

# PARASITIC AWARE AUTOMATIC CMOS ANALOG CIRCUIT DESIGN USING EVOLUTIONARY ALGORITHMS

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

in

Electronics & Communication Engineering

by

Patel Subhash Jagadishchandra  
[119997111010]

under supervision of

Dr. Rajesh A Thakker



**GUJARAT TECHNOLOGICAL UNIVERSITY  
AHMEDABAD**

October-2018

# PARASITIC AWARE AUTOMATIC CMOS ANALOG CIRCUIT DESIGN USING EVOLUTIONARY ALGORITHMS

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

in

Electronics & Communication Engineering

by

Patel Subhash Jagadishchandra  
[119997111010]

under supervision of

Dr. Rajesh A Thakker



**GUJARAT TECHNOLOGICAL UNIVERSITY  
AHMEDABAD**

October-2018

**© Patel Subhash Jagadishchandra**

## DECLARATION

I declare that the thesis entitled “**Parasitic Aware Automatic CMOS Analog Circuit Design using Evolutionary Algorithms**” submitted by me for the degree of Doctor of Philosophy is the record of research work carried out by me during the period from **October-2011** to **October-2018** under the supervision **Dr. Rajesh A Thakker** of and this has not formed the basis for the award of any degree, diploma, associateship, fellowship, titles in this or any other University or other institution of higher learning.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis. I shall be solely responsible for any plagiarism or other irregularities, if noticed in the thesis.

Signature of the Research Scholar : ..... Date:.....

Name of Research Scholar: **Patel Subhash Jagadishchandra**

Place : **Ahmedabad**

## **CERTIFICATE**

I certify that the work incorporated in the thesis “**Parasitic Aware Automatic CMOS Analog Circuit Design using Evolutionary Algorithms**” submitted by **Shri Patel Subhash Jagadishchandra** was carried out by the candidate under my supervision/guidance. To the best of my knowledge: (i) the candidate has not submitted the same research work to any other institution for any degree/diploma, Associateship, Fellowship or other similar titles (ii) the thesis submitted is a record of original research work done by the Research Scholar during the period of study under my supervision, and (iii) the thesis represents independent research work on the part of the Research Scholar.

Signature of Supervisor: .....

Date: .....

Name of Supervisor: **Dr. Rajesh A Thakker**

Place: **Ahmedabad**

## Course-work Completion Certificate

This is to certify that **Mr. Patel Subhash Jagadishchandra** enrolment no. **119997111010** is a PhD scholar enrolled for PhD program in the branch **Electronics & Communication Engineering** of Gujarat Technological University, Ahmedabad.

**(Please tick the relevant option(s))**

- He/She has been exempted from the course-work (successfully completed during M.Phil Course)
- He/She has been exempted from Research Methodology Course only (successfully completed during M.Phil Course)
- He/She has successfully completed the PhD course work for the partial requirement for the award of PhD Degree. His/ Her performance in the course work is as follows-

Grade Obtained in Research Methodology (PH001)	Grade Obtained in Self Study Course (Core Subject) (PH002)
BB	BB

Supervisor's Sign

**Dr Rajesh A Thakker**

## Originality Report Certificate

It is certified that PhD Thesis “**Parasitic Aware Automatic CMOS Analog Circuit Design using Evolutionary Algorithms**” by **Patel Subhash Jagadishchandra** has been examined by us. We undertake the following:

- a. Thesis has significant new work / knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.
- b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.
- c. There is no fabrication of data or results which have been compiled / analysed.
- d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.
- e. The thesis has been checked using **Turnitin** (copy of originality report attached) and found within limits as per GTU Plagiarism Policy and instructions issued from time to time (i.e. permitted similarity index  $\leq 25\%$ ).

Signature of the Research Scholar: \_\_\_\_\_ Date: \_\_\_\_\_

Name of Research Scholar: **Patel Subhash Jagadishchandra**

Place: **Ahmedabad**

Signature of Supervisor: \_\_\_\_\_ Date: \_\_\_\_\_

Name of Supervisor: **Dr Rajesh A Thakker**

Place: **Ahmedabad**



## Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author: Subhash Patel  
Assignment title: Quick Submit  
Submission title: Thesis  
File name: Thesis-Subhash\_Patel.pdf  
File size: 1.43M  
Page count: 92  
Word count: 25,873  
Character count: 121,623  
Submission date: 04-Dec-2017 10:32PM (UTC+0530)  
Submission ID: 889895608

### Abstract

The performance of the analog integrated CMOS circuit is very sensitive to design parameters. Its design is a knowledge-intensive trade-off approach and requires lots of expertise. Thus, with increased complexity, the design of high-performance analog circuit becomes a very challenging task. In order to overcome difficulties associated with the analog integrated circuit design, many researchers have used classical and evolutionary algorithms to design analog circuits. However, the designs are limited to schematic-level only. The schematic-level design is an intermediate step of the circuit design process and, generally, based on schematic-level design, the layouts are prepared. Since the layout parasitics are not possible to consider during schematic-level design, the post-layout performance of the circuit differs from the schematic-level performance, especially, for the frequency sensitive specifications.

In this work, we propose a novel concept of the parasitic-aware automatic circuit design of analog CMOS circuits that extends the design automation from the schematic-level to layout-level. We have utilized evolutionary algorithms based optimization techniques. The PSO (Particle Swarm Optimization) and ABC (Artificial Bee Colony) algorithms are widely used evolutionary algorithms. Based on these algorithms, we propose two efficient evolutionary algorithms named MPSO (Modified Particle Swarm Optimization) algorithm and EABC (Enhanced Artificial Bee Colony) algorithm are proposed. The MPSO algorithm uses the partial re-initialization scheme to overcome the problem of diversity-loss in PSO algorithm. The EABC algorithm provides faster convergence speed compared to the ABC algorithm.

The MPSO, PSO, ABC and EABC algorithms are used to carry out the schematic-level design of the: (1) Two-stage operational amplifier (op-amp), (2) high-gain low voltage bulk-driven OTA, and (3) second generation current conveyor in the  $0.13\mu\text{m}$  and  $0.09\mu\text{m}$  CMOS technologies. The obtained results reveal the effectiveness of the proposed algo-

# Thesis

---

## ORIGINALITY REPORT

---

10%

SIMILARITY INDEX

4%

INTERNET SOURCES

9%

PUBLICATIONS

2%

STUDENT PAPERS

---

## PRIMARY SOURCES

---

- |   |   |     |
|---|---|-----|
| 1 | Chinchore, Jitendra B., and Rajesh A. Thakker. "Design of low dropout regulator using artificial bee colony evolutionary algorithm", 2015 International Conference on Circuits Power and Computing Technologies [ICCPCT-2015], 2015.<br>Publication | 1%  |
| 2 | <a href="http://www.philrutherford.com">www.philrutherford.com</a><br>Internet Source   | 1%  |
| 3 | H. Gossner. "ESD protection for the deep sub micron regime - a challenge for design methodology", 17th International Conference on VLSI Design Proceedings ICVD-04, 2004<br>Publication   | 1%  |
| 4 | Zhao, Xiao, Qisheng Zhang, and Ming Deng. "Low-voltage bulk-driven OTA with enhanced transconductance and dc gain", International Journal of Electronics Letters, 2015.<br>Publication  | <1% |
| 5 | Rajesh Amratlal Thakker. "Low-Power Low-Voltage Analog Circuit Design Using   | <1% |

**PhD THESIS Non-Exclusive License to  
GUJARAT TECHNOLOGICAL UNIVERSITY**

In consideration of being a PhD Research Scholar at GTU and in the interests of the facilitation of research at GTU and elsewhere, I, **Patel Subhash Jagadishchandra** having Enrollment No. **119997111010** hereby grant a non-exclusive, royalty free and perpetual license to GTU on the following terms:

- a) GTU is permitted to archive, reproduce and distribute my thesis, in whole or in part, and/or my abstract, in whole or in part (referred to collectively as the “Work”) anywhere in the world, for non-commercial purposes, in all forms of media;
- b) GTU is permitted to authorize, sub-lease, sub-contract or procure any of the acts mentioned in paragraph (a);
- c) GTU is authorized to submit the Work at any National / International Library, under the authority of their “Thesis Non-Exclusive License”;
- d) The Universal Copyright Notice (©) shall appear on all copies made under the authority of this license;
- e) I undertake to submit my thesis, through my University, to any Library and Archives. Any abstract submitted with the thesis will be considered to form part of the thesis.
- f) I represent that my thesis is my original work, does not infringe any rights of others, including privacy rights, and that I have the right to make the grant conferred by this non-exclusive license.
- g) If third party copyrighted material was included in my thesis for which, under the terms of the Copyright Act, written permission from the copyright owners is required, I have obtained such permission from the copyright

owners to do the acts mentioned in paragraph (a) above for the full term of copyright protection.

- h) I retain copyright ownership and moral rights in my thesis, and may deal with the copyright in my thesis, in any way consistent with rights granted by me to my University in this non-exclusive license.
- i) I further promise to inform any person to whom I may hereafter assign or license my copyright in my thesis of the rights granted by me to my University in this nonexclusive license.
- j) I am aware of and agree to accept the conditions and regulations of PhD including all policy matters related to authorship and plagiarism.

Signature of the Research Scholar:

Name of Research Scholar: **Patel Subhash Jagadishchandra**

Date:

Place: **Ahmedabad**

Signature of Supervisor:

Name of Supervisor: Dr Rajesh A Thakker

Date:

Place: **Ahmedabad**

Seal:

## Thesis Approval Form

The viva-voce of the PhD Thesis submitted by Shri Patel Subhash Jagadishchandra (Enrollment No. **119997111010** ) entitled “**Parasitic Aware Automatic CMOS Analog Circuit Design using Evolutionary Algorithms**” was conducted on **October 20, 2018** at Gujarat Technological University.

**(Please tick any one of the following option)**

- The performance of the candidate was satisfactory. We recommend that he/she be awarded the PhD degree.
- Any further modifications in research work recommended by the panel after 3 months from the date of first viva-voce upon request of the Supervisor or request of Independent Research Scholar after which viva-voce can be re-conducted by the same panel again.

(Briefly specify the modifications suggested by the panel)

- The performance of the candidate was unsatisfactory. We recommend that he/she should not be awarded the PhD degree.

(The panel must give justifications for rejecting the research work)

-----  
Name and Signature of Supervisor with Seal

-----  
1) (External Examiner 1) Name and Signature

-----  
2) (External Examiner 2) Name and Signature

-----  
3) (External Examiner 3) Name and Signature

# Abstract

The performance of the analog integrated CMOS circuits is very sensitive to design parameters. Its design is a knowledge-intensive trade-off approach and requires lots of expertise. Thus, with increased complexity, the design of high-performance analog circuit becomes a very challenging task. In order to overcome difficulties associated with the analog integrated circuit design, many researchers have used classical and evolutionary algorithms to design analog circuits. However, the designs are limited to schematic-level only. The schematic-level design is an intermediate step of the circuit design process and, generally, based on schematic-level design, the layouts are prepared. Since the layout parasitics are not possible to consider during schematic-level design, the post-layout performance of the circuit differs from the schematic-level performance, especially, for the frequency sensitive specifications.

In this work, we propose a novel concept of the parasitic-aware automatic circuit design of analog CMOS circuits that extends the design automation from the schematic-level to layout-level. We have utilized evolutionary algorithms based optimization techniques. The PSO (Particle Swarm Optimization) and ABC (Artificial Bee Colony) algorithms are widely used evolutionary algorithms. Based on these algorithms, we propose two efficient evolutionary algorithms named; MPSO (Modified Particle Swarm Optimization) algorithm and EABC (Enhanced Artificial Bee Colony) algorithm. The MPSO algorithm uses the partial re-initialization scheme to overcome the problem of diversity-loss in PSO algorithm. The EABC algorithm provides faster convergence speed compared to the ABC algorithm.

Initially, MPSO, PSO, ABC and EABC algorithms are used to carry out the schematic-level design of the: (1) Two-stage operational amplifier (op-amp), (2) high-gain low voltage bulk-driven OTA, and (3) second generation current conveyor in the  $0.13\ \mu\text{m}$  and  $0.09\ \mu\text{m}$  CMOS technologies. The obtained results reveal the effectiveness of the

proposed algorithms to design analog CMOS circuits.

The two-stage op-amp is designed at the schematic-level with major specifications such as 80dB gain, 100MHz bandwidth,  $60^\circ$  phase margin, 80dB PSRR, 75dB CMRR and  $40V/\mu S$  slew rate. The ABC algorithm designed two-stage op-amp in  $0.13\ \mu m$  CMOS technology with an average error of 1.01% while the EABC algorithm designed opamp with an average error of only 0.39%. The major design specifications for the OTA are 80dB gain, 60 phase margin, 1.5MHz bandwidth,  $5\mu W$  power consumption and  $0.1V/\mu S$  slew rate. The ABC algorithm designed high gain OTA with 0.69% average error, while MPSO algorithm designed the same circuit with 0% average error.

In order to investigate the effect of the layout parasitics on the performance on the circuit, layouts of circuits designed at schematics level were prepared and post-layout simulations were carried out. The post-layout simulation of the two stage op-amp indicated that, the performance of the op-amp is dropped by 6% due to the layout parasitics. The similar experiments are also carried out for the ring oscillator, CMOS buffer chain and, voltage controlled oscillator. The post-layout performance of the ring oscillator was degraded by 8%. For the CMOS buffer chain and VCO, the post-layout performances are degraded by 10.8% and 7.9%, respectively.

In order to overcome this problem and to achieve parasitic-aware automatic circuit design, the concept of the schematic-level automatic circuit design is extended to the layout-level by utilizing layout in the automatic design process. To achieve this, use of the configurable layouts is proposed instead of the conventional handcrafted layouts. Unlike conventional handcrafted layouts, the configurable layouts allow the user to change the size of various components of layout and the distances between them by changing the parameters of the layout. This feature of the configurable layout makes the preparation of layout flexible and suitable to use in automatic circuit design framework. The configurable layouts are developed using MAGIC VLSI Tool. By utilizing the configurable layouts in the automatic circuit design process, we achieve the parasitic-aware design of CMOS circuits. During the automatic circuit design process, extracted net-list generated from configurable layout is used. This enables the consideration of all kind of parasitics from the beginning of the circuit design process.

The automatic parasitic-aware design of the ring oscillator, CMOS buffer chain, VCO, two-stage op-amp, bulk-driven OTA, and enhanced bulk-driven OTA are carried out. The proposed concept does not require any human intervention and provides the parasitic-aware design of the high-performance circuit.

Further, it has also been demonstrated that process variations can also be considered in the design cycle. The two-stage op-amp and bulk-driven OTA are designed at layout-level considering the process variations. The obtained results reveal that using the proposed concept of parasitic-aware automatic circuit design even less-experienced designer can design high-performance analog circuits at the layout-level in an efficient manner.

# Acknowledgement

Firstly, I would like to express my sincere gratitude to my guide Dr. R A Thakker for the continuous and unconditional support of my Ph.D. study and related research, for his motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better guide and mentor for my Ph.D. study.

Besides my guide, I would like to thank the rest of my thesis committee: Dr. Mihir Shah, and Dr. Joycee Mekie, for their insightful comments and encouragement.

I am thankful to Dr, Akshai Aggarwal, Ex. Vice Chancellor, Dr. Navin Sheth, Vice Chancellor and, all staff members of Ph.D. Section, GTU.

I take this opportunity to thank Mr. Bhavin Gajjar, Mr. Abhishek Vaghela, Mr. Vrushank Shah and Dr. Minesh Thaker of Indus University for their support and timely help during the Ph.D. program.

I would like to thank my family: my parents, my sister and my son for supporting me spiritually throughout writing this thesis and my life in general.

# Contents

<b>List of Abbreviations</b>	<b>XIX</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem definition . . . . .	4
1.3 Objective and scope of work . . . . .	5
1.4 Contribution . . . . .	5
1.5 Organization of thesis . . . . .	6
<b>2 Evolutionary Algorithms</b>	<b>7</b>
2.1 Particle Swarm Optimization (PSO) algorithm . . . . .	8
2.2 Modified Particle Swarm Optimization (MPSO) algorithm . . . . .	9
2.3 Artificial Bee Colony Algorithm . . . . .	11
2.4 Enhanced artificial bee colony algorithm . . . . .	12
2.5 Benchmark functions . . . . .	14
2.6 Algorithm Parameters and Results . . . . .	14
<b>3 Automatic circuit design at schematic level</b>	<b>19</b>
3.1 Environment for schematic-level design automation . . . . .	19
3.2 Circuit design Examples . . . . .	21
3.2.1 Two-stage operational amplifier . . . . .	22
3.2.2 High-gain low-voltage bulk-driven OTA . . . . .	30
3.2.3 Second generation current conveyor (CCII+) . . . . .	35
3.3 Observations . . . . .	39
<b>4 Configurable layouts</b>	<b>40</b>
4.1 Design framework for configurable layouts . . . . .	40
4.2 Configurable layout of CMOS inverter . . . . .	45
4.3 Summary . . . . .	47

<b>5</b>	<b>Parasitic-aware automatic circuit design</b>	<b>48</b>
5.1	Environment for parasitic-aware automatic circuit design . . . . .	48
5.2	Ring-oscillator . . . . .	50
5.3	CMOS buffer chain . . . . .	55
5.4	Voltage controlled oscillator (VCO) . . . . .	57
5.5	Two-stage operational amplifier . . . . .	61
5.6	Bulk-driven OTA . . . . .	72
5.7	Enhanced bulk-driven OTA . . . . .	77
5.8	Observations . . . . .	80
<b>6</b>	<b>Conclusion</b>	<b>82</b>
	<b>APPENDICES</b>	<b>89</b>
<b>A</b>	<b>List of Publications</b>	<b>90</b>

# List of Abbreviations

<b>ABC</b>	Artificial Bee Colony Algorithm
<b>ACO</b>	Ant Colony Optimization Algorithm
<b>AMOSA</b>	Archived Multi-Objective Simulated Annealing
<b>BSIM</b>	Berkeley Short-channel IGFET Model
<b>CCII</b>	Second generation current conveyor
<b>CMOS</b>	Complementary Metal-Oxide-Semiconductor
<b>EABC</b>	Enhanced Artificial Bee Colony Algorithm
<b>GA</b>	Genetic Algorithm
<b>GABC</b>	G-best guided Artificial Bee Colony Algorithm
<b>LNA</b>	Low Noise Amplifier
<b>MABC</b>	Modified Artificial Bee Colony Algorithm
<b>MOACO</b>	Multi-Objective Ant Colony Algorithm
<b>MOPSO</b>	Multi-Objective Particle Swarm Optimization Algorithm
<b>MOS</b>	Metal-oxide-semiconductor
<b>MPSO</b>	Modified Particle swarm optimization Algorithm
<b>NMOS</b>	N-Channel Metal-oxide-semiconductor
<b>NSGA-II</b>	Non-dominated Sorting genetic algorithm
<b>OTA</b>	Operational transconductance amplifier
<b>PMOS</b>	P-Channel Metal-oxide-semiconductor
<b>PSO</b>	Particle swarm optimization Algorithm
<b>RMS</b>	Root Mean Square
<b>SA</b>	simulated annealing
<b>VCO</b>	Voltage Controlled Oscillator

# List of Figures

1.1	Analog integrated circuit design process flow. . . . .	2
3.1	Conceptual block diagram of optimizer for automatic CMOS circuit design at schematic-level. . . . .	20
3.2	Two-stage operational amplifier . . . . .	22
3.3	Two stage operational amplifier (Schematic-level): Variations in RMS error with circuit evaluations . . . . .	26
3.4	High-gain Low-voltage bulk-driven OTA . . . . .	30
3.5	High-gain low-voltage bulk-driven OTA(Schematic-level): Variations in RMS error with circuit evaluations . . . . .	33
3.6	Second generation current conveyor (CCII+) . . . . .	35
3.7	Second generation current conveyor(Schematic-level): Variations in RMS error with circuit evaluations . . . . .	38
4.1	User interface for configurable layout . . . . .	41
4.2	Configurable Layout. . . . .	41
4.3	Layout of two contacts connected with metal layer. The distance between contacts : (A) $15\mu m$ (B) $18\mu m$ (C) $20\mu m$ . . . . .	42
4.4	Script for layouts of two contacts connected with each other using metal layer . . . . .	42
4.5	Layout of NMOS transistor with gate and drain connected to metal wires. . . . .	44
4.6	Layout of inverters. . . . .	45
4.7	Layout of inverters with different parameters. . . . .	46
5.1	Conceptual block diagram of optimizer for the parasitic-aware automatic CMOS circuit design. . . . .	49
5.2	Circuit of Ring-oscillator. . . . .	51
5.3	Ring-oscillator: Hierarchy used in configurable layout. . . . .	51
5.4	Layout of ring-oscillator. . . . .	52

## List of Figures

5.5	Ring-oscillator: Output waveform. . . . .	53
5.6	Ring-oscillator: Variations in RMS error with circuit evaluations. . . . .	53
5.7	Ring-oscillator with finger transistors (parasitic-aware design): Layout of best design obtained from MPSO algorithm. . . . .	54
5.8	Circuit of CMOS buffer chain. . . . .	55
5.9	CMOS buffer chain: Variations in average RMS error with circuit evaluations. . . . .	57
5.10	CMOS buffer chain: Layout of best design obtained from EABC algorithm. . . . .	58
5.11	Circuit of VCO: (A) Lee-Kim delay cell (B) Duty-cycle corrector (C) VCO. . . . .	58
5.12	VCO: Hierarchy used in configurable layout. . . . .	59
5.13	VCO: Layout of best design obtained from PSO algorithm in $0.13\ \mu m$ CMOS technology. . . . .	61
5.14	VCO: Layout of best design obtained from PSO algorithm in $0.35\ \mu m$ CMOS technology. . . . .	62
5.15	Circuit of two-stage operational amplifier. . . . .	62
5.16	Two stage op-amp in $0.13\ \mu m$ technology: Variations in average RMS error with circuit evaluations. . . . .	65
5.17	Two stage op-amp in $0.35\ \mu m$ technology: Variations in average RMS error with circuit evaluations. . . . .	65
5.18	Two-stage op-amp in $0.13\ \mu m$ technology: Layout of best design obtained from MPSO algorithm. . . . .	66
5.19	Two-stage op-amp: Layout scheme for stacked capacitor. . . . .	66
5.20	Two-stage op-amp in $0.35\ \mu m$ technology: Layout of best design obtained from EABC algorithm. . . . .	67
5.21	Circuit of bulk-driven OTA. . . . .	72
5.22	Layout of PMOS with with input at bulk-terminal . . . . .	73
5.23	Bulk-driven OTA (parasitic-aware design): Layout of best design obtained from MPSO algorithm. . . . .	75
5.24	Circuit of enhanced bulk-driven OTA. . . . .	77
5.25	Enhanced bulk-driven OTA (parasitic-aware design): Variations in average RMS error with circuit evaluations. . . . .	79
5.26	Enhanced bulk-driven OTA (parasitic-aware design): Layout of best design obtained from EABC algorithm. . . . .	80

# List of Tables

2.1	Benchmark functions used for the experiment. . . . .	14
2.2	Results of solution of 30-dimensional benchmark functions with 30 independent runs. . . . .	15
2.3	Results of solution of 60-dimensional benchmark functions with 30 independent runs. . . . .	17
3.1	Two stage operational amplifier: search space for design variables. . . .	23
3.2	Two stage operational amplifier: Illustration for calculation of cost function. . . . .	24
3.3	Two stage op-amp: Average of obtained simulation results over 10 independent design trials. . . . .	25
3.4	Two stage operational amplifier (Schematic-level): Performance comparison of algorithms based on 10 independent design trials. . . . .	25
3.5	Two stage op-amp: Results of experiment to achieve best possible design in 0.13 $\mu m$ technology. . . . .	27
3.6	Two stage operational amplifier: The performance measure of the obtained solution at TT, FF, FS, SF and SS process corner obtained by ABC, EABC and MPSO algorithms in 0.13 $\mu m$ technology. . . . .	28
3.7	Two stage operational amplifier: The performance measure of the obtained solution at TT, FF, FS, SF and SS process corner obtained by ABC, EABC and MPSO algorithms in 0.09 $\mu m$ technology. . . . .	28
3.8	Two stage op-amp design considering process variations: values of obtained design parameters. . . . .	29
3.9	High gain low voltage bulk-driven OTA: Search space for design variables.	32
3.10	High-gain low-voltage bulk-driven OTA: Average of obtained simulation results over 10 independent design trials. . . . .	32
3.11	High gain low voltage bulk-driven OTA: Performance comparison of algorithms based on 10 independent design trials. . . . .	34

3.12	Low voltage CCII: Search space for design variables. . . . .	36
3.13	High-gain low-voltage bulk-driven OTA: Average of obtained simulation results over 10 independent design trials. . . . .	37
3.14	Low voltage CCII: Performance comparison of algorithms based on 10 independent design trials. . . . .	37
4.1	Basic macros for configurable layout . . . . .	43
4.2	Some of the technology dependent variables . . . . .	43
4.3	Parameters of configurable inverter layout . . . . .	46
5.1	Ring-oscillator: Search-space for design parameters. . . . .	51
5.2	Ring-oscillator: Average of the obtained results. . . . .	52
5.3	Ring-oscillator with finger transistors: Average of the obtained results. . . . .	54
5.4	CMOS buffer chain: Search-space for design parameters. . . . .	56
5.5	CMOS buffer chain (schematic-level design using ABC algorithm): Average of results from schematic-level simulations and corresponding post-layout simulation. . . . .	56
5.6	CMOS buffer chain (parasitic-aware design): Average of simulation results. . . . .	57
5.7	CMOS buffer chain: Design parameters for best design run. . . . .	58
5.8	VCO: Search-space for design parameters. . . . .	59
5.9	VCO (schematic-level design using ABC algorithm): Average of results from schematic-level simulations and corresponding post-layout simulation. . . . .	60
5.10	VCO (parasitic-aware design): Average of simulation results. . . . .	60
5.11	VCO: Design parameters for best design trial . . . . .	60
5.12	Two stage op-amp: Search-space for design parameters. . . . .	63
5.13	Two stage op-amp (schematic-level design using ABC algorithm): Average of results from schematic-level simulations and corresponding post-layout simulation. . . . .	63
5.14	Two stage op-amp (parasitic-aware design): Average of simulation results. . . . .	64
5.15	Two stage op-amp (parasitic-aware design): Design parameters for best design run . . . . .	66
5.16	Two stage op-amp design (parasitic-aware) using ABC algorithm considering process and temperature variations . . . . .	69
5.17	Two stage op-amp design (parasitic-aware) using MPSO algorithm considering process and temperature variations . . . . .	69
5.18	Two stage op-amp design (parasitic-aware) using EABC algorithm considering process and temperature variations . . . . .	70
5.19	The design parameters for two stage op-amp design (parasitic-aware) considering process and temperature variations . . . . .	70

5.20 Bulk-driven OTA (parasitic-aware design): Search-space for design parameters. . . . .	73
5.21 Bulk-driven OTA (parasitic-aware design): Average of simulation results.	74
5.22 Bulk-driven OTA (parasitic-aware): Design parameters for best design run	74
5.23 Bulk-driven OTA (parasitic-aware design): The performance measure of the obtained solution at TT, FF, FS, SF and SS process corner obtained by ABC, EABC, PSO and MPSO algorithms in $0.13 \mu m$ technology. . .	76
5.24 The design parameters for bulk-driven OTA (parasitic-aware) considering process variations . . . . .	77
5.25 Enhanced bulk-driven OTA (parasitic-aware design): Search-space for design parameters. . . . .	78
5.26 Enhanced bulk-driven OTA (parasitic-aware design): Average of simulation results. . . . .	79
5.27 Enhanced bulk-driven OTA: Design parameters for best design run . .	79

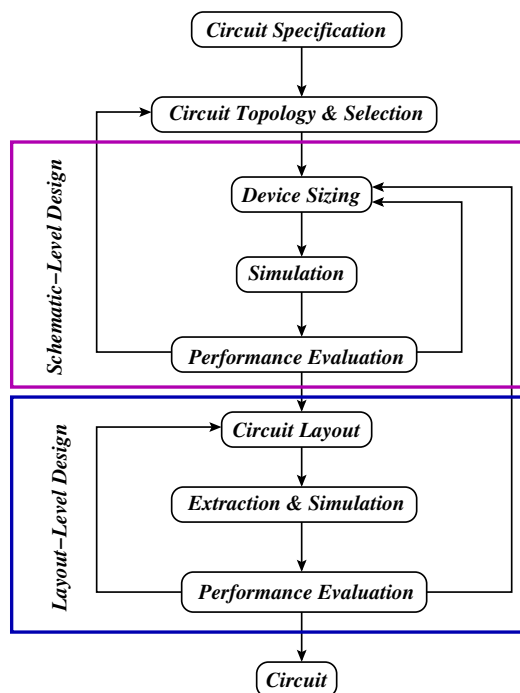
# CHAPTER 1

## Introduction

In recent years, with the increase of complexity, the design of the analog and mixed signal CMOS integrated circuits in an efficient manner is a critical task. In order to handle the complexity of integrated circuits, hierarchical blocks oriented design approach is used widely. For the analog CMOS circuits, such hierarchical blocks are generally amplifiers, filters, references, mixers, and oscillators. The design of these building blocks includes sizing of various MOSFETs to achieve target design specifications such as power consumption, gain, bandwidth, slew rate. Since the performance of such analog circuits is very sensitive to design parameters, their design requires lots of expertise. Further, improvement in one performance parameter may adversely affect the other performance parameters. Thus, the analog circuit design process is a knowledge intensive trade-off approach [1].

### 1.1 Background

The traditional analog circuit design is carried out in three steps as shown in Fig. 1.1: (1) Topology selection, (2) Component sizing i.e. schematic-level design, and (3) Layout-design and verification i.e. layout-level design [2]. In the topology selection, an appropriate topology of the circuit is selected to meet desired specifications. During the schematic-level design, generally, analytical calculations are followed by the circuit simulations. Finally, based on the schematic-level design, the layout of the circuit is prepared. The parasitics are extracted and the circuit is simulated to verify its



**FIGURE 1.1:** Analog integrated circuit design process flow.

performance. If desired performance is not achieved then the modifications in schematic and layout are carried out.

