

2012-12-06

A Computer-Aided Design Assistant Tool for Elementary Linear Circuit Topologies

Shahhosseini, Delaram

Shahhosseini, D. (2012). A Computer-Aided Design Assistant Tool for Elementary Linear Circuit Topologies (Master's thesis, University of Calgary, Calgary, Canada). Retrieved from <https://prism.ucalgary.ca>. doi:10.11575/PRISM/24678

<http://hdl.handle.net/11023/338>

Downloaded from PRISM Repository, University of Calgary

UNIVERSITY OF CALGARY

A Computer-Aided Design Assistant Tool for Elementary Linear Circuit Topologies

by

Delaram Shahhosseini

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA

November, 2012

© Delaram Shahhosseini 2012

UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled “A Computer-Aided Design Assistant Tool for Elementary Linear Circuit Topologies” submitted by Delaram Shahhosseini in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE.

Supervisor, Dr. Leonid Belostotski
Department of Electrical and Computer Engineering

Co-Supervisor, Dr. Laleh Behjat
Department of Electrical and Computer Engineering

Dr. Sebastian Magierowski
Department of Electrical and Computer Engineering

Dr. Svetlana Yanushkevich
Department of Electrical and Computer Engineering

Dr. Xin Wang
Department of Geomatics Engineering

Date

Abstract

In this thesis, a CAD tool called analog design assistant (ADA), is developed to help analog circuit designers find new circuit topologies. First, a methodology to automatically generate all analog circuit topologies containing two or three transistors is developed. For each topology, circuit characteristics, such as DC voltage gain, are calculated. The DC voltage gain of each generated circuit is maximized by formulating and solving an optimization problem. After solving the optimization problem, it is shown that over 5,000 out of 56,000 circuits can achieve a DC voltage gain higher than 1. All generated circuit topologies and corresponding characteristics are stored in a database. A GUI is developed to help analog circuit designers search the database and find new topologies. In order to demonstrate the capability of ADA in generating new topologies, a previously unknown high-gain amplifier is selected, and designed in a $0.13\text{-}\mu\text{m}$ standard CMOS technology.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Tables	v
List of Figures	vi
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Contributions	3
1.3 Thesis Structure	4
2 Analog Circuit Representation and Synthesis	6
2.1 Introduction	6
2.2 Circuit Representations	6
2.2.1 Schematic Representation	7
2.2.2 Graph Representation	8
2.2.3 Netlist Representation	9
2.2.4 Matrix Representation	10
2.3 CAD-based Analog Circuit Synthesis and Analysis	14
2.3.1 Previously known circuit topologies	14
2.3.2 Generating new circuit topologies	17
2.4 Optimization Applications for Circuit Synthesis	23
2.5 Summary	26
3 Proposed Systematic Hierarchical Methodology	27
3.1 Introduction	27
3.2 Circuit Representation	28
3.2.1 Connectivity Matrix Construction	29
3.3 Circuits Generation Using a Connectivity Matrix	30
3.3.1 2-transistor Circuits Generation	30
3.3.2 Multiple Transistors Circuits Generation	36
3.4 Theoretical Proof of Circuit Generation Methodology	38
3.4.1 2-transistor Circuits	38
3.4.2 3-transistor Circuits	51
3.5 Circuit Generation Results	58
3.6 Circuit Analysis	60
3.6.1 Circuit Optimization	61
3.7 Analog Design Assistant Tool	65
3.8 Summary	70
4 Analysis and Design of Selected Amplifier	71
4.1 Introduction	71
4.2 Optimization of Small-Signal Transistors' Parameters	71
4.3 Circuit Design	73
4.3.1 Current Sources	74
4.3.2 DC controller	77

4.4	Circuit Analysis	78
4.4.1	Voltage Gain And Input Power Match	78
4.4.2	Noise Analysis	79
4.5	Simulation Result	84
4.6	Summary	85
5	Conclusions and Further Directions	87
5.1	Conclusions	87
5.2	Contributions	88
5.3	Further Directions	88
	Bibliography	90

List of Tables

3.1	The maximum number of configurations of connecting the I/O and $C2T$ blocks for <i>configuration a, b</i>	50
3.2	The maximum number of configurations of connecting I/O and $C2T$ blocks for <i>configuration c</i>	51
3.3	The number of generated 2-transistor circuits	59
3.4	The number of generated 3-transistor circuits	59
3.5	The initial points for optimization problem	62
3.6	The number of optimized 2-transistor amplifiers	62
3.7	The number of optimized 3-transistor amplifiers	62
3.8	The small-signal transistor parameters and the circuit specifications of previously unknown amplifiers in Fig. 3.17	65
3.9	An example of required small-signal specifications.	68
4.1	The specifications of selected amplifier	73
4.2	The comparison of ADA and Cadence results	84
4.3	The value of g_m and g_{ds} for M_1, M_2 , and M_3	85
4.4	The size of transistors in final schematic of Amp2	85

List of Figures and Illustrations

2.1	The schematic representation of a circuit	7
2.2	The circuit graph of circuit illustrated in Fig.2.1	8
2.3	The circuit netlist of circuit illustrated in Fig.2.1	9
2.4	The adjacency matrix representation for circuit illustrated in Fig.2.1	10
2.5	The incidence matrix for the circuit illustrated in Fig.2.1	11
2.6	A radio receiver decomposed into three blocks	15
2.7	The procedure of analog circuits synthesis described in [6].	16
2.8	A two-port network and its ABCD matrix	18
2.9	A transistor is modelled as a VCCS.	19
2.10	Representation of two-port network with a MOS transistor by a graph.	19
2.11	The flowchart of circuit generations algorithm	20
2.12	The 2-MOSFET wideband amplifiers generated in [26, 27] (biasing not shown)	21
2.13	The evolved topology is located between fixed input and output terminals [29].	22
2.14	The flowchart of simulated annealing algorithm	25
3.1	Building blocks of an analog circuit.	28
3.2	The connectivity matrix, CM , and the position of its sub-matrices.	29
3.3	An analog circuit (biasing not shown) and corresponding connectivity matrix	30
3.4	All possible I/O block configurations.	31
3.5	All possible T block configurations	33
3.6	The algorithm for generation of the IO/T sub-matrix	35
3.7	The CM for 3-transistor circuit and its sub-matrices	36
3.8	A sample circuit with three transistors (biasing not shown) and its CM . . .	37
3.9	An example of a $C2T$ block and T_3 block.	37
3.10	Two equivalent circuits with two different CM s, when two transistors are swapped.	42
3.11	All possible T_i/T_j sub-matrices.	43
3.12	The elements of the T_i/T_j sub-matrix.	44
3.13	Two equivalent circuits with two different CM s, when two sides of the load are swapped.	46
3.14	The histogram for amplifiers with gain less than and equal to 140.	63
3.15	The histogram for amplifiers with gain higher than 140.	63
3.16	The three generated 3 transistor amplifiers with the highest gain.	64
3.17	Examples of three previously unknown amplifiers with three transistors. . . .	66
3.18	Small-signal model of the three amplifiers in Fig. 3.17.	67
3.19	Specifications are entered in "Required Specifications" window.	68
3.20	The provided topologies with their CM and small-signal specifications. . . .	69
3.21	The small-signal diagram of a circuit generated by ADA.	69
4.1	The topology and its small signal model of selected amplifier, Amp 2.	72
4.2	Schematic of the amplifier with ideal current sources.	74

4.3	A low headroom cascode current source.	76
4.4	The final schematic of Amp 2.	78
4.5	Thermal noise of a resistor.	80
4.6	The noise generated by the drain of a transistor.	81
4.7	Voltage gain and NF versus g_{m_2} and g_{m_3} values for $\gamma=1.33$ and $R_{in} = R_s$. . .	83
4.8	The voltage gain and the NF of Amp 2.	86
4.9	The input impedance and the reverse gain of the amplifier.	86

List of Terms

Acronyms:

ADA	:	Analog Design Assistant
ADC	:	Analog-to-Digital Converter
BSIM	:	Berkeley Short-channel IGFET Model
CAD	:	Computer-Aided Design
CMOS	:	Complementary Metal-Oxide-Semiconductor
DAC	:	Digital-to-Analog Converter
$C2T$:	Connected-2-transistor
$C3T$:	Connected-3-transistor
DC	:	Direct Current
DCC	:	DC Controller
DCT	:	Diode-Connected Transistor
DSP	:	Digital Signal Processing
EA	:	Evolutionary Algorithm
GBW	:	Gain-Bandwidth
GUI	:	Graphical User Interface
I/O	:	Input/Output
IC	:	Integrated Circuit
KCL	:	Kirchhoff's Current Law
KVL	:	Kirchhoff's Voltage Law
LNA	:	Low Noise Amplifier
LP	:	Low Pass
MNA	:	Modified Nodal Analysis
MOS	:	Metal-Oxide-Semiconductor

MOSFET	:	Metal-Oxide-Semiconductor Field-Effect Transistor
NDCT	:	Non-Diode-Connected Transistor
SA	:	Simulated Annealing
SoC	:	System on Chip
SPICE	:	Simulation Program with Integrated Circuit Emphasis
T_i	:	Transistor i
VCCS	:	Voltage Controlled Current Source

Variables:

μ	:	Charge-carrier effective mobility
λ	:	Channel-length modulation
γ	:	Gamma factor
A_f	:	Forward gain
A_{rev}	:	Reverse gain
ABCD	:	The two-port transmission parameters
C_{ox}	:	Gate oxide capacitance per unit area
f	:	Frequency
F	:	Noise factor
F_{M_i}	:	Noise factor of transistor i , M_i
GND	:	Ground
g_{ds}	:	Output transconductance of a MOS transistor
g_m	:	Transconductance of a MOS transistor
g_{m_u}	:	Upper bound of MOS Transconductance
g_{m_l}	:	lower bound of MOS Transconductance
K	:	Process-dependent constant
k	:	The number of non-attacking rook

L	:	Gate length of a transistor
N	:	Total number of components in a circuit
n_N	:	Total number of nodes in a circuit
n_s	:	Total number of voltage source
n_{NDC2k}	:	The number of unique 2-transistor circuit when all transistors are NDCT
NF	:	Noise figure
P_{av}	:	Average noise power of signal $x(t)$ in time domain
P_i	:	i_{th} port of $C2T$
R_L	:	Load resistor
R_{L_u}	:	Upper bound of the load resistor
R_{L_l}	:	Lower bound of the load resistor
V_{in}	:	Input Voltage
R_{in}	:	Input resistance
R_{in_u}	:	Upper bound of the input resistance
R_{in_l}	:	lower bound of the input resistance
R_o	:	Output resistance
R_{o_u}	:	Upper bound of output resistance
R_{o_l}	:	lower bound of output resistance
r_{ds}	:	Output resistance of a MOS transistor
R_{on_i}	:	Output resistance at node i , n_i
R_{cs}	:	Output resistance of low-voltage cascode current source
$r_{k_{m,n}}$:	The number of ways to place k non-attacking rook on an $m \times n$ board
$S_x(f)$:	Average noise power of signal $x(t)$ in frequency domain
SNR	:	Signal-to-noise ratio
SNR_{input}	:	Signal-to-noise ratio at input

SNR_{output}	:	Signal-to-noise ratio at output
$S_{(m,n)_k}$:	The number of symmetric $m \times n$ board with k rooks
T	:	Period of signal $x(t)$
V_{TH}	:	Threshold voltage of a MOS transistor
W	:	Gate width of a transistor
Z_{in}	:	Input impedance
Z_o	:	Output impedance

Scalars:

a_{ij}	:	Element of adjacency matrix in row i and column j
b_{ij}	:	Element of incidence matrix in row i and column j
CM_{ij}	:	Element of connectivity matrix in row i and column j

Vectors:

I_c	:	Components' Current Vector
-------	---	----------------------------

Matrices:

A	:	Adjacency matrix
B	:	Incidence matrix
M	:	MNA matrix
CM	:	Connectivity matrix
T_i/T_i	:	Connectivity sub-matrix of connection between T_i block ports
T_i/T_j	:	Connectivity sub-matrix of connection between T_i block ports and T_j block ports

IO/T : Connectivity sub-matrix of connection between IO block ports and T block ports

IO/IO : Connectivity sub-matrix of connection between IO block ports

Functions and Sets:

E : Set of edges

G : A graph

$L_n^\alpha(.)$: Laguerre polynomial of degree n

$R_{m,n}(.)$: Rooks polynomial for an $m \times n$ board

V : Set of vertices

Chapter 1

Introduction

1.1 Motivation

With the advancement of technology, the number of transistors on an integrated circuit (IC) has increased substantially and this number doubles every two years as predicted by Moore's Law, which was introduced in 1965 by Gordon E. Moore [1]. As a result, complex systems that previously were implemented on several boards can be integrated on a single chip, which is called system on a chip (SoC). This leads to a significant reduction in the fabrication costs and an increase in the overall performance [2].

Most of the functions in an SoC are implemented using digital circuits and in particular digital signal processing (DSP) circuits. In comparison with analog circuits, DSP circuits are immune to noise and flexible to technology scaling [3]. There are still several analog circuits that cannot be replaced by digital circuits because the real world is analog [4]. These functions include the input/output interfaces, analog-to-digital converters (ADCs), digital-to-analog converters (DACs). Furthermore, some of the high-speed digital circuits are custom designed like analog circuits in order to achieve high performance [4].

Computer-aided design (CAD) tools have been employed by digital circuit designers for several decades to enable them to manage the increased complexity of the today's complex SoCs [5, 6]. Use of CAD tools has resulted in a shorter time-to-market for digital circuits. However, CAD tools that can help analog circuit designers are not as commonly used. This is mainly because of the complexity of the analog circuits, which require many years of design experience for successful implementation. In particular, there are different parameters that should be considered in an analog circuit design such as noise, power consumption, gain,

bandwidth, input and output impedance, area, and fabrication issues.

Although, a fully automated analog CAD tool does not seem to be feasible yet, there is a strong motivation to develop tools to assist analog designers automate certain parts of the design procedure. Analog circuit synthesis is one part that has recently become a topic of interest in both academia and industry. Synthesizing an analog circuit is usually performed in two steps: 1) selecting the appropriate topology. 2) Sizing the components in the selected topology to get the desired performance.

Analog circuit designers usually resort to a handful of well-known elementary circuits to design bigger and more complicated circuits. This approach is based on the experience and knowledge of the analog designer. In order to automate this step, there are two different approaches in the literature. In the first approach, in order to find a topology, all the configurations of complementary metal-oxide-semiconductor (CMOS) transistors, a voltage supply, and a resistor are generated, and the best one is selected [7]. The second approach is to employ well-known elementary circuits, which are commonly used in different circuits [4]. For instance, differential pairs, cascode amplifier, single stage amplifiers such as a common-source amplifier, current mirror circuits such as Wilson current mirror are used to generate new architectures [8].

When a topology is selected, the transistors should be sized for best performance that meets the required specifications. In the past, this step was done by either hand analysis and computer simulations. In the recent years, there have been many efforts to automate this step [9], but they are not well adopted by analog designers as they do not give the analog designers enough degrees of freedom to design their circuits. In other words, analog designers prefer to employ a design assistant tool rather than a fully automated CAD tool, which usually fails to meet all the required specifications.

1.2 Thesis Contributions

In the following, a brief overview of the key contributions of this work is given.

- **Analog CAD Tool Development**

In this thesis, a CAD tool to help analog designers, analog design assistant (ADA), is developed. This tool helps analog designers synthesize and design analog circuits. In order to develop this tool, a methodology is proposed to automatically generate and classify all elementary circuit topologies that can be used in building analog circuits. Rather than storing circuit schematic representations, a sparse binary matrix representation is developed to represent the analog circuits. The proposed matrix representation allows efficient generation and storage of a large number of circuits. The proposed circuit generation algorithm results in over 56,000 circuits with three transistors. In addition, to obtain the maximum gain of each circuit, all generated circuit topologies are optimized. As an example of the potential usage of the developed tool, 5,103 unique circuits are identified as amplifiers with DC voltage gains higher than one.

Once the circuit topologies are generated, and optimized, the optimized circuits with corresponding circuit component values are stored in a database, which is easily accessible through a developed graphical user interface (GUI), where users can select their desired topologies by specifying the required range of circuit specifications. We anticipate that by using ADA many hours of design time can be saved and analog designers will have a bigger palette of designs to choose from.

- **Circuit Development**

In order to verify the capability of the ADA, one of the small-signal amplifiers

generated by ADA is selected and designed in a 0.13- μm CMOS technology. To the best knowledge of the author, this amplifier has been never published in the literature and can be considered as a novel architecture. The simulation results show that the proposed amplifier has a noise figure of less than 3.1 dB and a voltage gain of 39 dB with a gain-bandwidth product of 50 GHz while it consumes 6.8 mW from a 1.8-V supply.

1.3 Thesis Structure

The rest of this thesis is organized into the following chapters:

In Chapter 2, different circuit representations are introduced. A brief background of the design and synthesis of analog circuits is presented. Finally, different optimization methods are introduced in this chapter.

In Chapter 3 a new methodology is proposed to generate all functional and unique small-signal circuit topologies. Then, the specifications of generated circuits are calculated and analyzed to optimize their DC voltage gains. Generated circuit topologies with corresponding circuit component specifications are stored in a database. The database is provided with a user friendly graphical user interface (GUI) to help analog designers search through circuits based on their desired specifications. The GUI is called Analog Designer Assistant (ADA), since it is envisioned that it will assist analog designers to find new circuit topologies.

In Chapter 4 the design procedure of one of many 3-transistor amplifiers identified by ADA is presented. This circuit is optimized by ADA to have a high voltage gain and input power match. The selected configuration is then analyzed and designed in a 0.13- μm standard CMOS technology. Finally, the agreement between Cadence circuit simulation results and the ADA optimization results is demonstrated in this chapter to show the ability of the proposed ADA to identify and accurately optimized circuits.

In Chapter 5 the summary of major contributions of this thesis is presented. In addition, some suggestions for the future work are briefly discussed.

Chapter 2

Analog Circuit Representation and Synthesis

2.1 Introduction

Analog designers usually employ a few known topologies to design circuits such as operational amplifiers, current sources, mixers, low noise amplifiers (LNAs), filters, and frequency synthesizers. In the area of LNA design, for example, common-source and common-gate topologies are well known architectures that have been employed for several decades and are still widely used by the analog designers [10]. Discovering novel architectures with better performance has been one of the challenges in the analog design process. In addition, analog designers often use an often cumbersome combination of hand analysis and circuit simulations to evaluate the performance of their new circuits. Consequently, in order to save time and cost, analog computer-aided design (CAD) tools are needed to reduce the manual design process and offer new topologies. The main purpose of this thesis is to develop a new CAD tool that can be used as a design assistant and can generate and optimize novel circuit topologies.

The rest of this chapter is organized as follows: In Section 2.2 different representations for analog circuits are discussed. In Section 2.3 synthesis of analog circuits is reviewed. Circuit optimization approaches are discussed in Section 2.4. Finally, the chapter is summarized in Section 2.5.

2.2 Circuit Representations

One of the essential tasks in the design of analog circuits is to compute different characteristics of circuits and to evaluate circuit behaviour under different operation points. To obtain

these circuit characteristics appropriate formats need to be used. Some of these formats are schematics, graphs, netlists, and matrices, which will be discussed in the following section. Based on the criteria that have to be evaluated one of the formats is chosen. The different formats can be converted to each other, for example, most of the electronic design automation (EDA) tools use a netlist representation to simulate a circuit, but the analog designers prefers to draw their circuits in the schematic format. Therefore, the EDA tools convert the schematic representation to the netlist representation to simulate a circuit, and vice versa. These formats are briefly discussed in the following.

2.2.1 Schematic Representation

A schematic diagram is an abstract representation for analog circuits, which represents the circuit components as standard simplified symbols and shows the connection between them. In this representation, a node refers to a point where two or more components are connected to each other, and a branch indicates the connection between two nodes. An example of the schematic representation of a circuit is shown in Fig. 2.1. In this circuit nodes are numbered from 0 to 3 and the currents of branches are illustrated by I_1, I_2, I_3, I_4, I_5 . Commonly, the node connected to ground is considered node 0.

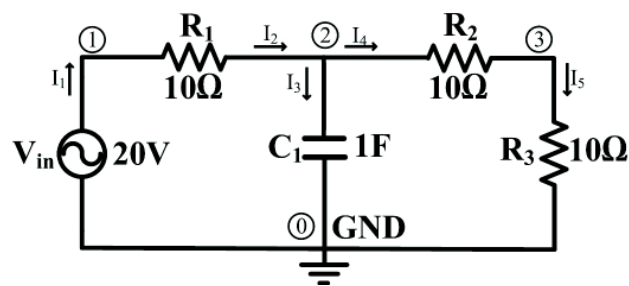


Figure 2.1: The schematic representation of a circuit

The schematic representation of a circuit is the most commonly used circuit representation for analog designers, since it is suitable for Kirchhoff's current law (KCL) and Kirchhoff's voltage law (KVL) in hand analyses.

Some of the EDA tools, such as Cadence [11] and Agilent ADS [12] accept schematic representations created by the user as input to their program. While a schematic diagram is a user friendly representation, it is not efficient to use when the size of memory to store circuit schematic is important. The memory size needed to store schematics is negligible when designers are dealing with limited number of circuit components, but it should be taken into account when a database of thousands of circuits is generated.

2.2.2 Graph Representation

Another method to represent a circuit is by using a graph representation [13, 14, 15, 16]. A graph, $G = (V, E)$, is defined as a set of vertices $V = \{v_1, v_2, \dots\}$ and a set of edges $E = \{e_1, e_2, \dots\}$, where, an edge connects exactly two vertices. Graphs are classified based on desired properties. For instance, if directions of edges are important, they are classified in two categories, directed and non-directed, where a directed graph is a graph in which a direction is shown for each edges. The direction of the edges shows the direction of current in the related component. Since, the current in each branch has direction, the directed graph representation is commonly used to analyze circuits.

In Fig.2.2, a graph representation of the circuit illustrated in Fig. 2.1 is given. In this example circuit components $R_1, R_2, R_3, C_1, V_{in}$ and GND are represented by vertices.

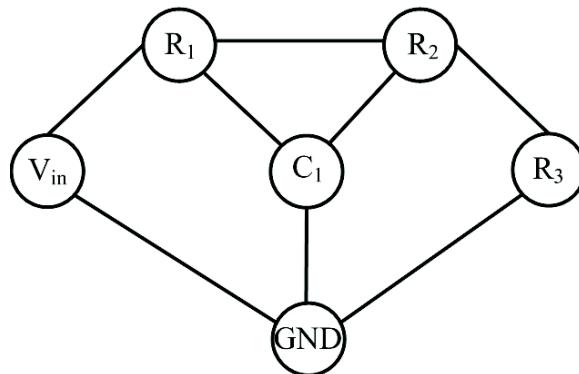


Figure 2.2: The circuit graph of circuit illustrated in Fig.2.1

Since the graph representation is an abstract representation and is simple to draw, it has been used to obtain proper values for parameters such as sizes of transistors or transistor biasing [13, 14, 15]. This representation is not commonly used for KCL or KVL analysis the circuit, since it is more abstract than the schematic representation.

