University of California
Santa Barbara

Multi-wavelength Selective Crossbar Switch

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Akhilesh Sanjay Prabhu Khope

Committee in charge:

Professor John E Bowers, Committee Co-Chair
Dr. Adel A M Saleh, Adjunct Professor, Committee Co-Chair
Professor Rod C Alferness
Professor Clint Schow
Dr. Roger Helkey, Associate Director, IEE

September 2019

The Dissertation of Akhilesh Sanjay Prabhu Khope is approved.

Dr. Roger Helkey, Associate Director, IEE

Professor Rod C Alferness

Professor Clint Schow

Dr. Adel A M Saleh, Adjunct Professor, Committee Co-Chair

Professor John E Bowers, Committee Co-Chair

August 2019

Multi-wavelength Selective Crossbar Switch

by

Akhilesh Sanjay Prabhu Khope

Acknowledgements

When I started out at UC Santa Barbara as a PhD student in 2015, there was no funding until the AIM project proposal got accepted. Without funding no work gets done, so first I am grateful to Prof. Rod Alferness for funding me for the first year and Prof. John Bowers for funding me for the rest of my PhD. To John, thanks for giving me complete freedom at work without which I could not have explored so many different areas in research. I enjoyed our conversation in the monthly meetings with Rod, John and Adel.

For the first few months of my PhD I spent a lot of time with Adel doing a literature survey, it was an exciting journey in a new field. To Adel, thanks for filtering out the unsound ideas and helping me frame my thoughts. I enjoyed our white board discussions.

To Roger, Clint, Takako and Andy at the Datacom meetings for discussions and helpful feedback. To Tin, for the fun we had at the New York Training conference. To Alex, Warren, Eric and Andy for the excellent office environment. To Alex, for inspiring me to get Scuba certified and also for tips on cooking Thai cuisine. To Aranya, for being my climbing buddy. To Sean, Simran Lawrence, Wilson and many others, for the fun time we had at UCSB badminton intramural. To Chirag, for providing helpful advice when needed.

Thanks to Nicolas, Geza, Christos, Sarat, Aditya for helpful discussions. I would also like to thank Minh, Tony, Paolo, Yuan and Alan for their helpful advice. Special Thanks to Songtao for help with high speed measurements.

Thanks to my Pappa, Mamma and Sairaj for the support during my PhD and for recharge time in India.

Finally I would also thank Jeremiah Hebding, Brett Attaway, and SUNY for their foundry services.

Curriculum Vitæ

Akhilesh Sanjay Prabhu Khope

Education

2019	Ph.D. in Electrical and Computer Engineering, University of California, Santa Barbara.
2014	M.S. in Electrical and Computer Engineering, University of California, Santa Barbara.
2013	M.Tech. in Engineering Physics with specialization in Nanoscience, Indian Institute of Technology Bombay
2013	B.Tech.(Hons.) in Engineering Physics, Indian Institute of Technology Bombay

Awards and Honors

- UC Santa Barbara Graduate Division Fellowship (2014-18)
- IIT Bombay, Department Topper (2009)
- National Balashree Award for Creative Scientific Innovation (2006)
- National Talent Search Examination Scholarship (2006)
- Represented India in Asian Physics Olympiad, Almaty, Kazakhstan (2006)

Publications

1. "Multi-wavelength selective crossbar switch" Optics express 27.4 (2019): 5203-5216.
   Khope, Akhilesh SP, et al.

2. "On-chip wavelength locking for photonic switches." Optics letters 42, no. 23 (2017): 4934-4937.
   Khope, Akhilesh SP et al.

3. "Forward bias operation of silicon photonic Mach Zehnder modulators for RF applications." Optics Express 25, no. 19 (2017): 23181-23190.
   Chao, Rui Lin, Jin Wei Shi, Aditya Jain, Takako Hirokawa, **Akhilesh SP Khope**, Clint Schow, J. E. Bowers, Roger Helkey, and James F. Buckwalter.

4. "Elastic WDM crossbar switch for data centers." In IEEE Optical Interconnects Conference (OI), 2016, pp. 48-49. IEEE, 2016.
   Khope, Akhilesh SP, Adel AM Saleh, John E. Bowers, and Rod C. Alferness.

5. "Elastic WDM switching for scalable data center and HPC interconnect networks." In OptoElectronics and Communications Conference (OECC) held jointly with 2016 International Conference on Photonics in Switching (PS), 2016 21st, pp. 1-3. IEEE, 2016.
   Saleh, Adel AM, **Akhilesh SP Khope**, John E. Bowers, and Rod C. Alferness.

6. "Elastic WDM optoelectronic crossbar switch with on-chip wavelength control." In Photonics in Switching, pp. PTh1D-3. Optical Society of America, 2017.
   Khope, Akhilesh, Andy M. Netherton, Takako Hirokawa, Nicolas Volet, Eric Stanton, Clint Schow, Roger Helkey, Adel Saleh, John Bowers, and Rod C. Alferness.

Abstract

Multi-wavelength Selective Crossbar Switch

by

Akhilesh Sanjay Prabhu Khope

Switches are used to connect servers together to form a network in datacenters and in telecommunication. This can be done with electronic switches or optical switches. There has been a tremendous effort in the optics community to miniaturize switches with integrated photonic technology. Many of these switches are single wavelength switches, i.e they connect two ports with a single wavelength channel. These switches under utilize the channel bandwidth provided by Wavelength Division Multiplexing.

In this thesis, I propose a multi-wavelength selective switch that can dynamically assign multiple channels for any connection. Such a switch can help scale data centers and drastically reduce cabling complexity. The theory of the switch is developed and measurements of the switch fabricated in a Silicon photonics foundry are reported. A novel on-chip locking mechanism for the switches is also proposed and demonstrated.

We measured an average path loss of 10 dB on Gen III 8x4 switch with a standard deviation of 2 dB. These losses were reduced by a factor of 4 in Gen IV 4x4 switch with undoped ring resonators to 2.65 dB with a standard deviation of 1.35 dB. The tuning range of the rings is twice the Gen III 8x4 switch and the Gen IV 4x4 switch is tunable across 4 channels at 400 GHz spacing. We measured a 10 to 90 % large signal switching time of 15 μs. Multiple input Bit Error Rate measurements showed negligible power penalty due to incoherent crosstalk. Ring resonators are also locked to two channels spaced at 100 and 200 GHz using an on-chip photodetector in an on-chip locking experiment.

Contents

List of Figures

Chapter 1

Introduction

Switches are used to interconnect servers in data centers as well as a variety of terminal equipment in global fiber-optic telecommunication networks. If quantity N servers are connected to a switch, it can reduce fiber connections from N(N-1)/2 in an all-to-all interconnected network to only N and also intelligently provision bandwidth on demand. The type of switch discussed in this thesis is an optical crossbar which will be fully explained later. One of the earliest demonstration of a crossbar switch was in 1915. This switch was an electromechanical switch used for switching telephone calls. The first widespread use of crossbar switches was in Sweden from 1926 and later in the US by Bell Company's 1XB crossbar exchanges in 1938. A detailed discussion on different types of switching networks follows later in the chapter.

Smaller crossbar switches are connected to form bigger networks, because it is difficult to build a crossbar switch with thousands of ports where the number of switching elements scales as N^2 (N is the number of ports or radix of the switch). Topologies such as Clos [6] made it possible to connect a large number of smaller switches in different configuration to enable strictly non-blocking connections. A discussion of types of switching networks can be found in the following section.

Most networks, from data centers to global telecommunications networks, consists of fiber-optics transmission links and electronic or optical switches at the nodes. Most switches used in datacenters are electronic switches, that convert photons to electrons, then apply the switching logic and then convert electrons to photons for transmission. These switches are fast, with latency $<1\ \mu s$ and can consume a large amount of power as they have multi-Tbps capacities, with $>$ 25 Gbps per port. Optical fiber switches that switch the entire fiber bandwidth are also used to route in data centers persistent connections. These connections do not change frequently with time [2]. The highest radix switch of this type was manufactured using MicroElectro Mechanical Systems (MEMS) technology and has 512 ports [7]. Calient, a Santa Barbara company, also sold multiple 320 port MEMS switches commercially.

1.1 Switching Networks

Fig. 1.1 shows the states of a 2x2 switch. The switch changes between a BAR state and a CROSS state. This is a fundamental building block in many switching networks which connect N inputs to N outputs.

There are three main types of switching networks :

1. **Strictly non-blocking:** Existing connections need not be disturbed for new connections.

2. **Rearrangeable non-blocking:** There exists another connection pattern to accommodate the new connection.

3. **Blocking:** The connection request sometimes cannot be satisfied.

Fig. 1.2 shows 4x4 crossbar switch and 4x4 Benes network. A crossbar switch is made up of N^2 switching elements as compared to $NlogN - N/2$ switching elements in Benes.

The crossbar configuration is also non-blocking, as compared to a Benes network, which is a rearrangeable non-blocking switch

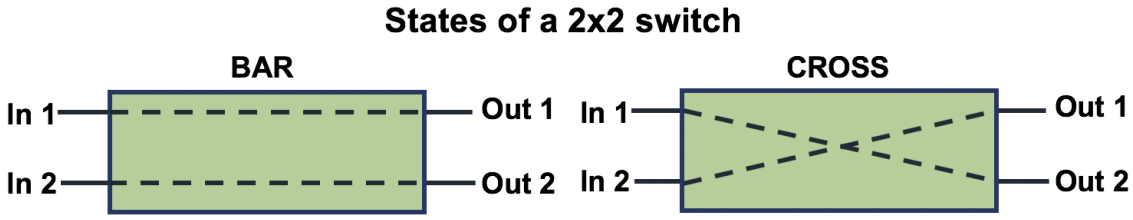


Figure 1.1: States of a 2x2 switch are shown. BAR state is shown on the left of figure where IN 1 connects to OUT 1 and IN 2 connects to OUT 2. CROSS state is shown on the right that connects IN 1 to OUT 2 and IN 2 to OUT 1.

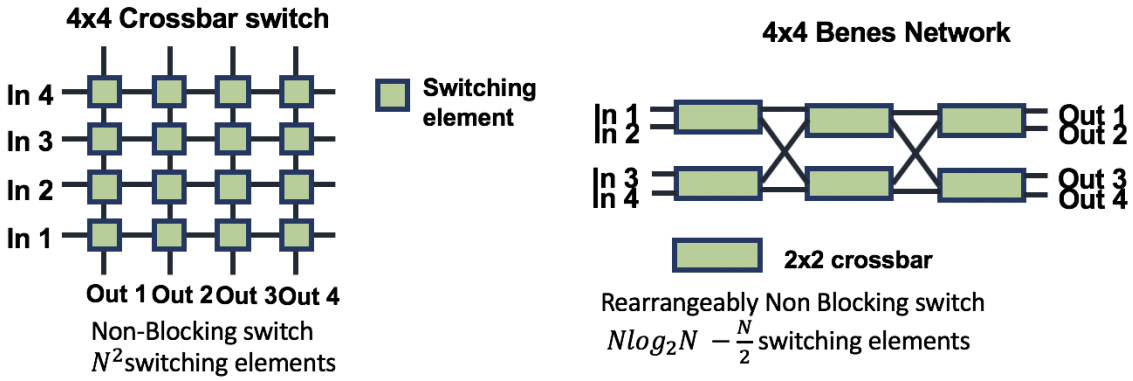


Figure 1.2: Two different switching networks are shown. A 4x4 crossbar switch is non-blocking and a 4x4 Benes network is rearrageable non-blocking.

Fig. 1.3 shows a leaf spine data center architecture. In this architecture, switching networks discussed earlier are connected together to form a distributed switching network that can interconnect tens to hundreds of thousands of servers residing in thousands of racks in a data center. Servers inside a rack are connected with copper wires, and higher bandwidth connections are made from the Top of the Rack (TOR) switch to the leaf layer using fiber-optic links. A single optical fiber can carry multiple channels, a scheme known as Wavelength Division Multiplexing (WDM). In the example shown in the figure,

one can see 4 wavelength (channels) separated by 400 GHz at 25 Gbps Non Return to Zero on-off key modulation scheme (NRZ).

1.2 Optical Switches

Optical interconnection networks used for intra- and inter-chip communications in future exascale systems can achieve improved scalability, i.e., higher throughput, lower latency, and increased energy efficiency, compared to their electrical counterparts [8]. Optical switches that leverage WDM can be used in datacenters to reduce fiber handling complexity of data center switches [1], and power consumption in the architectures [2] shown in Fig. 1.4.

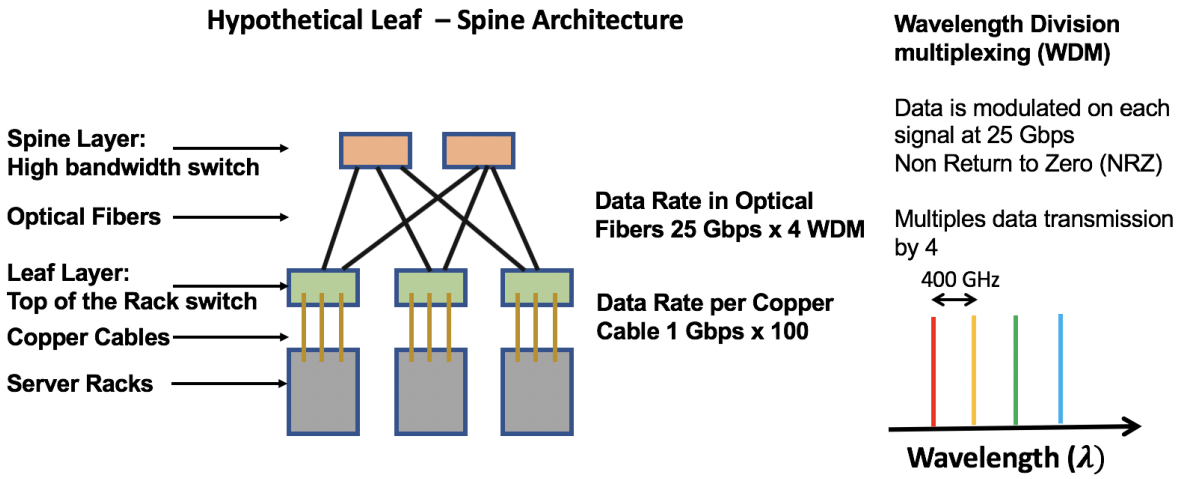


Figure 1.3: Hypothetical leaf - spine architecture shown with server racks connected by copper cables. Leaf layer and Spine layer connected by optical fibers using Wavelength Division Multiplexing (WDM). In WDM data is modulated on each wavelength (signal or channel) usually using Non Return to Zero (NRZ) format. Wavelength channels at 400 GHz separation are shown.

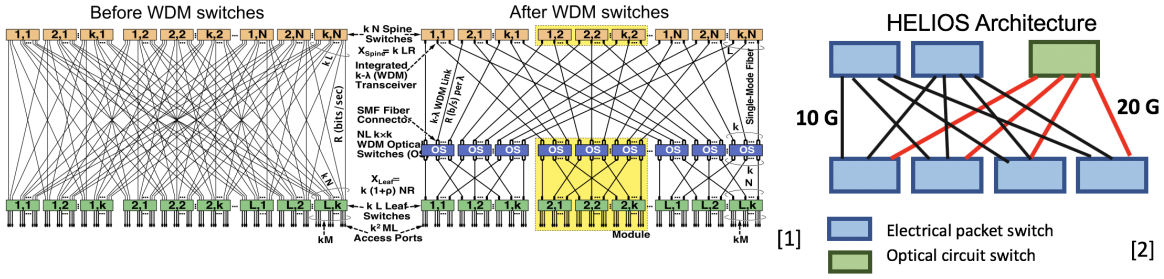


Figure 1.4: Optical switches in datacenters, examples. Left:Same datacenter size with less number of fibers and cable handling [1] © 2014 IEEE. Right: Power savings by using Hybrid architecture (HELIOS) [2], route persistent flows over Optical circuit switch for power savings.

Optical switches with N input, N output with M wavelengths/port can be subdivided into three categories:

1. **Single wavelength switches:** Single wavelength connectivity between ports

2. **Fat pipe or Broadband switches or Fiber switches:** All or no wavelengths connected. Fat pipe switches switch the entire Wavelength Division Multiplexed (WDM) signal between input/output (I/O) ports.

3. **Multi-wavelength switches:** Any subset of wavelengths can be routed

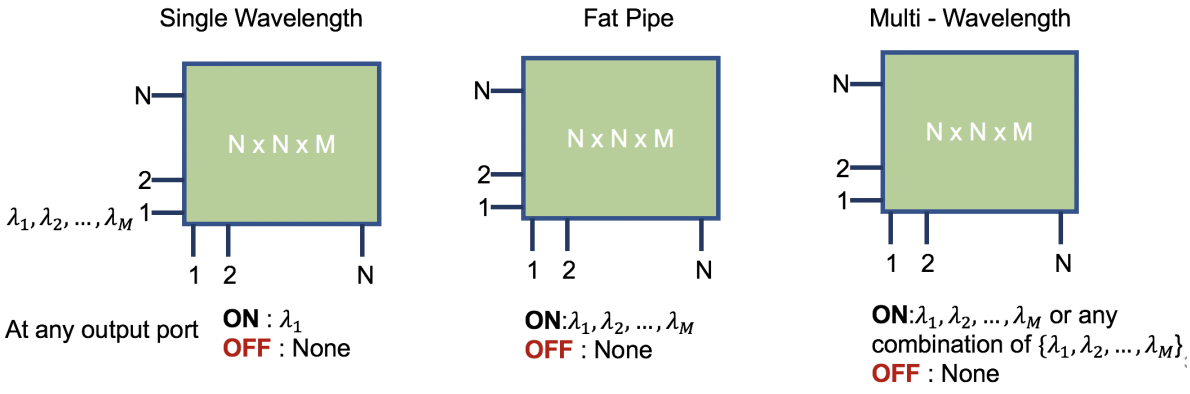


Figure 1.5: Three types of optical switches: single wavelength, fat pipe or fiber switch and (wavelength-selective) multi-wavelength switch

Reference	Port count	Type	Architecture	Loss (dB)	Crosstalk (dB)
[9]	5x5	Single Wavelength	Bidirectional optical router	8	> 16
[10]	8x4	Fiber switch	Crossbar	2-5	19.5-23.4
[11]	4x4	Single Wavelength	Benes	< 6.9	$<$-13.6
[12]	4x4	Single Wavelength	Switch and Select	$>$1.8	-51.4 to -31.6
[13]	4x4	Single Wavelength	Omega	4.4 to 8.4	-18.8 to -14.7
This work [5]	8x8	Multiwavelength	Multiwavelength Selective Crossbar	$>$5	$>$35/$>$40*
This work [4]	8x4	Multiwavelength	Multiwavelength Selective Crossbar	6 to 14	$>$20/$>$32*
This work Section. 3.5	4x4	Multiwavelength	Multiwavelength Selective Crossbar	0.8 to 5	22/32*

Table 1.1: Table comparing this work to previous work on Microring Resonator based integrated optical switches.
*Out of band rejection at 200 and 400 GHz

Single wavelength switches have been demonstrated with both Mach-Zehnder interferomerometers (MZ) and Micro Ring Resonators (MRR). A notable example of a single wavelength MZ switch is [14]. MRR based switches are tabulated in Table 1.1. Multiple variants of integrated fiber switches based on broadband MZ have been reported in a Silicon Photonic platform [14], Global Foundries 90-nm platform [15] and Silica Platform in [16]. A ring resonator-based fat pipe switch is reported in [17]. A 512 port MEMS switch is also demonstrated by [7]. MEMS switches from [7] and from Calient switch light in 3 dimensions with mirrors and light propagation in free space [18, 19]. A waveguide based 2 dimensional MEMS switch is also demonstrated by [20]. Integrated optical switches fall in the 2-dimensional switch category. Although higher port count MEMS switches have been commercialized, their footprint is much larger than integrated optical switches.

Other notable demonstrations of integrated optical switches are [21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35]. Wavelength-selective switches with single wavelength connectivity based on an Arrayed Waveguide Grating Router (AWGR) [36], cascaded microring crossbar switch [37], modular switch and select architecture [38], switch with multi-casting functionality [39] have been reported. A high-radix on-chip architecture is proposed in [40]. A major difference between AWGR [36] and our switch is that AWGR is a single wavelength switch within a Free Spectral Range (FSR) and routing is controlled through tunable lasers instead of on-chip heater based tuning of the MRRs.

Table 1.1 shows a comparison of various microring resonator-based switches. The switches demonstrated in this thesis are Multiwavelength switches. Multi-wavelength selective switches promise greater flexibility in connection patterns between I/O ports as compared to both single wavelength connectivity and fat pipe switches. We proposed a multi-wavelength selective switch initially in [3, 41].Our switches show comparable or better loss and crosstalk and at the same time also provide flexible allocation of bandwidth between connections due to their multiwavelength nature.

Multiple NxNxM switching approaches have been demonstrated [42, 43, 20], but they either require too many ring resonators for similar connectivity or have a higher insertion loss. The main difference in our approach as compared to the NxNxM crossconnect approach proposed in [42] is that the signal passes through multiple off-resonance ring resonators instead of through drop ports of multiple rings in the NxNxM approach [42]. We measured an off-resonance loss of 0.17 dB which is much smaller than the drop loss of 1.9 dB. Also the total number of rings is NxNx2, which is much smaller than NxNxM as reported in [42]. In [43], 4x4x4 switch was reported, which could be used as a building block in a bigger Benes [44] or Clos network[6]. The NxNxM Benes switch requires $M(Nlog_2(N) - N/2)$ rings implementation as compared to $2N^2$ in our case. This is the reason we chose a crossbar instead of a Benes configuration. An 8x8x8 MEMS wavelength selective switch is demonstrated in [20], which uses 8 single wavelength 8x8 switches in conjunction with eight 1x8 demultiplexers and eight 8x1 multiplexers. In our case, the multiplexing and demultiplexing functionality are implicitly performed in the microrings themselves.

1.3 Overview

In this thesis, we demonstrate thermally-tuned switching, which is compact and provides μs switching time, which matches datacenter requirements. While nanosecond switching is achievable through electro-optical tuning, this leads to larger switch footprint and higher optical losses. Moreover, the software algorithms needed to determine the required switch configurability in a data center can take a few microseconds, or more, to complete, which obviate the need for nanosecond switching. The work reported in this thesis also applies to nanosecond switching with electro-optic tuning.

Here, we demonstrate and analyze an optical interconnect switch architecture operat-

ing in the wavelength-time domain, and having interconnection capability ranging from static all-to-all wavelength connectivity (which is suitable for systems exchanging short messages) to on-demand microsecond-scale dynamically allocated multiple-wavelength connectivity (which is suitable for systems exchanging longer message sizes). For large enough number of wavelengths per port, the switch can simultaneously enable both connectivity modes, thus resulting in efficient, low-latency operations for systems with both short and long messages. This makes the switch suitable for a diverse set of applications ranging from interconnecting multiple processing nodes for HPC, to interconnecting individual servers or racks of servers in data centers.

The present crossbar switch configuration makes it possible to optimize the interconnection network dynamically (down to microsecond switching time), to respond to slow or fast variations in the workloads and the message-exchange rates in these systems. This switching time optimization requires a centralized control mechanism based on tracking the traffic demand in real time. This is not a unique requirement for our proposal; rather, earlier pioneering works on introducing slow [2] or fast [45] optical time-domain circuit switching in data centers also require centralized control to optimize the interconnection network. In fact, centralized control is currently in general use in many mega data centers [46].

Another important application of our switch in data centers is that it can enable the creation of flexible, scaled-out, low-latency, leaf-spine data center realization that does not require an increase in the overall size or the number of fiber ports of the associated electronic packet switches [1]. The advantage of this approach is that it reduces cabling complexity to scale datacenters to higher radix and also requires fewer electronic switches which results in a lower overall power consumption.

The main contribution of this work is as follows:

1. For an NxN crossbar switch with M wavelengths per input port, M rings per crosspoint are not required when buffers are present at transmitters. In most applications, near full connectivity can be obtained with just two rings per crosspoint. Diminishing reduction in latency is observed as the number of ring resonators are increased beyond 2 per crosspoint. A huge saving in the number of switching elements, tuning power consumption and path loss are also observed for the case of two ring resonators as compared to M ring resonators.

2. We demonstrate 8x4, 4x4, and 8x8 switches fabricated in a 220 nm Silicon Photonic platform and designed a driver PCB to characterize the switch.

3. Loss, resonant wavelength and power efficiency are measured for every ring resonator on a chip for multiple switches.

4. We measured impairments due to crosstalk sources by injecting 40 Gbps data through multiple input ports of the switch measuring the Bit Error Rate (BER) of the signal. Measurements for Pulse Amplitude Modulation - 4 Level (PAM4) at 60 Gbaud are also demonstrated.

5. A novel wavelength locking scheme for wavelength-selective optical switches is also demonstrated.

In Chapter 2, a thorough analysis of the switch is given. In this chapter, a discussion on architecture and system design, contention resolution, performance analysis is presented. This is followed by discussion on wavelength assignment and scalability. In this chapter we show that we do not require M ring ring resonators at each crosspoint, as 2 ring resonators are sufficient on the basis of latency and complexity.

In Chapter 3, experimental results of devices and switches are presented. In this chapter, multiple generations switches are presented with 2^{nd} order MRR with and without

doping for electro-optic tuning, and a switch with cascaded 1^{st} order resonators with half the number of pads is also reported. A discussion on previous switches and integration with electronics is also presented.

In Chapter 4, a novel locking scheme for multiwavelength switches is presented.

In Chapter 5, directions for future research are provided. This thesis also includes code in Appendix A and Appendix B. Code in Appendix A was used for latency simulations and code in Appendix B was used for automating measurements. Layouts are given in Appendix C.

Chapter 2

Switching Network Analysis

2.1 Introduction

In this chapter, a discussion on architecture and system design, contention resolution and performance analysis is presented. This is followed by discussion on wavelength assignment and scalability. We also show that we do not require M ring resonators at each crosspoint, as two resonators are sufficient on the basis of latency and complexity. The chapter also contains detailed discussion on latency simulations, scalability and wavelength assignment in the switching network.

2.2 Architecture and System Design

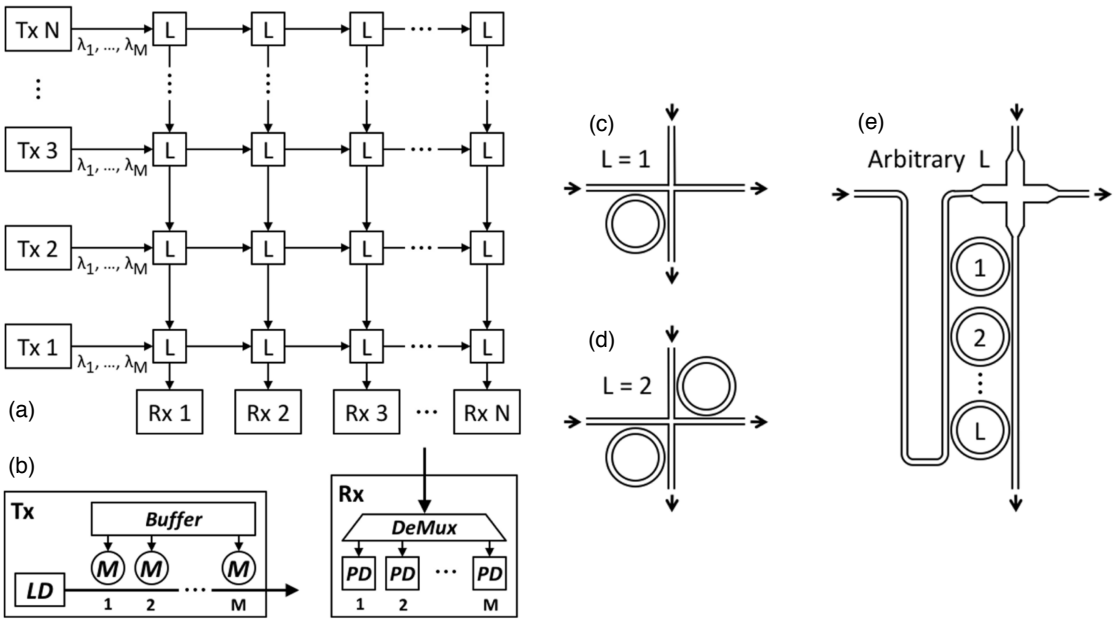


Figure 2.1: a) NxN crossbar interconnect switch. (b) Possible implementation of M-wavelength transmitter and receiver. Tx: Transmitter, Rx: Receiver, DeMux: Demultiplexer, PD: Photodetector, LD: Laser Diode Array, M: Microring-based modulator. Possible microring-based implementations of the switching blocks for (c) L=1, (d) L=2, (e) arbitrary L. The implementation in (e) includes a low-loss, low-cross-talk multimode crossing. Each microring is tuned to a different wavelength. Reprinted from [3]© The Optical Society.

