



Enhancing Analog Circuit Security Through Obfuscation

Venugopal Rao, Vaibhav

<https://researchdiscovery.drexel.edu/esploro/outputs/doctoral/Enhancing-Analog-Circuit-Security-Through-Obfuscation/991021899615304721/filesAndLinks?index=0>

Venugopal Rao, V. (2024). Enhancing Analog Circuit Security Through Obfuscation [Drexel University].
<https://doi.org/10.17918/00010486>
Document Version: PDF

Repository homepage: <https://researchdiscovery.drexel.edu/>
Contact: libsystems.drexel.edu
Open Access (License Unspecified)
Research:Open
Downloaded On 2025/10/12 00:40:58 -0400



Drexel UNIVERSITY GRADUATE THESIS APPROVAL FORM AND SIGNATURE PAGE

Instructions: This form must be completed by all doctoral students with a thesis requirement.
This form MUST be included as page 1 of your thesis via electronic submission to ProQuest.

Thesis Title: ENHANCING ANALOG CIRCUIT SECURITY
THROUGH OBFUSCATION

Author's Name: Vaibhav Venugopal Rao

Submission Date: 06/25/2024

The signatures below certify that this thesis is complete and approved by the Examining Committee.

Role: Chair Name: Baris Taskin
Title: Professor
Department: Electrical & Computer Engineering
Approved: Yes Date: 07/04/2024

Role: Member Name: Ioannis Savidis
Title: Associate Professor
Department: Electrical & Computer Engineering
Approved: Yes Date: 07/04/2024

Role: Member Name: Bahram Nabet
Title: Professor
Department: Electrical & Computer Engineering
Approved: Yes Date: 07/23/2024

Role: Member Name: Kapil Dandekar
Title: Professor
Department: Electrical & Computer Engineering
Approved: Yes Date: 07/08/2024

Role: Member Name: Goran Karapetrov
Title: Associate Professor
Department: Physics
Approved: Yes Date: 07/04/2024

ENHANCING ANALOG CIRCUIT SECURITY THROUGH OBFUSCATION

By:

Vaibhav Venugopal Rao

A Dissertation

Supervised By:

Professor Ioannis Savidis

Department of
Electrical and Computer Engineering



Drexel University
Philadelphia, Pennsylvania

2024

© Copyright by Vaibhav Venugopal Rao 2024
All Rights Reserved

Dedication

To my family, friends, and mentors, whose unwavering support and encouragement have been a constant source of strength and inspiration throughout this journey.

Acknowledgments

Acknowledging all individuals who have influenced, motivated, and molded me into the person I am today is a profoundly challenging task, as words often fall short in capturing the depth of my gratitude. This dissertation represents the culmination of years of dedication, perseverance, and support from an extensive network of family members, teachers, professors, and friends, each of whom has left an indelible mark on my academic and personal journey. It is with heartfelt appreciation and sincere humility that I attempt to express my gratitude to all who have walked alongside me on this path, providing unwavering encouragement, wisdom, and love.

I extend my deepest gratitude to my advisor, Professor Ioannis Savidis, for his unwavering support and exceptional guidance throughout my PhD studies. His mentorship has been instrumental not only in advancing my research but also in nurturing crucial life skills. Professor Savidis has imparted invaluable lessons in mastering effective presentation methods, honing the nuances of academic writing, and refining my verbal and communication skills. Moreover, his insights into building and leading teams have been invaluable, equipping me with the leadership abilities necessary for future success. His comprehensive approach to mentorship has profoundly shaped my academic and personal development, for which I am profoundly grateful.

I would also like to extend my heartfelt thanks to Professor Baris Taskin, Professor

Kapil Dandekar, Professor Bahram Nabet, and Professor Goran Karapetrov. Their insightful comments, critical reviews, and helpful advice have significantly enhanced both my research and this dissertation. Their expertise and constructive feedback have been instrumental in refining my work.

Special thank you to Dr. Tsz Kwok, my closest friend and guardian. You have been my first point of contact for any college-related questions and administrative matters, providing unwavering support throughout my journey. I am also profoundly thankful to Dr. Ragh Kuttappa for inspiring me to pursue my Ph.D., being an excellent roommate, and sharing six wonderful years with me. Your companionship has been a true pleasure and a cornerstone of my academic and personal life.

I would like to express my deepest gratitude to Dr. Sudhindra Gopalkrishna, my childhood friend. Our goals, education, and careers have closely aligned, and you have been with me through all the ups and downs. Your constant support in all my endeavors has been invaluable, and I am incredibly fortunate to have you by my side.

I am also grateful to my lab mate, Dr. Kyle Juretus, who has co-authored numerous papers with me and provided invaluable guidance throughout my Ph.D. journey. Your insights and support have been crucial to my success, and I deeply appreciate your collaboration and friendship. A heartfelt thanks goes to Dr. Shazzad Hossain, who has been like a big brother to me, offering guidance and support throughout my Ph.D. Your advice and encouragement have been instrumental in my development, both personally and professionally. In addition, I would like to thank my lab mates Jeff Wu, Ziyi Chen, Pratik Shrestha, Alec Avesta, and Ashish Sharma. It has been a pleasure working with you on designing our test chip. Your teamwork, dedication, and camaraderie have made this journey memorable and rewarding.

Finally, I want to thank my family, who have provided me with joy and happiness. I would especially like to thank my father, Dr. Venugopal Ramarao, and my mother, Dr. Varan Vasudeva Rao, for all their support and for raising me to be who I am. Their unwavering belief in me and their sacrifices have been the foundation of my achievements. My grandparents, Vasudeva Rao and Geetha, molded me into the person I am today, instilling moral values and inspiring my curiosity and love for science. I also thank my other grandparents, Ramarao and Saroja Bai, for their affection and for playing a crucial role in my upbringing. To my brother, (soon to be Dr.) Vaishnav Rao, your guidance, support, and encouragement throughout my life have been invaluable. I am grateful to my wife's parents, Arunachala and Padma, for their constant support during my studies. Most importantly, I thank my wife, Pavithra S.A., for her love, support, and endless encouragement. You have been my constant source of joy and strength.

Contents

Dedication	iii
Acknowledgments	iv
List of Tables	xiv
List of Figures	xix
Abstract	xxvii
1 Introduction	1
1.1 History of Analog Computing	3
1.1.1 Invention of the Transistor	5
1.1.2 Development of Integrated Circuits	7
1.2 Drivers for Development of Analog Integrated Circuits	7
1.2.1 Development of Analog MOS ICs	10
1.2.2 Development of BiCMOS RF ICs	13
1.3 Growth of Analog Computing	14
1.4 Challenges in Design and Development of Analog Circuits	17
1.5 Introduction to Analog Hardware Security	19

1.5.1	Security Vulnerabilities of a Horizontal IC Design Flow	20
1.5.2	Examples of Analog IP Piracy Attacks	21
1.6	Outline of the Dissertation	26
2	Analog Obfuscation Techniques	29
2.1	Split Manufacturing	30
2.1.1	Threat Model	31
2.1.2	Split Manufacturing Methodologies for Analog Circuits	32
2.1.3	Scenario I: Removal of the Top Metal Layer from FEOL Foundry . .	34
2.1.4	Scenario II: Removal of Top Two Metal Layers from FEOL Foundry	37
2.1.5	Scenario III: Insertion of Non-Functional Rings and Creation of Empty Zones	38
2.2	IC Camouflaging	41
2.2.1	Threat Model	42
2.2.2	Camouflaged Analog Components	43
2.2.3	Case Study	46
2.3	Key-based Locking	47
2.3.1	Key-based Locking Techniques to Protect Analog IP	50
2.4	Unique Key Generation and Storage Using Physically Unclonable Func- tions (PUFs)	57
2.4.1	Strong PUFs	58
2.4.2	Weak PUFs	60
3	Satisfiability Modulo Theory	61
3.1	Satisfiability Algorithm	63

3.1.1	Use of SAT Algorithms in VLSI	67
3.1.2	SAT Algorithms For Hardware Security	75
3.2	Satisfiability Modulo Theory (SMT) Algorithm	80
3.3	Use of SMT Algorithm For Analog Circuit Sizing	85
3.3.1	Analog Satisfiability (aSAT) for Design Space Exploration	87
3.3.2	Applying aSAT for Transistor Sizing of Circuit Parameters	93
3.3.3	Conclusions	100
4	Parameter Obfuscation Techniques	102
4.1	Threat Model	103
4.2	Parameter Obfuscation	104
4.2.1	Vector Based Parameter Obfuscation	104
4.2.2	Mesh Based Parameter Obfuscation	106
4.3	Challenges with Parameter Obfuscation	109
4.3.1	Selection of Transistor(s) to Obfuscate	113
4.3.2	Selection of Target Transistor Sizes	116
4.3.3	Selection of Obfuscation Transistor Sizes	116
4.4	Analog Satisfiability Algorithm for Transistor Sizing in Key-based Parameter Obfuscation	119
4.4.1	SMT Problem Formulation	120
4.4.2	SMT Algorithm	123
4.5	Circuit Implementation	125
4.5.1	Active Inductor Based Second-Order Band Pass Filter (BPF)	125
4.5.2	Operational Amplifier	128
4.6	Metric Evaluating the Security of Analog Obfuscation Techniques	132

4.7	Discussions	134
4.8	Conclusions	138
5	Simulation-based Bias Optimization Algorithm	140
5.1	Simulation Based Multivariate Bias Tuning	141
5.1.1	Considering Parameter Tuning for Variation when Implementing the Obfuscation Techniques	149
5.2	Conclusions	153
6	Security of Mixed Signal Circuits	154
6.1	Threat Model	156
6.2	Vulnerabilities of Independently Secured Analog and Digital Circuit Blocks	157
6.2.1	Proposed AMS Circuit Implementation	157
6.2.2	Implementation of the Obfuscation Techniques	162
6.2.3	Attacking AMS Blocks in Isolation	164
6.3	Adding Key Dependencies between Analog and Digital Circuit Blocks of an AMS Circuit	167
6.4	Securing AMS Systems	170
6.5	Conclusions	171
7	Security Vulnerabilities of Key-based Analog Obfuscation Techniques	173
7.1	Test Circuits	175
7.2	Key Spacing	176
7.3	Monotonic Circuit Response Attack	178
7.4	Design Insight	181
7.4.1	Equal Key Spacing	181

7.4.2	Creating Non-Monotonic Key Dependencies	182
7.5	Conclusions	183
8	Performance Analysis of Analog De-Obfuscation Attacks	185
8.1	Assumptions in Threat Models For Attack Techniques	188
8.1.1	Netlist Extraction	188
8.1.2	Extraction of PDK information	190
8.1.3	Extraction of bias voltages and currents	192
8.1.4	Generating Circuit Models	193
8.2	Circuit Implementations	195
8.2.1	Variable Gain Amplifier	196
8.2.2	Operational Amplifier	196
8.2.3	Mixer	197
8.2.4	Front-End	199
8.3	Security Evaluation of Analog Attack Algorithms	200
8.4	Performance of Different Analog Attack Methodologies	202
8.4.1	Equation-based SMT Attack	202
8.4.2	Genetic Algorithm-based Attack	206
8.4.3	Key Spacing	211
8.4.4	Monotonic Circuit Response Attack	212
8.5	Conclusions	215
9	Metrics For Evaluating Analog De-Obfuscation Attacks	217
9.1	Metrics of Evaluation	217
9.2	Resource Metric (T_R)	218

9.3	Performance Metric (T_P)	222
9.4	Attack Evaluation Metric (T)	225
9.5	Conclusions	227
10	DNA: DC Nodal Analysis Attack for Evaluation of Analog Obfuscation	
	Techniques	230
10.1	DC Nodal Analysis Attack	233
10.1.1	Application of DC Nodal Analysis Attack on Key-based Parameter Obfuscation	234
10.1.2	Application of DC Nodal Analysis Attack on Multi-threshold Ob- fuscation	237
10.2	Performance of DC Nodal Analysis Attack on Different Analog Obfusca- tion Techniques	239
10.2.1	Performance Evaluation For Key-based Parameter-Obfuscated Cir- cuits	242
10.2.2	Performance Evaluation on Multi-threshold Obfuscation	244
10.3	Performance Comparison of DC Nodal Analysis Attack With State-of-the- art Analog Deobfuscation Attacks	248
10.3.1	Resource Metric (T_R)	248
10.3.2	Performance Metric (T_P) Calculated for Key-based Obfuscation	250
10.3.3	Performance Metric (T_P) Calculated for Multi-threshold Obfuscation	257
10.4	Discussions	264
10.5	Conclusions	265
11	Concluding Remarks	268

12 Future Work	274
12.1 Security Metric Model for Analog Circuit	275
12.2 Security Integrated Analog Design Methodology	279
12.3 Developing Secure Memory For Key-based Analog Obfuscation	282
Bibliography	285

List of Tables

3.1	Equations of the circuit parameters of a two stage amplifier applied as input to the aSAT solver.	94
3.2	Equations of the circuit parameters of an LC tank VCO applied as input to the aSAT solver.	94
3.3	W/L ratios determined by executing the aSAT solver on a differential amplifier with a target gain of 40 dB, gain bandwidth of 5 MHz, slew rate of $5V/\mu m^2$, load capacitance of 10 pF, input common mode voltage range of 0.8 V to 1.6 V, and power dissipation of less than 100 μW	95
3.4	Analysis of gain A_v , gain bandwidth GBW, and power dissipation through SPICE simulation of a differential amplifier with transistor dimensions determined through aSAT.	96
3.5	Determined W/L ratio when applying aSAT to a two stage amplifier with target specifications of 60 dB gain, 30 MHz gain bandwidth, and power dissipation of less than 300 μW	97
3.6	Characterization of the gain A_v , GBW, phase margin, and power dissipation of a two-stage amplifier with transistor dimensions determined through aSAT analysis.	97

3.7	Results of applying aSAT to a VCO with a target frequency of 1.75 GHz. . .	99
3.8	Results from SPICE analysis of a VCO performed based on transistor dimensions determined through aSAT for multiple target frequencies.	100
4.1	Characterization of the output voltage for different key sequences applied to the voltage divider circuit shown in Fig. 4.5 obfuscated by the vector based and mesh based techniques.	113
4.2	Characterization of the center frequency, gain, bandwidth, and area of a BPF after implementing vector-based and mesh-based obfuscation. An unobfuscated BPF is also characterized for comparison.	128
4.3	Characterization of critical circuit parameters of an op-amp after applying the vector-based and mesh-based obfuscation techniques. The results are compared with an unobfuscated op-amp.	132
4.4	Analysis of the security provided by analog obfuscation techniques to an untrusted foundry and an untrusted end-user. The probability of determining the correct key is also evaluated.	133
4.5	Comparison of the total area (active and passive) of sub-blocks of an unobfuscated receiver and a receiver obfuscated with a 20-bit key.	136
5.1	Results of applying the simulation-based optimization algorithm that tunes the biasing conditions and body voltages of a BPF and op-amp to mitigate parasitic and process variations when applying vector-based and mesh-based obfuscation.	147

7.1	Results of the key spacing attack on a VCO, CS amplifier, differential amplifier, and operational amplifier obfuscated with a 10, 15, and 20-bit key for exclusion zone sizes of 1%, 5%, 10%, 15%, and 20%. The algorithm of the key spacing attack is characterized for the number of candidate keys and the time to determine the candidate solutions.	177
7.2	Results from executing a monotonic attack on a VCO, CS amplifier, differential amplifier, and operational amplifier all obfuscated with a 20-bit key. PVT variations of 1%, 5%, 10%, and 20% are analyzed. The algorithm implementing the monotonic attack is characterized for the number of candidate keys, execution time, and the number of iterations of the SMT solver.	179
8.1	Results of executing the equation-based SMT attack on the five obfuscated analog benchmark circuits described in Section III and for the four different scenarios described in Section IV. The “*” indicates that the SMT constraints are constructed accounting for gain and gain bandwidth parameters.	202
8.2	Results of executing the GA attack on the five obfuscated analog benchmark circuits described in Section III and for the four different scenarios described in Section IV.	206
8.3	Results of executing the key spacing attack on the five obfuscated analog benchmark circuits described in Section III and for the four different scenarios described in Section IV.	211

8.4	Results of executing the monotonic attack on the five obfuscated analog benchmark circuits described in Section III and for the four different scenarios described in Section IV.	213
9.1	Number of steps necessary for an adversary to determine needed circuit information to execute an attack. The number of steps required to complete extraction of the biasing voltages and currents, circuit attack model(s), and the bias information assumes that the attacker possesses the circuit netlist.	220
9.2	Required information needed to perform each attack on the five obfuscated analog circuits. A check mark implies needed information.	221
9.3	Total time required for each attack technique to obtain all the information necessary to de-obfuscate an analog circuit.	222
10.1	Results of executing the DC nodal analysis attack on four analog benchmark circuits obfuscated using the key-based parameter locking technique described in Section 8.2 and for three different scenarios described in Section 8.3.	240
10.2	Results of executing DC nodal analysis attack on four analog benchmark circuits masked using the multi-threshold obfuscation technique described in Section III and for the three different scenarios described in Section IV. .	245
10.3	Evaluation of the resource metric for the five analog deobfuscation attacks. A check mark implies needed information. The number of steps to determine all needed information is listed, where each step represents one unit of time.	249

- 10.4 Evaluation of the performance metric from the execution of the SMT, GA, key-spacing, monotonic response, and DC nodal analysis attacks on a VGA, 2-stage VGA, and front-end. The performance metric score is calculated as the sum of the execution time needed to determine the candidate keys and the total time needed to perform a brute-force attack on the returned candidate keys, which is computed as the product of the number of candidate keys and the time required to perform a transient simulation on a single candidate key. The “-” symbol indicates the failure of the deobfuscation algorithm in returning the correct key. 251
- 10.5 Evaluation of the performance metric from execution of the SMT, GA, key-spacing, monotonic response, and DC nodal analysis attacks on a VGA, 2-stage VGA, and front-end secured with multi-threshold obfuscation. The “*” symbol indicates that the equation-based SMT attack did not determine the threshold voltages of all the transistors of the circuit and the “-” symbol indicates the failure of the deobfuscation algorithm in returning the correct key. 257

List of Figures

1.1	A (a) photograph and (b) schematic representation of a K2-W operational amplifier [18].	4
1.2	The (a) first point contact transistor developed by Bardeen, Brattain, and Shockley at Bell Laboratories in 1947 [20, 21] and (b) the first integrated circuit developed by Jack Kilby at Texas Instruments in 1958 [21].	6
1.3	Evolution of monolithic analog integrated circuits observed from the family of Fairchild linear ICs produced in the 1960's, where a microphotograph is shown of the (a) first integrated circuit with four transistors, (b) first BJT operational amplifier μ A702 with nine transistors [32], and (c) the most successful μ A709 operational amplifier [33].	8
1.4	Timeline of key innovations and developments in analog, AMS, and RF integrated circuits [42].	12
1.5	Comparison of design flows depicting a a) stepwise reduction of degrees of freedom for digital circuits, and b) continuous reduction of degrees of freedom and overlap of design flow for analog circuits [81].	18
1.6	Vulnerabilities in the analog integrated circuit supply chain.	21

1.7	Microphotograph of the (a) packaged AD633 analog multiplier IC, (b) die of the original intended AD633 IC [88, 90], and (c) die of the fraudulent AD633, which on further investigation is the RC4200 from <i>Raytheon Semiconductor</i> [91].	22
1.8	Microphotograph of the (a) packaged OPA627 IC bought online, (b) die of the fraudulent OPA627, and (c) die of the authentic TI OPA627 IC [88, 90], which on further investigation is the AD744 from <i>Analog Devices</i> [93]. . .	23
2.1	Partitioning of an IC for split manufacturing, where a) the cross-sectional view and b) the top view are shown [102].	31
2.2	Schematics of a a) Class AB amplifier and b) Class E power amplifier [101].	33
2.3	Circuit schematic with missing inductor information due to removal of metal 6 for a a) Class AB amplifier, and b) Class E power amplifier [101]. .	35
2.4	Effect of supply voltage and gate biasing on a) output power, and b) power-added efficiency of an AB amplifier [101].	36
2.5	Effect of gate biasing and frequency on a) output power, and b) power-added efficiency of an AB amplifier [101].	36
2.6	Schematics of a a) Class AB amplifier, and b) Class E power amplifier with missing capacitor and inductor information due to the removal of metals 5 and 6 resulting information [101].	39
2.7	Effect of sweeping the inductor (RF choke) and output coupling capacitor on the a) output power and b) power-added efficiency of an AB amplifier [101].	39

2.8	IC camouflaging of a gate with dummy contacts, where a depiction is provided of a) the top view of standard NAND and NOR gates and camouflaged NAND and NOR gates, and b) the cross-section of dummy and true contacts [104].	42
2.9	Depiction of camouflaged multiple gate-finger transistor schematics and layouts, where a) two fake contacts are utilized to deactivate two gate fingers and drain contacts and b) fake contacts are used in the layout of a common-centroid differential transistor pair to deactivate transistors. Diffusion, polysilicon, and metal are shown in green, red, and blue, respectively. True contacts are depicted in black and fake contacts in orange [105].	44
2.10	The layout of the camouflaged capacitor and resistor using fake contacts. a) The obfuscated capacitor on the right has fake contacts connecting its capacitor plates and obfuscating the overall capacitance of the capacitor bank [105]. b) The obfuscated serpentine resistor layout. Poly, metal 1, metal 2, true contacts, fake contacts, and metal1-metal2 vias are shown in dark red, blue, light orange, black squares, orange squares, and pink squares.[105].	45
2.11	Schematic and block-level representation of the a) Miller op-amp and b) RF $\Sigma\Delta$ ADC, that are used as test circuits to validate the performance of the proposed AMS camouflaging techniques [105].	47
2.12	Block level representation of in-cone logic locking methodologies, including (a) XOR-based [109], (b) MUX-based [110], and (c) LUT based [111]. . .	48

2.13	The (a) structure of a metal-insulator-metal (MIM) memristor and (b) characterization of a MIM memristor indicating the HRS and LRS states and the corresponding V_{SET} and V_{RESET} voltages needed to change the states of the memristor [112].	51
2.14	Block-level representation of the (a) memristor cross-bar structure based on a voltage divider circuit and (b) application of the memristor cross-bar structure to tune the bias voltages of the differential pair transistor of the sense amplifier, which minimizes the mismatch from process variations [112].	52
2.15	Block level representation of the logic locking technique to obfuscate PVT tuning circuitry of the AMS circuit proposed in [113].	54
2.16	Block level structure describing the architecture of (a) stripped functionality logic locking technique [115] and (b) application of SFLL to obfuscate the digital sub-block of an AMS circuit [114].	55
3.1	Miter circuit with replicated A and B versions of the reverse engineered netlist that includes obfuscated gates. For different applied key, the corresponding outputs between the A and B versions of the circuit are XORed. Each XORed signal is then passed to an OR gate to check for any differentiating output.	78
3.2	Analysis of the execution time (left) and number of input/output combinations needed to decrypt the circuit (right). Each color represents a different locking technique and each marker represents a different allowed percentage overhead [166].	80
3.3	Circuit schematic of a differential amplifier.	95

3.4	Circuit schematic of a two stage amplifier.	97
3.5	Circuit schematic of a LC VCO.	98
4.1	Schematic representation of vector based parameter obfuscation implemented on a single transistor.	105
4.2	Schematic representation of mesh based parameter obfuscation implemented on a single transistor.	107
4.3	Circuit schematic of an active inductor based second order band pass filter.	109
4.4	Characterization of the center frequency of the BPF for widths of 0.5 μm to 20 μm for transistors M_4 , M_{11} , and M_{12}	110
4.5	Circuit schematic of implementing the (a) vector based and (b) mesh based obfuscation techniques on a voltage divider circuit that produces a target output voltage of 0.9 V. Transistor width over length (W/L) ratios and applied voltages are provided in the figure.	112
4.6	Three potential sub-divisions of a 34 μm target width into two sub-target transistor widths are shown. Both (a) and (b) violate criterion (1), whereas (c) meets the conditions set by criterion (1).	117
4.7	Consideration of selection criteria when sizing the <i>obfuscation transistors</i> for <i>target transistor</i> sizes of 14 μm and 20 μm , where the cumulative <i>target transistor</i> width is 34 μm . The four scenarios describe (a) when more than one effective width combination is within $\pm 10\%$ of the target width, (b) a large difference between the <i>target transistor</i> widths and the <i>obfuscation transistor</i> widths, (c) the presence of redundant widths, and (d) the desired <i>obfuscation transistor</i> widths that meet all design criteria.	119

4.8	Variation in the center frequency of the BPF for different currents through transistor M_{12} . An AC input signal of 1 mV is applied to determine the gain.	126
4.9	Variation in the gain of the BPF for different widths of transistor M_{14} . An AC input signal of 1 mV is applied to determine the gain.	127
4.10	Schematic of the key-delivery circuit that activates or de-activates the parameter obfuscation transistors based on the applied key bits.	127
4.11	Circuit schematic of a two-stage operational amplifier.	129
4.12	Variation in the gain of the op-amp for different widths of transistor M_5 . An AC input signal of 1 mV is applied to determine the gain.	130
4.13	Variation in the gain of the op-amp for different widths of transistor M_8 . An AC input signal of 1 mV is applied to determine the gain.	131
4.14	Block diagram of the direct conversion receiver.	135
4.15	Relation between the key-size and the area of an op-amp obfuscated using the vector-based obfuscation technique for different percentage variation of both the target transistor size and the obfuscation transistor size. . . .	137
4.16	The relation between key-size and the execution time of the SMT algorithm to determine the obfuscation transistor sizes when implementing the vector-based obfuscation technique.	138
5.1	Histogram of the gain of a BPF for different applied keys, where the gain is in the range of -15 dB to 35 dB. The BPF is obfuscated using the vector-based and mesh-based techniques, with analysis performed on the obfuscated BPF for different process corners.	148

5.2	Analysis of the variation in the gain of the op-amp for all 2^{10} effective sizes of M_5 and considering a 25% tuning range of bias current I_0 . The analysis is performed in the SS, SF, FS, and FF process corners. The vertical red line represents the current that produces the target gain of ~ 1500 V/V.	151
6.1	AMS circuit obfuscated with the proposed technique.	157
6.2	Schematic of the common-source amplifier with diode-connected load. . .	158
6.3	Schematic representation of the flash ADC.	160
6.4	FSM of the peak detection and counting circuit.	161
6.5	Schematic of the obfuscated common source amplifier using vector-based parameter obfuscation.	163
6.6	AMS circuit topology with dependencies between the analog and the digital blocks. The key for the digital circuit block is now dependent on the key set for the analog circuit.	168
7.1	Effective transistor widths producing corresponding gains when applying different keys to a CS amplifier obfuscated with a 5-bit key.	174
7.2	Effective transistor widths of a CS amplifier with corresponding gain when obfuscated using non-monotonic key dependencies. Both transistors of a CS amplifier are obfuscated resulting in a non-monotonic gain response. .	182
8.1	Analysis of the variation in μC_{ox} of the critical transistors of (a) - (d) the single-stage VGA shown in Fig. 8.2a and (e) - (h) the op-amp shown in Fig. 8.3 when obfuscating the transistor dimensions ($\frac{W}{L}$) of, respectively, M_3 and M_5 , which are utilized in the equation-based SMT attack [201]. . .	191

8.2	Circuit schematic of a (a) single stage VGA [212] and (b) two-stage VGA. Each circuit is secured with key-based parameter obfuscation, where the single stage VGA is obfuscated by a 10-bit key and the two stage VGA is obfuscated by a 12-bit key.	197
8.3	Circuit schematic of a two stage operational amplifier [213].	198
8.4	Circuit schematic of a Gilbert mixer [214].	198
8.5	Block diagram of the front-end circuit consisting of a mixer and a two-stage VGA.	199
8.6	Flowchart of the SMT-based monotonic attack used for the four scenarios described in Section 8.3, where (a) circuit equations are used to obtain input-output responses for Scenarios I to III and (b) Cadence-based oracle circuit emulation is used to obtain input-output responses for Scenario IV, where the magenta box illustrates the steps performed by the SMT solver and the blue box illustrates the steps performed in Cadence.	213
9.1	Characterization of the variation in the number of returned keys for a (a) single-stage VGA, (b) 2-stage VGA, and (c) analog front-end circuit, and an analysis of the execution time required to return candidate keys for a (d) single-stage VGA, (e) 2-stage VGA, and (f) analog front-end circuit. The analysis for both the number of returned keys and the execution time is performed for the SMT, GA, key-spacing, and monotonic response attack algorithms for key sizes of 9-bits to 30-bits.	223
10.1	Flowchart highlighting the primary steps of the DC nodal analysis attack on key or threshold voltage obfuscated circuit.	234

Abstract

The focus of this dissertation is the safeguarding of analog circuits against IP piracy attacks, which includes the development of a novel method to secure analog IP, the assessment of existing analog attack algorithms, and the development of metrics to measure the resilience of state-of-the-art analog obfuscation techniques. The security technique provides key-based obfuscation of transistor dimensions to mask the biasing conditions of an analog circuit. An operational amplifier (op-amp) obfuscated with a 10-bit key results in a probability of determining the correct key with a brute-force attack to $9.78 \cdot 10^{-2}\%$, where a correct key results in a gain of 64 dB, while an incorrect key results in at least 30% degradation in the gain of the op-amp. To measure the resiliency of analog obfuscation techniques, various attack algorithms are presented, and an analysis on the performance of analog attack algorithms when considering non-ideal threat scenarios is performed, which includes the development of metrics to assess the amount of time required to setup and execute an attack. The evaluation of the metric indicates that the monotonic attack is the fastest to execute for single-stage analog circuits, but fails to return a correct key for multi-stage circuits that provide non-monotonic output responses. For multi-stage analog circuits, the key-spacing attack is the fastest to execute; however, the attack is only applicable to key-based obfuscation techniques. The SMT-based attack is 224x slower than the monotonic attack and 2240x slower than the key-spacing attack, where the attack requires

approximately 15 hours to determine the correct key of an 18-bit obfuscated analog front-end. The genetic algorithm (GA) based attack is 121091x slower than a monotonic attack for a single stage analog circuit, and failed to return the correct key for multi-stage analog circuits. The analysis of the resilience metric indicates that the current analog attack algorithms are inefficient when considering non-ideal threat scenarios and are not applicable to all analog obfuscation techniques or to multi-stage obfuscated circuits. Notably, a novel DC-based nodal analysis algorithm is developed to measure the security resilience of analog obfuscation techniques considered under various attack scenarios. The DNA attack successfully eliminated 99.1% and 99.98% of the keys used to secure an analog front-end circuit with key-based and multi-threshold obfuscation, respectively. The utilization of the DNA attack for different real-world scenarios provide a standard metric to effectively evaluate the strength of analog obfuscation techniques. In summary, the work described in this dissertation enhances analog circuit security against IP piracy attacks and provides metrics to measure the efficacy of current and future obfuscation methodologies.

Chapter 1

Introduction

“Computing is not about computers anymore. It is about living”

– Nicholas Negroponte

The semiconductor industry has seen significant growth in recent years, driven by the increasing demand for electronic devices in nearly every aspect of modern society from consumer electronics to transportation and communication systems [1, 2]. Due to this ubiquity, the semiconductor industry has become a critical part of modern society, as suggested by Mr. Negroponte. The demand for continuous innovation in the semiconductor industry is a result of a rapidly expanding global semiconductor market that is projected to reach \$772 billion by 2030 [3].

In parallel to the expansion of the overall global semiconductor market, the analog integrated circuit (IC) market has seen significant growth due to increasing demand from the automotive industry, internet of things (IoT) applications, 5G communication, and

the general use of analog computation [4–6]. The market for analog ICs is expected to experience a 5.5% cumulative annual growth rate from 2019 to 2024 and reach \$134 billion by 2027, accounting for approximately 22% of the overall semiconductor market [4, 7]. The high demand and fragmentation of the analog IC market have led to competition among manufacturers based on product innovation and technology differentiation. In order to gain a competitive edge, many companies invest heavily in the development of analog ICs in order to beat competitors to market, which is often achieved through the use of off-shore fabrication, packaging, and testing facilities that reduce production costs and speed up the manufacturing process [8]. However, such globalization also increases the risk of security vulnerabilities to the IC supply chain [9].

A study by IBS group [10] found that on average 20% of the area of a chip is occupied by analog circuits. At the same time, 40% of the total effort is dedicated to analog design, and 50% of costly design re-cycles are due to the analog components. The complexity of designing analog circuits, the increased risk of failure, the lack of sophisticated analog EDA tools, and the potential for a first-mover advantage have all contributed to the increased attacks on analog IP. Consequently, analog circuits are the most commonly counterfeited ICs, accounting for nearly 25% of all counterfeit semiconductor electronics [11]. Ensuring the security of analog ICs is, therefore, vital to a secure technological landscape.

To ensure a secure technological landscape, it is imperative to address the challenges in

designing, developing, and producing analog ICs and understand the motivations behind the attacks on analog circuits. Therefore, this chapter provides an introduction to the history of analog ICs, providing background on the evolution of analog IC technology and the drivers for the growth in the industry. The expanding uses of analog circuits and the contributions of analog circuits to the present digital age are described, followed by a discussion of challenges in the design, development, and production of analog ICs. Finally, the primary motivations of attacking analog ICs and the resulting consequences are discussed.

1.1 History of Analog Computing

The use of analog computers dates back to ancient Greece, approximately, 85 to 60 before the common era (B.C.E) [12]. Analog computers that consisted of a complex system of gears and cogs implemented the Antikythera mechanism, which was used to track the movement of the sun, moon, and planets [12]. The Antikythera mechanism was manually operated by a hand crank and was significantly smaller than the solar system being modeled. In addition, celestial events of the past or the future, were simulated with the Antikythera mechanism.

Many medieval models and devices are considered analog by nature, often used to solve problems in fields such as astronomy, navigation, timekeeping, agriculture, construction, and engineering through complex calculations, measurements, and predictions

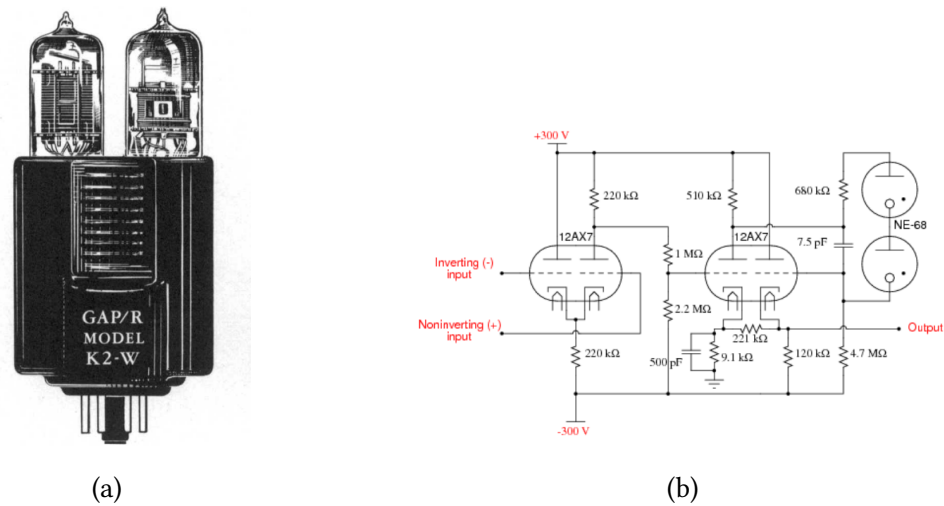


Figure 1.1: A (a) photograph and (b) schematic representation of a K2-W operational amplifier [18].

[13]. In the modern era, the differential analyzer developed by Vannevar Bush of MIT in 1930 was the first general-purpose mechanical analog computer capable of solving sixth-order differential equations [14]. The advent of World War II demanded faster and more portable analog computers capable of not only aiming and firing artillery shells but also predicting the trajectories of the fired shells to increase accuracy and minimize the amount of ammunition required. The first electronic analog computers were developed by Hendrik Bode, Claude Shannon, and Richard Blackman of Bell Labs in 1940 as part of the M-9 electrical gun director [15]. The M-9 electrical gun, coupled with the proximity-fuse ordnance, reduced the amount of ammunition expended in World War II by 10x [15, 16]. The computing for the M-9 gun director was executed by vacuum-tube feedback amplifiers configured to perform mathematical functions including integration, inversion, and summation [17].

The first commercial amplifier, the K2-W operational amplifier (op-amp), was developed by George A. Philbrick in 1952 [18]. The DC-coupled op-amp operated at $\pm 300\text{V}$ with a $\pm 50\text{V}$ peak-to-peak swing when a $50\text{ K}\Omega$ output load was applied. The op-amp was designed to produce a DC gain of 80 dB and a unity gain frequency of 300 KHz [18]. The K2-W op-amp achieved the 80 dB gain with two stages that utilized both a positive feedback loop and the Miller effect [19] that allowed for high stability with a near-single pole behavior over a wide frequency range.

1.1.1 Invention of the Transistor

The primary building block of modern integrated circuits (ICs) is the transistor, where the fundamental functionality of the transistor in the analog circuit domain is simplified to that of a switch that controls the amount of current flowing through the device. All complex functionality of an IC, such as for computer and cell phone processors, is the control of these high-speed switches.

The first transistor, the point-contact transistor shown in Fig. 1.2a, is most commonly attributed to John Bardeen, Walter Brattain, and William Shockley of Bell Laboratories in 1947 [20–22]. The point-contact transistor was first utilized in an amplifier circuit for an audio microphone and then in an oscillator to demonstrate power gain [23]. While the point contact transistor was a monumental achievement, the size of the original device and the required wiring prevented the integration of many such transistors to form a single

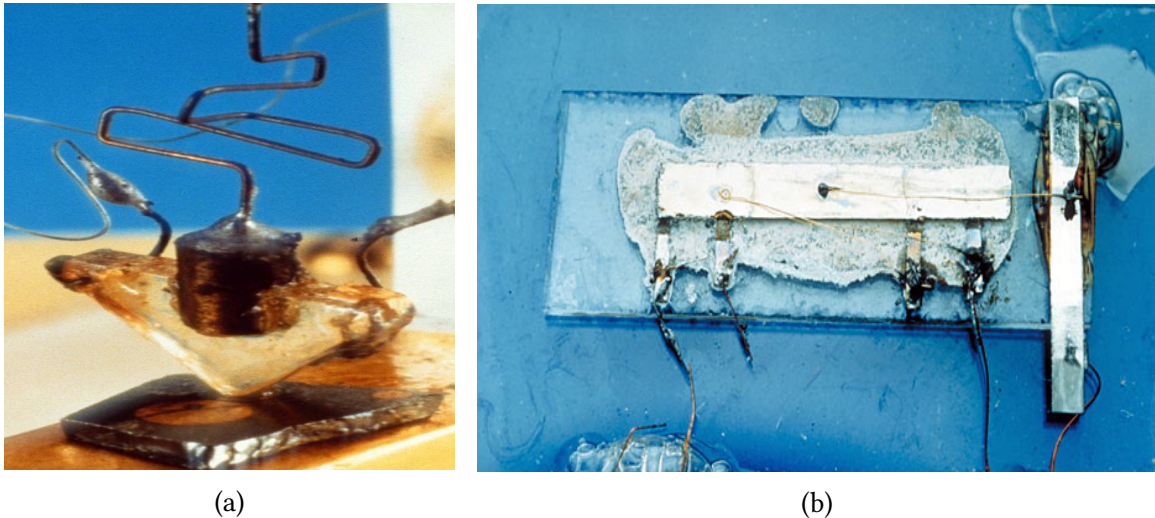


Figure 1.2: The (a) first point contact transistor developed by Bardeen, Brattain, and Shockley at Bell Laboratories in 1947 [20, 21] and (b) the first integrated circuit developed by Jack Kilby at Texas Instruments in 1958 [21].

circuit. The use of discrete solid-state semiconductor transistors in analog circuits became practical after the invention of the bipolar junction transistor (BJT) by William Shockley in 1948 [20, 21, 24], which facilitated the fabrication of transistors in large volumes [24]. In addition, the contributions of J. M. Early, J. J. Ebers, and J. L. Moll in understanding the operation, design, and large-signal transient behavior of the BJTs led to the widespread adoption of BJT transistors in analog circuits [25, 26].

1.1.2 Development of Integrated Circuits

Jack Kilby of Texas Instruments changed the paradigm of separately fabricating transistors to build an electrical circuit in 1958 by utilizing a single piece of material to integrate multiple devices [27–29]. Placing multiple devices on a single substrate credited Kilby with producing the first integrated circuit (IC), which utilized external interconnects to transmit signals between the transistors, as shown in Fig. 1.2b. The IC concurrently developed by Robert Noyce at Fairchild Semiconductor was publicly disclosed in 1959, a few months after Kilby’s initial IC was disclosed. Fairchild Semiconductor integrated both the transistors and the interconnects within a single substrate to form a monolithic process [27–30]. The modern fabrication process utilizes the same principles developed by Robert Noyce, where the transistors and the interconnects are integrated on a single substrate [30]. In the analog domain, the first monolithic integrated circuit was commercially introduced in 1963 as a five transistor 100 MHz μ A703 radio frequency (RF)-intermediate frequency (IF) amplifier [31].

1.2 Drivers for Development of Analog Integrated Circuits

The contributions from Robert Widlar [34, 35], Barrie Gilbert [36], and George Wilson [37] facilitated the early development of the new field of solid-state analog circuit

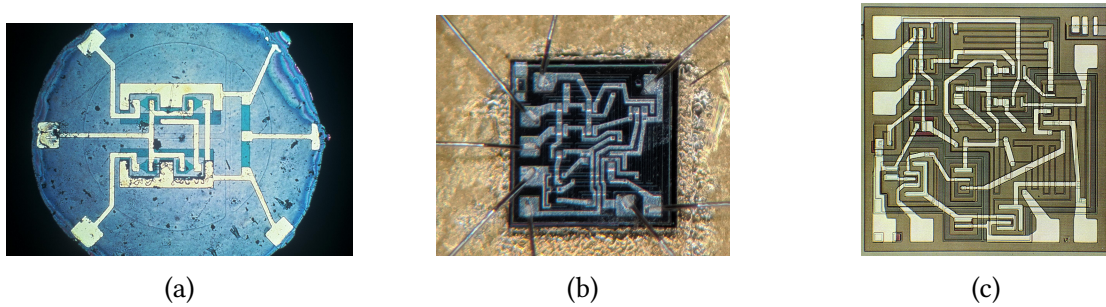


Figure 1.3: Evolution of monolithic analog integrated circuits observed from the family of Fairchild linear ICs produced in the 1960's, where a microphotograph is shown of the (a) first integrated circuit with four transistors, (b) first BJT operational amplifier $\mu A702$ with nine transistors [32], and (c) the most successful $\mu A709$ operational amplifier [33].

design, specifically the monolithic solid-state analog integrated circuit using bipolar transistor process. The introduction of the bipolar transistor process with a vertical N-P-N and lateral P-N-P in the monolithic solid-state analog integrated circuit provided both superior transistor performance and a simpler manufacturing process, which in turn, allowed for the high-performance and high-volume production of analog circuits. Robert Widlar, who was then an employee of Fairchild semiconductor pioneered the monolithic integrated operational amplifiers [32–35], that were offered commercially in the 1960s, three of which are shown in Fig. 1.3. Barrie Gilbert was a pioneer in the field of wideband amplification using bipolar junction transistors (BJTs) and the linearized trans-conductance operation of BJT transistors. Barrie Gilbert is also credited for the invention the BJT linear multiplier (known as the Gilbert multiplier) [36], which is still utilized even today in various applications, including the design of high-speed phase detectors, variable gain-controlled amplifiers, RF mixers and modulators, and precise linear multipliers. George

Wilson played a key role in the development of current sources and operational amplifier design. He was also responsible for creating circuits that effectively utilized both BJT and JFET transistors when integrated together in a monolithic process [37].

Although monolithic analog ICs are credited for high-performance specifications, early monolithic analog ICs were limited in frequency due to a lack of circuit models that accurately predicted the behavior of the ICs at higher frequencies and a shallow understanding of the second-order effects impacting the ICs at higher frequencies [38]. The development of a simulation program with integrated circuit emphasis (SPICE) in the 1970s revolutionized the design of monolithic ICs. SPICE provided better predictability of circuit performance in both low and high frequencies [38], which enabled the design and implementation of monolithic analog ICs for high frequency applications. Initially, customers with critical size and weight constraints, such as NASA for the Apollo Guidance Computer [39–41], were able to justify the cost. However, the public release of SPICE and the subsequent growth of circuit simulators from the nascent computer-aided design (CAD) industry resulted in faster and cheaper design and development cycles of integrated circuits and facilitated the widespread commercial adaptation of monolithic ICs [42]. The use of SPICE and the corresponding highly accurate circuit models that were utilized enabled the evolution of analog integrated circuits in two broad areas: (1) metal-oxide-semiconductor field-effect transistor (MOSFET) based ICs for analog and mixed signal (AMS) applications and (2) radio-frequency (RF) integrated circuits using bipolar and complementary

metal-oxide-semiconductor (BiCMOS) transistors [42].

1.2.1 Development of Analog MOS ICs

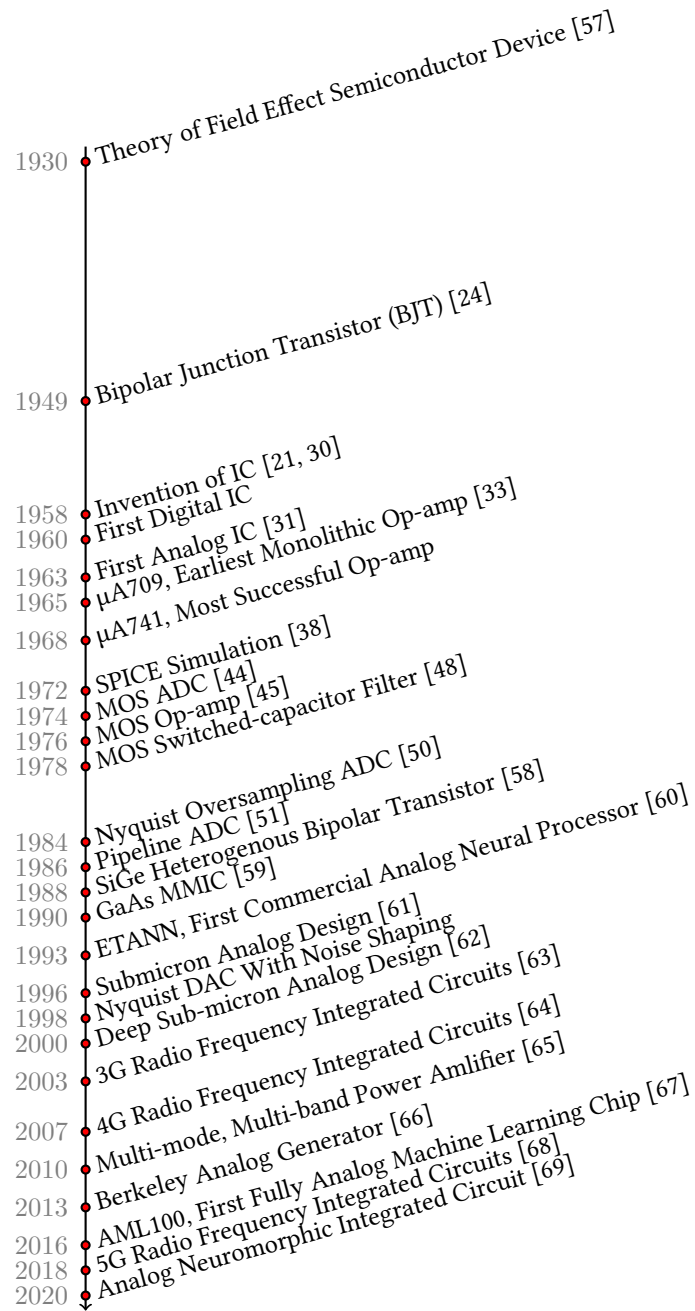
The advancement of circuit simulators and CAD tools for designing and developing ICs led to the adoption of digital ICs consisting of structured building blocks. The generation of logical gates utilized in digital circuits and standard structures allow for the automated placement, routing, and optimization of the sub-blocks of an IC. The growth of digital circuitry for computation and signal processing necessitated the need for low-cost analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) [42]. Analog Devices Inc. developed the first BJT-based monolithic DAC in 1970 that utilized both replicated biasing and operation with constant-current-density of binary-weights [43]. BJTs based on a P-N-P material structure became the primary device that implemented DACs and successive approximation (SAR) ADCs [43].

To achieve lower production costs and to allow for the integration of logical functions, the fabrication of analog ICs transitioned to metal-oxide-semiconductor (MOS) technology in the mid-1970s [42]. The contributions of Dr. David Hodges, Dr. Paul Gray, Dr. Robert Meyer, and Dr. Robert Broderon at the University of Berkeley enabled the large-scale integration of analog circuits with digital signal processing, logic, and memory [44–50]. The increasing level of integration of analog with digital circuits allowed for complex analog-mixed-signal (AMS) functionality. Utilizing AMS ICs allowed for the continuous

improvement in density and performance provided by digital MOS technology scaling while also easing the challenge of designing and implementing complex purely analog circuits. In addition, the utilization of digital circuits with analog allowed for the integration of digitally controlled compensation circuits as MOS transistors provide poor matching of threshold voltages as compared to bipolar transistors [42].

The extensive research and development of AMS ICs led to high-speed, high-bandwidth circuits. In addition, the introduction of digital filters and switched-capacitor circuit elements lead to the development of Nyquist ADCs and DACs [51, 52]. Prior to such advancements, the state-of-the-art ADCs at the time were only useful for low bandwidth applications due to limitations in noise shaping techniques utilizing oversampling and filtering. Nyquist ADCs and DACs sample the input signal at a rate that is at least twice the highest frequency component of the input signal (Nyquist rate) [53, 54]. In contrast, traditional ADCs and DACs do not necessarily adhere to the Nyquist rate and employ other sampling techniques, including oversampling [55] or delta-sigma modulation [56] to reduce the conversion error. Nyquist ADCs and DACs that do not require the additional sampling circuits therefore, offer superior speed and accuracy and are typically less complex and expensive than other types of ADCs [51, 52].

Figure 1.4: Timeline of key innovations and developments in analog, AMS, and RF integrated circuits [42].



1.2.2 Development of BiCMOS RF ICs

Concurrent with the development of CMOS analog ICs, narrowband RF ICs were introduced that leveraged the electrical and physical advantages provided by monolithic integrated circuits. Lithographic scaling of the bipolar transistors, which also resulted in higher transistor density, enabled greater performance, reduction in power consumption, and reduced costs. The advancement of analog MOS technology that provided higher integration with digital circuits and ease of fabrication, coupled with the bipolar technology that offered higher switching speed while maintaining good noise performance, led to the development of BiCMOS technology. The evolution of the BiCMOS process enabled novel circuit topologies and structures including optical network transceivers and low-cost RF transceivers for both cellular and wireless local area network (LAN) applications [70, 71].

RF ICs achieved much higher operating frequencies with the development of silicon-germanium (SiGe) heterojunction bipolar transistors, which provide a much higher maximum oscillation frequency (f_{max}) than silicon-based bipolar transistors. Transceivers utilizing SiGe transistors achieved much higher frequencies, lower noise, better stability, and larger bandwidth than transceivers utilizing traditional bipolar transistors [72].

A summary of the history and innovations in the field of analog integrated circuits is shown in Fig. 1.4. The drastic improvements in the performance, functionality, and per unit cost of analog ICs are due to two primary reasons: (1) improvements and innovations in the fabrication process, and (2) large contributions by the pioneers in the analog

domain with innovations in the fields of devices (inventions of BJT, MOS, BiCMOS, and SiGe technologies), circuits (design of feedback paths and use of Miller effect to improve performance and stability), and systems (development of ADCs, DACs, and transceivers). The combined contributions have resulted in the widespread adoption of analog ICs in commercial, military, and space-based applications for sensing, filtering, amplifying, converting, and computing.

1.3 Growth of Analog Computing

The use of analog circuits for complex calculations and data processing traces back to the 1970s. Analog computing consisted of comparators, multipliers, and function generators [73]. However, utilization of digital computing eventually surpassed analog computing due to improvements in digital clock speeds, flexibility in the use of algorithms, and the ability to share resources amongst executing tasks. In addition, analog circuits were more susceptible to errors caused by noise, temperature fluctuations, and manufacturing defects as compared to digital circuits [74].

In the past 15 years, due to the slowing of Moore's Law and the relative stagnation of processor clock speeds, alternatives including manycore architectures and vectorization have been explored [75]. However, such approaches also suffer from limitations, as described by Amdahl's Law, which quantifies the maximum improvement in performance achieved through parallelization for a fixed problem size [76]. The need to improve circuit

performance beyond clock speed resulted in the development of general-purpose graphical processing units (GPGPUs) and specialized digital computing circuits that leverage vectorized and data flow-oriented programming paradigms [77]. However, the performances of the GPGPUs are still limited by the intrinsic parasitic capacitances of the devices, which restrict the maximum clock frequency and consume significant energy [73].

Analog circuits excel at processing continuous real-time signals with fewer transistors than their digital equivalents, leading to faster and more efficient computing tasks including multiplication, summation, and integration [13]. In contrast to digital signal processing, analog signal processing requires no additional ADCs and DACs to convert signals between real-time and discrete-time domains for computation, leading to lower power consumption and increased energy efficiency. Error-checking circuitry is also unnecessary with analog signal processing [16]. As a result, analog circuits are well-suited for applications that demand fast and accurate processing of large amounts of real-time data, including audio, video, and image processing.

One study found that an analog neural network designed for surface recognition applications reduced the power consumption by 62% and the delay by 28.5% as compared to the same neural network implemented in the digital domain [78]. Another study found that analog circuits for computing consume about 96% less power and require about 95% less energy for floating point operations as compared to a state-of-the-art Intel mobile processor [73]. In addition, the utilization of resistive random access memory (RRAM) for

processing in-memory (PIM) along with an analog data flow has been shown to achieve an improvement of 5.4x and 3.4x in, respectively, energy efficiency and computational throughput on various deep neural network benchmarks including Alexnet, GoogleNet, MobileNet, and NeuralTalk [79]. In contrast, traditional neural networks implemented using digital circuits utilize a Von Neumann architecture, in which memory and processing elements are separated and interact through an interconnect network that limits performance [79].

The growth of artificial intelligence (AI) and the benefits analog computing provides over digital computing has led to a rise in the use of analog ICs. According to market estimates, the AI industry is expected to generate \$296 million in revenue by 2024, with a compound annual growth rate (CAGR) of 20.5% [80]. In addition, the automotive industry accounts for 29% of the total revenue of sold analog ICs and is expected to increase further due to the introduction of electric and autonomous vehicles [4]. With the increasing number of sensors integrated into electric and autonomous vehicles for LIDAR (light detection and ranging), blind-spot detection, driver assists, and backup cameras, there is a greater need for application-specific analog ICs to interface between the sensory data of the real world and the digital computing units found in the automobile.

The surge in demand for analog circuits is not only driven by the expansion of AI and automotive sectors, but also by the escalating need for internet of things (IoT) devices and the widespread adoption of 5G communications. The demand for IoT devices increased

by around 180% from 2019 to 2022 [6], where analog ICs are used for sensing, interpretation, communication, and power management. The advent of 5G communications and technologies has increased demand for analog and RF ICs, which are used in base stations, smartphones, and tablets [5]. Due to the increasing demand for analog circuits, the analog IC market is expected to grow to \$12.84 billion in 2027 from \$9.9 billion by 2023 at a CAGR of 6.71% [7].

1.4 Challenges in Design and Development of Analog Circuits

Although analog circuits often have superior performance in many computing applications as compared to digital circuits, wide use has yet to be achieved for two primary reasons. First, analog circuits are highly specific for a given application and are not readily adaptable to different applications in the way digital circuits are. Second, designing and developing analog circuits requires significantly more effort than digital circuits. A study by IBS group [10] found that even though analog circuits occupy 20% of chip area, 40% of the total effort in designing an IC is dedicated to analog design. In addition, 50% of overall costly design rework is due to the analog circuits. The significant differences in the design of analog circuits as compared to digital circuits contribute to the large design time of analog ICs. Design considerations for an analog IC include

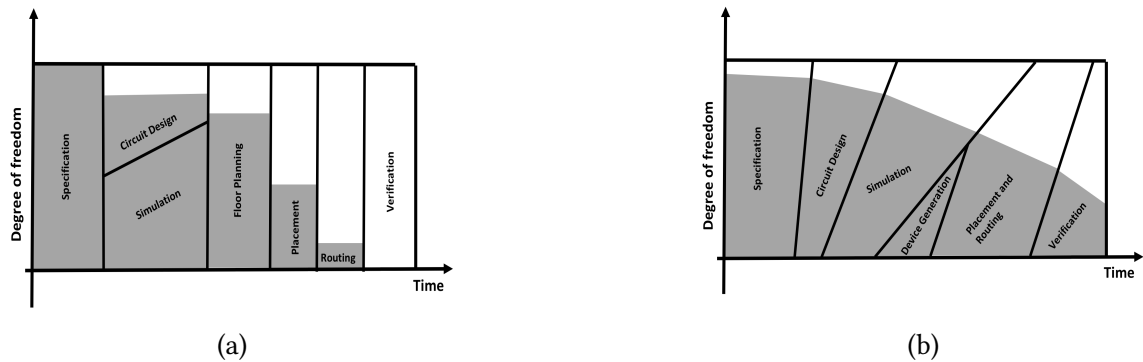


Figure 1.5: Comparison of design flows depicting a a) stepwise reduction of degrees of freedom for digital circuits, and b) continuous reduction of degrees of freedom and overlap of design flow for analog circuits [81].

1. greater precision in biasing conditions,
2. greater sensitivity to noise and temperature,
3. greater emphasis on signal integrity,
4. tighter noise margins, and
5. simultaneous consideration of multiple parameters and the overlap of several steps in the design flow (whereas digital circuits follow a sequential design flow), as shown in Fig.1.5.

Analog circuit design involves a range of manual tasks, including selecting circuit architectures and topologies, setting test vectors, sizing circuits, and creating layouts. The tasks are time-consuming and require expertise in analog design. In addition, the lack of automation leads to long design times and a large amount of manual input, even for small

circuits. Analog layout, in particular, is a tedious process [81]. In contrast, digital circuits benefit from complete synthesis and physical design flows. As a result, analog circuits are critical in terms of time-to-market and are often the main cause of circuit failure [82].

Technology scaling has resulted in increased failure rates of analog ICs, requiring redundancies in design to compensate for failures that lead to increased size, cost, and testing time [83]. In addition, a lack of robust analog models for advanced technology nodes has resulted in a greater challenge in the design of analog circuits, as the effect of individual elements on the active circuit is harder to predict [84].

1.5 Introduction to Analog Hardware Security

The projected increase in the total revenue of the analog IC market as described in Section 1.3, coupled with the challenges of designing and developing analog circuits as described in Section 1.4, has resulted in an increase in analog IP piracy. Analog ICs are, therefore, the most commonly counterfeited type of circuits [85]. In addition, electronic devices suffer from a longer time-to-market due to the increased complexities and challenges of designing and developing analog integrated circuits (ICs). To maintain the first-mover advantage and continue generating revenue, analog IC companies are increasingly moving away from a vertical process in which a single entity completes all aspects of IC design, fabrication, and packaging. Instead, a horizontal process is being adopted, where multiple third parties are involved in the design, fabrication, and testing of ICs [86].

1.5.1 Security Vulnerabilities of a Horizontal IC Design Flow

The inclusion of untrusted third parties throughout the horizontal IC design and manufacturing flow, including in standard cells (IP), automatic design tools (EDA), fabrication facilities, test facilities, and end-users, have resulted in increased security threats. The threats include Trojan insertion, counterfeiting, reverse engineering, IP theft, and overproduction of ICs. The vulnerabilities throughout the life cycle of an analog IC are shown in Fig. 1.6. During the design phase, an IC is vulnerable to IP piracy and the insertion of hardware Trojans. In addition, a rogue employee is capable of stealing IP and tampering with the RTL code, which leads to modified functionality or a leak of secret information through backdoor channels. During fabrication, an untrusted foundry has the potential to counterfeit an IC through overproduction of the circuit beyond contracted obligations. During assembly, the IC is prone to an untrusted facility supplying defective chips or changing the grade of the IC. Throughout the lifetime of a circuit, an attacker has the ability to reverse engineer and gather design information to counterfeit the circuit by cloning. Finally, at the end of the IC life-cycle, circuits are recycled from an old electronic system and resold on the open market as new [9].

The IC is most vulnerable during the fabrication phase and active lifetime of the circuit. A third-party fabrication facility has access to the design of an IC, typically in the GDS-II format, and possesses the required tools and knowledge to reverse engineer the circuit from the GDS-II file alone [87]. Access to the design files allows an untrusted foundry

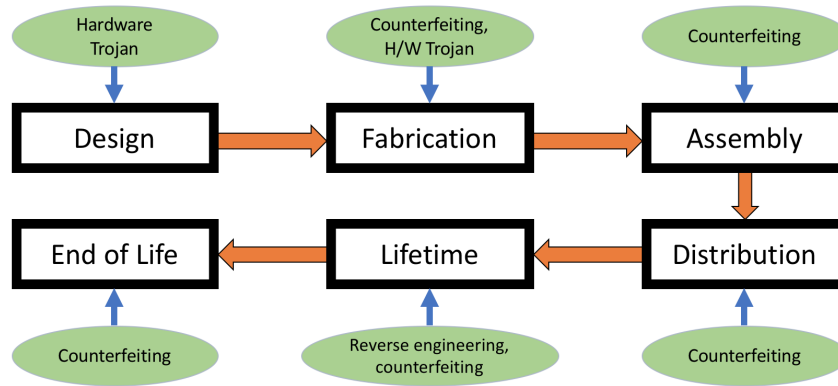


Figure 1.6: Vulnerabilities in the analog integrated circuit supply chain.

to steal, overproduce, and insert hardware Trojans into the circuit. Another vital threat scenario to consider is an untrusted end-user. IP theft and counterfeiting are possible with direct access to a fully functional IC and available capabilities in image-based reverse engineering [87].

1.5.2 Examples of Analog IP Piracy Attacks

In this sub-section, examples of pirated and counterfeit analog IP are provided to highlight the extent and the resulting consequences of such attacks.

1.5.2.a Counterfeited Analog Multiplier

A suspicious analog multiplier AD633 purchased online from *Analog Devices* is investigated [88, 89]. An image of the suspicious packaged analog multiplier is shown in Fig. 1.7a [88]. A closer examination of the package revealed that the top surface had been sanded. In addition, the profile of the package did not match the expected specifications,

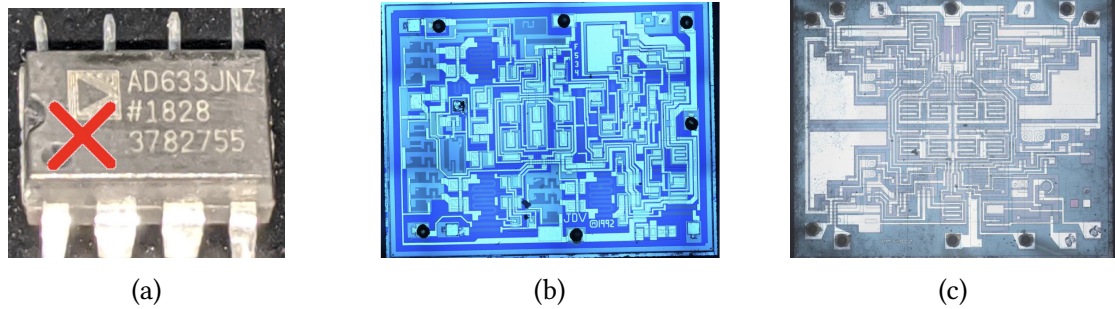


Figure 1.7: Microphotograph of the (a) packaged AD633 analog multiplier IC, (b) die of the original intended AD633 IC [88, 90], and (c) die of the fraudulent AD633, which on further investigation is the RC4200 from *Raytheon Semiconductor* [91].

where the thickness of the top surface measured 3.27 mm as compared to the expected value of 3.65 mm [88].

