Silicon Photonics Design and Fabrication of a Fabry Perot Cavity using Bragg Resonators and Waveguides to Achieve Highest Resonance Quality Factor

Jasmine Natasha Radu
Department of Electrical and Computer Engineering
University of British Columbia

***Abstract*—**

***Index Terms*—Fabry Perot, Quality Factor, Q-Factor, Bragg Grating, Waveguide, Silicon Photonics, Effective Index, Group Index, EBeam Lithography.**

I. INTRODUCTION

WILL WRITE LATER

II. MODELLING

A waveguide consists of a core and a cladding, the core is made of a material with a high index of refraction and the cladding of a low index of refraction. The light is then guided down the core by Total Internal Refraction (TIR). The core for this design is made up of silicon and the cladding is made up of (what will it be??). TE polarization will be the mode of the waveguide, TE mode is dependent on the transverse electric waves, relying on the electric field vector being perpendicular to the direction of propagation. Comparatively, TM waves are dependent on the magnetic field perpendicular to the waveguide propagation. The TE waveguides were designed with a Silicon strip with slab thickness of 220 nm thickness, making them single mode devices. The 220 nm thickness is a set parameter based on the chip that will be used.

To decide on a waveguide width at 220 nm thickness can be done by comparing the effective index. The group index is especially important to consider in photonics, more specifically the group velocity. However, increasing the width of the waveguide will increase the effective index, but decrease the group index. This is the tradeoff between the designs that were considered.

Additionally, the modeling was done using Lumerical MODE to design the waveguide and its properties. This software allows you to simulate and predict the behavior using different sizes, index of reflection of materials, and pathlength, to name a few. These design choices will alter the results of group index and free spectral range (FSR).

To design the device, fabrication bias needs to be considered. These values can be estimated based on similar processes done by Applied Nanotools Inc. Most importantly they have a “shrink” factor of 15 nm. To counter this factor the design choice of using 335 nm for simulations was used, in order to theoretically produce a fabricated device with a width of 350 nm.

mode #	effective index	wavelength (µm)	loss (dB/cm)	group index
1	2.376783+1.676153e-09i	1.31	0.00069829	4.527715+4.528549e-09i
2	1.991554+1.462919e-09i	1.31	0.00060946	4.561342+6.774526e-09i
3	1.451805+3.656075e-10i	1.31	0.00015231	1.964811+1.342595e-09i
4	1.379350+2.867623e-10i	1.31	0.00011947	1.881339+2.692620e-09i

Fig. 1 - Table of calculated effective and group index, as well as losses at 1310nm wavelength, using 335 width.

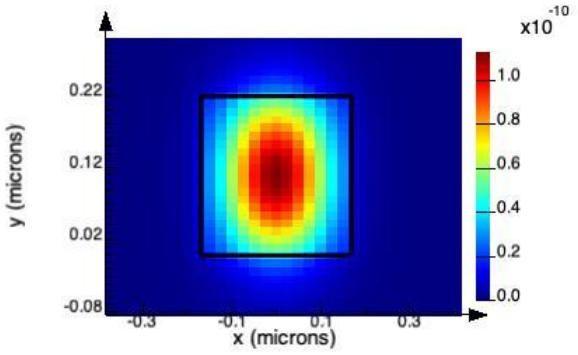

Fig. 2 - Energy density of a Si waveguide with 335 nm width and 220 nm height.

In Figure 2 you can see that most of the light is traveling inside the waveguide, which is desired.

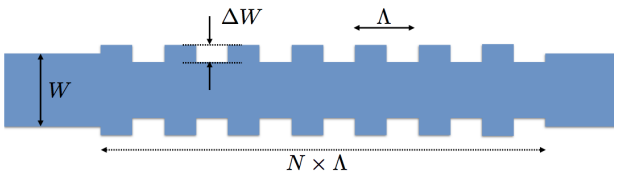

Fig. 3 - Design parameters in the figure were determined using Lumerical MODE, FDTD and INTERCONNECT [1].

STILL WORKING THESE OUT

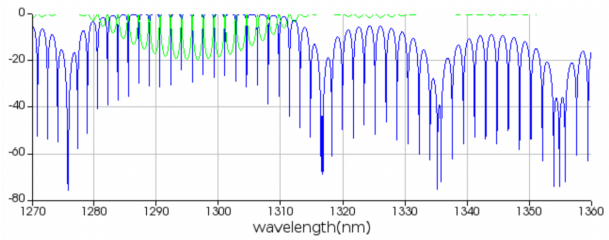

Fig. 4 - Response using interconnect, not centered at 1310nm because of bias.