2.2.3 Netlist Representation

In a netlist representation, the connection between circuit components are shown in a text format. Each component in the netlist representation is assigned a line. The first column of each line demonstrates the name of component in the circuit. Then, component terminal connections to circuit nodes are indicated in next columns. The last column is the value of the component.

The netlist representation of the example in Fig. 2.1 is shown in Fig. 2.3. For instance, resistor 1, R_1 , is connected to node 1 and 2, and has a value of $10\ \Omega$ as shown in line 2 of the netlist representation.

V_{in}	1	0	20
R_1	1	2	10
R_2	2	3	10
R_3	3	0	10
C_1	2	0	1

Figure 2.3: The circuit netlist of circuit illustrated in Fig.2.1

Simulator program with integrated circuit emphasis (SPICE) introduced in 1970 [17], and its derivatives PSPICE and HSPICE input circuits in the netlist representation and generate netlist files in the text format.

The netlist representation is efficient when a large number of circuits is considered, due to its low memory requirements. The disadvantage of this representation is that it is too abstract for analog designers.

2.2.4 Matrix Representation

A circuit can be represented by a matrix [18, 19]. Three types of matrix representation are discussed in this section: Adjacency matrix, Modified Nodal Analysis (MNA) matrix and Incidence matrix.

Adjacency Matrix

The adjacency matrix, A , shows the connections between the cells of a matrix. Normally, it is an $N \times N$ matrix, where N is the total number of components in the circuit. The adjacency matrix is constructed using only binary values, 0 and 1, where each column or row of the matrix represents a component in the circuit. If component i is connected to component j , the corresponding elements of A , a_{ij} and a_{ji} , are set to 1, otherwise the value is 0. Since a_{ij} and a_{ji} are equal, the adjacency matrix is symmetric. The diagonal elements of the matrix are normally set to zero. The adjacency matrix for the example in Fig. 2.1 is shown in Fig. 2.4.

$$\mathbf{A} = \begin{matrix} & \mathbf{V_{in}} & \mathbf{R_1} & \mathbf{R_2} & \mathbf{R_3} & \mathbf{C_1} & \mathbf{GND} \\ \mathbf{V_{in}} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{R_1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{R_2} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{R_3} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{C_1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{GND} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \end{matrix}$$

Figure 2.4: The adjacency matrix representation for circuit illustrated in Fig.2.1

In [20], the adjacency matrix representation is employed to represent and synthesize analog and digital circuits. This representation is efficient to store analog and digital circuits, since it requires low memory size. The adjacency representation is too abstract to analyze performance of circuits.

Incidence Matrix

The incident matrix, B , is an $n_N \times N$ matrix, where n_N is the number of nodes and N is the number of components. The elements of the matrix are -1 , 0 , and 1 . If the current of component i leaves node j , the corresponding element in the incident matrix (b_{ij}) is set to 1 and if the current of component i enters node j , the corresponding element in the incident matrix (b_{ij}) is set to -1 . Otherwise the value would be set to 0 . Hence, the incident matrix shows the direction of current through components. The incidence matrix for the circuit in Fig. 2.1 is shown in Fig. 2.5.

$$\mathbf{B} = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{matrix} \mathbf{V}_{in} \\ \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{R}_3 \\ \mathbf{C}_1 \\ \mathbf{GND} \end{matrix} & \begin{bmatrix} -1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

Figure 2.5: The incidence matrix for the circuit illustrated in Fig.2.1

Incident matrices are usually used to perform KCL to find branch currents and analyze the circuit performance. In order to use B matrix to perform KCL, the row and column corresponding to node 0, and ground are removed, since the voltage of node 0 is 0. Using the matrix B leads to obtain KCL equations as follows:

$$B \times I_c = 0, \quad (2.1)$$

where, I_c is a vector of components' currents. For example, the circuit in Fig. 2.1, can be

analyzed by solving the following equation:

$$\begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \times \begin{bmatrix} I_{c1} \\ I_{c2} \\ I_{c3} \\ I_{c4} \\ I_{c5} \end{bmatrix} = 0 \quad (2.2)$$

Modified Nodal Analysis (MNA) Matrix

Modified nodal analysis (MNA) matrix, M , shows the value of the components in the circuit between any two nodes. In this representation, a node refers to a point where two components are connected to each other, and a branch indicates the connection between two components. This resultant matrix is $(n_S + n_N - 1) \times (n_S + n_N - 1)$, where n_N is the number of nodes in the circuit and n_S is the number of voltage sources. The matrix is developed by using MNA method [21]. This method requires a three step of KCL. In the first step, the KCL is written for each node in the circuit, except the *GND* node, since the voltage of this node is 0. The voltage-current characteristic equation for each branch is written in the second step. At the end, the obtained equations from the second step are substituted in equations achieved from the first step. The final equations result in a matrix equation as shown in the following:

$$M \times x = z. \quad (2.3)$$

Matrix M , x , and z can also be obtained without writing KCL equations. In this approach, matrix M is developed as the combination of 4 sub-matrices, M_G , M_B , M_C , and M_D as follows:

$$M = \begin{bmatrix} M_G & M_B \\ M_C & M_D \end{bmatrix}. \quad (2.4)$$

M_G matrix is an $n_N - 1 \times n_N - 1$ matrix with its columns and rows representing the nodes. The diagonal elements of the matrix represent the sum of the components admittances connected to the corresponding node, and the off-diagonal elements indicate the negative of the component admittance of connected between two corresponding nodes. M_B matrix is an $n_S \times n_N - 1$ matrix, which shows the voltage source connected to each node. For instance if voltage source 2 is connected to node 3, element $M_{B_{23}}$ will be set to 1. M_C matrix is an $n_N - 1 \times n_S$ matrix, and it is a transpose of matrix M_B . M_D matrix is an $n_S \times n_S$ matrix, and if only independent sources are considered it is zero. When there is a dependent source in a circuit, the voltage of each node or the current of each branch is dependent on some other voltage or current in the circuit. Based on the equation of the dependent source, M matrix has to be modified.

x matrix is an $(n_N + n_s - 1) \times 1$ matrix. The first $n_N - 1$ rows in matrix x indicate voltages of the nodes in circuit, except the node connected to ground, and the last n_s rows in the matrix represent the currents flowing through each voltage source. For instance, for circuit in Fig. 2.1, there are three nodes, where their voltages are V_1, V_2 , and V_3 , and there is one voltage source, and its current is $I_{V_{in}}$.

z matrix is an $(n_N + n_s - 1) \times 1$ matrix. The first $n_N - 1$ rows in matrix x indicate the sum of currents flowing into nodes in circuit, except the node connected to ground (zero or the sum of independent current sources) and the last n_s rows in the matrix represent the voltages of the voltage sources. For instance, for circuit in Fig. 2.1, there are three nodes, where the currents flowing in to them are all 0, and there is one voltage source, V_{in} .

The MNA matrix representation for circuit in Fig. 2.1 is:

$$\left[\begin{array}{ccc|c} G_1 & -G_1 & 0 & 1 \\ -G_1 & G_1 + G_2 + sC_1 & -G_2 & 0 \\ 0 & -G_2 & G_2 + G_3 & 0 \\ \hline 1 & 0 & 0 & 0 \end{array} \right] \times \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ I_{V_{in}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ V_{in} \end{bmatrix}, \quad (2.5)$$

where G_i is the conductance of component i , and is equal to $\frac{1}{R_i}$.

Finally, by solving (2.5), frequency domain transfer functions of analog circuits with multi-input and multi-outputs are obtained and used to analyze circuit performance [22].

It should be mentioned that the matrix representation is used in this thesis, due to their low memory requirements, and the ease of applying KCL.

2.3 CAD-based Analog Circuit Synthesis and Analysis

Analog designers would normally use different tools to analyze and synthesize circuits. In analog circuit analysis, circuit topology and parameters are given, and the performance of a circuit is calculated and analyzed [23]. In circuit synthesis, desired specifications are given. Then blocks and sub-blocks are synthesized or selected, and the parameters for the sub-blocks, such as transistor size and biasing, are determined to obtain the desired specifications [24]. These sub-blocks are selected either by:

- Using previously used circuit topologies, or
- generating new circuit topologies

2.3.1 Previously known circuit topologies

In order to speed up circuit design and development, many of functional analog circuits can be catalogued into libraries. Therefore, analog designers can re-use known circuit topologies and do not need to recalculate their characteristics, which results in saving valuable computation and design time.

In [6], in order to select appropriate blocks, desired circuits are decomposed into sub-blocks. For instance, a radio receiver shown in Fig. 2.6, is decomposed into three sub-blocks: a tuner, a demodulator, and a voltage amplifier. To design this radio receiver, each initial sub-block is selected from the available topologies in the library. This selection is based on required specifications for the circuit, which are defined after investigating final product's cost and time.

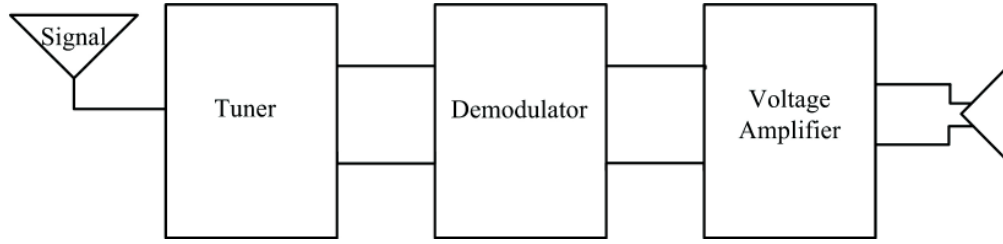


Figure 2.6: A radio receiver decomposed into three blocks in order to design in approach described in [6]

The proposed algorithm in [6] is shown in Fig. 2.7. First, the circuit specifications are defined which leads to the selection of suitable blocks from already known topologies stored in the library. Then, all the sub-blocks are assembled and the circuit specifications are calculated. If the required specifications are met, the algorithm proceeds to the next phase which is the layout generation. Otherwise, it keeps looking for the appropriate topologies for different sub-blocks until the required specifications are met. In the layout generation phase of the algorithm, the same procedure is followed to find the best layout for the selected circuit.

The described algorithm is effective as long as the existing topologies in the library can meet the required specifications. But there might be some new topologies that outperform the existing ones in the library, and the algorithm fails to find these new topologies.

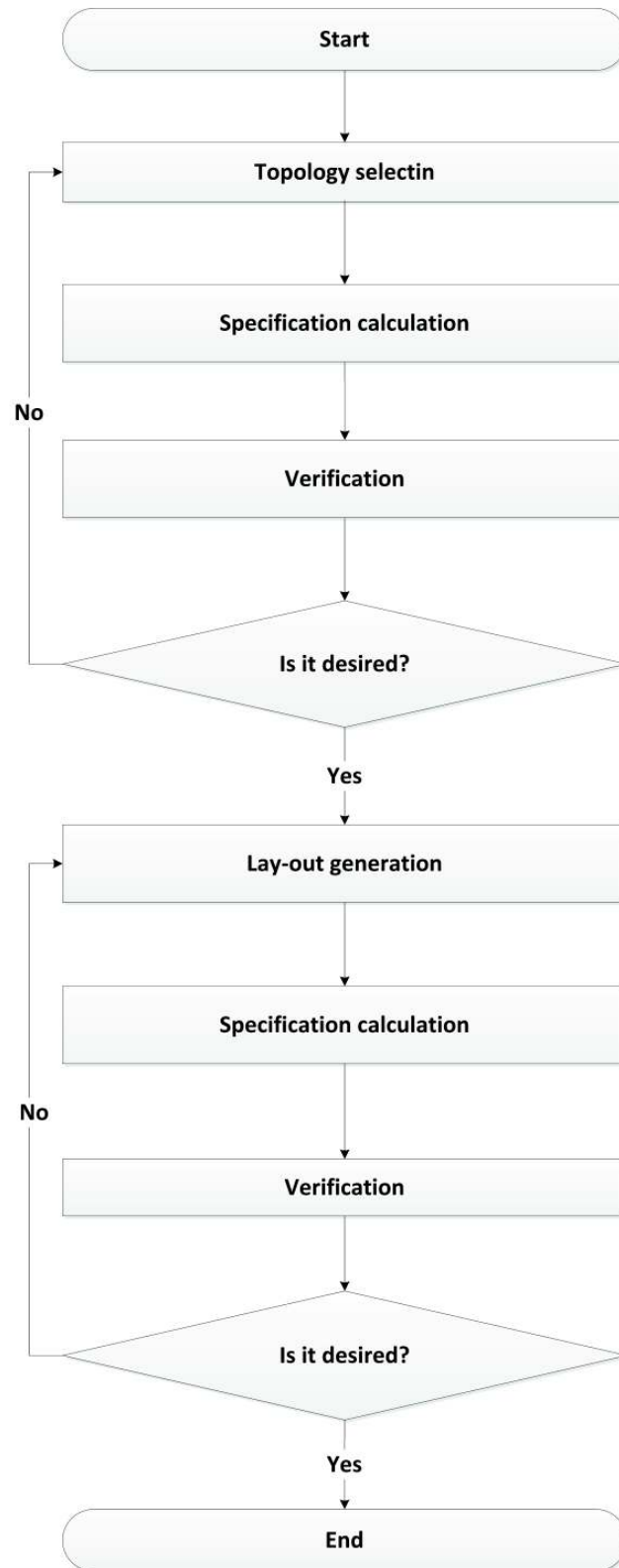


Figure 2.7: The procedure of analog circuits synthesis described in [6].

2.3.2 Generating new circuit topologies

Although, previously used circuit topologies are useful as a starting point for an analog designer, they are not efficient when a designer is looking for new topologies. There have been several attempts in the literature to generate new circuit topologies [7, 25]. In this section, two different approaches to generation of new topologies are discussed.

Manual Topology Generation

In [7, 25] a technique to generate all possible topologies for circuits with two transistors is proposed. In [7, 25] an analog circuit is represented by a small-signal two-port network with an input port and an output port as shown in Fig. 2.8. The source of the two-port network is represented by a source voltage, V_S , and a source impedance, R_S . The output port is modelled by a load impedance, R_L .

The two-port network is described by a set of ABCD parameters. These parameters relate the output voltage and current to the input voltage and current according to:

$$V_{in} = A \times V_{out} + B \times I_{out} \quad (2.6)$$

$$I_{in} = C \times V_{out} + D \times I_{out} \quad (2.7)$$

A MOS transistor is considered as a voltage controlled current source (VCCS), as shown in Fig. 2.9. The current of the VCCS is $I = g_m \cdot V$, where g_m is transconductance of the transistor.

In addition, the two-port network is modelled as a graph, where each circuit component is considered as an edge of the graph. The signal-source voltage and the load of the two-port network are modelled as two graph branches, V_{in} and R_L , respectively, and the branches V and I represents the current and the voltage in VCCS. The modelling approach in [7] is illustrated in Fig. 2.10.

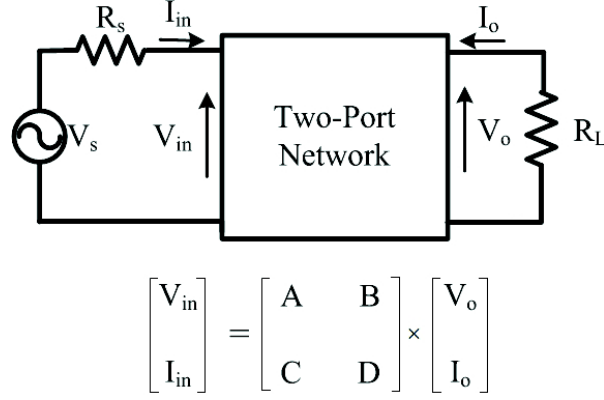


Figure 2.8: A two-port network and its ABCD matrix

By following the modelling approach in Fig. 2.10, a signal-source, a load and two VCCSs are modelled with 6 branches and 12 nodes. Different connections between these components result in a large number of different graphs. In order to reduce the number of graphs and find the functional circuits, some constraints need to be defined, which are:

- All branches must be connected in loops: if all branches are not connected in loops, the current will be zero and the circuit will be open circuit.
- No self-loops in the graph is allowed: a self-loop refers to a branch that has current enter and leave the same node. A self-loop results in a short-circuit.

Once an individual transistor graph is determined, all different graphs are generated by hand in [7, 25]. In addition, the ABCD parameters are calculated and analyzed by means of MAPLE to classify the two-ports networks. This results in 145 different graphs with non-zero ABDC parameters.

In [26, 27], all potentially useful 2-VCCS circuits generated are considered to be wide-band LNA. Graphs of wide-band amplifiers are extracted from the 2-VCCS database according

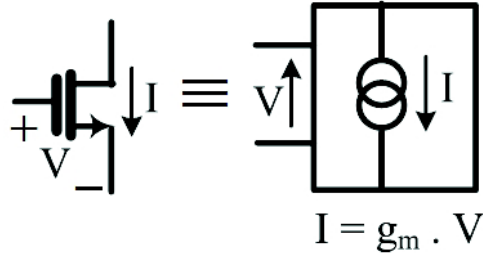


Figure 2.9: A transistor is modelled as a VCCS where g_m is the small-signal transconductance of the transistor

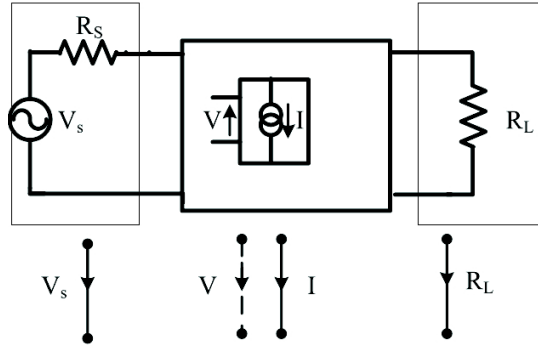


Figure 2.10: Representation of two-port network with a MOS transistor by a graph, where g_m is the small-signal transconductance of the transistor

to the previously defined constraints on the ABCD parameters. In Fig. 2.11 the flowchart procedure of selecting graphs as LNA is shown. In this procedure there are four steps, which are discussed as follows:

- Step 1: Source and load impedances requirements and amplifier functional requirements are defined. In the case of a wide-band amplifier, the input power and the output power match are required, and the DC forward gain should be greater than one. In addition, the circuit should be stable at all frequencies.
- Step 2: Based on Step 1 constraints for the ABCD parameters of the amplifiers

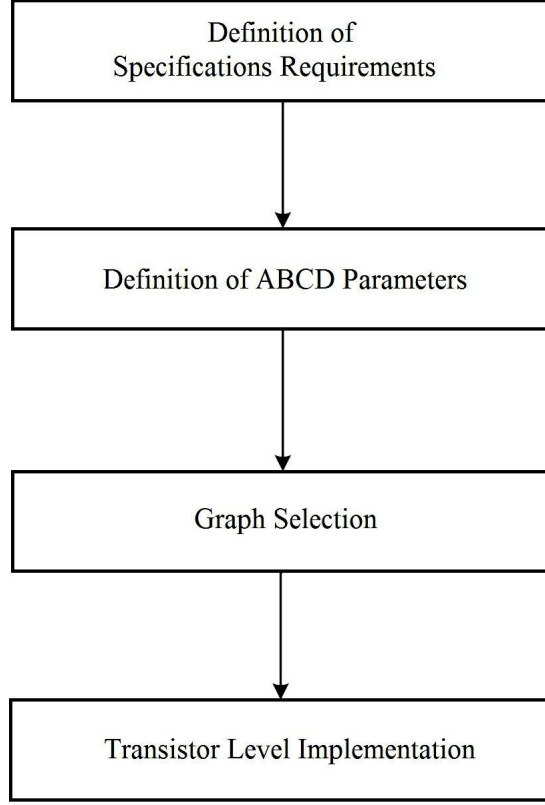


Figure 2.11: The flowchart of circuit generations algorithm proposed in [26, 27]

are defined.

- Step 3: Graphs that meet the defined constraints are chosen from the database.
- Step 4: MOS transistor circuits of the selected graphs are provided.

Finally, by following the described procedure, four elementary 2-transistor circuits, shown in Fig. 2.12 are selected and analyzed. Two of the amplifiers were known circuits, (c) and (d), and two of them were unknown, (a) and (b), to the author of [26, 27].

Automated Topology Generation

In order to avoid missing topologies and saving time when find elementary circuits, there have been attempts to generate new circuit topologies automatically. In [28], the procedure

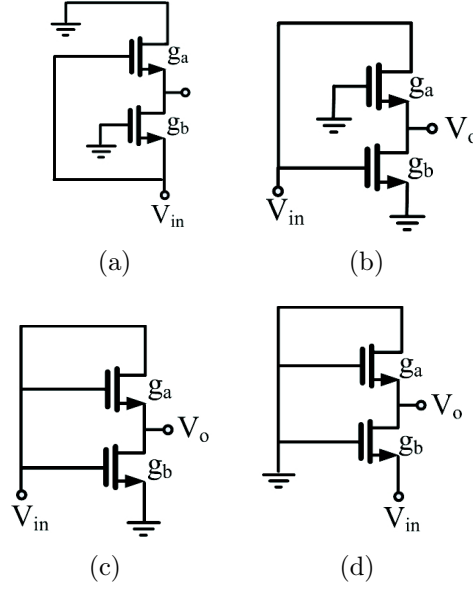


Figure 2.12: The 2-MOSFET wideband amplifiers generated in [26, 27] (biasing not shown) from [7] was automated by a computer, and all possible circuits containing two and three transistors were generated. This procedure is as follows:

- Generating all circuits with a source, a load, and two or three transistors, which were modelled as VCCSs according to the method described in [7].
- The symbolic modified nodal analysis (MNA) equations are set up by EASY-C to analyze symbolically. By using MAPLE the EASY-C results are calculated numerically.
- The generated circuits were stored with their analyzed parameters in a database.

The generated circuits in [28] are not unique, since no rules were defined to eliminate the identical circuits. From all generated topologies, the numbers of circuits with two and three transistors with specific ABCD parameters in the database are 628 and 88347, respectively, in which all generated circuits are not included. In addition, there is no mathematical proof for the total numbers of the generated circuits.

Another method, which is proposed in [29] generates circuit designs using an evolutionary

algorithm. In this work, circuit designs consist of resistors, capacitors, inductors, and bipolar transistors. These components are placed in a box between the input voltage source and the output load as shown in Fig. 2.13. Each component must have two terminals in order to be connected to other components. Therefore, a set of different architectures for a transistor is pre-defined to have only two floating terminals while the third terminal is connected to a fixed node such as ground, input voltage source, or output load.

