Stony Brook University

Analog Circuit Design Knowledge Mining and Circuit Causal Information Modeling

A Dissertation presented

by

Fanshu Jiao

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

December 2016

Stony Brook University

The Graduate School

Fanshu Jiao

We, the dissertation committe for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

Alex Doboli - Dissertation Advisor
Professor, Department of Electrical and Computer Engineering

Milutin Stanacevic - Chairperson of Defense
Associate Professor, Department of Electrical and Computer Engineering

Emre Salman
Associate Professor, Department of Electrical and Computer Engineering

Peter Milder
Assistant Professor, Department of Electrical and Computer Engineering

Xin Li - Associate Professor
Department of Electrical and Computer Engineering, Carnegie Mellon University

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Analog Circuit Design Knowledge Mining and Circuit Causal Information Modeling

by

Fanshu Jiao

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

2016

This thesis proposes novel analog circuit design knowledge mining and circuit causal information modeling approaches. Existing circuit synthesis in terms of topology synthesis and circuit sizing are mainly equation-based and optimization-based. Topology synthesis selects, reuses existing topology, or combines basic building blocks, which can hardly design complex or innovative structures. Circuit sizing is guided by goal-related optimization, whereas there is still gap between the way sizing tool works and how a designer reasons to size a circuit. The thesis aims at incorporating designers' reasoning in synthesis process, by mining analog circuit design knowledge and causal information, to narrow the gap between design automation and human-devised circuits. We propose design knowledge can be mined from three aspects, topology, achievable performance and design reasoning flow. They are defined by three components in the thesis, associative component, performance capabilities component, and causal reasoning component.

The associative component groups a set of circuit topologies, i.e., instances, into a hierarchy of abstract topologies, i.e., abstraction. An abstraction describes the similarities among its instances as well as the alternative topological features, i.e., nodes, edges, and electrical behavior of the instances, represented by symbolic expressions/models. We transfer topology into signal flow graph by symbolic modeling, extract topological features and construct abstraction from common topological features. The performance capabilities component describes an instance's or an abstraction's performance trade-offs and bottlenecks relevant to the performance requirements. The performance trade-offs, as well as parameter effects obtained from local sensitivity analysis are built into trade-off tables. Combining trade-off tables of the instances generates trade-off tables of the abstraction. The causal reasoning component refers to designer's reasoning flow. We summarize from design literature that a typical design flow involves possible topological features that served as starting ideas in devising new circuit, followed by design steps and their justifications.

In the experimental section, design knowledge was mined from analog circuit design literature and a complete knowledge representation was built on 34 modern op-amps/OTAs. It is concluded that the op-amp/OTA topologies are built into a four-level hierarchy based on common and different topological features, including input/output stage implementation, gain stage implementation, and alternative building blocks to achieve the desired performance requirements.

Mined design knowledge, in particular the causal reasoning component, is further applied to topology synthesis and design verification fields. Reasoning-based topology synthesis method involves starting ideas selection and design sequence generation. The thesis proposes five topology synthesis strategies based on different ways to derive starting ideas, and creates four high performance op-amps/OTAs. Reasoning-based design verification method verifies circuit design plan from the mined causal reasoning component. Topological

features corresponding to the starting ideas and design sequence are verified individually by replacing the related devices with ideal behavior model. Evaluating simulation results of the newly generated circuits and original circuit reveals incorrect functional issues and/or improvement potentials, which are negative causes of certain starting ideas or design step.

Expanding from parameter local sensitivity in trade-off table and individual performance potentials, the thesis develops causal information modeling to extract circuit parameter-performance, parameter-parameter effect at a global scale. Causal information embedded in analog circuits represents relations according to which a set of design parameters decide the values of the functional outputs, performance, and other parameters (parameters' causality). An ordered parameter sequence implies designers' sizing strategy in which each step utilizes causal information to traverse the solution space.

The modeling approach extracts causal information by six elements: (*i*) Causal relation characterizes the relation between a design parameter, performance attributes, and the other parameters. It expresses the parameter's influence in controlling performance values and trade-offs. (*ii*) Causal trace presents the comprehensive causal influence of a parameter for performance attributes with different importance (weights). (*iii*) Causal graph includes the causal traces of all circuit parameters, all performance attribute sets. (*iv*) Sequenced causal relations, and (*v*) graph of sequenced causal relations order parameters' causal relations based on different measures. Ordered causal relations reveal parameters' importance on determining output performance and circuit sizing strategies. Finally, (*vi*) causal Pareto front is formed to express high dimensional trade-off from ordered causal relations. Six causal elements of an op-amp circuit are extracted in the experiment, showing circuit parameter causal effects on performance attributes {Gain, Bandwidth, THD} and multi-dimensional trade-off.

Contents

List of Tables

List of Figures

Acknowledgements

The five years PhD study at Stony Brook University has been the most amazing and treasured time in my life. I would like to extend my sincere gratitude for everyone who made it memorable.

First of all, I would like to express my special appreciation and thanks to my advisor Dr. Alex Doboli, who has been a tremendous mentor for me. I would like to thank him for encouraging my research and for allowing me to grow as an independent researcher. His advice on both research as well as on my PhD study has been priceless. I would never have completed the PhD program without his knowledge, guidance, encouragement and inspiration. I would also like to thank my committee members, Dr. Milutin Stanacevic, Dr. Emre Salman, Dr. Peter Milder, and Dr. Xin Li for their unconditional support. I would like to thank them for letting both my prelim presentation and PhD defense be an enjoyable moment and for them brilliant comments and suggestions.

Also, thanks to my internship supervisor Dr. David Smart for his guidance when I was a CAD engineering intern at Analog Devices. He has been a great supervisor and mentor to me, giving me inspiration on both research and professional career.

I also want to thank my colleagues Cristian Ferent, Anurag Umbarkar, Yiqun Ji in VLSI Systems Design Laboratory for their friendship and all of their help over years. I would like to express my gratitude to Cristian Ferent, for his research work on "Systematic Modeling and Characterization

of Analog Circuits using Symbolic and Data Mining Techniques". As prior work to design knowledge mining, it has paved the way and offered me a lot of inspiration and enlightenment. I am also grateful to Yiqun Ji's tool on symbolic model generation and comparison, which has been helpful for my experiments on circuit signal flow graph representation. I would also like to thank the many individuals in the Department of Electrical and Computer Engineering who have made my stay at Stony Brook University a pleasant and memorable one.

I am grateful to all my friends from University of Science and Technology of China, for being supportive during the many years I stayed there and for their continued support there after. I would like to express my special thanks to my close friends Zhen He, Na Xiong, and high school mate Mengdan Xu, who always stand by my side and share my happiness and sadness for so many years.

A special thanks to my parents Shiwei Jiao and Lianrong Wang for their unyielding love, endless patience and encouragement. Words cannot express how grateful I am to my parents. I could not complete my PhD without their continued support and vote of confidence.

Finally, I acknowledge the research funding I received over the years. The materials presented in this dissertation is based upon work supported by the National Science Foundation under Grant BCS 1247971. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Chapter 1

Introduction

1.1 Thesis outline

Figure 1.1 shows the thesis methodology flow and organization. The thesis has the following structure. Chapter 2 presents analog circuit design knowledge mining, including algorithms to mine associative component, performance capabilities component, causal reasoning component. Chapter 3, 5 apply mined causal reasoning to topology synthesis, design verification and develop reasoning-based topology synthesis, reasoning-based design verification methods. Extending from the trade-off table method in Chapter 2, Chapter 4 characterizes circuit trade-off effects by modeling causal information and constructing causal Pareto front. It explains various circuit causal elements, and proposes approach to construct causal Pareto front in the end.

1.2 Motivation

The main tasks of analog circuit design automation range from simulation and modeling, synthesis and optimization, layout synthesis, to design for testability, etc [1]. Except simulation, which has been demonstrated successful in commercial tools, the other fields still need development for

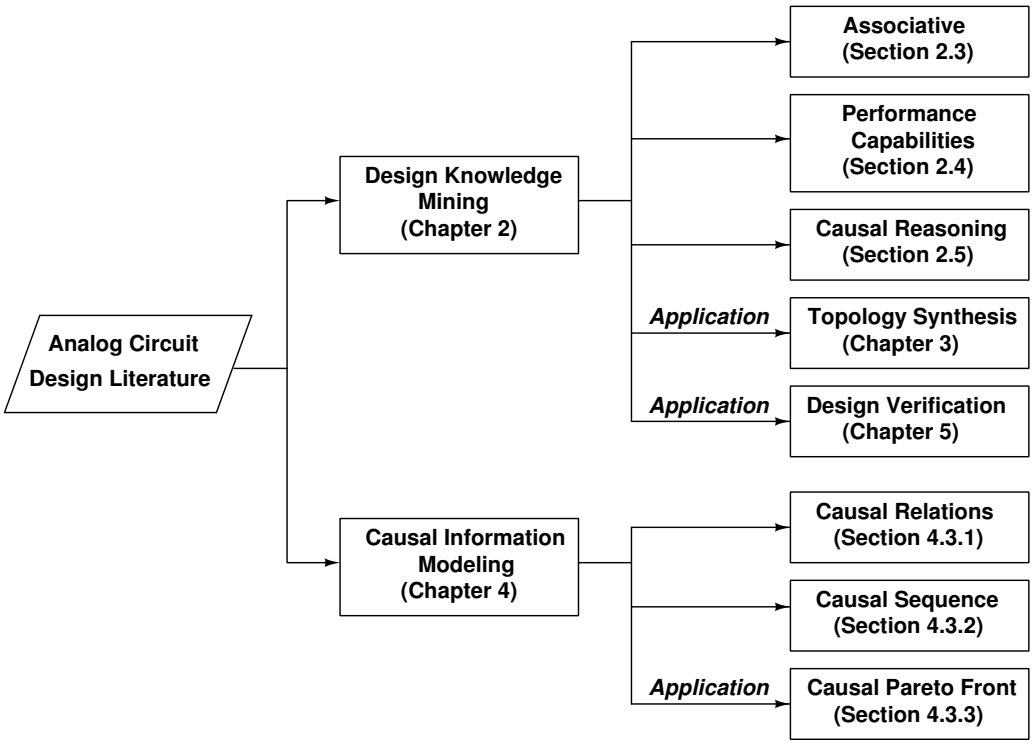


Figure 1.1: Methodology flow and thesis organization

automation. The main challenges lie in the nature that analog circuit exhibits continuous time, nonlinear behavior with strongly coupled parameters. Thus, analog design is knowledge intensive and heuristic, which heavily relies on designers' experience and expertise in years. Topology design nowadays is mainly tackled as designer's manual process. Designers usually come up with solutions based on manual reasoning to combine basic devices, sub-circuits, and ideas from similar solutions [2]. These combinations may include various circuit topological features, e.g., differential input, cascode, buffer, current source, etc, as well as design constraints among circuit parameters, e.g., transistor operation regions, device matching, technology constraint, etc. As new applications emerge and new manufacturing processes being devised, innovative design solutions, topological features, and analysis methods have been

constantly presented in the related literature on analog circuit design. Often, it is hard for designers to manually review and track the most relevant ideas and solutions. It is also difficult to summarize and organize the entire published literature, so that, while devising a novel solution, all related ideas and circuit designs are effectively identified, analyzed, and utilized as starting points for new design solutions. It is likely that superior designs are ignored or previous ideas are re-discovered, which may result in less effective design solution.

In design automation field, methods to automate topology generation and circuit sizing are mainly equation-based and optimization-based [1]. Topology generation method includes topology selection, re-use, or combining basic building blocks. They all incorporate and utilize certain simple design knowledge as input, e.g., simple building blocks, transistor operation regions, transistor sizing constraints, to some extend in order to reduce the gap between their capabilities and manual design [3], [4]. Circuit sizing is guided by goal-related optimization. Whereas, existing equation or optimization-based methods fall behind manual design in terms of two aspects.

(i) *Circuit sizing.* Even though optimization-based sizing has been available in commercial tools, there is still gap between the way synthesis tool works and how a designer reasons to size a circuit. The following example illustrates parameter sequence encoded in sizing tool, which is not able to sufficiently utilize or match designer's sizing process guided by parameter trade-offs.

To derive the sizing sequence utilized in sizing tool, we conducted a case study on op-amp circuit in Figure 1.3. We first obtain 64 design points from Cadence optimization tool [5], target at optimizing Gain. Based on the 64 design points, correlation coefficients are computed under two scenarios, correlation between Gain and parameter p_i modeled as $\mathrm{M}_{Gain}(\mathrm{p}_i)$, correlation between different parameters modeled as $\mathrm{C}_{p_i}(\mathrm{p}_j)$. The most correlated parameter, g_{ds3}, is first selected as important parameter. Taking g_{ds3} as a

Table 1.1: Sizing sequence derived from correlation coefficients

$M_{Gain}(g_{ds3})$	$M_{Gain}(g_{m3})$	$M_{Gain}(g_{ds13})$	$M_{Gain}(g_{m13})$	$M_{Gain}(g_{ds10})$	$M_{Gain}(g_{m10})$	
-0.76	-0.85	-0.8	-0.71	-0.71	-0.72	
Parameter sequence $M_3 \rightarrow M_{13} \rightarrow M_{10} \rightarrow M_0 \rightarrow M_4 \rightarrow M_7 \rightarrow M_1$						
$C_{g_{ds3}}(g_{ds13})$	$C_{g_{ds13}}(g_{ds10})$	$C_{g_{ds10}}(g_{ds0})$	$C_{g_{ds0}}(g_{ds4})$	$C_{g_{ds4}}(g_{ds7})$	$C_{g_{ds7}}(g_{ds1})$	$C_{g_{ds1}}(g_{ds6})$
0.88	0.73	0.73	0.75	0.9	0.75	-0.25
Parameter sequence $M_3 \rightarrow M_{13} \rightarrow M_7 \rightarrow M_4 \rightarrow M_0 \rightarrow M_{10}$						
$C_{g_{ds3}}(g_{ds13})$	$C_{g_{ds13}}(g_{ds7})$	$C_{g_{ds7}}(g_{ds4})$	$C_{g_{ds4}}(g_{ds0})$	$C_{g_{ds0}}(g_{ds10})$	$C_{g_{ds10}}(g_{ds6})$	
0.88	0.75	0.92	0.75	0.73	-0.21	

Table 1.2: Sizing sequence by modeling a subset of data points

$M_{Gain}(g_{ds13})$	$M_{Gain}(g_{m13})$	$M_{Gain}(g_{ds10})$	$M_{Gain}(g_{m10})$		
-0.77	-0.45	-0.66	-0.52		
Parameter sequence $M_{13} \rightarrow M_{10} \rightarrow M_0 \rightarrow M_6 \rightarrow M_{11}$					
$C_{g_{ds13}}(g_{ds10})$	$C_{g_{ds10}}(g_{ds0})$	$C_{g_{ds0}}(g_{ds6})$	$C_{g_{ds6}}(g_{ds11})$	$C_{g_{ds11}}(g_{ds3})$	
0.6	0.87	0.76	0.94	0.45	
Parameter sequence $M_{13} \rightarrow M_{10} \rightarrow M_3 \rightarrow M_6 \rightarrow M_{11} \rightarrow M_0$					
$C_{g_{ds13}}(g_{ds10})$	$C_{g_{ds10}}(g_{ds3})$	$C_{g_{ds3}}(g_{ds6})$	$C_{g_{ds6}}(g_{ds11})$	$C_{g_{ds11}}(g_{ds0})$	$C_{g_{ds0}}(g_{ds4})$
0.6	0.83	0.53	0.94	0.7	0.25

starting pointing in sizing sequence, we check models $C_{g_{ds3}}(p_j)$. Among all other parameters correlated with g_{ds3}, model $C_{g_{ds3}}(g_{ds13})$ characterizes g_{ds13} is most correlated to g_{ds3}. Thus, an order to size the circuit starts from g_{ds3} (device M_3) to g_{ds13} (device M_{13}). Similarly, starting from g_{ds13}, model $C_{g_{ds13}}(g_{ds10})$ shows g_{ds13} is most correlated to g_{ds10}. An order of highly correlated parameters $\{g_{ds3} \rightarrow g_{ds13} \rightarrow g_{ds10} \rightarrow g_{ds0} \rightarrow g_{ds4} \rightarrow g_{ds7} \rightarrow g_{ds1}\}$ reveals how the synthesis tool works on the transistors to achieve a sizing solution, $\{M_3 \rightarrow M_{13} \rightarrow M_{10} \rightarrow M_0 \rightarrow M_4 \rightarrow M_7 \rightarrow M_1\}$. Another possible sequence is $\{M_3 \rightarrow M_{13} \rightarrow M_7 \rightarrow M_4 \rightarrow M_0 \rightarrow M_{10} \rightarrow M_1\}$. The sequence ends at uncorrelated device M_6. Table 1.1, 1.2 summarize important parameters (devices) utilized in automatic sizing, and parameter (device) sequences encoded in sizing solutions.

From the result, we conclude synthesis tool starts sizing from the output stage and only considers signal path with one input transistor M_7, which is

not able to match the complete design plan proposed by a designer, $\{M_6 \rightarrow M_3 \rightarrow M_0 \rightarrow M_{10} \rightarrow M_{13} \rightarrow M_7 \rightarrow M_{10}\}$. Modeling a subset of 64 design points shows different sequences $\{M_{13} \rightarrow M_{10} \rightarrow M_0 \rightarrow M_6 \rightarrow M_{11}\}$, or $\{M_{13} \rightarrow M_{10} \rightarrow M_3 \rightarrow M_6 \rightarrow M_{11}\}$. Chapter 4 will explain other measures to model parameter-performance trade-off besides correlation coefficient, and strategies to derive sizing sequence (defined as causal information modeling).

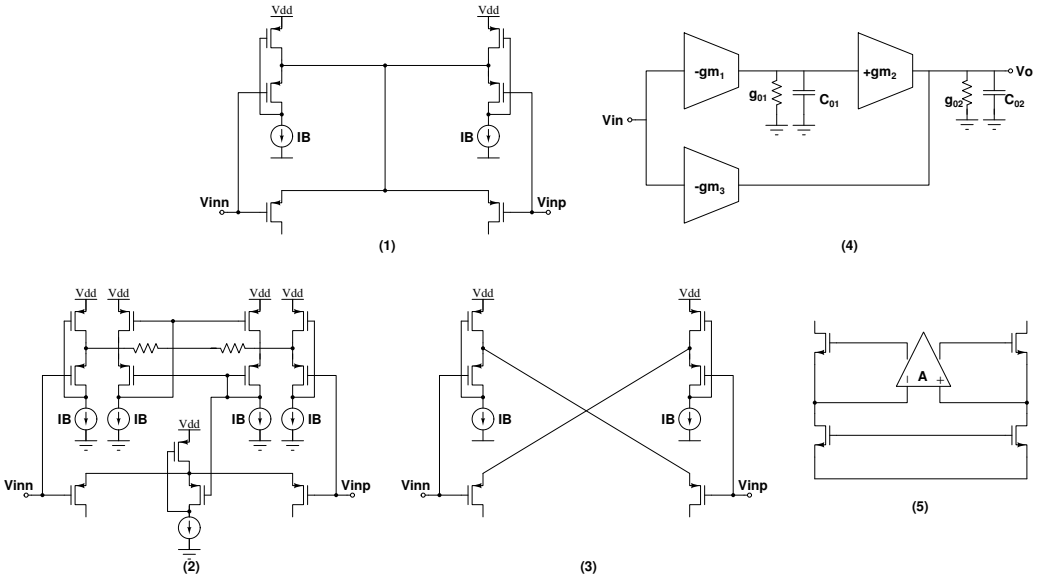


Figure 1.2: Complex structures or topological patterns

(*ii*) *Topology generation.* Current design automation methods tend to immediately re-use, refine recent circuit topology solutions, or create topology from basic building blocks [6]. They are not able to effectively utilize the innovative topology from literature, which is the key solution for advanced specifications. For example, designer's solution introduces different adaptive biasing topologies for low-voltage application (1∼3), feed-forward compensation (4), gain-boosting strategy (5) in Figure 1.2. Class AB op-amp with the adaptive biasing structure achieves near-optimal current efficiency in [7]. Re-use or refine topology [8] relies on pre-defined topology library, which is hard to keep updated with innovative topologies from literature. Topology

creation [9] is limited to basic building blocks and lacks of designer's reasoning like abstraction, instantiation, analogies, induction, concept combination in complex design strategies. It can hardly incorporate more complex structures or topological patterns like in Figure 1.2.

Existing synthesis methods lack the capabilities to efficiently and comprehensively represent analog circuit knowledge and reasoning space, which are essential elements in human design innovation. Designer's reasoning might involve optimization and equation solving, it also includes pattern identification both with respect to the utilized topological features, their justifications, and parameters' causality defining main influence in setting the performance values. The thesis proposes systematic approaches to mine analog circuit design knowledge and model causal information from literature. They extend current optimization- and equation-based synthesis methods by incorporating activities inspired by human reasoning utilizing knowledge mining techniques, to leverage the effectiveness and capabilities of automated tools by narrowing the gap between their solutions and human-devised circuits. The methods are motivated by explored techniques developed in machine learning [10], statistical data analysis [11], data mining [12], and the emerging shift to knowledge mining [13] to extract useful information from a variety of data [14], [15].

With the availability of big data technique nowadays, data mining algorithms analyze data from different perspectives and summarize it into useful information. Extending from the traditional tasks of finding useful patterns in data, knowledge mining targets new challenges in order to reuse the data patterns, to discover useful knowledge from data and to integrate knowledge base. It evolves various techniques to derive high-level concepts and descriptions from data involving both data and relevant background knowledge. Systematic methods utilize advanced knowledge representation and generate different types of knowledge from a given data source. Explored applications range from computational intelligence [16], text mining [17], and power

system [18], etc. Despite the fact that analog circuit design is knowledge intensive, there is no research work on knowledge mining in analog circuit design automation.

The proposed knowledge mining on analog circuit design includes knowledge discovery and organization. For modern op-amp/OTA circuits, design knowledge is mined from three aspects, circuit topological features, performance capabilities and design causal reasoning, which arguably cover the entire design knowledge space. Circuit topologies extracted from literature are first transferred into symbolic models, from which common topological patterns are identified. Performance capabilities express circuit's parameter sensitivity, performance trade-off and bottleneck. Design causal reasoning summarizes manual design procedures and how designer evolves innovative solutions. The thesis also presents method to model causal information encoded in the circuit, expressed by various relations of parameters with performance attributes and other parameters, revealing different sizing strategies designers reason when size circuit manually.

1.3 Goals and contributions

The goal of the thesis is to reduce the gap between current analog synthesis result and designer's solution by mining design knowledge and causal information, as well as applying mined knowledge to topology synthesis, design verification. It is achieved through (*i*) mining techniques to build topology hierarchical representation, performance trade-off tables and design causal reasoning flows, (*ii*) applying mined causal reasoning to topology synthesis and design verification, (*iii*) modeling circuit parameter causal relations and constructing causal Pareto front. The novel contribution of the thesis includes:

- We propose the first systematic knowledge mining technique on analog circuit design, to efficiently and comprehensively represent design

knowledge space, and keep updated with innovative topology solutions.

- We propose original algorithms of reasoning-based topology synthesis and design verification, to capture and mimic human design activity.
- We propose original algorithm of modeling parameter trade-off effect (causal information) encoded in circuit, to obtain different circuit sizing strategies.

1.4 Methodology Overview

This section gives an overview of the proposed knowledge mining and causal modeling methodologies, and conceptually explains how they can be applied for topology synthesis and design verification. Knowledge mining algorithm builds design knowledge presentation in terms of circuit topologies, performance capabilities, and causal reasoning, to comprehensively cover the entire design knowledge space.

(*i*) The algorithm first extracts circuit topologies from electronic documents. Transistors are transferred into linear symbolic models which express linear AC performance by parameters $\{g_{ms}, g_{ds}, C_{gd}, C_{db}\}$. From symbolic models of circuits (instances) group, an abstraction is an abstract symbolic model, which describes the similarities among its instances as well as the symbolic expressions that cumulatively express the alternative topological features (i.e., nodes and edges) and electrical behavior of the instances. For each circuit instance, we also identify possible topological features from design documents and build associative component based on common topological features.

(*ii*) Circuit's (or an abstract circuit's) performance trade-offs and bottlenecks relevant to the performance requirements are summarized as performance capabilities component. Performance trade-offs are built into tables in which columns represent performance attributes and rows indicate how cir-

cuit parameters control performance attributes around certain design point. Trade-off also reveals parameter local sensitivity, which is computed symbolically into trade-off table. Circuits' trade-off tables are combined to generate performance trade-offs for abstraction by combining common effects in the trade-off tables. Combined tables indicate the common attribute variations of the circuits and the distinct attribute variations of the alternative designs.

(*iii*) While reading design literature on op-amps/OTAs, we summarize possible design flows involve starting ideas followed by the sequence of steps utilizing innovative topological features. Causal reasoning component composes possible topological features that served as starting points in devising new circuits as well as the design steps and their justifications in creating the design. Topological features of starting ideas are characterized into different sets based on their origins, i.e., previous idea from the same group, previous idea from other groups, new design insight, or common design knowledge. The thesis summarizes five patterns the way starting ideas are originated and combined to derive design flows, i.e., combining different topological features, combining topological and abstract features, combining different abstract features, not using certain topological features, creating new abstraction.

For example, Figure 1.3 shows a high performance operational transconductance amplifier (OTA) schematic [2]. Knowledge mining on the circuit results three components presented in Figure 1.4(a), (b), (c). Circuit topology is first transferred to signal flow graph, in which nodes correspond to the circuit nodes and arcs describe node coupling from transistor symbolic models (Figure 1.4(a)). Besides symbolic model, circuit topological features are extracted based on devices, transistor types, connection patterns, IO terminals. For example, subset {Vinp, Vinn, V3, V5, V6, V4, V0, V1} in Figure 1.4(a) refers to devices {M_6, M_7, M_8, M_9} and their corresponding connections, which is complimentary differential input feature. Complete feature set includes {complementary differential input, folded cascode, differential output,

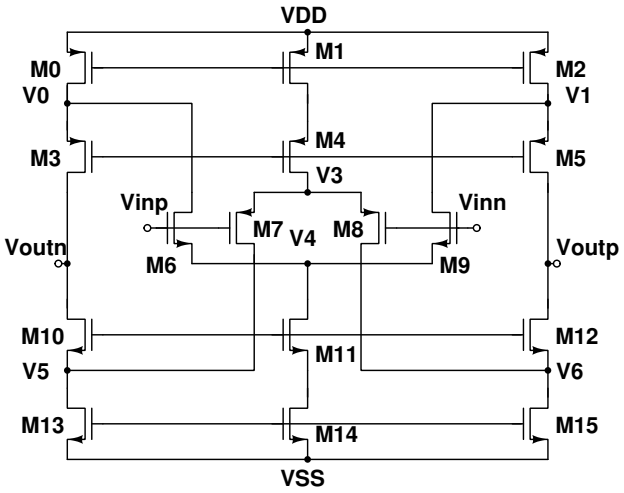


Figure 1.3: OTA schematic

basic current source, fully symmetric}.

Performance trade-off table in Figure 1.4(b) expresses parameters sensitivity effect on different linear performance ({Gain, Noise, Pole, Gain-Pole Product}). Table entries of up/down arrows are computed based on symbolic models that increment of $\{g_{md3}, g_{md10}\}$ decreases Gain, whereas increases dominant pole frequency (Bandwidth). We further utilize ellipse function to fit design points for a comprehensive trade-off presentation. Different measures to characterize ellipse models, i.e., correlation, linearity, and similarity, result ordered parameter sequence and causal information that encoded in circuit sizing.

Figure 1.4(c) summarizes designer's steps of creating the circuit which starts from using fully differential structure for modulator application. Starting ideas also include complementary folded cascode as for high speed specification [2]. Following steps add common mode feedback and biasing circuitry for correct operation. Knowledge mining on 34 op-amp/OTA circuits extracts and abstracts individual circuit's topology, trade-off table, causal reasoning. Abstraction further reveals common topological features and their corresponding trade-offs. Circuits' causal reasoning indicates types of strate-

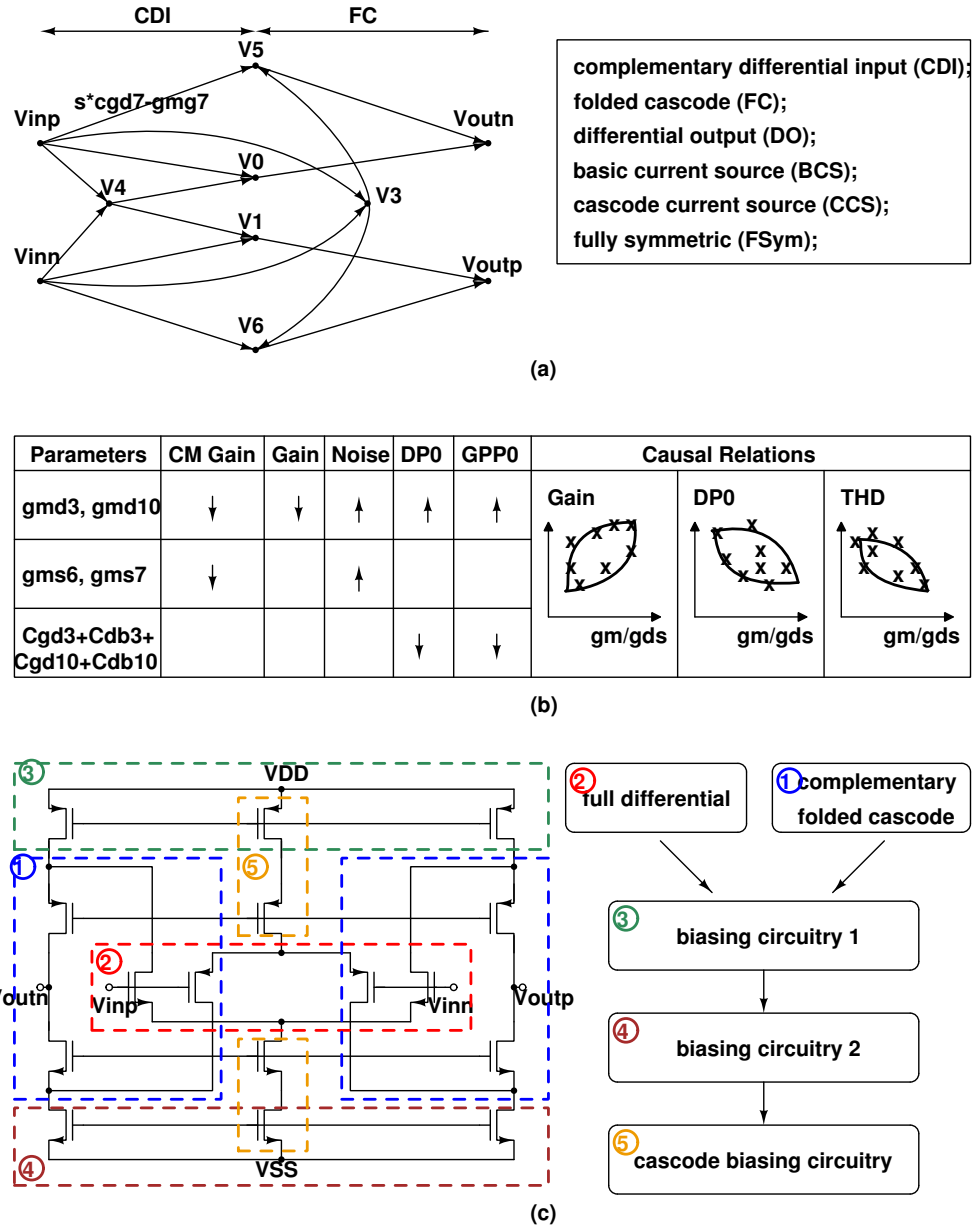


Figure 1.4: Design knowledge mining on OTA Figure 1.3

gies to start a design and combine starting ideas to derive novel solution. Experiment was conducted on 34 op-amps/OTAs from modern design litera-

ture, in tsmc 0.2μm process. Mined topological features offer novel synthesis solutions, and causal reasoning helps understand human design innovation.

Topology synthesis and design verification through a reasoning-based flow are explored as applications of the proposed knowledge mining techniques. Mined design causal reasoning flow indicates human reasoning usually begins with a set of starting ideas and then continues with a sequence of design steps to complete the solution. Starting ideas can be topological/abstract features, which correspond to the main conceptual ideas utilized in creating a new topology. Each step of the design sequence is justified by the fact that it either introduces a new topological feature that further improves performance or relaxes the constraints of the design. Experiments present four synthesis case studies using different ways to obtain starting ideas.

Reasoning-based design verification checks the correctness of the starting ideas and design steps that form a design flow. The proposed method first generates new circuits by replacing building blocks individually with ideal behavior models. Comparing simulation results of the newly evaluated circuit, incorrect functional issues and performance potentials for improvement are identified and linked to building blocks from either starting idea and/or design step. Thus, reasoning-based verification helps offer design diagnostic insight, like the starting idea and/or design step implementation that should be modified to correct functional errors. Experiments illustrate three op-amp examples to verify and diagnose design plan.

Knowledge mining in Figure 1.4 proposes trade-off tables showing circuit parameter sensitivity in a local range as parameter variation is small enough to stay around one design point (sizing solution). When size a new circuit, multiple design points at the global scale should be considered. The thesis extends trade-off table method and models circuit causal information to explore multi-dimensional performance space. Causal information presents circuit parameter-performance effect, i.e., how design parameters deciding the values of the functional outputs, performance, and other parameters,

as causal relations. Multiple strategies to characterize and order causal relations based on correlation, linearity, similarity measure are proposed, offering alternative sizing strategies to transverse the performance space.

The thesis proposes six elements to characterize causal information. (i) Causal relation characterizes the connection between a design parameter, performance attributes, and the circuit parameters. It expresses the parameter's influence in controlling performance values and trade-offs. (ii) Causal trace presents the comprehensive causal influence of a parameter for different importance (weights) of the performance attributes. (iii) Causal graph includes the traces of all circuit parameters, all performance attribute sets. (iv) Sequenced causal relations, (v) Graph of sequenced causal relations present the order in which the causal relations of the parameters are used in tackling a design problem. Ordered causal relations reveal parameters' importance on output performance and are considered as circuit design methodology (sizing strategy). Finally, (vi) causal Pareto front is formed from different sequenced causal relations. Causal relations are modeled by ellipse fitting, thus the ellipse shape describes the degree of causal control a parameter has over performance. For certain specification (i.e., performance attribute set), ellipse model and Pareto front are constructed based on sampled design points obtained from Cadence sizing tool [5].

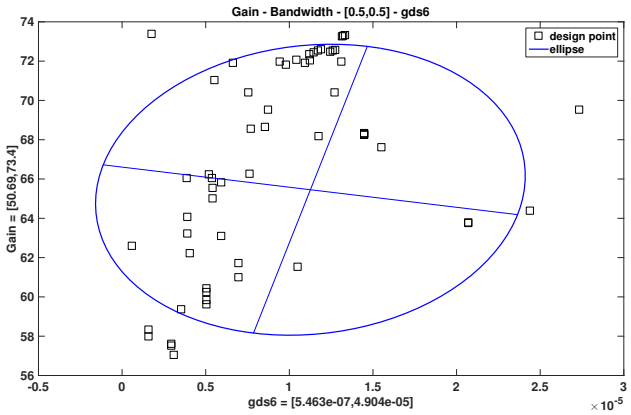


Figure 1.5: Causal relation by ellipse fitting

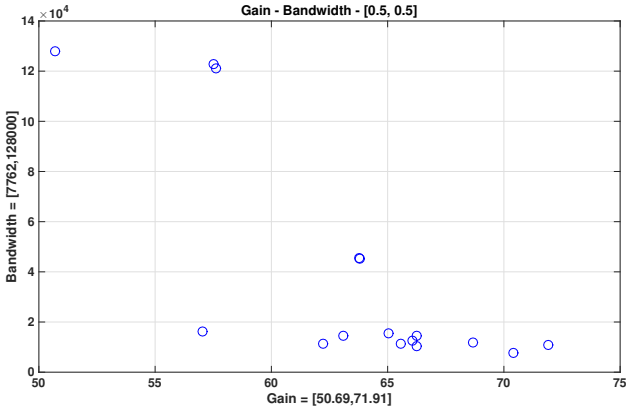


Figure 1.6: Two dimensional Pareto front

Figure 1.5 shows an illustrative ellipse model example for OTA in Figure 1.3. Experiment runs sizing tool to sample each transistor on the signal path, {M_3, M_6, M_7, M_{10}}, 16 times {15μm, 30μm,..., 240μm} evenly along channel width. Performance specification is to optimize {Gain, Bandwidth} with equal weights {0.5, 0.5}. A total of 64 optimal design points, including parameters and performance results in Figure 1.5 show parameter's causal effect on performance, i.e., g_{ds6} *vs* Gain. Figure 1.6 illustrates a two dimension Pareto front, i.e., Gain *vs* Bandwidth with negative trade-off effect.

Besides weight {0.5, 0.5}, experiment also considered a sampling of 8 weight sets to comprehensively present the design space, {[1, 0], [0.875, 0.125], [0.75, 0.25], [0.625, 0.375], [0.375, 0.625], [0.75, 0.25], [0.125, 0.875], [0, 1]}, i.e., [1, 0] = {weight_{Gain}=1, $\text{weight}_{Bandwidth}$=0}. A list of ellipse models for 9 weights composes causal trace. Instead of simulating all combinations of high dimensional performance space, causal traces of two dimensional performance are merged. In order to have a comprehensive understanding of parameter trade-offs, for each performance attribute set, various sensitivity measures are computed to order causal trace in terms of linearity, correlation, similarity, etc. For example, ordered parameter list based on linearity, {$g_{ds3} \rightarrow g_{m6} \rightarrow g_{ds6} \rightarrow g_{m3} \rightarrow g_{ds10} \rightarrow g_{m10} \rightarrow g_{m7} \rightarrow g_{ds7}$}, reveals g_{ds3} the most, g_{ds7} the least linear dependency on {Gain, Bandwidth}.

Chapter 2

Analog Circuit Design Knowledge Mining[1]

2.1 Introduction

Automatically mining design knowledge and building knowledge representation from literature (electronic documents) offer an efficient way to systematically organize, re-use, disseminate, and expand state-of-the-art design knowledge in human design activity. In engineering field, knowledge discovery through reading and mining information from electronic documents inspires a lot of problem solving and innovation. Knowledge discovery is important not only in devising new design solutions, but also in identifying new business opportunities for emerging applications. As engineering innovations are constantly being proposed and new challenges being raised, it is unlikely to continuously and comprehensively access and review the entire technical literature to find the most relevant knowledge and solution.

More specifically, in electronic circuit design, it is hard to constantly track innovative designs from design literature for new application and new technology constraints. In applications like telecommunication, mobile ap-

[1] [19], [20]

plication, and health care, problem specifications have continuously changed with respect to the required size, speed, bandwidth, linearity, power consumption, and robustness of the circuits. Therefore, innovative design ideas, solutions, topological features, and analysis methods have been constantly presented in the related literature to address current challenges and opportunities. Existing analog design automation methods tend to reuse previous work or refine recent design topologies presented in the literature, lacking in capabilities to automatically mine and organize related knowledge from the entire published literature. It is likely that while devising a novel solution, superior related ideas and circuit designs are ignored or previous ideas are re-discovered. This limits the automation's possibilities and capabilities to derive more innovative and effective design solution, hence lower design quality, productivity, and market impact.

This chapter presents knowledge mining methodology to help innovate design solutions, tackle new design opportunities and applications [20]. As analog circuit design involves related design knowledge and iterative reasoning process, an important step in knowledge mining is not only mining and representing the topological features and performance capabilities of new circuits, but also discovering the most likely reasoning steps that produced a design. By reading and summarizing manual reasoning flow written in documents, we propose it includes a set of starting ideas which trigger the solution, design sequence to complete the solution, and their justifications. The reasoning steps can be reused or extended to tackle new problems by synthesizing new circuit topologies, or might expose design steps that can be improved to create more efficient solutions. Also, discovering the most likely ideas (topological features) that served as design starting points are utilized to understand how ideas evolve and propagate within a design community. This can be used to uncover unexplored design opportunities and new market niches.

Figure 2.1 offers an overview of the proposed methodology flow to mine

design knowledge from analog circuits. Inputs are circuit topologies, which are first converted into their signal-flow graph (SFG) representation using symbolic modeling method [21]. Comparing signal-flow graphs and identifying topological features reveal topological similarities and differences among the circuits. Similar and different topological features are summarized into sets and further utilized to produce hierarchy. This step generates the associative component of the knowledge representation. Sensitivity analysis on each circuit reveals parameter's trade-off effect on performance, which is built into tables. Trade-off tables and individual performance bottleneck compose performance capabilities component. The following step produces trade-off tables of the abstraction by combining trade-off tables of the instances under the abstraction. Finally, the causal reasoning component is mined by using the circuit schematics, abstractions and trade-offs to identify the starting ideas and design sequence that were likely used to create the design. The three main components of design knowledge representation, {associative, performance capabilities, causal reasoning}, will be explained in detail.

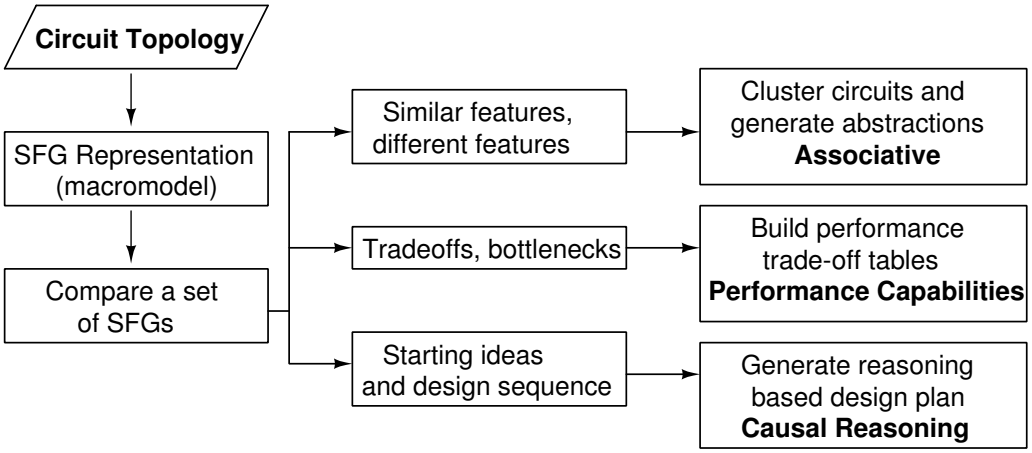


Figure 2.1: Methodology flow to mine analog circuit design knowledge

The chapter is organized as follows. Section 2.2 summarizes related work

of mining techniques on circuit modeling, mining system causality in fields other than electronic design automation. Section 2.3, 2.4, 2.5 present the main theory of analog circuit design knowledge mining: the knowledge representation components. Section 2.6.1, 2.6.2, 2.6.3 propose the algorithms to mine the three components accordingly. Finally, conclusions are offered.

2.2 Related work

There are various ways of applying modeling/mining techniques on electronic circuit design, including identification of performance modeling, circuit classification and clustering, etc.

On the digital side, for the high degree of regularity in digital circuit, structural regularity extraction has been studied [22], [23]. Analog circuits are different as structural regularity is weaker and performance specifications are more complex. There are performance modeling techniques by nonlinear regression models [24], neural-network models [25], genetic programming [26]. The circuit modeling procedure in [27] extracts piecewise linear models from trained neural networks, to represent the linear dependencies between circuit performances and design parameters. [28] presents an exploration procedure for mapping given functional specifications of an analog system to the specification parameters of individual component blocks of the system topology. [29] proposes Bayesian model to present process variation space.

Several data mining algorithms have been explored to model design space, including collecting Internet available performance data to build trade-off model [30]. In terms of classification and clustering, in [31], support vector machines are used as classifier to identify the feasible design space of analog circuits. As an integral part of the SVM, some new kernels and some other kernels composed through modification on the some of the standard kernels, are explored. More uses of SVMs on performance macromodeling and feasibility modeling are least-squares support vector machine [32], [33]. [34] takes a

mining perspective on a Pareto optimal set of sized analog circuit topologies. It extracts a specs-to-topology decision tree, global nonlinear sensitivity analysis on topology and sizing parameters, and determines analytical expressions of performance trade-offs. Knowledge extraction technique using binary decision trees is discussed in [35]. Decision trees utilize performance attributes to partition a set of circuit topologies based on their achievable performance, e.g., topologies that offer a certain Gain value, slew rate, and so on. Clustering features has been proposed to build classification schemes for analog circuits [36]. However, these methods did not consider knowledge structures that include topology similarities, differences, abstractions or design causal reasoning of the represented circuits. Meanwhile, it is very important for us to understand the purpose of certain topology and its performance trade-off. Moreover, traditional clustering methods do not use symbolic expressions, which are powerful in giving insight into circuit behavior and performance.

Identifying design causal reasoning steps to create a circuit is similar to understanding system or process causality. Understanding causality has been an important research topic in biomedical, neurological field, etc.

[37] studies the causality between two simultaneously recorded biological signals, and investigate whether one signal is causing the other. Granger causality is utilized to study the influence between brain regions [38], the network connectivity across multiple brain regions [39]. [40] proposes noval causality measures (in time and frequency domains) for the linear regression model, which is a natural extension of Granger causality. Causal mechanism of a system, represented by causality diagram, is developed with fuzzy set theory [38]. Causality analysis is also applied in stock market area [41]. In physics field, [42] studies space-time causality. Causality-related algorithms learn the structure of the causal model of a process and its parameters from a set of samples. Simon's causal ordering algorithm [43], a seminal work, generates partial causal graphs to indicate the causal relations among the parameters of a structural description of a system modeled as sets of lin-

ear equations. Nayak [44] studies causal approximations through bipartite matching as a way to tackle the complexity of causal modeling of physical systems. [45] extends the algorithm by Simon to under-constrained, structural equations. In spite of this work, to the best of our knowledge, mining design causal reasoning from design literature has not been yet studied.

To conclude, current analog circuit design automation techniques evolve different algorithms on design space and performance space modeling. Results obtained from modeling/data mining algorithms has not been further studied on how to interpret and convert them into design knowledge. Besides, there is no systematic approach to characterize topology similarities, build topology abstraction, as well as understanding causality encoded in the design process.

2.3 Associative component

Design knowledge mining first mines associative component, the hierarchical description for a group of circuits and their corresponding abstract circuits at various levels of abstraction. Every circuit topology is characterized by symbolic, mathematical relations between a set of parameters and holds for a set of constraints. Circuit parameters are physical properties that can be measured (observed), like device dimensions, small signal parameters, value ranges of the signals (voltages and currents), etc.

More details on theoretical characteristics of building symbolic models are presented in [46].

For a set of topologies, signal flow graphs are first generated by transferring transistor into its symbolic model, which expresses linear AC performance and noise [21]. The nodes of SFG correspond to the circuit nodes and arcs describe node coupling. Every node is annotated with a symbolic expression of the node's pole. Every arc has a symbolic expression denoting node coupling shown in Figure 2.3. Furthermore, we identify circuit topology

features based on the devices, transistor types, connection patterns, and IO terminals. If circuit netlist or symbolic model is not available, other simulation data-based rules will be needed for the feature learning. Common topological features among multiple topologies will be abstracted. For example, Figure 2.2 shows schematics and SFGs of three topologically similar OTAs. We observe from SFGs that all circuits have differential inputs as common building block, subset {Vinp, Vinn, V3, V5, V6, V4, V0, V1}.

