# Monolithic Electronic-Photonic Integration in State-of-the-Art CMOS Processes

by

Jason S. Orcutt

M.S. Electrical Engineering
Massachusetts Institute of Technology, 2008
B.S. Electrical Engineering
Columbia University, 2005

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2012



Signature of Author: _____
Department of Electrical Engineering
February 3, 2012

Certified by: _____
Rajeev J. Ram
Professor of Electrical Engineering
Thesis Supervisor

Accepted by: _____
Leslie Kolodziejski
Professor of Electrical Engineering
Chair, Committee for Graduate Students

# Monolithic Electronic-Photonic Integration in State-of-the-Art CMOS Processes

by

Jason S. Orcutt

Submitted to the Department of Electrical Engineering and Computer Science
on February 3, 2012 in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy in
Electrical Engineering and Computer Science

ABSTRACT

As silicon CMOS transistors have scaled, increasing the density and energy efficiency of computation on a single chip, the off-chip communication link to memory has emerged as the major bottleneck within modern processors. Photonic devices promise to break this bottleneck with superior bandwidth-density and energy-efficiency. Initial work by many research groups to adapt photonic device designs to a silicon-based material platform demonstrated suitable independent performance for such links. However, electronic-photonic integration attempts to date have been limited by the high cost and complexity associated with modifying CMOS platforms suitable for modern high-performance computing applications.

In this work, we instead utilize existing state-of-the-art electronic CMOS processes to fabricate integrated photonics by: modifying designs to match the existing process; preparing a design-rule compliant layout within industry-standard CAD tools; and locally-removing the handle silicon substrate in the photonic region through post-processing. This effort has resulted in the fabrication of seven test chips from two major foundries in 28, 45, 65 and 90 nm CMOS processes. Of these efforts, a single die fabricated through a widely available 45nm SOI-CMOS mask-share foundry with integrated waveguides with 3.7 dB/cm propagation loss alongside unmodified electronics with less than 5 ps inverter stage delay serves as a proof-of-concept for this approach. Demonstrated photonic devices include high-extinction carrier-injection modulators, 8-channel wavelength division multiplexing filter banks and low-efficiency silicon germanium photodetectors. Simultaneous electronic-photonic functionality is verified by recording a 600 Mb/s eye diagram from a resonant modulator driven by integrated digital circuits. Initial work towards photonic device integration within the peripheral CMOS flow of a memory process that has resulted in polysilicon waveguide propagation losses of 6.4 dB/cm will also be presented.

Thesis Supervisor: Rajeev J. Ram
Title: Professor of Electrical Engineering

# ACKNOWLEDGMENTS

Many people have contributed in various ways to the creation of this thesis. Most directly, my colleagues at MIT over the years have been a pleasure to work alongside. Ranging from formal collaborations to more random exchanges of ideas and favors, the MIT research community as a whole has exceeded any expectations for approachability and camaraderie. Outside of campus, friends and family have made the experience enjoyable throughout tapeouts, trials and tribulations.

From the various corporate partners who have served as fabrication facilities, intermediaries or collaborators, there are many people to whom I owe many thanks for information and assistance that directly led to the ability to execute on the ideas in this thesis. At Texas Instruments, Dennis Buss gave us the first kind ear to figure out what we were trying to do and why we were trying to do it. He provided key technical insights and put us in touch with the right people to get the details of their state-of-the-art-fabrication processes and how a submitted design truly gets transferred into silicon. Tom Bonifield was one of the key people who provided guidance in many of the judgment calls that had to be made on early design framework integration choices that I had been making with way too little information. At IBM Foundry Design Services, Bill Smith, Kurt Tallman and Joanne Newman have been invaluable in their assistance not only for their information, but also for their patience. To get to them in the first place as well as everything practical that goes into submitting a CMOS design for fabrication under tight deadlines and high stress, Joe Meinhardt and James Gay have provided much appreciated assistance. Recently, through our collaboration with Micron Technology it has been a great pleasure to work with Roy Meade, Zvika Sternberg, Ofer Tehar-Zahav, Reha Bafrali, Gurtej Sandhu and many other people within their organizations. Sanh Tang, Steve Kramer and Nishant Sinha deserve special thanks for our first test fabrication run before the launch of the full POEM program. In aggregate, they have provided me with an absolute deluge of information on what it actually takes to make CMOS work from the STI to the OSG and back to the ULC.

Funding for this work has largely stemmed from Dr. Jagdeep Shah of the Defense Advanced Research Project Agency (DARPA) through a variety of programs. The full alphabet soup of DARPA programs that have contributed to funding the work of this thesis through seedlings or programs is from last to first: POEM, APS, CiPHER, PICO, LEAP, UNIC, GHOST and EPIC. Graduate fellowship funding from the NSF enabled me to pursue ideas that were further afield than existing funded programs at the start of this research.

The intrachip photonics team has consisted of many influential members over the years. Milos Popovic has provided me with extremely insightful information on high index contrast and resonant photonic device design the entire time that I have known him. All that I needed to know about fabrication when Trey Holzwarth was around was that he would somehow make it work. Once he left, Jie Sun and Hanqing Li have led many new projects in campus fabrication post-processing efforts. Anatoly Khilo provided many insights and novel photonic designs during his time with our project in the first few process generations. Erman Timurdogan and Jeff Shainline are continuing excellent photonic design work for novel tuning and modulation devices. I have spent more sleepless nights than I

care to remember to meet design deadlines working alongside a great team of circuit designers in Michael Georgas, Ben Moss, Jonathan Leu and Chen Sun.

The contribution of expertise and time from many different faculty members from Professors Hank Smith, Karl Berggren and Judy Hoyt on the fabrication front helped shape our early discussions to be more productive. Prof. Franz Kaertner led the EPIC team that kick started our efforts as well as served on my thesis committee. His insights are always helpful and he has taught me through his example that there is no problem too big to be solved by scribbling estimates of relevant physics on a scrap of paper to factor of two precision. Prof. Ippen helped clarify initial project ideas and has graciously loaned us the use of his broadband laser for years over the course of this project. Prof. Mike Watts, who is a recent addition to the team, has brought with him extensive experience in silicon photonics and I have greatly benefitted from discussions with him and his students.

I have had the great benefit of working closely with Prof. Vladimir Stojanovic, who has served as the project lead on this effort, for nearly 6 years. The rate at which he picks up new topics and becomes an expert has been fun to watch. A significant fraction of the ideas of this thesis down to the nittiest, grittiest, all-nighter-in-lab-with-the-students details have been jointly his or resulted from the continual process of bouncing ideas off of him. He has also steadfastly pursued the goal of getting this project to work one way or another.

Prof. Rajeev Ram has provided me with an ideal academic home among wide variety of different research topics in the Physical Optics & Electronics Group. I do not have a clue where else I could have had the breadth of interaction with interesting people and projects from thermodynamics to anything you can do with PDMS. Rajeev has served as a valuable advisor over the years by providing deep insight into many areas, unique opportunities and unusual freedom to take charge of my own research. He is also an extremely friendly person who has been a pleasure to get to know over my time here. Harry Lee, Peter Mayer and Tom Liptay provided the senior expertise when I joined the lab. All three helped me figure out what it meant to do research. They have also turned out to be lasting friends after graduation. Kevin Lee and Reja Amatya were great wire-to-wire lab companions with whom I shared between 1300 and 1500 lunches. Kevin managed to not accidentally kill me and along the way enhanced my love of the kludge. Reja has always been a friendly voice of reason and provided a constant example of not getting worked up over the little things in research and in life. Parthi Santhanam has added a unique viewpoint on thinking about problems both theoretical and practical that I have increasing been able to understand and appreciate over the years. Dodd Gray and Joe Summers were great colleagues to have around both in lab and over a barbeque. Karan Mehta has already proven his ability to unconfuse me on any number of topics and has been a welcome addition the project and our office.

My friends and family have provided a much needed counterweight to this thesis. There was never the lack of a worthwhile distraction when one was needed. Throughout it all, from rolling deadlines that just kept rolling to the minor victories, it was unexpected and wonderful to share my time in grad school with my wife Kelly. It has been a fun journey and far more interesting than I expected when I started at MIT back in 2005 without knowing what I was getting myself into.

# TABLE OF CONTENTS

# 1. INTRODUCTION

Over the course of this thesis, I will present a design, layout and fabrication methodology to integrate nanophotonic devices within the three largest and most important electronics manufacturing processes: bulk CMOS, silicon-on-insulator (SOI) CMOS and DRAM. Importantly, the addition of the nanophotonic elements should not reduce the performance of the integrated transistors or require specialized processing in the CMOS cases that would limit the traditional foundry model of the industry. Integration within these processes not only extends the economies of scale present in the electronics industry to the ultraprecise fabrication required for nanophotonic devices, but also enables new classes of electronic-photonic integrated circuits (EPICs). Since the frequencies of operation of the transistors within these silicon-based processes are exceeding 300 GHz, it is a suitable platform for very high speed application specific functionality that has historically only been achievable with specialized III-V semiconductor based technologies. Additionally, although the lithographic precision is comparable to state-of-the-art nanophotonics fabrication results achieved by electron beam lithography, the lithographic throughput of CMOS is seven billion times higher. This offers the opportunity to dramatically lower the cost associated with such systems.

The application area that is the focus of much of this thesis is the integration of photonic links between the processing cores on a multicore CPU fabricated in either a bulk or SOI CMOS process and the main memory chips. The chief metrics of this system are energy efficiency and bandwidth density. This technology enables the communication interface to processing elements within future computing systems to fit within the required thermal budget and physical real estate footprint [1-4]. The required components are those of a typical wavelength division multiplexing (WDM) data link: modulators, receivers, add-drop filters and coupling interfaces. For all systems considered in this thesis, the laser source will remain off-chip and will not be integrated. Designs and initial test results for all of these components will be presented. The passive components are discussed in Chapter 4 and the active components will be discussed in Chapter 5. A detailed study of the microprocessor core-to-memory link application to set device specifications as well as to further motivate this work is included in Chapter 2.

The design and fabrication challenge for monolithic integration of photonics within an existing process is to identify a suitable set of layers that provide enough flexibility to produce all of the required devices. Due to the general-purpose model of CMOS foundries, the fabrication process steps are standardized for all users. Instead, the sole flexibility is in the design of the masks used to pattern these existing layers. This constraint has required the structures used to implement all of the photonic components to be rethought and adapted to fit inside the existing process box. There are two exceptions to this general principle. First, the layer structure of the electronic processes does not allow for the formation of a sufficiently thick low index under-cladding for the photonic devices in the standard process. To address this problem, a post-process localized substrate removal technology has been developed that does not affect electronic performance or compromise the foundry model. Second, the DRAM process manufactures a single product and therefore process technology is optimized to that purpose. This specificity allows for process tweaks and additions to integrate photonic devices with optimized processes by introducing steps compatible with the existing process flow. These changes do not present the problem they would in CMOS since there is no DRAM foundry system that forces process uniformity.

Another important aspect of the development of this platform is to integrate the layout of the photonic devices within the industry standard electronic design tools. In addition to enabling co-design of the photonic and electronic blocks, photonic device design preparation that is consistent with standard electronic layout is required for the foundry to accept the design for fabrication. Since the processing mask set for a state-of-the-art electronics process cost between two and five million dollars [5-7], it is necessary for small projects to join together to share this expense. This is common practice in the electronics industry. Therefore if the photonic designs can be produced such that they are indistinguishable from electronic designs, nanophotonics can piggyback on the much larger industry. Additionally, since the processing conditions are optimized for a specific set of design parameters, geometric rules must be verified to be within an acceptable range for successful fabrication. Since the on-wafer processing is shared with other customers, compliance with these rules is mandatory. Furthermore, the complexity associated with a mature fabrication platform results in between 40,000 and 60,000 rules that must be automatically verified in a process known as design rule checking (DRC) [8-11]. Care must be taken in the layout of the non-traditional, photonic

structures to ensure compliance. The details of the computer-aided design infrastructure that was developed to enable the fabrication of photonic devices using the standard electronic design infrastructure will be presented in Chapter 3.

For the remainder of this introductory chapter, further background and information relevant to the thesis work will be presented. The first sections will review prior platforms for opto-electronic integration. Next, a process overview of the CMOS electronics processes used for photonic integration will be presented. The general approach taken for fabricating photonic devices within these processes will be discussed next. The final section of the introduction will present a brief overview of each type of photonic device required for an integrated photonic link.

## 1.1. A Survey of Electronic-Photonic Integrated Circuits

The oldest optoelectronic platforms are based upon the periodic table group III-V element heterostructures such as InGaAsP. This platform has robust functionality including optical emission devices and high speed transistors. The semiconductor materials themselves have significant advantages to silicon such as high electron and hole carrier mobilities and direct bandgaps. The direct bandgap enables photon-mediated electron-hole interactions without requiring a phonon as in the case of indirect bandgap materials such as silicon. The higher equilibrium-state absorption and excited-state gain enables the creation of efficient lasers, modulators and detectors.

Indium phosphide photonic integrated circuits (PICs) have achieved complex functionality such as 40-wavelength 40-Gbps transmitters to enable 1.6 Tbps from a single chip using hundreds of integrated components [12-14]. Engineering of the epitaxial material stackup has allowed the same III-V wafers to be used for both electronic and photonic devices [15-17]. The compromises in the epitaxial stackup design has resulted in lower maximum frequency operation of the transistors with the highest demonstrated $f_{max}$ operation of 150 GHz [16].

The chief difference from the silicon platforms that are the focus of this thesis is that III-V-based material systems are not used for large scale electronic circuits. The scale of integration is dramatically less than the billion-transistor-scale present in the silicon electronics industry. With device yields of ~95% in GaAs ICs, the circuit yields for >100 integrated transistors are unacceptable [18]. As such, it is not a suitable platform for the creation of computer systems that require the best-available integration scale [19-21]. Ongoing work for III-V logic transistor

demonstration has not enabled significant VLSI circuit production in that material system [22, 23].

The difficulties facing the high-yield production of III-V electronic integrated circuits has required electronic-photonic integrated circuits that require large numbers of integrated transistors to focus on silicon-based material platforms. Fundamentally, the first challenge is that light-emission and absorption in silicon is a significantly less efficient process than is available in III-V material systems. Some groups have tried to engineer light emission from silicon by the introduction of defect states [24, 25]. Others have focused on building lasers using materials that may be epitaxially grown on a silicon substrate [26-28]. The final option for laser source integration has been to bond direct band gap layers to the silicon wafer to integrate gain materials where required [29, 30]. For the most part, however, most silicon-based optoelectronic integration platforms have instead relied upon non-integrated laser sources as will be the case in this work.

Silicon electronic-photonic integrated circuit platforms have been demonstrated in other work by modifying the fabrication processes of older CMOS electronic foundries [31-34]. These platforms have been able to demonstrate integration scales of similar numbers of photonic devices as III-V PICs while including orders of magnitude more transistors [35]. Since the CMOS processing technology used has lagged at least five years behind the state-of-the-art, these platforms still do not provide a suitable method to fabricate photonic links monolithically integrated into microprocessors and memory.

## 1.2. Other Examples of Zero-Change CMOS Integration

To enable access to state-of-the-art CMOS processing technology, compliance with the existing process is required to leverage the economies of scale already present in the industry. Since billions of dollars are required to develop state-of-the-art CMOS processes, it is prohibitively expensive to develop customized processes. Significant research effort has already been performed with the goal of integrating non-standard devices within CMOS processes without requiring changes to the in-foundry processing. At the outset of this thesis work, the demonstrated device set was limited to photodiodes operating above the silicon bandedge [36-47]. Although impressive performance in this application space, 1.24 A/W and multi-gigabit operation at 850 nm [47], had been achieved, no waveguides, modulators or photodiodes operating below the silicon bandedge had been demonstrated. At the

time of this writing, a zero-change CMOS demonstration with a metal-clad waveguide was reported [25]. Light emission and detection were observed between two proximate diodes in the silicon substrate. The demonstrated platform differs dramatically from the work in this thesis and does not have comparable scalability.

The first successful optical integration topology that did not require modifications to the CMOS processing was the bonding of parallel optics modules of VCSELs and photodiodes on top of separately fabricated silicon VLSI chips [48-53]. The applications for the technology has varied from neural network implementation [54], to chip-to-chip interconnect [55, 56] and to short-distance data communication within servers [50]. It is unclear, however, how such an approach can be scaled to enable the scale of optical connectivity required for the computer systems discussed in Chapter 2.

Alternatively, continued deposition and patterning of optical components on top of normally fabricated CMOS chips has been demonstrated [19]. This approach provides a viable alternative to the work presented in this thesis, but does have some drawbacks. The fundamental challenge in the demonstration of optical integration in this method is that the maximum processing temperature is limited to less than 450 °C. A photodetector with dark current below 1 μA has not been demonstrated in this platform [19]. Additionally, since the inserted photonic devices do not share any processing steps with the electronic devices, the final solution may be significantly more expensive to manufacture. Other tradeoffs including increased thermal impedance are discussed in Section 1.6.

Instead, the goal of this project is to monolithically fabricate the photonic devices using the existing front-end process alongside the transistors in a state-of-the-art CMOS foundry. A similar approach has been taken by the MEMS community in prior work [57-62]. The actual devices to sense either sound or acceleration are fabricated in the standard process steps. After complete fabrication in the CMOS foundry, additional "post-CMOS" processing is often required to allow the sensor suspension. A similar approach has been followed for photonic integration and will be presented over the course of this thesis.

## 1.3. CMOS Process Overview

Due to the high cost of the tools required to produce state of the art electronics, standardization and collaboration have produced significant consistency across CMOS foundries. The customer may select from a small menu of standard options

such as the number of metal layers, presence of specific passive components and transistor optimizations. Beyond this flexibility, the process is completely fixed in production processes due to the high cost of modifications. CMOS foundries are divided into two main groups depending on the starting substrate used. In the bulk CMOS process, which represents 92% of the market, the active device layer is fabricated directly on this silicon wafer. In the SOI CMOS market, the active device layer is a separate 50-200 nm silicon thick layer that is separated from the handle wafer by a 100-300 nm thick $SiO_2$ layer. Although this structured starting wafer has a higher cost, typically $1000 as compared to a $300 bulk wafer [63], the reduced transistor capacitance and altered device physics offers a 20-35% performance increase [64].

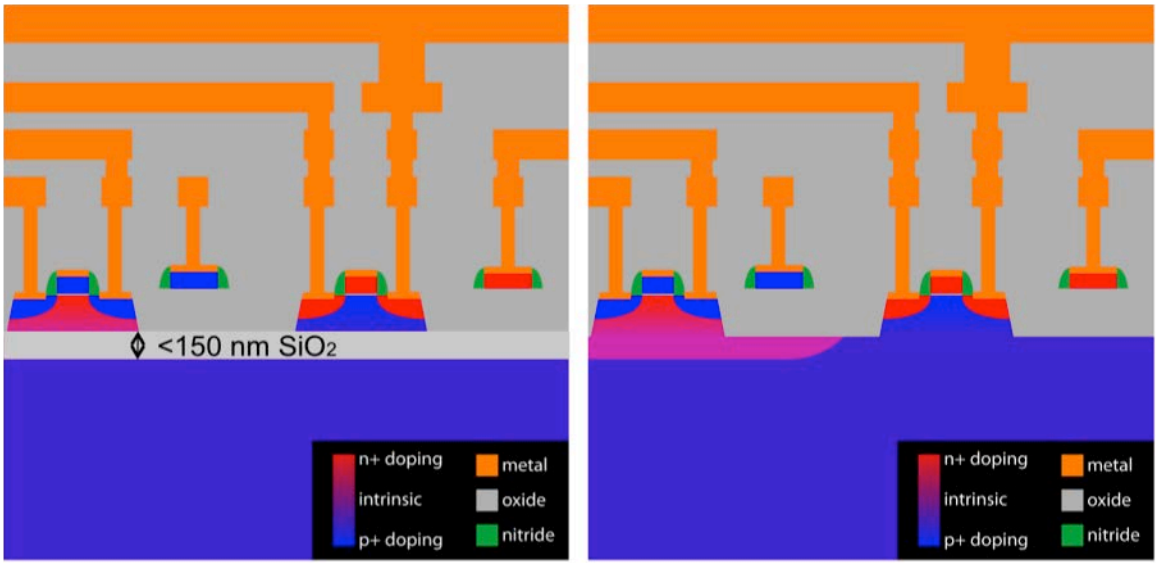


**Figure 1.1** Cross section cartoons for (a) the SOI CMOS process and (b) the bulk CMOS process.

The buried oxide (BOX) layer of the SOI wafer does present a slight tradeoff by increasing the thermal impedance of the wafer. The ability to fabricate a large number of transistors in a small area has resulted in most large circuits such as microprocessors operating in a thermally-limited regime. The total power that may be dissipated on a single chip is therefore limited to approximately 100 W today. Since the thermal conductivity of silicon dioxide is ~100 times lower than silicon, replacing 200 nm of silicon with oxide is roughly equivalent to increasing the substrate thickness by 20 μm. Considering an average silicon thickness after back-

grinding of approximately 300 μm, the addition of the SOI layer results in less than a 7% reduction in substrate thermal conductivity. Since almost 50% of the power is dissipated through the metallization instead of the substrate, the total impact of the BOX layer on the effective thermal impedance for the circuits should be less than 5%. Similar to these concerns of the impact of the SOI wafer, any modification of the CMOS process must also consider the impact on the total thermal impedance. Reductions to the effective thermal conductivity will directly impact the total power that may be dissipated on-chip and therefore reduce the total number of transistors that may be integrated together to form complex circuits.

At the highest-level, the CMOS fabrication process is divided into two halves, the front and back ends. The field effect transistors (FETs), diodes and all other semiconductor devices are fabricated in the front-end and metal interconnections are fabricated in the back-end. The chief differences between the two are the temperatures and materials used. In the front end, high temperatures, in excess of 1050 °C, are common, but the available materials are restricted to a select set that are known to yield high quality electronics. The back end processing is relatively free of material restrictions, but the maximum allowable processing temperature is restricted to approximately 450 °C to prevent diffusion of the metal layers. A brief overview of these flows, focusing on aspects relevant for photonic devices will be presented below. Further details can be found from any number of introductory electronics books [65, 66].

The field effect transistor (FET) at the heart of the front end process consists of a heavily doped polysilicon gate separated from the doped single-crystalline silicon channel by a 1-2 nm dielectric layer. The silicon channel is abutted by oppositely doped source and drain regions. A cross-sectional TEM of an advanced FET as would be found in a current microprocessor chip is shown in **Figure 1.2**. Each transistor is separated from its neighbors by a oxide-filled trench known as shallow trench isolation (STI) as shown in **Figure 1.1**. The front-end processing to define the transistors utilizes precise projection lithography to mask a large number of deposition and implant steps. The alignment between the tranistor gate and the implants that define the source/drains is controlled by a dielectric spacer. The spacer of nitride and oxide is formed by the isotropic deposition of layers that are anisotroptically etched to leave material only on the sidewall of the transistor gates. By implanting both before and after spacer formation, the complex doping profiles for advanced transistors can be realized.

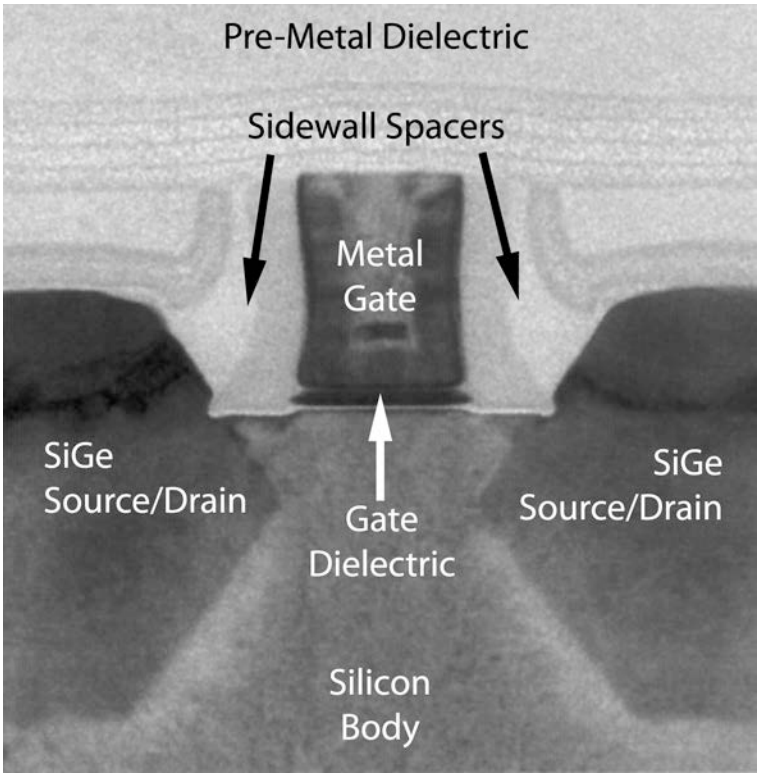


**Figure 1.2** Scanning TEM of a bulk-CMOS 45nm scaled p-type FET transistor. Source: Intel.

A relatively new addition to CMOS front end processing is the epitaxial growth of low mole fraction silicon germanium source/drains in p-type FETs. The SiGe is intended for transistor stress engineering, but also provides the photonic device designer with material that has a smaller bandgap than silicon for photodiode formation. The Ge mole-fraction is typically between 25% and 30% to produce a thin coherently strained layer that imparts compressive stress on the silicon channel to increase hole mobility [67]. The SiGe stressor regions are visible in both **Figure 1.2** and **Figure 1.3**.

The material sheet and contact resistances are reduced by the formation of a silicon-metal composite known as a silicide. Since the STI separates the transistors from each other and the sidewall spacers separate the transistor gate from the source/drains, all silicon transistor contacts are separated by dielectric layers. As a result, a self-aligned process can be used to metalize the silicon. A metal such as nickel or cobalt is blanketly deposited on the wafer. The wafer is then annealed between 500 °C and 600 °C to react the metal and the silicon. The unreacted metal is then etched away to form isolated silicide regions on all transistor contacts. The

self-aligned nature of this process has resulted in the term salicide as a short-hand for self-aligned silicide.

Next, a nitride liner is deposited to encapsulate the front-end transistors as shown in **Figure 1.3**. The first reason that the nitride layer was added to the front-end process was to provide a diffusion barrier to isolate the transistors from any contamination that may diffuse out of the back-end process. Since the stress of the layer is determined by deposition conditions, the nitride layer can also be used for further transistor stress engineering. This is typically used to provide tensile strain to the n-channel transistors to improve electron mobility.

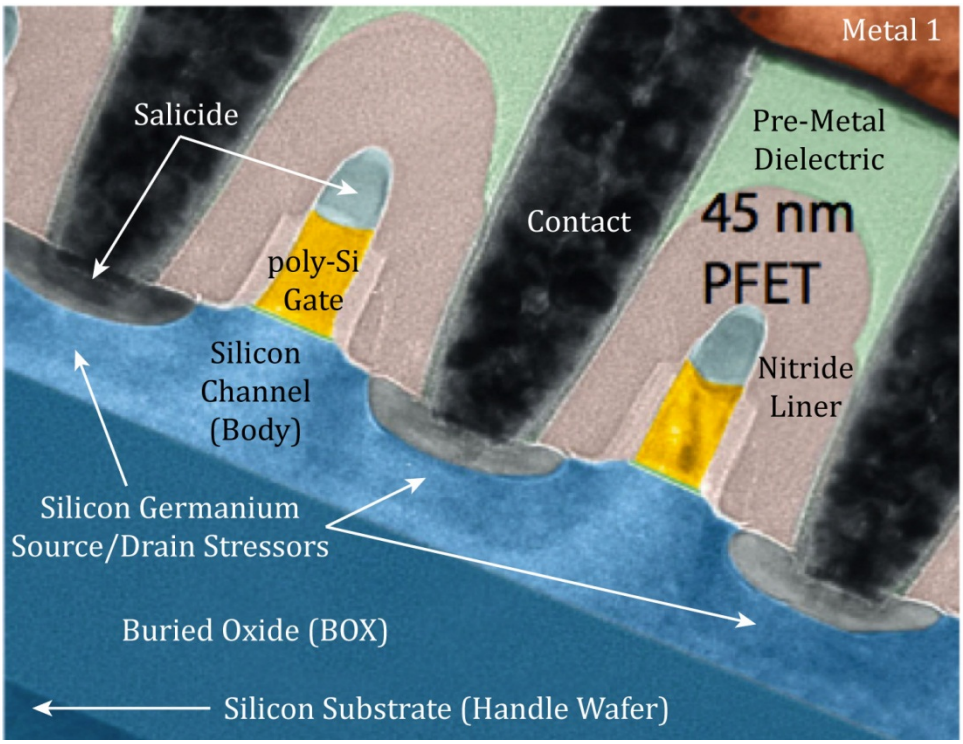


**Figure 1.3** Scanning TEM of a SOI-CMOS 45nm scaled p-type FET transistor with surrounding. Source: IBM.

Once the transistors are defined, the process progresses to the back-end processing by depositing a thick layer of dielectric, typically phospho-silicate glass (PSG). This planarizes the wafer for further processing and isolates the sensitive device layer from contamination by materials used in the back-end. At this point, narrow square plugs are etched to make electrical contact to the devices in the silicided region and filled with a refractory metal, typically tungsten. The cross-section of a finished front-end process is shown in **Figure 1.3**. These initial contact

points then need to be wired together to create the final circuit by the copper dual-damascene process. To define each interconnection layer, a dielectric is deposited and then etched to pattern the desired planar wires. Then a thin diffusion barrier metal, typically TaN, is deposited and the copper is uniformly plated. The copper is then chemically-mechanically polished to form the wires and planarize the stackup. In order to ensure uniform thickness across the wafer, strict pattern density rules are enforced such that a consistent ratio of metal to oxide is maintained. A cross-section of such a finished stack-up is shown in **Figure 1.4**. To provide the necessary data and power connections to the increasingly dense device layer, as many as 13 of these back-end wiring layers are currently required. The total interconnect thickness in modern processes is greater than 8 μm.

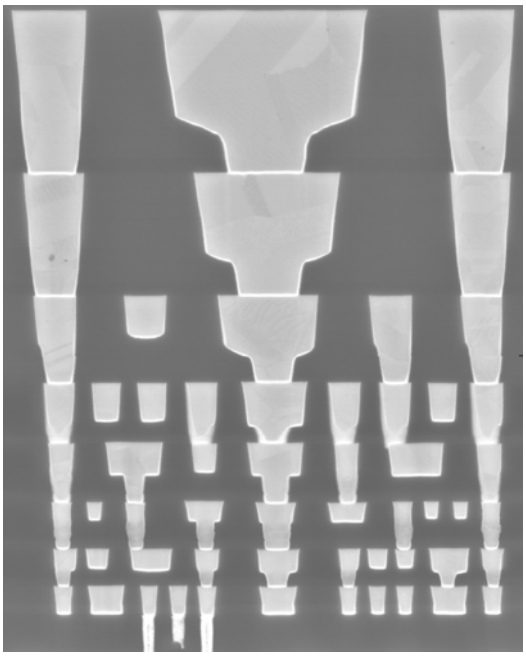

**Figure 1.4** Cross sectional scanning electron micrograph showing the interconnecting metal layers in the backend stackup of a finished microprocessor. Source: Intel.

The external interfaces to the package are then formed on top of the copper wiring in either a wirebond or area array configuration. In the wirebond case, gold wires are bonded to aluminum pads at the perimeter of the circuit to the package connections. In the area array, or flip-chip, case, C4 solder balls are deposited on top of the aluminum pads that are arranged into a 150 μm pitch array to mate to a matching array in the package. In both cases, the finished CMOS chip is encapsulated

in a 5 μm thick polymer layer consisting of some combination of BCB and polyamide with openings only over the aluminum interface pads. The final in foundry step is to cut the wafers into individual die containing only each customer's design by a dicing saw.

## 1.4. DRAM Process Comparison

DRAM memory modules are divided into two main sections: the storage array and peripheral access devices. The storage banks are comprised of a 2D arrays of capacitors isolated from many shared access wires, known as bit lines, by individual access transistors. The presence or absence of charge on the capacitors, which range in value between 15 fF and 50 fF depending on technology generation, stores either a data '1' or '0'. A single capacitor per bit line is then accessed for reading or writing data by raising or lowering the gate voltage of the connecting transistors that share a common word line across the bank. Data from all of the bit lines is then read out by sense amplifiers. These sense amplifiers and all circuitry to control the bit and word lines are the peripheral access devices located outside the storage array. Further peripheral circuitry provides data I/O to connect the DRAM to external components such as a memory controller or CPU. The peripheral circuitry resembles the bulk-CMOS process that has already been discussed. The storage array devices requires significantly different processes to fabricate the capacitors that are frequently known as containers. The total process flow must then be shared between these two regions and jointly optimized for acceptable performance for both circuit types.

Although the general process flow and required tools of the DRAM manufacturing process greatly resembles that of CMOS, there are several key differences, both technical and logistical, worth noting here. As a primary constraint, the low cost model of memory manufacturing prohibits the use of the expensive starting substrates that are required for SOI-based processes. Instead, the memory process utilizes similar starting silicon substrates to bulk-CMOS processes. Technically, high density capacitors are fabricated in the pre-metal dielectrics separating the transistors from the metal layers. The capacitors themselves and the connections to them are constructed by up to five different polysilicon layers to create a front end process of far greater complexity than that found in traditional CMOS processes as shown in **Figure 1.5**. In contrast, the back end is much simpler containing only two or three metal layers due to the simpler circuit topologies.

Finally, due to low off-current requirements, transistor performance metrics such as saturated current drive and maximum switching frequency are reduced by approximately two process generations compared to CMOS.

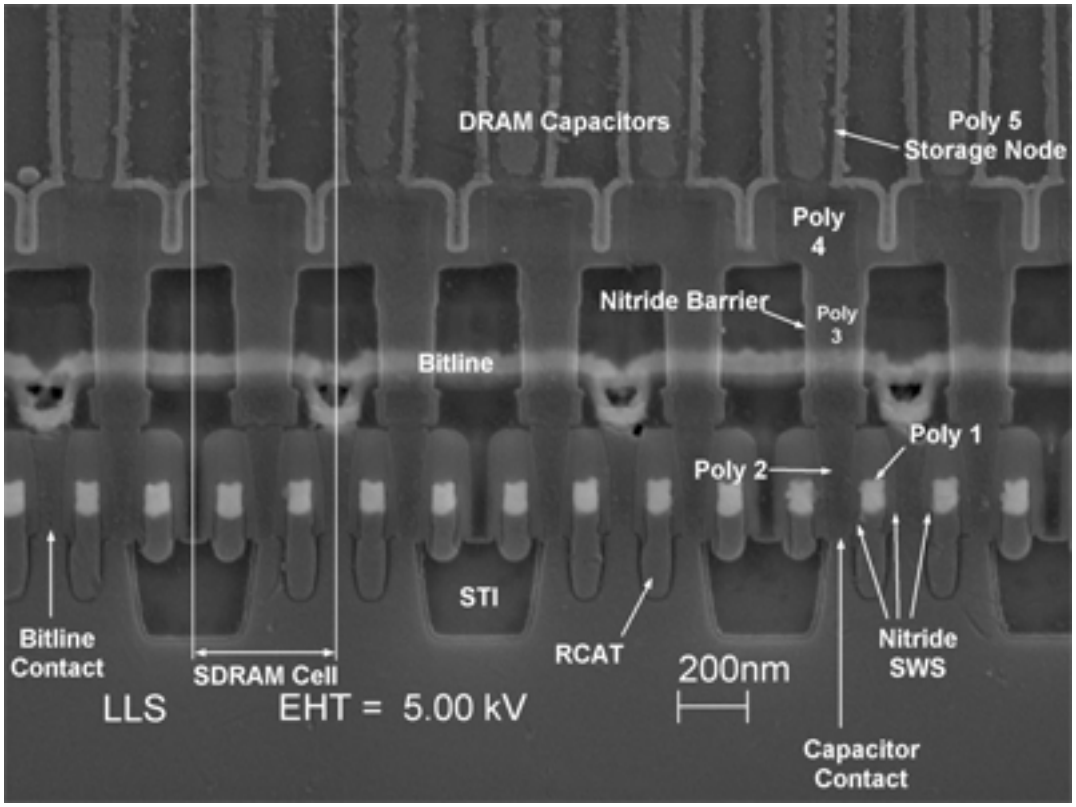


**Figure 1.5** Cross sectional scanning electron micrograph showing the Samsung 45 nm DRAM front end process with stacked DRAM capacitors above the transistor layer. Source: EE Times.

Since the business model of the memory manufacturers differs greatly from the foundry model common in the CMOS industry, the integration constraints differ greatly. First, processing masks are never shared among different projects. Second, the processing steps are customized for optimal performance of every product. These factors enable process modifications to optimize photonic devices. The chief challenges are to minimize the additional cost of integration and to prevent modifications of electronic device characteristics. Any additional processing steps required for photonic integration should then be designed as the minimum number of simple modules that can be inserted into the existing memory process without shifting transistor characteristics.

## 1.5. Standard Process Integration Methodology

For many important EPIC systems, electronic performance and density are paramount. As has been discussed, previously proposed integration platforms compromise these desirable features that exist today within state-of-the-art electronic processes. To eliminate these barriers, we have developed a monolithic integration platform that neither alters the dominant business models present in the CMOS and DRAM industries nor degrades electronic performance.

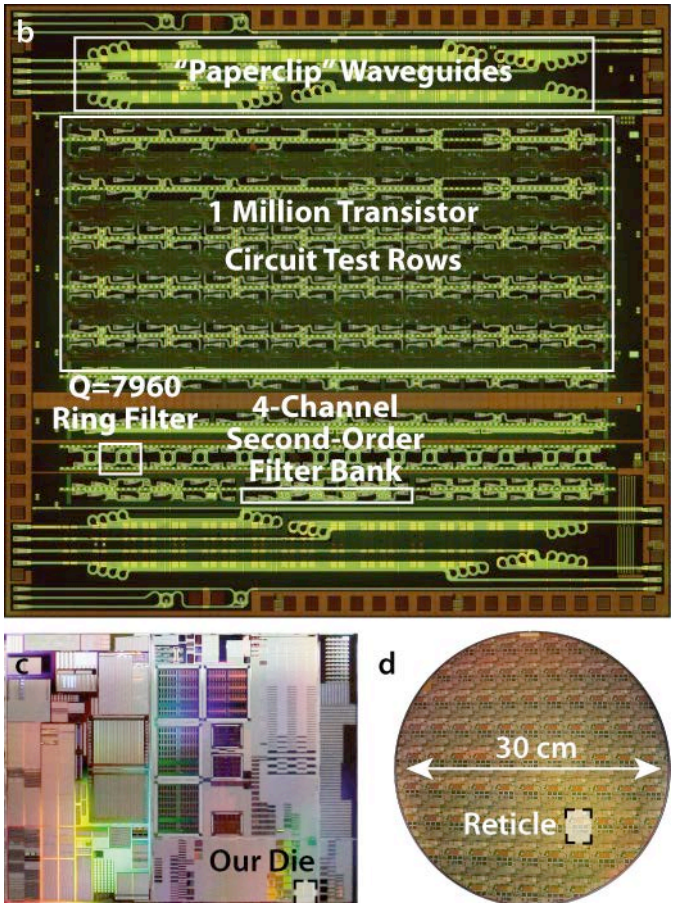


**Figure 1.6** (a) Integrated photonic-electronic die produced in a 32 nm bulk CMOS process. Die shared a (b) mask set with a full reticle of standard electronic projects submitted by other customers. All in foundry fabrication steps were performed in the standard electronic process on (c) a 30 cm wafer.

For CMOS integration, we have acted as a standard multi-project electronics customer at Texas Instruments and IBM, requiring zero in foundry process changes. Our integrated electronic photonic designs share mask sets and all in foundry

processes with standard electronics customers as shown in **Figure 1.6**. This approach allows us to leverage the existing infrastructure and economies of scale established by the $250B yearly revenue silicon semiconductor electronics industry. For DRAM integration, we have worked closely with Micron Technology to insert photonic device layers within the existing manufacturing flow that do not degrade performance and cause minimal product cost increases. In contrast to the CMOS case where the work took place on the production manufacturing lines, we have pursued a short-flow methodology with DRAM integration to test the photonic devices in a less expensive process representative of the full production line.

### 1.5.1. Waveguide Core Formation in Standard CMOS

In our monolithic integration platform, the waveguide cores are implemented in the patternable front-end silicon layers. In the SOI-CMOS process, there are two such available layers: the single-crystalline silicon transistor body layer further referred to as the body-Si layer, and the polycrystalline silicon transistor gate layer further referred to as the poly-Si layer. In the bulk-CMOS process, the transistor body is fabricated directly in the handle wafer leaving only the poly-Si layer available as the waveguide core. By configuring the surrounding material stack-up using available design layers [7], the EPIC platform cross-sections, shown in **Figure 1.7**(a) and **Figure 1.7**(b) for the SOI- and bulk-CMOS processes respectively, are available in standard electronic foundries.

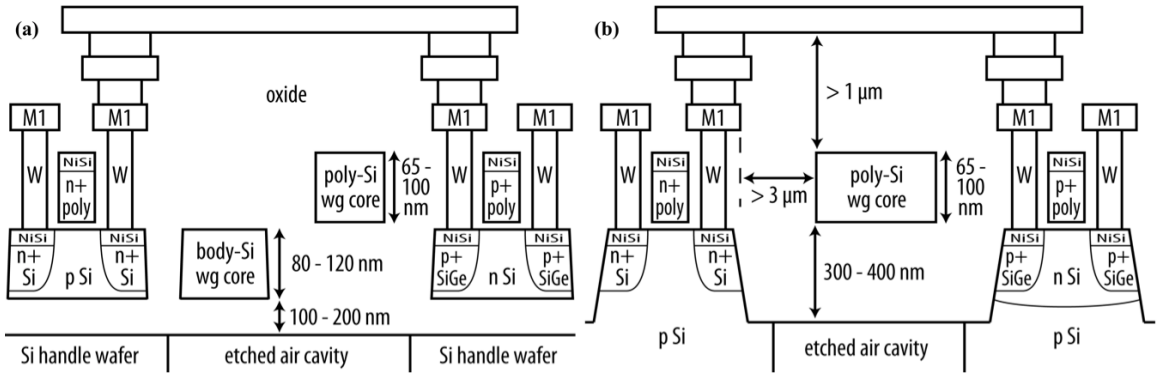


**Figure 1.7** Monolithic EPIC cross-sections integrated within (a) sub-65 nm node SOI-CMOS and (b) bulk-CMOS processes.

Utilizing the existing layers constrains the waveguide core thicknesses to values chosen to be optimal for transistor design at the current process generation. Instead of the ~220 nm thickness currently used in silicon photonic projects, the body-Si layer thickness ranges from 80-120 nm and the poly-Si layer thickness ranges from 65-100 nm depending on process generation. For passive photonic devices, this limits the minimum allowable bend radius for a given operating wavelength. SOI processes offer the designer flexibility to vertically stack the body-Si and poly-Si layers, separated only by a nanometer-scale oxide, for tighter bends where necessary. Results demonstrating high performance passive photonic devices fabricated in this platform will be presented in Chapter 4.

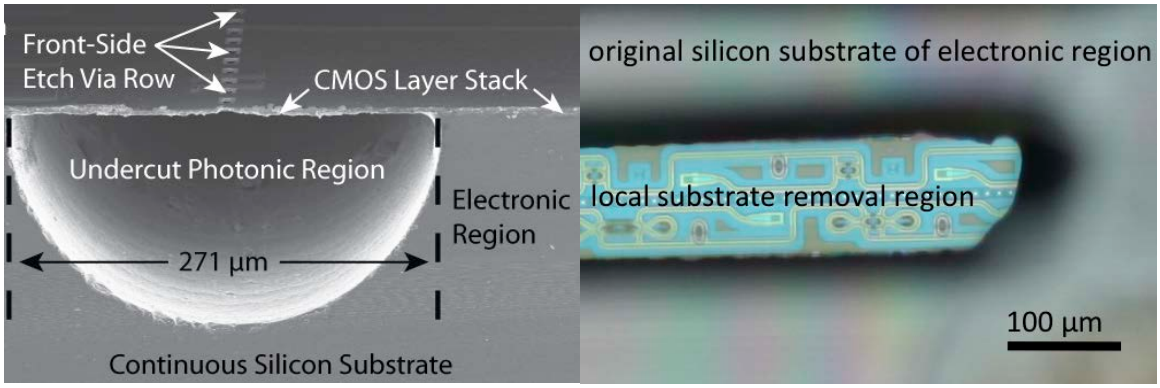


**Figure 1.8** (a) Cross-sectional scanning electron micrograph (SEM) of a 28nm bulk-CMOS die after localized substrate removal in the photonic region [70]. Total undercut width of 271 µm is more than 3x the largest design width to demonstrate film stability and planarity. (b) Optical micrograph of back-side etched local substrate removal region of a 45nm SOI-CMOS die.

### 1.5.2. Localized Substrate Removal

The oxide layers below the polysilicon layer in a modern bulk process, known as the shallow trench isolation (STI), and the BOX below the single crystalline silicon layer in a modern SOI process are all thinner than 300 nm. These thin oxide layers would cause propagation loss greater than 500 dB/cm due to the leaky optical mode [68]. To prevent this problem without process modification or the use of specialized starting substrates, a scalable, post-process front side localized substrate removal technology has been developed [69]. The end result of this process is an air pocket under the photonic regions and an unmodified local environment in the electronic regions as shown in **Figure 1.8**. Alternatively, back-side local substrate removal can

be used achieve the same end goal as also shown in **Figure 1.8**. Further details on required post-processing techniques are provided in Appendix B.

### 1.5.3. DRAM process integration

The flexibility of the process development available within DRAM enables a slightly different process integration methodology. Similar to a bulk-CMOS process, the polysilicon used for the transistor gate formation is used as the high index waveguide core. Additional process modules can be added to aid in device creation. For example, instead of requiring local substrate removal, deep trench isolation may optionally be added to the process to eliminate the requirement of further post-processing to enable optical functionality. Further functionality for device creation can be added by adding a partial etch to enable two thicknesses of the waveguide core. A basic set of the required photonic devices that must be integrated in the process is shown in **Figure 1.9**.

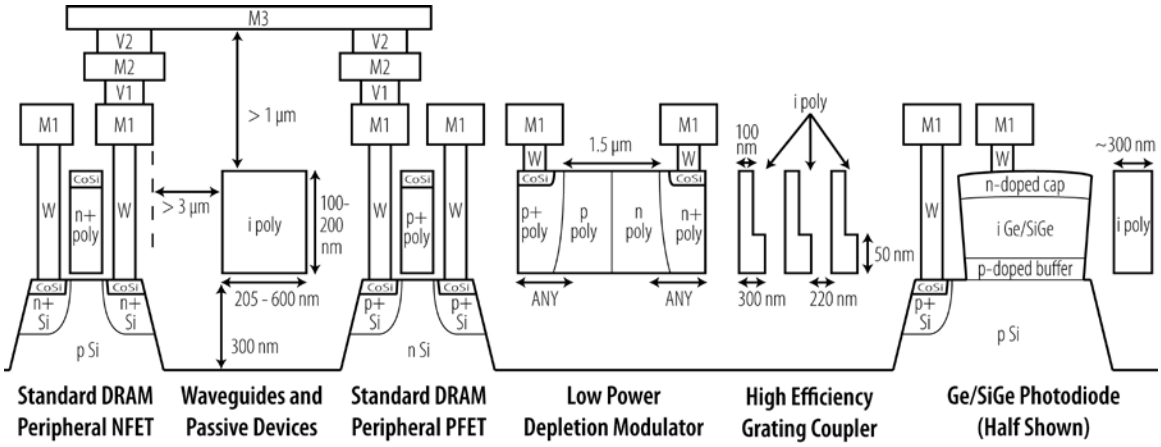


**Figure 1.9** Monolithic DRAM photonic cross-sectional integrated device proposal.

To test the required processes and demonstrate monolithic integration, a modified memory process has been developed by adding photonics-specific processing modules into an existing product manufacturing line. To simplify the fabrication during process development, all steps associated with the memory array process are not part of the integration flow. Instead, only the steps associated with the peripheral transistor process are explicitly performed. A technology computer aided design (TCAD) simulation of the complete proposed process is shown in **Figure 1.10**.

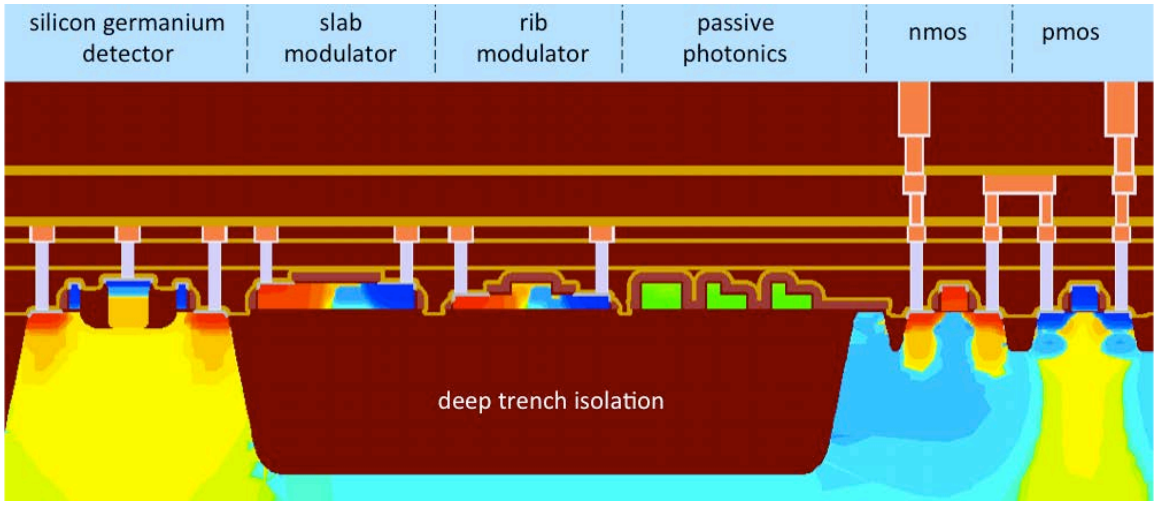


**Figure 1.10** Monolithic DRAM integration process TCAD simulation.

The requirement to develop a new fabrication process requires dedicated masks instead of being able to share masks with other projects as is in a CMOS foundry. The design size on silicon has ranged from 20mm x 22mm to 26mm x 33mm. The dramatically increased area relative to the ~9mm$^2$ CMOS test chips enables a more thorough study of process parameters as well as parametric design variation. An example project consisting of a complete reticle set and resulting processed wafer is shown **Figure 1.11** in contrast to the foundry integration work shown in **Figure 1.6**.

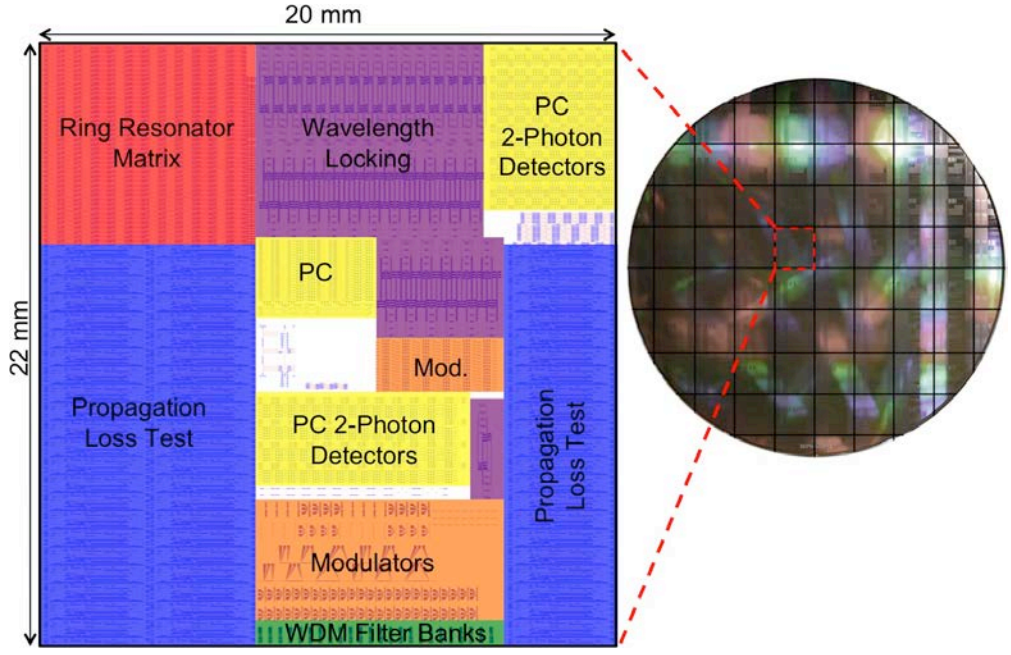


**Figure 1.11** Labeled design environment screen shot from the design environment and finished wafer identifying the stepped reticle fields.

## 1.6. Alternative Approaches to Electronic-Photonic Integration

Due to the widely-recognized value of integrating silicon-based electronic and photonic devices, several alternative integration platforms have been previously proposed in the literature. These approaches can be broken down into three main subgroups that I will address individually.

### 1.6.1. Thick-SOI

The first approach is to modify the starting substrates and in foundry processes to integrate photonic devices within the CMOS front end [33, 68, 71-79]. Successful implementations of this approach include the modified 130 nm SOI-CMOS process line at Freescale that was developed in cooperation with Luxtera [31, 80-82]. The chief drawback with any approach that requires specialized processing is that modifications isolate photonics process development from the standard electronics foundry infrastructure. This requires the photonics developers to fully support the cost of the process line as well as individual production masks. The process development work to modify the wafer-level fabrication steps then occur not only the non-recoverable engineering cost of the modifications, but also an opportunity cost of occupying a fabrication line that may be used for high-volume manufacturing of electronic circuits.

To date, front-end monolithic integration has required non-standard silicon-on-insulator (SOI) starting wafers in which the buried-oxide (BOX) thickness is an order of magnitude thicker than is used in electronic SOI-CMOS processes. The resulting cross section is illustrated in **Figure 1.12**. The thicker BOX may degrade the performance of nanoscale transistors via drain-induced barrier lowering (DIBL) [83, 84]. Additionally, the thermal impedance of the thicker BOX impacts the circuit performance and requires the recalibration of many wafer-level processes [85]. Increasing the buried oxide thickness from under 200 nm to above 2 μm increases the total thermal impedance of the substrate by an order of magnitude. First of all, the higher thermal impedance reduces the allowable power dissipation per unit area of the integrated circuits for a maximum transistor operation temperature. This may limit the scale of the integrated circuits as many important application areas such as microprocessors operate in a thermally-limited regime today [86, 87]. Second, the higher thermal impedance increases the thermal time constant for the individual transistors. This would therefore increase the relative importance of

thermal history effect and increase the difficulty of designing reliable circuits [88]. Finally, the increased thermal impedance may require a recalibration of on-wafer processing steps [89]. A hybrid approach that utilizes non-uniform buried oxide thickness across the wafer may resolve some to these constraints that currently limit the usefulness of such a platform [90]. Although this platform has been demonstrated in an academic setting, further process development is required in an industrial setting as this approach requires a radical change to the initial front-end process of a CMOS fabrication line.

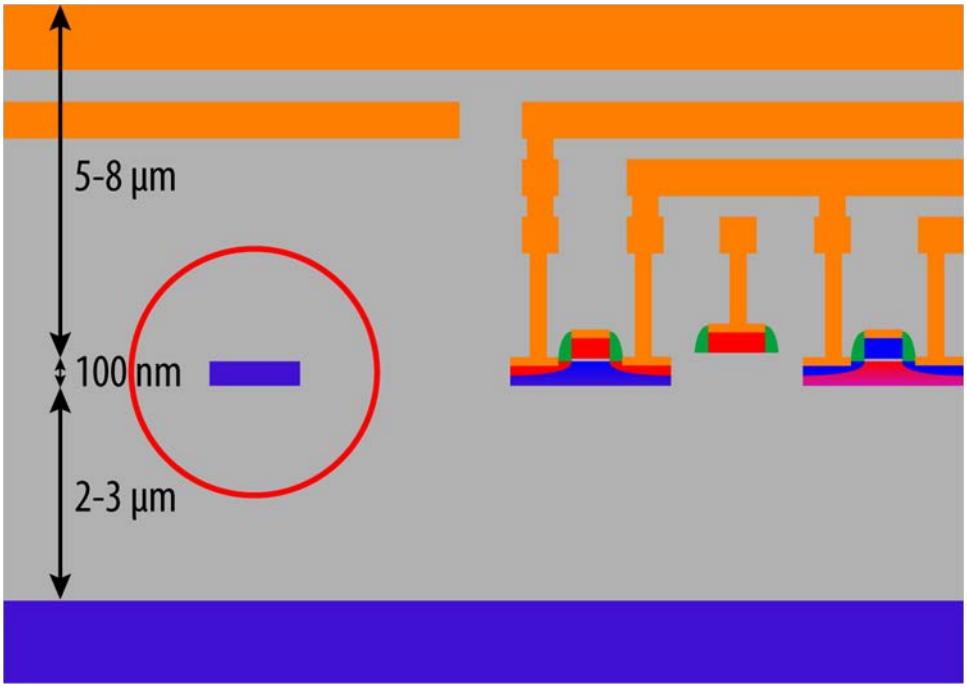


**Figure 1.12** Cross-section schematic illustrating the thick-SOI CMOS platform.

Many high-performance silicon photonic devices have been demonstrated in a thick-SOI monolithically integrated platform. Integrated waveguide loss in the single-crystalline silicon layer of 1.7 dB/cm has been reported [75]. Epitaxially grown germanium photodiodes have demonstrated 0.85 A/W with a 26 GHz bandwidth [91]. The lowest power reported ring-resonator modulator achieved ~3 fJ/bit efficiency at 12.5 Gbps operation [92].

### 1.6.2. Backend, On-CMOS processing

The second approach is to introduce new steps into the backend of the CMOS process to fabricate photonic devices on top of the existing stackup as shown in **Figure 1.13** [19, 93-97]. This offers some benefit for standard foundry compatibility as some process development for this approach may be done as post-processing on existing electronic designs. Additionally, no processing modifications are required in the performance critical front end of line as in the previous approach. However, long-term usefulness of such a platform still remains in question. Since no processing is shared with the existing electronic devices, many additional steps will be required, potentially increasing the product cost. This additional processing needs to also reverse the standard CMOS trend of reduced backend precision. Beyond a convenience issue of reducing process complexity, the lithographic imaging environment far from the silicon wafer surface is degraded by the topology and non-uniform reflections introduced by the underlying material stackup. This may present a significant technological barrier to manufacturing fabrication sensitive nanophotonic devices with high yield in this manner. Additionally, the thermal budget of the electronics may be reduced since this approach still requires the formation of additional thick layers in the top thermal path. Finally, this approach would still represent a significant process customization that would require a large photonics user base to support the development and share mask costs.

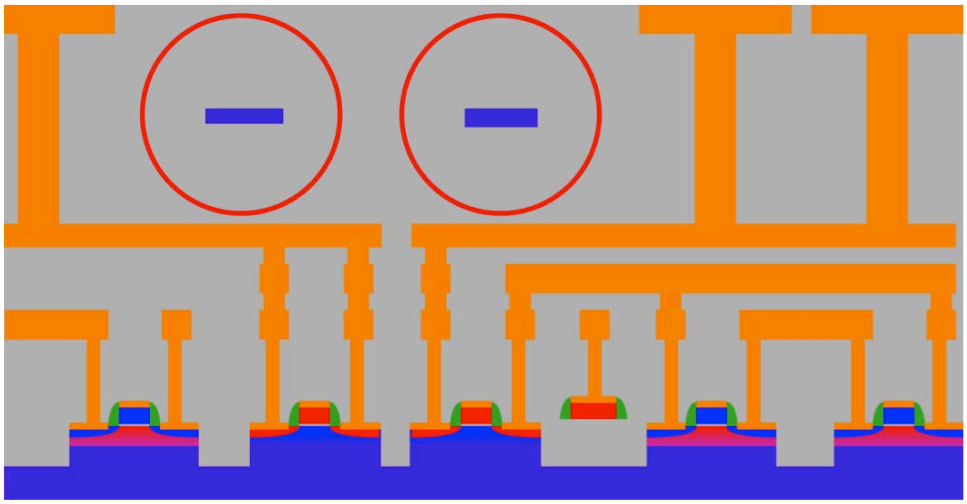

**Figure 1.13** Cross-section schematic illustrating the backend integration platform.

The backend photonic platform is composed of the least mature set of photonic devices. All wafer-level processes must have peak temperatures below 450 °C to prevent diffusion of the back end metal layers. This constraint places strict limits on allowable processing technologies. Either low temperature nitride or amorphous silicon waveguides are available. Amorphous silicon waveguides with ~1.2 dB/cm propagation loss have been demonstrated [98]. Low temperature nitride waveguides have achieved similar waveguide losses [99, 100]. In addition to silicon-based modulators, electro-optic polymer modulators are also available since the thermal budget of the process is dramatically reduced. Nitride ring resonators with 8 dB extinction ratios at 20 Gbps that require 6 V drive signals have been demonstrated [100]. Assuming that laser-crystallization can enable polysilicon modulators, the best literature results include a 2.5 Gbps modulator with 10 dB extinction and 950 fJ/bit energy dissipation [101]. The biggest missing component is a suitable photodetector. Defect-state polysilicon resonant photodetectors achieving 0.15 A/W may be integrable [102]. Otherwise, polycrystalline germanium MSM Shockley-barrier photodetectors are an available option, but suffer from large dark currents. The lowest leakage current reported for such devices is 77 μA [103].

### 1.6.3. Hybrid Integration

The final major proposed approach for photonic device integration is a modification of the thick-SOI platform. Since the modifications required to integrate photonics in the CMOS front end reduce the density and performance of integrated electronics, a layer stacking approach in which a separately fabricated largely photonic layer would be bonded to the standard state-of-the-art high density electronics CMOS layer as shown in **Figure 1.14** [104-108]. Two variations of this approach are differentiated by whether or not the photonic layer includes drive transistors or relies on transistors in the standard CMOS chip. If drivers are not included in the photonic layer, the additional parasitic of 20-25 fF in state-of-the-art face-to-face bonding demonstrations [107] would roughly triple the intrinsic capacitance of the modulator and photodiode components [108] and decrease link energy efficiency. If drivers are included in the photonic layer, the two layers could efficiently communicate with each other, as well as perhaps additional electronic layers bonded in the stack, through copper vias fabricated using 3D integration technology. Both approaches eliminate the high cost of modifying state-of-the-art electronics processes. However, as a result, the photonics designer no longer has

access to include such state-of-the-art transistors in the photonic layer. Additionally, the presence of the additional layer bonded in the top thermal path of the electronics reduces the allowable thermal budget even more than the backend integration approach. The largest problem with this approach is still the high costs associated with 3D integration which result from several well known issues: the separate fabrication of the two active layers, the additional complicated equipment and processing required for the bonding, and the coupling of the yield of the two chips and the bonding process.

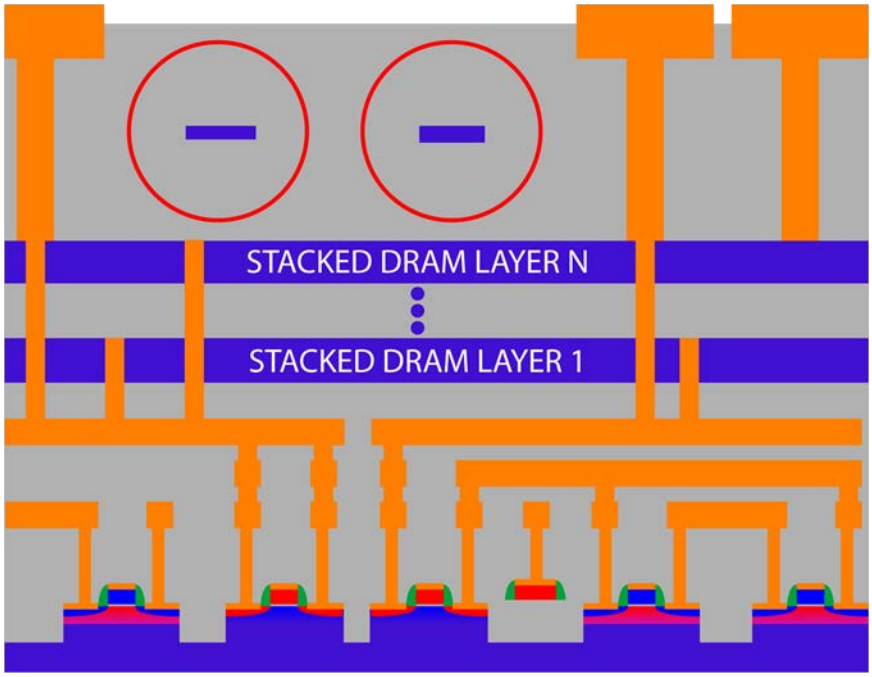


**Figure 1.14** Cross-section schematic illustrating the 3D integration platform.

Since the 3D integration platform enables separate optimization of the photonic components from electronic functionality, best-in-class photonic results are available. Waveguide losses in shallow-ridge-etch silicon waveguides are as low as 0.274 dB/cm [80]. Ring resonator modulators achieve 12.5 Gbps operation and enable 135 fJ/bit energy efficiency with hybrid-integrated CMOS drivers [109]. Integrated germanium photodiodes achieve >20 GHz bandwidth with 1.05 A/W at 1550nm [110]. These results that were all demonstrated in similar waveguide cross-sections at Kotura represent the highest performance photonic platform available.

The electrical interface to all of these photonic components must also be considered as well. To accurately compare any interconnect topology, the total link from clocked, logic-level latch on one chip to clocked, logic level latch on the other must be considered. Further details on the implications of this requirement and introduction to the circuit design concepts required to understand the tradeoffs will be discussed in Chapters 2 and 5. For initial discussion of integration approaches, it is sufficient to first generally consider the important aspects of the connections between the photonic and electrical devices. For devices that require the driving voltage to switch billions of times a second, as is the case for the modulators considered for this thesis, the total capacitance of the switching node largely determines the energy dissipation. As such the connection parasitics, or stray wiring capacitance, must be minimized. To first order, this means that the physical distance between the driving circuits and active photonic components must be minimized. Further, this same stray capacitance directly impacts the sensitivity of the photoreceiver circuit. Since the photodetector transduces the absorbed photons into current, the initial signal to be sensed is a finite charge per data bit. In CMOS circuitry, the measureable signal can always be expressed as a voltage, or voltage difference on specific transistor gates. Since the transistor gate, as well as the wiring parasitics and the photodetector itself, can be modeled to first order as a capacitor, minimizing the total capacitance maximizes the effective signal resultant from the received optical bit. The stray capacitance therefore impacts the net sensitivity of the optical link as well as the energy efficiency. These constraints therefore dictate that the magnitude of parasitics present at the connections between the electronic and optical devices is an important figure-of-merit for an integrated optical link fabrication platform.

## 1.7. Integrated Optical Components

Waveguides, fiber couplers, filters, modulators and detectors have all been demonstrated over the course of this thesis. No effort towards integrated laser sources was undertaken or is anticipated as a system need for interchip communication. Since optical fibers are connected to provide the chip-to-chip communication link, optical power delivery can be provided from an off-chip source. There appears to be no technical benefit for laser source integration in such a system. Although the fiber coupling losses may be avoided on-chip, higher efficiency may be achieved off chip while avoiding numerous technical hurdles including

cavity isolation, power delivery and thermal stability of an integrated source. In contrast, other systems such as tightly integrated microwave photonic circuits or intrachip-only photonic links that do not explicitly optical fiber connections to the electrical die may benefit dramatically from such laser integration.

The nominal operation wavelength for integrated photonic devices will be focused in the 1.2 μm to 1.3 μm window for use with the integrated silicon germanium photodetectors. For the primary system application of photonic links in a processor to memory network, there is no requirement to adhere to standard telecommunication bands near the fiber loss minima of 1.55 μm. Further, from a purely technical laser perspective, it is in fact preferable to operate at shorter wavelengths. In nearly every parameter – threshold current [111], output power [111], efficiency [112], temperature dependence [111, 113], and reliability [114] – better laser performance has been achieved near 1.3 μm as compared to 1.55 μm. Scaling to yet shorter wavelengths is limited by the silicon bandgap of 1.12 eV at which point low-loss silicon-core waveguides are impossible. Many photonic devices in this work are also designed for the 1.55 μm window due to the availability of higher quality test equipment at that wavelength in our labs. Additionally, since the scaling of various phenomena such as propagation loss as a function of wavelength provides insights to the underlying physics, device performance has been characterized over a wider wavelength range than is necessary to implement a given system.

### 1.7.1. High-Index-Contrast Waveguides

At a basic level, the integrated waveguides consist of an undoped, unmetalized silicon or polysilicon strip that is surrounded by at least a micron of low-index, low-optical-loss material in all directions. Total internal reflection at the interface of the high-index silicon and low-index cladding such as silicon dioxide provides optical confinement. The geometries and materials for transistor formation, combined by the localized substrate removal post-processing where required, provide the necessary flexibility to remove extrinsic loss sources such as metals and other high-index layers from the proximity of the well localized mode.

Fundamentally, the waveguides formed for this work strongly resemble the silicon photonic waveguides that have been used by many research groups in previous work. The main differences in this work are the exact silicon layer thickness and the operating wavelength. Most silicon photonic work has previously

relied on 200-300 nm silicon layer thickness and operated at a wavelength of 1550nm. In this work, the operating wavelength is significantly shorter, between 1200nm and 1300nm and the layer thicknesses for integrated waveguides have ranged from 65nm to 225nm. DRAM emulation process, the target polysilicon thickness of approximately 200nm matches well with previous silicon photonic work. For the scaled-CMOS processes, the layers used to fabricate the transistors are significantly thinner.

To date, the best published propagation loss values for similar strongly-confined silicon waveguides formed by photolithography and dry-etching have remained in the 2 dB/cm to 3 dB/cm range [68, 71, 115]. Since the dominant loss mechanism is the sidewall scattering of the waveguides, specialized techniques such as the masked oxidation of silicon to form the waveguides has resulted in waveguide losses down to 0.3 dB/cm [116, 117]. Alternatively, shallow ridge etches using standard processing techniques have also demonstrated reduced waveguide losses down to 0.2 dB/cm [118].

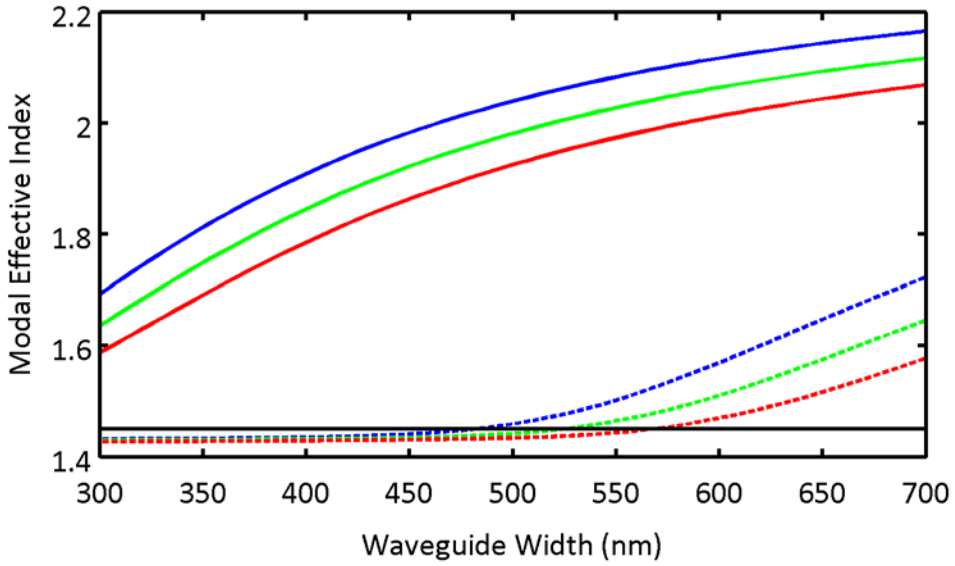


**Figure 1.15** Mode solving simulations for the first-order (solid lines) and second-order (dashed lines) modes for wavelengths of 1200nm (blue), 1250nm (green) and 1300nm (red). The second-order mode's effective index is compared to the oxide cladding with a 1.45 refractive index.

To examine waveguide characteristics in the thin-core silicon layers, we will focus on the geometry found in partially-depleted silicon-on-insulator (PD-SOI)

CMOS processes. Using layer thicknesses for such a standard wafer from SOITEC, 145 nm BOX and 80 nm silicon thicknesses, the properties of a strongly-confined singlemode waveguide can be simulated as is discussed in further detail in Chapter 4. First, the effective indices of the first and second order modes are simulated as a function of waveguide width as shown in Figure 1.15. Since the boundary conditions of the simulation domain artificially permit higher order modes when they would be physically forbidden, the cutoff of the second-order mode is taken to be the intersection of the solved effective index with that of the n=1.45 oxide cladding.

For the described layers of the PD-SOI wafer, a silicon core size of 80nm by 500nm represents the strongest confined singlemode waveguide in the 1200nm to 1300nm wavelength range and will be used for the purposes of generic integrated device simulation for the rest of this thesis. The resulting optical mode, shown in Figure 1.16, confines the guided light to a small cross-sectional area of 0.159 μm$^2$. The small mode area is advantageous for high-density on-chip routing, but does increase the relative importance of non-linear phenomena. The optical power that is able to be guided by such an integrated waveguide is therefore limited to approximately 30mW.

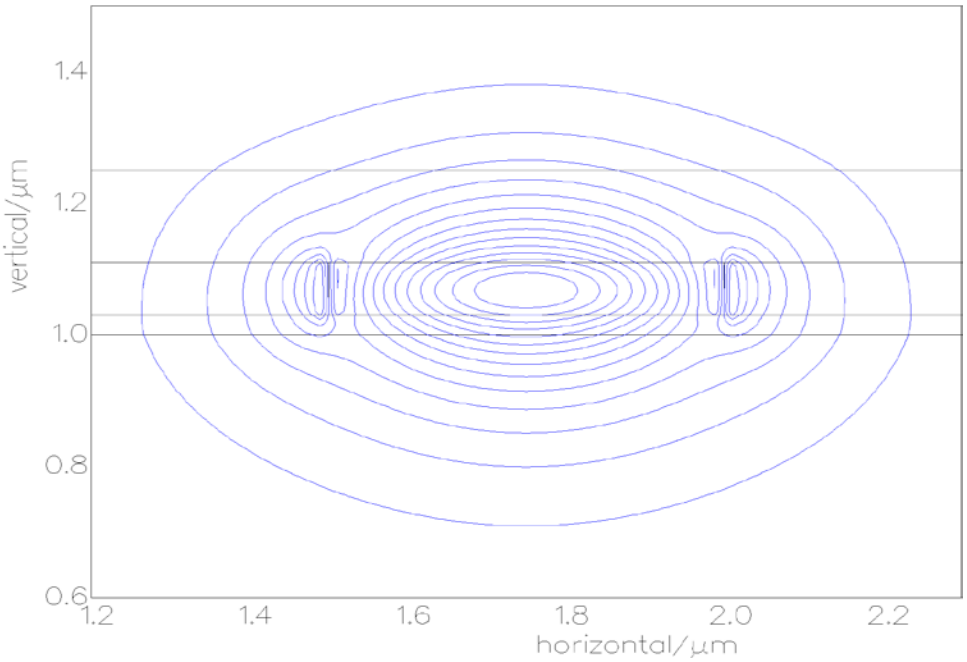


**Figure 1.16** Optical mode simulation for 500nm wide single mode waveguide with an effective index of 1.98 and a group index of 3.4 at 1250nm. Optical mode area is 0.159 μm$^2$ with a fill factor of 41% and a confinement factor of 70%.

The strongly-confined mode of such a high-index-contrast waveguide results in the increased importance of other linear phenomena that are often neglected in low-index contrast photonic devices. Due to the high index contrast difference between the core and cladding materials, the effective index is a strong function of the relative confinement between the silicon and oxide regions. Since this confinement can be changed by either the wavelength or a change in dimensions, in addition to the exact properties being strongly dependent on fabrication, the waveguides are strongly dispersive. The effective geometric dispersion is also maximized in the strongly-confined single-mode region of greatest interest for the integrated photonic devices. This can be schematically understood by plotting the modal wavevector as a function of optical frequency as shown in Figure 1.17. As the waveguide enters the strong-confinement region as a function of frequency, the modal effective index changes from following the cladding dispersion relation towards following the core dispersion relation.

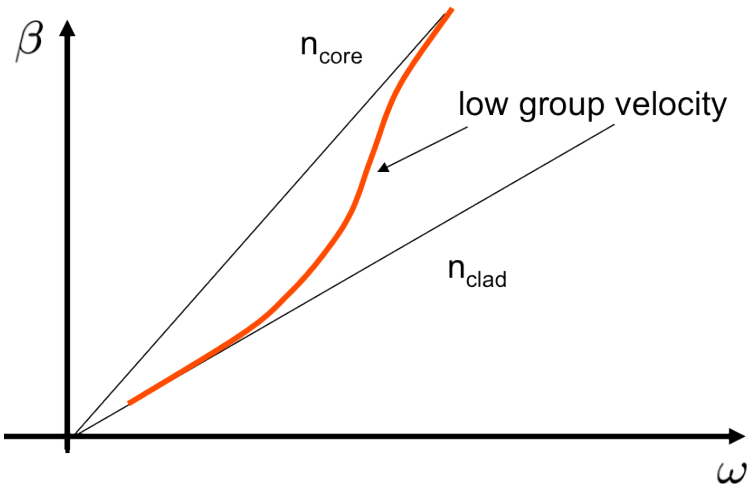


**Figure 1.17** Dispersion diagram for an optical mode of a strong-confinement waveguide.

The wavevector dispersion relation as a function of electro-magnetic field frequency, β(ω), determines several important waveguide characteristics. First, the phase velocity is given by:

$$v_\phi = \frac{\omega}{\beta} \tag{1.1}$$

Since the modal effective index is then the speed of light in vacuum divided by the phase velocity, this strong dispersion is evident in the simulated modal effective

indices for the various wavelengths of the 500nm wide waveguide mode shown in Figure 1.15. The velocity of power transfer may then vary from this phase velocity in such a dispersive system. Since the single-frequency representation of the electromagnetic field is non-causal, the power transfer velocity can be obtained by considering the velocity of a wave packet. The envelope function of the wavepacket then processes at the constructive interference condition of the multiple frequencies that compose the pulse. This velocity of power transfer, known as the group velocity, is then given by the inverse of the slope of the dispersion relation:

$$v_g = \frac{d\omega}{d\beta} \tag{1.2}$$

The geometric dispersion of the high index contrast silicon waveguide therefore causes low group velocity over a specific wavelength range for a given core sizing. The core sizing resulting in a modal effective index midway between the core and cladding refractive indices corresponds to the point of strong-confinement for a single supported mode. Since this is the intended operation point of the integrated photonic devices, it is worth further considering the physical implications of this low group velocity. In agreement with intuition for non-dispersive systems, a linear dispersion relation causes the group velocity to be simply equal to phase velocity. To simplify further equations, it is helpful to introduce the definition of the group index $n_g$ as the ratio of the speed of light in vacuum to the group velocity. The group index may then be defined by expanding derivative of the dispersion relation in terms of the refractive indices:

$$n_g = c\frac{d\left(n_{eff}\,\omega/c\right)}{d\omega} = n_{eff} + \omega\frac{dn_{eff}}{d\omega} \tag{1.3}$$

$$n_g = n_{eff} - \lambda\frac{dn_{eff}}{d\lambda} \tag{1.4}$$

The most important consequence of this low group velocity region for waveguide propagation is the relative increase of importance of material absorption in regions with significant modal electric field overlap. Since the material absorption is governed by a transition rate in the presence of the electric field, the propagation loss per unit length of a waveguide mode is then determined by the velocity of the mode and the electric field overlap with the lossy material. It is helpful to be able to define the modal loss rated as the product of the isolated material loss ($\alpha_A$) for region A and an effective linear confinement factor ($\Gamma_A^{(1)}$) for that region:

$$\alpha_{eff} \approx \Gamma_A^{(1)}\,\alpha_A \tag{1.5}$$

Using a variational method perturbation of the waveguide mode the full solution of the confinement factor in the absence of material dispersion is derived [119]:

$$\Gamma_{A}^{(1)} = \frac{c\,\varepsilon_0 \iint_{A} n(s)\cdot\left|E(s)\right|^2 ds}{\iint_{\mathbb{R}^2} \mathrm{Re}\left\{E(s)\times H^*(s)\right\}\bullet \hat{e}_z\, ds} \tag{1.6}$$

This equation is tedious to evaluate and is typically further simplified for low-index contrast systems by assuming that the integrals are evaluated in nearly homogenous materials. The homogenous media assumption allows the H-field to be trivially defined by the electric field to simplify the denominator:

$$\iint_{\mathbb{R}^2} \mathrm{Re}\left\{E(s)\times H^*(s)\right\}\bullet \hat{e}_z\, ds \approx \sqrt{\frac{\varepsilon_0}{\mu_0}} \iint_{\mathbb{R}^2} n(s)\cdot\left|E(s)\right|^2 ds \tag{1.7}$$

By then recognizing that the speed of light can be rewritten as $1/\sqrt{\varepsilon_0\mu_0}$, Equation 1.6 can be rewritten in an approximate form:

$$\Gamma_{A}^{(1)} \approx \frac{\iint_{A} n(s)\cdot\left|E(s)\right|^2 ds}{\iint_{\circ^2} n(s)\cdot\left|E(s)\right|^2 ds} = \left(\frac{n_A}{n_{eff}}\right)\frac{\iint_{A}\left|E(s)\right|^2 ds}{\iint_{\circ^2}\left|E(s)\right|^2 ds} \tag{1.8}$$

$$\Gamma_{A}^{(1)} \approx \text{fill factor} = \frac{\iint_{A}\left|E(s)\right|^2 ds}{\iint_{\mathbb{R}^2}\left|E(s)\right|^2 ds} \tag{1.9}$$

The approximate form of the confinement factor is often presented without the explicit mention that this is an approximate form that is only valid in low-index contrast systems. It is not valid for the integrated waveguides presented in this work and is instead referred to as the fill factor of the waveguide as a pure electric field overlap for the region. The relative velocity change in the waveguide mode as compared to the bulk material can be corrected for by multiplying Equation 1.8 by the ratio of the group index to the refractive index of the region:

$$\Gamma_{A}^{(1)} = \left(\frac{n_g}{n_{eff}}\right)\frac{\iint_{A}\left|E(s)\right|^2 ds}{\iint_{\mathbb{R}^2}\left|E(s)\right|^2 ds} \tag{1.10}$$

The resulting correction term of the group index divided by the effective index matches the 80nm by 500nm waveguide simulation in which the ratio of the group index of 3.4 to the effective index of 1.98 matches the ratio of the confinement factor of 70% to the fill factor of 41%. The large difference between these values

demonstrates the importance of considering the correct form of the confinement factor for loss analysis in such strongly-confined high-index contrast waveguide systems.

In addition to the corrections for the linear absorption in high-index contrast systems, the nonlinear processes must be considered as well. The fundamentally non-linear process of interest is two-photon absorption. Although the bandgap energy of the semiconductor is larger than the photon energy, band-to-band carrier excitation can still occur by the simultaneous absorption of two photons. Since the guided optical power is confined to a very small area, the resulting intensity increases the relative importance of such processes. In a free-space nonlinear media, the two-photon process can be added to the traditional linear absorption process for plane-wave power loss by the addition of a two-photon absorption coefficient ß$_{TPA}$:

$$\frac{dI}{dz} = -\alpha I - \beta_{TPA} I^2 \tag{1.11}$$

The magnitude of this coefficient is a weak function of wavelength with values of 1.8 cm/GW at 1200 nm and 1.5 cm/GW at 1550 nm for silicon. The two-photon absorption coefficient can be significantly larger than the literature silicon data in polysilicon where the grain boundary defects can effectively raise this value to ~310 cm/GW [120].

The power in the mode can then be related to the intensity by defining an effective area. The absorption of power in a waveguide mode then depends on the effective electric field overlap with the nonlinear or absorptive media.

$$A_{eff} = \frac{\left( \iint_{\mathbb{R}^2} I(s) ds \right)^2}{\iint_{\mathbb{R}^2} I(s)^2 ds} \tag{1.12}$$

$$\frac{dP}{dz} = -\alpha_A \Gamma_A^{(1)} P - \frac{\beta_{TPA} \Gamma_A^{(2)}}{A_{eff}} P^2 \tag{1.13}$$

The total power loss of the integrated silicon waveguides can then be calculated using an effective area of 0.159 μm$^2$ and a confinement factor of 70% for both the linear and non-linear process. The resulting power losses for two-photon coefficients of 1.5, 1.8 and 310 cm/GW are shown in **Figure 1.18**.

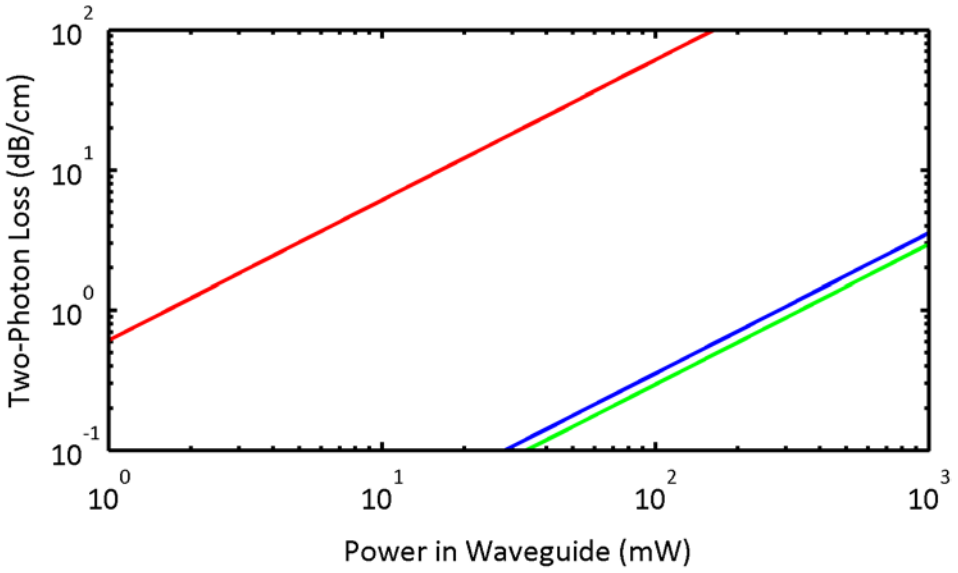


**Figure 1.18** Two-photon absorption losses for the integrated silicon waveguides of 1.5, 1.8 and 310 cm/GW for the lowest to highest lines respectively.

Although this equation now correctly expresses the optical power loss to the two-photon process, it neglects the effects of the carriers excited by the two-photon process. While the excess carriers will recombine inside the semiconductor at the rate determined by the electron and hole lifetimes, a steady state carrier population will exist at optical powers with substantial two-photon absorption. Since the free carriers can also linearly absorb light similarly to electrons in a metal, the free carrier absorption must be explicitly considered. To first calculate the resulting carrier densities, it is convenient to switch to the time domain. By converting the spatial derivative to the time derivative through the group velocity and focusing only on the second-order process, the equation can be rewritten:

$$\frac{dP}{dt} = -\frac{v_g \beta_{TPA} \Gamma_A^{(2)}}{A_{eff}} P^2 \tag{1.14}$$

$$P = \bar{N}_p \hbar \omega v_g A_{eff} \tag{1.15}$$

$$\frac{d\bar{N}_p}{dt} = -\frac{\beta \Gamma_A^{(2)}}{A_{eff}^2} P^2 \tag{1.16}$$

The rate of change of photons in the optical mode can then be directly related to the state of the electronic system in the absence of other absorption sources:

$$\frac{d\bar{N}_e}{dt} = -\frac{1}{2}\frac{d\bar{N}_p}{dt} - \frac{\bar{N}_e}{\tau_e} \tag{1.17}$$

$$\frac{d\bar{N}_e}{dt} = \frac{\beta_{TPA}\Gamma_A^{(2)}}{2\hbar\omega A_{eff}^2}P^2 - \frac{\bar{N}_e}{\tau_e} \tag{1.18}$$

The steady state carrier density can then be easily solved by setting the time derivative to zero:

$$N_e = \frac{\tau_e\beta_{TPA}\Gamma_A^{(2)}}{2\hbar\omega A_{eff}^2}P^2 \tag{1.19}$$

Various experimental studies have measured the impact of free carriers on optical propagation in silicon. In addition to the absorption, the enforcement of the Kramers-Kronig relations between the real and complex parts of the dielectric function requires an index change as well. The fits to the experimental data at 1250nm are shown in Figure 1.19 and Figure 1.20 and estimated as:

$$\Delta n_{electron} = -1.1\times10^{-4}\left(\frac{N_{electron}}{2\times10^{17}cm^{-3}}\right)^{1.045}\left(\frac{\lambda}{1.3\mu m}\right)^2 \tag{1.20}$$

$$\Delta n_{hole} = -2.3\times10^{-4}\left(\frac{N_{hole}}{1\times10^{17}cm^{-3}}\right)^{0.812}\left(\frac{\lambda}{1.3\mu m}\right)^2 \tag{1.21}$$

$$\Delta \alpha_{electron} = -0.4\left(\frac{N_{electron}}{1\times10^{17}cm^{-3}}\right)^{1.176}\left(\frac{\lambda}{1.3\mu m}\right)^2 \tag{1.22}$$

$$\Delta \alpha_{hole} = -0.3\left(\frac{N_{hole}}{1\times10^{17}cm^{-3}}\right)^{1.1}\left(\frac{\lambda}{1.3\mu m}\right)^2 \tag{1.23}$$

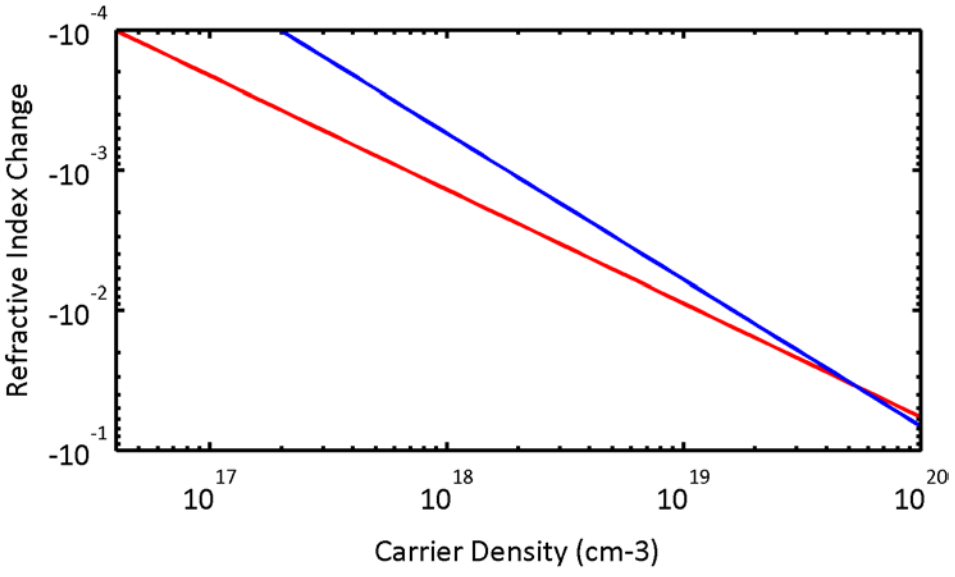


**Figure 1.19** Refractive index change due to free carriers at 1250nm for electrons (blue) and holes (red).

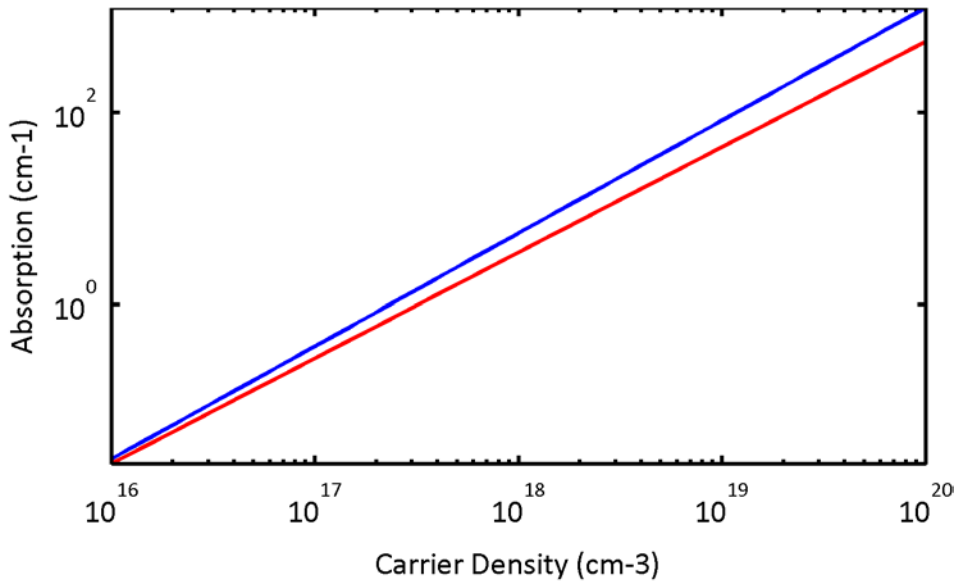


**Figure 1.20** Absorption change due to free carriers at 1250nm for electrons (blue) and holes (red).

The total propagation loss for the silicon waveguide including nonlinear effects of interest can then be written as:

$$\frac{dP}{dz} = -\left(\alpha_A + \Delta\alpha_{electron}(\bar{N}_e) + \Delta\alpha_{hole}(\bar{N}_e)\right)\Gamma_A^{(1)}P - \frac{\beta_{TPA}\Gamma_A^{(2)}}{A_{eff}}P^2 \tag{1.24}$$

$$\alpha_{eff} = \left(\alpha_A + \Delta\alpha_{e+h}\left(\frac{\tau_e\beta_{TPA}\Gamma_A^{(2)}}{2\hbar\omega A_{eff}^2}P^2\right)\right)\Gamma_A^{(1)} - \frac{\beta_{TPA}\Gamma_A^{(2)}}{A_{eff}}P \tag{1.25}$$

The total absorption of the combination of the two-photon and free-carrier absorption effects for the integrated waveguides is shown in **Figure 1.21**. For the ~3ns free carrier lifetime of the integrated waveguides, the guided power limit for less than 1 dB/cm of propagation loss is 76 mW.

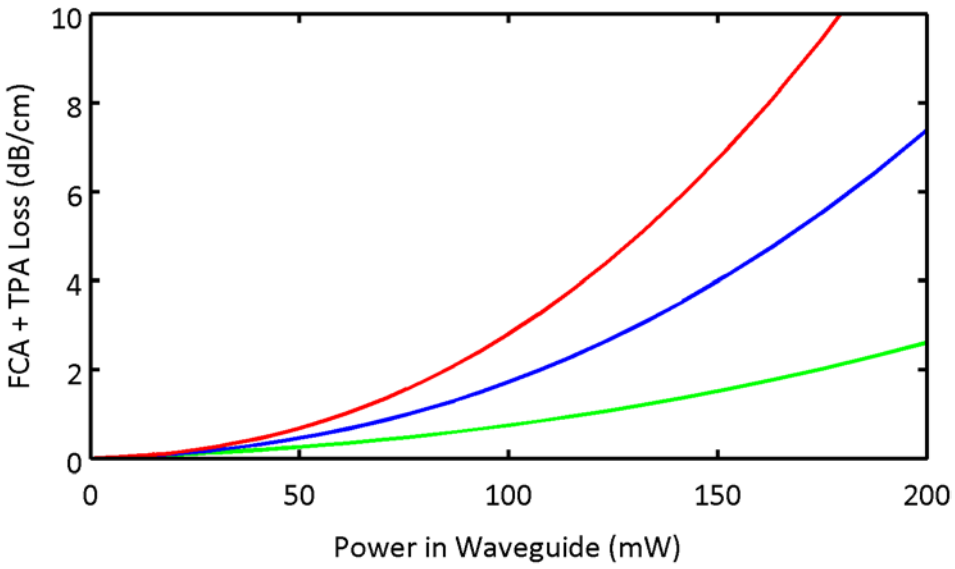


**Figure 1.21** Free-carrier absorption (FCA) and two-photon absorption (TPA) for the integrated silicon waveguides at 1250nm for carrier lifetimes of 1ns, 3ns and 5ns.

The strong-confinement of the high-index contrast system has additional benefits for standard waveguide properties such as a small minimum bend radius. Although the 80nm x 500nm silicon core layer is a non-standard geometry, similar aspect ratios were proposed for optimized waveguide geometries for 1550nm operation [121]. It was previously seen that the index contrast is still sufficient to confine the optical mode to a small area in such a waveguide geometry. Mode solving in cylindrical coordinates, as described in further detail in Chapter 4, allows for the solution of the radiation loss in a bent waveguide. This analysis was performed for three waveguide widths at three wavelengths as shown in Figure 1.22. The minimum ring diameter can then be chosen for a given waveguide loss ceiling over the expected operating wavelength range. For example, if fabrication variation can be neglected, excess radiation loss may be less than 1 dB/cm in a 500nm wide waveguide that is bent at a 3.6 μm radius for operating wavelengths below 1250nm.

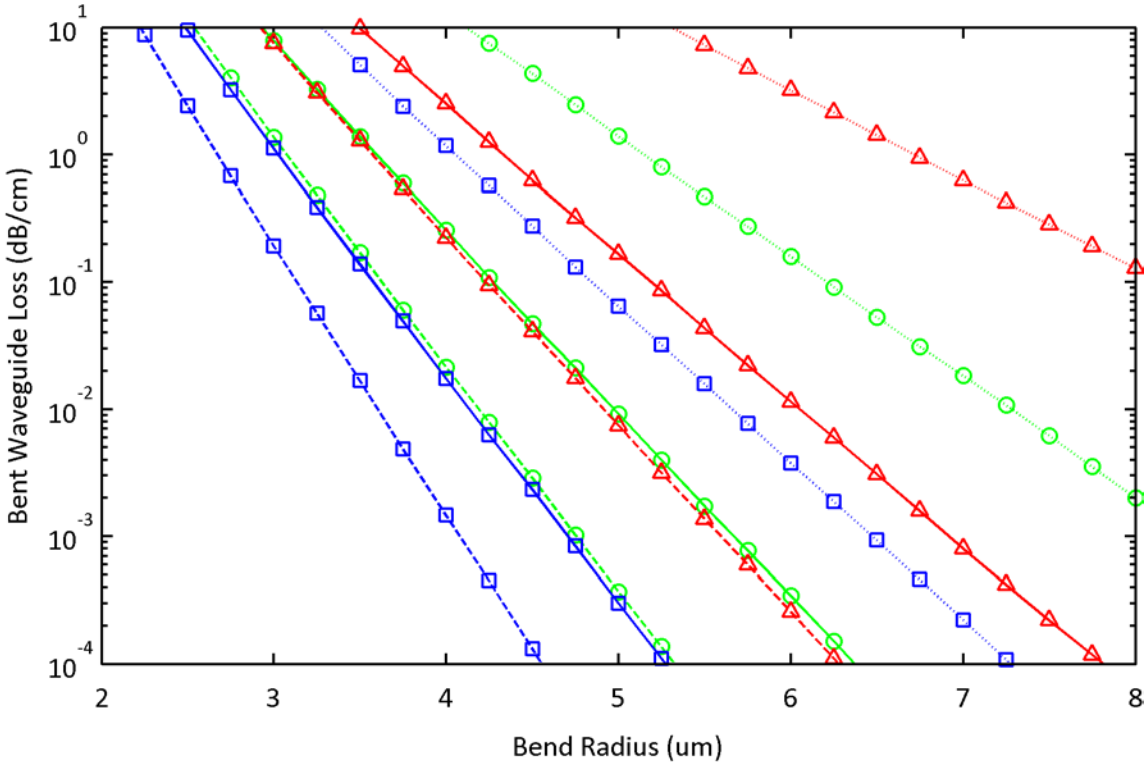


**Figure 1.22** Bend loss simulations for 1200nm (blue squares), 1250nm (green circles) and 1300nm (red triangles) with waveguide widths of 400nm (dotted lines), 500nm (solid lines) and 600nm (dashed lines).

It is important to note that this analysis does not directly determine the loss that may be caused from mode mismatch if a straight waveguide abruptly bends at a given radius. Since the optical mode differs from the straight waveguide to the bent waveguide, excess loss may be incurred unless the transition is made adiabatically or the mode mismatch is corrected by offsetting the straight and bent waveguide mode. This phenomena is a strong function of bend radius and the mode mismatch may be eliminated by using significantly larger bend radii than the 1 dB/cm radiation-limited loss criteria. A crude approximation for radius at which no significant mode mismatch loss may be expected can be achieved by setting a much lower maximum radiation loss criteria for the waveguide bend used for signal routing. Setting such a threshold at $10^{-4}$ dB/cm, yields a conservative bend radius of 8 μm for routing a 80nm x 500nm core waveguide over the 1200nm to 1300nm wavelength range. This value matches the majority of the thin-core photonic layout designs for this thesis where the area consumption of the photonic structures was not a primary constraint.

### 1.7.2. Fiber-to-Chip Couplers

Two major obstacles face coupling the optical mode of the integrated waveguides: optical mode size mismatch and physical fiber access. The traditional way to fiber-couple a waveguide is to form a facet by cleaving, etching or polishing and align an optical fiber in the direction of propagation. This approach, also known as butt-coupling, is well suited to situations where the optical device is easily cleaved and the optical mode size is similar to that of optical fiber. Neither of these two conditions is true for integrated silicon photonics. The optical mode area of standard single-mode optical fiber is approximately 87 $\mu m^2$ or over 500 times the size of the silicon waveguide mode. The silicon waveguide mode may be expanded through a taper in one lateral dimension to provide a better interface, but little can be done to increase the optical mode size in both transverse dimensions without the addition of thick optimized dielectric layers that are not compatible with the CMOS process. The fiber may also be lensed to focus the mode to a smaller area through additional complication during the packaging process. Achieving suitable access to for the optical fiber to enable such a butt-coupling approach is another significant challenge.

**Figure 1.23** Optical micrograph of a waveguide facet in a cleaved 90nm bulk CMOS chip showing the non-planar facet and fragmentation of the heterogeneous backend layer stackup.

Since the optical waveguides are buried under up to 10 μm of heterogeneous dielectric and metal layers, the formation of a suitable facet is not a trivial problem. As shown in Figure 1.23, instead of forming a smooth facet, the residual stress

mismatch and ductility of the metal interconnect causes non-uniform fragmentation. Although such a butt-coupled approach was attempted for one test chip in this thesis, the difficulty in facet formation dramatically complicated optical testing. Other approaches such as dry-etching the facet or careful polishing that does not lead to layer delamination can enable suitable facet formation.

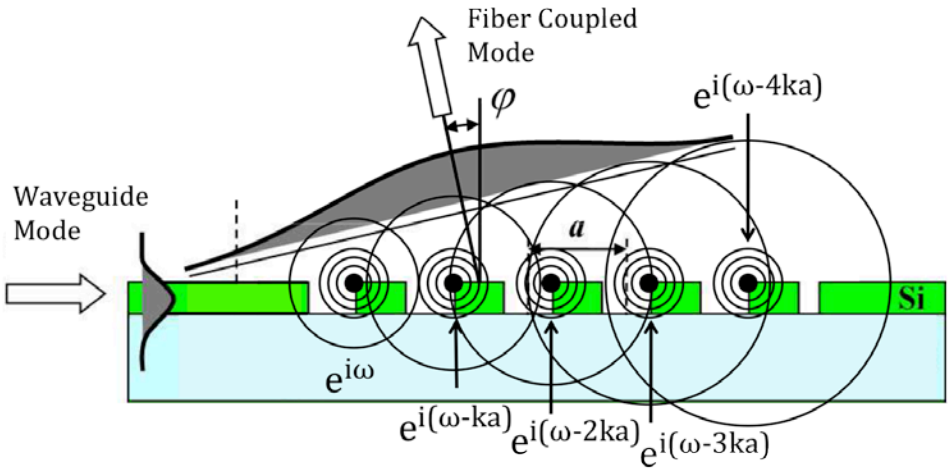


**Figure 1.24** Schematic of coherent field scattering illustrating a Huygens' Principle interpretation of a vertical grating coupler.

Instead, the preferred fiber coupling method and the one used for most of the work presented in this thesis is to form a grating in the waveguide core to radiate the light out-of-plane and into an optical fiber that is placed approximately surface normal to the chip. Similar coupling devices, known as vertical grating couplers, have been developed by many research groups in silicon photonics [122-124]. Although the detailed design of such a structure will be presented in Chapter 4, the basic operation principle of the couplers can be understood by Huygens' Principle as in Figure 1.24. Roughly interpreted, Huygens' Principle states that the aggregate constant-phase wavefront of an electromagnetic wave can be calculated by the envelope of the spherical wavelets emanating from scattering points along the source surface. In the case of a grating coupler, the scattering sources are the grating teeth etched into the silicon waveguide layer. The spacing between these grating teeth in the direction of waveguide mode propagation then defines the phase relation of the scattering sources. As shown in Figure 1.24, the radiating sources corresponding to each tooth are separated in phase from neighboring teeth by the product of the wavevector and tooth separation. If this phase separation equals 2π,

the out-scattered wavefront will be exactly surface-normal. If the ka product decreases, the more distant teeth will be linearly phase-advanced relative to the initial tooth as illustrated in the cartoon. The out-scattered wavefront will then be angled away from surface normal in the direction towards the input waveguide.

It is important to note that there are only two planes of constructive interference from the vertical grating coupler in this simple picture, one above and one below the plane of the waveguide layer. If the structure is fabricated with complete symmetry, the maximum theoretical efficiency of coupling to a single fiber would then be 50%. This can be avoided by either placing a reflective layer on one side of the grating, or by engineering the shape of the tooth. Instead of one scattering point as shown in the cartoon, all refractive index discontinuities of the grating tooth form scattering lines. If the tooth is geometry is correctly designed, deconstructive interference can be obtained for one of the scattering planes to remove theoretical limits to grating coupling efficiency.

The output angle of the grating coupler can also be determined analytically. The output wavevector component in the direction of propagation, $k_z$, must be satisfy the Bragg condition that results from the combination of the input mode wavevector, ß$_{wg}$ and the effective grating wavevector, $K_{grating}$:

$$k_z = \beta_{wg} + nK_{grating} \quad \text{for} \quad n = \ldots,-2,-1,0,1,2\ldots \tag{1.26}$$

with the effective grating wavevector defined as a function of the effective indices in the tooth, $n_{tooth}$, and gap, $n_{gap}$, regions of the grating and the widths of the tooth and gap regions, $W_{tooth}$ and $W_{gap}$ respectively:

$$K_{grating} = \frac{2\pi}{n_{tooth}W_{tooth} + n_{gap}W_{gap}} \tag{1.27}$$

The coupling angle as defined in Figure 1.24 can then be calculated by matching the longitudinal component of the output wavevector into a cladding with a refractive index of $n_{clad}$ for the -1 diffraction order:

$$\varphi = \arcsin\left(\frac{\beta_{wg} - K_{grating}}{n_{cladding}\,2\pi/\lambda}\right) \tag{1.28}$$

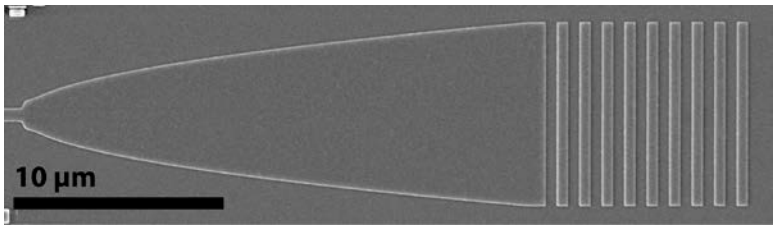


**Figure 1.25** Scanning electron micrograph of vertical grating coupler fabricated in the gate polysilicon layer of a 28nm bulk-CMOS process.

For a complete grating coupler such as the example shown in Figure 1.25, there are four contibutions to the overall fiber coupling efficiency. First, the singlemode waveguide must be tapered in plane to a dimension close to the fiber mode size. This taper loss may be below 1% in short structures (less than 25 µm) by the use of non-linear adiabatic tapers [125]. The resulting transverse optical mode of the widened waveguide will be approximately equal to the outgoing scattered radiation for the direction into the plane of the paper that is not shown in Figure 1.24. Any mode mismatch between the fiber mode and this waveguide mode results in reduced fiber coupling efficiency. Similarly, the relative strengths of the grating tooth scattering sources combined with the reduction of power in the waveguide mode generally combine to create a non-gaussian output intensity. Any mismatch between the scattered field intensity distribution and the fiber mode will result in further losses. Finally, the total radiation efficiency in the direction of the optical fiber provides the final contribution to overall fiber coupling efficiency.

Other workers in the field have achieved total fiber coupling losses below 1.6 dB through multiple methods [126, 127]. The best published result is an insertion loss of 1.2 dB [127] with Luxtera reporting below 1 dB insertion loss couplers in public forums. In this work, simple grating coupler designs have resulted in total fiber coupling efficiencies of approximately 4dB, close to the theoretical maximum of 3dB for symmetric grating designs. The complete design methodology and a comparison with experimental results is presented in Chapter 4.

### 1.7.3. Wavelength Division Multiplexing Filters

Low optical loss, tight bend radii and lithographic patterning to create small gap dimensions combine to enable the formation of microring-resonator wavelength-division-multiplexing filterbanks [68, 71, 128-131]. Similar to a Fabry-Perot cavity formed between two mirrors, a closed ring formed by an optical waveguide forms

constructive interference resonances when the round trip phase condition for the guided field equals 2π. Light may then be coupled into and out of the cavity by placing "bus" waveguides in sufficient proximity that the evanescent fields of each waveguide overlap significantly. Over the optical interaction length of each coupling region, power is therefore coupled from one waveguide to the other. Expressed in terms of electric field coupling coefficients, the various contributions to the field evolution in a ring resonator are diagrammed in Figure 1.26. The field coupling coefficients to the two bus waveguides are the square root of the power coupling coefficient that may be simulated independently as will be discussed in Chapter 4. Expressed in terms of the power loss per unit length α, effective index $n_{eff}$ and ring radius R, the in-ring transmission coefficients may be defined as:

$$t_2 = t_4 = \exp\left(\pi R\left(i\frac{n_{eff}}{\lambda} - \frac{\alpha}{2}\right)\right) \tag{1.29}$$

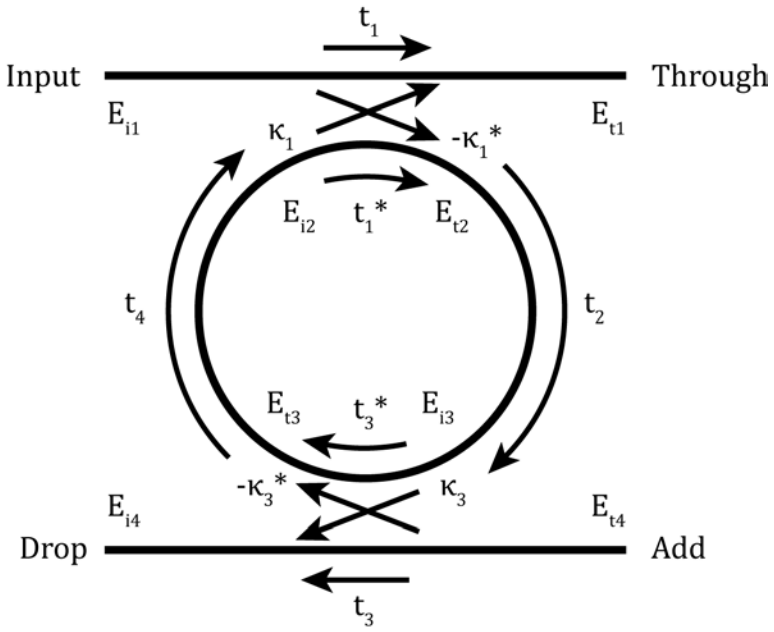


**Figure 1.26** Field transfer diagram for a ring resonator with a drop port.

The round-trip characteristics may then be formally written in a transmission matrix framework to express the various couplings:

$$\begin{bmatrix} E_{t1} \\ E_{t2} \end{bmatrix} = \begin{bmatrix} t_1 & \kappa_1 \\ -\kappa_1^* & t_1^* \end{bmatrix} \begin{bmatrix} E_{i1} \\ E_{i2} \end{bmatrix} \tag{1.30}$$

$$\begin{bmatrix} E_{i2} \\ E_{i3} \end{bmatrix} = \begin{bmatrix} e^{\pi R\left(i\frac{n_{eff}}{\lambda}-\frac{\alpha}{2}\right)} & 0 \\ 0 & e^{\pi R\left(i\frac{n_{eff}}{\lambda}-\frac{\alpha}{2}\right)} \end{bmatrix} \begin{bmatrix} E_{t3} \\ E_{t2} \end{bmatrix} \tag{1.31}$$

$$\begin{bmatrix} E_{t3} \\ E_{t4} \end{bmatrix} = \begin{bmatrix} t_3^* & -\kappa_3^* \\ \kappa_3 & t_3 \end{bmatrix} \begin{bmatrix} E_{i3} \\ E_{i4} \end{bmatrix} \tag{1.32}$$

Setting the input port amplitude to 1 and the add port amplitude to 0, the system of equations can be solved to determine the electric field amplitudes at the through and drop ports:

$$E_{t1} = \frac{t_1 - t_3^* e^{2\pi R\left(i\frac{n_{eff}}{\lambda}-\frac{\alpha}{2}\right)}}{1 - t_1^* t_3^* e^{2\pi R\left(i\frac{n_{eff}}{\lambda}-\frac{\alpha}{2}\right)}} \tag{1.33}$$

$$E_{t4} = \frac{-\kappa_1^* \kappa_3 e^{\pi R\left(i\frac{n_{eff}}{\lambda}-\frac{\alpha}{2}\right)}}{1 - t_1^* t_3^* e^{2\pi R\left(i\frac{n_{eff}}{\lambda}-\frac{\alpha}{2}\right)}} \tag{1.34}$$

Due to the calibration of the input port amplitude of 1, these formulas can be directly converted to transmission functions by squaring the electric field amplitudes:

$$T_{through}(dB) = 20\log\left|\frac{t_1 - t_3^* e^{2\pi R\left(i\frac{n_{eff}}{\lambda}-\frac{\alpha}{2}\right)}}{1 - t_1^* t_3^* e^{2\pi R\left(i\frac{n_{eff}}{\lambda}-\frac{\alpha}{2}\right)}}\right| \tag{1.35}$$

$$T_{drop}(dB) = 20\log\left|\frac{-\kappa_1^* \kappa_3 e^{\pi R\left(i\frac{n_{eff}}{\lambda}-\frac{\alpha}{2}\right)}}{1 - t_1^* t_3^* e^{2\pi R\left(i\frac{n_{eff}}{\lambda}-\frac{\alpha}{2}\right)}}\right| \tag{1.36}$$

Example amplitude and phase transfer functions for a 7 μm radius ring are shown in Figure 1.27 and Figure 1.28. The multiple resonances supported by the cavity are observed by plotting the transmission function over a sufficient wavelength range. Each cavity resonance corresponds to the round trip constructive interference condition for any positive integer m:

$$m = 2\pi R \frac{n_{eff}(\lambda)}{\lambda} \tag{1.37}$$

The free spectral range (FSR) is then defined by the frequency or wavelength range between adjacent resonances:

$$FSR = \lambda_2 - \lambda_1 = 2\pi R\left(\frac{n_{eff}(\lambda_2)}{m+1} - \frac{n_{eff}(\lambda_1)}{m}\right) = \frac{\lambda^2}{2\pi n_g R} \tag{1.38}$$

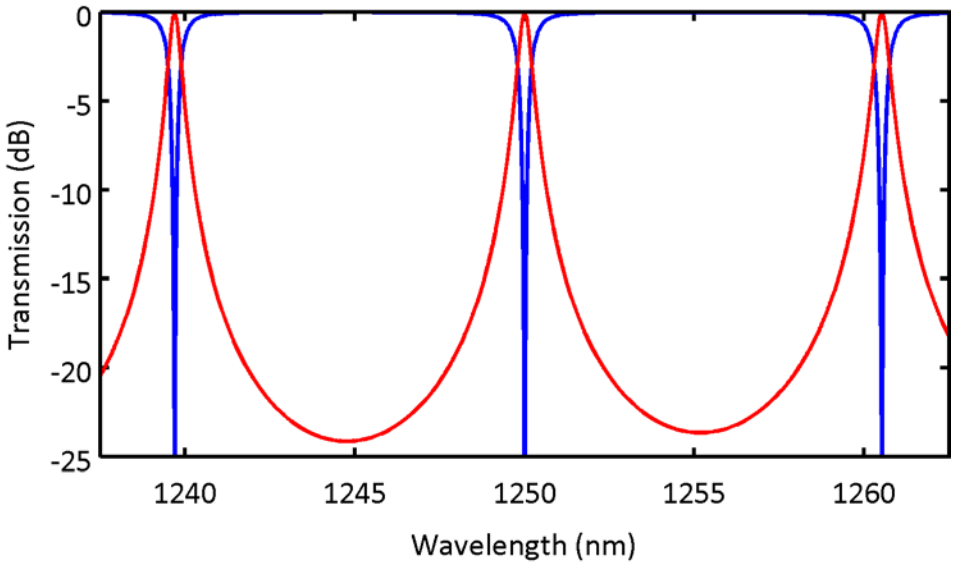


**Figure 1.27** Amplitude transfer function for a 7 μm radius ring formed by an 80nm x 500nm waveguide that is strongly coupled to bus waveguides with a 12% power coupling coefficient.

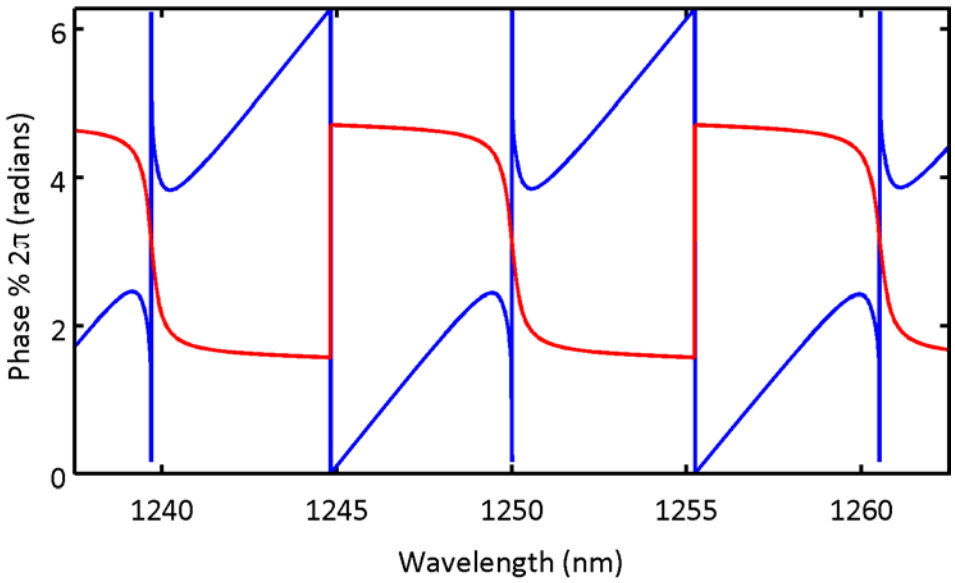


**Figure 1.28** Phase transfer function for a 7 μm radius ring formed by an 80nm x 500nm waveguide that is strongly coupled to bus waveguides with a 12% power coupling coefficient.

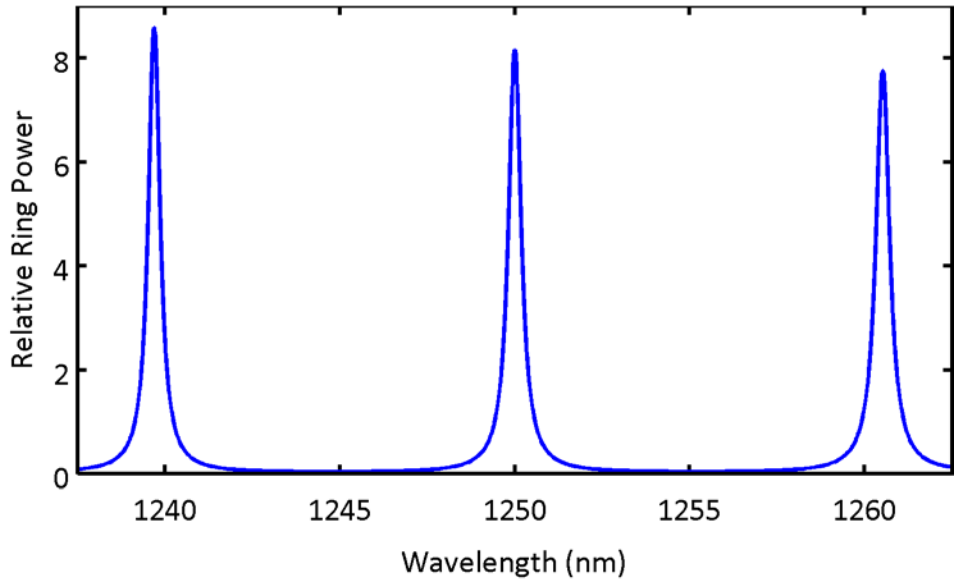


**Figure 1.29** Relative intracavity power for a 7 μm radius ring formed by an 80nm x 500nm waveguide that is strongly coupled to bus waveguides with a 12% power coupling coefficient.

The basic performance characteristics may then be defined for the ring resonator. The full width at half maximum (FWHM), or equivalently the 3-dB bandwidth, can be calculated as a measure of the resonance bandwidth. The ratio of the resonance center frequency to the bandwidth then defines the quality factor (Q). The quality factor is then a generic measure as defined 2π multiplied by the energy stored per optical cycle divided by the energy lost. However, since the FSR of the resonator is typically much less than the carrier frequency, it is also useful to equivalently define the finesse ($\mathcal{F}$) as the ratio of the FSR to the FWHM. In systems without the drop port, the finesse defines the power enhancement in the cavity on resonance. The presence of the drop port of the resonator reduces the intracavity power below the value that would be calculated by the finesse. Using the transmission matrices of Equations 1.12-1.14, the intracavity power enhancement ratio can be calculated as $\left|E_{t2}\right|^2 / \left|E_{i1}\right|^2$ :

$$\text{Relative Intracavity Power} = \frac{-\kappa_1^*}{1 - t_1^* t_3^* e^{2\pi R\left(i\frac{n_{eff}}{\lambda} - \frac{\alpha}{2}\right)}} \tag{1.39}$$

The intracavity power for the example resonant filter is shown in Figure 1.29. The maximum power allowable for an integrated photonic system must then not only consider the nonlinear absorption in a waveguide, but also the ring resonator filters where the circulating power may be over an order of magnitude higher on

resonance. To maintain the same 1 dB/cm limit on non-linear absorption within the ring resonator, the input power to a bus waveguide for such a ring resonator is ~10 mW instead of the 76 mW limit set by a limit of absorption in the waveguide alone.

Ring resonators with slightly different ring radii may then be cascaded within one free spectral range to construct a filter bank to be used for wavelength division multiplexing. The through port is then a cascade of all filters connected to one bus and the drop ports of each filter should select unique ranges of wavelengths to be rerouted. An example cascade of four of the example filters with a 20nm radius step between adjacent channels is shown in Figure 1.30. The multiple wavelength channels that are aggregated onto one bus waveguide may then be independently routed, manipulated and received. This enables a very large bandwidth density to route data on- and off-chip while using few optical fibers and consuming very little silicon real estate.

