International Innovative Research Journal of Engineering and Technology
Vol. 8, Issue. 2, Dec 2022, pp. 20~28
ISSN: 2456-1983, DOI: 10.32595/iirjet.org/v8i2.2022.166

Implementations of Artificial Intelligence for Automated Electrical Design in Integrated Circuits

Sangwoo Jeon[1], R.Dhivya[2]
[1]Department of Computer Science and Engineering, Konkuk University, Seoul, South Korea.
[2]Department of Artificial Intelligence and Data Science, PSNA College of Engineering and Technology, Dindigul, Tamil Nadu, India.

Article Info

Article history:

Received Sep 9, 2022
Revised Oct 18, 2022
Accepted Nov 21, 2022

Keywords:

Artificial Intelligence
Integrated circuits
Machine learning
ASIC
Field programmable gate array

ABSTRACT

In this paper, we propose a unique approach that uses information set to automatically create a trained-classifier integrated circuit. A dataset in comma-separated value format is accepted by the framework, which then goes through several processing processes to produce a trained model. Once the model has been created, the framework produces a tree-based ML classifier in Verilog and XML. We display the structure of the created graph using the XML participation, and we express the trained model using the Verilog code as an electronic specification tool. In our framework, a Field Programmable Gate Array (FPGA) design validation flow receives the Verilog code as input. A customized classifier integrated circuit is then constructed, and the Application Specific Integrated Circuit (ASIC) flow implementation is automated. When addressing the implementation of ML classifiers, the suggested framework's originality is in bridging the gap between the training of ML models and their hardware design. The design automation and deployment of bespoke ML classifier chips from raw dataset files provide several issues our approach tackles. In this paper, we address these issues in depth and describe how researchers can use our suggested methodology to create classifier chips with cheap costs and great performance. With an average multiplication accuracy of 80.79% throughout several distinct information sets, our algorithm runs at hundred MHZ.

Corresponding Author:

Sangwoo Jeon,
Department of Computer Science and Engineering,
Konkuk University, Seoul, South Korea.
Email: jswp5580@konkuk.ac.kr

1. INTRODUCTION

ML classification has emerged to be among the most effective approaches for situations involving enormous amounts of processing [1]. Even though ML network algorithms are presently being used to categorize records according to instructional information and to uncover unconscious trends within information sets, these methods' effectiveness must continue to be enhanced. Instantaneous applications like live video streaming picture identification and real-time natural language processing particularly require this enhancement. Using a specially designed ASIC processor to carry out the detection operation is one method of enhancing the capacity of ML classifiers. There are many design and implementation hurdles during the flow construction spanning information to layout files. As far as we are aware, there is an EDA application that can produce an HDL file that represents the physical structure of a trained ML classification without the need for supervision, given an assortment of data in raw format. A unique methodology for automatically creating an experienced algorithm IC from an array of data is presented in this paper.

Surprisingly investigators have used the text classification method to streamline the grouping and identification of design patterns, it is similar to other domains like author identification, sentiment analysis, web page classification, and email filtering [2]. Nonetheless, the gap between the features that are utilized to

organize design patterns and their semantic relationship must be resolved. The method we suggest in this paper makes use of a potent deep learning algorithm called DBN, which learns from the semantic structure of documents expressed as feature vectors. An automated system that uses text categorization to classify and choose software design patterns was the subject of our case research. Two primary research goals were the focus of the case study: to assess the notable improvement in classifier classification decisions employing the suggested approach, and to empirically examine the impact of characteristic sets created using international filter-based techniques for feature selection in addition to the suggested approach Figure 1 show that developers can create a more descriptive feature set by adjusting DBN settings like iteration, amount of hidden layers, and nodes.

