SEMARAK ILMU PUBLISHING

Semarak Engineering Journal

Journal homepage:
https://semarakilmu.my/index.php/sej/index
ISSN: 3036-0145

Integrating Design for Testability Technique into OpenLane with Skywater 130-Nanometer Process Design Kit

Calvin Chew[1], Chia Yee Ooi[1,*], Chuan Ning Chye[2]

[1] Electronic System Engineering Department, Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kampung Datuk Keramat, 54100 Kuala Lumpur, Malaysia
[2] SkyeChip Sdn Bhd, 1-18-12, Suntech @ Penang Cybercity, Lintang Mayang Pasir 3, Bandar Bayan Baru, 11950 Bayan Lepas, Penang, Malaysia

ARTICLE INFO

Article history:
Received 15 September 2023
Received in revised form 15 October 2023
Accepted 10 December 2023
Available online 28 December 2023

Keywords:
Design-for-testability; design rule check; layout versus schematic; OpenLane

ABSTRACT

This paper highlights the key findings and outcomes of a case study on the implementation of the SHA-1 design using OpenLANE and the SkyWater 130-nanometer process design kit. The study demonstrates the successful execution of various analyses and checks, including static timing analysis, design rule check (DRC), and layout vs schematic (LVS) verification. The results indicate that the design meets timing requirements, complies with manufacturing regulations, and accurately reflects the intended schematic circuit. The implementation of a scan chain in the design using Fault is also discussed. The study further explores the area, power consumption, and timing analysis of the SHA-1 chip, providing insights for optimization and future developments. The result of this project has passed the design rule check and layout vs schematic with the period of 11.87ns and 14.62ns for case 1 and case 2 respectively. Overall, this case study emphasizes the efficacy and reliability of OpenLANE in RTL-to-GDS implementation and underlines the commitment to open-source development.

1. Introduction

Open-source EDA (Electronic Design Automation) refers to the use of software tools released under an open-source license for designing electronic devices, such as integrated circuits, printed circuit boards, and field-programmable gate arrays. Open-source EDA tools allow developers to customize, modify, and share their designs [2,11], as well as to collaborate with other users and communities. The tools are capable of producing physical layouts for realistic examples [20]. The release of Skywater Technologies' open-source Process Development Kit (PDK) in June 2020 has removed the obstacles to ASIC design, which was previously thought to be expensive and difficult to acquire [8,13,15]. To achieve the aim of open-source circuit blocks that may be imported for reuse and modification in ASIC design, the PDK is the first concrete step. To map physical design to the new process technology, the demand for entirely digital design, which can be synthesised in any common Automatic Place and-Route (APR) tool. It has significantly grown as process technologies scale down

* *Corresponding author.*
E-mail address: ooichiayee@utm.my

https:/doi.org/10.37934/sej.3.1.1421b

for improved performance. To create GDS from RTL, the physical design flow makes use of OpenLANE, an open-source end-to-end tool [4,12]. OpenLane combines different free and open-source tools capable of performing such flow [18].

The beginning of open-source hardware as a conscious movement may be traced back to Bruce Perens' introduction of an open hardware certification scheme in the late 1990s, which received the support of several business partners. Perens was also one of the key protagonists in the actual beginning of the open-source software movement. Writing software at an earlier time required a thorough understanding of hardware because many applications were written in assembly code and were less portable than it is nowadays [3]. Open-source hardware is a term used to describe hardware, an ASIC, or even a circuit that has made its full design, specification, and documentation available to anybody for use, modification, or distribution. All schematics, logic designs, layout data, and netlists must be made available for revision by anyone with access to the tools to read, manipulate, and update the existing design with new features, typically aiming for better performance and sharing the improved design back to the community for additional improvements. Open-source hardware speeds up breakthroughs by allowing more people to collaborate and share information, cutting costs while lowering design time and allowing software developers to efficiently optimise their source codes based on hardware capabilities. This shows why open-source hardware is essential for accelerating technical progress [4]. It is noteworthy that the open-source hardware developments have even opened new opportunities in the space sector – which is traditionally characterised by very low volumes and very high non-recurrent (NRE) costs [17].