In Fig. 2.1(a), we present an NxN crossbar switch with switching blocks at every intersection. The transmitter injects a WDM signal with M wavelengths in each input port. The 'L'blocks at the crossings in Fig. 2.1 (a) consist of wavelength-selective components that can switch up to L wavelengths from an input port to an output port. The wavelengths that are not switched pass through the block. For proper operation, we assume that the switching elements in any two L blocks will not be tuned to the same wavelength for any row or any column. Thus, wavelengths that have been switched by an L block go straight down to the receiver through the intermediate L blocks below the switching L

block. Microring-based implementations of an L block are shown in Fig. 2.1 (c),(d) and (e) for various values of L. This design allows an all-to-all connectivity with flexibility in bandwidth between input and output ports to accommodate different traffic patterns. Our switch can operate under these two extreme connectivity patterns: all input ports to a single output port with one wavelength per input port, and L wavelength connectivity between M/L input ports and any output port. An example of these two extreme connectivity patterns for L=2, N=M=3 is shown in Fig. 2.3. Example configurations for L=1 and L=3 are given in Fig 2.2 and Fig 2.4. The ability to have L wavelength connectivity results in reduced latency, reduced contention and higher throughput, as will be explained in Section 2 of this chapter, contention resolution.

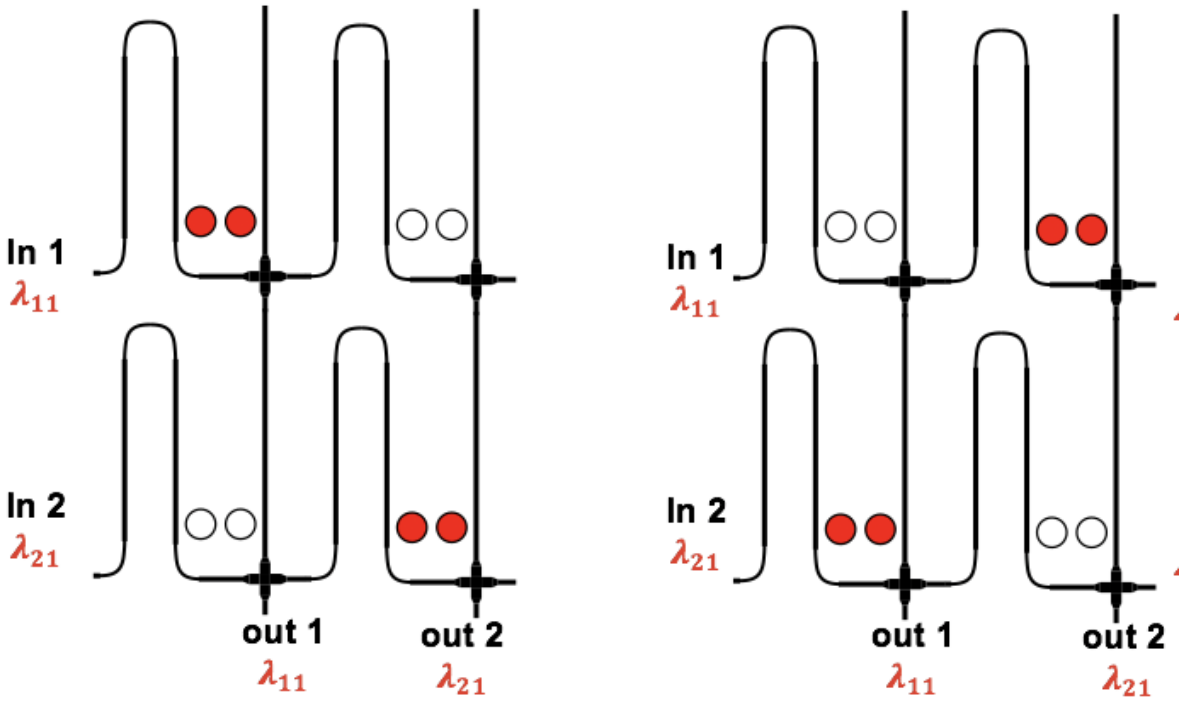


Figure 2.2: Example with two switch configurations for L=1, N=2, M=1. Wavelength channels are denoted by colors, and blank ring denotes a MRR tuned to off channel. Red colored rings are tuned to wavelength corresponding to channel red. In this case only two configurations are possible.

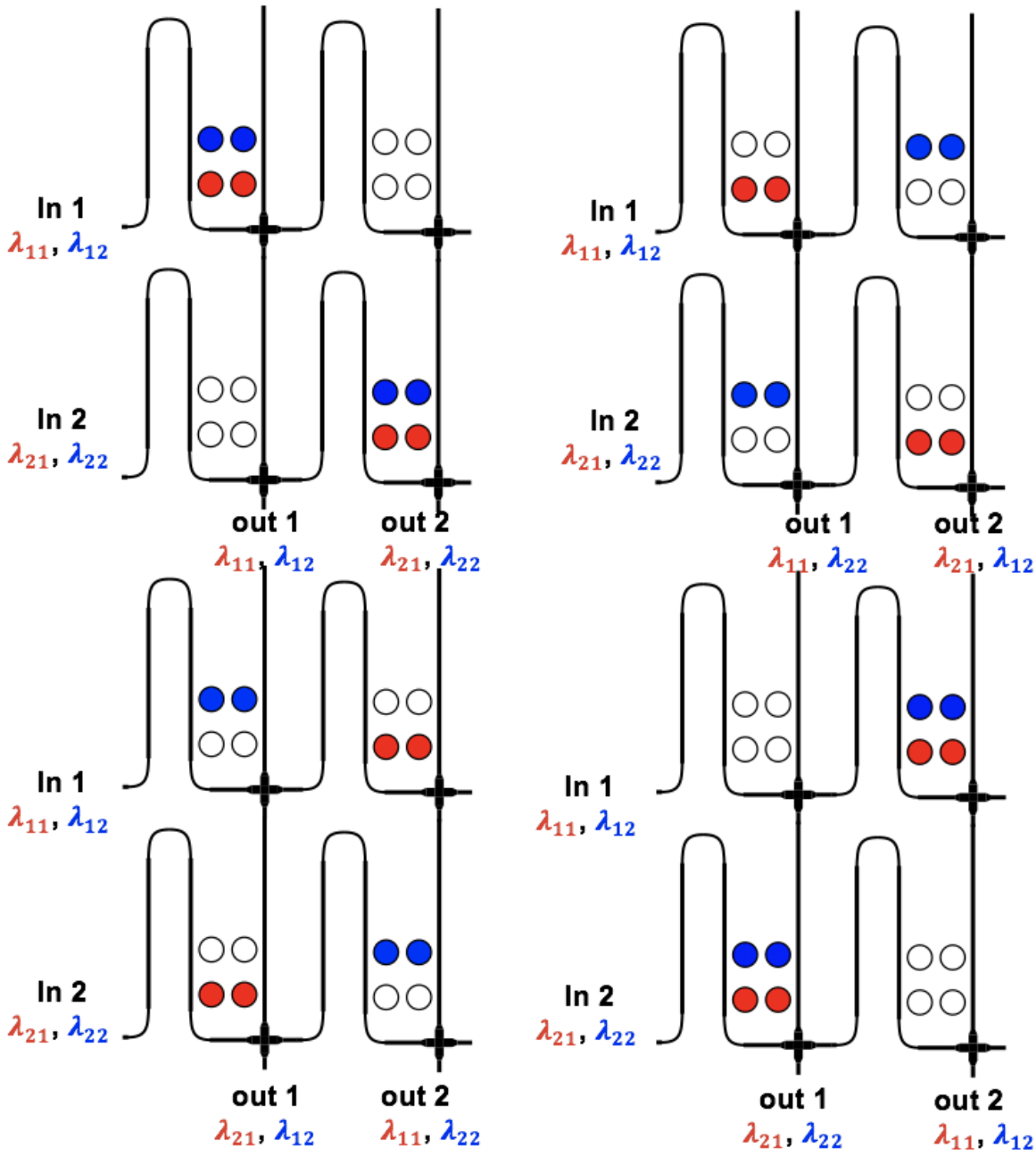


Figure 2.3: Example with four switch configurations for L=2, N=M=2. Wavelength channels are denoted by colors and blank ring denotes a MRR tuned to off channel. Configurations with two wavelengths at one crosspoint corresponds to a higher bandwidth connection whereas the rest corresponds to a configurations dynamically switched at run time to provide higher bandwidth for some links and provide all to all connectivity

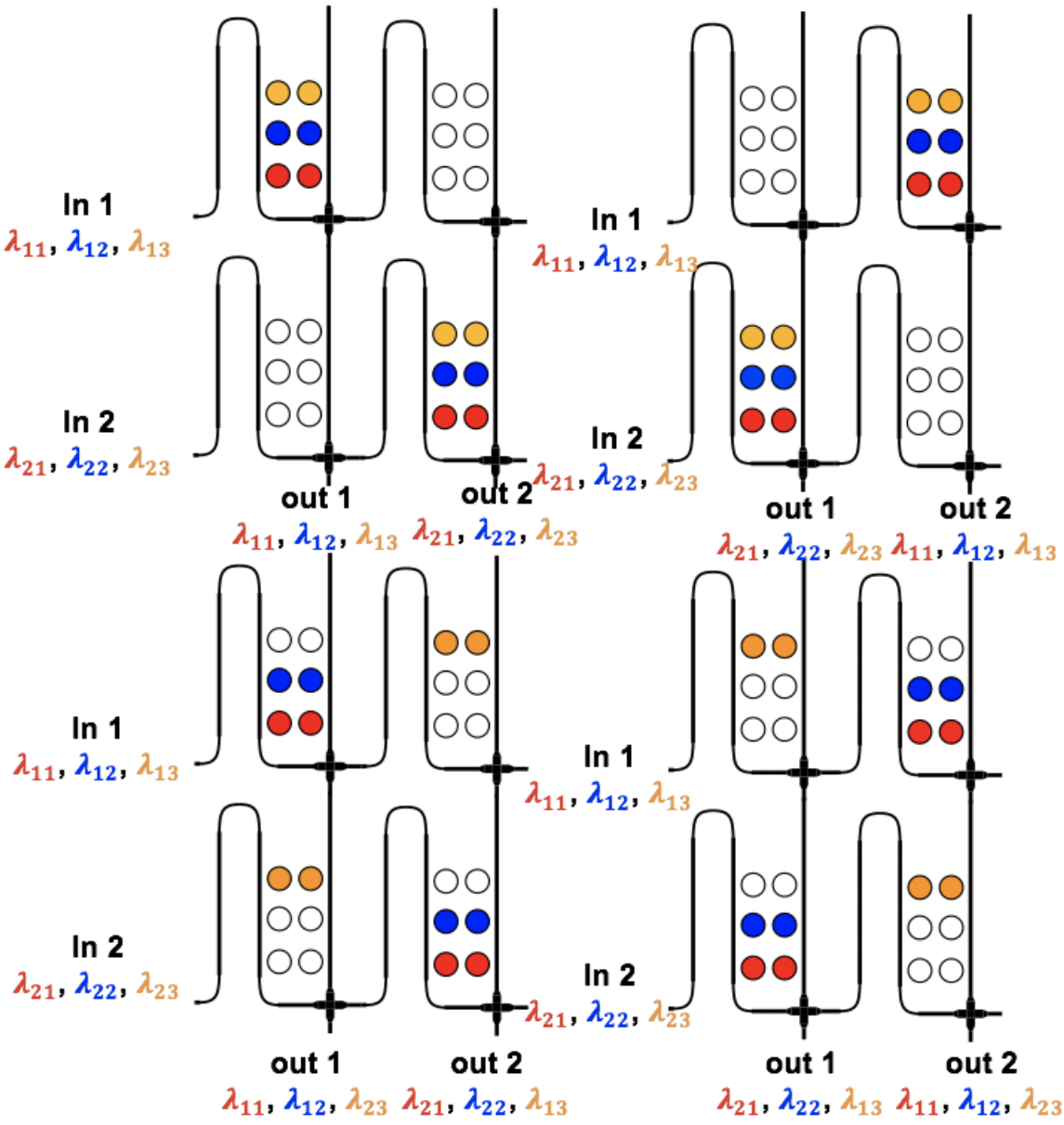


Figure 2.4: Some Examples of configurations for L=3, N=2, M=3. Wavelength channels are denoted by colors and blank ring denotes an MRR tuned to off channel.

We use centralized arbitration, which requires a single shared controller with a separate low-speed control network connecting each transmitter to the controller. A buffer is placed at each of the transmitter sites, as shown in Fig. 2.1 (b). The system is assumed to be synchronous, i.e., all transmitters share the same clock and time is divided into equal timeslots. Within each timeslot, the transmitters inform the controller with the number of requested connections and their respective destinations and gets ready for transmission at the start of the next timeslot. The controller then runs the wavelength assignment and contention resolution algorithm (described in the next subsection), assigns wavelengths to each request, informs the transmitters about the wavelength assignments of the various connections, and tunes the appropriate microrings placed in the various switching blocks. The requests that cannot be transmitted are placed in the buffer and transmitted when a wavelength becomes available in a subsequent timeslot.

If a connectivity configuration within any timeslot satisfies the following constraints:

1. Any input port can transmit a maximum of M wavelengths, and a maximum of M can be received at any output port. This is due to the placement of M modulators at each transmitter and M photodiodes at each receiver. (All the transmitters use the same M wavelengths.)

2. Maximum of L wavelengths need to be switched from any input port to any output port. All wavelengths at an output port must be distinct.

Then, the switch can be shown to be non-blocking. More precisely, the switch is rearrangeable non-blocking since it is reconfigured each time slot.

For any number of ports (N) and for one wavelength connectivity (L=1), no ring tuning is required and the system is similar to an arrayed waveguide grating router. Contention resolution is required in our switch as resources (wavelengths) are shared between multiple input ports, and the method is described in the next subsection.

2.3 Contention Resolution

In our system, connection requests which are assumed to follow a uniform random distribution, are generated within each timeslot for transmission in the subsequent timeslots. To distinguish between request arrival times within a timeslot, we divide the timeslot into k parts. The probability that a request is generated by each transmitter within each of these k parts is given by $p = \Lambda/k$, where Λ is the mean rate of request generation per timeslot for each transmitter. We choose a large enough k such that the request generation per timeslot follows a Poisson distribution with mean Λ.

We now define the connectivity matrix, and the buffer matrix. The connectivity matrix is the matrix of requests that satisfies the conditions for non-blocking mentioned in the previous subsection and the buffer matrix represents the requests placed in the buffer. In both of these matrices, the rows and columns correspond to the input and output ports, respectively; and each element corresponds to a list of times at which the packets are generated within a timeslot. For lowest latency, we would like to maximize the number of requests transferred from the buffer matrix to the connectivity matrix. The procedure to generate the connectivity matrix such that it satisfies a non-blocking condition is given as follows.

The connectivity matrix is initially empty and all the new requests are placed in the buffer matrix. The oldest M packets from each row and column of the buffer matrix are transferred to the connectivity matrix such that each input-output link supports less than L packets. At this stage, a heuristic algorithm to maximize the number of packets transmitted is implemented. The requests in the connectivity matrix are now transmitted at the start of the next timeslot.

Note that, so far in this algorithm we have not checked the wavelength continuity of the links, i.e., whether or not an end-to-end wavelength can be assigned to each

connection without encountering wavelength contention or dropping any requests from the connectivity matrix. If the connectivity matrix satisfies the two constraints given in the previous subsection, then it turns out that there exists an efficient, contentionless wavelength assignment algorithm associated with it. This is proved in the Appendix using graph theory arguments based on Hall's Marriage Theorem [47].

Detailed code of this contention resolution is given in Appendix: Code for Latency Simulations.

2.4 Performance Analysis

System without buffers

If the transmitters have no buffers, then the system performance would be measured in terms of the blocking probability, P_B, which is generally a function of L, M, N and Λ. For the case of L=1, $M >= N$, P_B is independent of M and N, and is given by

$$P_B(L=1) = \frac{\sum_{n=2}^{\infty} nP(n,\Lambda)}{\sum_{n=0}^{\infty} nP(n,\Lambda)} = \frac{\sum_{n=2}^{\infty} nP(n,\Lambda)}{\Lambda} = 1 - \frac{1}{\Lambda} + \frac{e^{-\Lambda}}{\Lambda} \tag{2.1}$$

Here,

$$P(n,\Lambda) = \frac{\Lambda^n}{n!}e^{-\Lambda} \tag{2.2}$$

(i.e., the Poisson Distribution)

n = number of requests generated

Λ = mean number of requests generated per timeslot

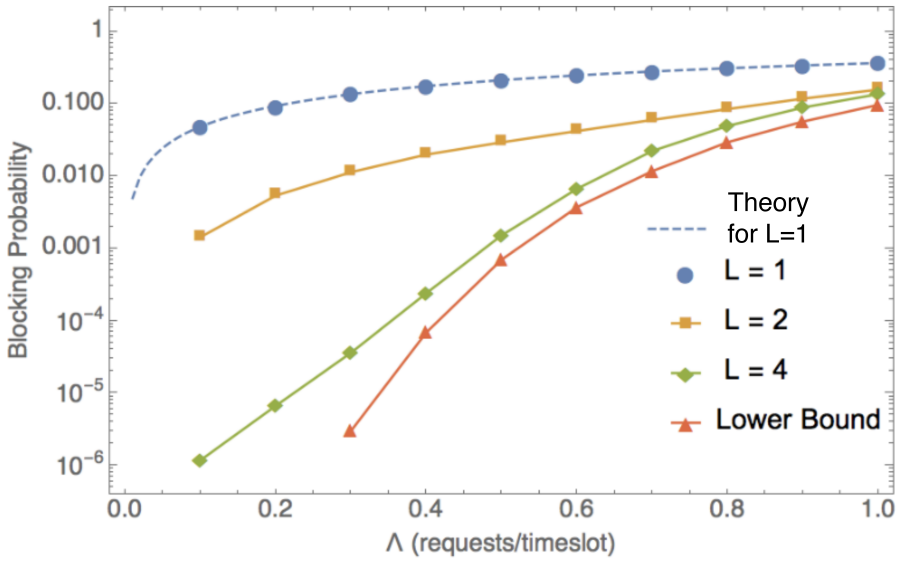


Figure 2.5: For systems without buffers, simulation of the Blocking Probability for M = N and different values of L.

Fig. 2.5 shows a simulation of the blocking probability as a function of Λ and L, for M=N=16. As expected, we can see from the figure that larger L (i.e., more connectivity) leads to lower blocking probability due to contention. In the figure, the lower bound is defined as the blocking probability occurring only at an input port when the number of requested wavelengths exceeds M. In some cases, if the packet generation rate, Λ, is small, the corresponding packet loss due to the blocking probability can be sufficiently small, e.g., $\leq 10^{-4}$, such that no buffers may be needed. However, in most systems, as the one we are considering here, no packet loss is acceptable. Thus, buffers are provided at the transmitters, as shown in Fig. 2.1 (b), and as discussed next.

System with buffers

With the transmitters equipped with buffers, the system performance would be measured in terms of packet latency, which is defined as the average time spent by a packet in the system, and is measured in units of timeslots. Eq. 2.3 gives the latency for L=1,M≥N.

For this case, the system reduces to a single link, which corresponds to the M/D/1 queueing model [48].

$$Latency(L = 1, M \geq N) = \frac{\Lambda}{(1 - \Lambda)} \tag{2.3}$$

In general, a lower bound on latency can be obtained by modeling the system as an M/D/m multi-server queue [48], with m=L=M, for which blocking occurs only due to the M-wavelength limit at each transmitter.

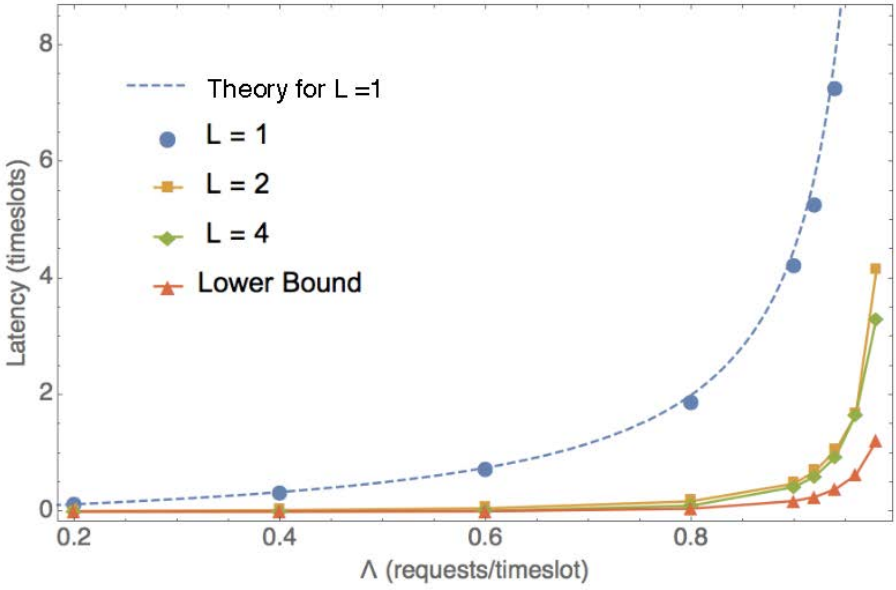


Figure 2.6: For system with buffers, simulation of Latency for different values of Λ and L for M = N=16. The lower limit on latency corresponds to an M/D/m queuing model with m=M, as explained in the text. Reprinted from [4] © The Optical Society.

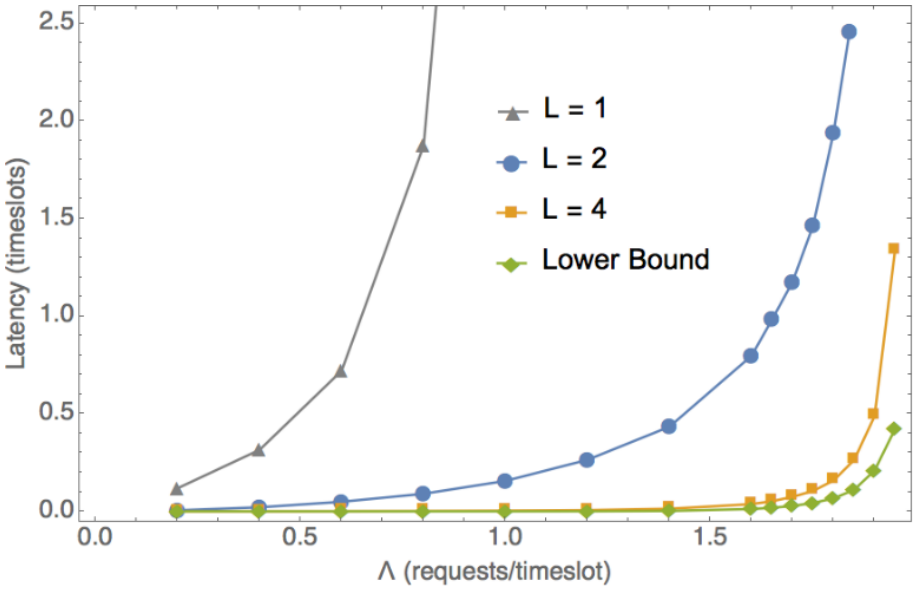


Figure 2.7: For system with buffers, simulation of Latency for different values of Λ and L for M = 2 N and N=16. The lower limit on latency corresponds to an M/D/m queueing model with m=M, as explained in the text.

Figures 2.6 and 2.7 show a simulation of latency for a number of wavelengths (M) and link connectivity (L) for a 16-port system (N=16). An intuitive estimate of the maximum throughput (Λ_{max}) is given by Minimum(L,M/N). For a fixed N, to increase Λ_{max}, one has to increase both L and M simultaneously. For a fixed N and M, higher L leads to reduced latency. The diminishing returns shown in the figures for increasing L is due to limit of M/N on Λ_{max}. This leads to smaller probability of generation of more than M/N requests per timeslot. The comparison between M=N and M=2N shows this effect, where for the latter case there is a significant latency reduction between L=2 and 4 as compared to the former case. For fixed N and L, no additional benefit is gained by increasing M beyond LxN.

In the blocking probability and latency simulations we have considered traffic pattern which changes every timeslot. If this switch is used in high performance computing,

traffic patterns might be quasi static, i.e. might have long holding times. For this case, choosing L more than the limit of 2 or 4 might give lower latency. Ultimately, the choice of L, M and N depends on the packet generation rate at the nodes, technology constraints on the wavelength selective switching elements and the application for which the switch is used.

We have not accounted in the above calculations for the network setup time, which includes the time for the contention-resolution/wavelength-assignment algorithm discussed in the section on wavelength assignment, and the time needed to tune the microrings to the desired wavelengths. A fixed network setup time should be added to the latency curves. To minimize the penalty due to this added time, the setup time needs to be much smaller than the message size. We believe that a total setup time on the order of a microsecond is possible, which would result in an efficient system for packet of intermediate and long sizes, e.g., on the order of a few microseconds. Our measurements show that the switching time for microring resonators is in 10 's of μs.

2.5 Wavelength assignment

Here, we describe an efficient, contentionless, wavelength assignment algorithm for our switch. Our algorithm is similar to an algorithm introduced in [45] for time-domain scheduling of a fast optical circuit switch in data centers. Let $A = [a_{i,j}, i, j = 1, 2, ..., N]$ be the NxN connectivity matrix discussed in Section on Contention resolution, where each $a_{i,j}$ is an integer representing the wavelength connectivity requested between input port i and output port j. The assignment constraints defined at the start of this chapter imply that $\sum_{i=1}^{N} a_{i,j} \leq M$,$\sum_{j=1}^{N} a_{i,j} \leq M$. The algorithm applies for arbitrary values of L.

By adding appropriate dummy connection requests, one can easily construct a com-

pleted connectivity matrix, $A' = [a_{i,j}, i, j = 1, 2, ..., N]$ such that

$$\sum_{i=1}^{N} a_{i,j} = M, \sum_{j=1}^{N} a_{i,j} = M \tag{2.4}$$

Argument is given in [49] using graph theoretic techniques based on Hall's Marriage Theorem [47] which proves that, if the matrix A'is represented by a bipartite graph with the input and output ports, and i,j, is the partitioned vertices of the graph, in which an edge exists between i and j if and only if $a'_{i,j} \neq 0$, then there exists a perfect matching in that graph (i.e., every input port in the graph is associated with a unique output port). The connectivity matrix for such a matching is an NxN permutation matrix, say, P_1, containing ones when i is matched with j, and zeroes elsewhere. Defining a matrix $A" = A' - P_1$, one can show that the elements of A" obey a similar relation to Eq. 2.4 above, with M replaced by M-1.

By induction, it is clear that the completed connectivity matrix A' can be expanded into the sum of M permutation matrices, i.e.,

$$A' = \sum_{k=1}^{M} P_k \tag{2.5}$$

Relating this to the problem of finding wavelength assignments for our switch, we can interpret each permutation matrix as a set of complete input-output connections for each one of the available M wavelengths. Efficient graph theory algorithms exist [47] for determining each matching, i.e., each permutation matrix. Thus, following this algorithm, one can obtain a full, contentionless, wavelength assignments associated the 'completed'connectivity matrix, A'.

The final connections and wavelength assignments associated with the original connectivity matrix A are easily obtained by removing the dummy connections that were

previously added to A to form A'.

2.6 Scalability

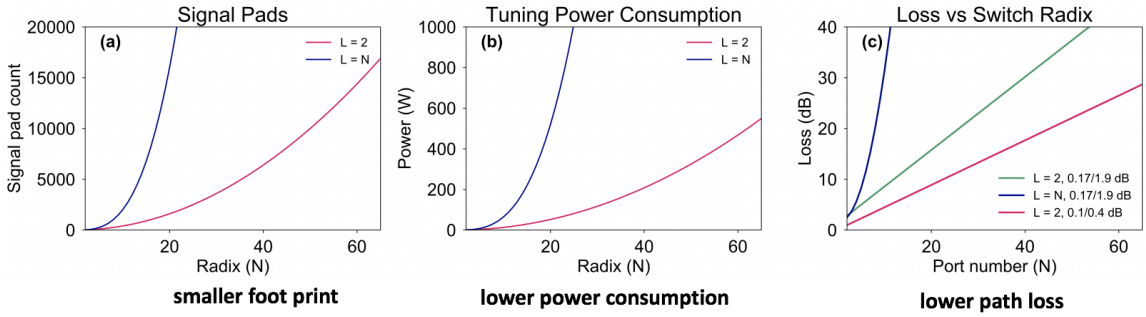


Figure 2.8: Scaling analysis of NxN switches for different L. (a) Complexity, (b) Tuning power consumption and (c) Loss vs. switch radix for different drop and through loss. The values for ring resonator off-resonance through and drop loss are the values of the ring resonators we measured in the lab. Reprinted from [4] © The Optical Society.

