Novel Circuit Topology Synthesis Method using Circuit Feature Mining and Symbolic Comparison

Cristian Ferent and Alex Doboli
Department of Electrical and Computer Engineering
Stony Brook University, Stony Brook, New York, USA
Email: {Cristian.Ferent, Alex.Doboli}@stonybrook.edu

***Abstract*—This paper presents a reasoning-based approach to analog circuit synthesis using ordered node clustering representations (ONCR) to describe alternative circuit features and symbolic circuit comparison to characterize performance trade-offs of synthesized solutions. Case studies illustrate application of the proposed methods to topology selection and refinement.**

I. Introduction

Deciding an analog circuit's topology is a critical design step as the schematic primarily sets the performance. Often, an existing topology is resized to meet new specification. However, topology resizing might prove insufficient, if the performance trade-offs inherent to a topology cannot satisfy new requirements.Then, deciding a circuit's topology includes three situations: (i) *topology selection* involves picking another topology from an existing pool of schematics, (ii) *topology refinement* incrementally modifies existing topologies, and (iii) *topology synthesis* creates a completely new circuit.

Circuit topology synthesis has been more recently tackled using optimization-based techniques [1], evolutionary algorithms [2]–[4], and template-based synthesis [5]. In spite of some success, analog circuit topology synthesis has proved to be difficult to automate, especially if the generated schematics must be feature-wise similar to designs that humans create. This is important as synthesized topologies must be manually verified and validated. However, a recent study [6] shows that synthesized topologies might have completely different structural features than designer produced schematics. Synthesized circuits can have long signal paths, contain isolated paths, and incorporate strongly coupled input and output nodes compared to manually created topologies. Adding constraints to eliminate *unusual* features addresses some of the limitations, but does not guarantee that synthesized topologies are consistently similar to designer-created solutions.

Arguably, many topology synthesis methods fail to match the versatility and creativity of designer reasoning. To the best of our knowledge, there are no reported instances in which synthesized topological features became generally accepted by the circuit design community. A possible reason is that current techniques over-emphasize the importance of optimization/solving/evolution, even though topology synthesis is rather a decision making problem. Many current synthesis techniques are "black-box" methods, where the user does not receive much insight about the causal effect of synthesized structures on trade-offs and performance. In manual design, insight is a significant part of the reasoning process that creates effective and innovative solutions.

978-3-9815370-2-4/DATE14/©2014 EDAA

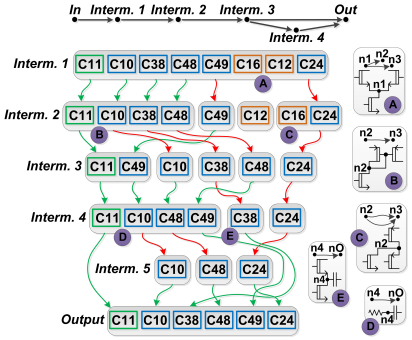

Fig. 1. ONCR fragment for a pool of fifty OpAmp circuits

This paper presents a new approach for circuit topology selection and refinement based on a reasoning-like process. The approach conducts successive steps in which the performance trade-offs, bottlenecks, and values of schematics are used to guide the process of selecting and incorporating only topological features likely to improve performance. The features are selected from a large pool of existing designs. Similar to reasoning in manual design, every synthesis step is justified by the trade-offs and bottlenecks that are improved by the related structural feature. Therefore, the synthesis output is not only a design, but also the justifications for the performed design decisions. Reasoning and knowledge-based approaches have been used by early analog CAD tools [7]–[9], but then dropped as automated reasoning was considered less effective. This work is based on a new representation to describe the structural similarities and differences in a pool of topologies and on a symbolic circuit comparison operator generating the performance trade-offs and values that distinguish two circuits, hence explaining the impact of specific structural features on performance. The paper presents case studies for which the final solutions are difficult to generate with optimization/solving/evolutionary approaches alone. The paper discusses topology selection and refinement, but the flow can be extended for topology synthesis too.

