5-2018

Designs and Fabrications On Array Waveguide Gratings for Hybrid Integration

Xiangfeng Chen
Clemson University, xiangfc1991@gmail.com

Designs and Fabrications on Array Waveguide Gratings for Hybrid Integration

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Photonic Science and Technology

by
Xiangfeng Chen
May 2018

Accepted by:
Dr. Lin Zhu, Committee Chair
Dr. Hai Xiao
Dr. Fei Peng

Abstract

Array waveguide gratings (AWG) plays an important role in the field of information and communication technologies through implementation of wavelength division multiplexing(WDM). An AWG consists of an input waveguide, two star couplers, and waveguide phase arrays which provides the wavelength dispersion effect along with the output star coupler. In this thesis, I developed and fabricated AWGs and expect to use the WDM inversely to make them compatible to proposed Wavelength Beam Combining(WBC) system by our group on the silicon nitride platform. In essence, the AWG theory is based on the theory of diffraction and interference. Design rules are summarized under the guidance of AWG theory. Mask layouts and routing algorithms are demonstrated. Simulation results are compared with fabricated devices. Applications and further improvements of AWG are discussed as well. Due to the limitation of fabrication, only features larger than 0.5um can be resolved in our stepper lithography. Best AWG performance of our fabricated devices can reach the insertion loss of 8 dB with 2 dB non-uniformity. The average channel spacing is around 2.8nm. 3 dB channel bandwidth is larger than 1nm with 12.65nm free spectrum range(FSR). Crosstalk is around 19 dB.

Acknowledgments

First, I would like to thank my advisor, Professor Lin Zhu. When I just started this project, I have no prior knowledge of integrated optics let alone array waveguide grattings. But he put his faith in my capability from designs and simulations to fabrications and characterizations. My time in his group helps me to broaden the horizon about optics; we had so many meaningful discussions from nonlinear optics, to passive waveguides and to laser designs.

Second, I would thank Professor Timo Heister. He provided me a good guidance on how to write a project in Python which my now and future research will sure benefit from it. I must thank William Frederick Delaney and Wenzhe Li for their professional cleanroom lessons and useful discussion on CMOS fabrications. I would also like to thank my labmates Yeyu Zhu, Xiaolei Zhao and Siwei Zeng. I am always thankful for their friendly company and passionate discussions about research topics. I learned a lot from them.

Of course, I would never have made it this far if there were not the unconditional support and love from my parents and grandmothers. Here at Clemson, I have made a bunch of friends. Thanks for their company and support. When we look at the sunset of Clemson, we would amaze that it is a place full of miracles!

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The concept of Arrays Waveguide Gratings(AWG) was first reported by Smit[10] as PHASERs at 1988. It employs a phase arrays design. Due to this phase accumulation, signals with different wavelengths from input waveguides will be routed to different output waveguides. So it can be used for wavelength division multiplexer(WDM). AWG has been playing an important role in the field of data communication and information technologies. The integration of phase arrays and star couplers in a small footprint makes the system robust and compact but still with quite good performance in terms of WDM. They can be integrated with semiconductor gain chips and high speed modulators. Different platforms have been investigated by researchers due to their applications in Photonic Integrated Circuits(PIC) and the potential for all-optical signal processing.[11].

When the idea of AWG was popular at 90s, researchers mainly focused on Silica-based planar lightwave circuits(PLC). PLC opens up the world for the on-chip integration.[12] It has many applications and potentials [13]. PLC is using the same materials as fibers[14]. Because along with researches on materials for fibers, the fabrication of Germanium doped silicon dioxide is already mature. Silica-based optical

waveguides are fabricated on silicon substrates by a combination of flame hydrolysis deposition (FHD) and reactive ion etching (RIE). Kawachi and colleagues developed a hydrolysis reaction for the deposition of thin film GeO_2 and fabricated very low loss waveguides (0.01dB/cm) by annealing and dry etching the film.[15]. With more investigation on details of AWG, Fabrications provide feedbacks to the design consideration of an AWG.[16] Improvement of fabrication on PLC and completion of AWG theory make AWG as a compact and robust Multi/Demultiplexers for data communication.[17].

It is necessary to reduce the bending radius by increasing the contrast of refractive index between the core material and cladding materials to achieve higher density PLCs. Researchers used low (0.25%) waveguides to match conventional PLCs to single mode fibers(SMF) and used high contrast (0.75%) waveguides with a bending radius of 5 mm for applications of the compact integration.[18] As shown in Table 1.1, further increasing the value can reduce the bending radius. If it is raised to 1.5%, the bending radius decreases to 2 mm.[17] Increasing the Δ of waveguides in PLCs reduces the waveguide bending radius and makes it possible to realize high-density integrated PLCs.[19, 20, 21]

For the purpose of on-chip compact integration, we need much smaller bending radius to further reduce the footprint. Different platforms with higher refractive index contrast are investigated and fabricated like silicon oxynitride(SiON)[8, 22, 23] on insulator, silicon on insulator(SOI)[24, 25], silicon nitride on insulator[26, 27]and even relative new platforms like Ta_2O_5 with SiO2 cladding.[28] In Table 1.1, I summarize the performance such as the channel spacing, crosstalk, insertion loss and the footprint of different platforms mentioned above.

Among them, the silicon nitride layer can be well prepared through low pressure chemical vapor deposition(LPCVD) and plasma-enhanced chemical vapor deposi-

Platform	Cross-section ($\mu m * \mu m$)	Channel Number	Channel Spacing(GHz)	Insertion Loss(dB)	Cross talk(dB)	Footprint ($mm * mm$)
GeO2/SiO2 0.75%	6*6	16	100	-1	-40	26*21
GeO2/SiO2 1.5%	4.5*4.5	16	100	0.7 0.9	-40	16*16
GeO2/SiO2 1.5%	4.5*4.5	400	25	3.8 6.4	-24	120*60
GeO2/SiO2 1.5%	4.5*4.5	1010	10	-13	-24	-
GeO2/SiO2 1.5%	4.5*4.5	1080	25	-4.5	-24	-
InGaAsP	1.7*0.5	4	400	-5	-24	0.23*0.33
SOI	0.22*0.5	16	200	-2.2	-20	0.5*0.2
SOI	0.22*0.45	8	400	-1.1	-25	0.2*0.35
Si3N4	0.22*0.9	12	$2nm$@892	-1.2	-16.9	$0.34mm^2$
SiON	2*0.52	41	$5.5nm$@881	-1.1	-20	$340mm^2$

Table 1.1: Previous research on AWGs[1, 2, 3, 4, 5, 6, 7, 8]

tion (PECVD)[29]. LPCVD Si_3N_4 are reported achieving lower propagation loss with 8-9 dB/m[26] and 5.5 dB/m[30]. High Q ring resonator has also been demonstrated on LPCVD Si_3N_4 with ultra-high quality factors (Q) of 19 million, 28 million, and 7 million at 1060 nm, 1310 nm, and 1550 nm, respectively.[31] Following the previous work, record high loaded Q factor of 35 million at 1550nm and an extracted intrinsic Q factor of 55 million at 1620nm measuring at the add drop of ring resonator.[32] So LPCVD Si_3N_4 with SiO_2 cladding or just the insulator layer is a low loss platform and the with refractive index contrast in between of silicon and silicon dioxide. If we use this platfrom for our AWG, in return, we will have AWG devices which are a real balance among the performance, on-chip loss, and the compactness. The larger refractive index contrast is, more compact the AWG will be. However, the delay lines implemented with narrow waveguides width are prone to phase errors due to

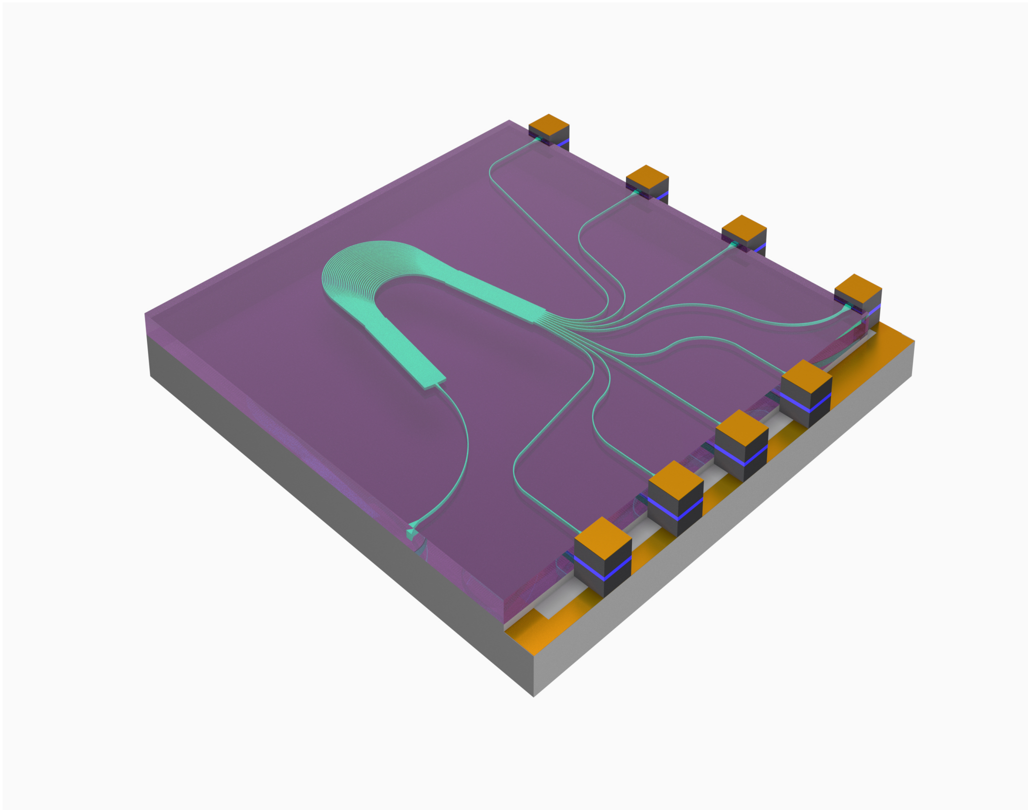


Figure 1.1: The schematic drawing of Wavelength Beam Combing

small fluctuations of the waveguide dimension.[16] Also, as we choose a platform like SOI, the high refractive index contrast makes it difficult to implement good coupling between free propagation region and arrays. There is a significant vertical mode mismatch between the waveguide modes of arrays and the slab modes in the star couplers. This can result in more unwanted reflections compared with one with lower Δ, which increases the overall insertion loss of the device. SOI platfrom requires DUV stepper and well control on the sidewall roughness and dimension variations. Moreover, considering the process limitation in Clemson, silicon nitride[33] is a suitable platform for us to design and fabricate AWG for hybrid integration application.

Considering the case that only one input waveguide and several output channels in AWG, the system serves as a WDM. Hybrid integration with photo detectors

will introduce a compact spectrometer.[34] While use it inversely, it serves as a beam combiner. Our group has proposed a hybrid integration system which we directly integrate laser gain chips (with AR coating on coupling facet) with AWG input channels as in Figure 7.2. At the end of the output waveguide, we will employ a reflector to provide the feedback to each channel. It can realize wavelength controlling and beam combining together in a compact manner as opposed to external cavity which we will discuss about in last section of this thesis for the application of my designed AWGs.

This thesis, in essence, is about applying the theory of AWG to design works and use CMOS compatible process[35] to fabricate AWGs for our application of Wavelength Beam Combining System.[9] Therefore, I layout the structure of following chapters in my thesis as:

In Chapter 2, I will briefly review the theory of AWG. In Chapter 3, AWG designs and simulations will be investigated and I will try to explain the impact of different parameters on performance of AWG according to the theory derived in Chapter 2. I will also use the BPM simulation tool PHASER to compare the device performance and optimize the AWG designs. In Chapter 4, mask layout and waveguide routing algorithm are investigated and codes are provided in the Appendix. In Chapter 5, I will summarize my process flow for fabrication of AWG and guidelines for improvement on recipe and characterization. In Chapter 6, the measurements setups and results are demonstrated. In the Last Chapter, application of AWGs and future work are discussed.

Chapter 2

Theory of Array Waveguide Gratings

AWG theory in essence is based on the theory of N-slits interference and diffraction. The phase from each phase array will be integer times 2π by sequentially increasing the length of each array by an integer times the central wavelength. It will tilt the phase plane of output waveguides at wavelength other than the central wavelength. It is demonstrated as ΔL in Equation 2.10. With curved junction, these arrays will focus at receiving end which we label it image plane. It follows the same principle as N-slits interference and diffraction. The only difference is with the tilted phase plane controlled by ΔL which will steer the light signals with the respect of wavelength. Due to this steering, we receive light signals at different locations of image plane for different wavelengths. Then the light will be tapered in and guided into different waveguides which we name output channels. The phase arrays along with Output Star Coupler is the crucial part for the AWG performance. Therefore, in this chapter, I will briefly review the theory and key equations about interference and diffraction as in 2.3 and 2.5. I will associate them with phase arrays to derive

wavelength dispersion of an AWG. These derivation are to give us a theory guidance for our AWG designs on wavelength dependent dispersion around the central wavelength. Normally, if no special functions needed to adjust the image plane, the Input Star Coupler is the same as Output Star Coupler. The Input Star Coupler provides a free propagation region for input light from fan-out taper. Only single slit diffraction theory is investigated. As for design, only far-field simulations of given width of input waveguide are in our interest as given in Figure 2.6.

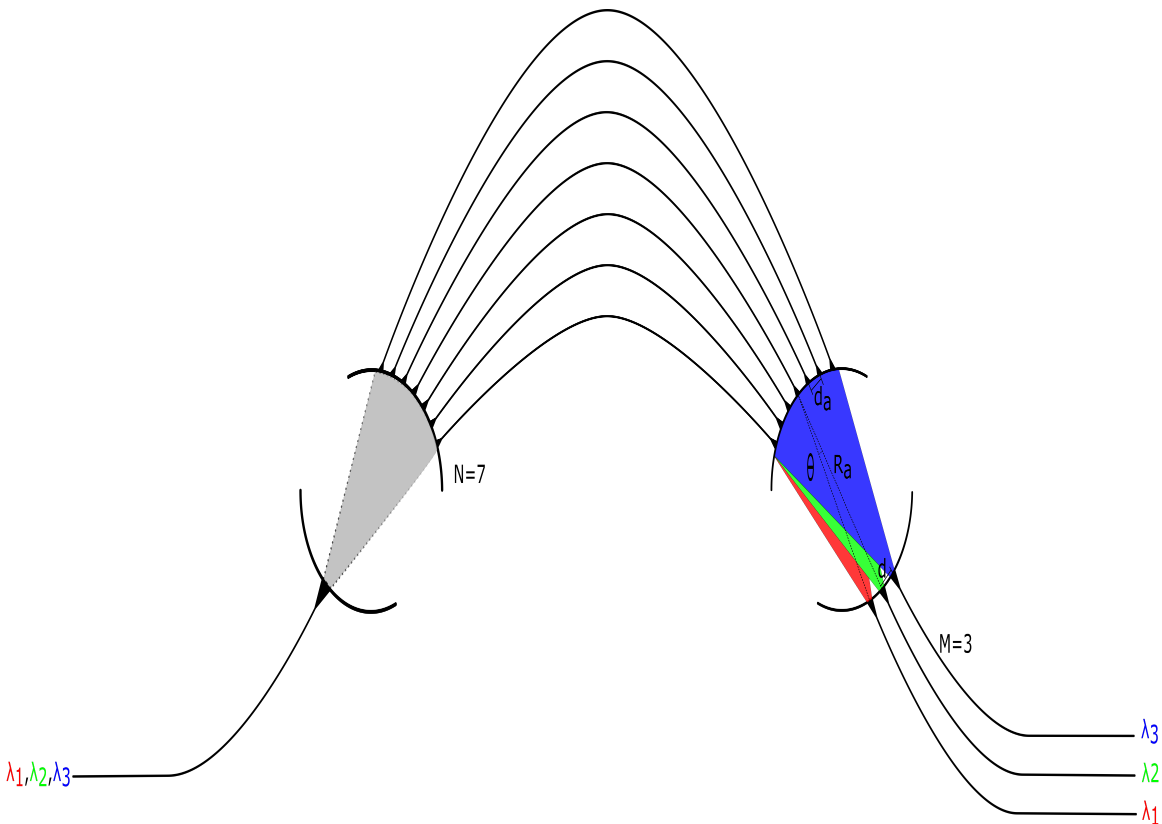


Figure 2.1: Schematic drawing of the AWG

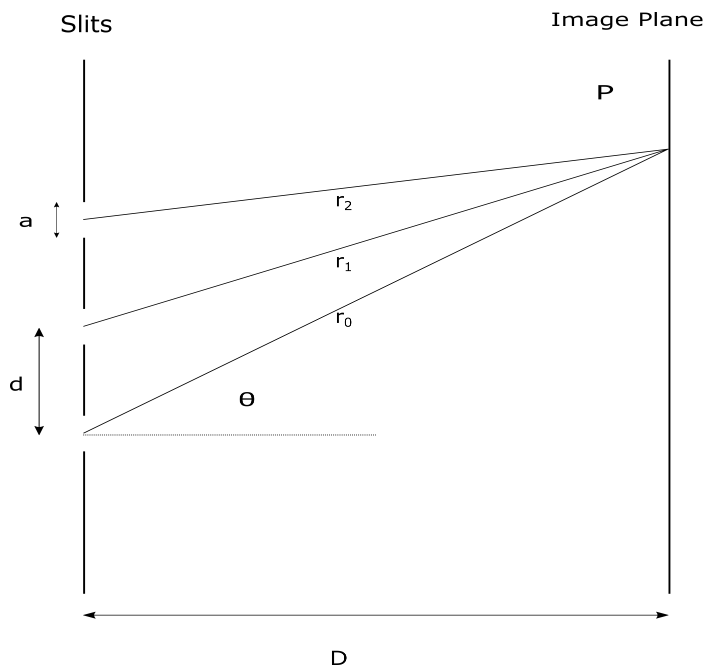


Figure 2.2: N Slits Schematic

2.1 Diffraction and Interference

The light path difference at point P can be derived from the equations below. If the distance to image plane is much greater than the distance(d), the sum of r_1 and r_2 is approximated to be $2r$:

$$r_2^2 - r_1^2 = (r_2 + r_1)(r_2 - r_1) = 2dr\sin\theta \tag{2.1}$$

$$r_2 - r_1 = d\sin\theta \tag{2.2}$$

For constructive interference to occur, we have:

$$\delta = d\sin\theta_m = m\lambda, m = 0, \pm 1, \pm 2, \pm 3, \pm 4, \ldots \tag{2.3}$$

The intensity of interference pattern is:

$$I = I_0 \cos^2(\pi d \sin(\theta/\lambda)) \tag{2.4}$$

For destructive interference of one slit with width a , we have:

$$a \sin\theta_m = m\lambda, m = 0, \pm1, \pm2, \pm3, \pm4, \ldots \tag{2.5}$$

The intensity of diffraction pattern is:

$$I = I_0[\frac{\sin(\beta/2)}{\beta/2}]^2 = I_0[\frac{\sin(\pi a \sin(\theta/\lambda))}{\pi a \sin(\theta/\lambda)}]^2 \tag{2.6}$$

If we consider the diffraction of each slit, the intensity pattern will have an modulation of diffraction envelope:

$$I = I_0 \cos(\pi d \sin(\theta/\lambda)^2)[\frac{\sin(\pi a \sin(\theta/\lambda))}{\pi a \sin(\theta/\lambda)}]^2 \tag{2.7}$$

If we have N slits, the modulation envelope caused by diffraction of single slit is the same. While the total field sum will be:

$$E = E_1 + E_2 + E_3 + \cdots + E_N = \sin(wt) + \sin(wt + \Delta\beta) + \cdots + \sin(wt + (N-1)\Delta\beta) \tag{2.8}$$

We can not tell the difference between a wide slit and an infinitesimal slit when an appropriate amount of light coming through. And the N slits intensity pattern can

be treated as N-slit interference modulated by one-wide slit diffraction.

$$I = I_0\{\frac{sin(1/2ka\sin\theta)}{1/2ka\sin\theta}\frac{sin(1/2Nkd\sin\theta)}{1/2Nkd\sin\theta}\}^2 \tag{2.9}$$

The simulation results are given in the Figure 2.3 as a show case of the diffraction and interference simulation of 10 slits. First, the intensity envelop is modulated by diffraction. Most importantly, the roll-off is controlled by the diffraction. For the same wavelength, the smaller the width of a waveguide is, the slower the roll-off of intensity will be. For the same waveguide width, the larger wavelength is, the slower the roll-off of intensity will be.