The eFabless Corporation and its partners created OpenLANE, an open-source ASIC design tool, to offer a clear and open platform for creating integrated circuits. It automates the RTL-to-GDSII process by integrating a number of open-source EDA tools, including Yosys, ABC, OpenROAD [16], and Magic. OpenLANE supports several foundries and process technologies with its extensive collection of design rules and standards. Its openness encourages cooperation, information exchange, and ongoing improvement among ASIC design professionals [7].

With design-for-testability (DFT), the testing process can be made more effective, cutting down on testing costs and time, by designing a chip with built-in testability capabilities. This could result in significant manufacturing cost reductions. The DFT functionality is currently not implemented in OpenLANE flow [9]. Without DFT, the design has a negative impact on reliability and raises failure rates. Additionally, it is necessary to translate the SHA-1 ASIC design from RTL to GDS using OpenLANE to produce a manufacturing-ready layout for little to no money. Although OpenLANE is free to the public, one of the challenges is its installation and setup of the open source that consists of multiple tools. Besides, it requires significant efforts to master each tool but documentation and guidelines related to the tools may not be sufficient to support the users who are new to the tools.

This paper is to demonstrate how to integrate Fault into the open-source EDA tool, OpenLANE to enable the design-for-testability feature. We also implement the ASIC design of SHA-1 as a case study from RTL to GDS using OpenLANE to demonstrate the possibility of producing a manufacturing-ready layout at essentially zero cost. The paper outlines are as follows. Section 2 details the ASIC design flow including the addition of the DFT feature into the flow.

2. ASC Design Methodology with DFT

The methodology is a thorough way to build digital integrated circuits using open-source tools and a case study SHA-1. Providing design inputs like RTL code and constraint is the first step in the procedure. The RTL code is then synthesized into a gate-level netlist using tools like Yosys and abc. The chip layout is then established in the floorplan stage, and standard cells are then put in the core

area of the chip to maximize the use of available space. Clock Tree Synthesis (CTS) provides balanced clock distribution to reduce clock skew while routing creates connections between cells to minimize wirelength [10]. Compliance with manufacturing regulations and anticipated circuit behavior is ensured by design rule checking (DRC) and layout vs schematic (LVS). Power distribution network design ensures proper power delivery, and timing analysis verifies timing constraints. Design closure is attained by resolving any problems and fulfilling all criteria throughout the process, especially on capacitance and slew violation. The OpenLANE methodology harnesses the power of open-source tools, enabling efficient and collaborative ASIC design, reducing time-to-market, and promoting innovation within the open-source community. Figure 1 shows the flow chart of the process from RTL to GDS.

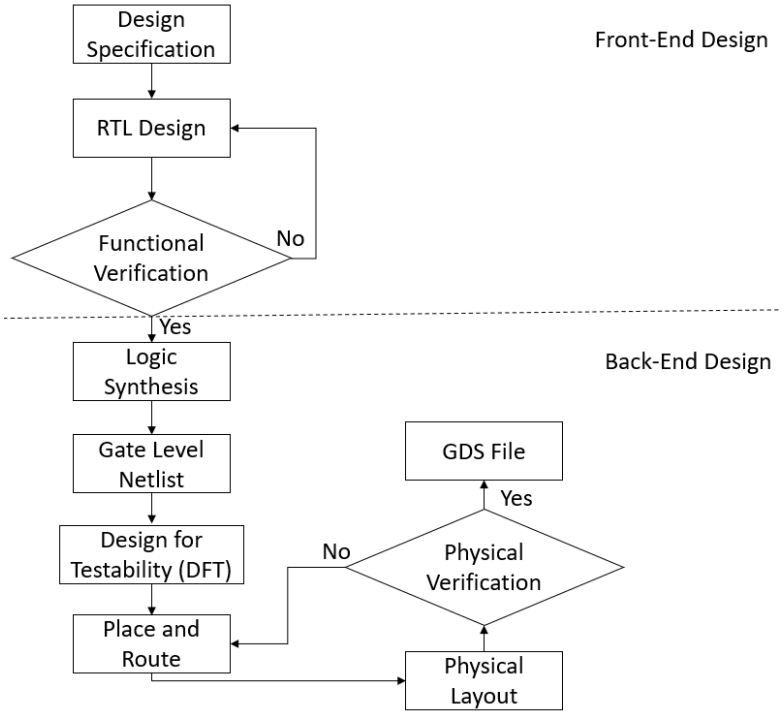

Fig. 1. Flow From RTL to GDS

The following sub-section details the DFT tool called Fault which is integrated into the OpenLANE ASIC design flow.

