iEDA: An Open-source infrastructure of EDA

(Invited Paper)

Xingquan Li[1,8,3*], Zengrong Huang[1], Simin Tao[1], Zhipeng Huang[1], Chunan Zhuang[1], Hao Wang[2,3], Yifan Li[1], Yihang Qiu[11,3], Guojie Luo[5], Huawei Li[2,11,1], Haihua Shen[11,1], Mingyu Chen[2,11], Dongbo Bu[11,1], Wenxing Zhu[4,1], Ye Cai[6,1], Xiaoming Xiong[9], Ying Jiang[7,1], Yi Heng[7,1], Peng Zhang[1], Bei Yu[10,✉], Biwei Xie[2,1,✉], Yungang Bao[2,11,3,1,✉]

[1]Peng Cheng Laboratory, [2]Institute of Computing Technology, Chinese Academy of Sciences, [3]Beijing Institute of Open Source Chip, [4]Fuzhou University, [5]Peking University, [6]Shenzhen University, [7]Sun Yat-sen University, [8]Minnan Normal University, [9]Guangdong University of Technology, [10]The Chinese University of Hong Kong, [11]University of Chinese Academy of Sciences
Email: *lixq01@pcl.ac.cn, ✉byu@cse.cuhk.edu.hk, ✉xiebiwei@ict.ac.cn, ✉baoyg@ict.ac.cn

***Abstract*—By leveraging the power of open-source software, the EDA tool offers a cost-effective and flexible solution for designers, researchers, and hobbyists alike. Open-source EDA promotes collaboration, innovation, and knowledge sharing within the EDA community. It emphasizes the role of the toolchain in accelerating the development of electronic systems, reducing design costs, and improving design quality. This paper presents an open-source EDA project, iEDA, aiming to build a basic infrastructure for EDA technology evolution and closing the industrial-academic gap in the EDA area. As the foundation for developing EDA tools and researching EDA algorithms and technologies, iEDA is mainly composed of file system, database, manager, operator and interface. To demonstrate the effectiveness of iEDA, we implement and tape out four chips of different scales (from 700k to 500M gates) on different process nodes (110nm and 28nm) with iEDA. iEDA is publicly available on the project home page https://github.com/OSCC-Project/iEDA.**

I. INTRODUCTION

As Moore's Law gradually reaches its limits, innovation is the only driving force that can further enhance integration. To achieve better performance, power, area, and cost (PPAC), electronic design automation (EDA) plays a crucial role in the development of electronic systems, encompassing a wide range of research and development activities. The current landscape of EDA research and development is characterized by its complexity. First, the entire chip design chain is too long and is broken down into many different steps, each of which is interdependent and jointly decides on design goals and requirements. This separation design leads to a large gap between early steps and later steps on optimization and evaluation. Second, EDA is a multidisciplinary research direction, involving at least electronic engineering, integrated circuits, software engineering, computer science, mathematics, and physics. Third, EDA is constantly evolving with Moore's Law and chip design requirements, and new research and design challenges will continue to emerge. Fourth, the research on EDA in English is becoming more systematic and practical, and the academic community has lagged far behind the industry in both aspects. It is increasingly difficult for academia to propose and study systematic and authentic scientific research problems [1].

To effectively address these challenges and unlock the full potential of EDA, there is a growing need for an EDA research and training platform, which can be accessed by anybody without distinction. One of the best solutions is building an open-source EDA platform that can foster valuable research and training of talents in the field. Since 1999, the number of ongoing open-source EDA tools has steadily increased [2], as shown in Fig. 1. Especially since 2018, the number of open-source tools has significantly increased due to the implementation of DARPA's electronic rejuvenation plan [3]. These open The rapid growth of open-source EDA tools has created a pressing need for an open-source EDA infrastructure to support developing EDA tools and researching EDA algorithms and technologies. To construct open-source EDA foundations, many contributors to have made efforts in various aspects such as parsers, databases, EDA tools, chip design flows, and process design kits (PDKs).