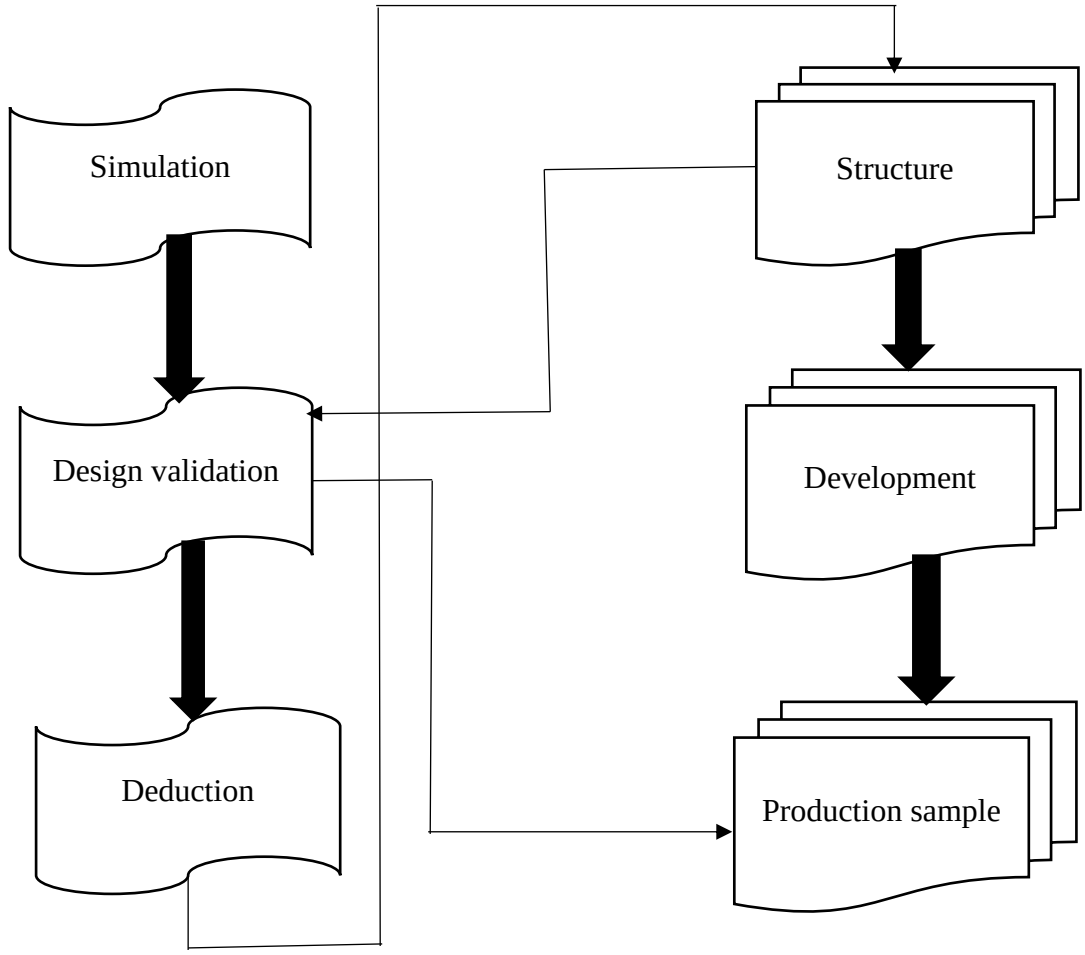

Figure 1. Computerization of electrical design

In analog electronics, a lot of non-idealities originate from both systematic and random mistakes made during circuit implementation. These mistakes stand for a circuit's time-independent reliability issues [3]. Because numerous chemical reactions that occur during the production of integrated circuits are stochastic, including line edge roughness and random dopant fluctuations, unpredictable errors—often referred to as variability—are the outcome. One major factor limiting the precise operation of analog circuits is device discrepancy amongst devices with equivalent designs. Thankfully, the mismatch is rectified after manufacturing though it is random in the sense that it is unforeseen before fabrication and thus random at design time. Consequently, post-fabrication calibration techniques can be used to partially adjust for all of these static, time-independent defects after fabrication. Usually, these make use of the enormous potential of digital circuits, which are essentially widely and inexpensively available in nanoscale CMOS technology. The basic architecture of technologically guided analog designs can be achieved by applying the same compensating principle to additional analog circuit non-idealities.

This format will be used for the remainder of the paper of this study presented in Section 2. In Section 3, the specification of FPGA, and electronic design automation AISC is demonstrated. We go into great depth

about how learning through performance evaluation metrics is used to accomplish a great deal of the EDA using in Section 4. The topic is concluded, and future directions are outlined in Section 5.

2. RELATED WORKS

A number of the most significant areas in the discipline of electrical engineering is the construction of electronic automation. Processor architecture has become increasingly standardized and complex over the last few decades [4]. The automatic conversion of C-based requirements to languages that describe hardware is made possible by high-level synthesis. Because HLS enables the architect to use general terms for a hardware system, it greatly simplifies hardware design. Completing the analysis in HLS can frequently require a lengthy time when dealing with large-scale systems. A key issue in ML is the conversion of the psychological-level specification to the transistor-level description through logic synthesis.

