WHITE PAPER

Flexible Path Delay Control With Layout Flexibility in OptoCompiler

Authors

Bowen Wang
Sr. Application Engineer, Synopsys

Arjen Kriswandhi Bakker
Sr. R&D Engineer, Synopsys

Teus Dubbink
R&D Engineer, Synopsys

Thomas Hakkers
R&D Engineer, Synopsys

Francesc Vila
Sr. R&D Engineer, Synopsys

Remco Stoffer
Sr. R&D Engineer, Synopsys

Introduction

Phase or delay control in optical waveguides is key in many applications such as LIDAR[1] (Light Detection and Ranging) and microwave photonics (e.g. OBFN[2], Optical Beam Forming Networks). In many cases, the phase or delay control relies on path length control. In those chip designs, the length of the waveguides needs to be designed accurately to make sure certain phase or delay differences between different paths can be achieved.

Challenge

In principle, the path length of each channel can be calculated by adding the geometrical length of all the segments. The difficulty is to route waveguides to meet the path length difference between each other while the input and output locations are restricted.

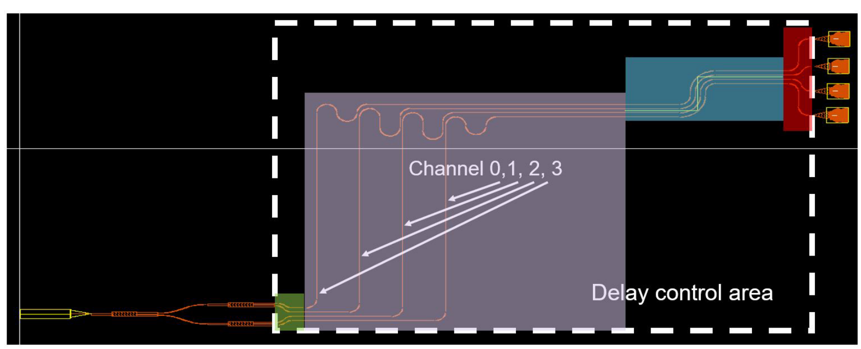


Figure 1: OptoCompiler layout view of a path control structure with the highlight of the delay control area

In figure 1 from left to right, there is a rib waveguide, a rib to strip transition, an MMI1x2, two sine bends, and two MMI1x2 till the left edge of the delay control area. Four fiber couplers locate at the right edge of the delay control area. The outputs of the two MMI1x2 are the inputs of the delay area. The inputs of the fiber couplers are the outputs of the delay area. The delay control area consists of four sections—a pitch converter, a delay compensation section, a flexible bus waveguide section, and another pitch converter. In this case, there are four channels made of pitch converters covered by green and red blocks, a delay component covered by the gray block, and a bus connector covered by the blue block. The path of each channel needs to be designed accurately to make sure the waveguides link the inputs and outputs while the path length difference requirements are met.

Solution

In OptoCompiler, the path length difference among the channels can be controlled by several kinds of delay components. In this paper, BusDelay and BusToPitchDelayFlexConnectorManhattan will be used. The first way is to use separate components including BusDelay to manually compensate for the unintended path length difference introduced by other components. The second way is to use a single component BusToPitchDelayFlexConnectorManhattan to automatically control the path in the entire channel. Please note the flow of schematic design -> pre-layout simulation > layout implementation -> back-annotated to schematic design -> post-layout simulation is used in this application.

Manual Compensation

In this approach, BusToPitchManhattan, BusDelay, BusFlexConnectorManhattan, and BusToPitchManhattan comprise the delay control area as shown in figure 2. BusDelay is the only component that can set the path length difference among different channels. A mode-locked laser model is used here to generate a single pulse for this simulation.

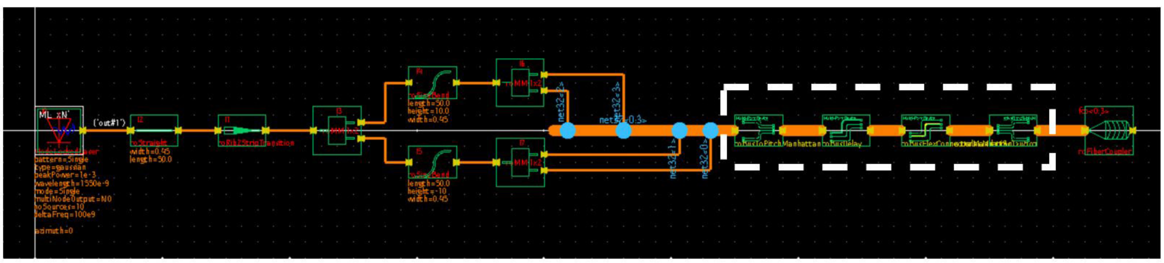


Figure 2: OptoCompiler schematic of a path delay design with manual compensation approach

When the difference among the channels is set to zero, the path lengths are the same for all four channels.

