Compact Single-Ended Transceivers Demonstrating Flexible Generation of 1/N-Rate Receiver Front-Ends for Short-Reach Links

Myungguk Lee, *Graduate Student Member, IEEE*, Jaeik Cho, Junung Choi, *Graduate Student Member, IEEE*, Won Joon Choi, *Graduate Student Member, IEEE*, Jiyun Lee, *Graduate Student Member, IEEE*, Iksu Jang, *Graduate Student Member, IEEE*, Changjae Moon, *Graduate Student Member, IEEE*, Gain Kim, *Member, IEEE*, and Byungsub Kim, *Senior Member, IEEE*

***Abstract*— This paper presents compact single-ended wireline transceivers with software-generated receiver front-ends. The developed software framework significantly shortens the physical design time of 1/N-rate wireline receiver front-ends. The physical layouts of various receiver front-ends were software-generated in four different CMOS technology nodes (28 nm, 40 nm, 65 nm, and 90 nm) with four different front-end architectures targeting various data rates. In the post-layout simulation, the receiver front-ends generated within a second by the software achieved nearly the same performances as the manually-designed receiver front-ends that require more than about 30 hours of design time. For demonstration, we generated 8 Gb/s full-rate, 10 Gb/s half-rate, 12 Gb/s, and 20 Gb/s quarter-rate receiver front-ends, and fabricated them with a manually-designed feed-forward equalization transmitter in 28 nm CMOS process. The transceivers were measured with the data rate up to 20 Gb/s while consuming 1.39 pJ/b at the channel loss of −9.2 dB. The transceiver with software-generated receiver achieved the highest data rate per area as well as the smallest area among the relevant prior arts while reducing the physical design time of the receiver front-end by more than 140,000 times.**

***Index Terms*— Wireline communications, short-reach links, single-ended signaling, layout design automation, analog layout generator, receiver front-end generator.**

Manuscript received 3 May 2023; revised 4 August 2023 and 17 September 2023; accepted 9 November 2023. This work was supported in part by the Commercializations Promotion Agency for Research and Development Outcomes (COMPA) Grant funded by the Korea Government [Ministry of Science and ICT (MSIT)] under Grant 2023I100; in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korea Government (MSIT) under Grant 2022-0-01171; and in part by the Brain Korea 21 Program for Leading Universities and Students (BK21) FOUR of NRF for the Department of Electrical Engineering, Pohang University of Science and Technology. This article was recommended by Associate Editor L. Shen. *(Corresponding author: Byungsub Kim.)*

Myungguk Lee, Junung Choi, Won Joon Choi, Jiyun Lee, Iksu Jang, and Changjae Moon are with the Department of Electrical Engineering, Pohang University of Science and Technology, Pohang-si 37673, South Korea.

Jaeik Cho is with Samsung Electronics, Hwaseong-si 18448, South Korea.

Gain Kim is with the Department of Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology, Daegu 42988, South Korea (e-mail: gain.kim@dgist.ac.kr).

Byungsub Kim is with the Department of Electrical Engineering, the Department of Convergence IT Engineering, and the Department of Semiconductor Engineering, and the Graduate School of Artificial Intelligence, Pohang University of Science and Technology, Pohang-si 37673, South Korea, and also with the Institute for Convergence Research and Education in Advanced Technology, Yonsei University, Seoul 03722, South Korea (e-mail: byungsub@postech.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2023.3332391.

Digital Object Identifier 10.1109/TCSI.2023.3332391

I. Introduction

With the advancement of multi-chip module (MCM) and system-in-package (SiP) technologies, single-ended short-reach parallel links are becoming increasingly important for high-performance computing systems. As high-performance MCM-based processors incorporate numerous wireline transceivers to support ever-increasing chip-to-chip communication bandwidth with limited area and power budget, each transceiver design must be as compact and energy-efficient as possible.

High-bandwidth chip-to-chip communication is a typical example where careful optimization from circuit level to system level, including fine-tuning MOSFET sizes and architecture choices such as time-interleaving order, is particularly important. However, with the limited development time budget, optimizing the short-reach transceiver design for the target application can be very challenging due to the extended physical design time required for high-speed building blocks. Moreover, frequent design modification and technology migration make this issue more serious.

Over the past few decades, researchers have been continuously investigating automatic layout generation to reduce the physical design time of analog circuits [1], [2], [3], [4], [5], [6], [7], [8]. However, despite the efforts of many researchers, analog layout generation tools have not yet been widely adopted in the semiconductor industry. To establish layout automation in the semiconductor industry, it is important to demonstrate the practical applicability of layout automation. Although many studies [1], [2], [3], [4], [5], [6], [7], [8] have developed frameworks and methodologies for layout automation, there are not many research results that focus on demonstrating the usefulness of the layout automation technique by showing practical examples. Especially, there are only a few research reports that deal with high-speed wireline transceiver, where layout quality significantly affects the overall performance of the implemented circuits. Prior arts [6], [7], [8] focused on the development of the layout generation framework and methodologies. They implemented the high-speed circuits such as a DAC-based PAM-4 transmitter, a 1:16 data de-serializer, and a transceiver front-end using their frameworks, respectively. However, the layout generator developed in [6], [7], and [8] did neither aim for single-end short-reach links nor demonstrated their performances through chip fabrication and measurement.

High-speed serial data transmitters or a receiver have been developed using the layout generation techniques and validated with silicon in [9], [10], and [11]. Choi et al. [9] used the Berkeley Analog Generator (BAG) [8] to fabricate the 200 Gb/s PAM-4 transmitter with 5-tap FFE in 28 nm CMOS process. Various building blocks for the ultra-high-speed transmitter were software-generated in [9], achieving a data rate of 200 Gb/s, the highest among the transmitters designed with the analog layout generators to date. Han et al. [10] developed the software for the layout generation of the 4-tap FFE SST transmitter. Seven different 4-tap FFE SST transmitters in three different technology nodes (40 nm, 65 nm, and 90 nm CMOS) were software-generated in [10], achieving a data rate of 36 Gb/s with the post-layout simulation. E. Chang et al. [11] demonstrated the analog layout generation of the wireline receiver front-end. To compensate for the 15-dB channel loss, various high-speed equalizer building blocks such as continuous-time linear equalizer (CTLE), 1-tap feed-forward equalizer (FFE), and 4-tap decision feedback equalizer (DFE) were implemented in [11] using the BAG at the receiver side. While the prior arts [9], [10], [11] achieved reasonable area and energy efficiency considering significantly improved design time with software layout generation, manual designs [12], [13], [14], [15], [16], [17] still outperform in terms of area and energy efficiency as they are optimized for short-reach links with optimized architecture parameters such as interleaving order.

In this work, we developed a layout generator specialized for high-speed short-reach links, demonstrating that the cost-efficient receiver front-ends suitable for short-reach links can be generated very quickly and reliably. To implement a compact and energy-efficient receiver front-end, it is important to properly choose the time-interleaving order as well as the device sizes for the target speed. To achieve this, we developed generator that allows to change the time-interleaving order of the generated receiver front-end. The developed 1/N-rate receiver front-end generator with flexible choices of N reduced the design time of high-speed short-reach links for optimization of MOSFET sizing and architectural parameters by automatically generating the netlists and layouts of the circuits. From input design parameters, the appropriately sized receiver front-ends in various time-interleaved architectures could be automatically generated in multiple technology nodes in a second.

