GDS PHOTONICS LIBRARY IN MATLAB

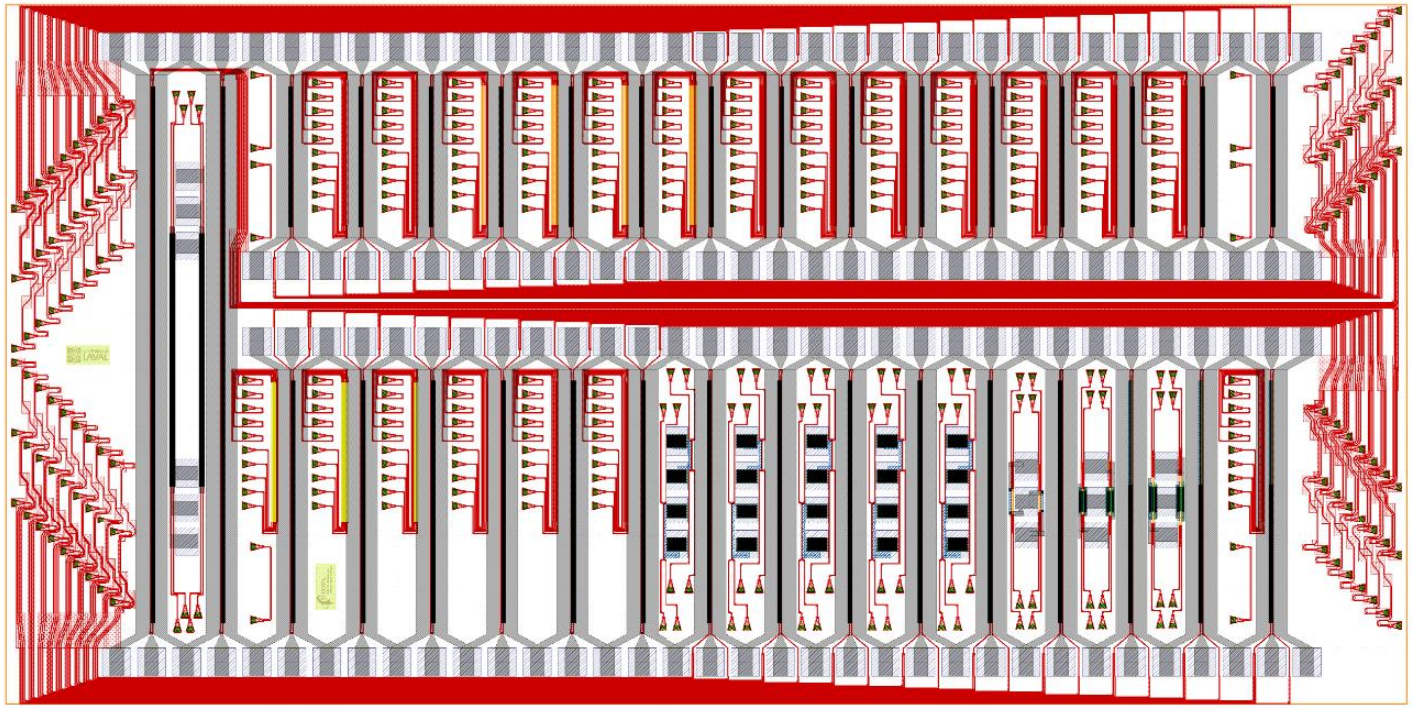
Nicolas Ayotte
Alexandre D. Simard

THE OBJECTIVE

- Have access to a powerful scripting interface
- Create functions to address some of the main issues of photonic floor plan integration like fast plan drawing, easy routing, length calculation and adaptability (Notable exception of DRC check)
- Be fully compatible with everyone everywhere with GDSII format (shout-out to KLayout editor).
- Have the capacity to export a floor plan to multiple layer schemes and fabrication facility formats

THE OBJECTIVE

- **Be more adaptable**



THE IDEA

- **When you have an idea but someone already did it…**
  - Thanks to Ulf Griesmann for his gdsii-toolbox project and his decision to simply give it to the public.

  https://sites.google.com/site/ulfgri/numerical/gdsii-toolbox
- **Build on it!**
  - No one wants to draw a floor plan point by point, polygon by polygon. (Especially not by code in this case!)
  - Hence this more comprehensive library of tools for photonics devices.

UNIVERSITÉ LAVAL

STARTING A PROJECT

- **The Project folder**

Name	Date modified	Type	Size
Cells	23/05/2014 11:59 ...	File folder	
Author_TEST_fab_projectName_v1.gds	23/05/2014 12:00 ...	GDS File	8,676 KB
Author_ulaval_projectName_v1.gds	23/05/2014 11:59 ...	GDS File	8,676 KB
braggProfile.mat	15/04/2014 12:41 ...	MATLAB Data	86 KB
CellA_StraightWG.m	23/05/2014 10:52 ...	MATLAB Code	2 KB
CellB_Microrings.m	15/05/2014 1:49 PM	MATLAB Code	3 KB
CellC_CompactIBGs.m	15/05/2014 1:36 PM	MATLAB Code	4 KB
CellD_RidgeIBGs.m	15/05/2014 1:36 PM	MATLAB Code	3 KB
CellE_CustomIBGs.m	15/05/2014 1:36 PM	MATLAB Code	3 KB
CellF_RoutingWG.m	23/05/2014 10:22 ...	MATLAB Code	6 KB
ExportMap.m	23/05/2014 11:46 ...	MATLAB Code	1 KB
Main.m	23/05/2014 11:59 ...	MATLAB Code	1 KB
MergeCells.m	15/05/2014 1:46 PM	MATLAB Code	1 KB
ProjectDefinition.m	15/05/2014 1:46 PM	MATLAB Code	2 KB

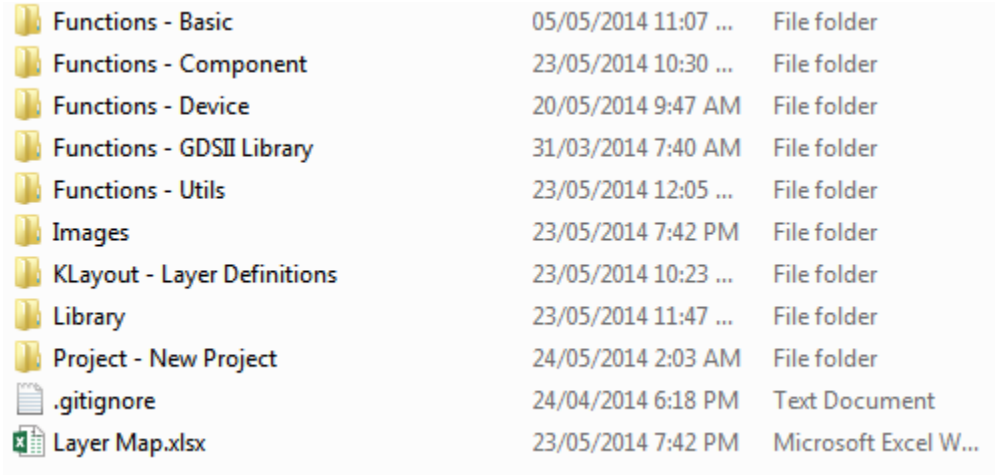
UNIVERSITÉ LAVAL

STARTING A PROJECT

- The Library Folder

This is Ulf Griesmann's library

Refence GDS (PDK, etc.)

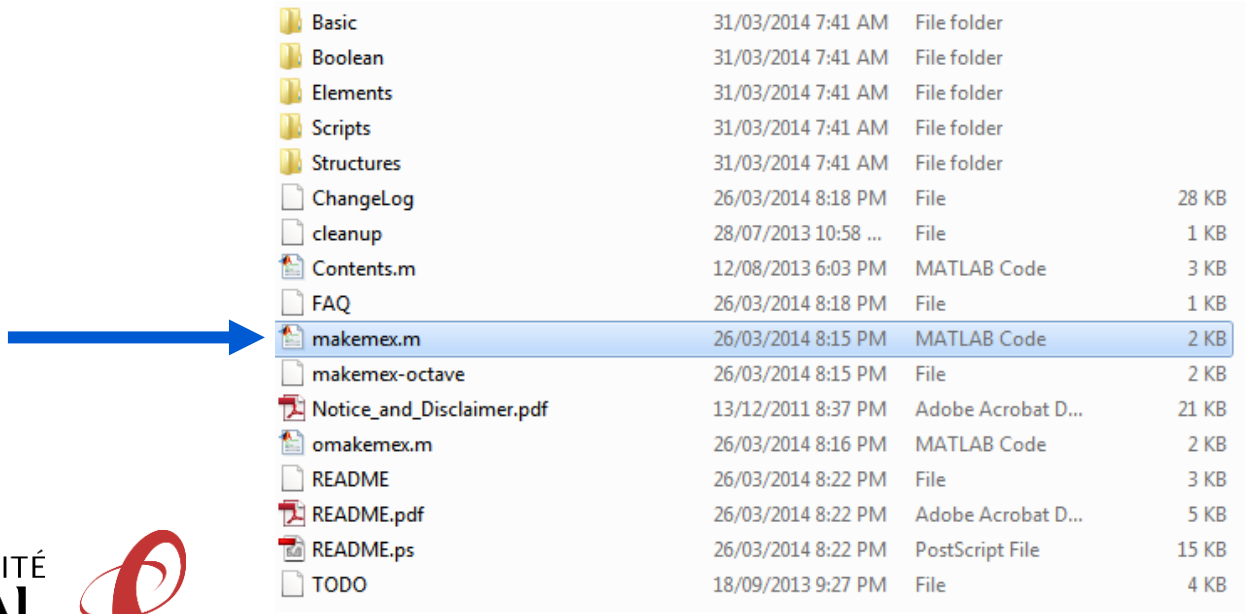


Functions - Basic	05/05/2014 11:07 ...	File folder
Functions - Component	23/05/2014 10:30 ...	File folder
Functions - Device	20/05/2014 9:47 AM	File folder
Functions - GDSII Library	31/03/2014 7:40 AM	File folder
Functions - Utils	23/05/2014 12:05 ...	File folder
Images	23/05/2014 7:42 PM	File folder
KLayout - Layer Definitions	23/05/2014 10:23 ...	File folder
Library	23/05/2014 11:47 ...	File folder
Project - New Project	24/05/2014 2:03 AM	File folder
.gitignore	24/04/2014 6:18 PM	Text Document
Layer Map.xlsx	23/05/2014 7:42 PM	Microsoft Excel W...