2.1 Design for test toolchain, Fault

The netlist can optionally be modified by including scan chains and the appropriate IO ports to scan and test the design after manufacturing using an open-source Design For Testability (DFT) toolchain, Fault [9]. Fault is implemented in Swift, which is a statically typed, secure, native programming language that can also interface with Python-based libraries idiomatically. To seamlessly interact with the Pyverilog [14] library, which generates an abstract syntax tree for direct manipulation and is required for the cutting behaviour, scan chain stitching, and other things, Fault makes use of the Swift-Python interoperability developed by Google Inc. as part of their Swift for Tensorflow project. While Swift's native programming language is more memory-friendly than Python, which gives compiled Python a small speed bump, this is a huge advantage when modelling complex hardware designs [1]. Figure 2 shows the process flow in Fault.

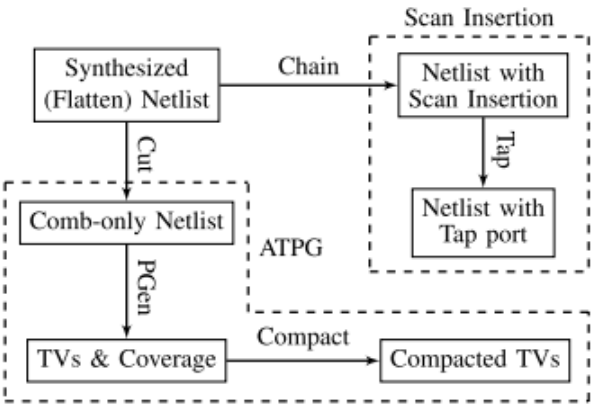

Fig. 2. Flow in Fault

Fault is made up of five parts: Cut, PGen, Compact, Chain, and Tap. It uses Verilog-synthesized netlists as its input. First, Cut is used to transform the flattened netlist into a pure combinational design. Automatic Test Pattern Generation, or ATPG, is a process used in semiconductor electrical testing wherein the vectors or input patterns required to check a device for faults are automatically generated by a program. The automatic test pattern generation (ATPG) process performed by PGen uses this amended netlist, and the produced test vectors (TVs) and final coverage are output in javascript object notation (JSON) format. After that, the created TV set is compressed by Compact, which lowers the number of generated TVs without compromising coverage. Finally, Chain inserts a scan chain, and Tap sews a JTAG controller to the inserted scan chain [1].

To make each D flip-flop accessible while creating test patterns, the circuit is "Cut" by substituting each flip-flop with an input and an output. The output of each flip-flop becomes an input for the remainder of the circuit, and the input of each flip-flop becomes an output for the rest of the circuit, as illustrated in Figure 3. In this way, any inter-register logic may be tested by modifying the register outputs as needed and assessing the register inputs in addition to examining the ordinary inputs and outputs. This method creates an ephemeral circuit; however, it is just employed to create patterns before being discarded.

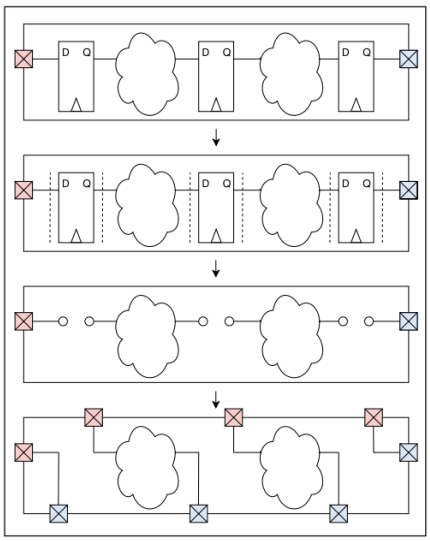

Fig. 3. Converting a sequential circuit to a combinational circuit by “Cutting”

ATPG for stuck-at faults is carried out using PGen. According to the stuck-at fault model, nodes become trapped at logic 0 or logic 1 as a result of manufacturing flaws. PGen employs fault simulation in conjunction with pseudorandom ATPG. This is a more straightforward alternative to algorithmic techniques like PODEM and D algorithms. Algorithmic approaches need to be "path sensitised," which makes it difficult for them to handle netlists that have been mapped using any random standard cell library. TVs are produced pseudo-randomly with PGen and then simulated. To compare a GM to a model whose fault sites are gradually injected using Verilog force statements, PGen creates a testbench for each produced TV. Each model's output is compared with the other, and any trouble location that can be identified using said TV is submitted back to Fault to be tagged as covered. When the intended coverage or the maximum number of TVs is attained, PGen ceases to generate TVs. The final coverage is then exported to a file in the popular and flexible JSON format. However, if only pseudo-random test vectors are used, fault coverage could be degraded for circuits with random resistant fault [6].

