Implementation of a Graph Neural Network within OpenROAD

Kevin Guan*
Henry Samueli School of Engineering, B.S. Electrical Engineering
University of California, Irvine, California
Email: *guankw@uci.edu

***Abstract*—A graph-based neural network is proposed to facilitate runtime improvements during floorplanning by offering an alternative to the RePlAce utility already provided by OpenROAD for end-users.**

***Index Terms*—GNN, OpenROAD**

I. Introduction

Floorplanning has traditionally been a difficult aspect of ASIC design to automate. Many human hours and other resources are spent on physically designing and verifying the outcomes of this step in the chip design flow. However, tools such as OpenROAD seek to reduce human inspection of the placement of such IP blocks, especially when modern architectures include billions of macros and standard cells [1]. However, tools like OpenROAD's RePlAce are based on analytical methods that can degrade runtime performance. Other possible methods used for this core functionality include partitioning and stochastic ones. This proposal opts for a graph-based neural network to complete the floorplanning stage for OpenROAD, offering it as an option for end users.

II. Method

Since the OpenROAD codebase is written in C++, Tensorflow will be used to develop the edge-based GCN. This will involve porting the models in [2] to implement the base functionality of the system. Potential modifications could include utilizing an Long Short-Term Memory (LSTM) layer which can be utilized to capture past decisions utilized in the architectural exploration process that includes floorplanning.

An adjacency matrix will be used as the input to the GNN that consists of an edge-based GCN and feed-forward networks. The weights are tuned to meet the constraints of power, performance, area, and cost (PPAC) through wirelength, congestion, and density. Hyperparameters specified by [3] will be used as a baseline from which modifications can be done to optimize the OpenROAD flow for each custom silicon design. The loss function used to modify the weights will be the mean squared error (MSE) based on wirelength and congestion with the ReLU (rectified linear unit) activation function. Density is considered a hard constraint and will be solved as a Linear Programming (LP) problem. Different open-source libraries include Google Or-Tools are potential candidates for calculating feasible regions based on probability distributions, which also has been used in design space exploration [4].

Once the macros are placed, a partitioning method will be utilized to form standard cell clusters that are optimized using a force-directed analytical method to complete the floorplanning. This floorplanning utility will deploy the open-source Ariane RISC-V CPU as a test case [5].

III. Conclusion

The objective of this investigation is to implement a Graph-based Neural Network to optimize for power, performance, area, and cost given the constraints of wirelength, congestion, and density. The results of this pull request will directly benefit flows that utilize the OpenROAD tool such as OpenLane and OpenFASOC. This tool can be applied to chiplet-based designs and to facilitate hardware acceleration.

Acknowledgment

The author would like to thank the VSD team for their workshops and programs that provided learning outcomes targeting advanced VLSI topics and open-source tools. The author would also like to thank Matt Liberty for providing direction on the different avenues toward improving runtime for OpenRoad and opportunities available for GSoC 2023.

References

[1] "OpenROAD," Mar. 2023, original-date: 2019-10-28T17:48:14Z. [Online]. Available: https://github.com/The-OpenROAD-Project/OpenROAD

[2] "Circuit Training: An open-source framework for generating chip floor plans with distributed deep reinforcement learning." Mar. 2023, original-date: 2021-11-02T17:44:11Z. [Online]. Available: https://github.com/google-research/circuit_training

[3] A. Mirhoseini, A. Goldie, M. Yazgan, J. W. Jiang, E. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, A. Nazi, J. Pak, A. Tong, K. Srinivasa, W. Hang, E. Tuncer, Q. V. Le, J. Laudon, R. Ho, R. Carpenter, and J. Dean, "A graph placement methodology for fast chip design," *Nature*, vol. 594, no. 7862, pp. 207–212, Jun. 2021, number: 7862 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41586-021-03544-w.

[4] S. Li and A. Hemani, "Three-Dimensional Design Space Exploration for System Level Synthesis," in *2014 17th Euromicro Conference on Digital System Design*, Aug. 2014, pp. 419–426.

[5] F. Zaruba and L. Benini, "The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-Ready 1.7-GHz 64-Bit RISC-V Core in 22-nm FDSOI Technology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 11, pp. 2629–2640, Nov. 2019, conference Name: IEEE Transactions on Very Large Scale Integration (VLSI) Systems.