For demonstration, four different receiver front-ends targeting the data rates ranging from 8 Gb/s to 20 Gb/s were generated, fabricated in 28 nm CMOS process, and tested with a bit error rate tester (BERT) equipment or a manually-designed FFE transmitter. In the experiment, the fastest receiver front-end among the prototype chips achieved a data rate of 20 Gb/s and energy efficiency of 0.18 pJ/b, while occupying the smallest area compared to the relevant prior arts [11], [13], [14], [15], [16], [17]. In addition, we also demonstrated reliability and fast generation time of the proposed generator through various analyses.

The rest of this paper is organized as follows. To help the readers better understand the analog layout generator, Section II briefly explains how our software automates the design process. Section III describes the 1/N-rate receiver front-end generator developed using our software and discusses the performances of the various receiver front-ends generated in multiple processes with post-layout simulation results. The measurement results and comparison with the relevant prior arts are shown in Section IV. Finally, Section V provides the conclusion of this paper.

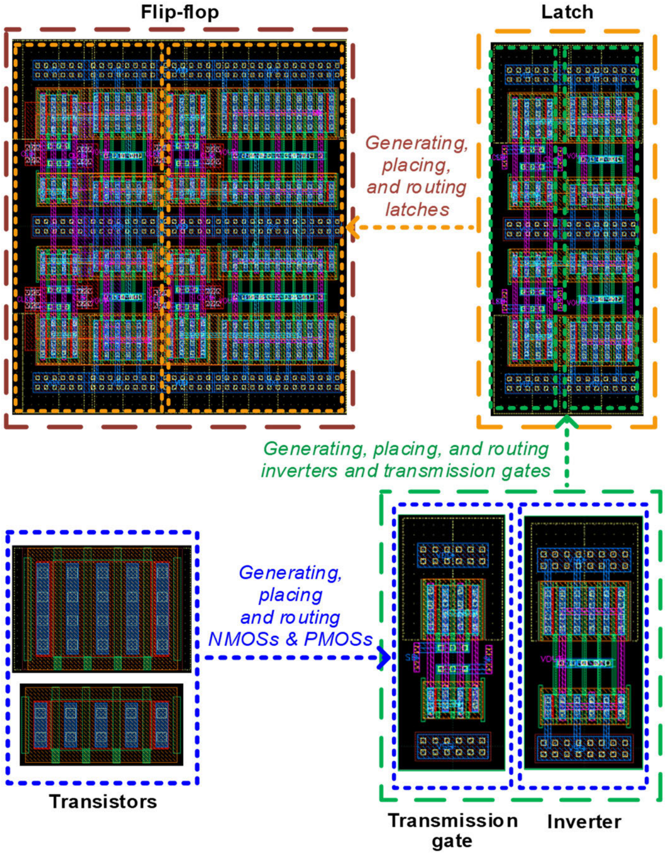
Flip-flop
Latch
Generating, placing, and routing latches
Generating, placing, and routing inverters and transmission gates
Generating, placing and routing NMOSs & PMOSs
Transistors
Transmission gate
Inverter

Fig. 1. An example hierarchical generation of a flip-flop layout.

II. Layout Generation

Our software was developed based on the layout generation framework [10] that hierarchically describes and generates layout elements. The layout of the top-level cell is generated by hierarchically importing, placing, and routing the sub-cell layouts multiple times. The sub-cell layouts are generated by the sub-cells' layout generators that comply with the design rules provided by the semiconductor foundry. Therefore, the designers can import the sub-cell generators into the software library without considering the complex design rules within the sub-cells. A hierarchical generation of a flip-flop is shown in Fig. 1 as an example. The flip-flop is generated by repeatedly utilizing the generators of its sub-cells. First, the transmission gate and the inverter can be generated by placing the lowest-level basic layouts generated from the standard generators (e.g. transistors) and routing between them. A higher-level cell, a latch, can also be generated by

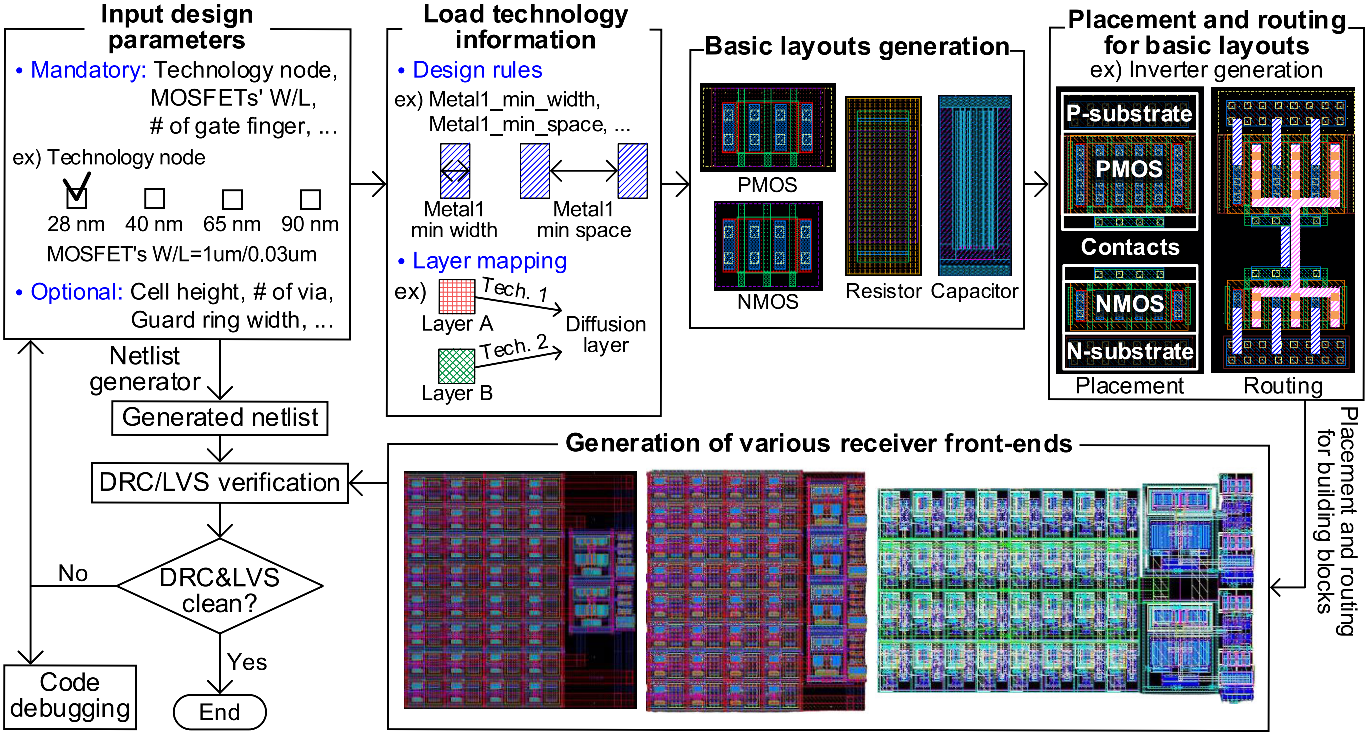

Fig. 2. Layout generation flow of the proposed software.

importing the generated sub-cell layouts (transmission gates and inverters) and performing the placement and routing for these sub-cells. In the same way, the top-level cell, a flip-flop, is generated by utilizing the generated sub-cell layouts of latches. With this hierarchical layout generation approach, the designer just needs to generate the sub-cell layouts and proceed with placement and routing. Therefore, hierarchical layout generation allows the designers to efficiently generate various layout instances using the sub-cell generators.

