ELEC413 Project – Fabry Perot with Bragg Grating Design

Or Bahari #51277200
University of British Columbia, Canada
orbahari@mail.tau.ac.il

Abstract

Fabry-Perot cavity is a common cavity designed for creating a laser or as a highly selective filter. It contains a waveguide with mirrors at both ends. In this project we will create a Fabry-Perot (FB) cavity using a Bragg Grating technology. Our **Objective** is Designing such Fabry Perot cavity made from Bragg Gratings, as well as Achieving a high Transmission Peak and a high Quality Factor as a secondary objective. Layout designs were fabricated and measured and compared to a simulation model, retrieving some devices with sufficient parameters.

Theory

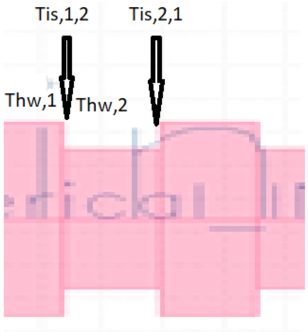


Fig. 1

Bragg Grating Demonstration

Brag Grating is a structure consisting of a waveguide with a periodic variation in effective index that acts as a high reflective mirror for certain wavelengths. A long enough Bragg Gratings structure can be used as a mirror. **Transfer Method Matrix (TMM)** is a model used to calculate the reflection/transmission rates of our Bragg Gratings structure. **Fabry Perot Cavity** is an optical structure consisting two series of properly spaced mirrors which create a resonator when light is shined on them (light is shined from one side). Different corrugation widths have different refractive indices, n1 and n2. $T_{hw,1}$ and $T_{hw,2}$ are the transfer matrices for the homogenous waveguide (diagonal eigenvalues matrices), $T_{\text{is}1,2}$ and $T_{\text{is}2,1}$ are the step Index matrices from n1 to n2 and vice versa respectively. These matrices illustrated in Fig. 1. To create a Fabry-Perot cavity in my design I used 2 front-to-back bragg grating structures, with the same Δn and the same period, but different number of gratings. In our Bragg Grating design, the transfer function was calculated as fallows: $T = (T_{hw,1}T_{\text{is}1,2}T_{hw,2}T_{\text{is}2,1})^{NG1}T_{hw,1}(T_{hw,1}T_{\text{is}1,2}T_{hw,2}T_{\text{is}2,1})^{NG2} = [\text{2x2 matrix}] = \begin{bmatrix} T11 & T12 \\ T21 & T22 \end{bmatrix}$

Where NG1 stands for the Brag-Grating in front of our coupler and NG2 is the Brag-Grating attached to NG1 (back to back). We derive the Reflection and the Transmission values from:

$T = \left(\frac{1}{T1}\right)^2, R = \left(\frac{T21}{T1}\right)^2$

We will extract Reflection value of the peak in the resonance frequency from the Reflection Vs. Wavelength graph and plug it in the Fabry-Perot power transmission ratio:

$$\frac{I_t}{I_i} = \frac{(1-R)^2}{(1-R)^2 + 4R\sin^2(\frac{\delta}{2})}, \delta = \frac{-4\pi\, n\, l}{\lambda}, \text{ 'l' is the cavity length}$$

The Quality Factor Q increases with number of gratings NG – see reference [1] , though the Transmission Peak (T) decreases. Our goal in this design is to achieve a good trade-off between the two, a high Transmission Peak and a high-Quality Factor. Computing Q factor: $Q = \frac{w(T_{peak})}{bandwidth(w_{\frac{1}{2}})}$ and the insertion loss: $Insertion_{Loss} = 10\log(1-R)\ [dB]$ R is the peak in the resonant frequency. See reference [2].

Modelling, Design and Simulations

In our model, a silicon substrate of 500nm length and 220nm width surrounded by SiO2 (Box) was chosen. The waveguide's profile (main modes) as simulated with **Lumerical Mode** is shown in Fig. 2A-2B. The Black rectangle edges represent the waveguide edges.

TE mode: **TM mode**:

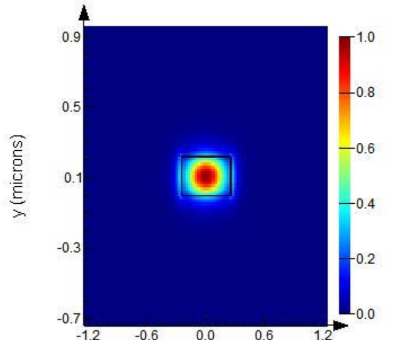


Fig. 2A

Electric Field Intensity of Waveguide in TE mod

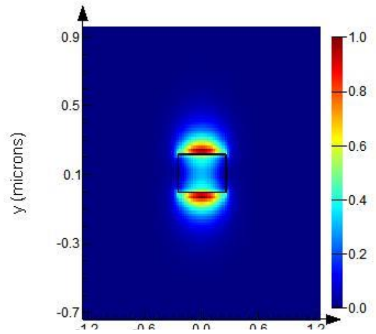


Fig. 2B

Electric Field Intensity of Waveguide in TM mo

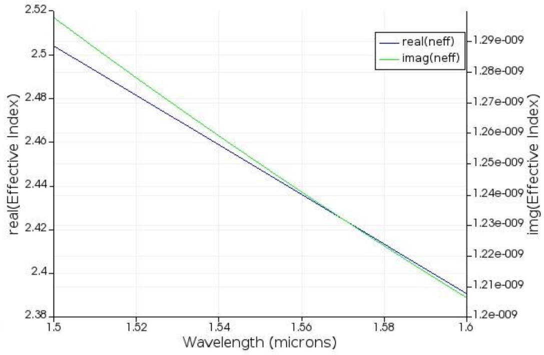


Fig. 3

Waveguide's effective index Vs. wavelength for TE1 mode

Each of these TE and TM modes of the model provide the following index of refraction at wavelength 1550nm (only real part shown): $n_{eff,TE1} = 2.50, n_{eff,TM1} = 1.83$, $n_{eff,TE2} = 1.53$.
Other modes do not practically/physically exist having index of refraction value which is lower than the basic index of refraction of the material (n=1.44 for silicone). We can derive the effective index and the group index using a frequency sweep simulation, with wavelengths in range 1500-1600nm, as shown in Fig.3 and Fig. 4 (Lumerical MODE plots). We can see that the effective index decreases with wavelength and the group index increases with wavelength. Note: for the TM mode both the effective index and the group index decrease with wavelength. From now on we will proceed with the TE mode only. Also, for a central wavelength of 1550nm we get a group index: $n_g = 4.203$. Next, using a smart script (Appendix C), we fit the effective index results for TE1 mode using a 2nd order polynomial approximation and derive its coefficients (shown in Fig. 5). Note: The approximation is only close the real behavior only about wavelengths in range 1500-1600nm.
$n_{eff}(\lambda) = n_1 + n_2(\lambda - 1.55) + n_3(\lambda - 1.55)^2$,λ is the wavelength in microns. The fit results:
$n_{eff}(\lambda) = 2.44 + -1.13(\lambda - 1.55) + -0.043(\lambda - 1.55)^2$

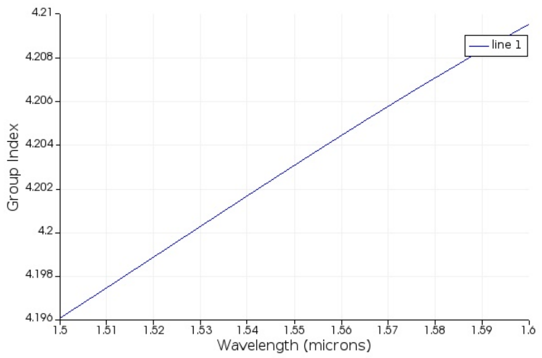


Fig. 4

Waveguide's group index Vs. wavelength for TE1 mode

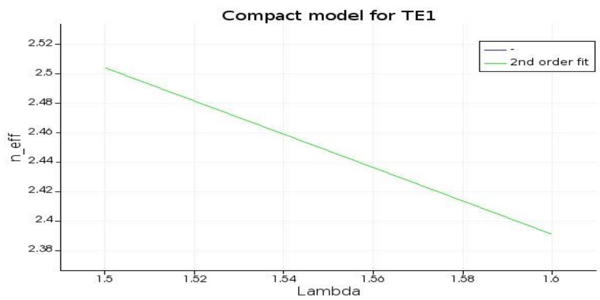


Fig. 5

Compact model of TE1

Mutual Parameter table:

Silicon Substrate	500x220nm
Wavelength range	1500-1600nm, centered at 1550nm (most used wavelength in long range optical communication)
Loss	69db/cm
Period	318nm
Δw	80nm
Δn	0.08
Compact model polynomial	$n_{eff}(\lambda) = 2.44 + -1.13(\lambda - 1.55) + -0.043(\lambda - 1.55)^2$
Resolution	0.0125nm
Span	40nm
Fabry Perot cavity length	1.94μm
Corrugation type	Rectangle

Note: Δn was derived from empiric results which are shown in Fig.6 (see reference [3]). My simulation double sweeps NG1 and NG2 valuse with base NG1 value of **50** and base NG2 value of **60**. I chose these numbers because the insertion loss is too big (over -10db) for larger numbers of gratings. All the results are shown in Fig 7A. Simulation results for NG1=60 NG2=130 are demonstrated in Fig.7B and Fig.8. Quality Factor and Insertion loss results in the tables below. In orange the results with best Quality factor and Insertion loss, these are the ones that were fabricated:

Experimental data – Δn

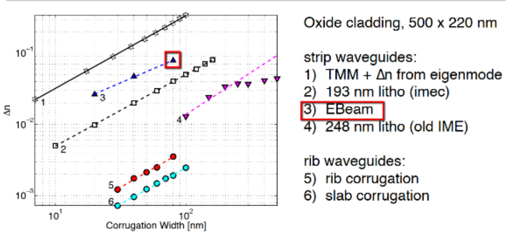


Fig. 6

Relation between dn and dw

Quality Factor

NG1/NG2	50	90	130	170	210
60	3005.23	2600.57	3100.89	3043.44	3035.86
100	3020.42	3177.50	3316.54	3188.89	3187.08
140	2835.45	2480.01	2973.92	2723.06	3135.37
180	3131.33	3088.67	3097.89	2948.69	3182.80
220	3123.54	3173.99	3109.20	3130.62	2973.72
260	3087.36	3103.31	2982.01	3026.12	3094.53

Insertion Loss

NG1/NG2	50	90	130	170	210
60	-0.8	-4.8	-15.3	-26.2	-38.2
100	-9.3	-13.3	-23.7	-35.3	-20.5
140	-14.7	-21.1	-31.9	-43.0	-31.0
180	-26.1	-32.6	-43.0	-54.0	-43.5
220	-37.0	-43.5	-54.8	-65.7	-54.8
260	-49.4	-54.3	-65.8	-74.3	-43.1

