

# OpenSerDes: An Open Source Process-Portable All-Digital Serial Link

Gaurav Kumar K, Baibhab Chatterjee and Shreyas Sen

School of ECE, Purdue University, West Lafayette, USA-47906. email:{gauravk, bchatte, shreyas}@purdue.edu

**Abstract**—Over the last decade, the growing influence of open source software has necessitated the need to reduce the abstraction levels in hardware design. Open source hardware significantly reduces the development time, increasing the probability of first-pass success and enable developers to optimize software solutions based on hardware features, thereby reducing the design costs. The recent introduction of open source Process Development Kit (OpenPDK) by Skywater technologies in June 2020 has eliminated the barriers to Application-Specific Integrated Circuit (ASIC) design, which is otherwise considered expensive and not easily accessible. The OpenPDK is the first concrete step towards achieving the goal of open source circuit blocks that can be imported to reuse and modify in ASIC design. With process technologies scaling down for better performance, the need for entirely digital designs, which can be synthesized in any standard Automatic Place-and-Route (APR) tool, has increased considerably, for mapping physical design to the new process technology. This work presents a first open source all-digital Serializer/Deserializer (SerDes) for multi-GHz serial links designed using Skywater OpenPDK 130nm process node. To ensure that the design is fully synthesizable, the SerDes uses CMOS inverter based drivers at the transmitter, while the receiver front end comprises a resistive feedback inverter as a sensing element, followed by sampling elements. A fully digital oversampling CDR at the receiver end recovers the transmitter clock for proper decoding of data bits. The physical design flow utilizes OpenLANE, which is an open source end-to-end tool for generating GDS from RTL. Cadence Virtuoso has been used for extracting parasitics for post-layout simulations, which exhibit the SerDes functionality at 2 Gbps for 34 dB channel loss while consuming 438 mW power. The generated GDS and netlist files of the SerDes, along with the required documentation, are uploaded in a GitHub repository for public access.

**Index Terms**—OpenSerdes, Open source, OpenPDK, Skywater

## I. INTRODUCTION

### A. The Need for open source Hardware

With the predominance of open source software in recent times, the term “open source hardware”, introduced by Bruce Perens in 1997, is gaining momentum in the design ecosystem [1], [2]. Open source hardware refers to a hardware, an ASIC, or even, a circuit that has provided access to its entire design, specifications, and documentations, which can be used, altered, or distributed by anyone. Like source code in case of open source software, all the schematics, logic designs, layout-data, and netlists need to be made available for revisions by anyone, who has access to the tools to read, manipulate and update the existing design with new features, usually aiming for better performance and share the improved design back to the community for further enhancements/evaluations.

Open source hardware reduces the design time, enables software developers to effectively optimize their source codes based

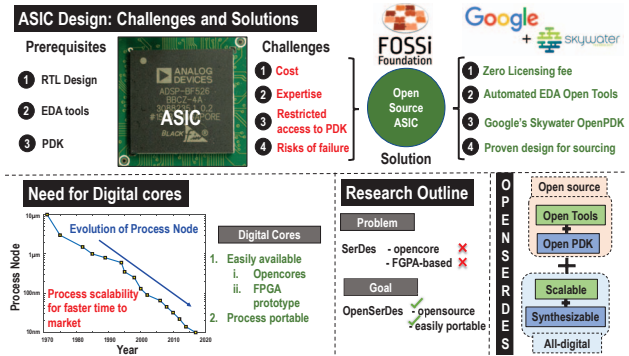


Fig. 1. Motivation for open source hardware design and OpenSerDes

on hardware features, accelerates innovations with multiple people working together and sharing their knowledge, reducing the cost. This illustrates the need for open source hardware for faster technological advancement.

Several Open source hardware projects, like RISC-V [3], UC Berkley’s BAG [4], Google’s OpenTitan [5], OpenRoad [6] etc., have started uploading their design specifics in the repositories for public use. Berkeley Analog Generator (BAG) [7] is an open source tool that provides framework for generating Analog and Mixed Signal (AMS) circuits from the design specifications. Numerous Open source tools like magic, ngspice, etc. have been used extensively for design layout and simulations. The Free and Open Source Silicon (FOSSi) foundation promotes free and open digital hardware designs, supporting open standards development. The recent introduction of Skywater 130 nm open source PDK [1], [8] by Google and Skywater Foundries, has opened several opportunities for circuit designers. The conventional PDKs from TSMC, Global Foundries, UMC etc. are quite expensive and involve strict licensing process, making them difficult to access for all. OpenPDK has removed the roadblocks for ASIC designing, providing access to make custom chips for free initially for faster development cycle or at very low-cost eventually.