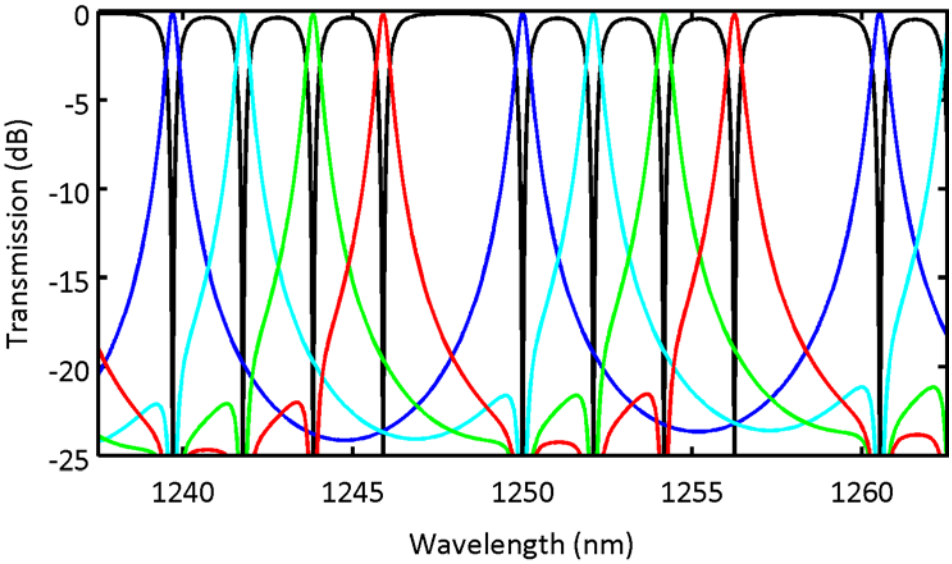


**Figure 1.30** Four channel wavelength division multiplexing filter bank constructed by nominally 7 μm radius rings with a 20nm radius step between adjacent channels.

### 1.7.4. Modulators

To encode data from the electrical domain onto the optical carrier, some form of transduction modulation is required. The free carrier induced index change can be leveraged to shift the effective index of the waveguide mode to generate a phase change of the passing optical signal [132]. The electrical information can then be transduced into the optical domain by the presence or lack of free carriers

overlapping with the optical mode. Since many doping and metallization steps from the standard CMOS process can be applied to the front-end silicon layers used for waveguide fabrication, diodes may be formed across the optical waveguide to control this free carrier overlap based upon the electrical data. The design of these contacts and diode junctions to minimize optical loss while enabling such modulation is the core of optical modulator design and will be discussed in Chapter 5. This section will neglect specific discussion of the diode contacts to focus on the fundamental operating principles and performance characteristics.

Although it is possible to transmit information directly as the differential phase of the optical carrier, the optical data transmission system is greatly simplified in an amplitude-based coding scheme. An optical structure based upon phase-sensitive interference, such as a ring resonator or Mach-Zehnder interferometer, can then be used to convert the phase modulation into an amplitude change [73, 74, 93, 107, 109, 133-136]. Since the resonant enhancement of the ring geometry requires a much smaller quantity of optical carriers to achieve the amplitude modulation, only ring resonator modulators are considered in this work.

Compared to the ring resonators used for wavelength division multiplexing filters, the same round-trip transmission characteristics apply. The difference is that instead being fixed parameters, the ring transmission coefficients $t_2$ and t4 are modified by the presence of free carriers:

$$t_2 = t_4 = \exp\left(\pi R\left(i\frac{n_{eff} + \Delta n(N_e)\Gamma^{(1)}}{\lambda} - \frac{\alpha + \Delta\alpha(N_e)\Gamma^{(1)}}{2}\right)\right) \tag{1.40}$$

The ring resonator can then be electrically modulated by either injecting carriers into nominally intrinsic silicon or by depleting carriers introduced through material doping. To build a carrier depletion modulator, opposite polarity implant regions are abutted at the peak intensity of the optical mode to form a pn-junction. The width of the depletion region that forms is then a function of applied diode bias voltage ($V_A$) and dopant density ($N_D$):

$$W_D = \sqrt{\frac{4\varepsilon_{Si}}{q N_D}\left(\frac{k_B T}{q}\ln\frac{N_D^2}{n_i^2} - V_A\right)} \tag{1.41}$$

where $\varepsilon_{Si}$ and $n_i$ are the dielectric constant of silicon and intrinsic carrier density of silicon respectively. In the depletion modulation case, applying a voltage to the pn-junction modifies the overlap of constant carrier density with the optical mode,

thereby modulating $\Gamma^{(1)}$ in Equation 1.36. Figure 1.31 then shows the transmission functions for a ring resonator modulator using $N_D=5x10^{17}$ $cm^{-3}$ modulated from an effective confinement factor of 70% in the *off* state to 46% in the *on* state. If the incoming optical signal has a wavelength of 1250nm, the light will observe high loss in the modulator's *off* state (zero bias) and high transmission in the *on* state (reverse bias). For such a design without consideration for how the diode is electrically contacted in such a resonant optical structure, the extinction ratio between *on* and *off* states is greater than 20 dB with less than 0.5 dB insertion loss in the *on* state. It is also noted that the modulation is observable for only a narrow range of wavelengths. This allows such modulators to be cascaded on a single optical bus provided that the ring radii are stepped such that the modulators have different resonant frequencies.

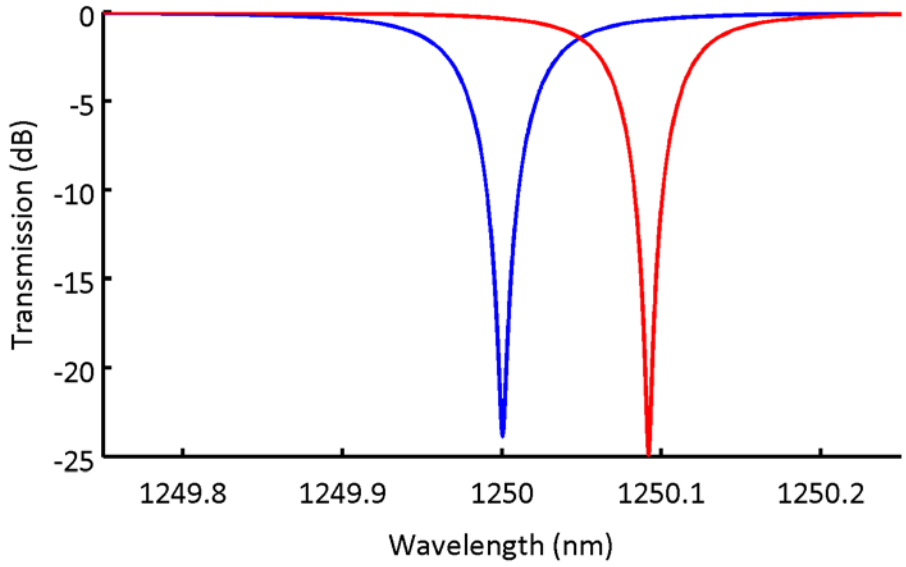


**Figure 1.31** Optical transmission functions for the *on* (red) and *off* (blue) states of the depletion ring modulator. Design parameters: ring radius 4 μm, background waveguide loss 3 dB/cm, through port coupling 1%, drop port coupling 0.25%, and doping concentration $5x10^{17}$ $cm^{-3}$. The operating laser wavelength would be 1250 nm for this example.

Electrically, a depletion modulator can be accurately modeled as a capacitive load with a series resistance. The capacitance is dominated by the narrow separation of the electron and hole carrier populations in the pn-junction that varies as a function of reverse bias. The energy consumption and maximum switching frequency can then be simply calculated as an RC circuit. The best published results for depletion modulators have achieved a 12.5 Gbps data rate with 3 fJ/bit energy

consumption [92]. A design for a depletion modulator in the thin-SOI platform is presented in Chapter 2 for an integrated photonic link.

Alternatively, to inject carriers into the optical mode region, a p-i-n diode can be formed across the optical waveguide. The modulation signal is then the presence or lack of free carriers in the optical mode volume. Since the "off" state does not require a background carrier concentration, it is possible to make a lower loss resonator. In comparison the depletion modulator example that required $5x10^{17}$ $cm^{-3}$ doping density, the injection modulator shown in **Figure 1.32** requires only $2x10^{17}$ $cm^{-3}$ injected carriers to achieve modulation.

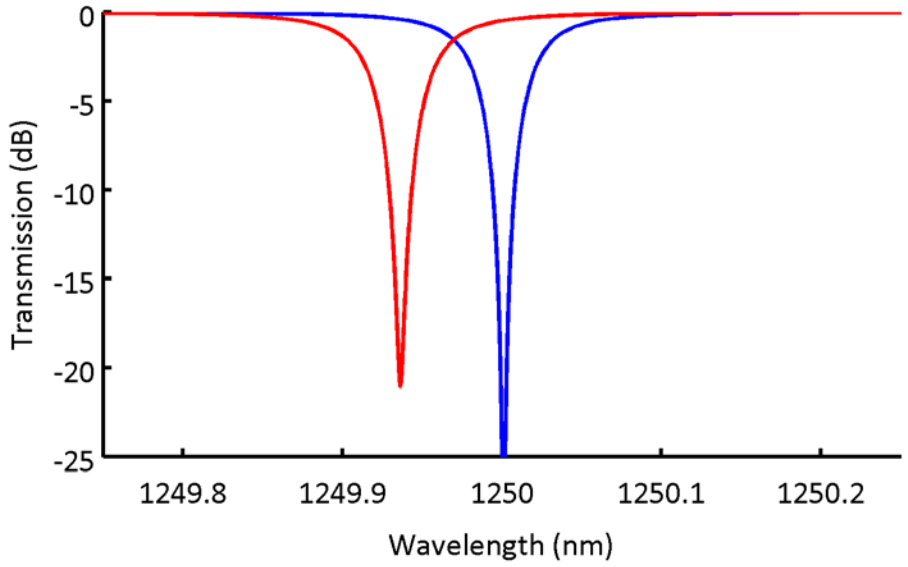


**Figure 1.32** Optical transmission functions for the *on* (red) and *off* (blue) states of the injection ring modulator. Design parameters: ring radius 4 μm, background waveguide loss 3 dB/cm, through port coupling 0.7%, drop port coupling 0.55%, and injected carrier concentration of $1x10^{17}$ $cm^{-3}$. The operating laser wavelength is 1250 nm for this example.

The chief tradeoff of the injection modulator design results from the electrical characteristics of the modulator. Instead of being well approximated as a capacitive load, significant current flows through the junction in forward bias. The power dissipation then requires the inclusion of the I-V product in addition to the capacitive switching dissipation. Additionally, instead of the modulation signal being a simple function of voltage as in the case of the depletion modulator, the relevant state variable is the carrier density in the active region. The transient behavior is determined by the combination of the injection (turn on) and depletion (turn off) time constants.

The rate of removal for the excess carrier density in the intrinsic region of the p-i-n diode is the combination of the carrier drift due to the electric field and recombination due to the finite carrier lifetime. The inverse of the resulting turn off time constant can be found by the summation of the inverses of the recombination and drift time constants. The relevant parameters for the slowest carrier species provide are then used, i.e. the electron recombination lifetime $t_e$ and the hole mobility $\mu_h$. To calculate the drift time constant, the width of the intrinsic region $W_i$ built-in voltage of the diode $V_{bi}$, and applied reverse bias voltage $V_R$ are also required. The resulting expression is then [137]:

$$\frac{1}{\tau_{off}} = \frac{1}{\tau_e} + \frac{2.4\,\mu_h\left(V_{bi}+V_R\right)}{W_i^2} \tag{1.42}$$

The injection time constant involves a more complicated state change of the entire diode operating point. The complication as compared to the depletion time constant is that the applied current results in both increasing the carrier density of the intrinsic region and injected minority carrier recombination in the quasineutral regions. Accurate representations of the relevant physical processes are best obtained by solution of the semiconductor transport equations through finite element modeling of the exact device geometry as further described in Chapter 5. A basic analytical approximation for the time constant of the injection phenomena can be taken from prior work focused on the turn-on delay in semiconductor lasers. The laser "turns on" when the carrier density, N, in the intrinsic region reaches the threshold carrier density, $N_{th}$. Similarly, the optical modulator will switch states from the on to off resonant frequencies when the carrier density in the intrinsic region of the encompassing diode reaches $N_{on}$. The relevant physical processes between the carrier-injection modulator and semiconductor laser below threshold are sufficiently similar to enable application of the same analytical techniques. The injection time constant can then be approximated with the turn-on time constant of a laser diode. The previously derived and experimentally verified approximation then relates the time constant to the steady-state drive current density $J_{on}$ that corresponds to the on-state carrier density as a function of the injection drive current density $J_{inj}$ and carrier lifetime $t_e$ [137]:

$$\tau_{on} = \tau_e \ln\left(\frac{J_{inj}}{J_{inj}-J_{on}}\right) \tag{1.43}$$

Since the carrier lifetime in a single-crystalline silicon layer may range from 1-5 ns, significant tricks are required to achieve the 100 ps bit-times required for 10 Gbps communication. The transient response therefore requires a significantly larger drive current to turn on the diode than to maintain the desired carrier density to achieve modulation. The logarithmic relation simply relates the required turn on time constant relative to the carrier lifetime in the structure: a factor of 2 reduction requires a factor of 2.5 current overdrive and a factor of 10 reduction requires a factor of 10.5 current overdrive for examples. Successful overdrive pre-emphasis with single-crystalline ring modulators to achieve 12.5 Gbps operation has been previously demonstrated with external circuit components [73]. The power dissipation, however, must include not only these current spikes, but also the required drive energy for the input capacitance for the low-resistance transistors used to achieve this pre-emphasis. As such, depletion modulators provide the most desirable path towards energy efficient operation at 10 Gbps data rates.

### 1.7.5. Detectors

The final photonic device type, photodetectors, traditionally requires an additional semiconductor material that has a lower bandgap energy than silicon. This allows the efficient absorption of the optical signal that had been propagating in the silicon guide. Although recent work has explored novel approaches to enable photodetection in the silicon waveguide material itself [138, 139], this thesis will primarily consider traditional photodetector designs that are based upon the single photon band-to-band absorption of light.

Ideally, all incident light should be absorbed only in the high field depletion region. This goal is often aided by introducing a smaller bandgap material without doping between the p and n regions. This type of heterostructure p-i-n diode allows for high device performance, but requires complicated fabrication. The flexibility of the process integration within the memory process enables for the fabrication of such a p-i-n diode. For integration within standard CMOS processes, the photodiode geometry must be adapted to the available processing flexibility and will discussed in detail in Chapter 5. A cartoon of a prototypical p-i-n diode is shown in Figure 1.33 and will be the focus of the introductory device physics discussed in this section.

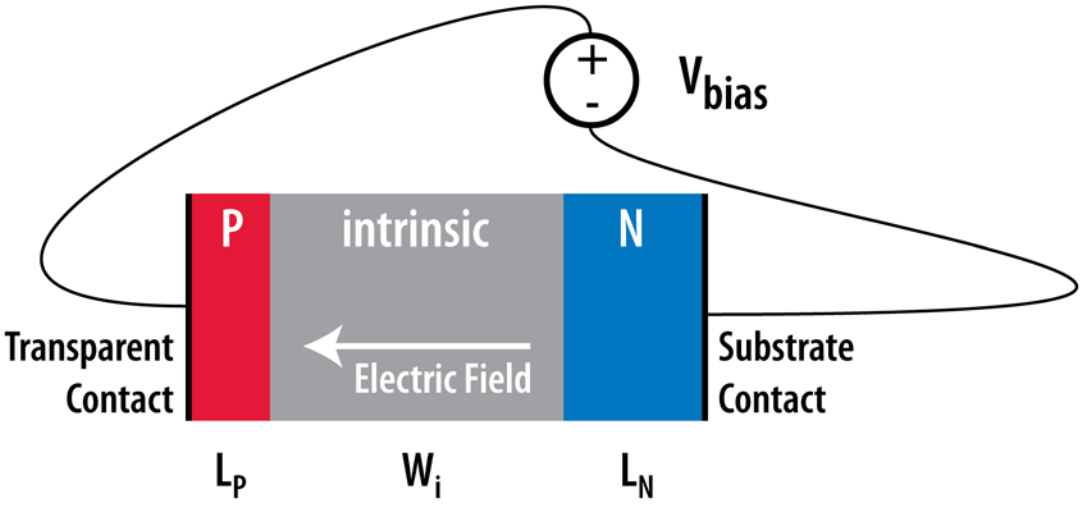


**Figure 1.33** Cartoon of representative p-i-n diode structure for photodiode applications.

Current generated by incident photons is the desired signal, but other sources of current are present as well. In addition to photon excitation, shown in **Figure 1.34**, thermal generation independent of incident light can cause electron-hole pairs that contribute to current as shown in **Figure 1.35**. These other current sources are often called dark current as they are present without optical illumination and are a parasitic noise source that interferes with device function.

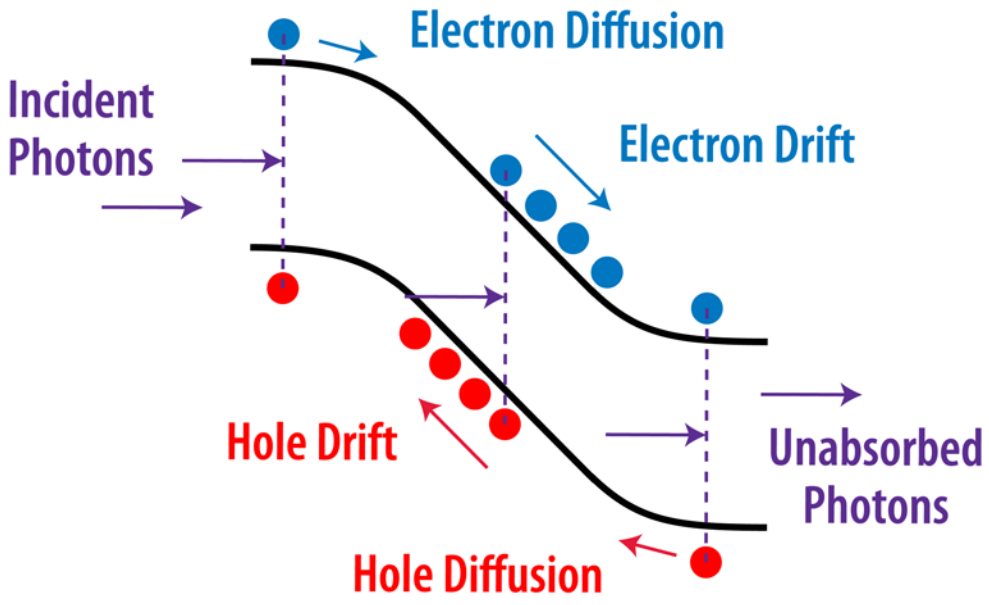


**Figure 1.34** Energy band diagram for optical absorption in a p-i-n photodiode under reverse bias.

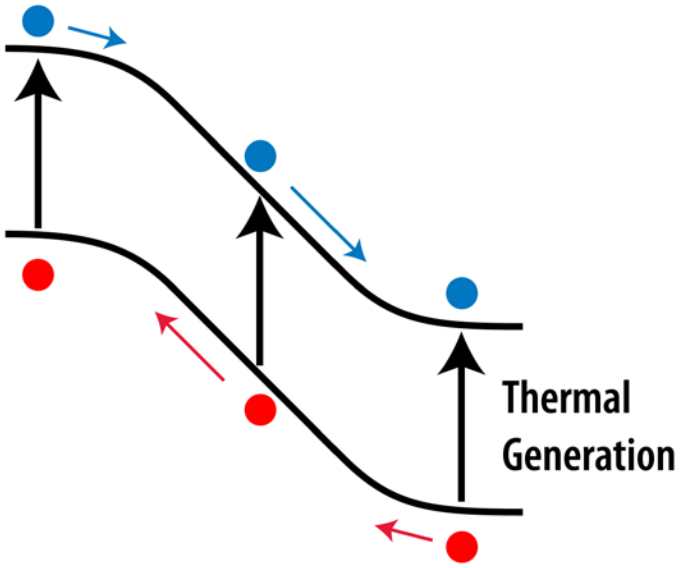


**Figure 1.35** Energy band diagram showing thermal generation that results in leakage current in a p-i-n photodiode.

The magnitude for these two current sources can be easily approximated. For the optical generation, the incident optical power is reduced by device reflection and incomplete absorption. Additionally, the incident photons need to be calculated by dividing the power by the photon energy. The current is then determined by multiplying by the charge of the electron modified by the internal quantum efficiency, η, that is the fraction of generated electron-hole pairs that contribute to current. The resulting product can be expressed in terms of the reflectivity, $r$, incident power, $P_0$, absorbing layer thickness, $t_{film}$, absorption coefficient, $\alpha_{Ge}$, and photon energy, $E_{photon}$, as:

$$I_{opt} = q\eta \frac{(1-|r|^2)P_0}{E_{photon}}\left[1-\exp\left(-\alpha_{Ge}t_{film}\right)\right] \tag{1.44}$$

All of these quantities except for the internal quantum efficiency can be easily estimated by simple calculations and literature review. The internal quantum efficiency depends on several transport parameters but can be roughly estimated as follows. In order for the absorbed photons to contribute to current, they must be absorbed within approximately one diffusion length of the depletion region with a total length of $L_p + L_n + W_i$. Recombination within this region further reduces the current by the fraction of generated carriers that recombine. The approximate internal quantum efficiency can therefore be written for a photodiode illuminated

from the p-side in terms of the excess carrier recombination rate, $U_{excess}$, optical generation rate, $G_{opt}$, p-side width, $x_p$, and n-side width, $x_n$, as:

$$\eta \approx \frac{G_{opt} - U_{excess}}{G_{opt}} \left[ e^{-\alpha_{Ge}(x_p + W_i + x_n)} - e^{-\alpha_{Ge}(x_p + W_i + L_n)} - \left( e^{-\alpha_{Ge}(x_p - L_p)} - e^{-\alpha_{Ge} x_p} \right) \right] \quad (1.45)$$

The relation between incident optical power is very often consolidated into the experimental parameter of responsivity and expressed as such as shown in Equation 1.45. This parameter having units of A/W contains all of the dc photoresponse information for a given wavelength.

$$I_{opt} = RP_0 \quad (1.46)$$

A similar expression can be obtained for thermal generation by the use of minority carrier lifetimes. Thermal current can be estimated by calculating the generation in each region that contributes to current. For the quasineutral regions, the generation is equal to the recombination by the principles of detailed balance and can therefore be expressed by dividing the minority carrier density with the minority lifetime. The thermal current can therefore be expressed in terms of the area, $A$, minority carrier densities, $n_{p_0}$ and $p_{n_0}$, minority lifetimes, $\tau_n$ and $\tau_p$, and the depletion region generation, $G_0$, as:

$$I_{therm} = qA \left( \frac{n_{p_0}}{\tau_p} L_p + G_0 W_i + \frac{p_{n_0}}{\tau_n} L_n \right) \quad (1.47)$$

The depletion region generation can be estimated by first assuming that midgap flaw recombination observing Shockley-Read-Hall statistics dominates as is often the case for indirect gap semiconductors. The recombination rate can therefore be written in terms of the electron density, $n$, hole density, $p$, and intrinsic carrier density, $n_i$, as:

$$U^{SRH} = \frac{np - n_i^2}{\tau_p (n - n_i) + \tau_n (p + n_i)} \quad (1.48)$$

When electron and hole densities are much less than intrinsic carrier density as is the case in the depletion region of a diode, the recombination rate becomes negative and the expression for the intrinsic region generation in this limit can be obtained:

$$G_0 = \frac{n_i}{\tau_p + \tau_n} \quad (1.49)$$

This generation rate can be inserted back into Equation 1.46 and the minority carrier densities can be expressed in terms of the majority carrier densities, $n_n$ and $p_p$, and the intrinsic carrier density:

$$I_{therm} = qA\left(\frac{n_i^2 L_p}{p_p \tau_p} + \frac{n_i W_i}{\tau_n + \tau_p} + \frac{n_i^2 L_n}{n_n \tau_n}\right) \tag{1.50}$$

For wide bandgap semiconductors where $n_i$ is small, thermal generation is dominated by the contribution from the depletion region. The difference between silicon, bandgap 1.12 eV, and germanium, bandgap 0.661 eV, results in a depletion region to diffusive region current ratio of approximately 3000 for silicon and 0.1 for germanium. This means that for germanium the diffusion needs to be included in dark current analysis which requires a full accurate model of the device for realistic theory.

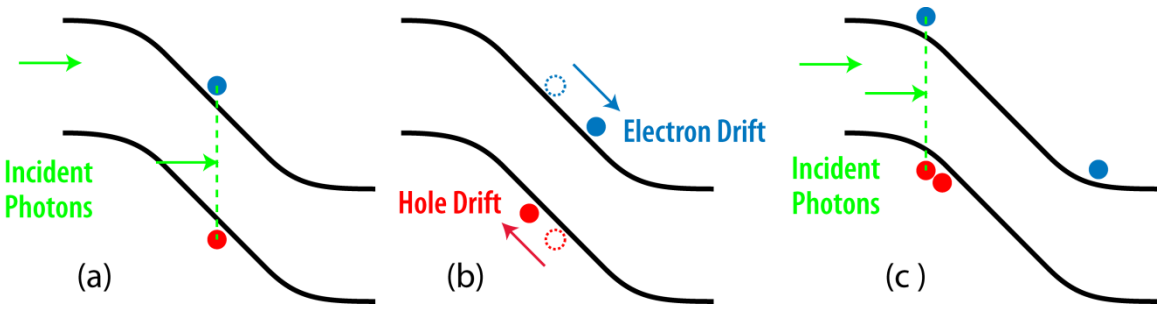


**Figure 1.36** Time domain picture of frequency response limitations caused by finite drift transit times. (a) Carriers are generated by an incident photon. (b) Electric-field induced drift causes the carriers to move to the contacts and generate photocurrent. (c) If incident photons corresponding to the next distinguishable input signal, e.g. adjacent peak of sinusoid or optical data bit, generates an electron-hole pair while the carriers are still drifting to the device contacts, the resulting photocurrent will be indistinguishable between the two distinct input photon events.

The transient or frequency responses of a p-i-n diode are less tractable by analytical theory. To gain some insight into the factors that affect device performance, the simplification of only considering depletion region electron-pair generation yields analytical formulas. This analysis can be setup by considering sinusoidally modulated light being absorbed at the two edges of the depletion region. The electron hole pairs must then drift across the depletion region, a process that takes a finite time determined by the drift velocity. The electrons or holes coming from the other side of the depletion region are then out of phase with the

electrons and holes being generated at that location as shown in **Figure 1.36**. The result is a dephasing of the absorbed light, limiting the frequency response of the photodiode:

$$I_{opt}(t) = RP_0 \int_0^{W_i} \exp\left( j\omega \left( t - \frac{x}{v_{drift}} \right) \right) dx \tag{1.51}$$

$$I_{opt}(t) = RP_0 \frac{v_{drift}}{W_i} \frac{1 - \exp\left( -j\omega \frac{W_i}{v_{drift}} \right)}{j\omega} \exp\left( j\omega t \right) \tag{1.52}$$

$$\left| \frac{v_{drift} \exp\left( j\omega_{3dB} t \right)}{j\omega_{3dB} W_i} \left[ 1 - \exp\left( -j\omega_{3dB} \frac{W_i}{v_{drift}} \right) \right] \right|^2 = \frac{1}{2} \quad \rightarrow \quad \omega_{3dB} = \frac{2.4}{W_i / v_{drift}} \tag{1.53}$$

This analysis should be a rough guide to understand the fundamental limit to the frequency response of a p-i-n photodiode. Photodiode frequency response goes down with increasing intrinsic region width for a given drift velocity. It should also be noted that the relevant drift velocity is the hole drift velocity which is slower than the electron drift velocity.

For similar p-i-n diodes monolithically integrated in customized CMOS processes, the highest performance device achieves 0.85 A/W with a 26 GHz bandwidth [91]. Absent of electronic integration constraints, germanium photodiodes achieve >20 GHz bandwidth with 1.05 A/W at 1550nm [110]. In zero-change CMOS integration, no published results exist for photodetectors operating below the silicon bandgap. The progress towards this goal will be presented in Chapter 5.

## 1.8. Summary

Over the course of the introduction, the various processing constraints associated with the electronics processes of interest have been presented. The goal of the thesis is to produce integrated photonic devices of comparable performance to those demonstrated by the alternative photonic integration methodologies that were discussed. For the CMOS processes, the goal is to design the devices to require zero changes to the in-foundry processing. For the DRAM process, a combination of device and process design is required to optimize both the processing steps and device design. The tutorial on high-index contrast photonic devices that concludes

this chapter provides a context for the achieved device performance and more specific device design.

# 2. CORE-TO-MEMORY LINKS FOR MULTICORE PROCESSORS

The provisionable bandwidth between the processing core and the main memory has become a key performance bottleneck for modern computers. Limitations on the number of physical connections, channel bandwidth and power dissipation constrain the aggregate data rate for the system's electrical links. As discussed in Chapter 1, photonic links can overcome these limitations by their inherent capabilities of wavelength division multiplexing (WDM), high carrier frequency and external modulation. These advantages provide strong motivations to develop a suitable optoelectronic platform to be leveraged for microprocessor applications. To meet this need, the monolithic integration platform presented in this thesis enables photonic device fabrication within the exact state-of-the-art electronics processes used for the processor and memory chips that comprise modern computers.

A greatly simplified picture of computing systems will be used in this analysis. Due to the complexity of the systems involved, accurate modeling and architectural predictions of computers utilizing photonic links are major research undertakings in their own rights. Instead, the goal of this chapter is to look broadly at the scaling of a general purpose, multi-core central processing unit (CPU) while focusing on the technology requirements of the core-to-memory interconnect. The context of this analysis will be mostly focused on the interface between a power-constrained multi-core CPU chip and DRAM memory modules.

In this chapter, I will first further develop why electrical links face difficulty providing the required core-to-memory bandwidth in existing systems. This situation is in contrast to other computer communication links such as core-to-core links where the advantage of integrated photonic links is less profound. To ensure the suitability of photonic links for the end system, I will then analyze physical link and network architectures that evaluate the required total link energy, laser power and chip area consumption. When available, realistic photonic device performance estimates will be established based on devices fabricated over the course of this thesis. Where fabricated devices are too immature to be representative of an end system, a combination of literature sources and device modeling calibrated to the achieved technology platform performance will be used. I will end with a brief review of more advanced photonic link architectures.

## 2.1. Target System

From a high level, the general purpose, multicore microprocessor has several subsystems of interest for the current discussion: computation, clock distribution, core-to-core communication on die, socket-to-socket communication, and core-to-memory communication. These subsystems must share on-chip power, silicon area, backend metallization wiring, and off-chip I/O pins. The chief constraint that has traditionally driven trade-offs between each of these subsystems is how much power may be allocated for a given function. Standard total power dissipation for thermally-limited microprocessors has stabilized to be approximately 100 W [86, 87]. Joint optimization of various microprocessor subsystems has driven a coarse breakdown of power as follows [138, 140-144]: 60% is allocated to computation internal to the various processing cores; 20% is allocated for all on chip communication including clock distribution and core-to-core links; and 20% is allocated for off-chip communication including socket-to-socket and core-to-memory links.

Simultaneously, the performances of the various subsystems are linked together to enable optimal system function. Fundamentally, the communication bandwidth should support sufficient data exchange on core-to-core and core-to-memory links such that the operands are always available at each processing core to enable high utilization of the computing hardware. This means that bandwidth provisioning should not be scaled arbitrarily just to meet the total power dissipation specification. Specifically in the context of the core-to-memory bandwidth of a system, the relationship between floating point operations (FLOPs) in the computing cores and the data fetched from main memory is typically desired to be one byte of data per FLOP [145-150].

Although this high-level breakdown is not fundamental for all application types, deviation from this linked scaling forced by technical obstacles can be seen to result in sub-optimal systems. The byte per FLOP desired bandwidth target has not been met in general purpose CPUs and instead is typically below 0.5 B/FLOP in Intel's Xeon line of processors [151-154]. The primary challenges are the number of I/O pins and total power dissipation that may be devoted to the memory interface [152]. The performance impact of this change has been reduced for many applications by the introduction of a large on-chip memory cache [155]. Since the memory bandwidth available to fetch operands from main memory can no longer be

sufficiently provisioned, the size of on-chip caches have greatly increased to hide the bottleneck from impacting several applications. As shown by comparing Intel processors fabricated 8 years apart in **Figure 2.1**, the cache sizes have greatly increased as a fraction of die area. If the memory size of data that is frequently used for the current application is sufficiently small to fit into the on-chip cache, the application execution avoid main-memory access bottlenecks and proceed at a computationally limited pace. This circumstance is unfortunately not the case for most high-performance computing applications and many consumer-level tasks. Additionally, the die area used for the caches may not be used for additional processing cores. In the current, memory-limited environment, this limit does not present a significant tradeoff since the increased computation power on chip would not have sufficient memory bandwidth to run at peak speed for most applications. The elimination of the memory interconnect bottleneck therefore offers a coupled scaling of processor performance; increased bandwidth increases the efficiency of processing cores while simultaneously increasing the number of cores that may fit on a single chip.

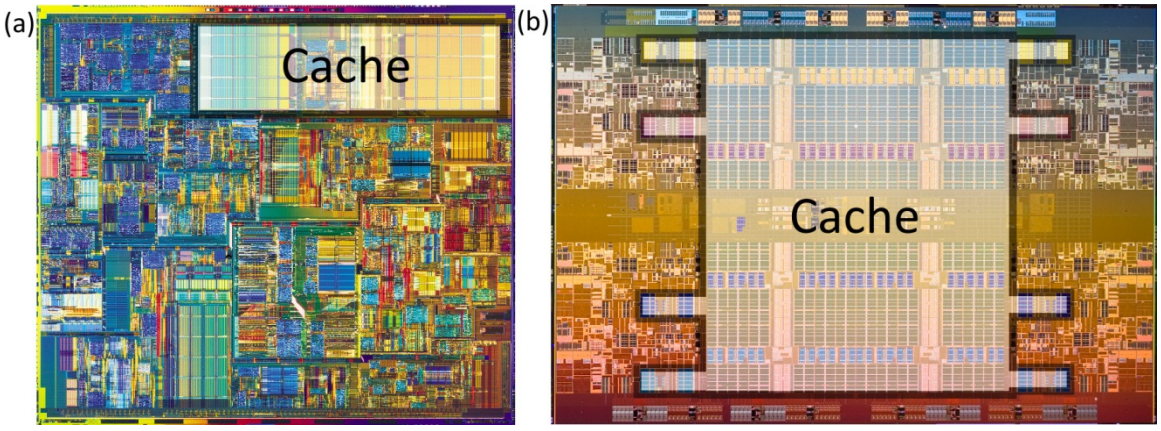


**Figure 2.1** Die photos of an Intel Pentium 4 "Northwood" released in 2002 implemented in 130 nm technology (a) and an Intel Xeon 7500 "Beckton" released in 2010 implemented in 45 nm technology (b). The dramatic increase in chip area devoted to local memory cache as compared to computing cores is indicative of the broken scaling trends caused by a bottleneck in off-chip electronic links from the cores to main memory.

For end-system comparisons, it is helpful to start with some assumptions on the computational performance of the logic cores to set the interconnect need. As the saturation of logic clock rates has existed for much of the last decade with no signs of change, it is assumed that the core clock rate will remain between 2 and 3 GHz. In

contrast, the continuing advances in reducing computation energy cost will be set in line with the active DARPA Ultra High Performance Computing (UHPC) program to be 10 pJ/FLOP [156]. The total processor power considered will be 100 W as the heat dissipation limited average of the last decade of processors [86, 87]. The budgeted computation energy allows for 6 TFLOP in the target processor. To maintain the 1 B/FLOP target, the read and write data paths each require 48 Tbit/s link bandwidth. Since the off-chip communication must be shared between system connectivity, e.g. socket-to-socket CPU links and storage-area-network links, and the memory interface, roughly half of the off-chip communication power budget is available for the memory interface. In summary, these system assumptions require an off-chip memory bandwidth of 96 Tbit/s within an energy budget of 10 W. It is first interesting to compare these requirements with estimates for what electrical off-chip interconnect may be able to provide for such a system.

## 2.2. Future scaling of electrical core-to-memory links

The fundamental challenge in off-chip communications is that as the size and power dissipation of the computation and memory devices have shrunk with semiconductor scaling, the macroscopic electrical connections have largely stayed the same. As the system is scaled to fit more computational and memory devices within a fixed area and power budget, these macroscopic electrical links connecting them become a bottleneck for system performance. Although it is generally a foolish bet to place a limit on achievable electronic circuit performance, some general targets need to be set as will be outlined in the following three sections.

### 2.2.1. Channel Bandwidth

Although the limited bandwidth of the off-chip electrical channel is often given as a motivation for the transition to photonic interconnect, we do not intend to operate the photonic interconnect at a higher signaling rate than the electrical interconnect that is being replaced. The reason for this is that the practical clock rate of the interconnect to operate with high energy efficiency is limited to a low multiple of the computation clock rate [157]. To push the clock rate higher, complicated clocking circuits must be added to the serialization / deserialization (SERDES) circuits of the link and would dramatically increase the power dissipation above target levels. By setting the maximum serialization rate to 4x the core clock

for both the electronic and photonic links considered in this thesis to maintain energy efficiency, the data line rate will be set at 8-12 Gbit/s.

### 2.2.2. Channel Density

The fixed channel bandwidth is expected to be more than offset by a substantial increase in parallelism enabled by optics. The off-chip electrical channel density currently faces hard packaging limits that are the largest impediment towards increasing the off-chip bandwidth of processors. The dominant packaging technology for CPUs, flip-chip bonding, will enable scaling-limited electrical connections at a pitch of approximately 100 μm in a 2D array. This density has been supported by low-cost packaging technology at approximately $0.01 per pin [158, 159]. As a result, a yield-limited 1 $cm^2$ die is limited to approximately 10,000 pins to be connected in a $100 package [160-162]. Considering that current microprocessor packages in high volume products have been limited to roughly 4000 pins and slowly growing, this estimate appears to be an optimistic upper-bound. From this limit, it is possible to estimate the total available channel count by first eliminating the required power supply pins. Assuming a current limit of 100 mA per pin in a fine pin-pitch package [140], the 100 A required for a 100 W processor running at 1 V consumes 2000 pins. Of the remaining 8000 pins, it is further assumed that half of the total signal pins may be used for the memory interface. Since off-chip electrical channels must be differential at high-speed for signal integrity concerns, this yields a rough assumption of 2000 electrical channels. This rough analysis, coupled with the above line rate target, caps the off-chip electrical bandwidth at 20 Tbit/s bidirectional (10 Tbit/s each way). Compared to the system target of 96 Tbit/s, this is seen to be well below a quarter of the target bandwidth with very optimistic assumptions.

### 2.2.3. Energy Efficiency

Energy efficiency of future electrical off-chip links is likely the most difficult parameter to estimate. It may seem that since this has been a critical parameter for many years and that the materials and techniques used to make the electrical channel have remained constant that therefore performance would be saturating at a minimum value. Instead, designers have scaled the capacity of individual memory channels, continually improved drive circuit topologies as well as replaced analog

equalization technology with lower power digital techniques. Complicating the matter further, results presented by academic researchers often do not address the complexities of a real system that would result in increased power dissipation. To baseline the technology prediction with past progress, a summary of energy efficiency for realized and proposed commercial memory technologies is presented in Table 2.1. The pace of progress has been roughly a 2x improvement in energy efficiency every five years. A reasonable target for near term commercial systems is therefore on the order of 5 pJ/bit to 10 pJ/bit.

| Initial Volume Manufacturing | Memory Technology | Link I/O Energy |
|---|---|---|
| 2001 - 2003 | DDR-333 | 257 pJ/bit |
| 2005 - 2006 | DDRII-667 | 121 pJ/bit |
| 2009 - 2010 | DDR3-1333 | 64.7 pJ/bit |
| 2012 + (target) | DDR4-2667 | 38.7 pJ/bit |
| 2013 + (target) | HMC | 10.8 pJ/bit |

**Table 2.1** Memory energy efficiency as a function of technology generation and introduction year. Source: Micron Technology

Academic papers have routinely reported power per bit numbers in the low pJs for several years [157, 163-168]. Recent work has reported power levels approaching 0.2 pJ/bit for specific channel assumptions at lower data rates [169-171]. However, the best academic results would have to be significantly modified to handle the poor electrical channel present in a massively parallel microprocessor package. Given this outlook, it may be reasonable to assume as an upper limit that deployed systems may achieve the one to two orders of magnitude improvement to reach 0.5 pJ/bit within the next 5 years. Perhaps a more realistic limit would place 10 Gbps link performance in the 1 pJ/bit to 2 pJ/bit range for a high capacity system where the electrical channel is significantly degraded. To meet the total system target bidirectional memory bandwidth of 96 Tbit/s, these links would burn 24 W for the most optimistic 0.5 pJ/bit case. The optimistic estimated electrical link performance still falls over a factor of two short of the desired system targets. Again as was seen for the packaging density case, realistic limits place the electrical links at least a factor of four away from targets required for system enablement.

### 2.2.4. Conclusions and Outlook

It is clear under the assumptions of this rough analysis that electrical off-chip memory links will have a difficult challenge to meet the required mid-term compute system requirements. Although the estimations used in the analysis are crude, the end implications are compelling. Since both density and energy limits are being encountered simultaneously, there is no engineering tradeoff available to alleviate a single limit. If only one was constrained, the line rate could be either increased or decreased beyond the roughly optimal 10 Gbit/s assumption to trade, for example, higher energy at higher line rates to alleviate a density bottleneck. The magnitude of the electrical link performance shortfall completely forbids an energy efficient implementation at a 48 Gbit/s off-chip data rate in any near-term system. Equally compelling is the acknowledgement among major semiconductor manufacturers that there will be a time when photonics will break the processor package boundary at some point in the future. Less agreed upon, however, is the exact manner in which this photonic integration will occur. In this thesis, we advance the most radical manifestation of this approach in which the photonics share front-end silicon area with the state-of-the-art electronic devices. To attempt to address some of the concerns raised when photonic interconnect is brought up to the microprocessor community, the remainder of this chapter will be devoted to developing accurate system metrics of photonic devices integrated utilizing this approach.

## 2.3. Monolithic Photonic Interconnect

To maintain a simple overall system architecture, I will consider the simplest case of photonic I/O in which point-to-point electrical links to the DRAM memory chips are replaced by photonic interconnect. Each memory photonic access point is integrated within the standard electronic memory controllers (MCs) that are distributed across the processor die as shown in **Figure 2.2**. The photonic access point consists of 3-fibers: one to deliver the CW wavelength comb to the memory module, one to transmit upstream data, and one to receive downstream data. This 3-fiber data bus is then connected to a DRAM memory module that can support high sustained I/O bandwidth such as the recently announced hybrid memory cube (HMC) architecture shown in **Figure 2.3**. The example chip floorplan has been simplified to include only 8 wavelengths and 8 memory module connections implemented as 4 horizontally stacked 8-fiber ribbon connections. The numbers of

wavelength and memory channels do not necessarily need to be equal as they happen to be in this example. The passive power splitting can deliver the wavelength comb to an arbitrary number of waveguides for memory access. A unique feature of the proposed floorplan is that the entire on-chip optical network may be implemented without any waveguide crossings. The zero-crossing topology can be extended to a nearly arbitrary size network. Placing the input optical coupling ports in the center of the processor die minimizes the on-chip routing distance. The zero-crossing constraint then only trivially increases waveguide routing distance by local obstructions of the vertical grating couplers.

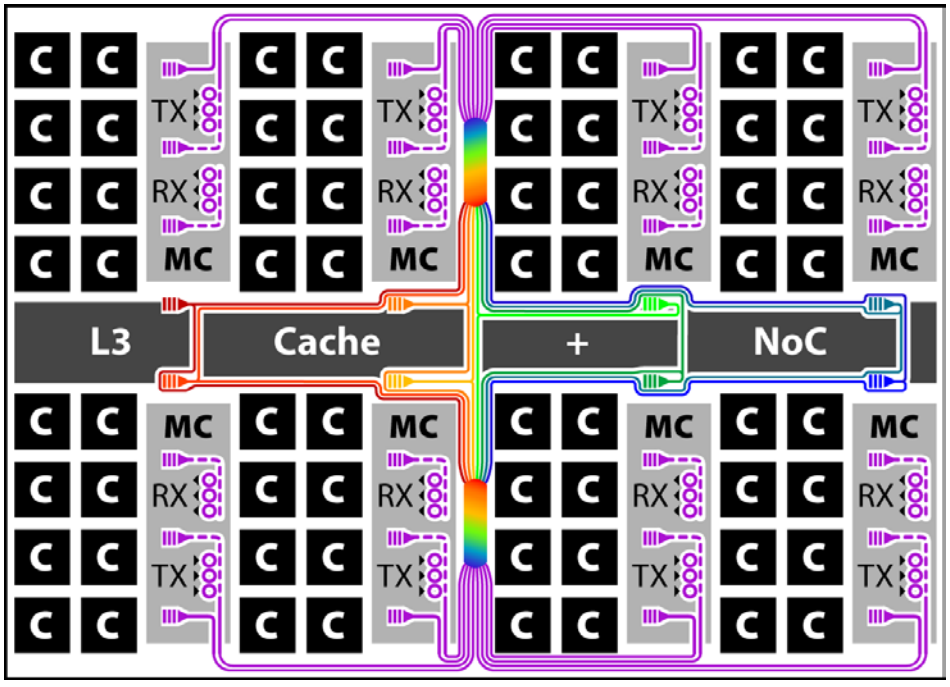


**Figure 2.2** Example physical floorplan for the integrated photonic link infrastructure. Single wavelength optical power supply interfaces are located in the middle of the die and passively split for distribution as a wavelength-division-multiplexed optical power supply. The center region of the die is coarsely designated for the L3 cache and network-on-chip NoC in addition to all shared control and other electrical I/O.

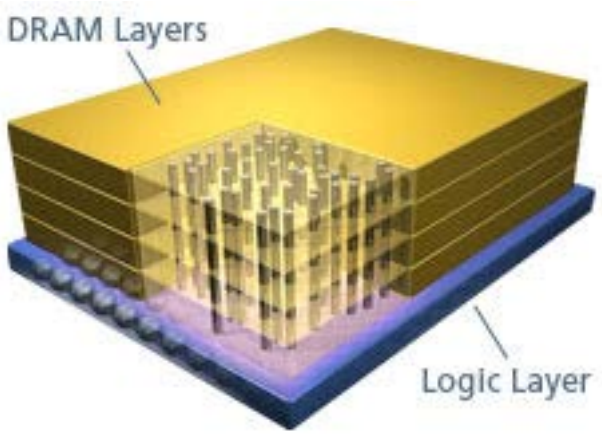


**Figure 2.3** Product cartoon of hybrid memory cube (HMC) architecture. Image source: Micron Technologies.

The major considerations that will be addressed in the following sections then include the nature of the optical packaging solution, the transmit and receive optical components and the resulting overall link metrics. For the transmit side of the optical link, the first global consideration is that the laser sources are off-chip and coupled on-chip through vertical couplers. Since each laser must be split into various waveguides and each output waveguide needs components of all input laser sources, the bulk of the multiplexing and splitting operations can be achieved in one or two star-couplers [172-174] as shown in **Figure 2.2** or any other suitable passive splitter. The demultiplexing operations must then be performed in ring resonator filters. The performance of the ring resonator devices then set the maximum wavelength channel count that may be demonstrated in the technology platforms developed in this thesis. The resonant frequency of each ring resonator must then be precisely-controlled with thermal feedback. The power required to heat the rings must be included into the overall link energy budget. The ring modulators are narrow-band devices that can be electronically-driven to either block or allow the light for a single wavelength channel to pass. The receive side of the link then requires the aggregated wavelength channels to be demultiplexed in a similar ring resonator filter bank before the light may be absorbed with integrated detectors. For both the transmit and receive sides of the photonic link, the energy dissipated for the driver and receiver circuits will be considered in the context of achievable device capacitance.

### 2.3.1. Packaging platform

To make meaningful statements about an I/O technology, a packaging platform must be defined. In this thesis, we base our analysis on normal incident fibers integrated into the flip chip package in a multi-fiber ribbon connector at the center of the die. This packaging strategy allow for the possibility of passive alignment using the micron-scale surface tension alignment of the solder ball array. Low cost, high fiber count multi-row fiber ribbon connectors have already been developed for telecommunication applications requiring large fiber parallelism such as fiber-to-the-home. Full development to a manufacturable part is required to ensure the practicality of this approach. Early analysis shows that this approach is feasible if organic packages with their accompanying warping and high coefficient of thermal expansion are avoided in favor of ceramic packages [175]. Although high electrical pin counts has typically prohibited the use of ceramic packages within the last 10 years, the dramatic reduction in electrical connections required as a result of the photonic integration will remove this constraint. Further discussion of envisioned packaging technology including initial prototype attempts is included in the future work section of Chapter 6.

At the highest level, we will assume that the fiber used will be Corning RC FLEX 1060 that allows tight bending radii, small mode size and low loss across the 980 - 1600 nm window. Additionally, the 80 μm cladding diameter provided by this fiber allows for high-density connections. We will further use a connection pitch equal to the coating diameter, or 175 μm for our analysis to avoid the complicated fan-out strategies required for tighter ribbon pitches. Compared to the scaling-limited, differential electrical bump-bond array at a 100 μm, the off-chip access density is ~1.5x worse than the electrical case. We will see, however, that additional degrees of freedom obviate this connector density constraint to the point that the available optical connection sites will not be filled for near term systems. To match available optical fiber ribbons, we will consider optical connectors that contain one or more rows of 48-fiber ribbons. Since the total width of such a connector is approximately 8.4 mm, the connector size is well matched to the expected die area. The total number of required connections is therefore a multiple of 48 fibers, to be determined by the per fiber bandwidth density to yield the system total of 96 Tbps.

### 2.3.2. Wavelength Division Multiplexing

The high optical bandwidth of the optical channel enables the chief density advantage of optics: wavelength-division multiplexing (WDM). In this approach, multiple carrier wavelengths, each carrying separate data signals, are aggregated onto one waveguide to reduce the density of off-chip interfaces. It is important to first explain the relevant technologies required to examine the number of available channels as well as the optical loss and physical area utilization of the required components. The target specification that governs how closely channels may be packed together is the allowed optical cross talk between adjacent channels. This number is typically set to be an allowed cross talk ceiling of 20 dB. In the context of relatively power-equalized channels utilizing low extinction modulators, this represents a conservative bound. In this thesis, I have demonstrated an 8-channel WDM filter bank utilizing first-order ring resonator filters that meets all desired optical specifications that will be presented in Chapter 4. If such a filter technology was utilized for the discussed core to memory network, the total number of fibers required, without considering fibers used for optical power supply delivery, would be 1000. Even with 48-fiber ribbons, 21 rows of fiber ribbon connectors would be required. Since the current state-of-the-art for fibers per connector is approximately 144 [176] and standard MPO connectors contain up to 72 fibers, the connector required to provide such optical packaging technology would be impractical at best. It is therefore of interest to consider what may be achievable given the current technological constraints.

#### *2.3.2.1. Provisionable optical bandwidth*

The provisionable optical bandwidth is set by the free spectral range (FSR) of the ring resonator filters used to demonstrate the wavelength division multiplexing filter banks. As shown in Equation 1.34, the FSR is proportional to the wavelength squared divided by the group index and ring radius. Using the nominal design dimensions derived in Chapter 1 for our target SOI-CMOS process, the bend losses as a function of radius for various wavelengths are shown in **Figure 2.4**. Fitting the minimum radius as a function of wavelength and using the correct group indices, the maximum achievable FSR as a function of wavelength is shown in **Figure 2.5**. As a point of reference, 4 THz at a center wavelength of 1250nm corresponds to a

20.8nm wavelength range. The number of wavelength division multiplexing filter channels that may fit within this span may then be determined.

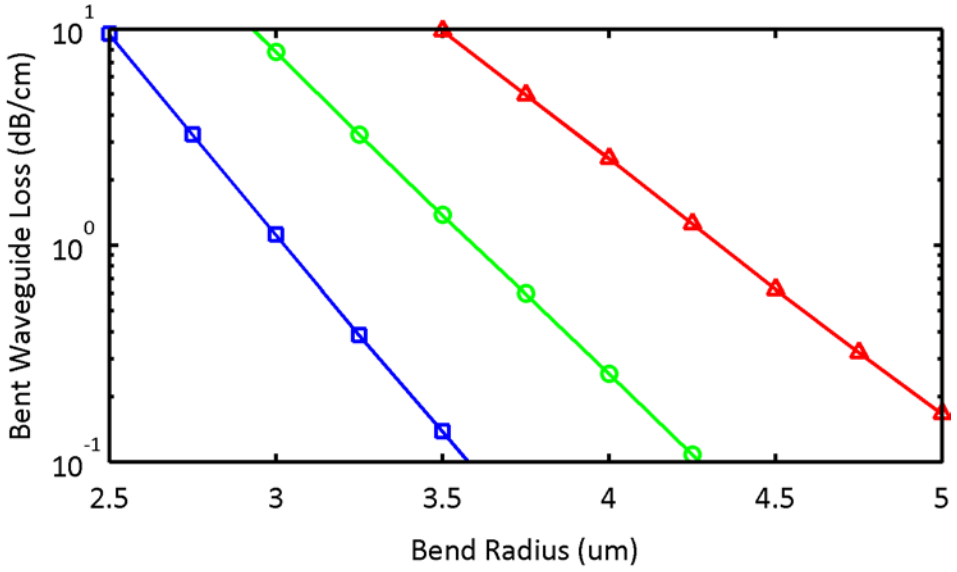


**Figure 2.4** Bend loss simulation detail for a 500nm width waveguide at wavelengths of 1200nm (blue squares), 1250nm (green circles) and 1300nm (red triangles).

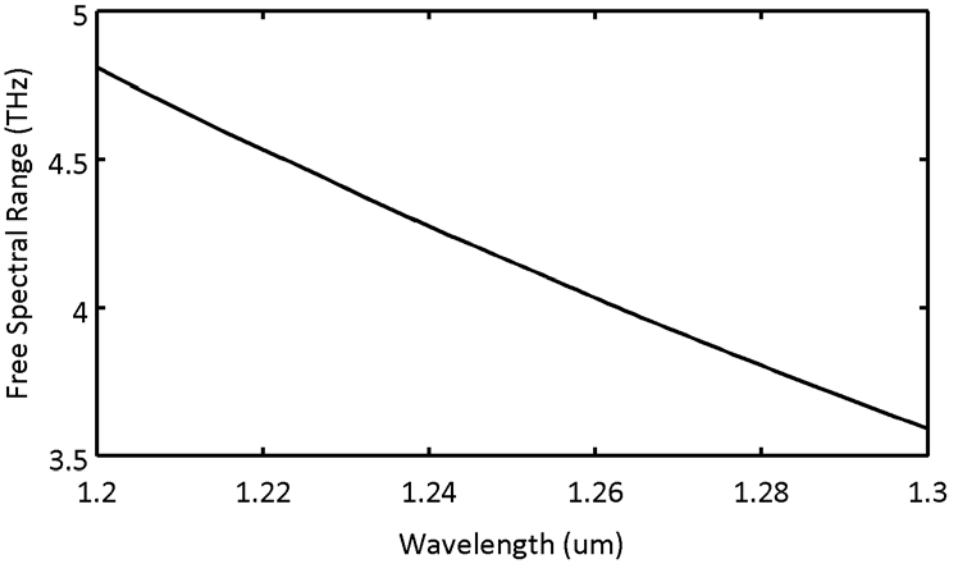


**Figure 2.5** Maximum free spectral range (FSR) as a function of wavelength for a 1 dB/cm bend loss limit for a 500nm width waveguide.

### *2.3.2.2. First-order filter realizations*

First-order (single ring) microring resonator filters represent the simplest and smallest optical component available for wavelength division multiplexing. Using

the technology parameters presented in Chapter 1, we may now analyze how well such a filter may perform for the wavelength division multiplexing application within the system. To target operating wavelength ranges between 1200nm and 1250nm, the nominal bend radius for the filters will be set to 4μm enabling roughly 4 THz of usable optical bandwidth. We will set the optical bandwidth to minimize any bit error rate penalties for 10 Gbit NRZ data transmission. A generic constraint for optical data transmission is typically given by specifying a filter full width at half maximum (FWHM) bandwidth that is 1.5 times the symbol rate or 15 GHz in this case. This general metric, however, does not consider the filter shape to truly assess the resulting data impairment.

To analyze the required bandwidth of a first-order ring resonator, we will instead consider the 90% bandwidth of the filters. Since the optical data bandwidth is well confined within 7 GHz for such a data stream, we will set the constraint that the 90% transmission bandwidth should be greater than 7 GHz. As shown in **Figure 2.6**, a 1.8% power coupling coefficient to the bus waveguides produces a ring filter with a 7.2 GHz wide 90% pass band and a FWHM of 22 GHz. Taking into account the measured propagation loss of 3 dB/cm that will be presented in Chapter 4, the drop loss for such a filter is 0.4 dB with an off-resonant insertion loss of $4 \times 10^{-4}$ dB.

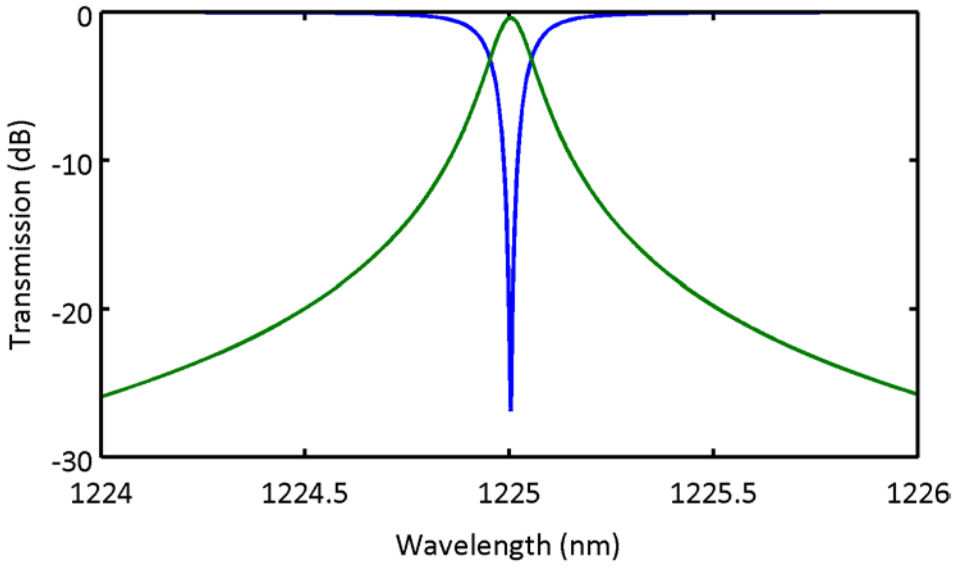


**Figure 2.6** First-order ring resonator through (green) and drop (blue) transmission functions for a 4μm bend radius, with 3 dB/cm waveguide and a 1.8% power coupling coefficient to the through and drop bus waveguides.

The packing density of the filterbank is then set by the allowable level of adjacent channel cross-talk. In standard telecommunication systems, most optical components are required to meet strict requirements for adjacent channel cross-talk suppression of 50dB or more. This requirement is set by the possibility of large optical power differences between adjacent wavelength channels. In the chip-to-chip interconnect scheme proposed in this work, all wavelengths on a given waveguide are transmitted or received at a single location for all channels and traverse identical optical paths. This system design feature ensures equal power levels across all wavelength channels. The extreme constraints of the telecommunication systems may therefore be significantly relaxed. Instead a proper specification can be determined by considering the impact of cross-talk from an aggressor channel transmitting a '1' in a channel receiving a '0'. The important metric is then the relative strength of the cross-talk to the extinction ratio of the modulation. If we consider modulation extinction ratios of 10 dB, a 20 dB cross-talk suppression specification ensures that the aggressor '1' is an order of magnitude suppressed relative to the received '0' level. Although this is the constraint that we will consider in the following analysis, even this limit is relatively arbitrary. For a given system implementation the resulting signal-to-noise ratio from the combination of receiver technology, modulator extinction ratio, modulator cross talk, filter channel bandwidth and filter channel cross-talk should be considered jointly to optimize performance.

Examining the transmission characteristics of the target filter design, the cross talk suppression of 20dB may be achieved for an adjacent channel spacing of 110 GHz. Within the provisionable optical bandwidth of a single FSR, the 32 channel filter bank may be fabricated by stepping the radius of adjacent filters by 3nm as shown in **Figure 2.7**. The transmission matrix model including the full lossy dispersion relation of the waveguide illustrates many relevant non-idealities that a fabricated filter bank would possess. The simulated on-resonance insertion loss of 0.44 dB is increased by 0.04 dB relative to an isolated filter in agreement with the target cross talk specification of -20 dB. Second, for a fixed radius offset, the adjacent channel spacing changes by approximately 2% between the highest and lowest wavelength channels due to waveguide dispersion. More importantly, since the group index of the waveguide changes significantly over this wavelength range, the effective free spectral range also changes. As a result the adjacent channel separation between the highest and lowest wavelength channels out of band filter

replicas changes by over 60% as is clearly evident in **Figure 2.7**. The 32 channel filter bank target still yields desirable system transmission performance while including all of these non-ideal characteristics.

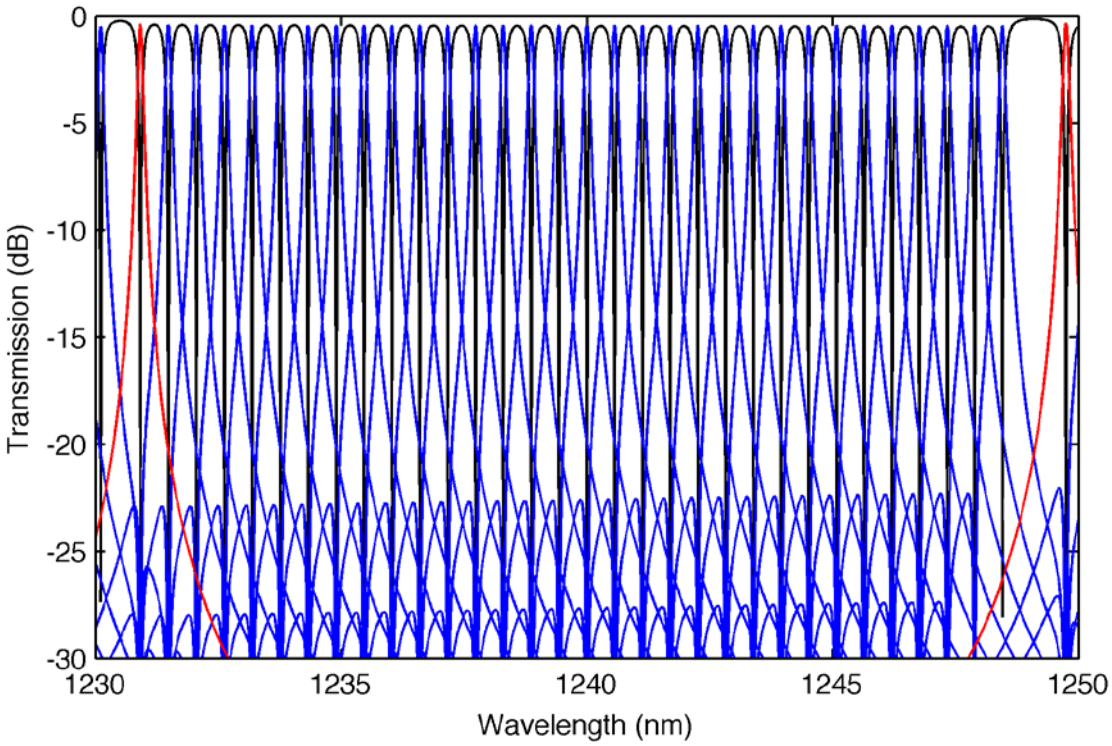


**Figure 2.7** 32 channel filterbank simulated with 3nm radius step per channel.

The factor of 4 increase in filter bank channel count relative to what has been demonstrated in this thesis reduces the minimum number of connecting optical fibers to the CPU die to 250. This is still a large number of fibers, but there is sufficient real estate on-die at the target pitch to enable over an order of magnitude more fiber connections. Additionally, the fiber count is comparable to the state-of-the-art in high-density singlemode fiber connectors.

#### *2.3.2.3. Higher-order filter realizations*

Similar to higher-order electronic filters, sharper transitions from the pass-band to the stop-band are achievable in coupled resonator, or higher order, filter designs. The trade-off is primarily between the filter complexity and number of optical fibers required to achieve a specified off-chip bandwidth. The difficulty in assessing the merits of a given design are therefore split between the device performance in terms of complexity, tuning power and drop-loss and the cost of the packaging.

Further, from a system perspective, each optical fiber is expected to be connected to a single DRAM stack. The DRAM chip will have to provide high utilization of the provisioned per-fiber bandwidth. If 32 wavelength channels are utilized, each DRAM stack will be provisioned for 320 Gbit/s of read and 320 Gbit/s of write bandwidth. For comparison, a DDR3-1600 module has a data bandwidth of up to 100 Gbit/s today. Proposals for future stacked memory solutions such as the Hybrid Memory Cube have predicted provisioned bandwidths up to 1 Tbit/s. Significant computer architecture considerations are required to address such tradeoffs. In light of this uncertainty, first-order resonators will be the focus of this work.

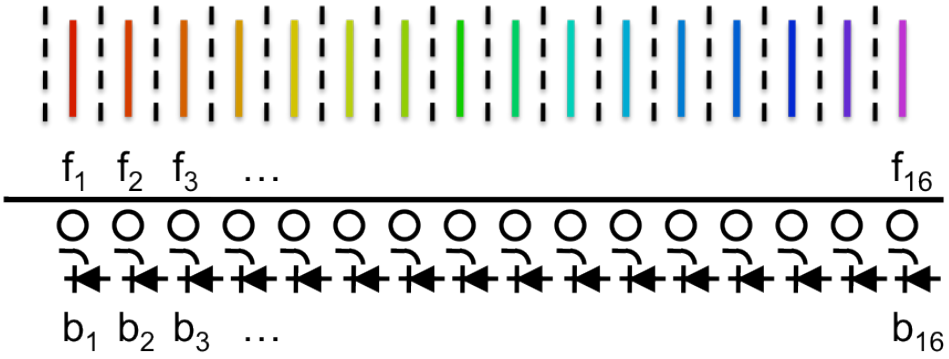


**Figure 2.8** Ideal matching of filter bank resonance frequency to a receiver demultiplexing filter bank to convert incoming frequencies f1 through f16 into parallel data bits b1 through b16.

### 2.3.3. Channel locking and thermal stabilization

A single ring filter for a wavelength division multiplexing filter bank may not be deterministically designed to be resonant at a specific absolute frequency. Instead, the fluctuations in silicon layer thickness and patterned waveguide width alter the exact group index and therefore the resonance frequency of the fabricated device. The variation in these parameters while large in absolute uncertainty, may be very small relative to other nearby fabricated devices on the same wafer. For this reason, it is possible to match or offset filter resonant frequencies with high precision. As discussed further in Chapter 4, local frequency relative variations are below 50 GHz on average. The ideal modeled filterbank of **Figure 2.7** would then have channels randomly shifted relative to each other to a small degree and a large uncertainty of what filter channel resonance is closest to a specific wavelength. However, if the filter channels are designed to completely fill the free spectral range, the resonance aliasing will distribute a channel to be nearby each desired wavelength.

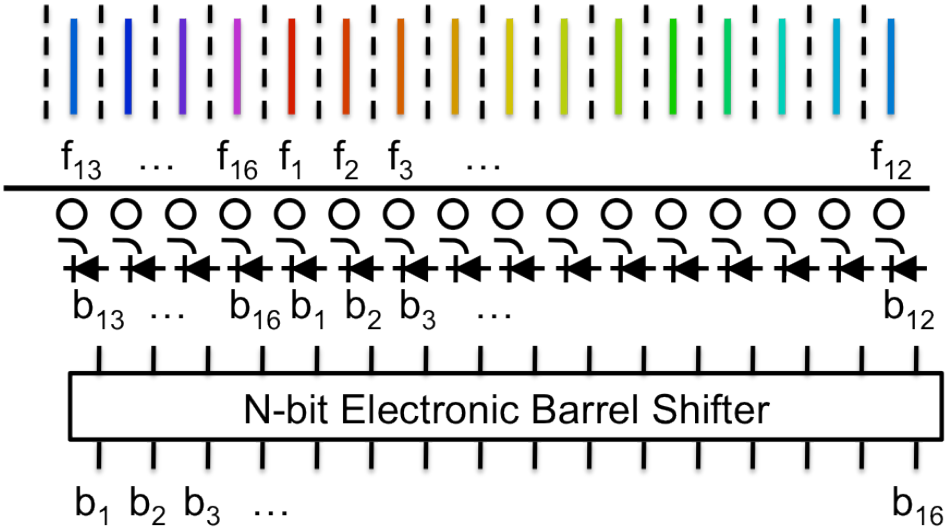


**Figure 2.9** Filter bank receiver with aliased channels due to fabrication variation. Channel offsets can be corrected by the use of the electronic barrel shifter.

The total data link can greatly reduce the burden of correcting each wavelength channel if nearest neighbor wavelength grid locking is allowed for the filter channels. This may be achieved without loss of performance in point-to-point links where all wavelength channels are dropped at a single physical location. To understand this principle, first consider the ideal receiver filter bank shown in **Figure 2.8** that is fabricated by offsetting adjacent channel radii. Each filter bank output channel will receive bits for the corresponding wavelength channel analogous to a multi-wire, parallel electrical bus. The good relative and poor absolute dimensional control will result in the filter channels aliasing over the free spectral range as shown in **Figure 2.9**. If no further provisioning is added to the system, the bits at each filter bank output channel will not correspond to the data from the intended wavelength channel. This simple, deterministic channel offset can then be corrected with digital electronics. The barrel shifter pictured in **Figure 2.9** can then correctly realign the data bits for the expected channel ordering. The control signal for this barrel shifter must then be initially configured during system initialization. The relative bit ordering between the nearest neighbor wavelength-locked transmitter and receiver must be established at system initialization to set the barrel shifter offset.

The locking of the fabricated filters to this nearest neighbor wavelength grid can then be accomplished through thermal tuning. The impact of temperature on the resonant frequency of the resonators is dominated by the thermo-optic coefficient

of the silicon core material. Since the refractive index of the material increases with temperature primarily due to a reduction in the semiconductor's bandgap, the resonance frequency reduces or "red shifts." The measured effective thermo-optic coefficient has varied from 7.9 GHz/°C to 9.6 GHz/°C on chips fabricated over the course of this thesis and presented in Chapter 4. The measured value for the SOI-CMOS platform was 9.6 GHz/°C and will be the value used for all further tuning efficiency analysis. If efficient or directly-integrated heaters are added to the filterbank, the net electrical tuning efficiency will be the product of the effective thermo-optic coefficient and the thermal impedance. The best measured heater thermal impedance achieved in this work is 44 °C/mW to yield a tuning power of 2.4 μW/GHz that is comparable to record results from the literature. For the sake of coarse estimation, the average tuning power per ring filter can be calculated using the channel spacing and process variation. Since the thermal tuning can only operate in one direction, i.e. red shift the filter, twice the stochastic variation estimate should be used. Additionally, since the filters must align to a predetermined wavelength grid, an average tuning range of one filter channel spacing frequency should be considered. The resulting total frequency shift is then 210 GHz to yield an average power estimate of 0.5 mW per filter. The control electronics required to achieve the locking is then assumed to be insignificant in comparison since the control loop can run roughly two orders of magnitude slower than the data link. Since only one half of the optical link will be located on the processor die, a 50 fJ/bit overhead for thermal tuning energy must be added to the on-chip link energy calculation for the 10 Gbps data rate.

### 2.3.4. Modulators

The most obvious contribution to on-chip electrical power dissipation for the interconnect is then the optical modulator itself. For the depletion device that offers the lowest possible power dissipation, the modulator acts as a capacitive load on the electrical driver. It is first necessary to specify an exact target device to analyze performance. Since depletion modulators have not yet been fabricated as part of this thesis, a target device based upon the measured technology parameters will be considered. The 3 dB/cm waveguide loss of the integrated CMOS platform enables narrow optical resonances. The optical bandwidth of the resonator can then be set to a 3dB bandwidth equal to 70% of the datarate, or 7 GHz, with appropriate choices of the bus through and drop coupling coefficients.

At such narrow line widths, a small electro-optic phase shift is required to achieve sufficient modulation. Relatively low doping concentrations may then be used to form the pn-junction region. The n-type and p-type regions are then chosen to equal the existing transistor well doping concentrations of 2 x $10^{-17}$ $cm^{-3}$ that are available in the standard CMOS process. A maximum scaled-CMOS compatible voltage of 1.5V in reverse bias results in a depletion width of 175nm as calculated from Equation 1.41. This maximum depletion region width forms one logical state of the modulator. The other state is achieved by forward biasing the device below the diode turn on voltage that would result in considerable current flow. If this safe forward bias voltage is taken to be 0.7V, the minimum depletion region width is then 47nm. The depletion width variation of 128nm then represents 43% variation of the effective mode width that is approximately 300nm wide. Finally assuming a 70% silicon core confinement factor, the on and off state of the designed depletion-mode modulator is shown in **Figure 2.10**. The calculated optical performance characteristics are then a 17.6 dB extinction ratio with a 0.5 dB insertion loss. The narrow linewidth relative to the 22 GHz FWHM filter bank ensures that the modulator acts only on one wavelength channel. The total span of **Figure 2.10** corresponds one wavelength grid channel spacing of 110 GHz to demonstrate the isolation between neighboring wavelength channels.

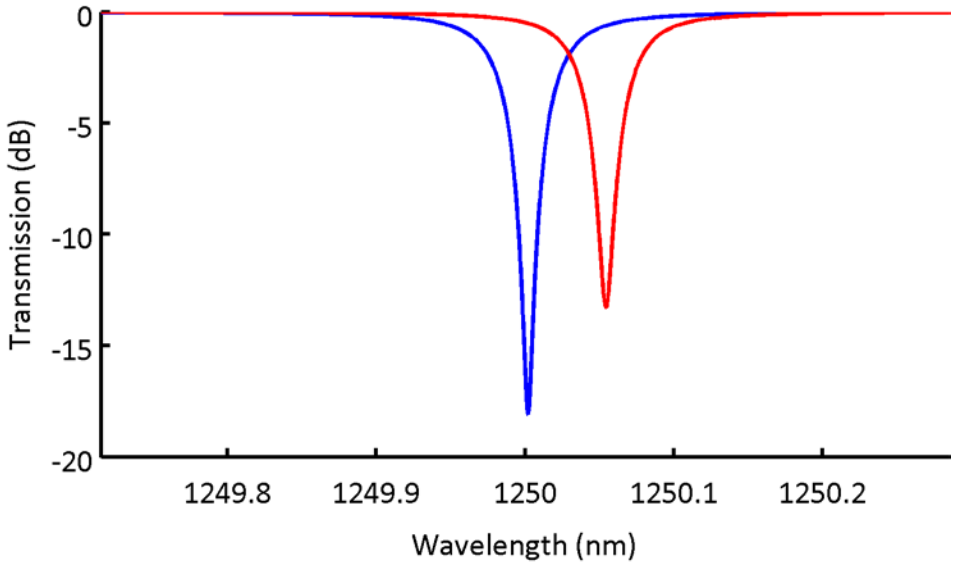


**Figure 2.10** On (red line) and off (blue line) state transmissions for depletion-mode modulator. The total wavelength span matches the 110 GHz designed channel spacing for the 32 channel filter bank. Power coupling coefficients are 0.8% and 0.2% for the through and drop ports respectively. Design input laser wavelength is 1250 nm for peak modulation.

The power dissipation of the modulator may then be determined by the total capacitance of the junction and all associated parasitic capacitances. If a planar junction model is used, the calculated capacitance for the forward, zero and reverse bias states are 4.4 fF, 1.9 fF and 1.2 fF respectively. This simple model for a parallel plate capacitor, however, neglects the fringing field lines that result from the thin junction. Rigorously calculating the capacitance for the two states including the first metal level using the Raphael parasitic modeling tool from Synopsys, forward, zero and reverse bias capacitances of 8.0 fF, 5.2 fF and 4.1 fF respectively were obtained as shown in **Figure 2.11** and **Figure 2.12**. Additionally, the wire length between the driving circuit and the modulator must be considered. If the photonic and electronic devices must be separated by approximately 10 μm for the localized substrate removal post-processing, an extra 2 fF of wiring capacitance must be added. For simplicity, we will assume average capacitances of 8.6 fF and 6.7 fF for the forward and reverse states respectively. Since the modulator will only dissipate power when it's state is toggled, the energy dissipation is multiplied by an estimated switching activity factor of 50%. The energy dissipation per bit for the modulator itself for a NRZ data stream is then given by the average of the forward and reverse states:

$$Energy_{\mathrm{mod}} = \frac{1}{8}\bar{C}_{forward}V_{forward}^{2} + \frac{1}{8}\bar{C}_{reverse}V_{reverse}^{2} \qquad (2.1)$$

For the stated parameters and assumptions, the energy dissipation for the modulator itself is then calculated to be 2.4 fJ/bit.

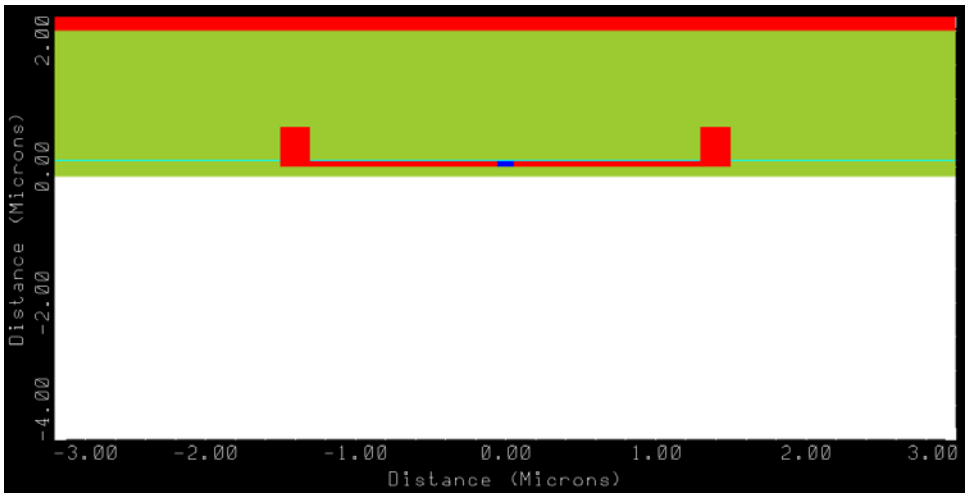


**Figure 2.11** Structure used to simulate the modulator capacitance using Synopsys Raphael RC2. The higher metal levels 2 μm away from the depletion modulator forms the ground plane in the substrate-removed photonic region.

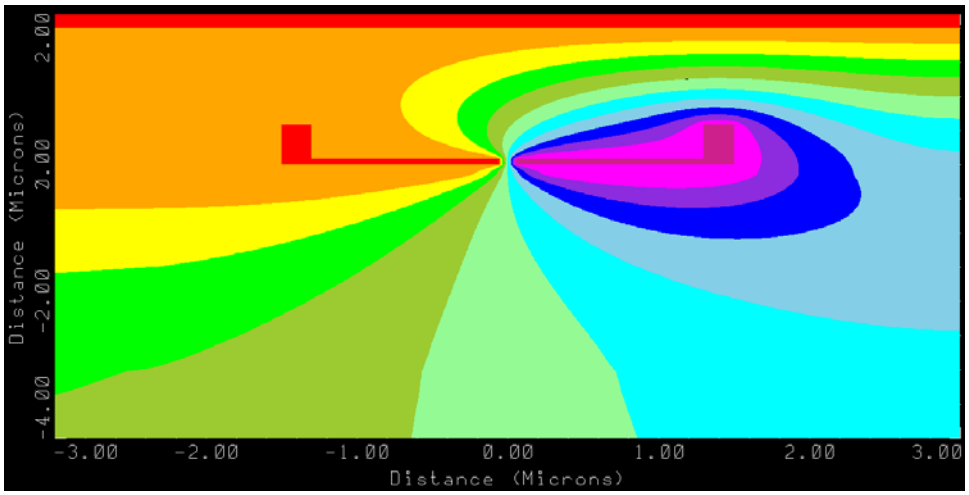


**Figure 2.12** Simulated potentials used for capacitance extraction.

Two additional components are then required to interface the modulator to the logic level output of the latched data of the microprocessor. First, the data rate of the optical interconnect is expected to be up to 4x the clock rate. Although the serialization/deserialization operation can in general require significant energy consumption due to the requirement for local clock generation, low multiplexing ratios can be achieved with simple circuits. First, doubling the data rate only requires triggering on both the rising and falling clock edges. The final data rate doubling can then be achieved by delaying both edges through an inverter delay chain by half the duty cycle. Since this delay has the potential to introduce clock jitter as the supply voltage fluctuates, it is difficult to extend this technique to higher multiplexing ratios. Since all of these multiplexing operations may be performed with less than 100 minimum width static logic transistors in the deeply scaled logic process, the power dissipation of this multiplexing is approximately 30 fJ/bit when the full parasitic capacitance of transistor gates, drains and interconnecting wires are considered.

The ~9 fF load of the modulator represents a significantly larger capacitance than the ~90 aF input capacitance of a minimally sized CMOS logic gate. This factor of 100 capacitance ratio is significantly larger than factor-of-4 fan-out ratio (FO4) that is typically used to construct CMOS logic. For the standard FO4 logic design, the electrical stage delay is ~10 ps in our technology. For the 10 Gbps data operation, it is desirable to have rising and falling edges of approximately 10 ps. The driving buffer chain can then be constructed with a FO4-loaded modulator driver. If three FO4 buffers are required to return the drive level to minimum width inverter

strength, the total buffer chain therefore represents approximately 20 fF of additional switching capacitance. Since the switching activity factor is again 50% for the NRZ data stream, the pre-buffer and driver only contributes 5 fJ/bit to the total modulator power dissipation.

The summary is that the actual modulation operation of the optical interconnect link can be expected to consume ~38 fJ/bit for such a depletion mode device designed in the thin-SOI technology platform developed in this thesis. This small number is roughly an orders of magnitude less than the electrical link target. It is apparent that other aspects of the photonic link can be expected to dominate the total energy consumption. Already, it has been seen that the thermal control of the wavelength division multiplexing filter banks more power than the fundamental transmission operation.

As a point of comparison, the predicted ~38 fJ/bit performance is significantly lower than the best transmitter efficiency of 135 fJ/bit achieved through hybrid integration [177]. The largest difference between this previous work and the predicted performance of a monolithically integrated modulator is the excess parasitic capacitance of 200 fF in the hybrid integration work from the photonic side metallization-to-substrate capacitance of 150 fF, wiring parasitic of 10 fF on each side and 30 fF resulting from the actual hybrid integration bond interface [178]. If a similar low-capacitance depletion modulator was used in a similar hybrid integration platform, the expected energy efficiency of the transmitter would rise to 109 fJ/bit by performing the same set of calculations with the additional capacitive parasitic. If the metallization-to-substrate parasitic capacitance was replaced with a 100 fF electrostatic discharge protection diode for high-volume manufacturability, a realistic link efficiency of 94 fJ/bit would then be predicted. Although the transmit energy would still be low in comparison to an electrical link, over a factor of two reduction in energy efficiency may be expected for the transmitter in a hybrid-integrated photonic interconnect platform.

### 2.3.5. Detectors

The receive operation then draws electrical power to convert the photocurrent signal that results from the incident optical signal into a logic-level data signal for use in the circuit. In terms of the on-chip electrical power dissipation, the photodetector efficiency is an irrelevant quantity for the moment. As such, the required energy is primarily a function of circuit design. A key parameter for the

circuit performance, however, is the capacitance of the photodiode. Unlike in the modulator case where a simple nearly-linear relationship exists between efficiency and capacitance, the magnitude of the device capacitance dictates available circuit topologies [179]. The most efficient designs are only practical for extremely small device capacitances.

For traditional telecommunication receivers, the optical module is constructed of several discrete modules such as the photodetector (typically III-V), transimpedance amplifier (typically SiGe), and clock and data recovery chip (typically CMOS). The power dissipation for such modules has traditionally been orders of magnitude larger than is necessary for the integrated photonic interconnect link. Recently, progress in integrated receivers has reduced the power to 2.8 pJ/bit [180]. However, even in this work, a discrete photodetector that must be wirebonded to the receiver limits scalability. Using photodetectors fabricated in a standard CMOS process for vertically-incident 850nm light, previous workers have achieved fully integrated receivers [181-184]. Due to the pF-scale device capacitance and low bandwidths of the integrated detectors, the lowest energy consumptions for full receivers are 4.7 pJ/bit at 8.5 Gbps [182] and 6.6 pJ/bit at 10 Gbps [181].

Instead of relying on large mode area detectors intended for optical fiber, the small mode size of silicon photonic waveguides may be interfaced to small integrated detectors with a small capacitance. Germanium-on-silicon detectors have demonstrated capacitances as low as 1.2 fF [185]. Capacitances in this range should enable low power receiver topologies such as sense amplifiers [179]. The lowest reported monolithically integrated device capacitance 20 fF [186], however, was combined with an inefficient transimpedance amplifier topology where the total power dissipation was measured to be 1.5 pJ/bit [187]. Other workers have achieved a low device capacitance of 10 fF [188] without integrated circuits and therefore relied on hybrid packaging to demonstrate a receiver. After wire bonding to a 90nm transimpedance amplifier, total energy efficiencies of 4.7 pJ/bit at 15 Gbps [189] and 7.7 pJ/bit at 20 Gbps [190] were measured due to packaging parasitics.

The previous state-of-the-art for an energy efficient receiver was achieved through advanced micro-bump packaging a separately fabricated germanium detector with a 40nm CMOS receiver [177]. Although the fundamental diode capacitance is only 10 fF, the bond pads add 30 fF with an extra wiring capacitance

of 10 fF [107]. At 10 Gbps data rate, the received energy efficiency was measured to be 395 fJ/bit [177]. It is unclear that this result can scale to lower values in the future in a hybrid integration technology. Additionally, traditional electrostatic discharge (ESD) protection, which would significantly increase the interface capacitance, was eliminated in this research result. Manufacturing reliability may require the addition of ESD protection diodes and increase the interface parasitic capacitance by up to 100 fF.

In this work, integrated silicon-germanium photodetectors are connected to novel sense-amplifier-based receivers through the low-metal levels of the CMOS to enable record-low energy efficiency by eliminating parasitic capacitances. A similar circuit topology was originally developed for a flip-chip bonded photodetector, but the interface parasitic prevented the demonstration of ultra-low power [191]. In our platform, the receiver energy efficiency has been measured to be 60 fJ/bit at 3.5 Gbps as fabricated in a 45nm SOI CMOS process through an integrated electrical test circuit as discussed further in Chapter 5 [179] with a photodiode capacitance estimated to be 5fF. It is expected that this number may decrease as the technology becomes more refined. The achieved data rate is limited by the timing integrity of the digital backend interface electronics. Since the efficiency increases with data rate as sources of static power dissipation are shared by an increasing data volume, further analysis will assume a 50 fJ/bit achievable performance for a 10 Gbps data rate. Given the current capacitance load of the integrated photodiode, the required received photocurrent modulation is ~1.2 μA for a 3.5 Gbps data rate. Since the sensitivity is expected to degrade with data rate, a receiver sensitivity of 5 μA in the SOI-CMOS platform will be used for further analysis. Since the DRAM process transistors typically lag CMOS device performance by approximately one technology generation, it may not be possible to achieve the same sensitivity in the memory process. As such, a factor of 2 degradation may be expected to result in a 10 μA receiver sensitivity in the DRAM process. From a CPU optical power sourced system perspective, this increased minimum receiver sensitivity on the DRAM side is offset by a lower optical power loss budget for the CPU write path. The more sensitive receivers that may be demonstrated on the CPU die must detect light that must complete the round trip communication to the DRAM module.

The photodetector efficiency for the integrated photonic system that may be ultimately achievable is the hardest parameter to estimate at the current state of platform development. Over the course of this thesis, a maximum external quantum

efficiency of 1% has been demonstrated at 1240nm in the SOI-CMOS platform for a first generation device. Since the silicon germanium fabrication module within the memory process is still under development, no detector device results are available for that platform. Proof-of-concept photodetectors in a fully-integrated CMOS process have demonstrated 0.85 A/W responsivity that may be operated with 99% of peak efficiency under short-circuit biasing as required for use in the sense-amplifier-based receiver [91]. The process flexibility of the memory integration process should enable similar results to be achieved. Therefore, 80% photodetector efficiency for the memory links will be assumed for optical link budget analysis. Since the zero-change integration method for the thin-SOI CMOS process forbids process modification, the higher background doping of the silicon germanium photodiode may limit device efficiency. As a result of this uncertainty, 50% photodiode efficiency will be assumed for the SOI-CMOS process for optical link budget analysis.

### 2.3.6. On-chip electrical link energy

The total on-chip energy consumption is the combination of thermally tuning the ring resonator filters, the modulator driver circuit and the receiver circuits. Since the 96 Tbps total off-chip bandwidth is a combination of transmit and receive links, the on-chip power dissipation is equivalent to 48 Tbps full photonic links. The total link energy budget from the previous section totals is then 188 fJ/bit including 100 fJ/bit for thermal tuning, 38 fJ/bit to drive the modulator and 50 fJ/bit to receive the data. The total on-chip power dissipation for all of the links is then 9 W to fall within the required system limit of 10 W.

It is also worth comparing the monolithic photonic integration platform to a hybrid integration photonic platform. The modulator power dissipation has already estimated to be limited to 94 fJ/bit under optimisitic assumptions. The best available receiver energy literature value of 395 fJ/bit may be able to be further reduced, but it is difficult to estimate without significantly more detailed analysis. For the moment, we will assume that this can be reduced by an to 200 fJ/bit. If the same thermal tuning efficiency is then assumed, a total link energy of ~400 fJ/bit is estimated. The total power dissipation would then be estimated to equal 19 W. Although this represents a significant improvement over the most optimistic electrical scaling assumption, it still does not directly enable the desired 6 TFLOP processor system.

The final aspect of the link energy budget that has not yet been considered is the circuitry required to recover the clock from the received data [192]. In traditional fiber optic data links, the clock is typically recovered from the data during the receive operation. The circuitry to accomplish this, however, is complex and dissipates significant power. Typical low-power reports for discrete CMOS chips to accomplish this task at 10 Gbps are approximately ~50 mW, or 5 pJ/bit [193, 194]. Since the wavelength division multiplexed data channels traverse the same physical channel, it is possible to use a single recovered clock to receive all of the data. If this 5 pJ/bit overhead was shared between 32 wavelength channels, the effective clock recovery energy overhead would be reduced to ~150 fJ/bit. This overhead would double the total link energy.

Instead, the approach that has been taken in this work is to implement source-clock-forwarded links in which one wavelength channel is used to transmit and recover a clock. The clock recovery circuit can then be implemented as an injection-locked oscillator to improve power efficiency. An injection-locked receiver has been designed and fabricated by collaborators on this project that has been measured to dissipate 47 μW [195]. The total system bandwidth may then remain unchanged by marginally increasing the data rate on the other 31 wavelength channels from 10 Gbps to 10.32 Gbps. The total effective energy efficiency for the clock-forwarded link would then be increased by only 2 fJ/bit for the rest of the data channels including the modulation energy. The total effective link energy of 190 fJ/bit would still meet the projected system targets.

### 2.3.7. Optical loss budget

Although the required optical power does not figure directly into the on-chip thermally-limited power budget, it may be undesirable or impossible to implement a optical system if the required laser power is unreasonably high. The energy to generate the photons must still be considered if the actual end goal of energy efficient computing is to be achieved in terms of power drawn from the system wallplug per FLOP. The required optical power can then be calculated by working backwards from the receiver sensitivity targets and considering the sources of optical loss. All of the relevant insertion loss metrics have been presented in the previous sections and the total link optical losses for both link directions are shown in Table 2.2 to arrive at an average link loss of 8.9 dB. Combining the link loss information with the estimated receive sensitivities, the required optical power per

link is calculated in Table 2.3. In terms of the system wall-plug efficiency, ~29 fJ/bit of power should be added to the on-chip 158 fJ/bit calculation if the lasers used are 30% efficient as will be discussed further in the next section. Considering the total system, the average optical power of 87.4 μW per link, each single wavelength source laser is required to output only 26.2 mW. This allows the power in each waveguide to stay below the nonlinear absorption threshold. The optical power is also well within the power range of widely available laser sources.

| | Coupler Insertion Loss (dB) | Splitting Excess Loss (dB) | Modulator Insertion Loss (dB) | Waveguide Propagation Loss (dB) | Receiver Demux Loss (dB) | **Total Link Loss (dB)** |
|---|---|---|---|---|---|---|
| Upstream | 3 | 1 | 0.5 | 3 | 0.4 | 7.9 |
| Downstream | 5 | 1 | 0.5 | 3 | 0.4 | 9.9 |

**Table 2.2** Integrated photonic link optical loss budgets for upstream (data from CPU to memory "write" path) and downstream (optical power from CPU to memory module, data from memory module to CPU "read" path).

| | Photodiode Efficiency | Receiver Sens. (μA) | Link Optical Power (μW) | Energy at 10 Gbps (fJ/bit) |
|---|---|---|---|---|
| Upstream | 0.8 | 10 | 77.07 | 25.69 |
| Downstream | 0.5 | 5 | 97.72 | 32.57 |

**Table 2.3** Total optical power required per link. Energy per bit for wall-plug analysis is included by assuming a laser efficiency of 30%.

### 2.3.8. Optical power supply

As discussed at the start of this section, the nominal architecture proposal is to utilize an array of single wavelength laser sources that are independently coupled onto the CPU chip. Cheap, highly-reliable DFB lasers have been developed near the target wavelength for the upstream data link in fiber-to-the-home (FTTH) applications. An example widely available part is the Mitsubishi ML725AA11F InGaAsP multiple quantum well (MQW) laser diode [196]. It is designed to be operated in an uncooled environment between -40°C and 85°C for an output power of 10 mW. Each laser is guaranteed to operate at a single wavelength, but the exact wavelength is not specified within a 40nm range due to fabrication variance. The wavelength comb source may then be constructed by binning the fabricated DFB lasers and constructing a comb of lasers spaced by approximately 110 GHz each.

Since no global constraints on absolute wavelength within this 40nm range are required by the construction of the system, the comb laser source should be able to utilize all fabricated lasers. The lasers may then be allowed to drift with temperature at a rate of approximately 0.1 nm/°C as long as the system packaging maintains similar temperatures for the laser diode bars within approximately 1°C.

The laser efficiency is an important metric when considering the link wall-plug efficiency. As seen in the previous section, 30% wall-plug efficiency yields sufficient link energy efficiency to impact the total link energy by less than 20%. For the uncooled MQW lasers such as the Mitsubishi DFB, a total efficiency of approximately 35% is achieved at 25°C [196]. Since the laser is uncooled, the efficiency significantly degrades at higher temperature due to the limited energy separation in the quantum well active region and increased non-radiative recombination pathways. The same MQW laser efficiency at 85°C is only 16% [196]. This degradation of laser efficiency with increasing temperature can be greatly reduced by using quantum dot gain media that have been demonstrated to work well at the wavelength range of interest [197].

In systems with more complicated optical networks and therefore higher optical powers, the laser efficiency may become a more important parameter for total system energy efficiency [198]. The highest wall-plug semiconductor laser efficiencies of greater than 70% today are found in the high power lasers above 50W output powers [199-204]. The wavelength range of these high efficiency sources are typically focused between 700nm and 980nm for use as amplifier pump sources or materials processing. The same material systems are equally applicable for achieving high efficiencies in the 1.2μm range [112]. For smaller output powers, up to a 63% wall-plug efficiency has also been achieved in a single transverse mode design suitable for coupling to a single-mode fiber [205].

Recently, significant concern has been raised within the supercomputer community regarding the reliability of the VCSELs that are being deployed for rack-to-rack system connectivity. Although each VCSEL is a highly reliable component, the vast number of devices required has made the aggregate failure rate a legitimate system concern. Briefly, failure rates of most semiconductor components are specified by two related metrics: failures in time (FIT) and mean time between failures (MTBF). The FIT rate is calculated by measuring the number of failures observed relative to the total number of device-hours tested relative to 1 billion device-hours:

$$FIT = \frac{\#\,Failures}{\#\,Devices} \cdot \frac{1 \times 10^9\ Hours}{Hours\ of\ Operation} \tag{2.2}$$

The MBTF is then expressed as the lifetime for a single device by the FIT rate:

$$MTBF = \frac{1 \times 10^9\ Hours}{FIT} \tag{2.3}$$

The FIT rate for a 10 Gbps VCSEL has been measured by Finisar to be approximately 2.3 [206]. At first, this specification sounds very reliable when compared to the 19-23 year MTBF, which corresponds to a ~5000 FIT rate, of an Intel server motherboard [207, 208]. The total failure rate, however, scales linearly with the number of devices. To enable the 96 Tbps total bandwidth per motherboard, 9600 VCSELs are required to yield a MTBF of only 2.3 years. The problem for the supercomputing system can be seen by considering the case of IBM's recently canceled Blue Waters supercomputer that was scheduled to use 1 million total VCSELs. The expected MBTF for VCSELs in the entire system would only be ~18 days. It is also important to point out that the FIT rate of VCSELs scales super-linearly with data rate [209]. It is therefore impossible to solve the reliability problem by reducing the number of VCSELs for the same total data rate.

The expected reliability for the laser sources required for the proposed silicon photonic interconnect architecture can then be analyzed in a similar context. As an example reliability specification for uncooled 1310nm edge-emitting lasers, CyOptics reports a reliability of less than 15 FIT as measured in 200 billion field service hours [210]. The measured reliability includes directly-modulated lasers that expected to be less reliable than the CW operation required for the silicon photonic source lasers. Even using the 15 FIT estimate, the system reliability is dramatically improved relative to the VCSEL case. The 96 Tbps system bandwidth is enabled by only requiring one laser for each of the 32 wavelengths. The aggregate MTBF for the system is then 237 years and an order of magnitude longer than the published Intel server motherboard MTBF.

There are several other possible implementations for the multi-wavelength optical source. One interesting option is to use the multiple longitudinal modes of a spatially-hole-burned Fabry-Perot (FP) laser. Since the active-layer diffusion is suppressed in quantum dot (QD) gain media, excellent comb-laser characteristics have been obtained using QD-FPs. [197]. Alternatively, a photonic integrated circuit (PIC) source may be used to combine multiple on-chip lasers together into a single

output fiber [211]. The total power out of one off-chip source may then be split to feed the various data channels. Yet another potential solution is to generate an on-chip wavelength comb through non-linear processes from a high-power off-chip single-wavelength laser source [212]. Many technological considerations may influence the effectiveness of these approaches including efficiency, reliability and non-linear absorption in on-chip waveguides.

## 2.4. Future Core-to-DRAM Photonic Networks

The photonic interconnect scheme presented to this point simplifies the total computer architecture in several important aspects. First, careful attention has not been paid to the memory side of the link. Second, the on-chip network has not been considered in conjunction with the photonic-network. Finally, no attention has been paid as to how the network can be expanded to a multi-socket system. I will briefly consider each of these issues, referring to published literature for more detailed analysis where available.

The memory side of the photonic link in the presented architecture only consists of three fiber connections. The input CW optical light is modulated with the upstream data and the downstream data is received all at a centrally-located point on the DRAM module. This scheme does not leverage the photonic interconnect to distribute the data across the DRAM module. Since the primary energy dissipation in the photonic network is contained at the transmit and receive points, energy-efficient data transport across the large DRAM die may allow for further system energy minimization [213]. Even with the simple point-to-point interconnect proposal presented here, however, significant memory system benefit may be gained from the photonic integration. In the case of the hybrid memory cube DRAM proposal shown in **Figure 2.3**, the off-chip electrical interface layer requires a separate logic chip to be added to the DRAM memory stack. For a fixed layer count, this constraint reduces the memory capacity of a single module. Since the small footprint of the photonic interface module allows for integration with the DRAM modules themselves, the photonic interconnect integration may significantly increase the memory capacity of the total stacked chip. Additionally, the removal of the logic chip from the proposal may simplify fabrication by eliminating the requirements associated with the heterogeneous bonding of semiconductor layers fabricated using different wafer processes from different manufacturing vendors.

### 2.4.1. Memory Module Architecture

Little attention has been paid to the manner in which the photonic access network interfaces with the on-chip electrical network for the CPU side of the link. To first order, this matches a direct replacement of the point-to-point off-chip electrical network that it replaces. Additional capabilities of the photonic network include low-energy on-chip routing and independent wavelength routing. As discussed for the memory case, cross-chip routing can enable energy savings by eliminating receive and retransmission "hops." The photonic network distribution of memory access points across the die to be proximate to different cores may be a suitable architectural implementation. This does then place the burden of accessing distant memory controllers from cores not in the local area entirely on the on electrical on-chip network. It is then only assumed that the 20% of total on-chip power budgeted for the network-on-chip (NoC) would be sufficient to provide suitable communication without creating a bottleneck for total system performance.

### 2.4.2. Wavelength-Routed Architecture

In the developmental architecture work for this project, an alternative proposal that leveraged optical wavelength routing dramatically reduced this on-chip electrical network burden [1]. Instead of a single photonic access point being integrated with a memory controller and having full control of an attached DRAM module, each photonic access point controlled only a fraction of the wavelengths that would be aggregated onto the output fiber connected to a given DRAM module. The total wavelength space for each DRAM module connection is then partitioned between "groups" of cores on the chip that are locally connected together with a small-scale electrical network. To gain access to any block of data in main memory, any core must only communicate electrically within its own group before accessing the off-chip photonic link. The number of such groups then divides the effective radix of the on-chip electrical network to a manageable size for a high number of cores. As shown in **Figure 2.13** for a 16 core chip, the cores are partitioned into 4 separate electrical networks while wavelength routing still enables access to the complete memory space.

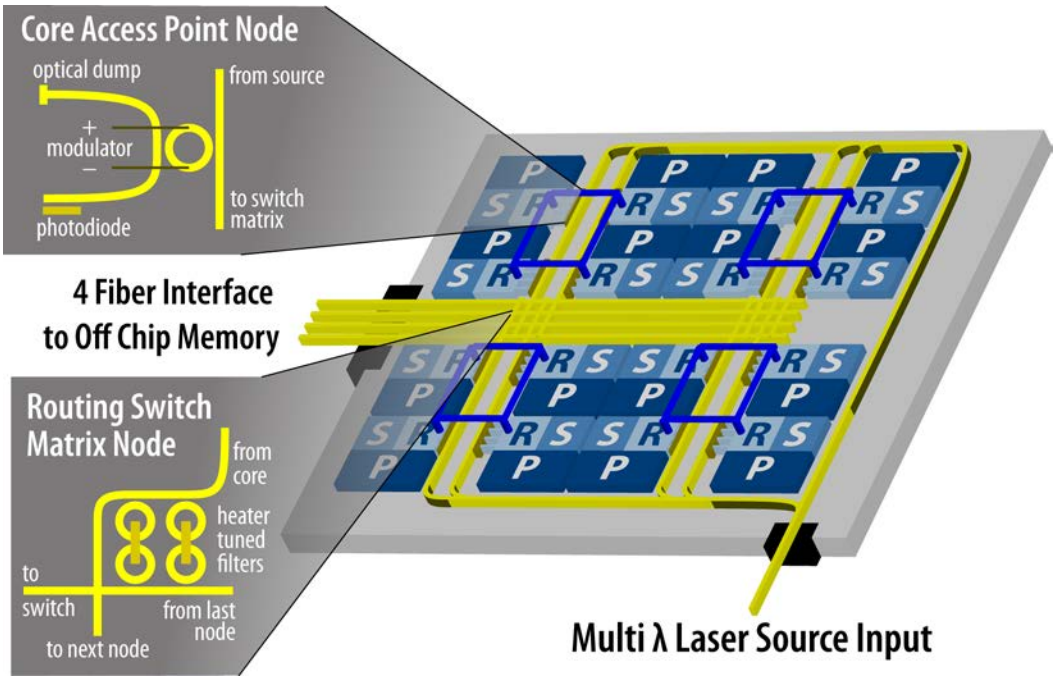


**Figure 2.13** Wavelength routed photonic network to divide access to a global memory space into 4 on-chip groups with independent electrical mesh networks.

Although the wavelength-routed on-chip network enables a much simpler electrical network, it significantly complicates the memory module. The memory module must now act as an arbiter between independent incoming memory requests from different core groups as shown in **Figure 2.14**. In principle, contention may be simply resolved by denying access to one request, e.g. sending a NACK signal, whenever memory access requests conflict. The difficult nature of the problem is maintaining cache coherency for locally-stored memory. Since the electrical networks updating a shared memory operate independently, updates to a piece of shared memory are not globally distributed. If local copies of memory stored in caches are not known to be obsolete, incorrect data can propagate in the system without additional architectural provisioning. Despite the complications, a full system analysis into the tradeoffs of a multi-socket system built using a wavelength-routed photonic network such as the 256 total core system shown in **Figure 2.15** was performed by computer architecture collaborators to a cycle-accurate level and published in the literature [1].

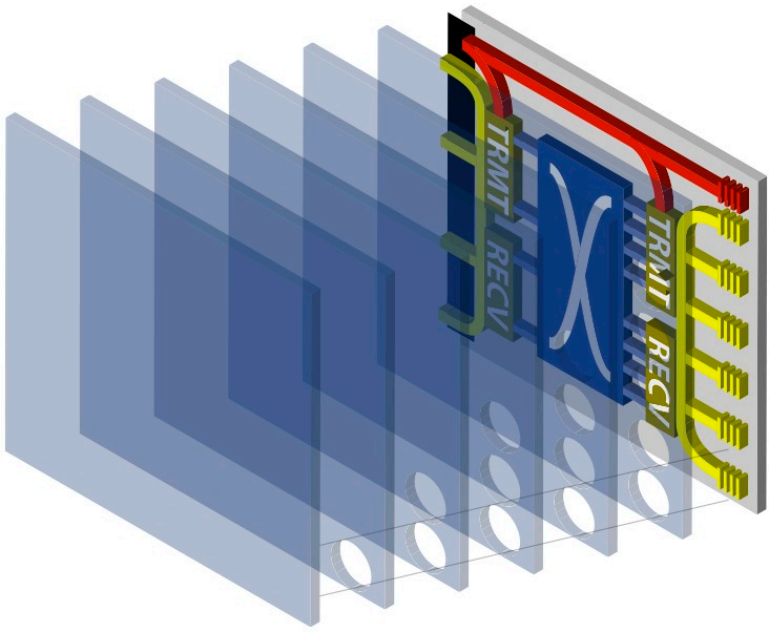


**Figure 2.14** Stacked photonic memory module in which independent access requests must be arbited between and distributed to the relevant DRAM chip.

Additionally, many interesting ideas for future networks arose from this early work. Beyond a host of interesting exchanges that emerged from the intersection of the contrasting ideologies of computer architecture and device engineering, some new component ideas that may impact the way future networks are constructed were generated. One important concept is that rich connectivity between independent systems can be achieved by connecting the ribbon fibers between them orthogonally in passive off-chip connectors. A simple example such connector is shown as the cross-plane connector in **Figure 2.15**. In general, this notional connector, informally labeled as a "star-coupler" has proven especially helpful in connecting the socket-to-socket coherency network of more recent system proposals [214].

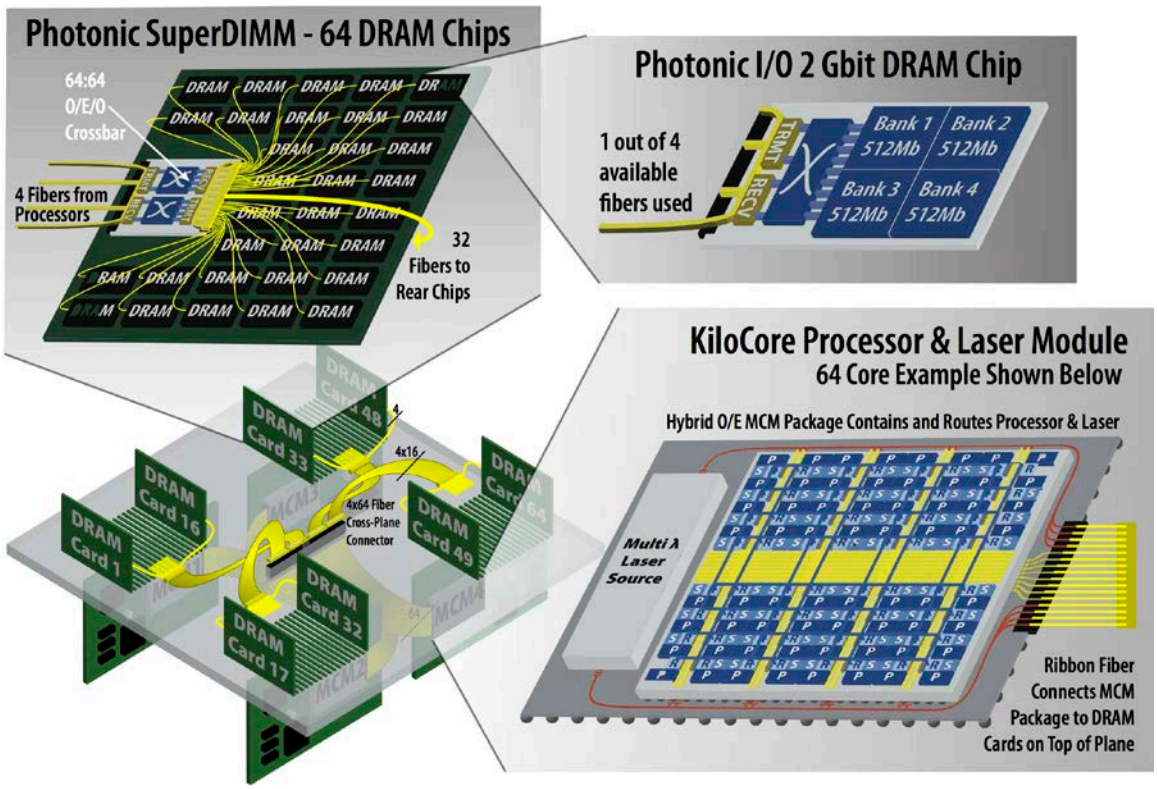


**Figure 2.15** Full system implementation of a multi-socket system using a wavelength routed photonic access platform.

### 2.4.3. Survey of Current and Future Architecture Work

Recent computer architecture work among project collaborators has focused on advanced photonic memory access networks that include explicit socket-to-socket coherency networks [213]. Although this is the project topology that is currently the focus of the newly launched DARPA POEM project, the link level details of this proposal is beyond the scope of this thesis. The coherency connectivity is closely modeled after Intel's quick path interconnect (QPI) proposal [215]. Due to a focus on other aspects of this work, I was no longer directly involved in the development of this architecture. Interested readers can find the details of this proposal in the published literature [213]. For comparison, several other research groups have put forward alternative network proposals [216-219].

In this section, a basic network architecture has been used to motivate the device work. From a simplified analysis of the electronic problem, competing density and energy-efficiency constraints appear to prevent realization of the desired computing systems by at least a factor of four in each constraint. Photonic

interconnect has the capability to enable high density off-chip communication interfaces to memory for scaled microprocessor systems within the desired system energy budget. To realize the opportunity, integrated photonic devices must achieve certain performance metrics. The waveguide loss target of 3 dB/cm enables not only low loss cross-chip routing, but also low wavelength multiplexing filter insertion losses and highly resonant modulators. Wavelength division multiplexing filter banks must be fabricated with high precision to minimize the thermal tuning requirements that comprise almost one third of the total on-chip electrical link energy. The multiple fiber-to-chip interfaces of the proposed optical network encourages significant development to minimize the vertical coupler insertion loss. Integrated photodiode efficiency has a linear proportionality to the total optical power required for the links. The architecture targets assume that 0.5 A/W photodetectors can be fabricated in the logic process with a 0.8 A/W photodetector integrated through the insertion of a new silicon germanium module into the memory target. We will revisit the total system performance after discussing the achieved device performance in Chapters 4 and 5.

# 3. FOUNDRY DESIGN ENVIRONMENT INTEGRATION

In the past, photonic devices have traditionally relied on customized processes with little industry-wide standardization. One result of this is that each company or research group has typically maintained their own accompanying design environment that is specific to their fabrication method. This heterogeneity has been aided by the simplicity of most photonic integrated circuits that rarely exceeded several hundred devices. In this environment, photonic designers have typically laid out devices and circuits in simulation software tool suites made by companies such as RSoft and Photon Design. The processing masks used for fabrication were then constructed in software tools that focused upon geometry alone, such as LEdit or AutoCAD, therefore separating functionality from layout. Recently, the increase in scale of photonic integrated circuits has spurred the development of photonic device layout within computer aided design software packages traditionally used by the microelectronics industry such as Cadence Virtuoso. Although the development prior to this thesis in this area has been done by companies such as Luxtera, no public tool suite for this purpose has been released. Additionally, many challenges exist to design photonic integrated circuits of sufficient complexity to meet the needs of emerging applications.

In this chapter, I will introduce the methodology that has been developed to integrate photonic devices within the standard foundry process and large-scale design environment of the modern CMOS electronics industry. The design environments for existing photonic foundries as well as the standard CMOS foundries will be introduced. Next, the design environment integration platform used for photonic fabrication will be discussed. Finally, a summary of fabricated CMOS electronic-photonic test chips will be presented.

## 3.1. A Comparison to Photonic-Specific Foundries

In parallel to the CMOS foundry integration work that has been the focus of this thesis, several efforts within the silicon photonic community have sought to create a parallel foundry infrastructure to the existing electronics industry. The first effort, EPIXfab, is a collaboration between the IMEC and LETI semiconductor research centers in Europe. Mask-share runs of both passive photonic processes, which

comprise only of undoped, silicon patterning, as well as active photonic flows which add metallization and implant steps have been made available to the broader photonics community. A more recent effort, Opsys out of the University of Washington, that is offering its first early stage shuttle runs seeks to add some electronic devices into processes to offer the opportunity of complete photonic-electronic integration.

The photonics design database preparation, submission and verification architecture significantly deviates from the sophisticated electronic computer aided design (ECAD) environment used in the foundry electronics industry. Although there is less infrastructure development required for such an approach, the scalability of the platform to complicated designs is limited and any electronic integration is made significantly more complicated. The approach taken in this work is to leverage the existing tools and techniques present in the electronics industry to enable photonic layout. Although significant infrastructure needed to be developed, it is now possible for a single designer to lay out many complicated photonic structures in close collaboration with electronic designs that require the capabilities only found in mature ECAD.

## 3.2. CMOS Foundry Design Environment

The realization of photonics in standard foundry-CMOS requires strict compliance with the established process flow. The scaled-CMOS processes of interest employ 40-60 masks to fabricate dense transistor patterns on a 100-200 nm pitch connected by 6-13 interconnect layers. To ensure reliable fabrication, physical verification of device layout is required. This verification process is structured as a set of greater than 40,000 software checked design rules based on the physical constraints of the wafer processes, lithographic projection distortions and mask fabrication. Although the most important enabler of increased integrated circuit performance has been the progress of fabrication technology, improvements in the scope and sophistication of ECAD tools have also been critical to enable the reliable production of increasingly complex integrated circuits [220]. The modern ECAD-based paradigm partitions design complexity into well-defined stages. The geometric objects placed on all design layers are checked against the accepted set of design rules in an automated process known as design rule checking (DRC). This critical step separates the ideal geometry represented on the design layers from the physical manufacturing process. As long as the geometry meets these constraints,

the generation of the complex projection lithography masks and subsequent wafer processes will produce suitable silicon features. As such, the logic designer is not required to understand computational lithography and the mask-manufacturing house doesn't have to consider the functionality of input geometry.

This paradigm must be broken to some level to integrate photonic devices within the existing ECAD flow. Since the later stages of the manufacturing are designed and optimized for fabricating electronic device geometries, the photonic designer acting as a foundry customer must consider the later data preparation and fabrication steps. This has two major implications: first, care must be taken in designing photonic structures to yield acceptable performance; and, second, the photonic structures must not affect process yield for other standard electronic customers running on the same wafers.

### 3.2.1. The Designer's Perspective

Traditionally, everything that the electronic designer needs to know in order to optimally design the integrated circuit is provided by the foundry in the process design kit (PDK). Included in this kit are the rules to physically draw transistors that will on average meet target specifications that are guaranteed to fall within minimum and maximum bounds referred to as "corners." Electronic files to enable automation exist for all stages of this process. The simulation in schematic form uses the compact-models that describe transistor function. The physical layout is designed using either the standard cells that implement higher-level functionality such as Boolean logic or the parameterized cells (p-cells) to design individual transistors. The automatically-checked physical design rules verify that transistors will function as designed. Extraction tools then identify both designed transistors and parasitic elements from the physical layout to be matched with the schematics to verify that the fabricated silicon will match the simulated functionality providing complete design closure.

For the photonic devices, the foundry-provided process design kit does not provide much assistance. The electronic process is not specified for its applicability to photonics such that there is no analog to compact models for simulation, standard or parameterized cells for layout, or functional verification of any kind. Providing the functionality to enable a close approximation to the electronic process flow has been a major component of the work associated with this thesis. Addressing the inputs (process parameters) and outputs (design rule checking through mask

manufacture) to the design process will be covered in the following subsections with later sections addressing the layout and process compliance issues required.

### 3.2.2. Determining Process Parameters

Although the standard design kit does not specifically address photonic devices, many required parameters are explicitly provided. The first required piece of information is the layer thicknesses and materials of all layers surrounding and including the waveguide core. The silicon layer thicknesses for the transistor body and gate (the potential waveguide core layers) are nearly universally provided with all standard PDKs. Additionally the thickness of dielectric layers separating these silicon layers from the substrate and higher lying metal layers are also almost universally provided. What is not typically provided are the details on liner layer thicknesses or even presence or the compositions of most dielectric layers. In this regard, it is extremely important to have a good understanding of the overall CMOS process. Close collaboration with Texas Instruments at the start of this project as well as extensive reading of the available literature including patents has typically provided a satisfactorily complete picture to enable the results achieved in this thesis. Where there is ambiguity in layer thicknesses and composition, indirect information such as the effective inter- and intra-layer dielectric constants for the extraction of wire capacitance may provide additional insight.

The lateral patterning maximum and minimum parameters for the silicon layers must follow the design rules provided for electronics. This removes all ambiguity from minimum line and gap design rules for use in the design of photonic elements. As such the photonic elements can utilize the foundry-provided design rules for manufacturability guidelines of physical geometry shapes for 99% of cases with exceptions discussed in the process compliance section of this chapter.

In the case of the polysilicon layer used for the transistor gate, the refractive index is an additional unknown process parameter. To be rigorously correct, this parameter should be measured in a test run of the process by ellipsometry or by matching the fabricated characteristics of a sensitive device such as a grating with optical simulations using the exact fabrication dimensions determined through destructive analysis. In our initial work with Texas Instruments, a blanket deposition of such a polysilicon film was provided for ellipsometric analysis resulting in the data shown in **Figure 3.1**. Since most CMOS processes utilize similar columnar-grain structure polysilicon that is grown under similar conditions, we

have used this data for all following work with no indication that this assumption is insufficient.

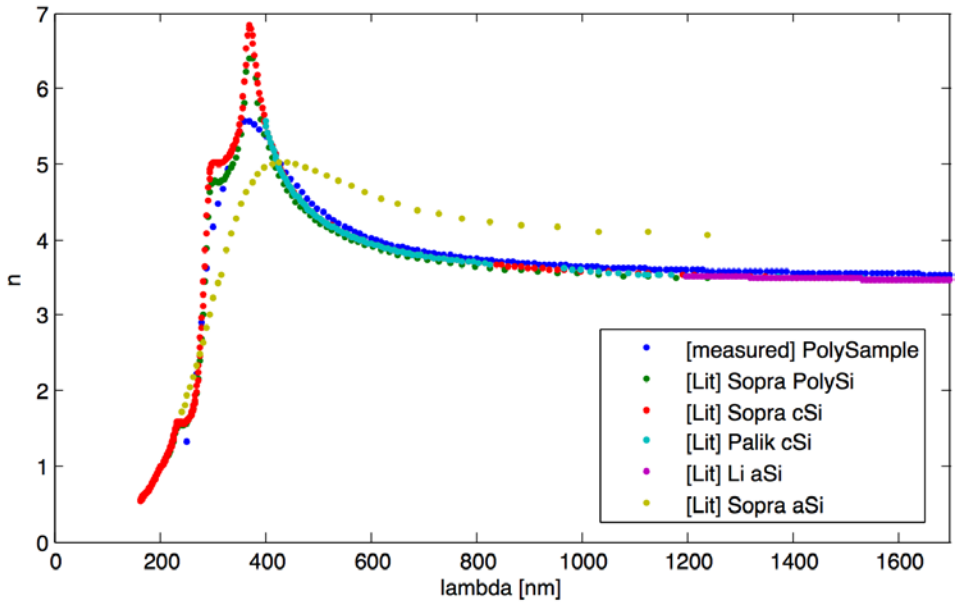


**Figure 3.1** Measured polysilicon ellipsometry data compared to various literature references. Figure prepared by Eugen Zgraggen. Palik data [221],

The doping densities for the implants in the process are required to satisfactorily design depletion-mode modulators. This information is typically only indirectly provided for extremely specialized cases such as the sheet resistance of silicide-blocked precision resistors. In all other cases, the general range of implant concentrations can be inferred from their role in the transistor formation process or explicitly measured. To characterize a process of interest, sheet-resistance test structures have been placed on our fabricated chips. We have used greek-cross based structures that operate under the van der Pauw principle. The sheet resistance of the intersecting arms is measured by forcing a current through adjacent intersecting arms and observing the voltage that develops across the opposing arms as shown in **Figure 3.2**. The sheet resistance is then given by the van der Pauw relation:

$$R_{\square} = \frac{\pi}{\ln 2} \cdot \frac{V}{I} \tag{3.1}$$

The material resistivity can then be calculated by multiplying the sheet resistance by the layer thickness. The calculated resistivity then may provide an estimate of the

doping concentration. This conversion requires the knowledge of the layer thickness and carrier mobility. The layer thickness is nominally known as discussed previously, but may vary within the upper and lower process bounds. Accurate measurements therefore require destructive analysis to determine the fabricated layer thickness. Further, the carrier mobility must be determined by the dopant polarity silicon layer information. Since the functionality within the CMOS process is known, it is typically trivial to determine the dopant polarity. For the single-crystalline silicon layer, the carrier mobility is then well known if mesoscopic scattering effects in the thin layers may be neglected due to the high degree of surface passivation. The polysilicon does provide additional ambiguity in this regard. If an accurate measurement is required, additional information would be required by either benchmarking the electronic test structures with SIMS or otherwise measuring the carrier mobility through Hall effect test structures.

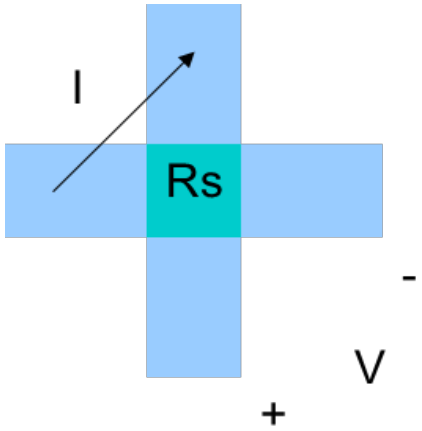


**Figure 3.2** Greek cross sheet resistance test structures indicating the region where the sheet resistance is measured in addition to the sourced current and measured voltage.

The silicon germanium photodetectors require the highest degree of in-situ process characterization for accurate design. Since the silicon germanium is only used for PFET stress engineering and not provided as an independent knob to the electronic designer, little to no information is provided in the standard design kit. The absorption of the material and all other characteristics must the be determined by the creation of test structures. This work will be discussed in the photodetector section.