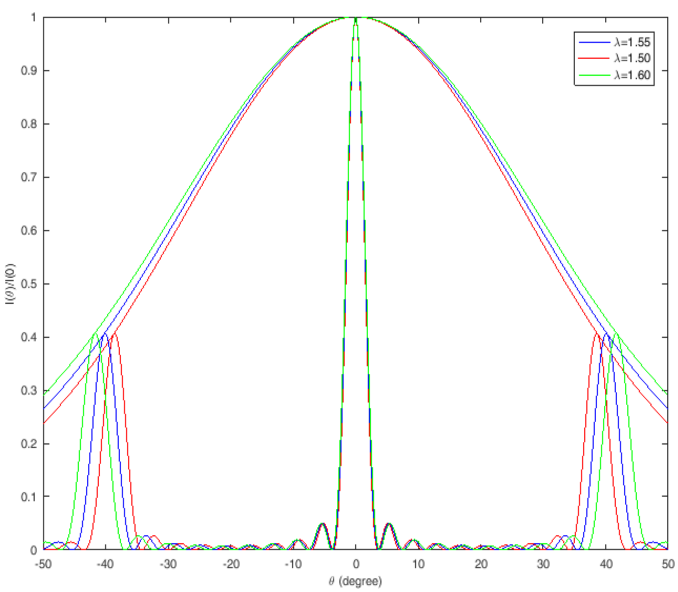


Figure 2.3: 10 slits simulation of the intensity with variation of Θ among different central wavelengths with d=2.4um, a =1.2 um

2.2 Phase Arrays

Waveguide arrays are engineered in such way that the sequential increase of length difference is an integer times the central wavelength. This will give us path difference of each array as ΔL. For the central wavelength, it will reach the condition of constructive interference when the phase plane is just perpendicular to the center of phase arrays. After focusing due to curved plane, it will land at the center of the image plane. As for other wavelengths, given the path difference, they will have a tilted phase plane and have a beam steering effect which in theory as wavelength dependent dispersion. No matter what case we are investigating, it all follows the same theory of constructive interference in Equation 2.3. For light propagating in waveguide, the adjustment is needed compared with free space propagation. We have $\lambda_c/n_{wg}(\lambda_c)$ instead as in 2.10.

$$\Delta L = \frac{m\lambda_c}{n_{wg}(\lambda_c)} \tag{2.10}$$

2.3 Output Star Coupler

As showed in Figure 2.4, d_a is the separation distance between the center of two adjacent waveguides and R_a is the length of free propagation region. The drawing of Rowland Circle is not straight forward in the schematic as it should be at receiving waveguides. The radius for the curvature at receiving waveguides is half of the R_a. θ is the dispersion angle for wavelengths that are not central wavelength. We use the interference theory on the Output Star Coupler. Light comes through phase arrays and then from the output junction propagates to receiving end. It has an accumulated phase. When the phase equals integer times 2π, it will interference constructively as

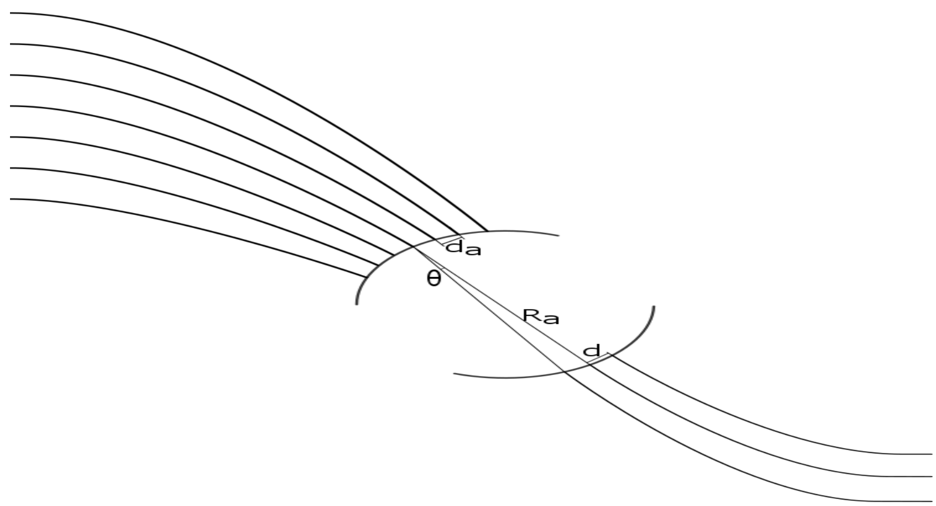


Figure 2.4: Schematic drawing for phase Arrays and receiving Waveguides

in Equation 2.3. Combing the path difference of as ΔL and interference condition, we have Equation 2.3:

$$\frac{2\pi n_{slab}(\lambda)}{\lambda} d_a \sin\theta = \frac{2\pi n_{wg}(\lambda)}{\lambda}\Delta L - m2\pi \tag{2.11}$$

After algebra operation, we derive:

$$\sin\theta = \frac{n_{wg}(\lambda)\Delta L - m\lambda}{n_{slab} d_a} \tag{2.12}$$

Substituting equation 2.10 to 2.12, we can have equation 2.13:

$$\sin\theta = m\frac{n_{wg}(\lambda)\lambda_c - n_{wg}(\lambda_c)\lambda}{n_{slab} n_{wg}(\lambda_c) d_a} \tag{2.13}$$

In order to investigate the wavelength dependence of the diffraction angle, $\sin\theta$ is further differentiated. Also, the n_{wg} in equation 2.13 is not independent regarding variation in wavelength.

$$n_{g,wg} = n_{wg}(\lambda_c) - \lambda \frac{dn_{wg}}{d\lambda} \tag{2.14}$$

Due to the refractive index itself is wavelength dependent as in equations 2.14, we use a group refractive index substituted in the following equations:

$$\frac{d\theta}{d\lambda} = -\frac{mn_{g,wg}}{n_{slab}n_{wg}(\lambda_c)d_a} \tag{2.15}$$

$$\frac{d\theta}{df} = -\frac{mn_{g,wg}\lambda_c^2}{n_{slab}n_{wg}(\lambda_c)d_a c} \tag{2.16}$$

We can further simplify the dispersion equations above as in 2.18. The dispersion response with respect of the wavelength and frequency can be converted according to equation 2.20.

$$\frac{d\theta}{d\lambda} = -\frac{n_{g,wg}\Delta L}{\lambda_c n_{slab} d_a} \tag{2.17}$$

$$\frac{d\theta}{df} = -\frac{n_{g,wg}\Delta L}{f_c n_{slab} d_a} \tag{2.18}$$

$$\Delta\theta_{FSR} = \frac{\lambda_c}{n_{slab} d_a} \tag{2.19}$$

$$\Delta\theta_{FSR} = \frac{d\theta}{df} f_{FSR} \tag{2.20}$$

Equation 2.18 and 2.21 is the key equation that guides our AWG design. Because we design our device to specific FSR and dispersion performance.

$$f_{FSR} = \frac{c}{n_{g,wg}\Delta L} \tag{2.21}$$

2.4 Input Star Coupler

As in the Figure 2.5, the propagation constant can be divided into two perpendicular vector.

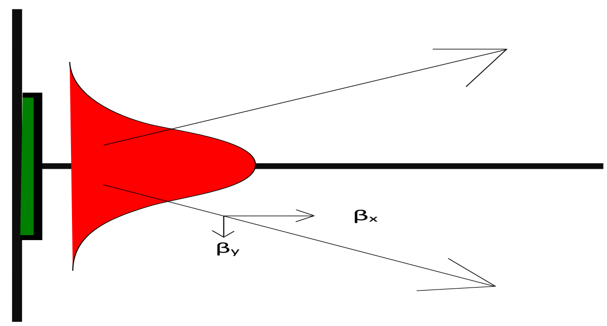


Figure 2.5: Schematic drawing of the free propagation at FPR inside Input Star Coupler

$$E(\beta_x, \beta_y) = \int f(x, y)exp(-j(\beta_x x + \beta_y y))dxdy \tag{2.22}$$

$$\beta_y = \beta \sin\theta = \frac{2\pi n_{eff}(\lambda)}{\lambda} \sin\theta \tag{2.23}$$

$$E(\beta_y) = \int f(y)exp(-j\beta_y y)dy \tag{2.24}$$

The y direction profile will diverge to the junction between the star coupler and phase arrays. And the profile can be obtained through beam propagation method as Figure 2.6. This figure provides three far-field envelops of silicon nitride waveguides with width of 1.2um, 3.0um and 5.8um, respectfully. We can conclude that for the same

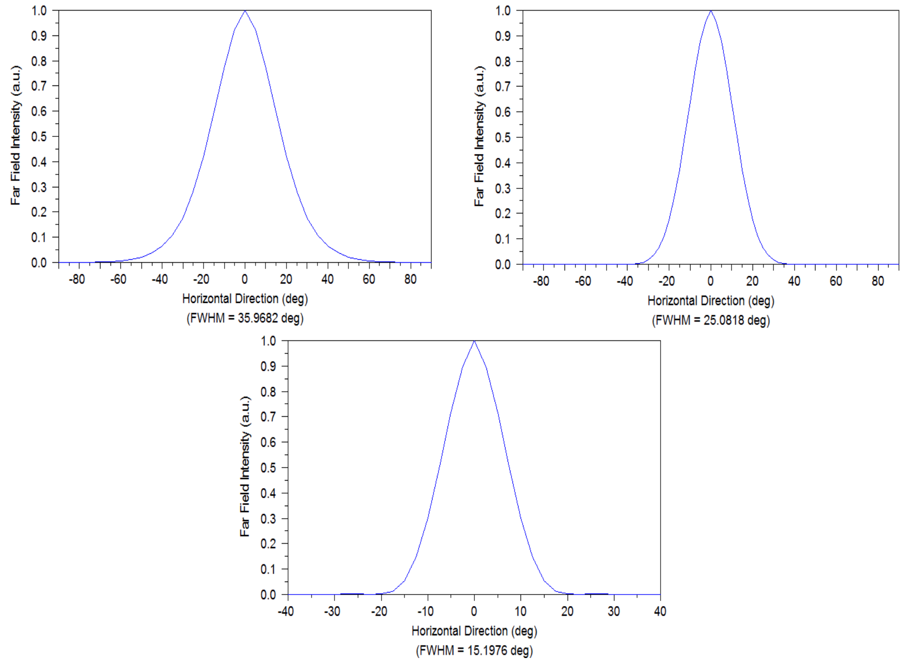


Figure 2.6: Far-field envelope of waveguides with 1.2um, 3.0um and 5.8um width and 300nm height

material platform, waveguides with larger width will have smaller Full Width Half Maximum(FWHM). Since the length of free propagation length R_a and waveguide separation d_a are chosen while designing the performance of Crosstalk and Output Star Coupler, the number of phase arrays needed is determined by the far-field profile of the input waveguide. Also, to get a relative uniform phase distribution around junctions, the radius of the input star coupler should be sufficiently large to have circular phase fronts, which should be larger than Rayleigh Radius from the given aperture width approximated in Equation 2.25.

$$R_{Rayleigh} \approx \frac{\pi w^2 n_{eff}}{2\lambda} \tag{2.25}$$

with w the width of the input aperture.

Chapter 3

Design and Simulation

3.1 Insertion Loss

3.1.1 Loss Mechanism

Loss is of the very importance in design of Array Waveguide Gratings. So I itemize these mechanism as below:

- Mode mismatch during transition
- Loss due to gaps between tails of adjacent fan-in tapers at transition junction
- Propagation loss from waveguides and free propagation regions
- Bending loss

The loss can be introduced from mode mismatch at junctions, for example, from modes of the input waveguide to modes of the free propagation region and from modes at the free propagation region to modes of receiving waveguides. Ideally, we expect to guide as much light as possible at the junctions between free propagation regions(FPR) and array waveguides. However, we need take crosstalk from

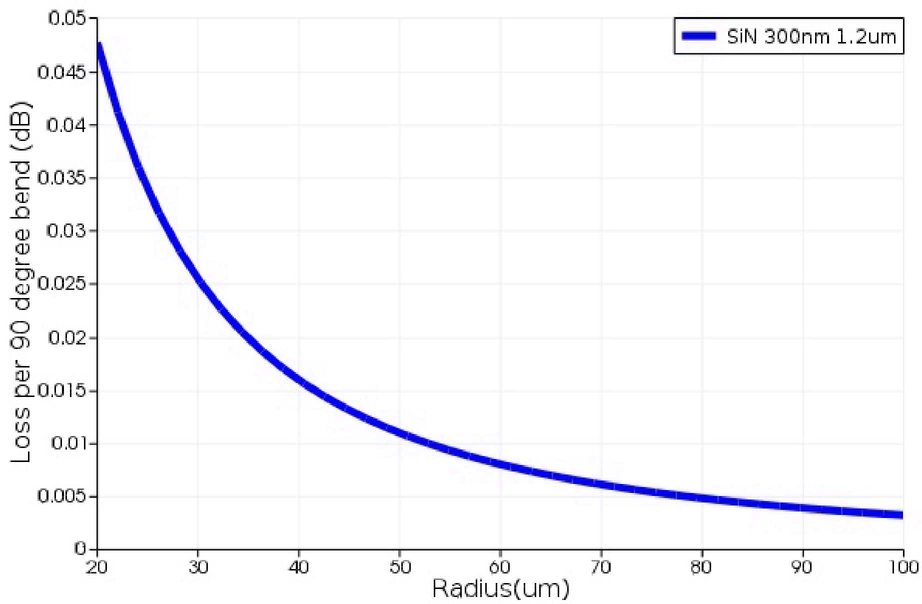


Figure 3.1: Bending loss of the silicon nitride waveguide with 1.2um width and 300nm height

two adjacent waveguides into consideration. The minimum gap we have decided to choose for acceptable crosstalk level inherently introduces loss at the transition region. Waveguides' propagation loss is normally introduced by scattering due to surface and sidewall roughness, and from materials absorption due to defects and molecular bond vibration. For silicon nitride, other than defects and Si-N inherent abosprbtion, we may introduce extra absorption of hydrogen bond at 1550 nm wavelength range by wet fabrication. Moreover, bending loss is another loss mechanism that light will radiate out as waveguide tends to curve. The mode simulation will give us a good picture that modes tend to confine closed to outer radius of waveguide. From mode transition perspective, there is mode mismatch between two modes from continuous meshes at the bending region. In Figure 3.1, as bending radius increase, the loss is gradually approaching the value of straight waveguide. It may not occupy as many percentage of the total loss as junction loss and propagation loss.

3.1.2 Strategies

By understanding the loss mechanism, we can develop a strategy to improve the performance of an AWG from both perspectives of the design and the fabrication. I itemize these strategy as below:

- Keep gaps between arrays as small as possible
- Taper the input waveguides and arrays width adiabatically
- Improve etching recipes to reduce sidewall roughness
- Anneal Silicon Nitride devices after wet processing
- Increase the number of arrays
- decrease array waveguide width d_a
- Double etched method(wide shallowly etched aperture)

To realize these strategies, we aim at designing an AWG with long adiabatic tapers to single mode at every transition junction. This will decrease the loss coming from mode mismatch. And also the tapper width is ideally to equal the distance of waveguide seperation in arrays to collect as much light as possible at junctions between FPRs and arrays. Less polymerization will provide us smoother sidewall roughness and planner surface by improving the etch chemistry. 1050 Celsius annealing is promising to get rid of hydrogen bond in the device which will improve the propagation loss of silicon nitride core materials.[12] More light will be coupled into arrays if we increase the number of arrays. It will also decrease the linewidth of main lobes of images for receiver waveguides as well which will ideally reduce the crosstalk between adjacent receiving channels. Decreasing separation of array waveguides, $(\Delta\theta_{FSR})$ will

increase, which means the power coupled into adjacent order will decrease. Double etched method will reduce the loss introduced by mode mismatch at transition regions because the shallow trench provides a week guidance for light.

3.2 Non-uniformity

According on the far field of the individual array, part of the power will couple into adjacent orders. This fraction of light induces loss in the center channel. Other wavelengths will experience higher loss compared to the central wavelength as the light follows the far-field envelop of the individual aperture of the waveguide array. This difference in loss is characterized as non-uniformity.

3.2.1 Improvements

- Non-cyclic FSR
- Decrease the array waveguide width

With the trade-off of introducing more propagation loss and device area, we can choose a reasonable length of free propagation region according to Equation 2.18. As showed in equation 2.21 if equal intensity at both side of central axis, maximum 3dB loss will be introduced. So it is better to not have cylic FSR and normally we design FSR with two dummy channel as in equation 3.1.

$$FSR = (M + 2) * Channel\ Spacing \tag{3.1}$$

The diffraction angle of a wavelength will be the same if we keep $\frac{m}{d_a}$ constant, according to the equation 2.13. An wider far-field slows the roll-off which improves the non-

uniformity of insertion loss of the other wavelengths.

3.3 Phase Error

Phase error will deteriorate the image quality at the receiving junction which will significantly influence the performance of an AWG.[16] So in list below, I itemize some strategies:

- Reduce the sidewall roughness and dimension variation
- Decrease the number of arrays
- Make the same bending radius for the routing of arrays

Increasing the width and decreasing the height will reduce the percentage of dimension variation in longitudinal direction. The vertical dimension is well controlled by deposition process such as LPCVD. Additionally, dn/dw can be an indication that the dimension variation causes refractive index change which eventually introduces phase error regarding to designed phase arrays. For phase error, the wider waveguides are, the less phase error in the AWG. However, we need AWG to operate in single mode region so the largest width we can choose is around 1.7um for an silicon nitride AWG with central wavelength near 1550 nm, as showed in figure 3.3. It is reasonable to think that the more arrays there will be, the more chances to have dimension variations. We expect to decrease the number of arrays considering the trade-off of the insertion loss due to transition between FPRs and arrays. We can not ignore the parameter, bending radius. Waveguides with different bending radius have variation of group refractive index. This is another source for the phase error. Ideally, it is better to have the same bending radius for all arrays. For example, Photon Design and Luceda Photonic will have this design option for array arms. However, our group

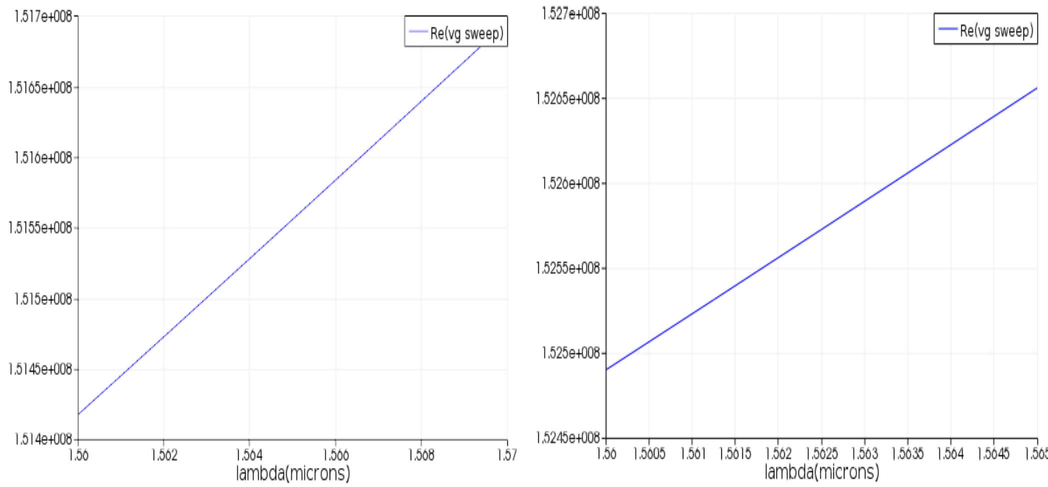


Figure 3.2: A comparison of group index between waveguides with two bending radius

is using PHASER to do simulations and layouts of AWGs. As for the simulation using beam propagation method(BPM), PHASER software did not take phase error into simulation formula, so we can only have feedback from fabrications.

3.4 Single Mode Operation

Every mode has corresponding refractive index and group refractive index. The key theory the AWG is based on is diffraction and interference of phase arrays and it is case sensitive in terms of refractive index and group refractive index of waveguides like equation 2.10. All our design is based on single mode operation purpose.

3.5 Simulations

According to the theory in the previous chapter, we have a guideline for design processes in Figure 3.4 and Appendix B. Based on these strategies above, below is the simulation results confirming these design rules.

