OptiSPICE

Tutorials - Advanced

Opto-Electronic Circuit Design Software

Version 5.2

Optiwave

OptiSPICE

Tutorials - Advanced

Opto-Electronic Circuit Design Software

Disclaimer

Optiwave makes no representation or warranty with respect to the adequacy of this documentation or the programs which it describes for any particular purpose or with respect to its adequacy to produce any particular result. In no event shall Optiwave, its employees, its contractors, or the authors of this documentation be liable for special, direct, indirect, or consequential damages, losses, costs, charges, claims, demands, or claim for lost profits, fees, or expenses of any nature or kind.

Table of Contents

Advanced Tutorials

To learn more about the advanced features available with OptiSPICE 5.2 it is recommended to perform the tutorials included in this document. Included in this document are the following examples:

- ***OptiSPICE Netlist Commands*** which shows how to use the OptiSPICE command menu to quickly setup simulation parameters.
- ***Python Post Processing*** which includes various design examples that demonstrate how to use Python post processing for viewing simulation results
- ***OptiSystem and OptiSPICE Co-simulation*** which shows the steps involved in setting up a co-simulation between OptiSystem and OptiSPICE (for a transceiver circuit)
- ***Optical/Electrical Signal File Input*** which shows how to run simulations with user defined inputs using the ***Vpwl-File input*** (voltage) element
- ***Creating Sub-circuits*** which shows how to create a sub-circuit from a combination of OptiSPICE devices (in this case an optical Chebyshev filter).

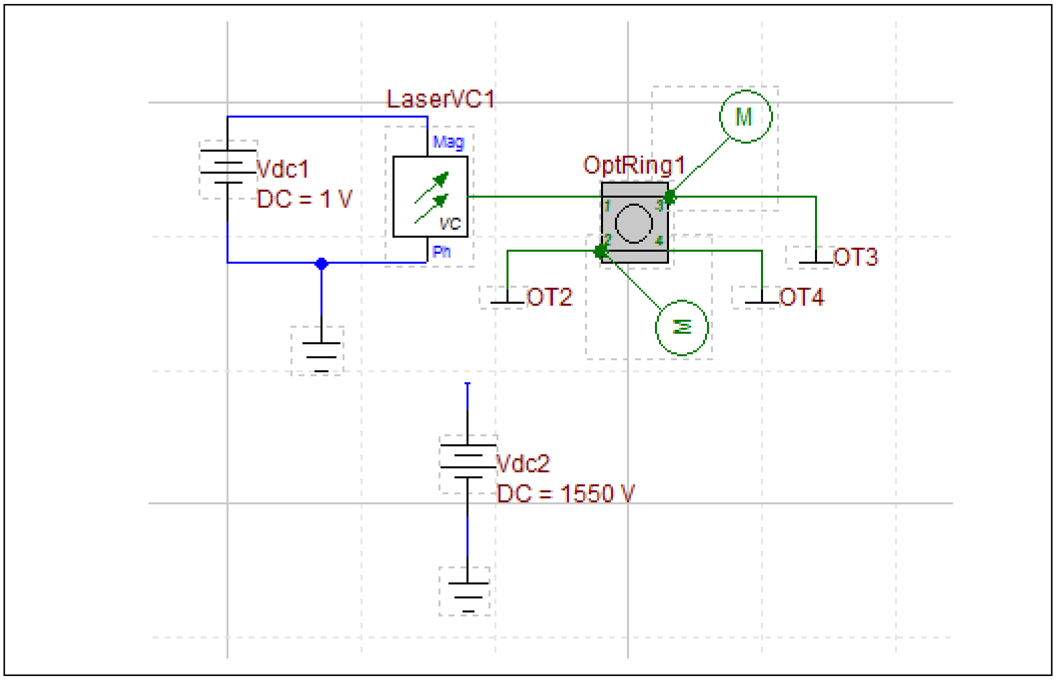
OptiSPICE Netlist Commands

The OptiSPICE command menu is used for directly adding commands to the OptiSPICE Netlist. It fully supports the Netlist commands described in ***OptiSPICE Simulator Command Reference Guide***.

Perform the following steps to setup a lambda sweep simulation with OptiSPICE commands

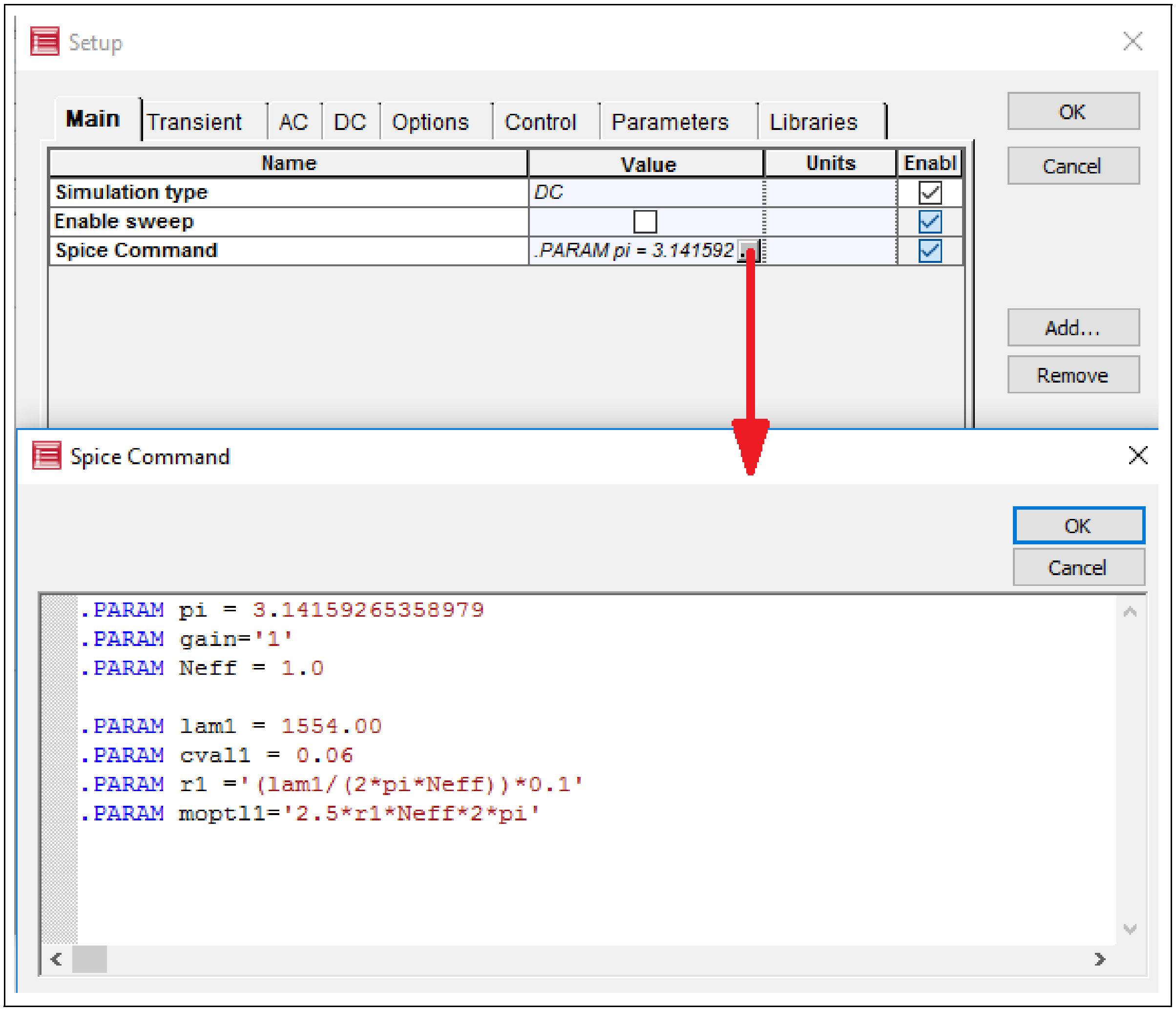
Step	Action
1	Drag and drop components and connect them as shown in Figure 1.

Figure 1 Ring resonator circuit



Step	Action
2	Go to **Analysis/Setup** and open the SPICE command menu (see Figure 2).
3	Enter the following parameters and equations as shown in Figure 2 (right-side) *They will be used to set up a ring resonator with a user defined resonant frequency.*

Figure 2 SPICE Command Window

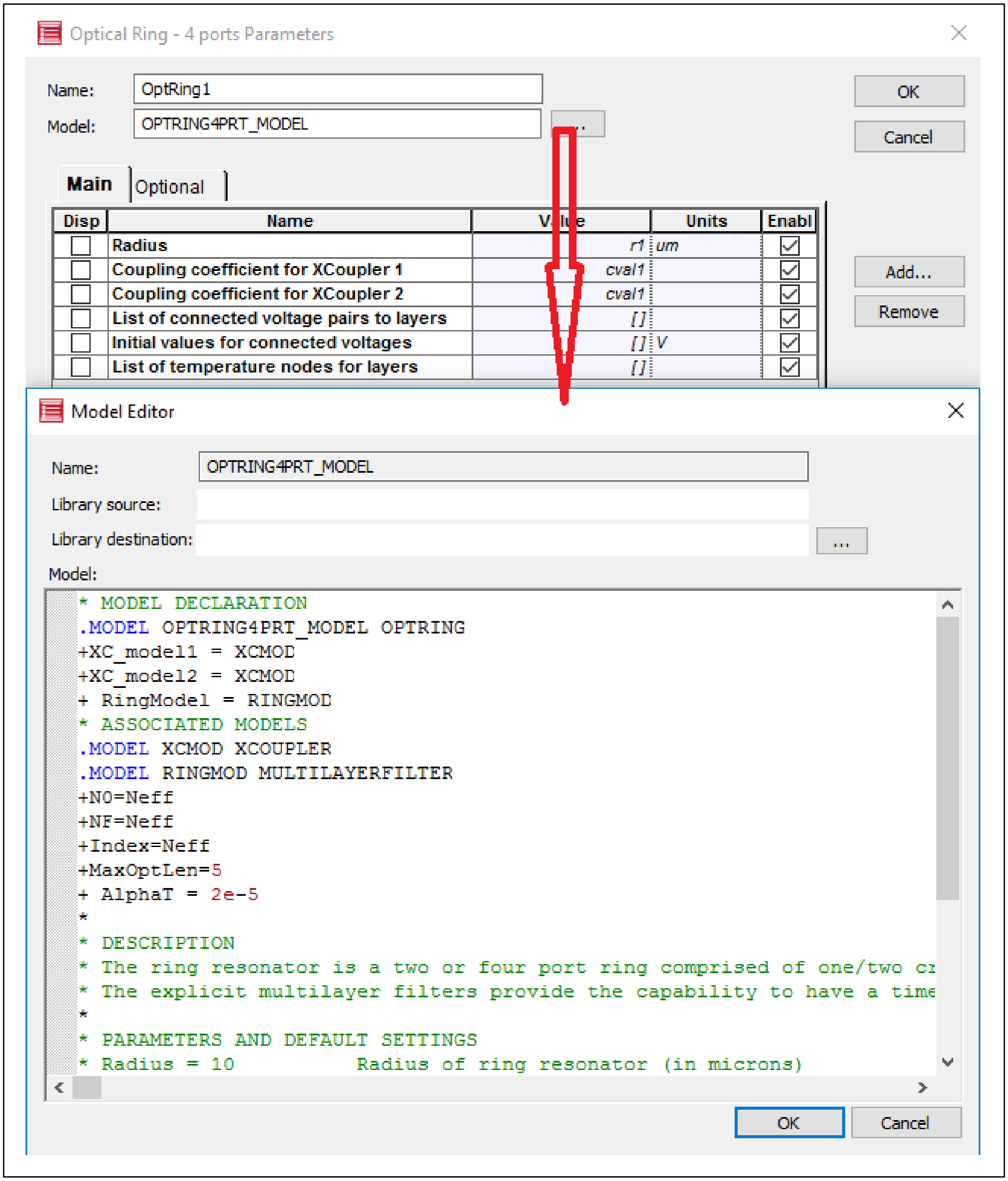


Name	Value	Units	Enabl
Simulation type	DC		☑
Enable sweep	□		☑
Spice Command	.PARAM pi = 3.141592		☑

```
.PARAM pi = 3.14159265358979
.PARAM gain='1'
.PARAM Neff = 1.0

.PARAM lam1 = 1554.00
.PARAM cval1 = 0.06
.PARAM r1 ='(lam1/(2*pi*Neff))*0.1'
.PARAM moptl1='2.5*r1*Neff*2*pi'
```

4 Set up the ring resonator element and model parameters as shown in Figure 3. Please note that the Radius and Coupling coefficient parameters must be entered as text parameters (“r1” and “cval1”) to ensure that the values are correctly retrieved from the associated parameters defined in the OptiSPICE command line.

Figure 3 4 Port Micro-Ring resonator parameters



Name: OptRing1

Model: OPTRING4PRT_MODEL

Disp	Name	Value	Units	Enabl
□	Radius	r1	um	☑
□	Coupling coefficient for XCoupler 1	cval1		☑
□	Coupling coefficient for XCoupler 2	cval1		☑
□	List of connected voltage pairs to layers	[]		☑
□	Initial values for connected voltages	[]	V	☑
□	List of temperature nodes for layers	[]		☑

Name: OPTRING4PRT_MODEL

Library source:

Library destination:

Model:

```
* MODEL DECLARATION
.MODEL OPTRING4PRT_MODEL OPTRING
+XC_model1 = XCMOD
+XC_model2 = XCMOD
+ RingModel = RINGMOD
* ASSOCIATED MODELS
.MODEL XCMOD XCOUPLER
.MODEL RINGMOD MULTILAYERFILTER
+N0=Neff
+NF=Neff
+Index=Neff
+MaxOptLen=5
+ AlphaT = 2e-5
*
* DESCRIPTION
* The ring resonator is a two or four port ring comprised of one/two cr
* The explicit multilayer filters provide the capability to have a time
*
* PARAMETERS AND DEFAULT SETTINGS
* Radius = 10          Radius of ring resonator (in microns)
```

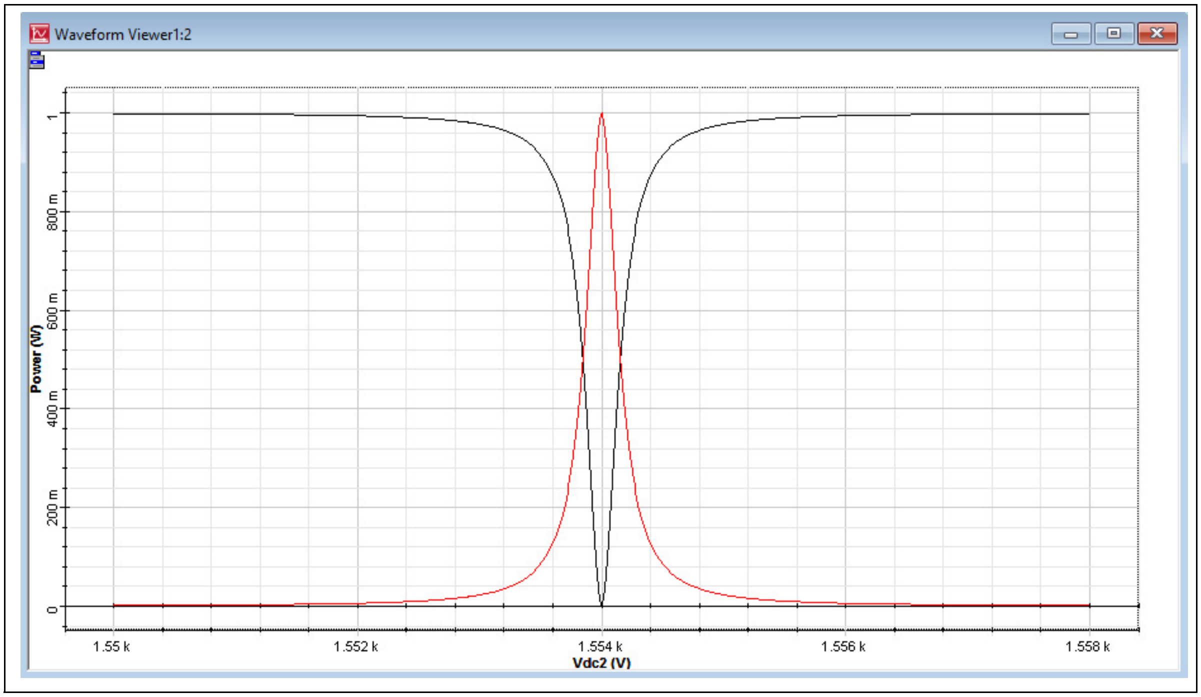
5 Set the up the VCLaser such that its frequency is controlled by Vdc2. Set up a sweep between 1550 nm and 1558 nm with 0.01 increments (see the tutorial “Wavelength Sweep” in *OptiSPICE_Tutorials_Basic* for more information on how to setup a wavelength sweep).