II. Overview of Proposed Algorithms

Circuit Representation: A pool of circuits is described using Ordered Node Clustering Representation (ONCR). ONCRs are automatically constructed using an entropy-based clustering scheme. Fig. 1 illustrates a fragment of the ONCR for 50 amplifier circuits. Fig. 3 presents the schematics of some considered circuits. ONCR is a graph to illustrate the similarity and variation of the structural features in set of topologies.

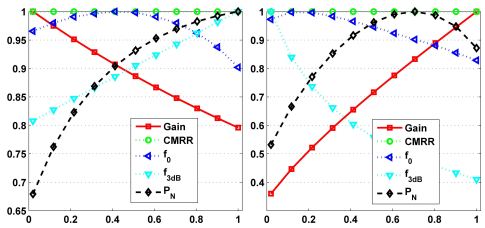

Fig. 2. Normalized performance of C_{38} w.r.t. $W_8 \propto W_9 \propto W_{10}$ ($I_8 \nearrow$) and W_6 ($I_8 \searrow$) ranges, respectively

Each ONCR group at a level represents a set of circuits that include a similar feature. Similarity is defined w.r.t. the pole and coupling of the corresponding node. A detailed description of ONCRs is offered in [10].

Symbolic Comparison: This operator takes two circuit schematics as inputs and generates output data indicating their differences w.r.t. the behavior of their nodal signals, performance trade-offs, and performance attributes (gain, bandwidth (f_0, f_{3dB}), CMRR, and noise (P_N)). The information offered by ONCR about the structural differences is used to compute symbolically the similar and distinguishing expressions of the nodal behavior and trade-offs. Trade-off expressions are plotted for specific circuit sizings as shown in Fig. 2 to represent the performance advantages and limitations of circuit C_{38} w.r.t. the other circuits in Fig. 3. A detailed description of the operator is offered in [11].

Topology Selection: Alg. 1 illustrates the topology selection method. The procedure takes as input a reference circuit topology C_k, the set S_C of N known designs for the given application, and the ONCR of this set. Input Max_{diff} is used to control the amount of variance with respect to the reference topology considered in exploration. The procedure outputs a sorted list L_k of alternative topologies.

Step 1 initializes the list of topology candidates to the set of known solutions.In Step 2, the list of candidates L_k is pruned based on the desired degree of differences w.r.t. reference C_k. Using the ONCR, only those designs are kept in L_k for which the number of unmatched structures (with C_k) at most Max_{diff}. The method checks if circuits C_j and C_k have same (or different) node structures by verifying if their nodes are grouped in the ONCR.

Step 3 uses the circuit comparison to characterize the performance trade-offs of all circuits from the list. Comparing circuit $C_j \in L_k$ with the reference circuit C_k generates the trade-off profile T_j of the candidate, which illustrates how the identified differences between the two topologies impact performance.The characterization illustrates the relative performance trends w.r.t. varying design parameters and is used to rank candidate topologies in Step 4. The sorting mechanism analyzes the performance trade-off profiles T_j of circuits $C_j \in L_k$ and orders the list based on: (i) performance trade-offs eliminated, (ii) relaxed trade-offs, and (iii) range of variables over which performance improves. A topology that relaxes the gain-noise trade-off over a wider variable range is preferred for its improved flexibility.

Topology Refinement: The topology refinement procedure in Alg. 2 takes as inputs a topology to be refined C_k, a

Algorithm 1 Topology Selection Procedure

Inputs: C_k; $S_C = \{C_i, i = \overline{1,N}\}$; ONCR($S_C$); Max_{diff};
Output: $L_k = \{C_j, C_j =$ alternative topology for $C_k\}$;
(1) ***Initialize*** $L_k = S_C - \{C_k\}$;
(2) ***For all*** circuits $C_j \in L_k$
  If $|\text{ONCR}(C_j) - \text{ONCR}(C_k)| \leq \text{Max}_{\text{diff}}$ ***then***
    $L_k = L_k - \{C_j\}$;
(3) ***For all*** circuits $C_j \in L_k$
  Generate trade-off profile $T_j = \text{Compare}(C_j, C_k)$;
(4) ***Sort*** L_k based on T_j ***such that***
  Circuit C_j is before circuit C_i in L_k ***if***
    T_j relaxes and/or eliminates more trade-offs than T_i;
(5) ***Return*** L_k;