With the process technologies scaling down for better performance, the existing designs or circuits need to be ported to latest process, reinstating the fact that “process-portability” of the design plays a key role in selecting the circuit topology. Although Analog/Mixed signal circuit design approach may exhibit superior performance than constituent synthesizable digital-only circuits, in terms of speed, power, etc., the entire design needs to be redrawn from scratch, to port to a new process. This promotes for synthesizable, digital-friendly designs

that can be easily ported from one process to another.

SerDes, an important building block in communication systems, helps in transmitting a parallel stream of data from block to another over serial links, thereby reducing the number of I/O pins, device dimensions, and power consumption of the system. The unavailability of opencore or FPGA-based digital IO/SerDes prevents faster time-to-market for designs, as they demand considerable time and effort for porting across technologies. This work presents the functionality of fully open source, multi-GHz, all-digital SerDes link, built using Skywater 130nm Open PDK, ensuring process-portability, thereby reducing the overhead porting time.

### B. Our Contribution

- This work presents the first open source **all-digital SerDes**, with multi-GHz capabilities. Unlike traditional mixed-signal SerDes designs, the all-digital design makes the present design **process-portable**, which is a key aspect of **Open-Source Hardware**.
- This work uses, for the first time, an **open source PDK**, skywater 130 nm, for the realization of **SerDes**.
- The work illustrates the importance of open source hardware and synthesizable design techniques, motivating their adoption in circuit design.

### C. Paper Organization

The remainder of the paper is organized as follows. Section II illustrates the importance for open source hardware by comparing with traditional design techniques. In Section III, the typical architecture of the SerDes link is demonstrated, discussing the various design considerations, while promoting the need for all-digital design. Section IV presents the architecture of the automated SerDes design. Section V discusses the implementation results of the SerDes. Section VI explains the pros and cons of the design with use-cases and design flow of the OpenLANE tool. We conclude the paper in Section VII.

## II. OPEN SOURCE HARDWARE VS. TRADITIONAL DESIGN

Open hardware removes the common hindrances to system design and ensures faster development time. Hardware/ASIC designers can reduce the risk of design failure by leveraging pre-existing designs in an open source hardware library, which enables a “plug and play” feature for previously proven successful hardware constructs, which was not possible earlier [2]. Designers possess the complete design that can be modified based on their required specifications. This prevents re-inventing the designs, enables rapid innovation and progress, while lowering the design cost. Figure 2 shows the comparison between the costs incurred in manufacturing chip using traditional PDKs (i.e., TSMC, GF, Intel, etc.), and skywater 130nm openPDK. It is seen that openPDK reduces the PDK licensing cost, bringing down the overall chip manufacturing cost. Note that the PDK licensing cost is not publicly accessible and thus, it has been scaled relative to fabrication cost and process technology [9].

The involvement of community facilitates faster evolution and reception of the design [10]. It enables people to interact and share their knowledge of hardware design, thereby opening

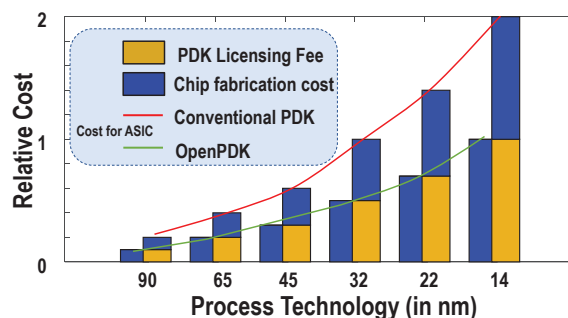


Fig. 2. Comparison of Relative chip fabricating cost using Open source PDK and traditional PDKs [9]

more avenues for innovation. It also provides the software developers the ability to optimize and enhance their solutions based on the hardware features, impacting the overall performance of the system. Ultra-low power pervasive IoT, being at the juncture of Moore’s Law and Shannon’s law, calls for innovation and collaboration at both software and hardware fronts [11], and hence would benefit immensely from open-source hardware designs.

