for a single-layer MReC architecture is approximately 50W (conservative estimate). This is lower than the state-of-the-art photonic RC architectures demonstrated in [35] which consumes 76W, in [34] which consumes 90W, and in [33] which consumes 200W. The higher power consumption in [35, 34, 33] is due to extra hardware usage e.g. more AWG in [35] and [33] , and analog readout in [34]. The total power consumption of MReC architecture is a function of laser power and MZI power consumption. Note that there is only one MZI per reservoir layer. The number of states or nodes in each layer i.e. N depends on the sampling of input and the length of delay line. A multilayer photonic RC requires slightly higher optical signal strength compared to single-layer to sustain a longer optical transmission time. A multilayer approach also requires extra number of MZIs which results in additional MZI driver power. Figure 2-6 depicts variation in power consumption from 1-layer to 4-layer photonic RC. A 4-layer MReC consumes 72W which is lower than the best reported result (76W) by a single-layer state-of-the-art system in [35]. We can conclude that the multilayer photonic RC design is scalable in terms of power consumption.

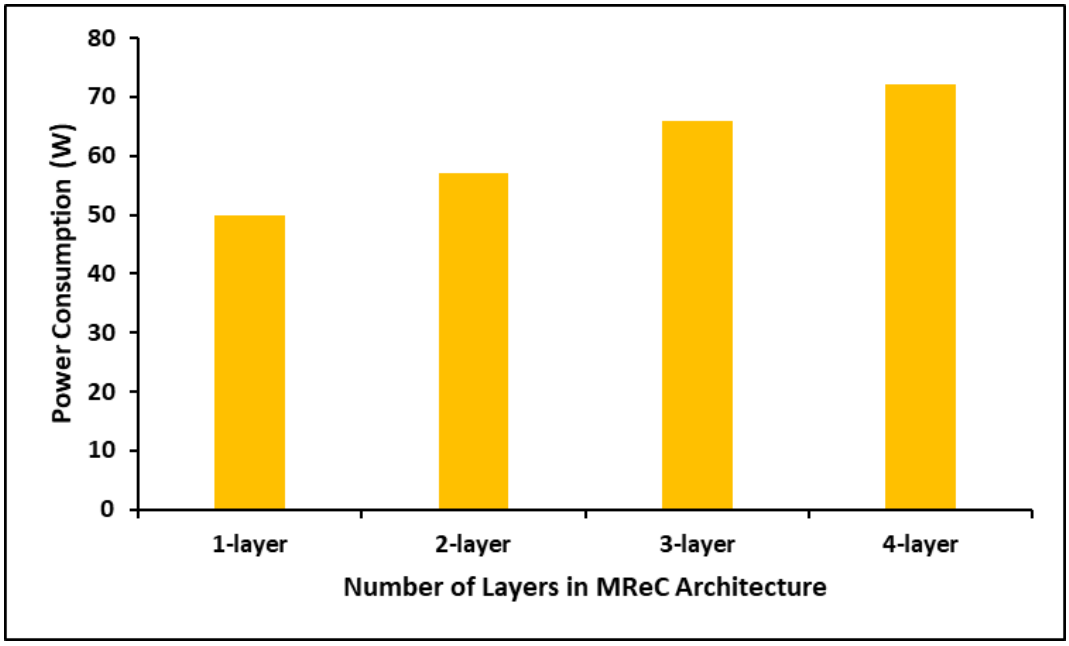


Figure 2-6 Power Consumption For MReC Architecture Variations.

2.8. Summary

In this chapter, we demonstrate MReC, novel multilayer photonic RC architecture for large-scale classification and prediction tasks. Each layer of the proposed architecture comprises of an MZI based nonlinearity and fiber optic delay line to emulate reservoir computing. We synthesize the proposed multilayer design using a standard photonic CAD tool called IPKISS [38] and execute three well-known classification benchmarks and one widely used prediction benchmark to demonstrate: (1) up to 26.8% reduction in prediction error rate when single-layer MReC is compared with state-of-the-art single-layer photonic RC architecture [34]; (2) up to 50% reduction in prediction error rate when 4-layer MReC is compared with state-of-the-art design and (4) up to 34.21% improvement in power

consumption compared to best reported result [33]. These improvements in MReC come at a cost of 12% area overhead.

The use of passive components in MReC allow it for a low power approach in photonic paradigm in addition to bandwidth advantages [56].

3. MULTILAYER PHOTONIC RESERVOIR COMPUTING USING TIME DIVISION MULTIPLEXING FOR PARALLEL COMPUTATION [2]

In the era of big data, large scale classification and prediction problems pose new challenges that the traditional Von-Neumann architecture struggles to address. This calls for implementation of new computational paradigms. Photonic reservoir computing is a promising paradigm for large-scale classification and prediction problems. Reservoir Computing (RC) has three layers: the input layer, reservoir layer and output layer. The reservoir layer is a random interconnected network of neurons that is independent of the task being performed using RC. This enables a particular reservoir to be used for multiple tasks, as only the output layer needs to be trained. The independent nature of reservoir layer provides an opportunity for parallel processing of multiple tasks at the same time. Unfortunately, the optoelectronic architectures for RC in literature do not exploit this capability. Therefore, in this chapter, we propose a multi-layer opto-electronic hardware architecture for parallel RC. Our architecture employs time division multiplexing to perform jobs in parallel.

3.1. Motivation

Reservoir Computing (RC) has emerged as a promising candidate to provide reduced training time with similar accuracy as that of deep RNN [57]. RC is a subset of

[2] adapted with permission from Hasnain, Syed Ali, Dharanidhar Dang, and Rabi Mahapatra. "Multilayer photonic reservoir computing architecture using time division multiplexing for parallel computation." Optoelectronic Devices and Integration IX. Vol. 11547. International Society for Optics and Photonics, 2020.

RNN. Like RNN, it also has two stages: feature extractor and feature classifier. However, unlike RNN, in RC the weights of feature extractor stage are randomly generated and remain fixed. Only the weights of feature classifier stage are trained to fine tune the output. As we know, RC has an input layer, a reservoir layer and an output layer. The reservoir layer comprises of hidden layers of artificial neurons whose weights were randomly generated. This makes the reservoir layer a generic computing tool, which is independent of the task being performed using RC. In theory, a single reservoir can therefore be used to perform multiple tasks[58].

3.2. Related Work

Several studies have argued that RC concept is similar to the workings of a brain, where an external stimulus excites the internal states of network and processes information[59, 25, 60]. Furthermore, the reservoir section of RC is independent of task being performed. This means that multiple tasks can use the same reservoir with a different readout layer for each task. A reservoir can be considered as a generic computation tool that is task independent. As a result, several tasks can be performed in parallel. As an example, recently, it has been shown that the same reservoir can be used for speech recognition as well as speaker recognition[26].

However, RC uses many non-linear nodes in its reservoir layer, hence the hardware implementation of RC using traditional approach has proved to be challenging[28]. To solve this challenge, the delayed feedback reservoir (DFR) model has emerged has a potential solution. In this model, only one non-linear node is employed along with a delay line[60, 61]. The non-linear node coupled with a delay line forms the reservoir. In this

approach there are many virtual neurons, and the single non-linear element is referred to as a node. An implementation based on optoelectronic components is ideally suited for RC as photonic components like Mach Zehnder Interferometer (MZI) have a nonlinear response and a delay line can provide delayed feedback. Following the DFR model, several silicon photonics based single layer RC implementations have been presented in the literature[33, 34, 35]. These single layer silicon photonic implementations have shown promising results with improvements in speed of up to 100x. However, the accuracy of these implementations is lower compared to RNN approaches. Intuitively, a multi-layer RC approach would improve the accuracy of RC system similar to deep neural network. Software implementation of multi-layer RC has indeed demonstrated that it can achieve identical accuracy as compared to RNN implementations at the expense of large execution time[37, 36]. Encouraged by these results, researchers have also proposed a hardware implementation of multi-layer RC that has shown superior performance to single layer systems[62]. However, none of these architectures demonstrate parallel processing of independent tasks. The reservoir layer has the capability to process multiple tasks in parallel and therefore an architecture for parallel RC can be designed.

3.3. Contribution

The major contributions of this chapter are as follows:

1. We propose a new time-shared multi-layer photonic architecture for RC to perform tasks in parallel.

2. Through experiments we show that our architecture can outperform some of the leading single layer architectures by up to 90% for NARMA task while performing analog speech recognition in parallel.
3. We also show that our proposed architecture closely matches the performance of leading multi-layer photonic RC architecture[14] with an increased error of 8% only due to parallel processing.
4. It is also shown that the proposed high-speed architecture has a power consumption of ~50W for a 4-layer network.

3.4. Time Shared Multilayer Photonic Architecture

The RC designs execute one input data stream per system. However, parallel computation with different data streams or tasks would be extremely beneficial when considering big data computing. One can argue that several independent tasks can be simultaneously executed using separate similar RC systems in parallel. However, such a system will have high power and area overhead. Any attempt to run multiple tasks in parallel on a single RC system has its own challenges as well. It is obvious that parallel computation of multiple photonic data streams on a single RC system will result in crosstalk and hence performance degradation. This is obviously not desirable.

On the other hand, if we carefully study the single node photonic computing model, such as the one shown in figure 3-1, we notice that the internal states that

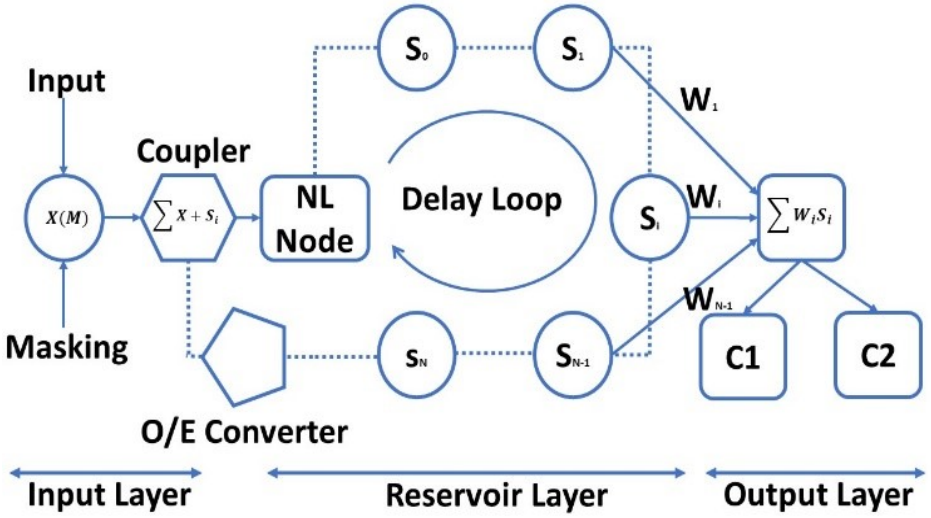


Figure 3-1 Single Node Photonic Model With Internal States Corresponding To Virtual Neurons

correspond to virtual neurons can be related to different tasks. The reservoir will simply process them and provide excited states as input to read out layers.

Therefore, we investigate the design for a new photonic RC system that overcomes the multitasking demands of big data applications and also satisfies the aforementioned technological needs (power, area, and crosstalk). This chapter introduces a time-multiplexed RC system based on a multilayer RC system in which time-division-multiplexing (TDM) is integrated with the input layer to execute multiple tasks in parallel[63, 64]. Fig 3-2. , reprinted with permission from [65], shows the proposed architecture. The details of the architecture are as follows.

3.4.1. TDM Integrated Input Layer

In the input layer of proposed architecture, multiple inputs are first sampled using a a sample & hold (S/H) circuit. The S/H circuit is controlled using an electronic control block that runs a round robin algorithm giving each input a sampling time. The sampling converts a continuous signal *x(t)* to a discrete time signal *p(n)*.

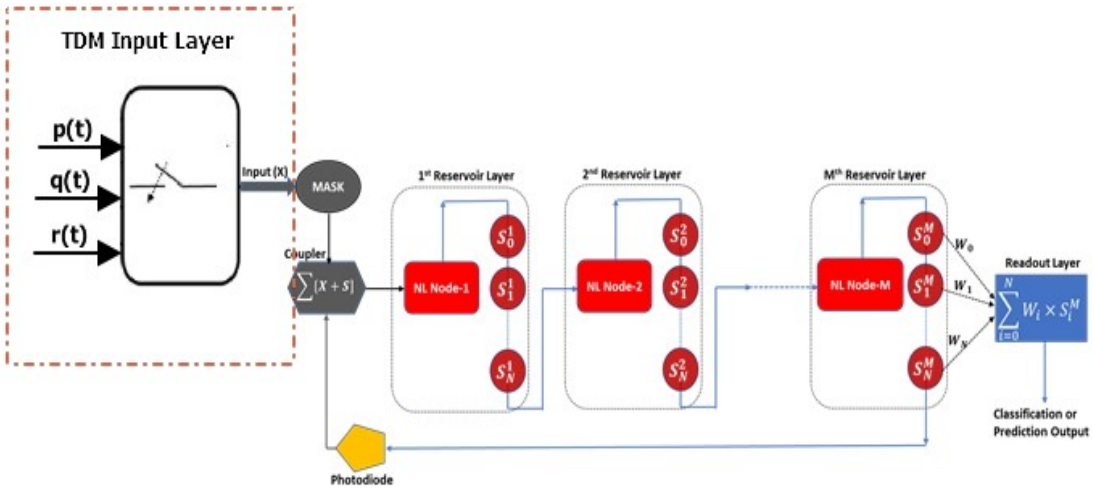


Figure 3-2 Time Shared Multi-Layer Photonic Reservoir Computing Architecture

Let $p_l(n)$ represent the *N* input signal, where $1 \leq l \leq P$. The input layer obtains a masked version of input signals by multiplying the signals with a random masking signal $m(n)$. The masking signal is randomly chosen from [-1,+1]. The result of this multiplication is a masked input $p_l'(n)$ which drives the reservoir layer in the 'reservoir computing segment'.

The random masking function is analogous of the random weights that are applied in the hidden layers. They help in generating the randomness and a dynamical response of the reservoir. The coupler unit, combines any past state with our current state and serves

as a feedback mechanism. This is analogous to the internal memory of an RNN. The S/H circuit runs a round robin algorithm, sampling each task for a certain period of time, before moving on to sample the next task.

3.4.2. Reservoir Layer

Our proposed architecture uses multiple reservoir layers. Each layer has one MZI and a single delay line[66]. The masked discrete time inputs are fed into the MZI to pass virtual states into the delay line[67, 68, 69]. Since the input layer uses a round robin algorithm to sample each of the input signals, it divides the virtual states inside a reservoir layer among different tasks. The electronic block can be programmed such that at discrete time *n,* input signal of task one (l=1) is sample and passed to reservoir layer and at discrete time $n+1$, input signal of task 2 (l=2) is passed to reservoir layer. The system cycles between the tasks in a round robin method distributing the resources in an equal manner. The resource distribution can be unequal if desired. For each reservoir we select the virtual neurons states to be $N = 50$. The selection is made based on the response time of the slowest component. For an electronically tunable delay line, made of a waveguide of length 2cm, a delay of 660ps is achieved through electronic tuning[70]. Each sampled input is of length 13.2ps.

