RUI MANUEL LEITÃO SANTOS TAVARES

TIME-DOMAIN OPTIMIZATION OF AMPLIFIERS BASED ON DISTRIBUTED GENETIC ALGORITHMS

Thesis presented in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the subject of Electrical and Computer Engineering.

LISBOA
2010

In memory of my father and my mother.

ACKNOWLEDGEMENTS

Several persons have contributed to the success of this work. To all of them I wish to express my deepest gratitude and thanks. In particular I would like to thank: Prof. Nuno Paulino, my supervisor, to whom I wish to express my sincere gratitude. His continuous guidance, encouragement, support, and valuable discussions have granting me the opportunity to pursue this work. Prof. João Goes, my co-supervisor for always supporting the project and for sharing his skills, passion and experience in analog circuit design.

Also, I am grateful to Professor A. Steiger-Garção, president of the Departamento de Engenharia Electrotecnica da Faculdade de Ciências de Technologia da Universidade Nova de Lisboa, for his friendship, understanding, and helped me to pursue my academia goals.

To the Electrical and Engineering Department (DEE) staff and colleagues, big thanks for the friendship, support and brief sessions of chat to clear a bit of the stress.

To my colleagues at the Microelectronics and Circuits (MESP) of the Centre for Technology and Systems (CTS), at UNINOVA, for support and friendship, notably, Michael Figueiredo, Edinei Santin and Bruno Esperança that used and tested the platform in their research works, e.g. Ph. D., and, after, allowed me to use the respective results in this document.

I also thank the Portuguese Foundation for the Science and Technology (FCT) for the grant to help realize the presented work. FCT also supported, financially, several research projects that originated the circuits and prototype.

ABSTRACT

The work presented in this thesis addresses the task of circuit optimization, helping the designer facing the high performance and high efficiency circuits demands of the market and technology evolution. A novel framework is introduced, based on time-domain analysis, genetic algorithm optimization, and distributed processing.

The time-domain optimization methodology is based on the step response of the amplifier. The main advantage of this new time-domain methodology is that, when a given settling-error is reached within the desired settling-time, it is automatically guaranteed that the amplifier has enough open-loop gain, A_{OL}, output-swing (OS), slew-rate (SR), closed loop bandwidth and closed loop stability. Thus, this simplification of the circuit's evaluation helps the optimization process to converge faster. The method used to calculate the step response expression of the circuit is based on the inverse Laplace transform applied to the transfer function, symbolically, multiplied by $1/s$ (which represents the unity input step). Furthermore, may be applied to transfer functions of circuits with unlimited number of zeros/poles, without approximation in order to keep accuracy. Thus, complex circuit, with several design/optimization degrees of freedom can also be considered. The expression of the step response, from the proposed methodology, is based on the DC bias operating point of the devices of the circuit. For this, complex and accurate device models (e.g. BSIM3v3) are integrated. During the optimization process, the time-domain evaluation of the amplifier is used by the genetic algorithm, in the classification of the genetic individuals. The time-domain evaluator is integrated into the developed optimization platform, as independent library, coded using C programming language.

The genetic algorithms have demonstrated to be a good approach for optimization since they are flexible and independent from the optimization-objective. Different levels of abstraction can be optimized either system level or circuit level. Optimization of any new block is basically carried-out by simply providing additional configuration files, e.g. chromosome format, in text format; and the circuit library where the fitness value of each individual of the genetic algorithm is computed.

Distributed processing is also employed to address the increasing processing time demanded by the complex circuit analysis, and the accurate models of the circuit devices. The communication by remote processing nodes is based on Message Passing interface

(MPI). It is demonstrated that the distributed processing reduced the optimization run-time by more than one order of magnitude.

Platform assessment is carried by several examples of two-stage amplifiers, which have been optimized and successfully used, embedded, in larger systems, such as data converters. A dedicated example of an inverter-based self-biased two-stage amplifier has been designed, laid-out and fabricated as a stand-alone circuit and experimentally evaluated. The measured results are a direct demonstration of the effectiveness of the proposed time-domain optimization methodology.

SUMÁRIO

O trabalho apresentado nesta dissertação aborda a tarefa do dimensionamento de circuitos (em concreto, amplificadores), e pretende ajudar o engenheiro no projecto de circuitos, automatizando parte desta mesma tarefa. A nova metodologia de optimização é baseada na resposta temporal do amplificador ao escalão e utiliza algoritmos genéticos com processamento distribuído.

A principal vantagem da análise da resposta ao escalão, é o facto de um dado tempo de estabelecimento, da resposta, dentro de um dado erro de estabelecimento é suficiente para garantir que o circuito amplificador tem suficiente ganho em malha aberta, "*output-swing*" (OS), "*slew-rate*" (SR), e através da resposta ao escalão, concluir sobre a estabilidade quando realimentado em malha fechada. Esta simplificação na avaliação dos circuitos ajuda o processo de optimização a convergir mais rapidamente. O procedimento para determinação da expressão da resposta ao escalão utiliza a transformada inversa de Laplace, aplicada à função de transferência, do circuito, multiplicada, simbolicamente, por $1/s$ (que representa o escalão à entrada do circuito). Mais, este procedimento pode ser aplicado a funções de transferência com um número ilimitado de zeros e pólos, sem necessidade de utilizar qualquer tipo de aproximação, evitando perda de precisão. Desta forma, é possível optimizar circuitos complexos, com vários graus de liberdade. O cálculo da resposta ao escalão, utilizando a expressão, descrita anteriormente, é baseado nos valores do ponto de funcionamento em repouso (PFR) do circuito. Neste contexto são utilizados modelos de transístores complexos e precisos (e.g. BSIM3v3) para calcularo PFR. Este método de avaliação de circuitos, baseado no domínio do tempo é utilizado, durante o processo de optimização, pelo algoritmo genético, para classificar, ordenar e, posteriormente, gerar novas populações de indivíduos (circuitos). O bloco de software responsável pela avaliação dos circuitos, no domínio do tempo, é uma biblioteca independente, codificada utilizando linguagem de programação C. Esta biblioteca é integrada na plataforma de optimização desenvolvida.

Os algoritmos genéticos demonstram ser uma boa abordagem para optimização: suficientemente flexíveis e independentes do tipo de optimização. Diferentes níveis de abstracção podem ser abordados: ao nível do sistema, ou ao nível do circuito. A instanciação de uma nova optimização apenas requer alguns ficheiros de configuração, e.g. descrição do cromossoma, em formato de texto; e biblioteca, em C, contendo função de avaliação dos indivíduos (*fitness*).

O emprego do processamento distribuído/paralelo diminui o tempo de processamento, o estudo de circuitos mais complexos e a utilização de modelos de transístores mais precisos. A comunicação entre os nós de processamento remoto e o servidor baseia-se no conceito de *Message Passing Interface* (MPI). É demonstrado que a utilização de processamento distribuído reduz o tempo de optimização, em mais do que uma ordem de grandeza.

O desempenho da plataforma foi verificado com vários exemplos de amplificadores de dois andares que foram optimizados e posteriormente utilizados, com sucesso, embutidos em sistemas integrados mais completos, como por exemplo conversores analógico/digital. O exemplo do amplificador de dois andares, inversor, auto-polarizado, foi desenhado, integrado e fabricado, e avaliado experimentalmente. Os resultados experimentais medidos são a demonstração, directa, da eficácia da metodologia de optimização baseada no domínio do tempo proposta.

KEYWORDS

Time-Domain Optimization

Amplifiers

Distributed Processing

Message Passing Interface

Genetic Algorithms

Computer-Aided Design

Integrated Circuits Design Automation

PALAVRAS-CHAVE

Optimização no Domínio do Tempo

Amplificadores Analógicos

Processamento Distribuído

« Message Passing Interface »

Algoritmos Genéticos

Desenho Assistido por Computador

Projecto Automático de Circuitos Integrados

ABBREVIATIONS

DAC	Digital-to-Analog Converter
ADC	Analog-to-Digital Converter
MADC	Multiplying Analog-to-Digital Converter
MOS	Metal–Oxide–Semiconductor
MOSFET	Metal–Oxide–Semiconductor Field-Effect-Transistor
PMOS	P-channel MOSFET
NMOS	N-channel MOSFET
CMOS	Complementary Metal–Oxide–Semiconductor
BiCMOS	Bipolar Complementary Metal-Oxide Semiconductor
OPAMP	Operational Amplifier
OTA	Operational Transconductance Amplifier
SC	Switched-Capacitor
DC	Direct Current
IEEE	Institute of Electrical & Electronics Engineers
SoC	System-On-Chip
IC	Integrated Circuit
AWE	Asymptotic Wave Evaluation
SPICE	Simulation Program with Integrated Circuit Emphasis
BSIM	Berkeley Short-Channel IGFET Model
EKV	Mathematical model of MOSFET transistor
BSP	Behavioral Signal Path
AMD	Advanced Micro Devices
AMS	Austrian Micro Systems
UMC	United Microelectronics Corporation
CAD	Computer Aided Design
EDA	Electronic Design Automation
IP	Intellectual Property
AI	Artificial Intelligence
SA	Simulated Annealing
GA	Genetic Algorithms
GP	Geometrical Programming
NN	Neural Networks

DSP	Digital Signal Processing
GBW	Gain-Bandwidth Product
CMRR	Common-Mode Rejection Ratio
PSRR	Power Supply Rejection Ratio
SNR	Signal-to-Noise Ratio
THD	Total Harmonic Distortion
UGB	Unity-Gain Bandwidth
UGF	Unity-Gain Frequency
RF	Radio Frequency
GUI	Graphical User Interface
VLSI	Very Large Scale Integration
VHSIC	Very-High-Speed Integrated Circuit
VHDL	VHSIC Hardware Description Language
DRC	Design Rule Check
LVS	Layout *Versus* Schematic
UNIX	Uniplexed Information and Computing System
LINUX	Linus Torvald's UNIX (flavor of UNIX for PCs)
GNU	GNU’s not UNIX!
GPL	GNU Public License
EDIF	Electronic Design Interchange Format
CIF	Caltech Intermediate Form
GDSII	Graphic Database System II
POSIX	Portable Operating System Interface [for UNIX]

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

The electronic industry is increasingly focused on electronic devices that contain more and more features. Furthermore, these features are supposed to occupy the smallest possible volume, have the highest performance and as much autonomy (battery) as possible. A direct consequence of these factors is the design of circuits with higher complexity and integration of complete systems on a single chip (SoC), as exemplified in Figure 1-1. Furthermore, the market demands more circuits with better performance in a shorter development cycle, which makes the design cycle even more difficult and complex.

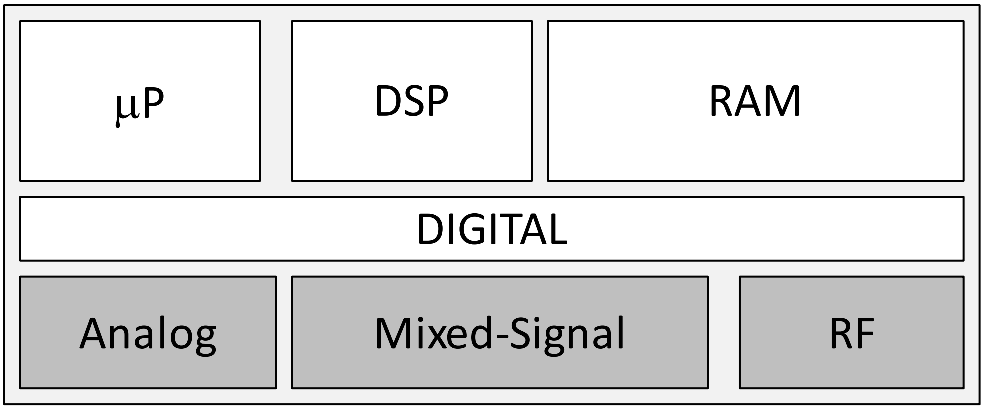


Figure 1-1 Analog mixed signal system on chip[1]

Although most of these features are intensively performed by digital circuits, the interaction with the real world is achieved by analog circuits. Since the signals in the real world are (still) analog signals. The interaction with these analog signals will, ultimately, be through analog-to-digital (A/D) and digital-to-analog (D/A) converters. Thus, it is necessary the coexistence of digital and analog circuitry in the same silicon die, if a SoC is to be implemented.

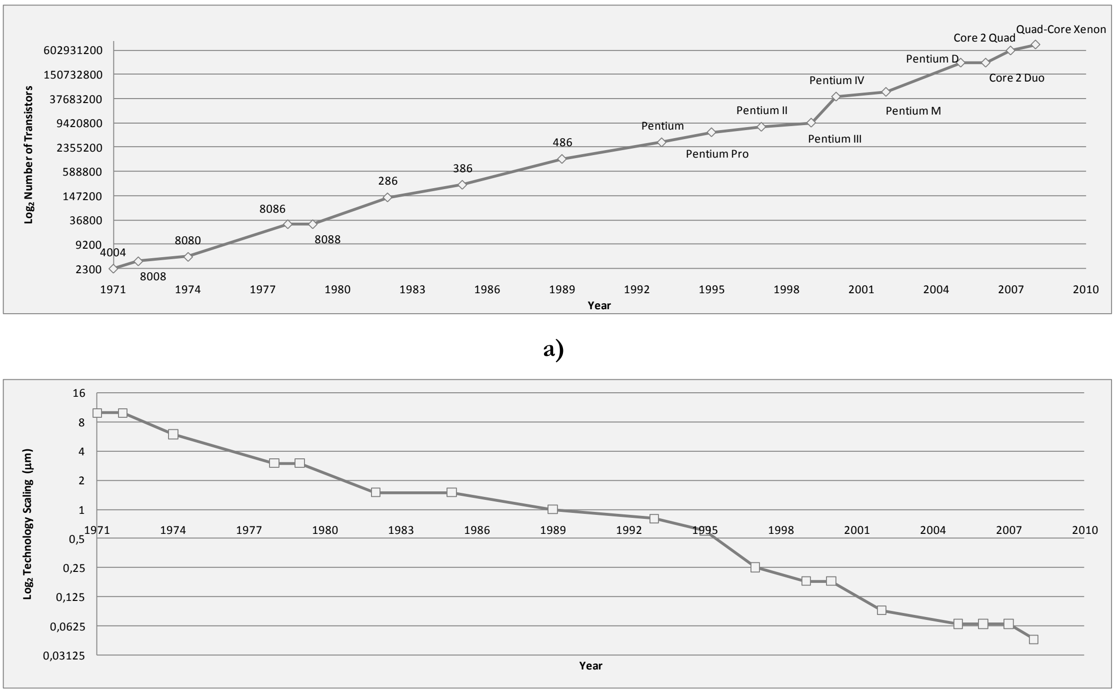
In the design of digital circuits, there are several tools that facilitate the development work [1]. For instance, digital circuits tolerate a fair amount of high order effects and modeling errors that could ruin analog circuitry performance. Analog design requires tools that can deal with the superior complexity of the circuit behavior and device modeling. Although currently, analog circuits occupy less area in SoCs, they require the longest developing time and effort, and it is a task that must be performed by engineers with a high degree of knowledge [2]. As someone mentioned – "*Analog design is somehow considered an art*" [3].

Most of the time, in analog design, is spent in the optimization of the circuits. The non-linear relation between the dimensions of the components and the specifications of circuits is a complex problem. Since there are complex relations between input and output variables and since every design variable affects multiple performances. Each design can have multiple good solutions depending on the initial specifications. Thus, due to the huge design space, it becomes extremely difficult for human designers to manage a good compromise between all the specifications, and the design variables. Furthermore, it is necessary to study the sensitivity of the circuit to the manufacturing process, supply voltage, and temperature (PVT) variations. There is no systematical way of producing new analog designs. Even intellectual property (IP) reuse requires an expert designer or an expert system to map designs in new technologies. All the factors, previously mentioned, make analog design the bottleneck of SoC design. This is a problem because the percentage of analog design in SoC rises every year, based on predictions from the IBS Corporation[4].

Good design methodologies are needed to manage the complexity of analog design, in order to better explore the space of values of design variables and thus, reduce the effort, time and cost of production of new analog circuits.

A concrete example of the increasing difficulty of the design of analog circuits is associated to the increase of the specifications such as conversion-rate and dynamic linearity of the A/D interfaces. This implies the design of operational amplifiers (*opamps*) and operational transconductance amplifiers (OTAs), with increasing DC gains and gain-bandwidth products (GBW). According to the literature, in order to achieve these high requirements, using the latest sub-micrometer manufacturing technologies and required low power specifications, it is necessary to employ multistage amplifiers topologies. This technique partially overcomes the low value of drain-source resistance, r_{ds}, of transistors with sub-micrometer channel dimensions (short-channel) and with its variability as well. However, cascading several gain stages implies the use of complex compensation techniques in order to obtain stable amplifiers with a large GBW value. The resulting amplifier transfer function has several poles (some of them complex conjugated pairs) and zeros, making the amplifier design a complex task. Therefore the final design accuracy depends on the availability and quality of a powerful optimization algorithm. Another issue, when using deep sub-micrometer manufacturing technologies is the high order effects in the electrical characteristics of the transistors, e.g. short-channel effects. These are quite relevant and, only advanced models such as BSIM3v3 [5], BSIM4 [6] and PSP [7], can provide the required accuracy for calculating the I-V characteristics of the transistors.

Moore's law has stipulated that the number of transistors inside integrated circuits (IC) doubles every two years [8], as shown in the Figure 1-2. The problem is that the device size is reaching atom size, which imposes a limit to further scaling down of the transistors. Another issue is economical, it is expensive to build new IC, and the small size promotes high order effects and demands precise fabrication tools [9]. So, probably, the solution for circuit design innovation is on the Electronic Design Automation (EDA) side. It needs to came up with novel methodologies and tools to help better designing circuits and systems.



a)

b)

Figure 1-2 Moore's law [8]: a) Number of transistors per processor *versus* year; b) Technology scaling *verus* year

The work presented in this thesis addresses all this challenges, in particular, the problem of amplifier's optimization, describing a methodology based on the analysis, in the time domain, of the step response of the amplifier. This design methodology allows the analysis and optimization of complex topologies of amplifiers with transfer functions that can have an unlimited number of poles and zeros.

The optimization process uses a genetic algorithm with parallel/distributed processing, integrated with the code of the models of transistors, BSIM3v3[1]. The scientific community has been making an effort in this direction. A good study of techniques and tools to help the design of analog circuits is available in [2].

1.1 Analog Design Flow

The simplified view of the steps required to design an analog system is depicted in Figure 1-3.

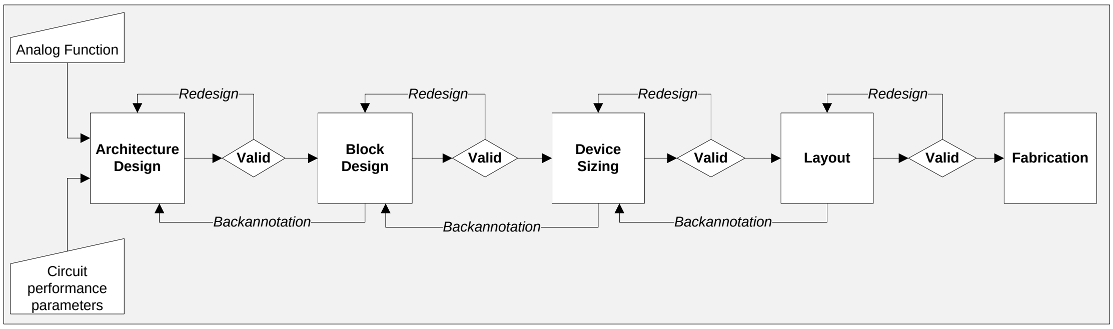


Figure 1-3 Analog design flow

The global design inputs are: the analog function to be implemented and the specifications of the function. The first step is to determine a suitable architecture, which should meet the given specifications.

The process then continues by decomposing the architectures into high-level blocks. Each block is further decomposed into low-level block until the corresponding circuit level is reached. The verifications are carried out at each stage, providing the lower-level block specifications. A back-annotation is also executed backwards to the higher-level block, if necessary. At this point, optimization helps to choose the best blocks for the given set of specifications.

[1] This model was adopted, rather the improved version BSIM4, simply because most of the demonstrations were designed on a 130 nm CMOS technology. However, the extension for BSIM4 is relatively straightforward

After the building blocks are well established, they are mapped into circuit-level devices. Each device is sized properly to ensure that the circuit performs according to the respective block specifications. This task is a multi-variable optimization procedure that must meet multiple specifications.

Finally, the layout produces the different layers mask based on the device sizes. These plans are then sent to fabrication after exhaustive simulations of the extracted layout (XRC - Extracted Resistance and Capacitances).

During the design phases, successive verifications are carried out. In the case of failure, the respective design stage must be revised. These results can be used to improve the previous design stage (back-annotation). Also, depending on the severity of the failure, the design process can reverse several phases.

The focus of this work is mainly on circuit optimization. Therefore, the next section discusses in more detail the circuit-level sizing.

1.1.1 Circuit-level design

The circuit design is characterized by the sequence shown in Figure 1-4.

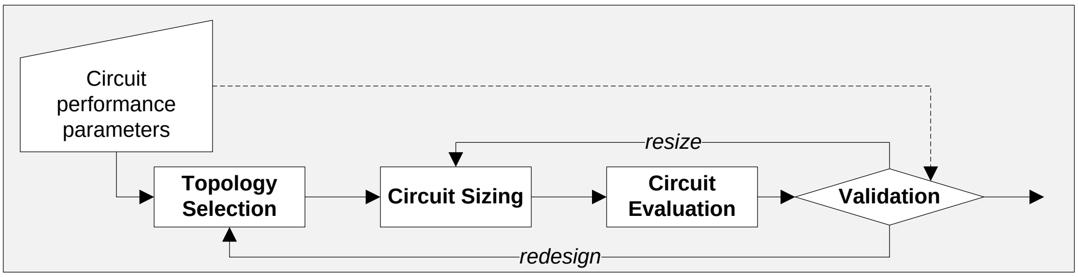


Figure 1-4 Analog circuit design

Based on the circuit performance parameters obtained from the block level design, the selection of the circuit topology takes place. The next step is the circuit instantiation, with circuit devices being sized. The circuit instance is evaluated and validated against the defined indicators. The design process continues to layout, if validation is succeeded, or, in case of failure, redesign is needed.

1.1.1.1 Topology Selection

The selection of the best topology should choose the best candidate that meet all the circuit performance parameters. This process can be based on previous design instances, or designer knowledge (in addiction to some key calculations), and/or rules of thumb. If the circuit validation fails, the amount of failure determines if only a circuit resize is needed or if the selection process should go back and select a more suitable topology. After the topology selection, the design continues with the component sizing.

1.1.1.2 Device Sizing

This task modifies the design variables in order to meet the circuit performance parameters. The design variables are the values and sizes of each transistor, capacitor and resistor. Some of these components might get their values directly from the specifications but the majority of the sizes must be computed, based on the topology and specifications.

The relation between design variables and circuit specifications is nonlinear. It is a multi-variable with a multiple objective function. These facts make it a complex task. On simple circuits, the process relies on designer knowledge and simple calculations, based on simplified (level 1 or 2) device equations. When considering more complex circuits, sub-micrometer technologies and state-of-the-art design demand, the design task cannot rely on simplified device models. Accurate methods and models involve more time and increased computation effort.

Next, validation is carried with a circuit simulator and the results are compared to the specifications. Most certainly, the first results do not meet the specifications, and thus, the devices must be resized. Even on simple circuits, the decision on what parameters to change is not trivial. Rules of thumb and the designer expertise could lead to a good hit. Nevertheless, with complex circuits, this task cannot be achieved by humans only. Some commercial simulators provide a functionality that sweeps circuit parameters and provide some insight about the parameter influence on circuit performance. Parameter sweep is done at the cost of several simulations, which is time and computational costly.

The sizing and validation cycle must succeed after some iteration, otherwise the redesign process back iterates to select a different topology. Topology selection is a key task to avoid redesign. In a worst case scenario, the topology selections do not provide an acceptable performance match; the specifications for the circuit are too tight and should be revised or, as an alternative, a different technology node should be selected.

1.2 Motivation

As stated before, analog design might be considered an art, there is no systematical way of producing new designs. Circuit specifications affect multiple performance parameters. Changing a design variable interferes with several performance parameters. Even if the relation between specifications, design performance parameters and design variables are well known, it is a complex operation for a human mind to take in account. For instance, in a given amplifier, changing a transistor width can improve the low frequency gain but, it may cause a decrease of the GBW value.

Even if it is possible to manage the circuit sizing complexity, handling the nonlinear relations and complex calculations, there is the problem of the design-to-fabrication time. The time-costly operations in circuit sizing do not leave much freedom to evaluate all the design space.

The complexity and the amount of variables involved in the sizing process, handled manually, without process and data integration, are error-prone.

To improve systems yield there are two well-disseminated methods that may be used combined: process corners and Monte Carlo simulations. The foundries provide set of the fabrication technology parameters, which include the nominal and the worst-case values. The values of the last set are derived from process corners. In each Monte Carlo simulation, the design and process parameter values are altered based on statistical distribution. Often, during sizing step, design variables calculations are simply based on nominal values of the fabrication process. The corner values and Monte Carlo simulations are only considered after the last sizing step that succeeded. As a consequence, the circuit design ends with a considerable number of time-costly simulations.

In summary, design tools need to be developed, to efficiently cope with the analog design bottlenecks. These include the following:

- **Decrease design time**: the processing capacity of a computer is largely higher than a human, manually, designing a circuit;
- **Reduce cost**: reducing the time on design will decrease the time designers spent on circuit sizing (design effort);
- **Decrease errors**: integrating and automating the design process frees designers from routine tasks and decrease errors resulting from human intervention, which normally follows a trial-and-error approach;
- **Increase circuits performance**: using computers in the design process more processing capacity is available. This facilitates a larger design space exploration,

including process corners and Monte Carlo simulations. Thus both, the design performance and Yield are improved.

1.3 Scope of this Thesis

The main subject of this thesis is the study and application of new techniques and methods to enhance the circuit (amplifier) sizing and optimization stage of analog design flow. This contributes to the improvement of the design automation task of analog circuits.

The tool developed and described in this work is able to compute the size of devices to meet the performance specifications given for an amplifier, thus contributing to decrease the time-to-tape-out, and to first-pass success.

1.4 Main Research Contributions

Two major contributions can be highlighted in this work:

1. An optimization EDA platform for amplifiers based on genetic algorithms [10] and distributed processing [11], following an efficient time-domain equation-based/simulation-based approach. Furthermore, the developed optimization tool is fully based in open-source code [12].
2. This platform was successfully applied in particular, in the design of two-stage amplifiers in the following way:
   - I. how an extra degree-of-freedom can be added to the design space allowing enhanced performance [13];
   - II. how to achieve optimum compensation of two-stage amplifiers [14];
   - III. how to achieve a DC gain above 100 dB, with gain-boosting techniques and optimization [15];
   - IV. how to achieve a power efficiency figure-of-merit (FOM) in a new class A amplifier, comparable to similar amplifiers employing class AB output stage through the optimization of and inverter-based self-bias two-stage amplifier [16], [17]. This work was demonstrated in silicon and the experimental evaluation results are shown in section 5.4 of chapter 5.

Moreover, the demonstration of the practical effectiveness of the developed EDA platform has been shown throughout the design of several energy-efficient pipeline [18] and two-stage algorithmic analog-to-digital converter (ADC)[15]. Some of these circuits have been, later on, laid out, prototyped and evaluated. Hence, the targeted design performance parameters of the designed amplifiers were indirectly confirmed in silicon though experimental evaluation of the ADC [19].

The main contributions of this work are focused on the suitability of novel methods and techniques for optimization of complex analog circuits, in particular, two-stage amplifiers: the genetic algorithms [20] as the base for optimization, time-domain for circuit analysis [14] and distributed processing [11] for platform performance enhancement. As research contribution and work assessment, four application examples were also published:

1. Optimization of amplifiers for a power-and-area efficient multiplying digital-to-analog converter (MDAC) [15];
2. Transistor sizing and compensation capacitance schema of multi-stage amplifiers [14];
3. Optimization of amplifiers for MDAC stages, low-voltage and low-power efficient high-speed moderate resolution pipelined ADC [18].