### 3.2.3. Design Rule Checking

The submitted photonic design will be checked by the foundry against the standard electronic design rule deck. Many design rules that are present to check the proper functionality of the electronic devices such as transistors will incorrectly recognize the photonic device shapes as malformed transistors. Since most all electronic designs will have some violation of the standard design rules due to the complexity of both the designs and rule decks, a robust foundry infrastructure exists to acquire waivers to allow the violation specific rules. All violations require such waivers and this process is discussed further in the design compliance section. The aspect of the design process that has not been addressed to this point is an additional design rule set to check that the photonic devices have been formed as desired. As the maturity of the photonic platform increases, this additional set of design rules must be developed for reliable fabrication. An additional photonic design rule checking deck has been developed for use in the IBM process due to the use of layers that are recognized but not checked in the standard foundry design rule deck. As any errors on these layers would not be caught until the actual mask review checking (MRC) immediately prior to fabrication, additional protection is required by the photonic customer to ensure process compliance. The developed deck has been provided to IBM for use by other customers.

### 3.2.4. Data Preparation and Mask Manufacture

The layers that the designer provides to the foundry in the submitted .gds file do not directly describe the processing masks. There exists a non-trivial relationship between these provided “design” layers and the actual processing masks used in the wafer level processes. First, many processing masks have no direct correspondence to design layers. Instead they may be generated from the combination of other layers. For example, the source / drain and well implant masks are typically only specified by the designer for one transistor polarity (e.g. PFET) while the implants for the other transistor are formed by Boolean operations such that all of the area is nominally doped heavily n-type or p-type. This presents an obvious conflict for photonics. Since the photonic waveguides must be nominally undoped for low-loss propagation, the proper combination of design layers is required to result in all processing masks blocking implants to the photonic regions. Typically, close contact with the Foundry technical support is required to develop the combination of layers

required to block all implants. Some fractions of the Boolean mask generation operations (typically referred to as data preparation tables) are provided to designer in the PDK. This information may be helpful for resolving many questions such as the combinations of layers required to generate the correct doping patterns, but utilization of more unusual combinations of layers must be verified with the foundry to ensure that the mask release code actually generates the desired processing mask combinations.

The second important aspect of data preparation is that even design layers that have a relatively direct correspondence to processing masks such as the etch layer for the silicon layer, may be biased either directly in the data preparation flow or physically on-wafer due to specific lithography and etch processes. Some processes actually obscure the physical dimensions from the designers to prevent the designer from having to modify dimensions for gate length changes. This biasing information is typically not provided to the designer, but may be in some cases in order to calculate line resistances for various shapes. To accurately determine the mapping between design dimensions and fabricated structures, electronic bridge test structures are provided adjacent to the greek cross structures used to determine sheet resistance. The bridge test structures measure the voltage that develops over a specified length of material where the current is forced through independent contacts. If the sheet resistance is known by the greek cross structures, the line width of the fabricated structure may be extracted. This mapping of design to physical dimensions is compensated for in the parameterized cell infrastructure developed for this work. The photonic designer therefore is allowed to always specify physical dimensions by the calibrated design kit.

The final aspect of data preparation generates the actual processing masks. In addition to the described translation of design layers to physical processing layers, many computational steps are performed to precompensate for lithographic distortions and export the resulting large data set for mask manufacture. Due to the extreme complexity of these steps, the format of input geometry is limited to the point that shapes such as arcs explicitly prohibited. The first step is known as optical proximity correction (OPC). This precompensates for the low-pass filtering effect of diffraction in the lithography system. By emphasizing high-frequency shapes, e.g. turning a rectangle into a “dog-bone,” shapes with features below the diffraction limit may be fabricated with high yield. The second step is known as mask fracture in which the high data-volume post-OPC processing masks are

streamed out into a suitable format for the mask-writer. Both steps require strict compliance with the standard input geometry formats typical in the electronics industry.

## 3.3. Design Environment Integration

### 3.3.1. Full-Custom Layout of Photonic Structures

To allow co-design with the electronic circuits and to ensure the generation of design databases compatible with the standard electronics infrastructure, we chose to develop our photonic design platform within the dominant full-custom very large scale integration (VLSI) layout tool, Cadence Design Systems' Virtuoso. Other major CAD manufacturers such as Synopsys and Magma, provide other suitable tools, but we chose to develop this platform in Virtuoso due to its near universal foundry support. The first obstacle in this platform is that the vertices of all objects must conform to a discrete grid. Additionally, all relevant shapes must be made of lines at either 0˚ or 90˚, i.e. Manhattan geometry. This is the result of extensive optimization of the subsequent manufacturing processes for traditional electronic designs. This presents a significant challenge to produce the smooth curves required for many photonic devices as shown in **Figure 3.3**. Additionally, proper design hierarchy and automation must be enabled within this platform to enable complex integration.

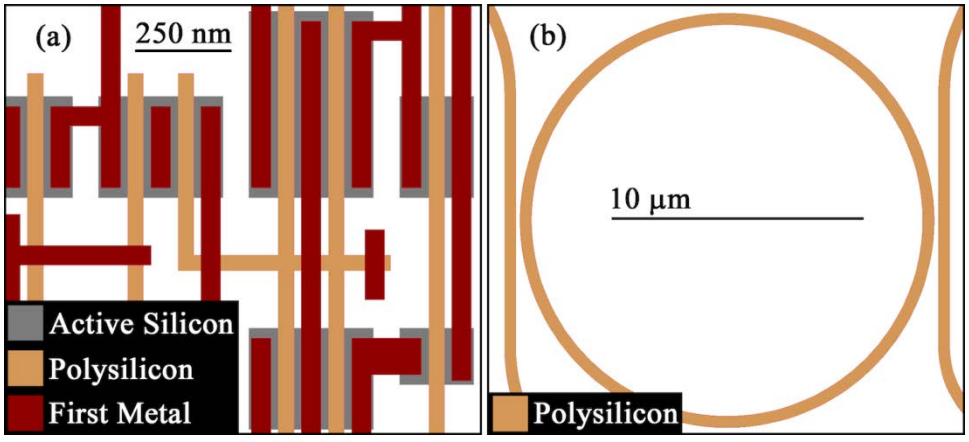


**Figure 3.3** Layout comparison of a scaled bulk-CMOS transistor circuit (a) and a photonic ring resonator (b). The widths of the waveguides have varied with fabrication process details over the range of 300 nm to 600 nm. The gaps between bus and ring waveguides have also varied from 100 nm to 950 nm.

### 3.3.2. Design Layer Overview

Relevant design layers for photonic devices, shown in **Figure 3.4**(a), can be divided as follows: core formation, doping block and fill block. Waveguide core layers include both polysilicon and active silicon in the silicon-on-insulator (SOI) process or just polysilicon in the bulk process. The design layer shape corresponds to features in photoresist on-wafer to block the etching of the high-index semiconductor material. This layer may be biased during data preparation to come out either larger or smaller than designed as part of the foundry's transistor process optimization and therefore needs to be precompensated to achieve the desired width in silicon for photonic applications. Additionally, the core is the only layer that fundamentally requires smooth discretization to prevent optical scattering loss in bends. These layers also correspond to the highest resolution photolithography steps available in the electronics process. This high resolution and accompanying process control is a significant asset to the fabrication of nanophotonic devices.

Without the designer specifying any other layer than just the active (body layer in the SOI process) or poly core shapes, the created silicon layer would be heavily doped n-type above a $1 \times 10^{20}$ $cm^{-3}$ density and covered by a ~20 nm metal layer known as a salicide. The automatic nature of these generated doping and metallization processes ensure that it is harder for the electronic designer to make a mistake such as creating an unintended high-resistance region in his or her transistor. Therefore, creating a suitable undoped, unsilicided silicon region for photonic integration requires the addition of several design layers, which we further refer to as the "doping blocks," to completely cover the waveguide core. Since these layers only serve to modify the local processing rather than to guide the light, they may be coarsely placed around the waveguide core as shown in **Figure 3.4**. The doping block layers may actually result in dielectric layers that are present at the end-of-line chips. As such, the doping block layers should either be placed with sufficient overlap to eliminate scattering loss from their boundaries or finely discretized where short overlaps are necessary.

The actual design layers that compose the doping block stackup are unique to each foundry process. Although the doping block solution should always be verified by foundry engineers, there are some general rules that the designer can follow to identify suitable design layers as an initial suggestion. Considering the silicidation process, it is first necessary to identify which electronic devices require blocking the

salicide to function normally. Although it is possible the process does not include this functionality, all CMOS processes that we have studied have included a silicide blocking layer for either precision polysilicon resistors or extended drain transistors. The design layer for precision polysilicon resistors may combine the silicide block with a special implant to minimize the thermal variation of the resistivity. Instead, experience has shown that the best design layer is the silicide blocking layer to form extended-drain devices for electrostatic discharge protection and I/O where this functionality is available. The required information to understand the relation between the design layers and processing masks is typically provided in a table that lists the various Boolean logic operations performed during data preparation.

The ability to block all implants from a given region may not be required for any electronic devices in the process. This is not a common goal in the creation of CMOS electronics. Typically, the designer only specifies the implant layer for one transistor polarity with the other generated by overlapping Boolean operations. For example, the most common configuration is that n+ source/drain implants into a p-well is the default background. The designer then is only responsible for drawing the p+ source/drain implants and the n-well regions. There is generally not a standard design kit layer to block all implants from a specific region. By experience, some processes that are commonly used for advanced analog design include a specific "high resistivity" design layer to accomplish this task. In all other cases, layers that are not explained in the standard design kit or explained in the Boolean layer generation table are required. The foundry may then provide a set of input gds design layers to accomplish this task that the designer is responsible for adding to the standard design kit. In general, an internal solution to create a suitable undoped region will exist internal to the foundry for process engineering.

The final set of surrounding design layers required to create a suitable local photonic environment block the foundry's automatic placement of shapes near the integrated waveguide core. These so-called fill block layers must be included for the active silicon layer, the polysilicon gate layer and low-lying metal levels. The foundry typically inserts these shapes to equalize pattern density for wafer-level processes. Although the addition of these shapes has little to no impact on most electronic devices since they are guaranteed not to touch any customer-placed shape, fill exclude layers are typically included in the standard design kit to lower losses in microwave components such as inductors. While the silicon layers would

introduce scattering sources, any metal levels present near the waveguide core would cause very large material absorption losses. Although the exclusion of these layers may be allowed in the standard design kit, strict design rules apply to ensure that local density is maintained on the length scales required for the wafer-level processes. Further details on the fill-blocking are provided in Section 3.4.1.

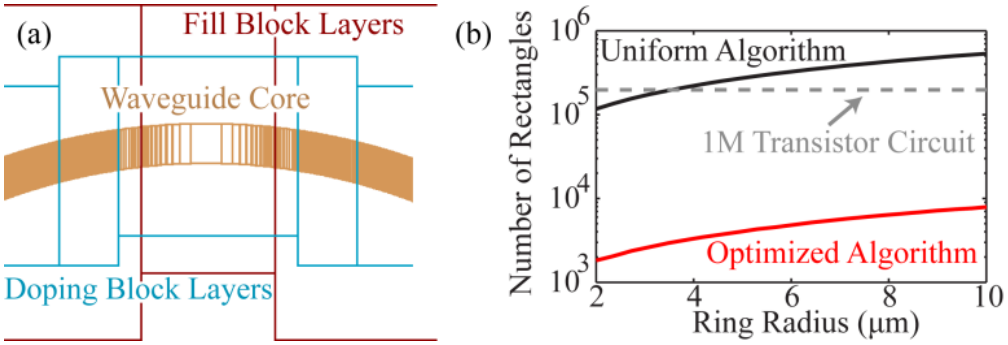


**Figure 3.4** Optimized discretization algorithm on design layers utilizing different grid resolutions (a); and the necessary quantity of rectangles to represent a 0.5μm wide ring with uniform and optimized algorithms (b). The rectangle quantity of a 1 million transistor circuit is also shown for reference. Optimized algorithm grids: 1nm (core), 0.2μm (doping block) and 0.8μm (fill block).

### 3.3.3. Efficient Photonic Device Layout on Design Layers

Instead of relying on manufacturing processes optimized for curves, the resolution of scaled-CMOS lithography can be used to discretize the required shapes on the 1-5nm design grid. Due to the large size of the photonic structures of interest compared to the design grid, an efficient representation of the curves is necessary. Options for building block objects are limited to many-vertex polygons and rectangles. Polygons have the benefit of minimizing the total number of objects in the design database. However, since integration into the existing data preparation and mask manufacturing flow is required, we have chosen to use rectangles as the basic building block to more closely resemble standard electronic geometry databases. Since mask manufacturing is not done in-house by CMOS foundries, the risk of complex polygons generating patterning errors cannot be easily assessed by foundry engineers available to a mask share customer.

Uniform rectangular slicing of a 500nm width, 10μm radius ring on a 1nm grid results in ~40,000 rectangles per layer. Since 14 layers are required to define the core and surrounding material stack up, the rectangle quantity exceeds 500,000.

This would require more than double the number of rectangles, 197,600, present in a >1 million transistor electronic design recently fabricated in a 32 nm process. The compression of rectangle volume, due to the design hierarchy discussed in Section 3.2.3, is required to reduce the computation required for foundry mask pattern generation to a practical length of time [222]. Therefore, a design submitted with many photonic devices represented in such a simple form will not complete the pattern generation flow in an acceptable time.

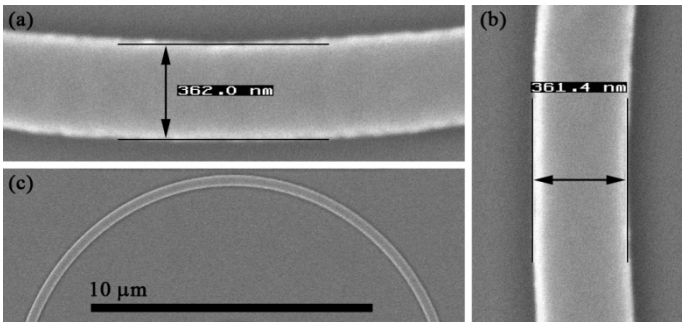


**Figure 3.5** Scanning electron micrograph (SEM) of a polysilicon ring waveguide core fabricated within a 32 nm bulk-CMOS process showing east-west linewidth (a), north-south linewidth (b) and no visible angular distortion (c).

To minimize data volumes, an efficient rectangle discretization algorithm is then required. The discretization grids are chosen to be as large as possible for each design layer. Waveguide cores such as the polysilicon gate layer and single-crystalline body layer in SOI processes are discretized on the minimum available design grid of 1 or 5 nm, depending on process generation. The resulting fabricated curves do not show any signs of the underlying mask discretization or any other optically observable distortions as shown in **Figure 3.5**. The other design layers used to ensure the correct material surroundings such as the doping and fill blocking layers do not require smooth edges and are discretized coarsely far from the waveguide core. An example of the resulting structure is shown in **Figure 3.4**(a) with the resulting data volume as a function of ring radius plotted in **Figure 3.4**(b). The total data volume and related information is provided for each of the test chips designed over the course of the thesis in Section 3.5.

### 3.3.4. Hierarchical Parameterized Cell Design

In VLSI layout, a careful design hierarchy reduces the data volume of complex integrated circuits. The full chip is comprised of many building block cells replicated

and positioned in higher level cells. Importantly, cell layouts can be generated by scripts based on designer input variables. These parameterized cells, or p-cells, typically describe electronic components such as transistors and resistors. For Cadence Virtuoso, the parameterized cells may be created as a fixed geometry with a variable parameter through the graphical user interface or as a completely script generated object using the SKILL programming language. Due to the complexity of the generated shapes, SKILL-based parameterized cells have been exclusively used for this work. One consequence is that the p-cell infrastructure is only compatible with Cadence Virtuoso. Recently, open p-cell standard known as PyCells have been developed by Ciranova that promise to be compatible with any layout tool based upon the industry-standard OpenAccess database format. Porting the SKILL scripts to the required Python format is not a major technical challenge. An initial attempt to migrate to the PyCell platform, however, was aborted when it was discovered that certain caching mechanisms that greatly speed layout were not supported for the PyCells within Virtuoso at the time.

The photonic layout p-cells were then developed for the multiple foundry technologies explored in this thesis. An extensive hierarchy of p-cells implements all required functions from basic geometry to complete photonic device blocks. These p-cells are then available for placement in the chip layout view by the designer in a process known as structured-custom design. By placing the p-cell, the photonic designer has full control over the desired structure by editable parameters such as device width, radius and port spacing, while not having to manually place the many required layer geometries. At the lowest level, non-Manhattan shapes such as circular arcs, tapers and sinusoidal offsets are discretized based on geometric input parameters for an arbitrary design layer. Next, basic photonic object cells, e.g. rings and waveguides, place these geometries on the correct layers for finely discretized waveguide cores and coarsely discretized surrounding stackup layers. By isolating the highest data volume geometric objects at the lowest level in the hierarchy, basic building blocks are referenced across the design instead of replicated. This is applied even within an object with symmetry axes such as a ring where a single quarter arc is referenced in four locations. Since Virtuoso creates only a single instance of a p-cell for a given set of input parameters, data volume is greatly reduced to allow manageable compute times.

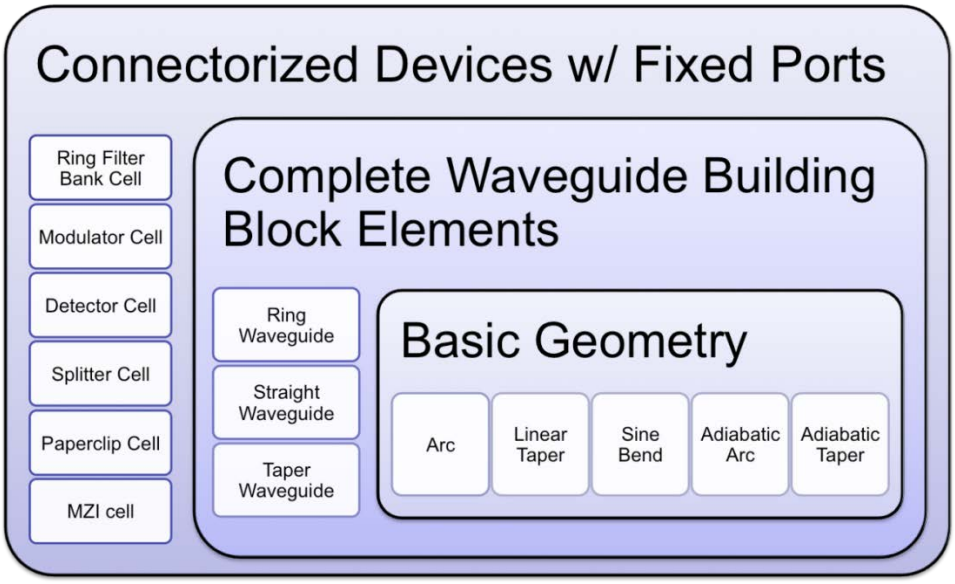


**Figure 3.6** Hierarchy of implemented p-cells for data volume management. Only a subset of the total p-cell library is pictured.

At the final level of the design hierarchy, the basic object p-cells are placed into standardized layouts with waveguides connecting desired devices to fixed input and output port locations. The exact sizes of internal waveguides and bends therefore resize automatically when a device geometry parameter such as the spacing between a ring resonator and a bus waveguide changes. At this level, the designer can easily place the photonic components into complex networks and test structures through the standard Virtuoso user interface. Real-time user input device parameters and port sizing then allow automatic generation the physical device. The p-cell design hierarchy with example cells for each level is shown in **Figure 3.6**.

The parameterized cells are configured to create photonic device geometries by either cell-specific user-input parameters or globally defined variables. An example user-environment screenshot is shown in **Figure 3.7**. For a given wavelength and core thickness, the singlemode waveguide width for any section of the photonic device may determined by the globally set parameter. Alternatively, if a given device must deviate from this value, the user may switch a parameter to a fixed value to display an input field for the specific dimension. More generally, the flexibility to show or hide input parameters enables a single p-cell to implement a wide variety of devices. For example, a generic resonator cell may be switched from a single or double ring filter to a modulator by a "resonator type" setting that displays the appropriate configuration settings to the user. This enables the centralization of

general purpose input and output waveguide generation code into a single SKILL script to simplify code organization.

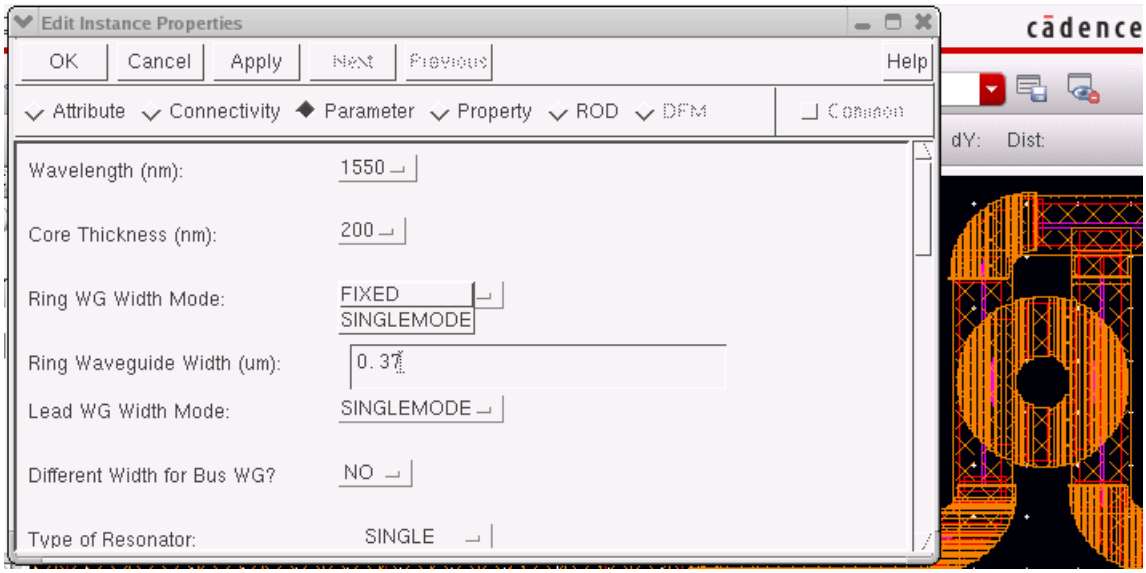


**Figure 3.7** Adaptive p-cell user interface. Design parameters such as singlemode waveguide sizing for a given wavelength can use default global variables or be altered to fixed user-specified values. Input cells for variable parameters are hidden if not applicable to the current device set.

## 3.4. Design Rule Compliance

For the foundry to accept the design for fabrication, successful final automatic verification of design rule compliance is required. Since the typical separation of circuit design from process knowledge has been broken by the inclusion of novel devices, the photonic designer must work closely with the foundry representatives to ensure that rules that misinterpret photonic devices as malformed transistors are waived. More importantly, the photonic designer must ensure that no aspect of the submitted design violates rules, such as minimum geometry feature rules and areal density rules that would affect the process yield of other customers on the wafers. In our experience, the state-of-the-art lithography used in scaled-CMOS technology allowed all geometric size rules for waveguide core layers to be met with minimal modifications to desired photonic structures. For surrounding material stackup layers, the coarse discretization, larger than minimum spacing and notch rules, results in zero violations within a basic object p-cell. Locking these grids to a global fixed grid regardless of p-cell location or orientation then eliminates spacing violations generated between p-cells.

### 3.4.1. Exclusion of Lossy Materials from Local Environment

Multilayer copper interconnect processes require strict adherence to rules for the areal density of metal. Foundry-generated autofill typically hides the complexity of maintaining consistent pattern density from the digital designer. This process is broken by the fill block layers and therefore requires the designer to take care to maintain compliance. Since microwave designers often attempt to modify local metal density to reduce parasitic losses in their components, metal pattern density violations are the most frequent cause of design rejection by CMOS foundries.

#### *3.4.1.1. Photonic Requirements*

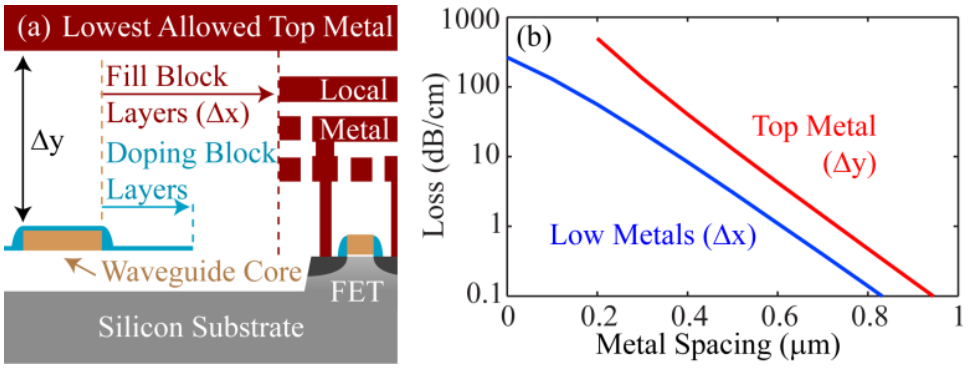


**Figure 3.8** Cross section showing the relevant design layers (a), and the impact metal spacing on waveguide loss (b). Simulation performed at $\lambda$=1550nm for a 600x100nm core. Simulation details: 300nm cladding under core; $n_{substrate}$=1, $n_{core}$=3.55, $n_{clad}$=1.46, $n_{cu}$=0.76, $k_{cu}$=10.4.

The extent of the metal exclusion region is governed by optical loss. The required isolation around the waveguide core is shown in **Figure 3.8**(a). As shown in **Figure 3.8** (b), the minimum size of the metal exclusion region is typically 1 μm both laterally and vertically. The thin vertical exclusion requires only that the first few metal layers of the process are subject to the fill-block exclusion regions. The global metals (typically >1.5 μm above the silicon core layers) may then be routed irrespective of the photonics. This freedom greatly simplifies the high level design and allows optimal power and clock distribution. Since transistor circuits typically require the first two to three metal layers for local routing, very few global signals would ever attempt to route within the excluded metal layers. Blocking higher metal levels is only required for the vertical coupler input and output ports. These 10 μm by 10 μm regions must then be added as local obstructions to the global place-and-route floorplan. The small size of these sparse regions pose little threat to process

integration. Of greater concern is that the required lateral exclusion on low metals still interrupts uniform pattern density on those layers. The typical width of these exclusion regions then is only required to be ~3 μm. Due to the uneven discretization of the fill block layers, the total exclusion widths have typically remained in the 4 μm to 6 μm range for the fabricated designs.

#### *3.4.1.2. Process Compliance*

The length scales (25-150 μm depending on process) over which the density must be uniform for the chemical mechanical polishing and patterning steps are much longer than these ~5 μm exclusion regions. As a result, the design rules only check the total pattern density within "check windows" of 20-30 μm per side at minimum. The minimum and maximum densities required within the check windows are typically close to 20% and 80% respectively. Therefore, high metal density cells can be placed around the photonic regions to meet design rule targets for minimum pattern densities within these check windows. It is important to note that stricter pattern density targets exist for progressively larger pattern densities to ensure process uniformity. These further check windows range from 100 μm tiles to full chip density targets. At the highest level, density targets can be typically as strict as ±5%. These targets are typically easily met if the photonics shares significant chip real estate with intermixed electronic circuits or space that is not explicitly filled by the designer.

#### *3.4.1.3. Automatic Fill Generation*

To ensure that the local density variations introduced by the photonic cells do not affect the end design rule compliance of the layout, high-density fill cells are required around the photonic shapes. In the first (TI 65 nm) tape-out, this was done manually, by the creation of a library of high fill density "picture frames" for each type of photonic device. In the second (TI 32 nm) tape-out, the design rule checking and foundry fill generation were sophisticated enough such that all density design rules were met by adequate spacing between photonic device fill-exclusion regions. However, to be robust against simple design rule checking and foundry fill generation routines, an automatic fill cell placement routine has been developed. This process was first developed for the September 2009 IBM12SOI tape-out. To accomplish this, a Mentor Graphics Calibre input file has been designed in collaboration with Benjamin Moss and Fred Chen that recognizes the exclude fill

shapes in the layout. Next, a set of geometry operations grow this region, subtracting out sections that overlap with other existing geometries. The fill check window stepping infrastructure within Calibre is then leveraged to output a text file of lower left corner coordinates of 0.8 x 0.8 um squares requiring autofill.

A set of SKILL routines available in the Virtuoso user environment through the custom "Photonics" menu places the high-density fill cells at the Calibre generated coordinates as shown in **Figure 3.9**. The instantiated fill cells are identified by the "autofill_" prefix to the instance name for easy removal from the layout. Similar data volume concerns for the curve discretization algorithm are applicable to the autofill problem. If ~25% of a chip area is subject to autofill on a 3mm x 3mm photonic test chip, up to 3.5 million fill cells may be required. Without conscious attention to this data volume, the design rule checking job for the first version of an autofilled test chip had to be terminated after over 500 CPU hours using 128 GB of memory when the job stalled on a single rule check for two human days. Several design techniques needed to be employed to reduce the computational load. After optimization, more complicated autofilled photonic test chips complete full design rule checking in 2.3 hours real time using 8 CPUs.

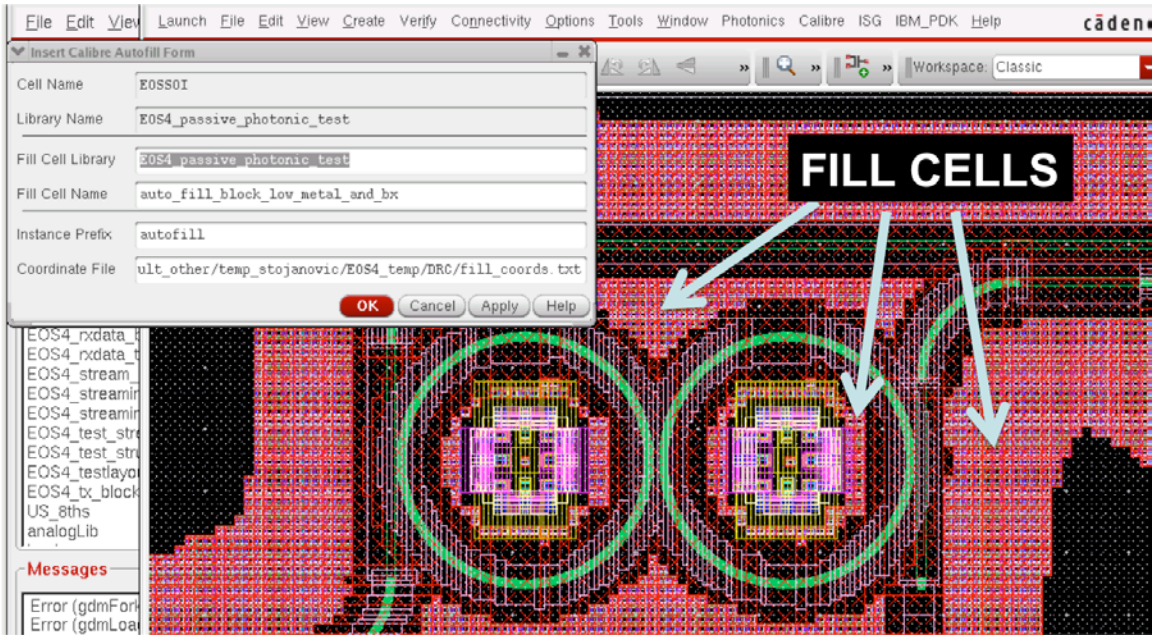


**Figure 3.9** User environment screen shot of the automatically placed fill cells with the insertion dialog box.

The fundamental difference between this type of designer fill placement as compared to the automatic foundry fill shape insertion is that the foundry fill instantiation is performed after complete design rule checking on the submitted design. The foundry's fill shape instantiation is therefore excluded from normal design rule checking due to the extensive verification that the algorithm cannot introduce new violations. Since it is not possible to bypass design rule checking on a customer-submitted design, the only option is to minimize the computational burden. The most important step is to engineer the replicated fill cell to connect with adjacently placed cells. Since there are maximum shape width rules, a uniform pattern is not allowed. Instead, it is generally optimal to have the fill cell shapes connect to form arrays of long parallel wires. Further optimization can be achieved by analyzing the internal workings of the checked design rule checker for the rules that are taking the longest time to compute. For example, most rule checkers such as Calibre find violations by performing Boolean logic operations on grown or shrunk versions of the design layers. Choosing specific layer sizes and spacings may cause the check regions from many arrayed shapes to merge into a single quickly-computed polygon. This engineering for the IBM 12soi process resulted in an 80% fill density 0.8 μm x 0.8 μm tile as determined by the transistor body and polysilicon layers shapes that had the largest impact on the DRC runtime.

Other data reduction techniques had a smaller total impact on the computational runtime. One technique is to merge the individual autofill cell instances into array placements. This is performed as the fill cell coordinate text file is parsed by the SKILL placement script. The script is designed to keep up to 100 array locations in memory in an attempt to grow existing array placements before of individually placing a cell. The final option is to use design layers that are explicitly designated for fill shapes. Where available, these design layers may be subject to fewer design rule checks. More importantly, these shapes are generally excluded from the optical proximity correction algorithms to speed the processing time for the in-foundry data processing flow.

### 3.4.2. Minimum/Maximum Geometry Design Rules

The most obvious design rules that need to be followed for the layout of the photonic devices are the geometric layout restrictions such as the minimum gap between two features. In general, the high-resolution lithography present in these processes enables most nanophotonic devices to be fabricated without requiring the

designer to modify dimensions for design rule compliance. Instead, minimum feature rules typically result from spurious notches or tiling errors between cells.

One violation that stems from the discretization of the curves occurs when a tight bend is abutted to either a mirrored bend or several other shapes such as abrupt taper. Without modification, the last nanometer-scale notches could present either a minimum spacing or width violation when combined with the other shape. As shown in **Figure 3.10**, these violations can be prevented by pulling back "concave" discretization shapes and removing "convex" discretization shapes. The pictured example is an extreme case of a 1 μm radius bend with a 0.25 μm minimum spacing/width rule. Even in this case, the resulting geometry distortion is less than four 1nm grid points. For most structures including tight bends, the distortion resultant from design rule compliance is less than 1nm.

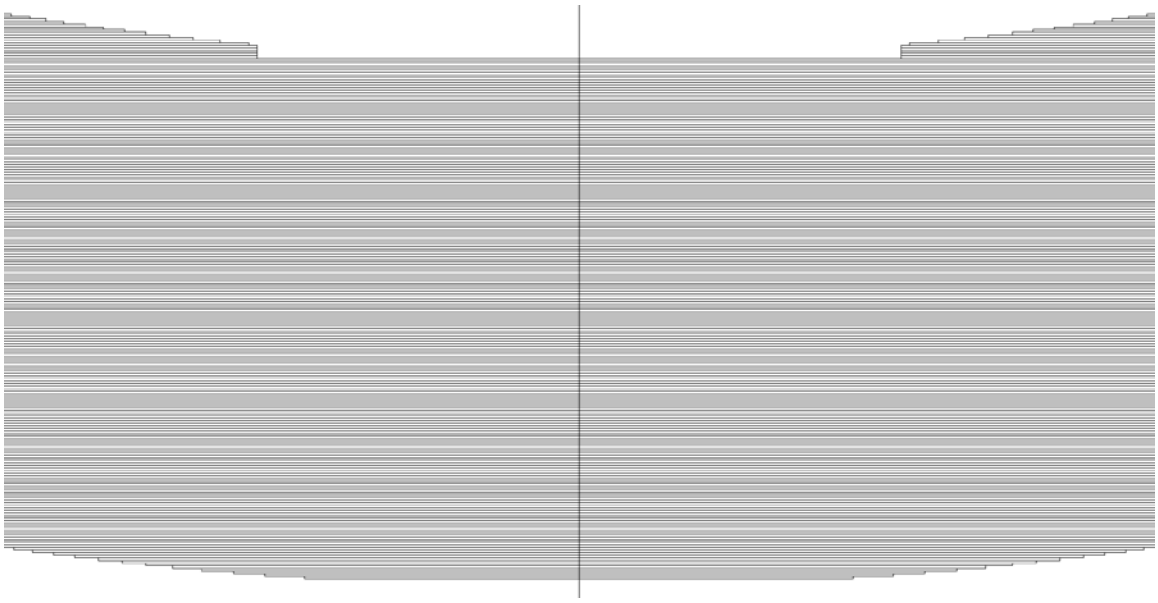

**Figure 3.10** Discretization gap and short-edge removal.

The next class of important violations to consider stem from the overlapping doping block layers between two neighboring photonic p-cells. Since the doping block layers need to extend beyond the waveguide cores, any junction or coupling region requires the transition to overlapping doping block layers. If the doping block layers were finely discretized, the junction would create narrow slivers on the doping block layers that would cause a variety of minimum width and spacing violations on the implant and silicide-blocking layers. The coarse discretization of

the doping block layers on a grid size equal to or greater than the minimum width and spacing design rules may eliminate this problem. The further complication is that the discretization grid in each parameterized cell must then be locked to a global grid to prevent similar slivers as shown in **Figure 3.11**. This can be accomplished if the placement and rotation information is passed down the p-cell hierarchy. From the top-level user layout, the placement and rotation information for each p-cell instance can be automatically set using the custom "Update P-Cell Origins…" command from the "Photonics" menu. This command sets the rotation and only the offset of the cell placement by taking the modulo of the placed coordinates and the relevant discretization grids. Each sub-p-cell instantiation command within SKILL p-cells must then call functions to calculate the correct offset information based on the instance placement within the cell as well as the offset information of the higher-level cell. This strategy allows for discretization calls within an arbitrary hierarchy depth to be locked to a global grid for the elimination of cell tiling errors.

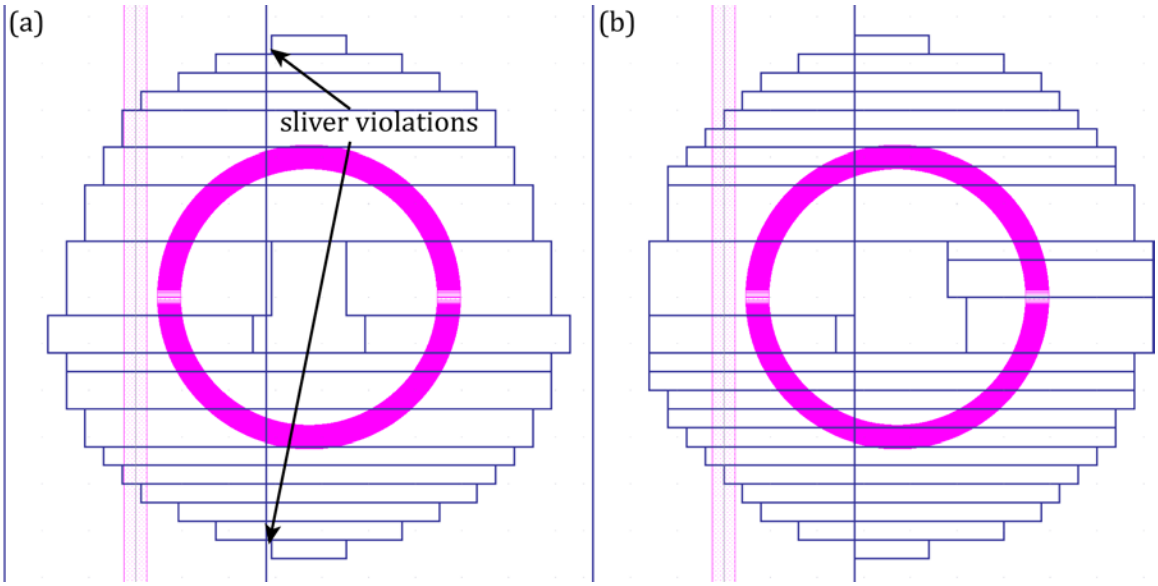


**Figure 3.11** (a) Slivers may be formed between adjacent shapes if the discretization is not locked to a global grid as in (b).

The final category of potential geometry design rule violations that may be flagged for the integrated photonic devices are short-edge or notch violations. Some CMOS processes contain explicit design rules forbidding notches that are defined as a line between two vertices of less than a specified length such as 50nm. The

presences of the notches are typically indicative of designer mistakes abutting wires in traditional electronics processes. Since the notch would not represent a fabricable feature, the presence of a design rule to catch this type of violation improves the fabrication predictability electronic layouts. For photonics, however, the notches represent a necessary discretization to generate smooth curves. It is not desirable to comply with this design rule. As such, we have required a waiver for the violation to be granted by the foundry to allow these features in processes that contain notch design rules on the polysilicon or active silicon layers. An existing foundry infrastructure exists for such waiver requests as many standard electronic designs require similar design rule waivers for specific features. Since the violation of the notch rule has not been seen to present a threat to process yield (as verified by numerous successful fabrication runs), the notch rule waiver has always been granted.

### 3.4.3. Device Functionality Design Rules

When the full electronic design rule checking deck encounters the active photonic devices such as modulators and detectors, the design layer geometries are often misinterpreted as improperly formed transistors. The flagged rule violations are only intended to alert the electronic designer from making circuit functionality errors. For these rules, formal foundry waivers are not required. The subset of design rules that may potentially cause foundry yield problems are typically listed in a "waiver-required" list that can be obtained from the design aggregator. It is not necessary to alter the photonic design to comply with rules that are not on this list.

## 3.5. Summary of Photonic Integration CMOS Test Chips

The photonic integration strategy outlined in this chapter has been used for 10 tapeouts with three major semiconductor manufacturers. The memory integration test chips that were fabricated by Micron are not discussed in this section.

### 3.5.1. AURORA (EOS1) – Texas Instruments 65nm bulk-CMOS

This was our first photonic integration attempt that leaned heavily upon the support of engineers at Texas Instruments to educate us on deeply-scaled bulk-CMOS processes and the best way to submit our photonic designs to be most compatible with in-foundry data processing requirements. The design rule checking

was performed using Chameleon, which is an internal Texas Instruments tool. This test chip will not be discussed in detail in this work, but provided the background knowledge required to develop the integration platform presented in this work. Waveguide loss of 55 dB/cm was measured at 1260nm and served as a proof of concept for the zero-change integration approach [223].

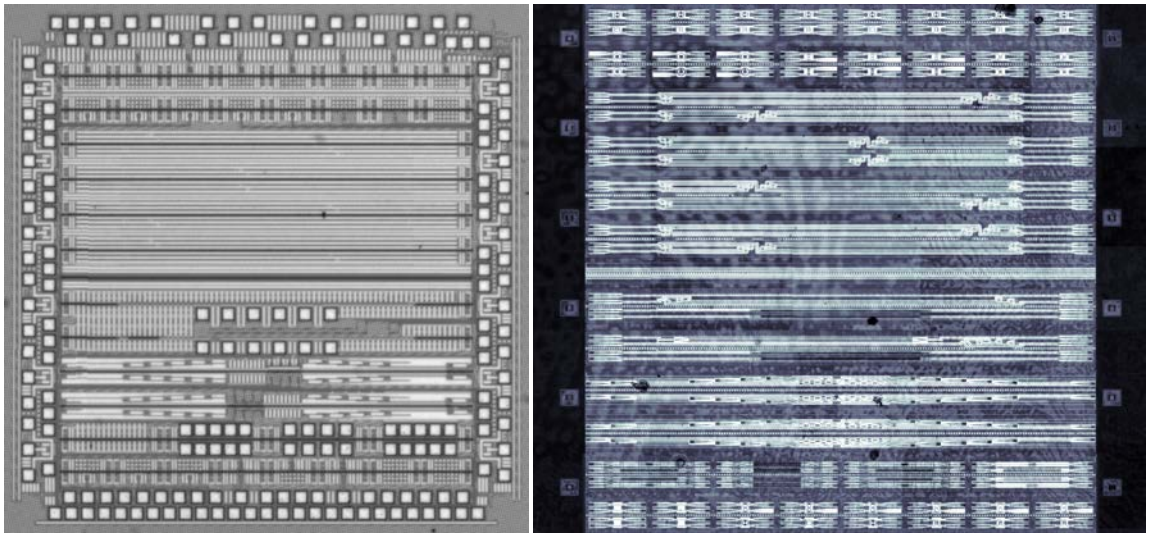

**Figure 3.12** Die photos showing front and back side of EOS1

### 3.5.2. EOS2 – Texas Instruments 28nm bulk-CMOS

The second photonic integration attempt with Texas Instruments was in the 28nm process that was under active development. This was the first platform that included silicon germanium stress engineering for photodetector integration. However, the immature state of process development never resulted in wafers with photonic devices and both polarities of transistors simultaneously functioning. Synopsys Hercules design rule checking of ~50,000 rules took approximately 8 hours running on 6 CPUs. In addition to similar performance waveguides implemented at 1550nm for the first time as compared to EOS1, high lithographic precision enabled the fabrication of second-order ring resonator filters with an average linewidth mismatch of approximately 5Å [70].

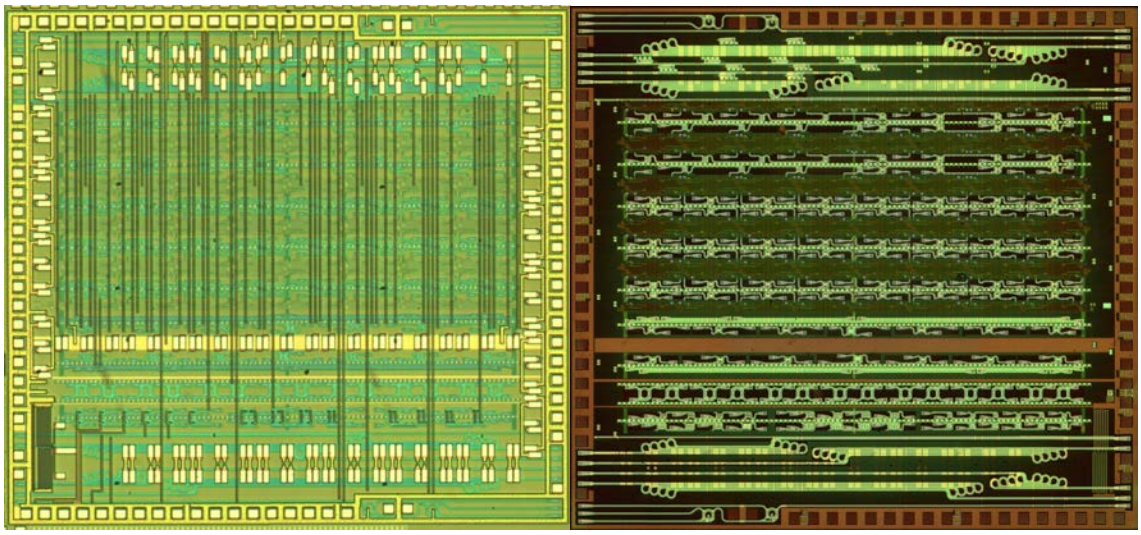

**Figure 3.13** Die photos showing front and back side of EOS2

#### *3.5.2.1. Data Volume*

Streamout took 1 hour to generate a 966 MB .gds file

- Instances: 1,498
- Arrays: 571
- Paths: 10,815
- Rectangles: 15,840,633

### 3.5.3. EOS3 – IBM 90nm bulk-CMOS (9sf)

The first photonic integration attempt with IBM focused on an older technology to determine design rule compliance and to develop the layer combinations required to block all doping and silicidation steps from the photonic region. This only attempt at edge coupling by either cleaving, polishing or facet etching did not result in accurate loss characterization due to coupling variability. It did verify a integration with IBM's foundry process by successfully blocking all doping and silicide processes in the waveguide region while complying with design rules.

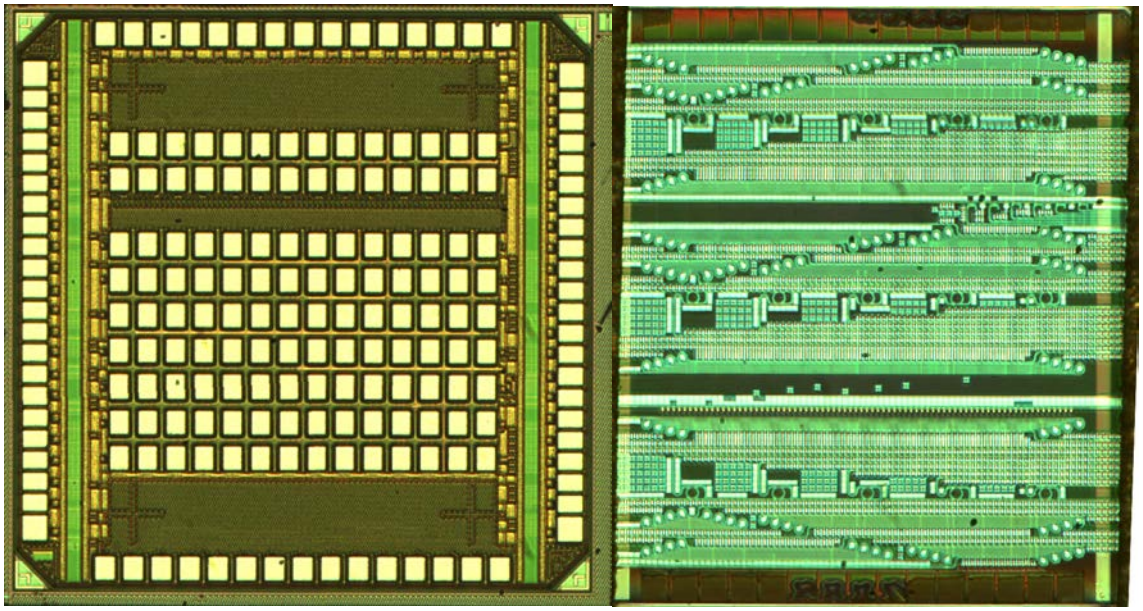

**Figure 3.14** Die photos showing front and back side of EOS3

#### *3.5.3.1. Data Volume*

Streamout took 2.3 hours to generate a 124 MB .gds file

- Instances: 1,491
- Arrays: 228
- Paths: 45
- Rectangles: 2,018,576

#### *3.5.3.2. Design Rule Summary*

6 minutes real time 1338 design rules 5643 operations

### 3.5.4. EOS4 – IBM 45nm SOI-CMOS (12soi)

The first silicon-on-insulator (SOI) test chip. The photonic integration strategy developed for the 9sf process was supposed to be compatible with all future IBM processes. This chip however, resulted in highly-doped waveguides since the layers that were intended to block the doping were not part of the standard design kit and were removed during mask aggregation.

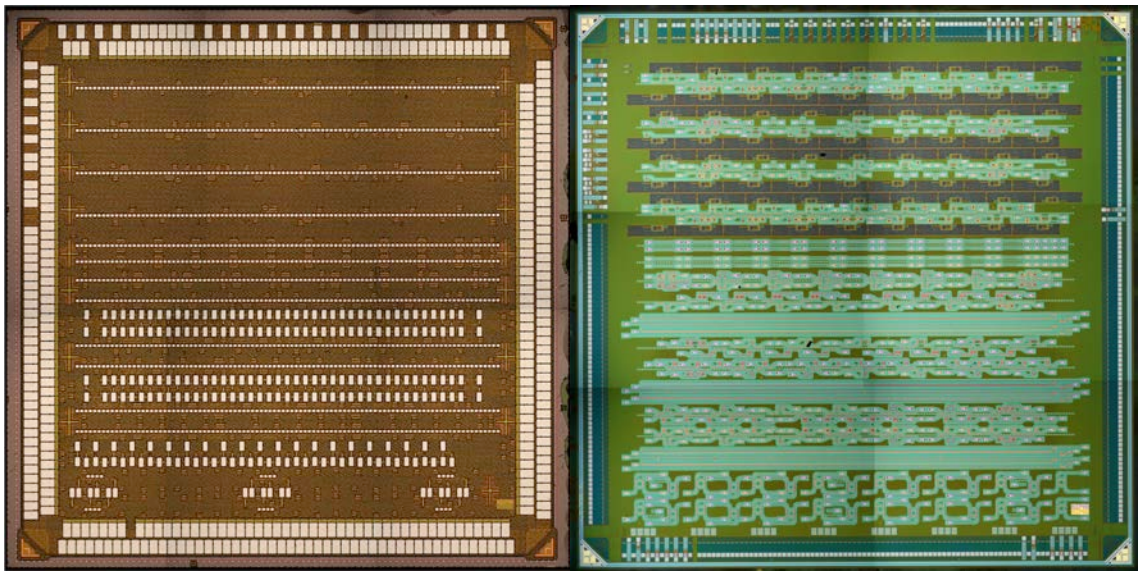

**Figure 3.15** Die photos showing front and back side of EOS4

#### *3.5.4.1. Data Volume*

Streamout took 2.3 hours to generate a 227 MB .gds file

- Instances: 78,905
- Arrays: 69,538
- Paths: 46,553
- Rectangles: 2,143,184

#### *3.5.4.2. Design Rule Summary*

CPU time 23 hours, 5.8 hours real time 5806 design rules 26,120 operations

### 3.5.5. EOS6, EOS8 and EOS10 – IBM 45nm SOI-CMOS (12soi)

The revised versions of the same design that resulted in our first successful SOI test chip (EOS8). EOS6 failed due to the release code for the 12soi process not supporting the internal IBM doping block solution that was developed for EOS3 once the design submission issues associated with EOS4 were resolved. A new doping block methodology was created to successfully block all doping steps into the photonic regions resulting in low integrated waveguide loss for both EOS8 and EOS10. EOS8 was largely repeated on EOS10 as a backup plan in case further processing errors were discovered early enough on in EOS8 to allow the revision of EOS10. Low waveguide loss, electronic-photonic integration, high-extinction

modulators and high-speed photodiodes were all demonstrated on EOS8 and EOS10.

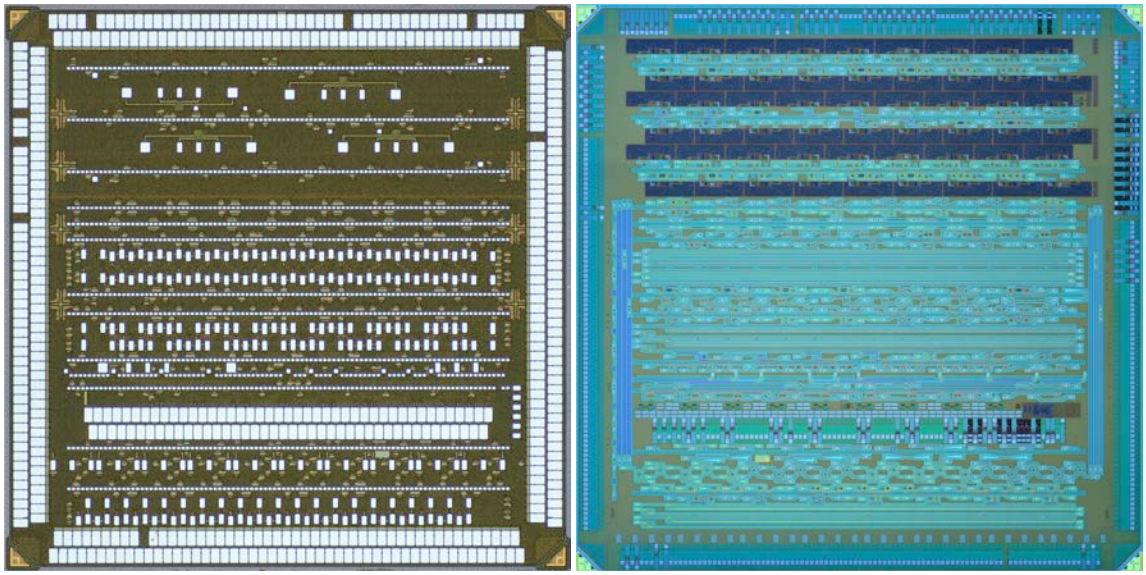

**Figure 3.16** Die photos showing front and back side of EOS8

#### *3.5.5.1. Data Volume*

Streamout took 45.2 seconds to generate a 238 MB .gds file

- Instances: 226,703 (EOS6) / 757,806 (EOS8) / 763,399 (EOS10)
- Arrays: 70,677 (EOS6) / 60,642 (EOS8) / 65,813 (EOS6)
- Paths: 175,759 (EOS6) / 174,546 (EOS8) / 174,678 (EOS6)
- Rectangles: 2,358,507 (EOS6) / 3,121,850 (EOS8) / 3,236,274 (EOS10)

#### *3.5.5.2. Design Rule Summary*

CPU time 53.8 hours, 13 hours real time 6,166 design rules 27,646 operations

### 3.5.6. EOS12 – IBM 45nm SOI-CMOS (12soi)

The first revision of the functional SOI-CMOS platform was taped-out in August 2011. No results are available at the time of writing for this thesis. The functional links that were initially demonstrated on EOS8 are arranged into an 8-channel multiplexed link with source-forwarded clocking. The digital backend for this link includes channel locking feedback and is internally known as the “megacell”. Longer length paperclip waveguides for both narrow and wide waveguides are included for more accurate propagation loss characterization. 32 variants of silicon germanium

photodiode designs are included in an attempt to improve the demonstrated responsivity of the integrated detectors.

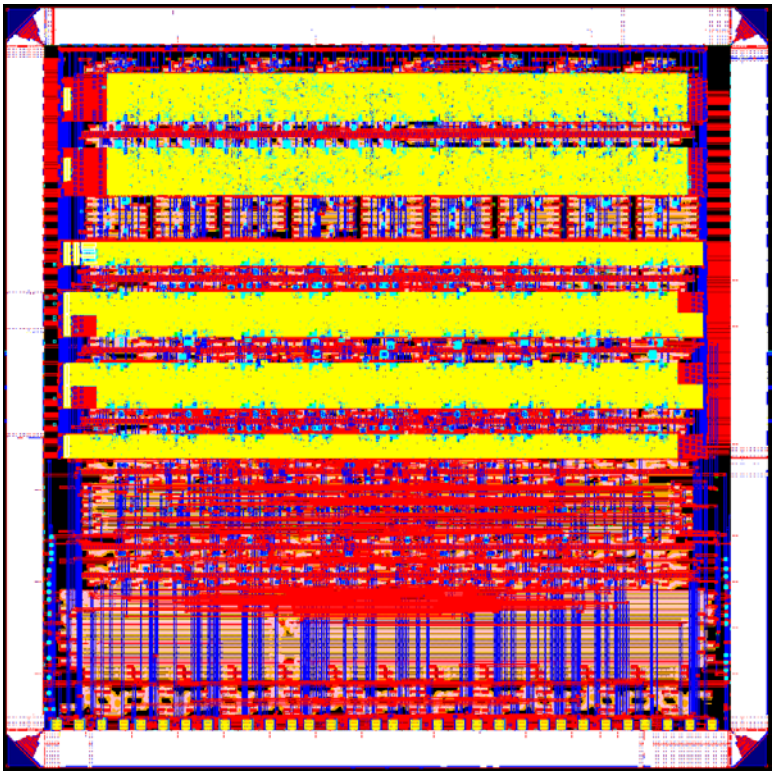

**Figure 3.17** Design environment screen shot of EOS12

#### *3.5.6.1. Data Volume*

Streamout took 59 seconds to generate a 202 MB .gds file

- Instances: 1,234,523
- Arrays: 49,014
- Paths: 762,561
- Rectangles: 1,821,365

#### *3.5.6.2. Design Rule Summary*

CPU time 12.6 hours, 2.3 hours real time 6,166 design rules 27,646 operations

## 3.6. Conclusions and Future Work

Since the initial integration demonstration, this design platform has enabled the fabrication of photonic devices in two other scaled-CMOS processes from two major

semiconductor manufacturers. Due to the modular nature of the Virtuoso environment, process information such as the names of required design layers can be isolated into a single technology file to enable code reuse. Several design rule violations existed for all of these designs, however, none had any impact on process yield. As a result all violations were waived and successfully completed the standard foundry data preparation flow. Final photonic device geometries such as the curve show no angular distortion and tight linewidth control from the highly Manhattan geometry optimized lithography and data preparation flow.

It has been a longstanding goal to release the code framework generated over the course of this thesis to the wider community. A step towards that goal is found in Appendix D, where the building block SKILL functions and multiple levels of p-cell hierarchy have been included. Due to non-disclosure and intellectual property constraints, all identifying information that is traditionally associated with such code has been stripped. The one function that must be modified to properly reference a given technology library is BlockWriter that provides an abstraction interface to all other cells. The included cells enable the compilation and use of the top-level resonator_cell. This is a simplified version of the cell used to lay out ring resonators, filter banks, modulators and detectors over the course of this thesis. It has the included functionality to realize many configurations of first order and second order resonant filters. The full p-cell infrastructure includes many more basic block discretized shapes, intermediate building blocks and flexible user interface instantiation objects. If the full code base was included in an Appendix, this thesis would be over 3000 pages. Instead a more appropriate release of this code that eliminates 3rd party intellectual property is planned when time allows.

Now that the layout of the photonic devices has been integrated within the standard electronic design environment, the advanced design tools developed for complex circuits can be leveraged for silicon photonics. Future work includes automatic verification of connectivity, known as layout-versus-schematic (LVS), by using existing tools such as Mentor Graphics Calibre to extract the photonic p-cells.

# 4. WAVEGUIDES AND PASSIVE DEVICES

This chapter introduces the physical integrated nanophotonic platform starting with waveguide fabrication and propagation loss analysis. Waveguides have been fabricated using both single-crystalline silicon and polysilicon as the core material. Single-crystalline silicon waveguides available in the thin-SOI process have yielded the lowest demonstrated waveguide losses of approximately 3 dB/cm for multiple wavelengths. Polysilicon waveguides in zero-change, bulk-CMOS processes have remained limited to greater than 50 dB/cm due to top-surface-roughness induced scattering. Since DRAM integration opens the possibility of process modification, customized polysilicon deposition conditions have enabled bulk-loss-limited waveguides to achieve 3 dB/cm waveguide loss. Utilizing this platform, a variety of passive photonic devices have been fabricated.

This chapter is organized to first introduce the relevant high-index contrast simulation tools that have been utilized to design the passive photonic devices. Next, the general considerations that have been used for waveguide sizing will be presented. The structures required for waveguide loss measurement will be introduced. The achieved waveguide performance will then be assessed for the various integration platforms studied in this thesis. The remainder of the chapter will present the design and measured results for the most important wavelength-division-multiplexed link passive devices: off-chip couplers and ring resonator filters.

## 4.1. High-Index Contrast Photonic Design Tools

Tools that have been verified to provide reliable passive photonic designs include Fimmwave and Omnisim from Photon Design as well as the open-source FDTD solver Meep. The modesolvers within Fimmwave yield accurate solutions for high index contrast structures. The included finite difference method (FDM) solver requires grid sizes on the order of 10nm to ensure accurate representation of the dielectric stackup that includes very thin layers, but the required computational domain can fit within 1 GB of memory for most structures to enable rapid design iteration. Perfectly matched layer boundary conditions enable the calculation of modal loss due to bends or substrate leakage.

Two dimensional propagation simulations for the vertical grating couplers were performed using the finite element frequency domain (FEFD) solver within Omnisim. In this type of reduced dimension analysis, the slab mode effective index for the structure in the third dimension is used for the represented silicon waveguides. This approximation greatly reduces computational complexity and is well suited to the large transverse mode sizes of the grating couplers as will be discussed in more detail in following sections.

Several integrated photonic structures such as the directional couplers that interface the ring resonators to the bus waveguides are not accurately approximated by the effective index method required for 2D simulation. In these cases, time consuming 3D-FDTD simulation is the only option. For this work, both Omnisim and Meep have been used to simulate coupling coefficients. Due to the time required to simulate each coupling coefficient, only a few dimensions are simulated and then interpolated between using an exponential fit of the coupling. An example coupling gap parameter sweep simulation in Omnisim over a wide range of gap widths as shown in **Figure 4.1** is used to verify this assumption. This simplifies calculations to set the coupling coefficients of many photonic devices as a function of wavelength, gap width and bus width using a set of sparse data points as shown in **Figure 4.2**.

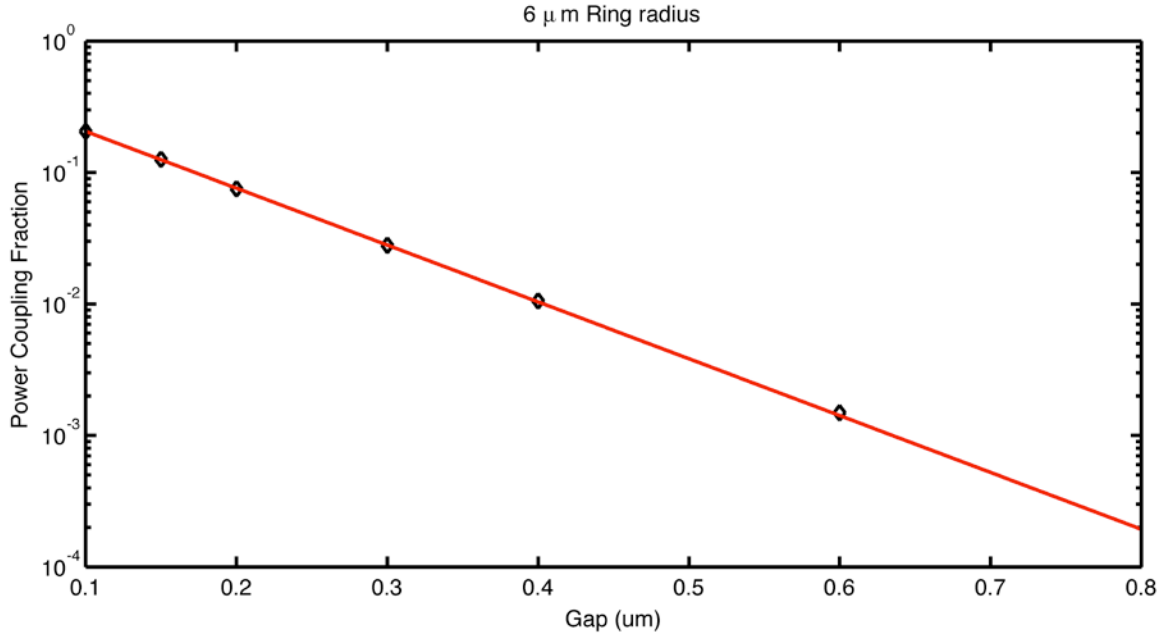


**Figure 4.1** Example FDTD simulated power coupling coefficients simulated over a wide range of bus-ring coupling gaps to demonstrate the suitability of the exponential couping fit.

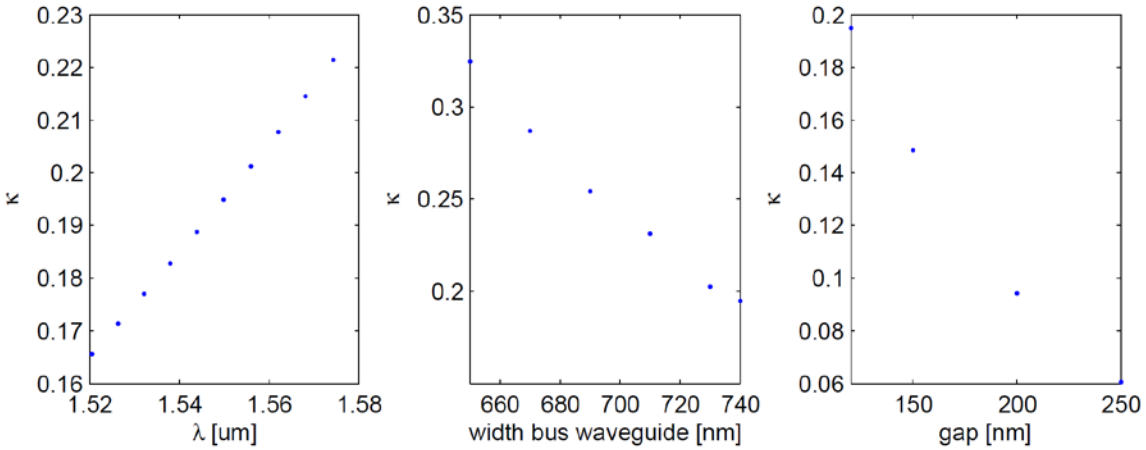


**Figure 4.2** Example FDTD simulated power coupling coefficients as a function of wavelength, bus width and gap for a 1550nm design.

The only exceptions to this general rule are cases where the excess loss due to coupling to radiation modes has an important impact on device function. In these cases, extra care must be taken in the 3D-FDTD simulation construction and parameter interpolation. Since scattered optical power from the simulation excitation that does not couple to the waveguide mode results in false signals on monitor power sensors elsewhere in the computational domain, it is important to utilize a simulator that can launch a separately simulated waveguide mode such as Omnisim as opposed to Meep. Additionally, in the monitoring of power coupled to each port it is advantageous to perform a mode overlap integral with the optical mode for the straight or bent waveguide instead of power flux sensors to further isolate coupled modal power from scattered light. In such a simulation, the excess loss of the directional couplers for the ring resonators can be calculated as shown in **Figure 4.3** for the same ring parameters used in **Figure 4.1**.

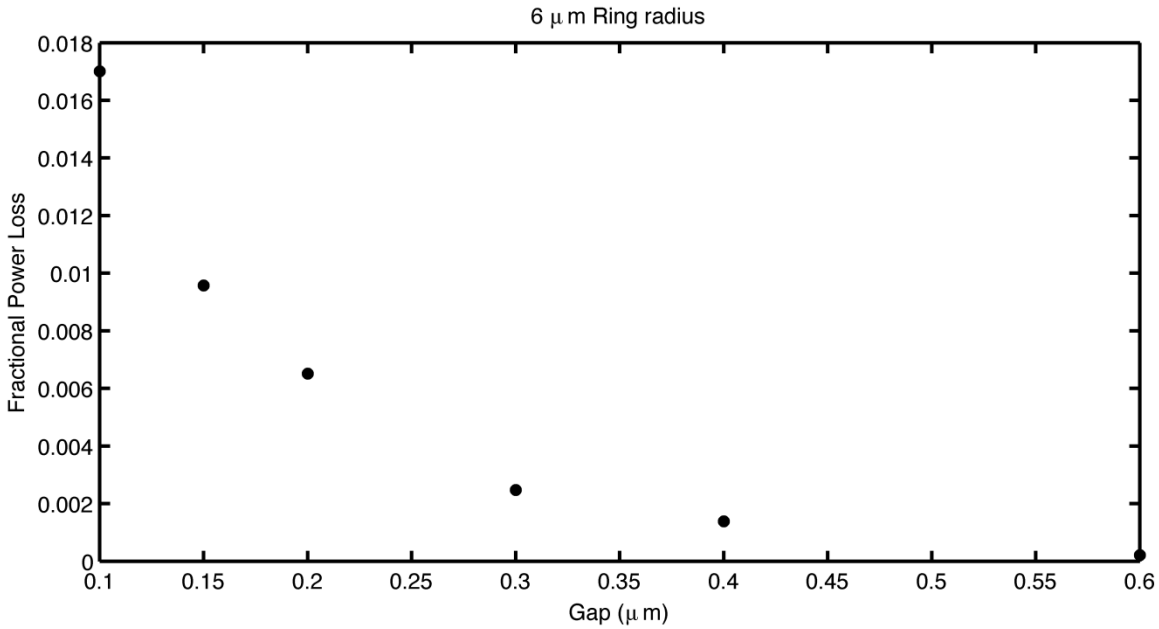


**Figure 4.3** Example FDTD simulated excess coupler loss simulated in Omnisim over a wide range of bus-ring coupling gaps.

## 4.2. Waveguide Design

For compliant process integration, the waveguide core material and thickness are largely fixed parameters for a given electronics flow. The sole concern for the waveguide designer is then the target width. Several concerns affect sizing.

The overall system is generally required to operate in a single transverse mode. Fundamentally, this requirement stems from the use of resonant devices for wavelength-division-multiplexing that can't efficiently route multiple modes. As such, coupling to higher order modes is indistinguishable from material absorption in that it leads to fundamental mode power loss. The nominal waveguide sizing constraint is therefore limited to be narrower than the width at which a second-order mode is guided. This statement is an inherently conservative constraint however, because the true system constraint is the minimization of coupling to higher order modes. Ensuring that the propagation constants of the first and higher order modes are sufficiently separated such that the bends, roughness and process variation do not induce coupling is the true constraint. However, due to the complexity of the structures designed in this work and the uncertainty associated with fabricating photonic devices in a simple, completely understood process flow, the conservative second-order cutoff condition has been used for waveguide sizing in this work. To determine this width, mode solving in Fimmwave is required. Since

most solvers will still display modes that exist only due to interaction with the domain boundary conditions, the second-order mode is determined to be cut-off when the solved effective index is less than the refractive index of the oxide cladding.

Further, since no fabrication process is perfect, adjustment must be made for the fact that the waveguides may be thicker and wider than designed. Since most fabrication processes have a well-defined "window," the second-order cutoff is simulated at the maximum possible thickness that can be expected. Since the silicon and polysilicon layer thicknesses are useful parameters for the electronic designer, this variability is typically provided in the standard process design kit. With the simulated width thus determined, the actual design width reduced from this value by the expected line width variance. This line width variance comes from two factors: the systematic biasing of the data preparation and lithographic process and the stochastic variance. In standard foundry processes, both parameters are often unknown. The biasing and variance may be determined by the use of electronic linewidth measurement structures across many die and process runs of the same process. Due to the long-term development effort in the IBM 12soi process, both of these variables may be extracted to aid in device design. The simulated design width is therefore shifted by the measured process bias and then narrowed such that the 3-sigma variance limit is below the second-order mode cutoff.

The waveguide width may also be generally smaller than the maximum single-mode width, but is typically useful only for special situations. The issue with reducing the waveguide width is that the mode becomes less confined and grows in size. This may result in loss due to either substrate leakage as the evanescent tail of the bound mode overlaps with the substrate to allow coupling or absorption as the overlap with the higher-lying metal layers in the process becomes significant. Additionally the bend loss and straight-bend junction loss may become significant concerns for the small radius waveguide bends that are often desirable for on-chip routing.

There are two significant cases where it may be desirable to use smaller waveguide widths. First, if the dominant source of loss results from the bulk material loss, e.g. the absorption and grain boundary scattering present in polysilicon, the lowest propagation loss may be attained for narrow waveguides by reducing the electric field overlap with the waveguide core. This loss scaling was observed for several processes in this work and will be discussed in detail in the

"Optimized Polysilicon Waveguides" section. Second, specific devices that require the interaction of multiple waveguides often favor smaller waveguide widths. A key example of this is in the coupling of the bus waveguide to a ring resonator. In a such a directional coupler, the input waveguide mode is decomposed into the supermodes of the two-waveguide system. Nominally, the two supermodes are the symmetric and anti-symmetric combinations of the individual waveguide modes. However, if both waveguides are near the second-mode cutoff, the combined system may also support higher-order supermodes. Any input light coupled to such a higher-order mode is then not coupled to either output waveguide mode at the end of the directional coupler and contributes only to device loss. In such cases, it is desirable to narrow the bus waveguide to cut off such supermodes in the coupled system. In both cases, the waveguides are traditionally tapered back to the nominal design with prior to a bend to get the benefits of the more strongly-confined mode.

There are also two important cases that require wide waveguide widths. The first, nearly universal need, is to expand the on-chip mode size to be more comparable to that of the ~10.5 μm single-mode fiber mode. This mode expansion is only required at the input and output interfaces and must be carefully designed to not excite power into the higher-order modes. The second need for wide waveguides is to reduce the contribution of edge roughness to the total waveguide loss. Since the scattering loss is proportional to the electric field overlap at the waveguide sidewall, wider geometries minimize propagation loss when other loss sources such as bulk absorption may be neglected [224]. This is often the case for low-loss silicon-on-insulator waveguides where other loss sources are already minimized. The width dependence of our silicon-on-insulator waveguides have not yet been measured experimentally, but structures for that purpose have been designed and are in fabrication at the time of writing of this thesis. When such width expansion is used for long distance routing, the waveguides are typically narrowed before bending or coupling to other devices such as ring resonator filters.

In general, the waveguide sizing is also specific to the wavelength of operation. This point is especially important in this work where device operation has been explored over the range of 1.15 μm to 1.64 μm instead of the traditional long-distance telecommunication work that is typically limited to no more than 100 nm bandwidth centered around 1.55 μm. Since the integrated electronic-photonic platform development may be suitable for a wide range of applications, a wide range of wavelengths needed to be examined. Additionally, the scaling of waveguide

loss with wavelength, in conjunction with geometric scaling dependencies, helps to identify the dominant loss sources in each photonic platform. Determining the loss sources for each technology has enabled lower waveguide losses to be demonstrated by process and device geometry engineering. As such, up to three single-mode waveguide sizings are examined in each technology to allow device operation and propagation loss extraction. The operating bandwidth of a given set of loss extraction test structures has been up to 150 nm. Similar to the previous discussion on process variation, the second-order mode cutoff should be calculated at the low-wavelength end of the desired operation bandwidth for conservative device sizing.

## 4.3. Waveguide Loss Test Structures

### 4.3.1. "Paperclip" Waveguides

The most frequently used method to test waveguide loss in III-V based materials where good facets are easily achievable by cleaving was to literally cut-back a waveguide in steps and measure the transmission for each length. The measured fiber-to-fiber insertion loss in dB may then be plotted against the length of the waveguide and fit to a straight line. The slope of that line would simply be the propagation loss of the integrated waveguides.

In silicon photonics, achieving good waveguide facets by cleaving is not nearly as easy. For example, in our work we utilize vertical grating couplers to avoid ever cleaving our samples for edge coupling. The loss must then be determined by various lithographically defined structures. It is also desirable to have set of test structures that is designed to characterize the waveguide loss as a function of waveguide width. The basic test cell is built from "paperclip" structures where different lengths of straight waveguides are connected by identical bends and coupling structures. Single-mode bends connect through tapers to the straight test section of various waveguide widths to ensure that the transmission of the fundamental mode is measured. Consistent input and output port locations for each test section length allows for the sample to be translated beneath the vertically coupled test stage for rapid characterization of multiple lengths with repeatable coupling conditions.

For a test chip to develop polysilicon waveguides in an integrated DRAM platform, eight different straight section lengths were included for each waveguide

width and wavelength center. The low loss test set comprised of three bend paperclips has total test section lengths of 60 μm, 3.3 mm, 6.9 mm, 10.5 mm. The high loss test set comprised of single bend paperclips has total test section lengths of 20 μm, 0.9 mm, 2.4 mm, 3.9 mm. The differential transmission can then be related to the propagation loss as in the cutback method. An example micrograph of two such test cells is shown in **Figure 4.4**. The mask includes waveguide loss test cells for 13 different waveguide widths between 200 nm and 2 μm repeated for wavelength centers of 1550 nm, 1400 nm and 1250 nm. Broadband grating couplers with 150 nm bandwidth below 10 dB insertion loss and conservative single-mode bend sizing then enable testing of waveguide loss over the continuous wavelength spectrum from 1150 nm to 1650 nm.

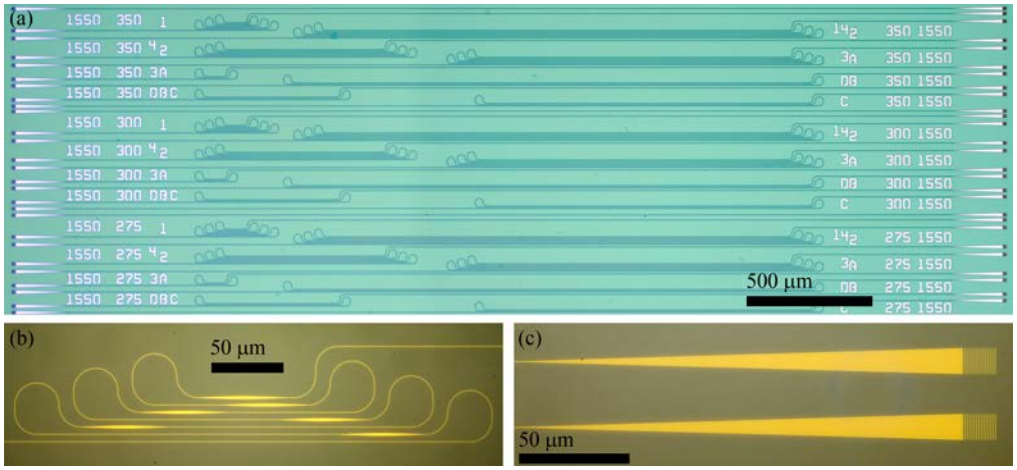


**Figure 4.4** (a) Optical micrograph of "paperclip" waveguide loss test structures for three waveguide widths. Four differential lengths each are used to measure propagation losses in two sets designed for high and low loss cases. (b) The width of the straight section is varied by introducing tapers between single-mode waveguide bends to ensure that the transmission of only the fundamental mode is measured. Single-mode lead widths of 376 nm, 476 nm and 600 nm are used for 1.25 μm, 1.4 μm and 1.55 μm wavelength centers respectively. (c) Uniform grating couplers with 200 μm long linear tapers are designed for 10.5 μm input mode diameter for an 8° off-normal incident angle. Design grating periods 819 nm, 974 nm and 1067 nm with duty cycles 37.5%, 37.5% and 42.5% are used for 1.25 μm, 1.4 μm and 1.55 μm wavelength centers respectively.

Similar loss as a function of test section length fitting of the measured data is then possible as in the cut-back method. An example set of measured transmission functions from a set of paperclip test structures where strong coupler reflections created rapid variations in insertion loss as a function of wavelength is shown in **Figure 4.5**. Even though the individual transmission measurements have

approximately +/- 2 dB fluctuations, the fit waveguide loss averaged over this wavelength range is measured to be 14.7 +/- 0.3 dB/cm.

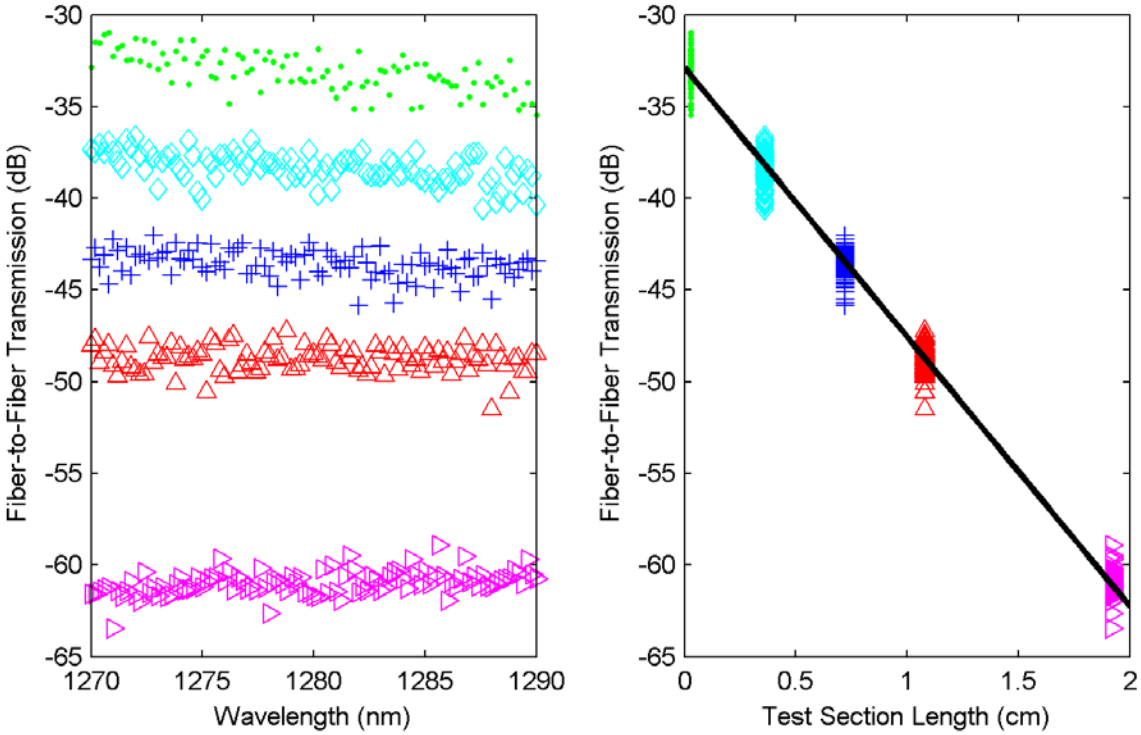


**Figure 4.5** Example of paperclip loss extraction as a function of a narrow range of wavelengths for five test section lengths.

It is also import to monitor the $R^2$ fit parameter of the paperclip data to ensure that the data is well matched to a linear transmission as a function of test section length relationship. If large deviations from the expected characteristic is observed in the fit data, defects from improper cleaning or handling damage may be the source and the measurement should be repeated on another sample. For the example data shown in **Figure 4.5**, the fit $R^2$ is 0.9998. The minimum $R^2$ for data to be considered valid for measurements included in this thesis is 0.98.

### 4.3.2. Loss Rings

The waveguide loss may also be extracted without coupling ambiguity by fitting the measured transfer function of a ring resonator to an analytical model. Since the quality factor and therefore the resonance width is a function of the round trip loss inside of the ring, the waveguide loss may be extracted from a single measurement. However, since the coupling in and out of the ring with the directional couplers also

contributes to the round trip loss of the ring, the extracted loss becomes more accurate as the coupling is reduced. The formula for the waveguide loss including coupling loss is given by the following formula:

$$Loss\ (dB/cm) = \frac{n_g}{\lambda} 10 \log\left(1 - \frac{2\pi}{Q} + \frac{2\kappa}{order}\right) \tag{4.1}$$

A fit for the extracted waveguide loss as a function of quality factor using the group index and coupling coefficients for the 28nm bulk-CMOS test chip is shown in **Figure 4.6**. Additionally, the optical mode may be modified by the bend radius, such that the closest approximation to the straight waveguide loss is achieved in a large radius bend. In this work, 20 μm radius rings with various coupling coefficients were placed on the chip with the weakest coupled ring with a measurable drop port signal used for loss extraction.

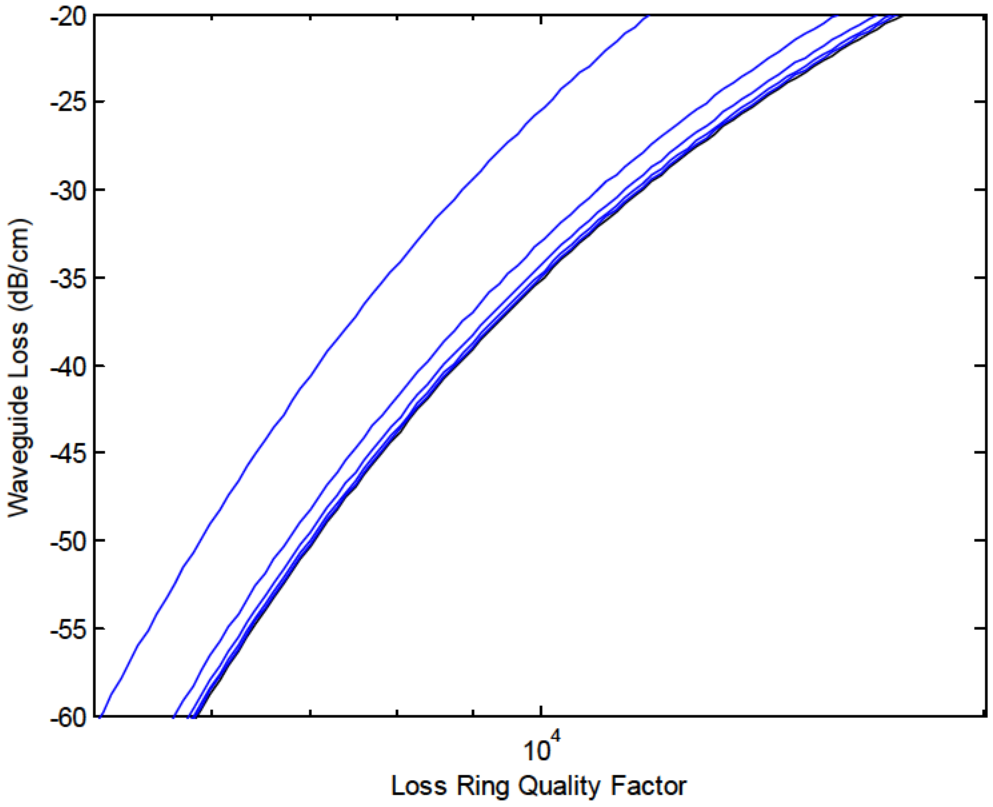


**Figure 4.6** Waveguide loss as a function of measured quality factor for the 5 coupling coefficients (1.4%, 0.31%, 0.12%, 0.05% and 0.02%) used on the Texas Instruments 28nm bulk-CMOS test chip (group index of 2.9). Since there is ambiguity in the fabricated coupling coefficient in relation to the design, the accuracy of this loss measurement technique improves as the coupling is weaker in proportion to the round trip cavity loss.

## 4.4. Measured Waveguide Performance

### 4.4.1. Single-Crystalline Silicon Waveguides

To be of interest to the broadest photonic community, the integrated waveguide loss must be comparable to the best available silicon photonic results using photolithography that have remained in the range of 1.5-3 dB/cm for strongly confined modes [68, 71, 115]. In the SOI-CMOS process, in contrast to the previous zero-change integration work in bulk-CMOS, the single-crystalline silicon layer that forms the transistor floating body may be patterned to form an oxide-clad, strip waveguide. Although the silicon layer thickness of less than 100 nm and the specific surrounding dielectric environment is unique to the deeply scaled electronics process used, the integrated waveguides otherwise closely resemble optimized waveguides from past electronic integration work and photonics-only processes. Strongly-confined waveguide propagation loss is measured to be approximately 3 dB/cm from 1260nm to 1350nm as well as from 1560nm to 1630nm as shown in **Figure 4.7**(a-b). The waveguide loss was measured in 4.1mm differential length paperclips. Due to space constraints, only two lengths of waveguides were included in contrast to the 4-7 test section lengths used on other test chips where more layout real estate was available. To acquire more accurate loss data, the measurement was repeated on four samples.

The loss measurement achieved from the paperclip waveguides can be verified by the included loss ring structures. Extracted intrinsic quality factors of 227,000 and 112,000 were obtained for 1280nm and 1550nm rings respectively as shown in **Figure 4.7**(c-d). At 1550 nm, the extracted waveguide loss using Eqn. 4.1 and a measured group index of 2.94 is 4.6 +/- 0.8 dB/cm. At 1280 nm, the extracted waveguide loss using Eqn. 4.1 and a measured group index of 3.92 is 3.7 +/- 0.3 dB/cm. The close agreement of these two independent loss characterization techniques confirms the low waveguide loss performance.

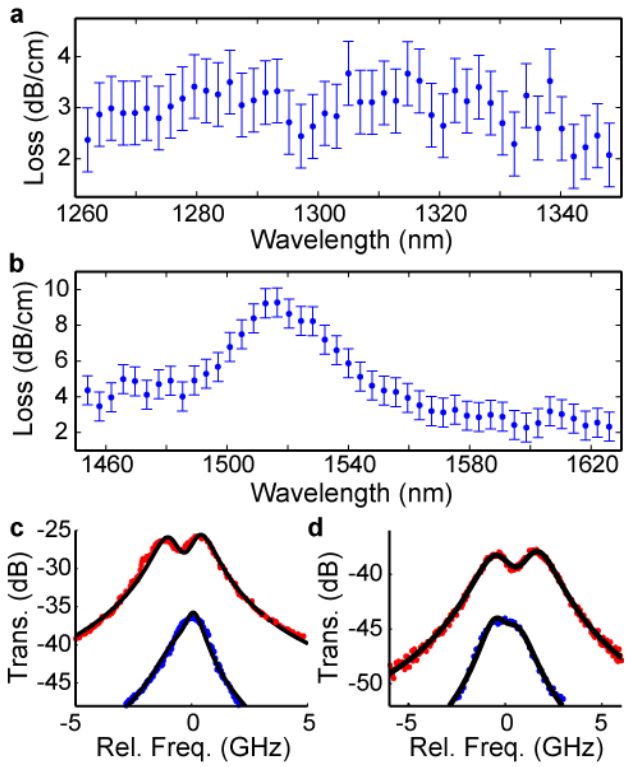


**Figure 4.7** Propagation loss in 470nm wide waveguides was measured from 1260nm to 1350nm (a) and in 700nm wide waveguides was measured from 1450nm to 1630nm (b). The observed loss peak reaching 9 dB/cm at 1515nm is hypothesized to result from molecular resonances in the surrounding dielectric material. Measured fiber-to-fiber transmission spectra of the drop ports of moderately-coupled (red data points) and weakly-coupled (blue data points) 20 μm radius rings at 1280nm (c) and 1550nm (d) are fit to coupled resonator models (solid black lines) for intrinsic quality factor extraction. Ring and bus waveguide widths were 470nm for 1280nm and 670nm for 1550nm. Coupling gaps of 830nm and 740nm were used for 1550nm; 660nm and 600nm were used for 1280nm.

The low waveguide loss, comparable with the state-of-the-art for most wavelengths, enables highly resonant photonic devices. These results are very encouraging to the long-term usefulness of a zero change CMOS platform. The absolute minimum achievable waveguide loss is of high interest in long distance power routing across chip. Since the test structures from initial chips only focused on the loss of a strongly-confined waveguide, the minimum achievable value in addition to confirmation of loss sources remain unknown. For the EOS12 tapeout of August 2011, loss paperclips for both wide and narrow waveguides were included with long test length sections to enable the accurate measurement of a lower waveguide loss.

### 4.4.2. Zero-Change Polysilicon Waveguides

Due to the reliance on deposited microelectronic polysilicon as the waveguide core material, propagation losses in the zero-change bulk-CMOS processes studied for this thesis are significantly higher than reported silicon photonic devices. The fundamental problem for zero-change photonic integration is that the roughness of the top surface of the polysilicon has little to no importance to electrical functionality. As such, the polysilicon is not optimized to have a low top surface roughness. Instead, the polysilicon grain structure is deposited in the columnar grain structure, shown in **Figure 4.8**, that is optimized to promote vertical diffusion of dopants along grain boundaries [225]. Since the grains are growing during deposition, the surface is formed by the differential growth rates for the crystal orientations of each grain. The resulting rough surface is also clearly visible in **Figure 1.1**. The experiments conducted in various bulk-CMOS processes have supported the theory that the loss of these integrated polysilicon waveguides are limited by this top surface roughness.

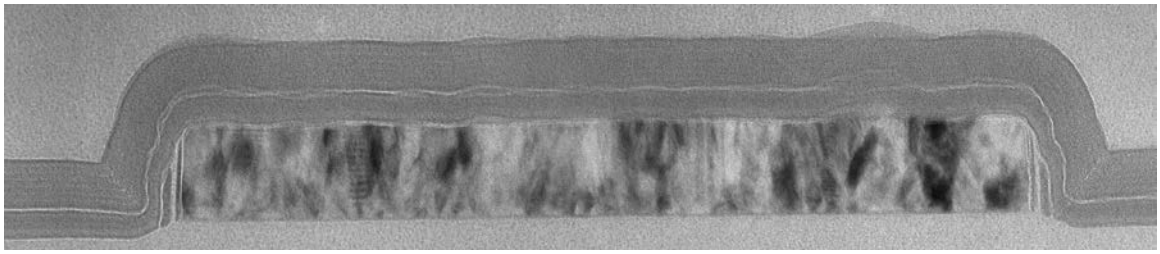

**Figure 4.8** Cross-sectional transmission electron micrograph (TEM) of a polysilicon waveguide integrated in a 28nm bulk-CMOS process. The fabricated core thickness is approximately 73 nm.

The first bulk-CMOS test chip (EOS1) only examined waveguide loss in the 1280nm wavelength region. The paperclip waveguide loss measurements, shown in **Figure 4.9**, indicated high waveguide losses for both the 345 nm wide and the 1185 nm wide waveguides. The ~76 dB/cm wide waveguide loss was significantly greater than the ~55 dB/cm waveguide loss. The loss scaling ruled out that the dominant loss source was the edge roughness, but did not strongly distinguish between the bulk material loss and top surface roughness. Both phenomena are expected to have similar scaling in thin waveguide geometries, the electric field overlaps with the top

surface and the overall waveguide core as a function of width are nearly proportional.

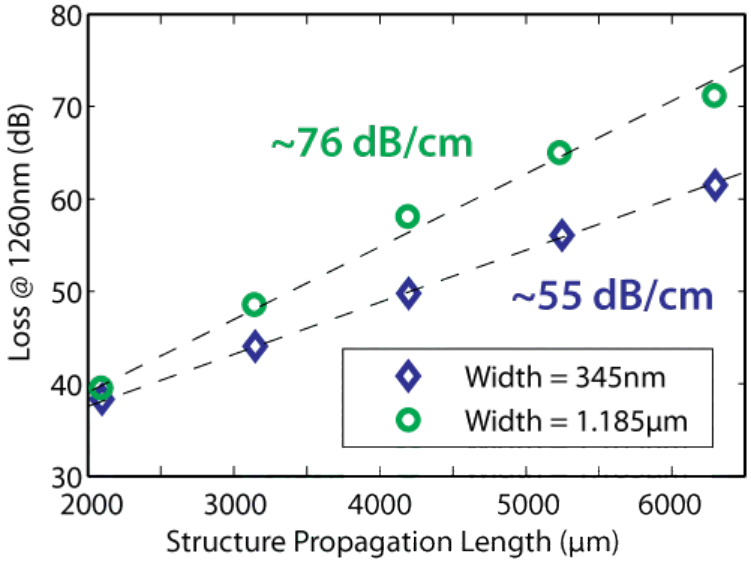


**Figure 4.9** Loss measurements for two different test section widths at 1260 nm for the 65 nm bulk-CMOS test chip manufactured at Texas Instruments (EOS1).

The second bulk-CMOS test chip examined waveguide loss at 1550nm and 1280 nm in a more deeply-scaled CMOS process that had the thinner polysilicon core thickness shown in **Figure 4.7**. Measured by two methods in **Figure 4.8**, waveguide propagation loss is approximately 55 dB/cm from 1520 to 1580 nm. At 1280nm, however, the waveguide loss was measured to be between 85 dB/cm and 95 dB/cm, but the high loss prohibited accurate characterization. The relative increase of the waveguide loss compared to the 65nm chip, utilizing similar, but thicker polysilicon material, identified the most likely waveguide loss source to be the top surface roughness. This hypothesis was further confirmed by estimating the predicted waveguide loss due to the roughness of 5 nm RMS with a 100 nm correlation length as measured by several TEM micrographs. Extrapolating theoretical loss curves from a published sidewall roughness analysis by flipping both waveguide geometry and electric field polarization, the waveguide sensitivity to top-surface roughness is approximately 10 dB/cm per nm RMS roughness for the measured 100 nm correlation length [224]. The close agreement of the 50 dB/cm prediction offered further confirmation that the top surface roughness was the dominant contribution to overall integrated polysilicon waveguide loss.

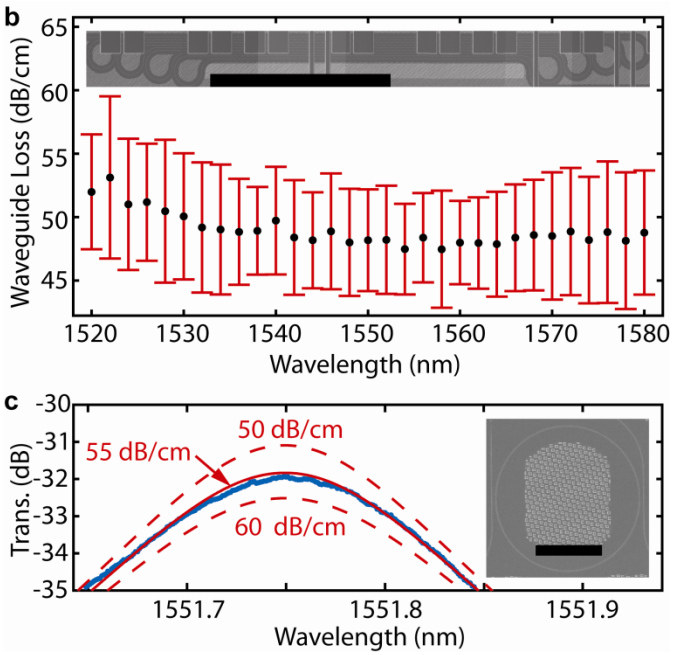


**Figure 4.10** Waveguide propagation loss (b) calculated by the differential loss through 2 waveguide paperclips (SEM of one paperclip shown in inset with 200 μm scale bar) with a length difference of 2.72 mm. Error bars calculated as the standard deviation for 4 samples. (c) Transmission of the drop port of a loosely coupled 20 μm radius ring resonator (SEM shown in inset with 20 μm scale bar). Measured data (blue dots) is most closely fit with a simulated ring resonator transmission response with a 55 dB/cm waveguide loss (solid red line). The measured quality factor for the loosely coupled resonator was 7960. The sensitivity of this technique is illustrated by the divergence of the 50 dB/cm and 60 dB/cm simulated responses (dotted red lines). A coupling coefficient of 9 x $10^{-3}$ was used for simulation by simultaneously fitting the measured through port transmission.

Within the silicon-on-insulator platform, the single crystalline silicon layer provides a suitable low loss waveguide for most all integrated photonic applications. There are some cases, however, where it is of interest to combine this silicon layer with the polysilicon layer that is still present in the process as the transistor gate. Since the gate formation is similar to that of the bulk-CMOS process, the waveguide loss is again high. In the IBM 45nm SOI-CMOS process, the polysilicon layer is thinner than the reported bulk-CMOS results. Due to the above mentioned scaling of loss with top surface roughness, the loss of the integrated polysilicon waveguides is higher than was measured in the bulk-CMOS process. The waveguide loss as measured by paperclip waveguides for strongly-confined singlemode geometries was ~75 dB/cm at 1550nm and ~110 dB/cm at 1280nm.

Beyond hybrid uses with the single-crystalline silicon body in the SOI-CMOS process, zero-change polysilicon waveguides in the bulk-CMOS process may still

offer a suitable photonic platform for many applications. Weakly-coupled rings with quality factors approaching 8,000 demonstrate that the waveguide loss is low enough to enable resonant photonic devices. Modulators, detectors and wavelength division multiplexing filters have been demonstrated in this platform over the course of this thesis. Although the zero-change bulk-CMOS platform is not under further development for the microprocessor-to-memory application in our team, this platform may be useful for systems that do not require long distance on-chip routing where waveguide loss would cause the largest performance impact.

### 4.4.3. Optimized Polysilicon Waveguides

While the single-crystalline waveguides within the SOI-CMOS process presents a complete, low-loss photonic platform for the microprocessor photonic interconnect integration, all high-volume memory manufacturing (DRAM) is performed using bulk silicon wafers where such a layer is not present. As presented above, the zero-change bulk-CMOS polysilicon waveguides resulted in waveguide losses exceeding 50 dB/cm. Since cross-chip waveguide routing on the centimeter scale DRAM die would improve system performance, it is important to reduce the integrated polysilicon waveguide loss. Due to the processing flexibility inherent in the DRAM process, it is possible to optimize the deposition and processing conditions used for waveguide fabrication. Such an optimization step may also be possible within a CMOS process during the development stage if the future integrated photonic application becomes of sufficient importance to justify the foundry investment. Similar process tweaks that enable functionality beyond basic digital circuits has precedence in the analog circuits community that has resulted in the inclusion of high quality capacitors, triple-well implants and modified threshold transistors.

#### *4.4.3.1. Deposition Condition Tradeoffs and Summary*

In this work, I designed a short-flow waveguide performance characterization mask that was fabricated under multiple polysilicon deposition and anneal conditions by Micron Technology. The primary fabrication optimization condition was to reduce the top surface roughness of the polysilicon layer that was determined to be the limiting loss mechanism in the bulk-CMOS work. The polysilicon in most CMOS processes is deposited above the crystallization threshold temperature such that the nucleated grains grow in the deposition direction. Since dopant diffusion is greatly accelerated along grain boundaries, the columnar grain

structure that results from this high temperature deposition favors diffusion in the vertical direction. This grain structure engineering allows the implanted, self-aligned source/drain dopants diffuse to form a high dopant density at the bottom of the polysilicon film. This helps to reduce polysilicon depletion effects in the transistors that would cause larger effective gate oxide thicknesses. This same grain structure, however, results in a top surface that is determined by the grain growth effects that are shaped by differential crystal growth rates. This presents a tradeoff for photonic integration. Two approaches have been taken to reduce the top surface roughness to integrate low loss polysilicon waveguides: optimizing deposition and anneal conditions; and post-deposition polish recipe development.

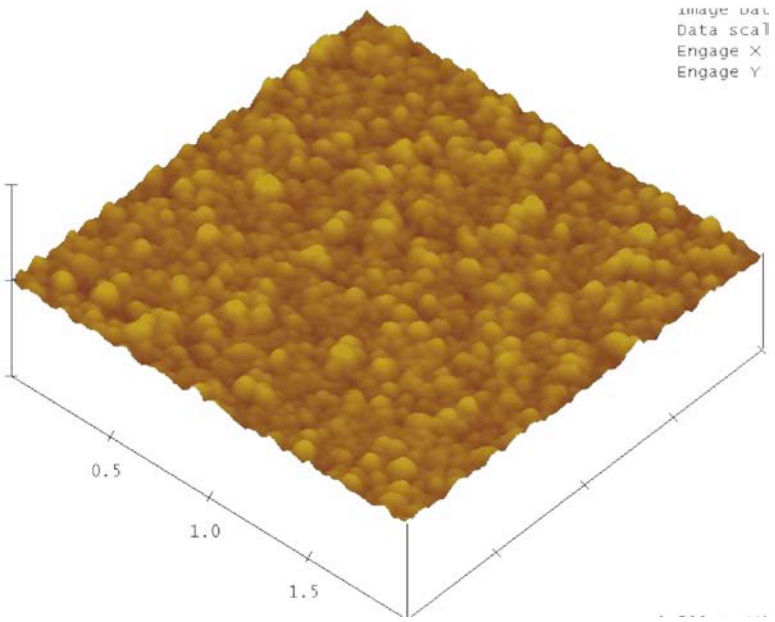


**Figure 4.11** Measured 2μm x 2μm atomic force microscope (AFM) surface topology standard process columnar grain structure polysilicon for the Fab 12 memory integration process. Measured roughness of 4.1 nm RMS. Z-axis range is ±150 nm.

To avoid a rough top surface formed by grain growth during deposition, a recipe from an existing memory process for an amorphously deposited silicon layer that is crystallized during later thermal processing was chosen. The amorphous nature of the initial deposition results in a smooth film analogous to a traditional dielectric deposition. The crystallization process then results in a square grain structure that does not significantly roughen the film top surface as has been shown in past work. Although the amorphous silicon in the standard process was incidentally crystallized by the later thermal cycling of the electronics process, specific anneals

were added to the process to control crystallization conditions. Since the polysilicon is deposited at an early point of the process with a very high thermal budget, the insertion of additional anneals does not pose process integration concerns. The initial study focused on two types of crystallization conditions: a rapid ~950°C, 20 second spike anneal and a longer ~600°C, 30 minute anneal. Due to the higher activation energy of crystal nucleation as compared to crystal growth, it is expected that the longer, lower temperature anneal would result in a larger grain size than the higher temperature anneal. In agreement with prior literature, both anneal conditions resulted in smooth films both at Fab 4 and were repeatable in transfer to Fab 12 for the memory integration process. As measured by atomic force microscopy (AFM) in Figure 4.12 through Figure 4.14, the post-crystallization roughness remains between 0.3 nm and 0.5 nm RMS for both anneal conditions.

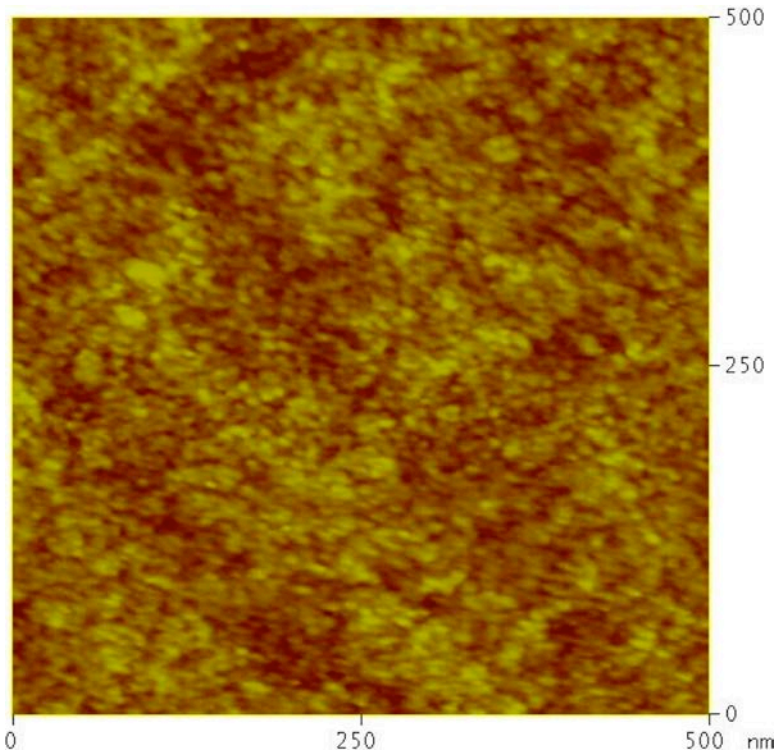


**Figure 4.12** Measured 500nm x 500nm atomic force microscope (AFM) surface topology of ~950°C, 20 second post-depostion crystallizing anneal at Micron Fab 4. Measured roughness of 0.3 nm RMS with an 18 nm characteristic length.

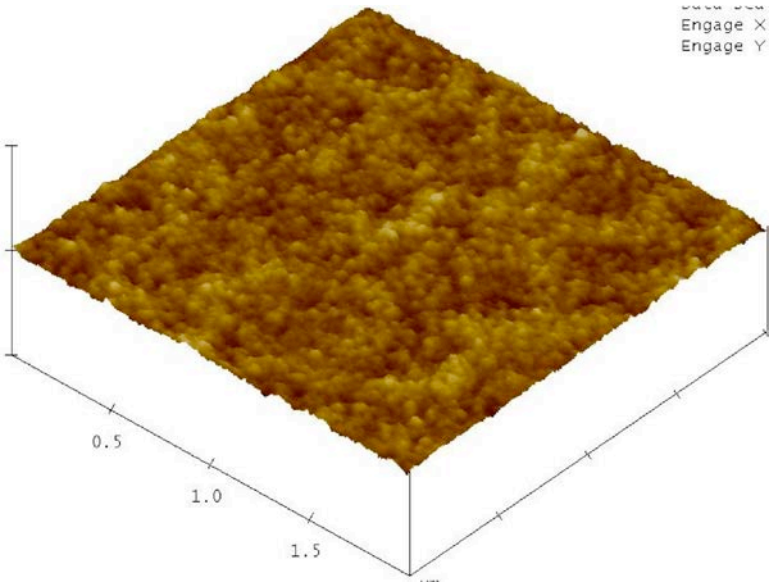


**Figure 4.13** Measured 2µm x 2µm atomic force microscope (AFM) surface topology of ~950°C, 20 second post-depostion crystallizing anneal for the Fab 12 memory integration process. Measured roughness of 0.5 nm RMS. Z-axis range is ±30 nm.

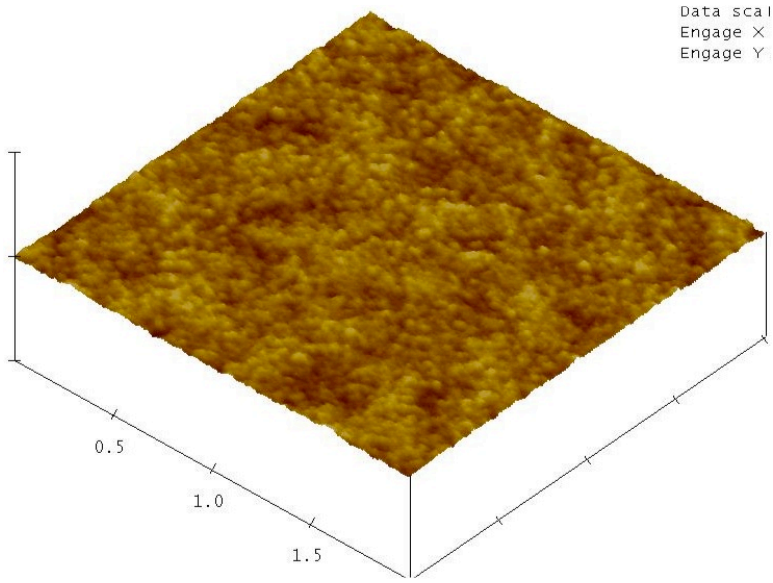


**Figure 4.14** Measured 2µm x 2µm atomic force microscope (AFM) surface topology of ~600°C, 30 minute post-depostion crystallizing anneal for the Fab 12 memory integration process. Measured roughness of 0.5 nm RMS. Z-axis range is ±50 nm.

The other roughness-reduction approach has been to polish the polysilicon. The first attempt utilized a standard chemical-mechanical polishing recipe that is used to planarize wafers containing various oxide and silicon features. To achieve a consistent polish rate between materials with different hardnesses, an isotropic

etchant etches the harder material in such a recipe. Due to the differential crystal plane etch rates of this chemical, however, such unoptimized CMP recipes typically result in roughening the polysilicon film as shown in Figure 4.15. In this first experiment, the initially smooth film is actually made rougher to result in a measured 2.9 nm RMS surface topology with clear polycrystalline etch planes.

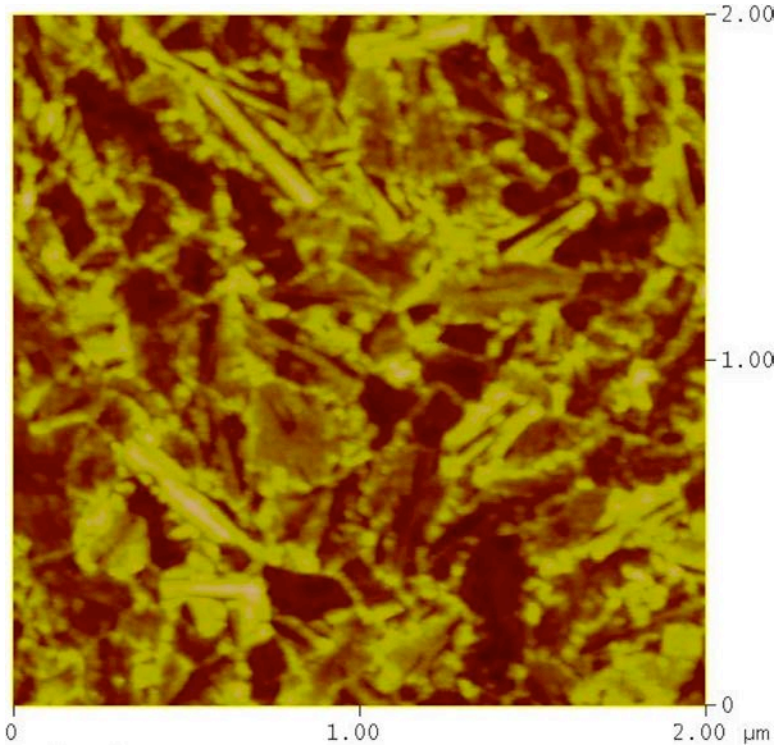


**Figure 4.15** Measured 2µm x 2µm atomic force microscope (AFM) surface topology of original chemical-mechanical polish (CMP) recipe on the ~950°C, 20 second post-depostion-crystallized polysilicon at Micron Fab 4. Measured roughness of 2.9 nm RMS with a 78 nm characteristic length.

Removing the chemical component of the polish is possible in applications where only one material is exposed across the wafer. Since the polish can occur immediately after deposition and/or crystallization, a purely mechanical polish was explored for producing smooth polysilicon films. The first experiment attempted to polish the post-deposition-crystallized polysilicon films that had starting roughnesses of 0.5 nm RMS. For both anneal conditions, the film roughnesses were observed to increase to 1.3 nm and 0.7 nm RMS for the low temperature and high temperature anneals respectively. The larger roughness increase was observed for the larger grain size polysilicon and the root cause of the poor performance of the

mechanical polish on the square grain structure polysilicon was hypothesized to be complete grain removal. Since the standard process transistor gate polysilicon is deposited above the crystal nucleation temperature, the columnar grain structure extends vertically throughout the film. This grain structure makes the removal of entire grains during the mechanical polish very unlikely. Using the same polish recipe on the columnar film, the low surface roughness of 0.1 nm RMS was achieved. This result supports the hypothesis of the complete grain removal as the polishing limit of the square-structure polysilicon.

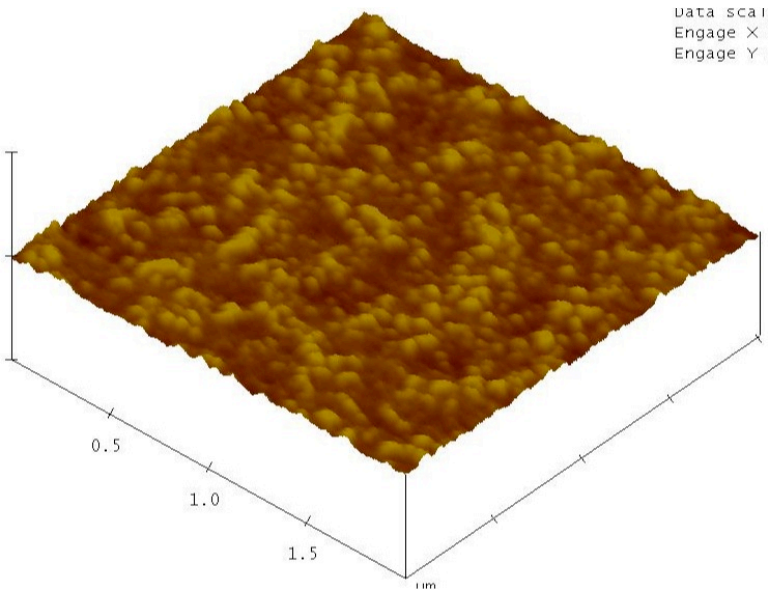


**Figure 4.16** Measured 2µm x 2µm atomic force microscope (AFM) surface topology of the mechanically-polished, ~600°C, 30 minute post-depostion-crystallized polysilicon at Micron Fab 12. Measured roughness of 1.3 nm RMS. Z-axis range is ±50 nm.

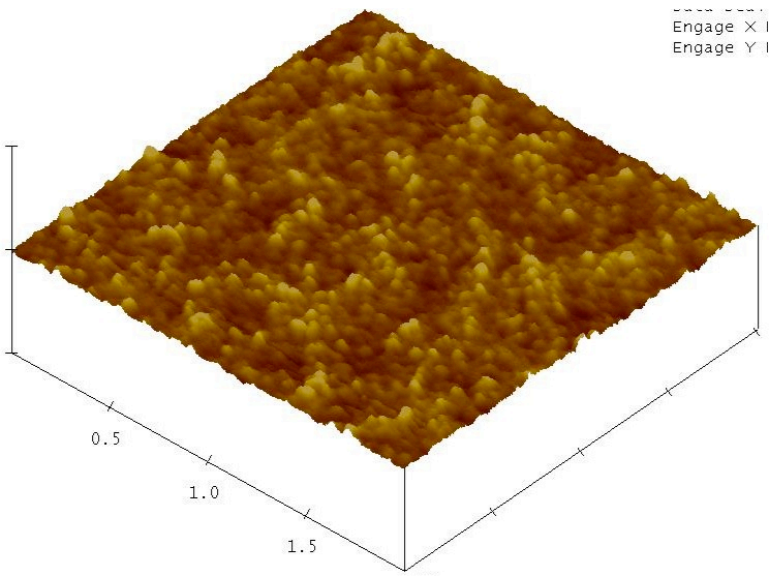


**Figure 4.17** Measured 2µm x 2µm atomic force microscope (AFM) surface topology of the mechanically-polished, ~950°C, 20 second post-depostion-crystallized polysilicon at Micron Fab 12. Measured roughness of 0.7 nm RMS. Z-axis range is ±30 nm.

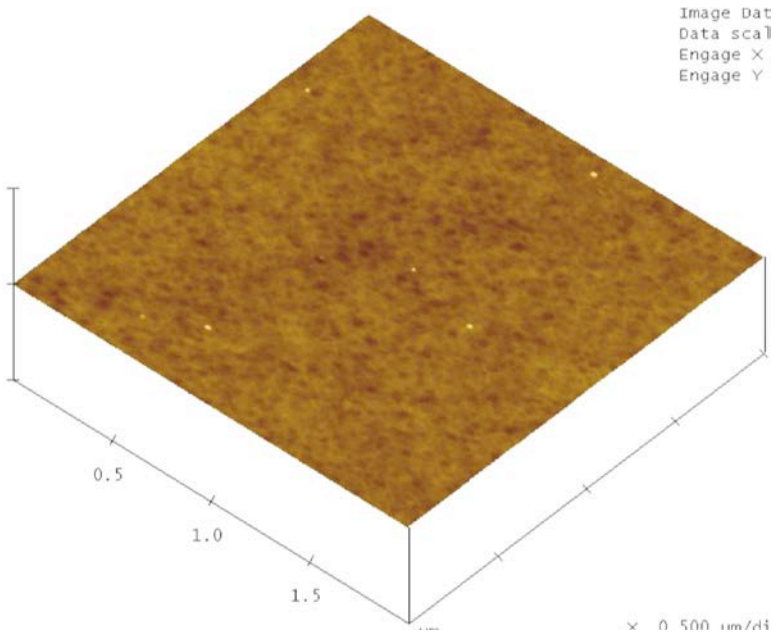


**Figure 4.18** Measured 2µm x 2µm atomic force microscope (AFM) surface topology of mechanically-polished, standard process columnar grain structure polysilicon for the Fab 12 memory integration process. Measured roughness of 0.1 nm RMS. Z-axis range of ±150 nm matches the pre-polished AFM data shown in Figure 4.11.

The various deposition conditions and polish recipe development has occurred over two fabrication runs at two different fabrication facilities within Micron. The

resulting loss for wide waveguides has been summarized in the table below for 1280nm and 1550nm. The wide waveguides are chosen for comparison as the bulk loss should dominate with an effective confinement factor close to 100% for all conditions. The initial work yielded reliable waveguide loss data for only the ~950°C post-deposition crystallization condition. A repeat of this condition in a different fabrication facility for a thicker layer has yielded much lower waveguide loss. It is currently hypothesized that either higher hydrogenation fraction in the amorphous silicon deposition tool or the presence of a high hydrogen content nitride liner are combining to increase defect passivation in the new process flow.

| Deposition and Anneal Conditions | Polysilicon Thickness | Fab Facility | Process Thermal Cycling | Wide WG λ=1280 nm Loss (dB/cm) | Wide WG λ=1550 nm Loss (dB/cm) |
|---|---|---|---|---|---|
| a-Si Deposition ~950°C 20s Anneal | 120 nm | Fab 4 (D-1) | Full DRAM | 32.5 ± 0.5 | 19.8 ± 0.4 |
| a-Si Deposition ~950°C 20s Anneal | 120 nm | Fab 4 (D-1) | None | 23.4 ± 1.0 | 13.4 ± 0.4 |
| a-Si Deposition ~950°C 20s Anneal | 220 nm | Fab 12 (D0) | None | 6.0 ± 0.7 | 3.0 ± 0.8 |
| a-Si Deposition 600°C 30m Anneal | 220 nm | Fab 12 (D0) | None | 9.1 ± 0.7 | 4.5 ± 0.9 |
| poly-Si Deposition POR + Polish | 220 nm | Fab 12 (D0) | None | 13.8 ± 0.7 | 5.5 ± 0.5 |

**Table 4.1** Summary of experimentally measured polysilicon waveguide losses for various deposition and anneal conditions.