The reservoir layer in our architecture consists of LiNbO3 Mach Zehnder Interferometer (MZI). The non-linearity considered in the design is sinusoidal. For a timestep n and task l, masked input $p_l(n)$ from the coupler is fed to the MZI. The MZI converts $p_l(n)$ into a reservoir state S_i^l where l stands for task being processed and $i = 1, 2 \ldots N$ (as we consider N reservoir states). At the end of each delay line is a photodiode. It

converts each state back to electrical form and feeds it to the coupler. The photodiode of each reservoir layer has an operating period of h second. The state of reservoir in any sublayer can be written as:

$$S_i^l(n) = Sin(\alpha S_i^l(n-1) + \beta m_i(n) p_l(n) + \emptyset) \quad (3.1)$$

Here $\propto$ and β are feedback gains; $\emptyset$ is a bias value; and $m_i(n)$ represents the mask input. $\propto$, β, and $\emptyset$ are adjustable parameters. The MZI used in our design has sinusoidal non-linearity; hence the above equation is based on a sin function.

3.4.3. Output Layer

The output of the reservoir layer is fed to an offline computer. The final output of a task l is given by the following equation:

$$f(n)^l = \sum_{i=1}^{N} W_i S_i^l(n) + W_{bias} \quad (3.2)$$

Where W_i and W_{bias} are trainable parameters that can be trained using linear regression.

3.5. Experimental Methodology

To Evaluate the architecture, we used synthesized components from the MReC Architecture [62], except for the delay line. Table 3-1 illustrates details of components used in our design. To simulate the Time-Shared reservoir model, we modified the Oger toolbox such that the nodes within the reservoir layer are divided among tasks. An electronically tunable delay line of length 2cm and delay 660ps was modeled in the architecture. The proposed architecture is evaluated for four layers of Reservoir with each layer containing N=50 nodes. We ran experiments using two benchmarks popular in the

area of Reservoir Computing: NARMA task and Speech Recognition. The details of the benchmarks are as follows

Table 3-1 Parametric Detail Of Components For Time Division Multiplexing Integrated RC Architecture

Components	Parameters	Values
Laser	Wavelength	1550nm
	Power	10W
MZI	Power	5W
Photodiode	Power	5watt
	Rise Time	15ps

3.5.1. NARMA Task

The Nonlinear Autoregressive Moving-average (NARMA) task is one of the most widely used benchmarks in RC[55]. The input u(k) for this task consists of scalar random numbers, drawn from a uniform distribution in the interval [0, 0.5] and the target y(k + 1) is given by the following recursive formula:

$$y_{k+1} = 0.3y_k + 0.05y_k\left[\sum_{i=0}^{9} y_{k-i}\right] + 1.5u_k u_{k-9} + 0.1 \tag{3.3}$$

Prediction performance for this benchmark is evaluated based on the normalized mean square error (NMSE) defined as:

$$NMSE = \frac{1}{n}\sum_{i=1}^{n}\frac{(O_i' - O_i)}{\sigma O_i^2} \tag{3.4}$$

where O_i' and O_i are predicted and expected values at time step *i*, *n* is total number of time step, and σ is the standard deviation. Here, *NMSE = 0* implies perfect prediction and *NMSE = 1* indicates no prediction.

3.5.2. Analog Speech Recognition

While studying the performance of proposed architecture we also employ an open source analog speech recognition dataset available on GitHub[71]. The data set is quantized first and then feed to the network. The performance metric for this task is BER which is the bit errors per unit time.

3.6. Results

To test the system, we designed a set of three experiments:

1. To get a base case, we first measure the standalone performance of the system. This is the configuration in only one task is run on the proposed architecture.
2. To evaluate performance during parallel processing of the proposed architecture, we run both the tasks in parallel, with sampling time divided equally among the tasks. The results of this experiment are reported as T-1
3. A third experiment was conducted with one of the tasks given a longer sampling time using the round robin algorithm. The results of this experiment are reported as T-2

3.6.1. Comparison Using NARMA Task

Figure 3-3 shows the results for NARMA task. The standalone configuration performs the best. However, experiment T-1 shows that the system performs NARMA task in parallel with speech recognition with a minor performance degradation. In

experiment T-1 the system assigns 50% of states to each task. Experiment T-2 shows the result for the case when NARMA task is given less priority compared to analog speech recognition task. In case of T-2 the number of states assigned to NARMA are 20% less than that for analog speech recognition task.

3.6.2. Comparison Using Speech Recognition

Figure 3-4 shows the performance results of speech recognition task in similar scenarios. The standalone performance again serves as the base case. In case of experiment T-1, we again see an increased error, however this is because of the smaller number of states that are assigned to the task. In fact, in T-1 the number of states for each task are 50% less compared to standalone scenario. In case of experiment T-2, analog speech is given a priority and we see a jump in performance.

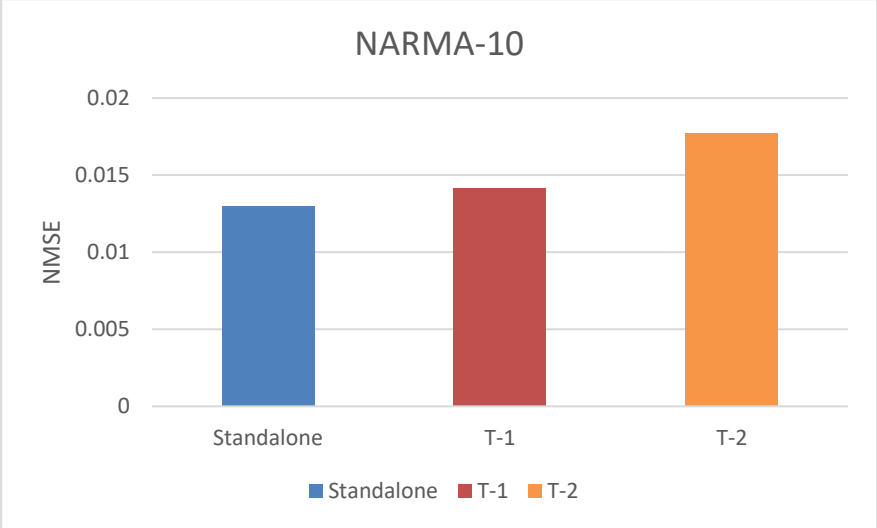


Figure 3-3 Performance Results For NARMA-10 Task While Using Time Shared Multi-Layer Reservoir System

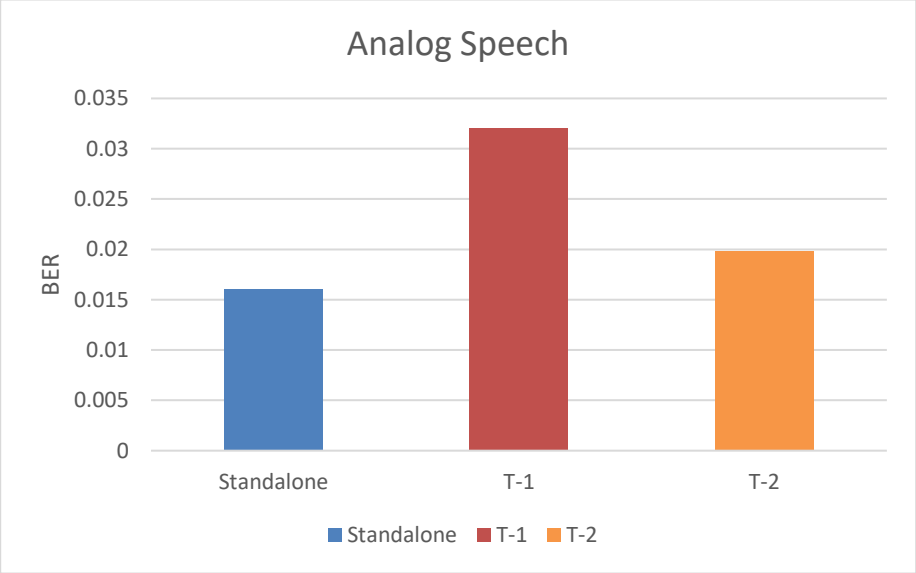


Figure 3-4 Performance Results For Analog Speech Recognition For Different Configurations

3.6.3. Comparison With Other Architectures

Table 3-2 shows the performance of the proposed architecture in comparison with other state of the art architectures in standalone configuration and has only a ~8% loss of accuracy while performing two tasks in parallel with equal priority. We notice that it outperforms other architectures. This is due to the use of low loss and short delay lines in our architecture. An analysis of the propagation loss in delay lines shows that propagation loss affects the accuracy of the system, hence by using state of the art, compact and low loss delay line of just length 2cm, we can make the propagation loss negligible and gain higher performance. This is shown in Figure 3-5. The proposed high-speed architecture has a power consumption of ~50W for a 4-layer network.

Table 3-2 Comparison Of Results For Different Photonic RC Architectures For The Common NARMA Benchmark

Architectures	NARMA (NMSE)
Standalone	0.013
T-1	0.0141
T-2	0.0177
MReC [62]	0.05
Brunner[33]	0.16
Vinker[35]	0.104
Duport [34]	0.24

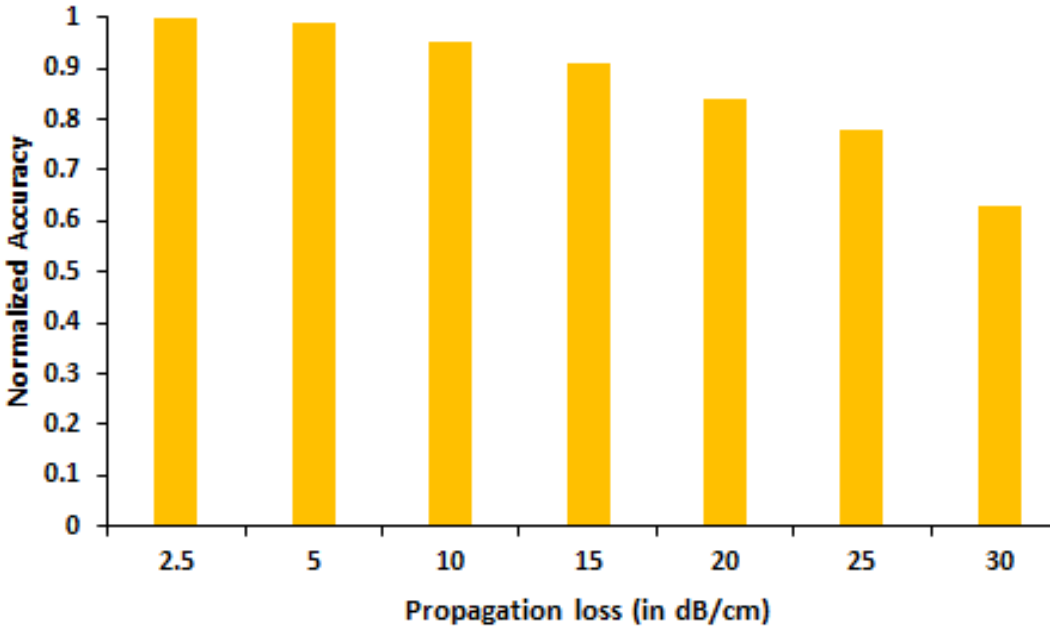


Figure 3-5 Analysis Of Propagation Loss In A Delay Line

3.7. Summary

Reservoir Computing systems have a reservoir layer that is made up of neurons with random connections and random weights that do not need training. The random weights and connections are independent of the task being performed using RC. The reservoir can therefore perform several tasks in parallel. In this paper we proposed a new architecture for photonic Reservoir Computing that uses Time Division Multiplexing (TDM) in its input layer to exploit the opportunity to perform multiple jobs in parallel using the same RC system. Our proposed system uses multiple reservoirs made up of MZI and low loss delay lines. Through simulations on NARMA and speech recognition, we show that our architecture can outperform some of the leading single layer architectures by up to 90% for NARMA task while performing analog speech recognition in parallel and closely matches the performance of leading multi-layer photonic RC architectures with an increased error of 8% due to parallel processing. The proposed high-speed architecture has a power consumption of ~50W for a 4-layer network.

4. ON-CHIP PARALLEL PHOTONIC RESERVOIR COMPUTING USING MULTIPLE DELAY LINES[3]

Silicon-Photonics architectures have enabled high speed hardware implementations of Reservoir Computing (RC). With a delayed feedback reservoir (DFR) model, only one non-linear node can be used to perform RC. However, the delay is often provided by using off-chip fiber optics which is not only space inconvenient but it also becomes architectural bottleneck and hinders to scalability. In this chapter, we propose a completely on-chip photonic RC architecture for high performance computing, employing multiple electronically tunable delay lines and micro-ring resonator (MRR) switch for multi-tasking.

4.1. Motivation

The hardware implementations for Reservoir Computing have proved to be difficult due to many non-linear nodes involved in the system. A potential solution for the hardware architectures of RC is the delayed feedback reservoir (DFR) model[60, 72, 30]. In this model, the reservoir is implemented using a single non-linear node and a delay line. This model can be implemented in optoelectronic architectures with relative ease, and provides high speed and high bandwidth of photonics systems. Therefore, employing this

[3]adapted with permission from Copyright © 2020, Hasnain, Syed Ali, and Rabi Mahapatra. "On-chip Parallel Photonic Reservoir Computing using Multiple Delay Lines." 2020 IEEE 32nd International Symposium on Computer Architecture and High-Performance Computing (SBAC-PAD). IEEE, 2020.

model, several architectures of RC have been proposed using silicon photonics [33, 62, 34, 35]. These architectures employ a fiber optic delay line and a non-linear photonic component to perform RC. Due to the use of fiber optics, these cannot be classified as completely on chip architectures.

Furthermore, these architectures do not exploit the in-built opportunity for parallelism in RC systems. In a reservoir computer, the reservoir provides a dynamic random projection of input which is then used in classification and prediction tasks. Since a reservoir is a network of random inter connections and weights, multiple tasks can be executed using the same reservoir. The randomness of a reservoir is further discussed in Section III.

Therefore, we propose a DFR based photonic architecture for reservoir computing, while carefully addressing these problems. Our proposed architecture employs multiple waveguide-based on chip electronically tunable delay lines. The on-chip delay lines reduce the size of the system while also providing high performance parallel processing.

