Journal of Theoretical and Applied Information Technology
31st December 2024. Vol.102. No. 24
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DEEP LEARNING-BASED CONGESTION CONTROL IN VLSI FOR PLACEMENT AND ROUTING

SHAIK ASIF HUSSAIN[1]*, SHAIK KARIMULLAH[2], FAHIMUDDIN SHAIK[3], SYED JAVEED BASHA[4]

[1]Technology Transfer Officer & Asst. Professor, Head in charge of Centre for Research & Consultancy, Middle East College, Muscat, Sultanate of Oman.
[2]Associate Professor, Department of Electronics and Communication Engineering Annamacharya University, Rajampet, Andhra Pradesh, India.
[3] Associate Professor, Department of Electronics and Communication Engineering Annamacharya University, Rajampet, Andhra Pradesh, India.
[4]Design Verification Engineer, SmartSoc Solutions India Pvt.Ltd, Banglore, India.
E-mail:[1]shussain@mec.edu.om, [2]munnu483@gmail.com, [3]fahimaits@gmail.com, [4]syed.javeed@smartsocs.com

ABSTRACT

This work presents the application of a recurrent neural network based on the deep learning technique to evaluate design constraints for routing and placement flow, with the assistance of existed floorplan approach, to estimate the value of congestion. Effective area utilization is essential in very large-scale integration (VLSI) circuit design, wherein congestion reduction is also associated with improving floorplan, placement, and routing. This improvement significantly helps a circuit's compact design and performance. Congestion is a fundamental key issue in VLI for estimating the density of area underlies routing among various computational blocks. We used the deep learning technique for the simulation of the standard benchmark circuit's planned area for better placement. Prior approaches estimated the values of congestion for standard architectures, whereas this work considered the floorplan and placement outputs of standard MCNCBM Circuit using the recurrent neural network-based algorithm (RNN-based algorithm), which yielded better results for placement and routing of VLSI circuits and simulated it to estimate the congestion for the circuit design. A recurrent neural network, an artificial neural network, uses sequential data or time series data. To handle problems that involve order or time, such as image captioning, language translation, and speech recognition, deep learning algorithms are frequently utilized. Learning occurs through the training data in RNNs, just as it does in feed-forward RNNs and convolutional neural networks (CNNs). The information "remembered" from earlier inputs can then be used to modify the current input and the output. This sets them apart from other systems. As they are currently understood, deep neural networks work under the assumption that inputs and outputs are independent of one another. On the other hand, the output of recurrent neural networks is, nevertheless, influenced by the information processed in the network before it. Although the occurrence of events in the future might help in determining the outcome of a particular sequence, unidirectional recurrent neural networks cannot consider these occurrences when generating predictions.

Keywords: *Congestion, Placement, RNN-based algorithm, VLSI, Partial blockage technique.*

1. INTRODUCTION

Identifying the shortest way of using the least expense from the beginning of the computational logic block to the end of the computational logic block in the chip area is one old but relevant issue in graph theory. Achieving the briefest path issue among computational logic blocks is significant in designing VLSI circuits, where the shortest path problem improves routing and placement. Traditional approaches and artificial intelligence algorithms make up most of the algorithms used to solve the shortest path problem. Dijkstra algorithm, A* algorithm, and the like, are among the most commonly used techniques. Numerous sensible deep learning methods have been developed since the turn of the 20th century as synthetic sensible technology improved. To solve the shortest path issue, several algorithms have been created, including the simulated annealing algorithm (SA), the particle swarm deep learning method (PSO), the genetic algorithm (GA), the Tabu search algorithm (TSA), and the ant colony algorithm. However, such conventional techniques cannot fulfill the operational performance requirements for excessive dynamic topology. Besides, the handiest can achieve the shortest route among points. They cannot discover an institution of the shortest route or the second shortest route. Consequently, most synthetic sensible algorithms not only have superior time performance in solving the shortest route problem but can also obtain the best results, making them effective tools for finding the shortest path. A heuristic global deep learning search method, the harmony search algorithm (HSA), mimics the improvisational process of a musician [7-9]. While GA and PSO have shown promising results in some cases, our algorithm has shown to be superior. The collection of rules generates a path based on the worry values of the nodes in the concord vector. It may take advantage of upgrading its concord memory by upgrading its concord memory. In addition, the set of rules improved the perturbation equations, which enhanced the performance of looking at the shortest path. Deep learning is everywhere and is, therefore, an essential paradigm with various packages. In nearly all engineering and industry packages, we usually seek to optimize something—whether or not to decrease the fee and electricity consumption or maximize the profit, output, overall performance, and performance.

