The Abdus Salam International Centre for Theoretical Physics
50th Anniversary 1964–2014

**2572-14**

# Winter College on Optics: Fundamentals of Photonics - Theory, Devices and Applications

*10 - 21 February 2014*

## Simulations Laboratory

Aitor V. Velasco
*Universidad Complutense de Madrid*
*Spain*

Strada Costiera, 11 - 34151 - Trieste - Italy • Tel. +39 0402240111 • Fax. +39 040224163 • sci_info@ictp.it • www.ictp.it
ICTP is governed by UNESCO, IAEA, and Italy, and it is a UNESCO Category 1 Institute

# SIMULATION LAB

Aitor V. Velasco
Carlos A. Ramos

UNIVERSIDAD COMPLUTENSE
MADRID

ICTP Winter College 2014

# OUTLINE – DAY 1 (BASIC)

1. Introduction
2. Electromagnetic approach: FDTD
3. Circuit approach: Aspic

# OUTLINE – DAY 2 (ADVANCED)

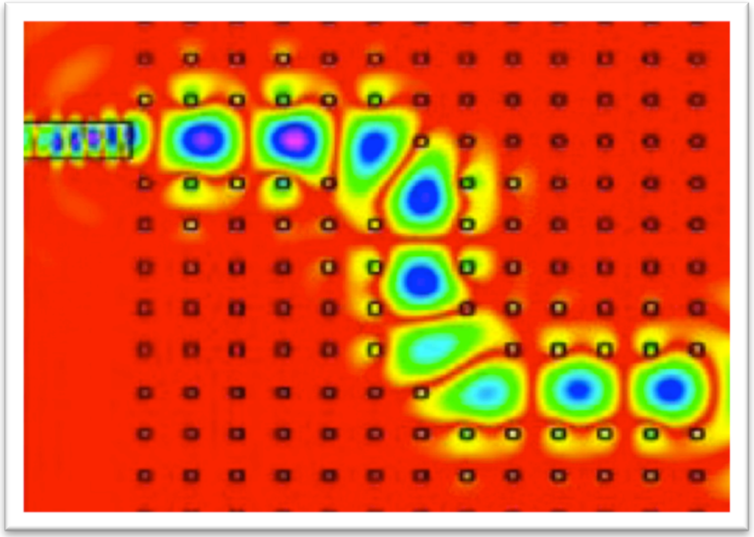
1. FDTD Simulations
   - Polarization converter
   - Subwavelength waveguides

2. Aspic Simulations
   - Resonant rings with Bragg reflector

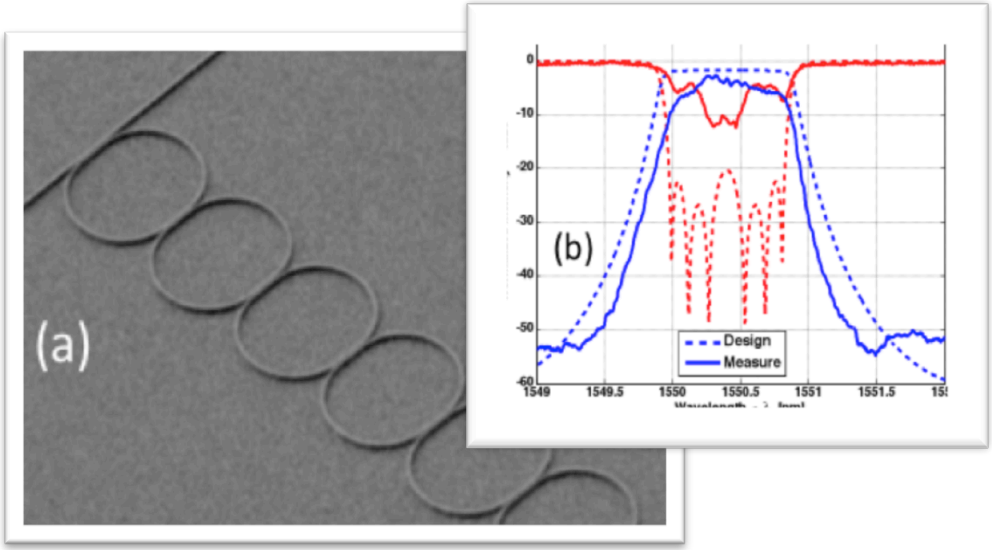
# 1. INTRODUCTION

# Introduction

- Photonic simulation is an essential tool for researchers and designers.
- It is also one of the cheapest ways of making science!
- The advances in computer hardware enables us to tackle a wide variety of problems with high computational load.



Light propagation in a photonic crystal




Intensity transfer in a 6 ring filter

# Two types of simulations

Electromagnetic approach
(physical modeling)

**vs**

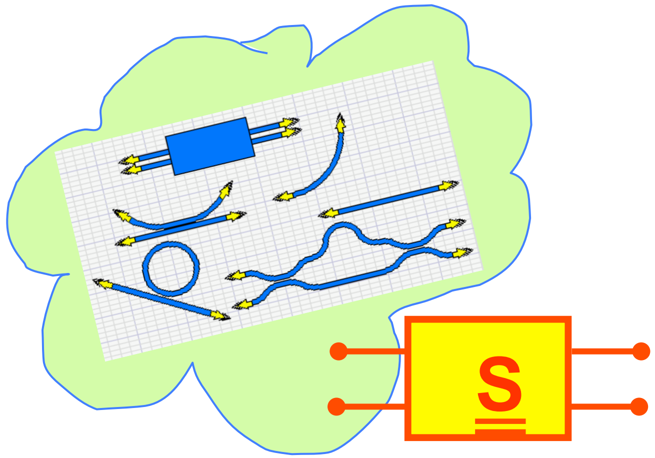
Circuit approach
(high abstraction level)

$$\nabla \times \mathbf{E} = -j\omega\mathbf{B}$$

$$\nabla \times \mathbf{H} = j\omega\mathbf{D} + \mathbf{J}$$

$$\mathbf{B} = \mu\mathbf{H}$$

$$\nabla \bullet \mathbf{D} = \rho$$




-Flexible: solves Maxwell's equations for an arbitrary design

-Large time and memory requirements

-Suitable for small elements

Lumerical FDTD Solutions

-Uses modeled components

-Access only to input/output port waves

-Smaller time and memory requirements

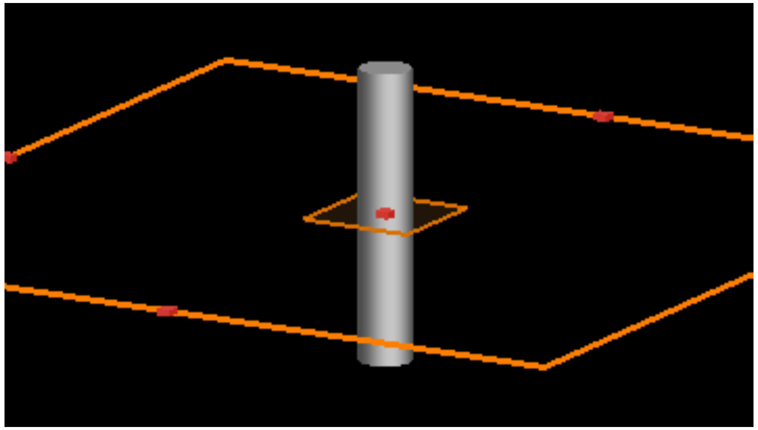
-Suitable for large circuits

Aspic Design

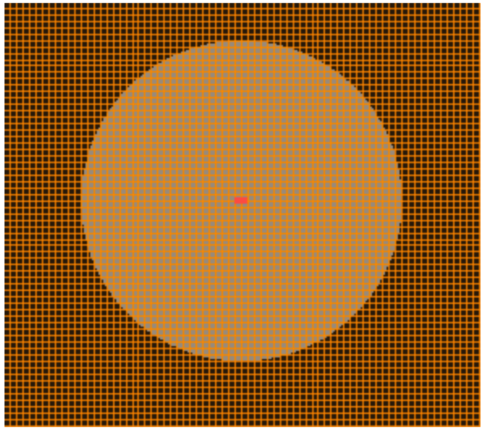
# 2. ELECTROMAGNETIC APPROACH: FDTD

# Overview

- The goal is to solve Maxwell equations on a discrete mesh structure (Yee cells). Each cell is defined by the material properties



Nanowire



Mesh grid

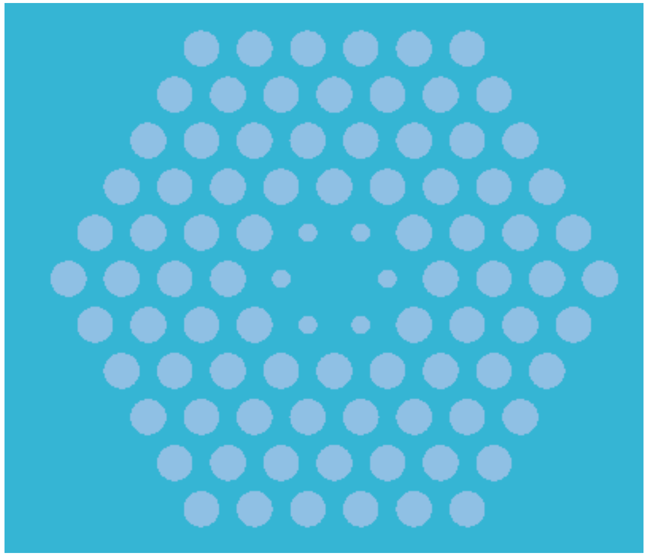
- Discrete time: E and H
  - Electric and magnetic field are computed alternatively for every time step
  - Simulates a short light pulse
  - The steady state is computed by Fourier transform

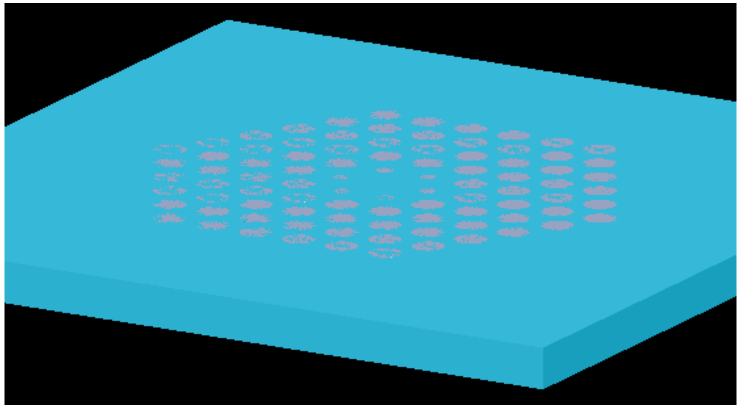
# Simulation process

- Define the physical structures
  - Assign a permitivity to each cell
- Define a simulation region
- Define a source of light
- Define monitors to record data