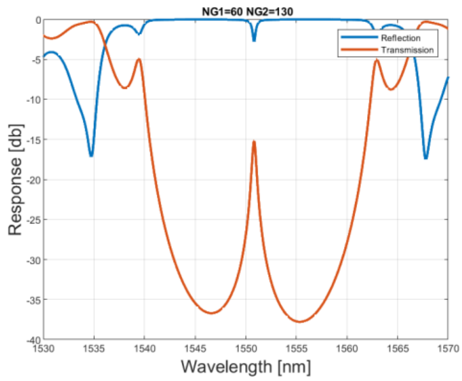


Fig. 7B

Reflection(blue) and Transmission(orange) of the cavity vs. wavelength

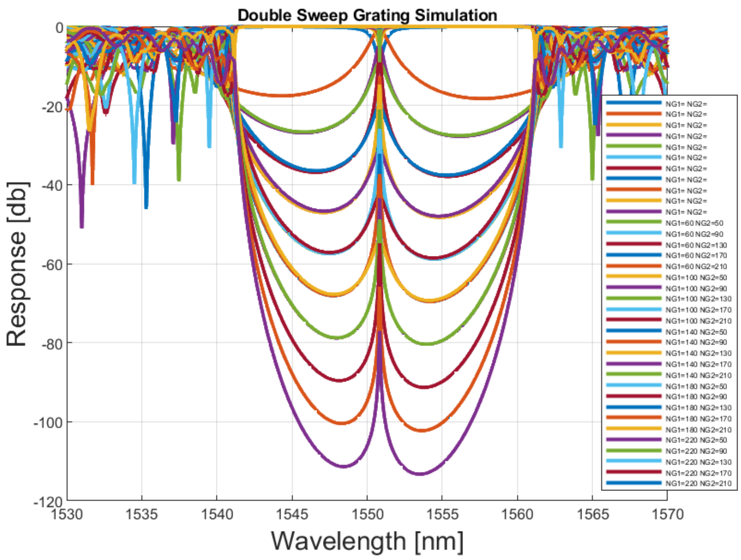


Fig. 7A

All simulation results

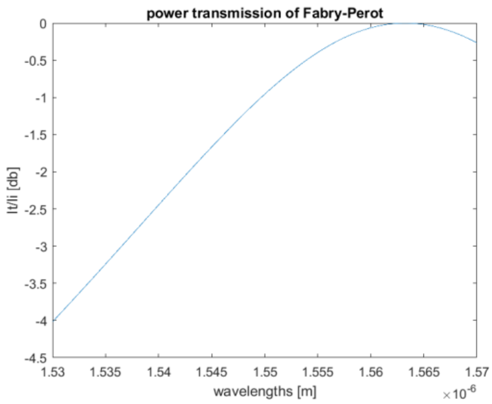


Fig. 8

NG1=60 NG2=130 Fabry-Perot power transmission ratio vs. wavelength

Mask Layout

The Klayout design that was sent for fabrication includes 8 different Brag Gratings Designs and a tester. The layout for each design includes 3 grating couplers connected to 2 Bragg gratings through a broadband splitter. The top coupler measures the reflection and the lower coupler measures the Transmission. Tester/Calibration unit measures the loss of a waveguides transmitted from a lasing coupler to a grating coupler detector In Fig. 9A-9C the layouts are shown using ‘Klayout’.

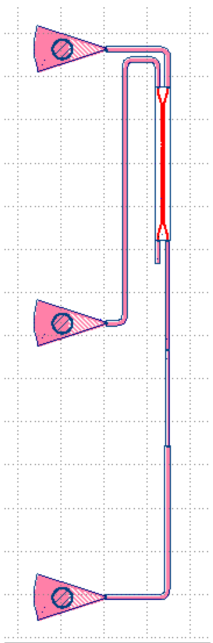


Fig. 9A

Klayout layout design

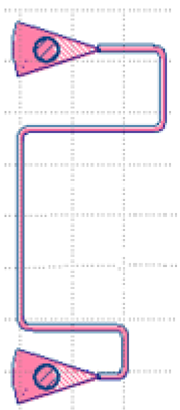


Fig. 9B

Tester Design

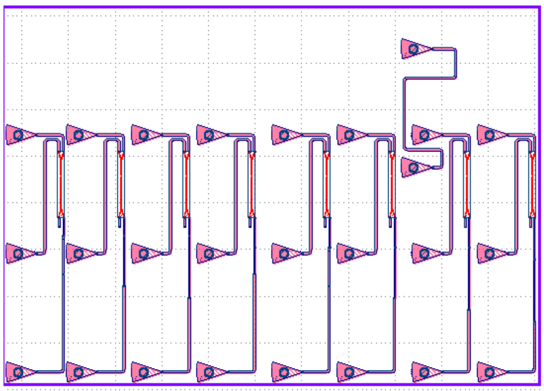


Fig. 9C

Entire Design

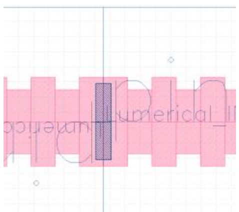


Fig. 9D

Back to back illustration

Fabrication

Chips are fabricated at two foundries: Applied Nanotools and Washington Nanofabrication Facility. More details attached in Appendix A.

Test Methodology

To characterize the devices, a custom-built automated test setup) see references [4] and [5] with automated control software written in Python was used (see reference [6]). An Agilent 81600B tunable laser was used as the input source and Agilent 81635A optical power sensors as the output detectors. The wavelength was swept from 1500 to 1600 nm in 10 pm step. A polarization maintaining (PM) fibre was used to maintain the polarization state of the light, to couple the TE polarization into the grating couplers (see reference [7]). A polarization maintaining fibre array was used to couple light in/out of the chip (see reference [8]).

Results

The results from UW were analysed as they provide better resolution. The best 4 results were compared to the simulation data (the results with clear, high and narrow Transmission Peak were chosen). Results compared to Matlab and InterConnect simulation (Zoomed in results) are shown in Fig 10A-10D. Other results are shown in Appendix B.