In reality, assets, money, and time are usually limited; consequently, deep learning is essential in practice. The most useful use of to-be-had assets of any kind calls for a paradigm shift in scientific thinking, which is because maximum real-international packages have a long way more complex elements and parameters that affect how the device behaves. Contemporary engineering layout is primarily based on PC simulations. This introduces extra problems to deep learning. The growing call for accuracy and the ever-growing complexity of systems and structure outcomes in the simulation system is increasingly time-consuming. In many engineering fields, the assessment of an unmarried layout can take so long as numerous days or even weeks. Any techniques that could accelerate the simulation time and deep learning methods can accordingly save time and money. For any deep learning trouble, the included additives of the deep learning method are the deep learning set of rules, a green numerical simulator, and a realistic illustration of the bodily techniques we want to version and optimize. This is mostly a time-eating method, and the computational prices are generally very high in many cases. Once we've got a very good version, the common computation prices are decided with the aid of using the deep learning algorithms used for seeking and the numerical solver used for simulation. Search algorithms are the equipment and strategies for optimizing the trouble of interest. This look for optimality is complex, in addition to the aid of using the reality that uncertainty nearly continually offers inside real international systems. Therefore, we are looking for the most suitable layout and a strong one in engineering and industry. Optimal answers, which aren't strong enough, aren't realistic. Suboptimal answers or top strong answers are frequently the selection in such cases. Simulations are frequently the maximum time-eating part. In many applications, a Deep learning method frequently includes the assessment of goal function many times, frequently heaps or even tens of thousands and thousands of configurations. Such reviews frequently use massive computational equipment together with a computational fluid dynamics simulator or a finite detail solver. Therefore, a green deep learning set of rules in aggregate with a green solver is extraordinarily important.

2. LITERATURE SURVEY

In [1], the authors briefly introduced various deep learning algorithms using the traveling salesman problem adopted for the routing and placement for application design. The work helps differentiate deep learning algorithms for IC design concepts. On the other hand, the paper [2] concentrates on the experimental analysis of deep learning techniques for placement and routing in ASIC design, which adopts a genetic algorithm and differential algorithm for various experimental analyses to allot macroblocks in a chip, and it shows a considerable reduction in area and wire length. The effort continues to reduce these parameters in [3], using nearest neighbors and SA-simulated annealing deep learning algorithms for various iterations to reduce the design's area and length further. The contributions of [4] in "Power, Performance and Area Deep Learning of I/O Design" emphasize the power consumption of input and output devices affecting the design performance. The study reveals a 20% reduction in power consumption and enhanced design performance. Additionally, G. Shivani and S. Neeraj discussed placement approaches using various techniques and showed the results using a synthesis process [5]. In A Study of Floor

Planning Challenges and Analysis of Macro Placement Approaches in Physical Aware Synthesis," the researchers used the VIVADO tool to estimate congestion using congestion reduction techniques in the horizontal space and in the vertical space directions and estimated GRC for the architectures they considered and showed significant improvement. IC Compile Employment User Guide Account [6] is a guide detailing the simulation and synthesis of any design simulated using the IC tool. In [7], "Physical Design Implementation Challenges in Highly Memory-Intensive Design in 40 nm," the authors used the software platform to implement memories and reduce the challenges that arise during placement in 40 nm technology.

In [8], a proposed deep learning process for quick floor planning macroblocks, including respective constraints, presents a 10% quick floor planning process compared to existing techniques. The contributions in [9] pioneered the evolution of the 3D experience and advanced study for estimating congestion for successful placement of blocks in chip areas. Moreover, in [10], the researchers utilized an ISE environment to estimate silicon-on-chip (SOC)_technology congestion. This work implemented a new methodology of placing blocks of computational logic blocks (CLBs) rather than individual CLBs because the run time for placing CLBs on-chip area for SOC technology has increased. Contrarily, the contributions of the authors in [11] include estimating real-time parameters and relationships, among others, during floor planning, which were estimated using voltage island–driven floor planning methodology. In [12], the authors recommended an "obstacle-aware clock-tree shaping during placement" where the design description can be estimated while considering clock signals, unlike existing methodologies that do not consider clock signals' effects. In [13], "Comparison of Hierarchical Mixed-Size Placement Algorithms for VLSI Physical Synthesis," the researchers showcased essential contributions done with placement algorithms in the physical synthesis of VLSISD. Specifically, they developed a crucial parametric comparative analysis and explained the performance of all placement algorithms with the related parameters. In [14], the authors explained the possible challenges during floor planning and the after-effects of design completion. Lastly, the authors' contributions in [15] discussed the effects of congestion and its value in placement and suggested a technique to reduce it by considering a reconfigurable processor.