Algorithm 2 Topology Refinement Procedure

Inputs: C_k; $S_C = \{C_i, i = \overline{1,N}\}$; ONCR($S_C$);
Output: Refined topology C_k^r;
Part 1:
(1) ***Identify*** performance bottleneck of C_k w.r.t. devices M_i using trade-off profile T_k (from comparison operator);
(2) ***Build*** set of circuit nodes K from C_k to which performance bottleneck devices M_i are connected;
Part 2:
(1) ***Select*** $n_k \in K$ and ***find*** ONCR cluster $Cls_k \supset \{n_k\}$;
(2) ***For all*** circuits $C_p \in Cls_k$ ($C_p \neq C_k$, $C_p \in S_C$)
  Build set of nodes $P = \{n_p, n_p$ from $C_p\}$ ***such that***
    $n_p \neq n_k$ and input_edges(n_p)=input_edges(n_k);
(3) ***Initialize*** $C_k^r = C_k$;
(4) ***Select*** $n_p \in P$ (n_p from C_p) and ***replace*** n_k with n_p in C_k^r;
(5) ***Continue*** adding n_{p+i} from C_p to C_k^r in sequence order ***until***
  n_{p+i} from C_p matches any n_{k+j} from C_k ***or***
  n_{p+i} is output node of circuit C_p ($i, j \geq 1$);
(6) ***Generate*** trade-off profile $T_k^r = \text{Compare}(C_k^r, C_k)$;
(7) ***If*** bottleneck changed and T_k^r acceptable ***then***
    Return refined topology C_k^r;
  Else if available do new selection of $n_p \in P$ in step (3)-(4);
  Else if available do new selection of $n_k \in K$ in step (1);
  Else return *failed*;

set S_C of known designs, and this set's ONCR. The output is the topology refinement C_k^r (if available). The method consists of: (1) identifying the circuit nodes that relate to the topology's performance bottleneck, and (2) finding alternative nodal structures that remove the bottleneck. Part 1 of the procedure uses the trade-off profile T_k of circuit C_k to identify design parameters that correlate to performance bottlenecks. For example, w.r.t. device M_i sizing, gain has a logarithmic behavior that saturates at a maximum. A feasible refinement changes this bottleneck to linear or exponential behavior. The list of nodes K of C_k to which performance bottleneck devices are connected is built.

Part 2 of Alg. 2 attempts to find a feasible topology refinement for circuit C_k starting from nodes $n_k \in K$. In steps (1)-(2), ONCR is used to find alternative nodal structures for a bottleneck node n_k. First, the cluster of n_k is found in the ONCR. Then, circuit nodes n_p from other circuits C_p (different groups than n_k) are identified. These candidates are aggregated in set P. To ensure structural integrity (i.e., generate *working* topologies), only nodes n_p which have matched input edges to n_k are included in P. In steps (3)-(5) a node n_p of C_p is selected. Its structure replaces that of n_k in the refined topology, C_k^r. In Step (5), the ONCR signal path [12] of C_p is followed and additional nodal structures of this circuit (n_{p+i}) are added to C_k^r until: (i) node n_{p+i} from C_p matches any of the subsequent nodes of n_k from C_k ($n_{p+i} = n_{k+j}$), or (ii) node n_{p+i} is the output

of circuit C_p. These conditions impose that only minimum possible structural changes are introduced from C_p to C_k^r. In Step (6), the comparison operator generates the trade-offs T_k^r of the refined circuit C_k^r w.r.t. the original C_k. This characterizes the performance implications of the newly added features. Analysis of trade-offs T_k^r in step (7) determines if the current refined topology changes the performance bottlenecks while other trade-offs show acceptable trends. If the current topology refinement solution is not acceptable, the procedure first iterates through other possible candidates in set P. If unsuccessful, the method backtracks to Step (1) of Part 2 and selects a different node n_k from set K of the input topology C_k to refine. The procedure returns a failure after exhausting all refinement alternatives, suggesting topology C_k from known S_C is the best overall choice.