Open source hardware has started to gain prominence in the design space, with numerous open source projects, like BAG, RISC-V, OpenTitan, OpenPDKs etc., promoting open standard developments. In a nutshell, open source hardware provides people with a platform to share their knowledge and determine the course of the technology.

## III. SERDES DESIGN CONSIDERATIONS

SerDes has emerged as a popular solution in communication system design since the paradigm shift away from traditional parallel communication standards to overcome the voltage scaling dependencies and performance limitations. Serial links, unlike parallel implementations, are scalable and require fewer I/Os, achieve smaller device footprints, thereby having lower power requirements. The growing demand for low-power high-speed link has motivated development of innovative circuit techniques for SerDes over the last few years [12]–[14]. Mixed-signal designs, having multiple feedback loops from analog and digital domains, have presented a cost-effective solution, reaping the benefits of both analog and digital worlds. But with consequent technology scaling, requiring faster time-to-market for lower process node designs, there is a need for all-digital SerDes architecture, which can be easily ported across technologies.

Figure 3 shows the block-level architecture of a typical SerDes. The transmitter (TX) front end comprises of a serializer, TX equalization block and a driver, while the Receiver (RX) front end has RX equalization, sampling element and a deserializer [15]. In case of non-forwarded-clock architecture, where timing signal of the transmitter, i.e. clock, is not transmitted along with data, a Clock and Data Recovery (CDR) circuit recovers the timing information and facilitates correct decoding of received data.

TX Feed Forward Equalizer (FFE), used as TX equalization block, pre-distorts the data over some bit durations to negate the

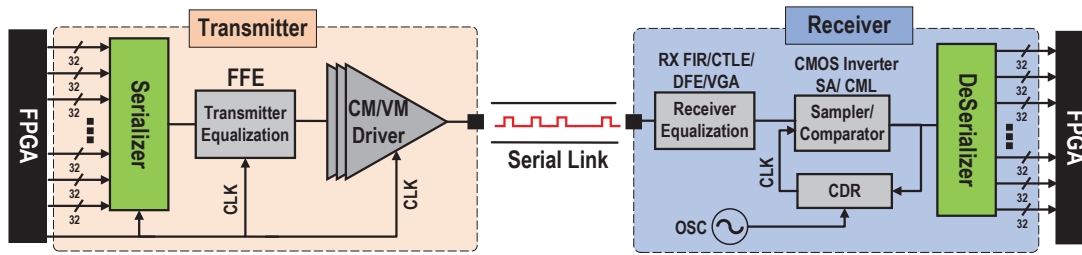


Fig. 3. Block-level overview of a typical SerDes Architecture

effect of channel loss/distortion. Transmit drivers, implemented either in Voltage Mode (VM) or Current Mode (CM) using differential amplifiers, provide required output voltage swing and match the channel impedance to reduce reflections. RX equalization improves signal gain and input referred noise of the received data. It is achieved using a Continuous Time Linear Equalizer (CTLE), a Decision Feedback Equalizer (DFE) or FIR filter based FFE [16]. Comparator/Sampler blocks resolve the data to logic levels based on some threshold or differential value. They can be realized using static amplifiers or clocked regenerative amplifiers. Regenerative comparators, such as a Strong Arm Latch [17] samples the input at clock edges and resolves the differential to logic levels by positive feedback based latching action. Recently, high speed ADCs are used as comparators to generate digital data from received signal [18]. A CDR extracts the timing information from the incoming data transitions, by adjusting the phase of the sampling clock, and uses this recovered clock to resample the data and decode the data correctly.

Our earlier works [13] [14] demonstrate high speed bi-directional electrical link architecture, with TX and RX Equalization, for Proximity communication in 22nm and 14nm CMOS technologies. However, these designs consist analog/mixed-signal circuits for higher performance, and hence is not amenable to a fully digital implementation.