UNIVERSITÉ LAVAL

STARTING A PROJECT

- **Ulf's part of the library contains C functions that need to be compiled from Matlab using the Mex function.**
  - For people with 64-bit windows I have compiled those for you. Else, run "mex –setup" from the command window in Matlab then call :



UNIVERSITÉ LAVAL

STARTING A PROJECT

- **The Project folder**

Name	Date modified	Type	Size
Cells	23/05/2014 11:59 ...	File folder	
Author_TEST_fab_projectName_v1.gds	23/05/2014 12:00 ...	GDS File	8,676 KB
Author_ulaval_projectName_v1.gds	23/05/2014 11:59 ...	GDS File	8,676 KB
braggProfile.mat	15/04/2014 12:41 ...	MATLAB Data	86 KB
CellA_StraightWG.m	23/05/2014 10:52 ...	MATLAB Code	2 KB
CellB_Microrings.m	15/05/2014 1:49 PM	MATLAB Code	3 KB
CellC_CompactIBGs.m	15/05/2014 1:36 PM	MATLAB Code	4 KB
CellD_RidgeIBGs.m	15/05/2014 1:36 PM	MATLAB Code	3 KB
CellE_CustomIBGs.m	15/05/2014 1:36 PM	MATLAB Code	3 KB
CellF_RoutingWG.m	23/05/2014 10:22 ...	MATLAB Code	6 KB
ExportMap.m	23/05/2014 11:46 ...	MATLAB Code	1 KB
Main.m	23/05/2014 11:59 ...	MATLAB Code	1 KB
MergeCells.m	15/05/2014 1:46 PM	MATLAB Code	1 KB
ProjectDefinition.m	15/05/2014 1:46 PM	MATLAB Code	2 KB

STARTING A PROJECT

- **ProjectDefinition()**
  - This will be called in every cell you make to unify your project information. You should edit that at the start of your project.

```
%% Project information
cad = struct('author', 'Author', ...      % project author or company
  'fab', 'ulaval', ...                    % fabrication facility
  'process', 'processName', ...           % fabrication process
  'run', 'runName', ...                   % run name
  'name', 'projectName', ...              % project name
  'layermap', 'ulaval', ...               % layer map name
  'uunit', 1e-6, ...                      % CAD scale (1e-6 - > microns)
  'dbunit', 1e-11, ...                    % CAD database unit (1e-11 - > nm)
  'size', [10000, 2000], ...              % Floorplan dimensions
  'margin', struct('left', 100, 'right', 0, 'top', 50, 'bottom', 50), ...   % safety margin
  'v', 'v1');                            % version number
```

THE MAIN

```
%% Main
% Author : Nicolas Ayotte
% Creation date : 31/03/2014
% The entire programs is in microns.
clear all; close all; clear classes; clc; format long; format compact;

% ProjectDefinition.m is the reference for all your project informations


%% Make all
CellA_StraightWG;
CellB_Microrings;
CellC_CompactIBGs;
CellD_RidgeIBGs;
CellE_CustomIBGs;
CellF_RoutingWG;


%% Merge the cells into a master GDS
MergeCells;


%% Cast the ulaval map
ExportMap;
```

THE MAIN

- **The main creates all your .gds cells**
- **Merges them into a master .gds file**
- **And casts that .gds onto one or many other layer maps as necessary for your exports (or imports).**

- **Let's see in details how to create a simple cell .gds**
  - (Then we'll move on to the more hardcore functions)

YOUR FIRST CELL

- **InitializeCell() gives your project information**
  - cad is the information structure for your project
  - cellname is the string of the name of the Matlab file and will be the name of your top cell.
  - topcell is the gds_structure Matlab object from the core library in which we will be adding our polygons.
  - layerMap contains the ulaval layer map information
  - (log is a tiny display tool for information)

YOUR FIRST CELL

- WAVEGUIDE
  - Create a waveguide information structure

- guide = Waveguide(widths, layers, minimumCurvingRadius, minimumSpacing)

FIELD NAME	SIZE	DESCRIPTION
'w'	1 x m	width of each layer
'layer'	1 x m	layer number
'dtype'	1 x m	datatype
'sp'	1	minimum center-to-center spacing
'r'	1	minimum radius of curvature

If at any time you panic, type: help Waveguide

PANICKING?

- If at any time you forgot how to use a function type:
  - help FunctionName
  - It this case help Waveguide provides:

```
Waveguide Create a waveguide information structure
Author : Nicolas Ayotte                                  Creation date : 21/03/2014


    guide = Waveguide(widths, layers, minimumCurvingRadius, minimumSpacing)

    This function implements the naming convention of the fields for a Waveguide.
    The layer information is meant to come from a layerMap structure created
    by the ReadLayerMap function.


    FIELD NAME      SIZE        DESCRIPTION
    'w'             1 x m       width of each layer
    'layer'         1 x m       layer number
    'dtype'         1 x m       datatype
    'sp'            1           minimum center-to-center spacing between waveguides
    'r'             1           minimum radius of curvature

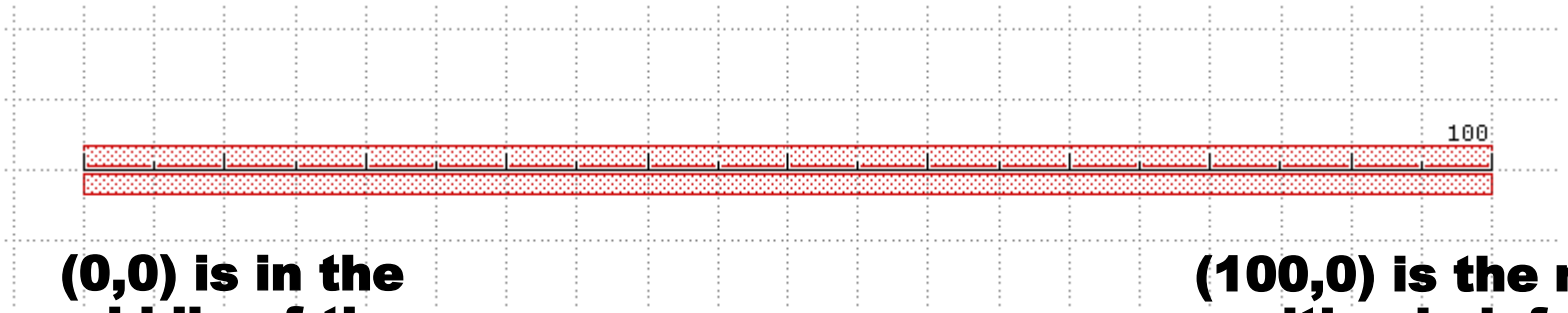
    See also Taper, FiberArray, ReadLayerMap
```

YOUR FIRST CELL

- CURSORINFO
- Create a cursor information structure. (This is your mouse!)
- That structure contains position and orientation vectorial information for one or many cursors.
- info = CursorInfo([0, 0], 90, [1, 2.7]);
- The last matrix is for effective indices, if desired, for length.
- Creates a cursor at the point (0, 0) pointing upwards and will cumulate the physical length and the optical length for a mode of index 2.7.

YOUR FIRST CELL – SOME ACTION AT LAST!

- phW = Waveguide([0.5, 3.5], [layerMap.FullCore, layerMap.FullClad], 5, 3.5);
- infoIn = CursorInfo([0, 0], 0, 1);
- [topcell, infoOut, infoIn] = PlaceRect(topcell, infoIn, 100, phW.w, phW.layer, phW.dtype);



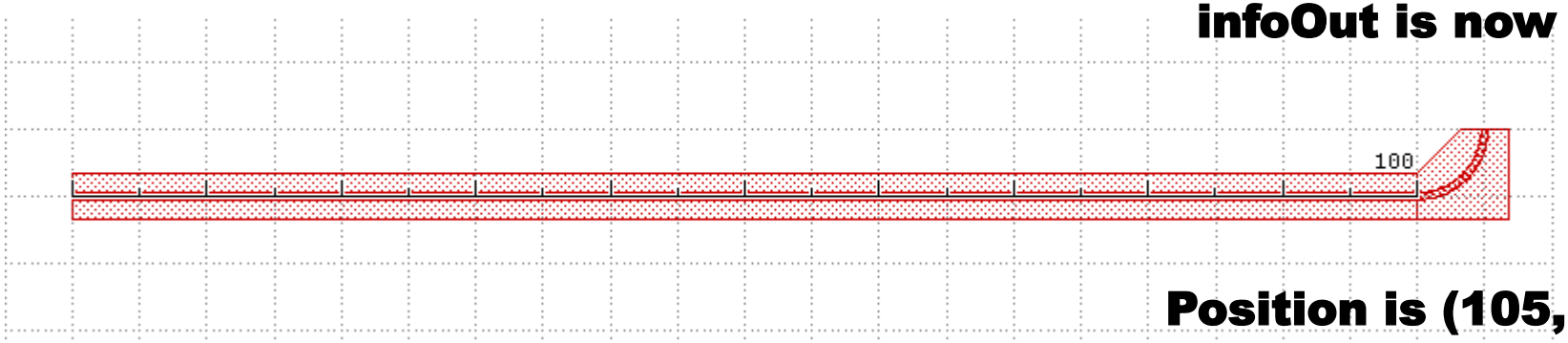
(0,0) is in the middle of the waveguide

(100,0) is the new position in infoOut

YOUR FIRST CELL – SOME ACTION AT LAST!