III. Case Study Examples

We presents case studies on using the proposed synthesis techniques. The methods use the ONCR of 50 amplifier circuits [10] together with symbolic comparison [11] to implement topology selection and refinement.

Topology Selection: The procedure in Alg. 1 is used to select a superior schematic from the ONCR for the two-stage Miller amplifier in Fig. 3 reference topology. In the 50 amplifier design set (S_C, Fig. 1), this circuit is labeled C_{11}. With $C_k = C_{11}$, topology selection is applied for $\text{Max}_{\text{diff}} = 2$ to identify alternatives that can improve performance.

Build Candidate List: In Step (2), the ONCR of known solutions is used to identify the topology candidates that satisfy the Max_{diff} condition. Fig. 1 illustrates the relevant subset of the 50 circuits ONCR. Reference C_{11} is highlighted in green while the potential selection candidates are marked in blue (C_{10}, C_{24}, C_{38}, C_{48}, C_{49}). Six clusters of the ONCR are shown corresponding to the signal path *Interm.1-Interm.5* and *Output* nodes. Arrows between clusters link the nodes of individual circuits. Red arrows indicate situations where candidate's nodes are unmatched with the reference design C_{11} (different groups). For example, in the *Interm.1* cluster all circuits match with C_{11} and form a single group. For *Interm.2*, topologies C_{49} and C_{24} utilize different structures and are in different groups. The list of candidates is $L_k = \{C_{10}, C_{38}, C_{48}, C_{49}\}$. For example, C_{48} has 2 different node structures at *Interm.3* and at *Interm.5* (without equivalent in C_{11}). Note that C_{24} is not in L_k as it consists of 4 different nodal features.

Generate Candidate Trade-offs: In Step (3), the list of candidates is compared with the reference design. The comparison operator correlates the structural differences with performance and generates the circuit's trade-off profile. The schematics of the reference design and selection candidates are shown in Fig. 3. Differences w.r.t. C_{11} are highlighted. The analysis considers the implications on performance of different devices, common devices introducing new signal paths (e.g., M_1 in C_{10}), critical circuit devices (e.g., input pair), and various combinations of these. Device sizes are varied over a predefined range and the normalized performance plots of gain,

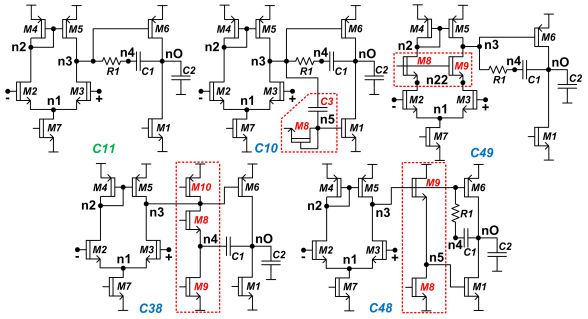

Fig. 3. Schematics of reference and topology selection candidates

CMRR, bandwidth (f_{3dB}), UGF (f_0), and total noise (P_N) are generated.