In the beginning of the algorithm, the input node is considered as the active node and in each step of the evolutionary algorithm, there should be only one active node in the circuit. In each iteration, a new component is added to the active node in one of the following ways: 1) Between the active node and ground node 2) Between the active node and input node 3) Between the active node and output node 4) Between the active node and the previous node 5) Between the active node and a newly-created node After adding each component, the circuit is analyzed by using SPICE and the results are compared with the required specifications. Finally, the output of the genetic algorithm is the circuit with the best performance. The proposed algorithm is used to design amplifiers as well as filters. The circuit designs in this work do not exhibit the level of performance of their hand-designed predecessors even after thousands of iterations in the algorithm. Another limitation of this work, which is even stated in [29], is the inherit restriction in circuit topologies which is basically because of the way they have to represent a transistor as a two-terminal component.

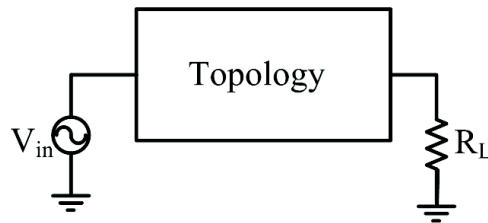


Figure 2.13: The evolved topology is located between fixed input and output terminals [29].

Although, generation new topologies results in having appropriate circuit topologies, transistor sizing and biasing are also important in a design process. Therefore, there are

different ways to optimize analog circuits, which meet the economic constraints such as time, and cost [30, 9].

2.4 Optimization Applications for Circuit Synthesis

Analog designers use optimization to synthesize their circuits [30, 32, 34]. There are different techniques to solve an optimization problem. All of these optimization techniques have their own advantages and disadvantages. Based on the size of the problem and accuracy, analog designers can select a proper technique to optimize during a design and synthesis process.

In [30], an algorithm to automate the synthesis of analog circuits was described. The first step is to generate a searchable, and flexible library. In this library, there are various known blocks, which are decomposed to several sub-blocks. A multi-objective optimization problem is used to find proper sizes for the circuit components. The power, area, and performances are considered in this problem. An evolutionary algorithm (EA) is used as an optimization method to solve the defined optimization problem [31]. EA is an algorithm for finding solutions based on the evolutionary process. After each iteration the results are stored in a set. These results are updated by comparison to previous solutions to find optimal results. Once all sub-blocks in the library are optimized, a set of sized topologies is returned.

EA is also used in [32], where the value of the components are optimized to synthesize circuits. In [32], the optimization is done by changing device parameters such as sizes and bias points, while checking the circuit performance. Different parameters are stored with the corresponding circuit performance.

Another technique to solve an optimization problem is called simulated annealing (SA). This technique was introduced by Kirkpatrick in 1983 [33]. This technique was inspired by the process of annealing in metals. In this process, a metal is heated to a certain temperature for a period of time, then cooled down to a certain temperature for a period of time.

This process repeats until the final temperature reaches the room temperature. The SA optimization is analogous to this process by using a global parameter T , analogous to temperature in the process of annealing in metallurgy. The T parameter is decreased gradually. In the each iteration, the algorithm replaces the current solution by a slightly perturbed solution. This new solution is accepted if the new solution is better than current solution, or it is accepted with a probability of $e^{-\Delta_f/T}$, where Δ_f is the difference between new function value and previous function value. After several iterations, when T is sufficiently reduces, the algorithm starts to look for the local minimum solution for the optimization problem. The higher T is, the higher the acceptance probability of a new solution is, which prevents from being trapped in the local minimum.

The flowchart of SA algorithm is shown in Fig. 2.14. In this algorithm, first the initial solution and temperature are set. Then, the stopping criteria for the algorithm is set that is a minimum temperature (similar to the room temperature in metal annealing). Also, the time that algorithm spends at each temperature is chosen for each step. At each temperature, new solutions are examined. If the cost of this new solution is less than the cost of the initial solution, the new solution is accepted. In [34], SA algorithm is employed to optimize the circuit parameters such as MOS transistors' width and length to achieve the desired circuit specifications. It can also choose the best topology among the known topologies based on the application requirements. But it is not able to generate new topologies.

The disadvantage of using EA and SA approaches for the synthesis of a large number of circuits is the long runtime. In this thesis, a constrained non-linear optimization problem is employed. The problem can be defined as:

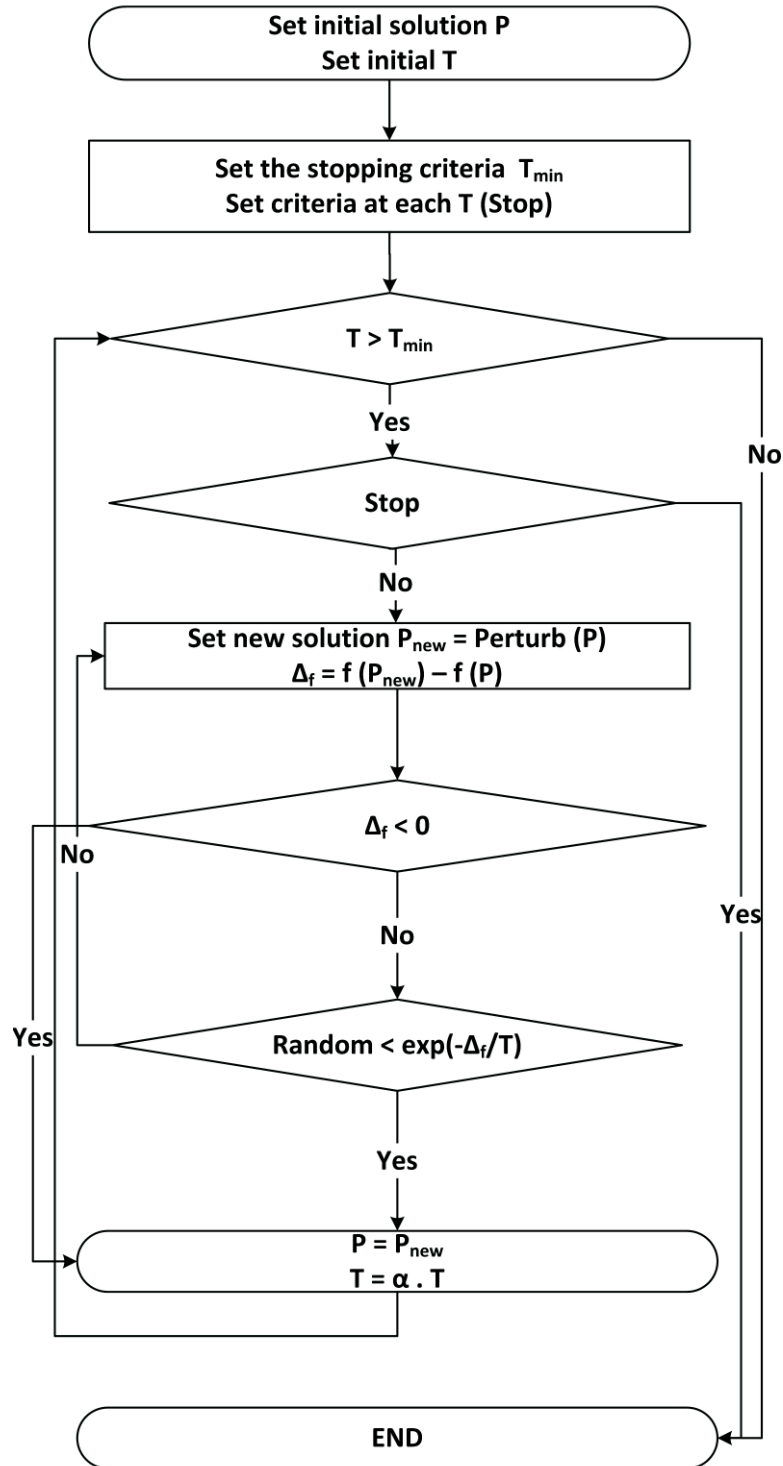


Figure 2.14: The flowchart of simulated annealing algorithm

$$\begin{aligned}
& \min_{x_1, \dots, x_n} && f_0(x_1, \dots, x_n) && (2.8) \\
& \text{s.t.} && f_i(x_1, \dots, x_n) \leq 0, i = 1, \dots, I \\
& && g_j(x_1, \dots, x_n) = 0, j = 1, \dots, J,
\end{aligned}$$

where f_0 is the objective, $f_i(x)$ $i = 1, \dots, I$ is the inequality-constraint and $g_j(x)$, $j = 1, \dots, J$ is equality-constraint. x_1, \dots, x_n are variables. There are several approaches to solve the constrained non-linear optimization problem, such as interior-point methods. Interior point method has got a linear time complexity [36].

2.5 Summary

In this chapter, different representations of analog circuit have been discussed. From these representations, matrix representation is selected to model analog circuits in this work. In addition, in this chapter the different methods to design and synthesize analog circuits have been discussed. Finally, the different optimization methods have been discussed in order to maximize the performance of the circuits. Also, circuit optimizations were discussed that could be used to achieve the best performance from analog circuits.

Chapter 3

Proposed Systematic Hierarchical Methodology

3.1 Introduction

In this chapter, an Analog Design Assistant, ADA, that can be used to generate new circuit topologies is represented. ADA automatically generates all analog circuit topologies containing two or three MOSFET transistors. Using ADA over 56,000 elementary functional circuits are found. ADA is designed to solve small-signal models of the generated circuits and, as an example, the voltage gain, the input and output impedances and the reverse gain are obtained for each of over 56,000 circuits. In addition to demonstrate optimization capabilities of ADA, a constrained optimization problem is solved for each topology to find the transistors' small-signal parameters. In this thesis, the optimization problem is set such that the maximum possible DC voltage gain of each circuit is found. The results of the optimization problem show that over 5,000 of the topologies act as input power matched amplifiers with gains higher than 1. The obtained characteristics and transistors' small-signal parameters are stored in an efficient database to be used as an assistance tool by analog designers for choosing the best circuit topology or for searching for new topologies. The pre-generated, pre-designed circuit topologies are accessible through a developed GUI, which allows a designer to select topologies by specifying the required range of circuit specifications.

This chapter is organized as follows: In Section 3.2, the analog circuit representation developed for ADA is introduced. In Section 3.3, the proposed systematic methodologies for circuit generation with two and three transistors are explained. In Section 3.4, the proof that the proposed technique covers all circuit topologies is given. The results of the proposed

methodology are given in 3.5. In Section 3.6, the small-signal characteristics of the generated circuits and their optimized versions are discussed. In Section 3.7, the GUI development of ADA tool is presented. Finally, concluding remarks are given in Section 3.8.

3.2 Circuit Representation

The first step in developing ADA was to divide circuit elements into building blocks based on their functionality. Two basic blocks were defined: an Input/Output block, I/O , and a Transistor block, T . The I/O block is made of the following 4 ports: ground (GND), input voltage (V_{in}), positive side of the load (L^+), and negative side of the load (L^-). A Transistor block, T_i , has three ports: gate (G_i), source (S_i), drain (D_i). Examples of an I/O block and a Transistor block are shown in Fig. 3.1.

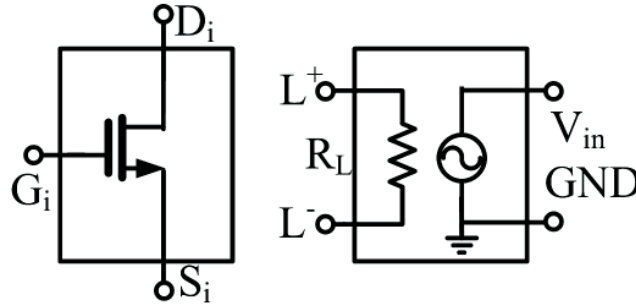


Figure 3.1: Building blocks of an analog circuit. The block on the right is the I/O block and the block on the left is the Transistor block.

Once the building blocks are defined, all possible topologies are generated by changing the internal block connections and the connections between blocks. A special connectivity matrix, CM , is designed to assist with generation of all non-duplicate topologies. The rows and columns of the connectivity matrix are organized with the ports of Input/Output block, and followed by the ports of Transistor blocks as shown in Fig. 3.2. In this figure, the matrix representation of an analog circuit containing two transistors, T_1 and T_2 , is shown. The connection between the ports of the I/O block to each other and the ports of the Transistor

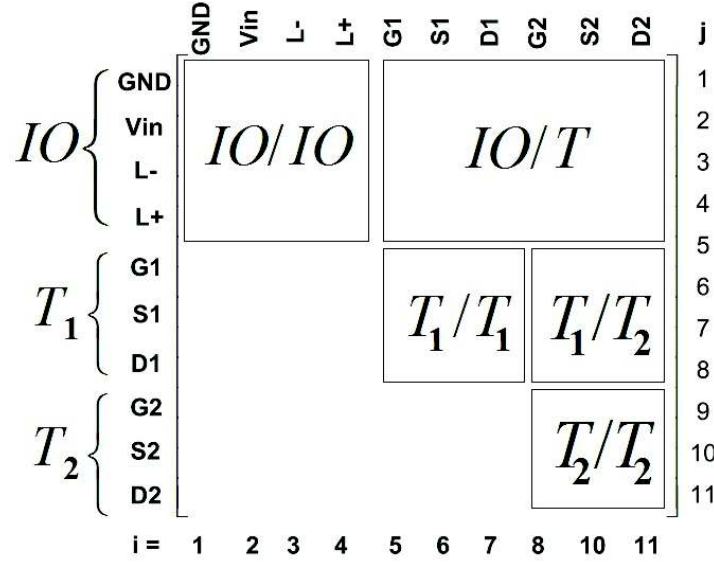


Figure 3.2: The connectivity matrix, CM , and the position of sub-matrices representing the connectivity between elements

block to each other are represented by IO/IO and T_i/T_i sub-matrices, respectively. These sub-matrices are located along the diagonal of CM . The connection between the ports of the I/O block and the ports of the transistor block are represented by IO/T sub-matrix. Finally, the connection between the ports of two different Transistor blocks, T_i , and T_j , are shown by T_i/T_j sub-matrix. As the connections between T_i and T_j and T_j and T_i are the same, CM is symmetric, and to save memory the elements below the diagonal are not stored.

3.2.1 Connectivity Matrix Construction

To construct CM , when two ports, i and j , are connected, the value of the corresponding elements, CM_{ij} and CM_{ji} , is set to 1, otherwise the value of CM_{ij} , and CM_{ji} is 0. All diagonal elements of CM are 0, since the ports cannot be connected to themselves. In Fig. 3.3 an example of circuit with two transistors, and the corresponding connectivity matrix is illustrated. In the connectivity matrix the source ($i = 6$) of (T_1) is connected to ground ($j = 1$). Therefore CM_{16} and CM_{61} are equal to 1.

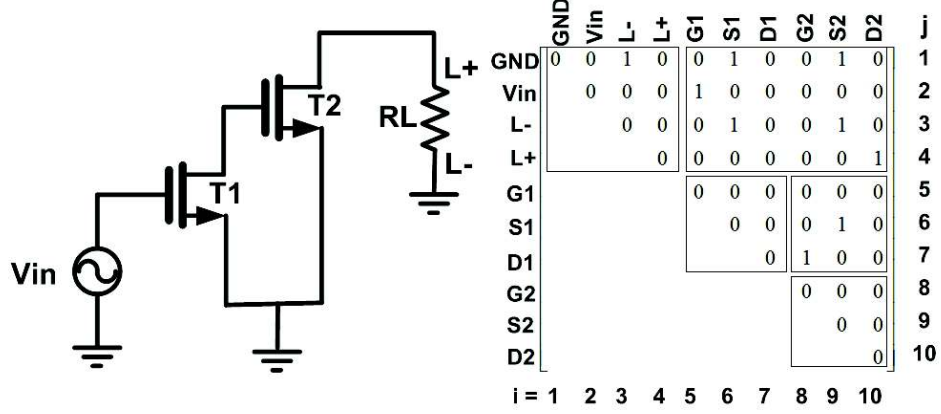


Figure 3.3: An analog circuit (biasing not shown) and corresponding connectivity matrix

3.3 Circuits Generation Using a Connectivity Matrix

In this section, a technique to generate all 2-transistor circuits by generating their corresponding CMs is presented. It is followed by discussions on generation of circuits with more transistors.

3.3.1 2-transistor Circuits Generation

To generate all possible CMs , all internal connections of the blocks, and then the interconnections between them need to be made. The main steps of the algorithm to generate CMs of all circuits with two transistors are as follows:

1. Generate internal connections
 - (a) Generate IO/IO sub-matrix
 - (b) Generate T_1/T_1 and T_2/T_2 sub-matrices
2. Generate Blocks Interconnections
 - (a) Generate T_1/T_2 sub-matrix
 - (b) Generate IO/T sub-matrix

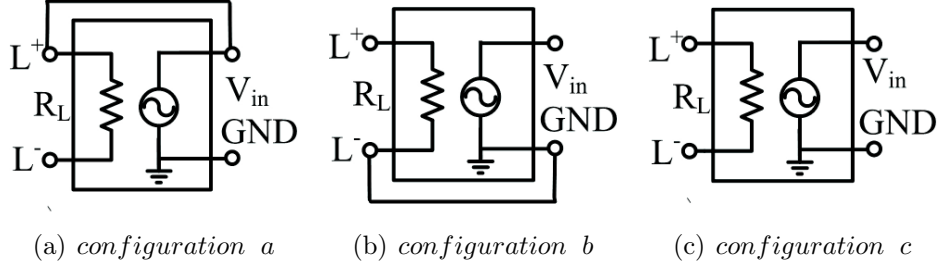


Figure 3.4: All possible I/O block configurations: (a) When one side of the load is connected to input voltage, (b) When one side of the load is connected to ground, and (c) When the ports are not connected to each other

Each step is described in detail in the following.

Internal Connections Sub-matrices

The rules to generate the internal blocks and the possible configurations for them are described in the following.

IO/IO sub-matrix generation: The sub-matrix representation of the I/O block's connectivities is a 4×4 symmetric matrix. The following circuit rules are set for this block to ensure functionality:

- The GND port cannot connect to V_{in} .
- Either of the load ports cannot connect to GND and V_{in} simultaneously.

Using the preceding rules, the IO/IO sub-matrix of CM can have three possible configurations as shown in Fig. 3.4. These configurations are referred to as *configuration a*, *configuration b*, and *configuration c* in this work and are as follows:

- *I/O configuration a*: In this configuration, one side of the load is connected to V_{in} , and the block has 3 ports to connect to other blocks' ports.
- *I/O configuration b*: In this configuration, one side of the load is connected to GND , and the block has 3 ports to connect to other blocks' ports.

- *I/O configuration c*: In this configuration, none of the IO/IO block's ports are connected to each other, and the block has 4 ports to connect to other blocks' ports.

Note that L^+ and L^- are interchangeable.

T_i/T_i sub-matrix Generation: Since a transistor has three ports, T_i/T_i sub-matrix is a 3×3 symmetric matrix. In order to have functional internal connections, the connections are made based on the following rules:

- When the gate of a transistor is connected to its source, the transistor is turned off. Therefore, the gate of a transistor is not allowed to connect to its source.
- When the drain of a transistor is connected to its source, the transistor is short circuited. Therefore, the drain of a transistor cannot be connected to its source.

Based on the above rules, the T_i/T_i sub-matrix has only two possible configurations: gate and drain are connected, or none of the ports are connected to each other.

In Fig. 3.5, the two configurations and the corresponding matrices are shown. The functionality of the two internal Transistor block configurations are described in the following:

- *T_i/T_i Configuration DCT*: In this configuration, there is a connection between drain and gate terminals of a transistor. When the drain of a transistor is connected to its gate, the transistor is called a diode-connected transistor (DCT) and acts as a non-linear resistor. In this work we are looking for amplifier circuits, and therefore circuits where all transistors are DCTs are not considered as their voltage gain is less than 1.
- *T_i/T_i Configuration NDCT*: In this configuration, the transistor's ports are not connected to each other and are denoted as non-diode-connected transistors (NDCT).

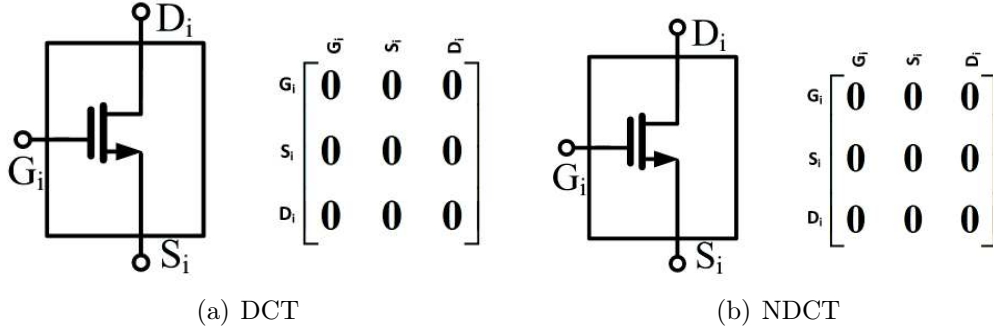


Figure 3.5: All possible T block configurations: (a) when the drain is connected to the gate (DCT) and (b) when the ports are not connected to each other (NDCT).

Generation of Blocks Interconnections

Once the I/O and Transistor block configurations have been generated, the interconnections between them needs to be determined. Circuit rules governing these interconnections and possible configurations are explained in the following.

T_i/T_j sub-matrix generation

First, the possible configurations of connections between the two transistors, i.e., T_i/T_j ($i \neq j$) sub-matrix, called Connected 2 Transistors, $C2T$, are generated. Each $C2T$ block is stored in a database for future use in connection to other circuit components. The advantage of generating $C2T$ blocks and storing them is that the number of ports considered at each stage is reduced, limiting the number of possible connections that can be made when connecting to other circuit blocks. In addition, each $C2T$ will have a small set of rules to avoid duplicate circuits, and ensure the uniqueness of larger circuits.

IO/T sub-matrix generation

At this stage, we have connected the transistors to each other and the I/O block elements to each other. But the connections between the I/O block and the transistors needs to be constructed. To build the IO/T sub-matrix, all possible interconnections between the ports of the I/O block, GND, V_{in}, L^+ and L^- , and the ports of the transistors block,

G_1, S_1, D_1, G_2, S_2 and D_2 , need to be constructed. For this, all internal connections of the blocks have to be respected. For example, if GND and L^- are connected, and if G_1 become connected to GND , $IO/T_{11} = 1$, then L^- and G_1 have to be connected as well, i.e. $IO/T_{31} = 1$. The same argument applies to any open circuit inside a block. For example, if G_1 become connected to GND , $IO/T_{11} = 1$, then there should be no connection between V_{in} and G_1 , $IO/T_{21} = 0$ as GND and V_{in} cannot be connected.

In the following the algorithm of connecting $C2T$ and I/O blocks is described. This algorithm is later implemented in MATLAB as part of ADA. The first connection is made between the first element in the I/O block, GND and the first element of the transistors, G_1 , by setting IO/T_{11} to either 1 or 0. The corresponding values of all other corresponding rows are also set to either 0 or 1 based on the internal block connections. The values of the elements of the next rows are dependent on the configuration of the I/O block and are explained as follows:

- *configuration a*: In this configuration, the second and the third row are identical, V_{in} is connected to L^- , and the values of the elements of only one row can be decided. The elements of the other two rows can be permuted depending on the internal transistor blocks configurations.
- *configuration b*: In this configuration, the first row and the third row are identical, GND is connected to L^- , and the values of the elements of only one of the rows can be decided. The elements of the other two rows can be permuted depending on the internal transistor blocks configurations.
- *configuration c*: In this configuration, all rows are independent and the permutations of the elements depend on the internal transistor blocks configuration.