Running the simulation and viewing results

Running the simulation is same as for transient analysis. Save the design and select **Analysis > Run**. Click on ***Launch Waveform Viewer*** once the simulation ends.

After running the simulation, you can directly plot the results from the waveform viewer (see Figure 4)

Figure 4 Micro ring resonator output



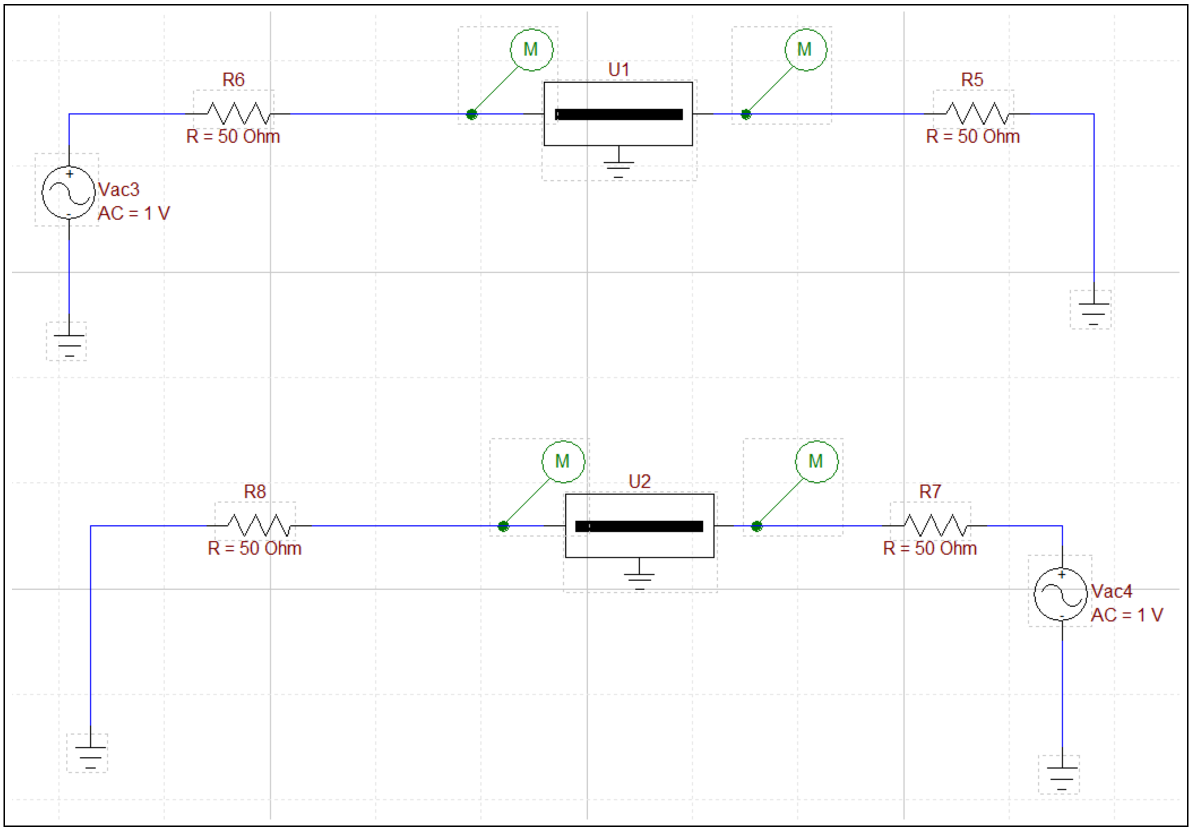
Python Post Processing

Please see the ***OptiSPICE Python Post Processing Guide*** to learn more on how to setup Python post processing for your simulation results.

Transmission line characteristics

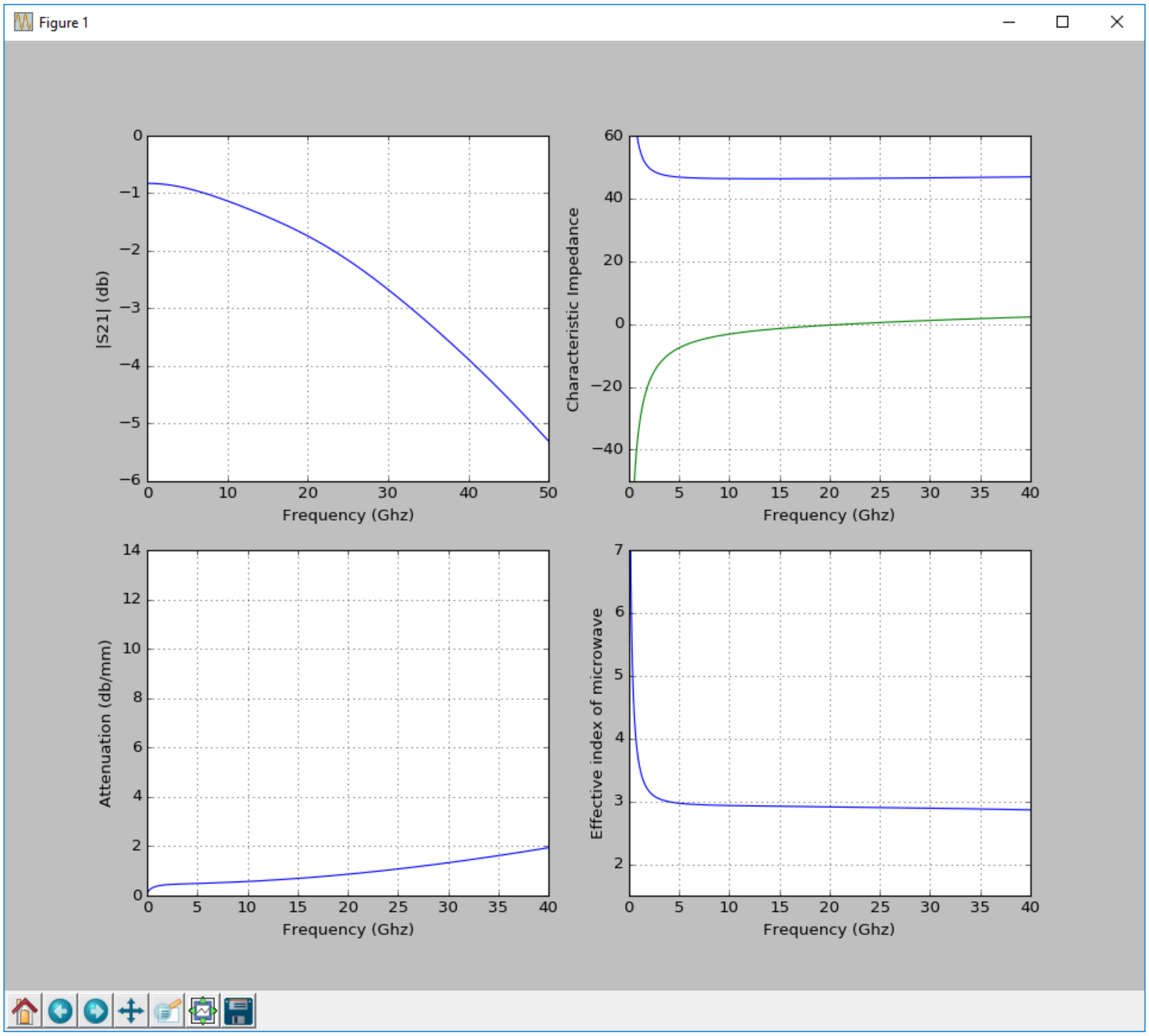
In this example (see Figure 5), AC simulation is used to measure the characteristics of a long metal contact modeled as a transmission line used for a traveling wave modulator with a 50 ohm load.

Figure 5 Measuring the frequency domain characteristics of a TWMZM using OptiSPICE



The simulation output (see Figure 6) is processed in Python to generate various characteristics for transmission line such as attenuation, effective index of microwave and characteristic impedance. The Python script, *SparamScript.py*, and associated OptiSPICE Schematic can be found in *OptiSPICE 5.2 Samples\Tutorials\Advanced\Python Post Processing\Transmission Line.*

Figure 6 Frequency domain characteristics of the TWMZM generated by Python post processing



Eye Histogram

This example (see Figure 7) illustrates the generation of eye histograms via Python. The eye histograms in Figure 8 show the signal received by the photo-diode before (top) and after (below) RC filtering. The Python script, *ElecEyeDiag.py*, and associated OptiSPICE Schematic can be found in *OptiSPICE 5.2 Samples\Tutorials\Advanced\Python Post Processing\Eye Diagram.*

Figure 7 Optical transceiver circuit

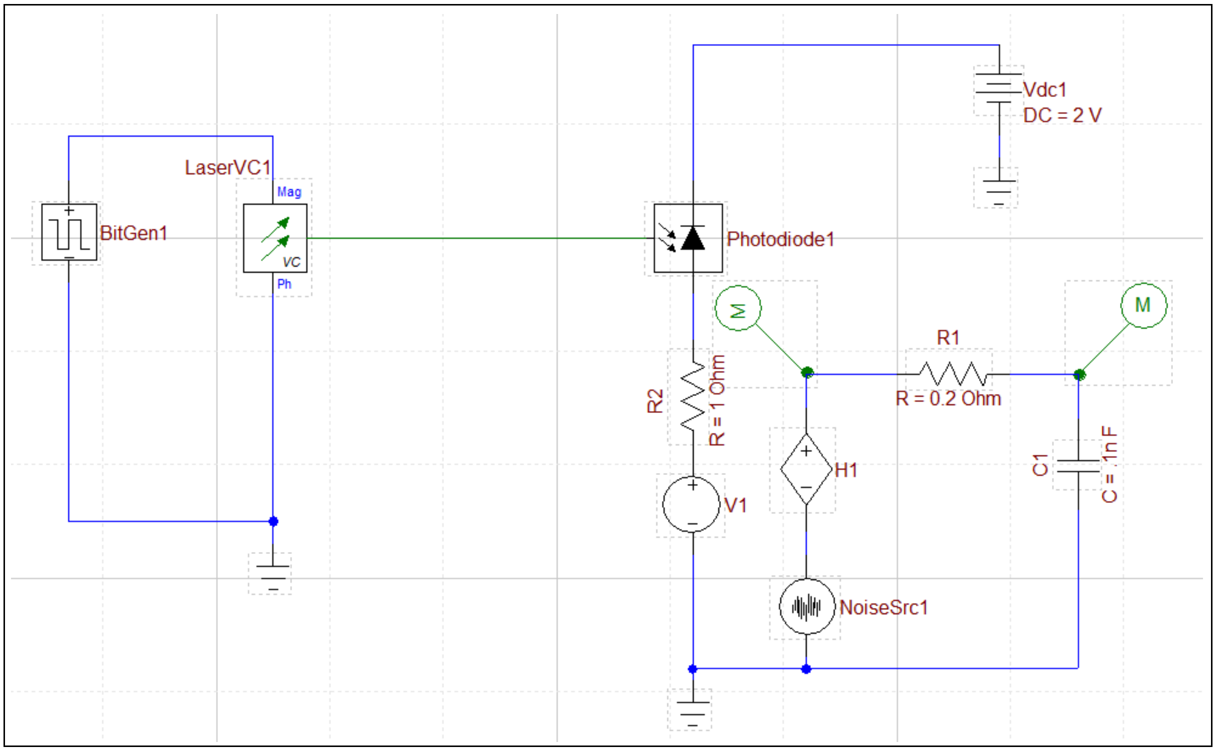
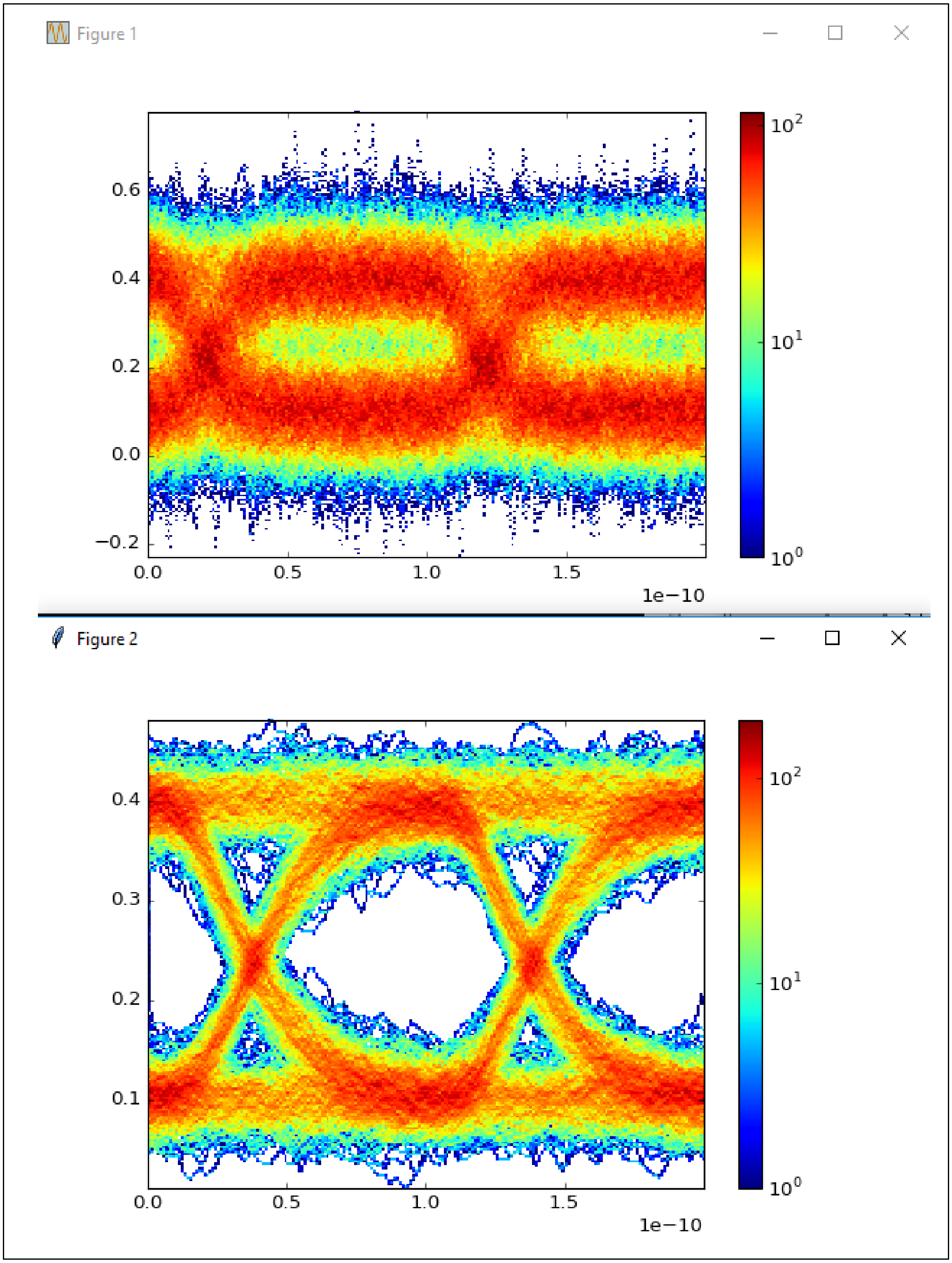


Figure 8 Eye histogram before (top) and after (below) the RC filter



Frequency domain analysis

The following design (Figure 9) shows the use of a micro-ring resonator as a channel filter. The bit generator directly modulates the magnitude of two lasers with different wavelengths. Later these signals with different wavelengths are combined together using a joiner and filtered through the ring resonator (see Figure 10). Python post processing allows the signals to be visualized either in time or frequency domain. The Python script, *RingChannelFilterFFT.py*, and associated OptiSPICE Schematic can be found in *OptiSPICE 5.2 Samples\Tutorials\Advanced\Python Post Processing\Ring Channel Filter*