A large and growing number of applications in EDA, as well as in many other areas of science and technology, frequently use Boolean Satisfiability as the underlying model [5]. With the development of new and effective SAT algorithms in recent years, substantially larger problem instances can now be solved. As with BDD packages, which have been around for more than ten years, SAT "packages" are now anticipated to affect EDA applications. Introducing the Binary satisfiability problem to the EDA professional is the goal of this tutorial article. In particular, we emphasize how several EDA issues across a variety of fields, including test design creation, are formulated using SAT concepts.

A number of the most significant areas in the discipline of electrical engineering is the construction of electronic automation. Processor architecture has become increasingly standardized and complex over the last few decades [4]. The automatic conversion of C-based requirements to languages that describe hardware is made possible by high-level synthesis. Because HLS enables the architect to use general terms for a hardware system, it greatly simplifies hardware design. Completing the analysis in HLS can frequently require a lengthy time when dealing with large-scale systems. A key issue in ML is the conversion of the psychological-level specification to the transistor-level description through logic synthesis.

The intricacy of chip design is always rising. Assuring scalability, stability, and appropriate time-to-market, EDA has successfully navigated the difficult, very large-scale integration process. EDA methods, however, require a lot of time and resources, and they frequently don't provide the best results. ML has been integrated into many design flow stages, including placement and routing, to mitigate these issues [7]. Despite the fact numerous EDA objects simply appear as graphs, several solutions use Euclidean data and machine learning approaches. With graph topologies for circuits, intermediary RTLs, and netlists, the popular Graph Neural Networks offer a direct solution to EDA difficulties. In this study, we give a thorough analysis of the literature that links Graph Neural Networks to the EDA cycle for chip design.

ML holds great promise for improving the operational performance of circuits and systems while drastically cutting the amount of design time and errors [8]. There has been some intriguing research in this area; for instance, the lack of a sizable annotated data set makes applying machine learning to CAD/VLSI design more difficult than simple picture or object identification. Since first-generation CNNs and DNNs are ill-suited for use due to the unavailability of this researchers are being encouraged to investigate alternative types of artificial neural networks. According to this paper, Bayesian learning is an exciting technique that works well with CAD algorithms.

In this era of the architecture on an element, IC designers face three specific challenges. The design's computer hardware and software parts don't have a common language, which is essentially a weakness [9]. As the number of logic gates per chip surpasses one million, it becomes increasingly difficult to verify the validity of a design while designing it. Finally, getting a design to match timing requirements without a lot of prototype modifications is becoming less of a goal as both gate counts and chip frequencies are increasing. As is common in the EDA community, start-up businesses run by a small group of people with huge ideas and just a small amount of seed money are attacking these challenges.

Digital system and electrical circuit mechanization has a long history of significant invention. The IC industry has been greatly impacted by the computing company operations, and vice versa [10]. The technology, methods, and methodologies utilized in EDA tools are reviewed in this study along with their business implications. Specifically, we concentrate on four areas—physical design, simulation/verification, creation, and test—that have become crucial in establishing the design processes as time went on. The potential future follows being lightly examined. More software programmability or some form of field configurability, such as FPGAs, will be the direction of design evolution. The types of tool sets required to facilitate design in this setting are discussed.

As integrated circuits are designed using nanoscale CMOS technologies, reliability is increasingly becoming a key consideration [11]. Deterioration of semiconductor circuitry over time is caused by issues with degeneration processes like NBTI or soft breakdown, in addition to increased outside noise like EMI and crosstalk. Variability only intensifies these issues. As a result, designers require new methods and resources to assist them deal with these issues of variability and dependability. As an initial steps towards the mechanized

design of ensured reliable analog circuits, this tutorial article provides a brief overview of tools for design for the effective study and diagnosis of reliability issues in digital circuits.

The present research introduces a Process Design Kit for Natural Thin-Film Transistor architecture that is printed using gravure [12]. The transistor model created in the PDK makes it possible to forecast complicated biological circuits' static, dynamic, and noise performance with accuracy. To increase the PDK's adaptability for alterations in the building process, the created EDA tools take advantage of an adaptive technique. The efficiency of the PDK is illustrated through the structure and tests of a Command Sensitive Amplifier. A dependable design process for intricate circuits based on organic printing should be made possible by the availability of a flexible and precise design tool Kit.