The proposed layout generator deterministically generates layouts with pre-planned optimal placement and routing strategies by experienced generator developers with expertise in high-speed circuit layouts. The performance of high-speed circuits is greatly influenced by the quality of their layout design. Therefore, to generate the physical design of receiver front-ends with optimal post-layout performance, the generator was developed to perform the following placement and routing strategies: 1) The placement of insensitive devices is parameterized using the minimum spacing rule to minimize the area. 2) We surround the sensitive amplifiers with guard rings to reduce the interference from other signals. 3) We placed the layout instances as close as possible to each other to minimize the impacts of parasitics, minimizing the wirelength of critical path such as the output of the sense amplifier and the high-speed clock. 4) We route the data and clock paths as symmetrically as possible to minimize the timing skew in the critical paths. 5) The power and ground rings are formed around the receiver front-end using the thicker metal to reduce the impedance and voltage drop in the power network. If the area of the receiver front-ends becomes larger due to the increase in time-interleaving order, vertical and horizontal power straps are generated to evenly distribute power to the circuits inside. 6) The critical input path of the receiver front-end, which is connected to the chip pad, has a mesh structure to minimize the impedance of the path. The layouts of the receiver front-ends using this placement and routing approach cannot be visually distinguished from the manually designed layouts, and their performances are also comparable. In addition, since the layout instances of the receiver front-ends are deterministically parameterized with the pre-planned placement and routing strategies, all receiver front-end layouts can be generated with a runtime of less than a second.

Fig. 2 illustrates the overall layout generation procedure of our software. Firstly, the layout generator receives mandatory and optional inputs from the designer. The mandatory inputs are the essential design parameters, including the technology node, MOSFETs' gate width and length, the number of gate fingers, etc. Optional inputs include the cell height, the number of vias, the guard ring width, etc., and are determined by the pre-defined default values if the user does not specify these values. After receiving the inputs, the layout generator loads the process design rules and layer information corresponding to the selected technology node. Then, the basic layouts at the lowest hierarchical level are generated based on the inputs that the designer specified. The simple building blocks such as an inverter can be generated by placing and routing the basic layouts. For larger circuits such as a receiver front-end, the layout generator properly repeats the placement and routing for the lower-level building blocks. At the same time, the netlist of the receiver front-end is generated to prepare the layout versus schematic (LVS) verification. After the top-cell layout instance is generated, the layout generator can automatically proceed to design rule and LVS verification using the commercial tool. The layout generator parses the output log and informs the user of any errors in the generated layouts. If no error occurs, the layout generation procedure is over. If errors exist, the user needs to revert to the first step and modify the input parameters, or debug the generator code to resolve the error. However, in most cases, the proposed layout generator produces the layouts without design rule violation and LVS violation. To support this claim, we verified the reliability of the generator, which will be discussed in more detail in Section III.

The automatic layout generator can drastically save time spent iterating on schematic and layout designs for design optimization. This design iteration for parameter tuning and layout modification is particularly very time-consuming in single-ended short-reach link applications because such applications usually require tight optimization to meet very high target levels of power and area efficiencies. Therefore, the automatic layout generation greatly reduces the design time.

In addition, the automatic layout generator is also beneficial when designing high-speed analog circuits since any users can easily create high-quality layouts with the aid of tools. For high-speed analog circuits, even for the circuits with the same design parameters, the performance of the fabricated chips (or post-layout performance) may differ significantly depending on the quality of the layouts. Because the level of expertise of the layout engineers greatly affect the layout quality, such layout-dependent performance variation necessitates experienced and skilled layout professionals for high-speed wireline transceiver design, increasing the development cost especially in advanced process nodes. On the other hand, the layout generator allows anyone to generate high-quality layouts for high-speed wireline transceivers using the tool, provided that the layout generator is developed in high quality. Therefore, using the layout generator can significantly reduce the manpower cost of high-speed wireline transceiver design.

It is necessary to address potential issues that may arise due to differences in device structures and design rules among various technology nodes when utilizing the layout generator to port designs to different technologies. In advanced technologies, each technology node has different device structures, layer organizations, and design rules, and even within one process, different design rules may be required depending on the layout patterns. These differences become more diverse as the technology gets scaled down such as FinFET. Therefore, being aware of these issues, the layout generator must be developed to operate flexibly, adapting to the situation as necessary. Our generator dynamically performs the layout instance generation, placement, and routing based on the situation, resolving the mentioned issues. In our software, when generating the devices, the software uses a single main code and technology-specific codes. The main code generates the common structures and layers that can be shared across different technology nodes. The technology-specific code refines the device layouts by slightly modifying or adding the structures or layers to address the special requirements of each technology node [6]. When updating the generator for migration to a new process, the main code is reused and only the technology-specific code is updated. Moreover, to address the layout scenarios arising from different design rules across the various technology nodes, the generator dynamically applies the appropriate design rules based on the given situation instead of using fixed design rules, performing placement and routing as well as the layout instance/pattern generation. This layout generation approach ensures that the generator can adapt flexibly to new layout scenarios with only minor code modifications when porting to a different technology node.

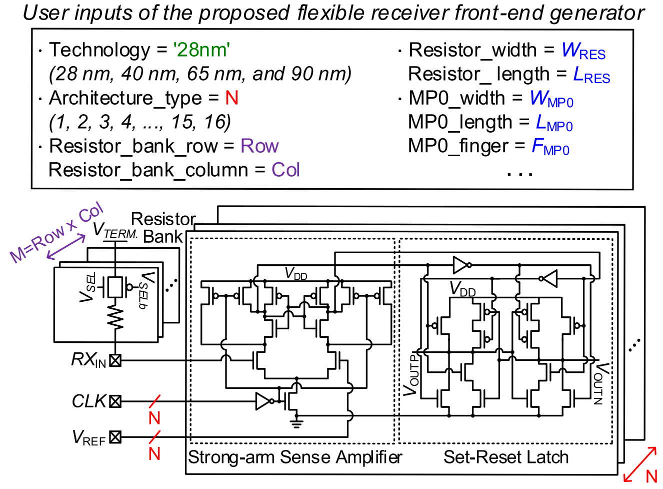

Fig. 3. A schematic diagram of a compact single-ended receiver front-end and the user inputs of its generator.