The hierarchy is built by aggregating common features among circuit topologies. Features of each circuit (instance) are summarized into sets: set I contains the common features for all instances at the same abstraction level, set U includes features that are unique to an instance or abstraction, hence distinguish it from the others, set E contains features related to enabling conditions/constraints, that must be met for the circuit to be functionally feasible, i.e., biasing building blocks.

Example: The case study discusses circuits 4(1), 4(2), 27 in Figure 2.2, which are topologically similar. Circuit 4(1) is a complementary folded cascode OTA, circuit 4(2) is nearly similar, but contains a feed-forward compensation (FFC). FFC connects the differential inputs to gate of M_3, M_5, M_{10}, M_{12}. Circuit 27 is also a complementary cascode OTA, but with a feed-forward-regulated cascode topology (FFR), {M_8, M_5}, {M_9, M_4}, {M_2, M_7}, {M_3, M_6}.

Symbolic flow graphs were built for the three circuits using symbolic modeling [47], [21]. Figure 2.4 presents their symbolic models and the model of the abstraction 4(1) & 4(2). The model of circuit 4(2) includes all nodes and edges of the model for circuit 4(1), but it also incorporates the arcs shown in bold, dotted lines. The four arcs between the differential inputs and outputs represent the additional feed-forward compensation of circuit 4(2). Figure 2.4 illustrates the model for circuit 27. It has the same topological features as circuits 4(1) and 4(2) but connected differently. The new coupling that shown in bold, dotted arcs are feed-forward regulation (FFR) for circuit

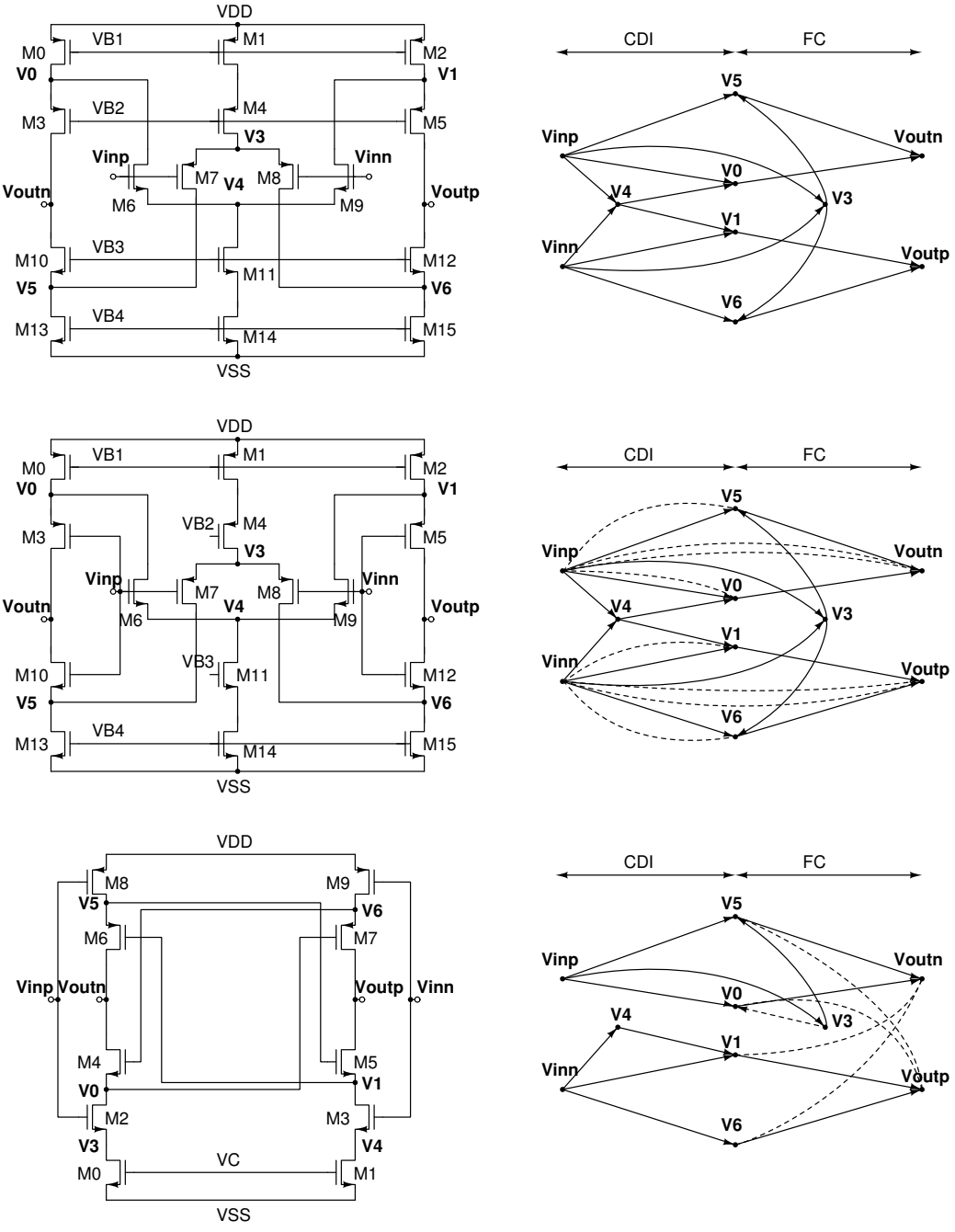


Figure 2.2: Circuits 4(1), 4(2) & 27 schematics and their symbolic models

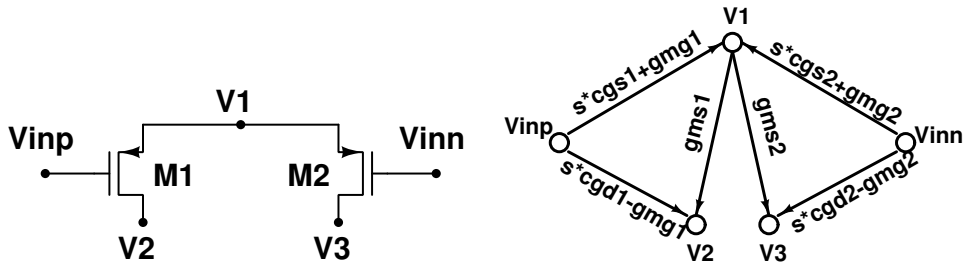


Figure 2.3: Differential input building block

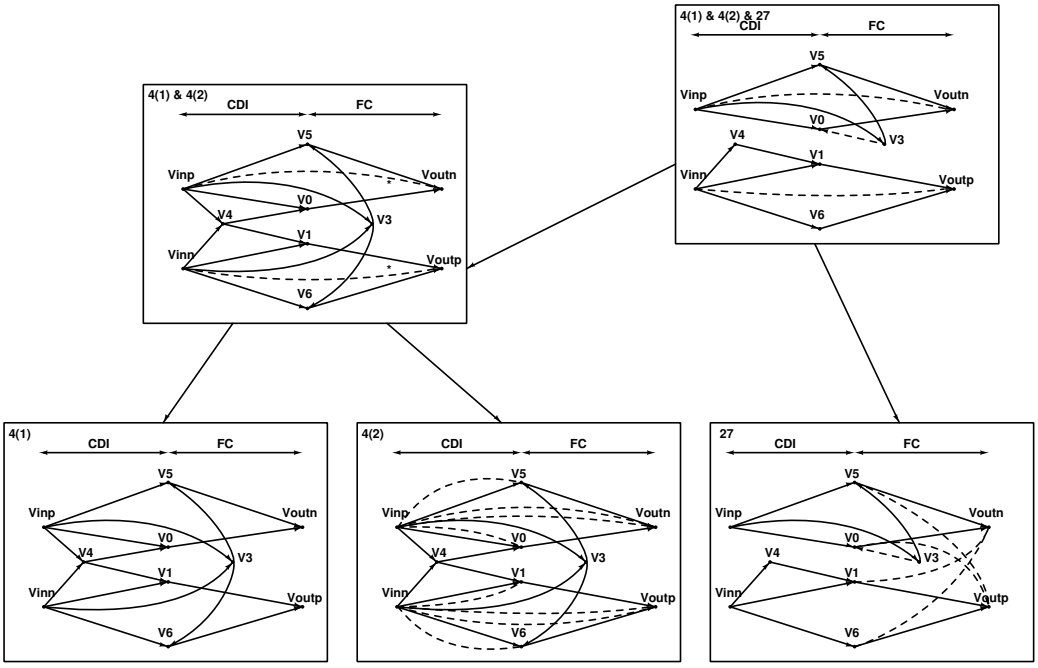


Figure 2.4: Symbolic circuit models for 4(1), 4(2) & 27 and their abstractions

27.

Figure 2.4 also presents the abstraction of the three circuits based on symbolic models. The nodes are the matched nodes of the three circuits (set I_1), and the edges in continuous line exist in all three circuits (set I_2). The four edges in bold, dotted lines between inputs and outputs are the overall couplings which each of three circuits includes, but using different ways of

implementing them (set I_3). The symbolic expressions of these arcs have the form $g_{m,x} + g_{m,y}(g_{m,z} + sC_{gd,v}) + \gamma$. The circuit nodes that are involved in implementing the overall arcs are part of alternatives (set I_4). Abstraction 4(1) & 4(2) describes circuits with or without feed-forward compensation, in which the two optional edges (marked with '$*$') between inputs and outputs introduce zeros $sC_{gd} - g_{mg}$ that are used in pole compensation. Abstraction 4(1), 4(2) & 27 gives a more general description with respect to input/output coupling, indicating optional edges can be implemented in different ways, e.g., feed-forward compensation, feed-forward regulation, etc.

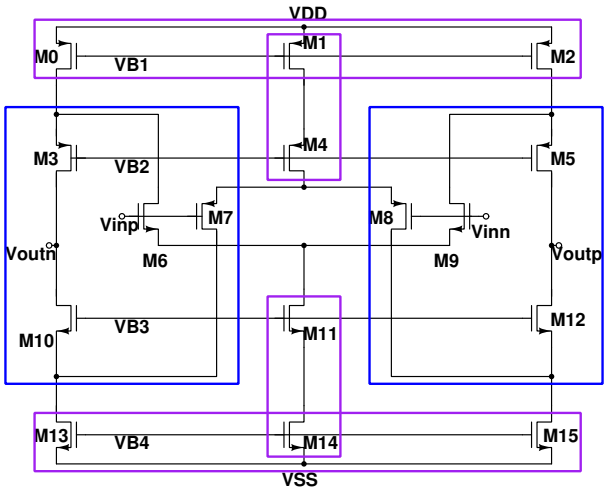


Figure 2.5: Circuit 4(1) schematics

Topological features are extracted in Figure 2.7. Common features of circuit 4(1), 4(2) are {complementary differential input, differential output, folded cascode} in set I. The unique feature of circuit 4(2), set U, is {feed-forward compensation}. Enabling features in circuit 4(1), 4(2) are {basic current source biasing, fully symmetry}. The corresponding devices and connections are colored in Figure 2.5, 2.6. Abstraction is built by extracting the common topological features for a higher hierarchy.

Summary. Topological features (or features) include not only well-known (traditional) building blocks, but also structural templates, hierar-

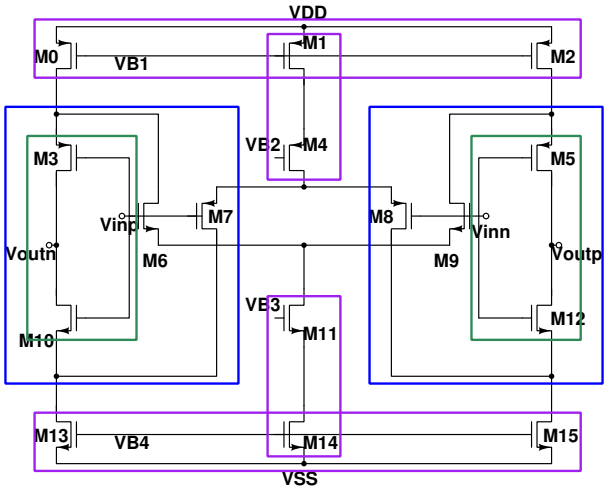


Figure 2.6: Circuit 4(2) schematics

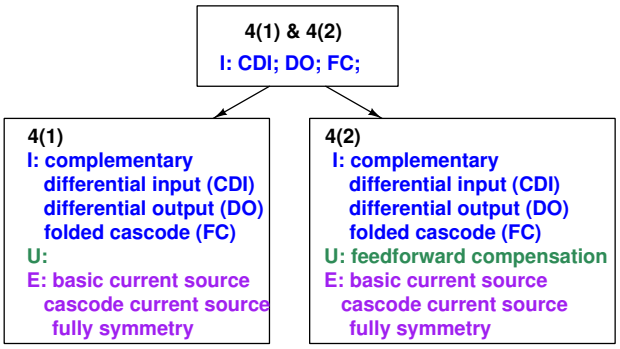


Figure 2.7: Topological features for 4(1), 4(2) and their abstractions

chical structures, feed-forward/feedback schemes [48]. As associative component focuses on topology attributes in terms of nodes, node couplings, higher level building blocks, and templates, technology aspects are not explored. Whereas feasibility of topology can be a technology dependent decision, linking topologies to technologies can be considered in future work, i.e., identifying topologies particular for advanced processes.

2.4 Performance capabilities component

Performance capabilities component expresses circuit/building block's parameter sensitivity, performance trade-offs and bottlenecks. It tends to cover all aspects of circuit/building block's performance potential that circuit designers considered in their design. Parameter sensitivity refers to how variation of parameter will make an impact on output performance, i.e., setting parameters to improve one performance attribute worsens another performance attribute. Performance trade-off refers to trade-offs between performance pairs or among high dimensional performance attribute sets, with bottleneck being single performance upper/lower bound. This chapter proposes trade-off tables in which columns represent performance attributes and rows indicate the parameters that control each attribute.

Table 2.1: Performance trade-offs of circuit 4(1) in Figure 2.5

Parameters	**CM gain**	**Gain**	**Noise**	**DP0**	**GPP0**
g_{md3}, g_{md10}	↓	↓	↑	↑	↑
g_{ms3}, g_{mg6}, g_{mg7}, g_{ms10}	↑	↑	↓	–	↑
C_{gd3}+C_{db3}+C_{gd10}+C_{db10}	–	–	–	↓	↓
g_{ms6}, g_{ms7}	↓	–	↑	–	–
g_{mg0}, g_{ms0}, g_{mg3}, g_{mg4}, g_{ms4}, g_{mg10}	–	–	↓	–	–
C_{gd6}+C_{gs6}+C_{gd7}+C_{gs7}	–	–	↑	–	–
g_{md0}, g_{md6}, g_{md7}, g_{md13}	↓	↓	↑	–	↑
g_{md4}, g_{md11}	↑	–	↑	–	–

Example: Table 2.1, 2.2 illustrate the performance trade-off tables for circuit 4(1), 4(2) in Figure 2.5, 2.6 individually. The columns are performance attributes {CM gain (common mode gain), Gain, noise, DP0, and GPP0}. Performance DP0 refers to the first dominant pole, GPP0 = Gain * DP0. Small signal parameters are chosen in the experiment as they correspond to symbolic models and reveal device operating insight. They are functions of device dimensions, thus can be tuned from circuit sizing input. The rows indicate different parameters (small signal parameters {g_{mg}, g_{md}, g_{ms}, C_{gd},

Table 2.2: Performance trade-offs of circuit 4(2) in Figure 2.6

Parameters	**CM gain**	**Gain**	**Noise**	**DP0**	**GPP0**
g_{md3}, g_{md10}	↓	↓	↑	↑	↑
g_{mg3}, g_{mg6}, g_{mg7}, g_{mg10}	↑	↑	↓	–	↑
C_{gd3}+C_{db3}+C_{gd10}+C_{db10}	–	–	–	↓	↓
g_{ms6}, g_{ms7}	↓	–	↓	–	–
g_{mg0}, g_{ms0}, g_{mg4}, g_{ms4}	–	–	↓	–	–
C_{gd3}+C_{gs3}+C_{gd6}+C_{gs6}+ C_{gd10}+C_{gs10}+C_{gd7}+C_{gs7}	–	–	↑	–	–
g_{md10}, g_{md6}, g_{md7}, g_{md13}, g_{md4}, g_{md11}	↑	–	↑	–	–
g_{ms3},g_{ms10}	↓	–	↓	–	–

C_{gs}, C_{db}} in this example). Upward arrows express that increasing the parameter value improves the performance attribute. Downward arrows show that increasing the parameter value decreases the performance attribute.

To further merge trade-off tables of instances for abstraction, we need to understand similar trade-offs, dissimilarity between two trade-offs. Concepts are explained as follows.

Definition - similar trade-offs: Let's assume two trade-off tables for two instances, trade-off table T_1 of instance C_1 and trade-off table T_2 of instance C_2. Elements in trade-off table present circuit parameters' trade-off effects on performance attributes. First, let's consider that each trade-off element is controlled by a single parameter: parameter v_1 for circuit C_1 and parameter v_2 for circuit C_2. Trade-offs T_1 and T_2 are similar, if parameters v_1 and v_2 originate the same kind of variations (e.g., improving or worsening) for the performance attributes of the trade-offs. The two controlling parameters v_1 and v_2 are similar with respect to the two trade-offs. Second, let's assume that each trade-off is controlled by a set of parameters, sv_1 and sv_2, respectively. Trade-offs T_1 and T_2 are similar, if there is a mapping between all parameters in sets sv_1 and sv_2, so that the associated parameters are similar with respected to the two trade-offs. The two sets of controlling parameters, sv_1 and sv_2, are called similar.

Example: In Tables 2.1 and 2.2, the two trade-offs on the first row are similar and involve the similar parameters $\{g_{md3}, g_{md10}\}$ of the two circuits. The trade-offs on the seventh row are not similar. They show different variations of the performance attributes. The distance between two trade-offs over a circuit parameter domain Δ is defined by the total differences in the performance attribute values computed for domain Δ. The distance between the trade-offs characterizes their dissimilarity.

Definition - distance/dissimilarity between two trade-offs: Given trade-offs T_1 of circuit C_1 and T_2 of C_2 and parameter domain Δ, the distance between the two trade-offs is computed by the following expression:

$$dist(T_1, T_2) = \sum_{i=1}^{p} \int_{\Delta} w_i |A_{1,i} - A_{2,i}| \tag{2.1}$$

Trade-offs T_1 and T_2 are characterized by p performance attributes A_i. w_i are weights associated to each attribute both for normalization and for indicating the importance of the performance attribute. These weights are used to build relation *rel* of the performance attribute description of the trade-off [46].

Example: Considering that trade-offs T_1 and T_2 include only two performance attributes, the distance is computed by the expression $w_1 \int_{\Delta} |A_{1,1} - A_{2,1}| + w_2 \int_{\Delta} |A_{1,2} - A_{2,2}|$.

If a circuit design does not meet the problem requirements, the distance of trade-off T to satisfying the problem requirement is equal to the minimum distance of the performance attributes that do not meet the constraints to the requirements computed over the domain Δ' for which the minimum distance is achieved.

Definition - distance to satisfying the problem requirements: The distance to satisfying the problem requirements is computed as follows:

$$dist(PC, T) = min \sum_{i=1}^{p} \int_{\Delta'} w_i |PC_i - A_i| \tag{2.2}$$

where PC_i are the p unsatisfied performance requirements, and A_i are the corresponding performance attributes.

Definition - Pareto criterion: For parameter domain Δ, trade-offs T_1 of circuit C_1 dominates trade-offs T_2 of circuit C_2, if for every performance attributes $A_{1,i} \in T_1$ and $A_{2,i} \in T_2$, $A_{1,i} > A_{2,i}$ or $|A_{1,i} - A_{2,i}| < \varepsilon$, where ε is application dependent.

The trade-offs and bottlenecks for abstractions are found by combining the trade-offs of their related instances. The resulting trade-off table includes the similar trade-offs as well as merged trade-offs, which express the alternative trade-offs of the instances. Merging two trade-offs pertaining to two circuits is performed according to the next definition.

Definition - merging two trade-offs: The trade-off tables of instances are combined to produce trade-off tables of the corresponding abstraction through similar trade-offs. The combined tables indicate the common trade-offs specific to the similar parameters of the instances and alternative trade-offs that are possible through the distinct (specific) parameters of the instances. The merged trade-off of two trade-offs $T_1 \in C_1$ and $T_2 \in C_2$ includes the similar performance attribute entries of the two trade-offs (e.g., if both entries are $\uparrow$ or $\downarrow$) or $-$ and the dissimilar entries, which express the alternative performance attribute variations, like $\uparrow$ / $\downarrow$, $\uparrow$ /$-$, and $-$/ $\downarrow$. The similar controlling parameters of the similar performance attributes are grouped together in the merged trade-off table.

Lemma: When grouping multiple trade-offs, the merged table includes only the following type of entries: $\uparrow$, $\downarrow$, $-$, $\uparrow$ / $\downarrow$, $\uparrow$ /$-$, $-$/ $\uparrow$, $\uparrow$ / $\downarrow$ /$-$.

Proof: Any other alternative is reduced to one of the above seven situations. For example merging $\uparrow$ / $\downarrow$ with $\uparrow$ results in $\uparrow$ / $\downarrow$, or merging $\uparrow$ / $\downarrow$ with $-$ produces $\uparrow$ / $\downarrow$ /$-$.

Example: Table 2.3 describes the trade-off table computed for the abstraction representing circuits 4(1) and 4(2). The first four rows present trade-offs that are common to the two circuits, hence are trade-offs of the

Table 2.3: Performance trade-offs of the abstraction 4(1) & 4(2)

Parameters	**CM gain**	**Gain**	**Noise**	**DP0**	**GPP0**
g_{md3}, g_{md10}	$\downarrow$	$\downarrow$	$\uparrow$	$\uparrow$	$\uparrow$
g_{mg6}, g_{mg7}, g_{mg10}, g_{ms3}/g_{mg3}	$\uparrow$	$\uparrow$	$\downarrow$	$-$	$\uparrow$
$C_{gd3}+C_{db3}+C_{gd10}+C_{db10}$	$-$	$-$	$-$	$\downarrow$	$\downarrow$
g_{ms6}, g_{ms7}	$\downarrow$	$-$	$\downarrow$	$-$	$-$
g_{mg0}, g_{ms0}, g_{md4}, g_{ms4}, $g_{ms3}/-$, $g_{mg10}/-\}$	$-$	$-$	$\downarrow$	$-$	$-$
C_{gd6}, C_{gs6}, C_{gd7}, C_{db7}, $-/C_{gd3}$,$-/C_{gs3}$,$-/C_{gd10}$,$-/C_{gs10}\}$	$-$	$-$	$\uparrow$	$-$	$-$
g_{md0}, g_{md6}, g_{md7}, g_{md13}, $-/g_{md4}$, $-/g_{md11}\}$	$\downarrow / \uparrow$	$\downarrow /-$	$\uparrow$	$-$	$\downarrow /-$
g_{md4}/g_{ms3}, $g_{md11}/g_{ms10}\}$	$\uparrow / \downarrow$	$-$	$\uparrow / \downarrow$	$-$	$-$

abstraction too. Each of the trade-offs have the same controlling parameters and express the same type of performance attribute variations. The trade-offs in rows five and six show for both circuits the same variation of noise, but the two circuits differ with respect to the controlling parameters that set the noise attribute. The parameters outside the accolades are specific to one of the circuits, e.g., $-/C_{gd3}$ in row six indicates that the parameter occurs in the second circuit, while the first circuit does not include a similar parameter (null parameter). Rows seven and eight present trade-offs for which the performance attributes have different attribute variations. The trade-off table indicates that the abstraction for circuits 4(1) and 4(2) offers more flexibility in improving the noise performance of the related circuits, but less capabilities in improving the gain and bandwidth. Parameters in row 1 and 2 improve GPP0, and row 1, 6, 7 better the noise performance. However, only row 2 improves gain, only row 1 improves DP0.

Tables 2.1, 2.2, 2.4 present trade-offs for circuits 4(1), 4(2), and 27 individually. Columns are the circuit parameters that control a specific performance. Rows include performance attributes CM gain, Gain, noise, dominant pole (DP0), and gain-pole product (GPP0). Up-arrows ($\uparrow$) represent that increasing the parameter improves the performance attributes, and down-arrow ($\downarrow$)

Table 2.4: Performance trade-offs of circuit 27

Parameters	CM gain	Gain	Noise	DP0	GPP0
C_{gd0}, C_{gs}, C_{gs4}, C_{gd6}, C_{gs6}	–	–	↑	–	–
g_{md0}, g_{md6}	↓	↓	↑	–	–
g_{mg0}, g_{mg6}	↑	↑	↓	–	↑
g_{ms0}	–	–	↓	–	–
g_{mg1}, g_{ms1}	–	–	↑	–	–
C_{gd2}	–	–	↓	–	–
g_{md2}, g_{md4}	↓	↓	↑	↑	↑
g_{mg2}	↓	↑	↑	–	↑
g_{ms2}	–	↓	–	–	↓
g_{mg4}	–	↑	↑	–	↑
g_{ms4}	↑	↓	–	–	↓
g_{ms6}	↓	↓	–	–	↓
g_{mg4}	↑	↑	↑	–	↑

describes that increasing the parameter decreases the performance attribute.

Table 2.5: Performance capabilities for the abstraction of circuits 4(1), 4(2) & 27

Parameters	CM gain	Gain	Noise	DP0	GPP0
g_{mg6}, g_{mg7}, g_{mg10}, $\{\{g_{ms3}/g_{mg3}\}/g_{mg2}\}$	↑	↑	↓	–	↑
$\{C_{gd6}, C_{gs6}, C_{gd7}, C_{db7}\}/$, $\{C_{gd0}, C_{gs}, C_{gs4}, C_{gd6}, C_{gs6}\}$	–	–	↑	–	–
g_{md0}, g_{md6}, $\{\{g_{md7}, g_{md13}\}/-$, $-/g_{md4}, -/g_{md11}\}/-$	↓ / ↑	↓ /–	↑	–	↓ /–
g_{ms0}, $\{\{g_{mg0}, g_{mg4}, g_{ms4}\}$ $\{g_{mg3}/-, g_{mg10}/-\}\}/-$	–	–	↓	–	–
$\{g_{ms6}, g_{ms7}\}/g_{mg2}$	↓	↓ / ↑	↑	↑ /–	↑
$\{g_{ms6}, g_{ms7}\}/\{g_{md2},g_{md4}\}$	↓	↓	↑	↑	↑

Abstraction for trade-off tables 4(1) & 4(2) was already discussed in Section 2.4. Table 2.5 presents the trade-offs of the abstraction build for circuits 4(1), 4(2) and 27 (abstraction 4(1), 4(2) & 27 in Figure 2.4). For brevity, the table includes only the trade-offs that show similarity of their attribute variations. Similar to Table 2.3, it indicates the higher flexibility of the re-

lated circuits to address frequency (i.e., DP0 and GPP0) and noise related attributes and less flexibility in achieving high CM and Gain. This suggests that design following the topological features {CDI, DO, FC} represented by abstraction in Figure 2.7 tackle better high-frequency requirements, but have less capability in achieving high gain. For example, using the last trade-off to decrease CM and Gain decreases noise, but then improving bandwidth reduces the circuit gain.

Summary. In performance capabilities component, trade-off table entries are computed by symbolic modeling, which limits output performances to ac/noise domain as symbolic model approximates circuit as linear system. With respect to modeling a global performance space, extended work is presented in Chapter 4 with the help of simulation tool.

2.5 Causal reasoning component

By reading analog circuit literature, mainly on op-amp/OTA design, and summarizing the design flows, we propose the main elements describing designer's reasoning process are as follows: (i) causality of a design step, (ii) utility, (iii) consistency, (iv) justified design step, (v) starting ideas and design sequence, and (vi) required design step. The elements are explained next.

Definition - causality of a design step: The causality of a design step expresses the reasons for using the step to produce a design, e.g., reducing the distance to specification (problem requirements):

$$(dist(goal, S)|\langle DF, Des, CP\rangle \rightarrow \langle DF', Des', CP'\rangle) \tag{2.3}$$

The definition states that a design step represents the association between the triplet $\langle$ (i) the set DF' of new design features introduced (at various abstraction levels) by the step, (ii) the resulting design Des' that in corporate features DF', and (iii) set CP' of the constraints for design Des' $\rangle$ and the

reason that justifies this decision represented by the distance *dist* between the desired *goal* and performance S of the current design Des evaluated for attributes DF incorporated by the design. Set CP are constraints over the design parameters of design Des, e.g., device dimensions and device operation modes.

Definition - utility: Utility captures quantitatively the justification of a design step with respect to the two criteria.

$$Utility = \langle \frac{dist(goal, S)}{dist(goal, S')}, \frac{R(CP')}{R(CP)} \rangle \tag{2.4}$$

Function R describes the constraining level defined by the constraints in set $CP(CP')$. For example, function R could represent the ranges of circuit parameters. The first utility term indicates the amount by which the performance of the new design S' is closer to the goal as compared to the performance of the initial design S. The second term shows the amount by which the design constraints were relaxed by the new design. Design constraints include transistor operation regions, device matching, etc.

Definition - consistency: A design sequence is consistent if the end design (the design after the last design step) meets the specification.

Definition - justified design step: Given a goal, a design step is justified, if at least one of its resulting utility components (distance to goal or constraining level) is greater than 1. A design sequence is justified if each design step is justified. Note that a design sequence might not be justified, if it includes at least one design step, which neither improves performance nor relaxes the design constraints. Unjustified design steps might arguably represent reasoning flaws.

Definition - starting ideas and design sequence: Given entire design flow, starting ideas are the triplets $\langle DF, Des, CP \rangle$ that originate the design flow. DF, Des, and CP have the same meaning as in the design step causality definition.

Design sequence is a succession of design steps, for which the causality of each step is expressed as in expression (2.3). The definition of design step causality indicates that there are two kinds of steps depending on their justification of being used to create a new circuit design: (i) Design steps that are justified by changing the performance trade-offs and bottlenecks through the performance attributes of the modified topological features, and (ii) Design steps that relax the design constraints of a solution due to the new topological features.

Definition - required design step: Given a design sub-sequence Sub, a design step D is required for sub-sequence Sub, if every consistent design sequence including Sub also includes step D following Sub. A required sub-sequence is a sequence of required steps. Required sub-sequence represent unique topological features that cannot be replaced by equivalent structures. Common sub-sequence of distinct design sequence characterize the flexibility of the circuit structures corresponding to the sub-sequence with respect to accommodating the different goal requirements tackled by the circuits. For example, biasing circuit 1, 2, cascode biasing circuit are required steps in the sequence in Figure 2.8 as the circuit would not operate without biasing.

Theorem: A design sequence consistent for goal G is inconsistent for goal G' ($G \neq G'$), if and only if it includes at least one design step that introduces a bottleneck for goal G'. The first design step (in the sequence) introducing a bottleneck is called *branching step* for goal G'.

Proof: If there was no bottleneck for goal G' then the sequence is justified, which contradicts the assumption.

Example: Figure 2.8 illustrates circuit 4(1) and its corresponding mined causal reasoning. Circuit 4(1) is a complementary folded cascode OTA for low voltage Δ Σ modulators [2]. The causal reasoning component includes how starting ideas are combined, followed by the design sequence to create the circuit. According to the document [2], the starting ideas included combining a fully differential structure with a complementary folded cascode feature.

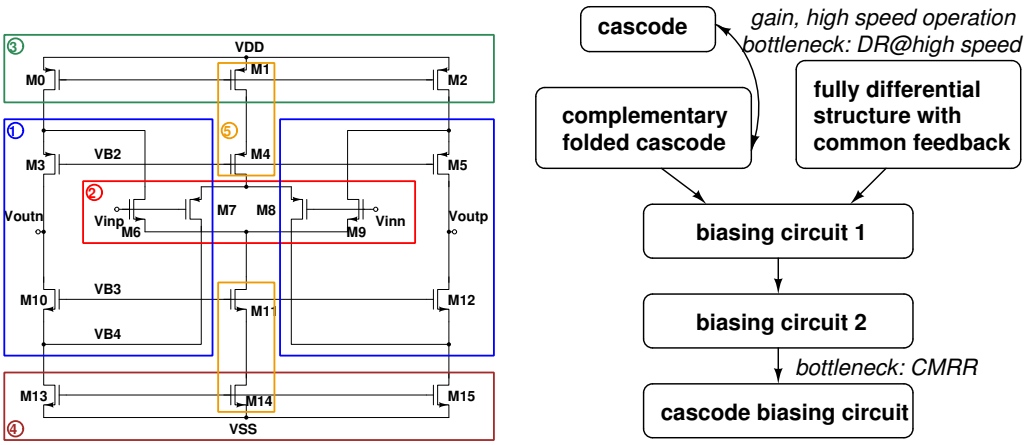


Figure 2.8: Circuit 4(1) causal reasoning component

The justification for using complementary folded cascode is its high gain, high speed operation, and better dynamic range (DR) for low voltage supply [2]. The corresponding structure in Figure 2.8 was labeled as 1, 2. The two design steps in the sequence introduced the two biasing circuits labeled as 3 and 4 in Figure 2.8. The last design step added cascode current source, labeled as 5 in the figure. This step is justified by the need to improve common mode rejection ratio (CMRR).

Based on different origins that starting ideas are obtained from, we classify the corresponding topological features into the following types:

• *Private set of features* (set Γ): The set includes features that have been used by the same group in their previous circuit designs and also used in the current circuit.

• *Collective set of features* (set Λ): The set includes the features that occur in designs devised by other groups and which are present in the analyzed circuit too. Such design features are discussed in other papers cited by the paper presenting the considered design features.

• *New design insight* (set Ψ): The set includes features that are new insight acquired by the authors of a circuit. This insight cannot be found in

the related design literature or in the previous design work of the authors.

• *Common set of features* (set Θ): It incorporates features that are well-known, traditional design knowledge in textbook [49].

Complete set of features (set Σ) represents all features in a circuit.

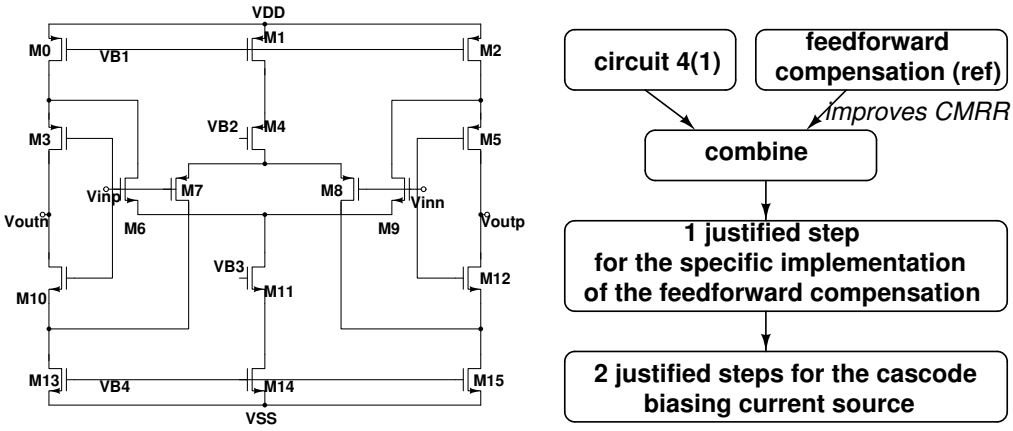


Figure 2.9: Circuit 4(2) causal reasoning component

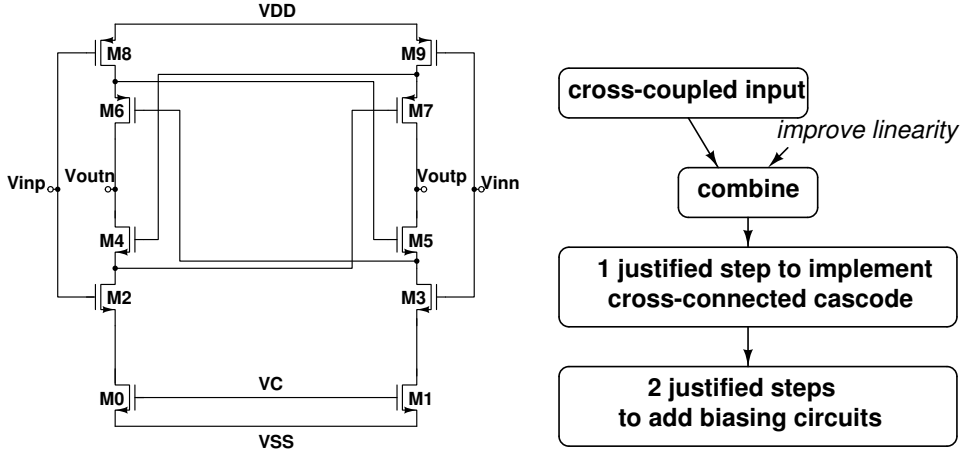


Figure 2.10: Circuit 27 causal reasoning component

Figure 2.9, 2.10 presents the causal reasoning component for circuits 4(2) and 27. The starting ideas for circuit 4(2) (Figure 2.6) are combining circuit

4(1) (set Γ) and feed-forward compensation (set Λ) [2]. The first justified step refers to the specific implementation of the feed-forward compensation scheme. The next two steps relate to the refinement of the cascode biasing current sources to accommodate the presence of feed-forward compensation. The starting idea for circuit 27 (Figure 2.2) is the abstract idea of using cross-coupled input as means to improve linearity. This idea is not part of sets Γ or Λ, and hence represents new insight (set Ψ). The first step of the sequence includes implementing the abstract idea by using cross-coupled cascodes. The next two justified design steps add necessary biasing.

Discussions. For topology selection purpose, performance trade-offs of the instances and abstractions indicate the circuits' capabilities to achieve a range of performance values as well as their bottlenecks. This supports early pruning of the abstractions (and their related instances) that are unlikely to meet a certain requirements, e.g., cluster 4(1) & 4(2) & 27 is not recommended for high gain applications.

Case studies: The next section presents four case studies to illustrate mining of the starting ideas and design sequence in other design papers.

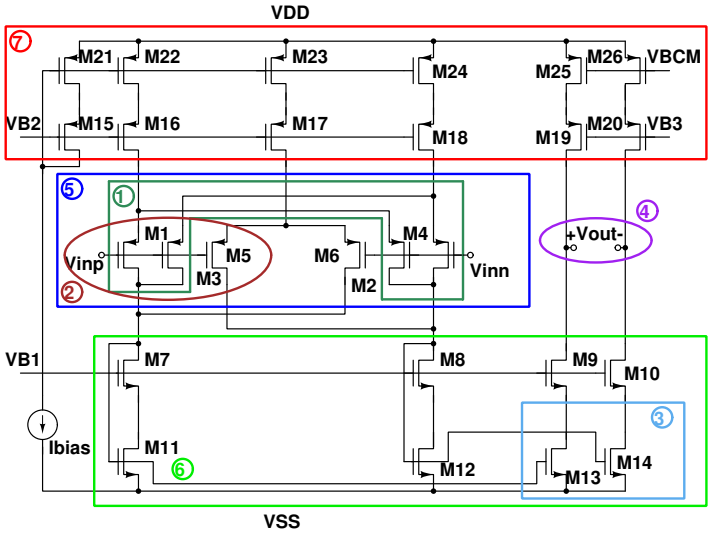


Figure 2.11: Circuit 5(1) schematic in Figure 2.19

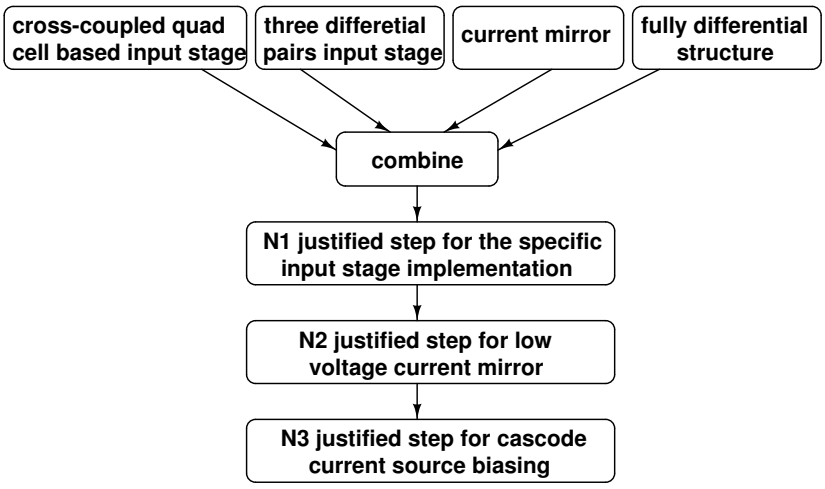


Figure 2.12: Starting ideas and design sequence for Circuit 5(1)

Case study 1. Figure 2.11 shows circuit 5(1) schematic in Figure 2.19, a highly linear, fully differential OTA [50]. The complete set of features (set Σ) includes all topological features of the circuit: three differential input stage (a cross-coupled quad cell based input stage together with an additional linearizing symmetrical differential pair), low-voltage current mirror, fully differential structure, and cascode current source biasing. Figure 2.12 illustrates the mined starting ideas and the corresponding design sequence. The starting ideas of the design (set S) include cross-coupled quad cell based input stage (set Λ), three differential pairs input stage (set Λ), current mirror at second stage (set Λ), and fully differential structure (set Θ). The starting ideas correspond to structures labeled as 1, 2, 3, 4 in Figure 2.11. Beginning with the starting ideas in set S, the uncovered features of the circuits are computed by the difference: set Σ − set S. These features are labeled as 5, 6, 7 in the figure. They behave as design sequence that will be added following starting ideas. The order to analyze feature justification starts first with the one that is more likely/frequently used. The design sequence is as follows.

Design step $N1$ implements a three differential input stage, a cross-coupled quad cell based input stage together with an additional linearizing

Table 2.6: Circuit 5(1): THD comparison of OTA, OTA2

Circuit	Ibias[uA]	input voltage[Vpp]	THD[%]
OTA	200	1	0.995
OTA2	200	1	8.185

symmetrical differential pair. The three differential input pair is justified by a previous and similar design (cited by the paper) that uses a cross-coupled quad cell input stage. The additional symmetrical differential pair realizes linear CMOS trans-conductance elements. To justify the need of having this structure as part of step $N1$, transient response of this circuit was compared with a straightforward reference, the circuit with single differential pair input (OTA2). Transistor sizing followed the design constraints presented in the paper. Both circuits are configured with the same biasing current and input signal. Table 2.6 summarizes the total harmonic distortion results, showing that OTA2 results 8 times worse linearity than OTA.

Table 2.7: Circuit 5(1): Performance trade-offs of OTA

Parameters	CM gain	Gain	Noise	Dominant Pole
g_{md7}	$\downarrow$	$\downarrow$	$\uparrow$	$\uparrow$
C_{gd7}+C_{db7}	$-$	$-$	$-$	$\downarrow$
g_{mg9}	$-$	$-$	$\uparrow$	$-$
g_{md9}	$\downarrow$	$\downarrow$	$\uparrow$	$-$
g_{ms9}	$\uparrow$	$\uparrow$	$\downarrow$	$-$

Design step $N2$ adds a low-voltage current mirror to the circuit. Using a low-voltage current mirror instead of a basic current mirror is due to its high output resistance and reduced drain-source voltage (only 0.4V margin is left across devices M9 and M13 for keeping both of them in saturation). In order to illustrate its advantage with respect to high output resistance, Table 2.7, 2.8 compare the trade-offs of the low-voltage current mirror (M7, M9, M11 and M13) (OTA) and basic current mirror (M11 and M13) (OTA3). The tables only include the parameters that cause different effects on gain,

Table 2.8: Circuit 5(1): Performance trade-offs of OTA3

Parameters	CM gain	Gain	Noise	Dominant Pole
C_{gs1}	–	–	↑	–
C_{gs2}	–	–	↑	–
C_{gs3}	–	–	↑	–
C_{gs4}	–	–	↑	–
C_{gs5}	–	–	↑	–
C_{gs6}	–	–	↑	–
g_{ms19}	–	–	↑	–

noise, and bandwidth. Table 2.8 indicates a higher flexibility of gain and pole position because of the cascode devices M7 and M9. The additional terms g_{ms9}/g_{md9} introduces an enhanced Gain for the low-voltage current mirror structure.

Design step $N3$ adds a cascode current source biasing to the circuit. Using cascode biasing instead of single current source biasing is justified by an improved power supply rejection ratio (PSRR). For unity gain configuration, simulation shows that circuit 5(1) achieves 5.44dB rejection. Replace 5(1) with single current source biasing results in -7.006dB rejection, which is more than 12dB less.

Case study 2. Circuit 21 in Figure 2.13 is a multipath OTA [51]. The complete set of features (set Σ) includes: three path OTA (a folded cascode OTA, a current-mirror cascode OTA, a current-mirror folded-cascode OTA), double differential cross-coupled input, CMFB circuit, and current source biasing. The starting ideas (set S) include the following features: a folded-cascode OTA (set Γ), a current-mirror cascode OTA (set Γ), a complementary folded-cascode OTA (set Λ), a multi-path OTA (set Λ), and a fully differential structure (set Θ). The starting ideas correspond to the structures labeled as 1 and 2 in Figure 2.13. The multi-path OTA is an abstract idea originally discussed in a cited paper, which implements a two-path OTA. Beginning with the starting ideas in set S, the uncovered features of the design are computed by the difference set Σ − set S, and labeled as 3, 4, and

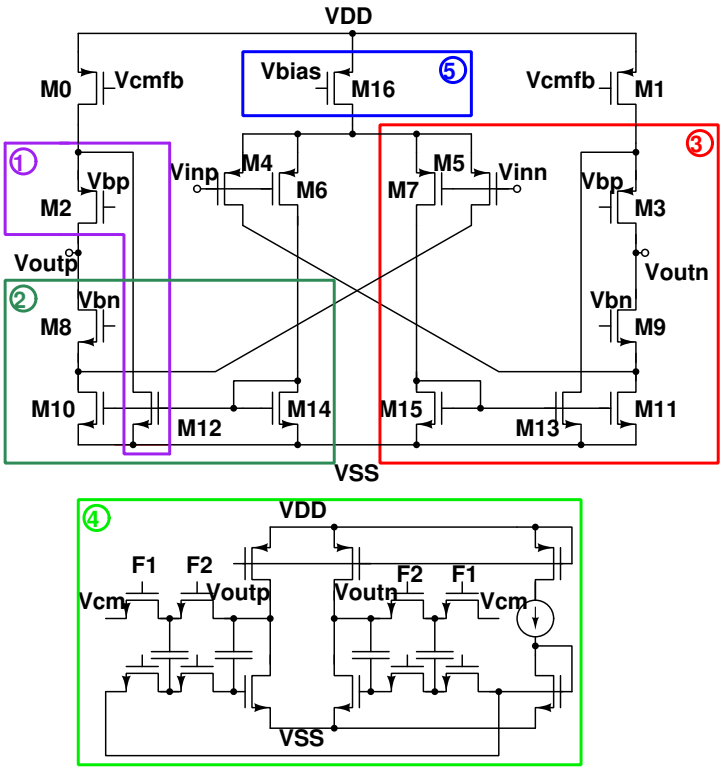


Figure 2.13: Circuit 21 schematic in Figure 2.21

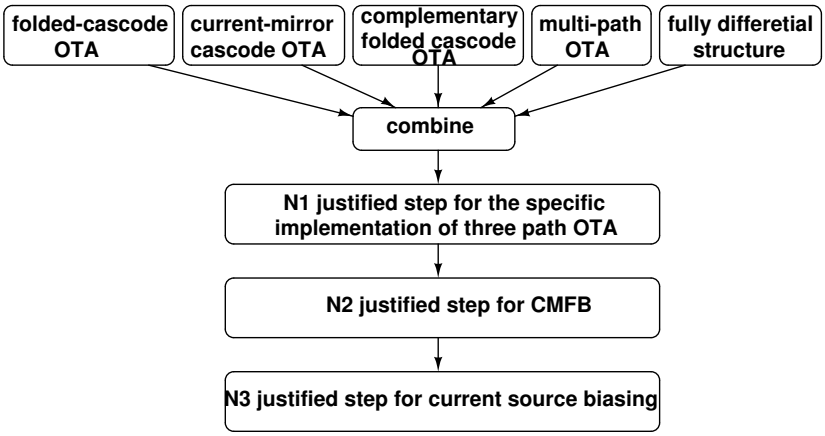


Figure 2.14: Starting ideas and design sequence for Circuit 21

5. Similar to circuit 5(1), for each feature, justification first starts with the more concrete feature. Figure 2.14 presents the causal reasoning for circuit 21.