#### A. Need for All-digital design

Although mixed signal systems offer greater speeds, higher sensitivity, and better power efficiency than both analog and digital counterparts, the push for an all-digital design is garnering much interest. As the technology scales down to a new one, digital designs ensure faster time to market than analog or mixed signal designs because significant amount of time is spent on refabricating the entire analog design from scratch with the updated design rules of new technology, whereas in case of all-digital designs, the complete circuitry can be obtained by running the RTL through Synthesis and APR tool that maps the design to latest technology, abiding all the latest design rules. This saves considerable amount of man-hours. Innovations in technology scaling is primarily focused on the performance of digital systems, compromising the performance of analog circuitry, because of the occurrence of various non-ideal effects and increased variability in MOS device, in such smaller dimensions and lower supply voltage regime. Fully synthesizable designs help in reaping full benefits from new

technology, where lower voltage supplies limit the circuit and performance feasibilities of analog circuits.

Note that all-digital designs are well suited in scenarios where the possibility of technology scaling is imminent (in case of Skywater 130 nm OpenPDK). The main challenge is to maximize the performance of all-digital design, in terms of speed, sensitivity and power consumption, to match up to the standards of mixed signal design.

### IV. AUTOMATED SERDES DESIGN

An all-digital SerDes is generated using the OpenLANE Tool [19], which provides an open source environment for automatic RTL to GDS generation without manual intervention. The primary goal is to ensure that the design is fully synthesizable, without compromising the performance of the SerDes, making it easier to port from one process technology to another. The design comprises of a serializer and CMOS inverter based transmit driver as a transmitter, a resistive-feedback inverter-based [20] sensing element and a CMOS inverter to generate rail-to-rail signal, with a flip-flop-based sampling element, followed by the deserializer, as receiver. An oversampling [21] CDR facilitates the proper detection of data by generating correct timing information based on received data. The complete design is available at: <https://github.com/SparcLab/OpenSERDES>.

#### A. Transmitter Architecture

The transmitter converts the input parallel data streams into serial bits, before transmitting them. It comprises of serializer followed by driver to drive the output voltage to the channel.

a) *Serializer*: The Serializer block is coded in Verilog HDL and synthesized using OpenLANE tool to generate the GDS, mapped to Skywater 130nm technology. Considering realistic scenarios, the serializer is designed to take in 8 parallel input data streams of 32 bits each and produces serial bits. It is implemented using a Finite State Machine (FSM) that sequentially takes one parallel data stream and serializes them.

b) *Transmit Driver*: A voltage-Mode (VM) CMOS driver is chosen provide rail-to-rail input swing at channel input. VM drivers have an advantage of reduced driver power, as compared to current-mode (CM) drivers. A three-stage inverter chain is used to implement the driver, to achieve the driving capability of 2 pF. Note that the individual CMOS stages are sized appropriately to obtain area and power optimal design. Figure 4 shows the schematic representation of designed CMOS inverter-based driver, with related input-output waveforms.

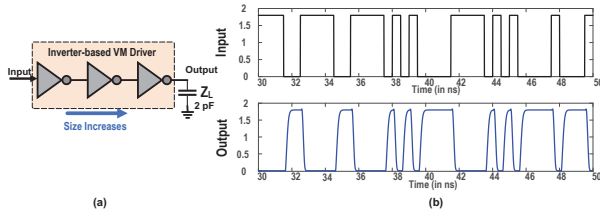


Fig. 4. (a) Schematic of CMOS Transmit Driver (b) Input/Output Waveforms of Driver at 2 Gbps with 2 pF termination

### B. Receiver Architecture

The receiver front end uses a resistive feedback inverter as gain and sensing element and a CMOS inverter based static comparator for sampling the received signal. These blocks are chosen to ensure that the SerDes receiver is synthesizable and achieves desired performance criteria. Figure 5 presents the complete receiver architecture of the SerDes.

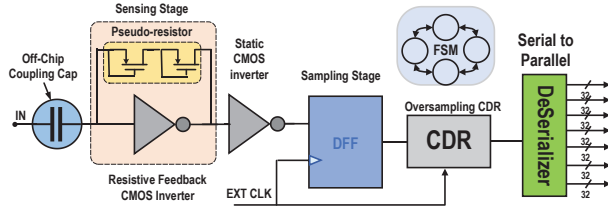


Fig. 5. Receiver architecture of presented SerDes

*a) Resistive Feedback Inverter:* The Resistive feedback inverter [20] is a self-biased CMOS inverter with a resistive feedback from output to input, aiming to bias the inverter near the switching threshold, i.e.  $0.5V_{DD}$  point, enabling the CMOS inverter to work on signals with amplitudes much lower than its threshold voltage. To ensure synthesizability, the feedback resistor is realized as pseudo resistor using PMOS, where the gate and source are shorted to operate in linear resistive region. An off-chip capacitor is used to AC couple the received signal and prevent the received signal to interfere with the generated self-bias voltage, thereby maintaining the DC bias. Since the inverter operates at the point where both PMOS and NMOS are in saturation, there always exists a path for current to flow from supply to ground, resulting in non-zero static power dissipation. Figure 6 shows the operating point of the designed self-bias inverter, with waveforms depicting its functionality.