3. METHODOLOGY

3.1 Proposed block diagram:

Backpropagation through time (BPTT) is a strategy for calculating gradients in recurrent neural networks (RNNs) that differs slightly from ordinary backpropagation since it is focused on sequence data. To train a model, the model calculates errors from its output layer to its input layer, like how a classical backpropagation does. These simulations allow us to fine-tune and improve the model's parameters. Because feed-forward networks do not exchange parameters across layers, they do not need to sum errors, whereas BPTT estimates the total amount of errors at each time step, as shown in Figure 1.

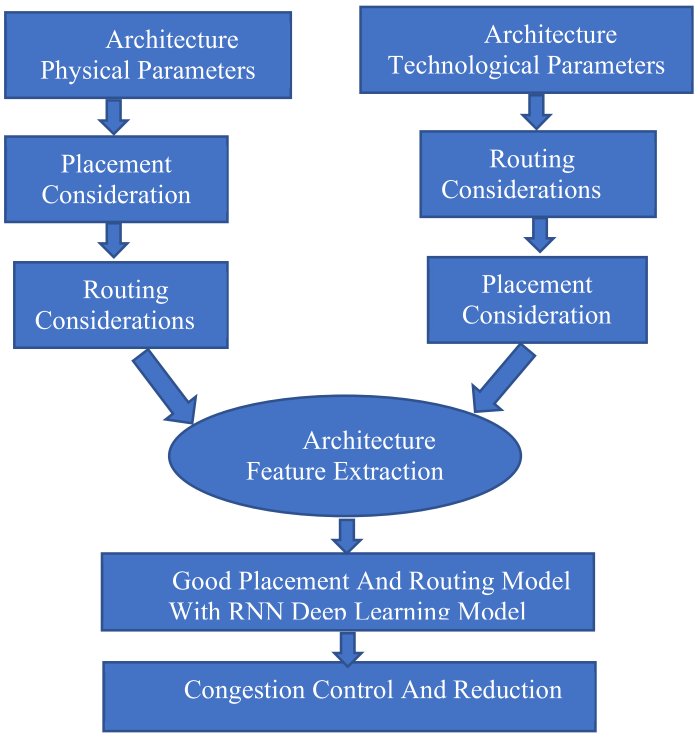

Architecture Physical Parameters
Architecture Technological Parameters
Placement Consideration
Routing Considerations
Routing Considerations
Placement Consideration
Architecture Feature Extraction
Good Placement And Routing Model With RNN Deep Learning Model
Congestion Control And Reduction

Figure 1. Methodology For Congestion Control With RNN.

As a result of this process, RNNs frequently experience "exploding gradients" and "vanishing gradients." These concerns are defined by the gradient, which is the slope of the loss function along the error curve. As the gradient declines, the weight parameters are updated until they are no longer significant (i.e., 0). When this happens, algorithms stop learning. The model becomes unstable when the gradient is too large, resulting in explosive gradients. If this occurs, the model weights become unmanageable and are presented as a negative number, or NaN. Hence, reducing the number of hidden layers in the RNN model is needed to reduce the RNN model's complexity.

3.2 Recurrent neural network

In recent years, RNNs have gained popularity as an artificial neural network. Unlike feed-forward networks, RNNs have recurrent connections, where the network may refer to previous states and process arbitrary input sequences. Figure 2 illustrates the RNN in its most basic form. This RNN has an MLP, and the inputs and the previous set of hidden unit activations are fed back into the network. At each discrete time step, the activations are updated, and the delay unit holds the updates until the next discrete time step.

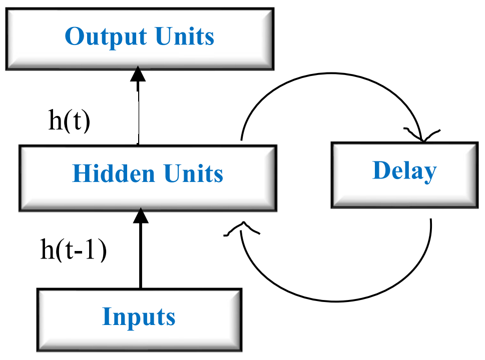


Figure 2. The RNN Structure.

3.3 Process flow and mathematical model

Estimating congestion depends upon the simulation output generated by deep learning and the type of inputs applied with the deep learning RNN method. Maximum congestion density exists in the on-chip area due to two factors—pin density and cell density. Pin density in the chip area is caused by the increase in cell or macro or CLBs space and the introduction of blockage. Cell density, on the other hand, relies on the type of architecture selected and the number of macros or computational logic blocks in its design within the available chip area.

3.4 Steps in estimating congestion through the simulation of congestion

1. Select sequential files as input using the recurrent method. These files provide timing information on the circuit and the logical relationships among various CLBs. The sequence is provided by eqn.