Design step $N1$ adds the three path OTA including a folded cascode OTA, a current-mirror cascode OTA, and a current-mirror folded-cascode OTA. The double differential cross-coupled input is the unique causal structure, which is required in consistent design sequence. The three path OTA structure is justified by a previous design that used a two-path OTA. In order to illustrate its advantage over a two path OTA, tables 2.9 and 2.10 present the trade-offs of circuit 21 (OTA) and the two path OTA without devices M4 and M5 (OTA2). The common structures in the two OTAs result in the same trade-offs on performance, which, for brevity, were not included in the tables. Regarding Gain, the additional parameters g_{mg4} and g_{md4} enhance gain by g_{mg4}/g_{md4} in the three path OTA. Meanwhile, C_{gd4}/C_{gs4} degrades the noise performance.

Table 2.9: Circuit 21: Performance trade-offs of OTA

Parameters	CM Gain	Gain	Noise	Dominant Pole
g_{ms4}	–	–	↓	–
C_{gd4}	–	–	↑	–
C_{gs4}	–	–	↓	–
g_{md4}, g_{md10}	↑	↓	↑	–
g_{mg4}	–	↑	–	–
g_{mg6}	–	↑	↑	–
g_{ms6}	–	–	↑	–
g_{ms8}	↓	↑	↓	–

Design step $N2$ adds CMFB to the circuit. A fully differential amplifier usually requires CMFB circuit to stabilize the common mode level of the outputs. To justify using a CMFB circuit, sensitivity analysis was performed on the common mode configured circuits. Sensitivity analysis studies the mapping of all circuit parameter variations onto the performance specifications of the circuit [52]. The results show that devices M12, M6, M16 and M14

Table 2.10: Circuit 21: Performance trade-offs of OTA2

Parameters	CM Gain	Gain	Noise	Dominant Pole
g_{md10}	↓	↓	↑	–
g_{mg6}	↑	↑	↑	–
g_{ms6}	↓	–	↑	–
g_{ms8}	↑	↑	↓	–

increase the common mode gain, which degrades the common mode performance. For comparison, in a circuit without CMFB circuit, devices M13, M1, M7, M16, M15, M6, M10, and M11 reduce the common mode performance for an equal sensitivity value. Therefore, $N2$ is justified by improved common-level of the outputs.

Design step $N3$ adds current source biasing to the circuit. Current source biasing is required as the circuit would not operate without biasing.

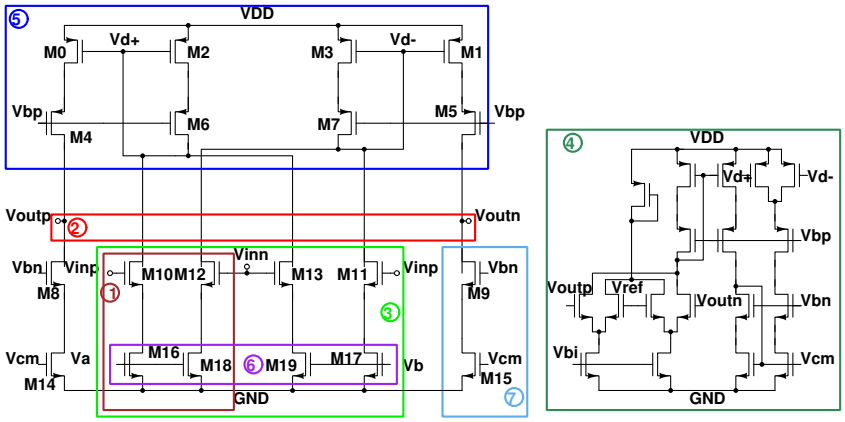


Figure 2.15: Circuit 26 schematic in Figure 2.21

Case study 3. Circuit 26 in Figure 2.15 is a linearized OTA for low-voltage and high-frequency applications [53]. The complete set of topological features (set Σ) includes the following features: fully differential, double cross-coupled pseudo differential pair input, low-voltage current mirror, linear region transistors, common mode feed-forward (CMFF) and CMFB circuits, and current source biasing. The starting ideas (set S) include the following: combining

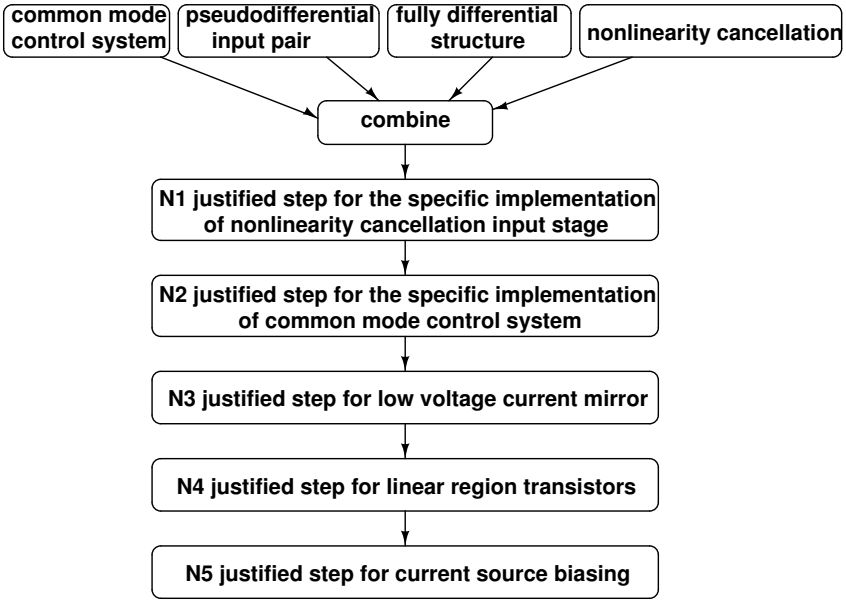


Figure 2.16: Starting ideas and design sequence for Circuit 26

common mode control system (set Γ), pseudo differential input pair (set Λ), fully differential structure (set Θ), and nonlinearity cancellation (set Ψ). The starting ideas correspond to the structures labeled as 1 and 2 in Figure 2.15. Nonlinearity cancellation is among the starting ideas, representing the abstract idea use trans-conductance linearization. Figure 2.16 shows the causal reasoning for circuit 26. The design steps include the justified steps for the input stage implementation for nonlinearity cancellation, the implementation of common mode control system, low-voltage current mirror, linear region resistors, and current source biasing circuit. The uncovered features are labeled as 3, 4, 5, 6 and 7 in Figure 2.15.

Table 2.11: Circuit 26: THD comparison of OTA, OTA2

Circuit	input voltage[Vpp]	THD[%]
OTA	1	6.778
OTA2	1	7.093

Table 2.12: Circuit 26: Performance trade-offs of OTA

Parameters	CM Gain	Gain	Noise	Dominant Pole
C_{gd13}	–	–	↑	–
g_{md13}	↓	↓	↑	–
g_{mg0}	↑	↑	↓	–
g_{ms0}	–	–	↑	–
g_{md4}, g_{md8}	↓	↓	↑	↑
g_{ms4}, g_{mg10}, g_{ms10}, g_{mg13}	↑	–	↓	–
g_{mg6}, g_{ms6}, g_{ms13}	–	–	↓	–
C_{gs10}, C_{gs13}	–	–	↓	–
g_{md16}, g_{md19}	↑	–	↑	–

Table 2.13: Circuit 26: Performance trade-offs of OTA2

Parameters	CM Gain	Gain	Noise	Dominant Pole
g_{mg0}, g_{mg10}	↑	↑	–	–
g_{ms0}, g_{ms4}, g_{mg6}, g_{ms6}	–	–	↓	–
g_{md4}, g_{md8}	↓	↓	↑	↑

Design step $N1$ is double cross-coupled pseudo differential pair with degenerated transistors input. Step $N1$ is justified by a previous design (cited by the paper) that uses pseudo differential input pair. The cross-coupled structure implements the abstract idea of nonlinearity cancellation. To justify the resulting structure, transient response was compared with a circuit with single differential pair input and with ideal current source (OTA2). Table 2.11 summarizes the total harmonic distortion for 10MHz sine input. The circuit proposed in Figure 2.15 achieves better linearity performance. Tables 2.12 and 2.13 present the trade-offs of these two circuits for gain, noise and bandwidth. Circuit 26 has reduced gain performance because of the cross-coupled input stage. Linearity is achieved at the cost of gain reduction. The dominant pole position is the same for two circuits, but the cross-coupled stage introduces more noise.

Design step $N2$ adds the CMFF and CMFB circuits. This step is justified by the requirement of a proper common mode control system in a pseudo

differential structure. The results for sensitivity analysis show that the circuit without common mode control has the same sensitivity but with larger cost. Devices M19, M16, M2, M8, M14, M4 and M6 degrade the common mode gain more in the circuit without common mode control.

Design step $N3$ add the low-voltage current mirror to the circuit. Having a low-voltage current mirror instead of a single current mirror is justified by its high output resistance and reduced output voltage (as only 0.4V headroom is left for keeping devices M0 and M4 in saturation).

Design step $N4$ adds the linear region transistors to the circuit. The degenerated resistors in the pseudo differential stage are implemented by transistors. The linear region transistors are justified by having the transconductance tuning ability that compensates for the variation caused by the fabrication process and temperature.

Design step $N5$ adds the current source biasing to the circuit. The current source biasing is a required step in the design sequence as otherwise the circuit would not operate.

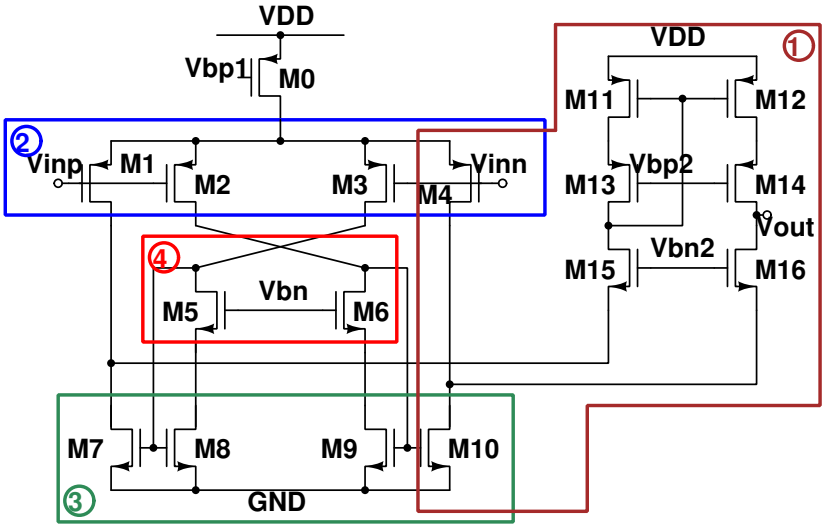


Figure 2.17: Circuit 28 schematic in Figure 2.21

Case study 4. Circuit 28 in Figure 2.21 is a recycling amplifier based on

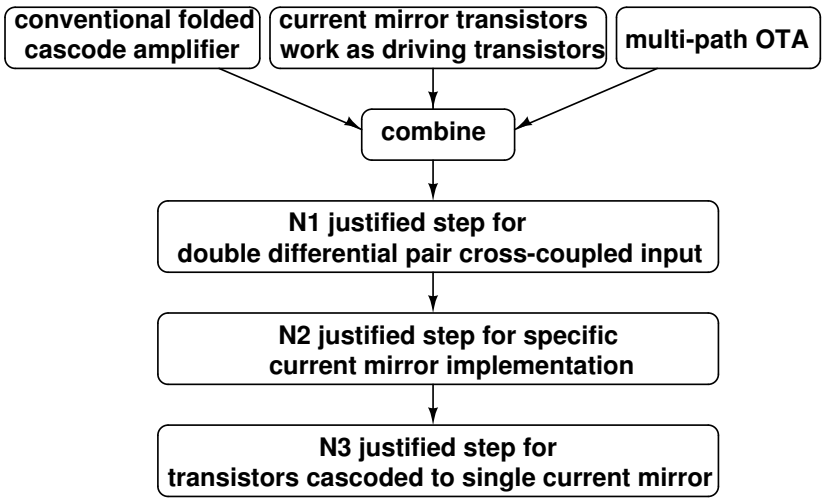


Figure 2.18: Starting ideas and design sequence for Circuit 28

folded-cascode OTA [54]. The complete set of features (set Σ) includes the following topological features: double differential pair cross-coupled input, current mirror transistors as driving transistors, and transistors cascoded to single current mirror. The starting ideas (set S) include conventional folded cascode amplifier (set Γ), current mirror transistors work as driving transistors (set Γ), multi-path OTA (set Γ), and single-ended output structure (set Ψ). The starting ideas correspond to the structures labeled as 1 in Figure 2.17. Figure 2.18 presents the causal reasoning information for circuit 28. The design sequence starts from combining conventional folded cascode amplifier, current mirror transistors work as driving transistors, and multi-path OTA. The design steps include the justified step for adding to the circuit the double differential pair cross-coupled input, specific implementation of the current mirror, and the transistors cascoded to a single current mirror. The features that are added by the design sequence are computed by the difference set Σ − set S, and are labeled as 2, 3 and 4 in Figure 2.17.

Design step $N1$ implements the double differential pair cross-coupled input. Step $N1$ is justified by the signal polarity since the output current is

the sum of positive input path and negative input path.

Table 2.14: Circuit 28: Performance trade-offs of OTA

Parameters	CM Gain	Gain	Noise	Dominant Pole
g_{ms13}	–	–	↓	–
C_{gd1}+C_{db1}, C_{gs2}+C_{sb2}, C_{gd10}+C_{db10}, C_{gs16}+C_{sb16}	–	–	–	↓
g_{md1}	↓	↓	↑	↑
C_{gd1}, C_{gs1}, C_{gs2}, C_{gd10}	–	–	↓	–
g_{md2}, g_{md5}, g_{md12}, g_{md13}, g_{md15}	↓	↓	↑	–
g_{mg1}, g_{mg2}	–	↑	↓	–
g_{ms1}	–	–	–	↑
C_{gd2}, C_{gd8}, C_{gd11}, C_{gd12}	–	–	↑	–
g_{ms2}	–	–	↓	↑
g_{mg10}	↑	↑	–	–
g_{md10}	↓	↓	↑	↑
g_{mg12}	–	↑	–	–
g_{ms12}, g_{mg14}, g_{mg16}	–	–	↑	–
g_{md14}, g_{md16}	↓	↓	↑	–
g_{ms16}	↓	↑	↓	↑

Table 2.15: Circuit 28: Performance trade-offs of OTA2

Parameters	CM Gain	Gain	Noise	Dominant Pole
g_{ms12}, g_{mg14}, g_{mg16}	–	–	↓	–
C_{gd11}+C_{db11}, C_{gs13}+C_{sb13}	–	–	–	↓
C_{gd1}, C_{gs1}, C_{gd11}, C_{gd12}	–	–	↑	–
g_{md1}, g_{md10}, g_{md12}, g_{md13}, g_{md14}, g_{md15}, g_{md16}	↓	↓	↑	–
g_{mg1}	↑	↑	↓	–
g_{ms1}	↓	–	↓	↑
g_{md11}	–	–	–	↑
g_{mg12}	–	↑	–	–
g_{ms13}	–	–	↓	↑
g_{ms16}	↓	↑	↓	–

Design step $N2$ adds the current mirror transistors as driving transistors. The recycling current mirror devices are justified by their additional current driving capability. In order to illustrate this advantage, tables 2.14 and 2.15 present the trade-offs of the recycling folded cascode and traditional folded cascode circuit (OTA2). For brevity, only the parameters having different effects on gain, noise, bandwidth performance are shown. Table 2.14 indicates

a higher flexibility of the gain and bandwidth performance. Gain is enhanced by devices M2 and M5, and the dominant pole is pushed further away by adjusting the input parameters g_{md1} and g_{ms2}.

Table 2.16: Circuit 28: DC mismatch comparison of OTA, OTA3

Circuit	DC Mismatch[V]
OTA	0.651
OTA3	0.824

Design step $N3$ adds the transistors cascoded to single current mirrors. Using additional cascoded transistors is due to the reduced DC mismatch since current mirrors have specific sizing. DC mismatch simulation is done on circuit 28 and the circuit without devices M5 and M6 (OTA3). Table 2.16 summarizes the simulation results. Both circuits are configured by the same biasing current and input voltage. The cascoded transistors minimize the DC mismatch by 21%, which justifies this design step.

Summary. Causal reasoning component summarizes designer's reasoning flow, which is not studied previously in design automation field. Starting ideas are considered from more concrete features to more abstract ones, as we emphasize more on implementation-oriented solutions than qualitative arguments. Design steps are in decreasing order by their frequency of use. Causal reasoning limits in the large simulation cost when justify design step's causality and efforts to obtain sets Γ, Λ manually.

The next section discusses the algorithms to construct associative component, performance capabilities component, and causal reasoning component respectively.

2.6 Algorithms to mine design knowledge

2.6.1 Mining associative component

Algorithm 1 presents the algorithm to construct the associative component of knowledge representation. Inputs are a set of circuit schematics C_1, C_2, C_3,..., C_n, and output is $KnowStr$, the corresponding associative component.

The algorithm starts by identifying all circuit sub-sets that incorporate the same topological features, like circuits with complementary inputs, circuits with folded cascode outputs, circuits with common mode feedback, and so on. This step is achieved by computing all maximal circuit clusters K_i based on the topological similarity of the circuits in a cluster, e.g., the number of structures shared by the circuits. Cluster maximality indicates that a cluster is not obtained in another cluster. Next, the outermost for-loop creates bottom-up the abstraction structure for each cluster K_i. The process first identifies set P_i of the most topologically-similar instances in set K_i (either circuits or abstractions), and then creates by calling procedure *Create abstraction* a new abstraction (C_a) that describes the instances. Abstraction C_a is added to the associative component KnowStr, and further considered to produce more abstractions.

After creating the abstraction structure for cluster K_i, the algorithm adds additional links to express feature combinations between the abstractions corresponding to the current cluster (denoted as abstraction M) and the abstractions of another cluster (described as abstraction Q). Procedure *Abstr&Par* checks that abstraction Q relates to a different cluster K_j and is the most abstract in that cluster to include the specific topological feature f_i. A link is added between abstraction Q and M.

Constructing abstraction for a set of instances (procedure *Create abstraction*) requires computing sets I, U, and E of the abstraction based on sets I, U, and E of the instances. The algorithm is described in [55]. These sets

are calculated using the following four sets $I_1 - I_4$:

• Set I_1 is the set of matched nodes of the instances. Two nodes in different circuits are matched, if they present the same electrical behavior within the two circuits, e.g., the symbolic transfer functions from inputs to the node and from the node to outputs are the same for the two nodes. Matched nodes are found using the method in [56].

• Set I_2 includes the arcs between matched nodes. Their identification is obvious starting from the matched nodes in set I_1.

• Set I_3 and I_4 express the alternative features of the instances. While each alternative is specific to some but not all instances, ignoring to include a description of the alternative features can result in abstractions that do not describe valid design solutions or do not express the performance achieved by its instances.

• Set I_4 includes the nodes labeled with star (nodes that are unique to an instance) and the arcs of the alternatives (arcs shown with dashed line). The information in Set I_4 about alternatives can be used to generate induction patterns that are used to create more instances of the same kind [55].

Set U of the abstraction includes the unique features of the abstraction as computed to the other abstractions that share with it the same parent in the knowledge representation. Set $I = I_1 \cup I_2 \cup I_3 \cup I_4 - U$.

Joined symbolic expressions include all similar parameters and operators in two symbolic expressions while new parameters represent the unmatched parameters and operators. The algorithm to compute the joined symbolic expressions for set I_3 was introduced in [55]. It finds the maximum associations (similarities) between the symbolic expressions of the paths connecting two pairs of matched nodes in two circuits and describes dissimilar terms by symbols that unify these terms.

```
Result: Create associative component
input  : circuit topologies C_1, C_2,..., C_n;
output: KnowStr (associative component);
begin;
KnowStr= Φ;
cluster circuit topologies into clusters K_i based on
 their similarity;
for each cluster K_i do
    while more clustering is possible for K_i do
        find subset P_i in K_i with highest topological
         similarity;
        Abi=Create abstraction (P_i);
        K_i=K_i ∪ Abi − P_i;
        KnowStr=KnowStr ∪ Abi with P_i as
         descendants;
    end
    for all features f_i of abstraction M in KnowStr do
        if feature f_i is present in Abstr & Par
         (abstraction Q) then
            create link from abstraction Q to
             abstraction M to indicate feature
             combination;
        end
    end
end
end procedure;
```

Algorithm 1: Algorithm to construct the associative component

Example: Given two circuits C_1 and C_2, let's assume that the matched nodes P and Q of the two circuits are connected through paths through unmatched nodes. If p= $(x_1+x_2)x_3+x_1+x_2+x_3$ is the symbolic expression of the path in circuit C_1 and q=$(x_1+x_5)x_3+x_6$ is the symbolic expression of the path in circuit C_2, then the abstraction includes an arc between nodes P and Q labeled with the following joined expression, $(x_1+\alpha)x_3+\beta$, where parameter α unifies parameters x_2 and x_5, and symbol β unifies $x_1+x_2+x_3$ and x_6. The domain of α is the intersection $dom(x_2) \cap dom(x_5)$ and the domain of *beta* is the intersection $dom(x_1+x_2+x_3) \cap dom(x_6)$. *dom* represents the domain of a parameter or expression.

2.6.2 Mining performance capabilities component

Algorithm 2, 3, 4 presents three algorithms to calculate Performance capabilities component. The algorithms assume that the considered trade-offs are relevant for the given problem description, otherwise they would have been pruned in a preliminary step.

The first algorithm computes the set S of similar trade-offs for two circuits C_1 and C_2. It considers all pairs of rows in the two trade-off tables, and groups them if their attributes present similar variations and their controlling parameters represent two situations: (i) all controlling parameters correspond to matched nodes in the two circuits, therefore, the nodes present similar electrical behavior, and (ii) all controlling parameters have similar natures, like trans-conductance (g_{mg}, g_{md}, g_{ms}) or capacitance (C_{gd}, C_{db}, C_{gs}). This guarantees that the parameters have similar value ranges, hence generate similar attribute variations in the two circuits.

The second algorithm implements the trade-off merging procedure. It considers two trade-offs (e.g., two rows of the trade-off tables of two circuits), and returns the merged trade-off. First, the similar controlling parameters of the two trade-offs are identified and grouped together. Any dissimilar parameters (e.g., parameters that are unique to a circuit) are grouped with a

null parameter (−) to indicate that the parameter is specific to an alternative (null parameters indicates the absence of a parameter that would correspond to the parameters of another circuit). The grouped parameters become the controlling parameters of the merged trade-off. The similar attributes are added (without change) to the merged trade-off. The dissimilar attributes are combined as presented in the definition for merging two trade-offs (Section 2.4).

Result: Find similar trade-offs
input : circuit topologies C_1, C_2, trade-off tables Tab_1, Tab_2;
output: set S;
begin;
S= Φ;
for *all pairs* T_1 *in* Tab_1 *and* T_2 *in* Tab_2 *are not yet similar* **do**
  if T_1 *and* T_2 *are similar and have same number of parameters correspond to matched nodes in* C_1 *and* C_2 **then**
    S=S $\cup$ {T_1, T_2};
  end
  if T_1 *and* T_2 *are similar and their controlling parameters are of the same kind* **then**
    S=S $\cup$ {T_1, T_2};
  end
end
end procedure;

Algorithm 2: Algorithm for constructing performance capabilities (a)

Result: Merge two trade-offs
input : T_1, T_2;
output: merged trade-off;
begin;
controlling parameters = similar controlling parameters and group them;
add similar attributes to the merged trade-off;
combine the dissimilar attributes according to the definition for trade-off merging;
return merged trade-off ;
end procedure;

Algorithm 3: Algorithm for constructing performance capabilities (b)

Result: Combine trade-off tables
input : circuit topologies C_1, C_2, trade-off tables Tab_1, Tab_2;
output: New table;
begin;
New Table = Find similar trade-offs (C_1, C_2, Tab_1, Tab_2);
Group the remaining trade-offs based on their highest similarity;
for *all pairs (T_1, T_2) of grouped trade-offs* **do**
| New Table=merge two trade-offs (T_1, T_2);
end
end procedure;

Algorithm 4: Algorithm for constructing performance capabilities (c)

Finally, the third algorithm computes the new trade-off table (*New Table*) that merges trade-off tables Tab_1 and Tab_2 of two circuits. First, it adds the similar trade-offs to the new table. Next, the remaining trade-off rows are grouped together, if they are reasonably similar (e.g., the number of similar attributes exceeds a threshold value). Dissimilar trade-offs (which are unique to a circuit) are grouped with a null trade-off (e.g., a trade-off without controlling parameters or any attribute variations) to indicate that the trade-off does not have a similar trade-off in the other circuit. Finally, the grouped trade-offs are merged together and added as rows to trade-off table *New Table*.

2.6.3 Mining causal reasoning component

A. Identifying starting ideas. Starting ideas are the design features and constraints (e.g., triplets $\langle DF, Des, CP \rangle$ in Section 2.5) of the design steps that originate the design sequence that creates a given circuit. Note that the very initial idea that might have jump-started the creation of a new circuit might be different from starting ideas. The very initial ideas do not necessarily lead to the final solution. Instead, the insight gained during an iterative, reasoning-based design process transform and evolve the very initial ideas into the starting ideas of the design sequence that produces the circuit design.

Algorithm 5 shows the algorithm to mine starting ideas for a given circuit C. Inputs are the circuit topology and the associative component $KnowStr$ built for a sufficiently large set of analog circuits, including previous designs by the authors of circuit C and related circuits such as the ones cited in the proposal. First, the algorithm computes sets Σ, Γ, and Λ for circuit C using the associative component $KnowStr$. set S is initialized to the union of the sets Γ and Λ as they are obviously starting ideas in producing circuit C. However, the starting ideas in sets Ψ and Θ are still unknown, and are identified next.

There are starting ideas in sets Ψ and Θ only if the set of all features (set Σ) is not equal to the reunion of the current set S (the starting ideas in $\Gamma \cup \Lambda$) and the features of the design steps that can be justified starting from the features in set S (set Q). The latter features are computed by procedure *Find design sequence*, which is described next. Then, for all remaining features x (the features of set $\Sigma - (Q \cup S)$), the algorithm finds the features in set Σ that can be justified by adding x to the set of starting ideas S. At the end of the for-loop, set Q includes all features in set Σ that can be justified by other features in set Σ. Thus, the features remaining in set $\Sigma - (Q \cup S)$ must be added to set S too. The last for-loop considers all features x in set $\Sigma - (Q \cup S)$ and adds them either to set Θ, if according to the associative component they represent common knowledge, or otherwise to set Ψ, hence indicating that these features are new design insight by the authors.

The first for-loop considers the more concrete features before analyzing more abstract features. This is because circuit presentations in publications arguably offer a more implementation-oriented description, in which physical details are presented, analyzed, and measured, while more abstract features are usually less emphasized. This presentation style is reasonable as abstract ideas offer a qualitative argument, which however can be secondary in the presentation.

B. Mining Design Sequence. The algorithm for mining design sequence identifies the design steps and their ordering that can be justified starting from set S of starting ideas, a given circuit C, and the associative component $KnowStr$. Each design step, defined as in equation (5), introduces features present in set Σ (all features) of circuit C, and is justified (hence, reduces the distance between problem requirements and solution performance or relaxes the design constraints). If set S includes all starting ideas of the circuit then the identified sequence must include all features in set $\Sigma - S$, hence it produces the final design. The algorithm is shown in algorithm 6.

Result: Find starting ideas
input : circuit topology C, associative component KnowStr;
output: sets S, Γ, Λ, Θ, Ψ;
begin;
compute Σ, Γ, Λ ;
S=Γ ∪ Λ ;
Q=Find design sequence (C, KnowStr, S) ;
if $\Sigma \neq Q \cup S$ **then**
  for *all features x in* $\Sigma-(Q \cup S)$*, from more concrete to more abstract features* **do**
    Q=Q ∪ Find design sequence (C, KnowStr, Q ∪ S);
  end
  S=S ∪ (Σ−Q);
  for *all features x in* $\Sigma-(Q \cup S)$ **do**
    if *features x is common* **then**
      Θ=Θ ∪ x;
    else
      Ψ=Ψ ∪ x;
    end
  end
else
  Θ, Ψ = Φ ;
end
end procedure;

Algorithm 5: Algorithm for finding starting ideas in a circuit design

Result: Find design sequence
input : circuit topology C, associative component KnowStr, starting ideas S;
output: Seq;
begin;
TempSeq=Φ;
T=Σ−S;
while *Seq changes or TempSeq changes* **do**
  for *all features x in T, in decreasing order of their frequency* **do**
    P=feature in S s.t. it justifies x;
    create design step N by associating feature x to P;
    T=T−x;
  end
end
end procedure;

Algorithm 6: Algorithm for finding design sequence in a circuit design

The algorithm terminates if no more design steps are added to the ordered sequence Seq and TempSeq. The for-loop considers the features x in set T (Σ − S), starting first with the more likely features (features that have been used more in previous designs). A new design step N is created to reflect the changes in the design introduced by adding features x as well as the modification in performance and design constraints. If the new step is not justified then it is added to sequence *TempSeq* of temporary steps. This sequence includes steps that individually are not justified yet, but can become justified later as a group.

If design step N is justified, the algorithm checks first if sequence TempSeq is not empty, hence indicating that a number of previous features have en-

abled the current feature x, even though the features were not justified when considered individually. Design step N is merged with the design steps in sequence $TempSeq$, thus producing a new step that simultaneously introduces multiple features. Design step N is appended at the end of the overall design sequence Seq.

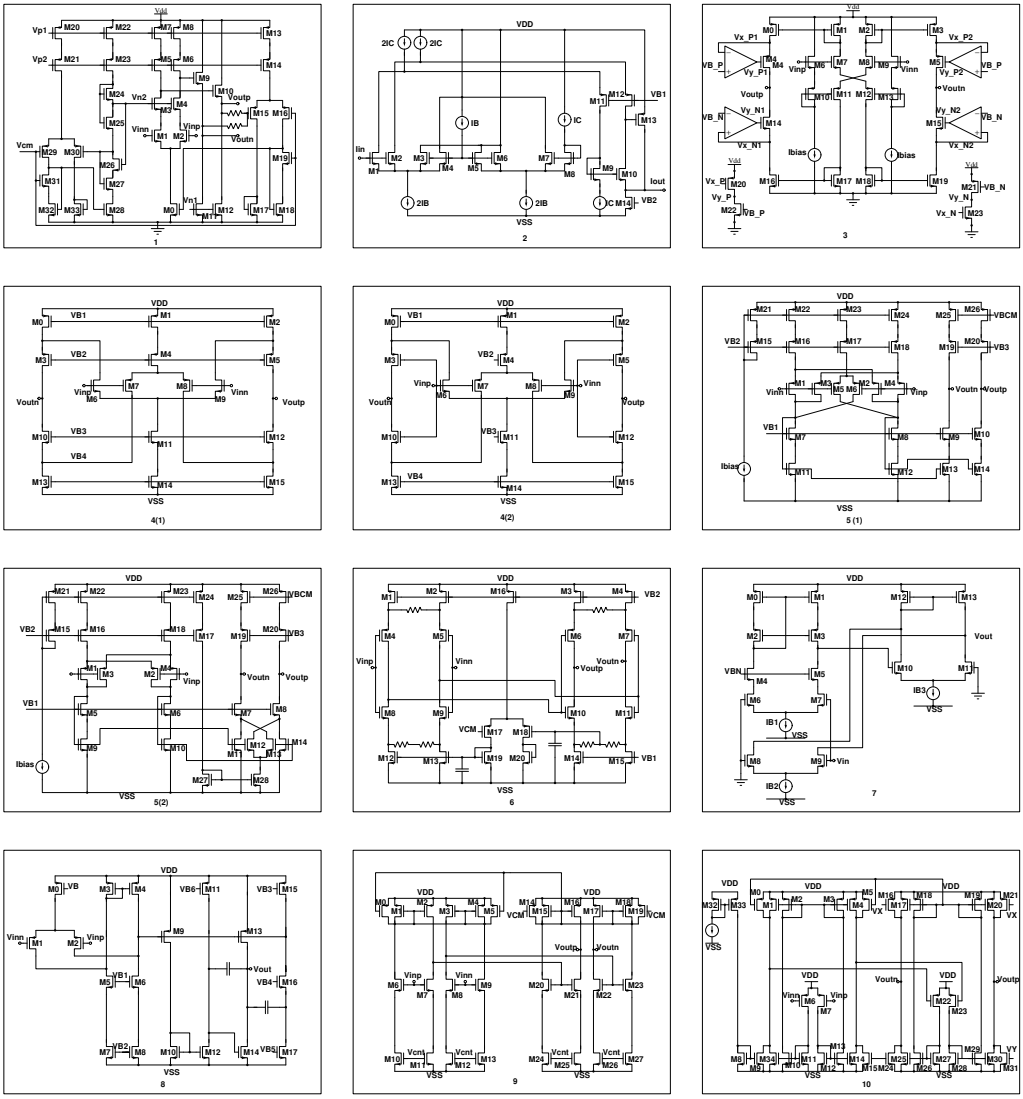


Figure 2.19: Circuits index 1-10 schematics

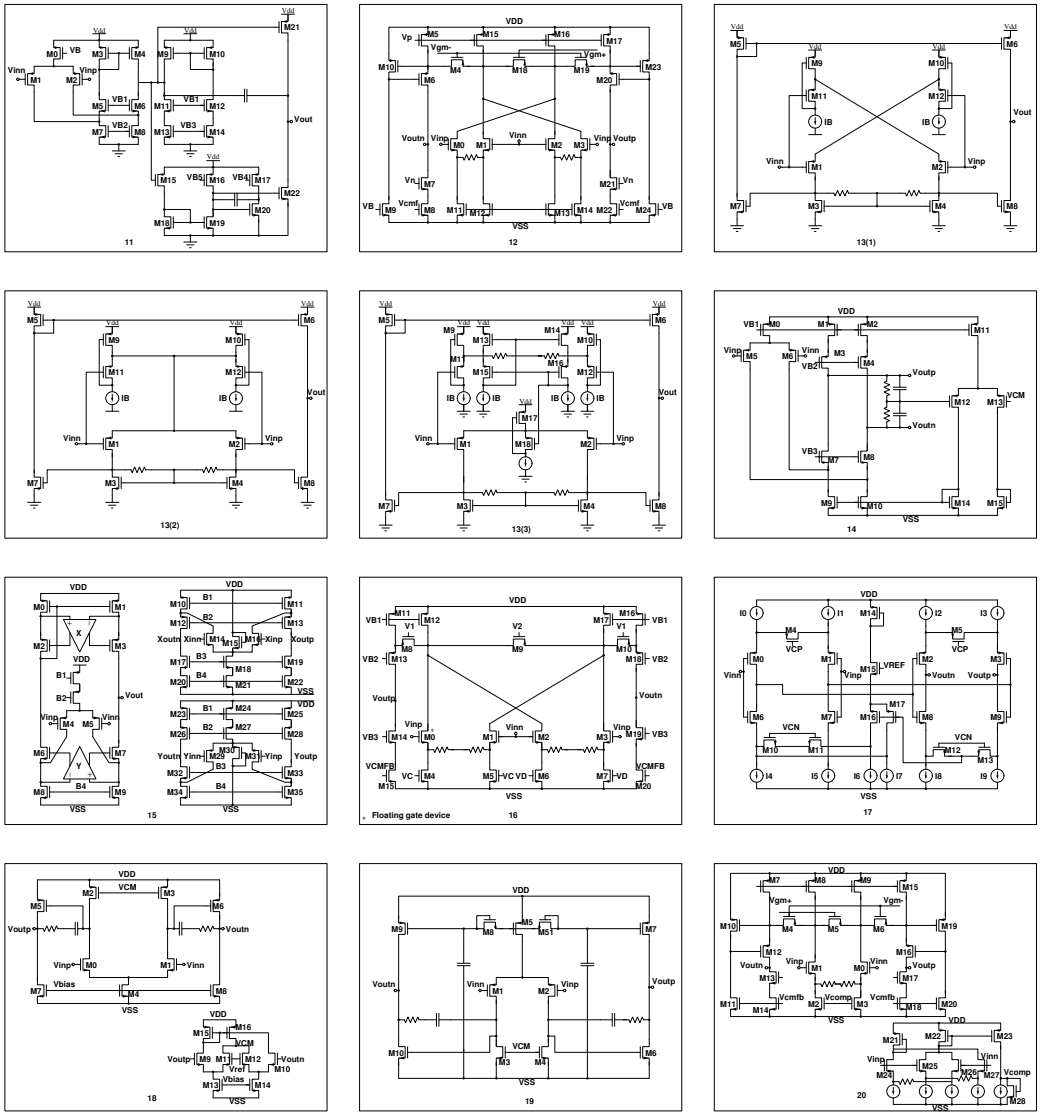


Figure 2.20: Circuits index 11-20 schematics

2.7 Experiments

Experiments considered a set of 34 modern op-amp/OTA circuits for high-frequency applications. Figure 2.19, 2.20, 2.21 show the circuit schematics. Table 2.17 enumerates the circuit topological features library that we con-

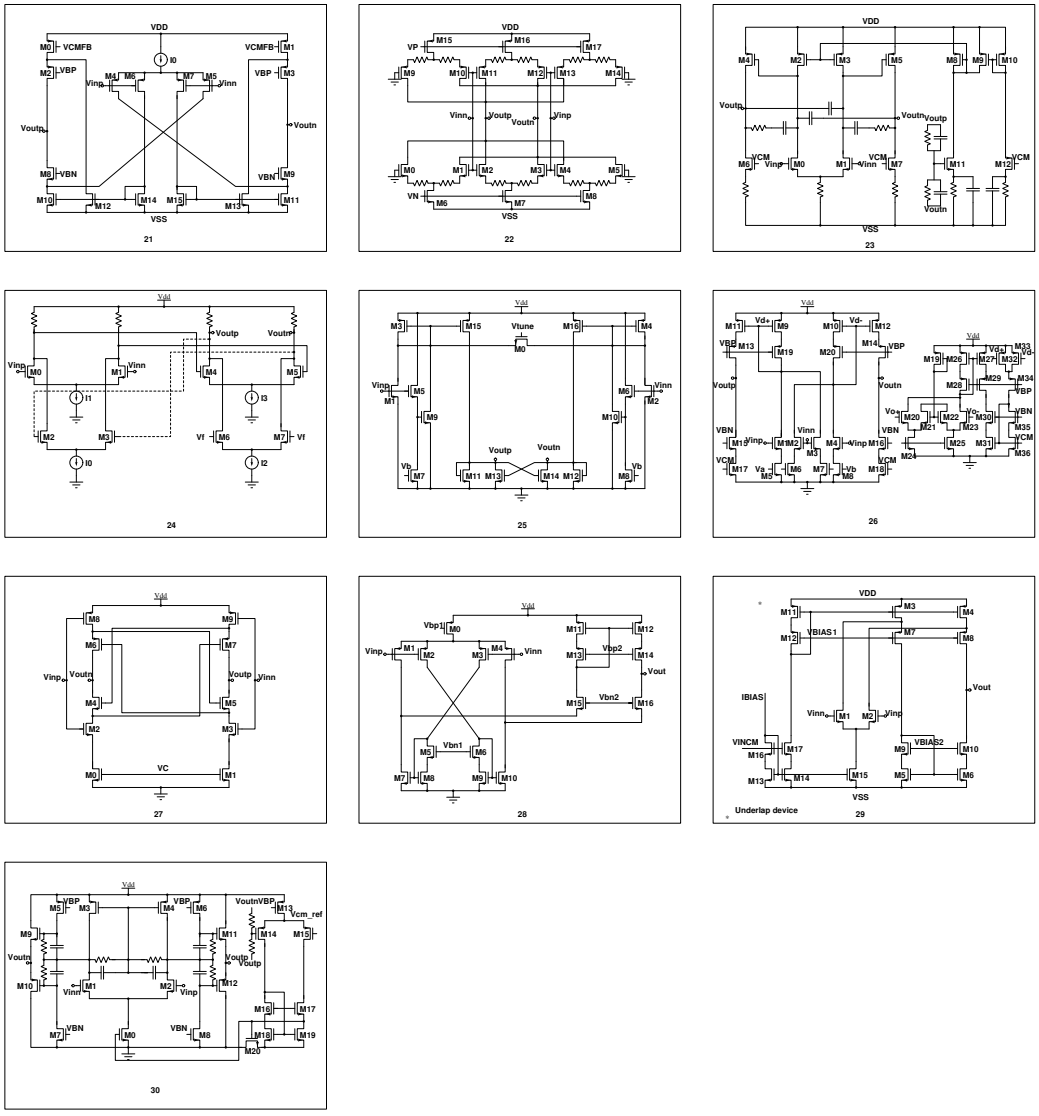


Figure 2.21: Circuits index 21-30 schematics

sidered and their abbreviations. Knowledge mining on 34 op-amps/OTAs to build associative component, performance capabilities component, causal reasoning component are summarized next.

Associative component. Figure 2.22 shows a fragment of the entire associative component. Circuit indexes are referred to Figure 2.19, 2.20, 2.21.

Table 2.17: Circuit features and their abbreviations

Abr.	**Features**	**Abr.**	**Features**
DO	differential output	DI	differential input
PDI	pseudo-differential input	DDI	double differential input
SO	single output	CO	cascode output
AB	class AB	2x	two copies of input
TT	trans-conductance tunning for PPV	MC-I	mobility compensation
FC	folded cascode	CCO	cross-coupled output
SD	source degeneration	CMFF	common mode feed-forward
CMFB	common mode feedback	FFC	feed-forward compensation
FFR	feed-forward regulation	ICS	ideal current source
BCS	basic current source	C3OTA	3-path OTA
FSym	fully symmetric structure	FBal	fully balanced
HRO	high-resistance output	CpSti	coupling structures
CM	current mirror	CS	current source
LCM	low-voltage current mirror	SCM	simple current mirror
VI-CO	voltage in, current out	CI-CO (CF)	current in, current out
RCY	recycling	FBop-amp	op-amp in feedback
SI	single input	CF	current mode
OS	common source output	NO	no separate output stage
MP	multi-path	MC	miller compensation
Tele	telescopic cascode	TS	three stage
AFFC	active feed-forward compensation	FC	folded cascode
r-biasing	replica biasing	DFCB	damping factor control block
CCI	cross-coupled input	a-biasing	adaptive biasing
CCFB	cross-coupled floating batteries	PP	pseudo-differential pair
LCMFB	local CMFB	WTA	winner take all
DO	differential output	AB	class AB
ABI	class AB input	ABIAS	adaptive biasing
DP	dual path	LCMFB	local common mode feedback
CMFB	common mode feedback	ABO	class AB output
D-D	drain to drain	P-P	push-pull
SS	single stage	MS	multiple stage
GBCA	gain-boosted cascode amplifier	AFFC	frequency compensation

The top node describes the abstraction with a number of differential inputs (indicated as DI+) and single (SO) or differential outputs (DO). For example, circuit 11 has a pair of DIs and circuit 2 includes four DIs. The abstraction has five descendants based on the nature of the output stage: circuit with folded cascode outputs (FC), circuits with cascode outputs (CO), circuits with class AB outputs (AB), circuits with only one stage (no separate out-

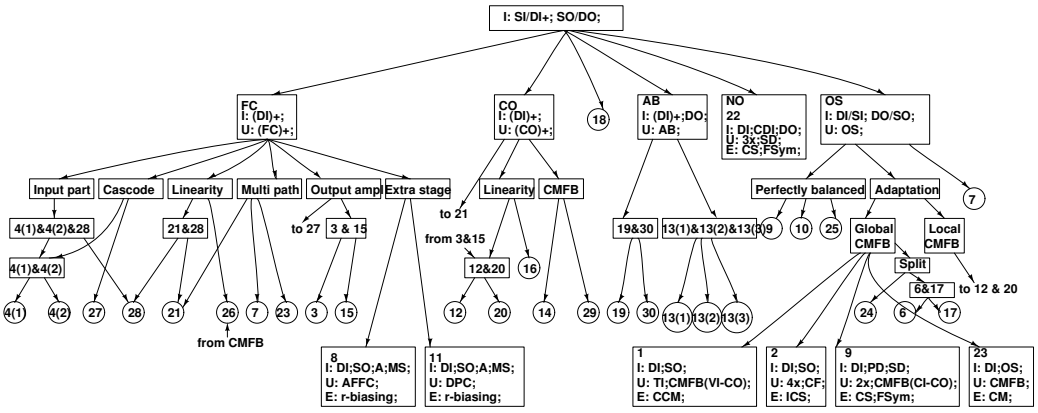


Figure 2.22: Fragment of the associative component for the circuits set

put stage NO), and circuits with common gate output stages (OS). Each of the five abstractions represents different circuits that implement specific output stages. In addition, circuit 18 presents an abstraction for an universal op-amp, which as the top abstraction, includes a number of DIs and a common output stage [57]. Input/output stage plays a critical role in amplifier design, thus is used as the splitting attribute. Other splitting criteria is also feasible, i.e., number of stages. The quality of splitting depends on the level of hierarchy (tree depth). Empirically, we would like to have level of hierarchy less than 5 to reduce the problem complexity.