This research work has been translated into the following authored / co-authored publications:

- M. Figueiredo, **R. Santos-Tavares**, E. Santin, J. Goes, "A Two-Stage Fully-Differential Inverter-based Self-Biased CMOS Amplifier with High Efficiency", submitted by invitation to *IEEE Transactions on Circuits and Systems I* (TCAS I), October 2010.
- E. Santin, M. Figueiredo, **R. Tavares**, J. Goes and L. B. Oliveira, "Fast-Settling Low-Power Two-Stage Self-Biased CMOS Amplifier Using Feedforward-Regulated Cascode Devices", to appear in the *IEEE International Conference on Electronics Circuits and Systems* (ICECS), Athens, Greece, December 2010.
- M. Figueiredo, E. Santin, J. Goes, **R. Santos-Tavares**, G. Evans, "Two-Stage Fully-Differential Inverter-based Self-Biased CMOS Amplifier with High Efficiency", *IEEE International Symposium on Circuits and Systems*, Paris, France, May 2010.
- **R. Santos-Tavares**, N. Paulino, J. Higino, J. Goes, J. P. Oliveira, " Optimization of Multi-Stage Amplifiers in Deep-Submicron CMOS Using a Distributed/Parallel Genetic Algorithm", *IEEE International Symposium on Circuits and Systems*, Seattle, USA, May 2008.
- B. Esperança, J. Goes, **R. Tavares**, A. Galhardo, N. Paulino, M. Medeiros Silva, "Power-and-Area Efficient 14-bit 1.5 MSample/s Two-Stage Algorithmic ADC based on a

Mismatch-Insensitive MDAC", *IEEE International Symposium on Circuits and Systems*, Seattle, USA, May 2008.

- **R. Santos-Tavares**, N. Paulino, J. Goes, J. P. Oliveira, "Optimum Sizing and Compensation of Two-Stage CMOS Amplifiers Based On a Time-Domain Approach", *IEEE International Conference on Electronics*, Circuits and Systems, Nice, France, December 2006.
- **R. Tavares**, B. Vaz, J. Goes, N. Paulino and A. Steiger-Garção, "Design and Optimization of Low-Voltage Two-Stage CMOS Amplifiers with Enhanced Performance", *IEEE International Symposium on Circuits and Systems*, Bangkok, Thailand, May 2003.
- B. Vaz, N. Paulino, J. Goes, R. Costa, **R. Tavares** and A. Steiger-Garção, "Design of Low-Voltage CMOS Pipelined ADC's using 1 pico-Joule of Energy per Conversion", *IEEE International Symposium on Circuits and Systems*, Arizona, USA, May 2002.
- B. Vaz, R. Costa, N. Paulino, J. Goes, **R. Tavares** and A. Steiger-Garção, "A General-purpose Kernel based on Genetic Algorithms for Optimization of Complex Analog Circuits", *IEEE Midwest Symposium on Circuits and Systems*, Dayton, Ohio, USA, August 2001.

1.5 Outline

Already on this chapter, an overview of the document context has been provided, highlighting the scope, motivation and research contributions of the work. The rest of this thesis is organized as follows:

- Chapter 2 starts with an overview of the circuit sizing/optimization approaches, previously implemented. It presents a comparative summary of the described approaches. Then, some brief considerations about layout automation are given. Although, layout automation is out of scope of this work, it is included in the future developments discussion. Next, some of the freely available tools and respective source-code/open-source, are presented. The chapter ends with the description of the proposed work and how it may contribute to innovate the circuit optimization task.
- Chapter 3 describes the proposed optimization methodology based on time-domain analysis of the amplifiers. A top-dowm approach is followed, starting with the definition of the main steps of the proposed methodology. Then, this chapter continues with the description of how to compute the time-domain step-

response of an amplifier. After, the method to extract the transfer function of an amplifier is presented, based on the behavioral signal path (BSP). Moreover, the most common performance parameters of amplifiers often used in optimization are presented. Also, it is described the method to compute the closed-loop transfer function of amplifiers, when using switch-capacitor circuits techniques. Finally, a summarized comparison of the time-domain *versus* frequency-domain optimization methodologies is done.

- Chapter 4 explains the implementation of the proposed optimization methodology in a software platform. First, a general overview of the platform and the main blocks is given. Then, a brief overview of the optimization algorithm, based on genetic algorithms, is presented. This chapter continues with the description of how the circuit code is integrated on the platform. Then, the exported statistics and results are explained. Finally, the version of the platform that employs the distributed computation concept is illustrated.

- Chapter 5 presents some case-study examples that validate the efficiency of the proposed optimization methodology and platform implementation. The first example is a two-stage cascode amplifier with active biasing. It demonstrates that the methodology is capable of handle the extra complexity introduced by adding an extra degree-of-freedom to enhance the performance. The second example shows how to use the proposed methodology in order to achieve optimum compensation schema and size for two-stage amplifiers with a cascode first stage. The third example demonstrates that the methodology is suitable to handle the high complexity of a two-stage gain-boosted amplifier, with two additional satellite amplifiers. The optimized amplifier instance achieves a DC gain above 100 dB. In the last example, the methodology is used to optimize a novel topology of two-stage self-biased amplifiers. A comparison of the optimization results on frequency-domain *versus* time-domain optimization is presented. In this example, silicon results are provided to demonstrate the effectiveness of the developed methodology.

- Finally, chapter 6 draws the conclusions and discusses the future work.

2 Computer Aided Design of Analog Circuits

This chapter presents a survey of known methods for analog design automation and a detailed analysis of the implementations of these methods.

As previously observed, the objective of design automation is to decrease the design time and free the designer from repetitive tasks to more qualified and useful ones. As more and more of these repetitive tasks are carried out by computers, fewer errors should occur during design process.

Considering the analog design flow presented in section 1.1, the goal is to have each design task executed by a computer tool, or by a platform - set of tools -. Currently, only a small set of design tasks are performed by software tools. Data integration and tool interaction, among the different design phases, are not truly available in practice, yet.

2.1 Circuit Sizing/Optimization

To handle the circuit sizing task, the automated design methods described on literature, followed mainly two approaches:

- Knowledge-based;
- Optimization-based.

Knowledge-based approaches use a set of predefined equations and procedures -design plans- to compute the size of each device. Optimization approaches exploit the strength of algorithms on making decisions during the sizing. Moreover, this last category maybe divided into:

- Equation-based;
- Simulation-based;

- Asymptotic Wave Evaluation-based;
- Learning-based.

2.1.1 Knowledge-based approaches

The first attempts to automate the design process implemented a knowledge based approach: IDAC [21], BLADES [22] and OASYS [23].

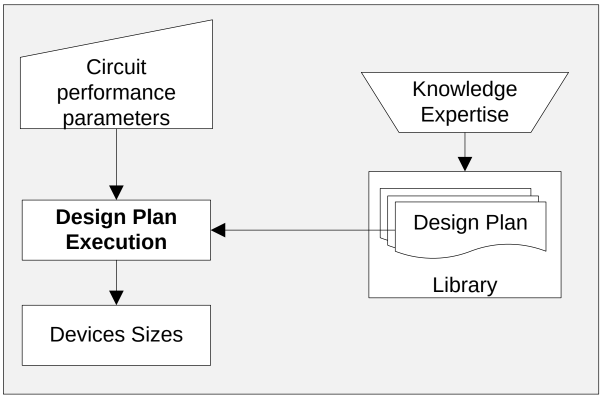


Figure 2-1 Knowledge-based circuit sizing

Figure 2-1 gives a general idea of this approach. The input and output data are circuit performance parameters and devices sizes, respectively. There is a library of design plans created by expert designers that specify how the devices sizes are computed, without any further optimization. Since the number of devices sizes exceeds the number of circuit performance parameters, design plans also include knowledge-based procedures to select part of the sizes and reduce the number of degrees of freedom left for functions calculations.

Although the execution time of a design plan is short, constructing the design plans is considerably time consuming and requires an expert designer to execute this task. Typically, these approaches are based on simple device models, which result in a poor estimation of the circuit performance. Mainly, these implementations evaluate the performance of circuit candidates using frequency domain parameters.

A library of specific design plans for different circuit topologies is used by IDAC [21]. A small variation in the topologies results in a complete new design plan to be saved in the library. To cover a wide range of scenarios, a large number of design plans must be realized. Each design plan is a set of circuit equations that compute the circuit specifications and are created by an experienced designer. After applying the design plan, the results are verified with an electrical simulator. If it fails, the designer readjusts the specifications and, executes the design plan, again. If there is a design plan, in the library, the circuit sizing is a fast task; otherwise, it takes a lot of time to setup the new design plan. Moreover, the equations are based on simplified models, which originate approximated circuit performance results. This tool was made available, commercially, by Mentor Graphics Corporation, in 1987 [24].

The divide-and-conquer strategy was used in BLADES [22]. In this implementation the circuit is decomposed in basic blocks, e.g. current mirrors; input stage; output stage. In each block, at transistor level, the device sizes are defined with values stored in lookup tables, previously filled with simulation results. This means that a high number of tables exist, for the variety of specifications, device models, and fabrication technologies. To select the blocks that constitute the complete circuit, it uses artificial intelligence (AI) rules in combination with lookup tables. Here, the setup time is also the main drawback, since one needs to build the design rules and lookup tables, or adjust the existing ones prior to the design start. Targeting accuracy, although the lookup tables are built using precise simulation results, these values are computed for sub-blocks and not as for a complete circuit, at once.

Another implementation, OASYS [23], makes use of hierarchy decomposition. Several hierarchy levels can be produced and each hierarchy level generates different sub-blocks. Then, each sub-block is a different design task with specifications derived from circuit performance parameters defined initially. For each sub-block, all the candidates are computed and the one with the best result is selected. During hierarchy decomposition, the selection of each sub-block topology is based on the performance estimated for each one. If significant discrepancy between estimated and computed circuit performance exists, there is a backtracking scheme to select a new sub-block. At transistor level, knowledge based circuit sizing is applied. Although this tool is based on simple device models, it requires a considerable time to build a new design plan. Even considering the reuse of the knowledge of sub-blocks already gathered.

Qualitative reasoning is employed in ISAID[25] to adjust the performance of the complete circuit. This method defines qualitative relations between device sizes and circuit performance parameters, e.g. *IF width of transistor_1 increases, THEN the DC gain will increase*; gathered from expert engineer knowledge. The device sizes are adjusted using a large number of qualitative rules. Circuit performance evaluation is carried out in the frequency-domain and the sign of the gradients of the different performance parameters is used to determine the effect of changing a particular design variable.

2.1.2 Optimization-based approaches

These approaches incorporate an optimization algorithm to guide the circuit sizing process to obtain an optimum circuit. The diagram on Figure 2-2 represents the basic idea of these techniques.

The algorithm iterates through a cycle where design variables are adjusted until the circuit performance parameters meet the initial specifications. Each iteration starts with the setup of a new circuit instance, with size values chosen from the design space. Next, the circuit is evaluated to determine the circuit performance. The circuit performance parameters are then matched with the initial specifications to compute how close to specifications the instance is. The iterations end when a circuit instance fulfils the specifications or, after some iterations, if it fails to meet the specifications. This class of approaches comprises different combinations: of search method; of circuit instance analysis; and of computer processing techniques.

- Search algorithm:
    - Gradient-based
        - Steepest Descendent
    - Geometric Programming
    - Stochastic Search
        - Simulated Annealing
        - Genetic Algorithms
- Circuit evaluation:
    - Equations-based;
    - Simulation-based
    - Asymptotic Wave Evaluation-based;
    - Learning-based.
- Computer processing:
    - Centralized;
    - Distributed;
    - Parallel.

Next, these classes of approaches are further detailed.

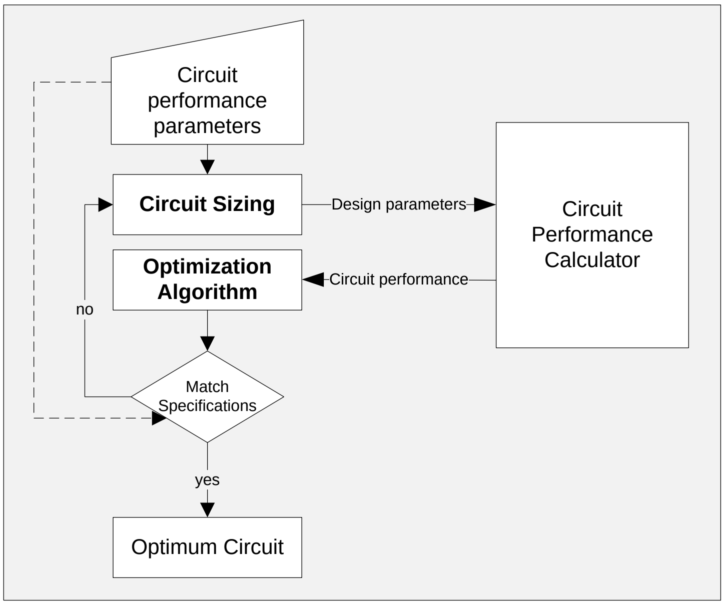


Figure 2-2 Optimization-based circuit sizing

2.1.2.1 About search algorithms

The search algorithm portions of the optimization approaches mentioned throughout this document are described in the following subsections.

2.1.2.1.1 Gradient-based

Gradient-based search algorithms assume that the problem can be translated into a real-valued function, *F(x)*, differentiable in a neighborhood of a given point, *a*. Also, *F(x)* decreases as one moves from the point *a* in the direction of the negative gradient of *F*, at *a*. Consequently, it starts to guess an initial value, X_0, as being the minimum of *F*, and continues to progress towards the minimum, with the sequence: X_0, X_1, X_2, X_3 … in such way that

$$F(X_0) > F(X_1) > F(X_2) > F(X_3) > \ldots$$

as depicted in the Figure 2-3.

The disadvantage of this method is the guessing of the starting point. Depending on the starting point, it can lead the search to a local minimum instead of a global minimum.

Although this is not an efficient algorithm, it was used in combination with other forms of search, and/or search refinement. OPASYN[26] implemented a multiple search instances with different starting points. On FRIDGE [27] after an initial search with a global-oriented search algorithm, it refines the search based on a gradient algorithm.

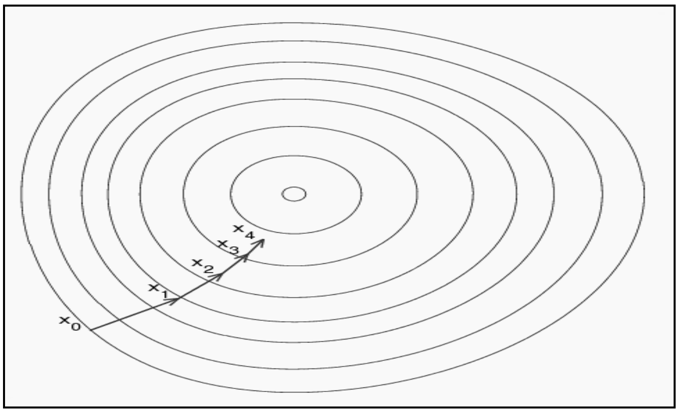


Figure 2-3 Gradient-based search illustration

2.1.2.1.2 Geometric programming

Geometric Programming (GP) is an optimization method based on posynomial functions. For example:

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 1, & i = 1,\ldots, m \\ & g_i(x) = 1, & i = 1,\ldots, p \end{array} \qquad (2.1)$$

where f_i are posynomial functions, g_i are monomials, and x_i are the optimization variables. (There is an implicit constraint that the variables are positive, i.e., $x_i > 0$.) In the standard form of a geometric program, the objective must be posynomial (and it must be minimized); the equality constraints can only have the form of a monomial equal to one, and the inequality constraints can only have the form of a posynomial less than or equal to one. The weakness of this approach is that not all problems are possible to be modeled with posynomials. In some cases it is possible by approximation the objective function, which could lead to a less accurate final result. The most positive aspect of this algorithm is the execution time. Once, and if the problem can be described into a geometric format, the processing time is relatively short [28].

2.1.2.1.3 Stochastic search

Another class of search algorithm is based on probabilistic elements and/or moves. This survey identified two subclasses: Simulated Annealing (SA) [29] and Genetic Algorithms (GA)[30].

The SA algorithm is based on discrete values movements to the neighborhoods of the present point, until the optimum point is reached. The starting point is randomly generated. Then, each move is selected, randomly, to the neighborhood configuration with the best probability of being an optimum one. Since it is based on discrete values, it provides only an acceptable approximation, rather than the best possible solution. The main drawback is that it can easily find a local minimum, rather than a global minimum, depending on the chosen starting point. ASTX/OLX is an example that implements this algorithm. As a search complement, the simulated annealing, implemented in FASY [42], is completed with a fine tuning based on the gradient algorithm. On other hand, simulated annealing is used for fine tuning, after the global search carried by a fuzzy-logic based algorithm in the APLAYDIN [45] implementation.

Similarly to the SA, GA also starts the search with an initial and randomly generated set of variable values – population of individuals. Then, on each move – generation -, the optimization progress consists of selecting the best classified individual(s), apply crossover and mutation operations until the optimum individual is found. The classification of each individual is the objective function.

Compared to the simulated annealing, two improvements arise here. The evolution is not based on fixed values and, on the other hand, since crossover and mutation operators are used, theoretically, the entire space design is covered. Consecutively, the points/individuals evaluation time is proportional to the problem complexity. Apparently, only the recent implementations adopted this type of search algorithm (stochastic search). Some used it as the primary global search, MAELSTROM [43] and ANACONDA [28], and GENOM [46] as the main, and only, search algorithm. The work presented here also adopted the GA for the search algorithm.

2.1.2.2 About circuit evaluation

The main idea throughout all optimization design tools is, basically the same, four classes can be further stated, based on the circuit evaluation: equations-based; simulation-based; asymptotic wave evaluation-based; and learning-based.

2.1.2.2.1 Circuit evaluation based on equations

There are several implementations of this approach: OPASYN [26], STAIC[31], MAULIK[32], ASTRX/OBLX[33], AMGIE[34], GPCAD[35], ISAID[25]. The circuit performance parameters are calculated by equations, as shown in the Figure 2-4. The equations are obtained, either, manually by expert designers, or by symbolic analyzers, directly from the circuit description, e.g. netlist[36]. Although these cost less run time, the accuracy of these approaches is quite low. The equations relay on a simplified model to describe the behavior of the devices.

Mainly, these implementations evaluate the performance of circuit candidates using frequency domain metric's equations.

Although some implementations provide some degree of hierarchy in the equation-models [31] usually, setting up a new circuit evaluator/calculator, is a time consuming task.

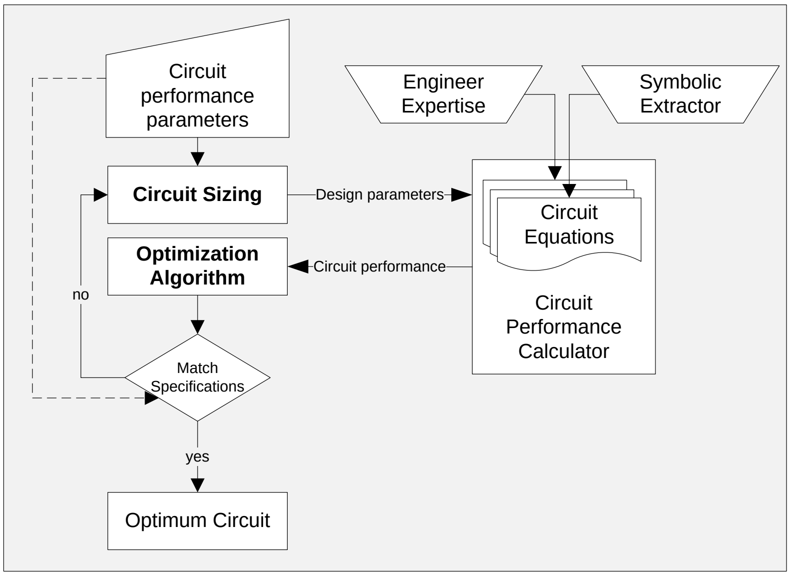


Figure 2-4 Equation-based circuit optimization

In OPASYN [26] the analytic circuit models are specifically derived for each amplifier and collected in a database. These are simplified models where independent parameters are eliminated to reduce the number of design variables to be computed, or computed directly from circuit sizing. They also include fitting parameters equations and upper and lower bounds for the design variables. Fitting parameters are used to refine the equation-based model, with values obtained from SPICE[37] simulations, carried out during the optimization. To simplify the task, the independent design variables are computed from the circuit sizing. A steepest descent algorithm is initiated at several starting points, with different sets of design variables values. At the end of the algorithm search, the best result is selected. This helps finding a global minimum, preventing the algorithm from being caught at a local minimum.

STAIC [31] methodology considers a two step optimization. A first design space scan is performed on grid based points, with simple circuit equations. The purpose of this first step is to provide the designer with a better insight on possible trade-offs. Next, the results from the grid based scan are used as the starting point for the designer to perform an additional refined search task, manually. The last optimization step employs a simulation based evaluation with more accurate models. The simplified device models employed in the tool are frequency domain equations, which provide a rough approximation of the circuit performance parameters. At the end, in the second optimization step, the designer uses more precise models, which give accurate results and, hypothetically, by a circuit simulator, that provides frequency and time domain parameters calculation.

In the MAULIK [32] a branch-and-bound optimization technique is applied to find the suitable topology and determine the device sizes. The circuit performance parameters calculations are based on a relaxed DC formulation. Since the DC equations are not analytically solved, it provides run time and computation effort to allow using high accuracy models, e.g. BISM, and accurately compute the device parameters. Nevertheless, with relaxed DC formulation it is not guaranteed that the circuits are feasibly. The small-signal circuit equations are simplified ones, derived manually.

Despite the fact that AMGIE [34] only uses equation-based circuit performance analysis, it combines global and local optimization methods. To increase convergence, on a first pass, it employs global search algorithm – simulated annealing –. After, for fine tune, a gradient-based algorithm is applied. It includes a symbolic analyzer to automatically generate frequency domain circuit equations, which are simplified to reduce the computation effort inside optimization loop. Time domain equations should be obtained and provided by the designer. Although, the partly automated process of extracting the circuit equations, it always requires an expert designer and some preparation time.

An attempt to formulate the circuit equations as posynomial was made with GPCAD [35]. This turns the sizing task into a convex optimization problem, which can find a global minimum in a short time. Accuracy is the main drawback, since equations must be defined as posynomials and accurate device models do not comply with the posynomial form. So, this approach offers short execution time at cost of accuracy [28].

The research effort of GPCAD [35] originated the spin-off Barcelona Design company and its products [38]. This company offered specific circuit Intellectual Property (IP) blocks that were optimized for a given specifications. These IP blocks include a specific optimization engine with the required design equations. The optimization task is formulated as a GP problem and the equations are written as posynomials. Although availably is limited to specific circuits, e.g. data converters, amplifiers; it could be reused and provide a fast method for circuit sizing, and layout design, as well.

2.1.2.2.2 Circuit evaluation based on simulation

Simulation-based circuit sizing uses an electrical simulator in the optimization loop, as circuit instance evaluator, as depicted in Figure 2-5. Furthermore, to achieve more precise results, the simulator links with complex accurate device models. The use of a circuit simulator means extra processing effort and higher optimization time.

In general, a simulator can handle many types of circuits. This fact permits the optimization of a wide range of circuits since the circuit performance parameters used in the cost function are provided by the simulator.

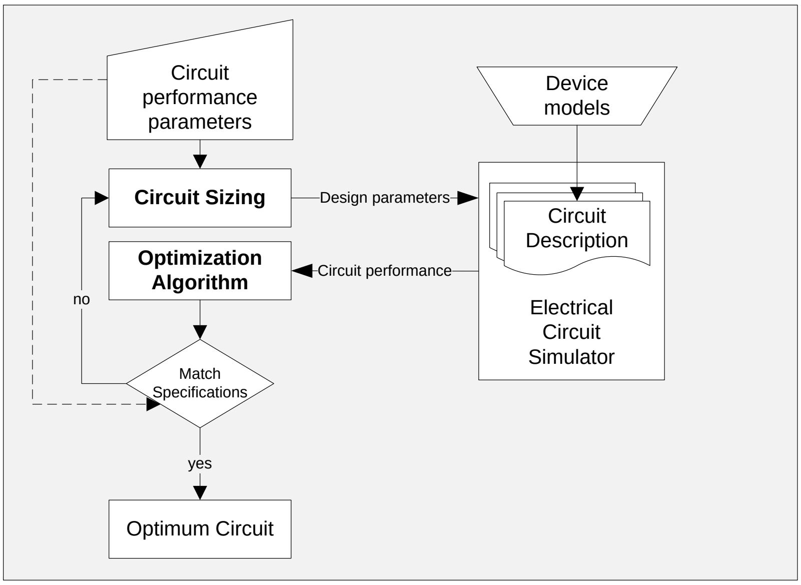


Figure 2-5 Simulation-based circuit optimization

DELGHT.SPICE [39] is a designer-interactive platform with single phase optimization approach based on feasible directions algorithm. It also considers yield optimization and computes the sensitivity of the performance parameters to device variation. Since the run time is the main drawback due to use a circuit simulator, this tool is intended to fine-tune manually designed circuits [40].

SD-OPT [41] is a tool specific to design (and optimize) sigma-delta modulators with a two phase optimization implementation. A first optimization step is performed at higher level, where the system performance is optimized and the low-level circuit specifications are computed. This step uses system equations that characterize the different circuit blocks of the modulator. At circuit-level, the sizing task is assisted by an electrical simulator for circuit performance calculation of each instance. Search algorithms on both phases are based on SA. Although this tool is considered an efficient way of designing modulators, adding new modulators topologies to the design database requires exhaustive analysis [41].

Two different optimization methods are used in FRIDGE [27] and FASY [42]. Both approaches do a global scan of design space, pos-complemented with a local fine tune. Global search is performed with a SA type algorithm, where design variables values are quantizing according to a grid of values, on the design space. This (and previous) grid-points evaluations are stored. This way it avoids multi (re)simulations with the same grid-point values. After the global search, the local search is based on gradients of circuit performance parameters. The circuit parameters are computed by an electrical simulator, on the frequency domain.

MAELSTROM [43] and ANACONDA [28] both have in common the concept of a wrapper interface that enables them to use several commercial simulators to perform frequency-domain evaluation. First, the search engine runs on multiple instances of an optimization algorithm based on SA, in parallel. During optimization all algorithm instances exchange data for better convergence. Then, for fine tuning, the search engine runs, based on GA plus Stochastic Pattern Search (SPS). The employment of distributed processing over a cluster of workstations provided a substantial optimization time reduction.

The authors of MAELSTROM [43] and ANACONDA [28] have made their research work available, commercially, under the name NeoCircuit™, by Neolinear, Inc (acquired by Cadence®)[44].

2.1.2.2.3 Circuit evaluation based on learning-paradigms

A neural-network (NN) provides a fast way of computing the performance parameters for a predefined set of design variables. Although fast, it requires a long training within the space design region of interest. The main idea is illustrated in Figure 2-6.

The amount of training data is proportional to the expected accuracy. This leaves room for possible trade-off between accuracy and run time. Consequently, in this approach the training time can be very large. Normally, the training data is collected with a high performance evaluator, such as a circuit simulator, without any human intervention, and consists of results relaying on highly precise data.

ALPAYDIN [45] is an implementation based on neural-fuzzy performance models, for some performance parameters on the frequency-domain. It can also include user defined equations to compute other parameters. Again, for every new topology, the designer must, manually, supply the performance parameters equations. The DC bias operating point is then calculated by a fast circuit simulator. It is reported that this approach can estimate both linear and non-linear circuit behavior combining Evolutionary Strategies (ES) and SA.

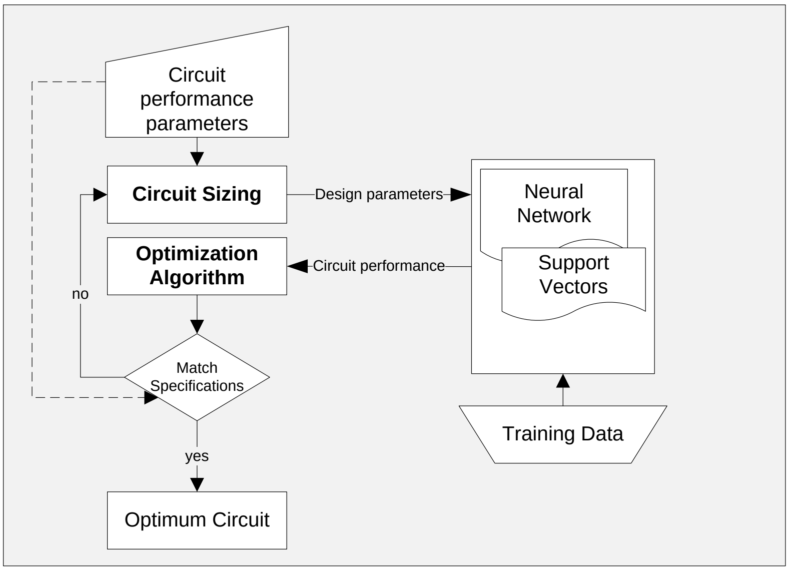


Figure 2-6 Learning-based circuit optimization

Another approach, partially, based on learning data is GENOM[46]. The optimization kernel is based on GA, where the evaluation task is split into two steps. First, using the learning data stored on support vectors[47], the selection operator of GA is guided to rejects the non promising individuals. Then, only the most promising individuals are evaluated by a circuit simulator. The number of circuit evaluations, and consequently, the time consuming is reduced. Since the learning data is collected during optimizations, it takes some initial period to start reducing the number of circuit evaluations. Discarding individuals can be a disadvantage, since genetic data is being dumped. After crossover, poor classified individuals can result in a better offspring.