# 2D vs 3D

- Problems can be defined in 2D and 3D

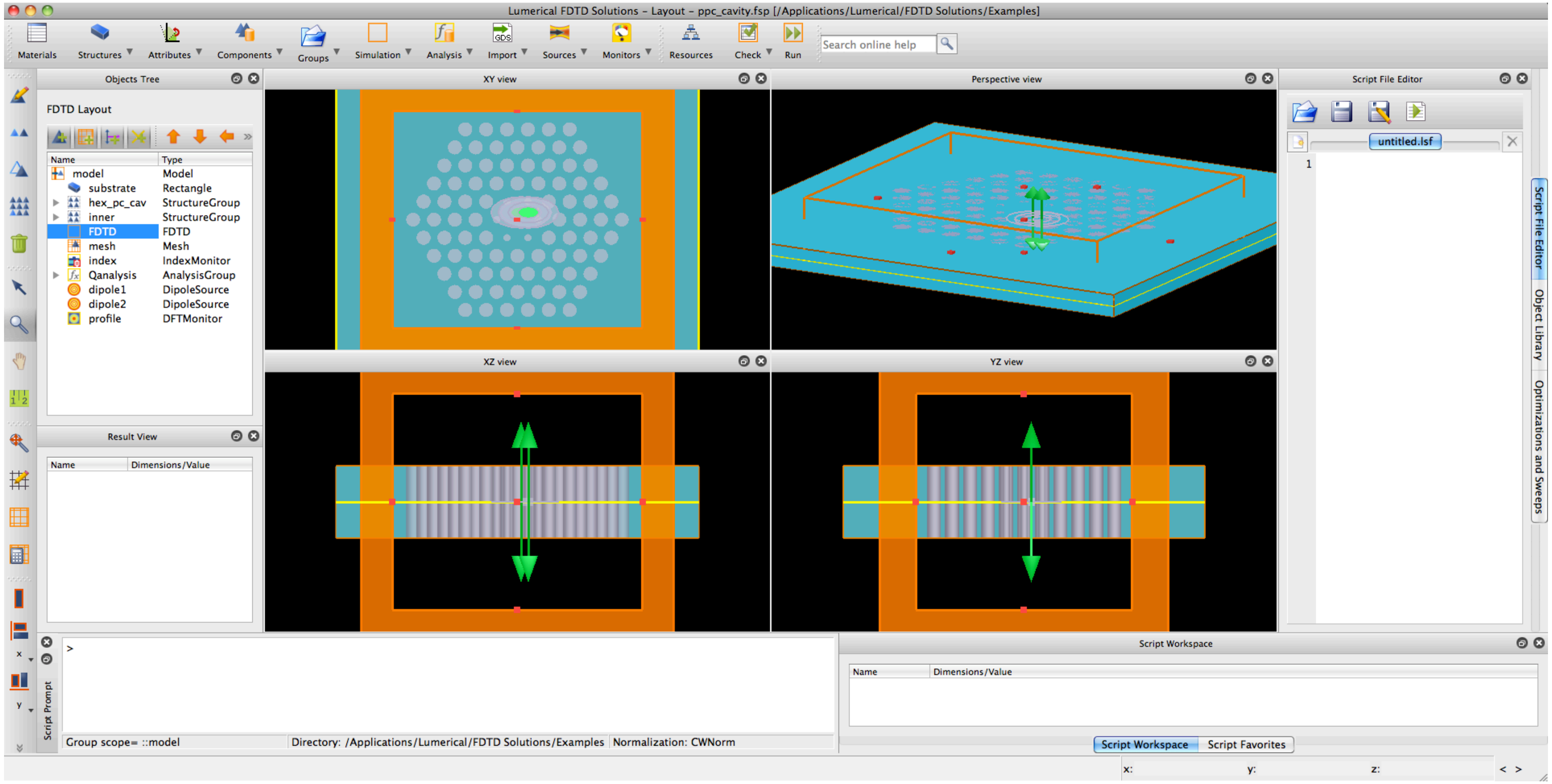




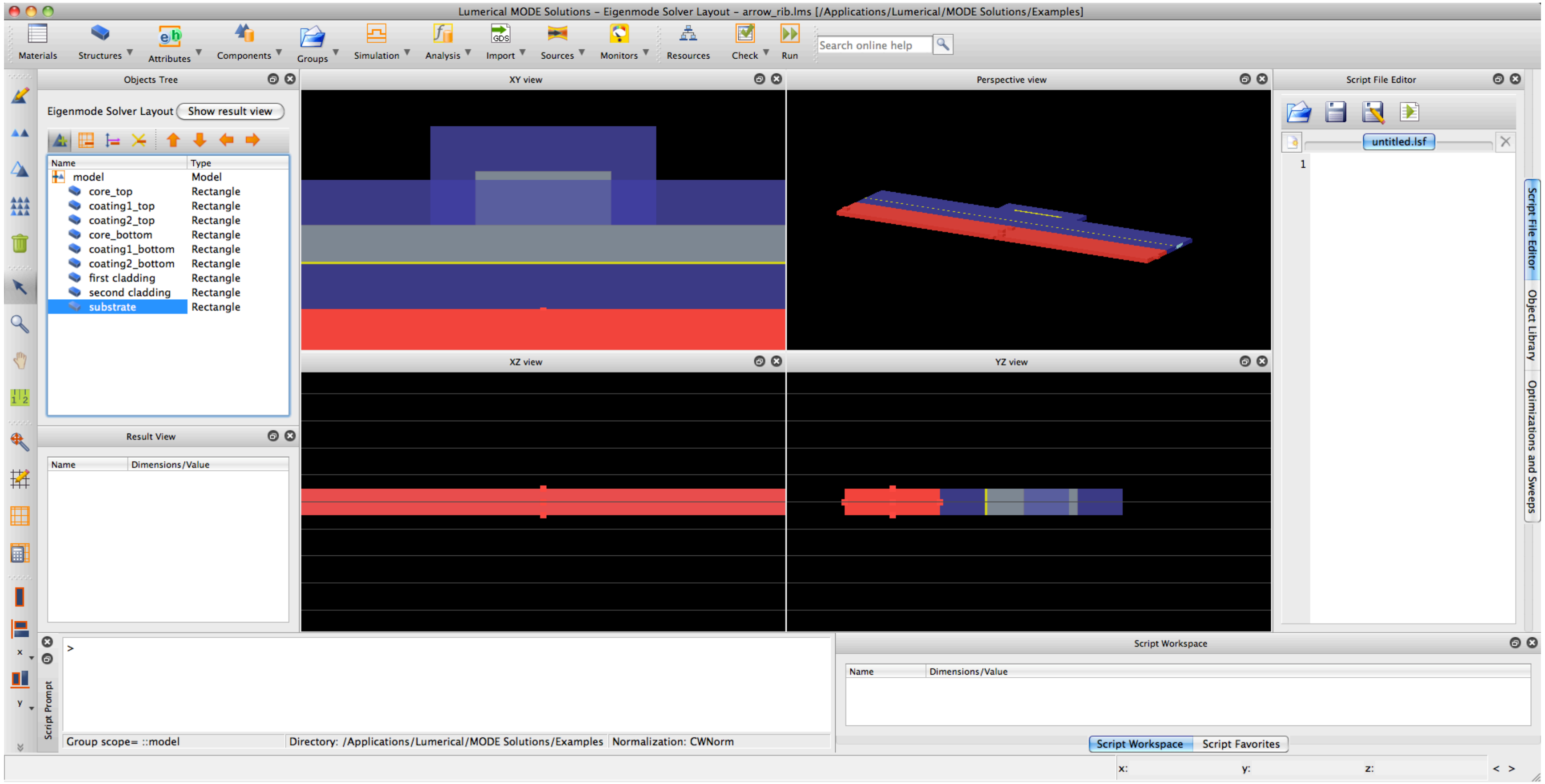
Computationally expensive

| | 2D | 3D |
|---|---|---|
| Memory requirements | $\sim A \cdot (\lambda/dx)^2$ | $\sim V \cdot (\lambda/dx)^3$ |
| Simulation time | $\sim A \cdot (\lambda/dx)^3$ | $\sim V \cdot (\lambda/dx)^4$ |

# Lumerical FDTD Solutions

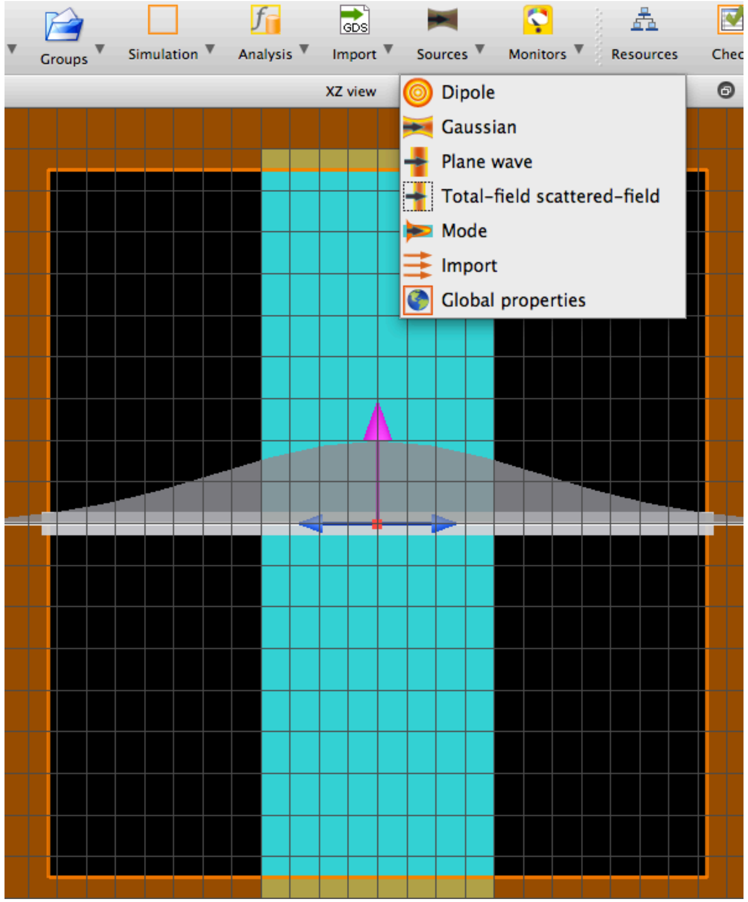


# Lumerical MODE solutions



# Sources

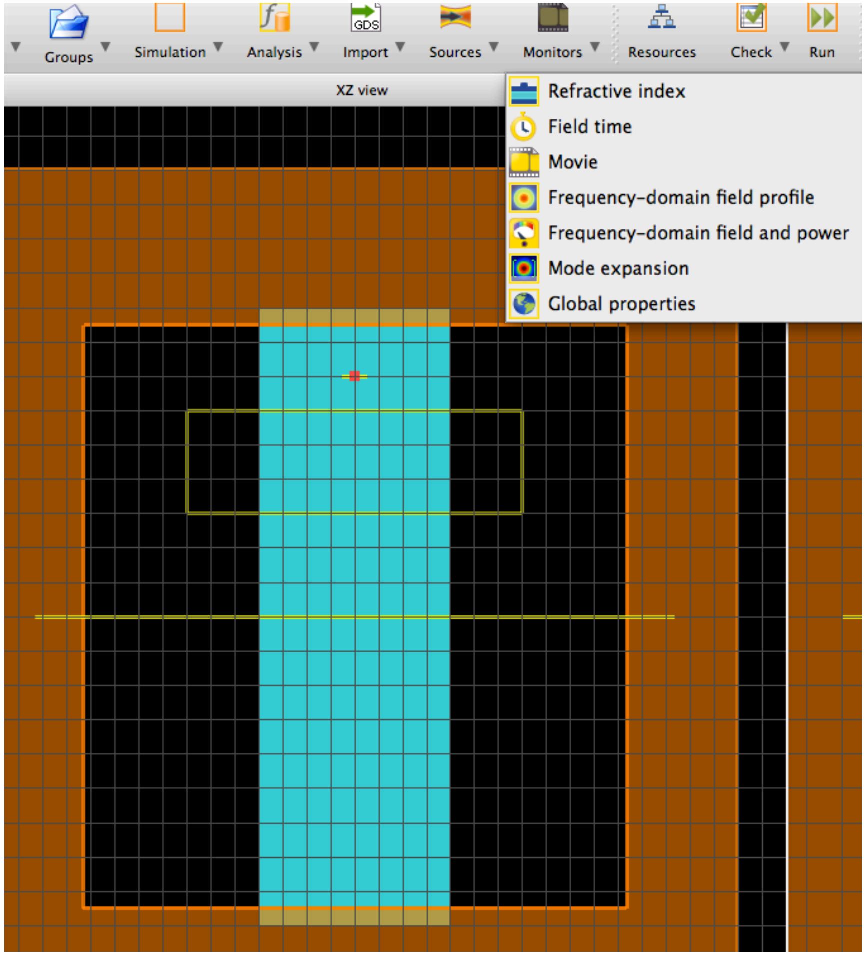
- Dipole
- Gaussian beam
- Plane wave
- Mode
- Total field/scattered field
- Large numerical aperture source
- User defined




Careful with boundaries!

