

# Interpretable machine learning in VLSI physical design

**Ben Sun**

Dept of Engineering, University College London, London, UK

Bensun20011109@foxmail.com

**Abstract.** Today's popularisation of portable devices largely depends on the progress in integrated circuits. Modern Very Large Scale Integration technology (VLSI) allows billions of transistors to be packed into the same chip. In the past years, digital design in VLSI has been developed compared to analogue design. The traditional method is hard to model the performance change in analogue or mixed-signal components caused by physical design. In the early 2000s, rapid advances in machine learning and computing power made analogue design automation possible. Despite their outstanding performance, the transparency issue has become significant. This paper introduces the history of VLSI physical design, which includes placement and routing in the early stages. The change that machine learning (ML) has made is mentioned in the third section. Analysis of the potential problem has been proposed, followed by a brief category of some well-known work in interpretable Machine Learning, which could be the primary direction for VLSI automation to be further popularised in the future.

**Keywords:** VLSI, ROUTING, LAYOUT, explainable Artificial Intelligence (XAI).

## 1. Introduction

With the rapid increase in demand for portability and computing power in modern society, integrated circuit design (IC) has become increasingly complicated. People want more functions in the circuit with the performance drop as little as possible. Well-designed schematic, layout and routing are three critical factors during the design process. However, those steps are time-consuming and expensive. Automating these steps can cut costs while increasing efficiency, freeing engineers from repeating work. Despite the progress in EDA tools and algorithms in the digital circuit that has experienced significant improvements in the early 21st century, most of them are targeted at digital design, focusing more on the speed and cost side. An analog IC is another case; it requires reliability, stability, and low distortion while obtaining a high S/N ratio. Electronic design automation (EDA) tools for analogue ICs are significantly less than those for digital design. The rising need for analogue and mixed-signal circuits is a significant challenge in LSI(Large Scale Integration)/VLSI(Very Large Scale Integration) design automation. In recent years, the rapidly developing machine learning (ML) and neuro network (NN) have provided a new perspective on dealing with these problems. The ML algorithms can take multiple inputs and deal with correlated parameters, which is hard to achieve by traditional methods. They are able to find hidden relations between elements and are helpful when the rules and instructions are vague. This paper focuses on the layout and routing algorithms of the VLSI design. Despite the advantages of ML, the transparency problem has been a more significant concern these days. These models are often treated as black box models because of the multiple hidden layers they contain, which makes the user unable to know the

reason behind a specific behaviour. This problem is important since the analogue IC design should be reliable. Also, this reliability should be accepted by the users of the EDA system because they simply won't rely on a tool they do not believe in. A new region of study on this interpretability issue has been widespread, and multiple algorithms aiming at the interpretability problem have been implemented in other areas. The potential and future development of this algorithm in VLSI and computer-aided design (CAD) are also discussed in the last part.

## **2. The history of LSI design in the early stage**

Despite replacing vacuum tubes with transistors significantly reducing the circuit size, circuits at that time still require designers to connect multiple monofunctional components manually. This situation has not improved until Jack Kilby and Robert Noyce found that multiple electrical components can be produced on the same semiconductor base. [1] The circuit built in this way is not only smaller and cheaper but also performs better because the physically shorter distance between functional blocks reduces the power loss. After that, an explosion in the number of transistors pushed the industry through SSI and MSI eras during the 1960s entering the LSI stage. LSI automation approaches in that years usually focused on a particular aspect of the production process [2], or on solving one specific design problem, for example, routing [3]. Routing was the most commonly discussed problem before the importance of layout was introduced. Multiple methods have been brought up. R. Prim proposed a router [4] that solves the routing problem as solving a maze; this kind of router connects sets of selected terminals, keeping the traces reasonably short while intersecting less. Maze router and its improved versions have great use in robot route plans, printed circuit board (PCB) global routing and other applications. The routing order is a problem since the existing tracks stop other lines from passing. Another significant issue is that maze routers are strictly restricted by the mesh formed; if the user requires better performance, the mesh has to be made finer, leading to an explosion in calculations and taking a long time to route. Later, David Hightower introduced an improved algorithm based on the continuous plane [5]. This kind of Hightower router was commonly used and integrated into other algorithms in PCB and LSI autorouting. However, this router may not always find the best solution and will cause congestion on a global level. Another router built by S. Kimura et al solves this problem by treating the routing problem at both global and local levels [6]. It also allows more delicate wires based on a gridless router [7].