2.1.2.2.4 Circuit evaluation based on asymptotic wave

A combination of equation and (sort-of) simulation evaluation is also applied in ASTRX/OBLX [33]. The small-signal circuit performance parameters are predicted using asymptotic waveform evaluation (AWE) [48] and a reduced complexity model. All other performance parameters are estimated from circuit equations. AWE is an efficient method to analyze linear circuits and considerably faster than a SPICE-like simulator. Non-linear devices are converted using linear device models, and then AWE is applied. Small-signal parameters of circuit devices are computed using high-accuracy device models, e.g. BSIM [5]. The search algorithm is based on SA.

The drawback of this type of approach is that nonlinear behavioral of circuits has to be approximated with a low order model, which renders some loss in terms of accuracy.

2.1.2.3 About computer processing paradigms

The processing paradigms that worth mention are: the centralized computing, distributed computing and parallel processing. The first two distinguish by where the computation is carried-on: on a single machine or several distributed machine, somehow connected in a network. The later one resumes to multi processes execution at the same time, in parallel, on a single machine.

Centralized approaches lack the performance boost of the distributed/parallel versions, but are relatively straightforward to implement and maintain. Distributed implementations are more immune to hardware failures, since several, independent, computer are being used.

Computers with multi-core processing units and hyperthread concepts were not implemented or not available to the major research individuals until a few years ago. Probably this is the main reason for the approaches described in this document, to use centralized or distributed processing, and not parallelism.

Most optimization approaches is based on centralized processing, except MAELSTROM [43], ANACONDA [28] and GENOM [46], which have the possibility to employ distributed processing.

Implementations	Date	Performance Evaluation	Search Method	Setup	Processing Time	Knowledge Extraction	Evaluation Domain*	Computing
IDAC [21]	1987	Knowledge based	Design plan	- -	+ +	Manually	Frequency	Single processing
DELIGHT-SPICE [39]	1988	Circuit simulator	Feasible directions	+ +	-	N.A.	Frequency	Single processing
OASYS [23]	1989	Knowledge based	Design plan	- -	+ +	Manually	Frequency	Single processing
BLADES [22]	1989	Knowledge based / Lookup tables	Artificial Intelligence	- -	+ +	N.A.	Frequency	Single processing
OPASYN [26]	1990	Simplified Circuit Equation	Multiple Steepest Descent	-	+ +	Manually	Frequency	Single processing
MAULIK [32]	1991	Simplified Circuit Equation	N.A.	- -	+ +	Manually	Frequency	Single processing
STAIC [31]	1992	Simplified Circuit Equation	Design space scan	- -	+ +	Manually	Frequency	Single processing
FRIDGE [27]	1994	Circuit simulator	SA + Gradient	+ +	+	N.A.	Frequency	Single processing
ISAID [25]	1995	Simplified Circuit Equation + Qualitative Reasoning	Gradient (of metrics)	+ / -	+	Manually	Frequency	Single processing
SD-OPT [41]	1995	Circuit equations	SA	- -	-	Manually	Frequency	Single processing
FASY [42]	1995	Circuit Simulator	Fuzzy Logic + SA + Gradient	+ / -	+ / -	N.A.	Frequency	Single processing
ASTRX/OBLX[33]	1996	Asymptotic Wave Evaluation + Equations	SA	-	+ / -	Manually	Frequency	Single processing
GPCAD [35]	1998	Simple Equations as posynomials	Geometric Programming	-	+ +	Manually	Frequency	Single processing
MAELSTROM [43]	1999	Circuit simulator	GA + SA	+	+ / -	N.A.	Frequency	Parallel Processing
ANACONDA [28]	2000	Circuit simulator	Stochastic Pattern Search(GA)	+	-	N.A.	Frequency	Parallel Processing
AMGIE [34]	2001	Simplified Circuit Equation	SA	+ / -	+ +	Symbolic analyzer	Frequency	Single processing
APLAYDIN [45]	2003	Neural Network Data	GA + SA	- -	+	Neural Network	N.A.	N.A.
GENOM [46]	2007	Support Vectors + Circuit simulator	GA	+ +	+ +	N.A.	Frequency	Distributed processing

Table 2-1 Summary of analog sizing implementations

N.A. – information not found in literature, or not applicable -- poor; +/- average; ++ very good ** in the case of optimization of amplifiers*

2.2 Comparative Summary of the Approaches

A common set of characteristics are considered to summarize the circuit design approaches presented in this chapter. These characteristics are summarized in the Table 2-1.

The first column, *performance evaluation*, enumerates the methods used to evaluate the circuit and compute the circuit performance parameters. These influence the processing time and the results accuracy. Accurate evaluators can be a sign of more processing time.

The technique to find a solution for the circuit sizing problem is sorted in the column named *search method*. Some find a solution from a random starting point; others require a good starting point to help the process. Generally, all converge to a feasible and practical solution.

The time used by the approaches is split into two subcategories. The *setup time* refers to the preparation of the problem for the sizing process to begin. In some cases the time needed to extract the circuit equation or to create a design plan can be considerable high. The *processing time* is the period during sizing and/or optimization of a circuit. The number of devices and performance parameters influence the processing time. Furthermore, some circuit parameters are more complex to calculate and demand significantly more processing time.

The *knowledge extraction* column classifies the way the circuit equations and/or design plans are obtained. An automated method should be faster and more error-free than a manually method.

The domain in which the circuits, in the particular case of amplifiers, are evaluated is shown in the column *evaluation domain*. Although the frequency domain provides a faster implementation, time-domain is considered more simple [49].

The *complexity* handled by each approach can be defined as the number of devices and the circuit parameters evaluated. There is not a standard test benchmark circuit with a defined number of devices. Circuit examples used vary from a few transistors to a large number of devices. It is difficult to make a fair comparison between the different tools. Furthermore, each circuit performance parameters computation demands processing time and some are more complex to compute than other. The parameters can also increase the complexity level.

Finally, last column reflects the group that specifies the *computing* type employed in each implementation. Sequential is the most simple to implement, but parallel and/or distributed computing provides a faster and powerful way to get the solution.

2.2.1 Knowledge-based *versus* optimization-based

The first generation of tools was based on knowledge. Using this approach, the circuit parameters are computed through a specific design plan.

The main disadvantage of knowledge-based is the time necessary for the design plan derivation. It is reported [50] that it is longer than a manually design of the same circuit instance. However, once the design is concluded, the circuit evaluation is much faster. This means that a fast design space exploration can be carried out faster than using (conventional) optimization-based implementations.

Design plans must be produced for each circuit and it's variations. This needs to be carried by an experienced designer, which can result in a tedious and error-prone task. Furthermore, an evaluation on the design technology implies a redesign of the design plan. Once more, maintain a library of design plans is time consuming.

The design plans are composed of design equations. These equations are bound to simple ones in order for a human designer to be able to handle them. This simplification results in a poor accuracy and it is not compatible with the modern process technologies and circuit specifications. The simplification of the design plans results in large deviations compared to the more accurate models used nowadays.

In optimization based approaches, the design decisions are made by a universal optimization kernel. The setup time is reduced and provides a general tool to handle a broader number of design problems. The optimization kernel exploits the design space attempting to find an optimal solution for the problem.

2.2.2 Summary of optimization-based approaches

An overview of developed tools over the last three decades is depicted on Figure 2-7. The vertical axis assembles two main categories of tools, in terms of circuit sizing: knowledge-based and optimization-based. Optimization is further partitioned on the circuit evaluation class: circuit equations and electrical simulation. Symbols / and // represent, respectively, the class of computing processing implemented: centralized or distributed/parallel.

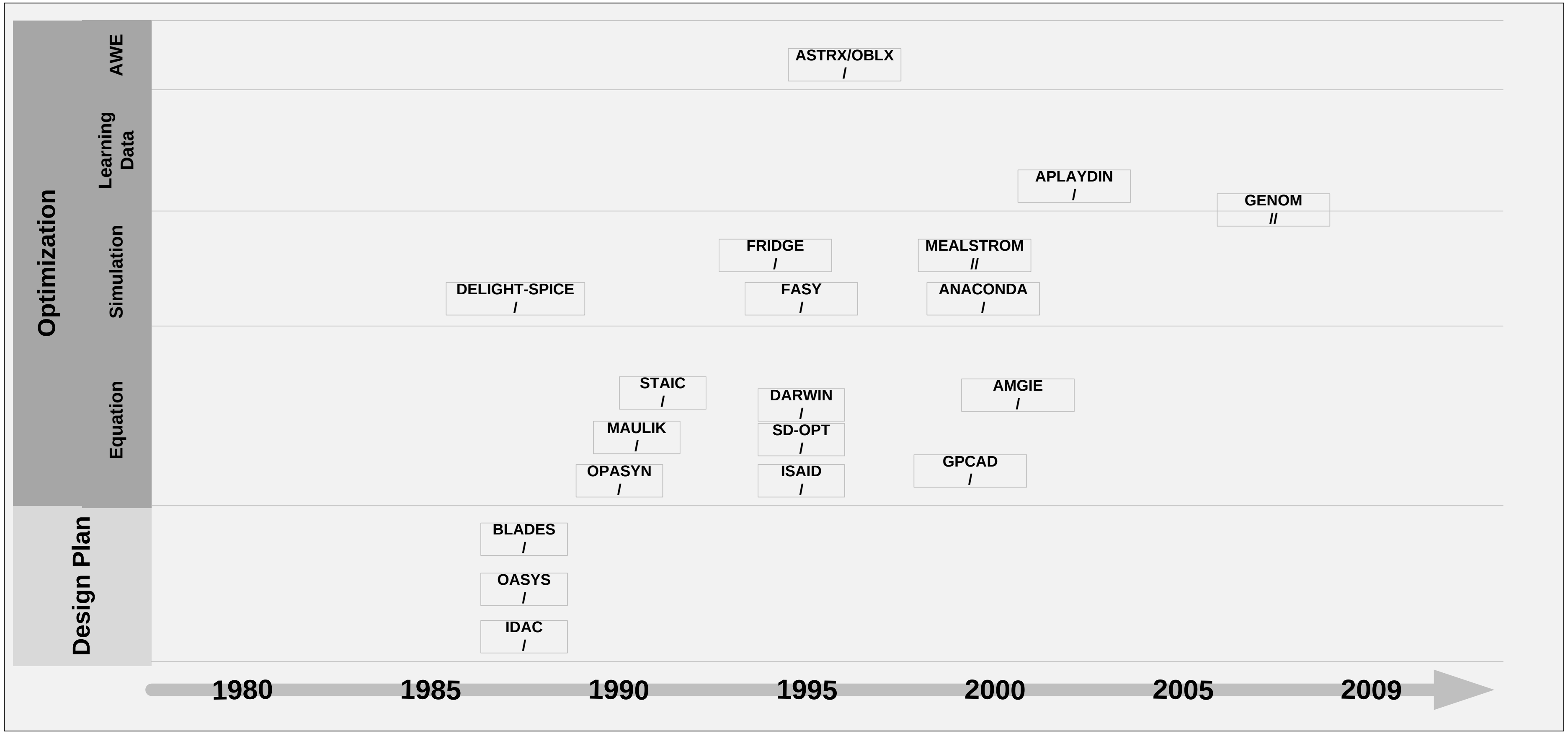


Figure 2-7 Classification *versus* date of analog sizing implementations

Knowledge-based methods were applied on early developments of Computer-Aided Design (CAD) tools to handle low level of abstraction and small circuits. Large systems large systems where decomposed on sub blocks to be handled. Although some automation was introduced with these tools, the setup time is prohibitive. Another disadvantage is the non-existence of a search engine to explore the design space.

On circuit sizing task, recent CAD implementations included an optimization kernel that creates a design loop to perform trade-offs and obtain the required circuit performance. This enhancement is more efficient technique to explore the design space, releasing the designer of a repetitive task. This loop incorporates a circuit performance evaluator based on circuit equations or electrical simulation.

2.2.2.1 Equation-based versus simulation-based

On equation-based methods, performance is evaluated using a set of analytical equations. In the first implementations, these "closed-form" equations were provided manually. Only simple and approximated equations were applied, which limited the accuracy. Later implementations included an automatically equation extractor that eliminated the errors on equation derivation, improved the setup time and increased evaluation accuracy. Automatic extractor still performs some approximation to keep the length of equations on a reasonable size. Once equations are established, the optimization run-time is very small.

Simulation-based methods incorporate an electrical simulator inside the optimization loop. Considering that the circuit performance can be measured with the electrical simulator, two problems overcome: a large range of design problems can be handled; and setup time is also shortened. Using an electrical simulator, as evaluator, means that high accurate models are used and, thus, the performance prediction is very good. This approach benefits from the fact that simulators use complex and accurate device models. Technology migration is as simple as changing the device models used by the simulator. The main drawback of using simulation-based methods is the execution time. Each optimization loop invokes the circuit simulator, and an optimization-based tool requires the evaluation of a large number of circuits.

2.2.2.2 Centralized versus distributed versus parallel processing

Figure 2-7 also shows distinction on tools in terms of computing type: centralized or distributed/parallel processing. Although centralized processing can be simple to implement, the distributed/parallel version improves tools performance in terms of processing time, since there are more processing units. Also, having more computer power, one can explore a wide range of design space variable values.

Another positive point about distributed processing is the use of idle processing time of the computers. Incorporating less powerful computer, on the distributed network, still raise the computation capacity.

2.3 Brief Considerations about Layout Automation

Although it is out of the scope of this thesis, it is important to provide some overview about the developments made in the physical device placement and routing.

As opposed to the digital domain, generally, the analog domain complex design requirements tend to difficult the development of automation tools. That is one of the reasons for the lack of automation tools and developments of EDA in the analog counterpart. For instance, in analog layout, one needs to consider device symmetries, different current densities in the wires, size and placement constraints for better performance and device matching.

The initial developments were simple computer editors that assist the humans to draw the physical masks, e.g. Magic[51]. Common structures started to be available as parameterized cells, pcell, which still are often used in, for instance, Cadence environment[52]. The engineer fills-in the size of the cell and the tools generate the layout for common structures, e.g. Hyper DevGen[53]. Besides trying to generate a complete circuit layout from the circuit schematic, only the devices structures were generated/parameterized, to assist the designer. The routing and placement were in total control of the engineer.

Actually, automation on layout only started with the procedural-based approaches [54][55]. These enabled the designers to code a parametric representation of the geometry of the circuit layout with the values resulting from the circuit sizing stage. Those were not flexible, not generic enough to, for instance, be reused on another topology. Moreover, it had a high implementation cost.

The next innovation was a template-based procedural approach[56]. With this approach, designers are able to code the circuit layout using predefined generic geometries, e.g. pcells. These pcells aggregated relative position of the cell elements and technology constrains. Some sort of backtracking information facilitated the sizing task, since the engineer could anticipate the type and/or shape the circuit elements, and how it would be later laid-out, e.g. considering parasitic at circuit sizing stage.

Next, optimization algorithm hinted the automatic layout generation. Receiving as input the circuit sizing results, tools were developed to automatically generate procedural layouts based on parameterized cells, e.g. [57][58]. These implementations search the best positioning for each customized cell, following the manufacturing process rules [59], e.g. minimum distance between two metal lines, and optimizing (reducing) the amount of area. These methods are more flexible and generic since are not tied to a specific topology and/or fabrication technology. The optimization algorithms most used in the layout automation include simulated-annealing[29] and genetic algorithms[30]. Hierarchical techniques were also applied to make the process more flexible, generic and faster [30].

Later, the layout tools also incorporated a feature to check the design rules on-the-fly[52] Direct-Rule-Check (DRC). This helps the designer to account the most common errors, and/or account the limitations of the fabrication process, while sketching the masks, manually.

Another improvement related with the circuit layout, are the layout *versus* schematic (LVS) tools. This class of tools compares the size and connections of circuit netlist, with the data extracted from the layout design. This prevents layout conception errors, which reduced the redesign cycles and cost of fabrication.

Nowadays, some effort focus on the perspective of tools integration with the major circuit design standard formats and databases, e.g. OpenAccess[60].

2.4 Open-Source Tools in Automation

Although the CAD/Electronic Design Automation (EDA) for automatic circuit design is a large community, it does not have the sufficient person-power to have a vast collection of open-source tools, as in, for instance, text processing, e.g. Open Office ®. Meanwhile, some implementations are starting to show up under the open-source license(s).

The most commonly used and freely available tools in the Very large Scale Integration (VLSI) design are: Electric, Magic, Alliance and gEDA.

The Electric VLSI Design System[61] is an open-source EDA system developed in the early 1980's, using the C programming language. Actually, it supported by Sun Microsystens Laboratories and it was ported to Java programming language, which provides more stability and platform independence.

Some of the tools that integrate the Electric system are designated below, to name a few only:

- Schematic capture, with textual languages, e.g. VHDL;
- Simulation ;
- Layout Generation;
- Design Rule Checking;
- Electrical Rules Checking;
- Network Consistency Checking (LVS)
- Printed Circuits Board

Magic is widely known as the 1st VLSI layout tool[51]. Source-code was written by John Ousterbhou in 1980's. The main advantage it was the open-source license that enabled the users to implement their own ideas, making it more advanced. It also comprises design-rule-check, hierarchical circuit extractor and routing features. The design style is based on Mead-Conway "scalable CMOS" which means it uses "lambda-based" dimensions. This allows Magic to generate different output files in order to implement the same design on different processes, and convert the lambda units to physical dimension at different scales.

Alliance is a free set of CAD tools[62] that have been developed by ASIM department of LIP6 laboratory of the Pierre and Marie Curie University (Paris VI, France), and it is mainly used for teaching VLSI design. It supports the standard VLSI description formats like SPICE, EDIF, VHDL, CIF and GDSII. Also, it supports both construction tools and validation tools. The design flow of Alliance is also based on Mead-Conway model, and is divided into five parts:

- Capture and simulation of the behavioral view;
- Capture and validation of the structural view;
- Physical design;
- Verification;
- Coverage evaluation

To support the design flow, every Alliance tool can easily interact with each other, but at the same time they can be used independently. Alliance has over 150 documented standard cells and six custom optimized generators.

The gEDA project[63] has produced and continues working on a full GPL'd suite and toolkit of Electronic Design Automation tools. These tools are used for electrical circuit design, schematic capture, simulation, prototyping, and production. Currently, the gEDA project offers a mature suite of free software applications for electronics design; including schematic capture, attribute management, bill-of-materials (BOM) generation, netlisting into over 20 netlist formats, analog and digital simulation, and printed circuit board (PCB) layout. The gEDA project was started because of the lack of free EDA tools for POSIX systems with the primary purpose of advancing the state of free hardware or open source hardware. The suite is mainly being developed on the GNU/Linux platform with some development effort going into making sure the tools run on other platforms as well.

Finally, it is worth to mention the open-source electrical simulator NGSPICE[12], which is also distributed with the gEDA package. In the present work, part of the source code was integrated in the developed platform, as an option for circuit performance evaluator engine.

2.5 Proposed Work

The proposed optimization methodology and platform developed is most suitable to contribute to innovate in the design automation of integrated circuit area. The main innovation implemented in this work is the proposed time-domain optimization methodology, described in chapter 3, verified with some practical examples given in chapter 5.

The developed platform is based on the optimization approach, e.g. GA, that is able to improve the performance of the existing topologies (or new ones), even when the fabrication technology is reaching the integration limit. The focus are the multi-stage amplifier topologies that are, probably, the most difficult analog circuit building blocks to design.

The time-to-market and the cost-reduction are also addressed with the incorporation of accurate elements models, e.g. BSIM3, and complete evaluation processes, e.g. time-domain analysis, which produces results compatible with the verification standards used in the industry. Moreover, process, voltage and temperature (PVT) variations are also taken into account during the optimization task to improve the robustness of the resulting circuit instance.

In terms of processing time, the exploration of distribute/parallel processing, e.g. Message Passing Interface (MPI), also proves to be a good performance increment. Furthermore, it uses the idle time of the workstations, e.g. personal computers, to gather more processing capacity.

3 Time-Domain Optimization Methodology

Optimization, in general, is a difficult task. Optimizing complex analog circuit blocks (i.e. amplifier or even analog-to-digital converters) can be a particularly difficult task. The methodology presented in this chapter as well as the software platform (described in the next chapter) are aiming to alleviate this problem. Moreover, the examples presented in chapter 5 are based on the optimization of the design of CMOS amplifiers, with complex topologies configuration and accurate high-order device models.

The evolution of the CMOS technology leads to smaller geometries and channel lengths, which requires a higher-order of complexity in the transistor models to be considered for accuracy. Moreover, the trend to incorporate complete systems into battery-powered portable equipment and the requirement of low power dissipation are driving the circuits, and particularly amplifiers, to operate at reduced supply voltages (1.2 V or less). That in turn, means loss of dynamic range, which imposes the use of rail-to-rail output stages in the amplifiers and reduce the number of stacked devices [64].

On other hand, market demands high-performance amplifiers/circuits (high low-frequency gain, high-frequency closed-loop poles and very fast settling response), which require the use of highly complex amplifier topologies and improved circuit techniques, which lead to complex design procedures, e.g. to deal with transfer functions with multiple poles and zeros.

A typical CMOS two-stage amplifier topology is depicted in Figure 3-1. It comprises a cascode input stage for high DC gain; a differential common-mode source output stage for superior dynamic range and hybrid cascade compensation for improved bandwidth. It is a complex circuit to design, since it is equivalent to a fourth-order system, assuming that proper compensation is used. To increase the power supply rejection ratio (PSRR), noise and bandwidth performance, alternatively to the traditional, a cascade-Miller compensation was proposed in [65], which consists of applying the compensation capacitor C_A, between a low impedance input-stage node and the amplifier output. In [66], an improved schema is discussed which can be achieved by only using capacitor C_B. This technique reaches the same compensation effects, while using lower power dissipation due to the fact that for a given transconductance a NMOS transistor needs smaller bias current than a PMOS transistor. A hybrid combination of the previous mentioned compensation techniques is proposed in [67]. This is obtained when C_A and C_B are used simultaneously and has the main advantage of increasing the amplifier unity-gain bandwidth when compared with other cascode-compensation schema. However, in this analysis [67] the system had to be reduced to 3rd order, by considering $C_A = C_B$, which could limit the scope of the proposed solution. Let's assume this amplifier as a working example.

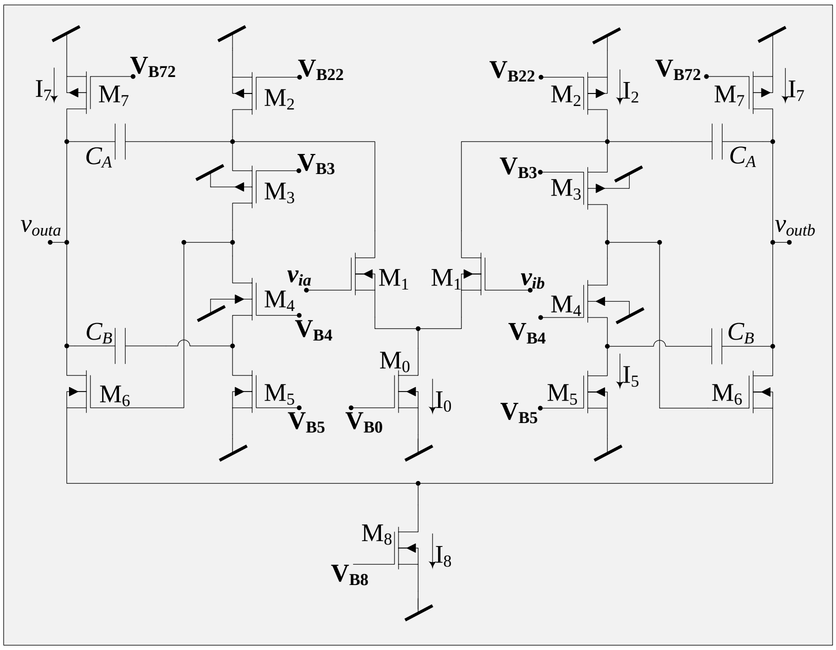


Figure 3-1 Low-voltage two-stage cascode-compensated amplifier (biasing and CMFB circuitry not shown)

As stated before, the optimization procedure can be decomposed into two main functions: search; and evaluation. The search consists of looking for, within the design space, the optimal device size and values that will meet the initial performance specifications. On other hand, the evaluation relies on the computation of the circuit performance parameters by solving a set of equations or directly simulating the circuit. Generally, the search step involves multiple circuit evaluation, which is not only a complex task but also computing and time consuming. The circuit equations, mentioned, are computed as function of the elements of the linear device model of the MOS transistor, as described in section 3.4. The most frequently used circuit performance parameters for amplifiers are (but not limited to):

- Gain, A_{OL};
- Gain-bandwith product, GBW;
- Output voltage swing, OS;
- Slew-rate, SR;
- Power Supply Rejection Ratio, PSRR;
- Common-mode Rejection Ratio, CMRR;
- Noise;
- Power dissipation;
- Die area.

These parameters are described in section 3.5.

Although some of these parameters may be calculated explicitly, some parameters can not be calculated in an easy way, typically, resulting in an unconstrained problem with too much degrees of freedom [68]. As described in chapter 2, computing aided design optimization approaches, implicitly, solve these degrees of freedom, while optimizing the performance of the circuit under the given specification constraints.

A time-domain optimization methodology can significantly simplify the calculus for circuit optimization of superior order topologies. The main advantage of this time-domain optimization is that, besides power dissipation and die area, the only main specification to consider is the settling-time for a given settling accuracy. Moreover, when a given settling-error is reached within a desired settling-time, it is automatically guaranteed that the amplifier has enough open-loop gain, A_{OL}, output-swing, OS, slew-rate, SR, closed loop bandwidth and closed loop stability. For example, in switched-capacitor circuits the objective is to have a stable amplifier with a given settling error, after a given available time. By analyzing the step response of the amplifier it is possible to obtain a single key performance indicator (KPI) that encloses all the traditional indicators, such as DC gain, GBW and phase margin (PM). Following this approach, the amplifier design can be accepted just by checking if the settling error is smaller than the desired value and that the closed-loop step response is stable.

Next, the main steps of the proposed optimization methodology will be described.

3.1 The Main Steps of the Proposed Optimization Methodology

The proposed methodology may be divided in three stages: preparation stage; integration stage; and optimization stage (and results). The Figure 3-2 illustrates these three stages.

The first stage: a preparatory work consists of the circuit knowledge extraction and the build of the closed-loop step-response equation of the circuit. It requires a circuit description in the format of a SPICE-like netlist. The second stage consists in the time-domain source-code integration into the optimization platform, and the optimization setup (i.e. circuit performance parameters definition). The third stage is the circuit sizing as described in chapter 4. At the end, the result is exported, as a circuit netlist, with the optimum size of the transistors as well as the value of the other devices.