The delayering and imaging of the suspicious IC revealed the microphotographed die shown in Fig. 1.7c. The expected microphotograph of the AD633 is shown in Fig. 1.7b. Upon further investigation, the suspicious die resembled that of RC4200 analog multiplier from *Raytheon Semiconductor* [89]. The primary differences between the two multipliers is that the AD633 costs approximately \$7.2 and operates at a supply voltage of ± 18 V [90], whereas the RC4200 operates at -22 V and costs approximately \$0.8 [91]. The potential use of a positive voltage (> 0 V) might lead to overheating and eventual failure of the counterfeited IC.

1.5.2.b Counterfeited Operational Amplifier

Another example of the scale of analog IC counterfeiting is described in [92]. The counterfeit IC was an audiophile high-quality operational amplifier (op-amp) OPA627 that

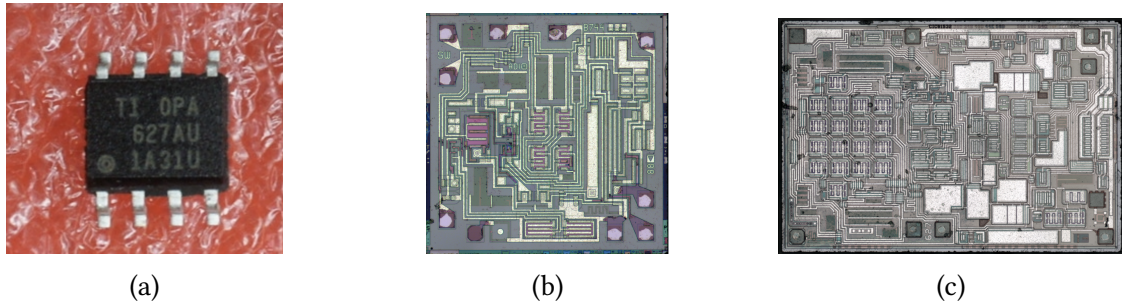


Figure 1.8: Microphotograph of the (a) packaged OPA627 IC bought online, (b) die of the fraudulent OPA627, and (c) die of the authentic TI OPA627 IC [88, 90], which on further investigation is the AD744 from *Analog Devices* [93].

was purchased online (eBay). A typical OPA627 costs anywhere from \$16-\$80 depending on the selected encapsulating package, whereas the OPA627 that was purchased online and is shown in Fig. 1.8a was marketed at \$2.7, including shipping, which raised suspicion [92].

When the suspicious OPA627 was delayered and imaged, the microphotographed die shown in Fig. 1.8b did not match the image of the true OPA627 produced by *Texas Instruments*, which is shown in Fig. 1.8c [94]. Upon further investigation, the suspicious die resembled that of the AD744 operational amplifier from *Analog Devices* [93]. The primary differences between the two ICs are that the AD744 produces 3x more noise than the OPA627 and the offset voltage is 0.5 mV, as compared to the 0.1 mV of the OPA627 [93, 95]. The potential use of the counterfeited OPA627 amplifier may result in errors and, in the worst-case, in the lack of function in the electronics that utilize the IC.

In addition to the two examples, various other instances involving counterfeited analog ICs have been reported [96, 97]. The use of counterfeits, including analog ICs, has been

strongly linked to the loss of life [96], and the loss and incorrect function of equipment, including military hardware, defibrillators, and braking systems in high-speed trains [98].

1.5.2.c Consequences of IP Piracy Attacks

Counterfeit semiconductor components are a significant threat to the safety, health, and security of individuals and systems that rely on electronic hardware. Electronic products and systems, including military, finance, and transportation infrastructure and consumer electronics, typically rely on a large number of semiconductor components to function correctly. The failure of a single counterfeit component has the potential to cause severe repercussions. Some known incidences and potential consequences of counterfeit semiconductor components as reported by the *Semiconductor Industry Association* [99] include:

1. A fake component in an automated external defibrillator caused an over-voltage condition that could have resulted in improper shocks being administered to heart attack victims.
2. A counterfeit semiconductor in the control circuitry of a vacuum cleaner caused a fire that was contained, which had the potential of causing significant property damage or even loss of life.
3. A counterfeit IC resulted in the power failure of airport landing lights.

4. Multiple instances of a broker admitting to selling counterfeit components intended for use in nuclear power plants, nuclear submarines, and intravenous (IV) drip machines have been reported.

In addition to posing a threat to health, safety, and security, counterfeit semiconductors also have a negative effect on economic output [100]. Original Component Manufacturers (OCMs) invest billions of dollars annually in developing, producing, and supporting high-quality, reliable products. In contrast, counterfeiters invest little to no resources in developing and producing low-quality products and provide no post-sale support. When products from OCM are counterfeited, the resulting low quality and poor reliability damages the reputation of the OCM and leads to loss of business, even though the OCM played no role in the counterfeiting [99, 100]. In addition, counterfeiters often sell products at lower prices than the OCMs, which is attractive to buyers, but there is no guarantee that the purchased components are authentic or reliable. The theft of the intellectual property (IP) of the OCMs as well as being able to operate at lower costs has allowed counterfeiters to undercut the prices set by the OCMs and authorized distributors of the product, resulting in counterfeited and pirated product contributing to an estimated \$461 billion of trade in 2013 [100]. Similarly, counterfeit and pirated products accounted for 3.3% of all world trade in 2019 [100].

1.6 Outline of the Dissertation

The research aims to develop circuit-level techniques to protect analog intellectual property (IP) from various forms of IP theft including reverse engineering, counterfeiting, and overproduction. Algorithms are developed to evaluate the performance and efficiency of analog obfuscation techniques under different threat scenarios. The developed algorithms provide designers a better understanding of the vulnerabilities of analog obfuscation techniques to deobfuscation attacks when considering an adversary that possesses a variable amount of information pertaining to the obfuscated analog circuit. Understanding the vulnerabilities of the analog obfuscation techniques allows designers to choose the most effective method that enhances the security of the circuit while meeting area and performance requirements.

A brief background on early analog obfuscation techniques that were developed utilizing existing digital obfuscation techniques is provided in Chapter 2. Concurrent to the developed parameter obfuscation techniques, different analog obfuscation techniques are also developed to protect analog IP. A brief discussion of different analog obfuscation techniques are also provided in Chapter 2. A developed parameter-based obfuscation technique is described in Chapter 4, which masks the size of critical transistors using a key. When implemented, the analog IP will only function correctly when the correct key

is applied. If the wrong key is applied, the analog IP either does not function or functions with reduced performance. A simulation-based optimization algorithm is presented in Chapter 5 to adjust the biasing conditions and transistor body voltages, which mitigates the effects of parasitic impedance and variations due to the implementation of the obfuscation circuitry. The parameter obfuscation technique, along with the presented algorithms, provide an efficient way to secure analog circuits while reducing the design time to implement the security features.

An evaluation of the vulnerabilities and the performance of current state-of-the-art analog obfuscation techniques is described in Chapter 6, where analog attack algorithms are utilized as metrics for the evaluation. Most analog reverse engineering algorithms assume that the attacker has all the necessary information needed to perform the attack. If one or more information is unavailable, the performance of current analog attack algorithms is negatively impacted. The challenges and limitations of obtaining the necessary information needed to attack an analog circuit are also discussed in Chapter 6. Metrics to characterize the setup and evaluation time needed to execute a deobfuscation attack are provided in Chapter 7.

A DC-based nodal analysis attack is presented in Chapter 8 that requires only the circuit netlist and the DC input-output responses of an oracle IC to deobfuscate a secured analog circuit. With the limited information necessary to execute the attack, the ability to perform the deobfuscation attack under real world attack scenarios, and the potential to

apply the attack to all current analog obfuscation techniques, the DC-based nodal analysis attack will be the de-facto standard to measure the resiliency and strength of analog obfuscation techniques.

Chapter 2

Analog Obfuscation Techniques

Obfuscation is defined as an action that makes something obscure, unclear, or unintelligible. Obfuscation is achieved by either removing a critical piece of information or by adding additional information to increase complexity. The primary objective of obfuscation is to protect proprietary intellectual property (IP). In hardware, obfuscation is utilized to protect circuits against IP attacks including reverse engineering, counterfeiting, and overproduction. Hardware obfuscation techniques were originally developed to protect digital IP, while analog obfuscation was mostly overlooked, as discussed in Chapter 1. However, early research efforts on analog circuit security explored modifying techniques used for digital obfuscation. This chapter provides an overview of the methodologies to obfuscate an analog IC, which include 1) split manufacturing, 2) IC camouflaging, 3) logic locking, and 4) physically unclonable functions. The associated threat models and the

limitations of each technique are also described.

2.1 Split Manufacturing

Split manufacturing is one of the first techniques proposed to prevent analog IC piracy. Split manufacturing protects an analog IC by dividing the fabrication process into a front-end-of-line (FEOL) process, fabricated by an untrusted foundry, and a back-end-of-line (BEOL) process that is fabricated by a trusted foundry. A cross-sectional view of an IC partitioned into front end of the line (FEOL) layers and back end of the line (BEOL) layers is depicted in Fig 2.1. The FEOL layers are fabricated on an advanced process node and include the transistors and often metal layers 1 and 2. In contrast, the BEOL layers are fabricated on an older process node and include higher-level metal layers.

By splitting the manufacturing process between two foundries, the complex design details and specifications are not fully disclosed to an un-trusted foundry. Due to the unique functionality of the metal layers of an analog IC, split manufacturing provides greater security to analog ICs than to digital ICs [101]. The two unique functions provided by the metal layers of an analog IC include

- fabrication of both interconnects and functional blocks including resistors, capacitors, and inductors, and
- utilization of the length and width of the metal layers as functional parameters that

impact the the overall resistance and capacitance of circuits.

Therefore, an untrusted foundry fabricating the FEOL must explore a large design space to reverse engineer an analog circuit.

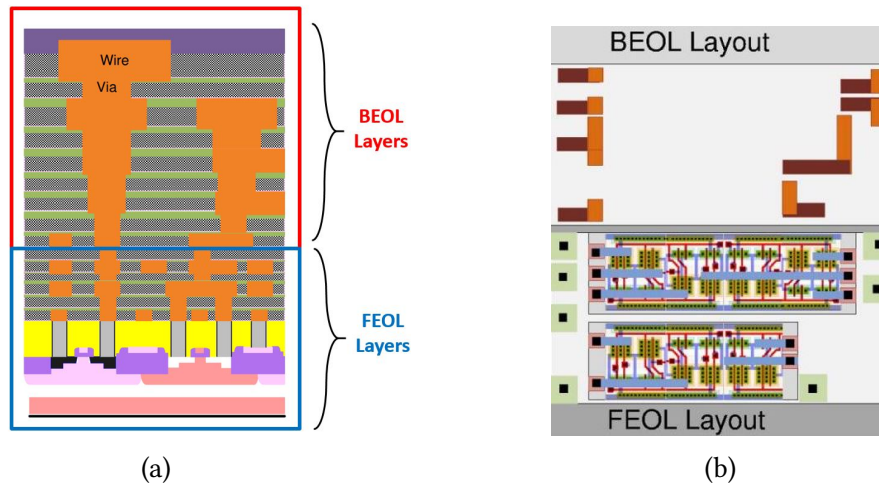


Figure 2.1: Partitioning of an IC for split manufacturing, where a) the cross-sectional view and b) the top view are shown [102].

2.1.1 Threat Model

An untrusted foundry is assumed, where the foundry has access to the design of a circuit and possesses the necessary tools and skills to counterfeit and overproduce the IC from the provided GDS-II file(s). In addition, the circuit in the design stage is assumed to be trusted and devoid of malicious components. Split manufacturing protects against a malicious foundry but does not prevent the end user from pirating the IC. Another critical security concern with split manufacturing, specifically for analog circuits that utilize

older technology nodes, is the amount of information leaked by the FEOL layers [101], which are provided to the untrusted foundry. An untrusted foundry can steal analog IC intellectual property with information leaked from the FEOL layers and knowledge of EDA algorithms.

2.1.2 Split Manufacturing Methodologies for Analog Circuits

The advantage of split manufacturing for analog circuits is that the untrusted foundry fabricating the FEOL must first identify the missing components found in the BEOL layers and then determine the respective values of each component to completely reverse engineer the design of the IC. To determine the trade-off between the increased security provided by split manufacturing and the additional design overhead needed to implement split manufacturing, three fabrication scenarios are analyzed in [101]:

- Scenario I: Removal of the top metal layer, which includes removal of inductors from the FEOL. An untrusted foundry is, therefore, not provide information on both the wire connections through the top metal layer and the inductor sizes,
- Scenario II: Removal of the top two metal layers, resulting in the removal of inductors and capacitors from the FEOL layout along with any wire connections through the top two metal layers, and
- Scenario III: Insertion of non-functional rings and the creation of empty zones in

the original design of the circuit to obfuscate the location, count, and sizes of the passive components of an analog IC.

To demonstrate the application of the split manufacturing technique and to characterize the security provided by each of the three scenarios, split manufacturing was implemented on a class AB amplifier and a class E power amplifier, the schematics of which are shown in Fig. 2.2. The technique was applied in a 180 nm TSMC technology node consisting of six metal layers. The Class AB amplifier operates at 5.8 GHz at a supply voltage of 1.9 V and was designed to provide 19.8 dBm of output power and 28.1% of power-added efficiency. The Class E power amplifier was designed to operate at 5.2 GHz and provide 12.5 dBm of output power and 25% of power added efficiency.

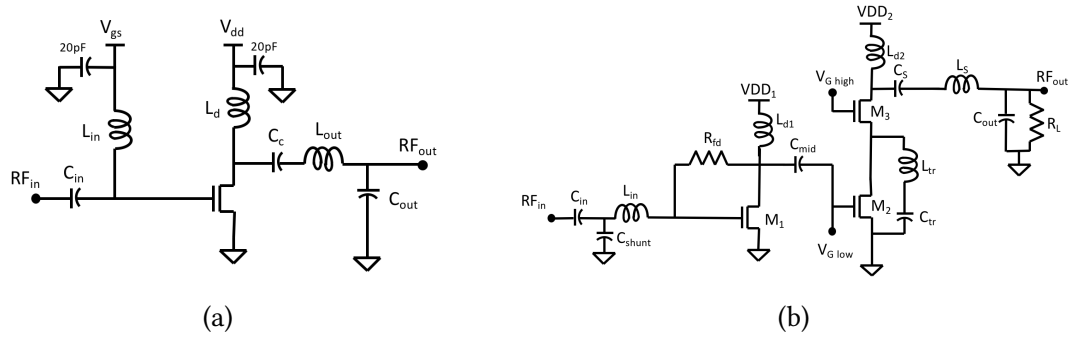


Figure 2.2: Schematics of a) Class AB amplifier and b) Class E power amplifier [101].

2.1.3 Scenario I: Removal of the Top Metal Layer from FEOL Foundry

Metal 6, the top metal layer of the 180 nm TSMC node, is removed from the FEOL, which obfuscates information on the implemented inductors, as shown in Fig. 2.3. Although the technique reveals the locations of the inductors, the size and structure are not known, requiring an adversary to follow a two-step procedure to recover the design of an analog IC as described in [101]:

- Step 1: Provided an adversary has access to the input-output ports, the operating conditions of an analog circuit, including the supply voltage, bias voltage, and operating frequency, are determined, respectively, by
 - applying at least 23 different voltages with a step size of 0.1 V in a range of 1 V to 3.3 V to determine the supply voltage of the 180 nm TSMC node,
 - applying at least 13 different voltages in a range of 0.4 V to 1 V and with a step size of 0.05 V to determine the gate biasing voltage, and
 - applying at least 53 different frequencies with a step size of 100 MHz to determine the operating frequency of the amplifier, which is in the range of 800 MHz to 6 GHz (commercial communication protocol range).

Although sweeping the inputs and observing the corresponding outputs is simple to perform, the original design is difficult to recover just by choosing the results

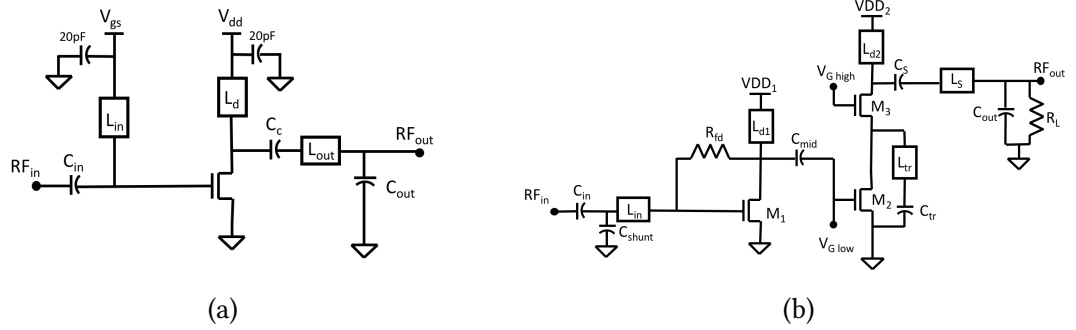


Figure 2.3: Circuit schematic with missing inductor information due to removal of metal 6 for a) Class AB amplifier, and b) Class E power amplifier [101].

that produce the best output power or best power efficiency. Specifically in the case of an AB amplifier, the desired output power and power efficiency are a challenge to obtain by sweeping the supply voltage and gate voltage as the maximum output power does not coincide with the maximum power efficiency, which are shown in Fig. 2.4. In addition, the 5.8 GHz target operating frequency of the amplifier coincides with a gate bias voltage of 1 V, which is located in an area of low power-added efficiency, as shown in Fig. 2.5. Therefore, finding the correct parameter settings when executing Step 1 of the circuit recovery procedure is highly challenging.

- Step 2: If the correct operating conditions are determined, an adversary must then deduce the biasing conditions to obtain the exact inductor size by
 - analyzing the input reflection coefficient S_{11} for different RF choke and input inductor values to determine the operating frequency of the amplifier,
 - based on the operating frequency, determine the range of suitable inductor

values, and

- optimize the output performance of the amplifier by tuning the RF choke and output inductor values.

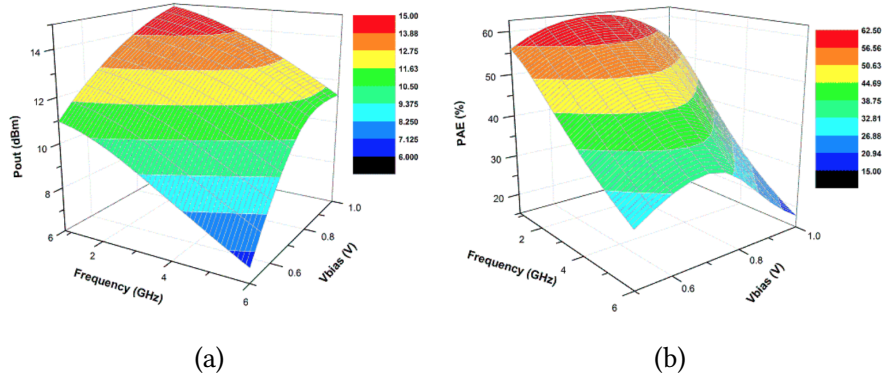


Figure 2.4: Effect of supply voltage and gate biasing on a) output power, and b) power-added efficiency of an AB amplifier [101].

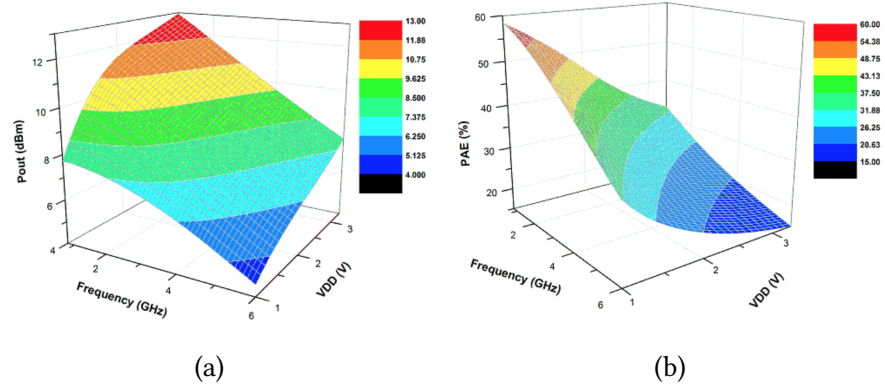


Figure 2.5: Effect of gate biasing and frequency on a) output power, and b) power-added efficiency of an AB amplifier [101].

For a class-E amplifier, where a voltage range of 0 V to 3.3 V for V_{DD1} , a voltage range of 0 V to 2 V for V_{DD2} , and voltage step size of 0.1 V are assumed, an attacker

must select from 34 voltages and 21 voltages when setting V_{DD1} and V_{DD2} , respectively. Provided an adversary determines the correct DC bias, gate biases, and supply voltage, a multi-parameter sweep of all the inductors is required to determine the exact operating frequency, which is a challenge. Therefore, Scenario I is effective in masking the inductors from an untrusted foundry fabricating the FEOL of an analog circuit.

2.1.4 Scenario II: Removal of Top Two Metal Layers from FEOL Foundry

For Scenario II, both the inductors and the capacitors are removed from the GDSII information provided to the untrusted foundry as shown in Figure 2.6, by not including both metal 5 and metal 6. The removal of capacitor and inductor information dramatically increases the search space needed to recover the design of an analog IC. The procedure to determine the capacitor and inductor values that results in the desired output performance of the amplifier includes five primary steps:

- Step 1: Same as Step 1 of Scenario I described in Section 2.1.3.
- Step 2: Based on the determined DC operating point and operating frequency, the inductor size for the RF choke and output coupling capacitor are obtained by sweeping through the inductor and capacitor values allowed by the process. For the 180 nm TSMC technology node, permitted, on-chip inductors and capacitors range in value from 0.5 nH to 5 nH and 1 pF to 10 pF, respectively. Assuming a step size of

0.1 nH and 0.1 pF for the inductors and capacitors, respectively, there are 45 possible inductor values and 90 possible capacitor values. The effect on output power and power-added efficiency when the inductor and capacitor values are swept is shown in Fig. 2.7. An attacker compares against the characterized output power and power-added efficiency of an active IC to recover the correct values of the inductors and capacitors.

- Step 3: Once the RF choke and the coupling capacitor are selected, output matching to $50\ \Omega$ is performed using load-pull simulations to determine the matching network.
- Step 4: Similarly, a source pull simulation is used to determine the input matching network.
- Step 5: Final tuning of the inductors and capacitors is needed to recover the analog IC design completely.

2.1.5 Scenario III: Insertion of Non-Functional Rings and Creation of Empty Zones

In order to minimize the deviation in output performance (output power and power-added efficiency) and to alleviate additional design costs, two obfuscation methodologies are proposed in [101]:

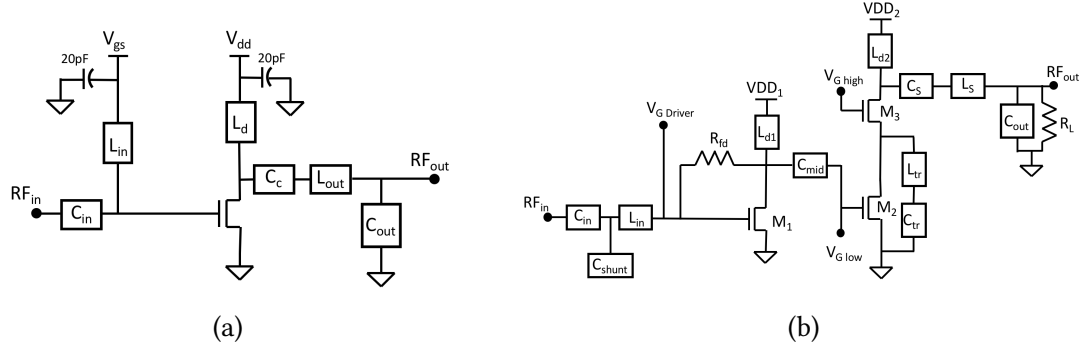


Figure 2.6: Schematics of a) Class AB amplifier, and b) Class E power amplifier with missing capacitor and inductor information due to the removal of metals 5 and 6 resulting information [101].

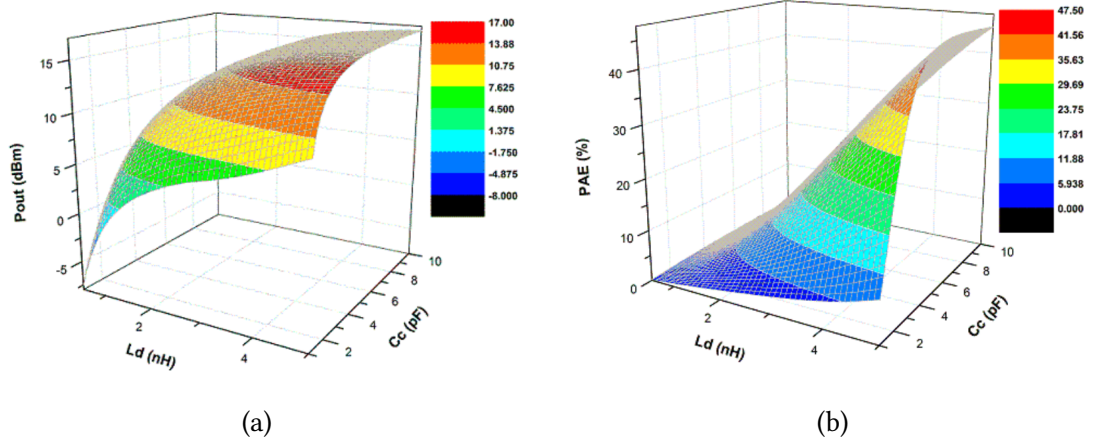


Figure 2.7: Effect of sweeping the inductor (RF choke) and output coupling capacitor on the a) output power and b) power-added efficiency of an AB amplifier [101].

- the addition of dummy capacitors and inductors to obfuscate the size and locations of the passive components, and
- the addition of unused spaces to obfuscate the output and input matching structures.

To address information leakage due to empty spaces, a methodology to increase the occupied area of passive components is proposed [101]. The method involves the addition

of dummy capacitors and inductors, which increase the area occupied by the passive components [101]. The second technique utilizes unused empty blocks in the original circuit to obfuscate the correct circuit structure. Both obfuscation techniques involve modification of the original layout. To address the complexity of obfuscating the original circuit, a new RF co-design methodology is proposed that considers security during the analog IC design flow [101]. The circuit deliberately includes both sub-optimal components and optimally designed components, which provides a challenge in determining the value of passive components.

Although split manufacturing [101] provides some security when protecting an analog circuit from a malicious foundry, the technique does not prevent an end user from pirating the IC as all FEOL and BEOL layer information is available after reverse engineering the circuit. Another critical security concern with split manufacturing is the amount of information leaked by the FEOL layers [101], which are provided to the untrusted foundry. With the provided FEOL layers and knowledge of EDA algorithms, an untrusted foundry can potentially steal analog IC intellectual property using a proximity attack [102, 103]. As the circuits are designed to optimize power, performance, and area, the connected circuit components tend to be placed in close proximity of one another to reduce area, wirelength, and delay, which allows for the possible identification of missing connections between the FEOL and BEOL by examining the input and output pins that are located close to each other [102, 103].

2.2 IC Camouflaging

Camouflaging obfuscates a circuit by modifying the physical layout of the IC in order to mask the actual functionality. IC camouflaging techniques include the addition of dummy metal fill, dummy vias, dummy contacts, and dummy devices, which obfuscate the layout of the circuit. IC camouflaging is implemented in digital circuits by inserting gates that exhibit the same topological structure as multiple logical functions. For example, the NAND and NOR gates depicted in Fig. 2.8a are designed to appear identical in structure. For a given gate placement, an adversary is not privy to whether the correct functionality is that of a NAND or NOR gate.

Another popular method to camouflage an IC is to utilize dummy contacts, as shown in Fig. 2.8b. When delayering the IC, an adversary only observes the top view of the metal and poly layers. Assuming the delayering and imaging process provides a limited resolution, an adversary cannot discern between a true contact and a dummy contact Fig. 2.8a.

Similar to the digital camouflaging technique proposed in [104], the analog camouflaging technique proposed in [105], utilizes the fake contacts proposed in [106] to obfuscate the size of the critical transistors of an analog and mixed signal (AMS) circuit. True contacts connect two layers (metal substrate) by spanning the entire dielectric, while fake contacts include a small gap of dielectric in the conductive path, which results in an open

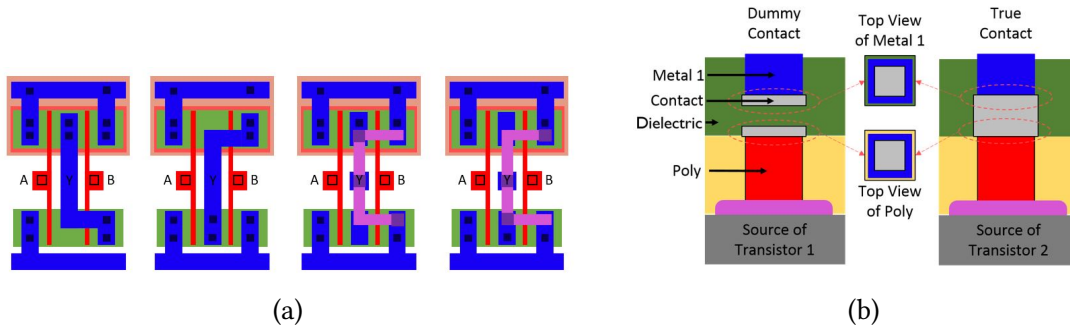


Figure 2.8: IC camouflaging of a gate with dummy contacts, where a depiction is provided of a) the top view of standard NAND and NOR gates and camouflaged NAND and NOR gates, and b) the cross-section of dummy and true contacts [104].

circuit. Fake contacts are implemented without any changes to the IC fabrication process and are CMOS compatible [106]. As both the true and the fake contacts look identical under a microscope, imaging the die will not reveal the gap, which does not allow an attacker to distinguish between true and fake contacts.

2.2.1 Threat Model

IC camouflaging provides defence against IP attacks from a malicious end-user. The threat model proposed in [105] assumes that a malicious end-user possesses a functional IC, has access to the technology design kit (PDK), and has all the resources needed to completely reverse engineer the obfuscated IC and determine the schematic, and layout of the circuit. The attacker also possesses a second IC used as an oracle to determine the correct input-output responses of the circuit.

The camouflaging of an analog IC does not protect against a malicious system-on-chip (SoC) integrator or an IC fabrication company, as all necessary information needed to manufacture the IP/IC including the GDS-II file is shared with the integration and fabrication facilities. Therefore, fake contacts are directly identified by the potentially untrusted integration and fabrication facilities.

2.2.2 Camouflaged Analog Components

Analog IC camouflaging obfuscates the circuit by utilizing fake contacts to hide the active geometry of layout components and, therefore, mask the true size of devices including transistors, capacitors, and resistors [105].

2.2.2.a Camouflaged Transistor Layout

In analog circuit layout, the non-minimum sized transistors are typically implemented as sub-transistors connected in parallel and with shared diffusion strips, which results in multiple gate fingers. The parallel transistors of a multi-finger device are typically equal in gate width, with each transistor sharing the inner diffusion regions of the drain or source with two neighbouring transistors. A transistor is obfuscated by including extra gate fingers and deactivating certain transistor connections by employing fake contacts in the drain, source, or gate terminals, which results in a floating node as shown in Fig. 2.9a.

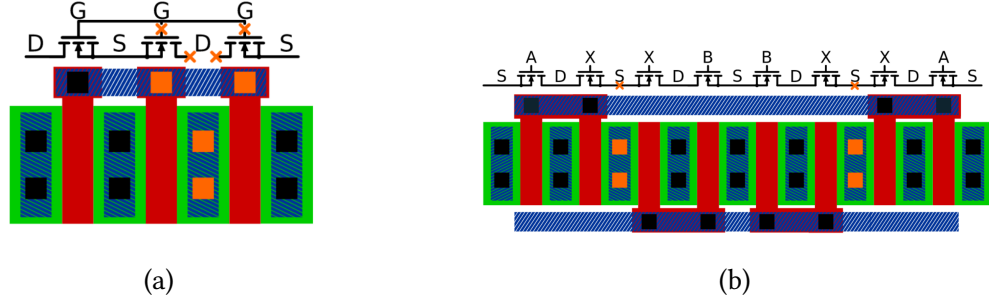


Figure 2.9: Depiction of camouflaged multiple gate-finger transistor schematics and layouts, where a) two fake contacts are utilized to deactivate two gate fingers and drain contacts and b) fake contacts are used in the layout of a common-centroid differential transistor pair to deactivate transistors. Diffusion, polysilicon, and metal are shown in green, red, and blue, respectively. True contacts are depicted in black and fake contacts in orange [105].

Another use of camouflaged transistors is in the layouts of differential pairs or current mirrors. Common-centroid layouts are typically utilized for differential transistor pairs that ensure variations across the die impact both transistors in a similar manner [107]. The camouflaged transistors replace differential transistor pairs, obfuscating the number of active transistors as shown in Fig. 2.9b. From Fig. 2.9b, the attacker observes that transistors A and B consist of four active transistors each, while the true number of active transistors is only two each.

2.2.2.b Camouflaged Capacitor Layout

The total capacitance of a capacitor bank is obfuscated by adding additional capacitor units and disconnecting a subset with fake contacts, as shown in Fig. 2.10a. Metcap and metal2 (M2) are the respective plates of the capacitor. The capacitor on the right half of

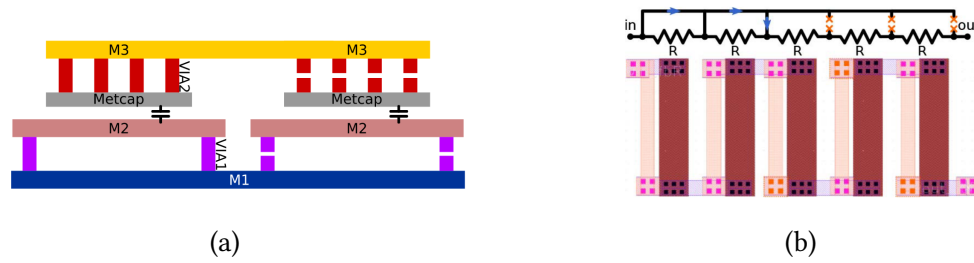


Figure 2.10: The layout of the camouflaged capacitor and resistor using fake contacts. a) The obfuscated capacitor on the right has fake contacts connecting its capacitor plates and obfuscating the overall capacitance of the capacitor bank [105]. b) The obfuscated serpentine resistor layout. Poly, metal 1, metal 2, true contacts, fake contacts, and metal1-metal2 vias are shown in dark red, blue, light orange, black squares, orange squares, and pink squares.[105].

Fig. 2.10a is disconnected from the bank using fake contacts, which are shown with a thin gap in the material connecting metals 1 and 2. From the perspective of the attacker, the capacitor bank consists of two capacitors connected in parallel, which obfuscates the overall capacitance.

2.2.2.c Camouflaged Resistor Layout

The layout of a serpentine resistor is obfuscated by adding additional unit resistors, which are deactivated with fake contacts to obfuscate the overall resistance. The use of a camouflaged resistor is shown in Fig. 2.10b, where a serpentine resistor composed of five unit resistors is depicted. The proposed technique uses wiring to create short circuits across each unit resistor. The placement of fake contacts is utilized to remove the short circuit from the unit resistors to activate. In contrast to the use of fake contacts to deactivate transistors and capacitors, fake contacts in camouflaged resistors are used to

activate unit resistors. For the example shown in Fig. 2.10b, the nominal resistance is $3R$, whereas the attacker observes a resistor value $k \cdot R$, where the value of k is any value between 1 and 5.

2.2.3 Case Study

The proposed AMS camouflaging techniques are implemented on a Miller op-amp and an RF $\Sigma\Delta$ ADC as shown in Fig. 2.11a and Fig. 2.11b, respectively [105]. For the Miller op-amp, the biasing transistor M1, the current mirror transistors M4 and M5, and the feedback capacitor C are obfuscated. M1 is placed (layout) as a multigate-finger transistor with 20 fingers, out of which ten are inactive. Transistors M4 and M5 are placed in an interdigitized pattern, with each including 12 gate fingers, of which eight are inactive. Capacitor C is implemented as a capacitor bank with four unit capacitors, of which one is inactive. The obfuscated layout resulted in an overhead in area of 15%. An attacker considering all contacts observes at least an 18 dB degradation in gain, a 5.1% increase in the total harmonic distortion (THD), and a 6.4% increase in the DC power consumption [105].

The pre-amplifier and the comparator stages of the RF $\Sigma\Delta$ ADC are obfuscated as shown in Fig. 2.11b. Two differential transistor pairs and a resistor are obfuscated in the pre-amplifier stage. In addition, a differential transistor pair in the latch of the comparator is also obfuscated. An attacker considering all contacts observes the signal-to-noise

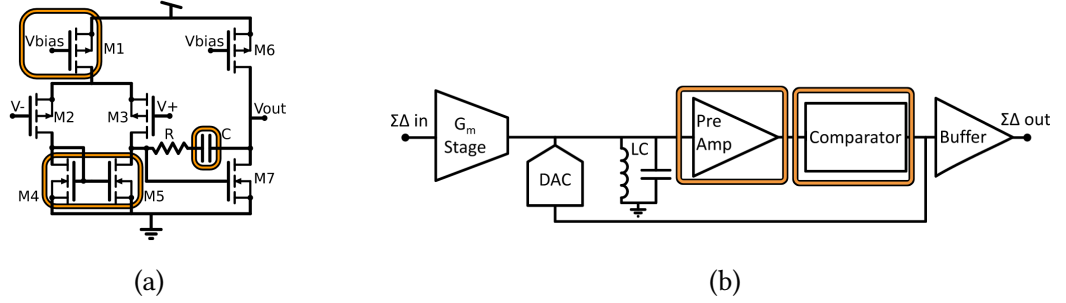


Figure 2.11: Schematic and block-level representation of the a) Miller op-amp and b) RF $\Sigma\Delta$ ADC, that are used as test circuits to validate the performance of the proposed AMS camouflaging techniques [105].

ratio (SNR) dropping below 0 dB, which results in the signal being within the noise floor.

Although the proposed AMS camouflaging techniques provide defense against IP attacks from an untrusted end-user, the technique does not provide defense from an untrusted SoC integration and fabrication facility, which require sharing of the IP/IC, including the GDS-II files, for design and manufacturing. In addition, the camouflaging defense is susceptible against satisfiability modulo theory (SMT) based attacks [108] and focused ion beam (FIB) probing [105].

2.3 Key-based Locking

The technique requires the application of a key to lock the functionality of a circuit. Only when the correct key is applied is the functionality of the obfuscated circuit correct. An incorrect key results in a degraded or incorrect functionality. The untrusted foundry or end-user is not provided the key unless released by the designer. The integrated circuit

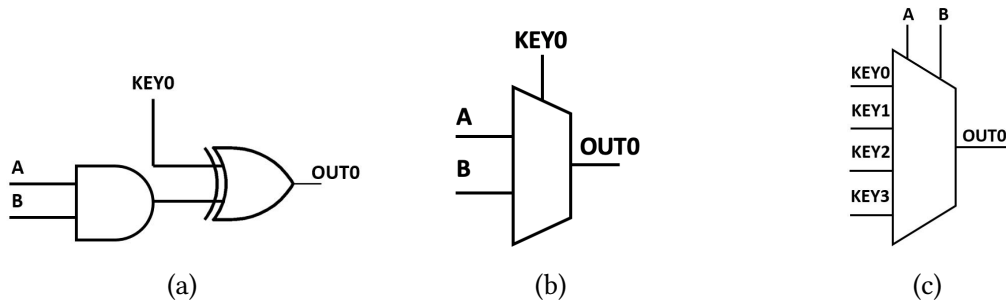


Figure 2.12: Block level representation of in-cone logic locking methodologies, including (a) XOR-based [109], (b) MUX-based [110], and (c) LUT based [111].

is not completely released to the foundry, and black-box usage is impossible without the key.

One example of a key-based locking technique is the logic locking, where the logic of the digital circuit is obfuscated using a key. Logic locking obfuscates the digital circuitry against a wide range of threat models including an untrusted foundry, an untrusted assembly facility, an untrusted testing facility, and untrusted end-user, thereby providing protection against IP theft, IC counterfeiting, overproduction, and hardware Trojan insertion. Three primary logic locking methodologies are proposed to secure the in-cone logic of a digital circuit. The three techniques include XOR [109], multiplexer (MUX) [110], and look-up table (LUT) [111] based locking.

2.3.0.a XOR Logic Locking

Logic locking is implemented by inserting an XOR/XNOR gate after a selected gate within the netlist, with a key bit as the second input to the XOR/XNOR gate. An example

of locking an AND gate with an XOR gate is depicted in Fig. 2.12a. When KEY0 is 0, the XOR gate acts as a buffer, whereas when KEY0 is 1, the XOR gate acts as an inverter [109]. The KEY0 input to the XOR gate controls the output OUT0, which only permits correct functionality when the correct key input, is applied. The same concept applies to an XNOR with only a simple inversion to the key input of the gate.

2.3.0.b MUX Logic Locking

Locking is implemented by applying a key to the select inputs of a MUX, as shown in Fig. 2.12b for a 2x1 MUX. The *A* input in Fig. 2.12b is the true net, while *B* is a “false” input from another gate of the circuit. Compared to the XOR-based technique, the inherent benefit of the 2x1 MUX locking technique is that no additional inverters are needed to mask the key bits [110].

2.3.0.c LUT-Based Logic Locking

LUT-based logic locking replaces a gate within the original netlist with the 4x1 MUX structure shown in Fig. 2.12c, which allows for the implementation of any 2-input function. The obfuscated gate is implemented by setting the key inputs to the binary values of the logical function being masked. If, for example, KEY0, KEY1, and KEY2 are set to logic 0, and KEY3 is set to logic 1, then an AND gate is realized.

The LUT-based locking methodology provides two significant benefits over the XOR and MUX-based methodologies. First, the key space is larger as each locked gate requires

more than one key bit. The more extensive key space increases both the number of key combinations an adversary must attempt and the input-output combinations needed to deduce the correct key combination. Second, the original gates in the circuit are replaced by the reconfigurable LUTs [111]. Replacing the original gates offers a distinct advantage over XOR-based logic locking, as a portion of the gates are replaced with unknown functions, in contrast to XOR-based locking, which adds an XOR gate to the circuit but does not remove the original gate.

2.3.1 Key-based Locking Techniques to Protect Analog IP

Similar to the digital locking techniques, analog locking techniques obfuscate analog circuit performance using a key. However, unlike digital locking, which only masks the output bits, analog locking techniques mask a wide range of performance parameters including gain, bandwidth, gain-bandwidth product, total power, oscillating frequency, center frequency, cut-off frequency, the third-order input intercept point, and the stability of an analog circuit. The locked analog circuit provides performance parameters that match the specifications only when the correct key is applied. Incorrect keys result in one or more performance parameter(s) that do not match the circuit specifications, which leads to a degradation in the overall performances of an analog circuit. Initial work on analog obfuscation utilized digital key-based locking techniques to obfuscate the digital components of an analog circuit, including the bias and PVT tuning circuits.

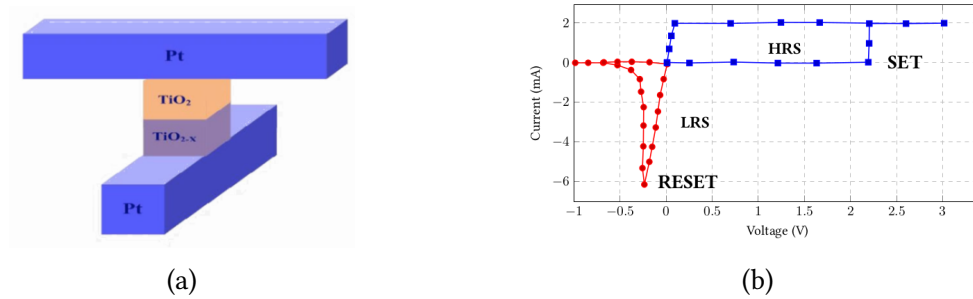


Figure 2.13: The (a) structure of a metal-insulator-metal (MIM) memristor and (b) characterization of a MIM memristor indicating the HRS and LRS states and the corresponding V_{SET} and V_{RESET} voltages needed to change the states of the memristor [112].

2.3.1.a Key-based Locking of a Locking Sense Amplifier

A key-based locking of a sense amplifier is proposed in [112]. The technique utilizes a memristor-based voltage divider to bias the body voltage of the input transistors of the sense amplifier to mitigate the effects of process variations. A memristor crossbar structure is used to program the voltage divider circuit, where the crossbar is configured correctly only when the correct 16-bit key is applied. An incorrect key results in the breakdown of the circuit.

The metal-insulator-metal (MIM) memristor shown in Fig. 2.13a includes at least two primary resistive states: a high resistance state (HRS) and a low resistance state (LRS) [112]. The appropriate voltage and polarity of bias voltages V_{SET} and V_{RESET} is applied to change the memristor states as shown in Fig. 2.13b. V_{SET} is applied to change a memristor from the HRS to the LRS and is termed a SET operation. V_{RESET} is applied to change the state of the memristor from the LRS to the HRS and is termed a RESET operation.

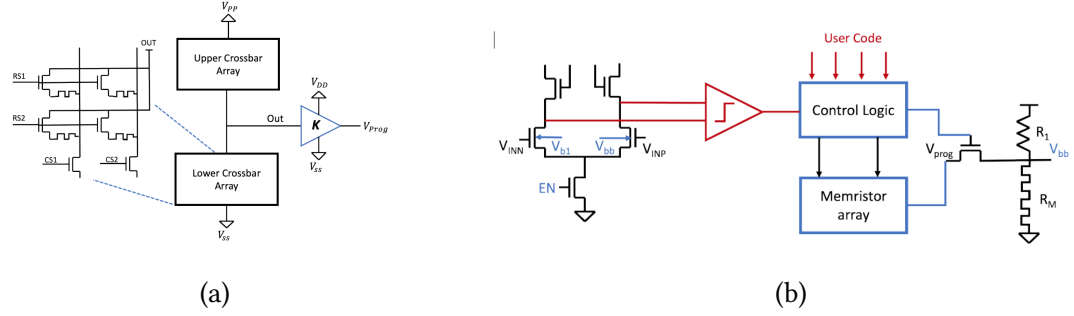


Figure 2.14: Block-level representation of the (a) memristor cross-bar structure based on a voltage divider circuit and (b) application of the memristor cross-bar structure to tune the bias voltages of the differential pair transistor of the sense amplifier, which minimizes the mismatch from process variations [112].

A memristor crossbar structure is implemented as a voltage divider circuit as shown in Fig. 2.14a that outputs a bias voltage based on the memristor divide ratio. A 16-bit key is applied to control the V_{SET} and V_{RESET} voltages, which sets the resistance of the memristors and the divide ratio of the memristor crossbar. The use of a memristor crossbar array to tune the bias voltages of the sense amplifier is shown in Fig. 2.14b. The memristor array sets the value of memristor R_M such that the correct body voltage V_{bb} is provided to the inputs of the sense amplifier to minimize the mismatch due to process variations. The resistance of the memristor R_M is initially set to LRS, resulting in a body voltage V_{bb} to a value lower than the target. Programming pulses V_{prog} are applied, whose voltage is equal to the V_{RESET} voltage of the memristor, that change the state of target memristors of the memristor array to the HRS, thereby increasing the value of R_M . The increase in memristor resistance R_M increases the body voltage V_{bb} , which is increased until the comparator indicates a zero mismatch. The crossbar structure is correctly configured only

when a correct key is applied, which results in a minimal mismatch in the differential pair of the sense amplifier. The application of an incorrect key results in an improper configuration of the crossbar structure and, therefore higher mismatch in the differential pair of the sense amplifier.

The proposed key-based locking utilizing a memristor array protects the sense amplifier from an untrusted foundry and an untrusted end-user. However, the technique only applies to analog circuits that utilize sense amplifiers. In addition, the proposed technique suffers from significant area overhead and limited options to fabricate memristors, which require a unique manufacturing process as compared to traditional silicon fabrication.

2.3.1.b Logic Locking the PVT Tuning Knob of AMS Circuits

A logic locking technique is proposed in [113] to obfuscate the process, voltage, and temperature (PVT) tuning circuitry of an AMS circuit. The primary motivation of the proposed technique is to intentionally degrade the performance of an analog circuit from the effects of process variations. The effects are mitigated only by applying the correct key, which results in the desired analog circuit performance. The application of an incorrect key results in the degradation of the performance of an analog circuit as the effects of process variations have not properly compensated for.

The proposed logic locking technique to protect analog circuits by obfuscating the digital PVT tuning circuitry is shown in Fig. 2.15. The tuning circuitry minimizes the

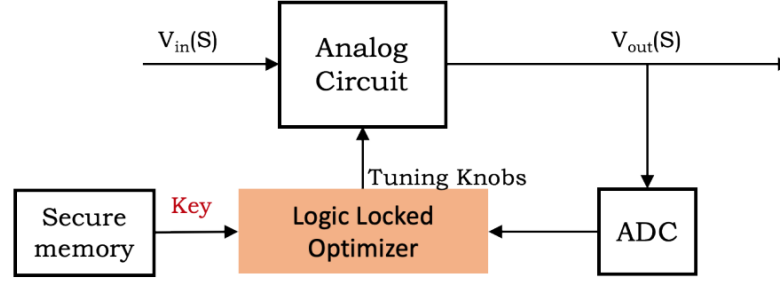


Figure 2.15: Block level representation of the logic locking technique to obfuscate PVT tuning circuitry of the AMS circuit proposed in [113].

effect of process variation by setting the correct values of the passive components of the analog circuit including the resistors and capacitors to properly set the body bias voltage of the transistors. The digital components of an analog circuit are responsible for determining and settings the optimal configuration of the tuning knobs that compensate for PVT variation.

2.3.1.c Logic Locking the Digital Sub-block of AMS Circuits

The logic locking technique proposed in [114] obfuscates the digital section of an analog and mixed-signal circuit by utilizing the stripped functionality logic locking (SFLL) method proposed in [115]. The objective of the proposed obfuscation technique is to convert the original digital sub-block of the AMS circuit to one that is functionally equivalent using a secret key that unlocks the correct functionality. Application of an incorrect key results in severely degraded AMS circuit performances.

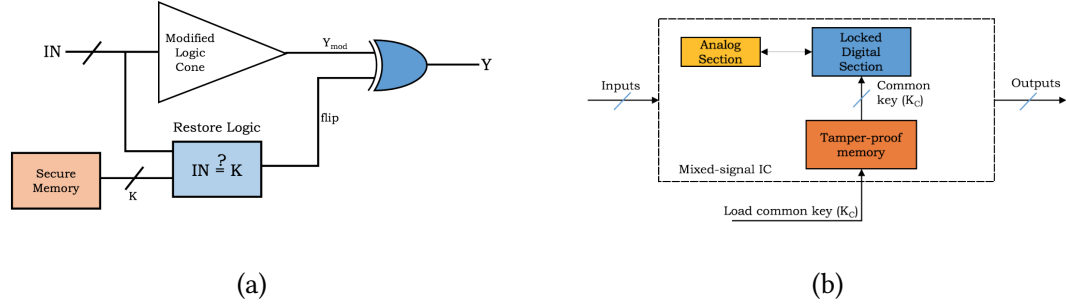


Figure 2.16: Block level structure describing the architecture of (a) stripped functionality logic locking technique [115] and (b) application of SFLL to obfuscate the digital sub-block of an AMS circuit [114].

The block-level representation of the SFLL technique [115] is shown in Fig. 2.16a. With SFLL, the output of the original circuit is flipped for all the input patterns that are h hamming distance away from the secret key. When a correct key is applied, the *restore logic* unit cancels the errors introduced by the *modified logic cone* and recovers the original circuit output for the particular input pattern. The SFLL technique protects against a wide range of IP piracy attacks, including reverse engineering, overproduction, and counterfeiting. In addition, the SFLL technique is also resilient against the SAT [116], removal [117], and approximate attacks [118] that are used to de-obfuscate digital locking techniques. The SFLL technique is therefore, widely employed to obfuscate digital circuits.

The block-level representation of the proposed SFLL-based obfuscation of the digital sub-blocks of an AMS circuit is shown in Fig. 2.16b. The original digital portion of the analog-to-digital converter (ADC) is modified, and the correct functionality of the sub-block is restored using the *restore logic* block. The application of an incorrect key results in significant degradation in the signal-to-noise ratio (SNR) and the frequency response

of the ADC [114].

The logic locking techniques proposed in [113, 114] secure an AMS circuit against an untrusted foundry and an untrusted end-user. In addition, the logic locking techniques do not require resizing the analog circuit, which results in faster design times. The logic locking of AMS circuits also provides resilience from traditional digital SAT attacks [116], as extensive use of the testing scan-chains is needed but is limited in AMS circuits [114]. However, the logic locking techniques only apply to large AMS circuits that include digital components and do not apply to pure analog circuits. In addition, logic locking techniques for AMS circuits [113] result in a significant increase in the physical size of the circuit, particularly for small AMS circuits, as a large number of passive components are required to obfuscate the circuit with a sufficiently large key size. In recent years, the leakage of information between the analog and digital sub-blocks of an AMS circuit has been utilized to circumvent the logic locking techniques, resulting in a vulnerability to SAT and SMT-based attacks [119].

2.4 Unique Key Generation and Storage Using Physically Unclonable Functions (PUFs)

The rise of key-based obfuscation to protect integrated circuits against IP piracy attacks has emphasized the need for secure key generation and storage. A physically unclonable function (PUF) serves as a security primitive at the chip level and functions as a physical root of trust [120] to enable various security operations including secure key generation, storage, and hardware authentication [121]. The unclonable nature of the PUF is achieved by hiding the relationship of the challenge-response pairs (CRPs) into the intrinsic physical parameters of the circuit, resulting in a very low probability of an adversary generating an exact copy of a PUF instance [122].

When an external stimulus (challenge) is applied to a PUF, the PUF produces an output response. The response to a challenge depends on the die specific process variations present in the PUF circuit. Stated differently, each PUF response is a function of both the applied challenge and the intrinsic variations of the circuit. Depending on the type of PUF implemented, the PUF is able to provide a response to a single challenge, a few challenges, or an exponential number of challenges [120]. A weak PUF provides a single CRP or CRPs that increase linearly with circuit parameters including area and the number of transistors [123]. A strong PUF allows for CRPs that increase exponentially in number with the circuit parameters utilized by the PUF [123]. In addition to the number of CRPs,

PUF instances are also evaluated for the following metrics:

1. Uniqueness: each PUF instance must generate a unique response to a challenge,
2. Unpredictability: observations of the CRPs must be exceedingly high to predict,
3. Unclonability: PUF instances must be exceedingly costly to clone even when the circuit designs are fully disclosed,
4. Robustness: the response for a particular challenge must be the same under noise, process, temperature, supply, and aging variations, and
5. Compactness: the PUF instance must have minimal impact on the footprint of an IC.

2.4.1 Strong PUFs

Strong PUFs were pursued first by the research community and include the arbiter PUF [124], power grid PUF [125], and optical PUF [126] as implementations. The arbiter PUF is implemented by creating two paths of similar length and dimension, where the delay of the signals passing through the two paths differs due to post-fabrication variations. An arbiter PUF determines the fastest of the two paths and generates an one bit output depending on which path is faster. The initial arbiter PUFs suffered from a lack of uniqueness and unpredictability as multiple CRPs are based on the delays of the same paths, which results in a correlation between the challenges and the corresponding PUF

responses. Modeling-based attacks utilize the correlation between path delays to accurately predict the output of the arbiter PUF for a given challenge [127].

To improve the uniqueness and unpredictability of the arbiter PUF, a delay-based PUF consisting of ring oscillators is proposed [128], where two ring oscillators are selected for a challenge based on the oscillating frequencies, and one bit output is generated depending on which oscillator produces a higher frequency. The reduced number of challenge-response pairs from the delay-based PUFs results in improved uniqueness as compared to arbiter PUFs. However, the robustness of the PUF is degraded due to the effects of process variation on the ring oscillators.

To improve the robustness of the delay-based PUFs, a bi-stable ring PUF was developed [129, 130]. The bi-stable ring PUF leverages the two states of a ring oscillator when utilizing an even number of inverters. The bi-stable PUF is robust to temporal noise, as the noise is averaged during the stabilization process, where the PUF bit settles to either 0 or 1. In addition, the settling time of the bi-stable PUF is used to remove unreliable CRPs. Strong PUFs are utilized for applications that require a large number of challenge-response pairs. However, strong PUF suffer from a lack of unpredictability and are susceptible to modeling attacks.

2.4.2 Weak PUFs

A weak PUF requires a dedicated set of circuit components to generate each bit, resulting in higher unpredictability than strong PUFs. However, the number of CRPs generated by a weak PUF is limited. Requiring greater number of CRPs results in an increase in area and a reduction in the compactness metric. Most weak PUFs utilize digital circuits including static random access memory (SRAM) [131], a latch with reset [132], buskeeper cells [133], and cross-coupled inverters [134]. In recent years, analog circuits, whose performance is more susceptible to variations as compared to digital circuits, have been utilized in the implementation of PUFs. The input offset of a sense amplifier has been used to generate one bit PUF output [135]. In [136], a weak PUF is developed that leverages the mismatches between current mirrors that are implemented with either PMOS or NMOS transistors. A PUF based on pairs of analog circuits that output responses that are either proportional to absolute temperature (PTAT) or complementary to absolute temperature (CTAT) is proposed in [137]. Although analog PUFs occupy a greater area than digital PUFs, analog PUFs provide greater robustness and unpredictability as compared to digital PUFs [137] and are suitable for unique key generation and key storage.

Chapter 3

Satisfiability Modulo Theory

Satisfiability (SAT) is a fundamental computational problem in computer science and mathematical logic that involves determining whether a given formula is satisfied, or made true, by assigning appropriate truth values to the input variables [138]. With satisfiability, the objective is to determine whether a formula that represents a constraint has a solution or a corresponding representative model. A broad range of problems are framed in terms of satisfiability, including Sudoku puzzles, graph problems, planning, scheduling, software and hardware verification, optimization, extended static checking, and test case generation [138]. The most popular constraint solving problem is *propositional satisfiability* or *Boolean satisfiability* (SAT). The formulated SAT problem is to determine whether a Boolean logical formula is satisfied. If the logical formula is satisfied, then the SAT solver returns the set of input values that resulted in the satisfied condition [139].

Not all problems are solved purely with Boolean functions and, therefore, require other computational methods including equality, uninterpreted functional symbols, quantifiers, arithmetic, arrays, and datatype operators. Such problems are formulated using first-order logic, which utilizes logical connectives (negation, conjunction, implication, and biconditional operators), variables, both universal and existential quantifiers, functions, and predicate symbols. First-order logic problems are solved with satisfiability modulo theory (SMT), where a solution or a model is constructed by using the variables, functions, and predicate symbols that result in a true outcome [139]. In recent years, SMT has gained traction due to the ability to solve non-Boolean problems including decision, completeness or incompleteness of logical statements, and complexity theory [140]. While SAT utilizes propositional logic, where problems are constructed solely with variables, SMT extends the problem formulation to include reasoning with variables for linear arithmetic, bit vectors, and arrays. One notable theory utilized by SMT is linear arithmetic, which is the foundation of linear programming and provides utility in many practical applications. Linear programming algorithms are capable of verifying the satisfiability of conjunctions of linear arithmetic inequalities, but are not directly applicable to Boolean combinations.

The SAT algorithm is a fundamental component of the SMT algorithm. To solve an SMT problem, the input formula, which includes variables from multiple domains, is first translated into an equivalent Boolean formula through a process described as Boolean

encoding [140]. The resulting Boolean formula is then solved using a SAT solver to determine satisfiability. As a result, an understanding of SAT algorithms is crucial for understanding SMT algorithms. Therefore, this chapter provides an overview of both the SAT and the SMT algorithms. In addition, the application of the SMT algorithm to solve for the transistor sizes of an analog circuit that meet the target specifications is also described in this chapter.

3.1 Satisfiability Algorithm

Modern SAT algorithms are based on a procedure proposed by Davis and Putnam (DP) [141] and later improved by Logemann and Loveland which led to the derivation of the Davis-Putnam-Logemann-Loveland (DPLL) method [142]. DPLL is a backtracking search algorithm for determining the satisfiability of a propositional formula in conjunctive normal form (CNF). The algorithm works by successively guessing assignments for unassigned variables, and propagating the consequences of the partial assignment with respect to the clauses. If a conflict is detected in the form of a false clause, the algorithm backtracks to the most recent decision level, undoes the assignments made at that level, and assigns a new guess to the unassigned variables.

The initial step of the DPLL algorithm is to convert the problem to conjunctive normal form, which is a standard representation of logical formulas that utilizes propositional logic. In CNF form, a formula is represented as a conjunction (AND logical operation) of

one or more clauses, where each clause is a disjunction (OR logical operation) of one or more literals [143]. The initial problem formulation is converted to CNF form by introducing new variables that represent labels of each sub-formula. To convert a formula to clausal form, appropriate clauses are added to each new variable. For example, to convert the formula $\neg p \vee (\neg q \wedge r)$ to clausal form, new variables b and a are introduced to represent $\neg q \wedge r$ and $\neg p \vee b$, respectively. The resulting clauses are a , $a \vee p$, $a \vee \neg b$, $\neg a \vee \neg p \vee b$, $b \vee q \vee \neg r$, $\neg b \vee \neg q$, and $\neg b \vee r$.

The input to the DPLL algorithm is a set of clauses K that represent the CNF formula. The algorithm generates a partial truth assignment for the variables in K by guessing an assignment for unassigned literals and then propagating the resulting changes of the partial assignment with respect to the clauses. If a conflict is detected in the form of a false clause, the procedure backtracks to the current partial literal assignment. The procedure terminates when an assignment that satisfies all clauses in K is found or when no such assignment exists. The search procedure is represented by a 4-tuple $\langle h, M, K, C \rangle$, that includes the decision level h , the partial assignment M , the input clause set K , and a set C of conflict clauses that are derived from K during the search.

The algorithm starts at decision level $h = 0$, with an empty partial assignment M_0 . At each decision level i , the algorithm selects an unassigned variable and assigns a logical value of either true or false. The process of determining, assigning, and propagating literals is defined as a branching decision, where the variable is the decision literal. The

algorithm then propagates the results of the partial assignment with respect to the clause, which results in a new partial assignment M_i . The propagation procedure involves two steps: unit propagation and literal elimination [142].