In comparison to the high temperature crystallization condition, the lower temperature anneal and columnar poly-Si recipe are observed to have higher loss. Since the top surface roughness of these films were measured to be equal to or lower than the ~950°C anneal condition, the difference is expected to result from a change in the bulk material loss. The loss that scales with the bulk of the polysilicon material can result from either defect state absorption or scattering at the grain boundaries. These two phenomena are expected to have inverse scaling relations with grain size. Defect state absorption should be proportional to the relative volume of grain boundaries in the material and should decrease with larger grain sizes. For grain sizes below the wavelength of light in the media, scattering is expected to result in larger propagation losses as the grain size and therefore

correlation length of the index heterogeneity increases. Consistent with the comparison to the ~950°C polysilicon between the two fabrication runs, increased hydrogen content appears to have resulted in a significantly higher fraction of passivated grain boundary states. Grain boundary scattering is then hypothesized to be the dominant loss source for the polysilicon deposition conditions studied for D0. More detailed analysis will be discussed in the following sections to identify the loss sources of the original D-1 polysilicon loss study.

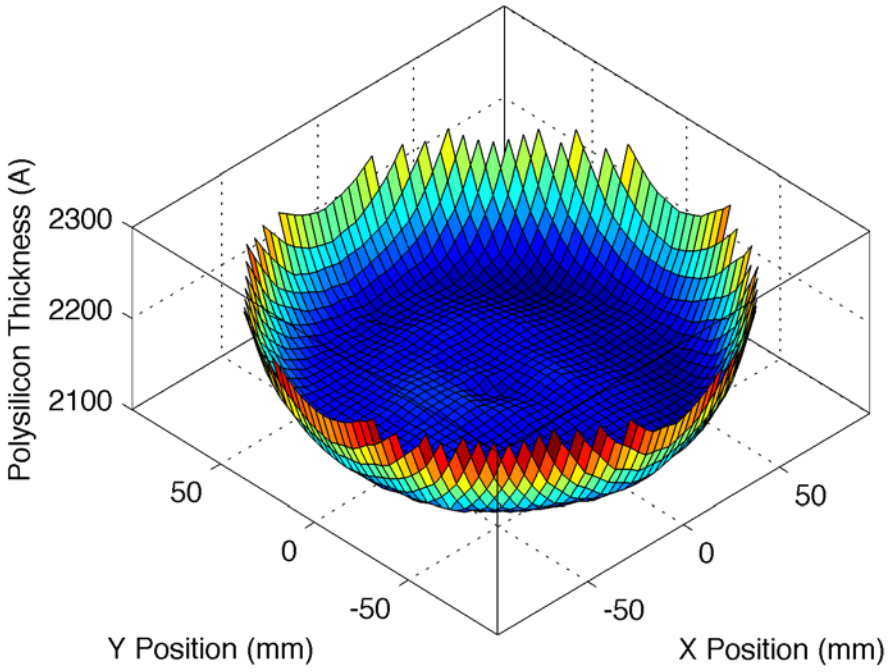


**Figure 4.19** Cross-wafer thickness uniformity for a polished polysilicon wafer as measured by an in-fab ellipsometric depth gauge for 600 points. Spread of maximum to minimum thickness is 14.5nm.

Modifying the polysilicon deposition conditions for the transistor gate requires further consideration for the impact on the electronic process. Since the columnar polysilicon is already used as the transistor gate in the process, the additional polish step adds complexity to the process but poses no further integration challenges. The polish does provide extra complications for the integrated photonic devices. First, the propagation loss of the polished POR polysilicon yielded the highest loss of the attempted polysilicon deposition recipes. More problematic, however, is the cross-wafer thickness variability introduced by the mechanical polish. As shown in Figure 4.19, the center-to-edge thickness variation of the polysilicon layer is approximately

15nm after the polish. This is an order-of-magnitude increase over the amorphous polysilicon cross-wafer variation that is shown in Figure 4.20. The nearly 10% uncertainty in the core thickness for the polysilicon waveguides would result in large variations in the photonic devices.

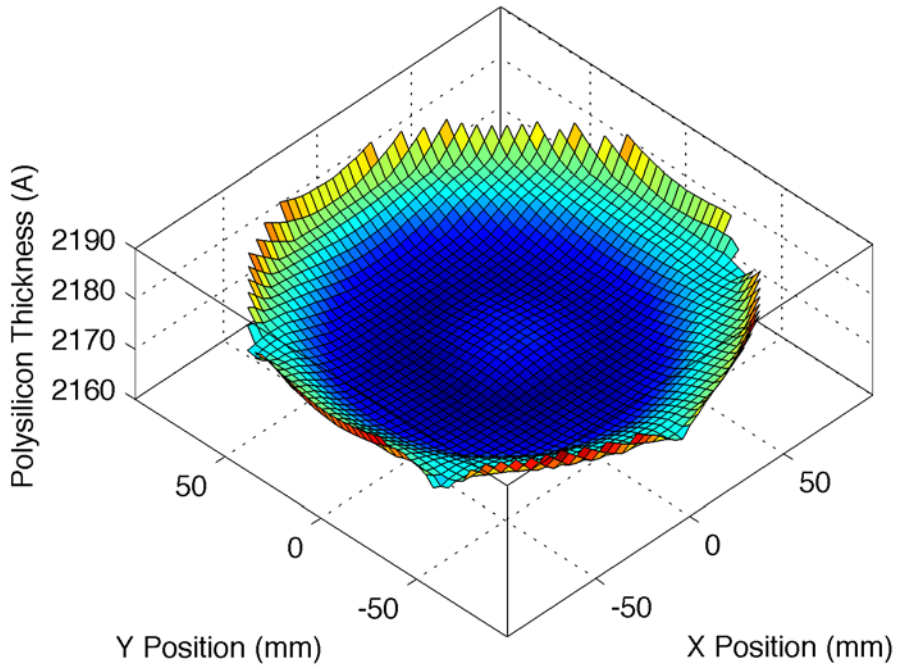


**Figure 4.20** Cross-wafer thickness uniformity for amorphously deposited polysilicon as measured by an in-fab ellipsometric depth gauge for 600 points. Spread of maximum to minimum thickness is 1.4nm.

Changing the polysilicon grain structure by post-deposition crystallization has been measured to shift the transistor characteristics of the standard process through embedded electronic test structures. As described previously, the columnar grain structure of the POR polysilicon optimizes vertical diffusion of the dopants to enable a high carrier density at the bottom oxide interface. Although the ~950°C anneal condition was taken from another memory process that used this exact recipe for the transistor gate, an extra 1.5Å of effective gate oxide thickness was observed in the target integration process through poly depletion. The resulting threshold voltage shift prevents transistors from meeting all performance targets. Process compliance may be maintained by compensating for the threshold voltage shift by adjusting the transistor implant doses. However, while the process integration issues surrounding the amorphous polysilicon conditions are being

resolved, the mechanical polish process of the POR polysilicon will yield sufficient photonic performance. Since the cross-wafer thickness variation is very predictable and ~51% of the wafer falls within ±1 nm of the median thickness, the polish may still be useful for process development while ensuring that the transistor performance meets design targets.

#### *4.4.3.2. Confinement Factor Scaling of Bulk-Loss Limited Waveguides*

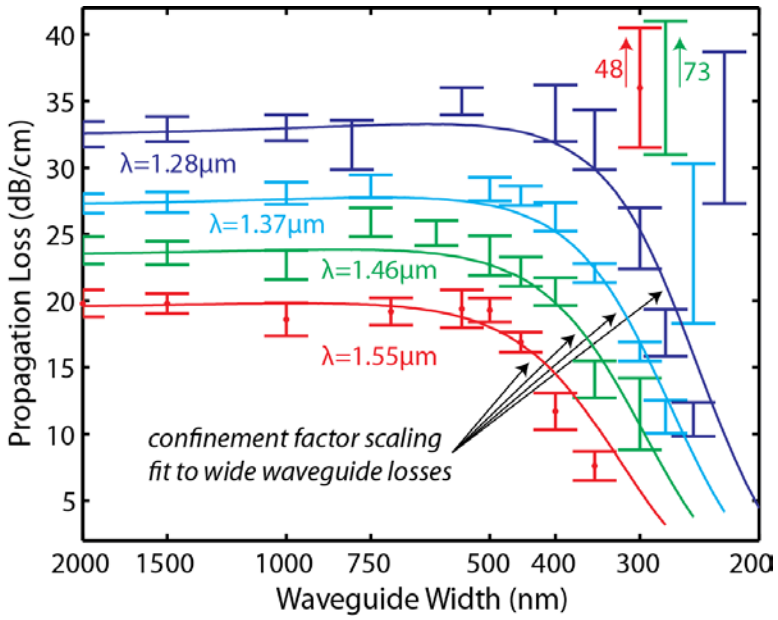


**Figure 4.21** Measured propagation loss as a function of wavelength and waveguide width for a thermally processed wafer with a 120 nm polysilicon thickness. Simulation curves for the confinement factor scaling of waveguide bulk loss are fit to the measured loss of the wide waveguides and shown for comparison.

The losses of the various test cells, shown in **Figure 4.21**, were then measured for wafers with a 120 nm polysilicon thickness after the thermal cycling representative of the full electronics process. Several trends emerge from the data. First, reduced waveguide widths enable confinement factor scaling to reduce the contribution of the propagation loss from the polysilicon material as illustrated in **Figure 4.22**. Measured propagation loss scales with decreasing confinement factor down to approximately 30%. This enables measured waveguide losses below 15 dB/cm across the telecommunications spectrum despite significantly higher waveguide loss at wider waveguide widths. A waveguide width of 350 nm enables the lowest reported propagation loss of 6.2 dB/cm for a thin-core polysilicon waveguide at 1550 nm.

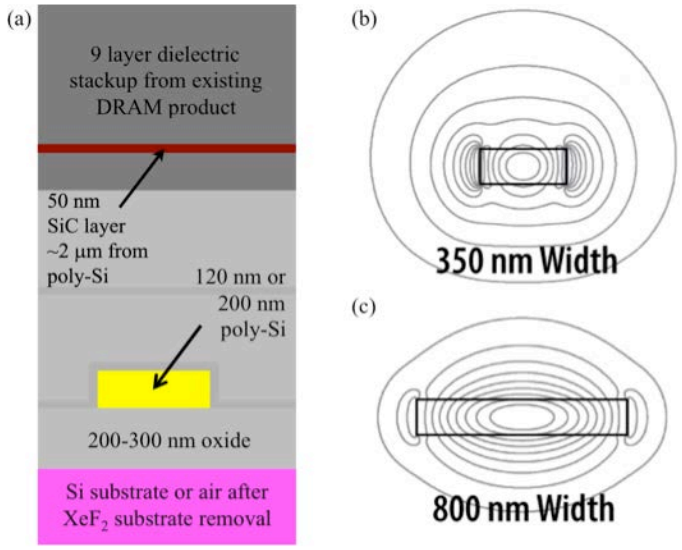


**Figure 4.22** (a) Cartoon cross-section of the memory process used for this work. All unlabeled dielectric layers in the immediate proximity of the waveguide core have a refractive index close to that of fused silica at the wavelengths of interest. (b) Waveguide mode profile contours for a narrow waveguide at 1550 nm with 120 nm polysilicon layer thickness illustrating the asymmetry introduced by the substrate removal and low polysilicon guided power fraction. (c) High-confinement waveguide modes such as that of an 800 nm waveguide width show no observable asymmetry and clearly confine the majority of the light in the polysilicon core region.

The second trend is the abrupt end to the loss reduction through confinement factor scaling achieved by narrowing the waveguide width for each wavelength. If this were to be attributed to sidewall scattering having a larger impact on propagation loss, a more gradual transition to increased loss would be expected [23]. Additionally, mode solving simulations verify that the effective indices of the fundamental modes of these high loss widths are above that of any dielectric layers bordering the polysilicon core such that the waveguide is not close to cutoff. There is, however, a thin, high-index silicon carbide layer that is part of the backend electronic stack-up as an etch stop layer as shown in **Figure 4.22**. The effective indices of the slab waveguide modes of this layer fall in between the effective indices of the low loss and high loss waveguide widths for each wavelength as shown in **Figure 4.23**. Although the SiC layer is ~2 μm away from the waveguide core, mode solving simulations show significant electric field overlap with the SiC layer, shown in **Figure 4.22**, at the widths where the sudden loss increase is observed. Therefore phase-matched coupling between the dielectric etch stop layers and the waveguide mode may set the limit to confinement factor scaling of waveguide loss when integrated in similar electronic platforms.

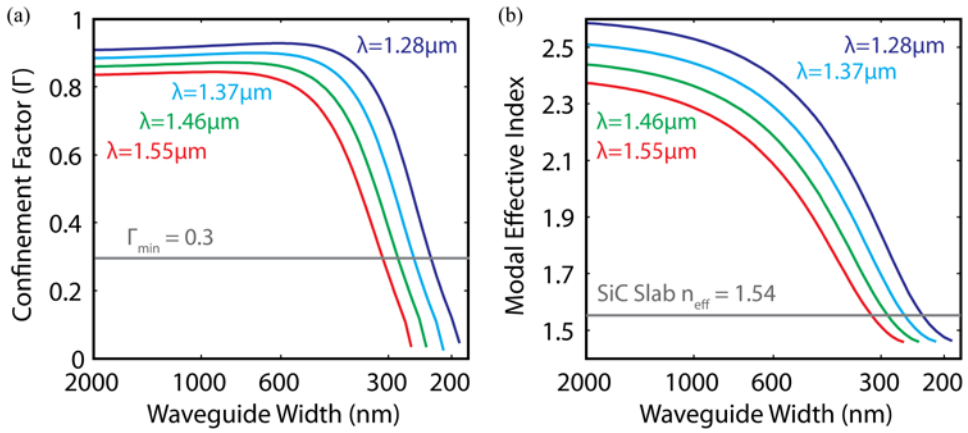


**Figure 4.23** Simulated waveguide mode (a) confinement factor and (b) effective index. Confinement factor curves from (a) are multiplied by the extracted bulk loss to generate the fit curves shown in **Figure 4.21**. Effective index curves are overlaid with the simulated 1D slab mode index for the SiC layer that is correlated with the observed loss increase across measured wavelengths.

A final trend visible from the data shown in **Figure 4.21** is a consistent increase in waveguide propagation loss at shorter wavelengths for wide waveguides. Since the majority of the optical power is guided in the polysilicon core region at these widths, this increase can be attributed to the bulk material loss scaling with wavelength. The approximate bulk material loss of the polysilicon can then be extracted by dividing the measured propagation loss of the wide waveguides by the simulated confinement factors. This data is then shown in **Figure 4.25** for a larger number of wavelengths by dividing the measured propagation loss by the simulated confinement factors for 1.5 μm and 2.0 μm waveguide widths shown in **Figure 4.23**. This same analysis was also performed for wafers prepared identically to the data presented in **Figure 4.21** with the exception that the thermal cycling associated with the full electronics process was not performed. Comparing the two data sets, a 32% increase in the extracted material loss is observed as shown in **Figure 4.25** with a standard deviation of 6% across the wavelength range studied. Importantly, this increase is far below the order of magnitude increase that has been observed in previous work [11,22].

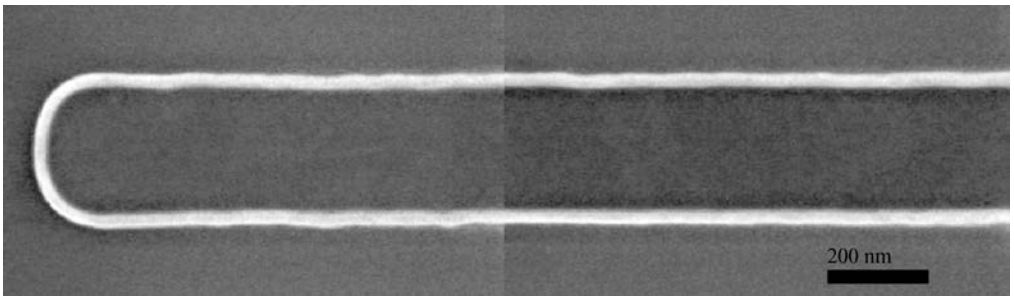


**Figure 4.24** High-resolution scanning electron micrograph (SEM) of a 275 nm wide isolated polysilicon line that was used for line edge roughness extraction. An estimated roughness of 3 nm RMS with a 50 nm correlation length was obtained from this analysis.

In this analysis, the scattering loss sources were assumed to be negligible in extracting the bulk material loss. The confinement factor scaling trends observed for all wavelengths suggest that this assumption is justified. The sidewall roughness of the fabricated waveguides was estimated from the scanning electron micrograph (SEM) shown in **Figure 4.24** to be approximately 3 nm RMS with a 50 nm correlation length. This roughness is expected to be reduced by using more mature photolithography recipes than the one used for this test run due to a non-standard choice of mask polarity. Still, the current roughness of the fabricated waveguides is expected to result in less than 2 dB/cm for high confinement width, single-mode waveguides at 1550 nm [23]. The sidewall roughness contributions for the widest waveguides that are used for the loss extraction should be further minimized by the low mode overlap at the waveguide edges as confirmed by the high correlation of extracted losses from the 1.5 μm and 2.0 μm test section widths. However, although the top surface roughness of the polysilicon was measured to be 0.3 nm RMS by AFM, the large electric field overlap with top surface still enables scattering loss from this surface to impact the net waveguide propagation loss. The impact of the top surface can be observed by comparing the 120 nm thickness samples presented above to a sample prepared with a 200 nm thick polysilicon layer. Comparing the propagation loss at a 2.0 μm width, the 200 nm thick waveguides are observed to have a smaller propagation loss than would be predicted by confinement factor scaling as shown in Table 4.2. The measured loss difference, however, is approximately 10% of the total loss. Although scattering from the sidewall and top surfaces do have a non-trivial impact on waveguide loss, the dominant loss mechanism is seen to be the bulk waveguide loss that was extracted for the defect absorption analysis. Deviations from bulk loss scaling follow scattering based trends

evident by the electric field overlaps with relevant surfaces. For example, since the extracted material loss was obtained from the wide widths, the bulk loss extraction underestimates the loss of the 550 nm wide waveguide widths where the sidewall electric field overlap is an order of magnitude higher.

| | | 120nm Thick, End-of-Line Polysilicon | | | 120nm Thick, As-Crystalized Polysilicon | 200nm Thick, End-of-Line Polysilicon |
|---|---|---|---|---|---|---|
| Material Loss Only | Design Width (μm) | 0.35 | 0.55 | 2.00 | 2.00 | 2.00 |
| | Extracted Material Loss ($cm^{-1}$) | 4.9 | | | 3.7 | 4.9 |
| | Simulated Confinement Factor (%) | 36.4 | 79.5 | 83.1 | 83.1 | 98.7 |
| | **Predicted Bulk Loss (dB/cm)** | **7.7** | **16.9** | **17.7** | **13.3** | **21.0** |
| Including Scattering | Sidewall E-Field Overlap, ±5nm (%) | 0.42 | 0.63 | 0.075 | 0.075 | 0.096 |
| | Top Surface E-Field Overlap, ±5nm (%) | 0.51 | 1.5 | 2.6 | 2.6 | 2.0 |
| | **Measured Loss (dB/cm)** | **6.2** | **19.4** | **17.6** | **13.6** | **18.1** |

**Table 4.2** Summary of experimentally measured waveguide losses, bulk absorption calculations and electric field overlaps with surfaces where roughness is a concern. Surface overlaps were calculated by integrating the electric field within 5 nm of each surface. All data presented is for $\lambda = 1550$ nm.

#### *4.4.3.3. Loss Source Analysis for Post-Deposition-Crystallized Waveguides*

To gain some insight into the source of the waveguide loss, the extracted wavelength-dependent bulk losses were compared to theoretical predictions. The material loss was calculated by accounting for electronic transitions between mid-gap states localized at grain boundaries and Bloch states in the conduction and valence bands. Assuming that the oscillator strengths for all involved transitions between band and midgap states average to a constant, the absorption coefficient calculated from Fermi's Golden Rule can be written as a function of the valence, conduction and midgap densities of states (DOS):

$$\alpha(\omega) = A\left\{\begin{array}{l}\int \rho_v(E_v)\,\rho_g(E_v+\hbar\omega)\,f(E_v)\left[1-f(E_v+\hbar\omega)\right]dE_v \ \ldots \\ \quad +\int \rho_g(E_g)\,\rho_c(E_g+\hbar\omega)\,f(E_g)\left[1-f(E_g+\hbar\omega)\right]dE_g\end{array}\right\}$$

The two integrals sum transitions from the valence band states (DOS $\rho_v$) to the midgap states (DOS $\rho_g$), and from the midgap states to the conduction band (DOS $\rho_c$), as diagrammed in **Figure 4.25**. The Fermi distribution function, $f$, enforces

that all considered transitions occur between filled initial and empty final states, separated by the photon energy. Unlike the valence and conduction band density of states, the midgap state density parameter represents only a functional form of the midgap state energy distribution, which is chosen to be Gaussian. The prefactor $A$ is then the product of the total density of midgap states and the defect-to-band transition oscillator strength to provide the sole fitting parameter in the calculation. This frequency-independent parameter serves to scale only the magnitude of the curve and not its shape.

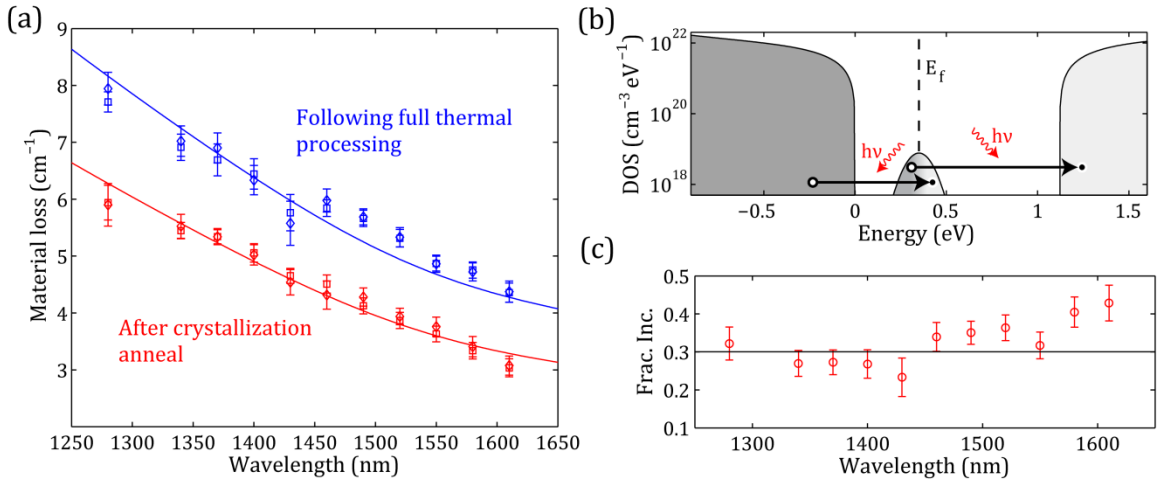


**Figure 4.25** (a) Propagation loss as a function of wavelength and waveguide width for the thermally processed wafer with data from 2.0 μm wide waveguides plotted as squares and 1.5 μm waveguides as diamonds, along with calculated fits for points both before and after the full thermal processing. (b) Polysilicon density of states (DOS), plotted as a function of energy relative to the valence band-edge, used for the fit calculations - 23% difference in the height of the peak around 0.35 eV produced the difference between the two fits in (a). The pinned Fermi level and example optical transitions are indicated. (c) Fractional increase in material loss as a function of wavelength following full thermal processing.

The conduction and valence band DOS used in the calculation were taken from IBM's DAMOCLES calculations [25]. The mid-gap density of states used was matched to that deduced by Jackson et al. [24], who inferred that these localized gap states manifest themselves in a broad peak ~0.35 eV above the valence band edge, as shown in **Figure 4.25**. They also deduced a strong valence band-tail DOS, which is observed not to play a strong role in our material and so not included in the calculation; since our material differs significantly from theirs (the observed loss is two orders of magnitude lower), the relative concentration of tail and mid-gap

localized states are expected to be different, so appropriately including the band tailing would require the addition of at least one additional fitting parameter. The Fermi-level at the grain boundary edges was assumed to be pinned at the energy of maximum midgap state density.

The results of the fit, calculated as described above, to data taken in samples removed from the flow after the crystallization anneal, as well as after the full thermal processing, are shown in **Figure 4.25**. The change in state density required to fit the measured absorption of the end-of-line and as-crystallized polysilicon can then be attributed to hydrogen out-gassing reducing the passivation-fraction of dangling bond states at the grain boundaries. The ~32% change in material loss is shown in reference to the measured data points in **Figure 4.25**. This provides technological feedback for further waveguide loss reduction. For example, introduction of a forming gas anneal after the completion of front-end fabrication may effectively passivate the dangling bonds. At this point of the process, the maximum processing temperatures would be reduced below 500 ºC due to the metal interconnect. As a result, effective passivation of dangling bonds, which have been shown to be the dominant loss source, may be introduced into the process without further hydrogen outgassing.

## 4.5. Vertical Grating Couplers

As discussed in previous sections, second-order gratings are used as vertical couplers for optical access [122-124]. This section will describe the theory, design and measured results for these couplers in bulk-CMOS, SOI-CMOS and DRAM processes. The couplers described here differ from most published reports in that the grating is formed in the waveguide core layer by etching the complete thickness of the polysilicon layer. This peculiarity has primarily been the result of processing constraints. However, since the layer thicknesses used for our work have typically been thin relative to most silicon photonic results, the completely etched gratings have yielded sufficient performance for process development. Both the SOI-CMOS process and integrated DRAM process flow that is under development have sufficient flexibility to pattern not only partially-etched gratings, but also three-level gratings. The additional dimensional flexibility is expected to yield significant performance gains as have been demonstrated by other workers in the field.

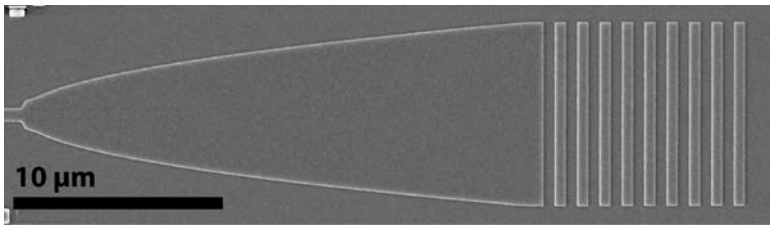


**Figure 4.26** Scanning electron micrograph of a vertical coupler from EOS2. Designed spot size was 5 µm for 15° off normal operation.

For most test chips, Milos Popovic has designed the grating couplers for all the integrated photonic devices. An example design that was fabricated in a 28nm bulk-CMOS process (EOS2) is shown in Figure 4.26. Similar devices have been designed for both 1260nm and 1550nm wavelength centers for the CMOS test chips EOS1, EOS2 and EOS6-12. As a point of reference, we have measured a minimum insertion loss of 4.8 dB with a 93 nm 1 dB bandwidth for the 1550nm EOS2 coupler as shown in Figure 4.27. Since the optical design does not break the symmetry between the upward and downward directions, the maximum theoretical coupling efficiency is 3dB.

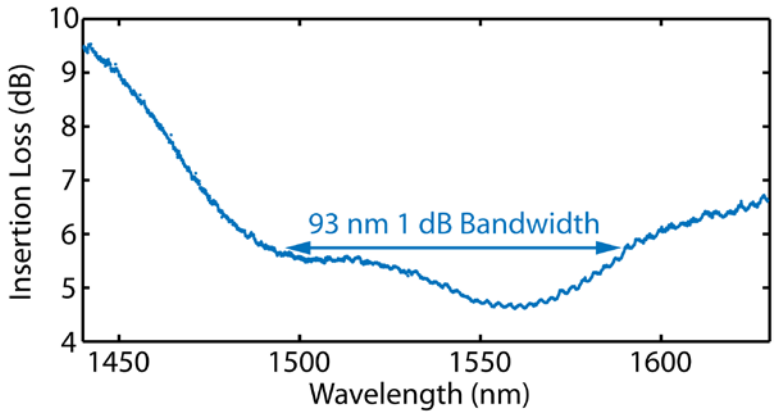


**Figure 4.27** Measured insertion loss (a) the vertical grating couplers (SEM shown in inset with 10 µm scale bar). Insertion loss calculated as half the fiber-to-fiber loss through 2 couplers connected by a 500 µm waveguide. Lensed fiber with 5 µm spot size positioned 15.5° from normal was used for these measurements.

Since the vertical grating couplers provide the only form of optical access to the integrated photonic structures, the coupler design has been optimized for fabrication tolerance. For the initial DRAM integration test mask, I designed the set of vertical grating couplers used to test the integrated photonic device performance from 1100nm to 1650nm. The following discussion will present the design

methodology for broadband performance with high fabrication variance tolerance used for such integrated grating couplers.

Given sufficient computational resources, it may be possible to simulate the coupling of the grating to the fiber by using 3D-FDTD. Since the computational domain would have to be on the order of tens of microns in all three dimensions, however, most research groups, including ours, do not have sufficient computational resources. Instead, the product of the coupling efficiency calculated by orthogonal 2D simulations is used to estimate the overall coupling efficiency: one in the transverse plane, that is the xy-plane, and the other orthogonal to the electric field polarization, that is the yz-plane. This approximation is valid to the limit that the electric field in the combined plane can be decomposed into independent components:

$$E(x,z) = E(x) \cdot E(z) \tag{4.2}$$

Due to the large mode size and minimal longitudinal component to the electric field, this approximation has been shown to be valid and improved techniques to combine the independent simulations have been developed [226]. Since this assumption, which is equivalent to the effective index, is only used to approximate the large area mode that couples to the fiber, it is significantly more valid than for the strongly-confined mode of the integrated silicon waveguides as discussed in Chapter 1. By comparing similar results that were achieved using the effective index method to full 3D simulations, other workers in the field have estimated the technique to be 5% accurate in the near field [226]. The accuracy reduces further as the fiber spacing is increased and reduces in accuracy another 1 dB with a 50 μm fiber gap [226].

The transverse plane coupling efficiency is calculated by the mode matching of the x-component tapered waveguide mode and a Gaussian with the mode field radius (MFR) equivalent to an optical fiber mode. The efficiency is then calculated by the overlap integral of the electric fields:

$$\eta_{transverse}(MFR) = \frac{\int \left| \frac{1}{MFR} e^{-\left(x/MFR\right)^2} \cdot E^*_{mode}(x) \right|^2 dx}{\int \left| E_{mode}(x) \right|^2 dx} \tag{4.3}$$

The waveguide mode electric field calculations were performed in Fimmwave and are shown in **Figure 4.28**. The resulting overlap integral calculations are shown in **Figure 4.29**.

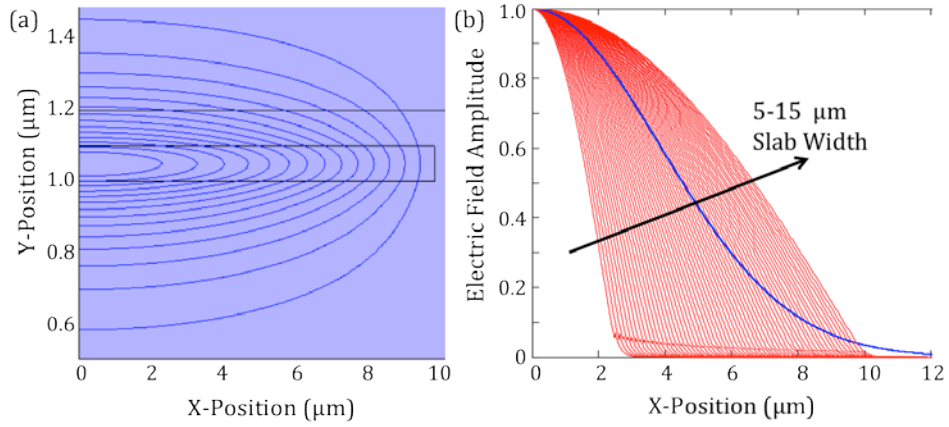


**Figure 4.28** (a) Electric field contours for simulation of a tapered waveguide mode as is used in the coupling region. (b) Electric field amplitude as a function of X-position for the center of the waveguide mode compared to the Gaussian fiber field for a 10 μm mode size.

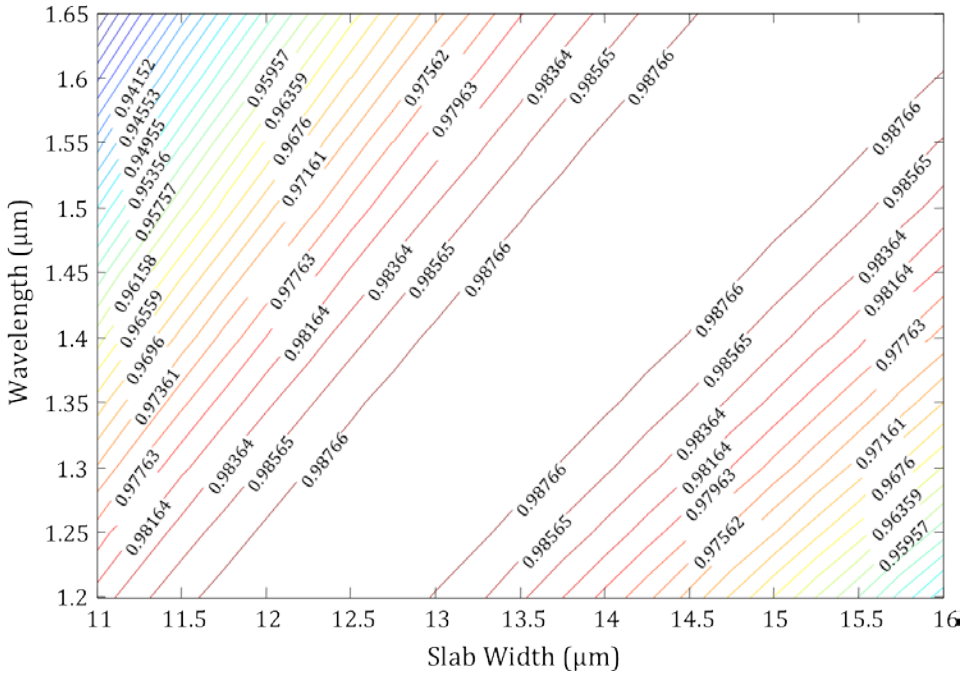


**Figure 4.29** Transverse mode electric field overlap ($\eta_{transverse}$) contour plot as a function of slab waveguide width and wavelength.

The coupling efficiency normal to the electric field polarization is then calculated using 2D-FEFD. Although 2D-FDTD would yield suitable results, the FEFD feature in Omnisim was used for the grating design calculations in this work for faster calculations. Since modal transmission is rigorously reciprocal in the absence of

magnetic materials, the coupling is only calculated as sourced from the waveguide mode and coupled into the optical fiber. The computational domain and simulated field emission profile are then shown in **Figure 4.30** and **Figure 4.31** respectively. The reflection is calculated by the standing wave ratio of the electric field in the input waveguide. The fiber coupling efficiency is measured by first calculating the radiation efficiency of the grating as the power flux ratio of the input power sensor to the fiber plane sensor. The mode overlap must then be calculated from the electric field profile. For the overlap calculation on the fixed plane of the fiber plane sensor, the fiber mode must be appropriately scaled and phased to allow calculation of any arbitrary fiber angle. As a function of center position $z_0$, angle relative to normal incidence $\varphi$ and fiber mode field radius, the optical fiber mode can then be written as:

$$E_{fiber}(z_0, \varphi, MFR) = \exp\left[-\left(\frac{(z - z_0)\cos\varphi}{MFR}\right)^2 - ik(z - z_0)\sin\varphi\right] \qquad (4.4)$$

The coupling efficiency for a given coupling angle and mode field radius is then calculated by the following mode overlap:

$$\eta_l(\varphi, MFR) = \max_{z_0} \frac{\int \left| E_{fiber}(z_0, \varphi, MFR) \cdot E^*_{sensor}(z) \right|^2 dz}{\int \left| E_{sensor}(z) \right|^2 dz \int \left| E_{fiber}(z_0, \varphi, MFR) \right|^2 dz} \qquad (4.5)$$

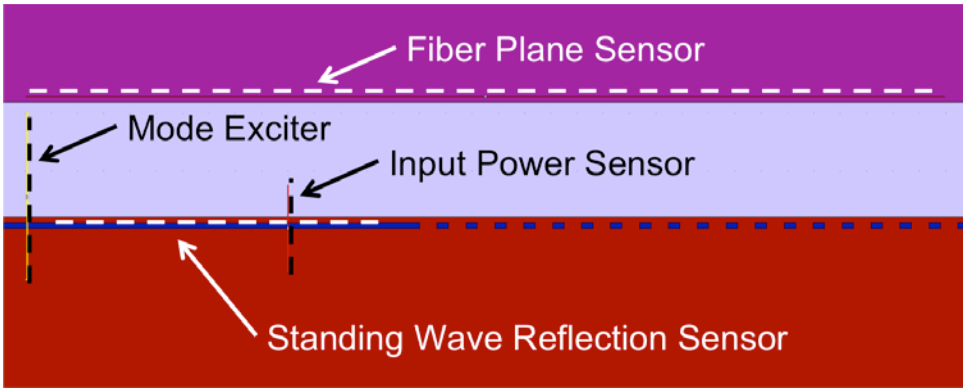


**Figure 4.30** Computational domain for grating simulation.

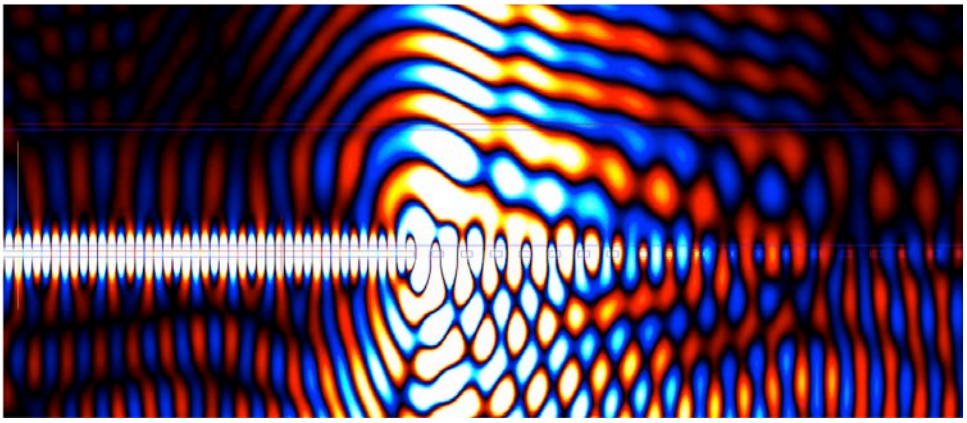

**Figure 4.31** Simulated electric field cross section.

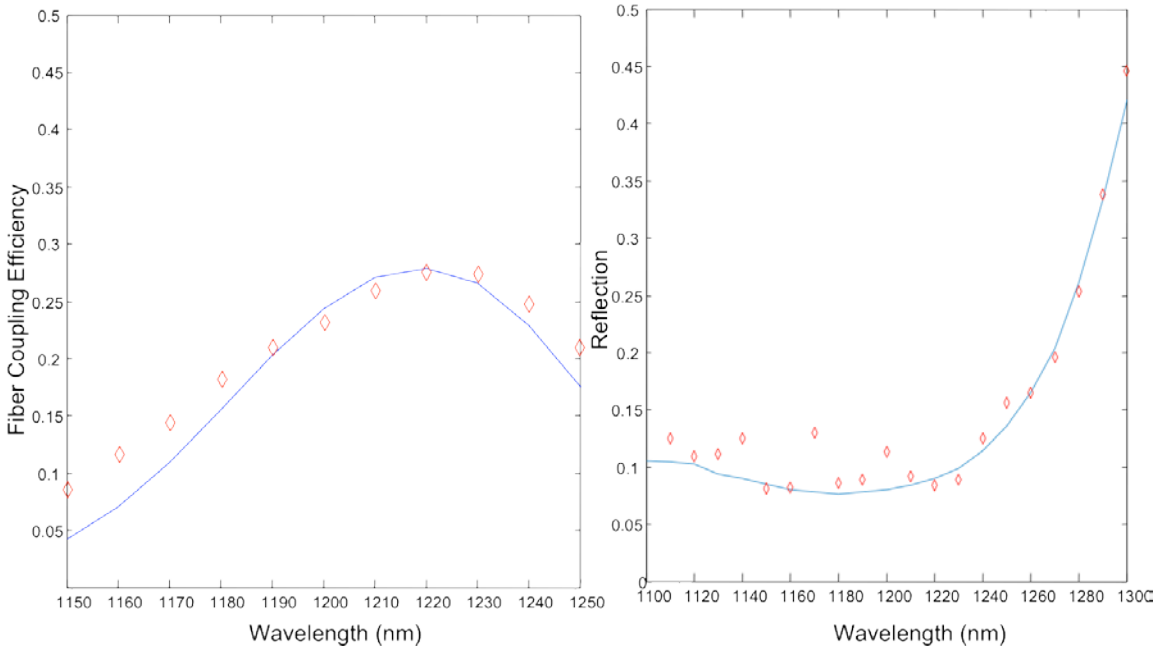


**Figure 4.32** Benchmarking simulations (solid lines) comparing the developed simulation framework with the simulations of Milos Popovic (open diamonds).

Before designing the grating couplers for fabrication, the simulation framework was benchmarked against the simulations used for previous CMOS chips. Since good model-to-hardware correlation had been observed for the simulations performed by Milos Popovic, I resimulated his grating designs. The two simulations are compared in **Figure 4.32**.

A large set of parameterized simulations were then used to determine grating designs that would yield acceptable performance over a wide range of wavelengths and fabrication conditions. For the 1550nm design, the contour plots for the

independent contributions of radiation efficiency and fiber mode matching for a 10.5 μm mode field diameter at an 8° off-normal incidence angle are shown in **Figure 4.33** with the resulting net coupling product shown in **Figure 4.34**. Since any reflections from the grating couplers create a Fabry-Perot cavity with interference fringes, minimization of coupler reflection is an important design characteristic to enable flat measured transmission characteristics. The simulated reflection characteristics over the same parameter range are shown in **Figure 4.35**.

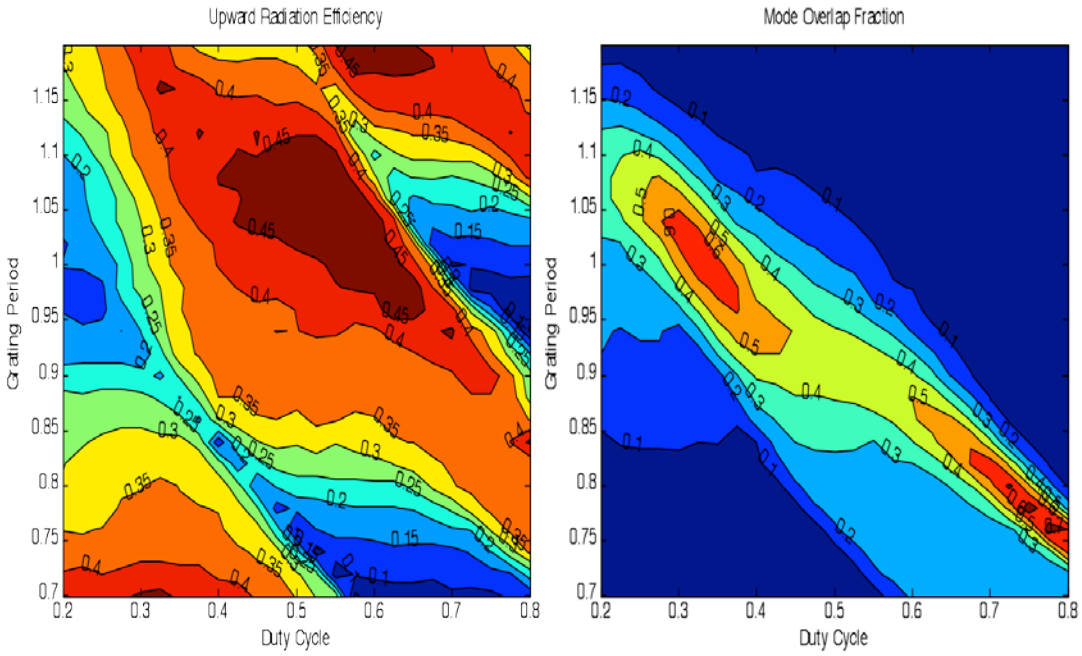


**Figure 4.33** Independent efficiency contours for the radiation efficiency and mode mismatching for the 1550nm grating designs.

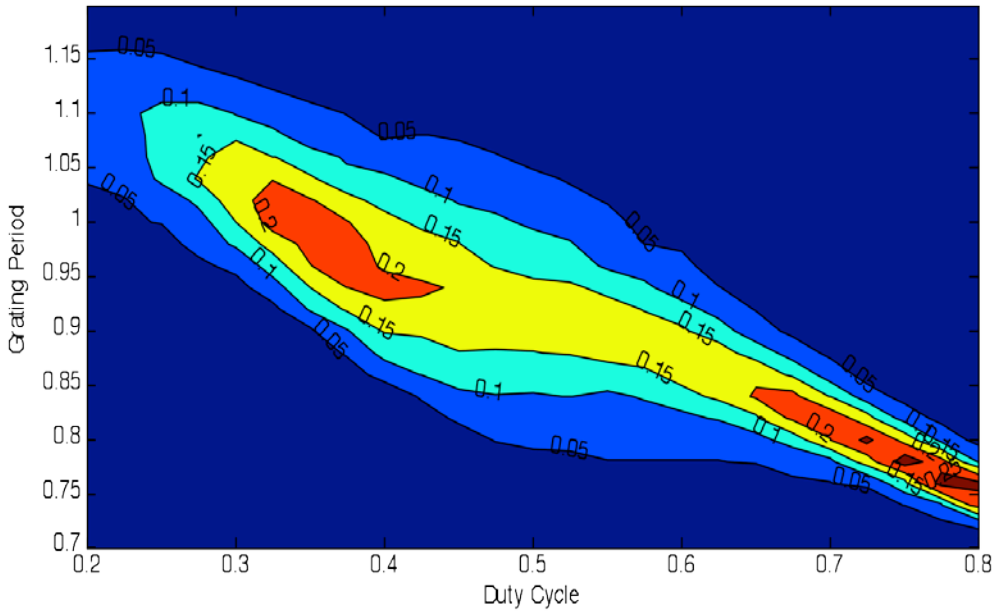


**Figure 4.34** Fiber coupling efficiency at an 8 degree fiber angle for the 1550nm design.

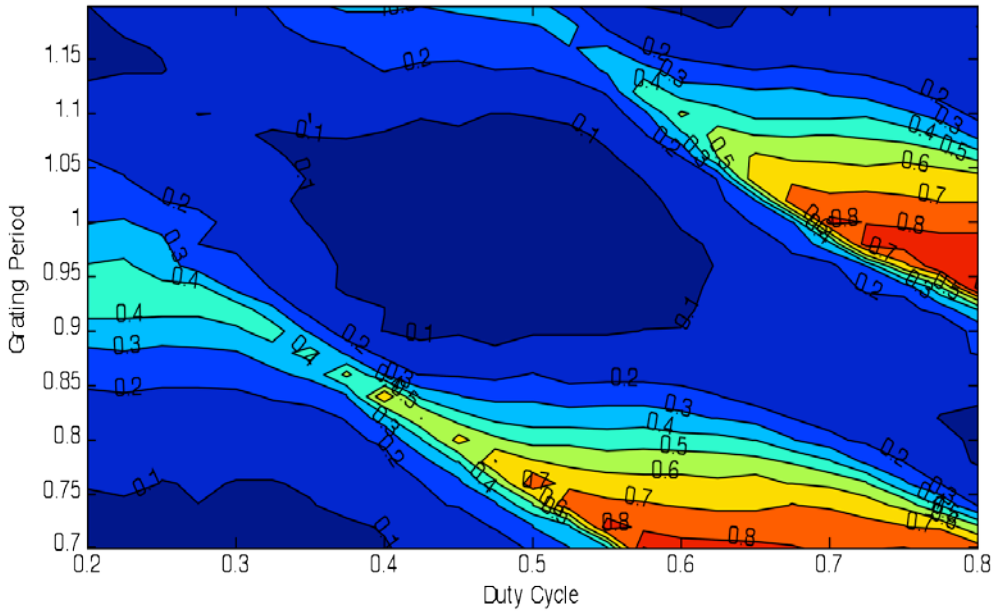


**Figure 4.35** Simulated reflected power into the input waveguide for various design dimensions at 1550nm.

Since the silicon and oxide regions that compose the grating have very different refractive indices, both the duty cycle and grating period change the phase relation between subsequent grating teeth. The phase difference between the subsequent grating teeth, which determines the output emission angle as discussed in Chapter

1, is therefore a function of the grating duty cycle and period. For further design optimization, grating designs along the lines of constant phase are selected by varying the grating period as a function of duty cycle. Each grating design is then simulated as a function of wavelength to form a new set of design contour plots such as the one shown in **Figure 4.36**. The broadband, fabrication tolerant design is then identified as the 42.5% duty cycle which corresponds to a 950nm period. Similar optimization was performed for 1400nm and 1250nm wavelength centers. The overlap of acceptable performance – below 10% reflection and above 10% coupling efficiency – for the 1400nm and 1550nm designs is shown in **Figure 4.37**.

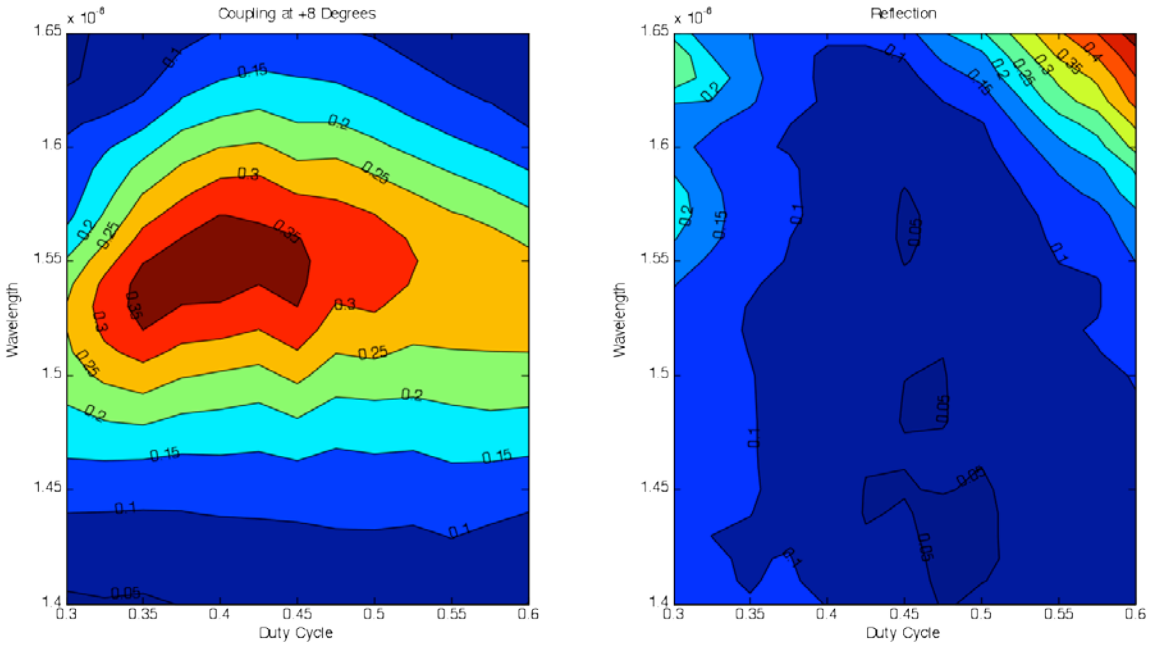


**Figure 4.36** Wavelength and duty cycle contour plots for a constant phase grating design exploration.

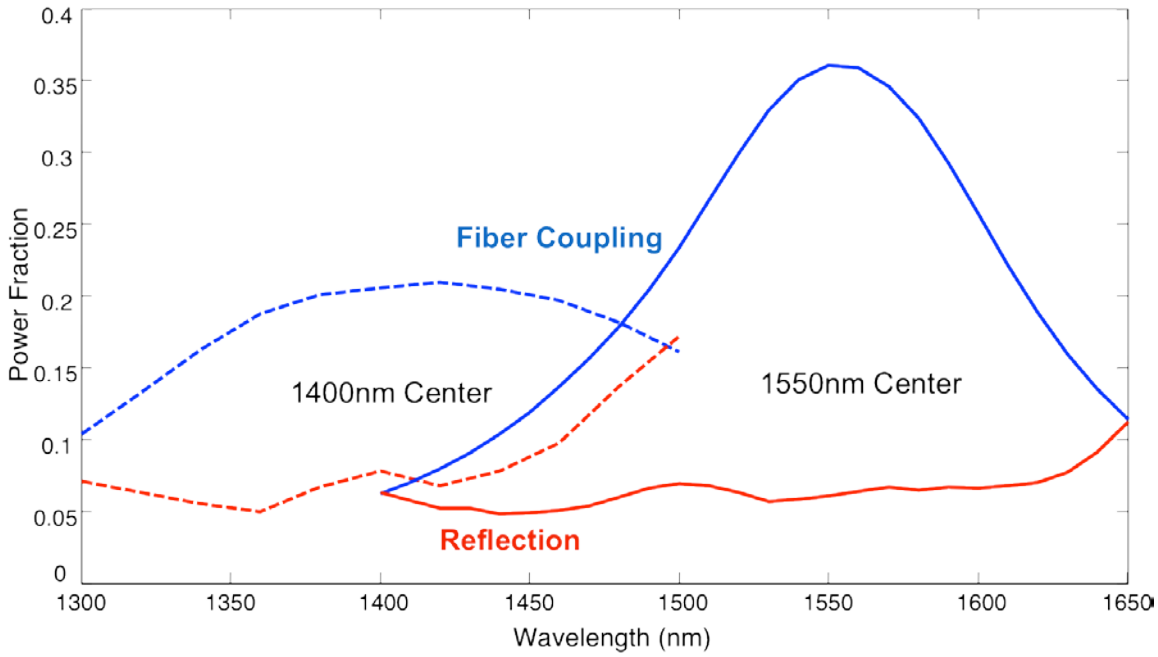


**Figure 4.37** Performance plots for the 1550nm and 1400nm grating designs.

The fabricated grating couplers were then measured with the 1550nm design shown in **Figure 4.38**. The measured characteristics are slightly below the simulated performance. Similar differences between simulated and measured performance has been observed by other workers in the field. The discrepancy may be generally attributed to the limits of validity of the independent 2D calculation or near field calculation of the mode overlap coupling. Instead of the fiber placed within a few microns of the grating, the 8° coupling angle forces the fiber core to be tens of microns away from the grating due to the 125 μm cladding diameter. Any diffraction effects that would alter the emitted grating mode are not included in the simulated coupling efficiency.

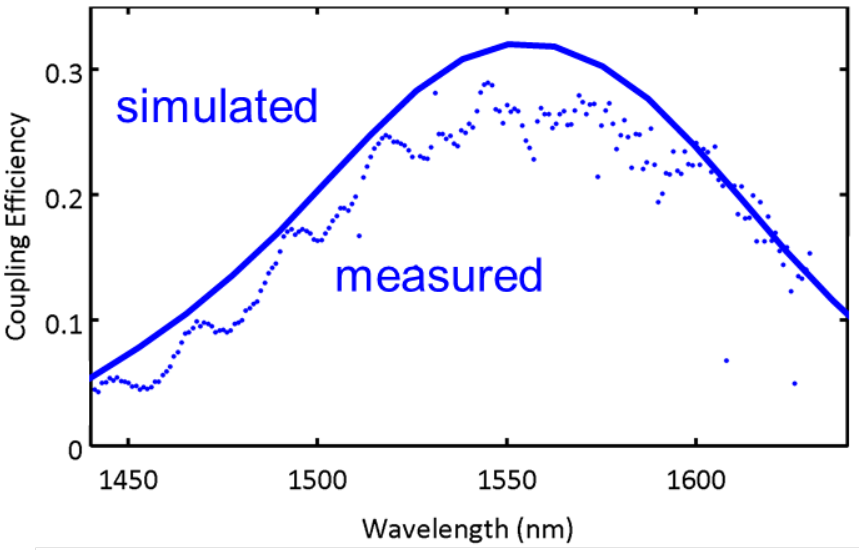


**Figure 4.38** Measured and simulated grating coupling efficiency for the 1550nm design.

Similar to the single polarization operation of all of the presented waveguides and integrated photonic devices, the grating couplers presented so far operate for a single fiber mode polarization. Since the grating redirects the TE waveguide mode vertically, the fiber mode must be polarized in the long dimension of the grating teeth. However, the change in propagation dimension results in a simple, 90° in-plane rotation for vertical grating couplers to address the TE or TM mode of the optical fiber under normal incidence conditions. Superposition principles therefore enable a hybrid grating coupler to be created to split the output of the TE and TM fiber modes to separate TE waveguide mode outputs 90° apart. Due to the difficulties of 3D simulation, a 2D simulation approach similar to the single polarization gratings were used for designing the polarization splitting gratings. In both cases of the 2D gratings, the indices for each region in the FEFD simulation was determined by the slab index for out of plane waveguide mode. Since the overlayed grating forms a checkerboard, the duty cycle of the simulated grating affects the index of the etched region.

After the nominal design was simulated to allow efficient normal emission, an array of polarization splitting gratings was fabricated with an example shown in **Figure 4.39**. The fabrication uncertainty of the grating due to the degree of corner rounding prohibited a single design to be fully simulated. The performance of various gratings were first examined by the infrared camera to establish the correct grating period for normal emission in the fabricated dimensions. The grating where

the output spot was not observed to shift as the microscope focus was adjusted was determined to have a normal output angle. Next, the output waveguides for the TE and TM port waveguides were measured as a function of polarization and wavelength as shown in **Figure 4.40**. Greater than 15 dB extinction is observed between the two polarizations at each output port.

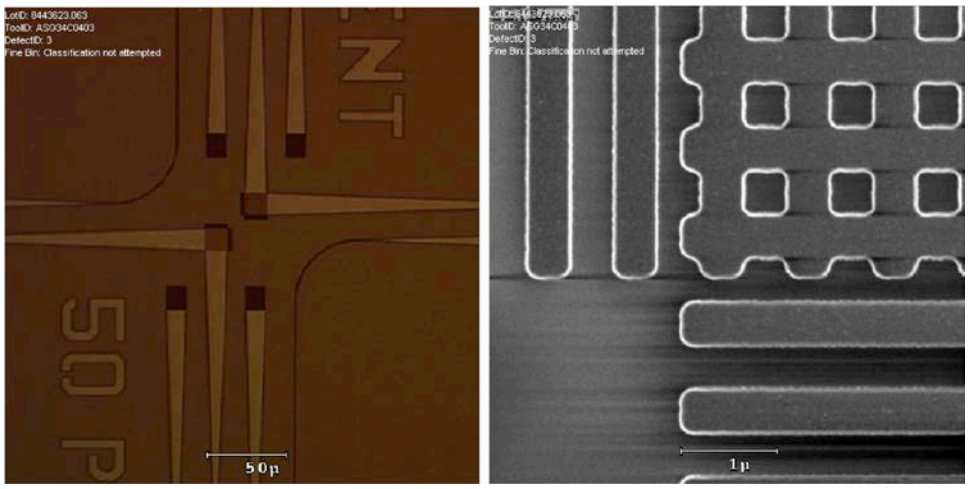


**Figure 4.39** Optical micrograph and scanning electron micrograph of the fabricated polarization splitting grating couplers fabricated in the same memory integration platform used for the single polarization gratings.

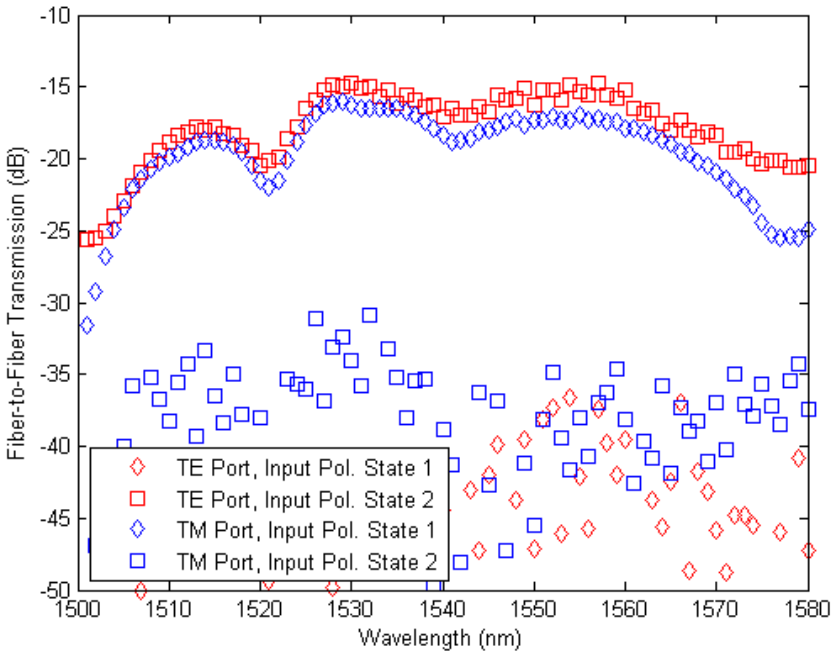


**Figure 4.40** Measured polarization selectivity between the two output ports of the polarization splitting grating for TE and TM input polarizations.

## 4.6. Wavelength Division Multiplexing Filters

### 4.6.1. First-Order Ring Resonator Filters

The high resolution photolithography used in such deeply-scaled electronics processes enables the precise lateral dimension control that is critical to a wide variety of photonic devices[9,17-19]. An important example of such a device is the ring resonator filter bank that creates the bandwidth-density advantage of silicon-photonics for interconnect applications by enabling DWDM[22,23]. In the design of the filter banks presented, the target resonant frequency of each channel is not specified to an absolute value due to the ambiguity of the exact fabricated silicon layer thickness and ambient temperature during operation. Instead, the eight channels are designed to be evenly distributed across the free spectral range (FSR) of a single filter. The fabrication precision therefore allows the measured 8-channel filter bank, shown in **Figure 4.41**, to be composed of evenly distributed channels by stepping the ring radius by 12nm. The as-fabricated transmission characteristics are shown in **Figure 4.42**.

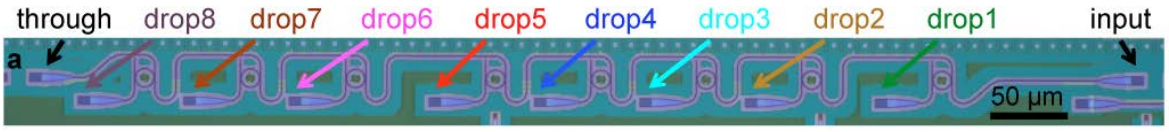


**Figure 4.41** Micrograph of 8-channel filter bank implemented with 7.0µm nominal radius first-order ring-resonator filters fabricated in the single-crystalline silicon layer of the thin-SOI process on the EOS8 test chip. Design dimensions: 470nm ring and bus waveguide widths, 217nm through and drop ring-bus gaps.

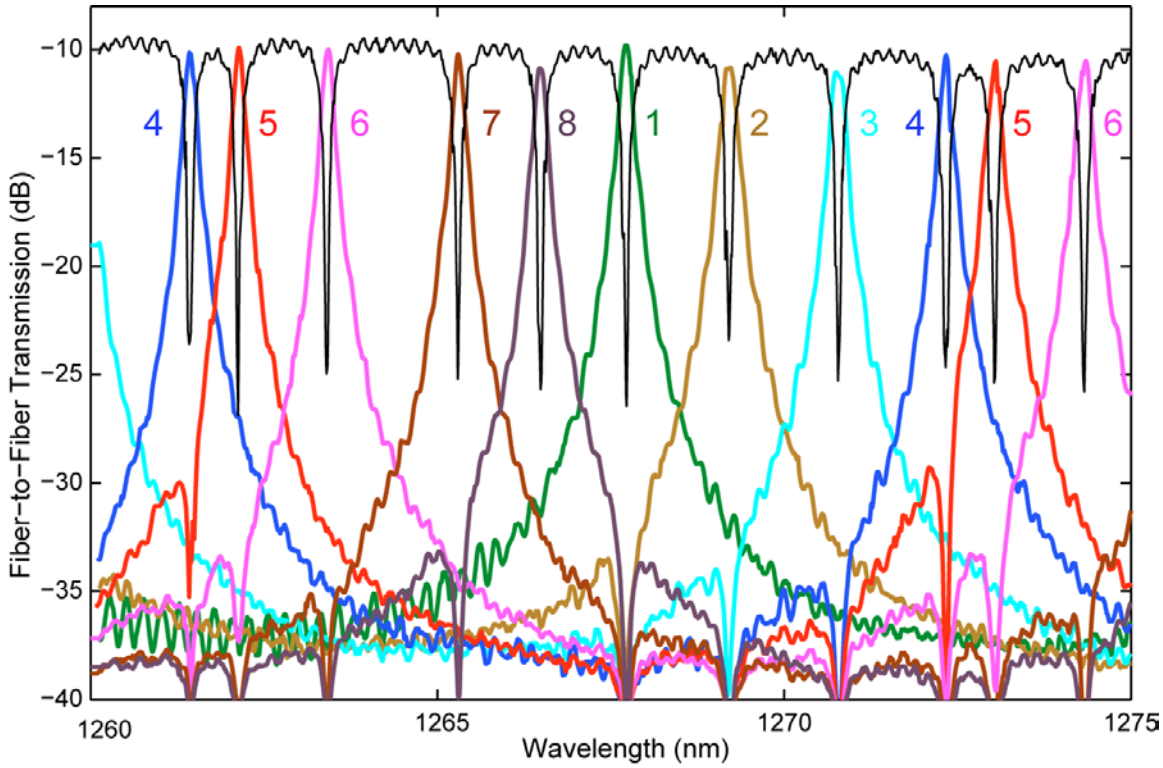


**Figure 4.42** Filterbank transfer function for all ports as measured for fiber-to-fiber transmission. The measured 10dB fiber-to-fiber loss is dominated by expected theoretical loss of the simple vertical grating coupler designs used in this initial work. Port labeling is consisted with the values shown in **Figure 4.41**.

To verify the transmission matrix formalism developed in the introduction and used to predict the possible WDM channel count in Chapter 2, the as-fabricated transmission characteristics were compared to model predictions. By varying only the radius of the rings in the filter bank for a single fiber-to-fiber insertion loss, the model predictions were compared to the measurements as shown in **Figure 4.43**.

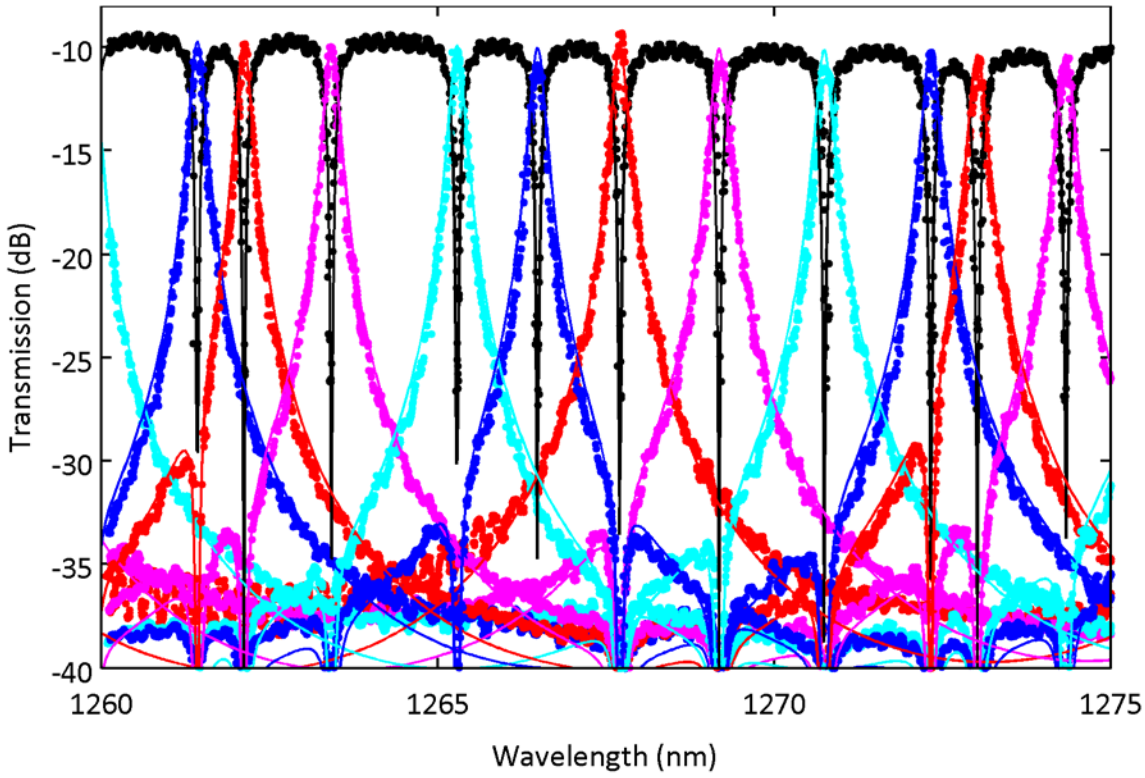


**Figure 4.43** Measured filterbank transfer function fit to the transfer matrix simulations using a single fiber-to-fiber insertion loss and resonator coupling coefficient with nonuniform channel spacing.

### 4.6.2. Second-Order Ring Resonator Filters

Second-order microring filters enable denser channel packing by having sharper pass-band to stop-band slopes analogous to higher-order electronic filters. We fabricated the four channel second order ring resonator filter bank shown in **Figure 4.44** in the 28nm bulk-CMOS process on the EOS2 test chip. Similar filter banks have been demonstrated in a variety of material platforms to enable complex wavelength routing [128, 130, 227, 228]. The transmission functions for all ports in this filter bank were measured without any thermal tuning or post-fabrication trimming, as shown in **Figure 4.45**. The advanced process lithography produces untuned drop port insertion losses below 5 dB and less than 15 dB cross talk between filter channels separated by less than 200 GHz for the first time in a high index contrast platform.

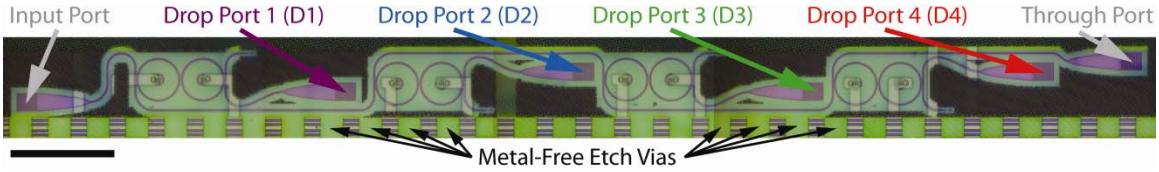


**Figure 4.44** Micrograph of 4 channel second-order filterbank. Filterbank organization minimizes required undercut distance from the etch vias indicated at the bottom of the figure. Design filterbank dimensions: 10 μm ring radius, 670 nm waveguide width, 80 nm polysilicon thickness, 7.63 x $10^{-2}$ bus-ring power coupling coefficient, and 3.01 x $10^{-3}$ ring-ring power coupling coefficient. Scale bar of 50 μm shown at bottom left.

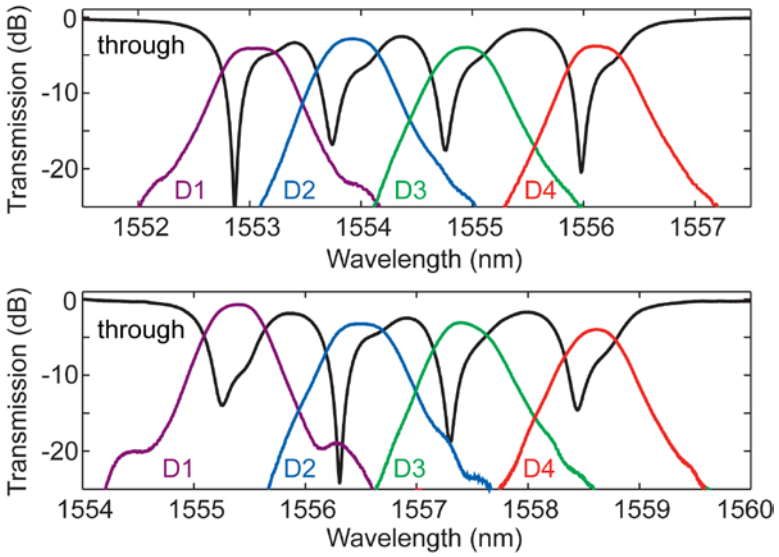


**Figure 4.45** (a) Measured transmission functions, scaled to the through port transmission away from resonance, for a 4-channel second-order filterbank. No thermal tuning or post-fabrication trimming was performed for these measurements. Port line colors and naming convention correspond to labels in **Figure 4.44** micrograph.

The fabrication precision of the unmodified CMOS process can then be quantified by comparing several copies of this filter bank from across the wafer to simulated models and original designs. The mean channel frequency spacing within a filter bank, histogram shown in **Figure 4.46**, of 137 GHz is within 15% of the designed value of 120 GHz. Over the short length of a single filter, the polysilicon thickness is approximately constant and the average lithographic line width control for nanophotonic applications can be measured from the frequency matching of the two rings. The mean frequency mismatch – extracted by simultaneously fitting the filter through and drop responses – is 30.9 GHz as shown in **Figure 4.47** and **Figure 4.48**.

With a simulated resonant frequency dependence on average resonator line width of 38 GHz/nm, this corresponds to an average lithographic matching of 810 pm, less than seven times the stochastic process variation limit of 120 pm previously achieved using intra-field distortion calibrated scanning electron beam lithography [229, 230]. This accuracy, unprecedented for projection lithography fabricated nanophotonics, allows for high device yield and therefore enables large scale integration.

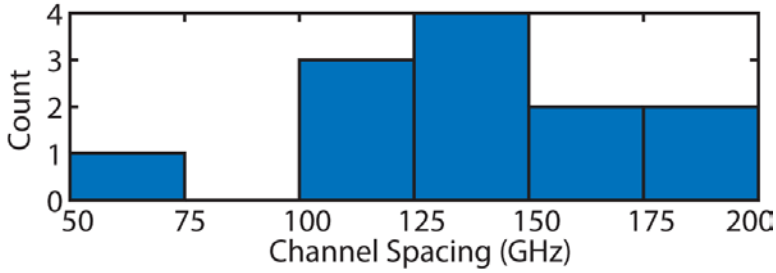


**Figure 4.46** Histogram of channel frequency spacing, calculated by drop port transmission maxima, measured from four filterbanks from across one wafer. Designed channel spacing is 120 GHz.

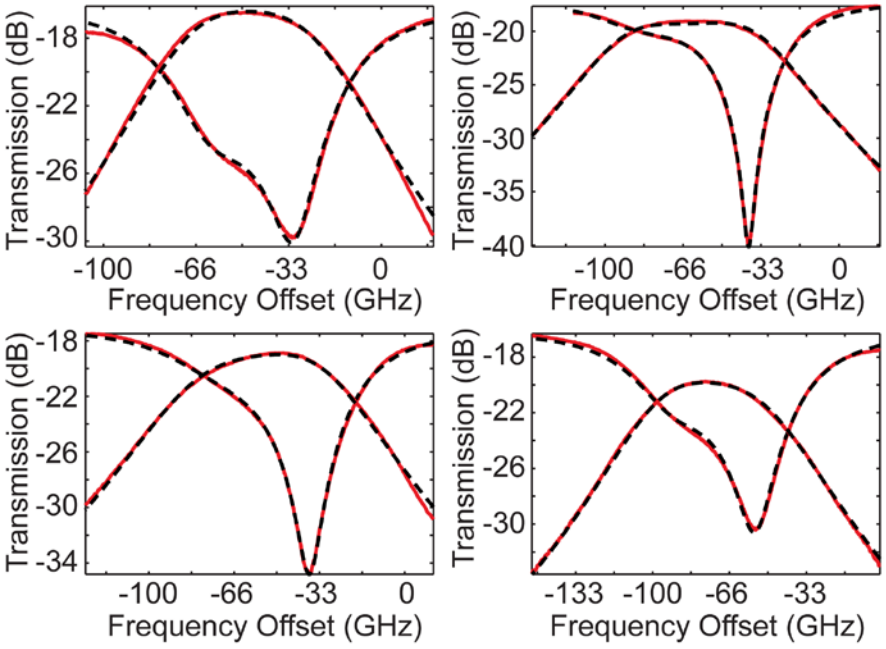


**Figure 4.47** To extract the resonant frequency mismatch for the two rings in the second-order filters, measured through and drop transmission functions were fit to ideal filter model with the following free parameters: bus-ring coupling coefficients, ring-ring coupling coefficients, ring round trip loss, and separate resonant frequencies for each ring. Resulting model fit (dotted black) lines for through and drop responses overlayed with measured transmission (solid red) lines for two example filters.

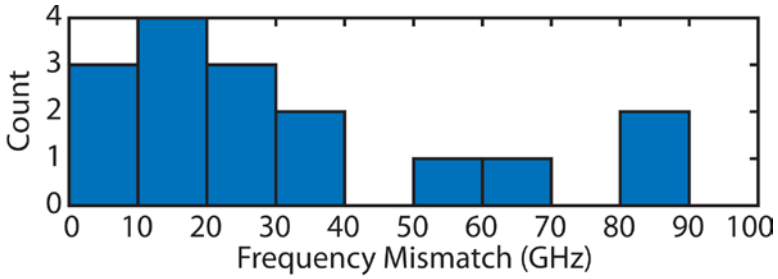


**Figure 4.48** Histogram of resonant frequency mismatch between the two rings in the second-order filters for the same four filterbanks used for part (b).

### 4.6.3. Thermal Tuning

In contrast to the precise matching within a filter, the absolute frequency of each filter bank channel varies by as much as 600 GHz across the wafer due to the variable thickness of the silicon or polysilicon layer. However, dense channel packing of the full FSR reduces constraints on absolute frequency control as long as individual filters can be locked to a nearest neighbor wavelength grid. This locking can then be achieved through thermal tuning using the filters' effective thermo-optic coefficient of 7.9 GHz/˚C as shown in **Figure 4.49**.

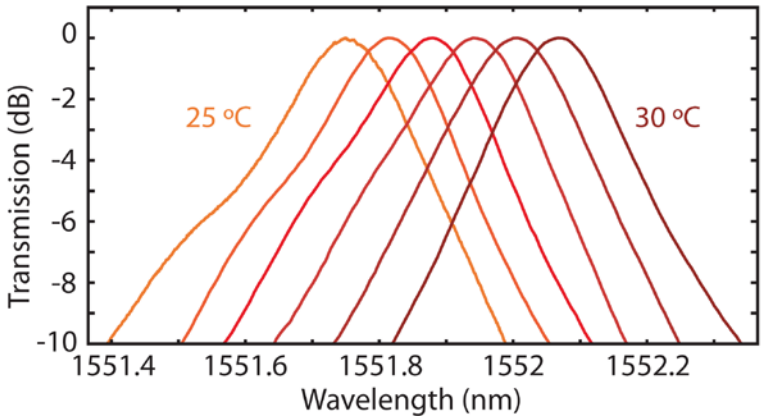


**Figure 4.49** Transmission functions of first order filter on temperature controlled stage stepped in 1 ˚C increments between 25 ˚C and 30 ˚C. A thermal tuning coefficient of 7.9 GHz/˚C is measured.

With the polysilicon heaters shown in **Figure 4.50**, the measured thermal efficiency was 1.8 °C/mW on the as-fabricated wafers. As shown in **Figure 4.51**, the undercut photonic region offers a 24x increase in heater thermal impedance and therefore over an order of magnitude reduction in thermal tuning power. If heaters are directly integrated into the ring filters, as has been demonstrated previously, the

estimated efficiency of 3 μW/GHz would represent the most power efficient tuning reported.

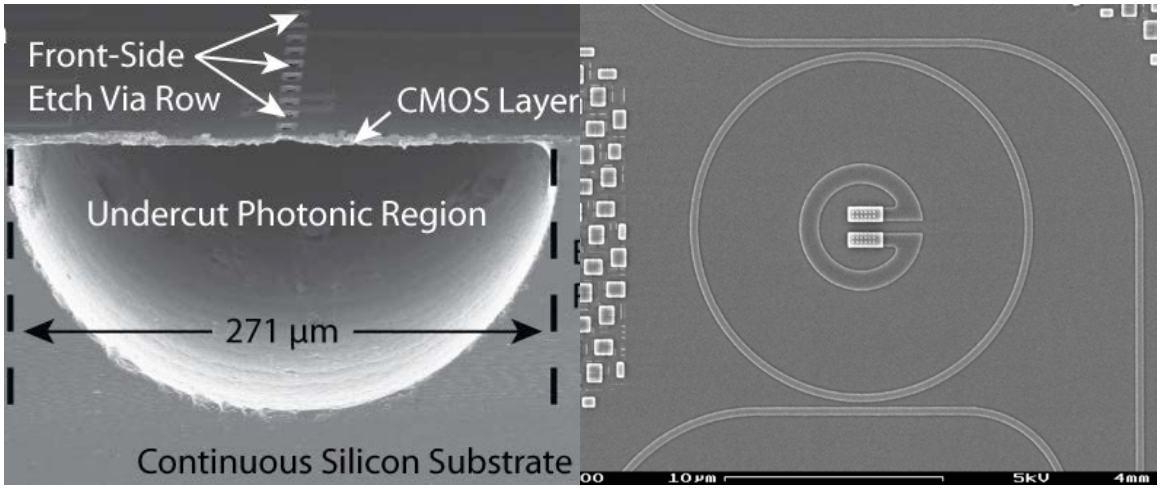


**Figure 4.50** SEM of the undercut localized-substrate removal processing used for measurement and the ring heater on EOS2.

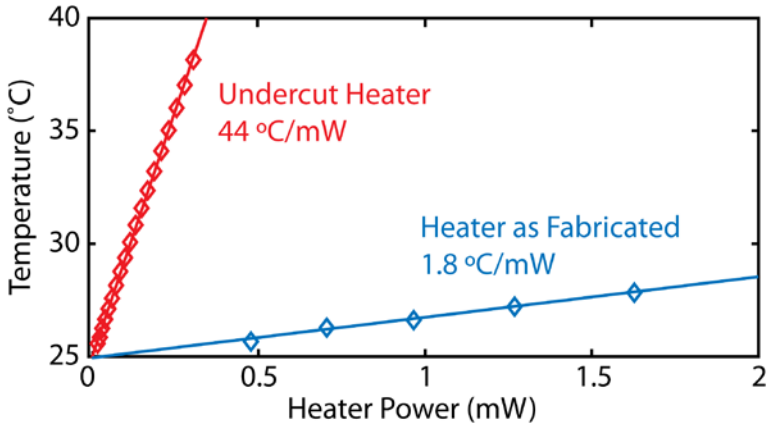


**Figure 4.51** Heater temperature plotted as a function of heater power in the photonic region before and after the localized substrate removal process. The thermal impedences are the slopes of the linear fit lines. A thermal impedence increase of ~4x, 11 mK/μW to 43 mK/μW, is observed, which yields a proportional reduction in thermal tuning power.

The total thermal impedance for the fabricated heaters is determined by a combination of the isolation from the substrate heat sinking as well as the isolation of connecting electrical leads. In the SOI platform, a more conservative heater design that is constructed out of standard design kit precision resistors is limited by a higher thermal conductivity of the connecting electrical leads. Including the poor

thermal coupling of the heater to the ring, the measured ring tuning efficiency is 34 μW/GHz with a 9.6 GHz/K effective thermo-optic coefficient in the localized substrate removal region shown in **Figure 4.52**.

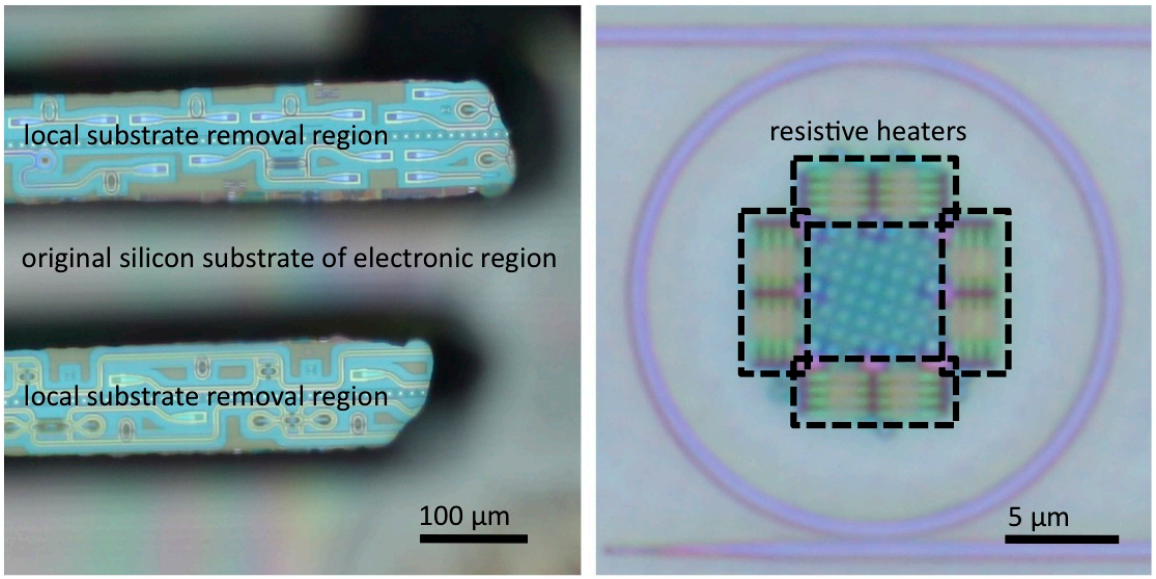


**Figure 4.52** Optical micrograph of integrated heaters and backside localized substrate removal region from EOS4-12.

The integrated electronics, shown in **Figure 4.53**, were then used to thermally tune the fabricated 8-channel first-order filter bank to a 250 GHz grid out of the 2.04THz FSR. Once tuned, the adjacent channel cross-talk is below -20 dB as shown by the overlayed drop ports in **Figure 4.53** (d). Although the filter tuning efficiency of 2.6mW/nm is relatively low [231] due to the simple heater geometry used in this work, the total power required to tune the filters to the grid was only 10.3mW including the integrated electronics on a substrate transferred die. In comparison to the best published full FSR tuning power of 21 mW for a single ring filter channel [232], the lithographic precision significantly reduces total thermal tuning power by enabling this nearest neighbor locking where link architecture allows for such a scheme [198].

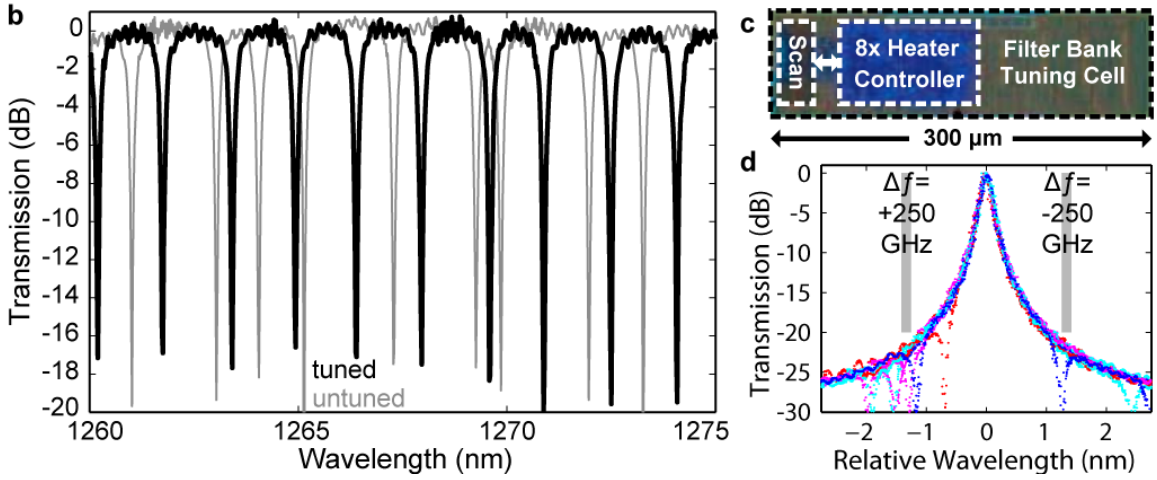


**Figure 4.53** The aggregate bank through port transmission function (b) is measured before and after the heater tuning by integrated controllers (c). The as-fabricated filter characteristics verify channels distributed throughout the FSR to yield no gaps in the wavelength grid. The overlayed drop port characteristics for all 8 channels are shown in (d) to demonstrate less than -20dB cross talk between adjacent 250 GHz spaced channels across the full 30 GHz channel width for all ports.

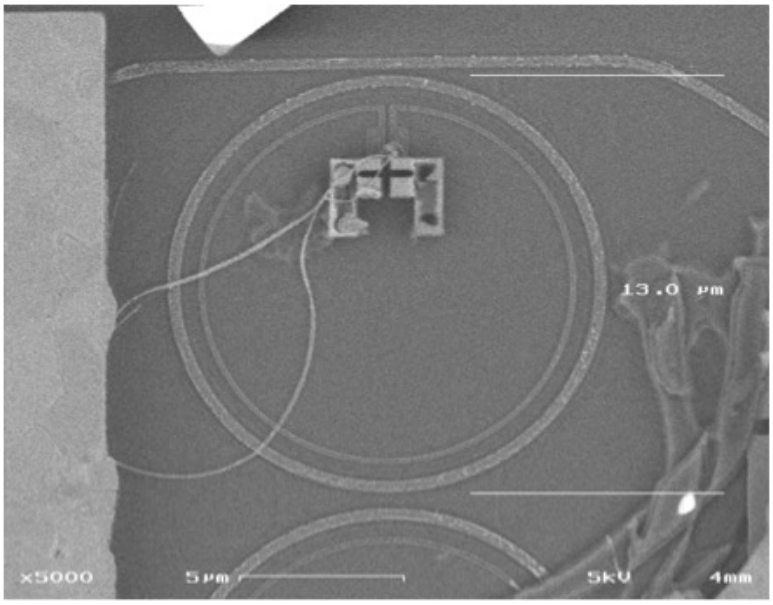


**Figure 4.54** SEM of heater on EOS1 after exposure through reactive ion etching.

Since the thermal tuning efficiency figures prominently in the total link energy efficiency, further improvements in heater design can improve system performance. Integrated heaters in adiabatic microring geometries may provide a suitable

solution when combined with the localized substrate removal thermal isolation [233]. However, intermediate solutions such as the tightly integrated microheater shown in **Figure 4.54**, may provide suitable thermal performance without requiring direct electrical contact to the optical resonator.

## 4.7. Future Work

Many aspects of the required passive photonic platform for wavelength division multiplexed integrated interconnects have been demonstrated in this work. Further development of nearly every component is required to meet desired system specifications. The one important exception is the integrated waveguide loss. Low loss waveguides have been demonstrated in both optimized polysilicon platforms and thin-SOI-CMOS foundries. The first-order microring resonator filters in the SOI platform have been demonstrated with suitable performance up to an 8-channel bank. The transmission matrix model fit to the realizable parameters predicts suitable performance up to 32-channel filter banks, but it has not yet been attempted to fabricate such a bank. Above such a channel count, a second-order filter bank would be required. Although tight fabrication tolerances have been demonstrated for second-order filters, post-fabrication trimming would still be required to precisely align the ring resonances. This additional complication may make second-order filters undesirable from a system perspective.

The vertical grating coupler is the major system component that still requires further development for desirable system characteristics. Since five grating coupler interfaces are required in the worst case for a single source processor-to-memory link, lower insertion loss than what has been demonstrated in the simple grating geometries attempted so far is a critical area of future development. Proof-of-concepts of suitable performance exist in the literature and the fabrication flexibility present in the process should enable similar low insertion loss couplers.

Further work is required for optimization of other passive components such as power splitters and optimized bend geometries are required to enable a compact and robust on-chip network. Polarization splitting grating couplers could enable the use of single mode optical fiber if simple on-chip coherent beam combining could be demonstrated. Additionally, vertical grating couplers with tunable output position or angle may be able to dramatically reduce packaging dimensional requirements.

# 5. ACTIVE DEVICES

The final aspects of the electronic integration platform that will be discussed in this thesis are the modulator and photodetector devices and their interaction with the CMOS circuitry. This chapter will introduce the modeling framework used to simulate the active devices, introduce the electronic integration platform that was used for device development and then proceed to describe the design and experimental results for the modulator and photodetectors that have been achieved over the course of the thesis work.

The modulator device design consists primarily of extending the high performance ring resonator devices presented in Chapter 4 to include electrical contacts. For the zero-change CMOS integration case, the problems that will be discussed primarily focus on the optical design of the electrical contact leads to the resonator. The DRAM process includes sufficient process flexibility to design and fabricate modulators that resemble successful prior work from the silicon photonic literature. The focus of the discussion of the DRAM integration of the modulator will primarily be on the process development required to fabricate such devices.

The photodiode device design differs significantly between the CMOS and DRAM platform. In the zero-change CMOS integration work, the constraints of the existing process require a novel device design that differs significantly from the traditional p-i-n photodiode that was discussed in Chapter 1. The relevant process constraints will be discussed before analyzing the predicted device performance. Initial experimental data for fabricated devices will then be presented. For the DRAM integration platform, the chief design challenge is the optical coupling from the polysilicon waveguide into the silicon germanium or germanium p-i-n photodetector that is fabricated on the silicon substrate. Unlike SOI processes used for photodetector integration in the literature, the single-crystalline layer used for the epitaxial growth template can't be optically isolated from the underlying silicon substrate with a thick low-index cladding layer. Instead, the epitaxial stackup will be engineered to support a rib-mode that will be butt-coupled to the input polysilicon waveguide. Since the DRAM integration platform is still under active development, no fabricated devices or experimental results are available for inclusion in this thesis.

The chapter will then conclude with a review of measured current and predicted future active device performance. The performance summary will then be compared to the targets required for the demonstration of the multi-core processor to memory network that is discussed in Chapter 2.

## 5.1. Modeling Framework

In addition to the optical design approaches described in Chapter 4, electrical device simulation is required to design and estimate performance of integrated active devices. Several tools have been used for this purpose over the thesis. A brief introduction to each tool is provided to provide context for the simulations presented in the following sections. The primary tool for active device electronic and process simulation has been the Sentaurus technology computer aided design (TCAD) software package from Synopsys.

The simulation of the electrical characteristics have been performed in Sentaurus Device (**sdevice**), which is a 2D and 3D finite element solver of the modified drift-diffusion equations. Other alternative simulation tools that are widely used include Medici, which is a legacy tool from Synopsys, Silvaco Atlas and Crosslight Apsys. At the most basic level, the coupled solution of the Poisson equation, electron continuity equation and hole continuity equation provides both the I-V characteristics and the carrier density distribution for the modulators and photodetectors. For the modulator, this information is sufficient to completely predict the electrical DC device operation characteristics. For the photodetector, simulations with and without optical carrier generation sources that are representative of the predicted optical absorption spatial distribution are required to predict DC operation. The frequency response of either the modulator or photodetector can also be simulated by either transient simulations or small signal AC analysis within the same simulation framework. Example **sdevice** input files for all relevant modulator and photodetector device simulations can be found in Appendix D.

The process development work required for DRAM photonic integration has required the use of the process simulation capabilities of the Sentaurus TCAD suite using the **sprocess** simulator. This finite element process simulator replaces the widely used TSUPREME-4 simulator also available from Synopsys. The process simulator enables all wafer-level processing steps including implants, diffusion and epitaxy to be explicitly modeled. Additionally, simple combinations of isotropic layer

deposition and isotropic/anisotropic etching of the various material layers enables structure creation that matches the wafer-level fabrication steps. Simulation of the entire process has proven invaluable in debugging problems that may arise as the photonic process modules are added to the full electronics fabrication process. The new DRAM memory processes to fabricate the photonic devices can be simulated and optimized within this framework. The example that will be discussed in the modulator section is the design of multiple dopant implant steps required to form a vertical junction suitable for the realization of a depletion modulator. The results of the process simulation can then be exported for device simulation within **sdevice**. An example of this simulation path will be presented for the design of the depletion modulator device in the DRAM process. Simulation input files can be found in Appendix D.

In some cases, the properties of interest are only the parasitic resistance and capacitance of a simple electrical junction. In this case, a full simulation in Sentaurus can be burdensome for the overhead in structure definition and mesh creation. Instead, parasitic resistances and capacitances can be accurately modeled in Synopsys Raphael that requires a significantly simpler input format. Simulation examples were presented in Chapter 2 for the design of a depletion mode modulator and example input and field plotting files can be found in Appendix D as well.

## 5.2. Electronic-Photonic Integration Platform

To provide the capability to test the integrated modulators and detectors, a complete integrated electronic link backend has been included on the fabricated CMOS chips since EOS2 (28nm Texas Instruments bulk-CMOS process). The circuits, which are designed by Michael Georgas, Jonathan Leu, Benjamin Moss, Chen Sun and Vladimir Stojanovic, are divided into independent test cells that share serial data in a scan chain. An example labeled micrograph of the circuit test cell is shown in **Figure 5.1** from EOS8 (IBM 45nm SOI-CMOS process). Inside each cell, interface circuits connect to the photodiode and modulator devices. The modulator is connected to a driver circuit that can take its input from either a fixed 32-bit repeating pattern or a pseudo-random bit sequence (PRBS) generator. The receiver circuit [179] can receive data from the attached photodetector or through an electrical self-test path. The received data is compared to the transmitted data to extract a bit error rate and verify functional data link operation. A separate receiver circuit is included to efficiently receive a clock signal [195]. The individual test cells

are then cascaded to perform independent experiments on many photonic device variants. The metal contact locations for each circuit driver are kept constant relative to cell origin for all test sites. The photonic devices are then varied while maintaining consistent contact locations to connect to the electronic drivers in all test sites.

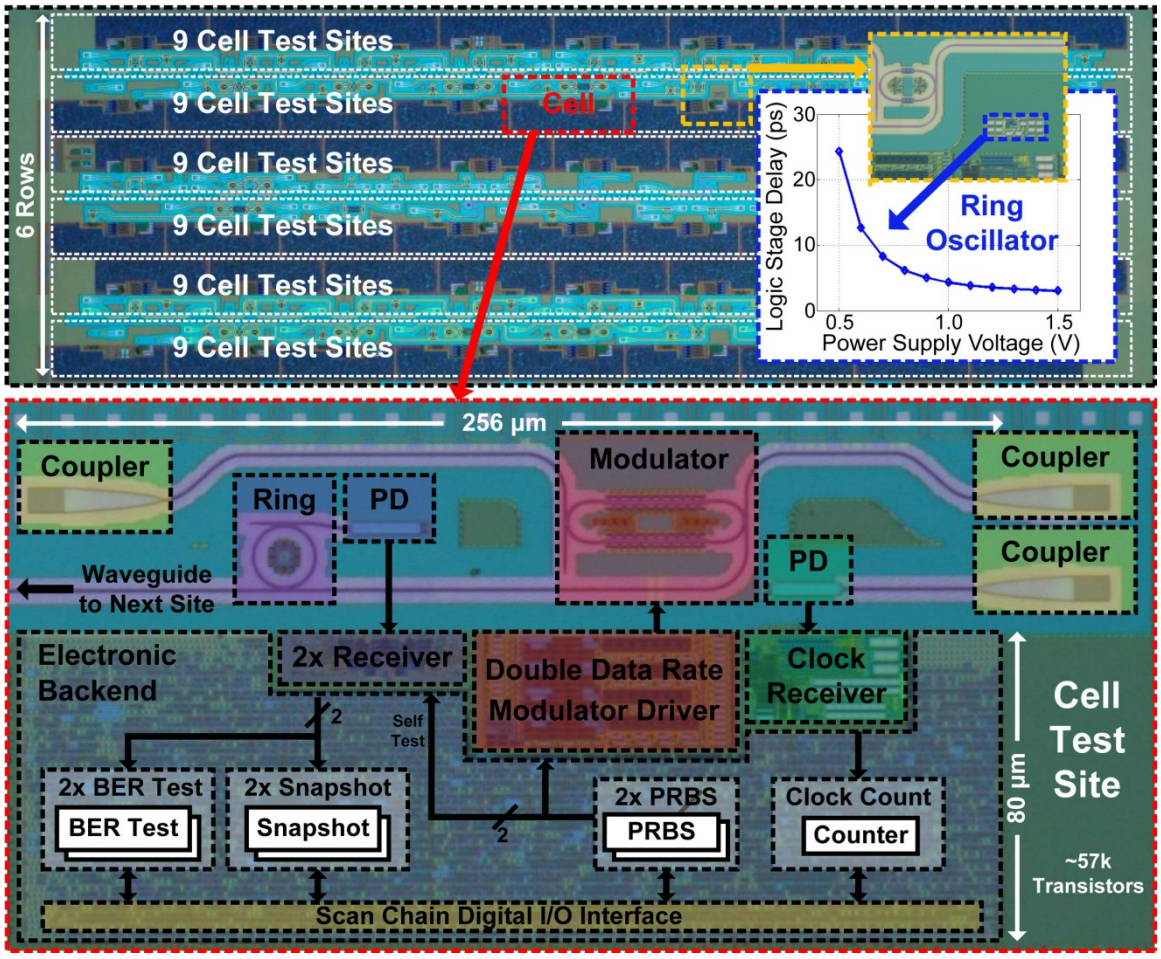


**Figure 5.1.** Illustrated micrograph of the integration platform containing 54 independently addressable electronic-photonic test cells. Each digitally interfaced cell contains the pseudorandom bit sequence (PRBS) generators and error comparators necessary for link testing. Only the transmit end of the link is reported in this work. The electronic ring oscillators verify the high-speed operation of the transistors in close proximity to the photonic devices, in the photonic integration regions.

The layout of the circuit integration test regions organizes photonic devices into isolated rows. The center of each test row contains an array of etch vias to enable localized substrate removal through a front-side undercut process. Alternatively, the

localization of the photonic devices into rows that are separated from the electronic regions enables trenches to be etched from the back side of the silicon die. These localized substrate removal techniques, introduced in Chapter 1, are described further in Appendix B.

Although the successful compliance with foundry design rules as described in Chapter 3 should ensure proper electronic device performance, integrated transistor performance test structures were included. As shown in the inset to **Figure 5.1**, ring oscillators were placed in the photonic rows. The electronic ring oscillators, which were designed by Michael Georgas, are loops of odd numbers of inverting transistor stages. Since there is no stable operating point of the circuit, any node of the circuit oscillates with a frequency that corresponds to the round trip logic delay. The inverse of the natural frequency of oscillation divided by the number of stages in the oscillator therefore characterizes the speed of the integrated transistors. An example measurement as a function of supply voltage is shown in **Figure 5.1**. The <5 ps stage delay at the nominal supply voltage not only verifies proper transistor functionality, but also represents the fastest reported transistors ever monolithically integrated with a waveguide-based photonic platform. For properly designed RF transistors, the measured $f_t$ after de-embedding external parasitics is ~485 GHz in this CMOS process [234]. This is over 5 times the highest $f_t$ reported for III-V-based opto-electronic integrated circuits due to compromises between the electronic and photonic device epitaxial stackup [16].

Although localized substrate removal is the desired post-processing methodology for a complete system, all initial active device results presented in this chapter were achieved using a complete substrate removal and transfer process. The details of the developed process are presented in Appendix B. The end result of the process is that the SOI-CMOS stackup from the buried oxide layer through the top bond pads and surface passivation has been removed from the silicon substrate and bonded to a 6H-SiC die using Norland Optical Adhesive 71. The process was developed to enable access to the top surface pads for direct probing of the independently accessible active photonic devices. It was assumed that the transistors would not function after the substrate modification due to prior literature emphasizing the importance of the silicon handle wafer on SOI transistor electrostatics [83, 84]. For completeness, the transistor performance was measured after the substrate transfer process. The surprising result, shown in **Figure 5.2** as the overlapping data points from multiple substrate-transferred and as-fabricated die,

is that the transistor characteristics were largely unchanged by the substrate transfer process. This behavior may be attributed to the fact that the SOI transistors in this process have only partially-depleted bodies instead of the fully-depleted body prediction that is present in most SOI transistor scaling theoretical studies [83, 84].

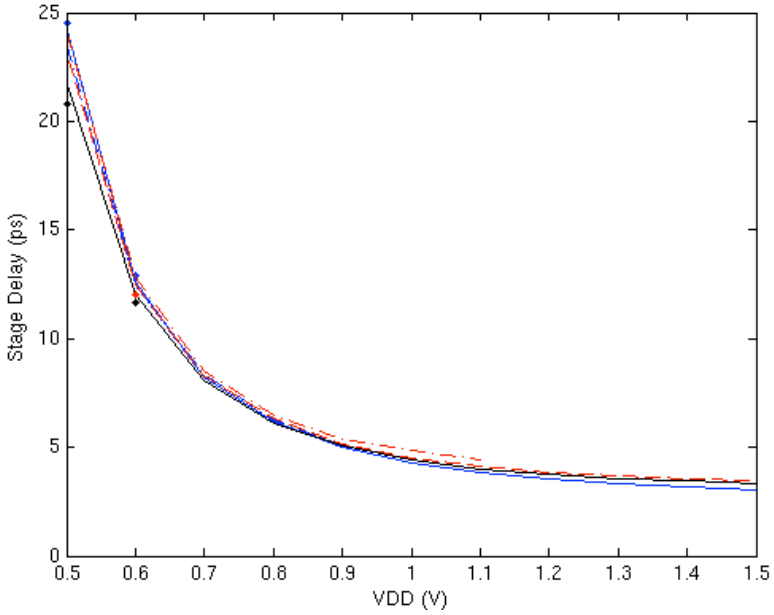


**Figure 5.2** Stage delay measured for ring oscillators from multiple as-fabricated and substrate-transferred dies. All measurements agree to within 5% extracted stage delay estimates.

With the positive result of the transistor functionality on the substrate-transferred die, a complete packaging test was performed to enable the backend electronic circuits. The substrate-transferred dies were sent to multiple commercial wire-bond vendors to connect the ~180 required pads into a ceramic pin grid array package for electro-optic testing. Although we have encountered wire-bond yield problems as discussed in Appendix B, completely functional electronic-photonic die were obtained by this method. The roughly 3 million integrated transistors meet all required timing margins for digital backend functionality on the substrate-transferred die. This has provided the first path to enable the monolithically integrated active devices and simultaneous electronic-photonic functionality.

## 5.3. CMOS Integrated Modulators

Integrated ring resonator modulators have been included on most test chips since EOS1, which was fabricated in the Texas Instruments 65nm bulk-CMOS process. Although high-performance silicon photonic modulators have been demonstrated in customized processes, new device designs are required to implement optical modulators that function with the appropriate layer structure required for zero-change CMOS integration. Initial devices for the first two CMOS test chips were not rigorously simulated, but instead designed based on a set of assumptions that did not lead to high performance devices. Revised designs and geometries that were not possible in these initial fabrication runs have led to the first examples of integrated CMOS modulators.

### 5.3.1. CMOS Process Constraints and Material Parameters

A major benefit of front-end integration in scaled-CMOS technologies is that there are many existing doping and metallization steps present for electronic device formation that are therefore available to the photonic device designer. Since both the body-Si and poly-Si layers must be used to form both polarities of transistors as well as resistors, local interconnect and capacitors, the layers must be doped in a wide variety of densities both n-type and p-type. Additionally, silicidation, typically a nickel silicide, to reduce sheet resistance is available for both layers as well as the capability to locally mask the process for high resistance regions. There are then 6-12 copper metal layers that are available for connecting the active photonic devices to the integrated transistors through low resistance tungsten contacts.

This inherent design flexibility of the process allows the creation of a lateral p-i-n diode, as shown in **Figure 5.3**(a). Although significant flexibility exists in the doping levels used for the junction, the physical silicon cross section differs significantly from prior work in silicon modulators. Most prior silicon modulator designs have used multi-level silicon cross sections to provide lateral optical confinement of the optical mode away from the lossy electrical contacts as shown in **Figure 5.3**(b-c). For SOI-CMOS processes, a similar approximation to these geometries can be achieved with some significant tradeoffs as will be discussed in Section 5.3.3. For planar bulk-CMOS processes, it is impossible to perform this sort of cross-section engineering. Prior work in significantly thicker slab waveguide geometries have demonstrated high performance modulators by including a vertical

pn-junction in tightly bent resonators as shown in **Figure 5.3**(d). Since a similar vertical-junction-based design is forbidden with the existing implant steps and layer thickness, a new modulator device geometry was required for zero-change bulk-CMOS integration.

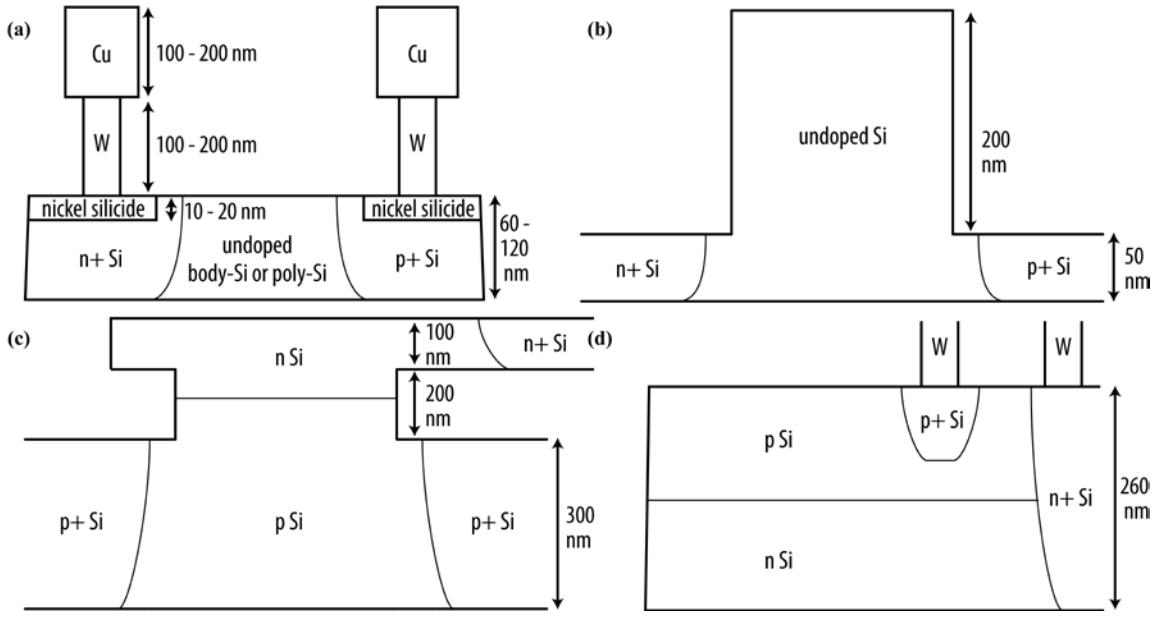


**Figure 5.3** (a) Slab waveguide CMOS modulator cross-section differs greatly from the geometries of previously demonstrated silicon modulators: (b) injection-mode rib waveguide p-i-n diode [73, 74, 235, 236], (c) depletion-mode vertical junction rib waveguide [237, 238] and (d) depletion-mode vertical junction slab waveguide diode [92].

### 5.3.2. Slab Contact Geometries

With the cross-section flexibility set by constraint of using the existing CMOS process, the remaining degree of freedom is to pattern the waveguide in the direction of optical propagation. The first idea was to leverage the high resolution lithography of the CMOS process to pattern very small contact arms to the silicon waveguide. The design shown in **Figure 5.4** leveraged 100 nm wide contact arms to separate the high optical loss contact metals from the optical mode while providing optical connectivity. The assumption was that if the contact arm was made narrow enough the light would not diffract significantly before rejoining the waveguide mode on the other side of the arm. The primary impact of each contact arm was assumed to be merely an impedance discontinuity that would primarily lead to a

reflection of the input mode. As such, the arms were placed at ¾ of the wavelength of the mode in the media such that adjacent reflections would interfere destructively.

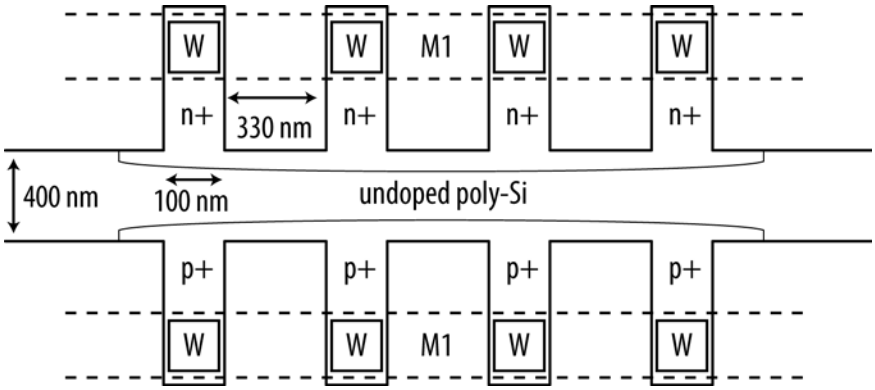


**Figure 5.4** Modulator diode top view detailing arm layout to contact the slab modulator.

Since the bent waveguide mode would not properly match at the input and output waveguides of each arm, the arm contact geometries were only placed in the straight sections of a racetrack resonator as shown in **Figure 5.5** and **Figure 5.6**. The straight-bend mode mismatch of an abrupt transition limits the minimum bend radius for such a racetrack geometry instead of the radiation-limited bend losses that were calculated in Chapter 1. In later test chips, an adiabatic bend design that was created by Milos Popovic was used to alleviate this constraint.

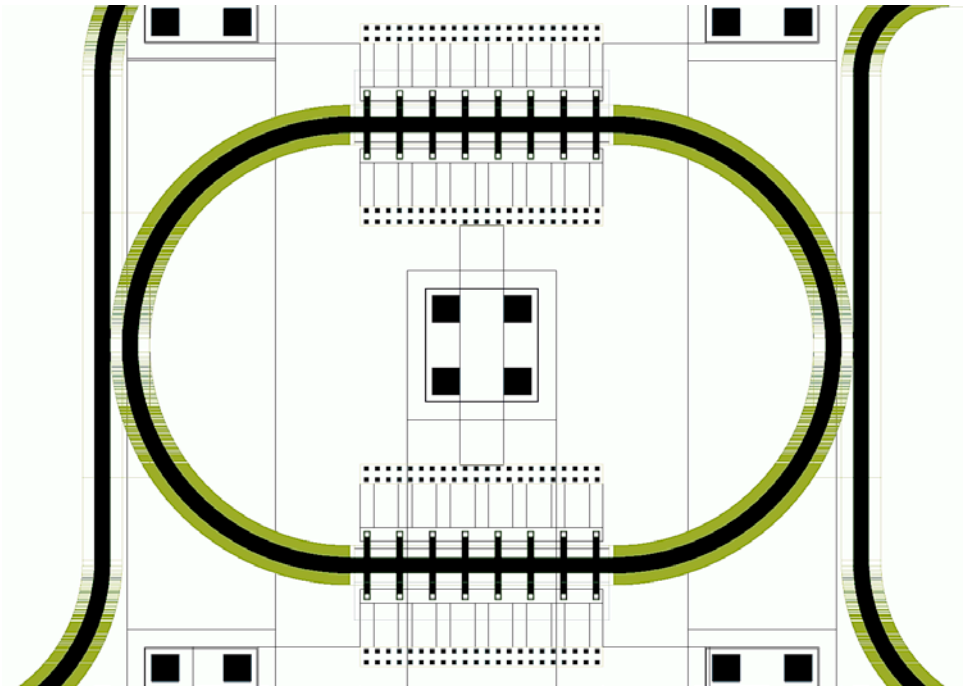

**Figure 5.5** Design environment screen shot of slab-contact racetrack resonator from EOS1.

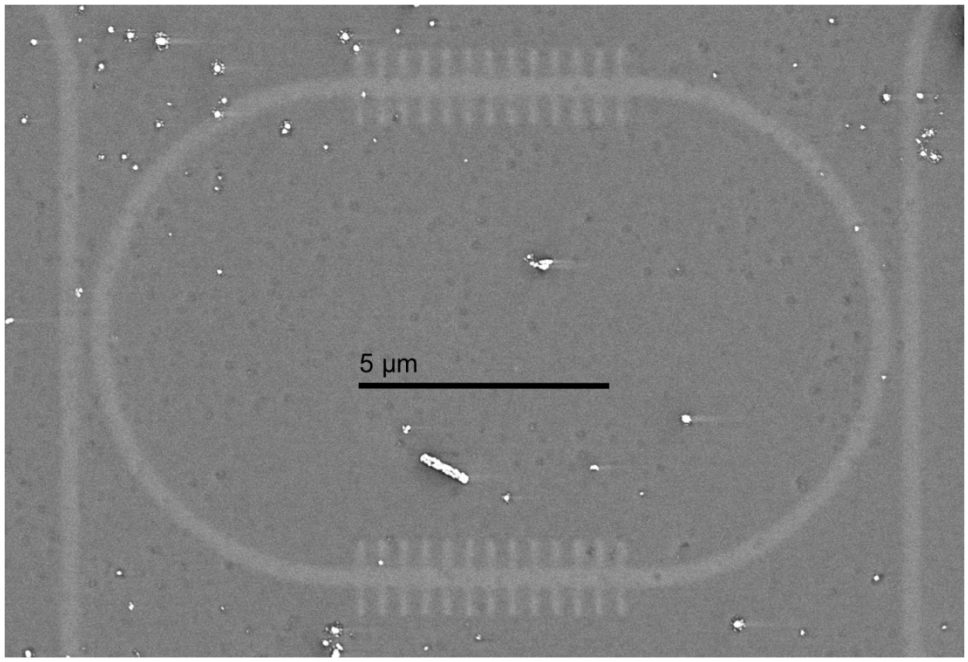


**Figure 5.6** Scanning electron micrograph (SEM) of a fabricated racetrack resonator from EOS1 after back-side etching to expose the polysilicon layer. SEM analysis and etching performed by Charles "Trey" Holzwarth III.

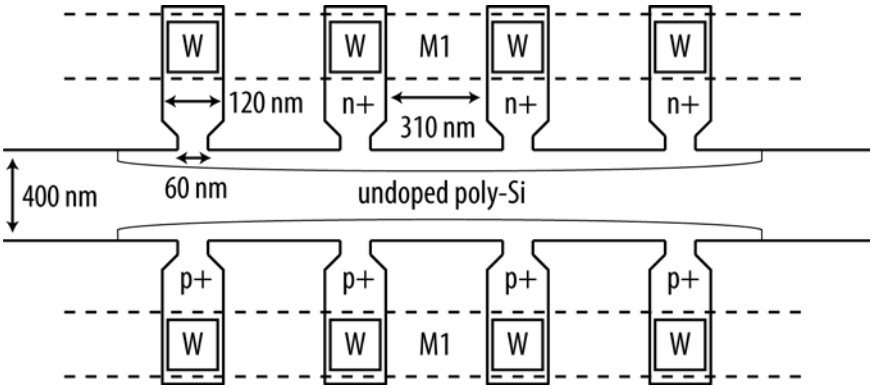


**Figure 5.7** Revised arm contact geometry employing a taper at the connection point to the waveguide to minimize free-diffraction region.

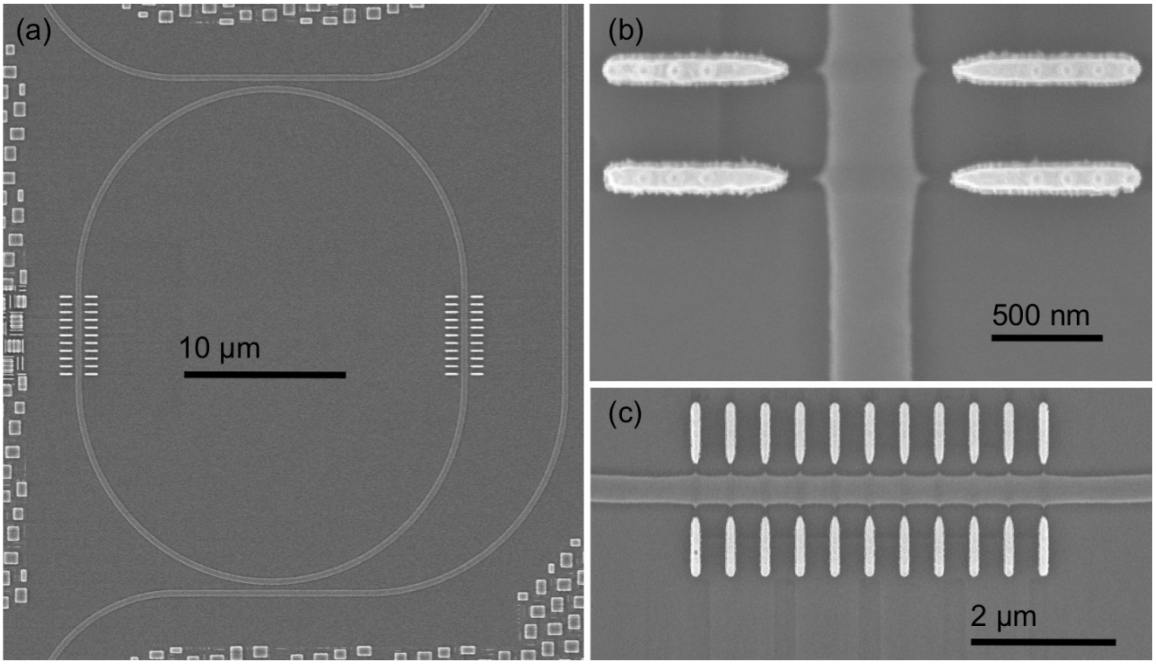


**Figure 5.8** Scanning electron micrograph (SEM) of a fabricated racetrack resonator from EOS2 after front-side etching to expose the polysilicon layer. The brighter sections of the contact arms are silicided to reduce the contact resistance. SEM analysis and etching performed by Charles "Trey" Holzwarth III.

Due to the schedule of the chip tape-outs a revised design was submitted for fabrication in the Texas Instruments 28nm bulk-CMOS process before the performance of the initial design could be characterized. The revised design included tapered contacts at the intersection of the waveguide as shown in **Figure 5.7**. The purpose of the tapered contacts was to minimize the length of the unguided

region in which the input optical mode diffracts in the plane of the contact arm. The fabricated silicon tethers were distorted by the lack of optical proximity correction for the EOS2. Unlike all other CMOS test chips, the structures on EOS2 were blocked from the standard foundry optical proximity correction algorithm due to data preparation concerns. As such, the optical diffraction effects were not pre-compensated for in mask design. The fabricated contact arms are narrower than anticipated and distort the waveguide to which they connect. Scanning electron micrographs (SEMs) of the fabricated arms after exposure through front-side etching are shown in **Figure 5.8**.

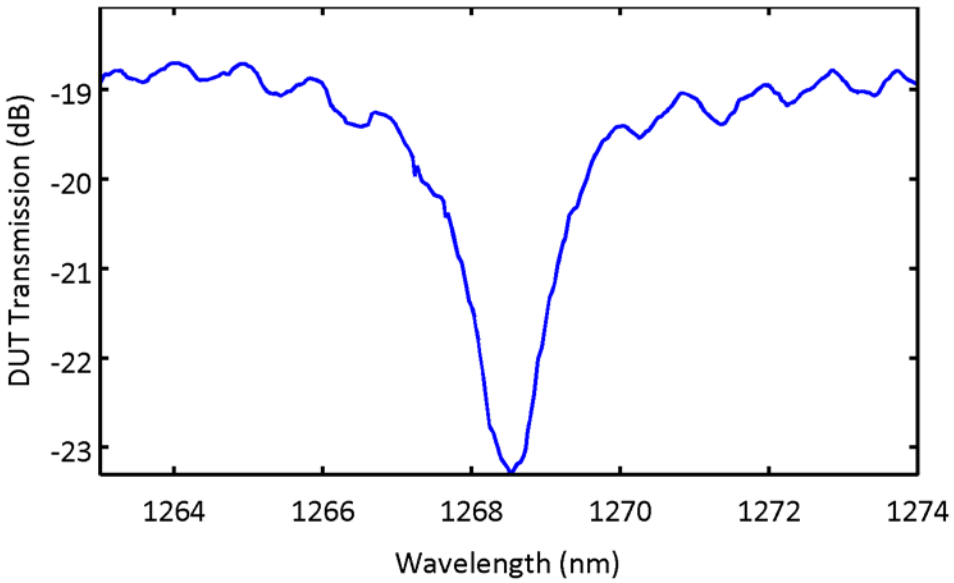


**Figure 5.9** Measured fiber-to-fiber transmission function for a slab contact optical modulator that was fabricated in EOS2. Measured quality factor is approximately 600.