Figure 9 Optical Micro-ring resonator as a channel filter

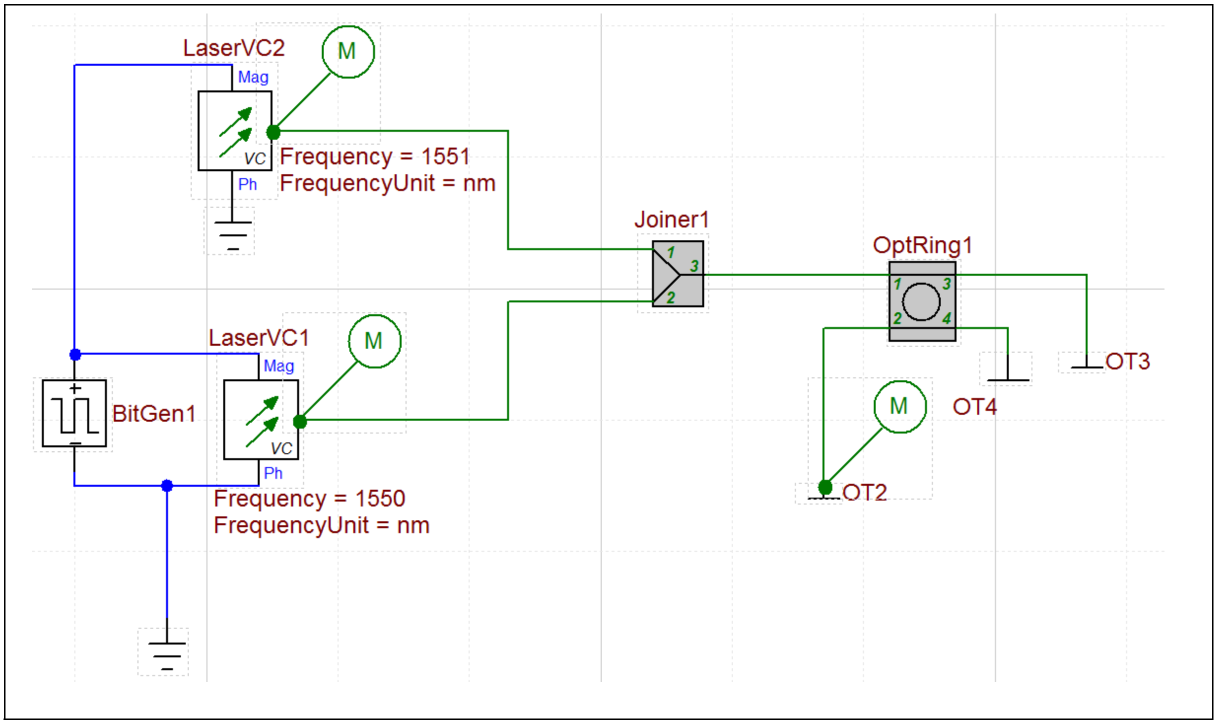
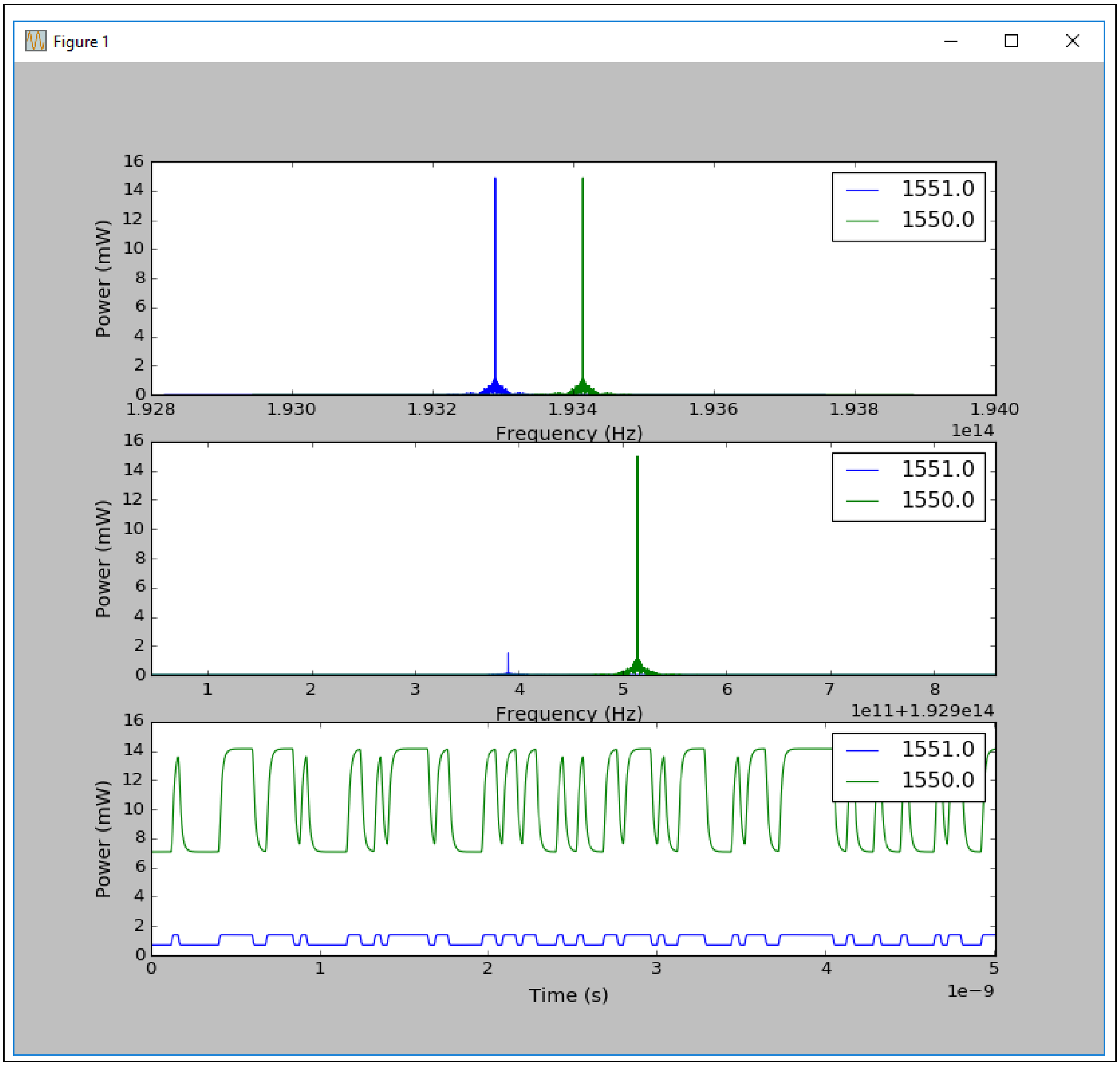


Figure 10 Input signals with different wavelengths (top) going through the ring resonator and the output of the ring resonator in frequency (middle) and time domain (bottom)



OptiSystem and OptiSPICE Co-simulation

It is possible to exchange data between OptiSPICE and OptiSystem during an OptiSystem simulation. Once the simulation ends the results can be viewed with the OptiSystem visualizer components.

OptiSPICE schematics can be designed to accept electrical and/or optical data from OptiSystem using the ***Electrical Input - Vsource***, ***Electrical Input - Isource*** or ***Optical Input*** elements. The probes placed in OptiSPICE are used to transfer data from OptiSPICE to OptiSystem. Once the inputs and outputs in OptiSPICE are defined (ElecInput_V1 and probes; see Figure 11), the schematic needs to be configured to run as a co-simulation. This can be done by using **OptiSystem > Configure Co-simulation** (see Figure 12)

Figure 11 OptiSPICE transceiver circuit

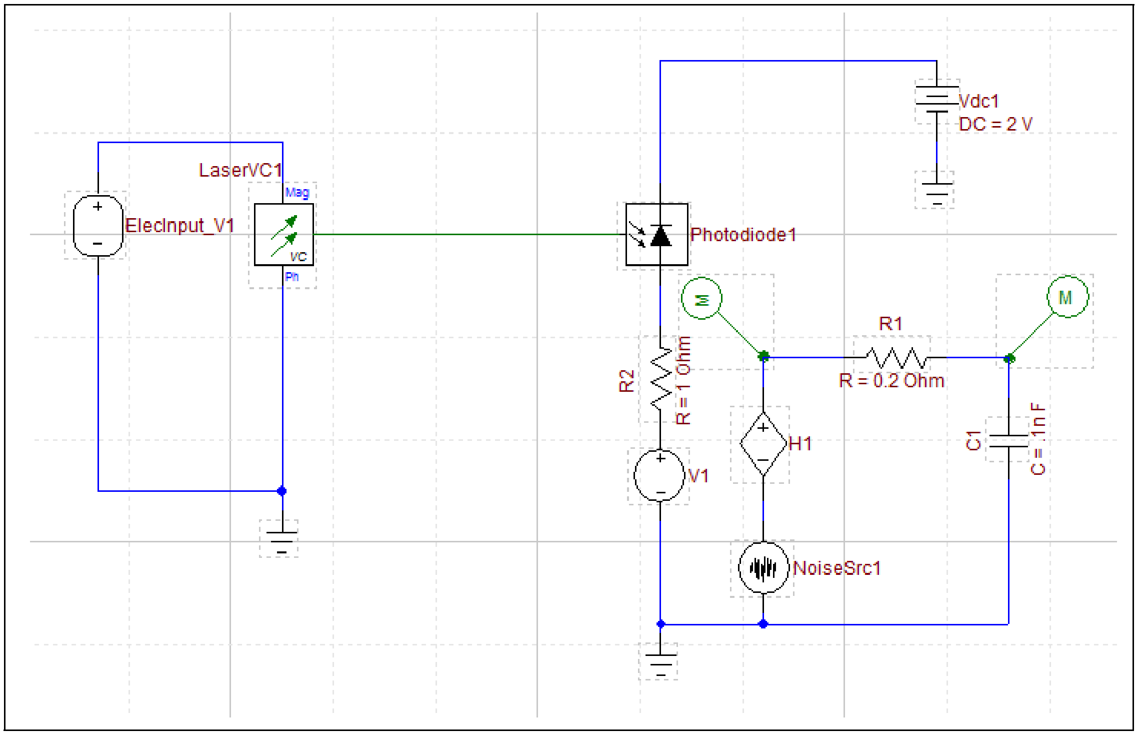
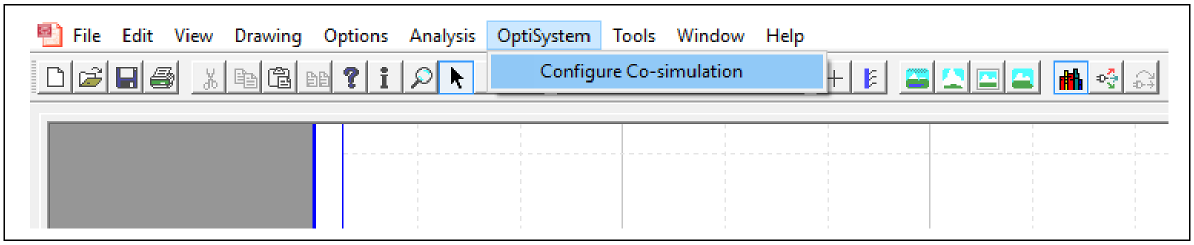
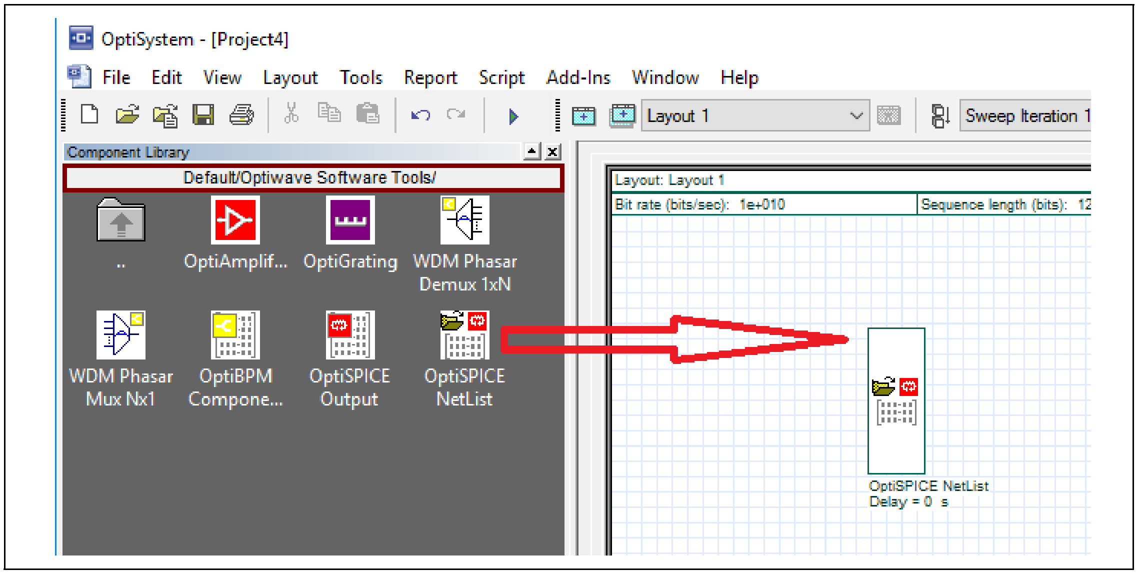


Figure 12 Configuring OptiSPICE for co-simulation



In OptiSystem, the OptiSPICE Netlist component can be found under ***Default/Optiwave Software Tools*** within the component library (see Figure 13). After the co-simulation has been configured in OptiSPICE, the OptiSPICE Netlist file needs to be linked to the OptiSPICE Netlist Component (see Figure 14).

Figure 13 Setting up OptiSPICE co-simulation in OptiSystem

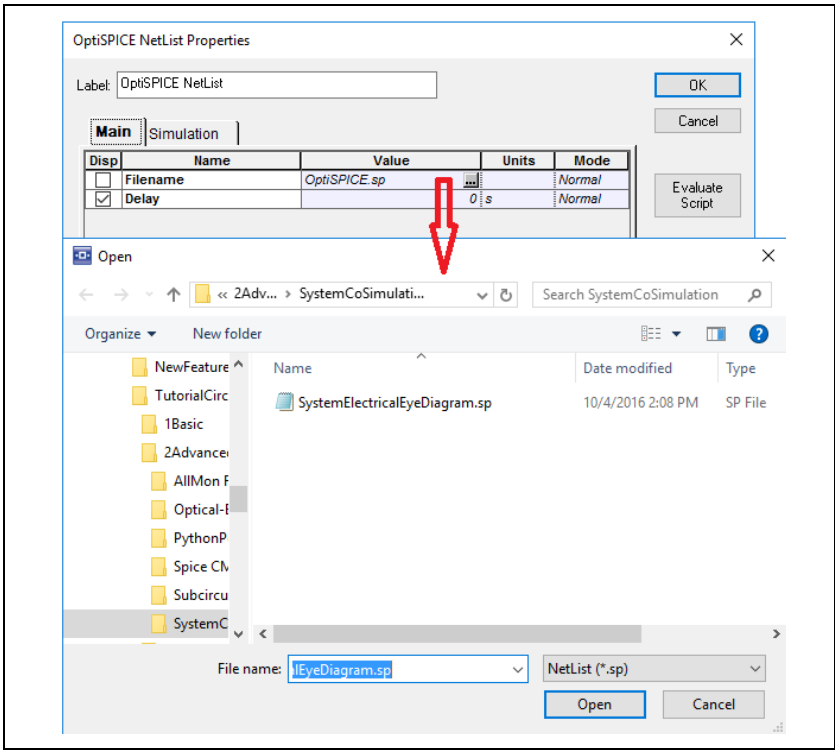


Bit rate (bits/sec): 1e+010

Sequence length (bits): 12

Delay = 0 s

Figure 14 Choosing a Netlist file for co-simulation



Label: OptiSPICE NetList

Disp	Name	Value	Units	Mode
□	Filename	OptiSPICE.sp		Normal
☑	Delay	0	s	Normal

Name	Date modified	Type
SystemElectricalEyeDiagram.sp	10/4/2016 2:08 PM	SP File

File name: lEyeDiagram.sp

NetList (*.sp)

After starting the OptiSystem simulation (Figure 15), the OptiSPICE Netlist receives the signal generated by OptiSystem. The OptiSPICE circuit simulation is then automatically initiated (a command line interface will appear providing a progress report on the simulation). After completion of the OptiSPICE simulation, OptiSystem then continues its simulation to completion. Various components in the visualizer library in OptiSystem such as the RF Spectrum Analyzer, Oscilloscope Visualizer, Optical Spectrum Analyzer can be used to analyze the data generated by OptiSPICE. In this example, the output from OptiSPICE is used to generate the eye diagrams shown in Figure 16. Before running the OptiSystem simulation it is important to check that the simulation time in OptiSystem matches the simulation time in OptiSPICE