In the schematic level design, to determine the length and width of the various MOSFETs, the behavior of the circuit is described by the set of mathematical equations. Based on this mathematical model, analytical calculations are carried out. This requires an accurate mathematical model of the MOSFET. Generally, BSIM spice models are widely used for describing the behavior of the MOSFET. The level-2 spice model of MOSFET has 13 parameters, and it is suitable to represent large-channel MOSFET (i.e.  $L \geq 2\mu m$ ). The level-28 model has 63 parameters. Similarly, the level-54 BSIM model, which is suitable for describing the MOSFET in  $0.13\mu m$  technology, has more than 150 parameters. This way, the BSIM models have become more complex to accommodate various short channel and other nonlinear effects of the devices. It is very difficult to consider such complex models in the analytical calculations of the circuit design process. This leads to the considerable difference in the result of analytical calculations and simulation. Thus, the dependence on the circuit simulation is increased and the design of analog circuit requires expertise.

In order to overcome the difficulties associated with the analog circuit design, the research community is working aggressively on the automation of the analog circuit design process. The various classical optimization techniques, such as, linear programming [3], dynamic programming [4], sequential quadratic programming [5] are utilized for the

## 1.1 Background

schematic-level design of analog circuits. However, they suffer from following problems. First, the quality of the solution depends upon the starting point. In the absence of good starting point, such algorithms may result in a locally optimized solution. Second, it is necessary to formulate accurate and differentiable objective function. Formulating such cost function for the analog circuit design problem is extremely difficult. Further, for the dynamic programming, computational efforts increase exponentially with the size of the problem [6]. The geometric programming technique is used to design operational amplifier circuit [1, 7] and latched comparator [8]. This convex optimization technique guarantees the global solution; however, it requires very good knowledge of the circuit design and requires physical models to prepare constraints. When the short-channel effects of MOSFET are not negligible, it is very difficult to prepare such physical models [9]. Thus, for the design of the large size analog CMOS ICs in submicron CMOS technologies, which are complicated and highly constrained, the classical optimization techniques and geometric programming are mostly not appropriate. Another option is a heuristic algorithm based optimization technique. Some examples of the heuristic algorithms are simulated annealing (SA) [10], Tabu search [11], Genetic algorithm (GA) [12], Particle Swarm Optimization (PSO) [13], Ant Colony Optimization (ACO) [14] and Artificial Bee Colony algorithm (ABC) [15]. These algorithms are easy to implement using the programming language. They do not include complex mathematical operations and do not require deep knowledge of circuit as well as of the physical model of MOSFET.

Many researchers have successfully employed evolutionary algorithms to design analog circuits at schematic-level. The design of the single-ended telescopic operational amplifier using the genetic algorithm is presented in [16]. The chaotic DE algorithm, standard DE algorithm, ABC algorithm and PSO algorithm are used in [17] to design Miller OTA and their performances are compared. The use of the A-NSGAI algorithm was demonstrated in [18] to design RF low noise amplifier, leapfrog filter, and ultra wideband LNA. The PSO algorithm and its variants are used for automatic design of low-power low-voltage CMOS circuits [9]. In [19], the performances of the genetic algorithm (GA), PSO algorithm and Simulated Annealing algorithm are compared by designing the LC voltage controlled oscillator. An evolutionary algorithm based approach is used to design RF low noise amplifier in [20]. The particle swarm optimization algorithm with aging leader and challenger (ALC-PSO) is utilized to design CMOS comparator and folded cascode op-amp [21]. The CMOS differential amplifier and two-stage op-amp are designed in [22] using the integration of the Gravitational search algorithm with particle swarm optimization (GSA-PSO). The optimum design of the symmetric CMOS inverter is demonstrated in [23] utilizing hybrid Harmony Search with Differential Evolution (HS-DE) and PSO algorithm. In [24], PSO algo-

rithm is used to design low noise amplifier (LNA) and second generation current conveyor (CCII). The performances of the Multi-Objective Particle Swarm Optimization (MOPSO), Multi-Objective Ant Colony Algorithm (MOACO), Non-dominated Sorting genetic algorithm (NSGA-II) and Archived Multi-Objective Simulated Annealing (AMOSAs) are compared in [25] by designing LNA, op-amp, OTA, and current conveyor circuits.

In the layout-level design phase, based on the device dimensions obtained from the schematic-level design, layouts are prepared. Due to the high irregularity among the device dimensions, the analog circuit has not been automated like the digital circuits [26]. Since the layout adds parasitic components to the circuit, there is a difference in the simulation of schematic-level design and layout-level design, especially, in frequency sensitive specifications such as bandwidth and slew rate. The unacceptable difference in simulation results may lead redesigning of the circuit. Thus, the efficient approach for the designing analog CMOS circuit is required to overcome these difficulties. Further, the concept of the automatic CMOS analog design is only limited to schematic-level and not yet applied to layout-level to include the layout-parasitics in the design process.

## 1.2 Problem definition

The aim of this research work is to develop a time-efficient method for the design of the high-performance analog CMOS circuits at layout-level. Many researchers have successfully applied optimization algorithms to efficiently design CMOS analog circuits such as comparator, operational amplifier, differential amplifier, inverter, low noise amplifier, current conveyor, and OTA. However, automation in the designs are limited to schematic-level. The schematic-level design cannot consider the final design because in this design process, layout parasitics are not considered and the exact value of the layout parasitics cannot be estimated. The irregularities in the device dimensions prevent the use of the automatic placement and routing tools for preparing the layouts of analog circuits, and hence, the layouts for analog circuits are prepared manually. Sometimes, the small change in such layouts may require large efforts. Due to such limitations, the traditional design approach fails to provide a time-efficient design for the analog circuits.

To overcome this problem, the concept of the configurable layouts is proposed. The configurable layouts allow the user to change various parameters of the layout. This flexibility allows the use of the optimization algorithms to carry out layout-level design.

### 1.3 Objective and scope of work

Thus, the use of the configurable layouts enables the consideration of all kinds of parasitics from the beginning of the circuit design process and allows parasitic-aware design automation for the analog CMOS circuits.

Further, we propose improved evolutionary algorithms namely modified particle swarm optimization (MPSO) algorithm and enhanced artificial bee colony (EABC) algorithm for the analog CMOS circuit design problem.

## 1.3 Objective and scope of work

There are two major objectives of this work. The first objective is to propose an approach for the parasitic-aware layout-level design automation for the analog integrated circuits. The second objective is to propose efficient evolutionary optimization algorithms for the analog circuit design.

The scope of the work includes the following tasks. The first task is to develop the efficient evolutionary algorithm and compare its performance with other evolutionary algorithms by solving the set of benchmark functions and designing analog CMOS circuits. The second task is to develop a framework for the parasitic-aware analog CMOS circuit design. The third task includes the demonstration of a parasitic-aware design of analog CMOS circuits using the proposed concept.

## 1.4 Contribution

The following is the contribution of this work.

- Two efficient evolutionary algorithms: (1) EABC (Enhanced Artificial Bee Colony) algorithm, and (2) MPSO (Modified particle swarm optimization) are proposed for the CMOS circuit design and optimization.
- Three circuits namely: (1) Two-stage operational amplifier, (2) High gain bulk-driven OTA, and (3) Second generation current conveyor are designed at schematic level using MPSO, PSO, ABC, and EABC algorithms in  $0.13\ \mu m$  and  $0.09\ \mu m$  CMOS technologies.

- The concept of the automatic circuit design is extended from schematic-level to layout-level by proposing configurable layouts.
- The framework for the configurable layout is developed using MAGIC VLSI Tool.
- The automatic parasitic-aware designs of the ring oscillator, CMOS buffer chain, VCO, two-stage operational amplifier, bulk-driven OTA, and Enhanced bulk-driven OTA are carried out in  $0.13\mu m$  CMOS technology at layout-level using proposed concept.
- The process and temperature variations are also considered in the automatic design of the two-stage operational amplifier.

## 1.5 Organization of thesis

The rest of the thesis is organized as follows. In chapter 2, the evolutionary algorithms such as ABC, EABC, PSO and, MPSO are discussed in detail and their performances based on the ability to solve various benchmark functions are compared. The concept of the schematic-level analog circuit design using evolutionary algorithms is described in chapter 3. Further, the schematic-level design of two-stage op-amp, high-gain low-voltage OTA and, second-generation current-conveyor are presented. In chapter 4, the concept of the configurable layout is discussed in detail. In chapter 5, parasitic-aware design of ring oscillator, CMOS buffer chain, VCO, two-stage op-amp, bulk-driven OTA and, enhanced bulk-driven OTA are presented. Finally, the conclusions are drawn in chapter 6.

## CHAPTER 2

# Evolutionary Algorithms

The optimization algorithm is a heart of the optimization process. In this work, particle swarm optimization (PSO) and artificial bee colony (ABC) algorithms are utilized. Further, based on these algorithms, modified particle swarm optimization (MPSO) and enhanced artificial bee colony (EABC) algorithms are proposed.

The PSO algorithm is widely used for solving various engineering problems. It is simple to implement and very effective. The convergence speed of the PSO algorithm is fast, however, it suffers from the saturation problem and likely to trap in locally optimized solution [27]. In the MPSO algorithm, this issue of the PSO algorithm is addressed.

The performance comparison of ABC, PSO, DE and swarm inspired evolution algorithm (PS-EA) algorithms based on their ability to solve benchmark functions reveals that the performance of the ABC algorithm is superior with the fewer number of control parameters [15,28]. The detailed survey on the applications of the ABC algorithm [29] shows that ABC algorithm is an effective tool to solve the various engineering problem. For any population-based algorithm, the balance between exploration and exploitation is necessary. The exploration is an ability of an algorithm to investigate the various unknown region of the search space, while exploitation is the process of visiting those regions of a search space within the neighborhood of previously visited points [30]. It has been found that ABC algorithm has good exploitation capacity, however, it lacks at exploration [31]. Because of this, the ABC algorithm exhibits slower convergence speed when handling unimodal problems and it gets trapped in local minima while solving complex multi-modal problems [32]. The various modified and hybrid ABC

algorithms are presented in the literature to overcome this issue [32–35]. In this work, the modified version of the ABC algorithm named enhance artificial bee colony algorithm (EABC) is presented. The modified search strategy adapted in EABC algorithm enhances the overall performance of the algorithm.

In this chapter, evolutionary algorithms such as Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Modified Particle Swarm Optimization (MPSO) and, Enhanced Artificial Bee Colony (EABC) are presented. The performances of these algorithms are compared solving the set of benchmark functions.

## 2.1 Particle Swarm Optimization (PSO) algorithm

The particle swarm optimization algorithm was suggested by Kennedy and Eberhart in 1995 [13]. It is based on the swarm optimization technique. It simulates the social behavior of birds flocking and fish schooling. The PSO algorithm models solution candidates by particles of a swarm. Each particle of the swarm is moving in the search space with a certain velocity. Each particle has information about its current position, its current velocity, best position ever visited by itself and, best position visited among all the particles. Based on these informations the position and velocity of each particle are calculated.

Let's assume that, the swarm of  $N$  particles is considered to solve the optimization problem having  $D$  dimensions. The position of each particle represents a solution candidate. The vector of  $D$  dimensions can be used to model the position of the particle mathematically. The current position of the  $i^{th}$  particle can be represented by vector  $x_i(t) = \{x_{i1}, x_{i2}, \dots, x_{iD}\}$ . Similarly, the current velocity associated with the  $i^{th}$  particle can be given by vector  $v_i(t) = \{v_{i1}, v_{i2}, \dots, v_{iD}\}$ . The new velocity  $v_i(t+1)$  and position  $x_i(t+1)$  of the  $i^{th}$  particle can be calculated as follows,

$$v_i(t+1) = w \cdot v_i(t) + C_1 \cdot R_1 (P_i(t) - x_i(t)) + C_2 \cdot R_2 (P_g(t) - x_i(t)) \quad (2.1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1)\Delta t \quad (2.2)$$

where,  $w$  is an inertia weight factor,  $R_1$  and  $R_2$  are random numbers between 0 and 1,  $C_1$  and  $C_2$  are acceleration constants.  $C_1$  is known as the cognitive coefficient and  $C_2$  is called the social coefficient.  $P_i(t)$  represents the best position ever visited by the  $i^{th}$  particle.  $P_g(t)$  indicates the best position ever visited among all the particles. The value of  $\Delta t$  is 1.

## 2.2 Modified Particle Swarm Optimization (MPSO) algorithm

The  $C_1 \cdot R_1 (P_i(t) - x_i(t))$  represents the cognitive component. This component causes the particle to move near the best position ever visited by itself. The social component  $C_2 \cdot R_2 (P_g(t) - x_i(t))$  tends to pull particle near position, which is represented by the best solution found so far by the swarm. The random variables  $R_1$  and  $R_2$  are responsible for causing stochastic influence of cognitive and social component on the velocity update.

The inertia constant  $w$  controls the influence of the previous velocity on the new velocity. The value of  $w$  can be kept constant or can be varied with the algorithmic iterations. The value of  $w$  can be varied linearly with algorithmic iterations as follows,

$$w = w_{up} - (w_{up} - w_{low}) \cdot \frac{i_c}{i_{max}} \quad (2.3)$$

where,  $w_{up}$  and  $w_{low}$  are upper and lower limits for the  $w$ , respectively, with  $w_{up} > w_{low}$ ;  $i_c$  is current algorithm iteration;  $i_{max}$  represents maximum allowed algorithm iterations. Further, along with bounded search-space, the concept of the velocity clamping is used widely in PSO algorithm. The velocity clamping limits velocity of the particle to a limited range of  $[-v_{max}, v_{min}]$ . This helps preventing swarm explosion and controlling global exploration [36]. In this work, we use  $v_{max} = (x_{max} - x_{min}) / 2$ , where  $x_{max}$  and  $x_{min}$  represent the boundaries of the search space.

The important controlling parameters of the PSO algorithm are  $w$ ,  $C_1$  and  $C_2$ . The values of these parameters can be tuned according to the nature of the problem. The PSO algorithm starts with the particle initialization process. When the nature of the problem is unknown, random initialization can be used. Similarly, the initial velocity of the particle is also selected randomly. In each algorithmic iteration, the new position of each particle is calculated and each particle is evaluated for its fitness.

## 2.2 Modified Particle Swarm Optimization (MPSO) algorithm

The PSO algorithm is used widely for the function optimization due to its simplicity and easy implementation. When PSO algorithm starts with the random initialization, the swarm is diverse. In the PSO algorithm, the new position of the particle depends upon; (1)  $P_g$  (best solution found so far by the swarm of particles), (2)  $P_i$  (best solution found so far by the particle) and, (3) the velocity of the particle. Since the each particle is following  $P_g$ , as algorithm goes ahead, it starts losing its diversity [37]. The diversity

of the swarm is very important in the PSO algorithm. As long as the swarm is diverse, new solutions will be generated. When the swarm lost its diversity, the movement of the particles in the search space vanishes, in other words, the algorithm stops providing new solutions. This situation is known as saturation of the algorithm. The situation of loss of diversity arises either when the global solution is achieved, or when the algorithm is trapped in the local minima. In Modified PSO algorithm, the aim is to maintain the diversity of swarm till the global solution is achieved.

In the MPSO algorithm, like a PSO algorithm, the new position of the particle is decided by Eq 2.1 and Eq 2.2. However, when the global solution does not improve after the algorithmic iteration, the velocity of each particle is examined. If the absolute value of the velocity of each particle of the swarm in more than  $DS_{max}$  dimensions (a control parameter for algorithm) falls below the  $V_{min}$  ( a predetermined algorithmic parameter), one of the saturated dimensions is selected randomly and partial reinitialization for the selected dimension is carried out. In the partial reinitialization, the selected dimension of all particles of the swarm is reinitialized on the reduced search space. The reduced search space is a sub-region of search space around the best solution found so far. The region (or area) for the reduced search space is described in the percentage of full search space using following formula.

$$A = S + (1 - S) \cdot fgb \times 100\% \quad (2.4)$$

where  $S$  is a number in the range  $[0,1]$  and it is a control parameter of the algorithm.  $fgb$  is the value of the cost function (a function that describes the optimization problem mathematically) calculated for the best solution found so far ( $P_g$ ). The value of the  $V_{min}$  can be described in the percentage of the allowed maximum velocity  $V_{max}$  which is used for the velocity clamping.  $DS_{max}$  is an integer number and it is also described in terms of the percentage of the  $D$  (the dimension of the problem). The reinitialization region during the process of the partial reinitialization is controlled by parameter  $S$ .

The partial reinitialization scheme helps to avoid the early saturation of the algorithm by maintaining the diversity of the swarm and, thus, helps algorithm to avoid local minima.

## 2.3 Artificial Bee Colony Algorithm

The artificial bee colony algorithm simulates the social behavior of bees searching food to find the solution of the optimization problem. The algorithm is suggested by D Karaboga in 2005. This algorithm is based on the swarm optimization technique. The details about the algorithm, its applications solving various optimization problems and comparison with other algorithms can be found in [15, 28, 38].

In the ABC algorithm, the swarm consists of three types of bees; (1) employee bees, (2) onlooker bees and, (3) scout bee. The employee bees and onlooker bees are same in numbers. The employee bee tries to find a new solution by carrying out a local search. The onlooker bees select the solutions having higher chances of improvement and only try to improve these selected solution candidates.

Let's consider an optimization problem with D dimensions. We assume that there are N number of employee bees and the same number of the onlooker bees in the swarm. In the ABC algorithm, the solution candidate is represented by food-source. Let's assume that there are N number of food-sources. The food-source can be modeled mathematically modeled using a vector of D dimension i.e.  $i^{th}$  food-source can be represented by,  $x_i = \{x_{i1}, x_{i1}, \dots, x_{iD}\}$ .

The ABC algorithm has four major phases; (1) initialization (2) employee bee phase (3) onlooker bee phase and, (4) scout bee phase. In the initialization phase, random initialization of the food-sources is carried out. In the employee bee phase, each employee bee tries to find a better food-source around existing one. To do this, each employee bee shares information about the food-source to neighbor bee. For example, a new food-source  $v_i$  around the  $i^{th}$  food-source  $x_i$  can be found by getting influence about  $k^{th}$  food-source in the following manner,

$$v_{ij} = \begin{cases} x_{ij}, & \text{if } j \neq p \\ x_{ij} + \phi \cdot (x_{ij} - x_{kj}) & \text{if } j = p \end{cases} \quad (2.5)$$

where,  $i = 1, 2, \dots, N$ ;  $j = 1, 2, \dots, D$ ;  $\phi$  is a random number in range  $[-1, 1]$ ;  $k$  represents the food-source of neighbor bee, it is randomly selected integer in range  $[1, N]$ , with  $i \neq k$ ;  $p$  is randomly selected number from range  $[1, D]$ .

The fitness  $fit_{v_i}$  of newly found food-source  $V_i$  is calculated as follows,

$$fit_{x_i} = \frac{1}{1 + f(x_i)} \quad (2.6)$$

where,  $f(x_i)$  represents the cost function representing optimization problem. This is followed by the greedy selection process between  $v_i$  and  $x_i$  i.e. If  $fit_{v_i}$  is greater than  $fit_{x_i}$ , new food-source  $v_i$  is accepted otherwise, it is rejected and  $x_i$  is retained.

In the onlooker bee phase, selected food-sources having the better probability of improvement are selected. The probability of improvement  $P_i$  associated with food-source  $x_i$  can be calculated in the following manner,

$$P_i = \frac{fit_{x_i}}{\sum_{i=1}^{i=N} fit_{x_i}} \quad (2.7)$$

The selected food-sources are updated according to Eq. 2.5 similarly like in employee bee phase. In scout bee phase, the food-sources which are not improved after certain pre-determined trials ( $T_{limit}$ ) are discarded and for each discarded food-source, new food-source is randomly initialized from the search-space. Thus, in the ABC algorithm, the employee bees try to improve each food-source equally, the onlooker bees only focus on the food-sources where the probabilities of improvements are higher.

## 2.4 Enhanced artificial bee colony algorithm

In the ABC algorithm, the new food-source is found around existing food-source based on the neighbor food-source. This provides exploitation. However, ABC algorithm does not utilize the best solution found so far. Due to this limitation, sometimes, the ABC algorithm exhibits poor convergence speed. The enhanced artificial bee colony (EABC) algorithm addresses this issue of the ABC algorithm.

Like the ABC algorithm, EABC algorithm has food-sources, employee bees, onlooker bees and scout bee. Further, EABC, algorithm has initialization phase, employee bee phase, onlooker bee phase and scout bee phase. The search strategy adopted in EABC algorithm is different. In order to improve the convergence speed during employee bee phase, new food-source  $v_i$  is found in the following manner,

## 2.4 Enhanced artificial bee colony algorithm

$$V_{ij} = \begin{cases} X_{ij}, & \text{if } j \neq p \\ A, & \text{if } j = p \text{ and } fit_{N_1} > fit_{N_2} \\ B, & \text{if } j = p \text{ and } fit_{N_1} < fit_{N_2} \end{cases} \quad (2.8)$$

$$A = X_{N_1j} + \psi \cdot (2X_{N_1j} - X_{N_2j} - GBest_j)$$

$$B = X_{N_2j} + \psi \cdot (2X_{N_2j} - X_{N_1j} - GBest_j)$$

where,  $i = 1, 2, \dots, SN$ ;  $j = 1, 2, \dots, D$ ;  $\psi$  is a random number in range  $[-0.5, 0.5]$ ;  $p$  is a random integer selected from range  $[1, D]$ ;  $N_1$  and  $N_2$  are mutually exclusive random integers selected from range  $[1, D]$ ;  $GBest$  is a best solution ( $GBest$ ) found by algorithm so far. The new food source  $V_i$  is found using two mutually exclusive food sources ( $N_1$  and  $N_2$ ) and the best solution found so far.

The fitness of the newly found food-source  $V_i$  is calculated and based on the fitness of  $X_i$  and  $V_i$ , greedy selection is applied between them i.e. if the fitness of the  $V_i$  is higher than that of  $X_i$ ,  $V_i$  is selected otherwise,  $X_i$  is retained. The fitness of the food-source is calculated using Eq. 2.6.

In the onlooker phase, the food-sources having higher probabilities of improvement are selected and updated. The probability of improvement associated with  $i^{th}$  food-source is calculated using Eq. 2.7. The new food-source  $V_i$  around selected food-source  $X_i$  can be calculated as,

$$V_{ij} = \begin{cases} X_{ij}, & \text{if } j \neq p \\ C, & \text{if } j = p \text{ and } fit_{N_1} > fit_i \\ D, & \text{if } j = p \text{ and } fit_{N_1} < fit_i \end{cases} \quad (2.9)$$

$$C = X_{N_1j} + \psi \cdot (2X_{N_1j} - X_{ij} - GBest_j)$$

$$D = X_{ij} + \psi \cdot (2X_{ij} - X_{N_1j} - GBest_j)$$

where,  $j = 1, 2, \dots, D$ ;  $\psi$  is a random number in range  $[-0.5, 0.5]$ ;  $p$  is a random integer selected from range  $[1, D]$ ;  $N_1$  is a random integer selected from range  $[1, D]$ ;  $GBest$  is a best solution found by algorithm so far. Based on the fitness of  $V_i$  and  $X_i$ , the greedy selection is carried out. The scout bee in EABC algorithm behaves similarly like that in ABC algorithm.

In the ABC algorithm, the employee bee phase is followed by the onlooker bee phase.

Thus, each algorithmic iteration of the ABC algorithm contains employee bee phase, onlooker bee phase, and scout bee phase. However, in the EABC algorithm, during each algorithmic iteration, there are two employee bee phase, one onlooker bee phase and, one scout bee phase.

## 2.5 Benchmark functions

In Table 2.1, the benchmark functions used for comparing performances of the evolutionary algorithms are described with their solutions and search spaces. The functions  $f_1$  and  $f_2$  are unimodal, while other functions are multimodal. The initialization range for each function is same as the search space of the algorithm. All these functions represent minimization problem.

**TABLE 2.1: Benchmark functions used for the experiment.**

	Function	Search-space	Minimum value
Sphere	$f_1(X) = \sum_{i=1}^N x_i^2$	$[-100, 100]^N$	0
Ackley	$f_2(X) = 20 + e - 20 \exp(-0.2 \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}) - \exp(\frac{1}{N} \sum_{i=1}^N \cos(2\pi x_i))$	$[-32, 32]^N$	0
Griewank	$f_3(X) = \frac{1}{4000} \sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^N$	0
Rastrigin	$f_4(X) = 10n + \sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i))$	$[-5.12, 5.12]^N$	0
Rosenbrock	$f_5(X) = \sum_{i=1}^{N-1} (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2)$	$[-10, 10]^N$	0
Alpine	$f_6(x) = \sum_{i=1}^N  x_i \sin(x_i) + 0.1 x_i $	$[-10, 10]^N$	0

## 2.6 Algorithm Parameters and Results

In this section, we discuss the experiment carried out on the benchmark functions to compare the performances of the ABC, PSO, EABC, MPSO, MABC (Modified Artificial Bee Colony) and, GABC (Gbest-guided Artificial Bee Colony) algorithms.

**TABLE 2.2: Results of solution of 30-dimensional benchmark functions with 30 independent runs.**

Function		ABC	GABC	MABC	EABC	PSO	MPSO
Sphere ( $f_1$ )	Average	1.3E-09	6.7E-20	8.3E-25	1.5E-33	6.1E+00	<b>0.0E+00</b>
	SD	7.5E-10	7.4E-20	6.3E-25	1.0E-33	3.0E+01	<b>0.0E+00</b>
	Best	2.4E-10	1.3E-20	1.8E-25	2.6E-34	<b>0.0E+00</b>	<b>0.0E+00</b>
	Worst	3.1E-09	3.7E-19	3.2E-24	5.6E-33	1.6E+02	<b>0.0E+00</b>
Ackley ( $f_2$ )	Average	8.3E-06	6.8E-11	2.8E-12	<b>3.1E-22</b>	9.7E+00	5.1E-16
	SD	2.9E-06	1.8E-11	9.8E-13	<b>3.7E-22</b>	8.7E+00	1.7E-16
	Best	2.6E-06	3.3E-11	1.1E-12	<b>1.4E-23</b>	1.6E-152	3.0E-22
	Worst	1.5E-06	1.0E-10	5.2E-12	<b>1.6E-21</b>	1.9E+01	5.8E-16
Griewank ( $f_3$ )	Average	4.0E-08	2.2E-10	1.3E-15	6.2E-24	5.9E-02	<b>0.0E+00</b>
	SD	8.4E-08	7.1E-10	3.8E-15	2.1E-23	1.8E-01	<b>0.0E+00</b>
	Best	8.1E-11	9.6E-22	2.5E-23	6.4E-30	0.0E+00	<b>0.0E+00</b>
	Worst	3.0E-07	3.6E-09	1.6E-14	1.1E-22	8.2E-01	<b>0.0E+00</b>
Rastrigin ( $f_4$ )	Average	1.0E-05	5.9E-17	1.2E-24	2.5E-31	5.9E-02	<b>0.0E+00</b>
	SD	4.5E-05	3.1E-16	1.5E-24	1.7E-31	1.8E-01	<b>0.0E+00</b>
	Best	1.0E-10	5.8E-20	1.1E-25	3.5E-32	<b>0.0E+00</b>	<b>0.0E+00</b>
	Worst	2.5E-04	1.7E-15	6.6E-24	7.1E-31	8.2E-01	<b>0.0E+00</b>
Rosenbrock ( $f_5$ )	Average	3.4E-01	1.7E+00	7.3E+00	<b>3.2E-02</b>	3.4E+01	1.5E+01
	SD	2.1E-01	4.5E+00	8.0E+00	<b>3.5E-02</b>	5.9E+01	1.4E+01
	Best	2.1E-02	1.2E-03	5.8E-01	<b>1.8E-03</b>	5.4E-04	1.5E-05
	Worst	7.9E-01	1.9E+01	3.5E+01	<b>1.4E-01</b>	3.4E+02	2.9E+01
Alpine ( $f_6$ )	Average	1.5E-04	9.9E-06	2.8E-12	5.8E-17	1.3E+00	<b>0.0E+00</b>
	SD	8.3E-05	1.2E-05	3.8E-12	5.9E-17	2.8E+00	<b>0.0E+00</b>
	Best	3.3E-05	6.8E-08	4.4E-13	9.4E-18	<b>0.0E+00</b>	<b>0.0E+00</b>
	Worst	3.6E-04	5.5E-05	2.0E-11	2.9E-16	1.3E+01	<b>0.0E+00</b>

The GABC and MABC algorithms are modified versions of the ABC algorithms and detailed description on these algorithms can be found in [33] and [34], respectively.

For the PSO algorithm, swarm size is taken 75. The values of the  $C_1$  (cognitive coefficient) and  $C_2$  (social coefficient) are set to 1.49. The parameters for the inertia constants  $w_{up}$  and  $w_{low}$  are set to 0.9 and 0.4, respectively. For the MPSO algorithms, the swarm size is set to 75. The values of  $C_1$  and  $C_2$  are set to 1.99 and 0.49, respectively. The parameters for the inertia constants  $w_{up}$  and  $w_{low}$  are set to 0.9 and 0.4, respectively. The value of the parameter  $S$ , used for the calculation of reduced search space during partial reinitialization, is set to 1% of the actual search space of the problem. The value of parameter  $DS_{limit}$  is set to 85% of the problem dimension. Similarly, parameter  $V_{min}$  which is used to determine the velocity saturation is set to 0.1% of the  $V_{max}$ .

For the ABC and EABC algorithms, the size of the swarm (i.e. the total number of the food-sources) is set to 75. The value of  $T_{limit}$  is set to 200. For the MABC algorithm, swarm size is set to 75 and value of the selection probability is taken 0.7 as suggested in [33]. For the GABC algorithm, the swarm size is 75 and the value of the parameter  $C$  is 1.5 as suggested in [32].