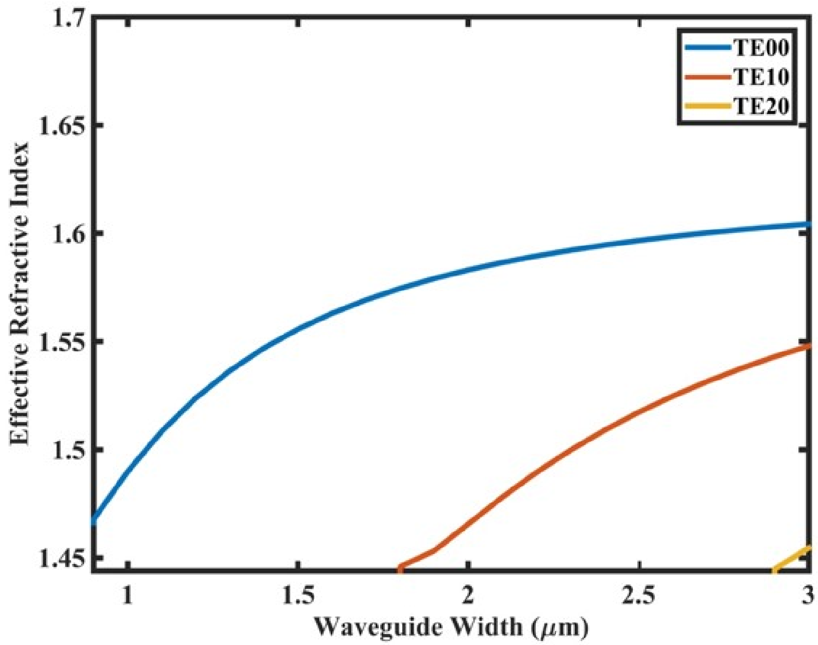


Figure 3.3: Effective refractive index with respect of width variation for 300nm height core silicon nitride on 4um silicon dioxide

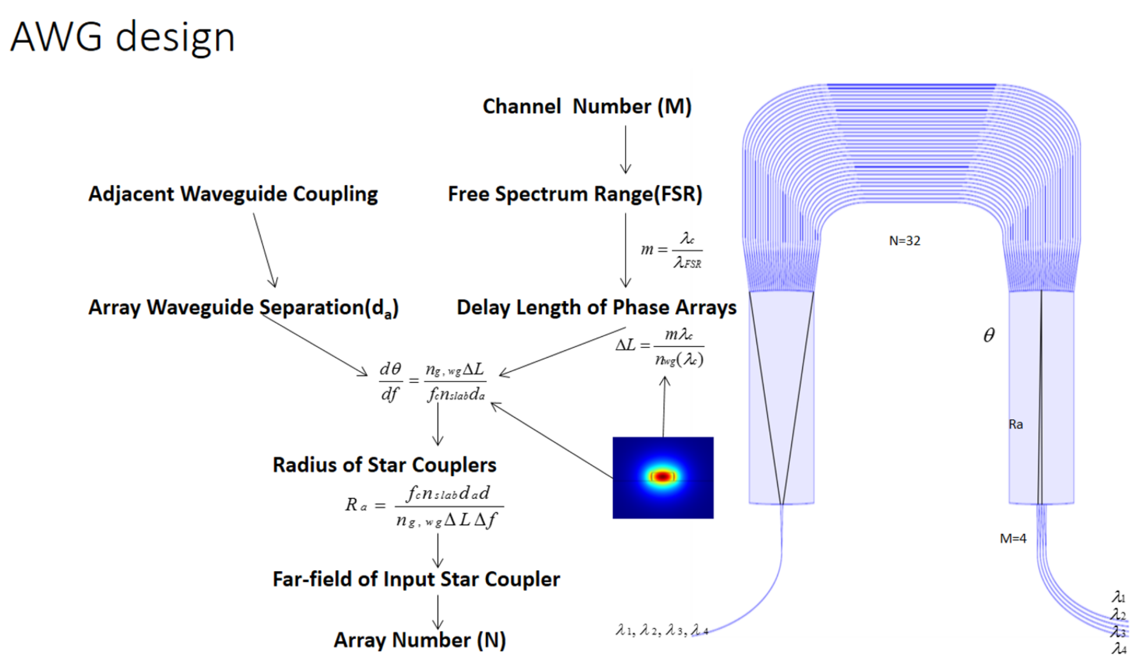


Figure 3.4: Design Flow for an AWG

- A1-A2-A3-A7: d_a increases from 1.4um, 1.6um, 1.8um to 2.0um
- A1-A5: A5 has 50um bending radius while A1 has 150um radius
- A1-A6: A6 has 64nm FSR while A1 has 32nm FSR
- A1-A8: A8 has 74 array arms while A1 has 56 array arms
- A3-A4: A4 has parameters: da=1.8um, d=5um, and w=4um while A3 has da=1.8um, d=5um, and w=4.8um
- A3-A9: A9 has 0.1nm as gap distance between taper tails at transition region while A3 has 0.2nm gap

As in the Figure 3.5 and Table 3.1, 3.2, 3.3, 3.7, d_a in designs are increasing from 1.4um, 1.6um, 1.8um to 2.0um. And the insertion loss is increasing as d_a increasing. And comparison between A3 and A9 is consistent with our assumption that insertion loss will improve if AWGs have less gap between array waveguides at the cost of the cross talk level in channels. In Figure 3.6 and Table3.5,3.5, it shows no improvement on the phase error because the PHASER does not consider group refractive index difference in the simulation. One more thing the PHASER does not consider is the propagation loss, however, larger radius may have total waveguide length increased. But the comparison is still useful for us after fabrication when we expect the improvement of phase noise if we increase the bending radius of array arms. Non-cyclic FSR is adopted for these designs to improve non-uniformity. We can see each channel becoming more even in terms of insertion loss while we doubling the FSR from 32nm to 64nm. The non-uniformity is decreasing from 1.4dB to 0.5dB as in Figure 3.7 and Table 3.1,3.6. If we use different width for input and output waveguides, d=5 and d=4 respectfully, in Figure 3.9, there will be some light loss in the receiving end.

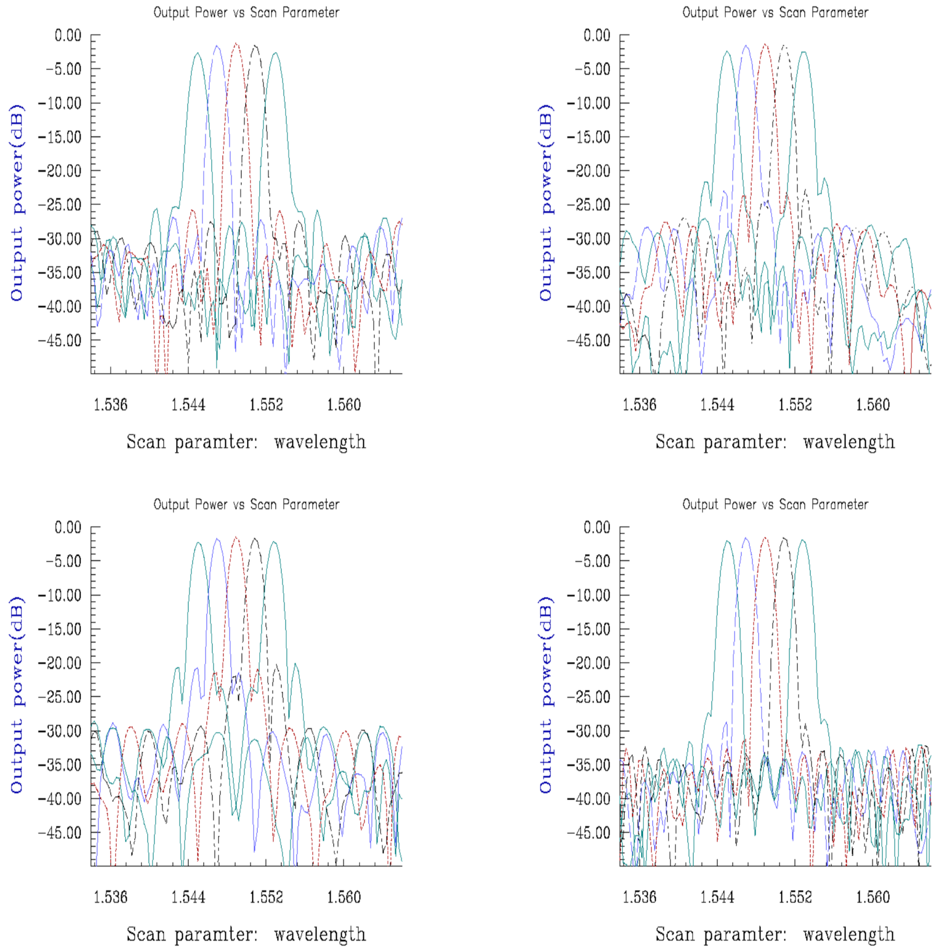


Figure 3.5: A2(upper right) has da=1.6um while A1(upper left) has da=1.4um; A7(lower right) has da=2.0um A3(lower left) has da=1.8um

What in return is the improvements in the final channel crosstalk and the elimination of the second lobes as we find out in the A3.

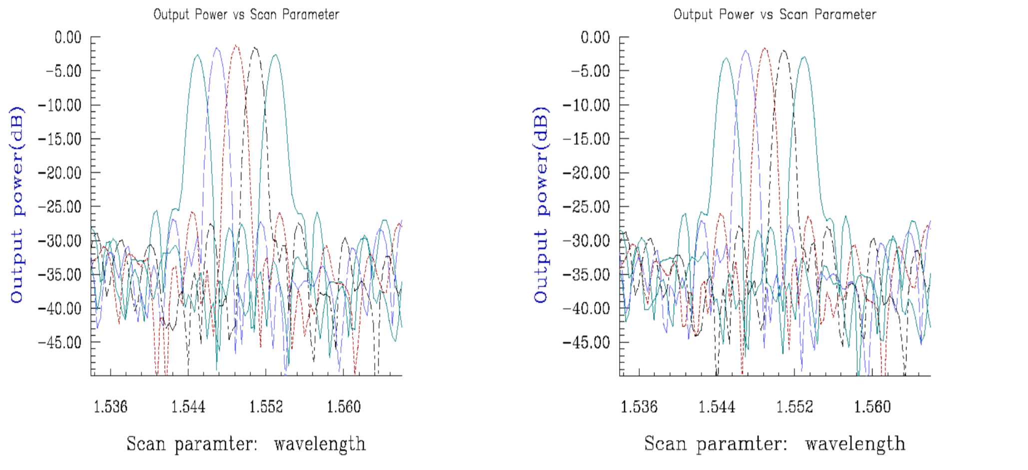


Figure 3.6: A5(right) has 50um bending radius while A1(left) has 150um radius

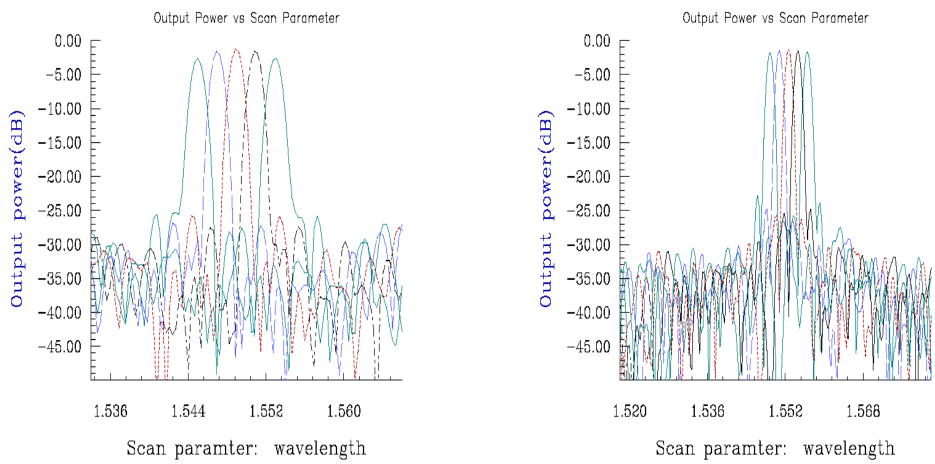


Figure 3.7: A6(right) has 64nm FSR while A1(left) has 32nm FSR

Channel No.	Peak	Width	CrossTalk	
1	-2.625888	0.003047	-30.028084	Spacing = 0.002263
2	-1.603955	0.002768	-35.185944	Spacing = 0.001939
3	-1.251185	0.002815	-42.618279	Spacing = 0.001939
4	-1.586480	0.002771	-37.444535	Spacing = 0.001939
5	-2.674024	0.002961	-29.374320	

Table 3.1: A1 channel performance

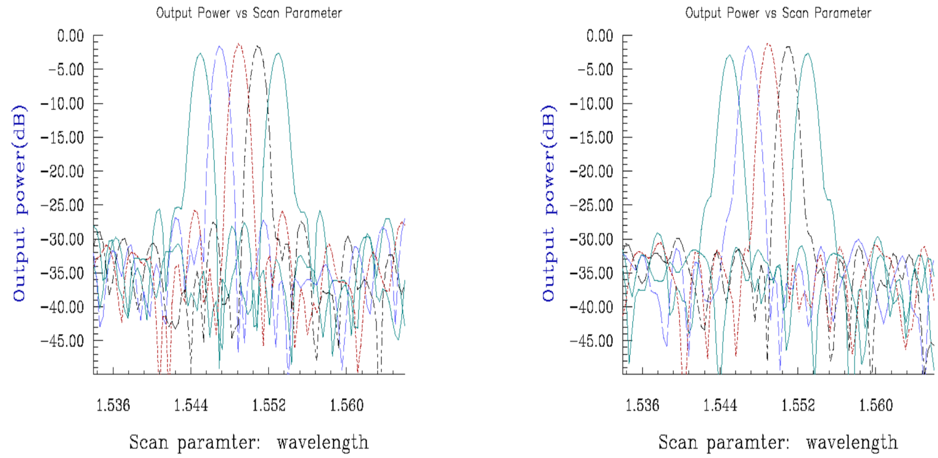


Figure 3.8: A8(right) has 74 array arms while A1(left) has 56 array arms

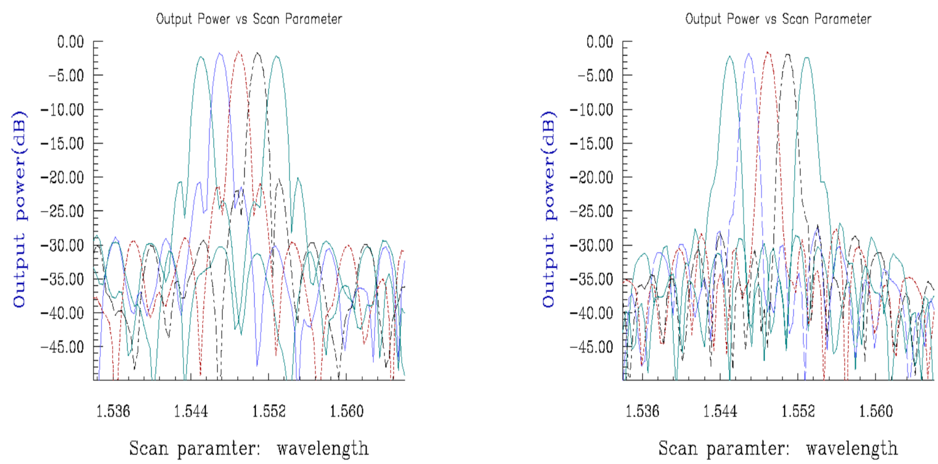


Figure 3.9: A4(right) has parameters: da=1.8um, d=5um, and w=4um while A3(left) has da=1.8um, d=5um, and w=4.8um

Channel No.	Peak	Width	CrossTalk	
1	-2.483989	0.002899	-25.289906	Spacing = 0.001939
2	-1.577062	0.002767	-26.169565	Spacing = 0.001939
3	-1.361011	0.002737	-24.708230	Spacing = 0.001939
4	-1.553678	0.002844	-23.756502	Spacing = 0.001939
5	-2.447373	0.002943	-23.466183	

Table 3.2: A2 channel performance

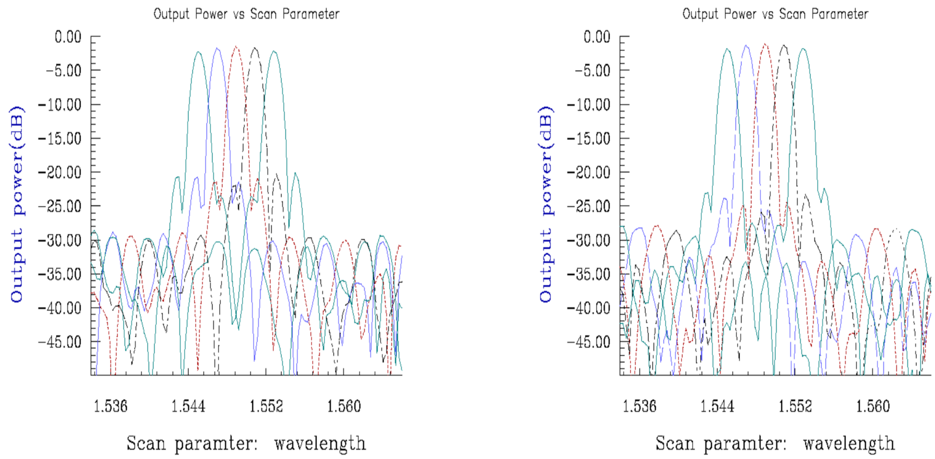


Figure 3.10: A9(right) has 0.1nm as gap distance between taper tails at transition region while A3(left) has 0.2nm gap

Channel No.	Peak	Width	CrossTalk	
1	-2.164245	0.002840	-21.915625	Spacing = 0.001939
2	-1.664921	0.002663	-22.671734	Spacing = 0.001939
3	-1.502223	0.002728	-21.994736	Spacing = 0.001939
4	-1.732137	0.002638	-21.466975	Spacing = 0.001939
5	-2.249250	0.002845	-20.759068	

Table 3.3: A3 channel performance

Channel No.	Peak	Width	CrossTalk	
1	-2.358376	0.002661	-27.194565	Spacing = 0.002263
2	-1.898130	0.002538	-33.085124	Spacing = 0.001939
3	-1.552537	0.002558	-39.894810	Spacing = 0.001939
4	-1.743108	0.002534	-35.467970	Spacing = 0.001939
5	-2.180813	0.002805	-26.492345	

Table 3.4: A4 channel performance

Channel No.	Peak	Width	CrossTalk	
1	-2.971627	0.003006	-31.466310	Spacing = 0.002263
2	-1.969164	0.002748	-35.527233	Spacing = 0.001939
3	-1.634068	0.002802	-41.688620	Spacing = 0.001939
4	-1.984340	0.002756	-37.518363	Spacing = 0.001939
5	-3.085747	0.002936	-30.481665	

Table 3.5: A5 channel performance

Channel No.	Peak	Width	CrossTalk	
1	-1.724679	0.003012	-29.326566	Spacing = 0.001930
2	-1.519135	0.002989	-32.048941	Spacing = 0.001930
3	-1.338786	0.002907	-40.551599	Spacing = 0.001930
4	-1.472986	0.002926	-39.767019	Spacing = 0.001930
5	-1.798761	0.002936	-35.810758	

Table 3.6: A6 channel performance

Channel No.	Peak	Width	CrossTalk	
1	-1.894538	0.002789	-32.489672	Spacing = 0.001939
2	-1.626086	0.002704	-36.665990	Spacing = 0.001939
3	-1.526747	0.002757	-33.200407	Spacing = 0.001939
4	-1.606943	0.002710	-31.302378	Spacing = 0.001939
5	-2.076382	0.002782	-28.682912	

Table 3.7: A7 channel performance

Channel No.	Peak	Width	CrossTalk	
1	-2.683770	0.003104	-29.221608	Spacing = 0.001939
2	-1.650919	0.002990	-28.088313	Spacing = 0.002263
3	-1.216866	0.002950	-31.161336	Spacing = 0.001939
4	-1.574172	0.003017	-28.054119	Spacing = 0.001939
5	-2.878575	0.003152	-26.019989	

Table 3.8: A8 channel performance

Channel No.	Peak	Width	CrossTalk	
1	-1.776631	0.002891	-25.621997	Spacing = 0.001939
2	-1.273352	0.002674	-27.952002	Spacing = 0.001939
3	-1.102879	0.002727	-26.572592	Spacing = 0.001939
4	-1.340008	0.002670	-25.516570	Spacing = 0.001939
5	-1.857084	0.002913	-23.974718	

Table 3.9: A9 channel performance

Chapter 4

Layout and Mask Design

4.1 GDS File

GDS II is a trademark of Calma Company which is a subsidiary of General Electric Company, U.S.A. In the field of integrated circuits design, the most popular format is the Calma GDS II stream format. Although Calma has updated the format as their Computer Aid Design systems developing, they have maintained backward compatibility for GDS II files. A GDS II file contains geometry or other cell references in terms of binary floating points. These cells are called structures which have up to 32 characters. A library of these structures is contained in a file with a header, a collection of structures, and a tail. Each structure consists of a header, a collection of elements, and a tail. There are seven kinds of elements: boundary defines a filled polygon, path defines a wire, structure reference reference a sub cell, array reference copy an array of the reference, text is for documentation and box places rectangular geometry.