*b) Sampling Block:* The sampling block consists of CMOS inverter, which provides rail-to-rail signal at the input of D flip-flop, which then samples and forwards the received data to the deserializer, using the timing information provided by clock signal. Although the realized sampling topology has limited gain as compared to regenerative samplers/comparators, the front-end resistive feedback inverter compensates and provides the necessary gain.

*c) Deserializer:* The deserializer block converts serial bits into parallel data streams, which are to be interfaced with other blocks. It is implemented through an FSM that takes in serial

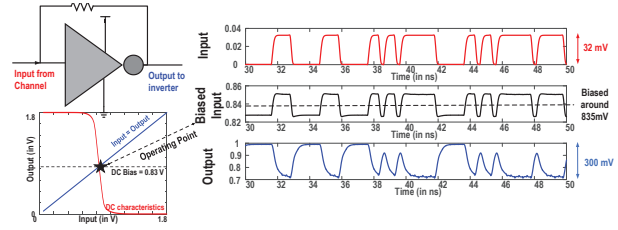


Fig. 6. (a) Operating point, (b) Input/Output Waveforms, of Resistive feedback inverter

bits and produces 8 parallel data streams of 32 bits each at the end of its operation. The entire design is synthesized using OpenLANE tool to generate the GDS and netlists.

### C. Oversampling CDR

A fully digital oversampling CDR [21] [22] is implemented to recover the clocking information. It samples the received data at multiple points, stores them in FIFO registers, before determining the optimal sampling point, which is used to correctly determine the transmitted data. The CDR is also equipped with tunable glitch and jitter correction logic using external scan bits. An external clock is fed to the phase generator block to produce multiple clock phases needed for sampling the input at different points. The sampled data is sent for processing and decision-making, using the predefined logic of the decision block and external glitch and jitter corrections inputs. Figure 7 shows the block-level architecture of implemented Oversampling CDR.

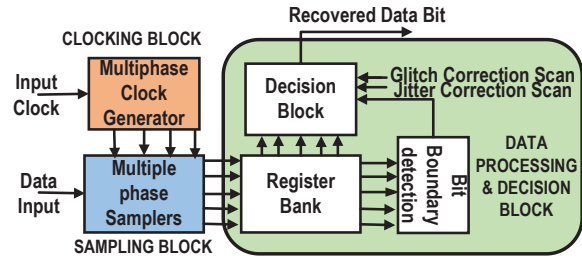


Fig. 7. Architecture of implemented Oversampling CDR

## V. RESULTS

The SerDes design (in GDSII format) is imported in cadence virtuoso to extract parasitics and perform post-layout simulations. Figure 8 shows the input/output waveforms of SerDes link at 2 GHz with PRBS-31 input. The received signal represents the signal reaching the RX front end, after channel attenuation of 34 dB. In contrast with Strong-Arm Latch [17] based receiver designs, which are sensitive to even 10 mV signals at 2 GHz, the presented all-digital design achieves  $\approx 32$  mV sensitivity, which is promising in several applications like PCIe interfaces, short-range interconnects, where the required bandwidths are less and channel losses are not significant. The variation of sensitivity of SerDes receiver with different operating frequencies is depicted in Figure 9. Note that the maximum channel loss corresponding to zero BER transmission is also shown.

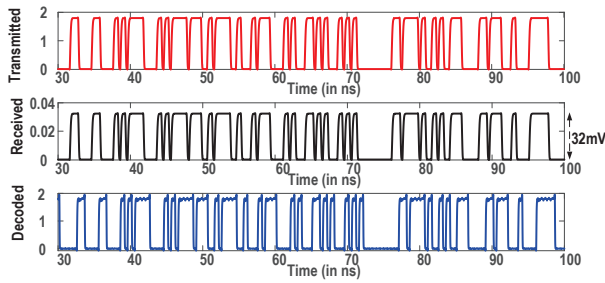


Fig. 8. Input/Output Wave-forms at 2Gbps for the SerDes link.