Figure 15 OptiSystem co-simulation project

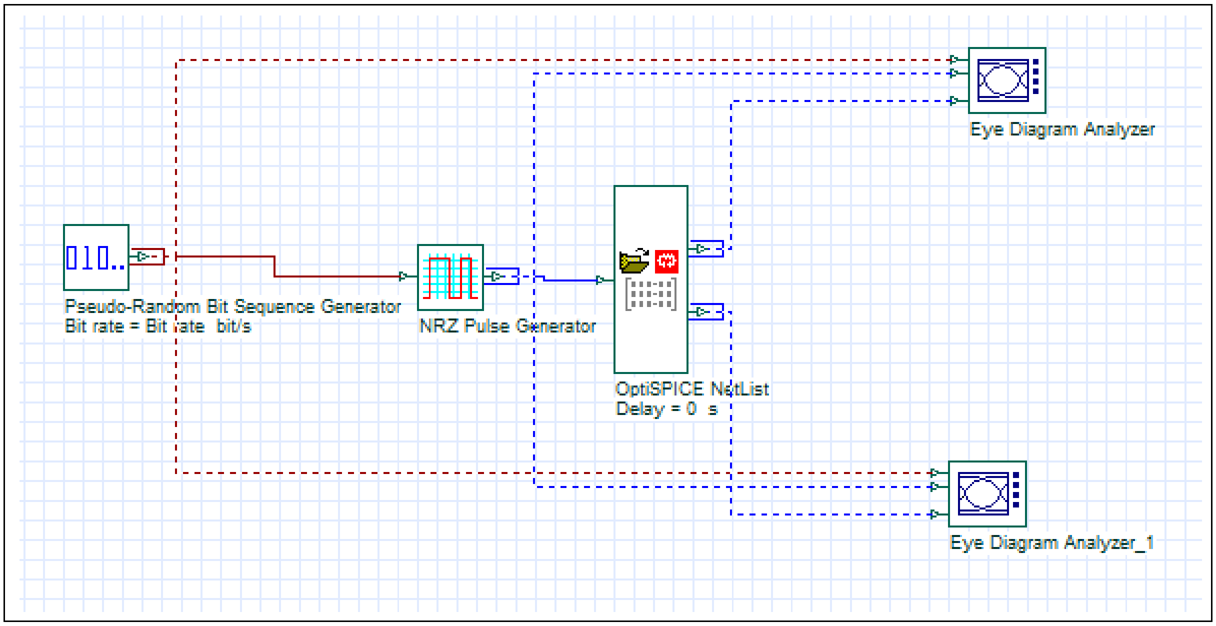
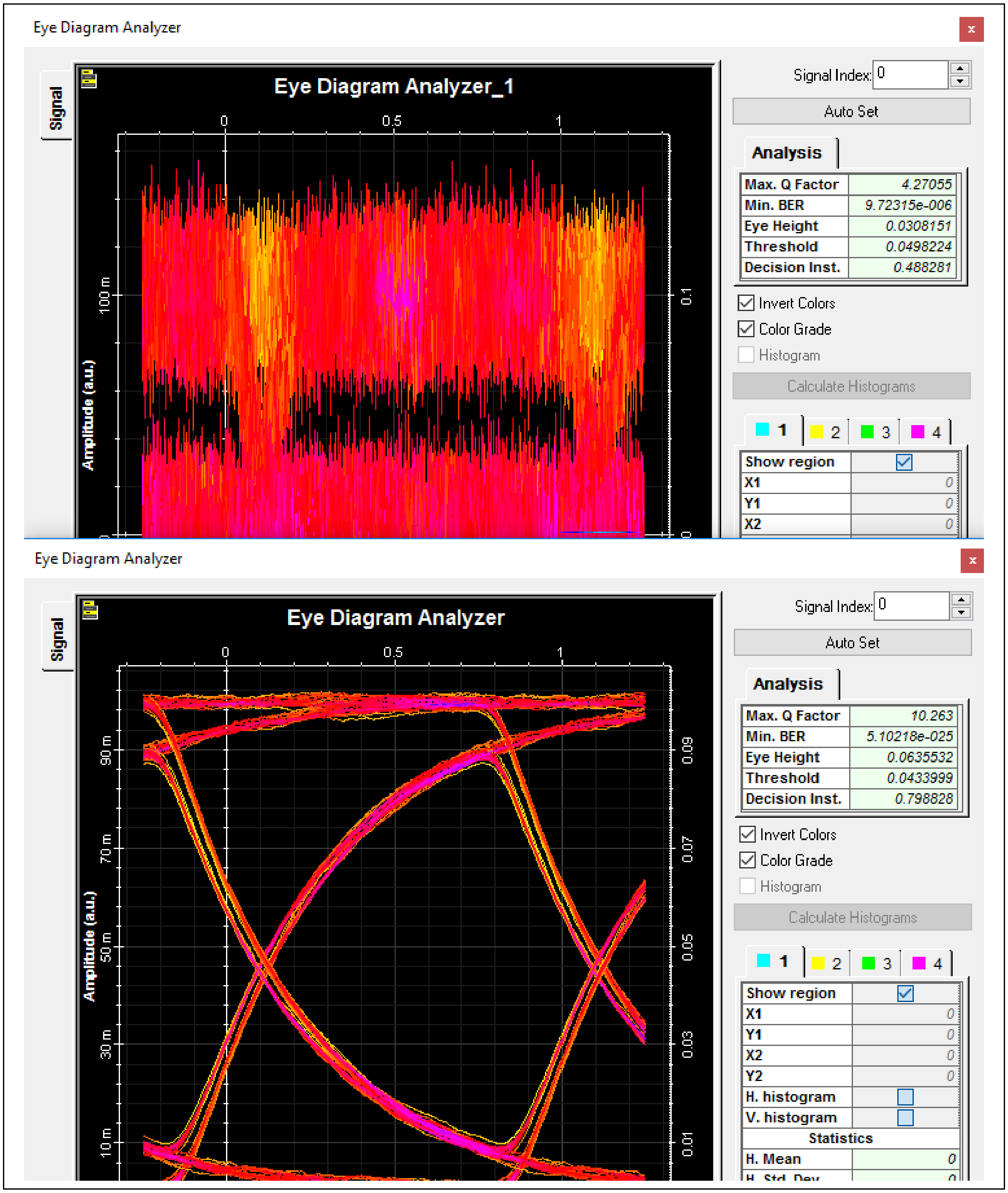


Figure 16 Eye diagrams generated from OptiSPICE Output



Signal Index: 0

Analysis	
Max. Q Factor	4.27055
Min. BER	9.72315e-006
Eye Height	0.0308151
Threshold	0.0498224
Decision Inst.	0.488281

☑ Invert Colors
☑ Color Grade
☐ Histogram

1 2 3 4

Show region	☑
X1	0
Y1	0
X2	0

Signal Index: 0

Analysis	
Max. Q Factor	10.263
Min. BER	5.10218e-025
Eye Height	0.0635532
Threshold	0.0433999
Decision Inst.	0.798828

☑ Invert Colors
☑ Color Grade
☐ Histogram

1 2 3 4

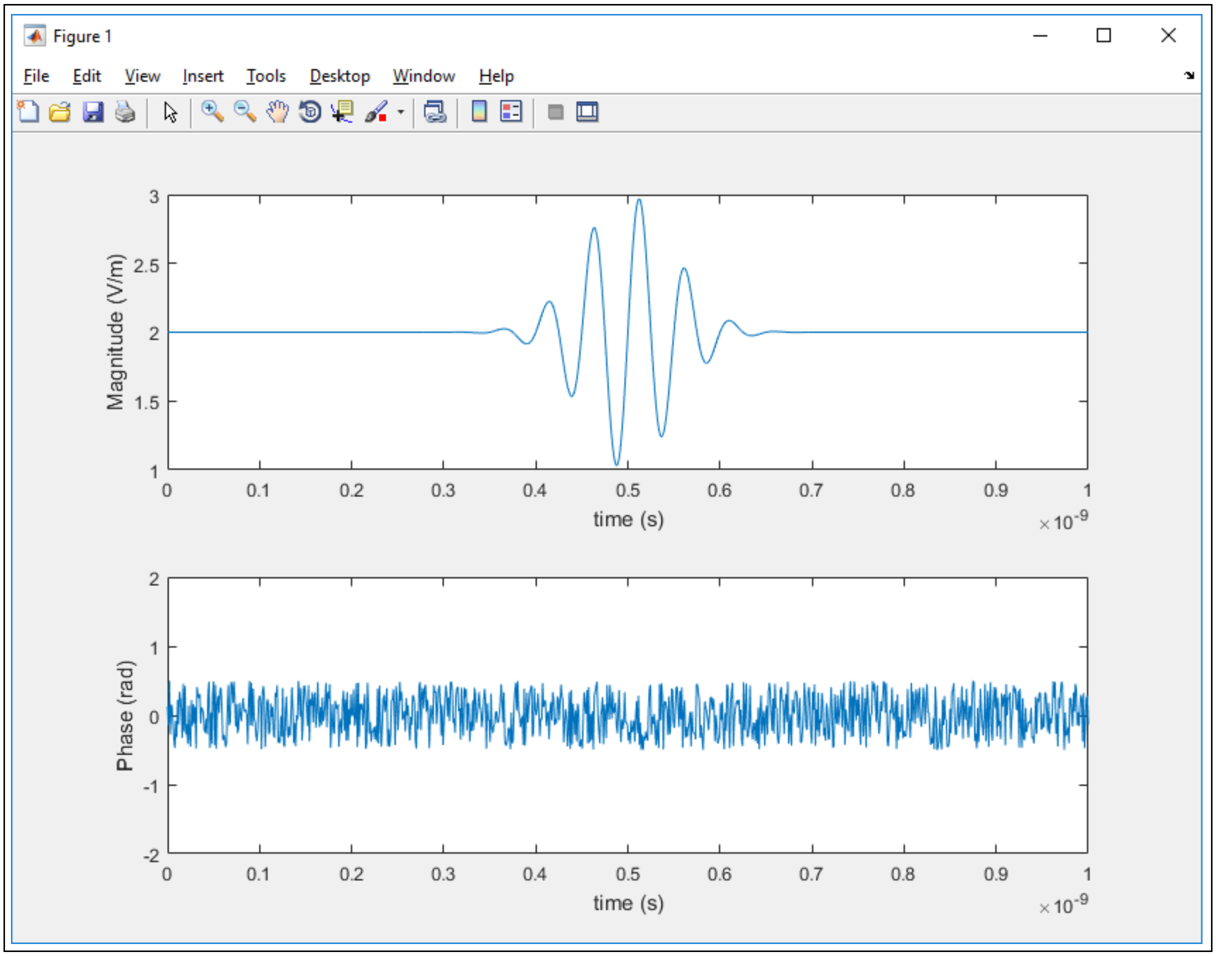
Show region	☑
X1	0
Y1	0
X2	0
Y2	0
H. histogram	☐
V. histogram	☐
Statistics	
H. Mean	0
H. Std. Dev.	0

Optical/Electrical Signal File Input

OptiSPICE is able to run simulations with user defined inputs using ***Ipwl - File input*** (current) and ***Vpwl-File input*** (voltage) elements which require a text file with two columns: time and current/voltage. The following MATLAB code is used to generate magnitude and phase text files which will be used to drive a voltage controlled laser. (Figure 17 shows the plot of the generated signals).

```
close all
clear all
ni = 1e3;

t = linspace(0,1e-9,ni);

f=20e9;
mag = sin(2*pi*f*t).*exp(-(t-0.5e-9).^2/0.5e-20)+2;
magnitude = [t' mag'];

ph = 1*(rand(1,ni)-0.5);
phase = [t' ph'];

subplot(2,1,1);plot(t,mag);
ylabel('Magnitude (V/m)')
xlabel('time (s)')
subplot(2,1,2);plot(t,ph);
ylim([-2 2])
ylabel('Phase (rad)')
xlabel('time (s)')

dlmwrite('magnitude.txt',magnitude,' ')
dlmwrite('phase.txt',phase,' ')
```

Figure 17 Magnitude and phase data file generation (MATLAB)



The circuit in Figure 18 uses two separate text files to control the magnitude and the phase of the laser through ***Vpwl-File input*** voltage sources. The generated text file needs to be selected in the parameter menu for each source (Figure 19).

Figure 18 Using Vpwl_File elements for magnitude and phase input from a text file

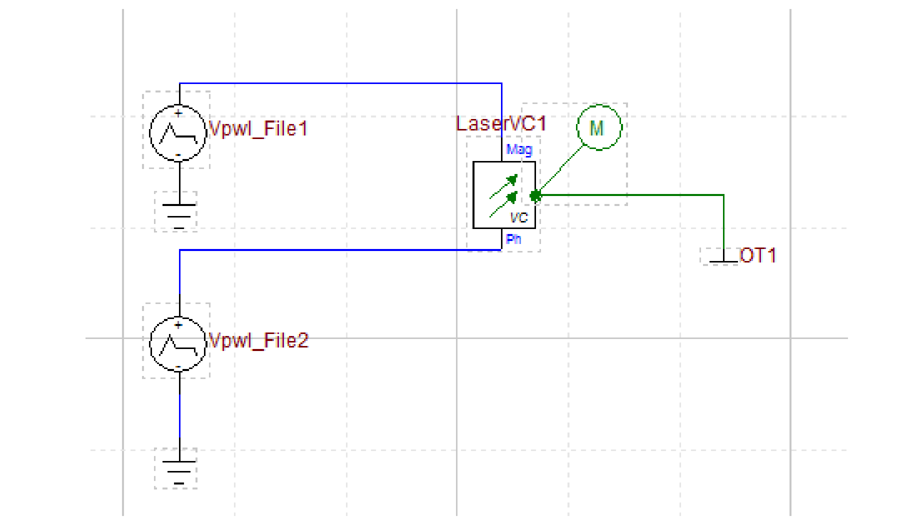
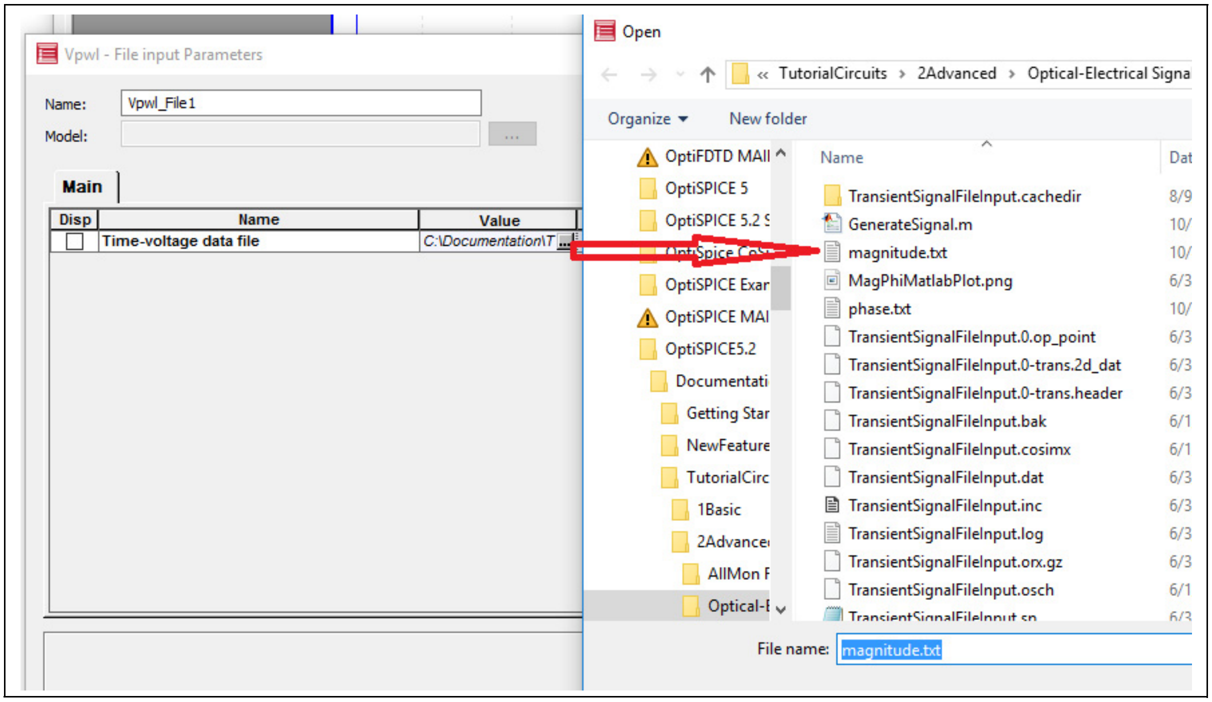


Figure 19 Selecting an input text file



Vpwl - File input Parameters

Name: Vpwl_File1

Model:

Main

Disp	Name	Value
□	Time-voltage data file	C:\Documentation\T

Open

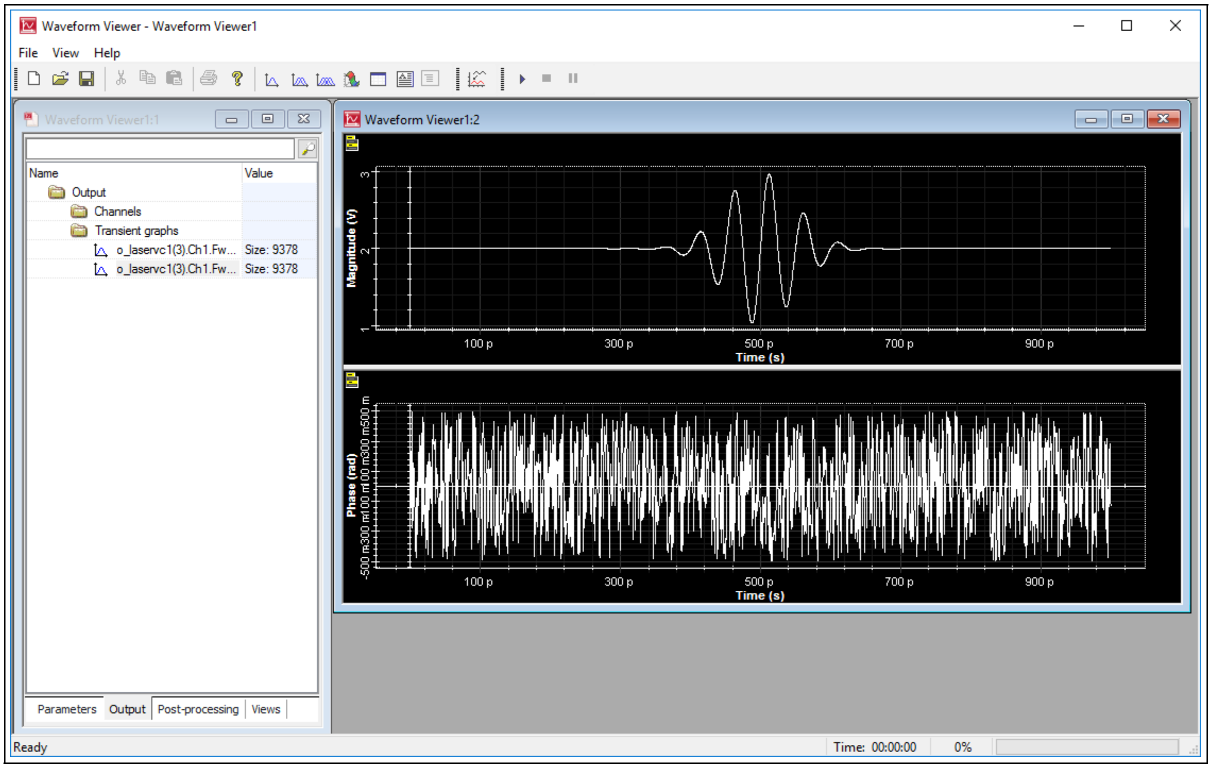
« TutorialCircuits › 2Advanced › Optical-Electrical Signa

Name	Dat
TransientSignalFileInput.cachedir	8/9
GenerateSignal.m	10/
magnitude.txt	10/
MagPhiMatlabPlot.png	6/3
phase.txt	10/
TransientSignalFileInput.0.op_point	6/3
TransientSignalFileInput.0-trans.2d_dat	6/3
TransientSignalFileInput.0-trans.header	6/3
TransientSignalFileInput.bak	6/1
TransientSignalFileInput.cosimx	6/1
TransientSignalFileInput.dat	6/3
TransientSignalFileInput.inc	6/3
TransientSignalFileInput.log	6/3
TransientSignalFileInput.orx.gz	6/3
TransientSignalFileInput.osch	6/1
TransientSignalFileInput.sp	6/3

File name: magnitude.txt

Figure 20 shows the output of LaserVC1 which matches the plots generated in MATLAB.

Figure 20 OptiSPICE simulation results

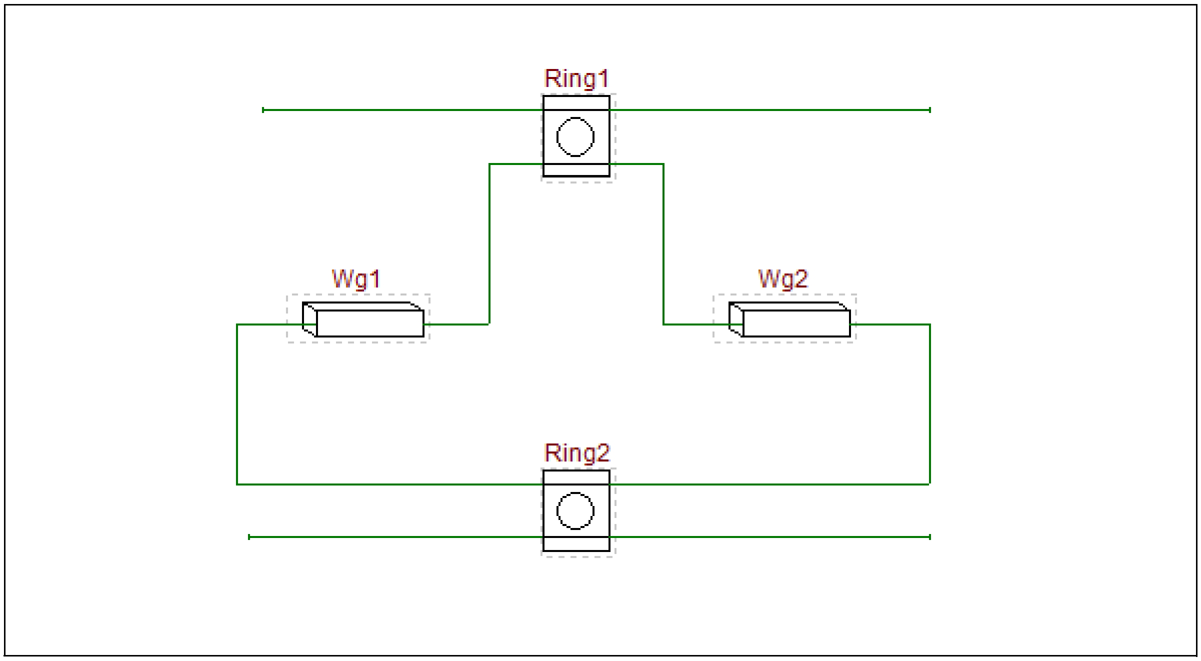


Name	Value
Output	
Channels	
Transient graphs	
o_laservc1(3).Ch1.Fw...	Size: 9378
o_laservc1(3).Ch1.Fw...	Size: 9378

Creating Sub-circuits

In the OptiSPICE Schematic Editor it is possible to create sub-circuits using any combination of components. This tutorial shows how the optical components of an optical Chebyshev filter (Figure 21), located in *OptiSPICE 5.2 Samples\Tutorials\Advanced\Subcircuit\Optical Chebyshev Start Circuit*, can be combined to create a sub-circuit element.

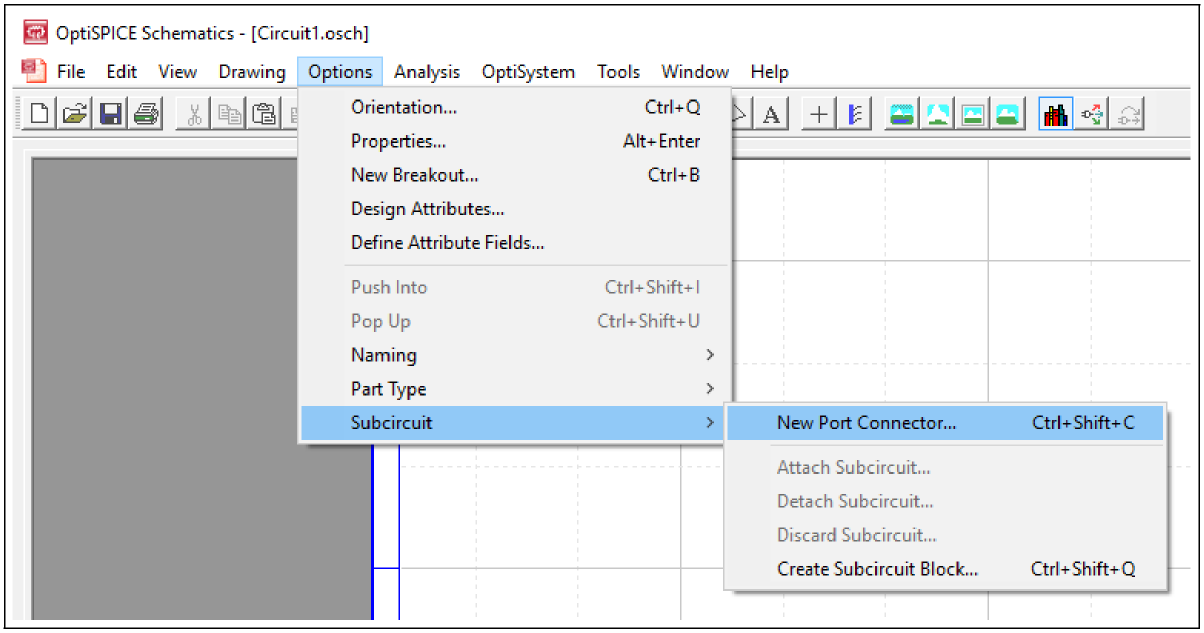
Figure 21 Optical Chebyshev filter



Perform the following steps to build a sub-circuit based on the Optical Chebyshev filter circuit design

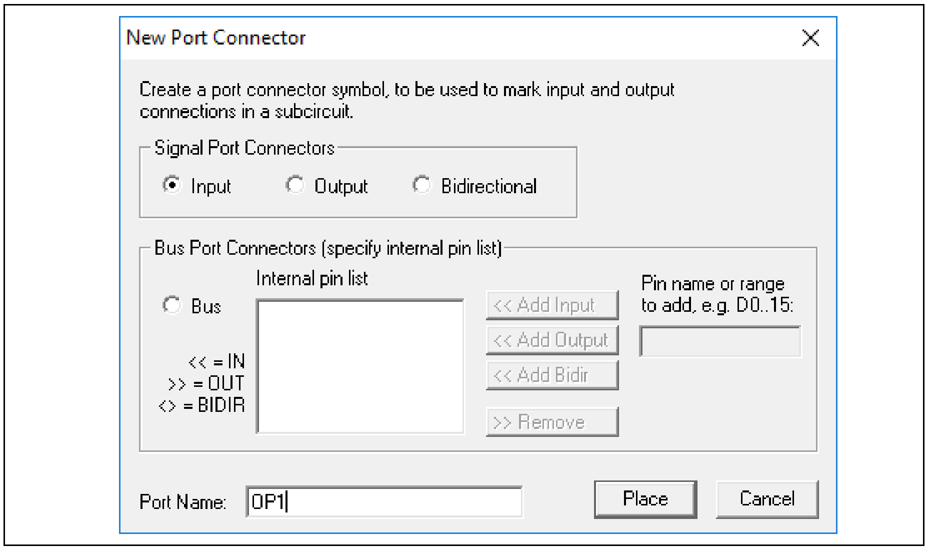
Step	Action
1	To add ports to the schematic go to **Options>Subcircuit>New Port Connector** (see Figure 22)

Figure 22 Adding port connectors



Step	Action
2	Name the port as “OP1” and click the Place button (Figure 23).

Figure 23 Naming a new port



Create a port connector symbol, to be used to mark input and output connections in a subcircuit.

● Input

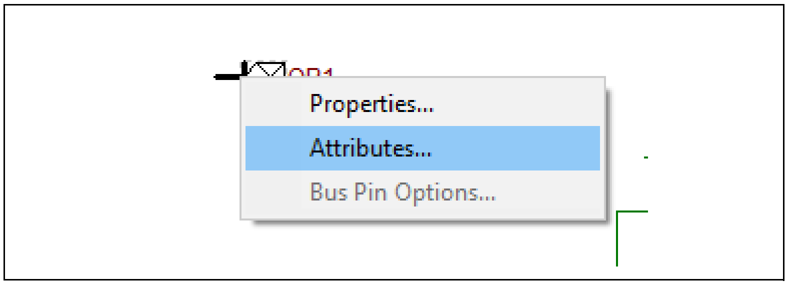
Port Name: OP1

3 Once the port is placed on the canvas, right click on the Pin and select ***Attributes*** (Figure 24).

Steps 3 and 4 are only required for optical ports since all the pins are electrical by default

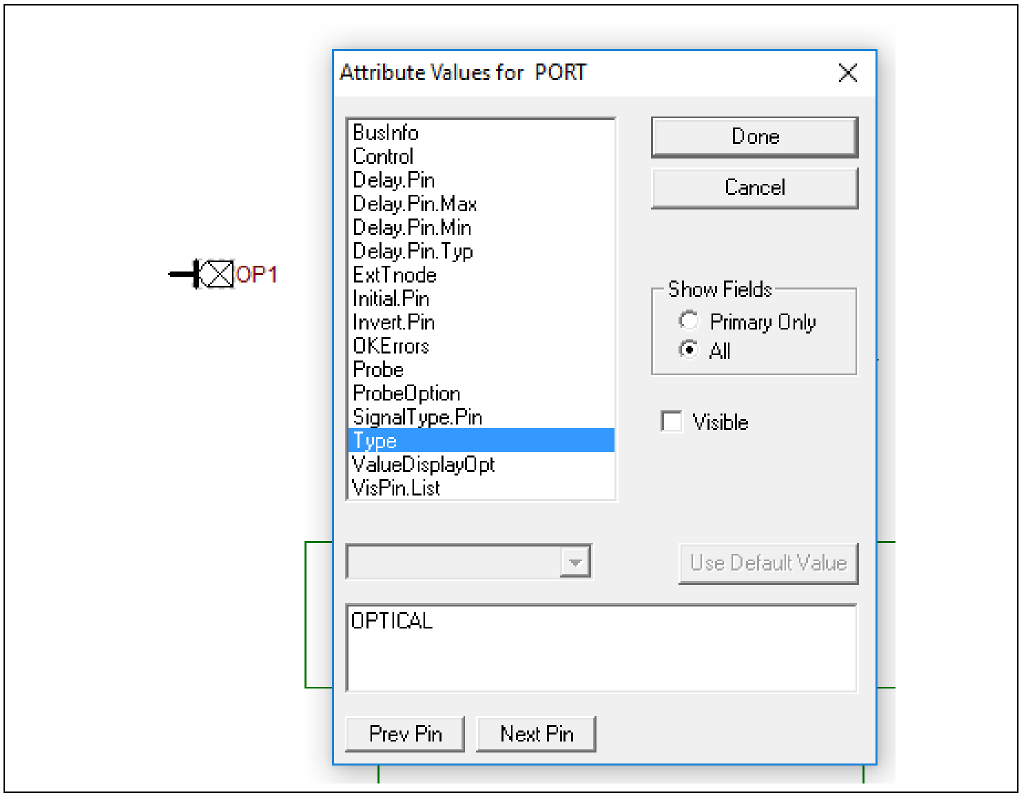
Note: It is important to select the Pin (not the element). When the Pin is selected, a black T-shaped icon will appear.

Figure 24 Pin attributes



4 **Click** on the parameter ***Type*** and write OPTICAL in the text box below (Figure 25) and **select** Done.