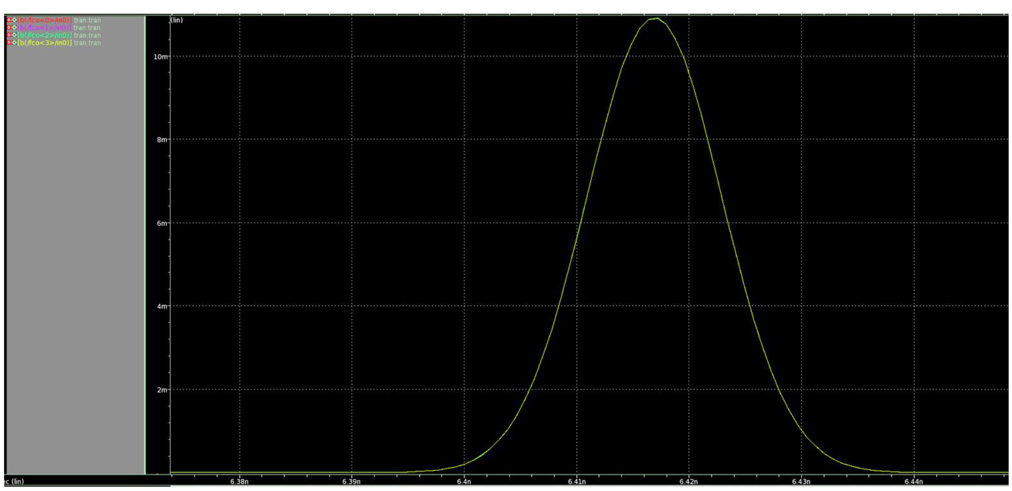
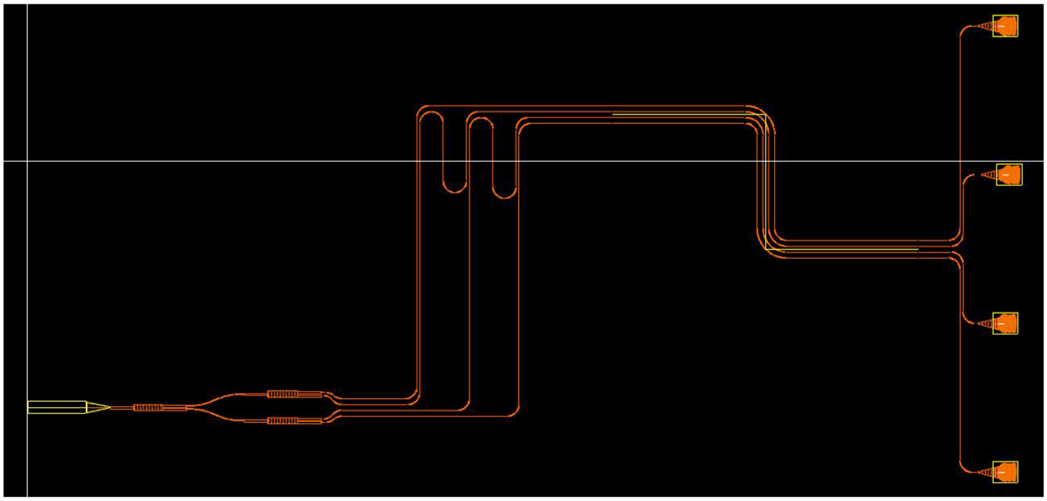
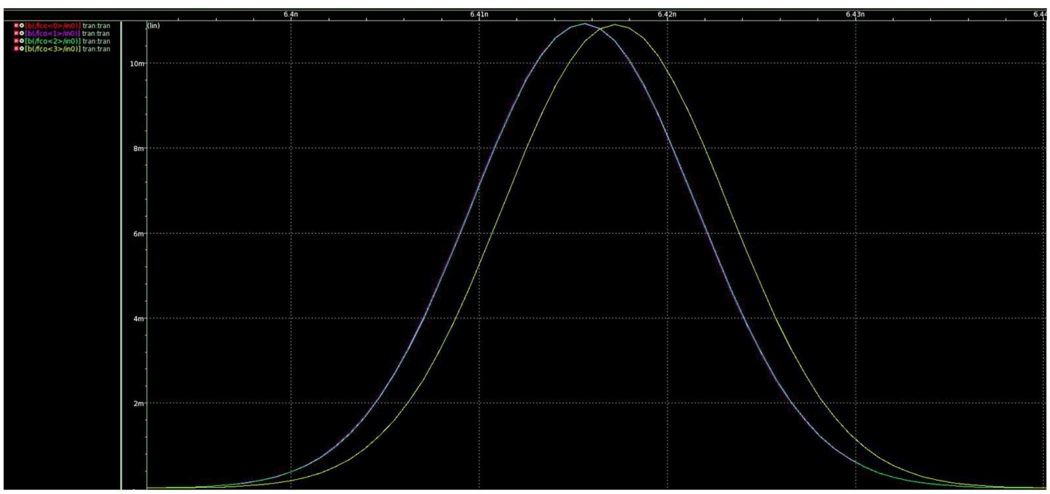