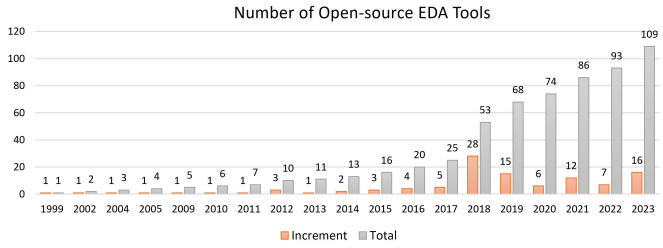


Fig. 1: The number of ongoing open-source EDA tools.

Qflow [4] is one of the earliest open-source EDA toolchains that originated from a personal project. It's also a backend tool of the online chip design platform efabless. VSDFlow [5] originated from a teaching project and is an open-source EDA toolchain based on Qflow. It's slightly easier to use than Qflow, but there's no breakthrough improvement. DATC-RDF [6] was initiated by IEEE CEDA, aiming to provide a more objective and comprehensive evaluation of EDA technology solutions in the academic field and reduce the code development cost of comparing EDA solutions. One notable attempt in this direction is the OpenROAD project [7], which has gained credibility and recognition. OpenROAD is an open-source EDA toolchain and chip design flow whose objective is to design a chip within 24 hours without human intervention. This project connects existing open-source EDA tools to form an EDA toolchain which can be used to support chip agile design. This project

showcases the potential of open-source EDA in automating design processes. OpenLane [8] is an automated flow that converts RTL to GDSII flow. It utilizes various components, including OpenROAD, Yosys, Magic, Netgen, CVC, SPEF-Extractor, KLayout, and custom scripts for design exploration and optimization. SiliconCompiler [9] is a modular and open-source build system that automates the translation process from hardware design source code to silicon. The project is built upon a standardized Schema that supports flexible combinations of design, tools, and PDKs. The design philosophy behind the schema is to enable complex possibilities while keeping simplicity intact. In addition, Ophidian [10] and Rsyn [11] are two open-source flows that focus on physical design and physical synthesis.

The OpenAccess [12] database from Cadence is the world's most widely used, open reference database for IC design. It was developed to create authentic interoperability between EDA companies and semiconductor designers and manufacturers. The OpenAccess database is however not open-source but requires a license. OpenDB [13] is a design database to support tools for physical chip design. It was originally developed by Athena Design Systems. Nefelus, Inc. acquired the rights to the code and open-sourced it in 2019. OpenEDI [14] is another open-source database for EDA design. EPFL presents a collection of open-source libraries for the development of logic synthesis applications [15], the libraries are composed of Alice, Kitty, Lorina, and Mockturtle.

NanGate donated a free 45 nm open-source digital library through Si2 to promote interoperability and independent testing of standard cell-based software products [16]. Toward advanced technology process, a 7nm predictive PDK, called the ASAP7, developed in collaboration with ARM Ltd. [17] for academic use. The PDK is realistic but is not tied to any specific foundry. The SkyWater open-source PDK is a collaboration between Google and SkyWater Technology Foundry to provide a fully open-source PDK and related resources, which can be used to create manufacturable designs at SkyWater's facility [18].

For the EDA foundation, in addition to facilitating the connection between EDA developers, chip designers, and semiconductor manufacturers to support chip design work, a more important task is to support researching EDA methods and algorithms, designing EDA structures and solutions, and developing tools and AI models. Motivated by the importance of building an open-source infrastructure of EDA, our project, iEDA, aims to attract diverse academic disciplines to contribute to EDA research and development (R&D). By fostering interdisciplinary collaboration, iEDA seeks to facilitate the development of more valuable EDA design methodologies. We recognize the significance of bridging the gap between industry and academia, as it enables the seamless transfer of technology and promotes the practical application of research findings in real-world scenarios. Through the availability of iEDA as an open-source project, we aim to ensure its continuous evolution and improvement, closing the gap between industry and academia in the EDA area.