An example optical transmission function for these initial slab contact modulator designs is shown in **Figure 5.9**. Although the resulting device characteristics enable modulator functionality, significant improvement is required to meet desired performance specifications for the integrated photonic links outlined in Chapter 2. The optical quality factor of ~600 would require an electro-optic shift of almost 400 GHz to achieve modulation with a 3 dB extinction ratio. Although such shifts are achievable through carrier injection, this shift is significantly larger than the ~20 GHz resonance shift that is achievable through the more energy efficient carrier depletion mode of operation at CMOS-compatible voltages as discussed in Chapter 1. Additionally, since the contact-induced loss is significantly larger than expected at

design time, the coupling coefficient of the bus waveguide to the ring resonator was insufficiently strong to enable high on-resonance extinction.

The electrical properties of the junction demonstrated acceptable performance for the integrated drivers with the measured IV characteristics shown in **Figure 5.10** on wafers where transistors also met performance targets. A forward bias voltage of 1.25V was required to drive the 1 mA through the junction that is the approximate requirement to switch the modulator. This voltage is within the 1.5 V achievable in scaled-CMOS processes. The measured series resistance of 417 Ω matches calculations of the sheet resistance of the narrow contact arms. Depending on the driving circuit, it may be desirable to reduce this series resistance with alternate contact geometries.

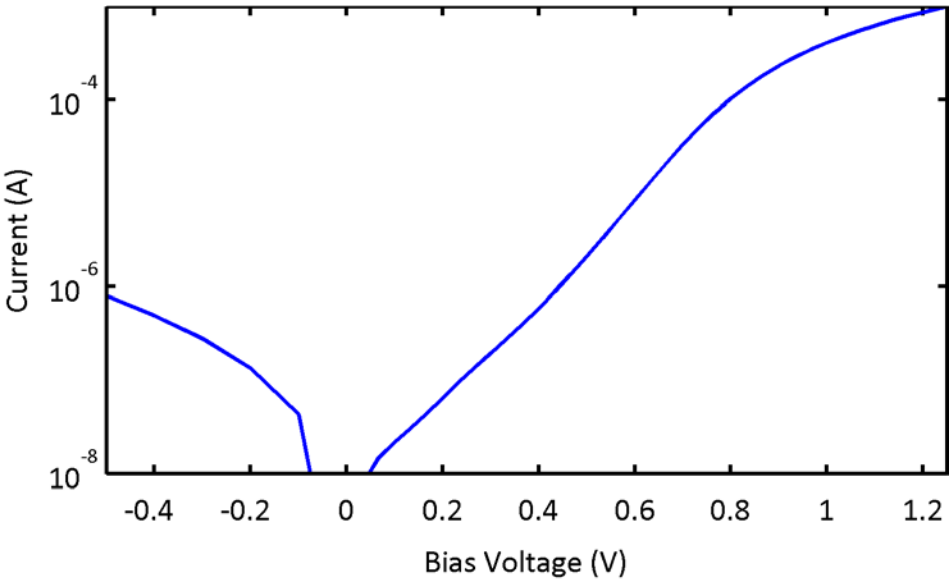


**Figure 5.10** IV characteristics of modulators fabricated on EOS2. Forward current of 1 mA obtained at 1.25 V. Measured series resistance is approximately 417 Ω.

Due to the developmental stage of the process used for this test chip, significant differences in device characteristics were observed from wafer-to-wafer as processing conditions were varied. Although the IV characteristics of modulator devices matched the expected series resistance and turn on voltage on wafers where transistors met performance targets, wafers that had suitable optical properties for electro-optic testing had much weaker turn on characteristics consistent with significantly increased intrinsic region widths. The measured IV characteristic for one such modulator fabricated on an optically functional wafer is shown in **Figure**

**5.11** (a). Although the large intrinsic region significantly altered intended device operating characteristics, a wavelength shift was observable under forward bias as shown in **Figure 5.11** (b). The direction of the wavelength shift indicates that the resonance is tuning by the thermo-optic effect. Any free carrier dispersion effects, which would cause a blue shift, that may be present are masked by the thermal shift under DC testing. Further testing such as modulation experiments were not performed for these modulators.

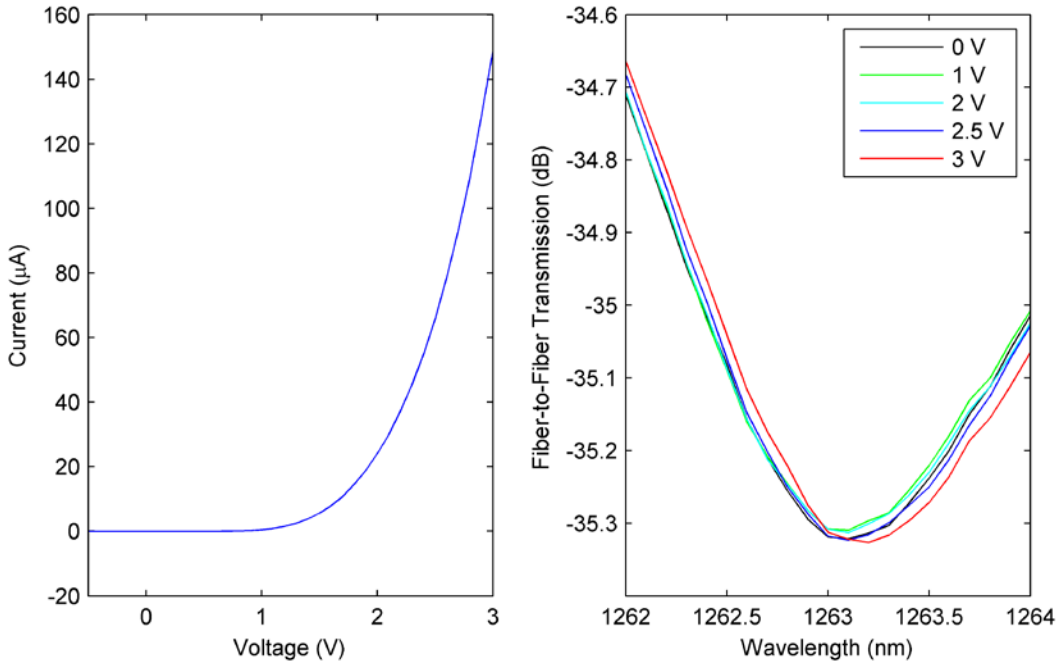


**Figure 5.11** (a) IV characteristics for modulator on optically functional wafer showing increased turn on voltage. (b) Fiber-to-fiber transmission characteristics for substrate transferred slab contact modulator from the EOS2 test chip.

After this initial slab-contact modulator design exploration, it was necessary to simulate the tradeoffs involved in the slab contact geometry to optimize the design. For this effort, I worked with Eugen Zgraggen to first verify a simulation model using the measured data and to then use the model to develop an optimized design [239]. The simulation model, which is fully described in [239], was verified against the EOS2 modulator designs as shown in **Figure 5.12**. In summary, it consists of an extension of the transmission matrix model of the ring resonator presented in Chapter 1 to include a scattering matrix of the contact arm geometry that is obtained by a FIMMPROP simulation. The reduced performance relative to the expectation of the contact geometry was seen to primarily involve the assumption

that the role of the waveguide impedance discontinuity was to primarily cause reflections. Unlike an impedance discontinuity experienced by a plane wave in infinite media, the disturbance to a localized waveguide mode can also cause coupling to radiation modes that will not destructively interfere in the contact array. The cancelling of the reflected mode was observed to be quite successful in simulation, however, it was not the dominant source of loss.

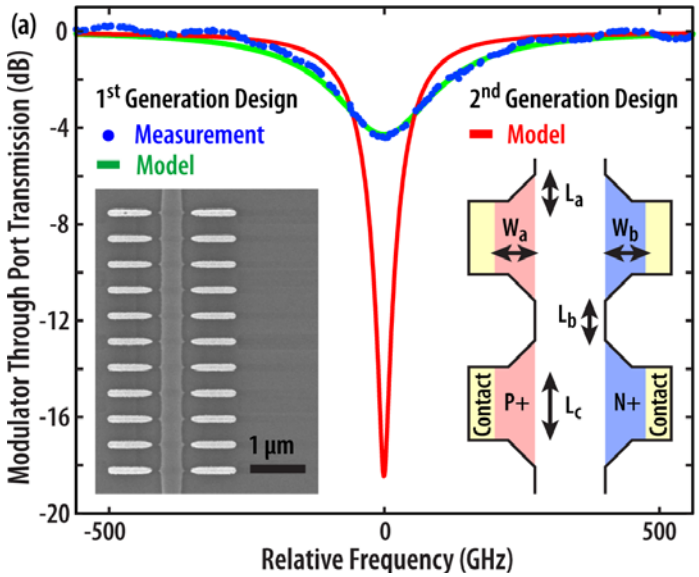


**Figure 5.12** Modulator contact geometry optimization shown through measured and modeled transmission functions.

The simulation model was then used to develop the new slab contact that is detailed in **Figure 5.13**. The transition from the singlemode waveguide to the electrical contact region was then tapered at a 45° angle. Additionally, the width of the contact area was increased to minimize the number of contacts required to achieve sufficient carrier injection. The width of each contact, referred to as fingers, was then optimized to maximize transmission from the input to output singlemode waveguides. After the parameter optimization, the reflection of the finger was minimal such that the spacing between fingers did not have a significant impact on device function. The independence of device function to spacing between adjacent fingers was verified experimentally by measuring the optical transmission function of resonant modulators with identical path lengths and numbers of fingers, but different distances between adjacent contacts. A micrograph of the fabricated device employing the adiabatic bends developed by Milos Popovic and six total carrier-injection fingers is shown in **Figure 5.14**.

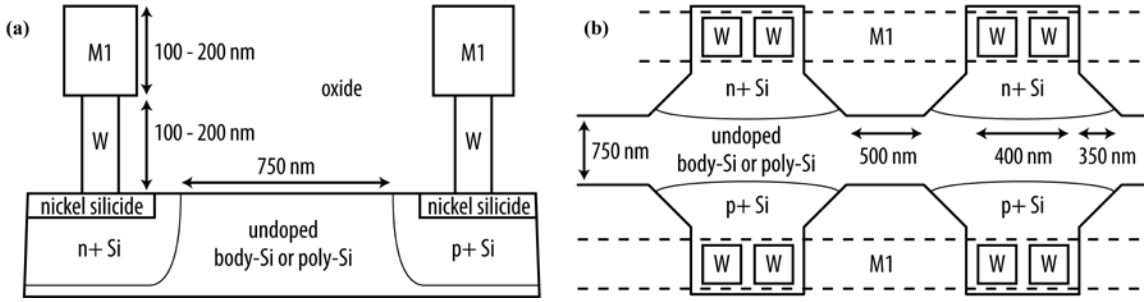


**Figure 5.13** Optimized modulator slab contact geometry developed for EOS4.

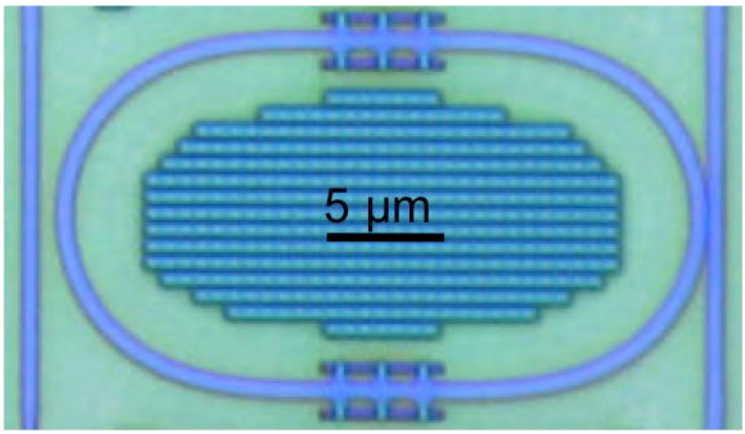


**Figure 5.14** Optical micrograph of optimized slab-contact modulator device fabricated on EOS8.

The optical transmission characteristics of the finger-contact resonators were then measured for 1280nm and 1550nm as shown in **Figure 5.15** and **Figure 5.16** respectively. The maximum measured quality factor of ~1150 for the 1280nm design closely approximates the ~1280 that was predicted by the design model. The high on-resonance extinction of greater than 25 dB indicates that the fabricated structure is nearly critically-coupled due to the accurate simulation prediction. Compared to the original narrow arm contacts, the improved finger design also enabled a factor of two increase in resonance quality factor.

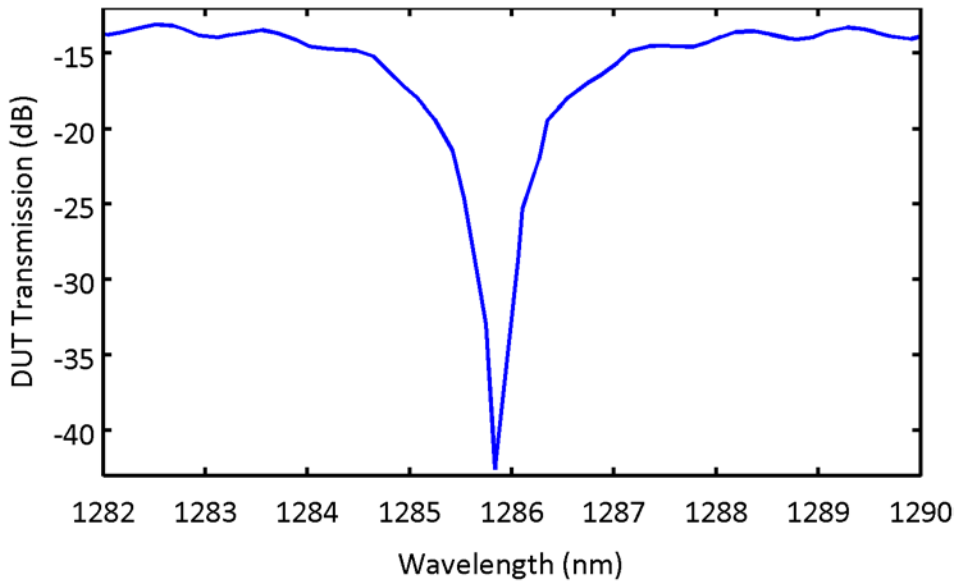


**Figure 5.15** Optical transmission function for the slab modulator finger-contact design for the 1280nm wavelength range from as measured on EOS8.

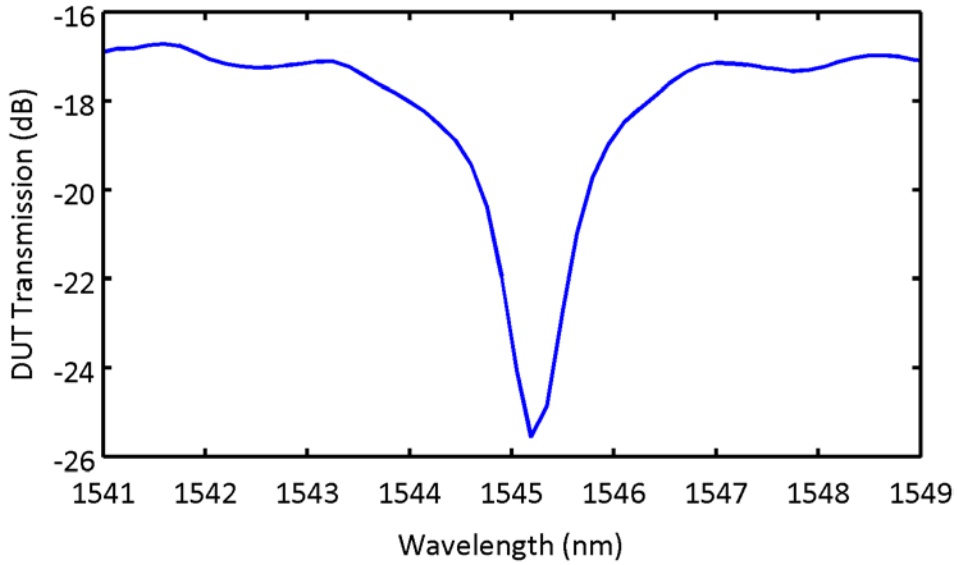


**Figure 5.16** Optical transmission function for the slab modulator finger contact design for the 1550nm wavelength range from as measured on EOS8.

The larger finger contacts to the optical waveguide also enabled lower series resistance for the p-i-n diodes. As shown in **Figure 5.17**, the revised diode geometries achieve 1 mA at 1.14 V instead 1.25 V as in the case of the original contact design. The series resistance of 53 Ω is ~8x lower as well. It is interesting to note that although the series resistance is lower than the previous design, the

voltage required to inject 1 mA of current is larger than in the original contact design. The other variable between these two cases is the intrinsic region width of the p-i-n diode that is formed using the contacts. For the diode formed in **Figure 5.17**, the intrinsic region length is 500 nm. This is roughly 5x the ~100nm intrinsic region formed in the diode of **Figure 5.10**.

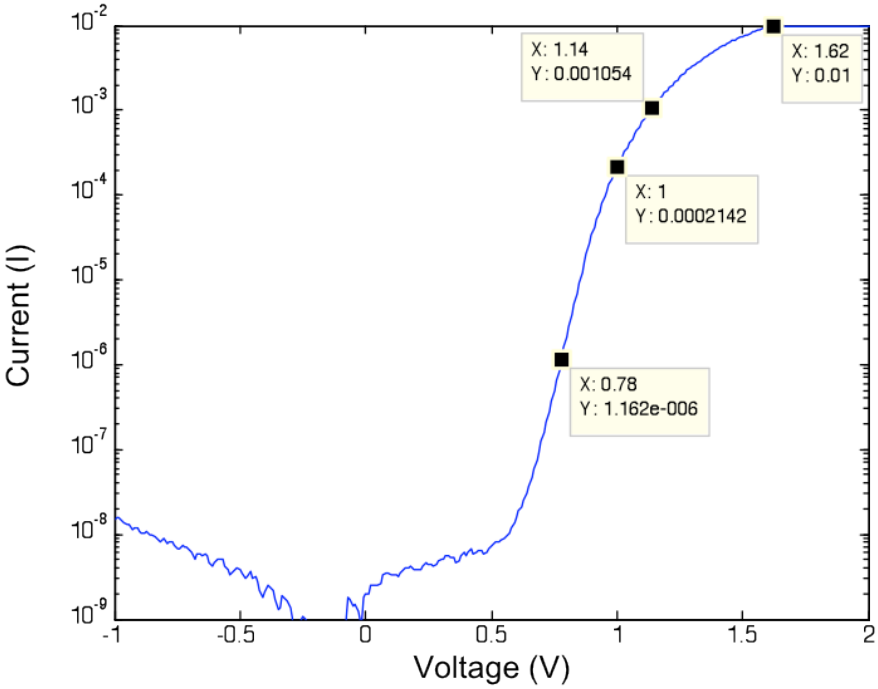


**Figure 5.17** Measured IV characteristics for the finger contact geometry optical modulators fabricated on EOS8.

The required electro-optic shift of ~150 GHz or ~1.5 nm is still almost an order of magnitude more than what may be achieved in a carrier depletion modulator. As such, further investigation into novel slab modulator contact geometries is required for zero-change bulk-CMOS integration. This remains an active research project.

### 5.3.3. Rib Contact Geometries

The SOI process enables an additional degree of process flexibility for optical modulator design by the presence of two independently patternable silicon layers: the single-crystalline silicon layer used to form transistor bodies and the polysilicon layer used to form transistor gates. This flexibility enables the design of a modified rib modulator in which a strip of polysilicon in the center of a wider single-crystalline silicon layer provides lateral optical confinement. To design a rib

modulator suitable for fabrication in the existing SOI-CMOS process, I worked with Jeff Shainline to develop the design shown in **Figure 5.18** and **Figure 5.19**. The polysilicon width was sized to support only a single bound mode that is shown in **Figure 5.19**(b). The bend radii of 8 μm at 1280nm and 10 μm at 1550nm were set to support a radiation-limited quality factor of 1 million.

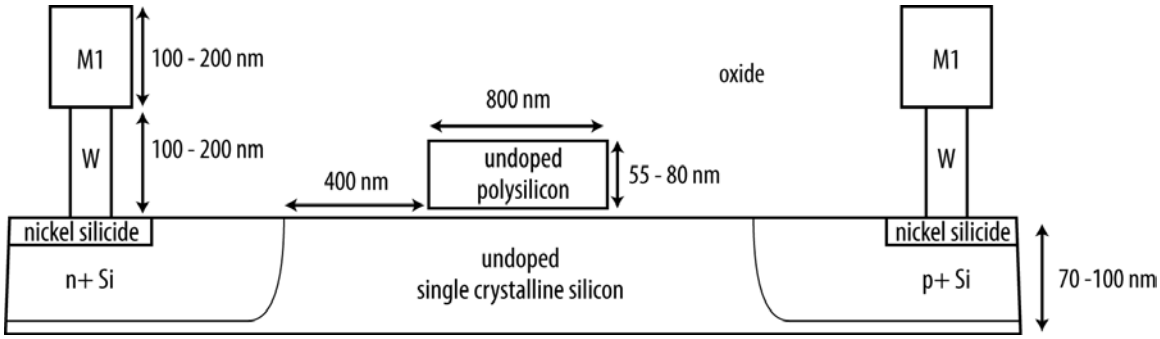


**Figure 5.18** Cross-section sketch of the modified rib modulator design for 1550nm.

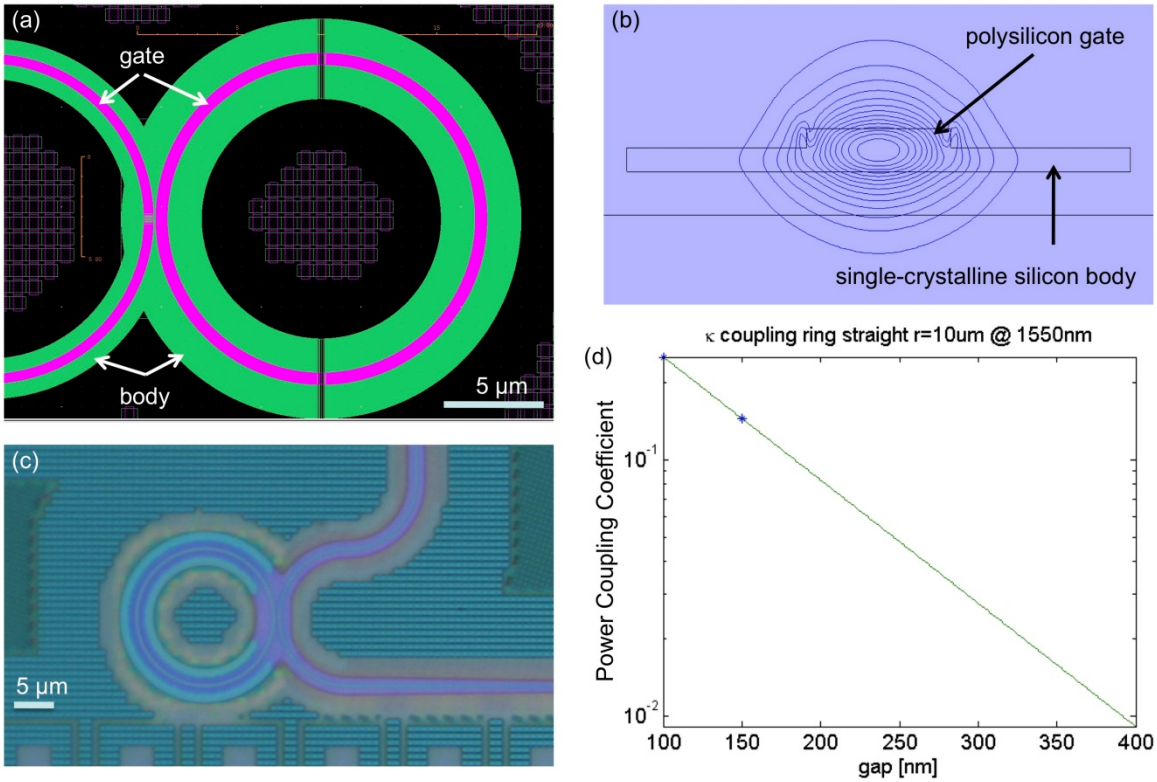


**Figure 5.19** Rib modulator design details. (a) Design environment screenshot showing only the two silicon layers. (b) Simulated electric field mode profile demonstrating the lateral optical confinement. (c) Optical micrograph of the fabricated structure. (d) Coupling vs. gap finite difference simulation used to design the coupling to the resonator.

The coupling, simulated by FDTD and shown in **Figure 5.19**(d), was designed to support an optical loss set by a doping density of 2 x $10^{18}$ $cm^{-3}$ to allow the modulator to be used as a depletion modulator. The equivalent waveguide loss of this doping level was estimated to be ~100 dB/cm as shown in **Figure 1.20**. The fabricated modulator, however, was not subject to the intended doping steps and the resulting structure was the p-i-n diode shown in **Figure 5.18**. Despite the change in the design, the measured optical transfer functions, shown in **Figure 5.20** and **Figure 5.21** for 1280nm and 1550nm respectively, demonstrate near critical coupling. This behavior can be largely explained by the higher loss than expected in the polysilicon layer. Independent polysilicon waveguides were measured to have ~110 dB/cm propagation loss. The expectation from previous test chips was closer to 50 dB/cm. The fabricated resonances were measured to have quality factors of 3970 and 4290 for 1280nm and 1550nm respectively. The corresponding 3dB optical bandwidths of 60 GHz and 45 GHz at 1280nm and 1550nm respectively are within a factor of 2 or 3 of the anticipated electro-optic shifts required for depletion-mode operation.

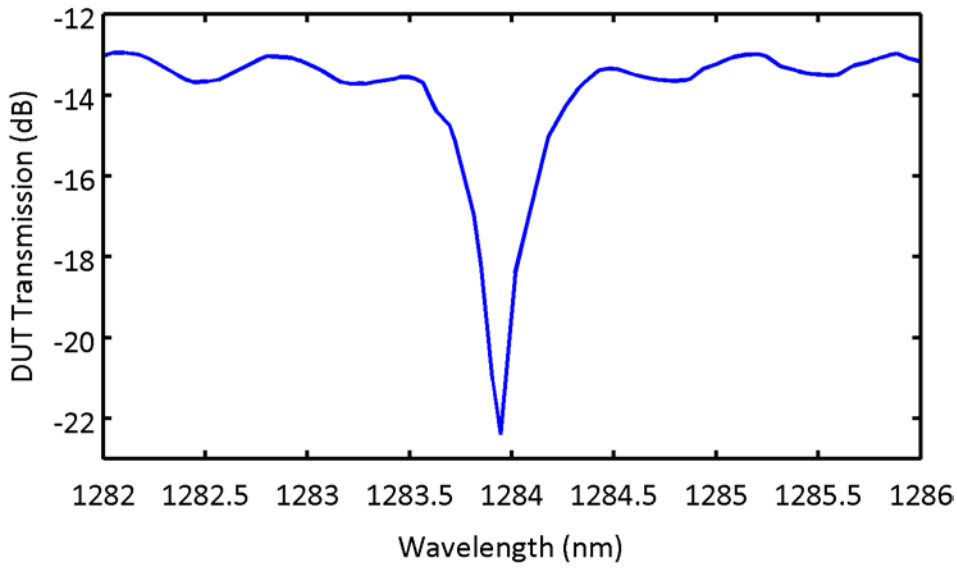


**Figure 5.20** Optical transmission function for the rib modulator design for the 1280nm wavelength range from as measured on EOS8.

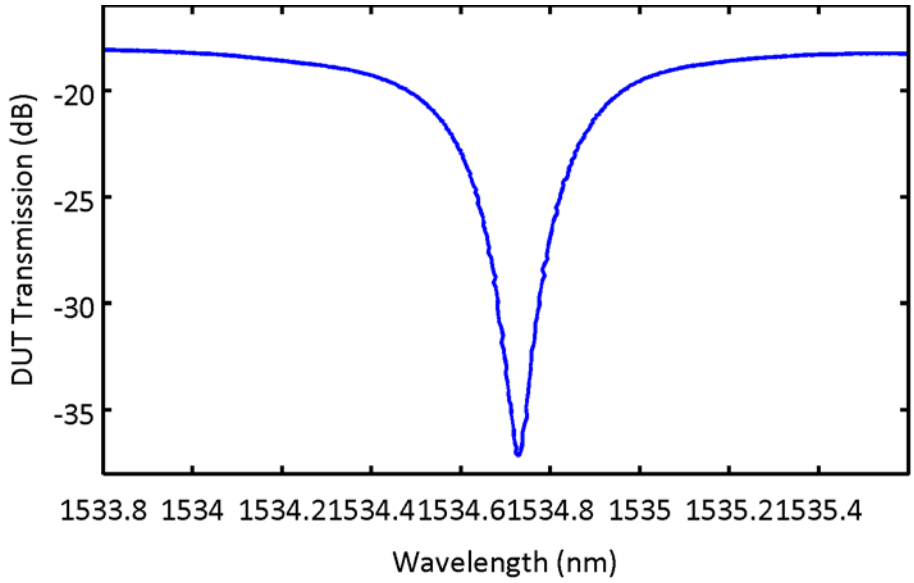


**Figure 5.21** Optical transmission function for the rib modulator design for the 1550nm wavelength range from as measured on EOS8.

Similar IV characteristics to the finger contact modulator were obtained in the rib geometry as shown in **Figure 5.22**. The continuous contacts create a significantly wider effective junction around the ring, but a longer intrinsic region of 1.6 μm was used in the fabricated ribs as opposed to the 444 nm intrinsic region of the finger contact geometry. Still, the wider width lowers the voltage required for 1 mA of current to 1 V. The measured series resistance was 47Ω.

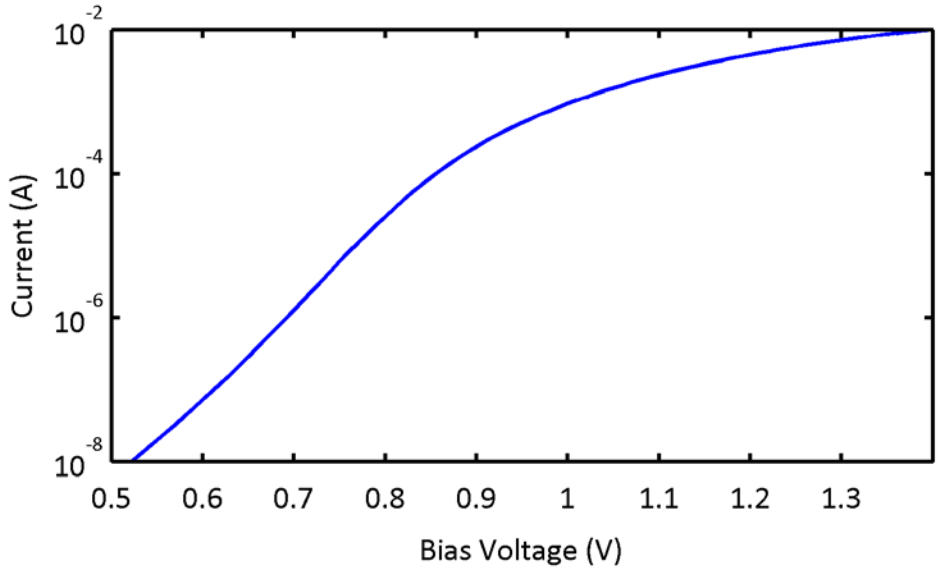


**Figure 5.22** Measured IV characteristics of rib modulator fabricated on EOS8.

### 5.3.4. High-Speed Measurements

Characterization the electro-optic behavior of the rib contact modulators was performed with high speed testing. Since the thermo-optic effect has an opposite refractive index change to the electro-optic effect, the self-heating of the modulators under forward bias complicates DC measurement. Although a shift in the electro-optic frequency direction is observed at low bias, the thermo-optic effect dominates at current levels required to shift the carrier-injection modulators by any significant fraction of the resonance width.

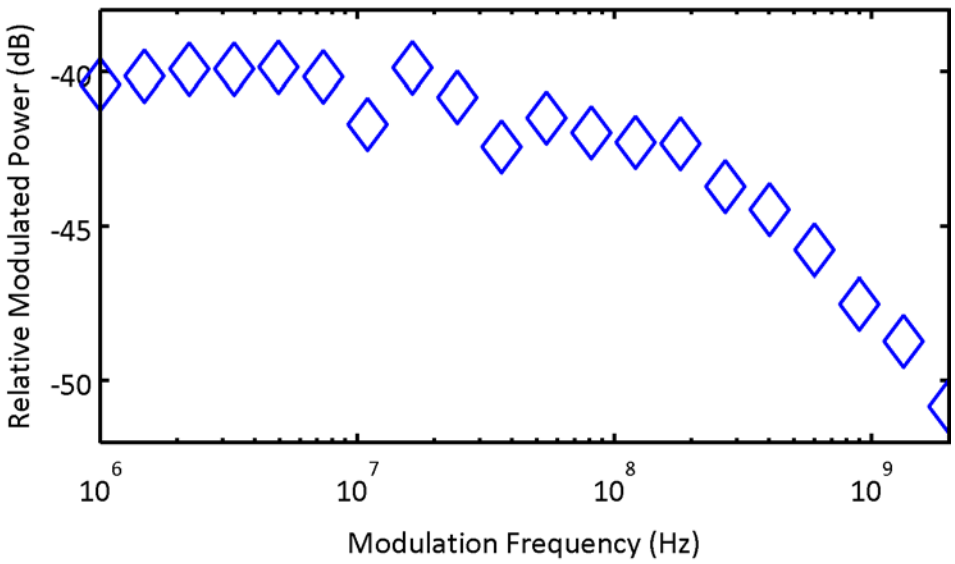


**Figure 5.23** Frequency response of a rib modulator fabricated on EOS8.

The first test performed on the rib contact modulators was to characterize the frequency response for a sinusoidal electrical input. After complete substrate removal with $XeF_2$, the released CMOS film of an EOS8 die was bonded to a 6H-SiC carrier with Norland Optical Adhesive 71 in the substrate transfer process that is further described in Appendix B. In addition to being directly connected to the integrated driver circuits as shown in **Figure 5.1**, modulators independently connected to ground-signal-ground (GSG) pads suitable for RF testing are included outside the circuit integration test regions of the die. The modulator was then simultaneously coupled to optically and contacted through the electrical pads with a Cascade Microtech i40-100-GSG probe in the vertically-coupled test setup described in Appendix A. The modulator DC-biased at 0.9V to place the modulator near diode

turn on through a Picosecond Pulse Labs Model 5542 bias tee. An input 8 dBm RF signal of variable frequency was then input to the AC-coupled port of the bias tee. The resulting modulated optical power was measured on an HP 70900B microwave spectrum analyzer with an internal optical frontend (HP 70810B) to produce the small signal response plot shown in **Figure 5.23**. The modulated optical power was consistent with a 10% modulation depth at 1 MHz.

The measured frequency response demonstrates an intrinsic device electrical bandwidth of approximately 300 MHz. This is consistent with an effective carrier lifetime of 2-3 ns in the intrinsic region of the p-i-n diode. The measured carrier lifetime is longer than the ~1 ns carrier lifetimes measured in past silicon photonic carrier injection modulators, presumably due to the high-quality surface passivation present in the CMOS electronics process [74].

Before testing with the integrated drive circuits, data transmission was first attempted using external test equipment through the microwave probe connections. The data source was an HP 71612B Bit Error Rate Tester (BERT). To create a suitable electrical input signal, the output of the BERT was first amplified using a JDSU H301 modulator driver and then attenuated using an HP 8494B stepped attenuator to achieve an appropriate signal level. The resulting electrical signal was then input to the AC-coupled port of the bias tee as in the frequency response measurement. The eye diagram was then measured on a HP 83480 Digital Communications Analyzer after being amplified by an erbium doped fiber amplifier (EDFA) and a spontaneous emission notch filer. A clean eye diagram for the finger-contact modulator was obtained for a 100 Mbps data rate as shown in **Figure 5.24**. The best measured extinction ratio for the finger-contact design was the 4.8 dB shown. Similar eye diagrams were also measured for the rib modulator. Measured eye diagrams for 100 Mbps and 600 Mbps operation are shown in **Figure 5.25** and **Figure 5.26** respectively. At 600 Mbps, the measured eye for the rib modulator driven by the external test equipment is mostly closed.

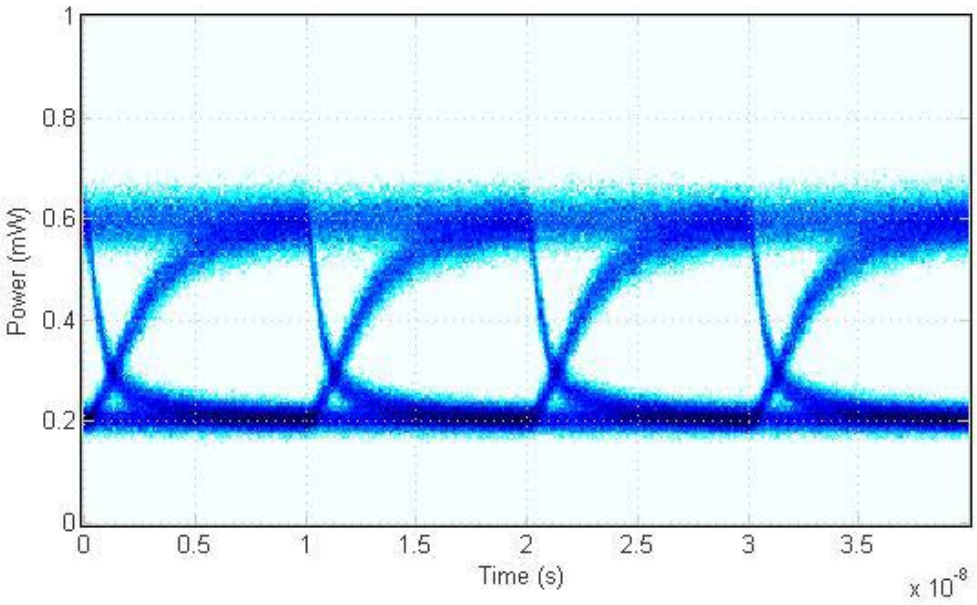


**Figure 5.24** 100 Mbps eye diagram for the EOS8 finger-contact modulator driven by external PRBS source.

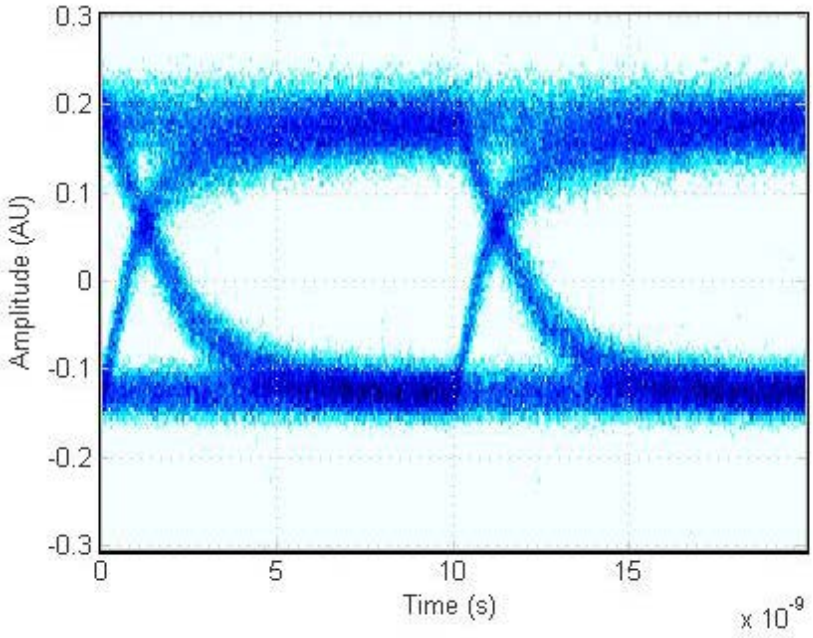


**Figure 5.25** 100 Mbps eye diagram for the EOS8 rib modulator driven by external PRBS source.

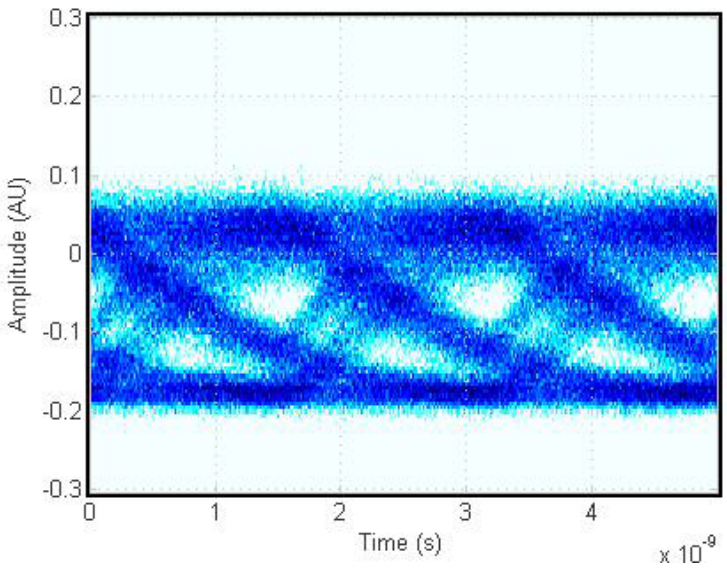


**Figure 5.26** 600 Mbps eye diagram for the EOS8 rib modulator driven by external PRBS.

### 5.3.5. Integrated Driver Results

After measuring the fabricated modulators with external test equipment, the modulators were tested with the integrated drivers on a fully-packaged substrate-transferred die. The packaged die was interfaced electrically with a FPGA while mounted in a circuit board under a similar vertically-coupled fiber test station in Vladimir Stojanovic's research group. The integrated circuit drivers are level-shifting push-pull inverter drivers that were designed by Benjamin Moss [240]. The output optical eye diagrams were measured at 500 Mbps as shown in **Figure 5.27** for the rib modulators. Keeping the supply voltage for the driver stage limited to 1.5 V did not produce open eye diagrams for the finger-contact modulators.

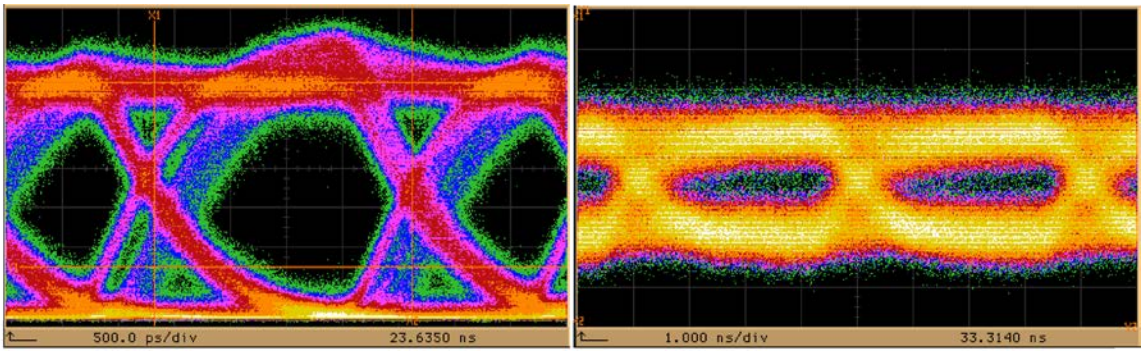


**Figure 5.27** Measured rib modulator eye diagrams at 1550nm (a) and 1280nm (b) driven by the integrated driver circuits at 500 Mbps.

The highest data rate that resulted in open eye diagrams was 600 Mbps as shown in **Figure 5.28**. An extinction ratio of ~10 dB was achieved at this data rate. The maximum achievable data rate of roughly twice the measured 3 dB bandwidth matched expectations for NRZ data transmission. Higher data rate transmission experiments resulted in closed eye diagrams. Significant eye opening reduction is visible at 800 Mbps as shown in **Figure 5.29**(a). An example eye diagram is shown in **Figure 5.29**(b) for a 1 Gbps data rate with a nearly closed eye.

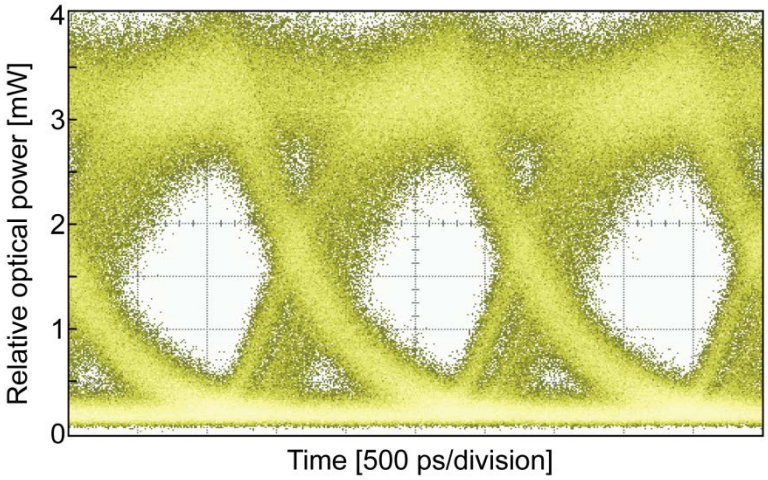


**Figure 5.28** Measured 1550nm rib modulator eye diagram driven by the integrated driver circuits at 600 Mbps.

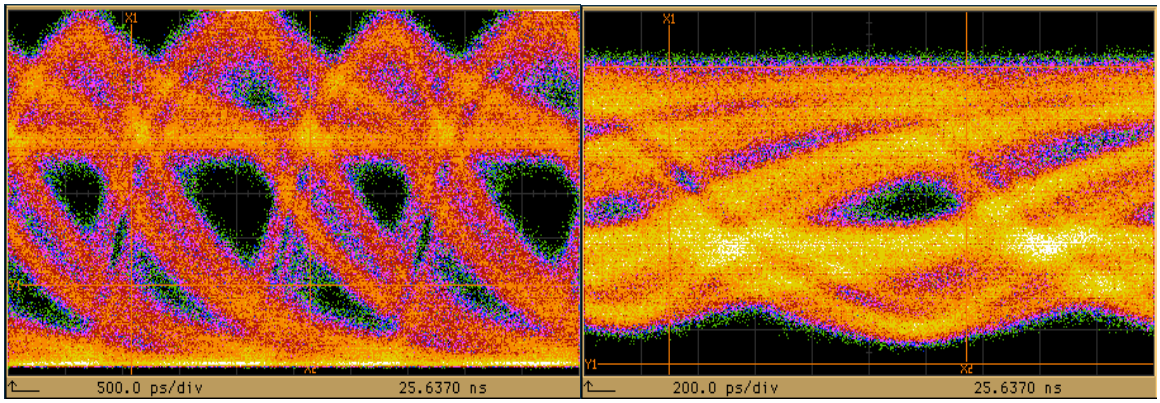


**Figure 5.29** Measured 1550nm rib modulator eye diagram driven by the integrated driver circuits at (a) 800 Mbps and (b) 1 Gbps.

The separate supply for the integrated drivers enabled an estimate for the total energy efficiency of the circuit operation. As measured by the current draw for a

sustained '1', the rib modulator was operated at ~1.5 mA forward bias for the included eye diagrams. Extracting this static current dissipation from the dynamically measured power dissipation, a total pre-driver switching energy of ~130 fJ/bit was extracted. For the 600 Mbps data transmission example, the resulting total transmitter energy efficiency was 2 pJ/bit. To first order, if higher bandwidths were achievable in a similar modulator design, the same static current draw would be divided by a significantly higher bit rate. By simply dividing the static power draw by a 10 Gbps data rate and including the same pre-driver switching energy, a crudely approximated power dissipation of ~250 fJ/bit could be estimated. The actual power dissipation, however, would have to include the power required for the pre-emphasis drivers required to increase the data rate.

## 5.4. DRAM Modulators

At the time of this writing, no experimental DRAM modulator results are available for inclusion in this thesis. The process flexibility of the DRAM photonic integration platform that is under development does include a partial polysilicon etch step to form rib waveguide geometries that are very similar to extensive prior literature results for silicon photonic modulators. The first process development vehicle, however, did not include a partial etch step. As such, a new variant of the finger contact geometry was created as an alternative option to modulator designs that were included by other team members including Jeff Shainline, Milos Popovic and Erman Timurdogan.

The modified slab contact design shown in has been redesigned for a polysilicon core thickness of 225nm. Due to exact dimension and material index uncertainty of the early stage of process development and the large real estate available on the ~4 $cm^2$ dedicated process masks, 96 modulator variants were included to vary 16 junction designs across 6 coupling coefficients. Although the partial etch is expected to be included in the nominal process flow of the DRAM photonic integration platform, the high cost sensitivity of DRAM manufacture requires continued effort to determine the absolute minimum number of required processing steps to achieve suitable photonic device performance.

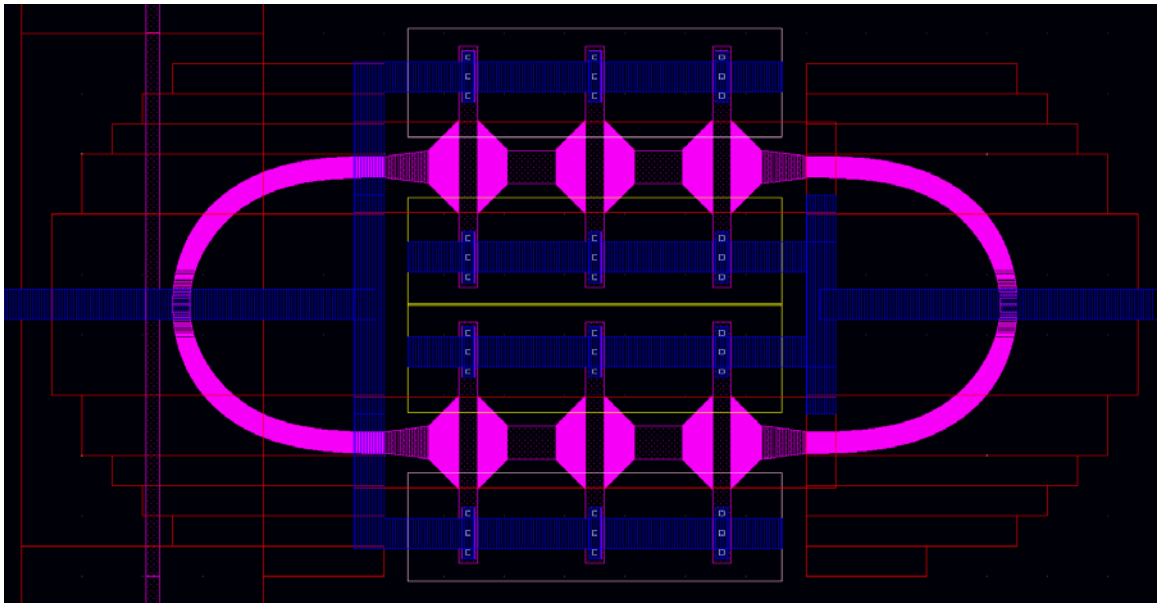

**Figure 5.30** Slab contact modulator designed for the DRAM integration process.

A key area of design flexibility within the DRAM integration platform is the ability to custom tailor the implant levels for the photonic devices. Several different implant levels that exist for transistor formation may be tweaked to simultaneously optimize photonic device performance. An extremely desirable doping configuration that will aid in the demonstration of a depletion modulator within this technology is a vertical pn-junction. Since the horizontal depletion region that can be engineered to be in the center of the waveguide to provide high overlap waveguide mode, the effective confinement factor of the modulated carrier region is significantly higher than in the case of a lateral junction [92]. Given the relevant process information from the process engineers, I designed the vertical junction for depletion modulators shown in **Figure 5.31**. The doping concentration of each side of the junction is approximately 2 x $10^{18}$ $cm^{-3}$. This doping concentration enables a depletion region that is 50nm thick at -1V bias and 70nm thick at the maximum -3V bias of the integrated circuits. If the device is successfully fabricated, the design would enable 6 dB extinction ratio with a 1 dB insertion loss in carrier depletion operation. In contrast to the carrier injection designs that have been fabricated to this point, a carrier depletion modulator would enable circuit-limited power dissipation and bandwidth of approximately 50 fJ/bit at data bandwidths of 5-10 Gb/s as discussed in Chapter 2.

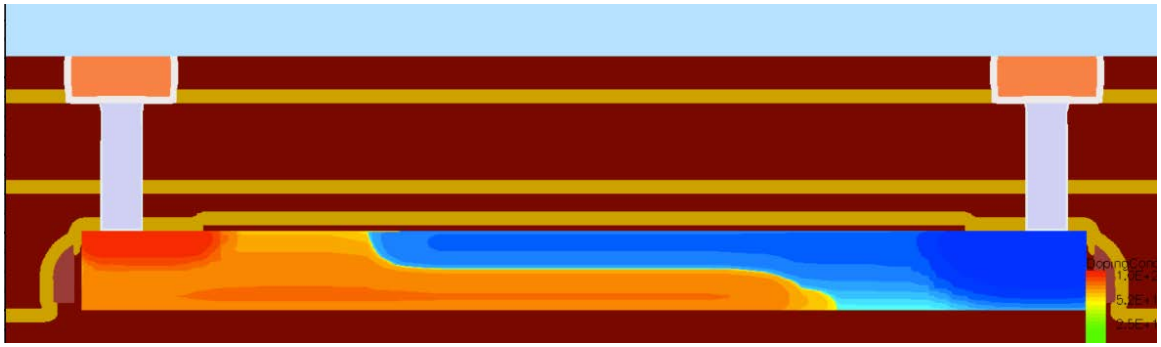

**Figure 5.31** Vertical pn-junction doping profiles designed for depletion-mode modulators within the DRAM photonic integration process.

## 5.5. CMOS Silicon Germanium Photodiodes

For zero-change integration in standard CMOS processes, the only available material for linear absorption of light that is not absorbed in silicon is the p-doped silicon germanium that is present in the process for PFET strain engineering. As a result, most traditional photodetector geometries such as the p-i-n diode discussed in Chapter 1 are not achievable given the existing processing constraints. Instead, a new device geometry has been proposed. Initial proof-of-concept devices have been fabricated and measured but exhibit low external quantum efficiencies of ~1%.

### 5.5.1. CMOS Process Constraints and Material Parameters

Scaled CMOS processes include SiGe for strain engineering in the p-type transistor source/drains. Although the more common silicon photonic detector material of pure germanium absent from the process, the lower bandgap of the SiGe layer enables the formation of a photodetector at wavelengths where the silicon waveguides are transparent. The use of this existing layer for photodiode design imposes two additional constraints. First, since the germanium mole fraction of this SiGe layer only ranges between 20% and 30%, the comparatively large bandgap requires much shorter wavelength operation than typical telecommunication devices. Second, since the SiGe layer is used only for p-type transistors, it is therefore grown with *in-situ* p-type doping. This is in contrast to the silicon layers that are used for a wide variety of p-type and n-type devices and therefore start the process with doping densities below $1 \times 10^{16}$ $cm^{-3}$.

The first constraint imposes more system-level implications than device specific ones. If the system application allows for free design of the transmit side of the photonic link, shifting to the wavelength region around 1200 nm is a viable option. This has the supplemental benefits of dramatically reducing the bend radius for the thin integrated waveguides and increasing the laser efficiency as discussed further in Chapter 2. Alternatively, further independent research into novel photodiodes based upon either two-photon absorption or defect-state absorption [13, 14] can enable standard telecommunication wavelength operation.

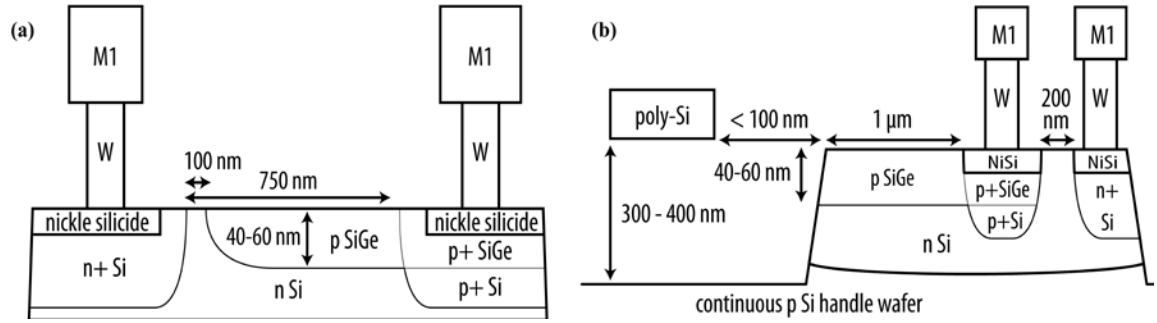


**Figure 5.32** Cross-sections of the silicon germanium nanophotovoltaic structure designed the SOI-CMOS (a) and bulk-CMOS (b) processes.

The second constraint, in-situ p-type doping, requires a novel diode design. Since the absorbing layer is heavily doped, traditional p-i-n diode designs are not achievable. Instead, a pn-junction photodetector design must be constructed with electrical implications that will be discussed in subsequent sections. For SOI platform integration, a high efficiency diode can be constructed using the cross section shown in **Figure 5.32**(a) and similar contact geometries to the slab modulator. The overall efficiency of such a device is then limited by free carrier absorption due to the heavy doping and surface recombination at the SiGe/Oxide interface. In bulk-CMOS platforms, the SiGe layer is grown on the handle wafer instead of a waveguide layer that is suitable for local substrate removal. Therefore, lateral optical coupling from the poly-Si waveguide must occur in regions where the silicon substrate is not removed as shown in **Figure 5.32**(b). This further limits efficiency by optical coupling losses to the substrate in these regions.

There are several variables that affect the relevant material parameters for device design. In addition to the exact mole fraction being unknown, the levels of doping and strain present in the process significantly affect the bandgap and therefore absorption coefficient of the silicon germanium material. In the SOI

platform, the isolation of the silicon body layer on which the silicon germanium is epitaxially grown from the handle silicon wafer by the buried oxide layer allows the formation of silicon-germanium-on-silicon waveguides. Suitable test structures of 2mm silicon waveguides with 0µm, 10µm, 50µm and 100 µm silicon germanium sections were fabricated on the EOS8 SOI-CMOS test chip. Optical transmission measurements shown in **Figure 5.33** showed that the effective absorption increase possible in the silicon-germanium-on-silicon waveguide relative to the normal silicon waveguide is approximately 0.5 dB/µm from 1200nm to 1240nm. This measurement therefore predicts that a silicon germanium slab photodetector may absorb 90% of the input light with a 20 µm propagation length. To perform these measurements in the 1160nm to 1260nm spectral window where commercial tunable lasers were not available at the start of this work, a new tunable laser was constructed using an InGaAsP MQW gain media purchased from Covega and described in Appendix C.

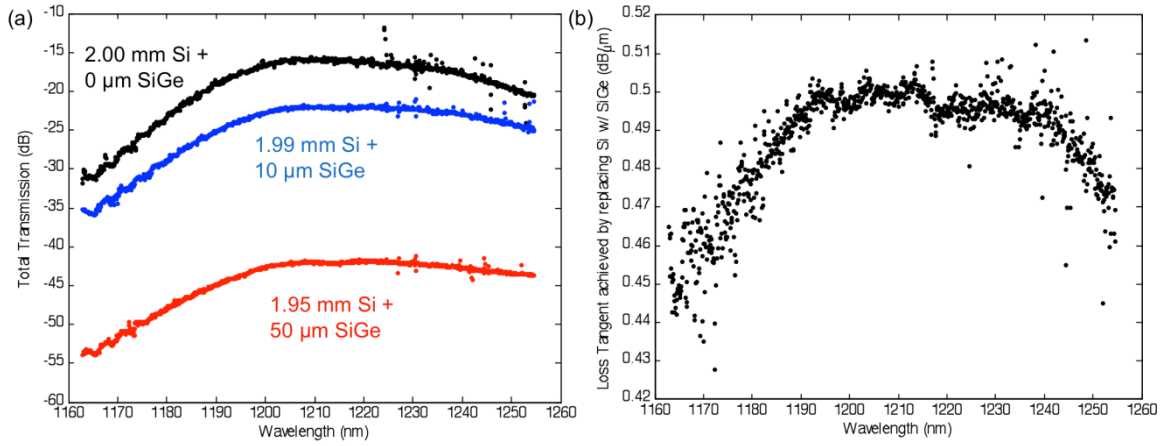


**Figure 5.33** Measured silicon germanium loss tangent from 1160nm to 1260nm from EOS8.

### 5.5.2. Electrical Junction Design

To utilize the in-process silicon germanium necessitates the alternative diode design based on a p-n junction must be utilized. Instead being able to rely on carrier generation in the high electric field intrinsic region to immediately generate current due to the dipole formed by separating the electrons and holes, generation in a p-n junction largely occurs in the quasi-neutral regions where the electric field is negligibly small. Current is generated from the absorbed light only when the generated carriers are separated by diffusing to the depletion region where the

electric field can cause carrier separation. This regime of operation is most commonly associated with silicon solar cells where pn-junction diodes are the dominant device geometry. Two major concerns are worth considering to assess the suitability of a pn-junction based detector for high-speed data receivers: efficiency and bandwidth.

To estimate the efficiency of the pn-junction, the relative rate of the process that contributes to current, generated minority carrier diffusion to the depletion region edge, must be compared to the recombination within the quasi-neutral region due to the bulk material, surfaces and device contacts. The diffusion time constants for each carrier polarity can be estimated from the size of the quasineutral region L and the diffusivities $D_n$ and $D_p$:

$$\tau_n^{diffusion} = \frac{4L^2}{\pi^2 D_n}; \quad \tau_p^{diffusion} = \frac{4L^2}{\pi^2 D_p} \tag{5.1}$$

To estimate transport parameters, diffusivities of 5.1 $cm^2/s$ and 3.7 $cm^2/s$ for electrons and holes in 1 x $10^{19}$ $cm^{-3}$ doped, 25% silicon germanium were calculated using carrier mobilities of 197 $cm^2/V{\cdot}s$ and 142 $cm^2/V{\cdot}s$ respectively [241]. The predicted diffusion time constants for a 160 nm quasineutral region are 20 ps and 28 ps for electrons and holes respectively. Since these time constants are nearly two orders of magnitude faster than the nanosecond-scale expected carrier lifetimes including bulk and surface recombination, baseline analytical predictions indicate that it is possible to create an efficient pn-junction detector in small quasineutral regions. The contribution of recombination at the device contact is highly dependent on the specific photodiode geometry and is best simulated through a finite element model as discussed in later sections.

The bandwidth of the pn-junction detectors is also set by the diffusion time constant. Since the observed photocurrent occurs when the generated minority carrier transits the depletion region, a photogenerated carrier pulse in the quasineutral region is filtered by the diffusion time constant. However, as seen in the efficiency analysis, the diffusion time constants are much shorter than the 100 ps relevant bit time for 10 Gb/s data.

The promising efficiency and bandwidth characteristics for the silicon germanium pn-junction based detectors are enabled by the small size of the device geometries that may be fabricated in a deeply-scaled CMOS process. If the size of the quasineutral region was increased to a 1 μm feature size, the hole diffusion time constant would increase to over 1 ns. Not only would the slower time constant

reduce the achievable device bandwidth below suitability for 10 Gb/s operation, but also it would result in reduced efficiency. Since the carrier lifetimes can be expected to be on the order of 1 ns, if the diffusion lifetime was on the same scale as the minority carrier lifetime, the photogenerated minority carriers may recombine before contributing to photocurrent by diffusing to the depletion region edge.

### 5.5.3. Bulk-CMOS Detectors

Full device design simulations were performed for the proposed silicon germanium photodiode geometries within the 65nm bulk-CMOS process (EOS1). The actual fabrication process used for EOS1 didn't include the silicon germanium formation steps. Similar designs to the simulated structures were fabricated on the following 28nm bulk-CMOS test chip (EOS2). Due to a change of fabrication facilities, however, the exact process and material parameters used for the silicon germanium layer remain unknown. The switch to the thin-SOI CMOS process and early state of the post-processing and test facilities resulted in most parameters going unmeasured for these simulated, bulk-CMOS devices. The lack of *a priori* process knowledge of the thin-SOI silicon germanium formation prevented similar device simulation work for these detectors that have been more fully characterized. As such, the device simulations presented in this section will serve only as a remaining proposal that may be useful for future work and aid in the understanding of the function of the fabricated detectors.

#### *5.5.3.1. Optical Design and Simulation*

The bulk-CMOS process does not include a single-crystalline waveguide. Since the silicon germanium may only be grown on the single-crystalline silicon layer, most standard silicon photonic device geometries are forbidden within the standard process. The light must somehow be then coupled from the polysilicon waveguide to the silicon germanium layer grown on the single crystalline silicon substrate. The proposed photodiode geometry to accomplish such coupling is shown in **Figure 5.34**. The silicon germanium in the processes of interest are typically recessed into the silicon substrate such that it in a plane below the polysilicon layer. This difference makes a simple butt-coupled geometry inefficient since the silicon germanium layer has low spatial overlap with the polysilicon waveguide mode. Since the silicon germanium formation step happens after the polysilicon deposition, it is also impossible to fabricate device geometries where the silicon germanium is formed

directly under the polysilicon layer. Instead, the proposed device geometry relies on lateral optical coupling between the polysilicon waveguide and neighboring silicon germanium ribs.

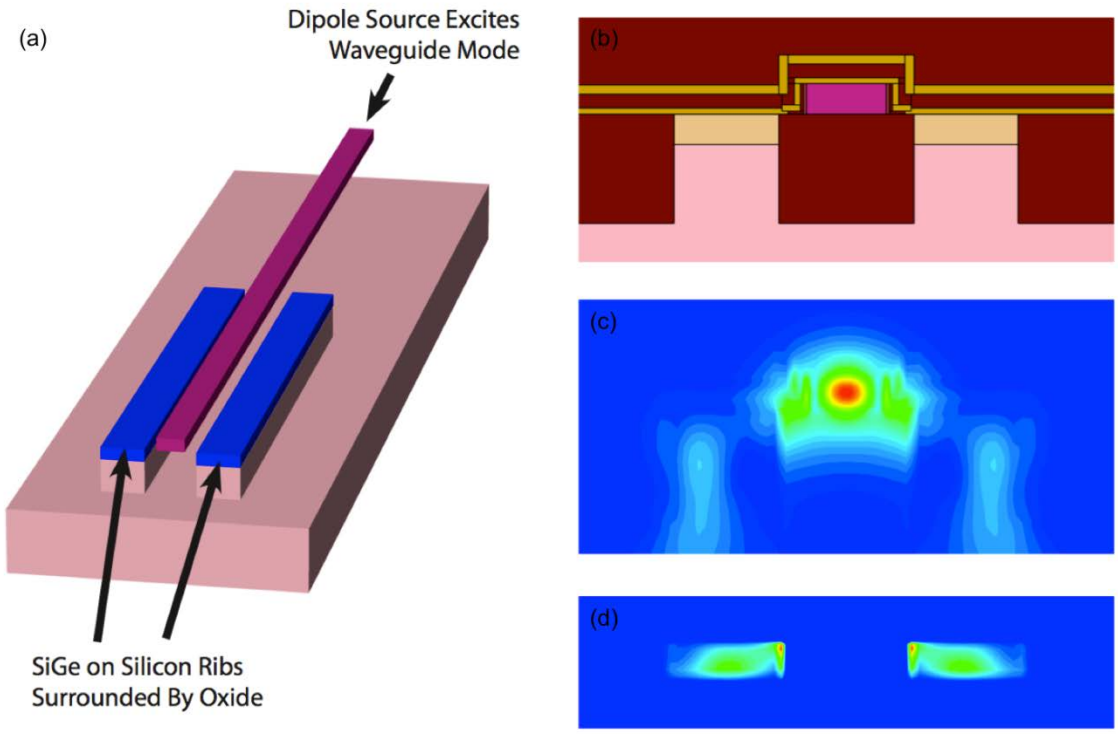


**Figure 5.34** (a) 3D FDTD simulation domain. (b) Cross section in the center of the diode. (c) Optical intensity plot showing polysilicon waveguide mode coupling to the silicon germanium ribs. (d) Optical generation profile within the silicon germanium regions.

The optical design of the silicon germanium detector structure was performed using the 3D FDTD solver within the Sentaurus TCAD suite. The simulation domain without the surrounding dielectric layers is shown in **Figure 5.34** (a). The polysilicon waveguide is excited by a TE dipole source 10 μm from the silicon germanium ribs. The FDTD solver in the Sentaurus tool suite does not include the capability to launch a pre-simulated waveguide mode. Instead, a ~5 μm waveguide length is required to form the mode before measuring the launched waveguide mode power. The resulting waveguide mode then couples into the detector region with the cross-section shown in **Figure 5.34** (b). The light that couples into the silicon germanium rib, shown in **Figure 5.34** (c), is either absorbed by the silicon germanium or couples to the silicon germanium substrate. Since the silicon substrate can't be locally removed in the silicon germanium regions, this leakage

source is a major efficiency limitation to such device geometries. The optical absorption within the silicon germanium regions results in the electron-hole generation profile shown in **Figure 5.34** (d). For various silicon germanium layer thicknesses of interest for the bulk-CMOS processes under study, simulated generation efficiencies, i.e. ratio of electron-hole pairs to incident photons in the waveguide mode, of between 7% and 15% were achievable. For these calculations a silicon germanium absorption coefficient of 500 $cm^{-1}$ was used based upon literature review [241, 242]. If the measured absorption in the silicon germanium from the IBM process can be entirely attributed to the band-to-band absorption, the increased absorption coefficient of ~1200 $cm^{-1}$ would result in a significantly increased predicted efficiency. If an 80 nm silicon germanium was used, which is thicker than what is present in any bulk-CMOS process that has been considered over the course of this thesis, the simulated generation efficiency was 38%. The relatively low efficiencies were dominated by the substrate leakage. If the detectors were made longer, no increases in efficiency were possible as no light remained in the waveguide mode. Further efficiency improvements would require a thicker oxide trench isolation in the process or advanced substrate-removal post-processing that could be used in the silicon germanium regions.

#### *5.5.3.2. Electrical Design and Simulation*

The electrical junction must then be designed to comply with the optimal optical design. In the case of the bulk-CMOS silicon germanium detectors, the fingered contact geometry shown in **Figure 5.35** was designed. The lateral finger design first served to place the optically lossy contact metals as far as possible from the optical mode to minimize extrinsic absorption. The width of each p-type and n-type silicon germanium finger was chosen to be 160nm to correspond to the calculated ~20 ps diffusion time constants calculated in Section 5.5.2. Not shown in **Figure 5.35** is the substrate contact to the n-well in which the pn-junction silicon germanium detector is fabricated. The electrode of the n-well contact is connected to the n-type silicon germanium detector at the input to the receiver circuit.

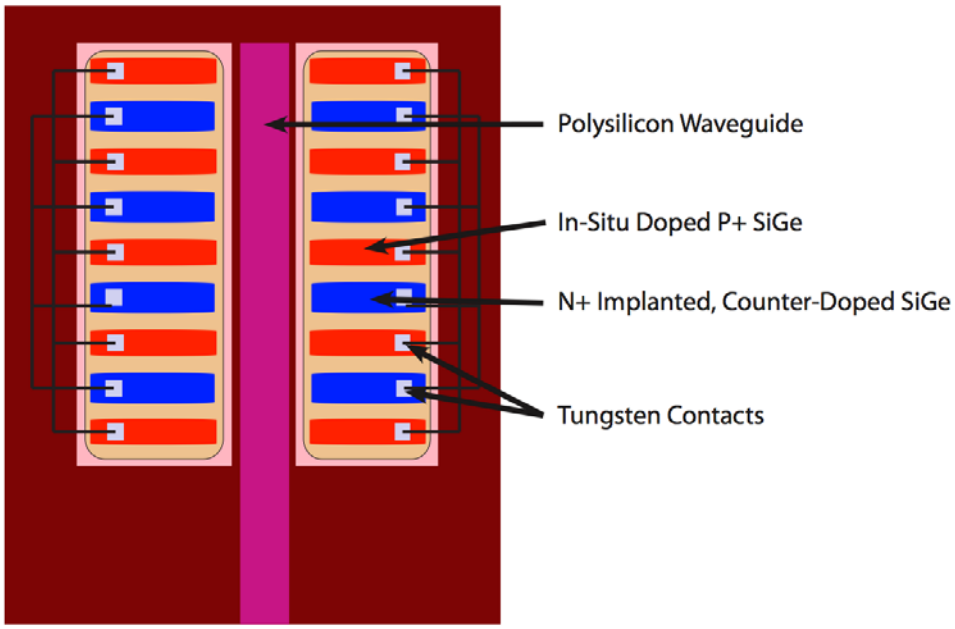


**Figure 5.35** Top-view cartoon of interdigitated silicon germanium photodetector.

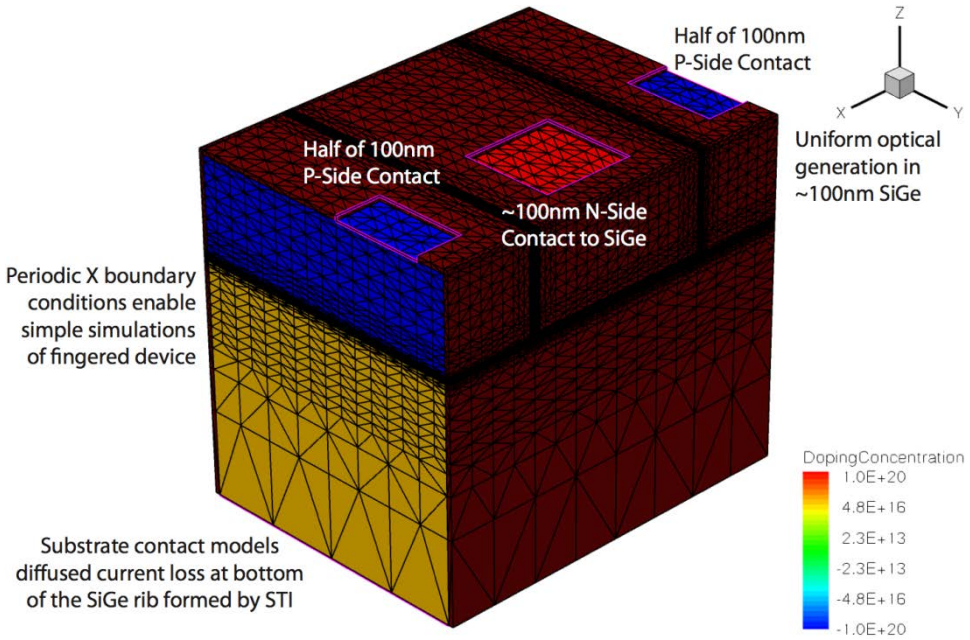


**Figure 5.36** Finite element model constructed in Synopsys Sentaurus for the bulk-CMOS silicon germanium fingered pn-junction photodetector.

To accurately model the internal quantum efficiency and the device bandwidth, the finite element model shown in **Figure 5.36** was constructed in Synopsys Sentaurus. Instead of simulating the entire structure, a single finger cell with periodic boundary conditions was used for the simulation domain. It is important to

note that the boundaries of the computational domain were placed at the center of a quasineutral region instead of the depletion region at the p-type / n-type finger boundaries. Although the computational domains are logically equivalent, significant computational convergence issues can occur if the high field regions of the depletion regions coincide with the periodic boundary condition cells.

The constructed simulation domain can then be placed under optical illumination to assess the bandwidth and quantum efficiency. If a full computational domain was constructed, it would be possible to directly load the simulated carrier generation profiles from the FDTD simulation to source the electrical simulation. Due to the immaturity of the devices and process uncertainty, such an involved simulation was not performed for this work. Instead, a single point source was placed in the center of the n-type finger where the lower minority carrier hole mobility should result in the longer diffusion time constant. The location of the generation source and the resulting spatial minority carrier profiles are shown in **Figure 5.37**. A 50 ps temporal envelope was applied to the optical generation source to simulate a 20 Gb/s NRZ data stream, which is double the system target data rate as discussed in Chapter 2. The resulting time domain electrical waveforms for each contact are plotted in **Figure 5.38**. The magnitude of the on-state verifies a 76% internal quantum efficiency. The fast rise and fall times demonstrate the design's suitability for use as a high speed data receiver.

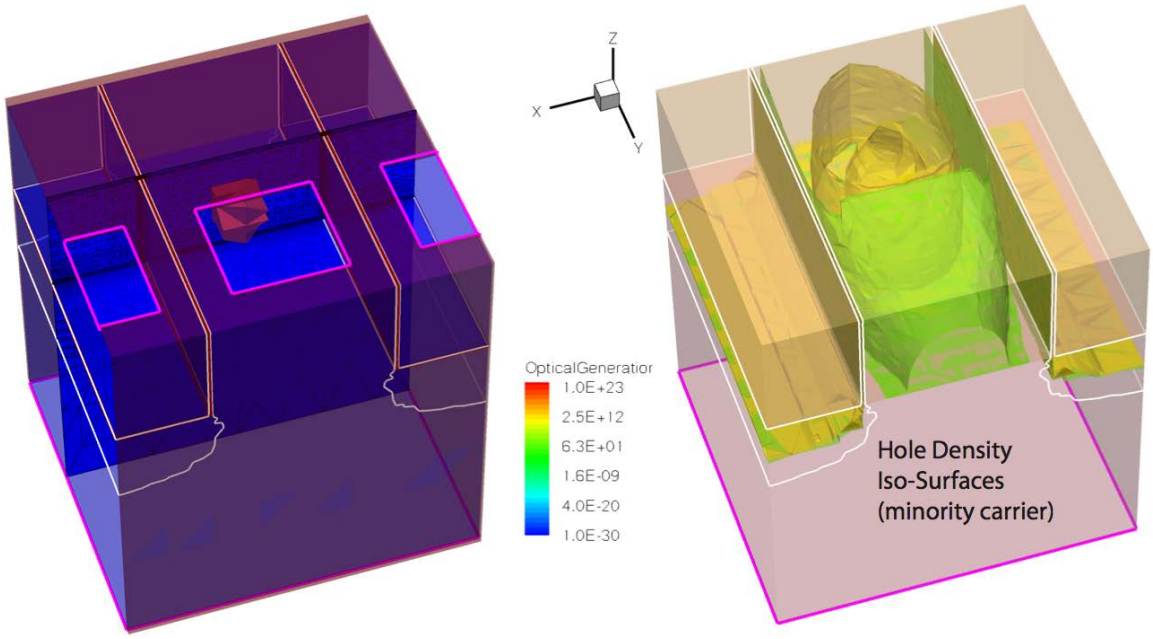


**Figure 5.37** Optical generation spatial profile used for the pulse response simulation and resulting minority carrier spatial contours.

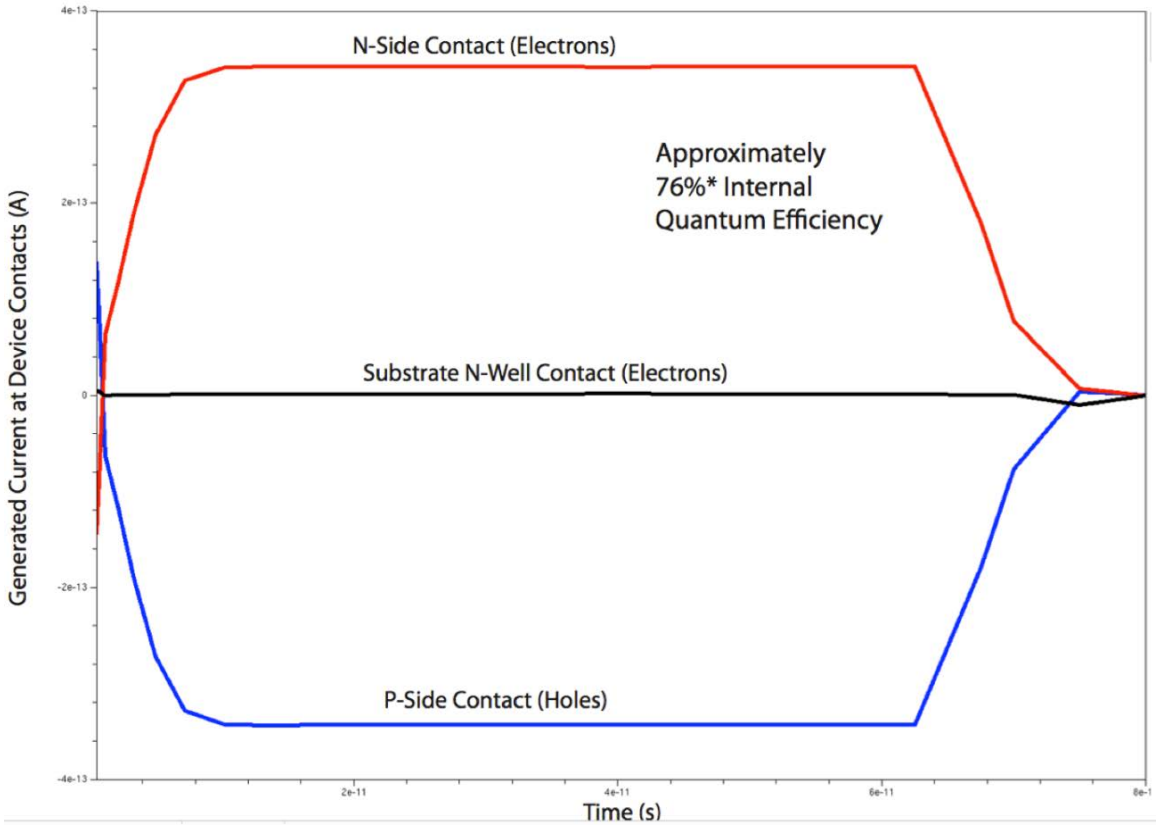


**Figure 5.38** Simulated electrical response to a 50 ps optical generation pulse for a pn-junction formed in the silicon germanium layer.

Since the internal time constant is several-fold faster than the effective carrier lifetimes of 1 ns used in the simulation, the incomplete collection of photogenerated carriers is dominated by minority carrier recombination at the device contact. The effect of the contact recombination is observable in the minority carrier spatial contours shown in **Figure 5.37**. The diffusion of the minority carrier species to the contacts can be effectively blocked by introducing a higher doping density near the contacts relative to the absorbing region of the finger to introduce a potential barrier. Similar techniques have been utilized to increase the internal quantum efficiency of silicon solar cells that operate on a similar pn-junction principle [243].

#### *5.5.3.3. Fabrication and Electrical Measurements*

The fabricated bulk-CMOS photodetectors have not been fully characterized as a result of the shift in focus to the thin-SOI CMOS process over the course of this thesis. The fabricated device is shown in **Figure 5.39**(a) after destructive dry etching of the fabricated die. The electrical measurements, shown in **Figure 5.39**(b), did

reveal that the opening width of the silicon germanium epitaxial window may be an important design parameter. Since the silicon germanium is grown in a strained state as a result of the lattice mismatch with the silicon crystal template, it is possible to observe strain relaxation for wide silicon germanium epitaxial openings. This can be understood as the strain potential energy of the film becoming larger than the activation energy of defect formation [244]. This parameter must be determined for every CMOS process as interest as the exact processing conditions will greatly affect the presence of a threshold width. When the transition to a defected film occurs, the diode rectification observed in high film quality samples disappears. For the studied 28nm bulk-CMOS process, the transition was observed for a 1 μm epitaxial window opening width.

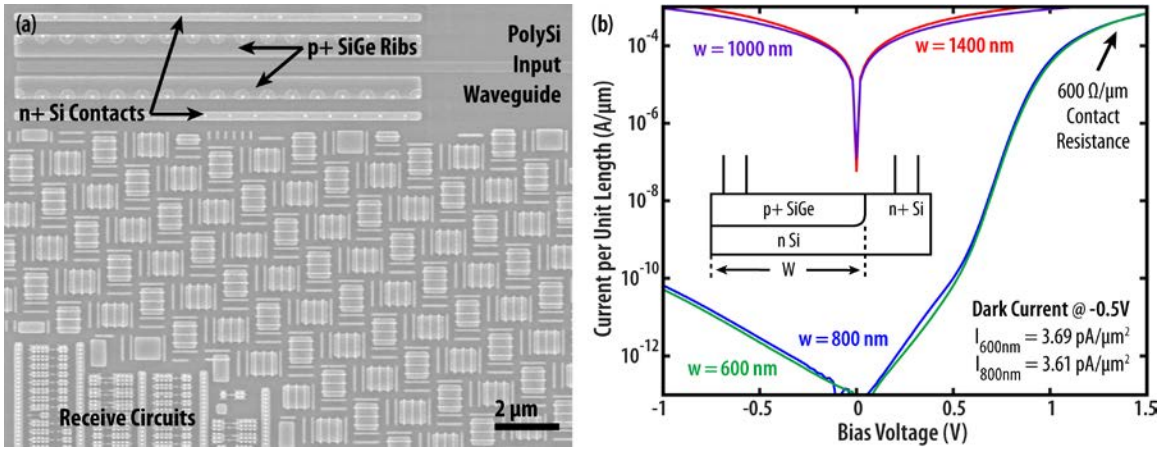


**Figure 5.39** (a) Integrated waveguide SiGe photodiode SEM and (b) measured IV characteristics as a function of epitaxial window opening width

### 5.5.4. Thin-SOI CMOS Detectors

In the silicon-on-insulator process, incomplete process knowledge has prohibited the scope of simulations performed for the bulk-CMOS devices to this point. Instead, various geometries of first generation detectors and process test structures were included on 45nm SOI-CMOS test chips to provide initial device results and extract relevant parameters.

Since the silicon germanium is grown on the body silicon layer that is separated from the silicon substrate by the buried oxide layer, it is possible to fully remove the substrate under the detector regions without damaging the silicon germanium layer. This has the advantage of eliminating the substrate optical leakage source that was shown to have a large impact on the simulated device performance in the bulk-

CMOS FDTD simulations. Additionally, instead of requiring a lateral optical coupling between the silicon germanium region and input polysilicon waveguide, the regrowth of the silicon germanium into the body silicon waveguide provides a nearly ideal butt coupling.

#### *5.5.4.1. Resonant Cavity Designs*

The first optical detector design possibility for the thin-SOI process is to introduce a silicon germanium absorption region into the contact regions of a ring resonator using the slab contact geometry developed for the modulators in Section 5.3.2. This cavity design that is shown in **Figure 5.40** has the advantage of enabling high optical mode overlap with the intended silicon germanium absorption region while minimizing the optical loss in the contact metallization. The chief disadvantage is that the path length per round trip in the silicon germanium regions is limited to the width of the contact region of approximately 1 μm total. Using the absorption measured in the stand alone silicon germanium test structures, the silicon germanium is expected to account for 0.5 dB round trip absorption. For comparison, the two slab contacts are expected to account for 0.9 dB total round trip extrinsic loss. The fraction of the optical power absorbed by the silicon germanium is therefore expected to be limited to 39% of the power coupled into the resonator.

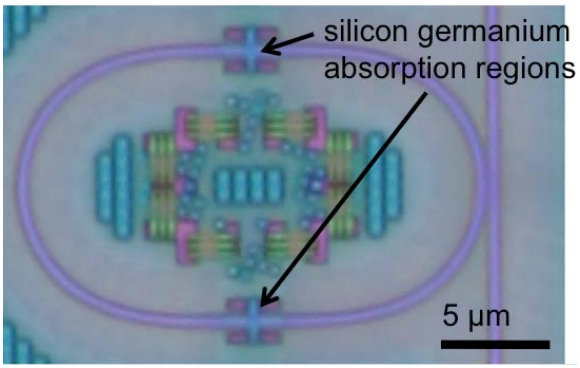


**Figure 5.40** Optical micrograph of the resonant silicon germanium photodetectors fabricated in EOS8 (IBM 45nm thin-SOI CMOS).

Since no drop port was included to reduce extrinsic round-trip loss in the resonator, the through coupling coefficient is required to match the cavity loss to achieve critical coupling. Three coupling strength designs were included on EOS8/EOS10 tapeouts. The 100 nm coupling gap design with nominal bus and ring

waveguide widths of 380 nm yielded the highest on-resonance extinction and therefore demonstrated the closest to critical coupling for the fabricated devices. The resulting optical transmission and photocurrent spectra for such a resonant silicon germanium detector is shown in **Figure 5.41**. Dividing the generated photocurrent by the incident optical power that is estimated to be coupled into the through waveguide yields a measured responsivity of ~8 mA/W or equivalently an external quantum efficiency of ~0.8%.

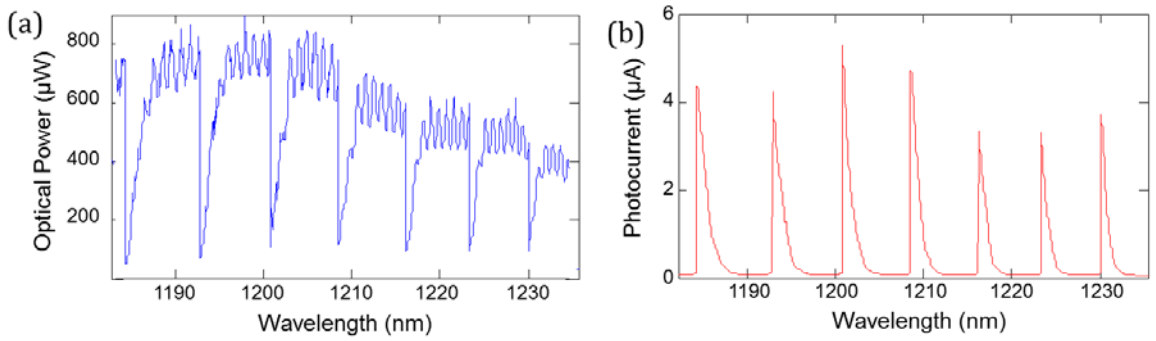


**Figure 5.41** (a) Optical transmission of the resonant silicon germanium photodetector normalized to in-waveguide power. (b) Measured resulting photocurrent from the optical excitation showing ~0.8% external quantum efficiency.

Although relatively low efficiency is expected from the extrinsic optical contact loss, it does not fully explain the low value measured. Instead of solely being limited by the absorbed power in the silicon germanium layer, it is likely that the internal conversion of generated electron-hole pairs to photocurrent, i.e. the internal quantum efficiency, is also significantly lower than ideal. The measured IV-characteristics of the detector do not look like those of a properly functioning diode suitable of efficient rectification of the photogenerated electron-hole pairs as shown in **Figure 5.42**. The weak forward current is limited by a large series resistance of 1.6 kΩ and is not necessarily an indication of poor diode performance. The major non-ideality evident in the IV curve is the high leakage current. Often, high leakage current is an indication of poor material quality due to defect states that would negatively impact the internal quantum efficiency of the diode. The various explanations for reduced internal quantum efficiency will be discussed in the context of the slab photodetectors in the following section. Although modifications to the device geometry may fix this problem, the majority of future device design will focus on slab geometries.

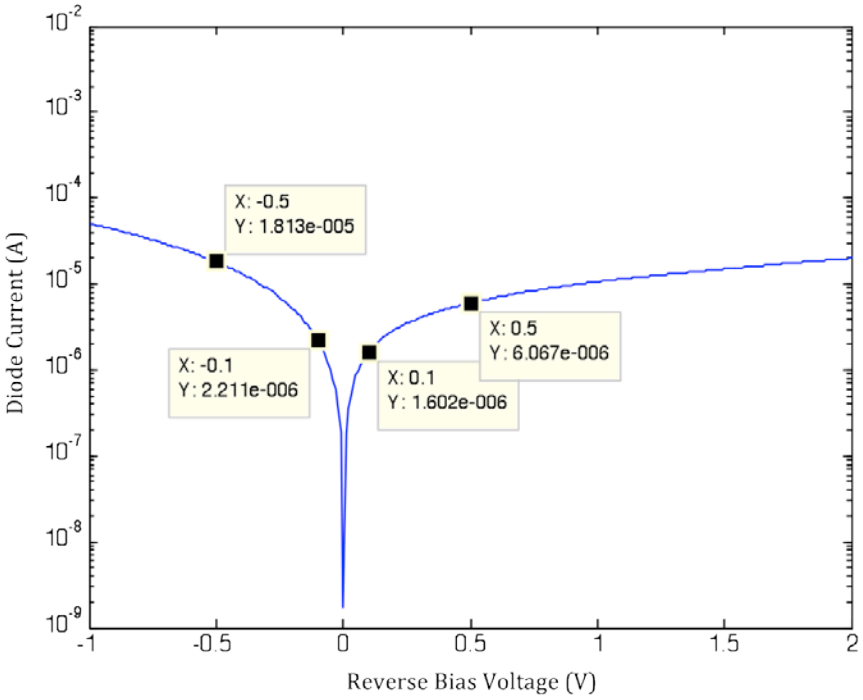


**Figure 5.42** Measured IV characteristics for a resonant cavity photodetector fabricated in the EOS8 test chip (IBM 45nm SOI-CMOS).

### *5.5.4.2. Single-Pass Detector Designs*

The alternate design that was fabricated on the intial test chips was a single-pass "slab" design as opposed to the resonant cavity. It is preferable to fabricate a broadband detector such that there is not an additional wavelength specific element that must be tuned to the data wavelength. Additionally, with the relatively high measured absorption of 0.5 dB/μm, a reasonable absorption length of 20 μm should be able to absorb 90% of the incident light in a single pass.

The nominal slab photodetector device design for the first tape out is shown in **Figure 5.43**. The nominal device design methodology was to transition from the input body silicon waveguide to a poly-Si rib waveguide to isolate the optical mode near the intended silicon germanium absorption region. The symmetry of the device was intended to enable connection to a double-data rate receiver by providing an evenly split detector [179]. The downside of this contact layout is that the nominal diffusion path requires diffusion beneath the contact regions that were observed to negatively impact the internal quantum efficiency by acting as recombination sites in the bulk-CMOS detector simulations.

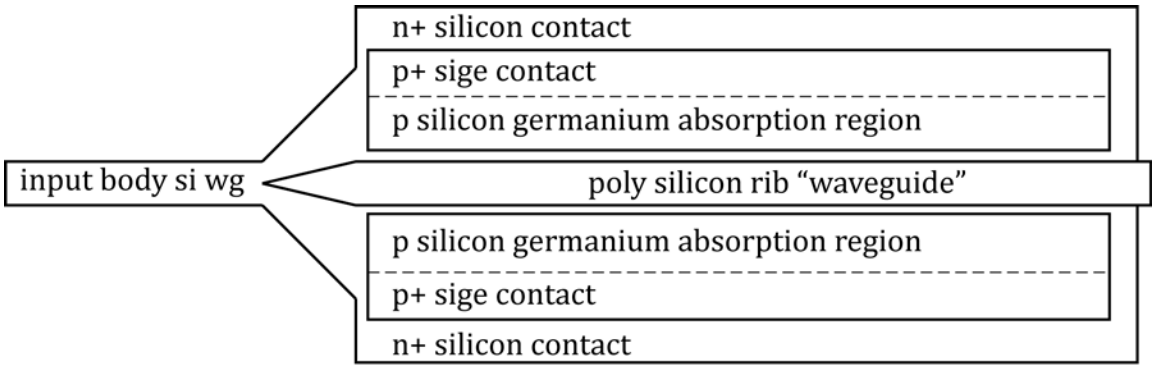


**Figure 5.43** Sketch of initial silicon germanium photodetector initial slab contact geometry. Detector lengths of 20 μm and 40 μm were fabricated.

The additional complication of the initially fabricated slab contact geometries is that the geometry shown in **Figure 5.43** resulted in extremely leaky diodes that did not produce significant rectification or diode behavior. The measured IV characteristic for one such diode is shown in **Figure 5.44**. The most likely explanation is that the n+ silicon to p+ silicon germanium diode leakage is significantly affected by band-to-band tunneling. In Sentaurus simulation of the electrical junction, significant band-to-band tunneling is observed for junction dopings in excess of 1 x $10^{20}$ $cm^{-3}$. In this case, the diode does not demonstrate significant rectifying junction characteristics required to demonstrate a photodetector. A cartoon illustrating this effect is shown in **Figure 5.45**, using the simulated Fermi levels for 1 x $10^{20}$ $cm^{-3}$ doping densities.

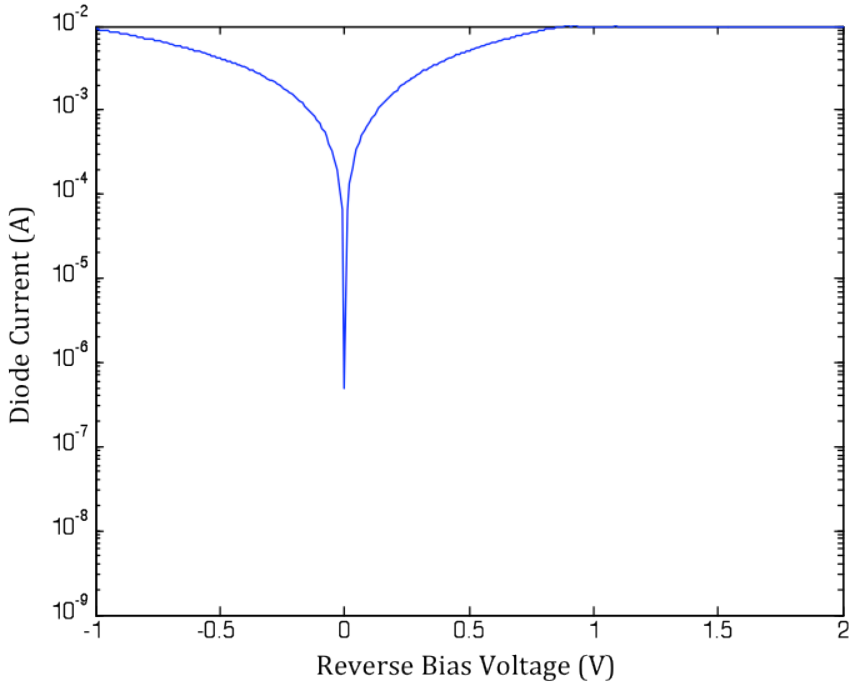


**Figure 5.44** Measured IV characteristics for a 20 µm long slab contact photodetector fabricated in the EOS8 test chip (IBM 45nm SOI-CMOS) containing an n+/p+ junction.

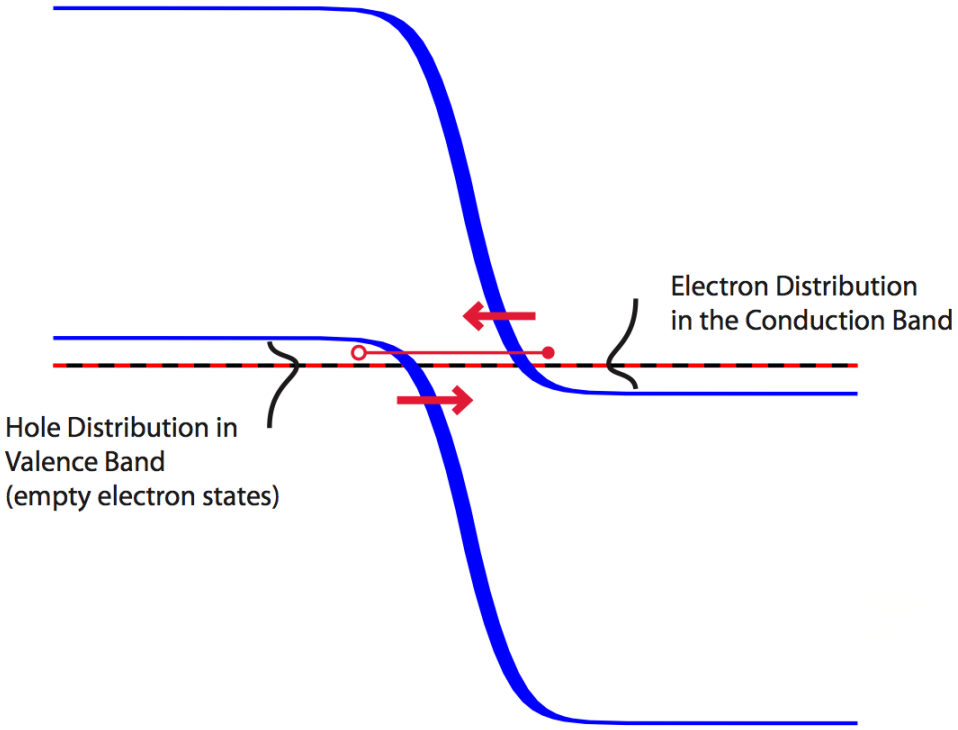


**Figure 5.45** Cartoon illustrating band-to-band tunneling leakage at the heavily doped p+ / n+ junction using simulated Fermi levels corresponding to 1 x $10^{20}$ $cm^{-3}$ doping densities.

Since the possibility of p+ / n+ tunnel junction leakage was known at design stage, an alternate design was fabricated on the die that utilized an additional poly-Si spacer layer to form a barrier between the heavily doped contact regions to improve diode behavior as shown in **Figure 5.46**. The diode leakage of 200 µA at 0.5 V reverse bias, full IV characteristics shown in **Figure 5.47**, was still large in this contact geometry, but measurement of the optical properties was possible.

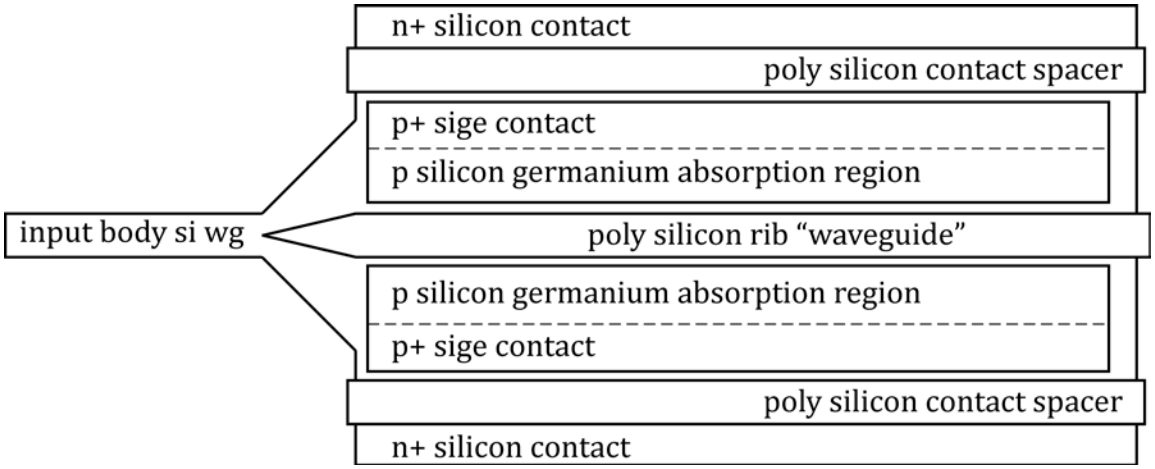


**Figure 5.46** Sketch of initial silicon germanium photodetector initial slab contact geometry including polysilicon contact spacers. Detector lengths of 20 µm and 40 µm were fabricated.

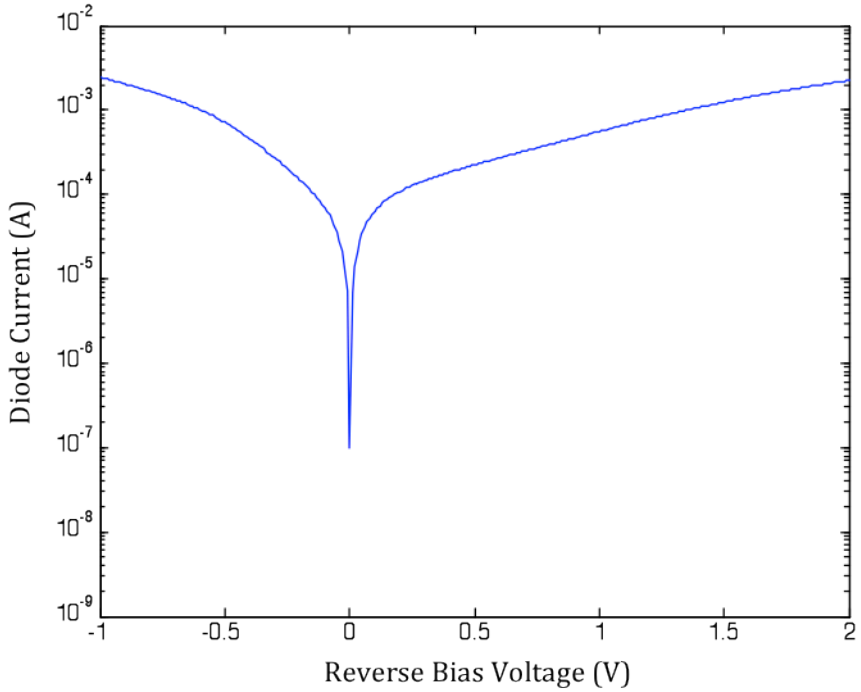


**Figure 5.47** Measured IV characteristics for a 20 µm slab contact photodetector fabricated in the EOS8 test chip (IBM 45nm SOI-CMOS) with a poly-Si rib between n+/p+ regions.

For the fabricated slab contact photodetector shown in **Figure 5.48**, a peak responsivity of approximately 4 mA/W was measured as shown in **Figure 5.49**. Several issues affecting the particular layout of the silicon germanium photodiode are believed to contribute to this poor measured performance. Post-layout optical simulation of the particular device design revealed several excess loss sources. The poor IV characteristics point to poor internal quantum efficiency. Also, the fraction of optical power absorbed due to free carrier absorption relative to band-to-band absorption is not known at this point and may contribute a significant excess absorption source that does not contribute to electron-hole pair generation.

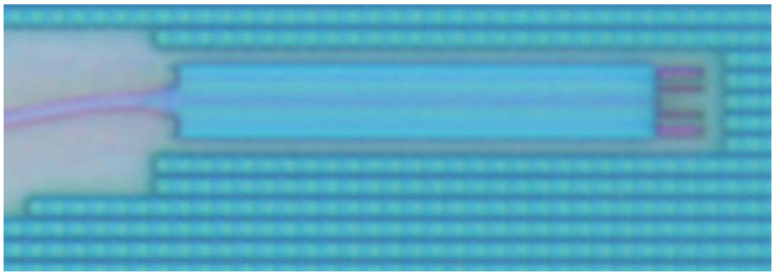

**Figure 5.48** Slab photodetector micrograph. The length of the wide slab region is 20 μm.

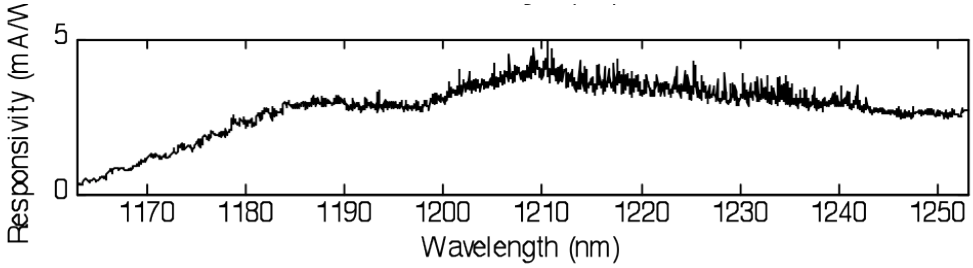


**Figure 5.49** Measured poly-Si contact rib slab photodetector responsivity.

First, specific to this slab photodetector geometry, the input taper to the detector region was intended to reduce reflections from the interface and was designed to be only 1 μm in length. As shown in **Figure 5.50**, the short input taper did not adiabatically expand the input mode while reducing the junction reflections. In this simulation, the metal contacts and absorbing silicon germanium region are not included to focus the attention on the passive optical design. The simulated power transmission into the detector slab was only 43%. It also had the unintended consequence of dispersing the input optical mode into the detector slab region. No

single mode of the detector slab region received more than 15% of the input waveguide power. This reduced the localization of the light to central polysilicon waveguide rib mode that overlapped the intended absorption regions and resulted in significantly more absorption in the contact regions.

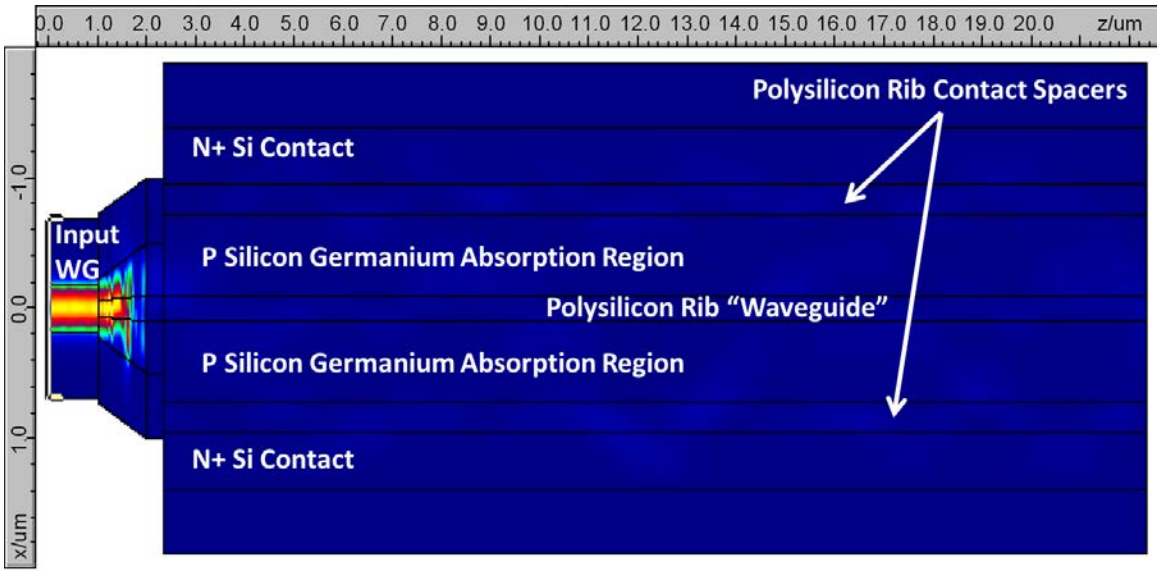


**Figure 5.50** FIMMPROP simulation of silicon germanium slab photodetector input taper from the first generation EOS8 tapeout.

Next, the poly-Si contact spacer ribs introduced additional waveguide modes that could be coupled into from the intended central poly-Si waveguide rib. Any light that was coupled into the outer poly-Si contact ribs possessed large overlap with the metal contacts and therefore had significantly higher loss. Neglecting the effect of the improper input taper, a further passive cavity FIMMPROP simulation is shown in **Figure 5.51**. When the contacts are added into the simulation domain without including the absorption of the silicon germanium, the excess attenuation causes only 12% of the input power to propagate to the end of the 20 μm detector region as shown in **Figure 5.52**. Combined with the taper loss and mode mixing, the optical parasitics of the fabricated structure are observed to significantly impact device performance in post-layout simulation.

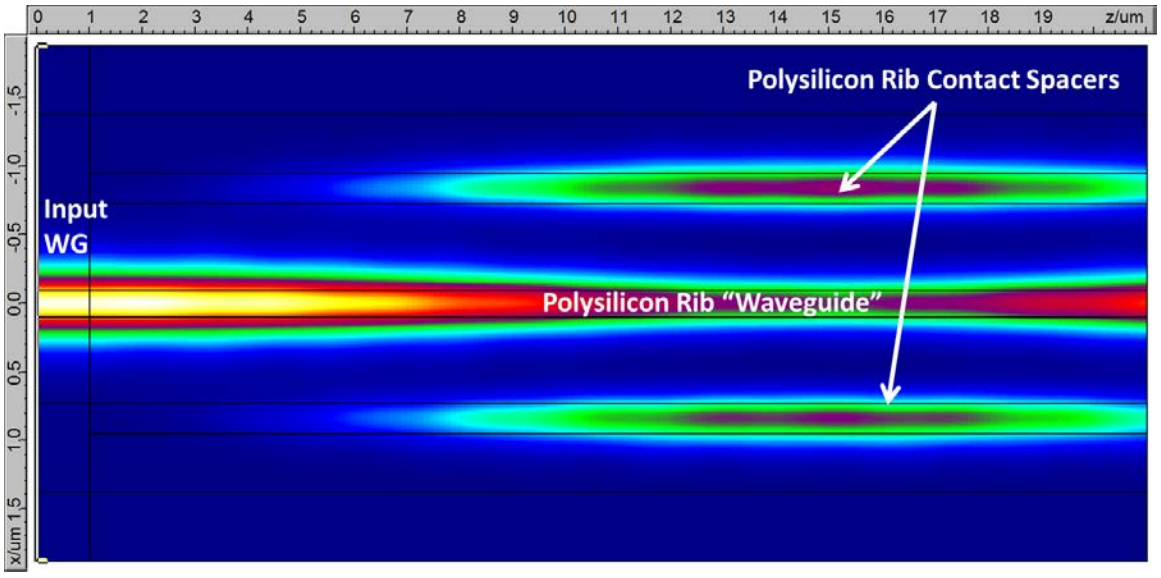


**Figure 5.51** FIMMPROP simulation of the passive device geometry of the silicon germanium slab photodetector including proper input mode launching. The coupling to the outer polysilicon contact rib spacers is clearly evident.

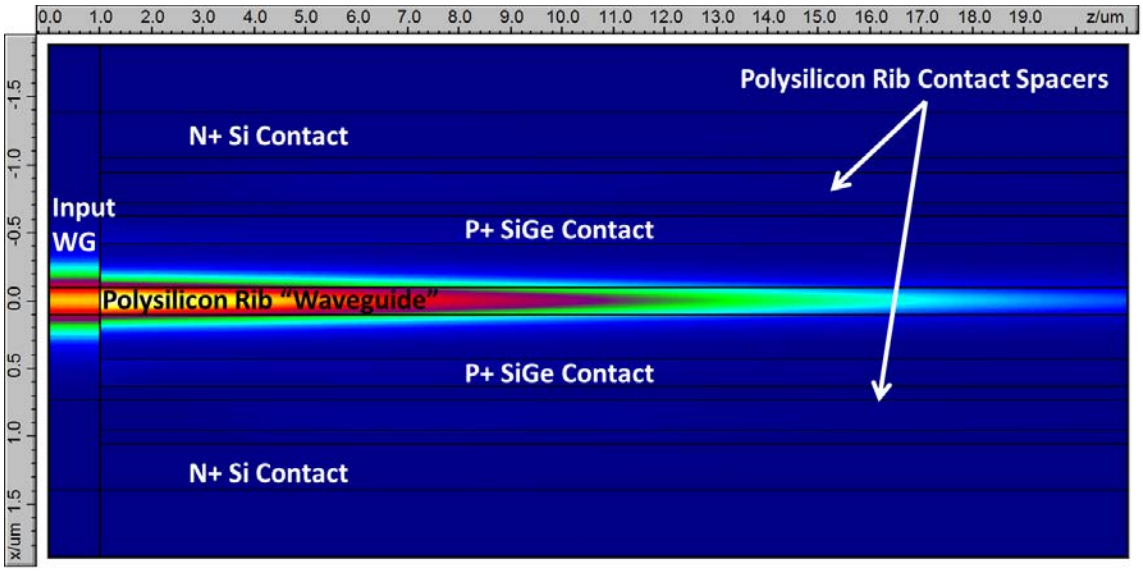


**Figure 5.52** FIMMPROP simulation of the fabricated device geometry of the silicon germanium slab photodetector including proper input mode launching. The silicon germanium absorption is turned off in this simulation and the only absorption present is due to the device contacts.

Beyond the optical losses, the high reverse bias leakage points to an electrical diode problem that may negatively affect the internal quantum efficiency. Since the

p+ silicon germanium is used as the source/drain of the p-channel transistors, proper circuit function guarantees that similar p+ silicon germanium to n-well were fabricated elsewhere on this test chip outside the photodetector devices. Even when the n+/p+ shunt path was removed by the polysilicon contact spacer, high leakage current was still observed. This observation tentatively points to a material quality issue in the silicon germanium layer. It is possible that internal design rules must be established to ensure epitaxial window openings that result in higher material quality. As in the case of the bulk-CMOS measurements, there may be a maximum epitaxial window width beyond which strain relaxation leads to defects. The width dependence of diode leakage will be further characterized in future process runs.

The final potential source for lower than expected performance is the free carrier absorption in the doped silicon germanium layer. In the layout of the structures, no special design layers are used in the silicon germanium regions to block standard transistor implants. This ensures that the silicon germanium diode formation process matches that of the source / drain contact in an attempt to minimize contact resistance. The doping density, however, has already been seen to be sufficiently large so as to cause large leakage in abrupt n+ / p+ junctions. With doping densities that match this tunnel junction behavior in electrical simulation of $1 \times 10^{20}$ $cm^{-3}$, the predicted material absorption would be roughly 550 $cm^{-1}$ and of a similar magnitude to the expected 500 $cm^{-1}$ band-to-band absorption in a coherently strained 25% silicon germanium alloy that was used for *a priori* simulations [132, 241, 242]. Expressed in similar units, the measured absorption in the IBM polysilicon film is 1160 $cm^{-1}$. Since the absorption coefficients are linearly proportional to the time domain rate, the ratio of the band-to-band and free-carrier absorption coefficients will directly multiply the total photodiode efficiency. For a $1 \times 10^{20}$ $cm^{-3}$ doping density, the impact on overall efficiency would be a ratio of 47% of total absorption that generates electron-hole pairs. It may therefore be necessary to block all further implants into the silicon germanium to reduce free carrier absorption in future designs. If the end-of-line doping density that results from the in situ doping can be reduced to $1 \times 10^{19}$ $cm^{-3}$, this ratio would increase to 92%.

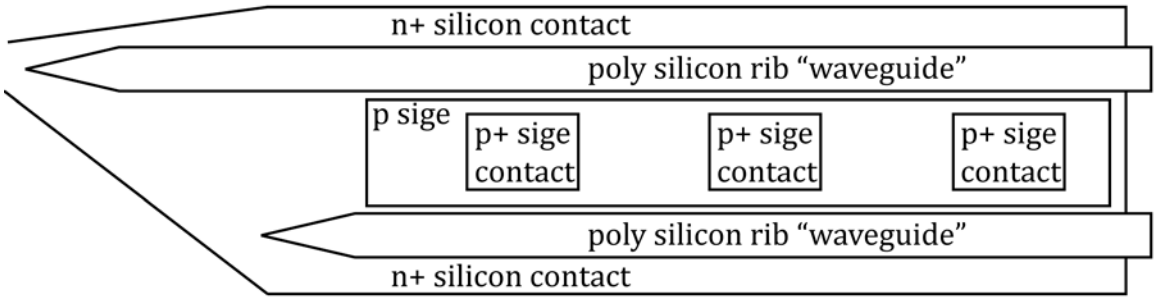


**Figure 5.53** Sketch of the revised silicon germanium photodetector slab contact geometry based around an asymmetrical input taper.

After the measurement of the initial slab photodetectors, revised designs were submitted for fabrication on EOS12. Although measurement results are not yet available at the time of this writing, the updated device geometry is shown in **Figure 5.53**. Two key differences with the previous designs are the improved optical design that was simulated in FIMMPROP for various device sizings to confine the light to optical modes with high overlap with the intended silicon germanium absorption regions while ensuring low overlap with the contact metallization. As shown in **Figure 5.54**, over 90% of input power is absorbed within 30 μm of silicon germanium device length. The absorption is due primarily due to the presence of the silicon germanium material instead of the silicon contacts and metallization. As shown in **Figure 5.55**, when the silicon germanium layer is replaced with silicon in the simulation, approximately 74% of the input light propagates to the output plane in the first and second modes corresponding to the two polysilicon rib modes. Additionally, the electrical junction design was modified to eliminate n+/p+ junctions and reduce the impact of minority carrier recombination due to the silicon germanium contacts. 32 different device variants were fabricated around this nominal design in order to enable a richer set of measured data to better calibrate device design simulations.

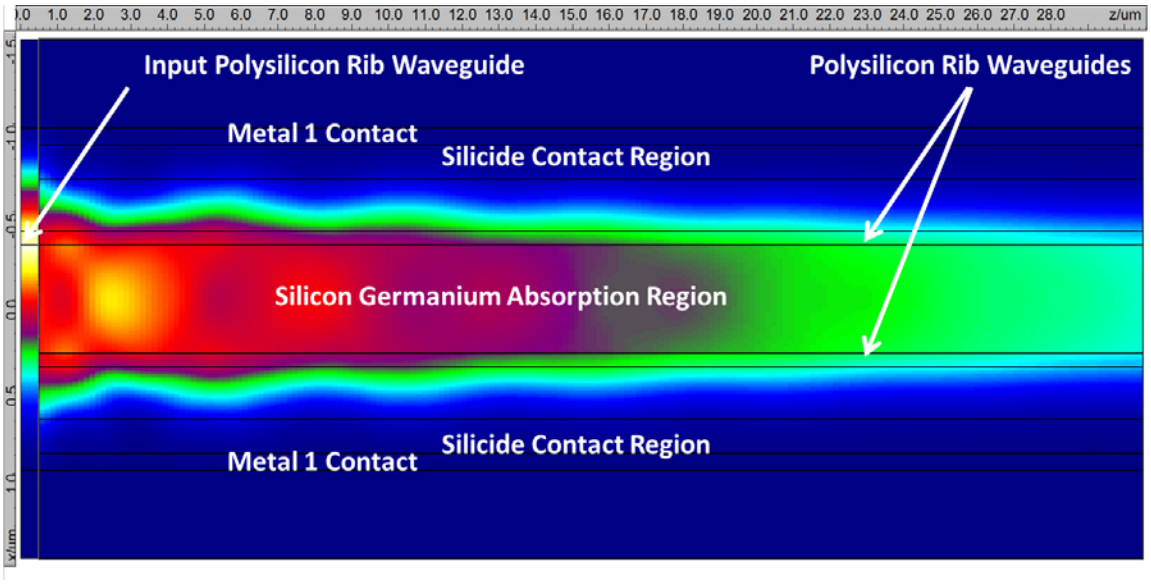


**Figure 5.54** FIMMPROP simulation of silicon germanium photodetector design used for EOS12.

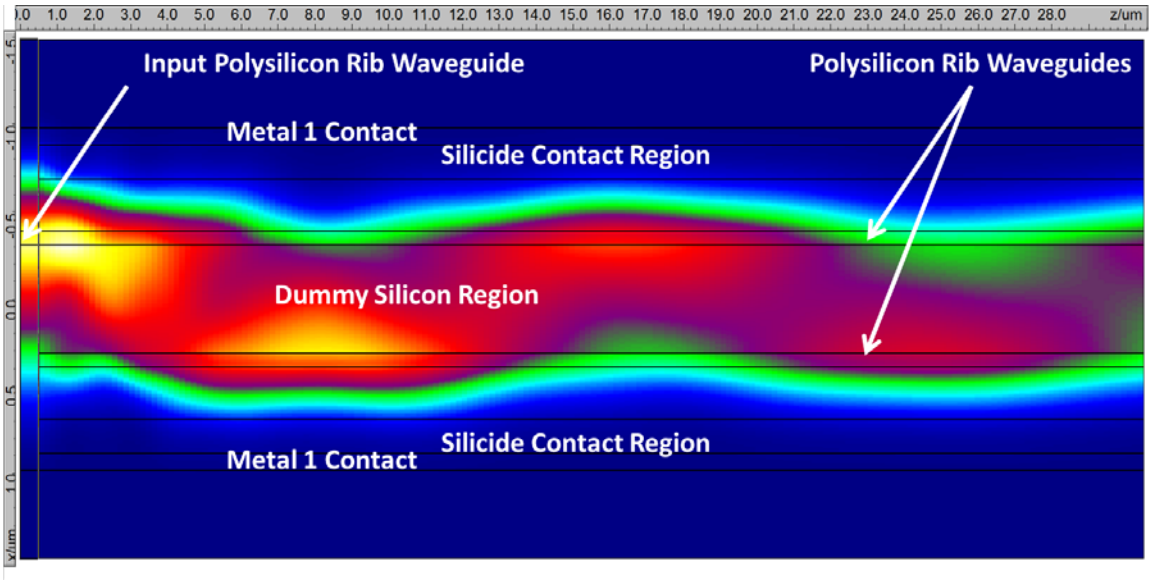


**Figure 5.55** FIMMPROP simulation of silicon germanium photodetector design used for EOS12 without the absorption of the silicon germanium layer present in the simulation to demonstrate low extrinsic contact optical loss.