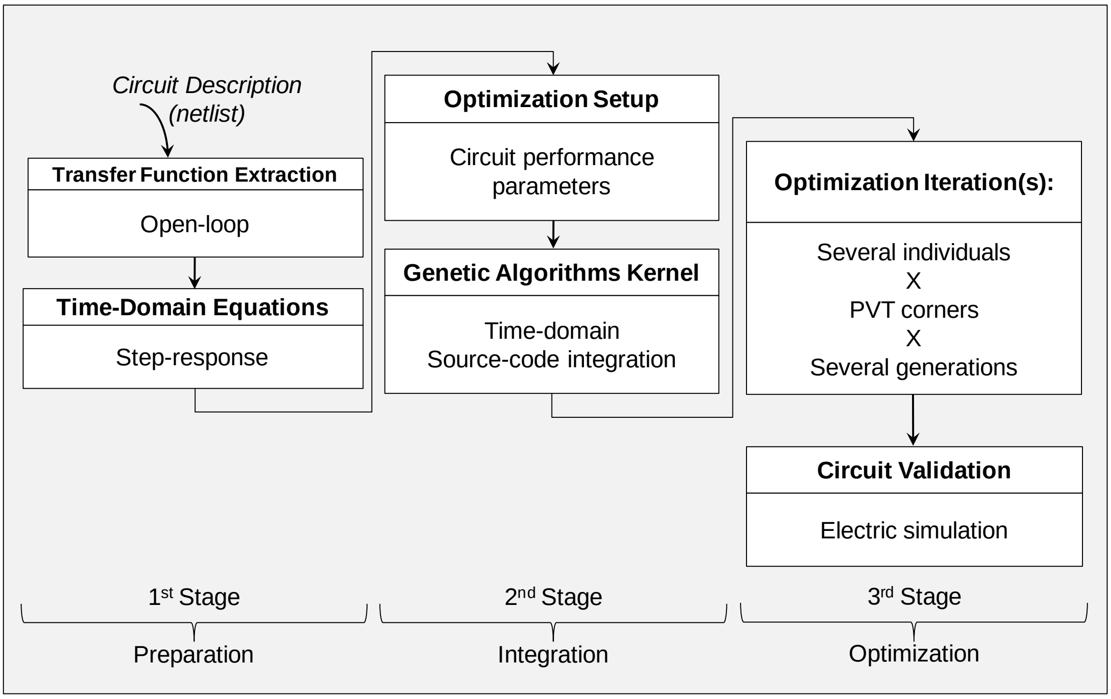


Figure 3-2 Steps of the proposed time-domain optimization methodology

The transfer function extraction is carried by an external software tool, developed by other authors [69], in the same research group of the author. It uses the procedure described in section 3.3 to compute the symbolic open-loop transfer function, $H_{OL}(s)$, of the circuit. While extracting the transfer function, other performance parameters are also defined, based on the circuit topology. After, the circuit response formula is built according to the procedure described in the next section.

3.2 Time-Domain Step-Response

The steps to built the time-domain step response, *h(t)*, of the circuit are depicted in Figure 3-3.

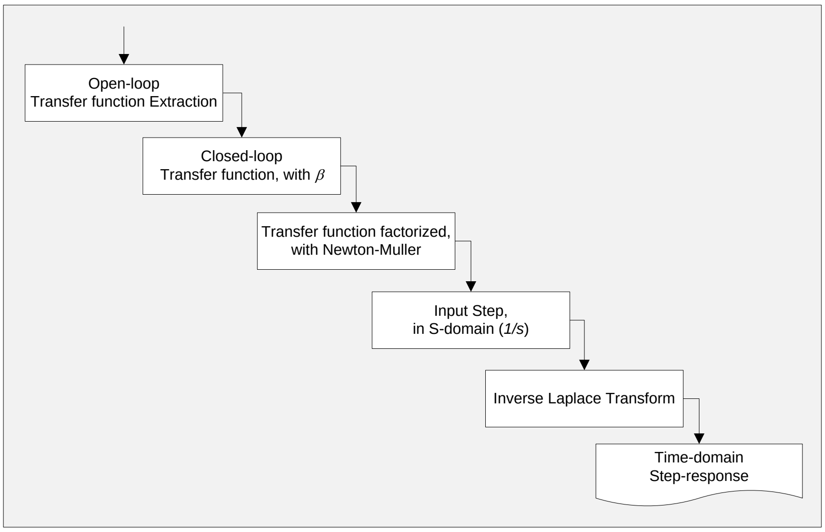

Open-loop
Transfer function Extraction
Closed-loop
Transfer function, with β
Transfer function factorized,
with Newton-Muller
Input Step,
in S-domain (*1/s*)
Inverse Laplace Transform
Time-domain
Step-response

Figure 3-3 Flow of the extraction of the time-domain step-response

Using symbolic analysis and calculus, the definition of the *h(t)* starts with the extraction of the open-loop transfer function, $H_{OL}(s)$, preferably without any order reduction/simplification. Using the behavioral-signal path (BSP) method described in[70], the circuit open-loop transfer function $H_{OL}(s)$ can be obtained in the form:

$$H_{OL}(s) = \frac{N_{OL}(s)}{D_{OL}(s)} \tag{3.1}$$

Then, the closed-loop transfer function of the amplifier, $H_{CL}(s)$, is computed for the desired feedback factor, β (since the amplifier is supposed to be embedded in a certain application, normally, in a closed-loop configuration).

$$H_{CL}(s) = \frac{H_{OL}(s)}{1+\beta \cdot H_{OL}(s)} = \frac{N_{OL}(s)}{D_{OL}(s)+\beta \cdot N_{OL}(s)} \tag{3.2}$$

Independently of the number of poles an zeros, $H_{CL}(s)$ can be always numerically factorized into *nz* complex zeros (in the left half-plane (LHP) and right half-plane (RHP) of the complex plane) and *np* complex poles (with $np \geq nz$) and written in the form

$$H_{CL}(s) = \frac{(s-z_1)\cdot(s-z_2)\cdot\ldots\cdot(s-z_{nz-1})\cdot(s-z_{nz})}{(s-p_1)\cdot(s-p_2)\cdot\ldots\cdot(s-p_{np-1})\cdot(s-p_{np})} \tag{3.3}$$

Equation (3.3) is computed using the Newton-Muller method and the DC bias operating point of the circuit. Primary, the DC bias operating point values of the devices components (e.g. *gm*, g_{ds}, etc.), are used to compute the numerical values of the coefficients of the transfer function. Afterwards, using the Newton-Muller method one have (3.3). Symbolically, a unity-step, in the *s*-domain, is, previously, applied to the transfer function, multiplying it by *1/s*. Finally, the closed-loop time-domain step-response, *h(t)*, is obtained using the Inverse-Laplace Transform, L^{-1}, according to

$$h(t) = L^{-1}\left(\frac{H_{CL}(s)}{s}\right) = \sum_{i=1}^{np} k_i \cdot e^{p_i \cdot t} + k_{pz} \tag{3.4}$$

where k_i and k_{pz} are constants dependent on the numerical values of the poles and zeros, defined by

$$k_i = \frac{\prod_{j=1}^{nz}(p_i - z_j)}{p_i \cdot \prod_{j=1 \wedge j\neq i}^{np}(p_i - p_j)}; \qquad k_{pz} = \frac{\prod_{n=1}^{nz}(z_n)}{\prod_{m=1}^{np}(p_m)} \cdot (-1)^{(np-nz)} \tag{3.5}$$

as previously mentioned, z_i and p_i are the complex roots of $H_{CL}(s)$.

3.3 Circuit Behavioral Signal Path Analysis

Circuit modeling is a paramount task in circuit design. It provides insight of circuit operation, which is useful for design, redesign and technology migration.

Different modeling methodologies exist. Traditional techniques like modified nodal analysis (MNA) [71] create an exact model of the circuit but they do not provide physical insight of the device parameters. Symbolic simulation [72] gives an approximate transfer function and it provides additional qualitative insight. BSP technique provides separated contributions of small-signal device parameters, to the transfer function [70].

Considering only one half of the circuit (due to the differential nature, it is only necessary to analyze one half) of the amplifier shown in Figure 3-1, shown in Figure 3-4,

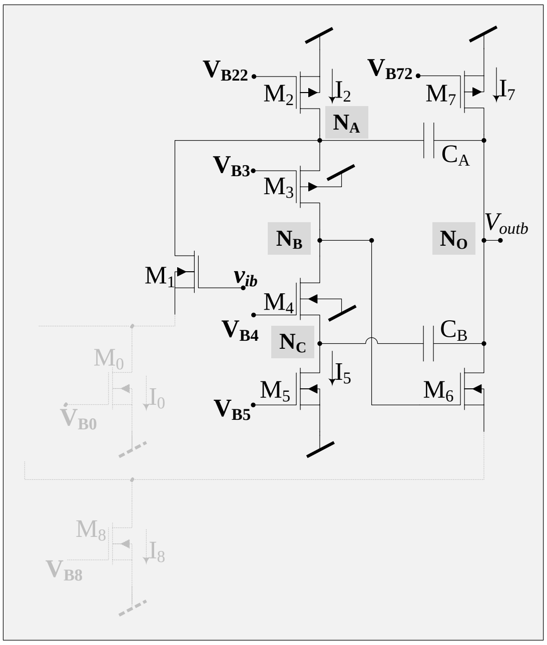


Figure 3-4 Half of the circuit amplifier shown in Figure 3-1.

Figure 3-5 depicts an example of signal flow in a system with different poles and zeros, and several feedbacks and feed-forward paths. It shows which poles/zeros cause a decrease/increase of the transfer function, independently of the numerical value. Also, the poles and zeros are a function of the small-signal device parameters, and are described with compact symbolic equations. This offers the possibility of control the correct placement of poles and zeros, in manual design or automated optimization. Since these models are based on the values of the operating bias point, the circuit representation can track the DC bias operating point variations sourced by the optimization kernel. This methodology also offers some degree of abstraction, for instance:

- isolate the effects between different nodes by explicit components, see section 3.4.6;
- replace dynamic cascade loads with a equivalent output impedance;
- lump series and parallel of components, which reduces the number of expression terms and, consequently, the execution time;
- ground bias nodes, e.g. tail node of a differential pair, and avoid bias transistors, which only affect the common-mode behavior.

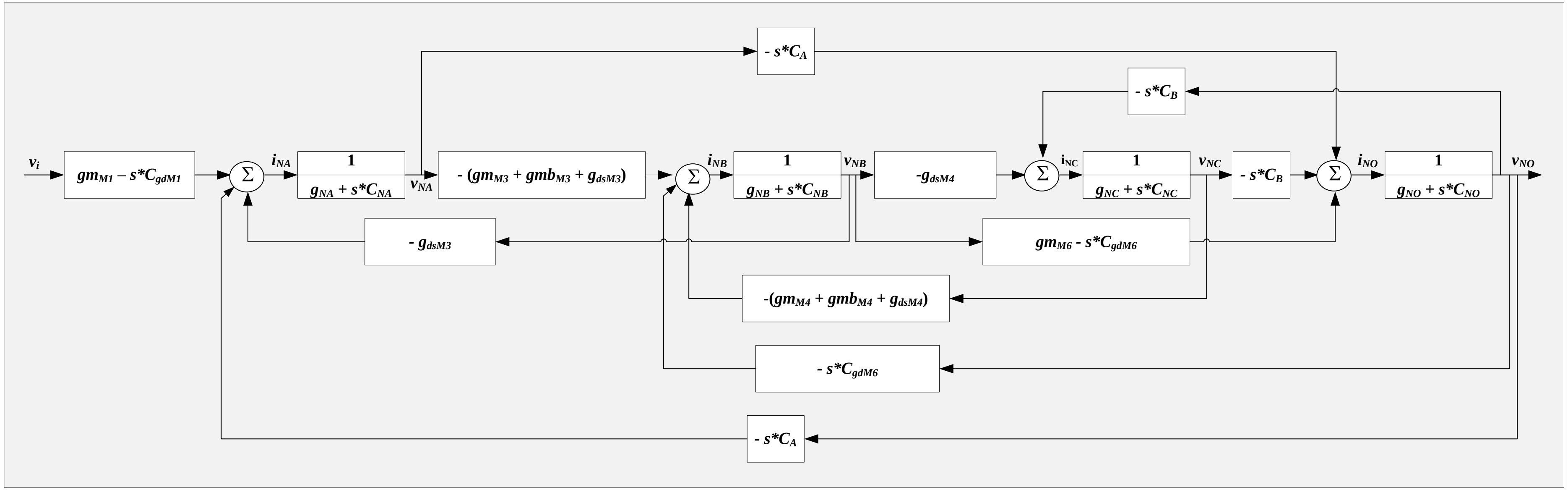


Figure 3-5 An example of the BSP of half of the circuit amplifier shown in Figure 3-1.

$$g_{NA} = g_{dsM1} + g_{dsM2} + g_{dsM3} + gm_{M3} + gmb_{M3} \quad (3.6)$$

$$C_{NA} = C_{dbM1} + C_{dbM2} + C_{gdM2} + C_{gsM3} + C_{sbM3} + C_{gdM1} + C_{gsM1} + C_A \quad (3.7)$$

$$g_{NB} = g_{dsM3} + g_{dsM4} \quad (3.8)$$

$$C_{NB} = C_{dbM3} + C_{gdM3} + C_{dbM4} + C_{gdM4} + C_{gsM6} + C_{gdM6} \quad (3.9)$$

$$g_{NC} = g_{dsM5} + gm_{M4} + gmb_{M4} \quad (3.10)$$

$$C_{NC} = C_{gsM4} + C_{sbM4} + C_{gdM5} + C_{dbM5} + C_B \quad (3.11)$$

$$g_{NO} = g_{dsM6} + g_{dsM7} \quad (3.12)$$

$$C_{NO} = C_{dbM6} + C_{gdM6} + C_{dbM7} + C_{gdM7} + C_B + C_A + C_L \quad (3.13)$$

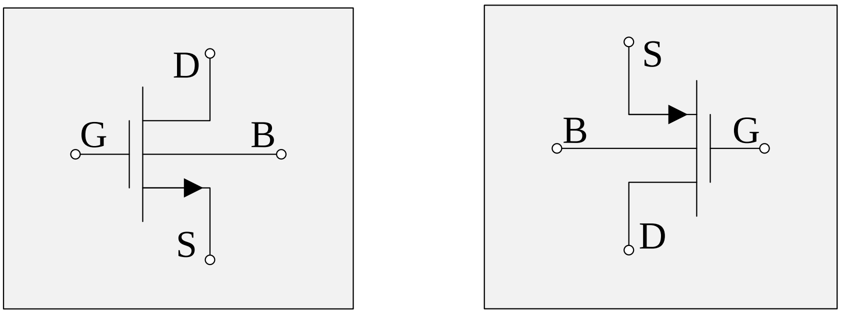
These abstractions produce expressions more compact and, thus, more interpretable.

The BSP provides insight knowledge of the circuit behavior and is a powerful tool to understand the effects of each component on the circuit performance. However, it should not be used to size and/or optimize the circuit because the components of the linearized circuit are not independent of each other, e.g. changing a transconductance, *gm*, also alters the parasitic capacitances.

Next, a brief description of the MOS transistor: modeling (level 2 equations) and linearization (small signal model) is given.

3.4 Basic Equations of the MOS Transistors

The behavior of the field-effect transistor (FET) is described by his name since the degree of the cut-off region or of the conduction region is defined by the existing electric field. There are several types of FETs but the Metal-Oxide-Semiconductor FET transistor (MOSFET or simply MOS) is, by far, the most used device since the fabrication process is relatively inexpensive, which rapidly captured both, the analogue and the digital markets. In this section, the modes of operation of a MOSFET at an elementary level and the small-signal equivalent model are presented.



a)

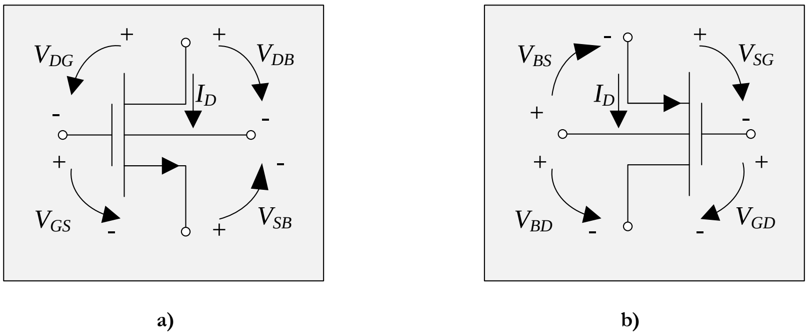
b)

Figure 3-6 Symbols of MOS transistors: a) NMOS; b) PMOS

The Figure 3-6 presents the typical 4-terminal symbols of NMOS and PMOS transistors[73]. These devices are assumed to have seven operating regions: cut-off and weak inversion; moderate and strong inversion; linear and triode; and saturation (active region). These regions are characterized by the bias voltages applied at the terminals.

3.4.1 Large-signal equivalent model of MOS transistors

The symbols and conventions used in the large-signal equivalent model equations of MOS transistors are depicted in Figure 3-7.



a)

b)

Figure 3-7 Symbols and conventions used in the large-signal model equations of MOS transistors: a) NMOS; b) PMOS

The regions of operation are described with respect to an NMOS transistor, initially, with source and bulk terminals connected to the ground. The generalization for PMOS devices is straightforward since the same equations can be applied. Considering the Figure 3-7, note that, for PMOS devices a negative sign is applied to every voltage variable. Thus, V_{GS} becomes V_{SG} and, V_{DB} becomes V_{BD}. The threshold voltage, V_{TN} (NMOS), also becomes -V_{TP} (PMOS) where V_{TP} is now a negative quantity and slightly higher than V_{TN} (in absolute value). Hence, a PMOS transistor is, for example, in the active region if $V_{SD}>(V_{SG}-|V_{TP}|)=(V_{SG}-V_{TP})$.

3.4.1.1 Cut-off and weak-inversion regions

For gate voltages smaller than threshold voltage, V_T (V_{TN} and V_{TP}, respectively for NMOS and PMOS transistors), the drain and the bulk form a reversed biased p-n junction and the transistor is in the cut-off region where:

$$\begin{cases} I_D = 0, \qquad if \qquad V_{GS} < V_{TN} \\ \qquad\qquad\qquad\qquad V_S = V_B = 0 \end{cases} \tag{3.14}$$

For gate voltages around V_{TN}, the positive carriers in the channel under the gate are initially repulsed and the channel starts to change from a *p* region into an *n* region, i.e. the channel is being inverted. The exact gate-source voltage, V_{GS}, for which the concentration of electrons under the gate is equal to the concentration of holes in the p-substrate is usually referred to as the transistor threshold voltage, V_{TN}. For small positive gate voltages (smaller than V_{TN}, i.e., V_T for the case of the NMOS device), very small amounts of current can flow. The transistor is said to be in weak-inversion. For the sake of simplicity we consider here that the transistor remains with a drain current, $I_D \approx 0$ but, in fact, the transistor behaves like a slow bipolar transistor with $I_D \approx I_S \cdot e^{\frac{V_{GS}}{V_T}}$, where I_S and V_T represent, respectively, the limit weak inversion current (proportional to W/L) and the thermal voltage (kT/q). This region of operation is out of the scope of this work but it has many interesting low-frequency applications, namely, biomedical circuits (e. g. implanted pacemakers and hearing heads), electronic watches, etc [74].

3.4.1.2 Moderate and strong inversion regions

As V_{GS} increases, the drain-to-source current, I_D, becomes more significant. Although an inversion coefficient can be defined to characterize the level of inversion [75], it can be approximately defined by the gate-to-source voltage. The lower end of the weak inversion region is the subthreshold region that exists for values of V_{GS} less than V_{TN} when positive drain current flows. As ranges from subthreshold values up to about 20 mV above V_{TN}, the device is in the weak inversion region. From an empirical knowledge, one can say that starting with a value of 20 mV above V_{TN} to a V_{GS} of approximately 220 mV the device operates in the moderate inversion region [76]. Above this value of V_{GS}, the device is considered in the strong inversion region. The strong inversion region was perhaps the most commonly used among the three regions but, in deep-submicron CMOS technologies with reduced supply voltages, the moderate inversion region has become the dominant one, since the efficiency gm/I_D is maximized and also because the device modeling have improved substantially with BSIM3 and BSIM4 models.

Due to the high-speed constraints, in the examples presented in chapter 5, the MOS transistors are all sized in these operating regions (either in moderate or in strong inversion).

3.4.1.3 Linear and triode regions

When the gate-source voltage, V_{GS}, is larger than V_{TN}, the channel is created. The drain current becomes positive and proportional to (V_{GS} - V_{TN}) as long as the drain-source voltage, V_{DS} is positive but relatively small. This region is called the linear region and the transistor behaves like a resistor (since I_D is proportional also to V_{DS} and R can be defined by V_{DS}/I_D) according to:

$$\begin{cases} I_D = K_N \cdot \dfrac{W}{L} \cdot \left(V_{GS} - V_{TN}\right) \cdot V_{DS}, & if \quad V_{GS} > V_{TN} \\ & \quad\quad V_{DS} \approx 0V \end{cases} \tag{3.15}$$

where K_N is technology-dependent constant defined by the product between the gate capacitance per unit area and the mobility of the electrons near the silicon surface. For PMOS devices a similar constant exists, K_P. However, since the mobility of holes is about 1/3 to 1/4 the mobility of the electrons, K_P is usually about 1/3 to 1/4 of K_N and, for this reason, PMOS are typically three to four times "slower" than NMOS transistors. The amount (V_{GS} - V_{TN}) is often called the *effective gate-source voltage* or, simply, the *overdrive-voltage*.

For larger drain-to-source voltages but smaller than the overdrive-voltage the potential of the channel is increased and the expression that defines the drain-current becomes more complex since the transistor enters in the non-linear triode region yielding

$$\begin{cases} I_D = K_N \cdot \dfrac{W}{L} \cdot \left[\left(V_{GS} - V_{TN}\right) \cdot V_{DS} - \dfrac{V_{DS}^2}{2} \right], & if \quad V_{GS} > V_{TN} \\ & 0 < V_{DS} < \left(V_{GS} - V_{TN}\right) \end{cases} \tag{3.16}$$

Usually, when accuracy is not that important, is very common to use a first order approximation of (3.16). The term ${V_{DS}}^2/2$ vanishes and (3.16) is reduced to the form (3.15).

3.4.1.4 Saturation (active) region

As the drain-source voltage, V_{DS}, is increased the channel becomes smaller close to the drain region. The electrons travelling through the drain region are velocity saturated and the drain current no longer increases with increasing V_{DS}. Thus, at the drain end, the channel becomes asymmetrical and pinched-off near the drain terminal, as illustrated in Figure 3-8.

A transistor is biased in the saturation region when its drain-source voltage is larger than its overdrive-voltage, *i.e.* $V_{DS} > \left(V_{GS} - V_{Tn}\right)$. For this reason, the amount $\left(V_{GS} - V_{TN}\right)$ is also called the drain-source saturation voltage, V_{dsat}. For analogy with bipolar transistors the MOS saturation region is often also known as active region. In this region, the drain current becomes independent of V_{DS} as follows

$$\begin{cases} I_D = \frac{K_N}{2} \cdot \frac{W}{L} \cdot \left(V_{GS} - V_{TN}\right)^2, \qquad if \qquad V_{GS} > V_{TN} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad V_{DS} > \left(V_{GS} - V_{TN}\right) \end{cases} \tag{3.17}$$

Since I_D is independent (in the first order of approximation) of V_{DS} this region is of great importance to design analogue amplifiers, in which the transistors are traditionally biased in the active region. As it will be shown in the last practical example, described in chapter 5, sometimes, after optimization over the fabrication process, voltage and temperature (PVT) variation, some devices might be biased in the boundary of the triode/active regions.

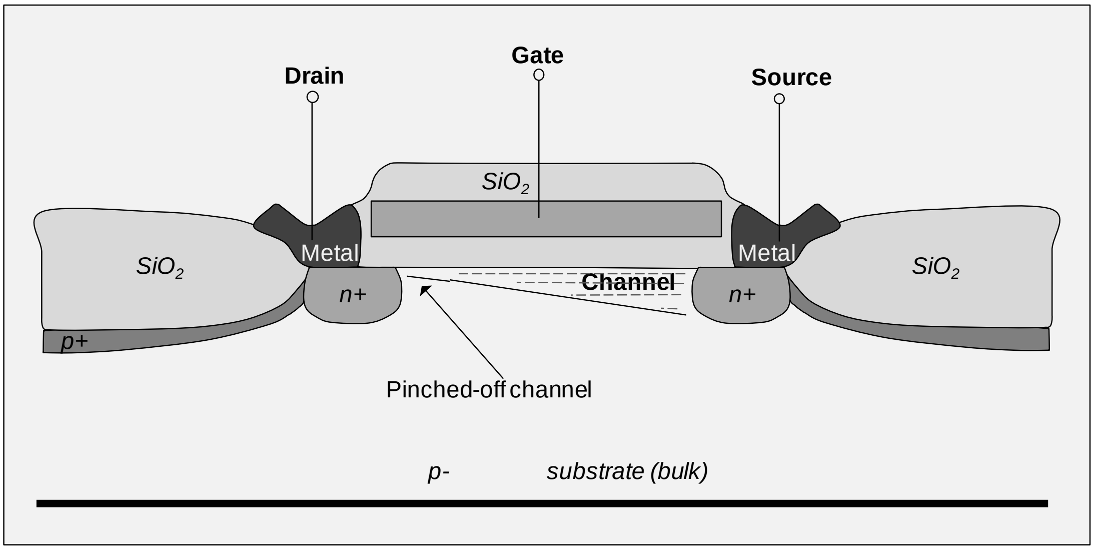


Figure 3-8 Cross-section of an NMOS transistor in the active region (saturation)

3.4.1.5 Channel modulation and short-channel effects

As just mentioned, as it appears in (3.17), in saturation the drain current is independent of the drain-source voltage. However, as V_{DS} increases, the channel length decreases, the drain current, I_D, is increased. This second-order effect corresponds to an effective shift of the pinch-off point and it is commonly referred to as channel-length modulation. Thus, the drain current becomes dependent of V_{DS}, and (3.17) results in

$$I_D = \frac{K_N}{2} \cdot \frac{W}{L} \cdot \left(V_{GS} - V_{TN}\right)^2 \cdot \left(1 + \lambda \cdot V_{DS}\right), \qquad V_{GS} \geq V_{TN} \tag{3.18}$$

where λ is the channel-modulation constant. When V_{DS} is large enough or when L is close to the technology minimum, second-order effects become relevant and the channel-length modulation effects become more critical. Figure 3-9 shows an I_D *versus* V_{DS}

characteristic of an NMOS transistor illustrating the channel-length modulation and the short-channel effects.

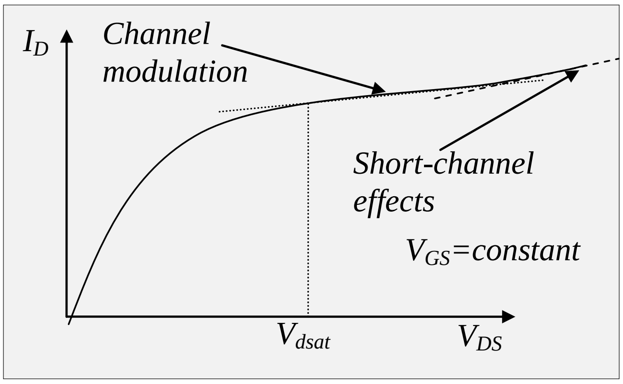


Figure 3-9 I_D *versus* V_{DS} characteristic of an NMOS transistor with channel-length modulation and with short-channel effects.

The concept used in high accuracy models, e.g. BSIM3v3 [5], is similar to the simplified (level 2) models shown here. However, the expressions for computing, for instance, the drain current of the transistor include many more effects. Thus, the expressions used within these high accurate models are not suited for hand calculations.

3.4.1.6 Body-effect

All derived equations assume that the source terminal (S) of an NMOS device is connected to its bulk (B) which, in turn, is connected to the most negative voltage of the circuit (*V_{SS}*). However, often the source and the substrate (bulk) can be at different voltage potentials. In this situation the threshold voltage, *V_{TN}*, increases when the reverse-bias source-bulk voltage, *V_{SB}*, increases. This effect is known as the body-effect. The dependence of *V_{TN}*, on the voltage *V_{SB}* can be represented in the following form:

$$V_{TN} = V_{TN0} + \gamma \cdot \left(\sqrt{2\phi_p + v_{SB}} - \sqrt{2\phi_p} \right) \tag{3.19}$$

where *V_{TN0}* is the threshold voltage for *$v_{SB}=0$*, *γ* is the body factor that depends upon the doping concentration in the channel region, *ϕ_F* is the Bulk Fermi-potential and *v_{SB}* is the source-to-bulk voltage[77].

Table 3-1 summarizes the large-signal, low-frequency, drain current of an NMOS transistor in the linear/triode and saturation regions of operation (whenever biased in either moderate or strong inversion).

3.4.2 I-V transistor characteristics

In moderate or strong inversion and, simultaneously, in active region the MOS transistors provide a drain current whose value is practically independent of the drain-source voltage, V_{DS}, and it is determined by the gate voltage according to the square-law relationship in (3.17), a sketch of which is shown in Figure 3-10 for an NMOS device.