3. METHODS AND MATERIALS

3.1 The Specification of the FPGA Modular Design

Programmable gate arrays are a member of a huge family of modular logic parts. An FPGA is a matrix of programmable logic blocks connected by a fully able-to-be interconnection network [13]. To ensure that the component satisfies the necessary application criteria, the memory cells regulate the logic blocks and links. Numerous technologies can be altered. Only reprogrammable ones are noteworthy among them since they offer the same degree of flexibility as microprocessors. But even if Flash-based technology has an order of magnitude fewer configuration cycles, it is still useful for some demanding specialized applications, such as the aerospace and science industries. Since Flash technology maintains the FPGA's configuration even when its power is turned off, the chip is prepared for operation as soon as it is powered on.

The most modern FPGAs are made with a copper process that runs at 65 nm. Considering timing device frequency over fifty MHz, the number of them can exceed a billion comparable gates per chip. It is crucial to remember, though, that as technology grows, this type of knowledge only holds temporarily. Figure 2 show that the FPGA general design is made up of an array of CLBs, with the bigger ones having an increasing amount of rows and columns. There is a ring of programmable input/output blocks surrounding this matrix core, with a maximum quantity of a thousand user IOBs. Ultimately, these resources communicate with one another via a network of configurable connections. Additionally, the recent addition of an analog-digital converter to the fusion component is quite intriguing for applications in control. However, this trend can be viewed as an addition to this initial matrix rather than a replacement for the previous generic design.