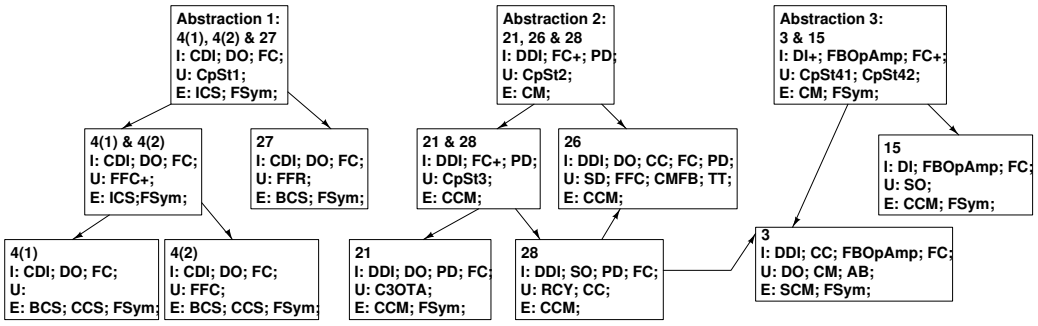


Figure 2.23: Associative representation for circuits 4(1), 4(2), 27, 21, 26, 28, 3, 15

Each of the top five abstractions further includes more specific abstractions. For example, the circuits in abstraction FC are grouped using various criteria, such as the means of achieving the desired problem requirements of high gain and bandwidth, and the topological similarity of the features. Abstraction *Input part* includes circuits that offer a variety of ways of implementing differential inputs (DI+) as a way of achieving high gain, high linearity requirements. Abstraction *Linearity* refers with designs with cross-coupled input to improve high linearity. Abstraction *Multi path* incorporates designs with multiple signal paths from input to output to achieve the performance requirements. Abstraction *Output ampl* uses local feedback at the output stages to boost gain. abstraction *Extra stage* represents circuits that have extra structures connected to the folded cascode structure of the circuits. Note that each of these abstractions represents a different abstract strategy to address the problem requirements. Figure 2.23 shows the detailed abstraction built for circuits with complementary differential inputs pairs (CDI) and folded cascode (FC) differential outputs (DO) devised using fully symmetric structures (FSym).

There is some similarity between knowledge representation (like Figure 2.22) and MOJITO's hierarchical tree of building blocks [35]. However, the mined associative component is constructed using symbolic models and the complete knowledge mining also includes performance capabilities, causal reasoning component.

Performance capabilities component. From associative component, topological features of circuit 4(1), 4(2), 27 are extracted and built into abstraction. Trade-off tables of circuit 4(1), 4(2), 27 and their abstractions are explained in Table 2.1, 2.2, 2.4, 2.3, 2.5. The sensitivity effect of circuit parameters on performance attributes, trade-off effect between performance attributes are modeled as arrows in the trade-off table. Symbolic modeling considers parameter ranges around local design point, which limits the exploration of global design space. Beside, symbolic model constraints per-

formance attributes to be linear or weak non-linear. Chapter 5 proposes more comprehensive causal information modeling algorithms to extract parameter effects and build Pareto front out of circuit.

Table 2.18: Causal reasoning component for the circuits in Figure 2.19

Cir.[Citation]	Starting Ideas						Design Sequence		
	Type	#	Γ	Λ	Ψ	Θ	#	Mapping	Justification
1 [58]	1	5		1	1	3	2	Λ, Ψ	CM, match, stabil.
2 [59]	2,4	2			2		3	Ψ	BW, bias, volt follow.
3 [60]	2	4		3	1		3	Λ	gain, pow.
4(1) [2]	1	3		2	1		1	Λ	bias
4(2) [2]	1	2		2			1	Λ	bias
5(1) [50]	1	4	3	1		3		Λ	gain, PSRR, lin.
5(2) [50]	1	2		1	1		2	Λ	gain, PSRR, lin.
6 [61]	1	4		2	1	1	1	Ψ	CM
7 [62]	2,4	3		1	1	1	2	Λ	BW, gain
8 [63]	2,4	4		3	1		4	Λ, Θ	BW, gain, bias
9 [64]	3	3		2	1			Λ, Θ	CM, match
10 [65]	2,4	3		2	1		2	Λ, Ψ	CM, match

Table 2.19: Causal reasoning component for the circuits in Figure 2.20

Cir.[Citation]	Starting Ideas						Design Sequence		
	Type	#	Γ	Λ	Ψ	Θ	#	Mapping	Justification
11 [66]	4	2	1	1			2	Γ, Λ	BW
12 [67]	1	4		3	1		2	Λ	gain
13(1) [7]	1	2		1	1		2	Λ	pow.
13(2) [7]	1	2		1	1		2	Λ	pow.
13(3) [7]	1	2		1	1		2	Λ	pow.
14 [68]	1	2			1	1	2	Θ	speed
15 [69]	1	1	1				1	Γ	gain, speed
16 [70]	1	8	2	2	1	3	4	Λ, Θ	lin, gain, bias, Gm tun.
17 [71]	1	4	2		1	1	2	Λ	CM
18 [57]	2	2	1			1	1	Λ	Universal
19 [72]	1	3	1	1		1	2	Γ, Λ, Θ	SR, BW
20 [73]	1	5		4	1		3	Λ	lin, gain, bias

Table 2.20: Causal reasoning component for the circuits in Figure 2.21

Cir.[Citation]	Starting Ideas						Design Sequence		
	Type	#	Γ	Λ	Ψ	Θ	#	Mapping	Justification
21 [51]	2	5		2	1	2	5	Λ, Ψ	gain, SR
22 [74]	2	3		2	1		2	Λ	lin., gain
23 [75]	1	2		1	1		2	Λ	BW
24 [76]	1	3		2	1		1	Λ	BW
25 [77]	5	3			1	2	5	Θ	lin., gain
26 [53]	2	4	1	1	2		4	Γ, Λ, Θ	lin., CM, gain, Gm tun.
27 [78]	5	2			2		2	Ψ	BW, bias
28 [54]	1	3	2			1	3	Γ, Θ	gain, SR, match
29 [79]	1	1			1		1	Λ	gain, BW
30 [80]	1	3			1	2	3	Ψ, Θ	BW, pow.

Causal reasoning component. Starting ideas and design steps of circuits' design reasoning are extracted individually. Table 2.18, 2.19, 2.20 summarize the causal reasoning component for the entire circuit set. The first column indicates circuit index, column 2 is the type index how starting ideas are combined, column 3 shows the total number of topological features in starting ideas, column 4-7 give the number of starting ideas belong to sets Γ, Δ, Ψ, Θ based on their origins. Column 8 offers the number of steps in the design sequence, column 9 gives the kind of starting ideas to which the design sequence are mapped to, and column 10 indicates the justification of the design sequence.

By studying the design reasoning written in papers, we observe that starting ideas are obtained from a few types of combinations. Thus, we classify starting ideas into five types based on how they are combined:

- *Type 1*: combine topological features.
- *Type 2*: combine topological and abstract features.
- *Type 3*: combine abstract features.
- *Type 4*: not using certain feature.

- *Type 5*: novel abstract/topological features used for other purposes.

Column 2 in Table 2.18, 2.19, 2.20 summarizes the starting idea type for each circuit. Most starting ideas are either of Type 1 (21 ideas), or of Type 2 (9 ideas). Ideas of Type 4 are less frequent (5 ideas) suggesting that it is less common to start from ideas that require not to incorporate specific features. Similarly, in only two cases, starting ideas are of Type 5, e.g., the designer started by creating an abstraction for existing physical features, e.g., a dual path op-amp was generalized into a multi-path (i.e., three path) circuit. One design included starting ideas of Type 3, combination of two or more abstract features.

The starting ideas are also categorized into different sets, Γ (used by the same group), Λ (used by the other group), Ψ (new design insight), Θ (common set of features). Out of total 101 starting ideas, only 14 ideas were previously used by the authors in their designs (set Γ), and 22 ideas represented common analog circuit design knowledge (set Θ). Hence, ideas in sets Γ and Θ account for about one third of total ideas. Most ideas (42 ideas) come from related work (set Λ) and new insight acquired by designers (29 ideas in set Ψ). For example, circuit 9 proposes a new feed-forward method to avoid using miller capacitor in high frequency circuits. Circuits 10 and 13 use active feedback stage compensation. Circuit 21 is a three-path OTA realizing the idea of multi-path op-amp. While a design might incorporate multiple starting ideas in sets Γ, Λ, and Θ, they use in general only one insight-related feature (set Ψ).

Design sequence include up to five steps, which are often (25 circuits) linked to ideas from related work (set Λ). Some of the design steps are mapped to starting ideas if they are implementations of the starting ideas. Design sequence are moderately often connected (10 circuits) to starting ideas expressing new insight (set Ψ). Only rarely, design sequence correspond to previously used features by the same group in set Γ (5 circuits) or to common design features in set Θ (6 circuits).

2.8 Conclusions

Knowledge mining technique is beyond the capability of data mining by interpreting and representing the knowledge space. This chapter proposes analog circuit design knowledge mining from literature. It contributes a new perspective to design automation compared to the traditional optimization-based, equation solving-based approaches. The proposed mined knowledge mainly focuses on op-amp/OTA design, whereas the methodology is suitable for other analog blocks at the same level or higher (e.g., LNA, filter, oscillator) in general. Analog blocks tend to use repetitive building blocks or have topological patterns, which is an important character to build abstraction. With analog designer's effort moving toward more complicated system level design nowadays, knowledge mining technique helps to ease analog design regarding basic analog blocks (i.e., op-amp/OTA). It can also be used to understand better the current topology trend, devise new design solutions and identify design innovations.

This chapter explains mining technique and algorithm to build analog circuit design knowledge representation. The proposed mined knowledge includes three components: an associative component presenting a hierarchy of the considered circuits, a component expressing the performance capabilities (e.g., trade-offs and bottlenecks) of the circuits, and a causal reasoning component describing the most likely starting ideas and design plans used to create a circuit. The associative component groups circuits into hierarchical descriptions based on the symbolic similarities of the instances. Starting from the performance trade-offs and bottlenecks of the circuits, performance capabilities express the trade-offs of abstractions by combining the trade-off tables of their instances. Finally, the causal reasoning component identifies the starting ideas that support finding sequence of justified design steps for a given design. The mined knowledge can be used to tackle new applications (e.g., by selecting or refining a circuit topology), identify new design opportunities (by analyzing the combination of design features that have never been

used together before), and validating design correctness by showing that all steps in a design sequence are justified. Extended applications (reasoning-based topology synthesis, design verification) are presented in Chapter 3, 5.

Algorithms on mining associative component, performance capabilities, causal reasoning component, are presented. Experiment shows knowledge representation of 34 high performance op-amps /OTAs from analog circuit design literature. In some cases, the found starting ideas were probably the initial ideas of the designer, e.g., when topological features were combined to create a new circuit. However, other starting ideas are only the originators of the design sequence that create a circuit, but not the actual initial ideas of the designer.

An essential aspect of devising a representation to support reasoning-based analog circuit synthesis is explicitly presenting the causal information on how topological features and their parameters decide performance trade-offs and bottlenecks. Structural (white box) symbolic models offer a causal link between circuit structure, nodal behavior, and performance. Also, symbolic expressions are effective in comprehensively representing the similarity and differences among circuits, including the construction of abstractions to describe a group of circuits. The main limitations of symbolic methods are their difficulties to handle strong non-linearity and complex circuits. Also they express circuit parameter/performance ranges constrained in local space.

Building association and extracting design reasoning are generally technology independent, whereas feasibility of topology can be a technology dependent decision. When new technology emerges, topological features need to be updated with new design constraints (e.g., supply voltage, transistor operation region). Simulation needs to be conducted individually with new design constraints involved. Various technologies are not explored in the thesis and relatively heavy simulation cost can be one limitation too.

Chapter 3

Reasoning-based Topology Synthesis[1]

3.1 Introduction

Topology synthesis refers to the method to create new circuit topologies for a set of performance requirements and constraints. Analog circuit topology has been difficult to synthesize as it requires searching an open-ended, widely extensible, and strongly discontinuous solution space. For example, inventing new topological features changes the types of the solution space because these features can be utilized to produce more circuits, which otherwise are hard to abstract. While there has been significant advancing in synthesis tools for layout design and circuit sizing, automatically devising or refining circuit topologies (schematics) remains difficult. Circuit sizing and layout design are often tightly coupled to topology synthesis as incremental modifications in the schematics can simplify transistor sizing and layout design.

From the perspective of how a topology is provided for the next synthesis step, topology synthesis methods can be summarized by topology selection,

[1] [81], [82]

topology refinement, and topology generation [6]. Topology selection relies on a library of well-defined circuit topologies and expert designer to do the selection. Topology refinement has been explored with different attempts on modifying knowledge-base templates. Topology generation evolves genetic algorithms to create circuit from basic building blocks. Existing methods are limited by designers' effort on decision making, and the performance aid of simple building blocks.

Based on the proposed design knowledge mining techniques, analog circuit topology synthesis through a reasoning-based flow is developed to extend existing work. The proposed design knowledge mining provides a mechanism to structure the solution space, reuse previous design results, and prune infeasible or less-optimal regions. More specifically, the associative component serves as a mechanism to structure the solution space by highlighting feature similarities and differences among abstractions. Abstractions partition the solution space into sub-regions, which helps prune less optimal topology options. The performance capabilities indicate each topology's parameter sensitivity, performance trade-off, and bottlenecks. Moreover, design sequence from the causal reasoning component indicate groups of features that should be utilized together to tackle certain performance requirements as well as the conditions (constraints) under which the features can be used effectively. Hence, the causal component helps reusing previous design results while eliminating infeasible feature combinations.

With the analog circuit design knowledge mining techniques and experimental results in Chapter 2, this chapter proposes a knowledge-intensive, reasoning-based approach to create analog circuit topologies for emergent and innovative applications, e.g., problems that involve tackling of novel performance trade-offs and bottlenecks. This procedure attempts to mimic human reasoning, even though at a very simple level [83], [84]. We are targeting at creating op-amp/OTA topologies that are more similar to a human design.

Reading and summarizing design documents reveals the ideas that trigger a design process can be originated in five ways: Type 1 combines topological features from different circuits. Type 2 combines topological and abstract topological features. Type 3 combines only abstract features. Type 4 is not using certain feature, and Type 5 is novel topological/abstract feature used for other purposes. Abstract topological features refer to the abstraction elements in associative component, thus they can be implemented by various topological features in the lower level. The thesis discusses in details five different reasoning-based topology synthesis flows depending on the different combination types of starting ideas. New circuit topologies are created by following different synthesis flows. In the experiment, the thesis offers four novel op-amp/OTA designs following different types of synthesis flows. The flow utilizes an expandable design knowledge representation in Chapter 2 that stores circuit topological features at various levels of abstraction and their performance trade-off.

The chapter has the following structure. Section 3.2 summarizes related work. Section 3.3 presents the overview of the synthesis methodology. Section 3.4 proposes synthesis algorithms. Examples are offered in Section 3.5. Conclusions end the chapter.

3.2 Related work

Existing topological synthesis approaches in design automation tackle the problem from three aspects, topology selection, topology refinement, topology creation. The first approach is to automate topology selection from a library of predefined topologies or circuit structures based on if-then rules [85], [86], [87]. OASYS tool [85], represents circuit topologies as a hierarchy of templates of abstract functional blocks, each with associated detailed design knowledge. Topology is selected hierarchically translating performance specifications from top level to lower level. [8] adds a high-speed

comparator design style to the OASYS tool. [86] utilizes stored knowledge of formal mathematical techniques, intuitive reasoning procedures. It relies heavily on the mature analog design expertise. If performance trade-offs are complex, it is difficult to devise effective circuit selection rules.

Topology refinement aims to modify an existing structure with alternative building blocks to improve the performance of current topology, through geometric programming [88], or topological feature extraction [47], [36]. Different than topology selection or creation methods, this method treats analog synthesis as a decision making process instead of black box. This chapter extends previous work in [47] by automating knowledge base, deriving better feature selection rules by causal reasoning approach.

Topology creation uses genetic (GA) or evolutionary algorithms to create structures by interconnecting CMOS devices or simple sub-structures [89], [35]. The well-known operators are selection, mutation, and combination (and their extensions), which are sometimes extended with analog design related steps or constraints. [89] first builds topology up from basic building blocks by genetic programming. Later systems are by Lohn et al. [90], and the WYWIWYG system [91]. [92] improves the system by current flow analysis to prune faulty components. Design inspired constraints are embedded into the algorithms to limit the evolutionary process and increase the likelihood of producing effective structures [35]. While this approach can, in theory, evolve any circuit topology, in reality, creating performance-efficient yet minimal structures is hard as it involves searching an open-ended, widely extensible, and strongly discontinuous solution space. Moreover, the repeated applying of the three operators can result in very complex structures, which are less common to the topologies that a designer devises through knowledge-intensive reasoning. The synthesized circuit topologies include unique features but such features are rarely used by designers.

A higher level of synthesis problem deals with modulators, converters, etc. A library of analog cells (op-amps, comparators, Mux) and a set of transfor-

mation rules are predefined to convert a signal-flow graph expressed in a language like VHDL-AMS into an implementation [93]. [94] synthesizes CMOS and BICMOS analog circuit for specified performance constraints. The input is modeling languages of hierarchical circuit description, while the drawback is that library circuit topologies and layout styles are non-hierarchical. Another approach describes a class of circuit topologies as a template with all possible feed-forward and feedback signal paths, and then uses the template to decide which of the paths should be used in an implementation. It uses the set of building blocks and connection rules to generate different topologies but which all realize the same signal flow as the given template. The two latter approaches are limited to structured systems, which follow systematic signal flows, like state-space filters and $\Delta\Sigma$ ADC [95], [96]. However, less-systematic structures, like op-amps or OTAs, are hard to synthesize.

Above library-based or genetic algorithms cannot generate topologies beyond a constrained set of structures, or experience difficulties in evolving performance-effective yet manual design like circuits. This chapter proposes reasoning based method, inspired by the mined causal reasoning from human design process. The method develops different strategies to select and combine justified topological features from current design literature. It is able to keep updated with current innovative solutions, and result topology similar to human design.

3.3 Methodology for reasoning-based topology synthesis

Figure 3.1 presents an overview of the proposed reasoning-based methodology for synthesis of analog circuit topologies. The methodology includes two main steps: (*i*) a strategy to select starting ideas from the associative component of the domain knowledge representation and (*ii*) a reasoning-based procedure to use the starting ideas in creating design sequence that is

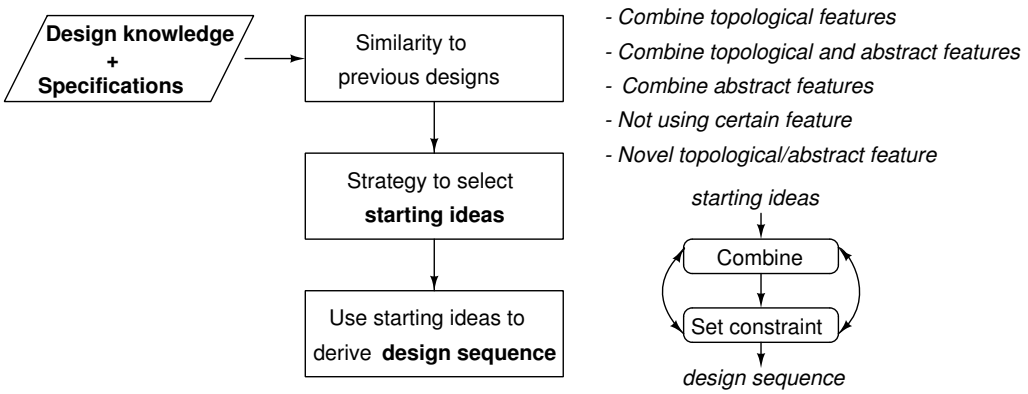


Figure 3.1: Reasoning-based synthesis flow

a solution to the problem specification, e.g., the considered performance requirements. Both steps utilize information produced by the similarity of the current requirements to the requirements of previously solved problems [56].

3.3.1 Selecting starting ideas

Starting ideas are often considered to be the most essential elements in the process of devising an innovative solution [97]. There is active research in cognitive psychology [97] and neuro-science [98] attempting to understand the mechanism through which starting ideas emerge. This process is not always conscious, therefore making it difficult to understand the process. However, once they were selected, starting ideas are characterized as either similar to previous designs (analogies [99]), combinations of existing design features [100], generalizations, e.g., through induction [56], and sudden insight (like through restructuring the knowledge representation). The design flow in Figure 3.1 produces starting ideas using similarity, idea combination, and induction rather than attempting to model the neural, neuro-cognitive mechanism of idea emergence. There are five synthesis flows depending on the types how starting ideas are combined. Synthesis flows/starting ideas are classified into the following categories: Type 1 combines topological fea-

tures from different circuits. Type 2 is a mixture of topological and abstract features. Type 3 involves only abstract features. Type 4 represents starting ideas of not using certain feature, and Type 5 is novel topological/abstract feature used for other purposes than in previous circuits.

3.3.2 Generating design sequence

The reasoning-based procedure utilizes the starting ideas to identify design sequence that create a performance satisfying solution. Each step of the sequence is justified by the fact that it either improves performance or relaxes the design constraints, so that the subsequent steps can further improve performance. Hence, design sequence are produced through decision making, in which every step is causally-explained (justified) by its design improvements. The decision making process can be expressed by various, invariant patterns, which can be utilized to tackle starting ideas of a certain kind [101].

The synthesis flow in Figure 3.1 offers a systematic way of implementing divergent-convergent thinking, well known to be the cognitive mechanism that originates innovations [97], [99]. Starting ideas implement divergent thinking and are the main way of introducing novel features that are beyond the abstractions already utilized in existing designs. The reasoning-based procedure offers convergent thinking by working out the implementation details that make the starting ideas operational (functional) in a circuit design.

3.4 Topology synthesis algorithm

3.4.1 Type 1 - Combine topological features

Algorithm 7 illustrates the reasoning flow for topology synthesis. Given specification, the first step selects a circuit from knowledge representation with performance close to the problem requirements. This circuit's topological features, including the corresponding abstraction in the associative

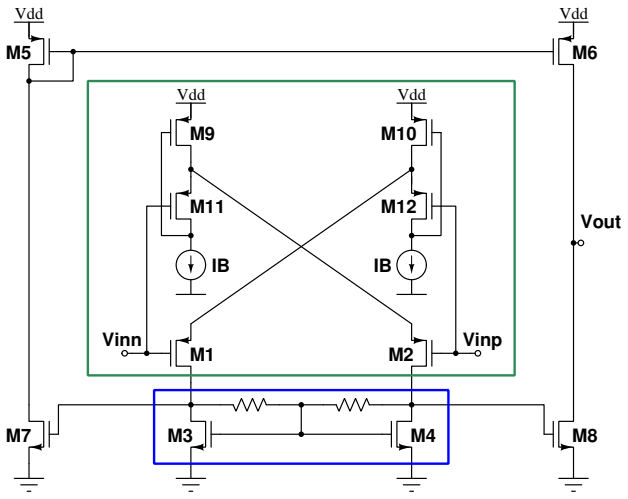


Figure 3.2: Example for starting ideas of Type 1

component, and this circuit's causal reasoning, are used next to identify which features and design step introduce the trade-offs that prevent from satisfying the performance requirements. Then, the methodology selects a second circuit with features that can tackle the unsatisfied trade-offs of the first circuit. The features of the two circuits are utilized to create the starting ideas for the methodology. The starting ideas are followed by adding the topological features required to implement the constraints needed for correct operation of the structures represented by the starting ideas. The final design is produced, if there are no trade-offs preventing the satisfaction of the problem requirements. Otherwise, the reasoning flow iterates by considering new starting ideas to address any unsatisfied requirements. If unsuccessful, other options for the second circuit are analyzed. If still no constraint-satisfying design is created then the features of another circuit are used as starting ideas.

Example. Circuit 13(1) in Figure 3.2 shows a low-voltage power-efficient class AB OTA. The design flow is based on the combination of class AB differential input stage and local common-mode feedback [7], labeled in Figure 3.2. The design is a refinement/advancement of traditional class AB,

using local common mode feedback to tackle {slew rate, small-signal performance} trade-off introduced by class AB stage. These two topological features compose the starting ideas. The design sequence adds a common source gain stage.

Result: Create topology synthesis flow for Type 1
Input : Design knowledge representation;
Output: Topology synthesis flow for Type 1;
begin;
Select C_1 with performance capabilities close to requirements;
Identify the unsatisfied trade-offs & bottlenecks;
Select C_2 with features that could address unsatisfied trade-offs & bottlenecks;
Produce starting ideas by C_1;
Produce starting ideas by C_2;
Combine the selected starting ideas;
Implement the constraints required by the starting ideas;
if *There is no unsatisfied trade-offs & bottlenecks* **then**
    Output final design;
else
    Add new starting ideas,otherwise select another C_2, or select another C_1;
end
end procedure;

Algorithm 7: Algorithm for starting ideas of Type 1

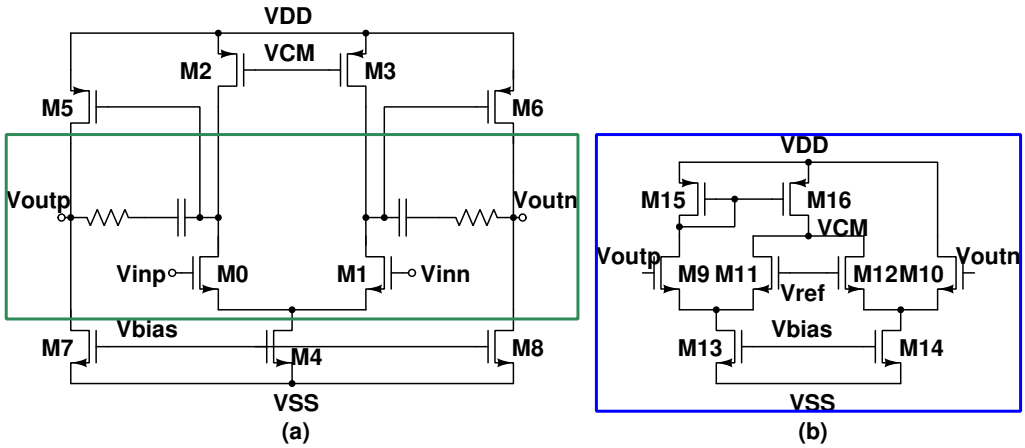


Figure 3.3: Example for starting ideas of Type 2

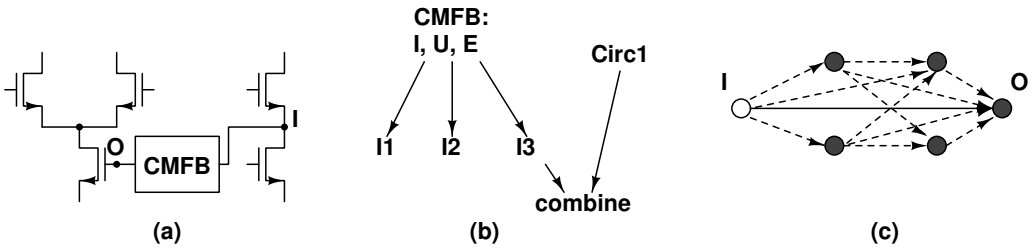


Figure 3.4: Reasoning for starting ideas of Type 2

3.4.2 Type 2 - Combine topological and abstract features

Algorithm 8 illustrates the flow. The first two steps of the reasoning-based synthesis flow are the same as the flow in algorithm 7. However, the third step selects an abstraction, e.g., CMFB, instead of topological features. An abstraction in the associative component corresponds to different implementations (topological features). They are considered next as candidates of starting ideas to be combined with starting ideas of C_1. The pursued flow is similar to steps in algorithm 7. If none of the available feature alternatives can tackle the unsatisfied requirements then the reasoning flow uses a bottom-up induction step to create more topological feature alternatives

that correspond to the CMFB abstraction. Each of the generated features is added to the associative component and then analyzed to decide if it is component of the solution by being combined with the features of the first circuit. This process is similar to steps in algorithm 7. The bottom-up induction step attempts to create new alternatives for a abstraction by using the following mechanism. The input and output nodes of the alternative are similar to all instances of the abstractions, e.g., nodes I and O for the abstraction in Figure 3.4(a). In addition, the alternative will include all nodes and node couplings that are common to all instances of the abstraction, hence are component of the set I $\cup$ O describing the abstraction. We indicated such couplings with a continuous line in Figure 3.4(c). The bottom-up process creates new structures by gradually adding new nodes to the structure until a feasible alternative is identified and the structure is not too complex, e.g., there are too many new nodes added to the network. These nodes are darkened in the figure. These arcs represent the total solution space available for a given set of nodes to create a new alternative for the abstraction. Note that induction creates new features (like mutation in genetic algorithms).

Example. Circuit 18 is a fully differential op-amp with CMFB in Figure 3.3. According to [57], the topological features of fully differential, miller compensation capacitor are combined with an abstraction representing common-mode feedback (CMFB). Figure 3.4 shows an illustrating example for CMFB abstraction, in which three specific instances I_i are topological features of the abstraction. In this example, feature I_3 (Figure 3.3(b)) is selected and combined with the topological features of $Circ_1$ to create a new circuit topology.

Result: Create topology synthesis flow for Type 2
input : Design knowledge representation;
output: Topology synthesis flow for Type 2;
begin;
Select C_1 with performance capabilities close to requirements;
Identify the unsatisfied trade-offs & bottlenecks;
Select abstraction that could address unsatisfied trade-offs & bottlenecks;
Produce starting ideas by C_1;
Produce starting ideas by candidate of abstraction;
Combine the selected starting ideas;
Implement the constraints required by the starting ideas;
if *There is no unsatisfied trade-offs & bottlenecks* **then**
    Output final design;
else
    Add new candidate, otherwise select another abstraction, or select another C_1;
end
end procedure;

Algorithm 8: Algorithm for starting ideas of Type 2

3.4.3 Type 3 - Combine abstract features

Different from combining topological features in Algorithm 7, combining abstract feature and topological feature in Algorithm 8, type 3 combines both abstract features. Candidates of the abstract features are combined. The reasoning process of the synthesis method instantiates the features following the same sequence of steps as the instancing that is performed for starting ideas of Type 2.

Example. Circuit 21 in Figure 3.5 is a multi-path OTA in high gain, high bandwidth applications. Based on [51], the starting ideas describe combining

multi-path with feed-forward compensation. Both features are abstract and can be implemented by different physical structures. Next, the related design sequence adds implementations, i.e., three paths to implement the multi-path, common source feed-forward path to implement the abstract feature of feed-forward compensation, and current source biasing.

Result: Create topology synthesis flow for Type 3
input : Design knowledge representation;
output: Topology synthesis flow for Type 3;
begin;
Select abstraction with performance capabilities close to requirements;
Identify the unsatisfied trade-offs & bottlenecks;
Select abstraction that could address unsatisfied trade-offs & bottlenecks;
Produce starting ideas by candidate of abstraction 1;
Produce starting ideas by candidate of abstraction 2;
Combine the selected starting ideas;
Implement the constraints required by the starting ideas;
if *There is no unsatisfied trade-offs & bottlenecks* **then**
    Output final design;
else
    Add new candidate, otherwise select another abstraction 1, or another abstraction 2;
end
end procedure;

Algorithm 9: Algorithm for starting ideas of Type 3

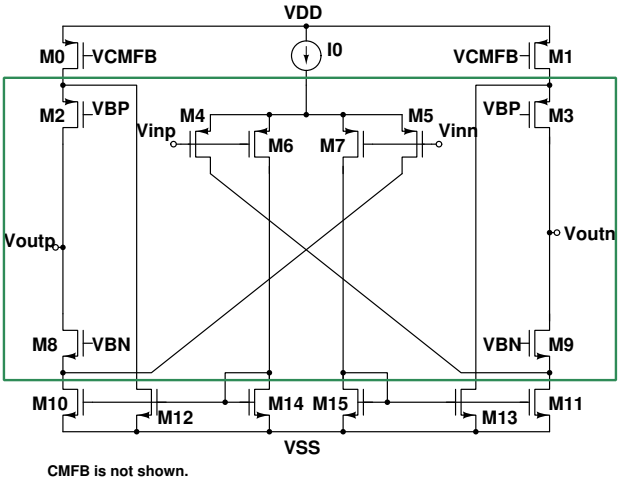


Figure 3.5: Example for starting ideas of Type 3

3.4.4 Type 4 - Not using certain feature

Algorithm 10 presents the proposed reasoning methodology for starting ideas of Type 4. The first step of the methodology identifies the circuit instances or abstractions that include an unwanted feature x. Then, the causal component of the identified circuits and abstractions are utilized to find the justifications for using features x, e.g., the specific performance improvement due to x. In addition, trade-off tables expressing performance capabilities indicate the variables due to feature x and the related performance trade-offs and bottlenecks. The justifications relevant to the problem requirements are used next to identify homonym feature y, e.g., feature that can create the same justifications as feature x, but do not introduce trade-offs that negatively impact the performance requirements. The third step of the methodology selects circuit C_1 with performance capabilities closest to the desired problem requirements.

Next, the features of the selected circuit is combined with feature y. If feature y is at the topological level then the flow continues with the reasoning methodology used for starting ideas of Type 1 (algorithm 7) as the process

resembles now that of combining starting ideas that are topological features. If feature y is abstract, then the reasoning methodology continues by using the flow for combining ideas of Type 2 (algorithm 8) as the process is similar to combining topological features (from circuit C_1) with abstract features (e.g., feature y). Finally, if there are no valid feature y in the associative component then the reasoning flow proceeds by creating an abstraction of feature y. New topological features are produced for this abstraction by using the same bottom-up induction step also used for combining ideas of Type 2. The reasoning process proceeds iteratively to analyze the set of homonym feature y as well as other circuits C_1, if the currently considered circuits do not produce a performance-satisfying solution.

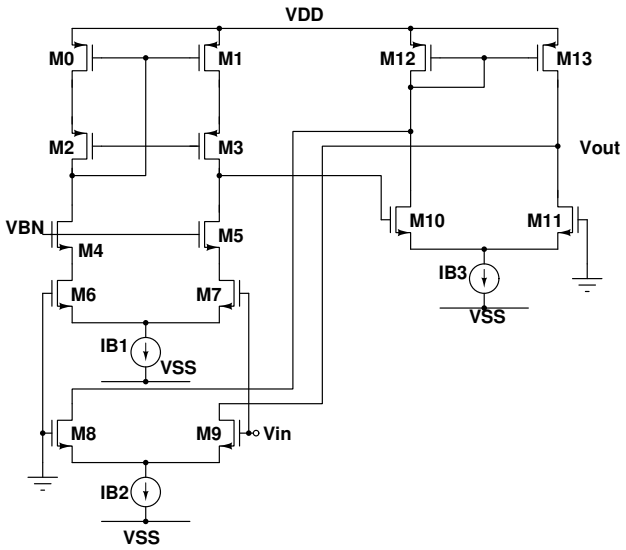


Figure 3.6: Example for starting ideas of Type 4

Example. Circuit 7 in Figure 3.6 is a multistage op-amp with a feed-forward compensation without miller capacitor. In [62], high bandwidth op-amp design might not use miller compensation since the capacitor pushes the dominant pole to lower frequency in spite of good phase margin. Then, the reasoning strategy attempts to identify alternative features that offer the

same justification (i.e., phase margin and high bandwidth). For example, a feed-forward path extends bandwidth through the introduced zeros to cancel second dominant poles for better phase margin. The related design steps add common source path to a regular three-stage op-amp.

Result: Create topology synthesis flow for Type 4
input : Design knowledge representation;
output: Topology synthesis flow for Type 4;
begin;
Identify in the associative component in circuits and abstraction with feature x;
Identify justifications, trade-offs & bottlenecks for x;
Select C_1 with performance capabilities close to the requirements;
if *Exist feature y with the same justification as x* **then**
    Combine y with C_1;
    if *y is topological* **then**
        Reasoning for starting ideas of Type 1;
    else
        Reasoning for starting ideas of Type 2;
    end
else
    Generate abstraction for x;
    Reasoning for starting ideas of Type 2 or 3;
end
end procedure;

Algorithm 10: Algorithm for starting ideas of Type 4

3.4.5 Type 5 - Novel feature for other purposes

The algorithm 11 starts by identifying the topological features f_1, f_2,..., f_n that are candidates to produce a new abstraction. Next, the justifications of each feature are found by examining the causality of the feature (e.g., the performance improvement created by feature f_1 in circuit I_1). The attributes and the controlling variables of each of the selected features are also identified using the performance trade-off tables of the knowledge representation. Feature abstractions are also added to the set of candidate features at this step. Next, the justifications of the candidate features are changed by selecting problem-related performance attributes (that are component of the requirements), which are improved by the feature. The resulting set is called set S. The following steps identify features in other abstractions of the associative component, such that a new abstraction results by combining the features with those in set S. If the new abstraction addresses the problem requirements, then the abstraction is added to the knowledge structure. Otherwise, the reasoning flow iterates by considering other sets of features.

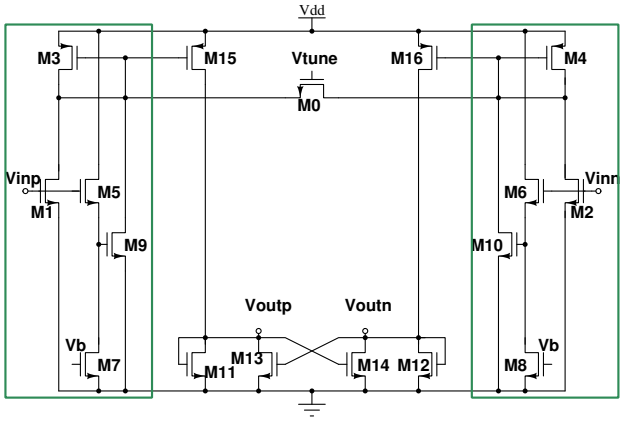


Figure 3.7: Example for starting ideas of Type 5

Example. Circuit 16 in Figure 3.7 is a linearized cross-coupled OTA. In [70], to design a highly linear, high bandwidth circuit, the starting ideas include a pseudo-differential OTA circuit and a feed-forward path crossing

input stage, which is novel feature justified for non-linearity. However, the feed-forward path is extended from input stage to output stage to provide also frequency compensation (the new justification) besides non-linearity cancellation (the traditional justification). The related design step is the implementation of the feed-forward path, adding common mode feedback circuit.

Result: Create topology synthesis flow for Type 5
input : Design knowledge representation;
output: Topology synthesis flow for Type 5;
begin;
Identify features f_1, f_2,..., f_n;
Identify the justification for features f_1, f_2,..., f_n;
Identify the involved performance attributes and controlling variables;
Modify the justification of features in (f_1, f_2,..., f_n and abstractions);
Identify features (set S) that support the modified justifications of the features in (f_1, f_2,..., f_n and abstractions);
Identify features (set T) in other abstractions;
Generate new abstraction by combing features in set S & T;
if *New abstraction address requirements* **then**
    Add new abstraction to the knowledge representation;
else
    Generate new abstraction, otherwise generate modified justification, or identify new features;
end
end procedure;

Algorithm 11: Algorithm for starting ideas of Type 5

3.5 Experiments

This section discusses four new circuits created using different types of reasoning-based topology synthesis algorithms, which are concluded from human design process. We utilized the topological features from the mined design knowledge, which result circuit structures at the same complexity of current manual design solutions.

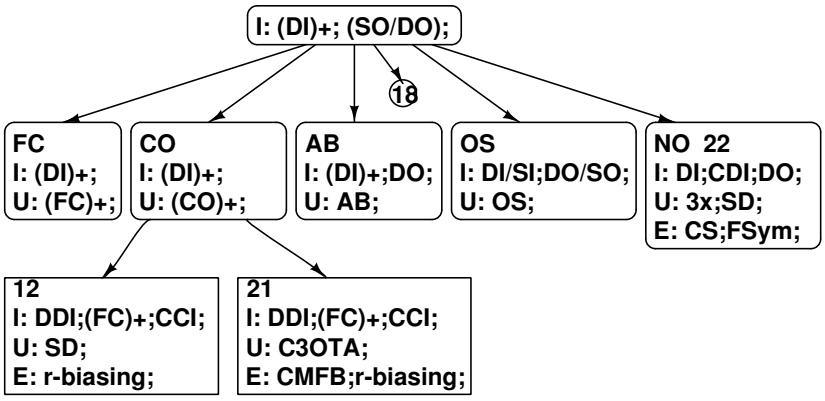


Figure 3.8: Partial associative component for case study 1

3.5.1 Case study 1 - High linearity, slew rate OTA

The specification is to create a circuit that maximizes linearity and slew rate (SR) while keeping gain and bandwidth reasonably high. The used knowledge representation was the one shown in Figure 2.22 and corresponds to the circuits in Figure 2.19, 2.20, 2.21. No circuit in Figure 2.19, 2.20, 2.21 is optimized for both high linearity and high SR. Figure 3.8 is a detailed fragment of associative component in Figure 2.22.

Figure 3.9 illustrates the starting ideas and the design sequence that produced the new circuit. The starting ideas were of Type 2 as they combined cross-coupled input stage (topological feature) and multi-path structure (abstract feature). The starting ideas were identified as follows. The knowledge

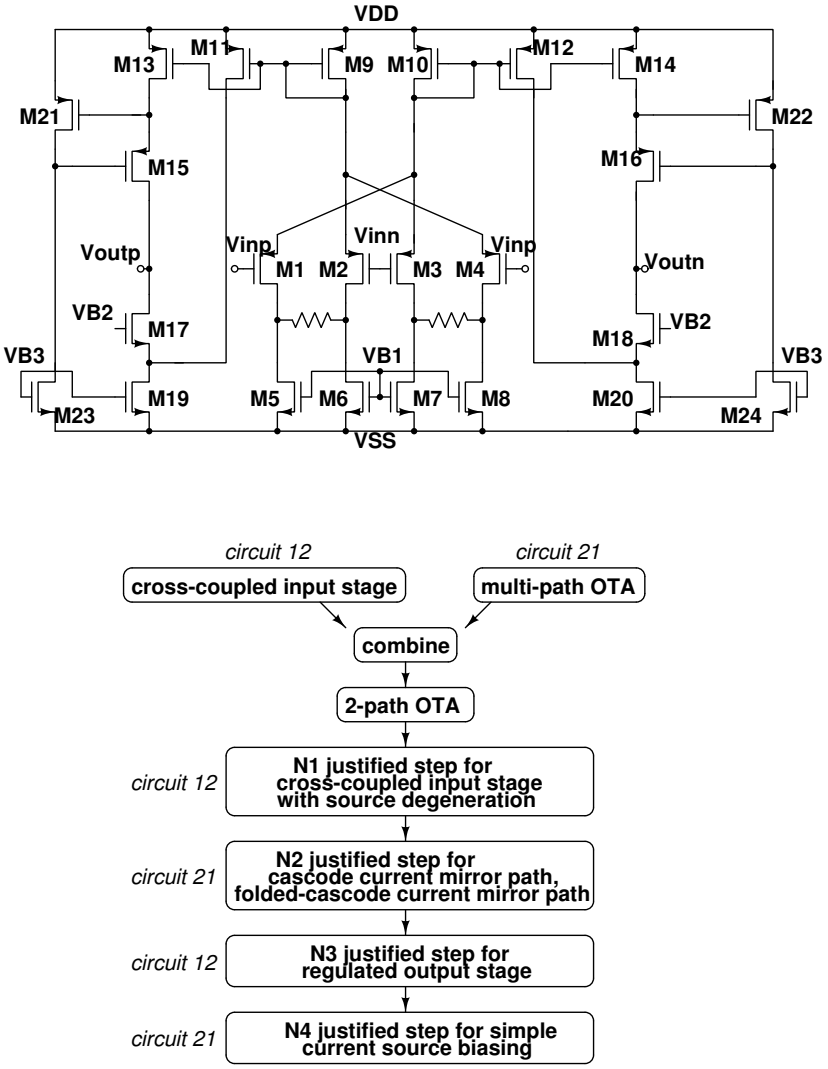


Figure 3.9: Case study 1

representation contains the following highly linear circuits: 5(1), 5(2), 12, 16, 20, 22, 25, and 26. The corresponding abstraction includes cross-coupled input stage as a feature, except circuit 25, which uses a mobility compensation structure. Circuit 21 is the only high SR circuit in the set. Its distinguishing feature is three-path architecture. Cross-coupled input stage feature was selected as it appears in more circuits, hence it has a higher flexibility for being

used in various designs. Combining mobility compensation and multi-path structures could have been considered as starting ideas too.

The design sequence includes the following steps. The starting idea of using an abstract multi-path architecture had to be implemented as a two-path structure. This is justified by relaxing the topological constraints of the circuits, e.g., it is simpler to combine a two-path structure and cross-coupled input stage. Step N_1 introduced source degeneration feature at the input stage (from circuit 12), which is justified by improving circuit linearity. Step N_2 introduced folded cascode and cascode current mirrors to implement the two-path structure (feature from circuit 21). The step is justified by high SR, gain, and output current. Step N_3 adds the regulated output stage (from circuit 12) justified by the need to increase gain. Step N_4 introduces simple current source biasing required for correct operation of the circuit (feature from circuit 21).

Table 3.1: performance comparison of case study 1

Performance	**12**	**21**	**Case study 1**
Technology [μm]	0.6	0.6	0.6
Supply voltage [V]	$\pm$1.65	$\pm$1.65	$\pm$1.65
Static power [mw]	2.34	2.35	3.18
Gain [dB]	51	47	45
Bandwidth [MHz]	0.32	1.34	0.36
unityGainFreq [MHz]	109	285	120
Noise [V^2/Hz]@20MHz	1.45e-14	8.47e-14	1.81e-14
settling time [ns]	45.94	63.63	70.41
Slew Rate [V/μs]	60.5	122	107.9
THD	0.268	1.093	0.403

The resulting circuit topology is shown in Figure 3.9. Table 3.1 summarizes the performance of the new circuit as compared to circuits 12 and 21. SR of the new circuit is by 78% higher than that of circuit 12 and only by

12% lower than that of circuit 21. The linearity (THD) of the new circuit is 50% worse than that of circuit 12 and by 63% better than that of circuit 21. While none of the circuits' Pareto dominates the other two, the new circuit offers a better compromise with respect to achieving simultaneously high linearity and SR.

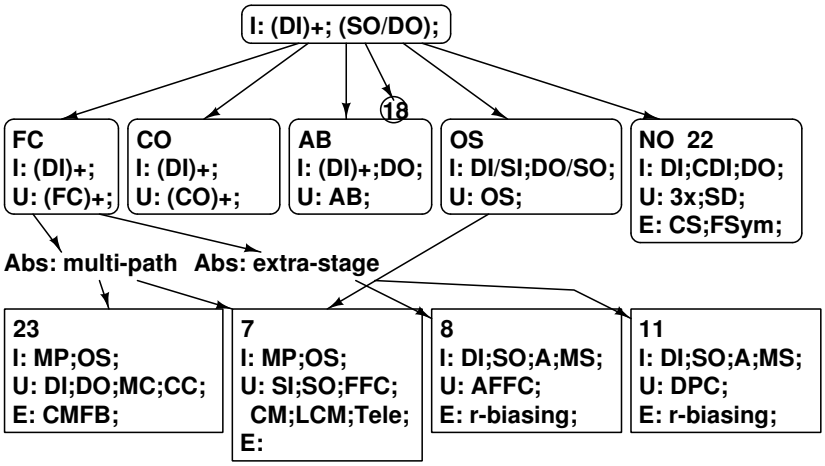


Figure 3.10: Partial associative component for case study 2

3.5.2 Case study 2 - Low power op-amp

The goal of the 2nd case study was to create a low-power amplifier that optimizes the gain-bandwidth product. Figure 3.10 is a detailed fragment of associative component in Figure 2.22, which specifies high gain-bandwidth product. Table 2.17 enumerates the features of the op-amps stored in the database and their abbreviations.

Figure 3.11 illustrates the starting ideas and the design sequence that produced the new circuit. The starting ideas were of Type 4 as they added the constraint of not using compensation capacitors. The starting ideas combined three stage, feed-forward compensation path, and single-ended output features. These ideas were selected from circuits 7, 8, and 11, which are the

high gain, high frequency circuits in the knowledge representation. Feed-forward compensation path was selected as a starting idea as it is in the knowledge set the only way to compensate frequency without using Miller capacitor.