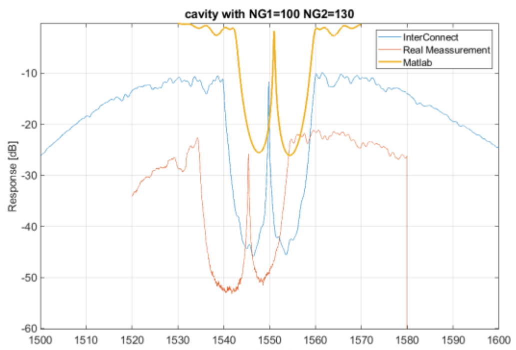


Fig. 10A

Design with NG1=100, NG2=130

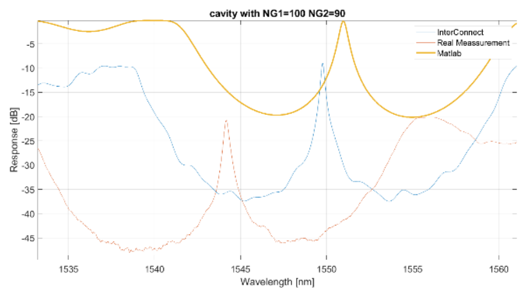


Fig. 10B

Design with NG1=100, NG2=90

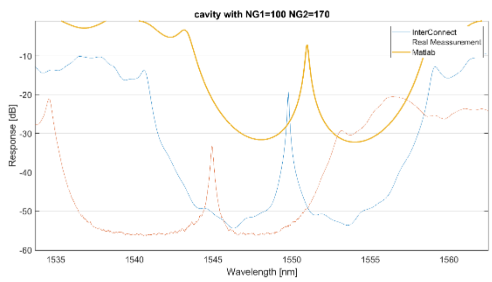


Fig. 10C

Design with NG1=100, NG2=170

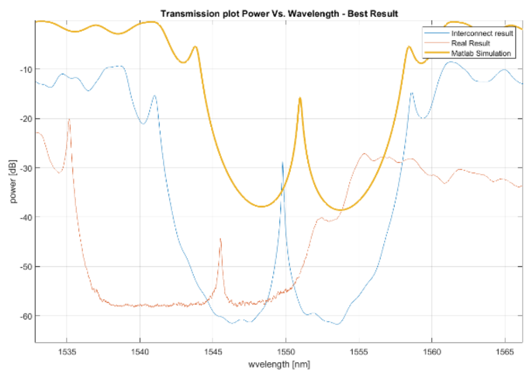


Fig. 10D

Design with NG1=100, NG2=210

Results Analysis

The 4 best results were analysed. The Quality Factor was extracted using a Matlab Lorentzian Curve Fit and the Insertion Loss was calculated using Matlab. Codes attached in Appendix C. The results are shown in the table below:

Bragg	Rpeak [dB]	Lorentzian Q	Fitted	Insertion loss experimental [dB]
NG1=100 NG2=130	-25.88	9638	Yes	-23
NG1=100 NG2=170	-33.27	9841	Yes	-16
NG1=100 NG2=210	-44.27	2540	No	-26
NG1=100 NG2=90	-20.73	5546	No	-7

The best result considering all the parameters is the NG1=100 NG2=170 (marked in yellow), and this is the one we proceeded with. The highly sampled Lorentzian curve fit for the result about the Transmission Peak:

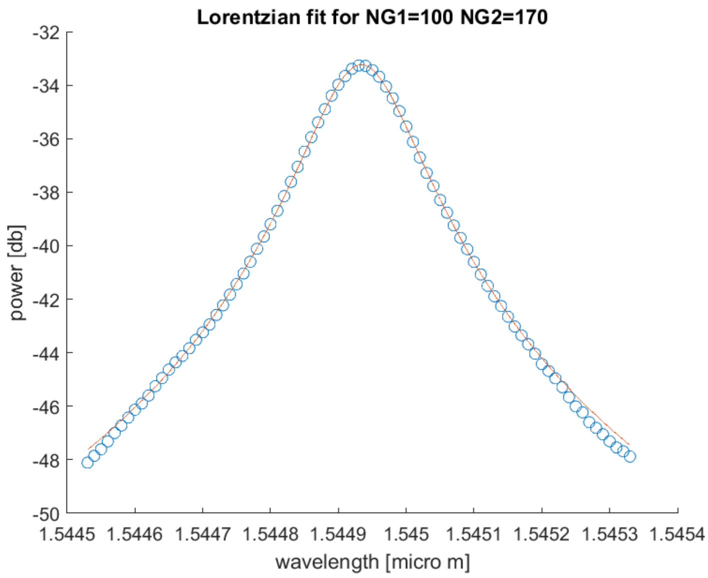
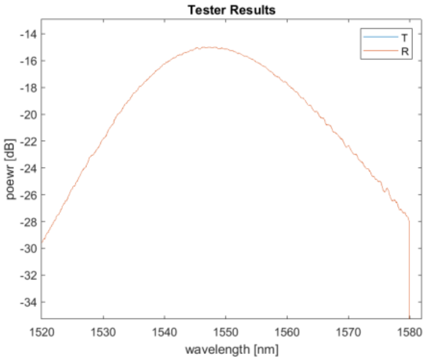


Fig. 11

40 points Lorentzian fit for best result

The couplers transmission loss was extracted from the 'tester coupler', with minimum loss of -15dB and maximum loss of -27.74. The average loss was taken:

$$\alpha_{coupler} \cong -19.89 \left[\frac{dB}{cm}\right]$$



Tester loss extraction illustration

Discussion

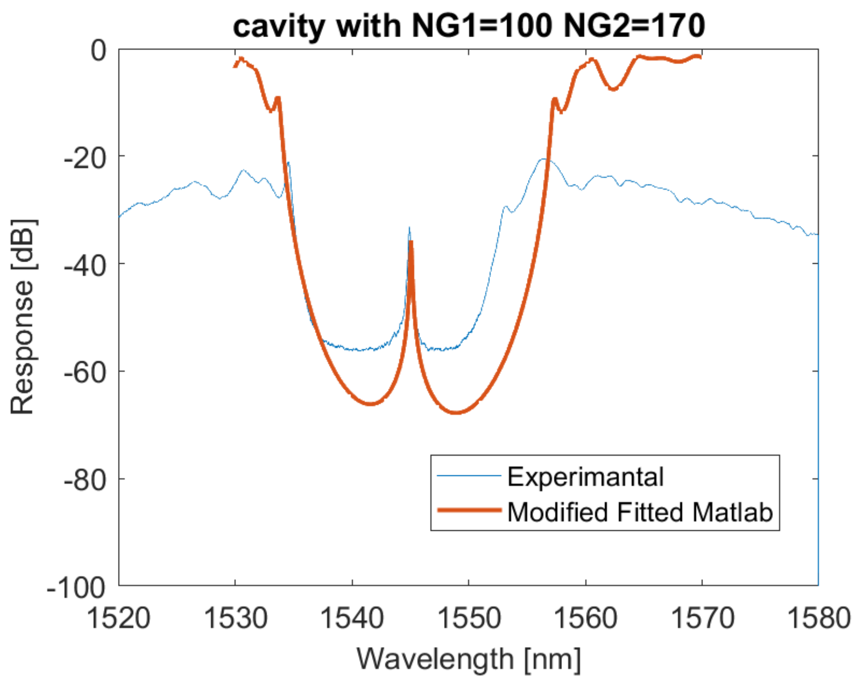


Fig. 13

Fitted Simulation to best result

As seen in Fig. 10A-10D, the resonance wavelength is smaller (left shift) compared to the simulation (Matlab), and the peak is lower and narrower. One reason for the lower peak is the -3dB loss assumption in the model (Matlab) which is not the loss in the real device. Furthermore, the fabricated detector can't detect power below -60dB, which explains why we didn't get "Round Transmission Curves" in the experimental results below -60db. The device with the best performance has NG1=100, NG2=170 as discussed before. Fabrication Limitations and errors should also be considered, resulting in a different Bragg Period or different Corrugation Width (dw). The Fitted results are shown in Fig. 13 together with the parameters which were used for the fitting.