By promoting collaboration, innovation, and knowledge sharing within the EDA community, iEDA aims to accelerate the development of electronic systems, reduce design costs, and improve design quality. This paper presents an overview of the iEDA project, highlighting its features, components, and potential applications. The infrastructure of iEDA encompasses parsers for file format compatibility, a powerful database for efficient data management, an operator for comprehensive design evaluation, analysis, and report, and multiple interfaces for user interaction. These components collectively contribute to the functionality, versatility, and usability of iEDA, enabling designers and researchers to effectively analyze, optimize, and refine their EDA tool and chip design. The main applications of iEDA include supporting research on EDA algorithms and tool development, chip design, evaluation of EDA objects (such as designs, tools, algorithms, and flows), talent cultivation in EDA and chip fields, as well as participating in academic competitions related to chip, EDA, software, performance, and AI.

II. iEDA Project

The iEDA project is introduced in this section, focusing on the objectives, its framework, and key features.

A. iEDA Objective

The primary objective of the iEDA project is to establish a robust open-source infrastructure of EDA that not only attracts more academic disciplines to engage in the EDA field but also facilitates the development of valuable EDA design methodologies. Furthermore, it will enable more disciplines of knowledge and methodologies to enter the EDA field, break through the existing EDA design methodology, and further improve the quality and efficiency of EDA design chips.

B. Pan iEDA Platform

The iEDA platform consists of four key components: a basic infrastructure, a functional module-level data model, a series of valuable solvers (algorithms and AI models) for EDA problems, and multiple digital chip design EDA tools as shown in Fig. 2. The infrastructure includes a database and a series of EDA software development kits. The data model provides pre-design data services for upper-level EDA functions. The AI models and algorithms are the core solvers that determine the quality and performance of the platform. And EDA toolchain is responsible for chip design functions, which include design, analysis, and verification tools. The iEDA platform combined with PDK, enables the flow of chip design from Netlist design to GDS-II layout.

C. iEDA Features

To support the development of the EDA tool and research of algorithms, iEDA is carefully designed. It has several features, including adherence to a project schedule, a uniform infrastructure and tool structure, support for the netlist-GDS-II flow, API support for multiple programming languages, indicators evaluation, data snapshot and recovery capabilities, and the adoption of the Mulan PSL-2.0 license. These aspects collectively contribute to the development of a comprehensive

and accessible open-source EDA infrastructure. One of the notable features of the iEDA project is its commitment to conducting research and development activities according to a well-defined project schedule.

Supporting Netlist to GDS-II toolchain. Support for the netlist-GDS-II flow is another key feature of the iEDA project. This capability enables the translation of electronic designs from the abstract netlist representation to the physical layout in the GDSII format, which is essential for manufacturing integrated circuits. By providing this functionality, iEDA ensures compatibility with industry-standard design flows.

System Integration. The iEDA platform supports the seamless integration of various tools, achieving a smooth design flow and reducing the time and effort designers spend switching between different tools. This ensures a systematic and organized approach to the development of the open-source EDA infrastructure, allowing for efficient progress and timely deliverables.

Uniform Foundation and Interface. The iEDA project also emphasizes the importance of a uniform infrastructure and tool structure. Establishing a consistent framework, it enables seamless integration and interoperability among different components of the EDA toolchains. This uniformity enhances the overall efficiency and effectiveness of the design process.

Multiple Program Languages. To enhance accessibility and usability, the iEDA project offers API support for multiple programming languages, including C++, Rust, TCL, and Python. This allows designers and researchers to interact with the EDA infrastructure using their preferred program language, facilitating customization and integration with existing design methodologies.

Rich Metric Evaluation and Analysis. The iEDA also incorporates indicators evaluation at each stage of the design process. This feature enables designers to assess the performance and quality of the intermediate process of their designs, providing valuable insights for optimization and refinement.

Data Snapshot and Recovery. Additionally, the project includes data snapshot and recovery capabilities, ensuring the preservation and retrieval of design data in case of unexpected events or system failures.