III. 1/N-Rate Receiver Front-Ends Generator

We developed the 1/N-rate receiver front-ends generator which helps the designer easily customize not only the device sizes but also the order of the time-interleaved architecture. Fig. 3 shows the schematic diagram of a receiver front-end generated by the developed software. The receiver front-end consists of only essential circuits without linear or nonlinear equalizers for area and power efficiency. The 1/N-rate receiver front-end is composed of the N-way time-interleaved strong-arm sense amplifiers, the symmetric set-reset latches, the local clock buffers, and the digitally configurable resistor bank for termination, where N is the input parameter between 1 and 16 that determines the time-interleaved architecture. For the input N, the receiver front-end can be flexibly generated in 1/N-rate (N-way time-interleaved) architecture. The resistor bank consists of M digitally configurable resistor units, where the input parameter M is the product of resistor unit counts per row (Row) and per column (Col) so that the array of the resistor unit layouts can be placed in a rectangular shape: $M = \text{Row} \times \text{Col}$. The sizes of the individual transistors and resistor elements are also parameterized as inputs to the software. Therefore, by using the appropriate input parameters, the designer can easily customize not only the device sizes but also time-interleaved architecture for the target data rate, the supply voltage, and the technology node in order to maximize the area and power efficiency.

1/N-rate flexibility of the receiver front-end is one of the major features of this work. To implement this feature, we developed the software considering the following points. The placement of the resistor unit arrays was adjusted by the resistor unit count parameters (Row and Col) considering the area efficiencies and the aspect ratio. The critical paths were carefully generated with a higher priority than other signal paths to prevent performance degradation due to timing skew and parasitics across various receiver front-end architectures. Especially in receiver front-ends with high time-interleaving orders, the input path of the receiver front-end was generated with a mesh structure to minimize the timing skew. The path connected to the termination was also generated using a mesh structure, which not only reduces parasitics caused by the long wirelength and ensures accurate impedance matching, but also guarantees the same common-mode voltage across all sense amplifiers regardless of their location. The mesh configuration of these critical paths is adjusted to fit the changing aspect ratio of the receiver front-ends due to the time-interleaving orders, rather than being fixed. After the generation of the critical signal paths, the power network is formed. The performance of a sensitive sense amplifier that needs to detect a voltage difference of several millivolts can be affected by supply noise. Therefore, as the area of the receiver front-end layout increases due to the rise in time-interleaving order, the power network must be carefully generated to ensure uniform power delivery to each circuit. To reduce the supply noise fluctuations across the receiver front-end, the software was developed to generate power straps both vertically and horizontally at intervals of about 20 μm based on post-layout simulation, taking into account the time-interleaving order and the aspect ratio of the receiver front-end.

To verify the flexible receiver front-end generator, layouts of various different receiver front-ends were generated by the developed software and verified. Firstly, to test the reliability of the generator, 1,000 pairs of netlists and layout instances of receiver front-ends were generated using random input parameters in each of 28 nm, 40 nm, 65 nm, and 90 nm CMOS technology nodes. All generated 4,000 netlists and layout instances successfully passed the DRC and the LVS verification. Secondly, layouts of various different receiver front-ends (RX1-RX7) in four architectures (full-rate, half-rate, quarter-rate, and octa-rate) were generated using four different technology nodes (28 nm, 40 nm, 65 nm, and 90 nm CMOS) aiming for various target speeds (4 Gb/s, 8 Gb/s, 10 Gb/s, 12 Gb/s, 20 Gb/s, and 32 Gb/s) (Fig. 4). Sizing parameters and time-interleaved architectures were appropriately determined based on simulation. The RX6 and the RX7 differ in device sizes because their target speeds (12 Gb/s and 20 Gb/s, respectively) are different. Also, it is noticeable that the placement of the resistor unit arrays was adjusted by the resistor unit count parameters (Row and Col) considering the area efficiencies and the aspect ratios.

The generated layouts of receiver front-ends achieved the performance similar to manually-designed ones in post-layout simulation. Table I summarizes the target specifications, architectures, and post-layout simulation results of the generated and manually-designed receiver front-ends.

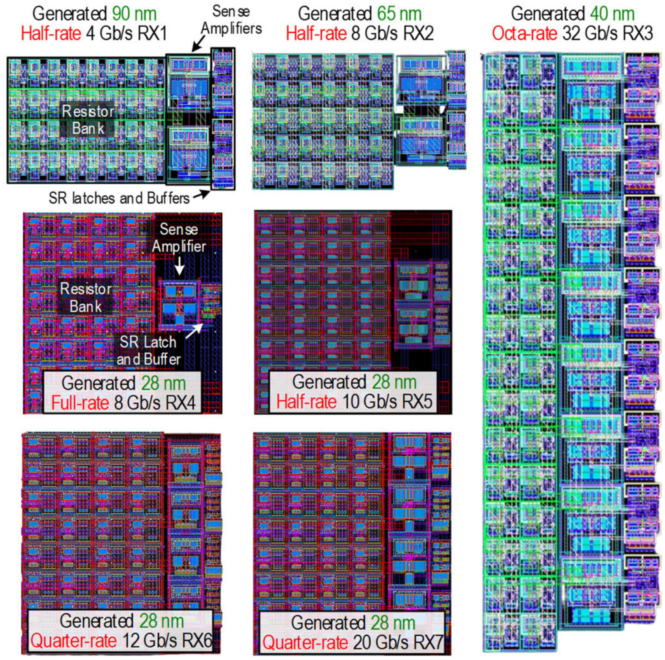

Fig. 4. Layouts of receivers with software-generated front-ends.

TABLE I

POST-LAYOUT SIMULATION RESULTS OF SOFTWARE-GENERATED AND MANUALLY-DESIGNED RECEIVER FRONT-ENDS

	Software-Generated Layout							Manual Layout	
	RX1	RX2	RX3	RX4	RX5	RX6	RX7	RX8	RX9
Technology (nm)	90	65	40	28					
Data Rate (Gb/s)	4	8	32	8	10	12	20	12	20
Architecture	Half-rate		Octa-rate	Full-rate	Half-rate	Quarter-rate			
Energy Efficiency (pJ/b)	0.234	0.18	0.154	0.075	0.077	0.073	0.107	0.07	0.107
Area (μm^2)	1,160	821	1,040	408	443	491	542	548	546
Sensitivity* (mV)	1.4	0.85	4.6	1.7	1.1	1.25	1.9	1.2	1.85
3σ Offset* (mV)	10	9.14	15	14.79	17.98	20.73	16.74	21.36	15
Input Referred Noise* (mV_{rms})	0.384	0.428	0.312	0.418	0.475	0.548	0.385	0.576	0.388
Physical Design Time	0.643 sec	0.667 sec	0.697 sec	0.713 sec	0.719 sec	0.723 sec	0.732 sec	29.1 hr	48.8 hr

* : These performances are the post-layout simulation results for the strong-arm sense amplifier.

For performance comparison between software-generated and manually-designed receiver front-end, the RX8 and the RX9 were manually designed for the same design targets and parameters of RX6 and RX7, respectively. It is noteworthy that the performances of the generated RX6 and RX7 are very similar to those of the manually-designed RX8 and RX9, respectively. In addition, the octa-rate receiver front-end generated in 0.697 seconds using the 40 nm CMOS process achieved a maximum data rate of 32 Gb/s while occupying small area of 1,040 μm^2 (18.4 μm × 56.5 μm) and dissipating 154 fJ/b in post-layout simulation.