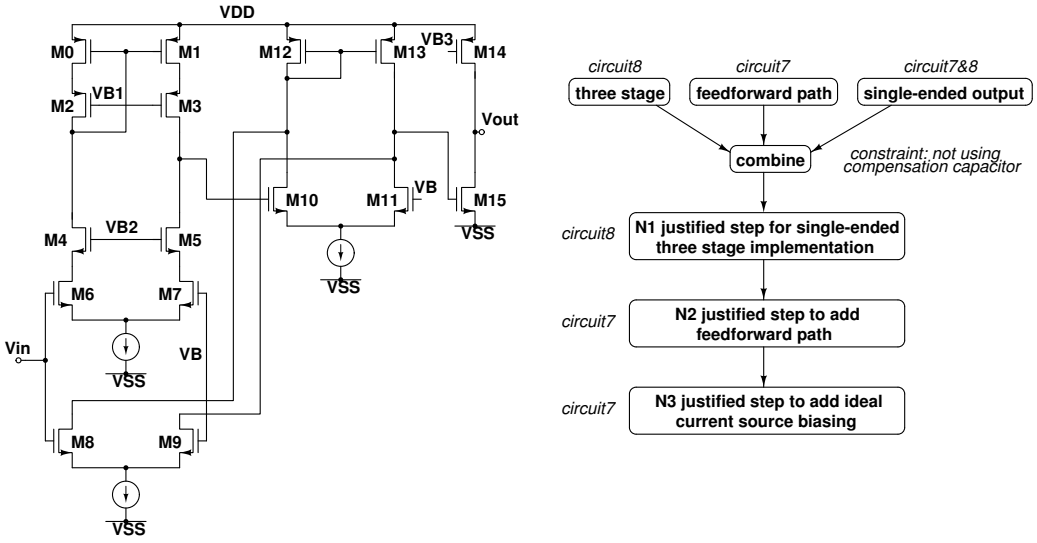


Figure 3.11: Case study 2

The design sequence includes the following steps. Step N_1 implemented three gain stages by cascode, current mirror and common source stages, and the circuit is single-input, single-ended output (features from circuit 8). The step is justified by multistage boosted gain and higher power efficiency. Step N_2 introduced common source to implement the feed-forward path (feature from circuit 7). The step is justified by high gain-bandwidth product and good phase response. Step N_3 added ideal current source biasing required for correct operation of the circuit (feature from circuit 7). The resulting circuit topology is shown in Figure 3.11.

Table 3.2 summarizes the performance of the new circuit as compared to circuits 7 and 8. The gain-bandwidth product is by 13% higher than that of circuit 7 and by 80% higher than that of circuit 8. As a result of

Table 3.2: performance comparison of case study 2

Performance	**7**	**8**	**Case study 2**
Technology[μm]	0.6	0.6	0.6
Supply voltage[V]	$\pm$1.25	2	$\pm$1.25
Static power[mw]	0.63	0.42	0.65
Gain[dB]	71	80	73
Bandwidth[MHz]	0.15	0.012	0.15
GainBwProd[MHz]	539	123	620
Noise[V^2/Hz]@20MHz	1.8e-12	2.7e-14	2.3e-12
settling time[ns]	33.16	63.22	27.24
Slew Rate [V/μs]	40.55	56.12	7.02e3

the maximized gain-bandwidth product, the new circuit has the fastest step response with bet settling time and slew rate accordingly.

3.5.3 Case study 3 - Low power, high gain op-amp

The goal was to create a low power, high gain op-amp that uses a low supply voltage. Figure 3.12 is a detailed fragment of associative component in Figure 2.22, which specifies high gain. Figure 3.13(a) illustrates the starting ideas and the design sequence that produced the circuit in Figure 3.13(b).

The starting ideas were of Type 1 as they combined two topological features: class AB input stage and feedback amplifier. The starting ideas were identified as follows. The knowledge representation contains the following low power circuits: 3, 13(1), 13(2), 13(3), 17, 19, and 30. The corresponding abstraction includes the features: class AB input, local common-mode feedback, and class AB output. Class AB input stage and local common-mode feedback (from circuit 13) were selected as this combination achieves near optimal current efficiency. In contrast, the feature of adaptive biasing class AB input stage (used in circuit 3) was applied only in switched capacitor

(SC) circuits. Also, the feedback of circuit 15 was selected as it is used in a two-stage amplifier.

The design sequence includes the following steps. Step N_1 introduced a simple gain-boosting stage justified by improving the gain for low supply voltage. Step N_2 added a low voltage current mirror (used in circuit 15) justified by increased output resistance. The resulting circuit topology is shown in Figure 3.13(b).

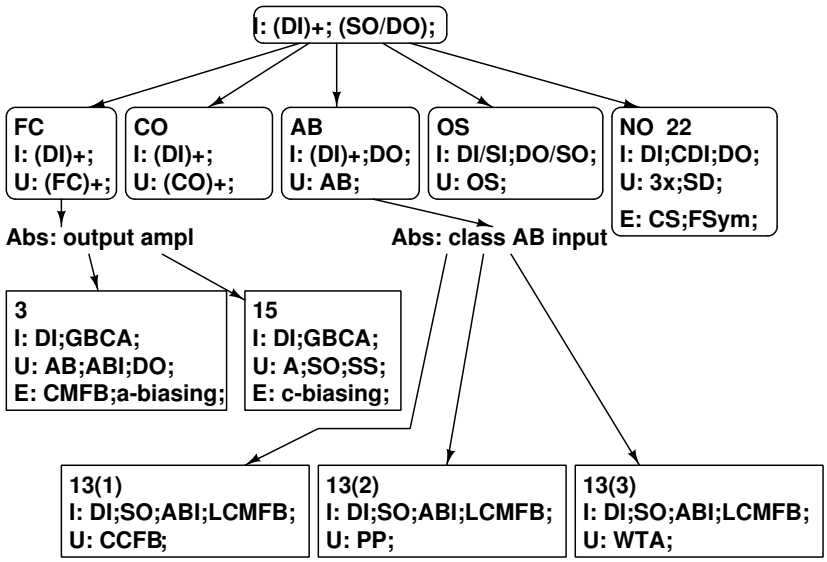


Figure 3.12: Partial associative component for case study 3

Table 3.3 summarizes the performance of the new circuit as compared to circuits 13(1) and 15. Gain of the new circuit is by 38% higher than that of circuit 13(1) and by 24% higher than that of circuit 15. The power consumption is close to that of circuit 13(1) and much less than circuit 15.

3.5.4 Case study 4 - Low voltage, low power op-amp

The goal is to create a low voltage, low power op-amp. A detailed fragment of associative component in Figure 2.22 is shown in Figure 3.14 , which

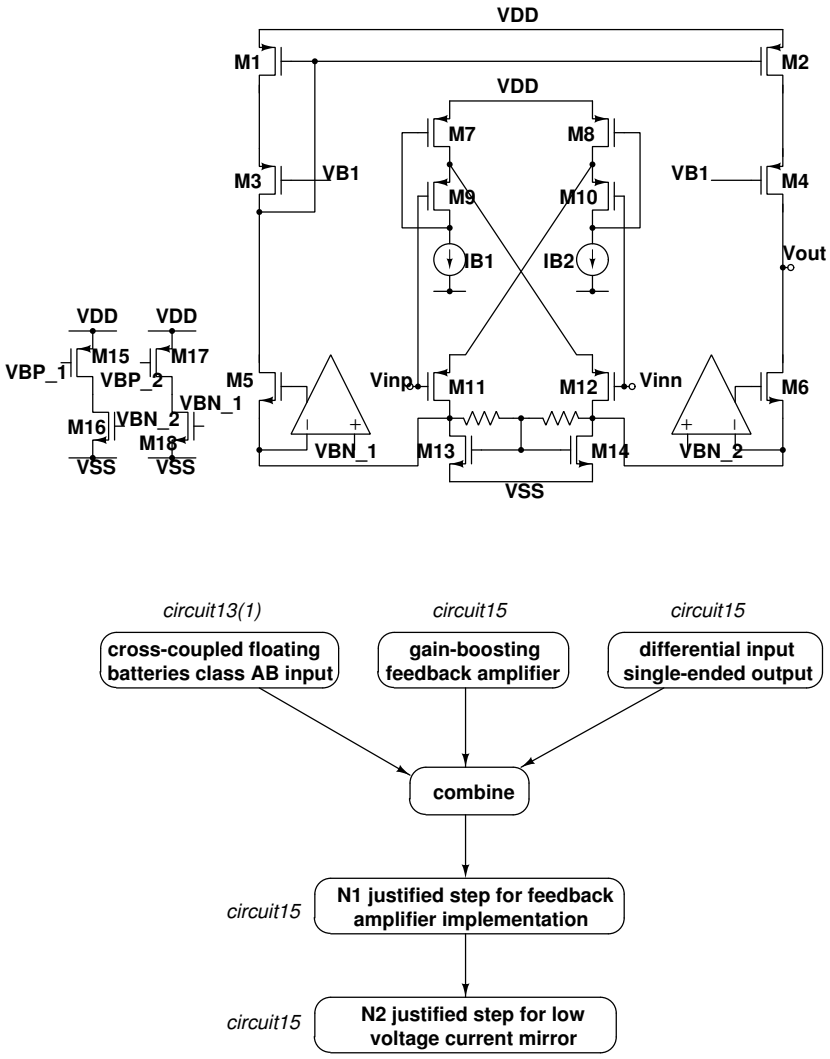


Figure 3.13: Case study 3

specifies low voltage, low power performance. Figure 3.15 illustrates the starting ideas and the causal design sequence.

The starting ideas are identified as follows. They belong to Type 2. Topological features, like adaptive biasing class AB input and local common-mode feedback from circuit 13(1) (Figure 3.12), were selected as they improve slew rate and achieve near-optimal current efficiency. The input stage is

Table 3.3: performance comparison of case study 3

Performance	**13(1)**	**15**	**Case study 3**
Technology[μm]	0.6	0.6	0.6
Supply voltage[V]	$\pm$1	5	$\pm$1
Static power[mw]	0.097	23.8	0.1
Gain[dB]	45	50	62
Bandwidth[MHz]	0.18	0.32	0.02
GainBwProd[MHz]	32	104.8	27.4
PhaseMargin	61°	78.5°	54°
Noise[V^2/Hz]@20MHz	5.3e-16	5.1e-15	1.2e-14

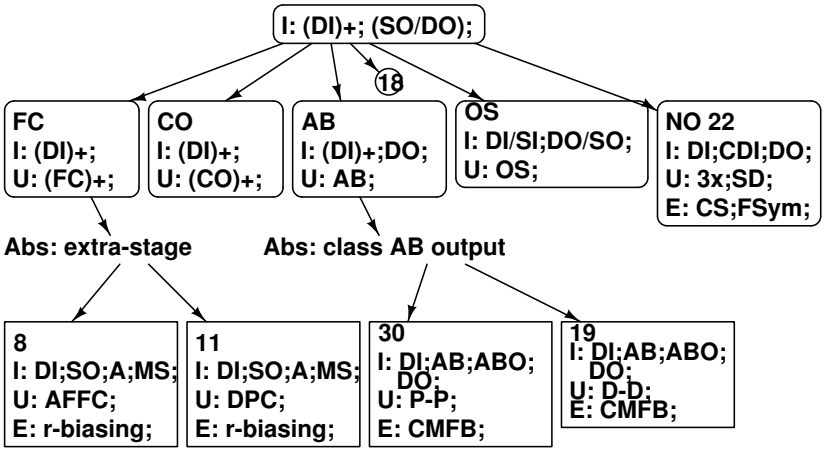


Figure 3.14: Partial associative component for case study 4

general and can be extended to virtually any class AB input stage. Regarding adaptive biasing in circuit 3 for decreasing current during sampling phase, this scheme is only suitable for switched capacitor circuit. Class AB output stages of circuits 19, 30 also realize large output current boosting, but they are at the cost of low gain, which is not desired in high GBP design. To achieve high gain, cascode op-amps, circuits 3, 15 are limited to a very low output voltage swing, under 1-V supply. Stability issues of the gain-boosted

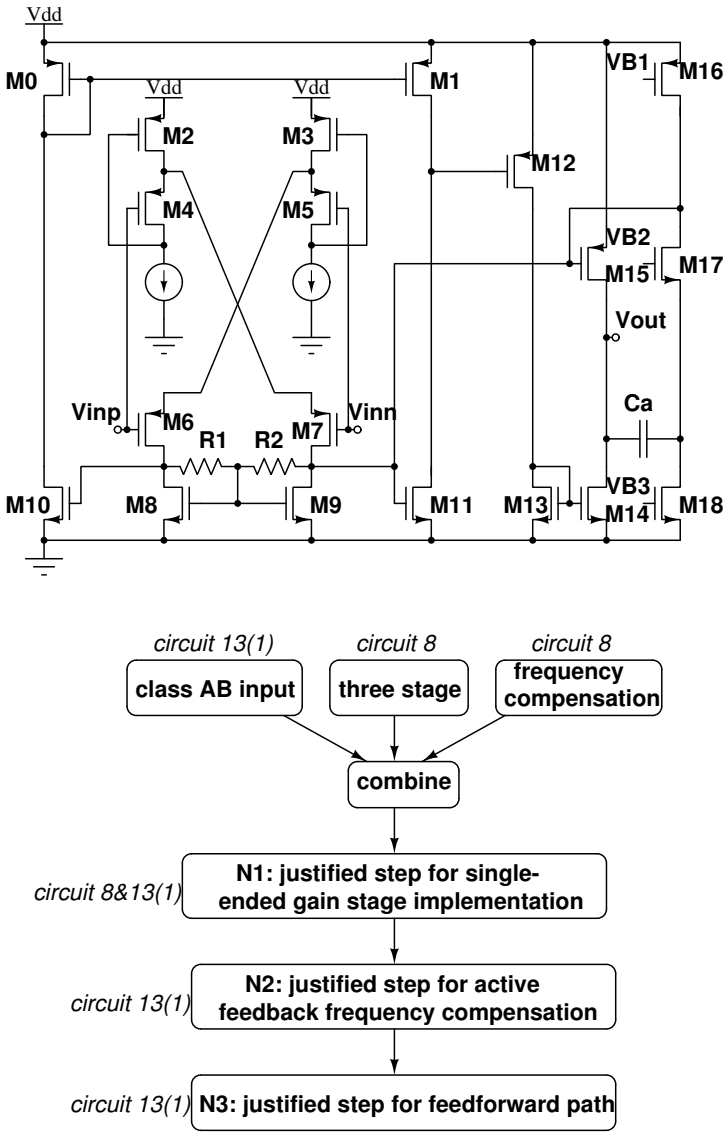


Figure 3.15: Case study 4

cascode structures also need to be carefully addressed. Thus, the starting ideas selected the abstract feature of three stage amplifier. Active feedback frequency compensation technique from circuit 8 was also identified to solve stability issues as it is easier to design and does not consume additional

power.

The causal design sequence includes the following steps. Step N_1 builds a three stage amplifier including class AB input stage and cascaded current mirror gain stages. Class AB input stage is composed of two level shifters and local common mode feedback. Resistors R_1 and R_2 are matched and the common mode of the drain voltages of devices M_6 and M_7 is thus fed back to their common gate, proving additional dynamic current boosting. Using a cascaded current mirror was justified by achieving high gain and high GBP. The output stage is single-ended as it consumes less power compared to fully differential structure. Step N_2 introduced active feedback frequency compensation (AFFC). The step is justified by adding LHP zero to increase phase margin and ensure stability. As an active common gate stage is added in series with the compensation capacitor C_a, the required compensation capacitor is reduced and lowers the physical dimension of the amplifier. Step N_3 added feed-forward path and eliminated Miller capacitor C_m at the output. AFFC and feed-forward path compose a high speed block and control the high frequency operation of the amplifier, i.e.,, non-dominant poles, bandwidth, and phase. The causal design sequence stops whenever a complete design is achieved and bottlenecks are addressed, instead of resulting complicated topologies by adding up features. 3.15 shows the schematic of the final circuit topology.

A comparison with the performance of circuits 13(1), 8, 28 (two sizing solutions) is offered in Table 3.4. All circuits have stable open loop frequency response (sufficient phase margin) and large output swing (at least 0.6V). Two figures of merit FOM_S and FOM_L are used to characterize small-signal (GBW) and large-signal (slew rate) performances. A larger FOM_S, FOM_L indicate better small-signal, large-signal characteristics. Therefore, the proposed circuit is a high performance amplifier under low voltage, low power condition.

$$FOM_S = \frac{GBW * C_L}{power} \tag{3.1}$$

$$FOM_L = \frac{SR * C_L}{power} \tag{3.2}$$

Table 3.4: performance comparison of case study 4

Performance	**13(1)**	**8**	**28(1)**	**28(2)**	**Case study 4**
Technology[μm]	0.2	0.2	0.18	0.18	0.2
Static power [μw@Vdd]	120@±1	400@2	1440@1.8	720@1.8	68@1
DC Gain [dB]	37.5	100	53.6	54.9	58.2
Phase Margin [°]	90	65	70.6	79.8	84.8
GBP [MHz]	0.41	4.5	134.2	70.4	24
Average Slew Rate [V/μs]	84	1.49	94.1	48.1	5.1
Average settling(1%) [ns]	64	–	11.2	20.8	396.4
Capacitive load [pF]	80	120	5.6	5.6	5
FOM$_S$ [$\frac{MHz*pF}{\mu W}$]	56000	1350	522	274	1760
FOM$_L$ [$\frac{V/\mu*pF}{\mu W}$]	273	447	366	187	380

3.6 Conclusions

Analog circuit topology synthesis is important in devising novel design solutions for emerging applications and new design specifications. Existing library, or optimization-based methods rely on predefined, well-known circuit structures, which is hard to involve current design innovations. Genetic algorithm by selection, mutation, and combination operators derives complex circuit with is hard to interpret and compete with manual solution.

As existing techniques differ a lot from manual design process, this chapter develops reason-based topology synthesis utilizing and extending design knowledge mining in Chapter 2. It implements a design knowledge-intensive,

reasoning-based process to create novel circuit structures with all their features justified by the problem requirements.

Knowledge mining summarizes in causal reasoning component five types of reasoning flow in manual design process. Five methods are characterized based on how starting ideas are combined, (*i*) Combine topological features from different circuits. (*ii*) Combine topological and abstract topological features. (*iii*) Combine only abstract features. (*iv*) Not using certain feature, and (*v*) Novel topological/abstract feature used for other purposes. We propose to automate op-amp/OTA design by following the reasoning flow. Topological or abstract features are selected from mined knowledge base of 34 published op-amps/OTAs, to keep updated with current novel design solutions. Four synthesis experiments demonstrate the capability of the method by creating circuits beyond the capabilities of existing topology synthesis algorithms. The synthesized topologies are able to meet specifications and similar to designer-created circuits.

The proposed reasoning-based topology synthesis automates op-amp/OTA design. Future work can explore the design of higher level analog blocks, e.g, filter, oscillator, etc. Even though the methodology is suitable for analog circuits synthesis in general, it needs the mined design knowledge (Chapter 2) as prerequisite. The initial effort to mine design knowledge is one limitation of the reasoning-based synthesis method. Besides, given specification, how to determine the reasoning flow type and select features from knowledge base may need repeated, various attempts. Future work can develop more scientific and efficient way to select reasoning flow strategy and topological feature for particular specification.

Chapter 4

Circuit Causal Information Modeling[1]

4.1 Introduction

Analog circuit sizing in design automation has been described and solved as a constrained optimization problem [3], [102], [103], [104]. These constraints (sizing rules) can include "design-space constraints", "dimension constraints", "manufacturability and operationality constraints", "component constraints", or "soft constraints" [4]. The constrained optimization approaches are significantly different from the designers' sizing strategies, i.e., designers size circuit based on parameters' trade-offs. Designers' thinking and reasoning apply to both topology generation and circuit sizing tasks.

In Chapter 2, we define the mined designers' reasoning on topology generation as causal reasoning. Causal reasoning includes starting ideas, i.e., main topological features being part of a design solution and then utilizes a sequence of topological features to complete the solution. Causal reasoning is guided by explicit knowledge on the causality for using a certain

[1]This work was done jointly with Hao Li. Theory (section 4.3) belongs to teamwork. Experiment (section 4.4) belongs to independent work.

topological/abstract feature in a solution, e.g., to help achieving a specific performance requirement. The starting ideas can include more abstract design concepts and/or detailed topological sub-structures as well as the main way of combining them to process the signals of the design solution. The reasoning process utilizes (a) a set of primitives that are applied to the circuit features, (b) the sequence in which the primitives are analyzed and linked together, (c) the preference in selecting the initial ideas, (d) the priority of selecting topological sub-circuits with similar semantics. In term of circuit sizing, designers' sizing strategy involves a sequence of steps in which each step utilizes the cause-effect connections (causal information) between parameters and performance to traverse the solution space. The priority (importance) associated to circuit parameters and their influence on performance versus the influence of other, alternative parameters on performance are not explored in design automation field.

Chapter 2 utilizes symbolic model to express circuit's performance capabilities including circuit performance trade-offs and parameter sensitivity. Trade-off tables are computed by symbolic modeling, which limits output performances to ac/noise domain as symbolic model approximates circuit as linear system. With respect to finding important parameters to various performance output at a global scale, this chapter extends the work in Chapter 2 by modeling the relations of design parameters deciding the values of the functional outputs, performance, and other parameters (defined as causal information). It models causal information of tool generated design points. Causal information is defined by the following elements: (*i*) Causal relation characterizes the connection between a design parameter, performance attributes, and the other circuit parameters. It expresses the parameter's influence in controlling performance values and trade-offs. (*ii*) Causal trace presents the comprehensive causal influence of a parameter for different importance (weights) of the performance attributes. (*iii*) A causal graph includes causal traces for all circuit parameters, all performance at-

tribute sets. (iv) Sequenced causal relations, (v) graph of sequenced causal relations present the order in which the causal relations of the parameters are used in tackling a design problem. Ordered causal relations reveal parameters' importance on output performance and are considered as circuit sizing strategy. Finally, (vi) causal Pareto front is formed based on original optimal design points and different causal traces.

Characterizing a circuit's causal information is critical. The quality of circuit sizing can be improved by using the parameters causality to decide the parameters that are most effective in addressing the performance tradeoffs of a circuit for a specification. Parameter sequence corresponding to the ordered causal relations can be used as a sizing strategy.

The chapter has the following structure. Section 4.2 discusses the related work. Section 4.3 presents the six causal information elements in circuit design. Section 4.4 discusses experimental results. Conclusion in Section 4.5 ends the chapter.

4.2 Related work

Existing automated circuit sizing methods mainly focus on simulation-based optimization and analytical equations-based optimization.

Simulation-based optimization involves circuit specifications and sizing rules. Circuit specifications refer to performance requirement. Sizing rules [4] focus on transistor sizing constraints (saturation and matching conditions), which guarantee circuit basic function. Simulation-based optimization method solves transistor sizing by structure recognition of decomposed building blocks. One concern is that it's difficult to build a sufficient and canonical building block library. Driven by the fact that symmetric structures commonly exist in analog circuits for reduced offset error and CMRR, [105] extracts matching-driven sizing constraint. Sizing constraint papers differentiate from other optimization papers because of its focus on device itself other than the al-

gorithm. The purpose is to provide better starting point, and establish optimization based on better starting point. Another aspect is getting inner device operating performance by observing transistor parameters, i.e., drain current I_D, inversion level IC, channel length L [106].

Geometric programming is an equation-based optimization method. Geometric programming [107], [108], [109] for automated synthesis has been widely applied in CAD tools because of its advantages on time and performance efficiency, while the drawback lies in the fact that analytical equations need to be derived by expert designer. An alternative bias-driven optimization approach is proposed in [110]. Using the g_m/I_d design concept instead of g_m in equation-based optimization, unknown variables in the performance equations are transformed to the bias voltage regardless of device dimensions. Geometric programming equations are built on lookup-table based models to avoid regression errors. These constrained optimization, i.e., simulation-based and equation-based methods, can be understood as implicit problem solving guided by goal-related gradients and trial-and-error for diversification, but nevertheless with limited or no learning about designer's strategies that involve parameter trade-offs to solve the problem. Example in Chapter 1 shows synthesis tool follows $\{M_3 \rightarrow M_{13} \rightarrow M_{10} \rightarrow M_0 \rightarrow M_4 \rightarrow M_7 \rightarrow M_1\}$, or $\{M_3 \rightarrow M_{13} \rightarrow M_7 \rightarrow M_4 \rightarrow M_0 \rightarrow M_{10} \rightarrow M_1\}$ to size the OTA in Figure 1.3. The sizing sequence starts from the output stage and only considers signal path with one input transistor M_7, which is not able to match the complete design plan proposed by a designer.

Modeling causality has been an important topic in artificial intelligence [111]. Traditional topics include representing, inferring, and using causality in decision making. In analog circuit synthesis (topology generation and circuit sizing), designer's reasoning employs explicit solving strategies that are grounded and guided by causal information. Causal reasoning mining has been discussed in Chapter 2 and applied for topology synthesis in Chapter 3. To the best of our knowledge, causality has been not researched in context

of analog circuit design automation. This problem introduces new issues, like the existence of multiple causal sequences (orderings) corresponding to different problem requirements and alternative ways of achieving the desired requirements. Also, the connections between parameters and outcomes are more diverse and complex than traditional problems, which tend to discuss problems with less coupled parameters. For example, decomposition of the statistical models is an important requirement for method in [111]. Also, causality is mainly approached as an intrinsic part of a system rather than a variable of a broader design process, including systematic diversification through the solution space.

4.3 Causal information modeling

The thesis proposes six elements to model the causality aspects of analog circuit - causal relation, causal trace, causal graph, sequence of causal relations, graph of sequenced causal relations, and causal Pareto front. They cover parameter *vs* performance effect, parameter priority for certain specification and Pareto front exploration. Figure 4.1 shows the causal information modeling flow and a summary of modeling results.

Parameters in Figure 4.1 refer to device parameters that have a main influence on setting the values of performance or of other parameters in a circuit. We define the influence as parameter's causality effect on performance.

Definition - parameter causality: We consider parameter p_i as causal parameter if ($\exists$) performance attribute A_j and ($\exists$) variation $\Delta\ p_i$, so that (i) for $\Delta\ p_i$ the corresponding $\Delta\ A_j > Thresh_j$, a specified threshold value, and (ii) there are feasible values for the other parameters p_k so that $\Delta\ A_j$ can be achieved. p_i is a causal parameter also if ($\exists$) parameter p_k strongly correlated to p_i and the values of p_i must be decided before those of p_k. Experiments considered small signal parameters, $\{g_m, g_{ds}\}$, of all transistors.

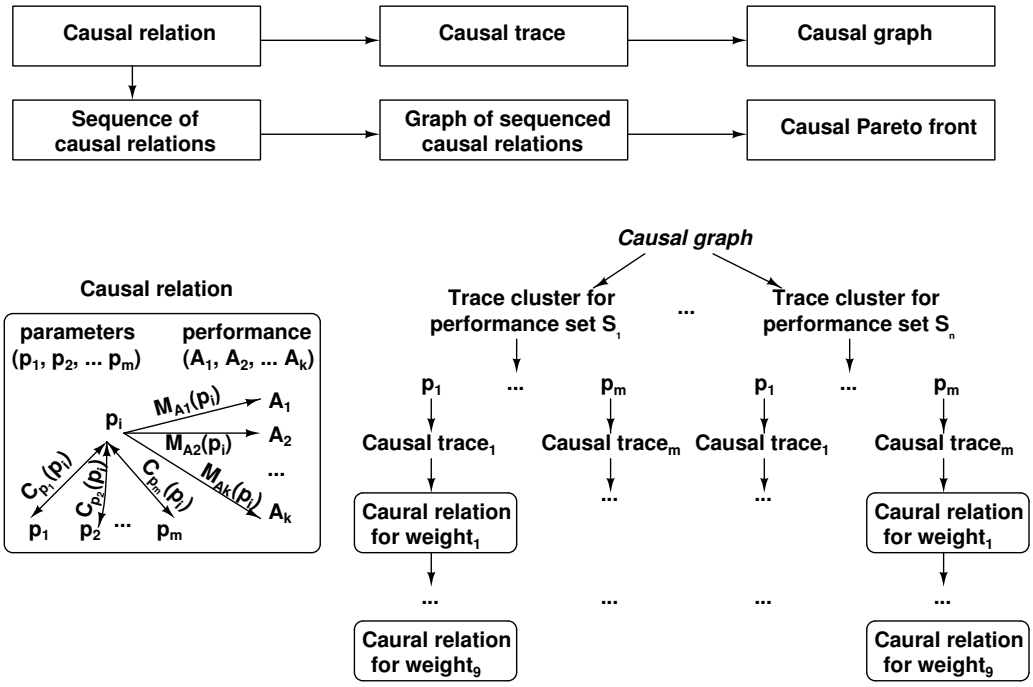


Figure 4.1: Causal information modeling flow

For example, among the entire parameter set, $\{g_{ds3}, g_{m6}$, and $g_{ds6}\}$ are causal parameters with respect to gain as their values are strongly correlated to gain, in Figure 1.3. Performance set in Figure 4.1 refers to the specifications, i.e., {Gain, Bandwidth, THD} in the experiment.

Circuit causal information depends on performance specifications and the constraints on other related parameters. This is due to the performance trade-offs inherent to a circuit. The problem specification might reveal some trade-offs while eliminating others, which ultimately decides how parameters participate in forming causal relations (causal effect on different performance attributes). Similarly, the causal relation between parameters might exist only if related parameters meet certain constraints, e.g., specific values or ranges. In the experiment, we make sure the parameters are all in their feasible ranges by using feasible design points obtained from sizing tool.

4.3.1 Causal relation, causal trace, causal graph

Causal relation: Circuit causal relation refers to a design parameter p_i's effect on performance attributes (defined as model M), and also p_i's correlation matrix with different parameters (defined as model C) in Figure 4.1. It is represented as $\langle$ $M_{A_1}(p_i)$, $M_{A_2}(p_i)$, ..., $M_{A_k}(p_i)$, $C_{p_1}(p_i)$, $C_{p_2}(p_i)$, ..., $C_{p_m}(p_i)$ $\rangle$, where component $M_{A_k}(p_i)$ describes the variation of performance attribute A_k as a function of p_i, while the other parameter p_m relate to parameter p_i according to correlation coefficient $C_{p_m}(p_i)$. This work describes the variation function of $M_{A_k}(p_i)$ as ellipse model. $C_{p_m}(p_i)$ includes the correlation coefficient matrix and also constraints between their value ranges, i.e., parameter's range within the available design space, similar to the approach in [104]. Hence, parameters can have stronger, weaker, or no causal relation with respect to a performance attribute or other parameters.

We implement $M_{A_k}(p_i)$ by matlab ellipse fitting to express the stochastic dependency. Important ellipse parameters include the slope, two axes, vertical and horizontal ranges. The shape of ellipse describes the degree of causal control a parameter has over a performance attribute: (*i*) very thin shapes describe close-to linear causal control, (*ii*) parameters are independent, if the shapes are circular or parallel with the x-axis, and (*iii*) parameters that are enabling (parameter values must meet a constraint for the entire range of the performance values), if the ellipse model has a thin, vertical shape.

The individual causal relation (model M, C) is based on optimized transistor sizing solution for a particular performance attribute set (weighted specifications). The sizing solutions are obtained using automated analog circuit sizing tool in Cadence Virtuoso [5]. To generate a comprehensive causal description, thus to avoid the tool being repeatedly trapped in a local region, the optimization step performs a uniform sampling of the feasible value range of each circuit parameter. Each transistor sizing solution is obtained from optimization on each sampling set. The enforced parameter sampling is an approximation of the actual sizing space, especially by not sampling combi-

nations (clusters) of constrained parameters as in [104]. Under the condition of large execution time running the automated sizing tool, the uniform sampling scheme is chosen in the experiment. An alternative way of avoiding the synthesis tool being trapped in a local region is by repeatedly sizing a circuit.

Causal trace: Causal relation models the relation between individual parameter and individual performance/other parameters. While for a performance attribute set which contains multiple performance attributes, we propose causal trace is a trace of causal relations for performance attributes with different weights (importance), in Figure 4.1. For example, let's consider two performance attributes, {Gain, Bandwidth}. The causal trace includes the causal relations for a relevant uniformly sampled performance weights, like {weight$_{Gain}$ = 1.0, weight$_{Bandwidth}$ = 0.0}, {weight$_{Gain}$ = 0.875, weight$_{Bandwidth}$ = 0.125}, ..., {weight$_{Gain}$ = 0.0, weight$_{Bandwidth}$ = 1.0}, in Figure 4.2. The theorem states that if the same causal relations describe the end points of a range for the weights, then the same causal relations hold also for the weight values inside the range. Hence, no sampling is needed inside the range.

Causal graph: The trace cluster for a set of performance attributes includes causal traces for all circuit parameters. A causal graph includes the trace clusters of a circuit for all possible performance attribute sets relevant to the problem description. Figure 4.2 illustrates a causal graph for performance attribute sets {Gain, Bandwidth}, {Gain, THD}, each trace cluster includes causal traces corresponding to circuit parameters {g$_{m0}$, g$_{ds0}$, ..., g$_{ds10}$}.

After computing the causal traces for each possible pair of performance attributes (A$_i$ and A$_j$), the obtained causal traces are merged to create the causal traces for triplets of performance attributes, traces for triplets are merged to produce quadruples of attributes until the entire specification set is covered. The optimum solution would be to compute the causal relations separately for each performance attribute set by repeatedly sizing the circuit

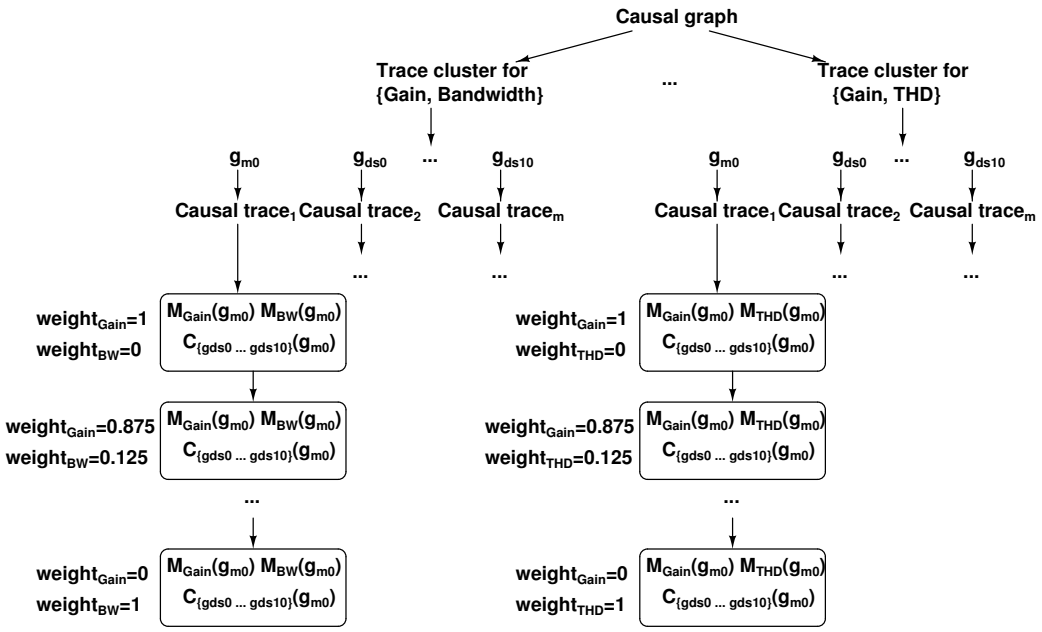


Figure 4.2: Causal trace example

for different weight samples of the performance attributes. However, the execution time would be unfeasibly high due to the need to synthesize inside the modeling loop. Instead, it re-uses the relations computed for k-attribute sets to find the causal relations for k+1 attribute sets. Trace merging is inspired by bottom-up data cube construction and frequent item set mining in data mining [12].

The merging step first considers all common causal relations (CR) in trace cluster 1 (TC1) and trace cluster 2 (TC2), e.g., traces that present the causal relation of the same parameter p on the same attribute of TC1 and TC2. The next loop identifies model M overlap of $CR_1 \in$ TC1, $CR_2 \in$ TC2 (their intersection is not void). A new causal relation (new_CR) is created next to express the model overlapping, which then allows to identify the corresponding domain of the controlling parameter p. The two distinct performance attributes are updated by finding the closest design point based on p in the existing design points.

Example: Computing the causal relations for performance attributes {Gain, Bandwidth, THD} produces the same results whether using traces {Gain, Bandwidth}, {Gain, THD} or traces {Gain, THD}, {Bandwidth, THD}.

The weights associated to a merged causal trace is computed as follows. Initially, let's assume that the weights of the two traces are sets $\{\alpha_1, \alpha_2\}$ ($\alpha_1 + \alpha_2$=1) and $\{\beta_1, \beta_2\}$ ($\beta_1 + \beta_2$=1). α_1, β_1 correspond to the common performance attribute. The merged trace is characterized by set $\{(\alpha_1 + \beta_1)/2, \alpha_2, \beta_2\}$. Experiments only merge causal traces up to three dimensions. This procedure for associating weights to the merged trace is justified by the causal continuity theorem for which the end points are the weights of each individual trace used in merging. Qualitatively, the resulting set of weights is described by the set $\{\mu_i, \sigma_i\}$, where μ_i and σ_i are the mean and standard deviations of the corresponding set of weights.

Analog circuit behavior can be typically expressed as a set of coupled differential equations and explained by solving equations within reasonable accuracy. There are no intrinsic orderings among circuit parameters, whereas a specific design methodology (sizing strategy) of assigning design parameters different importance is involved in human design. The strategy utilizes preferences among parameters based on their capability, revealing a designer's perspective on how parameters can be used in order to meet the problem requirements. The thesis proposes that ordered causal relations of circuit parameters can be mined and modeled that extend the existing metrics to explore a solution space. Traditionally, metrics like descending gradients or frequency based diversification have been used in continuous-valued optimization problems [104]. However, these metrics are less efficient in design reuse or diversification guided by the pursued problem goals.

4.3.2 Sequence of causal relations, graph of sequenced causal relations

Sequence of causal relations: A sequence of causal relations represents an ordering of causal relations, i.e., as introduced by different ways to order model M. Each node corresponds to the causal relations for a device parameter and the directed arcs describes the parameter sequencing of the ordering. It is an ordered causal trace.

Mining different sequenced causal relations reveals the different design steps pursued according to sizing strategies. Related work in engineering design science suggests that solutions should attempt (i) to maximize the linearity through which design parameters control functionality and performance [112], [113] and (ii) to minimize the degree of coupling between competing performance attributes by minimizing the number of common controlling parameters for different attributes. Also, a sizing strategy should (iii) minimize the coupling degree (correlation) between the control parameters and the other parameters of a solution [113]. Finally, (iv) the solution strategies must attempt to maximize the sharing of causal relation sub-sequences as they encourage reusing [20]. Different sizing strategies materialize the four requirements in different ways.

Graph of sequenced causal relations: Sequences of causal relations corresponding to different orderings, are captured by the graph of sequenced causal relations.

There are multiple ways to order causal relations for a given performance attribute set. They correspond to specific sizing strategies to optimize performance specification. The thesis proposes different measures to order the causal traces. Firstly, the circuit parameters that occur in the causal graph for all performance attributes and all associated weights are summarized into four sets (Set_i). The four sets are as follows: Set_1 contains all parameters. Set_2 is a set of independent parameters with respect to the performance attribute set of a trace (performance attributes with weights). A parame-

ter might be independent with respect to a given attribute set (e.g., {Gain, Bandwidth}), but not with respect to the other (e.g., {Gain, THD}). Set_3 is the set of parameters that have partially constrained ranges they must meet independently of the performance attributes and weights (ellipse model M tends to have a vertical shape). Set_4 refers to the set of controlling parameters that we interest in ($\text{Set}_1 - \text{Set}_2 \cap \text{Set}_3$).

Next, we propose four different measures to order causal relations. For each performance set, sequences of parameters are stored in variables Table_i. Parameters which are independent to performance or have narrow ranges are pruned (Set_2, Set_3). Only controlling parameters in Set_4 are considered to build Table_i. Table_1 orders the controlling parameters depending on their linearity in controlling the performance attributes and the size of the performance range they can offer. Parameters that offer higher linear control over a larger range are ordered first. Table_1 measure uses the following weighted sum:

$$Lin_i = \sum_j \sum_k \beta_{lin} linearity_{jk} + \beta_{ran} range_{jk}; \tag{4.1}$$

$$linearity_{jk} = 1/(2 * ellipse_{jk}.short\_radius); \tag{4.2}$$

$$range_{jk} = (max(P_{jk}) - min(P_{jk})/(max(P) - min(P)); \tag{4.3}$$

where index i refers to parameter index, j refers to performance index in the performance attribute set, and k refers to performance weight index. linearity_{jk} is the inverse of the ellipse short axis diameter. range_{jk} is the ratio of parameter i offered performance attribute range under weight k to the overall performance attribute range. Narrower ellipse model indicates shorter diameter, larger linearity_{jk}, plus larger range of parameter's control on performance attributes.

$$Global\_corr_i = \sum_j \sum_k \sum_{m \neq i} \rho^{jk}_{m \neq i} \tag{4.4}$$

Table_2 stores the controlling parameters in increasing order of their correlations with the other parameters (the parameters with a low correlation are first).

Where index m refers to the other parameters than parameter j. $\rho^{(jk)}_{m \neq i}$ is the correlation coefficient between parameter p_m and p_i for performance j, weight k.

Table_3 orders the controlling parameters based on their similarity with other parameters, i.e., the parameters with a less similar effect as other parameters are shown first. The ordering presents the uniqueness of a parameter in setting performance attribute, i.e., the degree of control including linearity, range, and the correlation with other parameters. The similarity measure of parameter p_i is as follows:

$$Sim_i = \sum_{\forall PAS \in CG} \frac{1}{NumPara} * (\gamma_{Lin} Lin_i + \gamma_{Corr} Global\_corr_i) \tag{4.5}$$

PAS is performance attribute set present in causal graph CG. *NumPara* is the number of controlling parameters in *PAS*. γ_{Lin} and γ_{Corr} are the weights for linearity and correlation coefficients.

Table combines three criteria of Table_1, Table_2, Table_3 depending on their significance assigned by each strategy. It uses a weighted sum of coefficients Lin_i, $Global\_corr_i$, Sim_i:

$$Cumm_i = \Psi_{Lin} Lin_i + \Psi_{Corr} Global\_corr_i + \Psi_{Sim} Sim_i \tag{4.6}$$

Finally, Table_4 further extracts a subset of correlated parameters in Set_4, i.e., with correlation coefficients larger than threshold.

4.3.3 Causal Pareto front

A complete representation of multi-objective performance trade-off is through Pareto front. The thesis proposes causal Pareto front, which is derived from a circuit's Pareto front and the ordered causal traces.

The less important circuit parameters that are not part of the causal relations can still have some influence on the circuit performance. Hence, the concept of Pareto front is relaxed, so that it represents all performance trade-off surfaces that are within θ from the actual Pareto front where parameter p_i is correlated to parameter p_j. Computing causal Pareto front has two main for-loops considering all performance sets and parameters in $Table_i$. Each trace cluster expresses the Pareto front for certain performance attribute set, and causal Pareto front is expressed by updating design points in the trace cluster. It traverses the parameters ordered by $Table_i$, causal relations of the first parameter are copied into current trace (CR_1) while keeping only the Pareto dominant design points of the causal relations. Causal relations of the other parameters (CR_2) are compared with those in CR_1. Each design point y_1 in CR_1 is compared with point y_2 in CR_2 where y_1, y_2 have the same parameter t as the controlling parameter, and values of t are closest in CR_1, CR_2. If y_1 is dominated by y_2, y_1 is removed from CR_1 and the dominant point y_2 is added to the new causal relation (new_CR) that will act as a causal updated CR_1. With updated design points, model M and C are recalculated in new_CR. The new trace is added as a segment of causal Pareto front, if it is not dominated by any of the traces already present.

Figure 4.3 presents an illustrating example of different parameter orders for weighted performance attributes. The top level refers different performance attribute sets (PAS_i) corresponding to different segments on causal Pareto front. The attributes in PAS_i are partitioned into the subsets that are separately tackled by the design parameters, e.g., {Gain, Bandwidth} = {Gain} $\cup$ {Bandwidth}, {Gain} $\cap$ {Bandwidth} = $\emptyset$. For each attribute under weight, the representation indicates the ordered causal relations that

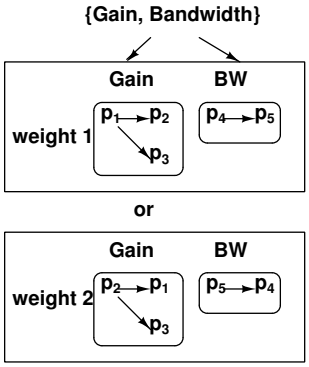


Figure 4.3: Parameter sequence example

generate the Pareto-dominant design points. For example, for set {Gain}, the causal relation based on parameter p_1 and correlated parameters p_2 and p_3 offer the dominant design points for the weighted attribute {Gain} and the causal relation using parameter p_4 and correlated parameter p_5 decide the weighted attribute {Bandwidth}. For the same attribute set PAS with different weight, there is another sequence of causal relations that generates a different region of Pareto points. Taking order $p_1 \rightarrow p_2$, parameters p_1, p_2 correspond to two sets of sampled optimal design points (CR_1, CR_2), which compose the original Pareto front. If design point y_1 in CR_1 is dominated by y_2 in CR_2, y_1 in CR_1 will be updated by y_2. Updated causal relation shows one strategy to achieve better gain following $p_1 \rightarrow p_2$, which composes one segment of causal Pareto front.

There are different ways in which circuit parameters and the corresponding causal relations characterize specific regions of the solution space (different segments of Pareto front). The parameters are ordered into different sequences based on their different effects on performance attributes. We already showed that different circuit parameters can decide gain and bandwidth. Performance trade-offs emerge when the same parameter has opposite influences on two or more performance attributes, like parameter g_{13} must be decreased to improve gain (with correlation coefficient -0.8) but increased

to improve bandwidth (with correlation coefficient 0.9). Causal Pareto front shows circuit trade-offs in multi-dimensional space. Parameter ordering offers sizing strategy to improve different segment on the solution space.

4.4 Experiments

Experimental section shows causal information modeling results on op-amp 4(1) (in Figure 1.3), including six elements - causal relation (CR), causal trace (CT), causal graph (CG), sequence of causal relations (SCR), graph of sequenced causal relations (GSCR), and causal Pareto front.

4.4.1 Experiments on CR, CT, CG

Causal relation. Utilizing Cadence advanced optimization tool, we are able to obtain optimal design points (sizing solutions) under different sets of weighted performance specifications. Experiments considered 2 performance attribute sets (PAS) {Gain, Bandwidth}, {Gain, THD}. Each performance attribute set is assigned 9 different weight sets, [1, 0], [0.875, 0.125], [0.75, 0.25], [0.625, 0.375], [0.5, 0.5], [0.375, 0.625], [0.75, 0.25], [0.125, 0.875], [0, 1], i.e., [1, 0] = {weight$_{Gain}$=1, weight$_{Bandwidth}$=0}. For each weighted PAS, 64 sampled design points are generated by single objective optimization. To simplify the sampling process, only transistors on the signal path are sampled. Each transistor on the signal path, {M_3, M_6, M_7, M_{10}}, is sampled 16 times {15μm, 30μm, ..., 240μm} evenly along channel width due to technology constraint. Thus, each causal relation is constructed based on the 64 optimal design points, including circuit sizing solutions (transistor width), small signal parameters {g_m, g_{ds}}, performance attributes {Gain, Bandwidth, THD}.