In unit propagation, the algorithm identifies clauses that contain only one unassigned literal and assigns the selected unassigned literal as true. The unit propagation procedure is repeated until no more unit clauses are found. The intuition behind unit propagation is that if a clause contains only one unassigned literal, the given literal must be true in any satisfying assignment [142]. In literal elimination, the algorithm identifies literals that occur only with the same polarity in the formula, and assigns the literals the corresponding polarity value. The literal elimination procedure is repeated until no additional literals are found. The intuition behind literal elimination is that if a literal occurs only with the same polarity, the literal is assigned that polarity in any satisfying assignment [142].

If a conflict is detected during propagation, the algorithm backtracks to the most recent decision level $(d - 1)$ and undoes the assignments made at the current decision level d . The algorithm then assigns the variable the opposite truth value from the previous assignment. The algorithm terminates when a truth assignment is found that satisfies each of the clauses, and the satisfying assignment is then returned. In contrast, if the algorithm determines that no such satisfying assignment is possible, then “*UNSAT*” (unsatisfiable) is returned.

The pseudocode of the DPLL routine is provided as Algorithm 1. The input to the

Algorithm 1: Pseudocode of the DPLL algorithm [144].

```

Input: CNF Formula
Output: UNSAT or SAT + Satisfying assignment
begin
    decision level  $\leftarrow$  0
    status  $\leftarrow$  UnitPropagate
    if status = CONFLICT then
        | return UNSAT
    end
    while TRUE do
        if no more free variables then
            | return SAT + Satisfying assignment
        end
        DecideNextBranch
        while TRUE do
            status  $\leftarrow$  UnitPropagate
            if status = CONFLICT then
                | if decision level == 0 then
                    | | return UNSAT
                end
            end
            blevel  $\leftarrow$  AnalyzeConflictAndLearn
            Backtrack(blevel)
        end
        else
            | break
        end
    end
end

```

algorithm is a set of clauses that represent the CNF formula. The algorithm maintains a decision level, which starts at zero and is increased with each branching decision. The decision level is decremented by one or more when backtracking is required. The *Unit-Propagate* procedure simplifies the formula by eliminating true clauses and false literals, which sets any resulting unit clauses to true, and repeats until no further simplifications

are possible. If an empty clause is encountered at decision level zero, the algorithm returns *UNSAT*, indicating that the formula is unsatisfiable. If all variables have been assigned without generating the empty clause, the algorithm returns *SAT* along with the satisfying assignment.

The *DecideNextBranch* function selects a free variable, assigns a value to the selected variable, and increments the decision level by one. However, instead of explicitly branching with both values of a variable, the algorithm utilizes clause learning to flip the value of the assigned variable upon backtracking. The decision level to which the algorithm backtracks upon encountering a conflict is determined dynamically using a graph-based analysis of the unit propagation steps that led to the conflict. The *AnalyzeConflictAndLearn* function performs the graph-based analysis that enables the algorithm to add a new clause that accounts for the current conflict being analyzed and prevents unnecessary future branching on variables that lead to a similar conflict [144].

3.1.1 Use of SAT Algorithms in VLSI

The ability of SAT solvers to efficiently resolve large-scale problems, including complex Boolean formulations, has led to widespread adoption of SAT algorithms in the design, modeling, and testing of VLSI circuits. SAT algorithms are primarily utilized in electronic design automation (EDA) for testing, verification, and optimization of digital circuits [145]. The use of SAT algorithms in VLSI include:

1. Automatic Test Pattern Generation (ATPG): SAT algorithms are used for ATPG to detect faults in digital circuits [146]. The ATPG technique involves formulating a Boolean satisfiability problem that is applied to a SAT solver to determine a test pattern that results in a fault. The SAT-based ATPG process begins with the generation of a fault model, which describes the types of faults that occur in the circuit. The fault model is typically based on stuck-at faults, which represent a signal stuck at a particular logic value (0 or 1) due to a fault [147, 148].

Once the fault model is defined, a set of candidate test patterns are generated for each fault in the circuit. The generation of candidate test patterns is done by formulating a SAT problem to find test patterns that activate faults, where the input variables are the unknown bits of the test pattern, and the output is the result of applying the test pattern to the circuit. The SAT solver finds a satisfying assignment for the input variables that activates the fault under test. The process is repeated for each fault considered, generating a set of test patterns that are used to detect the faults in the circuit [147].

SAT-based ATPG provides several advantages over traditional ATPG methods, including the ability to address complex circuits with a large number of inputs and the ability to generate high-quality test patterns in a relatively short amount of time. However, there are limitations to consider when utilizing SAT-based ATPG, which include the need for efficient fault models and the potential for the generation of

redundant or inefficient test patterns [147].

2. Equivalence checking: Equivalence checking is the process of verifying that two digital circuits or sub-circuits are functionally the same. Typically, the register-transfer logic (RTL) and the synthesized netlist of the circuit are functionally equivalent. SAT-based equivalence checking algorithms have been developed to verify the correctness of large circuits that are not easily verified using traditional methods. The technique is based on the principle that if two designs are functionally equivalent, then there are no inputs for which the two designs produce different outputs [149, 150]. The SAT-based equivalence checking methodology involves the following steps:

- (a) Symbolic representation: Both designs are first converted into a symbolic representation using Boolean logic gates that is utilized by a SAT solver.
- (b) Constructing a property: The next step is to construct a property that represents the functional difference between the two designs. Constructing a property refers to the process of formalizing and specifying the properties or behaviors of the two digital circuits or sub-circuits that are analyzed for equivalence. The specifications for constructing properties include functional correctness, timing constraints, and operating frequencies.
- (c) Encoding the problem: The property is then encoded into a form that is compatible with a SAT solver. Specifically, the Boolean expression is modified into

Conjunctive Normal Form (CNF), which is the standard input format required for input to a SAT solver.

- (d) Solving the SAT formulated problem: The CNF encoding of the property is then passed to a SAT solver, which determines an assignment of the input variables that satisfies the property. If the SAT solver returns *unsatisfiable* indicating that there is no input for which the two designs produce different outputs, then the two designs are deemed to be functionally equivalent. If the SAT solver returns "satisfiable" then there exists an input for which the two designs produce different outputs.
- (e) Counter-example generation: For the case that the SAT solver returns satisfiable, the returned inputs generated by the SAT solver provide a counter-example that demonstrates the functional difference between the two circuits.
- (f) Debugging and verification: The counter-example is used to debug the designs and to verify the correctness of the synthesized netlist. The SAT-based equivalence checking process is repeated until the two designs are deemed to be functionally equivalent for all possible input combinations [149].

3. Model checking: Verification that a model of a digital circuit satisfies a given set of properties is deemed as model checking. SAT-based algorithms have been developed to efficiently check the correctness of large digital circuits that operate in

different functional modes including at different clock frequencies and with different input/output configurations. The objective of model checking is to reduce the complexity of verifying the correctness of a circuit in all operating modes to a propositional satisfiability (SAT) problem [151].

The first step in SAT-based model checking is to identify the operating modes of a circuit and generate a Boolean formula that encodes the behavior of the circuit in each mode. Additional constraints are added to the Boolean formula that ensure that the circuit operates correctly in all modes. The resulting formula is then applied as input to a SAT solver, which determines whether the circuit violates the correctness constraints for any mode.

The additional added constraints depend on the nature of the circuit and the types of errors that must be detected. For example, if the circuit includes memory elements, constraints are added that ensure the state of the memory elements is correctly initialized and maintained for all modes of operation. If the circuit includes inputs that are not utilized in certain modes, constraints are added that ensure that the inputs do not affect the behavior of the circuit [152].

SAT-based model checking provides several advantages over other techniques when verifying circuits with multiple functional modes, which include

- the ability to check large and complex circuits with numerous modes,

- the ability of detect subtle errors that are often missed by traditional model checking techniques, and
 - the ability to return counter examples that help debug the circuit and highlight causes of error [152].
4. Logic synthesis: Logic synthesis is the process of automatically generating a logical circuit from a Boolean expression that is primarily represented as a logic equation or a truth table [153]. The objective of SAT-based logic synthesis is to generate a circuit that performs the desired function with the minimum possible number of logic gates and wires. For logic synthesis, a SAT solver is utilized to determine an efficient implementation of a circuit that satisfies a set of constraints, including minimizing the number of gates and reducing the delay [154].

SAT-based logic synthesis includes:

- Simplification of the Boolean Expression: First, a Boolean expression is simplified by applying various optimization techniques including algebraic simplification [153], Karnaugh maps [153], and the Quine-McCluskey algorithms [153, 154]. The resulting expression is in a form suitable for further optimization by a SAT solver.
- Generation of a Functionally Equivalent Circuit Representation: The next step is to map the Boolean expression to a functionally equivalent circuit, which is

typically in the form of a gate-level netlist. The circuit is represented as a graph, with nodes representing logic gates and edges representing wires.

- **Application of Circuit Optimizations:** Once the circuit representation is generated, various techniques are executed to optimize the logical gates and interconnects of the circuit. The objective is to reduce the number of gates and wires required to implement the circuit while maintaining the same functionality.
- **Verification of the Functionality of the Circuit:** The optimized circuit is then verified to ensure that the desired functionality is provided. Verification is performed by characterizing the output of the circuit with SPICE simulation, formal verification, model checking, and/or ATPG.
- **SAT-Based Logic Synthesis:** A SAT solver is then utilized to determine an efficient circuit implementation that satisfies all specified constraints of a Boolean expression. The SAT solver uses backtracking and constraint propagation to find a solution that satisfies the constraints. The output of the SAT solver is a minimal set of logic gates and wires that implement the desired function [153].

5. **Circuit optimization:** SAT algorithms are utilized to optimize the performance of digital circuits. The algorithms find an optimal assignment of parameter variables that minimize the circuit delay, power consumption, and/or area [155]. The primary objective is to model the circuit optimization as a Boolean satisfiability problem and

then use a SAT solver to find an optimal solution [155, 156].

The optimization problem, which also includes constraints based on the target objective(s), is typically formulated as a set of Boolean equations that describe the circuit behavior. For example, the area of the circuit is minimized subject to constraints on the performance and functionality of the circuit. The constraints are represented as Boolean formulas that encode the desired behavior of the circuit.

Once the optimization problem is formulated, a SAT solver is utilized to determine a satisfying assignment of all constraints. A satisfying assignment indicates an optimal solution to the optimization problem. The advantages of executing SAT-based circuit optimization include:

- a guarantee that the determined solution is optimal provided that the constraints are set correctly, whereas heuristic algorithms only provide approximate solutions,
- the ability to optimize with large circuits that include a large number of variables and constraints, which are often difficult to address with heuristic algorithms, and
- an ease of integration into existing design flows, which allows for rapid design iterations and exploration of the design space [156].

3.1.2 SAT Algorithms For Hardware Security

SAT solvers are capable of efficiently resolving large-scale Boolean problems, which has led to the utilization of the solvers for the evaluation of the security of cryptographic hardware. A SAT solver is capable of evaluating the Boolean functions of cryptographic hardware for vulnerabilities that compromise the security of the system or circuit [157]. SAT attacks, which utilize SAT solvers to exploit vulnerabilities in Boolean functions, are particularly effective against symmetric key cryptographic algorithms that use the same key for encryption and decryption. A SAT attack is executed first modeling the circuit as a set of Boolean equations and then using a SAT solver to find solutions to the formulated equations that correspond to secret keys or other sensitive circuit information. SAT attacks are particularly effective when executed on hardware implementations of cryptographic algorithms as subtle differences in timing and power consumption are exploited to extract secret information [158]. The ability of SAT attacks to exploit subtle differences in the functionality, timing, and power consumption of two circuits makes the SAT attack highly effective in the assessment of the security provided by an implemented obfuscation methodology including the logic locking technique developed to prevent IP piracy [158]. The need for logic locking techniques arise to combat the threats posed by counterfeiting, piracy, and overproduction of integrated circuits by unauthorized contract foundries [109, 159–162].

The locking techniques introduce additional logic to the IC, which modify the functionality of portions of the original circuit with an applied key. The key bits are stored in tamper-proof memory, and the modified IC produces the correct output only if the stored key is set correctly. The correct key is not revealed to the foundry. Once the IC is fabricated, packaged, and returned to the design house, the tamper-proof memory is loaded with the correct key, which activates the IC. The activated IC is then sold to an end-user [157]. Logic locking effectively protects the IC from unauthorized production and counterfeiting by ensuring that only the design house possesses the means to activate the IC with the correct key. As a consequence, novel attacks have been developed [118, 163, 164] that exploit the vulnerabilities of the various logic locking techniques. One of the primary threats to logic locked circuits is the satisfiability (SAT) attack [157, 165], which remains one of the most efficient methods to de-obfuscate a logic locked circuit and is often utilized as a metric for the evaluation of the security provided by techniques that mask the functionality of a circuit.

The pseudocode of the SAT attack is provided as Algorithm 2. The combinational logic space is represented as $C(\vec{X}, \vec{K}, \vec{Y}) \subseteq \mathbb{B}^{M+L+N}$. The $\mathbb{B}$ represents the binary domain of $\{0, 1\}$, $\vec{X} \in \mathbb{B}^M$ represents the M number of primary inputs, $\vec{Y} \in \mathbb{B}^N$ represents the N number of primary outputs, and $\vec{K} \in \mathbb{B}^L$ represents the L number of key inputs [118, 163, 164].

The algorithm consists of three primary steps: 1) generation of a distinguishing input

Algorithm 2: SAT Attack Algorithm [166]

Input: C and $eval$

Output: $\vec{K}_C$

$i := 1;$

$F_1 = C(\vec{X}, \vec{K}_1, \vec{Y}_1) \wedge C(\vec{X}, \vec{K}_2, \vec{Y}_2);$

while $sat[F_i \wedge (\vec{Y}_1 \neq \vec{Y}_2)]$ **do**

$\vec{X}_i^d := sat_assignment_{\vec{X}}[F_i \wedge (\vec{Y}_1 \neq \vec{Y}_2)];$

$\vec{Y}_i^d := eval(\vec{X}_i^d);$

$F_{i+1} := F_i \wedge C(\vec{X}_i^d, \vec{K}_1, \vec{Y}_i^d) \wedge C(\vec{X}_i^d, \vec{K}_2, \vec{Y}_i^d);$

$i := i + 1;$

end

$\vec{K}_C := sat_assignment_{\vec{K}_1}(\vec{F}_i)$

pattern (DIP), 2) evaluation of the DIP on an oracle IC, and 3) the addition of constraints to the SAT model of the circuit based on the oracle response. To generate the DIP, the algorithm utilizes a miter circuit of the locked netlist, which is obtained through reverse engineering of the IC. The miter circuit is a logical representation that includes two identical copies of a locked circuit with different key vectors applied to each and where each corresponding output is XORed to check for equivalence. The resulting outputs are then connected to an OR gate, as shown in Fig. 3.1.

A SAT solver generates an assignment $\vec{X}_i^d$ for the primary input vector $\vec{X}$ that satisfies the formula $F_i \wedge (\vec{Y}_1 \neq \vec{Y}_2)$, where F_i represents the Boolean formula that describes the miter circuit. To efficiently convert the logical representation of the miter circuit into conjunctive normal form (CNF), the algorithm employs a Tseytin transformation [167].

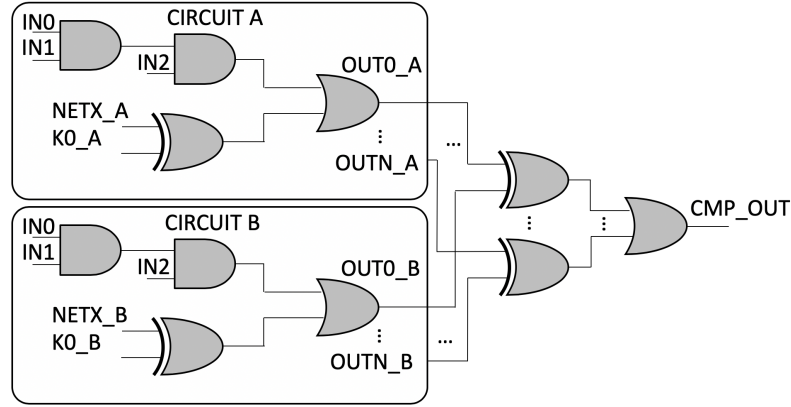


Figure 3.1: Miter circuit with replicated A and B versions of the reverse engineered netlist that includes obfuscated gates. For different applied key, the corresponding outputs between the A and B versions of the circuit are XORed. Each XORed signal is then passed to an OR gate to check for any differentiating output.

Converting the logical function of the circuit to CNF reduces the complexity of assigning values to the variables, evaluating the Boolean expressions, and backtracking when executing the SAT algorithm as the CNF formula typically consists of clauses that are disjunctions of literals, where a literal is either a variable or a negation of the variable. The Tseytin transformation converts a logical expression into a series of simpler clauses by introducing new variables that represent intermediate results and then formulating rules that link the new variables such that the original logic function is expressed. Finally, the CNF representation of the circuit is passed to the SAT solver to generate the DIP(s). The SAT solver utilizes the constraints generated from the oracle response to ensure that the generated DIPs are valid for the locked circuit.

If there is a difference in the logical output between the two copies of the circuit, the miter circuit produces a logic one. As the applied keys are the only difference between the

two copies of the circuit, a generated DIP indicates non-equivalent outputs from the miter circuit due to applied keys. The generated DIPs are then used as inputs to an activated IC, where the unlocked IC is utilized to determine the correct outputs of the circuit for the given input pattern such that $\vec{Y}_i^d := eval(\vec{X}_i^d)$. As there is a mismatch in at least one output from the two identical circuits, applying the DIP to an activated IC eliminates at least one key.

The inputs and outputs are propagated through the CNF equivalent representation of the locked netlist, generating a set of logical constraints that are appended to the original SAT formulation such that $F_{i+1} := F_i \wedge C(\vec{X}_i^d, \vec{K}_1, \vec{Y}_i^d) \wedge C(\vec{X}_i^d, \vec{K}_2, \vec{Y}_i^d)$. The process of DIP generation, evaluation on an oracle IC, and adding input/output constraints to the SAT model is repeated until no further generation of DIPs is possible.

The effectiveness of the SAT attack to de-obfuscate logic locked circuits is characterized by results shown in Fig. 3.2 [166]. Five different logic locking methodologies are analyzed, which include random insertion (RND) [109], fault-analysis based XOR (ToC'13/xor) and MUX (ToC'13/mux) insertion [110], LUT insertion (DTC'10/LUT) [111], and strong logic locking (DAC'12) [168], and a technique to increase the switching probability of nets that facilitates hardware Trojan detection (IOLTS'14) [169]. The maximum allowed increase in the area of the circuit, as measured by the number of extra gates added to the original netlist, was set to 5%, 10%, 25%, and 50%. The ISCAS'85 and Microelectronics Center of North Carolina (MCNC) benchmark circuit suites were used for evaluation of

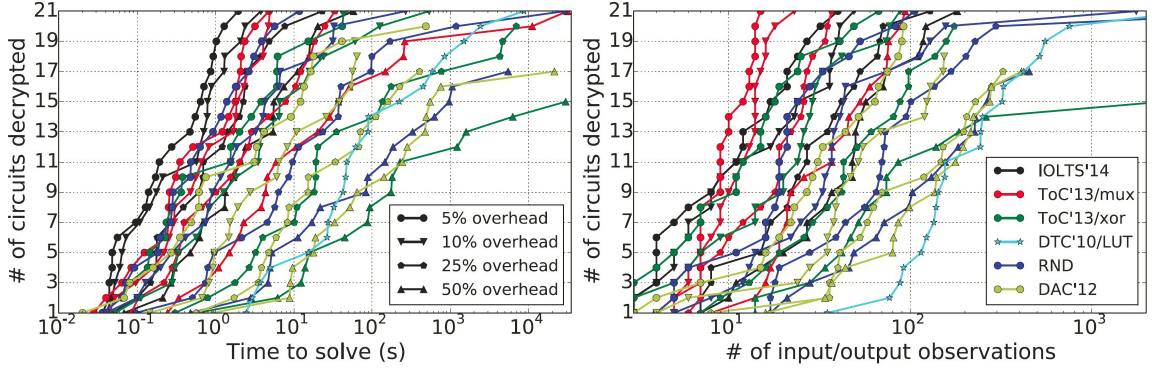


Figure 3.2: Analysis of the execution time (left) and number of input/output combinations needed to decrypt the circuit (right). Each color represents a different locking technique and each marker represents a different allowed percentage overhead [166].

the SAT attack. The results shown in Fig. 3.2 highlight the primary advantage of utilizing SAT algorithms, the quick de-obfuscation of secured circuits as the SAT attack decrypted 95% of the obfuscated benchmark circuits within ten hours. SAT solvers are, therefore, the primary means of evaluating the efficacy of logic locking techniques due to the efficiency of the solvers to de-obfuscate logic locked circuits.

3.2 Satisfiability Modulo Theory (SMT) Algorithm

SAT algorithms are widely used to solve large and complex Boolean problems. However, SAT solvers are limited to Boolean logic and are unable to solve non-Boolean formulated problems. To address the limitation of SAT solvers, algorithm based on Satisfiability Modulo Theory (SMT) were developed as an extension of the Boolean Satisfiability algorithms [138]. The primary difference between the two algorithms is that while the SAT

algorithm operates only on Boolean variables, the SMT algorithm is capable of accepting variables of differing mathematical theories that results in SMT algorithms particularly useful in solving problems that include non Boolean constraints [138, 170, 171].

Theories in SMT refer to collections of axioms and inference rules that describe mathematical objects and corresponding properties [170]. The theories define the set of logical formulae that are considered meaningful in the context of a given structured problem. For example, the theory of integers is comprised of axioms that define the basic arithmetic operations of addition, subtraction, multiplication, and division. SMT solvers utilize theories to check the satisfiability of logical formulae across the given structured problem. SMT solvers require that the input formulae is encoded in a language that is suitable for structured reasoning given the set of relevant theories. The encoded satisfiability of the formulae is then determined using a combination of techniques, including constraint propagation, quantifier elimination, and Boolean satisfiability [138, 170, 171]. Common theories utilized by SMT solvers include [171]

- linear arithmetic, which utilizes linear equations and inequalities with real or integer variables,
- bitvector, which utilizes fixed-size sequences of bits and supports bitwise operations including AND, OR, and XOR, and
- array, which utilizes arrays or matrices of elements of some fixed type, including integers and Booleans.

Algorithm 3: Simplified pseudocode of the SMT algorithm [172].

Input: Set of constraints C
Output: UNSAT or SAT + Satisfying assignment
begin
 $F = \text{Convert-to-Standard-Form}(C)$
 $\text{SATResult} = \text{SAT-Solver}(F)$
 if $\text{SATResult} == \text{UNSAT}$ **then**
 return UNSAT
 end
 else
 $\text{model} = \text{Get-Model}(\text{SATResult})$
 if model satisfies *all* theories in F **then**
 return SAT + Satisfying assignment
 end
 else
 $\text{conflict} = \text{Get-Conflict}(\text{model}, F)$
 $F = \text{Add-Conflict}(\text{conflict}, F)$
 end
 end
end

The pseudocode of the SMT routine is provided as Algorithm 3, where the input to the algorithm is a set of constraints C from different allowed theories. The SMT algorithm begins by combining different theories to form a single set of logical constraints C , which is then translated into a SAT problem. The translation operation *Convert-to-Standard-Form* is completed by encoding each theory-specific constraint into a set of Boolean clauses that are then combined to form the single SAT problem. The translation procedure is explained through the following example: Assume the expression

$$(x > 0) \text{ AND } (y < 5) \text{ AND } ((x + y) > 10), \quad (3.1)$$

is given, where both bit vector and linear arithmetic theories are utilized. The arithmetic constraints $(x > 0)$ and $(y < 5)$ are first represented using a technique called bit-blasting, which involves encoding each arithmetic operation into a set of Boolean operations on the individual bits of the operands. For example, one possible representation of the constraint $x > 0$ for a 32-bit operation is

$$(x[31] \text{ OR } x[30] \text{ OR } \dots \text{ OR } x[1] \text{ OR } x[0]), \quad (3.2)$$

where, $x[i]$ represents the i^{th} bit of x . Similarly, one possible representation of the constraint $y < 5$ for a 32-bit operation is

$$(\text{NOT } y[31] \text{ AND NOT } y[30] \text{ AND } \dots \text{AND } y[2] \text{ AND NOT } y[1] \text{ AND } y[0]), \quad (3.3)$$

where *NOT* represents the logical negation operator.

To allow for both arithmetic and Boolean operations in the case of $((x + y) > 10)$, the SMT solver utilizes bit-blasting to represent the arithmetic operation $(x + y)$ as a set of Boolean operations on the individual bits of x and y . The SMT solver then converts the resulting Boolean expression into Conjunctive Normal Form (CNF), which is a standard format to represent Boolean formulae.

After encoding all the theory-specific constraints into Boolean clauses, the SMT solver

combines the clauses into a single set of logical constraints that are applied to a SAT solver. If the SAT solver is not able to determine a solution to the logical constraints, then the algorithm returns *UNSAT*. If the SAT solver returns a satisfying assignment, the algorithm checks whether the assignment satisfies all theories utilized in C . If all theories in C are satisfied, then the algorithm returns the assignment as the solution to the given set of input constraints.

If the assignment does not satisfy all theories in C , the algorithm calls a procedure *Get – Conflict* to obtain a conflicting constraint that is not satisfied by the model. When the solver encounters a conflict, a clause is recorded that is the logical negation of the conjunction of the literals in the conflicting clause, where the generated clause described as a conflict clause. The execution of the *Get – Conflict* operation continues with the retrieval and analysis of the conflict clause to identify the subset of variables that must be false for the conflict to occur. The subset of variables is then used to construct a new clause that is added to the set of constraints being considered by the solver.

The assignment that resulted in the conflict is then added to F using the procedure *Add – Conflict*. The algorithm then loops and calls the SAT solver again on the updated set of constraints F . The loop continues until either the SAT solver returns "*UNSAT*," indicating that there is no satisfying assignment for the input constraints, or a satisfying assignment is found that satisfies all theories in C [172].

3.3 Use of SMT Algorithm For Analog Circuit Sizing

In recent years, the utilization of the Davis-Putnam-Logemann-Loveland (DPLL) algorithm with interval based arithmetic constraints [142] has led to large performance gains in multi-dimensional search space exploration [138, 140]. The primary advantage is achieved by combining the superior handling of large interval based arithmetic constraints with the learning provided by the DPLL algorithm within the SMT framework. The SMT solvers are, therefore, utilized to efficiently explore the multi-dimensional search space observed when determining the sizes of the transistors of an analog circuit to meet target specifications, including the gain, bandwidth, and power consumption of the circuit. The solver uses generic analog design equations and translates the problem into SMT aware nonlinear constraints. The nonlinear constraints along with the permitted range of transistor dimensions are utilized by the algorithm to explore the search space. The SMT solver exhaustively explores the search space and outputs transistor sizes that meets all the performance parameters.

Related work pertaining to analog circuit sizing fall under two primary categories: (1) cost-based optimization methodologies, and (2) equation based methodologies. The optimization based methodologies include DELIGHT.SPICE [173], STAIC [174] and OPASYN [175], which are classical optimization techniques, and IDAC [176], MAELSTROM [177], ASTRX/OBLX [178], ASLIC [179], and OASYS [180], which are optimization techniques

based on heuristics. More recently, evolutionary computation (EC) algorithms including genetic algorithms and differential evolutions have been leveraged to design and optimize analog circuits [181]. The CMOS operational amplifier synthesizer (DARWIN) is an EC algorithm that when executed, selects the amplifier and sizes the circuit [182]. In [183], evolutionary algorithms are utilized for the automated design of a circuit. where genetic programming is used to generate new circuit topologies and current-flow analysis to screen and correct the generated circuits. In [184], a CMOS op-amp is designed by solving a convex optimization problem using geometric programming techniques. A memetic single-objective evolutionary algorithm (MSOEA) is introduced in [185] for automated analog circuit sizing. The MSOEA is capable of solving sizing problems with a large number of constraints. The described optimization techniques are only applicable to a particular analog circuit family or topology and can not be utilized with different categories of analog circuits. In addition, the techniques are sensitive to various search parameters and often provide solutions that do not meet all designer specifications. In [140], which is similar to our proposed methodology, an SMT solver is used to exhaustively search the design space to determine a rough approximation of the biasing parameters of the analog circuit. An interval arithmetic solver is used to remove solutions that do not meet the target specifications. The determined range of the bias is then mapped to transistor sizes utilizing polynomial regression. The work described in this dissertation differs as generic analog design equations are provided as input to the SMT solver to directly determine the

sizes of transistor, which improves the computational performance.

3.3.1 Analog Satisfiability (aSAT) for Design Space Exploration

Satisfiability based verification for analog and mixed signal (AMS) circuits has gained significant importance due to the development of powerful SAT solvers. The SAT based techniques provide both the capacity and the efficiency required to solve linear as well as non-linear equations with interval arithmetic constraints [140, 143, 186, 187].

3.3.1.a Problem Formulation

The developed satisfiability technique uses generic analog circuit design equations for the gain, operating frequency, phase noise, and bandwidth to determine transistor sizes that meet the given circuit constraints and specifications. Depending on the complexity of the analog circuit, the sizing of the transistors is formulated as a single or multi-objective constraint optimization problem. Single-objective constraint optimization problems are formulated for single stage analog circuits including filters, voltage dividers, and amplifiers. For single-objective constraint optimization problems, solving a single objective function is sufficient to meet the circuit specifications. The single objective optimization problem is formulated as

$minimize_x f(x)$, given

$$g_i(x) \geq 0,$$

$$g_j(x) \leq 0, \text{ and} \tag{3.4}$$

$$h(x) = 0, \text{ for}$$

$$x \in [X_L, X_H],$$

where $f(x)$ is the objective function to minimize and $h(x)$ represents the equality constraints that utilize Kirchhoff's voltage law (KVL) and Kirchhoff's current law (KCL). Design variables are denoted by the vector x , whose range is denoted by the lower and upper bound of X_L and X_H respectively. The $g(x)$ vector corresponds to user-defined inequality constraints.

Multi-objective constraint optimization problems are formulated for complex circuits and multi-stage analog circuits including operational amplifiers, VCOs, active filters, and PLLs, and require the utilization of more than one objective function. In multi-objective analog circuit sizing, the simultaneous optimization of all objective functions is performed to obtain an approximate Pareto-optimal front [140]. With multi-objective constraint optimization problems, performances parameters are set as either objective functions or as constraints that involve sensitivity analysis between different objective functions. A multi-objective constraint problem is formulated as

$$\begin{aligned}
& \text{minimize area,} \\
& \text{maximize gain, and} \\
& \text{assure transistor operation in saturation, given} \\
& g(x) \geq 0, \\
& i(x) \leq 0, \text{ and} \\
& h(x) = 0, \text{ for} \\
& x \in [X_L, X_H].
\end{aligned} \tag{3.5}$$

The range of widths and lengths along with the circuit constraints are applied as inputs to an analog satisfiability (aSAT) solver. The general formulation of the SAT problem is given by

$$\begin{aligned}
X_{min} &\leq X \leq X_{max}, \\
Y_{pmin} &\leq Y_p \leq Y_{pmax}, \\
y_j &= f(x_i)
\end{aligned} \tag{3.6}$$

where,

- $[X_{min}, X_{max}]$ is the permitted range of transistor sizes,

- $[Y_{pmin}, Y_{pmax}]$ are the permitted ranges of the circuit constraints or performance parameters,
- $X = \{x_i = 1 \dots n\}$ are the transistor dimensions (length and width) for n number of transistors,
- $Y_p = \{y_j = 1 \dots m\}$ are the performance parameters,
- $y_j = f(X)$ for $j = 1 \dots n$ are the mapping equations from X to Y , and
- p is the index representing each individual constraint.

3.3.1.b SAT algorithm

A satisfiability (SAT) decision problem is NP-complete, wherein for a given formula ϕ , a SAT solver answers whether a variable assignment exists such that the formula ϕ is true. If such an assignment exists, then the formula ϕ is said to be satisfiable. Otherwise, formula ϕ is said to be unsatisfiable. The SMT solver utilizes the conflict driven clause learning (CDCL) algorithm, which is based on the DPLL algorithm, for the *decision* and *deduction* sub-routines applied to real variables.

The pseudocode that describes the use of SMT to determine the widths of transistors for a given set of analog circuit constraints is provided as Algorithm 6. The constraint formula ϕ is provided as input to the SMT solver along with the technology dependent constraints. The SMT solver begins by selecting a random range for the width X and the

performance parameter (Y_p) for each circuit node along with the corresponding interval range of the unassigned variables during the execution of the *decision* sub-routine. The SMT solver selects one of the unassigned variables and splits the interval range into two subintervals of the same length. The solver then temporarily discards one of the subintervals and selects the other. Next, the interval constraint propagation (ICP) technique is applied to the formula ϕ . If the ICP routine terminates with no conflict, then the algorithm returns to the *decision* sub-routine. If a conflict exists, denoted by the reduction of the interval of a variable to null, the source of the decision that lead to a conflict is determined by executing the conflict driven clause learning algorithm. When the union of the sources of the conflict covers the entire search space, the algorithm returns UNSAT. Otherwise, a backtracking sub- routine is called and the algorithm returns to the *decision* sub-routine after adding a conflict clause to the formula ϕ . The union of all the intervals is the superset of the solution space.

The SMT algorithm executes as a branch-and-prune process, where the *deduction* sub-routine recognizes the previous decision as a branch point if the solver makes an incorrect decision that leads to a subinterval containing no solution and prunes the corresponding subinterval by adding the clause that led to the empty set to the problem formula ϕ . If the SMT solver prunes the entire search space, then no solution exists within the search space and the solver returns UNSAT. In contrast, if the formulated problem ϕ is satisfiable, then the branch and prune process returns the pruned interval range. The branch

Algorithm 4: SMT algorithm to determine the widths of transistors for a given set of analog circuit constraints.

Input: circuit constraint formulae ϕ
Output: S
S=empty set
while *variable with interval greater than δ exists* **do**
 decision()
 pick the variable and divide the range in half
 select one of the subintervals
 deduction()
 if $ICP(\phi)=UNSAT$ **then**
 find conflict-source s
 $S = S \cup s$
 if $S=entire\ state\ space$ **then**
 return UNSAT
 end
 else
 undo all decision and deduction after s
 $\phi=\phi \cap \bar{s}$
 end
 end if
end
end while
return UNSAT

and prune process continues to execute indefinitely as, no matter how small of a range, there is always half of a remaining interval that was not explored. Therefore, termination conditions are needed, which are implemented by specifying a threshold δ . When the upper bound of a variable is lower than the specified threshold δ , the variable is no longer considered during execution of the *decision* sub-routine. The solver terminates when no variable has an interval greater than the threshold δ , with the current intervals of each

variable returned as the result. The result returned by the solver is not an exact point, but rather, an interval containing the optimal solution point. Provided a small enough termination threshold δ , the returned space provides an approximate solution point. However, a reduction in δ comes at the cost of increased computational time.

The primary objective of the parameter space exploration algorithm is to determine a feasible performance space and transistor operating range for the given constraints and specifications. The computational cost of solving SMT based circuit equations increases exponentially with the complexity and number of constraints and with a wider parameter range. Large dimensions lead to a large initial performance space, which is computationally expensive to search.

3.3.2 Applying aSAT for Transistor Sizing of Circuit Parameters

The developed aSAT design methodology is applied to a two stage differential amplifier and an LC tank voltage controlled oscillator (VCO). All parameter selections based on aSAT solutions are obtained using iSAT3 [188], which is a satisfiability checker for Boolean combinations of arithmetic constraints that utilize real and integer valued variables. The iSAT3 solver is capable of taking linear arithmetic, non-linear arithmetic, and transcendental functions as input. The widths and lengths obtained from the aSAT solver for a 180 nm CMOS process are then validated through SPICE simulations.

Table 3.1: Equations of the circuit parameters of a two stage amplifier applied as input to the aSAT solver.

Circuit Parameter	Dependent Parameters	Governing Equation
DC Gain	Transconductance	$g_m = \sqrt{2\mu_n C_{ox} W/L * I_D}$
	Output Conductance	$g_{ds} = 1/2 * \mu_n C_{ox} W/L * (V_{gs} - V_t)^2 * \lambda_n$
	First Stage Gain	$\frac{g_{m2}}{g_{ds2} + g_{ds4}}$
	Second Stage Gain	$\frac{g_{m7}}{g_{ds7} + g_{ds8}}$
	Total DC Stage Gain	$A_{DC} = \frac{g_{m2}}{g_{ds2} + g_{ds4}} * \frac{g_{m7}}{g_{ds7} + g_{ds8}}$
Phase Margin	Gain Bandwidth	$GBW = \frac{g_{m1}}{2\pi C_c}$
	Zero (Transfer Function)	$z = \frac{g_{m7}}{C_c}$
	First Pole (Transfer Function)	$P_1 = \frac{1}{g_{m7} R_1 R_2 C_c}$
	Second Pole (Transfer Function)	$P_2 = \frac{g_{m7}}{C_1 + C_2}$
	Phase Margin	$PM = 180 - \tan^{-1}(\frac{GBW}{z}) - \tan^{-1}(\frac{GBW}{P_1}) - \tan^{-1}(\frac{GBW}{P_2})$

Table 3.2: Equations of the circuit parameters of an LC tank VCO applied as input to the aSAT solver.

Circuit Parameter	Dependent Parameters	Governing Equation
VCO Operating Frequency	Varactor Capacitance	$C_{var} = C_{ox} * W_{var} * L_{var} + C_{gs} * W_{var} + C_{gd} * W_{var}$
	VCO Core Capacitance	$C_{core} = 2/3 * C_{ox} * W_{core} * L_{core} + C_{gs} * W_{core} + C_{gd} * W_{core}$
	Total Capacitance	$C_{total} = C_{var} + C_{core}$
	Frequency	$f_0 = \frac{1}{2\pi\sqrt{LC_{total}}}$
VCO Gain	Varactor Capacitance Tuning Range	$C_{var} = \frac{C_{var}}{(1 + \frac{V_{control}}{V_j})^m}$ for $V_{control} \in [0, V_{dd}]$
	VCO Gain	$z = \frac{g_{m7}}{C_c}$

3.3.2.a Application of aSAT to a Differential Amplifier

Differential amplifiers, in addition to amplifying differences between two signals, also suppress common mode voltage noise between the two inputs. The differential amplifier topology considered shown in Fig. 3.3. Using the equations listed in Table 3.1 as circuit constraints, the problem is formulated and is applied as input to an aSAT solver. The transistor sizes (W/L ratio) determined through execution of the aSAT solver for a differential amplifier with target performance specifications of 40 dB in gain, 5 MHz of gain bandwidth, 5 V/ μm^2 of slew rate, 10 pF of load capacitance a C_{out} , 0.8 V to 1.6 V range for the input common mode voltage, and less than 100 μW of power dissipation are listed in Table

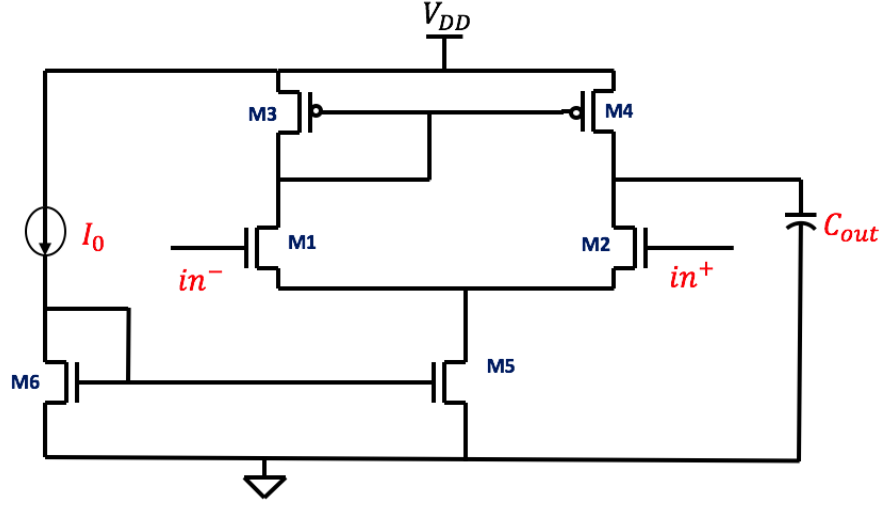


Figure 3.3: Circuit schematic of a differential amplifier.

Table 3.3: W/L ratios determined by executing the aSAT solver on a differential amplifier with a target gain of 40 dB, gain bandwidth of 5 MHz, slew rate of $5V/\mu m^2$, load capacitance of 10 pF, input common mode voltage range of 0.8 V to 1.6 V, and power dissipation of less than 100 μW .

Parameter	Range of W/L Ratio	Selected W/L Ratio
M1, M2	[1, 1000]	6.58
M3, M4	[1, 1000]	84
M5, M6	[1, 1000]	5.75

3.3. The differential amplifier with transistor sizes obtained from the aSAT solver is then characterized with SPICE simulation, and the resulting performance values are compared with target circuit specifications. The results of the comparison are listed in Table 3.4.

3.3.2.b Application of aSAT to a Two Stage Amplifier

The two stage amplifier topology shown in Fig. 3.4 is considered. The supply voltage V_{dd} is set to 1.8 V and the load capacitance C_{out} is set to 2 pF. The input common

Table 3.4: Analysis of gain A_v , gain bandwidth GBW, and power dissipation through SPICE simulation of a differential amplifier with transistor dimensions determined through aSAT.

Performance Metrics	aSAT target Value	SPICE Value
A_v (dB)	40	35
GBW (MHz)	5	4.4
Power Dissipation (μ W)	≤ 100	90

mode voltage range is set between 0.8 V and 1.6 V to ensure that the transistors operate in saturation. To ensure the stability of the circuit and to meet the phase margin requirements, C_c is set to the smallest value greater than $0.22C_{out}$. Using the equations listed in Table 3.1, while also considering the input common mode range (ICMR), slew rate, and power dissipation with transistors constrained to operate in saturation, the problem is formulated as circuit constraint equations and applied as input to an aSAT solver. The transistor sizes (W/L ratio) of a two-stage amplifier when targeting specifications of 60 dB gain, 30 MHz of gain bandwidth, and power dissipation of less than 300 μ W are listed in Table 3.5. The transistor sizes obtained from the aSAT solver are then characterized with SPICE simulation. The performance metric calculated from the simulation of the two stage amplifier with widths and lengths determined through aSAT analysis are compared against the targeted circuit specifications. The results of the comparison are listed in Table 3.6. The simulated results indicate that the gain, phase margin, and power dissipation all fall within the targeted specifications. However, there is a 7 MHz drop in the gain bandwidth of the amplifier.

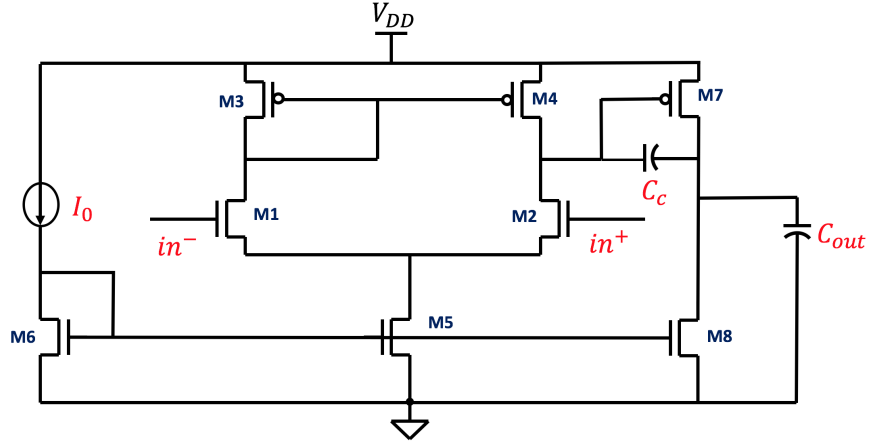


Figure 3.4: Circuit schematic of a two stage amplifier.

Table 3.5: Determined W/L ratio when applying aSAT to a two stage amplifier with target specifications of 60 dB gain, 30 MHz gain bandwidth, and power dissipation of less than 300 μ W.

Parameter	Range of W/L Ratio	Selected W/L Ratio
M1, M2	[1, 1000]	3.62
M3, M4	[1, 1000]	7.81
M5, M6	[1, 1000]	7.41
M7	[1, 1000]	117.81
M8	[1, 1000]	55.88

Table 3.6: Characterization of the gain A_v , GBW, phase margin, and power dissipation of a two-stage amplifier with transistor dimensions determined through aSAT analysis.

Performance Metrics	aSAT target Value	SPICE Value
A_v (dB)	60	63.02
GBW (MHz)	30	23
Phase Margin $^\circ$	$40^\circ \geq \text{PM} \geq 60^\circ$	49.77°
Power Dissipation (μ W)	≤ 300	206.8

3.3.2.c Application of aSAT to a VCO

An LC tank voltage controlled oscillator (VCO) topology is considered as shown in Fig. 3.5. To highlight the variability of the aSAT algorithm in solving constraint driven equations, a problem is formulated that applies as input ranges of widths and lengths of transistors to an aSAT solver. The algorithm exhaustively and simultaneously explores both the range of transistor widths and the range of transistor lengths and outputs transistor dimensions that satisfy the performance constraints. Using the equations listed in Table 3.2 and ensuring the transistors are constrained to operate in saturation, the problem is formulated and applied as input to an aSAT solver. The resulting transistor lengths and widths for a target frequency of 1.75 GHz are listed in Table 3.7.

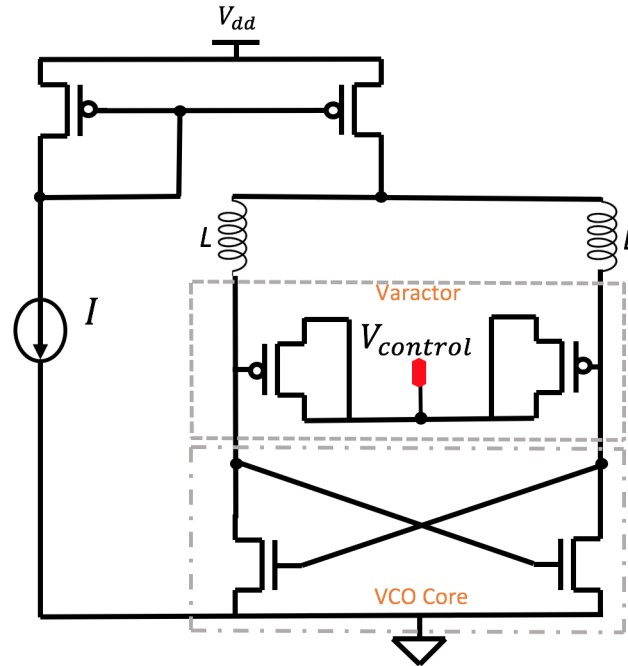


Figure 3.5: Circuit schematic of a LC VCO.

The results of executing the aSAT algorithm to determine the transistor sizes of the varactor and VCO core for different target frequencies are listed in Table 3.8. Column 2 and 3 of Table 3.8 are the aSAT generated transistor dimensions, which are implemented on the VCO circuit topology shown in Fig. 3.5 and are characterized with SPICE simulations for target frequencies of 1.5 GHz, 1.75 GHz, 2 GHz, 2.25 GHz, and 2.5 GHz. The resulting frequencies from characterization with SPICE simulation are listed in column 4 of Table 3.8. The percentage error in frequency for VCOs with transistors sized through aSAT analysis and characterized with SPICE simulations are listed in column 5 of Table 3.8. The simulation results indicate a maximum deviation of 6% for all target frequencies when using the transistor widths and lengths determined through aSAT analysis. A more accurate sizing of transistor dimensions is possible by incorporating equations on parasitic impedance and inductance models during problem formulation.

Table 3.7: Results of applying aSAT to a VCO with a target frequency of 1.75 GHz.

Parameter	Specified Range	Selected Value
Varactor Width	[220 nm, 50 μm]	10.5 μm
Varactor Length	[180 nm, 5 μm]	1.8 μm
VCO Core Width	[220 nm, 50 μm]	6.8 μm
VCO Core Length	[180 nm, 5 μm]	1.56 μm
VCO Gain [MHz/V]	[110, 125]	120

Table 3.8: Results from SPICE analysis of a VCO performed based on transistor dimensions determined through aSAT for multiple target frequencies.

Target Frequency	Varactor transistor sizes from aSAT [L, W] μm	VCO core transistor sizes from aSAT [L, W] μm	Frequency	% Error
1.5 GHz	[18.3, 1.7]	[5.5, 1.2]	1.59 GHz	6%
1.75 GHz	[10.5, 1.8]	[6.8, 1.56]	1.8 GHz	2.86%
2.00 GHz	[10.5, 1.5]	[5.5, 1.1]	2.08 GHz	4%
2.25 GHz	[10.5, 1.1]	[5.5, 1.1]	2.24 GHz	0.4%
2.5 GHz	[10.5, 0.8]	[5.5, 1.1]	2.41 GHz	3.6%

3.3.3 Conclusions

The capacity and the efficiency of the SMT algorithm to solve complex non-linear equations with interval arithmetic constraints is highlighted in Section 3.3.2, where the SMT algorithm is utilized to determine the transistor dimensions of an analog circuit that satisfy the target performance constraints. The aSAT algorithm was executed on a two stage amplifier and an LC tank VCO. For the two stage amplifier, the W/L transistor ratios determined through aSAT analysis met the gain, phase margin, and power consumption requirements of the circuit, but a mismatch in gain bandwidth of 7 MHz was observed. For the LC tank VCO, the aSAT methodology was used to estimate transistor sizes for different target frequencies, with the algorithm producing no more than a 6% deviation from a given target frequency. The simulated results indicate that the aSAT methodology is an accurate technique for sizing analog circuits.

Similar to the utilization of SMT in the aSAT algorithm, the SMT algorithms are used in this dissertation to select the transistors of an analog circuit that are best suited for

implementing parameter obfuscation. In addition, the SMT algorithm is also used to determine the sizes of the obfuscation transistors such that only one key results in the target functionality of the analog circuit, while any incorrect key results in significant degradation in the performance of the analog circuit. The use of SMT algorithms in parameter obfuscation are described in Chapter 5. Similar to the SAT algorithms that are used to de-obfuscate digital logic locking techniques, SMT algorithms are utilized to de-obfuscate analog circuit as a metric to evaluate the security provided by the obfuscation techniques. The use of SMT algorithms in analog de-obfuscation are described in Chapter 5 and 7.

Chapter 4

Parameter Obfuscation Techniques

In this chapter, key-based parameter obfuscation techniques are proposed to protect analog circuits against intellectual property (IP) piracy. Vector-based and mesh-based obfuscation are developed that provide different degrees of circuit security with trade-offs in design complexity and area. In addition, an algorithm for the selection of obfuscation transistors, an SMT-based algorithm to determine the sizes of the obfuscation transistors is developed to reduce design complexity and efficiently implement the proposed parameter obfuscation technique. The primary contributions of the proposed work include

1. a key-based parameter obfuscation technique that masks the biasing conditions and target performance metrics of an analog IC based on an applied key,
2. the development of an algorithm for the selection of obfuscation transistors and an SMT based algorithm for determining the sizes of obfuscation transistors, where

the principle innovations include

- an efficient integration of the parameter obfuscation technique during the design phase of an analog circuit to minimize complexity and reduce overall design time, and
- a means to auto-determine and size the transistor(s) to obfuscate that effectively mask the performance parameters of the circuit, and

4.1 Threat Model

An untrusted foundry model is assumed, where the foundry has access to the circuit design and possesses the necessary tools and skills to counterfeit and overproduce the IC from the provided GDS-II file [189]. In addition, the circuit is assumed trusted and devoid of any malicious components when in the design phase. An additional threat model considered is an untrusted end user, where the adversary has access to the primary key inputs that, based on a given applied key, alters the response of the circuit. Note that analog circuits are typically a part of a much larger system, which limits the probing of internal node voltages and currents. Therefore, an adversary is assumed unable to determine the internal node voltages and currents of the circuit.

4.2 Parameter Obfuscation

Applying the parameter obfuscation technique involves replacing a single (or multiple) transistor(s) in an analog circuit with an array of obfuscation transistors of different sizes that are controlled by a digital key. The strong correlation between the biasing conditions and the performance of an analog circuit is utilized when securing analog components with the parameter obfuscation technique. Based on the applied key sequence, which sets the transistor dimensions, the biasing conditions and performances of the circuit are set. Only when the correct key sequence is applied, and certain transistor(s) are active, are the target biasing conditions set. The technique is applicable to setting the voltage at a node, the current through a node, and/or modifying the gain of the circuit. The parameter obfuscation technique has been implemented with both vector-based and mesh-based configurations [190].

4.2.1 Vector Based Parameter Obfuscation

The topology of the vector-based parameter obfuscation technique is shown in Fig. 4.1. A single transistor is replaced by multiple transistors of the same length but of different widths placed in parallel, with each parallel transistor controlled by an independent key bit. When a digital bit is set to logic high, current flows through the corresponding parallel transistor. The total current through the parallelized transistor depends on the effective

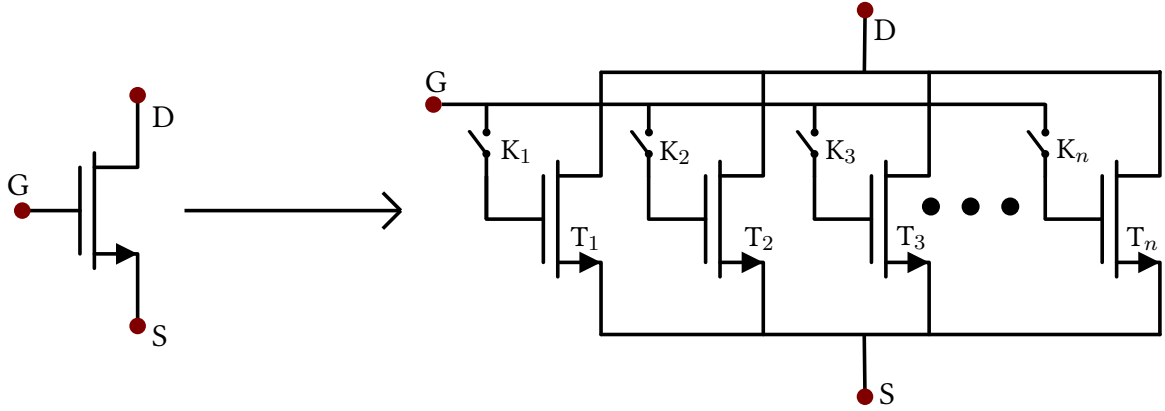


Figure 4.1: Schematic representation of vector based parameter obfuscation implemented on a single transistor.

total width of the obfuscated transistor set by the applied key, where the total current converging at node S is equal to the sum of the activated transistor paths (the on-paths) [191]. Only on the application of the correct n -bit key are the target transistors turned on, resulting in the desired currents and target performance parameters.

The original width that produces the target current is divided into multiple target transistor widths of smaller dimension that when all are properly activated result in the original target current. *Obfuscation transistors* are also added in parallel to mask the target width. The effective current through the vector obfuscated transistor is given by

$$I_{vector} = \sum_{i=1}^n I_i K_i, \quad (4.1)$$

where I_i is the drain to source current of the i^{th} parallel transistor and $K_i \in \{0, 1\}$ represents a single digital key bit applied to the i^{th} transistor. The effective current I_{vector} is the sum of the currents through the active transistor paths when the applied key bits K_i are

1. Under ideal conditions, $K_p = \mu C_{ox}$ and voltages V_{GS} , V_T , and V_{DS} are equal amongst all parallel transistors. Therefore, the cumulative effective width over length ratio T_{vector} of the activated transistors for the vector-based obfuscation technique is given by

$$T_{vector} = \sum_{i=1}^n T_i K_i, \quad (4.2)$$

where $T_i = (\frac{W}{L})_i$ is the width over length ratio of the i^{th} transistor. As the transistor lengths are kept constant, the $(\frac{W}{L})_i$ ratio is dependent solely on the transistor width W_i .

4.2.2 Mesh Based Parameter Obfuscation

The implementation of the mesh-based obfuscation technique on a single transistor is shown in Fig. 4.2. Adding transistors in parallel leads to an increase in the overall transistor width while keeping the length constant, whereas adding transistors in series results in a composite structure that effectively increases the transistor length [190].

For a series connected composite transistor, the topmost transistor of the stack typically operates in saturation while the remaining transistors operate in linear mode. The overall effective current through a series connected transistor of m rows is given by

$$I_{series} = \frac{1}{2} \left(\frac{1}{\beta_1} + \frac{1}{\beta_2} + \dots + \frac{1}{\beta_m} \right)^{-1} \cdot (V_{GS'} - V_T)^2, \quad (4.3)$$

where β is the product of the electron mobility μ , oxide capacitance C_{ox} , and the transistor

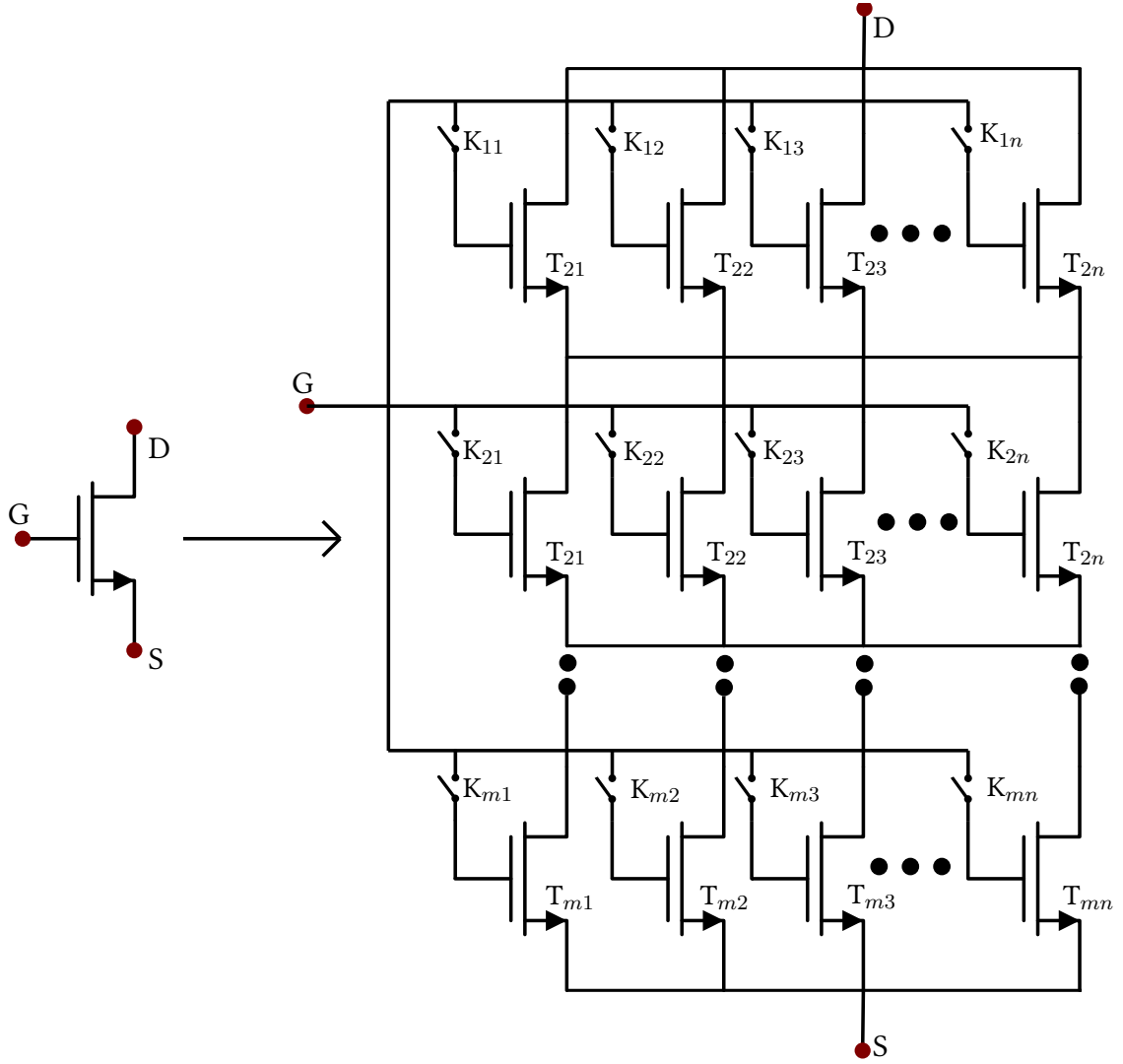


Figure 4.2: Schematic representation of mesh based parameter obfuscation implemented on a single transistor.

width over length ratio T . The $V_{GS'}$ and V_T terms are the gate to source voltage and threshold voltage of the top most transistor, respectively. Based on (4.3), the effective width over length ratio of the series connected transistor T_{series} is given by

$$\frac{1}{T_{series}} = \frac{1}{T_1} + \frac{1}{T_2} + \cdots + \frac{1}{T_m}. \quad (4.4)$$

The primary advantage of using the mesh-based obfuscation technique as compared to vector-based obfuscation is that, in addition to masking the transistor dimensions, the threshold voltage and small signal parameters of a transistor are also masked. Therefore, an adversary must determine both the biasing conditions and the small signal parameters to properly set the target transistor dimensions. Similar to the vector-based technique, *obfuscation transistors* are added to mask the *target transistors*, and the activation of each transistor of the mesh is controlled by a single key bit $K_{ij} \in \{0, 1\}$. Therefore, the effective width over length ratio of the mesh transistor T_{mesh} consisting of m rows and n columns, as shown in Fig. 4.2, and for an applied key K is given by

$$\frac{1}{T_{mesh}} = \sum_{i=1}^m \left(\frac{1}{\sum_{j=1}^n T_{ij} K_{ij}} \right), \text{ where} \quad (4.5)$$

$$\text{for each row } i \in \{1, 2, \dots, m\}, \sum_{j=1}^n K_{ij} \geq 1. \quad (4.6)$$

Applying (4.6) ensures that at least one j^{th} column of n total columns in each row i is activated. Only on the application of the correct key, are the target transistors turned on, which contribute to the desired current and small signal parameters that set the target performance of the obfuscated transistor. As the voltage at the source of the component transistor(s) nearest to the drain terminal of the composite transistor increases due to the stacking effect, the number of rows in the mesh is limited by $V_{GS} - m \cdot V_{th} \geq 0$, for m

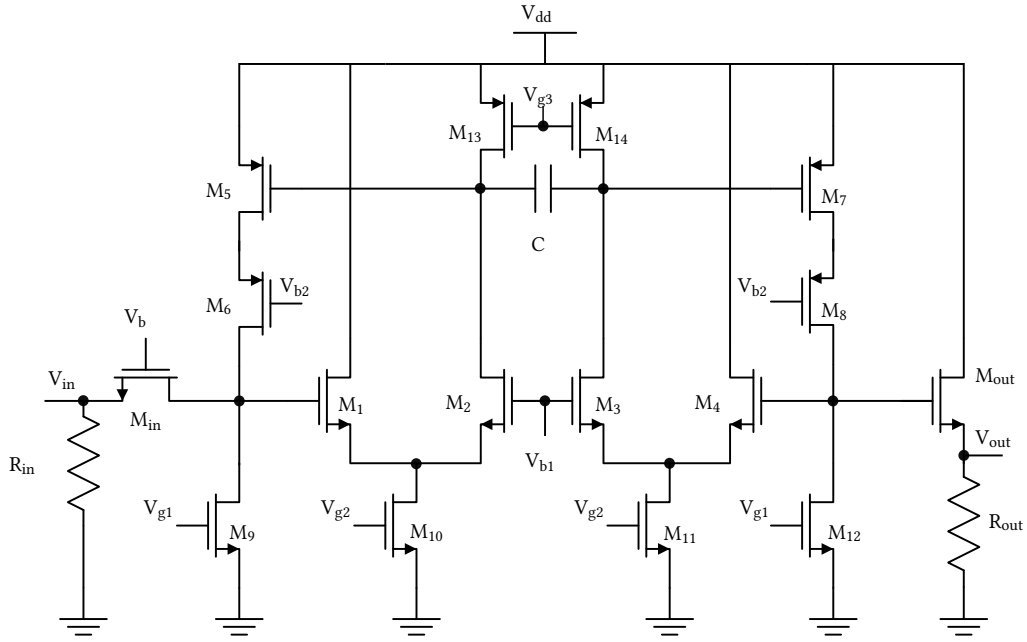


Figure 4.3: Circuit schematic of an active inductor based second order band pass filter.

number of rows.