L. Corrigan [8] mentioned a placement algorithm which has reached a minimum wire length and improved the wire density. This is done by improving the placement of components in the package. This method was used in the Hughes CAD system and outputs layout data for other systems. H. Shiraishi et al. combined the placement and routing for master slice LSIs [9]. This algorithm has four phases, with an initial and an improving stage for each part. Tests were done on this algorithm, with 2000 gates being placed and routed in three minutes. C. -L. Ding et al. proposed a technique to improve placement by implementing a clustering algorithm before placing [10]. In the late 2000s, M. Aoyagi et al. mentioned a placement and routing algorithm [11] that met the needs of increasing complexity and higher frequency. A layout of a 1GHz chip was outputted when testing. This system also considers the power balance; the balancing on each power phase will be as close to each other as possible to make the power supply circuit stable while the heat is evenly distributed through the chip.

## **3. Modern VLSI design with machine learning**

With the ongoing demand for more powerful ICs, LSI has progressed to the point where a single chip can contain more than 100k gates, known as VLSI. Although the boundary between these two is not that clear, and most LSI algorithms could be expanded to VLSI design with some modifications, increasing gate density and requirements for the mixed signal has been challenging. The introduction of machine learning and neuro networks has brought new ideas to the industry. A. B. Kahng [12] points out that the high cost of new nodes and the lack of ability to get enough benefit from new nodes in the design process are two of the leading crises. Learning-based models, which can improve from additional experiences, are needed for future development. It is necessary to integrate ML into every stage of the IC design.

Back in 1985, algorithms based on simulated annealing were used to optimise placing and routing. The TimberWolf [13] package was tested on ILLIAC computer boards and reduced the circuit size by up to 62% while saving 10% total wire length compared to other algorithms simultaneously. This algorithm is slow but has been the infrastructure for multiple improved methods later. Recently, Compute Unified Device Architecture (CUDA) has been implemented successfully on TimberWolf [14], speeding up the algorithm by 55 times on average.

K. Zhu et al mentioned a routing method based on machine learning [15]. This router focuses on the analogue routing problem, which is usually considered hard in VLSI design since a minor change in the routing path could impact the performance. At the same time, some decisions made by human experts are a combination of trade-offs based on past experiences and requirements, which sometimes make the design rule ambiguous and conflicting. Since ML is good at learning the hidden design rules from the dataset, the constraints in the GeniusRouter are implicitly generated by ML rather than directly coded by engineers. Some additional rules will also be generated by ML where necessary. The result of the GeniusRouter was simulated in Cancade ADE under TSMC's 40nm process. The simulation result of both the comparator and the operational transconductance amplifier (OTA) is compelling to a hand-crafted design.

H. Chen et al [16] developed an open-source MAGICAL framework [17] that takes circuit network and design rules as inputs and outputs the layout in a standard format, without human intervention. This framework could automatically generate parameters and extract constraints for sensitive devices, which Engineers usually do before the placement and global routing algorithms take place. It separates a high-level requirement into smaller blocks, building and verifying them before integrating them to build the system. The modules in the MAGICAL framework are also separatable, providing flexibility while making it highly expandable and can work with various ML and NN algorithms to improve its performance. WELLGAN [18] is a network which could replace the default well-generating submodule in MAGICAL. It converts well-generation problems to computer vision and learns from experts. The routing problem could be handled by the GeniusRouter mentioned earlier. A 3-D CNN [19] could predict the placement quality and reduce the training set. By taking advantage of ML algorithms in a different design stage, the MAGICAL framework was able to build circuits similar to a human expert despite the requirements of repetitive simulations before minizine the performance drop. It is tested on OTA and CTDSM, and the simulation of its layout is performed close to the schematic. This framework is full of potential and could benefit from the open source community in the future.

#### **4. Problems of machine leaning on CAD applications**

In the early stage of CAD and EDA applications, the reason for a specific decision made was somewhat self-interpretable. Take the classic routing Lee [20] as an example. The code for this algorithm is simple but indicates why the path it found is the shortest. Some simple ML algorithms could also be self-explainable, for example, linear regression, simulated annealing and SVM. However, it is not the case when it comes to NN. NNs tend to have multiple hidden layers, and the users could not know how the iterations were done. Almost all current work regarding ML applications in VLSI design rarely talks about the explainability of the model. Those NNs are treated as a black box and focused on their accuracy. Although the interpretability may be sacrificial in many fields, for example, providing ads on a webpage, It is vital in the industry because a wrong decision may cause resources and time to be wasted on debugging or a redesign. Currently, users are just providing parameters for the NN applications in EDA and CAD and getting a result after some time. Although the experts who were able to test the model may potentially know the implicit reason leading to that decision, the users are getting a result without reasoning. It is mentioned by R. Mina et al [21] that introducing new nodes could lead to new effects that are not significant enough in the older process. The same applies to the layout and routing problem, where the model learnt from datasets of the older process may not apply to the current situation. In this case, it is hard for engineers to trust that the algorithm will always make a good decision in all related applications. Trust issues aside, thinking about an algorithm could explain the reason behind that decision; it would also be helpful for the users to understand how are the parameters they are inputting

will impact the outcome. This could give users more control over the algorithm and make their ideal design easier. All those ideas came together to lead to a new category of ML in recent years, the Interpretable Machine Learning.