The generator greatly reduced the physical design times of the receiver front-ends from about 30-50 hours to less than 1 second at the cost of a one-time coding of 44.3 hours to

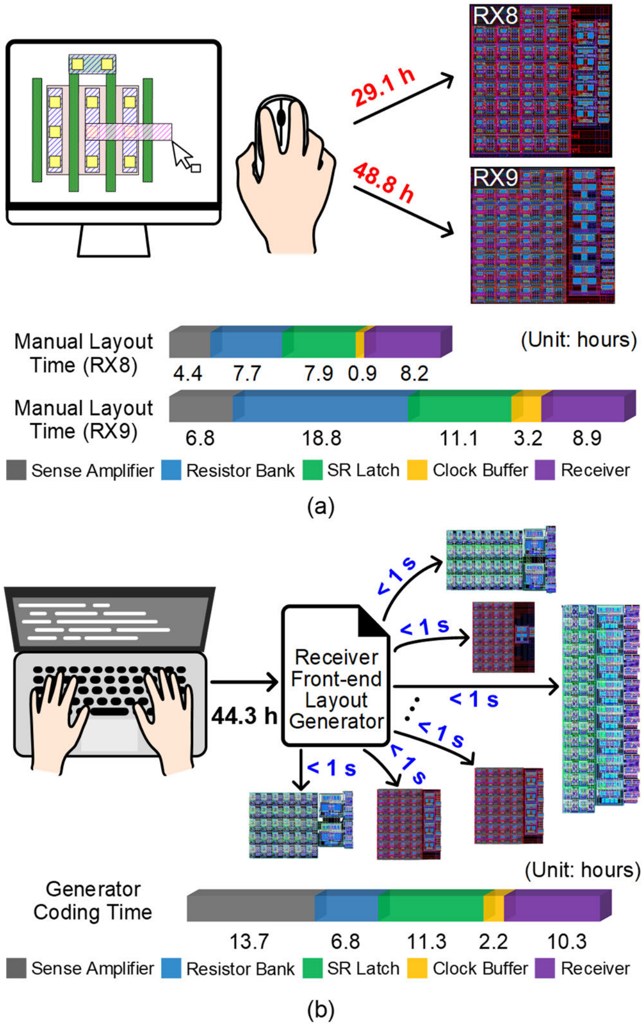

Fig. 5. Breakdowns of (a) the manual physical design times and (b) the generator coding time.

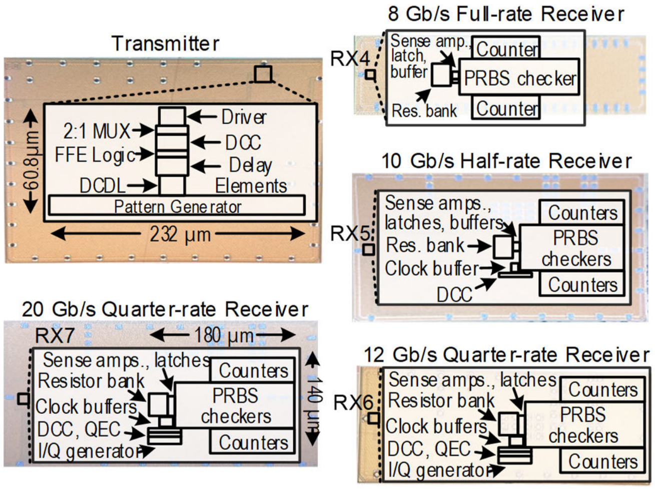

Fig. 6. Chip micrographs of the fabricated receivers designed with the developed layout generator, together with the manually-designed transmitter.

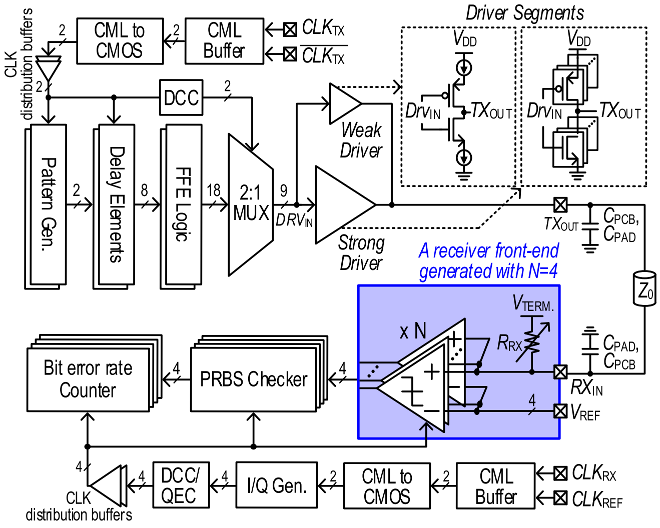

Fig. 7. The overall transceiver architecture.

describe the receiver front-end. Table I and Fig. 5 summarizes the physical design times by the human designers and the generator as well as the coding time of the generator. Regardless of technology node and architecture, the proposed software generated layouts of various receiver front-ends within an instant, while human designers had to spend tens of hours for drawing each layout. The generator coding time is 52% longer and 9% shorter than the manual layout design times of RX8 and RX9, respectively. Writing the source codes to describe the receiver front-end took 44.3 hours while the manual layout of a quarter-rate 12 Gb/s receiver front-end (RX8) took 29.1 hours; coding of the generator took 52% more time (15.2 hours) than the manual layout of the RX8 because the RX8 is a relatively simple for layout drawing. On the other hand, coding of the generator required 9% less time (4.5 hours) than the manual layout of the quarter-rate 20 Gb/s receiver front-end (RX9) because the RX9 requires more careful physical design due to the higher target data rate. Once the architecture is properly described, the software can generate the layout of the receiver front-end within a second. In the two examples, the generator reduces the layout development time by about 140k – 240k times at the cost of a one-time overhead about between −9% and 52% of the manual layout design time. Although there are not many samples of development times, it is noticeable that the one-time overhead for coding is comparable to the manual layout design times of the similar circuits, and the time reduction in layout design by the software is drastic. This result demonstrates the usefulness of the generator for design iteration, modification, and porting to different technology nodes. With the developed software, multiple physical designs of receiver front-ends with various sizes can be swiftly generated for different technology nodes, targeting various speed requirements.