A few observations on the sizing results are as follows.

(i) From design points obtained by sizing tool, we observe that in some cases sizing solutions are the same in both dimensions and performance. For example, among design points of performance attribute {Gain, Bandwidth},

Table 4.1: Design points example from sizing tool

M_0 [μm]	M_1 [μm]	M_{10} [μm]	M_{11} [μm]	M_{13} [μm]	M_{14} [μm]	M_3 [μm]	M_4 [μm]	M_6 [μm]	M_7 [μm]
vb1 [V]	vb2 [V]	vb3 [V]	vb4 [V]	Gain [dB]	Bandwidth [Hz]				
250	32.74	10	8.442	3.055	8.442	150	4.287	45.94	4.287
2.7	2.4	1.183	0.7	69.48	10950				
250	32.74	10	8.442	3.055	8.442	165	4.287	45.94	4.287
2.7	2.4	1.183	0.7	69.48	10950				
250	32.74	10	8.442	3.055	8.442	180	4.287	45.94	4.287
2.7	2.4	1.183	0.7	69.48	10950				
250	32.74	10	8.442	3.055	8.442	195	4.287	45.94	4.287
2.7	2.4	1.183	0.7	69.48	10950				

weight [1, 0] by sampling transistor M_3 in Table 4.1, 4 design points result the same dimensions.

(*ii*) Figure 4.7, 4.8, 4.9 indicate that for different weights and different sweeping transistors, the design points obtained do not span the trade-off space evenly. For different weights, it is seen that sweeping transistors M_6, M_7 (input transistors) is more likely to cluster in regions. Sweeping transistors M_3, M_{10} (output transistors) covers more area of the trade-off space. The partial segments provided by M_6, M_7 are more separated, indicating a relatively independent relationship in term of sizing. Whereas M_3, M_{10} are more dependent on each other.

(*iii*) There exists design points corresponding to higher performance weight from sizing tool but with worse performance result. In Table 4.2, 4.3 sweeping M_3 under different weights, the design points with higher weight priority but worse performance are marked with *. Figures 4.10, 4.11 show an overall distribution of design points under different weights.

The Causal relation models the effect of parameter on performance as ellipses, e.g., g_{ds0} *vs* Gain, g_{m1} *vs* Bandwidth, etc. The shape of the ellipse, i.e., how narrow it is and how much it is tilted, represent the correlation and dependence between parameters and performance attributes. Figure 4.4 presents ellipse models for {Gain, Bandwidth} under weight [0.5, 0.5]. As there are 10 independent transistors due to circuit symmetry, in Figure 4.4 shows a subset of complete parameters set, {g_{ds0}, g_{m0}, g_{ds1}, g_{m1}, g_{ds10}, g_{m10},

Table 4.2: Sweeping M_3 [15μm$\sim$120μm] under different weights

M_3	[1, 0]	[0.875, 0.125]	[0.75, 0.25]	[0.625, 0.375]	[0.5, 0.5]
M_3	[0.375, 0.625]	[0.25, 0.75]	[0.125, 0.875]	[0, 1]	
15μ	[70.99, 8242]*	[65.76, 15440]*	[69.76, 8366]*	[72.51, 8185]	[70.41, 7762]
15μ	[70.34, 9370]	[70.23, 11410]	[70.18, 8528]	[43.92, 84310]	
30μ	[71.80, 10910]	[71.42, 10870]	[71.63, 10600]	[69.87, 10200]	[71.91, 10790]*
30μ	[67.74, 10960]	[70.85, 12010]*	[58.44, 8134]	[45.87, 91570]	
45μ	[63.46, 15950]*	[65.94, 11680]	[65.71, 11150]	[66.23, 10420]*	[66.24, 10400]*
45μ	[63.74, 14950]	[54.66, 13620]	[68.31, 10540]*	[41.32, 340800]	
60μ	[65.92, 10890]*	[66.83, 11970]	[65.92, 10890]	[66.80, 12000]*	[66.26, 14410]*
60μ	[66.12, 8226]*	[63.96, 10840]	[65.08, 11600]*	[46.32, 74870]	
75μ	[64.01, 11130]	[65.43, 11290]*	[65.58, 11380]*	[68.93, 9612]*	[65.56, 11390]*
75μ	[63.50, 8634]	[66.55, 11180]	[66.38, 12480]	[40.12, 560200]	
90μ	[63.83, 13860]*	[71.42, 9568]	[63.83, 13860]	[69.56, 12920]*	[50.69, 128000]
90μ	[63.54, 8571]*	[63.54, 8571]	[59.05, 122900]	[39.01, 336600]	
105μ	[60.37, 19870]*	[62.54, 9920]*	[64.34, 12830]*	[70.88, 9265]	[63.10, 14440]
105μ	[63.32, 8808]	[63.32, 8808]	[67.05, 13750]*	[43.75, 82410]	
120μ	[69.22, 8558]	[59.50, 13010]*	[62.88, 13790]*	[68.90, 9578]	[62.23, 11420]
120μ	[54.60, 24700]	[53.79, 21390]	[63.36, 20000]*	[37.76, 389100]	

g_{ds11}, g_{m11}, g_{ds13}, g_{m13}, g_{ds14}, g_{m14}, g_{ds3}, g_{m3}, g_{ds4}, g_{m4}, g_{ds6}, g_{m6}, g_{ds7}, $g_{m7}\}$. Comparing the ellipse models for different parameters, we conclude that g_{ds}, g_m belonging to the same transistor have similar models, i.e., g_{ds10} *vs* g_{m10}, g_{ds6} *vs* g_{m6}, g_{ds13} *vs* g_{m13}. Parameters $\{g_{ds10}, g_{m10}, g_{ds13}, g_{m13}\}$ have negative effect on Gain, whereas $\{g_{ds6}, g_{m6}\}$ have positive effect. $\{g_{ds10}, g_{m10}, g_{ds13}, g_{m13}\}$ are more correlated with Gain, with correlation coefficient 0.7, but $\{g_{ds6}, g_{m6}\}$ are less correlated, with correlation coefficient 0.5. Correlation coefficients between different parameters indicate a trade off relation. Table 4.4 shows the parameter ranges and the correlation coefficients between different parameters. Gain ranges from 50.69dB to 73.4dB. From the table we can also conclude, parameters g_{ds} and g_m are highly correlated, transistors M10 and M13 are correlated in a positive trend.

Causal trace, causal graph. Causal trace composes a list of causal

Table 4.3: Sweeping M_3 [135μm$\sim$240μm] under different weights

135μ	[70.73, 8387]	[65.57, 11020]	[65.72, 11110]	[52.68, 189400]	[65.03, 15380]*
135μ	[50.70 , 108500]	[60.05, 42330]	[64.59, 11110]	[38.11, 375000]	
150μ	[69.48, 10950]	[70.97, 9194]	[59.46, 53920]	[65.48, 57300]	[57.04, 16190]
150μ	[53.37, 105100]	[63.37, 8679]*	[51.05, 99340]	[38.02, 508300]	
165μ	[69.49, 10920]*	[70.97, 9169]	[65.74, 11030]*	[69.90, 9133]	[68.66, 11890]
165μ	[54.21, 109400]	[48.88, 84040]	[65.59, 50320]*	[42.36, 328400]	
180μ	[66.55, 29280]*	[70.97, 9144]	[65.75, 10990]	[63.75, 45670]	[66.06, 12590]*
180μ	[54.39, 106900]	[51.03, 84400]	[65.57, 54110]*	[36.85, 227500]	
195μ	[69.49, 10850]	[70.47, 10820]*	[62.24, 74220]	[58.78, 71710]	[63.77, 45500]*
195μ	[55.28, 80510]	[56.88, 93720]	[65.65, 53470]*	[47.79, 189800]	
210μ	[69.50, 10820]	[69.49, 30640]	[60.17, 50880]*	[62.09, 65280]	[63.80, 45280]
210μ	[54.62, 85450]	[55.91, 84570]	[65.72, 52890]	[35.14, 678100]	
225μ	[61.56, 46320]*	[69.52, 10780]	[60.23, 50430]	[60.01, 63530]	[57.51, 122900]
225μ	[55.35, 79460]	[55.99, 83500]	[65.78, 52370]*	[32.67, 830000]	
240μ	[69.47, 31050]	[59.59, 70880]*	[61.90, 74270]	[60.25, 90180]	[57.61, 121200]
240μ	[55.38, 79030]	[56.08, 82500]	[65.83, 51880]*	[41.69, 379500]	

Table 4.4: Parameter ranges and correlation coefficients

Parameter	Range [s]	g_{ds10}	g_{ds6}	g_{ds13}	g_{m10}	g_{m6}	g_{m13}
g_{ds10}	[1.178e-6, 8.022e-5]	1	-0.1205	0.6716	0.8920	-0.1103	0.7403
g_{ds6}	[5.463e-7, 4.904e-5]	-0.1205	1	-0.4224	-0.3382	0.9991	-0.4335
g_{ds13}	[1.981e-6, 2.119e-4]	0.6716	-0.4224	1	0.8411	-0.4221	0.9328
g_{m10}	[1.412e-4, 9.597e-3]	0.8920	-0.3382	0.8411	1	-0.3356	0.8902
g_{m6}	[5.904e-5, 5.446e-3]	-0.1103	0.9991	-0.4221	-0.3356	1	-0.4331
g_{m13}	[7.569e-5, 1.163e-2]	0.7403	-0.4335	0.9328	0.8902	-0.4331	1

relations for each parameter under 9 different weights. Causal traces belonging to different performance attribute sets are further merged to create triplets of performance attributes. e.g., causal traces of {Gain, Bandwidth}, {Gain, THD} are merged to obtain {Gain, Bandwidth, THD}. Causal graph summarizes all causal traces for 20 parameters and 2 performance attribute sets. Figure 4.5 shows a subset of causal graph for performance attributes {Gain, Bandwidth}, {Gain, THD}. Three parameters {g_{ds6}, g_{ds3}, g_{m10}},

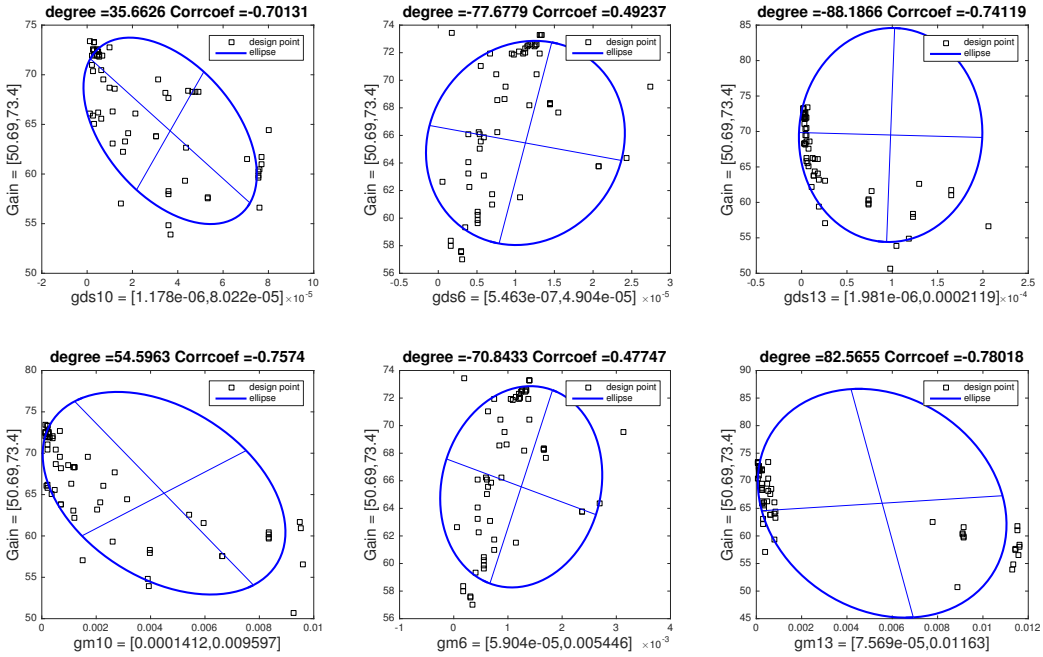


Figure 4.4: Circuit 4(1) causal relation ellipse models

three weight pairs {[0.5, 0.5] [0.5, 0.5], [0.875, 0.125] [0.875, 0.125], [1, 0] [0, 1]} are selected to illustrate different merging cases. Figure 4.6 presents the merged ellipse modeling results. The number of design points falling into the common area of two ellipses varies. 67 points are in the common area of g_{ds6} models, 81 points are in g_{ds3} models, 21 common points in g_{m10} models.

In the first two cases, the weights optimizing {Gain, Bandwidth}, {Gain, THD} are the same, whereas the last case [1, 0], [0, 1] are far apart. Therefore, the solution space on the Pareto front of the first two cases share more area than that of weights [1, 0], [0, 1]. With the common design points, we were able to predict a triplet of performance attributes {Gain, Bandwidth, THD} by projecting common parameters on {Gain, Bandwidth}, {Gain, THD} models. Taking 5 common points in "Gain - Bandwidth - THD - [0.875, 0.125, 0.125] - g_{ds3}" merged model, marked in Figure 4.6, Table 4.5 presents the g_{m10} performance attribute ranges. The 5 points are also syn-

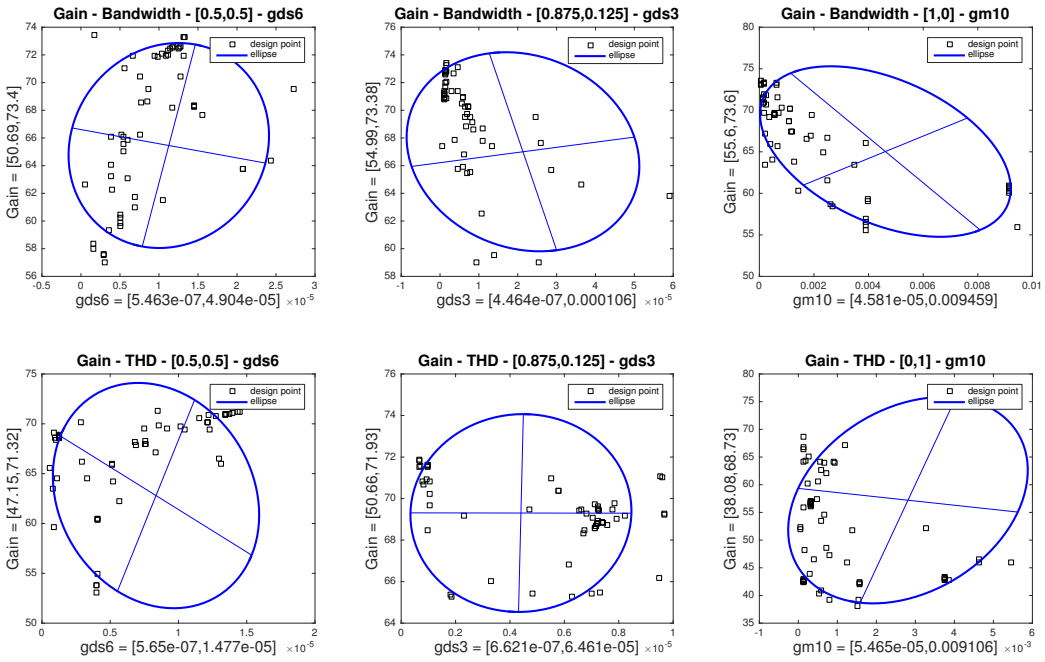


Figure 4.5: Subset of causal graph ellipse models

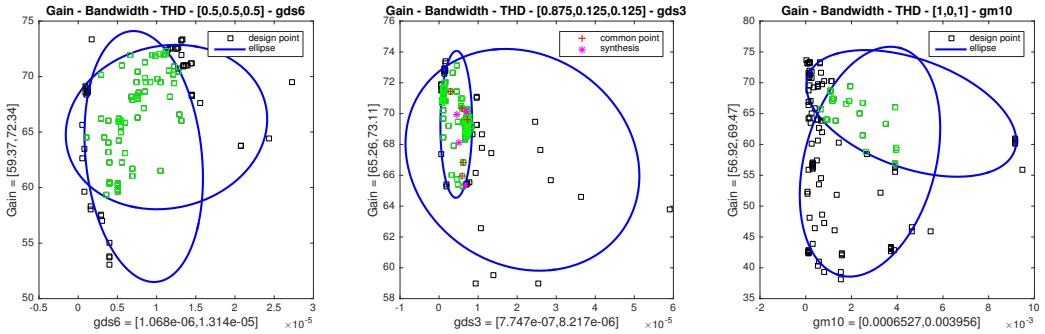


Figure 4.6: Causal relation merging cases

thesized by the tool, and Table 4.5 includes the synthesis results.

4.4.2 Experiments on SCR, GSCR, causal Pareto front

Sequence of causal relation, graph of sequenced causal relations. Sequenced causal graph presents sequence of causal traces for performance

Table 4.5: Performance attributes and synthesis results for g_{m10}

Common	Gain [dB]	Bandwidth [Hz]	THD [%]	Synthesis	Gain [dB]	Bandwidth [Hz]	THD [%]
1	71.42	[6.6898e3, 2.1703e4]	[4.2034, 14.9182]	1	69.96	10930	8.872
2	65.94	[6.6578e3, 2.5957e4]	[5.0289, 16.3628]	2	68.14	12570	5.811
3	66.83	[6.6821e3, 2.6199e4]	[5.0816, 16.4528]	3	65.33	11450	4.349
4	69.62	[6.8654e3, 2.7532e4]	[5.3842, 16.9654]	4	70.15	8680	6.048
5	70.36	[6.635e3, 2.5702e4]	[4.9737,16.2683]	5	70.15	8626	5.748

attribute sets {Gain, Bandwidth}, {Gain, THD}, {Gain, Bandwidth, THD}. For each performance attribute set, different causal trace orders present different sizing strategies that used to size the circuit for certain specification. In order to further analyze the parameter control on performance and parameters coupling, we pruned possible parameters that are less correlated to performance attributes or constrained by certain range (relatively vertical and narrow ellipse models). By doing this, we cluster parameters into different sets. Set 1 includes all parameters. Set 2 includes less correlated parameters, correlation coefficients on performance no more than 0.2. Parameters with ellipse models degree within [80, 90] and the sub short axis of ellipse within [0, 0.1] (relatively vertical and narrow ellipses) are clustered in Set 3. Table 4.6 shows the pruned parameters (Set 2, Set 3) and we only keep the rest controlling parameters for further study (Set 4).

Further analysis of the controlling parameters in Set 4 includes computing their linearity in controlling the performance, the performance range that offered, correlations with other parameters, etc. Since each parameter corresponds to a causal trace representing parameter control for different weights, an order of parameters based on control criteria reflects the sizing strategy used when sizing the circuit. Different sizing strategies exist based on different control criteria. For example, to meet {Gain, Bandwidth} specification, parameter which has the most linear control on Gain/Bandwidth is considered first. The experiment computes the different criteria, linearity, correlation, uniqueness as follows. Linearity reflects on the parameter linear control, i.e., how linear the associated ellipse model is. Ellipse model with

Table 4.6: Causal trace parameter sets and tables

Gain - Bandwidth trace cluster	
Set 1	g_{ds0}, g_{ds1}, g_{ds10}, g_{ds11}, g_{ds13}, g_{ds14}, g_{ds3}, g_{ds4}, g_{ds6}, g_{ds7}, g_{m0}, g_{m1}, g_{m10}, g_{m11}, g_{m13}, g_{m14}, g_{m3}, g_{m4}, g_{m6}, g_{m7}
Set 2	g_{ds11}, g_{ds14}
Set 3	$\emptyset$
Set 4	g_{ds0}, g_{ds1}, g_{ds10}, g_{ds13}, g_{ds3}, g_{ds4}, g_{ds6}, g_{ds7}, g_{m0}, g_{m1}, g_{m10}, g_{m11} , g_{m13}, g_{m14}, g_{m3}, g_{m4}, g_{m6}, g_{m7}
table 1	g_{ds3}, g_{ds13}, g_{m1}, g_{ds0}, g_{ds1}, g_{m6}, g_{ds6}, g_{m11} , g_{m14}, g_{m13}, g_{m3}, g_{ds10}, g_{m0}, g_{m10}, g_{m4}, g_{m7}, g_{ds7}, g_{ds4}
table 2	g_{m11} , g_{m0}, g_{ds10}, g_{m14}, g_{ds6}, g_{m6}, g_{ds1}, g_{ds0}, g_{m10}, g_{m1}, g_{ds13}, g_{m3}, g_{ds4}, g_{ds3}, g_{m4}, g_{m13}, g_{ds7}, g_{m7}
table 3	g_{ds3}, g_{ds13}, g_{m11} , g_{m0}, g_{ds0}, g_{m6}, g_{m1}, g_{ds6}, g_{ds1}, g_{m1}, g_{ds10}, g_{m10}, g_{m3}, g_{m13}, g_{m4}, g_{m7}, g_{ds7}, g_{ds4}
table	g_{ds3}, g_{ds13}, g_{m11} , g_{m0}, g_{ds0}, g_{m6}, g_{m1}, g_{ds6}, g_{ds1}, g_{m14}, g_{ds10}, g_{m10}, g_{m3}, g_{m13}, g_{m4}, g_{m7}, g_{ds7}, g_{ds4}
table 4	g_{ds3}, g_{ds6}, g_{ds7}, g_{ds10}, g_{m3}, g_{m6}, g_{m7}, g_{m10}
Gain - THD trace cluster	
Set 1	g_{ds0}, g_{ds1}, g_{ds10}, g_{ds11}, g_{ds13}, g_{ds14}, g_{ds3}, g_{ds4}, g_{ds6}, g_{ds7}, g_{m0}, g_{m1}, g_{m10}, g_{m11} , g_{m13}, g_{m14}, g_{m3}, g_{m4}, g_{m6}, g_{m7}
Set 2	g_{ds1}, g_{ds11}, g_{ds14}, g_{ds7}, g_{m1}, g_{m4}, g_{m7}
Set 3	$\emptyset$
Set 4	g_{ds0}, g_{ds10}, g_{ds13}, g_{ds3}, g_{ds4}, g_{ds6}, g_{m0}, g_{m10}, g_{m11} , g_{m13}, g_{m14}, g_{m3}, g_{m4}, g_{m6}
table 1	g_{ds10}, g_{ds13}, g_{ds3}, g_{m3}, g_{m10}, g_{m0}, g_{m11} , g_{ds0}, g_{m14}, g_{m13}, g_{ds4}, g_{m6}, g_{ds6}
table 2	g_{ds10}, g_{m0}, g_{m11} , g_{ds0}, g_{m10}, g_{ds3}, g_{m3}, g_{m14}, g_{ds13}, g_{ds4}, g_{m13}, g_{ds6}, g_{m6}
table 3	g_{ds10}, g_{m0}, g_{ds3}, g_{m10}, g_{m11} , g_{m3}, g_{ds13}, g_{ds0}, g_{m14}, g_{m13}, g_{ds4}, g_{ds6}, g_{m6}
table	g_{ds10}, g_{m0}, g_{ds3}, g_{m10}, g_{m11} , g_{m3}, g_{ds13}, g_{ds0}, g_{m14}, g_{m13}, g_{ds4}, g_{ds6}, g_{m6}
table 4	g_{ds3}, g_{ds6}, g_{ds10}, g_{m3}, g_{m6}, g_{m10}
Gain - Bandwidth - THD trace cluster	
Set 1	g_{ds0}, g_{ds1}, g_{ds10}, g_{ds11}, g_{ds13}, g_{ds14}, g_{ds3}, g_{ds4}, g_{ds6}, g_{ds7}, g_{m0}, g_{m1}, g_{m10}, g_{m11} , g_{m13}, g_{m14}, g_{m3}, g_{m4}, g_{m6}, g_{m7}
Set 2	g_{ds11}
Set 3	$\emptyset$
Set 4	g_{ds0}, g_{ds1}, g_{ds10}, g_{ds13}, g_{ds14}, g_{ds3}, g_{ds4}, g_{ds6}, g_{ds7}, g_{m0}, g_{m1}, g_{m10}, g_{m11} , g_{m13}, g_{m14}, g_{m3}, g_{m4}, g_{m6}, g_{m7}
table 1	g_{m10}, g_{ds10}, g_{m0}, g_{m3}, g_{m13}, g_{m6}, g_{ds0}, g_{ds6}, g_{ds13}, g_{ds3}, g_{ds14}, g_{ds1}, g_{ds4}, g_{m7}, g_{ds7}, g_{m1}, g_{m11} , g_{m4}, g_{m14}
table 2	g_{ds14}, g_{ds10}, g_{m11} , g_{m0}, g_{m14}, g_{ds6}, g_{m6}, g_{m3}, g_{m1}, g_{ds1}, g_{m13}, g_{m4}, g_{m10}, g_{ds7}, g_{m7}, g_{ds0}, g_{ds4}, g_{ds13}, g_{ds3}
table 3	g_{ds14}, g_{ds10}, g_{m0}, g_{m10}, g_{m3}, g_{m11} , g_{m13}, g_{m6}, g_{ds6}, g_{m14}, g_{ds0}, g_{ds1}, g_{m1}, g_{ds13}, g_{m7}, g_{ds7}, g_{m4}, g_{ds3}, g_{ds4}
table	g_{ds14}, g_{ds10}, g_{m0}, g_{m10}, g_{m3}, g_{m11} , g_{m13}, g_{m6}, g_{ds6}, g_{m14}, g_{ds0}, g_{ds1}, g_{m1}, g_{ds13}, g_{m7}, g_{ds7}, g_{m4}, g_{ds3}, g_{ds4}
table 4	g_{ds3}, g_{ds6}, g_{ds7}, g_{ds10}, g_{m3}, g_{m6}, g_{m7}, g_{m10}

small short axis but large performance range is ordered first. Correlation presents the coupling between parameters. The parameters less correlated with the others are ordered first. Uniqueness is an overall of linearity and coupling.

The parameters which are more unique are ordered first. Table 4.6 presents different ordering of causal traces in Set 4 computed by different criteria. The orders in table 1, table 2, table 3, table represent different sizing strategies with respect to linearity, global correlation, uniqueness, and combined criteria. To simplify the process, we only consider a subset of pa-

rameters which have more control on performance than the others (table), i.e., table = {g_{ds3}, g_{m3}, g_{ds6}, g_{m6}, g_{ds7}, g_{m7}, g_{ds10}, g_{m10}} for {Gain, Bandwidth}, table = {g_{ds3}, g_{m3}, g_{ds6}, g_{m6}, g_{ds10}, g_{m10}} for {Gain, THD}. Each order represents one exclusive design plan, whereas we may also consider other orders in design process. Based on the linearity in controlling performance, the sizing strategy should be {g_{ds3}, g_{m6}, g_{ds6}, g_{m3}, g_{ds10}, g_{m10}, g_{m7}, g_{ds7}}. We may also consider {g_{ds10}, g_{ds6}, g_{m6}, g_{m10}, g_{m3}, g_{ds3}, g_{ds7}, g_{m7}}. Permuting parameters expands the orders in table 1, table 2, table 3, table, and results 6 orders for {Gain, Bandwidth}, 48 orders for {Gain, THD}, orders for {Gain, Bandwidth, THD}. Table 4.7 lists a subset of orders expanding design possibilities.

Table 4.7: Causal trace parameter orders

Gain - Bandwidth sequenced causal traces	
table 1	$g_{ds3} \rightarrow g_{m6} \rightarrow g_{ds6} \rightarrow g_{m3} \rightarrow g_{ds10} \rightarrow g_{m10} \rightarrow g_{m7} \rightarrow g_{ds7}$
table 2	$g_{ds10} \rightarrow g_{ds6} \rightarrow g_{m6} \rightarrow g_{m10} \rightarrow g_{m3} \rightarrow g_{ds3} \rightarrow g_{ds7} \rightarrow g_{m7}$
table 3/table	$g_{ds3} \rightarrow g_{m6} \rightarrow g_{ds6} \rightarrow g_{ds10} \rightarrow g_{m10} \rightarrow g_{m3} \rightarrow g_{m7} \rightarrow g_{ds7}$
Gain - THD sequenced causal traces	
table 1	$g_{ds10} \rightarrow g_{ds3} \rightarrow g_{m3} \rightarrow g_{m10} \rightarrow g_{m6} \rightarrow g_{ds6}$
table 2	$g_{ds10} \rightarrow g_{m10} \rightarrow g_{ds3} \rightarrow g_{m3} \rightarrow g_{ds6} \rightarrow g_{m6}$
table 3/table	$g_{ds10} \rightarrow g_{ds3} \rightarrow g_{m10} \rightarrow g_{m3} \rightarrow g_{ds6} \rightarrow g_{m6}$
Gain - Bandwidth - THD sequenced causal traces	
table 1	$g_{m10} \rightarrow g_{ds10} \rightarrow g_{m3} \rightarrow g_{m6} \rightarrow g_{ds6} \rightarrow g_{ds3} \rightarrow g_{m7} \rightarrow g_{ds7}$
table 2	$g_{ds10} \rightarrow g_{ds6} \rightarrow g_{m6} \rightarrow g_{m3} \rightarrow g_{m10} \rightarrow g_{ds7} \rightarrow g_{m7} \rightarrow g_{ds3}$
table 3/table	$g_{ds10} \rightarrow g_{m10} \rightarrow g_{m3} \rightarrow g_{m6} \rightarrow g_{ds6} \rightarrow g_{m7} \rightarrow g_{ds7} \rightarrow g_{ds3}$

Causal Pareto front. For causal traces corresponding to {p_1, p_2} in order Table 4.7, we mark causal trace of p_1 as current trace (CR_1), causal trace of p_2 as CR_2. The design points in CR_1 are compared with design points in CR_2 where CR_1, CR_2 have the same controlling parameter values. Dominated points are removed from CR_1 and the dominant points from

CR_2 are added. Thus, CR_1 correlates with partial Pareto front which is controlled by parameter g_{ds3}. The causal graph with causal traces for all performance attribute sets, all weights compose the entire Pareto front. They are high priority orders with respect to parameter linearity, correlation, and uniqueness. Other permutations derived from orders in table 1, table 2, table 3 and table are also added for later analysis.

For example, depending on parameter linearity in controlling performance attribute {Gain, Bandwidth}, the design sequence based on the order should be $\{g_{ds3} \rightarrow g_{m6} \rightarrow g_{ds6} \rightarrow g_{m3} \rightarrow g_{ds10} \rightarrow g_{m10} \rightarrow g_{m7} \rightarrow g_{ds7}\}$, as g_{ds3} has the most linear control over Gain, Bandwidth and g_{ds7} has the least linear control. Figure 4.12 shows the performance improvement following sequence $\{p_1 \rightarrow p_2\}$ (p_1, p_2 are parameters on the sequence). Original design points are marked as stars, whereas the new design points obtained from $\{p_1 \rightarrow p_2\}$ are marked as boxes. Stars and boxes overlap if there is no improvement by moving to p_2. From Figure 4.12, we see design points are improved in $\{g_{ds3} \rightarrow g_{ds10}\}$, $\{g_{ds3} \rightarrow g_{m10}\}$. There are other sizing strategies, i.e., parameter correlation with others, uniqueness. Design points are improved differently in these cases, as shown in Figure 4.13. Figure 4.12 show the improvement sequence for partial solution space corresponding to M_3. Figure 4.13 shows the improvement sequence for partial solution space corresponding to M_{10}. Figure 4.16 shows an example for performance attribute {Gain, THD} based on sequence $\{g_{ds10} \rightarrow g_{ds3} \rightarrow g_{m3} \rightarrow g_{m10} \rightarrow g_{m6} \rightarrow g_{ds6}\}$. Figures 4.18, 4.19, 4.20 are the results for merged performance attribute {Gain, Bandwidth, THD}.

For each partial solution space obtained by sweeping transistors M_3, M_6, M_7, M_{10} (transistors on the signal path), there are a few points get improved, but not a lot. The improvement depends on the design sequence. For example, for order $\{g_{ds3} \rightarrow g_{m6}\}$, design point controlled by g_{ds3} (from sweeping M_3) is improved if we can find a design point which achieves better performance controlled by g_{m6} (from sweeping M_6). Table 4.8 shows an example that two points from sweeping M_3, M_6 have close g_{m6} value, thus the first

design point is able to improve based on sequence $\{g_{ds3} \rightarrow g_{m6}\}$. Figure 4.7 shows a summary of design points from the synthesis tool by sweeping transistors M_3, M_6, M_7, M_{10} under weight [0.5, 0.5]. From the figure we observe that, transistors M_3, M_{10} (at the output stage) tend to cover more evenly over the solution space, whereas transistors M_6, M_7 (at the input stage) are more clustered in high Gain/Bandwidth/THD area.

Table 4.8: Design points improved by sequence gds3-gm6

M_0	M_1	M_{10}	M_{11}	M_{13}	M_{14}	M_3	M_4	M_6	M_7
0.0001782	0.00009047	0.000006016	0.000002177	0.000003055	0.000008442	0.00012	0.00001	0.000008442	0.00001185
vb1	vb2	vb3	vb4	Gain	BW				
2.7	2.233	1.183	0.7	69.22	8558				
gds0	gds1	gds10	gds11	gds13	gds14	gds3	gds4	gds6	gds7
0.00001975	0.00001364	0.000003142	0.00001282	0.000006714	0.0001007	0.000005806	0.000009405	0.000005106	0.00000377
gm0	gm1	gm10	gm11	gm13	gm14	gm3	gm4	gm6	gm7
0.001258	0.0006086	0.0003432	0.0003438	0.0003185	0.0007138	0.0004712	0.0003268	0.0005806	0.0002418
M_0	M_1	M_{10}	M_{11}	M_{13}	M_{14}	M_3	M_4	M_6	M_7
0.0001782	0.00003274	0.000004287	0.00001662	5.613E-07	0.000008442	0.00001185	0.000008442	0.000015	0.000004287
vb1	vb2	vb3	vb4	Gain	BW				
2.7	2.35	1.3	0.7167	69.66	8586				
gds0	gds1	gds10	gds11	gds13	gds14	gds3	gds4	gds6	gds7
0.00002542	0.00000566	0.000004532	0.0000428	0.000001753	0.00001611	0.000001447	0.000003293	0.00000745	0.00000139
gm0	gm1	gm10	gm11	gm13	gm14	gm3	gm4	gm6	gm7
0.001232	0.0002213	0.0001539	0.001275	0.00007937	0.0008931	0.0001486	0.000167	0.0008548	0.00008733

Besides the high priority orders that calculated in Table 4.7, during the design process, we may also consider other orders. For order $\{g_{m6} \rightarrow g_{ds3} \rightarrow g_{ds6} \rightarrow g_{m3} \rightarrow g_{ds10} \rightarrow g_{m10} \rightarrow g_{m7} \rightarrow g_{ds7}\}$ in {Gain, Bandwidth}, among high linearity parameters g_{m6}, g_{ds3}, g_{ds6}, we could switch g_{m6}, g_{ds3} to have $\{g_{m6} \rightarrow g_{ds3} \rightarrow g_{ds6} \rightarrow g_{m3} \rightarrow g_{ds10} \rightarrow g_{m10} \rightarrow g_{m7} \rightarrow g_{ds7}\}$, in Figure 4.14. If we start from g_{ds3} (sizing transistor M_3), following order $\{g_{ds3} \rightarrow g_{m6} \rightarrow g_{ds6} \rightarrow g_{m3} \rightarrow g_{ds10} \rightarrow g_{m10} \rightarrow g_{m7} \rightarrow g_{ds7}\}$, we size the circuit by $\{M_3 \rightarrow M_6 \rightarrow M_{10} \rightarrow M_7\}$, in Figure 4.12. Or we can size the circuit by $\{M_{10} \rightarrow M_6 \rightarrow M_3 \rightarrow M_7\}$, in Figure 4.13, $\{M_6 \rightarrow M_3 \rightarrow M_{10} \rightarrow M_7\}$, in Figure 4.14. Different sequences indicate different sizing patterns, corresponding to different partial solution space on the Pareto front. The design sequence starting from M_3 achieves slightly worse Gain, better Bandwidth than sequence starting from M_{10}. The design sequence starting from M_6 achieves much better Gain

but worse Bandwidth. Design sequence beginning with M_7 is not considered as a feasible solution as M_7 ranks low in linear control, correlation, and uniqueness. Thus, parameter g_{m6} (at the input stage) is better at achieving relatively high Gains, whereas g_{ds3}, g_{ds10} (at the output stage) have advantages in higher Bandwidth. The three different design sequences result three different solution space on the Pareto front.

Figure 4.15 shows the design sequence for weight [0.25, 0.75]. It achieves slightly less Gain than weight [0.5, 0.5] as Gain has less priority, and draws the same conclusion as weight [0.5, 0.5].

For performance attribute set {Gain, THD}, parameters g_{m7}, g_{ds7} are weakly correlated with performances. g_{m6}, g_{ds6} rank low in linear control, correlation, and uniqueness. Several design sequence results are shown in Figure 4.16, 4.17. M_{10} indicates slightly higher Gain and worse THD than M_3.

For performance attribute set {Gain, Bandwidth, THD}, in Figure 4.18, 4.19, 4.12, parameters g_{ds10}, g_{m10} result different design sequence even though they all start with M_{10}. Starting with g_{ds10} achieves slightly higher Gain, less Bandwidth, higher THD. Besides the high priority order, other orders in Figure 4.21, 4.22 indicate g_{m3} achieves close performance as g_{ds10}, g_{m6} achieves better Gain, worse Bandwidth, worse THD.

Table 4.9: Sizing improvement case study

Case	Gain [dB]	BW [Hz]	THD [%]	M0 [μm]	M1 [μm]	M10 [μm]	M11 [μm]	M13 [μm]	M14 [μm]	M3 [μm]	M4 [μm]	M6 [μm]	M7 [μm]
0	64.26	5.61k	9.64	23.33	6.02	10	8.44	3.06	1.11	23.33	8.44	10	6.02
1(1)	70.29	4.29k	7.15	62.92	15.84	10	3.99	2.52	10	10	15.84	17.42	4.6
1(2)	66.16	13.4k	8.94	2.52	10	2.52	62.92	6.31	3.99	10	6.31	5.74	52.85
2(1)	69.13	4.47k	6	62.92	10	7.17	1.59	2.52	10	10	2.52	21.75	1.11
2(2)	67.44	16.99k	12.5	39.73	10	7.168	39.73	10	39.73	21.75	10	21.75	11.17
2(3)	66.79	9.12k	9.17	128.4	65.99	7.17	5.74	11.17	10	21.75	65.99	21.75	11.17

Case study: Sequenced causal trace indicates an ordered trade-off effect of parameter *vs* performance, which can be used to improve automated sizing as a parameter preference list. The traditional way to size op-amp 4(1) is through sampling all transistor dimensions equally without considering

trade-off effects, e.g., sampling each transistor 20 times evenly along 0.4μm $\sim$ 250μm. One sizing solution is shown on row 0 in Table 4.9.

The causal trace sequence for performance attributes set {Gain, Bandwidth, THD} in Table 4.7 is {$g_{m10} \rightarrow g_{ds10} \rightarrow g_{m3} \rightarrow g_{m6} \rightarrow g_{ds6} \rightarrow g_{ds3} \rightarrow g_{m7} \rightarrow g_{ds7}$}. It indicates devices {M_{10}, M_3, M_6, M_7} are more correlated to performance attributes, and a preferred order to size the circuit is {$M_{10} \rightarrow M_3 \rightarrow M_6 \rightarrow M_7$}. Thus, we propose adaptive scheme to sample {M_{10}, M_3, M_6, M_7} more frequently than the other devices and add constraint $width_{M10} = width_{M3}$. The newly generated design point is shown in Table 4.9 row 1(1), $width_{M10} = width_{M3} = 10\mu$m. With the constraint of $width_{M10} = width_{M3} = 10\mu$m, we sample {$M_6$, M_7} more frequently to run a second round optimization. The result on row 1(2) achieves better performance in all attributes {Gain, Bandwidth, THD}, i.e., dominates the traditional solution on row 0. Thus, adaptive sampling scheme based on causal trace sequence is able to improve automated sizing to achieve better performance.

Other causal trace sequence in Table 4.7, {$g_{ds10} \rightarrow g_{ds6} \rightarrow g_{m6} \rightarrow g_{m3} \rightarrow g_{m10} \rightarrow g_{ds7} \rightarrow g_{m7} \rightarrow g_{ds3}$}, shows an order of {$M_{10} \rightarrow M_6 \rightarrow M_3 \rightarrow M_7$}. Similarly, another iterative optimization with adaptive sampling achieves better sizing solution as shown on rows 2(1), 2(2), 2(3) in Table 4.9.

Besides adaptive sampling scheme in sizing optimization, the parameter sequence is can also be utilized in the following ideas, which can be studied in future work, (*i*) The ordered causal trace (parameter sequence) can be reused to size the same circuit in newer process. (*ii*) For higher dimensional specifications (dimension > 4), it is more difficult for the synthesis tool to obtain a feasible sizing solution. In this case, the proposed method is able to obtain high dimensional trade-off results by reusing and aggregating low dimensional data. (*iii*) In design reuse, for a new circuit with minor modification, the causal trace can be used to size the circuit as a reference starting point.

4.5 Conclusions

Characterizing the causality model of a circuit is critical for analog circuit design automation activities. Generating new circuit topologies or incrementally updating an existing circuit to meet different requirements can use the causal information to identify which devices and building blocks can tackle the specification, hence be part of a new circuit, or which of the structures produce the performance bottlenecks of a circuit, so that they are replaced with features that do not cause these limitations. Second, the causal information can be used to verify the validity of an analog circuit by verifying which of the causality defined by the connected building blocks are consistent with each other (i.e., they do not annual each others advantages), capable of achieving the needed specification, and optimal with respect to other causal possibilities achieved by using alternative circuit topologies. Third, the quality of circuit sizing can be improved by using the parameter causality to decide the parameter sets that are most effective in addressing the performance trade-offs of a circuit for a specification.

This chapters model causal information by six elements respectively: (i) Causal relation characterizes the connection between a design parameter, performance attributes, and the other circuit parameters. (ii) Causal trace presents the comprehensive causal influence of a parameter for different importance (weights) of the performance attributes. (iii) A causal graph includes the traces of all circuit parameters, all performance attribute sets. (iv) Sequenced causal relations, (v) graph of sequenced causal relations present the order in which the causal relations of the parameters are used in tackling a design problem. Finally, (vi) causal Pareto front is formed from original optimal design points and different causal relation orders. Experimental results illustrate performance multi-dimensional trade-off effect and different reasoning strategies in automatic sizing.

While exploring a broader performance space, the work is limited by large execution time of the sizing tool. A more efficient sampling strategy can be

studied as future work. Various parameter sequences are generated, whereas a systematic selection method to aid the sizing process can be explored too.