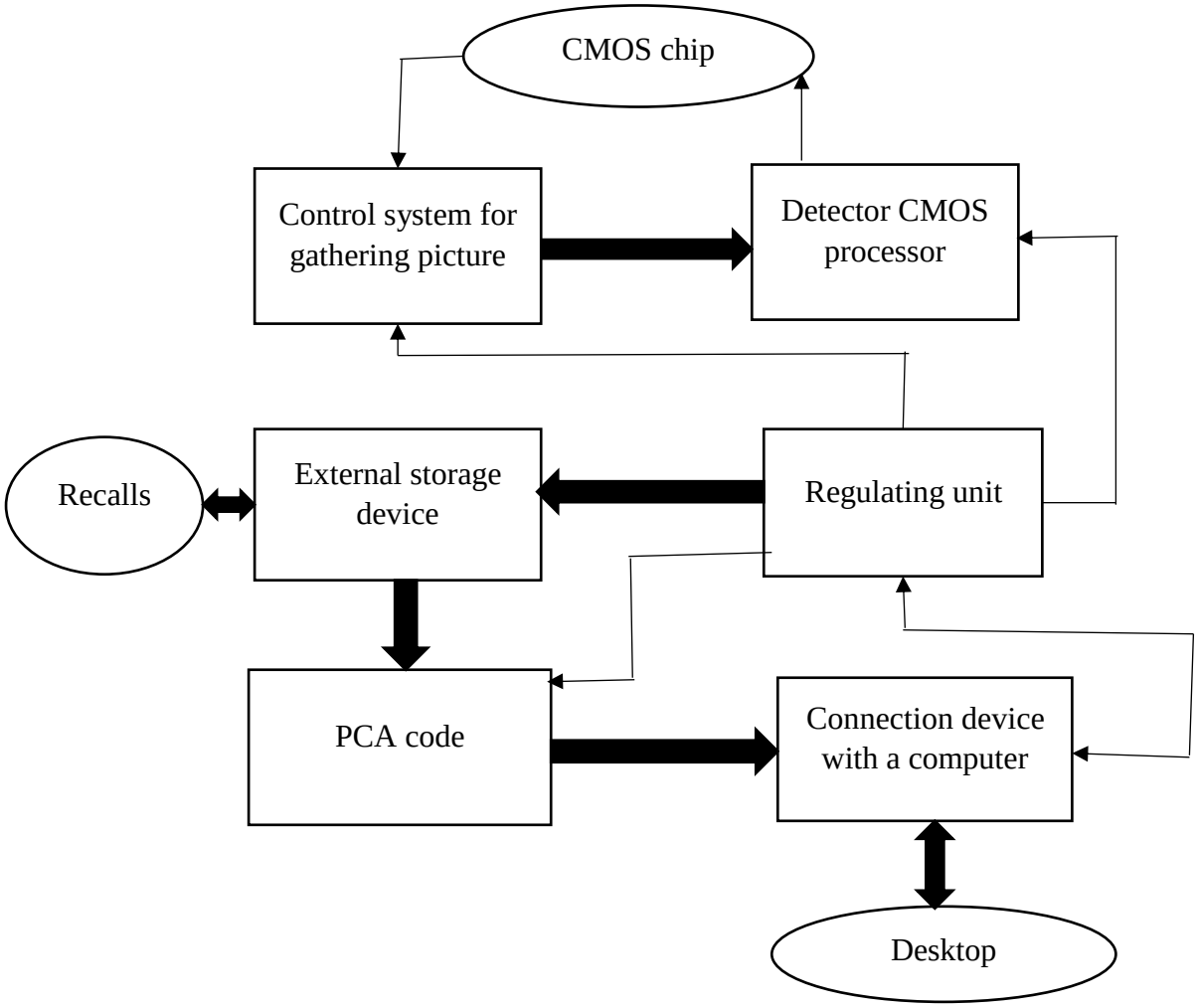

Figure 2. The general design of an FPGA

3.2 System structure for Processor design

The applications, variants of packaging, and base components needed to develop and build a design are included in the ASIC design kit. In addition to being optimized for performance and final product cost, the ASIC design kit must be built to reduce design time to market (TAT) because it might serve as the foundation for many different designs. Several architectural choices are chosen to achieve this when creating the architecture of an ASIC kit. In the end, these choices will influence the capabilities of a design. Three of these options are examined in this section: the electronic library content needed for design; the supported image/package combinations; and the technology design point, which includes transistor selection and metal-level layering.

3.2.1Modern technology characteristics

A system-on-a-chip design requires a high degree of operational insertion, which entails integrating silicon elements that were formerly optimized in different technological offerings [14]. Each of the discrete functional blocks was tailored to meet its particular functional needs, which were generally limited. These characteristics need to be merged economically in the Semiconductor world, which could lead to technical trade-offs and difficulties. Numerous device options are available from today's top semiconductor techniques. One of the most crucial architectural choices made during the development of the ASIC collection is the selection of the transistor or transistors to be used in it. This decision will have an impact on the amount of energy, cost, and performance characteristics that can be achieved on chip designs that make use of the library.

The ASIC designer must decide which semiconductor option to offer before knowing the final application because development costs do not allow for the generation of IP optimized for every design point. The architects aim for a technological advancement by taking into account enhanced performance from the prior product while choosing a transistor selection for the ASIC library. The group also takes into account the power needs of current or potential clients, as well as the financial effects. In this case, the standard n-FET is chosen as the basis for a product due to it has exceptional leakage characteristics but falls short of the performance

improvement goal, whereas the device has great performance but an excessive current loss for low-leakage programs.

In addition to offering more wiring connections per unit area, thin levels also reduce die size and expense. For the distribution of energy, high-speed clocking, and signal routing that needs to go over long distances, thicker metal is preferable. The metal levels utilized are fixed once the solid core is routed and cannot be altered without a major redesign, which is another limitation of the metal stack. To avoid using incompatible metal stacks for distinct hard cores, the ASIC design kit must indicate which metal stack options can be utilized for hard-core networking. Sometimes, this metal stack compromise leads to a rigid core that is larger than is required. This trade-off is widely acknowledged as the cost necessary to enable the rapid design of Microprocessor designs with the ASIC design system and current IP.

3.2.2 Development Approach for ASIC

Individuals begin the ASIC design with the front-end phase, where the synthesis tool receives inputs from the Verilog code, the design constraints, and the standard library for UMC technology. The synthesis process was carried out using Synopsys Design Compiler. The DDC file, netlist file, SDF file, and SDC file are the outputs of the synthesis tool. The design is saved in Synopsys' internal database format by the DDC file. The gate-level netlist for the place and route procedure is included in the netlist file. The Verilog simulation's timing limitations are included in the SDF file. Timing restrictions for the place as well as the route process are included in the SDC file. Next, we run the post-synthesis model to confirm the design's timing and functioning.