For the fair comparison of the algorithms, the algorithms are terminated based on the number of the function evaluations. The maximum number of the function evaluations are set to 150000 for the problem with 30 dimensions and it is set to 300000 for the problem with 60 dimensions. Further, we utilize the ‘‘Mersenne Twister’’ method [39] for the generation of the random number. In order to achieve very high accuracy in the calculations, we use MPFR library [40] for multiple-precision floating-point computations with correct rounding.

In order to compare the performances of the various algorithms, each benchmark function described in Table 2.1 is solved 30 times independently with problem dimensions 30 and 60. The obtained results are illustrated in Table 2.2 and Table 2.3. We consider average and standard deviation (SD) of obtained solutions over 30 independent runs along with their best and worst case values.

The obtained results reveal that the MPSO algorithm provides better results in solving four benchmark functions (Sphere, Griewank, Rastrinsin and, Alpine) compared to other considered algorithms. In the case of the remaining two algorithms (Ackley and Rosenbroke), the performance of the EABC algorithm is found better than the other algorithms. Further, the performance of the MPSO algorithm is always found

**TABLE 2.3: Results of solution of 60-dimensional benchmark functions with 30 independent runs.**

Function		ABC	GABC	MABC	EABC	PSO	MPSO
Sphere ( $f_1$ )	Average	1.4E-09	4.4E-19	8.6E-23	5.8E-32	3.3E+02	<b>0.0E+00</b>
	SD	1.3E-09	3.1E-19	4.3E-23	2.9E-32	1.7E+03	<b>0.0E+00</b>
	Best	4.1E-10	6.2E-20	3.3E-23	1.1E-32	<b>0.0E+00</b>	<b>0.0E+00</b>
	Worst	5.3E-09	1.6E-18	1.8E-22	1.2E-31	1.0E+04	<b>0.0E+00</b>
Ackley ( $f_2$ )	Average	1.4E-05	3.5E-10	3.1E-11	<b>1.4E-21</b>	1.7E+01	4.6E-16
	SD	3.5E-06	8.3E-11	1.6E-11	<b>3.3E-21</b>	5.6E+00	2.3E-16
	Best	7.8E-06	2.3E-10	1.5E-11	<b>2.4E-23</b>	8.3E-01	2.1E-20
	Worst	2.2E-05	5.3E-10	8.9E-11	<b>1.8E-20</b>	1.9E+01	5.8E-16
Griewank ( $f_3$ )	Average	1.4E-09	1.2E-07	2.9E-20	2.9E-25	6.2E+00	<b>0.0E+00</b>
	SD	2.4E-09	6.9E-07	1.2E-19	1.5E-24	2.2E+01	<b>0.0E+00</b>
	Best	1.0E-10	3.1E-22	5.9E-24	2.1E-30	<b>0.0E+00</b>	<b>0.0E+00</b>
	Worst	1.2E-08	3.8E-06	6.9E-19	8.8E-24	9.0E+01	<b>0.0E+00</b>
Rastrigin ( $f_4$ )	Average	2.9E-01	3.0E-14	1.3E-22	1.1E-29	1.8E+02	<b>0.0E+00</b>
	SD	5.8E-01	7.9E-14	7.0E-23	8.0E-30	5.5E+01	<b>0.0E+00</b>
	Best	8.0E-11	2.8E-18	3.3E-22	1.3E-30	2.4E+01	<b>0.0E+00</b>
	Worst	1.9E+00	4.1E-13	4.2E-22	3.4E-29	2.3E+02	<b>0.0E+00</b>
Rosenbrock ( $f_5$ )	Average	4.1E-01	8.1E+00	1.5E+01	<b>4.0E-02</b>	3.9E+02	5.2E+01
	SD	4.1E-01	1.7E+01	1.4E+01	<b>4.5E-02</b>	1.7E+03	1.7E+01
	Best	3.3E-02	4.8E-03	1.2E+00	<b>4.6E-04</b>	2.8E-03	3.2E-05
	Worst	1.9E+00	8.4E+01	5.8E+01	<b>1.6E-01</b>	1.0E+04	5.9E+01
Alpine ( $f_6$ )	Average	2.5E-03	8.4E-05	8.4E-11	6.2E-16	5.2E+00	<b>0.0E+00</b>
	SD	2.9E-03	7.3E-05	5.1E-11	5.0E-16	1.8E-01	<b>0.0E+00</b>
	Best	1.8E-04	8.9E-06	1.7E-11	1.1E-16	1.6E+01	<b>0.0E+00</b>
	Worst	1.2E-02	2.5E-04	2.3E-10	2.3E-15	4.3E+00	<b>0.0E+00</b>

better than the PSO algorithm. This indicates that the partial reinitialization scheme used in MPSO algorithm has improved performance of the algorithm significantly. Similarly, the EABC algorithm has provided better solutions than ABC, GABC and, MABC algorithms. Thus, modified strategies of the EABC algorithm have improved the overall performance of the algorithm significantly.

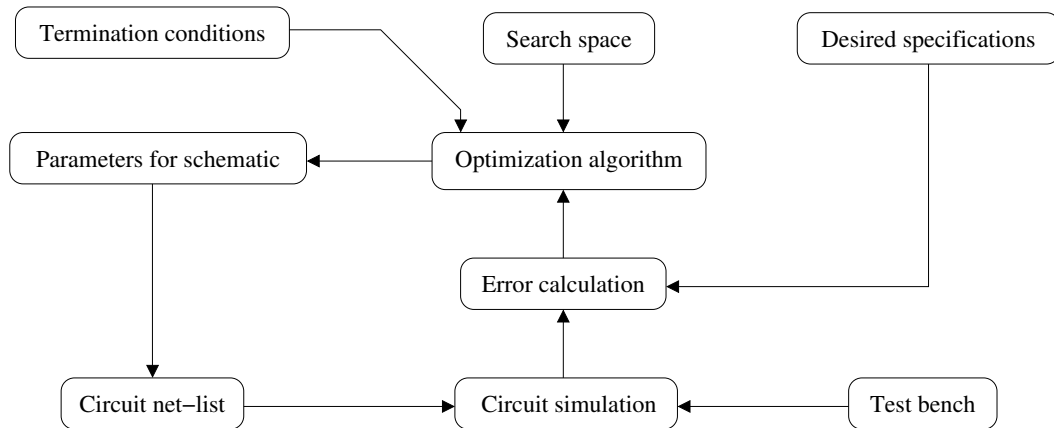
## CHAPTER 3

# Automatic circuit design at schematic level

In this chapter, the automatic circuit design framework for the schematic-level circuit design is presented. The design of three analog circuits; (1) Two-stage operational amplifier, (2) High gain low voltage OTA and, (3) Second generation current conveyor are presented. These circuits are designed with the help of the ABC, PSO, MPSO and EABC algorithms. The obtained results are used to compare the performances of these algorithms.

### 3.1 Environment for schematic-level design automation

The design of the analog CMOS circuit involves determining width and length of various MOSFET transistors and value of various passive components such as resistors, capacitors and, inductors. Due to the nonlinearities of the MOSFET device, the traditional approach fails to provide solutions in time efficient manner. To overcome this difficulty, many researchers have used evolutionary algorithms to design analog circuits at schematic-level. In Fig. 3.1, the conceptual block diagram of the optimizer for the automatic analog circuit design at schematic-level is illustrated. The optimization algorithm may be any evolutionary algorithm. The search-space generally contains the



**FIGURE 3.1: Conceptual block diagram of optimizer for automatic CMOS circuit design at schematic-level.**

information about the upper and lower bounds for the various design parameters. The proper search-space helps algorithm to avoid generating non-practical solutions. From the search space, optimization algorithm generates set of solutions containing design parameters for the circuit. From the search space, optimization algorithm generates set of solutions containing design parameters for the circuit. For the each provided solution, the circuit is simulated against pre-determined test-bench. The error is calculated from the obtained simulation results and the calculated error is used by the algorithm to generate a new set of solutions. In order to calculate the error, the formula of the root-mean-square (RMS) is utilized in this work. Let's assume that there are  $N$  specifications considered for the circuit design. Then, the RMS error is calculated as follows,

$$f_e (\text{RMS Error } (\%) ) = \sqrt{\frac{1}{N} \sum_{i=1}^{i=N} E_i} \times 100 \quad (3.1)$$

$$E_i = \begin{cases} 0, & \text{if } Spect_i \text{ is satisfied} \\ \left( \frac{Spect_i^{sim} - Spect_i}{Spect_i} \right)^2, & \text{Otherwise} \end{cases}$$

where,  $i = 1, 2, \dots, N$ ;  $Spect_i$  is  $i^{th}$  desired specification;  $Spect_i^{sim}$  is  $i^{th}$  specification obtained after circuit simulation. The RMS error gives equal importance to each specification.

The automatic circuit design using evolutionary algorithm is an iterative process. The algorithm continues to generate new solutions until the termination criteria are satisfied or all the specifications are obtained. The termination criteria are generally described

## 3.2 Circuit design Examples

in term of the design time or allowed number of circuit evaluations.

### 3.2 Circuit design Examples

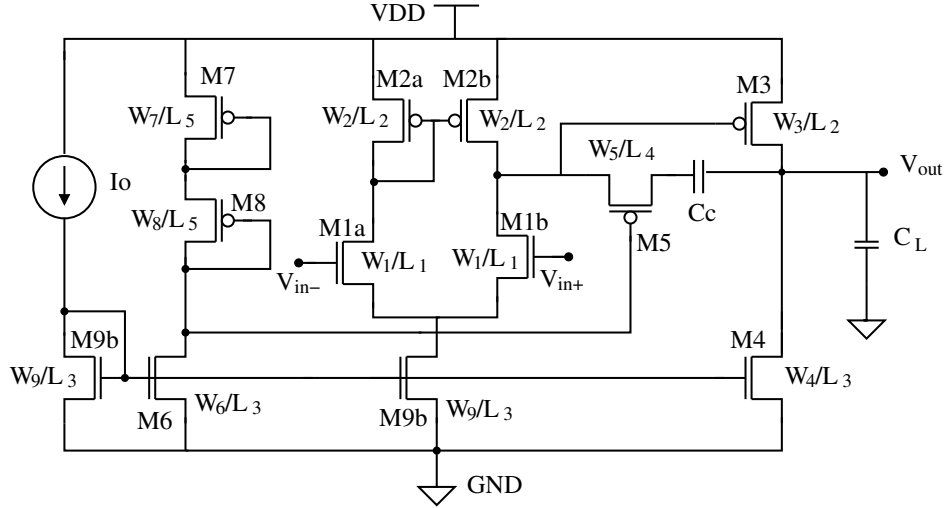
In order to compare the performances of the ABC, EABC, PSO and MPSO algorithms, we designed following analog CMOS circuits using these algorithms.

- Two-stage operational amplifier
- High-gain low-voltage bulk-driven OTA
- Second generation current conveyor

Further, these circuits were also designed using two variants of ABC algorithm named G-Best Artificial Bee Colony (GABC) [33] algorithm and Modified Artificial Bee Colony (MABC) [32] algorithm. In order to compare consistency of the algorithms and their quality of solutions, each circuit was designed 10 times independently in  $0.13\ \mu m$  and  $0.09\ \mu m$  CMOS technologies. Further, the average of the obtained specifications is considered. For the simulation of the circuit, NG-SPICE [41] simulator was utilized. In this work, PTM [42] device models are utilized. The design is carried out on the computer having following major specifications.

- Operating System: UBUNTU 12.04
- Processor: AMD-FX8350
- Processor clock: 4GHz
- RAM: 4GB

The parameters for the various algorithms are set in the following manner. For the ABC and EABC algorithms, we took 20 employee bees and the same number of onlooker bees. The value of the maximum number of trial before reinitialization of food-source ( $T_{limit}$ ) is set to 200. For the GABC algorithm, 20 employee bees and 20 onlooker bees are taken. The value of the constant ‘ $C$ ’ is set to 1.5 as suggested in [33]. Further, the re-initialization limit ( $T_{limit}$ ) is set to 200. For the MABC algorithm, 20 employee bees are taken with chaotic initialization scheme. The value for the selection probability is set to 0.7 as suggested in [32]. For the PSO algorithm, the parameters are set as



**FIGURE 3.2: Two-stage operational amplifier**

follows: The value for the cognitive coefficient ( $C_1$ ) and social coefficient ( $C_2$ ) is set to 1.5; the upper and lower values of the inertia constant  $w$  are set to 0.9 and 0.4, respectively. For the MPSO algorithm, the value of the cognitive coefficient is set to 0.99 and the value of the social coefficient is set to 1.99. The value of parameter ‘ $S$ ’ to control the reinitialization region in search-space is set to 0.25 [43]. The maximum circuit evaluations are set to 5000 for all considered algorithms.

### 3.2.1 Two-stage operational amplifier

The two-stage operational amplifier (op-amp) is a very versatile circuit. It is widely used in the analog CMOS integrated circuits. It is a building block of many important circuits such as filter, mixer, modulator, ADC, DAC, amplifier, analog front-end etc. The circuit of the designed op-amp is shown in Fig. 3.2 [9].

The input is applied to a differential pair of differential amplifier formed by the transistors M1a, M1b, M2a, M2b, and M9. The second stage amplifier is formed by the transistors M3 and M4. The Miller capacitor  $C_c$  and transistors M5 provide the frequency compensation. The gain of the op-amp can be improved by reducing biasing current of the amplifier. However, this deteriorates slew rate and bandwidth. The bandwidth and phase-margin can be affected by changing the value of  $C_c$  and size of transistor M5. The improvement in the power consumption adversely affects the slew rate. Thus the design of the op-amp requires trade-off approach.

The specifications considered for the design of the op-amp in both  $0.13\mu m$  and  $0.09\mu m$

### 3.2 Circuit design Examples

**TABLE 3.1: Two stage operational amplifier: search space for design variables.**

Parameter	Search space (0.13 $\mu m$ )	Search space (0.09 $\mu m$ )
$W_1$ to $W_9$	0.5 $\mu m$ to 10 $\mu m$	0.5 $\mu m$ to 10 $\mu m$
$L_1$ to $L_5$	0.2 $\mu m$ to 1 $\mu m$	0.1 $\mu m$ to 1 $\mu m$
$I_0$	1 $\mu A$ to 10 $\mu A$	1 $\mu A$ to 10 $\mu A$
$C_C$	0.001pF to 1pF	0.001pF to 1pF

CMOS technologies are as follows,

- Gain  $\geq 80dB$
- Unity gain bandwidth (UGB)  $\geq 100MHz$
- Phase margin  $\geq 60^\circ$
- Power consumption  $\leq 20\mu W$
- Rise and fall slew rate  $\geq 40V/\mu S$
- Power supply rejection ratio (PSRR)  $\geq 75dB$
- Common mode rejection ratio (CMRR)  $\geq 80dB$

The circuit is designed to drive the load of 0.05 pF. The value of the power supply voltage ( $V_{DD}$ ) is 1.2V and 1.0V for 0.13  $\mu m$  and 0.09  $\mu m$  CMOS technologies, respectively.

The design parameters for the op-amp are width and length of various transistors, the value of the biasing current  $I_0$  and value of Miller capacitor  $C_C$ . The design parameters with their upper and lower bounds are mentioned in Table 3.1. There are total 16 design parameters. The aim of the optimization algorithm is to find the appropriate values of these design parameters so that all the specifications are satisfied. The op-amp is designed using ABC, GABC, MABC, PSO, EABC and MPSO algorithms. The op-amp is designed 10 times independently using each algorithm and average of the obtained results are considered for the comparison.

The cost function for the op-amp design problem is formulated using RMS error formula described in Eq. 3.1. This cost function gives equal importance to all the specifications. There are total 8 specifications. During the design process, cost function value is

**TABLE 3.2: Two stage operational amplifier: Illustration for calculation of cost function.**

Specification	Desired value	Simulation result	$E_i$
Gain (dB)	$\geq 80$	82.7	0.0
Bandwidth(MHz)	$\geq 100$	98.8	0.000144
Phase Margin( $^\circ$ )	$\geq 60$	60.5	0.0
Power consumption( $\mu W$ )	$\leq 20$	20.6	0.0009
Rise slew rate ( $V/\mu S$ )	$\geq 40$	41.4	0.0
Fall slew rate ( $V/\mu S$ )	$\geq 40$	38.7	0.00105625
CMRR (dB)	$\geq 75$	78.8	0.0
PSRR (dB)	$\geq 80$	79.9	0.000001562
Cost function value (RMS error )			1.63%

calculated after every circuit evaluation. The example calculation of the cost function is illustrated in Table 3.2. When all the desired specifications are satisfied, the value of cost function becomes zero.

The average of the obtained specifications over 10 independent design trials for various algorithms are shown in Table 3.3. The Table 3.4 represents various performance parameters for the comparison of the algorithms. In Fig. 3.3, the variations in the RMS error with the circuit evaluations are shown.

The following observations can be made from the obtained results.

- The average RMS error, which indicates the overall performance of the algorithm, is the minimum with the EABC algorithm for the design of op-amp in  $0.13 \mu m$  CMOS technology. However, the average RMS error for op-amp design in  $0.09 \mu m$  technology is the minimum for the MPSO algorithm.
- The worst RMS error in ten independent design trials is the minimum for the EABC algorithm for the design in  $0.13 \mu m$  CMOS technology. For the design in  $0.09 \mu m$  technology, MPSO algorithm has minimum value for worst RMS error.
- For the design in  $0.13 \mu m$  CMOS technology, GABC algorithm has minimum design time and minimum value for the average circuit evaluations. However, its overall performance is inferior compared to the EABC and MPSO algorithm.
- The EABC algorithm has designed op-amp with zero RMS error i.e. all the design specifications are achieved five times and nine times for design in  $0.13 \mu m$  and

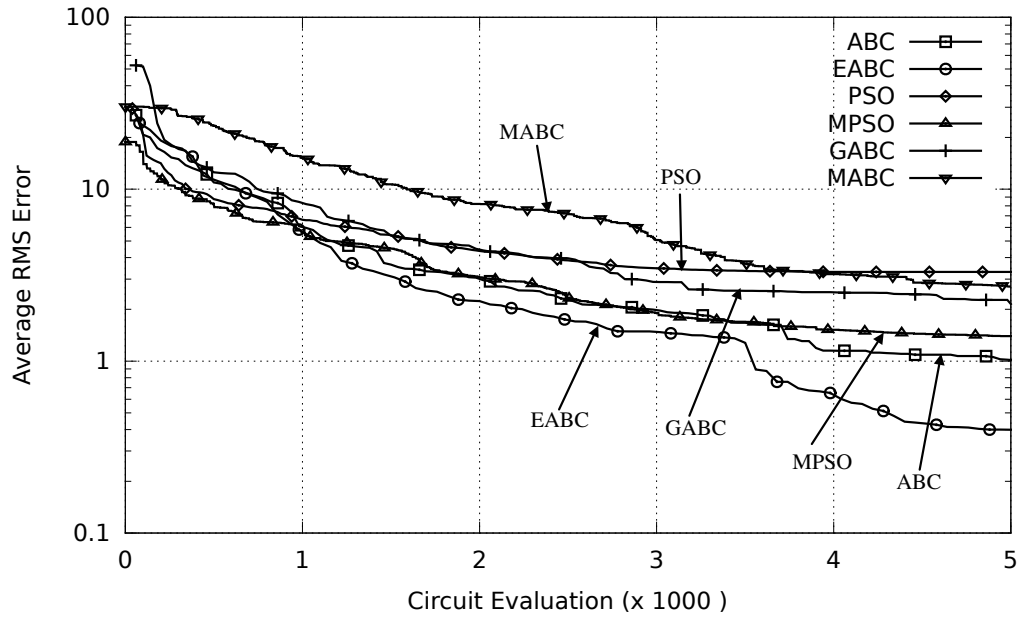
### 3.2 Circuit design Examples

**TABLE 3.3: Two stage op-amp: Average of obtained simulation results over 10 independent design trials.**

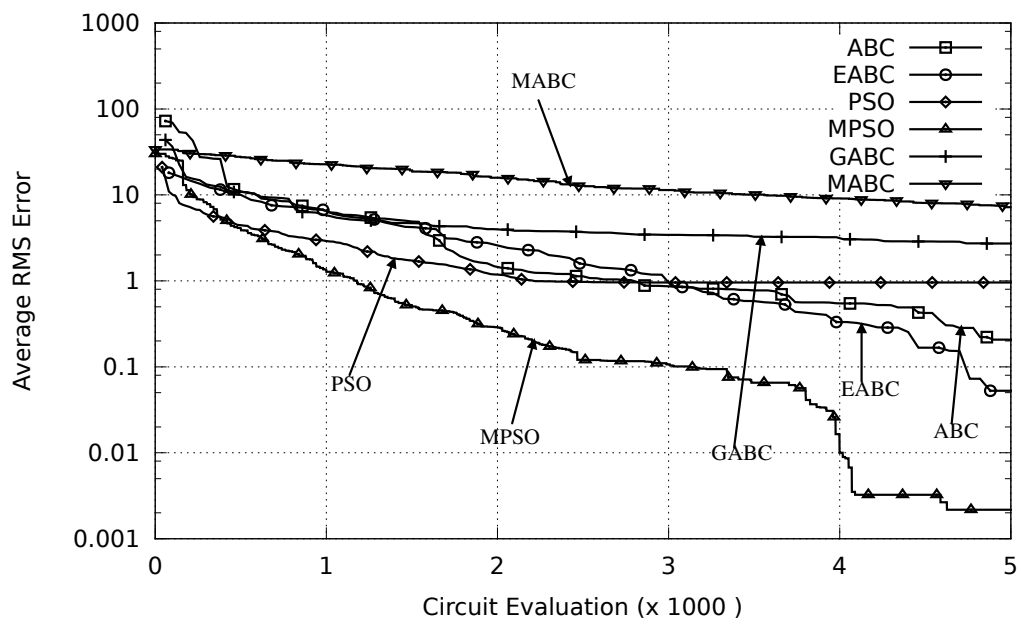
Technology	Specification	Desired value	ABC	GABC	MABC	PSO	EABC	MPSO
0.13 $\mu m$	Gain (dB)	$\geq 80$	80.1	77.8	75.9	76.4	79.5	79.6
	PM ( $^{\circ}$ )	$\geq 60$	63.9	64.0	62.7	60.7	63.8	60.1
	UGB (MHz)	$\geq 100$	242	184	117	129	153	151.2
	PSRR (dB)	$\geq 80$	80.7	83.5	78.4	77.9	85.5	76.2
	CMRR (dB)	$\geq 75$	87.7	81.9	79.6	83.9	82.8	84.3
	PC ( $\mu W$ )	$\leq 20$	19.6	19.3	20.5	21.0	19.7	21.1
	RSR ( $V/\mu S$ )	$\geq 40$	56.2	69.8	43.4	43.5	66.5	45.8
	FSR ( $V/\mu S$ )	$\geq 40$	39.6	40.1	38.6	37.6	40.1	40.8
0.09 $\mu m$	Gain (dB)	$\geq 80$	80.2	79.0	72.7	79.7	80.6	80.7
	PM ( $^{\circ}$ )	$\geq 60$	66.1	62.6	61.5	61.5	63.5	60.5
	UGB (MHz)	$\geq 100$	119	144	155	131	137.0	129.1
	PSRR (dB)	$\geq 80$	75.5	74.1	73.2	77.1	76.6	76.4
	CMRR (dB)	$\geq 75$	80.5	78.8	71.7	79.2	80.6	80.2
	PC ( $\mu W$ )	$\leq 20$	19.7	19.4	16.9	20.1	19.4	19.7
	RSR ( $V/\mu S$ )	$\geq 40$	45.7	48.2	43.7	46.8	50.4	46.8
	FSR ( $V/\mu S$ )	$\geq 40$	39.9	39.3	37.3	39.3	42.4	39.3

**TABLE 3.4: Two stage operational amplifier (Schematic-level): Performance comparison of algorithms based on 10 independent design trials.**

	Technology	ABC	GABC	MABC	PSO	EABC	MPSO
Average RMS error(%)	0.13 $\mu m$	1.01	2.02	2.71	3.30	<b>0.39</b>	1.39
	0.09 $\mu m$	0.20	2.71	7.18	0.95	0.05	<b>0.002</b>
Worst RMS error(%)	0.13 $\mu m$	5.77	6.74	8.84	6.80	<b>1.97</b>	2.11
	0.09 $\mu m$	0.58	5.62	21.2	4.68	0.52	<b>0.02</b>
Average circuit evaluations	0.13 $\mu m$	4482	<b>3604</b>	5000	5000	3608	4680
	0.09 $\mu m$	4720	5000	5000	4406	3500	<b>2946</b>
Average design time (Minutes)	0.13 $\mu m$	39.6	<b>31.2</b>	40.7	43.2	36.2	41.1
	0.09 $\mu m$	16.41	17.3	15.9	15.8	12.2	<b>10.8</b>
Number of times zero RMS error is obtained	0.13 $\mu m$	3	<b>5</b>	0	0	<b>5</b>	1
	0.09 $\mu m$	3	0	0	2	<b>9</b>	<b>9</b>



(a) 0.13  $\mu\text{m}$  CMOS technology



(b) 0.09  $\mu\text{m}$  CMOS technology

**FIGURE 3.3:** Two stage operational amplifier (Schematic-level): Variations in RMS error with circuit evaluations

**TABLE 3.5: Two stage op-amp: Results of experiment to achieve best possible design in  $0.13 \mu m$  technology.**

Specification	ABC	EABC
Gain (dB)	84.3	83.7
PM ( $^{\circ}$ )	60.5	70.0
UGB ( $MHz$ )	152.4	206.6
PSRR (dB)	83.0	82.6
CMRR (dB)	79.6	79.9
PC ( $\mu W$ )	20.1	17.9
RSR ( $V/\mu S$ )	70.9	62.7
FSR ( $V/\mu S$ )	46.0	44.3

$0.09 \mu m$  technologies, respectively. The MPSO algorithm has designed op-amp with zero RMS error one time and nine times in  $0.13 \mu m$  and  $0.09 \mu m$  CMOS technologies, respectively.

- For the design of op-amp in the  $0.13 \mu m$  technology, EABC algorithm provides better results, while for the design in  $0.09 \mu m$  technology the MPSO algorithm provides better results.
- The PSO algorithm in its initial stage converges fast, however, due to loss of diversity of population, it performs poorly.

In order to exploit the maximum capacity of the automatic circuit design tool and to achieve the best possible design of op-amp in  $0.13 \mu m$  technology, a new experiment is conducted using ABC and EABC algorithm in the following manner. First, the design of the op-amp is carried with the normal design specifications as described in Table 3.3. Once these specifications are achieved, the specifications are made tougher i.e. gain requirement is increased by  $2 dB$ , the bandwidth requirement is increased by  $2 MHz$  and, the slew rate is increased by  $2 V/\mu s$ . The other specifications are maintained at the same value. When these specifications are achieved, the specifications are again made tougher in a similar manner. This continues till the maximum circuit evaluations are reached. The design specifications of the generated op-amp are illustrated in Table 3.5.

Further, the process variations can also be considered in the process of the schematic-level automatic design. The op-amp is also designed considering the process variations using ABC, EABC and MPSO algorithms in  $0.13 \mu m$  and  $0.09 \mu m$  CMOS technologies.