- [topcell, infoOut] = PlaceArc(topcell, infoOut, 90, phW.r, phW.w, phW.layer, phW.dtype);
- This turns 90 degrees with a radius defined in our phW object (minimum turn radius)
  - Note the shape of the cladding.

Orientation in infoOut is now 90.

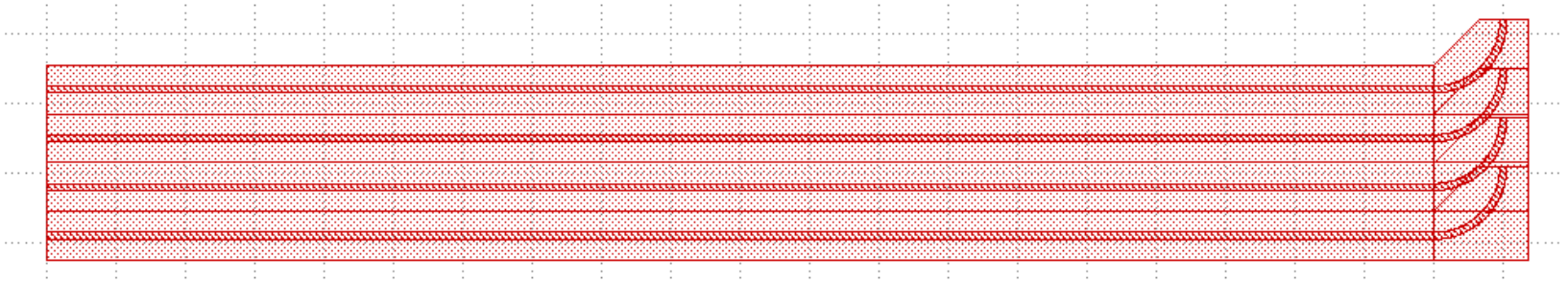


Position is (105, 5)

But as everything is relative, you probably never need to know exactly where you are.

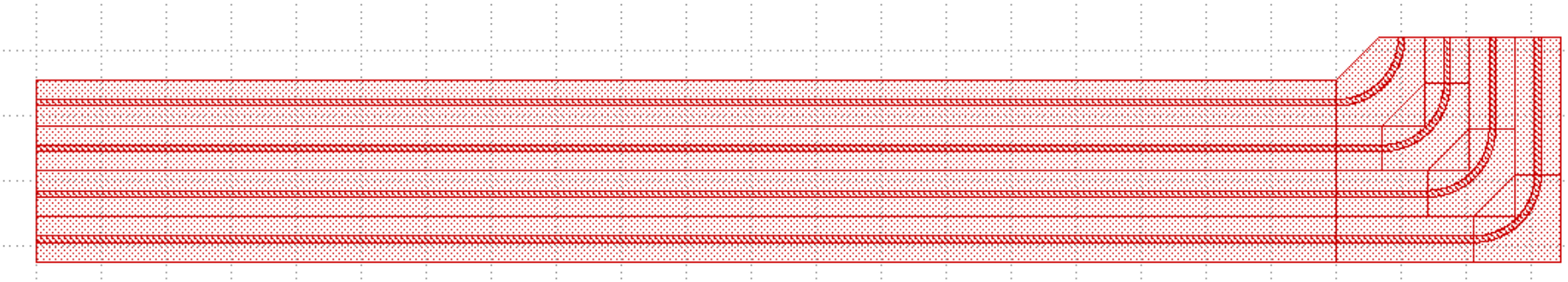
YOUR FIRST CELL – SOME ACTION AT LAST!

- infoIn = CursorInfo([0, 0;…
- 0, phW.sp; …
- 0, 2 * phW.sp; …
- 0, 3 * phW.sp], 0, 1);
  - Multiple cursors simultaneously through rows
  - Common problem : turning groups



YOUR FIRST CELL – SOME ACTION AT LAST!

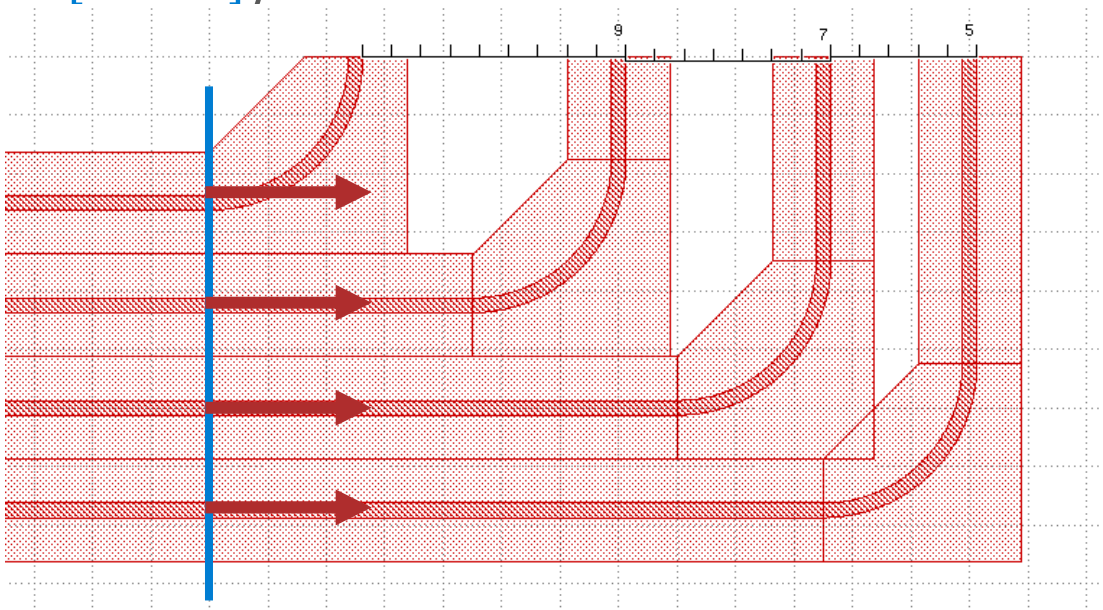
- [topcell, infoOut] = PlaceArc(topcell, infoOut, 90, phW.r, phW.w, phW.layer, phW.dtype, 'group', true)



The cladding shape makes more sense now.

YOUR FIRST CELL – CHANGE DISTANCE

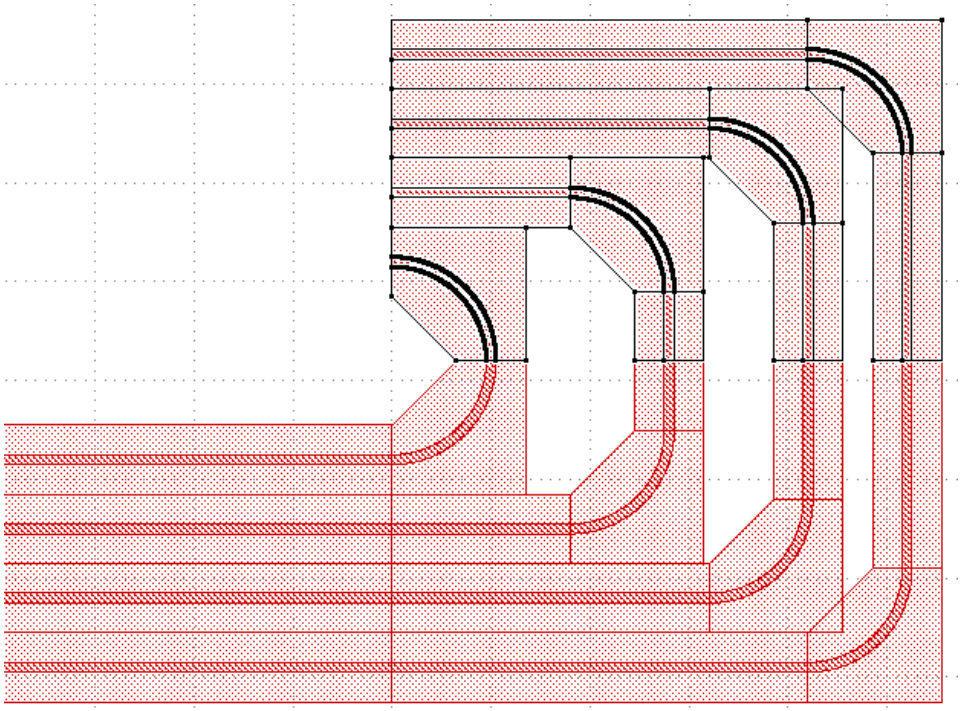
- [topcell, infoOut] = PlaceArc(topcell, infoOut, 90, phW.r, phW.w, phW.layer, phW.dtype, 'group', true, 'distance', [5 7 9]);



The guides need to be parallel and aligned on the normal line.