Although we synchronized the flow of all design parts using the same temporal frequency, we were able to avoid performing clock domain crossover checks on the resulting design. Every tree node is implemented as a combinational logic element thanks to the Verilog code's structure. To synchronize the writing processing of the decision and the process of reading from the dataset, the data entering stage and results stage are constructed as registers. The back-end phase is the second part of the physical layout. To complete this phase, we utilize the Cadence Encounter tool. This step connects all of the nets via routing and translates the conceptual netlist to the standard cells that utilize UMC 180 nm technology.

3.3 The research selected a solution for simulated annealing

The majority of optimization-based design methods for analog automation of designs that have been presented in recent decades use SA, one of the most widely used stochastic techniques. SA is a point-to-point stochastic search technique that is analogous to solid annealing. It is expected that in a reduction issue, the current answer u is shifted to a relative v, which can be approved based on the likelihood, without losing generality.

$$M\big(d(x), d(y)\big) = \frac{1}{1+e^{[(d(x)-d(y))(v^{*}d(x)]}} \quad (1)$$

The respective performances of the present and neighboring solutions are denoted by f(x) and f(y), respectively. Depending on the corresponding performance, the likelihood may be more flat or more sudden; this is regulated by the value of v, which is often planned as a logarithmic declining relationship to time.

$$V(v) = V_{max}.\, e^{-r}\left(\frac{v}{k}\right) \quad (2)$$

Where V max is the starting temperature, R is the rate at which it decreases, and k is the factor of scale for the repetition counter v. When T is high, the probability p becomes flat at about 50%, meaning that the solution space is explored. When V is low, the probability M can be roughly determined by the exploit.

$$M(\, d(x)\,, d(y)\,) = \{1\ when\ d(x) \geq d(y) \quad (3)$$

It can accept unfavorable solutions probabilistically in this approach. Through the use of stochastically controlled hill-climbing, it is able to steer clear of local minima while optimizing.

4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The structure presented in this study is a changeable Logic in this program, with each component intended to be programmable. The self-testing of any type of electronics with numerous configurations is thus guaranteed by this architecture. The addition of multipliers to the architectures of PRPG and MSR structures allowed for adaptability by allowing for freedom in the location of tap additions, regulating the feedback polynomials and designs produced. Figure 3 show that altering the frequency affects the power usage of each learned classifier. For power analysis, we ran many simulations with clock frequencies ranging from 1 MHz to

100 MHz. Due to the design's small amount of sequential pieces and having an assemblage of multiplexers and additional logic gates that builds the tree-hierarchical trained model

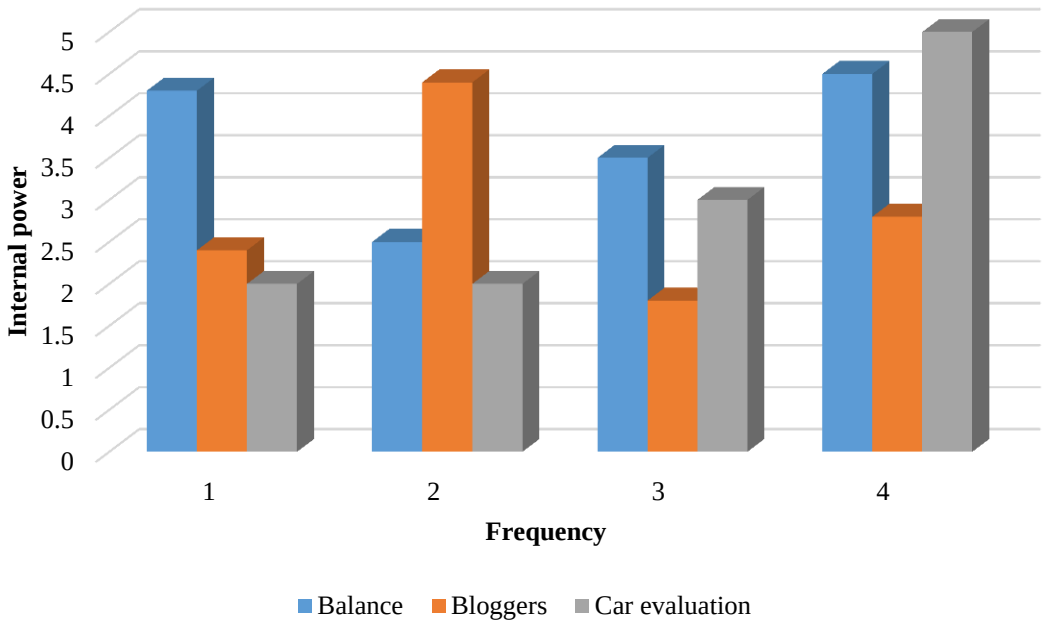

Figure 3. The internal capacity of each data collection

As shown by the research of this work, modular structure outperforms other types of programmable LFSR or MISR in terms of execution speed and gate utilization. ASIC synthesis gate utilization by different long-term sequential kinds is demonstrated via an examination of power consumption across all The Low - Frequency types. Figure 4 show that hybrid forms had speeds comparable to normal ones and used fewer gates because of their effective designs. As a future development, sleep transistors or power gating strategies could be used to further lower the structure's power consumption.