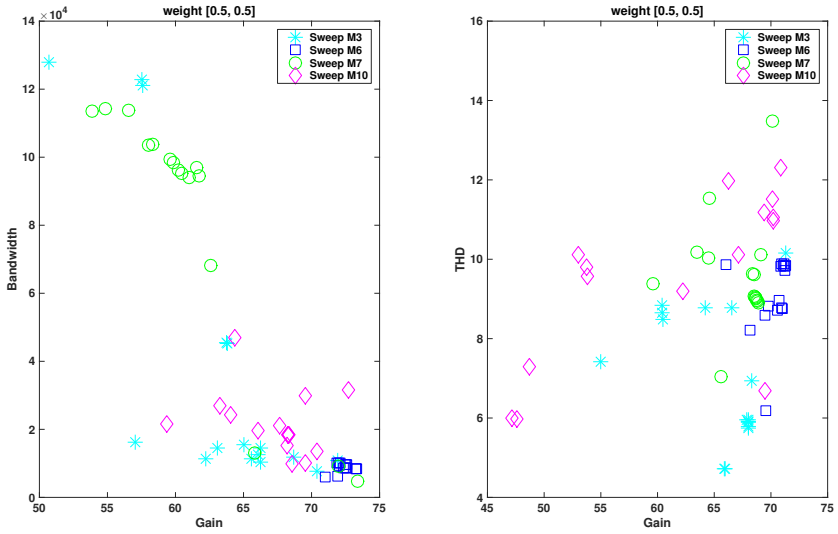


Figure 4.7: Sweep M_3, M_6, M_7, M_{10} under [0.5, 0.5]

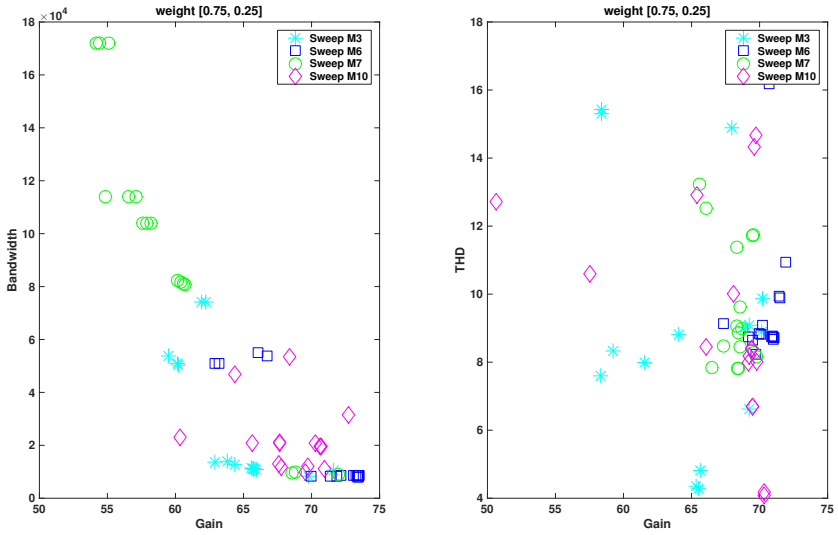


Figure 4.8: Sweep M_3, M_6, M_7, M_{10} under [0.75, 0.25]

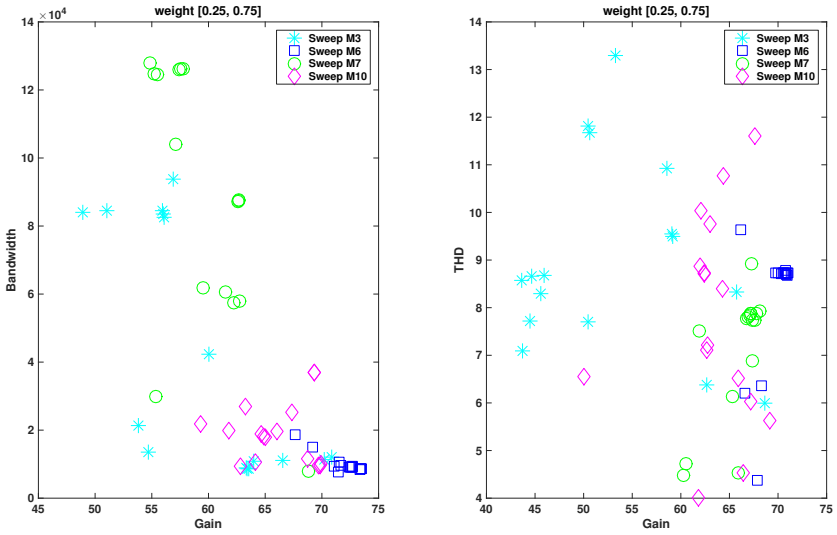


Figure 4.9: Sweep M_3, M_6, M_7, M_{10} under [0.25, 0.75]

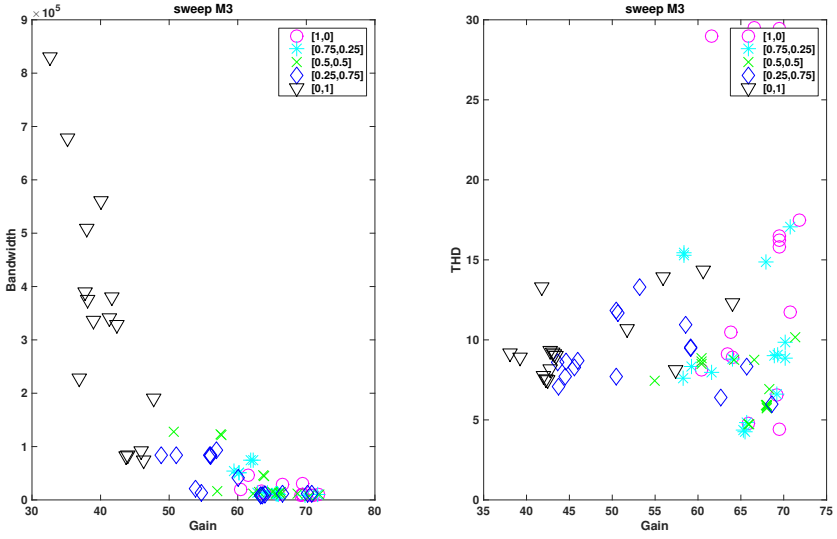


Figure 4.10: Sweep M_3 under different weights

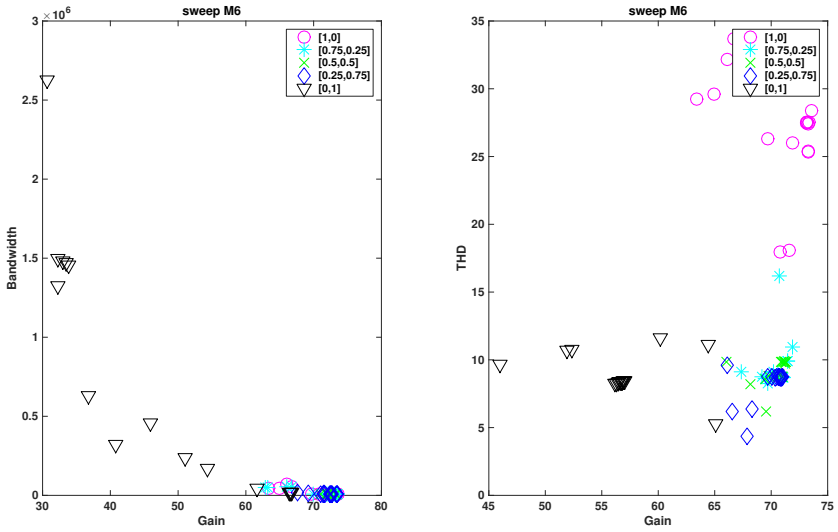


Figure 4.11: Sweep M_6 under different weights

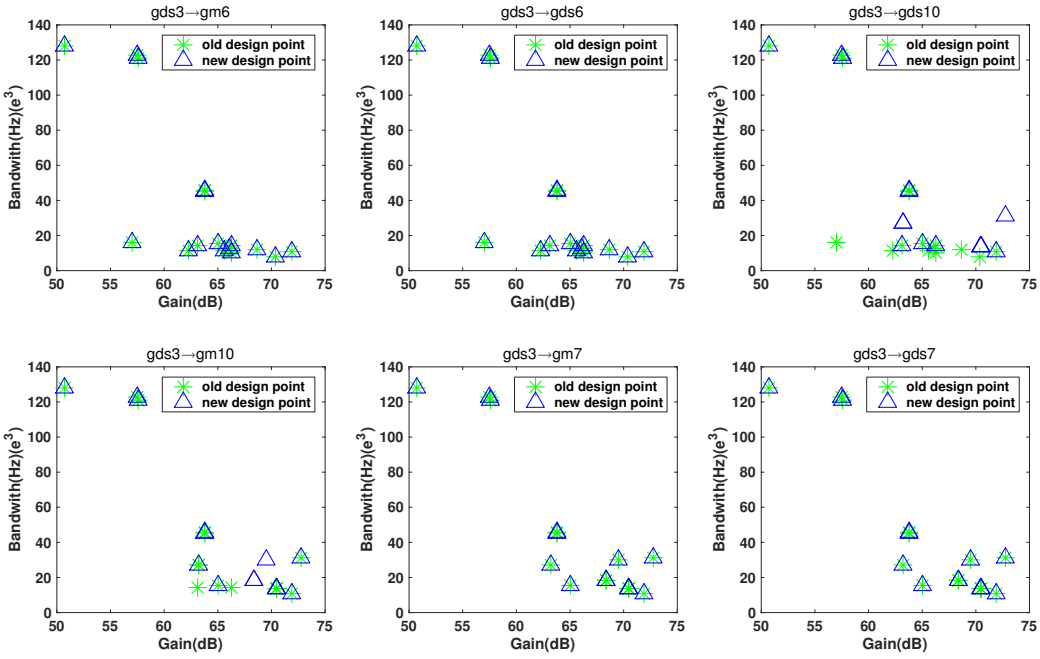


Figure 4.12: Gain-Bandwidth design order in table 1

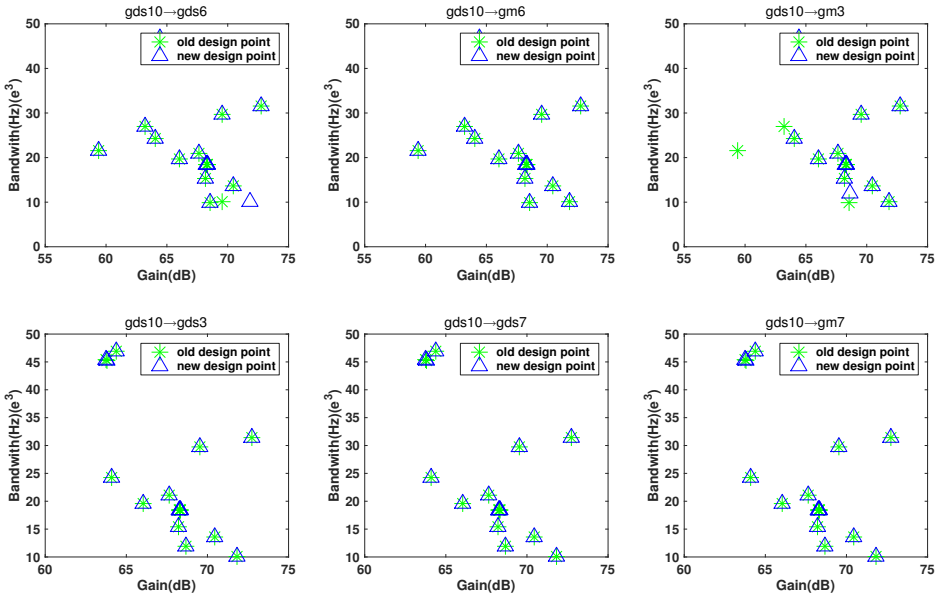


Figure 4.13: Gain-Bandwidth design order in table 2

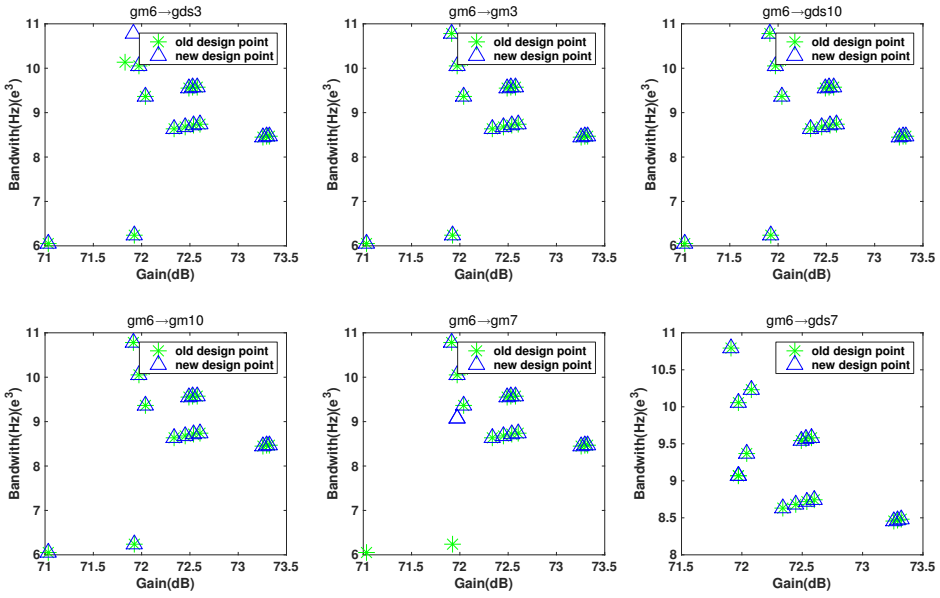


Figure 4.14: Gain-Bandwidth other order example

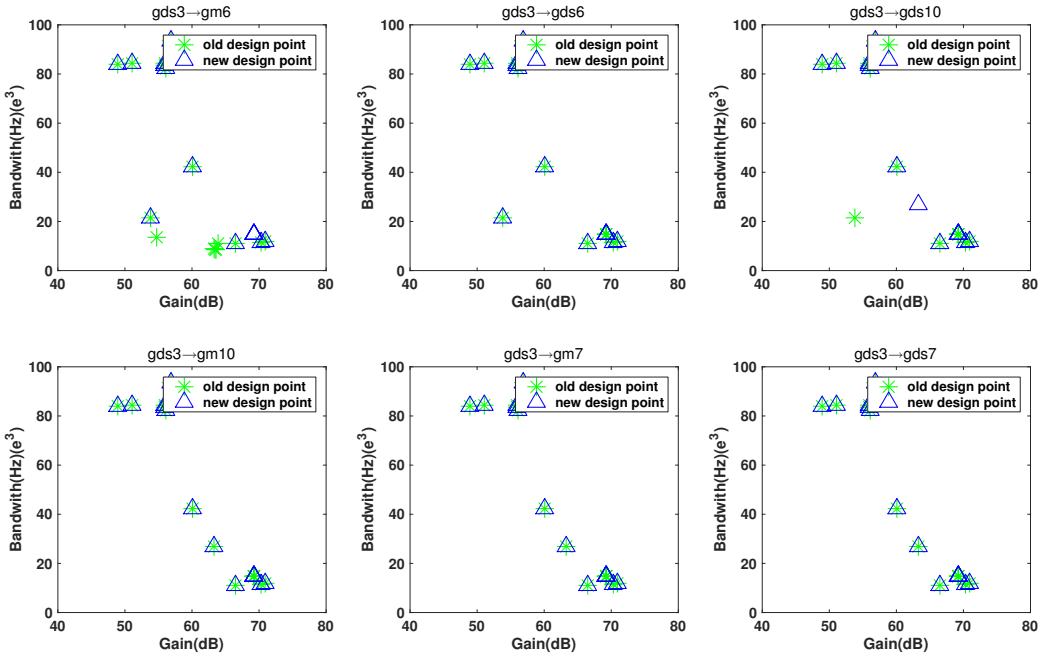


Figure 4.15: Gain-Bandwidth other order example

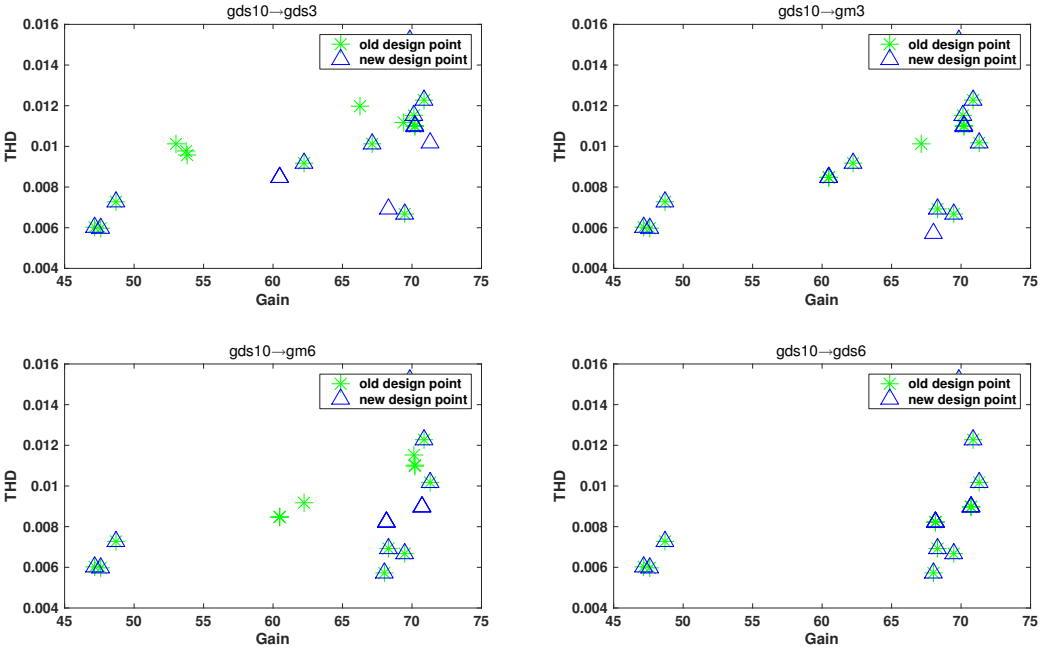


Figure 4.16: Gain-THD design order 1

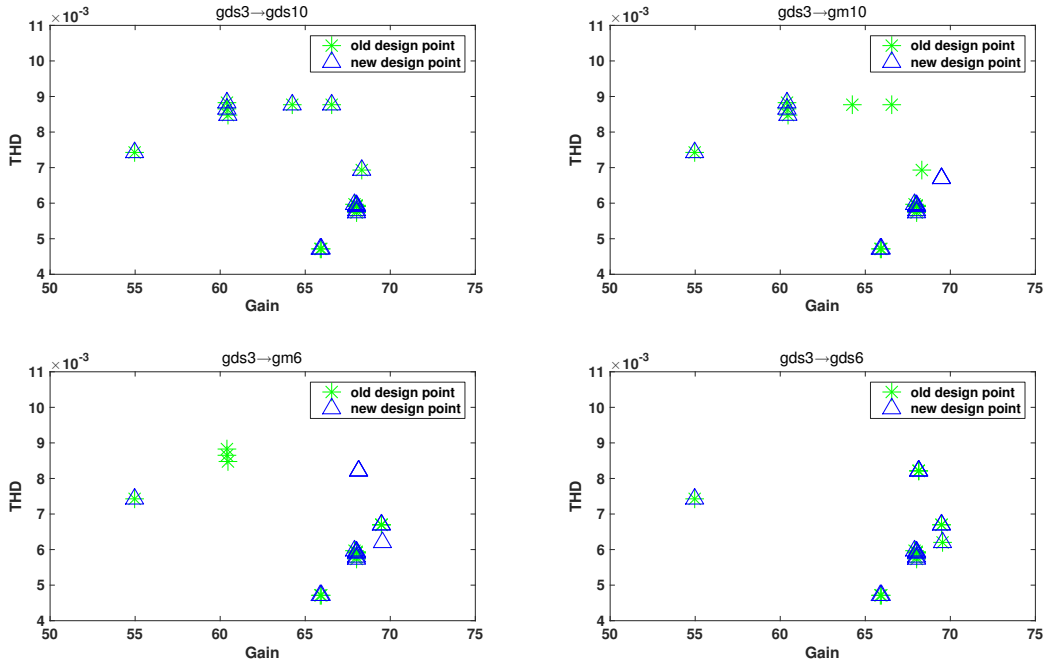


Figure 4.17: Gain-THD design order 2

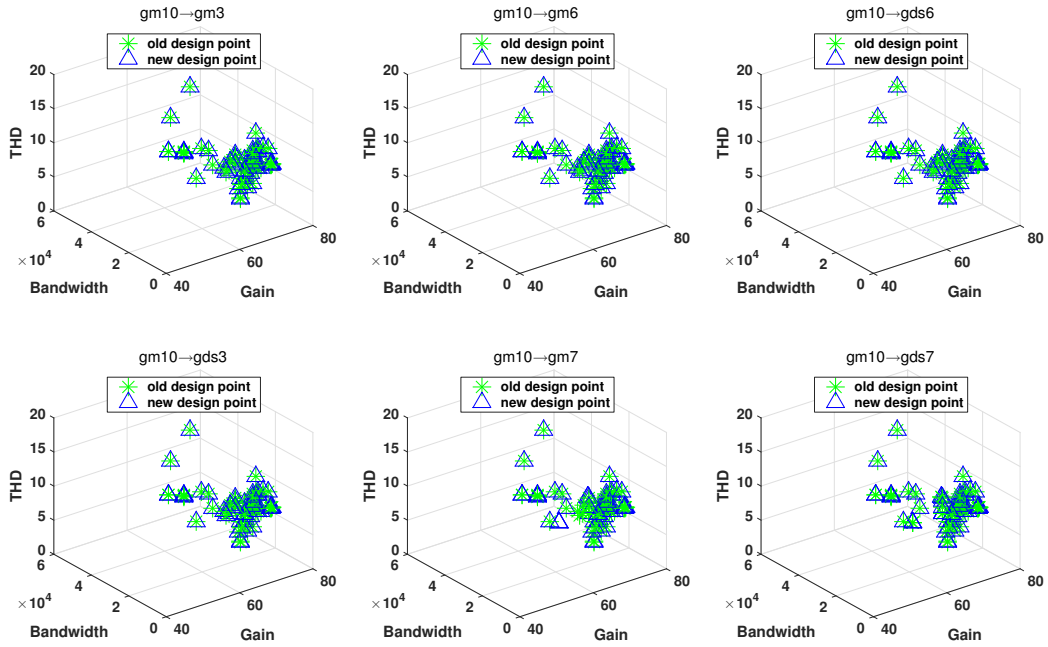


Figure 4.18: Gain-Bandwidth-THD design order in table 1

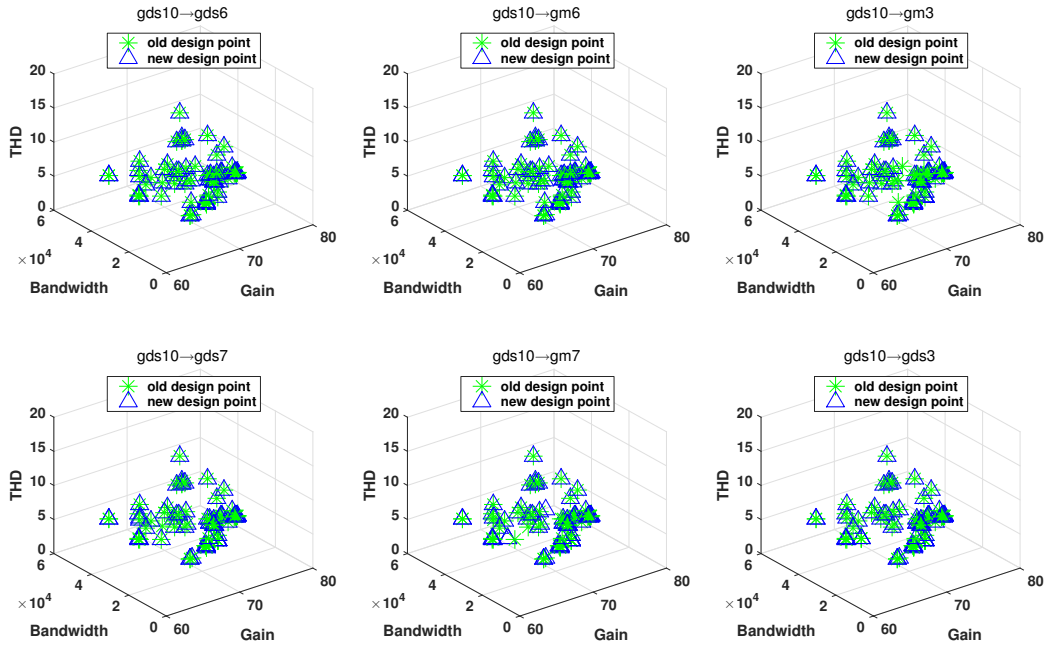


Figure 4.19: Gain-Bandwidth-THD design order in table 2

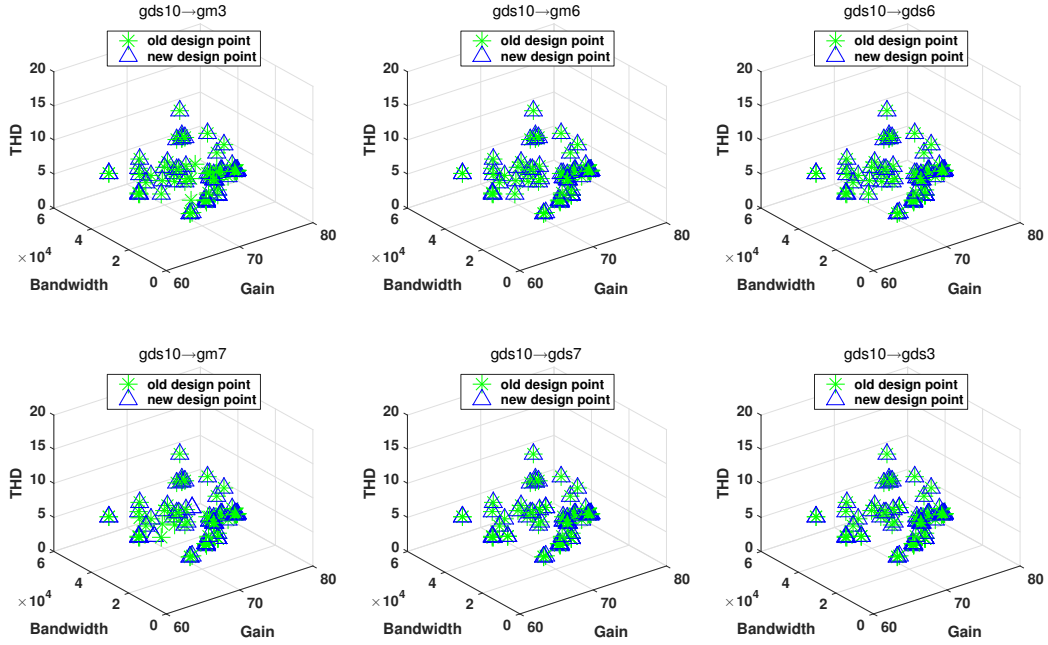


Figure 4.20: Gain-Bandwidth-THD design order in table 3

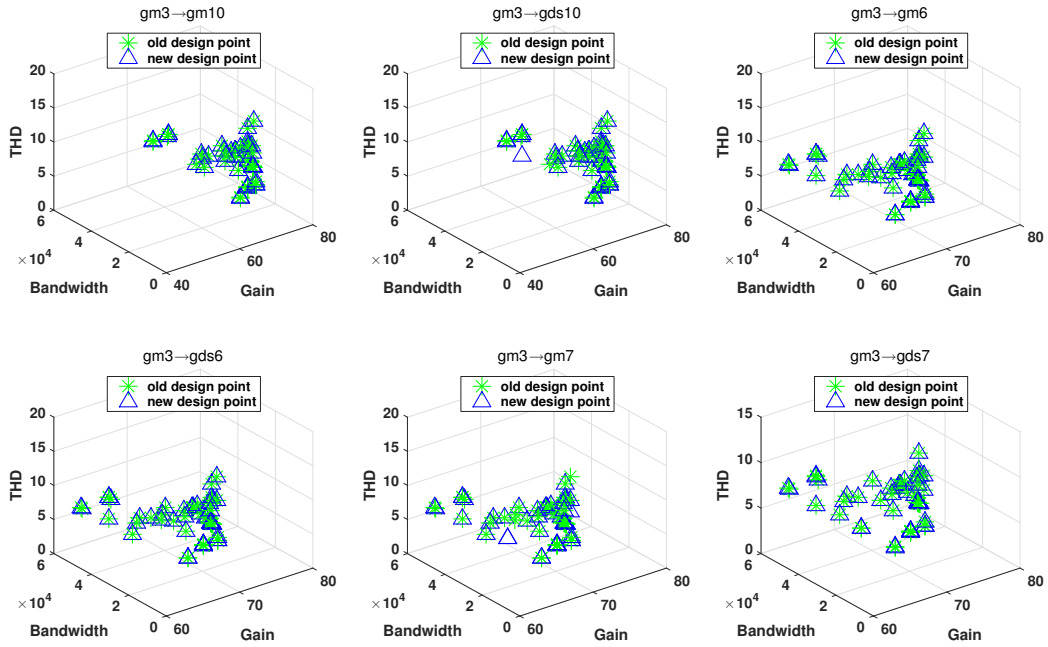


Figure 4.21: Gain-Bandwidth-THD other design order 1

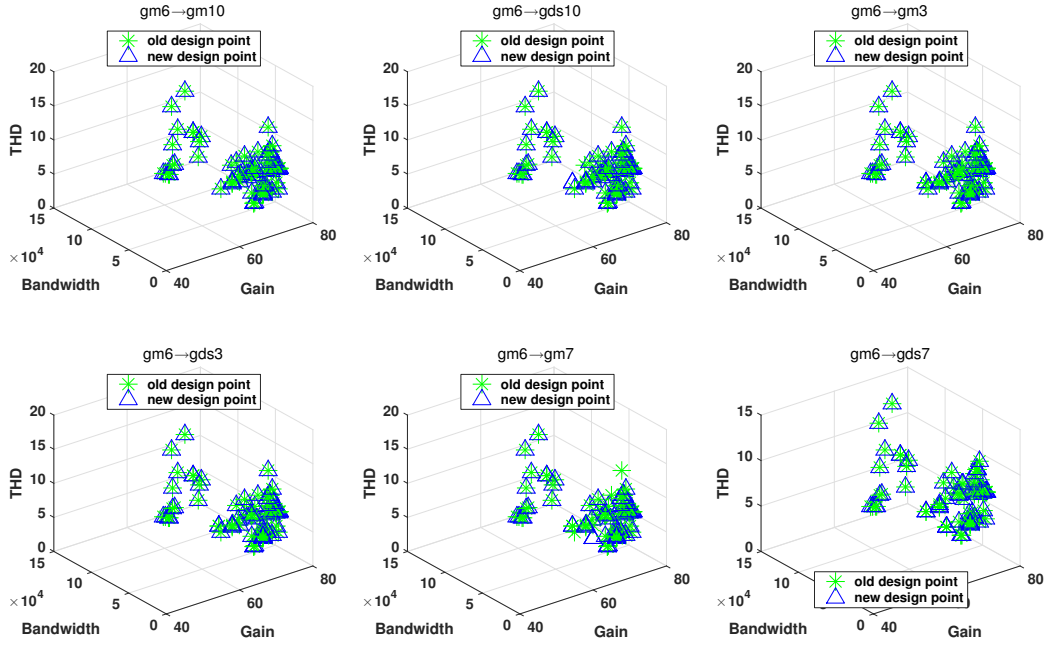


Figure 4.22: Gain-Bandwidth-THD other design order 2

Chapter 5

Reasoning-based Design Verification[1]

5.1 Introduction

Analog design verification verifies functional performance on analog, mixed-signal and RF circuits, which has been a critical step in design flow. Modern analog circuits in advanced CMOS technologies are more complex and with more demanding performance constraints. In traditional system on chips (SoCs), analog and mixed-signal circuitry take up 20% of the area, but this percentage is expected to steadily increase to half of the chip area in SoCs designed for advanced nodes [116]. An increasing number of functional errors and unmet performance requirements occur in the analog portion of large mixed-signal integrated chips [117].

Design refinement and redesigns are more severe in order to meet time-to-market requirements, hence driving the need for consistent analog verification methodologies and tools. Analog verification requires performing functional verification on analog, mixed-signal and RF integrated circuits and SOCs, which still depends largely on designer's experience and exper-

[1] [114], [115]

tise. For digital systems, there are various reliable verification methods, i.e., formal verification of RTL. These methods do not directly apply to analog circuits, not only because analog signals process continuously in value and time, but also the complexity of analog performance specifications.

In Chapter 2, design knowledge mining (especially causal reasoning component) provides intuitive reasoning procedure, which arguably offers the diagnostics insight of the design flow (i.e., the reasons that cause the performance errors). Inspired by causal reasoning to generate new circuit in Chapter 3, this chapter also applies it to verify existing circuit. While existing methods, simulation-based verification and formal verification, aim at covering and verifying the entire performance space, they arguably offer less diagnostics insight, i.e., the reasons that cause the performance errors. By mining and understanding design reasoning, we are able to utilize the information to verify functional errors and potentials for improvement.

With the mined causal reasoning in Chapter 2, this chapter proposes a new reasoning-based verification approach as an application of design knowledge mining. The verification method checks the correctness of the starting ideas and design sequence that form a design plan. Incorrect functional issues and performance potentials for improvement are identified and linked to either starting ideas and/or design sequence. The result of verification offers diagnostics insight, like the starting idea and/or design step implementation that should be modified to correct functional errors.

The chapter is organized as follows: Section 5.2 discusses the related work. Section 5.3 presents the overview of the verification methodology. Section 5.4 presents the algorithm. Section 5.5 presents two experiments on verifying opamps to show how design issues and improvement potentials are identified. Conclusions in Section 5.6 end the paper.

5.2 Related work

Recent research on analog circuit verification can be distinguished into two categories: simulation-based methods and formal verification methods [118].

Simulation-based methods verify a system or a circuit design at various levels of abstraction of the analog or mixed-signal (AMS) domain. Transient simulation, a main tool in AMS verification, generally requires long simulation time and large computing resources. Simulation time can be reduced by using behavioral models of the circuits [21, 119, 120]. Existing behavioral modeling methods include black box modeling [27, 121], white box or structural modeling [21, 120], base-band modeling, and event-driven modeling [122].

For RF circuits, there are specific, enhanced simulation methods, i.e., periodic steady-state analysis and harmonic balance. However, these methods are not applicable to AMS circuits which are not periodic [123]. Base-band modeling isolates the modulated data from the carrier frequency. The main issue with base-band modeling is using different pin definitions while switching between different abstraction levels. Base-band modeling tools like SPW [124], SpectreRF [125] and simulator AMS designer [126] support the verification of a RF subsystem at different levels of abstractions. [127] uses enhanced base-band behavioral models for an industrial-level multi-band, low IF GSM receiver. Event-driven modeling approach isolates the high-frequency signal path from the low-frequency bias and control by using different simulation domains. The event-driven simulation approach in [128] verifies RF front-ends using a new double precision data type (called wreal) in Verilog-AMS. It separates high frequency signal paths and saves time by enabling fast analog verification using the digital simulation domain. Simulation-based verification is widely used, but limited by the fact that it is hard to identify input patterns that guarantee the covering of all errors that might occur in the whole design space.

Formal verification methods, though have not been widely applied to

AMS and RF design verification, can mathematically verify a circuit's operation and/or performance space. Existing formal verification methods include model checking, reachability analysis, equivalence checking, and runtime verification. Performance properties are described in the steady-state space. A first attempt to formulate analog simulation as a Boolean satisfiability (SAT) problem is proposed in [129]. [130] extends the SAT-based modeling method to parameter variations. Conventional temporal logic model for AMS circuit verification is discussed in [131]. It applies finite state machines to represent continuous state space of transient response. [132] proposes two symbolic model checking algorithms, a binary decision diagram-based model checker and a symbolic modular-based model checker. Both model checkers utilize a Boolean symbolic model derived from the labeled hybrid petri net model. It includes Boolean signals to represent digital circuitry and continuous variables to model voltages and currents of analog circuits. Besides various other model checking approaches, [133] introduces dynamic stability verification by zonotope-based reachability analysis. Steinhorst, et al [134] present an algorithm for behavioral equivalence checking of two circuit implementations for dynamic state space. Model-checking and reachability analysis validate the design over a range of parameters and input possibilities, therefore suffers from expensive computation [135]. [136] presents a runtime verification algorithm using a time-augmented rapidly-exploring random tree for incremental runtime monitoring.

Current simulation-based and formal verification methods mainly rely on intensive simulation and focus on verifying circuit states, but arguably offer less diagnostics insight. Diagnosis of analog and mixed-signal circuits still heavily relies on the engineer's experience and intuition because of poor fault models, component tolerances, and nonlinearity issues [137]. Existing automation methods range from expert systems, built-in self-test circuits to pattern recognition methods. Early method using expert systems [138] requires sufficient knowledge and many test points. Other approaches intro-

duced alternative built-in self-test circuit [139], or pattern recognition to associative patterns in the test response data with the corresponding faults [140]. [141] applies neural network to classify faults considering parametric variations. [142] proposes fault model by fuzzy math. It combines both sensitivity analysis and fuzzy analysis for parametric fault diagnosis. This chapter aim to help diagnose the reasons in circuit design plan that cause circuit performance error/drawback.

5.3 Methodology for reasoning-based design verification

Each analog circuit design follows certain design reasoning procedure, which can be extracted by using the procedure in Chapter 2 applied to the circuit design literature. Mining design reasoning identifies possible design features that served as starting ideas in devising new circuits as well as the design steps and their justifications in creating the complete designs. The proposed reasoning-based verification flow in Figure 5.1 begins with mining design causal reasoning for individual circuit design. Circuit topological features corresponding to starting ideas, design steps are identified, replaced with ideal behavior model. Re-evaluating the newly generated circuit reveals possible improvement, therefore offers diagnostic information regarding certain starting idea or design step.

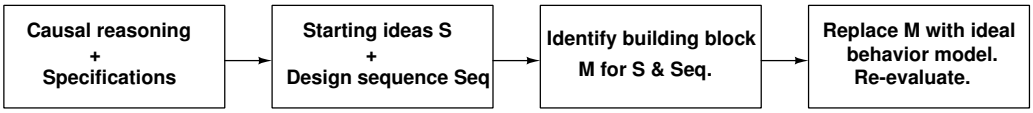


Figure 5.1: Causal reasoning-based verification flow

5.3.1 Mining design causal reasoning

As explained in Chapter 2, the main elements describing design causal reasoning include starting ideas, design steps, and justifications. Mining algorithm is presented in Chapter 2, Section 2.6.

5.3.2 Circuit verification using causal reasoning

Given a sized circuit, the verification flow first compares performance simulation results with specifications. In the experiment, circuit sizing is obtained by Cadence advanced analysis optimization tool [5]. Performance specifications are the objectives for optimization, e.g., Table 5.2. Performances are evaluated through Spectre simulator. Starting from the mined causal reasoning information, the topological features in the starting ideas and design sequence are individually verified. Features are verified from starting ideas to design sequence as starting ideas are considered more important than the design sequence. Figure 5.2 shows an illustrating example. For each topological feature x, the corresponding related devices are identified as a building block. Each block performs certain function and might introduce drawback insight. Replacing the block with an ideal functional model removes the drawbacks and sometimes functional errors in the circuit.

A two terminal Verilog-AMS behavioral amplifier model [143] is chosen as the ideal functional model. It has only two parameters and simplifies the verification process. Two model parameters are gain and input offset. Gain is set as the same with the replaced block to keep the new circuit having the same gain as the original. Input offset is set to zero.

Multiple circuits are generated, with each corresponds to the new circuit replacing one building block. We conclude from newly evaluated simulation results that for each performance p, if p is marked as functional error but p is solved in the new circuit F using the ideal functional model, feature x (belonging to starting ideas or design sequence) is the reason (or part of

reason) for p not meeting the specification. Otherwise, if performance p is relaxed in F, it indicates that we can look for better candidate of feature x to improve p.

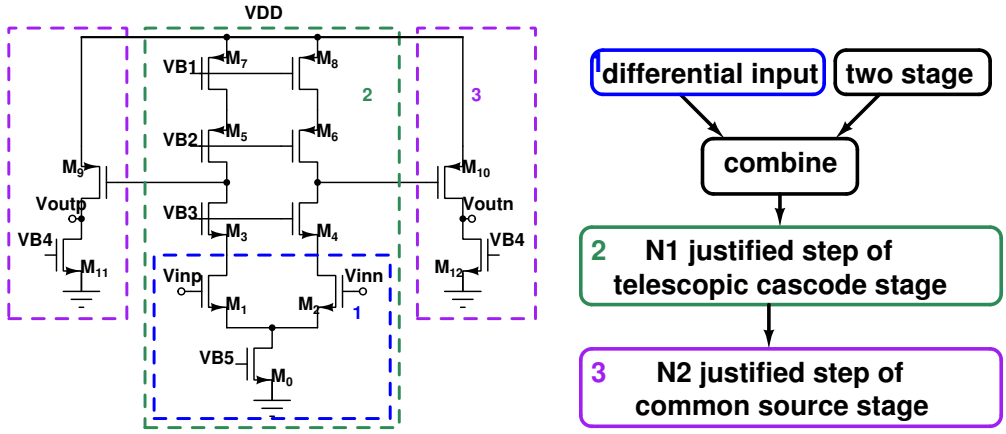


Figure 5.2: Two-stage op-amp causal reasoning

5.4 Design verification algorithm

The reasoning-based verification method is shown in Algorithm 12. It links circuit functional errors or performance potentials for improvement with the corresponding starting ideas/design steps in the causal reasoning component. Verification results are a set of functional errors, potentials for improvement, and the features in design that produced them (the diagnostics information).

```
input  : sized analog circuit C, starting ideas S, design
         sequence Seq;
output: functional errors and potentials for improvement;
begin;
for each performance p of circuit C do
    if simulation result of p does not meet specification then
        mark p as "functional error";
    else
        mark p as "performance potentials";
    end
end
for each topological feature x in S & Seq do
    identify the related devices {M_i, M_j, ... M_n} of feature x
     as block M;
    replace block M from C with ideal model to get new
     circuit F;
    evaluate F performance;
    for each performance p of circuit F do
        if p in C is marked with "functional error" &
         simulated result of p in F meets specification then
            Link "functional error" of p with feature x;
        end
        if p in C is marked with "non-linearity" & simulated
         result of p in F is greatly improved then
            Link "performance potentials" of p with feature x;
        end
    end
end
end procedure;
```

Algorithm 12: Algorithm for reasoning-based verification

Example: Figure 5.2 shows the schematic and causal reasoning component of a two stage op-amp circuit [49]. The complete topological features (building blocks) of the circuit (set Σ) = {differential input, telescopic cascode stage, common source stage}, labeled 1, 2, 3 in Figure 5.2. The initial starting ideas (set S) = {an abstract feature two-stage}, which has been used in previous design as indicated by the author [49] (set Γ). Beginning with the initial starting idea in set S, the uncovered features = Σ − S.

In this case, set Σ and S don't overlap, as the designer used the concept of two-stage but decided his own physical implementations. "Two-stage" is analyzed further to derive design sequence. {Telescopic cascode stage, common source stage}, the two-stage implementations, labeled as 2 and 3, are mined as design sequence. Therefore, design step N_1 implements the first gain stage as telescopic cascode stage, justified by its advantages in high gain. Design step N_2 implements the second gain stage as common source stage, as it offers high output voltage swing. The remaining features, differential input, labeled 1, is common design knowledge (set Θ). Therefore, it is added to set S as the starting ideas. Set S includes all starting ideas of the circuit and the identified design sequence includes all features in set Σ − S, hence it produces the final design. Figure 5.2 illustrates the mined starting ideas and the design step sequence.

The op-amp circuit is sized using Cadence advanced analysis optimization tool [5], 0.2μm CMOS technology. All transistor L dimensions are chosen 3 times the minimum size. Table 5.1 lists the optimized sizing solution. The verification flow first compares performance simulation results with specifications to identify functional errors or potentials. Starting from causal reasoning, topological features corresponding to starting ideas and design sequence are verified by replacing feature x with an ideal functional model. A two terminal Verilog-AMS behavioral amplifier model [143] is chosen as the ideal functional model. It has only two parameters and thus simplifies the verification process. Two model parameters are gain and input offset. Gain

is set as the same with the replaced block to keep the new circuit having the same gain as the original. Input offset is set to zero. Replace feature x generates new circuit F. For newly evaluated performance p, if p is marked as functional error but p is solved in F, then feature x (belonging to starting ideas or design sequence) is the reason (or part of reason) for p not meeting the specification. Otherwise, if performance p is relaxed in F, it indicates that we can look for better candidate of feature x to improve p.

The overall performance set = {static power, DC gain, bandwidth, gain-bandwidth product (GBP), slew rate, rmsNoise, total harmonic distortion (THD), output voltage swing}. Two figures of merit FOM_S and FOM_L were used to characterize small-signal and large-signal performances [63]. All circuits have load capacitor C_{load} = 10pF. Therefore, we considered to verify the performance set {FOM_S, Phase Margin, FOM_L, rmsNoise, THD, output voltage swing}. Specifications were given empirically. Columns "Specs", "Optimized results" in Table 5.2 show that phase margin, FOM_L and noise don't meet the specification, which is shown in bold and marked as "functional error".

The verification flow for each topological feature in the starting idea and design sequence in Figure 5.2, labeled as 1, 2, and 3, is as follows.

$$FOM_S = \frac{GBP * C_L}{power}, FOM_L = \frac{SR * C_L}{power} \tag{5.1}$$

Table 5.1: Transistor sizing for the op-amp circuit in Figure 5.2

devices	M_0	M_1/M_2	M_3/M_4	M_5/M_6	M_7/M_8	M_9/M_{10}	$\text{M}_{11}/\text{M}_{12}$
optimized sizing [μ/μ]	15/0.6	69.75/0.6	90/0.6	14.6/0.6	26.8/0.6	1.1/0.6	0.3/0.6

Feature 1. Differential input is identified as devices {M_0, M_1, M_2} in Figure 5.2. Replacing {M_0, M_1, M_2} with the ideal amplifier model (shown as a triangle) generates a new circuit F_1 in Figure 5.3(a). The ideal model has the same gain with the block being replaced. The input offset is set to 0V. Column "F_1" in Table 5.2 summarizes the newly evaluated performances

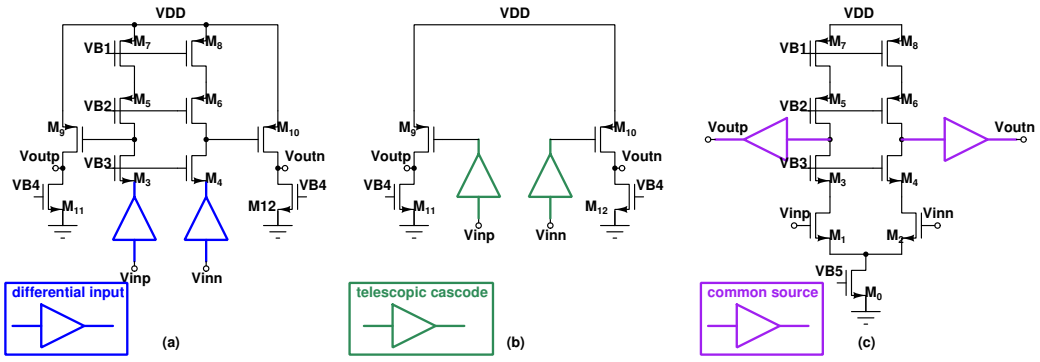


Figure 5.3: Replace circuit features in Figure 5.2 with models

Table 5.2: Performance comparison of circuits in Figure 5.2, 5.3

Performance	**Specs**	**Optimized results**	$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$
Static power[μw]	minimize	405.072	405.096	8.212	396.84
DC gain[dB]	maximize	89.79	89.98	89.93	89.39
Bandwidth [kHz]	maximize	6.245	6.125	6.153	2.921e3
GBP [MHz]	maximize	193.2	193.7	193.6	86.3e3
FOM$_S$ [$\frac{MHz*pF}{mw}$]	>100	4769.5	4781.6	**235752.6**	**2174680**
Phase Margin[°]	[60, 90]	**3.4**	13.32	**81.68**	>90
Pos Slew Rate [V/μs]	>1	0.46	0.47	0.38	17.36e3
FOM$_L$ [$\frac{V/\mu*pF}{mw}$]	>100	**11.32**	11.51	467.85	**437455.9**
rmsNoise [1-1MHz band]	<10	**27.82**	**4.75**	**0.63**	32.76
THD [%]	<10	1.35	1.11	1.09	1.75
Output voltage swing [V]	>1.2	1.6	1.6	1.6	1

of circuit F_1. Evaluating each performance in set {FOM$_S$, phase margin, FOM$_L$, rmsNoise, THD, and output voltage swing}, phase margin is slightly improved, but still does not meet the specification. Thus, feature differential input introduces one non-dominant pole. Noise in the new circuit is greatly reduced. Thus we conclude that feature differential input introduces most of the noise. If improvement on noise is needed, we will focus on input stage and look for better candidate. Topological feature differential input, as one of the starting ideas, introduces one non-dominant pole, thus is not the reason for having insufficient phase margin.

Feature 2. The telescopic cascode stage is implemented as the first gain stage in the amplifier. The new circuit F_2 in Figure 5.3(b) is generated by

replacing the related devices $M_0 \sim M_8$. In circuit F_2, phase margin meets the constraint, so one dominant pole introduced by the first stage is eliminated. FOM_S and rmsNoise improved significantly as the new circuit replaced the entire first stage, including devices in differential input.

Feature 3. The second gain stage of the amplifier is implemented by a common source stage. Replacing the related devices $M_9 \sim M_{12}$ results in the new topology in Figure 5.3(c). Comparing performance results, phase margin > 90 means gain will not drop to zero within the considered frequency range. FOM_L is greatly improved. We can conclude that the second stage introduces one dominant pole. Also, the second stage greatly constrains the small signal and large signal performance.

In summary, the two-stage telescopic op-amp suffers from insufficient phase margin, low large signal performance and large noise. By verifying each topological feature in the design reasoning flow, we conclude there are two dominant poles in the circuit which cause the phase drop. One is after the telescopic cascode stage and one is at the output load. With respect to the three topological features in the starting ideas and design sequence, the two-stages implementation in the design sequence needs to be modified. Additional feed-forward stage or frequency compensation is required to ensure stability. The starting idea on using differential input contributes most on noise, and the idea of using the output stage constrains most large signal performance (slew rate).

5.5 Experiments

Experiments illustrate the proposed verification flow for two more advanced op-amps/OTAs.