File Header	Bytes 3 and 4	Parameter Type
HEADER	0002	2-byte integer
BGNLIB	0102	12 2 byte integer
LIBNAME	0206	ASCII string
REFLIBS	1F06	2 45-charter ASCII
FONTS	2006	4 44-charter ASCII
ATTRTABLE	2306	44-charter ASCII
GENERATIONS	2202	2-byte integer
FORMAT	3602	2-byte integer
MASK	3706	ASCII string
ENDMASK	3800	No data
UNITS	0305	2 8-byte floats
File Tail	Bytes 3 and 4	Parameter Type
ENDLIB	0400	No data
Structure Head	Bytes 3 and 4	Parameter Type
BGNSTR	0502	12 2-byte integers
STRNAME	0606	up to 32 characters
Structure Tail	Bytes 3 and 4	Parameter Type
ENDSTR	0700	No data

Table 4.1: Records for GDSII Header File

4.2 IPKISS and Picazzo

The IPKISS software platform is conceived as a modular system and is developed by Photonic Research Group in Ghent University. It aims at application in the design work of photonic components and circuits as spirals, spliters, ring resonators and so on. It can be customized for use in many field related to microelectronics and microfluidics.[36]

IPKISS engine is based on object-oriented language, Python. It is such a powerful platform that you can do photonics design for every module, Picazzo, like waveguides, MZI, MMI and combine them with routing schemes and finally use Pysimul to do 2D mode and 3D FDTD simulations. After feedback and verification from simulation, it also has function leading GDS file generation and manipulation such as rotation, translation, magnification, scaling, horizontal, vertical and center-point

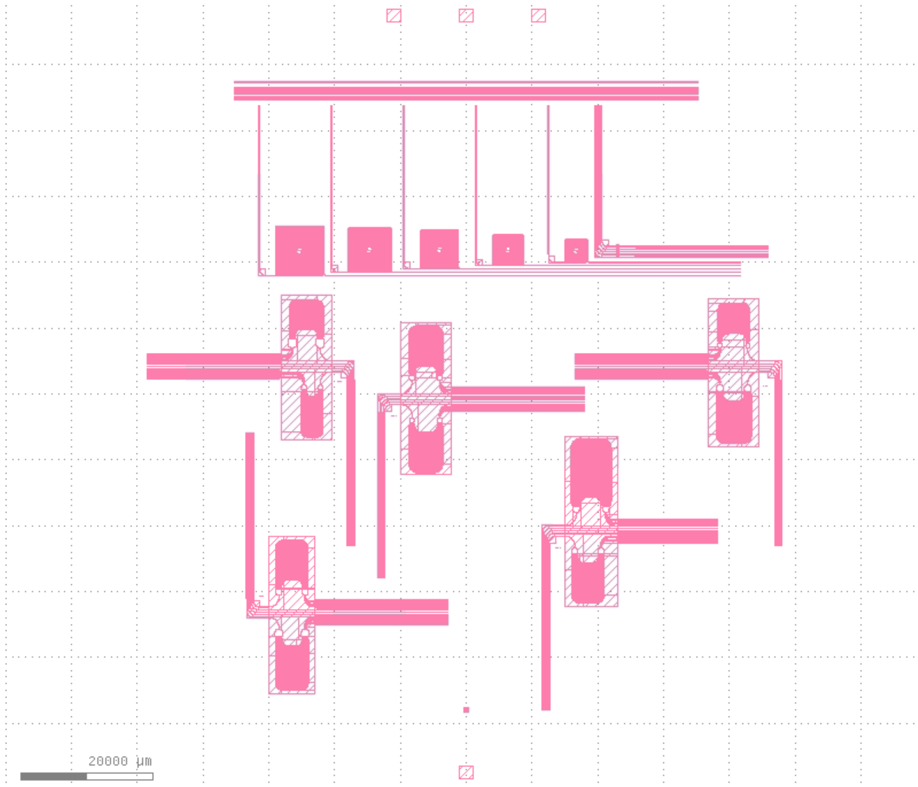


Figure 4.1: Waveguide routings on top of the patterns generated from PHASER and loss measurement spirals in the mask

mirroring, etc. It gives a lot of inspiration for my own routing design, for example, the square trenches in bending area and also the trench areas for waveguides which helps much with the proximity effect of my photolithography processes.

Fig.[**?**] shows the screenshot of one mask design with the routing algorithm managed to automatically route input output waveguides perpendicularly for easy measuring.

4.3 Waveguide Routing with Extra Turn Cost

4.3.1 Algorithms for Route Finding

4.3.1.1 A* for large grid numbers

In reality the gds mask is 100mm in dimension and the wire is 6 microns wide. Dijkstra is slow. I expect to figure out a way to apply A* which is friendly to my turn cost program.

4.3.1.2 Multi-Paths

What if there are more than two paths? This small project has reached the goal to find the one source shortest path for the graph. Moreover, if it is a two-sources problem. I tried and only managed to find one shortest path and then another by bypassing the first one and label the device boundary in the forbidden() list. Naively, comparison between the total loss can be made in the end and we can write a function to compare and finally choose one. When they will tend to cross each other, a program is needed to reason about the cost and whom to get priority to take the spot.

4.3.2 Implementation and Integration

This small program is aimed to solve shortest path in photonic wire routing. The goal is to develop a scheme that can take input such as wire width, design area, device area, starting points and ending points and return a shortest path accordingly.

4.3.2.1 Manhattan Routing and Wire Width

The application requires the Manhattan shape. Because angled routes tend to have discrete roughness along the edges due to the Cr mask limitation which will

introduce more propagation loss for light. The wire has width, normally several microns. So we need consider a graph with constant grid width.

4.3.2.2 Turns cost more

Each turn costs propagating light power because light will radiate out at bending. In essence, it is to find a lowest cost path for light propagation. We need to implement a method to choose as small amount of turns as possible.

4.3.2.3 No overlapped and bypassed devices

The routing is required to not bypass devices in the gds file and wires should not overlap with each other. I implement a class called PhotonicRouting() which takes width, area, device boundary and points from start to end as input using modified Dijkstra algorithms and return paths and cost accordingly. A function PlotPath() is chosen to display the data that we are finding through Routing() by the matplot library in python. At the bottom is a graph of diamond network which shows a plot implemented by assigned symmetrical coordinates for 4 direction nodes. On top of that is a collection of unavailable nodes due to device boundaries and the paths we already got from our own routing like two pictures below.

4.3.2.4 Boundary and Diamond nodes for each grid

A function NotAllowed() is applied to mark all direction nodes unavailable for routing paths visit for given boundaries of photonic devices in given gds. The boundary data is from initiation of the class. This function translates it into grid numbers and stored it for following routing in terms of dictionary in python.

A function Diamond() is utilized to give each grid 4 direction nodes(N,S,W,E) and labels weights for one turn as 10 (N to W, N to E, S to W, S to E) and weights

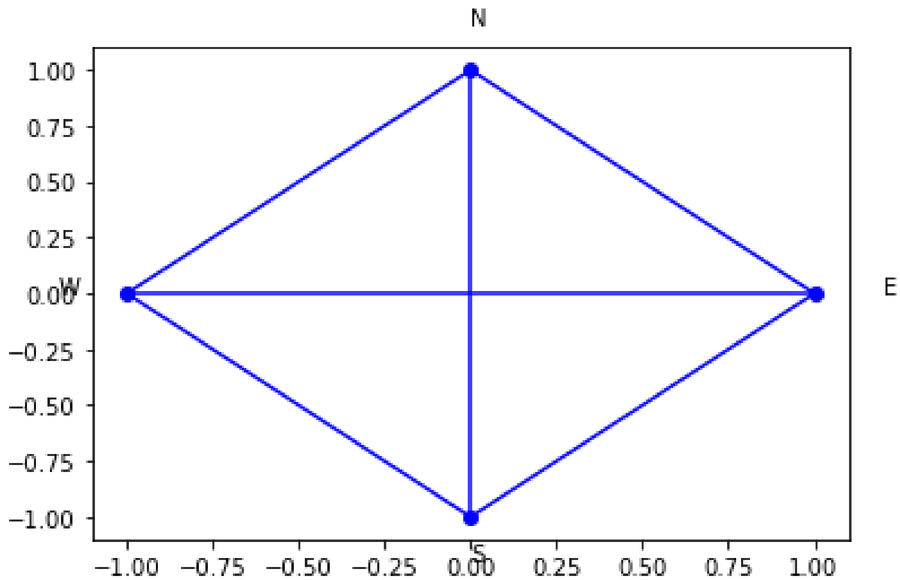


Figure 4.2: Schematic diamond design for each grid which is implemented in the code

for one straight connection as 1. Grid width is chosen according to given width of the photonic wire and grid number is converted with respect to routing area. Area information is provided by initiation of the class.

4.3.2.5 Modified Dijkstra

A function Routing() is implemented using adjacent matrix stored in a Network of Diamond nodes to traverse every adjacent vertex to find the lowest cost and absorb it in the path route. When the traversal reaches the end point, we get the route and stored it in the taken() list. Due to overlapping issues, I mark not only the points in route but also the direction nodes related to the grid which that route is crossing. This will mediate the overlapping concerns and allow the second or more paths to be added in the graph.

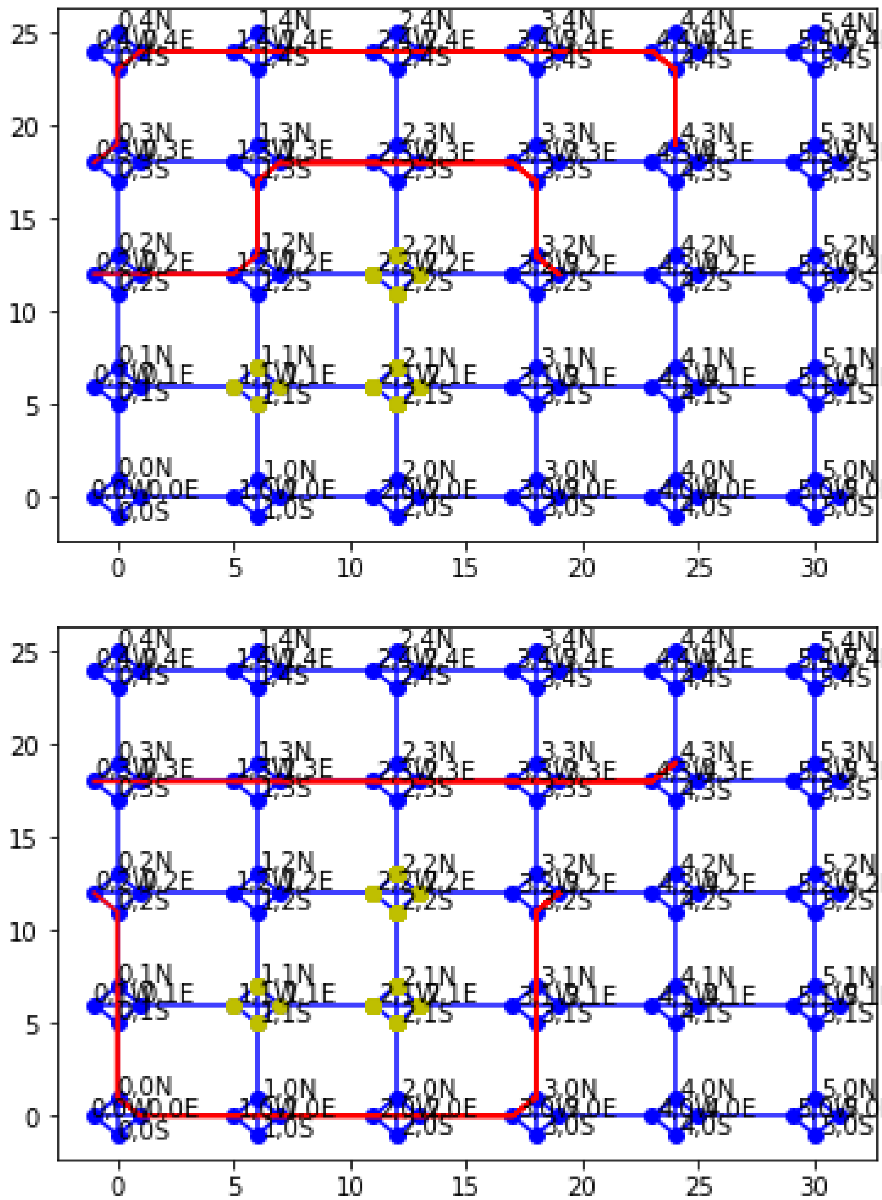


Figure 4.3: Comparison between routing results with the same starting and ending points but only with a different input order

Chapter 5

Fabrications of Array Waveguide Gratings

Previous chapters discuss the theory and design rules for AWG and introduce a way to route the waveguides once we generate patterns from the PHASER. This chapter I will summarize the process flow and guidelines for recipe improvements and sample characterizations like the microscopy. The fabrication process flow is demonstrate in the Figure 5.1 from step (a) to (i). The platform we are using is 300nm silicon nitride on 4um silicon dioxide on top of 500um silicon substrate. At step (a), we put material stacks into the 60 degree in Celsius Nanostrip to further clean the wafer stacks. Nanostrip residue will be minimized in the running water for 2 min. Then we use nitrogen to get rid of DI water and put wafers on 100 degree hot plate for 2 min. At step (b), we use electron beam(E-beam) Evaporator to deposit 200nm Chromium as a hard mask for dry etch of the silicon nitride layer. The recipe is in the Table 5.2. At step (c), the BARC is spined on the Cr mask and baked for 90s at 200 degree in Celsius. At step (d), after cooling down to room temperature, we spin on positive photoresist AZ701 on top of (c). The recipes and pre-baking temperature

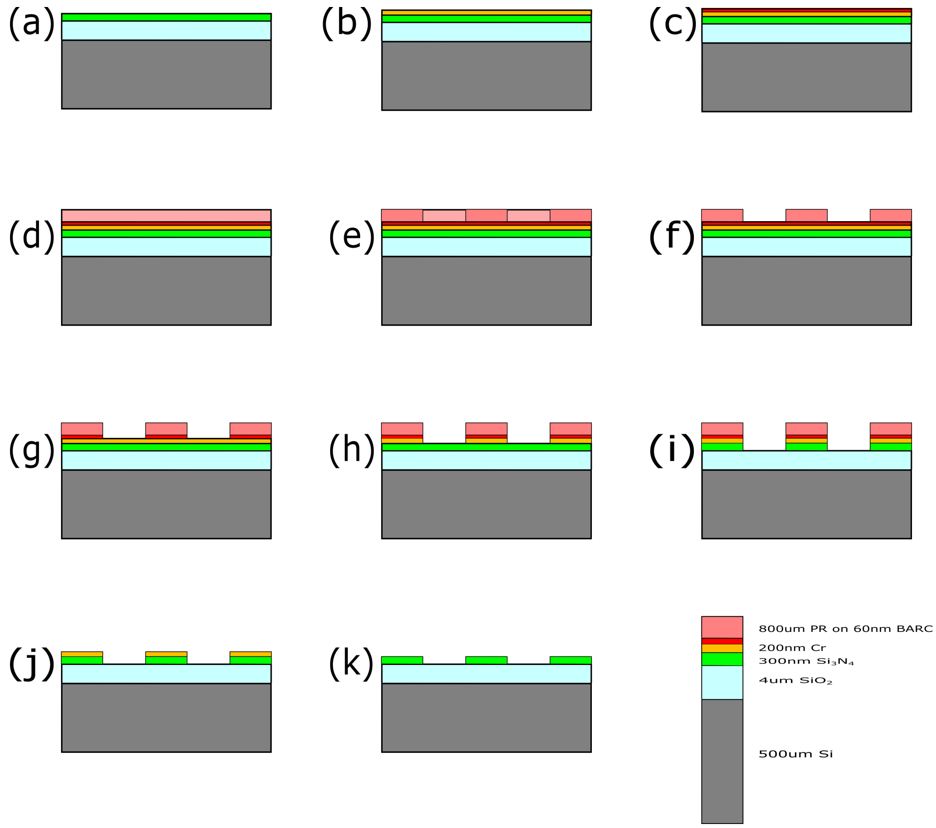


Figure 5.1: Process demonstration of an AWG device on the paltform of silicon nitride on silica

will be concluded in Table 5.3. At step (e), we use a 5:1 reduction stepper to expose UV light at 365nm. Post exposure bake is necessary since we will use AZ701 as a mask to dry etch Cr. At step (f), the region exposed to light will dissolve into develop AZ300. The pattern in the mask we designed will be transfered to photoresist. At step (g) and (h), the anti-reflection layer and Cr will be etched by an ICP etcher using recipe in Table 5.5. At step (i), the pattern got further transfered to the layer of the silicon nitride. This silicon nitride layer is our device layer. The effects of parameters in dry etch on sidewall roughness, anisotropy, selectivity and etch rate are investigated as summarized in Table 5.4. At step (j), the photoresist and BARC residue are removed in Nanostrip. The precess is the same as cleaning in step (a). At

step (k), the Cr mask is wet etched by commercial Chromium Etch. Then the wafer is cleaved into small chips for characterizations and measurements.

5.1 E-beam Evaporation

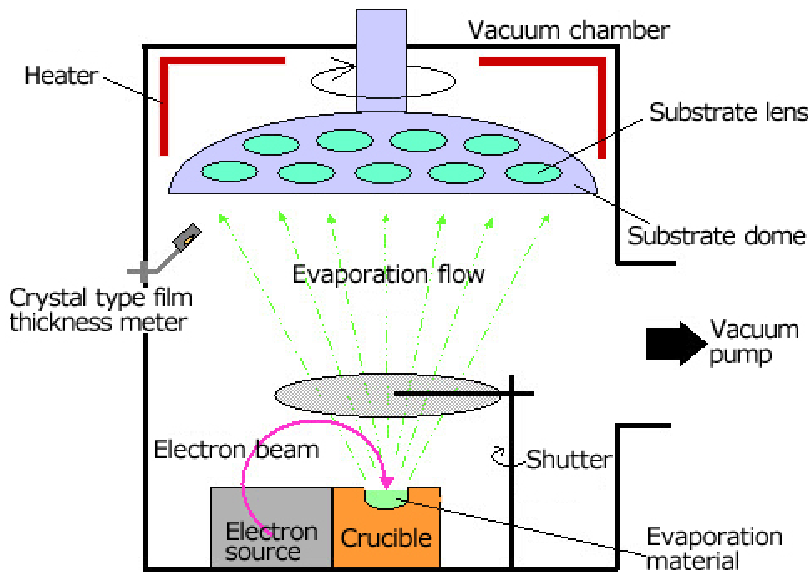


Figure 5.2: Schematic drawing of E-beam Evaporator from JEOL

As the schematic drawing in the Figure 5.2, the electron beam is hitting the precursor in the crucible and the evaperant material is heated directly. The evaporation flow is radiated from raw materials in the vacuum chamber. Substrate will be rotating and the chamber can be heated. The Crystal Film Thickness Meter is employed and used as a feedback to adjust tooling factor in recipes. E-beam evaporation is more efficient than filament due to direct e-beam heating. Because heating with resistance can not control the power immediately, the thickness control is more accurate as for E-beam evaporation. Moreover, the beam can be easily and quite precisely controlled by the electric magnetic field. Due to its high density, materials with high melting point is possible.

Material	Chromium (Cr)
Atomic Weight	51.996
Thermal Conductivity	94W/m.K
Melting Point	1857 Celsius
Density	732 g/cc
Temperature for Vacuum Pressure	10^{-6}:977 Celsius

Table 5.1: Properties of Cr are good for e-beam evaporation

Recipe Name	Depo Process 2
Crossover Pressure	100 mTorr
Deposition Pressure	3 uTorr
Max Temperature	150 Celsius
Gas Dead-band	2%
Gas Stabilization	10 s
Tooling Factor	150
Film Thickness	200 nm

Table 5.2: The Recipe for Cr mask Deposition

Below is attached Table 5.2 of my recipes for CVC E-beam Evaporator here at COMSET, Clemson. The capability of evaporation for Cr is investigated and the Vacuum Pressure is chosen to be 3 mTorr. Tooling facter is sdjusted 150 to have 200nm Cr deposition. In our fabrication, we are using Chromium as a hard mask to dry etch the silicon nitride layer. So we need a rough thickness control on Cr. Two reference wafers are deposited along the silicon nitride wafers. These wafers are also cleaned by Nanostrip. Only difference is that reference wafers are 500um plain silicon and the sample wafers are 300nm silicon nitride on 4um silicon dioxide on top of 500um silicon.

5.2 Lithography

5.2.1 GCA Stepper

GCA Stepper is capable of 5:1 reduction i-line exposure (UV 365nm). I design some test patterns to evaluate the performance of it in terms of critical dimension(CD) and overlay accuracy. The pattern is showed in the center of the mask in Figure 5.3. We have patterns in circles, squares, and lines with CD from 0.5um to 2.0um. We find out this stepper have 0.6um resolution and the limit we have pushed is 0.5um with detailed care of dose and focus. The stepper have 4 blades that we can control the frame opening through job editing.Therefore, we can have several designs in one mask and select the region in a mask that needs to be exposed. As any other lithography, the focus and dose are essential for photoresist to resolve the pattens. Beam intensity will change due to life of UV lamp, however, dose keeps the same for desired performance. Using the capability of interchange of focus and shutter time between each exposure in an array, we can have massive samples to compare in the scanning electron microscope. The better profile and CD achieved as we expose 0.26s using beam of 280 $mJ/cm^2/s$ at -0.3um focus.