In order to migrate to a new technology node, the designer must spend additional time on the following two tasks: 1) the configurations for the new technology node must be provided to the framework, and 2) the technology-specific code of the layout generator must be updated for the new technology node. If framework has not been configured for the new technology node, then the user must specify the configuration for the newly added technology node once. This process can be time-consuming if the configuration for the new technology

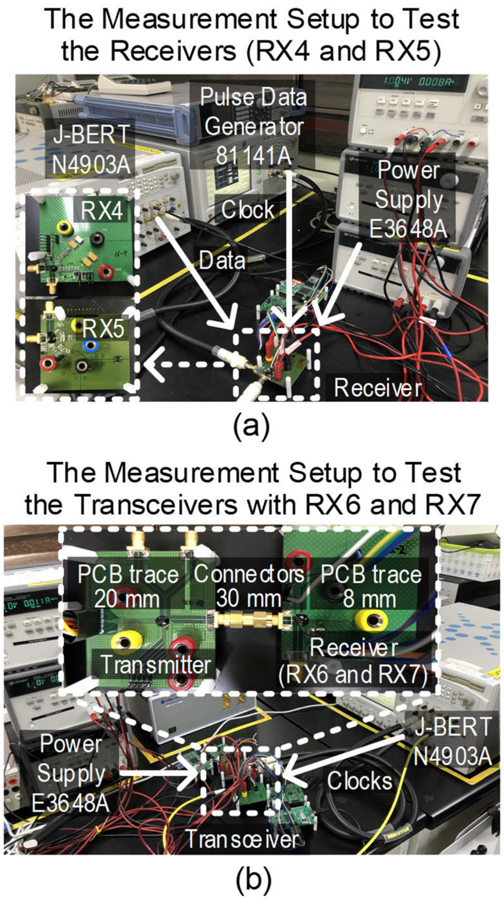

Fig. 8. The measurement setup to test (a) the 8 Gb/s full-rate and 10 Gb/s half-rate receivers and (b) the transceivers with the 12 Gb/s and 20 Gb/s quarter-rate receivers.

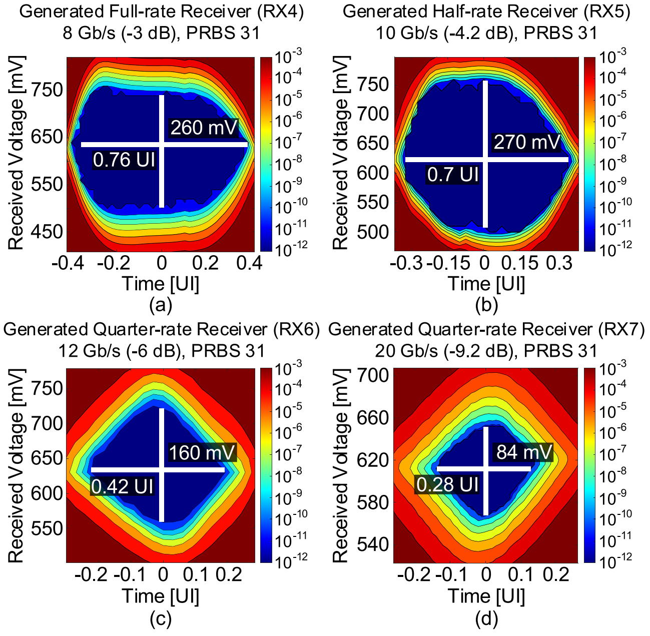

Fig. 9. The measured *in-situ* eye diagrams.

TABLE II
POWER CONSUMPTION AND AREA OF TRANSMITTER AND RECEIVERS

Transmitter			
Architecture		Half-rate	
Data Rate (Gb/s)		12	20
Power Consumption (mW)	Driver	6.5	12.3
	2:1 MUX	1.1	2.18
	FFE logic	0.29	0.52
	Delay units	0.34	0.73
	DCC & Buffers	4.6	8.46
	Total	12.83	24.19
Area (μm^2)	Driver	363	
	2:1 MUX	130	
	FFE logic	104	
	Delay units	206	
	DCC & Buffers	346	
	Total	1,149	

Software-generated recievers					
Architecture		Full-rate	Half-rate	Quarter-rate	
Data Rate (Gb/s)		8	10	12	20
Power Consumption (mW)	Sense amps.	0.28	0.44	0.62	1.35
	SR latches	0.03	0.04	0.06	0.12
	Clock buffers	0.18	0.3	0.48	1.09
	Termination	0.12	0.23	0.48	1.1
	Total	0.61	1.01	1.64	3.66
Area (μm^2)	Sense amps.	29	54	81	117
	SR latches	7	14	28	39
	Clock buffers	4	7	14	18
	Termination	368	368	368	368
	Total	408	443	491	542

node differs significantly from the existing one. After one-time configuration, the framework is ready for design migration to the new technology node. If this configuration is previously done, then this process is not necessary. For every layout generator, the user must update the technology-specific code for the new technology in order to migrate to the new technology node. The technology-specific code is written to meet special requirements for a specific technology node. Because the technology-specific code is much shorter than the main code, updating the technology-specific code will not take a lot of time.

IV. MEASUREMENT RESULTS

Four different receiver front-ends (8 Gb/s full-rate RX4, 10 Gb/s half-rate RX5, 12 Gb/s quarter-rate RX6, and 20 Gb/s quarter-rate RX7) were generated by the developed tool, then fabricated with a manually-designed FFE transmitter in 28 nm CMOS process (Fig. 6). The overall architecture of the transceiver including the test-support blocks is depicted in Fig. 7. The manually-designed single-ended inverter-based FFE transmitter with relaxed impedance matching [12] was utilized for good area and power efficiency of the transceiver. Various test-support blocks such as pseudo-random binary sequence (PRBS) pattern generators, PRBS checkers, and bit error rate counter circuits were included in the fabricated chips for in-situ eye measurement. The in-phase and quadrature clock generator (I/Q generator) [18] provides the I/Q clocks for the quarter-rate transceivers. The duty-cycle error and the skew error of the I/Q clocks were compensated by the duty-cycle correctors (DCC) and the quadrature error correctors (QEC), respectively. The refined quadrature clock by the DCC and QEC is conveyed to the receiver front-end and test-support blocks through clock distribution buffers composed of CMOS inverter stages. To reduce not only hardware cost and power consumption but also jitter accumulation, the number of stages in the clock buffer has been minimized, taking into account the fan-out factor.

The generated full-rate receiver RX4 and the half-rate receiver RX5 were tested with −3 dB and −4.2 dB channel losses (a 10 mm PCB trace, SMA connectors, and a one meter SMA cable), respectively, using a BERT equipment (Agilent N4903A) without the transmitter, as shown in Fig. 8(a). As shown in Fig. 9(a) and (b), the RX4 and the RX5 achieved