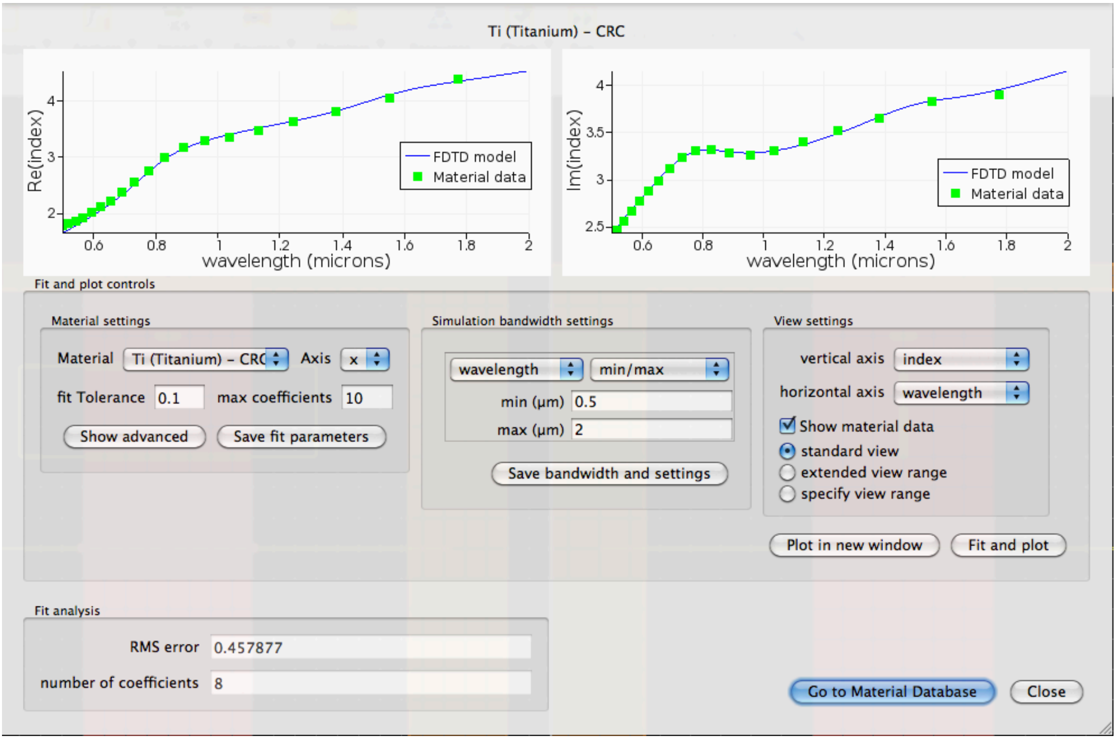
# Monitors

- Refractive index:
  - To verify the design
- Time monitors:
  - Record fields as a function of time
- Frequency monitors:
  - Perform Fourier transforms during simulation
- Movie monitors:
  - Create mpg movies → Qualitative analysis



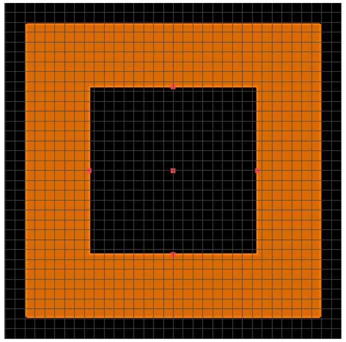
# Materials

- You can use the material database or define your own

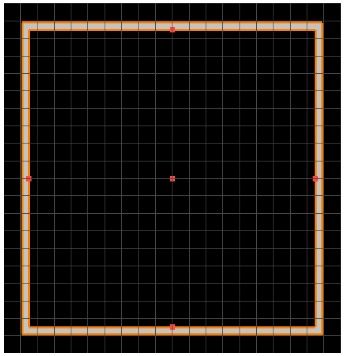



- Scripts can be used to define custom refractive index gradients and structures
- Many types of materials supported (dielectric, conductive, plasma, nonlinear…)

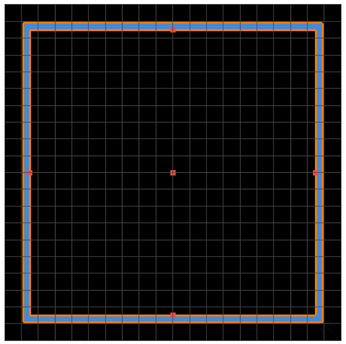
# Boundaries



- PML
  - Absorbs incident fields
  - Use when fields are meant to propagate away from the simulation

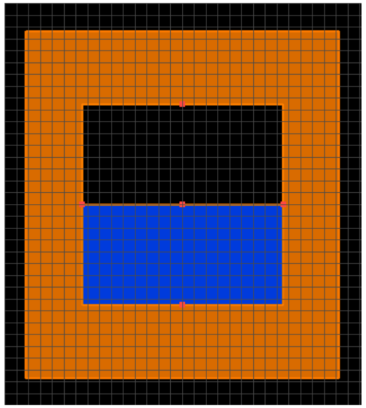


- Metal
  - Perfect metal boundaries
  - 100% reflection, 0% absorption

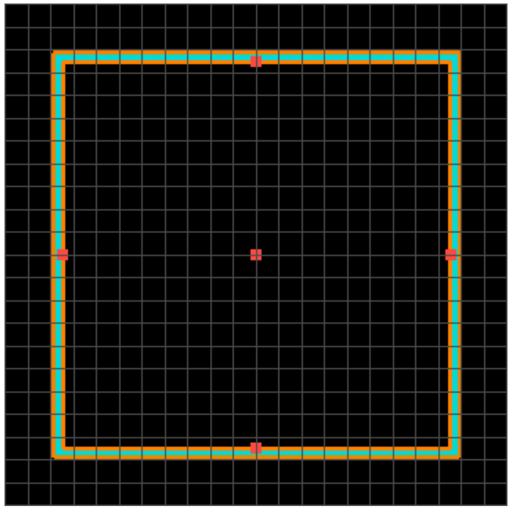


- Periodic
  - Affects both structures and fields

# Boundaries



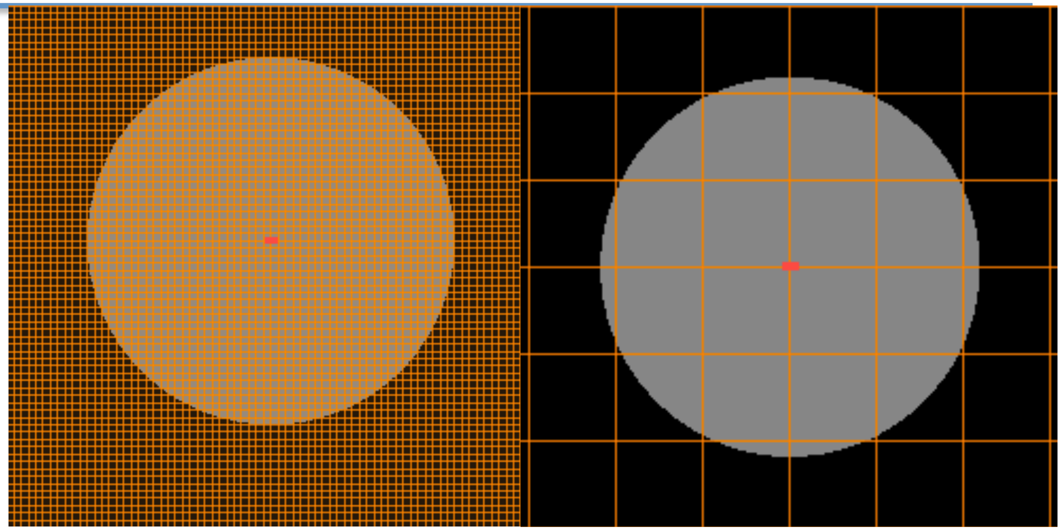
- Symmetric/Antisymmetric
  - Can reduce computational load



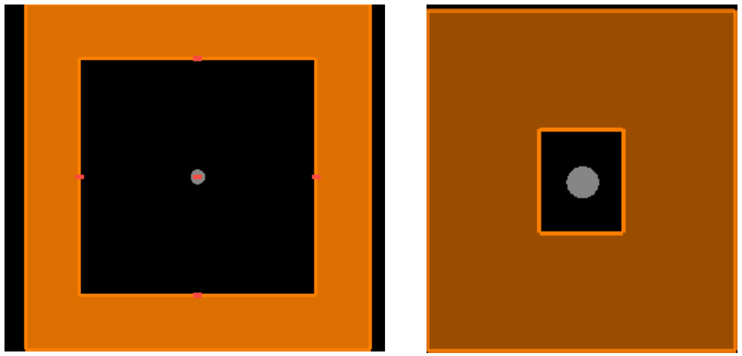
- Bloch
  - Similar to periodic structures
  - Use bloch when the plane wave source is at non-normal incidence

# Simulation parameters

- Mesh size



- Simulation area



- Simulation time

Finer simulation → More accurate results → Higher computational load

# Minimizing computational load

- Only use 3D when necessary!

- Take advantage of symmetries

- Perform convergence studies:
  - Start with rough simulation parameters
    - Simulation time
    - Mesh size
    - Simulation area
  - Use finer simulation parameters
  - Compare the results
  - Repeat until the results converge (more accurate simulations provide the same results)

# Other advanced features

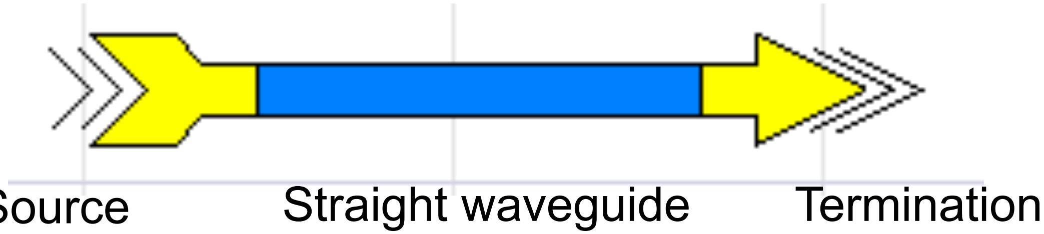
- Scripting
- Parallel computing
- Optimization and parameter sweeps

# Let’s see some examples!

# 3. CIRCUIT APPROACH: ASPIC

# Circuit simulation

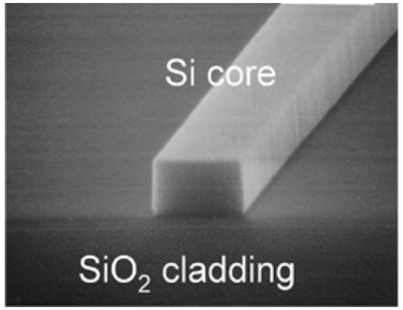
- Instead of simulating a mesh, each element of the circuit is modeled as a whole
- For example, a straight waveguide can be modeled according to different parameters

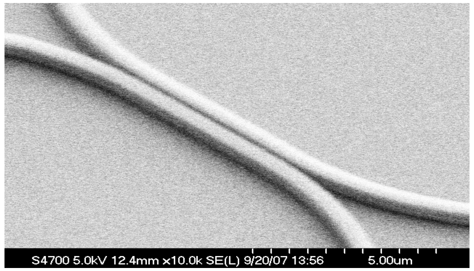


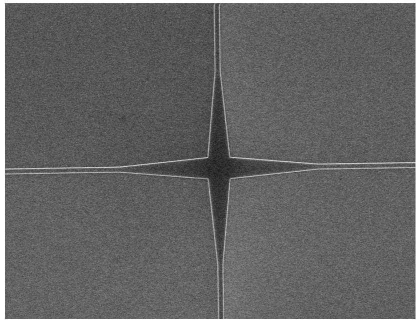

- *Losses*
- *Temperature dependence*
- *TE / TM (birefringence), multimode*
- *Mode coupling (high order modes, polarization rotation)*
- *Input/output modal mismatch*
- *Parasitic interactions, backscatter*
- *nonlinearity*

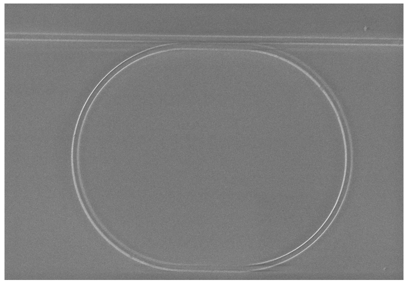
# Circuit simulation

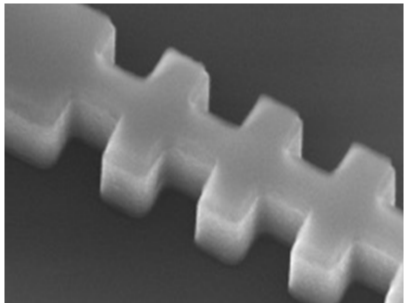
- Most circuits can be build combining basic building blocks (BB).