Complexity which is the number of switching elements is plotted against the Radix (N) in Fig 2.8 (a). For an arbitrary L and N the *complexity* $= LN^2$. Fig 2.8 (b) shows a plot of tuning power consumption vs. radix(N). Here a 30 mW/FSR tuning is assumed ,where on average each ring is tuned by a half FSR. Thus *tuning power consumption* $= 15LN^2$. The tuning power of 30 mW is a projected future value and the second order rings with the highest tuning range require 45 mW/ring/FSR tuning power. The best first order rings tune by one full FSR in 32.5 mW but due to its low resistance on the order of Ω require more than 20 mA of current to tune. The scalability simulations of loss vs. switch radix (N) for different L and off loss/drop loss values is shown in Fig 2.8 (c). In this plot, we observe 10.25 dB reduction in the insertion loss of the longest path between L=2, N=32 and 0.17/1.9 dB and 0.1/0.4 dB (off-resonance through port loss / on-resonance drop port loss) microresonator losses. For L = N and N = 32 we observe

56.6 dB reduction in insertion loss while at the same time reducing power consumption and complexity and hence footprint of the switch as shown in Fig 2.8(a) and (b).

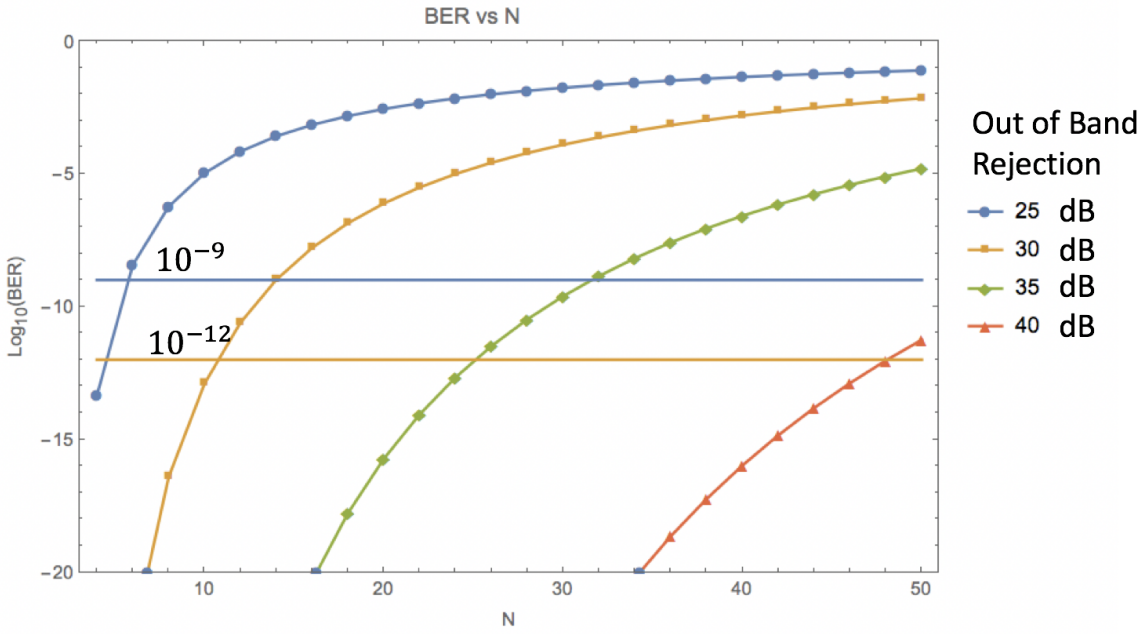


Figure 2.9: Simulation results of Bit Error Rate (BER) vs. Number of ports (N). For different filters with different out-of-band rejection one can see that to scale beyond 50 ports more than 35 dB out of band rejection is required.

Fig. 2.9 shows the BER vs. radix of the switch for different out of band rejection. One can see that a rejection greater than 35 dB is required for scaling beyond 50 ports. Crosstalk simulations follow a procedure similar to [50]. This analysis is for single wavelength input and ignores the effect of wavelength input from other input ports. The loss of the switch is path dependent, and the loss difference between the smallest loss path and the largest loss path needs to be within the receiver dynamic range. A constraint on the receiver dynamic range of 10 dB, sets the difference in minimum and maximum path dependent insertion loss of 10 dB. For an $L = 2$, 32×32 and 64×64 switch this corresponds to a loss of 0.16 dB and 0.08 dB per unit cell where both resonators are tuned off resonance. This corresponds to an off-resonance ring loss of 0.0325 dB and

0.025 dB assuming ultralow loss silicon photonic waveguide crossings are used [51]. This projected future loss by these improvements in technology should result in junction loss much lower than the measured off resonance loss of 0.3 dB in the later section.

2.7 Summary and Conclusion

In this chapter, we discussed the architecture and system design of the switch. As the wavelengths in the switch are the key resources which are shared, a contention resolution scheme is required. We proposed and analyzed the tradeoffs with various number of ring resonators on every crosspoint of the switch with respect to latency, blocking probability and footprint, power consumption and complexity of the switch.

Chapter 3

2 λ switches

3.1 Introduction

In the previous chapter, we proposed a new switch network and performed a theoretical analysis of performance metrics. In this chapter, we experimentally demonstrate 8x8, 8x4, and 4x4 switches across various generations. 8x8 and 8x4 switches were implemented with thermal and electro-optic control and 4x4 switches were implemented with only thermal control. This is followed by a discussion on previous generation of switches.

3.2 Ring resonators

In this chapter, unless otherwise stated fiber coupling losses due to edge couplers and fiber connector losses are subtracted from the measured loss. We measured a coupling loss of 2.7 dB per facet for edge couplers. The measured distribution for the first two generations is given later in the chapter in the section on previous generation switches. Typical measurement procedure was to first test on an optical waveguide connected by

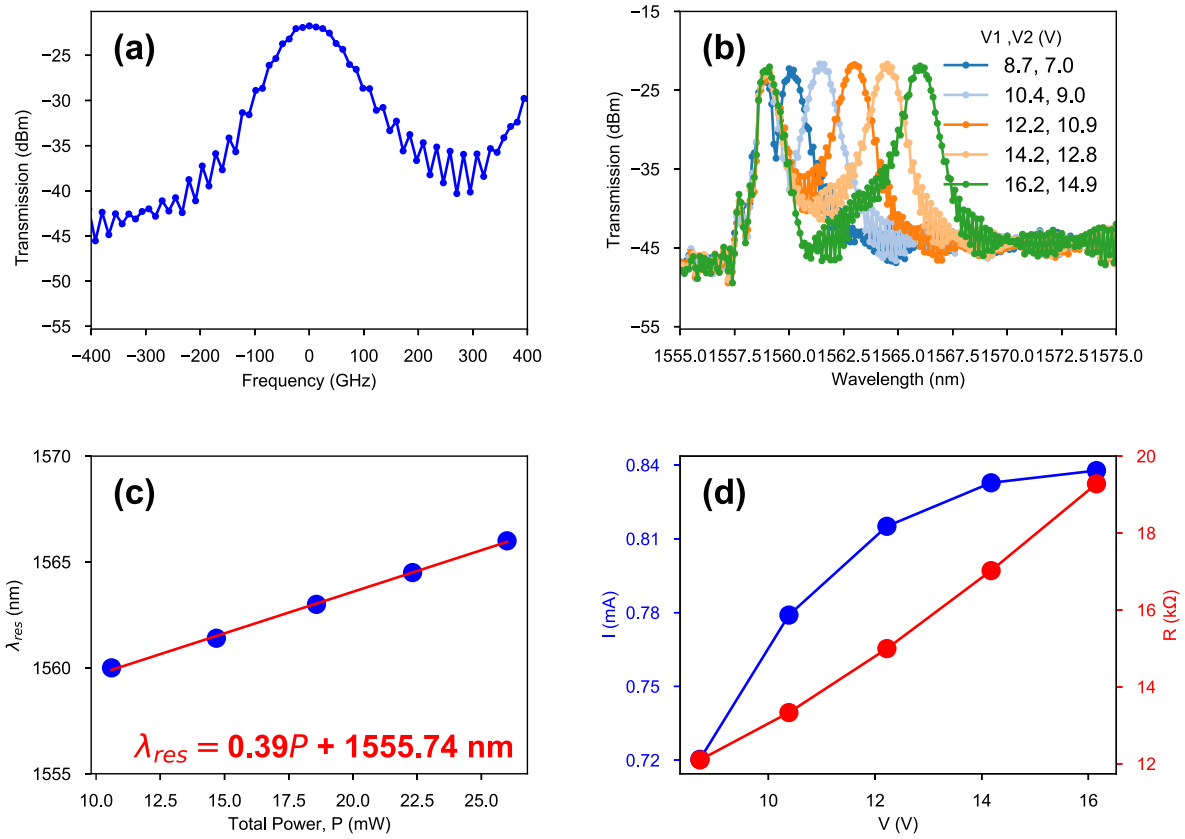


Figure 3.1: Measurements on 2nd order ring resonator filter. This ring resonator was manufactured in a 220 nm silicon foundry with AIM Process Design Kit (PDK) ring resonators. The exact specs of the ring are not known. The ring resonator can perform both electrooptic tuning and thermooptic tuning. Measurements were performed with a tunable laser at 1550 nm. Fiber Coupling losses due to edge couplers and coupling losses from instrument connectors are not subtracted in this measurement.

two edge couplers and then subtract this loss from the measured device loss. Fig 3.1 (a) shows the measured second order ring resonator filter response. The asymetry in the filter shape is because this measurement was done on a unit cell with two second order ring resonators with another 2nd order filter present at approximately 400 GHz spacing. We measure a 3-dB filter bandwidth of 100 GHz and Free Spectral Range (FSR) of 25.6 nm. Out of band rejection at 200 and 400 GHz channel spacing is 20 and 32 dB respectively.

Fig 3.1 (b) shows the tuning plot of second order filters. The voltages on the legend are different because the heater resistances on the two coupled rings are slightly mismatched.

These voltages are optimized by genetic algorithms for faster convergence as described in Chapter 3, on chip locking.

Fig 3.1 (c) shows the extracted tuning curve from Fig 3.1 (b). I used Numpy polyfit [52] for first order polynomial fitting. I observe a 0.39 nm/mW tuning efficiency and this value is calculated for two second order ring resonators. The resonance wavelength at zero bias is 1555.74 nm. In Fig 3.1 (d), I report the heater IV and RV curves for the ring resonator heater resistance. The figure shows that the heater resistance increases from 12 $k\Omega$ to 20 $k\Omega$ with a maximum current of 0.84 mA.

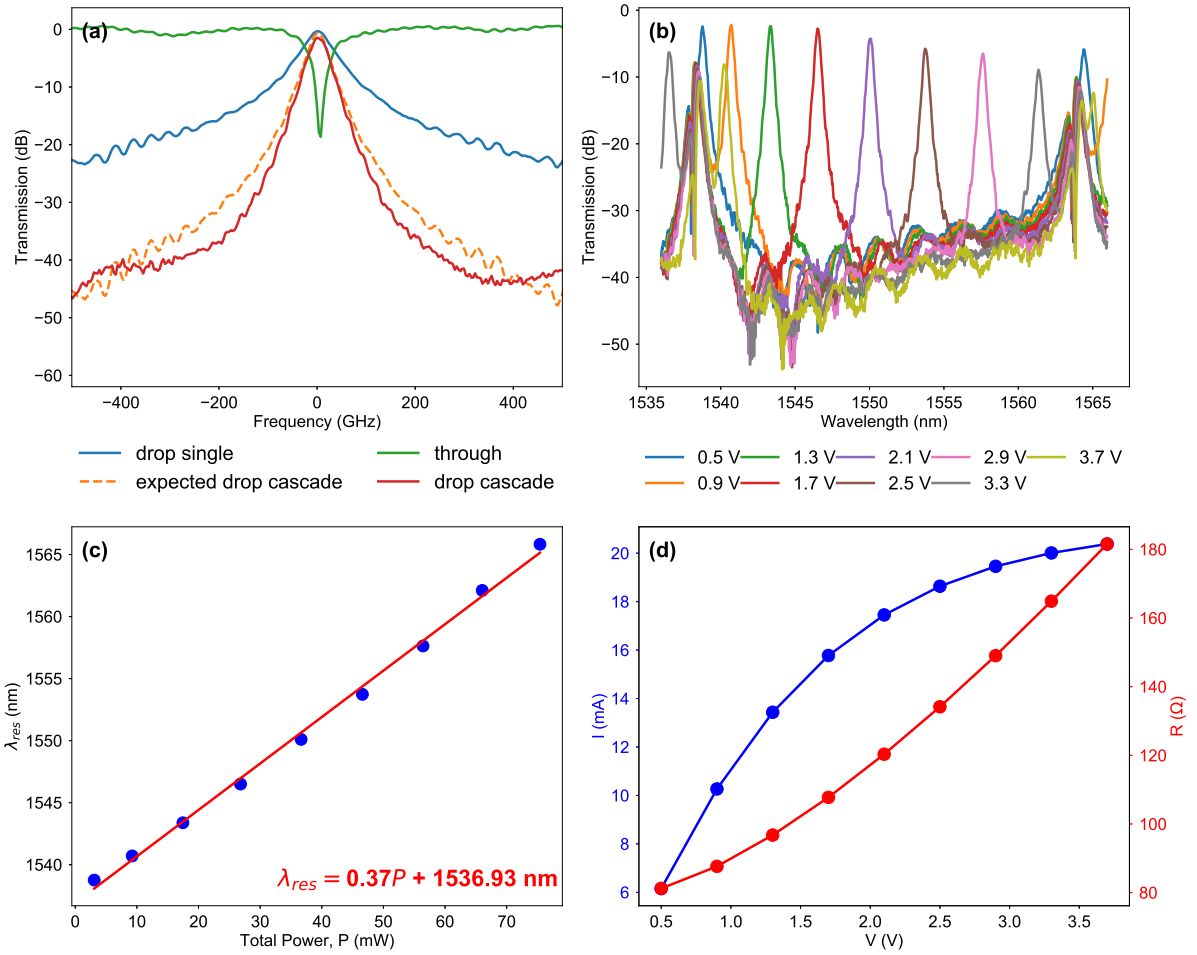


Figure 3.2: Measurements on cascaded first order ring resonator (a) Drop and through transfer spectra of a single first order ring resonator, Expected transfer spectra of cascaded order ring and the best measured cascaded ring, (b) Tuning curve of cascaded ring in a unit cell arrangement where bias is applied to only one ring,(c) Full FSR tuning is demonstrated. The ring tunes by 27.6 nm (> FSR = 25.6 nm) with power efficiency of 0.37 nm/mW for two rings.(d) IV and RV curves for the cascaded ring resonators. Nonlinear dependence of R vs. V is measured.

Fig 3.2 (a) shows the measured Transfer spectra comparison of one and two cascaded first-order ring resonators. The out-of-band rejection 400 GHz away from resonance wavelength of ring resonator increases from -21 dB to -40 dB as we change the ring filter from one to two cascaded filters. The 3dB bandwidth of the cascaded ring resonator changes from 64 GHz to 38 GHz. This is the expected 3dB bandwidth of the cascaded

filters. The curve in red shows the measured cascaded filter response from one of the unit cells in the switch. The filter 3dB bandwidth is 40 GHz which is larger than the expected 3dB bandwidth of 38 GHz because the heaters of the two ring resonators are tied together. Thus, individual control of resonators is not possible and any misalignment between the resonators changes the filter shape. The motivation behind this design approach is further elaborated in Section 3.5. The extinction on the through port is -17 dB.

Fig 3.2(b) shows the tuning curve of the cascaded first order ring filter. In this measurement, an increasing voltage is applied to one cascaded ring from a unit cell of the 4x4 switch from Section 3.5, while no bias is applied to the other ring. The ring tunes by 27.06 nm/25.6 nm with a tuning efficiency of 0.37 nm/mW (for both rings). This ring tunes by at least one Free Spectral Range and thus can be used for selecting all WDM channels present in the system. The cascaded second order ring resonator has similar efficiency as compared with second order resonators from the AIM PDK Fig. 3.1. The drop and off resonance loss of this filter are less than 0.2 dB and 0.5 dB respectively which is much smaller than the measured loss for second order ring filters from Fig 3.1.

The low loss and full FSR tuning make this ring filter an ideal candidate for opto-electronic switches. Tuning curve and Heater IV and RV are reported in Fig 3.2 (c) and (d). The tuning curve demonstrates full FSR tuning. The resonance wavelength is 1536.93 nm and the histogram of resonant wavelengths measured on different filters on the die is given in Section3.5. Heater IV in Fig 3.2 (d) shows that the current changes from 6 mA to 20 mA as the voltage changes from 0 to 3.7 mA. The resistance changes from 80 Ω to 180 Ω.

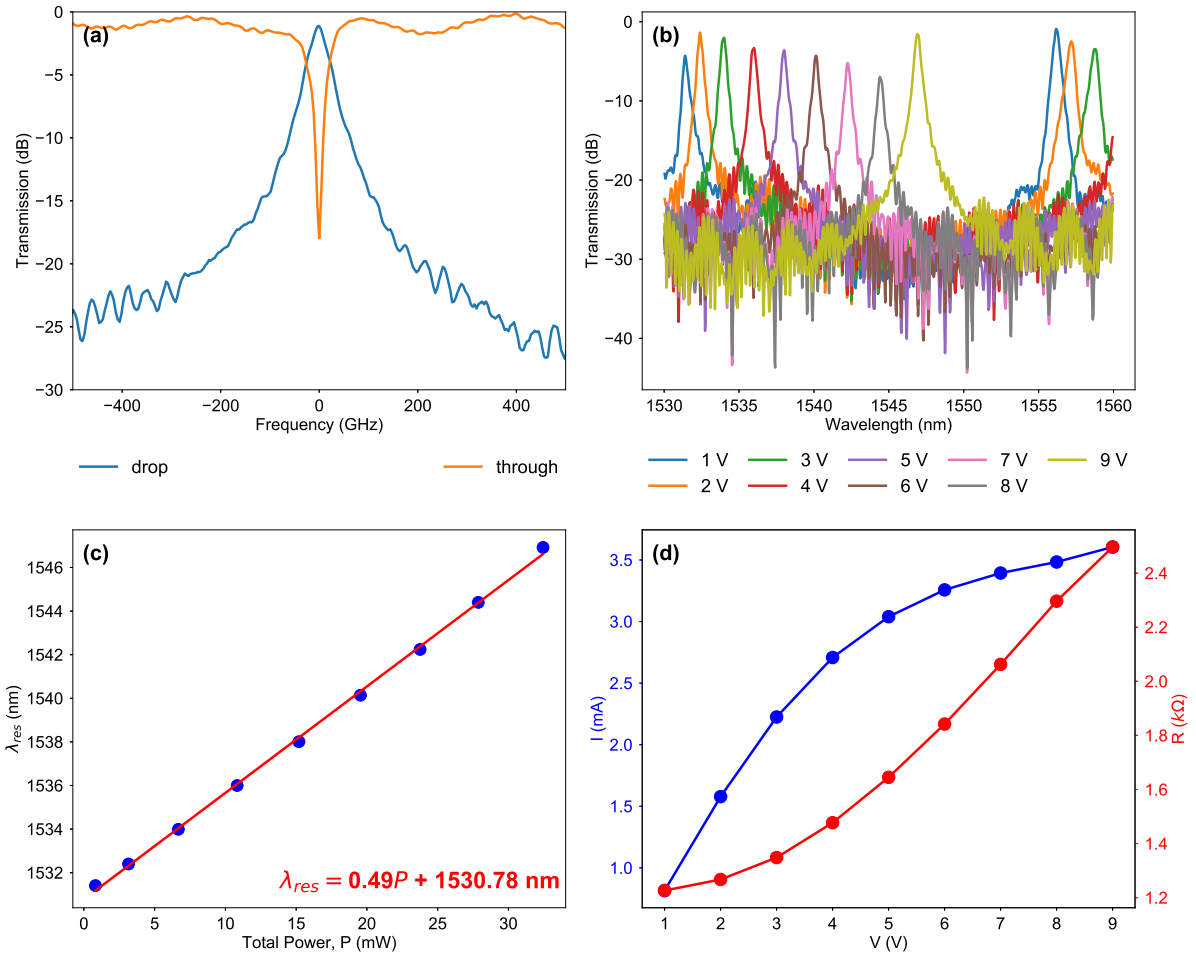


Figure 3.3: Measurements on custom first order ring resonator (a) Drop and through transfer spectra of a single first-order ring resonator, (b) Tuning curve of custom ring,(c) Tuning is demonstrated. The ring tunes by 16 nm (FSR = 25.6 nm) with power efficiency of 0.49 nm/mW. (d) IV and RV curves for the cascaded ring resonators. Nonlinear dependence of R vs. V is measured.

Fig 3.3 shows a custom ring resonator with adiabatic heater design. There was no contact of heater with the waveguide, and measurements shows a very low drop and through loss. The ring resonator coupler was designed using Bezier curves and optimized for minimum scattering loss using Lumerical 3D FDTD.

3.3 8x4 L=2 switch: doped second order

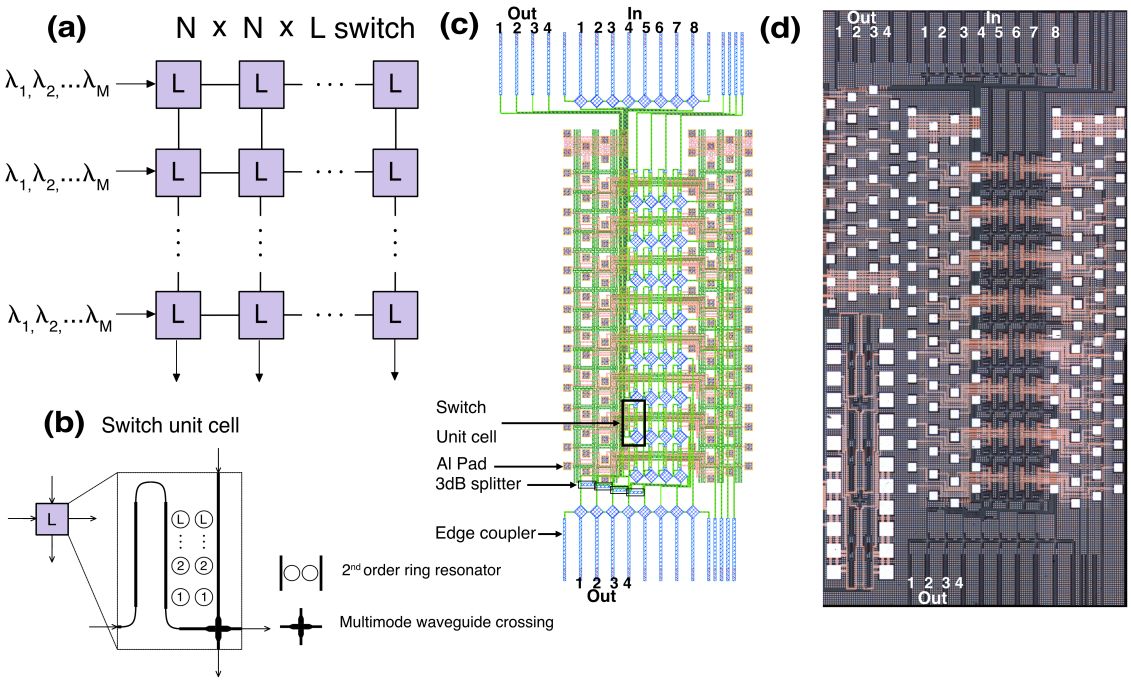


Figure 3.4: (a) Schematic of switch with L=2, (b) Switch unit cell (c) Layout of a 8x4,L=2 switch (d) Die Shot of the switch with I/O marked. Reprinted from [4] © The Optical Society.

The photonic switching chip was fabricated on 220 nm Si photonics platform at AIM Photonics foundry at SUNY Albany [53]. The waveguides used for routing were deep etched 400 nm with an expected loss of $\leq$2 dB/cm, although subsequent process optimization has improved this waveguide loss significantly. Edge couplers were used for optical I/O. Diced dies had a 100 μm trench etch for fiber coupling.

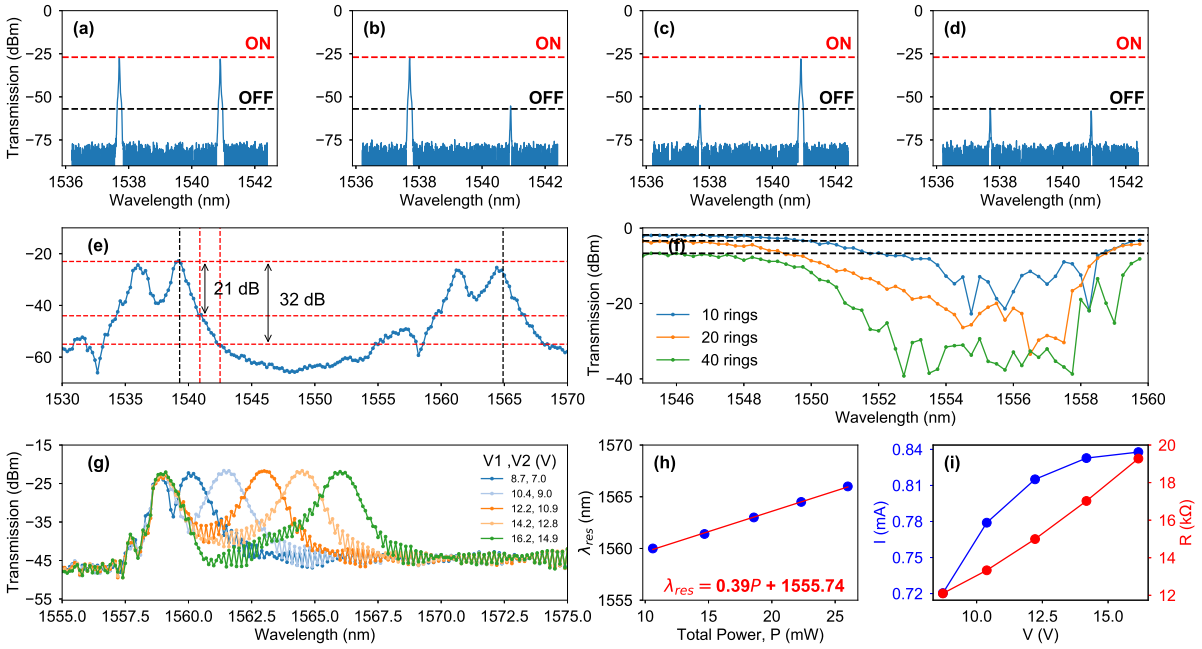


Figure 3.5: Two λ switching is demonstrated in (a) λ_1 ON and λ_2 ON, (b) λ_1 ON and λ_2 OFF, (c) λ_1 OFF and λ_1 ON, (d) λ_1 OFF and λ_1 OFF, (e) Transmission spectra of a typical unit cell, the black lines correspond to one FSR = 25.6 nm and typical out of band rejection is 32 dB for 400 GHz spacing (f) Off-resonance loss of 10, 20, and 40 ring resonators in series, (g) Tuning curve of a micro-ring resonator in a unit cell, (h) Tuning efficiency of 0.39 nm/mW is calculated from the tuning curve. Here we plot λ_{res} vs. Total Power,P (mW), (i) Heater I-V and R-V is plotted and shows that heater resistance changes monotonically with bias voltage. Reprinted from [4] © The Optical Society.

Fig. 3.5 (a), (b), (c) and (d) shows Optical Spectrum Analyzer (OSA) traces demonstrating two-wavelength switching for a unit cell in the switch (Fig. 3.4 (b)). For this experiment two wavelength channels at 400 GHz/3.2 nm spacing were injected with a fused silica splitter into the input port 1, and light is collected from output port 1. Here we observe a 32-dB extinction between ON and OFF signal. In this measurement, ring resonators have a resonance wavelength of 1534.5 nm at zero bias voltage. This wavelength is the guard band where all the ring resonators filters are parked when no connection is assigned to them. Two channels at 400 GHz spacing at 1537.7 nm and 1540.9 nm are used in the measurement.

Fig. 3.5 (e) shows the filter spectrum at the output port 3 with input at port 3. The figure shows transmission spectra of a unit cell with channels spaced at 3.2 nm (400 GHz) channel spacing. We measure a 100 GHz optical 3-dB bandwidth of the filter and an out-of-band rejection of 21 dB/32 dB at 200 and 400 GHz channel spacing respectively. The ring resonators have a measured Free Spectral Range (FSR) of 25.6 nm.