4.3 Challenges with Parameter Obfuscation

The implementation of the parameter obfuscation techniques on an analog circuit poses design challenges. One of the primary design challenges is the drift in the performances of the analog circuit due to the parasitic impedances and non-linearities that result from the inclusion of both the *obfuscation* and *target* transistors and the switches implementing the key-bits. Therefore, additional steps are needed to assure that the target performance specifications of the circuit are satisfied while compensating for the added

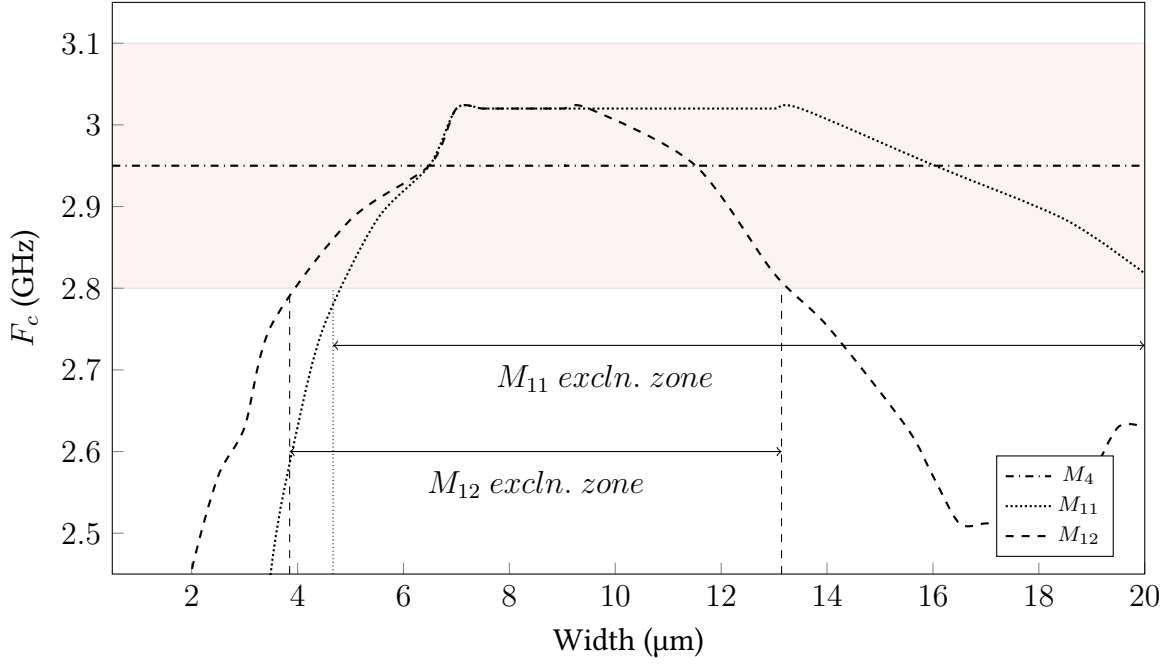


Figure 4.4: Characterization of the center frequency of the BPF for widths of 0.5 μm to 20 μm for transistors M_4 , M_{11} , and M_{12}

parasitic impedances and non-idealities resulting from the implementation of the obfuscation techniques.

Additional design challenges specific to the implementation of the parameter obfuscation techniques include 1) the determination of the transistor(s) to obfuscate in the analog circuit, 2) the presence of multiple correct keys, and 3) the possibility of incorrect keys resulting in “good enough” circuit functionality (performance close to the target specifications). The variation in the center frequency F_c of the active inductor based second-order band pass filter (BPF) shown in Fig. 4.3 for widths of 0.5 μm to 20 μm for transistors M_4 , M_{11} , and M_{12} is shown in the Fig. 4.4. The size of transistor M_4 has no effect on the F_c of the BPF, which indicates that M_4 is not an appropriate selection for the implementation

of the obfuscation technique. The size of transistors M_{11} and M_{12} produces exclusion zones of 15.33 μm and 9.3 μm , respectively, for $\pm 5\%$ variation in the target F_c . The characterization of F_c indicates that transistor M_{12} produces the smallest exclusion zone and ensures a wider range of *obfuscation transistor* sizes as compared to transistor M_{11} , and is, therefore, better suited for obfuscation. The analysis of the results shown in Fig. 4.4 indicates that proper obfuscation of the transistors of an analog circuit must be completed to effectively mask the target transistor sizes and, therefore, the performance specifications of the circuit.

The implementation of the proposed obfuscation technique on a voltage divider is shown in Fig. 4.5, with results from the analysis of the voltage at the output node of the divider listed in Table 4.1. The two key sequences listed in rows 1 and 2 of Table 4.1 for both the vector and mesh-based obfuscated voltage divider circuit produce the desired output voltage of 0.9 V. In addition, the keys listed in rows 3 and 4 produce voltages that are close to the desired output voltage of 0.9 V. The results listed in Table 4.1 highlight the challenge of producing a unique key when obfuscating analog parameters.

Three design criteria are proposed to address the challenges of implementing the obfuscation techniques: (1) determine the proper transistor(s) in the circuit to obfuscate, (2) ensure the proper sub-division of the width and length of the *target transistor*, and (3) determine the proper size of the *obfuscation transistors*. The primary constraints of

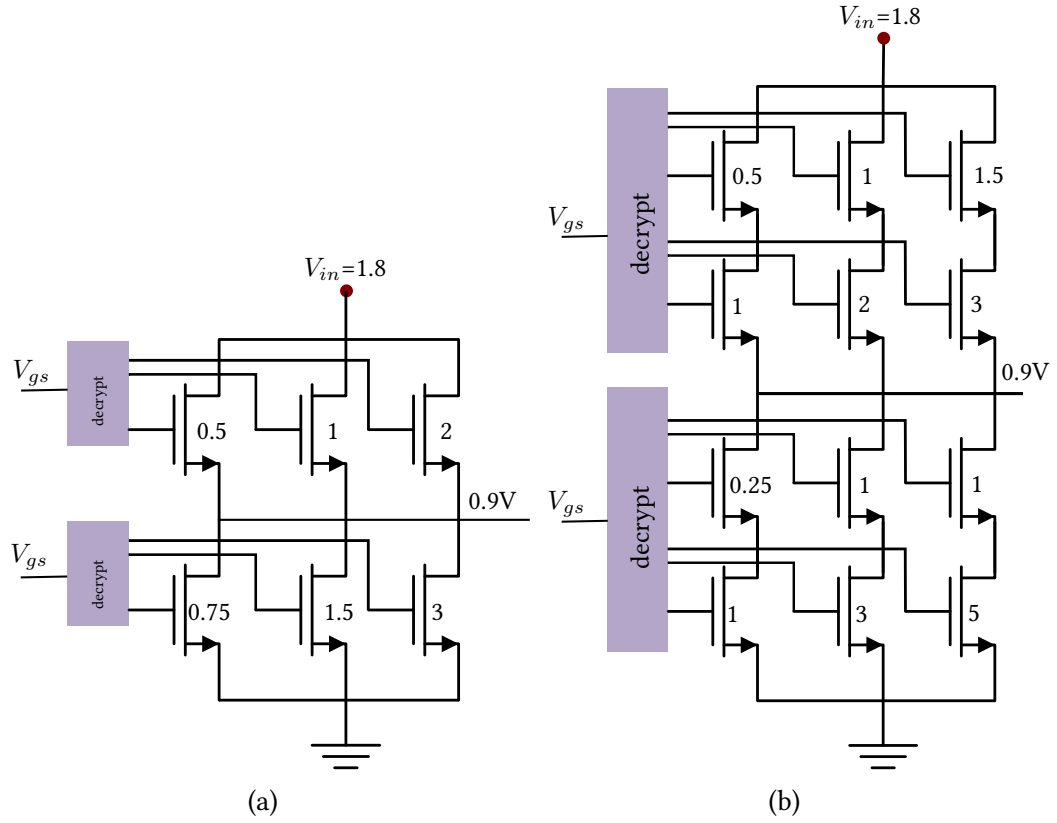


Figure 4.5: Circuit schematic of implementing the (a) vector based and (b) mesh based obfuscation techniques on a voltage divider circuit that produces a target output voltage of 0.9 V. Transistor width over length (W/L) ratios and applied voltages are provided in the figure.

criteria (2) and (3) include 1) the generation of a single correct key and 2) the circuit performances that are set by applying the closest incorrect key must result in at least a target percentage difference from the specified circuit performances produced by the correct key.

Table 4.1: Characterization of the output voltage for different key sequences applied to the voltage divider circuit shown in Fig. 4.5 obfuscated by the vector based and mesh based techniques.

Vector Based Obfuscation			Mesh Based Obfuscation		
Key 1	Key 2	V _{out}	Key 1	Key 2	V _{out}
110	010	0.9	110001	011010	0.9
011	001	0.9	101010	101001	0.9
010	100	1.01	010100	111100	1.02
001	110	0.85	001001	101010	0.85

4.3.1 Selection of Transistor(s) to Obfuscate

The primary objectives when selecting transistors to obfuscate are to 1) determine the transistor(s) that provide the smallest range of dimensions that produce circuit performances close to the target performance and 2) ensure a wider range of *obfuscation transistor* sizes. To select the transistor(s) to obfuscate, the target performance parameters are varied by a user specified percentage, and the range of transistor sizes that produce performance values within the computed performance range, given as W_{EZ} , is determined. The W_{EZ} range is subtracted from the permissible range of transistor sizes to determine the obfuscation transistor range, W_{obfus} . The transistor having the largest W_{obfus} range is selected for obfuscation. The selection process continues with the transistor with the next largest W_{obfus} range and proceeds until the requisite number of key bits for the circuit is met. From the results shown in Fig. 4.4, transistors M_{11} and M_{12} produce obfuscation ranges W_{obfus} of 4.67 μm and 10.7 μm , respectively, for $\pm 5\%$ variation in the target center frequency F_c , which indicates that transistor M_{12} is better suited for obfuscation and is,

therefore, more effective in masking the F_c of the BPF.

Algorithm 5: Selection of the transistor(s) to obfuscate by sorting the obfuscation range of the transistors in decreasing order.

Input: T_{array} , P_T , var , W_R
Output: T_{order}
for $i = 1$ **to** T_{array} **do**
 $P(i) = \text{ParaSweep}(T_{array}(i), W_R)$
 $P_{high} = P_T + P_T \cdot var$
 $P_{low} = P_T - P_T \cdot var$
 $W_{i_{EZ}} = T_i$ sizes producing performance in range(P_{high}, P_{low})
 $W_{i_{obfus}} = W_R - W_{i_{EZ}}$
 $Response = \text{VertCat}(T_{array}(i), W_{i_{obfus}})$
end
 $R_{order} = \text{DecreasingOrder}(Response, W_{obfus})$
 $T_{order} = R_{order}(\text{Column1})$
return T_{order}

To reduce the time and design complexity of determining the transistor(s) to obfuscate, an algorithm is developed that outputs an array of transistors in descending order of the transistor obfuscation range W_{obfus} . The pseudo-code for the selection of the transistor(s) for obfuscation is provided as Algorithm 5. The inputs to the algorithm include the list of all transistors of an analog circuit whose order must be determined (T_{array}), the target performance parameter(s) P_T , the variation in the performance value(s) var , and the parametric range W_R of T_{array} sizes. Based on the inputs, the algorithm begins by selecting a single transistor $T_{array}(i)$ from T_{array} . The size of $T_{array}(i)$ is varied in the range W_R and the performance of the analog circuit $P(i)$ is determined. The range of the *performance exclusion zone* (P_{high} , P_{low}) is computed, and the range of $T_{array}(i)$

sizes that produce performances that fall between P_{high} and P_{low} is determined and is defined as the transistor exclusion range W_{iEZ} . The transistor obfuscation range W_{iobfus} is computed by subtracting the transistor exclusion range W_{iEZ} from the parametric sweep range W_R . The obfuscation range for all transistors in T_{array} is computed. The transistors are arranged in decreasing order of the computed obfuscation range values (T_{order}). The algorithm returns the list of best transistor(s) to obfuscate, T_{order} , that effectively mask the performance(s) of the analog circuit.

From the determined T_{order} , additional design constraints including other performance parameters and signal integrity requirements are applied to further guide the selection of transistor(s) to obfuscate. Obfuscation of the input and output transistors is avoided, as adding *obfuscation transistors* along with the key-delivery circuit directly to nodes tied to inputs and outputs affects the signal integrity. Obfuscating the transistors used for voltage biasing to compensate for the drift in the performance of the circuit due to the implementation of the obfuscation technique results in a less complex tuning of the circuit post-fabrication as compared to obfuscating transistors used for current biasing. Tuning only the applied gate voltage is sufficient to achieve the target circuit performances when obfuscating transistors that set the voltage bias of the circuit, whereas the transistor sizes, currents, and small signal parameters are modified when obfuscating transistors that set the current bias. Based on the number of transistors to obfuscate (key size), the area, the performance parameters to obfuscate, and any additional design constraints, the designer

selects the best suited transistor(s) from T_{order} .

4.3.2 Selection of Target Transistor Sizes

Once the transistor to obfuscate is determined, the next step is to ascertain the target transistor sizes. The importance of ensuring proper sub-division of the size of each target transistor is shown through Fig. 4.6. The goal is to sub-divide a 34 μm target width into two sub-transistor sizes while ensuring that only one combination of sub-widths results in the target width. The objective for the remaining combinations of the widths (non-target) is to differ by at least 10% from the target width, where $\pm 10\%$ of 34 μm is 30.6 μm and 37.4 μm . For the combination of widths shown in Fig. 4.6a, two key sequences, "11" and "10", result in effective widths of 32 μm and 34 μm , respectively, which fall within $\pm 10\%$ of the target width of 34 μm . The sub-divided target widths shown in Fig. 4.6b are also undesired as multiple key combinations ("10" and "01") result in redundant effective transistor widths of 17 μm . The sub-division of the target width shown in Fig. 4.6c satisfies both design criteria and provides the correct total width.

4.3.3 Selection of Obfuscation Transistor Sizes

The selection of the sizes of the obfuscation transistor(s) is depicted through the example dimensions provided in Fig. 4.7. For the widths shown in Fig. 4.7a, two keys "1100" and "0011" result in an effective transistor width of 34 μm . In addition, keys "1001" and

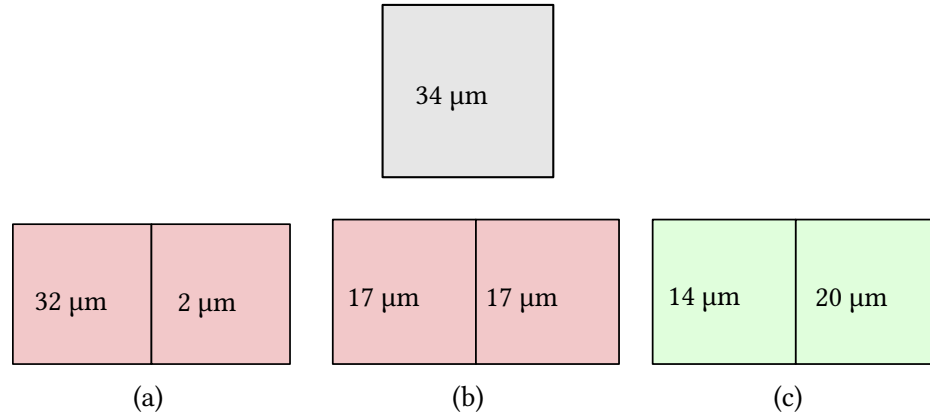


Figure 4.6: Three potential sub-divisions of a 34 μm target width into two sub-target transistor widths are shown. Both (a) and (b) violate criterion (1), whereas (c) meets the conditions set by criterion (1).

“0110” result in effective transistor widths that are within $\pm 10\%$ of the target width of 34 μm , which fails to meet the target design constraint that restricts the sizing of the transistors from producing multiple legitimate values. For Fig. 4.7b, the size of the obfuscation transistors shown in red is significantly larger than the target transistor sizes. To minimize the leakage of information, the sizes of the obfuscation transistors must be set to values similar to those of the target transistor sizes while still ensuring the presence of a single correct key and that the application of any incorrect key results in significant degradation in the circuit performances. When both the target and obfuscation transistors are close in size, subtle differences in the circuit performances are observed, which prevents an attacker from eliminating a large portion of the key space. When the obfuscation transistor sizes are either too large or too small as compared to the sizes of the target transistors, the disparity in the output performances is large and more readily observable.

Therefore, an attacker must analyze the performances of the circuit for only two transistor sizes, the largest effective transistor size from the smaller range and the smallest effective transistor size from the larger range. The analysis generates two performance values, one closer to the target and the other significantly different. Due to the large disparity, the attacker eliminates the entire range of transistor sizes from which the selected transistor size resulted in a large difference in the performance values, which significantly reduces the key/search space.

The selection of redundant dimensions for the obfuscation transistors, as shown in Fig. 4.7c, leads to redundant keys that further reduce the key space. The sizes selected for the obfuscation transistors shown in Fig. 4.7d meet all design criteria and are also close to the target transistor sizes, which effectively masks the composite transistor width.

The computational resources needed to solve for the dimensions of the target transistors and the sizing of the obfuscation transistors increase exponentially with key size (2^n for an n -bit key), which results in a lengthy and potentially expensive design cost. A satisfiability modulo theory (SMT) based algorithm is, therefore, developed that reduces the overall time needed to select the dimensions of both the *target transistors* and the *obfuscation transistors*. The proposed SMT algorithm reduces the computational complexity while ensuring that 1) the computed transistor sizes produce a single unique key, and 2) the closest circuit response due to an incorrect key results in performances that are at least a set percentage away from the specified values.

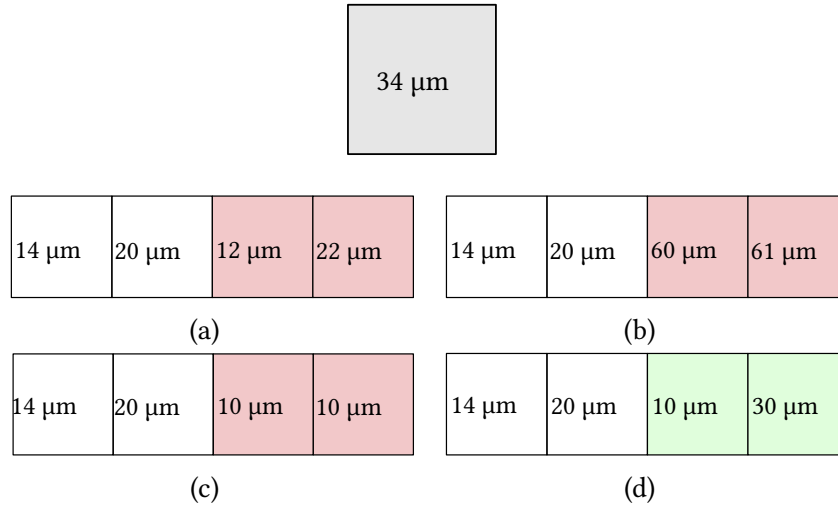


Figure 4.7: Consideration of selection criteria when sizing the *obfuscation transistors* for *target transistor* sizes of 14 μm and 20 μm , where the cumulative *target transistor* width is 34 μm . The four scenarios describe (a) when more than one effective width combination is within $\pm 10\%$ of the target width, (b) a large difference between the *target transistor* widths and the *obfuscation transistor* widths, (c) the presence of redundant widths, and (d) the desired *obfuscation transistor* widths that meet all design criteria.

4.4 Analog Satisfiability Algorithm for Transistor Sizing in Key-based Parameter Obfuscation

A satisfiability modulo theory (SMT) based algorithm is implemented to efficiently determine the sizes of the obfuscation transistors that meet the given circuit constraints and obfuscation criteria. The algorithm is executed with the iSAT3 SMT solver [188]. Based on the key size, type of obfuscation (vector or mesh based), and range of permissible transistor sizes, an SMT problem is formulated that accounts for the transistor topologies described in Sections 4.2.1 and 4.2.2. The constraints provided to the SMT algorithm limit

the number of effective cumulative widths that are possible around the target width.

4.4.1 SMT Problem Formulation

For the vector based obfuscation technique implemented with an n -bit key, (4.2) is modified to match the format required by the SMT solver as given by (4.7). The problem is formulated as a two step procedure. The execution of the first step determines the set of all possible combinations of the effective transistor sizes, $S_{T_{eff}}$, for 2^n key combinations, as given by

$$S_{T_{eff}} = \{T_{eff1}, T_{eff2}, T_{eff3}, \dots, T_{eff2^n}\}, \text{ for}$$

$$T_{eff} = \vec{T} \cdot \vec{K}^T \text{ and } \forall \vec{K} \in \{0, 1\}^n, \quad (4.7)$$

where T_{eff} defines the effective transistor width for a given key sequence $\vec{K}$ computed using (4.2). In the second step, the target transistor(s) that produce the desired cumulative width T_{eff}^* are selected. Constraints are added to ensure only one correct key is produced and that any incorrect key results in at least a target percentage variation in performance from the circuit specifications when setting the remaining effective transistor widths. The SMT problem is formulated as

$$\begin{aligned}
\phi_{vector} &= T_{eff}^* \\
&\wedge (T_{eff_1} \leq Width_{min} \vee T_{eff_1} \geq Width_{max}) \\
&\wedge (T_{eff_2} \leq Width_{min} \vee T_{eff_2} \geq Width_{max}) \\
&\wedge \\
&\wedge (T_{eff_{2^n}} \leq Width_{min} \vee T_{eff_{2^n}} \geq Width_{max}),
\end{aligned} \tag{4.8}$$

where ϕ_{vector} defines the problem as provided to the solver, T_{eff}^* is the effective transistor width of the target transistors, and $Width_{min}$ and $Width_{max}$ represent the lower and upper bound of the desired range of widths, respectively.

The range of the desired widths is a user defined parameter that is set based on two primary constraints: 1) the tuning range of the bias voltages and currents of the analog circuit to compensate for the effects of parasitics, PVT variation, and aging, and 2) the desired variation in the target performance of the circuit due to changes to the transistor widths. The consideration of PVT variations during the design phase of the circuit accounts for the required post-silicon margins, which must be considered to meet the target design criteria when sizing the target and obfuscation transistors. Limiting the total acceptable variation ensures the target transistor sizes are close to the obfuscation transistor sizes, which reduces the amount of leaked information on the sizes of the target transistors.

The steps involved in formulating the transistor sizing problem for the mesh-based obfuscation technique with an $m \cdot n$ bit key are similar to that of the vector-based obfuscation technique. The set of effective transistor widths T_{eff} for the mesh-based technique are calculated using (4.5), which is modified to SMT format as

$$S_{T_{eff}} = \{T_{eff_1}, T_{eff_2}, T_{eff_3}, \dots, T_{eff_{m \cdot n}}\}, \text{ where}$$

$$T_{eff} = \sum_{i=1}^m \left(\frac{1}{\vec{T}_i \cdot \vec{K}^T} \right) \text{ and } \forall \vec{K} \in \{0, 1\}^n. \quad (4.9)$$

The formulated SMT problem for the mesh-based technique, including the constraint that restricts the number of correct keys and the constraint that results in at least a target percentage variation in the performance specifications due to an incorrect key, is similar to (4.8) and is given by

$$\begin{aligned} \phi_{mesh} = & T_{eff}^* \\ & \wedge (T_{eff_1} \leq Width_{min} \vee T_{eff_1} \geq Width_{max}) \\ & \wedge (T_{eff_2} \leq Width_{min} \vee T_{eff_2} \geq Width_{max}) \\ & \wedge \dots\dots\dots \\ & \wedge (T_{eff_{m \cdot n - 1}} \leq Width_{min} \vee T_{eff_{m \cdot n - 1}} \geq Width_{max}). \end{aligned} \quad (4.10)$$

4.4.2 SMT Algorithm

The pseudo-code of the algorithm to determine the sizes of the *obfuscation transistors* in a topology that consists of two rows ($i = 2$) after applying mesh based obfuscation is provided as Algorithm 6, where ϕ defines the SMT formulated problem as given by (4.10) and is provided as input to the SMT solver. The algorithm is similar for the vector-based technique; however, ϕ is now given by (4.8). The SMT solver begins by selecting a random obfuscated transistor in the vector or mesh and splitting the range of dimensions into two subintervals of equal length. The solver then temporarily discards one of the subintervals and reduces the selected interval. The interval constraint propagation (ICP) technique is then applied to ϕ , where the ICP technique determines whether only one target size exists in the *EffectiveWidth* parameter. All other transistor combinations produce sizes that are smaller or larger than a user defined percentage of the target transistor dimensions, which is set to $\pm 10\%$ of the transistor size in this dissertation, as described in Section 4.3. If the ICP routine terminates with no conflict, then the algorithm returns to the decision step and selects a different obfuscation transistor until all transistor sizes in the vector or mesh are set. If a conflict exists, as indicated by a reduction to null of the range of sizes of the given obfuscation transistor, the source of the decision that leads to the conflict is located by a conflict-driven clause learning (CDCL) algorithm. When the union of sources that result in conflict covers the entire search space, the algorithm returns UNSAT. Otherwise, a backtrack routine is called, and the algorithm returns to the decision process after adding

a conflict clause to ϕ . The union of all intervals is the superset of the solution space.

Algorithm 6: aSAT for obfuscation transistor size optimization for a 2xN mesh

Input: circuit constraint formulae ϕ
Output: S
S=empty set
while *unassigned ObfusTran with interval greater than δ exists* **do**
 decision ()
 assign ObfusTran to the mesh and divide the range in half
 select one of the subintervals
 deduction ()
 CombFirst=combination(1st row mesh values)
 SumFirst=sum(CombFirst)
 CombSec=combination(2nd row mesh values)
 SumSecond=sum(CombSec)
 EffectiveWidth=apply (4.4)
 if $ICP(\phi)=UNSAT$ **then**
 find conflict-source s
 $S = S \cup s$
 if $S=entire\ state\ space$ **then**
 return UNSAT
 end
 else
 undo all decision and deduction after s
 $\phi=\phi \cap \bar{s}$
 end
 end if
 end
 end if
end
end while
return UNSAT

4.5 Circuit Implementation

The proposed parameter obfuscation techniques are implemented on an active band pass filter (BPF) and an operational amplifier. The selection algorithm is used to determine the transistor(s) to obfuscate, while the SMT algorithm is executed to determine the sizes of the *target transistors* and *obfuscation transistors*. The circuits are designed in a 180 nm technology using Cadence Virtuoso, where performance analysis is completed using Spectre.

4.5.1 Active Inductor Based Second-Order Band Pass Filter (BPF)

The schematic of a second order band pass filter is shown in Fig. 4.3. The BPF is composed of two resonator circuits that consist of capacitor C and two active inductors implemented by transistors M_1, M_2, M_5, M_6 and M_3, M_4, M_7, M_8 . Input matching is achieved through resistor R_{in} and a common gate transistor M_{in} . The source follower topology that consists of transistor M_{out} and resistor R_{out} form the output matching circuit. Transistors M_9 to M_{14} implement current sources.

The band pass filter is designed to operate with a center frequency F_c of 3 GHz, a maximum gain of 30 dB, and a bandwidth of 500 MHz. The parameter obfuscation technique is applied to mask the gain and the center frequency of the BPF. From the execution of the algorithm to select the transistor(s) to obfuscate, M_{12} was determined to mask the

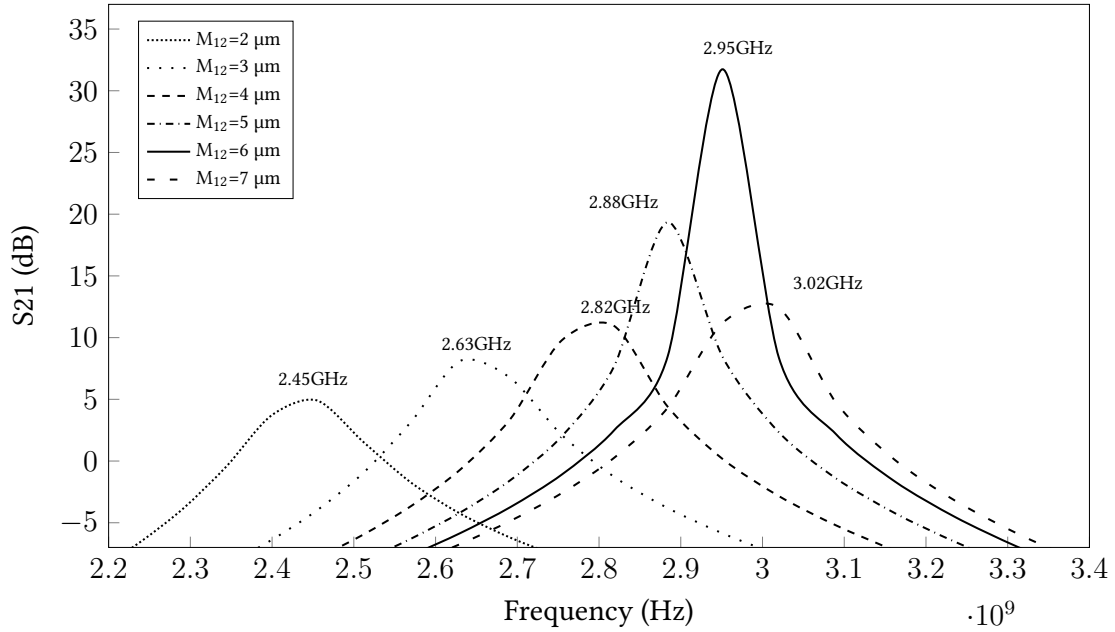


Figure 4.8: Variation in the center frequency of the BPF for different currents through transistor M_{12} . An AC input signal of 1 mV is applied to determine the gain.

center frequency F_c , while transistors M_{13} and M_{14} were determined to mask the gain A_v of the BPF. The variation in the center frequency F_c and the gain A_v of the BPF as a function of the width of transistor M_{12} and transistor M_{14} is shown in Fig. 4.8 and Fig. 4.9, respectively. The size of transistor M_{13} has the same effect on the gain A_v of the BPF as does the size of transistor M_{14} . Therefore, both M_{13} and M_{14} are obfuscated to mask the gain of the BPF.

Transistors M_{12} , M_{13} , and M_{14} of the BPF are obfuscated using both the vector and mesh-based techniques. The schematic representation of the key delivery circuit is shown in Fig. 4.10, where the activation of the transistors is controlled by an applied gate voltage through a decrypt circuit. For key bits that are set to logic high, V_{bias} is applied to the gate

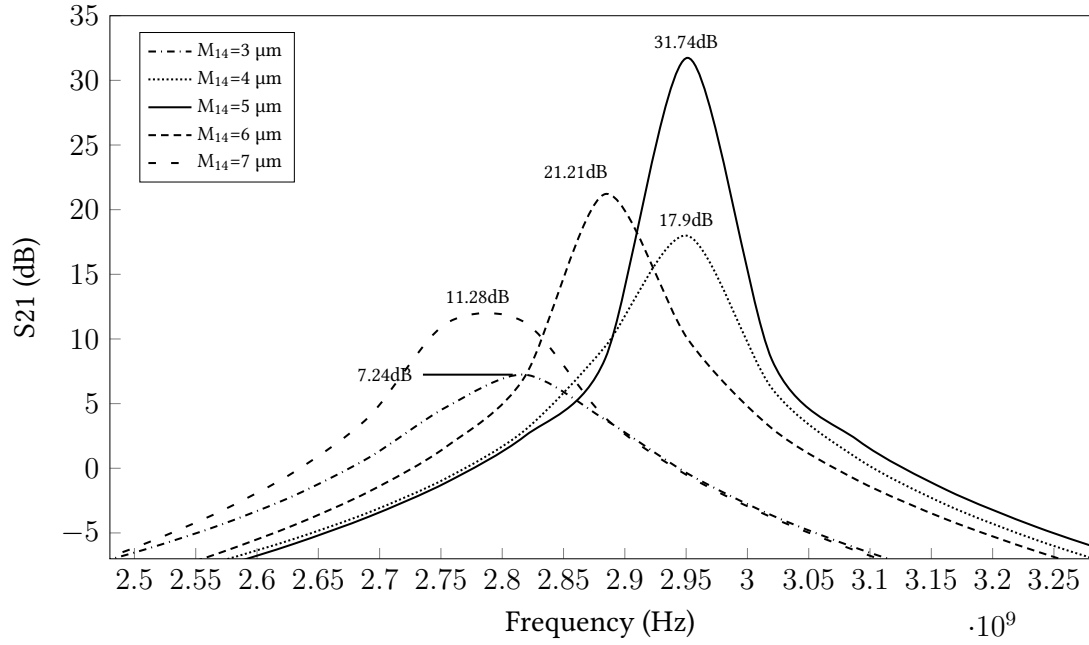


Figure 4.9: Variation in the gain of the BPF for different widths of transistor M_{14} . An AC input signal of 1 mV is applied to determine the gain.

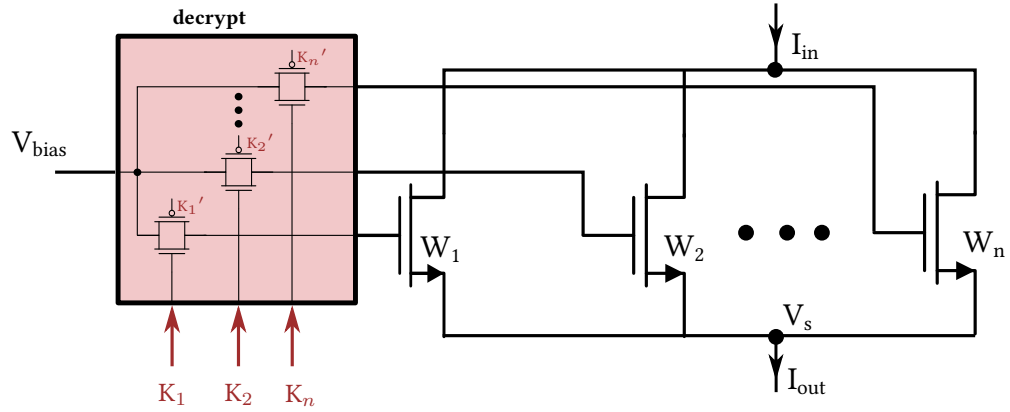


Figure 4.10: Schematic of the key-delivery circuit that activates or de-activates the param-eter obfuscation transistors based on the applied key bits.

of the corresponding *obfuscation* or *target* transistor. When the key bit is set to logic low, the corresponding transistor is off and no current flows. The total current (I_{in} or I_{out}) is equal to the sum of the currents through the activated paths.

Table 4.2: Characterization of the center frequency, gain, bandwidth, and area of a BPF after implementing vector-based and mesh-based obfuscation. An unobfuscated BPF is also characterized for comparison.

Parameter	Unobfuscated	Vector-based	Mesh-based
Center Frequency (F_c)	2.95 GHz	3.02GHz	2.88GHz
Gain (A_v)	31.74dB	31.98dB	33.31dB
3-dB Bandwidth	17.4 MHz	12 MHz	11.2 MHz
Area	$11\mu\text{m}^2$	$24.35\mu\text{m}^2$	$26.91\mu\text{m}^2$

For both the vector-based and mesh-based obfuscation of the BPF, transistors M_{12} , M_{13} , and M_{14} are each obfuscated with a 4 bit key, which results in a total key length of 12 bits. The attacker now has to determine the correct gain and center frequency to completely reverse engineer the BPF. The comparison of the critical parameters of an unobfuscated and obfuscated BPF are listed in Table 4.2.

Due to the additional obfuscation transistors and key decryption circuit, deviation in the performance of the BPF is observed. The performance parameters of the re-tuned obfuscated circuit match closely with the unobfuscated circuit, with results indicating no more than 2.3% deviation in the center frequency F_c and no loss in the gain A_v . However, the area of the BPF increases by 2.2x and 2.7x when implementing the vector-based and mesh-based obfuscation techniques, respectively.

4.5.2 Operational Amplifier

A two-stage operational amplifier (op-amp) with a topology as shown in Fig. 4.11 is considered. The first stage consists of a differential amplifier implemented by transistors

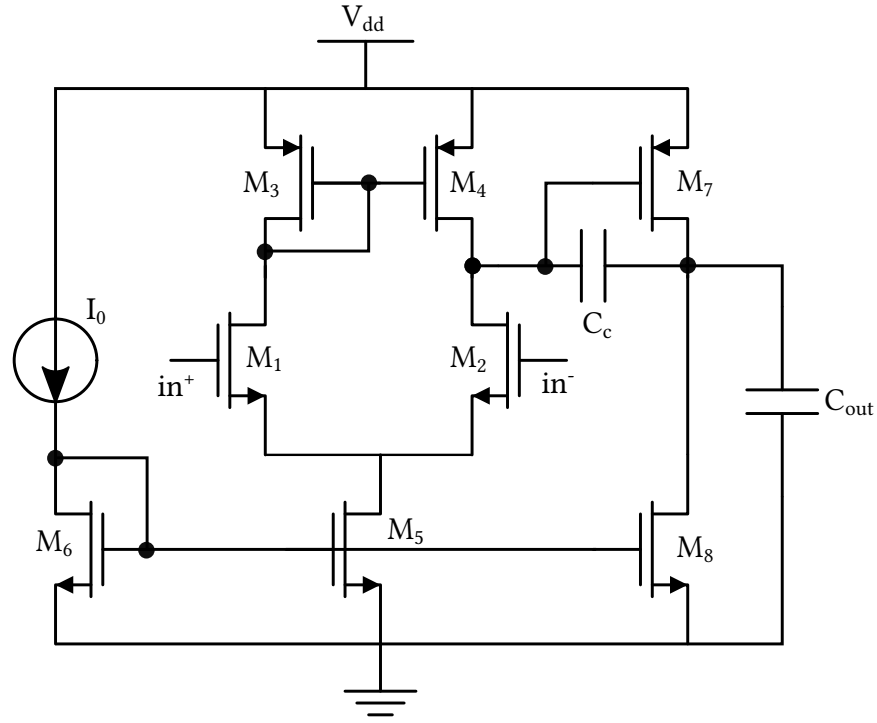


Figure 4.11: Circuit schematic of a two-stage operational amplifier.

M_1 to M_6 , while the second stage consists of a common source amplifier implemented by transistors M_7 and M_8 . The load capacitance C_{out} is set to 10 fF and the input common-mode voltage range is set between 0.8 V and 1.6 V. The operational amplifier is designed to operate with a 60 dB gain, unity gain bandwidth greater than 20 MHz, and power dissipation of less than 1 mW.

The parameter obfuscation techniques are implemented on the first and second stage of the op-amp to mask the total gain and the gain-bandwidth. From the execution of Algorithm 5, which selects transistor(s) to obfuscate, M_5 and M_8 were determined best suited to mask the gain of the amplifier. The change in the gain of the op-amp, defined by the S_{21} parameter as a function of frequency for a peak-to-peak AC input signal of 1 mV,

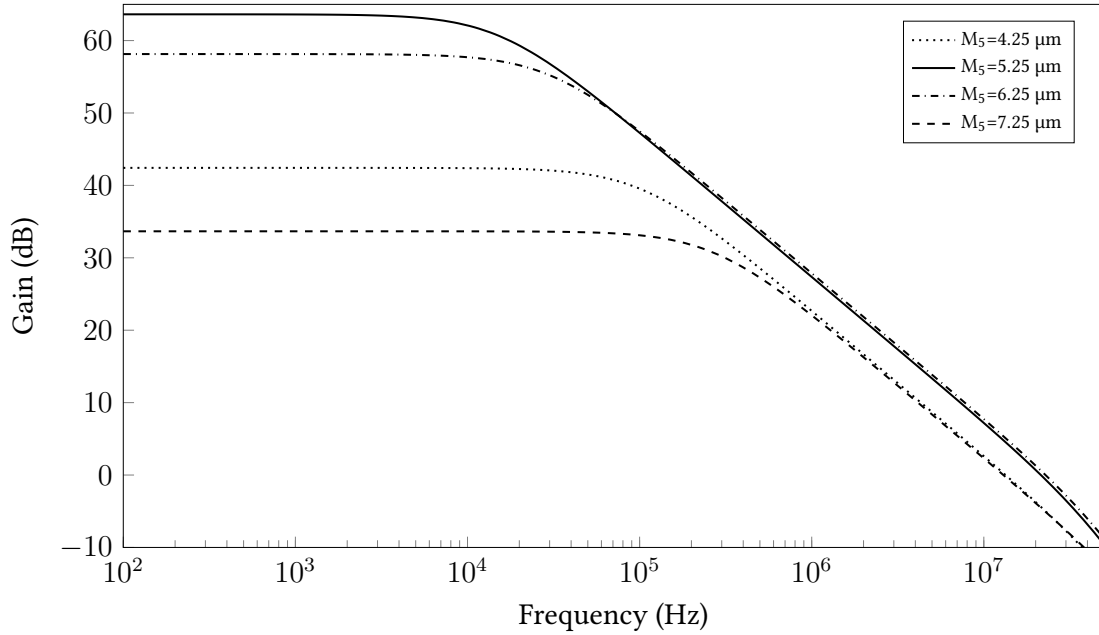


Figure 4.12: Variation in the gain of the op-amp for different widths of transistor M_5 . An AC input signal of 1 mV is applied to determine the gain.

is shown in Fig. 4.12 and 4.13 for different widths of transistors M_5 and M_8 , respectively.

Only when the size of transistors M_5 and M_8 are set to $5.25 \mu\text{m}$ and $35 \mu\text{m}$, respectively, is the maximum gain of approximately 63 dB achieved, while the remaining widths produce gains of less than 40 dB.

The op-amp is obfuscated using both the vector-based and mesh-based techniques, where transistors M_5 and M_8 are each obfuscated with a 5 bit key (total key size of 10 bits). For the mesh-based obfuscation of transistors M_5 and M_8 , two rows are included, where the first row implements 3 bits of the key and the other row the remaining 2 bits. Utilizing Algorithm 2, the sizes of the obfuscation transistors are determined such that only the correct key produces a gain of 63.6 dB and any incorrect key results in at least a 30%

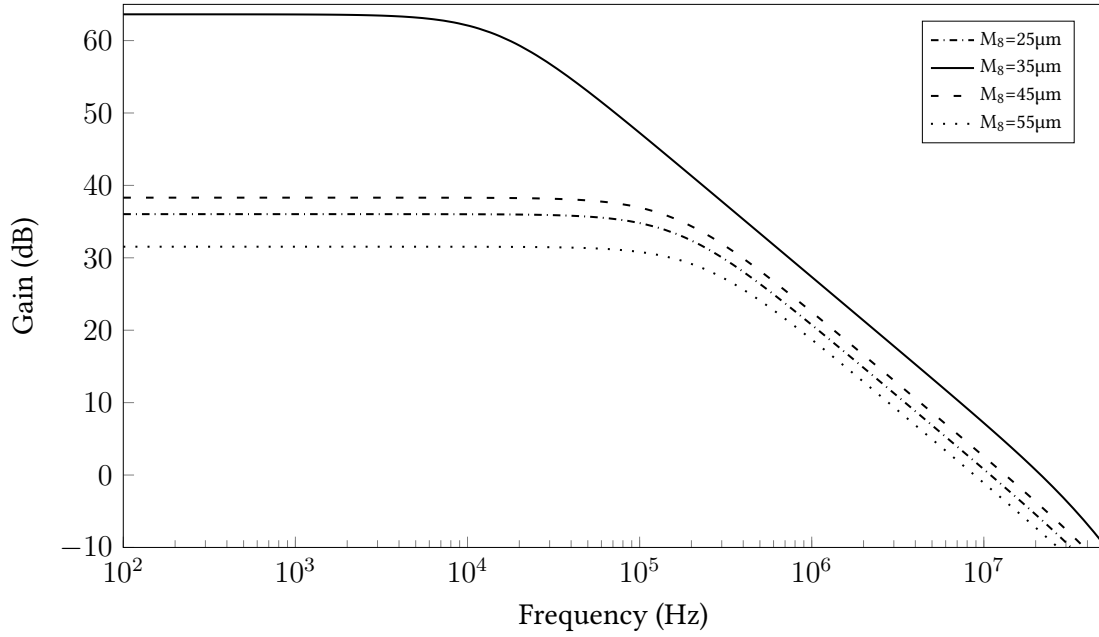


Figure 4.13: Variation in the gain of the op-amp for different widths of transistor M_8 . An AC input signal of 1 mV is applied to determine the gain.

degradation in the gain (dB) of the op-amp, which is achieved by sizing the obfuscation transistors of M_5 and M_8 to be at least 40% and 30% shifted from the correct target dimensions, respectively. The results from the characterization of the gain, gain-bandwidth, 3-dB bandwidth, power dissipation, and area of an obfuscated and unobfuscated op-amp are listed in Table 4.3 and indicate that there is a maximum deviation of 0.4 dB in the gain and 0.3 MHz in the unity gain bandwidth. However, due to the additional obfuscation circuitry, there is a 1.57x and 2.24x increase in the area of the op-amp after implementing the vector-based and mesh-based obfuscation techniques, respectively.

Table 4.3: Characterization of critical circuit parameters of an op-amp after applying the vector-based and mesh-based obfuscation techniques. The results are compared with an unobfuscated op-amp.

Parameter	Unobfuscated	Vector-based	Mesh-based
A_v	63.6 dB	63.8 dB	64.0 dB
GBW	21.6 MHz	21.3 MHz	21.4 MHz
3-dB BW	15.4 KHz	14.7 KHz	15.3 KHz
Power Dissipation (μW)	330 μW	337 μW	350 μW
Area	213.25 μm^2	334.75 μm^2	477.57 μm^2

4.6 Metric Evaluating the Security of Analog Obfuscation Techniques

The proposed parameter obfuscation technique is compared with current mirror-based obfuscation [192] and multi-threshold based obfuscation [193]. The threat models considered include an untrusted foundry and an untrusted end-user. The attacker is assumed to possess the circuit netlist and has knowledge of the target specifications through a datasheet or an active IC. In addition, access to the key and primary inputs is assumed, through which an attacker is capable of changing the key or primary inputs and observing the corresponding outputs. However, the attacker does not possess advance models or knowledge that facilitates further pruning of the key space. For the given threat model, the attacker is only able to execute a brute-force attack, where the metric to evaluate the security of the obfuscation techniques is the probability of determining the correct key.

The results from the comparison of the three analog obfuscation techniques are listed

Table 4.4: Analysis of the security provided by analog obfuscation techniques to an untrusted foundry and an untrusted end-user. The probability of determining the correct key is also evaluated.

Obfuscation Technique	Threat Model		BPF		Op-Amp	
	Untrusted Foundry	Untrusted End-user	No. of Keys	Probability	No. of Keys	Probability
Vector-based	✓	✓	12-bit	$\frac{1}{2^{12}}$	10-bit	$\frac{1}{2^{10}}$
Mesh-based	✓	✓	12-bit	$\frac{1}{2^{12}}$	10-bit	$\frac{1}{2^{10}}$
Current Mirror-based	✓	✓	Not-applicable		10-bit	$\frac{1}{2^{10}}$
Multi-threshold	✗	✓	16-bit	$\frac{1}{3^{16}}$	8-bit	$\frac{1}{3^8}$

in Table 4.4, where equal key lengths are applied for parameter obfuscation and for current-mirror based obfuscation. The search space for the multi-threshold based obfuscation technique is given by 3^P , where P is the number of transistors in the circuit. The probability of determining the correct V_T for all transistors is $\frac{1}{(3^P)}$. Therefore, even assuming there are multi-threshold devices available in a given fabrication process, the application of the multi-threshold obfuscation technique is limited to analog circuits where P is large. For the parameter and current-mirror based obfuscation techniques, the key size and the search space is only limited by a given area overhead. Therefore, a much larger number of combinations of parameter values is possible with a trade-off in increased area. The multi-threshold obfuscation technique does not protect analog ICs from an untrusted foundry as the V_T information of the transistors is provided to the foundry for fabrication. In addition, the multi-threshold obfuscation technique proposed in [193] does not address the possibility of different V_T transistor combinations producing the target specifications or even specifications that are close enough to the target.

Given the same key size, the probability of determining the correct key for both the parameter and current-mirror based obfuscation techniques is the same for the op-amp. However, the current-mirror based technique can not be applied to the BPF as no current mirrors are present in the circuit to obfuscate. In addition, due to key sharing of transistors among the branch currents, the number of combinations is reduced to $2^{\min(\text{branches}, \text{keysize})}$ as all the transistors in a given branch must be active for the current to flow through the branch. The vector-based and mesh-based obfuscation techniques provide flexibility in the key size, provide security against both an untrusted foundry and an untrusted end-user, and can be implemented on any analog circuit topology.

4.7 Discussions

The primary aim of the developed parameter obfuscation techniques is to thwart analog IP theft by making an attack extremely time consuming, complex, and expensive. In addition, the resources required to attack the protected analog circuit and determine the key must far outweigh the resources needed to design the analog circuit, which implies a higher cost to reverse engineer and counterfeit a circuit. Implementing the parameter obfuscation techniques increases the time and cost of determining the key. To increase the resilience of the parameter obfuscation techniques to overproduction, body voltages are also obfuscated along with biasing conditions at a cost of added area and complexity.

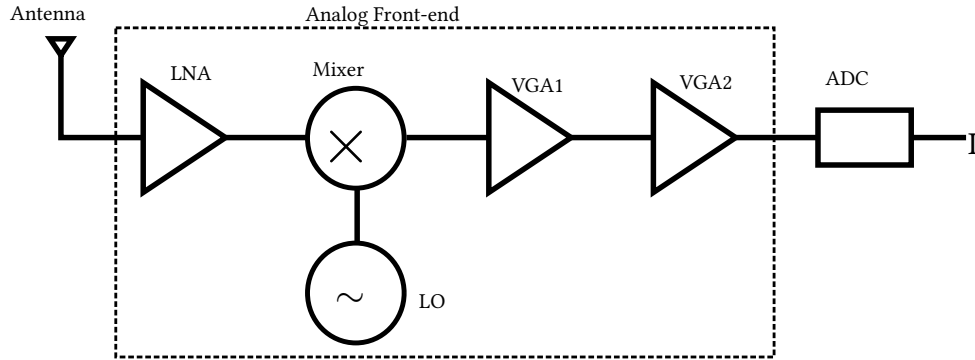


Figure 4.14: Block diagram of the direct conversion receiver.

The characterization of the area due to the implementation of the parameter obfuscation techniques is in comparison to an unobfuscated version of the circuit block, which results in a significant increase. However, the percentage of the area when considering the entire analog system or even an entire mixed signal IC is a fraction of the total. For the direct conversion receiver shown in Fig. 4.14, which is implemented in a TSMC 65 nm process, the variable gain amplifiers (VGAs) are obfuscated, masking the gain and the center frequency (F_c) of the analog front-end circuit. The receiver is designed to operate in the 2.5 GHz band with 100 MHz channel bandwidth. For an incorrect key, the gain and F_c of the analog front-end signal is degraded, which leads to bit errors in the signal I at the output of the analog to digital converter (ADC). Both variable gain amplifiers, VGA1 and VGA2, of the receiver are obfuscated with a 10-bit key, which results in a total key length of 20-bits. The total area of the sub-blocks, which includes the area of both active and passive devices, of the obfuscated and unobfuscated receivers is listed in Table 4.5. The results indicate that when comparing only the VGA1 block, the area increased by $4.4\times$ or 342%,

Table 4.5: Comparison of the total area (active and passive) of sub-blocks of an unobfuscated receiver and a receiver obfuscated with a 20-bit key.

Circuit	Unobfuscated Area	Obfuscated Area	% Area Increase ($\times$ area)
VGA1	$672 \mu\text{m}^2$	$2971 \mu\text{m}^2$	342% ($4.4\times$)
VGA2	$16434 \mu\text{m}^2$	$16434 \mu\text{m}^2$	0% ($1\times$)
Receiver - ADC	1.107044 mm^2	1.109343 mm^2	0.21% ($1.002\times$)
Receiver + ADC	1.22774 mm^2	1.230073 mm^2	0.18% ($1.001\times$)

but when compared to the entire analog front-end block, the total area increases by only 0.21%. In addition, for VGA2, which consists of two large metal-insulator-metal (MIM) capacitors, the obfuscation transistors and the corresponding lower metal layer interconnects are placed beneath the MIM capacitors, which results in no increase in the overall area. Furthermore, the key storage and key delivery circuit are integrated with the bias tuning circuit of the receiver, which results in no area penalty for the obfuscated receiver.

The relation between the key size and the active area of the vector obfuscated op-amp is shown in Fig. 4.15 for different percentage variation of both the *target transistor* size and the *obfuscation transistor* size. The active area of the analog circuit linearly increases with the key-size. In addition, for a fixed key-size, the active area of the analog circuit also increases linearly with the percentage variation of both the *target transistor* size and the *obfuscation transistor* size. However, the execution time of the SMT algorithm to determine the sizes of the *obfuscation transistors* increases exponentially with key-size as shown in Fig. 4.16, since the computational resources needed to solve for the dimensions of the *obfuscation transistors* increase at the rate of 2^n for an n -bit key. In order to reduce

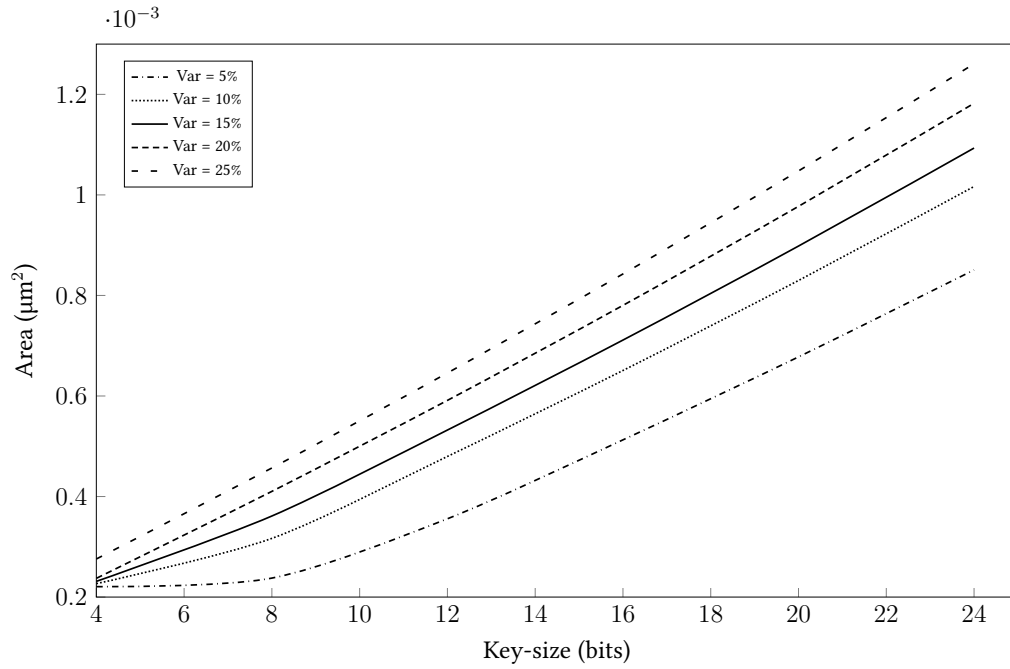


Figure 4.15: Relation between the key-size and the area of an op-amp obfuscated using the vector-based obfuscation technique for different percentage variation of both the target transistor size and the obfuscation transistor size.

the computational time for the same key-size, multiple transistors in the analog circuit are obfuscated. For example, the computational complexity for obfuscating a single transistor in an analog circuit secured with 20-bit key is 2^{20} , whereas the computational complexity is reduced to 2×2^{10} for obfuscating two transistors each with a 10-bit key. In addition, obfuscating multiple transistors has an added advantage of masking multiple performance parameters, and along with the key-sharing technique proposed in [119] further increases the security provided by an implemented obfuscation technique.

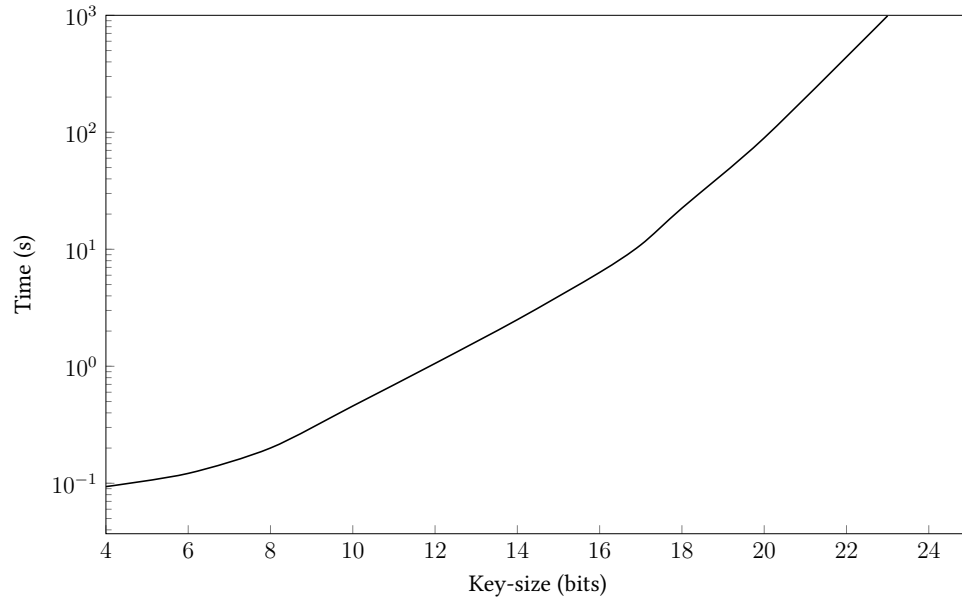


Figure 4.16: The relation between key-size and the execution time of the SMT algorithm to determine the obfuscation transistor sizes when implementing the vector-based obfuscation technique.

4.8 Conclusions

Parameter obfuscation is proposed to protect analog circuits against IP piracy attacks including reverse engineering, counterfeiting, and overproduction. Two techniques, vector-based and mesh-based obfuscation, are described, which provide a varying degree of security to an analog circuit. The vector-based obfuscation technique masks the transistor dimensions and is less complex to implement. The mesh-based obfuscation in addition to masking the transistor dimensions, the threshold voltage and the small signal parameters of the transistor/circuit are also masked, but is complex to implement and incurs a larger area overhead compared as to the vector-based obfuscation technique. The two

obfuscation techniques are implemented to mask the gain and center frequency of a BPF with a 12 bit key and the gain and gain bandwidth of an op-amp with a 10 bit key. The results of implementing the two obfuscation techniques on a BPF and op-amp indicate that the critical circuit performances are properly obfuscated with at least a 15% variation in the properties of the circuit when setting the correct and incorrect transistor sizes. To reduce the complexity of determining the sizes of the obfuscation transistors that ensure a single unique key and that incorrect keys produce a significant difference in the performance of the circuit, a satisfiability modulo theory (SMT) based search space exploration algorithm is proposed. The parameter obfuscation techniques are resilient against different attack vectors including brute-force and removal attacks, as well as more advanced techniques including SMT and modeling attacks. The performance and security analysis provided in this section verify that the developed parameter obfuscation techniques are effective in protecting analog circuits against IP piracy and theft. In addition, the SMT and transistor ordering algorithms prove to be efficient in reducing the design time and the number of recycles when implementing the parameter obfuscation techniques on an analog circuit.

Chapter 5

Simulation-based Bias Optimization

Algorithm

As technology scales below 65 nm, process variation has a more dominant affect on the reliability and robustness of an analog integrated circuit [194]. Technology scaling increases random device mismatches due to doping fluctuations, which affects the threshold voltage of the transistors and results in deviation in the performance targets of the circuit. In addition, the threshold voltage of the transistor is also affected by bias temperature instability (BTI) due to aging, which also impacts the performance of an analog circuit [195]. Post-silicon tuning methodologies are applied to negate the drift in the performances of an analog circuit due to process variations, aging, and obfuscation where the body voltage

of the transistor or biasing voltages and currents are tuned to properly calibrate the functionality of the circuit [196]. The voltage to the body of the transistor is modulated in the body biasing technique [197], which changes the threshold voltage (V_T) to calibrate the circuit behavior. One technique, adaptive body biasing (ABB) [198, 199], which applies both forward body biasing (FBB) and reverse body biasing (RBB) to a group of transistors, allows for variable tuning of the threshold voltage of the grouped transistors to achieve the target circuit performances. In addition to technology scaling, the integration of the obfuscation transistors and the key delivery circuit to implement the key-based parameter obfuscation technique introduces parasitic impedances that result in a deviation in the performance of an analog circuit from specified values. In this chapter, a simulation-based optimization algorithm is proposed to tune the biasing conditions and transistor body voltages, which mitigates the effects of both the parasitic impedance of the circuit and any variation due to the implementation of the obfuscation circuitry

5.1 Simulation Based Multivariate Bias Tuning

To mitigate the drift in the performance of an analog circuit from target values due to the implementation of the obfuscation technique, process variation, and aging, a simulation-based multi-variate optimization methodology is developed to tune the body voltage of

the transistors and/or the biasing voltages and currents of the circuit. Simulated annealing (SA) is applied in conjunction with SPICE simulation of the circuit in an iterative optimization loop. Optimal bias voltages, bias currents, and body voltages $\vec{B} = \{B_1, B_2, B_3, \dots, B_k\}$ are determined through the optimization algorithm, and a cost function Φ is evaluated from results produced through SPICE simulation. Execution of the SPICE-based simulation loop returns performance values $\vec{P} = \{P_1, P_2, P_3, \dots, P_l\}$ that are used to calculate the cost function Φ . Φ is computed as a minimization of the difference between the simulated performance $\vec{P}_{simulated}$ and the target performance $\vec{P}_{target}$ of the circuit.