$f(x) = n1, n2, n3, \ldots \ldots \ldots . . n33 \ldots \ldots \ldots \ldots \ldots$ (1)

i. where $n = CLBs$ of ami33 MCNC architecture
   ii. The area occupied by each block is given by
   iii. $A1 = x1 *$ $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$ (2)
   iv. $A2 = x2 * y2 \ldots \ldots \ldots \ldots \ldots \ldots$ (3)
   v. For all blocks up to n blocks, the areas of individual blocks are as shown above,
   vi. where x1 and y1, x2 and y2, x3 and y3, and x4 and y4 are lengths and widths of all CLBs.
2. Select .lef files as input.
   These files provide physical information about the CLBs of the respective architecture.
3. Select the Optimized Floorplan design as input.
4. Provide the synopsis design information.
5. Provide the synthesized netlist information about the architecture as input.
6. Select .lf file to provide all layer information (i.e., from m1 to m9 for the entire architecture).
7. Simulate or run the architecture through the "Scandef" function, through which the placement of CLBs can be completed through the automatic placer.
8. Based on the congestion methodology involved, we can simulate the architecture through pin density and partial blockage congestion techniques for the estimation of the congestion of the architecture.
   a. The tool estimates the length between two blocks using the following simple mathematical formula:
   b. $WL = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \ldots \ldots$ (4)
   c. where X_2, X_1, and Y_2, Y_1 are the lengths and widths of the first block placed on the X and Y axes.
   d. Congestion is considered the densities of wire lengths among all CLBs: WL_1, WL_2 ,……………………WL_n
9. The recurrence-based approach generates a congestion estimation report and the corresponding figures that indicate the density distribution of routing paths amongst the CLBs of the architecture.
10.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. SIMULATION RESULTS AND DISCUSSION

Subsequent utilization of excitation architecture, which finished the floorplan using the harmonic improved search optimized method, the apparatus acknowledges the design as the contribution with the related number of CLBs and organizes all blocks on the Chip region as per its self-built automatic placer.

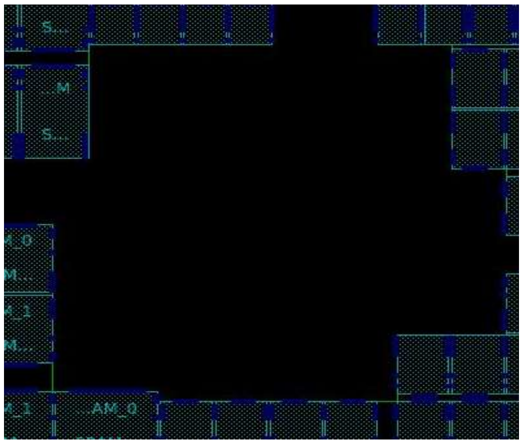


Figure 3. Block Placement With RNN.

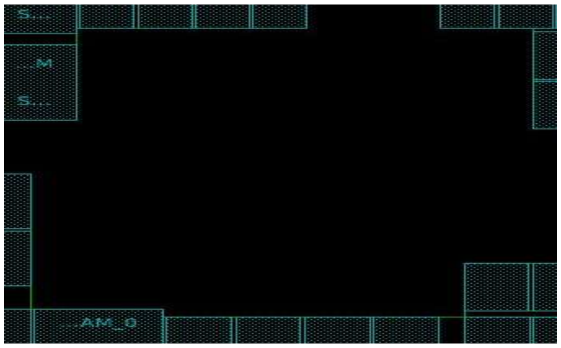
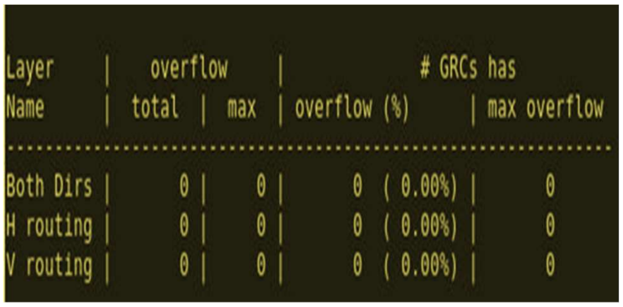


Figure 4. Pin Allocation Using RNN.

After putting macros or computational squares, the device starts the pin task cycle to all the macros or CLBs in chip territory. This cycle will be finished by the apparatus in an awry way (i.e., all the macros or CLBs were given equivalent significance to have pins at their appearances). For all the large-scale blocks or CLBs at the potential faces, the apparatus appoints info and yields pins with less conceivable pin thickness and less possible cell compactness. With the assistance of hardware typical capacities, we will adjust the dimension of the inserted macroblocks and the spots of the available connecting pins outside of the large-scale cells or CLBs. In the wake of mimicking the apparatus, it produces the position of large-scale blocks or CLB with their territory on-chip space, as demonstrated in Figure 2. It illustrates the pins accessible at the essences of the separate macros. Pins related to the particular macros or CLBs appeared in blue tones and the macros or CLBs appeared as quadrangular and rectangular boxes with green tone by tool. The apparatus comprises of predefined in-assembled approach for dispensing places for large-scale blocks or CLBs and pins for them on the chip zone, as exemplified in Figure 3. After culminating the placement of the large-scale blocks or CLBs beside pins aimed at the elements, we reenact the device to achieve congestion analysis.