### *5.5.4.3. High Speed Measurements*

Although the efficiency of the first generation of silicon germanium photodetectors was below desired targets, it is still possible to verify the high speed

operation of similar nanostructured pn-junction device geometries. To measure the small signal bandwidth of a resonant detector, a JDSU APE 20GHz lithium niobate modulator designed for 1310nm operation was driven with an Anritsu MG3690C signal generator. The resulting electrical signal from the detector was measured on a HP 8000 microwave spectrum analyzer with a lightwave front end module. The setup transfer function was calibrated using a factory-measured Discovery Semiconductor DSC20S photodetector with a 40 GHz bandwidth as the reference sample. The relative received power difference between the silicon germanium device-under-test and the reference detector is plotted in **Figure 5.56**. Although the setup bandwidth of the 20 GHz source and microwave spectrum analyzer is not sufficient to characterize the full bandwidth of the detector, less than 3 dB drop in the modulated photoresponse is observed at a 20 GHz modulation rate. This demonstrates the suitability of such a detector for high speed communication system applications.

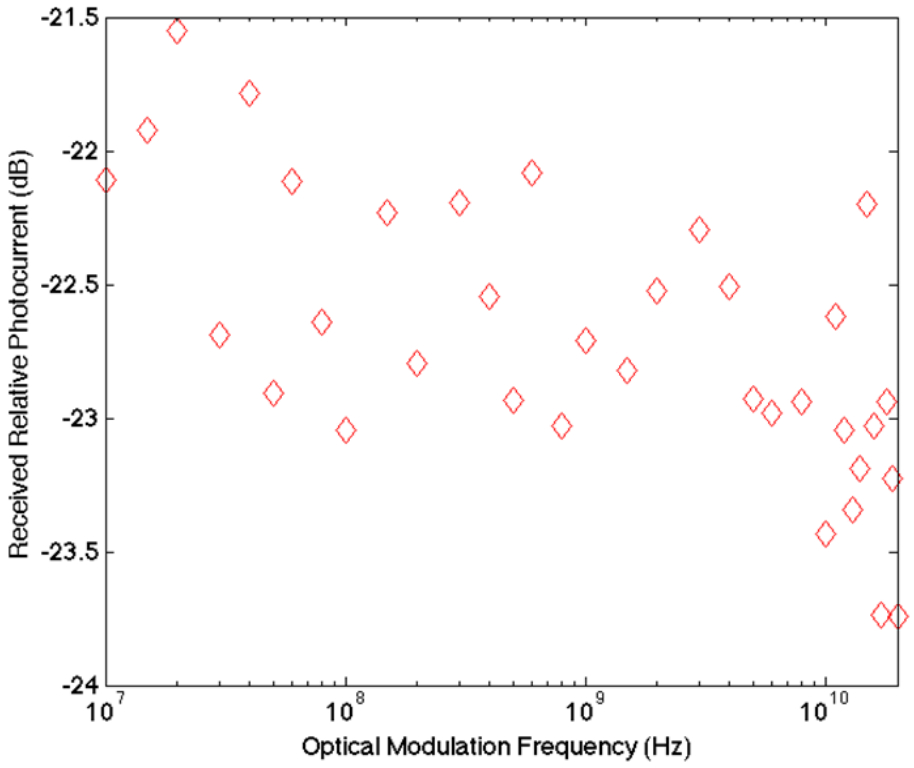


**Figure 5.56** Measured frequency response of a resonant silicon germanium pn-junction photodetector demonstrating less than 3 dB roll off in the modulation response at 20 GHz.

## 5.6. DRAM Photodetectors

For the memory processes under consideration for photonic integration, silicon germanium stressors are not part of the standard process. As such, linear photodetector integration requires the introduction of a new silicon germanium process module. The creation of a new process layer, however, does increase design flexibility to enable optimized performance of the photodetectors. A key integration constraint, however, is to form the photodetector prior to the transistor implantation and anneal processes. The SiGe-first integration methodology allows all contact and implant steps to be reused for the detector formation to simplify the final process. Additionally, since the epitaxial step requires high temperature processing, the transistor operating characteristics would be modified by increased dopant diffusion. A post-transistor SiGe integration proposal would require redesigning the transistors which is not within the scope of our current effort. The implication of forming the detectors prior to the formation transistors is that the detector material must withstand the thermal processing of the high temperature spike anneals that exceed 1050 °C. Due to the melting point of silicon germanium alloys, this processing constraint limits the maximum germanium mole fraction to 50%.

Similar to the bulk-CMOS process, the silicon germanium must be formed on the handle silicon substrate, so local substrate removal in the detector region is impossible. Two differences of the memory photonic integration process, however, greatly simplifies the optical design of the detector. First, a deeper trench isolation of from 1.0 μm to 1.5 μm is proposed for the photonic integration regions. This flexibility eliminates optical substrate leakage as a major performance degradation source. Second, the silicon germanium layer can be regrown above the surface of the silicon substrate such that proper mode matching to the input polysilicon waveguide may be achieved. The proposed device geometry is then shown in **Figure 5.57**.

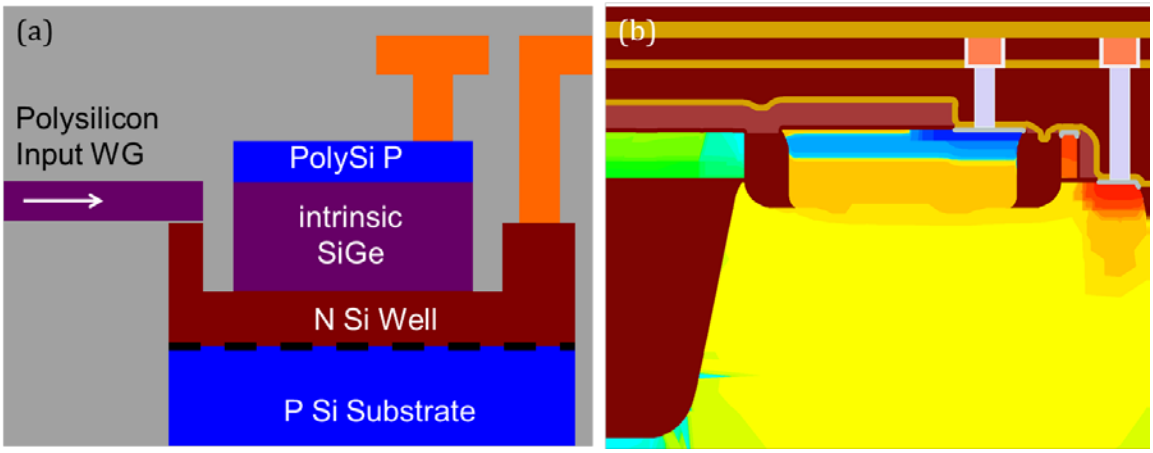


**Figure 5.57** Proposed DRAM integrated photodetector cartoon (a) and Sentaurus TCAD process simulation (b).

The two key optical design criteria are that the higher index of the silicon germanium layer forms a bound rib waveguide mode and that the coupling from the input polysilicon waveguide mode incurs minimal excess loss. The formation of a bound waveguide mode primarily affects the choice of epitaxial stackup for the silicon germanium layers. Since the critical layer thickness at which defect formation begins for silicon germanium decreases rapidly with germanium mole fractions above 25%, one proposal is to employ only a thin, strained >30% silicon germanium layer after a thin graded buffer of silicon germanium mole fraction from 0% to 25% [241]. Since the highly strained peak mole fraction silicon germanium is likely to only be quasi-stable [241], a silicon capping layer is designed to stabilize the strain field prior to the peak thermal processing of the transistor formation process. An example epitaxial stackup that has promising material properties and yields a bound optical mode is a 250nm linearly graded buffer from 0% to 25%, followed by a 50nm thick 35% silicon germanium layer and capped with a 100nm silicon layer. The refractive index profile and fundamental optical mode for a 2 μm rib waveguide formed in such a waveguide is plotted in **Figure 5.58**. The second criteria of mode matching to the input polysilicon mode is reasonably well matched to the input 2 μm waveguide mode with a 73% modal overlap integral as calculated in FIMMPROP.

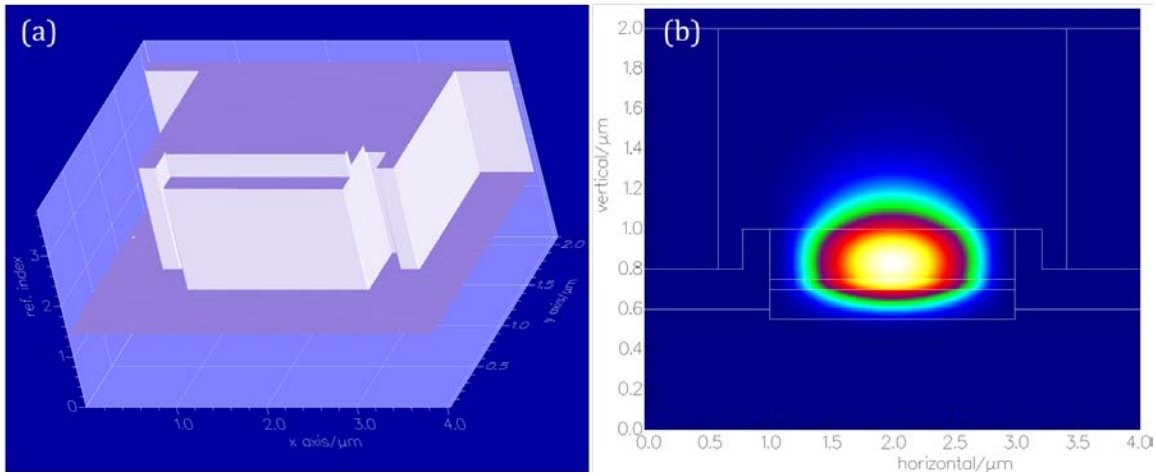


**Figure 5.58** Refractive index plot (a) and resulting fundamental optical mode (b) simulated for proposed epitaxial stackup. Note that the y-axis is reversed such that the silicon substrate is "up."

At the time of writing of this thesis, the silicon germanium integration for the memory photonic integration platform is under significant development. No fabricated results are available for inclusion.

## 5.7. Conclusion and Future Work

The states of the active photonic devices fabricated over the course of this thesis are not sufficiently mature to directly address many real world applications such as the microprocessor-to-memory photonic links discussed in Chapter 2. The modulators may not have achieved a data rates above 1 Gb/s and photodetectors have not demonstrated efficiencies above 1%.

The modulator performance characteristics in carrier injection operation mode are limited by the suitability of the integrated drivers to the fabricated device characteristics. However, even with matched pre-emphasis integrated driver circuits to increase the data rate, the energy efficiency of ~1 pJ/bit is also an order of magnitude above targets. As such the switch to carrier-depletion-mode operation is imperative to generate suitable devices. The progress towards that goal is most explicitly discussed for the DRAM modulators where a vertical junction design can be leveraged to fabricate modulators similar to the best published literature examples [92]. In the CMOS process, suitable doping densities on the order of 1 x $10^{18}$ $cm^{-3}$ are present in all scaled-CMOS fabrication flows for transistor well and halo implants. Improved contact geometries to reduce the resonance line width to

the ~20 GHz range to enable depletion-mode operation are the active research goals of project collaborators Jeffrey Shainline and Milos Popovic at the University of Colorado at Boulder.

Improvements to the photodetector devices are already under way with the latest round of fabricated device variants from the EOS12 test chip that have not yet been tested. Importantly, the known sources of extrinsic optical absorption in the contact structures have been significantly reduced. The electrical junction design has been simplified in an attempt to improve the leakage characteristics and internal quantum efficiency. Additionally, setting design rules such as maximum epitaxial window opening sizes to prevent defect formation may further improve the internal quantum efficiency and reduce dark current leakage. Future work will focus on better characterizing the process parameters and exact details of the fabricated structures. Including implant-blocking design layers in the absorbing silicon germanium regions may reduce the free-carrier-absorption in these regions.

It is also important to note that we have not yet successfully demonstrated receiving data using the silicon germanium photodetectors in conjunction with the integrated receivers. Several testing efforts were not successful in achieving this result. The low efficiency and high leakage characteristics are likely the root cause of this difficulty. The modulated bandwidth testing demonstrated the ability of the detectors to operate at high speed for a ~10% modulation depth input sinusoid. Although no unusual processes that would affect the pattern dependent operation characteristics of the photodetectors are expected, an independent demonstration of receiving data with the silicon germanium detectors is planned using a commercially available transimpedance amplifier. For this purpose, 10 bare die of TA205C transimpedance amplifiers have been purchased from Euvis, Inc.

# 6. CONCLUSIONS AND FUTURE WORK

The research project associated with this thesis started with the rough goal of assessing the possibilities of fabricating nanophotonic devices in state-of-the-art electronics manufacturing processes such as CMOS. Luxtera had already demonstrated that novel electronic-photonic circuits could be fabricated in older CMOS process technology that had been modified to facilitate the fabrication of silicon photonic devices [245]. In order to get access to the highest performance transistors and lowest manufacturing costs, however, we chose instead to attempt to modify the design and layout of the photonic devices to comply with all established design rules required for an established CMOS foundry to fabricate our electronic-photonic chip as if it were a standard electronic design. Prior examples of research using zero-change CMOS technology had demonstrated a wide range of non-standard devices such as MEMS resonators [57], but no examples of in-plane nanophotonic devices such as waveguides, multiplexing filters or modulators existed in the published literature.

In general, the consensus at the outset from most members of the CMOS community that we discussed our work with was that it was simply not possible to fabricate the desired photonic devices in a standard, high performance process. Chief perceived obstacles were that the Manhattan, uniform processing optimizations required for deeply-scaled transistor fabrication would prohibit the integration of the large, inhomogeneous, curved shapes required for photonics. Additionally, the electronic optimizations for pervasive doping, silicidation and metallization processes were expected to forbid low loss optical propagation. Although the concerns were justified in the absence of testing, integrated photonic components with waveguide losses approaching the best independent demonstrations were successfully fabricated over the course of the thesis. Preliminary demonstrations of all major device variants, modulators, photodetectors, filters and vertical grating couplers were all fabricated as well. As a result, changing the assumptions regarding suitability of standard foundry CMOS electronics processes for the fabrication of nanophotonic devices is perhaps the single largest contribution of this thesis.

## 6.1. Core-to-Memory Photonic Networks

After early success in providing a proof-of-concept for the feasibility of integrated photonic devices within state-of-the-art CMOS processes, the long term goal shifted to utilizing the devices to provide photonic interconnect that would impact the way future computer systems were built in such processes. The resulting application problem of photonically interconnecting the processing cores to main memory was described in Chapter 2. Over the course of this thesis, all photonic devices required for system demonstration were fabricated with various measured performance characteristics. The key photonic components are identified on the network architecture cartoon and fabricated CMOS chip micrograph in **Figure 6.1**.

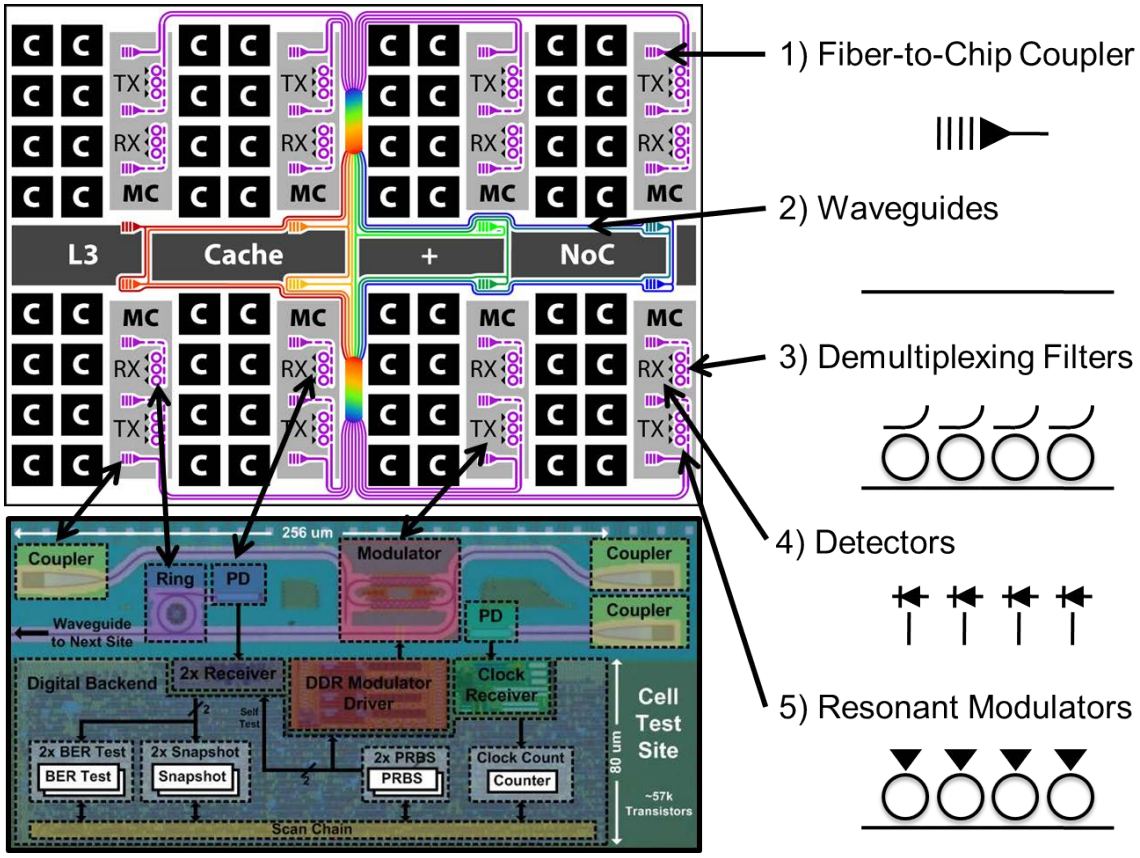


**Figure 6.1** Identification of key photonic devices on the multicore processor and photonic interconnect architecture cartoon from Chapter 2 and the labeled EOS8 CMOS die micrograph from Chapter 5.

To assess the performance of the photonic devices, we will analyze what total interconnect system metrics may be achievable to demonstrate the 96 Tb/s

aggregate core-to-memory bandwidth compute system discussed in Chapter 2. The photonic device performance characteristics can then be calculated in the context of the wavelength division multiplexed read and write links shown in **Figure 6.2**. Using current, modeled and predicted photonic device performance characteristics, total interconnect system performance metrics are shown in **Figure 6.3**.

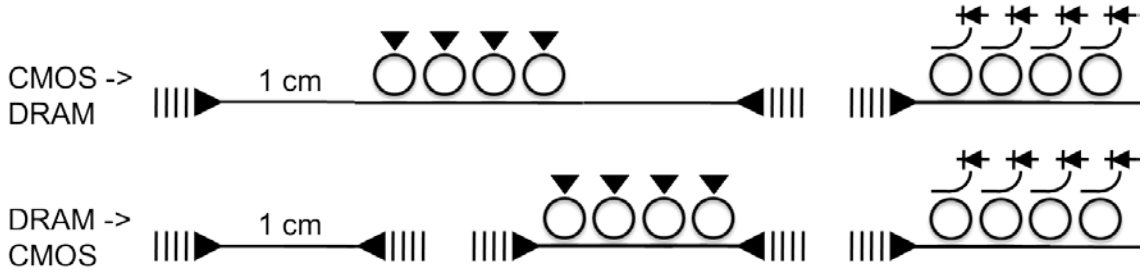


**Figure 6.2** Write (CMOS->DRAM) and read (DRAM->CMOS) photonic link paths for the compute system using photonic device symbols introduced in **Figure 6.1**.

| | DDR3 | HMC | research |
|---|---|---|---|
| **Component** | **Current** | **Modeled** | **Future** |
| Thermal Tuning | 4.3 pJ (1.3mW) | 200 fJ (0.5 mW) | 35 fJ (0.2 mW) |
| Modulator Driver | 2 pJ | 500 fJ | 38 fJ |
| Receiver Energy | 53 fJ | 50 fJ | 50 fJ |
| Receiver Sensitivity | 5 μA | 5 μA | 3 μA |
| **Total Photonic** | **6.6 pJ** | **755 fJ** | **123 fJ** |
| **Electrical** | **64.7 pJ** | **10.8 pJ** | **1 pJ** |

| | 2011 | '12-'13 | '13-'15 |
|---|---|---|---|
| **Component** | **Current** | **Modeled** | **Future** |
| Coupler | 3.5 dB | 3 dB | 1.5 dB |
| Waveguide | 3 dB | 2 dB | 2 dB |
| Mod. Insert. | 1 dB | 1 dB | 0.5 dB |
| Mod. Rate | 600 Mb/s | 5 Gb/s | 10 Gb/s |
| Filter Insert. | 0.5 dB | 0.5 dB | 0.5 dB |
| WDM Count | 8 | 32 | 128 |
| Detector Eff. | 0.008 | 0.1 | 0.5 |
| Laser Power per Link | 44 mW (244 pJ) | 1.8 mW (1.2 pJ) | 48 μW (16 fJ) |
| **# Fibers** | **26,674** | **832** | **240** |

**Figure 6.3** Status of achieved, modeled and predicted photonic device performance relative to various generations of electrical link technologies.

With the caveat that data receiver operation with the integrated photodetectors has not yet been demonstrated, the current state of photonic devices is competitive with existing and next generation on-chip power dissipation numbers. Practically realizing such a system with photonic devices that have been fabricated and tested in the lab over the course of this thesis, however, would require 26,674 fibers and

44 mW per link of laser power. This is not an achievable physical system in the context of even extremely relaxed cost constraints. If device improvements are made that seem likely given current process and device understanding, the modeled system improves significantly. Comparing the 2012-2013 timeline of these photonic devices to the proposed hybrid memory cube (HMC) architecture, the estimated on-chip power dissipation is over an order of magnitude smaller than the electrical projection. Additionally, the physical system becomes significantly more realizable with an 832 fiber count and 1.8 mW of optical power per link. Finally, device targets based on several design iterations of improvement to target the 2013-2015 timeline still beat research electrical link projections by a factor of five while providing physically realizable fiber and input optical power requirements.

Consistent with this timeline, initial photonic device results including those presented in this thesis combined with significant computer architecture and link-level research have resulted in the launch of a project team under the DARPA Photonically Optimized Embedded Microprocessors (POEM) program. Starting in August of 2011, the POEM team led by principle investigator Vladimir Stojanovic seeks to demonstrate a photonically interconnected compute system in a two phase program under the management of Dr. Jagdeep Shah. At the end of the first approximately year and a half phase of the program, the project goal is to demonstrate the bidirectional 8 wavelength link at a 5 Gb/s data rate between a die fabricated in the SOI-CMOS process and a die fabricated in the memory photonic integration process shown in **Figure 6.4**.

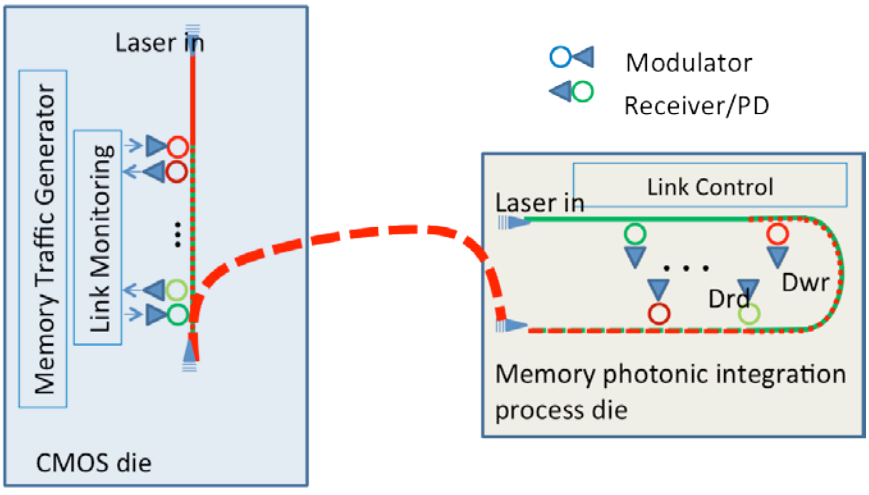


**Figure 6.4** Project team link goal for first phase of POEM program. Based on figure provided by Vladimir Stojanovic.

## 6.2. Future Work

Beyond the future improvements associated with individual devices that have been discussed in previous chapters, further areas of research are needed to accomplish the larger goals of photonic interconnect within compute systems and / or other photonic application areas within state-of-the-art CMOS foundries.

### 6.2.1. Packaging

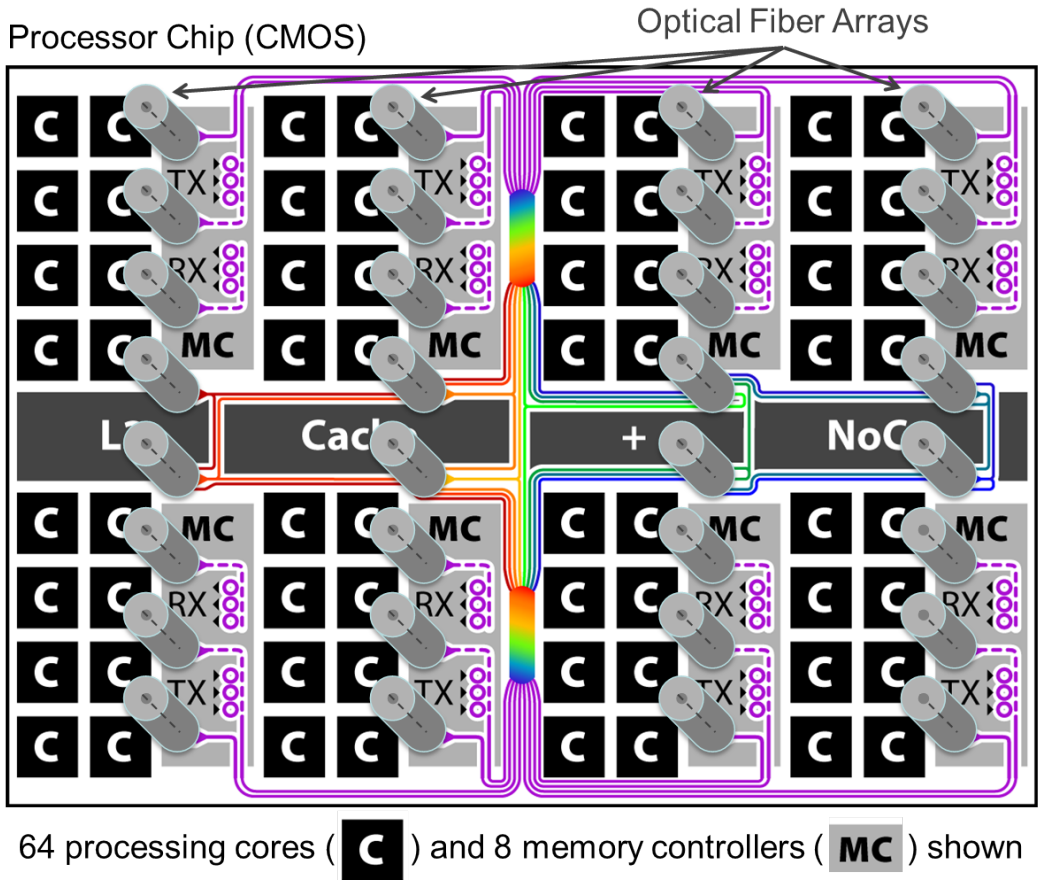


**Figure 6.5** Processor network architecture cartoon including the required placement of optical ribbon cable connectors for a small scale system.

The realization of a photonic interconnect within a computer system may be limited by available opto-electronic packaging technology in the very near future. Microelectronic packaging for CPUs has been aimed at increasing the electrical connector density while maintaining reasonable total cost. In this work, potentially tens of thousands of electrical connections may be replaced by hundreds of optical fibers. A cartoon of the desired placement of a traditional fiber ribbon I/O over the

vertical coupler array designed in the photonic link network architecture introduced in Chapter 2 is shown in **Figure 6.5**.

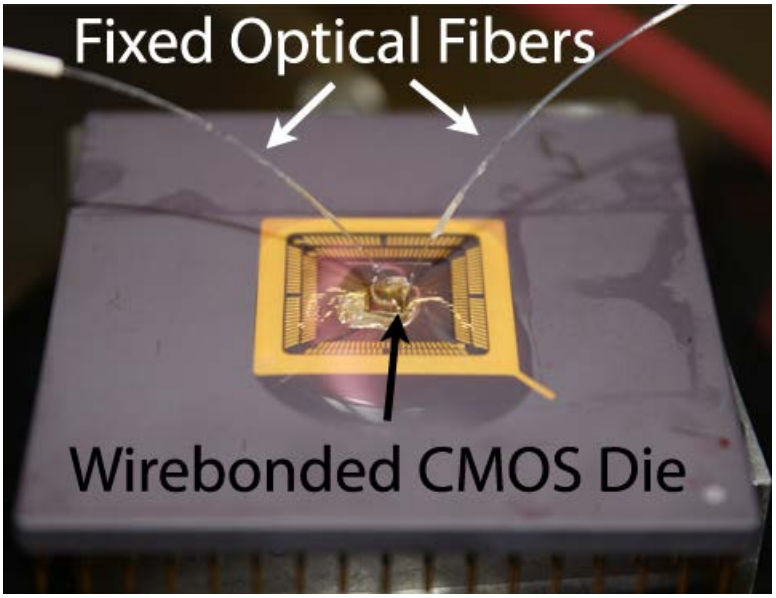


**Figure 6.6** Labeled photograph of initial electro-optic packaging attempt on a wirebonded CMOS die. The substrate transfer process described in Appendix B was used to create a functional electro-optic die.

Due to the importance of an available packaging technology for system demonstration even at a prototype level, various ideas for possible solutions have been conceived and initial packaging techniques have been attempted. Since the traditional microelectronic wirebond packaging technology that we have used for initial CMOS test chips leaves the center area of the die free of any electrical connections, optical fibers can be interfaced to the included photonic devices. In an initial lab demonstration, I actively aligned input and output fibers, which were mounted to positioning arms with Crystalbond 555 wax, to a photonic device on a packaged CMOS die. After optimizing initial coupling, I added a drop of Norland Optical Adhesive 71 to the chip surface to wet the fibers and the regions over the vertical couplers. Next, I reoptimized the coupling to account for the index change any fiber misalignment from the liquid. The NOA was then cured in place with a UV source to bond the fiber tips to the chip. Since this bond is not enough to support the fiber flex, the fibers were further secured using Sylgard 184 PDMS that was poured and cured in place as an encapsulant. The optical fibers were then released from the positioning arms by heating the arms with a solder iron above the Crystalbond 555

melt temperature of 55 °C. The final encapsulated electro-optic packaged die is shown in **Figure 6.6**. The total change in transmission from reoptimized coupling to released, bonded optical fibers was ~1.5 dB in this attempt. Further encapsulation or fiber management solutions beyond the packaged die must also be developed, however, as the fibers snapped under rough handling of the packaged part.

Although this electro-optic packaging attempt enabled the addition of only two optical fibers, replacing the single fibers with fiber arrays would yield a significantly more scalable solution. Included on the initial polysilicon waveguide D-1 test chip were vertical couplers arranged in a 12-port array to match an 8 degree end-face angle polished v-groove fiber array. Optical coupling experiments with an assembled array purchased from OZ Optics (VGA-16-250-8-A-10.3-7.8-2.03-S-1300/1550-9/125-X-0.25-2-0) have been performed with measured excess fiber-to-fiber coupling loss of less than 1 dB. Since two optical positioning arms are available in the setup described in Appendix A, 24 simultaneous optical connections could be made to the wirebonded photonic die using two 12-port arrays in this simple packaging approach. This port count is sufficient for end of program POEM demonstration goals that have been estimated to require 20 fibers.

The manual labor entailed in such a packaging approach, however, makes it unsuitable for mass production. Instead, optical fiber arrays that are either integrated within or attached via a passively-aligned connector to the electrical package are desirable for that application. There are two basic ways that such a packaging platform is currently envisioned in conjunction with the localized substrate removal technology required for photonic functionality within zero-change processes. The one, shown in **Figure 6.7**, utilizes the undercut process for localized substrate removal and integrates the optical fibers into the flip-chip electronic package. The other, shown in **Figure 6.8**, integrates the optical fibers into the heat sink or associated connectors and uses a trench etch for localized substrate removal. In both cases, it is desirable to use the existing electronics packaging alignment step of carefully aligning the die to the electronic package to enable the optical alignment as this approaches micron-scale precision today without any optical constraints. If optical fibers are integrated into the electrical package, that is the only required connectorization packaging step. Otherwise it may be possible to place alignment pins or holes into the carefully aligned electronic package to allow optical connectors to passively align to these features. Close coordination with industrial partners in future work is required to address such issues in future work.

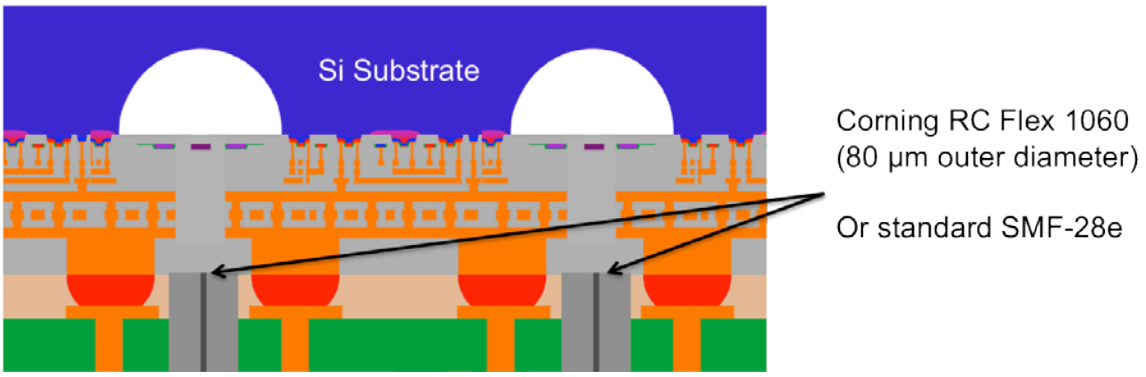


**Figure 6.7** Electro-optic packaging technology based on front-side localized substrate removal (undercut) and optical fibers integrated into the flip-chip electronic package.

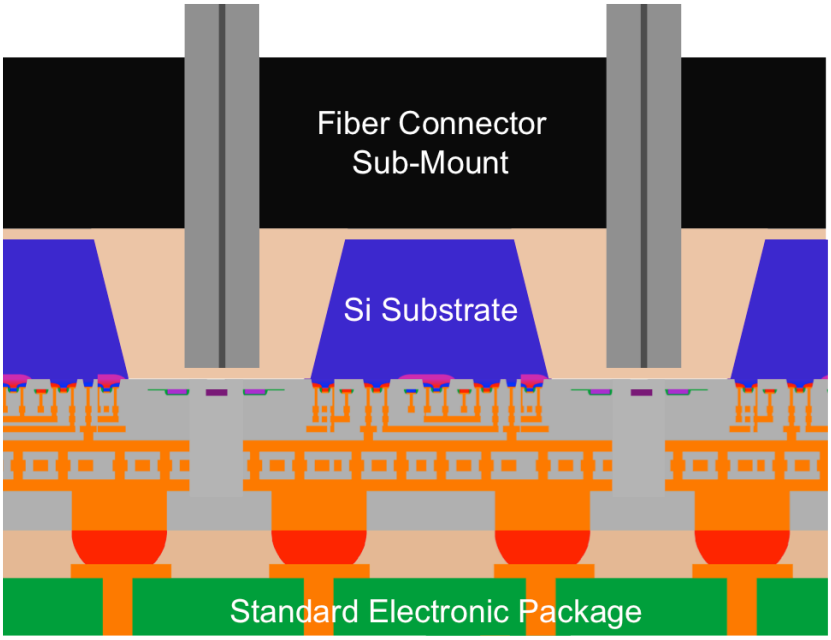


**Figure 6.8** Electro-optic packaging technology based on back-side localized substrate removal (trench etch) and optical fibers integrated into the heat sink.

### 6.2.2. Development of Memory Photonic Integration Platform

Under the POEM program, further development of the polysilicon-based memory integration platform introduced in this thesis will be continued between the academic research teams of MIT, University of Colorado at Boulder and University of California at Berkeley and our industrial partner Micron Technology. Improvements in polysilicon waveguide loss that have been demonstrated in short-

loop and process integration fabrication flows must be translated to the full process flow. Additionally, the transistors must utilize the same polysilicon layer used for the photonic waveguides while maintaining their existing target electrical specifications. Importantly, both the electronic and photonic devices must maintain desired performance characteristics on the same wafer to enable the photonically addressed memory platform. Already, various tests conducted in the foundry have revealed cases where a potential photonic optimization path is measured to negatively affect transistor performance as well as where more complete wafer level processing has degraded photonic performance that had looked promising on short-loop wafers.

### 6.2.3. Interoperability and Network Protocol Standardization

In order for complex systems such as computers to be developed in a commercial setting, different companies must be able to manufacture independent parts that can interoperate. The highest level of such a matching is at least a consensus on wavelengths on which to focus device and process development as well as general ideas of operating data rate and multiplexing ratios. At the time of writing of this thesis, even this level of consensus does not exist. Further degrees of specificity will eventually be required to tackle protocol level issues such as the ability to due nearest neighbor wavelength locking or dynamic channel shifting to compensate for thermal drift as discussed in Chapter 4. Such basic assumptions on protocol and network provisioning can have an enormous impact on the practicality of a given technology development platform. For example, if absolute wavelengths must be specified, a fabrication platform that may dramatically reduce thermal tuning power is significantly more preferable than other options that may have merits in other device performance metrics.

### 6.2.4. Photonic-Friendly CMOS Foundry Process Modules

In this thesis, no changes to the in-fab processing of the CMOS foundry platform were made. This has been necessary to get access to the most advanced technology at the lowest incremental cost. For any foreseeable growth in the importance of the integrated photonic devices, it is highly unlikely that any change beneficial to the photonics that would degrade digital logic transistor performance would ever be included in advanced CMOS technology. It may be possible, however, for additional

process modules that may significantly enable photonic device performance and is benign to the transistors if the demand from large customers or internal microprocessor projects exists. A similar pathway has emerged for analog circuit design in scaled-CMOS processes. Although the transistors are fully optimized for digital functionality, additional complications such as high precision capacitors, varactor doping levels and triple-well dopings are included as extra process options to enable improved analog device performance. Similar work has been undertaken for the memory integration process over the course of this thesis. The availability of such options may be directly related to how convincing a case can be made for integrated photonic interconnect in the academic literature as well as initial commercial and government research projects in the coming years.

# 7. APPENDIX A: TESTING OF VERTICALLY-COUPLED OPTOELECTRONICS: λ=1.1-1.6μm

To test photonic devices that were fabricated over the course of this thesis, a broadband, vertically-coupled test platform was designed and built. This platform consists of a two generations of electro-optic test stations that accurately position two fibers at a variable angle to the chip surface, fiber-optic components to control the optical path and a variety of laser sources and detectors to cover the optical wavelength range from 1.1 to 1.6 μm. To test below the telecommunications O-band edge of 1260 nm, a new external-cavity, tunable laser source was built and characterized due to the lack of commercially available sources. This appendix will describe the details of the components of this test setup as well as techniques and structures used to characterize the integrated optical devices.

## 7.1. Vertical-Coupling Test Requirements

By changing the optical access direction from the edges of the test chips to the surface normal direction, many aspects of fiber positioning and alignment feedback need to be changed in optical test station design. Traditionally, fiber-positioning stages have been built to accommodate six axes of motion over small travel ranges such as 4 mm for translation and 6 degrees for rotation [246]. This model fits test needs when the approximate position and angle of the fiber is known *a priori*, as in the case of edge coupling. For edge coupling alignment feedback in the simplest case, the test station user views the chip and fibers through a microscope normal to chip surface. In this case, the user can easily position the fibers on either side of the chip to be approximately aligned in the plane of the chip with the appropriate working distance. The out of plane alignment can also be roughly accomplished by ensuring that the fiber tip is in the same focal plane as the chip surface under a high magnification objective. After these rough alignment steps, the fiber-to-fiber transmission is typically sufficient to enable the alignment to be optimized by maximizing the measured transmission. To enable better initial alignment, some test station designers include an infrared-sensitive camera to either image through the surface-normal microscope or through a separate imaging path aligned to the output facet. This assists in turning a two unknown problem, the simultaneous

alignment of each fiber, two separate alignments: first aligning the input fiber to maximize the coupled power visible on the IR camera, and second aligning the output fiber to maximize the received power.

However, in vertical coupling, the test situation is significantly more complicated. First, the exact angle of the fiber relative to the surface normal is dependent upon the working angle of the second-order grating at the test wavelength. Due to design considerations and fabrication error, this angle may vary over +/- 20 degrees. To accommodate such variability, additional rotation range must be added to the setup. Second, the fiber-positioning arm, preferably with a widely variable coupling angle, must now fit between the surface normal microscope and the chip surface. This places a significant constraint on the size of this component as compared to the working distance of the microscope objective. Additionally, the critical alignment of the spacing between the fiber and the coupler is no longer directly observable under the microscope. This then requires a separate imaging path in the plane of the chip to monitor this spacing. Further, when observing the fiber alignment relative to the integrated grating couplers, the large fiber cladding diameter hides the exact location of the much smaller coupler that is sized to the fiber mode size. This roughly 25x size mismatch is further complicated by the coupling angle that may differ significantly from normal incidence. The result is that instead of the optimal fiber alignment of centering the fiber outer diameter over the coupler, the two must be offset some arbitrary distance in the direction of the coupling angle. The result is that the rough alignment process is sufficiently complicated to require the use of an IR camera. With the IR camera mounted on the surface normal microscope, the relative scattering behavior can be observed and related back to the actual position of the focused optical mode. Additionally, the output coupling can be viewed and optimized prior to aligning the second fiber to separate the two required alignments. The design of the components required to achieve this alignment scheme will now be discussed in the following sections.

### 7.1.1. Positioning Mechanics and Imaging Paths

This section will describe the basic design of the fiber positioning setup that was designed to vertically couple into the test chips. The surface normal microscope was chosen to be a Mitutoyo FineScope 70 with a Xenics Xeva-1.7-320 IR camera mounted on the trinocular port. The primary objective used in this imaging setup was a Mitutoyo 20x M Plan Apo NIR with a 20 mm working distance. This objective

was chosen to provide sufficient detail of chip features and a 0.4 numeric aperture to provide an angular acceptance cone of 23.6 degrees to allow viewing of the output coupler light. The moderate working distance allows for compact fiber holders to be positioned under this objective. Initially, the fiber arms were machined at fixed angles and adjustable only under the narrow angle range provided by the six-axis fiber positioning stage as shown in Fig. 7.1. Two magnets on the end of the positioning arms pull a Kapton-coated steel shim against the fiber in a precision cut V-groove for mounting. The distance between the fibers and the chip surface was then controlled by a standard color camera connected to a 40x zoom macro lens. To provide sufficient light for this camera, a separate illuminator connected to two lightpipe focusers were aimed at the coupling setup from the rear side. The focusers and the macro lens are visible behind the fiber arms at the center of Fig. 7.1.

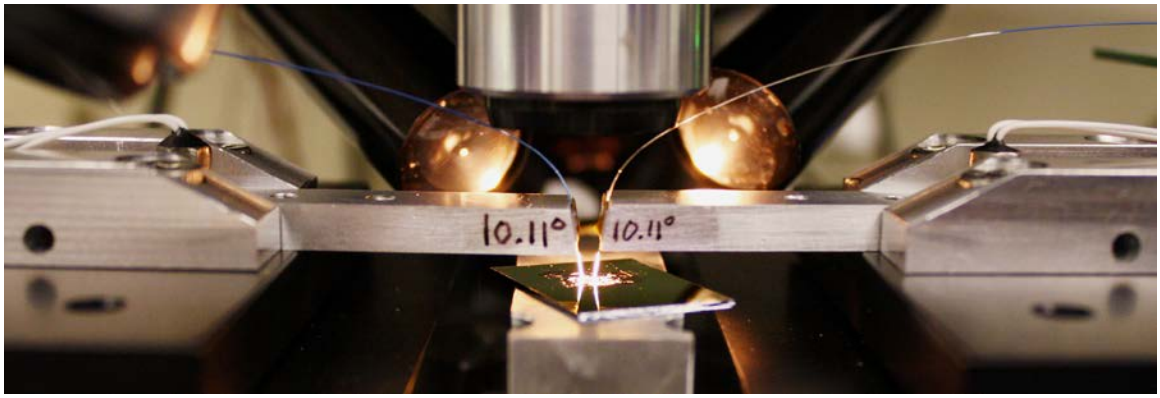


**Figure 7.1.** Photograph of the basic vertical coupling setup. The Mitutoyo 20x objective is visible at the top center of the image. Under this objective, fixed fiber positioning arms with 10.1 degree end angles hold two fibers above the chip surface. The illuminators and 40x macro lens are visible in the background. On either side, the arm holders allow the positioning arms to be extended any arbitrary length and held in position via a set-screw through the visible hole.

The sample under test is held down to the temperature-controlled vacuum-chuck shown in the bottom center of Fig. 4.1. The custom-machined aluminum chuck has three 0.050" holes drilled at a 0.1" spacing. These holes connect to a vacuum line drilled from the rear of the stage that terminates in a 1/8 NPT hose connection. The vacuum is provided by a Vaccon VDF 100 venturi-pump running off of the house compressed air line. The large pneumatic capacitance of the house air line and preregulator prevent low-frequency compressor vibrations from coupling to the vacuum stage. This situation is in contrast to traditional small vacuum pumps

traditionally used for vacuum chucks that generate significant pump vibrations that are difficult to isolate from the sensitive sample stage. The stage is temperature controlled by a Wavelength Electronics LFI3751 controller connected to a 3/8" 500 W cartridge heater and 10 kΩ thermistor in the front side of the chuck. Under normal test conditions, the stage temperature is maintained at 25 °C ± 0.02 °C and may be swept for thermal tuning. The sample positioning is then coarsely controlled by a Newport 562 three-axis stage. This positioning flexibility allows the relative port spacings of the fibers to be maintained when moving to another test location on the sample.

The relative positions of the fiber holding arms are then controlled by multi-axis nanopositioning stages. In the two different test setups constructed over the course of this thesis, two types of stages have been employed. The first was a six-axis New Focus 8095 kinematic stage. Six PicoMotor actuators that employ a slip-stick, piezo-drive to adjust fine pitch screws accomplish the stage positioning with a minimum incremental motion of less than 30 nm. The movable platform, which mounts the fiber-holder arm, is held down to these actuators by three springs. The directly screw-mounted design of this stage enables high stability and low drift as compared to traditional flexure stage designs. The chief downside of this design, however, is that multiple motors must be used simultaneously to provide pure X-, Y- or Z-axis motion. The variability in motor motion therefore prevents knowledge of the exact or even relative fiber position, preventing any attempt to automate the fiber positioning without additional feedback systems.

To enable a path towards automation, more traditional flexure stages were chosen for a second, complementary test station. To enable complete fiber positioning freedom two-axis tilt platforms (Thorlabs PY003) were attached to X-, Y- and Z-axis flexure stages (Thorlabs MAX301) for use with the variable-angle fiber mounts described in the next section. The positioning for the X-, Y- and Z-axes are controllable then by closed-loop piezo drive over 100 um range and up to 4 mm by interchangeable coarse actuators. For manual positioning, differential micrometers are used in the current test setup, but replacement with stepper motors would enable full coupling automation.

### 7.1.2. Compact, Variable-Angle Fiber Mounts

In order to enable a wider angular range than is available with the fixed angle fiber arms, a compact, variable-angle fiber holder was designed and fabricated as

shown in Fig. 7.2. To fit within the 20 mm working distance of the objective, a variable radius arc holder was machined to mount the fiber with minimal bending induced loss. A V-groove machined in the end of the holder ensures that the fiber is mounted straight. Superglue is used to affix the fiber to the holder along the curved section. The holder is attached to the main arm by a Riverhawk flexural pivot bearing (part number 6004-800). This pivot bearing, shown in detail in Fig. 4.3, provides a restoring spring force over a ± 20 degree travel to keep the arc holder pressed against the screw of a New Focus 8353 Tiny PicoMotor actuator. By placing the angular rotation pivot point close to the fiber tip, the displacement of the fiber tip when adjusting the coupling angle is made small.

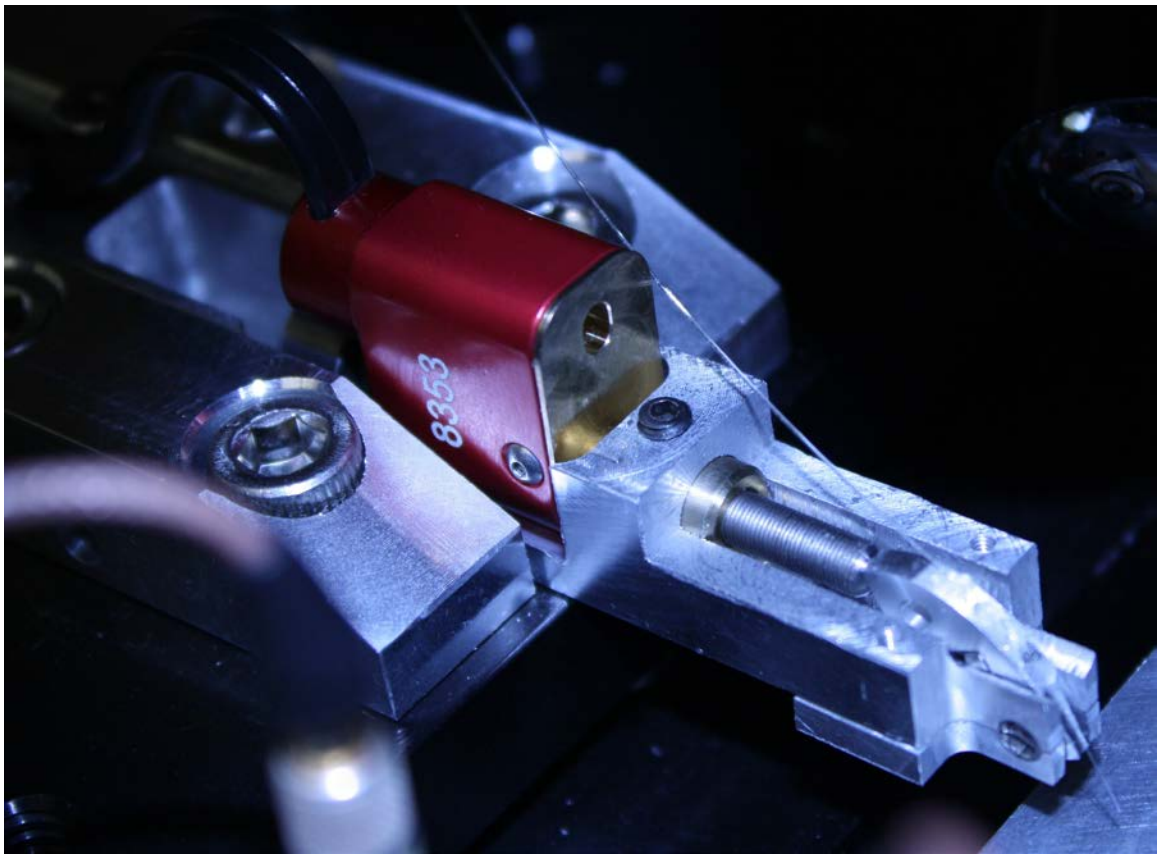


**Figure 7.2.** Variable-angle holder with fiber attached to curved arc mounted on a flexural pivot bearing is designed to operate within the working distance of the Mitutoyo objective. The New Focus 8353 Tiny PicoMotor is used to adjust the angle relative to the sample over a 20 degree range.

The end section of the variable angle mount is designed to allow simultaneous electrical probing of the device under optical test. First, the fiber tip is extended 0.2" beyond the end of the holder. Enabled by the compact arc holder, this clearance

allows many standard varieties of DC electrical probes to fit under the fiber arm. This clearance alone is not sufficient for the much larger RF probes that are required to enable microwave testing up to 40 GHz. To resolve this issue, the normal 0.5" wide arm is narrowed to 0.2" to clear the body section of the Cascade Microtech line of Infinity Microwave probes to be used for high-speed optoelectronic device testing.

## 7.2. Calibrated Optical Measurement from 1260 nm to 1630 nm

In this section, I will describe the optical test and calibration paths in the setup that have been optimized over the full wavelength range of all common NIR telecommunication bands. The diagram for the test setup using SMF-28e fiber paths is shown in Fig. 7.3.

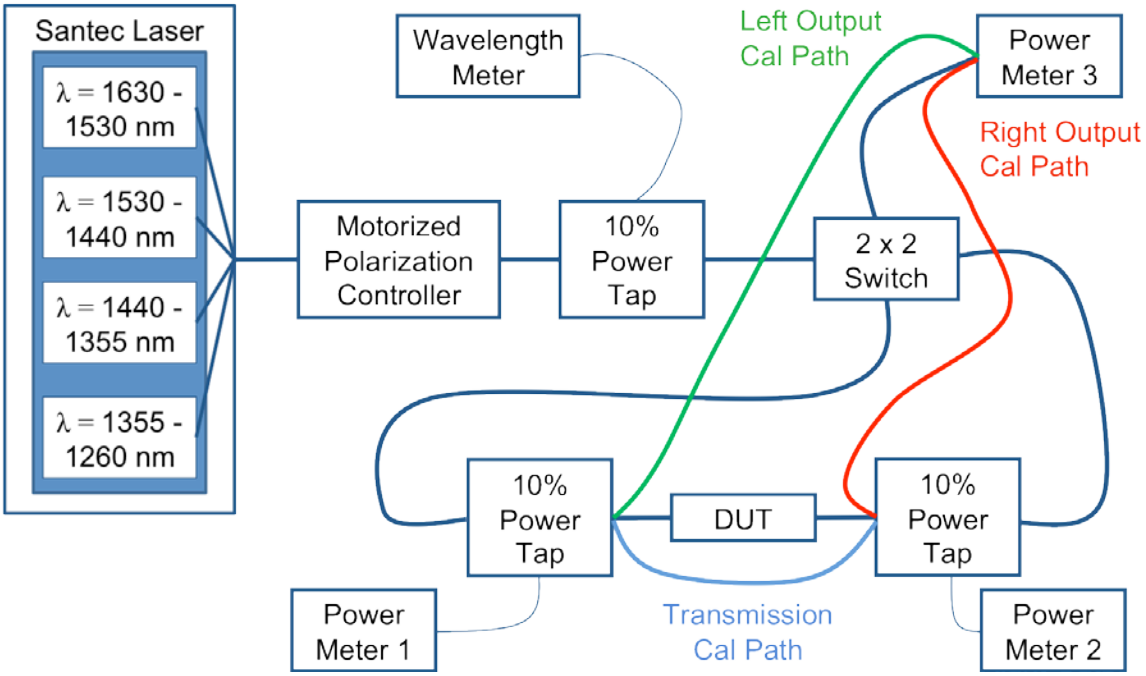


**Figure 7.3.** Test station diagram illustrating the test and calibration fiber paths for the broadband test station.

The calibrations used for the test station will also be discussed in detail. This will include the fiber-to-fiber calibrations as well as the photodiode responsivity calibrations that utilize a NIST calibrated Ge detector for absolute power measurements at the output of the lensed fibers.

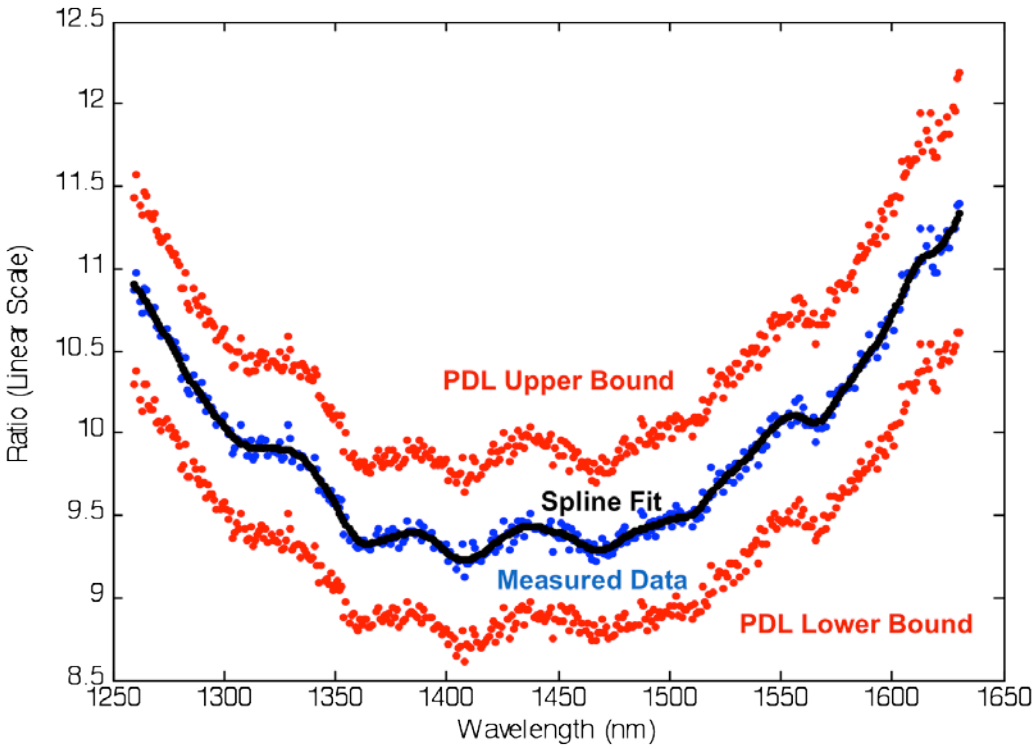


**Figure 7.4.** Calibration ratio for the left-to-right calibration path. The polarization dependent loss bounds (PDL) and average ratio are determined by the min, max, and mean of 100 polarization states swept by the motorized polarization controller.

The PDL uncertainty, while a significant consideration for most measurements as shown in Figs. 7.4 and 7.5, can be removed by performing a measurement specific calibration at a specific operating polarization state. The majority of the uncertainty stems from the 10/90 power splitters used in the broadband test setup. Lower PDL uncertainty could be obtained by using power splitters that have been optimized for use over a narrow wavelength range.

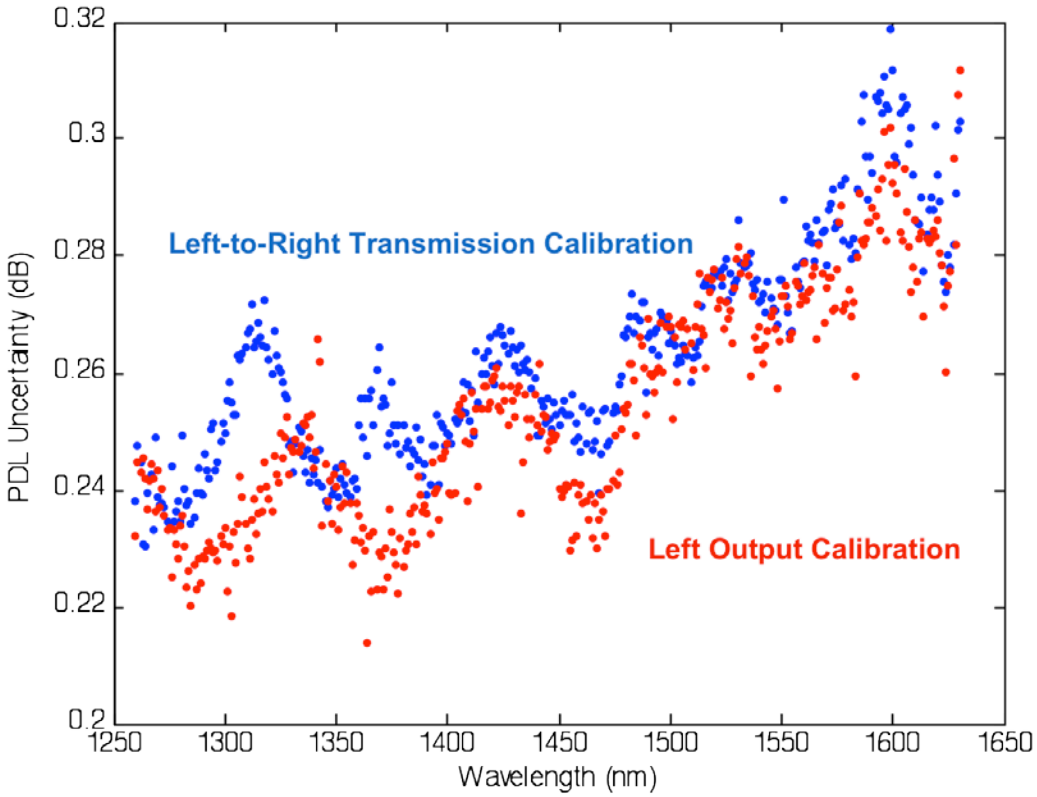


**Figure 7.5.** Absolute polarization dependent loss (PDL) uncertainty calculated from 100 randomly chosen polarization states for the left-to-right optical paths.

# 8. APPENDIX B: POST-PROCESSING

The requirement to remove the silicon substrate underneath the photonic integration regions has led to the creation of several post-processing techniques. This appendix will outline first the localized substrate removal techniques employed, focusing on the formation of in-process etch vias. Next, full substrate removal processing for optical-only testing will be discussed. Finally, the substrate transfer process will be described.

## 8.1. Localized Substrate Removal

The chief benefit of the localized substrate removal process is that the electronic local environment is completely identical to the as fabricated die. No deviation from the intended transistor characteristics is conceivable in such a post-processing solution since the peak processing temperatures never exceed 100 °C. The first

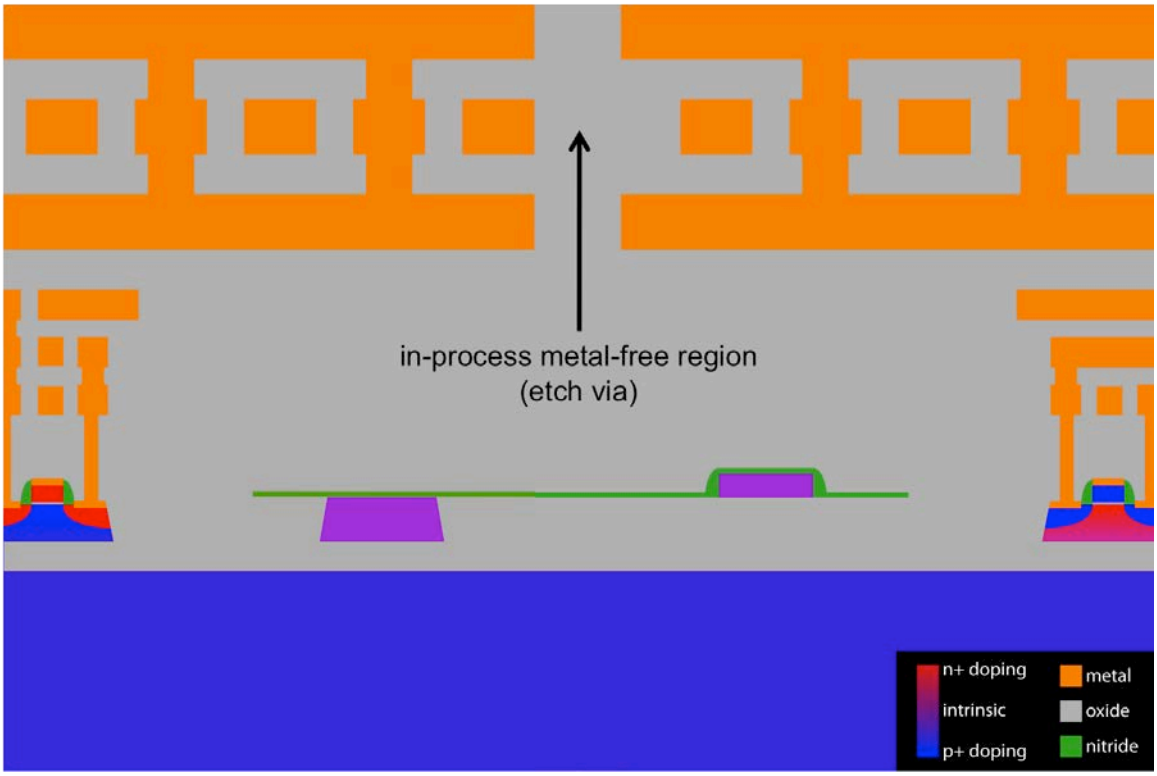


**Figure 8.1** Process cartoon for the front-side localized substrate removal flow: as fabricated cross-section identifying the metal-free region designed into the layout.

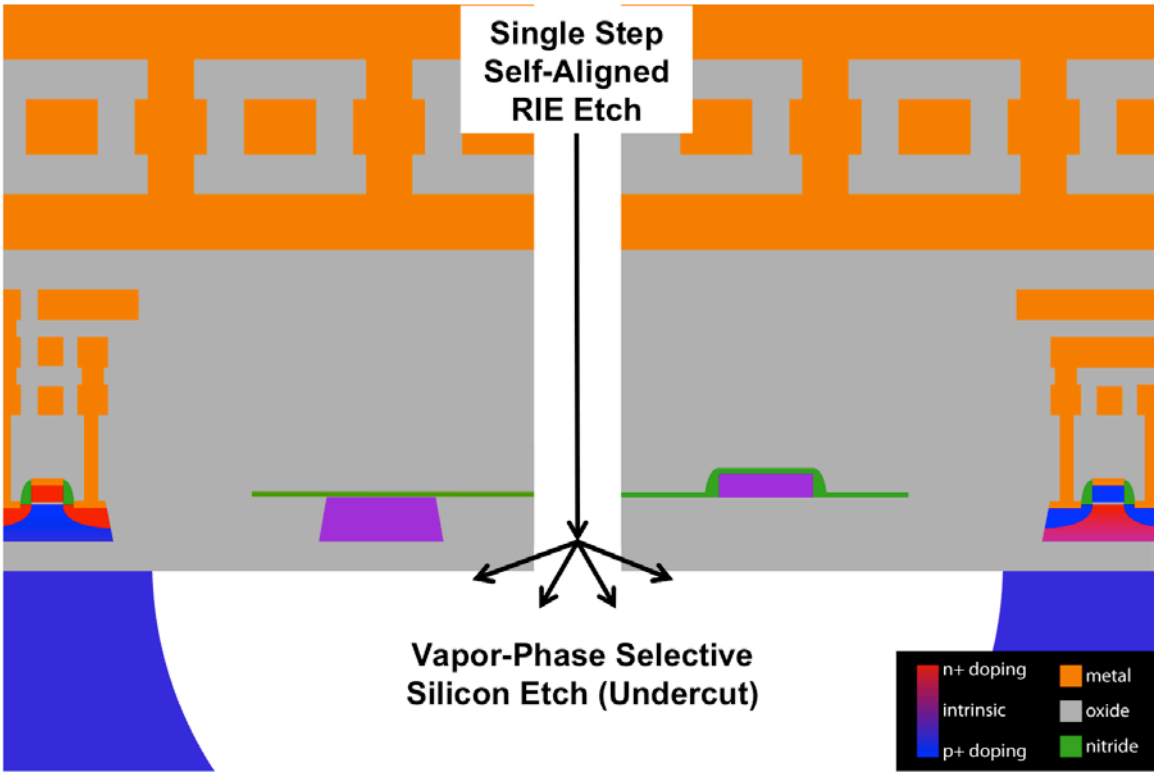


**Figure 8.2** Process cartoon for the front-side localized substrate removal flow: two-step etch process.

method attempted for local substrate removal is a two-step etch from the front-side of the die or wafer. As shown in **Figure 8.1**, a metal-free region is included in the layout in the center of the photonic integration region. After normal foundry fabrication, reactive ion etching (RIE) of the dielectric layers is used to expose the silicon substrate under these "etch vias" from the front side of the die or wafer. A previous etch recipe for this process, which was performed by Charles "Trey" Holzwarth and Jie Sun, was to use $CF_4$ gas at a bias of 250 V in five minute intervals for a two hour total etch time. The second half of the etch was to undercut the photonic region by etching the silicon substrate through these holes using a pulsed etch of highly-selective, vapor-phase $XeF_2$. A typical recipe, which was performed by Hanqing Li, was to utilize 10 s etch followed by 50 s pump cycles for a total etching time of 430 s to remove the underlying silicon. The final result of the process is shown in **Figure 8.2**.

An important aspect of this process is the design of the dielectric etch vias to enable front-side etch access. As shown in **Figure 8.3**, the nominal design for the test chips discussed over the course of this thesis has been to place photonic devices in a

~80 µm wide stripe around a row of 5 µm x 5 µm etch vias. The etch vias are formed by excluding the metal fill from the desired dielectric region and compensating local pattern density with high surrounding metal density. Next, access through the thick passivation layers formed on top of CMOS stackup is achieved by placing dummy aluminum pads over the dielectric etch vias. In the top surface micrograph of **Figure 8.3**, the dummy aluminum pads have already been etched to expose the vias. Since the minimum feature size of the aluminum pad layer is significantly larger than 5 µm, a single aluminum pad is used to expose multiple etch vias.

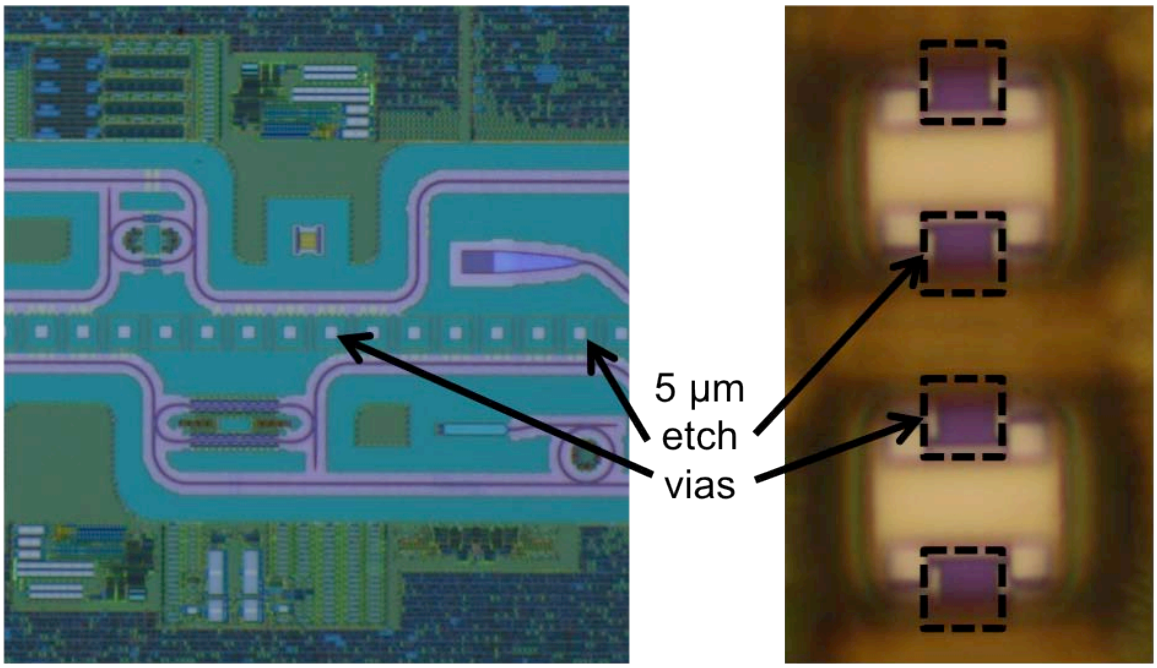


**Figure 8.3** Micrograph from backside and front side of the EOS8 test chip identifying etch vias.

The removal of the dummy aluminum pads are achieved by a wet etch process that proceeds before the two-step etch described in the previous sections. If no photoresist is required to mask the wet etch of the dummy pads, a single wet etch step of 5 N NaOH heated to 80 °C to which a 5% solution of $H_2O_2$ is added for a 5 minute etch will remove all metallic layers present in the dummy pads. If a selective wet etch is required so as not to damage photoresist on the wafer or die, a two part wet etch of Transene Aluminum Etch A and Transene Tantalum Etch 111 is required to remove both the aluminum pad and its Ta / TaN adhesion layer.

As an alternative to the front-side localized substrate removal process described here, back-side trench etching to locally remove silicon substrate regions has

recently been studied as an alternative. As shown in Figure 4.48, trenches etched from the back-side of the die can expose the photonic integration regions without the dielectric via-etch process. In this case, the photonic regions are patterned onto the back of the die using photolithography. To align the photomask, an IR viewing contact aligner is preferable, but a standard contact aligner may be used by aligning to the diced corner positions. Once the desired pattern is created, a two-step etch of reactive ion etching through the majority of the substrate thickness followed by $XeF_2$ etching of the final ~10 μm of silicon is employed. Depending on the selectivity of the RIE process to the photoresist, it may first be required to transfer the photoresist pattern to an oxide or metal hard mask for the RIE etch step. Although the back-side process option has been demonstrated on the CMOS test die, it is not believed to be as scalable a process when compared to the front-side, self-aligned undercut process described first.

## 8.2. Full Substrate Removal

For optical device testing, it is not required to ensure that the silicon substrate removal step is localized to the photonic regions. For such cases, complete substrate removal has been used to speed device testing and ease sample preparation. Since the selectivity of the $XeF_2$ gas etchant is greater than 1000:1 for silicon as compared to silicon dioxide, $XeF_2$ is used to remove the silicon substrate even though its vapor-phase properties for undercut release is not required in this case.

Since $XeF_2$ is an exothermic etch, the die for full substrate removal is mounted to an oxidized silicon wafer to act as a heat sink during etching. For easy removal, the standard full substrate removal recipe, which was developed by Hanqing Li, the die is mounted to the wafer using Crystalbond 509. Since Crystalbond melts at 125 °C, if the die completes the full substrate removal without the Crystalbond reflowing, peak processing temperatures well below the maximum allowed post-backend CMOS processing temperature of 400 °C are guaranteed. The etch progress is monitored optically at various time points until the substrate is observed to be removed.

The sample often requires thorough cleaning after the $XeF_2$ etch process. First of all, the etch chamber cleanliness of the tool is degraded by many users of the shared equipment using many different materials that produce particulates. Second, the back-ground die received from a CMOS multi-project wafer run has significant contamination in the substrate that end up on the film surface after substrate

removal. The residual Crystalbond around the CMOS film can crumble once the substrate is removed and fall onto the film surface as well. Finally, in order to have a clear view of the film surface in the test setup side-view camera, all residual Crystalbond surrounding the die must be removed so as not to block the view of the fiber relative to the film. To clean the samples, a solvent sequence of acetone, methanol, isopropanol and DI water is used. Often, some residual particulates are not removed by the solvent rinse sequence. In these cases, the final cleaning step has been to provide some mechanical removal by wiping the chip surface with a cleanroom Q-tip under a running stream of DI water.

## 8.3. Substrate Transfer

An intermediate solution to enable electro-optic testing was developed based on the full-substrate removal process. Fundamentally, when the substrate removal process occurs, the electrical pads required to connect to the integrated circuits are facing the silicon substrate. Additionally, there is no thermally or electrically conductive substrate under the electronic regions to enable proper circuit functionality. The substrate transfer process is designed to replace the silicon with a suitable thermally and electrically conductive substrate separated by a thin layer of low-refractive-index, transparent adhesive and allow the transferred layer to be removed from the processing handle wafer to enable access to the top-surface electrical pads.

In the first step, shown in **Figure 8.4**, the received CMOS die is mounted pad-side down onto an oxidized silicon wafer using Crystalbond 508 as in the full substrate removal process. The next step also follows the full substrate removal process. As shown in **Figure 8.5**, the exposed silicon substrate is fully etched using $XeF_2$. Following on the full substrate removal, a small droplet, roughly ~200 μm in diameter, of Norland Optical Adhesive 71 (NOA71) is placed in the center of the exposed CMOS film. A diced 3x3 mm (the same size of the CMOS film to be transferred) piece of 6H silicon carbide is placed onto the drop. If the correct quantity of NOA71 is present on the film, the entire interface will be wetted with minimal overflow to ensure that a thin layer of material is present. Inspection under an optical microscope is possible at this point to verify expectations and the process can be repeated if acetone is used to remove the uncured NOA71. If the process appears to be proceeding correctly, the NOA71 film can then be cured under a UV lamp. Using a 6W lamp, typical cure times of 1 hour have been common. The

resulting state of the process is shown in **Figure 8.6**. The solvent resistance of the NOA71 bond between the 6H-SiC and CMOS film can be improved by a 12 hour 50 °C anneal to activate the cross linking agent within the NOA71. Next, the bonded film stack can be removed from the handle wafer by melting the Crystalbond 509 on a 100 °C hot plate. After cleaning off the Crystalbond 509 residue using acetone followed by methanol, isopropanol and DI water rinses, the transferred die is ready for optoelectronic testing or packaging as shown in **Figure 8.7**.

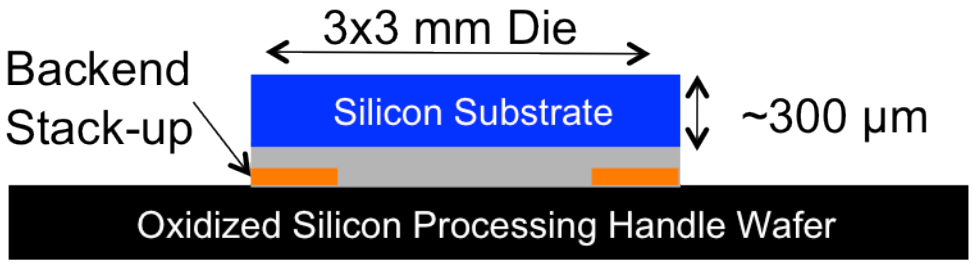


**Figure 8.4** Step 1 of the substrate transfer process. The received CMOS die is mounted pad-side down onto an oxidized silicon wafer using Crystalbond 509 as in the full substrate removal process.

**Figure 8.5** Step 2 of the substrate transfer process. The exposed silicon substrate is fully etched using $XeF_2$ as in the full substrate removal process.


**Figure 8.6** Step 3 of the substrate transfer process. A 6H-SiC lid has been bonded to the released CMOS film using NOA71.

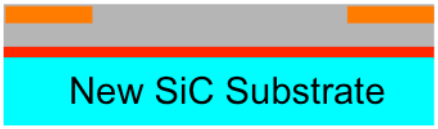


**Figure 8.7** Step 4 of the substrate transfer process. The transferred die has been released from the processing handle wafer and is suitable for optoelectronic testing or packaging.

The transferred dies have been observed to be very mechanically stable on the new substrate. Aggressive probing using DC needle probes and Cascade-Microtech i-GSG-100 probes have not resulted in any observable film damage. Wirebond packaging has also been successfully performed with no evidence of film damage. It is important to note that wirebond yield difficulties have been observed in cases where the cleanliness of the bond pads on the transferred die have been compromised during processing.

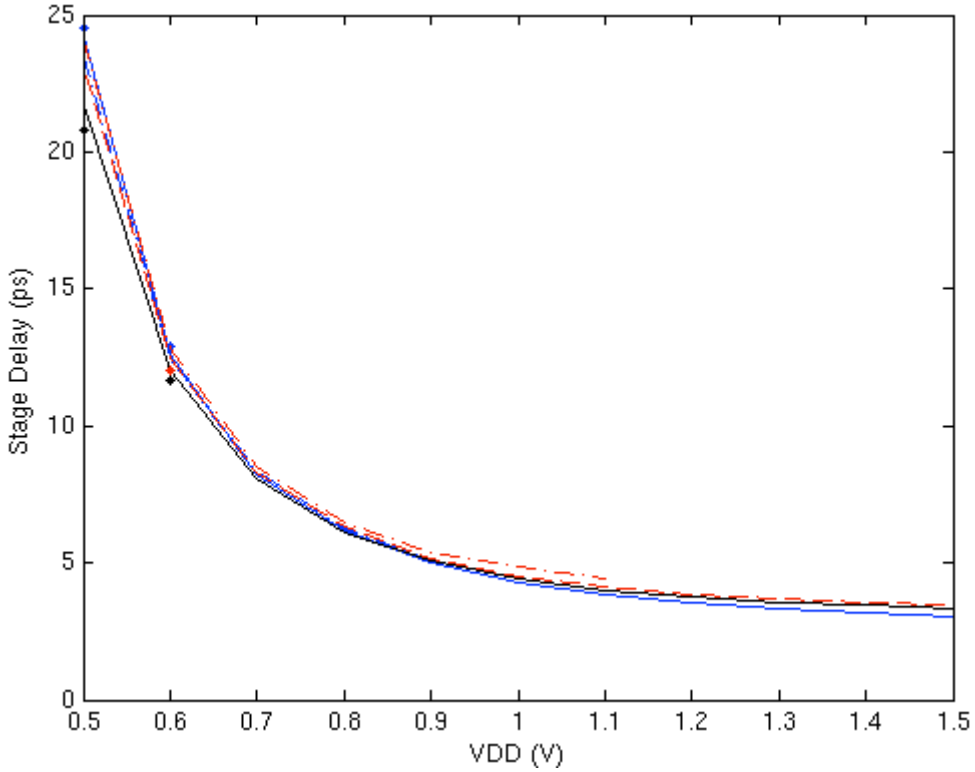


**Figure 8.8** As shown in the above plot of stage delay for several ring oscillators both before and after substrate transfer, it is observed that not only do circuits still function correctly, but also there is no noticeable degradation in speed.

On the SOI test chips where the transistors are separated from the fabrication silicon wafer by a >100 nm silicon dioxide layer such that they are not damaged during full substrate removal, complete electronic functionality has been observed. By electrically probing multi-stage ring oscillators fabricated on the die, the logical stage delay of the integrated transistors were measured before and after substrate transfer. As shown in **Figure 8.8**, the transistor speed varies by less than 5% before and after processing. Additionally, after wirebond packaging, the complete, ~3

million transistor digital backend included on the EOS8/EOS10 test chips has been verified with no observed changes in functional performance on a fully substrate transferred die.

# 9. APPENDIX C: DESIGN AND TESTING OF AN EXTERNAL CAVITY LASER CONTINUOUSLY TUNABLE FROM 1150 NM TO 1260 NM

Over the course of the work associated with this thesis, some commercial tunable lasers in the wavelength range near 1220 nm. Example lasers are the fully packaged NewFocus TLB-7023 (1220-1250nm) and more kit-based construction lasers such as the Sacher-Laser TEC-100-1260-10 (1240-1310nm) and Thorlabs TLK-L1220R (1180-1280nm). At the time that the first integrated silicon germanium photodetector, no commercial tunable lasers were available to cover any wavelength range below 1260nm, the cutoff of the telecommunications O-band.

By attending trade shows at conferences, I found that Covega had grown gain media wafers centered near 1200nm. Several initial chip-on-heatsink single-angled-facet gain media were purchased to construct early versions of an external cavity laser using flexure-aligned lensed fiber and manual tuning mirror positioning. After several design revisions and a conversion to a half-butterfly packaged gain media that is now available from Thorlabs after their acquisition of Covega, a reliable external cavity laser that can be continuously tuned from 1150 nm to 1260 nm was constructed.

## 9.1. Theory of Operation

The external cavity laser gain media used for this project contain both a flat, mirrored facet and an angled anti-reflective facet. In the absence of any back-reflection at the angled facet from an outside source, the gain media does not lase. Instead it will only act as an amplified spontaneous emission (ASE) source when pumped by current. A cartoon diagram of the gain media is shown in **Figure 9.1**.

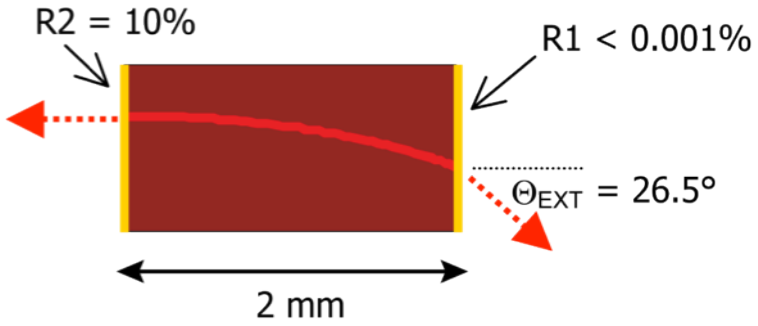


**Figure 9.1** Single angled facet used for external cavity laser.

A Fabry-Perot laser containing many transverse modes near the peak of the gain spectrum could be created by placing an external mirror such that a significant quantity of light was coupled back into the gain media. In order to create a single-mode, tunable laser, a wavelength selective back reflection must be used. The traditional method of doing this is to use a diffraction grating as the back mirror. If used in the configuration as shown in **Figure 9.2**, only light that is incident at the Littrow angle of the grating will be reflected back into the laser cavity. The Littrow angle is given by the following relation:

$$\theta_{Littrow} = \arcsin\left(\frac{\lambda}{2 \bullet grating\_pitch}\right)$$

If the wavelength of the light is too long relative to the grating pitch such that the argument of the arsin becomes greater than 1, the first-order retroreflection of the Littrow angle does not exist. This sets a maximum wavelength for any given grating for this mode of operation.

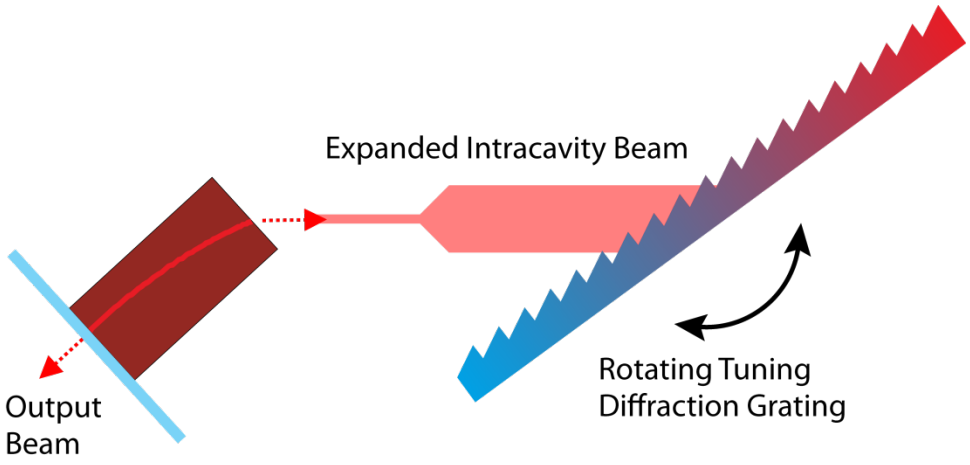


**Figure 9.2** Littrow configuration external cavity laser.

Since the goal of introducing the grating into the laser system is to provide a wavelength selective reflector, it is desirable to operate near the maximum wavelength of the grating. At this near grazing incidence operating point, the dispersion, given by the following relation, is maximized:

$$D_{grating} = \frac{2 \tan \theta_{incident}}{\lambda}$$

In order to operate in a single transverse mode, the spectral width of the back-reflected signal must be small relative to the spacing of the intracavity Fabry-Perot modes. Therefore, in order to maximize the separation between the longitudal

modes, the grating must be placed as close as possible to the gain media. The cavity design problem can be seen as the product of the Fabry-Perot and grating back-reflection signal as shown in **Figure 9.3**.

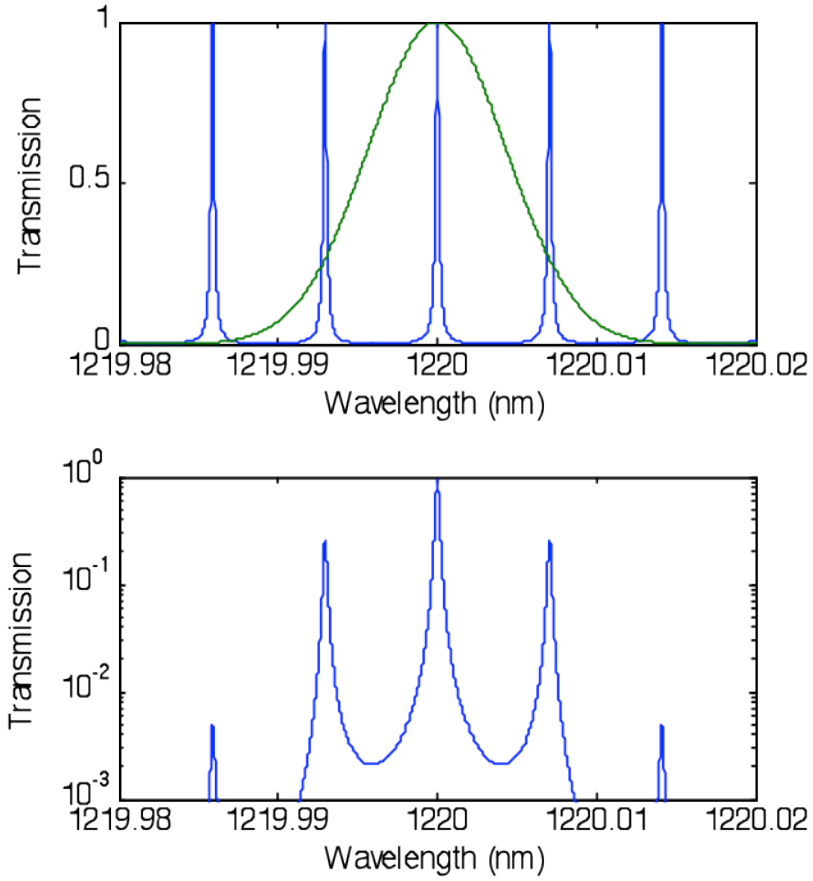


**Figure 9.3** (a) Overlaid cavity Fabry-Perot modes and grating back reflection signal. (b) Product cavity spectral response for illustration purposes.

## 9.2. Cavity Optical Components

For optimal Littrow cavity operation near 1200nm, a grating with a line density on the order of 1200 lines/mm is desired. Originally, a 1200 lines/mm ruled grating with a 1 μm blaze was purchased from Thorlabs (GR25-1210). Since the gain media emits polarized light, high back reflection efficiency (82% specified by manufacturer) was observed near 1200nm for the perpendicular input polarization. However, the grating exhibited unstable tuning characteristics and did not result in single transverse mode operation as measured by a self-heterodyne measurement on an microwave spectrum analyzer with a lightwave front end. Significant performance improvements were observed when the initial grating was replaced

with a 1480 lines/mm, gold-coated, etched silicon holographic grating from LightSmyth (SLG-C14.8-2410-AU). After replacing the grating, no ghosting or multi-mode operation was observed.

Since the light diverges immediately out of the waveguide facet of the laser gain media, an intracavity collimating lens is required to achieve a high back reflected power. Since the light diverges over a wide angular spread, a high numeric aperture lens is required. Additionally, since it must be focused into a ~3 μm mode area, diffraction-limited performance is required for high efficiency. After several lens variations, the 355660-C unmounted aspheric lens was purchased from Thorlabs.

## 9.3. Cavity Construction

Two revisions to the physical construction of the external cavity have been made. The first version of the cavity utilized a die-on-heatsink gain media package. The heatsinked gain media was mounted on a copper block that was thermally controlled using a TEC with thermistor feedback. The "hot side" of the TEC was connected to a natural convection heatsink that was mounted to the optical breadboard using Invar support rods to minimize thermal expansion drift. A 3 μm spot-size lensed fiber was used to couple to the reflecting facet. The fiber was coarsely aligned visually through a stereo microscope before being fine-tuned by coupled optical power measurements. Laser safety glass with OD7+ protects the viewer from any laser light while looking through the microscope. Photographs of the cavity construction are shown in **Figure 9.4**.

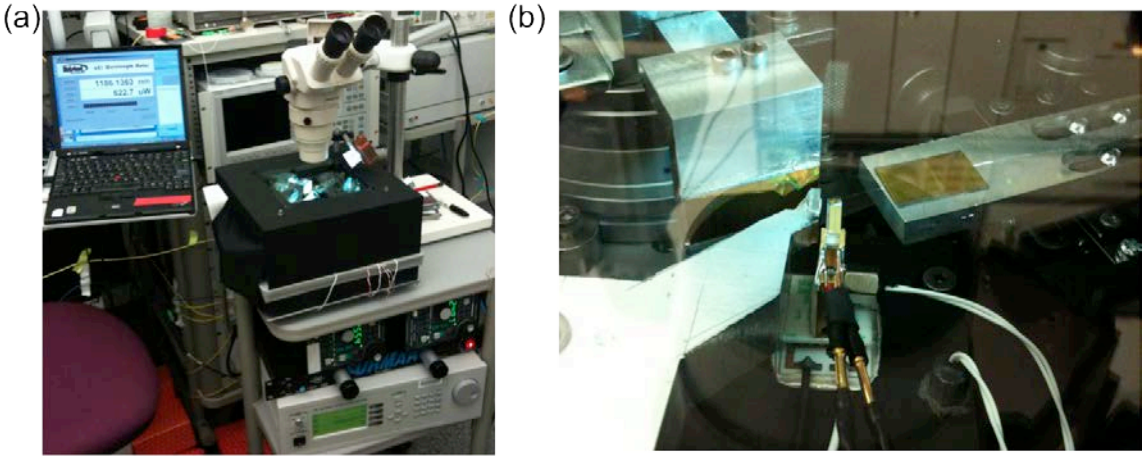


**Figure 9.4** (a) Photograph of die-on-heatsink external cavity laser. The setup was built on a cart to enable portability while including a stereo microscope for fiber alignment. (b) Photograph of laser cavity showing fiber coupling, gain media, intracavity lens and grating.

A revised version of the laser cavity was built when a fiber pigtailed gain media became available from Covega through Thorlabs. This eliminated the manual fiber coupling alignment of the first design. Additionally, the pigtailed package included an internal TEC cooler to simplify the thermal mounting. The revised cavity design is shown in **Figure 9.5**. At this cavity revision, a motorized tuning arm was added using a RS65 stage from Newport with a DC servo motor. The thermal design of the enclosing cavity was tuned by thermal isolation and thermally stabilized copper baseplate controlled by a large TEC with natural convection heatsink. The aluminum fins of the TEC heatsink are shown in **Figure 9.5**(a). The cavity is thermally controlled near room temperature for maximum isolation from variable cooling as ambient airflow changes.

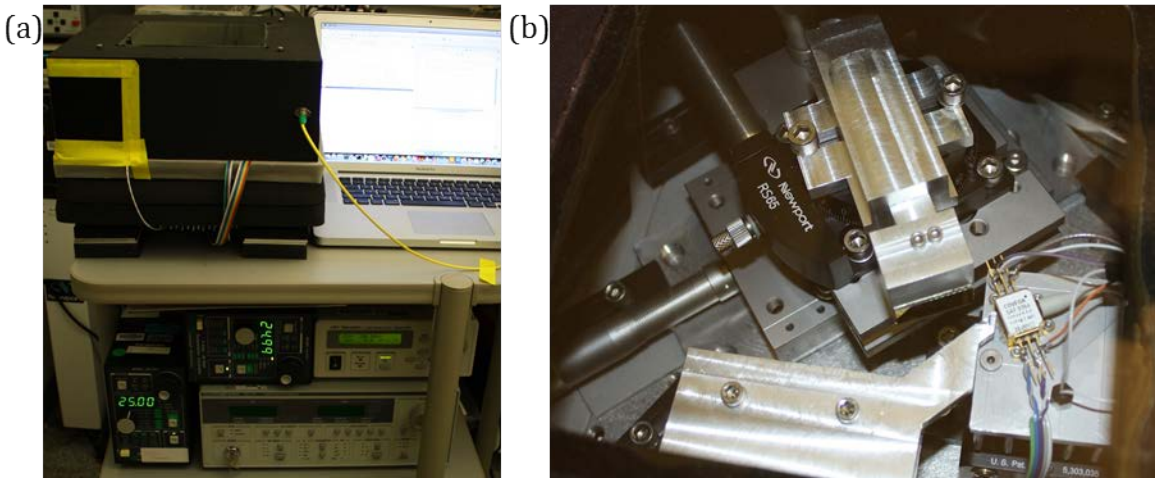


**Figure 9.5** (a) Photograph packaged gain media external cavity laser. (b) Photograph of laser cavity showing gain media, intracavity lens, grating with DC servo motor.

## 9.4. Cavity Alignment

The laser cavity must be aligned periodically to minimize intracavity loss. The standard procedure for the alignment is to first move the diffraction grating out of the path of the collimated beam. The laser should be biased above threshold but below maximum power to prevent damage. Next, the collimation is optimized by adjusting the position of the lens. Placing an IR viewing card very far from the laser (~10m) enables the collimation to be fine-tuned. Additionally, care should be taken that the beam is passing normal to the central axis of the lens and not refracting from this central axis. Once the collimation is optimized, the grating is replaced in

the optical path at a tuning angle that is near the center of the laser wavelength range. The tilt of the grating should then be adjusted until a lasing mode is observed on the OSA. Using the IR card to assess whether or not the specular reflection is deviating from the horizontal plane can speed this search.

## 9.5. Measured Performance Characteristics

The laser has been confirmed to maintain a single-mode operation at a stable operating point by multiple self-heterodyne measurements. The cavity mode spacing is observed to be 4.6 GHz corresponding to a free-space equivalent path length of 3.2 cm. The wavelength range without adjusting the fine-tuning lock point of the rotation stage is approximately 90nm. The output wavelength as a function of DC motor position as measured by a Bristol 621B wavemeter is shown in **Figure 9.6**.

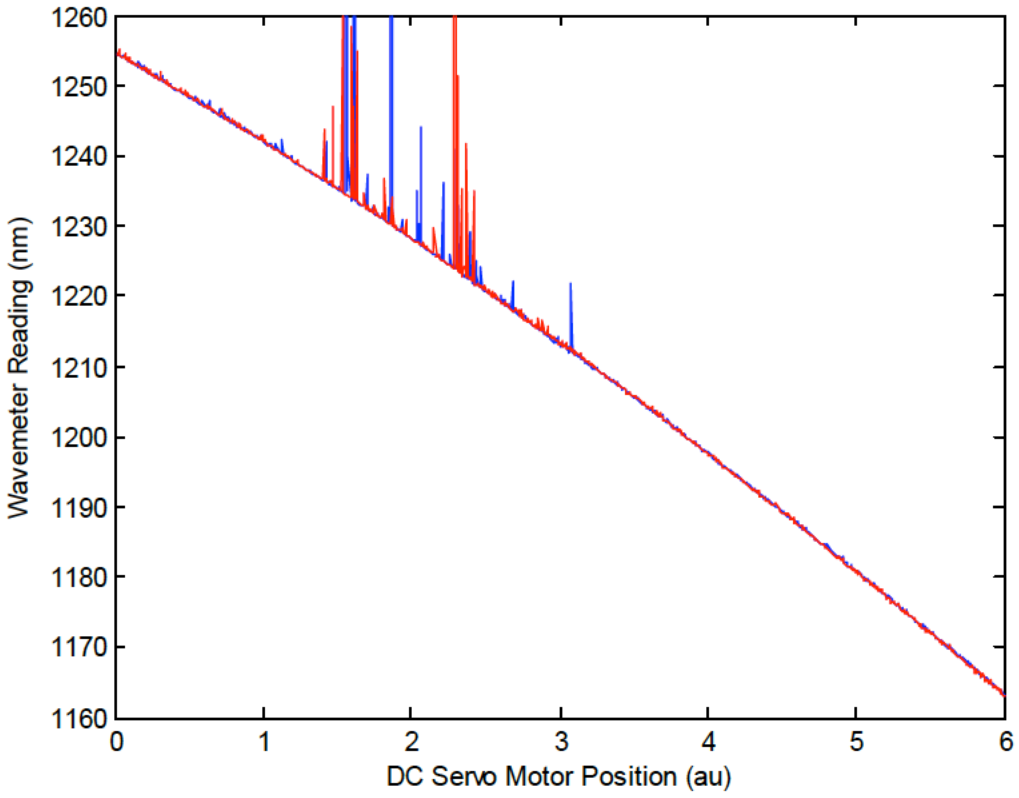


**Figure 9.6** Wavelength as a function of DC servo motor for bidirectional scans. The range where spurious peaks are observed corresponds to points where the laser hops between longitudinal modes. The actual wavelength does not deviate from the linear ramp by more than 0.1 nm (instrumentation limited) as observed by an optical spectrum analyzer.

The output power over the continuously tuned wavelength range is shown in **Figure 9.7**.

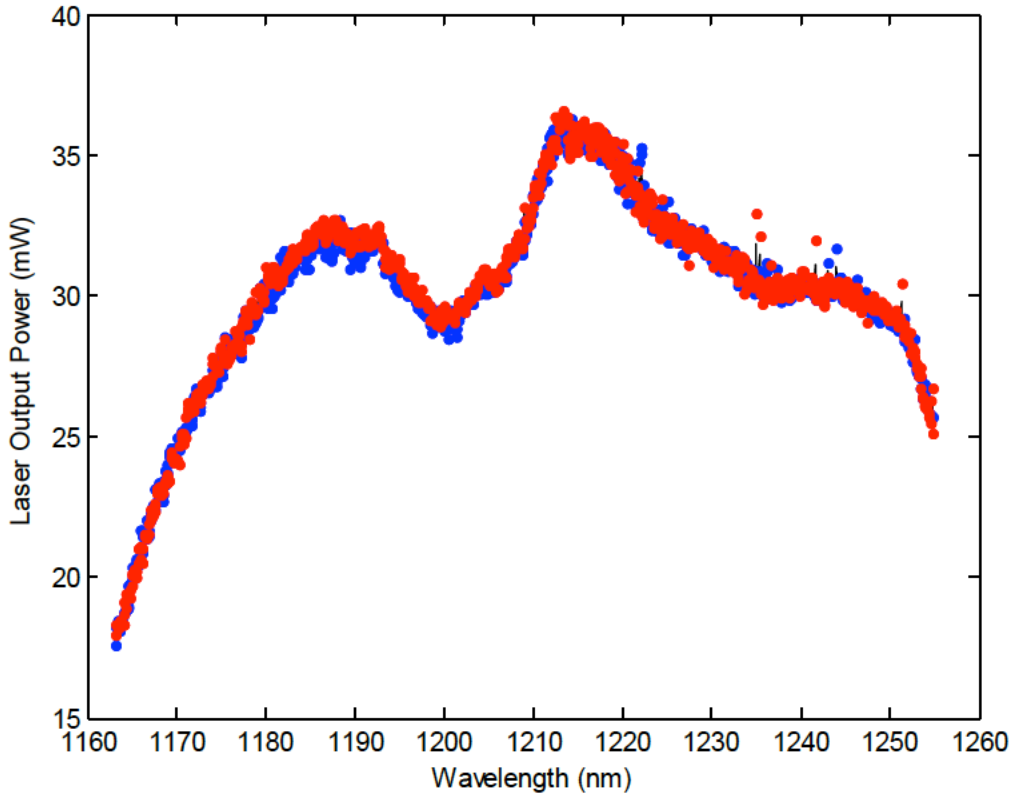


**Figure 9.7** Output power as a function wavelength for bidirectional scans. The two overlaid scans were taken 10 minutes apart.

The thermal stability of the cavity has enabled long term operation without wavelength drift. The power fluctuations can be eliminated by monitoring the power with a fiber tap coupler, but it is critical for reliable measurements to have wavelength stability. The overnight (12h) wavelength traces demonstrated repeatable ~5 pm stability. Experiments resulting in similar stability traces such as the one shown in **Figure 9.8** were repeated three nights in a row.

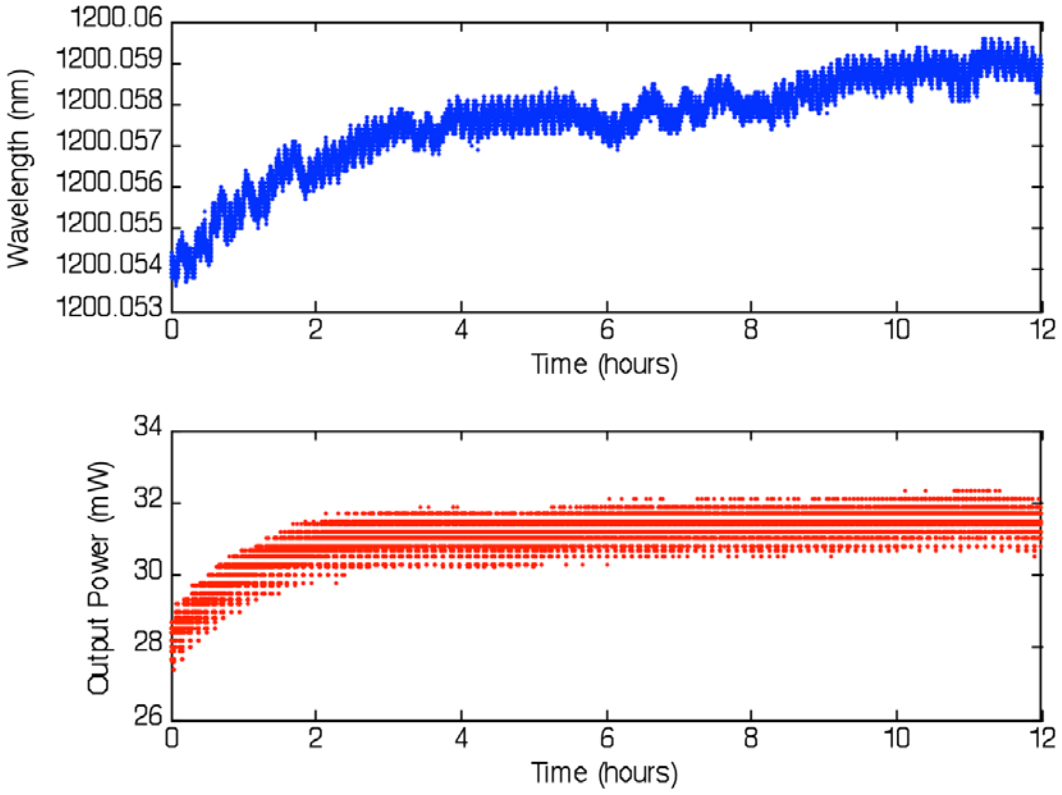


**Figure 9.8** Overnight laser stability measurements for wavelength and power.

## 9.6. Manufacturer-Supplied Gain Media Test Data

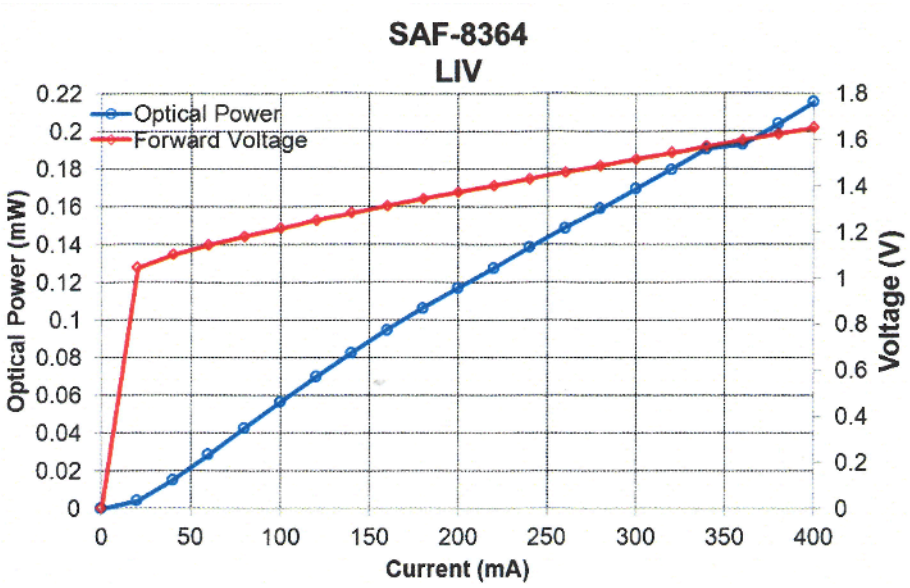


**Figure 9.9** L-I characteristics for Serial Number SAF-8362-11719.7.A01.

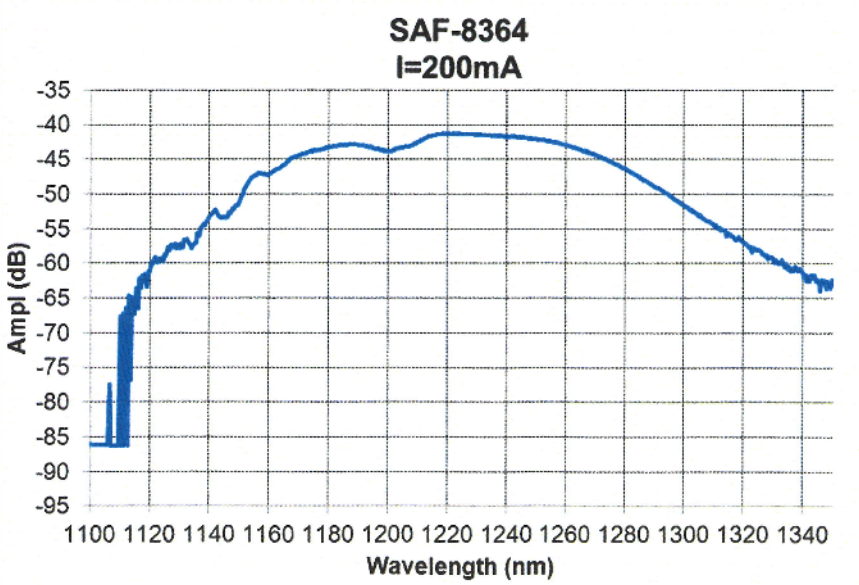


**Figure 9.10** Amplified spontaneous emission (ASE) characteristics for Serial Number SAF-8362-11719.7.A01.

# 10. APPENDIX D: CODE REFERENCE

Code generated used for device simulation and analysis over the course of this thesis. This also includes a protected-technology-stripped copy of the Cadence Virtuoso p-cells used for photonic layout. Note that due to the length of the relevant code, not all of the code is included, but merely representative cells capable of fully generating a single top level layout of representative complexity and all global functions. All global utility functions are included as well as geometry creation functions for a few basic shapes. Parameterized cells for geometry, waveguide routing and top level cell hierarchy levels are included. A complete set of shapes are included to enable the simple version of the resonator_cell p-cell used for ring resonator layout for test chips. Included cell lays out single or double ring resonators while full version also handles racetrack modulators, photodetectors and custom ring discs with or without variable couplers and heaters.