Device	Original NG1=100 NG2=130	FITTED NG1=100 NG2=130
Coupler loss [dB/cm]	-3	-19.89
Period [nm]	318	316
Delta_n []	0.04	0.09

Conclusion

In conclusion, we managed to design a device that produces a high Quality Factor of 9841 and a low insertion loss of -16dB, as well as a well-fitted Lorentzian curve fit. The experimental results are different than the InterConnect and Matlab simulation due to various factors such as fabrication limitations and errors, temperature influence on the chip causing a frequency shift that can explain the shifted results, Interference from other devices, detector limitations. Future experiments are suggested with varying corrugation width and grating period, to assure the speculation of fabrications errors.

Acknowledgments

I acknowledge the edX UBCx Phot1x 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. The devices were fabricated by a) Richard Bojko at the University of Washington Washington Nanofabrication Facility, part of the National Science Foundation's National Nanotechnology Infrastructure Network (NNIN), and/or b) Cameron Horvath at Applied Nanotools, Inc. Hossam Shoman performed the measurements at The University of British Columbia. I acknowledge Lumerical Solutions, Inc., Mathworks, Python, and KLayout for the design software.

References

[1] Course Slide Set ELEC413-2019-Week3-Bragg.pdf Slide 73.

[2] Course Slide Sets: ELEC413-2019-Week3-Bragg.pdf, ELEC413-2019-FP.pdf.

[3] Course Slide Set ELEC413-2019-Week3-Bragg.pdf Slide 57.

[4] Lukas Chrostowski, Michael Hochberg, chapter 12 in "Silicon Photonics Design: From Devices to Systems", Cambridge University Press, 2015.

[5] http://mapleleafphotonics.com, Maple Leaf Photonics, Seattle WA, USA.

[6] http://siepic.ubc.ca/probestation, using Python code developed by Michael Caverley.

[7] Yun Wang, Xu Wang, Jonas Flueckiger, Han Yun, Wei Shi, Richard Bojko, Nicolas A. F. Jaeger, Lukas Chrostowski, "Focusing sub-wavelength grating couplers with low back reflections for rapid prototyping of silicon photonic circuits", Optics Express Vol. 22, Issue 17, pp. 20652-20662 (2014).

[8] www.plcconnections.com, PLC Connections, Columbus OH, USA.

[9] R. J. Bojko, J. Li, L. He, T. Baehr-Jones, M. Hochberg, and Y. Aida, "Electron beam lithography writing strategies for low loss, high confinement silicon optical waveguides," J. Vacuum Sci. Technol. B 29, 06F309 (2011).

[10] Codes written by Xu Wang, UBC.

[11] Code taken from edX WaveGuide Compact Model Unit. https://edge.edx.org/courses/course-v1:UBC+ELEC413-201+2018W2/courseware/45441ee6e8f246b487913dd34c013558/96743f4fce7e4874a6a5e4ad702fc6c2/7?activate_block_id=block-v1%3AUBC%2BELEC413-201%2B2018W2%2Btype%40vertical%2Bblock%40680d11eca25a4551a84d7b2da71b7240

**All the rest is my code/my results/my plots/my layouts.

Appendix A – Fabrication Details

The following are the process descriptions.

Applied Nanotools, Inc. NanoSOI process:

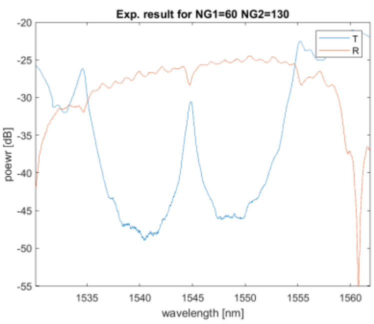
The photonic devices were fabricated using the NanoSOI MPW fabrication process by Applied Nanotools Inc. (http://www.appliednt.com/nanosoi; Edmonton, Canada) which is based on direct-write 100 keV electron beam lithography technology. Silicon-on-insulator wafers of 200 mm diameter, 220 nm device thickness and 2 µm buffer oxide thickness are used as the base material for the fabrication. The wafer was pre-diced into square substrates with dimensions of 25x25 mm, and lines were scribed into the substrate backsides to facilitate easy separation into smaller chips once fabrication was complete. After an initial wafer clean using piranha solution (3:1 H_2SO_4:H_2O_2) for 15 minutes and water/IPA rinse, hydrogen silsesquioxane (HSQ) resist was spin-coated onto the substrate and heated to evaporate the solvent. The photonic devices were patterned using a Raith EBPG 5000+ electron beam instrument using a raster step size of 5 nm. The exposure dosage of the design was corrected for proximity effects that result from the backscatter of electrons from exposure of nearby features. Shape writing order was optimized for efficient patterning and minimal beam drift. After the e-beam exposure and subsequent development with a tetramethylammonium sulfate (TMAH) solution, the devices were inspected optically for residues and/or defects. The chips were then mounted on a 4" handle wafer and underwent an anisotropic ICP-RIE etch process using chlorine after qualification of the etch rate. The resist was removed from the surface of the devices using a 10:1 buffer oxide wet etch, and the devices were inspected using a scanning electron microscope (SEM) to verify patterning and etch quality. A 2.2 µm oxide cladding was deposited using a plasma-enhanced chemical vapour deposition (PECVD) process based on tetraethyl orthosilicate (TEOS) at 300ºC. Reflectrometry measurements were performed throughout the process to verify the device layer, buffer oxide and cladding thicknesses before delivery.