Figure 25 Type value for Port

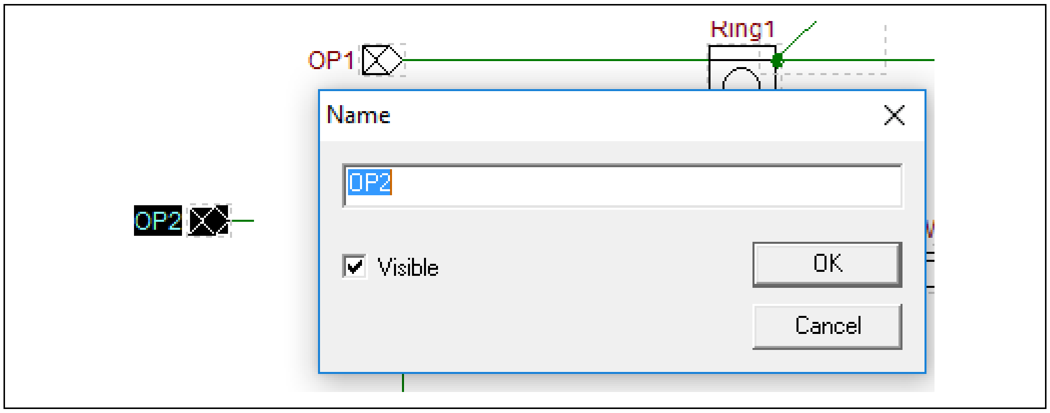


OPTICAL

5 Repeat steps 1-4 for the other optical ports. To do so, simply copy and paste the existing “OP1” pin and re-name the port connectors to “OP2”, “OP3” and

"OP4" (to rename the port connector select the component name field, left double-click, and update the name field - see Figure 26)

Figure 26 Re-naming of the optical port connector



Name

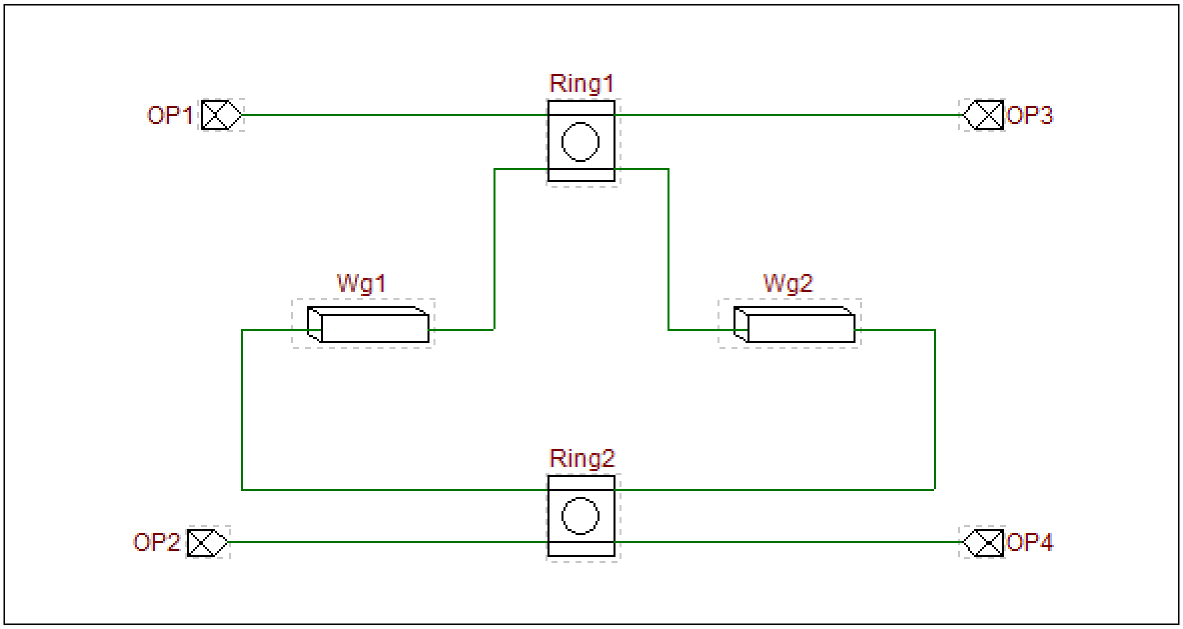
OP2

☑ Visible

6 Join the port connectors to the ring resonator ports as shown in Figure 27

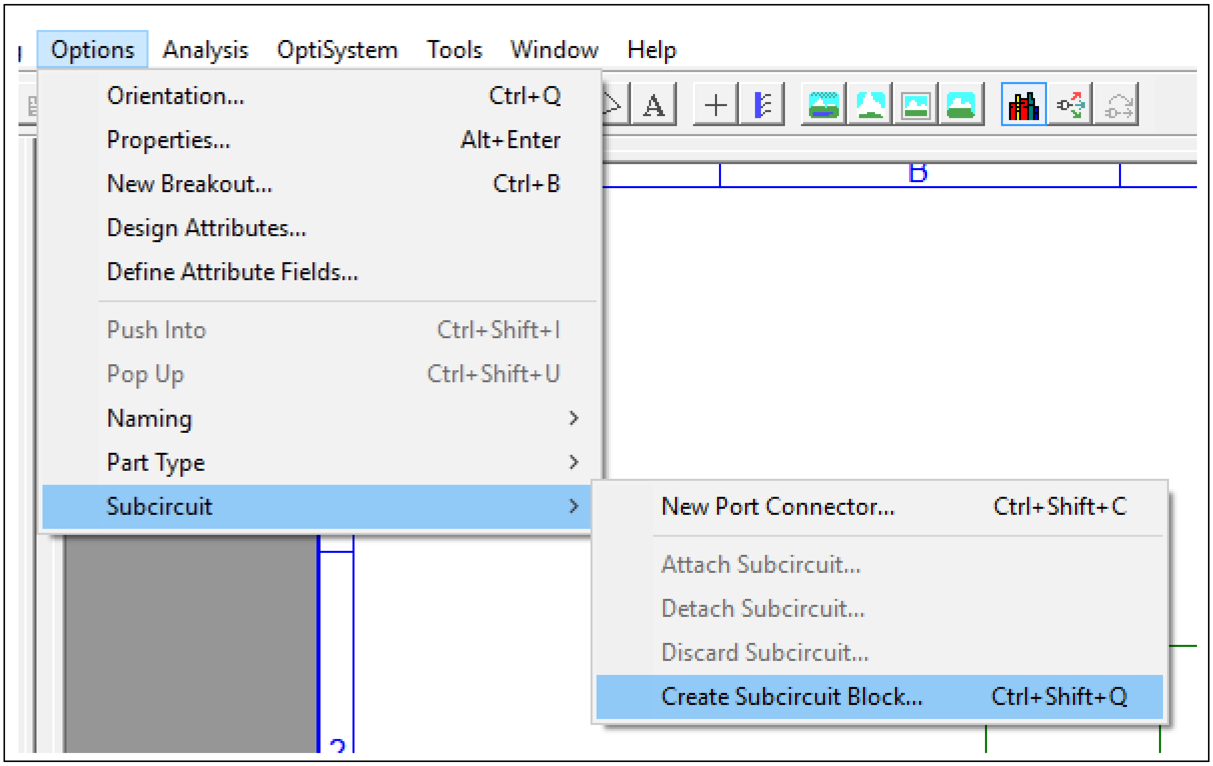
Note: To correctly orientate port connectors OP3 and OP4, select the element and then CTRL-Y, or alternatively right-click and select "Flip Horizontal".

Figure 27 Placement of the optical port connectors



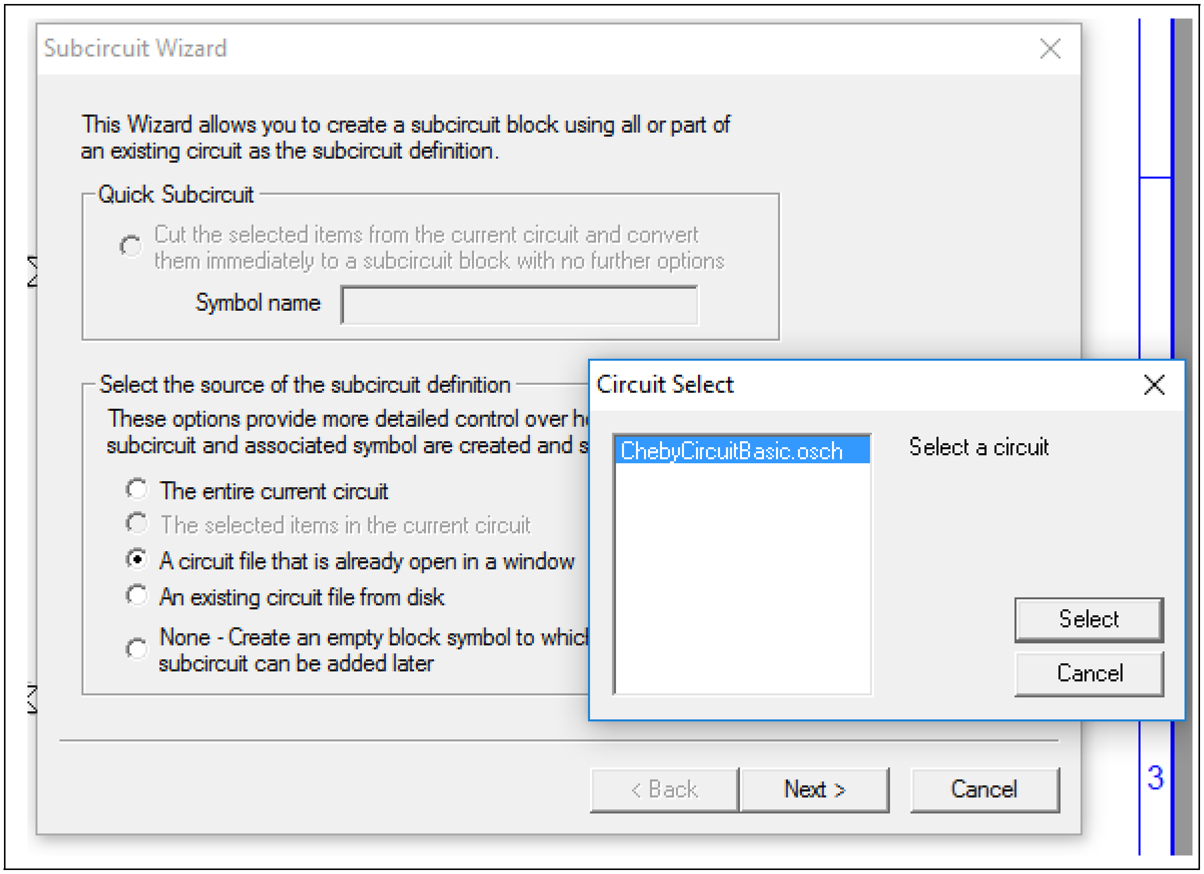
7 Click on **Options>Subcircuit>Create Subcircuit Block** (Figure 28).

Figure 28 Create Subcircuit block



8 Select the schematic for the subcircuit (ensure the option “*A circuit file that is already open in a window”* is selected) and click on Select (Figure 29).

Figure 29 Subcircuit Wizard



Subcircuit Wizard

This Wizard allows you to create a subcircuit block using all or part of an existing circuit as the subcircuit definition.

Quick Subcircuit

○ Cut the selected items from the current circuit and convert them immediately to a subcircuit block with no further options

Symbol name

Select the source of the subcircuit definition

These options provide more detailed control over h subcircuit and associated symbol are created and s

○ The entire current circuit

○ The selected items in the current circuit

● A circuit file that is already open in a window

○ An existing circuit file from disk

○ None - Create an empty block symbol to whicl subcircuit can be added later

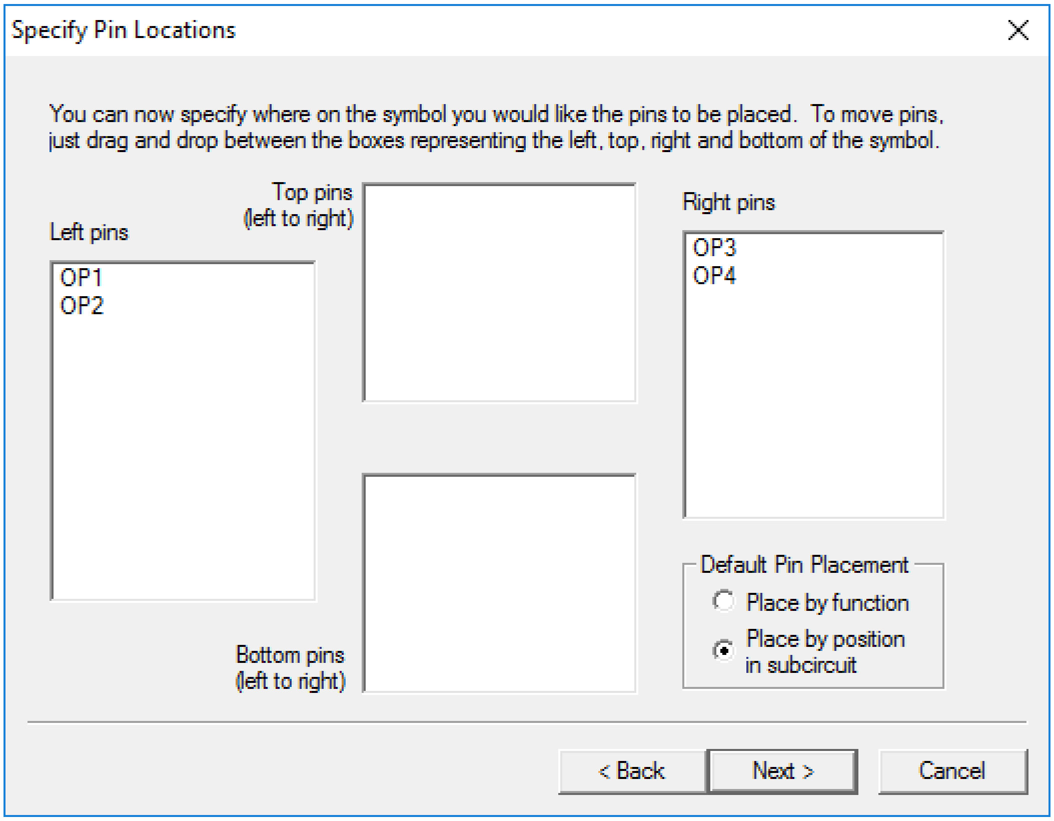
Circuit Select

ChebyCircuitBasic.osch

Select a circuit

9 Choose the location where the ports will appear on the sub-circuit - left, top, right and bottom (for this example we will leave the pin placements as is) and select “Next” (see Figure 30)

Figure 30 Specify pin locations



Specify Pin Locations

You can now specify where on the symbol you would like the pins to be placed. To move pins, just drag and drop between the boxes representing the left, top, right and bottom of the symbol.

Left pins: OP1, OP2

Top pins (left to right):

Right pins: OP3, OP4

Bottom pins (left to right):

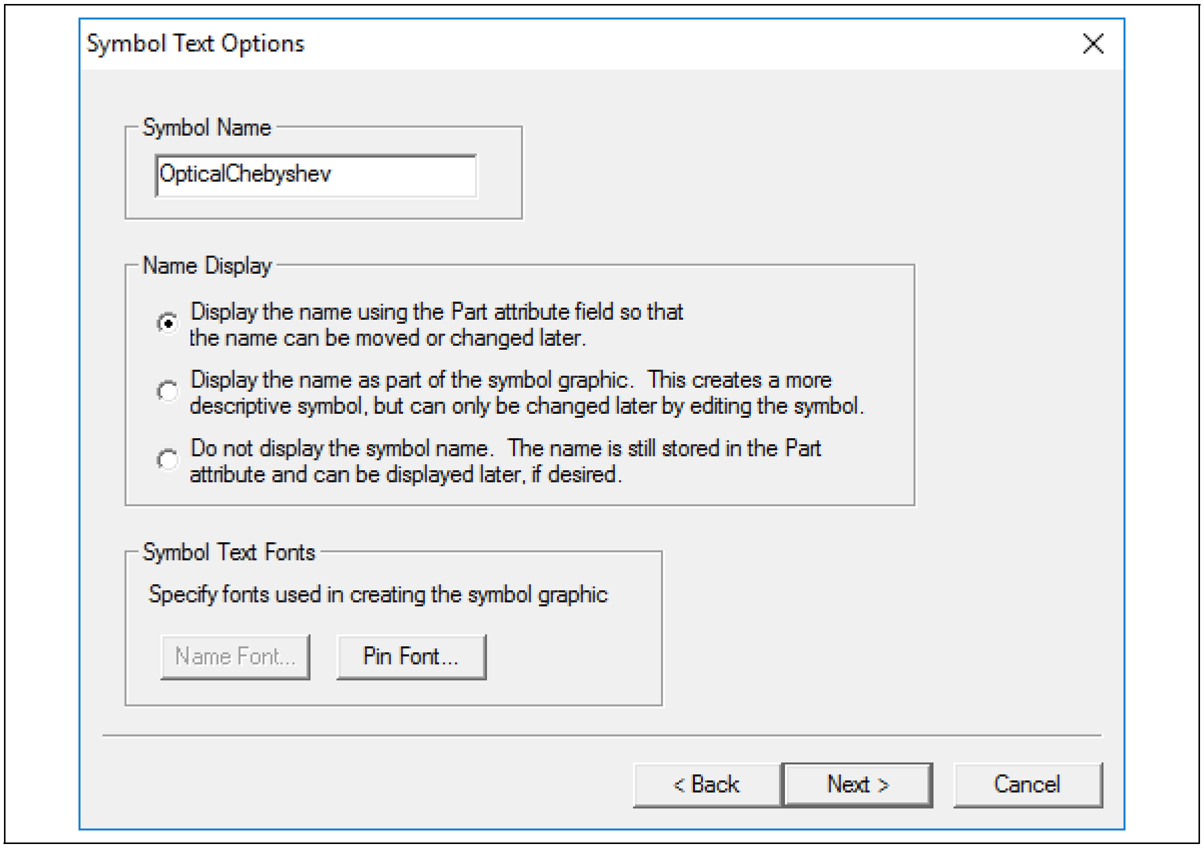
Default Pin Placement

○ Place by function

● Place by position in subcircuit

10 Enter a name for the Subcircuit symbol (“OpticalChebyshev”) under the ***Symbol Name*** field and select ***Next***.

Figure 31 Symbol text options



Symbol Text Options

Symbol Name: OpticalChebyshev

Name Display

- ● Display the name using the Part attribute field so that the name can be moved or changed later.
- ○ Display the name as part of the symbol graphic. This creates a more descriptive symbol, but can only be changed later by editing the symbol.
- ○ Do not display the symbol name. The name is still stored in the Part attribute and can be displayed later, if desired.