Figure. 5. Initial Test Congestion Estimation Report



```
Layer     |   overflow     |              # GRCs has
Name      |  total  | max  | overflow (%)      | max overflow
-------------------------------------------------------------
Both Dirs |       0 |    0 |       0  ( 0.00%) |        0
H routing |       0 |    0 |       0  ( 0.00%) |        0
V routing |       0 |    0 |       0  ( 0.00%) |        0
```

For RNN-Based Architecture.

By mimicking the order “report_congestion,” the instrument creates congestion analysis for the RNN-based circuit that contributes to the apparatus. Figure 5 describes the beginning congestion analysis report with the engineering subtleties and aggregate and greatest flood in even and vertical ways for layers and the directing among the CLBs. Post-recreation results show the assessment of horizontal steering assets and vertical directing assets. Furthermore, the apparatus gives data about the all-out flood, maximum flood for, H-steering, V-directing, both directions and similar assessments for GRC. On the off chance that we need any progressions for the situation of full-scale blocks or CLBs and relegating pins at the essences of squares, we are permitted to do such using the instrument and demand for the adaptation about them. When this solicitation is finished, the device produces a congestion assessment figure with blockage thickness appropriation on chip region with the assistance of different tones, as demonstrated in Figure. 6.

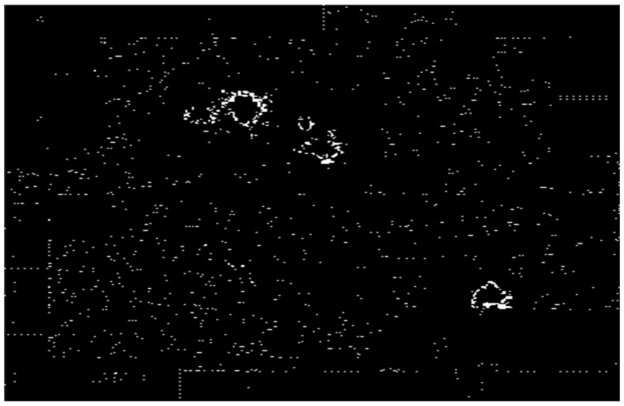
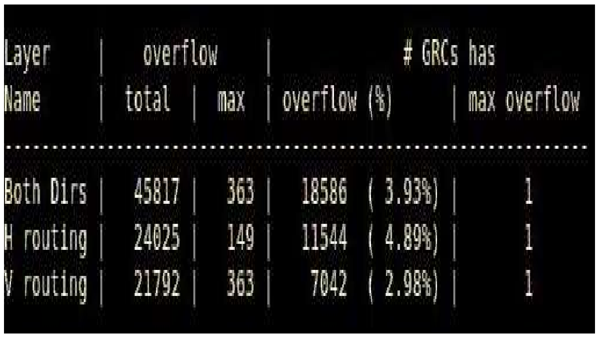


Figure. 6. Congestion Spreading In RNN Architecture.

The device shows the general blockage with four shadings—the pink tone speaks to less congestion, the blue tone refers to direct blockage, the yellow shading shows impressive blockage, and the red tone speaks to the greatest blockage at particular places on the chip region. It can be gleaned from Fig. 6 that the vast majority of the congestion density is at the pins beside the spots close to the greatest cell thickness. Even though the deep learning calculation gives the best outcome for the placement and routing of macros or CLBs in the chip zone, it gives negligible blockage to the IC plan. To decrease the congestion of a similar design with deep learning, an arrangement is furnished with the assistance of the pin thickness congestion procedure diminishing the blockage. After finishing the placement of macros or CLBs, we need to reproduce the apparatus using "report_congestion" to achieve a congestion report. Fig. 7 shows the blockage report for the RNN-cased circuit by Horizontal Routing (HRouting) esteem 24025, Vsteering esteem 21792, and absolute flood of 45817. The GRC simulation for the RNN-based circuit with Horizontal Routing esteem 11544, V-directing worth 7042, and complete flood 15856 (i.e., in H, heading 4.89 percent, and in V way, 2.98% through all-out flood blockage esteem as 3.93%).



```
Layer     |    overflow    |              # GRCs has
Name      |  total  |  max | overflow (%)      | max overflow
----------------------------------------------------------------
Both Dirs |  45817  |  363 |  18586  ( 3.93%) |      1
H routing |  24025  |  149 |  11544  ( 4.89%) |      1
V routing |  21792  |  363 |   7042  ( 2.98%) |      1
```

Fig. 7. Congestion Report Before Using The RNN Congestion Technique.