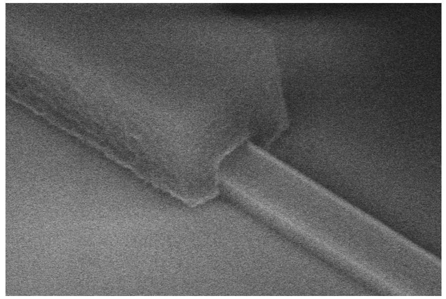


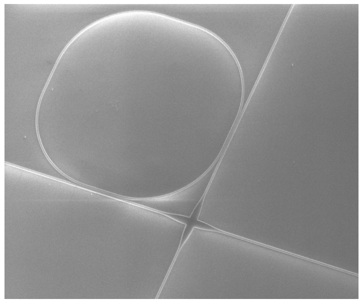



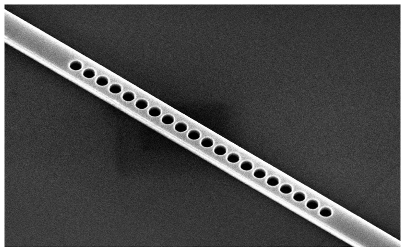



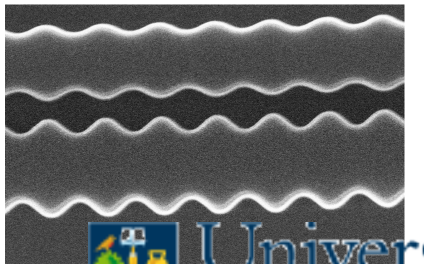




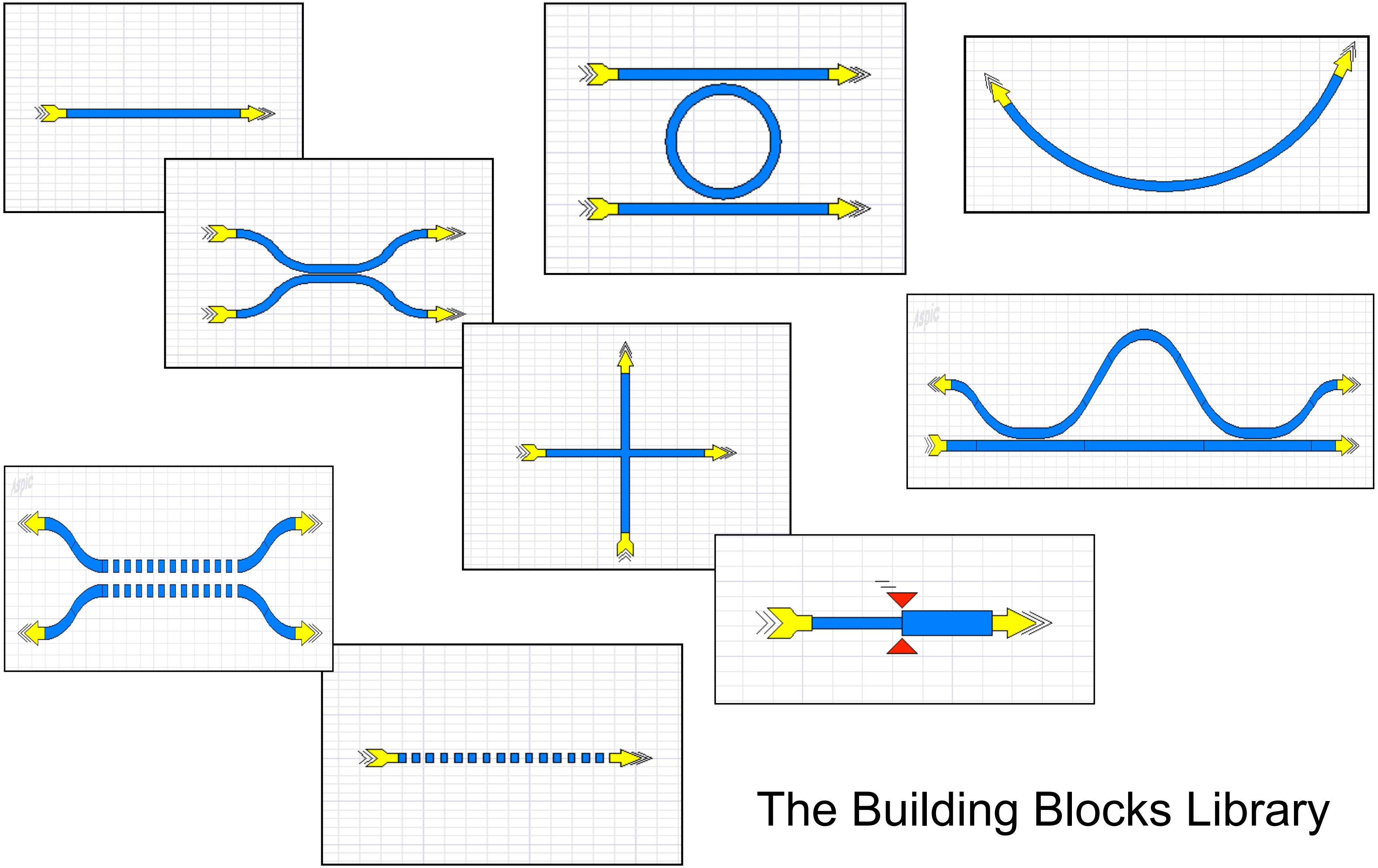








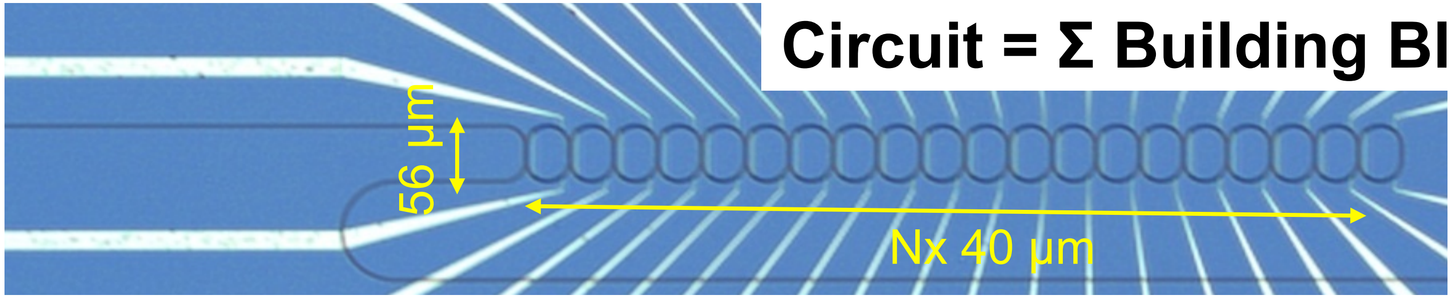
# Circuit simulation



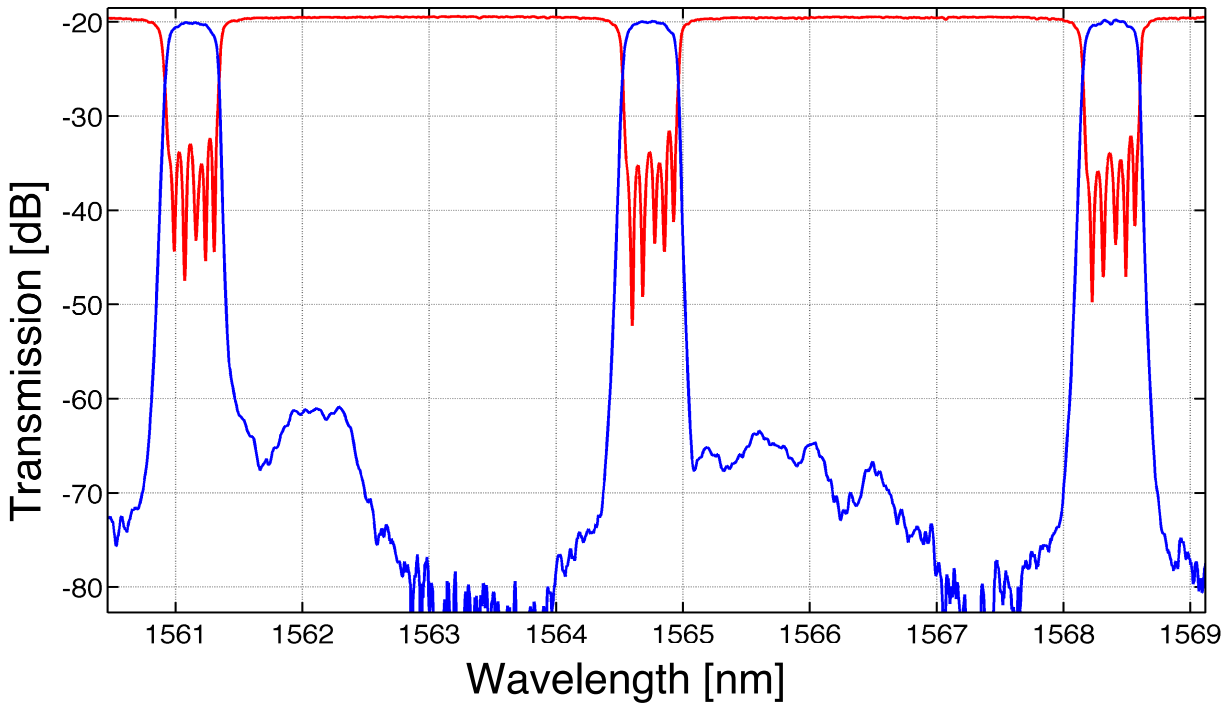
## The Building Blocks Library

# Circuit simulation

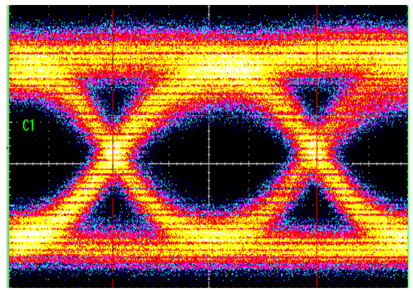
Circuit = Σ Building Blocks



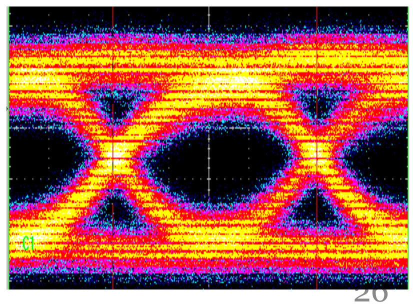

Tunable delayline 0÷8 bit @ 10…100 Gbit/s




OOK, 10 Gbit/s

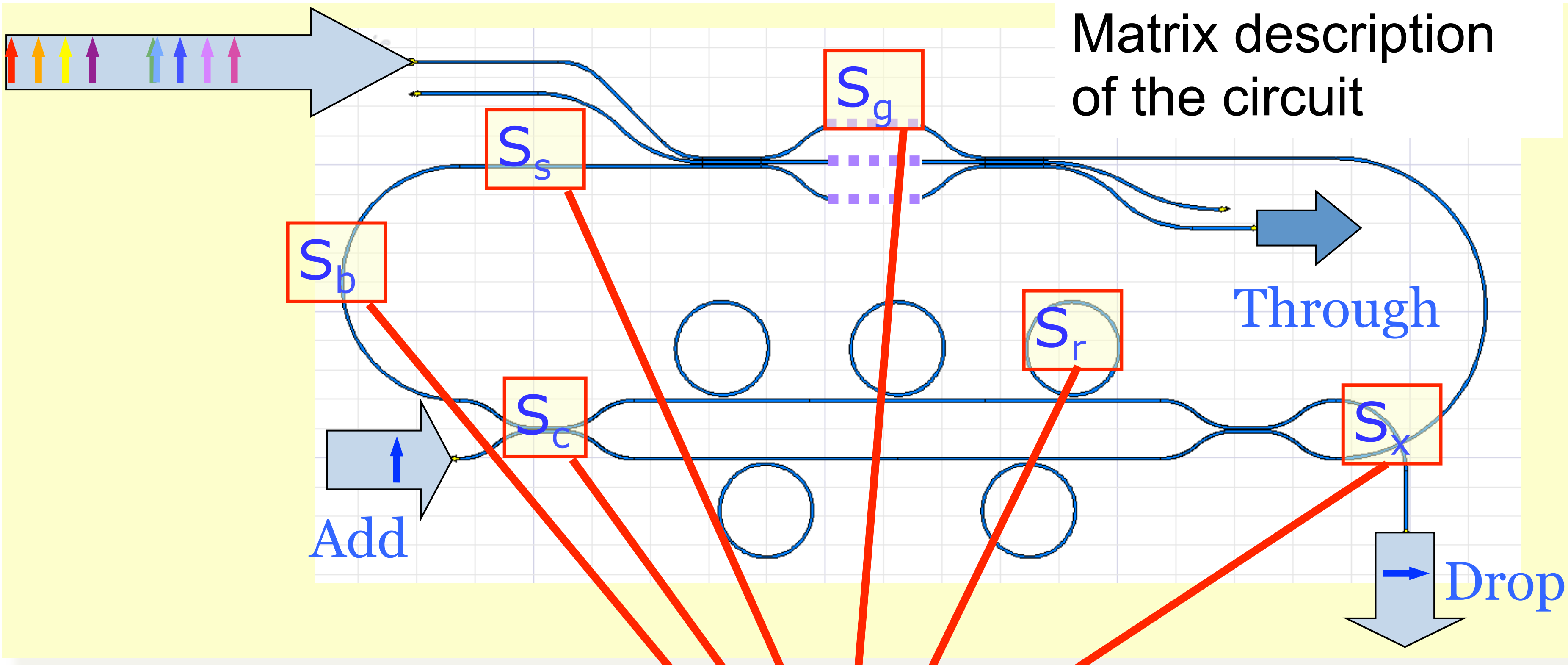


Delay: 50 ps



F. Morichetti et al, **Nature Comm**, May 2011

# Circuit simulation




$$Out=[S]\cdot In$$

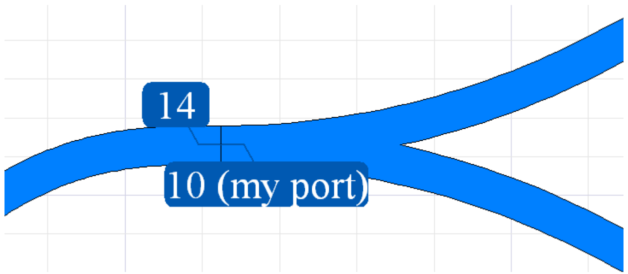
# The models and the BBs

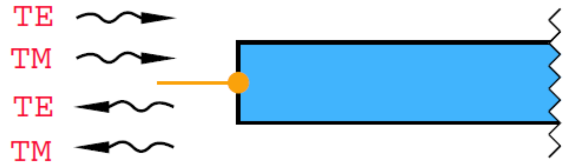
**Organized in Libraries:**

- Generic Aspic Library
- Oclaro Library
- HHI Library
- …

**Ports:**

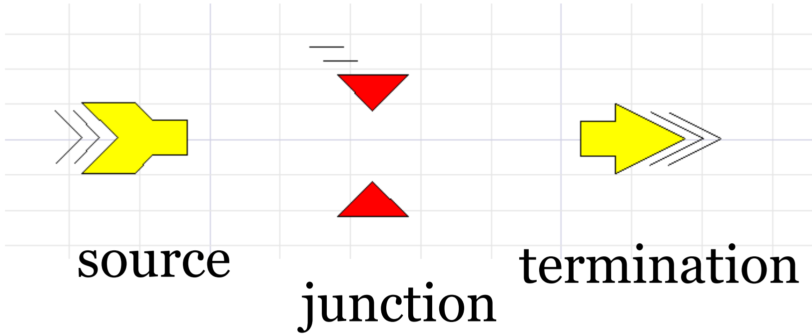
- the only accessible point of a BB
- each port has a number and name
- numbered anti-clockwise
- at least 1 port
- bidirectional
- internal and external
- only two ports can be connected
- up to four fields per port