Figure 3: OptSim pre-layout simulation result of a path delay design with zero difference between channels

In the pre-layout simulation results as shown in figure 3, the pulse split into 4 channels, arrives at the fiber couplers at the same time and the curves overlap with each other.



(a)



(b)

Figure 4: OptoCompiler layout (a) and simulation result (b) based on the layout

During the layout implementation shown in figure 4, the four outputs from the two MMIs don't have the desired pitch. BusToPitchManhattan converts the different pitches at its input to the desired pitch at its output. BusDelay makes sure the length difference among channels is zero. A bus connector connects the outputs of the BusDelay and the outputs of the other BusToPitchManhattan. Channel #1 and channel #2 in the middle are shorter than the other two so in the post-layout simulation results the pulses that traveled in channels #0 and #3 take more time to arrive at the fiber couplers than the pulses traveling in channel #1 and #2.

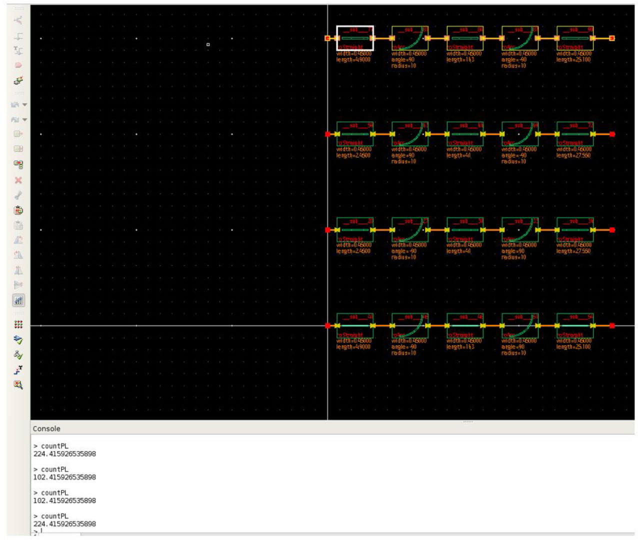


Figure 5: OptoCompiler schematic of BusToPitchManhattan segments

The main difference is caused by the different path lengths from the BusToPitchManhattan before the fiber couplers among the four channels. In figure 5, channel #0 and channel #3 are about 122 μm longer than channel #1 and channel #2. In the schematic, the designer can easily dive into the hierarchical component and check the segments. Please note the design database is fully accessable via Si2 OpenAccess and thus the designer can create their own analysis / processing scripts in any language supported by Si2 - typically Tcl or Python. Alternatively the designer can just browse it and inspect properties via the normal element interface.