Table 1: Congestion Table Using RNN.

Layer name	Overflow		#GRC		
	Total	Max	Overflow	Percent (%)	Max overflow
Both-Dirs	45817	363	18586	3.93	1
H-R	24025	149	11544	4.89	1
V-R	21792	363	7042	2.98	1

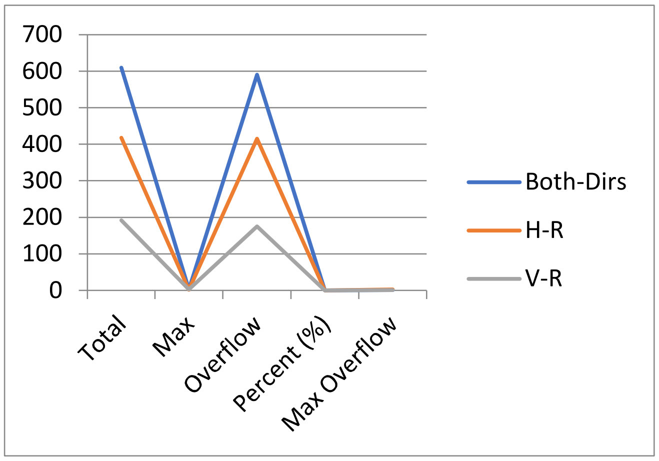


Figure 8. Graphical Representation Of The Congestion Report.

The underlying recreation report estimates a high level of congestion. To decrease this, we need to apply a pin thickness blockage strategy or cell-dispersing congestion method. To decrease blockage by utilizing deep learning methods with fewer assignment adjustments among Macro-blocks or CLBs, we want to choose "Pin Density" and mimic the design by using the command "report_congestion."

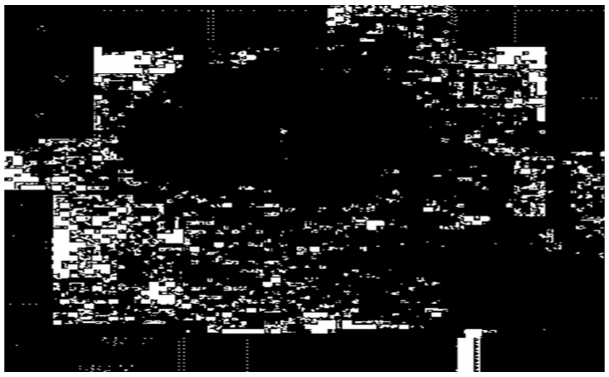
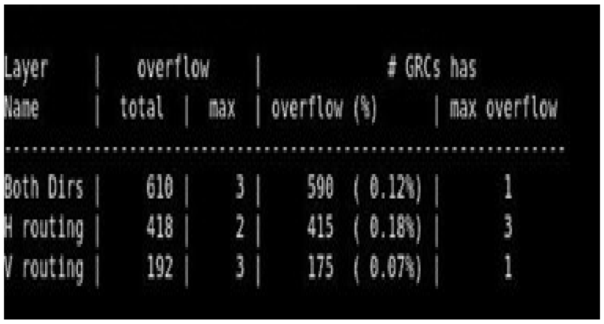


Figure 9. Congestion Circulation After The RNN Congestion Procedure



```
Layer     |  overflow   |          # GRCs has
Name      | total | max | overflow (%)     | max overflow
..........................................................
Both Dirs |   610 |   3 |   590  ( 0.12%) |      1
H routing |   418 |   2 |   415  ( 0.18%) |      3
V routing |   192 |   3 |   175  ( 0.07%) |      1
```

Fig. 10. Congestion Report After Applying The RNN Congestion Technique

The post-recreation results suggest a critical decrease in blockage for the design and separate reenactment report as portrayed in Fig. 7. The simulation result displays a decrease in congestion for the RNN-based circuit with an H-routing esteem of 418, a V-steering esteem of 192, and an all-out flood of 610. The Global Routing Congestion simulation report for the RNN-based circuit has an H-routing esteem of 415, V-steering esteem of 175, and all-out flood of 590 in a Horizontal course of 0.18 percent and vertical space of 0.07% with all-out flood blockage esteem as 0.12%. Figure 8 shows a decrease in blockage of routings at the edge spaces (i.e., pins of CLBs). The pin thickness congestion method diminishes blockage as the reproduced simulation result does not have a red tone at any spot of the chip region representing a significant decrease in blockage for the architecture.