# The models and the BBs

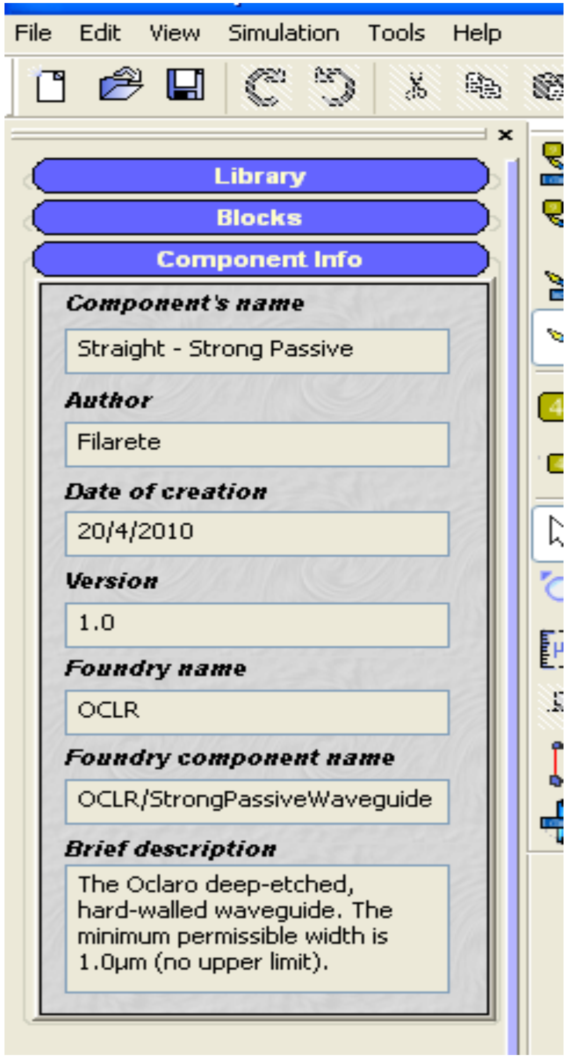
- Physical BB (most)
- Virtual BB




- Sources and terminations are matched (no reflections)
- A source acts also as a termination
- All the sources are perfectly coherent (same wavelength and phase)
- Floating ports are not allowed

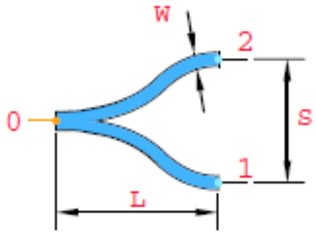
# The models and the BBs

## Each BB has HELP and INFO






Y Branch




DESCRIPTION

Y Branch models a component that divide the optical power into two output branches or, conversely, can combine the optical power from two input branches. The model includes attenuation and dispersion of both TE and TM modes.

INPUT

*Display*

| Type | Name | Symbol | Description |
|---|---|---|---|
| Edit | Length [$\mu m$] | $L$ | Length of Y branch |
| Edit | Width [$\mu m$] | $W$ | Waveguide width |
| Edit | Separation [$\mu m$] | $S$ | Output waveguides separation |

*General Parameters*

| Type | Name | Symbol | Description |
|---|---|---|---|
| Radio | Geometrical Length | - | Enables/Disables geometrical length input |
| Radio | Optical Length | - | Enables/Disables optical length input |
| Edit | Geometrical Length$_{0j}$ [$\mu m$] | $L_{0j}$ | Geometrical length of arm $0j$ ($j = 1, 2$) |
| Check | Attenuation | - | Enables/Disables attenuation |
| Check | Dispersion | - | Enables/Disables chromatic dispersion |

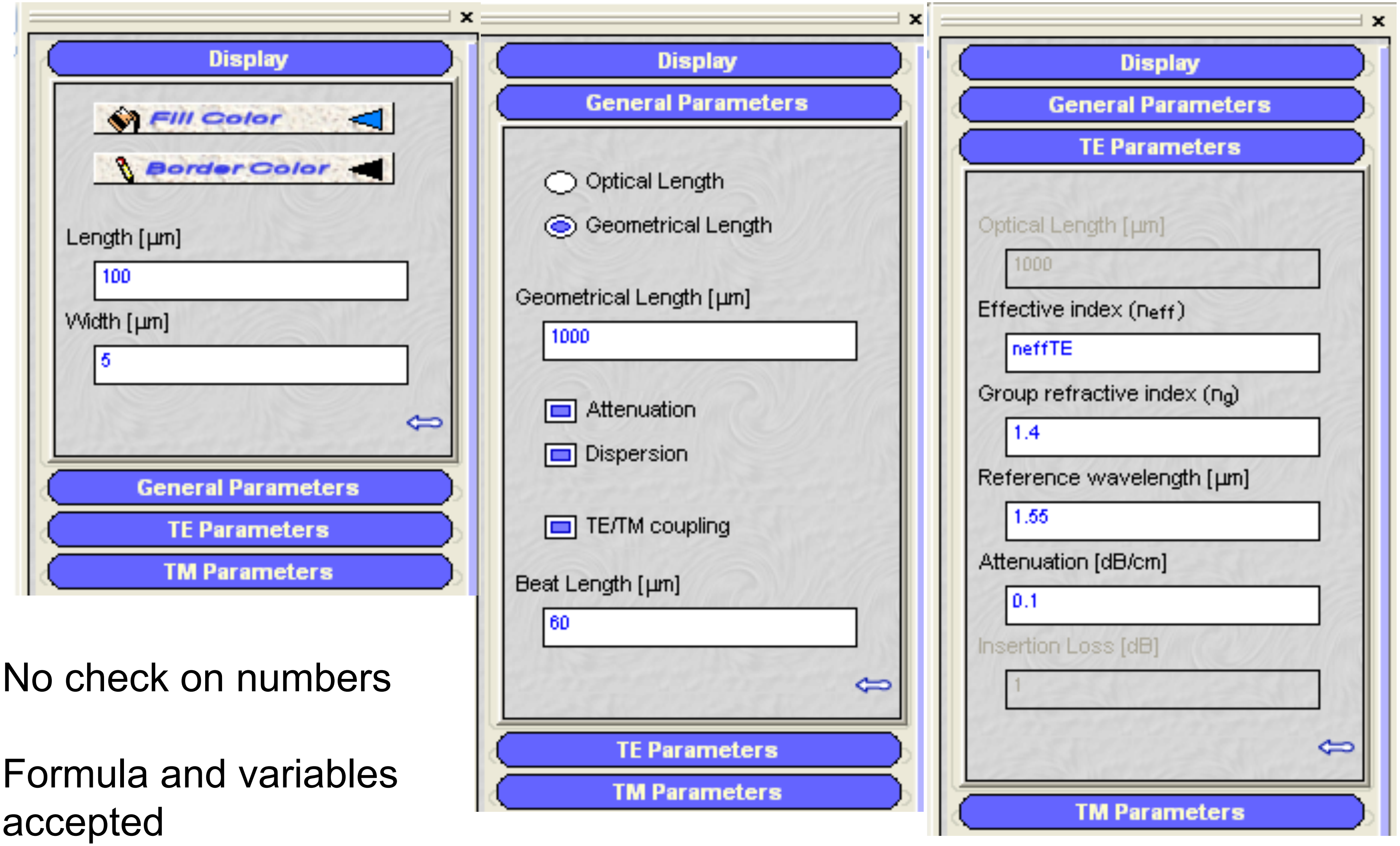
*Specific Parameters for TE and TM modes*

| Type | Name | Symbol | Description |
|---|---|---|---|
| Edit | Splitting Ratio | $K$ | Power splitting coefficient |
| Edit | Optical Length$_{0j}$ [$\mu m$] | $L_{0j}$ | Optical length of arm $0j$ ($j = 1, 2$) |
| Edit | Effective index | $n_{eff}$ | Effective refractive index |
| Edit | Group effective index | $n_g$ | Group refractive index |
| Edit | Reference wavelength [$\mu m$] | $\lambda_0$ | Central wavelength for $n_{eff}(\lambda)$ calculation |
| Edit | Attenuation [dB/cm] | $\alpha$ | Waveguide attenuation |
| Edit | Insertion Loss [dB] | $I_{L_{0j}}$ | Insertion loss of the arm $0j$ ($j = 1, 2$) |

MODEL

Y Branch models a branch that split the optical power into two output arms or combine the optical power from two input arms. Two waveguides can have different lengths but identical optical characteristics ($n_{eff}$, $n_g$ and $\alpha$).