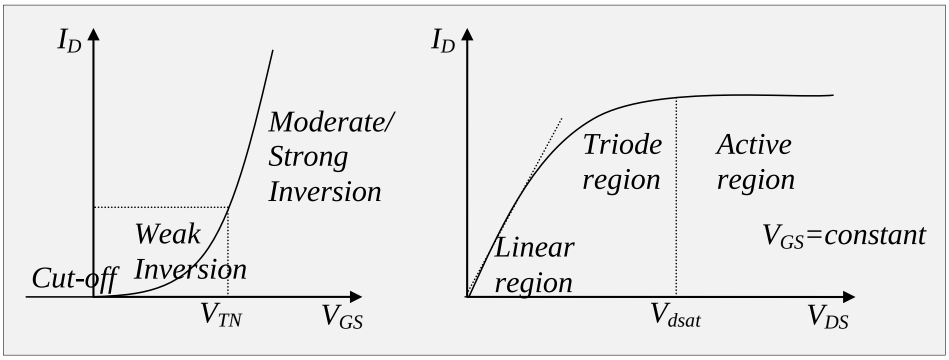


Figure 3-10 *I_D versus V_{GS}* and *I_D versus V_{DS}* characteristics of an NMOS transistor

Thus, the MOS behaves as an ideal current source whose value is controlled by V_{GS} according to a nonlinear relationship. For V_{GS} positive and smaller than the threshold voltage the device operates in weak-inversion the drain current rises exponentially with V_{GS}. However, since this current is of the order of a few tens of nA (nA = 10^{-9}A), one may assume $I_D \approx 0$. The I_D *versus* V_{DS} characteristic also shown in Figure 3-10 indicate that, for a given V_{GS} (constant) there are three distinct regions of operation: the linear region for very small values of V_{DS}, the triode region for $V_{DS} < V_{dsat}$ and the active region used whenever the MOS transistor may acts as a single device amplifier.

Table 3-1 Drain-current for MOSFET in large-signal and for low-frequency operation.

	NMOS	PMOS
Triode region:		
$\lvert V_{GS} \rvert > V_{T\{N,P\}}$ $\lvert V_{DS} \rvert < \left(\lvert V_{GS} \rvert - V_{T\{N,P\}} \right)$	$I_D = K_N \cdot \frac{W}{L} \cdot \left[\left(V_{GS} - V_{TN} \right) \cdot V_{DS} - \frac{V_{DS}^2}{2} \right]$	$-I_D = K_N \cdot \frac{W}{L} \cdot \left[\left(V_{GS} - V_{TN} \right) \cdot V_{DS} - \frac{V_{DS}^2}{2} \right]$
Saturation region:		
$\lvert V_{GS} \rvert > V_{T\{N,P\}}$ $\lvert V_{DS} \rvert > \left(\lvert V_{GS} \rvert - V_{T\{N,P\}} \right)$	$I_D = \frac{K_{\{N,P\}}}{2} \cdot \frac{W}{L} \cdot \left(V_{GS} - V_{T\{N,P\}} \right)^2 \cdot \left(1 + \lambda \cdot V_{DS} \right)$	$-I_D = \frac{K_{\{N,P\}}}{2} \cdot \frac{W}{L} \cdot \left(V_{GS} - V_{T\{N,P\}} \right)^2 \cdot \left(1 + \lambda \cdot V_{DS} \right)$

(devices assumed to be biased in moderate/strong inversion)

3.4.3 Low-frequency small-signal equivalent model

The equivalent model presented here is for small signals applied to the transistors in order to guarantee the DC bias operating point, usually, confined to the active region. Thus it is assumed that the drain current and the gate-source and the drain-source voltages have a DC component as well as small AC component defined as

$$\begin{aligned} i_D &= I_D + i_d \\ v_{GS} &= V_{GS} + v_{gs} \\ v_{DS} &= V_{DS} + v_{ds} \end{aligned} \tag{3.20}$$

If an NMOS transistor is operating in the active region and if we replace (3.20) in (3.18) yields

$$\begin{aligned} \left(I_D + i_d\right) &= \frac{K_N}{2}\cdot\frac{W}{L}\cdot\left(V_{GS} + v_{gs} - V_{TN}\right)^2\cdot\left(1+\lambda\cdot\left(V_{DS}+v_{ds}\right)\right) \\ &= \left(1+\lambda\cdot\left(V_{DS}+v_{ds}\right)\right)\cdot\left[\frac{K_N}{2}\cdot\frac{W}{L}\cdot\left(V_{GS}-V_{TN}\right)^2 + \right. \\ &\left. 2\cdot\frac{K_N}{2}\cdot\frac{W}{L}\cdot\left(V_{GS}-V_{TN}\right)\cdot v_{gs} + \frac{K_N}{2}\cdot\frac{W}{L}\cdot {v_{gs}}^2\right] \end{aligned} \tag{3.21}$$

Considering (3.18) one can assume that

$$\begin{aligned} i_D = \left(1+\lambda\cdot\left(V_{DS}+v_{ds}\right)\right)\cdot\left[2\cdot\frac{K_N}{2}\cdot\frac{W}{L}\cdot\left(V_{GS}-V_{TN}\right)\cdot v_{gs} + \right. \\ \left. \frac{K_N}{2}\cdot\frac{W}{L}\cdot {v_{gs}}^2\right] \end{aligned} \tag{3.22}$$

if $v_{gs} << \left(V_{GS} - V_{TN}\right)$ (small signal) the behavior is nearly linear (${v_{gs}}^2 \approx 0$) and a first order approximation can be done resulting in

$$i_D \cong \left(\frac{\partial i_d}{\partial v_{gs}}\right)\cdot v_{gs} + \left(\frac{\partial i_d}{\partial v_{ds}}\right)\cdot v_{ds} \tag{3.23}$$

The two most important small-signal parameters are the transconductance, *gm*, and the finite output conductance, g_{ds}, of the transistor defined as

$$gm = \left(\frac{\partial i_D}{\partial v_{GS}}\right) \tag{3.24}$$

$$g_{ds} = \left(\frac{\partial i_D}{\partial v_{DS}} \right) \tag{3.25}$$

Again, considering the active region:

$$\begin{aligned} gm &= K_N \cdot \frac{W}{L} \cdot \left(V_{GS} - V_{TN}\right) \cdot \left(1 + \lambda \cdot V_{DS}\right) = \\ &= \frac{2 \cdot I_D}{\left(V_{GS} - V_{TN}\right)} = \frac{2 \cdot I_D}{V_{dsat}} \end{aligned} \tag{3.26}$$

$$\begin{aligned} g_{ds} &= \frac{\lambda \cdot I_D}{1 + \lambda \cdot V_{DS}} \\ &\approx \lambda \cdot I_D \end{aligned} \tag{3.27}$$

For transistors where V_{SB} is nonzero, there will be an additional component of i_D, *gmb*.V_{SB}. The body-effect transconductance, *gmb* is computed by the following expression:

$$gmb = \frac{\partial i_D}{\partial v_{TN}} \frac{\partial v_{TN}}{\partial v_{SB}} \tag{3.28}$$

If considering the active region, it results in

$$gmb = -gm \frac{\gamma}{2\sqrt{2\phi_F + V_{SB}}} = \eta gm \tag{3.29}$$

where γ is the body factor that depends upon the doping concentration in the channel region, ϕ_F is the Bulk Fermi-potential and V_{SB} is the source-to-bulk voltage[77].

The most commonly used small-signal model for an NMOS transistor operating in the active region is then shown in Figure 3-11. Basically it comprises the voltage-controlled current source *gm*.v_{gs} and the finite output conductance g_{ds}. For the PMOS a similar model can be used.

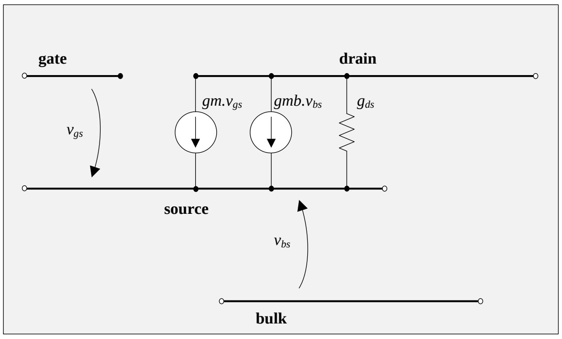


Figure 3-11 Low-frequency small-signal equivalent model of an NMOS transistor

3.4.4 Medium/high frequency small-signal equivalent model

In order to model the MOS operation at higher frequencies more accurately a number of parasitic capacitances are added to the low-frequency model, namely the following capacitances:

- gate-to-source, C_{gs}, is composed of two components: C_{gs0}, the gate-to-source overlap, and the gate-to-channel capacitance.
- gate-to-drain, C_{gd}, is due to the overlap of the gate and the drain diffusion. It is a thin-oxide capacitance, and hence, to a good approximation, it can be regarded as being voltage independent;
- source-to-bulk, C_{sb}, is also composed of two components: the *p-n* junction capacitance between the source terminal and the substrate (bulk), plus the active channel and bulk overlap;
- drain-to-bulk, C_{db}, is also composed of two components: the *p-n* junction capacitance between the drain terminal and the substrate (bulk), plus the active channel and bulk overlap;
- gate-to-bulk, C_{gb}, this parasitic capacitance exists between the gate and substrate overlap. In saturation, only the pinched-off region of the channel permits the gate and substrate overlap, which results in a small C_{gb}.

The resulting medium/high frequency AC model for a NMOS is displayed in Figure 3-12. Note that, for very high frequencies, other elements have to be considered in the AC model, namely, the non-zero distributed resistance of the polisilicon gate. However, this ultra-high frequency model is more useful for radio-frequency (RF) design

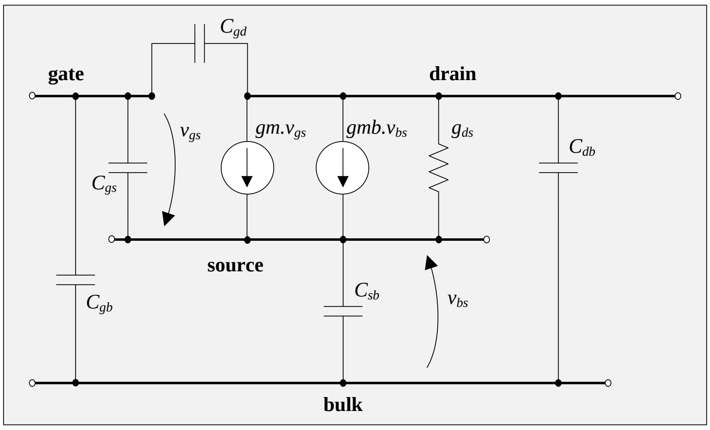


Figure 3-12 Medium/high frequency small-signal equivalent model of an NMOS transistor

Figure 3-13 show the cross-section of a NMOS transistor layout where the parasitic capacitances are represented.

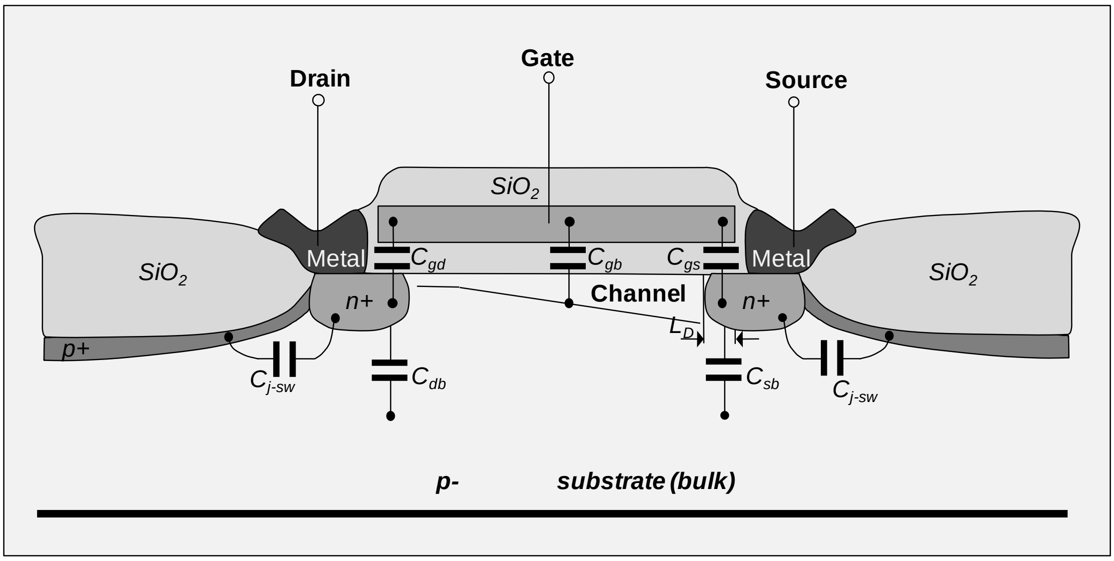


Figure 3-13 Cross-section of an NMOS transistor with the most relevant associated capacitances [78]

Due to fabrication process tolerances, the transistor dimensions that actually produce parasitic capacitances are different. The top-view of a transistor's layout shown in the Figure 3-14 identify the effective channel width, W_{eff}, and length, L_{eff}, the lateral diffusion length, L_D and the oxide encroachment width, W_{OV}, that makes reduce the effective channel width, W_{eff}.

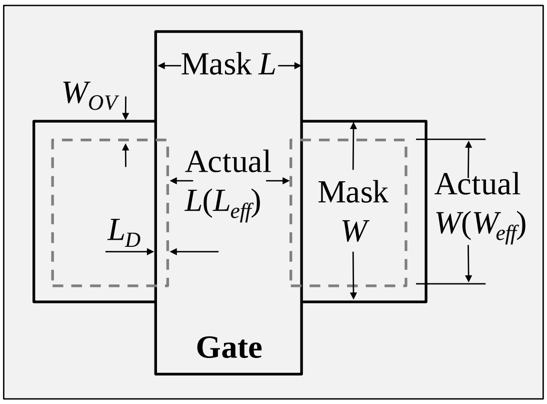


Figure 3-14 Top-view of an NMOS transistor layout masks [78]

The parasitic capacitance originated by the active channel is created by the overlap of the gate oxide and the active channel. This capacitance value varies, depending on the operation region of the transistor and it is defined by the effective size of the channel, through the following expression

$$C_g = C_{ox} \cdot W_{eff} \cdot L_{eff} \tag{3.30}$$

Depending on the operating region of the transistor, this capacitance is added to raise different parasitic capacitances: C_{gs}; C_{gd}; C_{gb}; as described next.

In cut-off region an active channel does not exit and the gate-to-drain and gate-to-source are only due to the overlap of the gate and the two terminals, as in

$$C_{gs0} = C_{gd0} = C_{ox} \cdot W_{eff} \cdot L_D \tag{3.31}$$

As the operation of the transistor enters in triode region, the channel exists uniformly from source to drain, and the gate-channel capacitance is divided in two equal parts at the drain and source, as defined by

$$C_{gs} = C_{gd} = C_{ox} \cdot W_{eff} \cdot \left(L_D + \frac{L_{eff}}{2} \right) = C_{gs0} + \frac{C_g}{2} \tag{3.32}$$

In the saturation, however, the channel pinches off at the drain side and the drain voltage exerts little influence on either the channel or the gate charge. As consequence the intrinsic portion of C_{gd} is essentially the overlap capacitance, given by

$$C_{gd} = C_{ox} \cdot W_{eff} \cdot L_D = C_{gd0} \tag{3.33}$$

and C_{gs} is [79]

$$C_{gs} = C_{ox} \cdot W_{eff} \cdot \left(L_D + \frac{2}{3} \cdot L_{eff} \right) = C_{gs0} + \frac{2}{3} \cdot C_g \tag{3.34}$$

Capacitance C_{gb} between gate and bulk models the parasitic oxide capacitance between gate-contact material and the substrate outside the active channel area. During device normal operation (saturation/triode/linear) this capacitance results from the gate-to-bulk overlap, excluding the active channel area, according to:

$$C_{gb0} = C_{ox} \cdot L_{eff} \cdot W_{OV} \tag{3.35}$$

In cut-off region, C_{gb} increases with the oxide parasitic capacitance of the channel area, as described by

$$C_{gb} = C_{ox} \cdot L_{eff} \cdot \left(W_{OV} + W_{eff} \right) = C_g + C_{gb0} \tag{3.36}$$

Figure 3-15 shows the distribution of the gate-associated parasitic capacitance, C_g, among the different parallel plate associated parasitic capacitances: C_{gs}; C_{gd}; C_{gb}; over the different operating regions of the transistor. In the graph shown in Figure 3-15, the variation of the operating region is represented by the variation of the gate-to-source voltage, v_{GS}, in the abscissa axis.

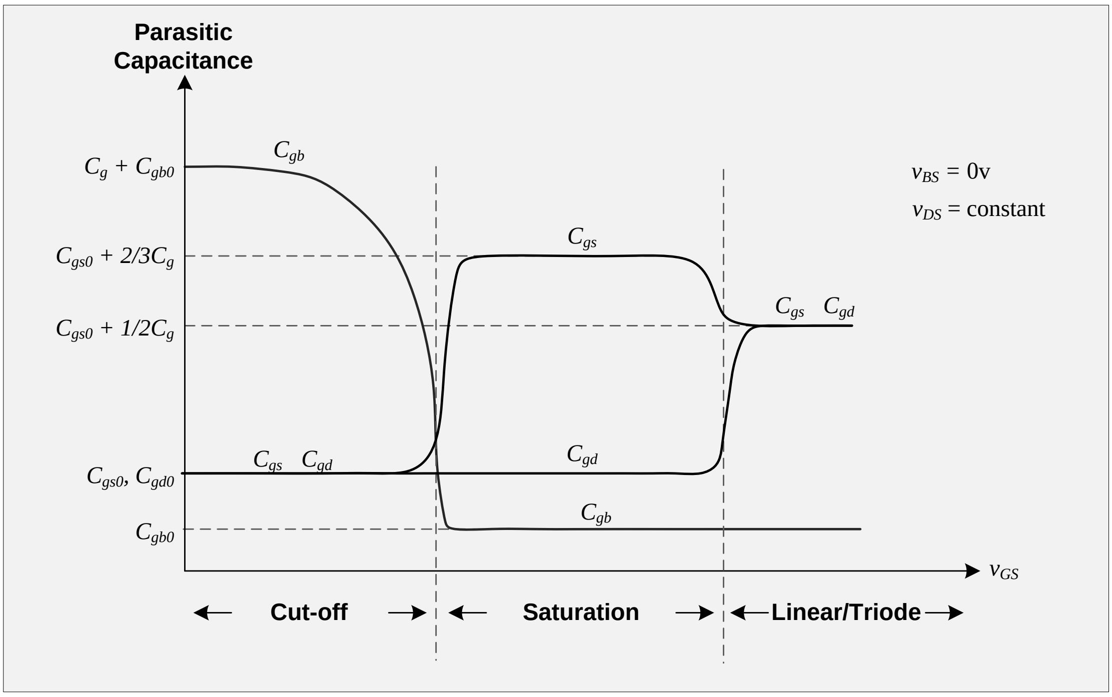


Figure 3-15 Parasitic capacitances *versus* gate-to-source voltage, v_{GS}[78]

The *p-n* junction parasitic capacitances are associated with the depletion region that results from the inverse voltage, $V_{junction}$, applied to the drain-to-bulk and source-to-bulk junctions. These *p-n* junction parasitic capacitances may be, further decomposed in two parts: bottom-plate, C_j, and side-wall, C_{j-sw}; given by

$$C_j = C_{j0} \cdot \left(1 + \frac{V_{junction}}{\psi_0}\right)^{mj} \tag{3.37}$$

$$C_{j-sw} = C_{j-sw0} \cdot \left(1 + \frac{V_{junction}}{\psi_0}\right)^{mj-sw} \tag{3.38}$$

where $V_{junction}$ is the voltage across the *p-n* junction, Ψ_0 represents the built-in potential of the junction, C_{j0} and C_{j-sw0} are the depletion capacitances *per* area and length unit, respectively, when the junction voltage is zero. Depending on the doping level of the *p-type* and *n-type* regions, the m_j and $m_{j\text{-}sw}$ represent the grading coefficient.

The parasitic capacitances of drain/source-to-bulk during different operating regions of the transistor are defined as follows:

Cut-off:

$$C_{db} = A_D \cdot C_{jd} + P_D \cdot C_{jd-sw} \tag{3.39}$$

$$C_{sb} = A_S \cdot C_{js} + P_S \cdot C_{js-sw} \tag{3.40}$$

where A_D and A_S represent the drain and source areas, respectively, and P_D and P_S are the drain and source perimeters, respectively.

Triode/Linear:

$$C_{db} = \left(A_D + \frac{A_{CH}}{2} \right) \cdot C_{jd} + \left(P_D - W_{eff} \right) \cdot C_{jd-sw} \tag{3.41}$$

$$C_{sb} = \left(A_S + \frac{A_{CH}}{2} \right) \cdot C_{js} + \left(P_S - W_{eff} \right) \cdot C_{js-sw} \tag{3.42}$$

where A_{CH} represents the active channel area.

Active/Saturation:

$$C_{db} = A_D \cdot C_{jd} + \left(P_D - W_{eff} \right) \cdot C_{jd-sw} \tag{3.43}$$

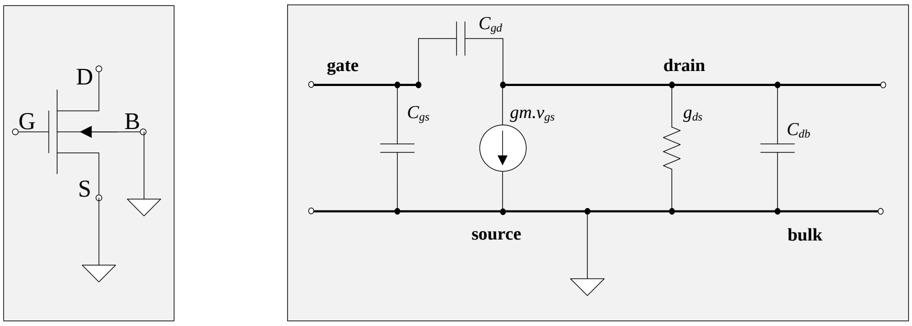
$$C_{sb} = \left(A_S + A_{CH} \right) \cdot C_{js} + \left(P_S - W_{eff} \right) \cdot C_{js-sw} \tag{3.44}$$

Table 3-2 summarizes the parasitic capacitances of an MOS transistor and their estimates values, in the three main regions of operation.

3.4.5 Linearization techniques for basic (single-device) MOS transistor circuits

Transistors are complex devices with four terminals and a non-linear behavior. This behavior is difficult to analyze, in analog design, especially during the circuit sizing task. The small signal modeling technique is an approximation to facilitate the evaluation of the transistor's behavior. It is used to translate the transistor's behavior into linear equations, which state the voltage and currents relations of the circuit nodes. Next, some of the basic configuration of the NMOS transistors and the correspondent graphical linear small signal equivalent model are presented. Similarly, it can be drawn the same small signal model equivalent for the PMOS devices.

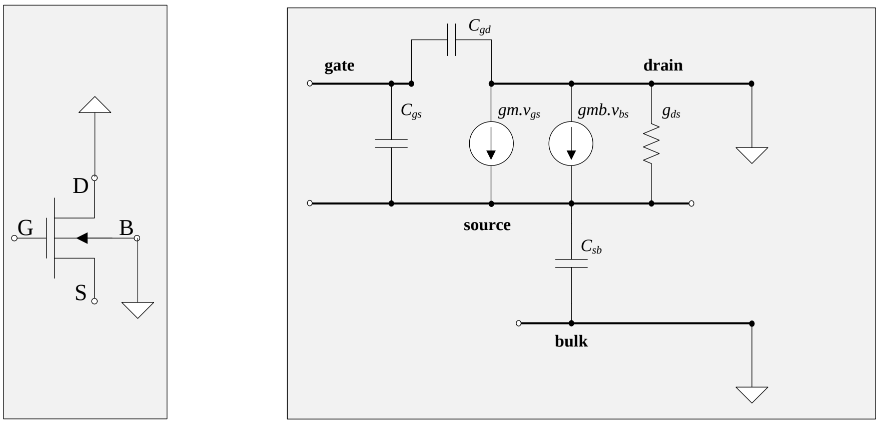
A transistor with the source connected to the same voltage as the bulk terminal, usually to the ground (NMOS), as depicted in Figure 3-16, it is designated as *common-source basic topology*. The small signal equivalent model consists of three capacitors, C_{gs}, C_{gd} and C_{db}, the transconductance, g_m, and the conductance, g_{ds}.



a) b)

Figure 3-16 Common-source transistor: a) Symbol; b) Small signal equivalent model

If the drain is connected to a DC voltage and the bulk terminal, normally to the ground (NMOS), as depicted in Figure 3-17, it is known as the *common-drain* basic topology. The behavior is governed by a set of four capacitors, C_{gs}, C_{gd}, C_{sb} and C_{db}, the transconductance, *gm*, the bulk transconductance, *gmb*, and the conductance, g_{ds}.



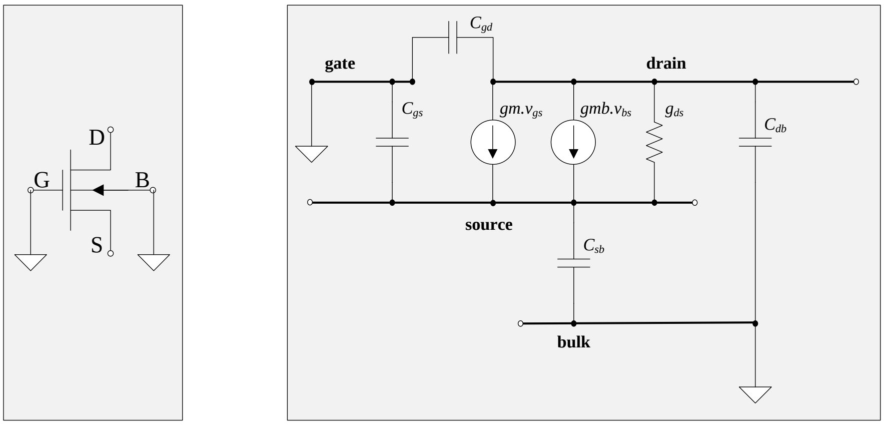
a) b)

Figure 3-17 Common-drain transistor: a) Symbol; b) Small signal equivalent model

With the gate connected to a constant DC voltage and the bulk terminal, typically to the ground (NMOS), as depicted in Figure 3-18, it is designated as the *common-gate* topology. The behavior is ruled by four capacitors, C_{gs}, C_{gd}, C_{sb} and C_{db}, the transconductance, *gm*, the bulk transconductance, *gmb*, and the conductance, g_{ds}.

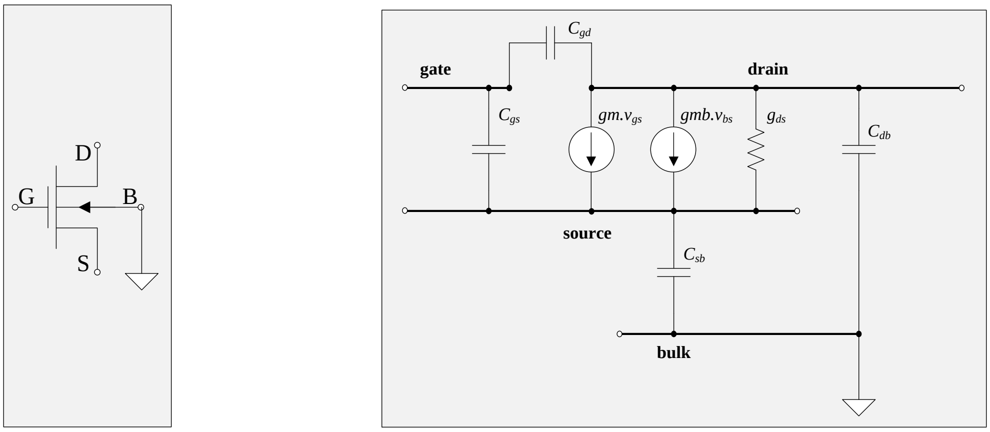
Table 3-2 Parasitic capacitances for MOS devices in the three main regions of operation.