### **5. Interpretable machine learning in VLSI design**

The most straightforward way is to make the algorithm explain themselves. This method tends to sacrifice some performance for interpretability because these two characteristics are linearly related, just as L. Breiman [22] has pointed out. One way to keep the performance of the NN is by modifying the black box and inserting some components in different stages. A relational network [23] was used with a CNN to increase its transparency. This model was tested on a graphical dataset called CLEVER and outperformed humans.

Alternative approaches are a kind of method called post-hoc. These methods won't affect the performance of the original Blackbox but make the model more understandable by visualising and reasoning. Some of them disrupt the input and provide explanations according to the change in output. The partial dependence plot (PDP) has been commonly used to explain a black box and other research areas where people want to figure out the factors affecting the final pattern. It shows the marginal effect of changing one or two features on the output. However, implementing them in the CAD region could be problematic; The parameters and characteristics in VLSI design are often related. In contrast, PDP assumes those features to be uncorrelated. It can only have two parts maximum, and the result of the plot is an averaged value, which may ignore the influences of other factors. An ICE graph [24] was invented in the following years to solve the misleading PDP problem when there are interactions of features. As its name (Individual Conditional Expectation) suggests, this method can only plot one feature meaningfully. P. Kumar et al [25] introduced a visualised way based on earlier Feature importance [26] and Feature interaction [27] algorithm. The feature importance algorithm is the random forest algorithm which was able to show how a specific feature contributes to the final result. In contrast, the feature interaction algorithm identifies the elements that are affecting others and the strength of their influence. In the FIFO graph, the importance and the power of interactions between features will be displayed on a single chart with the size of the circle indicating the importance and the width of an edge indicating how strong the interaction of the elements at its terminals is. This visualised approach could provide users with a brief feeling of the main reasons for a specific decision and has the potential to be used in the IC design region.

Some algorithms tried to use a simpler, more explainable model to explain a complex black box. These are surrogate algorithms. The idea is to use the trained black box to generate the predication of a dataset. And use the dataset and predicated set to train the surrogate model, which are simpler algorithms such as decision tree, linear regression and explicit rules and constraints. Depending on the range of interpretation, they can be global [28] or local [29]. Local surrogates, such as LIME [29], could better explain a specific data point. This module was tested on an SVM classifier with 94% accuracy and found the classifier unreliable due to the problematic dataset. Another test was done on the reason for a classifier which classifies the image of an acoustic guitar as electric guitar. Although the classifier made the wrong decision, after seeing the reason behind that, people still trust this classifier more because the explanation has shown that the model is working reasonably. However, this kind of algorithm is only faithful locally. When it comes to the global level, global surrogates express the whole black box better. The STACI [28] model was tested on the reason for a patient being diagnosed with diabetes. The reason was made by STACI and two additional algorithms and expressed in plain text. That result was then forwarded to 55 experts with computer science backgrounds to see their preferences on different aspects such as confidence, generality and number of features.

Despite having those powerful algorithms, few tests have been done on the IC design direction. P. Krishnamurthy et al [30] proposed a method based on an ML framework called Sentences in Feature Subsets. This framework will first extract the feature based on the graphical structure of the circuit. Since the classifier is not interested in all of those features, they are further selected by the RF algorithms, which compute the importance of the features. GA algorithms are used to search in the feature subsets.

A trained sentence classifier generates the best set of sentence combinations using a GA. The unnecessary part is that the combination is being pruned later to keep the best sentences. The human readability is also noticed here; In the last part, once an input triggers one or multiple sentences, an interpretation of the decision could be presented in Boolean expressions. This algorithm obtained successful results on the IC test, pointing out the critical part of the circuit for engineers to focus on.

## 6. Conclusion

In this paper, the brief history of VLSI autorouting and placement has been discussed. From hardcoded rules and constraints to ML, the complexity and performance were increased at the same time, raising questions about the reason for trusting them. Several explainable machine learning methods were introduced to provide a potential solution to this concern.