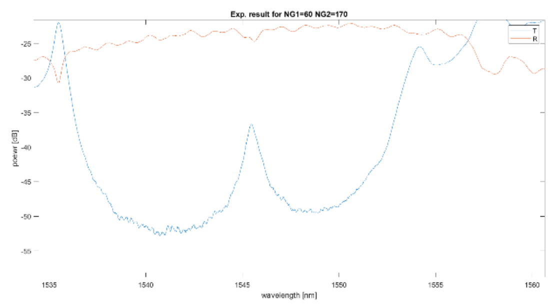
Washington Nanofabrication Facility (WNF) silicon photonics process:

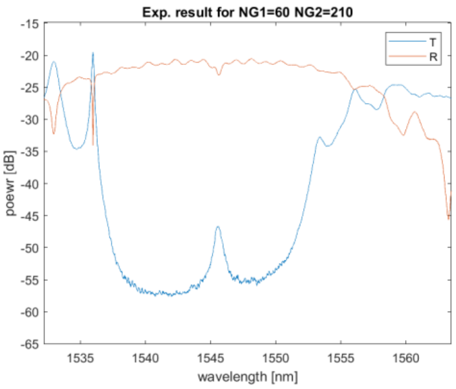
The devices were fabricated using 100 keV Electron Beam Lithography (see Reference [9]). The fabrication used silicon-on-insulator wafer with 220 nm thick silicon on 3 µm thick silicon dioxide. The substrates were 25 mm squares diced from 150 mm wafers. After a solvent rinse and hot-plate dehydration bake, hydrogen silsesquioxane resist (HSQ, Dow-Corning XP-1541-006) was spin-coated at 4000 rpm, then hotplate baked at 80 °C for 4 minutes. Electron beam lithography was performed using a JEOL JBX-6300FS system operated at 100 keV energy, 8 nA beam current, and 500 µm exposure field size. The machine grid used for shape placement was 1 nm, while the beam stepping grid, the spacing between dwell points during the shape writing, was 6 nm. An exposure dose of 2800 µC/cm2 was used. The resist was developed by immersion in 25% tetramethylammonium hydroxide for 4 minutes, followed by a flowing deionized water rinse for 60 s, an isopropanol rinse for 10 s, and then blown dry with nitrogen. The silicon was removed from unexposed areas using inductively coupled plasma etching in an Oxford Plasmalab System 100, with a chlorine gas flow of 20 sccm, pressure of 12 mT, ICP power of 800 W, bias power of 40 W, and a platen temperature of 20 °C, resulting in a bias voltage of 185 V. During etching, chips were mounted on a 100 mm silicon carrier wafer using perfluoropolyether vacuum oil. Cladding oxide was deposited using plasma enhanced chemical vapor deposition (PECVD) in an Oxford Plasmalab System 100 with a silane (SiH4) flow of 13.0 sccm, nitrous oxide (N2O) flow of 1000.0 sccm, high-purity nitrogen (N2) flow of 500.0 sccm, pressure at 1400mT, high-frequency RF power of 120W, and a platen temperature of 350C. During deposition, chips rest directly on a silicon carrier wafer and are buffered by silicon pieces on all sides to aid uniformity.

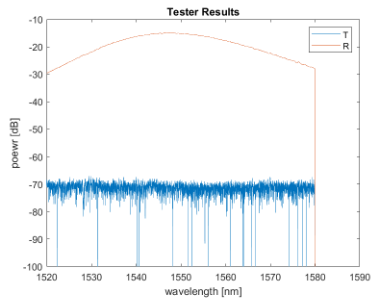
Appendix B – Results

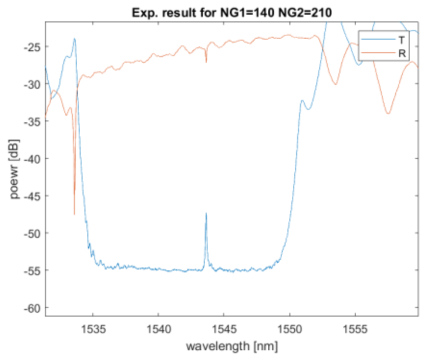
Hereby attached results for designs which did not have a sufficient Transmission peak (high value and narrow peak), resulting in low Quality Factor and low Transmission. Furthermore, the tester result is attached.











Appendix C - Matlab code:

Code for Fitting 2nd order polynomial for Lumerical Mode effective index

See reference [11].

Extracting the coefficients: nx(1) nx(2) nx(3)

```
% Phot1x_fit_wg_compactmoderel.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/xv4he4preyfa9v2/wg-export-TM.mat?dl=1'
url='https://s3.amazonaws.com/edx-course-phot1x-chrostowski/Phot1x/wg-export-TM.mat'
a=websave('wg.mat',url); % get data from Dropbox
load('wg.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 = 1.55;   % 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')
```