YOUR FIRST CELL – TIGHTEN YOUR GUIDES

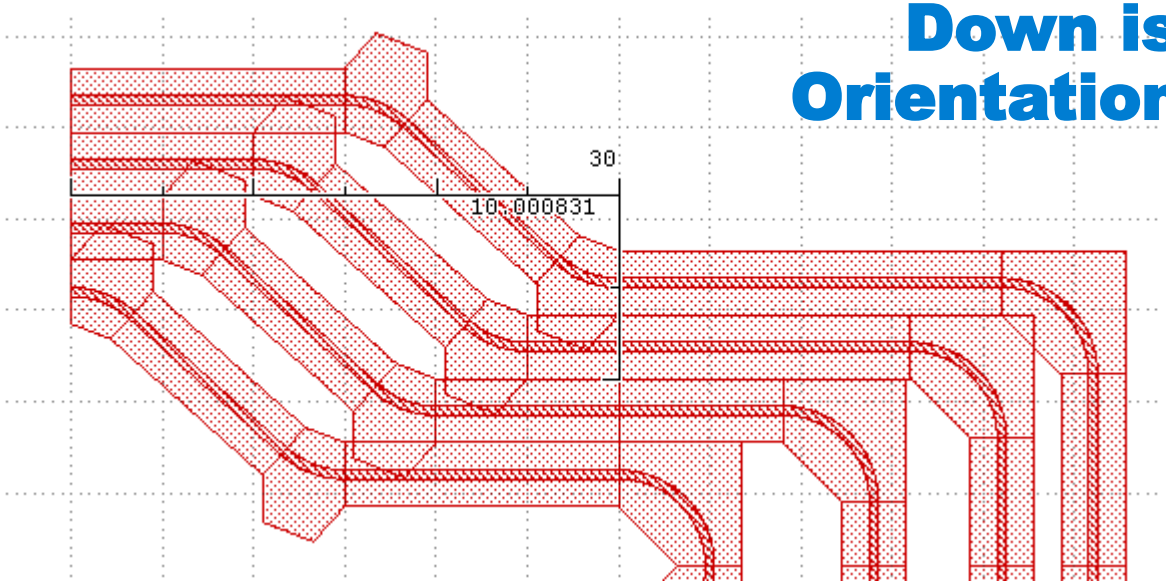
- [topcell, infoOut] = PlaceArc(topcell, infoOut, 90, phW.r, phW.w, phW.layer, phW.dtype, 'group', true, 'distance', phW.sp);



No need to know the original distance!

YOUR FIRST CELL – SBEND

- [topcell, infoOut] = PlaceSBend(topcell, infoOut, 30, -10, phW.r, phW.w, phW.layer, phW.dtype, 'group', true);

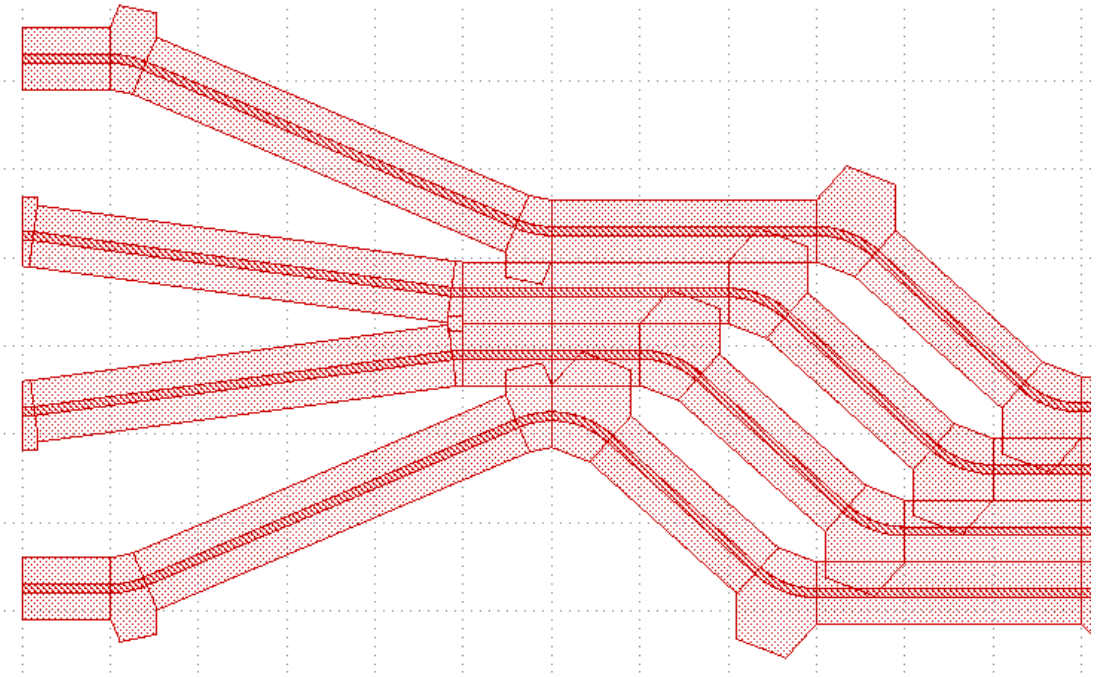


Down is up.
Orientation = 180°

But can you change the distance?

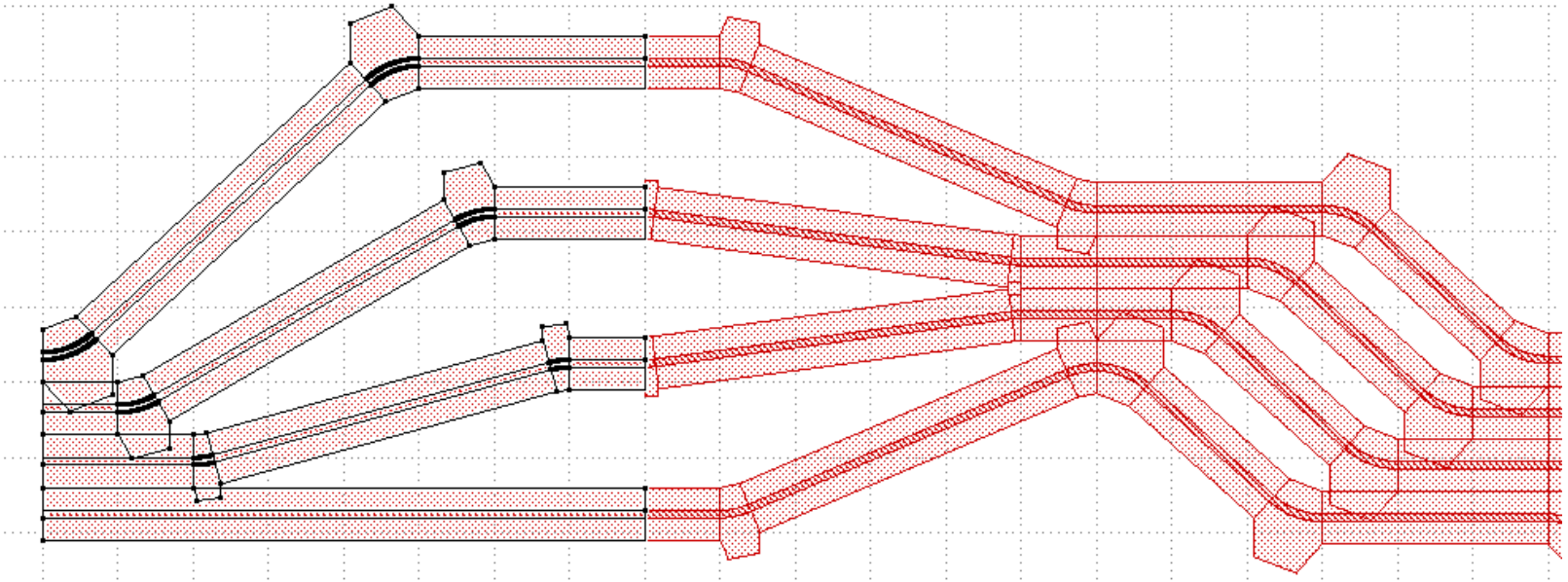
YOUR FIRST CELL – SBEND DISTANCE CHANGE

- [topcell, infoOut] = PlaceSBend(topcell, infoOut, 30, 0, phW.r, phW.w, phW.layer, phW.dtype, 'group', true, 'distance', 10);



YOUR FIRST CELL – SBEND ALIGN

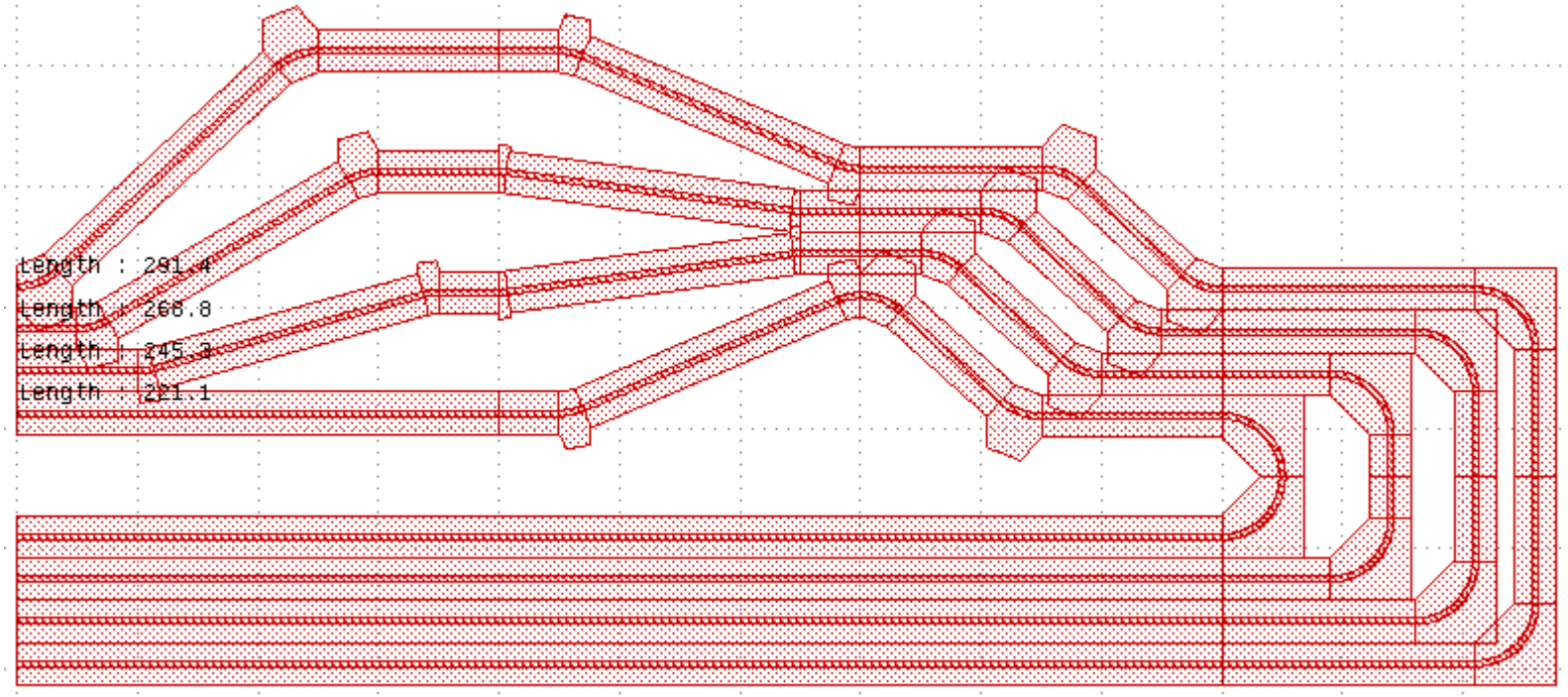
- [topcell, infoOut] = PlaceSBend(topcell, infoOut, 40, 0, phW.r, phW.w, phW.layer, phW.dtype, 'group', true, 'distance', phW.sp, 'align', 'top');