The PDK second order ring resonators are capable of both electro-optic tuning and thermal tuning. For electro optic tuning, the resonance wavelength was blue-shifted and attenuated with a loss of 3.6 dB/nm. Hence only thermal tuning was used for filter tuning. In Fig. 3.5 (f), a through-port measurement is done on 10, 20 and 40 second order ring resonators. Only one resonator in each arrangement is connected to electrical pads to save space on the layout. The measured through loss was 0.17 dB per ring resonator and 1.9 dB for the drop port.This measurement was done to calculate the off resonance loss of a ring resonator filter, which is a crucial factor for scalability of switches. Fig 3.5 (f) shows that many ring resonators in the series arrangement are not aligned to the same resonant wavelength. The standard deviation of ring resonance frequency as well as of free spectral range on a different process are well documented in [54].

Transmission spectra for the unit cell of the switch at different voltages are shown in 3.5 (f), and in 3.5 (h) we extract the tuning curve for the switch ring resonators. The tuning efficiency of the second order rings is 0.39 nm/mW with a measured resonance wavelength at zero bias of 1554.74 nm. The maximum ring resonator tuning range was maximum 11 nm. I-V sweep of ring resonator heaters and resistance vs. voltage is shown in Fig 3.5 (i). The figure shows that resistance changes from 12 $k\Omega$ to 20 $k\Omega$ as voltage is changed from 8 V to 16 V. The same plot also shows the change in current supplied. A smaller drive voltage $V < 10V$ is preferred for CMOS and HBT processes. Thus a smaller ring heater resistance than the one measured is preferred for electronic photonic integration.

The PCB was designed in Altium Designer and fabricated by Advanced Circuits. The PCB was designed to have a PIC of up to 6 mm x 9 mm mounted on its surface with one edge overhanging the edge of the board by 250 μm in order to easily accommodate edge coupling to a V-groove fiber array. Standard design rules for the widths and separations of signal traces were maintained, but a more advanced mechanical drill diameter of 300 μm was selected to reduce wire bond pad size. A maximum of 72 signal and 20 ground wirebondable pads on the PCB, were due to a constraint on the maximum wirebond length of 8.5 mm set by the vendor.

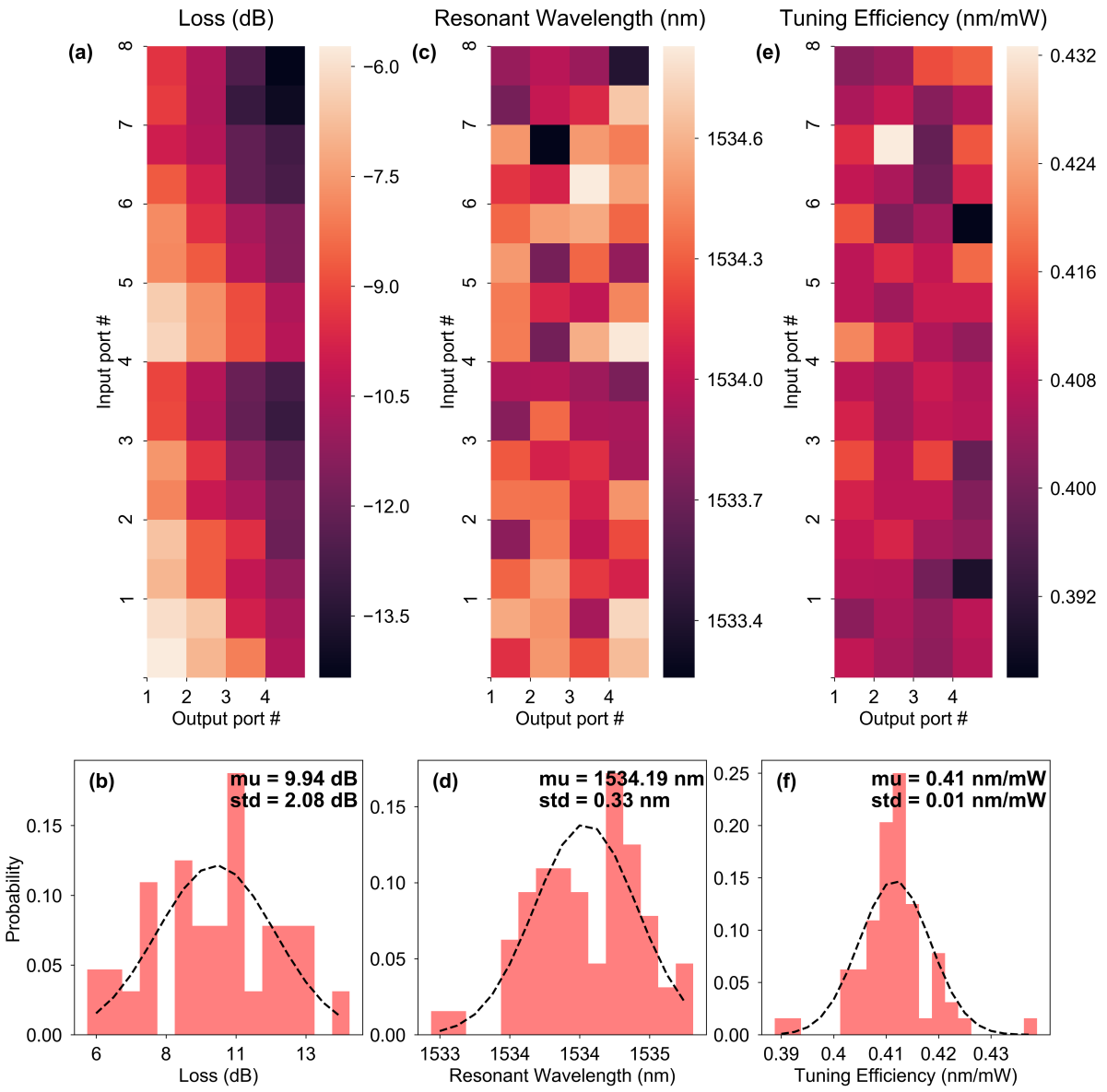


Figure 3.6: Heatmaps of (a) loss, (b) resonant Wavelength and (c) tuning Efficiency at different locations in the 8×4 switch, Corresponding histograms of (d) loss, (e) resonant wavelength and (f) tuning efficiency. Reprinted from [4] © The Optical Society.

In Fig. 3.6 we characterize a single switch with 64 2nd-order ring resonators. In Fig. 3.6(a) and (b) we measure a path loss across different paths in the 8×4 with mean 9.94 dB and standard deviation of 2.08 dB. The minimum and max loss in the switch path were 6 dB and 14 dB. These values correspond to input 1 - output 1 and input 8 -

output 4. This results in a waveguide crossing loss of 0.3 dB, through port loss of 0.3 dB and drop loss of 4.7 dB, measured in the C-band. In Fig. 3.6(c) and (d) we measure a resonant wavelength with mean 1534.19 nm and standard deviation of 0.33 nm. In Fig. 3.6(e) and (f) we measure tuning efficiency with mean and standard deviation of 0.41 nm and 0.01 nm.

We performed 256 total measurements to generate Fig. 3.6, 4 measurements are conducted on each of the 64 ring resonators of the 8×4 switch. For each measurement the two voltages on the micro rings are optimized to get the best peak drop port transmission [55]. As reported in [5], the peak power is unimodal function of the heater voltages, i.e. has only one peak. Resonant tuning optimization is automated with Python. In such cases, it is faster to optimize the voltages through a genetic algorithm rather than a grid search on the two heater voltages. All the measurements in this paper were conducted with Random Mutation Hill Climbing (SHC)[56]. In this method, random voltages are set on the resonator pads. In each iteration, gaussian noise is added to each voltage and if the light output is greater than previous, voltages are updated. Code is provided in the Appendix. For fast convergence, an approximate tuning curve is generated, and we assign random initial guess voltages to the ring heaters depending on the tuning curve and the wavelength of interest. As ring resonator resonant wavelengths follow a probability distribution given by Fig. 3.6 (d), we apply noise sampled from standard normal distribution to the voltages and select the best pair of voltages to maximize the peak optical transmission. Noise can be scaled by a scalar, which can control the search space around the random guess voltage.

The heater driver is designed to control 64 heaters. Each commercial DAC used for the driver has 16 channels, so 4 DACs are required to control 64 heaters. A USB to Serial-Parallel-Interface (SPI) converter block is also used for controlling the DACs. The data sent to the USB will be converted to SPI using an FTDI chip. Outputs of DACs

then are connected to photonic chip.To control 64 heaters individually, 4 DACs with 16 channels were utilized. Each DAC has a Serial-Parallel-Interface (SPI) interface. To control 4 DACs, a Master-Slave configuration is used. A Graphical-User-Interface (GUI) is designed in Microsoft Visual Basic.

The maximum current flowing through heaters is 2mA at the expected 5K resistance. We picked the LTC2668 for a ring driver to provide a driver level up to 10 V. The 10 V full scale range translates to 2.44 mV least significant bit (LSB) steps at 12 bits of resolution. Also, considering the small time constant of the heater, this DAC was chosen to have a faster settling time than the custom PIC heaters. Settling time of selected DAC is 8 μs for 0 to 10 V step. To avoid digital signals from coupling to outputs and decreasing the noise introduced to the circuit, analog and digital signals in the PCB are separated. An FT4222 is used which is functioning as both Master and Slave, which can control 4 SPI connections as required for controlling 4 DACs on the PCB and at high speed.

In this experiment, rise time is defined as 10 to 90 % rise time and fall time is defined as 90 to 10 % fall time. To measure the time constant of heaters, a photodetector (xpdv2120R) is used. The input to the heaters is applied using the heater driver board. For a double-ring switching element, one ring is fixed once at 10V and once at 0V and the other ring is changed using a step voltage from 0 to 10V, while rise and fall times are measured at the output of photodiode. This procedure is repeated for different voltage differences and the switching time is recorded. Fig 3.7 shows the block diagram for the experiment.

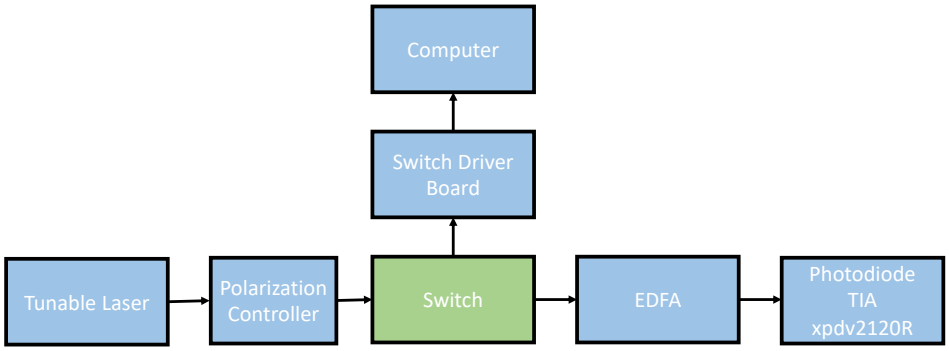


Figure 3.7: Block diagram for heater measurements. The voltages on the ring resonators were changed from the PC. Trans impedence amplifier was used to amplify the photodiode current.

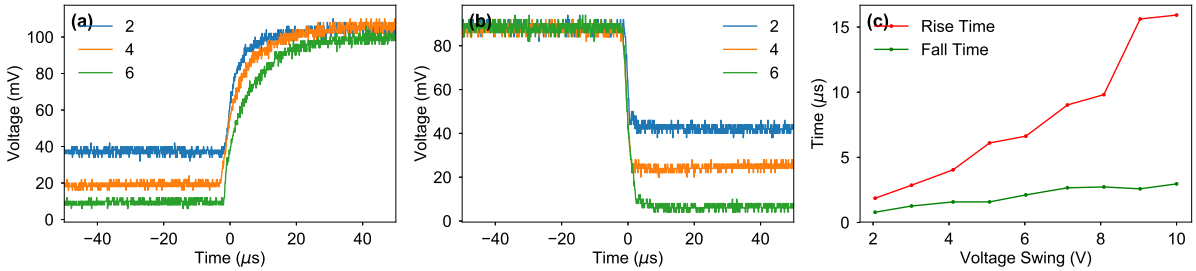


Figure 3.8: (a) 10 to 90 % Rise Time for three different voltage swing, (b) 90 to 10 % Fall Time for three different voltage swing and (c) 10 to 90 % Rise time, and 90 to 10 % Fall time vs. Voltage swing. These measurements were conducted in the C band. The ring FSR was 25.6 nm and the ring dimensions are unknown as this was a foundry PDK ring. Reprinted from [4] © The Optical Society.

Fig. 3.8 (a) shows the rise time for three different voltage swings. In this experiment, peak transmission of ring resonator was optimized at voltage of 10 V and voltage on one of the rings out of double ring resonator was changed by the voltage swing to 10 V. Fig. 3.8 (b) repeats the same experiment in reverse for the fall time. Fig. 3.8 (c)

shows that rise time and fall time monotonically increase with the voltage difference swing applied to the ring resonators. From this figure we can see that the worst case switching time is 15.92 μs. A rise time from 10 % to 90 % is measured and shown for three different voltage differences. As we tune from one channel to another, higher wavelength separation results in higher switching time. This large signal switching is very different from the small-signal switching time reported in papers on ring resonators [57] where switching time $\tau < 5$ s switching time was measured by applying a small electrical signal around the 3-dB point of the filter. This reported small-signal switching time does not apply to wavelength selective switching time as tuning across all channels is required, hence the motivation for measurements in Fig. 3.8 (c).

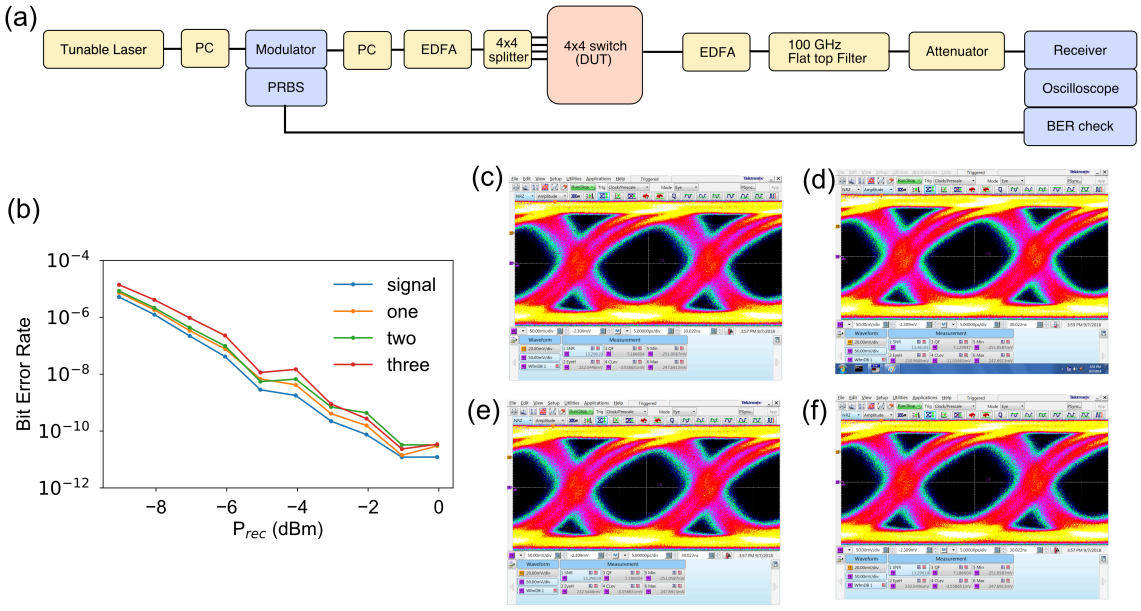


Figure 3.9: (a) Test setup from multi-channel BERT testing at 40 Gbps (b) Multi--wavelength crosstalk measurement for 0, 1, 2,and 3 crosstalk sources, Eye diagram for (c) no crosstalk source, (d) one crosstalk sources, (e) two crosstalk sources, and (f) three crosstalk sources The error floor in (b) is a measurement artifact. Reprinted from [4] © The Optical Society.

The Bit Error Rate (BER) test setup for multiple input crosstalk impairment testing

is shown in Fig. 3.9 (a). In this experiment, modulated signal at 1537.6 nm is split with a 4×4 splitter and injected into input port 1,2,3 and 4. We calculated the coherence length of the Yenista Tunics T100S-HP source as 238 m from the 400 kHz linewidth. Fiber delays of 500m ,1.4 km and 6.4 km were used to decorrelate the optical phases of different input channels, which would otherwise occur due to the optical source coherence time. We used a 40 Gbps modulator from Sumitomo (T.MXH1.5-4OPD-ADC-LV) with measured insertion loss of 4 dB. An optical preamplifier and post amplifier were required to compensate for interconnect losses including losses from 4×4 splitter and modulator insertion loss. A Finisar high Gain 43 Gb/s photoreceiver XPRV2022(A) was used for detection of transmitted data. Fig. 3.9 (b) shows a negligible penalty in BER when multiple crosstalk sources are added to the input port. Fig. 3.9 (c), (d), (e) and (f) correspond to eye diagrams with no crosstalk sources and sources at input 2, 3 and 4 respectively.

The loss of the switch is path dependent. The loss difference between the smallest loss path and the largest loss path needs to be within the receiver dynamic range. A constraint on the receiver dynamic range of 10 dB sets the difference in minimum and maximum path dependent insertion loss of 10 dB. For an $L = 2$, 32×32 and 64×64 switch. This implies an off-resonance loss of 0.16 dB and 0.08 dB per unit cell where both resonators are tuned off resonance. This corresponds to an off resonance ring loss of 0.065 dB and 0.025 dB assuming ultralow loss silicon photonic waveguide crossings are used [51]. This projected future loss resulting from process improvements occurring during the course of this research is much lower than the measured off resonance loss of 0.3 dB with histograms in Fig 3.6(b).

All the resonators voltages were optimized with a open loop control using photo detectors outside the chip package. A closed loop control similar to [5] is possible if on-chip wavelength selective taps are available. Another option is to control the switch

by calibrating the ring heater vs. temperature plot for all the ring resonators in the switch.

3.4 4x4 L=2 switch: undoped second order

In this subsection, we present a low loss 4x4 switch with L=2. This switch has the following differences as compared to the switch from the previous subsection:

1. Reduced average path loss of 2.65 dB which is smaller by a factor of 4 as compared to the 4x4 sub-section of the 8x4 switch measured in previous subsection.

2. Increased tuning range of 17 nm out of measured FSR of 20 nm which is 85 % as compared to 11 nm out of 26 nm (42 %) previously measured.

A box plot of measured path losses is shown in Fig. 3.10. This kind of plot is also called the box and whisker plot and compares two distributions based on minimum, maximum,median and interquartile range. We observe a smaller median (red line) for the custom switch and the maximum path loss of the custom switch is smaller than the minimum loss of the PDK switch. The Inter quartile range (IQR) or the width of the box is smaller for the custom switch as compared to the PDK switch.

The reduction in losses are attributed to lower doping in the ring resonator. We used ring resonator designs from [55]. This ring resonator uses a Bezier curve with parameters optimized to reduce the scattering loss from straight to curved transition. The ring resonator is undoped and heaters are disconnected from the optical waveguide. This reduces the drop and through loss of the switch, but at the same time reduces the tuning efficiency of ring resonator by 0.4 for the PDK rings to 0.2 nm/mW for this second order resonator design. The ring spectra shown in Fig. 3.11 (c) shows a FWHM of 0.96 nm (120 GHz). A higher bandwidth corresponds to a lower Q and thus lower losses as light

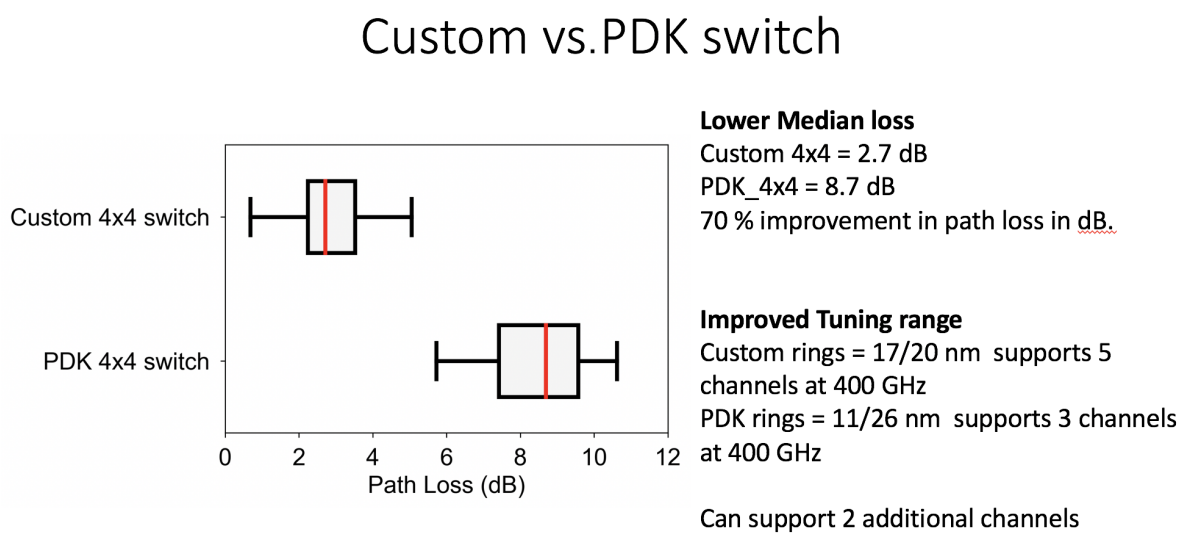


Figure 3.10: A box plot of custom vs. PDK switch. This plot shows that the maximum path loss measured for the custom switch is lower than the PDK switch. Red lines show the median path loss. The box edges correspond to the first and third quartile in the loss measurement data. The variation of interquartile range is the box width and one can see that custom switch exhibits a lower IQR.

circulates for less times in the ring. The figure also shows critical coupling in the second order resonator of ≥ 30 dB due to its ultra-low losses.

Fig. 3.11 (a) shows the architecture of the switch with second order resonators and this architecture is implemented in Fig. 3.11 (b). The dimensions of the switch are 3 mm x 1.5 mm. The switch has 4 inputs and 4 outputs spaced at 50 μm. This spacing is used to reduce optical crosstalk between edge couplers. Four through ports are also available for testing to debug the switch performance. The pad pitch of the switches is $160\mu m$ and pad dimensions are $60\mu m$ x $60\mu m$. Wires are routed in second to last metal layer with a thickness of 5 μm with a $1\mu m/mA$ routing rule. This number can be calculated from electro-migration limit of wires for a given fabrication process. Ground pads are connected with a grid of wires to maintain a uniform metal density for the fabrication

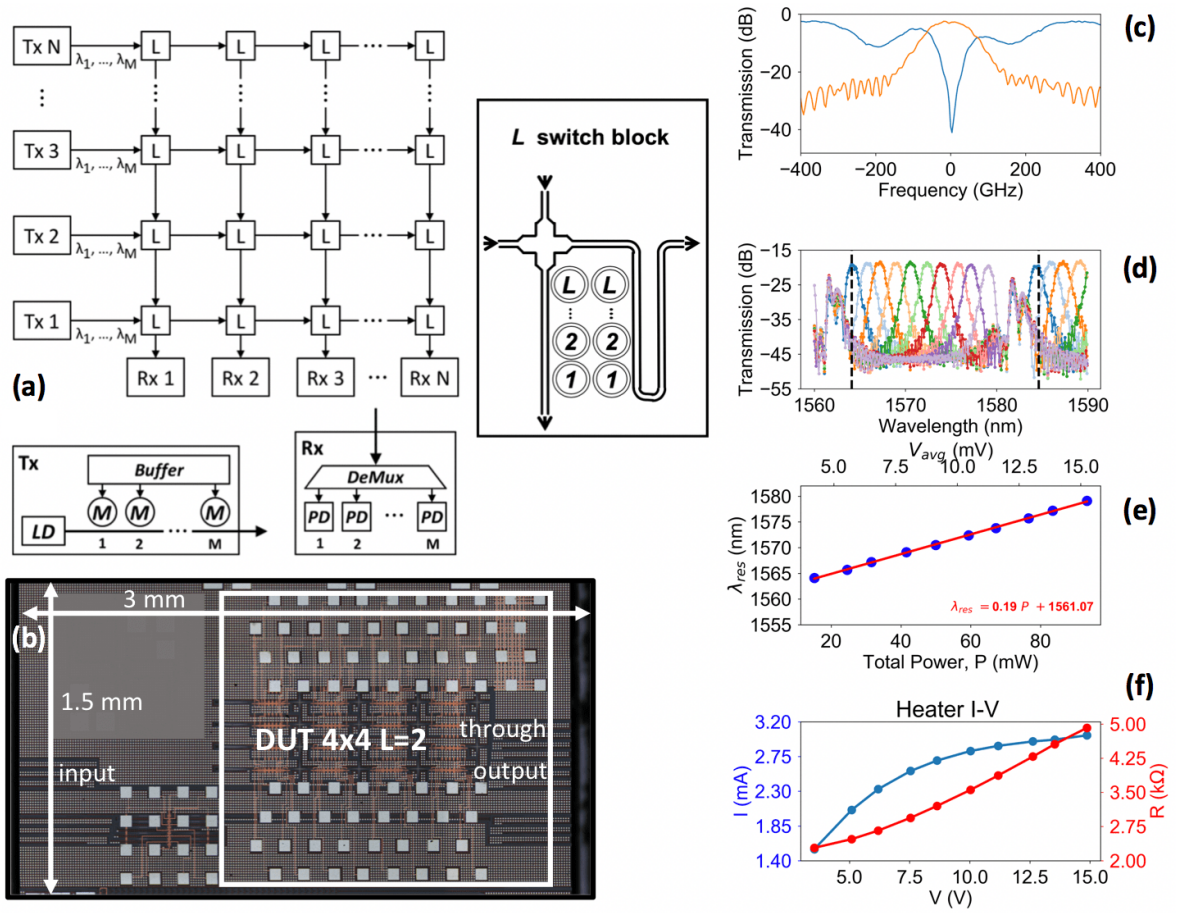


Figure 3.11: (a) shows the switch architecture, (b) Switch micrograph also the Device under test (DUT), (c) Through and drop spectra of the switch. FWHM of 120 GHz and through port extinction of $\geq 30dB$ is measured. (d)Transfer curved with various voltages applied, (e) Tuning curve with tuning efficiency of 0.19 nm/mW and tuning range of 17 nm (85 % of FSR). (f) Heater IV and RV curves.

process. The pads were filled with vias to prevent pad metal peel off.

Fig. 3.11 (d) shows the transfer spectrum of the second order ring as different voltages are applied to it. The plot shows tuning of 17 nm out of 20 nm FSR. This measurement was conducted on a unit cell containing two second-order ring resonators and one waveguide crossing. Voltage is applied to one of the ring resonators and no bias is applied to the other. Fig. 3.11 (e) shows the extracted tuning curve. The calculated tuning efficiency of the device is 0.19 nm/mW and 1561.07 nm. As mentioned before, this value

is half that of the pdk ring due to low efficiency heaters. Fig. 3.11 (f) shows heater IV and RV, these curves show a nonlinear dependence of heater current on applied voltage. The resistance of the ring heater changes from $2k\Omega$ to $5k\Omega$, while the current changes from $1.4mA$ and $3mA$. The power required to tune to the tuning limit (85 % of the FSR) of the ring is 90 mW.

Fig. 3.12 (a) shows the test setup used for the insertion loss measurements of the switch. This measurement was conducted with DC probes and cleaved fibers. The setup of the switch with vacuum suction stage is shown in Fig. 3.12 (b).Fig. 3.12 (c) shows a zoomed version of the DUT. Fig. 3.12 (d) shows the path loss for different input output port pairs. The path names are on the x axis. Path label of 13 corresponds to input 1 and output 3. The loss pattern corresponds to 0.91 dB average loss and 0.25 dB standard deviation of off-resonance or pass through loss for a unit cell as shown in Fig. 3.12 (f).