4.2. Related Work

4.2.1. Photonic RC Architectures

In literature, multiple single layer photonic RC architectures have been proposed [33, 34, 35]. These architectures are based on DFR model and employ fiber optics and Mach Zehnder Interferometer as the non-linear node. The delay line length varies from 20m to 1.7Km. These architectures employ a fiber spool and are only for single layer reservoir. Since they have off chip components, they cannot be classified as completely

on-chip architectures. The off-chip components also pose a challenge to scalability of such systems, especially when multi-layer architectures are considered.

4.2.2. Delay Lines

On chip optical delay lines have been an area of research for device level researchers. In literature, several optical delay lines have been proposed. The delay lines can be in form of guided resonant buffers [73], couple resonator waveguide based [74, 75, 76]or photonic crystal line defect waveguides[77]. However, in these approaches the loss can be very high compared to a spiral waveguide. The delay can be up to 100ps. As a compromise between area and loss a new class was proposed in [78]. The core concept behind this class is based on time delay spectrum of grating waveguides by apodizing the gratings' profile. Another low loss, compact and fast, wavelength independent and electronically tunable delay line was proposed in [70]. The delay can be as high as 660ps. While these advances in optical delay line are still an ongoing effort, they provide an opportunity for replacing the large fiber spools that are employed in photonic architectures based on delayed feedback.

4.2.3. Photodiodes

Photodiodes are designed for specific band as well as a rise/fall time. Multiple new types of photodiodes have been proposed in literature [79, 80] using different materials. For example, [80] uses InGaAs with nitrogen ion planted and operate in 1-67GHz band with a rise time of just 2ps. A graphene-based chip integrated photodetector was proposed with operating speed higher than 20GHz[81]. Further advances in the field were made by proposing a 100GHz plasmonic photodetector [82]. Plasmonic devices are based on

interaction of optical frequency electromagnetic field oscillations interacting with free electrons. The response time of the device is less than ~5ps. In our proposed architecture we conservatively design our system employing multiple photodiodes. Any photodiode with rise time less than 15ps and capable of operating in infrared band can be used.

4.3. Contribution

The contributions of this chapter are as follows:

1. We propose a new architecture for on-chip parallel photonic reservoir computing employing multiple electronically tunable delay lines along with an MRR switch for delay line selection.
2. Through simulations we show that the proposed architecture is up to 84% more accurate compared to a leading architecture in [62] while executing NARMA task alone and 80% more accurate when executing two tasks in parallel. It outperforms other architectures presented in literature.
3. We also show that the proposed architecture performs 46% more accurate compared to an RC architecture employing Time Division Multiplexing (TDM) at input layer to execute tasks in parallel[83].
4. It is shown that the architecture removes the off-chip fiber optics-based delay line at the cost of 0.0184 mm2 of on chip area. The power overhead is just 26mW.

4.4. Multiple Delay Line Based Photonic Rc

In our proposed architecture, we use an additional MRR switch to route the output of the non-linear node to one of the many delay lines. Such switches have been presented in literature[41]. This enables the parallel processing of different tasks. We also employ

an electronically tunable delay line [70] and an additional electronic control module. The proposed architectural schematic is shown in Fig 4-1, which has been reprinted with permission from [84].

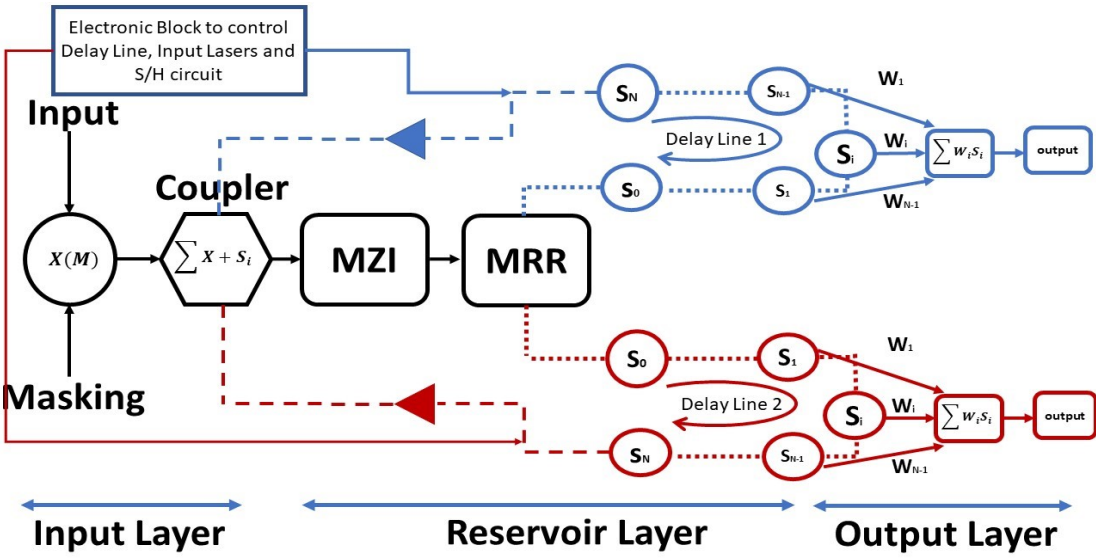


Figure 4-1 Schematic Of Proposed Architecture

4.4.1. Input Layer

The input electronic signal, x(t) is sampled with a period of T_S using a 'sample & hold' circuit. The S/H circuit is controlled using the electronic control block. This in turn converts each continuous-time task x(t) to a discretized piecewise constant function p(n) where $p(t) = p(n)$, $nT_S \leq t < (n+1)\,T_S$, n is a time step. Each discrete input $p(n)$ is multiplied with a random mask input $m(n)$ of period T_S. Here $m(n) = m_i(n)$ for $(i-1)\,(\frac{T_S}{N}) < n \leq (i+1)\,(\frac{T_S}{N})$; $i = 1, 2 \ldots N$; $m_i(n)$ is randomly chosen from $[-1, +1]$. The result of this multiplication is a masked input $p_i(n)$ which drives the reservoir layer in the 'reservoir computing

segment'. The random masking function is analogous of the random weights that are applied in the hidden layers. They help in generating the randomness and a dynamical response of the reservoir. The coupler unit, combines any past state with our current state and serves as a feedback mechanism. This is analogous to the internal memory of an RNN. The S/H circuit runs a round robin algorithm, sampling each task for a certain period of time, before moving on to sample the next task.

4.4.2. Reservoir Layer

Proposed architecture divides the reservoir layer into sublayers: one for each task that is being performed. Each sublayer consists of a delay line of its own which serves as the reservoir. An MRR based switch is used to select the delay line as shown in Fig 4-2. The MRR is electronically tunable such that when it is in OFF state its resonance frequency is different as compared to ON state. A single laser source of wavelength 1550nm is used. The delay line for different tasks is selected by means of an electronic block. The electronic block can be programmed such that for task i, the MRR is set OFF and delay line d_i is selected. At t_0, the system samples the input of task $i = 1$ and sends it to delay line d_1. At $t_{1,}$, it switches to task $i = 2$ and turns the MRR ON and selects delay line d_2. The system cycles between the tasks in a round robin method distributing the resources in an equal manner. The resource distribution can be unequal if desired. For each reservoir we select the virtual neurons states to be $N = 200$. The selection is made based on the response time of the slowest component. For an electronically tunable delay line, made of a waveguide of length 2cm, a delay of 660ps is achieved through electronic tuning[70].

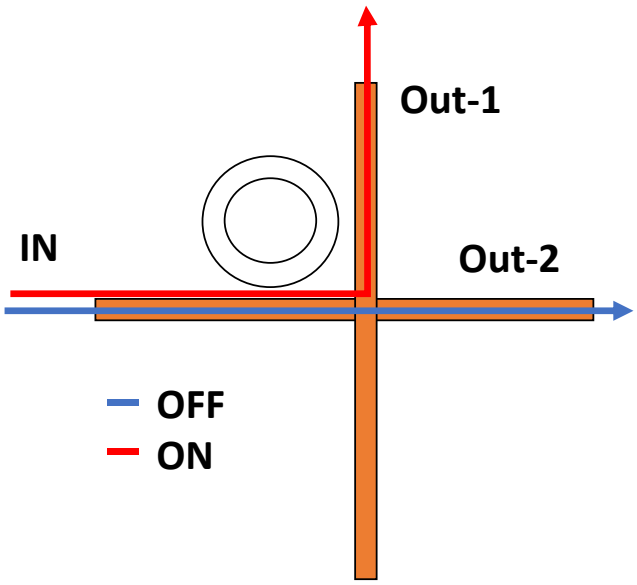


Figure 4-2 An Electronically Tuned MRR Switch To Direct Light Based On ON/OFF State

The reservoir layer consists of LiNbO3 Mach Zehnder Interferometer (MZI)[85]. The non-linearity considered in the design is sinusoidal. For a timestep 'n' and task 'n+1', masked input $p_i(n)$ from the coupler is fed to the MZI. The MZI converts $p_i(n)$ into a reservoir state S_i^{n+1} where n+1 stands for task being processed and $i = 1, 2 \ldots N$ (as we consider N reservoir states). At the end of each delay line is a photodiode. It converts each state back to electrical form and feeds it to the coupler. The photodiode of each reservoir layer has an operating period of h second. The state of reservoir in any sublayer can be written as:

$$S_i^{n+1}(n) = Sin(\alpha S_i^{n+1}(n-1) + \beta m_i(n)x(n) + \emptyset) \qquad (4.1)$$

Here $\propto$ and β are feedback gains; $\emptyset$ is a bias value; and $m_i(n)$ represents the mask input. $\propto$, β, and $\emptyset$ are adjustable parameters. The MZI used in our design has sinusoidal non-linearity; hence the above equation is based on a sin function.

4.4.3. Output Layer

All the states from each of the delay lines, can be converted to electrical form using the photodiode. They are then fed to an offline computer. The predicted output is determined using the following equation.

$$O(n) = \sum_{i=1}^{N} W_i S_i^{n+1}(n) + W_{bias} \tag{4.2}$$

Here Wi is calculated using linear regression training by comparing $O(n)$ with target output $\mathrm{O}'(\mathrm{n})$.

4.5. Evaluation Of Architecture

To evaluate our architecture thoroughly, we run multiple sets of experiments. In the first set we run different configurations of our architecture. A configuration is defined by length of delay in delay lines, which is electronically tunable and the number of tasks being run. For the scope of this work, we consider standalone systems which use only one delay line and run one task at a time and systems with two delay lines capable of running two tasks at a time. While our design is scalable for more delay lines and more tasks, this requires design of a multi-stage multiplexer based on MRR switches, which is out of the scope of this work and is an ongoing effort. We also vary delay lengths between tasks to analyze the effect on different tasks. In a second set of experiments, we compare our architecture to other single layer state of the architectures. In particular we compare our

work to a single layer photonic architecture based on TDM approach [83]. The details of our experimental methodology and benchmarks is discussed first, before discussing the results.

4.5.1. Experimental Methodology

We designed and synthesized optoelectronic components such as photodiode, coupler, MZI, and sampler using a commercial photonic design tool called IPKISS[38]. The synthesized components are used to design and simulate the proposed microarchitecture model. Python based deep learning toolbox, Oger and Keras[86, 87], were used to rapidly build, train and evaluate reservoir computing models.

4.5.2. Benchmarks

4.5.2.1. NARMA Task

The Nonlinear Autoregressive Moving-average (NARMA) task is one of the most widely used benchmarks in RC. The input u(k) for this task consists of scalar random numbers, drawn from a uniform distribution in the interval [0, 0.5] and the target y(k + 1) is given by the following recursive formula:

$$y_{k+1} = 0.3y_k + 0.05y_k\left[\sum_{i=0}^{9} y_{k-i}\right] + 1.5u_k u_{k-9} + 0.1 \tag{4.3}$$

Prediction performance for this benchmark is evaluated based on the normalized mean square error (NMSE) defined as:

$$NMSE = \frac{1}{n}\sum_{i=1}^{n}\frac{(O_i' - O_i)}{\sigma O_i^2} \tag{4.4}$$

where O_i' and O_i are predicted and expected values at time step *i*, *n* is total number of time step, and σ is the standard deviation. Here, *NMSE = 0* implies perfect prediction and *NMSE = 1* indicates no prediction.

4.5.2.2. Analog Speech Recognition

While studying the performance of proposed architecture we also employ an open source analog speech recognition dataset available on GitHub [71]. The performance metric for this task is BER which is the bit errors per unit time.

4.6. Results

4.6.1. Different Configurations Of The Proposed Architecture

In this set of experiments, we simulated the system first in a standalone configuration for one task. A single task is used as input and only one delay line is employed. This serves as our base case, and is reported as the 'standalone' case. Multiple tasks in parallel with varying lengths of delay line serve as other configurations. In first scenario two tasks were run with delay lines of equal delay and in second scenario the delays of the delay lines were unequal i.e. one delay line had a longer delay than the other one. The delay can be tuned electronically. In such a scenario we also employed a TDM based approach and divided the time to sample unequally among tasks. The results of first configuration are reported as E-1 while configuration 2 are reported as E-2.

Figure 4-3 shows the results for NARMA task. The standalone paradigm allows for a larger delay line and hence more states inside delay line. It performs the best. However, with only a slight change in performance we can run two tasks in parallel by using two delay lines based on proposed architecture. The length of each delay line in

experiment E-1 was half that of standalone case. A system with unequal delay lengths can prioritize a task execution when multi-tasking. This is achieved by electronically tuning the delay line for smaller or longer delay and giving the task more time in round robin

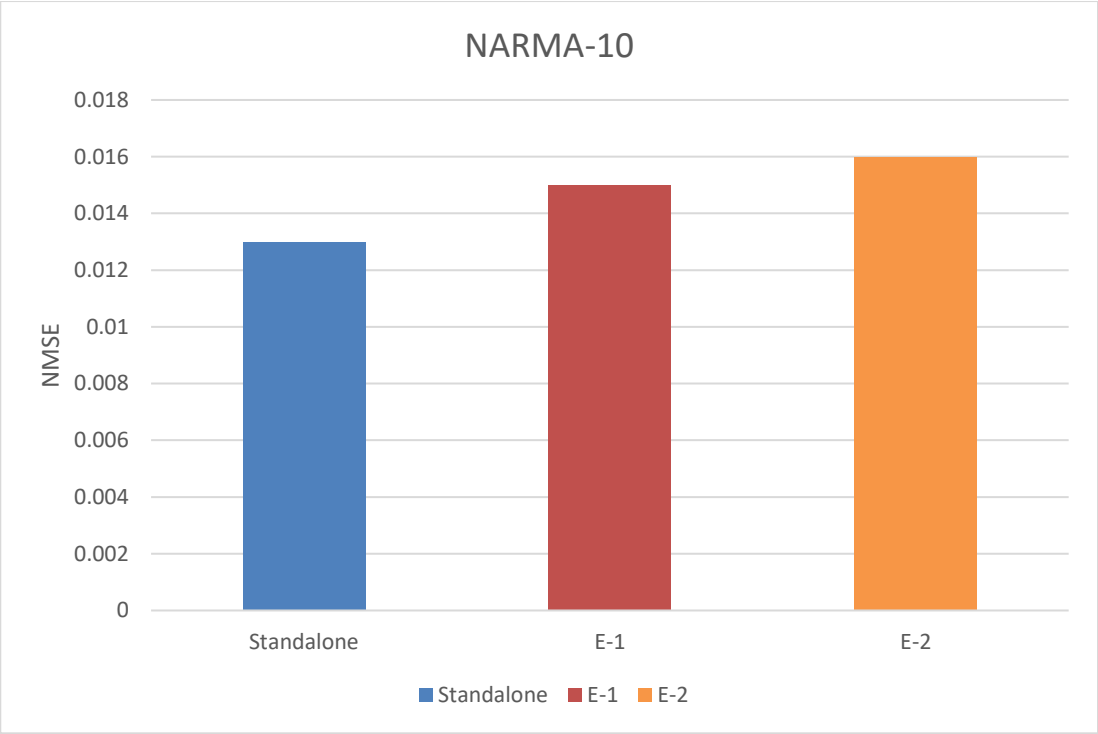


Figure 4-3 NARMA Results For Different Configurations

sampling. Experiment E-2 shows the result for the case when NARMA task is given less priority compared to analog speech recognition task. In case of E-2 the delay for NARMA is 20% less than the delay for analog speech recognition task.