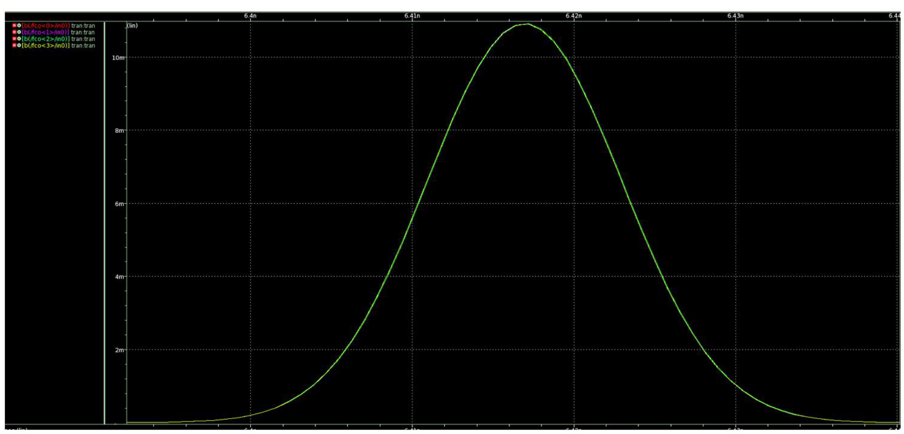


Figure 6: OptSim post-layout simulation result with manual compensation

There is also a length difference of about 1 μm between the outer and inner waveguides in the BusToPitchManhattan after the MMIs, so the total length difference that needs to compensated is about 121 μm for channel #1 and channel #4 relative to channel #2 and channel #3. After the compensation by adjusting the BusDelay component, all the curves overlap again in figure 6.

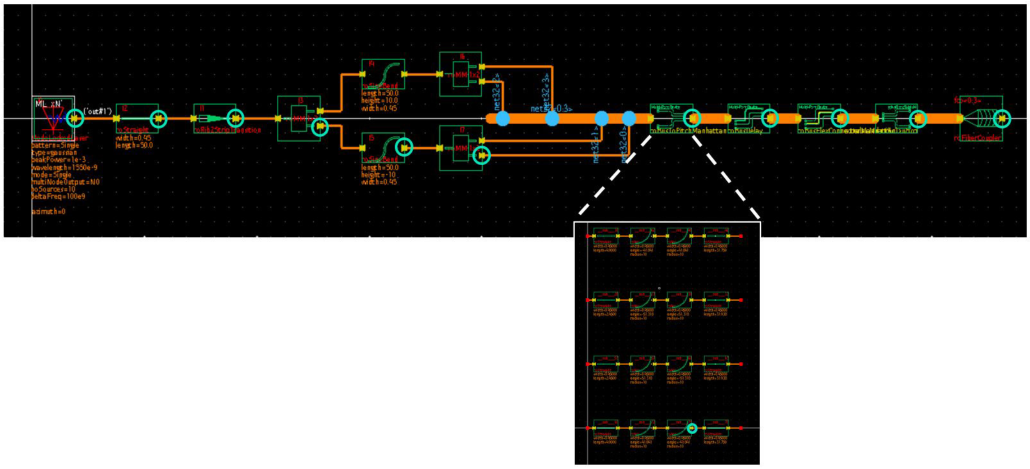


Figure 7: OptoCompiler schematic of a path delay design with the monitored ports

In most cases, it is also interesting to monitor the pulse at various points along the channels as shown in figure 7 to analyze loss and dispersion etc. at each stage.

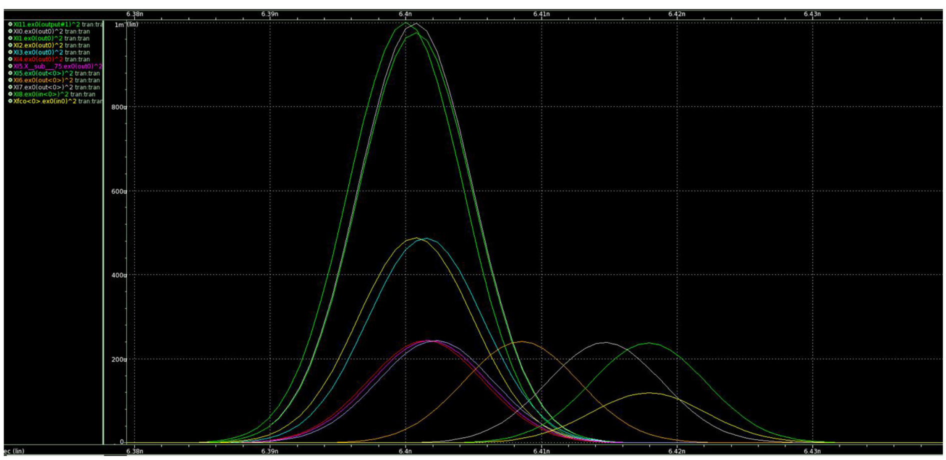
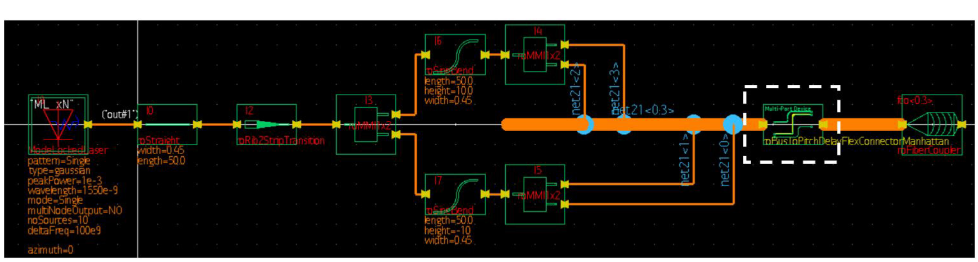