Due to the relative blankness of the current research direction, this paper fails to find enough applications of Interpretable machine learning on VLSI's placement and routing aspect. However, those XAI algorithms have shown enough benefit to be implemented in the VLSI design. By doing so, engineers will be able to know the logic behind every decision, which is helpful for them to know and trust their tools better. Also, this kind of algorithm can indicate and highlight the important part of the circuit that the engineer may otherwise not pay attention to. In the future, the application of XAI on VLSI design should be valued. At the same time, XAI and other machine learning algorithms should be able to take full advantage of the state-of-the-art hardware, improving their performance while cutting down the design cycle. One thing to mention about the XAI is that while researchers call the algorithm "explainable" or "interpretable", the way and the effectiveness of the answer they provide may vary. Thus, depending on the targeted audience, a human-friendly result may be required instead of a human-readable one. Overall, XAI (eXplainable Artificial Intelligence) is a broad topic that covers multiple subjects, including psychology and cognitive science. Its application to VLSI design could be challenging but full of potential.

## References

- [1] "The Nobel Prize in Physics 2000", NobelPrize.org, 2022. [Online]. Available: <https://www.nobelprize.org/prizes/physics/2000/kilby/facts/>. [Accessed: 30- Jul- 2022].
- [2] P. Cook, W. Donath, G. Lemke and A. Brennemann, "Automatic Artwork Generation for Large Scale Integration", IEEE Journal of Solid-State Circuits, vol. 2, no. 4, pp. 190-196, 1967. Available: <https://ieeexplore.ieee.org/document/1049817>. [Accessed 30 July 2022].
- [3] K. Chen, M. Feuer, K. Khokhani, N. Nan and S. Schmidt, "The chip layout problem: an automatic wiring procedure", Papers on Twenty-five years of electronic design automation - 25 years of DAC, 1988. Available: 10.1145/62882.62897 [Accessed 30 July 2022].
- [4] R. Prim, "Shortest Connection Networks And Some Generalizations", Bell System Technical Journal, vol. 36, no. 6, pp. 1389-1401, 1957. Available: 10.1002/j.1538-7305.1957.tb01515.x [Accessed 30 July 2022].
- [5] D. Hightower, "A solution to line-routing problems on the continuous plane", Proceedings of the 6th annual conference on Design Automation - DAC '69, 1969. Available: 10.1145/800260.809014 [Accessed 30 July 2022].
- [6] S. Kimura, N. Kubo, T. Chiba and I. Nishioka, "An Automatic Routing Scheme for General Cell LSI", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 2, no. 4, pp. 285-292, 1983. Available: 10.1109/tcad.1983.1270046 [Accessed 30 July 2022].
- [7] U. Lauther, "A Data Structure for Gridless Routing," 17th Design Automation Conference, 1980, pp. 603-609, doi: 10.1145/800139.804593 [Accessed 30 July 2022].
- [8] L. Corrigan, "A Placement Capability Based on Partitioning", 16th Design Automation Conference, 1979. Available: 10.1109/dac.1979.1600145 [Accessed 30 July 2022].
- [9] H. Shiraishi and F. Hirose, "Efficient Placement and Routing Techniques for Master Slice LSI," 17th Design Automation Conference, 1980, pp. 458-464, doi: 10.1145/800139.804570 [Accessed 30 July 2022].