Scan chain insertion is performed by using Pyverilog to add an abstract multiplexer specification to each flip-flop input port. The tool is called a chain that transforms the flip-flop cells in a netlist into scan cells. Figure 4 shows the diagram of the scan cell. Yosys is then used to map this description to a multiplexer (MUX) cell from the common cell library or, if available, a scannable flip-flop cell. Additionally, the chain adds boundary scan cells for the inputs and outputs of the netlist before sewing together all the scan cells into a single scan chain. In addition to the testing control signals, the chained netlist contains additional ports for the serial-in and serial-out of the built scan chain. Additionally, the chain provides a choice to automatically produce a testbench to confirm the integrity of the scan chain.

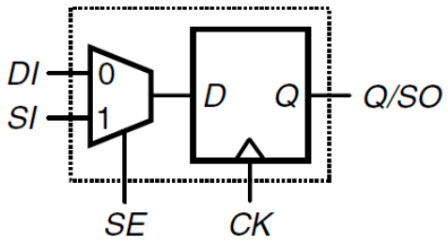

Fig. 4. Scan Cell

Figure 5 shows the architecture of the Joint Test Action Group (JTAG). Using Pyverilog, Tap provides the JTAG interface to a chained netlist. A special instruction is designed to choose the internal scan chain to do on-chip testing. The TAP controller's functionality is checked by automatically creating a testbench that shifts TVs through the scan chain, applies them to the on-chip logic by deactivating shift for one clock cycle, shifts out the recorded output, and compares the results to the fault-free response [1].

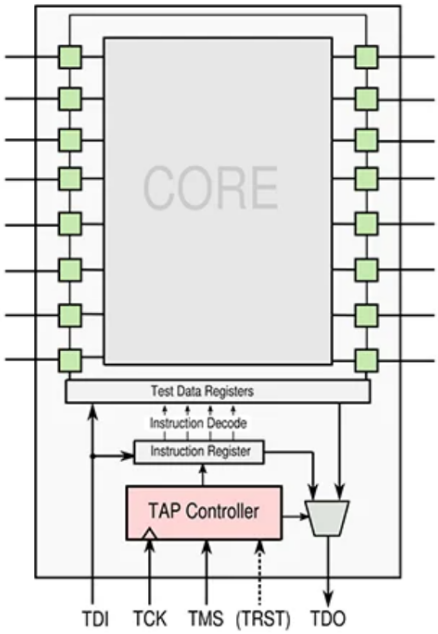

Fig. 5. JTAG Architecture Schematic

3. Result

The case study presented several key findings regarding the implementation of the SHA1 design using OpenLANE and the SkyWater 130-nanometer process design kit. The static time analysis showed that the design meets the meet of timing requirements since there were no hold or setup violations. Although the design was deemed satisfactory, there was still potential for further optimization.

Before the implementation of the SHA-1 design in the floorplan to GDS, a scan chain was inserted in the design using Fault. The scan chain included boundary scan cells that were added to the inputs and outputs of the netlist. These scan cells were then connected to form a single scan chain. The modified netlist now included additional ports for the test data in and test data out of the constructed scan chain.

The design rule check (DRC) examined the layout against the sky130 PDK design rules and found no violations. Sky130 is the first open-source process design kit created by SkyWater together with Google [19]. This minimized the possibility of manufacturing defects by ensuring that the design was implemented to the specific dimensions, spacing, alignment, and connection of the design. Similar to this, all net, device, pin and property in the design all match uniquely during the layout vs schematic (LVS) check, which demonstrated that the physical layout properly mirrored the intended schematic circuit.

The area of the design showed that the SHA1 chip occupied an area of 178460 μm^2 with 48% utilization, which means that core standard cells covered 48% of the area, while the remaining 52% was to wire routing, fill insertion for increased manufacturability, yield enhancement, and antenna insertion.