# The models and the BBs

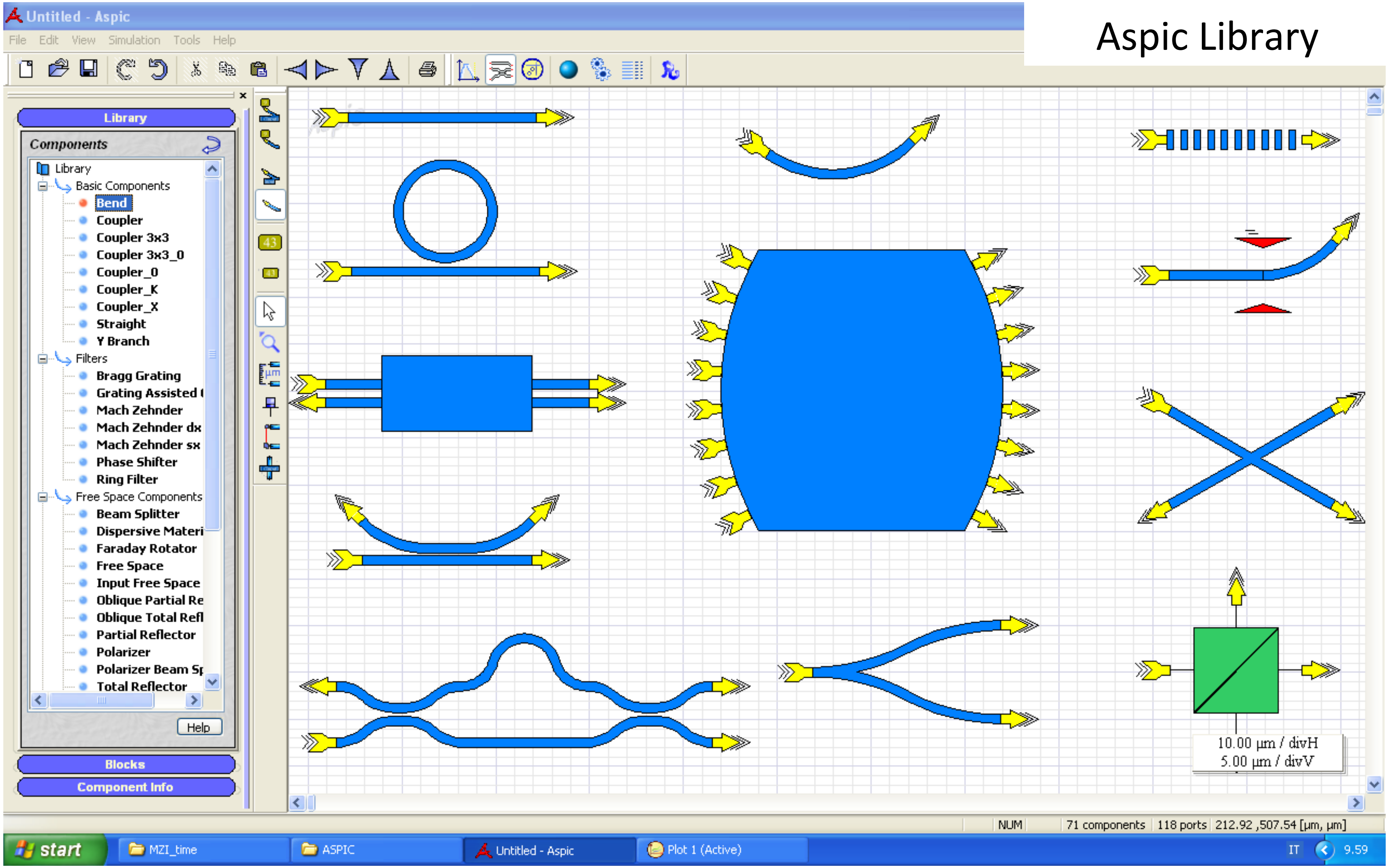



No check on numbers

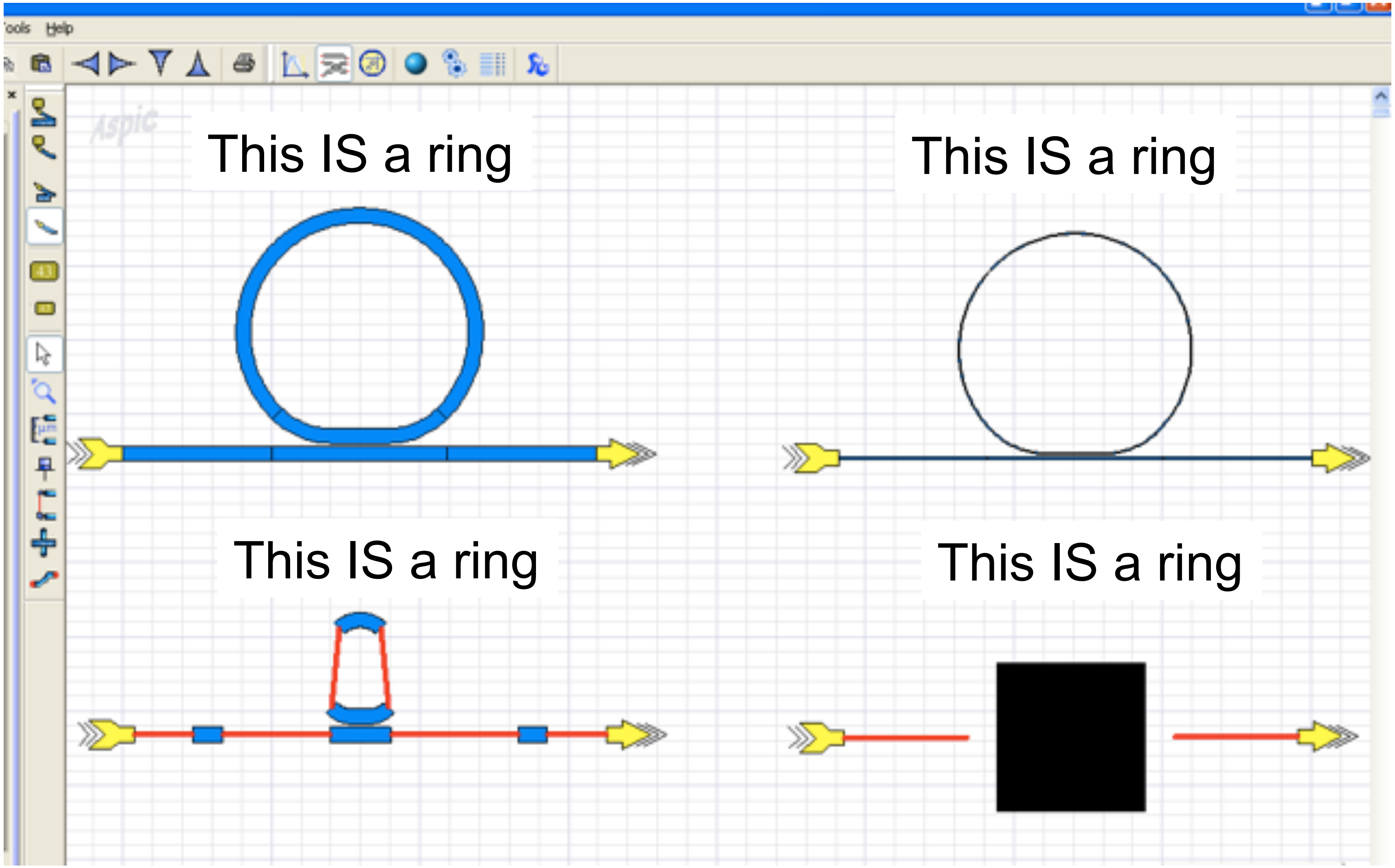
Formula and variables accepted

# The models and the BBs

Aspic Library



# Aspic's CAD environment



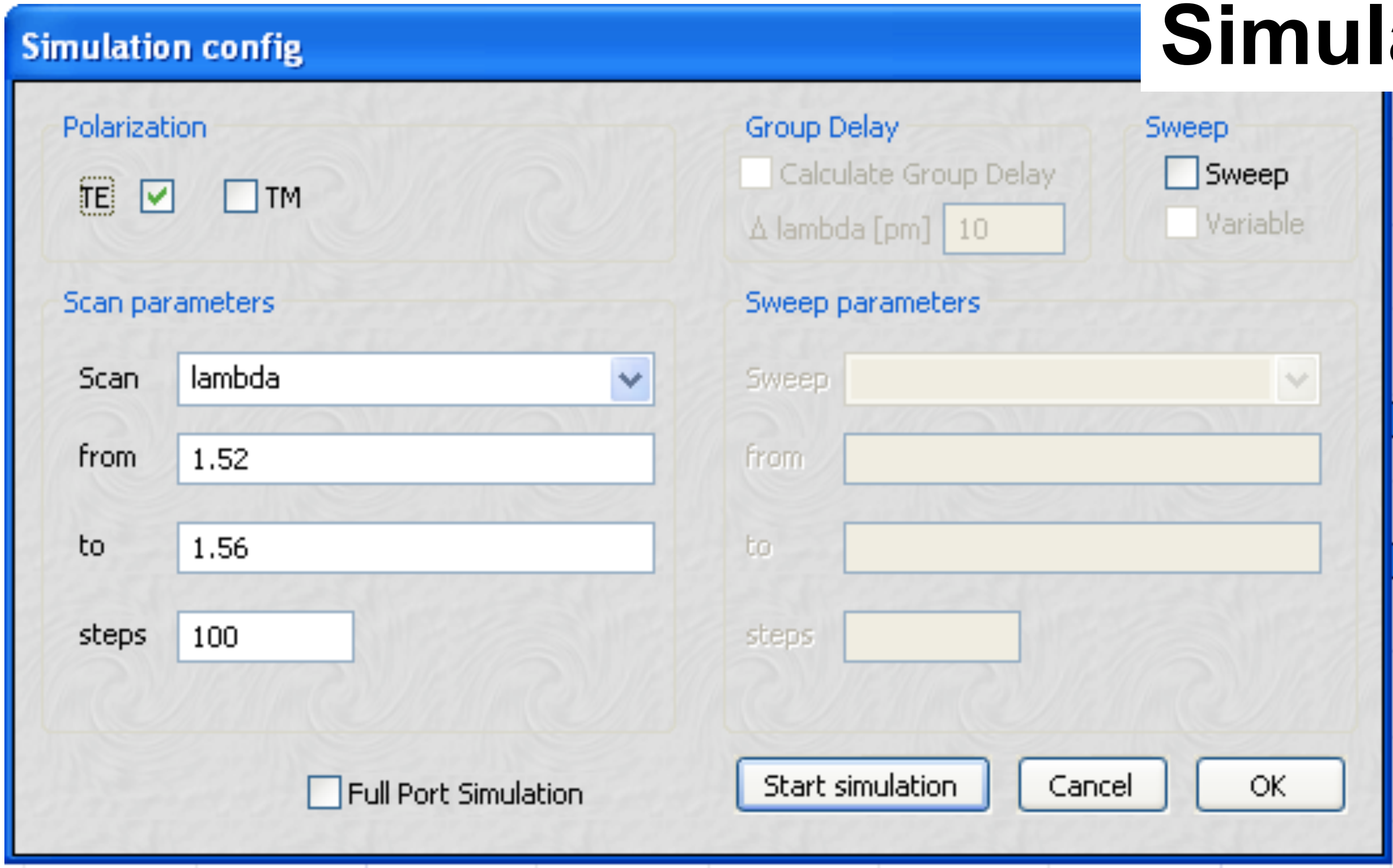
# Aspic's CAD environment

Default: Results are available ONLY at the external (accessible) ports

FULL ports: Results are available at EVERY port

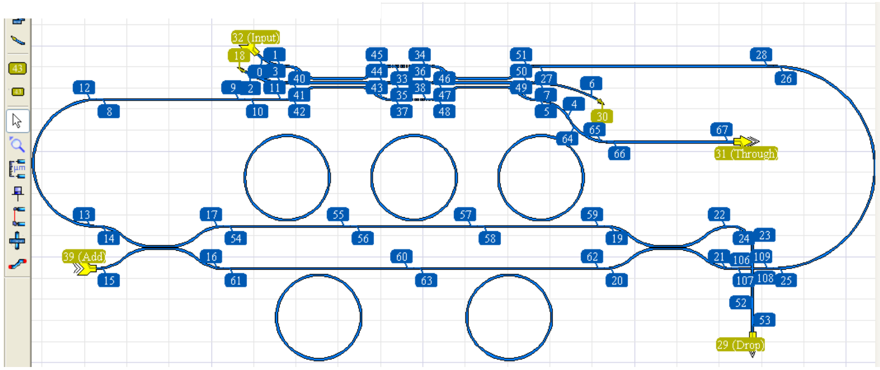
Group Delay: Calculate the group delay when scan variables