## 10.1. Matlab Code

### 10.1.1. first_order_ring_bank_32channel.m

```
% Transmission matrix resonator code used to simulate the
% predicted 45nm SOI filter bank.
% Based on work done in conjunction with Eugen Zgraggen for modulator simulation

lambda_target = 1.225e-6;
lambda_span = 50e-9;
lambda_steps = 100001;
coupling_lambda_target = 0.018;
coupling_slope = 0.005; % Fractional change in coupling per nm

num_filters = 32;
initial_radius = 3.797e-6;
radius_step = 3.0e-9;

wg_loss_dB = 3; % dB/cm

lambda_low = lambda_target-lambda_span/2;
lambda_high = lambda_target+lambda_span/2;
lambda_res = (lambda_high-lambda_low)/(lambda_steps-1);
lambda = lambda_low:lambda_res:lambda_high;
```

```
coupling_half_delta  = coupling_lambda_target*coupling_slope*1e9*lambda_span/2;
coupling  = linspace(coupling_lambda_target-
coupling_half_delta,coupling_lambda_target+coupling_half_delta,lambda_steps);

kappa1  = sqrt(coupling)*exp(1i*0*pi/180);
kappa2  = kappa1;
t1 = sqrt(1-abs(kappa1).^2)*exp(1i*0*pi/180);
t2 = t1;
a_loss  = log(10)*wg_loss_dB/10*100/2;  % conv. /cm to /m and /2 field
[nwg nwg_straight]  = load_neff(lambda*10^6);

bank_trans  = zeros(num_filters+1,size(lambda,2));
figure(1)
for filter_index  =1 num_filters
   rwg = initial_radius  + radius_step*(filter_index-1)
   a_round  = exp(-a_loss*2*pi*rwg);

   Et1 = zeros(1,size(lambda,2));
   Et4 = zeros(1,size(lambda,2));

   for m = 1:size(lambda,2)
      a = exp(2*pi*nwg(m)/lambda(m)*rwg*2*pi*i)*a_round;
      alpha  = abs(a);
      phi = angle(a);
      Et1(m)  = (t1(m) - conj(t2(m))*alpha*exp(-1i*phi))/(1-conj(t1(m))*conj(t2(m))*alpha*exp(1i*phi));
      Et4(m) = -conj(kappa1(m))*kappa2(m)*sqrt(alpha)*exp(1i*phi/2)/(1-conj(t1(m))*conj(t2(m))*alpha*exp(1i*phi));
   end

   % from  now on Etf and Edrop  are  power
   Etf = abs(Et1).^2;
   Edrop  = abs(Et4).^2;
   Etf_dB  = 10*log10(Etf);
   Edrop_dB  = 10*log10(Edrop);

   bank_trans(filter_index+1,:)  = bank_trans(1,:)+Edrop_dB;
   bank_trans(1,:)  = bank_trans(1,:)  + Etf_dB;
   plot(lambda*1e9,Edrop_dB,'b-')
   hold  on
end

figure(2)
plot(lambda*1e9,bank_trans(1,:),'k')
hold  on
for filter_index  =2:num_filters
   plot(lambda*1e9,bank_trans(filter_index+1,:),'b-')
   hold  on
```

```
end
plot(lambda*1e9,bank_trans(2,:),'r-')
hold off
xlim([1230 1250])
ylim([-30 0])
xlabel('Wavelength (nm)')
ylabel('Transmission (dB)')
```

### *10.1.1.1. load_neff.m*

```
%pre: lambda in um
%post: nwg_bend, nwg_straight (n_eff) EOS2

function [nwg_bend nwg_straight] = load_neff(lambda)

% import data for bent section
load bend_index_4um_w500nm.txt
% import data for straight section
load neff_straight_12_13.txt
wl = bend_index_4um_w500nm(:,1);
neff_bend = bend_index_4um_w500nm(:,2);
neff_straight = neff_straight_12_13(:,2);

cfun_nwg_bend = fit(wl,neff_bend,'pchipinterp');
cfun_nwg_straight = fit(wl,neff_straight,'pchipinterp');

nwg_bend = cfun_nwg_bend(lambda);
nwg_straight = cfun_nwg_straight(lambda);
```

### *10.1.1.2. bend_index_4um_w500nm.txt*

```
1.15   2.107953904  11.5171  0.741287
1.16   2.095877068  11.3524  0.735497
1.17   2.08385323   11.1908  0.729538
1.18   2.071915973  11.0324  0.723476
1.19   2.060023757  10.8769  0.717311
1.20   2.048228625  10.7245  0.71104
1.21   2.036487605  10.5749  0.70466
1.22   2.024814068  10.4281  0.698168
1.23   2.013240957  10.2842  0.691564
1.24   2.001722914  10.1429  0.684843
1.25   1.990292246  10.0043  0.678004
1.26   1.978957836  9.86838  0.671042
1.27   1.967698117  9.73497  0.663971
1.28   1.956527111  9.60408  0.656728
1.29   1.945444532  9.47565  0.649372
1.30   1.934455918  9.34965  0.641875
```

```
1.31   1.923564531  9.22604  0.634231
1.32   1.91277128   9.10477  0.626436
1.33   1.902080985  8.98581  0.618471
1.34   1.891496147  8.86912  0.610327
1.35   1.881021158  8.75467  0.601999
```

### 10.1.2. first_order_ring_single_depletion_modulator.m

```
% Transmission matrix resonator including index and absorption shift
% due to carrier overlap with waveguide mode.
% Based on work done in conjunction with Eugen Zgraggen

lambda_target = 1.250e-6;
lambda_span = 0.57e-9;
lambda_steps = 50001;
coupling_lambda_target = 0.008;
drop_coupling_lambda_target = 0.002;
coupling_slope = 0.005; % Fractional change in coupling per nm

rwg = 3.99896*10^-6;
wg_loss_dB = 3; % dB/cm
carrier_density_off = 2e17;
carrier_density_on = 2e17;
confinement_factor_off = 0.7;
confinement_factor_on = confinement_factor_off*0.57;

lambda_low = lambda_target-lambda_span/2;
lambda_high = lambda_target+lambda_span/2;
lambda_res = (lambda_high-lambda_low)/(lambda_steps-1);
lambda = lambda_low:lambda_res:lambda_high;

coupling_half_delta = coupling_lambda_target*coupling_slope*1e9*lambda_span/2;
coupling = linspace(coupling_lambda_target-
coupling_half_delta,coupling_lambda_target+coupling_half_delta,lambda_steps);

drop_coupling_half_delta = drop_coupling_lambda_target*coupling_slope*1e9*lambda_span/2;
drop_coupling = linspace(drop_coupling_lambda_target-
drop_coupling_half_delta,drop_coupling_lambda_target+drop_coupling_half_delta,lambda_steps);

kappa1 = sqrt(coupling)*exp(1i*0*pi/180);
kappa2 = sqrt(drop_coupling)*exp(1i*0*pi/180);
t1 = sqrt(1-abs(kappa1).^2)*exp(1i*0*pi/180);
t2 = sqrt(1-abs(kappa2).^2)*exp(1i*0*pi/180);

if(abs(abs(kappa1).^2+abs(t1).^2-(abs(kappa2).^2+abs(t2).^2))>10^-7)
   s = 'Abs. transmission != 1'
```

```
end

% structural parameters
[dnn dan] = soref_poly_na('n',carrier_density_off,lambda_target*1e6);
[dnp dap] = soref_poly_na('p',carrier_density_off,lambda_target*1e6);
dn = (dnn + dnp)*confinement_factor_off;
da = (dan + dap)*confinement_factor_off*10/log(10);
a_loss = log(10)*(wg_loss_dB+da)/10*100/2; %conv. from /cm to /m +field

[nwg nwg_straight] = load_neff(lambda*10^6);
nwg = nwg + dn;
a_round = exp(-a_loss*2*pi*rwg);
Et1 = zeros(1,size(lambda,2));
Et4 = zeros(1,size(lambda,2));

for m = 1:size(lambda,2)
   a = exp(2*pi*nwg(m)/lambda(m)*rwg*2*pi*i)*a_round;
   alpha = abs(a);
   phi = angle(a);
   Et1(m) = (t1(m) - conj(t2(m))*alpha*exp(-1i*phi))/(1-conj(t1(m))*conj(t2(m))*alpha*exp(1i*phi));
   Et4(m) = -conj(kappa1(m))*kappa2(m)*sqrt(alpha)*exp(1i*phi/2)/(1-conj(t1(m))*conj(t2(m))*alpha*exp(1i*phi));
end

% from now on Etf and Edrop are power
Etf = abs(Et1).^2;
Edrop = abs(Et4).^2;
pp_Etf = pchip(lambda,Etf);
pp_Edrop = pchip(lambda,Edrop);

% find resonances
resonances = [];
for index = 2:length(Etf)-1
   if Etf(index)-Etf(index-1) < 0 && Etf(index+1)-Etf(index) > 0
      resonances = [resonances index];
   end
end
off_resonance = resonances(1);
off_extinction = 10*log10(1/Etf(off_resonance))

% fwhm
lambda_left = lambda_low;
lambda_right = lambda_high;
lambda_mid = lambda(resonances(1));
value_left = ppval(lambda_left,pp_Etf);
value_right = ppval(lambda_right,pp_Etf);
% choose smaller one as half reference
```

```
value_target = 0.5;
fwhm_left = fzero(@(l)ppval(l,pp_Etf)-value_target,[lambda_left lambda_mid],10^-12);
fwhm_right = fzero(@(l)ppval(l,pp_Etf)-value_target,[lambda_mid lambda_right],10^-12);
fwhm = fwhm_right-fwhm_left;

% convert dlambda to dfreq
fwhm_freq = dlambda_to_bw(fwhm,lambda_mid);
off_fwhm_freq_GHz = fwhm_freq/1e9

Etf_dB_off = 10*log10(Etf);
Edrop_dB_off = 10*log10(Edrop);

% and now for on state
[dnn dan] = soref_poly_na('n',carrier_density_on,lambda_target*1e6);
[dnp dap] = soref_poly_na('p',carrier_density_on,lambda_target*1e6);
dn = (dnn + dnp)*confinement_factor_on;
da = (dan + dap)*confinement_factor_on*10/log(10);
a_loss = log(10)*(wg_loss_dB+da)/10*100/2; %conv /cm to /m and /2 field
[nwg nwg_straight] = load_neff(lambda*10^6);
nwg = nwg + dn;
a_round = exp(-a_loss*2*pi*rwg);
Etl = zeros(1,size(lambda,2));
Et4 = zeros(1,size(lambda,2));
for m = 1:size(lambda,2)
   a = exp(2*pi*nwg(m)/lambda(m)*rwg*2*pi*i)*a_round;
   alpha = abs(a);
   phi = angle(a);
   Etl(m) = (tl(m) - conj(t2(m))*alpha*exp(-1i*phi))/(1-conj(tl(m))*conj(t2(m))*alpha*exp(1i*phi));
   Et4(m) = -conj(kappa1(m))*kappa2(m)*sqrt(alpha)*exp(1i*phi/2)/(1-conj(tl(m))*conj(t2(m))*alpha*exp(1i*phi));
end
% from now on Etf and Edrop are power
Etf = abs(Etl).^2;
Edrop = abs(Et4).^2;
pp_Etf = pchip(lambda,Etf);
pp_Edrop = pchip(lambda,Edrop);
% find resonances
resonances = [];
for index = 2:length(Etf)-1
   if Etf(index)-Etf(index-1) < 0 && Etf(index+1)-Etf(index) > 0
      resonances = [resonances index];
   end
end
on_resonance = resonances(1);
on_extinction = 10*log10(1/Etf(on_resonance))
insertion_loss = 10*log10(1/Etf(off_resonance))
extinction_ratio = off_extinction-insertion_loss
```

```
% fwhm
lambda_left  = lambda_low;
lambda_right  = lambda_high;
lambda_mid  = lambda(resonances(1));
value_left  = ppval(lambda_left,pp_Etf);
value_right  = ppval(lambda_right,pp_Etf);
% choose  smaller  one as half  reference
value_target  = 0.5;
fwhm_left  = fzero(@(l)ppval(l,pp_Etf)-value_target,[lambda_left  lambda_mid],10^-12);
fwhm_right  = fzero(@(l)ppval(l,pp_Etf)-value_target,[lambda_mid  lambda_right],10^-12);
fwhm  = fwhm_right-fwhm_left;
% convert  dlambda  to dfreq
fwhm_freq  = dlambda_to_bw(whm,lambda_mid);
on_fwhm_freq_GHz  = fwhm_freq/1e9

Etf_dB_on  = 10*log10(Etf);
Edrop_dB_on  = 10*log10(Edrop);
figure(1)
hold  on
plot(lambda*1e9,Etf_dB_off,'b',lambda*1e9,Etf_dB_on,'r')
xlim([lambda_low  lambda_high]*1e9)
xlabel('Wavelength  (nm)')
ylabel('Transmission  (dB)')
ylim([-20  0])
```

### *10.1.2.1. bw_to_dlambda.m*

```
function  FSR_lambda  = bw_to_dlambda(FSR,f_0)

% constants
c = 299792458;
lambda_0  = c/f_0;
f_1 = f_0 + FSR;
lambda_1  = c/f_1;
FSR_lambda  = abs(lambda_1-lambda_0);
```

### *10.1.2.2. dlambda_to_bw.m*

```
function  FSR_freq  = dlambda_to_bw(FSR,lambda_0)

% constants
c = 299792458;
f_0  = c/lambda_0;
lambda_1  = lambda_0  + FSR;
f_1 = c/lambda_1;
FSR_freq  = abs(f_1-f_0);
```

#### *10.1.2.3. soref_poly_na.m*

```
% Soref refractive index change versus carrier injection
% Fit vs. carrier injection and wavelength
% Originally written by Eugen Zgraggen
% pre: type either 'n','p'; Nd doping concentraion in cm^-3; lambda /um
% post: dn, da fits from Soref, Electrooptical Effects in Silicon

function [dn,da] = soref_poly_na(dope_type,Nd,lambda)

if dope_type == 'n'
   dn = -1.1*10^-4*(Nd/(2*10^17)).^1.045*(lambda/1.3).^2;
   da = 4*10^-1*(Nd/10^17).^1.176*(lambda/1.3).^2;
elseif dope_type == 'p'
   dn = -2.3*10^-4*(Nd/10^17).^0.8115*(lambda/1.3).^2;
   da = 3*10^-1*(Nd/10^17).^1.1*(lambda/1.3).^2;
else
   dn = 0;
   da = 0;
end
```

### 10.1.3. calc_vertcoupl_fiber_coupling_omnisim_generated_fields.m

```
wavelength = 1.55e-6;
grating_period = 1.1;
duty_cycle = 0.4;

filename = sprintf('w%dnm_p%dnm_d%dper_FiberPlaneSensor_Ey_ReIm.txt', round(wavelength*1e9),
round(grating_period*1000), round(duty_cycle*100));
new_data = importdata(filename,' ',4);
Ey = new_data.data;
filename = sprintf('w%dnm_p%dnm_d%dper_FiberPlaneSensor_Hz_ReIm.txt', round(wavelength*1e9),
round(grating_period*1000), round(duty_cycle*100));
new_data = importdata(filename,' ',4);
Hz = new_data.data;
filename = sprintf('w%dnm_p%dnm_d%dper_FiberPlaneSensor_Hx_ReIm.txt', round(wavelength*1e9),
round(grating_period*1000), round(duty_cycle*100));
new_data = importdata(filename,' ',4);
Hx = new_data.data;
Hz = flipud(Hz);
Hx = flipud(Hx);
E_coupler_real_raw = Ey(:,2);
E_coupler_imag_raw = Ey(:,3);
z_raw = Ey(:,1)*1e-6;
E_coupler_real_fit = fit(z_raw, E_coupler_real_raw, 'pchipinterp');
E_coupler_imag_fit = fit(z_raw, E_coupler_imag_raw, 'pchipinterp');
```

```
z_min = z_raw(1);
z_max = z_raw(end);
z_step_approx = 10e-9;
z_points = round((z_max-z_min)/z_step_approx)+1;
z_step = (z_max-z_min)/(z_points-1);
z = linspace(z_min,z_max,z_points);

E_coupler_real = E_coupler_real_fit(z);
E_coupler_imag = E_coupler_imag_fit(z);
E_coupler = E_coupler_real + 1i*E_coupler_imag;

% Set up fiber angle looping

angle_min = -20;
angle_max = 20;
angle_step = 0.5;
eight_deg_index = (8-angle_min)/angle_step + 1;
fiber_angles = angle_min:angle_step angle_max;
num_angles = size(fiber_angles,2);
angle_index = 1;
max_efficiency_vs_angle = zeros(num_angles,1);

for fiber_angle=(angle_min:angle_step:angle_max)/180*pi

   mfr_1310 = 9.2e-6/2; % +/- 0.2 um
   mfr_1550 = 10.4e-6/2; % +/- 0.4 um
   mfr = mfr_1310+(mfr_1550-mfr_1310)/(1.55e-6-1.31e-6)*(wavelength-1.31e-6);

   fiber_neff_1310 = 1.4677;
   fiber_neff_1550 = 1.4682;
   fiber_neff = fiber_neff_1310+(fiber_neff_1550-fiber_neff_1310)/(1.55e-6-1.31e-6)*(wavelength-1.31e-6);
   fiber_kappa = fiber_neff*2*pi/wavelength;

   z_fiber_min = -10e-6;
   z_fiber_max = 10e-6;
   z_fiber = z_fiber_min:z_step:z_fiber_max;
   E_fiber = exp(-(z_fiber*cos(fiber_angle)).^2/mfr^2-1i*fiber_kappa*z_fiber*sin(fiber_angle));

   % Zero pad E_coupler
   fiber_size = size(E_fiber,2);
   pad_zeros = zeros(round(fiber_size/2),1);
   E_coupler_padded = [pad_zeros;E_coupler;pad_zeros];
   padded_size = size(E_coupler_padded,1);

   % Loop through integrating with different offsets
```

```
    offset_lengths = (2:10:padded_size-fiber_size-2)*z_step;
    num_offset_points = size(offset_lengths,2);
    fiber_mode_normalization = sum(abs(E_fiber).^2);
    coupler_mode_normalization = sum(abs(E_coupler).^2);
    norm_coeff = 1/(fiber_mode_normalization*coupler_mode_normalization);
    mode_overlap_fraction = zeros(num_offset_points,1);
    int_index = 1;
    for offset_index=2:10:padded_size-fiber_size-2
        left_pad_zeros = zeros(offset_index,1);
        right_pad_zeros = zeros(padded_size-fiber_size-offset_index,1);
        E_fiber_padded = [left_pad_zeros;E_fiber';right_pad_zeros];
        mode_overlap_fraction(int_index) = norm_coeff*abs(sum(conj(E_coupler_padded).*E_fiber_padded)).^2;
        int_index = int_index+1;
    end

    if (angle_index == eight_deg_index)
        [max_eff, max_eff_index] = max(mode_overlap_fraction);
        offset_step = 10*z_step*1e6;
        offset_points = size(mode_overlap_fraction,1);
        offset_min = -offset_step*(max_eff_index-1);
        offset_max = offset_min+(offset_points-1)*offset_step;
        offset_axis = linspace(offset_min,offset_max,offset_points);
        figure(2)
        plot(offset_axis,mode_overlap_fraction)
        xlim([-5 5])
        xlabel('Fiber Misalignment (um)')
        ylabel('Modal Coupling Fraction')
        title('Alignment Tolerance with 8 Degree Fiber Angle')
    end

    max_efficiency_vs_angle(angle_index) = max(mode_overlap_fraction);
    angle_index = angle_index+1;
end

figure(1)
plot(fiber_angles,max_efficiency_vs_angle)
xlabel('Fiber Angle (Degrees)')
ylabel('Modal Coupling Fraction')
```

### 10.1.4. fit_omnisim_calculated_resonator_coupling_coeff.m

```
run_identifier = '1550_wb660_wr660_gr20';
ring_radius = 8.0
grid = 20e-9;
gaps = [0.1, 0.15, 0.2, 0.3, 0.4, 0.6, 1.5];
num_gaps = max(size(gaps));
```

```
wavelength = 1.55e-6;
frequency_THz = 299792458/wavelength/1e12;
coupling_DFT = zeros(num_gaps,1);
loss_DFT = zeros(num_gaps,1);
coupling_fields = zeros(num_gaps,1);
loss_fields = zeros(num_gaps,1);
xmodedim = 2.66e-6;
ymodedim = 2.08e-6;
xregridsize = grid/4;
yregridsize = grid/4;
xnodes = round(xmodedim/xregridsize+1);
ynodes = round(ymodedim/yregridsize+1);

for i=1:num_gaps
   gap = gaps(i)
   filename = sprintf('data/DFT_Results_%s_rad%dum_gap%dnm.txt', run_identifier, round(ring_radius), round(gap*1000));
   new_data = importdata(filename,' ',5);
   DFT_data = new_data.data;
   dft_frequency = DFT_data(:,1);
   dft_output = DFT_data(:,2);
   dft_input = DFT_data(:,3);
   dft_ring = DFT_data(:,4);
   input_DFT_fit = fit(dft_wavelengths, dft_input, 'pchipinterp');
   output_DFT_fit = fit(dft_wavelengths, dft_output, 'pchipinterp');
   ring_DFT_fit = fit(dft_wavelengths, dft_ring, 'pchipinterp');
   input_DFT = input_DFT_fit(frequency_THz);
   output_DFT = output_DFT_fit(frequency_THz);
   ring_DFT = ring_DFT_fit(frequency_THz);
   coupling_DFT(i) = ring_DFT/input_DFT;
   loss_DFT(i) = (input_DFT-ring_DFT-output_DFT)/input_DFT;

   filename = sprintf('data/Ring_Fields_%s_rad%dum_gap%dnm.txt', run_identifier, round(ring_radius), round(gap*1000));
   new_data = importdata(filename,' ',4);
   ring_fields = new_data.data;
   x = ring_fields(:,2);
   y = ring_fields(:,1);
   ExRe = ring_fields(:,3);
   ExIm = ring_fields(:,4);
   EyRe = ring_fields(:,5);
   EyIm = ring_fields(:,6);
   HxRe = ring_fields(:,9);
   HxIm = ring_fields(:,10);
   HyRe = ring_fields(:,11);
   HyIm = ring_fields(:,12);

   Sz_avg = 0.5*((ExRe.*HyRe+ExIm.*HyIm)-(EyRe.*HxRe+EyIm.*HxIm));
```

```
    zg = gridfit(x,y,Sz_avg,xnodes,ynodes,'smoothness',0.1);
    ring_flux  = sum(sum(zg))*xregridsize*yregridsize;

    filename  = sprintf('data/Input_Fields_%s_rad%dum_gap%dnm.txt',  run_identifier,  round(ring_radius),  round(gap*1000));
    new_data  = importdata(filename,'  ',4);
    fields  = new_data.data;
    x = fields(:,2);
    y = fields(:,1);
    ExRe  = fields(:,3);
    ExIm  = fields(:,4);
    EyRe  = fields(:,5);
    EyIm  = fields(:,6);
    HxRe  = fields(:,9);
    HxIm  = fields(:,10);
    HyRe  = fields(:,11);
    HyIm  = fields(:,12);

    Sz_avg = 0.5*((ExRe.*HyRe+ExIm.*HyIm)-(EyRe.*HxRe+EyIm.*HxIm));
    zg = gridfit(x,y,Sz_avg,xnodes,ynodes,'smoothness',0.1);
    input_flux  = sum(sum(zg))*xregridsize*yregridsize;

    filename  = sprintf('data/Output_Fields_%s_rad%dum_gap%dnm.txt',  run_identifier,  round(ring_radius),
round(gap*1000));
    new_data  = importdata(filename,'  ',4);
    fields  = new_data.data;
    x = fields(:,2);
    y = fields(:,1);
    ExRe  = fields(:,3);
    ExIm  = fields(:,4);
    EyRe  = fields(:,5);
    EyIm  = fields(:,6);
    HxRe  = fields(:,9);
    HxIm  = fields(:,10);
    HyRe  = fields(:,11);
    HyIm  = fields(:,12);

    Sz_avg = 0.5*((ExRe.*HyRe+ExIm.*HyIm)-(EyRe.*HxRe+EyIm.*HxIm));
    zg = gridfit(x,y,Sz_avg,xnodes,ynodes,'smoothness',0.1);
    output_flux  = sum(sum(zg))*xregridsize*yregridsize;
    coupling_fields(i)  = ring_flux/input_flux;
    loss_fields(i)  = (input_flux-ring_flux-output_flux)./input_flux;
    ring_flux_fields(i)  = ring_flux;
    ring_flux_dft(i)  = ring_DFT;
    ring_ratio(i)  = ring_flux/ring_DFT;
end
```

```
fitgaps = linspace(0.1,0.8,701)';
fitcouplingmodel = fit(gaps(1:end-1)',coupling_fields(1:end-1)-coupling_fields(end),'exp1');
fitcouplings = fitcouplingmodel(fitgaps);
fitlossmodel = fit(gaps',loss_fields-loss_fields(end),'exp2');
fitlosses = fitlossmodel(fitgaps);

h = figure(1);
subplot(2,1,1);
semilogy(gaps(1:end-1),coupling_fields(1:end-1)-coupling_fields(end),'k.',fitgaps,fitcouplings,'r-')
title(strcat('Ring radius ', int2str(ring_radius), 'um'))
xlim([0.1 0.8])
xlabel('Gap (um)')
ylabel('Coupling fraction')
subplot(2,1,2);
plot(gaps(1:end-1),loss_fields(1:end-1)-loss_fields(end), k.',fitgaps,fitlosses,'r-')
title(strcat('Ring radius ', int2str(ring_radius), 'um'))
xlabel('Gap (um)')
ylabel('Radiative fraction')

filename = sprintf('data/Processed_%s_rad%dum.fig', run_identifier, round(ring_radius));
saveas(h, filename);
filename = sprintf('data/Processed_%s_rad%dum.mat', run_identifier, round(ring_radius));
save(filename, 'fitgaps', 'fitcouplings', 'fitlosses');
```

### 10.1.5. paperclip_plot_loss_1280.m

```
% Example paperclip data analysis script in the presence of noisy
% vertical couplers that cause rapid transmission fluctuations

load('PC_W2000_LA_1270-101-1290_2012#1#3--17#19#53');
waves=transmission(:,1);
pc1_pow = transmission(:,2);
pc1_dbm = 10*log10(pc1_pow*1e3);
len1 = 0.0016;

load('PC_W2000_LB_1270-101-1290_2012#1#3--17#22#34');
pc2_pow = transmission(:,2);
pc2_dbm = 10*log10(pc2_pow*1e3);
len2 = 0.1028;

load('PC_W2000_LC_1270-101-1290_2012#1#3--17#25#51');
pc3_pow = transmission(:,2);
pc3_dbm = 10*log10(pc3_pow*1e3);
len3 = 0.2040;

load('PC_W2000_LD_1270-101-1290_2012#1#3--17#28#28');
```

```
pc4_pow = transmission(:,2);
pc4_dbm = 10*log10(pc4_pow*1e3);
len4 = 0.3052;

%% Individual datapoint fitting
clear loss
clear rsquare
clear min_confint_loss_indiv
clear max_confint_loss_indiv

for i=1:length(waves)
   [fitr goodr] = fit([len1 len2 len3 len4]',[pc1_dbm(i) pc2_dbm(i) pc3_dbm(i) pc4_dbm(i)]','poly1');
   loss(i) = fitr.p1;
   rsquare(i) = goodr.rsquare;
   fitbounds = confint(fitr);
   min_confint_loss_indiv(i) = fitbounds(2,1);
   max_confint_loss_indiv(i) = fitbounds(1,1);
end

plot(waves,loss)
loss_indiv_avg = mean(loss)
loss_indiv_dev = std(loss)
min_confint_loss_indiv_avg = mean(min_confint_loss_indiv)
max_confint_loss_indiv_avg = mean(max_confint_loss_indiv)
rsquare_indiv_avg = mean(rsquare)

%% Averaged transmission fitting
[fitr goodr] = fit([len1 len2 len3 len4]',[mean(pc1_dbm) mean(pc2_dbm) mean(pc3_dbm) mean(pc4_dbm)]','poly1');
loss_global_avg = fitr.p1
fitbounds = confint(fitr);
max_confint_loss_global_avg = fitbounds(1,1)
min_confint_loss_global_avg = fitbounds(2,1)
rsquare_global_avg = goodr.rsquare
```

### 10.1.6. tmatrix_complex_anyangle.m

```
% The code calculates transmission through an arbitrary stackup
% Total transmission of a film in the presence of absorption can be
% calculated. This program has been used to analyze polysilicon film
% transmission measurements and wrapped into a Monte-Carlo loop to
% analyze optimal vertical coupler angles. Based on a VCSEL analysis
% script written by Peter Mayer. Technilogically relevant film
% thicknesses have been removed for non-disclosure.

lambda_start= 1260e-9;
lambda_stop = 1630e-9;
```

```
num_lambdas  = 371;
lambda_ref  = (lambda_start+lambda_stop)/2;
lambdas  = linspace(lambda_start,lambda_stop,num_lambdas);

% Polarization  and Angle
polarization  = 'P';
incident_angle_deg  = 00.0;
incident_angle  = incident_angle_deg*pi/180;

% relevant  indices
nair=1*ones(num_lambdas,1);
len_air=1/2*lambda_ref;
nSi02=1.472-0.02556*(lambdas'*1e6)+0.005342*(lambdas'*1e6).^2;
len_Si02  = 200e-9;  % lengths  are  in  units  of  m
nTeos  = 1.46-0.03096*(lambdas'*1e6)+0.00825*(lambdas'*1e6).^2;
len_teos  = 20e-9;
nUSG  = 1.488-0.02998*(lambdas'*1e6)+0.00676*(lambdas'*1e6).^2;
len_USG  = 100e-9;
nPSG  = 1.488-0.02998*(lambdas'*1e6)+0.00676*(lambdas'*1e6).^2;
len_PSG  = 1.00e-6;
nBPSG  = 1.488-0.02998*(lambdas'*1e6)+0.00676*(lambdas'*1e6).^2;
len_BPSG  = 100e-9;
nSiC  = 2.573*ones(num_lambdas,1);
len_SiC  = 100e-9;
nSiON  = 1.8*ones(num_lambdas,1);
len_SiON  = 0.5e-6;
nSiN  = 1.972*ones(num_lambdas,1);
len_SiN  = 100e-9;
nPolyamide  = 1.53*ones(num_lambdas,1);
len_Polyamide  = 1.0e-6;
apSi  = 100.0;  % inverse  cm
npSi=4 227-1.247*(lambdas'*1e6)+0.6795*(lambdas'*1e6)-0.1305*(lambdas'*1e6)-1i*(lambdas'*1e2)/4/pi*apSi;
len_pSi=100e-9;

% specify  basic  layer  structure  w/ lengths  and complex  indices
struct  = [len_air len_SiN len_Si02  len_pSi len_teos len_Si02  len_USG len_PSG len_BPSG len_SiC len_SiON len_Polyamide
len_air];
indices  = [nair nSiN nSi02 npSi nTeos nSi02 nUSG nPSG nBPSG nSiC SiON nPolyamide nair];

poly_start_z  = len_air+len_Si02_bot+len_SiN_sub+len_Si02_subtherm;
poly_end_z  = poly_start_z+len_pSi;

%***** beginning  of lambda  sweeping  loop  *****
for  i2=1:num_lambdas

   lambda=lambdas(i2);
```

```
    % calculate T matrix from arbitrary layer structure
    T_tot=[1, 0 ; 0, 1];  % initial T-matrix
    for ind=1:(length(struct)-1)
      na=indices(i2,ind);
      nb=indices(i2,ind+1);
      angle_a = asin(nair(i2)/na*sin(incident_angle));
      angle_b = asin(nair(i2)/nb*sin(incident_angle));
      L=struct(ind);
      if (polarization == 'S')
        r=(nb*cos(angle_b)-na*cos(angle_a))/(nb*cos(angle_b)+na*cos(angle_a));
      else
        r=(nb*cos(angle_a)-na*cos(angle_b))/(nb*cos(angle_a)+na*cos(angle_b));
      end
      t=sqrt(1-r^2);
      beta=2*pi*na/lambda;
    % this is the Tmat for propagating a length L of a material w/ index na
      T_bulk=[exp(1i*beta*L*cos(angle_a)),0;0,exp(-1i*beta*L*cos(angle_a))];
    % and this Tmat associated with the interface right after the propagation
      T_int=[1/t,-r/t;-r/t,1/t];
    % Tmatrices can be cascaded like this...
      T_tot=T_tot*T_bulk*T_int;
    end
    r_struct(i2)=-T_tot(1,2)/T_tot(1,1);  % this is the reflection coeff for the structure
    t_struct(i2)=1/T_tot(1,1);
end
%***** end of lambda sweeping loop *****

figure(1);
plot(lambdas,abs(t_struct).^2,'g');
title('sweep of transmission vs wavelength');
[trans_max_sweptlambda,ind_max_sweptlambda]=max(abs(t_struct));

figure(2);
plot(lambdas,abs(r_struct).^2,'g');
title('sweep of reflection vs wavelength');
[trans_max_sweptlambda,ind_max_sweptlambda]=max(abs(r_struct));

reflectivity_vec=abs(r_struct).^2;
transmissivity_vec=abs(t_struct).^2;
absorption_vec=1-reflectivity_vec-transmissivity_vec;
user_output=[1e6.*lambdas',reflectivity_vec',transmissivity_vec',absorption_vec']

num_lambdas = round((num_lambdas-1)/10)+1;
lambdas = linspace(lambda_start,lambda_stop,num_lambdas);
energy_ratio = ones(num_lambdas,1);
```

```
% ***** start of field plotting loop *****
for lambda_index=1:num_lambdas

  lambda = lambdas(lambda_index);
  acc=0;
  for i=1:length(struct)
    acc=acc+struct(i);
    interface_positions(i)=acc;
  end
  tot_length=acc;
  num=5000;
  delta=tot_length/num;  % spatial resolution of E-field calculation

  poly_start_index = round(poly_start_z/delta)+1;
  poly_end_index = round(poly_end_z/delta)+1;

  counter=1;
  n_prev=indices(lambda_index,1);
  T_tot=[1,0;0,1];
  for index=1:num
    pointer_pos=index*delta;
    if ((pointer_pos>interface_positions(counter))&(index<num))
      n_pres=indices(lambda_index,counter+1);
      length2=pointer_pos-interface_positions(counter);
      length1=delta-length2;
      n_indexed(index)=n_prev;
      nb=n_pres;
      na=n_prev;
      angle_a = asin(nair(lambda_index)/na*sin(incident_angle));
      angle_b = asin(nair(lambda_index)/nb*sin(incident_angle));
      if (polarization == 'S')
        r=(nb*cos(angle_b)-na*cos(angle_a))/(nb*cos(angle_b)+na*cos(angle_a));
      else
        r=(nb*cos(angle_a)-na*cos(angle_b))/(nb*cos(angle_a)+na*cos(angle_b));
      end
      t=sqrt(1-r^2);
      beta1=2*pi*n_prev/lambda;
      beta2=2*pi*n_pres/lambda;
      T_bulk1=[exp(1i*beta1*length1*cos(angle_a)),0;0,exp(-1i*beta1*length1*cos(angle_a))];
      T_int=[1/t,-r/t;-r/t,1/t];
      T_bulk2=[exp(1i*beta2*length2*cos(angle_b)),0;0,exp(-1i*beta2*length2*cos(angle_b))];
      T=T_bulk1*T_int*T_bulk2;
      counter=counter+1;
      n_prev=n_pres;
    else
      n_pres=indices(lambda_index,counter);
```

```
        n_indexed(index)=n_pres;
        nb=n_pres;
        angle_b  = asin(nair/nb*sin(incident_angle));
        beta=2*pi*n_pres/lambda;
        T=[exp(1i*beta*delta*cos(angle_b)),0;0,exp(-1i*beta*delta*cos(angle_b))];
        n_prev=n_pres;
      end
      T_indexed(:,:,index)=T;
      T_tot=T_tot*T;
      z_coord(index)=pointer_pos-delta;
    end

    z_coord(num+1)=num*delta;
    n_indexed(num+1)=n_indexed(num);
    ref21=-T_tot(1,2)/T_tot(1,1);

    field(:,num+1)=[ref21;  1];
    for  index=num:-1:1
      field(:,index)=T_indexed(:,:,index)*field(:,index+1);
    end
    for  index=(num+1):-1:1
      field(:,index)=field(:,index)/sqrt(n_indexed(index));
    end
    field_tot=field(1,:)+field(2,:);

    field_scale=max(real([field(1,:),field(2,:)]))/2;
    figure(3);
    hold  on;
    plot(z_coord*1e6,real(field(2,:))/field_scale,'g');
    plot(z_coord*1e6,real(field(1,:))/field_scale,'r');
    plot(z_coord*1e6,real(n_indexed),'b');
    title('left  and right  travelling  E-field');
    xlabel('distance  (microns)');
    ylabel('E-field  (a.u.)');

    energy_scale=max(abs(field_tot).^2)/2;
    figure(4);
    hold  on;
    plot(z_coord*1e6,real(n_indexed),'b');
    plot(z_coord*1e6,(abs(field_tot).^2)/energy_scale,'r');
    title('total  energy  in  E-field');
    xlabel('distance  (microns)');
    ylabel('energy  (a.u.)');

    energy_ratio(lambda_index)  = sum(abs(field_tot(poly_start_index:poly_end_index)).^2)/(poly_end_index-
poly_start_index+1)/abs(field_tot(end).^2);
```

```
end

average_energy_ratio = sum(energy_ratio)/num_lambdas

figure(5);
hold on
plot(lambdas,energy_ratio,'b',lambdas,average_energy_ratio*ones(size(lambdas)),'r')
hold off
```

### 10.1.7. tmatrix_complex_anyangle_MonteCarlo.m

```
% See previous program for further overview. This is the MonteCarlo version of the tmatrix solver

% wavelength of interest
lambda_start= min(lambdas);
lambda_stop = max(lambdas);
num_lambdas = max(size(lambdas));
lambda_ref = (lambda_start+lambda_stop)/2;

% Polarization and Angle
polarization = 'P';
incident_angle_deg = 00.0;
incident_angle = incident_angle_deg*pi/180;

%%%
%%% monte carlo variables
%%%
num_trials = 1000;
num_stored_solutions = 5;
goodness_list = zeros(num_stored_solutions,2);
RandStream.setDefaultStream(RandStream('mt19937ar','seed',sum(100*clock)));

nair=1*ones(num_lambdas,1);
nair_var = 0;
nair_dist = 'F';
len_air=1/2*lambda_ref;  % just some air to plot fields in
len_air_var = 0;
len_air_dist = 'F';
nSiO2=1.472-0.02556*(lambdas'*1e6)+0.005342*(lambdas'*1e6).^2;
nSiO2_var = 0.01;
nSiO2_dist = 'G';
len_SiO2_bot = 500e-9; % lengths are in units of m
len_SiO2_bot_var = 50e-9;
len_SiO2_bot_dist = 'U';
len_SiO2_top = 250e-9; % lengths are in units of m
```

```
len_SiO2_top_var = 25e-9;
len_SiO2_top_dist = 'U';
apSi = 0.0; % inverse cm
npSi=4.227-1.247*(lambdas'*1e6)+0.6795*(lambdas'*1e6)-0.1305*(lambdas'*1e6)-1i*(lambdas'*1e2)/4/pi*apSi;
npSi_var = mean(npSi)*0.05;
npSi_dist = 'G';
len_pSi=100e-9;
len_pSi_var = 2e-9;
len_pSi_dist = 'G';

% specify basic layer structure w/ lengths and complex indices
%          Layer 1    Layer 2       Layer 3    Layer 4       Layer 5
stack_mean = [len_air    len_SiO2_bot    len_pSi    len_SiO2_top    len_air];
stack_vars = [len_air_var  len_SiO2_bot_var  len_pSi_var  len_SiO2_top_var  len_air_var];
stack_dists = [len_air_dist len_SiO2_bot_dist len_pSi_dist len_SiO2_top_dist len_air_dist];
index_mean = [nair       nSiO2          npSi       nSiO2          nair];
index_vars = [nair_var    nSiO2_var      npSi_var    nSiO2_var      nair_var];
index_dists = [nair_dist    nSiO2_dist      npSi_dist    nSiO2_dist      nair_dist];

num_layers = max(size(stack_mean));

stack_peturbation = zeros(1,num_layers);
index_peturbation = zeros(1,num_layers);

for trial_index = 1:num_trials

  %***** create a structure instantiation *****

  for i=1:num_layers
    if (index_dists(i) == 'G')
      index_peturbation(i) = index_vars(i)*randn();
    end
    if (index_dists(i) == 'U')
      index_peturbation(i) = 2*index_vars(i)*rand()-index_vars(i);
    end
    if (index_dists(i) == 'T')
      index_peturbation(i) = index_peturbation(index_vars(i));
    end
    if (stack_dists(i) == 'G')
      stack_peturbation(i) = stack_vars(i)*randn();
    end
    if (stack_dists(i) == 'U')
      stack_peturbation(i) = 2*stack_vars(i)*rand()-stack_vars(i);
    end
  end
```

```
indices = index_mean + ones(num_lambdas,1)*index_peturbation;
stackup = stack_mean + stack_peturbation;

%***** beginning of lambda sweeping loop *****
for i2=1:num_lambdas

   lambda=lambdas(i2);

   % calculate T matrix from arbitrary layer structure
   T_tot=[1, 0 ; 0, 1];  % initial T-matrix
   for ind=1:(length(stackup)-1)
     na=indices(i2,ind);
     nb=indices(i2,ind+1);
     angle_a = asin(nair(i2)/na*sin(incident_angle));
     angle_b = asin(nair(i2)/nb*sin(incident_angle));
     L=stackup(ind);
     if (polarization == 'S')
        r=(nb*cos(angle_b)-na*cos(angle_a))/(nb*cos(angle_b)+na*cos(angle_a));
     else
        r=(nb*cos(angle_a)-na*cos(angle_b))/(nb*cos(angle_a)+na*cos(angle_b));
     end
     t=sqrt(1-r^2);
     beta=2*pi*na/lambda;
   % this is the Tmat for propagating a length L of a material w/ index na
     T_bulk=[exp(1i*beta*L*cos(angle_a)),0;0,exp(-1i*beta*L*cos(angle_a))];
   % and this Tmat associated with the interface right after the propagation
     T_int=[1/t,-r/t;-r/t,1/t];
   % Tmatrices can be cascaded like this...
     T_tot=T_tot*T_bulk*T_int;
   end
   r_struct(i2)=-T_tot(1,2)/T_tot(1,1);  % this is the reflection coeff for the structure
   t_struct(i2)=1/T_tot(1,1);
end

%***** end of lambda sweeping loop *****

transmissions = abs(t_struct).^2;
reflections = abs(r_struct).^2;

rel_errors = abs(transmissions-meas_transmission')./meas_transmission';
weighted_error = sum(scaled_error(rel_errors));
%weighted_error = sum(rel_errors.^2);

if (trial_index == 1)
   stored_solutions(1)=struct('weighted_error', weighted_error,...
               'reflections', reflections,        ...
```

```
                'transmissions', transmissions,     ...
                'stack_peturbation', stack_peturbation, ...
                'index_peturbation', index_peturbation, ...
                'stackup', stackup,                 ...
                'indices', indices );
    goodness_list(1,1) = 1;
    goodness_list(1,2) = weighted_error;
  else
    if (trial_index <= num_stored_solutions)
      stored_solutions(trial_index) = struct('weighted_error', weighted_error,   ...
                'reflections', reflections,         ...
                'transmissions', transmissions,     ...
                'stack_peturbation', stack_peturbation, ...
                'index_peturbation', index_peturbation, ...
                'stackup', stackup,                 ...
                'indices', indices );
      insertion_index = find(goodness_list(:,2) > weighted_error,1);
      if insertion_index > 0
        goodness_list(insertion_index+1:end,:) = goodness_list(insertion_index:end-1,:);
      else
        insertion_index = trial_index;
      end
      goodness_list(insertion_index,1) = trial_index;
      goodness_list(insertion_index,2) = weighted_error;
    else
      insertion_index = find(goodness_list(:,2) > weighted_error,1);
      if insertion_index > 0
        replacement_index = goodness_list(num_stored_solutions,1);
        stored_solutions(replacement_index) = struct(   'weighted_error', weighted_error,   ...
                'reflections', reflections,         ...
                'transmissions', transmissions,     ...
                'stack_peturbation', stack_peturbation, ...
                'index_peturbation', index_peturbation, ...
                'stackup', stackup,                 ...
                'indices', indices );
        goodness_list(insertion_index+1:end,:) = goodness_list(insertion_index:end-1,:);
        goodness_list(insertion_index,1) = replacement_index;
        goodness_list(insertion_index,2) = weighted_error;
      end
    end
  end
end

figure(1);
plot(lambdas*1e6,meas_transmission,'k');
title('sweep of transmission vs wavelength');
```

```
xlabel('Wavelength (um)')
ylabel('Normalized_transmission')

for i = 1:num_stored_solutions
   norm_transmission = stored_solutions(i).transmissions;
   hold on
   plot(lambdas*1e6,norm_transmission,'r');
   hold off
end
```

## 10.2. Sentaurus TCAD Scripts

### 10.2.1. Modulator (Generic) Junction Implant Process Simulation

```
set sim_left 0
set sim_right 3.0
set sim_bottom 0.1
set sim_top 0
pdbSetBoolean Grid MGoals UseLines 1
line x loc=0.0 tag=top
line x loc=0.1 tag=bottom
line y loc=0.0 tag=left
line y loc=3.0 tag=right
region Oxide xlo=top xhi=bottom ylo=left yhi=right
init slice.angle=[CutLine2D 0 0 3.0 0.0]
math coord.ucs
AdvancedCalibration

pdbSet Diffuse minT 500.0
pdbSet Grid SnMesh MaxPoints 2000000
pdbSet Grid SnMesh CoplanarityAngle 179.0
pdbSet Oxide_PolySilicon Boron BoundaryCondition Segregation
pdbSet Oxide_PolySilicon Arsenic BoundaryCondition Segregation
pdbSet Oxide_PolySilicon Phosphorus BoundaryCondition Segregation
pdbSet PolySilicon PolyCrystalline 1
pdbSet PolySilicon Dopant DiffModel Granular
pdbSet PolySilicon GrainShape Columnar
pdbSet PolySilicon GrainSize 7e-6
pdbSet PolySilicon GrainBoundaryThickness 3e-8
pdbSet PolySilicon LayerThickness 2e-5
pdbSet PolySilicon Dopant GBMaxDensity 1e15
pdbSet PolySilicon Dopant ActiveModel None

mgoals min.normal.size= 3<nm> max.lateral.size= 100<nm> normal.growth.ratio= 1.01
refinebox interface.materials= Oxide
sde off
```

```
update_substrate top relaxed.coord=0.0

deposit material = {PolySilicon} type= anisotropic thickness= 0.2
deposit material = {Oxide} type = isotropic thickness= 0.10

# Hard Mask
mask name=mask_1_2 segments = {0.5 2.5} negative
photo mask=mask_1_2 thickness=0.1
etch material = {Oxide} type=anisotropic rate = {0.5} time=1.1
strip Resist

#P Mask
mask name=mask_1_3 segments = {-0.5 1.2} negative
photo mask=mask_1_3 thickness=1.0
implant Boron dose=1e15 energy=3.5 tilt=0 rot=0 info=1
implant Boron dose=3e13 energy=37 tilt=0 rot=0 info=1
strip Resist
#N Mask
mask name=mask_1_4 segments = {1.8 3.5} negative
photo mask=mask_1_4 thickness=2.0
implant Arsenic dose=1e15 energy=50 tilt=0 rot=0 info=1
implant Phosphorus dose=6.5e13 energy=240 tilt=0 rot=0 info=1
strip Resist

struct tdr=vert_polySi_before_anneal

# Fake Spike Conditions
temp_ramp name= dt_sd time= 10<s> temperature= 400<C> t.final=600<C>
temp_ramp name= dt_sd time= 3<s> temperature= 600<C> t.final=1050<C>
temp_ramp name= dt_sd time= 10<s> temperature= 1050<C> t.final=400<C>
diffuse temp_ramp= dt_sd info=1

struct tdr=vert_polySi_final

contact box xlo=-0.199 ylo=0.0 xhi=-0.201 yhi=0.5 name=nside PolySilicon
contact box xlo=-0.199 ylo=2.5 xhi=-0.201 yhi=3.0 name=pside PolySilicon

struct smesh=vert_polySi_forsdevice
exit
```

### 10.2.2. Carrier Profile as a Function of Bias Device Simulation

```
Electrode {
  { Name="pside" voltage=0.0 }
  { Name="nside" voltage=0.0 }
}
```

```
File {
   Grid = "vert_polySi_forsdevice_fps.tdr"
   Parameter = "simple_sdevice_params.par"
   Plot = "vert_junction"
   Output = "vert_junction_reverse_device.log"
   Current = "vert_junction_reverse_des.plt"
}

Physics {
   Fermi
}

Math {
   Digits=5
   ErrReff(electron) = 1.0e6
   ErrReff(hole) = 1.0e6
   RelErrControl
}

Solve {
   Poisson
   Coupled(iterations=1000) { Poisson Electron Hole }

   Quasistationary ( InitialStep=1e-2 MaxStep=0.01 MinStep=1e-9
            Increment=1.5 Decrement=2.0
            Plot {Range=(0 1) Intervals=6}
            Goal { Voltage=3.0 Name=nside } ) {
      Coupled(iterations=15) { Poisson Electron Hole }
   }
}
```

## 10.3. Sentaurus Raphael Parasitic Calculation Scripts

### 10.3.1. Electromagnetic Calculation (RC2)

```
* RC2     RUN OUTPUT=depletion_modulator
$ Example Depletion Mode Device Structure
BOX NAME=air; CX=0; CY=0; W=6; H=8; DIEL=1.0; COLOR=1;
BOX NAME=dielect; CX=0; CY=0.955; W=6; H=2 2; DIEL=3.9; COLOR=3;
BOX NAME=nitride; CX=0; CY=0.110; W=6; H=0.020; DIEL=7.5; COLOR=5;
BOX NAME=polysi; CX=0; CY=0.050; W=0.180; H=0.100; DIEL=11.6; COLOR=4;
$ Metal line 1, with rounded corners
POLY      NAME=m1; COORD=-1.5,0; -0.090,0; -0.090,0.080; -1.3,0.080; -1.3,0.580; -1.5,0.580; VOLT=-1; COLOR=2;
POLY      NAME=m2; COORD=1.5,0; 0.090,0; 0.090,0.080; 1.3,0.080; 1.3,0.580; 1.5,0.580; VOLT=1; COLOR=2;
```

```
BOX     NAME=ground;  CX=0; CY=2.1; W=6; H=0.2; VOLT=0;  COLOR=2;

WINDOW  X1=-3.0; Y1=-4.0; X2=3.0; Y2=2.2; DIEL=1;

OPTIONS  SET_GRID=100000;  MAX_REGRID=0;

POTENTIAL

CAPACITANCE  m1; m2;
```

### 10.3.2. Field Plotting (dplot)

```
$ Read in the geometry and the potential data
DATA RAPHAEL FILE=depletion_modulator.pot
$ Plot the frame
PLOT.2D TITLE="Modulator: Structure & Grid" Y.LENGTH=8
$ Plot the structure and grid
$STRUCTURE BOUND GRID
STRUCTURE BOUND
$ Plot a new frame
PLOT.2D TITLE="Modulator: Potential Plot" Y.LENGTH=8
$ Plot the potential contours
CONTOUR POTENTIAL FILL
$ add the structure and grid
$STRUCTURE GRID
```

## 10.4. Cadence Virtuoso P-Cell Photonic Layout Code

### 10.4.1. Global Functions

#### *10.4.1.1. GetGlobal*

```
procedure( GetGlobal( var_name )
  let( (var_value)
  case( var_name
    ( ("mask_grid")
      var_value = 0.005
    )
    ( ("min_width")
      var_value = 0.250
    )
    ( ("min_space")
      var_value = 0.250
    )
    ( ("dopeblk_grid")
      var_value = 0.5
```

```
      )
      ( ("fillblk_grid")
         var_value  = 1.0
      )
      ( ("lcm_grid")
         var_value  = 1.0
      )
      ( ("dopeblk_overhang")
         var_value  = 2.0
      )
      ( ("fillblk_overhang")
         var_value  = 4.0
      )
   )
   var_value
   ) ;end let statement
) ;end procedure  GetGlobal
```

### *10.4.1.2. GetSingleModeWidth*

```
procedure(  GetSingleModeWidth( thickness wavelength )
   let( (single_mode_width)
      case( wavelength
         ( ("1200" "1220" "1250" "1280" "1300"  "1310")
            case( thickness
               ( "100"
                  single_mode_width  = 0.400
               )
               ( "200"
                  single_mode_width  = 0.280
               )
            )
         )
         ( ("1550")
            case( thickness
               ( "100"
                  single_mode_width  = 0.500
               )
               ( "200"
                  single_mode_width  = 0.350
               )
            )
         )
      )
      single_mode_width
   ) ; end let statement
```

```
) ; end procedure GetSingleModeWidth
```

### *10.4.1.3. BlockWriter*

```
procedure( BlockWriter( coords layer)
  let( (initial_coords final_coords x_min x_max y_min y_max x_min_mod x_max_mod y_min_mod y_max_mod min_delta
max_coord )

  initial_coords = nth(0 coords)
  final_coords = nth(1 coords)
  x_min = RoundToMaskGrid(nth(0 initial_coords))
  y_min = RoundToMaskGrid(nth(1 initial_coords))
  x_max = RoundToMaskGrid(nth(0 final_coords))
  y_max = RoundToMaskGrid(nth(1 final_coords))
  min_delta = min( abs(x_max-x_min) abs(y_max-y_min) )
  max_coord = max( abs(x_max) abs(x_min) abs(y_max) abs(y_min) )

  if( ((min_delta > 0) && (max_coord < 300000)) then
    case( layer
      ( ("FILLBLOCK")
        ; Real example would have all layers to block autofill here
        dbCreateRect( pcCellView list( "blah" "exclude" ) list( x_min:y_min x_max:y_max ) )
      )
      ( ("DOPEBLOCK")
        ; Real example would have all layers to block doping and silicidation process steps here
        dbCreateRect( pcCellView list( "blah" "drawing" ) list( x_min:y_min x_max:y_max ) )
      )
      ( ("POLY")
        ; Real example would have gate polysilicon layer reference here
        dbCreateRect( pcCellView list( "blah" "drawing" ) list( x_min:y_min x_max:y_max ) )
      )
      ( ("BODY")
        ; Real example would have single crystalline transistor body silicon layer reference here
        dbCreateRect( pcCellView list( "blah" "drawing" ) list( x_min:y_min x_max:y_max ) )
      )
      ( ("METAL")
        ; Real example would have first copper metal layer reference here
        dbCreateRect( pcCellView list( "blah" "drawing" ) list( x_min:y_min x_max:y_max ) )
      )
    )
  )
  ) ;end let statement
) ;end procedure BlockWriter
```

### *10.4.1.4. CalcOffsetX*

```
procedure( CalcOffsetX( pc_x_origin pc_y_origin pc_rotation design_grid mask_grid )
```

```
  let( ( grid_offset_x )
  case( pc_rotation
    ( ( "R0" "MX" )
      grid_offset_x = jsoRound((pc_x_origin-jsoFloor(pc_x_origin/design_grid)*design_grid)/mask_grid)*mask_grid
    )
    ( ( "R90" "MXR90" )
      grid_offset_x = jsoRound((pc_y_origin-jsoFloor(pc_y_origin/design_grid)*design_grid)/mask_grid)*mask_grid
    )
    ( ( "R180" "MY" )
      grid_offset_x = jsoRound((jsoFloor(pc_x_origin/design_grid)*design_grid-pc_x_origin)/mask_grid)*mask_grid
    )
    ( ( "R270" "MYR90" )
      grid_offset_x = jsoRound((jsoFloor(pc_y_origin/design_grid)*design_grid-pc_y_origin)/mask_grid)*mask_grid
    )
    ( t
      print( "Error in CalcOffsetX" )
    )
  )
  grid_offset_x
  ) ;end let statement
) ;end procedure
```

#### 10.4.1.5. *CalcOffsetY*

```
procedure( CalcOffsetY( pc_x_origin pc_y_origin pc_rotation design_grid mask_grid )
  let( (grid_offset_y)
  case( pc_rotation
    ( ( "R0" "MY" )
      grid_offset_y = jsoRound((pc_y_origin-jsoFloor(pc_y_origin/design_grid)*design_grid)/mask_grid)*mask_grid
    )
    ( ( "R90" "MYR90" )
      grid_offset_y = jsoRound((jsoFloor(pc_x_origin/design_grid)*design_grid-pc_x_origin)/mask_grid)*mask_grid
    )
    ( ( "R180" "MX" )
      grid_offset_y = jsoRound((jsoFloor(pc_y_origin/design_grid)*design_grid-pc_y_origin)/mask_grid)*mask_grid
    )
    ( ( "R270" "MXR90" )
      grid_offset_y = jsoRound((pc_x_origin-jsoFloor(pc_x_origin/design_grid)*design_grid)/mask_grid)*mask_grid
    )
    ( t
      print( "Error in CalcOffsetY" )
    )
  )
  grid_offset_y
  ) ;end let statement
) ;end procedure
```

#### 10.4.1.6. *CalcSubOriginX*

```
procedure( CalcSubOriginX( cell_pc_rotation cell_pc_x_origin x0 y0 )
  let( (sub_pc_x_origin)
  case( cell_pc_rotation
    ( ( "R0" "MX" )
      sub_pc_x_origin = cell_pc_x_origin + x0
    )
    ( ( "R90" "MYR90" )
      sub_pc_x_origin = cell_pc_x_origin - y0
    )
    ( ( "R180" "MY" )
      sub_pc_x_origin = cell_pc_x_origin - x0
    )
    ( ( "R270" "MXR90" )
      sub_pc_x_origin = cell_pc_x_origin + y0
    )
    ( t
      print( "Error in CalcSubOriginX" )
    )
  )
  sub_pc_x_origin
  ) ; end let
); end procedure
```

#### 10.4.1.7. *CalcSubOriginY*

```
procedure( CalcSubOriginY( cell_pc_rotation cell_pc_y_origin x0 y0 )
  let( (sub_pc_y_origin)
  case( cell_pc_rotation
    ( ( "R0" "MY" )
      sub_pc_y_origin = cell_pc_y_origin + y0
    )
    ( ( "R90" "MXR90" )
      sub_pc_y_origin = cell_pc_y_origin + x0
    )
    ( ( "R180" "MX" )
      sub_pc_y_origin = cell_pc_y_origin - y0
    )
    ( ( "R270" "MYR90" )
      sub_pc_y_origin = cell_pc_y_origin - x0
    )
    ( t
      print( "Error in CalcSubOriginY" )
    )
  )
  sub_pc_y_origin
```

```
  ) ; end let
); end procedure
```

#### 10.4.1.8. *CalcSubRotation*

```
procedure( CalcSubRotation( cell_pc_rotation placed_pc_rotation )
  let( (sub_pc_rotation)
  case( cell_pc_rotation
    ( "R0"
      sub_pc_rotation = placed_pc_rotation
    )
    ( "R90"
      case( placed_pc_rotation
        ( "R0"
          sub_pc_rotation = "R90"
        )
        ( "R90"
          sub_pc_rotation = "R180"
        )
        ( "R180"
          sub_pc_rotation = "R270"
        )
        ( "R270"
          sub_pc_rotation = "R0"
        )
        ( "MY"
          sub_pc_rotation = "MYR90"
        )
        ( "MX"
          sub_pc_rotation = "MXR90"
        )
        ( "MYR90"
          sub_pc_rotation = "MX"
        )
        ( "MXR90"
          sub_pc_rotation = "MY"
        )
      )
    )
    ( "R180"
      case( placed_pc_rotation
        ( "R0"
          sub_pc_rotation = "R180"
        )
        ( "R90"
          sub_pc_rotation = "R270"
```

```
        )
        ( "R180"
          sub_pc_rotation = "R0"
        )
        ( "R270"
          sub_pc_rotation = "R90"
        )
        ( "MY"
          sub_pc_rotation = "MX"
        )
        ( "MX"
          sub_pc_rotation = "MY"
        )
        ( "MYR90"
          sub_pc_rotation = "MXR90"
        )
        ( "MXR90"
          sub_pc_rotation = "MYR90"
        )
      )
    )
    ( "R270"
      case( placed_pc_rotation
        ( "R0"
          sub_pc_rotation = "R270"
        )
        ( "R90"
          sub_pc_rotation = "R0"
        )
        ( "R180"
          sub_pc_rotation = "R90"
        )
        ( "R270"
          sub_pc_rotation = "R180"
        )
        ( "MY"
          sub_pc_rotation = "MXR90"
        )
        ( "MX"
          sub_pc_rotation = "MYR90"
        )
        ( "MYR90"
          sub_pc_rotation = "MY"
        )
        ( "MXR90"
          sub_pc_rotation = "MX"
```

```
      )
    )
  )
  ( "MY"
    case( placed_pc_rotation
      ( "R0"
        sub_pc_rotation = "MY"
      )
      ( "R90"
        sub_pc_rotation = "MXR90"
      )
      ( "R180"
        sub_pc_rotation = "MX"
      )
      ( "R270"
        sub_pc_rotation = "MYR90"
      )
      ( "MY"
        sub_pc_rotation = "R0"
      )
      ( "MX"
        sub_pc_rotation = "R180"
      )
      ( "MYR90"
        sub_pc_rotation = "R270"
      )
      ( "MXR90"
        sub_pc_rotation = "R90"
      )
    )
  )
  ( "MX"
    case( placed_pc_rotation
      ( "R0"
        sub_pc_rotation = "MX"
      )
      ( "R90"
        sub_pc_rotation = "MYR90"
      )
      ( "R180"
        sub_pc_rotation = "MY"
      )
      ( "R270"
        sub_pc_rotation = "MXR90"
      )
      ( "MY"
```

```
            sub_pc_rotation = "R180"
        )
        ( "MX"
            sub_pc_rotation = "R0"
        )
        ( "MYR90"
            sub_pc_rotation = "R90"
        )
        ( "MXR90"
            sub_pc_rotation = "R270"
        )
    )
)
( "MYR90"
    case( placed_pc_rotation
        ( "R0"
            sub_pc_rotation = "MYR90"
        )
        ( "R90"
            sub_pc_rotation = "MY"
        )
        ( "R180"
            sub_pc_rotation = "MXR90"
        )
        ( "R270"
            sub_pc_rotation = "MX"
        )
        ( "MY"
            sub_pc_rotation = "R90"
        )
        ( "MX"
            sub_pc_rotation = "R270"
        )
        ( "MYR90"
            sub_pc_rotation = "R0"
        )
        ( "MXR90"
            sub_pc_rotation = "R180"
        )
    )
)
( "MXR90"
    case( placed_pc_rotation
        ( "R0"
            sub_pc_rotation = "MXR90"
        )
```

```
            ( "R90"
              sub_pc_rotation = "MX"
            )
            ( "R180"
              sub_pc_rotation = "MYR90"
            )
            ( "R270"
              sub_pc_rotation = "MY"
            )
            ( "MY"
              sub_pc_rotation = "R270"
            )
            ( "MX"
              sub_pc_rotation = "R90"
            )
            ( "MYR90"
              sub_pc_rotation = "R180"
            )
            ( "MXR90"
              sub_pc_rotation = "R0"
            )
          )
        )
    ) ; end case
    sub_pc_rotation
    ) ; end let
); end procedure
```

### *10.4.1.9. PolyMaskSize*

```
procedure( PolyMaskSize( physical_width )
    let( (fixed_bias mask_width)
    fixed_bias = 0.000
    mask_width = physical_width + fixed_bias
    mask_width
    ) ;end let statement
) ;end procedure PolyMaskSize
```

### *10.4.1.10. PolyMaskGap*

```
procedure( PolyMaskGap( physical_gap physical_width_1 physical_width_2 )
    let( (fixed_bias mask_gap)
    fixed_bias = 0.000
    mask_gap = physical_gap - fixed_bias
    mask_gap
    ) ;end let statement
) ;end procedure PolyMaskGap
```

### *10.4.1.11. jsoFloor*

```
procedure( jsoFloor(number)
  let( (num_string fix_num )
  if( floatp(number) then
    sprintf( num_string, "%f" number)
    fix_num = atoi(car(parseString(num_string ".")))
  else
    fix_num = number
  )
  fix_num
  ) ;end let statement
) ;end procedure
```

### *10.4.1.12. jsoRound*

```
procedure( jsoRound(number)
  let( (num_string round_num fix_num floating_part first_digit signplace)
  if( floatp(number) then
    sprintf( num_string, "%f" number)
    parsed_num_string = parseString(num_string ".")
    fix_num = atoi(car(parsed_num_string))
    floating_part = cadr(parsed_num_string)
    first_digit = atoi(substring(floating_part 1 1))
    if( first_digit > 4 then
      signplace = substring(num_string 1 1)
      if( (strcmp(signplace "-") == 0) then
        round_num = fix_num - 1
      else
        round_num = fix_num + 1
      )
    else
      round_num = fix_num
    )
  else
    round_num = number
  )
  round_num
  ) ;end let statement
) ;end procedure
```

### *10.4.1.13. jsoCeiling*

```
procedure( jsoCeiling(number)
  let( (num_string ceil_num fix_num floating_part first_digit second_digit third_digit signplace)
  if( floatp(number) then
    sprintf( num_string, "%f" number)
```

```
    parsed_num_string  = parseString(num_string ".")
    fix_num  = atoi(car(parsed_num_string))
    signplace  = substring(num_string 1 1)
    floating_part  = cadr(parsed_num_string)
    if(  strlen(floating_part)  > 0 then
       first_digit  = atoi(substring(floating_part  1 1))
       if(  first_digit  > 0 then
          if(  (strcmp(signplace  "-") == 0) then
             ceil_num  = fix_num  - 1
          else
             ceil_num  = fix_num  + 1
          )
       else
          if(  strlen(floating_part)  > 1 then
             second_digit  = atoi(substring(floating_part  2 1))
             if(  second_digit  > 0 then
                if(  (strcmp(signplace  "-") == 0) then
                   ceil_num  = fix_num  - 1
                else
                   ceil_num  = fix_num  + 1
                )
             else
                if(  strlen(floating_part)  > 2 then
                   third_digit  = atoi(substring(floating_part  3 1))

                   if(  third_digit  > 0 then
                      if(  (strcmp(signplace  "-") == 0) then
                         ceil_num  = fix_num  - 1
                      else
                         ceil_num  = fix_num  + 1
                      )
                   else
                      ceil_num  = fix_num
                   )
                else
                   ceil_num  = fix_num
                )
             )
          else
             ceil_num  = fix_num
          )
       )
    else
       ceil_num  = fix_num
    )
 else
```

```
      ceil_num  = number
   )
   ceil_num
   ) ;end let statement
) ;end procedure
```

### *10.4.1.14. RoundToMaskGrid*

```
procedure( RoundToMaskGrid(number)
   let( (rounded_num grid)
   grid = GetGlobal("mask_grid")
   rounded_num = jsoRound(float(number)/float(grid))*grid
   rounded_num
   ) ; end let
) ; end procedure
```

### *10.4.1.15. FloorToMaskGrid*

```
procedure( FloorToMaskGrid(number)
   let( (rounded_num grid)
   grid = GetGlobal("mask_grid")
   rounded_num = jsoFloor(float(number)/float(grid))*grid
   rounded_num
   ) ; end let
) ; end procedure
```

### *10.4.1.16. CeilingToMaskGrid*

```
procedure( CeilingToMaskGrid(number)
   let( (rounded_num grid)
   grid = GetGlobal("mask_grid")
   rounded_num = jsoCeiling(float(number)/float(grid))*grid
   rounded_num
   ) ; end let
) ; end procedure
```

### *10.4.1.17. RoundToNumGrid*

```
procedure( RoundToNumGrid(number grid)
   let( (rounded_num)
   rounded_num = jsoRound(float(number)/float(grid))*grid
   rounded_num
   ) ; end let
) ; end procedure
```

### *10.4.1.18. FloorToNumGrid*

```
procedure( FloorToNumGrid(number grid)
```

```
  let( (rounded_num)
  rounded_num = jsoFloor(float(number)/float(grid))*grid
  rounded_num
  ) ; end let
) ; end procedure
```

#### *10.4.1.19. CeilingToNumGrid*

```
procedure( CeilingToNumGrid(number grid)
  let( (rounded_num)
  rounded_num = jsoCeiling(float(number)/float(grid))*grid
  rounded_num
  ) ; end let
) ; end procedure
```

### 10.4.2. Shape Discretization Functions

#### *10.4.2.1. uwlBend90*

```
procedure( uwlBend90( x0 y0 pc_x_origin pc_y_origin pc_rotation ring_radius ring_width grid mask_grid )
  let( (min_space min_width y_domain_max x_start y_start x_min x_max y_min y_max x_max_last x_min_last coords
coordList n_max i j i_min i_max j_min active radius radius_min radius_max inside i_test loopkill firstrun grid_offset_x
grid_offset_y)

    min_space = GetGlobal("min_space")
    min_width = GetGlobal("min_width")
    grid_offset_x = CalcOffsetX(pc_x_origin pc_y_origin pc_rotation grid mask_grid)
    grid_offset_y = CalcOffsetY(pc_x_origin pc_y_origin pc_rotation grid mask_grid)
    ; find and discretize the computation space
    radius_min = ring_radius-ring_width/2
    radius_max = ring_radius+ring_width/2
    n_max = jsoCeiling(radius_max/grid) + 3
    x_start = jsoFloor(x0/grid)*grid - grid_offset_x
    y_start = jsoFloor((y0-radius_max)/grid-2)*grid - grid_offset_y
    y_domain_max = y_start+n_max*grid
    ; initialize the coordinate list and toggles
    coordList = nil
    active = 0
    inside = 0
    firstrun = 1
    x_min = x_start
    y_min = y_start
    i_min = 0
    j_min = 0
    i_max = -10000
    x_max = -10000
    y_max = y_start
```

```
i = 0
; loop to step through discrete space to find coords:
for( j 0 (n_max-1)
  loopkill = 0
  if( i_min == 0
    then
      i=0
    else
      i=i_min
  )
  while( (i < n_max && loopkill == 0)
    ; calculate radius based on grid center point
    radius = sqrt(((x_start+grid*(i+0.5))-x0)**2+((y_start+grid*(j+0.5))-y0)**2)
    if( active == 1
      then
        ; if outside the ring
        if( (radius_min > (radius - 0.0001) || radius_max < (radius + 0.0001))
          then
            if( inside == 1
              then
                if( (not(i == i_max) && firstrun == 0 && not(j == j_min))
                  then
                    y_max = y_start+j*grid

      ; START COORDINATE CHECKING ROUTINE
        if( (abs(grid-mask_grid) < 0.001) then
          if( ((x_min > min_space) || (x_min < mask_grid)) then
            if( (x_max > min_width) then
              if( ((y_domain_max-y_max) > min_space) then
                if( ((y_domain_max-y_max) > min_width) then
                  coords = list( x_min:y_min x_max:y_max)
                  coordList = cons( coords coordList )
                else
                  x_max = jsoFloor((x0+radius_max)/grid)*grid
                  coords = list( x_min:y_min x_max:y_max)
                  coordList = cons( coords coordList )
                )
              else
                x_min = jsoFloor((x0+radius_min)/grid)*grid
                if( ((y_domain_max-y_max) > min_width) then
                  coords = list( x_min:y_min x_max:y_max)
                  coordList = cons( coords coordList )
                else
                  x_max = jsoFloor((x0+radius_max)/grid)*grid
                  coords = list( x_min:y_min x_max:y_max)
```

```
                coordList = cons( coords coordList )
              )
            )
          )
        else
          if( (x_max > min_space) then
            x_min = jsoFloor((x_start+min_space)/grid)*grid
            coords = list( x_min:y_min x_max:y_max)
            coordList = cons( coords coordList )
          )
        )
      else
        coords = list( x_min:y_min x_max:y_max)
        coordList = cons( coords coordList )
      )
    ; END COORDINATE CHECKING ROUTINE

                  y_min = y_start+j*grid
                  j_min = j
              )
              firstrun = 0
              x_max = x_start+i*grid
              i_max = i
              inside = 0
              loopkill = 1
            else
              i=i+1
          )
        else
          if( inside == 0
            then
              active = 1

              inside = 1
              if( not(i_min == i)
              then
                y_max = y_start+j*grid

    ; START COORDINATE CHECKING ROUTINE
      if( (abs(grid - mask_grid) < 0.001) then
        if( ((x_min > min_space) || (x_min < mask_grid)) then
          if( (x_max > min_width) then
            if( ((y_domain_max-y_max) > min_space) then
              if( ((y_domain_max-y_max) > min_width) then
                coords = list( x_min:y_min x_max:y_max)
                coordList = cons( coords coordList )
```

```
                else
                  x_max = jsoFloor((x0+radius_max)/grid)*grid
                  coords  = list( x_min:y_min x_max:y_max)
                  coordList  = cons( coords  coordList  )
                )
              else
                x_min = jsoFloor((x0+radius_min)/grid)*grid
                if( ((y_domain_max-y_max) > min_width) then
                  coords  = list( x_min:y_min x_max:y_max)
                  coordList  = cons( coords  coordList  )
                else
                  x_max = jsoFloor((x0+radius_max)/grid)*grid
                  coords  = list( x_min:y_min x_max:y_max)
                  coordList  = cons( coords  coordList  )
                )
              )
            )
          else
            if( (x_max > min_space) then
              x_min = jsoFloor((x_start+min_space)/grid)*grid
              coords  = list( x_min:y_min x_max:y_max)
              coordList  = cons( coords  coordList  )
            )
          )
        else
          coords  = list( x_min:y_min x_max:y_max )
          coordList  = cons( coords  coordList  )
        )
      ; END COORDINATE CHECKING ROUTINE

                  y_min = y_start+j*grid
                  j_min = j
                )
                x_min = x_start+i*grid
                i_min = i
              )
              i=i+1
            )
          else
            if( (radius_min < (radius + 0.0001) && radius_max > (radius - 0.0001))
              then
                x_min = x_start+i*grid
                y_min = y_start+j*grid
                i_min = i
                j_min = j
                active = 1
```

```
                inside = 1
            )
            i=i+1
        ) ;end radius if blocks
    ) ;end x_steping loop
) ;end y_stepping loop

y_max = y_start+n_max*grid

        ; START COORDINATE CHECKING ROUTINE
            if( (abs(grid - mask_grid) < 0.001) then
                if( ((x_min > min_space) || (x_min < mask_grid)) then
                    if( (x_max > min_width) then
                        if( ((y_domain_max-y_max) > min_space) then
                            if( ((y_domain_max-y_max) > min_width) then
                                coords = list( x_min:y_min x_max:y_max)
                                coordList = cons( coords coordList )
                            else
                                x_max = jsoFloor((x0+radius_max)/grid)*grid
                                coords = list( x_min:y_min x_max:y_max)
                                coordList = cons( coords coordList )
                            )
                        else
                            x_min = jsoFloor((x0+radius_min)/grid)*grid
                            if( ((y_domain_max-y_max) > min_width) then
                                coords = list( x_min:y_min x_max:y_max)
                                coordList = cons( coords coordList )
                            else
                                x_max = jsoFloor((x0+radius_max)/grid)*grid
                                coords = list( x_min:y_min x_max:y_max)
                                coordList = cons( coords coordList )
                            )
                        )
                    )
                else
                    if( (x_max > min_space) then
                        x_min = jsoFloor((x_start+min_space)/grid)*grid
                        coords = list( x_min:y_min x_max:y_max)
                        coordList = cons( coords coordList )
                    )
                )
            else
                coords = list( x_min:y_min x_max:y_max )
                coordList = cons( coords coordList )
            )
        ; END COORDINATE CHECKING ROUTINE
```

```
    coordList   ;procedure returns this list
  ) ;end let statement
) ;end procedure uwlBend90
```

### *10.4.2.2. uwlBend180NoMirror*

```
procedure( uwlBend180NoMirror( x0 y0 pc_x_origin pc_y_origin pc_rotation ring_radius ring_width design_grid mask_grid )
  let( (x_start y_start x_min x_max y_min y_max x_max_last x_min_last coords coordList n_max i j i_min i_max j_min active
radius radius_min radius_max inside i_test loopkill firstrun grid_offset_x grid_offset_y)

    grid_offset_x  = CalcOffsetX(pc_x_origin pc_y_origin pc_rotation design_grid mask_grid)
    grid_offset_y  = CalcOffsetY(pc_x_origin pc_y_origin pc_rotation design_grid mask_grid)
    ; find and discretize the computation space
    radius_min  = ring_radius-ring_width/2
    radius_max  = ring_radius+ring_width/2
    n_max = jsoCeiling(2*radius_max/design_grid)  + 5
    x_start = jsoFloor(x0/design_grid)*design_grid  - grid_offset_x
    y_start = jsoFloor((y0-radius_max)/design_grid-2)*design_grid  - grid_offset_y
    ; initialize the coordinate list and toggles
    coordList  = nil
    active  = 0
    inside  = 0
    firstrun  = 1
    x_min = x_start
    y_min = y_start
    i_min = 0
    j_min = 0
    i_max = -10000
    x_max = -10000
    y_max = y_start

    i = 0
    ; loop to step through discrete space to find coords:
    for( j 0 (n_max-1)
      loopkill  = 0
      if( i_min == 0
        then
          i=0
        else
      if( j < n_max/2
      then
        i=i_min-1
      else
        i=max(i_min-10  0)
      )
```

```
)

while( (i < n_max && loopkill == 0)
  ; calculate radius based on design grid center point
  radius = sqrt(((x_start+design_grid*(i+0.5))-x0)**2+((y_start+design_grid*(j+0.5))-y0)**2)
  if( active == 1
    then
      ; if outside the ring
      if( (radius_min > (radius - 0.0001) || radius_max < (radius + 0.0001))
        then
          if( inside == 1
            then
              if( (not(i == i_max) && firstrun == 0 && not(j == j_min))
                then
                  y_max = y_start+j*design_grid
                  coords = list( x_min:y_min x_max:y_max)
                  coordList = cons( coords coordList )
                  y_min = y_start+j*design_grid
                  j_min = j
              )
              firstrun = 0
              x_max = x_start+i*design_grid
              i_max = i
              inside = 0
              loopkill = 1
            else
              i=i+1
          )
        else
          if( inside == 0
            then
              active = 1

              inside = 1
              if( not(i_min == i)
              then
                y_max = y_start+j*design_grid
                coords = list( x_min:y_min x_max:y_max)
                coordList = cons( coords coordList )
                y_min = y_start+j*design_grid
                j_min = j
              )
              x_min = x_start+i*design_grid
              i_min = i
          )
          i=i+1
```

```
            )
          else
            if( (radius_min < (radius + 0.0001) && radius_max > (radius - 0.0001))
              then
                x_min = x_start+i*design_grid
                y_min = y_start+j*design_grid
                i_min = i
                j_min = j
                active = 1
                inside = 1
            )
            i=i+1
        ) ;end radius if blocks
      ) ;end x_steping loop
    ) ;end y_stepping loop

    coordList  ;procedure returns this list
  ) ;end let statement
) ;end procedure uwlBend180NoMirror
```

### *10.4.2.3. uwlRingRectanglesNoMirror*

```
procedure( uwlRingRectanglesNoMirror( x0 y0 pc_x_origin pc_y_origin pc_rotation ring_radius ring_width grid mask_grid )
  let( (grid_offset_x grid_offset_y x_start y_start x_min x_min_full_last x_min_right_last x_min_left_last x_max
x_max_full_last x_max_right_last x_max_left_last y_min_full y_min_left y_min_right y_max x_max_last x_min_last coords
coordList n_max i j radius radius_min radius_max inside right_active left_active center_active
center_was_active_for_right)

    grid_offset_x = CalcOffsetX(pc_x_origin pc_y_origin pc_rotation grid mask_grid)
    grid_offset_y = CalcOffsetY(pc_x_origin pc_y_origin pc_rotation grid mask_grid)
    ; find and discretize the computation space
    radius_min = ring_radius-ring_width/2
    radius_max = ring_radius+ring_width/2
    n_max = jsoRound(2*radius_max/grid) + 5
    x_start = jsoFloor((x0-radius_max)/grid-2)*grid - grid_offset_x
    y_start = jsoFloor((y0-radius_max)/grid-2)*grid - grid_offset_y
    ; initialize the coordinate list and toggles
    coordList = nil
    inside = 0
    initial = 1
    right_active = -1
    left_active = -1
    center_active = -1
    center_was_active_for_right = -1
    x_min = x_start
    x_min_full_last = -10000
```

```
x_min_right_last  = -10000
x_min_left_last  = -10000
y_min_full  = y_start
y_min_left  = y_start
y_min_right  = y_start
x_max = -10000
x_max_full_last  = -10000
x_max_right_last  = -10000
x_max_left_last  = -10000
y_max = y_start

; loop to step through discrete space to find coords:
for( j 0 (n_max-1)
  for(i 0 (n_max-1)

; calculate radius based on grid center point
radius  = sqrt(((x_start+grid*(i+0.5))-x0)**2+((y_start+grid*(j+0.5))-y0)**2)

; if outside the ring
if( (radius_min  > (radius  - 0.0001) || radius_max  < (radius  + 0.0001))
  then
  if( inside  == 1
    then
    y_max = y_start+(j+1)*grid
    x_max = x_start+i*grid

    if( x_min < x0
      then
        if( x_max < x0
          then
          ; Left only rectangle
            left_active  = 1
            if( center_active  == -1
              then
                if( not(x_min  == x_min_left_last)  || not(x_max  == x_max_left_last)
                  then
                    coords     = list( x_min:y_min_left  x_max:y_max)
                    coordList  = cons( coords  coordList  )
                    y_min_left      = y_start+(j+1)*grid
                    x_min_left_last  = x_min
                    x_max_left_last  = x_max
                    left_active  = 0
                )
              else
                y_min_left  = y_start+j*grid
                if( center_active  == 1
```

```
                then
                    coords    = list( x_min_full_last:y_min_full x_max_full_last:y_min_left)
                    coordList = cons( coords coordList )
                )
                center_active = -1
                x_min_left_last = x_min
                x_max_left_last = x_max
                center_was_active_for_right = 1
            )
        else
        ; full rectangle
            center_active = 1
            if( left_active == -1 && right_active == -1
                then
                    if( not(x_min == x_min_full_last) || not(x_max == x_max_full_last)
                        then
                            coords    = list( x_min:y_min_full x_max:y_max)
                            coordList = cons( coords coordList )
                            y_min_full     = y_start+(j+1)*grid
                            x_min_full_last = x_min
                            x_max_full_last = x_max
                            center_active = 0
                    )
                    else
                        y_min_full     = y_start+j*grid
                        if( right_active == 1
                            then
                                coords    = list( x_min_right_last:y_min_right x_max_right_last:y_min_full)
                                coordList = cons( coords coordList )
                        )
                        if( left_active == 1
                            then
                                coords    = list( x_min_left_last:y_min_left x_max_left_last:y_min_full)
                                coordList = cons( coords coordList )
                        )
                        left_active = -1
                        right_active = -1
                        x_min_full_last = x_min
                        x_max_full_last = x_max
            )
    )
else
; right only rectangle
    right_active = 1
    if( center_was_active_for_right == -1
        then
```

```
                    if( not(x_min == x_min_right_last) || not(x_max == x_max_right_last)
                      then
                        coords     = list( x_min:y_min_right x_max:y_max)
                        coordList  = cons( coords coordList )
                        y_min_right      = y_start+(j+1)*grid
                        y_min_full       = y_start+(j+1)*grid
                        x_min_right_last = x_min
                        x_max_right_last = x_max
                        x_min_full_last  = -10000
                        right_active = 0
                    )
                  else
                    center_was_active_for_right = -1
                    y_min_right = y_start+j*grid
                    x_min_right_last = x_min
                    x_max_right_last = x_max
                )
            )
            inside = 0
        )
        else
        if( inside == 0
          then
          x_min = x_start+i*grid
          if( initial == 1
            then
              y_min_full = y_start+j*grid
              y_min_left = y_start+j*grid
              y_min_right = y_start+j*grid
              initial = 0
          )
          inside = 1
        )
    )
      ) ;end x_steping loop
    ) ;end y_stepping loop

    coordList   procedure returns this list
  ) ;end let statement
) ;end procedure uwlRingRectanglesNoMirror
```

### *10.4.2.4. uwlTaperRectangles*

```
procedure( uwlTaperRectangles( x0_1 y0_1 pc_x_origin_1 pc_y_origin_1 pc_rotation_1 wg_width end_width taper_length grid
mask_grid)
```

```
let( (min_width min_space y_domain_max x_max x_min y_max y_min coords coordList i i_max j j_max slope x_delta_last
x_delta slope active grid_offset_x grid_offset_y x_start y_start)

    min_width = GetGlobal("min_width")
    min_space = GetGlobal("min_space")
    grid_offset_x = CalcOffsetX(pc_x_origin_1 pc_y_origin_1 pc_rotation_1 grid mask_grid)
    grid_offset_y = CalcOffsetY(pc_x_origin_1 pc_y_origin_1 pc_rotation_1 grid mask_grid)
    x_start = jsoRound(x0_1/grid)*grid - grid_offset_x
    y_start = jsoFloor(y0_1/grid)*grid - grid_offset_y
    ; initialize the coordinate list and toggles
    coordList = nil
    slope = (end_width-wg_width)/float(taper_length)
    if( grid == mask_grid
      then
        j_max = jsoRound(taper_length/grid)-1
      else
        j_max = jsoRound(taper_length/grid)
    )
    y_domain_max = y_start+(j_max+1)*grid
    y_min = y_start
    y_max = -10000
    x_delta_last = jsoRound(wg_width/2)
    active = 0
    ; loop to step through discrete space to find coords:
    for( j 0 j_max
      active = 1
      x_delta = jsoRound(((j+0.5)*slope*grid+wg_width)/2.0/grid)*grid
      if( not( x_delta_last == x_delta )
        then
          y_max = y_start+(j+1)*grid
          x_min = x_start-x_delta
          x_max = x_start+x_delta

          ; START COORDINATE CHECKING ROUTINE
            if( ((abs(grid - mask_grid) < 0.001) && (abs(taper_length) > 1.0)) then
              if( (y_min < min_space) then
                x_min = x_start - jsoRound(wg_width/2.0/grid)*grid
                x_max = x_start + jsoRound(wg_width/2.0/grid)*grid
                coords = list( x_min:y_min x_max:y_max)
                coordList = cons( coords coordList )
              else
                if( ((y_domain_max-y_max) < min_space) then
                  x_min = x_start - jsoRound(end_width/2.0/grid)*grid
                  x_max = x_start + jsoRound(end_width/2.0/grid)*grid
                  coords = list( x_min:y_min x_max:y_max)
                  coordList = cons( coords coordList )
```

```
                else
                   coords = list( x_min:y_min x_max:y_max)
                   coordList = cons( coords coordList )
                )
             )
          else
             coords = list( x_min:y_min x_max:y_max )
             coordList = cons( coords coordList )
          )
       ; END COORDINATE CHECKING ROUTINE

       y_min = y_max
       x_delta_last = x_delta
       active = 0
    )
) ;end j_stepping loop

if( active == 1
   then
      y_max = y_start+(j_max+1)*grid
      x_min = x_start-x_delta
      x_max = x_start+x_delta

       ; START COORDINATE CHECKING ROUTINE
          if( ((abs(grid - mask_grid) < 0.001) && (abs(taper_length) > 1.0)) then
             if( (y_min < min_space) then
                x_min = x_start - jsoRound(wg_width/2.0/grid)*grid
                x_max = x_start + jsoRound(wg_width/2.0/grid)*grid
                coords = list( x_min:y_min x_max:y_max)
                coordList = cons( coords coordList )
             else
                if( ((y_domain_max-y_max) < min_space) then
                   x_min = x_start - jsoRound(end_width/2.0/grid)*grid
                   x_max = x_start + jsoRound(end_width/2.0/grid)*grid
                   coords = list( x_min:y_min x_max:y_max)
                   coordList = cons( coords coordList )
                else
                   coords = list( x_min:y_min x_max:y_max)
                   coordList = cons( coords coordList )
                )
             )
          else
             coords = list( x_min:y_min x_max:y_max )
             coordList = cons( coords coordList )
          )
       ; END COORDINATE CHECKING ROUTINE
```

```
    )

    coordList   ;procedure returns this list
  ) ;end let statement
) ;end
```

### *10.4.2.5. uwlSinusoid*

```
procedure( uwlSinusoid( x0 y0 pc_x_origin pc_y_origin pc_rotation wg_width delta_x delta_y design_grid mask_grid )
  let( (min_space min_width x_start y_start x_min x_max y_min y_max x_max_last x_min_last coords coordList n_max
m_max i j i_min j_min active radius radius_min radius_max inside i_test loopkill firstrun pi y_pos x_pos y_pos_min
y_pos_max x_pos_max slope w_eff grid_offset_x grid_offset_y grid)
    ; find and discretize the computation space

  grid = design_grid
  min_space = GetGlobal("min_space")
  min_width = GetGlobal("min_width")
  grid_offset_x = CalcOffsetX(pc_x_origin pc_y_origin pc_rotation design_grid mask_grid)
  grid_offset_y = CalcOffsetY(pc_x_origin pc_y_origin pc_rotation design_grid mask_grid)
  n_max = jsoRound((delta_y+wg_width)/design_grid)

  if( (abs(design_grid - mask_grid) < 0.001) then
    m_max = jsoRound(delta_x/design_grid) + 2
    x_start = jsoFloor(x0/design_grid)*design_grid - grid_offset_x
    x_pos_max = x0 + delta_x
  else
    m_max = jsoCeiling(delta_x/design_grid) + 10
    x_start = jsoFloor(x0/design_grid)*design_grid - design_grid - grid_offset_x
    x_pos_max = x0 + delta_x + 4*design_grid
  )
    y_start = jsoFloor((y0 - wg_width/2)/design_grid)*design_grid - grid_offset_y
    pi = 3.14159
    ; initialize the coordinate list and toggles
    coordList = nil
    active = 0
    inside = 0
    firstrun = 1
    x_min = x_start
    y_min = y_start
    i_min = 0
    j_min = 0
    i_max = -10000
    x_max = -10000
    y_max = y_start
    i = 0
    ; loop to step through discrete space to find coords:
```

```
for( j 0 (n_max-1)
  loopkill = 0
  if( i_min == 0
    then
      i = 0
    else
      i = i_min -5);
  while( ((i < m_max) && loopkill == 0)
    ; calculate radius based on grid center point
    x_pos = x_start + (i+0.5)*design_grid
    y_pos = y_start + (j+0.5)*design_grid
    slope = 0.5*delta_y/delta_x*pi*sin(pi*(x_pos-x0)/delta_x)
    w_eff = wg_width/cos(pi/2-atan(1/(slope + 1e-32)))
    y_pos_max = y0 + 0.5*delta_y*(1-cos(pi*(x_pos-x0)/delta_x)) + w_eff/2
    y_pos_min = y0 + 0.5*delta_y*(1-cos(pi*(x_pos-x0)/delta_x)) - w_eff/2
    if( active == 1
      then
        ; if outside the structure
        if( ( ((y_pos_min > (y_pos + 0.0001)) || (y_pos_max < (y_pos - 0.0001))) || (x_pos_max < (x_pos - 0.001)) )
          then
            if( inside == 1
              then
                if( (not(i == i_max) && firstrun == 0 && not(j == j_min))
                  then
                    y_max = y_start+j*design_grid

      ; START COORDINATE CHECKING ROUTINE
        if( (abs(grid - mask_grid) < 0.001) then
          if( ((x_min > min_space) || (x_min < mask_grid)) then
            if( (x_max > min_width) then
              if( (((x_pos_max-x_max) > min_space) || ((x_pos_max-x_max) < mask_grid)) then
                if( ((x_pos_max-x_min) > min_width) then
                  coords = list( x_min:y_min x_max:y_max)
                  coordList = cons( coords coordList )
                )
              else
                if( ((x_pos_max-x_min) > min_space) then
                  x_max = jsoFloor((x_pos_max-min_space)/grid)*grid
                  coords = list( x_min:y_min x_max:y_max)
                  coordList = cons( coords coordList )
                )
              )
            )
          else
            if( (x_max > min_space) then
              x_min = jsoFloor((x_start+min_space)/grid)*grid
```

```
            coords = list( x_min:y_min x_max:y_max)
            coordList = cons( coords coordList )
          )
        )
      else
        coords = list( x_min:y_min x_max:y_max )
        coordList = cons( coords coordList )
      )
; END COORDINATE CHECKING ROUTINE

                  y_min = y_start+j*design_grid
                  j_min = j
              )
              firstrun = 0
              x_max = x_start+i*design_grid
              i_max = i
              inside = 0
              loopkill = 1
            else
              i=i+1
          )
        else
          if( inside == 0
            then
              active = 1

              inside = 1
              if( not(i_min == i)
              then
                y_max = y_start+j*design_grid

; START COORDINATE CHECKING ROUTINE
  if( (abs(grid - mask_grid) < 0.001) then
    if( ((x_min > min_space) || (x_min < mask_grid)) then
      if( (x_max > min_width) then
        if( (((x_pos_max-x_max) > min_space) || ((x_pos_max-x_max) < mask_grid)) then
          if( ((x_pos_max-x_min) > min_width) then
            coords = list( x_min:y_min x_max:y_max)
            coordList = cons( coords coordList )
          )
        else
          if( ((x_pos_max-x_min) > min_space) then
            x_max = jsoFloor((x_pos_max-min_space)/grid)*grid
            coords = list( x_min:y_min x_max:y_max)
            coordList = cons( coords coordList )
          )
```

```
                    )
                  )
                else
                  if( (x_max > min_space) then
                    x_min = jsoFloor((x_start+min_space)/grid)*grid
                    coords = list( x_min:y_min x_max:y_max)
                    coordList = cons( coords coordList )
                  )
                )
              else
                coords = list( x_min:y_min x_max:y_max )
                coordList = cons( coords coordList )
              )
            ; END COORDINATE CHECKING ROUTINE

                      y_min = y_start+j*design_grid
                      j_min = j
                    )
                    x_min = x_start+i*design_grid
                    i_min = i
                  )
                  i=i+1
                )
              else
                if( ((y_pos_min < (y_pos + 0.0001)) && (y_pos_max > (y_pos - 0.0001)))
                  then
                    x_min = x_start+i*design_grid
                    y_min = y_start+j*design_grid
                    i_min = i
                    j_min = j
                    active = 1
                    inside = 1
                )
                i=i+1
            ) ;end radius if blocks
          ) ;end x_steping loop
        ) ;end y_stepping loop

        y_max = y_start+n_max*design_grid

            ; START COORDINATE CHECKING ROUTINE
              if( (abs(grid - mask_grid) < 0.001) then
                if( ((x_min > min_space) || (x_min < mask_grid)) then
                  if( (x_max > min_width) then
                    if( (((x_pos_max-x_max) > min_space) || ((x_pos_max-x_max) < mask_grid)) then
                      if( ((x_pos_max-x_min) > min_width) then
```

```
                        coords = list( x_min:y_min x_max:y_max)
                        coordList = cons( coords coordList )
                      )
                    else
                      if( ((x_pos_max-x_min) > min_space) then
                        x_max = jsoFloor((x_pos_max-min_space)/grid)*grid
                        coords = list( x_min:y_min x_max:y_max)
                        coordList = cons( coords coordList )
                      )
                    )
                  )
                else
                  if( (x_max > min_space) then
                    x_min = jsoFloor((x_start+min_space)/grid)*grid
                    coords = list( x_min:y_min x_max:y_max)
                    coordList = cons( coords coordList )
                  )
                )
              else
                coords = list( x_min:y_min x_max:y_max )
                coordList = cons( coords coordList )
              )
            ; END COORDINATE CHECKING ROUTINE

    coordList   procedure returns this list
  ) ;end let statement
) ;end procedure uwlSinusoid
```

### 10.4.3. Basic Geometry P-Cell Examples

#### *10.4.3.1. arc.il*

```
pcDefinePCell(
    list( ddGetObj("genericPhotonicsLib") "arc" "layout")
    ; parameters and their optional defaults:
    (
        (layer          "POLY")
        (width           0.370)
        (radius          5.000)
    ) ;end of parameter list
    ; geometry creation body of code:
    let(  (
       coords coordList x0 y0 biased_width mask_grid pc_x_origin pc_y_origin pc_rotation
    )
        biased_width = PolyMaskSize(width)
        mask_grid = GetGlobal("mask_grid")
```

```
        ; set the ring center
        pc_x_origin = 0
        pc_y_origin = 0
        pc_rotation = "R0"
        x0 = 0
        y0 = radius

        coordList = uwlBend90( x0 y0 pc_x_origin pc_y_origin pc_rotation radius biased_width mask_grid mask_grid)
        foreach( coords coordList
      BlockWriter(coords layer)
        ) ;end of rectangle creation loop

    ) ;end of body of code
) ;end of pcell definition
```

### *10.4.3.2. taper.il*

```
pcDefinePCell(
  list( ddGetObj("genericPhotonicsLib") "taper" "layout")
  ; parameters and their optional defaults:
  (
    (layer      "POLY")
    (shape      "LINEAR")
    (start_width  0.35)
    (end_width    1.00)
    (length     10.00)
  ) ;end of parameter list
  ; geometry creation body of code:
  let( (
      x0 y0 pc_x_origin pc_y_origin pc_rotation coords coordList biased_start_width biased_end_width mask_grid
    )
    pc_x_origin= 0.0
    pc_y_origin= 0.0
    pc_rotation= "R0"
    biased_start_width = PolyMaskSize(start_width)
    biased_end_width = PolyMaskSize(end_width)
    mask_grid = GetGlobal("mask_grid")
    x0 = 0.0
    y0 = 0.0
    ; make the taper
    case( shape
      ( "LINEAR"
        coordList = uwlTaperRectangles( x0 y0 pc_x_origin pc_y_origin pc_rotation biased_start_width biased_end_width
length mask_grid mask_grid)
      )
      ( t
```

```
            coordList = 0.0 ; Nonlinear taper functions not included in thesis code
          )
        )
        foreach( coords coordList
          BlockWriter(coords layer)
        ) ;end of rectangle creation loop
    ) ;end of body of code
) ;end of pcell definition
```

### *10.4.3.3. offset.il*

```
pcDefinePCell(
    list( ddGetObj("genericPhotonicsLib") "offset" "layout")
    ; parameters and their optional defaults:
    (     (width      0.370)
      (delta_x          3.000)
      (delta_y          1.000)
      (layer          "POLY")
    ) ;end of parameter list
    ; geometry creation body of code:
    let(  (
        biased_width flip coords coordList x_a x_b y_a y_b x0 y0 pc_x_origin pc_y_origin pc_rotation mask_grid
      )
      flip = nil
      biased_width = PolyMaskSize(width)
      mask_grid = GetGlobal("mask_grid")
      ; calculate the ring center
      x0 = 0
      y0 = 0
      pc_x_origin = 0
      pc_y_origin = 0
      pc_rotation = "R0"
      if( (delta_y < 0.0) then
        delta_y = 0.0-delta_y
        flip = 1
      )
      coordList = uwlSinusoid( x0 y0 pc_x_origin pc_y_origin pc_rotation biased_width delta_x delta_y mask_grid mask_grid)
      ; make ring rectangles calculated in the subroutine:
      foreach( coords coordList
        if( flip then
          x_a = caar(coords)
          y_a = -cadar(coords)
          x_b = caadr(coords)
          y_b = -cadadr(coords)
          coords = list( x_a:y_a x_b:y_b )
        )
```

```
        BlockWriter(coords layer)
      ) ;end of rectangle creation loop
   ) ;end of body of code
) ;end of pcell definition
```

### 10.4.4. Waveguide-Level Routing P-Cell Examples

#### *10.4.4.1. wg.il*

```
pcDefinePCell(
   list( ddGetObj("genericPhotonicsLib") "wg" "layout")
   ; parameters and their optional defaults:
   (
      (width_mode          "FIXED") ; or "SINGLEMODE"
      (thickness           "200")
      (wavelength          "1550")
      (layer          "POLY")
      (width           0.370)
      (length         1.000)
      (continuity          "THROUGH")
      (pc_x_origin      0.0)
      (pc_y_origin      0.0)
      (pc_rotation      "R0")
   ) ;end of parameter list
   ; creation body of code:
   let(  (
         x0 y0 x1 y1 x2 y2 grid_offset_x grid_offset_y design_grid overhang input_ov_truth output_ov_truth
         x_min x_max y_min y_max biased_width mask_grid dopeblk_grid fillblk_grid
         dopeblk_overhang fillblk_overhang dopeblock_layer_name
      )
      if( (strcmp(width_mode "SINGLEMODE") == 0) then
          width = GetSingleModeWidth(thickness wavelength)
      )
      biased_width = PolyMaskSize(width)
      mask_grid = GetGlobal("mask_grid")
      dopeblk_grid = GetGlobal("dopeblk_grid")
      fillblk_grid = GetGlobal("fillblk_grid")
      dopeblk_overhang = GetGlobal("dopeblk_overhang")
      fillblk_overhang = GetGlobal("fillblk_overhang")
      dopeblock_layer_name = "DOPEBLOCK"
      if( (strcmp(layer "POLY_NARROW_DOPEBLOCK") == 0) then
      layer = "POLY"
         dopeblk_overhang = 0.9
      )
            input_ov_truth = 1.0
            output_ov_truth = 1.0
```

```
case( continuity
   ( "THROUGH"
      input_ov_truth = 0.0
      output_ov_truth = 0.0
   )
   ( "INPUT"
      input_ov_truth = 1.0
      output_ov_truth = 0.0
   )
   ( "OUTPUT"
      input_ov_truth = 0.0
      output_ov_truth = 1.0
   )
   ( "ISOLATED"
      input_ov_truth = 1.0
      output_ov_truth = 1.0
   )
)

x_min = 0
x_max = length
y_min = 0-jsoRound(biased_width/2/mask_grid)*mask_grid
y_max = jsoRound(biased_width/2/mask_grid)*mask_grid
BlockWriter(list(x_min:y_min x_max:y_max) layer)

x1 = x_min
x2 = x_max
y1 = y_min
y2 = y_max
design_grid = dopeblk_grid
overhang = dopeblk_overhang
grid_offset_x = CalcOffsetX(pc_x_origin pc_y_origin pc_rotation design_grid mask_grid)
grid_offset_y = CalcOffsetY(pc_x_origin pc_y_origin pc_rotation design_grid mask_grid)
x_min = jsoFloor((x1-overhang*input_ov_truth)/design_grid)*design_grid-grid_offset_x
if( x_min > x1
   then
      x_min = x_min - design_grid
)

x_max = jsoCeiling((length+overhang*output_ov_truth)/design_grid)*design_grid-grid_offset_x
if( x_max < x2
   then
      x_max = x_max + design_grid
)
y_min = 0-jsoRound((biased_width/2+overhang)/design_grid)*design_grid-grid_offset_y
y_max = jsoRound((biased_width/2+overhang)/design_grid)*design_grid-grid_offset_y
```

```
    BlockWriter(list(x_min:y_min x_max:y_max) dopeblock_layer_name )

    input_ov_truth = 1.0
    output_ov_truth = 1.0
    design_grid = fillblk_grid
    overhang = fillblk_overhang
    grid_offset_x = CalcOffsetX(pc_x_origin pc_y_origin pc_rotation design_grid mask_grid)
    grid_offset_y = CalcOffsetY(pc_x_origin pc_y_origin pc_rotation design_grid mask_grid)
    x_min = jsoFloor((x1-overhang*input_ov_truth)/design_grid)*design_grid-grid_offset_x
    if( x_min > x1
      then
        x_min = x_min - design_grid
    )
    x_max = jsoCeiling((length+overhang*output_ov_truth)/design_grid)*design_grid-grid_offset_x
    if( x_max < x2
      then
        x_max = x_max + design_grid
    )
    y_min = 0-jsoRound((biased_width/2+overhang)/design_grid)*design_grid-grid_offset_y
    y_max = jsoRound((biased_width/2+overhang)/design_grid)*design_grid-grid_offset_y
    BlockWriter(list(x_min:y_min x_max:y_max) "FILLBLOCK")
  ) ;end of body of code
) ;end of pcell definition
```

### *10.4.4.2. wg_bend.il*

```
pcDefinePCell(
  list( ddGetObj("genericPhotonicsLib") "wg_bend" "layout")
  ; parameters and their optional defaults:
  (
    (width_mode    "FIXED") ; or "SINGLEMODE"
    (thickness    "200")
    (wavelength    "1550")
    (layer         "POLY")
    (angle              "90")
    (width           0.370)
    (radius          5.000)
    (pc_x_origin      0.0)
    (pc_y_origin      0.0)
    (pc_rotation      "R0")
  ) ;end of parameter list
  ; geometry creation body of code:
  let(  (
      coords coordList x0 y0 x_min x_max y_min y_max grid_offset_x grid_offset_y biased_width
      effective_radius effective_width mask_grid dopeblk_grid fillblk_grid dopeblk_overhang_inner
      dopeblk_overhang_outer fillblk_overhang_inner fillblk_overhang_outer arc_master inst dopeblock_layer_name
```

```
    )
    if( (strcmp(width_mode "SINGLEMODE") == 0) then
       width = GetSingleModeWidth(thickness wavelength)
    )
    biased_width = PolyMaskSize(width)
    mask_grid = GetGlobal("mask_grid")
    dopeblk_grid = GetGlobal("dopeblk_grid")
    fillblk_grid = GetGlobal("fillblk_grid")
    dopeblk_overhang_inner = GetGlobal("dopeblk_overhang")
    fillblk_overhang_inner = GetGlobal("fillblk_overhang")
    dopeblk_overhang_outer = GetGlobal("dopeblk_overhang")
    fillblk_overhang_outer = GetGlobal("fillblk_overhang")
    ; arc origins
    x0 = 0
    y0 = 0
    arc_master = dbOpenCellViewByType( ddGetObj("genericPhotonicsLib") "arc" "layout")
    inst = dbCreateParamInst( pcCellView arc_master nil x0:y0 "R0" 1
       list(
          list("layer" "string" layer)
          list("width" "float" width)
          list("radius" "float" radius)
       )
    )
    x0 = 0
    y0 = 2*radius
    if( (strcmp(angle "180") == 0) then
          inst = dbCreateParamInst( pcCellView arc_master nil x0:y0 "MX" 1
             list(
                list("layer" "string" layer)
                list("width" "float" width)
                list("radius" "float" radius)
             )
          )
    )
    ; calculate the ring center
    x0 = 0
    y0 = radius
    effective_radius = radius + (dopeblk_overhang_outer-dopeblk_overhang_inner)/2
     effective_width = dopeblk_overhang_outer+width+dopeblk_overhang_inner
    case(angle
       ( ("90")
          coordList = uwlBend90( x0 y0 pc_x_origin pc_y_origin pc_rotation effective_radius effective_width dopeblk_grid
mask_grid)
       )
       ( ("180")
```

```
        coordList = uwlBend180NoMirror( x0 y0 pc_x_origin pc_y_origin pc_rotation effective_radius effective_width
dopeblk_grid mask_grid)
      )
    )
    foreach( coords coordList
      BlockWriter(coords dopeblock_layer_name)
    ) ;end of rectangle creation loop

    effective_radius = radius + (fillblk_overhang_outer-fillblk_overhang_inner)/2
        effective_width = fillblk_overhang_outer+width+fillblk_overhang_inner
    case(angle
      ( ("90")
        coordList = uwlBend90( x0 y0 pc_x_origin pc_y_origin pc_rotation effective_radius effective_width fillblk_grid
mask_grid)
      )
      ( ("180")
        coordList = uwlBend180NoMirror( x0 y0 pc_x_origin pc_y_origin pc_rotation effective_radius effective_width
fillblk_grid mask_grid)
      )
    )
    foreach( coords coordList
      BlockWriter(coords "FILLBLOCK")
    ) ;end of rectangle creation loop

    grid_offset_x = CalcOffsetX(pc_x_origin pc_y_origin pc_rotation fillblk_grid mask_grid)
    grid_offset_y = CalcOffsetY(pc_x_origin pc_y_origin pc_rotation fillblk_grid mask_grid)

    x_min = jsoFloor(x0/fillblk_grid)*fillblk_grid - fillblk_grid - grid_offset_x
    x_max = jsoCeiling(x0/fillblk_grid)*fillblk_grid - grid_offset_x
    y_max = jsoRound((y0-effective_radius+effective_width/2)/fillblk_grid)*fillblk_grid - grid_offset_y
    y_min = jsoRound((y0-effective_radius-effective_width/2)/fillblk_grid)*fillblk_grid - grid_offset_y
    BlockWriter(list(x_min:y_min x_max:y_max) "FILLBLOCK")

    case(angle
      ( ("90")
        x_min = jsoRound((x0+effective_radius-effective_width/2)/fillblk_grid)*fillblk_grid - grid_offset_x
        x_max = jsoRound((x0+effective_radius+effective_width/2)/fillblk_grid)*fillblk_grid - grid_offset_x
        y_max = jsoCeiling(y0/fillblk_grid)*fillblk_grid + fillblk_grid - grid_offset_y
        y_min = jsoFloor(y0/fillblk_grid)*fillblk_grid - grid_offset_y
      )
      ( ("180")
        x_min = jsoFloor(x0/fillblk_grid)*fillblk_grid - fillblk_grid - grid_offset_x
        x_max = jsoCeiling(x0/fillblk_grid)*fillblk_grid - grid_offset_x
        y_max = jsoRound((y0+effective_radius+effective_width/2)/fillblk_grid)*fillblk_grid - grid_offset_y
        y_min = jsoRound((y0+effective_radius-effective_width/2)/fillblk_grid)*fillblk_grid - grid_offset_y
      )
```

```
    )
    BlockWriter(list(x_min:y_min x_max:y_max) "FILLBLOCK")

    effective_radius = radius + (dopeblk_overhang_outer-dopeblk_overhang_inner)/2
    effective_width = dopeblk_overhang_outer+width+dopeblk_overhang_inner
    grid_offset_x = CalcOffsetX(pc_x_origin pc_y_origin pc_rotation dopeblk_grid mask_grid)
    grid_offset_y = CalcOffsetY(pc_x_origin pc_y_origin pc_rotation dopeblk_grid mask_grid)
    x_min = jsoFloor(x0/dopeblk_grid)*dopeblk_grid - dopeblk_grid - grid_offset_x
    x_max = jsoCeiling(x0/dopeblk_grid)*dopeblk_grid - grid_offset_x
    y_max = jsoRound((y0-effective_radius+effective_width/2)/dopeblk_grid)*dopeblk_grid - grid_offset_y
    y_min = jsoRound((y0-effective_radius-effective_width/2)/dopeblk_grid)*dopeblk_grid - grid_offset_y
    BlockWriter(list(x_min:y_min x_max:y_max) dopeblock_layer_name)

    case(angle
      ("90")
        x_min = jsoRound((x0+effective_radius-effective_width/2)/dopeblk_grid)*dopeblk_grid - grid_offset_x
        x_max = jsoRound((x0+effective_radius+effective_width/2)/dopeblk_grid)*dopeblk_grid - grid_offset_x
        y_max = jsoCeiling(y0/dopeblk_grid)*dopeblk_grid + dopeblk_grid - grid_offset_y
        y_min = jsoFloor(y0/dopeblk_grid)*dopeblk_grid - grid_offset_y
      )
      ("180")
        x_min = jsoFloor(x0/dopeblk_grid)*dopeblk_grid - dopeblk_grid - grid_offset_x
        x_max = jsoCeiling(x0/dopeblk_grid)*dopeblk_grid - grid_offset_x
        y_max = jsoRound((y0+effective_radius+effective_width/2)/dopeblk_grid)*dopeblk_grid - grid_offset_y
        y_min = jsoRound((y0+effective_radius-effective_width/2)/dopeblk_grid)*dopeblk_grid - grid_offset_y
      )
    )
    BlockWriter(list(x_min:y_min x_max:y_max) dopeblock_layer_name)
  ) ;end of body of code
) ;end of pcell definition
```

### *10.4.4.3. wg_ring.il*

```
pcDefinePCell(
  list( ddGetObj("genericPhotonicsLib") "wg_ring" "layout")
  ( ; parameters and their optional defaults:
    (width_mode "FIXED") ; or "SINGLEMODE"
    (thickness "200")
    (wavelength "1550")
    (layer "POLY")
    (width 0.5)
    (radius 6.0)
    (pc_x_origin 0.0)
    (pc_y_origin 0.0)
    (pc_rotation "R0")
  ) ; end of parameter list
```

```
; geometry creation body of code:
let( (
    x0 y0 coords coordList mask_grid dopeblk_grid fillblk_grid arb_layer_grid dopeblock_layer_name
    dopeblk_overhang_inner dopeblk_overhang_outer fillblk_overhang_inner fillblk_overhang_outer arc_master
    arc_inst_1 arc_inst_2 arc_inst_3 arc_inst_4 effective_radius effective_width biased_width
  )
  if( (strcmp(width_mode "SINGLEMODE") == 0) then
    width = GetSingleModeWidth(thickness wavelength)
  )
  biased_width = PolyMaskSize(width)
  mask_grid = GetGlobal("mask_grid")
  dopeblk_grid = GetGlobal("dopeblk_grid")
  fillblk_grid = GetGlobal("fillblk_grid")
  dopeblk_overhang_inner = GetGlobal("dopeblk_overhang")
  fillblk_overhang_inner = GetGlobal("fillblk_overhang")
  dopeblk_overhang_outer = GetGlobal("dopeblk_overhang")
  fillblk_overhang_outer = GetGlobal("fillblk_overhang")
  ; calculate the ring center
  x0 = 0
  y0 = 0
  case( layer
    ( ("POLY" "BODY")
      arc_master = dbOpenCellViewByType( ddGetObj("genericPhotonicsLib") "arc" "layout")
      arc_inst_1 = dbCreateParamInst( pcCellView arc_master nil x0:y0-radius "R0" 1
        list(
          list("layer" "string" layer)
          list("width" "float" width)
          list("radius" "float" radius)
        )
      )
      arc_inst_2 = dbCreateParamInst( pcCellView arc_master nil x0:y0-radius "MY" 1
        list(
          list("layer" "string" layer)
          list("width" "float" width)
          list("radius" "float" radius)
        )
      )
      arc_inst_3 = dbCreateParamInst( pcCellView arc_master nil x0:y0+radius "MX" 1
        list(
          list("layer" "string" layer)
          list("width" "float" width)
          list("radius" "float" radius)
        )
      )
      arc_inst_4 = dbCreateParamInst( pcCellView arc_master nil x0:y0+radius "R180" 1
        list(
```

```
            list("layer" "string" layer)
            list("width" "float" width)
            list("radius" "float" radius)
          )
        )

        effective_radius = radius + (dopeblk_overhang_outer-dopeblk_overhang_inner)/2
        effective_width = dopeblk_overhang_outer+biased_width+dopeblk_overhang_inner
        coordList = uwlRingRectanglesNoMirror( x0 y0 pc_x_origin pc_y_origin pc_rotation effective_radius
effective_width dopeblk_grid mask_grid)
        foreach( coords coordList
          BlockWriter(coords dopeblock_layer_name )
        ) ;end of rectangle creation loop

        effective_radius = radius + (fillblk_overhang_outer-fillblk_overhang_inner)/2
        effective_width = fillblk_overhang_outer+biased_width+fillblk_overhang_inner
        coordList = uwlRingRectanglesNoMirror( x0 y0 pc_x_origin pc_y_origin pc_rotation effective_radius
effective_width fillblk_grid mask_grid)
        foreach( coords coordList
          BlockWriter(coords "FILLBLOCK")
        ) ;end of rectangle creation loop
      )
      ( t
        case( layer
          ( ("POLY_NO_CLAD")
            arb_layer_grid = mask_grid
            layer = "POLY"
          )
          ( ("FILLBLOCK")
            arb_layer_grid = GetGlobal("fillblk_grid")
          )
          ( ("DOPEBLOCK")
            arb_layer_grid = GetGlobal("dopeblk_grid")
          )
          ( t
            arb_layer_grid = mask_grid
          )
        )
        coordList = uwlRingRectanglesNoMirror( x0 y0 pc_x_origin pc_y_origin pc_rotation radius biased_width
arb_layer_grid mask_grid)
        foreach( coords coordList
          BlockWriter(coords layer)
        ) ;end of rectangle creation loop
      )
    )
  ) ;end of body of code
```

```
) ;end of pcell definition
```

### *10.4.4.4. wg_offset.il*

```
pcDefinePCell(
  list( ddGetObj("genericPhotonicsLib") "wg_offset" "layout")
  ; parameters and their optional defaults:
  (
    (width_mode "FIXED") ; or "SINGLEMODE"
    (thickness "200")
    (wavelength "1550")
    (layer "POLY")
    (width 0.370)
    (delta_x 3.00)
    (delta_y 1.00)
    (pc_x_origin 0.0)
    (pc_y_origin 0.0)
    (pc_rotation "R0")
  ) ;end of parameter list
  ; geometry creation body of code:
  let( (
      coords coordList x0 y0 x_a y_a x_b y_b biased_width flip mask_grid dopeblk_grid fillblk_grid
      dopeblk_overhang fillblk_overhang dopeblock_layer_name offset_master inst
    )
    flip = nil
    if( (strcmp(width_mode "SINGLEMODE") == 0) then
      width = GetSingleModeWidth(thickness wavelength)
    )
    biased_width = PolyMaskSize(width)
    mask_grid = GetGlobal("mask_grid")
    dopeblk_grid = GetGlobal("dopeblk_grid")
    fillblk_grid = GetGlobal("fillblk_grid")
    dopeblk_overhang = GetGlobal("dopeblk_overhang")
    fillblk_overhang = GetGlobal("fillblk_overhang")
    ; arc origins
    x0 = 0
    y0 = 0
    offset_master = dbOpenCellViewByType( ddGetObj("genericPhotonicsLib") "offset" "layout")
    inst = dbCreateParamInst( pcCellView offset_master nil x0:y0 "R0" 1
      list(
        list("layer" "string" layer)
        list("width" "float" width)
        list("delta_x" "float" delta_x)
        list("delta_y" "float" delta_y)
      )
    )
```

```
    if( (delta_y < 0.0) then
      delta_y = 0.0-delta_y
      flip = 1
    )
    coordList = uwlSinusoid( x0-dopeblk_grid y0 pc_x_origin pc_y_origin pc_rotation biased_width+2*dopeblk_overhang
delta_x+2*dopeblk_grid delta_y dopeblk_grid mask_grid)
    foreach( coords coordList
      if( flip then
                x_a = caar(coords)
                y_a = -cadar(coords)
                x_b = caadr(coords)
                y_b = -cadadr(coords)
                coords = list( x_a:y_a x_b:y_b )
      )
      BlockWriter(coords dopeblock_layer_name )
    ) ;end of rectangle creation loop

    coordList = uwlSinusoid( x0-fillblk_grid y0 pc_x_origin pc_y_origin pc_rotation biased_width+2*fillblk_overhang
delta_x+2*fillblk_grid delta_y fillblk_grid mask_grid)
    foreach( coords coordList
      if( flip then
                x_a = caar(coords)
                y_a = -cadar(coords)
                x_b = caadr(coords)
                y_b = -cadadr(coords)
                coords = list( x_a:y_a x_b:y_b )
          )
      BlockWriter(coords "FILLBLOCK")
    ) ;end of rectangle creation loop
  ) ;end of body of code
) ;end of pcell definition
```

### *10.4.4.5. wg_taper.il*

```
pcDefinePCell(
  list( ddGetObj("genericPhotonicsLib") "wg_taper" "layout")
  ; parameters and their optional defaults:
  (
    (start_width_mode     "FIXED") ; or "SINGLEMODE"
    (end_width_mode       "FIXED") ; or "SINGLEMODE"
    (thickness            "200")
    (wavelength           "1550")
    (layer          "POLY")
    (start_width  0.35)
    (end_width    1.00)
    (length     10.00)
```

```
    (pc_x_origin     0.0)
    (pc_y_origin     0.0)
    (pc_rotation     "R0")
  ) ;end of parameter list
  ; geometry creation body of code:
  let( (
       x0 y0 coords coordList taper_master taper_inst biased_start_width biased_end_width
       mask_grid dopeblk_grid fillblk_grid dopeblk_overhang fillblk_overhang dopeblock_layer_name
    )
    if( (strcmp(start_width_mode "SINGLEMODE") == 0) then
       start_width = GetSingleModeWidth(thickness wavelength)
    )
    if( (strcmp(end_width_mode "SINGLEMODE") == 0) then
       end_width = GetSingleModeWidth(thickness wavelength)
    )
    biased_start_width = PolyMaskSize(start_width)
    biased_end_width = PolyMaskSize(end_width)
    mask_grid = GetGlobal("mask_grid")
    dopeblk_grid = GetGlobal("dopeblk_grid")
    fillblk_grid = GetGlobal("fillblk_grid")
    dopeblk_overhang = GetGlobal("dopeblk_overhang")
    fillblk_overhang = GetGlobal("fillblk_overhang")
    x0 = 0.0
    y0 = 0.0
    taper_master = dbOpenCellViewByType( ddGetObj("genericPhotonicsLib") "taper" "layout")
        taper_inst = dbCreateParamInst( pcCellView taper_master nil x0:y0 "R0" 1
            list(
                list("layer" "string" layer)
         list("shape" "string" "LINEAR")
                list("start_width" "float" start_width)
                list("end_width" "float" end_width)
                list("length" "float" length)
            )
        )
    ; make the block layers
    coordList = uwlTaperRectangles( x0 y0 pc_x_origin pc_y_origin pc_rotation (biased_start_width+dopeblk_overhang*2)
(biased_end_width+dopeblk_overhang*2) length dopeblk_grid mask_grid)
    foreach( coords coordList
      BlockWriter(coords dopeblock_layer_name )
    ) ;end of rectangle creation loop
    ; make the block layers
    coordList = uwlTaperRectangles( x0 y0 pc_x_origin pc_y_origin pc_rotation (biased_start_width+fillblk_overhang*2)
(biased_end_width+fillblk_overhang*2) length fillblk_grid mask_grid)
    foreach( coords coordList
      BlockWriter(coords "FILLBLOCK")
    ) ;end of rectangle creation loop
```

```
  ) ;end of body of code
) ;end of pcell definition
```

### *10.4.4.6. wg_coupler.il*

```
pcDefinePCell(
  list( ddGetObj("genericPhotonicsLib") "wg_coupler" "layout")
  ; parameters and their optional defaults:
  (
  (width_mode          "FIXED") ; or "SINGLEMODE"
  (thickness           "200")
  (wavelength          "1550")
  (layer         "POLY")
  (width         0.320)
  (variant       "STANDARD")
  (pc_x_origin     0.0)
  (pc_y_origin     0.0)
  (pc_rotation     "R0")
  (top_fill_block      "YES")
  ) ;end of parameter list
  ; geometry creation body of code:
  let(  (
        coords coordList i x1 y1 x2 y2 x0 y0 x_min x_max y_min y_max taper_master taper_inst coupler_fill_block_master
        coupler_fill_block_inst ar_tooth ar_tooth_width uniform grating_vector split taper_type design_wavelength
        taper_length grating_width grating_input_length grating_gap_repeat grating_bar_repeat grating_num_repeat
        poly_input_offset poly_gap_repeat poly_bar_repeat poly_num_repeat mask_grid dopeblk_grid
        grid_offset_x grid_offset_y fillblk_grid dopeblk_overhang fillblk_overhang
    )
  ; Correct parameters for physical dimensions
  if( (strcmp(width_mode "SINGLEMODE") == 0) then
     width = GetSingleModeWidth(thickness wavelength)
  )
  mask_grid = GetGlobal("mask_grid")
  dopeblk_grid = GetGlobal("dopeblk_grid")
  fillblk_grid = GetGlobal("fillblk_grid")
  dopeblk_overhang = GetGlobal("dopeblk_overhang")
  fillblk_overhang = GetGlobal("fillblk_overhang")
  ; Grating Definition
  taper_length = 5.000
  grating_input_length = 0.100
  design_wavelength = 1550
  taper_type = "NONLIN_1550"
  ar_tooth = 0
  ar_tooth_width = 0.410
  grating_width = 12.0
  grating_gap_repeat =  PolyMaskGap(0.5 0.5 0.5)
```

```
grating_bar_repeat =  PolyMaskSize(0.5)
grating_num_repeat = 10
uniform = 1
split = 0
grating_vector = list()
case( variant
  ( ("STANDARD")
    case( wavelength
      ( ("1550")
        taper_length = 200.000
        grating_input_length = 0.100
        design_wavelength = 1550

        uniform = 1
        case( thickness
          ( "100"
            grating_width = 12.0
            taper_type = "LINEAR"
            ar_tooth =0
            grating_gap_repeat =  0.220
            grating_bar_repeat =  0.650
            grating_num_repeat = 18

          )
          ( "200"
            grating_width = 12.0
            taper_type = "LINEAR"
            ar_tooth =0
            grating_gap_repeat =  0.220
            grating_bar_repeat =  0.510
            grating_num_repeat = 22
          )
        )
      ) ; End 1550 nm Section
      ( ("1250" "1280" "1220")
        taper_length = 200.000
        grating_input_length = 0.100
        design_wavelength = 1200
        uniform = 1
        case( thickness
          ( "100"
            grating_width = 12.0
            taper_type = "LINEAR"
            ar_tooth = 0
            grating_gap_repeat =  0.220
            grating_bar_repeat =  0.470
```

```
                grating_num_repeat = 23
              )
              ( "200"
                grating_width = 12.0
                taper_type = "LINEAR"
                ar_tooth =0
                grating_gap_repeat = 0.220
                grating_bar_repeat = 0.380
                grating_num_repeat = 27
              )
            )
          ) ; End 1250 nm Section
        ) ; End Wavelength Case Statement
      ) ; End Standard Section (no nonuniform or special gratings included)
    ) ; End Variant Case Statement
    y0 = 0
      x0 = 0
      taper_master = dbOpenCellViewByType( ddGetObj("genericPhotonicsLib") "taper" "layout")
      taper_inst = dbCreateParamInst( pcCellView taper_master nil x0:y0 "R0" 1
      list(
        list("layer" "string" layer)
        list("start_width" "float" width)
        list( "end_width" "float" grating_width)
        list( "length" "float" taper_length )
        list( "shape" "string" taper_type )
      ) ; close list
    ); close inst

    ; make the block layers
    coordList = uwlTaperRectangles( x0 y0 pc_x_origin pc_y_origin pc_rotation (width+dopeblk_overhang*2)
(grating_width+dopeblk_overhang*2) taper_length design_wavelength dopeblk_grid mask_grid)
    foreach( coords coordList
      BlockWriter(coords "DOPEBLOCK")
    ) ;end of rectangle creation loop
    coordList = uwlTaperRectangles( x0 y0 pc_x_origin pc_y_origin pc_rotation (width+fillblk_overhang*2)
(grating_width+fillblk_overhang*2) taper_length design_wavelength fillblk_grid mask_grid)
    foreach( coords coordList
      BlockWriter(coords "FILLBLOCK")
    ) ;end of rectangle creation loop

    x0 = x0
    y0 = y0+taper_length
      x_min = x0 - jsoRound(grating_width/2/mask_grid)*mask_grid
      x_max = x0 + jsoRound(grating_width/2/mask_grid)*mask_grid
      y_min = y0
      y_max = y_min + grating_input_length
```

```
    BlockWriter(list( x_min:y_min x_max:y_max ) layer )
  if( (ar_tooth == 1) then
    y_min = y_max + grating_gap_repeat
    y_max = y_min + ar_tooth_width
    BlockWriter(list( x_min:y_min x_max:y_max ) layer )
  )
    for( i 1 grating_num_repeat
      if( (uniform == 1) then
        y_min = y_max + grating_gap_repeat
        y_max = y_min + grating_bar_repeat
      else
        grating_gap_repeat = car(grating_vector)
        grating_bar_repeat = cadr(grating_vector)
        grating_vector = cddr(grating_vector)
          y_min = y_max + grating_gap_repeat
          y_max = y_min + grating_bar_repeat
      )
      ; Split functionality in processes where maximum moscap width becomes a DRC violation
      case( split
        ( 0
          BlockWriter( list( x_min:y_min x_max:y_max ) layer )
        )
        ( 1
          case( layer
            ( "POLY"
              BlockWriter( list( x_min:y_min -0.061:y_max ) layer )
              BlockWriter( list( 0.061:y_min x_max:y_max ) layer )
            )
            ( "BODY"
              BlockWriter( list( x_min:y_min x_max:y_max ) layer )
            )
          )
        )
        ( 2
          case( layer
            ( "POLY"
              BlockWriter( list( x_min:y_min -2.561:y_max ) layer )
              BlockWriter( list( -2.439:y_min 2.439:y_max ) layer )
              BlockWriter( list( 2.561:y_min x_max:y_max ) layer )
            )
            ( "BODY"
              BlockWriter( list( x_min:y_min x_max:y_max ) layer )
            )
          )
        )
        ( 5
```

```
            case( layer
              ( "POLY"
                BlockWriter( list( 4.866:y_min x_max:y_max ) layer )
                BlockWriter( list( 2.944:y_min 4.744:y_max ) layer )
                BlockWriter( list( 1.022:y_min 2.822:y_max ) layer )
                BlockWriter( list( -0.900:y_min 0.900:y_max ) layer )
                BlockWriter( list( -1.022:y_min -2.822:y_max ) layer )
                BlockWriter( list( -2.944:y_min -4.744:y_max ) layer )
                BlockWriter( list( -4.866:y_min -x_max:y_max ) layer )
              )
              ( "BODY"
                BlockWriter( list( x_min:y_min x_max:y_max ) layer )
              )
            )
          )
        )
      )
    x1 = x_min
    x2 = x_max
    y1 = y_max
    grid_offset_x = CalcOffsetX(pc_x_origin pc_y_origin pc_rotation dopeblk_grid mask_grid)
    grid_offset_y = CalcOffsetY(pc_x_origin pc_y_origin pc_rotation dopeblk_grid mask_grid)
      x_min = jsoCeiling((x1 - dopeblk_overhang)/dopeblk_grid)*dopeblk_grid-grid_offset_x
      x_max = jsoFloor((x2 + dopeblk_overhang)/dopeblk_grid)*dopeblk_grid-grid_offset_x
      y_min = jsoRound(y0/dopeblk_grid)*dopeblk_grid-grid_offset_y
      y_max = jsoRound((y1 + dopeblk_overhang)/dopeblk_grid)*dopeblk_grid-grid_offset_y
      BlockWriter( list( x_min:y_min x_max:y_max ) "DOPEBLOCK" )

    grid_offset_x = CalcOffsetX(pc_x_origin pc_y_origin pc_rotation fillblk_grid mask_grid)
    grid_offset_y = CalcOffsetY(pc_x_origin pc_y_origin pc_rotation fillblk_grid mask_grid)
      x_min = jsoRound((x1 - fillblk_overhang)/fillblk_grid)*fillblk_grid-grid_offset_x
      x_max = jsoRound((x2 + fillblk_overhang)/fillblk_grid)*fillblk_grid-grid_offset_x
      y_min = jsoRound(y0/fillblk_grid)*fillblk_grid-grid_offset_y
      y_max = jsoRound((y1 + fillblk_overhang)/fillblk_grid)*fillblk_grid-grid_offset_y
      BlockWriter( list( x_min:y_min x_max:y_max ) "FILLBLOCK" )
  ) ;end of body of code
) ;end of pcell definition
```