The SHA1 chip used a total of 3.37×10^{-2} watts of power. The main source was internal power, which was followed by switching power (dynamic power) resulting from transitions in circuit activity. Leakage power, the power dissipated when transistors are in an off-state, was the least significant in terms of consumption.

These findings collectively demonstrate the successful implementation of the SHA1 design using OpenLANE and the SkyWater 130-nanometer process design kit. Static timing analysis, DRC, and LVS

checks showed that the design was compliant, accurate, and reliable. The area, power, and timing analysis assessments provide valuable insights for optimization and future developments in SHA-1 implementations using open-source EDA tools, OpenLANE.

Two Scenarios are compared. In the first case, increasing the maximum fanout to solve fanout violations led to capacitance and slew violations. Adjusting the PL_RESIZER_MAX_SLEW_MARGIN and GLB_RESIZER_MAX_SLEW_MARGIN parameters helped address these violations during the placement and routing phase of the physical design.

In the second case, a limitation on the maximum fanout value was set to 15 to avoid power consumption and delay issues. Buffer insertion was required to solve the fanout violations, but the "insert_buffer" command in the OpenROAD tool failed to execute. The "repair_design" command with the subcommand "-max_wire_length 100" was used as an alternative to insert buffers and limit the wire length.

Table 1 shows the results obtained from 2 cases. Case 1 shows a better result on timing and area. This is due to case 2 having more buffers inserted in the design leading to an increase in time and as the buffer increases, the area will be increased. Case 2 shows better results in power, this is because the wire length for case 2 is shorter than case 1 as there is a limitation on the wire length in the design.

Table 1
Comparison of two Scenario on Timing, Power and Area

Case	Power (mW)	Timing (ns)	Area (µm)
Case1	33.7	11.87	178460
Case2	27.0	14.62	225895

4. Conclusion

In conclusion, the secure HASH algorithm 1 (SHA-1) implementation using OpenLane and the SkyWater 130-nanometer process design kit has been successful. The timing analysis, layout vs schematic (LVS) check, and design rule check (DRC) results show that the design complies with time constraints and ensures efficient and reliable SHA-1 algorithm execution. By confirming that the design is accurate and complies with design constraints and requirements through the LVS and DRC tests, potential manufacturing faults are reduced. The use of OpenLane and the SkyWater pdk to implement SHA-1 illustrates the efficiency and dependability of the design method and tools. Overall, this successful case study emphasizes the abilities of OpenLane to implement RTL to GDS using the 130-nanometer process design kit and emphasises its commitment to developing reliable and secure systems using open-source EDA tools like OpenLane.

Acknowledgement

I would like to thank my supervisor Assoc Prof Ir Dr Ooi Chia Yee, as well as Madam Nordinah Bt Ismail and Mr Chye Chuan Ning for their guidance throughout this project.

References

[1] Abdelatty, Manar, Mohamed Gaber, and Mohamed Shalan. "Fault: Open-source EDA's missing DFT toolchain." *IEEE Design & Test* 38, no. 2 (2021): 45-52. https://doi.org/10.1109/MDAT.2021.3051850

[2] Hesham, Sarah, Mohamed Shalan, M. Watheq El-Kharashi, and Mohamed Dessouky. "Digital ASIC Implementation of RISC-V: OpenLane and Commercial Approaches in Comparison." In *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 498-502. IEEE, 2021. https://doi.org/10.1109/MWSCAS47672.2021.9531753

[3] Tozzi, C. "Open Source Hardware: What It Means and Why It Matters." (2016).