The algorithm for generating the IO/T block is shown in Fig. 3.6.

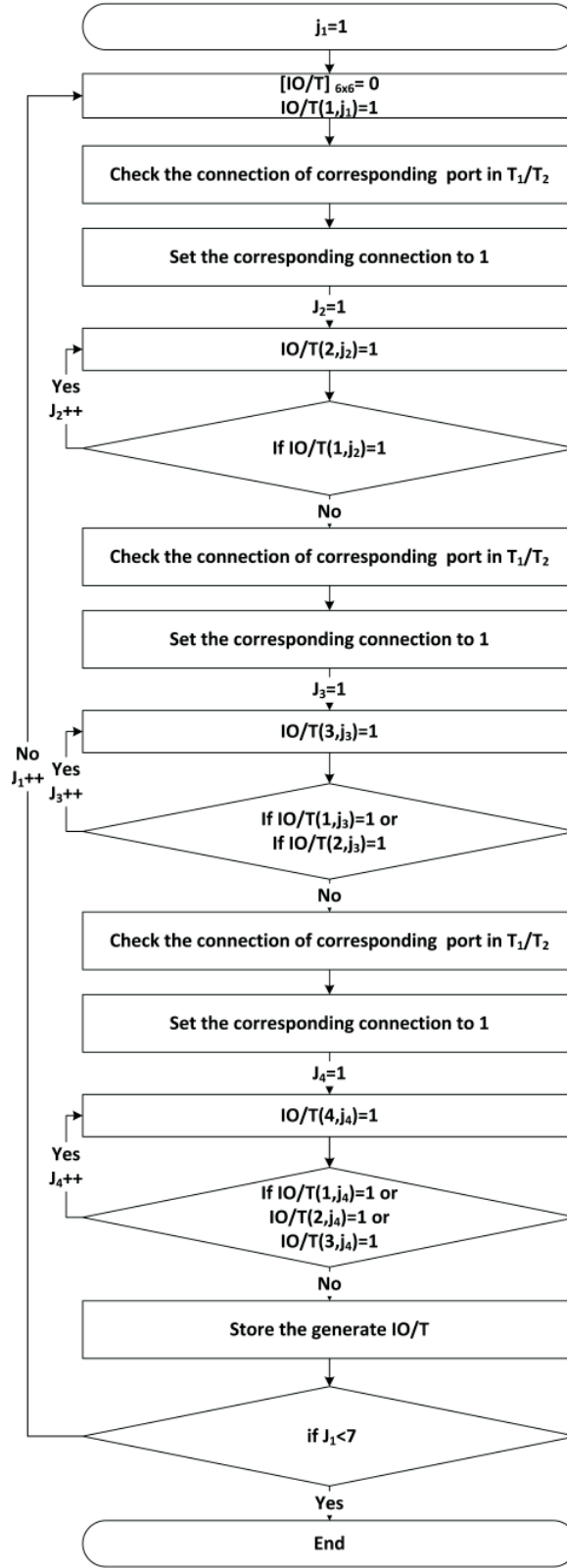


Figure 3.6: The algorithm for generation of the IO/T sub-matrix

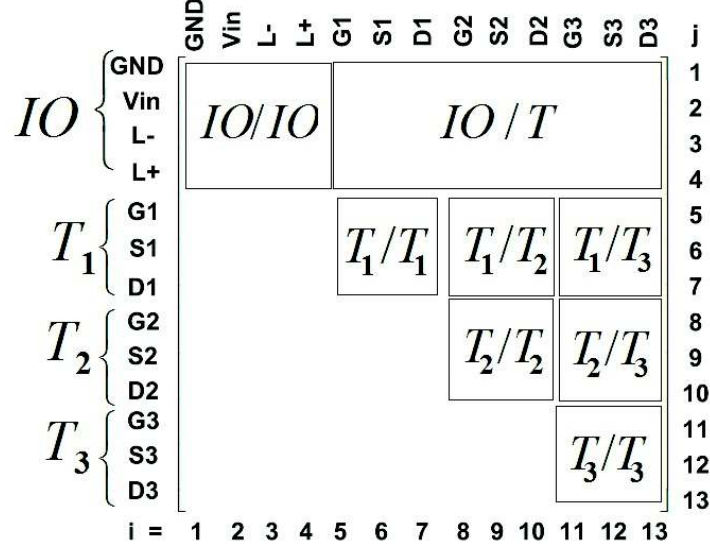


Figure 3.7: The CM for 3-transistor circuit and its sub-matrices

3.3.2 Multiple Transistors Circuits Generation

In this section, the proposed method to generate all possible circuit configurations with two transistors is expanded to the generation of circuits with a larger number of transistors. To illustrate this, circuits with three transistors are considered to show the process of extending the method to larger circuits. For circuits with three transistors, the size of the CM is increased to 13×13 , as shown in Fig. 3.7, where, the T_1/T_2 , T_1/T_3 , T_2/T_3 , and T_3/T_3 represent the connections between the three Transistor blocks, and T_3/T_3 sub-matrix shows the internal connections Transistor 3 (T_3) block. In Fig. 3.8, a sample circuit with three transistors and its CM are illustrated.

In order to generate circuits with three transistors, first internal configurations of the I/O block, and each transistor are generated as before: three possible configurations for the I/O block and two configurations for each T block. Then, the connectivity between the transistors needs to be determined. First, in order to generate T_i/T_j sub-matrices, two transistors are connected to each other to obtain the corresponding $C2T$ blocks as has been discussed in Section 3.3.1. After generating the $C2T$ block, it is connected to the third

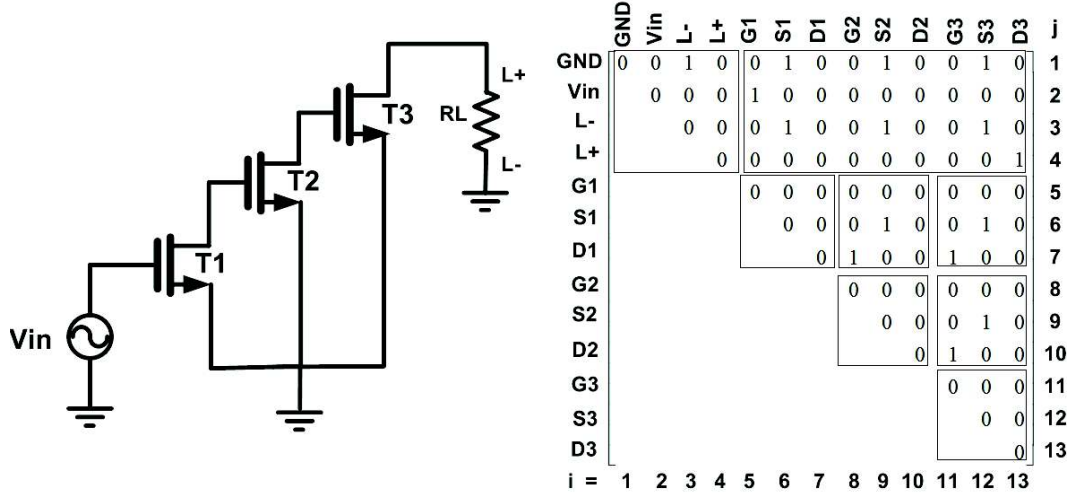


Figure 3.8: A sample circuit with three transistors (biasing not shown) and its CM .

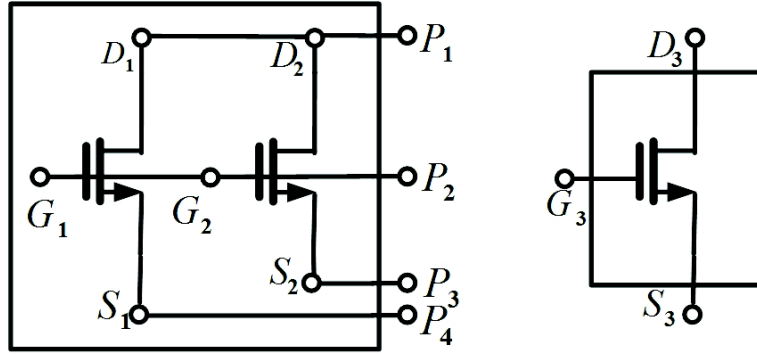


Figure 3.9: An example of a $C2T$ block and T_3 block.

transistor, T_3 . During this step, the number of ports in the $C2T$ block is related to the configuration of the transistors inside the block. Since the minimum number of connections between two transistors is 1 and the maximum is 3, the number of ports in a $C2T$ block varies between 3 to 5. In addition, T_3 has two configurations: DCT and NDCT. Note, in this thesis transistor T_1 is not allowed to be DCT to make sure that at least one of the transistors can have gain.

In Fig. 3.9, an example of a $C2T$ block and T_3 block are shown. When connected, a $C2T$ and T_3 form a block called $C3T$ block.

After generation all $C3T$ s, the final step in the proposed algorithm is to connect the I/O

block to the transistors in the similar fashion as was discussed earlier in Section 3.3.

In order to generate circuits with four transistors, the procedure is as same as generating circuits with three transistors, where all generated 3-transistor blocks ($C3T$) are connected to the fourth transistor. However, the process of generating other sub-matrices is as same as generating them for circuits with three transistors.

3.4 Theoretical Proof of Circuit Generation Methodology

In this section, the number of possible circuit topologies with two transistors is calculated. Also the maximum number of all possible $C3T$ configurations is found.

3.4.1 2-transistor Circuits

To show that all possible topologies with two transistors are found by the circuit generation method above, we first prove that the generation of the $C2T$ matrices can be reduced to non-attacking Rooks problems, and then derive the maximum number of $C2T$ blocks that can be generated.

Next, we calculate the total number of circuits that can be made using the generated $C2T$ blocks and different I/O block configurations. This number is the maximum number of circuit topologies that can be generated. Finally, we define a set of rules that differentiates the functional and non-functional circuits and calculate the number of functional circuits. The mentioned rules and procedures are described in detail in the following sections.

Generation of All C2T Blocks:

Based on non-attacking Rooks problems the number of ways to place 0 to k rooks on an $m \times n$, $n \leq m$ board, where $k = \min\{n, m\} = n$, can be calculated. There should be no more than one rook in same row or column. A Rook polynomial is a generating polynomial that the coefficients encode information about the number of ways to place k rooks on a board and powers encode information about the number of rooks [35]. A Rook polynomial

of a $m \times n$ board is:

$$R_{m,n} = \sum_{k=0}^{\min(m,n)} r_k(m,n)(x)^k, \quad (3.1)$$

where x does not represent a variable, and its coefficients, $r_k(m,n)$, represent the number of ways to place k rooks, which is x 's power. The rook polynomial is obtained by solving:

$$R_{m,n}(x) = n!x^n L_n^{m-n}(-x^{-1}), \quad (3.2)$$

where, $L_n^{m-n}(\cdot)$ is the Laguerre polynomial of degree n :

$$L_n^\alpha(x) = \sum_{i=0}^n (-1)^i \binom{n+\alpha}{n-i} \left(\frac{x^i}{i!}\right). \quad (3.3)$$

In order to calculate the total number of $C2T$ blocks by using Rook polynomial, a small set of rules are defined as follows:

Lemma 1: A $C2T$ block for two non-diode connected transistors is a matrix in which each row or column has at most one non-zero element: In the $C2T$ matrix, if there are two or more non-zero elements in a row, then the ports of the transistor represented by the non-zero columns are connected, which is unacceptable.

Lemma 2: A $C2T$ block with one DCT results in a sub-matrix in which each row or column has at most one non-zero element: If one of the transistors is diode-connected, then its gate and drain are connected. In this case the drain column of the DC transistor can be eliminated and the resultant sub-matrix representing the gate, source and drain of the NDCT makes up the rows of the new sub-matrix and the gate and the source of the DCT make the columns. Hence, this new sub-matrix is a 3×2 sub-matrix that, based on *lemma 1*, has at most one non-zero element in each row or column.

Lemma 3: Construction of the $C2T$ block is equivalent to the non-attacking Rook's problem with a 3×3 board, when there is no DCTs in the circuits, and 3×2 board, when one transistor is DCT. Based on Lemmas 1 and 2, each $C2T$ matrix, i.e. T_i/T_j sub-matrix, is a matrix, which has in each row or column only one non-zero element. When there is

no DCT in a circuit, $C2T$ matrix is a 3×3 non-attacking rooks problem. The number of rooks, k , for the $C2T$ blocks can vary from 0 to 3, where 0 means no connections between the two transistors and 3 means that the three ports of the transistors are connected to each other. When one of the transistors is diode connected (in T_i/T_j , $j > i$ only T_j can be diode-connected to have only one DCT in $C2T$ block), the column representing D_j , drain of the DCT, is not considered in the $C2T$ block, hence the $C2T$ matrix is a 3×2 non-attacking rooks problem, where the number of rooks, k , can vary from 0 to 2 and where 0 means no connections between the two transistors and 2 means that the two ports of the DCT, the gate and the source, are connected to the 2 ports of the NDCT.

Theorem 1: The number of all possible $C2T$ s is 13 when there is one DCT in the circuits and is 34 when there is no DCT in the circuits.

Based on *Lemma 3*, the $C2T$ construction problem is equivalent to the non-attacking Rooks' problem. When one transistor is diode connected, $m = 3$, $n = 2$. Hence the Rook polynomial will be:

$$R_{3,2}(x) = 6x^2 + 6x^1 + 1x^0, \quad (3.4)$$

which means that 1 configuration for zero connections between two transistors, and 6 configurations for one and two connections between two transistors each. It should be mentioned that there are no symmetric $C2T$ blocks when one of the transistors is diode connected, hence all the $C2T$ blocks are unique. The total number of configurations in this case is: $6 + 6 + 1 = 13$.

When there is no DCT in the circuits, $m = n = k = 3$, and the Rook polynomial equation will be:

$$R_{3,3}(x) = 6x^3 + 18x^2 + 9x + 1x^0. \quad (3.5)$$

Based on (3.5), there is only one way to arrange the two transistors when they have no connected ports. When there is only one connection, there can be at most 9 arrangements,

and when there are 2 and 3 connections, then there are 18 and 6 arrangements, respectively. Hence, when there is no DCT, we have at most $R_{3,3}(x) = 6 + 18 + 9 + 1 = 34$ configurations.

$R_{3,3}(x)$ represents the maximum number of possible $C2T$. However, some configurations can be equivalent, where one T_i/T_j sub-matrix is transpose of the other. Therefore, to check for uniqueness of a circuit, *Lemma 4* is proposed.

Lemma 4: Two $C2T$ blocks, $C2T_1$ and $C2T_2$, result in the equivalent circuit, when the corresponding matrices, T_i/T_{j_1} and T_i/T_{j_2} , are $T_i/T_{j_1}^T = T_i/T_{j_2}$ and $T_i/T_{j_1} = T_i/T_{j_2}^T$.

Swapping two transistors leads to generating two different $C2T$; however, the corresponding generated circuits are the same. When two transistors are swapped, the row and column of the T_i/T_j sub-matrices are transposed. For instance, as shown in Fig. 3.10, the T_i/T_j sub-matrix of Fig. 3.10.(a) is transpose of the T_i/T_j sub-matrix of Fig. 3.10.(b).

Theorem 2: The number of unique $C2Ts$ is 13 when there is only one DCT and the number of unique $C2T$ is 23, when there is no DCT.

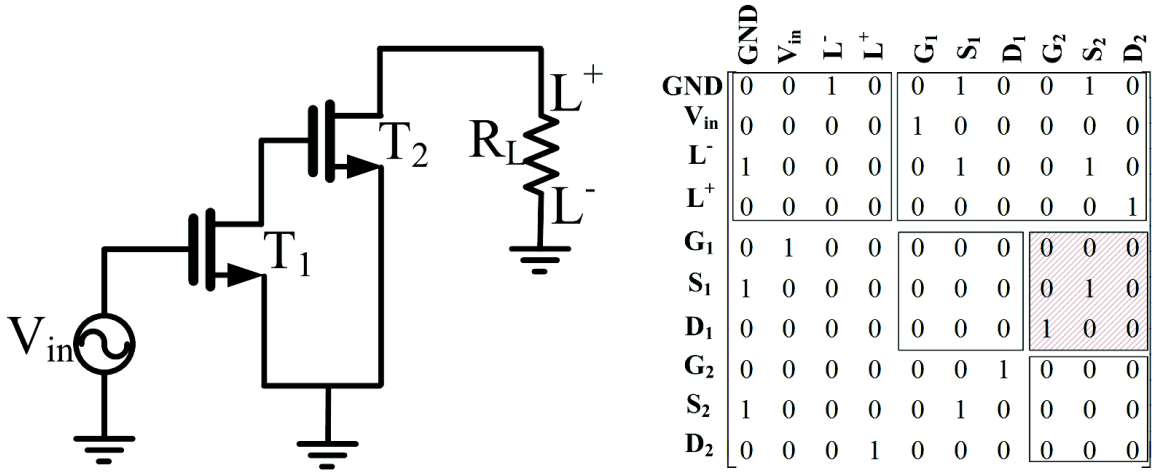
The total number of unique configurations when there is no DCT for circuits with 2 transistors, n_{NDC2k} , is calculated as follows:

$$n_{NDC2k} = S_{(3,3)_k} + \frac{R_{(3,3)_k} - S_{(3,3)_k}}{2}, \quad (3.6)$$

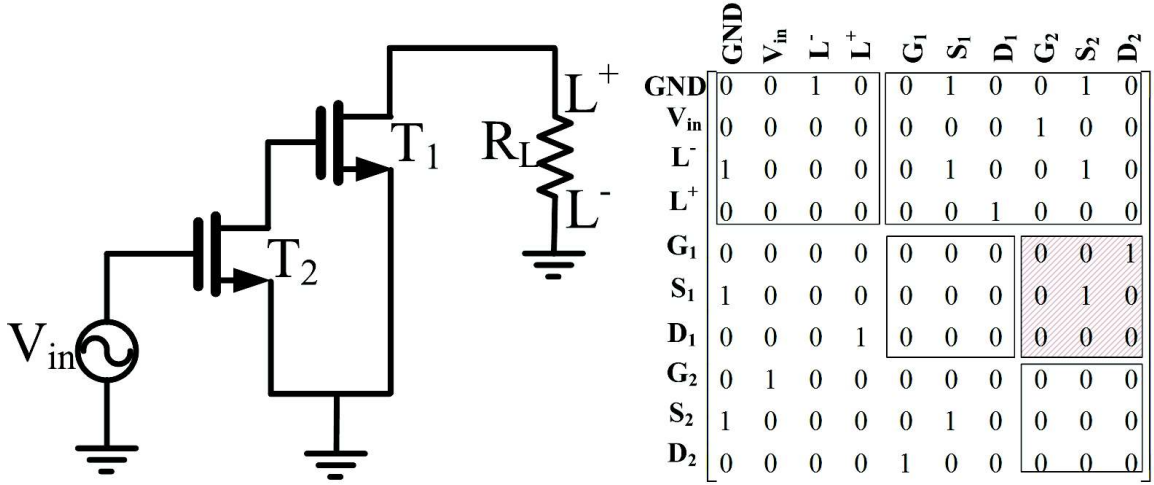
where $S_{(3,3)}$ is the number of symmetric T_i/T_j configurations, $R_{(3,3)_k} - S_{(3,3)_k}$ is the number of non-symmetric configurations, that half of them are transposed of others, as shown in Fig.3.11. A matrix is symmetric, when it is equal to its transpose, i.e. $T_i/T_j = T_i/T_j^T$.

In Fig. 3.12, the elements on the diagonal (d_1, d_2 and d_3), elements above the diagonal (u_1, u_2 and u_3), elements lower the diagonal (l_1, l_2 and l_3) of the T_i/T_j sub-matrix are shown. In order to have symmetric matrix, if the elements above the diagonal, u_k , is 1, the elements lower the diagonal, l_k , should be 1. The number of symmetric matrices of all T_i/T_j configurations is based on the number of rooks in the T_i/T_j sub-matrix, k . In particular,

- when $k = 1$, the symmetric matrix is a matrix that has the rook on one of



(a)



(b)

Figure 3.10: Two equivalent circuits with two different CM s, when two transistors are swapped (biasing not shown).

k=0			k=1			k=2			k=3		
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$		
Symmetric			Symmetric			Symmetric			Symmetric		
			$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$		
			$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$			$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$		
			$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$			$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$		
						Symmetric					
			$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$		
			Symmetric			Symmetric					
			$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$		
									Symmetric		
			$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$					
			$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$			$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$			$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$		
									Symmetric		
			$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$			$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$			$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$		
			Symmetric						Symmetric		

Figure 3.11: All possible T_i/T_j sub-matrices.

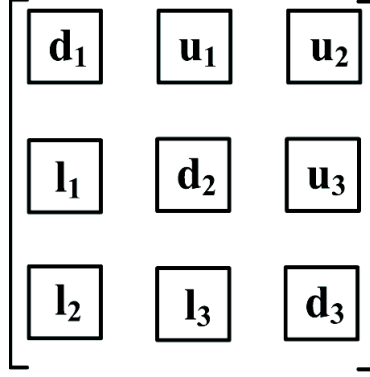


Figure 3.12: The elements on the main diagonal, d_1, d_2 , and d_3 , above the diagonal, u_1, u_2 , and u_3 , and lower the main diagonal, l_1, l_2 , and l_3 , of the T_i/T_j sub-matrix.

the elements of the main diagonal (d_1, d_2 , or d_3). Therefore, the number of symmetric matrices is: $\binom{3}{1} = 3$.

- when $k = 2$, the symmetric matrix is a matrix that has the two rooks on two elements of the main diagonal (d_1, d_2 , or d_3) or one rook on one element above the diagonal, u_1, u_2 or u_3 , and one rook on one element lower diagonal, l_1, l_2 , or l_3 . Therefore, the number of symmetric matrices is: $\binom{3}{2} + \binom{3}{1} = 3 + 3 = 6$.
- when $k = 3$, the symmetric matrix is a matrix that has the three rooks on three elements of the main diagonal (d_1, d_2 , or d_3), or one rook on one elements of the main diagonal and one rook on one element above the diagonal (u_1, u_2 or u_3) and the other one on one element lower the diagonal (l_1, l_2 or l_3). Therefore, the number of symmetric matrices is: $\binom{3}{3} + \binom{3}{1} \binom{1}{1} = 1 + 3 = 4$.