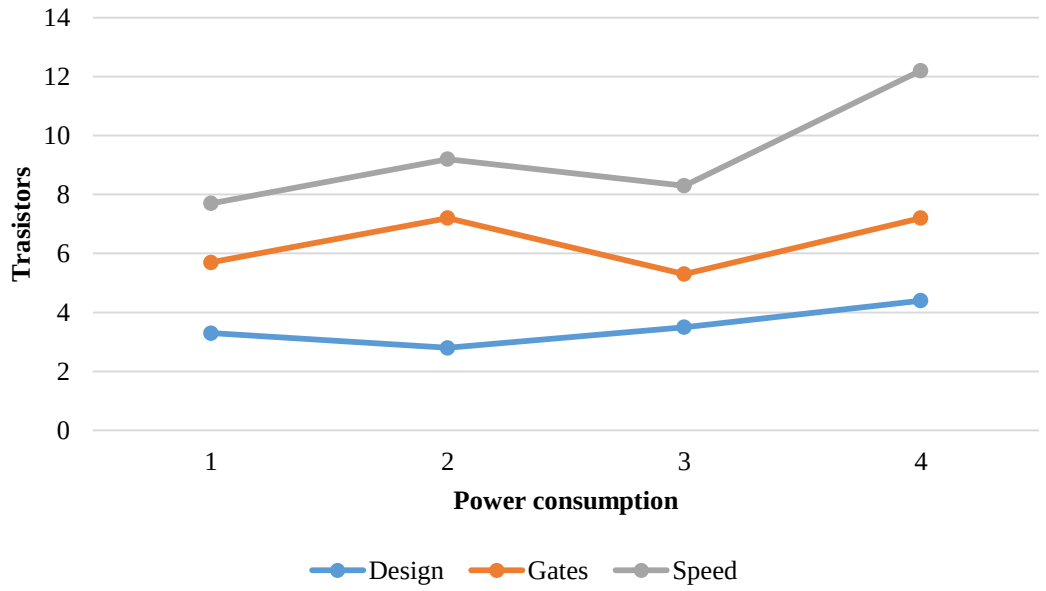

Figure 4. Representation of the FPGA configurable

The number of switches looks from the beginning tables and input/output pins indicates the FPGA resources used. The area and the delay are both affected by the tree size. Which is determined by the sum of the number of branches, the forest depth, and the amount of binary characteristics. The balance scale dataset had the most extensive in both areas as latency.

According to this work's research, modular structures perform better in terms of execution speed and gate utilization than other programmable LFSR or MISR kinds. An analysis of power consumption across all Low-Frequency types demonstrates the use of ASIC synthesis gates by various long-term sequential kinds. Because of their efficient designs, hybrid forms required fewer gates and had speeds comparable to those of conventional forms. To further reduce the structure's power usage, sleep transistors or power gating techniques may be employed in the future. Researchers conducted a number of power analysis simulations, ranging from 1 MHz to 100 MHz clock frequency. Due to a restricted number of sequential elements in the architecture and the use of multiplexers and other logic gates to build the trained model in a tree hierarchy, we found that the power consumption increases with frequency in an almost linear fashion.

Table 1. Findings of the Mega Scale VCU108 FPGA system.

Description of the data set	**Max delay in the information line**	**Volume of bund**	**Detail**	**Operations**
Measure	2.57	7.8	302	10.2
Internet bloggers	0.66	6	37	12
Automobile	1.33	12	130	153
Perspectives	0.45	12	13	22
Elements	1.85	11	132	1.07

After completing the FPGA prototyping, we moved on to the ASIC implementation. We started the ASIC implementation process by synthesizing the design. A summary of the report on synthesis. We exploited the 180 nm technologies of UMC. The table lists the relevant NAND gate count, ports, nets, cells, and total area for each classifier implementation. To find the similar NAND gate count, we synthesized and built an analog gate module. We then used the area of each of our classifiers to get the same circuit count, which came out to be 9.37 um2. The largest area was found in the power-hungry Balance-Scale dataset. We discovered that power usage increases with area.