The pseudo-code of the simulated annealing based optimization routine is provided as Algorithm 7. The input to the algorithm includes the target performance values $\vec{P}_{target}$, the initial temperature T , the number of times the temperature is decreased M , the number of neighbors searched around a candidate point N , the cooling rate α , and the circuit netlist (*circuit*). Based on the random initial values of $\vec{B}$ and T , an initial solution $\vec{P}$ is determined, and the cost function $CF_{current}$ is calculated. Next, a set of N random points in the search space that are slightly modified from the current values of $\vec{B}$ are explored, and the cost function for each point is computed. If the explored solution results in a lower cost function than the current solution (minimization problem) or if, on calculation of (5.1), the acceptance probability F is greater than that of a randomly selected point, then a decision to consider the selected solution is made. The temperature T is decreased at a rate α after each iterative selection of N points for M number of iterations, which

Algorithm 7: Simulated annealing algorithm that tunes the bias voltage and body voltage to properly calibrate the performances of an analog circuit.

Input: P_{target} , M , N , T , α , circuit
Output: B_{best}

$B_{current} = Rand()$
 $CF_{current} \leftarrow Calculate(B_{current}, circuit)$
 $CF_{best} = CF_{current}$
 $B_{best} = B_{current}$

for i **to** M **do**
 $T_{current} \leftarrow T$
 for j **to** N **do**
 $B_j \leftarrow SelectBestNeighbor(B_{current}, circuit)$
 $CF_j \leftarrow Calculate(B_j, circuit)$
 if $CF_j \leq CF_{best}$ **then**
 $B_{current} \leftarrow B_j$
 if $CF_j \leq CF_{best}$ **then**
 $B_{best} \leftarrow B_j$
 end
 end
 else if $Formula > Rand()$ **then**
 $B_{current} \leftarrow B_j$
 end
 end
 $T_{current} = T_{current} \cdot \alpha$
end

return B_{best}

Function $Calculate(B, circuit)$
 $sch \leftarrow schematic(circuit)$
 $net \leftarrow netlist(sch)$
 $\vec{P}_{simulated} = SpiceSimulation(B, net)$
 $CF = \Phi = |\vec{P}_{target} - \vec{P}_{simulated}|$
 return CF

results in a total of $M \times N$ number of iterations. The algorithm returns the best determined values of the biasing voltages, biasing currents, and body voltages ($\vec{B}_{best}$) that produce the lowest cost function CF_{best} , which indicates circuit performance parameters close to the target values.

$$F = \frac{1}{e^{\frac{CF_{current} - CF_i}{T}}} \quad (5.1)$$

The simulation-based optimization algorithm is applied to both an obfuscated BPF and an obfuscated op-amp, where the target circuit specifications are met after tuning the biasing points post-obfuscation. In addition, the optimization technique is applied and the circuit is analyzed to determine the body voltages and biasing voltages that result in the target performances at different process corners. For the BPF, NMOS transistors M_1 to M_4 and PMOS transistors M_5 to M_8 are grouped into two separate domains of body voltages. The two body voltage domains, V_{Tp} and V_{Tn} , as well as the biasing voltages V_b , V_{b1} , V_{g1} , V_{g2} , and V_{g3} are determined and set using the simulation-based optimization algorithm. The cost function of the BPF is to minimize the sum of the error between the target and simulated gain A_v and center frequency F_c .

For the op-amp, the simulation-based optimization algorithm is applied to tune the bias current I_0 and the body voltages V_{Tp} and V_{Tn} such that the gain and unity gain bandwidth match the target specifications. The NMOS transistors M_1 , M_2 , M_5 , M_6 , and M_8

and PMOS transistors M_3 , M_4 , and M_7 are grouped and tied to the V_{Tn} and V_{Tp} body voltage domains, respectively. The cost function is computed with the goal of minimizing the error between the target and simulated op-amp gain and unity bandwidth (UBW).

The resulting body voltages, bias voltages, and bias currents after executing the simulation-based optimization algorithm on the BPF and the operational amplifier are listed in Table 5.1. The simulated annealing algorithm was implemented in Python 2.7.5 and the simulation methodology was executed using the Cadence SKILL and OCEAN scripting languages. The number of iterations of the simulation-optimization loop for the BPF is set to 500, where the temperature is decreased 50 times ($M = 50$) from 1000°C to 0°C at a rate of 15% ($\alpha = 0.85$) per iteration. For each decrease in temperature ($M = 50$ iterations), the 10 nearest neighbors are searched ($N = 10$). The bias voltages V_b , V_{b1} , V_{g1} , V_{g2} , and V_{g3} are set in the range of 0.4 V to 1.2 V. The tunable range of the PMOS body voltage V_{Tp} is between 1.1 V to 1.8 V, while the NMOS body voltage V_{Tn} is tuned in the range of 0 V to 0.6 V. To prevent latchup, the substrate current is analyzed by varying the body voltage for fixed gate and source/drain voltages for both the PMOS and NMOS transistors. From the analysis, the substrate current exponentially increases for $V_{Tp} \leq 1.1$ V and $V_{Tn} \geq 0.6$ V. Therefore, the lower range of V_{Tp} is set to 1.1 V while the upper range of V_{Tn} is limited to 0.6 V. The biasing voltages and the body voltages of the BPF are determined by the optimization algorithm for the five process corners (TT, FS, SF, SS, and FF) that account for PMOS and NMOS transistors with typical (T), fast (F), and slow (S) operation and for the

case of an unobfuscated, vector obfuscated, and mesh obfuscated BPF, with results as listed in Table 5.1. The results indicate that there is no more than a 2.75 dB difference in the gain, 4.75% difference in the center frequency, and 10 MHz difference in the 3-dB bandwidth of the obfuscated BPF as compared to the unobfuscated BPF. For the cost function (CF) of the BPF calculated using 5.1, the critical performance parameters considered include the center frequency F_c and gain A_v . Therefore, the resulting difference between the target and simulated F_c and A_v is weighted by a factor of ten, which results in a greater value in the calculated CF as compared to the value produced by the bandwidth (BW) parameter. The optimization algorithm determines the tuning voltages and currents that minimize the error between the target and simulated values of the parameters with the greatest impact on the cost function, which in this case are the weighted center frequency and gain.

For the operational amplifier, the number of iterations of the simulation-based optimization loop is set to 70. The optimum biasing current I_0 , and the body bias voltages V_{Tp} and V_{Tn} are determined for the five process corners (TT, FS, SF, SS, and FF) and for an unobfuscated, vector obfuscated, and mesh obfuscated op-amp. The temperature is decreased 10 times ($M=10$) from 1000°C to 0°C at a rate of 15% ($\alpha = 0.85$) per iteration, and for each iteration of decreasing temperature, the 7 nearest neighbors are searched ($N=7$). The range of I_0 is set to between 10 μ A and 50 μ A with a maximum step size of 1 μ A. The tunable range of the PMOS body voltage V_{Tp} is set to between 1.1 V and 1.8 V, while the

Table 5.1: Results of applying the simulation-based optimization algorithm that tunes the biasing conditions and body voltages of a BPF and op-amp to mitigate parasitic and process variations when applying vector-based and mesh-based obfuscation.

Circuit	Parameter		Unobfuscated					Vector-based Obfuscation					Mesh-based Obfuscation				
			TT	FS	SF	SS	FF	TT	FS	SF	SS	FF	TT	FS	SF	SS	FF
BPF	Tuning Parameter	Vb (V)	1	1	1	1	0.74	1	0.9	0.8	0.8	1.05	1	0.75	0.94	1	1
		V_{b1} (V)	1	1	1	1	0.76	1	1	1	0.8	1	1	1	1	1	1
		V_{g1} (V)	0.6	0.56	0.64	0.63	0.51	0.68	0.6	0.72	0.67	0.57	0.66	0.66	0.72	0.57	0.59
		V_{g2} (V)	0.6	0.59	0.65	0.76	0.7	0.7	0.73	0.78	0.8	0.64	0.7	0.62	0.74	0.72	0.61
		V_{g3} (V)	1.2	1.15	1.16	1.06	1.04	1.03	1.03	1.06	1.18	1.12	1.09	1.03	1.06	1.03	1.18
		V_{Tp} (V)	1.8	1.8	1.8	1.5	1.8	1.8	1.8	1.7	1.8	1.8	1.8	1.8	1.8	1.8	1.8
		V_{Tn} (V)	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0.1	0
	Performance Parameter	A_v (dB)	31.74	29.0	30.86	30.55	31.07	31.2	29.2	29.12	32.0	29.0	32.3	28.2	29.8	29.55	30.97
		A_v Error (dB)	0	2.74	0.88	0.89	0.67	0.54	2.54	2.64	0.26	2.74	0.56	2.24	1.94	2.19	0.77
		F_c (GHz)	2.95	3.01	3.09	2.95	3.02	3.02	2.95	3.1	2.9	3.09	2.9	2.95	3.09	2.95	2.95
		F_v Error (%)	0	2.03	4.75	0	2.3	2.37	0	5.08	1.69	4.75	1.69	4.75	4.74	0	0
Op-Amp	Tuning Parameter	I_0 (uA)	20	20	20	20	20	19	20	22	22	18	20	20	24	20	20
		V_{Tp} (V)	1.8	1.4	1.8	1.7	1.8	1.8	1.6	1.8	1.8	1.7	1.8	1.8	1.7	1.8	1.8
		V_{Tn} (V)	0	0	0.4	0	0	0	0	0	0	0	0.5	0	0.5	0.5	0.6
	Performance Parameter	A_v (dB)	63.6	64.1	63.7	63.8	63.5	63.8	63.4	63.9	63.3	64.3	64.0	64.9	64.0	63.4	63.1
		A_v Error (dB)	0	0.5	0.1	0.2	0.1	0.2	0.2	0.3	0.3	0.7	0.4	1.3	0.4	0.2	0.5
		UBW (MHz)	21.6	21.9	22.9	21.9	21.1	21.3	22.3	22.2	21.5	22.5	21.4	21.6	23.8	21.1	22.8
		UBW Error (MHz)	0	0.3	1.3	0.3	0.5	0.3	0.7	0.6	0.1	0.9	0.2	0	1.7	0.5	1.2
		3-dB BW (kHz)	15.4	16.2	14.6	16.9	14.1	14.7	16.3	15.7	16.5	15.7	15.3	14.8	16.7	15.8	16.4
		3-dB BW Error (kHz)	0	0.8	0.8	1.5	1.3	0.7	0.9	0.3	1.1	0.3	0.1	0.6	1.3	0.4	1

NMOS body voltage V_{Tn} is tuned in the range of 0 V to 0.6 V to prevent latchup. The step size for both V_{Tn} and V_{Tp} is 0.1 V. The results of executing the optimization algorithm on the op-amp are listed in Table 5.1, where the determined I_0 and body voltages produced no more than a 0.7 dB difference in the gain, 1.7 MHz difference in the unity-gain bandwidth, and 1.5 kHz (less than 10%) error in the 3-dB bandwidth as compared to an unobfuscated op-amp.

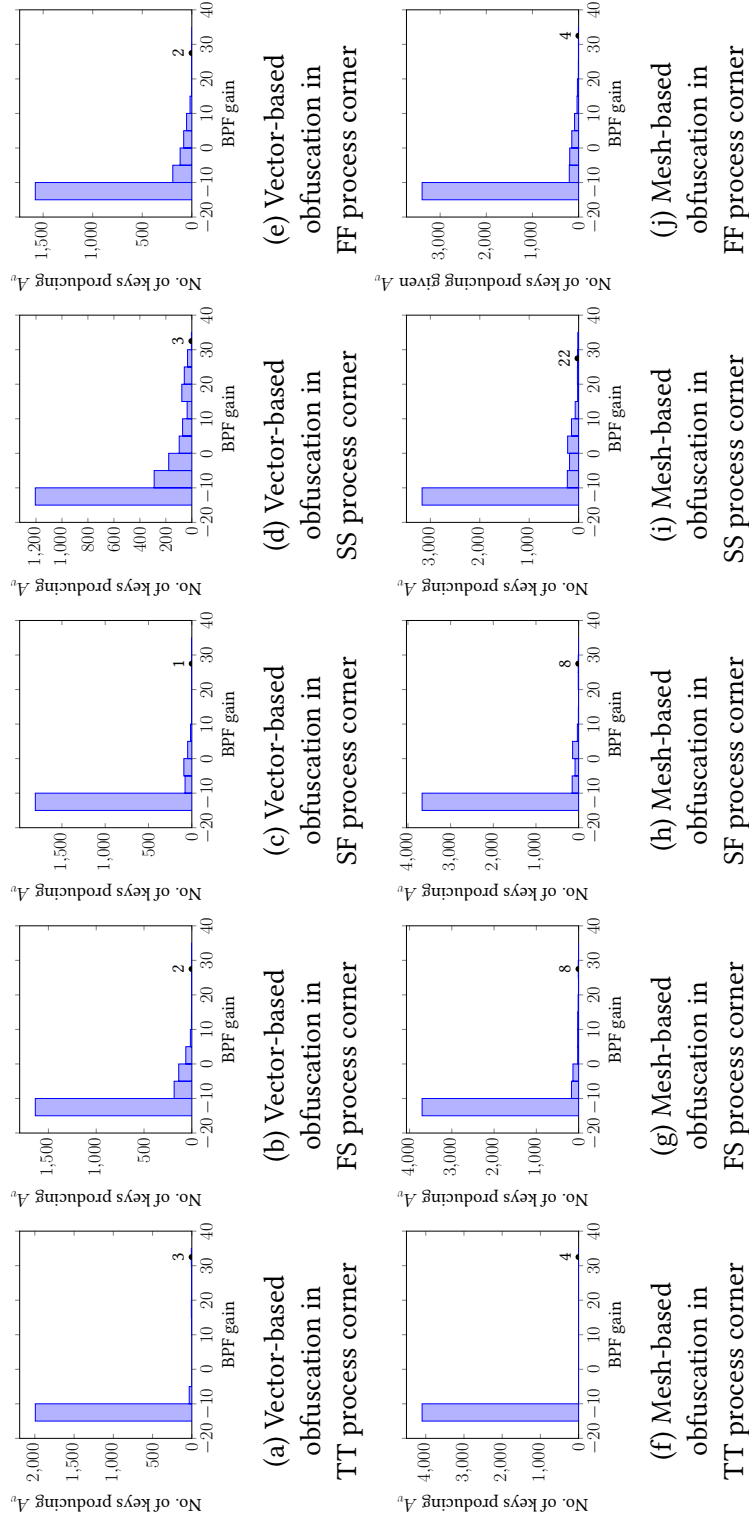


Figure 5.1: Histogram of the gain of a BPF for different applied keys, where the gain is in the range of -15 dB to 35 dB. The BPF is obfuscated using the vector-based and mesh-based techniques, with analysis performed on the obfuscated BPF for different process corners.

5.1.1 Considering Parameter Tuning for Variation when Implementing the Obfuscation Techniques

The effect of PVT variations and the implementation of the biasing compensation technique leads to drift in the performance of an analog circuit for a given applied key, which potentially results in the circuit performing close to the target specifications for incorrect keys and the generation of multiple correct keys. To prevent the application of incorrect keys from producing circuit performance values close to the target specifications, an SMT algorithm is developed and executed to size the obfuscation transistors such that any incorrect key results in an adequate difference in the performances parameters from the target values. For the BPF and the op-amp, a 15% difference in the effective transistor widths between the correct and incorrect keys results in at least a 25% error in the gain and 10% error in the center frequency of the BPF and at least a 43.7% error in the gain of the op-amp after applying the obfuscation techniques and characterizing across the five process corners (TT, FS, SF, SS, and FF).

The number of keys that produce a given gain for the BPF after implementing the vector-based and mesh-based obfuscation techniques and across different process corners is shown in Fig. 5.1. The histograms are composed of bins of 5 dB increments that include the subset of keys that produce a given gain in the range of -15 dB to 35 dB. Although a subset of keys exist within the bin containing the correct key, the gain produced by the

incorrect keys varies by at least 2.5 dB from the target gain of the BPF for all corners. In addition, the center frequency F_c of the BPF for an incorrect key varies by at least 2.7 MHz (10%) from the target F_c . For the worst-case conditions, where the BPF is obfuscated using the mesh-based technique and is characterized in the SS process corner, the number of keys that produce a gain between 25 dB and 30 dB is 22, as shown in Fig. 5.1i. The correct key produces a gain of 29.55 dB (30.02x), while the remaining 21 incorrect keys produce gains that are at least 2.5 dB less than the target gain (27.03 dB or 22.46x), which equates to at least a 25.2% reduction in the maximum gain of the BPF. The closest gain of the BPF to the target of 29.55 dB falls within the 30 dB to 35 dB bin, with a value of 33.07 dB (45.03x). The resulting gain is 52.4% (3.52 dB) greater than that of the target value. For a given application, where a BPF is required, gains greater than the target gain result in frequency and phase distortions of the output signal and force one or more transistors to operate in an incorrect region. Similar trends are observed for both the vector- and mesh-based obfuscation techniques and across process corners, which ensures that a unique key produces the target output performances at all corners while incorrect keys result in a significant difference in the performance of the BPF even when considering process variations.

In addition, the impact of the bias current I_0 on the gain of the op-amp shown in Fig. 11 is analyzed, with the objective being to assure that only the correct key produces the desired gain for all values of I_0 and at all PVT process corners. For the analysis, the size

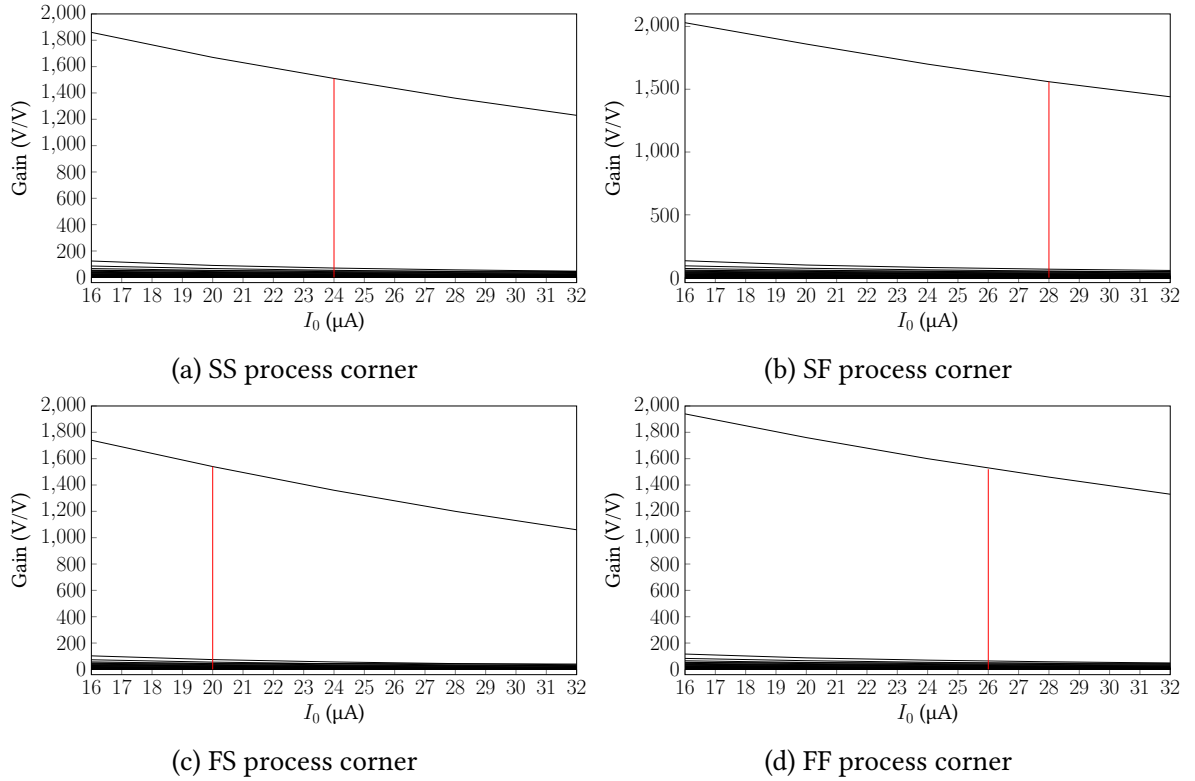


Figure 5.2: Analysis of the variation in the gain of the op-amp for all 2^{10} effective sizes of M_5 and considering a 25% tuning range of bias current I_0 . The analysis is performed in the SS, SF, FS, and FF process corners. The vertical red line represents the current that produces the target gain of ~ 1500 V/V.

of transistor M_5 is masked with a 10-bit key using the vector-based technique, which obfuscates the gain of the op-amp. The bias current I_0 is varied by $\pm 30\%$ of $24 \mu\text{A}$, which was determined for the TT process corner, with the results of sweeping I_0 to tune the gain to 63.6 dB shown in Fig. 5.2. The results indicate that when the sizes of the obfuscation transistors are designed properly, the biasing circuit tunes the performance of the op-amp to the target value for only the correct key. For an incorrect key, the bias tuning circuitry is not capable of compensating for the degraded performance of the op-amp at any corner.

The analog circuit is configured with the correct calibration voltages and/or currents

only while in test mode. In most analog circuits, the calibration circuitry is only accessible and configurable during testing, and is disabled while in normal operation. Therefore, an attacker possessing the oracle analog IC must enter the test mode to modify the calibration voltages and/or currents. For the assumed threat model, where an attacker possesses the circuit netlist, the attacker must determine both the effective transistor sizes as well as the calibration voltages and/or currents, which, therefore, significantly increases the search space. For the BPF, considering both the tuning and obfuscation transistor sizes as unknowns, the total number of unique combinations is approximately $13 \cdot 10^9$.

For circuits where the acceptable performance range is large, obfuscating the calibration circuit provides additional benefits to obfuscating the transistor sizes of the analog circuit. Obfuscating the calibration circuitry allows for the minimization of the effects of PVT variation and mismatch when a correct key is applied to ensure that the analog circuit operates within an acceptable performance range. Any incorrect key inaccurately compensates for PVT variations and mismatches and, therefore, results in severe degradation in the performance of the circuit. The flexibility of the proposed technique, including the proposed obfuscation circuit, is not limited to the obfuscation of only the analog circuit blocks, but is also applicable to the peripheral biasing and/or calibration circuitry.

5.2 Conclusions

The effects of parasitic and process variations are mitigated by a simulation-based optimization technique that tunes the biasing conditions (voltages and currents) and the body bias of the transistors to obtain the desired circuit performances. Execution of the simulation-based optimization algorithm determined the biasing conditions and body voltages in 500 iterations for the BPF, and 70 iterations for the op-amp. The implementation of simulation-based optimization algorithm along with the SMT and transistor ordering algorithms presented in Chapter 4 prove to be efficient in reducing the design time and the number of recycles when implementing the parameter obfuscation techniques on an analog circuit.

In addition, to increase the resilience of the parameter obfuscation techniques to over-production, body voltages are also obfuscated along with biasing conditions at a cost of added area and complexity. The results listed in Table 5.1 indicate that the biasing voltages of the BPF across fabricated dies vary for different process corners and, therefore, require distinct keys to properly obfuscate the biasing points. As a result, even though the adversary is able to determine the correct obfuscation key that is applicable to all the obfuscated analog IC, the adversary must determine the biasing key for each analog IC, adding to addition complexity, resources, and time.

Chapter 6

Security of Mixed Signal Circuits

The security of analog mixed-signal (AMS) circuits is of importance as 25% of the reported counterfeit incidences [11] occur in AMS systems. In addition, there is a large increase in demand for analog mixed-signal circuits with varied capabilities due to a growing demand for Internet of Things (IoT) devices, which are used in a wide array of applications from household products to defense systems. According to an IHS report, the number of IoT devices is estimated to increase by around 200% from 2019 to 2025 [85]. As more and more IoT devices are produced, securing the circuits against hardware security threats including counterfeiting, overproduction, reverse engineering, and Trojan insertion becomes challenging. As AMS systems are vulnerable to a wide range of threat vectors, the unsecured analog circuit blocks also present an exploitable point of entry to the IC even when the digital IP is protected.

To address the growing need of protecting AMS IP, this chapter presents a novel methodology to concurrently secure both analog and digital circuit blocks. The analog block is protected with a parameter based obfuscation technique [191] that masks the dimensions of the transistors used to implement vital components of the analog circuit. The analog circuit only operates as intended when the correct key is applied. The digital portion of the AMS circuit is protected with two different logic locking strategies; the XOR methodology [109] and the modified logic cone methodology [115]. The correct functionality of the digital block is also dependent on the application of the correct key to the circuit.

An analysis of the security of independently locked analog and digital circuit blocks is provided. An attack that isolates the analog and digital circuit blocks is described, limiting the effective key space of the secured circuit. The analog and digital blocks are modified to include correlated key dependencies within the circuit, which results in an increase in the level of difficulty to attack the IC. Both the analog and digital circuits are attacked with a satisfiability modulo theory (SMT) based technique that explores the logical and functional search space generated by a 16-bit key, where the attack is characterized by the number of iterations required to determine the correct key. The primary contributions of the work include:

1. protection of analog mixed-signal circuits by obfuscating both the analog and digital sub-components,

2. an analysis of the security of the analog mixed-signal circuit using a SMT based search space exploration technique, and
3. the development of a novel methodology to generate functional and behavioral dependencies between the analog and digital domains that results in an increase in the adversarial key search space.

6.1 Threat Model

The threat model for this work assumes that an adversary possesses the tools and knowledge necessary to reverse engineer the IC and produce an extracted locked netlist representation of the circuit. To obtain the logically functional representation of the circuit, an adversary must determine the correct keys for the obfuscated portions of the IC. In addition, an adversary is assumed to possess an activated IC, which is used to obtain input-output pairs that allow for the efficient pruning of the key space when applying a SAT based attack [166]. An adversary is assumed to have access to a second activated IC, which is used to modify a subset of the applied key bits. The adversary also has complete access to the scan chain of the IC, which allows for the read-out of the internal circuit state.

6.2 Vulnerabilities of Independently Secured Analog and Digital Circuit Blocks

Research on protecting ICs from a multitude of threats has resulted in methodologies to independently secure analog and digital circuits. In a typical AMS IC pipeline, multiple components of the circuit must be secured against reverse engineering. A discussion on potential vulnerabilities of independently securing the analog and digital blocks is provided in this section. An overview of the AMS pipeline used as an example throughout the chapter is described in Section 6.2.1. A description of the implemented obfuscation technique is provided in Section 6.2.2. An analysis of an attack to determine the key of the analog and digital circuits through isolation of each block is described in Section 6.2.3.

6.2.1 Proposed AMS Circuit Implementation

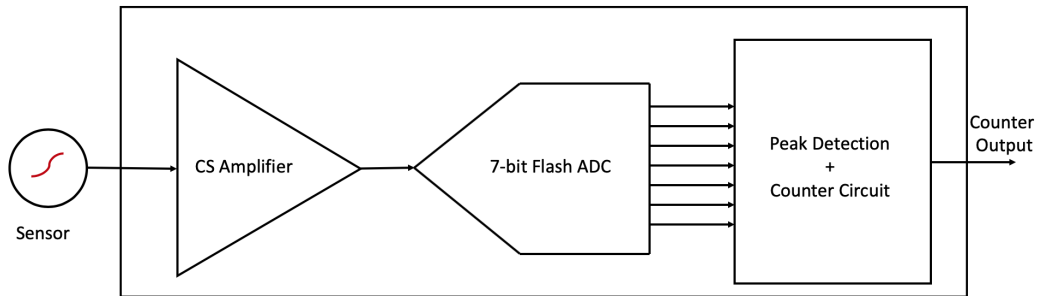


Figure 6.1: AMS circuit obfuscated with the proposed technique.

A peak detection and counting circuit is considered, which consists of an analog front

end, analog-to-digital converter, and digital back end as shown in Fig. 6.1. The peak detection and counting circuitry is applicable to a wide-range of applications including heart-rate monitoring devices, mass spectrometers, X-ray machines, and image processing applications. The analog front end consists of a common-source amplifier with a diode connected load used in the detection and amplification of a low-voltage output signal from a sensor. The output of the amplifier is applied to a 7-bit flash analog-to-digital converter (ADC). The output of the ADC is connected to a back-end digital computation block comprising of a peak detection circuit and a counter. The two most significant output bits from the ADC, referred to as Bit-1 (B1) and Bit-2 (B2), are applied to the peak detection circuit, where a bit flip from 0 to 1 on B1 implies the detection of a peak over a set threshold voltage and a bit flip from 1 to 0 of B2 returns the circuit to an active detection state.

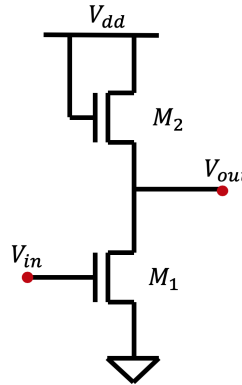


Figure 6.2: Schematic of the common-source amplifier with diode-connected load.

6.2.1.a Common Source Amplifier With Diode-Connected Load

The schematic of the presented common-source (CS) amplifier is shown in Fig. 6.2. Due to challenges in fabricating large on-chip resistors that fall within a tight tolerance, diode connected loads are preferred. The gain of the common source amplifier is given by

$$A_v = -g_{m1} \cdot \left(\frac{1}{g_{m2} + g_{mb2}} || r_{o2} || r_{o1} \right), \quad (6.1)$$

where g_{m1} and g_{m2} are the small signal transconductances of transistors M_1 and M_2 respectively, g_{mb2} represent the back-gate transconductance of the transistor M_2 , and r_{o1} and r_{o2} denotes the output resistances of transistors M_1 and M_2 respectively. Ignoring channel length modulation, $r_{o1} = r_{o2} = \infty$ and substituting formula for $g_m = \sqrt{2 \cdot \mu_n C_{ox} \frac{W}{L} \cdot I_D}$ (where μ_n is the mobility of electron, C_{ox} is the oxide capacitance of the transistor, $\frac{W}{L}$ represents the transistor width over length ratio, and I_D is the drain current flowing through transistor M_1 and M_2), (6.1) is simplified to

$$A_v = -\frac{1}{1 + \eta} \sqrt{\frac{\left(\frac{W}{L}\right)_1}{\left(\frac{W}{L}\right)_2}}, \quad (6.2)$$

where η is the back-gate transconductance parameter and $\left(\frac{W}{L}\right)_1$ and $\left(\frac{W}{L}\right)_2$ represent the width over length ratios of transistors M_1 and M_2 , respectively. From (6.2), the amplifier

gain is neither a function of the bias current nor the input signal, which results in a linear amplifier gain, tolerance to input and output voltage level fluctuations, a high input resistance, and a high open loop-gain. In addition, the gain of the amplifier is a function of the transistor sizes, which provides a direct design parameter available to obfuscate the amplifier gain. In the obfuscated circuit, the common-source amplifier is designed to produce a gain of 8x when biased by a DC voltage of 0.45 V for an input signal with a peak-to-peak amplitude of no more than 0.04 V.

6.2.1.b 7-Bit Flash Analog-to-Digital Converter

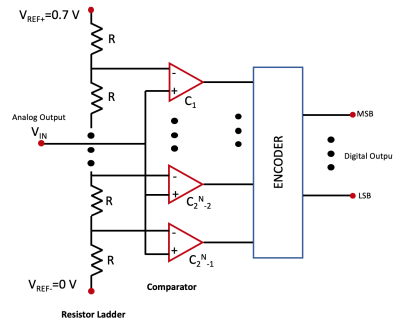


Figure 6.3: Schematic representation of the flash ADC.

The amplified signal is digitized using a 7-bit flash ADC as shown in Fig. 6.3. The ADC consists of a 2^N resistive divider providing the reference voltage for the $2^N - 1$ comparators. A logic 1 is produced when the analog voltage is higher than the set reference voltage applied to the comparator. Otherwise the output from the individual comparators is a 0. The combined outputs from all the comparators results in a thermometer code, which is then translated to the appropriate digital output code using an encoder.

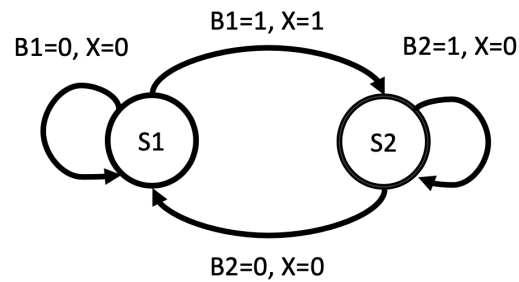


Figure 6.4: FSM of the peak detection and counting circuit.

6.2.1.c Peak Detection and Counting Circuit

The peak detection circuit connected to the output of the ADC is set to a voltage threshold of 0.6125 V, which determines if the most significant bit of the ADC (B1) transitions to a logic 1. A state machine diagram of the circuit is shown in Fig. 6.4, where state *S1* signifies the system is waiting for the detection of a peak and *S2* signifies the state of the system once a peak has been detected. The circuit only detects another peak after the voltage falls below a second set threshold (0.525 V for the designed circuit), which is represented by the second most significant bit of the ADC (B2).

Once a peak is detected, the output signal *X* is used to advance a 3-bit counter. By monitoring the output of the counter, the circuit is able to detect peak bursts or signal abnormalities if a large number of peaks are observed over a bounded period of time.

6.2.2 Implementation of the Obfuscation Techniques

Obfuscation techniques are implemented on both the analog and digital blocks of the circuit. The 7-bit flash ADC is the only unobfuscated block of the IC. The parameter obfuscating technique presented in [191] is applied to the analog circuit with a 10-bit key. The digital circuit is obfuscated with a 6-bit key using a combination of XOR-based logic encryption [109] and stripped functionality logic locking (SFL) [115]. The total key length applied to the circuit is 16-bits.

6.2.2.a Analog Circuit Obfuscation

The gain of the common source amplifier is masked using the vector-based parameter obfuscation technique described in [191]. Parameter obfuscation is a key based technique that targets the physical dimensions of the transistors used to set the optimal biasing conditions of the circuit. The width of a transistor is obfuscated and, based on an applied key sequence, provides a range of potential biasing points. Only when the correct key sequence is applied and certain transistor(s) are active, are the correct biasing conditions at the target node set.

The effective transistor $\frac{W}{L}$ ratio of the diode connected load is masked by using ten diode-connected load transistors each with different sized transistor widths connected in parallel. Each of the ten transistors is either activated or remains deactivated based on a digital key applied through a decryption block implemented using pass transistor logic.

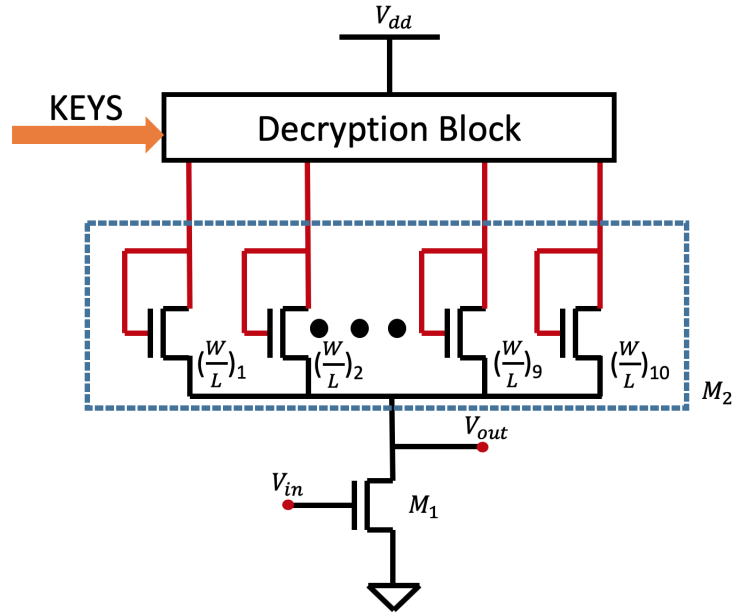


Figure 6.5: Schematic of the obfuscated common source amplifier using vector-based parameter obfuscation.

From (6.2), the gain of the CS amplifier is inversely proportional to the square-root of the $\frac{W}{L}$ ratio of transistor M_2 . The desired transistor width is only set when the correct key is applied, which produces the target amplifier gain A_v [191].

6.2.2.b Digital Circuit Obfuscation

The digital circuitry is obfuscated with two separate techniques. The first is an implementation of XOR based logic locking [109] within the counter of the digital block. The selection strategy for the XOR based locking technique was random and two XOR gates were inserted within the counter circuit. The second technique is applied to the logic cone within the peak detection circuit, where the circuit is modified for a single minterm input.

The circuit implemented in [115] that produces a zero Hamming distance from the inputs of the digital block and the applied key is utilized to correct the flipped minterm.

The circuit characterizing the Hamming distance compares four key inputs and four inputs from the peak detection circuit. The four monitored inputs include B1 and B2 from the output of the ADC and the two outputs from the registers of the state machine shown in Fig. 6.4. The added security of the circuit is, therefore, bounded by 2^4 distinguishing input patterns (DIPs), where the number of DIPs is correlated to the size of the circuit being secured.

6.2.3 Attacking AMS Blocks in Isolation

The system level schematic shown in Fig. 6.1 includes a 10-bit key applied to the CS amplifier and a 6-bit key applied to the digital peak detection and counter circuit. When protecting both circuit blocks, the objective is to include obfuscated circuitry that adds to the overall security of the IC, which is bounded by a key space of 2^{16} combinations. However, if not properly implemented, an adversary is able to isolate the analog and digital circuit blocks, which allows for the independent determination of both the key for the CS amplifier and the key for the digital peak detection and counter circuit. Two techniques to isolate the analog and digital blocks are described: 1) using saturating conditions of the analog circuit block (Section 6.2.3.a) and 2) partially altering the keys of an activated IC (6.2.3.b).

6.2.3.a Saturation Conditions of the Analog Circuit

The CS amplifier of the AMS circuit produces two responses that are known independent of the key. The first response is a result of applying a 0 V input to the CS amplifier. Independent of the applied gain set by the key, the output of the amplifier remains 0.45 V (the DC offset). A logic 0 is produced at the input of the digital circuit for both B1 and B2, which results in the generation of SAT based constraints that more efficiently attack the key space. The second response is due to the saturation of the CS amplifier, which results in an output of 1.8 V, the set power supply voltage (VDD) for the 180 nm technology node used for SPICE simulation. Therefore, applying VDD to the input of the amplifier produces a known output of VDD. DIPs are, therefore, generated for the digital circuit when an all one input is applied to the peak detection circuit. Applying the known logic zero and logic one constraints results in the SAT attack determining the key of the digital circuit in four iterations. Once the digital key is determined, the analog key is now vulnerable to an isolation attack, essentially reducing the key space to 2^{10} combinations from the maximum possible combinations of 2^{16} .

6.2.3.b Partial Activated IC by Applying Semi-Correct Key

As part of the threat model described in Section 6.1, an adversary is assumed to possess access to a second activated IC. The second activated IC allows for partial key modifications, whether applied solely to the digital or analog portions of the circuit. For example,

consider an adversary attempting to attack the key of the analog block in isolation. The adversary must determine a set of key values and inputs to the peak detection and counter circuit that exposes the value of the ADC bits (sensitizes the ADC bits). The miter circuit of the AMS circuit shown in Fig. 6.1 is utilized to determine an input pattern that is dependent on the value of B1 at the output of the digital block. The key inputs and scan chain inputs of the digital circuit are modified and applied to the second activated IC, which allows for the correct determination of the B1 bit based on the input voltage applied to the CS amplifier. The adversary is, therefore, able to set a voltage input to the amplifier that results in the switching of the output of the digital circuit. Inputting a voltage (0 V and VDD) as a constraint to the circuit eliminates all but the correct key, which allows the adversary to decrypt the analog block and reduce the key search space from a maximum of 2^{16} combinations to 2^6 .

The opposite condition also applies, where the adversary simply determines the output of the ADC that isolates the digital portion of the AMS circuit. The adversary applies a SMT solver to determine input and key conditions to the analog block that result in the desired ADC outputs, at which point the activated response of the digital system is obtained. The conventional SAT attack is then executed on the digital block of the circuit. For each determined DIP of the digital block by the SAT solver, the circuit input and key inputs to the analog block are determined such that the target B1 and B2 values are generated. Executing the described attack on the given AMS circuit requires only three

iterations of the SAT attack to determine the correct digital key.

6.3 Adding Key Dependencies between Analog and Digital Circuit Blocks of an AMS Circuit

An attack on a secured AMS circuit that isolates the digital and analog blocks by applying saturating inputs and/or partial key modifications on an activated IC is demonstrated in Section 6.2. The generation of key dependencies between the AMS circuit blocks is described in this section, which results in an increase in the difficulty of determining the key of an AMS circuit when isolating the digital and analog blocks.

To correlate the key response of the analog and digital circuits, an XOR gate with two key inputs, one from the analog circuit and the other from the digital block, is used. The system model shown in Fig. 6.1 is now altered to that shown in Fig. 6.6, with the added correlation between the keys of the analog and digital blocks shown in red dashed lines (interconnects between the analog obfuscation key and digital obfuscation key in Fig. 6.6).

To implement the interdependence between the analog and digital blocks, four random pairs of analog and digital key bits are chosen and XORed together. By including the correlated keys, the circuit is no longer vulnerable to the attack described in Section 6.2 that exploits saturating zero and one inputs to trim the key space. Instead, eight valid

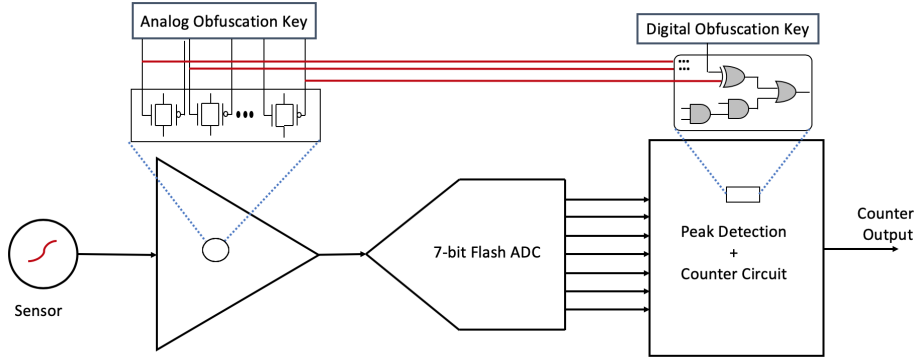


Figure 6.6: AMS circuit topology with dependencies between the analog and the digital blocks. The key for the digital circuit block is now dependent on the key set for the analog circuit.

digital keys remain after the application of the saturating conditions. Essentially, the SAT attack is able to deduce that the analog and digital keys are either equal or inverses of one another, but is not able to determine the value of either key.

Execution of the partial key attack described in Section 6.2.3.b no longer results in the determination of the key of the digital block as the output is also dependent on the analog key bits. As an example, consider applying the SAT solver to generate a condition that outputs a value of *1110000* by the ADC. Even though the output of the ADC is known, the generated analog key is also provided to the digital block, which results in a functionally incorrect output from the IC. The adversary is, therefore, forced to concurrently consider the analog and digital blocks of the IC to determine the key.

To attack the linked blocks, the saturating conditions are first applied. The key space is constrained to eight possible digital key values, which reduces the key space from 2^{16} to 2^{13} combinations. A SMT based attack is then applied to the analog portion of the AMS

circuit by generating a miter circuit that is inputted to a satisfiability modulo theory (SMT) solver. The SMT solver allows for the mathematical expression of the gain equation of the CS amplifier. Simply applying a miter circuit with analog parameters includes challenges as a floating point value differing by a single least significant bit (LSB) generates an undesired but valid miter circuit constraint. To eliminate a larger set of keys per generated DIP, a range of applied voltages is applied to the activated IC. As an example, consider a DIP generated with an input voltage of approximately 0.0378 V and the internal state registers in the digital block set to 01110. The saturating conditions of the circuit are first checked for the given internal register state. The circuit switches one of the output register values of the counter for the operating range of the amplifier (less than 0.04 V). Since the output switches based on the applied input to the CS amplifier alone, an adversary knows that the obfuscated gain of the amplifier is vulnerable to attack. To exploit the condition that an applied voltage within the operating input voltage range of the amplifier switches B1 or B2 from a logic 0 to a 1, the initial input voltage of the DIP (0.0378 V in this example) is varied by $\pm 30\%$ to determine if the logical output of B1 and B2 changes. The $\pm 30\%$ range on either side of the applied input voltage to the ADC generated by the DIP can be constrained more aggressively. A larger range provides a greater probability to observe the change at the output of the circuit, but results in a less constrained key space. A smaller range more effectively constrains the key space, but results in greater difficulty in determining the toggle voltage of B1 and B2. For the implemented peak-detection circuit, three

DIPs with a 30% search margin are sufficient to constrain the AMS circuit to return the correct key. The attack on the entire AMS circuit required 12 iterations of the SAT solver, which was substantially greater than the three total iterations required when attacking the analog block in isolation or the four total iterations when independently attacking the digital block.

As AMS circuits require protecting digital and analog IP, an assurance that the added security is not vulnerable to isolation attacks is needed. The presented method that links the keys of the analog and digital sub-blocks resulted in a significant enhancement of the security of the implemented circuit.

6.4 Securing AMS Systems

The analysis of the security of the AMS peak detection circuit indicates multiple considerations to account for when concurrently securing analog and digital circuit blocks. The following design criteria is provided as a guide to properly secure AMS circuits:

1. The analog and digital circuit blocks must be evaluated for input-output combinations that are independent of the applied key. For the AMS circuit in this chapter, an input voltage of 0 V sensitizes and isolates the digital circuit block as shown in Section 6.2.3.a. To avoid such conditions, the generated digital key must account for saturating DIPs, or the AMS circuit must be dependent on the key for all the possible inputs.

2. Independently securing digital and analog blocks must be avoided. Therefore, a technique is proposed to link the analog and digital keys. Such dependencies prevent an adversary from extracting circuit information by partially altering the key of an activated IC, which is discussed in Section 6.2.3.b.

3. Ensure that the scan chain and internal testing points are inaccessible to adversaries.

The AMS pipeline was highly susceptible to attack as every register within the digital block was accessible. The observability of all of the registers permits access to the ADC output, which results in increased isolation of the analog and digital circuit blocks. Obfuscation, or limitation, of the scan chain and testing circuits is needed to prevent an adversary from efficiently determining the key used for logic locking.

6.5 Conclusions

This work proposes a novel obfuscation methodology to protect AMS circuits against IP piracy and theft by implementing logic and performance locking techniques on the digital and analog domains, respectively. Security analysis utilizing a SMT-based attack on an obfuscated peak detection circuit is performed, indicating a vulnerability when independently implementing obfuscating techniques in the analog and digital circuit sub-blocks. Both saturating conditions of the analog block and application of a partial key to an activated IC allow for the determination of the keys for each circuit block in isolation. To

force an adversary to concurrently consider the keys of both circuit blocks, the key inputs between the analog and digital blocks are linked. The interconnection of the keys from the two blocks results in a 3x increase in the number of DIPs required to determine the key of the AMS circuit. Accounting for the security of the entire AMS circuit as opposed to the individual circuit blocks is, therefore, critical when designing an AMS system.

Chapter 7

Security Vulnerabilities of Key-based Analog Obfuscation Techniques

Key based performance locking of analog circuits proposed in [190, 191], and [192] target the physical dimensions of the transistors used to set the optimal biasing conditions of the circuit. The width and length of a transistor are obfuscated and based on an applied key sequence to the transistor a range of potential biasing points are set. Only when the correct key sequence is applied, which activates a subset of the composite transistor(s), are the correct biasing conditions at the target node set. Incorrect keys produce effective widths that lead to higher and lower target performance characteristics, such as for the gain, as shown in Fig. 7.1. The incorrect higher or lower gain results in signal degradation either at the current stage or a future stage of the analog circuit. The two main challenges

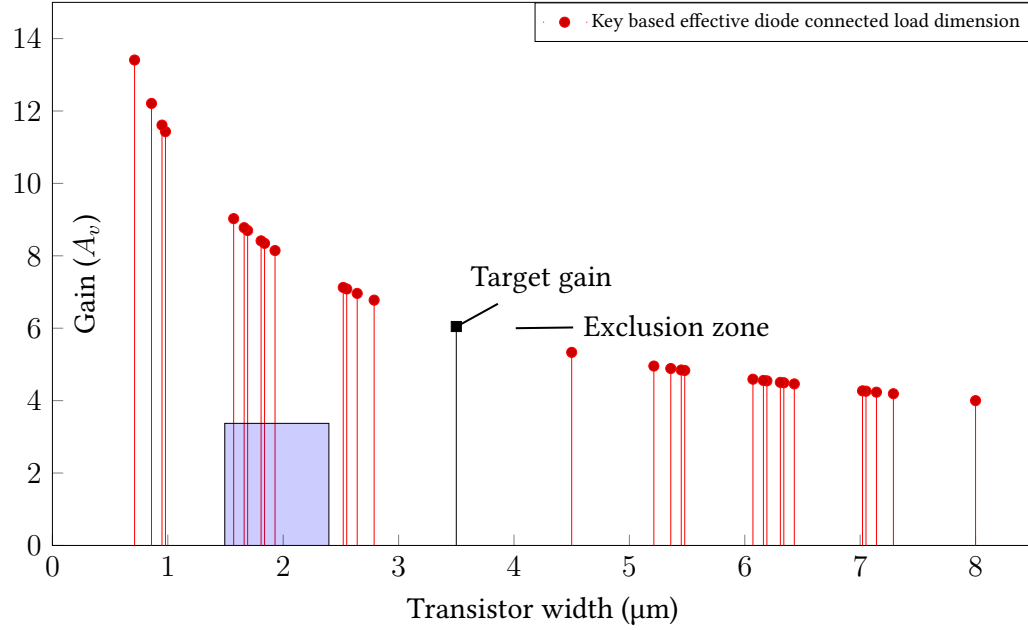


Figure 7.1: Effective transistor widths producing corresponding gains when applying different keys to a CS amplifier obfuscated with a 5-bit key.

of implementing parameter biasing obfuscation are 1) the presence of multiple correct keys and 2) the limited deviation in the performance of a circuit when an incorrect key is applied. To mitigate the challenges, a satisfiability modulo theory (SMT) based design methodology has been developed [190, 192] to ensure only a single key produces the correct circuit performance and that an incorrect key results in significant degradation in the operating conditions of the circuit. The formulated problem and the given constraints are provided as inputs to the SMT solver, which outputs transistor sizes that limit the number of correctly functioning keys. The resulting effective widths for a common source (CS) amplifier are shown in Fig. 7.1.

From the results shown in Fig. 7.1, two security vulnerabilities are evident, specifically 1) there exists an exclusion zone around the target width where no effective transistor widths are present and 2) the performance parameter has a decreasing monotonic response to the increasing effective width. An increasing monotonic response is also possible dependent on the obfuscated parameter. Two attack methodologies leveraging the observations have been developed as described in this chapter.

7.1 Test Circuits

To evaluate the proposed analog attack methodologies, four analog circuit building blocks are implemented in a 180 nm process and analyzed with SPICE simulation. The building blocks include a voltage controlled oscillator (VCO), common source amplifier with diode connected load, single-stage differential amplifier, and an operational amplifier. The analog circuits are obfuscated with 10, 15, and 20-bit keys, and with 1%, 5%, 10%, 15%, and 20% exclusion zones around the target width. The VCO is designed to operate at 3.25 GHz, where each varactor is obfuscated to mask the operating frequency of the VCO. The common source amplifier is designed to produce a gain of 15 dB, with the diode connected load obfuscated to mask the amplifier gain. The single stage differential amplifier is designed to produce a 30 dB gain and a 4 MHz gain-bandwidth product. The operational amplifier is designed to produce a gain of 60 dB and a gain-bandwidth product of 50 MHz. The differential input transistors of both the operational and differential amplifier circuits

are obfuscated to mask the gain and gain-bandwidth.

7.2 Key Spacing

The threat model considered for the key spacing attack is that of an adversary that possesses the circuit netlist through which the obfuscated transistor sizes are targeted. No oracle response is needed for execution of the attack. The methodology of the key spacing attack is based on the principle that the obfuscated transistors require an exclusion zone around the target device dimension. The pseudocode for the key spacing attack is provided as Algorithm 8. The attack utilizes an SMT solver to determine the active width segments of an obfuscated transistor W based on the key $\vec{X}$, where $\vec{X} \in b[0, 1]$. The nearest width W_a greater than W is determined through execution of an optimization algorithm. If W_a is greater than the specified tolerance value Var , then W is added to the list of candidate keys. The constraints of the model representing the circuit provided to the SMT solver are updated to ensure the next W generated is less than W_i or greater than $W_i + var_spacing$, where W_i is the candidate key from the previous SMT iteration and $var_spacing$ is the percentage separation between the given width and the next allowed width (exclusion zone). The previous constraint is valid since any keys within W_i and $W_i + var_spacing$ do not satisfy the key spacing constraints. As $var_spacing$ is increased, the number of invalid keys eliminated per iteration increases. The process is repeated until there are no additional satisfying widths.

Table 7.1: Results of the key spacing attack on a VCO, CS amplifier, differential amplifier, and operational amplifier obfuscated with a 10, 15, and 20-bit key for exclusion zone sizes of 1%, 5%, 10%, 15%, and 20%. The algorithm of the key spacing attack is characterized for the number of candidate keys and the time to determine the candidate solutions.

Circuit Type	Key Size	Exclusion Zone=1%		Exclusion Zone=5%		Exclusion Zone=10%		Exclusion Zone=15%		Exclusion Zone=20%	
		Candidate Keys	Time to Solve (s)	Candidate Keys	Time to Solve (s)	Candidate Keys	Time to Solve (s)	Candidate Keys	Time to Solve (s)	Candidate Keys	Time to Solve (s)
VCO	10	61	4.34	13	1.69	3	0.97	4	0.79	2	0.43
	15	98	12.73	14	10.46	5	6.16	5	0.43	4	2.52
	20	89	487.68	19	71.16	7	68.21	4	71.85	3	72.52
CS Amplifier	10	37	5.62	9	1.35	6	0.96	2	0.68	3	0.47
	15	100	11.39	8	12.11	8	3.21	4	3.80	3	2.86
	20	75	768.38	7	166.14	8	144.47	2	121.89	4	66.23
Differential Amplifier	10	90	1.28	9	1.11	5	1.07	7	0.91	4	0.40
	15	22	92.32	8	15.40	2	7.60	4	2.29	3	1.80
	20	54	671.58	7	367.97	4	101.29	4	64.80	4	48.85
Operational Amplifier	10	67	4.04	11	1.63	5	0.98	4	0.63	2	0.56
	15	35	47.61	21	5.43	5	4.92	5	3.72	4	1.92
	20	36	1520.31	22	88.48	7	100.32	5	117.05	3	89.56

Algorithm 8: Key Spacing Attack

Input: Width Values $\vec{W}_v$,
 Key Spacing Variance Var ;
 $W = \vec{W}_v * \vec{X}$;
 $S_1 = W \wedge (sum(\vec{X}) > 0)$;
 $candidate_keys = []$;
while SAT[S_i] **do**
 $W_a = find_adjacent_width(W_i)$;
 $var_spacing = W_i * Var$;
 if $W_a - W_i \geq var_spacing$ **then**
 $candidate_keys.append(W_i)$;
 $S_{i+1} = S_i \wedge (\vec{W}_v < \vec{W}_{v_i} \vee \vec{W}_v > \vec{W}_{v_i} + var_spacing)$;
end
return candidate_keys;

SMT analysis is performed to evaluate the execution of the key spacing attack on a VCO, operational amplifier, differential amplifier, and common source amplifier, with results listed in Table 7.1. The attack is more efficient when the PVT variation is greater, as the number of eliminated keys per iteration increases. The time to solve also increases

exponentially as the key size increases, which indicates a less feasible attack for circuits with very large key sizes.

7.3 Monotonic Circuit Response Attack

The threat model for the monotonic attack assumes an adversary has access to an oracle IC that is utilized as a black-box to obtain input-output responses. The attack makes use of the monotonic relationship between the given width and the output response of the circuit, which applies to a majority of the analog circuit blocks. For example, the monotonic decrease in the gain of a common sources (CS) amplifier as the transistor width is increased is shown in Fig. 7.1. The monotonic nature of an obfuscated circuit parameter allows an adversary to efficiently partition the keyspace with oracle responses and eliminate any need for modeling the analog circuit with governing equations.

The pseudocode for the monotonic function attack, which prunes the key space, is provided as Algorithm 9. The attack begins by querying an output of the oracle and applying a variation offset to the obtained value to represent any process, voltage, and/or temperature (PVT) variations that alter the output response of the circuit. The SAT constraints are then utilized to select a width, which is applied to the obfuscated transistor. The response from the non-activated IC for the given width is used to analyze the parameter characteristics of the circuit (i.e. gain, bandwidth, and operating frequency). The algorithm checks if the circuit response falls outside of the range of possible values as

Table 7.2: Results from executing a monotonic attack on a VCO, CS amplifier, differential amplifier, and operational amplifier all obfuscated with a 20-bit key. PVT variations of 1%, 5%, 10%, and 20% are analyzed. The algorithm implementing the monotonic attack is characterized for the number of candidate keys, execution time, and the number of iterations of the SMT solver.

Circuit	Exclusion Zone Margin (%)	PVT Variation=1%			PVT Variation=5%			PVT Variation=10%			PVT Variation=20%		
		Candidate Keys	Time to Solve (s)	No. Iter	Candidate Keys	Time to Solve (s)	No. Iter	Candidate Keys	Time to Solve (s)	No. Iter	Candidate Keys	Time to Solve (s)	No. Iter
VCO	1	3	0.57	28	27	1.16	58	71	2.67	101	181	13.84	201
	5	1	0.49	23	5	0.57	28	15	0.83	37	46	2.29	63
	10	1	0.46	19	2	0.44	20	11	0.72	31	35	1.43	55
	15	1	0.42	20	1	0.43	20	7	0.47	22	31	1.56	48
	20	1	0.30	13	1	0.32	13	3	0.39	15	20	0.79	37
CS Amp	1	1	0.45	23	5	0.50	26	9	0.58	31	20	0.73	38
	5	1	0.46	24	1	0.46	24	5	0.48	25	13	0.66	35
	10	1	0.50	25	1	0.51	25	4	0.55	29	24	0.93	47
	15	1	0.49	22	1	0.43	22	2	0.45	23	13	0.70	34
	20	1	0.53	24	1	0.53	24	1	0.53	24	6	0.52	26
Diff Amp	1	4	0.55	16	16	0.80	36	31	0.99	48	71	2.09	98
	5	1	0.45	21	6	0.63	30	46	1.53	67	577	24.76	604
	10	1	0.54	24	8	0.67	33	47	1.44	67	882	50.42	904
	15	1	0.36	15	1	0.33	15	6	0.55	25	19	0.81	40
	20	1	0.32	14	1	0.30	14	5	0.49	5	16	0.73	33
Op Amp	1	5	0.77	34	40	1.33	65	132	3.98	157	900	59.59	926
	5	1	0.52	21	7	0.51	24	18	0.78	36	46	1.56	60
	10	1	0.32	14	3	0.37	17	13	0.82	40	38	1.31	64
	15	1	0.37	18	1	0.40	18	9	0.54	25	27	1.03	45
	20	1	0.45	18	1	0.39	18	3	0.43	20	29	1.01	45

compared to the oracle response, and if so, constrains the selection of the width during the next iteration of the solver. If the returned response falls within the accepted range of circuit properties, then the given key is added to the list of candidate keys. The process continues until no further widths are generated. The algorithm concludes by returning the list of candidate keys.

The results of executing the monotonic attack on a VCO, operational amplifier, common source amplifier, and a differential amplifier are listed in Table 7.2, which indicate a very effective attack as the key space is cut in half each iteration. For 10% PVT variation, the maximum number of candidate keys returned is 132, which allows an adversary to perform a brute force attack to determine the correct key. The adversary is also able to

Algorithm 9: Monotonic Attack

Input: Width Values $\vec{W}_v$,
 Key Spacing Variance Var ;
 $W = \vec{W}_v * \vec{X}$;
 $S_1 = W \wedge (sum(\vec{X}) > 0)$;
 $candidate_keys = []$;
 $oracle_response = query_oracle()$;
 $oracle_min = oracle_response * (1 - Var)$;
 $oracle_max = oracle_response * (1 + Var)$;
while $SAT[S_i]$ **do**
 $ckt_response = get_circuit_response(W_i)$;
 if $ckt_response > oracle_max$ **then**
 $S_{i+1} = S_i \wedge (W < W_i)$;
 else if $ckt_response < oracle_min$ **then**
 $S_{i+1} = S_i \wedge (W > W_i)$;
 else
 $candidate_keys.append(W_i)$;
 $S_{i+1} = S_i \wedge (W \neq W_i)$;
end
return $candidate_keys$;

evaluate the oracle at additional operating conditions and re-execute the attack to further reduce the keyspace. Most of the attacks conclude on the order of a few seconds, with the longest attack requiring approximately one minute to complete, demonstrating a significant vulnerability. In addition, as opposed to the key spacing attack, the attack runtime does not increase significantly with key size since adding additional key bits only linearly increases the number of iterations required by the algorithm to execute.

7.4 Design Insight

Based on the vulnerabilities of analog circuits to the key spacing and monotonic attacks, two design considerations are presented to strengthen the security of the existing parameter obfuscation techniques: 1) equal key spacing and 2) obfuscating multiple nodes to create key dependencies that result in a non-monotonic output response. The presented techniques are compatible with existing SMT based methods used to determine the sizes of transistors to obfuscate [190].

7.4.1 Equal Key Spacing

The efficiency of the key spacing attack is dependent on keys existing within the bounds of the margins of the widths of a given key when accounting for PVT variations. For example, if the evaluated width is 5 μm and there are ten widths within a 5% margin, then the attack does not have to consider those 10 widths in subsequent iterations of the execution of the algorithm. However, if the keys produce widths of equal spacing, then the attack requires the same number of iterations as a brute force attack. The width spacing generated by the keys must be considered when obfuscating an analog circuit. The trade-off of using keys that produce equally spaced widths is the additional area required as the widths of the obfuscated transistors are larger. A reduction in area is possible by limiting

the widths to generate a desired number of candidate solutions, which results in a trade-off between the efficiency of the attack and the overhead in area when implementing a target security threshold.

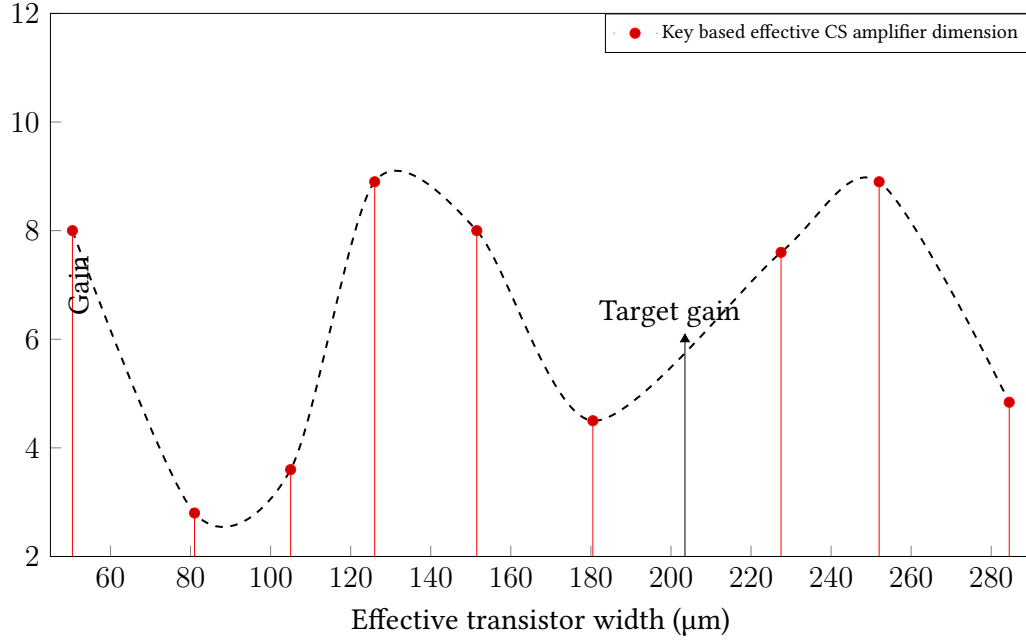


Figure 7.2: Effective transistor widths of a CS amplifier with corresponding gain when obfuscated using non-monotonic key dependencies. Both transistors of a CS amplifier are obfuscated resulting in a non-monotonic gain response.