Figure 8: OptSim simulation result of pulse intensity at monitored ports

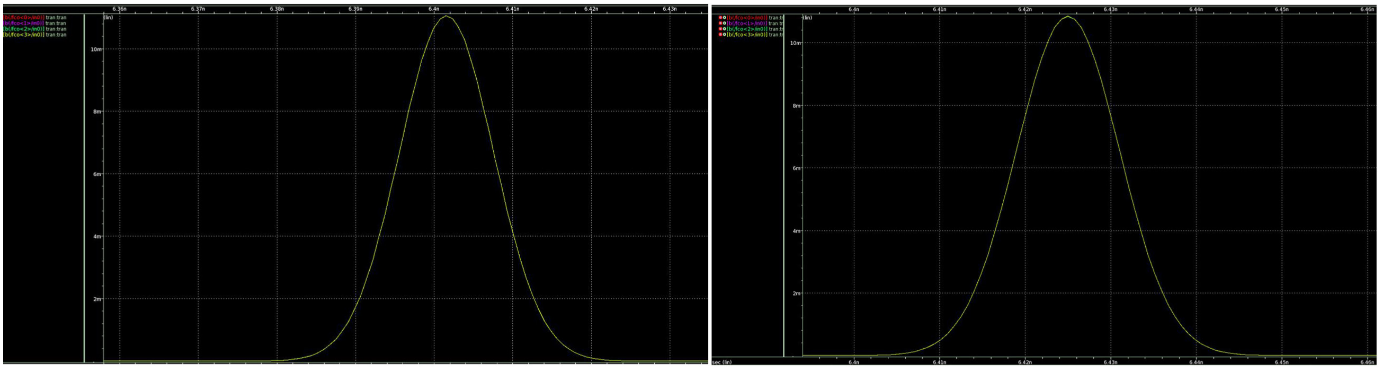
The simulation in figure 8 result shows the intensity of the pulse decreases and the peaks occur at later times when a pulse travels along with channel #0 in figure 7.

Automated Compensation

In this approach, only the component BusToPitchDelayFlexConnectorManhattan is needed. This component is equivalent to the sum of BusToPitchManhattan, BusDelay, BusFlexConnectorMahattan, and BusToPitchManhattan in the first approach. The BusDelay component can automatically adjust for length differences introduced by the other parts, so the designer can fully control the channel path length differences without doing manual compensation.



(a)



(b) (c)

Figure 9: (a) OptoCompiler schematic of a path delay design with automated compensation approach; OptSim (b) pre-layout simulation result (c) post-layout simulation result

In figure 9, the pre-layout and post-layout simulation results show the pulses from different channels arrive at the fiber couplers at the same time when the delay difference is set to zero. In the post-layout case, there is some additional path length compared to the pre-layout case, so all the pulses arrive 20 ps later compared to the pre-layout case—but due to the automatic compensation, they all arrive at the same time.