Friendly Secondary Development. The iEDA project continuously encapsulates some key EDA modules, algorithms, or processing methodologies into independent modules, and encapsulates the data input, output, and key processes flow of the solver module to facilitate developers to conduct rapid algorithm research, code integration, and verification.

Mulan PSL-2.0 License. An important aspect of the iEDA project is its adoption of the Mulan PSL-2.0 license[1]. The Mulan license provides users with significant freedoms and rights, making it relatively permissive for open-source project development and usage. This permissiveness can help promote the development and adoption of iEDA.

[1] http://license.coscl.org.cn/MulanPSL2

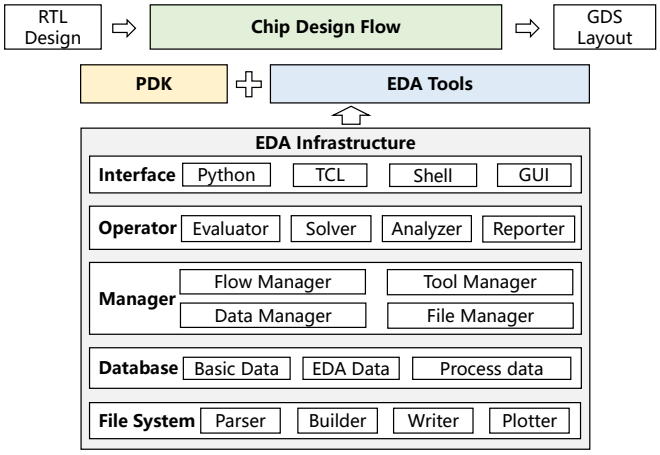


Fig. 2: The iEDA infrastructure.

III. iEDA Infrastructure

The infrastructure of iEDA consists of several key components that collectively enable its functionality and versatility, overall components are shown in Fig. 2. This section provides a detailed exploration of each component.

A. Parser

To read various EDA file formats, iEDA incorporates parsers that support the parsing of different standard files, including Verilog (netlist), SPEF (standard parasitic exchange format), Liberty, SDF (standard delay format), VCD (value change dump), SDC (Synopsys design constraints), LEF/DEF (library exchange dormat/design exchange format), ITF (interconnect technology format) and GDSII (graphic data system II). These parsers allow users to import and analyze design data in different formats, ensuring compatibility with industry-standard file formats.

To achieve better performance in data building and reduce the risk of memory leakage, we have implemented parsers in the Rust programming language. Rust follows the design principle of safety, concurrency, practicality. Most existing open-source EDA parsers are built using Flex and Bison, which are traditional C++ lexical and grammar generators. In contrast, we have chosen to use the elegant Rust parser called pest. Pest is written in Rust and utilizes the recently popular standard parsing expression grammar which makes it easy to express complex language syntax. Additionally, pest separates the grammar from the parser processing code, which improves readability and maintainability of the parser code. Rust excels in concurrency safety and performance. We can leverage Rust's asynchronous programming capabilities to execute the parser tasks, reducing multithread overhead and maximizing CPU utilization.

B. Database

iEDA includes a robust database system capable of storing and managing various types of design data. This includes design data, layout data, timing data, design rule data, and circuit network data, as shown in Fig. 3. The database serves as a

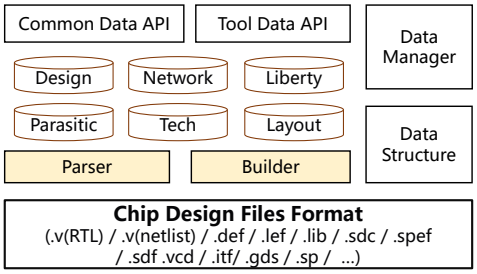
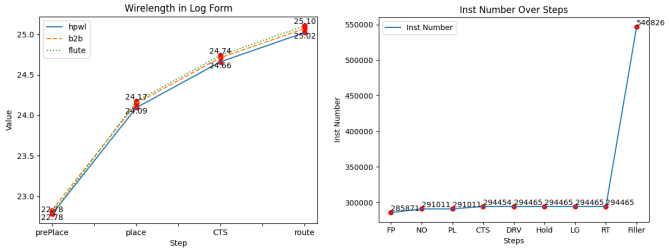


Fig. 3: The iEDA database.