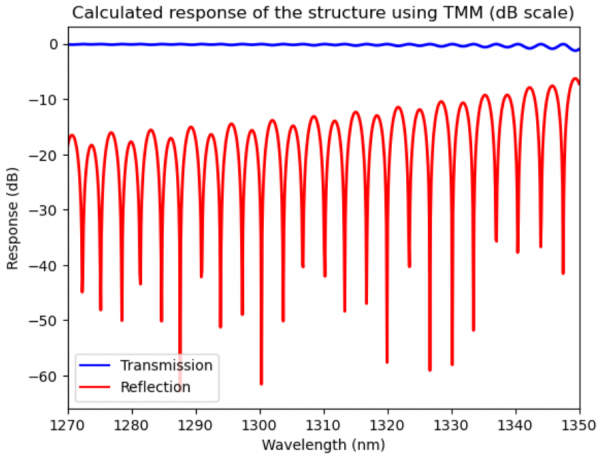

Fig. 5 - Python script that I am struggling to get right, i think it has to do with my kappa value…

NEED TO ADD MORE VALUES AND TABLE OF SIMULATION RESULTS AND PARAMETERS

III. DESIGN

The layout was made using black box grating couplers, which is common in fabrication designs where they are hesitant to share their designs with a research team. The splitters used was a subwavelength grating splitter that was already designed to accommodate the bias of ANT's fabrication. The waveguides are 350 nm to accommodate for the 335 nm biasing that was used in simulations.

The layout of the design can be shown in Figure 6 has 4 devices, each with the same cavity lengths of 100 um but varying number of periods on the bragg gratings. These values were 10, 50, 100, 120.

NEED A HIGHER Q, THESE VALUES NOT SET IN STONE

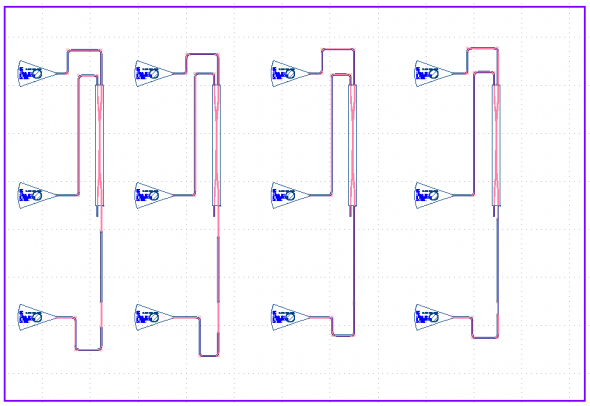

Fig. 6 - Layout of 4 Fabry-Perot resonators using KLayout.

The design was then merged with the other students of various classes to form one chip that will be sent to ANT for fabrication. This current layout is shown in Figure 7 below.

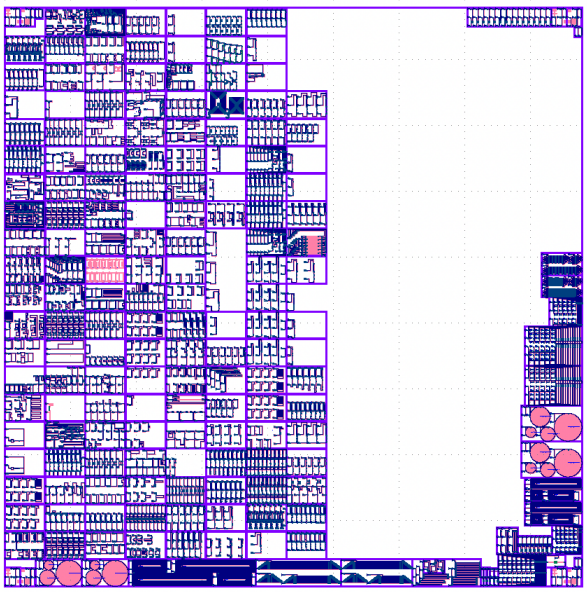

Fig. 7 - Merged designs of all students' final designs shown on KLayout.

IV. FABRICATION

The fabrication was done by Applied Nanotools Inc. (ANT) located in Alberta, Canada.