**TABLE 3.6: Two stage operational amplifier: The performance measure of the obtained solution at TT, FF, FS, SF and SS process corner obtained by ABC, EABC and MPSO algorithms in 0.13  $\mu m$  technology.**

Design in 0.13 $\mu m$ CMOS Technology									
Algorithm	Process corner	Gain (dB)	UGB (MHz)	PM ( $^\circ$ )	PC ( $\mu W$ )	RSR ( $V/\mu S$ )	FSR ( $V/\mu S$ )	PSRR (dB)	CMRR (dB)
EABC	TT	82.1	132.0	63.4	19.1	60.4	42.7	90.4	83.0
	FF	82.4	128.4	60.3	19.9	59.4	43.2	90.0	85.0
	FS	82.6	172.0	65.1	18.5	66.6	42.7	91.1	84.7
	SF	81.8	123.3	60.9	19.8	59.0	42.8	89.5	80.2
	SS	81.8	163.3	66.4	18.4	66.0	42.2	90.8	80.1
RMS Error: 0.0%					Design Time: 45.6 Minutes				
ABC	TT	80.7	197.5	66.1	16.5	64.6	40.8	103.3	82.1
	FF	80.2	193.0	68.1	16.7	65.3	41.2	95.0	83.7
	FS	81.5	202.8	62.3	16.7	64.9	40.8	91.1	80.7
	SF	80.0	186.1	69.4	16.6	64.2	40.8	91.2	82.7
	SS	81.3	196.1	63.3	16.6	64.0	40.4	104.0	81.0
RMS Error: 0.0%					Design Time: 93.3 Minutes				
MPSO	TT	80.7	185.7	63.2	17.8	62.4	44.3	89.0	84.4
	FF	80.2	184.7	65.1	17.9	62.0	44.6	87.3	85.5
	FS	81.6	189.1	60.0	17.8	63.8	44.3	96.0	84.0
	SF	80.0	178.1	66.0	17.8	61.5	44.2	86.2	83.7
	SS	81.4	183.1	61.1	17.8	63.2	43.8	91.8	83.0
RMS Error: 0.0%					Design Time: 142 Minutes				

**TABLE 3.7: Two stage operational amplifier: The performance measure of the obtained solution at TT, FF, FS, SF and SS process corner obtained by ABC, EABC and MPSO algorithms in 0.09  $\mu m$  technology.**

Design in 0.09 $\mu m$ CMOS Technology									
Algorithm	Process corner	Gain (dB)	UGB (MHz)	PM ( $^\circ$ )	PC ( $\mu W$ )	RSR ( $V/\mu S$ )	FSR ( $V/\mu S$ )	PSRR (dB)	CMRR (dB)
EABC	TT	77.9	154.9	66.8	20.0	49.0	40.0	77.0	86.7
	FF	78.1	173.8	71.0	20.2	49.3	40.8	77.5	91.8
	FS	77.5	174.5	66.0	19.0	49.6	40.2	73.6	90.7
	SF	78.5	166.1	71.9	20.1	48.8	40.5	78.9	79.5
	SS	78.0	168.9	67.1	18.9	49.4	40.0	76.3	81.5
RMS Error: 0.93%					Design Time: 140Min				
ABC	TT	74.4	116.2	72.3	20.8	44.6	40.4	76.2	76.9
	FF	74.8	131.2	63.6	21.9	45.8	41.1	76.8	81.5
	FS	74.2	282.2	70.3	19.8	44.0	38.7	73.4	91.0
	SF	75.2	123.7	64.5	21.6	45.2	41.4	77.9	69.7
	SS	74.7	264.4	77.6	19.5	43.4	38.9	75.5	70.7
RMS Error: 4.13%					Design Time: 171.5Min				
MPSO	TT	80.0	162.6	64.8	15.1	50.2	40.8	78.2	90.6
	FF	80.4	159.7	66.1	15.2	50.1	41.3	78.7	95.8
	FS	80.1	176.0	62.2	15.2	50.6	40.8	75.3	90.4
	SF	80.3	152.0	67.1	15.1	50.0	40.8	79.9	80.5
	SS	80.0	169.0	63.6	15.1	50.7	40.2	77.6	83.3
RMS Error: 0.0%					Design Time: 59.9Min				

### 3.2 Circuit design Examples

The five process corners i.e. TT, SS, FF, FS, and SF are considered. This leads to the simulation of the circuit at these process corners in each evaluation. This increases the design time. In order to reduce design time, instead of simulating circuit every time at five process corners, we use following strategy,

**TABLE 3.8: Two stage op-amp design considering process variations: values of obtained design parameters.**

Technology	0.13 $\mu m$			0.09 $\mu m$		
Design parameter	ABC	EABC	MPSO	ABC	EABC	MPSO
$W_1$ ( $\mu m$ )	5.4	1.4	4.9	2.2	4.0	3.7
$W_2$ ( $\mu m$ )	0.5	1.1	1.0	4.2	1.0	2.1
$W_3$ ( $\mu m$ )	4.9	4.5	7.5	6.2	4.2	7.8
$W_4$ ( $\mu m$ )	9.5	6.2	5.5	3.0	5.7	6.3
$W_5$ ( $\mu m$ )	2.7	1.5	5.8	0.6	4.7	7.7
$W_6$ ( $\mu m$ )	2.0	3.0	0.5	1.7	5.8	0.5
$W_7$ ( $\mu m$ )	4.5	0.5	2.7	3.1	1.1	7.0
$W_8$ ( $\mu m$ )	5.0	0.5	2.8	8.0	1.1	2.9
$W_9$ ( $\mu m$ )	3.9	3.3	2.4	2.0	2.2	2.1
$L_1$ ( $\mu m$ )	0.6	0.7	0.6	0.3	0.2	0.3
$L_2$ ( $\mu m$ )	0.3	0.4	0.3	0.3	0.3	0.3
$L_3$ ( $\mu m$ )	1.0	0.5	1.0	0.2	0.1	0.5
$L_4$ ( $\mu m$ )	0.3	0.8	0.5	0.1	0.2	0.3
$L_5$ ( $\mu m$ )	0.7	0.6	0.8	0.6	0.1	1.0
$I_0$ ( $\mu m$ )	2.8	3.7	3.3	5.1	2.8	2.9
$C_C$ ( $pF$ )	0.043	0.06	0.048	0.1	0.05	0.05

1. The circuit design starts considering only TT corner i.e. normal circuit optimization.
2. When the RMS error falls bellow 10%, FF and TT corners are considered. Thus the circuit is evaluated at two corners and average RMS error of these two corners is considered as a cost function.
3. When the average error of TT and FF corners falls bellow 9%, third corner SS is added in the design process.
4. Similarly, FS corner is added when average RMS error falls below 8% and SF corner is added when average error falls below 7%.



## 3.2 Circuit design Examples

terminal of the MOSFET instead of the gate terminal. Due to this, the supply voltage is not limited by the threshold voltage. Further, the use of the bulk-driven technique does not require any change in the MOSFET structure. This allows the use of the bulk-driven method in standard CMOS technology without any modifications [45]. However, the transconductance offered by the bulk-driven device is significantly low compared to the gate-driven device. This results in the lower gain of the amplifier.

In this work, the design of high-gain low-voltage bulk-driven OTA is presented. In Fig. 3.4, the circuit of the designed OTA is illustrated. This OTA uses the cross-coupled connection. The cross-coupled connection of transistors M1A and M2A imparts negative impedance degeneration which enhances the overall transconductance of the differential pair [46]. Further, two regulated current mirrors formed by M12, M14,  $I_C$  and M11, M15,  $I_C$  improve the gain of the OTA [47]. The high-gain low voltage bulk-driven OTA is designed using ABC, GABC, MABC, PSO and MPSO algorithms in  $0.13 \mu m$  and  $0.09 \mu m$  CMOS technologies to satisfy following specifications.

- Gain  $\geq 80dB$
- Unity gain bandwidth (UGB)  $\geq 100MHz$
- Phase margin (PM)  $\geq 60^\circ$
- Power consumption (PC)  $\leq 5\mu W$
- Rise and fall slew rate  $\geq 0.1V\mu S$

The supply voltage ( $VDD$ ) is set to  $0.5V$  for the design in both technologies. In order to give equal importance to each specification, the cost function is formulated using RMS error formula described using Eq. 3.1. The circuit is designed 10 times using considered algorithms and average of the obtained results are considered for the comparison. The design parameters are width and length of various transistors, the value of the current sources  $I_b$  and  $I_c$ . The search-space for these design parameters are illustrated in Table 3.9.

The average of the obtained specifications over 10 independent design trials for various algorithms are shown in Table 3.10. The Table 3.11 represents various performance parameters for the comparison of the algorithms. In Fig. 3.5, the variations in the RMS error with the circuit evaluations are shown.

The following observations can be made from the obtained results.

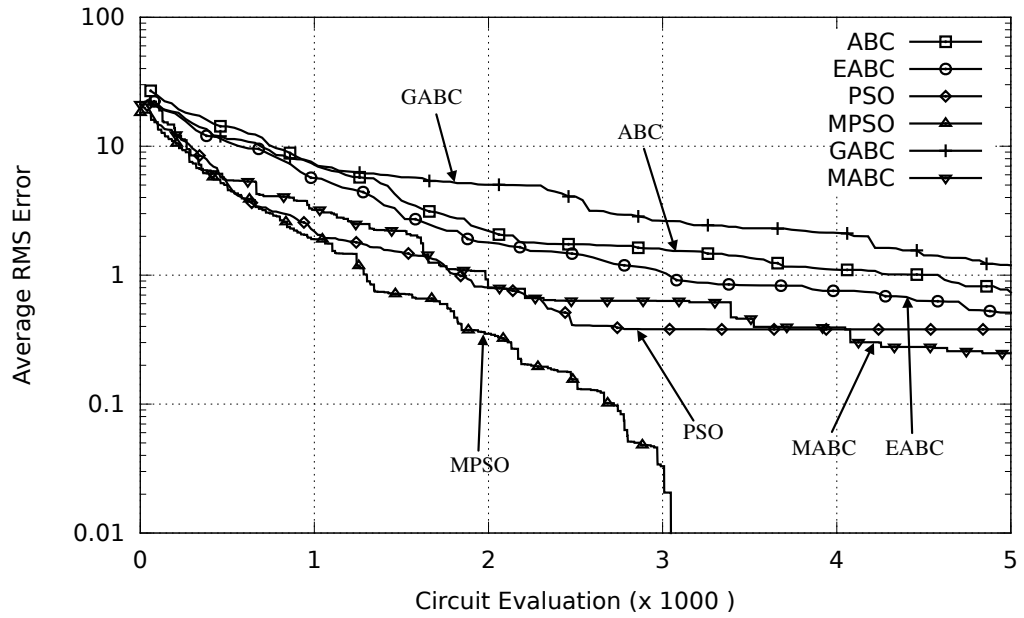
**TABLE 3.9: High gain low voltage bulk-driven OTA: Search space for design variables.**

Parameter	Search space (0.13 $\mu m$ )	Search space (0.09 $\mu m$ )
Width( $W$ ) of all transistors	1 $\mu m$ to 100 $\mu m$	1 $\mu m$ to 100 $\mu m$
Length( $L$ ) of all transistors	0.2 $\mu m$ to 2 $\mu m$	0.1 $\mu m$ to 2 $\mu m$
$I_b$	0.1 $\mu A$ to 2 $\mu A$	0.1 $\mu A$ to 2 $\mu A$
$I_c$	0.1 $\mu A$ to 1 $\mu A$	0.1 $\mu A$ to 1 $\mu A$

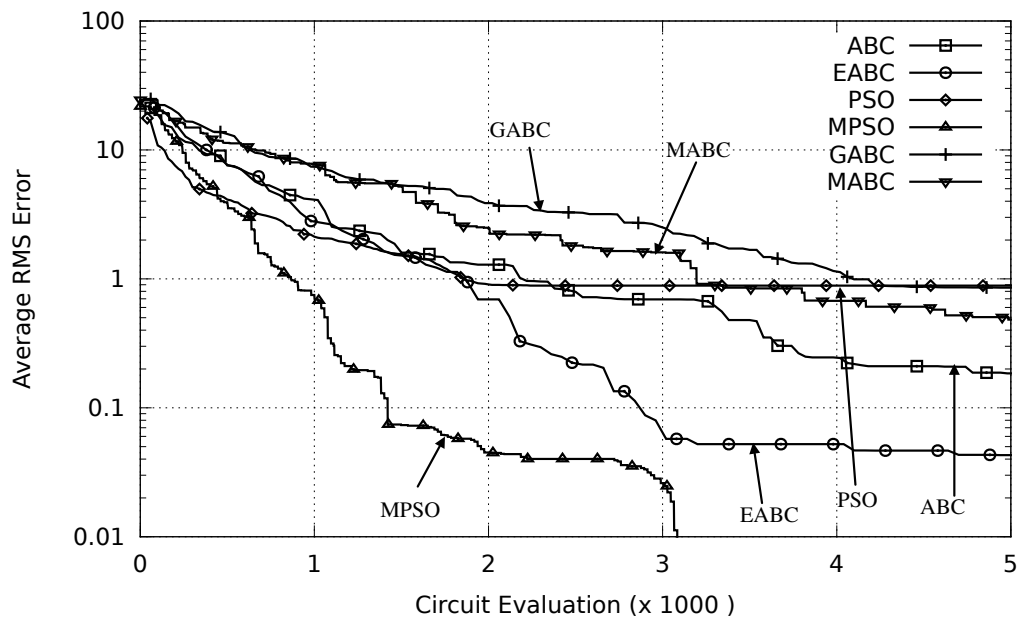
**TABLE 3.10: High-gain low-voltage bulk-driven OTA: Average of obtained simulation results over 10 independent design trials.**

Technology	Specification	Desired value	ABC	GABC	MABC	PSO	EABC	MPSO
0.13 $\mu m$	Gain (dB)	$\geq 80$	79.5	78.4	82.7	79.7	79.6	80.6
	PM ( $^\circ$ )	$\geq 60$	60.3	59.5	63.9	61.4	60.8	61.2
	UGB (MHz)	$\geq 100$	1.52	1.74	2.43	1.70	1.73	1.63
	PC ( $\mu W$ )	$\leq 5$	3.93	3.90	3.57	4.26	4.23	4.16
	RSR ( $V/\mu S$ )	$\geq 0.1$	0.25	0.24	0.24	0.24	0.27	0.24
	FSR ( $V/\mu S$ )	$\geq 0.1$	0.25	0.24	0.24	0.26	0.27	0.24
0.09 $\mu m$	Gain (dB)	$\geq 80$	80.5	78.9	82.9	79.3	79.3	83.6
	PM ( $^\circ$ )	$\geq 60$	60.8	60.0	62.8	59.9	59.9	62.5
	UGB (MHz)	$\geq 100$	1.66	1.61	2.05	1.59	1.59	1.67
	PC ( $\mu W$ )	$\leq 5$	3.68	3.72	3.14	3.62	3.62	4.67
	RSR ( $V/\mu S$ )	$\geq 0.1$	0.23	0.25	0.19	0.22	0.22	0.19
	FSR ( $V/\mu S$ )	$\geq 0.1$	0.22	0.24	0.19	0.22	0.22	0.28

### 3.2 Circuit design Examples



(a) 0.13  $\mu\text{m}$  CMOS technology



(b) 0.09  $\mu\text{m}$  CMOS technology

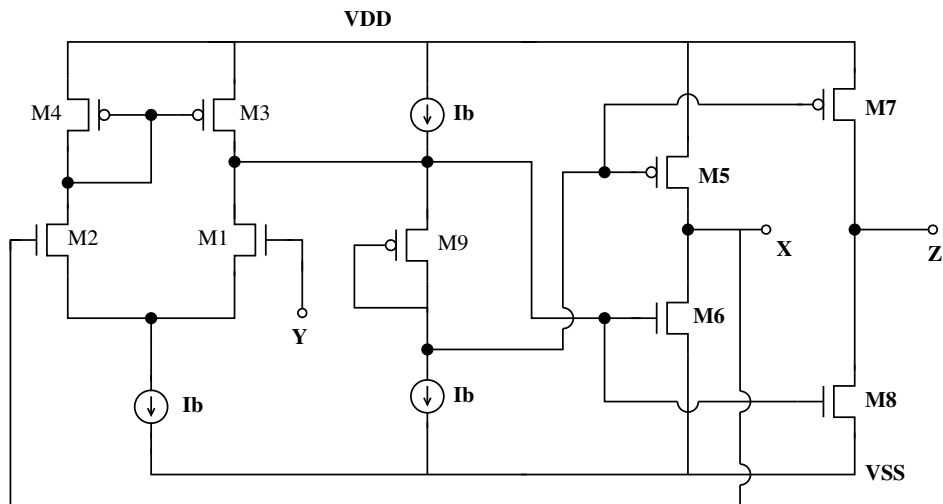
FIGURE 3.5: High-gain low-voltage bulk-driven OTA(Schematic-level): Variations in RMS error with circuit evaluations

**TABLE 3.11: High gain low voltage bulk-driven OTA: Performance comparison of algorithms based on 10 independent design trials.**

	Technology	ABC	GABC	MABC	PSO	EABC	MPSO
Average RMS error(%)	0.13 $\mu m$	0.69	1.18	0.16	0.37	0.51	<b>0.0</b>
	0.09 $\mu m$	0.18	0.85	0.48	0.88	0.04	<b>0.0</b>
Worst RMS error(%)	0.13 $\mu m$	1.64	3.98	1.42	3.74	1.88	<b>0.0</b>
	0.09 $\mu m$	0.96	4.16	2.46	5.51	0.42	<b>0.0</b>
Average circuit evaluations	0.13 $\mu m$	4148	4096	1991	2012	3950	<b>1572</b>
	0.09 $\mu m$	3212	4700	3097	2002	2396	<b>932</b>
Average design time (Minutes)	0.13 $\mu m$	55.8	53.5	28.4	32.8	53.6	<b>27.3</b>
	0.09 $\mu m$	44.5	59.9	38.0	28.8	32.7	<b>13.9</b>
Number of times zero RMS error is obtained	0.13 $\mu m$	3	4	8	8	6	<b>10</b>
	0.09 $\mu m$	7	3	6	8	9	<b>10</b>

- The MPSO algorithm is the most effective for designing OTA in both 0.13  $\mu m$  and 0.09  $\mu m$  technologies. It had designed OTA in both technologies all 10 times satisfying all the specifications. Further, average design times were found 27.3 and 13.9 minutes for 0.13  $\mu m$  and 0.09  $\mu m$  technologies, respectively.
- The MABC and PSO algorithms designed OTA in 0.13  $\mu m$  technology more effectively than EABC algorithm. However, in 0.09  $\mu m$  technology, EABC performed better.
- In 0.09  $\mu m$  technology, EABC has designed OTA nine times with zero error, while this number is 6 for the MABC algorithm.
- In the both technologies, the performance of EABC algorithm is superior to the ABC algorithm.
- The worst design error for the EABC algorithm in 0.09  $\mu m$  was 0.42% against 2.46% for MABC algorithm, 4.16% for GABC algorithm, 5.51% for PSO algorithm and 0.96% for ABC algorithm.

### 3.2.3 Second generation current conveyor (CCII+)



**FIGURE 3.6: Second generation current conveyor (CCII+)**

The current conveyor circuit is a building block for the current mode circuits. The current conveyor circuit was proposed by Sedra [48]. It can be utilized for building current to voltage converter, voltage to current converter, current amplifier, gm-c filter, VCO, amplifier, current integrator and, current differentiator. Further, it is a building block of current feedback operational amplifier and active filters [49]. Here, the automatic schematic-level design of the second generation positive current conveyor (CCII+) is presented. The circuit of the designed low-voltage current conveyor is shown in Fig. 3.6 [50]. The relation between inputs and output of the CCII+ can be described as follows,

$$\begin{bmatrix} I_y \\ V_x \\ I_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} V_y \\ I_x \\ V_z \end{bmatrix} \quad (3.2)$$

The ideal impedance of X-terminal is zero, while that for Z-terminal is infinite. The voltage gain between Y-terminal (input) and X-terminal,  $V_Y/V_Y$  is unity. Similarly, the current gain between input (X-terminal) and output (Z-terminal),  $I_Z/I_X$  is also unity. In the circuit of CCII+, when attempts are made to increase bandwidth, the impedance of X-terminal increases and that of Z-terminal decreases, which are undesirable effects. The design specifications for the CCII+ are as follows,

- $0.99 \leq \text{Voltage gain } (V_x/V_y) \leq 1.01$

- $0.99 \leq \text{Current gain } (I_z/I_x) \leq 1.01$
- Voltage bandwidth  $\geq 500MHz$
- Current bandwidth  $\geq 500MHz$
- Impedance of  $X$  terminal,  $Z_x \leq 50\Omega$
- Impedance of  $Z$  terminal,  $Z_z \geq 100K\Omega$
- Power consumption (PC)  $\leq 200\mu W$

**TABLE 3.12: Low voltage CCII: Search space for design variables.**

Parameter	Search space (0.13 $\mu m$ )	Search space (0.09 $\mu m$ )
Width( $W$ ) of all transistors	1 $\mu m$ to 100 $\mu m$	1 $\mu m$ to 100 $\mu m$
Length( $L$ ) of all transistors	0.2 $\mu m$ to 2 $\mu m$	0.1 $\mu m$ to 2 $\mu m$
$I_b$	1 $\mu A$ to 50 $\mu A$	1 $\mu A$ to 50 $\mu A$

The design parameters are width and length of various MOSFET transistors and value of the bias current  $I_b$ . The CCII+ is designed in 0.13  $\mu m$  and 0.09  $\mu m$  CMOS technologies using ABC, GABC, MABC, PSO, EABC and MPSO algorithms ten times independently. The value of the supply voltage is set to 0.8V for 0.13  $\mu m$  technology and that for 0.09  $\mu m$  technology is set to 0.6V. The average of the obtained specifications over ten independent design runs for various algorithms are shown in Table 3.13. The Table 3.14 represents various performance parameters for the comparison of the algorithms. In Fig. 3.7, the variations in the RMS error with the circuit evaluations are shown.

The following observations can be made from the obtained results.

- The MPSO algorithm designed CCII+ in the both technologies with minimum RMS error.
- The worst design error for the MPSO algorithm is also minimal compared to other algorithms in both technologies.
- The minimum average RMS error in 0.09  $\mu m$  technology is 15.9%. This indicates that all the considered algorithms have been failed to design CCII+ in 0.09  $\mu m$  CMOS technology.
- None of the algorithms is able to achieve all the specifications in any design-run.

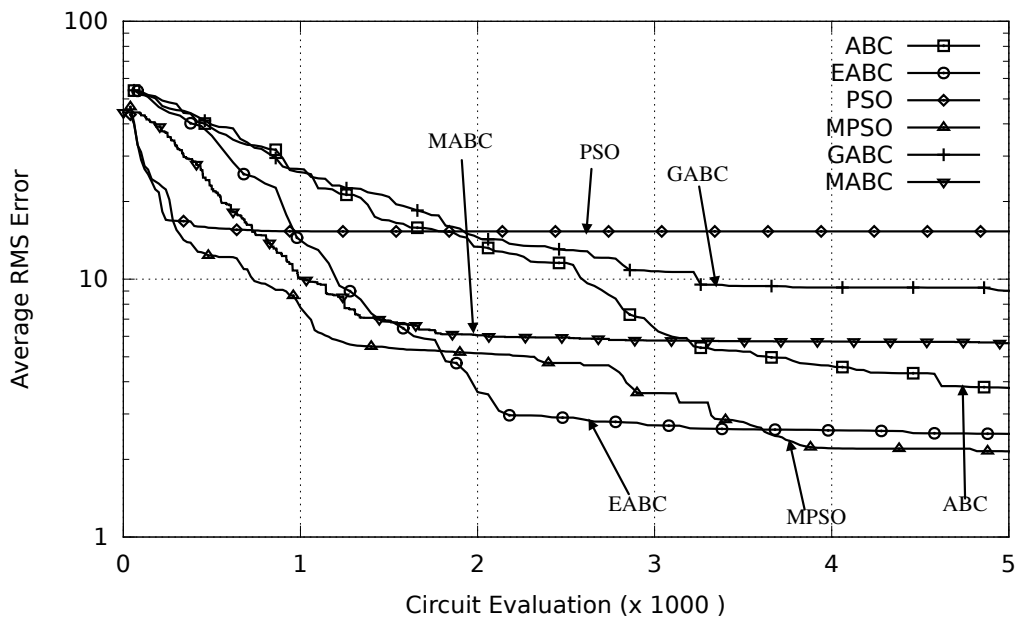
### 3.2 Circuit design Examples

**TABLE 3.13: High-gain low-voltage bulk-driven OTA: Average of obtained simulation results over 10 independent design trials.**

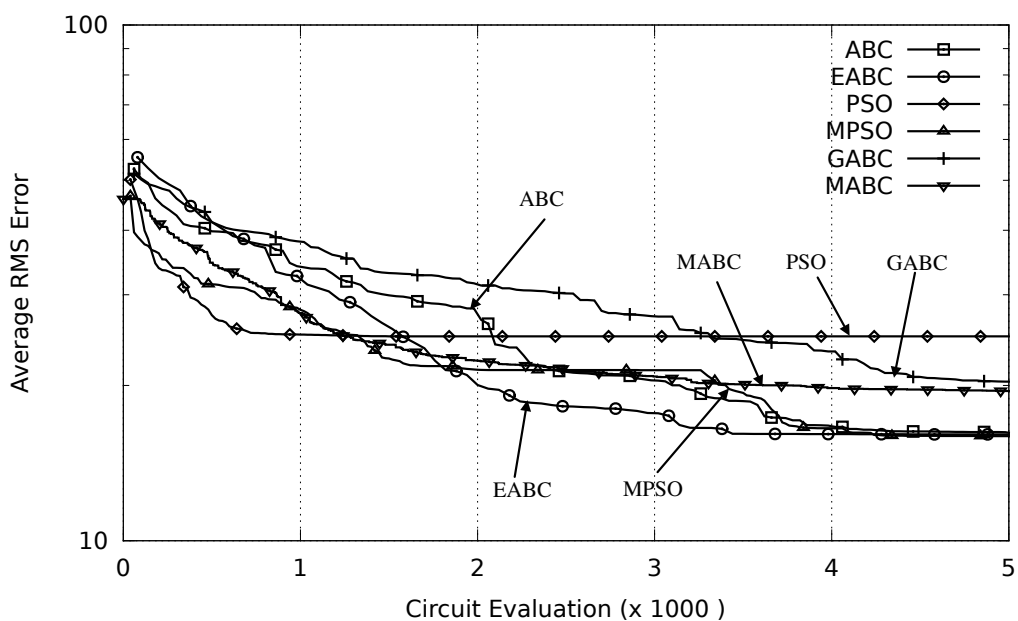
Technology	Specification	ABC	GABC	MABC	PSO	EABC	MPSO
0.13 $\mu m$	Voltage gain $V_X/V_Y$	0.99	0.99	0.99	0.99	0.99	0.99
	Current gain $I_Z/I_X$	0.99	0.99	0.99	0.99	0.99	0.99
	Voltage Bandwidth (MHz)	467	450	437	471	474	478
	Current Bandwidth (MHz)	717	660	784	623	724	713
	Impedance of $X$ terminal, $Z_X(\Omega)$	31.7	37.9	39.2	51.2	40.5	41.2
	Impedance of $Z$ terminal, $Z_Z(K\Omega)$	92.6	81.3	91.9	61.5	95.7	96.3
	Power consumption ( $\mu W$ )	99.5	100.3	58.6	110.3	102.7	110.7
0.09 $\mu m$	Voltage gain $V_X/V_Y$	0.99	0.99	0.99	0.99	0.99	0.99
	Current gain $I_Z/I_X$	0.99	0.99	0.99	0.99	0.99	0.99
	Voltage Bandwidth (MHz)	327	309	273	344	329	328
	Current Bandwidth (MHz)	530	499	505	513	528	520
	Impedance of $X$ terminal, $Z_X(\Omega)$	51.7	52.7	52.7	43.2	52.3	52.6
	Impedance of $Z$ terminal, $Z_Z(K\Omega)$	75.1	65.6	77.1	53.8	75.1	76.1
	Power consumption ( $\mu W$ )	60.2	61.7	37.2	93.5	60.7	61.7

**TABLE 3.14: Low voltage CCII: Performance comparison of algorithms based on 10 independent design trials.**

	Technology	ABC	GABC	MABC	PSO	EABC	MPSO
Average RMS error(%)	0.13 $\mu m$	3.78	9.02	5.64	15.32	2.51	<b>2.15</b>
	0.09 $\mu m$	16.14	20.34	19.51	24.91	16.08	<b>15.95</b>
Worst RMS error(%)	0.13 $\mu m$	8.86	15.34	6.84	22.46	3.36	<b>3.02</b>
	0.09 $\mu m$	16.79	36.61	22.63	49.44	16.58	<b>15.95</b>
Average circuit evaluations	0.13 $\mu m$	5000	5000	5000	5000	5000	5000
	0.09 $\mu m$	5000	5000	5000	5000	5000	5000
Average design time (Minutes)	0.13 $\mu m$	17.3	17.9	17.3	17.3	<b>17.0</b>	17.4
	0.09 $\mu m$	17.4	17.8	17.2	17.4	<b>16.9</b>	17.6
Number of times zero RMS error is obtained	0.13 $\mu m$	0	0	0	0	0	0
	0.09 $\mu m$	0	0	0	0	0	0



(a) 0.130  $\mu\text{m}$  CMOS technology



(b) 0.09  $\mu\text{m}$  CMOS technology

FIGURE 3.7: Second generation current conveyor(Schematic-level): Variations in RMS error with circuit evaluations

### 3.3 Observations

Following observations can be made from the conducted experiment of designing two-stage op-amp, high gain low voltage OTA and current conveyor circuit using the ABC, PSO, EABC, MPSO, GABC, and MABC algorithms in  $0.13 \mu m$  and  $0.09 \mu m$  CMOS technologies.

- The EABC algorithm has designed all three circuits in both technologies more efficiently than the ABC algorithm and its two variants GABC and MABC algorithms. Further, the complexity of the EABC algorithm is almost similar to that of ABC algorithm and both ABC and EABC algorithms have an equal number of parameters.
- The MPSO algorithm provides better performance than the PSO algorithm. The partial reinitialization scheme introduced in MPSO algorithm has successfully avoided saturation of algorithm and maintained swarm diversity. However, the MPSO algorithm has one more algorithmic parameter to control the reinitialization region compared to PSO algorithm.
- When the performances of MPSO and EABC algorithms are compared, the performance of MPSO algorithm is found better than that of EABC algorithm. Except for the op-amp design in  $0.13 \mu m$  CMOS technology, the MPSO algorithm provided better results compared to the EABC algorithm.
- Apart from the swarm size, the EABC algorithm has only one parameter (i.e.  $T_{max}$ ). On the other hand, MPSO algorithm has five parameters (i.e.  $C_1$ ,  $C_2$ ,  $W_h$ ,  $W_l$ ,  $S$ ). Thus, the MPSO algorithm can be tuned according to the nature of the problem. Such flexibility is not available with the EABC algorithm.

## CHAPTER 4

# Configurable layouts

In this chapter, the novel concept of the configurable layout is discussed. By passing appropriate parameters, the dimension of various components of layout and the distance between them can be changed. This flexibility of the configurable layouts makes them suitable for the layout-level design automation of the analog circuits.

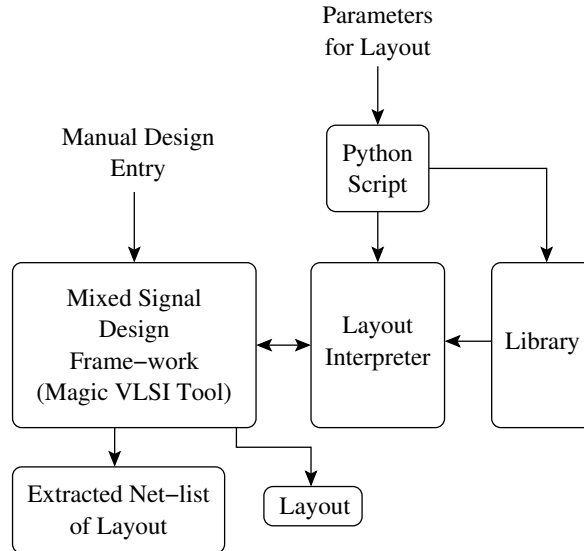
### 4.1 Design framework for configurable layouts

Traditionally, the layouts for the analog circuits are prepared manually using graphical interface. Once, the layout is prepared, the distance between various components of layout and their dimensions become fixed. If the changes are required in the later stage, it may require moderate to large efforts. Moreover, such layouts cannot be ported in other technologies directly. When the processes are improving continuously, the life time of such frozen layout is very short. Because of these difficulties, the task of the analog integrated circuit design is very much critical and development cycle of the analog integrated circuit is not time-efficient. In order to overcome these limitations, the concept of the configurable layout is proposed in this work. The configurable layouts are developed for the MAGIC VLSI Tool [51]. The design environment for the configurable layout is illustrated in Fig. 4.1.