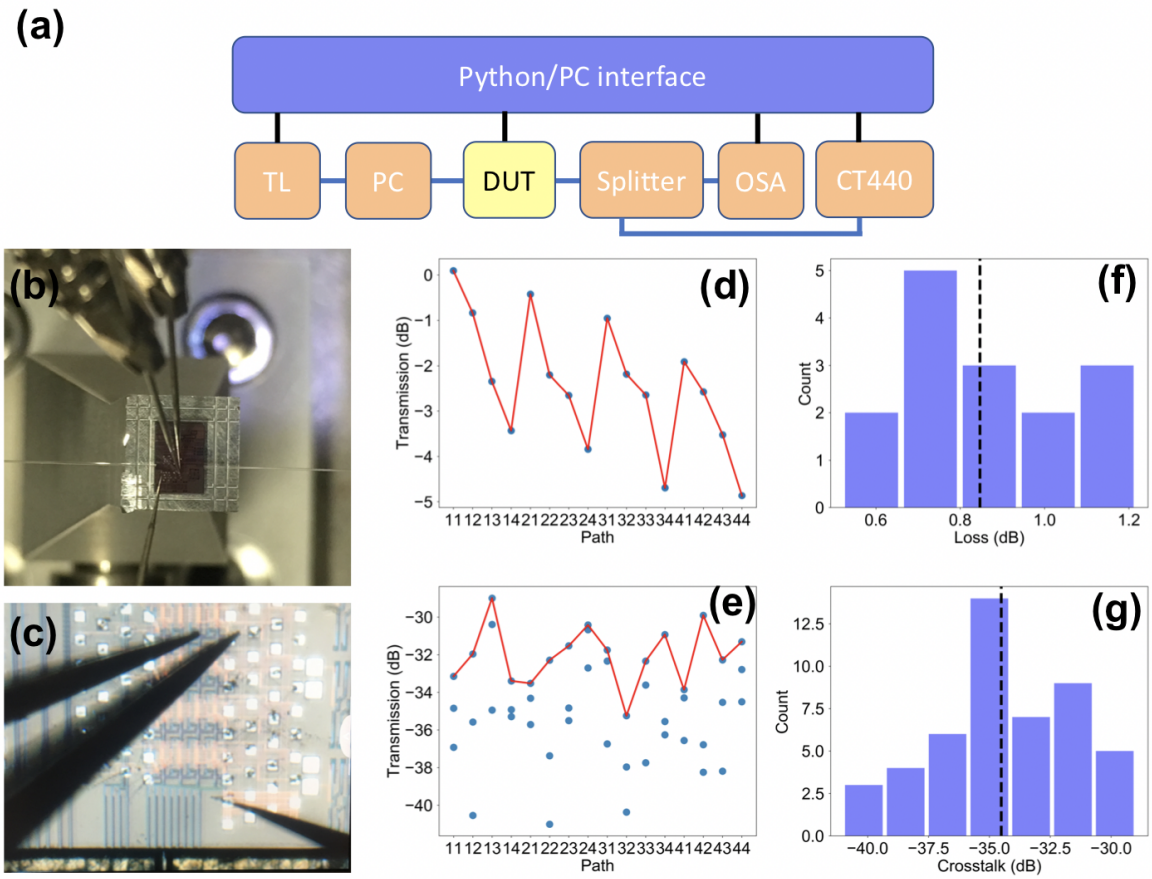


Figure 3.12: (a) schematic of the test setup. TL : Tunable Laser, PC : Polarization Controller, DUT :Device under test, OSA: Optical Spectrum Analyzer, CT440 used for fast wavelength sweep. (b) DUT setup with DC probs and cleaved fibers for optical coupling. (c) an example of path testing with probes on a unit cell and ground array. (d) Path losses corresponding to different connections.13 on the x axis corresponds to input port 1 and output port 3. (e) Off loss or leakage corresponding to different connections. (f) Histogram of off-resonance or pass through loss of the switch. Median pass through loss of 0.85 dB is measured (g) Histogram of Off loss or leakage. Median off loss of -35 dB and worst case -28 dB is measured.

Fig. 3.12 (e), shows the leakage or off loss of the switch. In this measurement, no voltage was applied to all resonators except the resonator corresponding to the signal path. Channel corresponding to wavelength λ is dropped at a given port and power at the same wavelength is measured at other ports. The signal channel is detuned by 400 GHz from its resonant wavelength. The x axis corresponds to a signal path and y axis

corresponds to the off loss minus the coupling loss at the other three output ports. Red line corresponds to worst case off loss. Fig. 3.12 (g) shows a histogram of the off losses which corresponds to a median of -35 dB and worst case off loss of -28 dB.

We also showed data transmission at 120 Gbps aggregate data rate through this switch at 60 Gbaud-PAM4.

3.5 4x4 L=2 switch: cascaded first order

In this subsection, I present a 4x4 switch with L = 2 switch that can tune over one full FSR with a more compact footprint. Ring resonator spectra are very sensitive to gap between waveguides. For the case of higher order ring resonators specific relationship between multiple kappa's must be met for a flattop spectra. An easier approach to achieve a given 3 dB bandwidth and at the same time a higher out of band rejection is to cascade the drop transmission spectra of multiple first-order ring resonators. This results in a higher power penalty as compared to second order ring resonator, but the ease of design makes this one of the choices for the filter in the 2 λ switch.

4x4 switch with L=2 with second-order cascaded first-order rings corresponds to 64 signal pads as switch signal pads and hence area scales as $2LN^2$. The area of this switch can be reduced by half if both the ring resonator heater are connected together,but this places stringent constraints on acceptable resonant wavelength variation of the fabrication process. We measure a switch fabricated in a 220 nm Si Photonics foundry and report the variation of loss, CW off loss, resonant wavelength at zero bias, and FWHM. An improvement fabrication process will enable better scaling of future switches. But we find that, switches cannot be designed by connecting the signals of the two ring resonators to the same pad with the current ring resonator uniformity. We compare two different chips and measure the dissimilarity between two rings by recording the voltage applied

at the two rings.

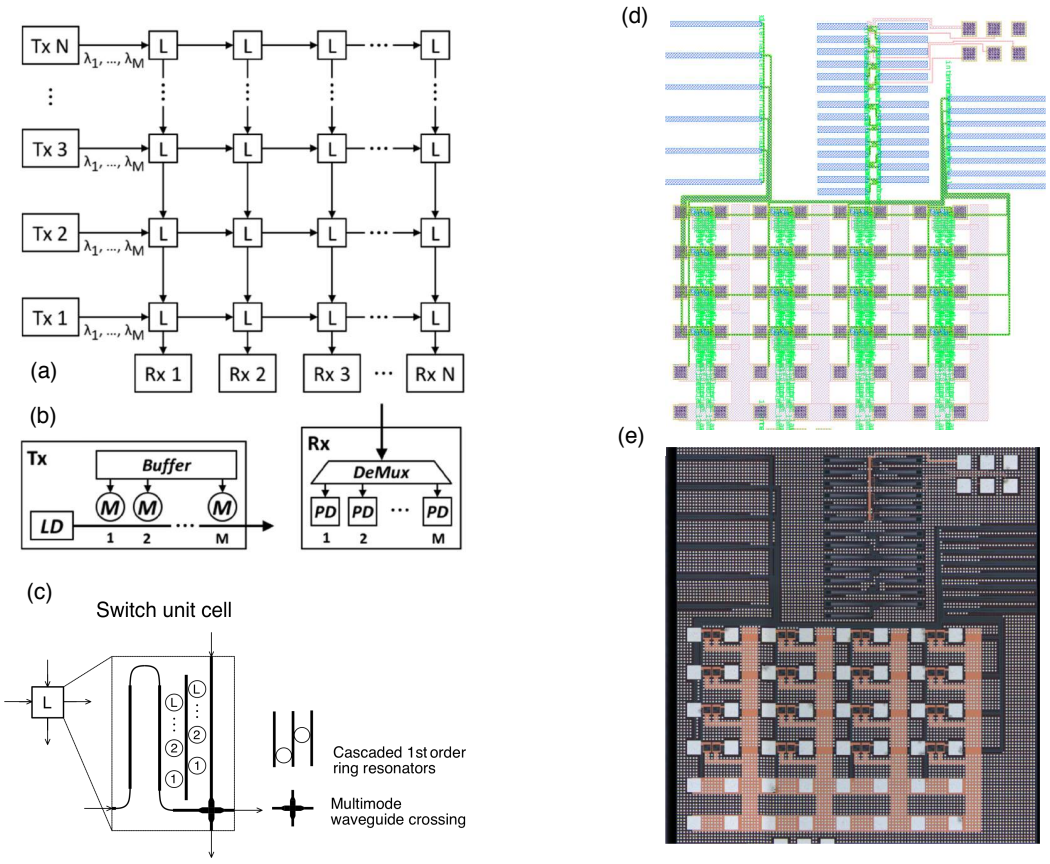


Figure 3.13: (a) NxN switch with L blocks. M wavelengths are input into the switch. (b) Tx: Transmitter, M microring modulators each driving one wavelength, Shared buffer so that data to any port can be modulated on any modulator. (c) Contents on the L block. L cascaded first order microring resonators and waveguide crossing is used.(d) Layout of 4x4 switch and (e) Die shot of 4x4 switch

There are total 32 signal pads and 16 ground pads with a total footprint of 1.6 x 1.55 mm^2. The signals were routed with a 1 $\mu m/mA$ rule to avoid burnout due to electro migration. The layout was done with Cadence Virtuoso and waveguide crossings were designed using Phoenix Optodesigner. Traces of 70 μm and 20 μm are used to connect to the ground pads. Due to low measured resistance of the ring resonators, traces twice the width or connecting all grounds to a ground plane on a different electrical routing

layer might be a better choice for future switches. Pad sizes are 60 μm x 60 μm with a pitch of 160 μm. A pitch of $<$ 135 μm is unsuitable for flip-chip bonding to organic carriers and thus this pitch was chosen even though we did not flip chip this chip in this work. All pads were filled with square vias, as larger number of vias connecting different pad layers prevent pad peel off problem.

The waveguide dimension used in routing is 220 nm x 400 nm and we use 5 μm radius bends for routing all waveguides. Foundry-specified waveguide loss is 2 dB/cm and edge coupler loss is 2.7 dB/facet. 100 μm trench is provided at the chip edge for ease of optical coupling. Waveguide crossings are designed with particle swarm optimization and have a reported average loss of 0.028 dB and worst-case crosstalk of 37 dB [51]. The designed footprint of the crossing is 17 μm x 17 μm due to 4 μm linear tapers used to taper the waveguide from 400 nm to 500 nm. Spirals with simultaneous tapered width from 400 nm to 150 nm and radius from 5 μm to 0.2 μm are used for waveguide termination. These spirals have a 20-dB reflection simulated with Lumerical 3D Finite Difference Time Domain package at 1550 nm.

Figure 3.14 (a) shows Transmission(dB) vs. Paths in the 4x4 switch. Blue dots correspond to measured data points at different wavelengths and red corresponds to median of the points for each path. Figure 3.14 (b) shows a median loss of 5.32 dB. Median is used as a metric instead of mean due to skewed distribution of the data. Median divides the second and third quartile and thus provides a more accurate measure of loss as compared to mean. The outliers in Figure 3.14 (a) and (b) correspond to cascaded filters with a large difference of resonant wavelengths. CW off loss is measured at 1546.8 nm. In Figure 3.14 (c) and (d), crosstalk measurement is done as follows, for a particular path ring is set at 1546.8 nm and leaked power is measured at other ports.

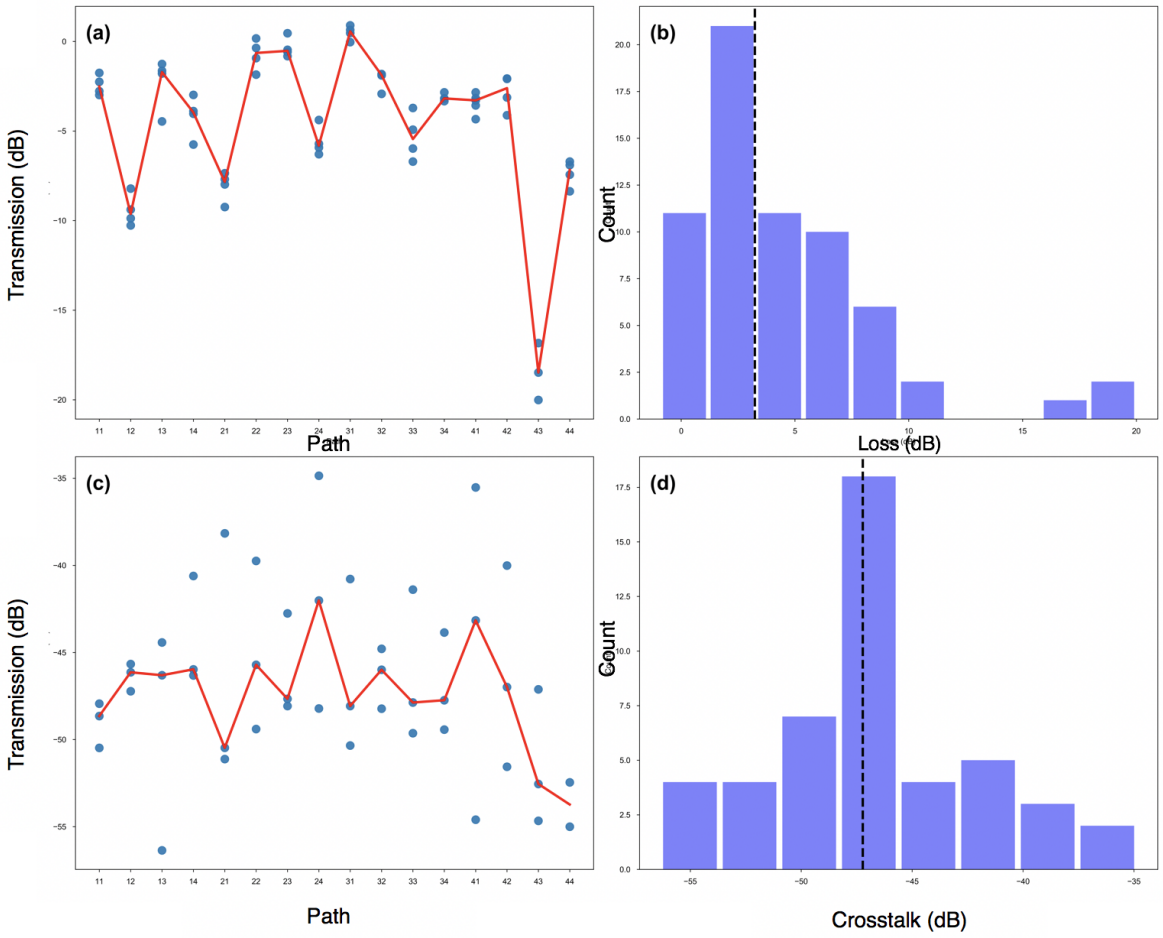


Figure 3.14: (a) Transmission (dB) vs. Paths in the switch. Each data point (blue) corresponds to measurement at a different wavelength. Red line corresponds to line connecting medians. (b) Histogram of Loss for different paths with a median of 5.32 dB. (c) CW off loss is measured at 1546.8 nm. This is the signal channel. (d) Median measured crosstalk is -47.16 dB

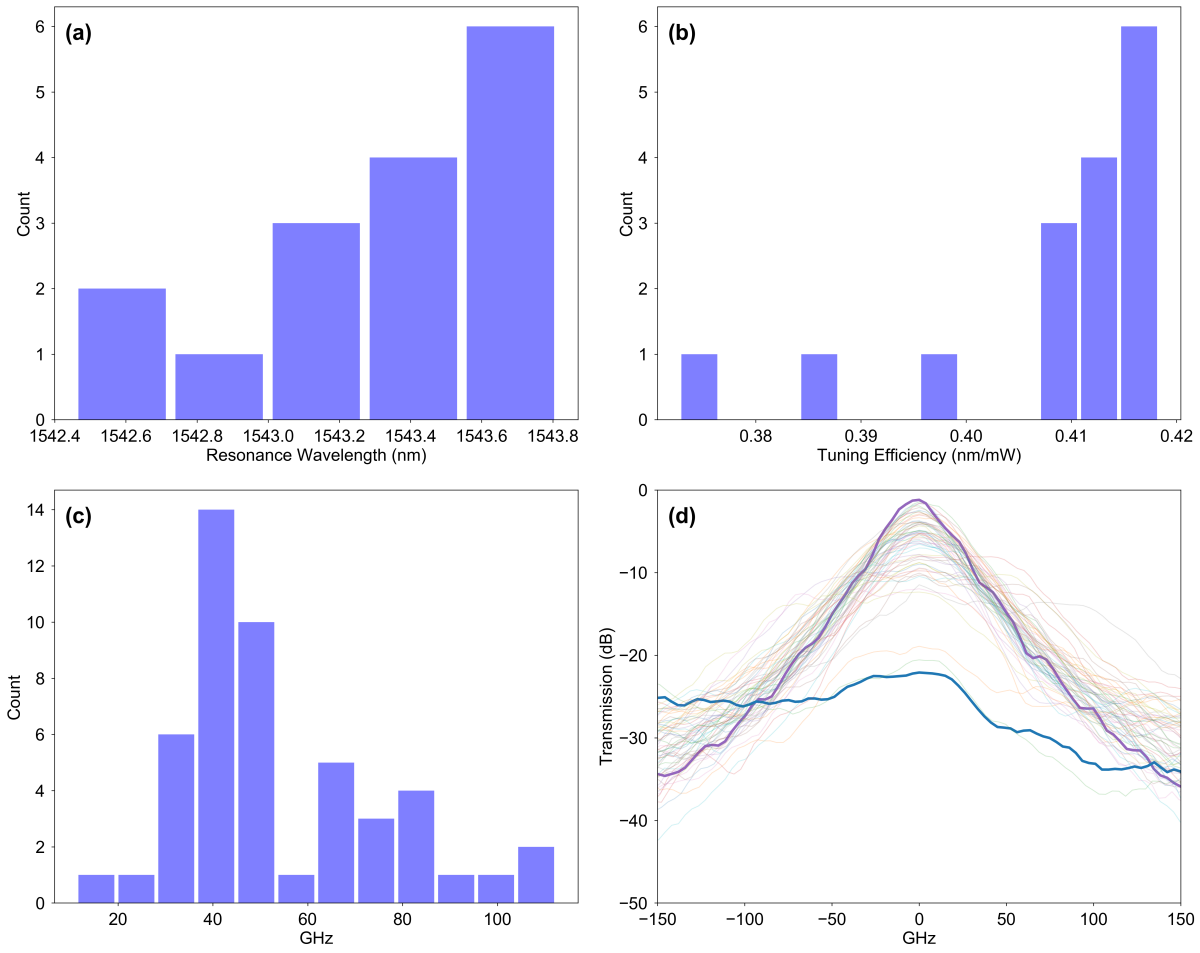


Figure 3.15: (a) Histogram of resonant wavelength, mean = 1543.35 and standard deviation = 0.39 nm. (b) Tuning efficiency mean = 0.4083 nm/mW and standard deviation =0.012 nm/mW (c) Bandwidth histogram with median = 48.69 GHz (d) Transfer spectra of all ring resonator filters

Figure 3.14 (a) shows the histogram of resonant wavelengths with mean of 1543.35 nm and standard deviation of 0.39 nm. This is similar to the measured standard deviation of 0.33 nm for second order resonators. Figure 3.14 (b) shows the histogram of tuning efficiency of 0.012 nm/mW. Figure 3.14 (c) shows bandwidth histogram with median bandwidth of 48.69 GHz. Figure 3.14 (d) shows all measured transfer spectra. One can see the outlier spectra with ring resonators with dissimilar resonant wavelengths.

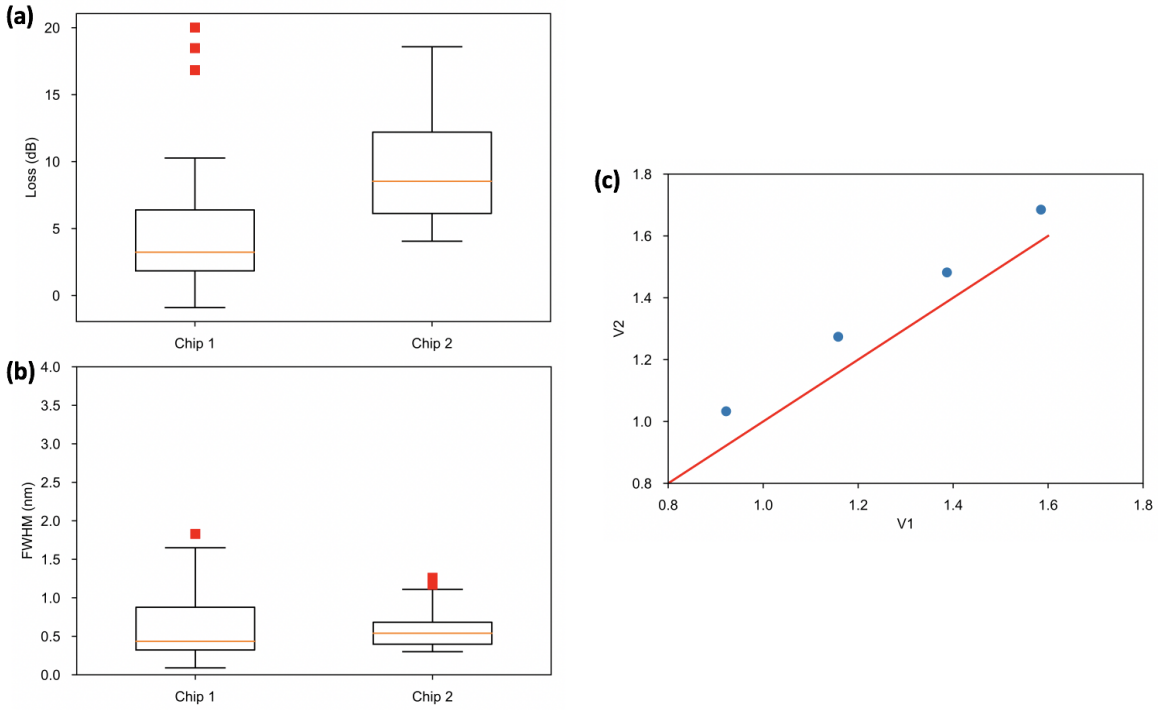


Figure 3.16: (a) Path loss comparison for chip 1 and chip 2. Red squares correspond to outliers and orange lines correspond to the median. (b) Full Width Half Maximum (FWHM) of different switches measured on chip 1 and chip 2. (c) A plot showing dissimilarity of ring 1 and ring 2. Red line corresponds to a 45-degree line where V1 = V2.

Fig. 3.16 (a) shows a comparison of path loss on two different chips. The orange lines correspond to the median loss. One can see that the median loss of chip 1 is 3.23 dB which is smaller than 8.53 dB of chip 2. The distribution of path losses of chip 2 is skewed towards lower values. Fig. 3.16 (b) shows the FWHM of chip1 and chip 2. The median FWHM of chip 1 is 0.435 nm which is smaller than 0.54 nm of chip 2. The IQR of chip 1 is 0.56 nm which is much greater than 0.29 nm of chip 2. This shows that there is a smaller variation in FWHM values. Fig. 3.16 (c) shows V1 and V2 applied to two rings on a test structure cascaded in the same arrangement as the rings on the switch. These voltages are optimized for maximum peak transmission of a given channel wavelength. Red line shows $V_1 = V_2$. The dissimilarity of the drive voltages could be due to two

factors, variation in heater resistances or variation in the fabrication process (thickness of the chip). As the rings are placed right next to each other it is unlikely that thickness can be factor. Different doping of resistors or other fabrication variation that can change dimensions of the resistors can change the effective index of the ring resonator. This could be the reason for different resonant wavelength and heater resistances.

3.6 Previous Generation switches

Histograms of previous generation of switches

Datacom switches were developed in four different AIM runs, Run1 through 4. We follow the following definition:

- 8x8 switch Run 1: Gen 1
- 8x8 switch Run 2: Gen 2
- 8x4 switch Run 3: Gen 3
- 4x4 switch Run 3: Gen 4
- 4x4 switch Run 4: Gen 5

Gen I and Gen II 8x8 switches were the first runs from the new AIM foundry, and showed a high pass through loss for unit cells of 2.8 dB and 4 dB respectively. This was due to high doping of the microring resonators. The results of multiple measurements measured on different chips are reported in Fig. 3.17.

Fig. 3.17 (a),(b) shows that the through and drop losses of the unit cell increased from Gen 1 to Gen 2. Through losses increased from 2.82 $\pm$ 0.26 dB to 4.72 $\pm$ 1.6 dB as compared to the expected value of 1.25 dB. Drop losses increased from 5.95 $\pm$ 0.36 dB

to 7.63 $\pm$ 0.77 dB as compared to expected loss of 2.05 dB. These values moved closer to the expected values in the later iterations.

Fig. 3.17 (c) and (d) show the edge coupler per facet loss and resonant wavelength variation measured in the two runs. Edge coupler losses are 3.25 $\pm$ 0.55 dB and 3.27 $\pm$ 0.18 dB in the two runs as compared with expected loss of 2.7 dB. Standard deviation of resonant wavelength in the two runs with zero bias is 2.48 nm and 0.22 nm. Resonant wavelength was measured by calculating the extrema in the peaks of the unbiased ring resonator. This method was later replaced by extracting the resonant wavelength and tuning efficiency from the spectra by fitting a tuning curve. We observed an inter chip resonant wavelength standard deviation of 0.33 nm.

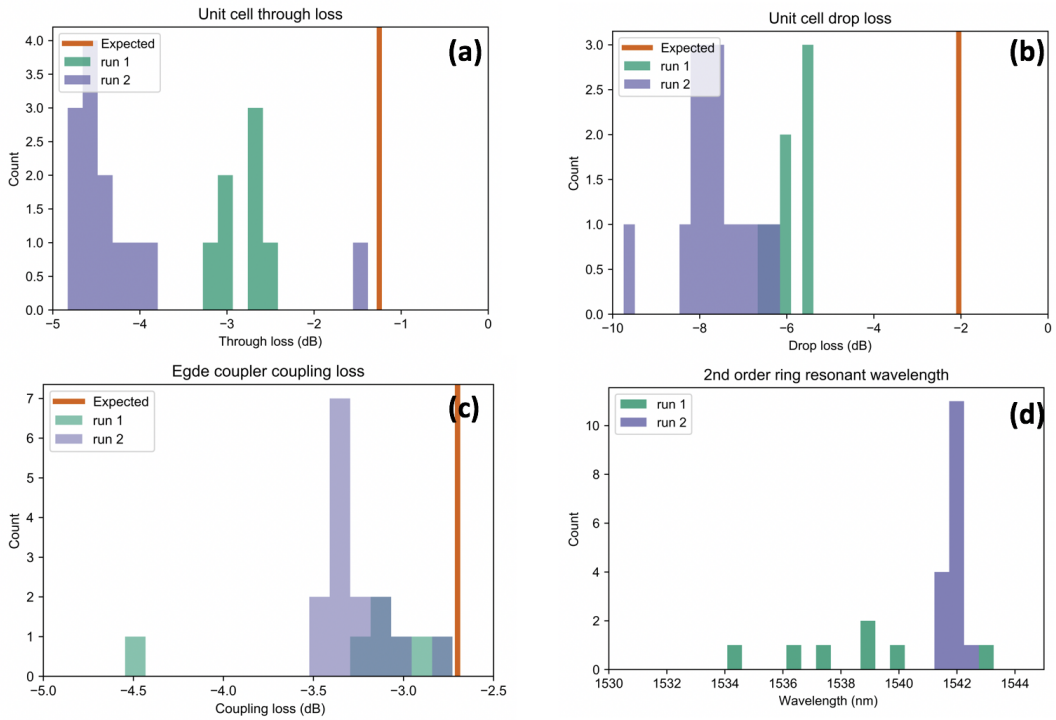


Figure 3.17: Histograms of (a) Unit cell through loss, (b) Unit cell drop loss, (c) Edge coupler coupling loss per facet and (d) second order rind resonator resonant wavelength at zero bias. Orange line corresponds to the expected value.

	Drop loss (dB) (mean/std)	Off-resonance through loss (dB)(mean/std)	Edge coupler loss/facet (dB) (mean/std)	Resonant wavelength (nm) (mean/std)
Generation I (Run 1)	5.94/0.35	2.81/0.25	3.25/0.54	1538.38/2.49
Generation II (Run 2)	7.63/0.77	4.71/1.6	3.27/0.18	1541.89/0.23

Table 3.1: Table summarizing histogram results from Fig. 3.17

Comparison of resistances

The measured mean of the ring resonator heaters remained the same while standard deviation of the heaters increased by a factor of three from 0.057 $k\Omega$ to 0.164 $k\Omega$.