The top (ori = 180°) guide has 0 um vertical offset

YOUR FIRST CELL – LENGTH

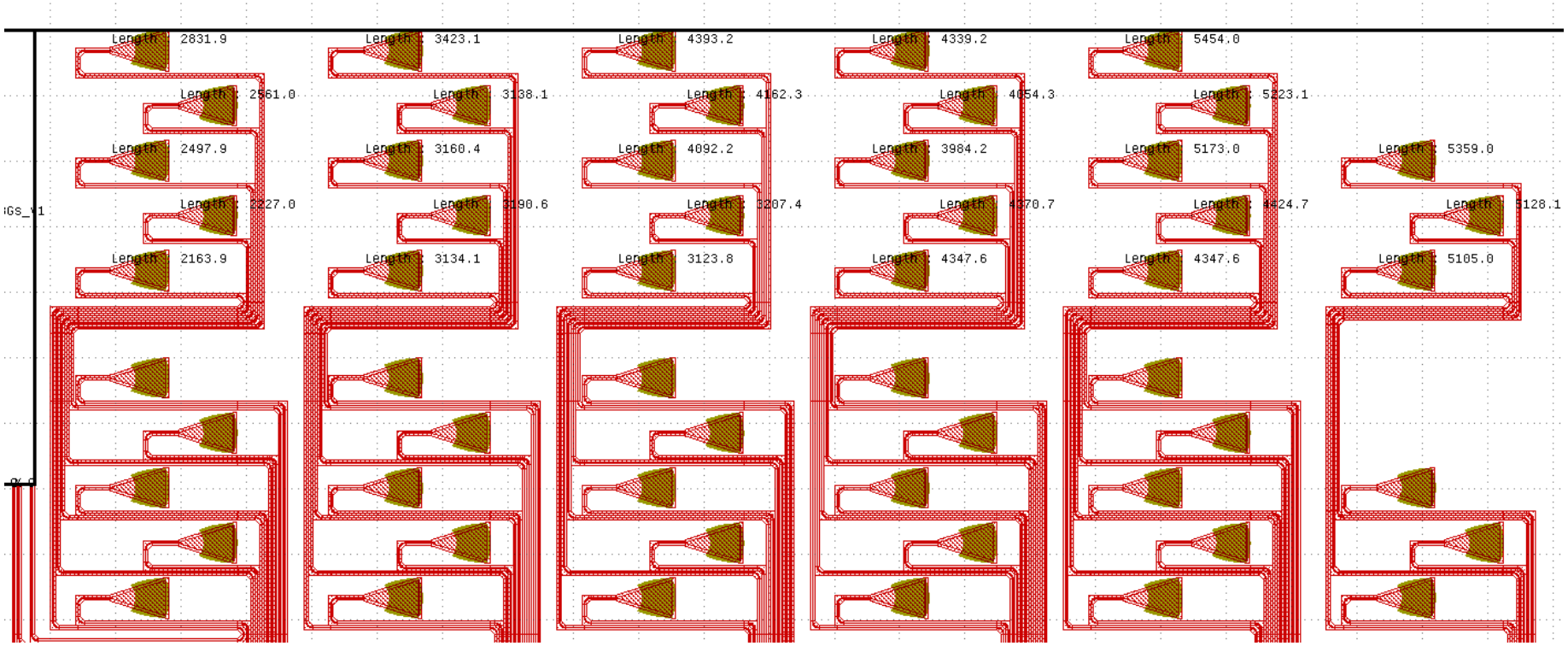
- totalLengths = infoIn.length + infoOut.length
- [291.38; 268.79; 245.26; 221.115] (micrometers)



This was done in 7 commands.

YOUR FIRST CELL – LENGTH

- Coupler to coupler physical length



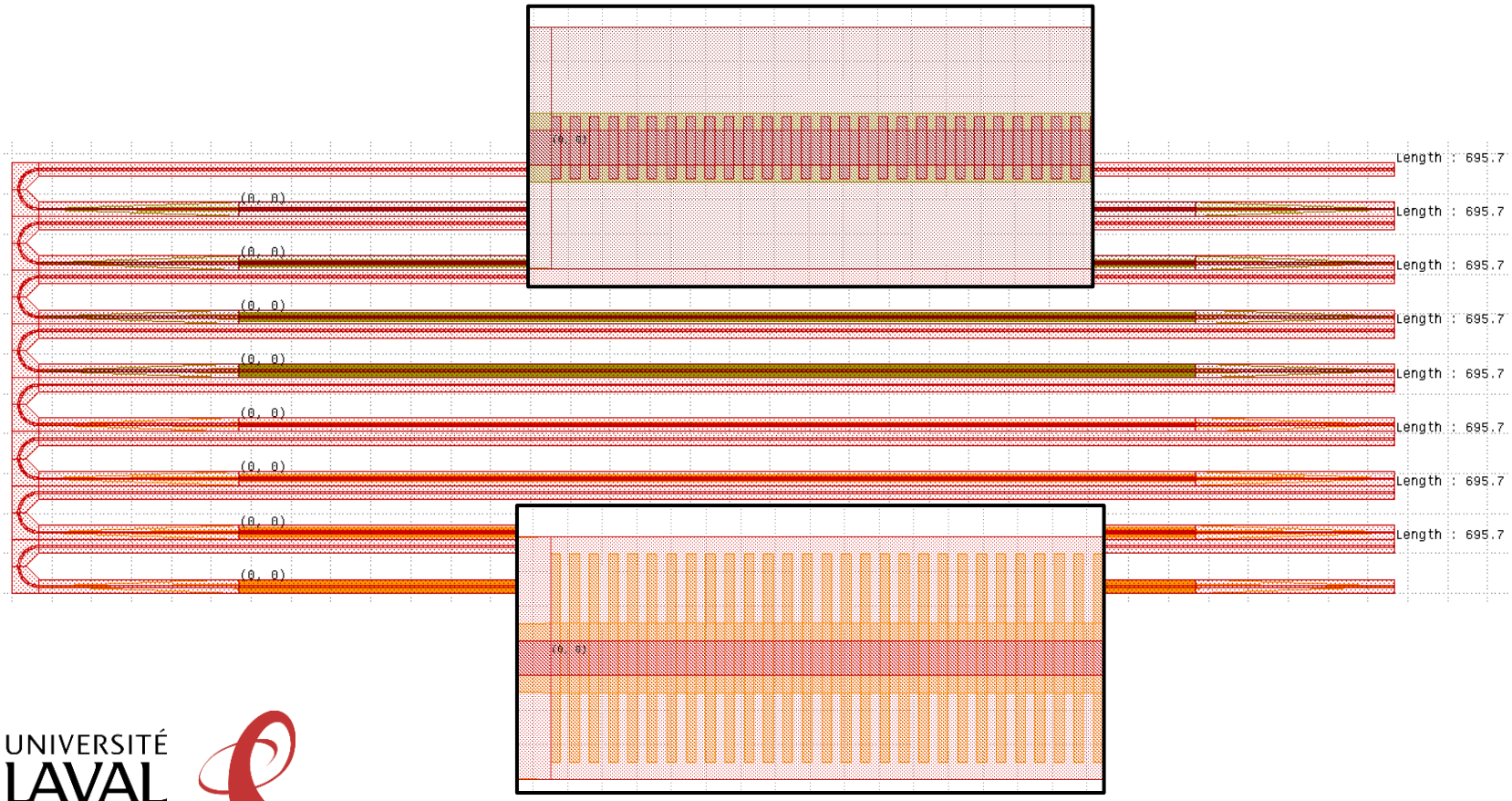
DARE I SAY BRAGG GRATINGS?