Table 2: Tabular Representation Of The Congestion Report After Applying The RNN Congestion Technique

Layer name	Overflow		#GRC		
	Total	Max	Overflow	Percent (%)	Max overflow
Both-Dirs	610	3	590	0.12	1
H-R	418	2	415	0.18	3
V-R	192	3	175	0.07	1

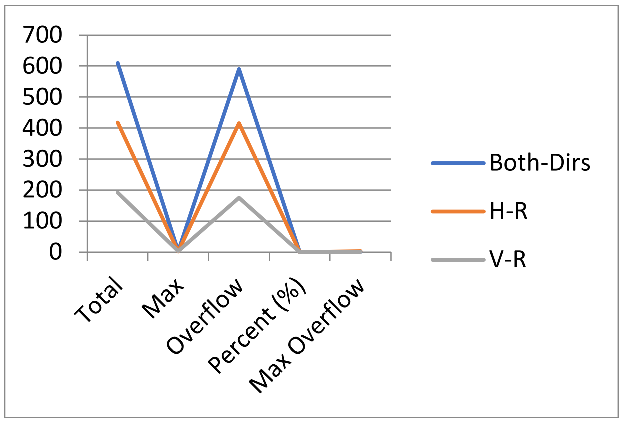


Figure 11. Graphical Representation Of The Congestion Report After Applying The RNN Congestion Technique

Table 3: Comparison Of The Proposed And Existing Congestion Estimation Techniques

Congestion at	Flyline congestion analysis	Data flow congestion analysis	Congestion technique using RNN
V direction Overflow	260	169	211
H direction Overflow	118	9	415
Overflow in both directions	378	178	625
Max Overflow (1GRC)	22	22	18

After getting the outcomes of the simulation, the projected congestion estimation method aimed at the architecture generating a report of the overflow characteristics in horizontal space, overflow values in vertical space, and total overflow, along with maximum overflow for a unit GRC compared to other methods like data flow analysis and FlyLine analysis. The proposed partial blockage congestion procedure gives a good estimate of congestion in the available chip space in H-V directions (vertical, horizontal, and GRC-Global Routing Congestion), as represented in Table 1 and. The relationship is shown with the help of the graph in Fig. 10.

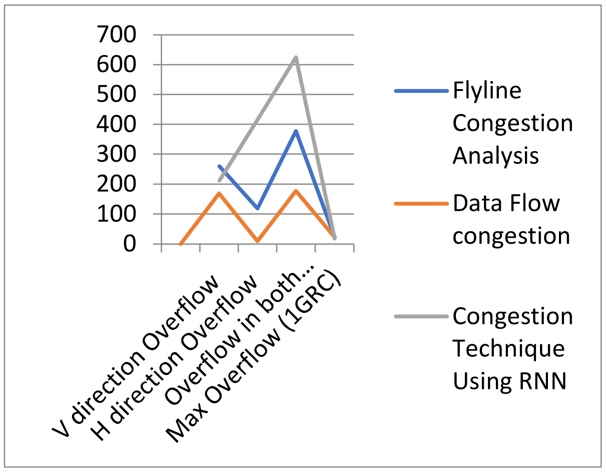


Figure. 12. Graphical Representation Of The Comparative Analysis

5. LIMITATIONS

A specific concern of this research is the employment of unidirectional recurrent neural networks (RNNs) which, because of their nature, cannot consider future events in the course of sequence prediction, which in turn can affect the accuracy of congestion estimation. This study is in its own way limited in scope since it makes use of the MCNCBM circuit benchmark that might restrict the applicability of the findings to other VLSI designs which have different features. Besides, the concern is mostly on congestion however, other important issues in VLSI Design such as power dissipation, signal integrity, and thermal issues are not considered. The trained RNN model has significant reliance on the quality and representativeness of the training samples, which may induce a weakness in the approach's robustness. In addition, even though the method has been proven to be effective, scalability can be an issue when using it on larger or more intricate circuits which may require enormous computational power. Such weaknesses define what can be improved and what can be considered in further studies in order to enhance the methodology.

6. CONCLUSION

In this work, the congestion of the designs in a VLSI circuit design is modeled using deep learning architecture housed within a recurrent neural network. The proposed approach is advantageous as it has lower placement and routing costs for standard MCNCBM circuits as compared to the traditional methods using floorplan and placement outputs. The research highlights how essential it is to deploy deep learning strategies to manage congestion so that the area available is optimally used resulting in miniaturized circuits. The RNN offers a salient advantage in recognizing the interdependencies of the computing blocks as it analyzes sequential data which in turn improves the effectiveness of the design.

Ethics statements

The authors have complied with ethical guidelines during this work, and no data was generated during this study. The details provided in the work have been carried out by the requirements of the Centre for Research and Consultancy and Technology Transfer office at Middle East College. The Article contains no studies with Human Participants performed by any authors.