The configurable layouts provide flexibility through which it is possible to change the dimensions of the dimensions of various layout components and distance between them

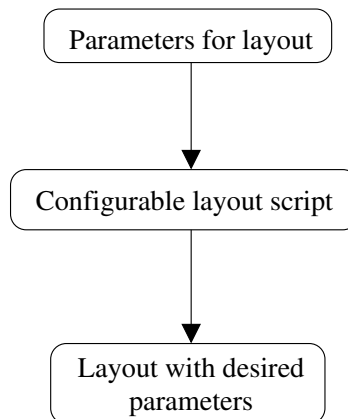
## 4.1 Design framework for configurable layouts

by changing the parameters of the layout. Further, the configurable layout can be ported to the other technologies very easily. Further, the design environment for the configurable layout is added to the design framework as an add-on and does not block any feature of the framework.



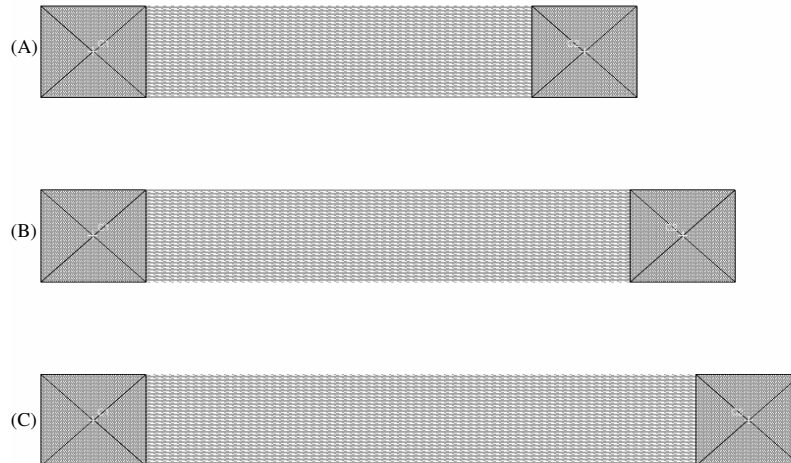
**FIGURE 4.1: User interface for configurable layout**

Unlike a traditional design approach for preparing a layout, the configurable layouts are described using the script containing specially designed macros. This script file is interpreted and converted into the design framework specific macros. Finally, design framework is triggered to generate the layout and netlist with all kind of the parasitics can be extracted for the simulation purpose. The script for the configurable layout is prepared using Python programming language [52] which is widely used user-friendly programming language. The use of such standard programming language allows the user to use all features of the language.



**FIGURE 4.2: Configurable Layout.**

In the script file of the configurable layout, the dimensions of the various layout components and distance between them are described in terms of the technology dependent variables. Further, the concept of configurable layout also supports hierarchical block oriented design methodology, which allows handling of the complex layout effectively. By passing appropriate parameters to configurable layout script, new layout according to desired parameters can be generated as shown in Fig. 4.2.



**FIGURE 4.3:** Layout of two contacts connected with metal layer. The distance between contacts : (A)  $15\mu\text{m}$  (B)  $18\mu\text{m}$  (C)  $20\mu\text{m}$

```
def con_simple(cellName='con_simple', distance=20, label=1) :
    startCell(cellName)
    x=0
    y=0
    #Placement of Terminal 'C1'
    pTerm(cellName=cellName, term=m1, x=0, y=0, name='C1',
    label=label)
    # Calculating distance between two contacts
    d=max(distance, poly_s, m1_s_m1)
    #Placement of Terminal 'C1'
    pTerm(cellName=cellName, term=m1, ref='C1', name='C2',
    xoff=d, label=label)
    #Placement of Contacts
    pCon(cellName=cellName, con=pc, w=4*mcon, h=4*mcon, ref='C1')
    pCon(cellName=cellName, con=pc, w=4*mcon, h=4*mcon, ref='C2')
    #Connecting contacts 'C1' and 'C2'
    pPath(cellName=cellName, layer=m1, ref1='C1', ref2='C2',
    width=4*mcon, direction='h_v')
    endCell(cellName)
```

**FIGURE 4.4:** Script for layouts of two contacts connected with each other using metal layer

In Fig. 4.3, three layouts of two contacts connected with each other through the metal layer are shown. These three layouts are generated using single configurable layout. Instead of describing the size of the metal contact using absolute value, it is described

**TABLE 4.1: Basic macros for configurable layout**

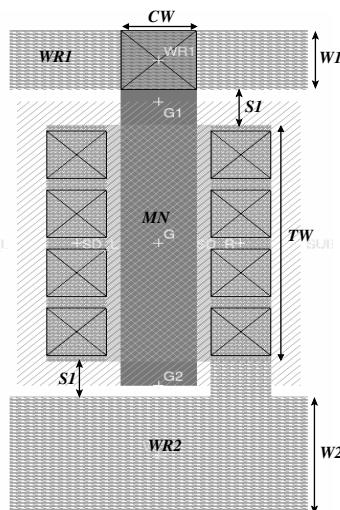
Macro	Description
startCell	Initialization of cell
endCell	End of cell
pCon	Place single contact
pMCon	Place multiple contact
pRect	Draw rectangular shape with desired layer
pPath	Draw path connecting two points
cCell	Generate cell of lower hierarchy
pCell	Place generated cell

using the technology dependent variable i.e. width and height of the metal contacts are four times greater than the minimum contact width supported by the technology. The distance between these two contacts is described using the parameter of the layout. Thus, by changing the value of this parameter, the distance between two contacts can be changed. For example, if it is desired to have distance between two contact  $15 \mu m$ , then the value of this parameter should be set to  $15 \mu m$ . In Fig. 4.4, the script file for this layout is illustrated. In Table 4.1, some of the implemented macros developed for the configurable layout are described. Table 4.2 illustrates some of the important technology variables. The technology dependent variables are described in the file named “Auxiliary technology file”. The porting of the layout in the other technology can be done by providing appropriate “Auxiliary technology file” for the target technology.

**TABLE 4.2: Some of the technology dependent variables**

Variable	Description
mcon	Minimum height or width of metal contact
pcon	Minimum height or width of polysilicon contact
dcon	Minimum height or width of diffusion contact
ndiff_s_pdiff	Minimum spacing between n-diffusion and p-diffusion
psub_s_nsub	Minimum spacing between p-substrate and n-substrate
p_o_diff	Minimum width of polysilicon over diffusion
well_w	Minimum width or length of n-well/p-well
m1_s_m1	Minimum spacing between metal-1 layers
m2_s_m2	Minimum spacing between metal-2 layers
poly_s	Minimum spacing between poly layers
m1_w	Minimum width of metal-1 layer
m2_w	Minimum width of metal-2 layer

Let’s consider simple configurable layout, in which the gate of NMOS transistor MN is connected to metal-1 wire WR1 and drain is connected to metal-1 wire WR2 as shown in Fig. 4.5. The parameters for this layouts are W1(width of wire WR1), W2(width



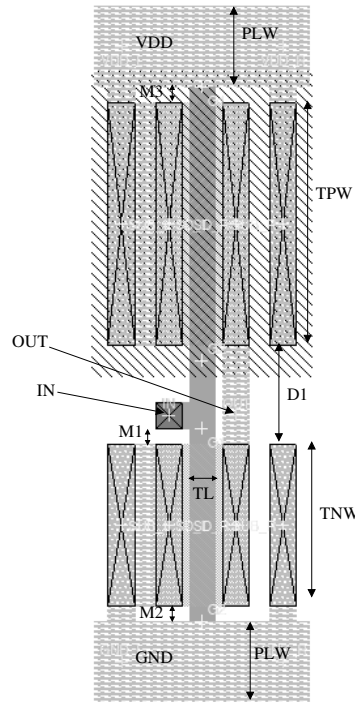
**FIGURE 4.5: Layout of NMOS transistor with gate and drain connected to metal wires.**

of wire *WR2*), *TW*(width of *MN*) and *TL*(length of *MN*). The NMOS transistor is generated as a lower level hierarchical block. The layout is generated as follows. First, wire *WR1* of metal-1 having width *W1* is placed. Then, the transistor *MN* is placed in a such a way that the distance between the upper edge of its drain-source and the lower edge of *WR1* becomes *S1*. The distance *S1* must be larger enough to avoid any DRC (Design Rule Check) error. Instead of selecting any absolute value for *S1*, technology dependent variable can be used. We use minimum spacing between two metal-1 layers supported by the technology as *S1*. The polysilicon contact is placed on wire *WR1*. The position for this placement can be calculated from the reference points of *MN*, i.e. “*G1*”. The height of the contact is kept same as *W1*, and its width *CW* is the maximum of *TL* and the minimum width of polysilicon contact supported by the technology. The polysilicon layer with a width equal to *TL* is used to connect the polysilicon contact with the gate of NMOS. Finally, the wire *WR2* is placed on another side of *MN*. The distance between the lower edge of the drain of *MN* and the upper edge of *WR2* is minimum spacing between metal-1 layers described by the technology. The drain of *MN* is connected with *WR2* using metal-1 layer. The width of this layer is equal to a minimum width of diffusion contact. Here, the position for placement of any component is decided based on the position of another component. Because of such relative placement, change in size or placement of one component causes a realignment of other components accordingly. For example, since the placement of *WR2* is in reference to the lower edge of *MN*, the change in the width of *MN* causes the corresponding change in the placement of *WR2*. The change in length of *TL* causes a corresponding change in the width of polysilicon contact. If the width of *WR1* is changed, then placements of both *MN* and *WR2* will be changed. Furthermore, the use of relative placement and technology dependent variables provide portability of layouts

## 4.2 Configurable layout of CMOS inverter

in other technologies. By providing appropriate auxiliary technology file, the layout can be systematically ported to other technology.

## 4.2 Configurable layout of CMOS inverter



**FIGURE 4.6: Layout of inverters.**

The layout of the CMOS inverter as shown in Fig. 4.6 is prepared in the following manner. The NMOS and PMOS transistors are generated as lower level hierarchy blocks. The parameters of the inverter are the length of both transistors ( $TL$ ), the width of NMOS ( $TNW$ ), width of PMOS ( $TPW$ ) and, width of the power line ( $PLW$ ). Initially, the layouts of lower hierarchical blocks i.e. NMOS and PMOS transistors are created. First, the placement of the NMOS transistor ( $MN$ ) is carried out. The placement of the PMOS transistor ( $MP$ ) is carried out with respect to the position of  $MN$ . Let's consider that the distance between  $MN$  and  $MP$  is  $D1$  as shown in Fig. 4.6. The value of  $D1$  should be sufficiently large to accommodate polysilicon contact for the input and to avoid any DRC error. For determining the value of  $D1$ , the minimum distance between n-diffusion and p-diffusion ( $NPD$ ) can be used. Thus, the value of the distance  $D1$  is the maximum of the  $NPD$  and minimum distance required to accommodate polysilicon contact for the input. The gates of the both transistors are connected using the polysilicon wire having the width of  $TL$ . The polysilicon contact for the input terminal is placed  $M1$  distance away from the upper edge of the  $MN$ . The

value of  $M1$  is equal to the minimum distance between two metal-1 layers supported by the technology. The size of this contact is equal to the minimum size supported by the technology. The drains of both transistors can be connected using metal-1 wire having the width equal to the width of source contact. The position of the polysilicon contact is determined with respect to the source contact of the  $MN$ . The power line of metal-1 for GND terminal of the inverter having the width of  $PLW$  can be placed  $M2$  distance away from the lower edge of the  $MN$ . The value of  $M2$  is the minimum distance between two metal-1 layers. Similarly, the power line for VDD can be placed with respect to  $MP$ . Finally, sources and substrates of both transistors are connected to corresponding power lines.

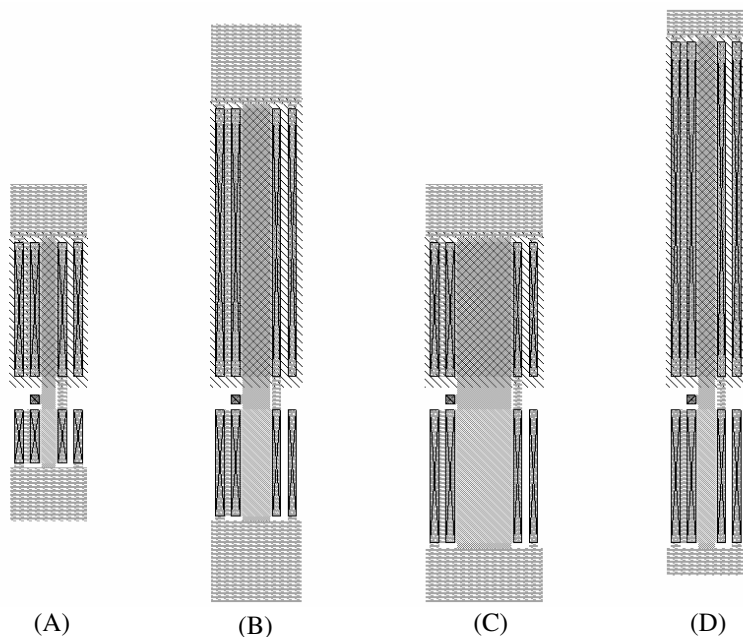


FIGURE 4.7: Layout of inverters with different parameters.

TABLE 4.3: Parameters of configurable inverter layout

Parameter	(A)	(B)	(C)	(D)
TNW ( $\mu m$ )	2	4	5	5
TPW ( $\mu m$ )	5	10	5	12.5
TL ( $\mu m$ )	0.5	1	2	0.6
PLW ( $\mu m$ )	2	3	2	1

Here, placement of various components of the layout and their dimensions are relative. If the width of  $MN$  is modified by the user by changing the  $TNW$  parameter of the layout, automatically placement of GND power line will be changed. Similarly, if the length of the transistors is changed, the width of the polysilicon connecting gates of two transistors is also changed automatically. Thus, passing the appropriate values of parameters, the desired layout can be generated instantly. In Table 4.3, the different

### 4.3 Summary

set of parameters for layout are shown and in Fig. 4.7, corresponding layouts generated using the configurable layout method are shown.

## 4.3 Summary

With the help of the configurable layouts, the layout can be modified instantly by passing appropriate value of the parameters. This provides the flexibility of the schematic to the layouts and makes them suitable for algorithm based design automation. Further, it also provides facility to implement algorithms for the customize routing in the layout.

## CHAPTER 5

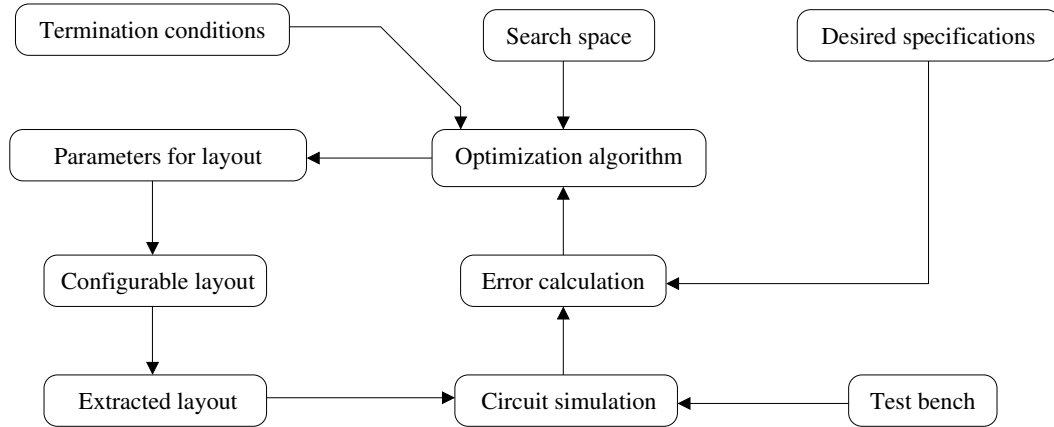
# Parasitic-aware automatic circuit design

In this chapter, the parasitic-aware design of the analog CMOS circuit is discussed. The parasitic-aware design for CMOS circuit is achieved with the help of the configurable layouts and evolutionary algorithms. First, the environment for the automatic circuit design is discussed which is followed by the circuit design examples. The parasitic-aware design of the ring-oscillator, CMOS buffer chain, VCO, two-stage op-amp and OTA are discussed. Further, in the process variations are also considered while designing op-amp and OTA.

### 5.1 Environment for parasitic-aware automatic circuit design

The schematic-level design of the analog CMOS integrated circuits has been done using the evolutionary algorithms. However, when the layouts are prepared from these optimized schematics, the parasitic components are added in the circuit. These parasitic components deteriorate the performance of the circuit. Further, estimating the exact value of the layout parasitic is very difficult. In [2], the parasitic models are used to include the layout parasitics in the circuit design process and demonstrated the design of the ring oscillator. However, the accuracy of such method depends upon

## 5.1 Environment for parasitic-aware automatic circuit design



**FIGURE 5.1: Conceptual block diagram of optimizer for the parasitic-aware automatic CMOS circuit design.**

the accuracy of the parasitic models.

In order to overcome this problem, the concept of the parasitic-aware automatic circuit design is proposed. In the proposed concept, the concept of the automatic circuit design at the schematic-level is extended to the layout-level using the configurable layout. In the schematic-level design automation, the optimization algorithm generates parameters for the design. Based on these generated parameters, the netlist of the schematic is updated and simulated against pre-determined test-bench. This is followed by the analysis of the simulation results and calculation of the RMS error. The layouts prepared using the traditional design approach are rigid, making changes automatically in the layout according to the design parameters like the schematics is not possible and it requires human intervention. With such frozen layout, design automation at layout-level is not feasible. These difficulties associated with the design automation at layout-level can be overcome by using configurable layouts. Unlike a traditional handcrafted layout, the dimensions of and the distances between various components of the configurable layout can be modified by changing the value of the layout parameters. This provides the layout flexibility of the schematic.

In Fig. 5.1, the conceptual block diagram describing the parasitic-aware automatic circuit design is illustrated. Based on the search-space of the design parameters, optimization algorithm generates the set of parameters for the layout. These parameters are passed to the configurable layout and a new layout is generated accordingly instantly without any human intervention. The layout is then extracted to netlist using layout extraction tool provided by the design framework. Such netlist contains all the parasitic components of the layout. This netlist is simulated and its performance is evaluated. Based on the simulation results and desired (target) specifications of the circuit, RMS design error represented by Eq. 3.1 is calculated. The optimization

algorithm uses this RMS error for generating a new set of the parameters.

In the proposed concept of the parasitic-aware automatic circuit design, extracted netlist containing all parasitics is utilized in the design process from the beginning of the circuit design instead of the simple schematics. Thus, without using any estimation or models, parasitic aware design can be achieved. Further, parasitic-aware automatic circuit design process does not require any human intervention.

In this work, parasitic-aware design of the following circuits are demonstrated

1. Ring oscillator
2. CMOS buffer chain
3. Voltage controlled oscillator
4. Two-stage operational-amplifier
5. Bulk-driven OTA
6. Enhanced bulk-driven OTA

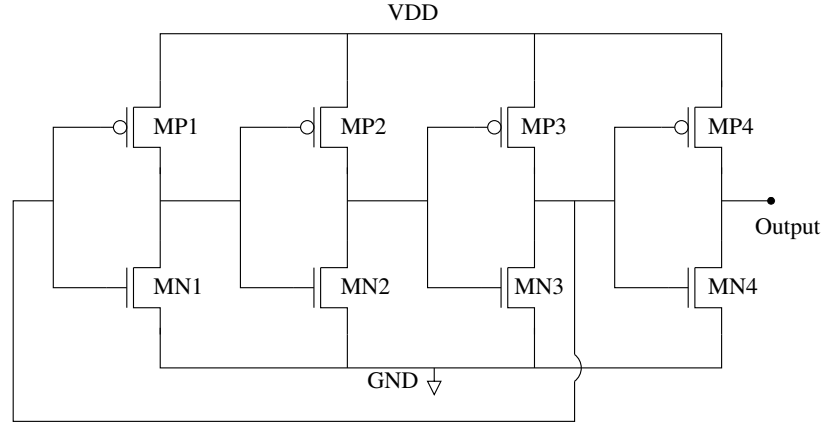
The parameters for the various algorithms are set as discussed in section 3.2. The specifications of computer used for the experiment has specifications as discussed in section 3.2. For the circuit simulation NGSPICE [41] is used, while for layout-extraction built-in extraction tool of MAGIC VLSI is utilized.

## 5.2 Ring-oscillator

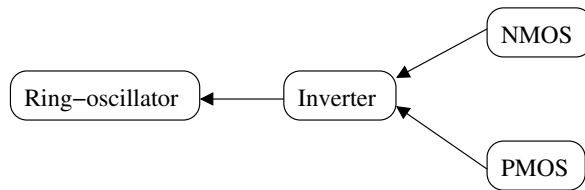
The parasitic aware design of the ring-oscillator is carried out in  $0.13\ \mu\text{m}$  CMOS technology with the help of ABC, PSO, EABC and MPSO algorithms. The circuit of the ring-oscillator is shown in Fig. 5.2. It contains four inverters. The first three inverters form oscillator while remaining one near output is used to remove the effect of the load capacitor on the oscillation frequency. In [2], the parasitic-aware design of ring-oscillator is demonstrated. However, it uses parasitic-models.

The ring-oscillator is designed to oscillate at the 1.5GHz frequency and VDD voltage

## 5.2 Ring-oscillator



**FIGURE 5.2: Circuit of Ring-oscillator.**



**FIGURE 5.3: Ring-oscillator: Hierarchy used in configurable layout.**

is set to 1.2V. The design parameters are width and length of various transistors. For the simplicity and to maintain the symmetry of the layout, sizes of all the inverters are kept same. The search space for the design parameters is shown in Table 5.1. The maximum circuit evaluations are set to 1000.

**TABLE 5.1: Ring-oscillator: Search-space for design parameters.**

Parameter	Search-space
Width of PMOS transistor	$2\mu m$ to $20\mu m$
Width of NMOS transistor	$2\mu m$ to $20\mu m$
Length of all transistor	$0.2\mu m$ to $2\mu m$

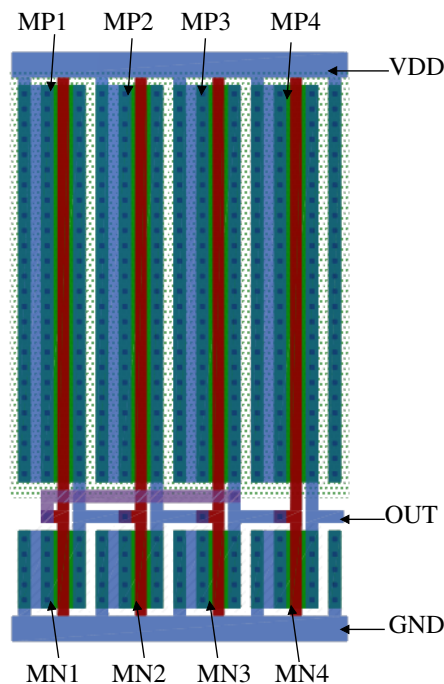
In the first experiment, the design of the ring-oscillator is carried out at schematic-level using the ABC algorithm. We carried out ten independent design trials. The obtained results show that the ring-oscillator is designed successfully and the average value of the output frequency is 1.499GHz. However, when the layouts are prepared from these optimized schematics, the post-layout simulation of these layouts show the average of the output frequency is 1.38GHz. This variation is due to the added parasitics by the layout which is not possible to consider during schematic-level design.

In the second experiment, the proposed concept of the parasitic-aware automatic circuit design based on the configurable layout is used. The parasitic-aware design is also carried out ten times using ABC, PSO, EABC and MPSO algorithms. The average of

**TABLE 5.2: Ring-oscillator: Average of the obtained results.**

Algorithm	Frequency	RMS error	Time
ABC	1.50 GHz	0.048%	18.4 Min
PSO	1.50 GHz	0.042%	15.7 Min
EABC	1.50 GHz	0.005 %	15.6 Min
MPSO	1.50 GHz	0.001%	15.6 Min

the obtained results is considered for the comparison. In the Fig. 5.3, hierarchy used for the preparing the configurable layout of the ring-oscillator is shown. In Table 5.2, the average value of the oscillation frequency, RMS error and design time are illustrated. The MPSO algorithm designed oscillator with 0.001% RMS error, which is followed by EABC, PSO and ABC algorithm. In Fig. 5.4, the layout of one of the designs by the MPSO algorithm is illustrated. The output waveform obtained by simulating this waveform is shown in Fig. 5.5. The convergence speed of various algorithms can be compared using Fig. 5.6. The average design time for the MPSO algorithm is only 15.6 minutes.

**FIGURE 5.4: Layout of ring-oscillator.**

The parasitic-aware design is an iterative process. In each iteration, the layout is created using configurable layout; netlist with parasitic is extracted using the extraction tool; the netlist is simulated and decisions are taken as per optimization algorithm. Thus, the time required to complete the single iteration involves summation of time required by configurable layout to generate the layout of the circuit, netlist extraction time, simulation time and time required by the optimizer to take decisions. The

## 5.2 Ring-oscillator

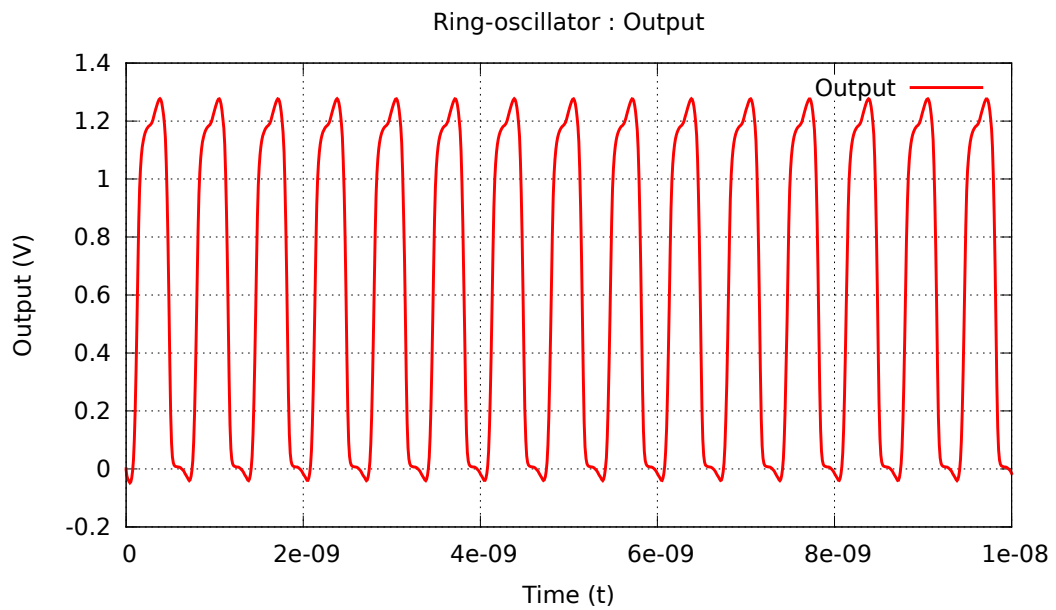


FIGURE 5.5: Ring-oscillator: Output waveform.

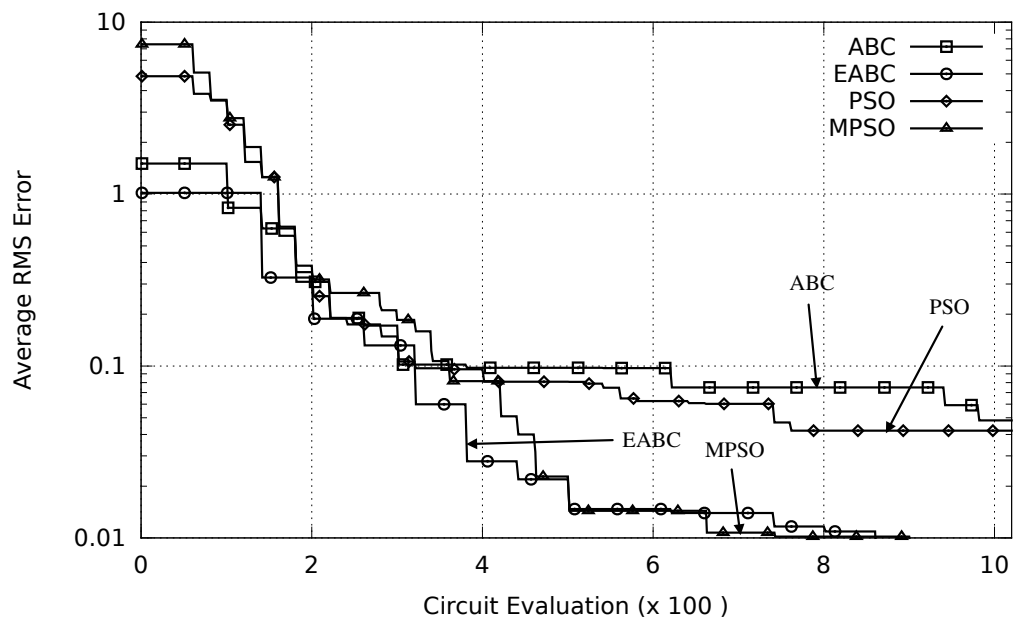
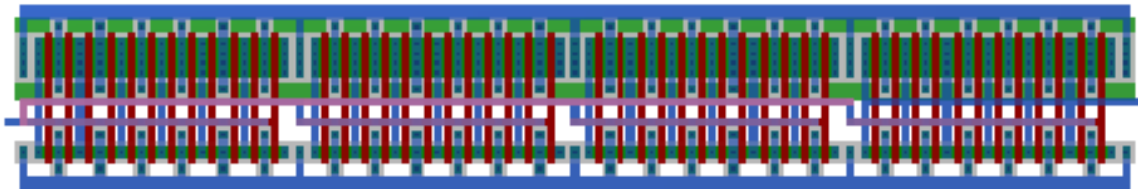


FIGURE 5.6: Ring-oscillator: Variations in RMS error with circuit evaluations.

simulation of the circuit based on the extracted netlist is carried out by the external circuit simulator (NG-spice). The time required in the simulation of the parasitic-aware net-list is quite large compared to the time required in other steps. The simulation time depends upon the number of elements in the net-list and the complexity of the circuit.