## 10.4.5. Simplified Top-Level User-Environment Cell Example

### *10.4.5.1. resonator_cell.il*

```
pcDefinePCell(
list( ddGetObj("genericPhotonicsLib") "resonator_cell" "layout")
( ; parameters and their optional defaults:
  (lead_width_mode "FIXED") ; or "SINGLEMODE"
```

```
  (ring_width_mode          "FIXED") ; or "SINGLEMODE"
  (thickness            "200")
  (wavelength           "1550")
  (layer          "POLY")
  (resonator_type       "SINGLE") ; DOUBLE RACETRACK CUSTOM
  (continuity       "ISOLATED") ; INPUT OUTPUT THROUGH THROUGH_PD_THROUGH THROUGH_PD_DROP OUTPUT_PD_DROP
ISOLATED_PD_DROP INPUT_PD_DROP ISOLATED_PD_THROUGH
  (coupler_variant   "STANDARD")
  (flip_input_lead "NO") ; YES
  (bus_wg_width     0.5)
  (taper_bus       "NO")
  (bus_taper_length       2.0)
  (lead_wg_width     0.464)
  (ring_wg_width     0.5)
  (through_coupling_gap   0.3)
  (drop_coupling_gap     0.3)
  (ring_ring_coupling_gap   0.3)
  (intracell_wg_pitch   4.0)
  (wg_offset_length      10.0)
  (ring_radius      5.0)
  (bend_radius       10.0)
  (resonator_horizontal_alignment  "CENTERED")
  (resonator_vertical_alignment "CENTERED")
  (ring_center_offset   50.0)
  (input_through_offset   100.0)
  (input_drop_offset    100.0)
  (alternate_internal_pitch  "NO")
  (through_drop_pitch 40.0)
  (internal_pitch      30.0)
  (coupler_bus_pitch   0.0)
  (pc_x_origin      0.0)
  (pc_y_origin      0.0)
  (pc_rotation      "R0")
  (drop_coupling_mode       "SAME_AS_THROUGH")
  (flip_couplers       "NO") ; YES
  (flip_through_coupler_only   "NO")
) ; end of parameter list
; geometry creation body of code:
let( (
      i x0 y0 x1 y1 x2 y2 x3 y3 x4 y4 inst ring_master wg_master wg_bend_master wg_coupler_master wg_offset_master
      wg_adiabatic_bend_master wg_adiabatic_narrowed_bend_master heater_master fill_cell_master
      bus_wg_length wg_taper_master wg_taper_inst rad_taper_length rad_min_width rad_start_width rad_bend_radius
      rad_wg_length bus_wg_length drop_lead_length through_lead_length input_lead_length mask_grid bus_pitch
      arm_length arm_pitch modulator_arm_master modulator_ring_master mod_ring_coupling_string mod_type_string
      truncated_bus_wg_length no_drop_wg Rmax Rmin bend_extension N opcblockall opcblockring customopcring
      opcblockextend ps_height rotation sub_pc_x_origin sub_pc_y_origin sub_rotation
```

```
  )
  mask_grid  = GetGlobal("mask_grid")
  if( (strcmp(lead_width_mode "SINGLEMODE") == 0) then
    lead_wg_width  = GetSingleModeWidth(thickness  wavelength)
  )
  if( (strcmp(ring_width_mode "SINGLEMODE") == 0) then
    ring_wg_width  = GetSingleModeWidth(thickness  wavelength)
  )
  if( (strcmp(taper_bus  "NO") == 0) then
    bus_wg_width  = lead_wg_width
  )
  if( (strcmp(alternate_internal_pitch  "NO") == 0) then
    internal_pitch  = through_drop_pitch
  )
  if( (strcmp(drop_coupling_mode  "SAME_AS_THROUGH") == 0) then
    drop_coupling_gap  = through_coupling_gap
    no_drop_wg  = "NO"
  else
    if( (strcmp(drop_coupling_mode  "NO_DROP_WG") == 0) then
      drop_coupling_gap  = through_coupling_gap
      no_drop_wg  = "YES"
    else
      no_drop_wg  = "NO"
    )
  )
  if( (strcmp(resonator_horizontal_alignment  "CENTERED_TO_THROUGH") == 0) then
    ring_center_offset  = jsoRound(input_through_offset/2/mask_grid)*mask_grid
  else
    if( (strcmp(resonator_horizontal_alignment  "CENTERED_TO_DROP") == 0) then
      ring_center_offset  = jsoRound(input_drop_offset/2/mask_grid)*mask_grid
    )
  )
  if( (strcmp(flip_through_coupler_only  "YES") == 0) then
    flip_couplers  = "NO"
  )
  arm_length  = 0.0
  bend_extension  = 0.0
  if( (strcmp("SINGLE"  resonator_type) == 0) then
    bus_pitch  = 2*ring_radius+bus_wg_width+ring_wg_width+through_coupling_gap+drop_coupling_gap
  else
    if( (strcmp("DOUBLE"  resonator_type) == 0) then
      bus_pitch  =
4*ring_radius+bus_wg_width+2*ring_wg_width+through_coupling_gap+drop_coupling_gap+ring_ring_coupling_gap
    )
  )
```

```
  drop_lead_length = input_drop_offset - input_lead_length - bus_pitch - 2*bend_radius
  through_lead_length = input_through_offset - input_lead_length - 2*bend_radius
  bus_wg_length = internal_pitch - 2*bend_radius
  if( ((strcmp("SINGLE" resonator_type) == 0) || (strcmp("RACETRACK" resonator_type) == 0) || (strcmp("CUSTOM"
resonator_type) == 0)) then
    output_bus_wg_length = internal_pitch - 2*bend_radius
  else
    if( (strcmp("DOUBLE" resonator_type) == 0) then
      output_bus_wg_length = internal_pitch - 2*bend_radius - intracell_wg_pitch
    else
      output_bus_wg_length = 0.0
      ;print("I had a problem identifying resonator_type")
    )
  )
  rad_taper_length = 2.0
  rad_min_width = 0.205
  rad_bend_radius = 5.0
  rad_wg_length = 0.1
  ; pcell masters
  coupler_master = dbOpenCellViewByType( ddGetObj("genericPhotonicsLib") "wg_coupler" "layout")
  wg_bend_master = dbOpenCellViewByType( ddGetObj("genericPhotonicsLib") "wg_bend" "layout")
  wg_master = dbOpenCellViewByType( ddGetObj("genericPhotonicsLib") "wg" "layout")
  ring_master = dbOpenCellViewByType( ddGetObj("genericPhotonicsLib") "wg_ring" "layout")
  wg_taper_master = dbOpenCellViewByType( ddGetObj("genericPhotonicsLib") "wg_taper" "layout")
  wg_offset_master= dbOpenCellViewByType( ddGetObj("genericPhotonicsLib") "wg_offset" "layout")
  x0 = 0.0
  y0 = 0.0
  x1 = 0.0
  y1 = 0.0
  x2 = 0.0
  y2 = 0.0
  x3 = 0.0
  y3 = 0.0
  x4 = 0.0
  y4 = 0.0
  if( (strcmp("YES" flip_input_lead) == 0) then
    ; make the 90 bend down region
    y0 = internal_pitch
    rotation = "MX"
    dbCreateParamInst( pcCellView wg_bend_master nil x0:y0 rotation 1
      list(
        list("width" "float" lead_wg_width)
        list("radius" "float" bend_radius )
        list("angle" "string" "90")
        list("layer" "string" layer)
        list("width_mode" "string" "FIXED")
```

```
        list("thickness" "string" thickness)
        list("wavelength" "string" wavelength)
        list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
        list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
        list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
    ) ; close list
); close inst
x0 = x0 + bend_radius
y0 = y0 - bend_radius
rotation = "R270"
dbCreateParamInst( pcCellView wg_master nil x0:y0 rotation 1
    list(
        list("width" "float" lead_wg_width)
        list("length" "float" bus_wg_length)
        list("continuity" "string" "ISOLATED")
        list("layer" "string" layer)
        list("width_mode" "string" "FIXED")
        list("thickness" "string" thickness)
        list("wavelength" "string" wavelength)
        list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
        list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
        list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
    )
)
x0 = x0
y0 = bend_radius
rotation = "R270"
dbCreateParamInst( pcCellView wg_bend_master nil x0:y0 rotation 1
    list(
        list("width" "float" lead_wg_width)
        list("radius" "float" bend_radius )
        list("angle" "string" "90")
        list("layer" "string" layer)
        list("width_mode" "string" "FIXED")
        list("thickness" "string" thickness)
        list("wavelength" "string" wavelength)
        list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
        list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
        list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
    ) ; close list
); close inst
x0 = x0 + bend_radius
y0 = 0.0
input_lead_length = input_lead_length - x0
x4 = 0.0
y4 = internal_pitch
```

```
else
  if(   ( (strcmp("ISOLATED" continuity) == 0)    ||
        (strcmp("ISOLATED_PD_THROUGH" continuity) == 0) ||
        (strcmp("INPUT" continuity) == 0)       ) then
    if( coupler_bus_pitch > 0.001 then
      y0 = coupler_bus_pitch
      rotation = "MX"
      dbCreateParamInst( pcCellView wg_offset_master nil x0:y0 rotation 1
        list(
          list("width" "float" lead_wg_width)
          list("delta_x" "float" wg_offset_length)
          list("delta_y" "float" coupler_bus_pitch)
          list("width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("layer" "string" layer)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
          list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
        ) ; close list
      ); close inst
      x0 = x0 + wg_offset_length
      y0 = 0.0
      input_lead_length = input_lead_length - x0
      x4 = 0.0
      y4 = coupler_bus_pitch
    )
  )
)
if(   ( (strcmp("ISOLATED" continuity) == 0)    ||
    (strcmp("ISOLATED_PD_THROUGH" continuity) == 0) ||
    (strcmp("INPUT" continuity) == 0)       ) then
  rotation = "R180"
  dbCreateParamInst( pcCellView wg_master nil x4:y4 rotation 1
    list(
      list("width" "float" lead_wg_width)
      list("length" "float" 1.0)
      list("continuity" "string" "THROUGH")
      list("layer" "string" layer)
      list("width_mode" "string" "FIXED")
      list("thickness" "string" thickness)
      list("wavelength" "string" wavelength)
      list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x4 y4))
      list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x4 y4))
      list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
    )
```

```
    )
    x4 = x4-1.0
    ; input coupler
    if( (strcmp("YES" flip_couplers) == 0) then
      rotation = "R90"
    else
      rotation = "MYR90"
    )
    inst = dbCreateParamInst( pcCellView coupler_master nil x4:y4 rotation 1
      list(
        list("width" "float" lead_wg_width)
        list("variant" "string" coupler_variant)
        list("layer" "string" layer)
        list("width_mode" "string" "FIXED")
        list("thickness" "string" thickness)
        list("wavelength" "string" wavelength)
        list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x4 y4))
        list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x4 y4))
        list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
      )
    )
)
; wg from input to up 90 bend
rotation = "R0"
dbCreateParamInst( pcCellView wg_master nil x0:y0 rotation 1
  list(
    list("width" "float" lead_wg_width)
    list("length" "float" input_lead_length)
    list("continuity" "string" "THROUGH")
    list("layer" "string" layer)
    list("width_mode" "string" "FIXED")
    list("thickness" "string" thickness)
    list("wavelength" "string" wavelength)
    list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
    list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
    list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
  )
)
  ; make the up 90 bend to the "through" coupling region
  x0 = x0+input_lead_length
  y0 = 0.0
  rotation = "R0"
  sub_pc_x_origin = CalcSubOriginX(pc_rotation pc_x_origin x0 y0)
  sub_pc_y_origin = CalcSubOriginY(pc_rotation pc_y_origin x0 y0)
  sub_rotation = CalcSubRotation(pc_rotation rotation)
  dbCreateParamInst( pcCellView wg_bend_master nil x0:y0 rotation 1
```

```
        list(
          list("width" "float" lead_wg_width)
          list("radius" "float" bend_radius )
          list("angle" "string" "90")
          list("layer" "string" layer)
          list("width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
          list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
        ) ; close list
      ); close inst
    ; make the ring(s)
    if( (strcmp("SINGLE" resonator_type) == 0) then
      if( (strcmp("YES" phase_shifter) == 0) then
        x1 = x0 + bend_radius + jsoRound(bus_pitch/2/mask_grid)*mask_grid
      else
        x1 = x0 + bend_radius + jsoRound((bus_wg_width/2 + ring_wg_width/2 + through_coupling_gap +
ring_radius)/mask_grid)*mask_grid
      )
      case( resonator_vertical_alignment
        ( "CENTERED"
          y1 = bend_radius + jsoRound(bus_wg_length/2/mask_grid)*mask_grid
        )
        ( "ALIGN_TO_BOTTOM"
          y1 = ring_radius + jsoRound(ring_wg_width/2/mask_grid)*mask_grid
        )
        ( "EQUALIZE_OUTPUT_LENGTHS"
          y1 = bend_radius + jsoRound((bus_wg_length + bus_pitch)/2/mask_grid)*mask_grid
        )
        ( t
          y1 = bend_radius + jsoRound(bus_wg_length/2/mask_grid)*mask_grid
        )
      )
      rotation = "R90"
      dbCreateParamInst( pcCellView ring_master nil x1:y1 rotation 1
        list(
          list("width" "float" ring_wg_width)
          list("radius" "float" ring_radius)
          list("width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("layer" "string" layer)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x1 y1))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x1 y1))
```

```
            list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
          )
        )
      else
        if( (strcmp("DOUBLE" resonator_type) == 0) then
          xl = x0 + bend_radius + jsoRound((bus_wg_width/2 + ring_wg_width/2 + through_coupling_gap +
ring_radius)/mask_grid)*mask_grid
          case( resonator_vertical_alignment
            ( "CENTERED"
              yl = bend_radius + jsoRound(bus_wg_length/2/mask_grid)*mask_grid
            )
            ( "ALIGN_TO_BOTTOM"
              yl = ring_radius + jsoRound(ring_wg_width/2/mask_grid)*mask_grid
            )
            ( t
              yl = bend_radius + jsoRound(bus_wg_length/2/mask_grid)*mask_grid
            )
          )
          rotation = "R90"
          dbCreateParamInst( pcCellView ring_master nil xl:yl rotation 1
            list(
              list("width" "float" ring_wg_width)
              list("radius" "float" ring_radius)
              list("width_mode" "string" "FIXED")
              list("thickness" "string" thickness)
              list("wavelength" "string" wavelength)
              list("layer" "string" layer)
              list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin xl yl))
              list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin xl yl))
              list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
            )
          )
          xl = xl + 2*ring_radius + ring_wg_width + ring_ring_coupling_gap
          yl = yl
          rotation = "R90"
          dbCreateParamInst( pcCellView ring_master nil xl:yl rotation 1
            list(
              list("width" "float" ring_wg_width)
              list("radius" "float" ring_radius)
              list("width_mode" "string" "FIXED")
              list("thickness" "string" thickness)
              list("wavelength" "string" wavelength)
              list("layer" "string" layer)
              list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin xl yl))
              list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin xl yl))
              list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
```

```
        )
      )
    )
  )
    if( (strcmp("YES" taper_bus) == 0) then
      ; input_taper
      x0 = x0 + bend_radius
      y0 = bend_radius
      rotation = "R0"
      dbCreateParamInst( pcCellView wg_taper_master nil x0:y0 rotation 1
        list(
          list("length" "float" bus_taper_length)
          list("start_width" "float" lead_wg_width)
          list("end_width" "float" bus_wg_width)
          list("start_width_mode" "string" "FIXED")
          list("end_width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("layer" "string" layer)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
          list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
        )
      )
      truncated_bus_wg_length = jsoRound((bus_wg_length-2*bus_taper_length)/mask_grid)*mask_grid
      ; wg from input up 90 bend and serves as the input bus waveguide
      x0 = x0
      y0 = y0 + bus_taper_length
      rotation = "R90"
      dbCreateParamInst( pcCellView wg_master nil x0:y0 rotation 1
        list(
          list("width" "float" bus_wg_width)
          list("length" "float" truncated_bus_wg_length)
          list("continuity" "string" "ISOLATED")
          list("layer" "string" layer)
          list("width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
          list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
        )
      )
      ; output_taper
      x0 = x0
      y0 = y0 + truncated_bus_wg_length
```

```
    rotation = "R0"
    dbCreateParamInst( pcCellView wg_taper_master nil x0:y0 rotation 1
      list(
        list("length" "float" bus_taper_length)
        list("start_width" "float" bus_wg_width)
        list("end_width" "float" lead_wg_width)
        list("start_width_mode" "string" "FIXED")
        list("end_width_mode" "string" "FIXED")
        list("thickness" "string" thickness)
        list("wavelength" "string" wavelength)
        list("layer" "string" layer)
        list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
        list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
        list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
      )
    )
    x0 = x0
    y0 = y0 + bus_taper_length
  else
        y0 = 2*ring_radius
        x0 = x0
        rotation = "R180"
        dbCreateParamInst( pcCellView wg_bend_master nil x0:y0 rotation 1
          list(
            list("width" "float" lead_wg_width)
            list("radius" "float" bend_radius )
            list("angle" "string" "90")
            list("layer" "string" layer)
            list("width_mode" "string" "FIXED")
            list("thickness" "string" thickness)
            list("wavelength" "string" wavelength)
            list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
            list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
            list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
          ) ; close list
        ); close inst
        y0 = y0-bend_radius
        x0 = x0-bend_radius
        rotation = "MYR90"
        dbCreateParamInst( pcCellView wg_bend_master nil x0:y0 rotation 1
          list(
            list("width" "float" lead_wg_width)
            list("radius" "float" bend_radius )
            list("angle" "string" "180")
            list("layer" "string" layer)
            list("width_mode" "string" "FIXED")
```

```
                list("thickness" "string" thickness)
                list("wavelength" "string" wavelength)
                list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
                list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
                list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
              ) ; close list
            ); close inst
            x0 = x0 - 2*bend_radius
            y0 = y0
            rotation = "R90"
            dbCreateParamInst( pcCellView wg_master nil x0:y0 rotation 1
              list(
                list("width" "float" lead_wg_width)
                list("length" "float" internal_pitch-bend_radius-y0)
                list("continuity" "string" "ISOLATED")
                list("layer" "string" layer)
                list("width_mode" "string" "FIXED")
                list("thickness" "string" thickness)
                list("wavelength" "string" wavelength)
                list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
                list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
                list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
              )
            )
            x0 = x0
            y0 = internal_pitch-bend_radius
  )
; block out drop coupler and radiation taper stuff
if( (strcmp("NO" no_drop_wg) == 0) then
    if( (strcmp("YES" taper_bus) == 0) then
      ; lower_taper
      x3 = x0 + bus_pitch
      y3 = bend_radius
      rotation = "R0"
      dbCreateParamInst( pcCellView wg_taper_master nil x3:y3 rotation 1
        list(
          list("length" "float" bus_taper_length)
          list("start_width" "float" lead_wg_width)
          list("end_width" "float" bus_wg_width)
          list("start_width_mode" "string" "FIXED")
          list("end_width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("layer" "string" layer)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
```

```
            list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
          )
        )
        truncated_bus_wg_length = jsoRound((output_bus_wg_length-bus_taper_length)/mask_grid)*mask_grid
        ; wg from input up 90 bend and serves as the input bus waveguide
        x3 = x3
        y3 = y3 + bus_taper_length
        rotation = "R90"
        dbCreateParamInst( pcCellView wg_master nil x3:y3 rotation 1
          list(
            list("width" "float" bus_wg_width)
            list("length" "float" truncated_bus_wg_length)
            list("continuity" "string" "ISOLATED")
            list("layer" "string" layer)
            list("width_mode" "string" "FIXED")
            list("thickness" "string" thickness)
            list("wavelength" "string" wavelength)
            list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x3 y3))
            list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x3 y3))
            list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
          )
        )
        x3 = x3
        y3 = y3 + truncated_bus_wg_length
      else
        ; output bus waveguide
        if( (strcmp("CUSTOM" resonator_type) == 0) then
          x3 = x0 + 2*bend_radius + ring_radius + bus_pitch
          y3 = bend_radius
        else
          x3 = x0 + bus_pitch
          y3 = bend_radius
          rotation = "R90"
          dbCreateParamInst( pcCellView wg_master nil x3:y3 rotation 1
            list(
              list("width" "float" lead_wg_width)
              list("length" "float" output_bus_wg_length)
              list("continuity" "string" "ISOLATED")
              list("layer" "string" layer)
              list("width_mode" "string" "FIXED")
              list("thickness" "string" thickness)
              list("wavelength" "string" wavelength)
              list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x3 y3))
              list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x3 y3))
              list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
            )
```

```
            )
            x3 = x3
            y3 = y3 + output_bus_wg_length
          )
        )
      ; radiation taper
      if( ((strcmp("SINGLE" resonator_type) == 0) || (strcmp("RACETRACK" resonator_type) == 0) || (strcmp("CUSTOM"
resonator_type) == 0)) then
        x2 = x3
        y2 = y3-output_bus_wg_length
        x3 = x3
        y3 = y3
        rotation = "R0"
      else
        if( (strcmp("DOUBLE" resonator_type) == 0) then
          x2 = x3
          y2 = y3
          x3 = x3
          y3 = y3 - output_bus_wg_length
          rotation = "R180"
        )
      )
      sub_pc_x_origin = CalcSubOriginX(pc_rotation pc_x_origin x3 y3)
      sub_pc_y_origin = CalcSubOriginY(pc_rotation pc_y_origin x3 y3)
      sub_rotation = CalcSubRotation(pc_rotation rotation)
      dbCreateParamInst( pcCellView wg_taper_master nil x3:y3 rotation 1
        list(
          list("length" "float" rad_taper_length)
          list("start_width" "float" bus_wg_width)
          list("end_width" "float" rad_min_width)
          list("start_width_mode" "string" "FIXED")
          list("end_width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("layer" "string" layer)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
          list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
        )
      )
      ; radiation wg
      if( ((strcmp("SINGLE" resonator_type) == 0) || (strcmp("RACETRACK" resonator_type) == 0) || (strcmp("CUSTOM"
resonator_type) == 0)) then
        x3 = x3
        y3 = y3 + rad_taper_length
        rotation = "R90"
```

```
    else
      if( (strcmp("DOUBLE" resonator_type) == 0) then
        x3 = x3
        y3 = y3 - rad_taper_length
        rotation = "R270"
    )
    )
    dbCreateParamInst( pcCellView wg_master nil x3:y3 rotation 1
      list(
        list("width" "float" rad_min_width)
        list("length" "float" rad_wg_length)
        list("continuity" "string" "OUTPUT")
        list("layer" "string" layer)
        list("width_mode" "string" "FIXED")
        list("thickness" "string" thickness)
        list("wavelength" "string" wavelength)
        list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x3 y3))
        list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x3 y3))
        list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
      )
    )
); end no drop if
; make the over 90 bend to the through port wg
x0 = x0
y0 = y0
rotation = "MXR90"
dbCreateParamInst( pcCellView wg_bend_master nil x0:y0 rotation 1
  list(
    list("width" "float" lead_wg_width)
    list("radius" "float" bend_radius )
    list("angle" "string" "90")
    list("layer" "string" layer)
    list("width_mode" "string" "FIXED")
    list("thickness" "string" thickness)
    list("wavelength" "string" wavelength)
    list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
    list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
    list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
  ) ; close list
); close inst
x0 = x0 + bend_radius
y0 = y0 + bend_radius
if( abs(through_drop_pitch-y0) > 0.001 then
  through_lead_length = through_lead_length - wg_offset_length
)
; wg to the through port coupler or offset
```

```
  rotation = "R0"
  dbCreateParamInst( pcCellView wg_master nil x0:y0 rotation 1
    list(
      list("width" "float" lead_wg_width)
      list("length" "float" through_lead_length)
      list("continuity" "string" "THROUGH")
      list("layer" "string" layer)
      list("width_mode" "string" "FIXED")
      list("thickness" "string" thickness)
      list("wavelength" "string" wavelength)
      list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
      list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
      list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
    )
  )
  if( ((strcmp("SINGLE" resonator_type) == 0) || (strcmp("RACETRACK" resonator_type) == 0) || (strcmp("CUSTOM"
resonator_type) == 0)) then
    if( abs(through_drop_pitch-y0) > 0.001 then
      x0 = x0 + through_lead_length
      y0 = y0
      rotation = "MX"
      sub_pc_x_origin = CalcSubOriginX(pc_rotation pc_x_origin x0 y0)
      sub_pc_y_origin = CalcSubOriginY(pc_rotation pc_y_origin x0 y0)
      sub_rotation = CalcSubRotation(pc_rotation rotation)
      dbCreateParamInst( pcCellView wg_offset_master nil x0:y0 rotation 1
        list(
          list("width" "float" lead_wg_width)
          list("delta_x" "float" wg_offset_length)
          list("delta_y" "float" internal_pitch - through_drop_pitch)
          list("width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("layer" "string" layer)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
          list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
        ) ; close list
      ); close inst
    )
  else
    if( (strcmp("DOUBLE" resonator_type) == 0) then
      if( abs(through_drop_pitch-y0) > 0.001 then
        x0 = x0 + through_lead_length
        y0 = y0
        rotation = "MX"
        dbCreateParamInst( pcCellView wg_offset_master nil x0:y0 rotation 1
```

```
            list(
              list("width" "float" lead_wg_width)
              list("delta_x" "float" wg_offset_length)
              list("delta_y" "float" internal_pitch - through_drop_pitch)
              list("width_mode" "string" "FIXED")
              list("thickness" "string" thickness)
              list("wavelength" "string" wavelength)
              list("layer" "string" layer)
              list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
              list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
              list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
            ) ; close list
          ); close inst
        )
      )
    )
    x0 = input_through_offset
    y0 = through_drop_pitch
    ; through coupler
    if( (strcmp("ISOLATED" continuity) == 0) || (strcmp("OUTPUT" continuity) == 0) then
      rotation = "R0"
      dbCreateParamInst( pcCellView wg_master nil x0:y0 rotation 1
        list(
          list("width" "float" lead_wg_width)
          list("length" "float" 1.0)
          list("continuity" "string" "THROUGH")
          list("layer" "string" layer)
          list("width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
          list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
        )
      )
      x0 = x0+1.0
      y0 = y0
      if( ((strcmp("YES" flip_couplers) == 0) || (strcmp("YES" flip_through_coupler_only) == 0)) then
        rotation = "MXR90"
      else
        rotation = "R270"
      )
      inst = dbCreateParamInst( pcCellView coupler_master nil x0:y0 rotation 1
        list(
          list("width" "float" lead_wg_width)
          list("variant" "string" coupler_variant)
```

```
            list("layer" "string" layer)
            list("width_mode" "string" "FIXED")
            list("thickness" "string" thickness)
            list("wavelength" "string" wavelength)
            list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x2 y2))
            list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x2 y2))
            list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
          )
        )
      else
        if( ( (strcmp("ISOLATED_PD_THROUGH" continuity) == 0) ||
          (strcmp("THROUGH_PD_THROUGH" continuity) == 0) ) then
          ; wg placeholder for a photodiode cell
          rotation = "R0"
          sub_pc_x_origin = CalcSubOriginX(pc_rotation pc_x_origin x0 y0)
          sub_pc_y_origin = CalcSubOriginY(pc_rotation pc_y_origin x0 y0)
          sub_rotation = CalcSubRotation(pc_rotation rotation)
          dbCreateParamInst( pcCellView wg_master nil x0:y0 rotation 1
            list(
              list("width" "float" lead_wg_width)
              list("length" "float" 2.0)
              list("continuity" "string" "OUTPUT")
              list("layer" "string" layer)
              list("width_mode" "string" "FIXED")
              list("thickness" "string" thickness)
              list("wavelength" "string" wavelength)
              list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x0 y0))
              list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x0 y0))
              list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
            )
          )
        )
      )
      if( (strcmp("NO" no_drop_wg) == 0) then
        ; make the down 90 bend to the drop port wg
        x2 = x2
        y2 = y2
        if( ((strcmp("SINGLE" resonator_type) == 0) || (strcmp("RACETRACK" resonator_type) == 0) || (strcmp("CUSTOM"
resonator_type) == 0)) then
          rotation = "R270"
        else
          if( (strcmp("DOUBLE" resonator_type) == 0) then
            rotation = "MXR90"
          )
        )
        dbCreateParamInst( pcCellView wg_bend_master nil x2:y2 rotation 1
```

```
        list(
          list("width" "float" lead_wg_width)
          list("radius" "float" bend_radius )
          list("angle" "string" "90")
          list("layer" "string" layer)
          list("width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x2 y2))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x2 y2))
          list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
        ) ; close list
      ); close inst
      ; wg to the drop coupler
      if( ((strcmp("SINGLE" resonator_type) == 0) || (strcmp("RACETRACK" resonator_type) == 0) || (strcmp("CUSTOM"
resonator_type) == 0)) then
        x2 = x2 + bend_radius
        y2 = y2 - bend_radius
        rotation = "R0"
        if( coupler_bus_pitch > 0.001 then
          drop_lead_length = drop_lead_length - wg_offset_length
        )
      else
        if( (strcmp("DOUBLE" resonator_type) == 0) then
          x2 = x2 + bend_radius
          y2 = y2 + bend_radius
          rotation = "R0"
          if( abs(through_drop_pitch-y2) > 0.001 then
            drop_lead_length = drop_lead_length - wg_offset_length
          )
        )
      )
      dbCreateParamInst( pcCellView wg_master nil x2:y2 rotation 1
        list(
          list("width" "float" lead_wg_width)
          list("length" "float" drop_lead_length)
          list("continuity" "string" "THROUGH")
          list("layer" "string" layer)
          list("width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x2 y2))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x2 y2))
          list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
        )
      )
```

```
    x2 = x2+drop_lead_length
    y2 = y2
    if( ((strcmp("SINGLE" resonator_type) == 0) || (strcmp("RACETRACK" resonator_type) == 0) || (strcmp("CUSTOM"
resonator_type) == 0)) then
      if( coupler_bus_pitch > 0.001 then
        rotation = "R0"
        dbCreateParamInst( pcCellView wg_offset_master nil x2:y2 rotation 1
          list(
            list("width" "float" lead_wg_width)
            list("delta_x" "float" wg_offset_length)
            list("delta_y" "float" coupler_bus_pitch)
            list("width_mode" "string" "FIXED")
            list("thickness" "string" thickness)
            list("wavelength" "string" wavelength)
            list("layer" "string" layer)
            list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x2 y2))
            list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x2 y2))
            list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
          ) ; close list
        ); close inst
        x2 = input_drop_offset
        y2 = coupler_bus_pitch
      else
        x2 = input_drop_offset
        y2 = 0.0
      )
    else
      if( (strcmp("DOUBLE" resonator_type) == 0) then
        if( abs(coupler_bus_pitch-y2) > 0.001 then
          rotation = "MX"
          dbCreateParamInst( pcCellView wg_offset_master nil x2:y2 rotation 1
            list(
              list("width" "float" lead_wg_width)
              list("delta_x" "float" wg_offset_length)
              list("delta_y" "float" abs(coupler_bus_pitch-y2))
              list("width_mode" "string" "FIXED")
              list("thickness" "string" thickness)
              list("wavelength" "string" wavelength)
              list("layer" "string" layer)
              list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x2 y2))
              list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x2 y2))
              list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
            ) ; close list
          ); close inst
        )
        x2 = input_drop_offset
```

```
        y2 = coupler_bus_pitch
      )
    )
    ; drop port coupler
    if(    ( (strcmp("ISOLATED" continuity) == 0)     ||
        (strcmp("ISOLATED_PD_THROUGH" continuity) == 0) ||
        (strcmp("OUTPUT" continuity) == 0)       ||
        (strcmp("INPUT" continuity) == 0)       ||
        (strcmp("THROUGH" continuity) == 0)     ) then
      rotation = "R0"
      dbCreateParamInst( pcCellView wg_master nil x2:y2 rotation 1
        list(
          list("width" "float" lead_wg_width)
          list("length" "float" 1.0)
          list("continuity" "string" "THROUGH")
          list("layer" "string" layer)
          list("width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x2 y2))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x2 y2))
          list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
        )
      )
      x2 = x2+1.0
      y2 = y2
      if( (strcmp("YES" flip_couplers) == 0) then
        rotation = "MXR90"
      else
        rotation = "R270"
      )
      inst = dbCreateParamInst( pcCellView coupler_master nil x2:y2 rotation 1
        list(
          list("width" "float" lead_wg_width)
          list("variant" "string" coupler_variant)
          list("layer" "string" layer)
          list("width_mode" "string" "FIXED")
          list("thickness" "string" thickness)
          list("wavelength" "string" wavelength)
          list("pc_x_origin" "float" CalcSubOriginX(pc_rotation pc_x_origin x2 y2))
          list("pc_y_origin" "float" CalcSubOriginY(pc_rotation pc_y_origin x2 y2))
          list("pc_rotation" "string" CalcSubRotation(pc_rotation rotation))
        )
      )
    )
  ) ; End no drop if
```

```
) ;end of body of code
) ;end of pcell definition
```

### 10.4.6. Component Description Language P-Cell UI Formatting Example

```
LIBRARY = "genericPhotonicsLib"
CELL = "wg_bend"
let( ( cellId cdfId cdfParamId )
  unless( cellId = ddGetObj( LIBRARY CELL )
    error( "Could not get cell %s" CELL )
  )
  when( cdfId = cdfGetBaseCellCDF( cellId )
    cdfDeleteCDF( cdfId )
  )
  cdfId = cdfCreateBaseCellCDF( cellId )
  cdfParamId = cdfCreateParam( cdfId
    ?name "width_mode"
    ?prompt "Width Mode:"
    ?type "radio"
    ?choices list("FIXED" "SINGLEMODE")
    ?defValue "SINGLEMODE"
    ?storeDefault "yes"
    ?display "t"
  )
  cdfParamId = cdfCreateParam( cdfId
    ?name "wavelength"
    ?prompt "Wavelength (nm):"
    ?type "cyclic"
    ?choices list("1280" "1400" "1550")
    ?defValue "1550"
    ?storeDefault "yes"
    ?use "cdfgData->width_mode->value != \"FIXED\""
  )
  cdfParamId = cdfCreateParam( cdfId
    ?name "thickness"
    ?prompt "Core Thickness (nm):"
    ?type "cyclic"
    ?choices list("100" "200")
    ?defValue "200"
    ?storeDefault "yes"
    ?use "cdfgData->width_mode->value != \"FIXED\""
  )

  cdfParamId = cdfCreateParam( cdfId
    ?name "width"
    ?prompt "Waveguide Width:"
```

```
    ?type       "float"
    ?defValue  0.400
    ?storeDefault   "yes"
    ?use    "cdfgData->width_mode->value  == \"FIXED\""
)
cdfParamId = cdfCreateParam( cdfId
    ?name     "radius"
    ?prompt       "Bend Radius:"
    ?type       "float"
    ?defValue  5.0
    ?storeDefault   "yes"
    ?display    "t"
)
cdfParamId = cdfCreateParam( cdfId
    ?name     "angle"
    ?prompt       "Bend Angle:"
    ?type       "radio"
    ?choices   list("90"  "180")
    ?defValue  "90"
    ?storeDefault   "yes"
    ?display    "t"
)
cdfParamId = cdfCreateParam( cdfId
    ?name     "layer"
    ?prompt       "Core Layer:"
    ?type       "cyclic"
    ?choices   list("POLY"  "BODY")
    ?defValue  "POLY"
    ?storeDefault   "yes"
    ?display    "t"
)
cdfParamId = cdfCreateParam( cdfId
    ?name     "pc_x_origin"
    ?prompt       "PCell X Origin:"
    ?type       "float"
    ?defValue  0.0
    ?storeDefault   "no"
    ?display    "nil"
)
cdfParamId = cdfCreateParam( cdfId
    ?name     "pc_y_origin"
    ?prompt       "PCell Y Origin:"
    ?type       "float"
    ?defValue  0.0
    ?storeDefault   "no"
    ?display    "nil"
```

```
  )
  cdfParamId = cdfCreateParam( cdfId
    ?name      "pc_rotation"
    ?prompt      "PCell Rotation:"
    ?type      "string"
    ?defValue  "R0"
    ?storeDefault   "no"
    ?display    "nil"
  )
  cdfId~>doneProc = ""
  cdfSaveCDF( cdfId )
)
```

### 10.4.7. Automatic P-Cell Layout Position Setting Script

```
procedure(poeRepositionPCells( cellDB )
 prog(
    (
       libName cellName viewName fillDB inst instName parseName instList i instNum instCnt maxCnt fillcnt
       instCellName instXOrigin instYOrigin instRotation ll_coord_x ll_coord_y ur_coord_x ur_coord_y num_rect
       file_port string_port dummy
    )
    if(!cellDB then
         cellDB=getWindowRep()
    );if
    if(cellDB then
     libName=cellDB~>libName
     cellName=cellDB~>cellName
     viewName=cellDB~>viewName
     dbReopen(cellDB "a")
     instList=cellDB~>instances
     i = 1
     while(i <= length(instList)
       inst=nthelem(i instList)
    instCellName = inst~>cellName
    if(     ( (strcmp("resonator_cell" instCellName) == 0)   ||
            (strcmp("resonator_rotated_cell" instCellName) == 0)   ||
            (strcmp("resonator_ADC_cell" instCellName) == 0) ||
            (strcmp("paperclip_simple_cell"  instCellName) == 0) ||
            (strcmp("wg_cell" instCellName) == 0)          ) then
     instXOrigin = car(inst~>xy)
     instYOrigin = cadr(inst~>xy)
     instRotation = cadr(inst~>transform)
     dbReplaceProp(inst "cell_pc_x_origin" "float" instXOrigin)
     dbReplaceProp(inst "cell_pc_y_origin" "float" instYOrigin)
     dbReplaceProp(inst "cell_pc_rotation" "string" instRotation)
```

```
    )
     if(    (  (strcmp("wg_bend"  instCellName)  == 0)     ||
            (strcmp("wg_offset"  instCellName)  == 0)      ||
            (strcmp("wg_taper"  instCellName)  == 0)     ||
            (strcmp("splitter_3dB_1200nm"  instCellName)  == 0)      ||
            (strcmp("coupler"  instCellName)  == 0)      ||
            (strcmp("wg"  instCellName)  == 0)          ) then
     instXOrigin  = car(inst~>xy)
     instYOrigin  = cadr(inst~>xy)
     instRotation  = cadr(inst~>transform)
     dbReplaceProp(inst "pc_x_origin"  "float"  instXOrigin)
     dbReplaceProp(inst "pc_y_origin"  "float"  instYOrigin)
     dbReplaceProp(inst "pc_rotation"  "string"  instRotation)
    )
      i=i+1
     );while i
  else
   print("Some  error  with cell DB")
    );if cellDB
);prog
);procedure
```

## 10.5. Autofill Generation Scripts

### 10.5.1. Generic Calibre Location Identification Script

```
// Autofill Generation Script
// Current geometric grow / check verision by Jason Orcutt
// Original author Ben Moss <benm@mit.edu>
// Based on Fred Chen's ISG command file for attaching power/ground nets.
//############################################
//# Variables
//############################################
VARIABLE MARKERSHAPE_GROW_DISTANCE 8.0
VARIABLE EXCLUDESHAPE_GROW_DISTANCE 1.0
//############################################
//# Environment Setup
//############################################
PRECISION   1000
RESOLUTION    1  // Layout grid is 0.001um
LAYOUT SYSTEM GDSII
LAYOUT PRIMARY "*"
#IFNDEF $DRC_RESULTS_DATABASE
  DRC RESULTS DATABASE "drc_results.ascii"
#ELSE
  DRC RESULTS DATABASE $DRC_RESULTS_DATABASE
```

```
#ENDIF
#IFNDEF $DRC_SUMMARY_REPORT
  DRC SUMMARY REPORT "drc_summary"
#ELSE
  DRC SUMMARY REPORT $DRC_SUMMARY_REPORT
#ENDIF
//###############################################
//# DRC FUNCTIONS
//###############################################
DRC MAXIMUM RESULTS ALL
//###############################################
//# MAPPING
//###############################################
LAYER MARKERSHAPE   300 // single layer used for generic process info
LAYER MAP 1 DATATYPE 0 300
LAYER EXCLUDESHAPE 301
LAYER MAP 2 DATATYPE 0 301
//###############################################
//# LOGIC OPERATION POOL
//###############################################
EXCLUDESHAPE_GROW = EXCLUDESHAPE SIZE BY EXCLUDESHAPE_GROW_DISTANCE
TOTAL_CUT = EXCLUDESHAPE_GROW
//###############################################
//# MANIPULATE EXCLUDE_REGION
//###############################################
MARKERSHAPE_GROW = MARKERSHAPE SIZE BY MARKERSHAPE_GROW_DISTANCE
REGION_TO_FILL = MARKERSHAPE_GROW NOT (MARKERSHAPE OR TOTAL_CUT)
REGION_TO_CHECK = REGION_TO_FILL OR MARKERSHAPE
//###############################################
//# RULE CHECKS
//###############################################
fillblock_find_check {
DENSITY REGION_TO_FILL > 0.99 INSIDE OF 0 0 3000 3000 WINDOW 0.8 STEP 0.8 PRINT '/generic_location/fill_coords.txt'
} //Output the verticies of each 800x800nm rectangle in the fill region
//###############################################
//# EXPORT CHECKS
//###############################################
DRC SELECT CHECK fillblock_find_check
```

## 10.5.2. SKILL Fill Cell Array Insertion and Deletion Scripts

### *10.5.2.1. poeInstantiateFillCells*

```
procedure(poeInstantiateFillCells( cellDB filllib fillcell instPrefix coordinateFile)
 prog(
    (
```

```
    libName cellName viewName fillDB inst instName parseName instList i instNum instCnt maxCnt xy fillcnt
    ll_coord_x ll_coord_y ur_coord_x ur_coord_y num_rect file_port string_port dummy array_y_count array_x_count
    cell_count current_y_count array_start_x array_start_y array_pos array_size prev_y_array array_y_count_list
    array_x_count_list current_y_count_list cell_count_list array_start_x_list array_start_y_list array_y_count_list_old
    array_x_count_list_old current_y_count_list_old cell_count_list_old array_start_x_list_old array_start_y_list_old
    max_array active_array array_index fill_block_placed cycles_since_update min_update_interval array_update_index
   )
  if(!cellDB then
    cellDB=getWindowRep()
  );if
  fillDB = dbOpenCellViewByType(fillLib fillCell "layout" nil "r")
   if(fillDB then
    print("Loaded fill cell");
   else
    print("ERROR: could not find fill cell");
    return;
   )
if(cellDB then
    libName=cellDB~>libName
    cellName=cellDB~>cellName
    viewName=cellDB~>viewName
    dbReopen(cellDB "a")
    instList=cellDB~>instances~>name
    instCnt=0
    i=1
    maxCnt=0
    while(i <= length(instList)
      instName=nthelem(i instList)
      parseName=parseString(instName "_")
      if(nthelem(1 parseName) == instPrefix then
        instCnt=evalstring(nthelem(2 parseName))
        if(instCnt > maxCnt then
          maxCnt = instCnt
        );if instCnt
       );if
      i=i+1
    );while i
    instCnt=maxCnt
    fillcnt = 0
    file_port = infile(coordinateFile)
  max_array = 20
  active_array = 0
  cell_count = 0
  cycles_since_update = list(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
    array_x_count_list = list(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
    array_y_count_list = list(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
```

```
cell_count_list = list(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
current_y_count_list = list(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
array_start_x_list = list( 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 )
array_start_y_list = list( 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 )
dummy = gets(line file_port)
while( dummy
    string_port = instring(line)
    dummy = fscanf(string_port "%f %f %f %f %d" ll_coord_x ll_coord_y ur_coord_x ur_coord_y num_rect)
    if( dummy then
     fill_block_placed = nil
     array_index = 0
     while( not(fill_block_placed || (array_index >= (max_array - 0.001)))

     array_index = array_index + 1
     array_start_x = nthelem(array_index array_start_x_list)
     array_start_y = nthelem(array_index array_start_y_list)
     array_x_count = nthelem(array_index array_x_count_list)
     array_y_count = nthelem(array_index array_y_count_list)
     cell_count = nthelem(array_index cell_count_list)
     current_y_count = nthelem(array_index current_y_count_list)
     if( ( (abs(ll_coord_x - (array_start_x+array_x_count*0.8)) < 0.001) &&
         (abs(ll_coord_y - array_start_y) < 0.001) ) then
       if( (abs(array_y_count - current_y_count) < 0.001) then
         array_x_count = array_x_count + 1
         current_y_count = 1
         cell_count = cell_count + 1
       else
         array_pos = list(array_start_x array_start_y)
         array_size = list((array_x_count-1) array_y_count)
         fillcnt = fillcnt+1
         instNum = instCnt + fillcnt
         sprintf(instName "%s_%d" instPrefix instNum)
     inst=dbCreateSimpleMosaic(cellDB fillDB instName array_pos "R0" cadr(array_size) car(array_size) 0.8 0.8)
         array_start_x = ll_coord_x-0.8
         array_start_y = ll_coord_y
         array_x_count = 2
         array_y_count = current_y_count
         current_y_count = 1
       )
       fill_block_placed = t
     else
       if( ( (abs(ll_coord_y - (array_start_y+current_y_count*0.8)) < 0.001) &&
           (abs(ll_coord_x - (array_start_x+(array_x_count-1)*0.8)) < 0.001) ) then

         fill_block_placed = t
         if( (array_x_count < 1.001) then
```

```
            cell_count = cell_count + 1
            array_y_count = array_y_count + 1
            current_y_count = current_y_count + 1
          else
            if( (current_y_count < (array_y_count-0.001)) then
              current_y_count = current_y_count + 1
              cell_count = cell_count + 1
            else
              array_pos = list(array_start_x array_start_y)
              array_size = list((array_x_count-1) array_y_count)
              array_start_x = ll_coord_x
              array_start_y = array_start_y
              array_x_count = 1
              current_y_count = current_y_count + 1
              array_y_count = current_y_count
              cell_count = 1
              ; make array
              fillcnt = fillcnt+1
              instNum = instCnt + fillcnt
              sprintf(instName "%s_%d" instPrefix instNum)
      inst=dbCreateSimpleMosaic(cellDB fillDB instName array_pos "R0" cadr(array_size) car(array_size) 0.8 0.8)
            )
          )
        else
          array_x_count = 1
          array_y_count = 1
          current_y_count = 1
          array_start_x = ll_coord_x
          array_start_y = ll_coord_y
          cell_count = 1
        )
      )
    ) ; end array index while
    if( fill_block_placed then
      array_start_x_list_old = array_start_x_list
      array_start_y_list_old = array_start_y_list
      array_x_count_list_old = array_x_count_list
      array_y_count_list_old = array_y_count_list
      cell_count_list_old = cell_count_list
      current_y_count_list_old = current_y_count_list
      array_start_x_list = list()
      array_start_y_list = list()
      array_x_count_list = list()
      array_y_count_list = list()
      cell_count_list = list()
      current_y_count_list = list()
```

```
    array_update_index = max_array
    while( array_update_index > 0.001
      if( (abs(array_update_index - array_index) < 0.001) then
        array_start_x_list = cons(array_start_x array_start_x_list)
        array_start_y_list = cons(array_start_y array_start_y_list)
        array_x_count_list = cons(array_x_count array_x_count_list)
        array_y_count_list = cons(array_y_count array_y_count_list)
        cell_count_list = cons(cell_count cell_count_list)
        current_y_count_list = cons(current_y_count current_y_count_list)
      else
        array_start_x_list = cons(nthelem(array_update_index array_start_x_list_old) array_start_x_list)
        array_start_y_list = cons(nthelem(array_update_index array_start_y_list_old) array_start_y_list)
        array_x_count_list = cons(nthelem(array_update_index array_x_count_list_old) array_x_count_list)
        array_y_count_list = cons( nthelem(array_update_index array_y_count_list_old) array_y_count_list)
        cell_count_list = cons( nthelem(array_update_index cell_count_list_old) cell_count_list)
        current_y_count_list = cons(nthelem(array_update_index current_y_count_list_old) current_y_count_list)
      )
      array_update_index = array_update_index - 1
    ) ; end array update while
  else
    if( (active_array < (max_array-0.001)) then
      ; start keeping track of the new array
      array_start_x_list_old = array_start_x_list
      array_start_y_list_old = array_start_y_list
      array_x_count_list_old = array_x_count_list
      array_y_count_list_old = array_y_count_list
      current_y_count_list_old = current_y_count_list
      cell_count_list_old = cell_count_list
      array_start_x_list = list()
      array_start_y_list = list()
      array_x_count_list = list()
      array_y_count_list = list()
      current_y_count_list = list()
      cell_count_list = list()
      active_array = active_array + 1
      array_index = active_array
      array_update_index = max_array
      while( array_update_index > 0.001
        if( (abs(array_update_index - array_index) < 0.001) then
          array_start_x_list = cons(ll_coord_x array_start_x_list)
          array_start_y_list = cons(ll_coord_y array_start_y_list)
          array_x_count_list = cons(1 array_x_count_list)
          array_y_count_list = cons(1 array_y_count_list)
          current_y_count_list = cons(1 current_y_count_list)
          cell_count_list = cons(1 cell_count_list)
        else
```

```
        array_start_x_list = cons(nthelem(array_update_index array_start_x_list_old) array_start_x_list)
        array_start_y_list = cons(nthelem(array_update_index array_start_y_list_old) array_start_y_list)
        array_x_count_list = cons(nthelem(array_update_index array_x_count_list_old) array_x_count_list)
        array_y_count_list = cons( nthelem(array_update_index array_y_count_list_old) array_y_count_list)
      current_y_count_list = cons(nthelem(array_update_index current_y_count_list_old) current_y_count_list)
        cell_count_list = cons(nthelem(array_update_index cell_count_list_old) current_y_count_list)
      )
      array_update_index = array_update_index - 1
    ) ; end array update while
  else
    ; find oldest
    array_update_index = max_array
    min_update_interval = -1
    while( array_update_index > 0.001
      if( (nthelem(array_update_index cycles_since_update) > min_update_interval) then
        min_update_interval = nthelem(array_update_index cycles_since_update)
        array_index = array_update_index
      )
      array_update_index = array_update_index - 1
    ) ; end cycle_update
    ; pop off oldest info
    array_start_x = nthelem(array_index array_start_x_list)
    array_start_y = nthelem(array_index array_start_y_list)
    array_x_count = nthelem(array_index array_x_count_list)
    array_y_count = nthelem(array_index array_y_count_list)
    current_y_count = nthelem(array_index current_y_count_list)
    if( ((array_y_count > 1.001) || (array_x_count > 1.001)) then
      if( (abs(current_y_count - array_y_count) < 0.001) then
        array_pos = list(array_start_x array_start_y)
        array_size = list(array_x_count array_y_count)
        ; make array
        fillcnt = fillcnt+1
        instNum = instCnt + fillcnt
        sprintf(instName "%s_%d" instPrefix instNum)
    inst=dbCreateSimpleMosaic(cellDB fillDB instName array_pos "R0" cadr(array_size) car(array_size) 0.8 0.8)
      else
        if( (current_y_count < (array_y_count - 0.001)) then
          ; make first array
          array_pos = list(array_start_x array_start_y)
          if( (abs(array_x_count) < 1.001) then
            array_size = list(1 array_y_count)
          else
            array_size = list((array_x_count-1) array_y_count)
          )
          fillcnt = fillcnt+1
          instNum = instCnt + fillcnt
```

```
            sprintf(instName "%s_%d" instPrefix instNum)
        inst=dbCreateSimpleMosaic(cellDB fillDB instName array_pos "R0" cadr(array_size) car(array_size) 0.8 0.8)
            ; make 2nd array
            array_pos = list(array_start_x+0.8*(array_x_count-1) array_start_y)
            fillcnt = fillcnt+1
            instNum = instCnt + fillcnt
            sprintf(instName "%s_%d" instPrefix instNum)
            if( (current_y_count > 1.001) then
              array_size = list(1 current_y_count)
        inst=dbCreateSimpleMosaic(cellDB fillDB instName array_pos "R0" cadr(array_size) car(array_size) 0.8 0.8)
            else
              inst=dbCreateInst(cellDB fillDB instName array_pos "R0" 1)
            )
          else
            println("what happened")
          )
        ) ; end current_y_cound == array_y_count if
        else
          array_pos = list(array_start_x array_start_y)
          ; make 2nd array
          fillcnt = fillcnt+1
          instNum = instCnt + fillcnt
          sprintf(instName "%s_%d" instPrefix instNum)
          inst=dbCreateInst(cellDB fillDB instName array_pos "R0" 1)
        ) ; end if mosaic if
        ; start keeping track of the new array in the refresh location
        array_start_x_list_old = array_start_x_list
        array_start_y_list_old = array_start_y_list
        array_x_count_list_old = array_x_count_list
        array_y_count_list_old = array_y_count_list
        current_y_count_list_old = current_y_count_list
        cell_count_list_old = cell_count_list
        array_start_x_list = list()
        array_start_y_list = list()
        array_x_count_list = list()
        array_y_count_list = list()
        current_y_count_list = list()
        cell_count_list = list()
        array_update_index = max_array
while( array_update_index > 0.001
 if( (abs(array_update_index - array_index) < 0.001) then
          array_start_x_list = cons(ll_coord_x array_start_x_list)
          array_start_y_list = cons(ll_coord_y array_start_y_list)
          array_x_count_list = cons(1 array_x_count_list)
          array_y_count_list = cons(1 array_y_count_list)
          current_y_count_list = cons(1 current_y_count_list)
```

```
            cell_count_list = cons(1 cell_count_list)
    else
            array_start_x_list = cons(nthelem(array_update_index array_start_x_list_old) array_start_x_list)
            array_start_y_list = cons(nthelem(array_update_index array_start_y_list_old) array_start_y_list)
            array_x_count_list = cons(nthelem(array_update_index array_x_count_list_old) array_x_count_list)
            array_y_count_list = cons( nthelem(array_update_index array_y_count_list_old) array_y_count_list)
            current_y_count_list = cons(nthelem(array_update_index current_y_count_list_old) current_y_count_list)
            cell_count_list = cons(nthelem(array_update_index cell_count_list_old) current_y_count_list)
    )
                        array_update_index = array_update_index - 1
                    ) ;end array update while
        ) ; end array < max array if
      ) ; end fill block_placed if
      cycles_since_update_old = cycles_since_update
      cycles_since_update = list()
      array_update_index = max_array
      while( array_update_index > 0
        if( (abs(array_update_index-array_index) < 0.001) then
          cycles_since_update = cons(0.0 cycles_since_update)
        else
          cycles_since_update = cons((nthelem(array_update_index cycles_since_update_old) + 1) cycles_since_update)
        )
        array_update_index = array_update_index - 1
      ) ; end cycle_update
        else
            print("ERROR parsing file line")
        )
        dummy = gets(line file_port)
    );while
    close(file_port)
    ; instantiate the open cell instance(s)
array_index = 0
while( (array_index < active_array)
    array_index = array_index + 1
    array_start_x = nthelem(array_index array_start_x_list)
    array_start_y = nthelem(array_index array_start_y_list)
    array_x_count = nthelem(array_index array_x_count_list)
    array_y_count = nthelem(array_index array_y_count_list)
    current_y_count = nthelem(array_index current_y_count_list)
    if( ((array_y_count > 1.001) || (array_x_count > 1.001)) then
      if( (current_y_count == array_y_count) then
        array_pos = list(array_start_x array_start_y)
        array_size = list(array_x_count array_y_count)
        ; make array
        fillcnt = fillcnt+1
        instNum = instCnt + fillcnt
```

```
        sprintf(instName "%s_%d" instPrefix instNum)
        inst=dbCreateSimpleMosaic(cellDB fillDB instName array_pos "R0" cadr(array_size) car(array_size) 0.8 0.8)
      else
        if( (current_y_count < array_y_count) then
          ; make 1rst array
          array_pos = list(array_start_x array_start_y)
            if( (array_x_count == 0) then
              array_size = list(1 array_y_count)
            else
              array_size = list((array_x_count-1) array_y_count)
            )
          fillcnt = fillcnt+1
          instNum = instCnt + fillcnt
          sprintf(instName "%s_%d" instPrefix instNum)
          inst=dbCreateSimpleMosaic(cellDB fillDB instName array_pos "R0" cadr(array_size) car(array_size) 0.8 0.8)
          ; make 2nd array
          array_pos = list(array_start_x+0.8*(array_x_count-1)    array_start_y)
          fillcnt = fillcnt+1
          instNum = instCnt + fillcnt
          sprintf(instName "%s_%d" instPrefix instNum)
          if( (current_y_count > 1.001) then
            array_size = list(1          current_y_count)
          inst=dbCreateSimpleMosaic(cellDB fillDB instName array_pos "R0" cadr(array_size) car(array_size) 0.8 0.8)
          else
            inst=dbCreateInst(cellDB fillDB instName array_pos "R0" 1)
          )
        else
          println("what happened")
        )
      )
    else
      array_pos = list(array_start_x    array_start_y)
      ; make 2nd array
      fillcnt = fillcnt+1
      instNum = instCnt + fillcnt
      sprintf(instName "%s_%d" instPrefix instNum)
      inst=dbCreateInst(cellDB fillDB instName array_pos "R0" 1)
    )
  ); end final cycle through open array while loop
  println(instName)
else
 print("Some error with cell DB")
);if cellDB
 );prog
);procedure
```

### *10.5.2.2. poeDeleteFillCells*

```
procedure(poeDeleteFillCells( cellDB instPrefix )
 prog(
     (
         libName cellName viewName fillDB inst instName parseName instList i instNum instCnt maxCnt xy fillcnt
         ll_coord_x ll_coord_y ur_coord_x ur_coord_y num_rect file_port string_port dummy output_string
     )
     if(!cellDB then
          cellDB=getWindowRep()
     );if
     if(cellDB then
      libName=cellDB~>libName
      cellName=cellDB~>cellName
      viewName=cellDB~>viewName
      dbReopen(cellDB "a")
      instList=cellDB~>instances
   i = 1
      while(i <= length(instList)
        inst=nthelem(i instList)
     instName = inst~>name
        parseName=parseString(instName "_")
        if(nthelem(1 parseName) == instPrefix then
      ; Delete the fill cell
        dbDeleteObject(inst)
;       sprintf(output_string "Deleted instance %L" instName)
;       println(output_string)
          );if
        i=i+1
      );while i
   else
    print("Some error with cell DB")
     );if cellDB
 );prog
);procedure
```

# BIBLIOGRAPHY

[1] C. Batten, A. Joshi, J. Orcutt, A. Khilo, B. Moss, C. W. Holzwarth, M. A. Popovic, L. Hanqing, H. I. Smith, J. L. Hoyt, F. X. Kartner, R. J. Ram, V. Stojanovic, and K. Asanovic, "Building Many-Core Processor-to-DRAM Networks with Monolithic CMOS Silicon Photonics," *Micro, IEEE*, vol. 29, pp. 8-21, 2009.

[2] M. Petracca, B. G. Lee, K. Bergman, and L. P. Carloni, "Photonic NoCs: System-Level Design Exploration," *Micro, IEEE*, vol. 29, pp. 74-85, 2009.

[3] D. A. B. Miller, "Device Requirements for Optical Interconnects to Silicon Chips," *Proceedings of the Ieee*, vol. 97, pp. 1166-1185, Jul 2009.

[4] D. A. B. Miller, "Optical interconnects to silicon," *Ieee Journal of Selected Topics in Quantum Electronics*, vol. 6, pp. 1312-1317, Nov-Dec 2000.

[5] K. Bhattacharyya, D. Sutherland, V. Hazari, and F. Mirzaagha, "A cost model comparing the economics of reticle requalification methods in advanced wafer fabs," *2007 Ieee/Semi Advanced Semiconductor Manufacturing Conference*, pp. 63-67, 2007.

[6] R. B. Lin, M. C. Wu, and S. C. Tsai, "Reticle design for minimizing multiproject wafer production cost," *Ieee Transactions on Automation Science and Engineering*, vol. 4, pp. 589-595, Oct 2007.

[7] S. Kyoh, S. Inoue, O. Ikenaga, T. Higaki, F. Shigemitsu, I. Mori, H. Kokubo, N. Hayashi, N. Irie, Y. Ishii, and T. Umatate, "Photomask repeater strategy for high quality and low cost reticle fabrication," *Photomask and Next-Generation Lithography Mask Technology X*, vol. 5130, pp. 16-27, 2003.

[8] S. Koranne, "A high performance SIMD framework for design rule checking on Sony's PlayStation 2 Emotion Engine platform," *Isqed 2004: 5th International Symposium on Quality Electronic Design, Proceedings*, pp. 371-376, 2004.

[9] A. P. V. Pais, M. L. Anido, and C. E. T. Oliveira, "Developing a distributed architecture for Design Rule Checking," *Proceedings of the 44th Ieee 2001 Midwest Symposium on Circuits and Systems, Vols 1 and 2*, pp. 678-681, 2001.

[10] L. Liebmann, J. Lund, F. L. Heng, and I. Graur, "Enabling alternating phase shifted mask designs for a full logic gate level: Design rules and design rule checking," *38th Design Automation Conference Proceedings 2001*, pp. 79-84, 2001.

[11] Z. Luo, M. Martonosi, and P. Ashar, "An edge-endpoint-based configurable hardware architecture for VLSI layout Design Rule Checking," *VLSI Design*, vol. 10, pp. 249-263, 2000.

[12] V. Lal, P. Evans, S. Corzine, M. Kato, M. Fisher, J. Gheorma, V. Dominic, P. Samra, A. Nilsson, J. Rahn, A. Dentai, P. Studenkov, M. Missey, D. Lambert, R. Muthiah, R. Salvatore, S. Murthy, E. Strzelecka, J. Pleumeekers, A. Chen, R. Schneider, R. Nagarajan, M. Ziari, J. Stewart, C. Joyner, F. Kish, and D. Welch, "Current view of large

scale photonic integrated circuits on indium phosphide," in *Semiconductor Laser Conference (ISLC), 2010 22nd IEEE International*, 2010, pp. 55-56.

[13] R. Nagarajan, C. H. Joyner, R. P. Schneider, Jr., J. S. Bostak, T. Butrie, A. G. Dentai, V. G. Dominic, P. W. Evans, M. Kato, M. Kauffman, D. J. H. Lambert, S. K. Mathis, A. Mathur, R. H. Miles, M. L. Mitchell, M. J. Missey, S. Murthy, A. C. Nilsson, F. H. Peters, S. C. Pennypacker, J. L. Pleumeekers, R. A. Salvatore, R. K. Schlenker, R. B. Taylor, T. Huan-Shang, M. F. Van Leeuwen, J. Webjorn, M. Ziari, D. Perkins, J. Singh, S. G. Grubb, M. S. Reffle, D. G. Mehuys, F. A. Kish, and D. F. Welch, "Large-scale photonic integrated circuits," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 11, pp. 50-65, 2005.

[14] D. F. Welch, F. A. Kish, R. Nagarajan, C. H. Joyner, J. R. P. Schneider, V. G. Dominic, M. L. Mitchell, S. G. Grubb, T.-K. Chiang, D. D. Perkins, and A. C. Nilsson, "The Realization of Large-Scale Photonic Integrated Circuits and the Associated Impact on Fiber-Optic Communication Systems," *J. Lightwave Technol.*, vol. 24, pp. 4674-4683, 2006.

[15] H. Kamitsuna, K. Ishii, T. Shibata, K. Kurishima, and M. Ida, "A 43-Gb/s clock and data recovery OEIC integrating an InP-InGaAs HPT oscillator with an HBT decision circuit," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 10, pp. 673-678, 2004.

[16] E. Shumakher, T. Magrisso, S. Kraus, D. Cohen-Elias, A. Gavrilov, S. Cohen, G. Eisenstein, and D. Ritter, "An InP HBT-Based Oscillator Monolithically Integrated With a Photodiode," *Lightwave Technology, Journal of*, vol. 26, pp. 2679-2683, 2008.

[17] M. Mikulics, P. Kordos, D. Gregusova, R. Adam, M. Kocan, W. Shuai, J. Zhang, R. Sobolewski, D. Grutzmacher, and M. Marso, "Monolithic Integration of Ultrafast Photodetector and MESFET in the GaN Material System," *Photonics Technology Letters, IEEE*, vol. 23, pp. 1189-1191, 2011.

[18] Y. Cheng-Guan, T. Der-Wei, S. M. J. Liu, R. Wu, J. Huang, F. Chen, Y. Shih-Wei, and W. Lai, "Quarter-micron optical gate 6″ power pHEMT technology," in *Microwave Conference, 2008. APMC 2008. Asia-Pacific*, 2008, pp. 1-4.

[19] I. A. Young, E. Mohammed, J. T. S. Liao, A. M. Kern, S. Palermo, B. A. Block, M. R. Reshotko, and P. L. D. Chang, "Optical I/O Technology for Tera-Scale Computing," *Ieee Journal of Solid-State Circuits*, vol. 45, pp. 235-248, Jan 2010.

[20] B. Stackhouse, S. Bhimji, C. Bostak, D. Bradley, B. Cherkauer, J. Desai, E. Francom, M. Gowan, P. Gronowski, D. Krueger, C. Morganti, and S. Troyer, "A 65 nm 2-Billion Transistor Quad-Core Itanium Processor," *Solid-State Circuits, IEEE Journal of*, vol. 44, pp. 18-31, 2009.

[21] J. Chang, H. Ming, J. Shoemaker, J. Benoit, C. Szu-Liang, C. Wei, C. Siufu, R. Ganesan, G. Leong, V. Lukka, S. Rusu, and D. Srivastava, "The 65-nm 16-MB Shared On-Die L3 Cache for the Dual-Core Intel Xeon Processor 7100 Series," *Solid-State Circuits, IEEE Journal of*, vol. 42, pp. 846-852, 2007.

[22] M. J. W. Rodwell, M. Wistey, U. Singisetti, G. Burek, A. Gossard, S. Stemmer, R. Engel-Herbert, Y. Hwang, Y. Zheng, C. Van de Walle, P. Asbeck, Y. Taur, A. Kummel, B. Yu,

D. Wang, Y. Yuan, C. Palmstrom, E. Arkun, P. Simmonds, P. McIntyre, J. Harris, M. V. Fischetti, and C. Sachs, "Technology development; design for 22 nm InGaAs/InP-channel MOSFETs," in *Indium Phosphide and Related Materials, 2008. IPRM 2008. 20th International Conference on*, 2008, pp. 1-6.

[23] N. Waldron, K. Dae-Hyun, and J. A. del Alamo, "A Self-Aligned InGaAs HEMT Architecture for Logic Applications," *Electron Devices, IEEE Transactions on*, vol. 57, pp. 297-304, 2010.

[24] S. G. Cloutier, P. A. Kossyrev, and J. Xu, "Optical gain and stimulated emission in periodic nanopatterned crystalline silicon," *Nat Mater*, vol. 4, pp. 887-891, 2005.

[25] B. Huang, X. Zhang, W. Wang, Z. Dong, N. Guan, Z. Zhang, and H. Chen, "CMOS monolithic optoelectronic integrated circuit for on-chip optical interconnection," *Optics Communications*, vol. 284, pp. 3924-3927, 2011.

[26] S. Xiaochen, L. Jifeng, L. C. Kimerling, and J. Michel, "Toward a Germanium Laser for Integrated Silicon Photonics," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 16, pp. 124-131, 2010.

[27] T. Wang, H. Liu, A. Lee, F. Pozzi, and A. Seeds, "1.3-?m InAs/GaAs quantum-dot lasers monolithically grown on Si substrates," *Opt. Express*, vol. 19, pp. 11381-11386, 2011.

[28] R. Chen, T.-T. D. Tran, K. W. Ng, W. S. Ko, L. C. Chuang, F. G. Sedgwick, and C. Chang-Hasnain, "Nanolasers grown on silicon," *Nat Photon*, vol. 5, pp. 170-175, 2011.

[29] A. W. Fang, H. Park, O. Cohen, R. Jones, M. J. Paniccia, and J. E. Bowers, "Electrically pumped hybrid AlGaInAs-silicon evanescent laser," *Opt. Express*, vol. 14, pp. 9203-9210, 2006.

[30] L. Liu, R. Kumar, K. Huybrechts, T. Spuesens, G. Roelkens, E.-J. Geluk, T. de Vries, P. Regreny, D. Van Thourhout, R. Baets, and G. Morthier, "An ultra-small, low-power, all-optical flip-flop memory on a silicon chip," *Nat Photon*, vol. 4, pp. 182-187, 2010.

[31] C. Gunn, "CMOS Photonics for High-Speed Interconnects," *Micro, IEEE*, vol. 26, pp. 58-66, 2006.

[32] L. C. Kimerling, D. Ahn, A. B. Apsel, M. Beals, D. Carothers, Y. K. Chen, T. Conway, D. M. Gill, M. Grove, C. Y. Hong, M. Lipson, J. Liu, J. Michel, D. Pan, S. S. Patel, A. T. Pomerene, M. Rasras, D. K. Sparacin, K. Y. Tu, A. E. White, and C. W. Wong, "Electronic-photonic integrated circuits on the CMOS platform," San Jose, CA, USA, 2006, pp. 612502-10.

[33] Y. Vlasov, W. M. J. Green, and F. Xia, "High-throughput silicon nanophotonic wavelength-insensitive switch for on-chip optical networks," *Nature Photonics*, vol. 2, pp. 242-246, 2008.

[34] W. A. Zortman, D. C. Trotter, A. L. Lentine, G. Robertson, and M. R. Watts, "Monolithic integration of silicon electronics and photonics," in *Winter Topicals (WTM), 2011 IEEE*, 2011, pp. 139-140.

[35] T. Pinguet, B. Analui, E. Balmater, D. Guckenberger, M. Harrison, R. Koumans, D. Kucharski, Y. Liang, G. Masini, A. Mekis, S. Mirsaidi, A. Narasimha, M. Peterson, D. Rines, V. Sadagopan, S. Sahni, T. J. Sleboda, D. Song, Y. Wang, B. Welch, J. Witzens, J. Yao, S. Abdalla, S. Gloeckner, P. De Dobbelaere, and G. Capellini, "Monolithically integrated high-speed CMOS photonic transceivers," in *Group IV Photonics, 2008 5th IEEE International Conference on*, 2008, pp. 362-364.

[36] L. D. Garrett, J. Qi, C. L. Schow, and J. C. Campbell, "A silicon-based integrated NMOS-p-i-n photoreceiver," *IEEE Transactions on Electron Devices*, vol. 43, pp. 411-416, 1996.

[37] C. L. Schow, J. D. Schaub, R. Li, J. Qi, and J. C. Campbell, "A 1-Gb/s monolithically integrated silicon NMOS optical receiver," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 4, pp. 1035-1039, 1998.

[38] M. I. B. Ciftcioglu, M. I. L. Zhang, J. Zhang, M. I. J. R. Marciante, J. Zuegel, R. Sobolewski, and H. Wu, "Integrated Silicon PIN Photodiodes Using Deep N-Well in a Standard 0.18-um CMOS Technology," *Journal of Lightwave Technology*, vol. 27, pp. 3303-3313, 2009.

[39] B. Ciftcioglu, J. Zhang, L. Zhang, J. R. Marciante, J. D. Zuegel, R. Sobolewski, and H. Wu, "3-GHz Silicon Photodiodes Integrated in a 0.18-um CMOS Technology," *IEEE Photonics Technology Letters*, vol. 20, pp. 2069-2071, 2008.

[40] T. Yoshida, Y. Ohtomo, and M. Shimaya, "A novel p-i-n photodetector fabricated on SIMOX for 1 GHz 2 V CMOS OEICs," pp. 29-32, 1998.

[41] S. M. Csutak, J. D. Schaub, S. Wang, J. Mogab, and J. C. Campbell, "Integrated silicon optical receiver with avalanche photodiode," *IEE Proceedings - Optoelectronics*, vol. 150, p. 235, 2003.

[42] Y. Ohtomo, T. Yoshida, M. Nishisaka, K. Nishimura, and M. Shimaya, "A single-chip 3.5 Gb/s CMOS/SIMOX transceiver with automatic-gain-control and automatic-power-control circuits," pp. 58-59, 2000.

[43] H.-S. Kang, M.-J. Lee, and W.-Y. Choi, "Si avalanche photodetectors fabricated in standard complementary metal-oxide-semiconductor process," *Applied Physics Letters*, vol. 90, p. 151118, 2007.

[44] M. Jutzi, M. Grozing, E. Gaugler, W. Mazioschek, and M. Berroth, "2-gb/s CMOS optical integrated receiver with a spatially Modulated photodetector," *Ieee Photonics Technology Letters*, vol. 17, pp. 1268-1270, 2005.

[45] W.-K. Huang, Y.-C. Liu, and Y.-M. Hsin, "A High-Speed and High-Responsivity Photodiode in Standard CMOS Technology," *Ieee Photonics Technology Letters*, vol. 19, pp. 197-199, 2007.

[46] S. M. Csutak, "High-speed short-wavelength silicon photodetectors fabricated in 130-nm CMOS process," pp. 135-145, 2003.

[47] L. Zi-Ying, C. Fang-ping, W. Ching-Wen, and H. Yue-ming, "A new 850-nm lateral Si avalanche photodiode in standard CMOS technology," in *OptoeElectronics and Communications Conference (OECC), 2011 16th*, 2011, pp. 479-480.

[48] C. Camperi-Ginestet, Y. W. Kim, N. M. Jokerst, M. G. Allen, and M. A. Brooke, "Vertical electrical interconnection of compound semiconductor thin-film devices to underlying silicon circuitry," *Photonics Technology Letters, IEEE,* vol. 4, pp. 1003-1006, 1992.

[49] K. W. Goossen, J. E. Cunningham, and W. Y. Jan, "GaAs 850 nm modulators solder-bonded to silicon," *Photonics Technology Letters, IEEE,* vol. 5, pp. 776-778, 1993.

[50] C. Cook, J. E. Cunningham, A. Hargrove, G. G. Ger, K. W. Goossen, W. Y. Jan, H. H. Kim, R. Krause, M. Manges, M. Morrissey, M. Perinpanayagam, A. Persaud, G. J. Shevchuk, V. Sinyansky, and A. V. Krishnamoorthy, "A 36-channel parallel optical interconnect module based on optoelectronics-on-VLSI technology," *Selected Topics in Quantum Electronics, IEEE Journal of,* vol. 9, pp. 387-399, 2003.

[51] A. V. Krishnamoorthy and K. W. Goossen, "Optoelectronic-VLSI: photonics integrated with VLSI circuits," *Selected Topics in Quantum Electronics, IEEE Journal of,* vol. 4, pp. 899-912, 1998.

[52] N. M. Jokerst, M. A. Brooke, C. Sang-Yeon, S. Wilkinson, M. Vrazel, S. Fike, J. Tabler, J. Yoong Joon, S. Sang-Woo, D. S. Wills, and A. Brown, "The heterogeneous integration of optical interconnections into integrated microsystems," *Selected Topics in Quantum Electronics, IEEE Journal of,* vol. 9, pp. 350-360, 2003.

[53] B. Ciftcioglu, R. Berman, Z. Jian, Z. Darling, W. Shang, H. Jianyun, X. Jing, A. Garg, M. Jain, I. Savidis, D. Moore, M. Huang, E. G. Friedman, G. Wicks, and W. Hui, "A 3-D Integrated Intrachip Free-Space Optical Interconnect for Many-Core Chips," *Photonics Technology Letters, IEEE,* vol. 23, pp. 164-166, 2011.

[54] K. D. Maier, C. Beckstein, R. Blickhan, W. Erhard, and D. Fey, "A multi-layer-perceptron neural network hardware based on 3D massively parallel optoelectronic circuits," in *Parallel Interconnects, 1999. (PI '99) Proceedings. The 6th International Conference on*, 1999, pp. 73-80.

[55] J. Xue, A. Garg, B. Ciftcioglu, J. Hu, S. Wang, I. Savidis, M. Jain, R. Berman, P. Liu, M. Huang, H. Wu, E. Friedman, G. Wicks, and D. Moore, "An intra-chip free-space optical interconnect," *SIGARCH Comput. Archit. News,* vol. 38, pp. 94-105, 2010.

[56] J. D. Meindl, "Interconnect opportunities for gigascale integration," *Micro, IEEE,* vol. 23, pp. 28-35, 2003.

[57] X. Huikai, L. Erdmann, Z. Xu, K. J. Gabriel, and G. K. Fedder, "Post-CMOS processing for high-aspect-ratio integrated silicon microstructures," *Microelectromechanical Systems, Journal of,* vol. 11, pp. 93-101, 2002.

[58] D. Ching-Liang, X. Fu-Yuan, J. Ying-Zong, and C. Chin-Fon, "An approach to fabricating microstructures that incorporate circuits using a post-CMOS process," *Journal of Micromechanics and Microengineering,* vol. 15, p. 98, 2005.

[59] G. K. Fedder, R. T. Howe, L. Tsu-Jae King, and E. P. Quevy, "Technologies for Cofabricating MEMS and Electronics," *Proceedings of the Ieee*, vol. 96, pp. 306-322, 2008.

[60] D. Ching-Liang, "A capacitive humidity sensor integrated with micro heater and ring oscillator circuit fabricated by CMOS–MEMS technique," *Sensors and Actuators B: Chemical*, vol. 122, pp. 375-380, 2007.

[61] O. Brand, "Microsensor Integration Into Systems-on-Chip," *Proceedings of the Ieee*, vol. 94, pp. 1160-1176, 2006.

[62] J. J. Neumann, Jr. and K. J. Gabriel, "A fully-integrated CMOS-MEMS audio microphone," in *TRANSDUCERS, Solid-State Sensors, Actuators and Microsystems, 12th International Conference on, 2003*, 2003, pp. 230-233 vol.1.

[63] D. Whelan, "Chillin' Chips," in *Forbes*: Forbes, 2005.

[64] G. G. Shahidi, "SOI technology for the GHz era," *IBM J. Res. Dev.*, vol. 46, pp. 121-131, 2002.

[65] Y. Taur and T. K. Ning, *Fundamentals of Modern VLSI Devices*. New York: Cambride University Press, 1998.

[66] R. S. Muller and T. I. Kamins, *Device Electronics for Integrated Circuits*, 3rd ed. New York: John Wiley & Sons, 2002.

[67] S. E. Thompson, M. Armstrong, C. Auth, M. Alavi, M. Buehler, R. Chau, S. Cea, T. Ghani, G. Glass, T. Hoffman, C. H. Jan, C. Kenyon, J. Klaus, K. Kuhn, M. Zhiyong, B. McIntyre, K. Mistry, A. Murthy, B. Obradovic, R. Nagisetty, N. Phi, S. Sivakumar, R. Shaheed, L. Shifren, B. Tufts, S. Tyagi, M. Bohr, and Y. El-Mansy, "A 90-nm logic technology featuring strained-silicon," *Electron Devices, IEEE Transactions on*, vol. 51, pp. 1790-1797, 2004.

[68] P. Dumon, W. Bogaerts, V. Wiaux, J. Wouters, S. Beckx, J. V. Campenhout, D. Taillaert, B. Luyssaert, P. Bienstman, D. V. Thourhout, and R. Baets, "Low-loss SOI photonic wires and ring resonators fabricated with deep UV lithography," *Photonics Technology Letters, IEEE*, vol. 16, pp. 1328-1330, 2004.

[69] C. W. Holzwarth, J. S. Orcutt, L. Hanqing, M. A. Popovic, V. Stojanovic, J. L. Hoyt, R. J. Ram, and H. I. Smith, "Localized substrate removal technique enabling strong-confinement microphotonics in bulk Si CMOS processes," in *Conference on Lasers and Electro-Optics*, 2008, pp. 1-2.

[70] J. S. Orcutt, A. Khilo, C. W. Holzwarth, M. A. Popovi, H. Li, J. Sun, T. Bonifield, R. Hollingsworth, F. X. Kärtner, H. I. Smith, V. Stojanovi, and R. J. Ram, "Nanophotonic integration in state-of-the-art CMOS foundries," *Opt. Express*, vol. 19, pp. 2335-2346, 2011.

[71] S. K. Selvaraja, P. Jaenen, W. Bogaerts, D. V. Thourhout, P. Dumon, and R. Baets, "Fabrication of Photonic Wire and Crystal Circuits in Silicon-on-Insulator Using 193-nm Optical Lithography," *Journal of Lightwave Technology*, vol. 27, pp. 4076-4083, 2009.

[72] S. K. Selvaraja, W. Bogaerts, D. V. Thourhout, and R. Baets, "Fabrication of Uniform Photonic Devices Using 193nm Optical Lithography in Silicon-on-Insulator," *Proceedings of the 2008 European Conference on Integrated Optics*, pp. 359-362, 2008.

[73] Q. Xu, S. Manipatruni, B. Schmidt, J. Shakya, and M. Lipson, "12.5 Gbit/s carrier-injection-based silicon micro-ring silicon modulators," *Opt. Express*, vol. 15, pp. 430-436, 2007.

[74] Q. Xu, B. Schmidt, S. Pradhan, and M. Lipson, "Micrometre-scale silicon electro-optic modulator," *Nature*, vol. 435, pp. 325-327, 2005.

[75] F. N. Xia, L. Sekaric, and Y. Vlasov, "Ultracompact optical buffers on a silicon chip," *Nature Photonics*, vol. 1, pp. 65-71, Jan 2007.

[76] P. Dumon, W. Bogaerts, R. Baets, J. M. Fedeli, and L. Fulbert, "Towards foundry approach for silicon photonics: silicon photonics platform ePIXfab," *Electronics Letters*, pp. 13-14, Jun 2009.

[77] T. Y. Liow, K. W. Ang, Q. Fang, J. F. Song, Y. Z. Xiong, M. B. Yu, G. Q. Lo, and D. L. Kwong, "Silicon Modulators and Germanium Photodetectors on SOI: Monolithic Integration, Compatibility, and Performance Optimization," *Ieee Journal of Selected Topics in Quantum Electronics*, vol. 16, pp. 307-315, Jan-Feb 2010.

[78] 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," *2009 Conference on Lasers and Electro-Optics and Quantum Electronics and Laser Science Conference (Cleo/Qels 2009), Vols 1-5*, pp. 812-813, 2009.

[79] L. C. Kimerling, L. Dal Negro, S. Saini, Y. Yi, D. Ahn, S. Akiyama, D. Cannon, J. Liu, J. G. Sandland, D. Sparacin, J. Michel, K. Wada, and M. R. Watts, "Monolithic Silicon Microphotonics," *Silicon Photonics*, vol. 94, pp. 89-119, 2004.

[80] P. Dong, W. Qian, S. R. Liao, H. Liang, C. C. Kung, N. N. Feng, R. Shafiiha, J. A. Fong, D. Z. Feng, A. V. Krishnamoorthy, and M. Asghari, "Low loss shallow-ridge silicon waveguides," *Optics Express*, vol. 18, pp. 14474-14479, Jul 5 2010.

[81] M. Asghari, D. Feng, P. Zhou, B. J. Luff, J. E. Cunningham, and A. V. Krishnamoorthy, "Applications and challenges for silicon photonics in optical interconnects," *2007 Ieee Leos Annual Meeting Conference Proceedings, Vols 1 and 2*, pp. 476-477, 2007.

[82] M. Asghari, "Silicon photonics: A low cost integration platform for datacom and telecom applications," *2008 Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference, Vols 1-8*, pp. 847-856
3037, 2008.

[83] T. Ernst and S. Cristoloveanu, "Buried oxide fringing capacitance: a new physical model and itsimplication on SOI device scaling and architecture," *1999 IEEE International SOI Conference Proceedings*, pp. 38-39, 10/04/1999 - 10/07/1999 1999.

[84] T. Ernst, C. Tinella, C. Raynoud, and S. Cristoloveanu, "Fringing fields in sub-0.1 μm fully depleted SOI MOSFETs: optimization of the device architecture," *Solid State Electronics*, vol. 46, pp. 373-378, March 2002.

[85] N. Bresson, S. Cristoloveanu, C. Mazuré, F. Letertre, and H. Iwai, "Integration of buried insulators with high thermal conductivity in SOI MOSFETs: Thermal properties and short channel effects," *Solid State Electronics*, vol. 49, pp. 1522-1528, 2005.

[86] H. Wei, R. S. Mircea, S. Kevin, S. Karthik, G. Shougata, and V. Sivakumar, "Compact thermal modeling for temperature-aware design," in *Proceedings of the 41st annual Design Automation Conference* San Diego, CA, USA: ACM, 2004.

[87] Z. Xin and A. Q. Huang, "A novel distributed control and its tolerance analysis for microprocessor power management," in *Applied Power Electronics Conference and Exposition, 2006. APEC '06. Twenty-First Annual IEEE*, 2006, p. 7, 2006.

[88] K. A. Jenkins, J. Y. C. Sun, and J. Gautier, "Characteristics of SOI FET's under pulsed conditions," *Electron Devices, IEEE Transactions on*, vol. 44, pp. 1923-1930, 1997.

[89] S. Patil, "Distributed Model for Thermal Characterization of Oxide Isolated Silicon Germanium Heterojunction Bipolar Transistors." vol. Ph.D.: University of Texas Arlington, 2011.

[90] N. Sherwood-Droz, A. Gondarenko, and M. Lipson, "Oxidized Silicon-on-Insulator (OxSOI) from bulk silicon: a new photonic platform," *Opt. Express*, vol. 18, pp. 5785-5790, 2010.

[91] J. Witzens, G. Masini, S. Sahni, B. Analui, C. Gunn, and G. Capellini, "10Gbit/s transceiver on silicon," in *Silicon Photonics and Photonic Integrated Circuits*, Strasbourg, France, 2008, pp. 699610-10.

[92] M. R. Watts, W. A. Zortman, D. C. Trotter, R. W. Young, and A. L. Lentine, "Vertical junction silicon microdisk modulators and switches," *Opt. Express*, vol. 19, pp. 21989-22003, 2011.

[93] K. Preston, S. Manipatruni, A. Gondarenko, C. B. Poitras, and M. Lipson, "Deposited silicon high-speed integrated electro-optic modulator," *Optics Express*, vol. 17, pp. 5118-5124, Mar 30 2009.

[94] K. Preston, B. Schmidt, and M. Lipson, "Polysilicon photonic resonators for large-scale 3D integration of optical networks," *Optics Express*, vol. 15, pp. 17283-17290, Dec 10 2007.

[95] G. Masini, L. Colace, and G. Assanto, "2.5 Gbit/s polycrystalline germanium-on-silicon photodetector operating from 1.3 to 1.55 mu m," *Applied Physics Letters*, vol. 82, pp. 2524-2526, Apr 14 2003.

[96] S. Kalluri, M. Ziari, A. T. Chen, V. Chuyanov, W. H. Steier, D. T. Chen, B. Jalali, H. Fetterman, and L. R. Dalton, "Monolithic integration of waveguide polymer electrooptic modulators on VLSI circuitry," *Ieee Photonics Technology Letters*, vol. 8, pp. 644-646, May 1996.

[97] B. A. Block, T. R. Younkin, P. S. Davids, M. R. Reshotko, P. Chang, B. M. Polishak, S. Huang, J. D. Luo, and A. K. Y. Jen, "Electro-optic polymer cladding ring resonator modulators," *Optics Express*, vol. 16, pp. 18326-18333, Oct 27 2008.

[98] K. Furuya, Y. Sakakibara, K. Nakanishi, R. Takei, E. Itoga, M. Suzuki, M. Okano, T. Kamei, and M. Mori, "Fine thickness control of amorphous silicon by wet-etching for low loss wire waveguide," in *Group IV Photonics (GFP), 2011 8th IEEE International Conference on*, 2011, pp. 109-111.

[99] N. Sherwood-Droz and M. Lipson, "Multi-layer low-temperature deposited CMOS photonics for microelectronics backend integration," in *Lasers and Electro-Optics (CLEO), 2011 Conference on*, 2011, pp. 1-2.

[100] I. A. Young, E. M. Mohammed, J. T. S. Liao, A. M. Kern, S. Palermo, B. A. Block, M. R. Reshotko, and P. L. D. Chang, "Optical technology for energy efficient I/O in high performance computing," *Communications Magazine, IEEE*, vol. 48, pp. 184-191, 2010.

[101] K. Preston, S. Manipatruni, A. Gondarenko, C. B. Poitras, and M. Lipson, "Deposited silicon high-speed integratedelectro-optic modulator," *Opt. Express*, vol. 17, pp. 5118-5124, 2009.

[102] K. Preston, Y. H. D. Lee, M. Zhang, and M. Lipson, "Waveguide-integrated telecom-wavelength photodiode in deposited silicon," *Opt. Lett.*, vol. 36, pp. 52-54, 2011.

[103] I. A. Young, B. Block, M. Reshotko, and P. Chang, "Integration of nano-photonic devices for CMOS chip-to-chip optical I/O," in *Lasers and Electro-Optics (CLEO) and Quantum Electronics and Laser Science Conference (QELS), 2010 Conference on*, 2010, pp. 1-2.

[104] J. A. Kash, "Leveraging optical interconnects in future supercomputers and servers," *16th Annual IEEE Symposium on High-Performance Interconnects, Proceedings*, pp. 190-194, 2008.

[105] J. A. Kash, F. E. Doany, L. Schares, C. L. Schow, C. Schuster, D. M. Kuchta, P. K. Pepejugoski, J. M. Trewhella, C. W. Baks, R. A. John, L. Shan, Y. H. Kwark, R. A. Budd, P. Chiniwalla, F. R. Libsch, J. Rosner, C. K. Tsang, C. S. Patel, J. D. Schaub, D. Kucharski, D. Guckenherger, S. Hegde, H. Nyikal, R. Dangel, F. Horst, B. J. Offrein, C. K. Lin, A. Tandon, G. R. Trott, M. Nystrom, D. Bour, M. R. T. Tan, and D. W. Dolfi, "Chip-to-chip optical interconnects," *2006 Optical Fiber Communication Conference/National Fiber Optic Engineers Conference, Vols 1-6*, pp. 2316-2318, 2006.

[106] A. F. Benner, M. Ignatowski, J. A. Kash, D. M. Kuchta, and M. B. Ritter, "Exploitation of optical interconnects in future server architectures," *Ibm Journal of Research and Development*, vol. 49, pp. 755-775, Jul-Sep 2005.

[107] X. Zheng, J. Lexau, Y. Luo, H. Thacker, T. Pinguet, A. Mekis, G. Li, J. Shi, P. Amberg, N. Pinckney, K. Raj, R. Ho, J. E. Cunningham, and A. V. Krishnamoorthy, "Ultra-low-energy all-CMOS modulator integrated with driver," *Opt. Express*, vol. 18, pp. 3059-3070, 2010.

[108] X. Zheng, F. Liu, D. Patil, H. Thacker, Y. Luo, T. Pinguet, A. Mekis, J. Yao, G. Li, J. Shi, K. Raj, J. Lexau, E. Alon, R. Ho, J. E. Cunningham, and A. V. Krishnamoorthy, "A sub-picojoule-per-bit CMOS photonic receiver for densely integrated systems," *Opt. Express,* vol. 18, pp. 204-211, 2010.

[109] N. N. Feng, S. R. Liao, D. Z. Feng, P. Dong, D. W. Zheng, H. Liang, R. Shafiiha, G. L. Li, J. E. Cunningham, A. V. Krishnamoorthy, and M. Asghari, "High speed carrier-depletion modulators with 1.4V-cm V pi L integrated on 0.25 mu m silicon-on-insulator waveguides," *Optics Express,* vol. 18, pp. 7994-7999, Apr 12 2010.

[110] S. Liao, N.-N. Feng, D. Feng, P. Dong, R. Shafiiha, C.-C. Kung, H. Liang, W. Qian, Y. Liu, J. Fong, J. E. Cunningham, Y. Luo, and M. Asghari, "36 GHz submicron silicon waveguide germanium photodetector," *Opt. Express,* vol. 19, pp. 10967-10972, 2011.

[111] U. Menzel, A. Barwolff, P. Enders, D. Ackermann, R. Puchert, and M. Voss, "Modelling the temperature dependence of threshold current, external differential efficiency and lasing wavelength in QW laser diodes," *Semiconductor Science and Technology,* vol. 10, p. 1382, 1995.

[112] D. Botez, L. J. Mawst, A. Bhattacharya, J. Lopez, J. Li, T. F. Kuech, V. P. Iakovlev, G. I. Suruceanu, A. Caliman, and A. V. Syrbu, "66% CW wallplug efficiency from Al-free 0.98 μm-emitting diode lasers," *Electronics Letters,* vol. 32, pp. 2012-2013, 1996.

[113] Y. Arakawa and H. Sakaki, "Multidimensional quantum well laser and temperature dependence of its threshold current," *Applied Physics Letters,* vol. 40, pp. 939-941, 1982.

[114] D. F. Welch, "A brief history of high-power semiconductor lasers," *Selected Topics in Quantum Electronics, IEEE Journal of,* vol. 6, pp. 1470-1477, 2000.

[115] F. Xia, L. Sekaric, and Y. Vlasov, "Ultracompact optical buffers on a silicon chip," *Nat Photon,* vol. 1, pp. 65-71, 2007.

[116] J. Cardenas, C. B. Poitras, J. T. Robinson, K. Preston, L. Chen, and M. Lipson, "Low loss etchless silicon photonic waveguides," *Opt. Express,* vol. 17, pp. 4752-4757, 2009.

[117] M. P. Nezhad, O. Bondarenko, M. Khajavikhan, A. Simic, and Y. Fainman, "Etch-free low loss silicon waveguides using hydrogen silsesquioxane oxidation masks," *Opt. Express,* vol. 19, pp. 18827-18832, 2011.

[118] P. Dong, W. Qian, S. Liao, H. Liang, C.-C. Kung, N.-N. Feng, R. Shafiiha, J. Fong, D. Feng, A. V. Krishnamoorthy, and M. Asghari, "Low loss shallow-ridge silicon waveguides," *Opt. Express,* vol. 18, pp. 14474-14479, 2010.

[119] T. D. Visser, H. Blok, B. Demeulenaere, and D. Lenstra, "Confinement factors and gain in optical amplifiers," *Quantum Electronics, IEEE Journal of,* vol. 33, pp. 1763-1766, 1997.

[120] K. M. Karan, S. O. Jason, and J. R. Rajeev, "Enhanced Two-Photon Absorption in Polycrystalline Silicon," 2011, p. JThB86.

[121] A. P. Milos, B. Tymon, P. I. Erich, and X. K. r. Franz, "Global Design Rules for Silicon Microphotonic Waveguides: Sensitivity, Polarization and Resonance Tunability," 2006, p. CTuCC1.

[122] L. Vivien, D. Pascal, S. Lardenois, D. Marris-Morini, E. Cassan, F. Grillot, S. Laval, J. M. Fedeli, and M. Loubna El, "Light injection in SOI microwaveguides using high-efficiency grating couplers," *Lightwave Technology, Journal of*, vol. 24, pp. 3810-3815, 2006.

[123] D. Taillaert, W. Bogaerts, P. Bienstman, T. F. Krauss, P. Van Daele, I. Moerman, S. Verstuyft, K. De Mesel, and R. Baets, "An out-of-plane grating coupler for efficient butt-coupling between compact planar waveguides and single-mode fibers," *Quantum Electronics, IEEE Journal of*, vol. 38, pp. 949-955, 2002.

[124] B. Schmid, A. Petrov, and M. Eich, "Optimized grating coupler with fully etched slots," *Optics Express*, vol. 17, pp. 11066-11076, Jun 2009.

[125] W. Bogaerts, D. Taillaert, B. Luyssaert, P. Dumon, J. Van Campenhout, P. Bienstman, D. Van Thourhout, R. Baets, V. Wiaux, and S. Beckx, "Basic structures for photonic integrated circuits in silicon-on-insulator," *Opt. Express*, vol. 12, pp. 1583-1591, 2004.

[126] S. K. Selvaraja, D. Vermeulen, M. Schaekers, E. Sleeckx, W. Bogaerts, G. Roelkens, P. Dumon, D. Van Thourhout, and R. Baets, "Highly efficient grating coupler between optical fiber and silicon photonic circuit," in *Lasers and Electro-Optics, 2009 and 2009 Conference on Quantum electronics and Laser Science Conference. CLEO/QELS 2009. Conference on*, 2009, pp. 1-2.

[127] C. Xia, L. Chao, C. K. Y. Fung, S. M. G. Lo, and H. K. Tsang, "Apodized Waveguide Grating Couplers for Efficient Coupling to Optical Fibers," *Photonics Technology Letters, IEEE*, vol. 22, pp. 1156-1158, 2010.

[128] X. Shijun, M. H. Khan, S. Hao, and Q. Minghao, "Silicon-on-Insulator Microring Add-Drop Filters With Free Spectral Ranges Over 30 nm," *Lightwave Technology, Journal of*, vol. 26, pp. 228-236, 2008.

[129] N. N. Feng, D. Feng, H. Liang, W. Qian, C. C. Kung, J. Fong, and M. Asghari, "Low-Loss Polarization-Insensitive Silicon-on-Insulator-Based WDM Filter for Triplexer Applications," *Ieee Photonics Technology Letters*, vol. 20, pp. 1968-1970, Nov-Dec 2008.

[130] T. Barwicz, M. A. Popovic, M. R. Watts, P. T. Rakich, E. P. Ippen, and H. I. Smith, "Fabrication of add-drop filters based on frequency-matched microring resonators," *Lightwave Technology, Journal of*, vol. 24, pp. 2207-2218, 2006.

[131] W. Bogaerts, P. Dumon, D. V. Thourhout, D. Taillaert, P. Jaenen, J. Wouters, S. Beckx, V. Wiaux, and R. Baets, "Compact Wavelength-Selective Functions in Silicon-on-Insulator Photonic Wires," *Journal of Selected Topics in Quantum Electronics*, vol. 12, pp. 1394-1401, 2006.

[132] R. Soref and B. Bennet, "Electrooptical effects in silicon," *Quantum Electronics, IEEE Journal of*, vol. 23, pp. 123- 129, Jan 1987.

[133] D. W. Zheng, B. T. Smith, and M. Asghari, "Efficient Silicon-photonic modulator with recessed electrodes - art. no. 68980A," *Silicon Photonics Iii*, vol. 6898, pp. A8980-A8980, 2008.

[134] K. Preston, S. Manipatruni, C. B. Poitras, and M. Lipson, "2.5 Gbps Electro-optic Modulator in Deposited Silicon," *2009 Conference on Lasers and Electro-Optics and Quantum Electronics and Laser Science Conference (Cleo/Qels 2009), Vols 1-5*, pp. 1273-1274, 2009.

[135] P. Dong, R. Shafiiha, S. R. Liao, H. Liang, N. N. Feng, D. Z. Feng, G. L. Li, X. Z. Zheng, A. V. Krishnamoorthy, and M. Asghari, "Wavelength-tunable silicon microring modulator," *Optics Express*, vol. 18, pp. 10941-10946, May 24 2010.

[136] C. A. Barrios, V. R. de Almeida, and M. Lipson, "Low-power-consumption short-length and high-modulation-depth silicon electrooptic modulator," *Lightwave Technology, Journal of*, vol. 21, pp. 1089--1098, 2003.

[137] S. M. Sze, *Semiconductor devices: physics and technology*: Wiley-India, 2009.

[138] J. Bravo-Abad, E. P. Ippen, and M. Soljačić, "Ultrafast photodetection in an all-silicon chip enabled by two-photon absorption," *Applied Physics Letters*, vol. 94, p. 241103, 2009.

[139] M. W. Geis, S. J. Spector, M. E. Grein, R. T. Schulein, J. U. Yoon, D. M. Lennon, S. Deneault, F. Gan, F. X. Kaertner, and T. M. Lyszczarz, "CMOS-Compatible All-Si High-Speed Waveguide Photodiodes with High Responsivity in Near-Infrared Communication Band," *Photonics Technology Letters, IEEE*, vol. 19, pp. 152-154, 2007.

[140] D. Brooks, R. P. Dick, R. Joseph, and L. Shang, "Power, Thermal, and Reliability Modeling in Nanometer-Scale Microprocessors," *Ieee Micro*, vol. 27, pp. 49-62, 2007.

[141] N. Magen, A. Kolodny, U. Weiser, and N. Shamir, "Interconnect-power dissipation in a microprocessor," p. 7, 2004.

[142] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch," *ACM SIGARCH Computer Architecture News*, vol. 28, pp. 83-94, 2000.

[143] P. E. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon, "High-performance microprocessor design," *Ieee Journal of Solid-State Circuits*, vol. 33, pp. 676-686, 1998.

[144] V. Tiwari, D. Singh, S. Rajgopal, G. Mehta, R. Patel, and F. Baez, "Reducing power in high-performance microprocessors," pp. 732-737, 1998.

[145] T. H. Dunigan, Jr., J. S. Vetter, J. B. White, III, and P. H. Worley, "Performance evaluation of the Cray X1 distributed shared-memory architecture," *Micro, IEEE*, vol. 25, pp. 30-40, 2005.

[146] R. Espasa, F. Ardanaz, J. Emer, S. Felix, J. Gago, R. Gramunt, I. Hernandez, T. Juan, G. Lowney, M. Mattina, and A. Seznec, "Tarantula: a vector extension to the alpha

architecture," in *Computer Architecture, 2002. Proceedings. 29th Annual International Symposium on*, 2002, pp. 281-292.

[147] S. R. Alam and J. S. Vetter, "An Analysis of System Balance Requirements for Scientific Applications," in *Parallel Processing, 2006. ICPP 2006. International Conference on*, 2006, pp. 229-236.

[148] P. M. Kogge, "An exploration of the technology space for multi-core memory/logic chips for highly scalable parallel systems," in *Innovative Architecture for Future Generation High-Performance Processors and Systems, 2005*, 2005, p. 10 pp.

[149] L. Christophe, S. Jack, F. Jean, and J. Norm, "The Potential Energy Efficiency of Vector Acceleration," in *SC 2006 Conference, Proceedings of the ACM/IEEE*, 2006, pp. 1-1.

[150] X. Li, J. Palma, P. Amestoy, M. Dayd, M. Mattoso, and J. o. Lopes, "Evaluation of Sparse LU Factorization and Triangular Solution on Multicore Platforms High Performance Computing for Computational Science - VECPAR 2008." vol. 5336: Springer Berlin / Heidelberg, 2008, pp. 287-300.

[151] J. Chang, S.-L. Chen, W. Chen, S. Chiu, R. Faber, R. Ganesan, M. Grgek, V. Lukka, W. W. Mar, J. Vash, S. Rusu, and K. Zhang, "A 45nm 24MB on-die L3 cache for the 8-core multi-threaded Xeon<sup>®</sup> Processor," in *VLSI Circuits, 2009 Symposium on*, 2009, pp. 152-153.

[152] S. Rusu, S. Tam, H. Muljono, J. Stinson, D. Ayers, J. Chang, R. Varada, M. Ratta, S. Kottapalli, and S. Vora, "A 45 nm 8-Core Enterprise Xeon¯ Processor," *Solid-State Circuits, IEEE Journal of*, vol. 45, pp. 7-14, 2010.

[153] P. Gepner, D. Fraser, M. Kowalik, R. Wyrzykowski, J. Dongarra, K. Karczewski, and J. Wasniewski, "Evaluating Performance of New Quad-Core IntelÂ®XeonÂ®5500 Family Processors for HPC Parallel Processing and Applied Mathematics." vol. 6067: Springer Berlin / Heidelberg, 2010, pp. 1-10.

[154] S. Sawant, U. Desai, G. Shamanna, L. Sharma, M. Ranade, A. Agarwal, S. Dakshinamurthy, and R. Narayanan, "A 32nm Westmere-EX Xeon<sup>®</sup> enterprise processor," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, 2011, pp. 74-75.

[155] R. Kuppuswamy, S. R. Sawant, S. Balasubramanian, P. Kaushik, N. Natarajan, and J. D. Gilbert, "Over one million TPCC with a 45nm 6-core Xeon® CPU," in *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, 2009, pp. 70-71,71a.

[156] Defense Advanced Research Project Agency, "https://www.fbo.gov/index?s=opportunity&mode=form&id=cbc05c86eb555a334708b570564dddca&tab=core&_cview=0." vol. 2011, 2011.

[157] F. O'Mahony, G. Balamurugan, J. E. Jaussi, J. Kennedy, M. Mansuri, S. Shekhar, and B. Casper, "The future of electrical I/O for microprocessors," pp. 31-34, 2009.

[158] J. H. Lau, *Low cost flip chip technologies : for DCA, WLCSP, and PBGA assemblies*. New York: McGraw-Hill, 2000.

[159] M. Datta, T. Osaka, and Y. Shacham-Diamand, "Microelectronic Packaging Trends and the Role of Nanotechnology Electrochemical Nanotechnologies," Springer New York, 2010, pp. 227-253.

[160] T. Tekin, "Review of Packaging of Optoelectronic, Photonic, and MEMS Components," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 17, pp. 704-719, 2011.

[161] P. Harvey, Y. Zhou, G. Yamada, D. Questad, G. Lafontant, R. Mandrekar, S. Suminaga, Y. Yamaji, H. Noma, T. Nishio, H. Mori, T. Tamura, K. Yazawa, Takiguchi, T. Ohde, R. White, A. Malhotra, J. Audet, J. Wakil, W. Sauter, and E. Hosomi, "Chip/Package Design and Technology Trade-offs in the 65nm Cell Broadband Engine," in *Electronic Components and Technology Conference, 2007. ECTC '07. Proceedings. 57th*, 2007, pp. 27-34.

[162] K. Gilleo, *Area array package design : techniques in high-density electronics*. New York: McGraw-Hill, 2004.

[163] K. Chang, L. Haechang, W. Ting, K. Kaviani, K. Prabhu, W. Beyene, N. Chan, C. Chen, T. J. Chin, A. Gupta, C. Madden, Mahabaleshwara, L. Raghavan, S. Jie, and S. Xudong, "An 8Gb/s/link, 6.5mW/Gb/s memory interface with bimodal request bus," in *Solid-State Circuits Conference, 2009. A-SSCC 2009. IEEE Asian*, 2009, pp. 21-24.

[164] B. Leibowitz, R. Palmer, J. Poulton, Y. Frans, S. Li, J. Wilson, M. Bucher, A. M. Fuller, J. Eyles, M. Aleksic, T. Greer, and N. M. Nguyen, "A 4.3 GB/s Mobile Memory Interface With Power-Efficient Bandwidth Scaling," *Solid-State Circuits, IEEE Journal of*, vol. 45, pp. 889-898, 2010.

[165] J. Poulton, R. Palmer, A. M. Fuller, T. Greer, J. Eyles, W. J. Dally, and M. Horowitz, "A 14-mW 6.25-Gb/s Transceiver in 90-nm CMOS," *Ieee Journal of Solid-State Circuits*, vol. 42, pp. 2745-2757, 2007.

[166] M. Kossel, C. Menolfi, J. Weiss, P. Buchmann, G. von Bueren, L. Rodoni, T. Morf, T. Toifl, and M. Schmatz, "A T-Coil-Enhanced 8.5 Gb/s High-Swing SST Transmitter in 65 nm Bulk CMOS With >16 dB Return Loss Over 10 GHz Bandwidth," *Ieee Journal of Solid-State Circuits*, vol. 43, pp. 2905-2920, 2008.

[167] R. Palmer, J. Poulton, A. Fuller, J. Chen, and J. Zerbe, "Design considerations for low-power high-performance mobile logic and memory interfaces," pp. 205-208, 2008.

[168] K. Hu, T. Jiang, J. Wang, F. O'Mahony, and P. Y. Chiang, "A 0.6 mW/Gb/s, 6.4–7.2 Gb/s Serial Link Receiver Using Local Injection-Locked Ring Oscillators in 90 nm CMOS," *Ieee Journal of Solid-State Circuits*, vol. 45, pp. 899-908, 2010.

[169] K. Hu, T. Jiang, J. Wang, F. O'Mahony, and P. Y. Chiang, "A 0.6mW/Gbps, 6.4-8.0Gbps serial link receiver using local injection-locked ring oscillators in 90nm CMOS," in *VLSI Circuits, 2009 Symposium on*, 2009, pp. 46-47.

[170] H. Kangmin, J. Tao, W. Jingguang, F. O'Mahony, and P. Y. Chiang, "A 0.6 mW/Gb/s, 6.4-7.2 Gb/s Serial Link Receiver Using Local Injection-Locked Ring Oscillators in 90 nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 45, pp. 899-908, 2010.

[171] H. Kangmin, J. Tao, S. Palermo, and P. Y. Chiang, "Low-power 8Gb/s near-threshold serial link receivers using super-harmonic injection locking in 65nm CMOS," in *Custom Integrated Circuits Conference (CICC), 2011 IEEE*, 2011, pp. 1-4.

[172] C. Dragone, C. H. Henry, I. P. Kaminow, and R. C. Kistler, "Efficient multichannel integrated optics star coupler on silicon," *Photonics Technology Letters, IEEE*, vol. 1, pp. 241-243, 1989.

[173] P. D. Trinh, S. Yegnanarayanan, and B. Jalali, "5x9 integrated optical star coupler in silicon-on-insulator technology," *Photonics Technology Letters, IEEE*, vol. 8, pp. 794-796, 1996.

[174] K. Okamoto, H. Takahashi, M. Yasu, and Y. Hibino, "Fabrication of wavelength-insensitive 8×8 star coupler," *Photonics Technology Letters, IEEE*, vol. 4, pp. 61-63, 1992.

[175] Y. Sawada, K. Harada, and H. Fujioka, "Study of package warp behavior for high-performance flip-chip BGA," *Microelectronics Reliability*, vol. 43, pp. 465-471, 2003.

[176] T. Connectivity, "Intrafacility Fiber Cable Assemblies." vol. 2011, 2011.

[177] F. Liu, D. Patil, J. Lexau, P. Amberg, M. Dayringer, J. Gainsley, H. F. Moghadam, Z. Xuezhe, J. E. Cunningham, A. V. Krishnamoorthy, E. Alon, and R. Ho, "10 Gbps, 530 fJ/b optical transceiver circuits in 40 nm CMOS," in *VLSI Circuits (VLSIC), 2011 Symposium on*, 2011, pp. 290-291.

[178] X. Zheng, D. Patil, J. Lexau, F. Liu, G. Li, H. Thacker, Y. Luo, I. Shubin, J. Li, J. Yao, P. Dong, D. Feng, M. Asghari, T. Pinguet, A. Mekis, P. Amberg, M. Dayringer, J. Gainsley, H. F. Moghadam, E. Alon, K. Raj, R. Ho, J. E. Cunningham, and A. V. Krishnamoorthy, "Ultra-efficient 10Gb/s hybrid integrated silicon photonic transmitter and receiver," *Opt. Express*, vol. 19, pp. 5172-5186, 2011.

[179] M. Georgas, J. Orcutt, R. J. Ram, and V. Stojanovic, "A monolithically-integrated optical receiver in standard 45-nm SOI," in *ESSCIRC (ESSCIRC), 2011 Proceedings of the*, 2011, pp. 407-410.

[180] T. Takemoto, F. Yuuki, H. Yamashita, T. Ban, M. Kono, Y. Lee, T. Saito, S. Tsuji, and S. Nishimura, "A 25-Gb/s, 2.8-mW/Gb/s low power CMOS optical receiver for 100-Gb/s Ethernet solution," in *Optical Communication, 2009. ECOC '09. 35th European Conference on*, 2009, pp. 1-2.

[181] J. Youn, M. Lee, K. Park, and W. Choi, "A 10-Gb/s 850-nm CMOS OEIC Receiver with a Silicon Avalanche Photodetector," *Quantum Electronics, IEEE Journal of*, vol. PP, pp. 1-1, 2011.

[182] L. Dongmyung, H. Jungwon, H. Gunhee, and P. Sung Min, "An 8.5-Gb/s Fully Integrated CMOS Optoelectronic Receiver Using Slope-Detection Adaptive Equalizer," *Solid-State Circuits, IEEE Journal of*, vol. 45, pp. 2861-2873, 2010.

[183] T. S. C. Kao, F. A. Musa, and A. C. Carusone, "A 5-Gbit/s CMOS Optical Receiver With Integrated Spatially Modulated Light Detector and Equalization," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 57, pp. 2844-2857, 2010.

[184] H. Shih-Hao, C. Wei-Zen, C. Yu-Wei, and H. Yang-Tung, "A 10-Gb/s OEIC with Meshed Spatially-Modulated Photo Detector in 0.18-um CMOS Technology," *Solid-State Circuits, IEEE Journal of*, vol. 46, pp. 1158-1169, 2011.

[185] C. T. DeRose, D. C. Trotter, W. A. Zortman, A. L. Starbuck, M. Fisher, M. R. Watts, and P. S. Davids, "Ultra compact 45 GHz CMOS compatible Germanium waveguide photodiode with low dark current," *Opt. Express*, vol. 19, pp. 24897-24904, 2011.

[186] M. Gianlorenzo, C. Giovanni, W. Jeremy, and G. Cary, "A Four-Channel, 10 Gbps Monolithic Optical Receiver in 130nm CMOS with Integrated Ge Waveguide Photodetectors," in *Optical Fiber Communication Conference and Exposition and The National Fiber Optic Engineers Conference*, 2007, p. PDP31.

[187] D. Kucharski, D. Guckenberger, G. Masini, S. Abdalla, J. Witzens, and S. Sahni, "10Gb/s 15mW optical receiver with integrated Germanium photodetector and hybrid inductor peaking in 0.13µm SOI CMOS technology," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, 2010, pp. 360-361.

[188] S. Assefa, F. Xia, S. W. Bedell, Y. Zhang, T. Topuria, P. M. Rice, and Y. A. Vlasov, "CMOS-integrated high-speed MSM germanium waveguide photodetector," *Opt. Express*, vol. 18, pp. 4986-4999, 2010.

[189] G. L. Benjamin, A. Solomon, S. Clint, M. G. William, R. Alexander, A. J. Richard, A. K. Jeffrey, and A. V. Yurii, "Hybrid-Integrated Germanium Photodetector and CMOS Receiver Operating at 15 Gb/s," in *CLEO:2011 - Laser Applications to Photonic Applications*, 2011, p. CFB4.

[190] A. Solomon, G. L. Benjamin, S. Clint, M. G. William, R. Alexander, A. J. Richard, and A. V. Yurii, "20Gbps Receiver Based on Germanium Photodetector Hybrid-Integrated with 90nm CMOS Amplifier," in *CLEO:2011 - Laser Applications to Photonic Applications*, 2011, p. PDPB11.

[191] A. Emami-Neyestanak, D. Liu, G. Keeler, N. Helman, and M. Horowitz, "A 1.6 Gb/s, 3 mW CMOS receiver for optical communication," in *VLSI Circuits Digest of Technical Papers, 2002. Symposium on*, 2002, pp. 84-87.

[192] B. Casper and F. O'Mahony, "Clocking Analysis, Implementation and Measurement Techniques for High-Speed Data Links—A Tutorial," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 56, pp. 17-39, 2009.

[193] F.-T. Chen, M.-S. Kao, Y.-H. Hsu, C.-H. Lin, J.-M. Wu, C.-T. Chiu, and S.-H. Hsu, "A 10 to 11.5GHz rotational phase and frequency detector for clock recovery circuit," in

*Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, 2011, pp. 185-188.

[194] J. Savoj and B. Razavi, "A 10-Gb/s CMOS clock and data recovery circuit with a half-rate binary phase/frequency detector," *Solid-State Circuits, IEEE Journal of*, vol. 38, pp. 13-21, 2003.

[195] J. Leu, "A 9GHz Injection Locked Loop Optical Clock Receiver in 32-nm CMOS," in *Electrical Engineering and Computer Science*. vol. M.S. Cambridge, MA: Massachusetts Institute of Technology, 2010.

[196] Mitsubishi, "ML7XX11 Series." vol. 2011, 2004.

[197] A. Gubenko, I. Krestnikov, D. Livshtis, S. Mikhrin, A. Kovsh, L. West, C. Bornholdt, N. Grote, and A. Zhukov, "Error-free 10 Gbit/s transmission using individual Fabry-Perot modes of low-noise quantum-dot laser," *Electronics Letters*, vol. 43, p. 1430, 2007.

[198] M. Georgas, J. Leu, B. Moss, S. Chen, and V. Stojanovic, "Addressing link-level design tradeoffs for integrated photonic interconnects," in *Custom Integrated Circuits Conference (CICC), 2011 IEEE*, 2011, pp. 1-8.

[199] M. Kanskar, T. Earles, T. J. Goodnough, E. Stiers, D. Botez, and L. J. Mawst, "73% CW power conversion efficiency at 50 W from 970 nm diode laser bars," *Electronics Letters*, vol. 41, pp. 245-247, 2005.

[200] A. Knigge, G. Erbert, J. Jonsson, W. Pittroff, R. Staske, B. Sumpf, M. Weyers, and G. Trankle, "Passively cooled 940 nm laser bars with 73% wall-plug efficiency at 70 W and 25°C," *Electronics Letters*, vol. 41, pp. 250-251, 2005.

[201] L. Lin, L. Guojun, L. Zhanguo, L. Mei, L. Hui, W. Xiaohua, and W. Chunming, "High-Efficiency 808-nm InGaAlA/AlGaAs Double-Quantum-Well Semiconductor Lasers With Asymmetric Waveguide Structures," *Photonics Technology Letters, IEEE*, vol. 20, pp. 566-568, 2008.

[202] P. Crump, H. Wenzel, G. Erbert, P. Ressel, M. Zorn, F. Bugge, S. Einfeldt, R. Staske, U. Zeimer, A. Pietrzak, and G. Trankle, "Passively Cooled TM Polarized 808-nm Laser Bars With 70% Power Conversion at 80-W and 55-W Peak Power per 100-um Stripe Width," *Photonics Technology Letters, IEEE*, vol. 20, pp. 1378-1380, 2008.

[203] L. Hanxuan, T. Towe, I. Chyr, D. Brown, N. Touyen, F. Reinhardt, J. Xu, R. Srinivasan, M. Berube, T. Truchan, R. Bullock, and J. Harrison, "Near 1 kW of Continuous-Wave Power From a Single High-Efficiency Diode-Laser Bar," *Photonics Technology Letters, IEEE*, vol. 19, pp. 960-962, 2007.

[204] P. Crump, W. Dong, M. Grimshaw, J. Wang, S. Patterson, D. Wise, M. DeFranza, S. Elim, S. Zhang, M. Bougher, J. Patterson, S. Das, J. Bell, J. Farmer, M. DeVito, and R. Martinsen, "100-W+ diode laser bars show > 71% power conversion from 790- nm to 1000-nm and have clear route to > 85%," *Proceedings of SPIE*, vol. 6456, p. 64560M, 2007.

[205] J. Heerlein, S. Gruber, M. Grabherr, R. Jager, and P. Unger, "High-efficiency laterally-oxidized InGaAs-AlGaAs single-mode lasers," in *Semiconductor Laser Conference, 1998. ISLC 1998 NARA. 1998 IEEE 16th International*, 1998, pp. 219-220.

[206] Finisar, "10Gb Oxide Isolated VCSEL Reliability Report." vol. 2011, 2007.

[207] Intel, "Intel Server Board S3420GP Calculated MTBF Estimates." vol. 2011, 2009.

[208] Intel, "Intel Server Board S1200BTL Calculated MTBF Estimates." vol. 2011, 2011.

[209] B. M. Hawkins, R. A. Hawthorne, III, J. K. Guenter, J. A. Tatum, and J. R. Biard, "Reliability of various size oxide aperture VCSELs," in *Electronic Components and Technology Conference, 2002. Proceedings. 52nd*, 2002, pp. 540-550.

[210] CyOptics, "Laser & detector chip & TO-can products." vol. 2011, 2011.

[211] C. H. Joyner, J. L. Pleumeekers, A. Mathur, P. W. Evans, D. J. H. Lambert, S. Murthy, S. K. Mathis, F. H. Peters, J. Baeck, M. J. Missey, A. G. Dentai, R. A. Salvatore, R. P. Schneider, M. Ziari, M. Kato, R. Nagarajan, J. S. Bostak, T. Butrie, V. G. Dominic, M. Kauffman, R. H. Miles, M. L. Mitchell, A. C. Nilsson, S. C. Pennypacker, R. Schlenker, R. B. Taylor, T. Huan-Shang, M. F. Van Leeuwen, J. Webjorn, D. Perkins, J. Singh, S. G. Grubb, M. Reffle, D. G. Mehuys, F. A. Kish, and D. F. Welch, "Large-scale DWDM photonic integrated circuits: a manufacturable and scalable integration platform," in *Lasers and Electro-Optics Society, 2005. LEOS 2005. The 18th Annual Meeting of the IEEE*, 2005, pp. 344-345.

[212] J. S. Levy, A. Gondarenko, M. A. Foster, A. C. Turner-Foster, A. L. Gaeta, and M. Lipson, "CMOS-compatible multiple-wavelength oscillator for on-chip optical interconnects," *Nat Photon*, vol. 4, pp. 37-40, 2010.

[213] S. Beamer, C. Sun, Y.-J. Kwon, A. Joshi, C. Batten, V. Stojanovi\, \#263, K. Asanovi\, and \#263, "Re-architecting DRAM memory systems with monolithically integrated silicon photonics," *SIGARCH Comput. Archit. News*, vol. 38, pp. 129-140, 2010.

[214] S. Beamer, K. Asanovi\, \#263, C. Batten, A. Joshi, V. Stojanovi\, and \#263, "Designing multi-socket systems using silicon photonics," in *Proceedings of the 23rd international conference on Supercomputing* Yorktown Heights, NY, USA: ACM, 2009, pp. 521-522.

[215] R. Kumar and G. Hinton, "A family of 45nm IA processors," in *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, 2009, pp. 58-59.

[216] A. Shacham, K. Bergman, and L. P. Carloni, "On the Design of a Photonic Network-on-Chip," in *Proceedings of the First International Symposium on Networks-on-Chip*: IEEE Computer Society, 2007, pp. 53-64.

[217] P. Koka, M. O. McCracken, H. Schwetman, X. Zheng, R. Ho, and A. V. Krishnamoorthy, "Silicon-photonic network architectures for scalable, power-efficient multi-chip systems," *SIGARCH Comput. Archit. News*, vol. 38, pp. 117-128, 2010.

[218] W. Howard, M. Petracca, A. Biberman, B. G. Lee, L. P. Carloni, and K. Bergman, "Nanophotonic Optical Interconnection Network Architecture for On-Chip and Off-Chip Communications," in *Optical Fiber communication/National Fiber Optic Engineers Conference, 2008. OFC/NFOEC 2008. Conference on*, 2008, pp. 1-3.

[219] A. K. Kodi, R. Morris, A. Louri, and Z. Xiang, "On-Chip photonic interconnects for scalable multi-core architectures," in *Networks-on-Chip, 2009. NoCS 2009. 3rd ACM/IEEE International Symposium on*, 2009, pp. 90-90.

[220] K. Jeong-Taek, "CAD for nanometer silicon design challenges and success," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 12, pp. 1132-1147, 2004.

[221] E. D. Palik, *Handbook of optical constants of solids*. Orlando: Academic Press, 1985.

[222] T. Jhaveri, I. Stobert, L. Liebmann, P. Karakatsanis, V. Rovner, A. Strojwas, and L. Pileggi, "OPC simplification and mask cost reduction using regular design fabrics," in *Optical Microlithography XXII*, San Jose, CA, USA, 2009, pp. 727417-8.

[223] J. S. Orcutt, A. Khilo, M. A. Popovic, C. W. Holzwarth, B. Moss, L. Hanqing, M. S. Dahlem, T. D. Bonifield, F. X. Kartner, E. P. Ippen, J. L. Hoyt, R. J. Ram, and V. Stojanovic, "Demonstration of an electronic photonic integrated circuit in a commercial scaled bulk CMOS process," in *2008 Conference on Quantum Electronics and Laser Science. CLEO/QELS 2008*, pp. 1-2.

[224] T. Barwicz and H. A. Haus, "Three-dimensional analysis of scattering losses due to sidewall roughness in microphotonic waveguides," *Lightwave Technology, Journal of*, vol. 23, pp. 2719-2732, 2005.

[225] G. Baccarani, B. Ricco, and G. Spadini, "Transport properties of polycrystalline silicon films," *Journal of Applied Physics*, vol. 49, pp. 5565-5570, 1978.

[226] W. J. Westerveld, H. P. Urbach, and M. Yousefi, "Optimized 3-D Simulation Method for Modeling Out-of-Plane Radiation in Silicon Photonic Integrated Circuits," *Quantum Electronics, IEEE Journal of*, vol. 47, pp. 561-568, 2011.

[227] J. V. Hryniewicz, P. P. Absil, B. E. Little, R. A. Wilson, and P. T. Ho, "Higher order filter response in coupled microring resonators," *Photonics Technology Letters, IEEE*, vol. 12, pp. 320-322, 2000.

[228] A. Melloni and M. Martinelli, "Synthesis of direct-coupled-resonators bandpass filters for WDM systems," *Lightwave Technology, Journal of*, vol. 20, pp. 296-303, 2002.

[229] C. W. Holzwarth, T. Barwicz, M. A. Popovic, P. T. Rakich, E. P. Ippen, F. X. Kartner, and H. I. Smith, "Accurate resonant frequency spacing of microring filters without postfabrication trimming," *Journal of Vacuum Science and Technology B*, vol. 24, pp. 3244-3247, November 2006.

[230] J. Sun, C. W. Holzwarth, M. Dahlem, J. T. Hastings, and H. I. Smith, "Accurate frequency alignment in fabrication of high-order microring-resonator filters," *Opt. Express*, vol. 16, pp. 15958-15963, 2008.

[231] P. Dong, W. Qian, H. Liang, R. Shafiiha, N.-N. Feng, D. Feng, X. Zheng, A. V. Krishnamoorthy, and M. Asghari, "Low power and compact reconfigurable multiplexing devices based on silicon microring resonators," *Opt. Express*, vol. 18, pp. 9852-9858, 2010.

[232] P. Dong, W. Qian, H. Liang, R. Shafiiha, X. Wang, D. Feng, G. Li, J. E. Cunningham, A. V. Krishnamoorthy, and M. Asghari, "1x4 reconfigurable demultiplexing filter based on free-standing silicon racetrack resonators," *Opt. Express*, vol. 18, pp. 24504-24509, 2010.

[233] 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 *Conference on Lasers and Electro-Optics*, 2009, pp. 1-2.

[234] L. Sungjae, B. Jagannathan, S. Narasimha, A. Chou, N. Zamdmer, J. Johnson, R. Williams, L. Wagner, K. Jonghae, J. O. Plouchart, J. Pekarik, S. Springer, and G. Freeman, "Record RF performance of 45-nm SOI CMOS Technology," in *Electron Devices Meeting, 2007. IEDM 2007. IEEE International*, 2007, pp. 255-258.

[235] W. M. Green, M. J. Rooks, L. Sekaric, and Y. A. Vlasov, "Ultra-compact, low RF power, 10 Gb/s siliconMach-Zehnder modulator," *Opt. Express*, vol. 15, pp. 17106-17113, 2007.

[236] C. W. Holzwarth, R. Amatya, M. Araghchini, J. Birge, H. Byun, J. Chen, M. Dahlem, N. A. DiLello, F. Gan, J. L. Hoyt, E. P. Ippen, F. X. Kartner, A. Khilo, J. Kim, M. Kim, A. Motamedi, J. S. Orcutt, M. Park, M. Perrott, M. A. Popovic, R. J. Ram, H. I. Smith, G. R. Zhou, S. J. Spector, T. M. Lyszczarz, M. W. Geis, D. M. Lennon, J. U. Yoon, M. E. Grein, R. T. Schulein, S. Frolov, A. Hanjani, and J. Shmulovich, "High speed analog-to-digital conversion with silicon photonics," in *Silicon Photonics IV*, San Jose, CA, USA, 2009, pp. 72200B-15.

[237] A. Liu, R. Jones, L. Liao, D. Samara-Rubio, D. Rubin, O. Cohen, R. Nicolaescu, and M. Paniccia, "A high-speed silicon optical modulator based on a metal-oxide-semiconductor capacitor," *Nature*, vol. 427, pp. 615-618, 2004.

[238] A. Liu, L. Liao, D. Rubin, H. Nguyen, B. Ciftcioglu, Y. Chetrit, N. Izhaky, and M. Paniccia, "High-speed optical modulation based on carrier depletion in a silicon waveguide," *Opt. Express*, vol. 15, pp. 660-668, 2007.

[239] E. Zgraggen, "Design and Optimization of a Silicon Electro-Optic Modulator." vol. M.S.: ETH Zurich, 2009.

[240] B. Moss, "High-Speed Modulation of Resonant CMOS Photonic Modulators in Deep-Submicron Bulk-CMOS," in *Electrical Engineering and Computer Science*. vol. M.S.: Massachusetts Institute of Technology, 2009.

[241] J. P. Douglas, "Si/SiGe heterostructures: from material and physics to devices and circuits," *Semiconductor Science and Technology*, vol. 19, p. R75, 2004.

[242] I. Balslev, "Influence of Uniaxial Stress on the Indirect Absorption Edge in Silicon and Germanium," *Physical Review*, vol. 143, pp. 636-647, 1966.

[243] M. A. Green, "Third generation photovoltaics: Ultra-high conversion efficiency at low cost," *Progress in Photovoltaics: Research and Applications,* vol. 9, pp. 123-135, 2001.

[244] B. Demczyk, A. King, and R. Gambino, "On the Two to Three Dimensional Growth Transition in Strained Silicon Germanium Thin Films," *Microscopy and Microanalysis,* vol. 13, pp. 732-733, 2007.

[245] C. Gunn, "CMOS Photonics for High-Speed Interconnects," *IEEE Micro,* vol. 26, pp. 58-66, 2006.

[246] M. Griot, "Six-Axis Parallel-Flexure Stages," 2010.