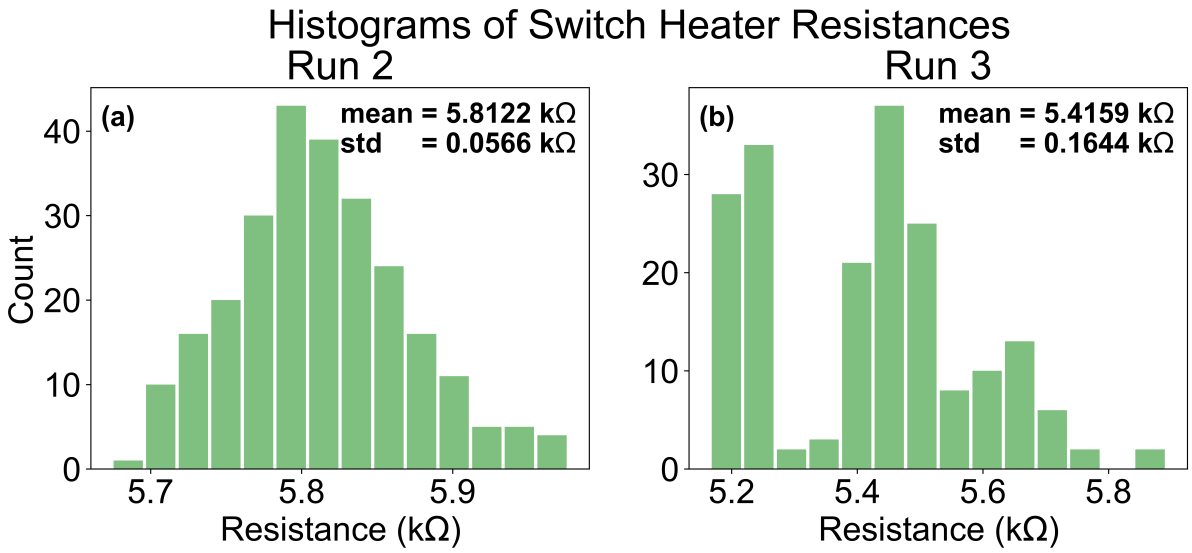


Figure 3.18: Histograms of (a)256 ring heater resistance measurements on run 2 switches and (b) 190 measurements on run 3 switches

Electronic Photonic integration

One of the hurdles while deploying such switches in applications is packaging. Wirebonding is not a scalable packaging approach when the switch size increases beyond 92 pads if a standard Printed circuit Board (PCB) is used with 5 mil/5 mil trace and space. The bottleneck is the pad pitch and pad size on the PCB. There are multiple ways around

this problem some of which are mentioned in Fig. 3.19. Three methods are explored and the advantages and disadvantages are given in Fig. 3.19.

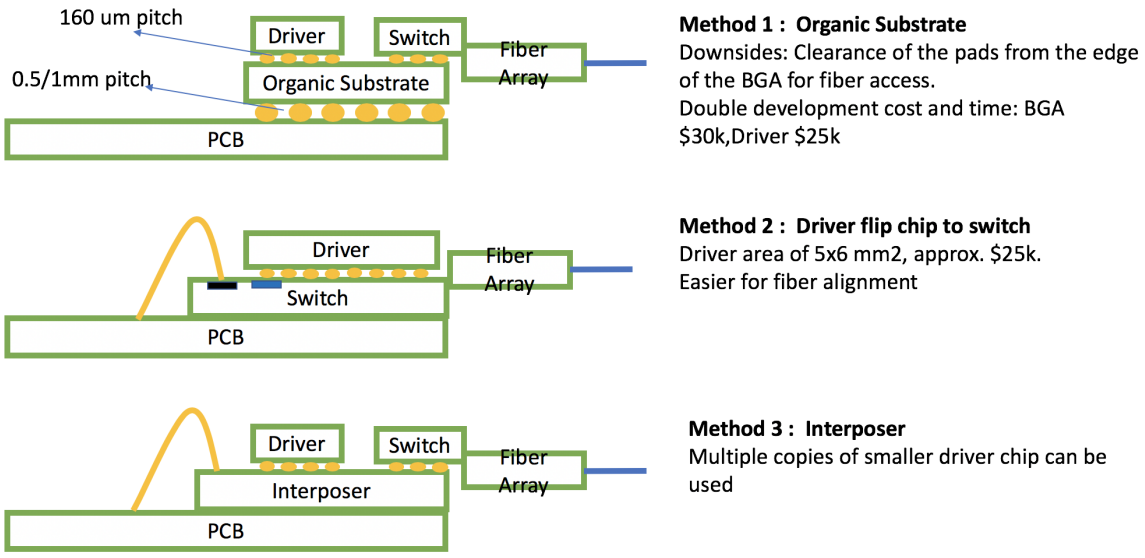
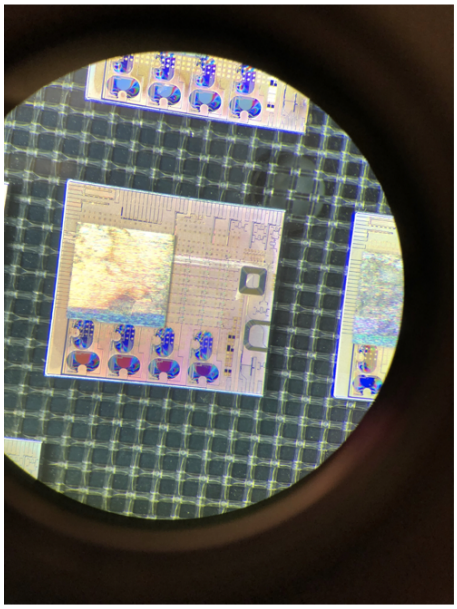
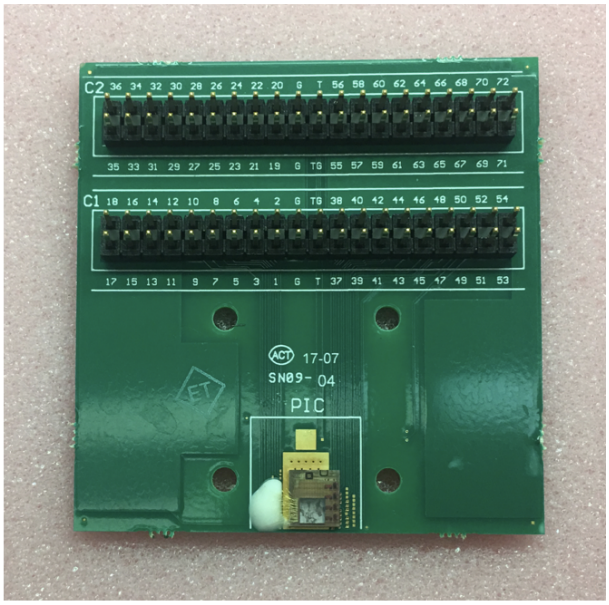


Figure 3.19: Three approaches to packaging. In Method 1: Organic Chip Substrate package with Ball Grid Arrays (BGA) or Land Grid Arrays (LGA). Method 2: Driver flip chip to switch and Method 3: Flip chip to interposer

We explored Method 2 in a collaboration with UC Berkley. The electronic driver was fabricated in ST Microelectronics process. The driver could supply upto 8 V with 3 mA current. The driver could supply up to 1.5 mA of current with 4 V bias through one of its unit cells, but unfortunately a connection to the switch did not work. The chip was packaged with multiple vendors and also wirebonding the switch chip was tried at UCSB. The pads peeled of the photonic switch. One of the reasons could be that the photonic chip pad was very thin and could not be wirebonded. Fig. 3.20 shows the images of the package. The ring resonators fabricated in the AIM process custom and PDK both require more than 30 mW to tune by full FSR.



Flip chip bonded die



Package with wire-bonds to PCB

Figure 3.20: (a) Flip chip bonded die, (b) Packaged with wire-bonds to PCB

3.7 Summary and Conclusion

In this chapter, we presented experimental results of 8x4,8x4 and 4x4 switches. We demonstrated multiwavelength switching and characterized the switches. The switches improved on the loss of previous generation and the FSR improved over generations to more than one FSR in the last generation.

Chapter 4

On chip wavelength locking

4.1 Introduction

In this chapter, we present an on-chip wavelength reference with a partial drop ring resonator and germanium photodetector. This approach can be used in ring-resonator-based wavelength-selective switches where absolute wavelength alignment is required. We use the temperature dependence of heater resistance as a temperature sensor. Additionally, we discuss locking speed, statistical variation of heater resistances, and tuning speed of the switches. This work is published in [5, 58].

4.2 Procedure

We proposed a flexible ring-resonator based architecture in [3], where any set of multiple wavelengths could be switched between input-output pairs. In [3] we also established that two rings per crosspoint are sufficient for a ring-based crossbar switch where number of wavelengths per port are equal to number of ports. Such micro-ring based switches that can establish a connection with any one or a set of multiple wavelengths from a WDM

signal require robust wavelength locking scheme. The locking scheme in this problem is different from other approaches [59, 60, 61, 62, 63], where a ring resonator or a bank of resonators can lock to a particular or all wavelengths in a WDM signal, but one cannot be certain to which wavelength a particular ring is locked without an external optical spectrum analyzer (OSA). For this we need an absolute wavelength reference on chip to lock any switch node to a given wavelength of choice. In this paper we demonstrate such an approach which requires only an approximate tuning curve of the ring resonators, does not require an external monitor signal for wavelength identification and can lock rings at different stage temperatures. Preliminary results from this configuration were presented in [58]. In this paper we present locking time, statistical variation of ring resonators and detailed discussions regarding the performance of the switch.

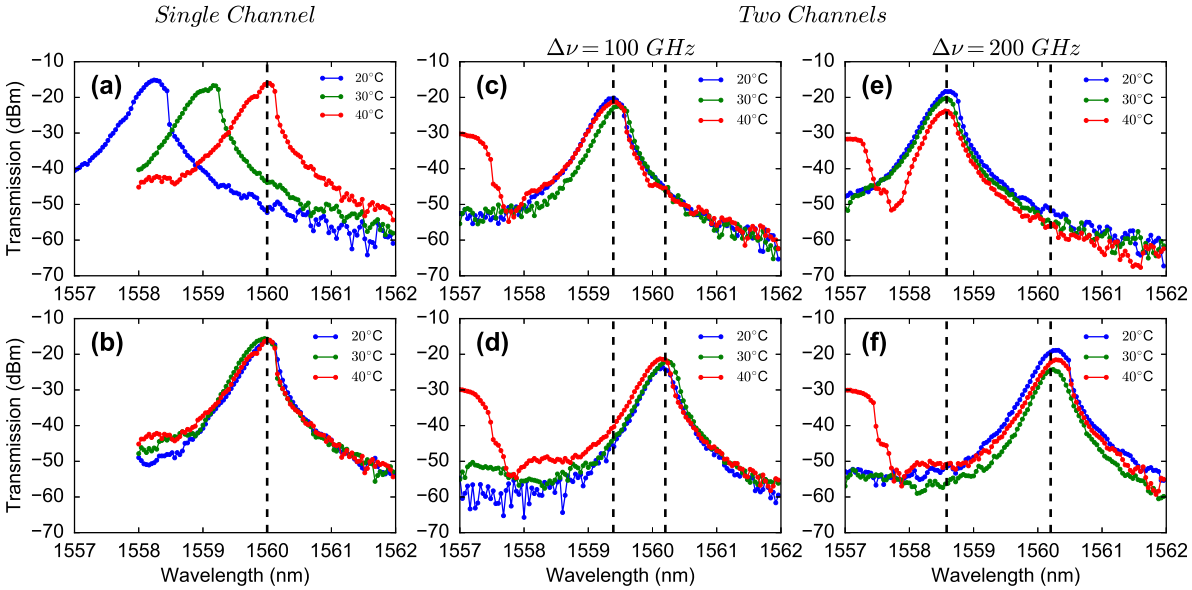


Figure 4.2: Transfer function of second order resonator after locking to predefined wavelengths at different stage temperatures.Wavelengths to which ring is locked is given as a black dashed line.**(a)** and **(b)** show results before and after locking in the presence of input single wavelength. **(c)** and **(d)** show results after locking to two channels separated by $\Delta\nu = 100\ GHz$. **(e)** and **(f)** show results for $\Delta\nu = 200\ GHz$. Reprinted from [5] © The Optical Society.

Fig. 4.1(a) shows a schematic of the experimental setup of the wavelength locking

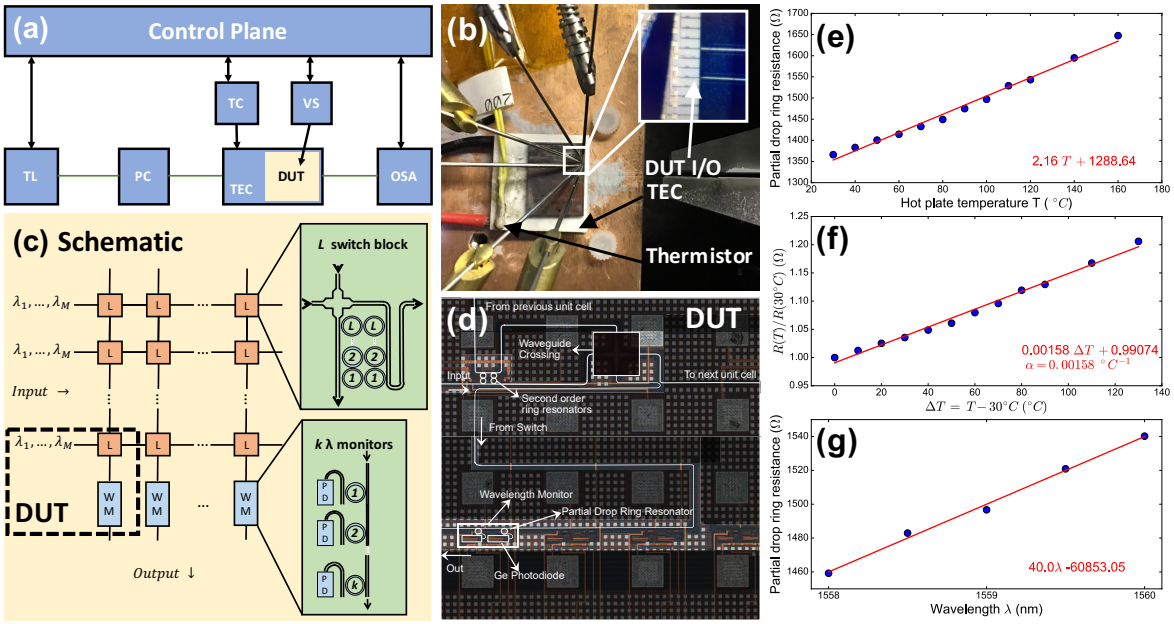


Figure 4.1: **(a)** Schematic of the setup. **TL** : Tunable Laser, **PC** : Polarization Controller, **DUT** :Device Under Test, **TEC** : Thermo Electric Cooler , **TC** : Temperature Controller , **OSA** : Optical Spectrum Analyzer, **VS** : Voltage Source **(b)** Test setup of the experiment. **(c)** General schematic of wavelength-selective switch shown has L second order ring switches and a waveguide crossing at each intersection and k Wavelength Monitors (**WM**) at the bottom of each column which have an on-chip photodiode (**PD**).The DUT is marked with a dashed box. **(d)**Micrograph of the switch. **(e)** Partial drop ring heater resistance vs. hot plate temperature. **(f)** Verification of Eq . 4.1. **(g)** Tuning map, partial drop ring heater resistance vs. locked wavelength at $20°C$[5]

experiment. A tunable laser (Yenista Tunics T100S-HP) is used to inject a wavelength at 1557 nm with TE polarization into the device. We control the polarization of light with a Newport Polarization Controller (PC: model F-POL-APC). The chip is mounted on top of a TEC, which we use to change the stage temperature with the help of a thermistor and ILX Temperature controller(model LDT-5980). Fig. 4.1(b) shows the experimental setup with TEC, thermistor, electrical probes, and input and outputs (I/O) of the DUT.

Fig. 4.1(c) shows the schematic of the architecture of the switch. Each switch matrix unit cell has L second order ring resonators that can select up to L wavelengths from incoming WDM channels and a waveguide crossing. The free spectral range (FSR) of the

ring resonators should be designed to fit all the channels, and the rings have to be tuned by thermal tuning as tuning across all channels is required. For example, for 8 WDM channels at 200 GHz channel spacing, an FSR of at least 1.6 THz (12.8 nm) is required. In our experiment, we used an FSR of 26 nm. The synchronous time slotted operation of the switch with centralized arbiter implies an input queued switch architecture [3]. The device under test (DUT) is shown by the dashed box in Fig. 4.1 (c) and by the micrograph of Fig. 4.1 (d). Our locking experiment was conducted by tuning only one of the two second order ring resonators. The optical output is recorded by an OSA (Yokogawa AQ6370).

These devices were a part of the initial release of the AIM Photonics Process Design Kit (PDK). The switch ring resonators were equipped with thermo-optic and electro-optic tuners. In our experiments we used thermal tuning due to a higher tuning efficiency of 0.84 nm/mW and negligible loss penalty as compared to electro optic tuning. For electro optic tuning, the resonance wavelength was blue-shifted and attenuated with a loss of 3.6 dB/nm. The 3-dB bandwidth of the resonators was 40 GHz and, with better than 30 dB of out of band rejection at the drop port. The measured insertion loss of each second order ring was measured at 1.2 dB for wavelengths off resonance at the through port and 3.5 dB on resonance at the drop port. Cleaved single mode fibers are used for optical coupling. The light is edge-coupled into the chip with a Si_3N_4/Si mode converter with a typical coupling loss of 3 dB in the C band.

The WM consists of a thermally tunable partial drop ring resonator that is designed to drop 10 % of the output power at a germanium (Ge) photodiode. These resonators were provided by the foundry PDK and their design was similar to [64]. If these partial drop resonators are tuned to the correct wavelength, the second-order switch can be tuned to maximize the photodiode current captured by the Ge photodiode. However, a change in the ambient temperature changes the resonant wavelength of the WM. This

problem can be circumvented by using the resistance of the heater in the partial drop filter as a temperature sensor. The resistance of the partial drop resonator heater is measured as a function of hot plate temperature T from 30°C to 140°C at a bias voltage of 0.1 V as given in Fig. 4.1 (e) and the linear relation given by the Eq.4.1 is verified with a linear polynomial fit in Fig. 4.1 (f).

$$R(T) = R(30°C)(1 + \alpha(T - 30°C)) \tag{4.1}$$

where T is the resistor temperature in °C and α is the temperature coefficient of resistance of $1.58 \cdot 10^{-3}$ °C^{-1} from Fig. 4.1 (f). This fit demonstrates that resistance can be used as a temperature sensor. At a stage temperature of 20°C, voltages on second order ring heaters (V_1, V_2) are optimized with OSA, then we choose the voltage on partial drop resonator(V_{WM}) corresponding to highest photodetector current. The value of resistance at this V_{WM} vs. optimized wavelength λ gives the tuning map of $R(\lambda)$ measured at stage temperature of 20°C given in Fig. 4.1(g). In Algorithm 1, we describe the locking procedure. Here the number of incoming WDM channels is assumed known and step 4 is useful for locking in the presence of multiple channels.

Algorithm 1 On-chip locking algorithm

1: **procedure** LOCK(V_1, V_2) ▷ for a given λ, $R(\lambda)$
2: At $T = T_{stage}$ and λ set V_{WM} corresponding to $R(\lambda)$
3: Coarse sweep in (V_1, V_2) with $V_1 = V_2$, and record PD current (I_{PD})
4: Pick a guess voltage $V_{guess} = V_1 = V_2$ corresponding to peak in I_{PD} corresponding to channel number
5: optimize (V_1, V_2) to maximize I_{PD}.

Next we present our results on locking in the presence of single channel and two channels spaced at $\Delta\nu = 100,\ 200GHz$. Fig. 4.2(a) shows the temperature dependence of the resonant wavelength. The resonant wavelength of the ring changes by 80 pm/K. In this figure, the voltages on the switch heaters were optimized for highest power transmission

at 40 °C and then stage temperature was changed to 20 °C and 30 °C with the heaters set to voltages corresponding to 40 °C. We then use our wavelength locking scheme to change the switch heater voltages with the help of the WM and lock wavelengths within 20 pm of the target wavelength. The tunable laser used for measurement had a 20 pm accuracy. The results can be seen in Fig. 4.2(b).The transfer function of the drop port spectra of the second order ring is recorded here after the locking.

Fig. 4.2(c) and (d) shows before and after locking results in the presence of two channels spaced at $\Delta\nu = 100\ GHz$. Fig. 4.2(e) and (f), shows before and after locking in the presence of two channels spaced at $\Delta\nu = 200\ GHz$. In these experiments system loss was measured at 20 dB and input power on both wavelengths was 10 dBm. In transfer function corresponding to 40°C in Fig. 4.2(c),(d),(e) and (f) the peak corresponding to the other second order switch in the unit cell with unoptimized heater voltages is also visible. The optimization is run with stochastic hill climbing (SHC)[56] with a noise standard deviation of 50 mV to limit search space around V_{guess}.

Fig. 4.3 (a) shows the speed of different algorithms used to optimize the heater voltages applied to the second-order switch ring. We compare three different optimization methods, namely the Nelder-Mead [65], the SHC and Powell's algorithms [66]. SHC, in this case Random Mutation Hill Climbing, is a type of genetic algorithm where we apply stochastic noise to the voltages and update the best guess to the heater voltages depending on the fitness criteria (transmission). Because of the stochastic nature of the search, the locking time is not deterministic and this is shown by the multiple blue trajectories in the figure. Both Nelder-Mead and Powell are derivative-free algorithms which give deterministic trajectories and locking times. These were implemented with the minimization module of SciPy library of Python. The top axis of locking time is generated assuming each iteration takes 7 μs, which is the measured thermal time constant. The heuristic algorithms do not terminate on a non-optimal local maxima as the second-order

switch transmission is a unimodal function of heater voltages, *i.e.* there exists only one maximum. These heuristic algorithms are also accurate and faster than a grid search over the two heater voltages of the second-order resonators since the function is unimodal.

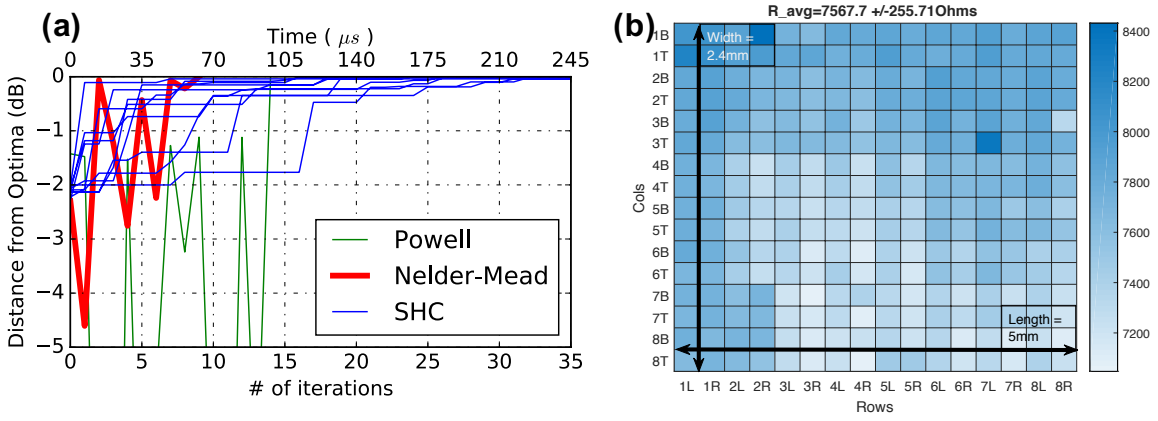


Figure 4.3: **(a)**Locking speed of different algorithms, with Nelder-Mead being the fastest locking algorithm. Here the distance from optima is the separation from the terminal optimized transmission of the second order ring resonator.**(b)** Measured variation of second order ring heater resistance across switch chip at stage temperature of $20°C$, with an average value of 7568 Ω, and a standard deviation of 256 Ω . Reprinted from [5] © The Optical Society.

The main reason why we pursued this approach was that there was an 8 % standard deviation in heater resistance across the switch chip as shown in Fig. 4.3 (b). A total of 256 heater resistances from 128 second order rings were used to measure this standard deviation. The size of the switch die across which measurements are done is 5 mm x 2.4 mm. This means that one cannot generate a tuning curve for a single resonator and use that to lock other ring resonators as each resonator will have a different tuning curve.

In Fig. 4.2, we were limited by the temperature range for which our approach of absolute wavelength locking worked from 20-40 °C. This was due to the limited tuning of the ring resonators fabricated. Though the resonators were designed to be fully tunable across an FSR of 26 nm, we saw performance degradation of the heater after tuning by ∼8 nm. Further investigation of the tuning limits of these structures is needed, but

tuning across this fraction of the FSR still allows for useful switching operation.

4.3 Discussion

An advantage of the wavelength locking scheme we described above is the ability to be resistant to thermal crosstalk across the switch. The wavelength monitors can be placed away from the switch rings and multiple ring resonators can be optimized simultaneously with two or more wavelength monitors in a column. For example, if two ring resonators are placed close to each other, their voltages can be optimized simultaneously with WM as reference. These voltages will self-adjust according to the thermal crosstalk penalty to maximize drop power at a particular wavelength.

Fortunately, it was not necessary to compensate for thermal crosstalk, as the observed thermal crosstalk was small. In most cases, the layout of the switch is pad limited and if the pads are laid out in a flip-chip configuration, for example in a square grid at 150 μm pitch, the switch rings can be placed far apart to avoid significant thermal crosstalk. This is because the footprint of a ring resonator is 20 μm x 10 μm, which is small compared to a 150-μm pitch. To scale such switches, optical interposers might be used in the future and the rings can be placed closer together for tighter packing and this can increase the thermal crosstalk. The motivation behind placing k WM per column in Fig. 4.1 (c) is that speed of locking can be parallelized. We placed identical partial drop rings in our layout. If the number of channels is equal to the number of WM then partial drop rings with different resonant wavelengths can be used.

4.4 Summary and Conclusion

To conclude, we present an approach to bias and stabilize ring resonator-based

wavelength-selective switches where absolute wavelength alignment is required. We use the temperature dependence of heater resistance as a temperature sensor to stabilize the ring switching wavelengths. The scheme can lock rings and distinguish between wavelengths when wavelength channels are separated by more than 100 GHz. Lower frequency spacing between WDM channels need a higher out of band rejection for the wavelength monitor ring resonator so that the locking scheme does not lock to the wrong channel due to uneven powers in the WDM channels.

Chapter 5

Summary and Directions for future research

5.1 Summary

In this thesis, a novel multiwavelength selective crossbar switch was proposed, analyzed and demonstrated. The latency simulations showed that just two wavelengths per crosspoint are sufficient, thus reducing footprint and power consumption. An arbitration scheme for the switch was also proposed. Five generations of switches were built and characterized. Experimental characterization of loss and crosstalk was conducted and path loss was reduced while FSR was improved over generations. A novel wavelength locking scheme was also demonstrated. The scheme was able to identify different wavelength at a channel separation of $>$ 100 GHz and lock a ring resonator to it.

5.2 Directions for future research

1. A major bottleneck in our design of optical switches was the specifications of the ring resonator. Ring resonators with low loss ~ 0.001 dB, one full FSR tuning (>20 nm or enough to fit more than 16 channels in one FSR), flat top filter with high out of band rejection, $\sim 10\mu m \times 10\mu m$ footprint, $<$ 1 mW/FSR power consumption and 1 ns switching speed will help scale an optical switch to hundreds of ports. A ring resonator with these specifications is not demonstrated yet. A ring resonator with 4.4 $\mu W/GHz$ and >30 nm FSR has been demonstrated in [57] with 1 μs switching time constant extracted by fitting to exponential decay. Ring resontor filter with lower power consumption with substrate undercut is also demonstrated in [67]. This resonator exhibited a switching time >100 μs due to the undercut but at the same time the power consumption 2.4 mW/FSR which is lower by a factor of 10 as compared to [57]. A novel device design that has the same transfer matrix as a ring resonator, or a different fabrication platform might be required to accomplish this.

2. An analysis was conducted for a hybrid integration of optical switches with electronic switches [2]. In this study MEMS based switches were used, and data rates of 20 Gbps were used in 2011. Electronic switches now can handle Tbps throughput with 400 ns latency. How do optical switches fit in with these switches? Can data center applications speed up, or are there power savings to be achieved when an entire data-center network with control plane and optical circuit switch is implemented? What fraction of the total power does interconnection consume? Is improvement in cooling and better electronic switches be a better path than improving and including optical switches in datacenters?

3. To scale optical switches to greater port counts and better electronic and photonic integration, a better packaging method for optical switches is required. Direct flip chip bonding make sense for DC (low frequency electronic) applications and for the sizes of the switches that can be built with the loss constraint, wire-bonding with a package is sufficient. Tight integration of electronic and photonics currently cannot be justified as photonic switches are pad limited and fabricating large electronic drivers compatible with flip chip to organic package, interposer or directly flip chip is expensive. To scale to higher port counts, photonic interposer and co-fabrication of both photonic devices and electronic devices is demonstrated. These approaches are expensive and due to the size mismatch between and photonics do not make economic sense. Is there a cheaper approach for electronic photonic integration? Or is a pcb driver board assembled with discrete components should suffice as there is very little benefit derived from tight integration of electronic and photonics?