Code used for double sweeping Matlab simulations

See reference [10].

```
% By Xu Wang, UBC
% Modified by Or Bahari, UBC

%code for NG1 and NG2 double-sweep%

function main

clear;
clc;
global Bragg n_eff NG1 NG2;
Bragg=1550e-9;  % Bragg wavelength
span=40e-9;  % Set the wavelength span for the simultion
resolution=0.0125e-9;  % Set the wavelength resolution
N=span/resolution;
disp([ 'Number of points: ' num2str(N) ])
Lambda=zeros(N+1,1);
R=zeros(N+1,1);
T=zeros(N+1,1);

jmax=6;
kmax=5;
%jmax=1;
%kmax=1;
for j=1:jmax
    for k=1:kmax
        for i=1:N+1
            wavelength=Bragg+(i-1-N/2)*resolution;    % Wavelength sweep
            Lambda(i)=wavelength*1e9;% in nm
            Grating_Parameters(Lambda(i),j-1,k-1,40)
            [r,t]=Grating_RT(wavelength); % Calculate the R and T
            R(i)=r;
            T(i)=t;
        end
        %figure(j);
        T_db=10*log10(T);
        R_db=10*log10(R);
        %subplot(1,kmax,k);
        plot(Lambda,[R_db T_db],'LineWidth',2);
        hold on
        set(gca,'FontSize',8);
        xlabel('Wavelength [nm]','FontSize',14);
        ylabel('Response [db]','FontSize',14);
        %title([' NG1=',num2str(NG1),' NG2=',num2str(NG2)]);
        legend('Reflection','Transmission');
        grid on;
       % Rpeak(j,k)=Center_peak_extraction(R,120);
        % figure(10+j);
        %[T_peak,Q(j,k),insertion_Loss(j,k)]=FP_transmission(Lambda*1e-9,R,k,kmax);

    end

end
    set(gca,'FontSize',8);
        xlabel('Wavelength [nm]','FontSize',14);
        ylabel('Response [db]','FontSize',14);
        %title([' NG1=',num2str(NG1),' NG2=',num2str(NG2)]);
        legend('Reflection','Transmission');
        grid on;
        hold off;
        legend('9');

a=3; %for DEBUG
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Grating_Parameters(Lambda,index1,index2,st)
%Set the parameters
global Period NG1 NG2 delta_n n1 n2 n_eff loss C_length;

%arm_length=2.5e-6;
Period=318e-9;  % Bragg period
n_eff=2.44 - 1.13 * ((Lambda/1000.0)-1.550)-0.043*((Lambda/1000.0)-1.550)^2;
NG1=60+index1*st;     % Number of grating periods #1
NG2=50+index2*st;     % Number of grating periods #2

L1=NG1*Period;     % Grating length #1
L2=NG2*Period;     % length #2
C_length=L1+L2; % total cavity length

delta_n=0.08;     % Index contrast between n1 and n2, matches to dw=40nm
n1=n_eff-delta_n/2;
n2=n_eff+delta_n/2;

loss=590; % (what was said in class we should use...)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [R,T]=Grating_RT(wavelength)
%Calculate the R and T for a certain wavelength

M=Grating_Matrix(wavelength);

T=abs(1/M(1,1))^2;
R=abs(M(2,1)/M(1,1))^2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Rpeak,position]=Center_peak_extraction(R,width)
%returns the Rpeak and it's index. to get the wavelength later we should
%use wavlengths[pos] (!)
Length=length(R);
[Rpeak,position]=min(R(Length*0.5:width+Length*0.5));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [TP,Q,Iloss]= FP_transmission(wavelengths,R,plot_index,total_plots)
    %wavelengths Units: m
    %Calculating Rpeak and pos, then Q factor, Insertion_Loss
    global  n_eff C_length loss
    delta=-4*pi*1.94*1e-6*n_eff./wavelengths;
    % 1.94*1e-6

    %finding the peak of R:
    acc=120;
    [Rpeak,pos]=Center_peak_extraction(R,acc);
    % modify the following line to output the result of your calculation, for the auto-grader
    output = ((1-Rpeak)^2)./(((1-Rpeak)^2)+4*Rpeak*(sin(delta/2)).^2);
    %output=(loss* (1-Rpeak).^2)./((1-loss*Rpeak).^2 +4*loss*Rpeak*(sin(delta/2)).^2);
    output10=10*log10(output);
    subplot(1,total_plots,plot_index);
    plot(wavelengths,output10);
    xlabel('wavelengths [m]');
    ylabel('It/Ii [db]');
    title('power transmission of Fabry-Perot');
    TP=max(output);
    Iloss=10*log(1-(0.1*10^(Rpeak)));
    %bw=powerbw(output,R); %Provides Bandwidth 3db
    bw=powerbw(R);
    Q=abs(wavelengths(pos)*10^9/bw);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [T]=Grating_Matrix(wavelength)
% Calculate the total transfer matrix of the gratings

global Period NG1 NG2;
global n1 n2 loss;

l=Period/2;

T_hw1=HomoWG_Matrix(wavelength,l,n1,loss);
T_is12=IndexStep_Matrix(n1,n2);
T_hw2=HomoWG_Matrix(wavelength,l,n2,loss);
```

```
T_is21=IndexStep_Matrix(n2,n1);
T1=T_hw1*T_is12*T_hw2*T_is21;

T=T1^(NG1)*T_hw1*T1^(NG2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function T_hw=HomoWG_Matrix(wavelength,l,neff,loss)
% Calculate the transfer matrix of a homogeneous waveguide.
%Complex propagation constant
beta=2*pi.*neff/wavelength-1i*loss/2;

v=[exp(1i*beta*l) exp(-1i*beta*l)];
T_hw=diag(v);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function T_is=IndexStep_Matrix(n1,n2)
% Calculate the transfer matrix for a index step from n1 to n2.

a=(n1+n2)/(2*sqrt(n1*n2));
b=(n1-n2)/(2*sqrt(n1*n2));
T_is=[a b; b a];
```

Experimental results extraction

```
clear all;
clc;

mat100130=load("orbahari_100130_1082 (1).mat");
mat10090=load("orbahari_10090_1085.mat");
mat100170=load("orbahari_100170_1083 (1).mat");
mat100210=load("orbahari_100210_1084 (1).mat");
%%
figure(82);
plot(mat100130.scandata.wavelength*1e9,mat100130.scandata.power(:,2));
%R=mat100130.scandata.power(:,1);
Rpeak100130=min(mat100130.scandata.power(2200:3000,1))
loss100130=10*log10(1-10^(Rpeak100130/10))
%%
figure(83);
plot(mat100170.scandata.wavelength*1e9,mat100170.scandata.power(:,2));
Rpeak100170=min(mat100170.scandata.power(2200:3000,1))
loss100170=10*log10(1-10^(Rpeak100170/10))
%%
figure(84);
plot(mat100210.scandata.wavelength*1e9,mat100210.scandata.power(:,2));
Rpeak100210=min(mat100210.scandata.power(2200:3000,1))
loss100210=10*log10(1-10^(Rpeak100210/10))
%%
figure(85);
plot(mat10090.scandata.wavelength*1e9,mat10090.scandata.power(:,2));
Rpeak10090=min(mat10090.scandata.power(2200:3000,1))
loss10090=10*log10(1-10^(Rpeak10090/10))

%% loading unused results
mat1087=load("orbahari_60130_1087 (1).mat");
mat1080=load("orbahari_60170_1080 (1).mat");
mat1081=load("orbahari_60210_1081 (1).mat");
mat1086=load("orbahari_140210_1086.mat");
tester=load("orbahari_tester_1088.mat");

%%
plot(tester.scandata.wavelength*1e9,tester.scandata.power(:,2));
title("Tester Results")
xlabel('wavelength [nm]')
ylabel('poewr [dB]')
hold on
plot(tester.scandata.wavelength*1e9,tester.scandata.power(:,1));
legend("T","R")
hold off;
```

