OptoDesigner 5.1.4

Phase-aware Connectors

PDKs now have a capability to store information on the dependence of effective index and group index on the wavelength, width, and radius of a waveguide. This information is tied to a mask cross-section, so the results are different depending on whether you use, for example, a deeply-etched waveguide or a shallowly-etched waveguide. Moreover, the software is capable of integrating the effective or group index over the path of a waveguide, taking into account the local radius and width, to obtain the true optical or group length of arbitrarily curved and tapered waveguides.

The availability of this information opens up a range of layout implementations that were previously hard or impossible to do. First of all, one can query the optical or group length of any waveguide (or concatenation of multiple waveguide sections), which can be used to analyze the behavior of a circuit. Secondly, optimization routines can be employed to enforce a waveguide to have a given optical or group length; for example, the hump you see in Figure 1 is calculated by moving it up and down until the total group length is correct. Users can employ this functionality to create any kind of phase-sensitive waveguides themselves, or use the built-in versions. The current suite of true phase-sensitive connectors and waveguides consists of sine connectors, S-bend connectors, curve connectors, and various spiral delay lines.

The picture below shows a Mach-Zehnder filter that has one input parameter – the free spectral range (FSR) of its transmission. To obtain the desired value of the FSR, the group length difference between the two arms needs to be equal to the speed of light divided by the FSR. The very simple script that builds this structure places the two MMIs and the bottom waveguide and queries the bottom waveguide for its group length. Then, it places a phase-sensitive connector between the top ports of the MMIs, whose group length is defined to be equal to the bottom waveguide group length plus the group length difference calculated above.

For more information about phase-aware connectors, see the OptoDesigner User Manual (for customers).

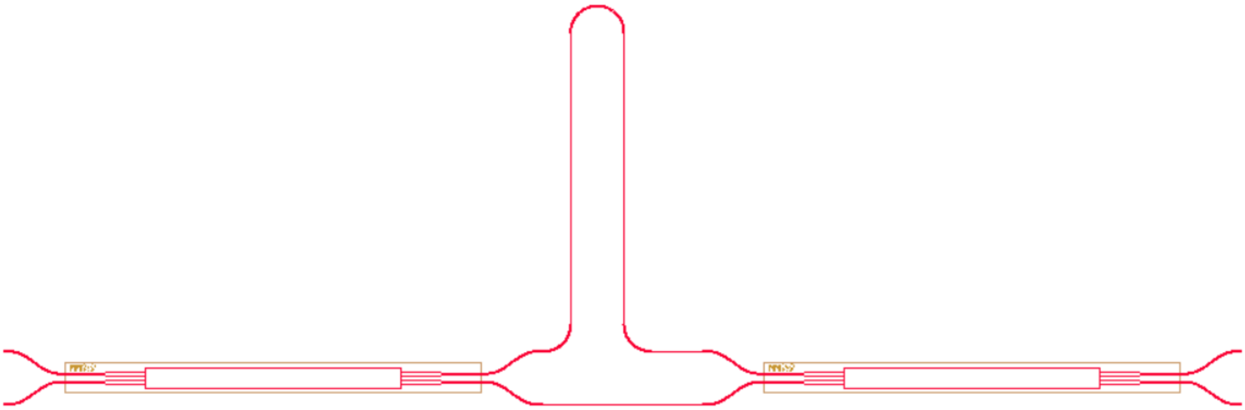

Figure 1: Phase-aware connector in a Mach-Zehnder filter

Boundary ports

Boundary ports are ports in layouts denoting the corners and edges of their bounding boxes. Typically, eight boundary ports exist – on the four corners and in the middle of the four edges of the bounding box of the layout. The picture below shows these ports.

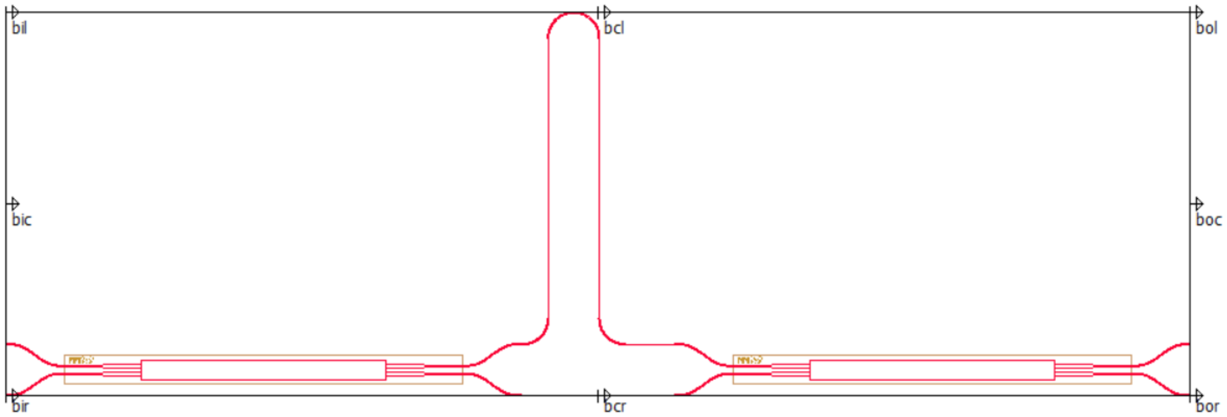

Figure 2: Boundary ports of the Mach-Zehnder filter

Boundary ports can be used for a variety of purposes. A first example would be the stacking of multiple devices close together – for example, if one wants to pack 10 of the above MZIs as tightly as possible above each other, one can connect the bir (*b*oundary *i*nput *r*ight) port of each MZI to the bil (*b*oundary *i*nput *l*eft) port of the previous one – perhaps with a small offset to make sure that the waveguides are not too close. If one then decides to change the parameters of one of the MZIs, which changes the size of its bounding box, all other MZIs will move to accommodate this different size while staying closely packed. A second example will be shown in more detail in the next section, where we will use the boundary port of an MZI to steer the waypoint-assisted waveguide router around the MZI.