Region of operation	**Capacitance**				
	$\boldsymbol{C_{gs}}$	$\boldsymbol{C_{gd}}$	$\boldsymbol{C_{gb}}$	$\boldsymbol{C_{sb}}$	$\boldsymbol{C_{db}}$
Cut-off	$C_{ox} \cdot W_{eff} \cdot L_D$	$C_{ox} \cdot W_{eff} \cdot L_D$	$C_{ox} \cdot L_{eff} \cdot \left(W_{OV} + W_{eff}\right)$	$A_S \cdot C_{js} + P_S \cdot C_{js-sw}$	$A_D \cdot C_{jd} + P_D \cdot C_{jd-sw}$
Saturation	$C_{ox} \cdot W_{eff} \cdot \left(L_D + \frac{2}{3} \cdot L_{eff}\right)$	$C_{ox} \cdot W_{eff} \cdot L_D$	$C_{ox} \cdot L_{eff} \cdot W_{OV}$	$\left(A_S + A_{CH}\right) \cdot C_{js} +$ $\left(P_S - W_{eff}\right) \cdot C_{js-sw}$	$\left(A_S + A_{CH}\right) \cdot C_{js} +$ $\left(P_S - W_{eff}\right) \cdot C_{js-sw}$
Linear/ Triode	$C_{ox} \cdot W_{eff} \cdot \left(L_D + \frac{L_{eff}}{2}\right)$	$C_{ox} \cdot W_{eff} \cdot \left(L_D + \frac{L_{eff}}{2}\right)$	0	$\left(A_S + \frac{A_{CH}}{2}\right) \cdot C_{js} +$ $\left(P_S - W_{eff}\right) \cdot C_{js-sw}$	$\left(A_D + \frac{A_{CH}}{2}\right) \cdot C_{jd} +$ $\left(P_D - W_{eff}\right) \cdot C_{jd-sw}$



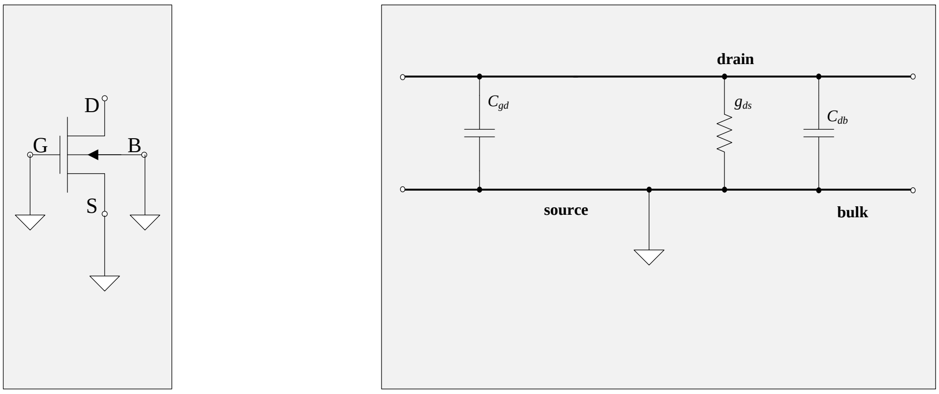
a) b)
Figure 3-18 Common-gate transistor: a) Symbol; b) Small signal equivalent model

Otherwise, with all the terminals connected to signal nodes, except the bulk terminal, as depicted in Figure 3-19, the transistor can be designated as *signal-transistor*. The small signal equivalent model consists of three capacitors, C_{gs}, C_{gd} and C_{db}, the transconductance, *gm*, the bulk transconductance, *gmb,* and the conductance, g_{ds}.



a) b)
Figure 3-19 Signal-transistor: a) Symbol; b) Small signal equivalent model

When the gate, source and bulk are all connected to constant DC voltages, generally to the ground (NMOS), as depicted in Figure 3-20, it is considered a *current-source* device. The gate-source, v_{gs}, voltage is null and the transistor behavior is resumed to two capacitors, C_{gd} and C_{db}, and the conductance, g_{ds}.



a) b)

Figure 3-20 Current-source transistor: a) Symbol; b) Small signal equivalent model

Considering again the circuit example shown in Figure 3-1, the small signal equivalent model is depicted in Figure 3-21. Since it is a differential circuit, it is shown only half of the equivalent model for the sake of simplicity.

At the input signal node, transistor M_1 it is connected in common-source configuration and the body-effect is not considered. Next, the M_2 is a current-source and contributes with the conductance, g_{ds}, and the two parasitic capacitances, C_{db} and C_{gd}. The M_3 is connected between two signal nodes, N_A and N_B, in common-gate configuration. The four parasitic capacitors are connected from the respective nodes to the ground. Conductance, g_{ds}, and the transconductances, *gm* and *gmb* are connected between the two signal nodes. Since the gate and bulk are ground-connected, the currents of the two transconductances of the M_3 are controlled by the source voltage, v_{NA}, according to:

$$v_{sg_{M3}} = v_{s_{M3}} - v_{g_{M3}} = v_{s_{M3}} = v_{NA} \tag{3.45}$$

$$v_{sb_{M3}} = v_{s_{M3}} - v_{b_{M3}} = v_{s_{M3}} = v_{NA} \tag{3.46}$$

The same approach is followed with the M_4, connected between the two signal nodes, N_B and N_C. The currents of the two transconductances of the M_4 are controlled by the source voltage, v_{NC}, according to:

$$v_{gs_{M4}} = v_{g_{M4}} - v_{s_{M4}} = -v_{s_{M4}} = -v_{NC} \tag{3.47}$$

$$v_{bs_{M4}} = v_{b_{M4}} - v_{s_{M4}} = -v_{s_{M4}} = -v_{NC} \tag{3.48}$$

From both nodes N_A and N_B, respectively, two compensation capacitors are connected to the output node, N_O. The output node connects to the M_7 in source-current configuration and to the M_6 in common-source configuration.

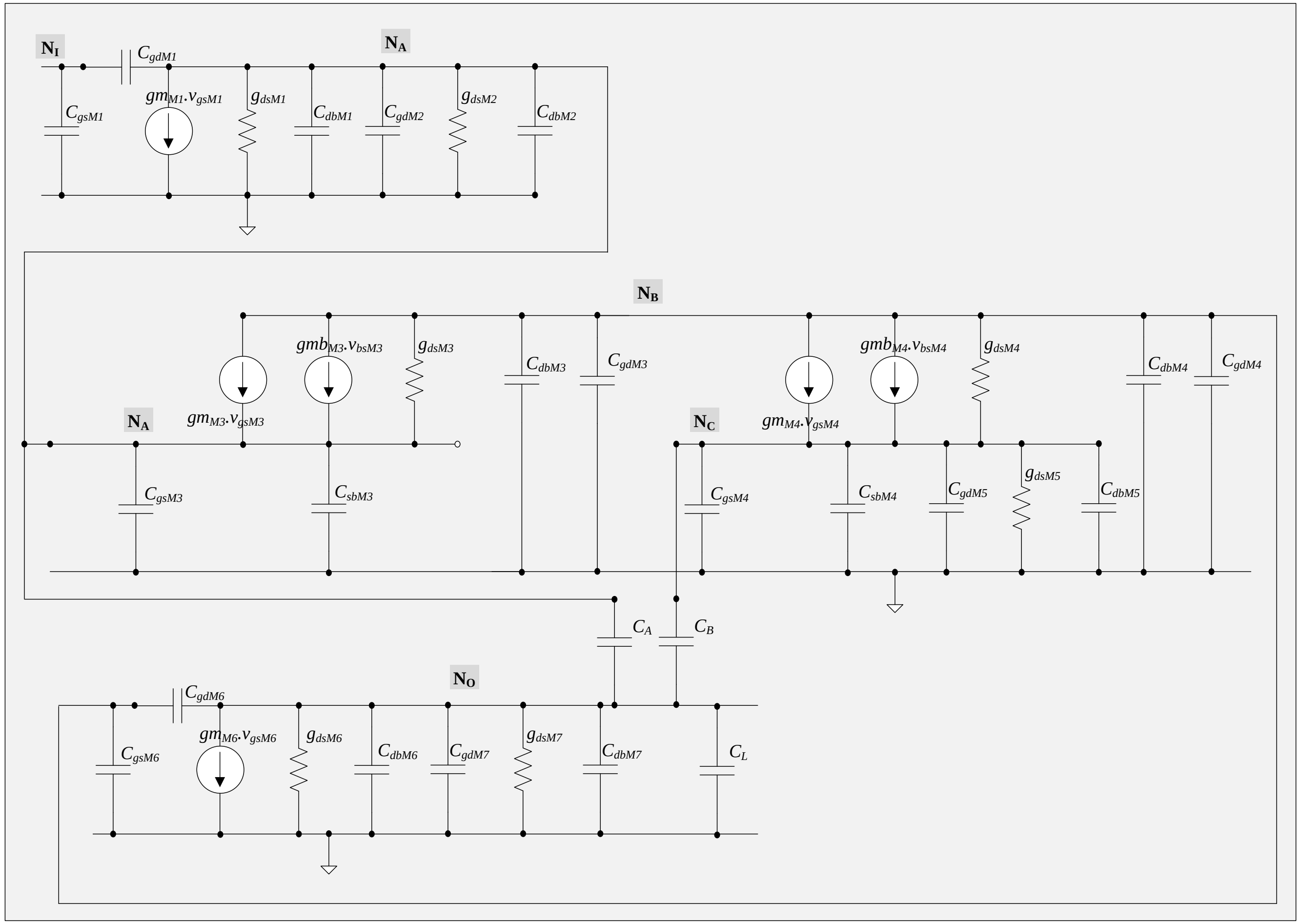


Figure 3-21 Small signal equivalent model example of half of the circuit amplifier shown in Figure 3-1.

To simplify the model and reduce the number of devices and equations, the parallel elements are lumped, throughout the circuit model. The Figure 3-22 shows the resulting simplified model of the half of the circuit amplifier shown in Figure 3-1.

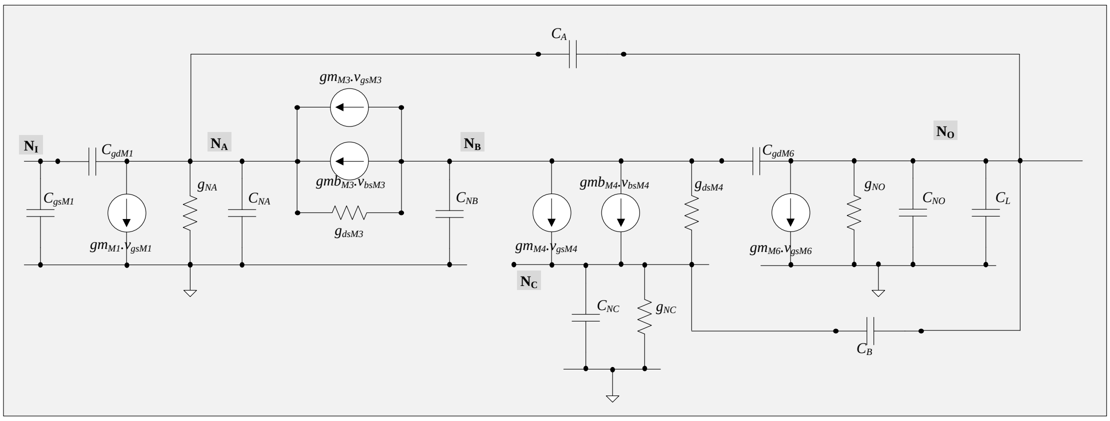


Figure 3-22 Simplified small signal equivalent model of the half of the circuit amplifier shown in Figure 3-1

Each g_N and c_N represents the sum of conductance and capacitance, respectively, elements in parallel connected from the respective node to ground.

The next step, described in the next section, is to isolate the nodes, including the mutual effects on each node.

3.4.6 Node isolation using Y-parameters

Admittance parameters or Y-parameters is a technique used to describe the linear behavior of electrical two-port networks. In the work it uses the two port Y-parameters, represented in Figure 3-23. Particularly in this work, the admittance parameters are used to isolate two nodes of the amplifier circuit. The relationship between the input voltages, output currents and the Y-parameter matrix is given by:

$$\begin{pmatrix} I_1 \\ I_1 \end{pmatrix} = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} \tag{3.49}$$

where

$$Y_{11} = \left(\frac{I_1}{V_1} \right)_{V_2=0} \tag{3.50}$$

$$Y_{21} = \left(\frac{I_2}{V_1} \right)_{V_2=0} \tag{3.51}$$

$$Y_{12} = \left(\frac{I_1}{V_2}\right)_{V_1=0} \tag{3.52}$$

$$Y_{22} = \left(\frac{I_2}{V_2}\right)_{V_1=0} \tag{3.53}$$

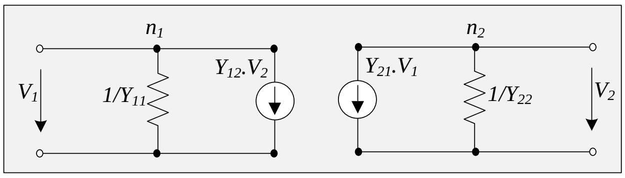


Figure 3-23 Y-Equivalent two port showing independent variables V_1 and V_2

The types of elements connected between two non-zero nodes, and considered throughout this work, along with the corresponding Y-parameters equivalents are presented next.

A capacitor connected between nodes n_1 and n_2, non-grounded, results on the following Y-parameters:

$$Y_{11} = \left(\frac{I_1}{V_1}\right)_{V_2=0} = \frac{sCV_1}{V_1} = sC \tag{3.54}$$

$$Y_{21} = \left(\frac{I_2}{V_1}\right)_{V_2=0} = -sC \tag{3.55}$$

$$Y_{12} = \left(\frac{I_1}{V_2}\right)_{V_1=0} = -sC \tag{3.56}$$

$$Y_{22} = \left(\frac{I_2}{V_2}\right)_{V_1=0} = sC \tag{3.57}$$

The Y-equivalent small signal model is rebuilt adding a capacitor and a transconductance connected to each node, n_1 and n_2, as shown in Figure 3-24.

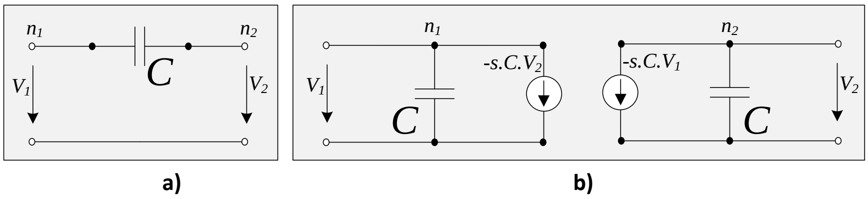


Figure 3-24 Y-paramenters of a capacitor: a) Capacitor; b) Y-Equivalent

The conductance connected between nodes n_1 and n_2, non-grounded, results on the following Y-parameters:

$$Y_{11} = \left(\frac{I_1}{V_1}\right)_{V_2=0} = G \tag{3.58}$$

$$Y_{21} = \left(\frac{I_2}{V_1}\right)_{V_2=0} = -G \tag{3.59}$$

$$Y_{12} = \left(\frac{I_1}{V_2}\right)_{V_1=0} = -G \tag{3.60}$$

$$Y_{22} = \left(\frac{I_2}{V_2}\right)_{V_1=0} = G \tag{3.61}$$

The Y-equivalent small signal model is rebuilt adding a conductance and a transconductance connected to each node, as shown in Figure 3-25.

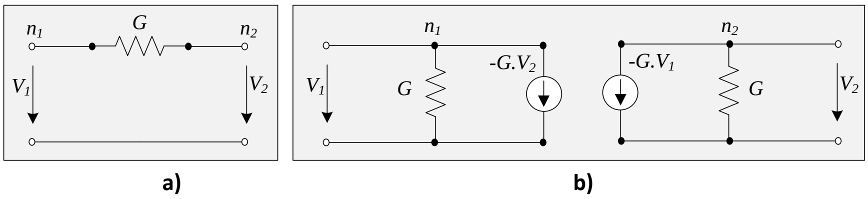


Figure 3-25 Y-paramenters of a conductance: a) Conductance; b) Y-Equivalent

The transconductance controlled by the voltage of a node, which the transconductance is connected to, results on the following Y-parameters:

$$Y_{11} = \left(\frac{I_1}{V_1}\right)_{V_2=0} = gm \tag{3.62}$$

$$Y_{21} = \left(\frac{I_2}{V_1}\right)_{V_2=0} = -gm \tag{3.63}$$

$$Y_{12} = \left(\frac{I_1}{V_2}\right)_{V_1=0} = 0 \tag{3.64}$$

$$Y_{22} = \left(\frac{I_2}{V_2}\right)_{V_1=0} = 0 \tag{3.65}$$

The Y-equivalent small signal model is rebuilt adding a conductance to the node that controls and a transconductance connected to the other node, as shown in Figure 3-26.

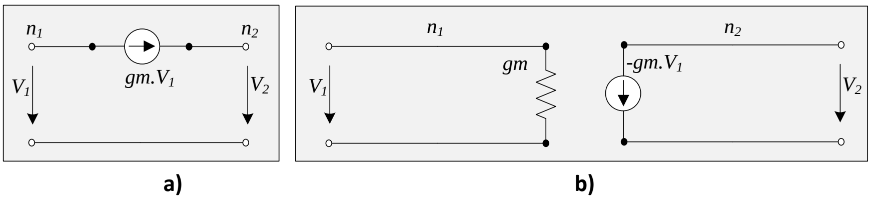


Figure 3-26 Y-paramenters of a transconductance, which controller-voltage is *V₁*:
a) Transconductance; b) Y-Equivalent

The transconductance controlled by the voltage of a third node, which the transconductance is not connected to, results on the following Y-parameters:

$$Y_{11} = \left(\frac{I_1}{V_1}\right)_{V_2=0} = 0 \tag{3.66}$$

$$Y_{21} = \left(\frac{I_2}{V_1}\right)_{V_2=0} = -gm \cdot V_X \tag{3.67}$$

$$Y_{12} = \left(\frac{I_1}{V_2}\right)_{V_1=0} = gm \cdot V_X \tag{3.68}$$

$$Y_{22} = \left(\frac{I_2}{V_2}\right)_{V_1=0} = 0 \tag{3.69}$$

The Y-equivalent small signal model is rebuilt adding a transconductance to each node, as shown in Figure 3-27.

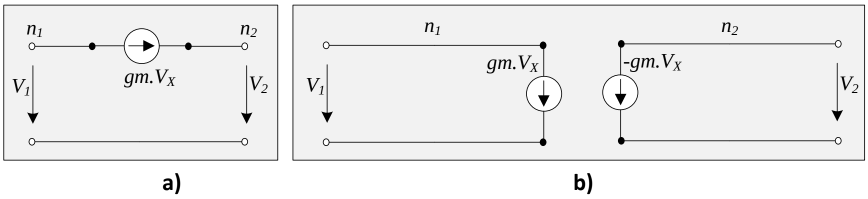


Figure 3-27 Y-paramenters of a transconductance, which controller-voltage is given by (other) voltage, *Vₓ*:
a) Transconductance II; b) Y-Equivalent

3.5 Performance parameters of the amplifiers

The MOS operational amplifier (opamp) is the most complex and most commonly building block in larger circuits and systems (i.e. SoC).

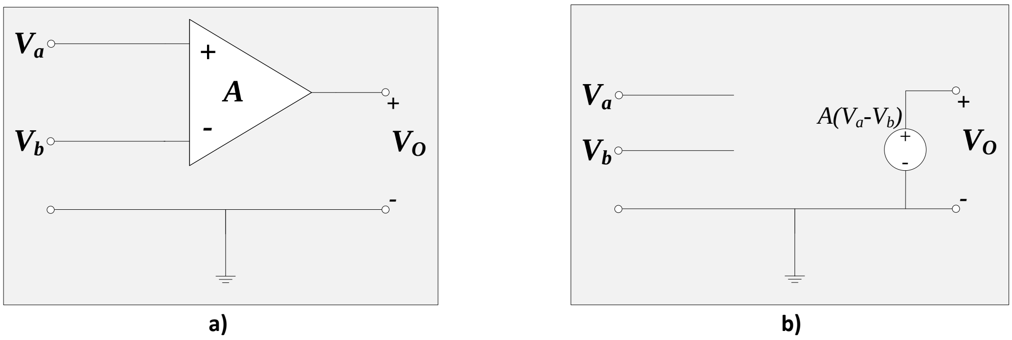


Figure 3-28 Ideal opamp: a) Symbol; b) Equivalent circuit

Figure 3-28 presents the symbol and the equivalent circuit for an ideal opamp. Although, this is an ideal representation, and hence, practical opamps can only approximate this ideal device, it helps to explain the performance parameters of the opamps. Some of these performance parameters will be used in the definition of the fitness function during the optimization process described in the next chapters. Note that the following sections are not a full coverage of all existing performance parameters but rather, they introduce the some of the concepts used later on this thesis.

3.5.1 Transfer function

The generic form of the transfer function of an amplifier, without feedback (the open-loop gain) is represented by (3.70). The A_0 indicates the finite gain of the amplifier at low frequency, .i.e. $s = 0$. The zeros are represented by z_N and the p_M are the poles of the transfer function.

$$A_{OL}(s) = A_0 \cdot \frac{\left(1+\frac{s}{z_1}\right)\cdots\left(1+\frac{s}{z_N}\right)}{\left(1+\frac{s}{p_1}\right)\cdots\left(1+\frac{s}{p_M}\right)} \tag{3.70}$$

The output signal of the amplifier can be written as (3.71)

$$V_o(s) = A_{CM}(s)\left[\frac{V_a(s)+V_b(s)}{2}\right] + A_{DM}(s)\left[V_a(s)-V_b(s)\right] \tag{3.71}$$

where $V_a(s)$ and $V_a(s)$ are the input signal, positive and negative, respectively, $A_{CM}(s)$ is the common mode gain, defined by

$$A_{CM}(s) = \frac{V_o(s)}{\frac{V_a(s)+V_b(s)}{2}} = \frac{V_o(s)}{V_{i_{CM}}(s)} \tag{3.72}$$

and $A_{DM}(s)$ is the differential gain, defined by

$$A_{DM}(s) = \frac{V_o(s)}{V_a(s)-V_b(s)} = \frac{V_o(s)}{V_{i_{DM}}(s)} \tag{3.73}$$

Using feedback theory, the transfer function of the amplifier with feedback (the closed-loop gain) is given by (3.74)

$$A_{CL}(s) = \frac{A_{OL}}{1+A_{OL}\beta} \tag{3.74}$$

where A_{OL} is the open-loop transfer function of the amplifier (either A_{DM} or A_{CM}, depending on the analysis), and the β is the feedback factor.

3.5.2 Gain-bandwith product

Due to the capacitances, finite carrier mobility and so on, the gain of the opamps are not constant over the frequency range, it decreases at high frequencies. One way to measure this effect is the gain-bandwidth product, GBW, The gain of the opamps is constant until the 1st pole occurs. The GBW is defined by

$$GBW = A_0(s) \cdot F_{p1}(s) \tag{3.75}$$

where $A_0(s)$ is the low-frequency gain and $F_{p1}(s)$ is the 1st pole frequency. In amplifiers with a dominant pole (and when the high frequency poles are at very high frequency) the GBW is equal to the unity-gain frequency, UGF.

3.5.3 Phase margin

For stability, all poles, p_M, must be in the negative half plane of the s-plane; that is, the real part of all poles must be negative [80]. One sufficient condition for stability is the phase margin, φ_M. This measure is based on the *loop-gain* factor, $\beta.A_{OL}(s)$, of the opamp transfer function (3.74). Let $s=j\omega$ and φ_M is defined by (3.76).

$$\varphi_M = \arg\left(\beta \cdot A_{OL}\left(j\omega_0\right)\right) - \left(-\pi\right) \tag{3.76}$$

where ω_0 is the frequency at which the product $\beta.A_{OL}$ is equal to 0 dB. A larger φ_M corresponds to a more stable opamap (typically, φ_M > 60º).

3.5.4 Positive and negative power supply rejection ratio

The power supply rejection ratio (PSRR) measures the amplifiers ability to suppress variations in the power supply voltages. In the ideal case, a change in supply voltage will not affect the performance of the amplifier. However, in reality, changing the power supply voltage will affect the bias levels and thereby the operation of the circuit. If large variations in the power supply voltage are present due to, e.g., high switching activity in surrounding digital circuitry, it is important that these variations have small impact on the performance degradation of the amplifier. Both positive PSRR$^+$, i.e., the suppressions of variation in the positive power supply voltage, and negative PSRR$^-$, i.e., the suppression of variations in the negative power supply voltage are of interest. The definitions are

$$\text{PSRR}^+(s) = 20Log_{10}\left(\frac{A_{DM}(s)}{A_{VDD}(s)}\right) \tag{3.77}$$

$$\text{PSRR}^-(s) = 20Log_{10}\left(\frac{A_{DM}(s)}{A_{VSS}(s)}\right) \tag{3.78}$$

where A_{VDD} is the magnitude of the frequency response from the positive power supply to the output terminal and A_{VSS} is the magnitude of the frequency response from the negative power supply to the output terminal.

3.5.5 Common mode rejection ratio

The common mode rejection ratio (CMRR) is a measure of how unwanted common mode signals (i.e. noise) on the amplifier input terminal are suppressed. In (3.79) the gain of the common-mode signal is compared to gain of the differential signal. In the ideal case the CMRR is infinitely large, i.e., the common-mode signal is not amplified at all. The CMRR is defined by

$$\text{CMRR}(s) = 20Log_{10}\left(\frac{A_{DM}(s)}{A_{CM}(s)}\right) \tag{3.79}$$

3.5.6 Slew rate

For a large input step voltage, some transistors in the opamp may be driven out of their saturation regions or even completely cut off. As result, the output will follow the input at a slower and finite rate. The slew rate (SR) is defined as the maximum rate of change of the output voltage of an amplifier and is defined by

$$\mathrm{SR} = \max\left(\left|\frac{dv_o(t)}{dt}\right|\right) \tag{3.80}$$

where $v_o(t)$ is the output voltage of the amplifier, as function of time, *t*.

Considering a two-stage opamap with a given compensation capacitance, C_c, connected between the outputs of the two stages, the SR effect is also due to the maximum supplied current to the output, I_0, available for charging up the compensating capacitor, as follows:

$$\mathrm{SR} = \left|\frac{dV_o}{dt}\right| = \left|-\frac{1}{C_C}\cdot\frac{dQ_c}{dt}\right| = \frac{I_0}{C_C} \tag{3.81}$$

3.5.7 Noise (thermal and flicker)

In MOS transistors there are mainly two types of noise sources: thermal, N_T, and flicker noise (1/*f*), N_F [81]. The thermal noise component N_T is the result of random motions of electrons due to thermal effects, for instance, in resistors. Even in absence of a current, a fluctuating voltage v_{NT} exists due to temperature, *T*, variation. In [80] it is demonstrated that the *mean square* of v_{NT}, for a MOS, is given by an equivalent voltage source connected, in series with the gate terminal with the following value:

$$\overline{v_{NT}^2} \approx \frac{8}{3}\cdot\frac{kT}{gm}\cdot\Delta f \tag{3.82}$$

where *k* is the Boltzmann constant, Δf is the bandwidth in which the noise is being integrated, in Hz, and *gm* is the transconductance of the transistor.

In an MOS transistor, the extra electron energy states, that can trap and release electrons from the channel, producing the flicker noise component N_F. Since this process is relatively slow, most of this noise energy appears at low frequency and decreases inversely with the frequency, *f*. From [80] it can be stated that the gate-referred noise voltage (voltage source connected, in series with the gate terminal) is given by the approximating formula:

$$\overline{v_{_{NF}}^2} \approx \frac{K}{C_{OX} \cdot W \cdot L} \cdot \frac{\Delta f}{f} \tag{3.83}$$

where *K* is dependent on the temperature and the fabrication process, *W* and *L* are the width and length of the transistor, respectively, and C_{ox} is the gate oxide capacitance *per* area.

In the small-signal model of a MOS transistor, the noise representation is a current source, in parallel with the current sources, *gm.*v_{GS} and *gmb.*v_{Sb},as shown in the Figure 3-29.

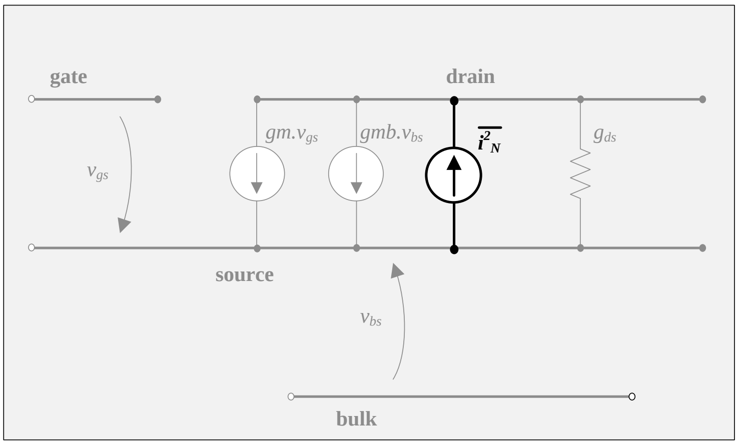


Figure 3-29 Transistor model for small-signal with noise source

Its value is the combination of the two noise sources described previously, which is:

$$\overline{i_{_N}^2} = \overline{i_{_{NT}}^2} + \overline{i_{_{NF}}^2} \approx \left[\frac{8}{3} \cdot k \cdot T + \frac{K \cdot gm}{C_{OX} \cdot W \cdot L \cdot f}\right] \cdot gm \cdot \Delta f \tag{3.84}$$

In an amplifier circuit, the power noise spectral density, for example, at the output is computed by summarizing the contribution of all independent noise sources in the circuit according to (3.85)

$$S_{out}(\omega) = \sum_{n=1}^{N} S_n \cdot \left|H_n(\omega)\right|^2 \tag{3.85}$$

where *N* is the number of noise sources, S_n is the power spectral density of the noise source, and H_n is the magnitude response from the noise when referred source to the output of the circuit. By integrating the noise over a frequency band the noise power is obtained.