5.2.2 Photoresist Recipes

Referenced with spin curves in online database of these commercial Brewer's anti-reflection layer(BARC) and ZA701 positive photoresist, below is edited recipes for spin precesses in the COMSET Cleanroom. Anti-reflection layer is crucial for our process. Since we are patterning on top of a reflective metal, chromium, the standing wave would be introduced and makes the sidewall of photoresist as the shape of saw. The layer thickness of ARC is chosen to be 60nm given reflectivity curve from Figure

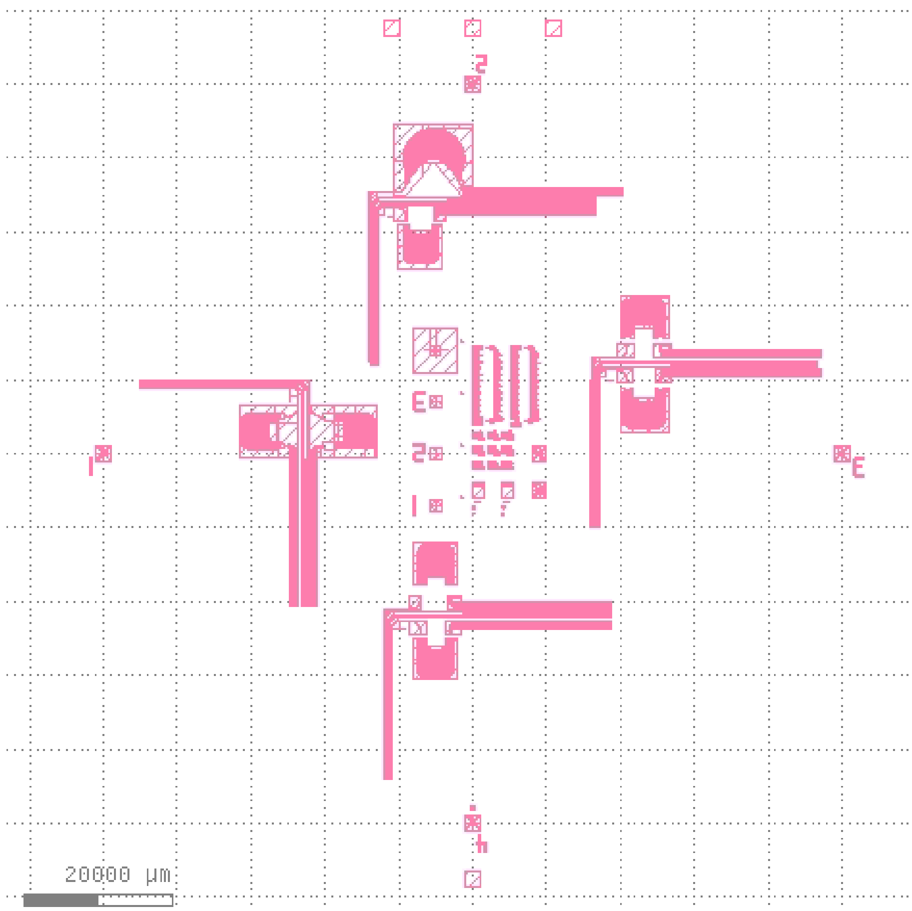


Figure 5.3: Mask design for CD measurements and routed AWGs with consideration of proximity effect

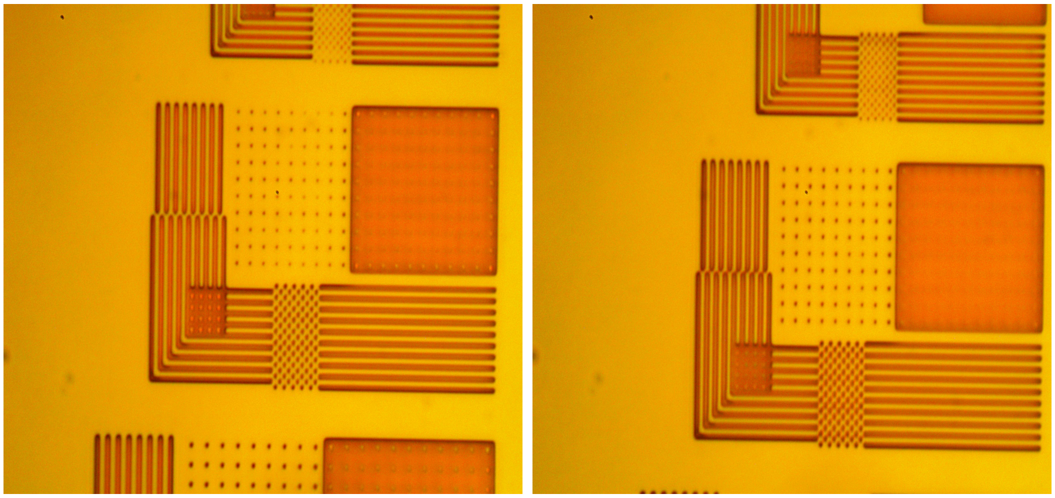


Figure 5.4: The pattern with 0.6um CD while exposed for 0.275(left) and 0.325(right) at 260 $mJ/cm^2/s$

5.5. And the layer thickness of AZ701 is chosen to be 800nm which is thick enough based on etching selectivity for 60nm ARC and 200nm Cr. The post exposure bake is important for the process as well. It will further cross link the photo resist making it fully cured. It will increase the selectivity of photoresist as a mask in the plasma dry etching.

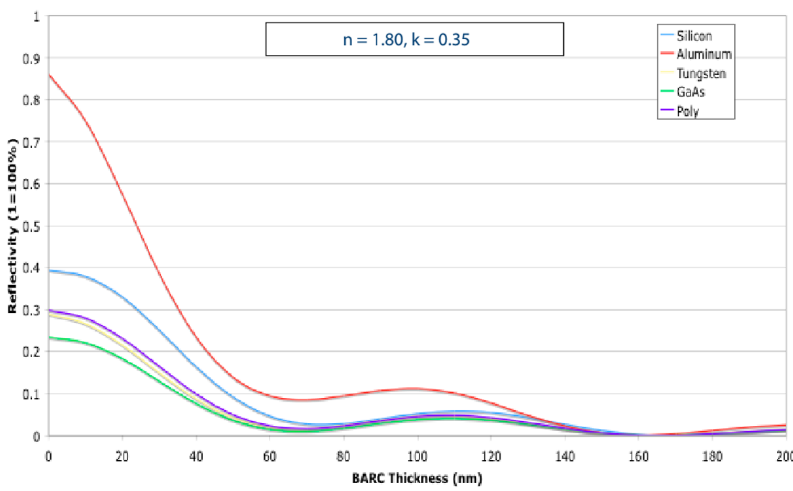


Figure 5.5: Reflectivity curve with respect of BARC Thickness

Photoresist	BARC	AZ701
Thickness	60 nm	800 nm
Baking Temperature	200 Celcius	110 Celcius
Baking Time	90 s	75 s
Cooling Down	2 min	1 min
Spinning RPM and Time	1900 rpm for 60 s	2500 rpm for 45 s

Table 5.3: The Process Recipes for BARC and AZ701(The time and temperature for post exposed bake is the same as pre-baking)

5.3 Dry Etch

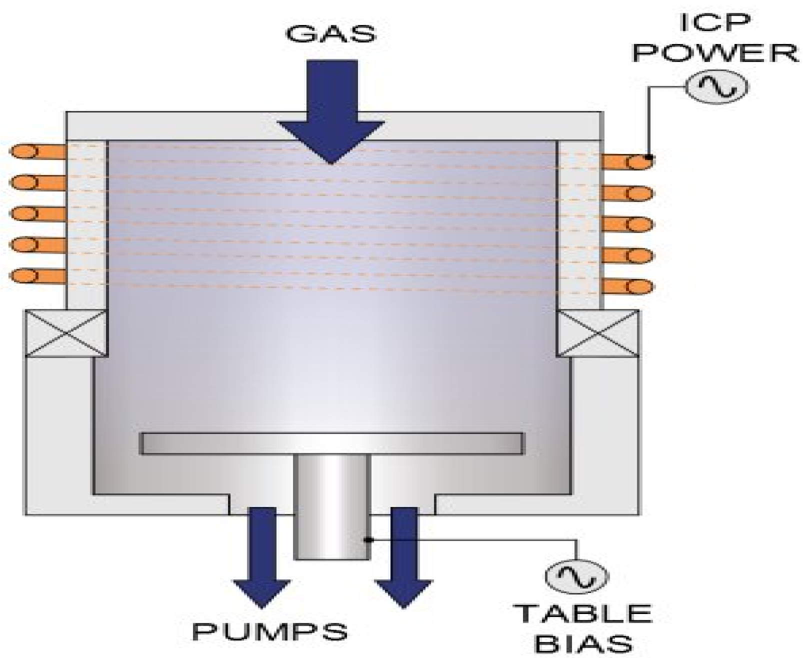


Figure 5.6: Schematic Dawing of ICP from Oxford

We have been using Oxford ICP 100 for 4 inch wafers. As in the Figure 5.6, gas comes out from the shower head of upper lid with ICP forward power and goes through the RF field between two lids and then hits the wafer. The storm of plasma has both chemical reaction and physical bombardment with the target wafer. There are so many factors to influence the etching performance. After numerous etching

	Inductive Power		RF Power		Pressure		CF_4		CHF_3	
	HI	LOW	HI	LOW	HI	LOW	HI	LOW	HI	LOW
Resist Etch Rate	HI	LOW	HI	LOW	LOW	HI	HI	LOW	LOW	HI
Si3N4 Etch Rate	HI	LOW	HI	LOW	LOW	HI	HI	LOW	LOW	HI
Anisotropy	LOW	HI	HI	LOW	LOW	HI	-	-	HI	LOW
Roughness	*	*	LOW	HI	HI	LOW	LOW	HI	HI	LOW
SiN:Resist	HI	LOW	LOW	HI	*	*	*	*	HI	LOW
Plasma Stability	HI	LOW	HI	LOW	HI	LOW	*	*	*	*
* means that the results did not show trend										

Table 5.4: Summary for parameters adjustment of Silicon Nitride Etching Recipes

comparisons, there will have a trend for a specific variation in recipes which I attached in Table 5.4 as a guideline for the etching improvement.

Plasma stability, sidewall roughness, etching rate, selectivity, anisotropy are all of importance for our etching results. Ideally, we expect more stability, selectivity, and anisotropy with less trade-offs of the sidewall roughness and etching rate. Associated with the individual etching chemistry, dry etching recipes for Cr and Silicon Nitride are in Table 5.5 and 5.6, respectfully.

5.3.1 Cr Dry Etch

The Chromium Mask is deposited with 200nm thickness. The measurement is done in a profilometer. It was dry etched 8 min for several silicon reference wafers. Then 8min is applied with the sample silicon nitride wafer. Chromium mask is etched in an Oxford 100 ICP machine and the recipe is in Table 5.5 with etching rate around 25nm/min.

Pressure	ICP Power	RF Power	Temperature	Cl_2	O_2	He
10mTorr	400 W	20 W	25 Celsius	26 sccm	4 sccm	10 sccm

Table 5.5: Dry etching recipe for Chrome with 25 nm/min etching rate

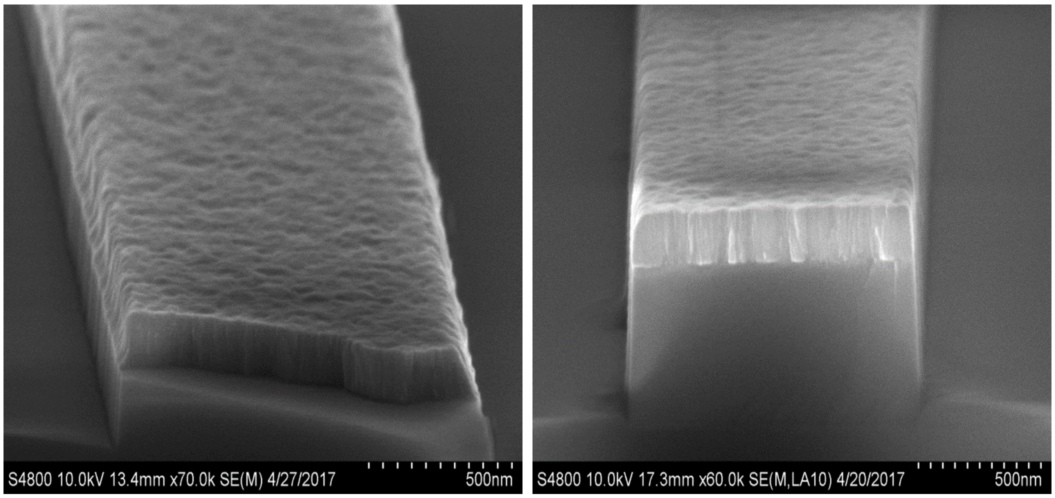


Figure 5.7: Comparison between the 30s and 1min etching for the silicon nitride platform

5.3.2 Silicon Nitride Dry Etch

Using the guideline above, several experiments have been done and SEM of samples are compared. We eventually adopt the recipe in the Table 5.6 As you can witness in Figure 5.7, the silicon nitride is further etched almost 100nm more which gives us 200nm/min etching rate. The anisotropy is quite good and the sidewall is almost perpendicular to the surface.

Pressure	ICP Power	RF Power	Temperature	CHF_3	O_2	He
10 mTorr	600 W	60 W	125 Celsius	65 sccm	15 sccm	5 sccm

Table 5.6: Dry etching recipe for Si_3N_4 with 200nm/min etching rate

5.4 Scanning Electron Microscope

Scanning Electron Microscope(SEM) employs an electron gun in an electric field and these electrons are accelerating towards sample target. Electromagnetic(EM) field serves as condenser lens just like the way optical lens focus light rays. In between two condenser lens, there is a Spray Aperture to further control the opening for the electron beam. Deflection coils use EM field to steer the e-beam which have the same function as objective lens in the optical system.

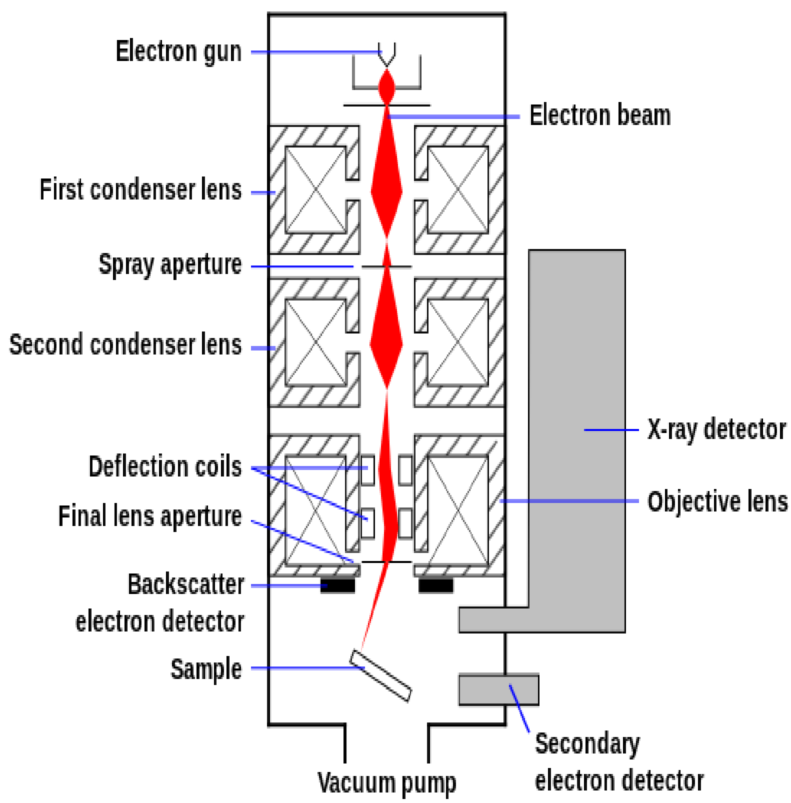


Figure 5.8: Schematic drawing of Scannig Electron Microscope from Wiki

The main signals SEM is probing are back-scattered electrons(BSE), secondary electrons(SE) and X-ray as given in Figure 5.8. After hitting the sample target, several signals are generated from primary electrons like in Figure 5.9. The SE gives

us surface and topographical informations and BSE gives us more information about depth and material differences. The schematic also shows the penetration of these signals which will guide our emphasis in the measurements.

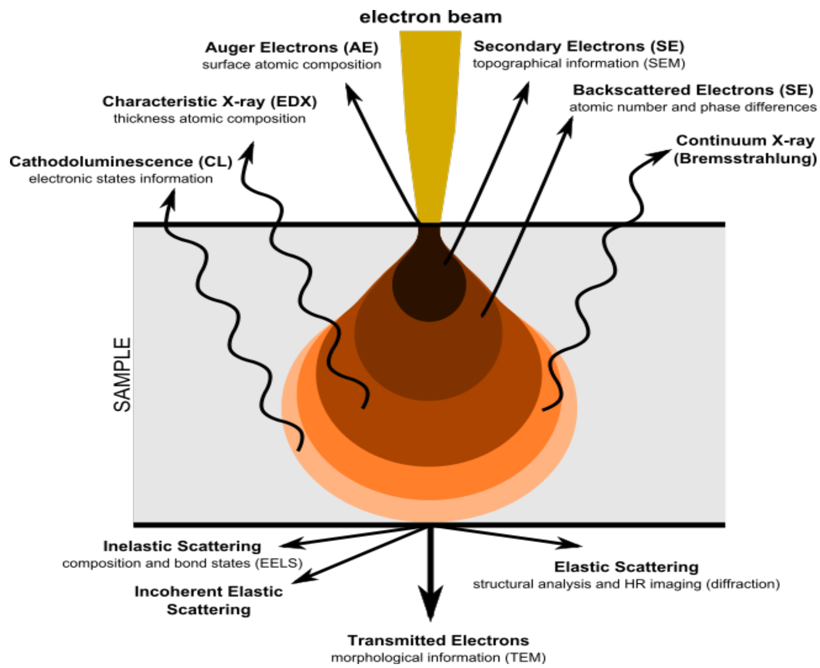


Figure 5.9: Signals available in the SEM

The working distance, object aperture size, strength of condenser lens , accelerating voltage are main parameters for us to adjust towards better images such as the topology details, sharpness, resolution, probe current, depth of the field(DoF) and the consideration of charging effect and light contamination. The adjustment of contrast, focus and stigma will secure a clear, and balanced image. A summary of compact table to serve as a guideline when I do the characterization of microchips in the SEM is in Table 5.7.

	Condenser Lens		Accelerating Voltage		Working Distance		Objective Aperture	
	HI	LOW	HI	LOW	HI	LOW	HI	LOW
SE	LOW	HI	-	-	-	-	HI	LOW
BSE	LOW	HI	HI	LOW	LOW	HI	HI	LOW
X-Ray	LOW	HI	HI	LOW	LOW	HI	HI	LOW
Surface Detail	-	-	LOW	HI	-	-	-	-
Topology	-	-	HI	LOW	-	-	-	-
Sharpness	HI	LOW	HI	LOW	LOW	HI	LOW	HI
Resolution	HI	LOW	HI	LOW	LOW	HI	LOW	HI
Current	LOW	HI	-	-	-	-	HI	LOW
Charging	LOW	HI	HI	LOW	-	-	HI	LOW
Spot Size	LOW	HI	LOW	HI	HI	LOW	HI	LOW
DoF	-	-	-	-	HI	LOW	LOW	HI

Table 5.7: Summary for parameters adjustment of SEM

Chapter 6

AWG Characterization and Measurement

6.1 Experiment Setup

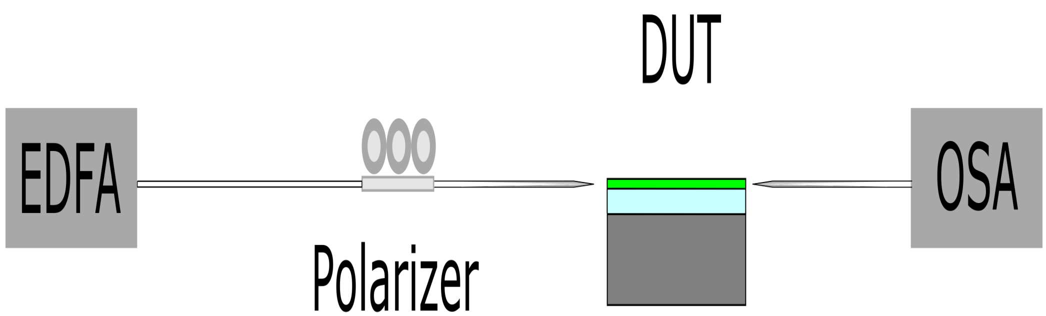


Figure 6.1: Measurement setup for AWG devices

As demonstrated in Figure 6.1, I use Erbium Doped Fiber Amplifier(EDFA) to provide broadband signals from 1500nm to 1700nm. Then the EDFA is connected with a Polarizer to select the designed TE single mode for AWG. The Device Under

Test(DUT) is our patterned silicon nitride AWG device. To couple signals in and out of waveguides of DUT, lensed fibers from OZ optics are implemented with 2 3D precision stages from Newport. After lensed fiber collects signals from DUT, the spectrum response is measured in Optical Spectrum Analyzer(OSA).

6.2 Characterizations

In Figure 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 6.10,and 6.11, waveguides with separation of 2.0um will have less problem. However, all of them can not resolve completely which introduces huge insertion loss and phase error as well at the transition region. The results measured by setup above are provided in Figure 6.12,6.13,6.14,6.15, and6.16.

As we can witness that the comparison shows the limitation of resolution of GCA stepper between the mask layout and photoresist patterns. This limitation hinders the design of less than 0.5um gap between two taper tails to improve insertion loss at transition region between FPRs and array waveguides. The measurement shows huge crosstalk due to phase error which damages the image plane at receiving waveguides. This is due to the fact that photoresist can not resolve patterns that are designed to expose.

All the designs are then edited to have 0.5nm gap between adjacent taper tails due to this feedback from these fabrications. The new designs have more insertion loss according to simulation results by Beam Propergation Method(BPM) in Figure 6.18 and 6.19. However, microscope pictures do show that photoresist can resolve these patterns if we design an AWG mask to have at least 500nm gap which is concluded as the CD limit for our stepper in Figure 6.17.