The total number of unique circuits based on the k and (3.6) are given as follows:

- When $k = 1$,

$$n_{NDC2_1} = 3 + \frac{9-3}{2} = 6. \quad (3.7)$$

- When $k = 2$,

$$n_{NDC2_2} = 6 + \frac{18-6}{2} = 12. \quad (3.8)$$

- When $k = 3$,

$$n_{NDC2_3} = 4 + \frac{6-4}{2} = 5. \quad (3.9)$$

When $k = 0$, there is no connection between the ports of two transistors, and there are 6 ports to be connected to 4 ports of IO block. This results in two transistor ports to be disconnected. Hence, when there is no connection between the transistors, the number of $C2T$ blocks is zero. The total number of unique configurations of $C2T$ block with no DCT is:

$$n_{NDC2} = 6 + 12 + 5 = 23. \quad (3.10)$$

When there is one DCT in the $C2T$, swapping the transistors results in different circuits. Therefore, the number of unique $C2T$ is 13, when one of the transistor is diode-connected. The total number of unique configurations for $C2T$ blocks is: $23 + 13 = 36$.

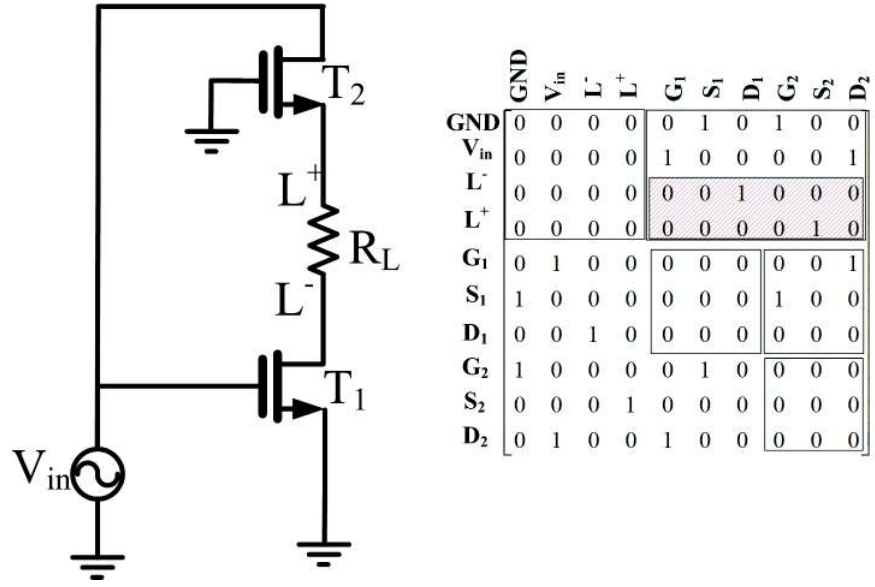
In order to connect the generated $C2T$ blocks to I/O block, there are sets of rules that are described as follows. With these rules and based on the internal connections of I/O block, the maximum number of possible connections are shown in a Tables 3.1 and 3.2.

Lemma 5: Swapping two terminals of the load, results in two identical circuit topologies. In Fig. 3.13, two identical circuit topologies are shown, with two different I/O sub-matrices. Based on *Lemma 5*, this halves the number of possible connections between I/O block, in *configuration a* or *configuration b*, and $C2T$ blocks.

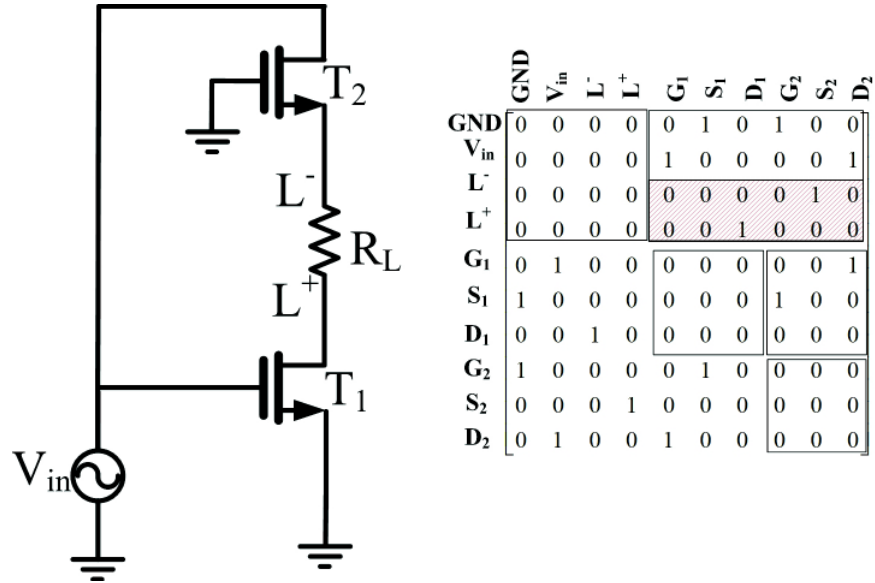
Lemma 6: When one transistor is diode-connected, the number of $C2T$ blocks ports is $5-k$. However, when there is no diode-connected in the $C2T$ block, $C2T$ has $6-k$ ports. The number of $C2T$ blocks ports depends on the number of connections between two transistors (k), and the number of diode-connected transistor in the block.

Lemma 7: After connecting the $C2T$ and I/O blocks, there should be no free port.

Free port in this work refers to a port, which is not connected to other ports in final circuit topology. For instant, if there is only one connection in the $C2T$ block, $C2T$ has 5



(a)



(b)

Figure 3.13: Two equivalent circuits with two different CMs , when two sides of the load are swapped (biasing not shown).

ports. By connecting this block to *configuration a* of *I/O* block, which has 3 ports, one port of *C2T* block will be free. A free port in a circuit results in open circuit, which is not acceptable.

Based on *Lemma 5*, *Lemma 6*, and *Lemma 7*, the maximum number of configurations of connecting the *I/O* and *C2T* blocks is shown in Tables 3.1 and 3.2 for different number of DCTs, the number of connections in the *C2T* blocks (k), and the configurations of the *I/O* block. Since the number of ports for *configuration a* and *configuration b* are both equal to 3, the number of configurations of connecting *I/O* and *C2T* blocks is shown in one table, Table 3.1.

In the first row of Table 3.1, the number of DCT in the *C2T* is shown. The columns from 2 to 5 show the number of connections (k) in *C2T*, when there is no DCT. The columns from 6 to 8 show the number of connections (k) in *C2T*, when there is one DCT. In the third row, the number of *C2T* blocks for each k is represented based on (3.7), (3.8) and (3.9) when there is no DCT, and based on (3.4) when there is one DCT. In the fourth row, the number of the *C2T* ports is shown based on *Lemma 6*. In row 5, the number of configurations to connect each *C2T* and the *I/O* block is represented, which are discussed as follows:

- The number of connections between the each *C2T* and the *I/O* block, when there is no DCT, based on k is obtained as:

$k = 0$: In this case, *C2T* block has 6 ports, which all of them are free. Based on *Lemma 7*, the number of configurations to connect the 6 ports of *C2T* and 3 ports of the *I/O* block is zero.

$k = 1$: In this case, *C2T* block has 5 ports, which 4 ports are free and 1 port is connected. Based on *Lemma 7*, the number of configurations to connect the 5 ports of *C2T* and 3 ports of the *I/O* block is zero.

$k = 2$: In this case, *C2T* block has 4 ports, which 2 ports are free and 2 ports

are connected. Therefore, the number of configurations to connect $C2T$ block ports to 3 ports of the I/O block is: $\binom{2}{1}3!$.

$k = 3$: In this case, $C2T$ block has 3 ports, which all ports are connected. Therefore, the number of configurations to connect $C2T$ block ports to 3 ports of the I/O block is $3!$.

- The number of connections between the each $C2T$ and the I/O block, when there is one DCT, based on k is obtained as:

$k = 0$: In this case, $C2T$ block has 5 ports, which 4 ports are free and 1 port is connected. Based on *Lemma 7*, the number of configurations to connect the 6 ports of $C2T$ and 3 ports of the I/O block is zero.

$k = 1$: In this case, $C2T$ block has 4 ports. The number of connected ports depends on the way of connection. If one port of transistor 1, is connected to gate or drain of transistor 2, the DCT, the $C2T$ block has 3 free ports and 1 connected ports. Therefore, the number of configurations to connect $C2T$ block ports to 3 ports of the I/O block is $3!$.

If one port of transistor 1, T_1 , is connected to source of transistor 2, the DCT, the $C2T$ block has 2 free ports and 2 connected ports. Therefore, the number of configurations to connect $C2T$ block ports to 3 ports of the I/O block is $\binom{2}{1}3!$.

From 6 $C2T$ blocks, 3 of them have 2 free ports and 2 connected ports and 3 of $C2T$ s have 3 free ports and 1 connected port. The number of free ports for each $C2T$ block is obtained by drawing all $C2T$ blocks.

$k = 2$: In this case, $C2T$ block has 3 ports, which 1 port is free and 2 ports are connected. Therefore, the number of configurations to connect $C2T$ block ports to 3 ports of the I/O block is $3!$.

In row 6, the number of configurations to connect the all $C2T$ blocks and the I/O block is represented for each k . In this row, for each column the numbers in row 3 is multiplied by the numbers in row 5, except column 7, which is $3 \times \left(\binom{2}{1} 3! \right) + 3 \times 3!$.

In Table. 3.2, the number of configurations to connect the all $C2T$ blocks and the I/O block is shown. Four first rows show the number of DCT in $C2T$ blocks, number of connections in $C2T$ blocks, number of total $C2T$ blocks, and the number of $C2T$ block ports, respectively. In row fifth, the number of connections between the each $C2T$ and the I/O block is represented, which are discussed as follows:

- The number of configurations to connect the each $C2T$ and the I/O block, when there is no DCT, based on k is obtained as:

$k = 0$: In this case, $C2T$ block has 6 ports, which all of them are free. Based on *Lemma 7*, the number of configurations to connect the 6 ports of $C2T$ and 4 ports of the I/O block is zero.

$k = 1$: In this case, $C2T$ block has 5 ports, which 4 ports are free and 1 port is connected. Therefore, the number of configurations to connect $C2T$ block ports to 4 ports of the I/O block is: $4!$. Based on *Lemma 5*, the number of unique configurations is $\frac{4!}{2}$.

$k = 2$: In this case, $C2T$ block has 4 ports, which 2 ports are free and 2 ports are connected. Therefore, the number of configurations to connect $C2T$ block ports to 4 ports of the I/O block is: $4!$. Based on *Lemma 5*, the number of unique configurations is $\frac{4!}{2}$.

$k = 3$: In this case, $C2T$ block has 3 ports, which all ports are connected. Therefore, the number of configurations to connect $C2T$ block ports to 4 ports of the I/O block is zero.

- The number of configurations to connect the each $C2T$ and the I/O block,

Table 3.1: The maximum number of configurations of connecting I/O and $C2T$ blocks based on k and the number of diode-connected transistors (DCTs) for *configuration a* and *configuration b*

Description	0DCT				1DCT		
	k=0	k=1	k=2	k=3	k=0	k=1	k=2
# $C2T$ blocks	0	6	12	5	1	6	6
# $C2T$ ports	6	5	4	3	5	4	3
# configurations to connect each $C2T$ to IO	0	0	$2 \times 3!$	$3!$	0	$2 \times 3!$ or $3!$	$3!$
# configurations to connect all $C2T$ to IO	0	0	12×12	$3! \times 5$	0	$12 \times 3 + 3 \times 6$	$3! \times 6$
Total	174				90		

when there is one DCT, based on k is obtained as:

$k = 0$: In this case, $C2T$ block has 5 ports, which 4 ports are free and 1 port is connected. Therefore, the number of configurations to connect $C2T$ block ports to 4 ports of the I/O block is: $4!$. Based on *Lemma 5*, the number of unique configurations is $\frac{4!}{2}$.

$k = 1$: In this case, $C2T$ block has 4 ports, which 2 ports are free and 2 port are connected. Therefore, the number of configurations to connect $C2T$ block ports to 4 ports of the I/O block is: $4!$. Based on *Lemma 5*, the number of unique configurations is $\frac{4!}{2}$.

$k = 2$: In this case, $C2T$ block has 3 ports, which 1 port is free and 2 port are connected. Therefore, the number of configurations to connect $C2T$ block ports to 4 ports of the I/O block is zero.

In row 6, the number of configurations to connect the all $C2T$ blocks and the I/O block is represented for each k . In this row, for each column the numbers in row 3 is multiplied by the numbers in row 5.

The maximum shown in Tables 3.1 and 3.2, are both functional and non-functional circuit topologies. The non-functional topologies in this thesis refer to topologies in which:

- the gates of different transistors are connected to each other when there are no connections to the gates.
- the gate of a transistor is connected to load when there are no other connections to that node.

Table 3.2: The maximum number of configurations of connecting I/O and $C2T$ blocks based on k and the number of diode-connected transistors (DCTs) for *configuration c*

Description	0DCT				1DCT		
	k=0	k=1	k=2	k=3	k=0	k=1	k=2
# $C2T$ blocks	0	6	12	5	1	6	6
# $C2T$ ports	6	5	4	3	5	4	3
# configurations to connect each $C2T$ to IO	0	$\frac{4!}{2}$	$\frac{4!}{2}$	0	$\frac{4!}{2}$	$\frac{4!}{2}$	0
# configurations to connect all $C2T$ to IO	0	$\frac{4!}{2} \times 6$	$\frac{4!}{2} \times 12$	0	$\frac{4!}{2}$	$\frac{4!}{2} \times 6$	0
Total	216				84		

The number of functional circuit topologies based on *Lemmas 1-7*, *Theorem 1* and *Theorem 2* are less than the maximum number of circuit topologies shown in Tables 3.1 and 3.2, 828, that will be determined in Section 3.5.

3.4.2 3-transistor Circuits

In this section, the total number of $C3T$ blocks is considered. Based on the number of connected ports in $C2T$ blocks, the maximum number of unique ways to connect three transistors to obtain functional circuits can be calculated by using non-attacking Rooks problem.

Lemma 8: When there is no DCT in $C2T$ blocks, and Transistor 3 is NDCT, the maximum number of $C3T$ is 471.

In order to generate circuits with two transistors, the $C2T$ block when there is no connection in $C2T$ is not considered. However, the $C2T$ block with no connection is considered in generation of $C3T$ blocks. The maximum number of $C3T$ are obtained as follows:

- When there is no connection in $C2T$ blocks, this block has 6 output ports. Therefore, the connection between these 6 ports and 3 ports of Transistor 3 can be represented by 6×3 matrix that has only one elements in each row or column. The maximum number of all generated matrices can be obtained by $R_{6,3}(x)$:

$$R_{6,3}(x) = 120x^3 + 90x^2 + 17x^1 + 1x^0, \quad (3.11)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 17 configurations for one connection, 90 configurations for two connections, and 120 configurations for three connections. The total number of configurations is: $120 + 90 + 17 + 1 = 228$.

- When there is one connection in $C2T$ block, this block has 5 output ports. Therefore, the connection between these 5 ports and 3 ports of Transistor 3 can be represented by 5×3 matrix that has only one elements in each row or column. The maximum number of all generated matrices can be obtained by $R_{5,3}(x)$:

$$R_{5,3}(x) = 60x^3 + 60x^2 + 15x^1 + 1x^0, \quad (3.12)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 15 configurations for one connection, 60 configurations for two connections, and three connections. The total number of configurations is: $1 + 15 + 60 + 60 = 136$.

- When there is two connections in $C2T$ block, this block has 4 output ports. Therefore, the connection between these 4 ports and 3 ports of Transistor 3

can be represented by 4×3 matrix. Therefore, the maximum number of all generated matrices can be obtained by $R_{4,3}(x)$:

$$R_{4,3}(x) = 24x^3 + 36x^2 + 12x^1 + 1x^0, \quad (3.13)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 12 configurations for one connection, 36 configurations for two connections, and 24 configurations for three connections. The total number of configurations is: $24 + 36 + 12 + 1 = 73$.

- When there is three connection in $C2T$ block, this block has 3 output ports. Therefore, the connection between these 3 ports and 3 ports of Transistor 3 can be represented by 3×3 matrix. Therefore, the maximum number of all generated matrices can be obtained by $R_{3,3}(x)$:

$$R_{3,3}(x) = 6x^3 + 18x^2 + 9x^1 + 1x^0, \quad (3.14)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 9 configurations for one connection, 18 configurations for two connections, and 6 configurations for three connections. The total number of configurations is: $6 + 18 + 9 + 1 = 34$.

Finally, the maximum number of $C3T$ blocks, when there is no DCT, is:

$$228 + 136 + 73 + 34 = 471.$$

Lemma 9: When there is one DCT in $C2T$ blocks, and Transistor 3 is NDCT, the maximum number of $C3T$ is 243.

The maximum number of $C3T$ can be obtained as follows:

- When there is no connection in $C2T$ block, this block has 5 output ports. Therefore, the connection between these 5 ports and 3 ports of Transistor 3 can be represented by 5×3 matrix. Therefore, the maximum number of all generated matrices can be obtained by $R_{5,3}(x)$:

$$R_{5,3}(x) = 60x^3 + 60x^2 + 15x^1 + 1x^0, \quad (3.15)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 15 configurations for one connection, 60 configurations for two connection, and three connections. The total number of configurations is: $1 + 15 + 60 + 60 = 136$.

- When there is one connection in $C2T$ block, this block has 4 output ports. Therefore, the connection between these 4 ports and 3 ports of Transistor 3 can be represented by 4×3 matrix. Therefore, the maximum number of all generated matrices can be obtained by $R_{4,3}(x)$:

$$R_{4,3}(x) = 24x^3 + 36x^2 + 12x^1 + 1x^0, \quad (3.16)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 12 configurations for one connection, 36 configurations for two connections, and 24 configurations for three connections. The total number of configurations is: $1 + 12 + 36 + 24 = 73$.

- When there is two connections in $C2T$ block, this block has 3 output ports. Therefore, the connection between these 3 ports and 3 ports of transistor 3 can be represented by 3×3 matrix. Therefore, the maximum number of all generated matrices can be obtained by $R_{3,3}(x)$:

$$R_{3,3}(x) = 6x^3 + 18x^2 + 9x^1 + 1x^0, \quad (3.17)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 9 configurations for one, 18 configurations for two connections, and 6 configurations for three connections. The total number of configurations is: $1 + 9 + 18 + 6 = 34$.

Therefore, the maximum number of $C3T$ blocks, when there is one DCT, is $136 + 73 + 34 = 243$.

Lemma 10: When there is no DCT in $C2T$ blocks, and transistor 3 is DCT, the maximum number of $C3T$ is 106.

The maximum number of $C3T$ can be obtained as follows:

- When there is no connection in $C2T$ block, this block has 6 output ports. Therefore, the connection between these 6 ports and 2 ports of Transistor 3 can be represented by 6×2 matrix. Therefore, the maximum number of all generated matrices can be obtained by $R_{6,2}(x)$:

$$R_{6,2}(x) = 30x^2 + 12x^1 + 1x^0, \quad (3.18)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 12 configurations for one connection, 30 configurations for two connections. The total number of configurations is: $1 + 12 + 30 = 43$.

- When there is one connection in $C2T$ block, this block has 5 output ports. Therefore, the connection between these 5 ports and 2 ports of Transistor 3 can be represented by 5×2 matrix. Therefore, the maximum number of all generated matrices can be obtained by $R_{5,2}(x)$:

$$R_{5,2}(x) = 20x^2 + 10x^1 + 1x^0, \quad (3.19)$$

which means that 1 configuration for zero connections between two blocks

($C2T$ and T_3), and 10 configurations for one connection, 20 configurations for two connections. The total number of configurations is: $1 + 10 + 20 = 31$.

- When there is two connections in $C2T$ block, this block has 4 output ports. Therefore, the connection between these 4 ports and 2 ports of Transistor 3 can be represented by 4×2 matrix. Therefore, the maximum number of all generated matrices can be obtained by $R_{4,2}(x)$:

$$R_{4,2}(x) = 12x^2 + 6x^1 + 1x^0, \quad (3.20)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 6 configurations for one connection, 12 configurations for two connections. The total number of configurations is: $1 + 6 + 12 = 19$.

- When there is three connections in $C2T$ block, this block has 3 output ports. Therefore, the connection between these 3 ports and 2 ports of Transistor 3 can be represented by 3×2 matrix. Therefore, the maximum number of all generated matrices can be obtained by $R_{3,2}(x)$:

$$R_{3,2}(x) = 6x^2 + 6x^1 + 1x^0, \quad (3.21)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 6 configurations for one connection, 6 configurations for two connections. The total number of configurations is: $1 + 6 + 6 = 13$.

Therefore, the maximum number of $C3T$ blocks, when there is no DCT, is $43 + 31 + 19 + 13 = 106$.

Lemma 11: When there is one DCT in $C2T$ blocks, and Transistor 3 is DCT, the maximum number of $C3T$ is 63.

The maximum number of $C3T$ can be obtained as follows:

- When there is no connection in $C2T$ block, this block has 5 output ports. Therefore, the connection between these 5 ports and 2 ports of Transistor 3 can be represented by 5×2 matrix. Therefore, the maximum number of all generated matrices can be obtained by $R_{5,2}(x)$:

$$R_{5,2}(x) = 20x^2 + 10x^1 + 1x^0, \quad (3.22)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 10 configurations for one connection, 20 configurations for two connections. The total number of configurations is: $1 + 10 + 20 = 31$.

- When there is one connection in $C2T$ block, this block has 4 output ports. Therefore, the connection between these 4 ports and 2 ports of Transistor 3 can be represented by 4×2 matrix. Therefore, the maximum number of all generated matrices can be obtained by $R_{4,2}(x)$:

$$R_{4,2}(x) = 12x^2 + 6x^1 + 1x^0, \quad (3.23)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 6 configurations for one connection, 12 configurations for two connections. The total number of configurations is: $1 + 6 + 12 = 19$.

- When there is two connections in $C2T$ block, this block has 3 output ports. Therefore, the connection between these 3 ports and 2 ports of Transistor 3 can be represented by 3×2 matrix. Therefore, the maximum number of all generated matrices can be obtained by $R_{3,2}(x)$:

$$R_{3,2}(x) = 6x^2 + 6x^1 + 1x^0, \quad (3.24)$$

which means that 1 configuration for zero connections between two blocks ($C2T$ and T_3), and 6 configurations for one connection, 6 configurations for two connections. The total number of configurations is: $1 + 6 + 6 = 13$.

- It is not possible to have three connections in $C2T$ block, when one transistor is DCT.

Therefore, the maximum number of $C3T$ blocks, when there is one DCT, is $31 + 19 + 13 = 63$. Finally, the total number for $C3T$ is $471 + 243 + 106 + 63 = 883$. These number is the maximum number for $C3T$, where equivalent circuits are not excluded. The equivalences can be reduced by eliminating equivalent circuits, which is done by removing matrix transposes of T_1/T_2 , T_2/T_3 , and T_1/T_3 .

Rooks number for $C3T$ does not show details regarding the number of free output ports and the number of connected ports for each configuration of $C3T$. This numbers are needed to calculate the maximum number of configurations for connecting the $C3T$ block to the I/O block.