Figure 4-4 shows the performance results of speech recognition task in similar scenarios. The standalone performance is better than both other cases. In the case of experiment E-1 when two tasks are run on a system with equal delay lines, we see a drop

in performance. This is because of the smaller number of states that are assigned to the task, due to shorter delay line in this configuration. In fact, in E-1 the number of states for

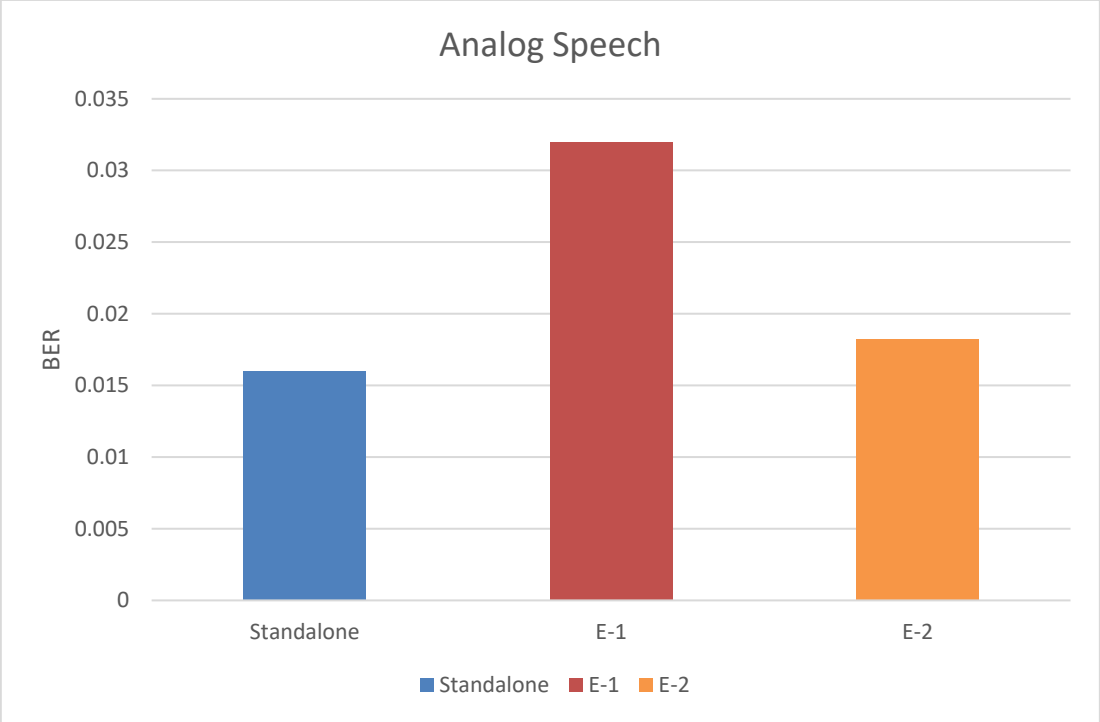


Figure 4-4 Analog Speech Recognition For Different Configurations

each task are 50% less compared to standalone scenario. In case of experiment E-2, analog speech is given a priority and larger delay line. It was observed that performance came close to stand alone system with only 20% longer delay line.

4.6.2. Comparison With Other State-Of-The-Art System

In this experiment we compare our results with other proposed architectures for photonic RC. Table 4-1 compares our results with other state of the art architectures for photonic RC. Each of these architectures uses single layer to execute a single task. We

Table 4-1 Results For Proposed Architecture vs Different Photonic RC Architectures For The Common NARMA Benchmark

ARCHITECTURES	NARMA (NMSE)
STANDALONE	0.013
E-1	0.015
E-2	0.016
MReC[62]	0.05
Brunner[33]	0.16
Vinker [35]	0.104
Duport[34]	0.24

notice that even in E-1 and E-2 configuration, our proposed architecture performs better. The gain in performance in the proposed architecture is due to use of low loss delay lines. The system is 84% more accurate than leading architecture in [62] when in standalone configuration, while it is up-to 80% more accurate while performing two tasks in parallel.

Dang et al [83] proposed an architecture using TDM. While they use only one delay line, the virtual nodes inside delay line are divided among different tasks based on sampling time. The reader must note that the performance of architecture in [83] is only better than that of leading architecture in [62] because [83] is using virtual nodes N=200 while [62] uses N=50. With N=200, architecture in [62] will outperform [83]. Our

approach is different from their approach in the following ways: a) To avoid cross talk and noise induced performance degradation, we use multiple on chip delay lines for parallel processing instead of a single fiber spool. b) Only TDM is employed in [83], where

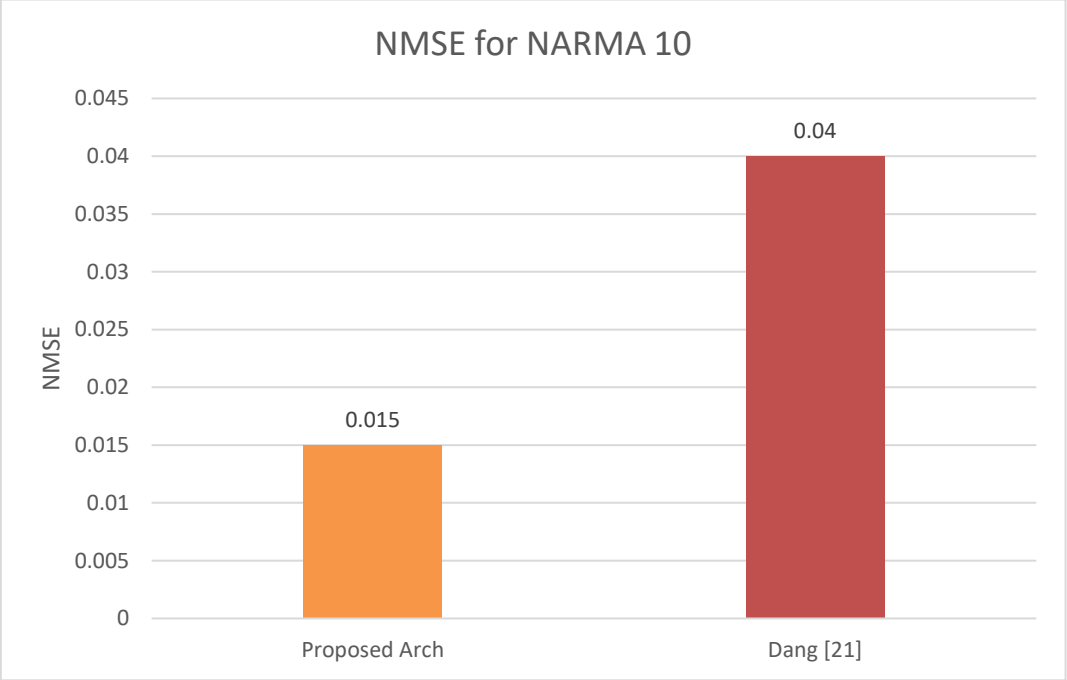


Figure 4-5 Comparison Between Proposed Architecture And TDM Based Approach Presented In [83]

as our architecture uses an MRR switch for delay line selection. Figure 4-5. shows a comparison between our proposed architecture and the TDM based architecture for NARMA under same conditions. The architecture proposed here, outperforms the architecture in [83].

The proposed architecture has less overall area compared to any single node computer for RC. This is because all other systems proposed in literature employ an off-chip fiber spool. Table 4-2 compares the dimensions of delay line in different

architectures. All other architectures employ delay lines within the range of 20m-1.7km. Our architecture uses on-chip delay line which results in on-chip 0.0184 mm^2 area

Table 4-2 Comparison Of Size Of Delay Lines In State-Of-The-Art Photonic RC Architectures

Architecture	Technology	Delay Line Length
Proposed Architecture	Silicon on Insulator waveguide	L=2cm, W=460nm, H=0.9um
MReC[62]	Fiber Spool	20m
Dang[83]	Fiber Spool	20m
Brunner[33]	Fiber Spool	~30m
Vinker[35]	Fiber Spool	230m
Duport[34]	Fiber Spool	1.7Km

overhead for two delay lines.

4.6.3. Speed And Power Comparison

In photonic RC architecture the slowest component is the delay line. Based on the delay line used in our architecture, the architecture is capable of operating at speed of up to 20Gb/s with a propagation loss as low as 2.2dB. The speed is similar to the architecture proposed in [83]. In our architecture, the power overhead is minimal due to control circuitry and additional MRR switch (~26mW). Hence based on power consumption of all other components: laser (10W), photodiode (5W), MZI (7W) and other components

(10W), our architecture can consume up to ~35W, which is less but similar to other architectures.

4.7. Summary

In this chapter, we propose a new on-chip architecture for parallel high-performance photonic RC. The architecture employs multiple electronically tunable delay lines with an electronically tuned MRR switch. It is 84 % more accurate for performing NARMA in standalone configuration and up to 80% more accurate while executing it in parallel with analog speech recognition task, as compared to [62]. The architecture is 46% more accurate compared to parallel processing photonic RC architecture employing TDM for inputs [83]. The area overhead is just 0.0184 mm^2 while power overhead is 26mW and operating speeds of 20Gb/s.

5. RECONFIGURABLE OPTOELECTRONIC HARDWARE ACCELERATOR FOR RESERVOIR COMPUTING[4]

Reservoir Computing (RC) is a subset of Recurrent Neural Networks (RNN) and has emerged as a powerful method for large scale classification and prediction of temporal problems with a reduced training time. Silicon-Photonics architectures have enabled high speed hardware implementations of Reservoir Computing (RC). With a Delayed Feedback Reservoir (DFR) model, only one non-linear node can be used to perform RC. Our proposed architecture in Chapter 2 has shown promising results for multi-layer or deep RC. However, in the subsequent proposed architectures we observed that as the architecture is modified and a task is assigned different resources, the performance of RC for that task changes. This motivates us to do an analysis of performance of multilayer RC system with varying parameters for different tasks. Our hypothesis is that the performance of RC architectures will saturate for a given task at different points. This hypothesis can lead to the need for a reconfigurable RC architecture.

5.1. Motivation

The multi-layer architecture is very promising and has enabled deep RC. However, the hardware architectures that have been proposed so far have all focused on implementation and improving performance. The philosophy followed in the design process has mainly been that bigger and deeper networks can perform better. While

[4]adapted with permission from Hasnain, Syed Ali, and Rabi Mahapatra. "Towards reconfigurable optoelectronic hardware accelerator for reservoir computing." Optoelectronic Devices and Integration IX. Vol. 11547. International Society for Optics and Photonics, 2020.

generally true, each task that needs to be performed can have different demands. The architectures have not been designed keeping this in mind. Hence, the lack the optimization of performance versus power motivates us for a new kind of architecture.

Therefore, in this chapter we study the multi-layer RC architecture in detail, while keeping performance and power in mind. We analyze through various experiments that high performance can be achieved by carefully optimizing and configuring the network parameters. Motivated by this we also propose a reconfigurable photonic RC architecture that can be configured to optimally perform classification and prediction problems. This has chapter has partially been reprinted with permission from [88]

5.2. Contributions

The contribution of this chapter are as follows:

1. We review the RC computing principles and explain the multi-layer photonic Reservoir computing paradigm.
2. We study the multi-layer photonic RC architecture in detail for performance vs different configurations.
3. We propose a new reconfigurable architecture for photonic RC and study it's performance.
4. Through experiments with NARMA task and analog speech recognition task show that by optimally configuring an up-to 4-layer architecture, power savings up to 40% can be achieved compared to state-of-the-art architectures while gaining up to 80% more accuracy. This is achieved at an on-chip area overhead of $0.0184mm^2$ for a single delay line and MRR switch.

5.3. Review Of Multi-Layer Reservoir Computing Architecture

Before discussing the performance of multi-layer architecture, we first review the basics of multi-layer photonic RC architecture for the ease of reader. For an in-depth review, the reader can refer to Chapter 2.

A multilayer RC is realized by simply including multiple reservoir layers (NL + delay) in between output of optical couplers and input of output layer as shown in Figure 5-1. Each reservoir layer stores multiple reservoir states. As shown, the output from 1st layer enters the NL Node of 2nd layer as input and so on. Each reservoir state from the last layer (Mth layer) is fed to the readout layer for training. The weights and bias value are trained using a linear regression technique in an offline computer to determine the final output.

The entire architecture can be divided into three parts: input layer, reservoir layer, and readout layer. These layers work in a pipeline fashion to process an input. The details are as follows.