7.4.2 Creating Non-Monotonic Key Dependencies

The efficiency of the monotonic attack is due to the monotonic relationship of the obfuscated width and the circuit response. If the keys produce a non-monotonic dependency in the functional output response of the circuit, the attack is no longer guaranteed to produce the correct key. To generate non-monotonic dependencies in the function of

the circuit, multiple circuit parameters must be concurrently obfuscated. For example, in a CS amplifier, obfuscating both transistors results in a non-monotonic gain as a function of width as shown in Fig. 7.2. Since the gain is no longer monotonic, the adversary is unable to significantly prune the keyspace when the oracle response does not match the obfuscated circuit response.

7.5 Conclusions

Security vulnerabilities based on both the spacing of key values and the monotonic response of the circuit as keys are applied are demonstrated in this chapter for key-based analog obfuscation techniques. The key spacing attack utilizes the presence of an exclusion zone around the target transistor size to determine candidate keys without the need of an oracle IC. The monotonic attack utilizes the response of an oracle IC to efficiently partition the keyspace without requiring the adversary to develop circuit equations of the target parameters (gain, bandwidth, frequency) of the obfuscated circuit. The developed attacks were evaluated on a VCO, common source amplifier, differential amplifier, and an operational amplifier for 10, 15, and 20-bit key sizes and for 1%, 5%, 10%, 15%, and 20% exclusion zones around the target width. The results indicate that an average of 5 candidate keys were returned in an average time of 5.47 seconds for the key spacing attack. For the monotonic circuit response attack, the average execution time was 1.18 seconds when

assuming 10% PVT variation. Methodologies to thwart the proposed attacks are developed, including equal key spacing and non-monotonic key dependencies, which provide increased security when implementing analog obfuscation.

Chapter 8

Performance Analysis of Analog De-Obfuscation Attacks

Concurrent to the development of key spacing attack [200] and the monotonic attack [200] discussed in Chapter 7, attack algorithms, including satisfiability modulo theory (SMT)-based [201] and optimization-based [202], have been proposed to evaluate the efficacy of techniques that obfuscate analog circuits. Some attack methodologies assume an oracle assisted model similar to that of the satisfiability (SAT) based attacks in the digital domain [203–205]. However, unlike the digital circuit domain, where the SAT attack is the de-facto standard for measuring the resiliency of key-based logic locking techniques, no such standard exists in the analog circuit domain, where developed attack algorithms address differing threat models.

Current analog attack algorithms assume an ideal threat model in which an attacker possesses all circuit information except for the correct key. However, when considering an attack executed in the field, an adversary typically does not possess all circuit information. Unlike for digital circuits, where only the circuit netlist and a live IC are needed for execution of the SAT attack to determine the key obfuscating the combinational logic [203–205], attack algorithms require additional information to effectively de-obfuscate a locked analog circuit. Analog attack algorithms typically assume that all necessary information is available to the adversary without considering the cost and effort to obtain needed information. As an example, for the equation-based SMT attack presented in [201], additional information needed for execution of the attack includes the biasing conditions of the circuit and process design kit (PDK) specific parameters such as the mobility of electrons/holes (μ), the gate oxide capacitance (C_{ox}), the thickness of the gate oxide (t_{ox}), and the threshold voltage (V_{th}). The loss of one or more sources of information adversely affects the performance of the analog attack algorithms, where the equation-based SMT attack [201] results in 30x more candidate keys on average and requires 17x more time to execute when the attacker does not possess the required PDK information as compared to the case presented in [201], where the attacker is assumed to possess all required information. Similar trends are observed for other analog attack algorithms including the GA attack [202] and the monotonic response attack [200].

The primary contributions of the chapter include:

1. Addressing the hidden cost, number of steps, and resources needed to successfully attack an analog circuit. Unlike the threat models for obfuscated digital circuits that require only the circuit netlist and a live IC to execute a SAT attack, analog attack techniques also require information on the biasing conditions of the circuit and technology specific parameters found in the process design kits (PDK). The cost, challenges, and resources to obtain the required information is overlooked when considering current threat models, but is considered in this dissertation.
2. The current state-of-the-art attack techniques assume that the adversary possesses all required circuit information. However, an attack executed in the field is far from ideal, where the attacker lacks one or more sources of required information. In this dissertation, four separate threat models are assumed that more closely mimic real world attack scenarios, with one or more sources of needed information made progressively unavailable to the attacker.
3. Current analog attack techniques are not evaluated on multi-stage and multi-node obfuscated analog circuits. This dissertation evaluates current analog attack techniques on five distinct analog circuit topologies that include a varying number of stages, obfuscated nodes, and key sizes.

8.1 Assumptions in Threat Models For Attack Techniques

To quantify the robustness of an attack algorithm, an adversary is assumed to possess a certain level of information pertaining to the obfuscated circuit under attack. However, obtaining this information when assuming practical considerations is challenging, time consuming, and expensive. In this section, an analysis of the difficulty in obtaining the assumed information possessed by the attacker is provided.

8.1.1 Netlist Extraction

For the threat model of an untrusted foundry, netlist extraction is a straight-forward procedure of converting a GDSII file to a netlist. But if the attacker is an untrusted end user, the process of extracting the circuit netlist is costly, expensive, and requires dedicated facilities [206]. To completely reverse engineer an IC, high quality images of both active layers and metal layers are needed. The reverse engineering process typically consists of five primary steps: depackaging, delayering, imaging, annotation, and extraction [206]. Depackaging requires either use of corrosive chemicals or dedicated mechanical shock equipment. In the delayering process, each of the metal layers is removed and scanned until the substrate is reached. Since each layer varies in thickness and type, the delayering methodology for each layer differs. In addition, the delayering process requires the use of dedicated facilities with strict climate control and the use of corrosive chemicals,

which adds time and expense. The delayered IC is imaged using a scanning electron microscope (SEM) to extract a precise representation of each metal and active layer. The cost of SEM imaging increases with advanced technology nodes [207]. Once the IC is delayered and imaged, the transistors and metal layers are identified and aligned to determine the connections between devices. The process of annotation requires dedicated reverse engineering software tools that align and help visualize all layers of the IC. Depending on the accuracy of the annotation, the netlist is automatically generated.

The most advanced imaging tools utilize ptychographic X-ray laminography, which requires around 30 hours to scan 0.3 mm^2 of a side and with 500 nm resolution [208]. In addition, to scan a $40 \text{ }\mu\text{m}^2$ area of a circuit, the laminography process required approximately 60 hours to complete when resolving 19 nm features. State-of-the art laminography techniques are only able to reverse engineer interconnects and are not mature enough to reverse engineer transistors [208]. Extrapolating the data provided in [208], reverse engineering the interconnect of a 3 mm^2 analog receiver using X-ray laminography requires about 12.5 days for 500 nm resolution and 187.5 days for 19 nm resolution. A typical cost of reverse engineering an IC at the block level ranges from \$10,000 to \$50,000, while the extraction of the full circuit netlist is estimated to cost upwards of a quarter million dollars [209]. In addition, the tools utilized for annotation are typically capable of identifying digital gates and interconnects implemented with standard place and route tools. However, additional steps are required to identify custom designed circuits and routing

typically seen in analog and mixed signal ICs. Therefore, the described steps highlight the complexity, cost, resources, and time needed to completely extract the netlist from a packaged analog IC.

8.1.2 Extraction of PDK information

When considering an untrusted foundry, the PDK information is assumed readily available to the adversary. However, when considering an untrusted end user, the attacker must gain access to the PDK to extract technology specific parameters including the charge mobility (μ), gate oxide thickness (t_{ox}), and threshold voltage (V_{th}). For the equation-based SMT attack [201], the SMT solver prunes the range of obfuscated transistor sizes to determine the candidate solutions. For the given range of obfuscated transistor sizes, the technology specific parameters do not remain constant, as shown in Fig. 8.1, where β , given by $\mu C_{ox} \frac{W}{L}$, varies by as much as 1500x and 4000x when considering the entire range of obfuscated transistor dimensions of an op-amp and VGA, respectively. Therefore, assuming typical values for technology specific parameters does not guarantee the return of the correct key in the list of candidate keys. In addition, allowing for a range of values for technology parameters increases the likelihood of returning the correct key at the cost of increased complexity of the SMT attack, increased execution time, and an increase in the number of returned keys that the attacker must then brute force.

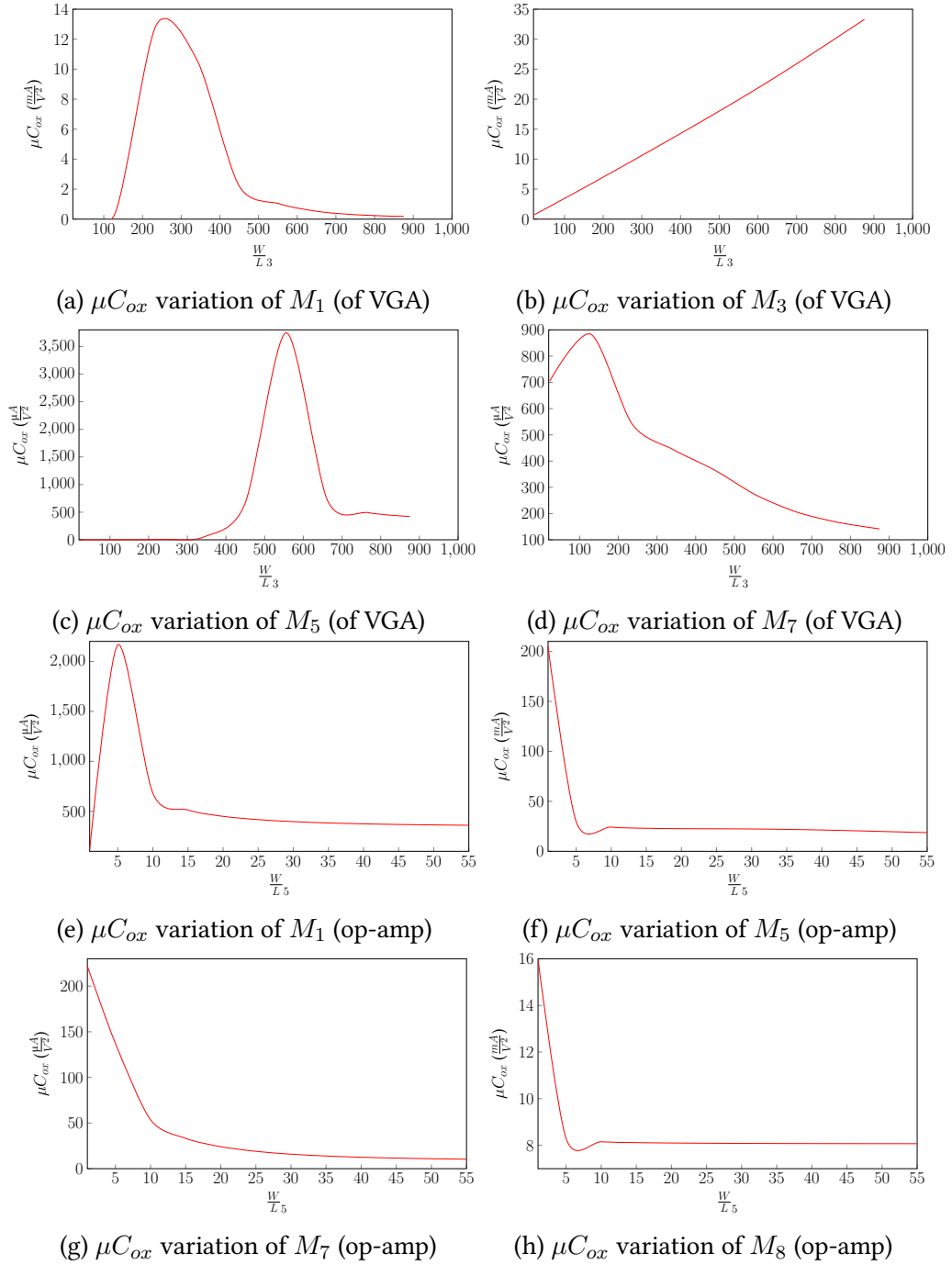


Figure 8.1: Analysis of the variation in μC_{ox} of the critical transistors of (a) - (d) the single-stage VGA shown in Fig. 8.2a and (e) - (h) the op-amp shown in Fig. 8.3 when obfuscating the transistor dimensions $\left(\frac{W}{L} \right)$ of, respectively, M_3 and M_5 , which are utilized in the equation-based SMT attack [201].

8.1.3 Extraction of bias voltages and currents

Irrespective of the threat model, determining the bias voltages and/or currents is challenging as neither the GDSII nor the datasheets include the biasing information of the internal node(s). When considering an untrusted foundry, the attacker is only able to estimate the range of the bias voltages and currents by analyzing the biasing circuitry. The bias voltage for most analog circuits is typically between the threshold voltage (V_{th}) and the power supply voltage (V_{DD}). For an untrusted end user, the biasing voltages and/or currents are determined in one of two ways: 1) by netlist extraction through reverse engineering and SPICE simulation, or 2) a probing attack on the depackaged IC. The extraction of the netlist through reverse engineering is a complex, expensive, time consuming, and destructive process, which results in the loss of the bias voltages and/or currents of the circuit. Therefore, similar to an untrusted foundry, the attacker must determine the ranges of the bias voltages and currents by analyzing the biasing circuitry.

A probing attack is an invasive method used to characterize the implementation and the functionality of an IC by depackaging and electrically probing internal nodes [210]. The steps involved in a probing attack include: 1) decapsulation, 2) reverse engineering, and 3) wire location and probing. The attacker must reverse engineer the entire IC to analyze and determine the locations to probe. Since the original IC used for reverse engineering is no longer functional, the probing attack is performed on a separate IC. For

the second IC, the attacker must once again perform decapsulation, where the IC packaging is removed to expose the die. An attacker performing the decapsulation process requires adequate practice and expertise in handling corrosive and harmful chemicals, while ensuring the die is not damaged during the process [210]. Once the IC is prepared and nodes for probing identified, the next step is to determine the absolute coordinates of the point to mill, which requires a precise kinematic mount and fiducial markers to determine. State-of-the-art focused ion beam (FIB) is then utilized to mill the hole and expose the target wiring [210]. Once a target wire is exposed, the probes must be placed in contact with the target interconnect without triggering any alarm signals from, if implemented, active or passive shielding techniques [210]. With contact to the target wire, the biasing voltage and/or current is extracted using a probe station, which potentially leads to an erroneous bias point due to the impedance of the probes. The described steps highlight the complexity, cost, and resources needed to extract the bias voltages and/or currents of an analog circuit.

8.1.4 Generating Circuit Models

In this dissertation, to reduce the time needed to determine the key of a locked analog IC, the attacks are performed on a model of a locked analog circuit. Performing an attack on a physical circuit typically involves preparing the IC sample for attack, inputting a key, extracting voltage/current information through probing, determining the circuit response,

and inputting the circuit response into an attack algorithm, which is mostly manual and time intensive. Therefore, circuit models are essential to efficiently attack a locked analog IC, where the accuracy of a circuit model directly affects the performance of an attack algorithm. For example, when executing a SAT-based attack on a single stage variable gain amplifier, the correct key is determined when the tolerance, the difference between the parameter specifications obtained from the model of the analog circuit and the specifications obtained from the oracle IC, is more than 50%. The results clearly indicate differences between the performance of the circuit models and the circuit simulator (*Cadence* in this dissertation). In addition, for the SAT-based attack, a higher tolerance results in a large number of candidate keys returned that the attacker must then brute-force, which reduces the efficiency of the attack. The SMT-based attack executed on a variable gain amplifier (VGA) secured with a 10-bit key resulted in 1009 candidate keys, with only 15 keys eliminated from the search space. Possible models of the circuit available for execution of the attack include the SPICE netlist or circuit equations of the performance parameters, which result in similar or even a larger differences in the expected performances as compared to the actual IC. The result is a larger number of returned candidate keys and an increased execution time of the attack algorithm.

Generating a SPICE netlist of an analog IC is part of the reverse engineering process. But the accuracy of the extracted SPICE netlist is dependent on the accuracy of the utilized imaging, annotation, and extraction tools as discussed in Section 8.1.1. The generation of

circuit equations utilized by the SMT attack requires reverse engineering the locked analog IC, identifying the different sub-blocks of the IC (amplifier, mixer, phase-locked loops, filters, etc.), determining the specific topology of a given sub-block (i.e. for an amplifier, determining whether the amplifier is single-ended, differential, an operational amplifier, an instrumentation amplifier, or a cascode amplifier), determining the equations that best model each of the sub-blocks, and correctly combining the equations of all the sub-blocks utilized by the analog attack algorithm. The efficacy of the circuit equation-based models, therefore, depends on the accuracy of the reverse engineering process and the equations that model each of the sub-blocks. As technology scales, the accuracy of the equations describing the behavior of the circuit diverges from the simulated and experimental behavior of the circuit, which, therefore, results in less accurate circuit models for attack [211]. In addition, the technology specific information of a particular process node and the biasing conditions of the circuit also affect the accuracy of the developed models. To compensate for the variation in the performance parameters resulting from an inaccurate circuit models, as compared to the performance of an active analog IC, the attack algorithms allow for a range of acceptable performance values.

8.2 Circuit Implementations

The four analog circuits that are secured and then used for evaluation of the analog obfuscation attacks are discussed in this section.

8.2.1 Variable Gain Amplifier

The circuit schematic of a single stage variable gain amplifier (VGA) is shown in Fig. 8.2a. Two topologies of a VGA are secured utilizing key-based parameter obfuscation, one consisting of a single-stage and the other consisting of two stages. Transistor M_3 of the single stage VGA is obfuscated with a 10-bit key to mask both the target output gain of 8.3 dB and target bandwidth of 100 MHz. The circuit schematic of the two stage VGA is shown in Fig. 8.2b, where two single stage VGAs are serially connected. A 12-bit key is utilized to obfuscate the total combined gain of the two stage VGA, which is set to 14.34 dB for a bandwidth of 100 MHz. A 6-bit key is used to obfuscate the M_3 transistor of each stage of the two stage VGA.

8.2.2 Operational Amplifier

The circuit schematic of a two stage op-amp is shown in Fig. 8.3. The target specifications of the op-amp that are obfuscated include a gain of 60 dB, a gain-bandwidth (GBW) of 10 MHz, and a slew rate of 10 V/ μ s. Both the first and second stage are secured by obfuscating, respectively, the sizes of transistors M_5 and M_8 each with a 6-bit key (12-bits total).

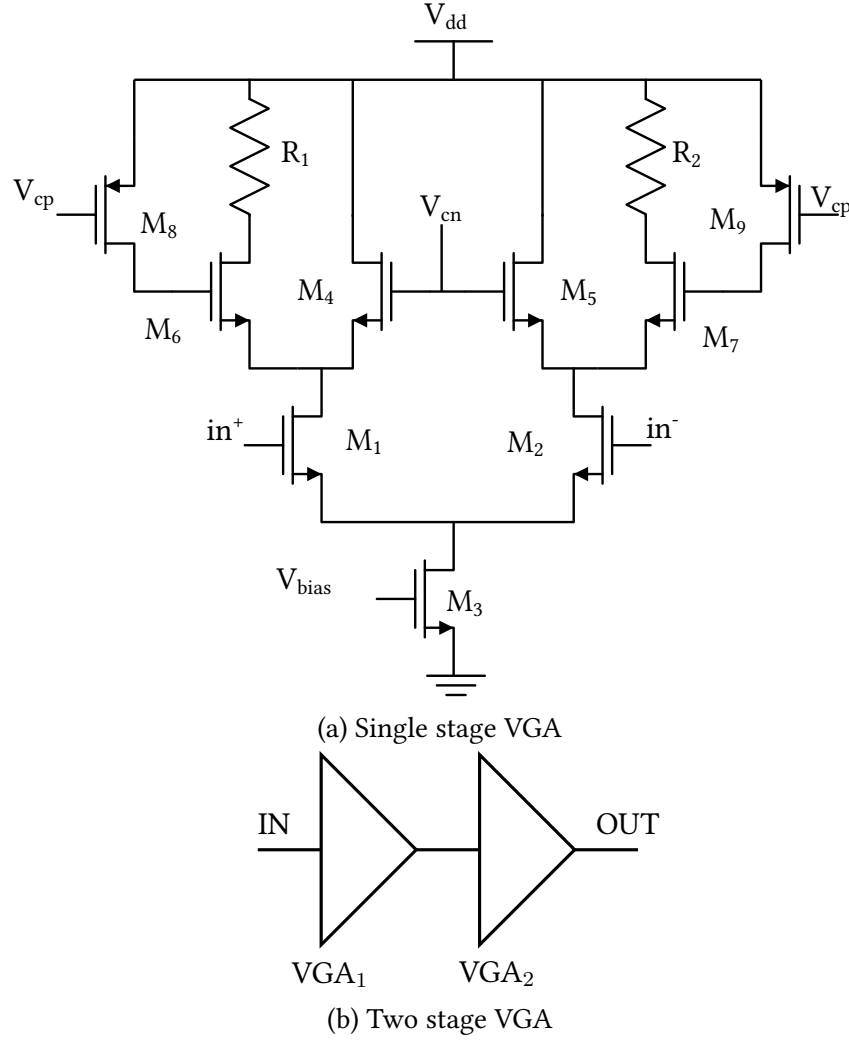


Figure 8.2: Circuit schematic of a (a) single stage VGA [212] and (b) two-stage VGA. Each circuit is secured with key-based parameter obfuscation, where the single stage VGA is obfuscated by a 10-bit key and the two stage VGA is obfuscated by a 12-bit key.

8.2.3 Mixer

The circuit schematic of a mixer is shown in Fig. 8.4. The mixer is designed to produce a 100 MHz output frequency V_{IF} for an input RF frequency V_{RF} of 2.4 GHz and a local oscillator frequency V_{LO} of 2.3 GHz. The conversion gain (CG) of the mixer is set to 3.25

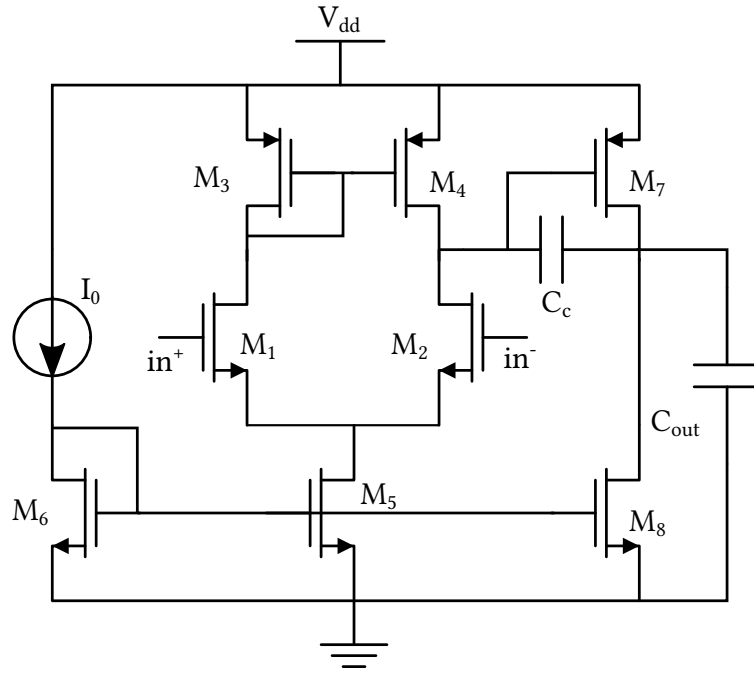


Figure 8.3: Circuit schematic of a two stage operational amplifier [213].

dB. The circuit is secured by obfuscating the dimensions of transistor M_7 with a 10-bit key, which masks the conversion gain of the mixer.

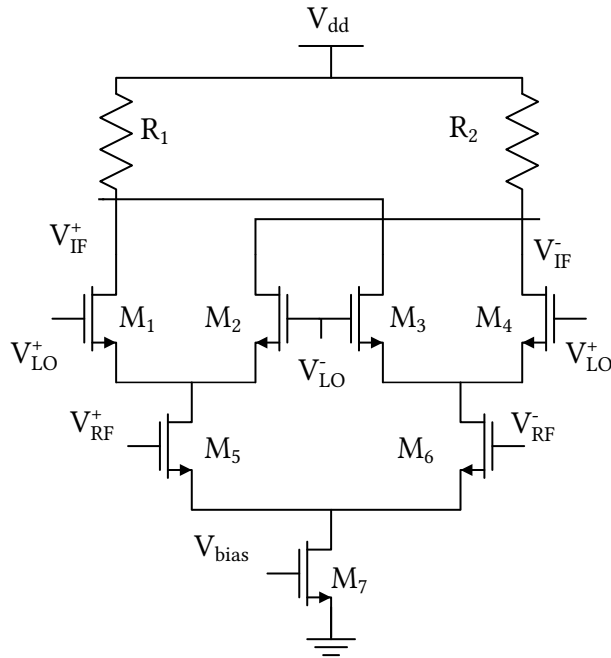


Figure 8.4: Circuit schematic of a Gilbert mixer [214].

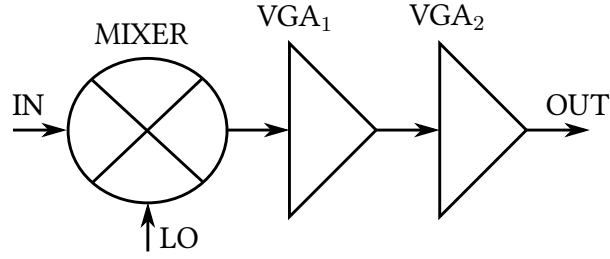


Figure 8.5: Block diagram of the front-end circuit consisting of a mixer and a two-stage VGA.

8.2.4 Front-End

An analog front-end circuit consisting of a mixer and a two stage VGA is designed as shown by the block level diagram in Fig. 8.5. The VGAs and the mixer are designed with the same specifications as described in Sections 8.2.1 and 8.2.3, respectively. Each sub-block of the front-end circuit is obfuscated with a 6-bit key resulting in an overall key-size of 18-bits. The transistor M_7 of the mixer shown in Fig. 8.4 is obfuscated to mask the conversion gain. Transistor M_3 of each VGA shown in Fig. 8.2a is obfuscated to mask the overall gain A_v and the center frequency F_c of the front-end. Only when the correct key is applied to each sub-block does the receiver operate at the desired frequency and properly amplify the intermediate frequency (IF) signal, with incorrect keys resulting in the attenuation of the IF signal. Therefore, obfuscating a transistor in each of the three sub-blocks of the front-end masks the overall target gain and the operating frequency of the multi-stage circuit.

8.3 Security Evaluation of Analog Attack Algorithms

In this section, four analog attack scenarios, based on the level of circuit information available to the adversary, are described.

8.3.0.a Scenario I: Attacker possesses all information except the correct key

Most analog attack algorithms are currently evaluated only for Scenario I, where the attacker possesses the obfuscated circuit netlist, all bias voltages and/or currents of the circuit, the PDK documentation, all circuit specifications, and an oracle IC. Utilizing the given information, the attacker efficiently models the obfuscated circuit and executes the attack. Attacks executed considering Scenario I typically ignore the cost and time of reverse engineering the netlist through delayering and imaging, obtaining the bias information through probing, obtaining the appropriate PDK information from the foundry, and obtaining and properly characterizing the oracle IC, which makes the attack both the most expensive and the most time consuming.

8.3.0.b Scenario II: Attacker does not possess PDK information

In addition to not having the correct key, the attacker is no longer assumed to possess the exact PDK information needed to develop the electrical model of the circuit, which includes the mobility μ , the gate oxide capacitance C_{ox} , the gate oxide thickness T_{ox} of the devices, and the threshold voltage V_{th} of the devices. In addition, each electrical parameter

varies with circuit conditions, which is also assumed unavailable to the attacker. However, based on the implemented circuit and basic technology information, the attacker is able to make an educated guess on the range of possible values each PDK parameter can assume. The circuit models developed under Scenario II are not as accurate and the search space of the attack algorithms is much larger than that of Scenario I, which results in increased iterations of the attack, longer time for execution, and a greater number of returned candidate keys that must be verified.

8.3.0.c Scenario III: Attacker does not possess biasing information

For Scenario III, the attacker is assumed not to possess biasing information and PDK documentation, while attempting to determine the correct key. The cost and time required to obtain the biasing information through probing is no longer considered. However, the circuit netlist and the PDK provide information on the biasing circuitry, through which the attacker determines applicable ranges of biasing voltages and/or currents.

8.3.0.d Scenario IV: Attacker only possesses the Oracle IC

For Scenario IV, the attacker only possesses the oracle IC. The attacker is assumed capable of obtaining I/O responses from the oracle for different applied keys. The attacks executed while considering Scenario IV are both the cheapest and fastest to initialize, but typically require more time to execute and return a larger number of candidate keys as compared with the other scenarios.

Table 8.1: Results of executing the equation-based SMT attack on the five obfuscated analog benchmark circuits described in Section III and for the four different scenarios described in Section IV. The “*” indicates that the SMT constraints are constructed accounting for gain and gain bandwidth parameters.

Scenarios	VGA (10-bits)				Mixer (10-bits)				Op-Amp (12-bits)				2-stage VGA (12-bits)				Front-End (18-bits)			
	Tolerance	Keys	Time	Comment	Tolerance	Keys	Time	Comment	Tolerance	Keys	Time	Comment	Tolerance	Keys	Time	Comment	Tolerance	Keys	Time	Comment
I	10%	294	2.34 s	No correct key	10%	2	0.62 s	No Correct key	10%	138	2.56 s	Correct key	10%	735	14.64 s	No correct key	10%	9735	799.97 s	No correct key
	50%	1009	6.30 s	Correct key	60%	8	0.67 s	Correct key	10%*	0	0.18 s	UNSAT	50%	3969	43.60 s	Correct key	50%	42948	2818.25 s	Correct key
II	10%	295	2.51 s	No correct key	10%	607	3.86 s	Correct key	10%	4089	31.96 s	Correct key	10%	3969	51.07 s	Correct key	10%	250047	13.35 H	Correct key
									10%*	0	0.13 s	UNSAT								
	50%	1009	6.74 s	Correct key					60%*	123	3.59 s	Correct key								
III	10%	511	3.75 s	Correct key	10%	1023	3.87 s	Correct key	10%	4089	42.24 s	Correct key	10%	3969	92.63 s	Correct key	10%	250047	14.67 H	Correct key
									50%*	251	19.59 s	Correct key								
IV	Cannot Perform Attack																			

8.4 Performance of Different Analog Attack Methodologies

In this section the SMT [201], GA [202], key spacing [200] and monotonic based [200] attacks are evaluated on obfuscated circuits for the four attack scenarios described in Section 8.3. The evaluation of different analog attacks is performed on a server that includes two 12-core Intel Xeon E5 CPUs and 96 GB of DDR4 memory.

8.4.1 Equation-based SMT Attack

A satisfiability modulo theory (SMT) based technique to circumvent analog obfuscation is presented in [201]. The technique requires formulating SMT equations based on the extracted netlist and on knowledge of the analog circuit, process design kit (PDK), bias voltages and currents, and circuit specifications. The SMT-based equation is formulated

from two fundamental equations derived from the analog circuit under attack. For the formulation of the first equation, the physical locations or sub-blocks of the circuit that are locked and the type of locking mechanism are determined. The relationship between the obfuscated parameter(s) and the biasing conditions (bias voltages and/or currents) are then formulated through analog circuit knowledge. The second equation is formulated based on the circuit performance of the oracle IC. The relationship between the analog circuit specification(s) and the biasing conditions are based on the topology and the type of analog circuit implemented. Finally, the SMT problem is formulated by equating the two derived equations, where the biasing conditions determined by setting the obfuscated parameter(s) of the analog circuit must equal the biasing conditions of the oracle IC that result in the correct analog circuit performances.

The described threat model [201] assumes that the attacker already has information on: 1) the sizes of all the transistors and passive devices of the obfuscated analog circuit except for the locked component(s) from the extracted netlist, 2) the technology dependent parameters including the mobility of electrons/holes (μ), gate oxide capacitance (C_{ox}), thickness of the gate oxide (t_{ox}), and the threshold voltage (V_{th}), 3) the set biasing voltages and currents, and 4) the circuit specifications from the datasheets. Obtaining all the above mentioned information is costly and time intensive as discussed in Section 8.1.

The equation-based SMT attack is executed on the benchmark analog circuits described in Section 8.2 and for the four attack scenarios presented in Section 8.3. The

results obtained from the analysis of the SMT attack are listed in Table 8.1, where *tolerance* refers to the difference between the parameter specifications obtained from the model of the analog circuit and the specifications obtained from the oracle IC. The SMT algorithm returns all the *candidate keys* that meet the set of circuit specifications within the defined search space. If a key exists within the *candidate keys* that produces the target circuit specifications, then the key is labeled as a *correct key*. For the equation-based SMT attack, the adversary must determine the correct key from the list of candidate keys through brute-force. The results obtained by setting the tolerance to 10%, which is the minimum tolerance used for the analysis, and the lowest tolerance for which the SMT attack returns the correct key, given as a percentage, are listed in Table 8.1. The objective from the perspective of the attacker is to minimize the tolerance, while also ensuring the presence of the correct key within the list of candidate keys. Setting a lower tolerance results in a fewer number of candidate keys and a faster execution-time of the attack algorithm. However, the presence of the correct key is not always guaranteed, as indicated by the results listed in Table 8.1 for a 10% tolerance. In contrast, setting a high tolerance increases the probability of finding the correct key within the list of candidate keys. However, a greater number of candidate keys are also returned, which the attacker must then brute-force to determine the correct key.

The SMT attack performs exceptionally well for Scenario I, where the adversary possesses all circuit information except for the correct key and is, therefore, able to construct

an accurate model of the obfuscated circuit. The SMT attack terminates within seconds and results in only a handful of candidate keys. A brute-force attack is then utilized on the candidate keys to determine the correct key. However, as the information that the attacker possesses is reduced, the search space drastically increases such that the SMT solver now has to search through ranges of PDK parameters and biasing voltages and currents. As the number of considered variables increase, the execution time and the number of returned candidate keys also increase. For the single stage VGA considered under Scenarios II and III and for a mixer considered under Scenario III, less than 1.5% of the total keys are eliminated.

For the op-amp, the SMT algorithm is constrained for two conditions: i) Type I: Only the gain parameter is used to constrain the solver and ii) Type II: Gain, gain-bandwidth, and slew rate are used to constrain the solver. The results of the Type II analysis are denoted with an "*" in the tolerance column of Table 8.1. The results of performing the equation-based SMT attack on the op-amp indicate that increasing the number of circuit specifications to constrain the formulated SMT problem reduces the number of returned candidate keys as compared to constraining the algorithm with a single circuit specification. However, an increase in the algorithm run-time and an increase in the probability of UNSAT (algorithm returning no keys) is observed. The number of returned candidate keys for the Type II analysis is reduced by over 90% as compared to the Type I analysis performed when considering Scenario III, even when allowing for a high tolerance

Table 8.2: Results of executing the GA attack on the five obfuscated analog benchmark circuits described in Section III and for the four different scenarios described in Section IV.

Scenarios	VGA (10-bits)					Mixer (10-bits)					Op-Amp (12-bits)					2-Stage VGA (12-bits)					Front-End (18-bits)				
	Tol.	Pop.	Gen.	Time	Comment	Tol.	Pop.	Gen.	Time	Comment	Tol.	Pop.	Gen.	Time	Comment	Tol.	Pop.	Gen.	Time	Comment	Tol.	Pop.	Gen.	Time	Comment
I-II	1%	35	30	15.4 H	No Correct Key	1%	35	30	15.3 H	No Correct Key	1%	65	65	65.5 H	No Correct Key	1%	65	65	65.7 H	No Correct Key	5%	90	85	Timeout	No Key
	5%	25	20	7.4 H	Correct Key	5%	20	15	4.16 H	Correct Key	5%	25	25	9.3 H	Correct Key	50%	65	65	65.7 H	No Correct Key	50%	85	85	114.3 H	No Correct Key
III - IV	10%	30	25	13.3 H	Correct Key	5%	30	35	11.2 H	Correct Key	10%	55	60	50.3 H	Correct Key	50%	65	65	65.7 H	No Correct Key	50%	85	85	114.3 H	No Correct Key

of 60% for the Type II analysis. For a standalone analog circuit (single-stage), determining different circuit parameters is straight forward (from the datasheet). However, for a larger analog circuit consisting of many sub-blocks (i.e. the analog front-end described in this dissertation), determining multiple circuit specifications is a challenge as described in Section 8.1. The results from the analysis indicate that the SMT attack is only viable for single stage analog circuits, where only one biasing node is obfuscated and the attacker is able to develop a highly accurate circuit model with tight parameter ranges to represent the oracle IC.

8.4.2 Genetic Algorithm-based Attack

In [202], an attack is presented that utilizes a genetic algorithm (GA) to determine the correct key used to secure an analog circuit. The GA attack requires information on the circuit netlist, biasing conditions, and circuit specifications. Based on all the necessary information, the attacker sets the population size, number of generations, probability of mutation, probability of crossover, chromosome size, and the initial chromosome structure [202]. In addition, based on the target circuit performances, the attacker formulates a cost-function (CF) that is minimized when executing the GA attack.

The GA attack consists of iteratively executing both the GA algorithm and simulation on the extracted netlist. The GA algorithm determines a key based on the starting chromosome. The determined key is applied as an input to the netlist representation of the circuit, where the simulator (*PySpice*) returns the circuit performances for the given key. The simulated circuit performances are then compared to the target circuit performances to calculate the cost function. The GA algorithm tunes the chromosomes that result in the lowest cost. The steps involved in execution of the GA attack include:

1. Obtaining the circuit netlist through reverse engineering and identification of the portion of the circuit that is obfuscated,
2. Mapping of the keys obfuscating the circuit to a chromosome,
3. Formulation of the cost function to minimize the error between the oracle response and the modeled circuit response based on the circuit specifications obtained from the oracle IC, and
4. Optimization of the search space and determination of the key that results in the lowest cost based on the specified population size and the number of generations.

For the GA attack, the cost function is formulated to only consider an applied key as a possible candidate when there is a 5% or less error between the target and simulated performance values. The probability of mutation is set to 20% and the number of generations and the population size are each initially set to five. Both the number of generations

and the population size are then incremented by five until the GA attack produces a correct key for three consecutive random initial chromosome values. For the analysis in this dissertation, the GA attack described in [202] is modified to utilize *Cadence* and *Spectre* for modelling, characterization, and analysis of the circuit response instead of *PySpice*. The use of *Cadence* and *Spectre* greatly improves the accuracy of the results and also facilitates the utilization of complex AC and noise related simulations at the expense of increased execution time. The single key returned by the GA attack is not guaranteed to be the actual target key used to lock the analog circuit and greatly depends on the formulation of the cost function. For the GA attack executed under Scenario I, all the information including biasing conditions, the PDK documentation, and the circuit specifications are assumed to be available to the adversary. Since the attack utilizes *Cadence* and *Spectre*, the PDK parameters including μC_{ox} , T_{ox} , and V_{th} are already assumed available. Therefore, the results for Scenario I and Scenario II are the same. For Scenario III, the attacker must also determine the biasing information in addition to the obfuscating key, where both unknowns are mapped to a single chromosome. The number of chromosome bits is defined by the size of the obfuscating key and the resolution of the biasing information. The algorithm inputs the chromosome into the simulator and based on the circuit response, tunes the chromosome such that the cost-function results in a lower error. Instead of utilizing a circuit simulator, the attack allows for the input of the chromosome into the obfuscated oracle circuit. The attacker is assumed to possess access to the key inputs

and the output pins of the oracle IC through which input-output responses are observed, and based on the responses, the chromosome values are tuned. In this dissertation, the GA attack is evaluated on the schematics of the analog benchmark circuits, which are also considered as the oracle circuits. Therefore, the attack executed considering Scenarios III and IV is identical.

The results of executing the GA attack algorithm on the five analog benchmark circuits are listed in Table 8.2. For all cases, the analysis of the GA attack is performed by initially setting the error in the *cost function* (CF) to 1% and both the population and generation sizes to 5. The *cost function* is defined as the ratio of the difference between the target circuit specification(s) and the simulated circuit specification(s) over the target circuit specification(s) as a percentage ($\frac{|Targetoutput - Simulatedoutput|}{Targetoutput} \cdot 100$). The timeout for the each attack is set to 5 days (120 hours). The population and the generations are each incremented by 5 until the 5 day timeout is reached, at which point, the error in the cost function (CF) is increased to 5%. If further timeouts occur, then the CF is increased by an additional 5%. Incrementing the error of the CF increases the probability of determining the correct key. The population and generations are both reset to an initial value of 5 each time the CF is incremented.

For a single stage circuit obfuscated by a 10-bit key, the product of the population and the generation size is set to approximately 2^{10} . The maximum population is set to 35 and the maximum number of generations is set to 30, which results in a total number

of iterations of $35 \cdot 30 = 1050$, which is approximately equal to 2^{10} . The results indicate that setting a very low CF of 1% does not lead to the correct key for any of the benchmark circuits even after the GA attack performs an equal number of iterations to a brute-force attack. When the CF is increased to 5%, the GA attack determines the correct key for the single stage benchmark circuits described in Section 8.2, specifically the VGA and the mixer, in less than 8 hours.

For the two stage circuits, specifically the op-amp and the two-stage VGA, the maximum population and the generation size are each set to 65, where the product of the two corresponds to the number of unique keys produced by a 12-bit key size ($65 \cdot 65 = 4225$, which is approximately equal to 2^{12}). Similarly, for the front-end circuit, the population and the number of generations are each set to 85, as any higher value in either the population or the generation results in a timeout. Executing the GA attack on the multi-stage analog circuits failed to determine the correct key even after the GA attack performs an equal number of iterations to a brute-force attack. However, for a two-stage obfuscated op-amp, the GA attack determined the correct key in 9.3 hours, as, unlike the two-stage VGA and the front-end, the op-amp response is mostly monotonic with respect to both the obfuscated transistor dimensions and the key-bits and includes a smaller number of local minimas. In contrast, the two-stage VGA and front-end circuits produce non-monotonic responses and consist of many local minima and maxima. Due to the higher cost and complexity, optimization algorithms including the genetic algorithm do not easily escape

Table 8.3: Results of executing the key spacing attack on the five obfuscated analog benchmark circuits described in Section III and for the four different scenarios described in Section IV.

Scen-arios	VGA (10-bits)				Mixer (10-bits)				Op-Amp (12-bits)				2-stage VGA (12-bits)				Front-End (18-bits)			
	Var	Keys	Time	Comment	Var	Keys	Time	Comment	Var	Keys	Time	Comment	Var	Keys	Time	Comment	Var	Keys	Time	Comment
I-III	1%	19	1.93 s	Correct Key	1%	11	3.02 s	Correct key	1%	44	0.52 s	Correct Key	1%	27	0.89 s	Correct Key	1%	36	1.25 s	Correct Key
	5%	8	1.27 s	Correct key	5%	7	1.34 s	Correct key	5%	40	0.48 s	Correct key	5%	13	0.91 s	Correct key	5%	21	1.46 s	Correct key
	25%	2	0.45 s	No Correct key	25%	2	0.81 s	No Correct key	30%	4	0.33 s	No Correct key	15%	9	0.61 s	No Correct key	15%	12	0.93 s	No Correct key
IV	Cannot Perform Attack																			

from such local minima and maxima. Therefore, the GA attack failed to determine the correct key applied to secure the two-stage VGA and the front-end.

8.4.3 Key Spacing

The key spacing attack presented in [200] leverages a vulnerability present in key-based obfuscation techniques. To avoid multiple correct keys and to ensure an incorrect key produces a significant degradation in the performance of a secured analog circuit, an obfuscated circuit requires an exclusion zone around the target device dimensions. The attack utilizes an SMT solver to determine candidate widths with sufficient spacing from the next closest widths, where the spacing interval is set by the attacker. The attack continues until no further candidate widths are determined. The threat model considered for the key spacing attack is that of an adversary that possesses the circuit netlist through which the obfuscated transistor sizes are targeted.

SMT analysis is performed to evaluate the execution of the key spacing attack on the five analog circuits described in Section 8.2 and for the four attack scenarios described in Section 8.3, with results as listed in Table 8.3. The attack is efficient in pruning the key

space, where the correct key for an 18-bit obfuscated front-end circuit was determined in less than 2 seconds. The user defined key spacing variance (*Var*) is the primary parameter to consider when executing the key spacing attack algorithm, which defines the spacing between two adjacent keys. If the *Var* between the two keys is greater than or equal to the user-defined parameter, then the keys are considered as candidate keys. Setting a larger *Var* results in the faster pruning of the key space and a smaller number of candidate keys, but does not guarantee the return of the correct key within the the candidate solutions as indicated by the last row of Scenarios I to III of Table 8.3 for a *Var* of 25%. In addition, a lower *Var* results in increased computational time and an increased number of candidate solutions that the attacker must then brute-force, but guarantees the presence of the correct key in the candidate solutions. Since the key spacing attack only requires the netlist of the extracted circuit, Scenarios I, II, and III result in the same solutions. However, the attack is infeasible for Scenario IV, where only the oracle IC is available to the adversary.

8.4.4 Monotonic Circuit Response Attack

For the monotonic attack, an adversary is assumed to have input-output responses of the obfuscated circuit. The attack algorithm utilizes the monotonic relationship between the given width and the output response of the circuit, which applies to a majority of the analog circuits attacked in isolation. The input-output response that is used to model the attack is determined in one of two ways: 1) simulation on the extracted netlist using

Table 8.4: Results of executing the monotonic attack on the five obfuscated analog benchmark circuits described in Section III and for the four different scenarios described in Section IV.

Scenarios	VGA (10-bits)			Mixer (10-bits)			OpAmp (12-bits)			2-Stage VGA (12-bits)			Front-End (18-bits)		
	Tolerance	Keys	Time	Tolerance	Keys	Time	Tolerance	Keys	Time	Tolerance	Keys	Time	Tolerance	Keys	Time
I	1%	1	0.22 s	1%	1	0.16 s	1%	2	0.33 s	1%	9	0.66 s	1%	313	12.58 s
	10%	3	0.26 s	10%	1	0.21 s	10%	4	0.36 s	10%	115	2.73 s	10%	2939	425.78 s
	25%	16	0.41 s	25%	30	0.63 s	25%	14	0.67s	25%	373	8.91 s	25%	9312	3051.84 s
II	1%	1	0.27 s	1%	1	0.15 s	1%	2	0.31 s	1%	1	0.50 s	1%	297	8.34 s
	10%	3	0.26 s	10%	1	0.16 s	10%	4	0.29 s	10%	38	1.13 s	10%	3097	181.98 s
	25%	16	0.46 s	25%	30	0.63 s	25%	15	0.61 s	25%	119	2.80 s	25%	8465	856.53 s
III	1%	1	0.23 s	1%	1	0.25 s	1%	2	0.35 s	1%	9	0.65 s	1%	417	12.55 s
	10%	3	0.24 s	10%	1	0.23 s	10%	4	0.42 s	10%	110	2.46 s	10%	1659	66.62 s
	25%	16	0.45 s	25%	36	0.86 s	25%	16	0.65 s	25%	372	9.46 s	25%	3538	198.64 s
IV	1%	0	65.08 s	1%	0	149.07 s	1%	0	62.66 s	1%	0	86.64 s	1%	0	119.53 s
	10%	276	3570.75 s	10%	0	148.58 s	10%	0	62.62 s	10%	No Correct Key	90.97 s	10%	No Correct Key	200.33 s
	25%	279	3662.48 s	25%	11 No Correct Key	308.42 s	25%	0	65.26 s	25%	10 No Correct Key	94.66 s	25%	13 No Correct Key	207.11 s
				90%	22	326.41 s	99%	0	62.26 s	90%	144 No Correct Key	2053.98 s	90%	10 No Correct Key	293.83 s

circuit simulators or 2) circuit response characterization on the oracle IC based on the applied key, where the adversary is assumed to possess access to the key inputs and there exists a one-to-one relation between the key inputs and the obfuscated circuit parameters.

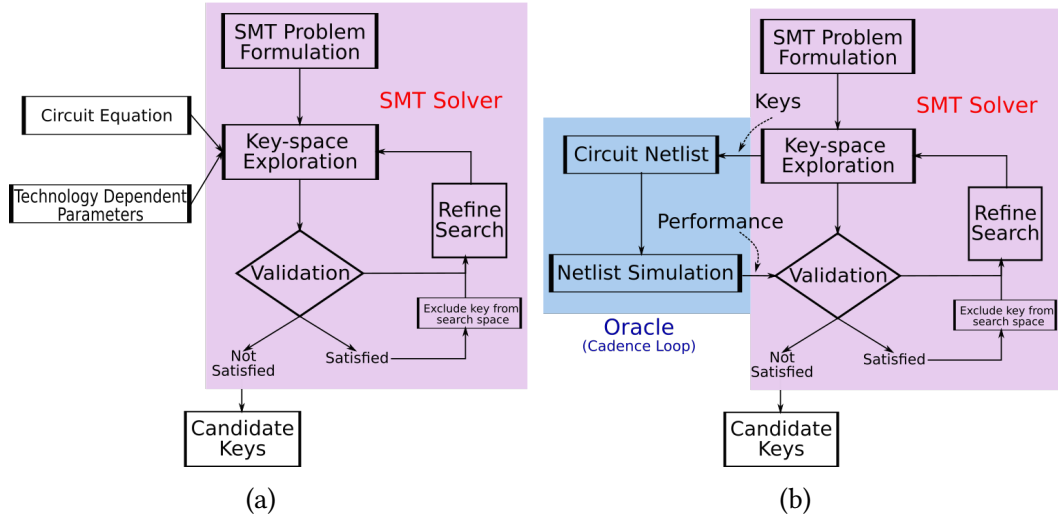


Figure 8.6: Flowchart of the SMT-based monotonic attack used for the four scenarios described in Section 8.3, where (a) circuit equations are used to obtain input-output responses for Scenarios I to III and (b) Cadence-based oracle circuit emulation is used to obtain input-output responses for Scenario IV, where the magenta box illustrates the steps performed by the SMT solver and the blue box illustrates the steps performed in Cadence.

The results of executing the monotonic attack on the five analog test circuits described in Section 8.2 and for the four different attack scenarios described in Section 8.3 are listed in Table 8.4. For Scenarios I to III, where the attacker possesses the circuit netlist, the monotonic attack is performed utilizing analog circuit models constructed from generic circuit equations with an attack flow as shown in Fig. 8.6a. However, for Scenario IV, where the attacker only possesses the oracle IC, Cadence-based schematic simulations are used only to model the oracle response and no knowledge of the circuit topology is assumed, as shown in Fig. 8.6b. The results of executing the monotonic attack on the five analog circuits indicate that the attack is very effective in determining the correct key for Scenarios I to III, where generic circuit equations, based on the extracted netlist are used to model the obfuscated analog circuit. The attack returned the keys in less than 1 second for the single-stage 10-bit obfuscated circuits, less than 10 seconds for the two-stage circuits obfuscated with a 12-bit key, and less than an hour for a three-stage front-end circuit obfuscated with an 18-bit key. The relatively fast execution time of the monotonic attack is a consequence of ignoring the transistor operating region and the non-linear parameter characteristics of the devices when developing equation based models of the circuit, which results in a monotonic circuit response and a halving of the key search space each iteration. The results of executing the monotonic attack for Scenario IV indicate that the technique is effective only for single-stage obfuscated analog circuits, specifically the single-stage VGA and the mixer. However, as the number of circuit stages increase, the

circuit response becomes non-monotonic, which results in the failure of the attack to determine the correct key of the op-amp, the two-stage VGA, and the front-end as indicated by the results for Scenario IV listed in Table 8.4.

8.5 Conclusions

In this chapter, the practical performance of four state-of-the-art analog attack techniques are evaluated for four threat scenarios. The challenges of determining the necessary information needed to efficiently execute an attack on an obfuscated circuit are described. Four different threat models are developed and analyzed, where an adversary is assumed to possess different degrees of information pertaining to the analog circuit and the implemented security features. The performance of the analog attack algorithms are evaluated on five distinct analog circuits obfuscated with a key-based parameter locking technique and with varying key-sizes and number of locked stages. The analysis of evaluating the four analog attack techniques on five distinct analog circuits indicates that the SMT-based attack performs exceptionally well in determining the correct key under Scenario I, where the attacker possesses all the circuit information except for the correct key. However, when one or more pieces of information is missing, the attack is costly and ineffective, where the SMT-attack pruned only 1.5% of the 10-bit keys obfuscating a single stage mixer when considering Scenarios II and III. In addition, as compared to the

monotonic and the key spacing attacks, the SMT attack is slower by 224x and 2240x, respectively. The results indicate that the SMT attack is only viable for single stage analog circuits where only one biasing node is obfuscated and the attacker is able to develop an extremely accurate circuit model with tight parameter ranges.

The GA attack determined the correct key for the single stage circuits in under 8 hours, but failed to determine the correct key for multi-stage circuits even when executing an equal number of iterations as a brute-force attack. The execution time of the GA attack is 121091x slower than a monotonic attack even for a single stage analog circuit attacked under scenario I. The key-spacing attack is efficient in pruning keys, where the attack is 10x faster than the monotonic attack and results in 8.3x fewer candidate keys for multi-stage analog circuits, but requires the circuit netlist. The monotonic attack determined the correct key in less than one second for single stage circuits, less than 10 seconds for two-stage circuits, and less than an hour for three-stage circuits. As the number of stages increases and for Scenario IV, the circuit response is non-monotonic, which results in the failure of both the GA and monotonic attacks in determining the correct key.

Chapter 9

Metrics For Evaluating Analog De-Obfuscation Attacks

9.1 Metrics of Evaluation

Four analog attack algorithms are evaluated on five analog circuit topologies for four attack scenarios representing a varying degree of circuit information possessed by the attacker. The four analog attack algorithms are evaluated utilizing both a performance metric T_P and a resources metric T_R . The performance metric evaluates the time required to successfully attack an analog circuit as well as the number of candidate keys returned that an attacker must then brute-force to determine the correct key. The resources metric provides an analysis of the time required to obtain necessary information needed by the

attacker to perform a successful attack on the analog circuit. Based on the performance and resource metrics, the overall attack evaluation metric T is formulated to characterize the efficacy and efficiency of the four analog attack algorithms.

9.2 Resource Metric (T_R)

The resource metric accounts for the time required to initialize the attack. The considered analog attack algorithms require prior knowledge of the obfuscated circuit and the obfuscation mechanism to effectively execute. The resource metric (T_R) accounts for the total time required to collect all the information needed to perform the attack. The resources needed to effectively execute the four analog attack algorithms on the five considered analog circuits are summarized in Table 9.2. The cost and the time to obtain each resource is correlated to the number of steps each requires.

The number of steps needed to initialize an attack on a locked analog circuit is listed in Table 9.1. From the discussion provided in Section 8.1 and in [206], reverse engineering the circuit to obtain the netlist involves the most number of steps and, therefore, is considered to be the most expensive and time consuming. The time to obtain the circuit netlist is denoted as T_N . The layering and probing methodology needed to obtain the biasing information, assuming that the attacker has already acquired the circuit netlist, involves the next highest number of steps and is, therefore, the second most expensive and time consuming, where the time to obtain the biasing information is denoted as T_B .

The process to model the circuit using standard analog circuit equations is the third most expensive and time consuming provided that the circuit netlist is already determined. The process to obtain the PDK information is considered as the fourth most expense. Although negligible, the time to obtain the PDK information from the extracted circuit netlist and the PDK documentation is denoted as T_P . The time required to model the obfuscated analog circuit is denoted as T_M . The circuit specification is the simplest and cheapest information to obtain, which involves access to the datasheets, assuming that the obfuscated analog circuits are standalone circuits and not sub-blocks of a larger circuit, where only system level specifications are provided. The time required to obtain circuit specifications is denoted as T_S .

The results of the evaluation indicate that the monotonic attack requires only the circuit model that represents the monotonic circuit performances and the circuit specifications. Obtaining the two resources accounts for a total of four steps as indicated in Table 9.2, and is, therefore, the cheapest and the fastest algorithm to initialize. However, as more biasing nodes of the circuit are obfuscated, the monotonic attack is less effective as the output responses of the circuit are non-monotonic.

The circuit information added to initialize an attack and the corresponding total number of steps required to determine the necessary information are listed in Table 9.2. The

Table 9.1: Number of steps necessary for an adversary to determine needed circuit information to execute an attack. The number of steps required to complete extraction of the biasing voltages and currents, circuit attack model(s), and the bias information assumes that the attacker possesses the circuit netlist.

Information	Steps To Determine The Information	Number of Steps
Circuit Netlist	• Target IC • Depackaging • Delayering • Imaging • Extraction • Annotation	6
Biasing Voltages and Currents	• IC for Probing • Circuit Netlist • Decapsulation • Wire Location • Probing	5
Circuit Attack Model(s)	• Oracle IC • Input-Output Responses • Circuit Modelling	3
PDK/ Process Information	• Circuit Netlist • PDK Documentation	2
Circuit Specifications	• Datasheet	1

key spacing attack is the second cheapest attack algorithm to initialize and execute. However, extensive modifications to the algorithm are needed to adapt the attack to non-key-based obfuscation techniques.

The execution of the GA attack requires the reverse engineered obfuscated netlist, the bias information, and an oracle IC [202], which results in 14 total steps to initialize the attack. The accuracy of the attack is greatly impacted by the formulation of the cost function, the initial chromosome values, the number of generations, the size of the population, the probabilities of crossover and mutation, and the tournament selection process.

Table 9.2: Required information needed to perform each attack on the five obfuscated analog circuits. A check mark implies needed information.

Attack Techniques	Required Information					Total No. Of Steps
	Netlist	Biasing Information	Models	PDK Information	Circuit Specifications	
SMT [201]	✓	✓	✓	✓	✓	17
GA [202]	✓	✓	✗	✓	✓	14
Key Spacing [200]	✓	✗	✗	✗	✗	6
Monotonic Attack [200]	✗	✗	✓	✗	✓	4

In addition, the single candidate key returned by the GA attack is not guaranteed to be the target key used to lock the analog circuit.

The SMT attack requires information on the circuit netlist, the bias voltages and/or currents, the process design kit (PDK) documentation, and the circuit specifications [201] to model the obfuscated circuit and formulate the SMT problem correctly. Obtaining all the necessary information to initialize the SMT attack requires a total of 17 steps as indicated in Table 9.2. The accuracy and the efficiency of the algorithm depends significantly on the circuit models, where the complexity increases for larger multi-stage circuits. When one or more pieces of information is missing, the attack is inefficient in pruning the key-space, as indicated by the results for attack Scenarios II and III listed in Table 8.1. In addition, as the key size increases, the number of returned candidate keys and the execution time exponentially increase. From analyzing results listed in Table 9.3, the resource metric (T_R) evaluated for different analog attack techniques is ordered as

$$T_{R_{MA}} < T_{R_{KS}} < T_{R_{SMT}} < T_{R_{GA}}, \quad (9.1)$$

Table 9.3: Total time required for each attack technique to obtain all the information necessary to de-obfuscate an analog circuit.

Attack Technique	Time to Obtain Required Information
SMT	$T_{R_{SMT}} = T_N + T_B + T_M + T_P + T_S$
GA	$T_{R_{GA}} = T_N + T_B + T_P + T_S$
Key Spacing	$T_{R_{KS}} = T_N$
Monotonic Attack	$T_{R_{MA}} = T_M + T_S$

where the time components for each attack algorithm are also listed.

9.3 Performance Metric (T_P)

The performance metric (T_P) accounts for the time to determine the candidate keys from the obfuscated analog circuit and the time to determine the correct key from the candidate keys when executing a brute-force attack. The time to determine the candidate key is denoted as T_{Solve} and the time to determine the correct key from the candidate keys executing a brute-force attack is denoted as T_{BF} . The results provided in Section 8.4 for the SMT, GA, key-spacing, and monotonic attacks are utilized to calculate the performance metric.