3.5.8 Output swing

The output swing (OS) is the range the output voltage can vary without suffer distortion caused by the output transistors leaving the saturation region. Both positive OS$^+$, i.e., the variation of the output signal until the positive power supply voltage, V_{DD}, and negative OS$^-$, i.e., the variations of the output signal until the negative power supply voltage, V_{SS} are of interest. The definitions are

$$\mathrm{OS}^+ = V_{DD} - V_{CMO} - \left(\sum_{i=1}^{k} V_{dsat_i} \right) - V_{margin} \tag{3.86}$$

$$\mathrm{OS}^- = V_{CMO} - V_{SS} - \left(\sum_{i=1}^{j} V_{dsat_i} \right) - V_{margin} \tag{3.87}$$

where V_{CMO} is the common-mode voltage value at the output, k, is the number of the transistors stacked at the output from the output node to the positive supply voltage, j, is the number of the transistors stacked at the output from the output node to the negative supply, V_{dsat_i} is the saturation voltage of each transistor at the output branch, and V_{margin} is a safety margin, typically, 50 mV to 100 mV. The overall OS is then defined by:

$$\mathrm{OS} = \min\left\langle \mathrm{OS}^+, \mathrm{OS}^- \right\rangle \tag{3.88}$$

Generally, and considering that the common-mode voltage of the output, V_{OCM}, is centered at $V_{DD}/2$, the negative supply voltage is 0 V, the OS can be computed by the following:

$$\mathrm{OS} = V_{DD} - \left(\sum_{i=1}^{t} V_{dsat_i} \right) - \left(2 \cdot V_{margin} \right) \tag{3.89}$$

t is the number of each transistor at the output stage.

3.5.9 Settling time

The settling time (ST) denotes the time required for the output signal of an amplifier to adjust (settle) when a step is applied to the input and it is in a given closed-loop configuration. Depending on the magnitude of the step, the settling can be linear or nonlinear. For a small step, only the bandwidth of the amplifier limits the ST. In this case the settling is linear. The linear settling determines an overall upper limit of the ST. However, when a large step is applied to the input terminal, the amplifier experiences slew rate limitation due to the finite current that can be supplied to the internal or output capacitive nodes. In this case the settling is nonlinear. The settling is computed by applying a step to the amplifiers input terminal and measure the time until the output signal is within a certain range of its final value as shown in Figure 3-30. The exact range may vary depending on the application of the amplifier.

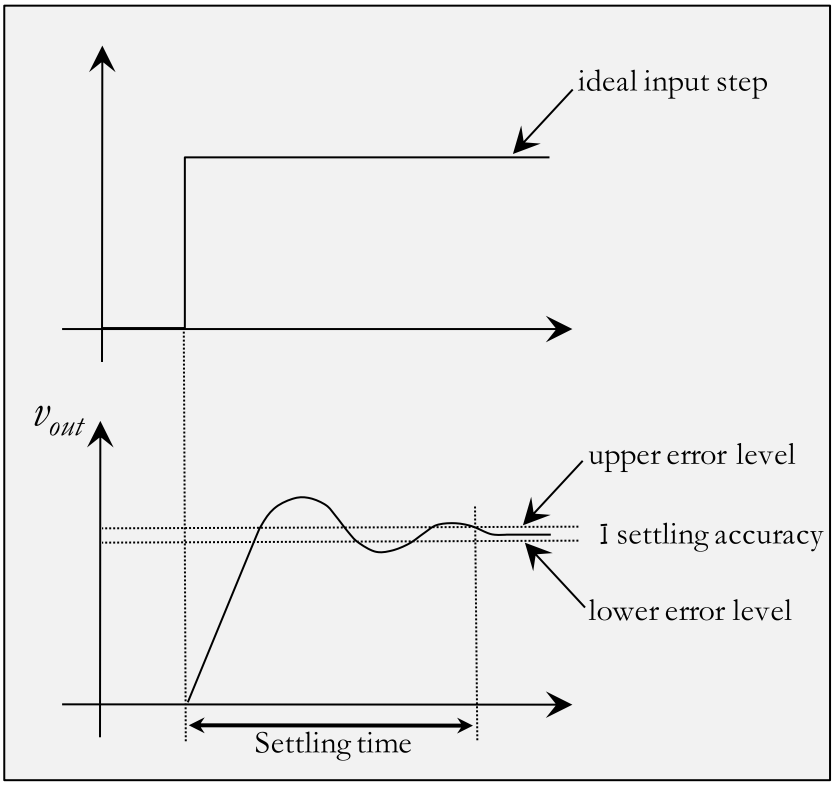


Figure 3-30 Settling time representation

3.5.10 Die area

Silicon area of chip is not directly linked to the circuit performance. It highly affects the manufacturing cost and therefore is important to make the circuit as small as possible. Considering the optimization theme, it is assumed that the active area is given by:

$$Area = \sum_{i=1}^{All} W_i \cdot L_i + \sum_{j=1}^{All} C_j \tag{3.90}$$

where, W and L are the width and length, respectively, of each transistor in the circuit, and C is the area of the passive elements, e.g. compensation capacitors. Moreover, the area occupied by the passive components is the most significant.

3.5.11 Power dissipation

The power dissipation is more important than ever, with a large number of applications nowadays running on battery. The power dissipation directly affects the operation times for such products and is therefore an important performance metric. It can be simply computed by:

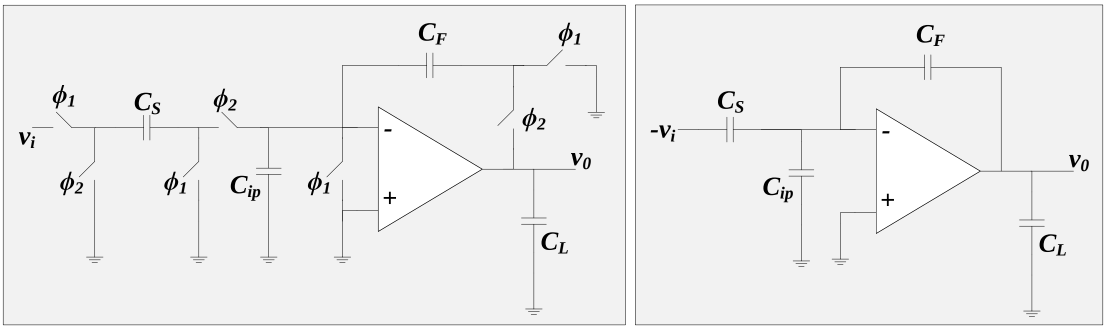
$$Power = V_{DD} \cdot I_{total} \tag{3.91}$$

where V_{DD} is the global voltage supply, which multiplies by the sum of currents, I_{total}, from all branches from V_{DD} to V_{SS}.

3.6 Transfer Function of the Amplifiers when Employed in Switched-Capacitor Circuits

In analog signal-processing, the absolute tolerances of resistors and capacitors used on continuous circuits are not good enough to perform most signal-processing functions [78]. In the early 1970s, analog sampled-data techniques were used to replace the resistors, from the circuits [82][83]. These circuits are called *switched-capacitors* (SC) *circuits* (theoretically invented by James C. Maxwell) and became very popular in implementing analog circuits in standard CMOS technologies. The main reason for the widespread usage of these circuits is that the accuracy of the circuits is proportional to the accuracy of capacitor ratios, which can be quite good (≈ 0.1%).

Figure 3-31 shows an example of a switched-capacitor circuit, an integrating-type sample-and-hold (S/H) circuit. At phase, ϕ_1, C_S samples the input voltage while C_F and C_L are connected to ground. During phase, ϕ_2, the capacitance of C_S is transferred to C_F and the previously sample input signal is applied to C_L. During phase, ϕ_2, the circuit can be approximately represented by the block diagram illustrated in Figure 3-32.



a) b)

Figure 3-31 Switched-Capacitor S/H: a) Full circuit; b) Equivalent circuit on phase ϕ_2

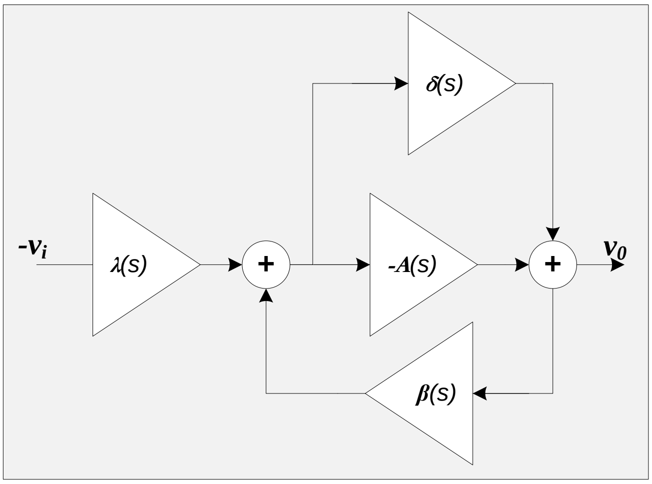


Figure 3-32 Circuit diagram during phase ϕ_2

where *A(s)* is the transfer function of the amplifier, *λ(s)* is the transfer function from the SC circuit input to the input of the amplifier, *β(s)* is the feedback factor, that is, the transfer function from the output of the circuit to the input of the amplifier, and *δ(s)* is the direct forward contribution of the input signal to the output signal. The global transfer function, TF, is given by:

$$\mathrm{TF}=\frac{v_0(s)}{v_i(s)}=\frac{\lambda(s)}{\beta(s)+\dfrac{1}{A(s)-\delta(s)}} \tag{3.92}$$

where *λ(s)*, *β(s)* and *δ(s)*, considering the circuit on Figure 3-32, are given by:

$$\lambda=\frac{C_S}{C_S+C_{ip}+C_F} \tag{3.93}$$

$$\beta = \frac{C_F}{C_S + C_{ip} + C_F} \tag{3.94}$$

$$\delta = \frac{C_F}{C_L + C_F} \tag{3.95}$$

3.7 Time-Domain *versus* Frequency-Domain Optimization

It is clear that there are innumerous performance parameters used for analog circuits. What makes the design process even more challenging is the nonlinear relation between them. For example, when trying to lower the power supply voltage the voltage range is reduced. If the voltage range is reduced the Signal-To-Noise Ratio (SNR) of a given circuit is, consequently, decreased. Thus, determining the relations between the performance parameters is crucial in analog circuit design. Simple device models can give some information of possible trade-offs. However, the real value of the performance may not be accurately predicted until simulated using high-order device models.

In the particular case of an amplifier, if the optimization process is performed in the frequency domain the objectives and trade-offs are multiple, and the system has to handle each of these variables independently. Furthermore, it increases the complexity that the search algorithm has to handle to converge to an optimal solution. For instance, it is necessary to use separate optimization goals for open loop gain, for the GBW and for the circuit stability.

To keep the complexity in a reasonably level for a system to attain different objectives, mentioned before, it is common to simplify the circuit to a third order system. As opposed to the methodology presented here, which, as mentioned, is capable of handling any number of zeros and poles in the transfer function. Consequently, the traditional frequency-domain analysis is not a good approach to efficiently size and properly compensate the amplifiers, because of the loss of accuracy in the circuit performance estimation[67].

The effect of the voltage variation on the r_{ds} value of the transistors is difficult to include in the frequency-domain analysis. To have a more exact design procedure, it is necessary to perform a time-domain –transient- simulation of the circuit.

Generally, the amplifiers must have first order behavior up to the unit-gain frequency (UGF). In order to obtain this, the different non-dominant poles and positive zeros in the signal path must occur at frequencies beyond the GBW. To impose these design requirements is called pole placement.

A quantity expressing a first order behavior is the PM. If the PM is below 60º, one can determine if non-dominant poles occur before the unity gain frequency, f_u. The PM could thus be used for pole placement. However, a pole or zero only has an influence on the phase in a frequency region of one decade before and one decade after its frequency (corresponding to one decade I_{DS} and/or V_{CS}-V_{TN}). This implies that if a pole or zero falls outside the region between one decade before and one decade after the UGF, its influence on the PM is negligible.

In time-domain optimization, the evaluation of the circuit is, mostly, driven by the settling-time and the correspondingly settling-accuracy. Considering these two functions, it is guaranteed that the circuit has enough gain (small settling-error), sufficient bandwidth response (settling-time) and that it is stable. Ahead of this, the optimization objective can attain the two different settlings: differential and common mode.

The later circuit designs optimized on this platform also included the common-mode step-response in the fitness calculation. Following this, it was observed that the differential-mode converged more rapidly to the desired results.

4 Platform Architecture and Genetic Algorithm Kernel

This chapter presents the software platform architecture and the genetic algorithm (GA) kernel developed to support the proposed methodology for the optimization of circuit amplifiers.

The GA runs during a number of generations, *nger*, each one composed of a population, *P*, with a certain number of individuals, *nind*, which are composed of a given number of genes, *ngen*. The genes represent the design parameters that the algorithm will search for the best value within the design space. The design space is limited by the range values defined for the values of the genes.

Each individual, *I*, is directly mapped into a different circuit, where the gene values represent, for instance, transistor sizes or compensation capacitance values. The classification of each individual depends on the performance parameters of the corresponding circuit. Furthermore, the circuits are classified by comparing the achieved performance parameters and the respective objective values initially defined.

The evaluation procedure of the proposed optimization methodology may use either analytical equations or electrical simulation as the evaluation method to assess the performance of the circuits. In order to allow some flexibility, in the implemented software platform, the evaluation procedure is realized as an independent source-code library. This way, it is possible to have an individual/circuit evaluator with different levels of accuracy. It can be composed of simplified equations (level 2) of MOS transistors, or based on the accurate model of the MOS transistor, e.g. BSIM3, which takes into account complex non-linear effects. Furthermore, the evaluation of each circuit includes the analysis for several fabrication process, voltage supply and temperature variations (PVT) corners, and considering the worst-case performance of a given corner.

Accuracy in the evaluation has a price: it is time consuming. However, the possibility of making the evaluations, simultaneously, diminishes this drawback and improves the platform performance. Considering that the genetic algorithm is suitable for distributed evaluation, as claimed before, a distributed/parallel version is implemented and will be described in section 4.6 of this chapter.

This platform integrates the search algorithm, the circuit evaluation module with the circuit knowledge, and the circuit elements models as different pieces of software. This configuration is appropriate for fast redesigns, for instance, using new silicon technologies parameters values and/or diverse elements models, e.g. EKV[84], because it is only necessary to change the source-code related to the models of the transistor models.

The next sections present a brief illustration of the implementation, based on genetic algorithm (GA), used to develop the software platform.

4.1 Platform Architecture

The architecture of the optimization and sizing platform is depicted in Figure 4-1.

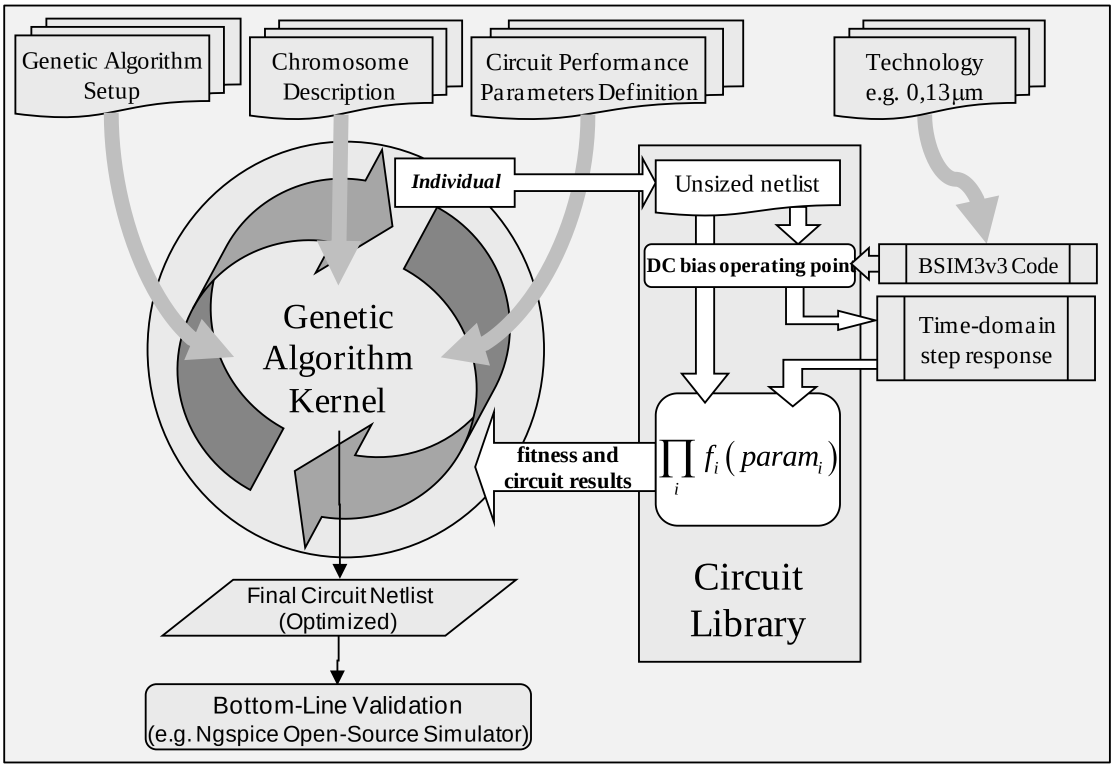


Figure 4-1 General architecture of the proposed optimization platform

The platform can be separated into two main parts: the search algorithm and the circuit evaluation library. With this approach, the calculations of the circuit performance parameters are encapsulated and the optimizer kernel is not dependent on the actual circuit evaluator and/or transistor model. Therefore, it is possible to integrate different forms of circuit evaluators and/or other transistors models into the optimization process.

The search part of the platform is based on GAs and it requires three input configuration files: the chromosome description; the circuit performance parameters definition; and the GA setup. At the end of the search, the result is a circuit description file (netlist) ready for verification through an electrical simulator, e.g. NGSPICE.

During the search, each circuit evaluation is performed either using analytical equations or electric simulations. This task also requires a configuration file that contains the value of the parameters of the target fabrication technology. The evaluation's results correspond to the specific circuit performance parameters values, which are used to classify the circuit. Moreover, this classification corresponds, at the search algorithm part, to the fitness of the individual that originated the corresponding circuit.

Generally, the platform's execution stages are displayed in the Figure 4-2.

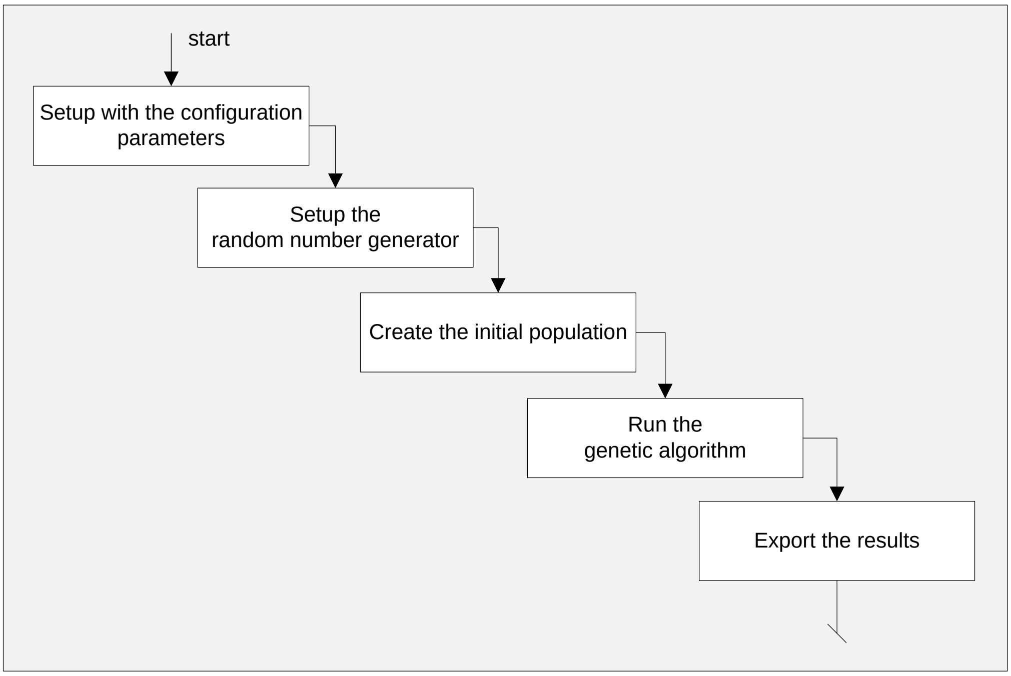

start
Setup with the configuration parameters
Setup the random number generator
Create the initial population
Run the genetic algorithm
Export the results

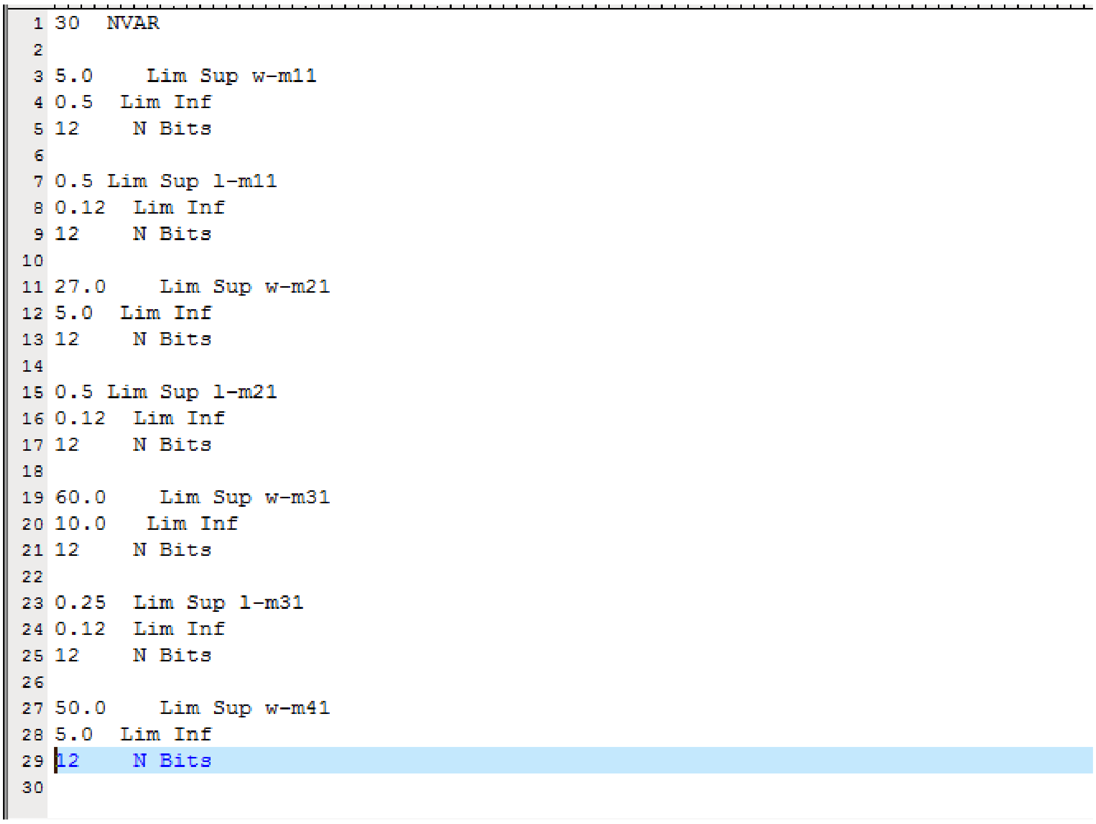
Figure 4-2 Flow of the execution steps of the proposed platform

The first stage of the platform execution consists of the reading the configuration parameters and setup of the GA, the random number generator initialization, the setup of the population specifications and circuit specific properties. The random number generation initialization is based on a remote system, which produces random numbers from atmospheric noise – random.org – [85]. After, the genetic algorithm optimization process runs. At the end, some statistics and the optimum result are exported.

4.1.1 Chromosome description file

This file describes the structure of the chromosome. It is composed of a set of genes. The genes act as the design parameters, which are the variables that the GA search, in order to find the best fit of the desired circuit performance parameters values (goals). A snap shot of a chromosome description file example is illustrated in Figure 4-3.

The first line specifies the total number of genes (*ngen*) in the file, and the next lines contain the description of each gene. For each gene, it is specified the upper (*Lim Sup*) and lower (*Lim Inf*) range values and, also, value resolution of the genes (R_g), as the number of bits used to represent its value. The number of bits is useful during crossover and mutation operations, as it will be described, later on, in sections 4.2.4 and 4.2.5. The user can add some text, such as, the name of the gene, i.e. *w-m11*, or some other comments.



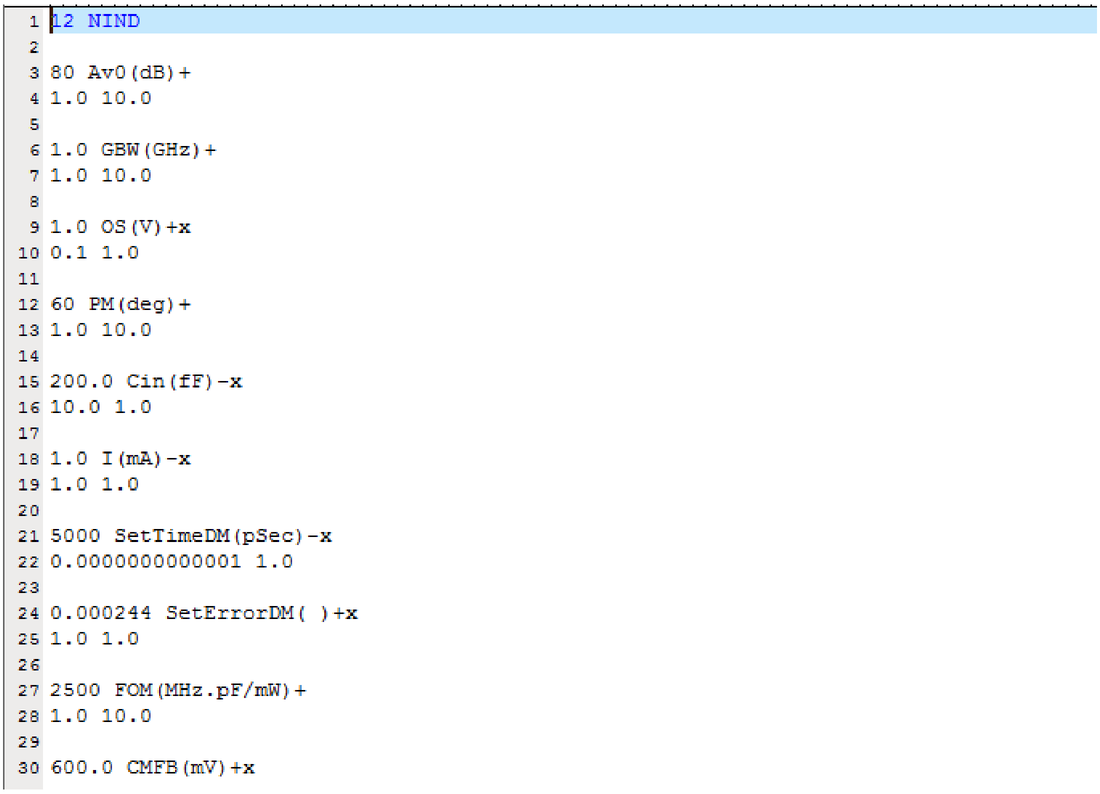
```
30   NVAR

5.0    Lim Sup w-m11
0.5  Lim Inf
12    N Bits

0.5 Lim Sup l-m11
0.12  Lim Inf
12    N Bits

27.0    Lim Sup w-m21
5.0  Lim Inf
12    N Bits

0.5 Lim Sup l-m21
0.12  Lim Inf
12    N Bits

60.0    Lim Sup w-m31
10.0   Lim Inf
12    N Bits

0.25  Lim Sup l-m31
0.12  Lim Inf
12    N Bits

50.0    Lim Sup w-m41
5.0  Lim Inf
12    N Bits

```

Figure 4-3 Chromosome description file snap shot

4.1.2 Circuit performance parameters definition file

In this file the user designates the circuit performance parameters to be provided, as an output, during and at the end of the optimization process. Furthermore, the search algorithm makes use of these values to compute the individual *fitness*, as described in the section 4.2.2. Moreover, these values act as the design goals that the algorithm will try to meet at the end of the optimization process. A snap shot of a circuit performance parameters definition file example is illustrated in Figure 4-4.