- bragg = BraggFromParameters(len, w, period, corw, dc, chirp, phaseShift, layer, dtype)
- ARGUMENT NAME SIZE DESCRIPTION
- 'len' 1 | n x 1 length of the grating
- ‘w' 1 | n x 1 | m width
- 'period' 1 | n x 1 | m period
- 'corw' 1 | n x 1 | m corrugation width
- 'dc' 1 | n x 1 | m duty cycle
- 'chirp' 1 | n x 1 | m chirp
- 'phaseShift' 1 | n x 1 | m phase shifts

UNIVERSITÉ LAVAL

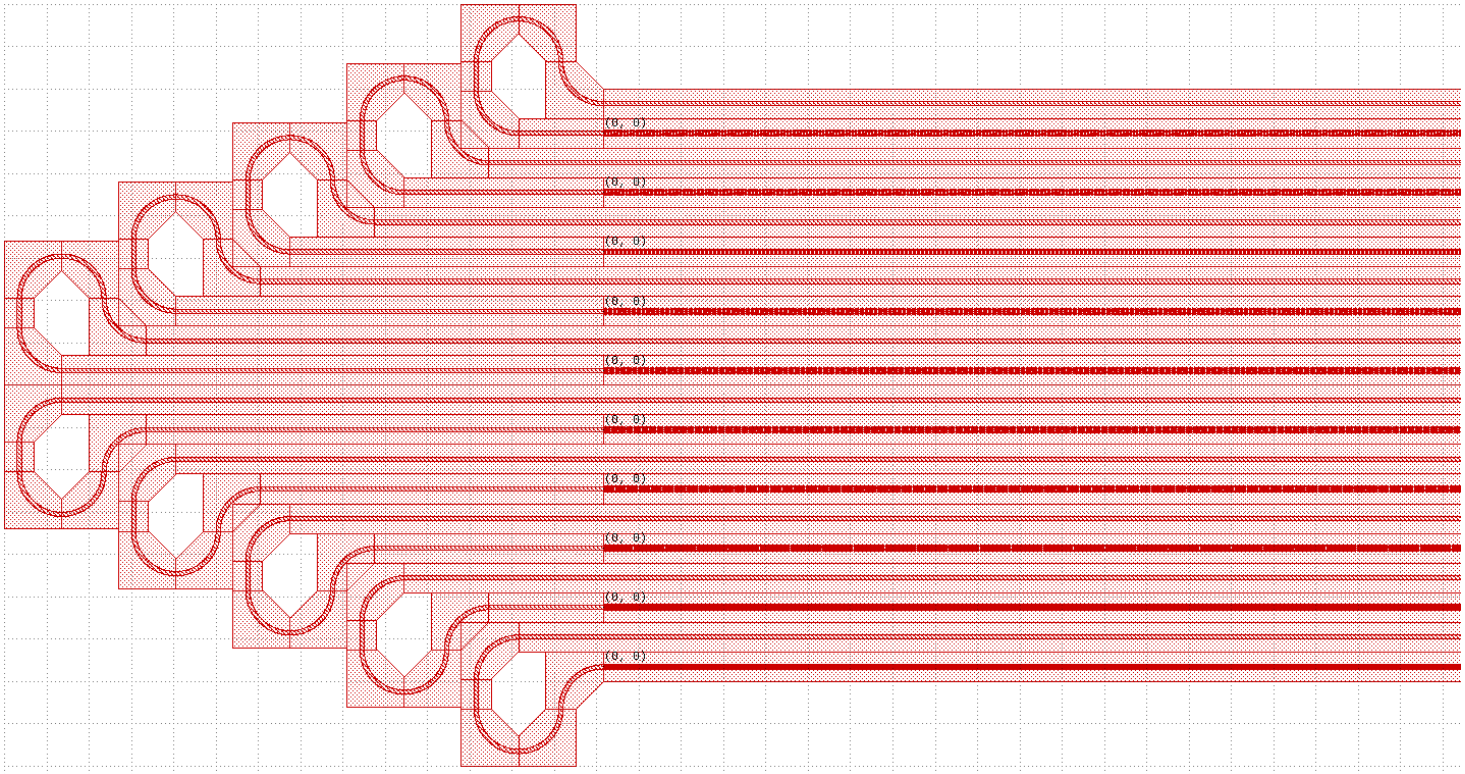
RIDGE BRAGG GRATINGS

- The code: CellD_RidgeIBGs.m



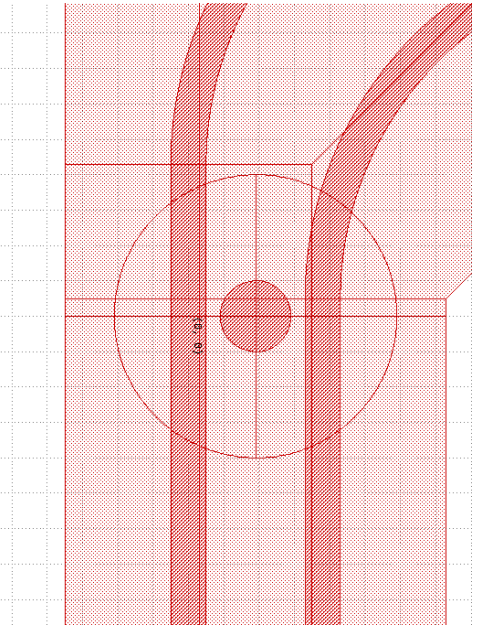
COMPACT BRAGG GRATINGS

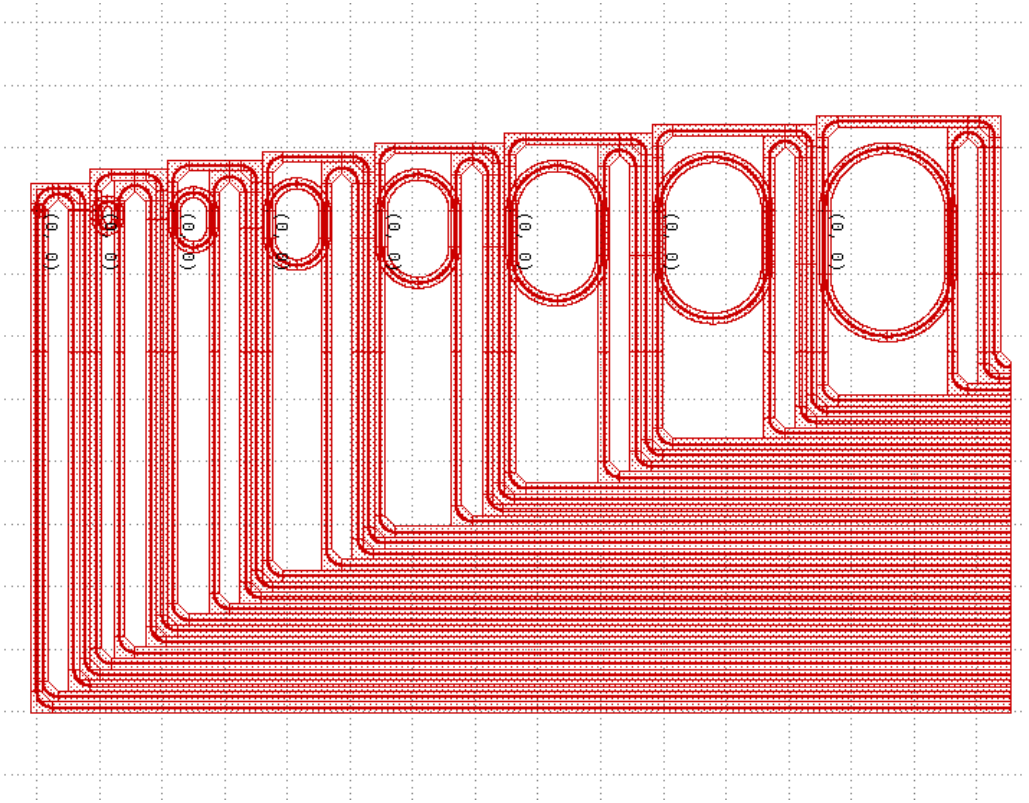
- The code for this cell is CellC_CompactIBGs.m