6.3 Measurements

The best AWG perfomance that we have measured from our devices is showed in Figure 6.2. The insertion loss of AWG is 8 dB. The average channel spacing

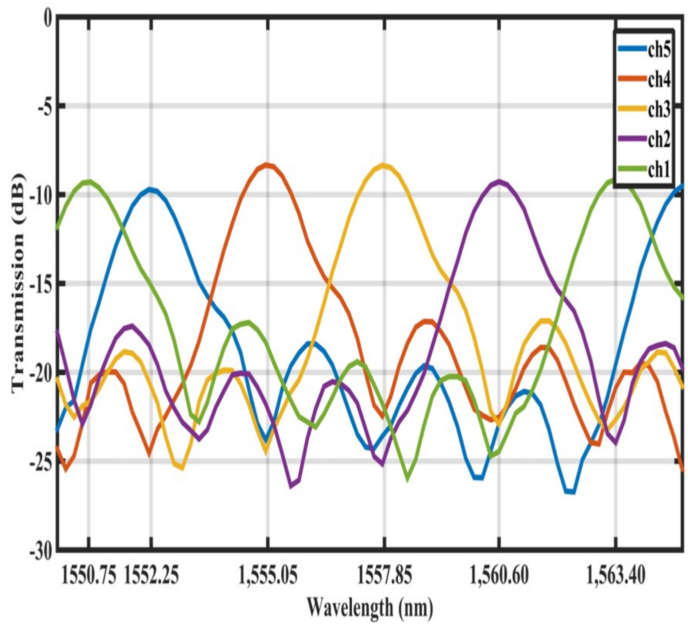


Figure 6.2: Passband of designed 5 channels AWG with 350GHz channel spacing measured using setup in 6.1

is around 2.8nm. 3dB channel bandwidth is larger than 1nm. This AWG design is measured with FSR 12.65nm which is designed to be $5*350GHz$. Crosstalk is around 19 dB. Due to the fact that two outer channels have less performance which cause the Nearest Crosstalk drop to 4 dB. The central channel will have 7-8 db Nearest Crosstalk. Since this is a cyclic-FSR AWG design, which has maximum 2 dB Non-uniformity. EDFA is directly connected to OSA to provide a reference spectrum. Then the spectrum of two reference waveguides with different length are measured. The propagation loss and coupling loss are calculated. Due to the fact that this device is measured without input tapers of AWG, so the coupling loss per facet is 11 dB and

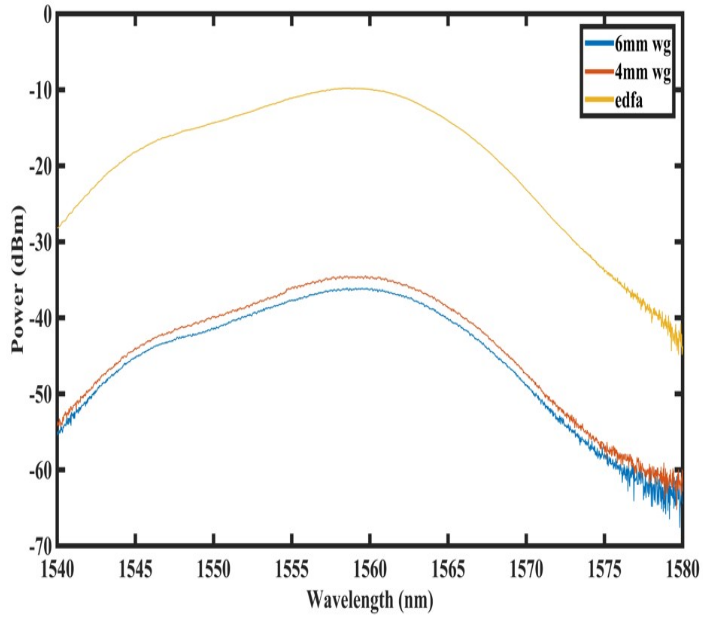


Figure 6.3: OSA Spectrum for EDFA and reference waveguides with different length

the estimated propagation loss is around 0.7 dB/mm as demonstrated in Figure 6.3.

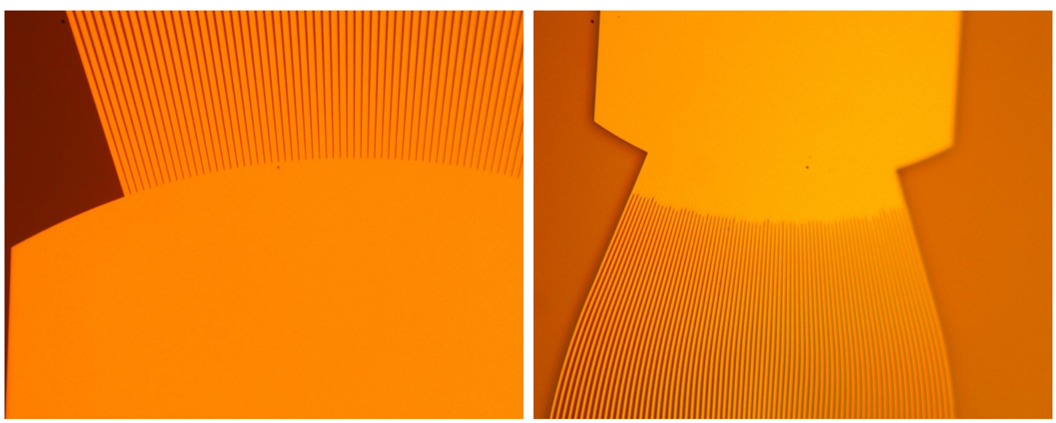


Figure 6.4: Mask(left) and photoresist(right) patterns of A1 in 50X Microscope

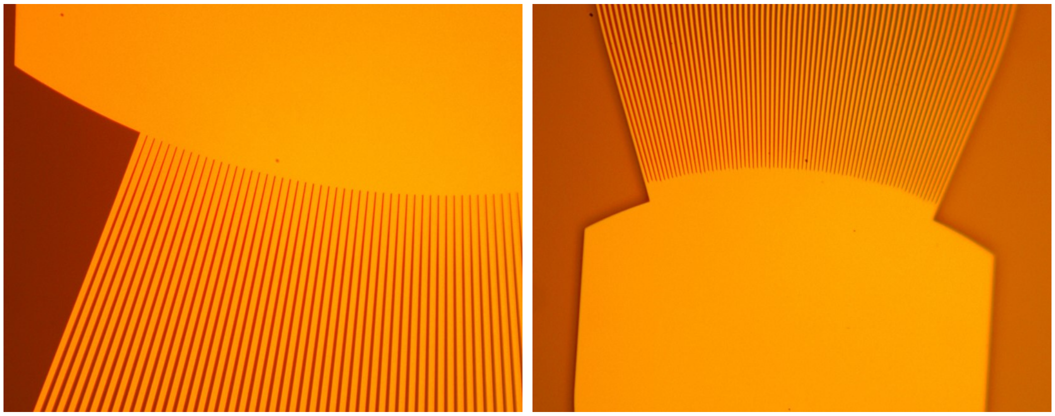


Figure 6.5: Mask(left) and photoresist(right) patterns of A2 in 50X Microscope

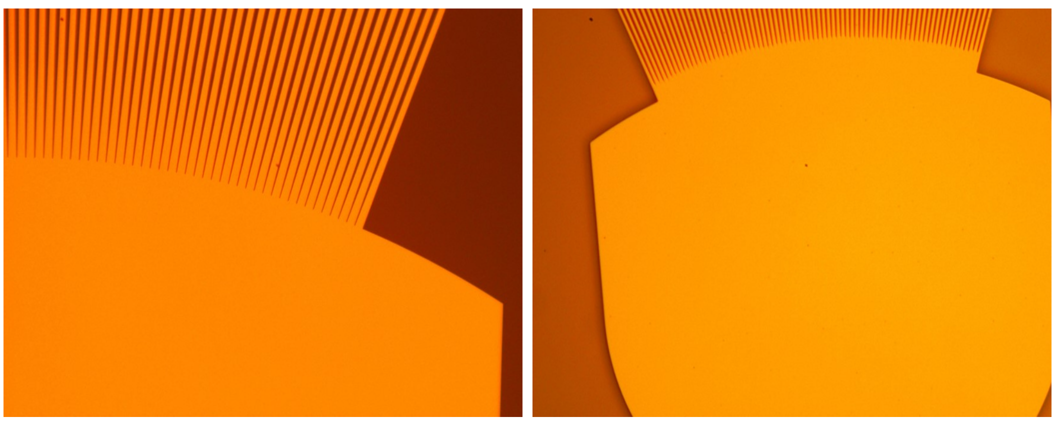


Figure 6.6: Mask(left) and photoresist(right) patterns of A3 in 50X Microscope

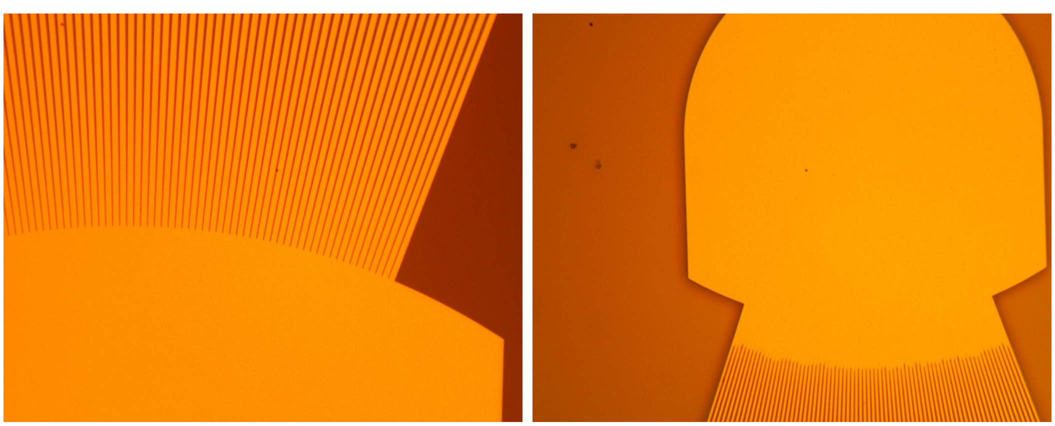


Figure 6.7: Mask(left) and photoresist(right) patterns of A5 in 50X Microscope

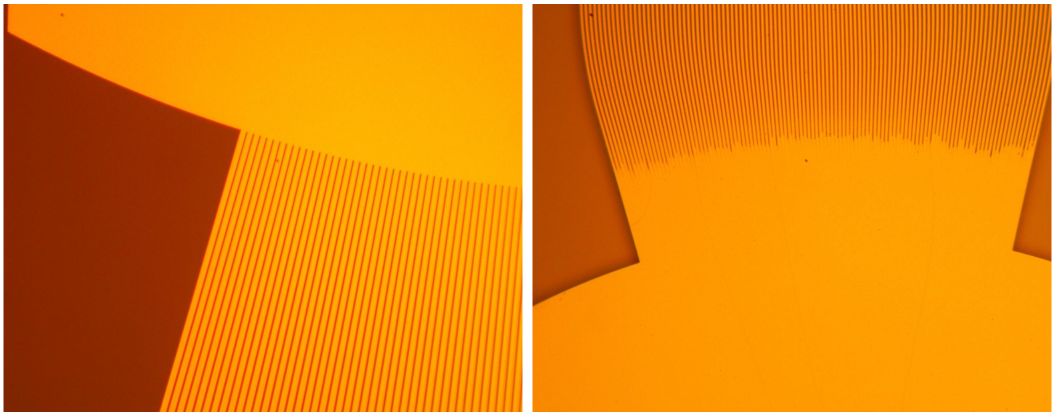


Figure 6.8: Mask(left) and photoresist(right) patterns of A6 in 50X Microscope

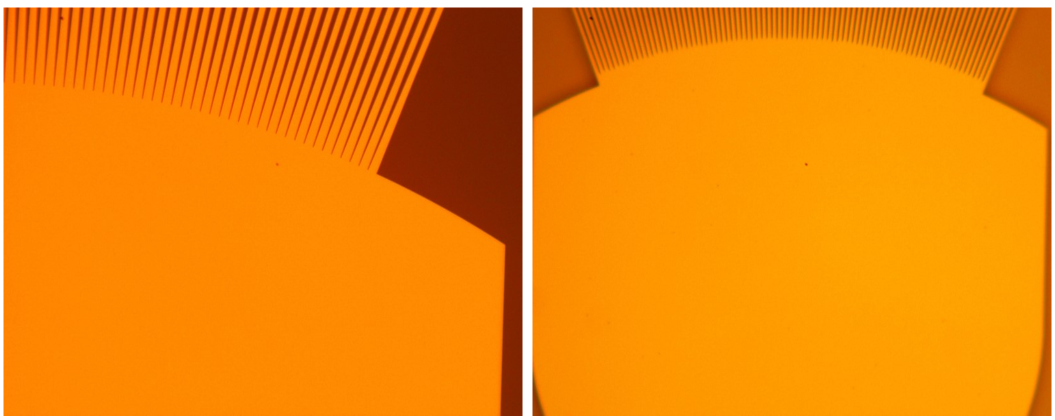


Figure 6.9: Mask(left) and photoresist(right) patterns of A7 in 50X Microscope

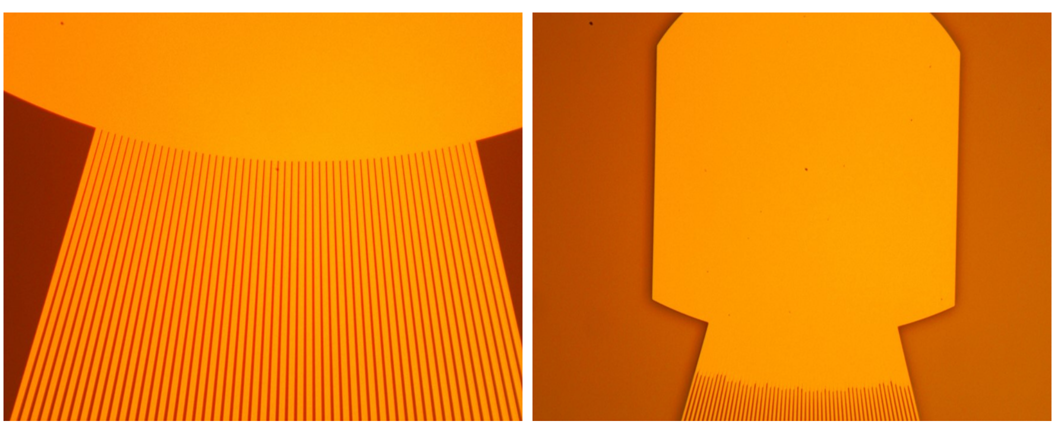


Figure 6.10: Mask(left) and photoresist(right) patterns of A8 in 50X Microscope

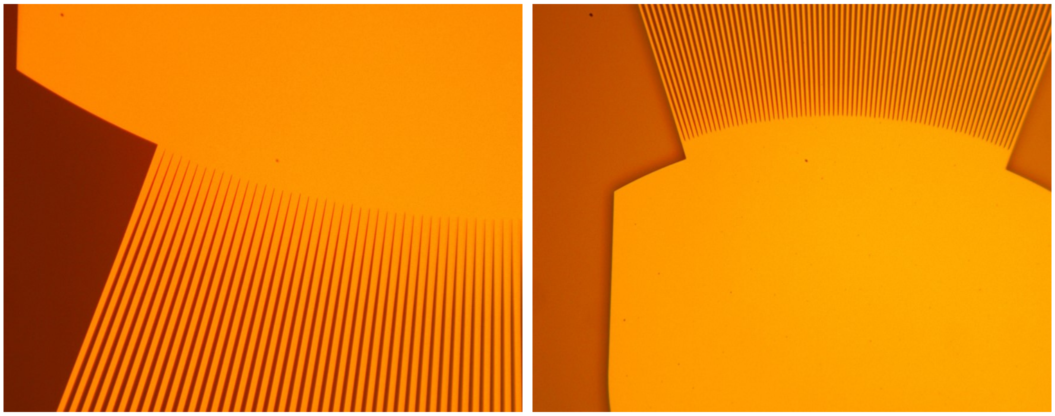


Figure 6.11: Mask(left) and photoresist(right) patterns of A9 in 50X Microscope

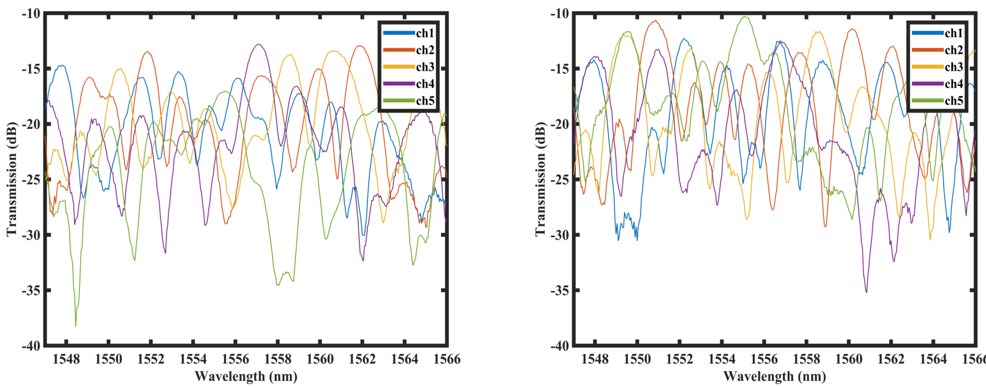


Figure 6.12: Transmission response with respect of wavelength for fabricated A1(left) and A2(right)

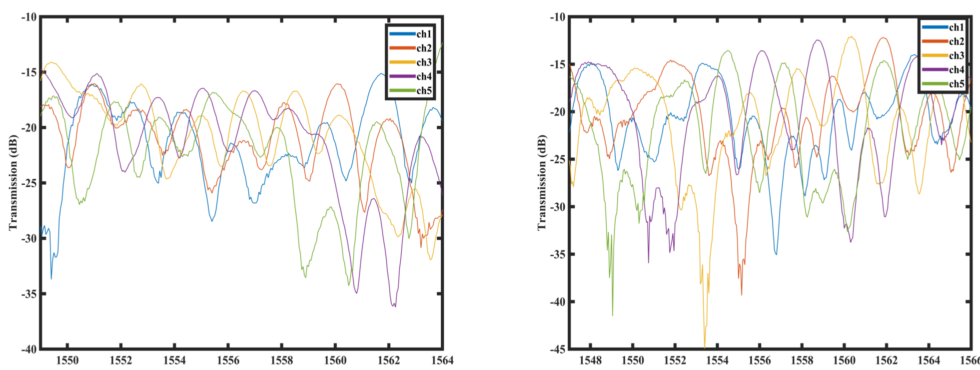


Figure 6.13: Transmission response with respect of wavelength for fabricated A3(left) and A4(right)

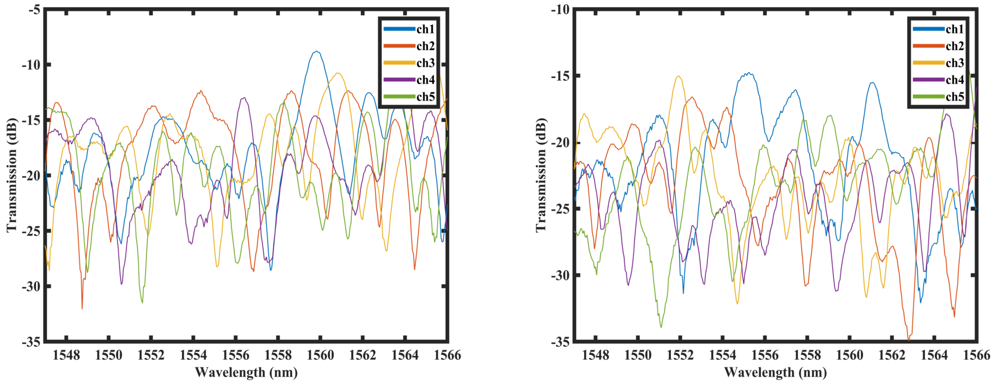


Figure 6.14: Transmission response with respect of wavelength for fabricated A5(left) and A6(right)

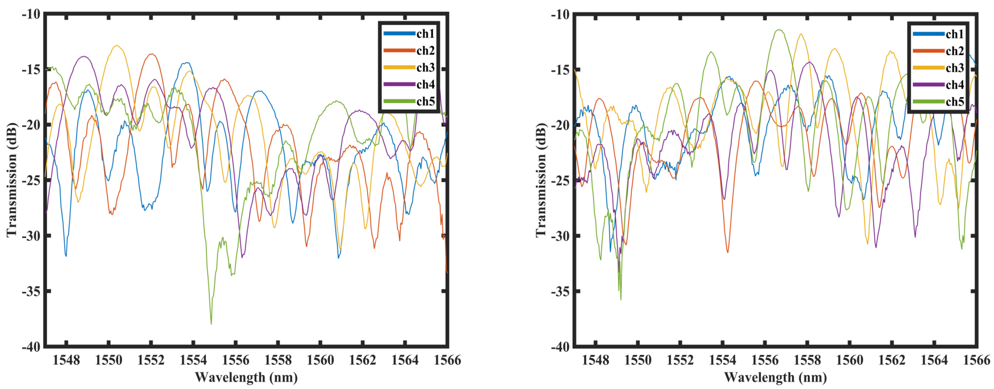


Figure 6.15: Transmission response with respect of wavelength for fabricated A7(left) and A8(right)