C_{38} trade-offs: Parameters used for analysis are $W_8 - W_{10}$ (and I_8), $W_2 = W_3$, and W_6. W.r.t. different devices sizing $W_8 - W_{10}$ (and I_8), W_8 and biasing I_8 have the dominant contribution. The trade-off is shown in Fig. 2 (left). Compared to C_{11}, CMRR remains unchanged and the new variables introduce a symmetric gain-bandwidth trade-off with both performances showing variations of 20%. UGF exhibits a relative maximum around the midpoint of the analyzed range. In the second half of this range it decreases nonlinearly by up to 10%. The P_N trade-off exhibits a pronounced variation of 32%. In the first half of the range, it increases sharply as gain linearly decreases. The increase in noise is also at a higher rate than that of f_{3dB}. The second half of the analyzed rage presents a better trade-off pattern as noise tends to saturate towards a maximum value while f_{3dB} continues to increase. When also considering W_6 and a decreasing current I_8, the trade-off profile w.r.t. C_{11} is changed. Shown in Fig. 2 (right), gain increases almost linearly across the range and shows a variation of 65%. In contrast, f_{3dB} decrease across the range by 60% in nonlinear fashion. For the later half of the range, gain increases faster than bandwidth deteriorates. This region is favorable for P_N as after its maximum it can be reduced by up to 10% while increasing gain. The maximum of f_0 is almost eliminated in this profile and performance has an accentuated decrease of 18%. A similar analysis was conducted for the other selection candidate circuits.

Sort Candidate List: Step (4) of topology selection sorts the list of candidates based on their trade-offs such that the topologies with the more pronounced advantages in improving performance are preferred. For our example, the ordered list is: $L_k = \{C_{10}, C_{48}, C_{38}, C_{49}\}$. Topology C_{49} is last as its analysis has shown that it does not significantly change the nature of the trade-offs w.r.t. reference C_{11}. Distinguishing between topologies C_{48} and C_{38} is done based on bandwidth and unity gain frequency behavior. Overall, topology C_{10} is preferred from the set. It presents the mechanism to eliminate the gain-noise trade-off across the analyzed parameter ranges while limiting the reduction in bandwidth.Topology C_{10} also requires one of the smallest structural changes w.r.t. reference C_{11} and does not require additional static power.

Topology Refinement: We now discuss an application of the topology refinement technique to modify the Miller two-stage topology (C_{11}) into a folded cascode schematics.

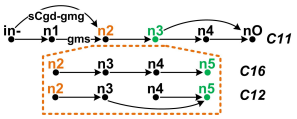

Fig. 4. Refinement options for C_{11} starting at node n_2

Identify Bottleneck: In Part 1 of the topology refinement Alg. 2, the performance bottlenecks are identified using the trade-off of the reference design. For $C_k = C_{11}$, the normalized profile shows that gain and CMRR exhibit a limiting behavior as sizes of devices $M_2 - M_3$ increase. The method identifies nodes $K = \{n_1, n_2, n_3\}$ to which devices $M_2 - M_3$ are connected, and proceeds in Part 2 to identify topology refinements. The procedure emulates designer reasoning. It precisely identifies the structure relating to performance limitations and attempts to only locally modify the topology.

Find Refinement Candidates: Consider that in Step (1) of Part 2, node $n_2 \in K$ from C_{11} is selected for refinement. This points to the *Interm.2* cluster of the 50 circuit ONCR in Fig. 1. Step (2) builds the list of refinement candidates consisting of nodes from circuits which are unmatched with C_{11} at this level in ONCR: topologies C_{16} and C_{12} are shown in orange in Fig. 1. The input edge constraint is satisfied for these two candidates: *Interm.1* cluster has a single group of matched nodes containing all circuits. Fig. 4 illustrates the matched input edge structure of node n_2 in circuits C_{11}, C_{16}, C_{12}. This requirement is important to maintain compatibility of the original and refined structures. Hence, the list of candidate nodes P includes $\{n_2(C_{16}), n_2(C_{12})\}$.

Incorporate Refinement: Steps (3)-(5) of Part 2 introduce the features of the candidates to the reference design. Fig. 4 illustrates the process for node n_2 of C_{11} using structures from C_{16} and C_{12} represented as signal flow graphs. Using C_{16}, node n_2 structure of C_{11} is first replaced by that of C_{16}. Following the signal flow in C_{16}, n_3 is reached. Its structure does not match any of nodes n_3 through n_O of the original C_{11} and is also included in the refined design. Similarly, node n_4 of C_{16} is added. At node n_5 in C_{16}, the refinement stops. The structure of this node precisely matches that of n_3 in C_{11} (both pole and edge symbolic expressions). The remaining nodes in C_{11} (n_4, n_O) are kept unchanged. The process is similar for refinement using C_{12}. Both refinement alternatives introduce the same number of additional nodes to C_{11}, but the corresponding structures are different. Fig. 5 shows the schematics of the two folded cascode alternatives with different CM loads.