Scenario I is considered for the evaluation of the performance metric for all the analog attack algorithms. The *tolerance* for the SMT attack, GA attack, and monotonic response attack and the key spacing variance *var* for the key-spacing attack are set to 10%. For the evaluation, the key size is increased from 9-bits to 30-bits for the single-stage VGA, 2-stage VGA, and the analog front-end described in Section 8.2. The number of candidate

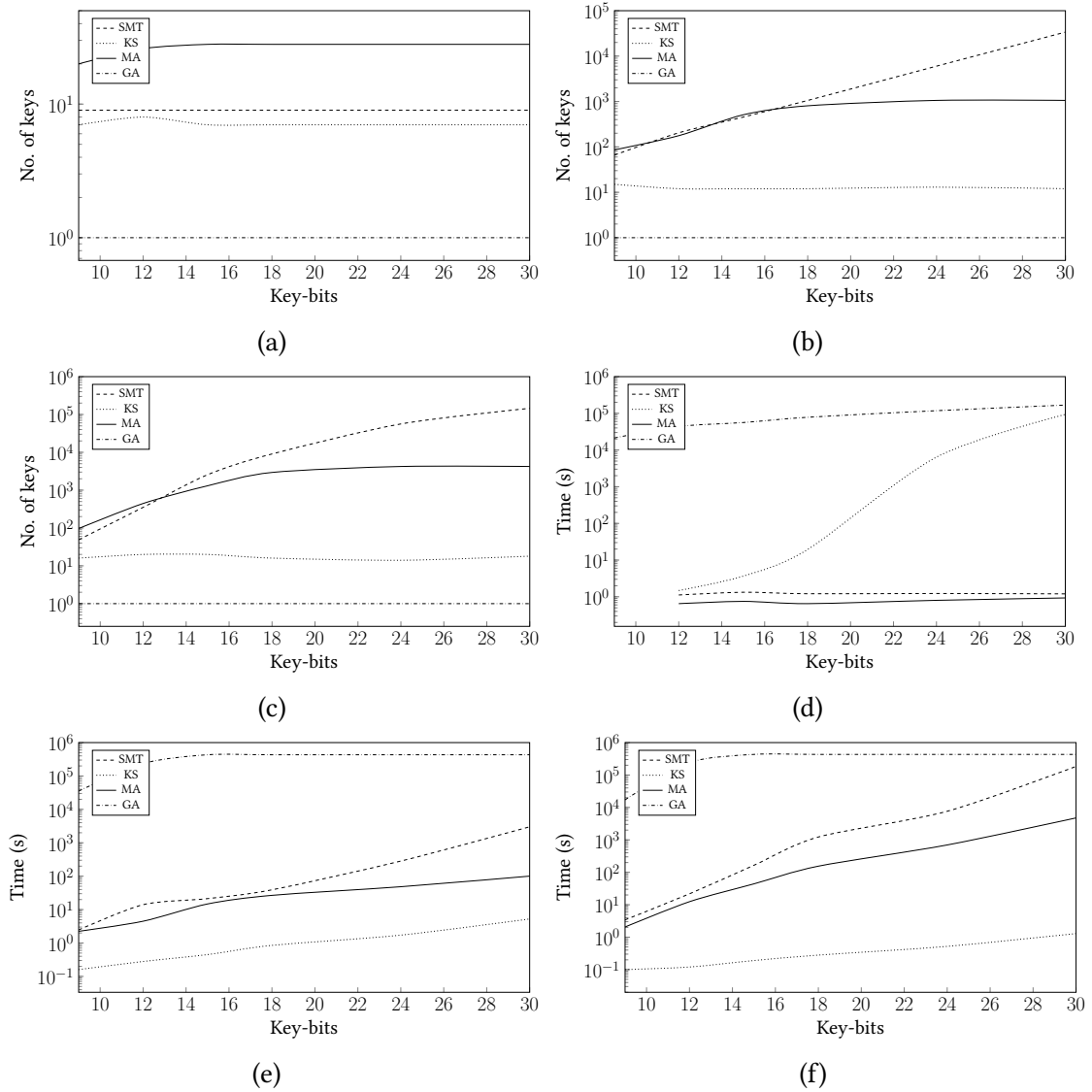


Figure 9.1: Characterization of the variation in the number of returned keys for a (a) single-stage VGA, (b) 2-stage VGA, and (c) analog front-end circuit, and an analysis of the execution time required to return candidate keys for a (d) single-stage VGA, (e) 2-stage VGA, and (f) analog front-end circuit. The analysis for both the number of returned keys and the execution time is performed for the SMT, GA, key-spacing, and monotonic response attack algorithms for key sizes of 9-bits to 30-bits.

keys and the time required to determine the candidate keys are shown in Fig. 9.1.

The results indicate that for a single-stage analog circuit, the SMT-attack is the fastest

to execute for key-sizes less than 10-bits, but as the number of key-bits increases beyond 10, the monotonic attack becomes the fastest attack to execute. The time to determine the candidate solution increases exponentially for the key-spacing attack executed on a single-stage analog circuit. The GA-based attack requires the longest amount of time to execute, with approximately 47 hours needed to determine a single key for a 30-bit obfuscated VGA. Based on Fig. 9.1a and Fig. 9.1d, the performance metric evaluated for the different analog attack algorithms and for a single-stage obfuscated analog circuit results in an order of execution time of

$$T_{P_{MA}} < T_{P_{SMT}} < T_{P_{KS}} < T_{P_{GA}}. \quad (9.2)$$

However, as the number of stages increases and as the key-bits are split across multiple sub-blocks, the key-spacing attack executes the fastest and returns the fewest number of candidate keys as compared to the SMT-based attack and the monotonic response attack, as indicated by the results shown in Figs. 9.1d - 9.1e and Figs. 9.1a - 9.1c, respectively. The advantage of executing the key-spacing attack is that, as long as the extracted netlist is available, the attack is not dependent on any individual circuit equation and is executed independently and in parallel across each obfuscated sub-block. The GA-based attack returns a single candidate key; however, the key is not guaranteed to be correct and the attack timed out after 5 days (432000 seconds) for both the 2-stage VGA and the analog front-end for key sizes greater than 15-bits. In addition, the GA-based attack required

the longest amount of time to execute as compared to all other attack techniques. The SMT-attack returns the fewest number of candidate keys, but requires a longer execution time than the monotonic attack for key sizes less than 12-bits. For key-sizes greater than 12-bits, both the number of candidate keys and the execution time are greater than that of the monotonic attack for both the 2-stage VGA and the front-end circuit. Based on the results shown in Fig. 9.1d - 9.1e and Fig. 9.1a - 9.1c, the performance metric evaluated for the different analog attack algorithms and for a multi-stage obfuscated analog circuit results in an execution order of

$$T_{P_{KS}} < T_{P_{MA}} < T_{P_{SMT}} < T_{P_{GA}}. \quad (9.3)$$

9.4 Attack Evaluation Metric (T)

The attack evaluation metric is defined as the sum of the resource metric with the performance metric. The attack evaluation metric is used to rank different analog attack techniques for two criteria. The first criterion accounts for Scenario I and considers the conditions described in [201], [202], and [200] for the SMT attack, GA attack, and both the key-spacing attack and monotonic attack, respectively. For the first criterion, the time to obtain the circuit netlist far outweighs the time to obtain any other information as well as the time to determine the correct key. Therefore, for Scenario I, the assessment of the attack evaluation metric for the different attack techniques results in a cost order of

$$T_{MA} < T_{KS} < T_{SMT} < T_{GA}. \quad (9.4)$$

For the second criterion, only the circuit netlist and the circuit specifications are assumed available. For the proposed scenario, assessment of the attack evaluation metric results in a cost order of

$$T_{KS} < T_{MA} < T_{SMT} < T_{GA}. \quad (9.5)$$

The results from the attack evaluation metric suggest that the monotonic attack is the fastest and the cheapest technique to de-obfuscate a locked analog circuit. The monotonic attack is considered a black-box attack, as only the oracle circuit and high-level circuit specifications are required. However, if the locked analog circuit consists of multiple sub-blocks that produce a non-monotonic parameter response, the monotonic attack fails to determine the correct key.

When the monotonic attack fails, time and resources are invested to extract the locked analog circuit netlist through reverse engineering. Once the circuit netlist is obtained, the key-spacing attack is the fastest and cheapest means to attack the obfuscated analog circuit. The key-spacing attack only requires the circuit netlist. However, the key-spacing attack is only applicable to an analog circuit obfuscated through key-based obfuscation. For analog circuits obfuscated with non-key-based obfuscation including multi-threshold

obfuscation, SMT-based and GA-based techniques are applicable.

Although the GA attack executes for Scenario I, the calculated performance metric of the GA attack T_{PGA} is higher than that of the SMT-based attack T_{PSMT} . Therefore, the SMT-based attack is cheaper and faster to execute than the GA attack. In addition, the GA attack failed to return the correct key for multi-stage locked analog circuits as indicated by the results listed in Table 8.2. Furthermore, the execution time of the GA attack is approximately 50 to 300 times longer than that of the SMT attack for single stage circuits when considering Scenario I. The execution time of the GA is further reduced by using less complex circuit simulators such as *PySpice* at the expense of accuracy. To compensate for less accurate circuit simulators, larger population and generation sizes are required, that increase the number of iterations needed to complete the GA attack, which results in higher number of iterations of the GA attack than that of the SMT-based attack. Therefore, the SMT-based technique is faster and more efficient than the GA attack.

9.5 Conclusions

Metrics to evaluate the performance of the four analog attack techniques are developed based on the attack initialization time and the attack execution time. The evaluation of the metrics indicate that the monotonic attack is the fastest and the cheapest attack when the attacker possesses no circuit information except the circuit specifications, but fails for multi-stage circuits with non-monotonic responses. However, when the attacker

possesses the circuit netlist, the key-spacing attack is the fastest and the cheapest attack to de-obfuscate a multi-stage obfuscated analog circuit. Execution of the SMT attack is faster than the key-spacing attack for a single stage analog circuit attacked under scenario I, but the execution time and the number of candidate solutions increases exponentially for an obfuscated multi-stage analog circuit. The execution time of the genetic algorithm (GA) is the longest, where the execution time is 121091x longer than the monotonic attack even for a single stage analog circuit attacked under scenario I.

Metrics Used to Characterize Digital Locking Techniques The primary metrics used in the digital hardware security to measure the security of digital locking techniques include key strength, corruptibility, and confidentiality. Key strength metric the difficulty of extracting a correct key from a locked circuit using a deobfuscation technique. The Boolean satisfiability (SAT) attack is prominently used to²²³ measure the key strengths of digital locking techniques because the methodology is applicable to wide range of key-based locking techniques. The corruptibility of an incorrect key is defined as the number of input patterns that produce at least one incorrect output value divided by the total number of input patterns. The objective of a digital locking technique is to thwart unauthorized access to a chip by requiring a correct key for operation. Consequently, applying an incorrect key to a circuit results in some degree of alteration to the output values. The degree of alteration induced by an incorrect key serves as an indicator of the adverse effect on functionality, which is commonly quantified by assessing changes in the truth or state

table. The confidentiality metric assesses the degree to which a sensitive function is concealed relative to its original functionality. In certain scenarios, only restricting access to the intended functionality is inadequate. There are instances where the design entity aims to conceal the functionality of the design entirely. In such cases, the primary objective of the lock is to ensure confidentiality by evaluating the extent to which a critical circuit functionality are masked when an incorrect key is applied compared to when the correct key is inputted. Logic locks inherently contribute to a degree of confidentiality since an incorrect key input results in alternative, incorrect functionality, while a correct key triggers the intended functionality.

Chapter 10

DNA: DC Nodal Analysis Attack for Evaluation of Analog Obfuscation Techniques

Analog attack methodologies including one that utilize satisfiability modulo theory (SMT) [200, 201] and optimization [202] algorithms have been presented to evaluate the security of analog obfuscated circuits that assume a similar oracle assisted threat model as that of digital SAT based attacks [203–205]. Current obfuscation techniques are utilized to protect analog circuits against threat models that assume a different degree of information available to the attacker. When one or more pieces of information is unavailable for the attack (Scenario II and III described in 8.3), the current attack algorithms result in a larger

number of returned candidate keys, increased computational time, and/or no correct returned candidate key as demonstrated in Chapter 8. Therefore, unlike the digital circuit domain, where the satisfiability (SAT) attacks are the de-facto standard for measuring the resiliency of the key-based logic locking techniques, no such standards exists in the analog circuit domain.

In this chapter, a novel DC nodal analysis attack algorithm is presented to evaluate the strength of different analog obfuscation techniques. The presented algorithm, similar to the SAT-attack in the digital domain, requires knowledge of only the circuit netlist and the DC input-output response of the oracle IC. In addition, the attack algorithm is applicable to all of the current analog obfuscation techniques. The utilization of the DC nodal analysis attack allows for the quantification of the security provided by analog obfuscation techniques. The primary contributions of the chapter include:

1. Development of a novel DC nodal analysis attack algorithm based on Kirchhoff's current and voltage laws (KCL and KVL) to efficiently explore the obfuscation space and determine the correct control parameters of the obfuscated analog circuit. The attack utilizes the SMT algorithm to propagate the DC signal from the input to the output of the circuit while determining the obfuscated parameters (obfuscation keys or the high, low, and nominal threshold transistors) of traversed intermediate nodes. Only when the DC voltage of the output matches the oracle, as a result of setting all of the intermediate node voltages and currents correctly, are the set of obfuscated

parameters considered as the candidate solutions.

2. Application of the DC nodal analysis attack on four novel threat models that emulate real world attack scenarios, where one or more sources of information are progressively made unavailable to the attacker. The current state-of-the-art attack algorithms assume that the adversary possesses all required circuit information [200–202]. However, an attack in the field is far from ideal. In this dissertation, the presented attack algorithm is shown to execute successfully and perform efficiently even when one or more sources of information on the IC is unavailable.
3. Evaluation of the DC nodal analysis analog attack methodology on four distinct analog circuit topologies with varying number of stages, obfuscated nodes, and key sizes. The current analog attack techniques are not evaluated on multi-stage and multi-node obfuscated analog circuits.
4. Execution of the algorithm on circuits secured with current state-of-the-art analog obfuscation techniques including key-based parameter obfuscation and multi-threshold obfuscation. In addition, the cost, set-up time, and the performance of the presented algorithm is compared with prior analog attack techniques.

10.1 DC Nodal Analysis Attack

The proposed SMT technique utilizes a general representation of Kirchhoff's voltage law (KVL) and Kirchhoff's current law (KCL) to efficiently explore the key-space and determine the correct key of an obfuscated analog circuit, which is in contrast to the equation-based SMT attack [201] that requires circuit specific equations that tend to be less accurate in advanced technology nodes and for scenarios where limited circuit information is available. Compared to the SMT key-spacing algorithm, the proposed technique is applicable to non-key-based analog obfuscation including multi-threshold obfuscation. In this dissertation, key-based parameter obfuscation and non-key-based multi-threshold obfuscation are evaluated with the DNA attack. The two locking techniques are selected as each represents a general family of obfuscation methodologies that apply to a wide range of threat scenarios and analog circuit topologies.

The assumed threat scenario includes access to the obfuscated analog circuit netlist, which is used to generate the SMT circuit model. In addition, access to an oracle IC is necessary to determine the DC voltage response of the circuit for a given input voltage and the overall current drawn by the circuit. The primary aim of the attack is to determine the keys that result in the same output voltage as the oracle given the same input voltage. The input voltage is applied to the circuit and internal node voltages and currents of the obfuscated circuit are computed using the generalized KCL and KVL equations and device

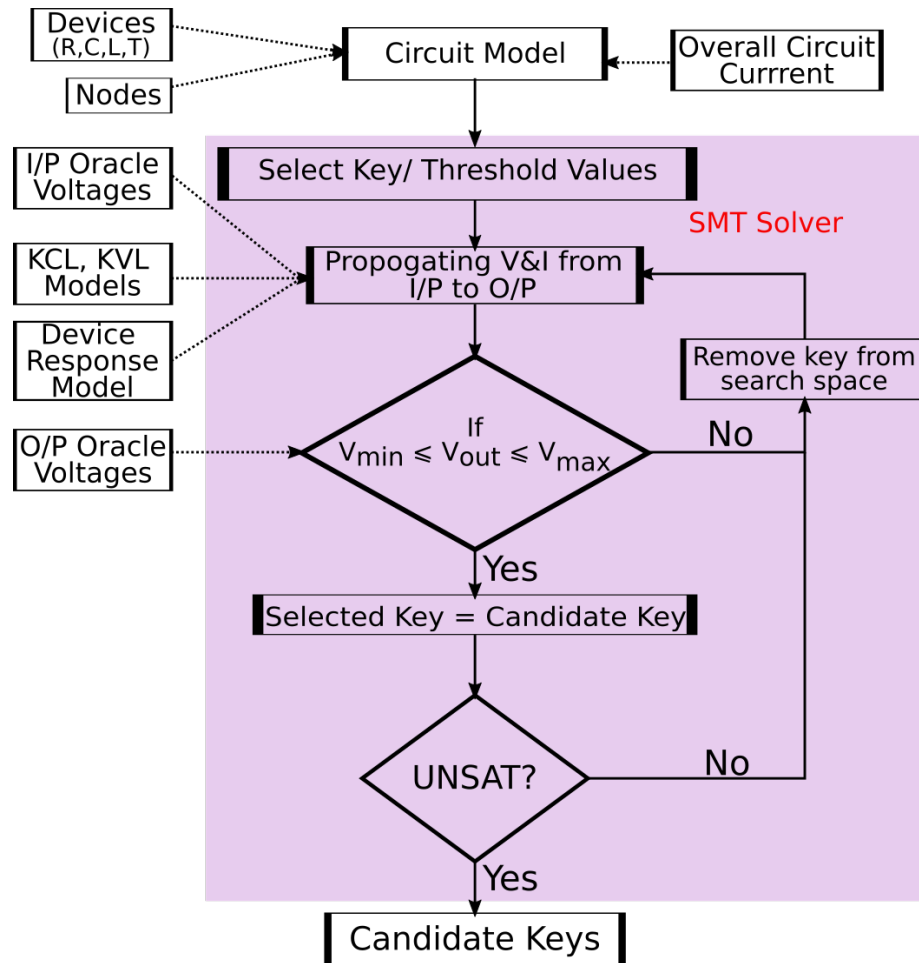


Figure 10.1: Flowchart highlighting the primary steps of the DC nodal analysis attack on key or threshold voltage obfuscated circuit.

models.

10.1.1 Application of DC Nodal Analysis Attack on Key-based Parameter Obfuscation

The primary steps of the DNA attack executing on the key-based parameter obfuscation technique are shown in Fig. 10.1, where, based on the extracted netlist, a simple

Algorithm 10: DC Nodal Analysis Attack for Key-based Analog Obfuscation

Input: circuit model, width $\vec{W}_v, \vec{X} \in b[0, 1]$, voltage tolerance $V_{Tol}, I_{total}, V_{in},$
 $V_{oracle_out};$
 $W_i = \vec{W}_v \cdot \vec{X}_i;$
 $S_1 = W \wedge (sum(\vec{X}) > 0);$
 $S_1 = SMTSolver;$
 $candidate_widths = [];$
 $oracle_min = V_{oracle_out} \cdot (1 - V_{Tol});$
 $oracle_max = V_{oracle_out} \cdot (1 + V_{Tol});$
 $S_1.add(V_{out} = compute_output_voltage(W));$
 $S_1 = S_1 \wedge oracle_min \leq V_{out} \leq oracle_max;$
while SAT[S_i] **do**
 $candidate_keys.append(\vec{X}_i);$
 $S_{i+1} = S_i \wedge (W \neq W_i);$
end
return candidate_widths;

verilog model is constructed of the obfuscated analog circuit that consists of devices including resistors, capacitors, inductors, and transistors. DC models of the devices are developed that govern the voltage at internal nodes and the current through given paths. When developing models of components, the inductors are treated as short circuits, the capacitors as open circuits, current and voltage drops across resistors are represented with standard resistance model, and the voltages and currents through the transistors are defined by a square-law model. The long channel (square-law) model of a transistor is utilized with the DC nodal analysis attack rather than a short-channel transistor model [215] as the long channel model provides less complexity when modeling transistor currents and voltages, which results in a faster execution time of the attack. In addition,

although digital circuits benefit from technology scaling, analog circuits are typically designed either in less advanced technology nodes or are implemented with non-minimum channel lengths to avoid a reduction in the gain of devices, shifts in the threshold voltage, increases in mismatch, and noise that arises due to short-channel effects [216–219]. Therefore, the use of long-channel models for the transistors provides a sufficiently accurate representation of the transistor voltages, currents, and region of operation for the attack to execute. Furthermore, the DC nodal analysis attack utilizes a range of values for μC_{ox} , V_{th} , and the output voltages, rather than a single value for each, which further minimizes the modeling error due to the use of the long channel model.

The pseudocode for the DNA attack is provided as Algorithm 10. The input to the algorithm includes the circuit model, the list of obfuscated widths ($\vec{W}_v$), the key-bits ($\vec{X}$) used to set the active widths, the user-defined acceptable variation in the computed output voltage V_{Tol} from the SMT solver, the total current of the oracle circuit I_{total} , the input DC voltage V_{in} , and the output voltage of the oracle IC $V_{oracle_{out}}$ for the given input voltage V_{in} . Initially, the SMT solver selects a random key sequence $\vec{X}_i$ and then determines the active width segment W_i of an obfuscated transistor based on the selected $\vec{X}_i$. The user-specified tolerance V_{Tol} is then used to compute the range of acceptable SMT-computed output voltages, bounded by the minimum voltage $oracle_min$ and the maximum voltage $oracle_max$ of the oracle IC. Based on the applied inputs, the SMT solver is used to compute the output voltage V_{out} , where the SMT solver propagates the V_{in} voltage through

the circuit to the output node using the KCL and KVL equations and the generated circuit model. Constraints to the SMT solver are added that force a search for widths that result in V_{out} within the range of $oracle_min$ and $oracle_max$. If the output voltage is within the user specified tolerance, the selected key $\vec{X}_i$ is considered as a candidate and constraints are added to allow for the determination of different keys that meet the target conditions. The SMT solver terminates after exhaustively searching the key-space until *UNSAT* (not satisfiable) conditions are met and returns the list of candidate keys. A brute-force attack is then performed on the returned candidate key list to determine the correct key.

10.1.2 Application of DC Nodal Analysis Attack on Multi-threshold Obfuscation

The DNA attack executed on circuits secured with the key-based parameter obfuscation technique is slightly modified for the analysis of the multi-threshold obfuscation technique. The circuit model from the extracted netlist is constructed, where instead of using keys to set the transistor parameters in the evaluation of the key-based parameter obfuscation technique, the keys in the analysis of the multi-threshold obfuscation technique are the selection of the process defined threshold voltages V_T for each transistor. Each transistor in the secured analog circuit is obfuscated by a three bit hot encoding corresponding to the three available transistor thresholds $V_{th_{nominal}}$, $V_{th_{high}}$, and $V_{th_{low}}$, where a high bit 1 represents the selection of a particular transistor threshold voltage. The least

significant bit (LSB) of the three bit encoding is mapped to represent a low threshold transistor, the middle bit is mapped to a nominal threshold transistor, and the most significant bit (MSB) represents a high threshold transistor. The aim of the attack is to select the threshold voltages of all the transistors in the circuit that result in the same output DC voltage as the oracle for the same DC input voltage.

The pseudocode describing the key steps of the DC nodal analysis attack executed on a circuit secured with multi-threshold obfuscation is provided as Algorithm 11. The inputs to the attack algorithm include the model of the circuit, the number of threshold voltages $\vec{V}_T$ available in the PDK, the key-bits $\vec{X}$ used to select the threshold voltage V_T of each transistor, the acceptable voltage tolerance V_{Tol} of the output signal set by the adversary, the current I_{total} drawn by the oracle, the DC voltage V_{in} of the input, and the DC voltage V_{out} of the output. An additional constraint $sum(\vec{X} == 1)$ specific to multi-threshold obfuscation is added to prevent the solver from assigning more than one V_T for a given transistor, while allowing the solver to select a V_T for each transistor. Initially, the solver selects a random key $\vec{X}_i$ that assigns a threshold voltage V_T for each transistor of the circuit and determines the corresponding DC output voltage V_{out} by solving the KCL and KVL equations. If the determined V_{out} falls within the $oracle_{min}$ and $oracle_{max}$ range, the selected set of V_T s are added to the list of candidate keys. Else constraints are added that force the solver to search for different V_T combinations that result in a V_{out} that falls within the $oracle_{min}$ and $oracle_{max}$ range. The solver terminates either when the five day

time constraint is met or when the search space is exhaustively searched resulting in an *UNSAT* condition. All the candidate keys are then returned by the solver.

Algorithm 11: DC Nodal Analysis Attack for Multi-threshold Obfuscation

Input: circuit model, threshold voltages $\vec{V}_T, \vec{X} \in b[0, 1]$, voltage tolerance V_{Tol} ,
 $I_{total}, V_{in}, V_{oracle_{out}}$;
 $W_i = \vec{V}_T \cdot \vec{X}_i$;
 $S_1 = SMTSolver$;

//Device specifications added to S1 model
 $S_1 = W \wedge (sum(\vec{X}_i) == 1)$;

$candidate_widths = []$;
 $oracle_min = V_{oracle_{out}} \cdot (1 - V_{Tol})$;
 $oracle_max = V_{oracle_{out}} \cdot (1 + V_{Tol})$;
 $S_1.add(V_{out} = compute_output_voltage(\vec{V}_T))$;
 $S_1 = S_1 \wedge oracle_min \leq V_{out} \leq oracle_max$;

while $SAT[S_i]$ **do**
 $candidate_key.append(\vec{X}_i)$;
 $S_{i+1} = S_i \wedge (W \neq W_i)$;
end
return $candidate_widths$;

10.2 Performance of DC Nodal Analysis Attack on Different Analog Obfuscation Techniques

In this section, the DNA attack is evaluated on both parameter obfuscation and multi-threshold obfuscation for the Scenarios I, II, and III as described in Section 8.3. As the proposed attack requires the obfuscated circuit netlist, the DNA attack is unable to execute under Scenario IV. In addition, four analog circuits used for evaluation of the proposed

Table 10.1: Results of executing the DC nodal analysis attack on four analog benchmark circuits obfuscated using the key-based parameter locking technique described in Section 8.2 and for three different scenarios described in Section 8.3.

Scenarios	Circuits	No. of Keys	Time	V_{th} Variation	β Variation	V_{out} Variation	Comment
I	Mixer (10-bits)	31	154.22 s	5.00%	5.00%	5.00%	Correct Key
	VGA (10-bits)	16	234.82 s	5.00%	5.00%	5.00%	No Correct Key
		23	518.59 s	5.00%	5.00%	10.00%	Correct Key
	2-stage VGA (12-bits)	598	154384.23 s	5.00%	5.00%	5.00%	No Correct Key
		668	181080.32 s	10.00%	10.00%	10.00%	Correct Key
	Front-end (18-bits)	634	432000 s	5.00%	5.00%	5.00%	No Correct Key
		711	432000 s	10.00%	10.00%	10.00%	Correct Key
II	Mixer (10-bits)	63	134.45 s	25.00%	50.00%	5.00%	Correct Key
	VGA (10-bits)	53	234.17 s	25.00%	50.00%	10.00%	Correct Key
	2-stage VGA (12-bits)	987	266410. 32 s	25.00%	50.00%	10.00%	Correct Key
	Front-end (18-bits)	1189	432000 s	25.00%	50.00%	10.00%	Correct Key
III	Mixer (10-bits)	378	789.64 s	25.00%	50.00%	5.00%	No Correct Key
		464	889.51 s	25.00%	50.00%	10.00%	Correct Key
	VGA (10-bits)	711	2368.38 s	25.00%	50.00%	10.00%	Correct Key
	2-stage VGA (12-bits)	1489	296287.56 s	25.00%	50.00%	10.00%	Correct Key
	Front-end (18-bits)	2467	432000 s	25.00%	50.00%	10.00%	Correct Key

DNA attack. The four analog circuits include two single-stage circuits, one two-stage circuit, and one three-stage circuit as described in Section 8.2.

The evaluation of the analog attacks is performed on a server that includes two 12-core Intel Xeon E5 CPUs and 96 GB of DDR4 memory. The timeout for the execution of the attack is set to 5 days (120 hours). If the set timeout is reached, the execution of the attack is terminated and the candidate keys determined to that point are returned for the analysis of the attack.

Three tuning knobs, the allowed variation in each of V_{th} , β , and V_{out} , provide different degrees of constraint to the attack. The specified variation in V_{th} sets a range of allowed threshold voltages centered around a nominal value, which is applied to the solver. As an example, when the threshold voltage is set to 0.4 V and the variation in V_{th} is set to 10%, the

threshold voltage range provided to the solver is 0.36 V to 0.44 V. The specified variation in β sets a range of μC_{ox} values centered around a nominal β value. The variation in V_{out} provides a range of allowed output voltages to the solver.

Initially, the maximum allowed variation in each of V_{th} , β , and V_{out} is set to 5%. If the correct key is not returned by the solver, the variation in V_{out} is increased by 5%. If the 5% increase in the variation of V_{out} does not result in a correct key, the variations in V_{th} and β are then increased by 5%. The cycle of incrementing the variation in V_{out} followed by the variations in both V_{th} and β is continued until the solver returns the correct key. The variation in V_{out} that results in the correct key under Scenario I is used for Scenarios II and III.

For Scenario I, where the attacker possesses all the information except for the correct key, ranges of the threshold voltage V_{th} , β , and the output voltage V_{out} are provided to the solver based on the initial specified percentage variation in each parameters, centered with ranges around target values provided in the PDK. For Scenarios II and III, where the attacker does not possess the PDK information, ranges for the threshold voltage V_{th} and β are provided to the solver by determining the variation of each centered around generic values of V_{th} and β , respectively. However, as V_{out} is known, the variation in V_{out} is centered around the correct value.

For Scenario III, where the attacker does not possess the biasing information of the

circuit, the range of bias voltages is set by the operating voltages of the transistors, specifically the gate to source voltage V_{gs} . For the NMOS transistors, the lower bound of the range is set to V_{th} and the upper bound to V_{dd} . For the PMOS transistors, the lower bound of the V_{bias} range is set to $|V_{th}| - V_{dd}$ and the upper bound to 0. To set the range of bias currents, the maximum and the minimum current through the branches of the circuit is determined. Lower bounds of the V_{th} and β ranges are used to calculate the smallest current through the transistors utilizing the square-law equation. The maximum current is determined by setting the gate voltages of the NMOS and PMOS transistors to V_{dd} and 0, respectively, while also setting the upper bound of both V_{th} and β . The range of bias currents through a transistor obfuscated with key-based parameter obfuscation is determined based on the size of the smallest and the largest transistor comprising the obfuscation sub-transistor array. The smallest transistor in the obfuscation sub-transistor array is used to determine the lower bound of the bias current. The largest transistor is the sum of all the obfuscation sub-transistors in the array and is used to determine the upper bound of the bias current.

10.2.1 Performance Evaluation For Key-based Parameter-Obfuscated Circuits

The DNA attack is executed on the benchmark analog circuits presented in Section 8.2 and for the three attack scenarios described in Section 8.3. As discussed previously, for

Scenario I, the variation in both V_{th} and β is centered around nominal values described by the PDK. For Scenarios II and III, the nominal V_{th_n} and V_{th_p} are set to 0.4 V and -0.4 V, respectively, and with an initial allowed variation of 25%. In addition, for Scenarios II and III, β_n and β_p are set to $200 \frac{\mu A}{V^2}$ and $55 \frac{\mu A}{V^2}$, respectively, and with an initial allowed variation of 50%.

The results of executing the DNA attack on the mixer, VGA, 2-stage, and front-end circuits are listed in Table 10.1. For the single stage circuits under Scenario I, the correct key is determined for V_{th} and β variation of 5%. However, a variation of 5% in V_{out} did not result in a correct key for a single-stage VGA obfuscated with a 10-bit key. Therefore, the allowed variation in V_{out} was increased by 5% for an overall variation of 10%, which allowed the solver to determine the correct key. On average 97.21% of the keys were eliminated from the search space of the single-stage circuits under Scenario I, with an average time to determine the candidate keys of 336.41 seconds. Under Scenario II, the proposed attack eliminated approximately 94.33% of the search space in under 185 seconds. Under Scenario III, the DNA attack executed on the mixer did not return the correct key when the variation in V_{out} was set to 5%. However, when the variation in V_{out} was set to 10%, the attack determined the correct key in under 15 minutes. In addition, for single stage circuits considered under Scenario III, the DC nodal analysis attack eliminated approximately 43% of the search space in less than 30 minutes.

For the two-stage circuit obfuscated with a 12-bit key, the 5% variation in V_{out} was not

sufficient to return the correct key. The variation in V_{out} was, therefore, increased to 10%, which resulted in the solver returning the correct key. Under Scenario I, the DC nodal analysis attack eliminated 85.4% of the key-space in under 51 hours. For Scenario II and Scenario III, the proposed attack eliminated approximately 76% and 64% of the key-space in under 74 hours and 83 hours, respectively.

For the three-stage front-end circuit obfuscated by an 18-bit key, setting the variation in V_{out} to 10% resulted in the solver returning the correct key. For Scenario I, the DNA attack eliminated 99.72% of the key space. However, the solver determined 711 candidate keys in five days, which require the adversary to perform a brute-force attack to determine the correct key. The proposed attack eliminated 99.5% and 99.1% of the keys from the search space when considering Scenarios II and III, respectively. The execution time of the attack for both Scenario II and III was limited to maximum of five days (432000 seconds).

10.2.2 Performance Evaluation on Multi-threshold Obfuscation

The execution of the DC nodal analysis attack on circuits secured with multi-threshold obfuscation is modified from the execution of the attack on circuits secured with key-based obfuscation. Instead of setting transistor dimensions with the locking key, the attack maps the three different transistor threshold voltages of the PDK, which is assumed to include low threshold (LVT), nominal threshold (NVT), and high threshold (HVT) transistors, to locking keys. Each transistor is obfuscated by a key, where the number of key

Table 10.2: Results of executing DC nodal analysis attack on four analog benchmark circuits masked using the multi-threshold obfuscation technique described in Section III and for the three different scenarios described in Section IV.

Scenarios	Circuits	N	No. of Keys	Time	Vth Variation	Beta Variation	Vout Variation	Comment
I	Mixer	4	3	51.26 s	5.00%	5.00%	10.00%	No Correct Key
			4	65.31 s	10.00%	10.00%	10.00%	Correct Key
	VGA	5	31	85.19 s	5.00%	5.00%	10.00%	Correct Key
	2-stage VGA	10	78	1068.15 s	5.00%	5.00%	10.00%	No Correct Key
			91	1550.34 s	10.00%	10.00%	10.00%	Correct Key
	Front-end	14	258	36843.57 s	5.00%	5.00%	10.00%	No Correct Key
			336	47864.24 s	10.00%	10.00%	10.00%	Correct Key
II	Mixer	4	4	559.79 s	20.00%	50.00%	10.00%	Correct Key
	VGA	5	44	285.31 s	20.00%	50.00%	10.00%	Correct Key
	2-stage VGA	10	220	2136.13 s	20.00%	50.00%	10.00%	Correct Key
	Front-end	14	543	77544.43 s	20.00%	50.00%	10.00%	Correct Key
III	Mixer	4	4	938.52 s	20.00%	50.00%	10.00%	Correct Key
	VGA	5	73	322.63 s	20.00%	50.00%	10.00%	Correct Key
	2-stage VGA	10	243	23418.37 s	20.00%	50.00%	10.00%	Correct Key
	Front-end	14	721	181440.35 s	20.00%	50.00%	10.00%	Correct Key

bits is equal to the number of different threshold voltages available in the given process node. If a set of differential pair transistors exist in a circuit, both the transistors of the differential pair are mapped to one key bit. The total number of key bits equals $T \cdot N$, where T is the number of threshold voltages available in the PDK and N is the number of transistors and/or number of differential transistor pairs in the obfuscated analog circuit. The total number of key combinations possible in an obfuscated circuit is T^N .

For the single-stage VGA shown in Fig. 8.2a, N is set to five, with transistors $M_{1,2}$, M_3 , $M_{4,5}$, $M_{6,7}$, and $M_{8,9}$ obfuscated. $M_{x,y}$ indicates that transistor M_x and transistor M_y share the same key bit as the transistors are part of a differential pair. The number of transistors and/or the number of differential transistor pairs N for the mixer circuit shown in the Fig. 8.4 is four, with transistors $M_{1,2}$, $M_{3,4}$, $M_{5,6}$, and M_7 comprising the set of obfuscated devices. The two-stage VGA shown in Fig. 8.2b is comprised of two VGAs,

each with N equal to five. The overall N for the two-stage VGA is, therefore, equal to 10. Similarly, for the three-stage front-end circuit shown in Fig. 8.5, which is comprised of a mixer and a two-stage VGA, N equals 14.

The DNA attack is executed on the benchmark analog circuits introduced in Section 8.2 and for the three attack scenarios described in Section 8.3. The four analog benchmark circuits are designed and characterized in a TSMC 65nm process that offers a low threshold (LVT), nominal threshold (NVT), and high threshold (HVT) voltage transistor. Therefore, T is set to three for the evaluation of the attack on the benchmark analog circuits. The results of executing the DNA attack on the four analog circuits are listed in Table 10.2. As indicated by the listed results, the attack requires a 10% variation in V_{out} to account for the difference between the threshold voltages of the transistors of the circuit and the nominal threshold voltage defined by the PDK.

The attack returned an average of 17.5 candidate keys in approximately 75.25 seconds for single-stage circuits analyzed considering Scenario I. In addition, the proposed attack eliminated approximately 96% of the search space of single-stage circuits considered under Scenario I. The mixer did not return a correct key for variations in V_{th} and β of 5%. The allowed variation in both parameters was, therefore, increased to 10%, which resulted in the solver returning a correct key. However, the DNA attack returned the correct key of a secured single stage VGA for an allowed variation of 5% in both V_{th} and β . As N is larger for the VGA than the mixer, the total number of candidate keys returned for the VGA was

243, as compared to the 81 returned for the mixer. In addition, the execution time of the attack is longer for the VGA as compared to the mixer. For Scenarios II and III, the attack eliminated approximately 88.5% and 82.5% of the search space in less than 8 minutes and 11 minutes, respectively.

The two-stage VGA obfuscated with multi-threshold obfuscation results in an N equal to 10, which, therefore, provides a total of $3^{10} = 59049$ unique key combinations. The proposed attack eliminated 99.85% of the search space in under 18 minutes when considering Scenario I. Under Scenario II, the attack eliminated 99.62% of the keys from the search space in 35.6 minutes. In addition, the attack eliminated 99.58% of the keys from the search space in 6.5 hours when considering Scenario III.

The three-stage front-end circuit is obfuscated with a 14-bit key. Even for a complex three-stage circuit, the proposed attack eliminated 99.99% of the keys from the search space when considering Scenario I, returning a total of 336 candidate keys. The execution time of the attack under Scenario I is approximately 10.2 hours. In addition, the proposed attack eliminated 99.9886% and 99.984% of the keys from the search space in 21.54 hours and 50.4 hours when executed under Scenarios II and III, respectively.

10.3 Performance Comparison of DC Nodal Analysis

Attack With State-of-the-art Analog Deobfuscation

Attacks

The performance of the proposed DNA attack is compared with state-of-the-art analog deobfuscation attacks that include the equation-based SMT attack, the genetic algorithm (GA) attack, the key-spacing attack, and the monotonic response attack. The metrics for evaluating the performance of the proposed deobfuscation attack are the resource metric (T_R) and the performance metric (T_P) described in Section 9.1. The resource metric accounts for the time required to initialize the attack [220]. In addition, the performance metric accounts for the time to determine the candidate keys from the obfuscated analog circuit and the time to determine the correct key from the candidate keys when executing a brute-force attack [220].

10.3.1 Resource Metric (T_R)

The resource metric accounts for the total time required to collect all the information needed to perform the attack. The resources needed to initialize the DNA attack include the circuit netlist, oracle IC, and DC input and output voltage(s) and current(s). Utilizing Table VI from [220], the resource metric is computed. The number of steps required to

determine the circuit netlist, attack model(s), and specification information are six, three, and one, respectively [220]. The resource metric is computed as the summation of all required steps. For the DNA attack, the resource metric is computed as 10, which is listed in the last row of Table 10.3.

Table 10.3: Evaluation of the resource metric for the five analog deobfuscation attacks. A check mark implies needed information. The number of steps to determine all needed information is listed, where each step represents one unit of time.

Attack Techniques	Required Information					Resource Metrics (Unit-time)
	Netlist	Biasing Information	Models	PDK Information	Circuit Specifications	
SAT [203–205, 221]	✓	✗	✗	✗	✓	7
SMT [201]	✓	✓	✓	✓	✓	17
GA [202]	✓	✓	✗	✓	✓	14
Key Spacing [200]	✓	✗	✗	✗	✗	6
Monotonic Attack [200]	✗	✗	✓	✗	✓	4
DNA Attack (This work)	✓	✗	✓	✗	✓	10

The computed resource metrics for all the state-of-the-art analog deobfuscation techniques is listed in Table 10.3. A comparison of analog deobfuscation attacks to digital deobfuscation is also performed, where the SAT attack is considered the de-facto standard to evaluate the resiliency of digital logic locking techniques [203–205, 221]. The results listed in Table 10.3 indicate that the proposed attack provides a better resource metric score as compared to both the equation-based SMT [201] and GA [202] attacks, where a lower value is preferred. The DNA attack requires seven and four fewer number of steps to initialize as compared to the SMT and GA attacks, respectively. However, the proposed attack results in a higher resource metric score than the key-spacing [190] and monotonic

response [190] attacks, where four and six additional steps are required, respectively. The benefit provided by the DNA attack is the ability to characterize multi-threshold locking techniques as the key-spacing and monotonic response attacks do not deobfuscate analog circuits secured with a non-key based obfuscation methodology.

10.3.2 Performance Metric (T_P) Calculated for Key-based Obfuscation

The performance metric accounts for the time to determine the candidate keys from the obfuscated analog circuit and the time to determine the correct key from the candidate keys when executing a brute-force attack [220]. The performance metric score of the DNA attack is compared to that of the equation-based SMT, GA, key-spacing and monotonic response attacks, with the results from the comparison listed in Table 10.4. The performance metric is evaluated for the five analog obfuscation techniques implemented on a single-stage VGA, two-stage VGA, and three-stage analog front-end, which are described in Section 8.2. The results provided in Section 8.4 are used to determine the number of candidate keys and the execution time of the attacks for the equation-based SMT, GA, key-spacing and monotonic response attacks. In addition, the results listed in Table 10.1 are used to compute the performance metric score of the DNA attack.

The execution of the brute force attack is performed through transient analysis of the

Table 10.4: Evaluation of the performance metric from the execution of the SMT, GA, key-spacing, monotonic response, and DC nodal analysis attacks on a VGA, 2-stage VGA, and front-end. The performance metric score is calculated as the sum of the execution time needed to determine the candidate keys and the total time needed to perform a brute-force attack on the returned candidate keys, which is computed as the product of the number of candidate keys and the time required to perform a transient simulation on a single candidate key. The “-” symbol indicates the failure of the deobfuscation algorithm in returning the correct key.

Deobfuscation Attacks	Scenarios	VGA (10-bit key)			2-Stage VGA (12-bit key)			3-Stage Front-End (18-bit key)		
		Ex. Time(s)	Candidate Keys	Perf. Metrics	Ex. Time(s)	Candidate Keys	Perf. Metrics	Ex. Time(s)	Candidate Keys	Perf. Metrics
Equation-based SMT [201]	I	6.30	1009	3436.9	43.60	3969	25842.1	2818.25	42948	352844.45
	II	6.74	1009	3437.34	51.07	3969	25849.57	48060	250047	2085943.05
	III	3.75	511	1741.15	92.63	3969	25891.13	52812	250047	2090695.05
GA [202]	I	26640	1	26643.4	23652	-	-	432000	-	-
	II	26640	1	26643.4	23652	-	-	411480	-	-
	III	47880	1	47883.4	23652	-	-	411480	-	-
Key Spacing [200]	I	1.93	19	66.53	0.89	27	176.39	1.25	36	294.65
	II	1.93	19	66.53	0.89	27	176.39	1.25	36	294.65
	III	1.93	19	66.53	0.89	27	176.39	1.25	36	294.65
Monotonic Response [200]	I	0.41	16	54.81	8.91	373	2433.41	3051.84	9312	78944.64
	II	0.46	16	54.86	2.8	119	776.3	856.53	8465	69846.28
	III	0.45	16	54.85	9.46	372	2427.46	198.64	3538	29033.34
DNA (This work)	I	518.59	23	596.79	181080.32	668	185422.32	432000	711	437794.65
	II	234.17	53	578.676	266410.32	987	272825.82	432000	1189	441690.35
	III	2368.38	711	4785.78	296287.56	1489	305966.06	432000	2467	452106.05

analog circuit, with each candidate key applied to the circuit in *Cadence schematic editor* and analyzed with *Spectre* simulation. The time required to perform each transient simulation on the server for each candidate key applied to a single-stage, two-stage, and multi-stage circuit is 3.4, 6.5, and 8.15 seconds, respectively. The time to input the candidate key on the obfuscated analog circuit in the schematic editor is ignored. Only the time to execute a 10 μ s transient simulation on the schematic view of the obfuscated analog circuit using the *Spectre* simulator is considered when computing the performance metric. The performance metric is, therefore, calculated as the sum of the attack execution time and the product between the number of candidate keys and the time required to perform each transient simulation on a candidate key as given by

$$PM = T_{execution} + K_{candidate} \cdot T_{simulation}, \quad (10.1)$$

where, $T_{execution}$ is the time required to perform the deobfuscation attack, $K_{candidate}$ is the total number of candidate keys returned from the execution of the deobfuscation attack, and $T_{simulation}$ is the time required to perform transient simulation on one candidate key.

10.3.2.a Comparison with Equation-based SMT Attack

Compared to the equation-based SMT attack, the DNA attack returns on average, 95.8% fewer candidate keys when executed on a single-stage VGA when considering Scenarios I and II. However, the DNA attack returned 28.1% more candidate keys for Scenario III. For the evaluated performance metric, the DNA attack provides, on average, an 82.9% lower score as compared to the equation-based SMT attack, where a lower score is desired.

For the two-stage VGA, the equation-based SMT attack eliminated only 127 keys, as compared to the 3428 keys eliminated by the DNA attack when considering Scenario I. Even for Scenario II, where the attacker does not possess the PDK information, the DNA attack eliminated 3109 keys, as compared to the 127 keys eliminated by the equation-based SMT attack. However, the DNA attack required 5153x more time to execute as compared to the equation-based SMT attack, resulting in a 10.55x higher performance metric score.

For the three-stage front-end circuit, although the DNA attack timed-out after passing

the set five day execution time limit, the number of candidate keys returned was 60.4x less than that of the equation-based SMT attack when considering Scenario I. For Scenarios II and III, the evaluated performance metric score of the DNA attack is, respectively, 4.7x and 4.6x lower than that of the equation-based SMT attack. In addition, as more information is made unavailable to the attacker (Scenario III), the number of candidate keys returned by the equation-based SMT attack increases, which results in an increase in the time to perform a brute-force attack on the returned candidate keys to determine the correct key. Therefore, for the three-stage front-end circuit, the equation-based SMT attack results in a higher performance metric score as compared to the DNA attack. The brute force attack is executed as a 10 μ s transient analysis of the analog circuit, applying each candidate key in the *Cadence* and *Spectre* simulation suite, which does not fully represent true execution of the attack, but provides a simplified basis for comparison. A true execution of the attack requires loading the keys on to an oracle IC before performing the attack on the oracle IC, which increases the time to execute the brute-force attack, and, therefore, results in a higher performance metric score. In addition, the execution of the brute-force attack is performed on a schematic view, which results in a faster execution time while trading-off accuracy. To improve the accuracy of the circuit simulation, the brute-force attack is executed on the extracted view, which results in additional computational time to both perform the parasitic extraction of the circuit and to complete the transient analysis of each iteration of the brute-force attack. The increase in computational time correlates to

a higher performance metric score.

10.3.2.b Comparison with Genetic Algorithm Attack

The results from the evaluation of the performance metric on the GA attack executed on three different analog circuits and for the three attack scenarios described in Section 8.3 are obtained from [220]. The results indicate that the GA attack successfully returns the one candidate key, which is the target key, only for a single stage circuit attacked under Scenarios I, II, and III. For the single stage VGA obfuscated by a 10-bit key, the performance metric score of the GA attack is approximately 45x higher than that of the DNA attack executed considering Scenarios I and II. The higher performance metric score is due to the longer execution time of the attack, where approximately 7.4 hours were required to determine the candidate key of the single-stage VGA attacked assuming the ideal conditions described by Scenario I. For the two-stage VGA and the three-stage front-end circuit, the GA attack does not return the correct key, even after an equal number of iterations are performed to that of a brute-force attack [220]. In addition, the execution time of the GA attack is the longest of the five deobfuscation techniques when executed on a single-stage circuit, as the attack is dependent on results from *Spectre* simulations to optimize a cost function.

10.3.2.c Comparison with Key Spacing Attack

The results listed in Table 10.4 indicate that the key-spacing attack is the fastest and returns the least number of candidate keys when executed on multi-stage analog circuits considered under all three attack scenarios, including as compared to the DNA attack. As the key spacing attack only requires the netlist of the extracted circuit, the returned keys are the same for Scenarios I, II, and III. The key-spacing attack results in a 71.9x, 1734.6x, and 1534.5x lower performance metric score for a single-stage VGA, two-stage VGA, and a three-stage front-end circuit under Scenario III, as compared to the DNA attack. However, unlike the DNA attack, the key-spacing attack is only applicable to key-based analog obfuscation techniques and is, therefore, not applicable to multi-threshold obfuscation [193].

10.3.2.d Comparison with Monotonic Response Attack

The results listed in Table 10.4 indicate that for a single-stage circuit obfuscated with a 10-bit key, the monotonic attack is the fastest and returns the least number of candidate keys. The monotonic response attack, therefore, provides the lowest performance metric score, where a lower score is desired. The monotonic response attack is performed on generic circuit equations based on the extracted netlist that are used to model the obfuscated analog circuit. The relatively fast execution time of the monotonic attack is a result of using generic analog equations that ignore both the transistor operating region and

the non-linear parameter characteristics of the devices when developing equation based models of the circuit.

For the single-stage VGA, analysis of the monotonic response attack indicates a 87.3x lower performance metric score under Scenario III. For multi-stage obfuscated analog circuits, where the circuit equations result in a non-monotonic circuit response, the number of candidate keys returned by the monotonic response attack increases. For the three-stage analog front-end circuit considered under Scenarios I, II, and III, the monotonic response attack returned, respectively, 8601, 7276, and 1071 more candidate keys than the DNA attack. However, the longer execution time of the DNA attack resulted in a higher performance metric score, where the DNA attack produces 5.5x, 6.3x, and 15.57x higher score than the monotonic response attack for Scenarios I, II, and III, respectively.

10.3.3 Performance Metric (T_P) Calculated for Multi-threshold Obfuscation

The performance metric score of the DNA attack is compared to that of the equation-based SMT, GA, key-spacing, and monotonic response attacks for non-key-based multi-threshold obfuscation, with the results as listed in Table 10.5. The performance metric is evaluated on two analog obfuscation techniques implemented on the single-stage VGA, two-stage VGA, and three-stage analog front-end circuits described in Section 8.2. The results listed in Table 10.2 are used to compute the performance metric score of the DNA

Table 10.5: Evaluation of the performance metric from execution of the SMT, GA, key-spacing, monotonic response, and DC nodal analysis attacks on a VGA, 2-stage VGA, and front-end secured with multi-threshold obfuscation. The “*” symbol indicates that the equation-based SMT attack did not determine the threshold voltages of all the transistors of the circuit and the “-” symbol indicates the failure of the deobfuscation algorithm in returning the correct key.

Deobfuscation Attacks	Scenarios	VGA (15-bit key)			2-Stage VGA (30-bit key)			Front-End (42-bit key)		
		Ex. Time(s)	Candidate Keys	Perf. Metrics	Ex. Time(s)	Candidate Keys	Perf. Metrics	Ex. Time(s)	Candidate Keys	Perf. Metrics
Equation-based SMT [201]	I	0.20	7*	24*	10.32	27	102.12*	71.54	65*	292.54*
	II	0.33	48*	163.53*	15.54	621	2126.94*	2820.33	463*	5829.83*
	III	0.47	62*	211.27*	47.37	728	2522.27*	17468.16	2186*	35284.06*
GA [202]	I	22078.1	1	22081.5	192627.38	1	192633.88	431982.75	-	-
	II	22078.1	1	22081.5	192627.38	1	192633.88	431982.75	-	-
	III	89416.31	1	89416.31	246655.53	1	246655.53	431982.75	-	-
Key Spacing [200]	I	Cannot Perform Attack								
	II									
	III									
Monotonic Response [200]	I	387.66	17	445.46	483.52	-	-	-	-	-
	II	427.68	19	551.18	483.52	-	-	-	-	-
	III	-	-	-	-	-	-	-	-	-
DNA (This work)	I	85.19	31	190.59	1550.34	91	2141.84	47864.24	336	50602.64
	II	285.31	44	434.91	2136.13	220	3566.13	77544.43	543	81969.88
	III	322.63	73	570.83	23418.3	243	24997.8	181440.35	721	187316.5

attack.

10.3.3.a Comparison with Equation-based SMT Attack

The equation-based SMT attack proposed in [201] requires generation of an equation based model of the obfuscated analog circuit netlist based on prior knowledge of the circuit netlist, process design kit (PDK) documentation, circuit specifications, and the input/output response pairs of the oracle IC. Unlike the key-based locking technique, where the devices selected for obfuscation are used to develop a circuit equation that models the locked parameter and corresponding circuit response, for multi-threshold obfuscation, all transistors in the circuit are assumed obfuscated. Therefore, constructing equation(s) that model all the obfuscated transistors in the circuit to the corresponding output response of

the circuit is extremely challenging. As an example, for the single-stage VGA described in Section 8.2.1, where transistor M_3 is selected for key-based parameter obfuscation, the general relation between the obfuscated transistor parameter and the gain (circuit response) is given by

$$A_v = \frac{g_{m_{1,2}}}{g_{m_{4,5}} + g_{m_{6,7}}}, \quad (10.2)$$

where A_v is the gain of the VGA and g_m is the transconductance of the transistor. Further, the transconductance of a transistor operating in the saturation region is given by

$$g_m = \sqrt{2 \cdot \mu C_{ox} \cdot \frac{W}{L} \cdot I_{ds}} = \frac{2 \cdot I_{ds}}{|V_{gs} - V_{th}|}, \quad (10.3)$$

while the transconductance of a transistor operating in the sub-threshold region is given by

$$g_m = \frac{2 \cdot I_{ds}}{n \cdot V_{th}}, \quad (10.4)$$

where μ represents the mobility of electrons or holes of, respectively, the NMOS or PMOS transistor, C_{ox} represents the gate oxide capacitance, V_{th} denotes the threshold voltage of the transistor, n denotes the sub-threshold slope, V_{gs} represents the gate to source voltage, and I_{ds} denotes the drain to source current of the transistor. In addition, to include the transistor M_3 in the gain equation of the VGA, the attacker must assume that the current

through M_3 is twice the current through M_1 , where I_{ds3} equals the sum of I_{ds1} and I_{ds2} .

Substituting, (10.3) and (10.4) in (10.2), the gain of the VGA is given by

$$A_v = \frac{\sqrt{2 \cdot \mu_n C_{ox} \cdot \left(\frac{W}{L}\right)_{1,2} \cdot \frac{I_{ds3}}{2}}}{\frac{2 \cdot I_{ds4,5}}{n \cdot V_{th4,5}} + \sqrt{2 \cdot \mu_n C_{ox} \cdot \left(\frac{W}{L}\right)_{6,7} \cdot I_{ds6,7}}}, \quad (10.5)$$

where,

$$I_{ds_i} = \frac{1}{2} \cdot \mu_n C_{ox} \cdot \left(\frac{W}{L}\right)_i \cdot (V_{gs_i} - V_{th_i})^2 \quad (10.6)$$

for the i_{th} transistor in the circuit.

From (10.5), only the threshold voltages of transistors M_3 , $M_{4,5}$, and $M_{6,7}$ are considered when calculating the gain of the VGA. The threshold voltage of the transistors M_1 and M_2 are indirectly determined, where I_{ds1} and I_{ds2} are assumed to be equal to $\frac{I_{ds3}}{2}$ due to the symmetrical differential pair formed by transistors M_1 and M_2 . Utilizing the square-law model of the transistor, the threshold voltages of M_1 and M_2 are determined assuming that V_{gs1} and V_{gs2} are available to the adversary, which requires significant effort to extract. In addition, the gain of the VGA is not directly impacted by the threshold voltages of transistors M_8 and M_9 as V_{th8} and V_{th9} are not considered in (10.5). Therefore, the equation-based SMT attack is not able to determine the threshold voltage V_T of transistors M_8 and M_9 . The example highlights the primary drawback of executing the equation-based SMT attack on circuits secured by multi-threshold obfuscation, as only the threshold voltage of transistors that directly affect the output response of the circuit

are determined by the attack.

The results of executing the equation-based SMT attack on a single-stage VGA, two-stage VGA, and three-stage front-end circuit secured with multi-threshold obfuscation are listed in Table 10.5. Utilizing the circuit response equations, only the threshold voltages of transistors M_3 , M_4 , M_5 , M_6 , and M_7 are determined for the VGA. In addition, utilizing the equations for the gain and the oscillating frequency of the mixer, only the threshold voltages of transistors M_1 , M_2 , and M_3 are determined. As only a subset of the threshold voltages of the transistors in the circuit are determined by the equation-based SMT attack, the number of candidate keys returned and the calculated performance metric score for the attack are marked by “*” in Table 10.5. The results indicate that the equation-based SMT attack returns a smaller number of candidate keys and is faster to execute as compared to the DNA attack for a single-stage VGA, which results in a 3.96x lower performance metric score on average. For the front-end circuit, the DNA attack returned a smaller number of candidate keys for Scenarios II and III. However, the execution time of the equation-based SMT attack is 19x lower, which results in an average 9.7x lower performance metric score as compared to the DNA attack.

10.3.3.b Comparison with Genetic Algorithm Attack

For the GA attack, the cost function is formulated such that an applied key is only considered a possible candidate if there is a 5% or less error between the target and simulated

performance values. The probability of mutation is set to 20% and the number of generations and the population size are each initially set to five. Both the number of generations and the population size are then incremented by five until the GA attack produces the same key for three consecutive random initial chromosome values. For the analysis in this thesis, the GA attack described in [202] is modified to utilize *Cadence* and *Spectre* instead of *PySpice* for modeling, characterization, and analysis of the circuit response. Since the attack utilizes *Cadence* and *Spectre*, the PDK parameters including μC_{ox} , T_{ox} , and V_{th} are already assumed available. Therefore, the results from analysis of Scenario I and II are the same. For Scenario III, the attacker must also determine the biasing voltages and currents of the circuit in addition to the obfuscating key, where all unknowns are mapped to a single chromosome.

The results of executing the GA attack algorithm on the three analog circuits secured using multi-threshold obfuscation are listed in Table 10.5. The results indicate that even though the GA attack returns the correct key used to secure the single-stage and two-stage VGA, the execution time of the GA attack is on average 138x longer than that of the DNA attack considered under Scenario I. As the required chromosome size to determine both the threshold voltages and the biasing voltages is larger as compared to Scenarios I and II, the population sizes and the number of generations is also larger for Scenario III, which results in a longer execution time. Therefore, the computed performance metric score of the GA attack considered under Scenario III for the single-stage and two-stage

VGA is on average 83x higher than that of the DNA attack. In addition, executing the GA attack on the front-end circuit failed to determine the correct key as the GA attack timed out after the algorithm reached the maximum set population size of 90 and the maximum number of generations of 85.

10.3.3.c Comparison with Key Spacing Attack

As the key-spacing attack [200] is developed to complete only on analog circuits obfuscated with key-based obfuscation, the attack is not able to execute on multi-threshold obfuscated circuits.

10.3.3.d Comparison with Monotonic Response Attack

For the monotonic attack (MA), an adversary is assumed to have access to the input-output responses of the obfuscated circuit. The attack algorithm utilizes the monotonic relationship between the given set of threshold voltages and the output response of the circuit. The results of executing the monotonic attack on the three analog test circuits obfuscated using multi-threshold obfuscation and for the different attack scenarios described in Section 8.3 are listed in Table 10.5. The attack was only able to determine the correct key of the single-stage VGA for Scenarios I and II. As more information is made unavailable to an adversary, the relationship between the obfuscated threshold voltages and the response of the circuit becomes non-monotonic. Therefore, the monotonic attack failed to determine the correct key for multi-stage circuits and for Scenario III of the single

stage VGA.

The results listed in Table 10.5 highlight the capability of the DNA attack in determining the correct threshold voltages of all transistors of both single-stage and multi-stage analog circuits secured with multi-threshold obfuscation. The proposed attack determined an average of 50 candidate keys for a single-stage VGA with an execution time of less than 10 minutes. In addition, the proposed attack eliminated 99.984% of the search space in under 51 hours for the multi-staged front-end circuit considered under Scenario III. In comparison, the current state-of-the-art analog deobfuscation techniques either failed to execute (key-spacing attack), are not capable of executing on multi-stage circuits (monotonic response attack), resulted in a much higher performance metric score (GA attack), or were not able to determine the threshold voltages of all transistors of the analog circuit (equation-based SMT attack).

10.4 Discussions

The primary objective of the DC nodal analysis attack is to quantify the security provided by analog obfuscation techniques. The obfuscation techniques are divided into two main categories: key-based and non-key-based. The key-based techniques include key-based parameter obfuscation [222] and current-mirror obfuscation [192]. In addition, key-based techniques including the obfuscation of post-silicon biasing points [113] and

analog-to-digital converters [114] are only applicable to mixed signal circuits. The non-key-based analog obfuscation techniques include split-manufacturing [101] and camouflaging using multi-threshold transistors [193].

Since the current-mirror obfuscation proposed in [192] is a subset of key-based parameter obfuscation, the execution of the DC nodal analysis attack on the current-mirror locking technique is just as effective in determining the keys of the secured circuit. The ability to execute the DC nodal analysis attack under Scenario III, where the adversary does not possess threshold voltage and bias voltage/current information, highlights the capability of the proposed technique as compared to the other attack algorithms. Both the threshold voltages and the bias voltages/currents are the key components used in post-silicon tuning [198]. Therefore, the proposed DC nodal analysis attack bypasses the obfuscated biasing tuning circuitry to determine the target un-tuned threshold voltages and biasing voltages/currents that result in the correct functionality of the analog circuit. The adversary is able to utilize the developed bias voltages/currents and threshold voltages with redesigned compensation circuitry.

10.5 Conclusions

A novel DC nodal analysis attack is proposed that utilizes the KCL and KVL laws to efficiently explore the obfuscated design space and determine the correct control parameters of a secured analog circuit. A preliminary evaluation of the proposed attack

is performed on three distinct analog circuits obfuscated using both key-based parameter obfuscation and non-key-based multi-threshold obfuscation. In addition, the attack is also evaluated on different real-world threat scenarios, where one or more pieces of information is made unavailable to the attacker. The proposed DNA attack effectively eliminated an average of 96% of keys from the search space of single-stage circuits obfuscated with key-based parameter obfuscation when considering Scenarios I and II. For the two-stage and three-stage circuits, the attack eliminated 75% and 99.4% of the key space, respectively, across all three scenarios. For multi-threshold obfuscation, the attack eliminated 99.98% of the search space of possible threshold voltages in less than 51 hours when considering a three-stage circuit under Scenario III. The comparison of the computed resource metric score for the DNA attack with current state-of-the-art deobfuscation attacks indicates that the DNA attack requires seven and four fewer number of steps to initialize as compared to, respectively, the SMT and GA attacks. The analysis of the performance metric score evaluated for analog circuits secured with key-based obfuscation indicates that the DNA attack resulted in a fewer number of returned candidate keys as compared to equation-based SMT and a faster execution time as compared to GA attacks. However, the DNA attack produced a 1534.5x and 15.57x higher performance metric score as compared to the key-spacing and monotonic response attacks, respectively. Note that, unlike the DNA attack, the key-spacing attack is not applicable to non-key-based techniques

including multi-threshold obfuscation. In addition, unlike the DNA attack, the monotonic response attack failed to determine the correct key for multi-stage analog circuits secured with multi-threshold obfuscation. The results indicate that the DNA attack is capable of determining the correct secured parameters of a circuit obfuscated with key and non-key-based locking techniques. In addition, the utilization of the DNA attack for different real-world scenarios validate the applicability of the proposed attack as a standard to effectively evaluate the strength of analog obfuscation techniques.

Chapter 11

Concluding Remarks

The increasing dependence on third-party hardware components and the continued prevalence of intellectual property (IP) reuse have brought attention to the importance of secure hardware. Prior research on hardware IP protection primarily focused on securing digital circuits. However, analog circuits are vulnerable to the same threats, including IP piracy, reverse engineering, and supply chain attacks. The work described in this dissertation enhances the security of analog circuits through the completion of three primary research objectives: 1) the development of circuit based locking techniques to protect analog integrated circuits (ICs) against IP piracy attacks, 2) the assessment of the limitations of existing analog deobfuscation algorithms in the context of considered attack scenarios and analog locking techniques, and 3) the development of an SMT-based methodology that establishes a benchmark for assessing the efficacy of an analog locking technique.