(a) (b)

Fig. 4: Vertical comparison of metrics "wirelength" and "#instances".

central repository for storing and retrieving critical information throughout the design process, facilitating efficient data management and analysis. In addition, besides storing data from files, iEDA database also has some basic data structures (includes graph and geometric) and various intermediate data generated by EDA tool and algorithm.

C. Operator

To better supporting EDA tool development, we designed an operator module in iEDA. The operator module composes of evaluator, reporter, solver and analyzer. The iEDA evaluator and analyzer play a crucial role in assessing the performance and quality of designs, and it has following characteristics: firstly, there are different metrics reflecting the quality of chip design from various perspectives in each chip design stage; secondly, there are various different calculation models for each metric in each stage; and lastly, each metric is compared vertically during different stages. As shown in Fig. 4, total wirelength of design and the number of instances are assessed in several stage by different calculation models.

The iEDA reporter offers a comprehensive set of evaluation capabilities, including timing analysis, power estimation, wirelength estimation, area calculation, resistance and capacitance analysis, slew and skew analysis, design rule violation checking, density analysis, congestion analysis, routability analysis and so on. And to achieve convenient interaction with designer, these metrics are organized as reports by the iEDA reporter. In Fig. 5, we list a part of metric reports. These evaluations and reports provide designers with valuable insights into the behavior and characteristics of their designs, enabling them to optimize and refine their designs for improved performance and manufacturability.

For EDA tool, the most important component that affects their quality is the inbuilt algorithm solvers. While effective

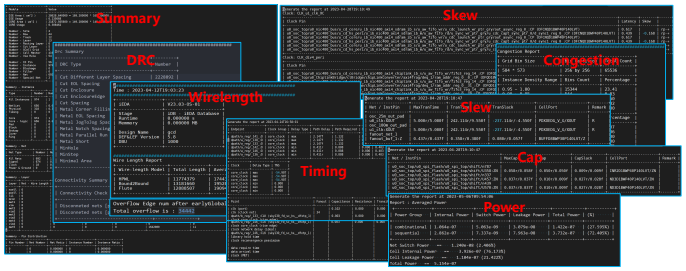


Fig. 5: Some Metrics in iEDA.

algorithm is designed by high-level researchers from various fields. iEDA abstracts and encapsulates a serious of solver from EDA tools, by doing so, personnel with various backgrounds and programming abilities can easily utilize the tools and develop their own tools and algorithms, truly reducing the threshold problem caused by the high interdisciplinary requirements of EDA. The analyzer provides comprehensive circuit analysis capabilities, including timing, power analysis and design rule check. With these tools, designers can efficiently verify their designs and optimize their performance.

D. Interface

To convenient using iEDA for algorithm researcher and tool designer, we provide multiple interface ways to cater to different user preferences and workflows. These interfaces include Python APIs, TCL APIs, a graphical user interface (GUI), and a shell interface. The Python and TCL APIs enable users to automate and customize design flows with iEDA using their preferred language. The GUI offers a visual and user-friendly interface, while the shell interface provides advanced users with flexible and convenient command-line access to iEDA's functionalities.

iEDA provides a unified, user-friendly interface for chip designers and EDA developers, with features like data management, parameter configuration, flow control, and performance evaluation. Its consistent naming scheme eases tool usage and interface invocation, while its unified parameter setup simplifies chip design and EDA parameter adjustments. Additionally, iEDA provides a visualization tool for chip layout, layer control, DRC visualization, clock tree visualization, and other data analysis functions.

IV. iEDA Applications

This section provides a detailed overview of the various applications and use cases of iEDA.