TABLE III
PERFORMANCE SUMMARY AND COMPARISON

		VLSI'18 [11]	JSSC'19 [13]	ISSCC'22 [14]	TCAS-I'22 [15]	ISSCC'21 [16]	ISSCC'21 [17]	**This work**			
Technology		16 nm FinFET	16 nm FinFET	28 nm LPP	28 nm LPP	7 nm FinFET	7 nm FinFET	**28 nm LPP**			
Signaling		NRZ (Diff.)	GRS (SE)	DECS (SE)	NRZ (SE)	NRZ (SE)	PAM-4 (SE)	**NRZ (SE)**			
Layout Generation	TX	Automation	Manual	Manual	Manual	Manual	Manual	**Manual**			
	RX	Automation	Manual	Manual	Manual	Manual	Manual	**Automation**			
Equalization		RX:1-tap FFE, 4-tap DFE, CTLE	TX:Edge boosting RX:Linear EQ	None	None	RX:CTLE	TX:5-tap FIR RX:CTLE, VGA	**TX: 4-tap FFE**			
Negative voltage		X	O	X	X	X	X	X			
Data Pattern		PRBS 7	PRBS 31	PRBS 31	PRBS 31	PRBS 31	PRBS 31	**PRBS 31**			
Architecture		Half-rate	Half-rate	Half-rate	Quater-rate	Quater-rate	Quater-rate	**RX4**	**RX5**	**RX6**	**RX7**
								Full-rate	**Half-rate**	**Quater-rate**	
Data Rate (Gb/s)		15	25	20	16	40	112	**8**	**10**	**12**	**20**
Eye Size		-, 0.375 UI (-15 dB)	-, 0.42 UI (-8.5 dB)	-, 0.99 UI (-2.5 dB)	114 mV, 0.32 UI (-7.4 dB)	-, 0.55 UI (-8 dB)	-, 0.14 UI (-3.7 dB)	**260 mV, 0.76 UI (-3 dB)**	**270 mV, 0.7 UI (-4.2 dB)**	**160 mV, 0.42 UI (-6 dB)**	**84 mV, 0.28 UI (-9.2 dB)**
Energy Efficiency (pJ/b)	TX	0.8	0.449	1.09	0.6	-	-	-(1)		**1.07**	**1.21**
	RX	1.16	0.108	0.15	0.56	-	-	**0.08(2)**	**0.1 (2)**	**0.14(2)**	**0.18(2)**

	Total	1.96(3)	0.557(3)	1.24(3)	1.16(3)	1.7	1.36(Analog)
							0.34(Digital)
								-(1)		**1.21(3)**	**1.39(3)**

Area (μm^2) (Nomalized with tech.)	TX	312,375	2,147	1,236	1,126	≈200,000(4)	2,672,000 (Analog) 976,000 (Digital)	-(1)		**1,149**	
	RX		2,968	1,192	673	≈200,000(4)		**408**	**443**	**491**	**542**
	Total	312,375(3)	5,115(3)	2,428(3)	1,799(3)	≈400,000(4)		-(1)		**1,640(3)**	**1,691(3)**
FoM (Gb/s/mm^2)		48	4,888	8,237	8,894	≈100	42(5)	-(1)		**7,317**	**11,827**

(1) RX4 and RX5 was tested using an instrument without the transmitter.
(2) The receiver termination impedance is connected to half-VDD, and its power consumption is included.
(3) Only a SerDes front-end is included
(4) Area is very roughly estimated based on the figure because area is not clearly reported.
(5) FoM is calculated with the area of analog parts, excepting the area of digital parts.

the *in-situ* eye heights (widths) of the 260 mV (0.76 UI) and 270 mV (0.7 UI) at the data rate of 8 Gb/s and 10 Gb/s, respectively, without any equalization such as FFE, CTLE, or DFE. The RX4 and the RX5 used 1.1 V supply voltage and consumed only 76 fJ/b and 101 fJ/b, respectively.

The transceivers with the quarter-rate receivers RX6 and RX7 were tested by communicating through two PCB traces connected by three SMA connectors (Fig. 8(b)). The overall channel length is 58 mm including the two PCB traces and the three SMA connectors. The transceivers with the RX6 and the RX7 communicated over the same channel at the data rates of 12 Gb/s and 20 Gb/s and have losses of −6 dB and −9.2 dB, respectively (Fig. 8(b)). With the aforementioned conditions, the transceivers with the RX6 and the RX7 achieved the eye heights (widths) of 160 mV (0.42 UI) and 84 mV (0.28 UI) and consumed 1.21 pJ/b and 1.39 pJ/b, respectively (Fig. 9(c) and (d)).

These results demonstrate that the receiver front-ends generated within a second by the developed software reliably recovered the data at various target data rates. The proposed receivers are fully software-generated specifically targeting short-reach links for the first time, and the generated 20 Gb/s receiver demonstrated the highest data rate among the published software-generated receivers to the best of authors' knowledge.

Table II provides the detailed breakdown of both the power consumption and the area for the transmitter and the receivers. Over 85% of the total power of the transceiver was consumed in the transmitter. However, the transmitter design was not our main contribution. Our main focus is on demonstrating that reliable and very fast generation of the compact and energy-efficient receiver front-ends targeting various data rates is possible by carefully developed generator code and by selectively changing time-interleaving order. Given that short-reach links have less channel distortion, employing the simpler and lower power-consuming equalizer such as a CTLE in the receiver may be a preferable choice rather than using the FFE in the transmitter in order to achieve better energy efficiency in short-reach links [19].

Table III summarizes the performances of the proposed transceivers and compares them with the prior arts [11], [13], [14], [15], [16], [17]. Among the prior arts in Table III, [11] is the only prior art that reports the software-generated receiver although its application is long-reach links, unlike the short-reach link in this work. The proposed receiver in this work occupied significantly less area (0.0054x) and achieved better energy efficiency (0.71x) at a faster data rate (1.33x) as compared to the receiver reported in [11], being that the short-reach target does not require power-hungry and area-occupying equalizers such as FFE, CTLE, and DFE. Even compared with manually designed prior arts [13], [14], [15], [16], [17] for short-reach links, the transceivers with the software-generated receivers achieved smaller area at the competitive data rate with decent energy efficiency, while reducing the development time using the proposed receiver front-end layout generator. The transceiver with the proposed software-generated receiver achieved the highest data rate per area as well as the smallest area among the receivers reported in the relevant prior arts [11], [13], [14], [15], [16], [17]. Although the area is not clearly reported in [16], the conclusion could be drawn from a rough estimate based on figures due to the large area difference.

V. Conclusion

Single-ended transceivers with software-generated receiver front-ends are introduced for high-speed short-reach links for the first time. The software allows designers to customize the device sizes as well as the time-interleaved architecture in multiple technology nodes, significantly reducing the development time spent to optimize the receiver front-end. Even with the great layout time reduction by about up to 240k times, the software-generated receiver front-end achieved nearly the same performance as the manually-designed ones in the post-layout simulation. To verify the software, various receiver front-ends with different transistor sizes and time-interleaved architectures aiming for different target speeds were generated and fabricated in 28 nm CMOS technology with the manually-designed inverter-based FFE transmitter. The measurement results showed that all generated receivers reliably recovered the data delivered from the BERT equipment or the manually-designed transmitter at the target data rates ranging from 8 Gb/s to 20 Gb/s with decent energy efficiencies. The proposed transceiver achieved the highest data rate per area as well as the smallest area compared to prior manually-designed transceivers for short-reach links. The proposed solution could be particularly beneficial when designing transceivers with multiple design iterations and modifications, and when porting the design into different process technology nodes.

Acknowledgment

The authors would like to thank the IC Design Education Center (IDEC) and Ansys for tool supports.

References

[1] V. M. Z. Bexten, C. Moraga, R. Klinke, W. Brockherde, and K.-G. Hess, "ALSYN: Flexible rule-based layout synthesis for analog IC's," *IEEE J. Solid-State Circuits*, vol. 28, no. 3, pp. 261–268, Mar. 1993.