3.5 Circuit Generation Results

The proposed algorithm for two and three transistor circuit construction was implemented in MATLAB. All possible circuit structures were generated and, as an illustration of the software capabilities, all amplifier circuits were stored in a database.

2-transistor structures

For the two transistor case a total of 582 circuits were generated. In Table 3.3, the numbers of the generated circuits based on the three I/O block configurations, and the number of diode-connected transistors (DCTs) in the circuits, are given. In this table, column 1 is the I/O configuration, columns 2 and 3 show the number of the circuits when there is

Table 3.3: The number of generated 2-transistor circuits based on the *I/O* block configurations and the number of diode-connected transistors (DCTs) in the circuits

I/O block	0DCT	1DCT	Total
configuration a	142	74	216
configuration b	132	74	206
configuration c	112	48	160
total	386	196	582

Table 3.4: The number of generated 3-transistor circuits based on the *I/O* block configurations and the number of diode-connected transistors (DCTs) in the circuits

I/O block	0DCT	1DCT	2DCTs	Total
configuration a	6338	6340	2140	14818
configuration b	6338	6340	1920	14598
configuration c	16440	8354	2070	26864
All	29116	21034	6130	56280

no DCT in a circuit, 0DCT, and when there is one DCT in a circuit, 1DCT, respectively. Finally, column 4 represents the total number of the circuits generated for the configuration.

3-transistor structures

In total 56,280 circuits containing three transistors were generated. In Table 3.4, the number of possible circuits for the three *I/O* block configurations are given. In this table, column 1 is the *I/O* configuration, columns 2 to 4 show the results for different number of DCTs, and column 5 represents the total number of the circuits generated for each *IO* configuration.

Since in this thesis the identical circuits are eliminated, the total numbers of all generated 2- and 3- transistor circuits, 582 and 56,280, are less than the total number of the generated 2- and 3-transistor circuits with defined specifications in [28], which are 628 and 88,347, respectively. It should be emphasized that the 628 and 88,347 generated circuits do not include all possible generated circuits in [28].

3.6 Circuit Analysis

To improve the functionality of analog circuit design assistant tool, once all circuit structures have been constructed, the tool should allow a designer to select a desired circuit. For this, the performance of each circuit, for example its DC voltage gain, are calculated by using CM and transistors' small-signal parameters, g_m and r_{ds} (transistor output resistance), and R_L .

While there could be a number of circuit performance metrics of interest to a designer, in this thesis only amplifier circuits are considered with performance requirements of:

- Forward Gain (A_f) : In order to amplify an input signal, the ratio of output voltage signal to the input voltage signal needs to be greater than 1.
- Reverse Gain (A_{rev}) : The reverse gain determines the isolation of the output of an amplifier from its input, and should be less than $1/A_f$.
- Input Impedance (Z_{in}) : To minimize power loss due to reflections, the input power matching is required. The input power matching is achieved when the input impedance is equal to the conjugate of the signal-source impedance, which is usually 50Ω . Also, the real part of Z_{in} should be positive to avoid instability.
- Output Impedance (Z_o) : If the amplifier's load is another on-chip circuit then the output impedance is not usually a critical performance parameter. However, if the load is connected to an off-chip circuit then Z_o should be a conjugate of the off-chip circuit input impedance, which is usually 50Ω . The real part of the Z_o should be positive to avoid instability.

After defining the required specifications, a technique to calculate the optimal small-signal parameters is proposed and implemented.

3.6.1 Circuit Optimization

In order to achieve an optimal performance, the maximum gain in this work, it is proposed to solve an optimization problem:

$$\begin{aligned}
& \max_{g_m, R_L} && A_f(g_m, R_L) && (3.25) \\
& \text{s.t.} && R_{in_l} \leq R_{in} \leq R_{in_u} \\
& && g_{m_l} \leq g_m \leq g_{m_u} \\
& && g_{ds} = 0.1 \times g_m \\
& && A_{rev} < \frac{1}{A_f} \\
& && R_{L_l} \leq R_L \leq R_{L_u},
\end{aligned}$$

where, g_{m_u} and g_{m_l} are the upper bound and lower bound of the g_m , R_{in_u} and R_{in_l} are the upper bound and lower bound of the R_{in} , R_{L_u} and R_{L_l} are the upper bound and lower bound of the R_L , respectively. In this thesis, the limits on g_m are set based on the experience with designing analog circuits in modern CMOS technologies. the $g_{m_u} = 0.1 \text{ A/V}$, and $g_{m_l} = 0.001 \text{ A/V}$. To set $g_{m_u} = 0.1 \text{ A/V}$, a relatively large transistor consuming significant amount of power is required.

In this optimization problem, input power match for the analog circuit topologies is considered. Therefore, the R_{in_u} is set to 100Ω and R_{in_l} is set to 25Ω .

In the optimization, the maximum load resistance is set to a large $10\text{k}\Omega$ value to represent a high input impedance of the following circuit and the lowest value of R_L is set to.

The optimization problem in (3.25) is solved using non-linear programming techniques and the best values of g_m and R_L that maximize the gain are stored along the CM and other circuit parameters.

The voltage gain functions may not be concave. A concave function is a function that for any x in an interval, χ , the $f(x)$ is less than $f(x^*)$, where x^* is the global optimal value

Table 3.5: The initial points for optimization problem

Variables	Initial points
$g_{m_1}(A/V)$	0.02
$g_{m_2}(A/V)$	0.02
$g_{m_3}(A/V)$	0.02
$R_L(\Omega)$	10k

Table 3.6: The number of optimized 2-transistor amplifiers based on the I/O block configurations and the number of DCTs in the circuits

I/O block	0DCT	1DCT	Total	Percentage
configuration a	16	15	31	14 %
configuration b	16	15	31	15 %
configuration c	8	4	12	7 %
total	40	34	74	12.7 %

and is in χ such that [36, 37] :

$$f(x) \leq f(x^*) \text{ for all } x \in \chi,$$

Since the voltage gain functions may not be concave, the optimal value are based on the initial points and can be local optimal. The initial values for solving (3.25) are shown in Table 3.5.

All of the generated circuits are optimized to obtain the highest gain. The number of 2-transistor circuits, and 3-transistor circuits that act as amplifiers based on the number of DCTs in circuits is shown in Tables 3.6 and 3.7, respectively. In the last column of Tables 3.6 and 3.7, the percentages of amplifiers out of all generated circuits are shown.

The value of gains can vary widely for different amplifiers. In Fig. 3.14, the histogram

Table 3.7: The number of optimized 3-transistor amplifiers based on the I/O block configurations and the number of DCTs in the circuits.

I/O block	0DCT	1DCT	2DCTs	Total	Percentage
configuration a	606	803	274	1,683	11 %
configuration b	614	788	274	1,676	11%
configuration c	822	658	264	1,744	6 %
All	2,042	2,249	812	5,103	9 %

for optimized input power matched amplifiers is shown. As shown, there are 32 amplifiers with gain greater than 140. The histogram for amplifiers with gain higher than 140 is shown in Fig. 3.15. Of these, there are 4 amplifiers with gain 1100. In Fig. 3.16, the amplifier with highest gain are shown.

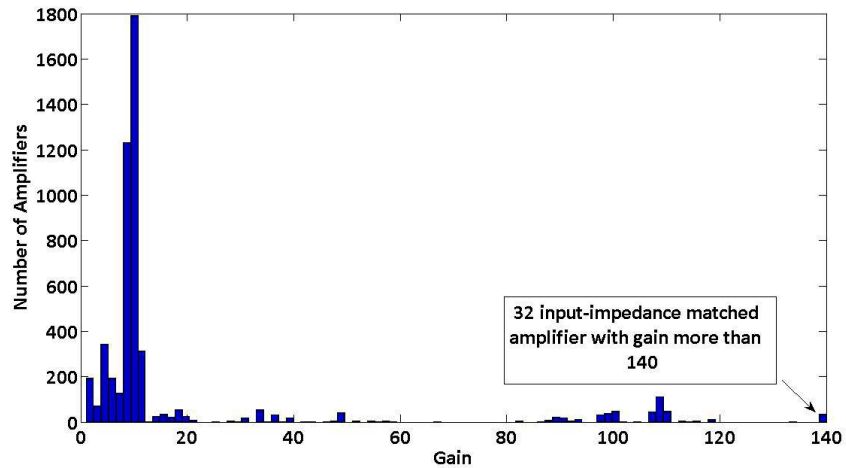


Figure 3.14: The histogram for optimized input power matched amplifiers with gain less than and equal to 140.

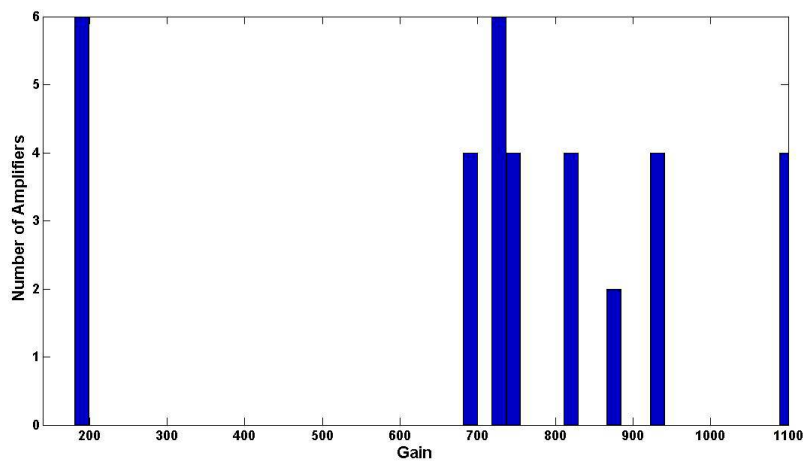


Figure 3.15: The histogram for optimized input power matched amplifiers with gain more than 140.

To show examples of circuits generated by the proposed method, three sample, previously unknown input power-matched circuits are shown in Fig. 3.17. The small-signal model of

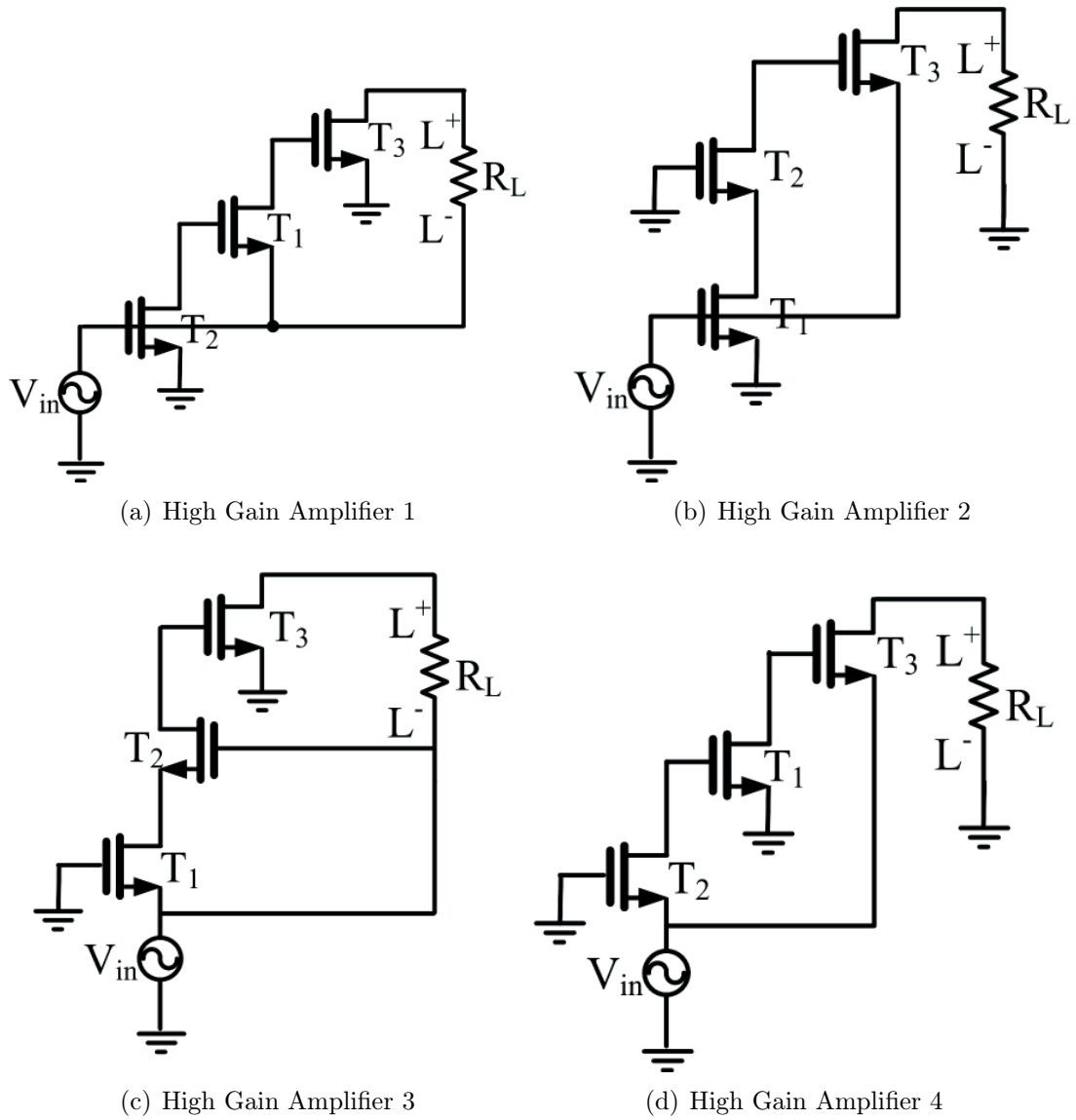


Figure 3.16: The three generated 3 transistor amplifiers with the highest gain (biasing not shown).

Table 3.8: The small-signal transistor parameters and the circuit specifications of previously unknown amplifiers in Fig. 3.17

Amplifier Number	1	2	3
A_f	9.9	99	10.8
A_{rev}	0.1	0	0.09
$Z_{in}(\Omega)$	28	100	26
$Z_o(\Omega)$	46	100	186
$g_{m_1}(A/V)$	0.03	0.1	0.05
$g_{m_2}(A/V)$	0.1	0.001	0.03
$g_{m_3}(A/V)$	0.02	0.1	0.1
$R_L(\Omega)$	10k	10k	10k

these three samples are also illustrated in Fig. 3.18. The gains and other small-signal characteristics of the circuits are given in Table 3.8.

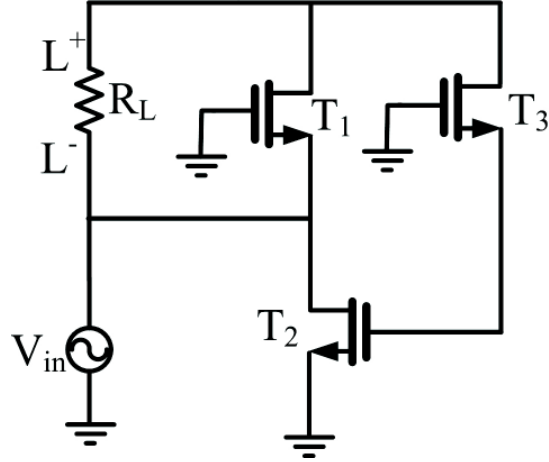
3.7 Analog Design Assistant Tool

One of the contributions of this thesis is the generation of new circuit topologies to help analog designers discover new circuits and improve their designs. This is aided by accessing the newly generated circuit topologies in an efficient way. For this purpose, the *CM* of the generated circuits and corresponding small-signal transistor parameters are stored in a database. A Graphical User Interface, GUI, is developed to access the database and serve as for ADA's user interface that allows designers find and design new circuit topologies.

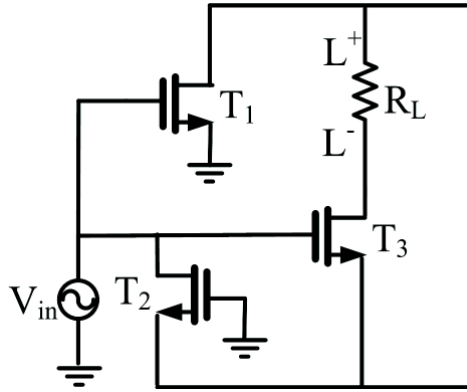
The user is prompted to enter circuit performance specification in "Required Specifications" window show in Fig. 3.19.

Once the user has entered the desired specifications, the *CMs* of circuit topologies, which meet the requirements, are provided in another window, as demonstrated in Fig. 3.20, where one of the circuit's *CM* and small-signal performance specifications are shown.

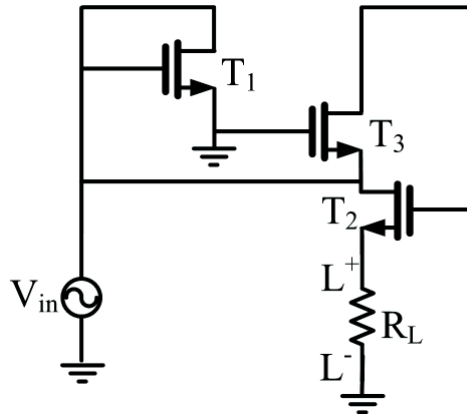
An added feature in ADA, is the ability to display the small-signal circuit of each circuit to help designers visualize the connections of the elements in the circuit. A small-signal diagram of one of the amplifiers is shown in Fig. 3.21.



(a) Amp 1

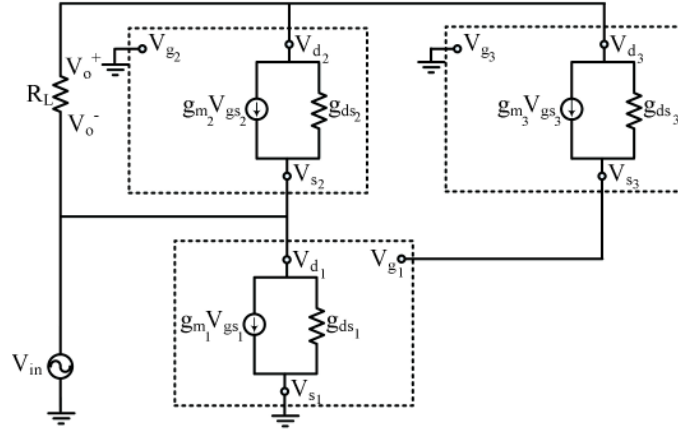


(b) Amp 2

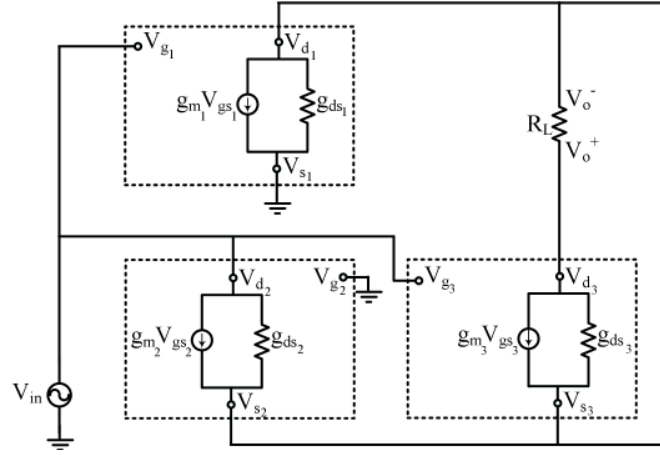


(c) Amp 3

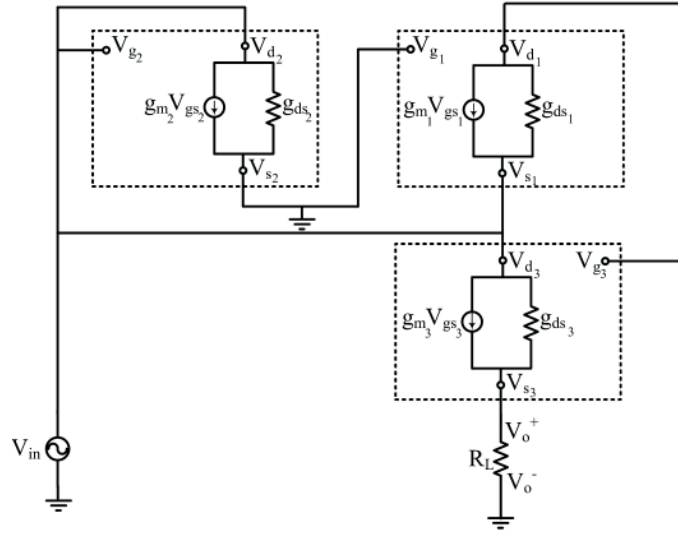
Figure 3.17: Examples of three previously unknown amplifiers with three transistors (biasing not shown).



(a) Amplifier 1



(b) Amplifier 2



(c) Amplifier 3

Figure 3.18: Small-signal model of the three amplifiers in Fig. 3.17.

Please enter the circuit specifications:

	From	To
Gain :	1	50
Reverse Gain:	0	0.01
Input Impedance:	40	100
Output Impedance:	40	600

Reset Continue

Figure 3.19: Specifications are entered in "Required Specifications" window.

Table 3.9: An example of required small-signal specifications.

Specifications	From	To
A_f	1	50
A_{rev}	0	0.01
$Z_{in}(\Omega)$	40	100
$Z_o(\Omega)$	40	600

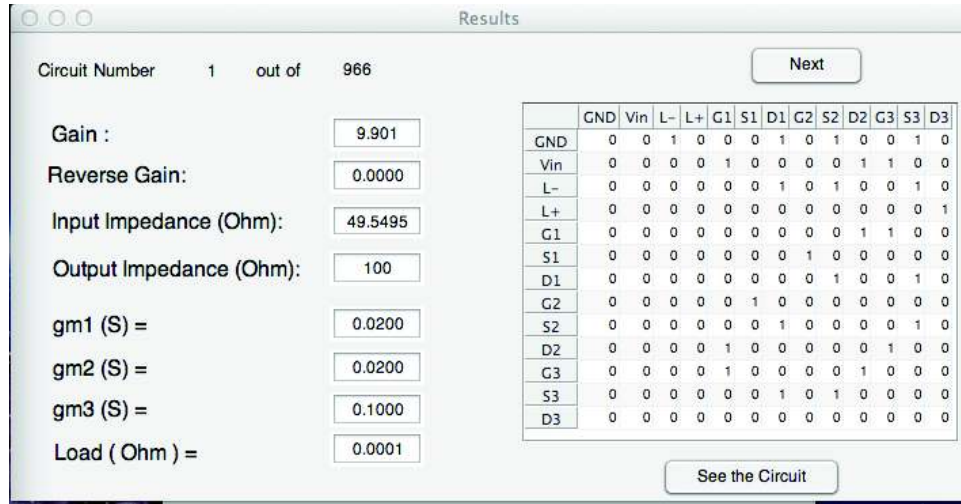


Figure 3.20: The provided topologies with their CM and small-signal specifications.