## Simulation

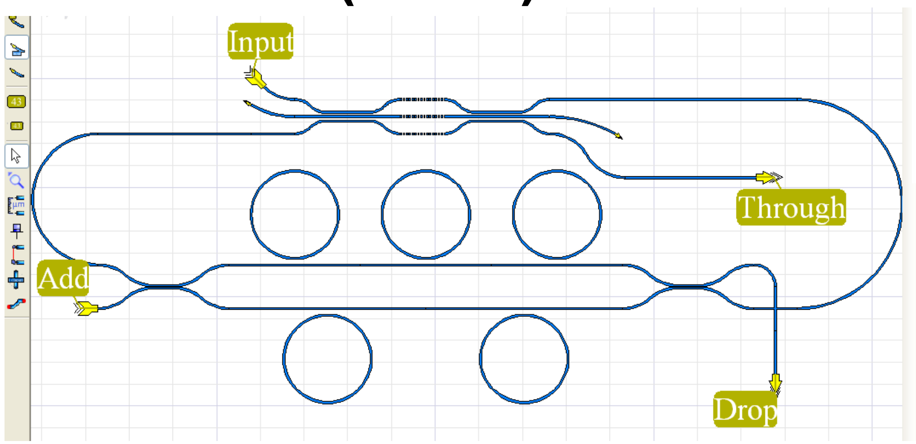


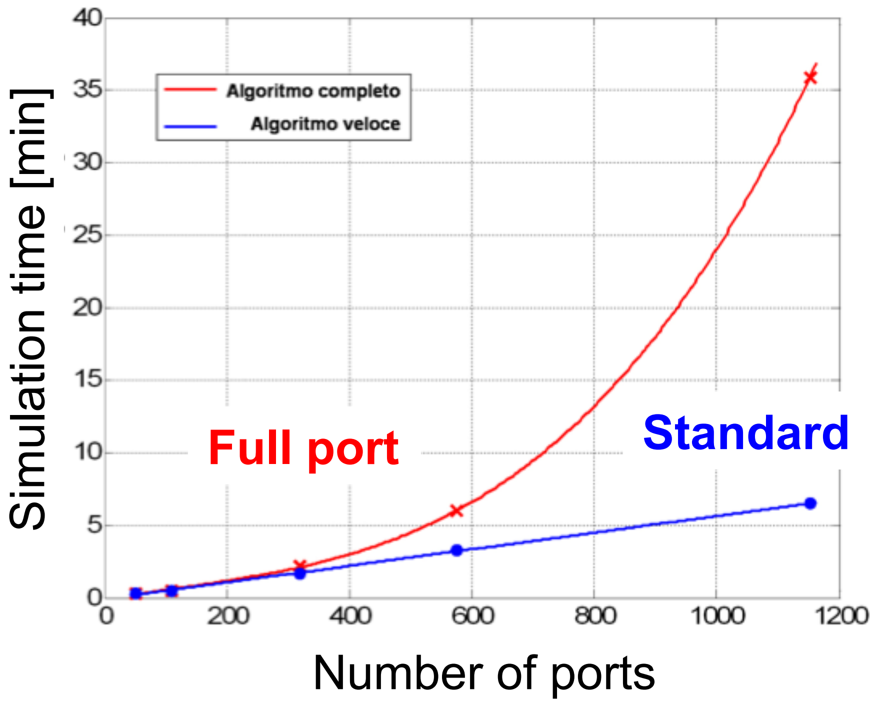
# Aspic’s CAD environment

## FULL PORTS



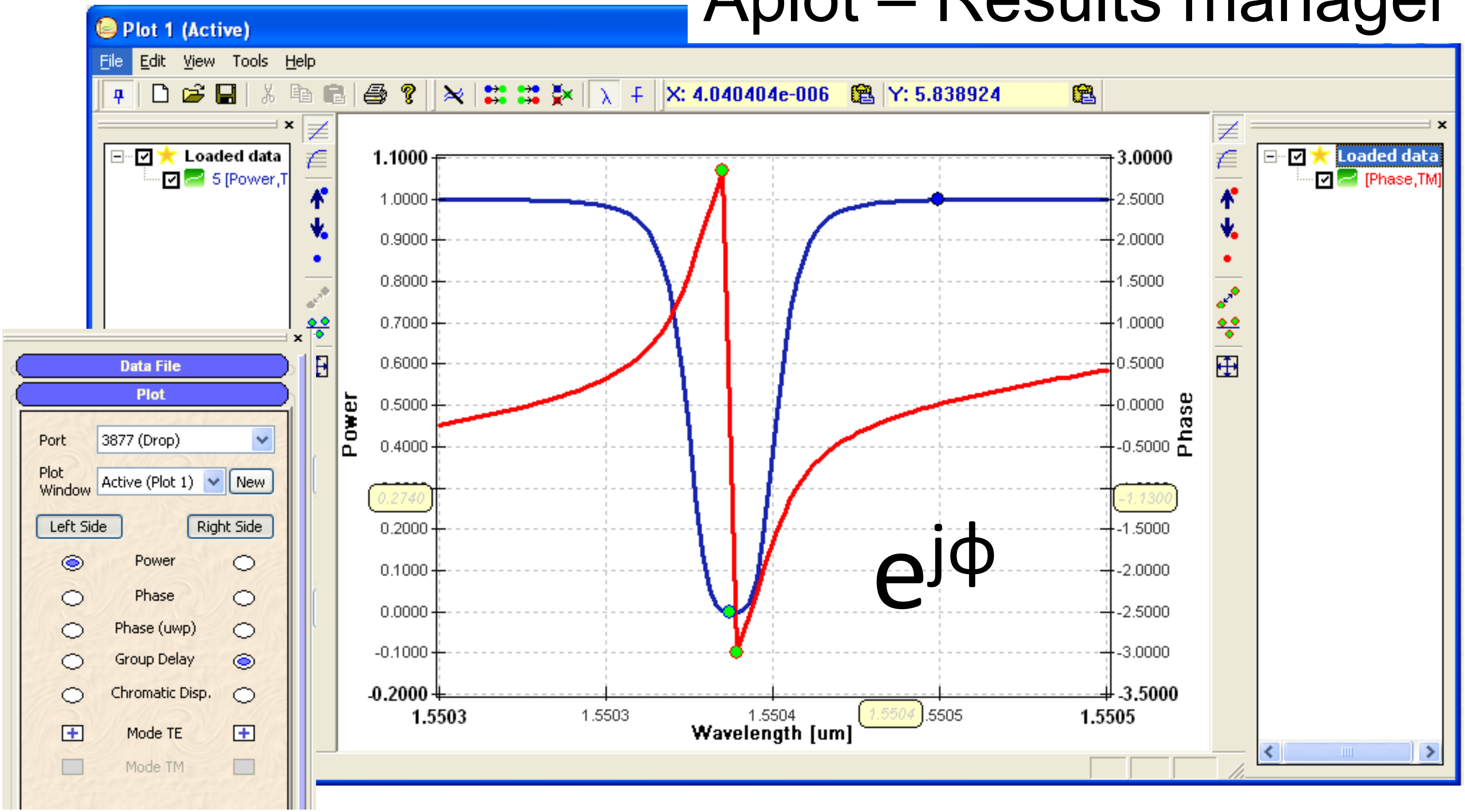
## STANDARD (FAST)