[2] R. Castro-Lopez, O. Guerra, E. Roca, and F. V. Fernandez, "An integrated layout-synthesis approach for analog ICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 7, pp. 1179–1189, Jul. 2008.

[3] R. Martins, N. Lourenço, and N. Horta, "LAYGEN II—Automatic layout generation of analog integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 11, pp. 1641–1654, Nov. 2013.

[4] E. Yilmaz and G. Dundar, "Analog layout generator for CMOS circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 1, pp. 32–45, Jan. 2009.

[5] J. Han, W. Bae, E. Chang, Z. Wang, B. Nikolic, and E. Alon, "LAYGO: A template-and-grid-based layout generation engine for advanced CMOS technologies," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 3, pp. 1012–1022, Mar. 2021.

[6] T. Shin et al., "LAYGO2: A custom layout generation engine based on dynamic templates and grids for advanced CMOS technologies," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, early access, Jul. 13, 2023, doi: 10.1109/TCAD.2023.3294462.

[7] H. Chen et al., "AutoCRAFT: Layout automation for custom circuits in advanced FinFET technologies," in *Proc. Int. Symp. Phys. Design*, Apr. 2022, pp. 175–183.

[8] E. Chang et al., "BAG2: A process-portable framework for generator-based AMS circuit design," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2018, pp. 1–8.

[9] M. Choi et al., "An output-bandwidth-optimized 200 Gb/s PAM-4 100 Gb/s NRZ transmitter with 5-tap FFE in 28 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 128–130.

[10] S. Han, S. Jeong, C. Kim, H.-J. Park, and B. Kim, "GUI-enhanced layout generation of FFE SST TXs for fast high-speed serial link design," in *Proc. 57th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2020, pp. 1–6.

[11] E. Chang, N. Narevsky, J. Han, and E. Alon, "An automated SerDes frontend generator verified with a 16 nm instance achieving 15 GB/S at 1.96 PJ/Bit," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2018, pp. 153–154.

[12] C. Moon, J. Seo, M. Lee, I. Jang, and B. Kim, "A 20 Gb/s/pin 1.18 pJ/b 1149μm^2 single-ended inverter-based 4-tap addition-only feed-forward equalization transmitter with improved robustness to coefficient errors in 28 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2022, pp. 450–452.

[13] J. W. Poulton et al., "A 1.17-pJ/b, 25-Gb/s/pin ground-referenced single-ended serial link for off- and on-package communication using a process- and temperature-adaptive voltage regulator," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 43–54, Jan. 2019.

[14] J. Seo, S. Lee, M. Lee, C. Moon, and B. Kim, "A 20-Gb/s/pin 0.0024-mm^2 single-ended DECS TRX with CDR-less self-slicing/auto-deserialization to improve tolerance on duty cycle error and RX supply noise for DCC/CDR-less short-reach memory interfaces," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 1–3.

[15] M. Lee, P. K. Kaur, J. Seo, S. Han, and B. Kim, "A compact single-ended inverter-based transceiver with swing improvement for short-reach links," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 9, pp. 3679–3688, Sep. 2022.

[16] K. McCollough, S. D. Huss, J. Vandersand, R. Smith, C. Moscone, and Q. O. Farooq, "A 480 Gb/s/mm 1.7 pJ/b short-reach wireline transceiver using single-ended NRZ for die-to-die applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 1–3.

[17] R. Yousry et al., "A 1.7 pJ/b 112 Gb/s XSR transceiver for intra-package communication in 7 nm FinFET technology," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 180–182.

[18] A. Cevrero et al., "A 64 Gb/s 1.4 nm CMOS FinFET," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 482–483.

[19] Optical Internetworking Forum (OIF). *112 Gb/s Electrical Interfaces—An OIF Update on CEI-112G*. Accessed: Mar. 2020. [Online]. Available: https://www.oiforum.com/wp-content/uploads/00311c-OIF-112G-OFC-slides_ofc20_presentation.pdf

Myungguk Lee (Graduate Student Member, IEEE) received the B.S. degree in electronic engineering from the Kumoh National Institute of Technology, Gumi, South Korea, in 2015, and the M.S. degree in electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include high-speed link circuits, signal/power integrity, and agile hardware design.

Jaeik Cho received the B.S. degree in electronic and electrical engineering from Hongik University, Seoul, South Korea, in 2021, and the M.S. degree in electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2023. Since 2023, he has been an Engineer with Samsung Electronics, Hwaseong-si, South Korea. His current research interests include high-speed link circuit and analog layout automation.

Junung Choi (Graduate Student Member, IEEE) received the B.S. degree in electrical and computer engineering from the University of Seoul, Seoul, South Korea, in 2021, and the M.S. degree in electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2023, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include high-speed link circuits and analog layout automation.

Won Joon Choi (Graduate Student Member, IEEE) received the B.S. degree in electronic and electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2022, where he is currently pursuing the Ph.D. degree. His research interests include high-speed link circuit and computer aided design.

Jiyun Lee (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2023, where he is currently pursuing the Ph.D. degree. His research interests include high-speed links and signal/power integrity.

Iksu Jang (Graduate Student Member, IEEE) received the B.S. degree in electrical and computer engineering from Ajou University, Suwon, South Korea, in 2018, and the M.S. degree from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2020, where he is currently pursuing the Ph.D. degree in electrical engineering. His current research interests include high-speed link circuit.

Changjae Moon (Graduate Student Member, IEEE) received the B.S. degree in electronic and electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2018, where he is currently pursuing the Ph.D. degree. His research interests include high-speed links and signal integrity.

Gain Kim (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering and the Ph.D. degree in microsystems and microelectronics from the Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, Switzerland, in 2013, 2015, and 2018, respectively. From September 2016 to July 2018, he was with the High-Speed Interconnect Technology Group, IBM Research—Zurich, Rüschlikon, Switzerland, working on the design of ADC-based wireline receivers. From 2018 to 2020, he was a Post-Doctoral Fellow with KAIST. From 2020 to 2022, he was with Samsung Research, Seoul, South Korea, as a Staff Engineer, working on a baseband modem for 6G wireless communications. In 2022, he joined the Faculty of the EECS Department, Daegu Gyeongbuk Institute of Science and Technology, Daegu, South Korea, where he is currently an Assistant Professor. He was a recipient of the 2018 IEEE Circuits and Systems Pre-Doctoral Scholarship Award.

Byungsub Kim (Senior Member, IEEE) received the B.S. degree in electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2000, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2004 and 2010, respectively.

He was an Analog Design Engineer with Intel Corporation, Hillsboro, OR, USA, from 2010 to 2011. In 2012, he joined the Faculty of the Department of Electrical Engineering, POSTECH, where he is currently a Professor.

Dr. Kim served as a member of the Technical Program Committee for the IEEE International Solid-State Circuits Conference. He received several honorable awards. He received the IEEE Journal of Solid-State Circuits Best Paper Award in 2009. In 2009, he was a co-recipient of the Beatrice Winner Award for Editorial Excellence from the 2009 IEEE International Solid-State Circuits Conference. He has been serving as the Chair for Wireline Sub-Com of the IEEE Asian Solid-State Circuit Conference.