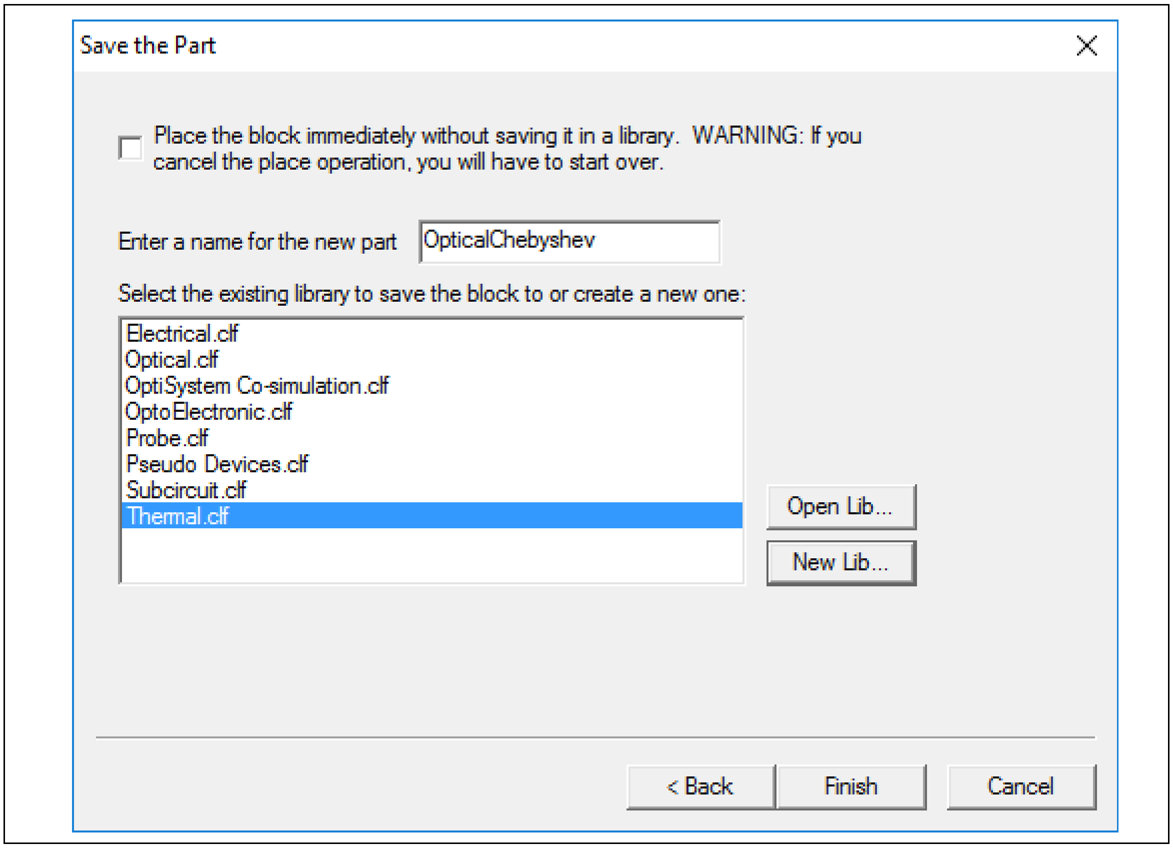
Symbol Text Fonts

Specify fonts used in creating the symbol graphic

You will now have the option to place the subcircuit block directly on to the design canvas (without saving it to a library) or to save it to an existing or new library so that it can be re-used for other projects or designs. In the next few steps we will save the subcircuit in a new library

11 Remove the check-box next to the option *“Place the block immediately without saving it in a library”* and select the button ***New Lib*** (see Figure 32)

Figure 32 Saving the subcircuit to a library



Save the Part

□ Place the block immediately without saving it in a library. WARNING: If you cancel the place operation, you will have to start over.

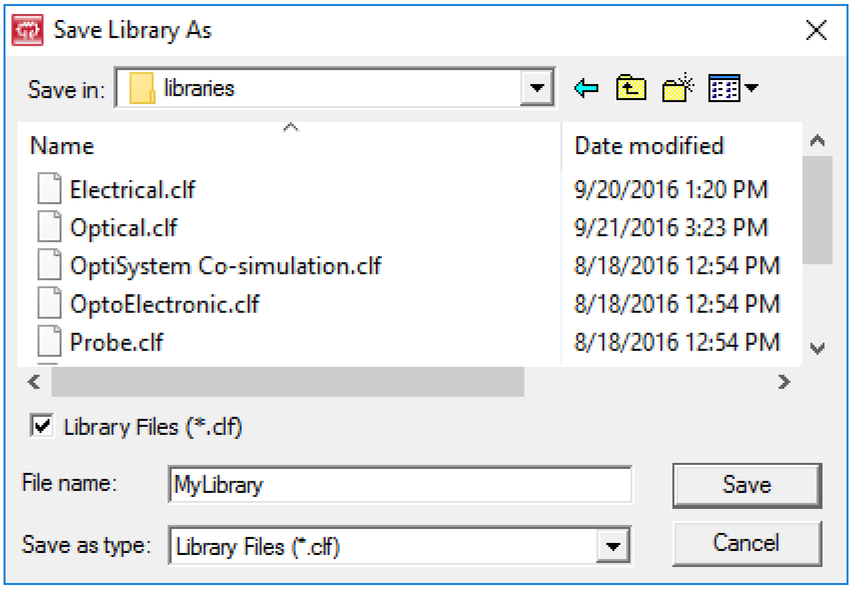
Enter a name for the new part: OpticalChebyshev

Select the existing library to save the block to or create a new one:

- Electrical.clf
- Optical.clf
- OptiSystem Co-simulation.clf
- OptoElectronic.clf
- Probe.clf
- Pseudo Devices.clf
- Subcircuit.clf
- Thermal.clf

12 Select the location where you would like to save the library and type “MyLibrary” within the ***File name*** field

Figure 33 Saving the subcircuit to a library



Save in: libraries

Name	Date modified
Electrical.clf	9/20/2016 1:20 PM
Optical.clf	9/21/2016 3:23 PM
OptiSystem Co-simulation.clf	8/18/2016 12:54 PM
OptoElectronic.clf	8/18/2016 12:54 PM
Probe.clf	8/18/2016 12:54 PM

☑ Library Files (*.clf)

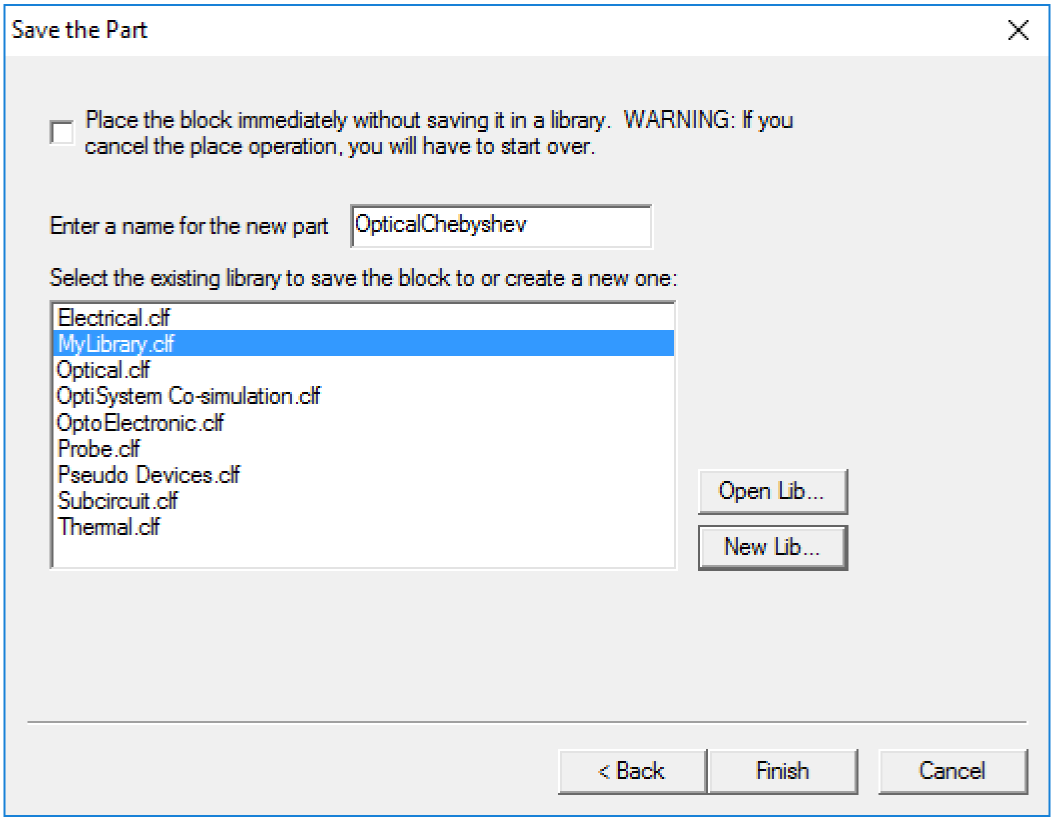
File name: MyLibrary

Save as type: Library Files (*.clf)

13 Click ***Save***

The new library "MyLibray.clf" will appear in the library list (Figure 34)

Figure 34 Save the Part menu



Save the Part

□ Place the block immediately without saving it in a library. WARNING: If you cancel the place operation, you will have to start over.

Enter a name for the new part: OpticalChebyshev

Select the existing library to save the block to or create a new one:

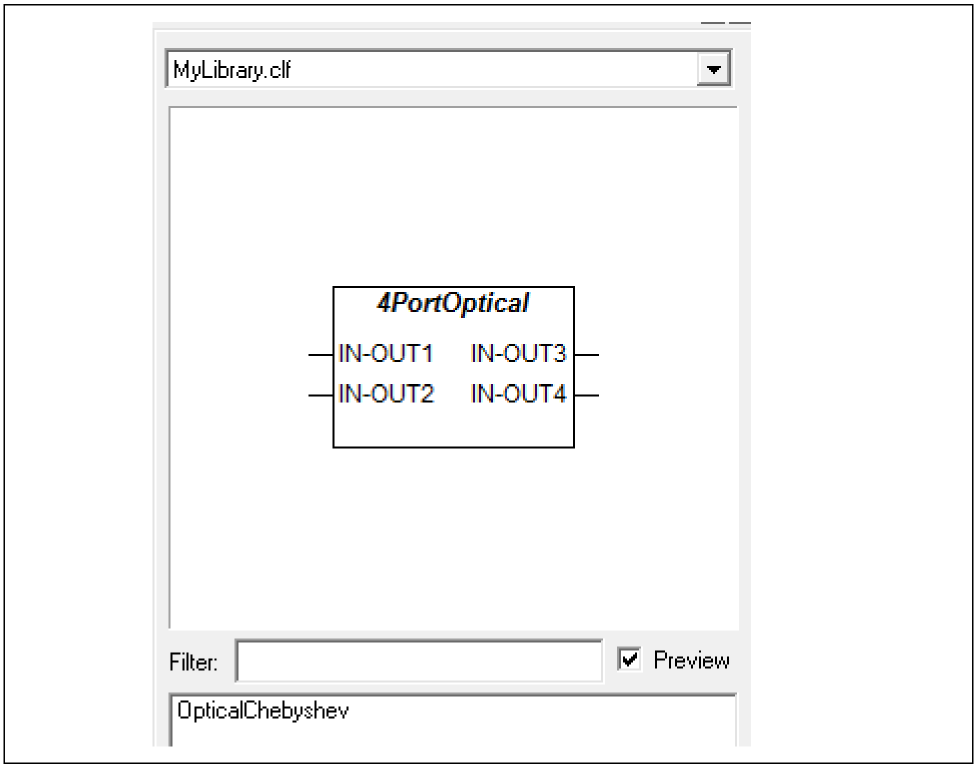
- Electrical.clf
- MyLibrary.clf
- Optical.clf
- OptiSystem Co-simulation.clf
- OptoElectronic.clf
- Probe.clf
- Pseudo Devices.clf
- Subcircuit.clf
- Thermal.clf

14 Click ***Finish***

The library should appear on the parts palette

15 Select “MyLibrary” from the top of the parts palette (Figure 35)

Figure 35 Parts palette



MyLibrary.clf

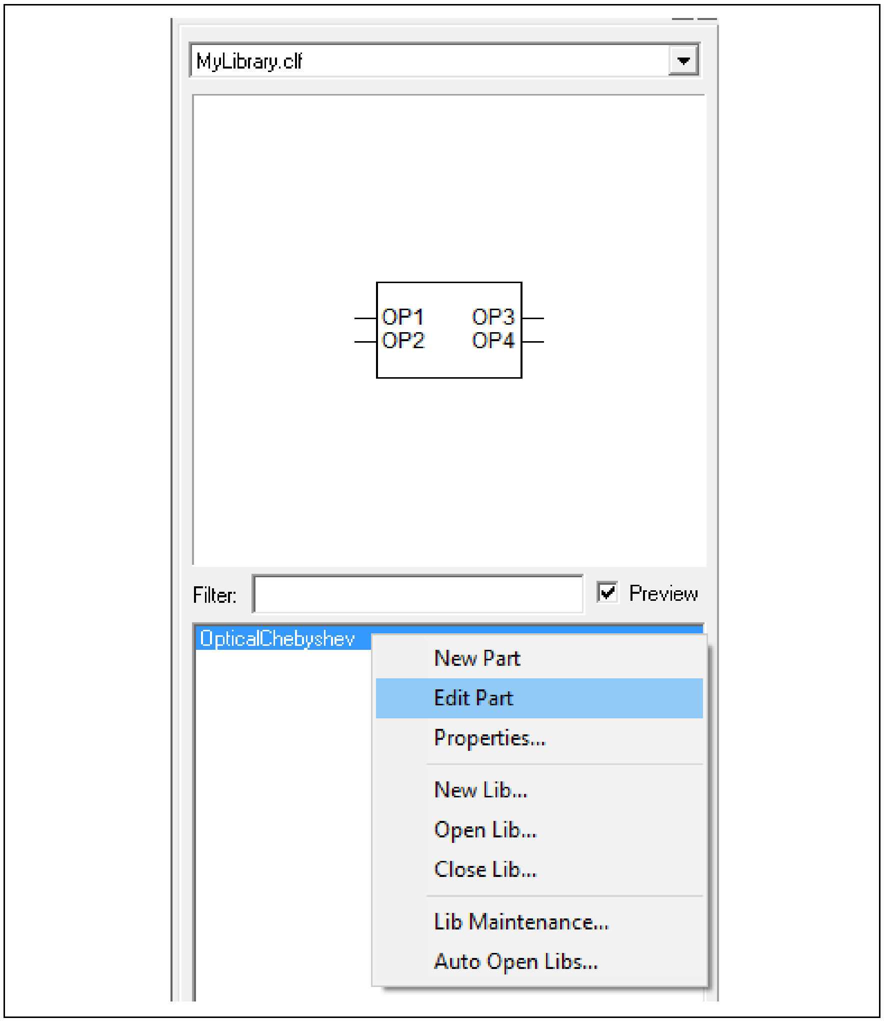
Filter:

☑ Preview

OpticalChebyshev

16 The pins of the subcircuit are set to electrical by default. In order to change the pin type, right click on the subcircuit in the parts palette and select ***Edit Part*** (Figure 36)

Figure 36 Edit part



MyLibrary.clf

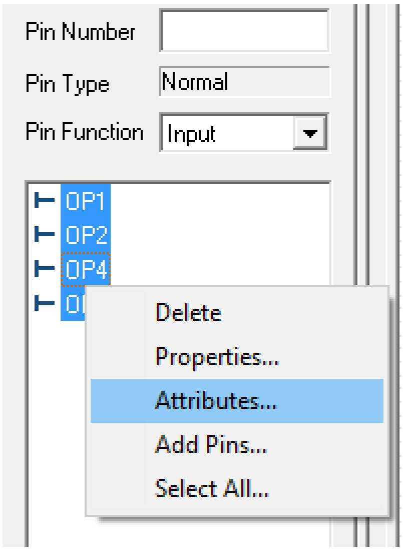
Filter:

☑ Preview

OpticalChebyshev

17 On the left pane of the Edit Part Schematic screen you will see the list of pins associated with the subcircuit. Select all the pins (CTRL and left-click on each pin icon), right-click and select ***Attributes***.