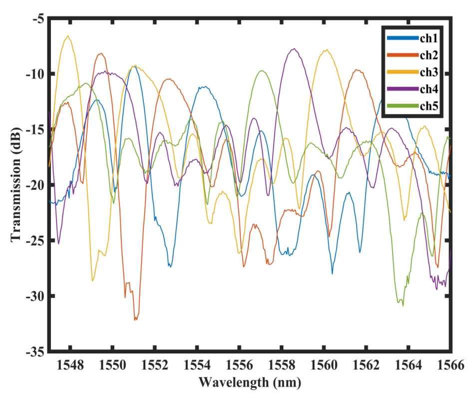


Figure 6.16: Transmission response with respect of wavelength for fabricated A9

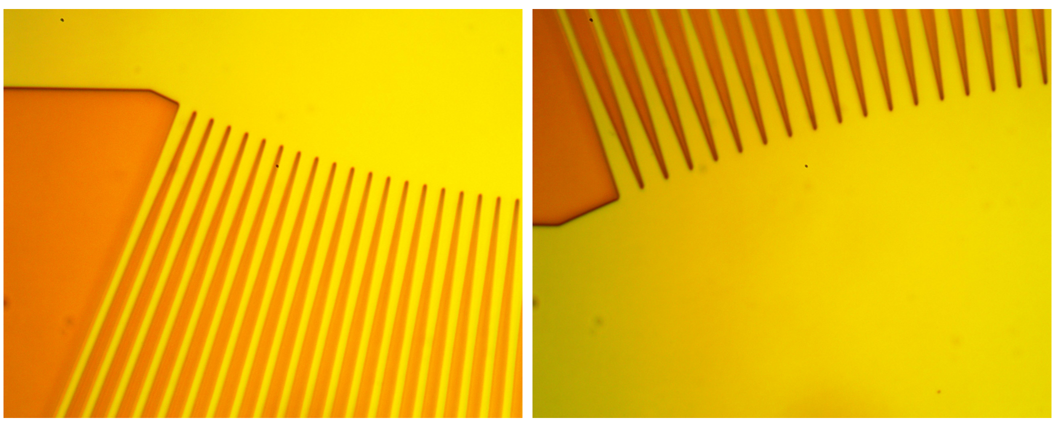


Figure 6.17: AWG P2_A1(left) and AWGP3_A56(right) with 0.5um gap between taper tails at transition junctions

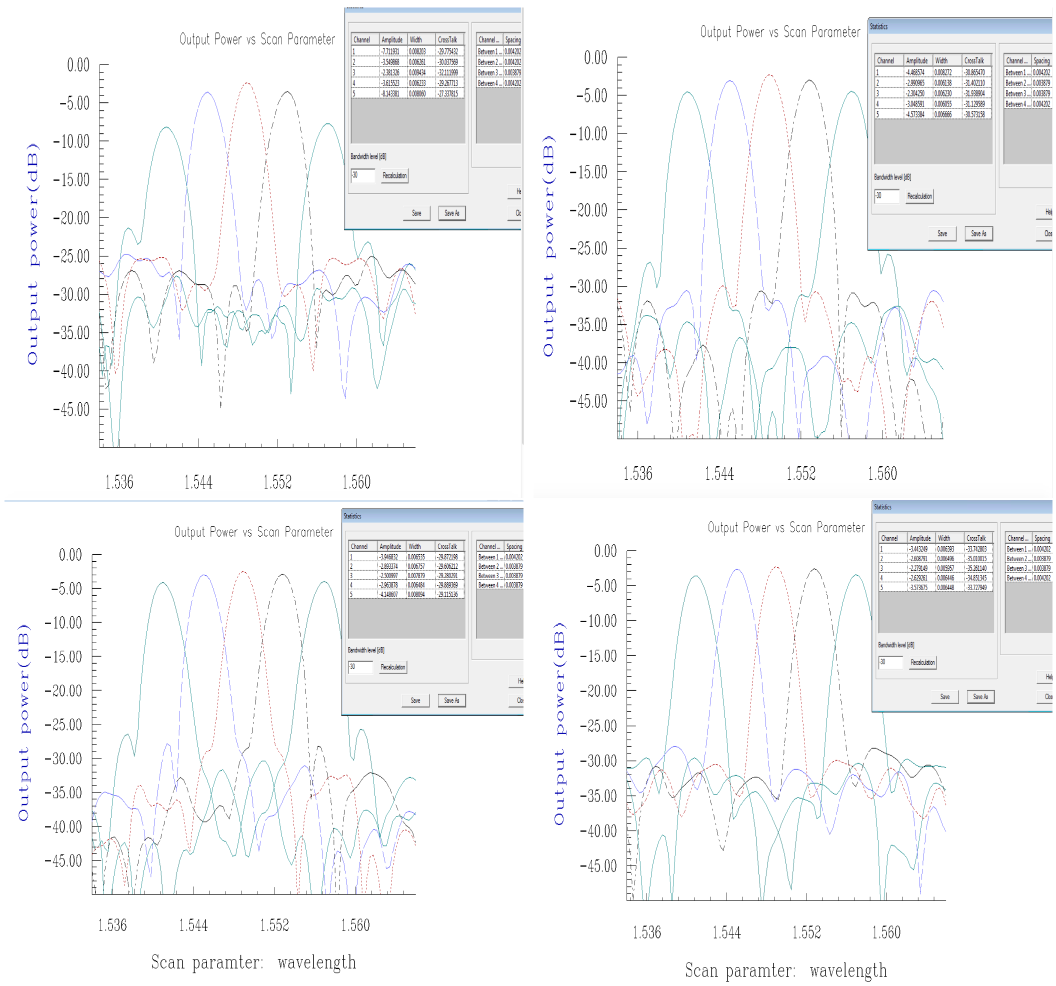


Figure 6.18: 5 channls design of the AWG with da 1.7um(left), 2.5um(right), 3um(lower left), and 3.5um(lower right) with spacing 400GHz at 1550nm

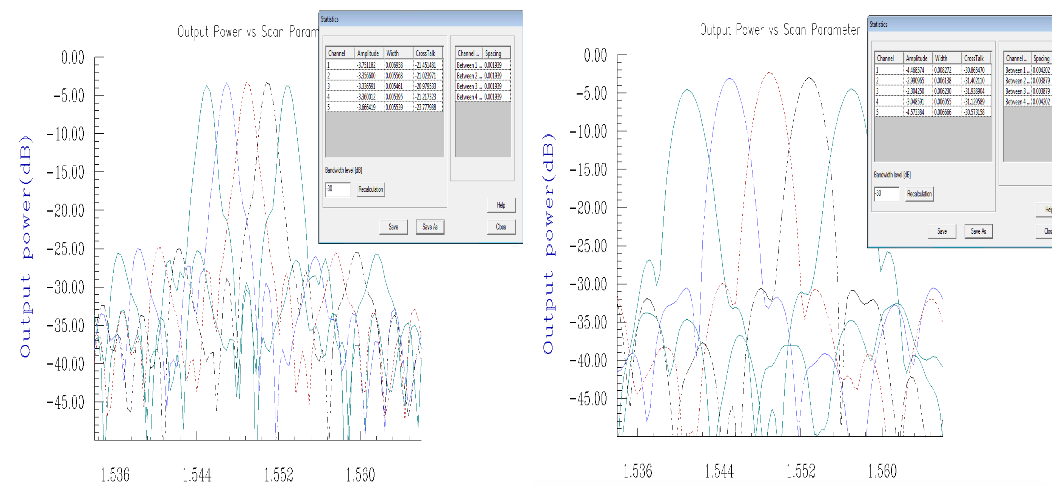


Figure 6.19: 5 channls design of the AWG with spacing 200GHz(left) and 400GHz(right) with 2.5da at 1550nm

Chapter 7

Applications and Future Work

7.1 Wavelength Beam Combining(WBC)

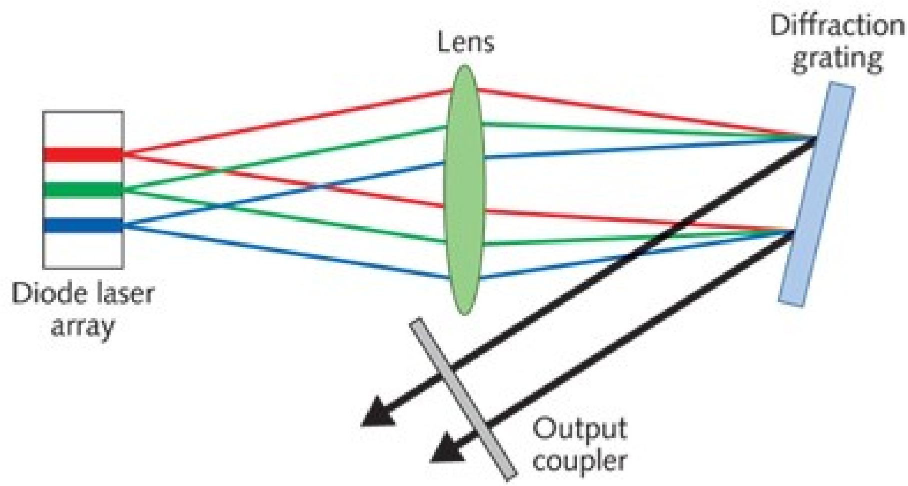


Figure 7.1: External Wavelength Beam Combining[9]

My work on designs and fabrications of AWGs is expected for the hybrid integration with laser diodes which our group proposes to create an on-chip Wavelength Beam Combining(WBC) system by replacing the external cavity in free space as showed in Figure 7.1 with an AWG as the schematic drawing in Figure 7.2. For this

application, we are using the capability of wavelength division of an AWG reversely. Laser diodes are directly coupled to the input channels of the AWG with high coupling efficiency. The phase arrays with curved junctions provides the same wavelength dispersion function as these diffraction gratings. Input channels and free propagation region(FPR) serve the same way as the transform lens in the system. Our group proposes to deposit thin films on facet to eliminate the reflection at the coupling interface between the lasers and waveguides. Therefore, these lasers for each channels can only have feedback from the output end of the AWG, which is to replace the function of the output coupler in the Figure 7.1[9]. It dramatically reduces the size for a WBC system by replacing all above with a compact AWG passive chip. Also, AWG is easily scale up but still in a compact format. AWG with 1000 channels and less than 0.08nm channel spacing have already been reported.[17] This means that more than 1000 channels of gain chips can be combined and lase from one waveguide output over a spectral span of 80nm. In essence, AWG is a filter. It has a passband as any other filters with respect of different wavelengths which we can use the theory and tools mentioned in previous chapters to design and fabricate. Different gain chips operate at different channels which have corresponding lasing wavelength within specific range of passbands of the AWG. All the optical power from different channels will be combined together in the output waveguide which has same function as the way that the output coupler forces a co-axial propagation of different beams in the free space WBC system. It can realize the beam combining and wavelength control at the same time. We can enlarge the separation of phase arrays and input waveguides so that there are no coupling between these laser channels. Every channel can be individually operated, which provides a great flexibility for wavelength control as a WBC system. This can serve as a compact multi-wavelength source with flexibility and potential of scalability.

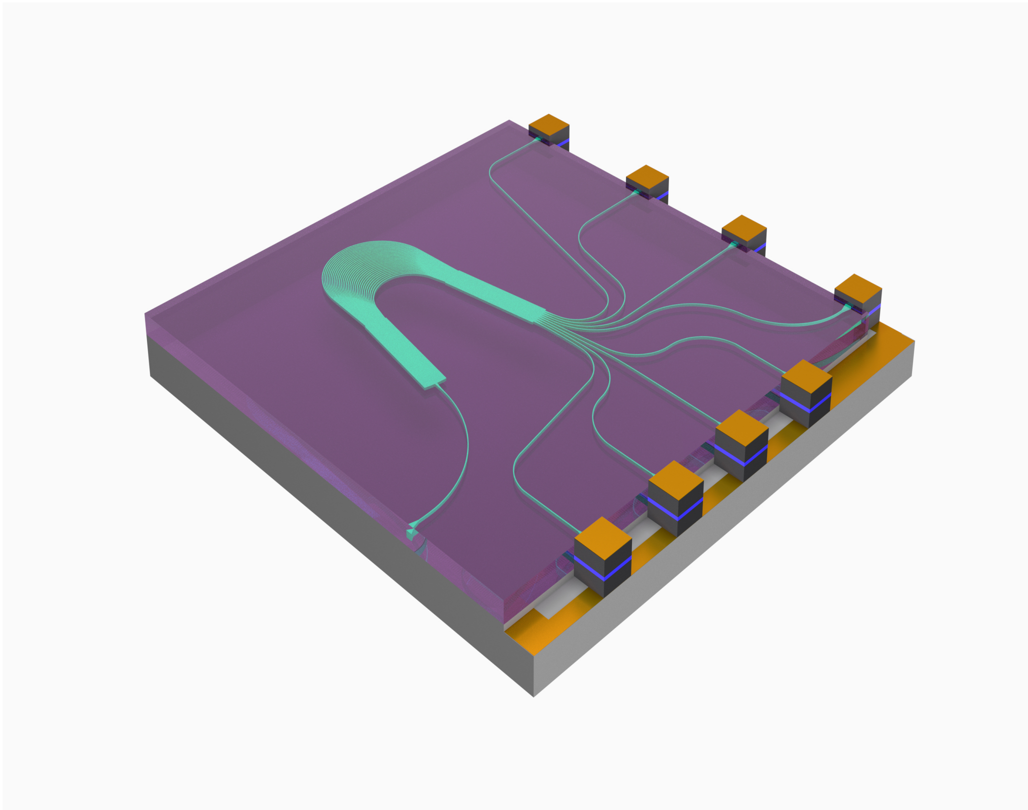


Figure 7.2: The schematic drawing of our proposed Wavelength Beam Combing

7.2 Future Improvements

- Adiabatic tapers from straight to bending waveguide

Because the dn/dw is decreasing as we increase the width of waveguides as in the Figure 3.3 [16], we can use width at multimode region for straight waveguides. Bypassing the trade-off of excitation of higher order modes, we can use adiabatic tapers transiting width at multimode regions to the single mode region from free propagation regions to straight waveguides and from straight waveguides to bending waveguides.[26] The rest of AWG is mirror symmetric regarding the center line across the AWG. By doing this, we can reduce the phase error due to dimension variation of the width of the fabricated waveguides. Moreover, less intensity will be at the sidewall if we

increase the width, so less scattering loss from the rough sidewall.

- Customized circuit by Python

Restricted by the PHASER, we cannot do the same bending radius for every arms which will introduce phase errors due to different group velocity in each arrays while bending. Furthermore, we can reduce the insertion loss by double etching processing.A weakly guided mode is ideal for mode transition from the free propagation region. And the Deep etched region has an intermediate state for light coupling which will increase the coupling efficiency.

- PECVD Deposited Silica as Cladding

One way to reduce propagation loss is to reduce the intensity at sidewall as mentioned above. While another way is to let mode push out to those materials with less absorption and defects. Silica is an ideal cladding material to reduce the sidewall scattering of the core materials.

- Anneal devices after cleaving

The Hydrogen Bond will be eliminated due to high temperature annealing.[12] The absorption at 1550nm will be reduced if we anneal the patterned silicon nitride device at 1050 Celsius Degree after cleaving.

- Decrease number of arms and increase waveguide separation at transition region

If we would be stick with PHASER, one thing we need to consider is to decrease the number of phase arms. If we suffer sever phase error, the insertion will be influence as well. Even other group tend to have hundreds of arms, we find out more arms may introduce more phase error. Less arms seems to have better performance as in our measurement.

7.3 AWG on other Platforms

7.3.1 AlGaAs on Silicon Dioxide

AlGaAs on silicon dioxide(AlGaAsOI) will have the refractive contrast alike silicon on insulter. We will have compact footprint of AWG based on AlGaAsOI. AlGaAs will have Electro-Optical(EO) effects which can tune phase in the AWG arms. Also, it will have better performance in terms of nonlinear response. Especially for AlGaAs with 18% Al, it will have less two and three photon absorption and free electron absorption. It would be an interesting topic if we integrate AWG with Nonlinear Optics.

7.3.2 Ge Doped Silica for High Power Application

As we mentioned above in WBC, we can have a high power multi-wavelength source if we use Ge doped silica as core material and silica as cladding. Because it will have less refractive index contrast, the dimension of the single mode will be around tens of micron for diameter like fibers. Waveguide with 0.75% contrast normally have 5um(width)*5um(height). It will handle 70 times more power compared with 1.2um(width)*0.3um(height) cross section, if we consider the same damage threshold for both. Moreover, the propagation loss is much less compared with the silicon nitride platform. The high refractive index contrast of waveguides makes it[5, 3] difficult to implement good coupling between free propagation region and arrays, as there is a significant vertical mode mismatch between the photonic waveguide mode and the slab modes in the star coupler. This can result in unwanted reflections, which increases the overall insertion loss of the device. Also, the delay lines implemented in narrow photonic waveguides are prone to phase noise due to small fluctuations in

the waveguide dimension.[1]

Appendices

Appendix A

Code Document for Waveguide Routing with Extra Turn Cost

In [2]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import math
#from collections import deque
from pylab import rcParams
```

In [3]:

```python
class PriorityQueue():
    '''
    The arguments passed to a PriorityQueue must consist of
    objects than can be compared using <.
    Use a tuple (priority, item) if necessary.
    '''

    def __init__(self):
        self._array = []

    def push(self, obj):
        # append at end and bubble up
        self._array.append( obj )
        n = len(self._array)
        self._bubble_up(n-1)

    def pop(self):
        n = len(self._array)
        if n==0:
            return None
        if n==1:
            return self._array.pop()

        # replace with last item and sift down:
        obj = self._array[0]
        self._array[0] = self._array.pop()
        self._sift_down(0)
        return obj

    def _parent(self, n):
        return (n-1)//2

    def _left_child(self, n):
        return 2*n + 1

    def _right_child(self, n):
        return 2*n + 2

    def _bubble_up(self, index):
        while index>0:
            cur_item = self._array[index]
            parent_idx = self._parent(index)
            parent_item = self._array[parent_idx]

            if cur_item < parent_item:
                # swap with parent
                self._array[parent_idx] = cur_item
                self._array[index] = parent_item
                index = parent_idx
            else:
                break

    def _sift_down(self,index):
        n = len(self._array)

        while index<n:
            cur_item = self._array[index]
            lc = self._left_child(index)
            if n <= lc:
```

```
            break

        # first set small child to left child:
        small_child_item = self._array[lc]
        small_child_idx = lc

        # right exists and is smaller?
        rc = self._right_child(index)
        if rc < n:
            r_item = self._array[rc]
            if r_item < small_child_item:
                # right child is smaller than left child:
                small_child_item = r_item
                small_child_idx = rc

        # done: we are smaller than both children:
        if cur_item <= small_child_item:
            break

        # swap with smallest child:
        self._array[index] = small_child_item
        self._array[small_child_idx] = cur_item

        # continue with smallest child:
        index = small_child_idx

def size(self):
    return len(self._array)

def is_empty(self):
    return len(self._array) == 0

def show(self,pidx=0,g=0):
    n=len(self._array)

    from collections import deque
    Q = deque()
    node=(pidx,g)
    Q.append(node)
    g_old=1
    while len(Q)>0:
        pidx,g = Q.popleft()
        children_idx = (self._left_child(pidx),self._right_child(pidx))
        if pidx < n:
            for child_idx in children_idx:
                node=(child_idx,g+1)
                Q.append(node)
            if g!=g_old:
                print ("\n",g,":",self._array[pidx],end=" ") # space: (n//2-
2^g)//2*"  "
            else:
                print(self._array[pidx],end=" ")
            g_old = g
    print("")
   # if idx < n:
   #     print (level*' _ ', self._array[idx])
   #     children_idx = (self._left_child(idx),self._right_child(idx))
   #     for child_idx in children_idx:
   #         self.show(child_idx,level+1)

def heapify(self, items):
```

```
        heap=[]
        for item in items:
            heap.append(item)
        pq._array=heap
        """ Take an array of unsorted items and replace the contents
        of this priority queue by them. """
        n=len(heap)
        for idx in range(n,0,-1):
            self._sift_down(self._parent(idx))

    def decrease_priority(self, old, new):
        # replace old by new and we can assume that new will compare smaller
        # (so priority is higher or the value is smaller)
        assert(new <= old)
        item_idx=0
        for item in self._array:
            if item==old:
                self._array[item_idx]=new
                self._sift_down(self._parent(item_idx))
            item_idx+=1
```

In [4]:

```python
class Graph(object):
    '''Represents a graph'''

    def __init__(self, vertices, edges):
        '''A Graph is defined by its set of vertices
           and its set of edges.'''
        self.V = set(vertices) # The set of vertices
        self.E = set([])       # The set of edges
        self.Adj = {}          # A dictionary that will hold the list
                               # of adjacent vertices for each vertex.
        self.Vcoord = {}       # A dictionary that can hold coordinates
                               # for the vertices.
        self.edge_labels = {}

        self.add_edges(edges)  # Note the call to add_edges will also
                               # update the Adj dictionary
        print ('(Initializing a graph with %d vertices and %d edges)' % (len(sel
f.V),len(self.E)))


    def add_vertices(self,vertex_list):
        ''' This method will add the vertices in the vertex_list
            to the set of vertices for this graph. Since V is a
            set, duplicate vertices will not be added to V. '''
        for v in vertex_list:
            self.V.add(v)
        self.build_Adj()


    def add_edges(self,edge_list):
        ''' This method will add a list of edges to the graph
            It will insure that the vertices of each edge are
            included in the set of vertices (and not duplicated).
            It will also insure that edges are added to the
            list of edges and not duplicated. '''
        for s,t in edge_list:
            if (s,t) not in self.E and (t,s) not in self.E:
                self.V.add(s)
                self.V.add(t)
                self.E.add((s,t))
        self.build_Adj()


    def build_Adj(self):
        self.Adj = {}
        for v in self.V:
            self.Adj[v] = []
        for e in self.E:
            s,t = e
            self.Adj[s].append(t)
            self.Adj[t].append(s)


    def degree_of(self,vertex):
        if vertex in self.V:
            return len(self.Adj[vertex])
        else:
            return None
```