MICRORINGS

- The code for this cell is CellB_Microrings.m

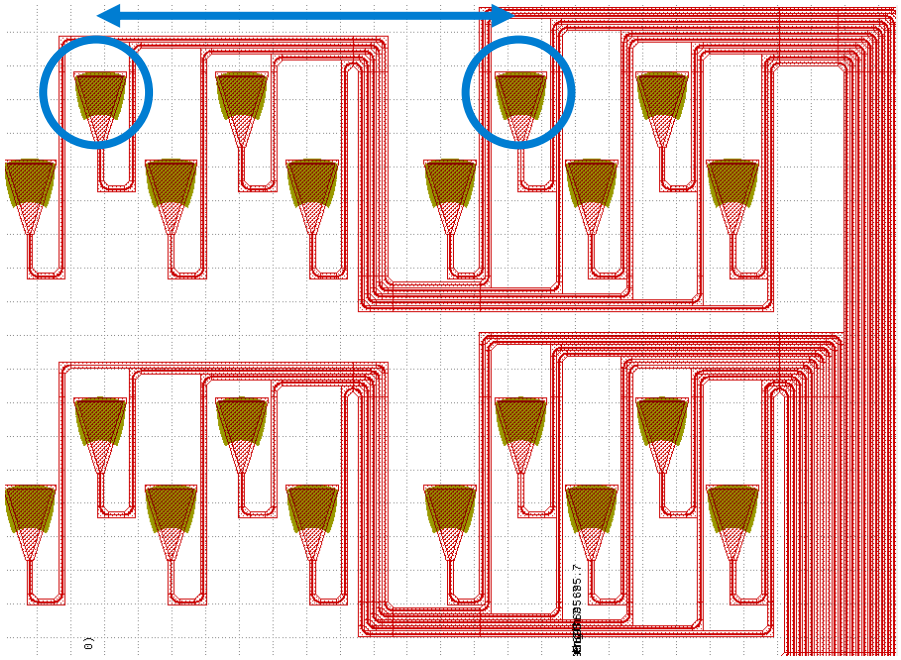




PLACE COUPLER ARRAY

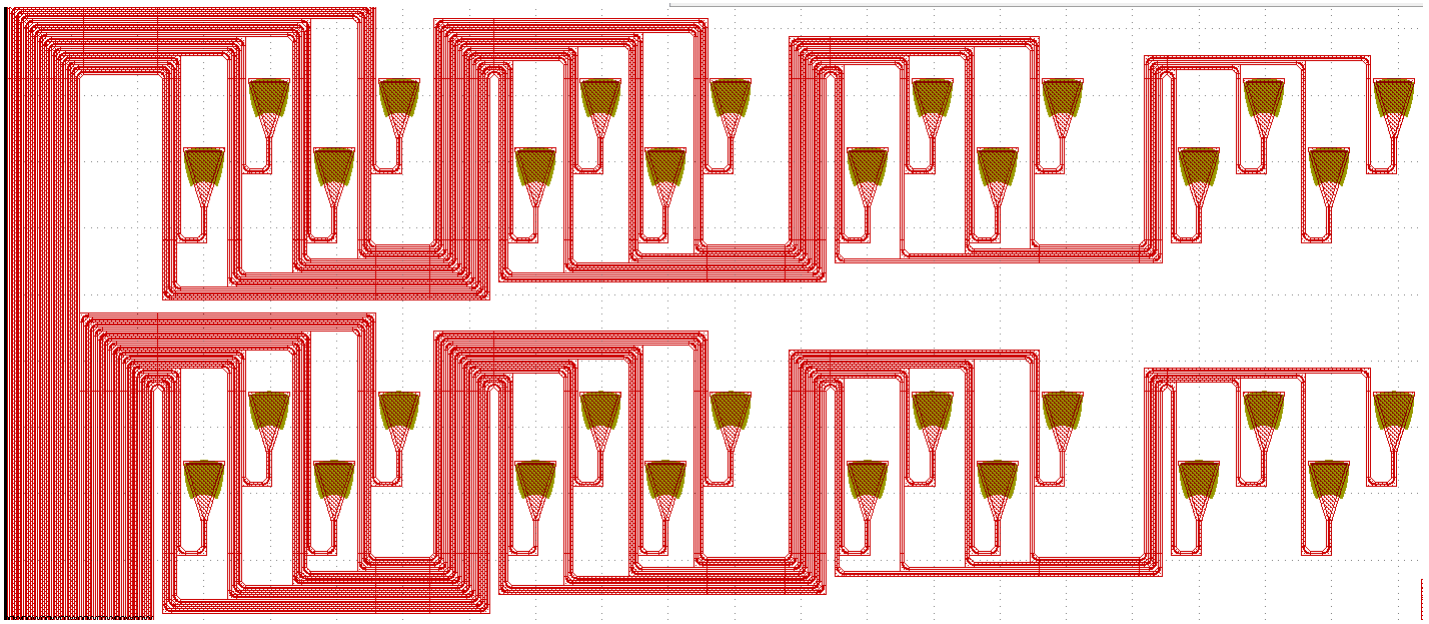
- For a waveguide pattern of interleaved inputs and outputs, the PlaceCouplerArray function does the work for you

250 um spacing for the fiber array



PLACE COUPLER ARRAY

- Just tell it how many I/Os one of your device has (shown here is 4 I/Os like for the microrings).

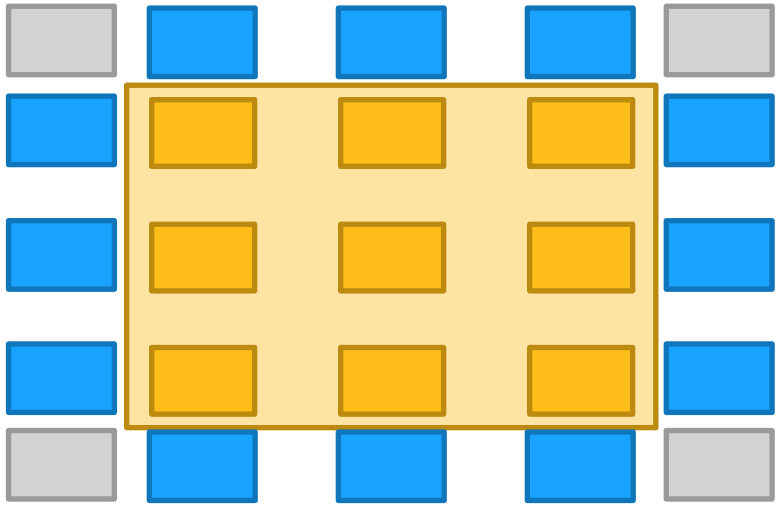


ROUTING SOLUTION

- Cell placement uses a system of anchors.
  - First the code draws the floorplan size and the safety margin rectangles

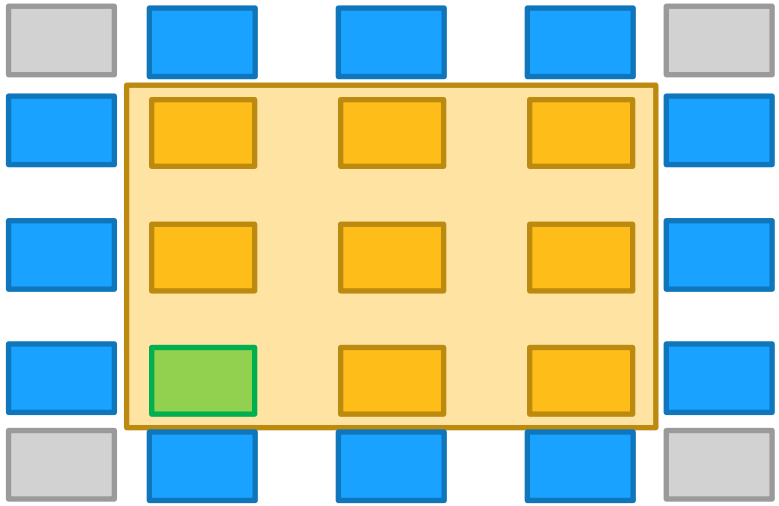
ROUTING SOLUTION

- Then you can anchor a cell to any other cell or to the floorplan in any of 25 position combinations
  - verticalAlign : topInside, topOutside, center, bottomInside, bottomOutside
  - HorizontalAlign: leftInside, leftOutside, center, rightInside, rightOutside