**FIGURE 5.7: Ring-oscillator with finger transistors (parasitic-aware design): Layout of best design obtained from MPSO algorithm.**

In this work, in order to investigate the effect of the layout-topology (floor-planning and placement), we have carried out the design of the Ring-oscillator with two different floor-planning and placement. The result of the first design case is illustrated in 5.4 and Table 5.2. In this experiment, we have considered the simplest possible layout. In the second experiment, we have considered different layout-topology with the finger transistors. The specifications and the experiment environment are kept same as the first experiment. In this experiment, we have designed the Ring-oscillator using ABC, PSO, EABC, and MPSO algorithms. In the Figure 5.7, the layout of the obtained ring oscillator is illustrated. The obtained results of the second experiments, illustrated in Table 5.3, indicate that the design time is increased along with the average design error.

**TABLE 5.3: Ring-oscillator with finger transistors: Average of the obtained results.**

Algorithm	Frequency	RMS error	Time
ABC	1.50 GHz	0.026%	31 Min
PSO	1.50 GHz	0.049%	27 Min
EABC	1.50 GHz	0.015 %	32 Min
MPSO	1.50 GHz	0.011%	28 Min

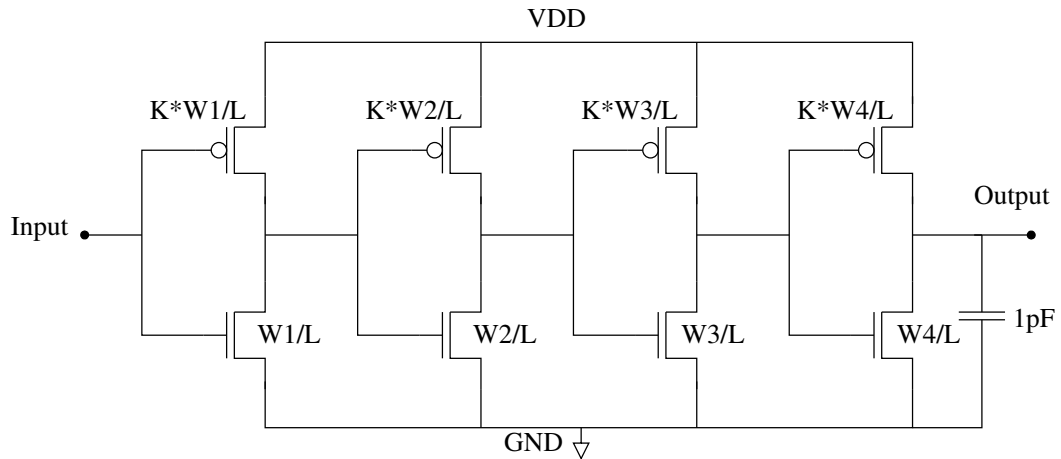
Due to the complex layout and use of the finger transistors, the number of the components in the parasitic-aware net-list is increased. This not only changes the cost function for the optimization algorithm but also changes simulation time. Thus, the performance of the optimization algorithms and the convergence speed may be affected.

In this work, for the other circuits, we have designed and optimized a circuit for a

### 5.3 CMOS buffer chain

given specification and for a particular layout topology. The layout topology is kept fixed during design cycle of a circuit. Layout topology is not considered as a part of optimization problem. But, the above example of the Ring-oscillator, clearly demonstrates that the layout topology affects design of circuits and results of optimization algorithm.

## 5.3 CMOS buffer chain



**FIGURE 5.8: Circuit of CMOS buffer chain.**

The CMOS buffer chain is generally used to drive the large capacitive load in the digital integrated circuits. The circuit of the buffer chain is shown in Fig. 5.8. It contains four inverters. The buffer chain is designed at layout-level in  $0.13 \mu m$  CMOS technology using ABC, EABC, PSO and MPSO algorithms. The supply voltage is set to  $1.2V$  and load capacitor is  $1pF$ . The design specifications are as follows:

- Rise time  $\leq 0.15nS$
- Fall time  $\leq 0.15nS$
- Peak power consumption (PC)  $\leq 0.25mW$

The design parameters are the width of various transistors. The width of the PMOS transistor in the inverter is kept  $K$  times higher than that of NMOS transistor, where  $K$  is an integer number. The width of the NMOS of the first stage inverter is fixed and set to  $0.5 \mu m$ , while the length of all the transistor is kept  $0.13 \mu m$ . The search-space for the design parameters are shown in Table 5.4. In the layout of buffer chain, the

smallest possible transistor width is  $0.5 \mu m$  and largest possible width is  $30 \mu m$ . In order to maintain good aspect-ratio of the layout and to make it more compact, finger transistors are utilized in buffer chain. Further, the finger transistor has fewer parasitics compared to a single transistor with equivalent width. The maximum number of the circuit evaluations during the automatic circuit design process is set to 3000.

**TABLE 5.4: CMOS buffer chain: Search-space for design parameters.**

Design parameter	Search-space
$W_2$	$1 \mu m$ to $30 \mu m$
$W_3$	$1 \mu m$ to $30 \mu m$
$W_4$	$1 \mu m$ to $30 \mu m$
$K$	1, 2, 3

In the first experiment, the buffer chain is designed at schematic level using the ABC algorithm. We took ten independent design trials. The obtained results indicated that the circuit is designed at schematic-level successfully all 10 times i.e. average RMS error is 0%. The average of the obtained specifications over ten independent design trials are described in Table 5.5. Now, from these optimized schematics, the layouts are prepared for all ten design trials and post-layout simulations were carried out. The post-layout simulation indicated that the average RMS error was jumped to 10.8%. This is due to added parasitic components in layout which were not possible to consider in schematic-level design.

**TABLE 5.5: CMOS buffer chain (schematic-level design using ABC algorithm): Average of results from schematic-level simulations and corresponding post-layout simulation.**

Specification	Schematic-level simulation	Post-layout simulation
Rise-time $T_r$ (ns)	0.146	0.149
Fall-time $T_r$ (ns)	0.085	0.091
PC (mW)	0.229	0.296
RMS Error(%)	0.0	10.82

In the second experiment, the parasitic-aware design of CMOS buffer chain at layout-level is carried out using ABC, PSO, EABC and MPSO algorithms. Using each algorithm, buffer chain is designed ten times independently and the average of obtained simulation results are considered for the comparison and shown in Table 5.6.

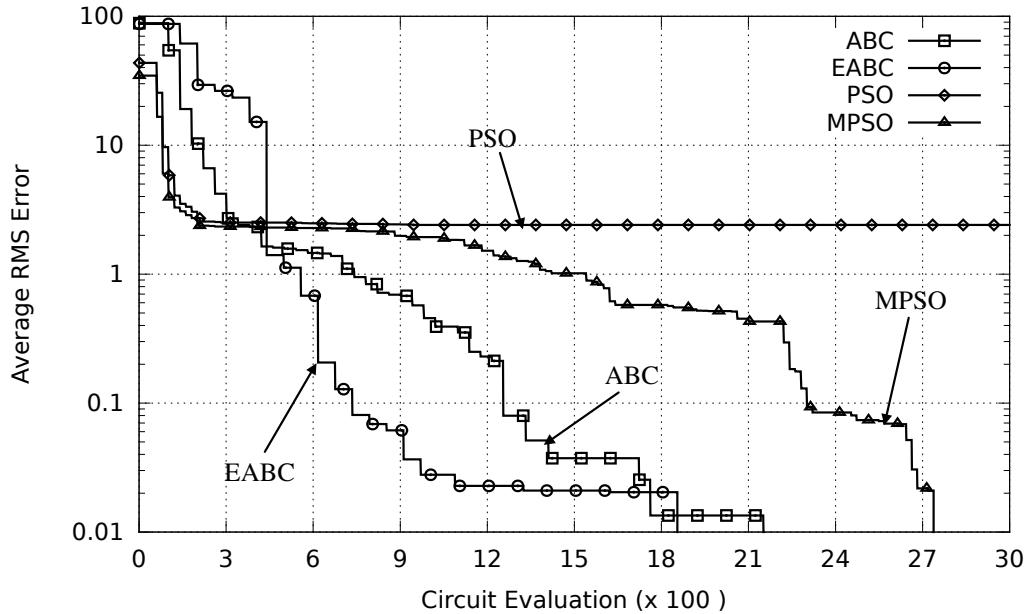
Except for the PSO algorithm, all other algorithms are able to design buffer chain with zero RMS error. However, the EABC algorithm is found most effective designing buffer chain at layout-level with average design time of only 29.56 minutes. In Fig. 5.9, the

## 5.4 Voltage controlled oscillator (VCO)

**TABLE 5.6: CMOS buffer chain (parasitic-aware design): Average of simulation results.**

Specification	ABC	PSO	EABC	MPSO
Rise-time $T_r$ (ns)	0.135	0.154	0.134	0.149
Fall-time $T_r$ (ns)	0.086	0.115	0.086	0.095
PC (mW)	0.216	0.257	0.216	0.240
RMS Error(%)	0.0	2.41	0.0	0.0
Desin Time (Minutes)	46.9	101.42	29.56	71.96

variations in average RMS error with the circuit evaluations are shown. In Table 5.7, various design parameters obtained by different algorithms for their best design trials are described. In Fig. 5.10, the layout generated in the best design run by the EABC algorithm is illustrated.



**FIGURE 5.9: CMOS buffer chain: Variations in average RMS error with circuit evaluations.**

## 5.4 Voltage controlled oscillator (VCO)

In this work, the parasitic-aware design of the Lee-Kim delay cell [53] based VCO is demonstrated in  $0.13\ \mu\text{m}$  and  $0.35\ \mu\text{m}$  CMOS technologies. The circuit diagram of the VCO is shown in Fig. 5.11. The use of Lee-Kim delay cell provides very good linearity to VCO. However, the disadvantage is that, the output of such VCO does not have 50% duty cycle. In order to overcome this problem, an additional circuit called duty-cycle

TABLE 5.7: CMOS buffer chain: Design parameters for best design run.

Design parameter	ABC	PSO	EABC	MPSO
$W_2$ ( $\mu m$ )	1.1	1.0	1.1	1.1
$W_3$ ( $\mu m$ )	3.0	2.1	3.0	3.4
$W_4$ ( $\mu m$ )	29.3	29.3	29.3	29.3
$K$	1	1	1	1

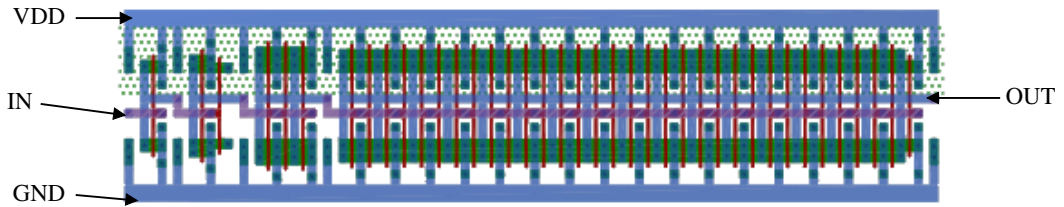


FIGURE 5.10: CMOS buffer chain: Layout of best design obtained from EABC algorithm.

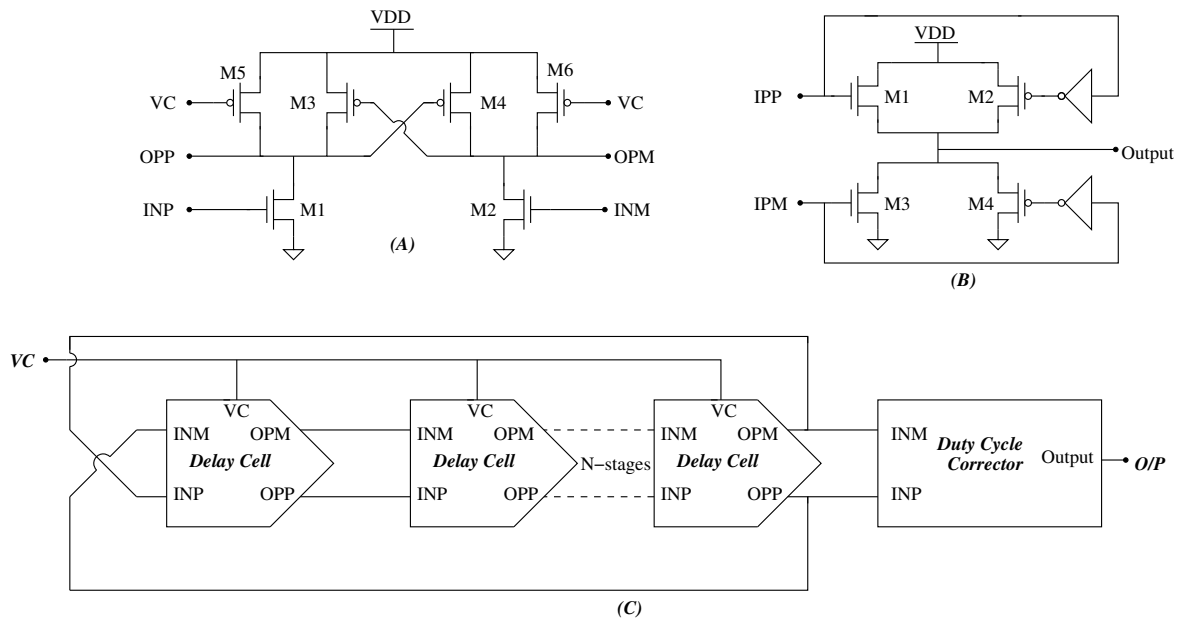
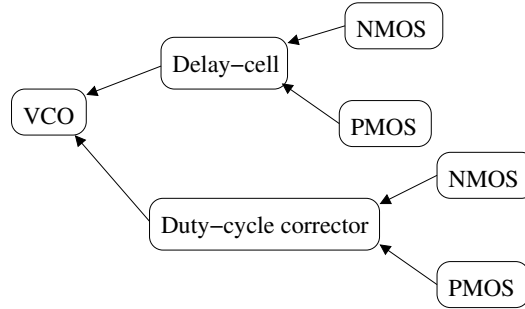


FIGURE 5.11: Circuit of VCO: (A) Lee-Kim delay cell (B) Duty-cycle corrector (C) VCO.

## 5.4 Voltage controlled oscillator (VCO)

corrector is needed. In order to prepare configurable layout, the hierarchy as shown in Fig. 5.12 is used.



**FIGURE 5.12: VCO: Hierarchy used in configurable layout.**

The VCO is designed to oscillate between two frequencies. The design specifications for VCO are as follows,

- Lower frequency of oscillation  $f_L$  is between 0.25 to 0.3GHz
- Upper frequency of oscillation  $f_H$  is between 1.0 to 1.1GHz.

The supply voltage (VDD) is set to 1.2V and 3.3V for  $0.13\ \mu m$  and  $0.35\ \mu m$  CMOS technologies, respectively. The maximum circuit evaluations are set to 1000. The design parameters are width and length of various transistors and the number of delay-cells. The design parameters with the corresponding search-space are shown in Table 5.8.

**TABLE 5.8: VCO: Search-space for design parameters.**

Design parameter	Search-space for $0.13\ \mu m$ technology	Search-space for $0.35\ \mu m$ technology
$W_P$	$1\ \mu m$ to $10\ \mu m$	$2\ \mu m$ to $10\ \mu m$
$W_N$	$1\ \mu m$ to $10\ \mu m$	$2\ \mu m$ to $10\ \mu m$
$L$	$0.2\ \mu m$ to $2\ \mu m$	$0.4\ \mu m$ to $2\ \mu m$
Number of delay cells	2 to 8 (even numbers only)	2 to 8 (even numbers only)

In the first experiment, VCO is designed at schematic level using ABC algorithm ten times in both technologies. The average of the obtained specifications over these design runs are shown in Table. 5.9. The results indicate that ABC algorithm is able to design VCO in both technologies with zero average RMS error i.e. in all ten design trials, VCO is designed at schematic level satisfying all the specifications. Next, the device dimensions provided by these schematic-level design are used to prepare the layout of VCO. When these layouts are simulated, the simulation results indicate that the average RMS error is jumped to 7.9% in case of  $0.13\ \mu m$  technology. The average RMS

error is jumped to 22.3% for  $0.35\mu m$  technology. This is due to the added parasitics in the circuit due to the layout.

**TABLE 5.9: VCO (schematic-level design using ABC algorithm): Average of results from schematic-level simulations and corresponding post-layout simulation.**

Specification	0.13 $\mu m$ technology		0.35 $\mu m$ technology	
	Schematic-level simulation	Post-layout simulation	Schematic-level simulation	Post-layout simulation
$f_L$ (GHz)	0.3	0.26	0.28	0.22
$f_H$ (GHz)	1.04	0.89	1.04	0.81
RMS Error(%)	0.0	7.9	0	22.3

In the second experiment, the parasitic-aware automatic circuit design approach is used to design VCO. The ABC, EABC, PSO and MPSO algorithms are utilized to design VCO in  $0.13\mu m$  and  $0.35\mu m$  technologies. Each algorithm is used to design VCO ten times and the average of the obtained results are used for the comparison.

**TABLE 5.10: VCO (parasitic-aware design): Average of simulation results.**

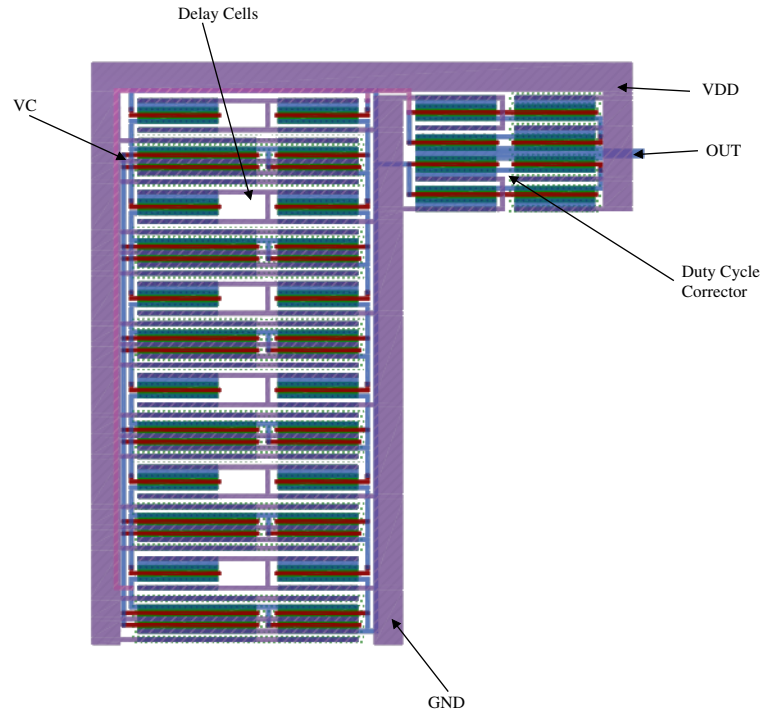
Specification	0.13 $\mu m$ CMOS technology				0.35 $\mu m$ CMOS technology			
	ABC	PSO	EABC	MPSO	ABC	PSO	EABC	MPSO
$f_L$ (GHz)	0.27	0.28	0.28	0.28	0.26	0.26	0.28	0.28
$f_H$ (GHz)	1.03	1.03	1.02	1.04	1.01	1.01	1.03	1.01
RMS Error(%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Time (Minutes)	16.2	4.6	10.3	7.9	10.6	4.1	10.4	4.6

**TABLE 5.11: VCO: Design parameters for best design trial**

Design parameter	0.13 $\mu m$ CMOS technology				0.35 $\mu m$ CMOS technology			
	ABC	PSO	EABC	MPSO	ABC	PSO	EABC	MPSO
$W_P$ ( $\mu m$ )	10.0	8.1	6.7	10.0	10	10	8.8	5.2
$W_P$ ( $\mu m$ )	5.1	5.5	4.5	6.1	4.7	4.7	4.2	2.8
$L$ ( $\mu m$ )	0.4	0.3	0.3	0.3	0.5	0.5	0.5	0.5
$K$	4	6	6	6	4	4	4	4

In Table 5.10, the average of the obtained simulation results are described. All algorithms are able to design VCO at layout-level successfully with zero average error i.e. satisfying all the design specifications. However, the PSO algorithm is fastest one in both technologies. It took the average time of 4.6 minutes to design VCO at layout-level which is followed by the MPSO, EABC and ABC algorithms in  $0.13\mu m$

## 5.5 Two-stage operational amplifier



**FIGURE 5.13: VCO: Layout of best design obtained from PSO algorithm in  $0.13\ \mu\text{m}$  CMOS technology.**

technology. In  $0.35\ \mu\text{m}$  technology, PSO algorithm took only 4.1 minutes to design VCO. In Table 5.11, the design parameters obtained in the best design trial by each algorithm are shown. In Fig. 5.13 and Fig. 5.14 the layouts generated during best design run by the PSO algorithm in both technologies are illustrated.

## 5.5 Two-stage operational amplifier

In this work, the parasitic-aware automatic circuit design of the two-stage operational amplifier is carried out in  $0.13\ \mu\text{m}$  and  $0.35\ \mu\text{m}$  CMOS technologies. The circuit of op-amp is shown in Fig. 5.15. The op-amp is designed for the following specifications,

- Gain  $\geq 80\text{dB}$
- Unity gain bandwidth (UGB)  $\geq 100\text{MHz}$
- Phase margin  $\geq 60^\circ$
- Power consumption  $\leq 30\ \mu\text{W}$  for  $0.13\ \mu\text{m}$  and  $80\ \mu\text{W}$  for  $0.35\ \mu\text{m}$

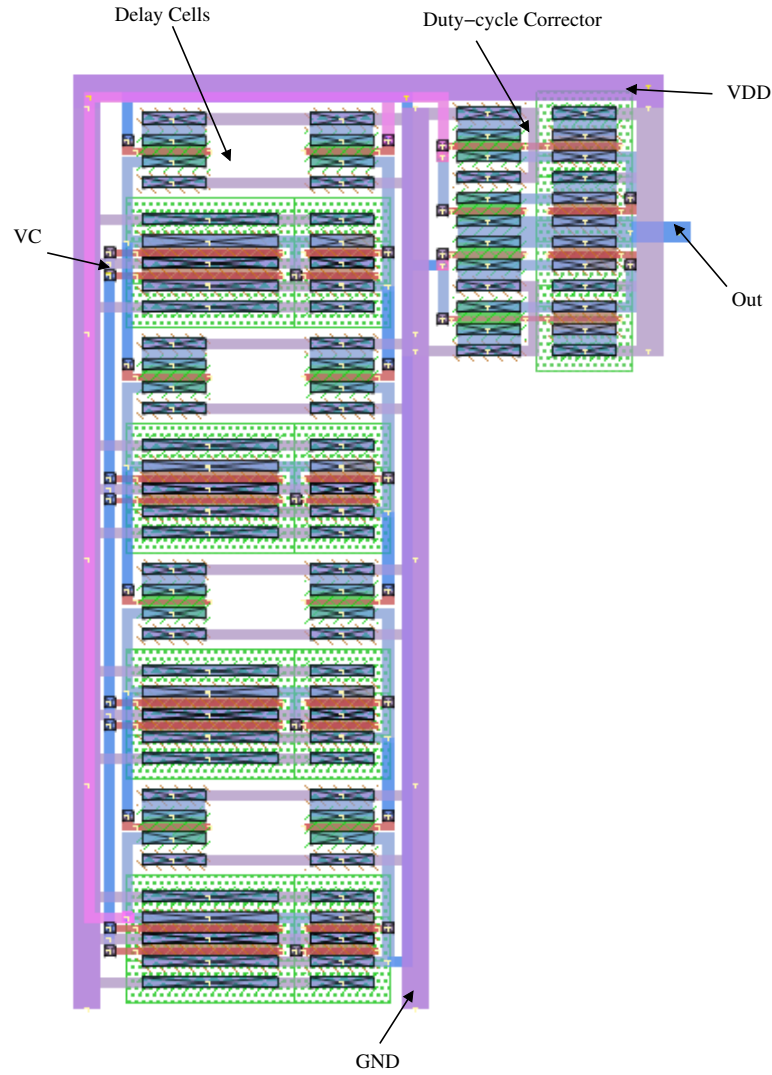


FIGURE 5.14: VCO: Layout of best design obtained from PSO algorithm in  $0.35\mu\text{m}$  CMOS technology.

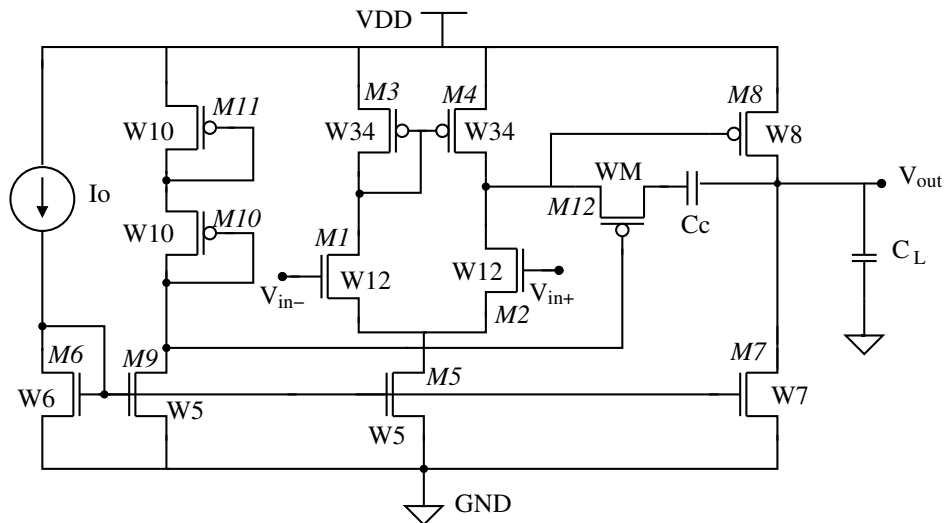


FIGURE 5.15: Circuit of two-stage operational amplifier.

## 5.5 Two-stage operational amplifier

- Rise and fall slew rate  $\geq 35V/\mu S$
- Power supply rejection ratio (PSRR)  $\geq 75dB$
- Common mode rejection ratio (CMRR)  $\geq 80dB$

**TABLE 5.12: Two stage op-amp: Search-space for design parameters.**

Design parameter	Search-space for $0.13\mu m$ technology	Search-space for $0.35\mu m$ technology
$W_{12}, W_{34}, W_5, W_6$	$2\mu m$ to $20\mu m$	$2\mu m$ to $20\mu m$
$W_7, W_8, W_{10}, W_m$	$2\mu m$ to $20\mu m$	$2\mu m$ to $20\mu m$
$L$	$0.2\mu m$ to $2\mu m$	$0.4\mu m$ to $2\mu m$
$CapH$	$6\mu m$ to $200\mu m$	$6\mu m$ to $200\mu m$

The various design parameters for the op-amp are width and length of various transistors and value of the Miller capacitor. The value of the supply voltage  $VDD$  is fixed to 1.2V and value of a bias current  $I_0$  is set to  $3\mu A$  for  $0.13\mu m$  technology. For the  $0.35\mu m$  technology, the supply voltage is set to 3.3V and bias current is set to  $4\mu A$ . In the layout, the value of the capacitor depends upon the area of the capacitor. The width of the Miller capacitor in the layout of the op-amp is decided automatically, while its height ( $CapH$ ) is a design parameter. Further, the length of all transistors is kept equal. The various design parameters are described in Table 5.12. The maximum circuit evaluations during the automatic design process are set to 5000.

**TABLE 5.13: Two stage op-amp (schematic-level design using ABC algorithm): Average of results from schematic-level simulations and corresponding post-layout simulation.**

Specification	0.13 $\mu m$ technology		0.35 $\mu m$ technology	
	Schematic-level simulation	Post-layout simulation	Schematic-level simulation	Post-layout simulation
Gain (dB)	81.66	81.66	78.7	78.7
PM ( $^\circ$ )	66.52	52.71	60.7	51.6
UGB (MHz)	169.33	147.79	110.9	96.3
PSRR (dB)	80.33	80.33	94.8	94.8
CMRR (dB)	92.69	92.69	82.6	82.6
PC ( $\mu W$ )	28.25	28.25	77.4	77.4
RSR ( $V/\mu S$ )	53.98	54.61	46.3	44.2
FSR ( $V/\mu S$ )	35.58	31.34	35.2	32.8
Error (%)	0.00	6.46	0.65	6.05

In the first experiment, the op-amp is designed in  $0.13 \mu m$  and  $0.35 \mu m$  technologies using the ABC algorithm at schematic-level where the layout parasitics are not possible to consider. We carried out ten independent design trials. The average of the obtained results is shown in Table 5.13. The obtained results reveal that the average RMS error is 0% i.e. the ABC algorithm had designed op-amp at schematic-level all ten times satisfying all the specifications in  $0.13 \mu m$  technology. However, when the layouts are prepared from these optimized schematics and post-layout simulation is carried out, the average RMS error is jumped to 6.46%. The obtained simulation results show that the frequency sensitive specifications such as phase-margin, unity gain bandwidth, and slew rate are affected worst. The deterioration in the performance at layout-level is due to the added parasitic-components at layout-level, which is not considered at schematic-level optimization. For the  $0.35 \mu m$  CMOS technology also, the ABC algorithm has designed op-amp at schematic-level with the average error of 0.65%, which is jumped to 6.05% when the post-layout simulation is carried out.