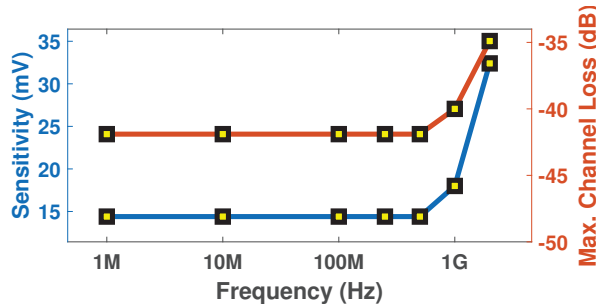


Fig. 9. Variation of Sensitivity and maximum operating channel loss of SerDes Receiver with frequency of operation

Figure 10 provides the power consumption (excluding serializer, deserializer and CDR blocks) and area of the all-digital serial link. The link needs total power of 15.7 mW operating at 2 GHz for the supply of 1.8V, with the receiver and transmitter consuming 11.2 mW, and 4.5 mW respectively. Serializer, Deserializer and CDR circuits require additional 235mW, 128mW and 59mW respectively. The high power numbers of serializer and deserializer blocks are the result of intended design choices to support large IO streams (eight parallel links with 32-bit input) for communication. Thus, the implemented all-digital SerDes operates at 2 GHz for 34 dB channel loss, while consuming power of 437.7 mW, thereby exhibiting energy efficiency of 219 pJ/bit. Figure 11 presents the layout of the design. The design layout extends over an area of  $0.24\text{mm}^2$ , with deserializer occupying 60% of the area, while the CMOS driver and RX Front end cover 0.2% and 1.1% respectively.

## VI. DISCUSSIONS

a) *Pros and Cons of the presented SerDes design:* The SerDes is equipped with the capability to support multi-GHz transmission, in addition to being a fully synthesizable design, ensuring process-portability. Except for serializer, deserializer and CDR, the entire serial link is implemented using a few CMOS inverters and MOS devices, thereby requiring less chip area compared to mixed signal circuit design, which require additional complex circuitry and I/O pins to provide appropriate biasing for functioning.

However, compared to conventional mixed signal circuit approach, the presented SerDes exhibits slightly lesser sensi-

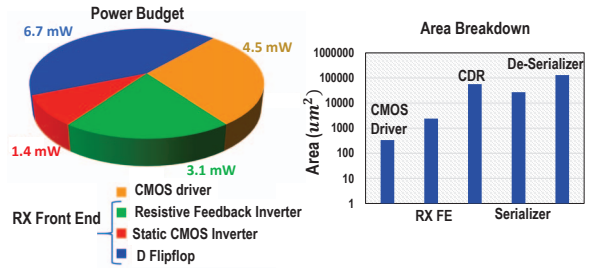


Fig. 10. Power Budget and Area Breakdown of the designed Serial link.

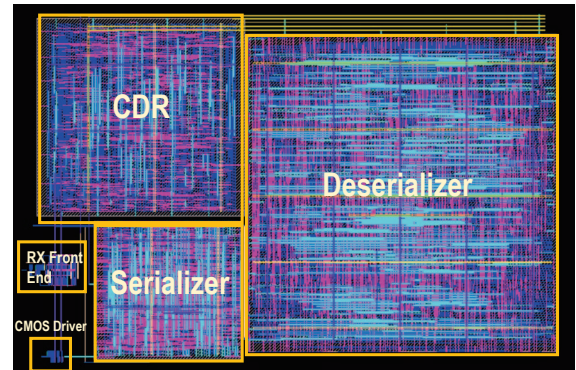


Fig. 11. Layout of the generated SerDes. The complete design is available at <https://github.com/SparcLab/OpenSERDES>

tivity, and operates at lower speeds, making it appropriate for applications, where the required operating speeds are smaller (1-2 GHz), channel losses are moderate (34dB - 40dB).