Appendix A

Code for latency simulations

A.1 Mathematica code

The following code is written in Mathematica [68]. There are four parts to this program.

1. Matgen: Generates a NxN matrix with random arrivals of packets for each input output pair.

2. lbuff: allows only l items to pass through for each input output pair. Requests that do not satisfy the above constraint are placed in buffer.

3. rowbuff: checks if the row sum is $\leq$ M, if not places newest items in buffer

4. reconsider: checks if extra requests can be transmitted

Some of the computation was run on a multi-core machine on the Knot Cluster at Center for Scientific Computing UC Santa Barbara.

matgen

```
matgen[iter_, R_, m_, l_, lambda_, k_] :=
 Module[{inpmat, buffer, timetrk, timebuff, x, p = lambda/k},
  inpmat = ConstantArray[0, {R, R}];
  buffer = ConstantArray[0, {R, R}];
  timetrk = ConstantArray[{}, {R, R}];
  timebuff = ConstantArray[{}, {R, R}];
  Do[
   x = RandomChoice[{1 - p, p} -> {0, 1}, k];
   (*implementing l buffer*)
   If[Total[x] > l,
    inpmat[[i]][[j]] = l;
    buffer[[i]][[j]] = Total[x] - l;
    timetrk[[i]][[j]] =
     Flatten[(iter - 1)*k + Take[Position[x, 1], {1, l}]];
    timebuff[[i]][[j]] =
     Flatten[(iter - 1)*k +
       Take[Position[x, 1], -(Length[Position[x, 1]] - l)]]
    ,
    inpmat[[i]][[j]] = Total[x];
    timetrk[[i]][[j]] = Flatten[(iter - 1)*k + Position[x, 1]]
    ]
   ,
   {i, R},
   {j, R}
   ];
  {inpmat, timetrk, buffer, timebuff}
  ]
(*here x is previous iteration and y is matrix in current iteration*)
```

lbuff

```
lbuff[x_, y_, iter_, R_, m_, l_, lambda_, k_] := Module[
  {buffer = x[[3]], buffertime = x[[4]],
   tempinpmat = y[[1]],  tempbuffer = y[[3]] , tempinptime = y[[2]] ,
   tempbuffertime = y[[4]],
   temp1, temp2, z},
  temp1 = tempinpmat + buffer;
  temp2 =
   Table[Join[buffertime[[i]][[j]], tempinptime[[i]][[j]]], {i,
     R}, {j, R}];
  Do[
   z = temp2[[i]][[j]];
   If[Length[z] > l,
    buffertime[[i]][[j]] =
     Take[z, -(Length[z] - l)];(*buffertime changed*)

    temp2[[i]][[j]] = Take[z, l]; (* new tempinptime changed*)

    temp1[[i]][[j]] = l;(* new tempinpmat changed*)

    buffer[[i]][[j]] =
     Length[buffertime[[i]][[j]]](*buffer changed*)
    ,
    buffer[[i]][[j]] = 0;
    buffertime[[i]][[j]] = {}
    ],
   {i, R},
   {j, R}
   ];
  buffer = buffer + tempbuffer;
  buffertime =
```

```
 Table[Join[buffertime[[i]][[j]], tempbuffertime[[i]][[j]]], {i,
   R}, {j, R}];
{temp1, temp2, buffer, buffertime}
]
```

rowbuff

```
rowbuff[z_, iter_, R_, m_, l_, lambda_, k_] :=
 Module[{temp1 = z[[1]],
   temp2 = z[[2]],
   temp3 = z[[3]],
   temp4 = z[[4]], count, totalchk, totalsum, t2, t4, t5, xpos,
   tpos},
  totalchk = (# > m ) & /@ Total[temp1, {2}];
  totalsum = (# - m ) & /@ Total[temp1, {2}];
  Do[
   count = totalsum[[i]];
   (*Print[count];*)

   t2 = temp2[[i]];(*transferring value to vaiable*)

   t4 = temp4[[i]];
   t5 = {};
   While[
    count != 0,
    tpos = Position[t2, Max[t2]];
    If[Length[tpos] > count, tpos = Take[RandomSample[tpos], count]];
    AppendTo[t5, {tpos[[#]][[1]], Max[t2]} & /@ Range[Length@tpos]];
    t2 = Delete[t2, tpos];
    count -= Length[tpos]
    ];
   temp2[[i]] =
    t2;(*tempinptime updated and transferring back from variable*)

   t5 = Reverse[Flatten[t5, 1]];
   Do[
    xpos = t5[[j]];
```

```
  t4[[xpos[[1]]]] =
   Sort[Append[t4[[xpos[[1]]]], xpos[[2]]]];(*buffertime updated*)


     temp3[[i]][[xpos[[1]]]] += 1;(*buffer updated*)


  temp1[[i]][[xpos[[1]]]] -= 1,(*tempinpmat updated*)
  {j,
   Length[t5]}
  ];
 temp4[[i]] = t4
 ,
 {i, Flatten[Position[totalchk, True]]}
 ];
{temp1, temp2, temp3, temp4}
]
```

reconsider

```
reconsider[y_, m_, l_] :=
 Module[{temp1 = y[[1]], temp2 = y[[2]], temp3 = y[[3]],
   temp4 = y[[4]], rowchk, colchk, flag, rowpos, inpchk, buffchk, val,
    pos, count = 0},
  (*Print[MatrixForm[#]&/@{temp1,temp2,temp3,temp4}];*)
  While[True,
   flag = {};(* this is required if Length@rowchk is zero,
   There will be no do loop*)
   count += 1;
   rowchk = Flatten@Position[Total[temp1, {2}], x_ /; x < m];
   colchk = Flatten@Position[Total[temp1], x_ /; x < m];
   Do[
    (*Print[i];*)
    flag = {};
    rowpos = Flatten@Position[temp1[[i]], x_ /; x < l];
    inpchk = Intersection[rowpos, colchk];
    (*Print[rowpos];
    Print[colchk];
    Print[
    inpchk];*)
    (*The above statement checks positions at which \
no. of packets less than l and row and col sum less than m*)

    buffchk =
     Intersection[inpchk, Flatten@Position[temp3[[i]], x_ /; x > 0]];
    If[Length[buffchk] > 0,
     pos = RandomChoice[buffchk];(*should not be random ,
     be based on minimum time*)
     val = Min[temp4[[i]][[pos]]];
     temp2[[i]][[pos]] = Sort[Append[temp2[[i]][[pos]], val]];
```

```
    (*tempinptime changed*)

    temp1[[i]][[pos]] = Length[temp2[[i]][[pos]]];(*tempinp changed*)

        temp4[[i]][[pos]] =
     Take[temp4[[i]][[
       pos]], -(Length[temp4[[i]][[pos]]] - 1)];(*buffertime changed*)

        temp3[[i]][[pos]] =
     Length[temp4[[i]][[pos]]];(*buffer changed*)

    AppendTo[flag, 1];
    Break[],
    AppendTo[flag, 0]]
   , {i, RandomSample[rowchk]}];
  (*Print[flag];*)

  If[Mod[count, 200] == 0, Print[count]; Print[flag];
   Print[MatrixForm[temp1]]];
  If[AllTrue[flag, # == 0 &] == True || flag == {}, Break[]]
  ];
 (*AppendTo[lengths2,Length[Flatten[
 temp2]]];*)
 {temp1, temp2, temp3, temp4}
 ]
```

Plot settings

```
R = 16;

k = 20;

SetOptions[Plot,

  BaseStyle -> {FontFamily -> "Helvetica", FontSize -> 16}];

SetOptions[ListPlot,

  BaseStyle -> {FontFamily -> "Helvetica", FontSize -> 16}];

SetOptions[ListLogPlot,

  BaseStyle -> {FontFamily -> "Helvetica", FontSize -> 16}];
```

Blocking probability plots

```
(*Blocking Probability Plots*)


blocking[lambda_, l_] := Module[{m, blockprob, steps},
  m = R;
  steps = If[l > 2 && lambda <= 0.4, 100000, 3000];
  blockprob = Accumulate[ParallelTable[Module[{x},
      x = matgen[1, R, m, l, lambda, k];
      x = rowbuff[x, 1, R, m, l, lambda, k];
      x =
       Transpose[#] & /@
        rowbuff[Transpose[#] & /@ x, 1, R, m, l, lambda, k];
      x = reconsider[x, m, l];
      If[Mod[i, steps/2] == 0, Print[i]];
      {Total[x[[3]], 2], Total[x[[1]], 2] + Total[x[[3]], 2]}], {i,
      steps}]];
  N[#1/#2] & @@@ blockprob
  ]
pblocking =
 Table[{lambda, Last[blocking[lambda, l]]}, {l, {1, 2, 4}}, {lambda,
   0.1, 1, 0.1}]
(*L=1 curve fit*)
lambda = 1;
func1[x_] := 1 - 1/x + Exp[-x]/x
pblockfit = Plot[func1[x], {x, 0, 1}, PlotStyle -> Red]

blockinglm[lambda_, l_] := Module[{m, blockprob, steps},
  m = R;
  steps = If[lambda < 0.5, 1000000, 2000];
  blockprob = Accumulate[ParallelTable[Module[{x},
      x = matgen[1, R, m, l, lambda, k];
```

```
      x = rowbuff[x, 1, R, m, l, lambda, k];
      If[Mod[i, 500000] == 0, Print[lambda]];
      {Total[x[[3]], 2], Total[x[[1]], 2] + Total[x[[3]], 2]}], {i,
      steps}]];
  N[#1/#2] & @@@ blockprob
  ]
blockinglmdata =
 Table[{lambda, Last[blockinglm[lambda, R]]}, {lambda, 0.1, 1, 0.1}]
```

Parallel computation of Latency of M = R

```
(*m=R*)
latl1 =
   ParallelTable[
    Module[{steps, intime, outtime, lengthbuffer, x, y, block1,
      block2, block3, latency, latency2},
     m = R;
     steps =
      If[lambda >= 0.9, 3000,
       1000];(*number of slots for which simulation is run*)

     x = matgen[1, R, m, l, lambda, k];
     x = rowbuff[x, 1, R, m, l, lambda, k];
     (*MatrixForm/@x*)

     Print[ToString@R <> " " <> ToString@l <> " " <>
       ToString@lambda];
     latency = Table[
       If[Mod[i, 1000] == 0,
        Print[ToString@R <> " " <> ToString@l <> " " <>
          ToString@lambda <> " " <> ToString@i]];
       y = matgen[i, R, m, l, lambda, k];
       block1 = lbuff[x, y, i, R, m, l, lambda, k];
       block2 = rowbuff[block1, i, R, m, l, lambda, k];
       block3 =
        Transpose[#] & /@
         rowbuff[Transpose[#] & /@ block2, i, R, m, l, lambda, k];
       x = reconsider[block3, m, l];
       Flatten[ConstantArray[i*k, Length[Flatten[x[[2]]]]]] -
        Flatten[x[[2]]]
       ,
```

```
        {i, 2, steps}
        ];
      latency2 = Floor[Flatten@latency/k];
      {lambda, N[Mean[latency2]]}], {l, {m}}, {lambda,
      Join[Range[0.2, 0.8, 0.2],
       Range[0.9, 0.98, 0.02]]}]; // AbsoluteTiming
ListPlot[latl1]

(*m=R*)
latmR =
   ParallelTable[
    Module[{steps, intime, outtime, lengthbuffer, x, y, block1,
      block2, block3, latency, latency2},
     m = R;
     steps =
      If[lambda >= 0.9, 3000,
       1000];(*number of slots for which simulation is run*)

     x = matgen[1, R, m, l, lambda, k];
     x = rowbuff[x, 1, R, m, l, lambda, k];
     x = Transpose[#] & /@
       rowbuff[Transpose[#] & /@ x, 1, R, m, l, lambda, k];
     x = reconsider[x, m, l];
     (*MatrixForm/@x*)

     Print[ToString@R <> " " <> ToString@l <> " " <>
       ToString@lambda];
     latency = Table[
       If[Mod[i, 1000] == 0,
        Print[ToString@R <> " " <> ToString@l <> " " <>
          ToString@lambda <> " " <> ToString@i]];
```

```
  y = matgen[i, R, m, l, lambda, k];
  block1 = lbuff[x, y, i, R, m, l, lambda, k];
  block2 = rowbuff[block1, i, R, m, l, lambda, k];
  block3 =
   Transpose[#] & /@
    rowbuff[Transpose[#] & /@ block2, i, R, m, l, lambda, k];
  x = reconsider[block3, m, l];
  Flatten[ConstantArray[i*k, Length[Flatten[x[[2]]]]]] -
   Flatten[x[[2]]]
  ,
  {i, 2, steps}
  ];
latency2 = Floor[Flatten@latency/k];
{lambda, N[Mean[latency2]]}], {l, {2, 4, 8}}, {lambda,
Join[Range[0.2, 0.8, 0.2],
 Range[0.9, 0.98, 0.02]]}]; // AbsoluteTiming
```

Parallel computation of Latency of M = 2R

```
(*m=2R*)
latm2R =
   ParallelTable[
    Module[{steps, intime, outtime, lengthbuffer, x, y, block1,
      block2, block3, latency, latency2},
     m = 2 R;
     steps =
      If[lambda >= 1.65, 3000,
       1000];(*number of slots for which simulation is run*)

     x = matgen[1, R, m, l, lambda, k];
     x = rowbuff[x, 1, R, m, l, lambda, k];
     x = Transpose[#] & /@
       rowbuff[Transpose[#] & /@ x, 1, R, m, l, lambda, k];
     x = reconsider[x, m, l];
     (*MatrixForm/@x*)

     Print[ToString@R <> " " <> ToString@l <> " " <>
       ToString@lambda];
     latency = Table[
       If[Mod[i, 1000] == 0,
        Print[ToString@R <> " " <> ToString@l <> " " <>
          ToString@lambda <> " " <> ToString@i]];
       y = matgen[i, R, m, l, lambda, k];
       block1 = lbuff[x, y, i, R, m, l, lambda, k];
       block2 = rowbuff[block1, i, R, m, l, lambda, k];
       block3 =
        Transpose[#] & /@
         rowbuff[Transpose[#] & /@ block2, i, R, m, l, lambda, k];
       x = reconsider[block3, m, l];
```

```
    Flatten[ConstantArray[i*k, Length[Flatten[x[[2]]]]] -
     Flatten[x[[2]]]
    ,
    {i, 2, steps}
    ];
  latency2 = Floor[Flatten@latency/k];
  {lambda, N[Mean[latency2]]}], {l, {2, 4, 8}}, {lambda,
  Join[Range[0.2, 1.6, 0.2],
   Range[1.65, 1.95, 0.05]]}]; // AbsoluteTiming
```

A.2 Exhaustive wavelength assignment

The following code runs an exhaustive depth first search algorithm on the traffic matrix for wavelength assignment. This was the first attempt to check whether a wavelength assignment exists. For faster approaches, reader is advised to read literature on Birkhoff-Van Neumann decomposition.

```
(*wavelength assignment algorithm*)
(*depth first search and constrained propagation*)
timemachine = {};
tried = {{}};
pos = {};
order = Position[x[[1]], z_ /; z > 0];
temporder = order
level = Length[timemachine];
flag = 0;
temp = ConstantArray[{}, {R, R}];
counter = 0;
While[True,
 hori = Table[Complement[Range[m], Flatten[temp[[i]]]], {i, R}];
```

```
verti = Table[
  Complement[Range[m], Flatten[Transpose[temp][[j]]]], {j, R}];
Do[
 flag = 0;
 Print[ind];
 i = ind[[1]];
 j = ind[[2]];
 inter = Intersection[hori[[i]], verti[[j]]];

 If[Length[inter] > x[[1]][[i]][[j]] && ! MemberQ[pos, ind],
  AppendTo[timemachine, temp];
  AppendTo[tried, {}];
  AppendTo[tried[[Length[timemachine]]],
   Last[Complement[Subsets[inter, {x[[1]][[i]][[j]]}],
     tried[[Length@timemachine]]]]];
  AppendTo[pos, ind]
  ];

 If[Length[inter] > x[[1]][[i]][[j]],
  temp[[i]][[j]] =
   Last[Flatten[Extract[tried, Position[pos, z_ /; z == ind]], 1]];
  hori[[i]] = Complement[hori[[i]], Flatten[temp[[i]]]];
  verti[[j]] = Complement[verti[[j]], Flatten[Transpose[temp][[j]]]]
  ];
 If[Length[inter] == x[[1]][[i]][[j]],
  temp[[i]][[j]] = inter;
  hori[[i]] = Complement[hori[[i]], Flatten[temp[[i]]]];
  verti[[j]] = Complement[verti[[j]], Flatten[Transpose[temp][[j]]]]
  ];

 If[Length[inter] < x[[1]][[i]][[j]],
```

```
While[True,
 temp = Last[timemachine];
 hori = Table[Complement[Range[m], Flatten[temp[[a]]]], {a, R}];
 verti =
  Table[Complement[Range[m], Flatten[Transpose[temp][[b]]]], {b,
    R}];
 a = Last[pos][[1]];
 b = Last[pos][[2]];
 inter = Intersection[hori[[a]], verti[[b]]];
 If[Complement[Subsets[inter, {x[[1]][[a]][[b]]}],
    tried[[-2]]] == {},
  timemachine = Drop[timemachine, -1];
  If[timemachine == {}, Break[]];
  temp = Last[timemachine];
  If[Last[tried] == {}, tried = Drop[tried, -2];
   AppendTo[tried, {}], tried = Drop[tried, -1]];
  pos = Drop[pos, -1];
  ,
  tried[[-2]] =
   Append[tried[[-2]],
    Last[Complement[Subsets[inter, {x[[1]][[a]][[b]]}],
      tried[[-2]]]]];
  temporder =
   Take[order, {Flatten[Position[order, z_ /; z == Last[pos]]][[
      1]], Length[order]}];
  flag = 1;
  Break[]
  ]
 ]
];
```

```
 If[timemachine == {} || flag == 1,
  If[timemachine == {}, Print["fail"]]; Break[]];
 , {ind, temporder}];
If[Length[Flatten[temp]] == Total[x[[1]], 2], Print["success"];
 Break[]];
If[timemachine == {}, Break[]]
]
```

Appendix B

Measurement automation with Python

B.1 Python code

This section forms the backbone of all the measurements in this Thesis. Keithley Source meters, Thorlabs Photodiode, Yokogawa OSA, Yenista/EXFO Tunable Laser, OSICS Frame and attenuator and many other instruments were controlled with Pyvisa [69] with NI visa backend. All code was run on Jupyter Notebook [70]. The advantage of running code on Jupyter notebooks is that all the code can run and visualized through a notebook interface on any web browser. Numpy[52], Scipy [71] and Pandas[72] packages are heavily used. All plots were implemented with Matplotlib package in python [73] Anaconda with Python 3.6 [74] was used for all code.

Following dependencies are a must for every python program that uses the following code along with specific packages the reader wishes to use in their research.

```
import visa
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
from time import sleep
from time import time
rm = visa.ResourceManager()
rm.list_resources()
```

Some important functions which are repeatedly used for measurements are described below:

1. keithely_init: Initializes a sourcemeter to a given voltage. Sets the compliance current to 3 mA. keithley_add is a list of GPIB id's, for ex. [22,21] or [5].

2. keithley_iv: performs an IV sweep with 20 measurements for each V. Average current for each V is returned.

3. yoko_pre: performs a full range sweep of the OSA

4. yoko_run: returns the maximum power for a given wavelength. The function conducts a scan from wav-span/2 , wav+span/2 with a resolution of 0.02 nm. A lower span reduces total scanning time when used for wavelength sweeping.

5. yenista_sweep: returns a tunable laser sweep given an input array with wavelength points. Input wavelength array points might not be equally spaced.

6. CE_ring_OSA_lock: through and drop version. Genetic algorithm for multivariate optimization. This is a blackbox optimization method [75].

7. hill_climb: through and drop port optimization functions optimize the power on the Yokogawa OSA and Thorlabs Powermeter PM100USB.

8. gen_wav_arr: helper function for a fine coarse tuning. This function generates points that are close together at the maximum and coarser away from the maxima. This can used as input into yenista_sweep for faster sweep/

Initialize Keithley Source meters Model 2401

```
#initialize the Keithleys
def keithley_init(keithley_add,voltset = 2):
    keithley = []
    for i in keithley_add:
        keithley.append(rm.open_resource("GPIB0::"+str(i)+"::INSTR"))
    print(keithley)
    for i in range(len(keithley)):
        keithley[i].write('*RST')
        keithley[i].Inputbuffersize = 100000
        keithley[i].OutputBuffersize = 2000
        limit_current  = 3 #magnitude of current in mA
        keithley[i].write(':SENS:CURR:PROT '+str(limit_current)+
        ↪  'E-3')
        keithley[i].write(':SOUR:CURR:LEV 0')
        keithley[i].write(':SOUR:VOLT:LEV 0')
        # change the level from 0 if required
        #keithley.write(':SOUR:CURR:RANG 1')
        keithley[i].write(':SOUR:VOLT:RANG 7')
        keithley[i].write('SOUR:FUNC VOLT') # source voltage
        keithley[i].write('SENS:FUNC "CURR"')
        #voltset = 2 #magnitude of voltage in V
        if voltset<10:
            keithley[i].write(':SOUR:VOLT '+str(voltset))
        else:
```

```
24              keithley[i].write(':SOUR:VOLT '+str(2))
25          keithley[i].write(':INIT')
26          keithley[i].write(':OUTP ON')
27          keithley[i].write(':SYST:LOC')
28      return keithley
```

IV sweep of Keithley Source meters Model 2401

```
1  def keithley_iv(keithley,initial,voltrange):
2      # give an initial voltage initial so that after the code is run
       ↪ the keithley is set back to its initial voltage
3      data = []
4      avg_meas = np.zeros(20)
5      for i in voltrange:
6          keithley.write(':SOUR:VOLT '+str(i))
7          sleep(0.1)
8          for j in range(20):
9              meas = keithley.read().strip().split(",")
10             avg_meas[j] = [float(meas[k]) for k in
                ↪ range(len(meas))][1]
11         data.append(np.mean(avg_meas))
12     keithley.write(':SOUR:VOLT '+str(initial))
13     return data
```

Yokogawa OSA AQ6370

```
# should run this before running sweep
def yoko_pre(yoko):
    yoko.write('*RST')
    yoko.timeout = None
    yoko.write(':INIT:CONT ON')
    yoko.write(':SENS:SWE:POIN auto')
    yoko.write(':init:imm;')
# for repeated runs during a sweep
def yoko_run(yoko,wav,span):
    yoko.timeout = None
    yoko.write(':SENS:SENS MID')
    yoko.write(':SENS:SWE:SPE 2x')
    yoko.write(':SENS:BWID:RES 0.02nm')
    yoko.write(':SENS:WAV:STAR '+str(wav-span/2)+'nm')
    yoko.write(':SENS:WAV:STOP '+str(wav+span/2)+'nm')
    yoko.write(':init:imm;')
    yoko.write(':INIT')
    yoko.write(':TRAC:DATA: Y? TRA;')
    sleep(0.1)
    data1 = yoko.read()
    yoko.write(':TRAC:DATA: X? TRA;')
    sleep(0.1)
    data2 = yoko.read()
    data1 = np.fromstring(data1,dtype = float,sep=',')
```

```
25      #print(data1)
26      data2 = 1e6*np.fromstring(data2,dtype = float,sep=',')
27      #print(data2)
28      data = [data2,data1]
29      return data
```

Wavelength sweep with Yenista Tunable laser

Used with Tunics T100S-HP.

```
#yenista wav sweep WLnm is wavelength array in nm
def yenista_sweep(yenista,yoko,WLnm,power = 0):
    data = []
    yenista.write('P=%f\n'%power)
    yenista.write('ENABLE')
    for j in range(len(WLnm)):
        yenista.write('L=%f\n'%WLnm[j])
        if j==0:
            yoko_pre(yoko)
        sleep(0.2)
        data.append(yoko_run(yoko,WLnm[j],0.1))
        sleep(0.1)
    return [max(data[k][1]) for k in range(len(data))]
```

Cross entropy algorithm: drop port

```
# CE OSA ring locking function
def CE_ring_OSA_lock(yenista,yoko,heater,wav,guess = 2,std_guess =
    0.3,res = 0.01):
    n1 = 100 # total number of samples per loop
    n2 = 10  # elite samples
    yenista.write('L=%f\n'%wav)
    mean = [ guess,guess]
    std = [ std_guess,  std_guess]
    print([mean,std])
    #y = guess*np.ones(10,dtype = float)
    # std limit of 0.01 V can be changed
    while std[0]> res or std[1] > res:
        v1 = np.random.normal(mean[0],std[0],n1)
        v1 = np.round(v1[(v1<10)],3)
        v2 = np.random.normal(mean[1],std[1],n1)
        v2 = np.round(v2[(v2<10)],3)
        #v1 = np.concatenate((y,v1))
        #v2 = np.concatenate((y,v2))
        if len(v1)<len(v2):
            v2 = v2[:len(v1)]
        else:
            v1 = v1[:len(v2)]
        maxv = np.zeros(len(v1))
```

```
23          # CE loop to measure detected powers for each voltage across
            ↪ two heaters
24          for j in range(len(v1)):
25              heater[0].write(':SOUR:VOLT '+str(v1[j]))
26              heater[1].write(':SOUR:VOLT '+str(v2[j]))
27              if j==0:
28                  yoko_pre(yoko)
29                  sleep(0.2)
30              sleep(0.1)
31              temp = yoko_run(yoko,wav,0.3)
32              maxv[j] = max(temp[1])
33          temp2 =v1[np.argsort(maxv)[-n2:]]
34          temp3 =v2[np.argsort(maxv)[-n2:]]
35          mean = np.round(np.mean([temp2,temp3],1),3)
36          std = np.round(np.std([temp2,temp3],1),3)
37          print([mean,std])
38
39      heater[0].write(':SOUR:VOLT '+str(mean[0]))
40      heater[1].write(':SOUR:VOLT '+str(mean[1]))
41      return [mean,std]
```

Cross entropy algorithm through port

```
def CE_ring_OSA_lock_through(yenista,yoko,heater,wav,guess =
    2,std_guess = 0.3,res = 0.01):
    n1 = 100 # total number of samples per loop
    n2 = 10  # elite samples
    yenista.write('L=%f\n'%wav)
    mean = [ guess,guess]
    std = [ std_guess,  std_guess]
    print([mean,std])
    #y = guess*np.ones(10,dtype = float)
    # std limit of 0.01 V can be changed
    while std[0]> res or std[1] > res:
        v1 = np.random.normal(mean[0],std[0],n1)
        v1 = np.round(v1[(v1<10)],3)
        v2 = np.random.normal(mean[1],std[1],n1)
        v2 = np.round(v2[(v2<10)],3)
        #v1 = np.concatenate((y,v1))
        #v2 = np.concatenate((y,v2))
        if len(v1)<len(v2):
            v2 = v2[:len(v1)]
        else:
            v1 = v1[:len(v2)]
        maxv = np.zeros(len(v1))
        # CE loop to measure detected powers for each voltage across
        two heaters
```

```
23          for j in range(len(v1)):
24              heater[0].write(':SOUR:VOLT '+str(v1[j]))
25              heater[1].write(':SOUR:VOLT '+str(v2[j]))
26              if j==0:
27                  yoko_pre(yoko)
28                  sleep(0.2)
29              sleep(0.1)
30              temp = yoko_run(yoko,wav,0.3)
31              maxv[j] = min(temp[1])
32          temp2 =v1[np.argsort(maxv)[-n2:]]
33          temp3 =v2[np.argsort(maxv)[-n2:]]
34          mean = np.round(np.mean([temp2,temp3],1),3)
35          std = np.round(np.std([temp2,temp3],1),3)
36          print([mean,std])
37
38      heater[0].write(':SOUR:VOLT '+str(mean[0]))
39      heater[1].write(':SOUR:VOLT '+str(mean[1]))
40      return [mean,std]
```

Random Stochastic Hill Climbing

```
def hill_climb(yenista,yoko,keithley,wav,guess = 4,std_guess =
    0.01,n_iter = 300):
    global iter_power
    keithley[0].write(':SOUR:VOLT '+str(guess))
    keithley[1].write(':SOUR:VOLT '+str(guess))
    yenista.write('L=%f\n'%wav)
    bestpair = [guess,guess]
    yoko_pre(yoko)
    sleep(1)
    for i in range(n_iter):
        if i==0:
            v1 = guess
            v2 = guess
        else:
            v1 = bestpair[0] +
             np.round(np.random.normal(0,std_guess),3)
            v2 = bestpair[1] +
             np.round(np.random.normal(0,std_guess),3)
        keithley[0].write(':SOUR:VOLT '+str(v1))
        keithley[1].write(':SOUR:VOLT '+str(v2))
        sleep(0.1)
        if i==0:
            print(i)
            bestvalue = max(yoko_run(yoko,wav,0.1)[1])
```

```
22              bestpair = [v1,v2]
23              print(bestvalue)
24          else:
25              value = max(yoko_run(yoko,wav,0.1)[1])
26              if value>bestvalue:
27                  bestvalue = value
28                  bestpair = [v1,v2]
29          if i%50==0:
30              print(i)
31              print(bestvalue)
32              print(bestpair)
33          iter_power.append(bestvalue)
34      keithley[0].write(':SOUR:VOLT '+str(bestpair[0]))
35      keithley[1].write(':SOUR:VOLT '+str(bestpair[1]))
36      print(i)
37      print(bestvalue)
38      print(bestpair)
39      return
```