```python
    def get_a_vertex(self):
        if 0 < len(self.V):
            v = self.V.pop()
            self.V.add(v)
            return v
        else:
            return None


    def plot(self):
        px = []
        py = []
        for v in self.V:
            px.append(self.Vcoord[v][0])
            py.append(self.Vcoord[v][1])
        plt.plot(px,py,'bo',hold=True)
        for vertex in self.V:
            p = self.Vcoord[vertex]
            pq = max(0.1,math.sqrt(p[0]**2 + p[1]**2))
            rx = p[0]/pq
            ry = p[1]/pq
            plt.text(p[0]+0.2*rx, p[1]+0.2*ry, str(vertex))

        for s,t in self.E:
            plt.plot([self.Vcoord[s][0], self.Vcoord[t][0]],
                     [self.Vcoord[s][1], self.Vcoord[t][1]],
                     'b',hold=True)
```

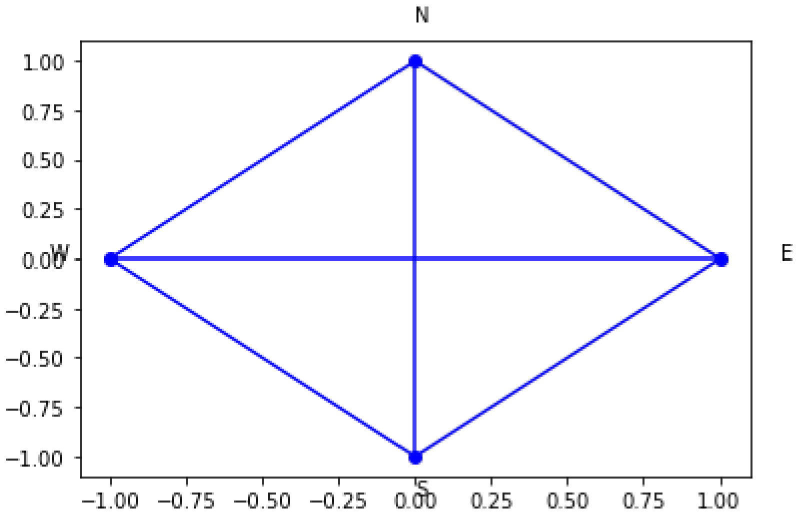
In [5]:

```python
class Network(Graph):
    def __init__(self, vertices, edge_weights):
        ''' Initialize the network with a list of vertices
        and weights (a dictionary with keys (E1, E2) and values are the weight
s)'''

        edges = []
        for e1,e2 in edge_weights:
            edges.append((e1,e2))

        Graph.__init__(self, vertices, edges)
        self.weights = {}
        for e1,e2 in edge_weights:
            weight = edge_weights[(e1,e2)]
            self.weights[(e1,e2)] = weight
            self.weights[(e2,e1)] = weight
        self.edge_labels = self.weights
```

In [8]:

```
V = ['N', 'S', 'W', 'E']
W = {('N','S'):1, ('N','W'):10, ('N','E'):10, ('S','E'):10, ('S','W'):10,('W',
'E'):1}
G1 = Network(V,W)
G1.Vcoord = {'N':(0,1),'S':(0,-1),'W':(-1,0),'E':(1,0)}
G1.plot()
print (G1.weights)
```

```
(Initializing a graph with 4 vertices and 6 edges)
{('N', 'S'): 1, ('S', 'N'): 1, ('N', 'W'): 10, ('W', 'N'): 10, ('N',
'E'): 10, ('E', 'N'): 10, ('S', 'E'): 10, ('E', 'S'): 10, ('S',
'W'): 10, ('W', 'S'): 10, ('W', 'E'): 1, ('E', 'W'): 1}

/usr/local/share/jupyterhub/env/jupyterhub/lib/python3.6/site-packag
es/matplotlib/pyplot.py:3316: MatplotlibDeprecationWarning: The 'hol
d' keyword argument is deprecated since 2.0.
  mplDeprecation)
```



In [6]:

```
class PhotonicRoute():
    """Arguments:(width, area_x, area_y, boundary, starts, ends)
    Width: set up the grids with width for the whole graph
    area_x, area_y: Lower left is (0,0)
    boundary: Device's boundary dimension to avoid bypass and overlaps
    starts, ends: the routing points with initial and ending direction"""
    def __init__(self, width, area_x, area_y, boundary, starts, ends):
        self.st=1
        self.bnd=10
        self.width=width
        self.area_x=area_x
        self.area_y=area_y
        self.boundary=boundary
        self.starts=starts
        self.ends=ends


        """bnd for bent cost and st for stright cost"""

        """Translate boundaries or blocks that would not allow the path to grid
 node"""
    def NotAllowed(self):
        width=self.width
        forbidden=[]
        for boundary_x, boundary_y in self.boundary:
            for i in range (boundary_x[0]//width,boundary_x[1]//width,1):
                for j in range (boundary_y[0]//width,boundary_y[1]//width,1):
                    for A in ["N","S","W","E"]:
                        forbidden.append(str(i)+','+str(j)+A)
        return forbidden

    def Diamond(self):
        """Assiagn the grid with diamond design
                N
                |
                |
        W---grid(x1,y1)---E
                |
                |
                S
        st:W-E,N-S
        bnd: W-N,W-S,E-N,E,S"""
        width=self.width
        m_pts = self.area_x//width
        n_pts = self.area_y//width
        st=self.st
        bnd=self.bnd
        vertex_list=[]
        vcoords={}
        edge_list = []
        weights = {}
        for i in range(m_pts):
            for j in range(n_pts):
                N=str(i)+','+str(j)+"N"
                vertex_list.append(N)
                vcoords[N]=(width*i,width*j+1)
                S=str(i)+','+str(j)+"S"
                vertex_list.append(S)
                vcoords[S]=(width*i,width*j-1)
```

```
                W=str(i)+','+str(j)+"W"
                vertex_list.append(W)
                vcoords[W]=(width*i-1,width*j)
                E=str(i)+','+str(j)+"E"
                vertex_list.append(E)
                vcoords[E]=(width*i+1,width*j)

                weights[(N,S)]=st
                weights[(W,E)]=st
                weights[(N,W)]=bnd
                weights[(N,E)]=bnd
                weights[(S,W)]=bnd
                weights[(S,E)]=bnd
                if i+1 < m_pts or j+1< n_pts:
                    if i+1 == m_pts and j+1 == n_pts:
                        pass
                    elif i+1 == m_pts:
                        weights[N,str(i)+','+str(j+1)+"S"]=st
                    elif j+1 ==n_pts:
                        weights[E,str(i+1)+','+str(j)+"W"]=st
                    else:
                        weights[N,str(i)+','+str(j+1)+"S"]=st
                        weights[E,str(i+1)+','+str(j)+"W"]=st


        G = Network(vertex_list,weights)
        G.Vcoord=vcoords
        return G


    def Routing(self, network):
        """Utilize Dijkstra to find the shortes path for given starts and ends i
n the class """
        forbidden=self.NotAllowed()
        starts=self.starts
        ends=self.ends
        s=len(starts)
        e=len(ends)
        paths={}
        total_cost={}

        if s != e:
            raise ValueError("Please put the same number of points for start and
 end!")

        taken=[]
        cross=set()

        for i in range(e):
            cost = {starts[i]:0}
            prev = {}
            done = {}
            pq = PriorityQueue()
            pq.push((0,starts[i]))
            route = [ends[i]]


            while not pq.is_empty():

                cost_u, u = pq.pop()
```

```
                if u in done:
                    continue
                done[u] = True

                for v in network.Adj[u]:

                    if v in forbidden:
                        """Make sure that boundary is not connected and crosse
d"""
                        continue

                    if v in taken:
                        """Not only the node it has been taken before but the 4
 points in the grid"""
                        continue

                    new_cost_to_v = cost_u + network.weights[(u,v)]

                    if not v in cost or cost[v] > new_cost_to_v:
                        cost[v] = new_cost_to_v
                        prev[v] = u
                        pq.push((new_cost_to_v, v))

                    if v == ends[i]:
                        while route[-1] != starts[i]:
                            route.append(prev[route[-1]])
                        break
                    total_cost[starts[i]] = cost[v]

            for pts in route:
                for A in ["N","S","W","E"]:
                    taken.append(pts[:-1]+A)
                    """Mark the pints which the route goes through with directio
n nodes to avoid wire overlaps"""

            paths[starts[i]]=route

        return paths, total_cost

    def PlotPath(self,G,paths):
        """Plot the diamond grids and hold on the pictures for show the boundary
 points and paths"""
        plt.figure()
        G.plot()
        starts=self.starts
        ends=self.ends
        px = []
        py = []
        forbidden=self.NotAllowed()
        for A in forbidden:
            px.append(G.Vcoord[A][0])
            py.append(G.Vcoord[A][1])
            plt.plot(px,py,'yo',hold=True)

        for start in starts:
            px = []
            py = []
            for A in paths[start]:
                px.append(G.Vcoord[A][0])
                py.append(G.Vcoord[A][1])
                plt.plot(px,py,'r',hold=True)
```

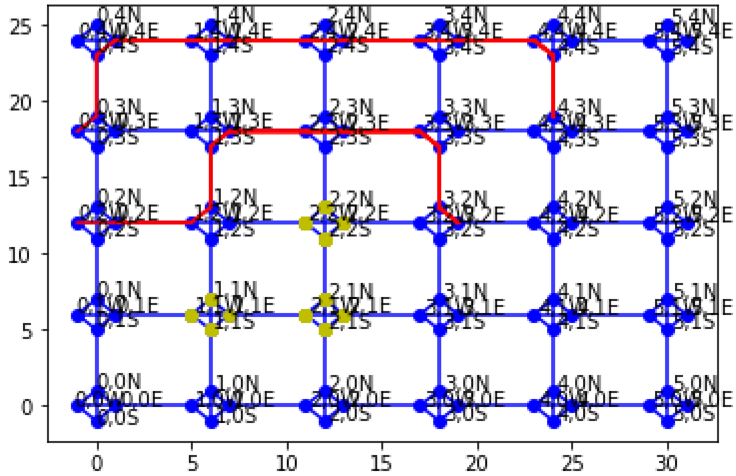
In [7]:

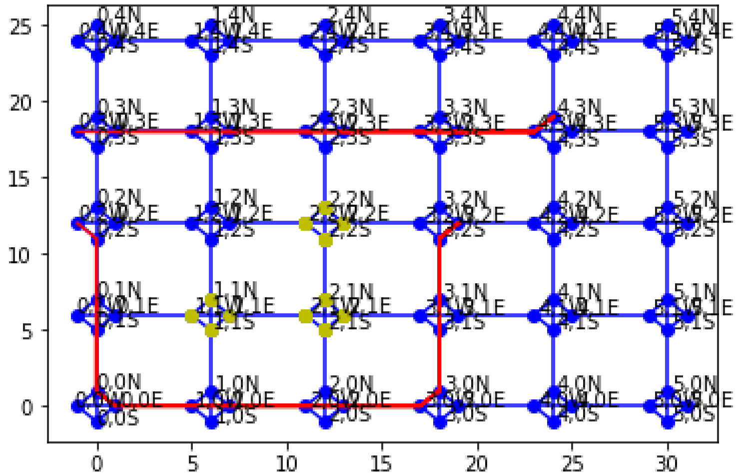
```
class Tryitout():
    width=6
    area_x=36
    area_y=30
    boundary=[([12,18],[6,18]),([6,12],[6,12])]
    #boundary_x=[12,18]
    #boundary_y=[6,18]
    #boundary=[([12,18],[6,18]),([6,12],[6,12])]
    starts=['0,2W','0,3W']
    ends=['3,2E','4,3N']
    #st=1
    #bnd=10
    P1=PhotonicRoute(width, area_x, area_y, boundary, starts, ends)
    G1=P1.Diamond()
    paths1,cost1=P1.Routing(G1)
    P1.PlotPath(G1,paths1)
    #starts=starts[::-1]
    #ends=ends[::-1]
    P2=PhotonicRoute(width, area_x, area_y, boundary, starts[::-1], ends[::-1])
    #P2=PhotonicRoute(width, area_x, area_y, boundary, starts, ends)
    G2=P2.Diamond()
    paths2,cost2=P2.Routing(G2)
    P2.PlotPath(G2,paths2)
    print(cost1)
    print(cost2)
```

```
(Initializing a graph with 120 vertices and 229 edges)
```

```
/usr/local/share/jupyterhub/env/jupyterhub/lib/python3.6/site-packag
es/matplotlib/pyplot.py:3316: MatplotlibDeprecationWarning: The 'hol
d' keyword argument is deprecated since 2.0.
  mplDeprecation)
```

```
(Initializing a graph with 120 vertices and 229 edges)
{'0,2W': 43, '0,3W': 83}
{'0,3W': 41, '0,2W': 78}
```





In [ ]:

```
v=['0,2N','0,1S']
print(v[1][:-1])
```

Bibliography

[1] Wim Bogaerts, Shankar Kumar Selvaraja, Pieter Dumon, Joost Brouckaert, Katrien De Vos, Dries Van Thourhout, and Roel Baets. Silicon-on-insulator spectral filters fabricated with cmos technology. *IEEE journal of selected topics in quantum electronics*, 16(1):33–44, 2010.

[2] By Yoshinori Hibino. An array of photonic filtering advantages: arrayed-waveguide-grating multi/demultiplexers for photonic networks. *IEEE Circuits and Devices Magazine*, 16(6):21–27, Nov 2000.

[3] Yohan Barbarin, X.j. M. Leijtens, E.a.j. M. Bente, C M. Louzao, J R. Kooiman, and M K. Smit. Extremely small awg demultiplexer fabricated on inp by using a double-etch process. In *Optical Amplifiers and Their Applications/Integrated Photonics Research*, page IThG4. Optical Society of America, 2004.

[4] K. Rausch. *Broadband arrayed waveguide grating multiplexers on indium phosphide*. PhD thesis, The University of Arizona, 2005.

[5] R. Baets, P. Dumon, W. Bogaerts, D. Van Thourhout, D. Taillaert, B. Luyssaert, P. Bienstman, V. Wiaux, S. Beckx, and J. Wouters. Silicon-on-insulator platform for wdm-components. 2:741–742 Vol.2, Nov 2004.

[6] Daan Martens, Ananth Z. Subramanian, Shibnath Pathak, Michael Vanslembrouck, Peter Bienstman, Wim Bogaerts, and Roel G. Baets. Compact silicon nitride arrayed waveguide gratings for very near-infrared wavelengths. *IEEE Photonics Technology Letters*, 27(2):137–140, 2015.

[7] Nur Ismail, Fei Sun, Gabriel Sengo, Kerstin Wörhoff, Alfred Driessen, René M De Ridder, and Markus Pollnau. Improved arrayed-waveguide-grating layout avoiding systematic phase errors. *Optics express*, 19(9):8781–8794, 2011.

[8] R. Germann, H. W. M. Salemink, R. Beyeler, G. L. Bona, F. Horst, I. Massarek, and B. J. Offrein. Silicon Oxynitride Layers for Optical Waveguide Applications. *Journal of The Electrochemical Society*, 147(6):2237, jun 2000.

[9] Antonio Sanchez-Rubio, Tso Yee Fan, Steven J Augst, Anish K Goyal, Kevin J Creedon, Juliet T Gopinath, Vincenzo Daneu, Bien Chann, and Robin Huang.

Wavelength Beam Combining for Power and Brightness Scaling of Laser Systems. *Lincoln Laboratory Journal*, 20(2):15, 2014.

[10] M.K. Smit. New focusing and dispersive planar component based on an optical phased array. *Electronics Letters*, 24(7):385, 1988.

[11] M. K. Smit. Progress in awg design and technology. pages 26–31, June 2005.

[12] Gregory Allen Vawter Scott Habermehl Charles T. Sullivan Michael J. Shaw, Junpeng Guo. Fabrication techniques for low-loss silicon nitride waveguides. *Proc.SPIE*, 5720:109, jan 2005.

[13] Yoshinori Hibino. Silica-based planar lightwave circuits and their applications. *MRS bulletin*, 28(05):365–371, 2003.

[14] Masao Kawachi, Shoichi Sudo, Noriyoshi Shibata, and Takao Edahiro. Deposition properties of SiO2-GeO2 particles in the flame hydrolysis reaction for optical fiber fabrication. *Japanese Journal of Applied Physics*, 19(2):L69–L71, 1980.

[15] T Kominato, Y Ohmori, H Okazaki, and M Yasu. Very low-loss GeO/sub 2/-doped silica waveguides fabricated by flame hydrolysis deposition method. *Electronics Letters*, 26(5):327–329, 1990.

[16] Takashi Goh, Senichi Suzuki, and Akio Sugita. Estimation of waveguide phase error in silica-based waveguides. *Journal of Lightwave Technology*, 15(11):2107–2113, 1997.

[17] Yoshinori Hibino. Recent advances in high-density and large-scale awg multi/demultiplexers with higher index-contrast silica-based plcs. *IEEE Journal of selected topics in quantum electronics*, 8(6):1090–1101, 2002.

[18] K Takada, M Abe, T Shibata, and K Okamoto. 10-ghz-spaced 1010-channel tandem awg filter consisting of one primary and ten secondary awgs. *IEEE Photonics Technology Letters*, 13(6):577–578, 2001.

[19] Masaki Kohtoku. Low-Loss Compact Silica-Based AWG Using Deep Ridge Waveguide. In *Integrated Photonics Research and Applications/Nanophotonics for Information Systems*, page ITuF1, Washington, D.C., apr 2005. OSA.

[20] Multimode-interference Couplers, H N J Fernando, M Hayden, and P J Hughes. Optimal Design , Fabrication and Characterisation of GeO 2 -SiO 2 doped Silica. *Transition*, 1(0):0–5, 2006.

[21] M Abe. Silica-based waveguide devices for photonic networks. *Journal of the Ceramic Society of Japan*, 116(1358):1063–1070, 2008.

[22] Franco Bruno, Massimo del Guidice, Roberto Recca, and Francesco Testa. Plasma-enhanced chemical vapor deposition of low-loss SiON optical waveguides at 15-?m wavelength. *Applied Optics*, 30(31):4560, nov 1991.

[23] F Ay and A Aydinli. Comparative investigation of hydrogen bonding in silicon based pecvd grown dielectrics for optical waveguides. *Optical Materials*, 26(1):33 – 46, 2004.

[24] U. Fischer, T. Zinke, J.-R. Kropp, F. Arndt, and K. Petermann. 0.1 dB/cm waveguide losses in single-mode SOI rib waveguides. *IEEE Photonics Technology Letters*, 8(5):647–648, may 1996.

[25] Stanley Cheung, Tiehui Su, Katsunari Okamoto, and S J B Yoo. Arrayed Waveguide Grating Router. 20(4), 2014.

[26] Jared F. Bauters, Martijn J. R. Heck, Demis John, Daoxin Dai, Ming-Chun Tien, Jonathon S. Barton, Arne Leinse, René G. Heideman, Daniel J. Blumenthal, and John E. Bowers. Ultra-low-loss high-aspect-ratio Si_3N_4 waveguides. *Optics Express*, 19(4):3163, feb 2011.

[27] C. H. Henry, R. F. Kazarinov, H. J. Lee, K. J. Orlowsky, and L. E. Katz. Low loss Si_3N_4SiO_2 optical waveguides on Si. *Applied Optics*, 26(13):2621, jul 1987.

[28] Michael Belt, Michael L. Davenport, John E. Bowers, and Daniel J. Blumenthal. Ultra-low-loss Ta_2O_5-core/SiO_2-clad planar waveguides on Si substrates. *Optica*, 4(5):532, may 2017.

[29] A. Z. Subramanian, P. Neutens, A. Dhakal, R. Jansen, T. Claes, X. Rottenberg, F. Peyskens, S. Selvaraja, P. Helin, B. DuBois, K. Leyssens, S. Severi, P. Deshpande, R. Baets, and P. Van Dorpe. Low-Loss Singlemode PECVD Silicon Nitride Photonic Wire Waveguides for 532900 nm Wavelength Window Fabricated Within a CMOS Pilot Line. *IEEE Photonics Journal*, 5(6):2202809–2202809, dec 2013.

[30] Alexander Gondarenko, Jacob S. Levy, and Michal Lipson. High confinement micron-scale silicon nitride high Q ring resonator. *Optics Express*, 17(14):11366, jul 2009.

[31] Ming-Chun Tien, Jared F. Bauters, Martijn J. R. Heck, Daryl T. Spencer, Daniel J. Blumenthal, and John E. Bowers. Ultra-high quality factor planar Si_3N_4 ring resonators on Si substrates. *Optics Express*, 19(14):13551, jul 2011.

[32] Daryl T. Spencer, Yongbo Tang, Jared F. Bauters, Martijn J. R. Heck, and John E. Bowers. Integrated Si3N4/SiO2 ultra high Q ring resonators. In *IEEE Photonics Conference 2012*, pages 141–142. IEEE, sep 2012.

[33] Shibnath Pathak. *Silicon nano-photonics based arrayed waveguide gratings*. 2014.

[34] Long Chen, Christopher R. Doerr, Larry Buhl, Yves Baeyens, and Ricardo A. Aroca. Monolithically Integrated 40-Wavelength Demultiplexer and Photodetector Array on Silicon. *IEEE Photonics Technology Letters*, 23(13):869–871, jul 2011.

[35] C Kopp, Stéphane Bernabé, B B Bakir, J Fedeli, R Orobtchouk, F Schrank, H Porte, L Zimmermann, and T Tekin. Silicon Photonic Circuits: On-CMOS Integration, Fiber Optical Coupling, and Packaging. *IEEE Journal of Selected Topics in Quantum Electronics*, 17(3):498–509, may 2011.

[36] W. Bogaerts, P. Dumon, E. Lambert, M. Fiers, S. Pathak, and A. Ribeiro. Ipkiss: A parametric design and simulation framework for silicon photonics. pages 30–32, Aug 2012.