Matlab Simulation Model

Note that the parameters are set to the fitting of the experimental result for the best device.

```
%Or Bahari
```

```
function main

clear;
clc;
global Bragg n_eff NG1 NG2;
Bragg=1550e-9;  % Bragg wavelength
span=40e-9;  % Set the wavelength span for the simultion
resolution=0.0125e-9;  % Set the wavelength resolution
N=span/resolution;
disp([ 'Number of points: ' num2str(N) ])
Lambda=zeros(N+1,1);
R=zeros(N+1,1);
T=zeros(N+1,1);

        for i=1:N+1
            wavelength=Bragg+(i-1-N/2)*resolution;     % Wavelength sweep
            Lambda(i)=wavelength*1e9;% in nm
            Grating_Parameters(Lambda(i))
            [r,t]=Grating_RT(wavelength); % Calculate the R and T
            R(i)=r;
            T(i)=t;
        end
T_db=10*log10(T);
R_db=10*log10(R);
hold on
plot(Lambda,T_db,'LineWidth',2);
set(gca,'FontSize',14);
xlabel('Wavelength [nm]','FontSize',14);
ylabel('Response [dB]','FontSize',14);
title(['cavity with NG1=',num2str(NG1),' NG2=',num2str(NG2)]);
legend('InterConnect','Real Meassurement','Matlab');
grid on;
Rpeak=Center_peak_extraction(R,120);
I_Loss=10*log10(1-Rpeak);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Grating_Parameters(Lambda)
%Set the parameters
global Period NG1 NG2 delta_n n1 n2 n_eff loss C_length;

%arm_length=2.5e-6;
Period=316e-9;  % 318
n_eff=2.44 - 1.13 * ((Lambda/1000.0)-1.550)-0.043*((Lambda/1000.0)-1.550)^2; %2.44
NG1=100;     % Number of grating periods #1
NG2=170;

L1=NG1*Period;     % Grating length #1
L2=NG2*Period;     % length #2
C_length=L1+L2; % total cavity length

delta_n=0.09;    % 0.04 Index contrast between n1 and n2, matches to dw=40nm
n1=n_eff-delta_n/2;
n2=n_eff+delta_n/2;

loss=1190; %69

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [R,T]=Grating_RT(wavelength)
%Calculate the R and T for a certain wavelength

M=Grating_Matrix(wavelength);

T=abs(1/M(1,1))^2;
R=abs(M(2,1)/M(1,1))^2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Rpeak]=Center_peak_extraction(R,width)
Length=length(R);
Rpeak=min(R(Length*0.5:width+Length*0.5));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [TP,Q]= FP_transmission(wavelengths,R)
    %wavelengths Units: m
    global  n_eff C_length loss
    delta=-4*pi* 1.94*1e-6*n_eff./wavelengths;
    % 1.94*1e-6
```

```
    %finding the peak of R:
    Rpeak=Center_peak_extraction(R,120);
    % modify the following line to output the result of your calculation, for the auto-grader
    output = ((1-Rpeak)^2)./(((1-Rpeak)^2)+4*Rpeak*(sin(delta/2)).^2);
    output10=10*log10(output);
    figure()
    %subplot(1,2,1);
    plot(wavelengths,output10);
    xlabel('wavelengths [m]');
    ylabel('It/Ii [db]');
    title('power transmission of Fabry-Perot');
    TP=max(output);
    %subplot(1,2,2);
    bw=powerbw(output,wavelengths); %Provides Bandwidth 3db
    Q=TP/bw;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [T]=Grating_Matrix(wavelength)
% Calculate the total transfer matrix of the gratings

global Period NG1 NG2;
global n1 n2 n_eff loss arm_length;

l=Period/2;

T_hw1=HomoWG_Matrix(wavelength,l,n1,loss);
T_is12=IndexStep_Matrix(n1,n2);
T_hw2=HomoWG_Matrix(wavelength,l,n2,loss);
T_is21=IndexStep_Matrix(n2,n1);
T1=T_hw1*T_is12*T_hw2*T_is21;

%code for back-to-back BGs
T_is22=IndexStep_Matrix(n2,n2);
T2=T_hw1*T_is12*T_hw2*T_is22*T_hw2*T_is21;

%code to silicon separtor between 2 back-to-back BGs
%T_is2n= IndexStep_Matrix(n2,n_eff);
%T_hwn=HomoWG_Matrix(wavelength,arm_length,n_eff,loss);
%T_isn1=IndexStep_Matrix(n_eff,n1);
%T2=T_hw1*T_is12*T_hw2*T_is2n*T_hwn*T_isn1;
%T=T1^(NG1)*T2*T1^(NG2);
T=T1^(NG1)*T_hw1*T1^(NG2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function T_hw=HomoWG_Matrix(wavelength,l,neff,loss)
% Calculate the transfer matrix of a homogeneous waveguide.
%Complex propagation constant
beta=2*pi.*neff/wavelength-1i*loss/2;

v=[exp(1i*beta*l) exp(-1i*beta*l)];
T_hw=diag(v);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function T_is=IndexStep_Matrix(n1,n2)
% Calculate the transfer matrix for a index step from n1 to n2.

a=(n1+n2)/(2*sqrt(n1*n2));
b=(n1-n2)/(2*sqrt(n1*n2));
T_is=[a b; b a];
```