A. Developing EDA Tool

The primary purpose of iEDA is to support EDA tool development. To verify the application value of iEDA on EDA tool development, we designed a physical design toolchain (named iPD). iPD is composed of nine EDA tools, including floorplan, macro placement, placement, clock tree synthesis, routing, static timing analysis, timing optimization, power analysis, and design rule check. iPD provides advanced algorithms and methodologies for efficient and optimized physical variables, enabling designers to achieve better performance,

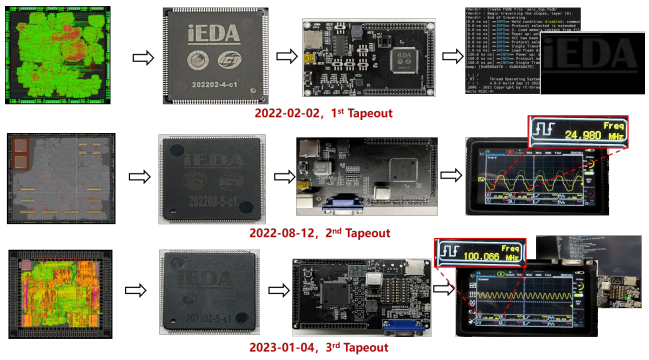


Fig. 6: Three times chip design and tape-out by using iEDA.

TABLE I: Three chips' parameters designed by using iEDA.

chip parameters	1^{st} **tapeout**	2^{nd} **tapeout**	3^{rd} **tapeout**
process node	110nm	110nm	28nm
chip area	3mm×3.5mm	4.8mm×4.8mm	1.5mm×1.5mm
#gates	>700 thousand	>1.5 million	>1.5 million
dynamic power	48mW	343mW	317mW
leakage power	7 mW	21 mW	29 mW
clock frequency	25MHz	25MHz	200MHz

power consumption, and manufacturability in their designs. The EDA tools in iPD wrap data from iEDA and use some iEDA operators and managers to auxiliary develop algorithms and flows, and access their interface into the API of iEDA.

B. Designing Chip

To demonstrate the effectiveness of iEDA, it has been utilized to design four chips at 110nm and 28nm process nodes. These chips vary in scale and complexity, ranging from smaller designs with 700k gates to larger designs with 500M gates. By successfully implementing chip layouts, iEDA demonstrates its capability to handle designs of different sizes and process technologies, providing designers with a flexible and scalable platform for chip implementation. Fig. 6 shows three tapeout chips and PCB results, and the corresponding chip parameters are listed in TABLE I.

C. Evaluating Chip Design, EDA Tool, Algorithm and Flow

iEDA can be used to support evaluating chip design, EDA tool, algorithm and flow.

Design. iEDA enables comprehensive evaluation of chip design quality, it collates information from multiple sources and delivers insights into design parameters that range from performance and power consumption to layout geometry and parasitic effects. By integrating all relevant information and tools into a single platform, iEDA removes the need for designers to toggle between multiple software environments, saving valuable time and resources. The platform's user-friendly interface makes it easy for designers to quickly access the necessary data and insights to support their decision-making process. TABLE II show a metric comparison between two designs "aes" and "aes_core" on Skywater 130 process node designed by iEDA.

EDA Tool. iEDA facilitates the evaluation and comparison of different EDA methods, algorithms, and tools. For example,

TABLE II: Metrics comparison between designs "aes" and "aes_core".

part metrics	**aes**	**aes_core**
PDK	Sky Water 130HS	Sky Water 130HS
instance area (um^2)	408034.76	271050.98
#IO pins	76	520
#instances	45854	42044
#nets	30634	28536
core area (um^2)	1352765.88	1230601.766
total wirelength (um)	2695657	2809505
#vias	280870	271884
setup slack (ns)	14.7	14.73
hold slacke (ns)	0.22	0.4
suggest frequency (MHz)	188.68	189.75

TABLE III: Metrics comparison between placement legalization methods "abacus" and "tetris".

part metrics	**abacus**	**tetris**
global placement HPWL	10127910	10127910
legalization HPWL	10426323	13168231
detail placement HPWL	9901517	10928985
detail placement STWL	10637190	11674987
maximum STWL	431085	415325
total movement	795829	8705103
maximum movement	5684	218214
average congestion	0.8215	0.8134
total overflow	49	49
peak bin density	1	1
legalization runtime (s)	0.0667	0.0064