Random Stochastic Hill Climbing : through port

```
1  def hill_climb_th(yenista,yoko,keithley,wav,guess = 4,std_guess =
   ↪  0.01,n_iter = 300):
2      keithley[0].write(':SOUR:VOLT '+str(guess))
3      keithley[1].write(':SOUR:VOLT '+str(guess))
4      yenista.write('L=%f\n'%wav)
5      bestpair = [guess,guess]
6      yoko_pre(yoko)
7      sleep(1)
8      for i in range(n_iter):
9          if i==0:
10             v1 = guess
11             v2 = guess
12         else:
13             v1 = bestpair[0] +
                ↪  np.round(np.random.normal(0,std_guess),3)
14             v2 = bestpair[1] +
                ↪  np.round(np.random.normal(0,std_guess),3)
15         keithley[0].write(':SOUR:VOLT '+str(v1))
16         keithley[1].write(':SOUR:VOLT '+str(v2))
17         sleep(0.1)
18         if i==0:
19             print(i)
20             bestvalue = max(yoko_run(yoko,wav,0.1)[1])
21             bestpair = [v1,v2]
```

```
22              print(bestvalue)
23          else:
24              value = max(yoko_run(yoko,wav,0.1)[1])
25              if value<bestvalue:
26                  bestvalue = value
27                  bestpair = [v1,v2]
28          if i%50==0:
29              print(i)
30              print(bestvalue)
31              print(bestpair)
32      keithley[0].write(':SOUR:VOLT '+str(bestpair[0]))
33      keithley[1].write(':SOUR:VOLT '+str(bestpair[1]))
34      print(i)
35      print(bestvalue)
36      print(bestpair)
37      return
```

Fine coarse array generation for laser scan

```
1  def gen_wav_arr(maxwav,res_fine = 0.04,res_coarse = 0.2,span = 2):
2      temp1 = np.arange(maxwav-span,maxwav+span,res_fine)
3      temp2 = np.arange(maxwav-26-span,maxwav-26+span,res_fine)
4      temp3 = np.arange(np.max(temp2),np.min(temp1),res_coarse)
5      WLnm = np.concatenate((np.concatenate((temp2,temp3)),temp1))
6      return WLnm
```

Stochastic Hill Climb with Thorlabs Photodiode

```
def hill_climb_pd(yenista,pd,keithley,wav,guess = 4,std_guess =
    0.01,n_iter = 300):
    keithley[0].write(':SOUR:VOLT '+str(guess))
    keithley[1].write(':SOUR:VOLT '+str(guess))
    keithley[0].read().strip().split(",")
    keithley[1].read().strip().split(",")
    yenista.write('L=%f\n'%wav)
    bestpair = [guess,guess]
    yoko_pre(yoko)
    sleep(1)
    for i in range(n_iter):
        if i==0:
            v1 = guess
            v2 = guess
        else:
            v1 = bestpair[0] +
             np.round(np.random.normal(0,std_guess),3)
            v2 = bestpair[1] +
             np.round(np.random.normal(0,std_guess),3)
        keithley[0].write(':SOUR:VOLT '+str(v1))
        keithley[1].write(':SOUR:VOLT '+str(v2))
        keithley[0].read().strip().split(",")
        keithley[1].read().strip().split(",")
        #sleep(0.1)
```

```
22          if i==0:
23              #print(i)
24              meas = PD[0].read().strip().split(",")
25              curr = [float(meas[k]) for k in range(len(meas))][1]
26              bestvalue = -curr
27              bestpair = [v1,v2]
28              #print(bestvalue)
29          else:
30              meas = PD[0].read().strip().split(",")
31              curr = [float(meas[k]) for k in range(len(meas))][1]
32              value = -curr
33              if value>bestvalue:
34                  bestvalue = value
35                  bestpair = [v1,v2]
36          if i%50==0:
37              print(i)
38              print(bestvalue)
39              print(bestpair)
40      keithley[0].write(':SOUR:VOLT '+str(bestpair[0]))
41      keithley[1].write(':SOUR:VOLT '+str(bestpair[1]))
42      keithley[0].read().strip().split(",")
43      keithley[1].read().strip().split(",")
44      print(i)
45      #print(bestvalue)
46      #print(bestpair)
```

Sample Scipy optimize code for microring resonators

```
maxwav = 1556.15
print(maxwav)
x0 = [2,2]
yoko_pre(yoko)
def heater_set(x,yenista = yenista,yoko = yoko,keithley = heater1,wav
    = maxwav):
    yenista.write('L=%f\n'%wav)
    global iter_power
    print(str(x[0]))
    print(x[1])

    if abs(x[0])>9:
        x[0] = 9
    if abs(x[1])>9:
        x[1] = 9
    keithley[0].write(':SOUR:VOLT '+str(x[0]))
    keithley[1].write(':SOUR:VOLT '+str(x[1]))
    sleep(0.1)
    power = max(yoko_run(yoko,wav,0.1)[1])
    #power = power[len(power)//2]
    iter_power.append(power)
    return power
from scipy.optimize import minimize
iter_power = []
```

```
24  res = minimize(heater_set, x0,args = (yenista,yoko,heater1,1537)\
25
26  ,method='Nelder-Mead',tol = 0.05,options={'disp': True})
```

Appendix C

Layouts

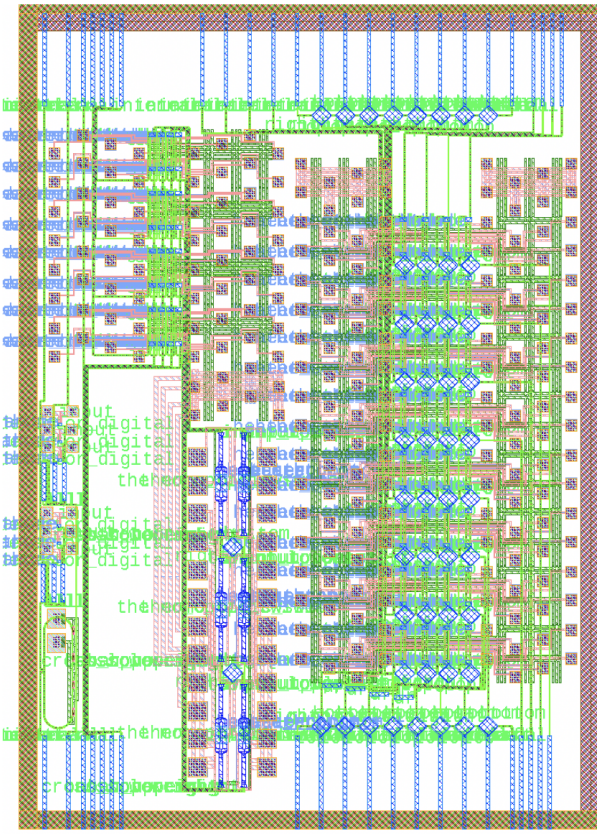


Figure C.1: Run 3 Layout 1

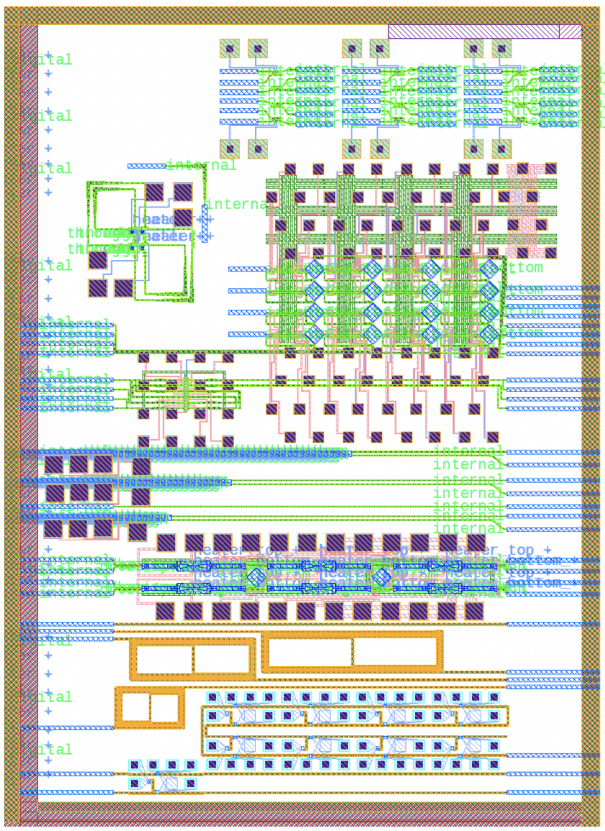


Figure C.2: Run 3 Layout 2

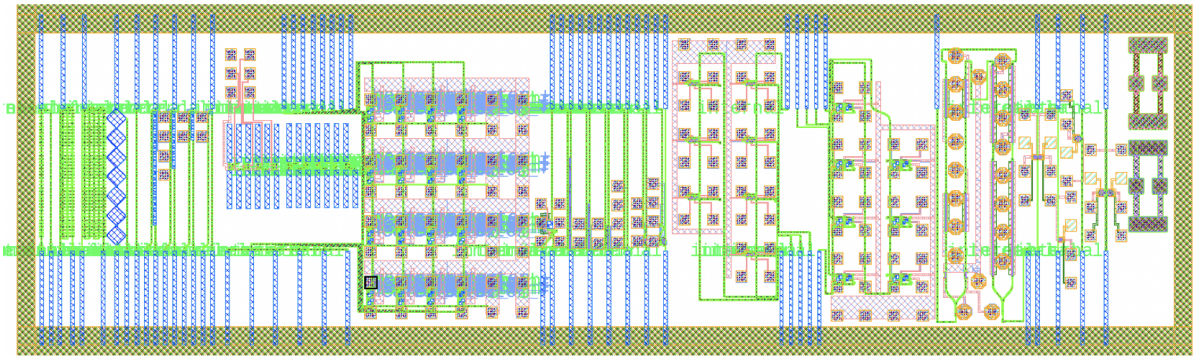


Figure C.3: Run 4 Layout

Bibliography

[1] A. A. M. Saleh, *Scaling-out data centers using photonics technologies*, in *Photonics in Switching*, pp. JM4B–5, Optical Society of America, 2014.

[2] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, *Helios: a hybrid electrical/optical switch architecture for modular data centers*, *ACM SIGCOMM Computer Communication Review* **41** (2011), no. 4 339–350.

[3] A. S. Khope, A. A. Saleh, J. E. Bowers, and R. C. Alferness, *Elastic wdm crossbar switch for data centers*, *IEEE Optical Interconnects (OI) Conference* (2016) 48–49.

[4] A. S. Khope, M. Saeidi, R. Yu, X. Wu, A. M. Netherton, Y. Liu, Z. Zhang, Y. Xia, G. Fleeman, A. Spott, *et. al.*, *Multi-wavelength selective crossbar switch*, *Optics express* **27** (2019), no. 4 5203–5216.

[5] A. S. Khope, T. Hirokawa, A. M. Netherton, M. Saeidi, Y. Xia, N. Volet, C. Schow, R. Helkey, L. Theogarajan, A. A. Saleh, *et. al.*, *On-chip wavelength locking for photonic switches*, *Optics letters* **42** (2017), no. 23 4934–4937.

[6] C. Clos, *A study of non-blocking switching networks*, *Bell System Technical Journal* **32** (1953), no. 2 406–424.

[7] Y. Kawajiri, *512× 512 port 3d mems optical switch module with toroidal concave mirror*, in *15ˆ< th> Microoptics Conference (MOC), Tokyo, Japan, Oct. 2009*, 2009.

[8] K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, K. Hill, J. Hiller, *et. al.*, *Exascale computing study: Technology challenges in achieving exascale systems*, *Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep* **15** (2008).

[9] R. Ji, L. Yang, L. Zhang, Y. Tian, J. Ding, H. Chen, Y. Lu, P. Zhou, and W. Zhu, *Five-port optical router for photonic networks-on-chip*, *Optics express* **19** (2011), no. 21 20258–20268.

[10] N. Sherwood-Droz, H. Wang, L. Chen, B. G. Lee, A. Biberman, K. Bergman, and M. Lipson, *Optical 4× 4 hitless silicon router for optical networks-on-chip (noc)*, *Optics express* **16** (2008), no. 20 15915–15922.

[11] Q. Zhu, X. Jiang, Y. Yu, R. Cao, H. Zhang, D. Li, Y. Li, L. Zeng, X. Guo, Y. Zhang, *et. al.*, *Automated wavelength alignment in a 4× 4 silicon thermo-optic switch based on dual-ring resonators*, *IEEE Photonics Journal* **10** (2018), no. 1 1–11.

[12] Q. Cheng, L. Y. Dai, N. C. Abrams, Y.-H. Hung, P. E. Morrissey, M. Glick, P. O'Brien, and K. Bergman, *Ultralow-crosstalk, strictly non-blocking microring-based optical switch*, *Photonics Research* **7** (2019), no. 2 155–161.

[13] Y. Huang, Q. Cheng, Y.-H. Hung, H. Guan, A. Novack, M. Streshinsky, M. Hochberg, and K. Bergman, *Dual-microring resonator based 8× 8 silicon photonic switch*, in *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1–3, IEEE, 2019.

[14] K. Tanizawa, K. Suzuki, M. Toyama, M. Ohtsuka, N. Yokoyama, K. Matsumaro, M. Seki, K. Koshino, T. Sugaya, S. Suda, G. Cong, T. Kimura, K. Ikeda, S. Namiki, and K. Hitoshi, *Ultra-compact 32× 32 strictly-non-blocking si-wire optical switch with fan-out lga interposer*, *Opt. Express* **23** (2015), no. 13 17599–17606.

[15] N. Dupuis, A. V. Rylyakov, C. L. Schow, D. M. Kuchta, C. W. Baks, J. S. Orcutt, D. M. Gill, W. M. Green, and B. G. Lee, *Nanosecond-scale mach-zehnder-based cmos photonic switch fabrics*, *J. Lightw. Technol.* **35** (2017), no. 4 615–623.

[16] T. Goh, A. Himeno, M. Okuno, H. Takahashi, and K. Hattori, *High-extinction ratio and low-loss silica-based 88 strictly nonblocking thermooptic matrix switch*, *Journal of lightwave technology* **17** (1999), no. 7 1192.

[17] P. DasMahapatra, R. Stabile, A. Rohit, and K. A. Williams, *Optical crosspoint matrix using broadband resonant switches*, *IEEE J. Sel. Top. Quantum Electron.* **20** (2014), no. 4 1–10.

[18] X. Zheng, V. Kaman, S. Yuan, Y. Xu, O. Jerphagnon, A. Keating, R. C. Anderson, H. N. Poulsen, B. Liu, J. R. Sechrist, *et. al.*, *Three-dimensional mems photonic cross-connect switch design and performance*, *IEEE Journal of Selected Topics in Quantum Electronics* **9** (2003), no. 2 571–578.

[19] J. Bowers, *Low power 3d mems optical switches*, in *2009 IEEE/LEOS International Conference on Optical MEMS and Nanophotonics*, pp. 152–153, IEEE, 2009.

[20] T. J. Seok, J. Luo, Z. Huang, K. Kwon, J. Henriksson, J. Jacobs, L. Ochikubo, R. S. Muller, and M. C. Wu, *Mems-actuated 8× 8 silicon photonic wavelength-selective switches with 8 wavelength channels*, in *CLEO: Science and Innovations*, pp. STu4B–1, Optical Society of America, 2018.

[21] R. Stabile, A. Albores-Mejia, and K. Williams, *Monolithic active-passive 16× 16 optoelectronic switch*, *Optics letters* **37** (2012), no. 22 4666–4668.

[22] R. Stabile, A. Rohit, and K. Williams, *Monolithically integrated 8× 8 space and wavelength selective cross-connect*, *Journal of Lightwave Technology* **32** (2014), no. 2 201–207.

[23] A. Rohit, J. Bolk, X. J. Leijtens, and K. A. Williams, *Monolithic nanosecond-reconfigurable 4 x 4 space and wavelength selective cross-connect*, *Journal of Lightwave Technology* **30** (2012), no. 17 2913–2921.

[24] L. Qiao, W. Tang, and T. Chu, *32× 32 silicon electro-optic switch with built-in monitors and balanced-status units*, *Scientific Reports* **7** (2017) 42306.

[25] L. Qiao, W. Tang, and T. Chu, *16× 16 non-blocking silicon electro-optic switch based on mach-zehnder interferometers*, in *Optical Fiber Communication Conference*, pp. Th1C–2, Optical Society of America, 2016.

[26] D. Celo, D. J. Goodwill, J. Jiang, P. Dumais, C. Zhang, F. Zhao, X. Tu, C. Zhang, S. Yan, J. He, *et. al.*, *32× 32 silicon photonic switch*, in *OptoElectronics and Communications Conference (OECC) held jointly with 2016 International Conference on Photonics in Switching (PS), 2016 21st*, pp. 1–3, IEEE, 2016.

[27] L. Lu, S. Zhao, L. Zhou, D. Li, Z. Li, M. Wang, X. Li, and J. Chen, *16× 16 non-blocking silicon optical switch based on electro-optic mach-zehnder interferometers*, *Optics express* **24** (2016), no. 9 9295–9307.

[28] Z. Lu, D. Celo, H. Mehrvar, E. Bernier, and L. Chrostowski, *High-performance silicon photonic tri-state switch based on balanced nested mach-zehnder interferometer*, *Scientific reports* **7** (2017), no. 1 12244.

[29] N. Dupuis, B. G. Lee, A. V. Rylyakov, D. M. Kuchta, C. W. Baks, J. S. Orcutt, D. M. Gill, W. M. Green, and C. L. Schow, *Design and fabrication of low-insertion-loss and low-crosstalk broadband* 2×2 *mach–zehnder silicon photonic switches*, *Journal of Lightwave Technology* **33** (2015), no. 17 3597–3606.

[30] R. Konoike, K. Suzuki, T. Inoue, T. Matsumoto, T. Kurahashi, A. Uetake, K. Takabayashi, S. Akiyama, S. Sekiguchi, K. Ikeda, *et. al.*, *Lossless operation of soa-integrated silicon photonics switch for 8× 32-gbaud 16-qam wdm signals*, in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pp. 1–3, IEEE, 2018.

[31] B. G. Lee, A. Biberman, P. Dong, M. Lipson, and K. Bergman, *All-optical comb switch for multiwavelength message routing in silicon photonic networks*, *IEEE Photonics Technology Letters* **20** (2008), no. 10 767–769.

[32] Y. Vlasov, W. M. Green, and F. Xia, *High-throughput silicon nanophotonic wavelength-insensitive switch for on-chip optical networks*, *nature photonics* **2** (2008), no. 4 242.

[33] K. Suzuki, R. Konoike, J. Hasegawa, S. Suda, H. Matsuura, K. Ikeda, S. Namiki, and H. Kawashima, *Low insertion loss and power efficient 32× 32 silicon photonics switch with extremely-high-δ plc connector*, in *Optical Fiber Communication Conference*, pp. Th4B–5, Optical Society of America, 2018.

[34] Q. Cheng, A. Wonfor, R. V. Penty, and I. H. White, *Scalable, low-energy hybrid photonic space switch*, *Journal of Lightwave Technology* **31** (2013), no. 18 3077–3084.

[35] Q. Cheng, S. Rumley, M. Bahadori, and K. Bergman, *Photonic switching in high performance datacenters*, *Optics Express* **26** (2018), no. 12 16022–16043.

[36] R. Yu, S. Cheung, Y. Li, K. Okamoto, R. Proietti, Y. Yin, and S. Yoo, *A scalable silicon photonic chip-scale optical switch for high performance computing systems*, *Optics Express* **21** (2013), no. 26 32655–32667.

[37] A. W. Poon, X. Luo, F. Xu, and H. Chen, *Cascaded microresonator-based matrix switch for silicon on-chip optical interconnection*, *Proc. IEEE* **97** (2009), no. 7 1216–1238.

[38] D. Nikolova, D. M. Calhoun, Y. Liu, S. Rumley, A. Novack, T. Baehr-Jones, M. Hochberg, and K. Bergman, *Modular architecture for fully non-blocking silicon photonic switch fabric*, *Microsystems & Nanoengineering* **3** (2017) 16071.

[39] Z. Su, E. Timurdogan, M. Moresco, G. Leake, D. Coolbaugh, and M. Watts, *Wavelength routing and multicasting network in ring-based integrated photonics*, in *Integrated Photonics Research, Silicon and Nanophotonics*, pp. IT4A–3, Optical Society of America, 2015.

[40] Z. Wang, J. Xu, P. Yang, Z. Wang, L. H. K. Duong, and X. Chen, *High-radix nonblocking integrated optical switching fabric for data center*, *Journal of Lightwave Technology* **35** (2017), no. 19 4268–4281.

[41] A. A. M. Saleh, A. S. P. Khope, J. E. Bowers, and R. C. Alferness, *Elastic wdm switching for scalable data center and hpc interconnect networks*, *OptoElectronics and Communications Conference (OECC) and Photonics in Switching (PS)* (2016).

[42] R. A. Soref and B. E. Little, *Proposed n-wavelength m-fiber wdm crossconnect switch using active microring resonators*, *IEEE Photonics Technology Letters* **10** (1998), no. 8 1121–1123.

[43] Y. Goebuchi, M. Hisada, T. Kato, and Y. Kokubun, *Optical cross-connect circuit using hitless wavelength selective switch*, *Optics express* **16** (2008), no. 2 535–548.

[44] V. E. Beneš, *Mathematical theory of connecting networks and telephone traffic*, vol. 17. Academic press, 1965.

[45] G. Porter, R. Strong, N. Farrington, A. Forencich, P. Chen-Sun, T. Rosing, Y. Fainman, G. Papen, and A. Vahdat, *Integrating microsecond circuit switching into the data center*, vol. 43. ACM, 2013.

[46] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano, A. Kanagala, J. Provost, J. Simmons, E. Tanda, J. Wanderer, U. Hölzle, S. Stuart, and A. Vahdat, *Jupiter rising: A decade of clos topologies and centralized control in google's datacenter network*, in *ACM SIGCOMM Computer Communication Review*, vol. 45, pp. 183–197, ACM, 2015.

[47] J. L. Gross and J. Yellen, *Graph theory and its applications*. Chapman and Hall/CRC, 2005.

[48] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data networks*, vol. 2. Prentice-Hall International New Jersey, 1992.

[49] alejopelaez, "Characterizing sums of permutation matrices." `http://math.stackexchange.com/q/583999`, 2013.

[50] Y. Shen, K. Lu, and W. Gu, *Coherent and incoherent crosstalk in wdm optical networks*, *Journal of lightwave technology* **17** (1999), no. 5 759.

[51] Y. Ma, Y. Zhang, S. Yang, A. Novack, R. Ding, A. E.-J. Lim, G.-Q. Lo, T. Baehr-Jones, and M. Hochberg, *Ultralow loss single layer submicron silicon waveguide crossing for soi optical interconnect*, *Optics express* **21** (2013), no. 24 29374–29382.

[52] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, *The numpy array: a structure for efficient numerical computation*, *Computing in Science & Engineering* **13** (2011), no. 2 22–30.

[53] A. Photonics, "American institute of manufacturing integrated photonics." `http://www.aimphotonics.com/pdk/`, 2016.

[54] A. V. Krishnamoorthy, X. Zheng, G. Li, J. Yao, T. Pinguet, A. Mekis, H. Thacker, I. Shubin, Y. Luo, K. Raj, *et. al.*, *Exploiting cmos manufacturing to reduce tuning requirements for resonant optical devices*, *IEEE Photon. J.* **3** (2011), no. 3 567–579.

[55] C. Manganelli, P. Pintus, F. Gambini, D. Fowler, M. Fournier, S. Faralli, C. Kopp, and C. Oton, *Large-fsr thermally tunable double-ring filters for wdm applications in silicon photonics*, *IEEE Photonics Journal* **9** (2017), no. 1 1–10.

[56] M. Mitchell, J. Holland, and S. Forrest, *Relative building-block fitness and the building block hypothesis*, *D. Whitley, Foundations of Genetic Algorithms* **2** (2014) 109–126.

[57] M. R. Watts, W. A. Zortman, D. C. Trotter, G. N. Nielson, D. L. Luck, and R. W. Young, *Adiabatic resonant microrings (arms) with directly integrated thermal microphotonics*, in *Lasers and Electro-Optics, 2009 and 2009 Conference on Quantum electronics and Laser Science Conference. CLEO/QELS 2009. Conference on*, pp. 1–2, IEEE, 2009.

[58] A. Khope, A. M. Netherton, T. Hirokawa, N. Volet, E. Stanton, C. Schow, R. Helkey, A. Saleh, J. Bowers, and R. C. Alferness, *Elastic wdm optoelectronic crossbar switch with on-chip wavelength control*, *Photonics in Switching* (2017) PTh1D–3.

[59] K. Padmaraju, D. F. Logan, T. Shiraishi, J. J. Ackert, A. P. Knights, and K. Bergman, *Wavelength locking and thermally stabilizing microring resonators using dithering signals*, *J. Lightw. Technol.* **32** (2014), no. 3 505–512.

[60] S. Grillanda, M. Carminati, F. Morichetti, P. Ciccarella, A. Annoni, G. Ferrari, M. Strain, M. Sorel, M. Sampietro, and A. Melloni, *Non-invasive monitoring and control in silicon photonics using cmos integrated electronics*, *Optica* **1** (2014), no. 3 129–136.

[61] Y. Li and A. W. Poon, *Active resonance wavelength stabilization for silicon microring resonators using slope-detection with an inresonator defect-state-absorption-based photodetector*, *Conference on Lasers and Electro-Optics (CLEO)* (2015) 1–2.

[62] A. Gazman, C. Browning, Z. Zhu, L. P. Barry, and K. Bergman, *Automated thermal stabilization of cascaded silicon photonic ring resonators for reconfigurable wdm applications*, . URL.

[63] P. Dong, R. Gatdula, K. Kim, J. H. Sinsky, A. Melikyan, Y.-K. Chen, G. de Valicourt, and J. Lee, *Simultaneous wavelength locking of microring modulator array with a single monitoring signal*, *Opt. Express* **25** (2017), no. 14 16040–16046.

[64] Z. Su, E. Timurdogan, J. Sun, M. Moresco, G. Leake, D. Coolbaugh, and M. R. Watts, *An on-chip partial drop wavelength selective broadcast network*, *Lasers and Electro-Optics (CLEO), 2014 Conference on* (2014) 1–2.

[65] J. A. Nelder and R. Mead, *A simplex method for function minimization*, *The computer journal* **7** (1965), no. 4 308–313.

[66] M. J. Powell, *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, *The computer journal* **7** (1964), no. 2 155–162.

[67] P. Dong, W. Qian, H. Liang, R. Shafiiha, D. Feng, G. Li, J. E. Cunningham, A. V. Krishnamoorthy, and M. Asghari, *Thermally tunable silicon racetrack resonators with ultralow tuning power*, *Optics express* **18** (2010), no. 19 20298–20304.

[68] S. Wolfram *et. al.*, *Mathematica*. Cambridge university press Cambridge, 1996.

[69] P. Authors, "Pyvisa: Control your instruments with python." `https://pyvisa.readthedocs.io/`, 2016.

[70] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. B. Hamrick, J. Grout, S. Corlay, *et. al.*, *Jupyter notebooks-a publishing format for reproducible computational workflows.*, in *ELPUB*, pp. 87–90, 2016.

[71] E. Jones, T. Oliphant, and P. Peterson, {*SciPy*}*: open source scientific tools for* {*Python*}, .

[72] W. McKinney *et. al.*, *Data structures for statistical computing in python*, in *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56, Austin, TX, 2010.

[73] J. D. Hunter, *Matplotlib: A 2d graphics environment*, *Computing in science & engineering* **9** (2007), no. 3 90–95.

[74] M. F. Sanner *et. al.*, *Python: a programming language for software integration and development*, *J Mol Graph Model* **17** (1999), no. 1 57–61.

[75] I. Szita and A. Lörincz, *Learning tetris using the noisy cross-entropy method*, *Neural computation* **18** (2006), no. 12 2936–2941.