The first research objective, the development of parameter obfuscation, a novel technique to protect analog circuits against IP piracy attacks that include reverse engineering, counterfeiting, and overproduction, is described in Chapter 4. Two techniques, vector and mesh transistor obfuscation, are introduced, which provide a varying degree of security to an analog circuit. The two obfuscation techniques are implemented to mask the gain and center frequency of a band pass filter (BPF) with a 12-bit key and the gain and gain bandwidth of an op-amp with a 10-bit key. The parameter obfuscation on the BPF resulted in a maximum deviation of 2.3% in the center frequency, no loss in the gain, and a 0.024% probability of determining the correct key when executing a brute-force attack. The implementation of parameter obfuscation on the op-amp resulted in a maximum deviation of 0.4 dB in the gain, 0.3 MHz in the unity gain bandwidth, and a probability of 0.098% of determining the correct key with a brute-force attack. A simulation-based optimization technique is described in Chapter 5, which minimizes the effects of parasitic impedance and process variations due to the implementation of the parameter obfuscation techniques by tuning the bias voltages, bias currents, and the body bias of select transistors of the circuit. The simulation-based optimization algorithm is iteratively executed 500 times for the BPF and 70 times for the op-amp to determine the optimal biasing conditions and body bias voltages that minimize the effects of parasitics due to the obfuscation techniques. The results of executing the algorithm in the design of the BPF indicate that no more than a 2.75 dB difference in the gain, 4.75% difference in the center frequency, and

10 MHz difference in the 3-dB bandwidth of the obfuscated BPF is observed as compared to the unobfuscated BPF across all five fabrication process corners (TT, FS, SF, SS, and FF). The simulation-based optimization algorithm efficiently reduces design time when implementing parameter obfuscation on an analog circuit.

The second research objective is the evaluation of existing state-of-the-art analog de-obfuscation techniques considered under real-world attack scenarios and for various analog obfuscation techniques. In prior work, analog deobfuscation algorithms, including the equation-based SMT, genetic algorithm, key-spacing attack, and the monotonic response attack, are developed to characterize the vulnerabilities of analog obfuscation techniques and to evaluate the resiliency of locking techniques to various attacks. In Chapter 8 and Chapter 9, the performance of the analog deobfuscation techniques is analyzed for scenarios considering a varying amount of information available to the adversary. The analysis indicates that the SMT attack pruned only 1.5% of the 10-bit keys obfuscating a single stage mixer. In addition, as compared to the monotonic response and the key spacing attacks, the SMT attack is slower by 224x and 2240x, respectively. The GA attack determined the correct key for the single stage circuits in less than 8 hours, but failed to determine the correct key for multi-stage circuits. The execution time of the GA attack is 121091x slower than that of the monotonic attack for a single stage analog circuit. The key-spacing attack is efficient in pruning keys, with an execution time that is 10x faster than that of the monotonic attack, and returns 8.3x fewer candidate keys for multi-stage analog circuits, which

the adversary must then brute-force attack to determine the correct key. The monotonic attack determined the correct key in less than one second for single stage circuits, less than 10 seconds for two-stage circuits, and less than an hour for three-stage circuits. As the number of stages increases, the circuit response typically becomes non-monotonic, which results in the failure of determining the correct key by both the GA attack and the monotonic response attack. Metrics, which the resources needed to initialize the attack and the time to execute the attack are developed to assess the performance of the analog deobfuscation techniques, as described in Chapter 9. The evaluation of the metrics indicates that the monotonic attack is the fastest to deobfuscate single-stage analog circuitry, where the attack determined the correct key in less than 10 hours. The key-spacing attack is the fastest and cheapest to execute when deobfuscating multi-stage analog circuits, but requires the circuit netlist. The SMT attack executes faster than the key-spacing attack for single stage analog circuits attacked under Scenario I, where the attacker possesses all the circuit information except for the correct key. However, the completion time and the number of candidate keys returned increase exponentially when executing the SMT attack on an obfuscated multi-stage analog circuit. The execution time of the genetic algorithm (GA) is the longest, where the attack timed out after 5 days without determining the correct key of a locked 2-stage VGA and a front-end receiver even when considering Scenario I.

The third research objective, the development of the DC-based nodal analysis (DNA);

attack provides a measure of the resilience of state-of-the-art analog obfuscation techniques to real-world attack scenarios. The results of evaluating the current analog deobfuscation attacks across four different attack scenarios as described in Chapter 8 and Chapter 9 indicate that the current deobfuscation techniques are inefficient when considered under real threat scenarios and are not applicable to all analog obfuscation techniques or to multi-stage obfuscated circuits. Therefore, unlike the digital circuit domain, where the satisfiability (SAT) attack is utilized as one de-facto standard for measuring the security resiliency of key-based logic locking techniques, no standard measure of the security resiliency exists in analog circuit domain. A DC nodal analysis (DNA) attack algorithm is, therefore, developed as described in Chapter 10 to evaluate the security provided by different analog obfuscation techniques. The DNA attack requires knowledge of only the circuit netlist and the DC input-output response of an oracle IC. In addition, the DNA attack is applicable to all current analog obfuscation techniques, where the attack effectively eliminated an average of 96% of keys from the search space of single-stage circuits obfuscated with key-based parameter obfuscation. For two-stage and three-stage circuits, the attack eliminated 75% and 99.4% of the key space, respectively, across scenarios where the attacker is assumed to not possess the biasing information and PDK documentation while attempting to determine the correct key. For multi-threshold obfuscation, the DNA attack eliminated 99.98% of the search space of possible threshold voltages to assign to transistors in less than 51 hours when executed on a three-stage circuit. The analysis of

the performance metric score evaluated for analog circuits secured with key-based obfuscation indicates that the DNA attack resulted in approximately 53x fewer returned candidate keys as compared to the equation-based SMT attack across all threat scenarios and provided an approximately 33.56x faster execution time as compared to the GA attack considered under Attack Scenario I. The DNA attack resulted in a 1534.5x and 15.57x higher performance metric score as compared to the key-spacing and monotonic response attacks, respectively. However, unlike the DNA attack, the key-spacing attack is not applicable to non-key-based techniques including multi-threshold obfuscation. In addition, and again unlike the DNA attack, the monotonic response attack failed to determine the correct key for multi-stage analog circuits secured with multi-threshold obfuscation. The results indicate that the DNA attack is capable of determining the correct parameters of a secured analog circuit obfuscated with key and non-key-based locking techniques considered under real-world scenarios. The utilization of the DC nodal analysis attack, therefore, allows for the general evaluation of the security provided by analog obfuscation techniques.

Chapter 12

Future Work

In recent years, concerns surrounding vulnerabilities within the analog IC supply chain have prompted the development of novel security measures, including the parameter obfuscation technique described in Chapter 4, that aim to safeguard analog integrated circuits from intellectual property attacks. The challenge of designing an analog circuit, as discussed in Chapter 1, is that approximately 40% of the overall IC design time and 50% of design cycles are directly attributed to the analog components [10]. Incorporating security measures after designing an analog circuit introduces additional complexities that result in increased cost and extended timelines due to redesigns, and therefore, necessitate a seamless implementation of security measures early in the design of an analog circuit. Currently, knowledge of techniques and methodologies to enhance the security of an analog circuit only exist among hardware security experts, while knowledge of efficiently

implementing analog circuits resides within the analog design community. The divisions are further exacerbated by the complexities and rapid innovations in both domains, which results in instances where state-of-the-art security practices are not consistently implemented in cutting-edge analog hardware. To address the divide, research towards automated security-aware analog circuit design is needed. By leveraging automated design processes, the approach holds promise in bridging the existing gap and ensuring that security considerations are an integral part of an analog circuit from the onset. The three primary components of the proposed security-aware analog design methodology include: 1) integrating security as a metric in addition to performance, area, power, and noise during the design phase of an analog circuit, 2) automating the security-driven analog design flow methodology, and 3) developing secure memory for key storage and key delivery as an additional feature.

12.1 Security Metric Model for Analog Circuit

Every designed circuit utilizes metrics for trade-offs between desired and undesired properties. In the context of analog integrated circuits, properties typically include area, performance, power, gain bandwidth, linearity, and noise. Security is not currently considered as a metric during the design of an analog circuit, which results in design recycles when implementing security features. In addition, the development of security metrics for analog design poses a unique challenge as the threat model is not static. Specifically,

an adversary may possess a varying degree of information pertaining to the analog circuit and attack with varying intentions to compromise the circuit. The complexity of the attack space implies that security metrics vary depending on the information that an adversary possesses and the attack objective. To address the dynamic nature of analog security, a general metric model is needed.

The primary objective of developing a security metric model is to provide traditional analog designers with 1) information on the potential security vulnerabilities of existing analog circuit designs and 2) knowledge of the best obfuscation technique suited for a particular analog circuit topology and security objective, without requiring the designer to have a background in security. The proposed security metric, therefore, aims to bridge the gap between the design and security domains, which enables the use of the best security techniques in state-of-the-art analog IP. The primary tasks to develop the proposed security metric model for analog circuits include

1. Creation of a database consisting of all the state-of-the-art analog obfuscation techniques. The proposed database will be expandable as novel analog obfuscation techniques are developed.
2. Based on the target application of the analog IP, determine the best and worst case threat scenarios to consider. Selecting the proper scenarios is needed to accurately calculate the security metric score of an obfuscated analog circuit.

3. Determine the best region of an analog circuit to implement an obfuscation technique. A further expansion on the selection and placement methodology described in Section 4.3 is needed to allow for implementation of novel state-of-the-art analog obfuscation techniques.
4. Automate the calculation of the analog security metric score for all applicable analog obfuscation techniques based on the resource and performance metrics introduced in Chapter 9. To accurately determine the resource metric, which includes the total number of steps the attacker must perform to successfully execute a deobfuscation attack, research is needed to quantify the time and cost to extract circuit netlist, bias voltages and/or currents, and process design kit (PDK) information.
5. Develop a methodology to quantify the additional overhead in area, power consumption, and circuit performance when implementing an obfuscation technique. One research direction is the use of graph neural networks (GNNs), where a recent study [223] utilizes GNNs to estimate the parasitic impedance of a circuit. Research in developing a similar methodology that leverages GNNs to estimate the area and power costs of an implemented analog obfuscation technique will provide benefit.

The security metric score is calculated for all existing analog deobfuscation attacks to determine the best suited obfuscation technique for the considered threat scenario. The evaluation of the security metric returns a list of scores for each analyzed state-of-the-art analog obfuscation technique considered under the best and worst case threat

scenarios. The analog obfuscation technique that provides the highest security metric score is implemented to protect the analog circuit. The results of executing the DC nodal analysis attack (DNA) on different analog circuits secured with various locking methods, as described in Section 10.2, indicate that deobfuscation methodologies are capable of determining the correct secured parameters of a circuit obfuscated with either key or non-key-based locking techniques. Therefore, the DC nodal analysis attack is utilized to determine the security metric score of obfuscated circuits.

However, the execution time of the current DC nodal analysis attack (DNA) is long for multi-stage obfuscated analog circuits, where, as the number of transistors in the analog circuit increases, the key space searched by the DNA attack increases exponentially. For example, when considering a single-stage VGA, the DNA algorithm must search through approximately $3.125 \cdot 10^{13}$ combinations of circuit parameters, which include 2^{10} obfuscated key combinations for a 10-bit key, 5^5 threshold voltage combinations (with 10% variation in V_{th} and a step size of 20 mV), and 5^{10} transistor β combinations (with 10% variation in β and a step size of $10 \frac{\mu A}{V^2}$). Similarly, for the two-stage VGA obfuscated with a 12-bit key, the total number of combinations the DNA algorithm must search is $4.096 \cdot 10^{18}$, which results in an approximately 350x longer execution time. Therefore, further research is needed to address the longer execution time of the DNA attack as circuit parameters increase. One technique to reduce the long execution time of the DNA attack is to utilize side channel analysis to determine the current through the individual analog sub-blocks

of the circuit, which constricts the parameter space that the SMT solver must search.

12.2 Security Integrated Analog Design Methodology

Recent research on the automation of analog circuit design represents a growing trend in the field of electronics [224]. Traditionally, the design and optimization of analog circuits relied heavily on manual input, requiring substantial expertise and intuition from experienced engineers. However, the current advancements in machine learning (ML) algorithms and optimization techniques has resulted in the pursuit of novel automated tools for analog circuit design. The objective of utilizing ML algorithms is not just to expedite the design process but to also explore novel circuit configurations that are challenging for human designers to conceptualize. The shift towards automation aligns with the broader design goal of achieving more efficient and precise analog circuit, topologies, while improving time to market.

An initial step towards an automated analog design methodology is the development of a transistor sizing algorithm that accounts for the selected circuit topology. Current analog sizing tools do not consider the effects of implementing security features. Additional iterations of the design flow are needed to properly size transistors when accounting for the increase in the parasitic impedance that attenuates the performance parameters of a circuit as a result of implementing a security feature. Therefore, there is a need to develop an automated analog sizing methodology that simultaneously accounts for the

parasitic impedances that result from the implementation of obfuscation circuitry and the performance specifications of the circuit. A successful implementation of a security-aware analog design methodology includes:

1. Determining the optimal key size and number of transistors obfuscated in the analog circuit for key-based and non-key-based obfuscation techniques. For key-based obfuscation, the optimal key size is determined based on area, power consumption, circuit performance, and security metric scores. For non-key-based obfuscation including multi-threshold camouflaging, the number of nominal threshold, high threshold, and low threshold transistors is determined based on the power consumption, circuit performance, and security metric scores.
2. For key-based obfuscation, determining the number of target and obfuscated transistor sizes.

The SMT-based analog transistor sizing algorithm discussed in Section 3.3.1 and the SMT transistor sizing algorithm for key-based parameter obfuscation described in Section 4.4 provide a strong foundation for the development of the security-aware analog design methodology. Currently, the analog satisfiability (aSAT) algorithm determines the transistor sizes based on the circuit specifications and generic analog circuit equations. However, for a more complex multi-stage analog circuit, the use of generic equations results in lower accuracy. Therefore, further research is needed into developing accurate models of the analog circuit, where area, power consumption, circuit performance, and

security metric scores are considered as constraints. A promising research direction is the use of deep reinforcement learning techniques for analog circuit sizing as described in [225].

The SMT transistor sizing algorithm described in Section 4.4 is only applicable to key-based analog locking, and, therefore, does not apply to non-key-based techniques including multi-threshold obfuscation. In addition, the current multi-threshold obfuscation proposed in [193] suffers from multiple combinations of transistor threshold voltages that result in a correct output, which undermines the security provided by the technique. Therefore, further research is needed that enhances the SMT-based transistor sizing algorithm described in Section 4.4 to 1) determine the correct combination of threshold voltages for the obfuscated transistors, 2) determine the size of the obfuscated transistors after determining and setting the threshold voltage of the transistors while ensuring the correct circuit functionality, and 3) ensure only one combination of transistors with set threshold voltages results in the correct circuit functionality and any incorrect combination results in severe degradation in the circuit performance parameters.

12.3 Developing Secure Memory For Key-based Analog Obfuscation

Key-base analog locking methodologies depend on the storage of a key, either off-chip or on-chip. Implementing off-chip memory, where the key information is transmitted from an external location to an on-chip location, introduces vulnerabilities that lead to the potential leakage of the key. Consequently, many techniques rely on on-chip memory for secure key storage. However, there is a lack of information regarding the design and validation of the security of such on-chip memory. In addition, a study [226] utilized probing attacks on registers and minimal mask modification attacks to extract key bits, bypassing any provable security provided against oracle or structural attacks. Therefore, future research is needed that explores methods to securely store a key on-chip that withstand various key retrieval attack vectors. The development of secure memory necessitates extensive physical validation and logical domain modeling to ensure that the keys are not propagated to any accessible locations within the integrated circuit.

Another limitation of key-based analog locking methodologies is the utilization of the same internal key for all fabricated dies of the same analog IP block. When an identical key is employed across all dies, discovering the key for one allows the attacker to determine the key for all instances of an analog circuit. Therefore, there is a critical need to

develop and implement a unique key for each individual analog die. Integration of physically unclonable functions (PUFs) offers a promising solution [120–123]. PUFs leverage inherent physical variations in semiconductor devices to generate unique identifiers for each chip or IP cell. Incorporating PUFs into analog circuits during fabrication associates a unique identifier for each individual analog die or IP block. The use of PUFs to enhance the security of an analog circuit by producing a unique key across dies provides significant challenge to attackers that attempt to compromise a system even if one key is recovered.

Current PUFs suffer from stability and reliability issues due to higher sensitivity to environmental factors, manufacturing variations, and aging. In addition, a study [227] has illustrated the use of machine learning-based modeling attacks to determine the response of a strong PUF to a challenge. A modeling attack assumes that an adversary has collected a subset of challenge-response pairs (CRPs) of the PUF, and attempts to derive a model from the CRP data. If an accurate model of the PUF is developed, then the adversary is able to predict the correct response to arbitrary challenges with high probability, which circumvents the security provided by a PUF and of any protocols that utilize the PUF. Therefore, further research is needed in the development of stable, area efficient, low cost strong PUFs that possess high modeling resilience.

The objective for key-based analog locking techniques is to create tamper-proof memory that is secure and utilized across the analog IP supply chain. The keys stored in the

tamper-proof memory will be utilized with PUFs to generate a unique key for each individual analog die, which thwarts an adversary from unlocking all dies of a given analog circuit by determining a single key.

Bibliography

- [1] R. Schaller, “Moore’s law: Past, Present and Future,” *IEEE Spectrum*, Vol. 34, No. 6, pp. 52-59, June 1997.
- [2] ITRS, “International Technology Roadmap for Semiconductors,” March 2011.
- [3] Precedence Research, “Semiconductor Market (By Component: Logic Devices, MPU, Power Devices, MCU, Analog IC, Memory Devices, Sensors, Discrete Power Devices, and Others; Application: Data Processing, Industrial, Networking and Communications, Consumer Electronics, Automotive, and Government) - Global Industry Analysis, Size, Share, Growth, Trends, Regional Outlook, and Forecast 2022 - 2030,” December 2021.
- [4] O. Research, “Analog Integrated Circuit (IC) Market by 2019 by Type, Application, Provider, Demand Analysis, Emerging Trends and Investment Opportunities to 2024,” August 2016.
- [5] A. R. C. Report, “Analog Integrated Circuit (IC) Market 2019 – Industry Size, Share, Dynamics, Status, Outlook and Opportunities: 2024,” pp. 1–120, September 2019.
- [6] I. M. Report, “IoT Platforms: Enabling the Internet of Things,” March 2016.
- [7] Mordor Intelligence LLP, “Global Analog Integrated Circuit (IC) Market - Growth, Trends, COVID-19 Impact, and Forecasts (2022 - 2027),” September 2022.
- [8] C. Brown and G. Linden, *Chips and Change: How Crisis Reshapes the Semiconductor Industry*. MIT Press, August 2009.

- [9] M. M. Alam, S. Chowdhury, B. Park, D. Munzer, N. Maghari, M. Tehranipoor, and D. Forte, "Challenges and Opportunities in Analog and Mixed Signal (AMS) Integrated Circuit (IC) Security," *Journal of Hardware and Systems Security*, Vol. 2, No. 1, pp. 15–32, March 2018.
- [10] I. Corporation, "IBS Corporation, Industry Reports," 2003.
- [11] I. Technology, "Electronics Equipment and Semiconductor Market Forecast Tool," September 2014.
- [12] T. Freeth, *et al.*, "Decoding The Ancient Greek Astronomical Calculator Known as the Antikythera Mechanism," *Nature*, Vol. 444, No. 7119, pp. 587–591, November 2006.
- [13] B. Ulmann, *Analog Computing*. Oldenbourg Wissenschaftsverlag, July 2013.
- [14] V. Bush, "The Differential Analyzer. A New Machine for Solving Differential Equations," *Journal of the Franklin Institute*, Vol. 212, No. 4, pp. 447–488, October 1931.
- [15] D. Mindell, "Automation's Finest Hour: Bell Labs and Automatic Control in World War II," *IEEE Control Systems Magazine*, Vol. 15, No. 6, pp. 72–80, December 1995.
- [16] K. Lundberg, "The History of Analog Computing: Introduction to The Special Section," *IEEE Control Systems Magazine*, Vol. 25, No. 3, pp. 22–25, June 2005.
- [17] J. Ragazzini, R. Randall, and F. Russell, "Analysis of Problems in Dynamics by Electronic Circuits," *Proceedings of the Institute of Radio Engineers (IRE)*, Vol. 35, No. 5, pp. 444–452, May 1947.
- [18] *Model K2-W Operational Amplifier*, George A. Philbrick Researches, Inc., rev. D. [Online]. Available: http://www.philbrickarchive.org/k2-w_refurbished.pdf
- [19] A. Filipkowski, "Poles and zeros in transistor amplifiers introduced by Miller effect," *IEEE Transactions on Education*, Vol. 42, No. 4, pp. 349–351, November 1999.
- [20] C. Weiner, "How the transistor emerged," *IEEE Spectrum*, Vol. 10, No. 1, pp. 24–33, Jan 1973.

- [21] W. F. Brinkman, D. E. Haggan, and W. W. Troutman, "A History of the Invention of the Transistor and Where it Will Lead Us," *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 12, pp. 1858-1865, Dec 1997.
- [22] J. Bardeen and W. H. Brattain, "The Transistor, a Semi-Conductor Triode," *Physical Review*, Vol. 74, pp. 230-231, Jul 1948.
- [23] I. Ross, "The Invention of the Transistor," *Proceedings of the IEEE*, Vol. 86, No. 1, pp. 7-28, January 1998.
- [24] W. Shockley, "The Theory of P-N Junctions in Semiconductors and P-N Junction Transistors," *Bell System Technical Journal*, Vol. 28, No. 3, pp. 435-489, 1949.
- [25] J. M. Early, "Effects of Space-Charge Layer Widening in Junction Transistors," *Proceedings of the Institute of Radio Engineers (IRE)*, Vol. 40, No. 11, pp. 1401-1406, November 1952.
- [26] J. J. Ebers and J. L. Moll, "Large-Signal Behavior of Junction Transistors," *Proceedings of the Institute of Radio Engineers (IRE)*, Vol. 42, No. 12, pp. 1761-1772, December 1954.
- [27] C. Pirtle, *Engineering the World: Stories from the First 75 Years of Texas Instruments*, Ph.D. Dissertation, Southern Methodist University, Dallas, Texas, 2005.
- [28] T. R. Reid, *The Chip: How Two Americans Invented the Microchip and Launched a Revolution*. Random House, 2001.
- [29] L. Berlin, *Man Behind the Microchip: Robert Noyce and the Invention of Silicon Valley*. Oxford University Press, 2005.
- [30] N. Noyce, "Microelectronics," *Scientific American*, Vol. 237, No. 3, pp. 62-69, September 1977.
- [31] $\mu A703$ RF-IF Amplifier, Fairchild Linear Integrated Circuits, rev. D. [Online]. Available: <https://datasheetspdf.com/pdf-file/619527/FairchildSemiconductor/UA703/1>
- [32] $\mu A702$ Wideband DC Amplifier, Fairchild Linear Integrated Circuits, rev. D. [Online]. Available: <https://datasheetspdf.com/pdf-file/1305480/Fairchild/UA702/1>

- [33] μ A709 *High Performance Operational Amplifier*, Fairchild Linear Integrated Circuits, rev. D. [Online]. Available: <https://datasheetspdf.com/pdf-file/303668/FairchildSemiconductor/UA709/1>
- [34] R. Widlar, "New Developments in IC Voltage Regulators," *IEEE Journal of Solid-State Circuits*, Vol. 6, No. 1, pp. 2-7, February 1971.
- [35] R. Widlar, "INVITED: New Approaches For The Design Of Monolithic Operational Amplifiers," *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC). Digest of Technical Papers*, pp. 10-11, February 1969.
- [36] B. Gilbert, "A DC-500MHz Amplifier/Multiplier Principle," *Proceedings of the IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, pp. 114-115, February 1968.
- [37] G. Wilson, "A Monolithic Junction FET-n-p-n Operational Amplifier," *IEEE Journal of Solid-State Circuits*, Vol. 3, No. 4, pp. 341-348, December 1968.
- [38] T. Perry, "Donald O. Pederson [Electronic Engineering Biography]," *IEEE Spectrum*, Vol. 35, No. 6, pp. 22-27, June 1998.
- [39] E. C. Hall, "Integrated Circuits in the Apollo Guidance Computer," pp. 1-24, 1963.
- [40] E. C. Hall, "A Case History of the AGC Integrated Logic Circuit," *Apollo The International Magazine Of Art And Antiques*, pp. 1-40, 1965.
- [41] L. D. Hanley, J. Partridge, and E. C. Hall, "The Application of Failure Analysis in Procuring and Screening of Integrated Circuits," Massachusetts Institute of Technology, Technical Report, October 1965.
- [42] I. Young, "A History of The Continuously Innovative Analog Integrated Circuit," *IEEE Solid-State Circuits Society Newsletter*, Vol. 12, No. 4, pp. 52-57, September 2007.
- [43] J. Pastoriza, H. Krabbe, and A. Molinari, "A High-Performance Monolithic D/A Converter Circuit," *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. XIII, pp. 122-123, February 1970.

- [44] R. Suarez, P. Gray, and D. Hodges, "An All-MOS Charge-redistribution A/D Conversion Technique," *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. XVII, pp. 194-195, February 1974.
- [45] A. Dingwall and B. Rosenthal, "Low-power Monolithic COS/MOS Dual-slope 11-bit A/D Converter," *1976 IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. XIX, pp. 146-147, February 1976.
- [46] R. Van De Plassche, "Dynamic Element Matching for High-accuracy Monolithic D/A Converters," *IEEE Journal of Solid-State Circuits*, Vol. 11, No. 6, pp. 795-800, December 1976.
- [47] Y. Tsvividis and P. Gray, "An Integrated NMOS Operational Amplifier With Internal Compensation," *IEEE Journal of Solid-State Circuits*, Vol. 11, No. 6, pp. 748-753, December 1976.
- [48] I. Young, D. Hodges, and P. Gray, "Analog NMOS Sampled-data Recursive Filter," *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. XX, pp. 156-157, February 1977.
- [49] T. Choi, R. Kaneshiro, R. Brodersen, P. Gray, W. Jett, and M. Wilcox, "High-frequency CMOS Switched-capacitor Filters for Communications Application," *IEEE Journal of Solid-State Circuits*, Vol. 18, No. 6, pp. 652-664, December 1983.
- [50] H.-S. Lee, D. Hodges, and P. Gray, "A Self Calibrating 12b 12 μ s CMOS ADC," *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. XXVII, pp. 64-65, February 1984.
- [51] S. Lewis and P. Gray, "A Pipelined 5MHz 9b ADC," *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, Vol. XXX, pp. 210-211, February 1987.
- [52] R. Adams and K. Nguyen, "A 113-dB SNR Oversampling DAC With Segmented Noise-shaped Scrambling," *IEEE Journal of Solid-State Circuits*, Vol. 33, No. 12, pp. 1871-1878, December 1998.
- [53] H. Nyquist, "Certain Topics in Telegraph Transmission Theory," *Transactions of the American Institute of Electrical Engineers*, Vol. 47, No. 2, pp. 617-644, April 1928.

- [54] C. Shannon, "Communication in the Presence of Noise," *Proceedings of the Institute of Radio Engineers (IRE)*, Vol. 37, No. 1, pp. 10-21, January 1949.
- [55] J. Candy, "A Use of Limit Cycle Oscillations to Obtain Robust Analog-to-Digital Converters," *IEEE Transactions on Communications*, Vol. 22, No. 3, pp. 298-305, March 1974.
- [56] T. Riley, M. Copeland, and T. Kwasniewski, "Delta-sigma Modulation in Fractional-N Frequency Synthesis," *IEEE Journal of Solid-State Circuits*, Vol. 28, No. 5, pp. 553-559, May 1993.
- [57] J. E. Lilienfeld, "Method and Apparatus for Controlling Electric Currents," 1930, u.S. 1 745 175A.
- [58] G. Patton, S. Iyer, S. Delage, S. Tiwari, and J. Stork, "Silicon-germanium Base Heterojunction Bipolar Transistors by Molecular Beam Epitaxy," *IEEE Electron Device Letters*, Vol. 9, No. 4, pp. 165-167, April 1988.
- [59] R. Pengelly and A. Podell, "MMIC CAD tools and workstations," *Proceedings of the IEEE (GaAs IC) Symposium, Gallium Arsenide Integrated Circuit. Technical Digest 1988.*, pp. 325-326, November 1988.
- [60] L. Kern, "Design and Development of A Real-time Neural Processor Using the Intel 80170NX ETANN," *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, Vol. 2, pp. 684-689, June 1992.
- [61] A. Chatterjee, C. Machala, and P. Yang, "A Submicron DC MOSFET Model for Simulation of Analog Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 14, No. 10, pp. 1193-1207, October 1995.
- [62] K. Bult, "Analog Design in Deep Sub-micron CMOS," *Proceedings of the IEEE European Solid-State Circuits Conference*, pp. 126-132, September 2000.
- [63] T. Quach, "Proceedings of the IEEE Radio Frequency Integrated Circuits (RFIC) Symposium. Digest of Papers (Cat. No.03CH37410)," *IEEE Radio Frequency Integrated Circuits (RFIC) Symposium, 2003*, June 2003.

- [64] J. Govil, J. and Govil, "4G Mobile Communication Systems: Turns, Trends and Transition," *Proceedings of the International Conference on Convergence Information Technology (ICCIT)*, pp. 13-18, November 2007.
- [65] D. Kang, D. Kim, J. Choi, J. Kim, Y. Cho, and B. Kim, "A Multimode/Multiband Power Amplifier With a Boosted Supply Modulator," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 58, No. 10, pp. 2598-2608, 2010.
- [66] J. Crossley, *et al.*, "BAG: A Designer-oriented Integrated Framework for the Development of AMS Circuit Generators," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 74-81, December 2013.
- [67] *AML100 AnalogMLTM Chip Product Brief*, Aspinity, rev. D. [Online]. Available: <https://www.aspinity.com/Files/Admin/Documents/PB3-1.0-AML100-Product-Brief.pdf>
- [68] J. D. Dunworth, *et al.*, "A 28GHz Bulk-CMOS Dual-polarization Phased-array Transceiver With 24 Channels for 5G User and Basestation Equipment," *Proceedings IEEE International Solid - State Circuits Conference - (ISSCC)*, pp. 70-72, February 2018.
- [69] E. J. Fuller, *et al.*, "Parallel Programming of an Ionic Floating-gate Memory Array for Scalable Neuromorphic Computing," *Science*, Vol. 364, No. 6440, pp. 570-574, April 2019.
- [70] B. Thompson, H.-S. Lee, and L. DeVito, "A 300-MHz BiCMOS Serial Data Transceiver," *IEEE Journal of Solid-State Circuits*, Vol. 29, No. 3, pp. 185-192, March 1994.
- [71] H. Mussman, "Technologies for Broadband Networks," *1987 IEEE International Solid-State Circuits Conference (ISSCC). Digest of Technical Papers*, Vol. XXX, pp. 246-247, February 1987.
- [72] M. Bopp, *et al.*, "A DECT Transceiver Chip Set Using SiGe Technology," *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 68-69, August 1999.

- [73] S. Köppel, B. Ulmann, L. Heimann, and D. Killat, “Using Analog Computers in Today’s Largest Computational Challenges,” *Copernicus GmbH Journal of Advances in Radio Science*, Vol. 19, No. D., pp. 105–116, February 2021.
- [74] G. Cowan, R. Melville, and Y. Tsividis, “A VLSI Analog Computer/Digital Computer Accelerator,” *IEEE Journal of Solid-State Circuits*, Vol. 41, No. 1, pp. 42–53, December 2006.
- [75] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, “McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures,” *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, p. 469–480, December 2009.
- [76] G. M. Amdahl, “Computer Architecture and Amdahl’s Law,” *IEEE Journal on Computer*, Vol. 46, No. 12, pp. 38–46, December 2013.
- [77] J. Sim, A. Dasgupta, H. Kim, and R. Vuduc, “A Performance Analysis Framework for Identifying Potential Benefits in GPGPU Applications,” *ACM Special Interest Group on Programming Languages (SIGPLAN) Notices*, No. 8, p. 11–22, February 2012.
- [78] L. Gatet, H. Tap-Beteille, and F. Bony, “Comparison Between Analog and Digital Neural Network Implementations for Range-Finding Applications,” *IEEE Transactions on Neural Networks*, Vol. 20, No. 3, pp. 460–470, 2009.
- [79] W. Cao, Y. Zhao, A. Bolor, Y. Han, X. Zhang, and L. Jiang, “Neural-PIM: Efficient Processing-In-Memory With Neural Approximation of Peripherals,” *IEEE Transactions on Computers*, Vol. 71, No. 9, pp. 2142–2155, October 2022.
- [80] MarketsandMarkets, “Artificial Neural Network by Component (Solutions, Platform/API and Services), Application (Image Recognition, Signal Processing, and Data Mining), Deployment Mode, Organization Size, Industry Size, Industry Vertical, and Region - Global Forecast to 2024,” October 2019.
- [81] J. Scheible and J. Lienig, “Automation of Analog IC Layout: Challenges and Solutions,” *Proceedings of the ACM International Symposium on Physical Design (ISPD)*, pp. 33–40, March 2015.

- [82] G. Gielen and R. Rutenbar, "Computer-aided Design of Analog and Mixed-signal Integrated Circuits," *Proceedings of the IEEE*, Vol. 88, No. 12, pp. 1825-1854, 2000.
- [83] Semiconductor Engineering, "Problems And Solutions In Analog Design," August 2020.
- [84] Semiconductor Engineering, "Increase In Analog Problems," October 2020.
- [85] I. Technology, "Electronics Equipment and Semiconductor Market Forecast Tool," September 2014.
- [86] IARPA, "Trusted Integrated Chips (TIC) Program," IARPA-BAA-11-09, October 2011.
- [87] R. Torrance and D. James, "The State-Of-The-Art in Semiconductor Reverse Engineering," *Proceedings of the ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 333-338, June 2011.
- [88] J. McMaster, "What's In A Counterfeit AD633 Analog Multiplier From @Ascii211?" September 2020.
- [89] K. Shirriff, "How to Multiply Currents: Inside A Counterfeit Analog Multiplier," September 2020.
- [90] *Low Cost Analog Multiplier AD633*, Analog Devices, rev. D. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad633.pdf>
- [91] *RC4200 Analog Multiplier*, Raytheon Semiconductor, rev. D. [Online]. Available: https://www.teledynedefenseelectronics.com/e2vhrel/Signal%20Chain%20Datasheets/Documents/Ray_FSC_RM4200.pdf
- [92] Zeptobars Blogpost, "Fake audiophile opamps: OPA627 (AD744?!)," April 2014.
- [93] *Precision, 500 ns Settling BiFET Op Amp*, Analog Devices, rev. D. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD744.pdf>

- [94] Zeptobars Blogpost, “OPA627, Genuine One This Time : Weekend Die-shot,” July 2014.
- [95] *OPA627 and OPA637 Precision High-Speed Difet[®] Operational Amplifier*, Texas Instruments, rev. D. [Online]. Available: <https://www.ti.com/product/OPA627?qgpn=opa627>
- [96] AirForce Times, “An F-16 Pilot Died When His Ejection Seat Failed. Was It Counterfeit?” September 2022.
- [97] IDG News Service, “Woman Admits Aiding Firm in Selling Fake Chips to U.S. Military,” November 2010.
- [98] Semiconductor Engineering, “Combating Counterfeit Semiconductors In The Military Supply Chain,” October 2019.
- [99] SIA Anti-Counterfeiting Task Force, “Winning the Battle Against Counterfeit Semiconductor Products,” *White Paper by Semiconductor Industry Association*, pp. 1–28, August 2013.
- [100] A. Ghadge, A. Duck, M. Er, and N. Caldwell, “Deceptive Counterfeit Risk in Global Supply Chains,” *Supply Chain Forum: An International Journal*, Vol. 22, No. 2, pp. 87-99, April 2021.
- [101] Y. Bi, J. Yuan, and Y. Jin, “Beyond the Interconnections: Split Manufacturing in RF Designs,” Vol. 4, No. 3, pp. 541-564, August 2015.
- [102] J. Rajendran, O. Sinanoglu, and R. Karri, “Is Split Manufacturing Secure?” *Design, Automation Test in Europe Conference Exhibition*, pp. 1259-1264, March 2013.
- [103] T. D. Perez and S. Pagliarini, “A Survey on Split Manufacturing: Attacks, Defenses, and Challenges,” *IEEE Access*, Vol. 8, pp. 184013-184035, October 2020.
- [104] J. Rajendran, O. Sinanoglu, and R. Karri, “Regaining Trust in VLSI Design: Design-for-Trust Techniques,” *Proceedings of the IEEE*, Vol. 102, No. 8, pp. 1266-1282, July 2014.

- [105] J. Leonhard, A. Sayed, M. M. Louërat, H. Aboushady, and H. G. Stratigopoulos, "Analog and Mixed-Signal IC Security via Sizing Camouflaging," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 40, No. 5, pp. 822-835, June 2021.
- [106] R. P. Cocchi, J. P. Baukus, L. W. Chow, and B. J. Wang, "Circuit Camouflage Integration for Hardware IP Protection," *Proceedings of the ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1-5, June 2014.
- [107] A. K. Sharma, M. Madhusudan, S. M. Burns, P. Mukherjee, S. Yaldiz, R. Harjani, and S. S. Sapatnekar, "Common-Centroid Layouts for Analog Circuits: Advantages and Limitations," *Proceedings of the IEEE Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1224-1229, February 2021.
- [108] N. G. Jayasankaran, A. S. Borbon, A. Abuellil, E. Sánchez-Sinencio, J. Hu, and J. Rajendran, "Breaking Analog Locking Techniques via Satisfiability Modulo Theories," *Proceedings of the IEEE International Test Conference (ITC)*, pp. 1-10, February 2019.
- [109] J. Roy, F. Koushanfar, and I. Markov, "EPIC: Ending Piracy of Integrated Circuits," *Proceedings of the IEEE/ACM Design, Automation and Test in Europe Conference*, pp. 1069-1074, October 2008.
- [110] J. Rajendran, H. Zhang, C. Zhang, G. Rose, Y. Pino, O. Sinanoglu, and R. Karri, "Fault Analysis-Based Logic Encryption," *IEEE Transactions on Computers*, Vol. 64, No. 2, pp. 410-424, February 2015.
- [111] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC Piracy Using Reconfigurable Logic Barriers," *IEEE Design and Test of Computers*, Vol. 27, No. 1, pp. 66-75, February 2010.
- [112] D. H. K. Hoe, J. Rajendran, and R. Karri, "Towards Secure Analog Designs: A Secure Sense Amplifier Using Memristors," *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pp. 516-521, July 2014.
- [113] N. G. Jayasankaran, A. S. Borbon, E. Sanchez-Sinencio, J. Hu, and J. Rajendran, "Towards Provably-secure Analog and Mixed-signal Locking Against Overproduction," *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, pp. 1-8, November 2018.

- [114] J. Leonhard, M. Yasin, S. Turk, M. N. Nabeel, M. Louërat, R. C. Avot, H. Aboushady, O. Sinanoglu, and H. G. Stratigopoulos, "MixLock: Securing Mixed-signal Circuits via Logic Locking," *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 84-89, March 2019.
- [115] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. Rajendran, and O. Sinanoglu, "Provably-Secure Logic Locking: From Theory To Practice," *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 1601–1618, November 2017.
- [116] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the Security of Logic Encryption Algorithms," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 137-143, May 2015.
- [117] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Removal Attacks on Logic Locking and Camouflaging Techniques," *IEEE Transactions on Emerging Topics in Computing*, Vol. 8, No. 2, pp. 517-532, August 2020.
- [118] K. Shamsi, M. Li, T. Meade, Z. Zhao, D. Z. Pan, and Y. Jin, "AppSAT: Approximately Deobfuscating Integrated Circuits," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust*, pp. 95-100, May 2017.
- [119] K. Juretus, V. V. Rao, and I. Savidis, "Securing Analog Mixed-signal Integrated Circuits Through Shared Dependencies," *Proceedings of the Great Lakes Symposium on VLSI*, pp. 483–488, May 2019.
- [120] U. Rührmair and D. E. Holcomb, "PUFs At A Glance," *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1-6, March 2014.
- [121] N. Beckmann and M. Potkonjak, "Hardware-Based Public-Key Cryptography with Public Physically Unclonable Functions," *Information Hiding: 11th International Workshop, IH 2009, Darmstadt, Germany, June 8-10, 2009, Revised Selected Papers*. Springer-Verlag, 2009, p. 206–220.
- [122] C. Herder, M. D. Yu, F. Koushanfar, and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," *Proceedings of the IEEE*, Vol. 102, No. 8, pp. 1126-1141, August 2014.

- [123] J. L. Danger, S. Guilley, P. Nguyen, and O. Rioul, "PUFs: Standardization and Evaluation," *Proceedings of the IEEE Mobile System Technologies Workshop (MST)*, pp. 12-18, September 2016.
- [124] J. Lee, D. Lim, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, "A Technique to Build A Secret Key In Integrated Circuits For Identification And Authentication Applications," *Proceedings of the IEEE Symposium on VLSI Circuits. Digest of Technical Papers*, pp. 176-179, June 2004.
- [125] R. Helinski, D. Acharyya, and J. Plusquellic, "A Physical Unclonable Function Defined Using Power Distribution System Equivalent Resistance Variations," *Proceedings of the IEEE/ACM Annual Design Automation Conference*, p. 676-681, July 2009.
- [126] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical One-Way Functions," *Science*, Vol. 297, No. 5589, pp. 2026-2030, September 2002.
- [127] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "PUF Modeling Attacks on Simulated and Silicon Data," *IEEE Transactions on Information Forensics and Security*, Vol. 8, No. 11, pp. 1876-1891, August 2013.
- [128] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, pp. 9-14, June 2007.
- [129] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, and U. Rührmair, "The Bistable Ring PUF: A New Architecture For Strong Physical Unclonable Functions," *Proceedings of the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 134-141, June 2011.
- [130] K. Yang, Q. Dong, D. Blaauw, and D. Sylvester, "14.2 A Physically Unclonable Function With $BER < 10^{-8}$ For Robust Chip Authentication Using Oscillator Collapse in 40nm CMOS," *Proceedings of the IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, pp. 1-3, February 2015.
- [131] A. R. Korenda, F. Afghah, B. Cambou, and C. Philabaum, "A Proof of Concept SRAM-based Physically Unclonable Function (PUF) Key Generation Mechanism for IoT

- Devices,” *Proceedings of the IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1-8, June 2019.
- [132] Y. Su, J. Holleman, and B. P. Otis, “A Digital 1.6 pJ/bit Chip Identification Circuit Using Process Variations,” *IEEE Journal of Solid-State Circuits*, Vol. 43, No. 1, pp. 69-77, January 2008.
 - [133] R. Maes, V. Rozic, I. Verbauwhede, P. Koeberl, E. Van der Sluis, and V. Van der Leest, “Experimental Evaluation of Physically Unclonable Functions in 65 nm CMOS,” *Proceedings of the IEEE Proceedings of the ESSCIRC (ESSCIRC)*, pp. 486-489, September 2012.
 - [134] S. K. Mathew, S. K. Satpathy, M. A. Anders, H. Kaul, S. K. Hsu, A. Agarwal, G. K. Chen, R. J. Parker, R. K. Krishnamurthy, and V. De, “16.2 A 0.19pJ/b PVT-variation-tolerant Hybrid Physically Unclonable Function Circuit For 100Stable Secure Key Generation in 22nm CMOS,” *Proceedings of the IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 278-279, February 2014.
 - [135] M. Bhargava, C. Cakir, and K. Mai, “Attack Resistant Sense Amplifier Based PUFs (SA-PUF) With Deterministic and Controllable Reliability of PUF Responses,” *Proceedings of the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pp. 106-111, June 2010.
 - [136] A. B. Alvarez, W. Zhao, and M. Alioto, “Static Physically Unclonable Functions for Secure Chip Identification With 1.9–5.8% Native Bit Instability at 0.6–1 V and 15 fJ/bit in 65 nm,” *IEEE Journal of Solid-State Circuits*, Vol. 51, No. 3, pp. 763-775, February 2016.
 - [137] J. Li and M. Seok, “Ultra-Compact and Robust Physically Unclonable Function Based on Voltage-Compensated Proportional-to-Absolute-Temperature Voltage Generators,” *Proceedings of the IEEE Journal of Solid-State Circuits*, Vol. 51, No. 9, pp. 2192-2202, September 2016.
 - [138] L. De Moura and N. Bjørner, “Satisfiability Modulo Theories: Introduction and Applications,” *Communications of the ACM*, Vol. 54, No. 9, p. 69–77, September 2011.
 - [139] R. G. Jeroslow and J. Wang, “Solving Propositional Satisfiability Problems,” *Annals of Mathematics and Artificial Intelligence*, Vol. 1, No. 1, pp. 167-187, September 1990.

- [140] O. Lahiouel, M. Zaki, and S. Tahar, “Towards Enhancing Analog Circuit Sizing Using SMT-based Techniques,” *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, pp. 1-6, June 2015.
- [141] M. Davis and H. Putnam, “A Computing Procedure for Quantification Theory,” *Journal of the Association for Computing Machinery (ACM)*, Vol. 7, No. 3, p. 201–215, July 1960.
- [142] M. Davis, G. Logemann, and D. Loveland, “A Machine Program for Theorem-Proving,” *Communications of the Association for Computing Machinery (ACM)*, Vol. 5, No. 7, p. 394–397, July 1962.
- [143] R. Mukul, A. Biere, and A. Gupta, “A Survey of Recent Advances in SAT-Based Formal Verification,” Vol. 7, No. 2, pp. 156–173, April 2005.
- [144] D. Loveland, A. Sabharwal, and B. Selman, “DPLL: The Core of Modern Satisfiability Solvers,” *Martin Davis on Computability, Computational Logic, and Mathematical Foundations*. Springer International Publishing, 2016, pp. 315–335.
- [145] J. P. Marques-Silva and K. A. Sakallah, “Boolean Satisfiability in Electronic Design Automation,” *Proceedings of the IEEE/ACM Design Automation Conference (DAC)*, p. 675–680, June 2000.
- [146] S. Eggersgluß, R. Wille, and R. Drechsler, “Improved SAT-based ATPG: More Constraints, Better Compaction,” *Proceedings IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 85-90, November 2013.
- [147] R. Drechsler, S. Eggersgluß, G. Fey, and D. Tille, “SAT-Based ATPG,” *Test Pattern Generation using Boolean Proof Engines*. Springer Netherlands, 2009, pp. 43–52.
- [148] J. Huang, H. L. Zhen, N. Wang, M. Yuan, H. Mao, Y. Huang, and J. Tao, “Accelerate SAT-based ATPG via Preprocessing and New Conflict Management Heuristics,” *Proceedings of the IEEE Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 365-370, January 2022.
- [149] E. Goldberg, M. Prasad, and R. Brayton, “Using SAT for Combinational Equivalence Checking,” *Proceedings the IEEE/ACM Design, Automation and Test in Europe. Conference and Exhibition (DATE)*, pp. 114-121, March 2001.

- [150] V. Paruthi and A. Kuehlmann, "Equivalence Checking Combining a Structural SAT-solver, BDDs, and Simulation," *Proceedings of the IEEE International Conference on Computer Design (ICCD)*, pp. 459-464, September 2000.
- [151] K. L. McMillan, "Interpolation and SAT-Based Model Checking," *Proceedings of the International Conference on Computer Aided Verification*, pp. 1-13, July 2003.
- [152] A. Biere and D. Kroning, "SAT-Based Model Checking," *Handbook of Model Checking*. Springer International Publishing, 2018, pp. 277-303.
- [153] H. T. Zhang, J. H. R. Jiang, and A. Mishchenko, "A Circuit-Based SAT Solver for Logic Synthesis," *Proceedings of the IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1-6, November 2021.
- [154] L. Amarú, F. Marranghello, E. Testa, C. Casares, V. Possani, J. Luo, P. Vuillod, A. Mishchenko, and G. De Micheli, "SAT-Sweeping Enhanced for Logic Synthesis," *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, pp. 1-6, 2020.
- [155] A. Mishchenko, R. Brayton, J. H. R. Jiang, and S. Jang, "Scalable Don't-Care-Based Logic Optimization and Resynthesis," *ACM Transactions on Reconfigurable Technology and System*, Vol. 4, No. 4, December 2011.
- [156] P. Tafertshofer, A. Ganz, and M. Henftling, "A SAT-based Implication Engine For Efficient ATPG, Equivalence Checking, And Optimization Of Netlists," *Proceedings of IEEE International Conference on Computer Aided Design (ICCAD)*, pp. 648-655, November 1997.
- [157] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," *2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 137-143, May 2015.
- [158] Y. Matsunaga and M. Yoshimura, "An Efficient SAT-Attack Algorithm Against Logic Encryption," *Proceedings of the IEEE International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pp. 44-47, July 2019.
- [159] M. Yasin, B. Mazumdar, J. Rajendran, and O. Sinanoglu, "SARLock: SAT Attack Resistant Logic Locking," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust*, pp. 236-241, May 2016.

- [160] K. Juretus and I. Savidis, "Reduced Overhead Gate Level Logic Encryption," *Proceedings of the IEEE/ACM Great Lakes Symposium on Very Large Scale Integration (GLSVLSI)*, pp. 15-20, May 2016.
- [161] K. Juretus and I. Savidis, "Increasing the SAT Attack Resiliency of In-Cone Logic Locking," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5, May 2019.
- [162] M. Yasin, A. Sengupta, M. T. Nabeel, M. Ashraf, J. Rajendran, and O. Sinanoglu, "Provably-Secure Logic Locking: From Theory To Practice," *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, p. 1601–1618, October 2017.
- [163] X. Xu, B. Shakya, M. M. Tehranipoor, and t. Forte, D.
- [164] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Removal Attacks on Logic Locking and Camouflaging Techniques," *IEEE Transactions on Emerging Topics in Computing*, August 2017.
- [165] M. El Massad, S. Garg, and M. Tripunitara, "The SAT Attack on IC Camouflaging: Impact and Potential Countermeasures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1-15, 2019.
- [166] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the Security of Logic Encryption Algorithms," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust*, pp. 137-143, May 2015.
- [167] G. S. Tseytin, "On the Complexity of Derivations in the Propositional Calculus," *Studies in Mathematics and Mathematical Logic*, Vol. II, pp. 115–125, 1968.
- [168] M. Yasin, J. Rajendran, O. Sinanoglu, and R. Karri, "On Improving the Security of Logic Locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 35, No. 9, pp. 1411-1424, September 2016.
- [169] S. Dupuis, P. S. Ba, G. Di Natale, M. L. Flottes, and B. Rouzeyre, "A Novel Hardware Logic Encryption Technique for Thwarting Illegal Overproduction and Hardware Trojans," *Proceedings of the IEEE International On-Line Testing Symposium*, pp. 49-54, July 2014.

- [170] L. De Moura and N. Bjørner, *Satisfiability Modulo Theories: An Appetizer*. Springer Berlin Heidelberg, 2009, pp. 23-36.
- [171] L. De Moura, N. Bjørner, and N. Shankar, *A Tutorial on Satisfiability Modulo Theories*. Springer Berlin Heidelberg, 2009, pp. 20-36.
- [172] R. Robere, A. Kolokolova, and V. Ganesh, *The Proof Complexity of SMT Solvers*. Springer International Publishing, 2018, pp. 275-293.
- [173] W. Nye, D. C. Riley, A. Sangiovanni-Vincentelli, and A. L. Tits, "DELIGHT.SPICE: an optimization-based system for the design of integrated circuits," *Proceedings of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 501-519, April 1988.
- [174] R. A. Rutenbar, G. G. E. Gielen, and B. A. Antao, *STAIC: An Interactive Framework for Synthesizing CMOS and BiCMOS Analog Circuits*. Wiley-IEEE Press, 2002.
- [175] H. Y. Koh, C. H. Sequin, and P. R. Gray, "OPASYN: a compiler for CMOS operational amplifiers," *Proceedings of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 113-125, February 1990.
- [176] R. A. Rutenbar, G. G. E. Gielen, and B. A. Antao, *IDAC: An Interactive Design Tool for Analog CMOS Circuits*. Wiley-IEEE Press, 2002.
- [177] M. Krasnicki, R. Phelps, R. A. Rutenbar, and L. R. Carley, "MAELSTROM: efficient simulation-based synthesis for custom analog cells," *Proceedings of the ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 945-950, August 1999.
- [178] E. S. Ochotta, R. A. Rutenbar, and L. R. Carley, "ASTRX/OBLX: Tools for Rapid Synthesis of High-Performance Analog Circuits," *Proceedings of the ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 24-30, June 1994.
- [179] J. Lee and Y.-B. Kim, "ASLIC: a Low Power CMOS Analog Circuit Design Automation," *Proceedings on the IEEE International Symposium on Quality Electronic Design (isqed'05)*, pp. 470-475, March 2005.
- [180] R. A. Rutenbar, G. G. E. Gielen, and B. A. Antao, *OASYS: A Framework for Analog Circuit Synthesis*. Wiley-IEEE Press, 2002.

- [181] M. Barros, J. Guilherme, and N. Horta, "Analog Circuits Optimization Based on Evolutionary Computation Techniques," *Proceedings of the ACM VLSI Integration Journal*, pp. 136–155, 2010.
- [182] D. L. W. Kruiskamp, "DARWIN: CMOS Opamp Synthesis by Means of a Genetic Algorithm," *Proceedings of the ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 433–438, June 1995.
- [183] T. Sripramong and C. Toumazou, "The Invention of CMOS Amplifiers Using Genetic Programming and Current-Flow Analysis," *Proceedings of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1237–1252, November 2002.
- [184] M. d. Hershenson, S. P. Boyd, and T. H. Lee, "Optimal Design of a CMOS Op-Amp Via Geometric Programming," *Proceedings of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–21, January 2001.
- [185] B. Liu, F. V. Fernández, G. Gielen, R. Castro-López, and E. Roca, "A Memetic Approach to the Automatic Design of High-performance Analog Integrated Circuits," *Proceedings of the ACM Transactions on Design Automation of Electronic Systems (TO-DAES)*, pp. 42:1–42:24, 2009.
- [186] R. Mukherjee, M. Purandare, R. Polig, and D. Kroening, "Formal Techniques for Effective Co-verification of Hardware/Software Co-designs," *Proceedings of the IEEE/ACM Annual Design Automation Conference (DAC)*, pp. 1–6, June 2017.
- [187] Y. Deng, "SAT Based Verification for Analog and Mixed Signal Circuits," *Masters Thesis, Texas A & M University*, pp. 1–65, 2012.
- [188] K. Scheibler, S. Kupferschmid, and B. Becker, "Recent Improvements in the SMT Solver iSAT," *Proceedings of the Methods and Description Languages for the Modeling and Verification of Circuits and Systems Conference*, pp. 231–241, March 2013.
- [189] R. Torrance and D. James, "The State-of-the-Art in Semiconductor Reverse Engineering," *Proceedings of the IEEE Design Automation Conference*, pp. 333–338, June 2011.

- [190] V. V. Rao and I. Savidis, "Mesh Based Obfuscation of Analog Circuit Properties," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5, May 2019.
- [191] V. V. Rao and I. Savidis, "Protecting Analog Circuits with Parameter Biasing Obfuscation," *Proceedings of the IEEE Latin American Test Symposium (LATS)*, pp. 1-6, March 2017.
- [192] J. Wang, C. Shi, A. Sanabria-Borbon, E. Sánchez-Sinencio, and J. Hu, "Thwarting Analog IC Piracy Via Combinational Locking," *Proceedings of the IEEE International Test Conference (ITC)*, pp. 1-10, October 2017.
- [193] A. A. Saki and S. Ghosh, "How Multi-threshold Designs can Protect Analog IP," *Proceedings of the IEEE International Conference on Computer Design (ICCD)*, pp. 464-471, October 2018.
- [194] L. Pileggi, G. Keskin, X. Li, Ken Mai, and J. Proesel, "Mismatch Analysis and Statistical Design at 65 nm and Below," *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC)*, pp. 9-12, September 2008.
- [195] N. Shiono and C. Hashimoto, "Threshold-voltage Instability of n-Channel MOSFET's Under Bias-Temperature Aging," *IEEE Transactions on Electron Devices*, Vol. 29, No. 3, pp. 361-368, March 1982.
- [196] Y. Lu, K. Subramani, H. Huang, N. Kupp, and Y. Makris, "Silicon Demonstration of Statistical Post-Production Tuning," *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pp. 628-633, July 2015.
- [197] Y. Lu, G. Volanis, K. S. Subramani, A. Antonopoulos, and Y. Makris, "Knob Non-Idealities in Learning-Based Post-Production Tuning of Analog/RF ICs: Impact Remedies," *Proceedings of the IEEE VLSI Test Symposium (VTS)*, pp. 1-6, April 2017.
- [198] X. Li, B. Taylor, Y. Chien, and L. T. Pileggi, "Adaptive Post-silicon Tuning for Analog Circuits: Concept, Analysis and Optimization," *Proceedings of the IEEE/ACM International Conference on Computer-aided Design (ICCAD)*, pp. 450-457, November 2007.

- [199] P. B. Bacinski, T. Murgan, K. Koch, and M. Glesner, "An Analog On-chip Adaptive Body Bias Calibration for Reducing Mismatches in Transistor Pairs," *Proceedings of the Design, Automation and Test in Europe Conference (DATE)*, pp. 698–703, March 2008.
- [200] V. V. Rao, K. Juretus, and I. Savidis, "Security Vulnerabilities of Obfuscated Analog Circuits," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5, October 2020.
- [201] N. G. Jayasankaran, A. S. Borbon, A. Abuellil, E. Sánchez-Sinencio, J. Hu, and J. Rajendran, "Breaking Analog Locking Techniques," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 28, No. 10, pp. 2157-2170, July 2020.
- [202] R. Y. Acharya, S. Chowdhury, F. Ganji, and D. Forte, "Attack of the Genes: Finding Keys and Parameters of Locked Analog ICs Using Genetic Algorithm," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 284-294, December 2020.
- [203] K. Juretus and I. Savidis, "Reduced Overhead Gate Level Logic Encryption," *Proceedings of the IEEE/ACM International Great Lakes Symposium on VLSI (GLSVLSI)*, pp. 15-20, August 2016.
- [204] M. Yasin and O. Sinanoglu, "Evolution of Logic Locking," *Proceedings of the IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, pp. 1-6, 2017.
- [205] M. Yasin, J. Rajendran, O. Sinanoglu, and R. Karri, "On Improving the Security of Logic Locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 35, No. 9, pp. 1411-1424, December 2016.
- [206] S. E. Quadir, J. Chen, D. Forte, N. Asadizanjani, S. Shahbazmohamadi, L. Wang, J. Chandy, and M. Tehranipoor, "A Survey on Chip to System Reverse Engineering," Vol. 13, No. 1, pp. 1-34, December 2016.
- [207] F. Courbon, "Practical Partial Hardware Reverse Engineering Analysis," Vol. 4, No. 1, pp. 1-10, March 2020.

- [208] A. F. J. Levi and G. Aeppli, "The Naked Chip: No Trade Secret or Hardware Trojan Can Hide From Ptychographic X-ray Laminography," *IEEE Spectrum*, Vol. 59, No. 5, pp. 38-43, May 2022.
- [209] J. Kumagai, "Chip Detectives [Reverse Engineering]," *IEEE Spectrum*, Vol. 37, No. 11, pp. 43-48, November 2000.
- [210] H. Wang, D. Forte, M. Tehranipoor, and Q. Shi, "Probing Attacks on Integrated Circuits: Challenges and Research Opportunities," *IEEE Journal on Design and Test*, Vol. 34, No. 5, pp. 63-71, July 2017.
- [211] L. Lewyn and N. Williams, "Is a New Paradigm for Nanoscale Analog CMOS Design Needed?" *Proceedings of the IEEE*, Vol. 99, No. 1, pp. 3-6, January 2011.
- [212] X. Zhu, Y. Yang, and H. Liu, "Cell-based Wideband Variable-gain Amplifier With Accurate Gain Adjustment In 65 nm CMOS Technology," *Proceedings of the IEEE International Workshop on Electromagnetics: Applications and Student Innovation Competition (iWEM)*, pp. 1-3, 2016.
- [213] B. Razavi, "Operational Amplifier," *Design of Analog CMOS Integrated Circuits*. McGraw Hill Education, 2017, Chapter 9, pp. 344-409.
- [214] B. Razavi, "Phase-Locked Loops," *Design of Analog CMOS Integrated Circuits*. McGraw Hill Education, 2017, Chapter 16, pp. 651-690.
- [215] B. Sheu, D. Scharfetter, P.-K. Ko, and M.-C. Jeng, "BSIM: Berkeley short-channel IGFET model for MOS transistors," *Transactions of IEEE Journal of Solid-State Circuits*, Vol. 22, No. 4, pp. 558-566, August 1987.
- [216] J. R. Baker, "Operational Amplifier I," *CMOS Circuit Design, Layout, and Simulation*. Wiley, 2019, Chapter 24, pp. 803-856.
- [217] J. R. Baker, "Operational Amplifier II," *CMOS Circuit Design, Layout, and Simulation*. Wiley, 2019, Chapter 26, pp. 889-931.
- [218] V. K. Khanna, "Short Channel Effects in MOSFETs," *Integrated Nanoelectronics: Nanoscale CMOS, Post-CMOS and Allied Nanotechnologies*. Springer, 2016, Chapter 4, pp. 73-93.

- [219] K. E. Kroell and G. K. Ackermann, "Threshold Voltage of Narrow Channel Field Effect Transistors," *Solid State Electronics*, Vol. 19, No. 1, pp. 77-81, January 1976.
- [220] V. V. Rao, K. Juretus, and I. Savidis, "Hidden Costs of Analog Deobfuscation Attacks," *Proceedings of the IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 31, No. 11, pp. 1802-1815, September 2023.
- [221] Y. Zhong and U. Guin, "Complexity Analysis of the SAT Attack on Logic Locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 42, No. 10, pp. 3143-3156, January 2023.
- [222] V. V. Rao and I. Savidis, "Performance and Security Analysis of Parameter-Obfuscated Analog Circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 29, No. 12, pp. 2013-2026, October 2021.
- [223] P. Shrestha and I. Savidis, "Graph representation learning for parasitic impedance prediction of the interconnect," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5, May 2023.
- [224] M. I. LLP, "Global Analog Integrated Circuit (IC) Market - Growth, Trends, COVID-19 Impact, and Forecasts (2022 - 2027)," September 2022.
- [225] Z. Zhao and L. Zhang, "Deep Reinforcement Learning for Analog Circuit Sizing," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5, October 2020.
- [226] S. Engels, M. Hoffmann, and C. Paar, "A Critical View On The Real-world Security of Logic Locking," *Journal of Cryptographic Engineering*, Vol. 12, No. 3, pp. 229-244, August 2022.
- [227] J. Rührmair, U. and Sölter, "PUF Modeling Attacks: An Introduction and Overview," *Proceedings of the IEEE Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1-6, April 2014.