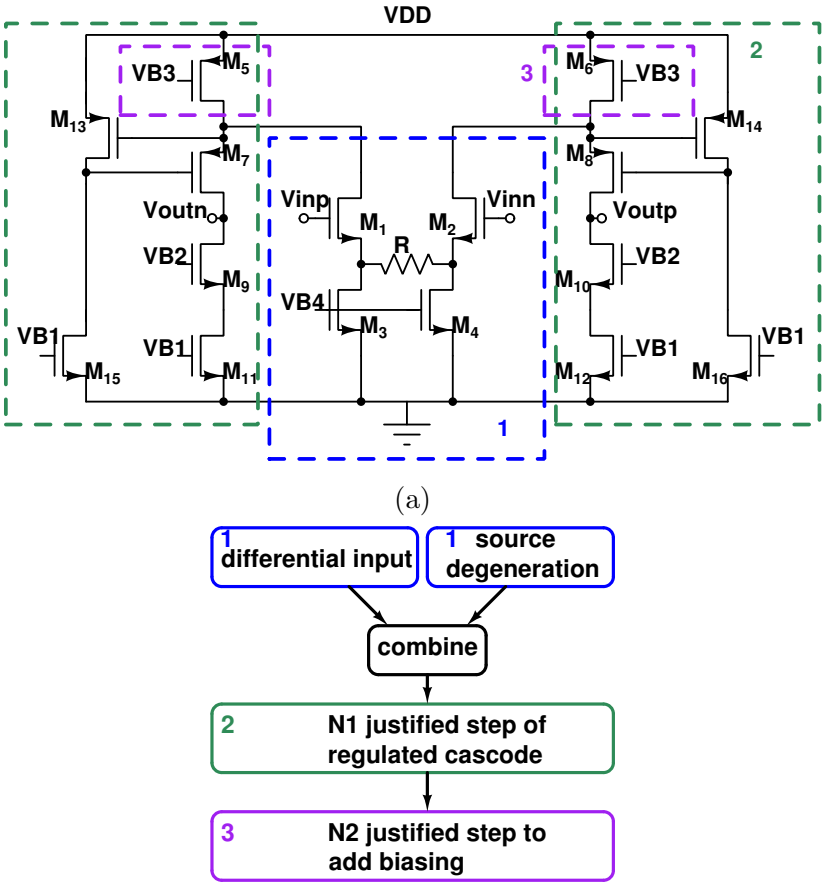


Figure 5.4: Two-stage OTA causal reasoning

5.5.1 Highly linear OTA

The circuit in Figure 5.4a is a single differential highly linear OTA with degeneration resistance [67]. Set Σ of the circuit = {differential input, resistance degeneration, regulated cascode, and current source biasing}. Set S = {source degenerated differential input}, which is referred from previous work (set Γ). It is labeled as 1 in Figure 5.4a. It offers high linearity by using higher source degeneration factors. The uncovered features (Σ − S) are labeled as 2 and 3. The uncovered features, {regulated cascode, current

source biasing}, if justified, are added to the design sequence of the circuit. Source degenerated differential input achieves higher linearity than regular differential input but limits the gain, thus regulated cascode is used at the output. Design step N_1 adds regulated cascode, justified by the starting idea to boost gain. Current source biasing is a required design step for the circuit's correct operation. Figure 5.4b illustrates the mined starting ideas and the design sequence. From the performance comparison in Table 5.3, output voltage swing does not meet specification, and are marked as functional errors. Phase margin, rmsNoise, and THD are labeled performance potentials.

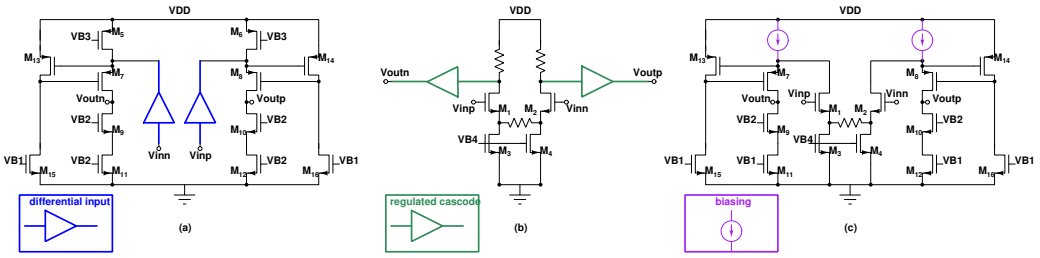


Figure 5.5: Replace circuit features in Figure 5.4a with models

Table 5.3: Performance comparison of circuits in Figure 5.4a, 5.5

Performance	**Specs**	**Optimized results**	**F_1**	**F_2**	**F_3**
Static power[μw]	minimize	1.017e3	1.017e3	456	1.25e3
DC gain[dB]	maximize	38.16	38.23	30.17	33.04
Bandwidth [kHz]	maximize	382.2	14.89e3	468.9e6	69.78
GBP [MHz]	maximize	30.99	1.217e3	15.15e3	3.14
FOM_S [$\frac{MHz*pF}{mw}$]	>100	304.65	**11961.86**	**332236.84**	25.11
Phase Margin[°]	[60, 90]	88.8	67.9	>90	91.11
Pos Slew Rate [V/μs]	>1	13.88	12.67	8.293e3	3.908
FOM_L [$\frac{V/\mu*pF}{mw}$]	>100	136.45	124.533	**181864.04**	31.25
rmsNoise [1-1MHz band]	<10	0.3394	0.02711	0.09976	1.836
THD [%]	<10	1.011	1.048	1.071	0.9783
Output voltage swing [V]	>1.2	**1.2**	1.2	**2**	1.2

Feature 1. The topological structures in the starting idea, differential input and source degeneration, are identified as one building block, $M_1 \sim$

M_4, and resistor R. The circuit with ideal model is shown in Figure 5.5(a). FOM_S is greatly improved, but the output voltage swing is still low. The input stage implementation is part of the reasoning for FOM_S improvement. The other performances do not have a significant improvement, thus the input stage does not introduce limitations on other performances.

Feature 2. The design step uses regulated cascode as the output stage. The new circuit in Figure 5.5(b) solves the functional error. The design step corresponding to the regulated cascode stage is the mainly reason for the small- and large signal performance improvement. It also dominates the output voltage swing. Same as with the input stage, it does not introduce constrains on other performances.

Feature 3. Step N_2 is a required design step. Biasing transistors M_5 and M_6 are modeled by the ideal current sources in Figure 5.5(c). According to Table 5.3, it does not have a major effect on performance. The simulation errors come from the transistor non-idealities working as current sources, which is not for verification purposes.

For low voltage application, using the regulated cascode implementation as the output stage should be avoided. Both gain stages cause not sufficient FOM_S and FOM_L, whereas the output stage dominates more on FOM_L.

5.5.2 High-speed, high-swing op-amp

Another circuit example in Figure 5.7a is a high-speed, high-swing op-amp used in low-voltage applications [69]. From the circuit topology, we extracted five topological features (building blocks) = {differential input single ended output, gain-boosted amplifier, folded cascode, low voltage current mirror, and current source biasing} (set Σ). These five topological features are labeled as $1 \sim 5$ in Figure 5.7a. According to the design description in [69], the reasoning process begins with ideas used previously, which is gain boosted feedback amplifier (set Γ). For the remaining features in $\Sigma - S$, folded cascode is justified by the feedback amplifier. The folded cascode

with feedback path is one strategy to enhance gain with additional power consumption. The step of adding current source biasing is a required design step. Differential input single ended output is a common input-output structure (set Θ). Meanwhile, low voltage current mirror is one way to connect differential input and single output stage, which is a justified design step. The causal reasoning in Figure 5.7b combines the starting ideas of differential input, single ended output and gain-boosted amplifier (set Γ), labeled as 1 and 2. Design step N_1 implements the input stage using folded cascode. Design step N_2 adds a low-voltage current mirror justified by the single-ended output. Design step N_3 adds the current source biasing for the circuit's correct operation. They are labeled as 3, 4, and 5 in Figure 5.7a.

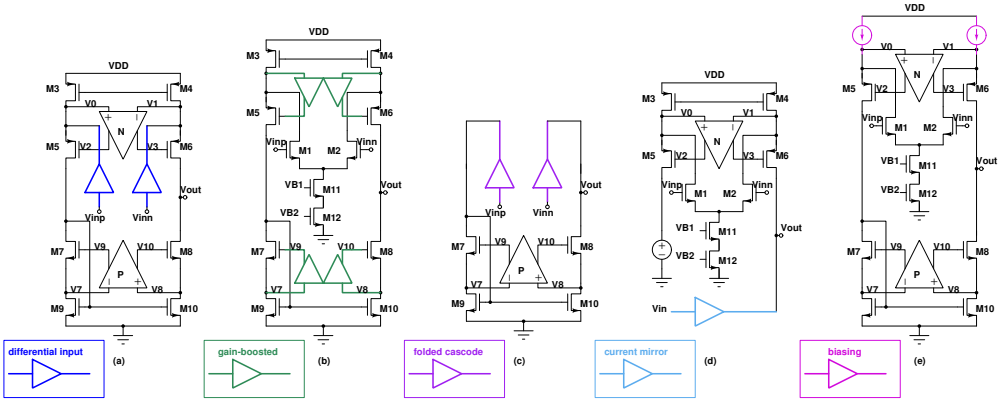
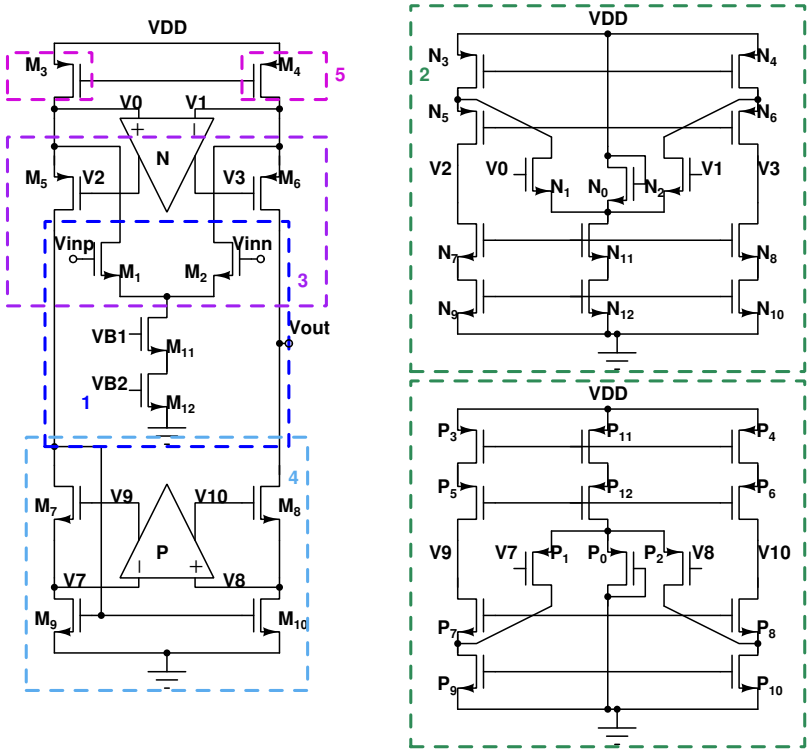


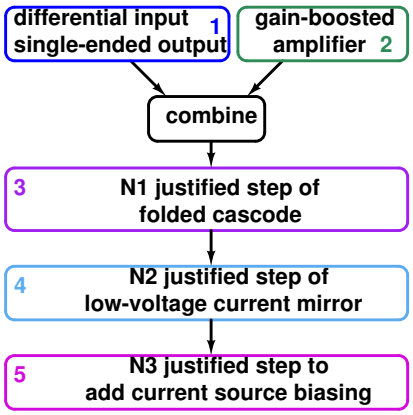
Figure 5.6: Replace circuit features in Figure 5.7a with models

As shown in Table 5.4, rmsNoise and output voltage swing are unmet performances. Figure 5.6 presents five new circuits generated by replacing each feature in set starting ideas and design sequence. From column "F_1", small signal and large signal performance are improved greatly, whereas noise is not improved.

Design steps of using folded cascode and low voltage current mirror, and the starting idea of using gain-boosted amplifier are the main reasons for the unsatisfied noise requirement. Steps of folded cascode and low-voltage current mirror determines output voltage swing, which is easy to conclude



(a)



(b)

Figure 5.7: High-speed and high-swing op-amp causal reasoning

Table 5.4: Performance comparison of circuits in Figure 5.7a, 5.6

Performance	**Specs**	**Optimized results**	$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$	$\mathbf{F}_4$	$\mathbf{F}_5$
Static power[μw]	minimize	994.44	994.44	738.8	0	738.8	752
DC gain[dB]	maximize	95.3	95.06	15.09	95.3	79.6	97.7
Bandwidth [kHz]	maximize	0.508	34.23	1023	$+\infty$	$+\infty$	0.3872
GBP [MHz]	maximize	29.64	1943	5.831	$+\infty$	$+\infty$	29.77
FOM$_S$ [$\frac{MHz*pF}{mw}$]	>100	298.06	**19538.63**	78.93	$+\infty$	$+\infty$	395.88
Phase Margin[°]	[60, 90]	86.01	75	106.4	>90	>90	86.9
Pos Slew Rate [V/μs]	>1	17.13	1517	0.0885	23.28e9	1198	17.3
FOM$_L$ [$\frac{V/\mu*pF}{mw}$]	>100	172.26	**15254.82**	1.65	$+\infty$	**16215.48**	230.05
rmsNoise [1-1MHz band]	<10	**240.1**	199.3	**0.076**	**0**	**0**	273.6
THD [%]	<10	0.99	1.527	61.28	0.3	0.226	0.99
Output voltage swing [V]	>1.2	**1**	1	**2**	1	**2**	1

as they form the output stage. Differential input introduces performance drawbacks on small- and large signal performance.

The presented verification examples identify functional errors, potential for improvement and their reasons, e.g., incorrect starting ideas or design steps (diagnostics) for a sized circuit. The insight obtained from verification examples can be used as a reference for further design. The proposed technique is not able to verifying circuit states, hence, it serves as a complement to existing methods.

5.6 Conclusions

This chapter presents a new analog circuit design verification approach based on mining causal reasoning. The method verifies circuit design plan to conclude functional errors and performance potentials for improvement and their causes (diagnostics).

The verification flow begins with mining the causal reasoning steps (design plan) that produced the circuit, including starting ideas, design step sequence, and their justifications. Topological features corresponding to the starting ideas and design step sequence are verified individually by replacing the related devices (transistors, capacitors, resistors, etc) with ideal behavior model. New circuits are generated and evaluated through simulation. Com-

paring simulation results with the new generated circuits reveals incorrect functional issues and/or performance potential for improvement. They are negative causes of certain starting ideas or design steps, which might have been omitted during the design process.

Three experiments on op-amp/OTA designs show that functional errors and performance potentials are identified as either incorrect starting design ideas or erroneous design steps in the design plan. Automating the verification process includes extracting circuit building blocks, mining causal reasoning information, and evaluating new circuits with replaced ideal model. For performance not meeting specification, experimental results identify certain topological features implemented, which are traced back to starting idea or design sequence.

The purpose of reasoning-based verification is to diagnose, thus, it is not able to verify circuit's complete performance states and operation. Limitation of the method lies in the need to mine causal reasoning component. Replacing features with behavior model and re-evaluating the circuit requires relatively large simulation effort. Future work can improve the evaluation process, e.g. behavioral level simulation.

Chapter 6

Conclusions

In this thesis, analog circuit design knowledge mining and causal information modeling methodologies have been presented. The development of knowledge mining and causal modeling to extract analog design knowledge from literature is motivated by the gap between existing automation methods and manual design. It is inspired by (*i*) Advances in basic techniques in statistic data analysis, machine learning, and data mining. (*ii*) Development of knowledge discovery in databases (KDD) to identify valid, novel, potentially useful, and ultimately understandable information in data.

6.1 Design knowledge mining

6.1.1 Summary

Knowledge mining utilizes the knowledge-intensive feature of analog circuit design and brings new perspectives to design automation field. It mines design knowledge from three aspects and tends to cover the entire knowledge space: an associative component presenting the conceptual hierarchy of the considered circuits, a component expressing the performance capabilities (e.g., trade-offs and bottlenecks) of the circuits, and a causal modeling

component describing the most likely starting ideas and design plans used to create a circuit. The associative component groups circuits into hierarchical abstractions based on the symbolic similarities of the instances. Starting from the performance trade-offs and bottlenecks of the circuits, performance capabilities component express the trade-offs of concepts by combining the trade-off tables of their instances. Finally, the causal reasoning component identifies the initial ideas that support finding sequences of justified design steps for a given design.

The proposed knowledge structure can be used to tackle new applications (e.g., by selecting or refining a circuit topology), identify new design opportunities (by analyzing the combination of design features that have never been used together before), In some cases, the found starting ideas were probably the initial ideas of the designer, e.g., when topological features were combined to create a new circuit. However, other starting ideas are only the originators of the design step sequences that create a circuit, but not the actual initial ideas of the designer. An essential aspect of devising a representation to support reasoning-based analog circuit synthesis is explicitly presenting the causal information on how topological features and their parameters decide performance trade-offs and bottlenecks.

Besides aid in designers' effort by knowledge organization/construction, the knowledge mining method is applied to automate op-amp/OTA design. A new topology synthesis method is proposed that implements a design knowledge-intensive, reasoning-based process to create novel circuit structures with all their feature justified by the problem requirements. Five methodologies are described depending on the nature of the starting design features. Four synthesis experiments demonstrate the capability of the method to generate circuits beyond the capabilities of existing topology synthesis algorithms. The synthesized topologies are similar to designer created circuits. Even though we do not have formal evidence, we think that such circuits are hard to be evolved by genetic algorithms.

In order to validate design correctness by showing that all steps in starting ideas and design sequence are justified. The novel reasoning-based analog circuit verification technique was developed based on verifying the mined circuit design knowledge. It helps to identify functional errors and performance potentials for improvement and their causes (diagnostics). The verification flow begins with mining the causal reasoning steps (design plan) that produced the circuit, including starting ideas, design step sequence, and their justifications. Topological features corresponding to the starting ideas and design step sequences are verified individually by replacing the related devices (transistors, capacitors, resistors, etc) with ideal behavior model. Comparing simulation results with the new generated circuits reveals incorrect functional issues and/or performance potential for improvement. They are negative causes of certain starting ideas or design steps, which might have been omitted during the design process.

6.1.2 Limitations and future work

In general, knowledge mining in terms of building association and extracting design reasoning is technology independent. With associative component focuses on topology attributes in terms of nodes, node couplings, higher level building blocks, and templates, technology aspects are not explored. Whereas feasibility of topology can be a technology dependent decision, when new technology emerges, new design constraints of topological features need to be updated. As for performance trade-off tables, simulation needs to be conducted again with different technology, which requires additional simulation cost. Linking topologies to different processes can be considered in future work, i.e., identifying topologies particular for advanced processes. The method to adapt topology to newer process can be explored too.

Experiment shows knowledge representation of 34 high performance op-amps/OTAs from literature on analog circuit design. Beside op-amp/OTA design, the methodology also works on other analog blocks at the same level

or higher (i.e., LNA, filter, oscillator) in general. Future work may include knowledge mining on higher level analog circuits.

6.2 Causal information modeling

6.2.1 Summary

With respect to finding important parameters to various performance output at a global scale, Chapter 5 extends performance capabilities by modeling circuit causal information, including the relations of design parameters deciding the values of the functional outputs, performance, and other parameters. Considering the limitation and availability of symbolic computation, causal information modeling utilizes simulation tool instead. Causal information includes: (*i*) Causal relation, expresses the parameter's influence in controlling performance values and trade-offs. (*ii*) Causal trace, presents the comprehensive causal influence of a parameter for different importance (weights) of the performance attributes. (*iii*) Causal graph, includes the traces of all circuit parameters, all performance attribute sets. (*iv*) Sequenced causal relations, (*v*) graph of sequenced causal relations, present the order in which the causal relations of the parameters are used in tackling a design problem, which are further used to generate Pareto front.

Modeling circuit causal information aids in the design automation in three aspects, (*i*) generating new circuit topologies or incrementally updating an existing circuit to meet different requirements can use the causal information to identify which devices and building blocks can tackle the specification, hence be part of a new circuit, or which of the structures produce the performance bottlenecks of a circuit, so that they are replaced with features that do not cause these limitations. (*ii*) the causal information can be used to verify the validity of an analog circuit by verifying which of the causality defined by the connected building blocks are consistent with each other (i.e., they do not annual each others advantages), capable of achieving the needed

specification, and optimal with respect to other causal possibilities achieved by using alternative circuit topologies. (*iii*) the quality of circuit sizing can be improved by using the causality parameters to decide the parameter sets that are most effective in addressing the performance trade-offs of a circuit for a specification.

Experimental results illustrate performance multi-dimensional trade-off effects and different reasoning strategies in automated sizing.

6.2.2 Limitations and future work

While exploring a broader performance space, the work is limited by large execution time of the sizing tool. A more efficient sampling strategy can be studied as future work. Furthermore, causality measure can be improved in future work for a better sizing strategy representation. Various parameter sequences are generated in the experiment, whereas a systematic selection method to aid the sizing process can be explored too.

Bibliography

[1] G. G. Gielen and R. A. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1825–1854, 2000.

[2] H. Lampinen and O. Vainio, "An optimization approach to designing otas for low-voltage sigma-delta modulators," *Instrumentation and Measurement, IEEE Transactions on*, vol. 50, no. 6, pp. 1665–1671, 2001.

[3] M. Krasnicki, R. Phelps, R. A. Rutenbar, and L. R. Carley, "Maelstrom: efficient simulation-based synthesis for custom analog cells," in *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*. ACM, 1999, pp. 945–950.

[4] T. Massier, H. Graeb, and U. Schlichtmann, "The sizing rules method for cmos and bipolar analog integrated circuit synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 12, pp. 2209–2222, 2008.

[5] "Cadence advanced analysis tools user guide," pp. 162–207, 2002.

[6] S. E. Sorkhabi and L. Zhang, "Automated topology synthesis of analog and rf integrated circuits: A survey," *Integration, the VLSI Journal*, vol. 56, pp. 128–138, 2017.

[7] A. J. López-Martín, S. Baswa, J. Ramirez-Angulo, and R. G. Carvajal, "Low-voltage super class ab cmos ota cells with very high slew rate and power efficiency," *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 5, pp. 1068–1077, 2005.

[8] R. Harjani, R. A. Rutenbar, and L. R. Carley, *Analog circuit synthesis for performance in oasys*. Springer, 2003.

[9] M. Meissner and L. Hedrich, "Feats: Framework for explorative analog topology synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 2, pp. 213–226, 2015.

[10] C. M. Bishop, "Pattern recognition," *Machine Learning*, vol. 128, 2006.

[11] G. Cowan, *Statistical data analysis*. Oxford University Press, 1998.

[12] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*. Elsevier, 2011.

[13] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, p. 37, 1996.

[14] K. Kaufman and R. Michalski, "From data mining to knowledge mining," *Handbook in Statistics*, vol. 24, no. 1, 2005.

[15] R. S. Michalski, "Knowledge mining: A proposed new direction," in *Invited talk at the Sanken Symposium on Data Mining and Semantic Web, Osaka University, Japan*, 2003.

[16] S. Dehuri and S.-B. Cho, *Knowledge mining using intelligent agents*. World Scientific, 2010, vol. 6.

[17] D. Sánchez, M. J. Martín-Bautista, I. Blanco, and C. J. de la Torre, "Text knowledge mining: an alternative to text data mining," in *2008*

IEEE International Conference on Data Mining Workshops. IEEE, 2008, pp. 664–672.

[18] G. editorial, "Enabling technologies and methodologies for knowledge discovery and data mining in smart grids," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 820–823, 2016.

[19] F. Jiao, S. Montano, C. Ferent, and A. Doboli, "I-flows: A novel approach to computational intelligence for analog circuit design automation through symbolic data mining and knowledge-intensive reasoning," in *Computational Intelligence in Analog and Mixed-Signal (AMS) and Radio-Frequency (RF) Circuit Design*. Springer, 2015, pp. 3–28.

[20] F. Jiao, S. Montano, C. Ferent, A. Doboli, and S. Doboli, "Analog circuit design knowledge mining: Discovering topological similarities and uncovering design reasoning strategies," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 34, no. 7, pp. 1045–1058, 2015.

[21] Y. Wei and A. Doboli, "Structural macromodeling of analog circuits through model decoupling and transformation," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 27, no. 4, pp. 712–725, 2008.

[22] A. Chowdhary, S. Kale, P. Saripella, N. Sehgal, and R. Gupta, "A general approach for regularity extraction in datapath circuits," in *Computer-Aided Design, 1998. ICCAD 98. Digest of Technical Papers. 1998 IEEE/ACM International Conference on*. IEEE, 1998, pp. 332–339.

[23] S. Hassoun and C. McCreary, "Regularity extraction via clan-based structural circuit decomposition," in *Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design*. IEEE Press, 1999, pp. 414–419.

[24] H. Liu, A. Singhee, R. A. Rutenbar, and L. R. Carley, "Remembrance of circuits past: macromodeling by data mining in large analog design spaces," in *Design Automation Conference, 2002. Proceedings. 39th.* IEEE, 2002, pp. 437–442.

[25] G. Wolfe and R. Vemuri, "Extraction and use of neural network models in automated synthesis of operational amplifiers," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 22, no. 2, pp. 198–212, 2003.

[26] T. McConaghy, T. Eeckelaert, and G. Gielen, "Caffeine: Template-free symbolic model generation of analog circuits via canonical form functions and genetic programming," in *Design, Automation and Test in Europe, 2005. Proceedings.* IEEE, 2005, pp. 1082–1087.

[27] S. Doboli, G. Gothoskar, and A. Doboli, "Piecewise-linear modeling of analog circuits using trained feed-forward neural networks and adaptive clustering of hidden neurons," in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, vol. 2. IEEE, 2003, pp. 1126–1131.

[28] S. Pandit, S. K. Bhattacharya, C. Mandal, and A. Patra, "A fast exploration procedure for analog high-level specification translation," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 27, no. 8, pp. 1493–1497, 2008.

[29] F. Wang, P. Cachecho, W. Zhang, S. Sun, X. Li, R. Kanj, and C. Gu, "Bayesian model fusion: Large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 8, pp. 1255–1268, Aug 2016.

[30] J. Tao, C. Liao, X. Zeng, and X. Li, "Harvesting design knowledge from the internet: High-dimensional performance tradeoff modeling

for large-scale analog circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 1, pp. 23–36, 2016.

[31] D. Boolchandani, C. Gupta, and V. Sahula, "Analog circuit feasibility modeling using support vector machine with efficient kernel functions," 2009.

[32] T. Kiely and G. Gielen, "Performance modeling of analog integrated circuits using least-squares support vector machines," in *Proceedings of the conference on Design, automation and test in Europe-Volume 1*. IEEE Computer Society, 2004, p. 10448.

[33] M. Ding and R. Vemuri, "A two-level modeling approach to analog circuit performance macromodeling," in *Proceedings of the conference on Design, Automation and Test in Europe-Volume 2*. IEEE Computer Society, 2005, pp. 1088–1089.

[34] T. McConaghy, P. Palmers, G. Gielen, and M. Steyaert, "Automated extraction of expert knowledge in analog topology selection and sizing," in *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*. IEEE, 2008, pp. 392–395.

[35] T. McConaghy, *Variation-aware analog structural synthesis: a computational intelligence approach*. Springer, 2009.

[36] C. Ferent and A. Doboli, "Novel circuit topology synthesis method using circuit feature mining and symbolic comparison," in *Proceedings of the conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2014, p. 17.

[37] M. El-Gohary, J. McNames, T. Ellis, and B. Goldstein, "Time delay and causality in biological systems using whitened cross-correlation analysis," in *Engineering in Medicine and Biology Society, 2006.*

EMBS'06. 28th Annual International Conference of the IEEE. IEEE, 2006, pp. 6169–6172.

[38] Z. Zhou, Y. Jiao, T. Tang, Z. Lu, Y. Liu, Y. Chen, and M. Ding, "Detecting effective connectivity in human brain using granger causality," in *BioMedical Engineering and Informatics, 2008. BMEI 2008. International Conference on*, vol. 2. IEEE, 2008, pp. 394–398.

[39] C. Chen, A. Maybhate, D. Israel, N. V. Thakor, and X. Jia, "Assessing thalamocortical functional connectivity with granger causality," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 21, no. 5, pp. 725–733, 2013.

[40] S. Hu, G. Dai, G. A. Worrell, Q. Dai, and H. Liang, "Causality analysis of neural connectivity: critical examination of existing methods and advances of new methods," *Neural Networks, IEEE Transactions on*, vol. 22, no. 6, pp. 829–844, 2011.

[41] M. Azri bin Mohd and A. Nawawi, "Causality linkages between usa and asian islamic stock markets," in *Business, Engineering and Industrial Applications (ISBEIA), 2011 IEEE Symposium on*. IEEE, 2011, pp. 123–128.

[42] V. Kreinovich and A. Ortiz, "Towards a better understanding of space-time causality: Kolmogorov complexity and causality as a matter of degree," in *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*. IEEE, 2013, pp. 1349–1353.

[43] H. Simon, *Causal ordering and identifiability. In Hood, W.C.?Koopmans, T.C. (Eds.), Studies in Econometric Method. Cowles Commission for Research in Economics*. John Wiley and Sons Inc., 1953, vol. Monograph No. 14.

[44] P. Nayak, *Causal approximations. Artificial Intelligence*, 1994, vol. 70.

[45] T.-C. Lu and M. J. Druzdzel, "Interactive construction of graphical decision models based on causal mechanisms," *European Journal of Operational Research*, vol. 199, no. 3, pp. 873–882, 2009.

[46] C. Ferent, "Systematic modeling and characterization of analog circuits using symbolic and data mining techniques," Ph.D. dissertation, STATE UNIVERSITY OF NEW YORK AT STONY BROOK, 2013.

[47] C. Ferent and A. Doboli, "Analog circuit design space description based on ordered clustering of feature uniqueness and similarity," *Integration, the VLSI Journal*, vol. 47, no. 2, pp. 213–231, 2014.

[48] H. Li, F. Jiao, and A. Doboli, "Analog circuit topological feature extraction with unsupervised learning of new sub-structures," in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2016, pp. 1509–1512.

[49] B. Razavi, *Design of analog CMOS integrated circuits*. Boston, MA: McGraw-Hill, 2000.

[50] S. Koziel and S. Szczepanski, "Design of highly linear tunable cmos ota for continuous-time filters," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 49, no. 2, pp. 110–122, 2002.

[51] J. Adut, J. Silva-Martinez, and M. Rocha-Perez, "A 10.7-mhz sixth-order sc ladder filter in 0.35-/spl mu/m cmos technology," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 53, no. 8, pp. 1625–1635, 2006.

[52] T. Das and P. Mukund, "Sensitivity analysis for fault-analysis and tolerance in rf front-end circuitry," in *Proc. of DATE*. EDA Consortium, 2007, pp. 1277–1282.

[53] T.-Y. Lo and C.-C. Hung, "A 40-mhz double differential-pair cmos ota with im3," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 55, no. 1, pp. 258–265, 2008.

[54] R. S. Assaad and J. Silva-Martinez, "The recycling folded cascode: a general enhancement of the folded cascode amplifier," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 9, pp. 2535–2542, 2009.

[55] C. Ferent and A. Doboli, "Formal representation of the design feature variety in analog circuits," in *FDL Conference*, 2013.

[56] ——, "Symbolic matching and constraint generation for systematic comparison of analog circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 4, pp. 616–629, 2013.

[57] J. Ramirez-Angulo and F. Ledesma, "The universal opamp and applications in continuous-time resistorless and capacitorless linear weighted voltage addition," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 53, no. 5, pp. 404–408, 2006.

[58] U.-K. Moon, "Cmos high-frequency switched-capacitor filters for telecommunication applications," *Solid-State Circuits, IEEE Journal of*, vol. 35, no. 2, pp. 212–220, 2000.

[59] A. M. Ismail and A. M. Soliman, "Novel cmos current feedback opamp realization suitable for high frequency applications," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 47, no. 6, pp. 918–922, 2000.

[60] P. Cusinato, F. Stefani, and A. Baschirotto, "Reducing the power consumption in high-speed delta-sigma bandpass modulators," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 48, no. 10, pp. 952–960, 2001.

[61] J. Silva-Martinez, J. Adut, J. M. Rocha-Perez, M. Robinson, and S. Rokhsaz, "A 60-mw 200-mhz continuous-time seventh-order linear phase filter with on-chip automatic tuning system," *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 2, pp. 216–225, 2003.

[62] B. K. Thandri and J. Silva-Martínez, "A robust feedforward compensation scheme for multistage operational transconductance amplifiers with no miller capacitors," *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 2, pp. 237–243, 2003.

[63] H. Lee and P. K. Mok, "Active-feedback frequency-compensation technique for low-power multistage amplifiers," *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 3, pp. 511–520, 2003.

[64] P. Kallam, E. Sánchez-Sinencio, and A. I. Karsilayan, "An enhanced adaptive q-tuning scheme for a 100-mhz fully symmetric ota-based bandpass filter," *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 4, pp. 585–593, 2003.

[65] A. N. Mohieldin, E. Sánchez-Sinencio, and J. Silva-Martínez, "A fully balanced pseudo-differential ota with common-mode feedforward and inherent common-mode feedback detector," *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 4, pp. 663–668, 2003.

[66] H. Lee, K. N. Leung, and P. K. Mok, "A dual-path bandwidth extension amplifier topology with dual-loop parallel compensation," *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 10, pp. 1739–1744, 2003.

[67] A. Lewinski and J. Silva-Martinez, "Ota linearity enhancement technique for high frequency applications with im3 below-65 db," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 51, no. 10, pp. 542–548, 2004.

[68] T. Ndjountche, F.-L. Luo, and R. Unbehauen, "A high-frequency double-sampling second-order delta-sigma modulator," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 52, no. 12, pp. 841–845, 2005.

[69] M. M. Ahmadi, "A new modeling and optimization of gain-boosted cascode amplifier for high-speed and low-voltage applications," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 53, no. 3, pp. 169–173, 2006.

[70] J. Chen, E. Sanchez-Sinencio, and J. Silva-Martinez, "Frequency-dependent harmonic-distortion analysis of a linearized cross-coupled cmos ota and its application to ota-c filters," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 53, no. 3, pp. 499–510, 2006.

[71] P. Pandey, J. Silva-Martinez, and X. Liu, "A cmos 140-mw fourth-order continuous-time low-pass filter stabilized with a class ab common-mode feedback operating at 550 mhz," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 53, no. 4, pp. 811–820, 2006.

[72] J. Ramirez-Angulo, R. G. Carvajal, J. A. Galán, and A. López-Martín, "A free but efficient low-voltage class-ab two-stage operational amplifier," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 53, no. 7, pp. 568–571, 2006.

[73] W. Huang and E. Sanchez-Sinencio, "Robust highly linear high-frequency cmos ota with im3 below-70 db at 26 mhz," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 53, no. 7, pp. 1433–1447, 2006.

[74] A. Lewinski and J. Silva-Martinez, "A high-frequency transconductor using a robust nonlinearity cancellation," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 53, no. 9, pp. 896–900, 2006.

[75] A. Vasilopoulos, G. Vitzilaios, G. Theodoratos, and Y. Papananos, "A low-power wideband reconfigurable integrated active-rc filter with 73 db sfdr," *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 9, pp. 1997–2008, 2006.

[76] R. Samadi and A. I. Karsilayan, "Uniform design of multi-peak bandwidth enhancement technique for multistage amplifiers," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 54, no. 7, pp. 1489–1499, 2007.

[77] T.-Y. Lo and C.-C. Hung, "A 1-v 50-mhz pseudodifferential ota with compensation of the mobility reduction," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 54, no. 12, pp. 1047–1051, 2007.

[78] Y. Zheng and C. E. Saavedra, "Feedforward-regulated cascode ota for gigahertz applications," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 55, no. 11, pp. 3373–3382, 2008.

[79] A. Kranti and G. A. Armstrong, "Nonclassical channel design in mosfets for improving ota gain-bandwidth trade-off," *IEEE Transactions on Circuits and Systems Part I: Regular Papers*, vol. 57, no. 12, pp. 3048–3054, 2010.

[80] L. Ye, C. Shi, H. Liao, R. Huang, and Y. Wang, "Highly power-efficient active-rc filters with wide bandwidth-range using low-gain push-pull opamps," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 60, no. 1, pp. 95–107, 2013.

[81] F. Jiao, S. Montano, and A. Doboli, "Knowledge-intensive, causal reasoning for analog circuit topology synthesis in emergent and innovative applications," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015*. IEEE, 2015, pp. 1144–1149.

[82] F. Jiao and A. Doboli, "A low-voltage, low-power amplifier created by reasoning-based, systematic topology synthesis," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2015, pp. 2648–2651.

[83] R. W. Gibbs Jr, *The Cambridge handbook of metaphor and thought*. Cambridge University Press, 2008.

[84] D. N. Osherson, D. Scarborough, and S. Sternberg, *An invitation to cognitive science*. Mit Press, 1998, vol. 4.

[85] R. Harjani, R. A. Rutenbar, and L. R. Carley, "Oasys: A framework for analog circuit synthesis," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 8, no. 12, pp. 1247–1266, 1989.

[86] F. El-Turky and E. E. Perry, "Blades: An artificial intelligence approach to analog circuit design," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 8, no. 6, pp. 680–692, 1989.

[87] G. Van der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. G. Gielen, W. Sansen, P. Veselinovic, and D. Leenarts, "Amgie-a synthesis environment for cmos analog integrated circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 20, no. 9, pp. 1037–1058, 2001.

[88] S. Maji and P. Mandal, "A geometric programming aided knowledge based approach for analog circuit synthesis and sizing," in *Proceedings of the 21st edition of the great lakes symposium on Great lakes symposium on VLSI*. ACM, 2011, pp. 411–414.

[89] W. Kruiskamp and D. Leenaerts, "Darwin: Cmos opamp synthesis by means of a genetic algorithm," in *Proceedings of the 32nd annual ACM/IEEE design automation conference*. ACM, 1995, pp. 433–438.

[90] J. D. Lohn and S. P. Colombano, "A circuit representation technique for automated circuit design," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 3, pp. 205–219, 1999.

[91] J. R. Koza, F. H. Bennett III, D. Andre, and M. A. Keane, "Automated wywiwyg design of both the topology and component values of electrical circuits using genetic programming," in *Proceedings of the 1st annual conference on genetic programming*. MIT Press, 1996, pp. 123–131.

[92] T. Sripramong and C. Toumazou, "The invention of cmos amplifiers using genetic programming and current-flow analysis," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 21, no. 11, pp. 1237–1252, 2002.

[93] A. Doboli, N. Dhanwada, A. Nunez-Aldana, and R. Vemuri, "A two-layer library-based approach to synthesis of analog systems from vhdl-ams specifications," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 9, no. 2, pp. 238–271, 2004.

[94] J. P. Harvey, M. I. Elmasry, and B. Leung, "Staic: An interactive framework for synthesizing cmos and bicmos analog circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 11, no. 11, pp. 1402–1417, 1992.

[95] Y. Wei, A. Doboli, and H. Tang, "Systematic methodology for designing reconfigurable σ modulator topologies for multimode communication systems," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 3, pp. 480–496, 2007.

[96] J. G. Kenney and L. R. Carley, "Clans: A high-level synthesis tool for high resolution data converters," in *Computer-Aided Design, 1988. ICCAD-88. Digest of Technical Papers., IEEE International Conference on*. IEEE, 1988, pp. 496–499.

[97] P. Thagard and T. C. Stewart, "The aha! experience: Creativity through emergent binding in neural networks," *Cognitive science*, vol. 35, no. 1, pp. 1–33, 2011.

[98] J. Luo and K. Niki, "Function of hippocampus in insight of problem solving," *Hippocampus*, vol. 13, no. 3, pp. 316–323, 2003.

[99] S. Vosniadou and A. Ortony, *Similarity and analogical reasoning*. Cambridge University Press, 1989.

[100] F. J. Costello and M. T. Keane, "Efficient creativity: Constraint-guided conceptual combination," *Cognitive Science*, vol. 24, no. 2, pp. 299–349, 2000.

[101] D. Kahneman and A. Tversky, "Choices, values, and frames." *American psychologist*, vol. 39, no. 4, p. 341, 1984.

[102] T. McConaghy, P. Palmers, M. Steyaert, and G. G. Gielen, "Trustworthy genetic programming-based synthesis of analog circuit topologies using hierarchical domain-specific building blocks," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 4, pp. 557–570, 2011.

[103] R. A. Rutenbar, G. G. Gielen, and B. A. Antao, "Anaconda: Simulationbased synthesis of analog circuits via stochastic pattern search," 2009.

[104] H. Tang, H. Zhang, and A. Doboli, "Refinement-based synthesis of continuous-time analog filters through successive domain pruning, plateau search, and adaptive sampling," *Computer-Aided Design of*

Integrated Circuits and Systems, IEEE Transactions on, vol. 25, no. 8, pp. 1421–1440, 2006.

[105] M. Eick and H. E. Graeb, "Mars: Matching-driven analog sizing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 8, pp. 1145–1158, 2012.

[106] D. Binkley, C. Hopper, S. D. Tucker, B. C. Moss, J. M. Rochelle, and D. Foty, "A cad methodology for optimizing transistor current and sizing in analog cmos design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 225–237, 2003.

[107] P. Mandal and V. Visvanathan, "Cmos op-amp sizing using a geometric programming formulation," *IEEE Transactions on Computer-Aided Design of Integrated circuits and systems*, vol. 20, no. 1, pp. 22–38, 2001.

[108] W. Daems, G. Gielen, and W. Sansen, "Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 5, pp. 517–534, 2003.

[109] S. P. Boyd, T. H. Lee *et al.*, "Optimal design of a cmos op-amp via geometric programming," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 1, pp. 1–21, 2001.

[110] Y.-L. Chen, W.-R. Wu, C.-N. J. Liu, and J. C.-M. Li, "Simultaneous optimization of analog circuits with reliability and variability for applications on flexible electronics," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 1, pp. 24–35, 2014.

[111] J. Pearl, "Causality: models, reasoning and inference," *Economet. Theor*, vol. 19, pp. 675–685, 2003.

[112] C. Ferent and A. Doboli, "Measuring the uniqueness and variety of analog circuit design features," *INTEGRATION, the VLSI journal*, vol. 44, no. 1, pp. 39–50, 2011.

[113] O. Z. Maimon and R. Horowitz, "Sufficient conditions for inventive solutions," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 29, no. 3, pp. 349–361, 1999.

[114] F. Jiao and A. Doboli, "A causal reasoning-based approach for analog circuit verification," in *Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), 2015 International Conference on*. IEEE, 2015, pp. 1–4.

[115] ——, "Causal reasoning mining approach to analog circuit verification," *Integration, the VLSI Journal*, vol. 55, pp. 376–383, 2016.

[116] E. Naviasky *et al.*, "Mixed-signal design challenges and requirements," *Cadence white paper*, 2009.

[117] K. Kundert and H. Chang, "Verification of complex analog integrated circuits," in *Proc. CICC*. IEEE, 2006, pp. 177–184.

[118] E. Barke, D. Grabowski, H. Graeb, L. Hedrich, S. Heinen, R. Popp, S. , and Y. Wang, "Formal approaches to analog circuit verification," in *Proc. DATE*. EDAA, 2009, pp. 724–729.

[119] Y. Wei and A. Doboli, "Systematic development of analog circuit structural macromodels through behavioral model decoupling," in *Proceedings of the 42nd annual Design Automation Conference*. ACM, 2005, pp. 57–62.

[120] ——, "Systematic development of nonlinear analog circuit macromodels through successive operator composition and nonlinear model decoupling," in *Proceedings of the 43rd annual Design Automation Conference*. ACM, 2006, pp. 1023–1028.

[121] F. De Bernardinis, M. I. Jordan, and A. SangiovanniVincentelli, "Support vector machines for analog circuit performance representation," in *Design Automation Conference, 2003. Proceedings*. IEEE, 2003, pp. 964–969.

[122] Y. Wang, S. Joeres, R. Wunderlich, and S. Heinen, "Modeling approaches for functional verification of rf-socs: limits and future requirements," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 5, pp. 769–773, 2009.

[123] K. Kundert, "Challenges in rf simulation," in *Radio Frequency integrated Circuits (RFIC) Symposium, 2005. Digest of Papers. 2005 IEEE*. IEEE, 2005, pp. 105–108.

[124] R. Frevert, J. Haase, R. Jancke, U. Knochel, P. Schwarz, R. Kakerow, and M. Darianian, *Modeling and Simulation for RF System Design*. Springer Science & Business Media, 2006.

[125] "Virtuoso spectre circuit simulator rf analysis user guide," 2007.

[126] "Cadence ams simulator user guide," 2001.

[127] S. Joeres and S. Heinen, "Functional verification of radio frequency socs using mixed-mode and mixed-domain simulations," in *Behavioral Modeling and Simulation Workshop, Proceedings of the 2006 IEEE International*. IEEE, 2006, pp. 144–149.

[128] S. Joeres, H.-W. Groh, and S. Heinen, "Event driven analog modeling of rf frontends," in *Behavioral Modeling and Simulation Workshop, 2007. BMAS 2007. IEEE International*. IEEE, 2007, pp. 46–51.

[129] S. K. Tiwary, A. Gupta, J. R. Phillips, C. Pinello, and R. Zlatanovici, "First steps towards sat-based formal analog verification," in *Computer-Aided Design-Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on.* IEEE, 2009, pp. 1–8.

[130] M. Miller and F. Brewer, "Formal verification of analog circuit parameters across variation utilizing sat," in *Proc. of DATE.* EDA Consortium, 2013, pp. 1442–1447.

[131] T. R. Dastidar and P. Chakrabarti, "A verification system for transient response of analog circuits," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 12, no. 3, p. 31, 2007.

[132] D. C. Walter, "Verification of analog and mixed-signal circuits using symbolic methods," Ph.D. dissertation, Citeseer, 2007.

[133] Y. Song, H. Yu, and S. M. P. DinakarRao, "Reachability-based robustness verification and optimization of sram dynamic stability under process variations," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 33, no. 4, pp. 585–598, 2014.

[134] S. Steinhorst and L. Hedrich, "Equivalence checking of nonlinear analog circuits for hierarchical ams system verification," in *Proc. VLSI-SoC.* IEEE, 2012, pp. 135–140.

[135] M. H. Zaki, S. Tahar, and G. Bois, "Formal verification of analog and mixed signal designs: A survey," *Microelectronics Journal*, vol. 39, no. 12, pp. 1395–1404, 2008.

[136] S. N. Ahmadyan, J. A. Kumar, and S. Vasudevan, "Runtime verification of nonlinear analog circuits using incremental time-augmented rrt algorithm," in *Proc. of DATE.* EDA Consortium, 2013, pp. 21–26.

[137] M. Aminian and F. Aminian, "Neural-network based analog-circuit fault diagnosis using wavelet transform as preprocessor," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 2, pp. 151–156, 2000.

[138] P. Dague, O. Raiman, P. Deves, and J. Marx, "Dedale: An expert system for troubleshooting analogue circuits," in *IEEE International Test Conference*, 1987, pp. 586–594.

[139] C.-L. Hsu, Y. Lai, and S.-W. Wang, "Built-in self-test for phase-locked loops," *IEEE Transactions on instrumentation and measurement*, vol. 54, no. 3, pp. 996–1002, 2005.

[140] S. Yu, B. W. Jervis, K. R. Eckersall, and I. M. Bell, "Diagnosis of cmos op-amps with gate oxide short faults using multilayer perceptrons," *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 16, no. 8, pp. 930–935, 1997.

[141] Z. R. Yang, M. Zwolinski, C. D. Chalk, and A. C. Williams, "Applying a robust heteroscedastic probabilistic neural network to analog fault detection and classification," *IEEE Transactions on computer-aided design of integrated circuits and systems*, vol. 19, no. 1, pp. 142–151, 2000.

[142] P. Wang and S. Yang, "A new diagnosis approach for handling tolerance in analog and mixed-signal circuits by using fuzzy math," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 10, pp. 2118–2127, 2005.

[143] "Cadence verilog-ams language reference," p. 293, 2005.