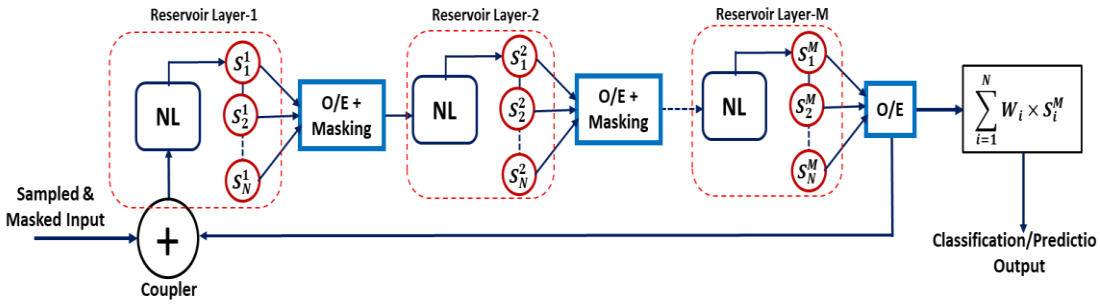


Figure 5-1 Review Of The Multi-Layer Photonic RC Architecture

5.3.1. Input Layer

The input electronic signal, x(t) is sampled with a period of T_S using a 'sample & hold' circuit. This in turn converts each continuous-time task x(t) to a discretized piecewise constant function p(n) where $p(t) = p(n)$, $nT_S \leq t < (n+1)\,T_S$, n is a time step. Each discrete input $p(n)$ is multiplied with a periodic mask input $m(n)$ of period T_S. Here $m(n) = m_i(n)$ for $(i-1)\,(\frac{T_S}{N}) < n \leq (i+1)\,(\frac{T_S}{N})$; i=1, 2,…N; $m_i(n)$ is randomly chosen from [−1, +1]. The result of this multiplication is a masked input $p_i(n)$ which drives the 1st reservoir layer in the 'reservoir computing segment'.

5.3.2. Reservoir Layer

Reservoir computing segment is the heart of multi-layer architecture. It comprises of 'M' reservoir layer as shown in Figure 5-1. Each reservoir layer consists of LiNbO3 Mach Zehnder Interferometer (MZI) as NL node and an optical fiber spool to provide delay. We consider sinusoidal nonlinearity in our design. The laser source provides optical carrier to the MZI of each reservoir layer. The output from the coupler is fed to the MZI of first reservoir layer through an electronic amplifier. For a timestep 'n', masked input $p_i(n)$ from the coupler is fed to the MZI of first reservoir layer. The MZI converts $p_i(n)$ into a reservoir state S_i^1 where 1 stands for 1st reservoir layer and i =1, 2…N (as we consider *N* reservoir states). The optical fiber spool provides a delay of T_S which is same as the sample time of 'sample & hold' circuit. For each time step n, the photodiode in the reservoir layer converts each reservoir state S_i^1 from optical form to electronic form. The

photodiode of each reservoir layer has an operating period of h second. Analytically, we can write $N = \frac{T_S}{h}$. The state of reservoir i in the first layer can be written as,

$$S_i^1(n) = Sin(\alpha S_i^1(n-1) + \beta m_i(n)x(n) + \emptyset) \tag{5.1}$$

The electronic output from the photodiode of one reservoir layer becomes an input to the next reservoir layer. This way, at any timestep n, i^{th} reservoir state of j^{th} reservoir layer can be written as,

$$S_i^j(n) = Sin(\alpha S_i^j(n-1) + \beta m_i(n)x(n) + \emptyset) \tag{5.2}$$

Here $\propto$ and β are feedback gains; $\emptyset$ is a bias value; and $m_i(n)$ represents the mask input. $\propto$, β, and $\emptyset$ are adjustable parameters. The MZI used in this particular design has sinusoidal non-linearity; hence the above equation is based on a *sin* function. One cycle of the architecture is defined as the time taken by a masked input $p_i(n)$ to travel from the 1st reservoir layer until the readout layer. This architecture has been referred to as the MReC architecture[62].

5.3.3. Readout Layer

Using the photodiode from the last reservoir layer (here the M^{th} layer), all the N reservoir states $S_i^M(n)$ are fed to an offline computer. The predicted output is determined using the following equation.

$$O(n) = \sum_{i=1}^{N} W_i S_i^M(n) + W_{bias} \tag{5.3}$$

Here Wi is calculated using linear regression training by comparing $O(n)$ with target output $O'(n)$.

5.4. Performance Of Multi-Layer Photonic RC Architectures

We study the performance of different configurations of multi-layer RC to understand the need for Reconfigurable RC architecture. To study this we use the following methodology.

To Evaluate the architecture, we used synthesized components from the MReC Architecture[14], except for the delay line. Table 5-1 illustrates details of components used in our design. We modified the Oger toolbox[86] to incorporate multiple reservoir layers and different number of nodes. An electronically tunable delay line of length 2cm and delay 660*ps* was modeled in the architecture. The proposed architecture is evaluated for four layers of Reservoir. We ran experiments using two benchmarks popular in the area of Reservoir Computing: NARMA task and Speech Recognition.

Table 5-1 Parametric Details Of Opto-Electronic Components

COMPONENTS	PARAMETERS	VALUES
LASER	Wavelength	1550nm
	Power	10W
MZI	Power	5W
PHOTODIODE	Power	5W
	Rise Time	15ps

5.4.1. Performance Of Multi-Layer RC With Different Configurations

It is interesting to study how increasing the number of nodes in reservoir layer effects the performance of the system. Hence, we try different configurations of the architecture i.e. different number of layers and different sizes of each reservoir layer. For the purpose of this analysis, we decided to vary the number of layers from 1 to 4, while varying the number of nodes from 50 to 300 with a step of 50. As shown in Fig 5-2, for the NARMA task, the performance of the system improved significantly as we moved to higher number of nodes in a single layer configuration. The performance also improved as we moved from single layer to double layer configuration. For a two layer and three-layer system we noticed that while the error rate decreased as we increased the number of

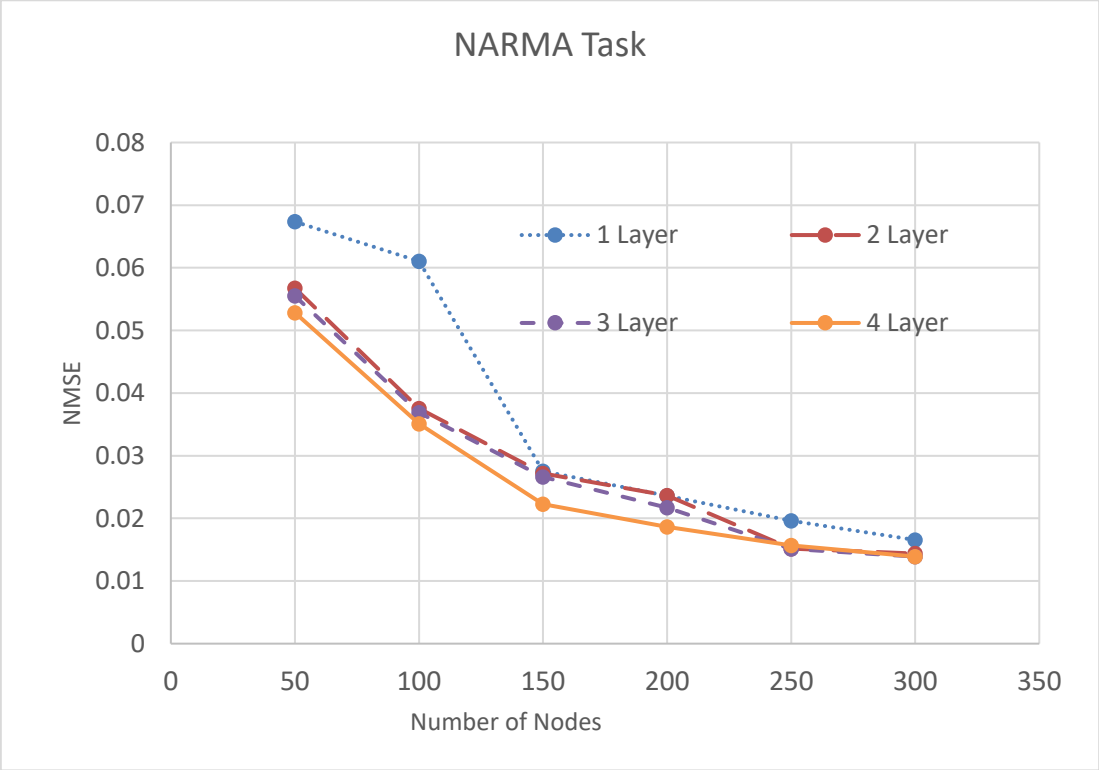


Figure 5-2 Effect Of Number Of Nodes And Reservoir Layers On NARMA Task

nodes, there is no significant gain as we move from two to three layers. The performance of the three-layer and four-layer configuration matched that of the two-layer configuration. Hence, for this task in particular, we observed that an optimal point does exist over which the performance more or less saturates.

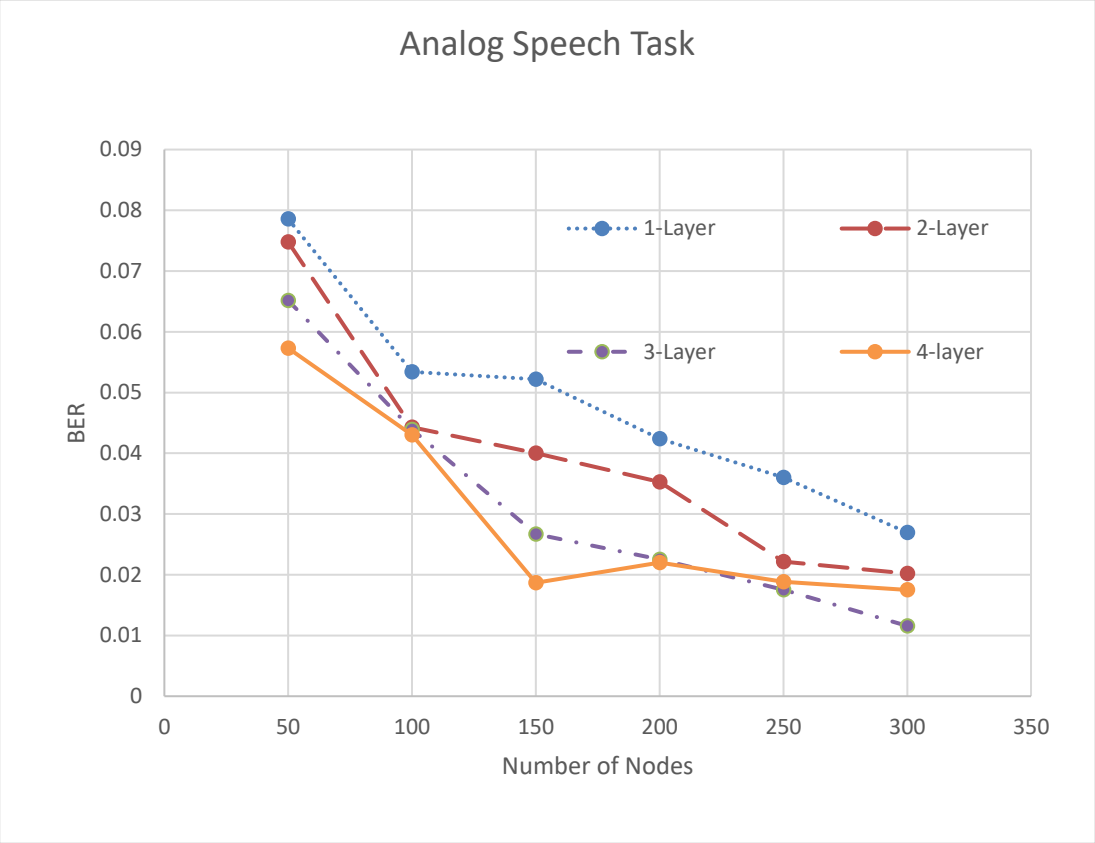


Figure 5-3 Effect Of Number Of Nodes And Reservoir Layers On Analog Speech Recognition Task

For the Analog Speech Recognition task, as shown in Fig 5-3, increasing the number of layers and nodes resulted in significant improvements. In our tests, we observed that in almost all cases, increasing the number of nodes and then layers helped improve

the performance of the system. Note that adding number of nodes in reservoirs vs increasing the number of reservoir layers is not the same. For instance, for a two-layer configuration, setting N=250 still performs worse than a three-layer configuration with N=150. In terms of total number of nodes, the former has 500 nodes whereas the later has 450 nodes. We found that performance of 4-layer configuration matched that of 3-layer when considering 200 and more nodes. Hence, we can say that for this task, the performance measure saturates at a point.

In a separate test, Fig 5-4, a system with 3 reservoir layers with N=300, performed the speech recognition task to give a BER= 0.01158. A single layer configuration with

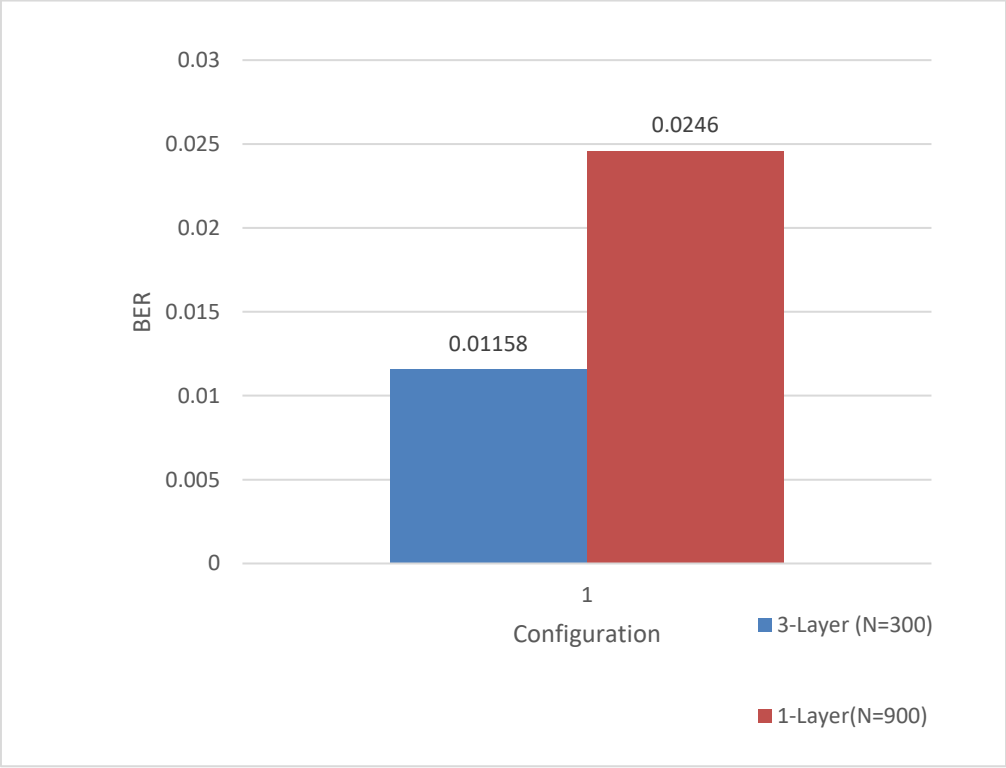


Figure 5-4 Performance Comparison Between Two Systems Configurations With Increasing Nodes And Increasing Layers

N=900 performed the same task to give a BER=0.0246. Hence, while increasing the number of nodes in a reservoir layer helps improve the performance, we notice that the gain in performance is not due to increased nodes but rather how they are connected. Dividing nodes in reservoir layers, and connecting them, may result in better performance than having more nodes in a single layer configuration. For the task at hand, different configurations may perform differently. This may encourage the research community to look into reconfigurable architectures in photonic RCs.