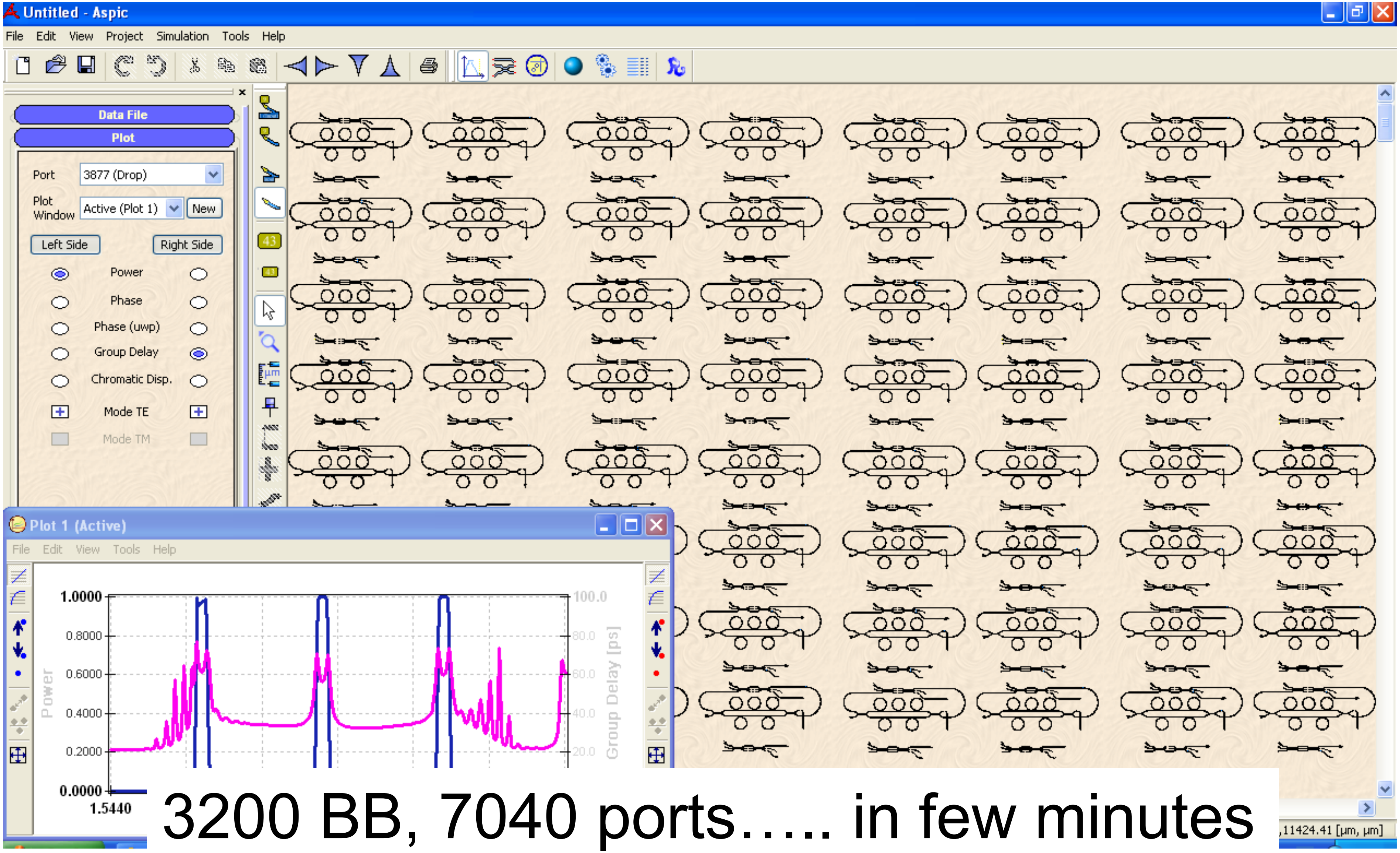



# Aspic's CAD environment

Aplot – Results manager



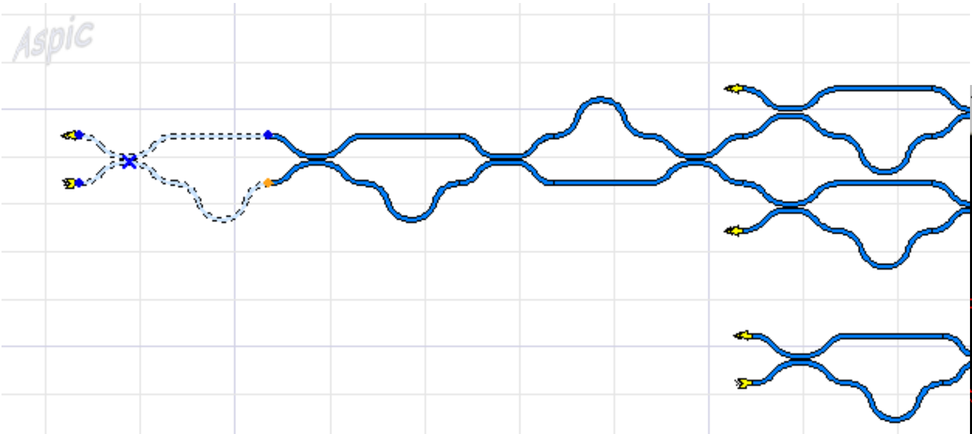
# Aspic's CAD environment

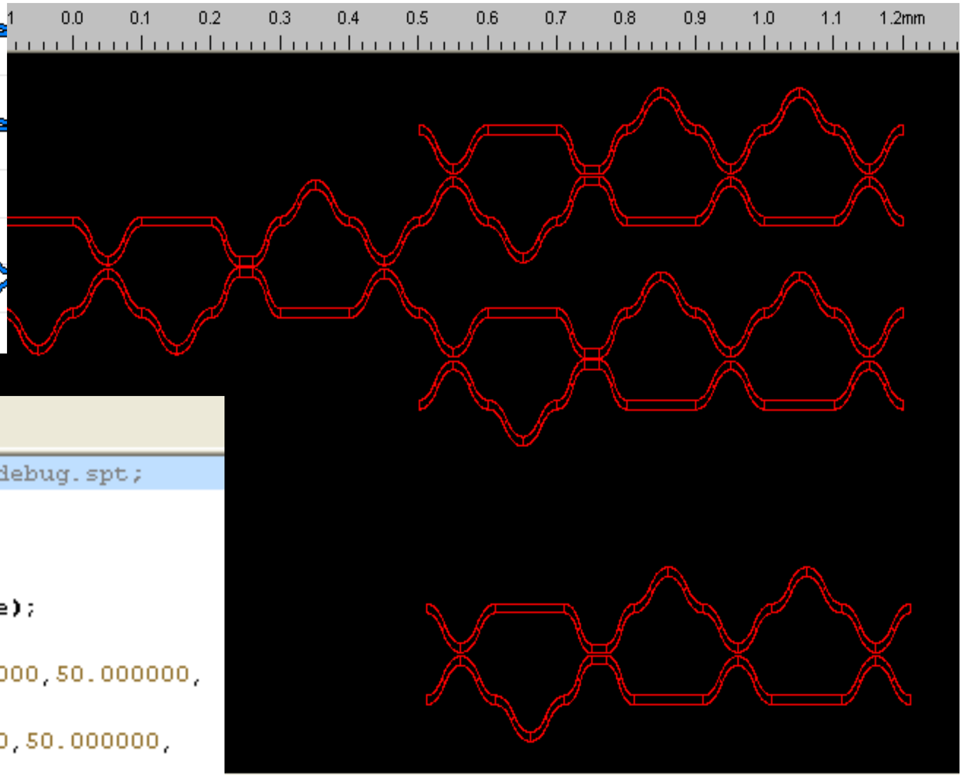



3200 BB, 7040 ports….. in few minutes

# Export to Mask Engineer

## Export through pdaFLOW to include foundry info



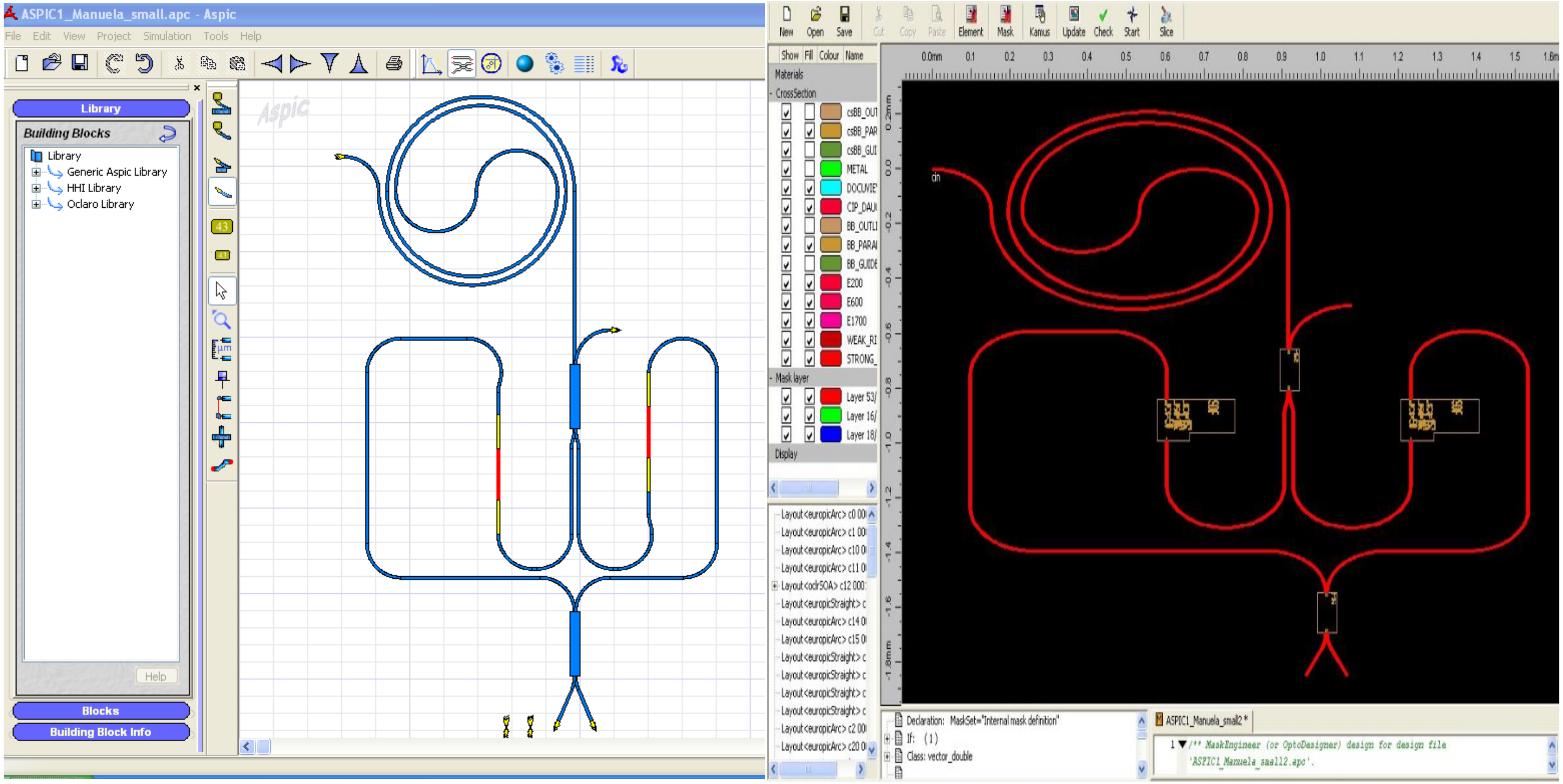




Interleaver

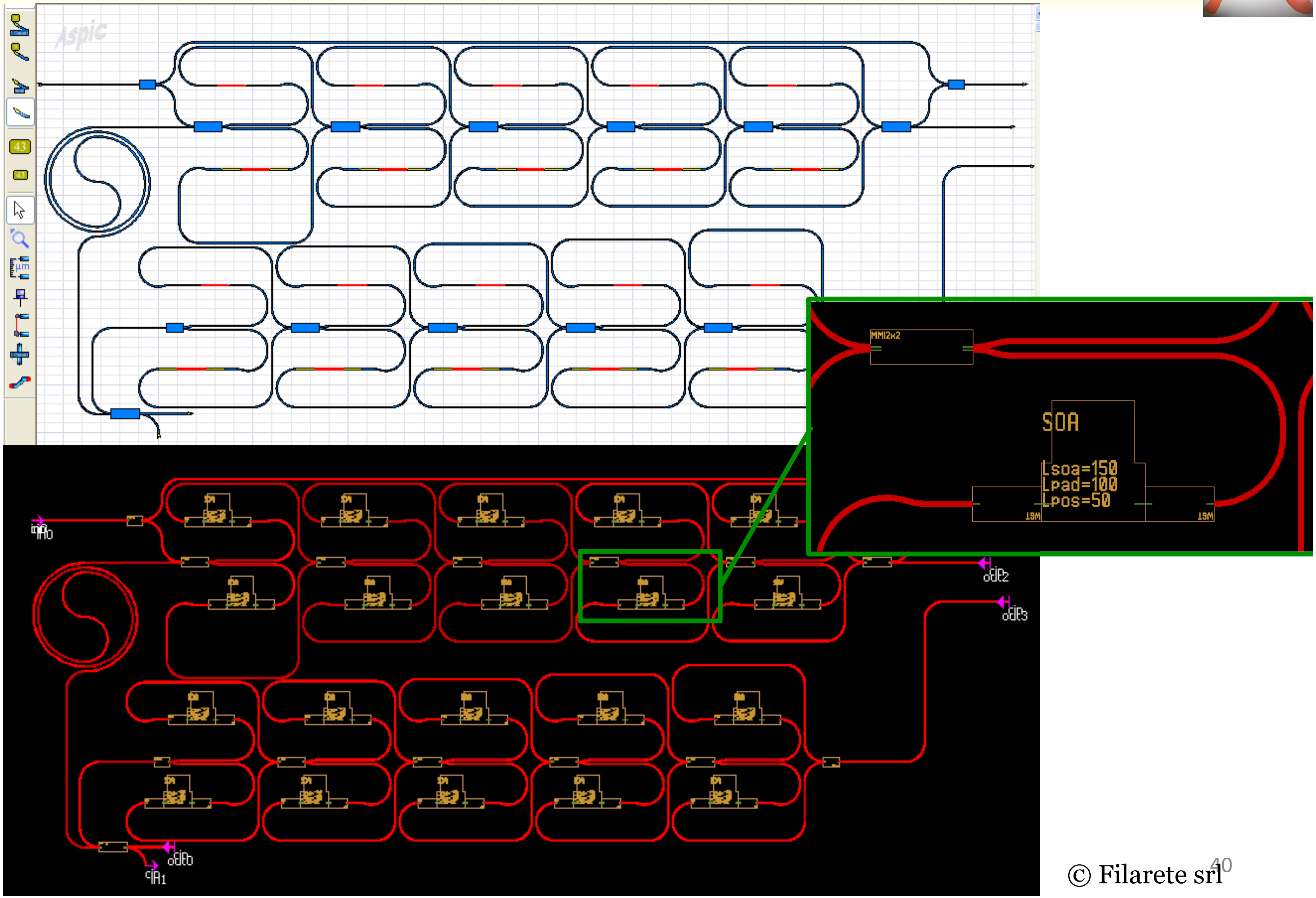
```
#include C:\Program Files\PhoeniX Software\Aspic\Data\\debug.spt;
#include C:\Program Files\PhoeniX Software\Aspic\Data\\aspic_library.spt;
dsp::clearInfoWin();
mask::CScreate("cs_aspic",filter,RGB(255,0,0),true,false);
mask::CSselect("cs_aspic");
ml::aspic_mach_zehnder_dx(P0->[0,0,-180.000000]:200.000000,50.000000,5.000000,2.000000) component_28;
ml::aspic_mach_zehnder_dx(P2->component_28@P1:200.000000,50.000000,5.000000,2.000000) component_20;
ml::aspic_coupler(P2->component_20@P1:100.000000,20.000000,5.000000,1.000000,50.000000) component_24;
```

# Export to Mask Engineer

## Export through pdaFLOW to include foundry info

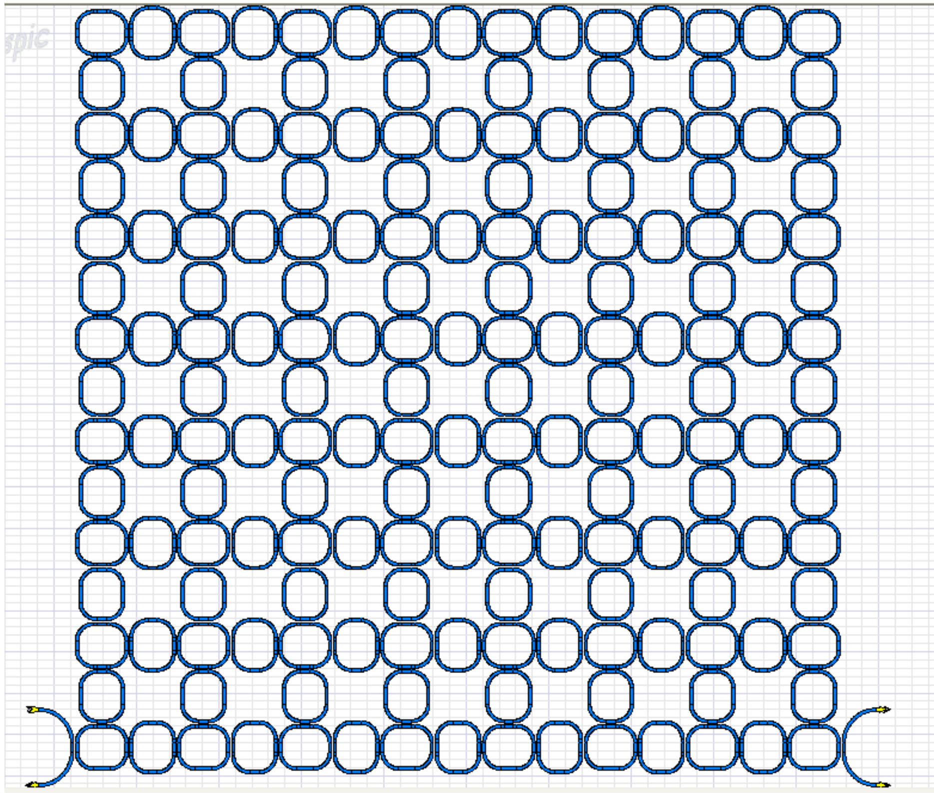


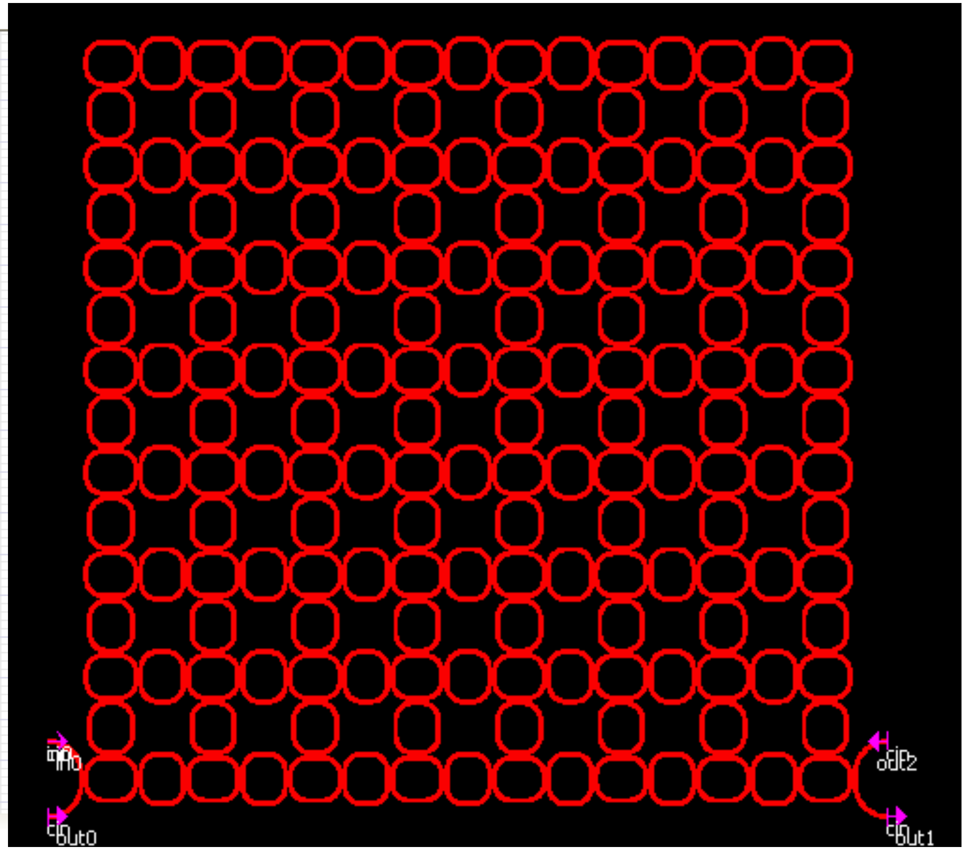
# Export to Mask Engineer



# Export to Mask Engineer

Robust optical delay lines with topological protection
M. Hafezi, NATURE PHYSICS, VOL. 7, NOVEMBER 2011





1656 Building Blocks, 3760 ports, 18 nested variables

# Let’s see some examples!