Figure 37 Changing Pin attributes



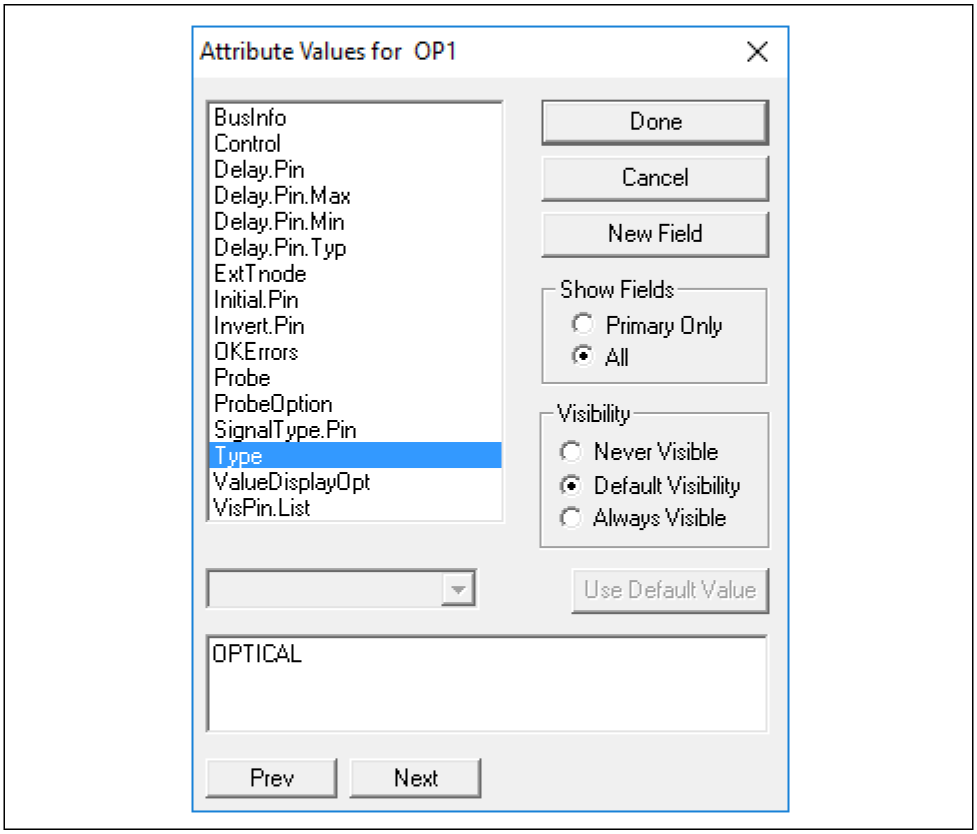
Pin Number

Pin Type: Normal

Pin Function: Input

18 Click on type and write “OPTICAL” in the text box. Click ***Next*** and repeat this step for all the pins. Click ***Done*** when finished (Figure 38)

Figure 38 Updating the value “Type” (Port Connector attributes)



Attribute Values for OP1

BusInfo
Control
Delay.Pin
Delay.Pin.Max
Delay.Pin.Min
Delay.Pin.Typ
ExtTnode
Initial.Pin
Invert.Pin
OKErrors
Probe
ProbeOption
SignalType.Pin
Type
ValueDisplayOpt
VisPin.List

Show Fields: ○ Primary Only ● All

Visibility: ○ Never Visible ● Default Visibility ○ Always Visible

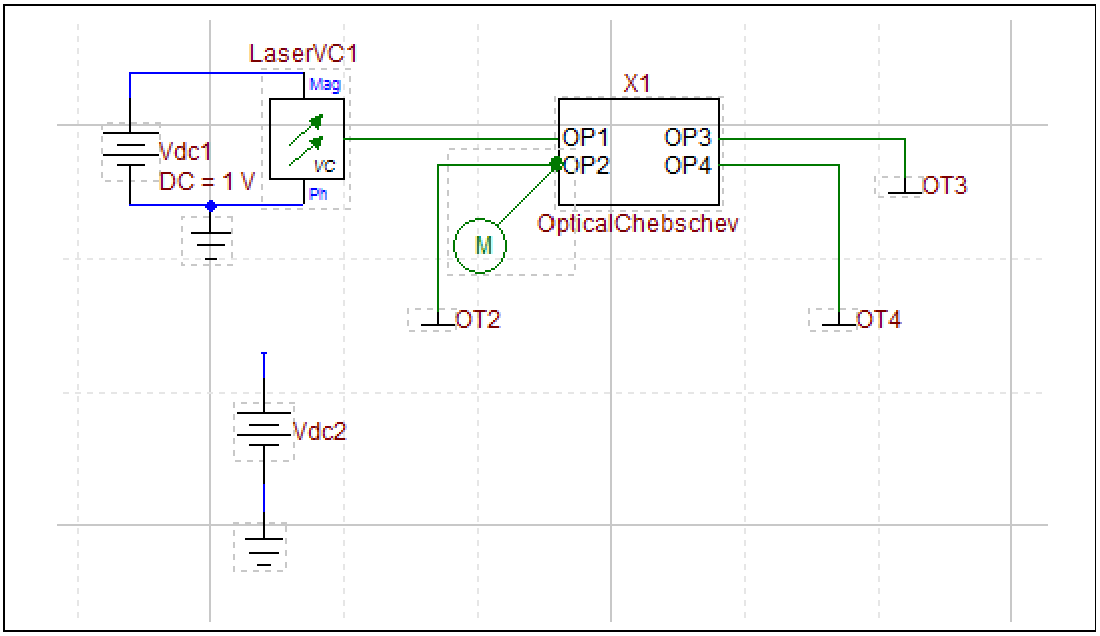
OPTICAL

19 Close the *Parts Editor* (make sure to select ***Yes*** when prompted to save the part) and close (and save) the schematic.

20 Create a new design and build the circuit shown in Figure 39 using the Chebyshev subcircuit. Setup the simulation as a DC sweep (where Vdc2 is used to sweep the frequency of LaserVC1).

For further information on how to set up parameter sweeps, please see the tutorial on parameter sweeping (located in *OptiSPICE Tutorials - Basic*). The parameter sweep design can also be found in the folder: *OptiSPICE 5.2 Samples\Tutorials\Advanced\Subcircuit\Optical Chebyshev Wavelength Sweep*

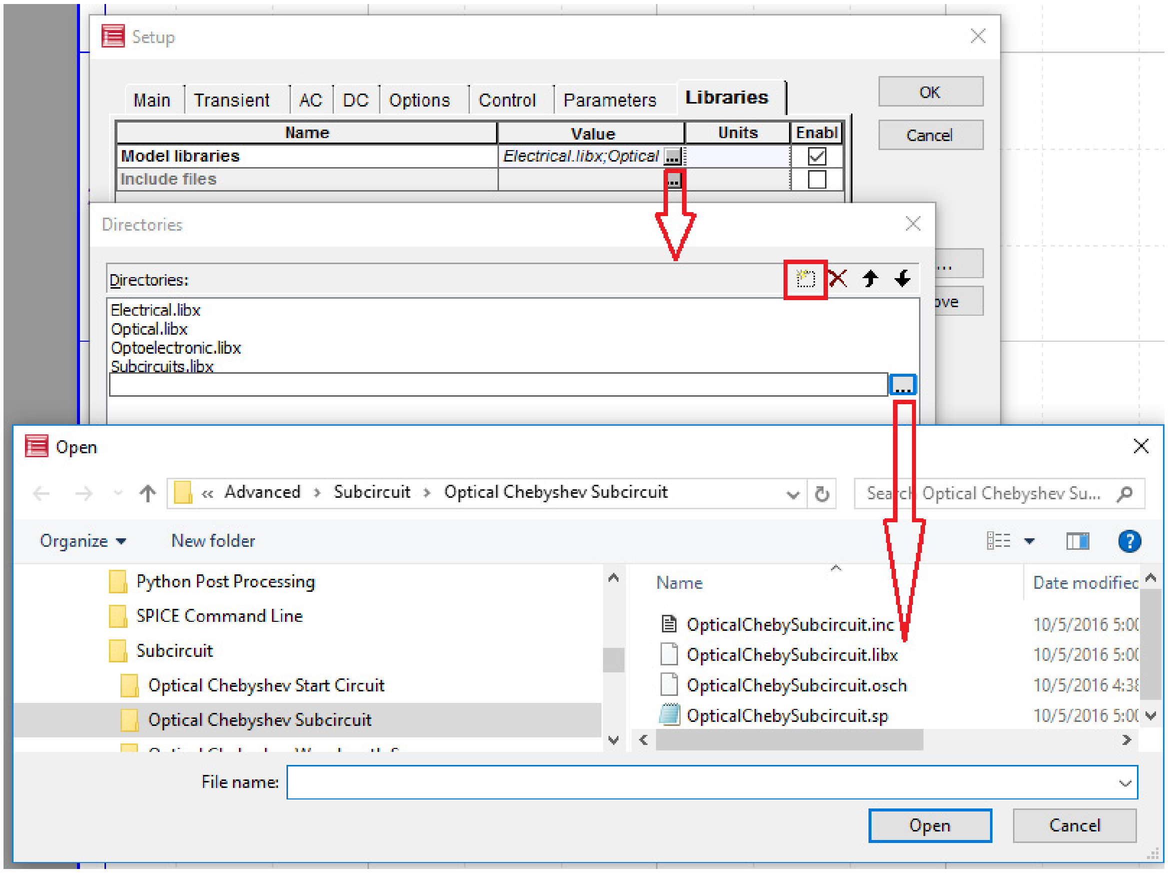
Figure 39 Wavelength sweep simulation



21 The model libraries are not automatically saved with the subcircuit so they need to be added to the design separately. Go to ***Setup>Libraries*** and click on the Grey icon box in the Model libraries data field. Add the library file created by the Chebyshev subcircuit schematic by first selecting the ***New Insert*** icon box (see RED box in Figure 40) and then selecting the Grey box at the end of the new line that has been added to the Directories list. Select “OpticalChebySubcircuit.libx” (from the folder where the subcircuit design was saved) and select ***Open*** and then ***OK***.

The new library directory folder will appear under the Directory window for the Model libraries.

Figure 40 Creating a link to the Subcircuit model



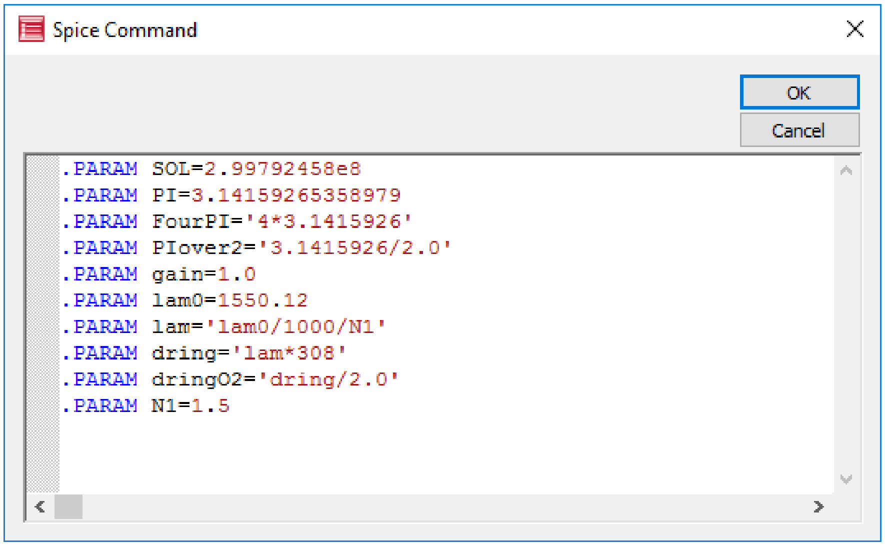
Name	Value	Units	Enabl
Model libraries	Electrical.libx;Optical		☑
Include files			□

Electrical.libx
Optical.libx
Optoelectronic.libx
Subcircuits.libx

Name	Date modified
OpticalChebySubcircuit.inc	10/5/2016 5:0
OpticalChebySubcircuit.libx	10/5/2016 5:0
OpticalChebySubcircuit.osch	10/5/2016 4:3
OpticalChebySubcircuit.sp	10/5/2016 5:0

22 Copy and paste the SPICE Command parameters for the subcircuit model onto the new schematic (Figure 41)

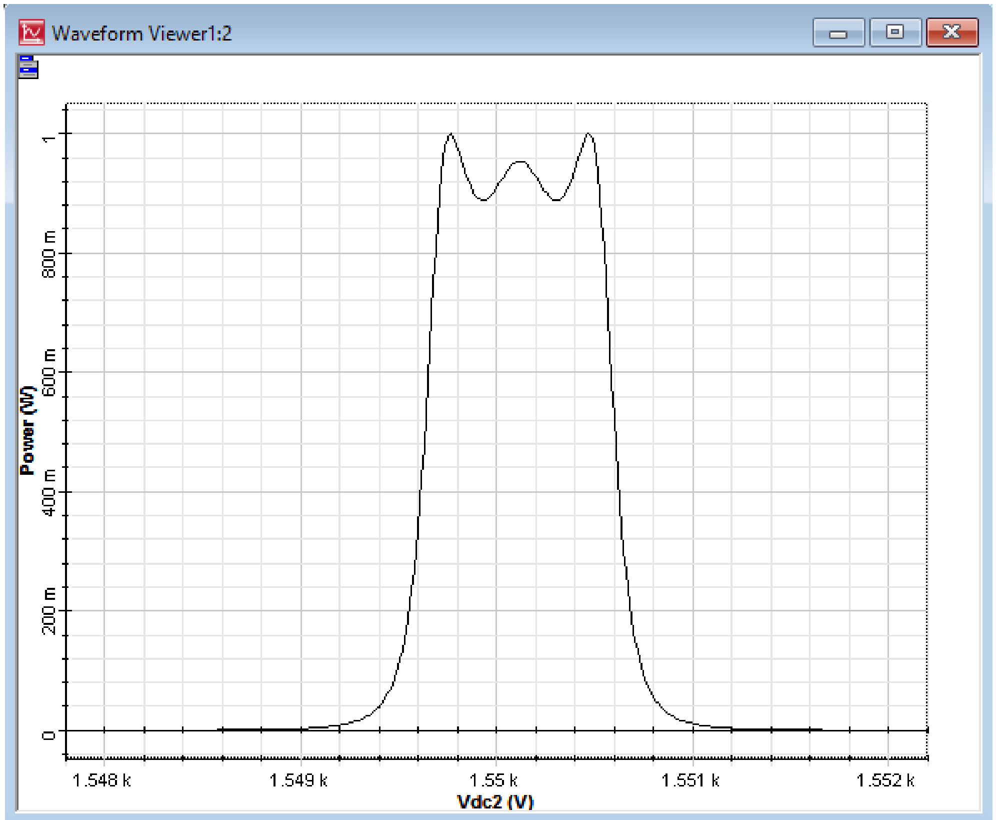
Figure 41 Parameters for the Subcircuit model



```
.PARAM SOL=2.99792458e8
.PARAM PI=3.14159265358979
.PARAM FourPI='4*3.1415926'
.PARAM PIover2='3.1415926/2.0'
.PARAM gain=1.0
.PARAM lam0=1550.12
.PARAM lam='lam0/1000/N1'
.PARAM dring='lam*308'
.PARAM dringO2='dring/2.0'
.PARAM N1=1.5
```

23 Run the simulation as a wavelength sweep. Figure 42 shows the simulation results

Figure 42 Ring resonator output results



Optiwave
7 Capella Court
Ottawa, Ontario, K2E 7X1, Canada

Tel.: 1.613.224.4700
Fax: 1.613.224.4706

E-mail: support@optiwave.com
URL: www.optiwave.com