5.4.2. Power Consumption Of Multi-Layer RC

The power consumption of the architecture is independent of the number of nodes in the reservoir layer. This is because the number of nodes can be determined based on size of delay line and sampling time of the input. The power consumption depends on the

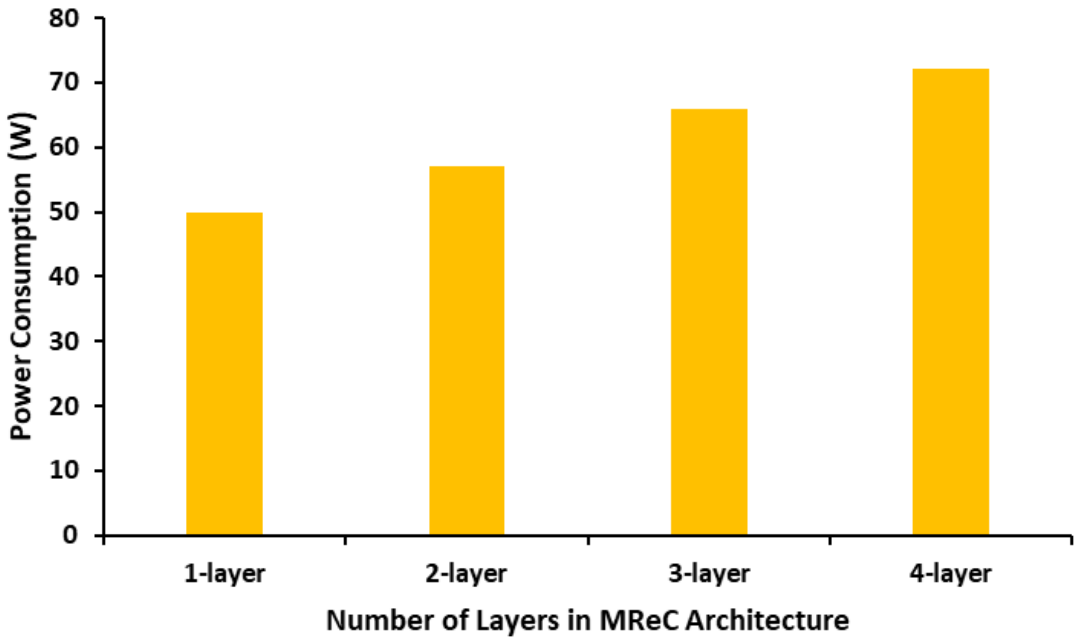


Figure 5-5 Power Consumption In Watt For 1-Layer, 2-Layer, 3-Layer, And 4-Layer MReC Architecture.

number of layers being used. MReC architecture has reported power of 50W for single layer to 72W for 4 layers, as shown in Fig. 5-5.

5.5. Reconfigurable Architecture For Photonic Reservoir Computing

From the study in section 5.4 we notice that performance of a RC architecture that has multiple reservoir layers saturates for each task at some point. We also observe that similar performance can be achieved by lowering number of layers used and increasing the virtual nodes in the reservoir. Motivated by this we propose that a reconfigurable RC hardware accelerator architecture is required that can be configured based on task at hand and is best suited to solve classification and predication problems. The proposed architecture, uses layer gating to reconfigure the number of layers being used and adjustment to sampling time to increase or decrease number of nodes in reservoir. While selecting number of nodes can be done through electronic control block, layer gating requires hardware modifications to the multi-layer architecture as this would require re-routing the laser light in the network.

A micro-ring resonator can be used to route laser light. Micro-ring Resonators (MRR) have been extensively discussed in literature[40, 39]. MRR uses resonances concept to couple light in waveguides to select a path for light. By means of an electric field the refractive index of a waveguide and hence the resonance frequency can be changed enabling selection of multiple paths. They are essential for the success of silicon photonics. Using resonance, they can enable the control of photonic path. In literature several MRR designs have been proposed[41, 42]. The main focus of these designs is speed, application and size. Recently a 2x2 MRR was proposed which can be employed

as reconfigurable DEMUX/MUX[43]. Such a component is ideally suited for selection of a delay line in a multi delay line-based architecture. In our proposed architecture we use a switching element that is made up of micro-ring resonator. Fig 5-6. Shows the switching element. The switching element is electronically tunable. Switching Element characteristics are:

1. Power: 0.5mW when on
2. Insertion loss: 1.5dB
3. Speed: >10Gb/s

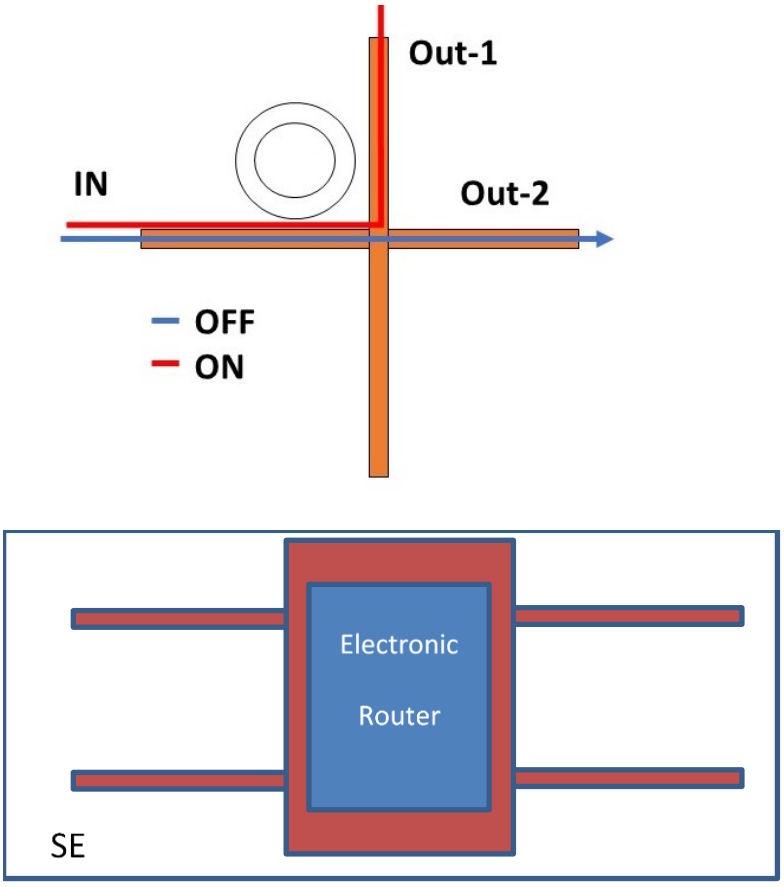


Figure 5-6 (Top) 2x2 MRR Based Switch (Bottom) Block Diagram For An Electronically Controlled Switching Element With MRR Switch Being Controlled By Electronic Router.

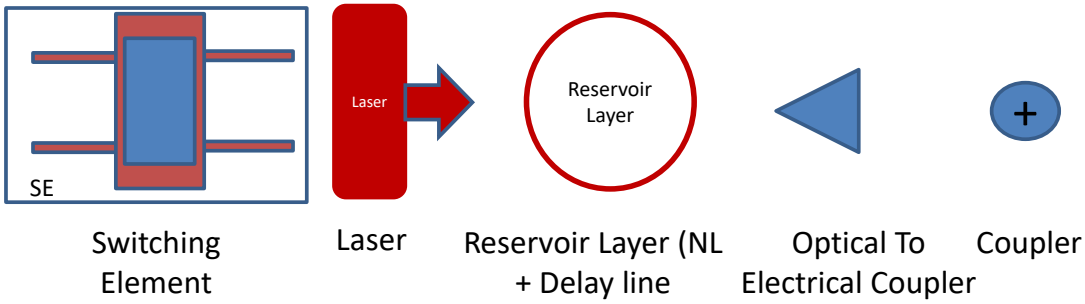


Figure 5-7 Symbols Used In The Reconfigurable Photonic RC Architecture

By introducing the switching element (SE), we propose an approach towards reconfigurable photonic RC. We divide the RC architecture into three vertical levels. The first is a layer of reservoirs whereas the second consists of all the elements required for coupling, masking and O/E conversions. In the third level we have all the SEs. The SEs can be controlled using an electronic block to select a layer and create a route for the data through the network. If a layer is selected, it is employed in the RC architecture. If its s not selected, we call that layer as gated layer and our system bypasses it. The power can be cut off to the elements of gated layer.

Fig. 5-7 shows the symbols used to define our architecture, whereas Fig. 5-8 presents the proposed reconfigurable architecture for RC. Our proposed architecture is similar to a circuit switch network approach. First a path is created from input to the output. In case, only one layer is required, the first layer is used. More layers can be added by selecting that layer by turning the SEs on.

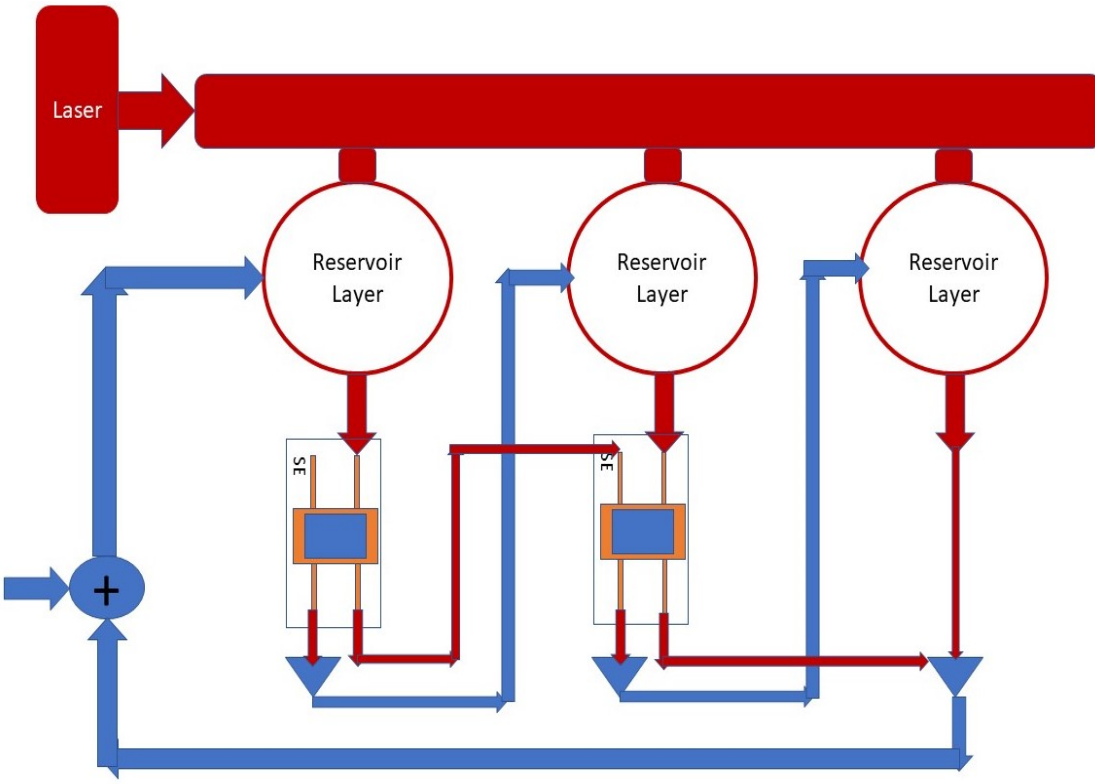


Figure 5-8 Propose Reconfigurable Architecture For Photonic Reservoir Computing

5.6. Results

We perform the NARMA and analog speech recognition task, based on optimized configurations of the proposed architecture and compare them to NARMA task. Table 5-2 shows the performance vs power consumption of MReC vs the proposed architecture. The power consumption is calculated by using the parameters in Table 5-1. The SE has a power consumption of just 0.5mW. We introduce a photodiode and a MZI per layer where as there is only one laser source in the architecture.

Table 5-2 Performance Vs Power Of MReC Vs Reconfigurable RC

Task	Architecture			
	MReC (4-layer, N=50)		Reconfigurable Photonic RC	
	Power	Accuracy	Power	Accuracy
NARMA	72W	NMSE=0.052 ± 0.0045	30W with 2 layers	NMSE=0.02, N=250
Analog Speech Recognition @ 0.02 BER	72W	BER=0.05	40W with 3 layers	BER=0.02, N=150

Our experiments with NARMA task and analog speech recognition task show that by optimally configuring an up-to 4-layer architecture, power savings up to 40% can be achieved compared to state-of-the-art architectures while gaining up to 80% more accuracy.

We also study the performance versus power consumption directly. Fig 5-9. Shows the performance of the system as we increase number of layers to gain more performance and as a result consume more power. We notice that although performance improves as more power is consumed but as we increase the number of nodes in the reservoir of the

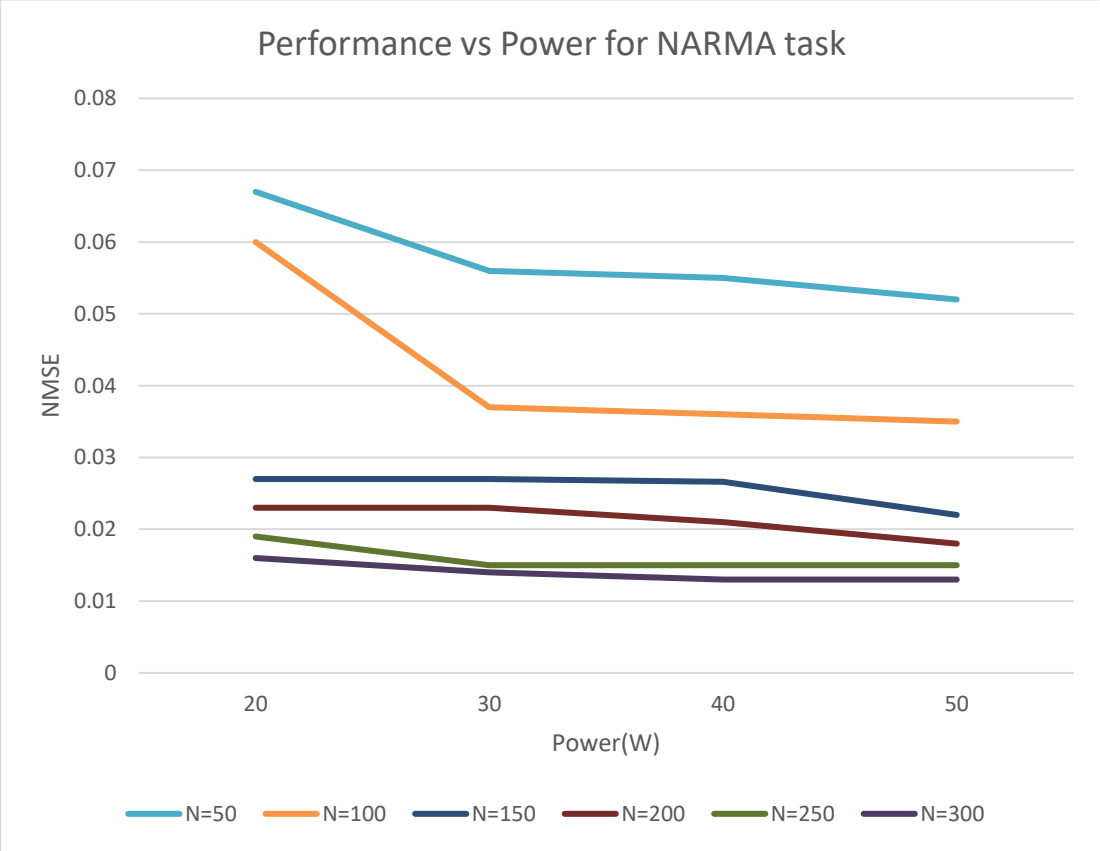


Figure 5-9 Performance Of The Architecture For NARMA Task Vs Power Consumed

architecture, the performance increase gained by increasing number of layers is very less. In fact, at N=250 and N=300 the performance matches after 30W. Hence one can argue that there is an optimal point of the system to operate which in this case would be N=250 and 2 layers.