Fig. 8 - Cross section of SOI Wafer (Prefabrication) with a Si layer with a thickness of approx. 220 nm, Buried Silicon Dioxide layer (BOX) and a Si Substrate layer.

V. EXPERIMENTS

NEED MEASUREMENTS AFTER FABRICATION

VI. ANALYSIS

NEED MEASUREMENTS AFTER FABRICATION

VII. DISCUSSION

NEED MEASUREMENTS AFTER FABRICATION

VIII. CONCLUSION

WILL WRITE LATER

IX. ACKNOWLEDGEMENT

I acknowledge the ELEC 413 professor, Lucas Chrostowski and his teaching assistant, Kithmin Randula. I acknowledge the edX UBCx Silicon Photonics Design, Fabrication and Data Analysis course, which is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Silicon Electronic-Photonic Integrated Circuits (SiEPIC) Program. I acknowledge Applied Nanotools (ANT), where the fabrication took place. I acknowledge Lumerical Solutions, Inc., Mathworks, Mentor Graphics, Python, and KLayout for the design software.

X. REFERENCES

[1] Photo taken from Edx page created by Lucas Chrostowski.

[2]

XI. APPENDIX

Matlab Code used to find python values for n1,n2,n3, from imported MODE values, provided by Lucas Chrostowski.

```
% Phot1x_fit_wg_compactmodel.m
% by Lukas Chrostowski, 2015
% User provides a matrix of neff
values vs. wavelength
% Matlab curve fits to an expression.
%
url='https://www.dropbox.com/s/xv4he4p
reyfa9v2/wg-export-TM.mat?dl=1'
%load('freq1.mat');
neff = real(neff);
% take the real part of the effective
index.
c=299792458;
% speed of light, m/s
lambdas = c ./ f;% f is the matrix of
frequency points,
% where the effective index is
recorded.
lambdas = lambdas * 1e6; % convert to
microns.
lambda0 = 1.31; % replace with desired
centre wavelength
figure; plot (lambdas,
neff,'o','MarkerSize',10); hold on;
% use Matlab anonymous function for
the effective index expression:
neff_eq = @(nx, lambda) (nx(1) +
nx(2).*(lambda-lambda0) +
nx(3).*(lambda-lambda0).^2);
% initial guess.
X=[2.4 0 0];
plot ( lambdas, neff_eq(X, lambdas),
'r')
% curve fit to find expression for
neff.
format long
X = lsqcurvefit (neff_eq, X, lambdas,
neff);
r=corrcoef(neff,neff_eq(X, lambdas));
r2=r(1,2).^2;
disp (['Goodness of fit, r^2 value: '
num2str(r2) ])
lambdas2=linspace(min(lambdas),
max(lambdas), 100);
plot ( lambdas2, neff_eq(X, lambdas2),
'k')
xlabel ('Wavelength [nm]');
ylabel ('Effective Index');
legend ('Data','Initial Guess','Curve
Fit')
```

