Fully Automated Analog Sub-Circuit Clustering with Graph Convolutional Neural Networks

Keertana Settaluri
University of California, Berkeley
Berkeley, CA, USA
ksettaluri6@berkeley.edu

Elias Fallon
Cadence Design Systems
Pittsburgh, PA, USA
fallon@cadence.com

***Abstract*—The design of custom analog integrated circuits is a contributing factor in high development cost and increased production time, driving the need for more automation. In automating particular avenues of analog design, it is then crucial to assess the efficacy with which the algorithm is able to solve the desired problem. To do this, one must consider four metrics that are especially pertinent in this area: robustness, accuracy, level of automation, and computation time. In this work, we present a framework that bridges the gap between schematic and layout generation by encapsulating the design intuition needed to create layout through identification of critical sub-circuit structures through the use of Graphical Convolutional Neural Networks (GCNNs) along with an unsupervised graph clustering technique. This framework is the first tool, to our knowledge, to entirely automate this clustering process. We compare our algorithm to prior work utilizing the four figures of merit, and our results show over 90% accuracy across six different analog circuits, ranging in size and complexity, while taking just under 1 second to complete.**

***Index Terms*—analog design automation, analog clustering, graphical convolutional neural networks**

I. Introduction

The design of analog integrated circuits (ICs) is cumbersome, making it one of the contributing factors in increased custom ASIC development cost and production time. This is in part because the design process heavily involves human expertise and intuition in order to create topological circuit structures that not only meet particular design objectives, but must also satisfy complex constraints and be robust to non-idealities. Analog automation efforts to mitigate the amount of manual design in analog circuits is therefore crucial to the ASIC development industry.

In order to assess the efficacy of an automated analog synthesis tool that encapsulates design knowledge, it is important to consider several metrics of interest specifically applicable to analog design automation:

- Robustness: ability to work on different analog circuits
- Accuracy: reliably output valid results
- Automation: mitigate involvement of the circuit designer
- Computational time: measure how long the algorithm takes to converge to a solution

In this work, we present a framework to bridge the gap between schematic and layout generation by encapsulating the design intuition needed to create layout through the identification of critical sub-circuit structures.

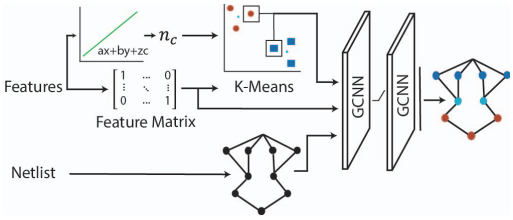

Fig. 1. Total system overview of the analog clustering algorithm

Prior work in this space can be split into three different categories: library based, knowledge based, and learning based approaches. Library based methods [2] [4] involve the manual creation of many sub-circuit structures in the form of templates; though they are accurate, these approaches lack robustness and automation, as these templates are often circuit specific. Knowledge based methods [3] codify rules to create constraints when generating clusters, and present the same pitfalls. Learning based [1] approaches use supervision to learn sub-circuit structures; because data must be labelled, they lack automation as well. In addition, most make use of graph or hierarchical structures to represent a circuit topology, and often ignore additional features of the netlist that could aid in more accurate structure recognition.

In this paper, we present a novel approach to the analog sub-circuit clustering problem by leveraging the ability of learning algorithms to recognize important features, while also generating the appropriate data needed to train this setup via an unsupervised clustering algorithm. We demonstrate that this methodology yields accurate results that work very well across different types of analog circuits with differing complexity, all while being entirely automated.

II. Framework Implementation

Figure 1 shows the proposed framework. The inputs are the circuit netlist and input features extracted from the netlist. These features include: length, the x and y coordinate of each device in the schematic, device orientation, device type and device weight (a measure of multiplicity and width), all of which are one-hot encoded into a feature matrix.