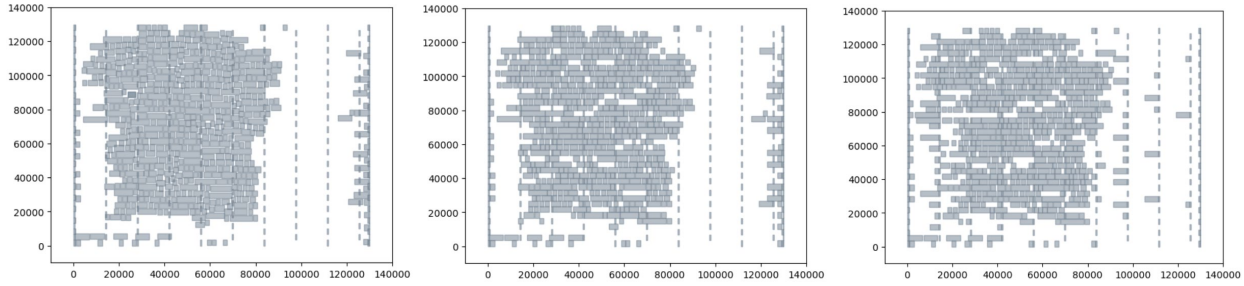
designers can compare the performance of iEDA's placement algorithm (iPL) with other popular placement tools like DreamPlace. This comparison helps designers make informed decisions about the most suitable algorithms for their specific design requirements.

Algorithm. iEDA also enables the comparison of different methods/algorithms for EDA tools. By evaluating and comparing these methods, designers can identify the most effective approach for the specific chip design. For example, at the placement legalization step, there are many effective algorithms, in which Abacus and Tetris are the two typical works. We make a comparison between the above two methods by iEDA, the corresponding results and metrics are shown in Fig. 7 and TABLE III.

Flow. iEDA was created to address the need for a standardized approach to evaluating the quality of backend design flows. It provides a repeatable and reliable method for assessing the quality and efficiency of design flows. This enables designers, engineers, and chip development teams to identify strengths and weaknesses in their current processes and make informed decisions about process improvements and tool selection. A comparison on the same chip "gcd" on Skywater 130 process with two different flows and parameters is presented in TABLE IV.

D. Training Talent

To support the EDA talent development effort, iEDA can be used to train talents in the EDA field. By providing an open-source platform with comprehensive tools and capabilities, iEDA enables researchers, students, and hobbyists to gain



(*a*) GP result (*b*) Abacus (*c*) Tetris

Fig. 7: Legalization results by "Abacus" and "Tetris".

TABLE IV: Metrics comparison between different flows.

part metrics	flow1	flow2
detail routing HPWL (um)	10879081	11025675
final wirelength (um)	11471595	12071042
setup slack (ns)	-0.492	-0.484
hold slacke (ns)	0.426	0.427
suggest frequency (MHz)	345.804	346.784
power (mW)	0.956	0.966
#DRC	755	643

hands-on experience in EDA methodologies and techniques. This support for talent development helps foster innovation and knowledge sharing within the EDA community.

E. Supporting EDA Contest

iEDA platform has good expansibility and customizability, and we can customize our own electronic design problems based on the platform. By using the iEDA platform to create competitive problems, we can help students improve their electronic design skills and ability to innovate. It has been used as a contest platform by the 2023 Integrated Circuit EDA Elite Challenge. iEDA also provides support for participants to participate in EDA contests. EDA contests are competitive events where participants showcase their skills and expertise in solving specific design challenges. By leveraging the capabilities of iEDA, participants greatly save the time of building prototype and focus on developing and optimizing their algorithm or tool for these contests, enhancing their chances of success and recognition.