Current Python Code (still need kappa):

```
"""
SiEPIC Photonics Package
```

```
Author:     Mustafa Hammood
           Mustafa@siepic.com

https://github.com/SiEPIC-Kits/SiEPIC_Photonics_Package
fixed by Lukas Chrostowski, 2020/02

Solvers and simulators: Bragg simulator using transfer matrix method (TMM) approach
"""

#%% dependent packages
import numpy as np
import math, cmath, matplotlib
import matplotlib.pyplot as plt
from numpy.lib.scimath import sqrt as csqrt
import scipy.io as sio

#%% user input

# set the wavelength span for the simultion
wavelength_start = 1270e-9
wavelength_stop = 1350e-9
resolution = 0.1

# Grating waveguide compact model (cavity)
# these are polynomial fit constants from a waveguide width of 350 nm
n1_wg = 2.3768
n2_wg = -1.6424
n3_wg = -0.0512

# Cavity waveguide compact model (cavity)
# these are polynomial fit constants from a waveguide width of 350 nm
n1_c = 2.433864
n2_c = -0.7829723
n3_c = -0.072348
```

```
# grating parameters
period = 270e-9     # period of pertrubation
n_delta = .0      # effective index pertrubation
lambda_Bragg = 1310e-9
dw = 50e-9
kappa = -1.53519e19 * dw**2+ 2.2751e12 * dw
n_delta = kappa * lambda_Bragg / 2
print(n_delta)
N_left = 200        # number of periods (left of cavity)
N_right = 200       # number of periods (right of cavity)

# Cavity Parameters
alpha_dBcm = 3     # dB per cm
alpha = np.log(10)*alpha_dBcm/10*100. # per meter

L = period/2    # length of cavity

#%% Analysis

def HomoWG_Matrix( wavelength, neff, l):
   beta = 2*math.pi*neff/wavelength-j*alpha/2
   v = [np.exp(j*beta*l), np.exp(-j*beta*l)]
   T_hw = np.diag(v)

   return T_hw

def IndexStep_Matrix(neff1, neff2):

a=(neff1+neff2)/(2*np.sqrt(neff1*neff2))

b=(neff1-neff2)/(2*np.sqrt(neff1*neff2))

   T_is=[[a, b],[b, a]]
   return T_is
```

```
def Grating_Matrix( wavelength, n1, n2, l
):
   T_hw1=HomoWG_Matrix(wavelength, n1, l)
   T_is12=IndexStep_Matrix(n1,n2)
   T_hw2=HomoWG_Matrix(wavelength, n2, l)
   T_is21=IndexStep_Matrix(n2,n1)


   type = 'Cavity'


   if type == 'Waveguide':
       # 1 cm
       T = HomoWG_Matrix(wavelength, n1,
0.01)

   if type == 'Bragg_left':
       Tp1 = np.matmul(T_hw1, T_is12)
       Tp2 = np.matmul(T_hw2, T_is21)
       Tp_Left = np.matmul(Tp1, Tp2)
       T =
np.linalg.matrix_power(Tp_Left,N_left)

   if type == 'Bragg_right':
       Tp1 = np.matmul(T_hw1, T_is12)
       Tp2 = np.matmul(T_hw2, T_is21)
       Tp_Right = np.matmul(Tp1, Tp2)

       T =
np.linalg.matrix_power(Tp_Right,N_right)

   if type == 'Cavity':
       Tp1 = np.matmul(T_hw1, T_is12)
       Tp2 = np.matmul(T_hw2, T_is21)
       Tp_Left = np.matmul(Tp1, Tp2)

       T_cavity =
HomoWG_Matrix(wavelength, n1, l)

       Tp1 = np.matmul(T_hw1, T_is12)
       Tp2 = np.matmul(T_hw2, T_is21)
       Tp_Right = np.matmul(Tp1, Tp2)

       T = np.matmul(np.matmul(
np.linalg.matrix_power(Tp_Left,N_left),
T_cavity),
np.linalg.matrix_power(Tp_Right,N_right))

   return T

def Grating_RT( wavelength, n1, n2, l ):
   M = Grating_Matrix( wavelength, n1, n2,
l )
   T = np.absolute(1 / M[0][0])**2
   R = np.absolute(M[1][0]/M[0][0])**2. #
or M[0][1]?
   return [T,R]

j = cmath.sqrt(-1)

l = period/2
lambda_0 = np.linspace(wavelength_start,
wavelength_stop,
round((wavelength_stop-wavelength_start)*1
e9/resolution))
neff0 = (n1_wg + n2_wg*(lambda_0*1e6) +
n3_wg*(lambda_0*1e6)**2)

n1 = neff0 - n_delta/2
n2 = neff0 + n_delta/2

# Grating length
L_left = N_left * period
L_right = N_right * period

R = []
T = []
for i in range(len(lambda_0)):
   [t, r] = Grating_RT(lambda_0[i], n1[i],
n2[i], l)

   R.append(r)
   T.append(t)
```

```
sio.savemat('bragg_tmm.mat', {'R':R,
'T':T, 'lambda_0': lambda_0})

#sio.loadmat('bragg_interconnect.mat')

#%% plot spectrum
plt.figure(0)
fig1 =
plt.plot(lambda_0*1e9,10*np.log10(T),
label='Transmission', color='blue')
fig2 =
plt.plot(lambda_0*1e9,10*np.log10(R),
label='Reflection', color='red')
plt.legend(loc=0)
plt.ylabel('Response (dB)', color =
'black')
plt.xlabel('Wavelength (nm)', color =
'black')
plt.setp(fig1, 'linewidth', 2.0)
plt.setp(fig2, 'linewidth', 2.0)
plt.xlim(round(min(lambda_0*1e9)),round(ma
x(lambda_0*1e9)))
plt.title("Calculated response of the
structure using TMM (dB scale)")
plt.savefig('bragg_tmm_dB'+'.pdf')

# %%
```