5.7. Summary

In this chapter, we propose a new reconfigurable optoelectronic architecture for multi-layer RC motivated by studying a multilayer RC architecture in detail. Our proposed architecture, is based on DFR model implemented by the use of Mach Zehnder Modulator

(MZM) [89, 90, 91] and on chip low loss delay lines for improved performance. The hardware implementation is inspired by MReC architecture; however, it integrates photonic switches based on Micro Ring Resonators (MRR) to enable reconfigurability[92, 93, 94, 95]. The architecture enables layer selection and layer gating to select the number of layers required for a task. Selection of number of layers can optimize the architecture for a specific application, resulting in huge power savings, while maintaining the overall accuracy. Our experiments with NARMA task and analog speech recognition task show that by optimally configuring an up-to 4-layer architecture, power savings up to 40% can be achieved compared to state-of-the-art architectures while gaining up to 80% more accuracy. Our scalable architecture has an on-chip area overhead of $0.0184mm^2$ for a single delay line and MRR switch.

6. CONCLUSION AND FUTURE DIRECTIONS

6.1. Conclusion

The dissertation focuses on design of high performance, scalable and energy efficient photonic hardware architectures. It begins with the study of single node photonic computing model, which is inspired by the delayed feedback reservoir model. Based on this study, a multi-layer photonic hardware accelerator for multi-layer reservoir computing is proposed. The dissertation further investigates and designs new architectures for photonic RC that are capable of parallel processing and bring the whole architecture on-chip. Lastly, the dissertation investigates the limitations of current photonic architectures for RC to operate at a power vs performance optimal point and designs a reconfigurable architecture for photonic reservoir computing.

In Chapter 2, we demonstrate MReC, novel multilayer photonic RC architecture for large-scale classification and prediction tasks. Each layer of the proposed architecture comprises of an MZI based nonlinearity and fiber optic delay line to emulate reservoir computing. We synthesize the proposed multilayer design using a standard photonic CAD tool called IPKISS [38] and execute three well-known classification benchmarks and one widely used prediction benchmark to demonstrate: (1) up to 26.8% reduction in prediction error rate when single-layer MReC is compared with state-of-the-art single-layer photonic RC architecture [8]; (2) up to 50% reduction in prediction error rate when 4-layer MReC is compared with state-of-the-art design and (4) up to 34.21% improvement in power

consumption compared to best reported result [9]. These improvements in MReC come at a cost of 12% area overhead.

In Chapter 3, we show that Reservoir Computing systems have a reservoir layer that is made up of neurons with random connections and random weights that do not need training. The random weights and connections are independent of the task being performed using RC. The reservoir can therefore perform several tasks in parallel. In this paper we proposed a new architecture for photonic Reservoir Computing that uses Time Division Multiplexing (TDM) in its input layer to exploit the opportunity to perform multiple jobs in parallel using the same RC system. Our proposed system uses multiple reservoirs made up of MZI and low loss delay lines. Through simulations on NARMA and speech recognition, we show that our architecture can outperform some of the leading single layer architectures by up to 90% for NARMA task while performing analog speech recognition in parallel and closely matches the performance of leading multi-layer photonic RC architectures with an increased error of 8% due to parallel processing. The proposed high-speed architecture has a power consumption of ~50W for a 4-layer network.

In Chapter 4, we propose a new on-chip architecture for parallel high-performance photonic RC. The architecture employs multiple electronically tunable delay lines with an electronically tuned MRR switch. It is 84 % more accurate for performing NARMA in standalone configuration and up to 80% more accurate while executing it in parallel with analog speech recognition task, as compared to [8]. The architecture is 46% more accurate compared to parallel processing photonic RC architecture employing TDM for inputs [21].

The area overhead is just 0.0184 mm^2 while power overhead is 26mW and operating speeds of 20Gb/s.

Lastly, in Chapter 5, we propose a new reconfigurable optoelectronic architecture for multi-layer RC motivated by studying a multilayer RC architecture in detail. Our proposed architecture, is based on DFR model implemented by the use of Mach Zehnder Modulator (MZM) and on chip low loss delay lines for improved performance. The hardware implementation is inspired by MReC architecture; however, it integrates photonic switches based on Micro Ring Resonators (MRR) to enable reconfigurability. The architecture enables layer selection and layer gating to select the number of layers required for a task. Selection of number of layers can optimize the architecture for a specific application, resulting in huge power savings, while maintaining the overall accuracy. Our experiments with NARMA task and analog speech recognition task show that by optimally configuring an up-to 4-layer architecture, power savings up to 40% can be achieved compared to state-of-the-art architectures while gaining up to 80% more accuracy. Our scalable architecture has an on-chip area overhead of 0.0184mm^2 for a single delay line and MRR switch.

6.2. Future Directions

Photonic Deep Reservoir Computing has emerged as a promising candidate. The architectures that have been proposed in this dissertation contribute mainly towards the implementation of input layer and reservoir layer. This means that once the data has been processed, it needs to be taken to an offline computer to train the classifier stage. Hence, the architectures proposed here are in hybrid in nature as they use photonic reservoir layers

but offline traditional computers for training purposes. The training of even a single layer with many neurons is computationally expensive. Therefore, in future, investigations can be carried out in order to have a fully integrated photonic end to end reservoir computing system. Such a system will not only have a on chip readout layer but also photonic memory integrated to store the trained weights. In literature, several preliminary studies have been made in this regards. For example, [96] proposes an online training method that if implemented in hardware can have performance equal to digital output layer. Similarly, on chip photonic memories have been proposed [97, 98]. These memories have the capability to be designed into the hardware accelerators. The complete on chip operation would result in a lot of performance gains and power savings.

In future, to further optimize the power consumption, new designs can be investigated that use less number of non linear nodes per layer. As we noticed in Chapter 5, most of the power consumption is a result of dynamic operations of the system. Each layer in the architecture has power consuming components. By moving towards a design, where a non-linear computational component can be used for more than one layers, we can further optimize the power consumptions while gaining performance. This can be achieved by separating the linear and non-linear parts of computation. Firstly, all linear computation can be performed and the non-linearity can be introduced just before the readout layer. If such an architecture is viable, it would result in huge power savings.

Another direction to investigate would be hybrid deep learning photonic architectures that combine RC, RNN and CNN to perform a job. Since one architecture may not be suitable for al tasks, a hybrid architecture can allow to extract temporal and

non-temporal features and by means of RC reduced training time. In such an architecture, RC can be one of the feature extractors where as classifiers can be more specific architectures for the job being performed.

Lastly, while architectures for photonic RC can be proposed, their usefulness should also be demonstrated through real world applications. In this direction, system leveraging RC for human activity recognition can be designed [99]. RC architectures can also be demonstrated as light weight time series predictors in robotics [100, 101], data interpreters in IoTs [102] and medical applications like fall detection or prevention, seizure prediction and detection[103, 104]. While preliminary studies have shown that RC is well suited for these applications, the demonstration of these applications on low power, highly efficient photonic RC architectures can result in hand held and even wearable devices for these applications.

REFERENCES

[1] J. A. Freeman and D. M. Skapura, *Neural networks: algorithms, applications, and programming techniques*. Addison Wesley Longman Publishing Co., Inc., 1991.

[2] J. L. McClelland, D. E. Rumelhart, P. R. Group *et al.*, "Parallel distributed processing," *Explorations in the Microstructure of Cognition*, vol. 2, pp. 216–271, 1986.

[3] J. M. Zurada, *Introduction to artificial neural systems*. West St. Paul, 1992, vol. 8.

[4] K. Mehrotra, C. K. Mohan, and S. Ranka, *Elements of artificial neural networks*. MIT Press, 1997.

[5] G. Dreyfus, *Neural networks: methodology and applications*. Springer Science & Business Media, 2005.

[6] A. I. Galushkin, *Neural networks theory*. Springer Science & Business Media, 2007.

[7] G. Montavon, G. Orr, and K.-R. Müller, *Neural networks: tricks of the trade*. Springer, 2012, vol. 7700.

[8] D. J. MacKay and D. J. Mac Kay, *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[9] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[10] B. Schrauwen, 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, pp. 471–482.

[11] F. Takens, “Detecting strange attractors in turbulence,” in *Dynamical systems and turbulence, Warwick 1980*. Springer, 1981, pp. 366–381.

[12] F. Rosenblatt, “Principles of neurodynamics: Perceptions and the theory of brain mechanisms,” Spartan 1962.

[13] H. Jaeger, “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*, vol. 148, no. 34, p. 13, 2001.

[14] W. Maass, T. Natschlager, and H. Markram, “Real-time computing without stable states: A new framework for neural computation based on perturbations,” *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[15] D. Verstraeten, B. Schrauwen, M. dâ€™Haene, and D. Stroobandt, “An experimental unification of reservoir computing methods,” *Neural Networks*, vol. 20, no. 3, pp. 391–403, 2007.

[16] H. Jaeger, “Echo state network,” *Scholarpedia*, vol. 2, no. 9, p. 2330, 2007.

[17] E. Najibi and H. Rostami, “Scesn, spesn, swesn: Three recurrent neural echo state networks with clustered reservoirs for prediction of nonlinear and chaotic time series,” *Applied Intelligence*, vol. 43, no. 2, pp. 460–472, 2015.

[18] A. Goudarzi, A. Shabani, and D. Stefanovic, “Product reservoir computing: Time-series computation with multiplicative neurons,” in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[19] D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. Van Campenhout, "Isolated word recognition with the liquid state machine: a case study," *Information Processing Letters*, vol. 95, no. 6, pp. 521–528, 2005.

[20] H. Jaeger, M. Lukoševicius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Networks*, vol. 20, no. 3, pp. 335–352, 2007.

[21] R. J. MacGregor, "Neural and brain modeling" Academic Press San Diego*, CA*, 1987.

[22] D. V. Buonomano and M. M. Merzenich, "Temporal information transformed into a spatial code by a neural network with realistic properties," *Science*, vol. 267, no. 5200, pp. 1028–1030, 1995.

[23] T. Yamazaki and S. Tanaka, "The cerebellum as a liquid state machine," *Neural Networks*, vol. 20, no. 3, pp. 290–297, 2007.

[24] M. Rabinovich, R. Huerta, and G. Laurent, "Transient dynamics for neural processing," *Science*, pp. 48–50, 2008.

[25] D. Nikolic, S. Häusler, W. Singer, and W. Maass, "Distributed fading memory for stimulus properties in the primary visual cortex," *PLoS Biology*, vol. 7, no. 12, p. e1000260, 2009.

[26] D. Verstraeten, "Een studie van de liquid state machine: een woordherkenner," Master's thesis*, Ghent University, ELIS Department*, 2004.

[27] T. Keith, "A general-purpose gpu reservoir computer," Master's thesis, University of Canterbury. Department of Electrical & Computer Engineering 2013.

[28] B. Schrauwen, M. Dâ€™Haene, D. Verstraeten, and J. Van Campenhout, “Compact hardware liquid state machines on fpga for real-time speech recognition,” *Neural Networks*, vol. 21, no. 2-3, pp. 511–523, 2008.

[29] A. Rodan and P. Tino, “Minimum complexity echo state network,” *IEEE Transactions on Neural Networks*, vol. 22, no. 1, pp. 131–144, 2010.

[30] J. Li, K. Bai, L. Liu, and Y. Yi, “A deep learning based approach for analog hardware implementation of delayed feedback reservoir computing system,” in *Proceedings of 19th International Symposium In Quality Electronic Design (ISQED)*, 2018.

[31] J. Li, C. Zhao, K. Hamedani, and Y. Yi, “Analog hardware implementation of spike-based delayed feedback reservoir computing system,” in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 3439–3446.

[32] K. Bai, Q. An, and Y. Yi, “Deep-dfr: A memristive deep delayed feedback reservoir computing system with hybrid neural network topology,” in *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2019, pp. 1–6.

[33] D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, “Parallel photonic information processing at gigabyte per second data rates using transient states,” *Nature Communications*, vol. 4, p. 1364, 2013.

[34] F. Duport, A. Smerieri, A. Akrout, M. Haelterman, and S. Massar, “Fully analogue photonic reservoir computer,” *Scientific Reports*, vol. 6, p. 22381, 2016.

[35] Q. Vinckier, F. Duport, A. Smerieri, K. Vandoorne, P. Bienstman, M. Haelterman, and S. Massar, "High-performance photonic reservoir computer based on a coherently driven passive cavity," *Optica*, vol. 2, no. 5, pp. 438–446, 2015.

[36] C. Gallicchio and A. Micheli, "Echo state property of deep reservoir computing networks," *Cognitive Computation*, vol. 9, no. 3, pp. 337–350, 2017.

[37] C. Gallicchio, A. Micheli, and L. Pedrelli, "Deep reservoir computing: A critical experimental analysis," *Neurocomputing*, vol. 268, pp. 87–99, 2017.

[38] W. Bogaerts, M. Fiers, M. Sivilotti, and P. Dumon, "The ipkiss photonic design framework," in *Optical Fiber Communication Conference*. Optical Society of America, 2016, pp. W1E–1.