b) *Applications:* The generated SerDes can be used to interface links where the bandwidth requirements is  $< 2$  Gbps, like the PCIe 1.x, PCIe 2.x, PCIe 3.x, PCIe 4.0 bus, where the required Bandwidth/lane ranges from 250 Mbps to 2 Gbps. The SerDes can also find applications in high bandwidth density, short-range interconnects between chiplets, like Intel's recent high density, multi-die packaging paradigm, Embedded Multi-die interconnects (EMIB) [23], where the primary focus is on the area of the serial link, requiring data speeds in the range of 1-4 GHz in a significantly low channel loss setting of 1-5 dB.

c) *OpenLANE Tool flow:* The OpenLANE tool [19], an open source tool with automated RTL to GDS flow, is used to synthesize the RTL and generate GDS for the SerDes design. The OpenLANE tool flow uses custom methodology scripts that perform design exploration and optimization and chose the best implementation, using a number of constituent open source tools, like magic, yosys, OpenSTA, etc., which are invoked at different times in the flow, to execute the Synthesis, APR, Verification and Export steps of the ASIC design flow. Figure 12 shows the overview of OpenLANE tool flow, also indicating the open source tools used in each step of the flow.

## VII. CONCLUSION

The common roadblocks to hardware/ASIC designs, i.e., the high commitment costs, the much-needed expertise of the

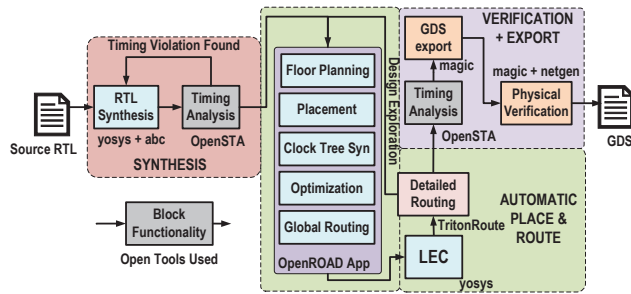


Fig. 12. Design Flow of OpenLANE tool

designer and the uncertainty (risk) of design functionality, prevent the designers from the required access to advanced technologies. With main objective to democratize the hardware design, the open source hardware revolution is gaining interest, with several open source EDA tools and designs are beginning to surface for public use. These designs and tools can be sourced, modified, or enhanced by the designers, thereby eliminating the involved design cost and the risk factors. The tools are incorporated with machine learning and optimization capabilities, with automated flow, that generate the fully optimized implementation of complex designs. This significantly brings down the applied human effort, and removes the level of needed expertise through automated open source popular EDA tools such as magic, ngspice, OpenRoad, OpenLANE, BAG etc. The advent of open source PDK by Google and Skywater technologies has provided designers with free access to the CMOS 130 nm process, equipping them with standard library needed for generating designs. Thus, Open source hardware significantly reduces the development time and costs for the design. Technology scaling is also shifting the focus to digital-only designs as analog circuits suffer from short channel effects and degraded performance at low voltage supplies. All-Digital or synthesizable designs also enable faster time-to-market than analog designs. With the development of APR tools, the all-digital designs can be easily ported to new process with minimal effort. Consequently, all-digital designs are being preferred today over analog or mixed signal designs.

The work presents the first open source SerDes using Skywater 130 nm OpenPDK, aiming to build a synthesizable design, ensuring process-portability, without compromising with the performance of the design. Simulation results show that the designed SerDes operates at speed of 2 GHz, has a receiver sensitivity of 32 mV to cope up with the channel losses of 34 dB, while consuming power of 438 mW. The generated SerDes finds applications in interfaces, where the required data rates are not very high, for example in PCIe bus interfaces and in short-reach interconnects between chiplets. The data speed and power is primarily limited by the Skywater 130 nm, the only Open PDK available presently, and will improve significantly as new OpenPDK's are made available. The presented work signifies the beginning of the open source ASIC designing, especially in traditionally mixed-signal-heavy circuits such as the SerDes, and may serve as a guide to prospective hardware/ASIC design-

ers, illustrating the importance of open source hardware, and motivating them towards digital-friendly designs using open source tools and PDK.

#### ACKNOWLEDGMENT

This work was supported in part by the Semiconductor Research Corporation (SRC) CHIRP Program, under Grant No. 2878.014 and the National Science Foundation (NSF) Career Award under Grant No. ECCS 1944602. The authors would like to thank Nirmoy Modak and Debayan Das, Ph.D. students at Purdue University for their valuable feedback during writing the manuscript.