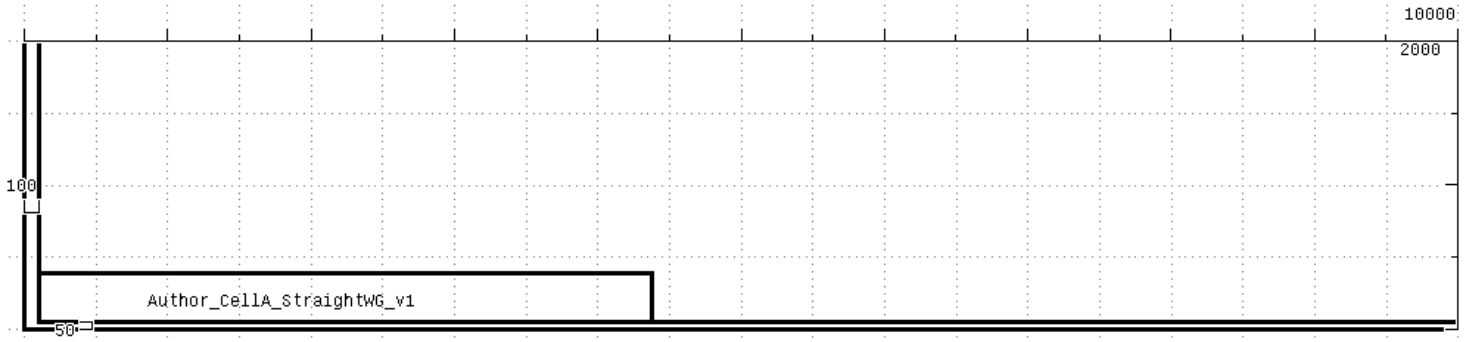
ROUTING SOLUTION

- For example :
- ```
  [topcell, cellInfo] = PutCell(topcell, cad, cellInfo, layerMap,
      'CellA_StraightWG', 'anchor', 'floorplan', ...
      'verticalAlign', 'bottomInside', ...
      'horizontalAlign', 'leftInside');
  ```
- Places the cell CellA_StraightWG at the position in green on the figure below in the floorplan (inside the margin).


  ```

ROUTING SOLUTION

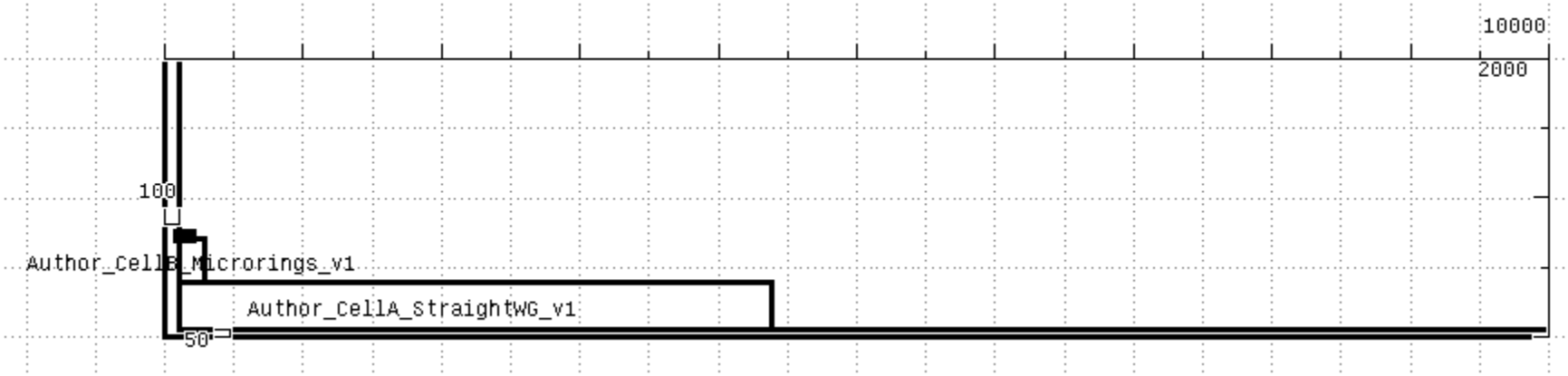
- Creating the following CellF_RoutingWG_v1.gds file



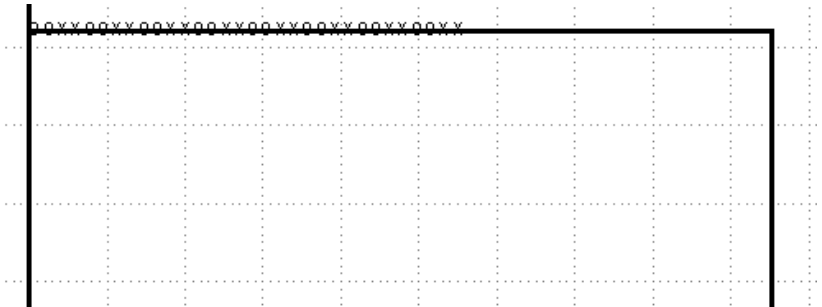
This rectangle and the name of the cell are there to help you route your waveguides. The actual cell reference is done when merging all the cells .gds together

ROUTING SOLUTION

- When we add cellB anchored leftInside and topOutside of cellA we get

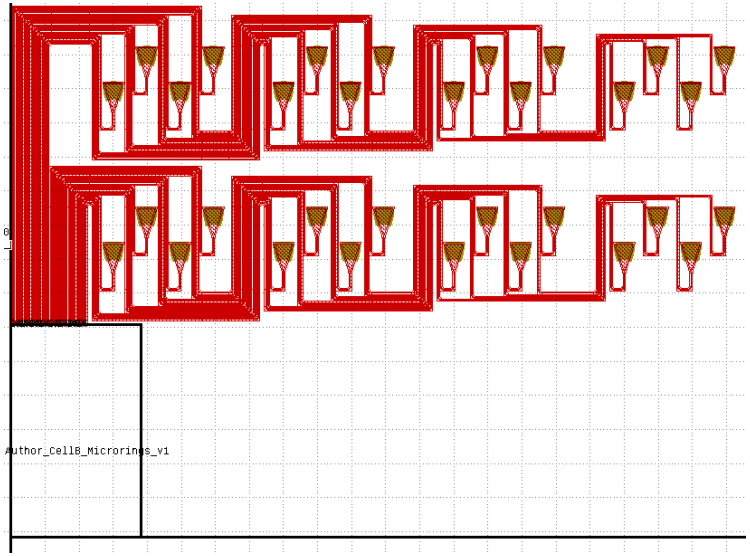


You can zoom in and see little 'X's and 'O's pointing where the cells outputs and inputs are located



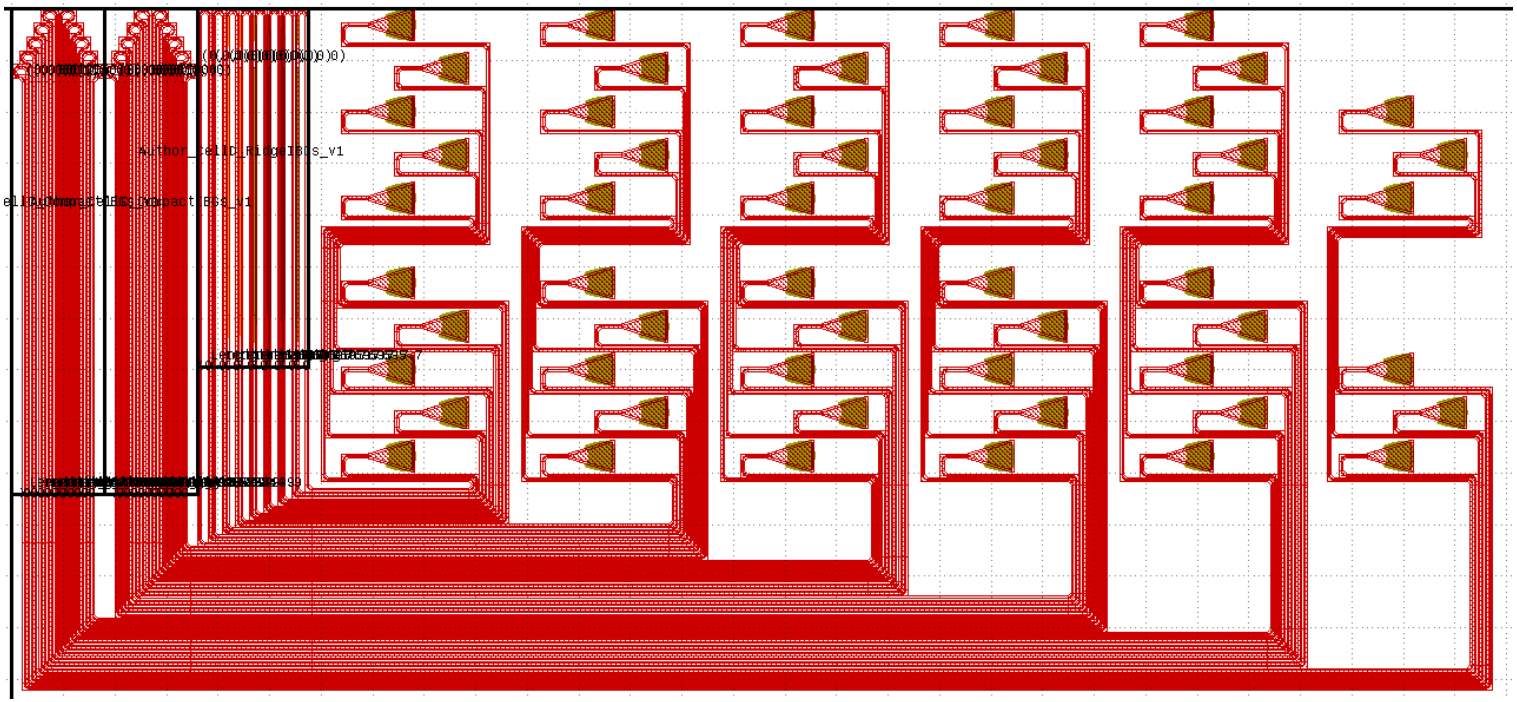
ROUTING SOLUTION

- If we take all the inputs and outputs and call PlaceCouplerArray
  - `infoB = MergeInfo(infoIn, infoOut);`
  - `[topcell, ~, ~, arraySize] = PlaceCouplerArray(topcell, infoB, 4, phW, fA, refs(1).cellname, 'type', 'cladding', 'direction', -1);`



ROUTING SOLUTION

- The Bragg grating cells.



CASTING LAYER MAPS

- This function switches layers following another layer definition.
- There is an Excel file to facilitate the creation of transfer maps between two facilities.
- Our layer maps has been designed to be extensive so that all other layer maps can be derived from ours through either simple layer correspondence or through layer boolean operations.

CASTING LAYER MAPS

- Simple enter the information and obtain a correspondence map

Conversion table				
	Ulaval		Example Fab/Process	
Name	Layer	Datatype	Layer2	Datatype2
FullCore	1	0	2	0
FullClad	1	1	3	0
FullHole	1	2	4	0
FullTrench	1	3	5	0
MidCore	2	0	6	0
MidClad	2	1	7	0

Map without operations
Paste these four lines into the corresponding switch : case in the DefineMap.m function
layerLaval = [1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 7, 8, 9, 11, 11, 11, 12, 12, 12, 13, 14, 21, 21, 21, 21, 22, 22, 22, 22, 22, 23, 29, 29, 31, 32, 32, 32, 33, 3
datatypeLaval = [0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 0, 0, 1, 2, 3, 1, 2, 3, 1, 1, 1, 3, 4, 5, 1, 2, 3, 4, 5, 1, 1, 2, 0, 3, 4, 0, 0, 1, 2, 3, 0, 0, 0, 0, 0, 1];
layerOther = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 35, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34, 26, 36, 37, 38, 39, 40,
datatypeOther = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];

UNIVERSITÉ LAVAL

CASTING LAYER MAPS

- This can be copied and pasted in the relevant Matlab file CastDefineMap.m to get customized export options

```
switch mapName
   case 'TEST_fab'
      layerLaval = [1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 7, 8, 9, 11, 11, 11, 12, 12
      datatypeLaval = [0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 0, 0, 1, 2, 3, 1, 2,
      layerOther = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 35, 15, 16, 17, 18, 19,
      datatypeOther = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
end
```

- Which can then be used in ExportMap.m by simply specifying the target layer map ‘TEST_fab’

CASTING LAYER MAPS

- This can include any Boolean layer operations using the (included) open source Clipper library courtesy of Angus Johnson © 2010-2014
  - I did ask his express permission to include it.
  - The function for Boolean information have been tested, though not included in the library as it is for they have not been necessary
  - It is in my objectives to include it.

THANK YOU ALL FOR LISTENING

- I have had a lot of fun designing and working at this project and I sincerely hope this can help some of you.
- I do have a Git setup for the project code for people who are interested you can simply message me at nicolasayotte@gmail.com. Or visit https://github.com/nicolasayotte/MatlabGDSPhotonicsToolbox

Do you have any questions?
Can I help you with something?

UNIVERSITÉ LAVAL

THE LOG

- Here is an example of the information the log provides for the CellA_StraightWG.m

```
>> CellA_StraightWG

_____________________________________________________________

FUNCTION CellA_StraightWG


  ProjectDefinition  -  0:00:00.02

    Author : Author
    Fabrication Facility : ulaval
    Process : processName
    Run : runName
    Design name : projectName
    Version : v1

  Output file name: Cells\Author_CellA_StraightWG_v1.gds
  Top Cell name: Author_CellA_StraightWG_v1

  ReadLayerMap  -  0:00:00.09

    Loading the layer map: ulaval

  AddRefsToLib  -  0:00:00.80

    Read gds: ../Library/FGC_1550_a20_si220_Wei.gds
      Adding cell: FGC_1550_a20_si220_Wei


END  -  0:00:01.00
```