When creating your own layouts (Pcells), you can add any port to the layout that you want. In the past, users and built-in elements used to add boundary ports to their layouts manually, which is an error-prone process. Now, all layouts have their boundary ports calculated automatically by the software, which means that they are always available and always correct.

For more information about boundary ports, see the OptoDesigner User Manual (for customers).

Waypoint-assisted and multiport Manhattan Routing

Most of the connectors in OptoDesigner enable easy optical or electrical connections between two components, but have little flexibility to avoid other components that may be located between the ports that you want to connect. Instead of manually inserting a bunch of bends and straights to go around an obstacle, waypoint-assisted Manhattan routing allows one to simply define a few waypoints for the connector to follow.

Manhattan routing is waveguide or metal routing in which all straight waveguides run either horizontally or vertically, connected by corners where the direction changes between horizontal and vertical. The connector 'pxConnectorManhattan' automatically creates a Manhattan router from one port to another; you only have to provide a starting port and end port, similar to the other connectors. However, the pxConnectorManhattan has important extra functionality that allows you to guide the waveguide, for example around obstacles. This works by assigning 'waypoints' – which are just additional ports. The connector then first goes through all waypoints, before reaching its final destination. Only half of the turning points need to be assigned; corners in-between are resolved automatically.

Using the boundary ports of components as a base for waypoints is a good way to avoid obstacles; in the example pictures below, the waveguide that runs above the MZI has two waypoints defined: One connected to the top-left boundary port of the MZI with an offset (Waypoint 1), and one close to the destination grating coupler (Waypoint 2). If the free spectral range of the MZI is decreased from 400GHz to 200 GHz, the 'hump' gets larger – and because the boundary ports then automatically move, the waypoint-assisted Manhattan waveguide automatically extends further upward.

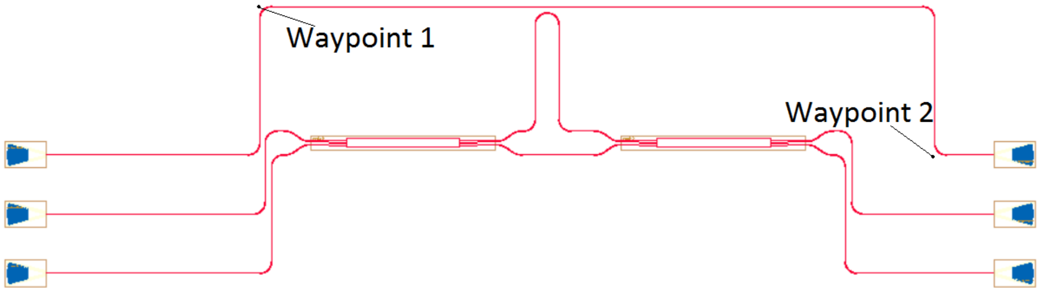

Figure 3: Free spectral range = 400 GHz

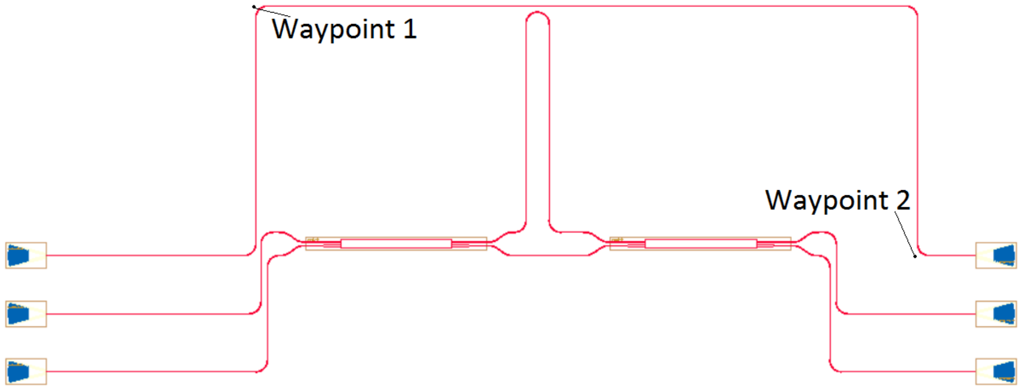

Figure 4: Free spectral range = 200 GHz

Figures 3 and 4 also show multiport Manhattan connectors, in which multiple (in this case 2) connections between components are made automatically and simultaneously. The lower two sets of grating couplers are connected to the MZI by single-line statements, which directly connect all relevant ports together, while making sure that the distance between the waveguides is as required. No more need to manually create each connection in an array of connectors; the software does it for you.

For more information about boundary ports, see the OptoDesigner User Manual (for customers).

Please contact us for free trials and web demos!

Sales@phoenixbv.com