- [10] C. -L. Ding, C. -Y. Ho and M. J. Irwin, "A new optimization driven clustering algorithm for large circuits," Proceedings of EURO-DAC 93 and EURO-VHDL 93- European Design Automation Conference, 1993, pp. 28-32, doi: 10.1109/EURDAC.1993.410612 [Accessed 30 July 2022].
- [11] M. Aoyagi, Y. Hamazaki, H. Nakagawa, I. Kurosawa, M. Maezawa and S. Takada, "Chip layout design of a Josephson LSI circuit for examining high-speed operability by using a standard cell automatic placement and routing technique," in IEEE Transactions on Applied Superconductivity, vol. 4, no. 3, pp. 169-176, Sept. 1994, doi: 10.1109/77.317833 [Accessed 30 July 2022].
- [12] A. B. Kahng, "INVITED: Reducing Time and Effort in IC Implementation: A Roadmap of Challenges and Solutions," 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), 2018, pp. 1-6, doi: 10.1109/DAC.2018.8465871 [Accessed 30 July 2022].
- [13] C. Sechen and A. Sangiovanni-Vincentelli, "The TimberWolf placement and routing package", IEEE Journal of Solid-State Circuits, vol. 20, no. 2, pp. 510-522, 1985. Available: 10.1109/jssc.1985.1052337 [Accessed 30 July 2022].
- [14] A. Al-Kawam and H. M. Harmanani, "A Parallel GPU Implementation of the Timber Wolf Placement Algorithm," 2015 12th International Conference on Information Technology - New Generations, 2015, pp. 792-795, doi: 10.1109/ITNG.2015.144 [Accessed 30 July 2022].
- [15] K. Zhu et al., "GeniusRoute: A New Analog Routing Paradigm Using Generative Neural Network Guidance," 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2019, pp. 1-8, doi: 10.1109/ICCAD45719.2019.8942164 [Accessed 30 July 2022].
- [16] H. Chen et al., "MAGICAL: An Open- Source Fully Automated Analog IC Layout System from Netlist to GDSII", IEEE Design & Test, vol. 38, no. 2, pp. 19-26, 2021. Available: 10.1109/mdat.2020.3024153 [Accessed 30 July 2022].
- [17] H. Chen et al., "MAGICAL 1.0: An Open-Source Fully-Automated AMS Layout Synthesis Framework Verified With a 40-nm 1GS/s  $\Delta\Sigma$  ADC," 2021 IEEE Custom Integrated Circuits Conference (CICC), 2021, pp. 1-2, doi: 10.1109/CICC51472.2021.9431521 [Accessed 30 July 2022].
- [18] B. Xu et al., "WellGAN: Generative-Adversarial-Network-Guided Well Generation for Analog/Mixed-Signal Circuit Layout," 2019 56th ACM/IEEE Design Automation Conference (DAC), 2019, pp. 1-6. [Accessed 30 July 2022].
- [19] M. Liu et al., "Towards Decrypting the Art of Analog Layout: Placement Quality Prediction via Transfer Learning," 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020, pp. 496-501, doi: 10.23919/DATE48585.2020.9116330 .[Accessed 30 July 2022].
- [20] C. Lee, "An Algorithm for Path Connections and Its Applications", IEEE Transactions on Electronic Computers, vol. -10, no. 3, pp. 346-365, 1961. Available: 10.1109/tec.1961.5219222 [Accessed 30 July 2022].
- [21] R. Mina, C. Jabbour and G. Sakr, "A Review of Machine Learning Techniques in Analog Integrated Circuit Design Automation", Electronics, vol. 11, no. 3, p. 435, 2022. Available: 10.3390/electronics11030435[Accessed 31 July 2022].
- [22] L. Breiman, "Statistical modeling: The two cultures (with comments and a rejoinder by the author)," Stat. Sci., vol. 16, no. 3, pp. 199–231, 2001
- [23] A. Santoro et al., "A simple neural network module for relational reasoning." Available: <https://arxiv.org/abs/1706.01427>. [Accessed 6 August 2022].
- [24] A. Goldstein, A. Kapelner, J. Bleich and E. Pitkin, "Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation", Journal of Computational and Graphical Statistics, vol. 24, no. 1, pp. 44-65, 2015. Available: 10.1080/10618600.2014.907095 [Accessed 30 July 2022].
- [25] P. Kumar and M. Sharma, "Feature-Importance Feature-Interactions (FIFI) graph: A graph-based Novel Visualization for Interpretable Machine Learning," 2021 International Conference on

- Intelligent Technologies (CONIT), 2021, pp. 1-7, doi: 10.1109/CONIT51480.2021.9498467 [Accessed 30 July 2022].
- [26] L. Breiman, "Random Forests", *Machine Learning*, vol. 45, pp. 5-32, 2001. Available: <https://link.springer.com/article/10.1023/A:1010933404324>. [Accessed 30 July 2022].
- [27] J. Friedman and B. Popescu, "Predictive learning via rule ensembles", *The Annals of Applied Statistics*, vol. 2, no. 3, 2008. Available: 10.1214/07-aos148 [Accessed 30 July 2022].
- [28] N. Radulovic, A. Bifet and F. Suchanek, "Confident Interpretations of Black Box Classifiers," 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9534234 [Accessed 31 July 2022].
- [29] M. Ribeiro, S. Singh and C. Guestrin, ""Why Should I Trust You?"" , Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. Available: 10.1145/2939672.2939778 [Accessed 31 July 2022].
- [30] P. Krishnamurthy, A. B. Chowdhury, B. Tan, F. Khorrami and R. Karri, "Explaining and Interpreting Machine Learning CAD Decisions: An IC Testing Case Study," 2020 ACM/IEEE 2nd Workshop on Machine Learning for CAD (MLCAD), 2020, pp. 129-134, doi: 10.1145/3380446.3430643 [Accessed 31 July 2022].