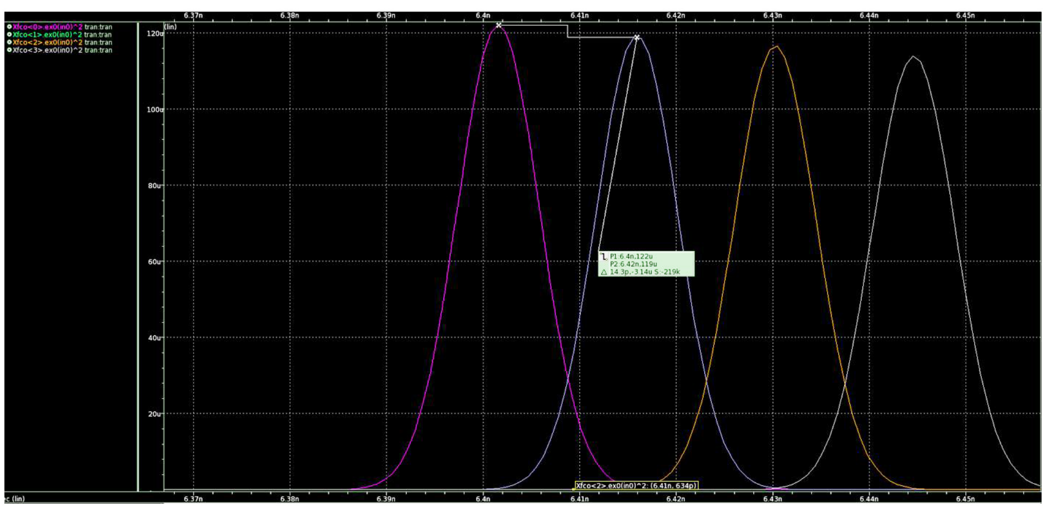


Figure 10: OptSim simulation result of pulse intensity with 1000 um path difference between each other

In figure 10, the difference is set to increase by 1000 μm from one waveguide to the next, so the pulses arrive at different times. The intensity decreases when a pulse travels through a longer path because of inherent waveguide losses.

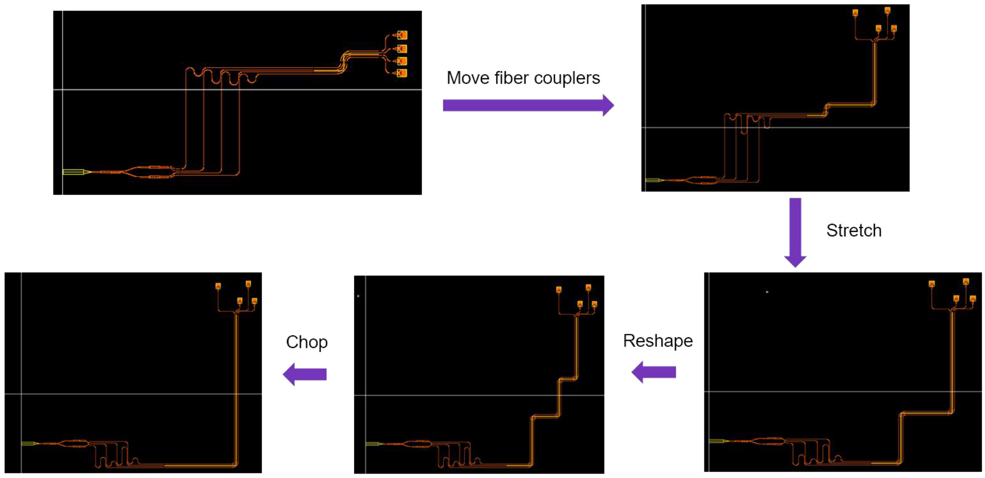


Figure 11: Automated compensation while layout editing

This component is very flexible from a layout perspective. During editing the fiber coupler locations, stretching, reshaping and chopping the connector part, the whole design remains connected and the difference between the channels remains kept as shown in figure 11. Please note this stretching, reshaping and chopping options are also available for the BusFlexConnectorManhattan used in the manual compensation.

Conclusion

In this paper, full control of the path length difference in channels with OptoCompiler is shown. The automated method can compensate for the extra path length introduced during layout implementation. This technology can be used in path length-sensitive applications such as LIDAR and OBFN.

References

[1] C. V. Poulton et al, “Coherent solid-state LIDAR with silicon photonic optical phased arrays,” Opt. Letters, vol. 42, no. 20, p. 4091, 2017

[2] L. Zhuang et al, “Single-Chip Ring Resonator-Based 1 × 8 Optical Beam Forming Network in CMOS-Compatible Waveguide Technology,” vol. 19, no. 15, p. 1130, 2007.

03/07/22.CS850313437-Flexible Path Delay Control With Layout Flexibility in OptoCompiler WP.

Pub: Feb. 2022