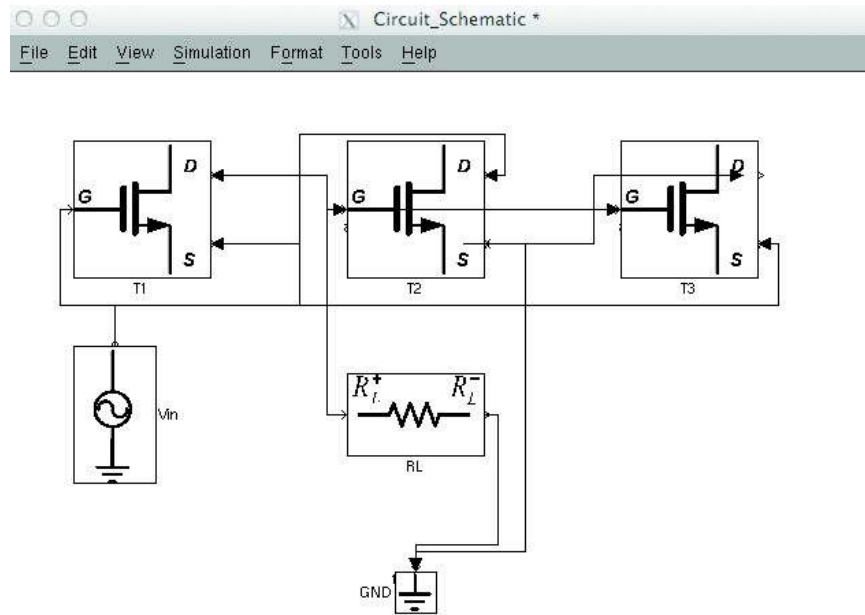


Figure 3.21: The small-signal diagram of a circuit generated by ADA.

3.8 Summary

In this chapter, an automatic methodology is described for generating all functional 2- and 3-transistor circuit topologies. In this methodology, a connectivity matrix is designed to represent analog circuits. The connectivity matrix does not require large memory for storing circuit topologies. Once all the functional circuit topologies are generated, their small-signal specifications are analyzed and optimized using the proposed optimization problem. In addition, the generated circuit topologies and their parameters also stored in a database. The number of generated circuit topologies with two and three transistors are 582, and 56,280, respectively. A GUI, which is called "Analog Design Assistant (ADA)" is developed to help analog designers find new topologies for their designs.

Chapter 4

Analysis and Design of Selected Amplifier

4.1 Introduction

In this chapter, the design of a new 3-transistor amplifier identified by ADA and achieving a high gain, 39 dB, is described. The small-signal model of the amplifier is generated by ADA as described in Chapter 3. From the several thousand topologies the ADA generated, this, previously unknown amplifier, is selected to demonstrate that the circuits generated by ADA are properly optimized. The circuit has been analyzed and designed in a 0.13- μm standard CMOS technology.

This chapter is organized as follows: In Section 4.2, the optimization problem in Chapter 3 is solved for the selected amplifier. In Section 4.3, the design procedure of the selected amplifier is introduced. In Section 4.4, the characteristics of the amplifier are analyzed. In Section 4.5, the Cadence simulation results are given, and compared with the ADA results. Finally, concluding remarks are given in Section 4.6.

4.2 Optimization of Small-Signal Transistors' Parameters

In this section, the optimization of the small-signal transistor parameters, g_m and R_L , of the selected 3-transistor amplifier, Amp 2 shown in Fig. 3.17, is discussed. The topology of Amp 2, and its small-signal model are repeated from Figs. 3.17 and 3.18 in Fig. 4.1. The topology, shown in Fig. 4.1.(a), is generated by ADA and does not include circuitry needed for DC biasing. In the small-signal model of Amp 2, shown in Fig. 4.1.(b).

Since, the Amp 2 is designed in a 0.13- μm standard CMOS technology, a transconductance of 100mA/V used in optimizations in Chapter 3 is not practical. Because of that the

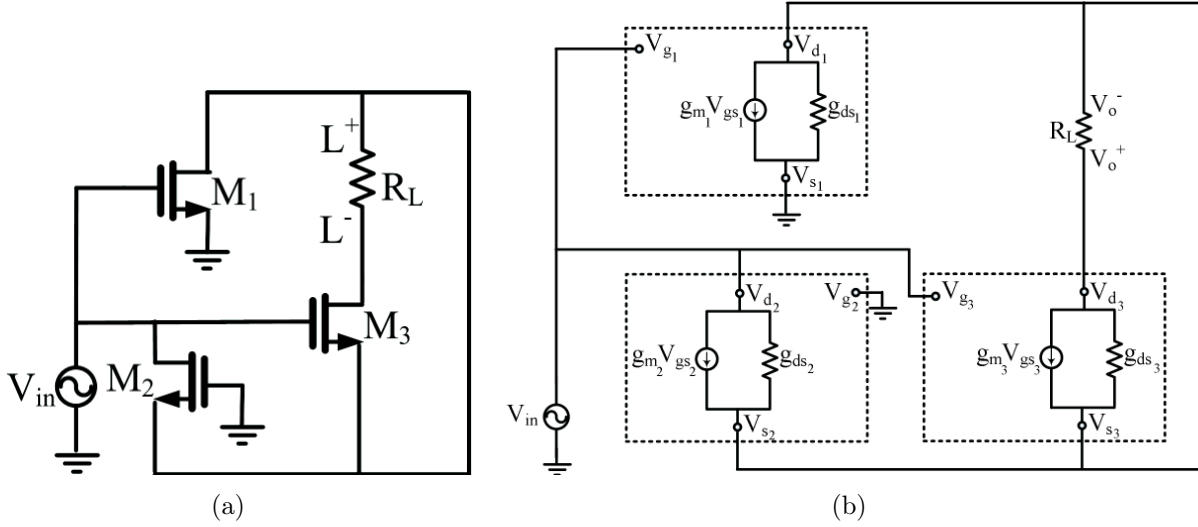


Figure 4.1: The topology and its small signal model of selected amplifier, Amp 2, generated by ADA(biasing not shown).

upper bound of g_{m_u} in (3.25) is set to 0.055 A/V, and g_{m_l} is 0.001 A/V. The optimization problem is then:

$$\begin{aligned}
 \max_{g_m, R_L} \quad & A_f(g_m, R_L) \\
 \text{s.t.} \quad & 25 \leq R_{in} \leq 100 \\
 & 0.001 \leq g_m \leq 0.055 \\
 & g_{ds} = 0.1 \times g_m \\
 & A_{rev} < \frac{1}{A_f} \\
 & 1 \leq R_L \leq 10,000.
 \end{aligned} \tag{4.1}$$

After solving (4.1), the optimal values of the DC voltage gain, A_f , reverse gain, A_{rev} , input and output impedance, R_{in} , R_o , of the amplifier are obtained and shown in Table 4.1. Note that these values are different from the values given in Table. 3.8. This is because the upper and lower bound of small-signal transistor parameters are slightly different. As shown in Table 4.1, in order to have maximum gain, the input impedance should be at the upper

bound of 100Ω . If the R_{in} decreases the optimal value of the DC voltage gain decreases too. The optimal value for small-signal transistor parameters are shown in Table. 4.1. As shown in this table, g_{m_1} and g_{m_3} should be at their upper bound, and g_{m_2} should be at the lower bound to have the maximum DC voltage gain.

Table 4.1: Optimal values of DC voltage gain, reverse gain, input and output impedance, small-signal transistor parameters for Amp 2 designed in a $0.13\text{-}\mu\text{m}$ standard CMOS technology

Specification	Optimal Value	Transistors' parameter	Optimal Value
A_f	90.1 (39 dB)	$g_{m_1}(A/V)$	0.055
A_{rev}	0	$g_{m_2}(A/V)$	0.001
$R_{in}(\Omega)$	100	$g_{m_3}(A/V)$	0.055
$R_o(\Omega)$	181	$R_L(\Omega)$	10K

4.3 Circuit Design

In this section, the proposed design procedure of the amplifier is presented. The first step is to ensure that all transistors are in saturation and have g_{m_s} as optimized by ADA. Since the gates of transistors M_1 and M_3 are connected and the drain of M_1 is connected to the source of M_3 , keeping transistors M_1 and M_3 in saturation region is dependent on the each others terminals voltage. To keep M_1 in saturation, the following relation between the voltage of terminals is required:

$$V_{DS_1} \geq V_{GS_1} - V_{TH}, \quad (4.2)$$

where V_{TH} is threshold voltage for transistors. Based on (4.2), V_{G_1} should be less than or equal to $V_{D_1} + V_{TH}$. In addition, to keep transistor M_3 on, the following equation should be considered:

$$V_{G_3} - V_{S_3} \geq V_{TH}. \quad (4.3)$$

Since $V_{G_3} = V_{G_1}$, and $V_{S_3} = V_{D_1}$, (4.3) is converted to:

$$V_{G_1} \geq V_{D_1} + V_{TH}. \quad (4.4)$$

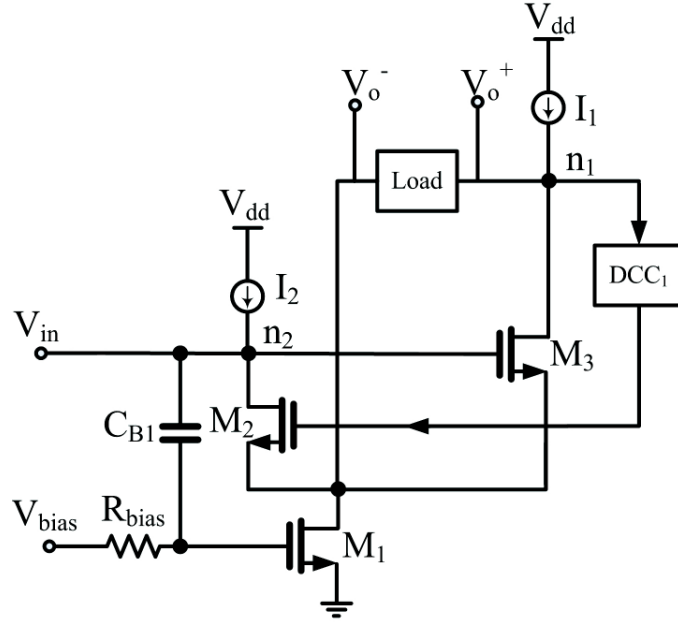


Figure 4.2: Schematic of the amplifier with ideal current sources.

Based on (4.2) and (4.4), boundaries for V_{G1} are found:

$$V_{D1} + V_{TH} \leq V_{G1} \leq V_{D1} + V_{TH}. \quad (4.5)$$

Therefore, V_{G1} becomes equal to $V_{D1} + V_{TH}$, where the transistors M_1 and M_3 may enter triode region. To assure that all the transistors are in saturation region, a DC blocking capacitor, C_{B1} , is used to separate the DC biasing of M_1 and M_3 as shown in Fig. 4.2. In order to bias the transistors M_1 , M_2 , and M_3 , two current sources, I_1 and I_2 , are used, which are described in the following section. A DC controller is employed to design the Amp 2, as shown in Fig. 4.2, which is discussed in Section 4.3.2.

4.3.1 Current Sources

The next step in is to replace the ideal current sources, I_1 , and I_2 , in Fig. 4.2 with their non-ideal counterparts. In order to do this several existing non-ideal current sources were studied [8, 38]. The selection of the current sources are normally based on the output resistance,

voltage headroom, and noise. The effects of them are discussed as follows:

- Output resistance: The output impedance of an ideal current source is infinity. Therefore, if the output resistance of a non-ideal current source is not high enough, the voltage gain of the amplifier may be reduced.
- Voltage headroom: The voltage headroom of the current source limits the output voltage swing. However, some voltage drop should be considered across the current source to keep its transistors in saturation region.
- Noise: The current source adds noise to circuit.

The voltage headroom and noise are not considered as very important in this work. This because the Amp 2 is optimized by ADA in terms of maximum gain. In this work ADA was not set up to optimize circuits for noise and output swing and therefore those performance metrics are not the driving factors in this design.

The output impedances of nodes n_1 and n_2 , where the ideal current sources, I_1 , I_2 , are connected to are important for this design. The output resistance seen at node n_1 , R_{on_1} , is obtained as follows:

$$R_{on_1} = \frac{1}{G_L + g_{ds_3}}, \quad (4.6)$$

where G_L is $\frac{1}{R_L}$. The output resistance seen at node n_2 , R_{on_2} , is calculated as follows:

$$R_{on_2} = R_s \parallel \frac{g_{ds_1} + g_{ds_2} + g_{m_2}}{g_{ds_1}g_{ds_2} + g_{ds_2}g_{m_1} + g_{m_1}g_{m_2}}, \quad (4.7)$$

where R_s is the resistance of input supply, and \parallel indicated that resistors are in parallel. As $g_{ds_i} = 0.1g_{m_i}$ ($i = 1, 2, 3$) in (4.1), it can be assumed that $g_{ds_i} \ll g_{m_i}$ ($i = 1, 2, 3$), resulting in:

$$R_{on_2} = R_s \parallel \left(\frac{g_{ds_1}}{g_{m_1}} + \frac{1}{g_{m_1}} \right) = R_s \parallel \frac{1}{g_{m_1}} \quad (4.8)$$

Since the output resistance of n_2 is not very high, the current source I_2 is implemented by a small pMOS diode-connected transistor. The impedance of node n_1 is high and the current source I_1 should have a high output impedance. As a result, a low-voltage cascode current source, as shown in Fig. 4.3, is employed[8, 38]. This current source has a high output impedance, R_{ocs} , which is given as follows:

$$R_{ocs} = \frac{g_{ds_{Mb3}} + g_{ds_{Mb1}} + g_{m_{Mb1}}}{g_{ds_{Mb3}}g_{ds_{Mb1}}}, \quad (4.9)$$

where, $g_{ds_{Mb1}}$, $g_{ds_{Mb3}}$ are output conductance of M_{b1} , and M_{b3} , respectively. The $g_{m_{Mb1}}$ is the transconductance of transistor M_{b1} .

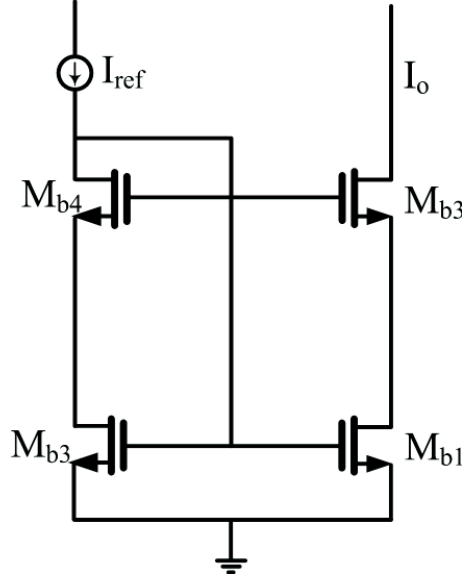


Figure 4.3: A low headroom cascode current source.

The output current of the current source, I_1 , can be tuned by changing the sizes of its transistors, M_{b1} , and M_{b1} , using the following procedure. The drain current of M_{b1} and M_{b3} are calculated as follows:

$$I_{M_{b3}} = I_{ref} = \frac{1}{2}\mu_n C_{ox} \left(\frac{W}{L}\right)_{Mb3} (V_{GS_{Mb3}} - V_{TH})^2 (1 + \lambda V_{DS_{Mb3}}), \quad (4.10)$$

where μ_n is the charge-carrier effective mobility, C_{ox} is the gate oxide capacitance per unit area, and the W and L are the gate width, and gate length of a transistor, respectively.

Similarity,

$$I_{Mb1} = I_1 = \frac{1}{2}\mu_n C_{ox} \left(\frac{W}{L}\right)_{Mb1} (V_{GS_{Mb1}} - V_{TH})^2 (1 + \lambda V_{DS_{Mb1}}). \quad (4.11)$$

Since the gates of M_{b1} and M_{b3} are connected to each other, and their sources are connected to ground, the ratio of output current I_1 to I_{ref} is,

$$\frac{I_1}{I_{ref}} = \frac{\left(\frac{W}{L}\right)_{Mb1} (1 + \lambda V_{DS_{Mb1}})}{\left(\frac{W}{L}\right)_{Mb3} (1 + \lambda V_{DS_{Mb3}})}. \quad (4.12)$$

Therefore, I_2 can be tuned by changing the sizes of transistors M_{b1} and M_{b3} .

4.3.2 DC controller

A sensitive node in an analog circuit refers to a node, where a small change in the input voltage may result in having large voltage changes in that node. The voltage changes may cause some transistors connected to that node to go into triode. In order to avoid triode, a DC controller network (DCC) is added to sense the node voltage and adjust the bias the transistors. In Amp 2, node n_1 is the only node prone to sensitive node. Therefore, a DCC, DDC_1 , is added between n_1 and gate of M_2 , as shown in Fig. 4.4. This circuit is made of three resistors, R_{D1} , R_{D2} , R_{D3} , and a transistor, M_{DCC1} .

With the DCC_1 shown in Fig. 4.4, if the drain voltage of transistor M_3 increases, the current of transistor M_{DCC1} decreases and the drain voltage of M_{DCC1} increases also increasing the gate voltage of transistor M_2 . By increasing the gate voltage of transistor M_2 , the source voltage of M_2 decreases since the current through M_2 is set by I_2 . This decreases the source voltage of M_3 and increase its current thus driving the drain voltage of M_3 lower and counteracting the initial increase of the drain voltage of M_3 . Therefore, the drain voltage of M_3 will remain constant and transistors of I_1 and M_3 avoid triode.

In order to have the maximum power delivery, the input impedance of the circuit, R_{in} , should be conjugately matched to the signal-source impedance, R_s . signal-source impedance in most of the application is 50Ω .

4.4.2 Noise Analysis

If the input signal, V_{in} , is weak, the noise becomes an important parameter limiting the ability of the circuitry to detect the input signal [39]. Since noise waveforms are random, the average power of noise is generally considered, where the average noise power in a noisy signal $x(t)$ is obtained by:

$$P_{av} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x^2(t) dt, \quad (4.15)$$

where T is the period of signal $x(t)$ [8]. In addition, the average noise power in frequency domain is represented by power spectral density $S_x(f)$, where f is the center frequency. Based on (4.15), the unit of measurement for average power in frequency domain is V^2/Hz [8].

To determine the amount of noise added by an analog circuit to the input signal, the noise factor of the circuit is considered. Noise factor, F , is the ratio of the signal-to-noise ratio (SNR) at the input, SNR_{input} , to the SNR at the output, SNR_{output} :

$$F = \frac{SNR_{input}}{SNR_{output}}, \quad (4.16)$$

where the SNR_{input} is:

$$SNR_{input} = \frac{\text{power of input signal}}{\text{power of noise from input source}}, \quad (4.17)$$

and the SNR_{output} is:

$$SNR_{output} = \frac{\text{power of output signal}}{\text{power of total output noise}}. \quad (4.18)$$

Noise factor, F , can also be calculated as:

$$F = \frac{\text{power of total output noise}}{\text{power of noise due to input}}. \quad (4.19)$$

In this work (4.19) is used to analyze the noise factor of the amplifier.

Noise figure, NF , is the noise factor in dB's:

$$NF = 10 \log F = 10 \log \left(\frac{SNR_{input}}{SNR_{output}} \right). \quad (4.20)$$

In order to analyze the noise of a circuit, all dominant noise sources should be considered. There are two important types of noise in an analog circuit: thermal noise, and flicker noise.

Thermal Noise

The thermal noise is the most common noise, which is caused by the fluctuations in the voltage measured across a conductor due to the random motions of electrons. This noise exists even when there is no current flowing through the conductor. The thermal noise is explained first in 1926 by John B. Johnson [40]. The thermal noise of a resistor can be modelled as a noiseless resistor in series with a voltage source, or as a noiseless resistor in parallel with a current source as shown in Fig. 4.5. The mean square noise voltage and the mean square noise current are given by:

$$\bar{V}^2 = 4kTBR, \quad (4.21)$$

$$\bar{I}^2 = \frac{4kTB}{R}, \quad (4.22)$$

where K is the Boltzmann constant ($K = 1.38 \times 10^{-23} \frac{\text{Joules}}{\text{Kelvin}}$), R is the value of resistance, and T is the temperature in Kelvin, B is the measurement bandwidth in Hz . Therefore, based on (4.15) the $S_v(f)$ is equal to $4KTR$.

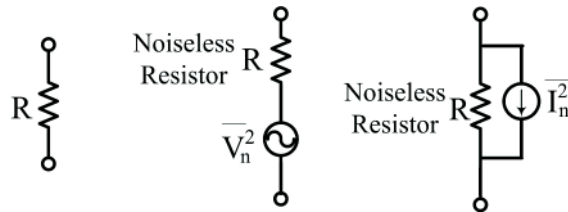


Figure 4.5: Thermal noise of a resistor is modelled as a noiseless resistor with voltage source or current source.

Flicker Noise

Flicker noise or $1/f$ noise is another type of noise in analog circuits, which occurs at the interface between the gate oxide and silicon substrate in a MOSFET. Some of the charged carriers are trapped at the interface, and when they are released the flicker noise results. The mean square noise voltage for bandwidth of 1 Hertz is obtained as:

$$\bar{V}^2 = \frac{K}{C_{ox}WLf}, \quad (4.23)$$

where K is a process-dependent constant ($10\text{-}25 \text{ V}\sqrt{F}$), C_{ox} is the gate-oxide capacitance per unit area, W and L are the width and length of the MOSFET. As seen in (4.23), frequency is an important parameter in flicker noise.

Noise in MOSFETs

The thermal noise dominates MOSFET noise at high frequencies. The drain noise is the most important noise source in a MOSFET. Noise in the drain of a saturated MOSFET is modelled by a current source, which is connected between the drain and source terminals as shown in Fig.4.6. This noise consists of channel noise and flicker noise. Since flicker noise at high frequencies is negligible, in high-frequency analog circuits flicker noise is not considered. Therefore, the dominant noise source in drain of transistor is the channel noise. The mean square noise drain current is expressed as follows:

$$\bar{i}^2 = 4kT\gamma g_m, \quad (4.24)$$

where coefficient γ is around $2/3$ for long channel transistor.

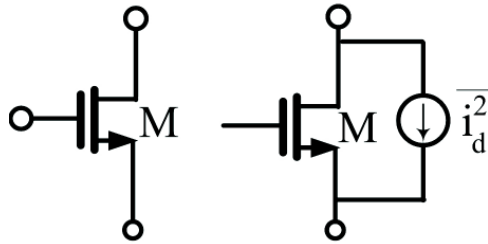


Figure 4.6: The noise generated by the drain of a transistor.

Amp 2 Noise Calculations

If the noise sources in a circuit are uncorrelated, the total noise figure is obtained by:

$$F = \frac{\text{power of total output noise}}{\text{power of noise due to input}} = 1 + \sum_{i=1}^{n_{ns}} \frac{\text{power of noise from } i^{\text{th}} \text{ source}}{\text{power of noise due to input}}, \quad (4.25)$$

where n_{ns} is to total number of noise sources in a circuit.