CRediT author statement

Shaik Karimullah: *Conceptualization, Asif: Methodology, Software, Data curation, Writing- Original draft preparation, Visualization, Investigation.* ***Syed Javeed Basha****: Software, Validation.* ***Fahimuddin Shaik:*** *Writing- Reviewing and Editing. For the titled manuscript, there are four authors, and each author has contributed in their way to support and obtain better results for the given work. Asif is the primary author who has taken the most responsibility for performing the work. Shaik Karimulla: holds the responsibility of simulation and writing the reflections for the literature* ***Syed Javeed Basha*** *:reviewed conceiving the design analysis, and* ***Fahimuddin Shaik*** *revised the manuscript.*

Acknowledgments

The authors would like to acknowledge the support of Middle East College and Annamacharya University for providing the facilities and support. This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] M.T. Akhtar, W. Mitsuhashi, Improving performance of the FxLMS algorithm for active noise control of impulsive noise, J. Sound. Vibr. 327 (2009) 647–656.

[2] C.L. Nikias, M. Shao, (1995). Signal processing with alpha-stable distributions and applications. New York: Wiley. Shao M and Nikias CL. Signal processing with fractional lower-order moments: stable processes and their applications.Proc IEEE 1993; 81: 986–1010

[3] I.T. Ardekani, W.H. Abdulla, On the stability of the adaptation process in active noise control systems. Journal of the Acoustical Soc. Am., 1291 (2011), 173–184.

[4] W. Wierzchowski, Review of active noise control algorithms for impulsive noise control. Pomiary, Automatyka, Kontrola 60 (2014) 358–361.

[5] S. Rao, Mechanical Vibrations, fourth ed., Pearson Prentice-Hall, Upper Saddle River, New Jersey, 2004.

[6] D. Shaik Karimullah, M.A. Vishnuvardhan, F.S. Vinit Kumar Gunjan, Kazy Noor-e-alam Siddiquee, An Improved Harmony Search Approach for Block Placement for VLSI Design Automation, Wirel. Commun. and Mob. Computing (2022), doi:10.1155/2022/3016709

[7] S.J.B. Shaik Karimullah, P. Guruvyshnavi, K. Sathish Kumar Reddy, B. Navyatha, A genetic algorithm with fixed open approach for placements and routings, ICE Publisher Springe (2020) 599–610.

[8] D.V. Shaik Karimullah, Experimental analysis of optimization techniques for placement and routing in ASIC design, ICDSMLA 2019, Lecture Notes in Electrical Engineering 601, Springer Nature Singapore Pte Ltd., 2020.

[9] D. Shaik Karimullah, V. Vardhan, S. Javeed Basha, Floorplanning for placement of modules in VLSI physical design using HarmonySearch Technique, ICDSMLA 2019, Lecture Notes in Electrical Engineering 601, Springer Nature Singapore Pte Ltd., 2020.

[10] Y. Kajikawa, C. Shi, Comparison of virtual sensing techniques for broadband feedforward active noise control, 2019 International Conference on Control, Automation and Information Sciences (ICCAIS). IEEE, 2019.

[11] L. Ying et al., Application study of adaptive tracking algorithm in active noise control system of transformer, Appl. Sci. 9.13 (2019) 2693.

[12] P. Hariobulesu et al., "Extreme Learning Machine (ELM) Algorithm for the Detection of Diabetic Retinopathy," 2024 International Conference on Electronics, Computing, Communication and Control Technology (ICECCC), Bengaluru, India, 2024, pp. 1-6, doi: 10.1109/ICECCC61767.2024.10593933.

[13] Karimullah, S., Vishnuvardhan, D., Gunjan, V.K., Shaik, F. (2024). Improved Spectral Efficiency Using Vehicular Visible Light Communication with 16-Bit DCO in OFDM. In: Gunjan, V.K., Zurada, J.M., Singh, N. (eds) Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough. Studies in Computational Intelligence, vol 1117. Springer, Cham. https://doi.org/10.1007/978-3-031-43009-1_15.

[14] K.A. De Jong, W.M. Spears, D.F. Gordon, Using Markov chains to analyze GAFOs. In D. Whitley and M. Vose (eds.), Foundations of Genetic Algorithms 3, Morgan Kaufmann, San Mateo, CA, 115–137.

[15] J. Rees, G.J. Koehler, An investigation of GA performance results for different cardinality alphabets. In L.D. Davis, K. DeJong, M.D. Vose and L.D. Whitley (eds.), Evolutionary Algorithms, 1999.

[16] M. Mitchell, J.H. Holland, S. Forrest, When will a genetic algorithm outperform hill climbing? In J.D. Cowan, G. Tesauro and J. Alspector (eds.), Advances in Neural Info., 1994.

[17] R. Reeves, Genetic algorithms for the operations researcher. INFORMS J. Comput. 9 (1997), 231–250.