V. CONCLUSIONS

iEDA aims to build an EDA infrastructure by providing a cost-effective and flexible solution through an open-source software development kit. This paper highlights the key components of iEDA, including the file system, database, manager, operator, and interface. The comprehensive infrastructure of iEDA fosters collaboration, innovation, and knowledge sharing within the EDA community. The successful implementation and tape-out of chips using iEDA demonstrate its effectiveness and compatibility with various process technologies. In addition to supporting chip design, iEDA also serves as a research and development platform for EDA designers. Moreover, iEDA can be utilized as an evaluator, training platform, project backbone, and more.

ACKNOWLEDGMENT

This work is supported in part by the Major Key Project of PCL (No. PCL2023A03), the National Key R&D Program of China (No. 2022YFB4500403), the National Natural Science Foundation of China (No. 62090024), the National Natural Science Foundation of China (No. 62090021), the National Natural Science Foundation of China (No. 62174033).

REFERENCES

[1] M. S. Lundstrom and M. A. Alam, "Moore's law: The journey ahead," Science, vol. 378, no. 6621, pp. 722–723, 2022.

[2] clin99, "Awesome-EDA," 2019. [Online]. Available: https://github.com/clin99/awesome-eda#2019.

[3] DARPA, "Electronics Resurgence Initiative," 2018. [Online]. Available: https://www.darpa.mil/news-events/2018-11-01a.

[4] R. T. Edwards, "Qflow," 2019. [Online]. Available: http://opencircuitdesign.com/qflow/.

[5] K. Ghosh and A. Ghosh, "Technology mediated tutorial on risc-v cpu core implementation and sign-off using revolutionary eda management system (ems)-vsdflow," in Proc. of CSTIC, 2018, pp. 1–3.

[6] J. Chen, I. H.-R. Jiang, J. Jung, A. B. Kahng, S. Kim, V. N. Kravets, Y.-L. Li, R. Varadarajan, and M. Woo, "DATC RDF-2021: Design flow and beyond," in Proc. of ICCAD. IEEE, 2021, pp. 1–6.

[7] T. Ajayi, D. Blaauw et al., "OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain," in Proc. of GMACTC, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:210937106

[8] M. Shalan and T. Edwards, "Building OpenLANE: A 130nm OpenROAD-based Tapeout-Proven Flow (Invited)," in Proc. of ICCAD, 2020, pp. 1–6.

[9] A. Olofsson, W. Ransohoff, and N. Moroze, "A Distributed Approach to Silicon Compilation (Invited)," in Proc. of DAC, 2022, p. 1343–1346.

[10] T. Fontana, R. Netto, V. Livramento, C. Guth, S. Almeida, L. Pilla, and J. L. Güntzel, "How Game Engines Can Inspire EDA Tools Development: A Use Case for an Open-Source Physical Design Library," in Proc. of ISPD. ACM, 2017, p. 25–31.

[11] G. Flach, M. Fogaça, J. Monteiro, M. Johann, and R. Reis, "Rsyn: An extensible physical synthesis framework," in Proc. of ISPD. ACM, 2017, p. 33–40.

[12] D. Cottrell and A. Graham, "What is OpenAccess?" 2003. [Online]. Available: http://www.si2.org/openaccess.

[13] Athena Design Systems. Nefelus, Inc, "OpenDB," 2019. [Online]. Available: https://github.com/The-OpenROAD-Project/OpenDB.

[14] NiCEDA, "OpenEDI," 2020. [Online]. Available: https://gitee.com/amongo/open-edi.

[15] M. Soeken, H. Riener, W. Haaswijk et al., "The EPFL logic synthesis libraries," Jun. 2022, arXiv:1805.05121v3.

[16] NanGate, Inc, "NanGate 45nm open cell library," 2008. [Online]. Available: https://si2.org/openeda.si2.org/projects/nangatelib.

[17] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, "ASAP7: A 7-nm finFET predictive process design kit," Microelectron. J., vol. 53, pp. 105–115, 2016.

[18] T. Edwards, "Introduction to the SkyWater PDK: The New Age of Open Source Silicon," 2021. [Online]. Available: https://github.com/gdsfactory/skywater130.