5. CONCLUSION

To create an optimized classification IC from a raw dataset, we suggested an innovative structure that automates the process. The learned ML classifiers' Verilog designs were produced by our approach, which also fulfilled the requirement for tailoring. Actual natural language processing and image recognition in live video streaming are two examples of ASIC designs for real-time applications. Using five distinct case studies, we verified the suggested framework's functionality. The mean precision of the classifiers produced by testing trained models. The precision of the tenfold cross-validation was 80.79% on typical. Total dynamic power was a value of 0. Small amounts on average. We used the ultra-scale FPGA VCU108 Evaluation Kit for the FPGA design and UMC 180 nm technology for the ASIC implementation. In future development, we plan to address the shortcomings of our current design. We will first enhance the multiplication technique to make sure that the structure is not restricted to just notional features. We'll look at the effects of reducing the split condition's heating to avoid any possible overfitting.

REFERENCES

[1] Sayed, R., Azmi, H., Nassar, A. M., & Shawkey, H. (2020). Design automation and implementation of machine learning classifier chips. IEEE Access, 8, 192155-192164.

[2] Hussain, S., Keung, J., Khan, A. A., Ahmad, A., Cuomo, S., Piccialli, F., ...& Akhunzada, A. (2018). Implications of deep learning for the automation of design patterns organization. Journal of Parallel and Distributed Computing, 117, 256-266.

[3] Gielen, G., Maricau, E., & De Wit, P. (2010, November). Design automation towards reliable analog integrated circuits. In 2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (pp. 248-251). IEEE.

[4] Huang, G., Hu, J., He, Y., Liu, J., Ma, M., Shen, Z., ... & Wang, Y. (2021). Machine learning for electronic design automation: A survey. ACM Transactions on Design Automation of Electronic Systems (TODAES), 26(5), 1-46.

[5] Marques-Silva, J. P., & Sakallah, K. A. (2000, June). Boolean satisfiability in electronic design automation. In Proceedings of the 37th Annual Design Automation Conference (pp. 675-680).

[6] Gubbi, K. I., Beheshti-Shirazi, S. A., Sheaves, T., Salehi, S., PD, S. M., Rafatirad, S., ... & Homayoun, H. (2022, June). Survey of machine learning for electronic design automation. In Proceedings of the Great Lakes Symposium on VLSI 2022 (pp. 513-518).

[7] Lopera, D. S., Servadei, L., Kiprit, G. N., Hazra, S., Wille, R., & Ecker, W. (2021, August). A survey of graph neural networks for electronic design automation. In 2021 ACM/IEEE 3rd Workshop on Machine Learning for CAD (MLCAD) (pp. 1-6). IEEE.

[8] Beerel, P. A., & Pedram, M. (2018, May). Opportunities for machine learning in electronic design automation. In 2018 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-5). IEEE.
[9] Geppert, L. (2000). Electronic design automation [Technology 2000 analysis and forecast]. IEEE Spectrum, 37(1), 70-74.
[10] MacMillan, D., Camposano, R., Hill, D., & Williams, T. W. (2000). An industrial view of electronic design automation. IEEE transactions on computer-aided design of integrated circuits and systems, 19(12), 1428-1448.
[11] Gielen, G., Maricau, E., & De Wit, P. (2010, November). Design automation towards reliable analog integrated circuits. In 2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (pp. 248-251). IEEE.
[12] Fattori, M., Fijn, I. A., Hu, L., Cantatore, E., Torricelli, F., & Charbonneau, M. (2019, March). Circuit design and design automation for printed electronics. In 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE) (pp. 42-47). IEEE.
[13] Monmasson, E., & Cirstea, M. N. (2007). FPGA design methodology for industrial control systems—A review. IEEE transactions on industrial electronics, 54(4), 1824-1842.
[14] Bednar, T. R., Buffet, P. H., Darden, R. J., Gould, S. W., & Zuchowski, P. S. (2002). Issues and strategies for the physical design of system-on-a-chip ASICs. IBM Journal of Research and Development, 46(6), 661-674.