#### REFERENCES

- [1] P. Wagner. "Produce your own physical chips. For free. In the Open.", [Online]. Available: <https://fossi-foundation.org/2020/06/30/skywater-pdk>, June, 2020.
- [2] C. Tozzi. "Open Source Hardware: What It Means and Why It Matters", [Online]. Available: <https://www.channelfutures.com/open-source/open-source-hardware-what-it-means-and-why-it-matters>, 2016.
- [3] "RISC-V", [Online]. Available: <https://github.com/riscv>.
- [4] "Berkeley analog generator, main framework", [Online]. Available: [https://github.com/ucb-art/BAG\\_framework](https://github.com/ucb-art/BAG_framework).
- [5] "OpenTitan", [Online]. Available: <https://github.com/lowRISC/opentitan>.
- [6] "The-OpenROAD-Project", [Online]. Available: <https://github.com/The-OpenROAD-Project>.
- [7] E. Chang et al. "BAG2: A process-portable framework for generator-based AMS circuit design". In *2018 CICC*, pages 1–8, 2018.
- [8] "Skywater", [Online]. Available: <https://github.com/google/skywater-pdk>.
- [9] B. Holt. "Advancing Moore's Law", [Online]. Available: <https://www.intel.com/content/www/us/en/silicon-innovations/advancing-moores-law-in-2014-presentation.html>, 2015.
- [10] V. Tigadi. "Take advantage of open-source hardware". Available: <https://www.edn.com/take-advantage-of-open-source-hardware/>, 2016.
- [11] B. Chatterjee, N. Cao, A. Raychowdhury, and S. Sen. "Context-Aware Intelligence in Resource-Constrained IoT Nodes: Opportunities and Challenges". *IEEE Design & Test*, 36(2):7–40, 2019.
- [12] B. Razavi. "Historical Trends in Wireline Communications: 60? Improvement in Speed in 20 Years". *IEEE Solid-State Circuits Magazine*, 7(4):42–46, 2015.
- [13] T. Musah et al. "A 4–32 Gb/s Bidirectional Link With 3-Tap FFE/6-Tap DFE and Collaborative CDR in 22 nm CMOS". *JSSC*, 2014.
- [14] C. Thakkar et al. "A 32 Gb/s Bidirectional 4-channel 4 pJ/b Capacitively Coupled Link in 14 nm CMOS for Proximity Communication". *JSSC*, 2016.
- [15] W. J. Dally and J. W. Poulton. *Digital Systems Engineering*. Cambridge University Press, USA, 1st edition, 2008.
- [16] B. Razavi. *RF Microelectronics (2nd Edition)*. Prentice Hall Press, USA, 2nd edition, 2011.
- [17] B. Razavi. "The StrongARM Latch [A Circuit for All Seasons]". *IEEE Solid-State Circuits Magazine*, 7(2):12–17, 2015.
- [18] J. Im et al. "6.1 A 112Gb/s PAM-4 Long-Reach Wireline Transceiver Using a 36-Way Time-Interleaved SAR-ADC and Inverter-Based RX Analog Front-End in 7nm FinFET". In *2020 ISSCC*, pages 116–118, 2020.
- [19] "OpenLANE", [Online]. Available: <https://github.com/efabless/openlane>.
- [20] B. Woorham. "CMOS Inverter as Analog Circuit: An Overview". In *Journal of Low Power Electronics and Applications*, volume 9, page 3:26, 2019.
- [21] S. I. Ahmed and T. A. Kwasniewski. "Overview of oversampling clock and data recovery circuits". In *Canadian Conference on Electrical and Computer Engineering, 2005.*, pages 1876–1881, 2005.
- [22] C. Hung, M. Chiang, A. Lee, and S. Lee. "A Sub-1V CMOS 2.5Gb/s Serial Link Transceiver Using 2X Oversampling". In *2005 IEEE Asian Solid-State Circuits Conference*, pages 37–40, 2005.
- [23] R. Mahajan et al. "Embedded Multi-die Interconnect Bridge (EMIB) – A High Density, High Bandwidth Packaging Interconnect". In *2016 ECTC*, pages 557–565, 2016.