The noise analysis of this circuit can be carried out by assuming that the channel thermal noises of the MOSFETs are dominant with the power spectral densities of $4kT\gamma g_m$ [41]. The noise contributions of transistors M_1 , M_2 , and M_3 , identified as F_{M_1} , F_{M_2} , and F_{M_3} , are given as follows:

$$F_{M_1} = \left(\frac{G_s - g_{m_2}}{g_{m_1} + g_{m_2}} \right)^2 \frac{\overline{i_{n_1}^2}}{\overline{i_{R_s}^2}}, \quad (4.26)$$

$$F_{M_2} = \left(\frac{G_s + g_{m_1}}{g_{m_1} + g_{m_2}} \right)^2 \frac{\overline{i_{n_2}^2}}{\overline{i_{R_s}^2}}, \quad (4.27)$$

$$F_{M_3} = \left(\frac{G_s (g_{ds_1} + g_{m_2}) + g_{m_1} g_{m_2}}{(g_{m_1} + g_{m_2}) g_{m_3}} \right)^2 \frac{\overline{i_{n_3}^2}}{\overline{i_{R_s}^2}}, \quad (4.28)$$

where, R_s is 50Ω , $G_s = 1/R_s$, $\overline{i_{n_i}^2} = 4kT\gamma g_{m_i}$ is the channel noise of M_i , and it is assumed that $g_{ds_i} \ll g_{m_i}$ ($i = 1, 2, 3$).

Since the noise sources of different transistors are uncorrelated, the total noise factor of the amplifier from (4.25) is given by:

$$F = 1 + F_{M_1} + F_{M_2} + F_{M_3}. \quad (4.29)$$

The noise factor in (4.29) can be simplified by using (4.14), and $\gamma g_{m_3} R_L \gg 1$ and considering the input impedance matching condition, i.e. $R_{in} = R_s$, to

$$F \approx 1 + \frac{1 + R_s^2 g_{m_1} g_{m_2}}{(g_{m_1} + g_{m_2}) R_s} \gamma + \left[\frac{2(g_{ds_1} + g_{m_2})}{g_{m_1} + g_{m_2}} \right]^2 \frac{\gamma}{R_s g_{m_3}}. \quad (4.30)$$

Equation (4.30) provides design insight for sizing MOSFETs for reducing the amplifier noise and shows that the noise factor is decreased by decreasing g_{m_2} and increasing g_{m_3} and

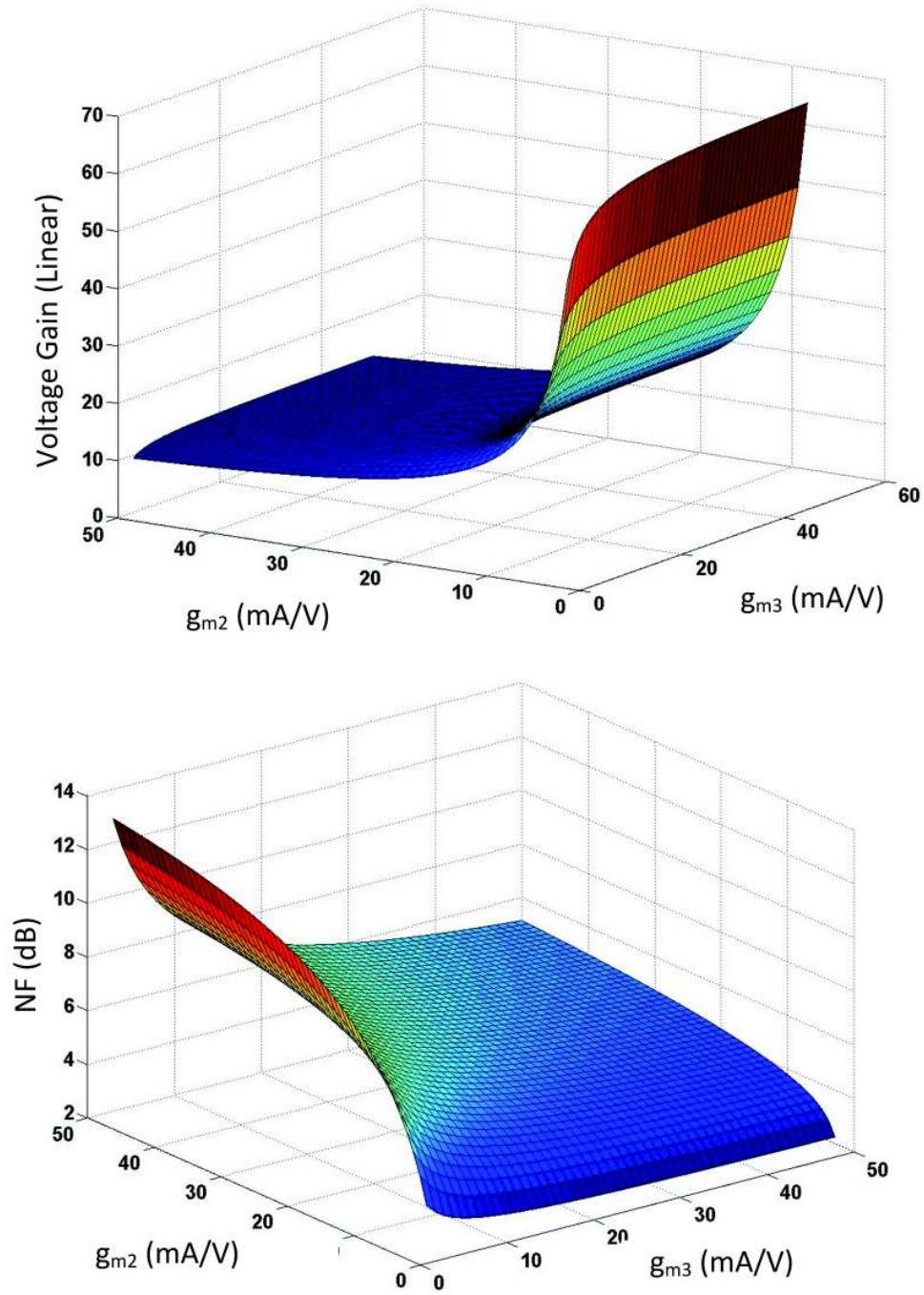


Figure 4.7: Voltage gain and NF versus g_{m2} and g_{m3} values for $\gamma=1.33$ and $R_{in} = R_s$.

Table 4.2: The comparison of ADA and Cadence results

	ADA	Cadence
$A_f(dB)$	39.1	39.4
A_{rev}	0	<0.01
$R_{in}(\Omega)$	100	90
GBW (GHz)	-	50
NF (dB)	-	2.4(Min)
V_{dd} (V)	-	1.8
I_d (mA)	-	3.8
Technology	0.13 μm CMOS	

g_{m_1} . The increase in g_{m_1} is, however, restricted by the the input matching constraint stated in (4.14).

In Fig. 4.7 the relationship between noise factor and g_{m_2} and g_{m_3} and voltage gain and g_{m_2} and g_{m_3} are illustrated. From Fig. 4.7, it can be seen that using the obtained value of g_{m_2} and g_{m_3} from ADA results in maximum voltage gain and minimum NF.

4.5 Simulation Result

In order to demonstrate the accuracy of ADA, we compare the results obtained from ADA and those from Cadence in Table 4.2. In Table 4.3 the comparison between the optimal value for g_m and g_{ds} for transistors M_1, M_2, M_3 obtained by ADA tool and Cadence are shown.

The difference between Cadence simulation results and ADA results in Table 4.2, and the difference between obtained g_m and g_{ds} from ADA and Cadence simulation in Table 4.3 are primarily due to g_{ds} being fixed to $0.1g_m$ in ADA optimizations, while g_{ds} is not exactly following this relation in Cadence BSIM simulations.

Moreover, the size of transistors M_1, M_2, M_3 , and transistors employed in current sources, and DCC are shown in Table 4.4, where the I_{ref} is $3.8mA$. The amplifier consumes 6.8 mW from a 1.8-V supply

In Fig. 4.8, the voltage gain and NF of the amplifier are shown. As shown in Fig. 4.8, the DC voltage gain is 39.4 and NF of 2.4 dB is achieved. The gain-bandwidth product of

Table 4.3: The comparison of the value of g_m and g_{ds} for M_1 , M_2 , and M_3 obtained from ADA and Cadence simulation

Transistor parameter	Results from ADA	Results from Cadence
$g_{m_1}(mA/V)$	55	39
$g_{m_2}(mA/V)$	1	3.2
$g_{m_3}(mA/V)$	55	44.5
$g_{ds_1}(mS)$	5.5	3.3
$g_{ds_2}(mS)$	0.1	0.2
$g_{ds_3}(mS)$	5.5	1.4

Table 4.4: The size of transistors in final schematic of Amp2

Transistor parameter	Value
W/L_1	$85\mu m/0.13\mu m$
W/L_2	$5\mu m/0.13\mu m$
W/L_3	$125\mu m/0.13\mu m$
$W/L_{b_{1-4}}$	$90\mu m/0.13\mu m$
W/L_{b_5}	$5\mu m/0.13\mu m$
W/L_{DCC_1}	$85\mu m/0.13\mu m$

the amplifier is 50 GHz. In Fig. 4.9, the input impedance and reverse gain of the amplifier are shown. The reverse gain is smaller than 0.01 within the entire bandwidth, and the input impedance stays lower than 90 Ω .

4.6 Summary

In this chapter, a new amplifier with a high gain is designed in a 0.13- μm CMOS technology. Detailed design descriptions and noise analysis of the amplifier are given. The simulation results show an input power matched amplifier with a noise figure of less than 3.1 dB and a voltage gain of 39 dB with a gain-bandwidth product of 50 GHz. The amplifier consumes 6.8 mW from a 1.8-V supply. This design illustrates the use of ADA to generate and optimize a previously unknown circuit.

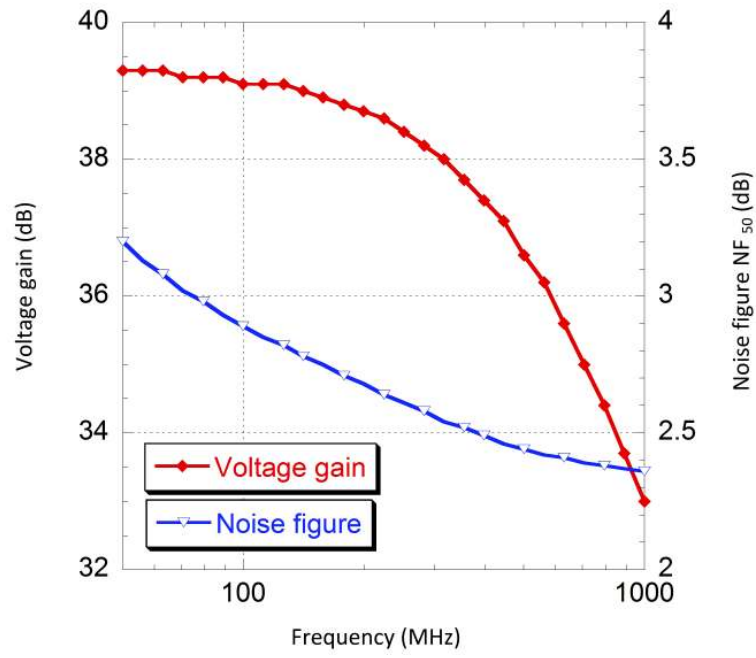


Figure 4.8: The voltage gain and the NF of Amp 2.

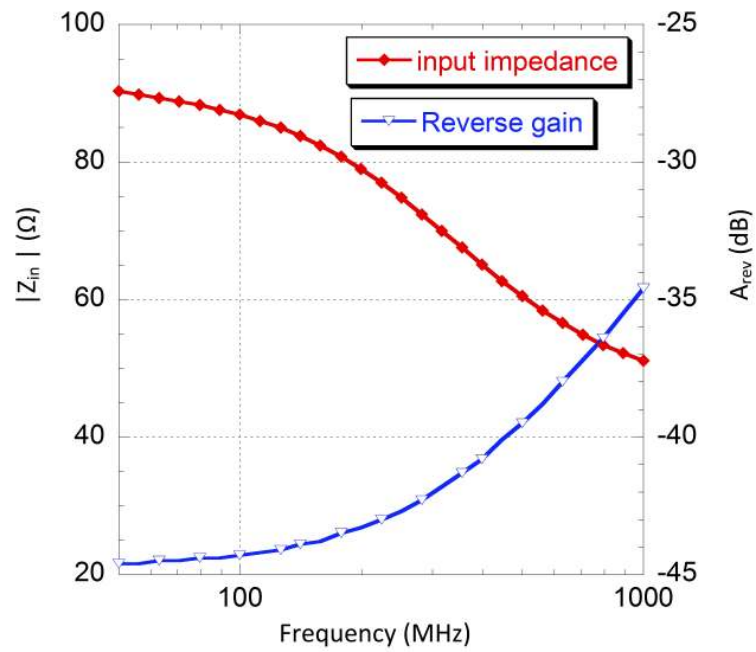


Figure 4.9: The input impedance and the reverse gain of the amplifier.

Chapter 5

Conclusions and Further Directions

5.1 Conclusions

In this thesis, development of an analog design assistant, ADA, tool is discussed. In addition, an amplifier with gain of 39 dB is designed. To the best knowledge of the author, this amplifier is previously unknown.

The methodology to generate all possible functional topologies for 2-transistor and 3-transistor circuits is proposed in Chapter 3. This algorithm can help analog designers find new topologies based on their required specifications. In addition, it is proven that ADA generates all possible 2-transistor topologies without any duplications using combinatorics techniques. In order to create all possible circuits, a matrix representation is employed, which requires low memory size for storing.

The small-signal specifications of all generated analog circuits are calculated symbolically and numerically. Then, all generated circuits with their small-signal models are stored in a database, which can be accessed via a custom developed GUI. The GUI is a front-end to the analog design assistant (ADA) tool, which performs circuit generation, optimization and helps with the selection of suitable circuit topologies.

In Chapter 4, a new high-gain input power-matched amplifier, which is generated and optimized by ADA, is introduced. The amplifier is designed in a 0.13- μm standard CMOS technology. Cadence simulation results are compared with ADA's results to demonstrate the capability of ADA to assist analog designer.

5.2 Contributions

In summary, the main contributions of this thesis are:

- Development of a computerized procedure of generating all possible and functional small-signal circuits with two and three transistors.
- Proof of uniqueness and completeness of the generated topologies.
- Construction of a database to store all generated circuits with their specifications.
- Creation of a GUI to help the designers search through the generated circuits.
- Implementation and design of a previously unknown high-gain amplifier.

5.3 Further Directions

In this thesis, a tool was developed to generate all possible configurations with two and three transistors. This tool can be extended to generate circuits with more transistors or elements such as resistors, and therefore, it can be employed to construct more complicated circuits. Furthermore, the computerized implementation of the algorithm is amendable to automated calculations of circuit performance characteristics. In this work only gain and input and output impedance were considered but including other characteristics such as noise, and bandwidth can be incorporated.

In this thesis, finding a new amplifier was the main goal. Therefore, an optimization problem to find the maximum DC voltage gain was defined. Circuit optimizations are not limited to only gain and optimization problems can be defined to optimize circuits for other applications.

In chapter 4, one of the previously unknown amplifiers was designed. However, there are more previously unknown amplifiers that are generated by the proposed algorithm. Analysis of these could lead to some new interesting circuits.

Bibliography

- [1] G. E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8):1825–1854, Apr. 1965.
- [2] C. M. Huang, C. M. Wu, C. C. Yang, and C. L. Wey. PrSoC: Programmable system-on-chip (SoC) for silicon prototyping. In *proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 58, pp. 3382–3385, May 2008.
- [3] V. Lahtinen, K. Kuusilinna, and T. Hamalainen. Optimizing finite state machines for system-on-chip communication. In *proceedings of IEEE International Symposium on Circuits and Systems ISCAS*, vol. 1, pp. 485–488, May 2002.
- [4] G. Gielen, R. Rutenbar, S. Borkar, R. Brodersen, J. H. Chern, E. Naviasky, D. Saias, and C. Sodini. Tomorrow’s analog: just dead or just different? In *43rd ACM/IEEE Design Automation Conference (DAC)*, pp. 709–710, June 2006.
- [5] J.G. Tront, J.J. Whalen, C.E. Larson, and J.M. Roe. Computer-aided analysis of RFI effects in operational amplifiers. *IEEE Transactions on Electromagnetic Compatibility*, EMC-21:297–306, Nov. 1979.
- [6] G.G.E. Gielen and R.A. Rutenbar. Computer-aided design of analog and mixed-signal integrated circuits. *Proceedings of the IEEE*, 88(12):1825–1854, Dec. 2000.
- [7] E.A.M. Klumperink and A.J.M. van Tuijl. Systematic generation of transconductance based variable gain amplifier topologies. In *proceeding of IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 1, pp. 289–292, May 1998.
- [8] B. Razavi. *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, 2000.

- [9] T. Eeckelaert, R. Schoofs, G. Gielen, M. Steyaert, and W. Sansen. An efficient methodology for hierarchical synthesis of mixed-signal systems with fully integrated building block topology selection. In *proceeding of Design, Automation and Test in Europe Conference and Exhibition*, pp. 1–6, Apr. 2007.
- [10] M. Meng Zhang and Z. Li. A 2.4 GHz low power common-gate low noise amplifier for wireless sensor network applications. In *proceedings of 2011 IEEE 13th International Conference on Communication Technology (ICCT)*, pp. 619–622, Sept. 2011.
- [11] Cadence. <http://www.cadence.com//>.
- [12] Agilent ads. <http://www.agilent.com//>.
- [13] F. Javid, R. Iskander, and M. M. Louerat. Simulation-based hierarchical sizing and biasing of analog firm ips. In *proceedings of Behavioral Modeling and Simulation Workshop*, pp. 43–48, Sept. 2009.
- [14] F. Javid, R. Iskander, M. M. Louerat, and D. Dupuis. Analog circuits sizing using bipartite graphs. In *proceedings of IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1–4, Aug. 2011.
- [15] A. Das and R. Vemuri. A graph grammar based approach to automated multi-objective analog circuit design. In *proceedings of Design, Automation and Test in Europe Conference and Exhibition*, pp. 700–705, Apr. 2009.
- [16] J. C. Tiernan. An efficient search algorithm to find the elementary circuits of a graph. *Communications Magazine of the ACM*, 13(12):722–726, Dec. 1970.
- [17] Technical report.
- [18] P. Corbishley and D.G. Haigh. Rules for systematic synthesis of all-transistor analogue circuits by admittance matrix expansion. In *proceedings of IEEE International*

Symposium on Circuits and Systems (ISCAS), vol. 6, pp. 5966–5969, May 2005.

- [19] A. Pradhan and R. Vemuri. Regression based circuit matrix models for accurate performance estimation of analog circuits. In *proceedings of International Conference on Very Large Scale Integration*, pp. 48–53, Oct. 2007.
- [20] A. Mesquita, F.A. Salazar, and P.P. Canazio. Chromosome representation through adjacency matrix in evolutionary circuits synthesis. In *proceeding of NASA/DoD Conference on Evolvable Hardware*, pp. 102–109, June 2002.
- [21] C. W. Ho, A. Ruehli, and P. Brennan. The modified nodal approach to network analysis. *IEEE Transactions on Circuits and Systems*, 22(6):504–509, June 1975.
- [22] A Yildiz B. Modified nodal analysis-based determination of transfer functions for multi-inputs multi-outputs linear circuits. *Automatika Journal for Control, Measurement, Electronics, Computing and Communications*, 51:353–360, 2010.
- [23] T.Y. Zhou, H. Liu, D. Zhou, and T. Tarim. A fast analog circuit analysis algorithm for design modification and verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(2):308–313, Feb. 2011.
- [24] A. Nunez-Aldana and R. Vemuri. An analog performance estimator for improving the effectiveness of CMOS analog systems circuit synthesis. In *proceeding of Design, Automation and Test in Europe Conference and Exhibition*, pp. 406–411, Mar. 1999.
- [25] E.A.M. Klumperink. *Transconductance Based CMOS Circuits: Circuit Generation, Classification and Analysis*. PhD thesis, University of Twente, 1960.
- [26] E. A. M. Klumperink, F. Bruccoleri, and B. Nauta. Finding all elementary circuits exploiting transconductance. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 48(11):1039–1053, Nov. 2001.

- [27] F. Bruccoleri, E. A. M. Klumperink, and B. Nauta. Generating all 2-transistor circuits leads to new wide-band cmos lnas. In *Proceedings of 26th European Solid-State Circuits Conference*, pp. 316–319, Sept. 2000.
- [28] J. Schmitz. Systematic design approach of MOSFET based low noise amplifiers for wide-band RF receivers. In *proceeding of European Microwave Integrated Circuit Conference (EuMIC)*, pp. 1253–1256, Munich, Oct. 2007.
- [29] J.D. Lohn and S.P. Colombano. A circuit representation technique for automated circuit design. *IEEE Transactions on Evolutionary Computation*, 3(3):205–219, Sept 1999.
- [30] T. McConaghy, P. Palmers, M. Steyaert, and G.G.E. Gielen. Variation-aware structural synthesis of analog circuits via hierarchical building blocks and structural homotopy. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28(9):1281–1294, Sept. 2009.
- [31] D. Ashlock. *Evolutionary Computation for Modeling and Optimization*. Springer, 2006.
- [32] T.R. Dastidar, P.P. Chakrabarti, and P. Ray. A synthesis system for analog circuits based on evolutionary search and topological reuse. *IEEE Transactions on Evolutionary Computation*, 9(2):211–224, Apr. 2005.
- [33] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.
- [34] G. Alpaydin, S. Balkir, and G. Dundar. An evolutionary approach to automatic synthesis of high-performance analog integrated circuits. *IEEE Transactions on Evolutionary Computation*, 7(3):240–252, June 2003.
- [35] J. Riordan. *An Introduction to Combinatorial Analysis*. Dover Publications, 2002.
- [36] S. Boyd. *Convex Optimization*. Cambridge University Press, 2004.

- [37] A. Barvinok. In *A Course in convexity*. American Mathematical Society, 2002.
- [38] Johns D. A. Martin K. W. Carusone, T. Ch. *Analog Integrated Circuit Design*. Wiley Publication, 2nd edition, 2011.
- [39] Ph. E. Allen and D. R. Holberg. *CMOS Analog Circuit Design*. Oxford University Press, USA, 3rd edition, 2002.
- [40] J. B. Johnson. Thermal agitation of electricity in conductors. *Physical Review Letters*, 32:97–109, 1928.
- [41] A.J. Scholten, H.J. Tromp, L.F. Tiemeijer, R. V. Langevelde, R. J. Havens, P. W. H. De-Vreede, R. F. M. Roes, P.H. Woerlee, A.H. Montree, and D.B.M. Klaassen. Accurate thermal noise model for deep-submicron CMOS. In *International Electron Devices Meeting (IEDM)*, pp. 155 –158, Aug. 1999.