[39] J. Heebner, R. Grover, T. Ibrahim, and T. A. Ibrahim, "*Optical microresonators: theory, fabrication, and applications."* Springer Science & Business Media, 2008, vol. 138.

[40] W. Bogaerts, P. De Heyn, T. Van Vaerenbergh, K. De Vos, S. Kumar Selvaraja, T. Claes, P. Dumon, P. Bienstman, D. Van Thourhout, and R. Baets, "Silicon microring resonators," *Laser & Photonics Reviews*, vol. 6, no. 1, pp. 47–73, 2012.

[41] Q. Xu, S. Manipatruni, B. Schmidt, J. Shakya, and M. Lipson, "12.5 gbit/s carrier-injection-based silicon micro-ring silicon modulators," *Optics Express*, vol. 15, no. 2, pp. 430–436, 2007.

[42] Q. Xu, B. Schmidt, J. Shakya, and M. Lipson, "Cascaded silicon micro-ring modulators for wdm optical interconnection," *Optics Express*, vol. 14, no. 20, pp. 9431–9436, 2006.

[43] P. Sethi and S. Roy, “All-optical ultrafast switching in 2× 2 silicon microring resonators and its application to reconfigurable demux/mux and reversible logic gates,” *Journal of Lightwave Technology*, vol. 32, no. 12, pp. 2173–2180, 2014.

[44] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, “How to construct deep recurrent neural networks,” *arXiv preprint arXiv:1312.6026*, 2013.

[45] Y. Guan, Z. Yuan, G. Sun, and J. Cong, “Fpga-based accelerator for long short-term memory recurrent neural networks,” in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2017, pp. 629–634.

[46] D. Neil, M. Pfeiffer, and S.-C. Liu, “Phased lstm: Accelerating recurrent network training for long or event-based sequences,” in *Advances in Neural Information Processing Systems*, 2016, pp. 3882–3890.

[47] M. S. Kulkarni and C. Teuscher, “Memristor-based reservoir computing,” in *2012 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*. IEEE, 2012, pp. 226–232.

[48] C. E. Merkel, Q. Saleh, C. Donahue, and D. Kudithipudi, “Memristive reservoir computing architecture for epileptic seizure detection.” in *BICA*, 2014, pp. 249–254.

[49] C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, and W. D. Lu, “Reservoir computing using dynamic memristors for temporal information processing,” *Nature Communications*, vol. 8, no. 1, p. 2204, 2017.

[50] S. Ortn and L. Pesquera, “Reservoir computing with an ensemble of time-delay reservoirs,” *Cognitive Computation*, vol. 9, no. 3, pp. 327–336, 2017.

[51] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*. ACM, 2014, pp. 675–678.

[52] C. Sun, C.-H. O. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L.-S. Peh, and V. Stojanovic, "Dsent-a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling," in *Networks on Chip (NoCS), 2012 Sixth IEEE/ACM International Symposium on*. IEEE, 2012, pp. 201–210.

[53] T. Instruments-Developed, "46-word speaker-dependent isolated word corpus (ti46)," *NIST Speech Disc*, pp. 7–1, 1991.

[54] C. Chatfield and A. S. Weigend, "Time series prediction: Forecasting the future and understanding the past: Neil a. gershenfeld and andreas s. weigend, 1994,the future of time series', in: As weigend and na gershenfeld, eds.,(addison-wesley, reading, ma), 1-70." *International Journal of Forecasting*, vol. 10, no. 1, pp. 161–163, 1994.

[55] T. Kubota, K. Nakajima, and H. Takahashi, "Dynamical anatomy of narma10 benchmark task," *arXiv preprint arXiv:1906.04608*, 2019.

[56] H. J. Caulfield and S. Dolev, "Why future supercomputing requires optics," *Nature Photonics*, vol. 4, no. 5, p. 261, 2010.

[57] F. Ponulak and A. Kasinski, "Introduction to spiking neural networks: Information processing, learning and applications." *Acta Neurobiologiae Experimentalis*, vol. 71, no. 4, pp. 409–433, 2011.

[58] R. M. Nguimdo, G. Verschaffelt, J. Danckaert, and G. Van der Sande, "Simultaneous computation of two independent tasks using reservoir computing based on a single photonic nonlinear node with optical feedback," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3301–3307, 2015.

[59] H. Safaai, M. von Heimendahl, J. M. Sorando, M. E. Diamond, and M. Maravall, "Coordinated population activity underlying texture discrimination in rat barrel cortex," *Journal of Neuroscience*, vol. 33, no. 13, pp. 5843–5855, 2013.

[60] K. Bai 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)*, vol. 14, no. 4, pp. 1–22, 2018.

[61] M. C. Soriano, S. Ortn, L. Keuninckx, L. Appeltant, J. Danckaert, L. Pesquera, and G. Van der Sande, "Delay-based reservoir computing: noise effects in a combined analog and digital implementation," *İEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 2, pp. 388–393, 2014.

[62] D. Dhang, S. A. Hasnain, and R. Mahapatra, "MReC: A multilayer photonic reservoir computing architecture," in *Proceedings of 20th International Symposium In Quality Electronic Design (ISQED)*, 2019.

[63] R. S. Tucker, G. Eisenstein, and S. K. Korotky, "Optical time-division multiplexing for very high bit-rate transmission," *Journal of Lightwave Technology*, vol. 6, no. 11, pp. 1737–1749, 1988.

[64] D. M. Spirit, A. D. Ellis, and P. E. Barnsley, “Optical time division multiplexing: Systems and networks,” *IEEE Communications Magazine*, vol. 32, no. 12, pp. 56–62, 1994.

[65] S. A. Hasnain, D. Dang, and R. Mahapatra, “Multilayer photonic reservoir computing architecture using time division multiplexing for parallel computation,” in *Optoelectronic Devices and Integration IX*, vol. 11547. International Society for Optics and Photonics, 2020, p. 115471S.

[66] Y. Ji, Y. Chung, D. Sprinzak, M. Heiblum, D. Mahalu, and H. Shtrikman, “An electronic mach–zehnder interferometer,” *Nature*, vol. 422, no. 6930, pp. 415–418, 2003.

[67] Y. Kuriki, J. Nakayama, K. Takano, and A. Uchida, “Impact of input mask signals on delay-based photonic reservoir computing with semiconductor lasers,” *Optics Express*, vol. 26, no. 5, pp. 5777–5788, 2018.

[68] L. Appeltant, G. Van der Sande, J. Danckaert, and I. Fischer, “Constructing optimized binary masks for reservoir computing with delay systems,” *Scientific Reports*, vol. 4, p. 3629, 2014.

[69] J. Nakayama, K. Kanno, and A. Uchida, “Laser dynamical reservoir computing with consistency: an approach of a chaos mask signal,” *Optics Express*, vol. 24, no. 8, pp. 8679–8692, 2016.

[70] S. Khan, M. A. Baghban, and S. Fathpour, “Electronically tunable silicon photonic delay lines,” *Optics Express*, vol. 19, no. 12, pp. 11780–11785, 2011.

[71] Z. Jackson, “Free spoken digit dataset (fsdd),” Technical Report, 2016.

[72] K. Hamedani, L. Liu, R. Atat, J. Wu, and Y. Yi, "Reservoir computing meets smart grids: Attack detection using delayed feedback networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 734–743, 2017.

[73] F. Xia, L. Sekaric, and Y. Vlasov, "Ultracompact optical buffers on a silicon chip," *Nature Photonics*, vol. 1, no. 1, p. 65, 2007.

[74] F. Morichetti, A. Melloni, C. Ferrari, and M. Martinelli, "Error-free continuously-tunable delay at 10 gbit/s in a reconfigurable on-chip delay-line," *Optics Express*, vol. 16, no. 12, pp. 8395–8405, 2008.

[75] A. Melloni, A. Canciamilla, C. Ferrari, F. Morichetti, L. O'Faolain, T. Krauss, R. De La Rue, A. Samarelli, and M. Sorel, "Tunable delay lines in silicon photonics: coupled resonators and photonic crystals, a comparison," *IEEE Photonics Journal*, vol. 2, no. 2, pp. 181–194, 2010.

[76] A. Yariv, Y. Xu, R. K. Lee, and A. Scherer, "Coupled-resonator optical waveguide a proposal and analysis," *Optics Letters*, vol. 24, no. 11, pp. 711–713, 1999.

[77] J. Adachi, N. Ishikura, H. Sasaki, and T. Baba, "Wide range tuning of slow light pulse in soi photonic crystal coupled waveguide via folded chirping," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 16, no. 1, pp. 192–199, 2009.

[78] S. Khan and S. Fathpour, "Complementary apodized grating waveguides for tunable optical delay lines," *Optics Express*, vol. 20, no. 18, pp. 19859–19867, 2012.

[79] Y. Gao, H. Cansizoglu, K. G. Polat, S. Ghandiparsi, A. Kaya, H. H. Mamtaz, A. S. Mayet, Y. Wang, X. Zhang, T. Yamada *et al.*, "Photon-trapping microstructures enable

high-speed high-efficiency silicon photodiodes," *Nature Photonics*, vol. 11, no. 5, p. 301, 2017.

[80] R. Horvath, J. Roux, J. Coutaz, J. Poette, B. Cabon, and C. Graham, "Ultrafast ingaas photoswitch for rf signal processing," in *2017 International Conference on Optical Network Design and Modeling (ONDM)*. IEEE, 2017, pp. 1–5.

[81] X. Gan, R.-J. Shiue, Y. Gao, I. Meric, T. F. Heinz, K. Shepard, J. Hone, S. Assefa, and D. Englund, "Chip-integrated ultrafast graphene photodetector with high responsivity," *Nature Photonics*, vol. 7, no. 11, p. 883, 2013.

[82] Y. Salamin, P. Ma, B. Baeuerle, A. Emboras, Y. Fedoryshyn, W. Heni, B. Cheng, A. Josten, and J. Leuthold, "100 ghz plasmonic photodetector," *ACS Photonics*, vol. 5, no. 8, pp. 3291–3297, 2018.

[83] D. Dang and R. Mahapatra, "A time-shared photonic reservoir computer for big data analytics," *arXiv preprint arXiv:1703.08211*, 2017.

[84] S. A. Hasnain and R. Mahapatra, "On-chip parallel photonic reservoir computing using multiple delay lines," in *2020 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. IEEE, 2020.

[85] M. Doi, N. Hashimoto, T. Hasegawa, T. Tanaka, and K. Tanaka, "40 gb/s low-drive-voltage linbo3 optical modulator for dqpsk modulation format," in *Optical Fiber Communication Conference*. Optical Society of America, 2007, p. OWH4.

[86] D. Verstraeten, B. Schrauwen, S. Dieleman, P. Brakel, P. Buteneers, and D. Pecevski, "Oger: modular learning architectures for large-scale sequential processing," *Journal of Machine Learning Research*, vol. 13, no. Oct, pp. 2995–2998, 2012.

[87] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[88] S. A. Hasnain and R. Mahapatra, "Towards reconfigurable optoelectronic hardware accelerator for reservoir computing," in *Proc. of SPIE Vol*, vol. 11547, 2020, pp. 115470W–1.

[89] L. Liao, D. Samara-Rubio, M. Morse, A. Liu, D. Hodge, D. Rubin, U. D. Keil, and T. Franck, "High speed silicon mach-zehnder modulator," *Optics Express*, vol. 13, no. 8, pp. 3129–3135, 2005.

[90] W. M. Green, M. J. Rooks, L. Sekaric, and Y. A. Vlasov, "Ultra-compact, low rf power, 10 gb/s silicon mach-zehnder modulator," *Optics Express*, vol. 15, no. 25, pp. 17106–17113, 2007.

[91] M. R. Watts, W. A. Zortman, D. C. Trotter, R. W. Young, and A. L. Lentine, "Low-voltage, compact, depletion-mode, silicon mach–zehnder modulator," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 16, no. 1, pp. 159–164, 2010.

[92] T. J. Todman, G. A. Constantinides, S. J. Wilton, O. Mencer, W. Luk, and P. Y. Cheung, "Reconfigurable computing: architectures and design methods," *IEE Proceedings-Computers and Digital Techniques*, vol. 152, no. 2, pp. 193–207, 2005.

[93] K. Bondalapati and V. K. Prasanna, "Reconfigurable computing systems," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1201–1217, 2002.

[94] S. Hauck and A. DeHon, *Reconfigurable computing: the theory and practice of FPGA-based computation*. Elsevier, 2010.

[95] A. DeHon and J. Wawrzynek, “Reconfigurable computing: what, why, and implications for design automation,” in *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*, 1999, pp. 610–615.

[96] P. Antonik, M. Haelterman, and S. Massar, “Online training for high-performance analogue readout layers in photonic reservoir computers,” *Cognitive Computation*, vol. 9, no. 3, pp. 297–306, 2017.

[97] Z. Cheng, C. Rós, N. Youngblood, C. D. Wright, W. H. Pernice, and H. Bhaskaran, “On-chip phase-change photonic memory and computing,” in *Active Photonic Platforms IX*, vol. 10345. International Society for Optics and Photonics, 2017, p. 1034519.

[98] S. Raoux, W. Wenic, and D. Ielmini, “Phase change materials and their application to nonvolatile memories,” *Chemical Reviews*, vol. 110, no. 1, pp. 240–267, 2010.

[99] F. Palumbo, C. Gallicchio, R. Pucci, and A. Micheli, “Human activity recognition using multisensor data fusion based on reservoir computing,” *Journal of Ambient Intelligence and Smart Environments*, vol. 8, no. 2, pp. 87–107, 2016.

[100] E. A. Antonelo and B. Schrauwen, “Supervised learning of internal models for autonomous goal-oriented robot navigation using reservoir computing,” in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2959–2964.

[101] T. Li, K. Nakajima, M. Cianchetti, C. Laschi, and R. Pfeifer, “Behavior switching using reservoir computing for a soft robotic arm,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 4918–4924.

[102] T. Yamane, H. Numata, J. B. Héroux, N. Kanazawa, S. Takeda, G. Tanaka, R. Nakane, A. Hirose, and D. Nakano, "Dimensionality reduction by reservoir computing and its application to iot edge computing," in *International Conference on Neural Information Processing*. Springer, 2018, pp. 635–643.

[103] P. Buteneers, D. Verstraeten, B. Van Nieuwenhuyse, D. Stroobandt, R. Raedt, K. Vonck, P. Boon, and B. Schrauwen, "Real-time detection of epileptic seizures in animal models using reservoir computing," *Epilepsy Research*, vol. 103, no. 2-3, pp. 124–134, 2013.

[104] M. A. Escalona-Morán, M. C. Soriano, I. Fischer, and C. R. Mirasso, "Electrocardiogram classification using reservoir computing with logistic regression," *IEEE Journal of Biomedical and health Informatics*, vol. 19, no. 3, pp. 892–898, 2014.