In order to train the graphical convolutional neural network (GCNN), two pieces of information are needed: a graph representing the circuit netlist and one labelled device per

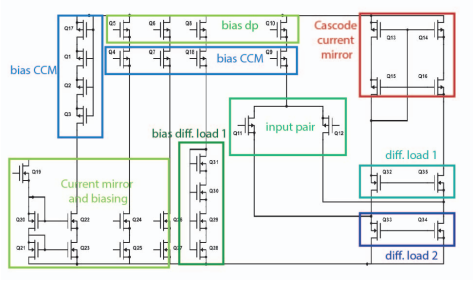

Fig. 2. Folded cascode with biasing schematic with clustering

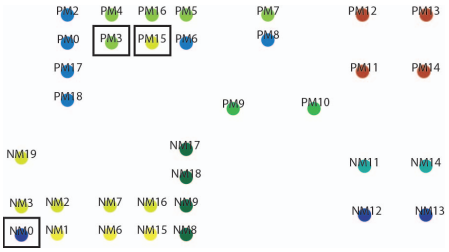

Fig. 3. Output of GCNN algorithm visualized, the black boxes show mislabelled devices

cluster in the circuit. For the graph, each node represents a device and each edge represents the corresponding net connections.

To generate the labels automatically, a traditional k-means algorithm is run on the extracted input feature matrix. K-means requires the correct number of clusters as input, however. To prevent our framework from requiring this, we use a linear model that takes as input total number of instances, number of N-type and P-type instances and number of nets in the circuit topology to predict the cluster count. The resulting output of the k-means algorithm then utilizes converged centroid locations and the device distances as a notion of confidence, wherein the device closest in distance to each centroid is used as the designated label. The graph and labels are then fed to a two layer, 50 neuron GCNN classifier, and trained until loss converges to a stable value. Note that the clustering accuracy was measured by hand-labelling the results, these labels were not used in the GCNN classifier training.

III. Experimental Setup and Results

Figure 2 shows a 35 transistor folded cascode circuit topology with biasing, along with the cluster assignments we tested against. The clusters are assigned based on the functionality of each sub-circuit, particularly focusing on the identification of differential loads, biasing, input pairs, and current mirrors for this circuit. Figure 3 shows the resulting node classification output of the GCNN, where the folded cascode obtains 91.3% accuracy. Note that the accuracy metric is calculated by comparing the correct cluster identifications with those obtained by the algorithm, device by device.

Table I shows how this algorithm compares with prior work. We test our algorithm on circuits with similar orders

TABLE I
Comparison table showing maximum number of devices (Max. ND) tested, number of circuits tested, accuracy of algorithm, and whether or not the flow is automated

Algorithm	Max. ND	Num. Circuits	Accurate?	Auto.?
Library based [2]	22	2	Y	No
Knowledge based [3]	37	5	Y	No
ML based [1]	34	34	N	No
This work	35	6	Y	Yes

TABLE II
Maximum Accuracy of algorithm for multiple circuits

Circuit	Num. Devices	Num. Clusters	Accuracy
pll buffer	4	2	100%
folded cascode	35	9	91.3%
analog bias enable	5	2	100%
current reference	19	3	94.7%
opamp	15	5	93.8%
comparator	17	4	100%

of complexity, and demonstrate that the framework is just as robust as prior library and knowledge based algorithms. Note that despite [1] tested on 34 circuits, their paper reports many inaccuracies in cluster assignments due to redundant, overlapping, or generic templates.

In order to assess our algorithm in the context of the important design automation metrics, we consider each separately below:

- Robustness: tested on six different circuits of varying complexity, size, and technology (see Table II). The GCNN architecture remained the same for each topology.
- Accuracy: the maximum accuracy for each circuit is shown in Table II. We obtain a greater than 90% accuracy for all tested circuits.
- Automation: the designer is not involved in any part of the process, other than providing a schematic with correct sizing.
- Computation time: the algorithm takes a maximum of 1.13 seconds to complete for the most complex circuit.

In summary, our algorithm is the first to fully automate the analog sub-structure recognition task. We demonstrate that this framework is accurate and robust to multiple circuit topologies.

IV. Acknowledgments

The authors would like to thank Weiyi Qi and Sheng Qian of Cadence Design Systems for their contribution to this work.

References

[1] H. Li, "Analog circuit topological feature extraction with unsupervised learning of new sub-structures," Proc. DATE, pp. 1509-1512, 2016.
[2] M. Meissner and L. Hedric, "FEATS: Framework for explorative analog topology synthesis," IEEE T. Comput. Aid D., vol. 34, no. 2, pp. 213-226, 2015.
[3] P.-H. Wu, "A novel analog physical synthesis methodology integrating existent design expertise," IEEE T. Comput. Aid D., vol. 34, no. 2, pp. 199-212, 2015.
[4] T. Massier, "The sizing rules method for CMOS and bipolar analog integrated circuit synthesis," IEEE T. Comput. Aid D., vol. 27, pp. 2209-2222, Dec. 2008.