**TABLE 5.14: Two stage op-amp (parasitic-aware design): Average of simulation results.**

Specification	0.13 $\mu m$ CMOS technology				0.35 $\mu m$ CMOS technology			
	ABC	PSO	EABC	MPSO	ABC	PSO	EABC	MPSO
Gain (dB)	81.7	81.5	80.7	81.4	77.7	73.2	77.9	77.4
PM ( $^{\circ}$ )	62.7	62.6	63.7	62.2	60.3	59.4	60.1	60.3
UGB (MHz)	136.8	110.0	124.8	110.2	102.9	108.4	104.9	103.6
PSRR (dB)	79.0	78.0	77.2	78.3	92.7	88.8	93.7	91.4
CMRR (dB)	81.8	83.4	82.4	81.9	81.5	80.2	84.0	81.1
PC ( $\mu W$ )	28.6	29.0	29.1	27.8	80.1	78.0	78.1	79.8
RSR ( $V/\mu S$ )	37.8	47.2	40.1	37.7	45.1	49.4	45.0	44.4
FSR ( $V/\mu S$ )	37.0	37.0	35.9	38.7	34.8	35.1	35.1	34.7
RMS Error (%)	0.0	0.23	0.0	0.0	1.19	3.74	0.96	1.35
Time (Minutes)	107.8	79.4	123.1	84.6	410.8	334.1	386.1	387.3

In the second experiment, the op-amp is designed at layout-level using the concept of the parasitic-aware automatic circuit design with the help of the ABC, PSO, EABC and MPSO algorithm. Each algorithm has designed op-amp at layout-level in  $0.13 \mu m$  and  $0.35 \mu m$  technologies ten times independently and an average of the obtained results are considered for the comparison. The average of the obtained results is shown in Table 5.14. For the op-amp design in  $0.13 \mu m$  technology, except for the PSO algorithm, all other algorithms have designed op-amp with zero average RMS error. However, the average design time for the PSO algorithm is minimum. This indicates the lack of consistency in the performance of the PSO algorithm. In Fig. 5.16, the variations in the

## 5.5 Two-stage operational amplifier

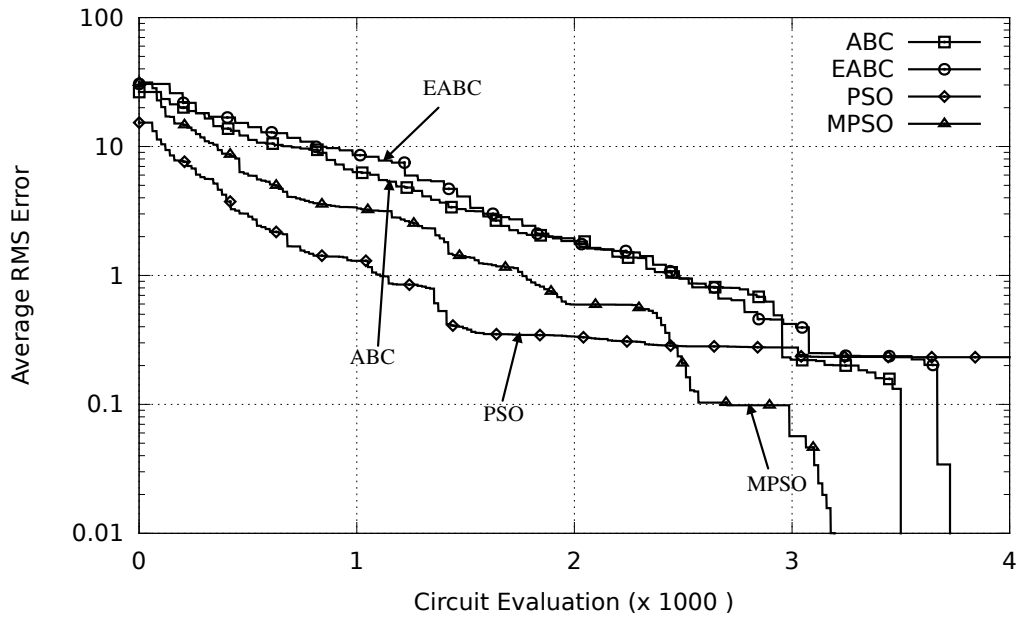


FIGURE 5.16: Two stage op-amp in  $0.13 \mu\text{m}$  technology: Variations in average RMS error with circuit evaluations.

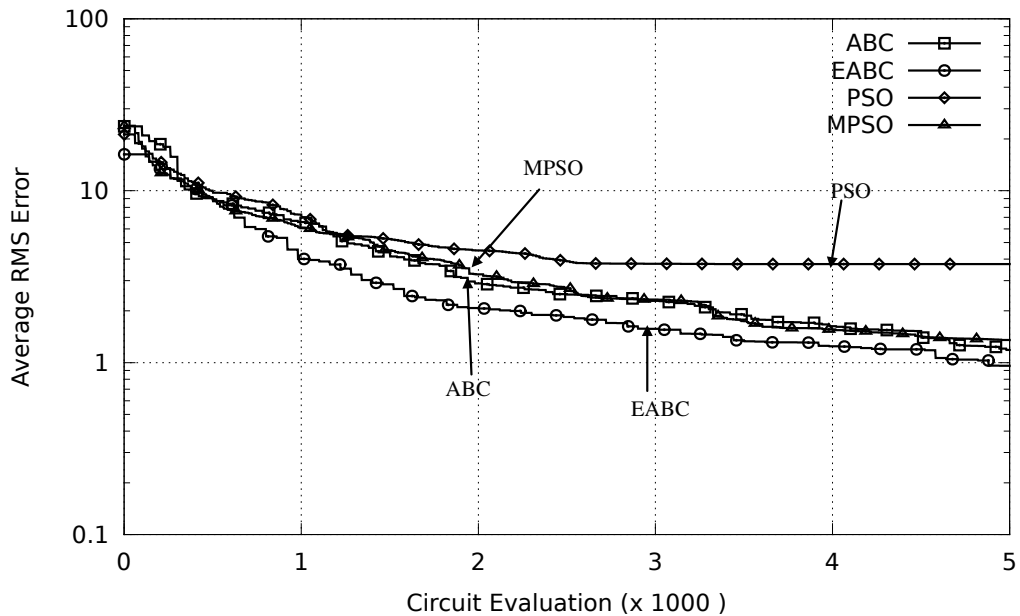
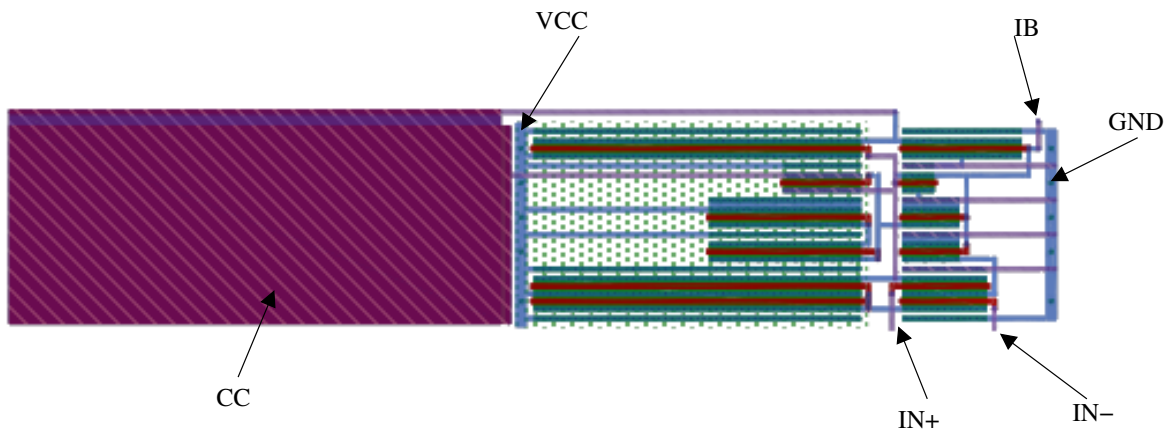


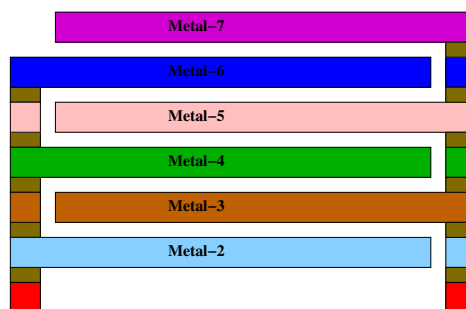
FIGURE 5.17: Two stage op-amp in  $0.35 \mu\text{m}$  technology: Variations in average RMS error with circuit evaluations.

**TABLE 5.15: Two stage op-amp (parasitic-aware design): Design parameters for best design run**

Design parameter	0.13 $\mu m$ CMOS technology				0.35 $\mu m$ CMOS technology			
	ABC	PSO	EABC	MPSO	ABC	PSO	EABC	MPSO
$W_{12}$ ( $\mu m$ )	7.4	10.4	10.5	5.2	5.0	2.0	2.2	2
$W_{34}$ ( $\mu m$ )	6.9	8.9	4.5	20.0	2.0	2.0	2.5	2.7
$W_5$ ( $\mu m$ )	10.6	8.9	10.2	3.5	2.8	3.3	5.3	6.7
$W_6$ ( $\mu m$ )	5.7	5.5	5.1	2.0	2.2	3.2	4.1	5.5
$W_7$ ( $\mu m$ )	18.3	20.0	15.8	7.3	4.7	6.8	7.8	11.8
$W_8$ ( $\mu m$ )	10.2	15.2	8.4	20.0	11.8	20.0	20.0	14.4
$W_{10}$ ( $\mu m$ )	7.3	9.2	7.8	9.3	7.1	3.3	8.0	3.8
$W_m$ ( $\mu m$ )	4.4	8.0	6.0	4.8	5.4	2.0	3.6	2.0
$L$ ( $\mu m$ )	0.4	0.4	0.4	0.4	1.0	1.1	1.1	1.3
$CapH$ ( $\mu m$ )	35	33	33	30	83	61	78	73

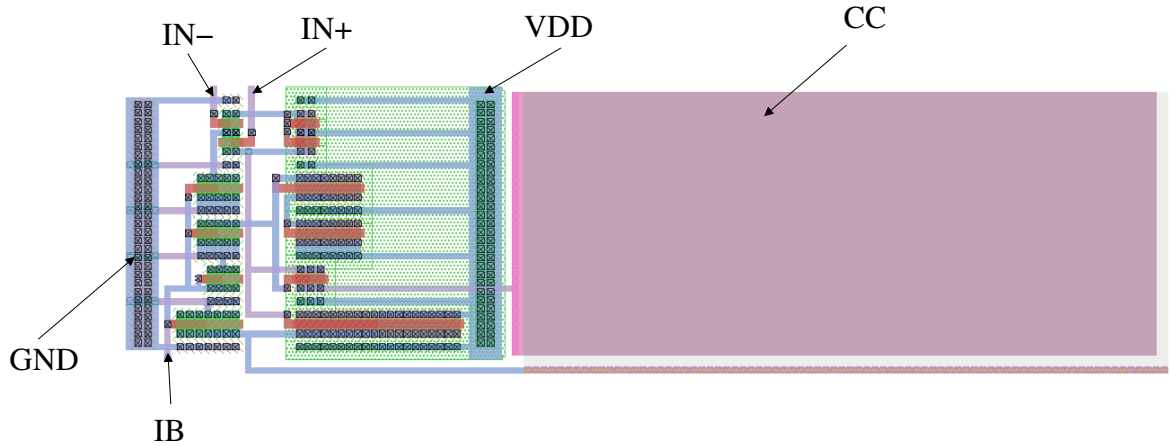


**FIGURE 5.18: Two-stage op-amp in 0.13  $\mu m$  technology: Layout of best design obtained from MPSO algorithm.**



**FIGURE 5.19: Two-stage op-amp: Layout scheme for stacked capacitor.**

## 5.5 Two-stage operational amplifier



**FIGURE 5.20: Two-stage op-amp in  $0.35\mu m$  technology: Layout of best design obtained from EABC algorithm.**

average RMS error with the circuit evaluations are shown for the  $0.13\mu m$  technology. It indicates that the MPSO algorithm has designed op-amp at layout-level most efficiently in  $0.13\mu m$  technology which is followed by the ABC, EABC and PSO algorithms.

For the op-amp design in  $0.35\mu m$  CMOS technology, the obtained result suggests that the EABC algorithm has most efficiently designed op-amp with 0.96% of average RMS error. The variations in the average RMS error with the circuit evaluation are shown in Fig. 5.17.

In Table 5.12, the design parameters obtained from best design trial by each algorithm are shown. In Fig. 5.18, the layout obtained from the best design trial using the MPSO algorithm is shown. In the layout of op-amp in  $0.13\mu m$  technology, to implement the Miller capacitor  $C_c$ , the concept of the stacked capacitor is used as shown in Fig 5.19. Such stacked capacitor increases capacitance per unit area and thus reduces the space requirement. In Fig. 5.20, the layout generated in best design trial of EABC algorithm in  $0.35\mu m$  technology is illustrated.

### **Considering process and temperature variations in design**

There is always some process variations in the manufacturing and because of this, all the devices do not have the same characteristics. In order to achieve the robust design, it is necessary to consider the process variations in the design process. It is also possible to include process and temperature variations with the proposed approach of parasitic-aware design. We also carried out the parasitic-aware design of two-stage op-amp in  $0.13\mu m$  technology using ABC algorithm considering process variations.

Further, the temperature variation between  $0^\circ$  and  $70^\circ$  is also considered. To consider process and temperature variations, five process corners TT, FF, FS, SF, and SS are simulated at three temperatures i.e.  $0^\circ$ ,  $25^\circ$  and  $70^\circ$  during the parasitic-aware design process. However, considering all these 15 process corners from the beginning of the circuit design results in a large design time. In order to save the design time, following strategy is adopted

- The parasitic-aware design of the op-amp started with considering only TT corner and no temperature variation.
- When the RMS error falls below 10%, FF corner is introduced in design process resulting simulation of layout at TT and FF process corners.
- When combined RMS error falls below 9% another process corner SS is included.
- Similarly, FS and SF corners are included step by step when RMS error falls below 8% and 7%, respectively.
- Once, RMS error for all process corners falls below 7%, the temperature of  $0^\circ$  is included. This leads to total 10 simulations to evaluate performance of the circuit i.e. five simulations with each process corner at  $25^\circ$  and five simulations with each process corner at  $0^\circ$ .
- When RMS error falls below 5%, the temperature of  $70^\circ$  is also considered in design process leading total 15 corner simulation to evaluate the performance of the circuit.

This method of introducing the process corner step by step reduces the design time. In Table 5.16, 5.17, and 5.18 illustrate the results of the post-layout simulation of op-amp designed considering process and temperature variations using ABC, MPSO, and EABC algorithms, respectively. The op-amp designed using ABC algorithm satisfies all design specifications at all design corners. The op-amp designed using MPSO algorithm fails to provide the desired CMRR at two process corners. In case of the EABC algorithm, the designed op-amp fails to provide desired results at five different process corners. The various design parameters of the designed op-amp using considered algorithms are illustrated in Table 5.19. In this design experiment, the op-amp is designed only one time using ABC, MPSO, and EABC algorithms. Thus, obtained results cannot be used to compare the performance of these algorithms.

**TABLE 5.16: Two stage op-amp design (parasitic-aware) using ABC algorithm considering process and temperature variations**

Process	Temperature ( $^{\circ}C$ )	Gain (dB) $\geq 80$	UGB (MHz) $\geq 100$	PM ( $^{\circ}$ ) $\geq 60$	PSRR (dB) $\geq 75$	CMRR (dB) $\geq 80$	PC ( $\mu W$ ) $\leq 30$	RSR ( $V/\mu S$ ) $\geq 35$	FSR ( $V/\mu S$ ) $\geq 35$
TT	0	81.8	137.8	70.3	80.2	82.4	29.6	40.1	36.4
	25	81.2	130.4	67.9	79.2	81.9	29.6	40.1	36.1
	70	80.2	116.8	64.8	77.5	81.2	29.6	40.0	35.8
FF	0	81.6	135.3	72.4	80.5	83.9	29.8	40.4	36.6
	25	81.1	129.2	70.5	79.5	83.6	29.8	40.3	36.4
	70	80.2	117.5	68.0	77.9	83.2	29.8	40.2	36.1
FS	0	81.8	142.6	66.9	78.8	82.1	29.6	40.3	36.6
	25	81.2	132.3	63.9	77.6	81.5	29.6	40.2	36.4
	70	80.2	117.1	60.3	75.5	80.6	29.6	40.1	36.1
SF	0	81.9	128.9	73.4	81.2	83.3	29.5	40.0	35.8
	25	81.3	123.4	71.6	80.3	82.8	29.5	39.9	35.5
	70	80.4	112.6	69.3	78.8	82.3	29.5	39.9	35.3
SS	0	82.1	137.5	68.1	79.9	82.1	29.4	39.9	36.0
	25	81.5	127.7	65.3	78.8	81.5	29.4	39.9	35.8
	70	80.5	113.2	61.9	77.0	80.6	29.4	39.8	35.5

**TABLE 5.17: Two stage op-amp design (parasitic-aware) using MPSO algorithm considering process and temperature variations**

Process	Temperature ( $^{\circ}C$ )	Gain (dB) $\geq 80$	UGB (MHz) $\geq 100$	PM ( $^{\circ}$ ) $\geq 60$	PSRR (dB) $\geq 75$	CMRR (dB) $\geq 80$	PC ( $\mu W$ ) $\leq 30$	RSR ( $V/\mu S$ ) $\geq 35$	FSR ( $V/\mu S$ ) $\geq 35$
TT	0	81.9	125.0	69.6	79.5	82.0	26.1	36.5	52.8
	25	81.3	117.7	67.7	78.4	81.3	26.0	36.4	53.1
	70	80.3	105.2	65.1	76.7	80.3	26.0	36.3	53.6
FF	0	81.7	122.3	71.0	79.7	82.5	26.2	36.7	47.8
	25	81.1	116.5	69.5	78.8	82.0	26.2	36.7	48.5
	70	80.2	105.6	67.3	77.1	81.1	26.1	36.6	49.7
FS	0	81.9	129.0	67.1	78.0	81.1	26.1	36.6	58.8
	25	81.3	120.1	64.6	76.7	80.3	26.1	36.5	58.6
	70	80.3	106.1	61.4	74.7	79.9	26.1	36.4	58.3
SF	0	82.0	116.5	71.8	80.5	82.5	26.0	36.4	43.6
	25	81.4	111.1	70.4	79.5	81.5	26.0	36.3	44.7
	70	80.5	100.7	68.4	78.0	80.7	26.0	36.3	46.2
SS	0	82.3	124.2	68.0	79.2	81.5	25.9	36.3	56.0
	25	81.7	115.6	65.8	78.0	80.7	25.9	36.2	56.1
	70	80.6	102.2	62.8	76.1	79.5	25.9	36.1	56.2

**TABLE 5.18: Two stage op-amp design (parasitic-aware) using EABC algorithm considering process and temperature variations**

Process	Temperature (°C)	Gain	UGB	PM	PSRR	CMRR	PC	RSR	FSR
		(dB) ≥ 80	(MHz) ≥ 100	(°) ≥ 60	(dB) ≥ 75	(dB) ≥ 80	( $\mu W$ ) ≤ 30	(V/ $\mu S$ ) ≥ 35	(V/ $\mu S$ ) ≥ 35
TT	0	81.0	131.6	72.4	78.9	81.1	28.4	39.9	36.2
	25	80.2	123.9	70.0	77.6	80.5	28.3	39.8	36.1
	70	78.7	113.5	62.6	72.4	78.2	28.4	39.8	36.0
FF	0	81.0	124.5	74.2	79.4	82.6	28.5	40.1	36.7
	25	80.3	119.5	72.4	78.1	82.1	28.5	40.1	36.5
	70	79.1	109.0	70.0	79.1	81.3	28.5	40.0	36.2
FS	0	80.9	139.2	69.1	76.9	80.4	28.4	40.0	36.3
	25	80.0	128.9	66.2	75.3	79.6	28.4	40.0	36.2
	70	78.7	113.5	62.6	72.4	78.2	28.4	39.8	35.9
SF	0	81.2	117.6	75.1	80.3	82.3	28.3	39.7	36.1
	25	80.5	113.1	73.4	79.2	81.7	28.3	39.6	35.9
	70	79.3	103.7	71.2	77.3	80.9	28.3	39.6	35.6
SS	0	81.2	133.2	70.3	78.5	80.7	28.2	39.6	35.8
	25	80.3	123.7	67.5	77.0	80.0	28.2	39.6	35.6
	70	79.0	109.1	64.0	74.5	78.8	28.2	39.6	35.4

**TABLE 5.19: The design parameters for two stage op-amp design (parasitic-aware) considering process and temperature variations**

Design parameter	ABC	EABC	MPSO
$W_{12}$ ( $\mu m$ )	6.4	7.5	2.1
$W_{34}$ ( $\mu m$ )	8.0	3.7	4.3
$W_5$ ( $\mu m$ )	10.3	10.2	2.0
$W_6$ ( $\mu m$ )	5.6	5.8	2.0
$W_7$ ( $\mu m$ )	19.3	19.0	8.3
$W_8$ ( $\mu m$ )	10.5	5.3	11.1
$W_{10}$ ( $\mu m$ )	7.7	5.7	6.6
$W_m$ ( $\mu m$ )	5.8	7.0	15.2
$L$ ( $\mu m$ )	0.4	0.4	0.4
$CapH$ ( $\mu m$ )	35	34	17

## 5.5 Two-stage operational amplifier

The performance of any evolutionary optimization algorithm varies with the nature of the optimization problem represented by the cost function. In the parasitic-aware circuit design and optimization process, the cost function depends upon the following factors.

- Circuit
- Circuit specifications
- Technology i.e. device characteristics and the model representing the behavior of the device
- Parasitics added by the layout

The change in any of these parameters leads to change in the cost function representing the optimization problem. Thus, whenever there is a change in the technology or layout topology (floor-plan and placement), there is a corresponding change in the cost function. This affects the performance of the optimization algorithm.

In this work, to compare the performances of the various optimization algorithms, we utilize average RMS error (the difference between the obtained and desired specifications) and the average design time. The average RMS error represents the rate of the convergence. The average design time represents the speed of the convergence.

In this work, we have designed VCO and Two-stage amplifier in two different CMOS technologies i.e.  $0.13\mu m$  and  $0.35\mu m$ . The circuit topology and the layout topology (floor-plan and placement) are identical for the both CMOS technologies. The comparison of the obtained results in both technologies are described in Table 5.10 for the VCO and Table 5.14 for the Two-stage op-amp.

In the case of VCO design, the average design time increases for design in  $0.13\mu m$  CMOS technology in comparison with  $0.35\mu m$  CMOS technology. However, the convergence rate for the considered algorithms (ABC, EABC, PSO, and MPSO) remains identical. In the case of the Two-stage op-amp design in  $0.13\mu m$  CMOS technology, except PSO algorithm, all other considered algorithms are able to design op-amp with the desired specifications. However, in the design of the  $0.35\mu m$  technology, no algorithm is able to design op-amp satisfying desired specifications. The performance of the EABC algorithm is found the best, which is followed by the ABC, MPSO and PSO algorithms.



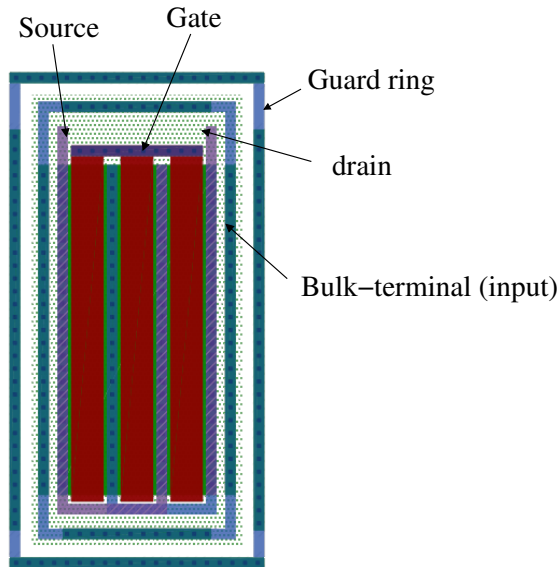
## 5.6 Bulk-driven OTA

- Rise and fall slew rate  $\geq 0.3V/mS$

Each algorithm has designed bulk-driven OTA at layout-level 10 times and for the comparison average of the obtained results is considered. The design parameters are width and length of the various transistor and value of the Miller capacitor. The design parameters with their search-space are shown in Table 5.25. The capacitor value is decided by its area. The width of the capacitor is calculated automatically in configurable layout and its height is a design parameter.

**TABLE 5.20: Bulk-driven OTA (parasitic-aware design): Search-space for design parameters.**

Design parameter	Search-space
$W_{12}, W_{34a}, W_{34b}, W_{5p}$	$5 \mu m$ to $50 \mu m$
$W_6, W_7, W_{89}$	$5 \mu m$ to $50 \mu m$
$L_C$	$0.5 \mu m$ to $5 \mu m$
$CapH$	$1 \mu m$ to $100 \mu m$



**FIGURE 5.22: Layout of PMOS with with input at bulk-terminal**

The width of the transistor is between  $5 \mu m$  to  $50 \mu m$ . This variation is considerably large. Thus, in order to maintain the aspect ratio of the layout of a transistor, finger transistor is utilized. Further, use of the finger transistor also reduces parasitics. The layout of the transistor with bulk input i.e. transistors  $M1$  and  $M2$  is shown in Fig. 5.22. The guard-ring is utilized to provide proper isolation and reduces the effect of noise.

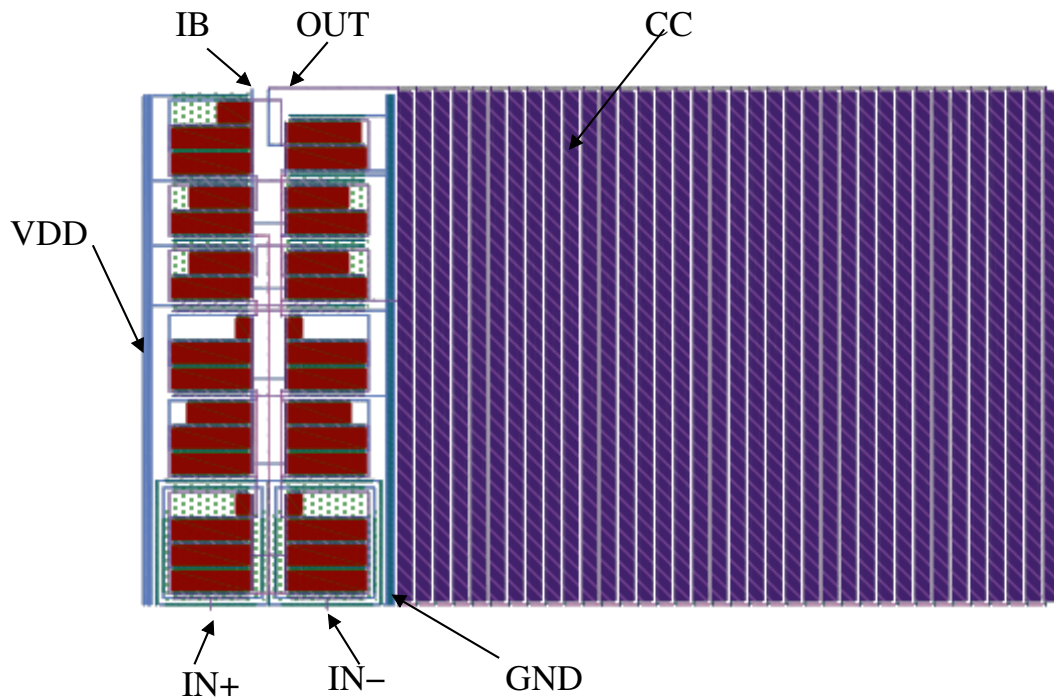
The bulk-driven OTA is designed 10 times with the help of each algorithm. The average of the obtained results is described in Table 5.21. Each algorithm is able to design bulk-

**TABLE 5.21: Bulk-driven OTA (parasitic-aware design): Average of simulation results.**

Specification	ABC	PSO	EABC	MPSO
Gain (dB)	41.5	41.1	40.8	41.0
PM ( $^{\circ}$ )	63.2	63.0	61.1	63.0
BW (KHz)	2.2	2.3	2.4	2.0
PC ( $\mu W$ )	19.8	19.5	19.2	18.9
RSR ( $V/mS$ )	0.69	0.72	0.69	0.63
FSR ( $V/mS$ )	0.99	1.01	1.14	0.97
RMS Error (%)	0.0	0.0	0.0	0.0
Time (Minutes)	8.4	5.77	8.78	3.14

**TABLE 5.22: Bulk-driven OTA(parasitic-aware): Design parameters for best design run**

Design parameter	ABC	PSO	EABC	MPSO
$W_{12}$ ( $\mu m$ )	49.3	13.1	40.3	48.5
$W_{34a}$ ( $\mu m$ )	22.7	18.3	44.2	25.9
$W_{34b}$ ( $\mu m$ )	16.8	47.8	39.0	27.5
$W_{5p}$ ( $\mu m$ )	7.6	20.4	47.7	19.7
$W_6$ ( $\mu m$ )	14.6	9.7	14.7	26.8
$W_7$ ( $\mu m$ )	25.1	50.0	29.5	47.8
$W_{89}$ ( $\mu m$ )	20.4	37.2	37.0	39.0
$L_C$ ( $\mu m$ )	3.3	3.5	4.0	3.1
$CapH$ ( $\mu m$ )	88	100	84	96



**FIGURE 5.23: Bulk-driven OTA (parasitic-aware design): Layout of best design obtained from MPSO algorithm.**

driven OTA at layout-level satisfying all the specifications. However, the average design time is a minimum for the MPSO algorithm which is only 3.14 minutes. In Table. 5.22, the design parameters obtained from the best design trial of each algorithm is listed. In Fig. 5.23, the layout of bulk-driven OTA obtained from the best design trial of MPSO algorithm is shown. The capacitor is implemented stacking the multiple metal layers.

### Considering process variations in design

Further, the bulk-driven OTA is also designed considering the process variations. This leads to the simulation of the circuit at five process corners i.e. TT, FF, FS, SF and, FF. However, in order to reduce design time following strategy is adopted in the design process.