Characterize Refinements: Steps (6)-(7) characterize the performance of the refined topologies using the comparison operator. W.r.t. the reference solution C_{11}, comparison identifies the additional design variables of new devices $M_8 - M_{11}$ in both alternatives from Fig. 5. Considering also the original bottleneck devices $M_2 - M_3$, the technique generates the trade-off profiles which are used to quantify the advantages of the refined topologies (profile not shown due to page limits). Both topologies include the mechanism to change the gain and CMRR bottleneck of the original design and show similar behavior for gain, CMRR, UGF, and bandwidth. Noise

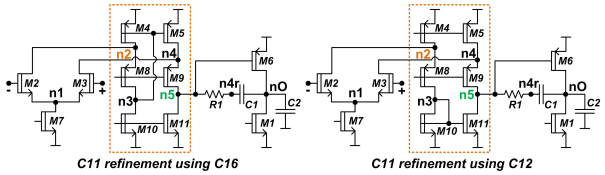

Fig. 5. Schematics of refined C_{11} topologies

behavior differentiates the two refined topologies. Overall, refined topology based on C_{16} presents greater flexibility than that of C_{12} in finding design parameter combinations for which both gain and noise performance are acceptable. "Black-box" synthesis methods have difficulties in producing such precise refinement solutions, as they lack *understanding* of the local feature's impact on performance. In contrast, the proposed method identifies the exact structures that need to be corrected and only employs new structures which can improve performance. This cause-effect mechanism is similar to designer practices of reasoning new solutions.

IV. Conclusion

A novel, reasoning-based approach to circuit topology selection and refinement was presented. The approach conducts successive steps, in which the performance trade-offs, bottlenecks, and values of a schematic guide selection and inclusion of new structural features. Features are selected from a large pool of existing designs. Every synthesis step is justified by the trade-offs and bottlenecks that are improved. This process resembles reasoning and decision making of manual design.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 0856038. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] A. Doboli and R. Vemuri, "Exploration-based high-level synthesis of linear analog systems operating at low/medium frequencies," *IEEE Trans. CADICS*, 2003.

[2] T. Sripramong and C. Toumazou, "The invention of CMOS amplifiers using genetic programming and current-flow analysis," *IEEE Trans. CADICS*, 2002.

[3] T. McConaghy and et al., "Variation-aware structural synthesis of analog circuits via hierarchical building blocks and structural homotopy," *IEEE Trans. CADICS*, 2009.

[4] A. Das and R. Vemuri, "Topology synthesis of analog circuits based on adaptively generated building blocks," in *Proc. DAC*, 2008.

[5] Y. Wei and et al., "Systematic methodology for designing reconfigurable delta sigma modulator topologies for multimode communication systems," *IEEE Trans. CADICS*, 2007.

[6] C. Ferent and et al., "Measuring the uniqueness and variety of analog circuit design features," *Integration, the VLSI Journal*, 2011.

[7] R. Harjani and et al., "A prototype framework for knowledge-based analog circuit synthesis," in *Proc. DAC*, 1987.

[8] M. Degrauwe and et al., "IDAC: An interactive design tool for analog cmos circuits," *IEEE Journal of SSC*, 1987.

[9] I. O'Connor and A. Kaiser, "Automated design of switched-current cells," in *Proc. CICC*, 1998.

[10] C. Ferent and A. Doboli, "Analog circuit design space description based on ordered clustering of feature uniqueness and similarity," *Integration, the VLSI Journal*, 2013.

[11] ——, "Symbolic matching and constraint generation for systematic comparison of analog circuits," *IEEE Trans. CADICS*, 2013.

[12] X. Huang and et al., "Modeling nonlinear dynamics in analog circuits via root localization," *IEEE Trans. CADICS*, 2003.