[4] Chatterjee, Baibhab, and Shreyas Sen. "OpenSerDes: An Open Source Process-Portable All-Digital Serial Link." *arXiv preprint arXiv:2105.13256* (2021).
[5] Ajayi, Tutu, Vidya A. Chhabria, Mateus Fogaça, Soheil Hashemi, Abdelrahman Hosny, Andrew B. Kahng, Minsoo Kim et al. "Toward an open-source digital flow: First learnings from the openroad project." In *Proceedings of the 56th Annual Design Automation Conference 2019*, pp. 1-4. 2019. https://doi.org/10.1145/3316781.3326334
[6] Wang, Laung-Terng, Cheng-Wen Wu, and Xiaoqing Wen. *VLSI test principles and architectures: design for testability*. Elsevier, 2006. https://doi.org/10.1016/B978-012370597-6/50006-8
[7] Edwards, R. Timothy, Mohamed Shalan, and Mohamed Kassem. "Real silicon using open-source EDA." *IEEE Design & Test* 38, no. 2 (2021): 38-44. https://doi.org/10.1109/MDAT.2021.3050000
[8] Veigas, Suresh Steven, and R. Elumalai. "Open source VLSI design automation." In *AIP Conference Proceedings*, vol. 2901, no. 1. AIP Publishing, 2023. https://doi.org/10.1063/5.0183307
[9] Abdelatty, Manar, Mohamed Gaber, and Mohamed Shalan. "Fault: Open-source EDA's missing DFT toolchain." *IEEE Design & Test* 38, no. 2 (2021): 45-52. https://doi.org/10.1109/MDAT.2021.3051850
[10] Rajaram, Anand, and David Z. Pan. "Robust chip-level clock tree synthesis." *IEEE transactions on computer-aided design of integrated circuits and systems* 30, no. 6 (2011): 877-890. https://doi.org/10.1109/TCAD.2011.2106852
[11] Reda, Sherief, Leon Stok, and Pierre-Emmanuel Gaillardon. "Guest Editors' Introduction: The Resurgence of Open-Source EDA Technology." *IEEE Design & Test* 38, no. 2 (2021): 5-7. https://doi.org/10.1109/MDAT.2020.3038851
[12] Shalan, Mohamed, and Tim Edwards. "Building OpenLANE: a 130nm openroad-based tapeout-proven flow." In *Proceedings of the 39th International Conference on Computer-Aided Design*, pp. 1-6. 2020. https://doi.org/10.1145/3400302.3415735
[13] Zezin, Denis. "Modern Open Source IC Design tools for Electronics Engineer Education." In *2022 VI International Conference on Information Technologies in Engineering Education (Inforino)*, pp. 1-4. IEEE, 2022. https://doi.org/10.1109/Inforino53888.2022.9782983
[14] Takamaeda-Yamazaki, Shinya. "Pyverilog: A python-based hardware design processing toolkit for verilog hdl." In *Applied Reconfigurable Computing: 11th International Symposium, ARC 2015, Bochum, Germany, April 13-17, 2015, Proceedings 11*, pp. 451-460. Springer International Publishing, 2015. https://doi.org/10.1007/978-3-319-16214-0_42
[15] Ansell, Tim, and Mehdi Saligane. "The missing pieces of open design enablement: A recent history of Google efforts." In *Proceedings of the 39th International Conference on Computer-Aided Design*, pp. 1-8. 2020. https://doi.org/10.1145/3400302.3415736
[16] Kahng, Andrew B. "Looking into the mirror of open source." In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1-8. IEEE, 2019. https://doi.org/10.1109/ICCAD45719.2019.8942131
[17] Rodriguez-Ferrandez, Ivan, Leonidas Kosmidis, Maris Tali, and David Steenari. "Space Shuttle: A Test Vehicle for the Reliability of the SkyWater 130nm PDK for Future Space Processors." In *2023 IEEE 29th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pp. 1-3. IEEE, 2023. https://doi.org/10.1109/IOLTS59296.2023.10224899
[18] Zezin, Denis. "Modern Open Source IC Design tools for Electronics Engineer Education." In *2022 VI International Conference on Information Technologies in Engineering Education (Inforino)*, pp. 1-4. IEEE, 2022. https://doi.org/10.1109/Inforino53888.2022.9782983
[19] Herman, Krzysztof, Mauricio Montanares, and Jorge Marin. "Design and Implementation of Integrated Circuits Using Open Source Tools and SKY130 Free PDK." In *2023 30th International Conference on Mixed Design of Integrated Circuits and System (MIXDES)*, pp. 105-110. IEEE, 2023. https://doi.org/10.23919/MIXDES58562.2023.10203216
[20] Chupilko, Mikhail, Alexander Kamkin, and Sergey Smolov. "Survey of Open-source Flows for Digital Hardware Design." In *2021 Ivannikov Memorial Workshop (IVMEM)*, pp. 11-16. IEEE, 2021. https://doi.org/10.1109/IVMEM53963.2021.00008