- Initially, the design is started considering FF corner. The design of the circuit is experimentally found difficult at this design corner. Thus, the circuit is designed at this corner.
- When the RMS error falls below 0.5% with FF corner, SS design corner is included in the design process. This leads to the simulation of the circuit at two corners

**TABLE 5.23: Bulk-driven OTA (parasitic-aware design): The performance measure of the obtained solution at TT, FF, FS, SF and SS process corner obtained by ABC, EABC, PSO and MPSO algorithms in 0.13  $\mu m$  technology.**

Algorithm	Process corner	Gain (dB)	UGB (KHz)	PM ( $^{\circ}$ )	PC ( $\mu W$ )	RSR ( $V/mS$ )	FSR ( $V/mS$ )
ABC	TT	46.1	2.10	61.5	19.5	0.72	0.94
	FF	40.6	1.80	65.4	19.7	0.56	0.91
	FS	41.3	2.00	64.1	19.5	0.58	0.95
	SF	40.4	1.88	65.2	19.5	0.69	0.88
	SS	41.1	1.97	63.7	19.3	0.72	0.92
RMS Error: 0.0%							
PSO	TT	45.6	1.81	60.8	18.4	0.59	0.86
	FF	39.0	1.62	65.3	18.6	0.44	0.83
	FS	39.0	1.62	65.3	18.3	0.45	0.84
	SF	41.7	1.70	62.9	18.4	0.61	0.81
	SS	41.9	1.70	62.7	18.1	0.62	0.82
RMS Error: 0.38%							
EABC	TT	46.9	2.27	60.8	19.3	0.70	1.00
	FF	40.0	1.90	66.3	19.5	0.55	0.94
	FS	41.5	2.14	63.5	19.3	0.58	1.03
	SF	41.4	1.94	65.1	19.3	0.70	0.90
	SS	42.9	2.17	62.3	19.1	0.72	0.98
RMS Error: 0.0%							
MPSO	TT	46.7	1.84	64.2	19.7	0.64	0.85
	FF	40.0	1.59	68.4	19.9	0.53	0.80
	FS	41.0	1.74	66.5	19.7	0.56	1.00
	SF	40.5	1.59	67.8	19.6	0.58	0.80
	SS	41.2	1.72	65.9	19.5	0.62	0.98
RMS Error: 0.0%							

and an average of RMS errors of these two design corners is considered.

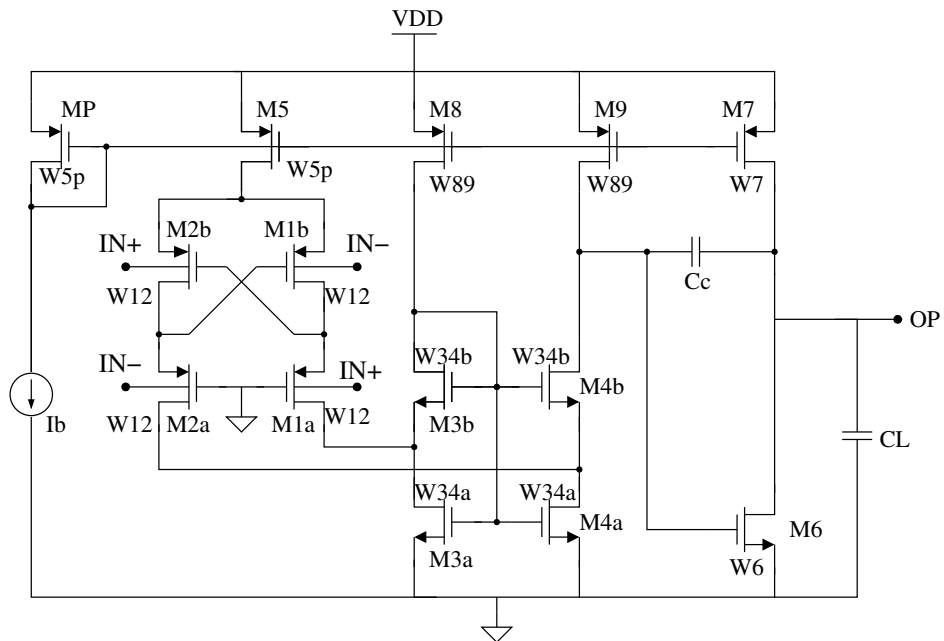
- The FF corner represents fast NMOS and fast PMOS. While SS corner represents slow NMOS and slow PMOS. This represents two extreme ends of the circuit performance.
- When the average RMS error of FF and SS corner falls below 0.5%, TT corner is included in the design. Thus, an average of RMS errors of FF, SS and, TT corners are considered.
- Similarly, step by step remaining two process corners i.e. SF and FS are introduced in the design process.

**TABLE 5.24: The design parameters for bulk-driven OTA (parasitic-aware) considering process variations**

Design parameter	ABC	PSO	EABC	MPSO
$W_{12}$ ( $\mu m$ )	35.2	14.7	32.3	46.6
$W_{34a}$ ( $\mu m$ )	25.4	25.2	16.4	22.1
$W_{34b}$ ( $\mu m$ )	50.0	50.0	38.0	50.0
$W_{5p}$ ( $\mu m$ )	20.4	23.4	26.2	17.6
$W_6$ ( $\mu m$ )	30.4	20.3	17.2	36.8
$W_7$ ( $\mu m$ )	40.0	24.1	31.1	31.7
$W_{89}$ ( $\mu m$ )	31.8	25.2	26.2	23.9
$L_C$ ( $\mu m$ )	4.6	3.6	2.7	5.0
$CapH$ ( $\mu m$ )	109	176	200	133

The obtained performance of the bulk-driven OTA designed (single design trial) considering process variations using the ABC, PSO, EABC and MPSO algorithms is illustrated in Table 5.23 and the corresponding values of the various design parameters are illustrated in Table 5.24. The obtained results reveal that except PSO algorithm, all other considered algorithms have successfully designed bulk-driven OTA at layout-level considering the process variations.

## 5.7 Enhanced bulk-driven OTA

**FIGURE 5.24: Circuit of enhanced bulk-driven OTA.**

In Fig. 5.24, the circuit of the enhanced bulk-driven OTA is illustrated. The bulk-driven technique suffers from the lower transconductance and it leads to a lower gain of the amplifier. In order to overcome this problem use of the cross-coupled pair of transistors is suggested in [46]. The cross-coupled pair of transistors imparts negative impedance degeneration which improves overall transconductance of the differential pair and enhances the overall gain of the OTA. The enhanced bulk-driven OTA is designed to satisfy the following specifications.

- Gain  $\geq 60dB$
- Bandwidth (BW)  $\geq 1KHz$
- Phase margin  $\geq 55^\circ$
- Power consumption  $\leq 20nW$
- Rise and fall slew rate  $\geq 0.8V/mS$

These specifications are based on the design of enhanced bulk-driven OTA presented in [46]. The design parameters are the size of various transistors and value of the Miller capacitor. The design parameters with their search-space are described in the Table 5.27. The supply voltage is set to 0.25V and the value of a bias current value is 10nA.

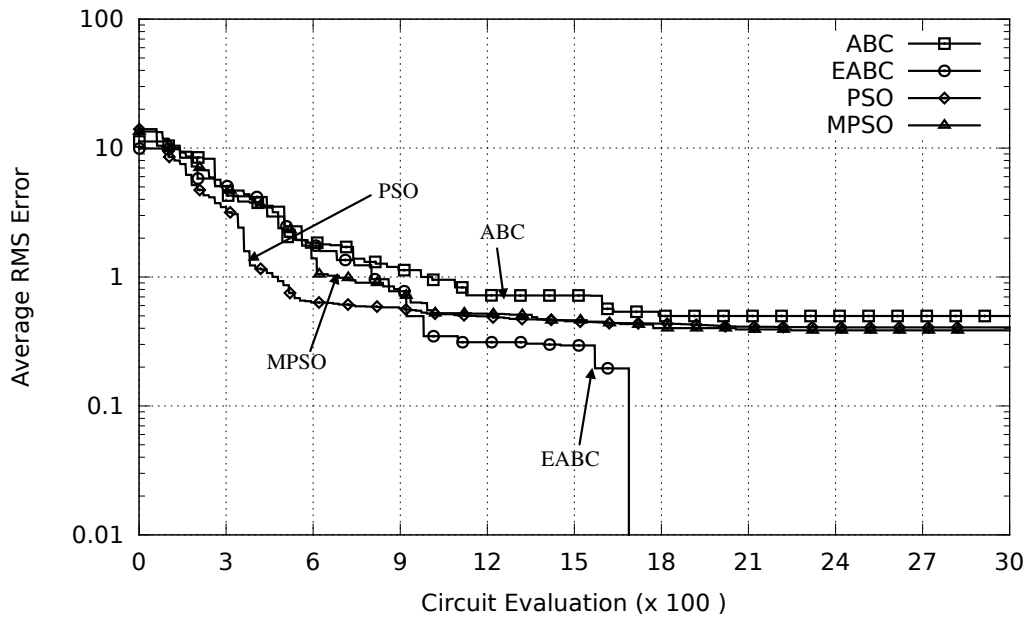
**TABLE 5.25: Enhanced bulk-driven OTA (parasitic-aware design): Search-space for design parameters.**

Design parameter	Search-space
$W_{12}, W_{34a}, W_{34b}, W_{5p}$	$5 \mu m$ to $100 \mu m$
$W_6, W_7, W_{89}$	$5 \mu m$ to $100 \mu m$
$L_C$	$0.5 \mu m$ to $5 \mu m$
$CapH$	$1 \mu m$ to $200 \mu m$

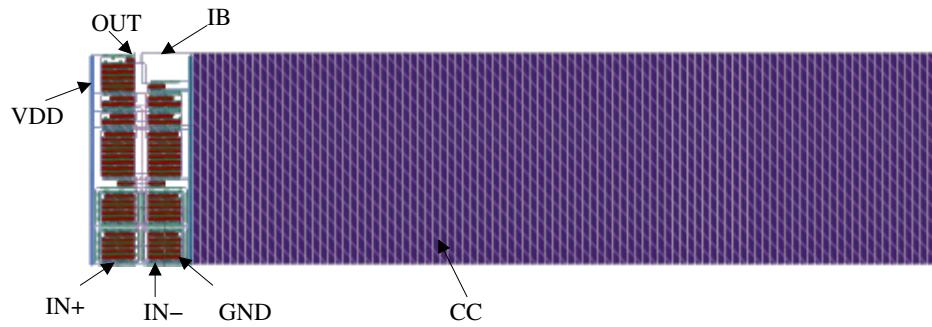
The OTA is designed in  $0.13 \mu m$  technology with ABC, PSO, EABC and MPSO algorithms. Each algorithm has designed enhanced bulk-driven OTA 10 times and the average of the obtained specifications are considered for the comparison. The obtained results are shown in Table 5.26. The EABC algorithm is effective in designing enhanced bulk-driven OTA at layout-level. It had designed OTA at layout level with zero average error satisfying all the specifications in all the design runs. The average design time of OTA with the EABC algorithm is only 64.2 minutes. The average design error for the ABC algorithm is 0.49%, for the PSO algorithm is 0.4% and, for the MPSO

**TABLE 5.26: Enhanced bulk-driven OTA (parasitic-aware design): Average of simulation results.**

Specification	ABC	PSO	EABC	MPSO
Gain (dB)	64.0	62.6	65.1	61.4
PM ( $^{\circ}$ )	61.2	58.6	58.8	61.3
BW (KHz)	4.6	5.0	4.0	5.0
PC ( $\mu W$ )	18.5	19.1	18.2	18.9
RSR ( $V/mS$ )	1.0	1.1	0.9	1.0
FSR ( $V/mS$ )	1.3	1.3	1.0	1.7
RMS Error (%)	0.49	0.4	0.0	0.38
Time (Minutes)	87.4	80.8	64.2	93.7

**FIGURE 5.25: Enhanced bulk-driven OTA (parasitic-aware design): Variations in average RMS error with circuit evaluations.****TABLE 5.27: Enhanced bulk-driven OTA: Design parameters for best design run**

Design parameter	ABC	PSO	EABC	MPSO
$W_{12}$ ( $\mu m$ )	99.2	69.9	49.1	48.8
$W_{34a}$ ( $\mu m$ )	16.0	14.3	5.0	5.0
$W_{34b}$ ( $\mu m$ )	97.7	81.9	79.2	45.3
$W_{5p}$ ( $\mu m$ )	66.5	46.6	71.1	65.7
$W_6$ ( $\mu m$ )	46.8	24.2	5.0	100.0
$W_7$ ( $\mu m$ )	78.8	95.2	52.3	79.9
$W_{89}$ ( $\mu m$ )	21.4	25.3	17.4	20.1
$L_C$ ( $\mu m$ )	3.9	2.6	1.2	1.5
$CapH$ ( $\mu m$ )	93	177	200	176



**FIGURE 5.26: Enhanced bulk-driven OTA (parasitic-aware design): Layout of best design obtained from EABC algorithm.**

algorithm is 0.38%. In Fig 5.25, the variations in the average RMS error with the circuit evaluation is illustrated. In Table 5.27, circuit parameters of best design run for each algorithm are illustrated. The layout of OTA of best design run for the EABC algorithm is shown in Fig 5.26.

## 5.8 Observations

From the conducted experiments on designing various analog circuits at layout level, following observations can be made

- With the help of the configurable layout, the layout of the analog circuits can be modified instantly by supplying the appropriate parameters. This flexibility of the configurable layout is utilized to carry-out parasitic aware automatic circuit design at layout-level.
- In the proposed concept of the parasitic-aware design of circuit, the net-list extracted from the layout is used. This extracted net-list is generated with the help of the extraction tool provided by the circuit design framework (i.e. MAGIC VLSI tool). Thus, it does not require additional models or estimation techniques for the consideration of the parasitics.
- The parasitic-aware layout-level automatic designs of the ring oscillator, CMOS buffer chain, VCO, op-amp, bulk-driven OTA and enhanced bulk-driven OTA are carried out successfully. Further, the proposed design methodology does not require any human intervention.
- The two-stage op-amp is designed in two CMOS technologies i.e.  $0.13 \mu m$  and

## 5.8 Observations

$0.35\mu m$ . The scripts for the configurable layout for these two technologies are almost same except the minor modifications in the layout of the capacitor. This is because the  $0.35\mu m$  technology has only four metal layers while in  $0.13\mu m$  technology has eight metal layers. Due to the limited number of metal layers available, the stacked capacitor is not used in the  $0.35\mu m$  technology.

- The concept of the parasitic-aware design is flexible enough to accommodate the process and temperature variations or other user-defined constraints in the design process. The two-stage op-amp is designed successfully at the layout-level considering process and temperature variations. Similarly, bulk-driven OTA is designed considering process variations successfully.
- The performance of the optimization algorithm depends upon the nature of the optimization problem represented by the cost function. The technology scaling results in the modification of the cost function. Thus, it affects the performance of the optimization algorithm.
- With the change in the layout topology, the number of the layout parasitic components and their value get affected. This modifies the nature of the optimization problem, resulting in performance variations of the optimization algorithms.

## CHAPTER 6

# Conclusion

Two efficient evolutionary algorithms, EABC and MPSO are proposed in this work. The EABC algorithm addresses the issue of the slow convergence speed of the ABC algorithm, while MPSO algorithm focuses on the diversity-loss issue of the PSO algorithm. The performances of these algorithms are compared with ABC, PSO, MABC and GABC algorithms by designing two-stage op-amp, high gain bulk-driven OTA and second generation current conveyor at schematic level using  $0.13\ \mu\text{m}$  and  $0.09\ \mu\text{m}$  CMOS technology. The EABC and MPSO algorithms have designed these circuits more efficiently than other considered algorithms. In case of the schematic-level design of the two-stage op-amp in  $0.13\ \mu\text{m}$  technology, EABC algorithm is the most effective. It has designed op-amp with 0.39% average RMS error. In case of the op-amp design in  $0.09\ \mu\text{m}$  technology, the MPSO algorithm is the most successful. It has designed op-amp with only 0.002% average error. For the schematic-level design of the high-gain low-voltage OTA in  $0.13\ \mu\text{m}$  and  $0.09\ \mu\text{m}$  technologies, the MPSO algorithm is the most successful. The MPSO algorithm designed OTA in both technologies satisfying all design specifications. Other algorithms (PSO, ABC, MABC, GABC) failed to design OTA. In case of CCII, the MPSO and EABC algorithms generated more promising results than other considered algorithms.

We have proposed the concept of the parasitic-aware automatic circuit design for the analog and mixed signal CMOS circuits at layout-level. The proposed concept is implemented using MAGIC VLSI tool by developing configurable layouts and including evolutionary optimization algorithms. The concept of the parasitic-aware design enables consideration of the layout-parasitics from the beginning of the design process.

The configurable layouts provide the possibility of exporting existing solutions in the other technologies. Further, the proposed concept of the parasitic-aware design provides layout-level design automation that does not require human intervention. With the help of the parasitic-aware design concept, six CMOS circuits such as ring oscillator, CMOS buffer chain, VCO, two-stage op-amp, bulk-driven OTA and, enhanced bulk-driven OTA are designed at layout-level using four different evolutionary algorithms namely; ABC, EABC, PSO and, MPSO algorithms. The VCO and two-stage op-amp are designed in  $0.13\ \mu\text{m}$  and  $0.35\ \mu\text{m}$  technologies, while the other circuits are designed in  $0.13\ \mu\text{m}$  technology.

The ring-oscillator is designed at layout-level with 0.001% average RMS error using MPSO algorithm. The average design time is found only 15.6 minutes. In order to investigate the effect of the change in the layout topology (floorplan and placement), we have also designed the Ring-oscillator with different layout topologies. In this design experiment, the Ring-oscillator is designed successfully. However, the average design time varies significantly to 29.5 minutes. This fact reveals that the layout topology affects the performance of the optimization algorithms. The CMOS buffer chain is designed at layout-level satisfying all design constraints in 29.6 minutes using EABC algorithm. The VCO is designed successfully, in both  $0.13\ \mu\text{m}$  and  $0.35\ \mu\text{m}$  technologies. This design example demonstrates the ability of the configurable layouts to export existing solutions from one technology to another technology. The design of the two-stage op-amp is carried out in both  $0.13\ \mu\text{m}$  and  $0.35\ \mu\text{m}$  technologies. When the op-amp is designed in  $0.13\ \mu\text{m}$  at schematic-level, the RMS error is zero, satisfying all design constraints. However, the post-layout simulation of the layouts generated from the optimized schematics resulted in average RMS error of 6.46%. On the other hand, the layout-level design of op-amp using the concept of parasitic-aware design resulted in zero average RMS error and satisfies all design requirements. Further, the op-amp is also designed successfully in  $0.13\ \mu\text{m}$  technology considering the process and temperature variations. It shows the flexibility of the parasitic-aware design concept to accommodate customized design constraints. Further, the obtained results of the VCO and Op-amp design in two different technologies, indicate that the technology affects the performance of the optimization algorithms. Similarly, the bulk-driven OTA and enhanced bulk-driven OTA are also designed at layout-level using the concept of the parasitic-aware design successfully. The obtained results reveal effectiveness of the proposed design methodology to design robust analog CMOS circuits in efficient manner.

# Bibliography

- [1] P. Mandal and V. Visvanathan, “Cmos op-amp sizing using a geometric programming formulation,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 20, no. 1, pp. 22–38, 2001.
- [2] P. K. Rout, “Fast and robust design of cmos vco for optimal performance,” Ph.D. dissertation, National Institute of Technology, Rourkela, India, National Institute of Technology, Rourkela, India, Aug. 2015.
- [3] M. R. Berkelaar and J. A. Jess, “Gate sizing in mos digital circuits with linear programming,” in *Proceedings of the conference on European design automation*. IEEE Computer Society Press, 1990, pp. 217–221.
- [4] T. Mak, K.-P. Lam, H. Ng, G. Rachmuth, and C.-S. Poon, “A cmos current-mode dynamic programming circuit,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 57, no. 12, pp. 3112–3123, 2010.
- [5] N. Menezes, R. Baldick, and L. T. Pileggi, “A sequential quadratic programming approach to concurrent gate and wire sizing,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 16, no. 8, pp. 867–881, 1997.
- [6] Y. Del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, “Particle swarm optimization: basic concepts, variants and applications in power systems,” *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 2, pp. 171–195, 2008.
- [7] S. P. Boyd, T. H. Lee *et al.*, “Optimal design of a cmos op-amp via geometric

## Bibliography

- programming,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 20, no. 1, pp. 1–21, 2001.
- [8] A. Purushothaman and C. D. Parikh, “A new delay model and geometric programming-based design automation for latched comparators,” *Circuits, Systems, and Signal Processing*, vol. 34, no. 9, pp. 2749–2764, 2015.
- [9] R. A. Thakker, M. S. Baghini, and M. B. Patil, “Automatic design of low-power low-voltage analog circuits using particle swarm optimization with re-initialization,” *Journal of Low Power Electronics*, vol. 5, no. 3, pp. 291–302, 2009.
- [10] C.-R. Hwang, “Simulated annealing: theory and applications,” *Acta Applicandae Mathematicae*, vol. 12, no. 1, pp. 108–111, 1988.
- [11] F. Glover and G. Kochenberger, *Handbook of Metaheuristics*, ser. International Series in Operations Research & Management Science. Springer US, 2006.
- [12] S. Sivanandam and S. Deepa, *Introduction to genetic algorithms*. Springer Science & Business Media, 2007.
- [13] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.
- [14] M. Dorigo and C. Blum, “Ant colony optimization theory: A survey,” *Theoretical computer science*, vol. 344, no. 2, pp. 243–278, 2005.
- [15] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm,” *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [16] M. Barros, J. Guilherme, and N. Horta, “Analog circuits optimization based on evolutionary computation techniques,” *INTEGRATION, the VLSI journal*, vol. 43, no. 1, pp. 136–155, 2010.
- [17] S. L. Sabat, K. S. Kumar, and S. K. Udgate, “Differential evolution and swarm intelligence techniques for analog circuit synthesis,” in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*. IEEE, 2009, pp. 469–474.

## Bibliography

- [18] G. Nicosia, S. Rinaudo, and E. Sciacca, “An evolutionary algorithm-based approach to robust analog circuit design using constrained multi-objective optimization,” *Knowledge-Based Systems*, vol. 21, no. 3, pp. 175–183, 2008.
- [19] P. Pereira, M. H. Fino, and M. Ventim-Neves, “Optimal lc-vco design through evolutionary algorithms,” *Analog Integrated Circuits and Signal Processing*, vol. 78, no. 1, pp. 99–109, 2014.
- [20] A. Somani, P. P. Chakrabarti, and A. Patra, “An evolutionary algorithm-based approach to automated design of analog and rf circuits using adaptive normalized cost functions,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 336–353, 2007.
- [21] B. P. De, R. Kar, D. Mandal, and S. Ghoshal, “An efficient design of cmos comparator and folded cascode op-amp circuits using particle swarm optimization with an aging leader and challengers algorithm,” *International Journal of Machine Learning and Cybernetics*, pp. 1–20.
- [22] S. Mallick, R. Kar, D. Mandal, and S. Ghoshal, “Optimal sizing of cmos analog circuits using gravitational search algorithm with particle swarm optimization,” *International Journal of Machine Learning and Cybernetics*, pp. 1–23, 2015.
- [23] B. P. De, R. Kar, D. Mandal, and S. Ghoshal, “Optimal design of high speed symmetric switching cmos inverter using hybrid harmony search with differential evolution,” *Soft Computing*, pp. 1–19, 2015.
- [24] M. Fakhfakh, Y. Cooren, A. Sallem, M. Loulou, and P. Siarry, “Analog circuit design optimization through the particle swarm optimization technique,” *Analog Integrated Circuits and Signal Processing*, vol. 63, no. 1, pp. 71–82, 2010.
- [25] A. Sallem, B. Benhala, M. Kotti, M. Fakhfakh, A. Ahaitouf, and M. Loulou, “Application of swarm intelligence techniques to the design of analog circuits: evaluation and comparison,” *Analog Integrated Circuits and Signal Processing*, vol. 75, no. 3, pp. 499–516, 2013.
- [26] T. R. Dastidar, P. Chakrabarti, and P. Ray, “A synthesis system for analog circuits based on evolutionary search and topological reuse,” *IEEE Transactions on evolutionary computation*, vol. 9, no. 2, pp. 211–224, 2005.

## Bibliography

- [27] M. Clerc, *Particle Swarm Optimization*, ser. ISTE. Wiley, 2010.
- [28] D. Karaboga and B. Basturk, “On the performance of artificial bee colony (abc) algorithm,” *Applied soft computing*, vol. 8, no. 1, pp. 687–697, 2008.
- [29] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, “A comprehensive survey: artificial bee colony (abc) algorithm and applications,” *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21–57, 2014.
- [30] M. Črepinšek, S.-H. Liu, and M. Mernik, “Exploration and exploitation in evolutionary algorithms: A survey,” *ACM Comput. Surv.*, vol. 45, no. 3, pp. 35:1–35:33, Jul. 2013.
- [31] N. Bacanin and M. Tuba, “Artificial bee colony (abc) algorithm for constrained optimization improved with genetic operators,” *Studies in Informatics and Control*, vol. 21, no. 2, pp. 137–146, 2012.
- [32] W.-f. Gao and S.-y. Liu, “A modified artificial bee colony algorithm,” *Computers & Operations Research*, vol. 39, no. 3, pp. 687–697, 2012.
- [33] G. Zhu and S. Kwong, “Gbest-guided artificial bee colony algorithm for numerical function optimization,” *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166–3173, 2010.
- [34] X. Zhang, X. Zhang, S. Ho, and W. Fu, “A modification of artificial bee colony algorithm applied to loudspeaker design problem,” *Magnetics, IEEE Transactions on*, vol. 50, no. 2, pp. 737–740, 2014.
- [35] J. Yang, W.-T. Li, X.-W. Shi, L. Xin, and J.-F. Yu, “A hybrid abc-de algorithm and its application for time-modulated arrays pattern synthesis,” *Antennas and Propagation, IEEE Transactions on*, vol. 61, no. 11, pp. 5485–5495, 2013.
- [36] F. Van den Bergh and A. P. Engelbrecht, “A study of particle swarm optimization particle trajectories,” *Information sciences*, vol. 176, no. 8, pp. 937–971, 2006.
- [37] K. Parsopoulos, *Particle Swarm Optimization and Intelligence: Advances and Applications: Advances and Applications*, ser. Advances in Computational Intelligence and Robotics:. Information Science Reference, 2010.

## Bibliography

- [38] D. Karaboga and B. Akay, “A comparative study of artificial bee colony algorithm,” *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.
- [39] M. Matsumoto and T. Nishimura, “Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator,” *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, 1998.
- [40] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélicier, and P. Zimmermann, “Mpfpr: A multiple-precision binary floating-point library with correct rounding,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 33, no. 2, p. 13, 2007.
- [41] “Ngspice user manual,” <http://ngspice.sourceforge.net/docs/ngspice-manual.pdf>, accessed: 2017-08-28.
- [42] “Predictive Technology Models (PTM) for cmos,” <http://ptm.asu.edu/>, accessed: 2016-04-17.
- [43] S. Patel and R. A. Thakker, “Automatic circuit design and optimization using modified pso algorithm,” *Journal of Engineering Science and Technology Review*, vol. 9, no. 1, pp. 89–94, 2016.
- [44] Y. Haga, H. Zare-Hoseini, L. Berkovi, and I. Kale, “Design of a 0.8 volt fully differential cmos ota using the bulk-driven technique,” in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*. IEEE, 2005, pp. 220–223.
- [45] R. He and L. Zhang, “Evaluation of modern mosfet models for bulk-driven applications,” in *Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on*. IEEE, 2008, pp. 105–108.
- [46] L. H. Ferreira and S. R. Sonkusale, “A 60-db gain ota operating at 0.25-v power supply in 130-nm digital cmos process,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 61, no. 6, pp. 1609–1617, 2014.
- [47] A. Yodtean, “A cmos ota and implementation,” in *Intelligent Signal Processing and Communication Systems (ISPACS), 2014 International Symposium on*. IEEE, 2014, pp. 094–098.
- [48] A. Sedra, G. W. Roberts, and F. Gohh, “The current conveyor: history, progress and new results,” in *IEE proceedings*, vol. 137, no. 2 Pt G, 1990, pp. 78–87.

## Bibliography

- [49] S. B. Salem, M. Fakhfakh, D. S. Masmoudi, M. Loulou, P. Loumeau, and N. Masmoudi, "A high performances cmos ccii and high frequency applications," *Analog Integrated Circuits and Signal Processing*, vol. 49, no. 1, pp. 71–78, 2006.
- [50] G. Ferri and N. C. Guerrini, *Low-voltage low-power CMOS current conveyors*. Springer Science & Business Media, 2003.
- [51] "Magic VLSI Layout Tool," <http://opencircuitdesign.com/magic/>, accessed: 2017-08-28.
- [52] "Python programming language," <https://www.python.org/>, accessed: 2017-08-28.
- [53] J. Lee and B. Kim, "A low-noise fast-lock phase-locked loop with adaptive bandwidth control," *IEEE Journal of solid-state circuits*, vol. 35, no. 8, pp. 1137–1145, 2000.

# APPENDIX A

## List of Publications

### Journal Papers

1. S. Patel and R. A. Thakker, “Automatic circuit design and optimization using modified pso algorithm,” *Journal of Engineering Science and Technology Review*, vol. 9, no. 1, pp. 89–94, 2016.
2. S. Patel and R. A. Thakker, “Parameter Space Exploration for Analog Circuit Design Using Enhanced Bee Colony Algorithm,” *Journal of Circuits, Systems, and Computers, World Scientific Publishing Company* [Accepted]

### Conference Papers

1. S. Patel and R. A. Thakker, “Parasitic Aware Automatic Analog CMOS Circuit Design Environment using ABC Algorithm,” *2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID)*, Pune, 2018.
2. S. Patel and R. A. Thakker, “Parasitic-Aware Automatic Analog CMOS Circuit Design Environment,” *2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)*, Dehli, 2019.[Accepted]