The first line specifies the total number of circuit performance parameters (*nparam*) in the file, and the remaining lines describe the circuit performance parameter. Each parameter is described with two consecutive lines. The first line specifies the target value followed by the indicator's text (i.e. name and the corresponding units). Again, this text is only indicative, for the user. The second line starts with the definition of the weight of the parameter, which is used to compute the individual *fitness*. The next field defines the stop criteria value, as it will be explained in the section 4.2.

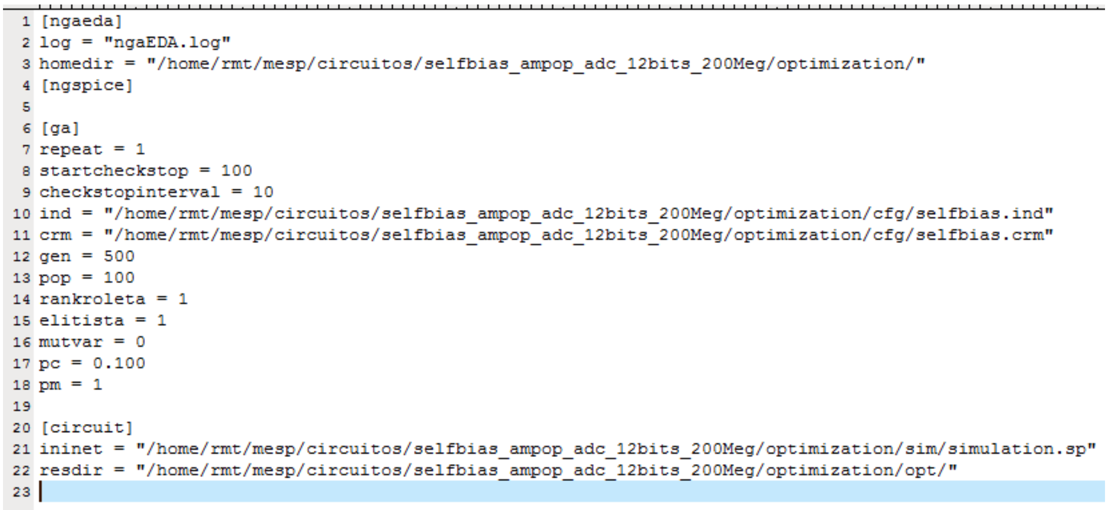


```
12 NIND

80 Av0(dB)+
1.0 10.0

1.0 GBW(GHz)+
1.0 10.0

1.0 OS(V)+x
0.1 1.0

60 PM(deg)+
1.0 10.0

200.0 Cin(fF)-x
10.0 1.0

1.0 I(mA)-x
1.0 1.0

5000 SetTimeDM(pSec)-x
0.0000000000001 1.0

0.000244 SetErrorDM( )+x
1.0 1.0

2500 FOM(MHz.pF/mW)+
1.0 10.0

600.0 CMFB(mV)+x
```

Figure 4-4 Circuit performance parameters definition file snap shot

4.1.3 Genetic algorithm setup file

Basically, this file contains the configuration options of the GA progress. A snap shot of a GA setup file example is illustrated in Figure 4-5.



```
[ngaeda]
log = "ngaEDA.log"
homedir = "/home/rmt/mesp/circuitos/selfbias_ampop_adc_12bits_200Meg/optimization/"
[ngspice]

[ga]
repeat = 1
startcheckstop = 100
checkstopinterval = 10
ind = "/home/rmt/mesp/circuitos/selfbias_ampop_adc_12bits_200Meg/optimization/cfg/selfbias.ind"
crm = "/home/rmt/mesp/circuitos/selfbias_ampop_adc_12bits_200Meg/optimization/cfg/selfbias.crm"
gen = 500
pop = 100
rankroleta = 1
elitista = 1
mutvar = 0
pc = 0.100
pm = 1

[circuit]
ininet = "/home/rmt/mesp/circuitos/selfbias_ampop_adc_12bits_200Meg/optimization/sim/simulation.sp"
resdir = "/home/rmt/mesp/circuitos/selfbias_ampop_adc_12bits_200Meg/optimization/opt/"

```

Figure 4-5 Genetic algorithm setup file snap shot

This text file is a configuration file based on the INI file format [86]. It is divided in different setup sections, which contain variables and the corresponding values. The most relevant are:

- **startcheckstop** – number of generations before verifying the stop criteria;
- **checkstopinterval** – number of generations interval to check the stop criteria;
- **ind** – path of the Circuit Performance Parameters Definition File;
- **crm** – path of the Chromosome Description File;
- **gen** – total number of generations (*nger*);
- **pop** – total number of individuals (*nind*);
- **rankroleta** – apply the rank operator (1), or apply the roulette operator (0) (binary choice);
- **elitista** – setup an elitism selection (1), or not (0) (binary choice);
- **mutvar** – mutation probably variable, during optimization (1), or not (0) (binary choice);
- **pc** – probably value of the cross-over: 0.0 to 1.0;
- **pm** – probably value of the mutation: 0.0 to 1.0;

4.2 Genetic Algorithm Overview

As stated before, the kernel of the optimization platform presented here is based on GAs. Although the full description of the GA theory is out of the scope of this thesis, a brief explanation of the key concepts is provided next.

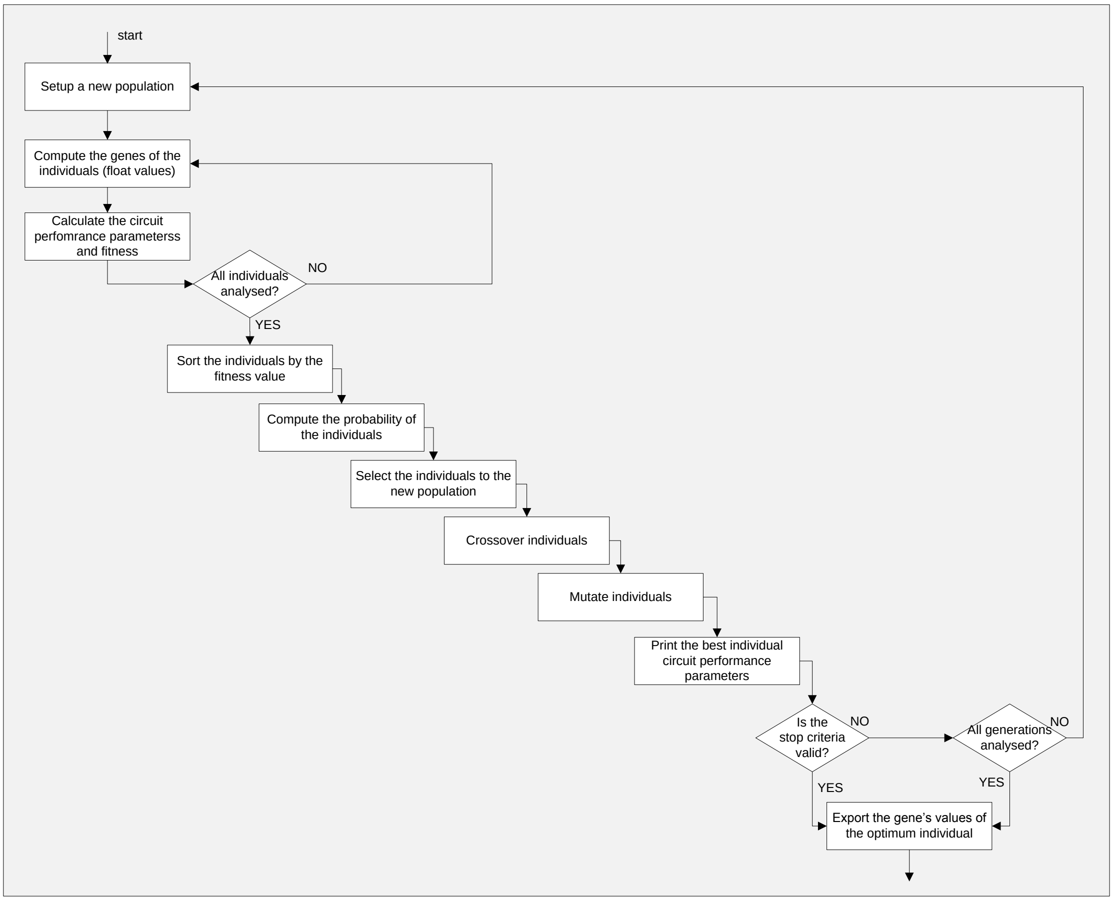

start
Setup a new population
Compute the genes of the individuals (float values)
Calculate the circuit perfomrance parameterss and fitness
All individuals analysed?
NO
YES
Sort the individuals by the fitness value
Compute the probability of the individuals
Select the individuals to the new population
Crossover individuals
Mutate individuals
Print the best individual circuit performance parameters
Is the stop criteria valid?
NO
YES
All generations analysed?
NO
YES
Export the gene's values of the optimum individual

Figure 4-6 Flow of the steps of the genetic algorithm

Figure 4-6 show the basic steps of the implemented GA. The initial step is to create a population with a given number of individuals (*nind*). The values of each individual's genes are randomly selected from the interval range values indicated in the chromosome description file. The number of individuals is defined in the genetic algorithm setup file. Then the execution enters in a loop until the stop criterion is reached or the last generation is reached.

The execution loop comprises the following four tasks:

1. classification of each individual;
2. selection of individuals to move-on into the new generation;

3. selecting the individuals to crossover and, consequently, generating new individuals for the new population;
4. choosing the individuals of this new population to mutate.

At the end, the best individual found is returned to the main execution loop, and the optimum circuit is provided, as an output, in a netlist format (i.e. SPICE compatible). Two options are available to end the optimization process: after the variation of the performance parameters values, from generation to generation, become less than a given percentage (stop criteria percentage); or if the number of generations reached the maximum number, *nger*, defined in the genetic algorithm setup file. If the values of the circuit performance parameters change less than the percentage amount (stop criteria percentage) defined within the circuit performance parameters definition file (see section 4.1.2) the optimization stops, automatically. This should indicate that the evolution of the values of the circuit performance parameters is reaching the optimum case and it will not change significantly. This verification occurs every *checkstopinterval* generations, after the generation *startcheckstop*. Both parameters are defined in the algorithm setup file.

4.2.1 Structure of an individual

In the GA each individual is a chromosome vector, $\vec{x}$, comprising one or more genes. The genes contain the design parameters, x_i, that the algorithm will find that best fit the design performance parameters, e.g. transistor dimensions (*W* and *L*). Figure 4-7 displays a generic form of an individual used in the optimization process.

The actual format of the individual depends on the topology of the circuit. As previously mentioned both, the total number of genes and the range values of each gene, are defined in a configuration text file that has to be provided by the user.

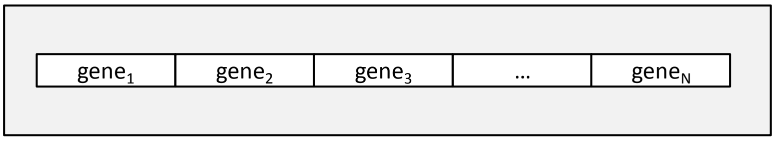


Figure 4-7 Format of a generic individual (chromosome)

Equation (4.1) is used to compute the float value of each gene, which is randomly built in binary format, with R_g bits.

$$x_i = x_{\min} + \left(x_{\max} - x_{\min}\right) \cdot \frac{\sum_{j=0}^{R_g-1} b_j \cdot 2^j}{2^{R_g-1}} \tag{4.1}$$

where x_{max} and x_{min} represent the upper (*Lim Sup*) and lower (*Lim Inf*) limits, respectively, of the gene value, R_g, is the number of bits defined to represent the value of the gene (as described in the chromosome description file, see section 4.1.1), and b_i is the binary value of the bit ('1' or '0').

4.2.2 The classification process

The process of classifying of each individual begins by mapping each chromosome into a circuit. The performances of the newly sized circuits are then evaluated, which originates the corresponding individuals' classifications, i.e. the fitness value. The fitness values result from the comparison between the behavior of each parameter of the circuit and the desired design performance parameters -how much the individual/circuit fits the desired specifications-. These specifications are described in the input file as already explained in section 4.1.2.

The individual fitness function is then computed according to (4.2)

$$fitness = \prod_i f_i\left(param_i\right) \tag{4.2}$$

where $f_i\left(param_i\right)$ are partial fitness of each design performance parameter of the circuit, compared to the desired specifications. It may assume three forms, depending on the desired type of optimization goal: maximize (4.3), minimize (4.4), or target value (4.5).

$$f_i\left(param_i\right) = 1 - e^{-weight_i \cdot \frac{param_i}{desired\_param_i}} \tag{4.3}$$

$$f_i\left(param_i\right) = 1 - e^{-weight_i \cdot \frac{desired\_param_i}{param_i}} \tag{4.4}$$

$$f_i\left(param_i\right) = \frac{2}{e^{-weight_i \cdot \frac{desired\_param_i - param_i}{desired\_param_i}} + e^{weight_i \cdot \frac{desired\_param_i - param_i}{desired\_param_i}}} \tag{4.5}$$

where *desired_param*$_i$ represent the objective value, *param*$_i$ is the present design performance parameter value achieved by the individual/circuit being evaluated, and *weight*$_i$ represents the importance (weight) of the respective indicator in the fitness calculation. The variable *weight*$_i$ acts as strength factor that induces the *param*$_i$ parameter evolution, proportionally to the *weight*$_i$ value.

Considering (4.3) and (4.4), when the *weight*$_i$ value is reduced, the exponent's argument is also reduced, which causes the *param*$_i$ parameter to increase, or decrease, depending on the partial fitness equation employed: maximize, (4.3); or minimize, (4.4), respectively. This effect is displayed in the Figure 4-8. The 'K' arrow points in the increasing direction of the *weight*$_i$ factor.

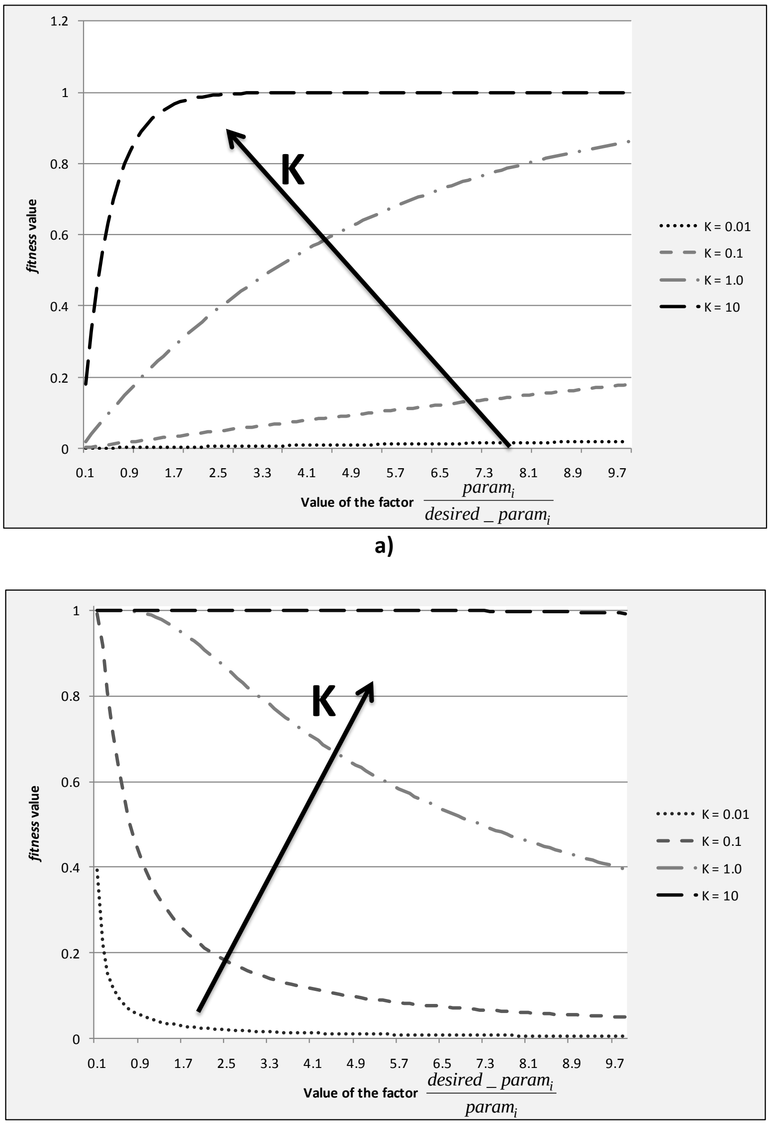
Considering (4.3), to maximize a design performance parameter, *param*$_i$. For demonstration purposes only, while maintaining constant the ratio and observing Figure 4-8 a), it is possible to visualize the effect of the variation of the *weight*$_i$ value in the format of the curve. Decreasing the value of *weight*$_i$, a larger value of *param*$_i$ (maximum) is needed to increase the value of the $\frac{param_i}{desired\_param_i}$ ratio and, consequently, to obtain a maximum value of fitness equal to 1.0.

In the case of minimizing a design performance parameter, *param*$_i$, the procedure is similar. While maintain constant the $\frac{desired\_param_i}{param_i}$ ratio, and observing Figure 4-8 b), decreasing the value of *weight*$_i$, a smaller value of *param*$_i$ (minimum) is needed to increase the value of the ratio and, consequently, to obtain a maximum value of fitness equal to 1.0.

In (4.5) the Gaussian's function argument, in the denominator, is augmented, which instigates *param*$_i$ parameter to converge to the target value, during optimization. This behavior is represented in Figure 4-9 by the 'K' arrow, which points the increasing direction of the *weight*$_i$ factor. Considering the example show in Figure 4-9, to obtain the maximum value of the fitness, the desired value of *desired_param*$_i$ is 5.0.

As previously explained, also in this case, it is possible to visualize the effect of *weight*$_i$ value. Here, the ratio value , $\frac{desired\_param_i - param_i}{desired\_param_i}$ in (4.5), is maintained and the effect is depicted in Figure 4-9. For instance, while increasing the value of *weight*$_i$, one should note that a value of *param*$_i$ closer to the target, *desired_param*$_i$, results in a value of fitness closer to 1.0.

As far as the author knows, this work uses, for the first time, exponential-based functions in the computation of the fitness of the individuals. This type of function, and the respective derivative, are continuous in time, which presuppose a better fitness result to help the optimization convergence. The theory to support this claim is currently being object of study. However, no proof is available yet.



a)

b)

Figure 4-8 Behavior of the factor *weight$_i$* in the maximize, a), and minimize, b), of fitness calculation

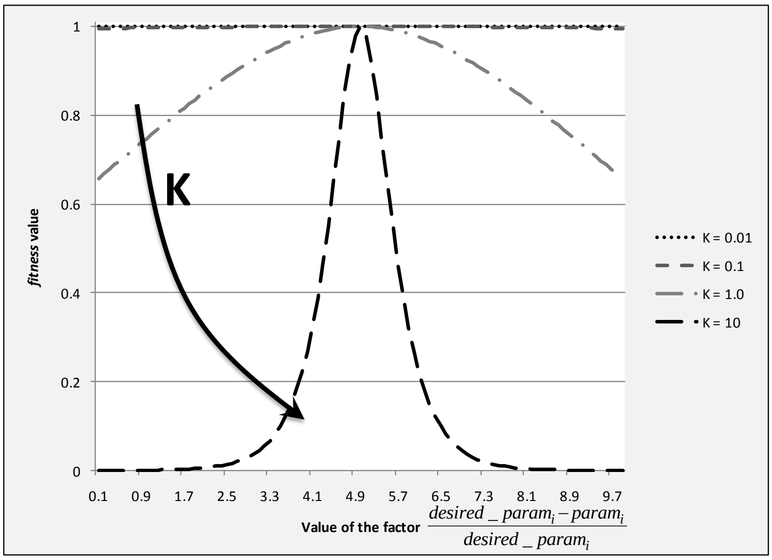


Figure 4-9 Behavior of the factor *weight$_i$* in the target fitness calculation

4.2.3 The selection scheme

Two options are available to select the individuals that will populate the new generation: with or without elitism. If elitism is selected a set of individuals are directly passed into the new population without changing the values of its genes. The other individuals are created by a combination of randomly selecting parents to crossed-over and mutation, afterwards.

Elitism is the process of selecting the best classified individuals and moving them unaltered into the new generation. This option enables that the best classified individuals remain unchanged through the generations providing good genetic material to reach the optimum result.

Continuing with individual selection, first, the population is ordered. Then, each individual is selected based on the probability that results from either the Roulette or the Rank methods.

In the Roulette method, the individuals are set with a normalized fitness, *fnorm$_i$*, between 0 and 1, according to (4.6)

$$fnorm_i = \frac{fitness_i}{\sum_i fitness_i} \tag{4.6}$$

The *fnorm$_i$* value represents directly the probability of each individual, *p$_i$*, to be chosen. Figure 4-10 depicts an example of the Roulette system probability.

The evolution of the algorithm tends to increase the best individual's fitness value, compared to the rest of the population. This produces an unbalanced Roulette system, naturally producing a high probability for the best classified individual to be selected again and again. Therefore, in the next stages of the algorithm, the same individual is selected most of the times, thus skewing the progress of the algorithm. Figure 4-11 shows an example of the unbalanced Roulette system.

Considering the Rank method, the individuals are ordered based on the value of their fitness. The relative position of the individual represents, afterwards, its probability of being selected, and the probability of each individual is represented by the normalized fitness value, according to (4.7)

$$fnorm_i = \frac{rank_i}{nind} \tag{4.7}$$

where *rank_i* represents the inverse order position and *nind* is the total number of individuals in the population. Figure 4-12 depicts an example of the Rank system ordering.

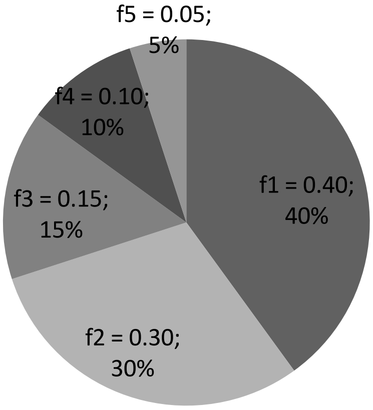


Figure 4-10 Example of a Roulette system for the case of 5 individuals

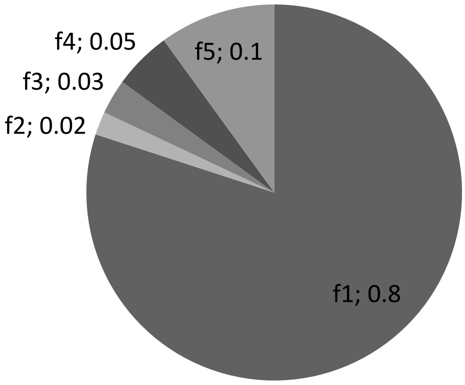


Figure 4-11 Example of an unbalanced Roulette system for the case of 5 individuals

The Rank method provides a linear method to distribute the probability of the individuals, which offers better results.

In both ordering methods, the fitness value is directly proportional to the probability of the individual being selected during the next steps of the optimization algorithm. After the initial individual selection, to create the new population, the remaining individuals are crossover, as explained in the next section.

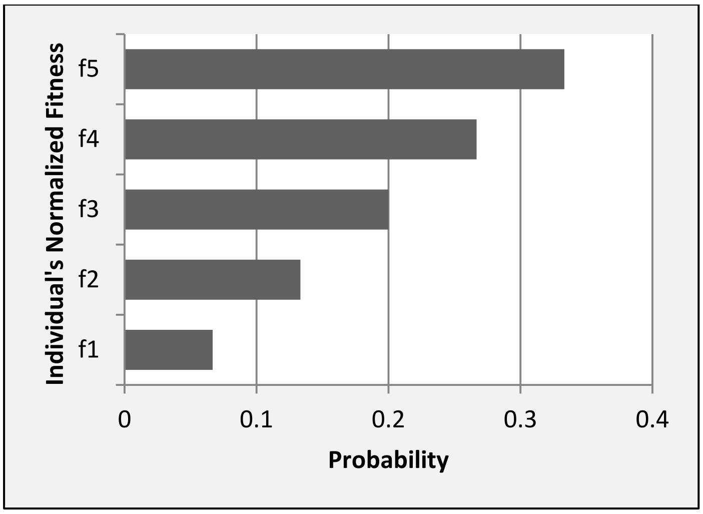


Figure 4-12 Rank system for 5 individuals

4.2.4 The crossover operator

To complete the population of the new generation, individuals are combined to produce off-springs with new genetic forms and values. The crossover is made by combining two individuals (parents) selected randomly, according to methods, and probability, previously described. Then, for each gene, g_i, the selected parents provide a portion of the value of the new (two) off-spring's genes.

As described in 4.2.1, each individual's gene is represented in binary format. At bit-level, a randomly chosen cross point is adopted and the crossover operator mixes bits from either parent's genes, from the corresponding part of the cross point, to compose the genes of the new off-springs. Figure 4-13 illustrates an example of the crossover operator.

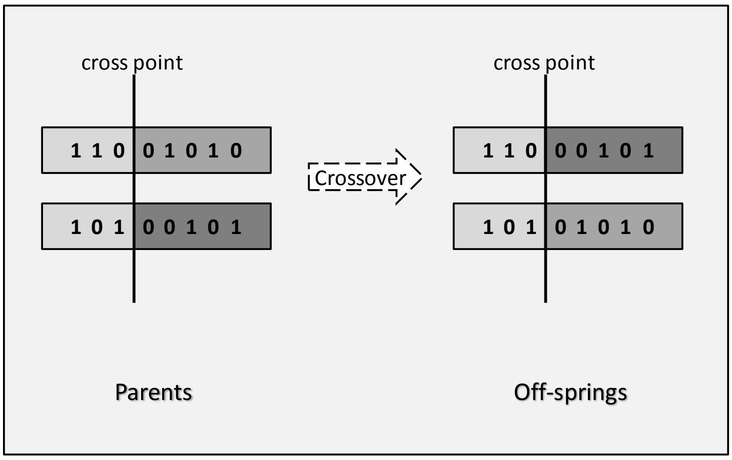


Figure 4-13 The crossover operator

4.2.5 The mutation operator

In order to provide a variation on the natural genetic material evolution, a mutation of a single bit of a gene might be applied.

First, the number of individuals to mutate*, nmut,* is randomly computed. Then, *nmut* individuals are, randomly, selected (except for the elite individuals). Every gene of the selected individual suffers then a mutation of a single bit, which is randomly selected with a given probability. Two possibilities exist at the stage of selecting bits to mutate, namely: a constant and equal probability for all bits or a variable probability. In the first case, all bits, at all time, are available with the same probability to be mutated. In the last case, the probability of one bit being mutated may vary according to the generation numbers. In the beginning, all the bits might be equally chosen to be mutated. As the number of generations increase, the most significant bits (MSBs) are left outside the set of bits that qualify to mutate. This prevents a significant variation of the gene's value, as the optimization process approaches the optimum point (end of optimization).

The later form of bit selection to mutate -variable probability- is given by (4.8)

$$m_i = (nbits - 1) - \frac{(nbits - 2) * z}{nger - 1} \tag{4.8}$$

where m_i is the number of bits available to be selected (to be mutated), starting from the least significant bit (LSB), *nger* represents the total number of generations, the *nbits* is the total number of bits of the gene, and z is the number of the present generation.

The mutation operator is illustrated in the Figure 4-14, where the 3rd MSB (bit) with value 0 is mutated to 1.

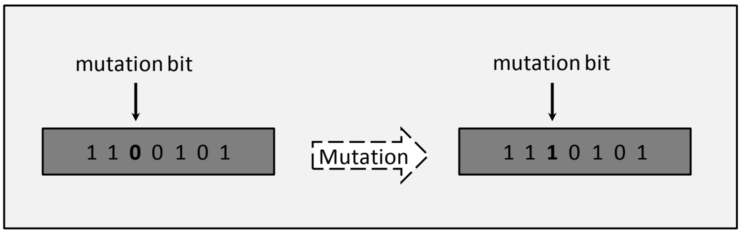


Figure 4-14 The mutation operator

4.3 Circuit Library

The next block of the platform is the evaluation of the circuit performance parameters. This block is separated from the main search algorithm section in order to enable the integration of multiple circuit libraries (e.g. different topologies of amplifiers), different circuit evaluation forms (e.g. time-domain or frequency-domain), various fabrication technologies (e.g. UMC[87] 130 nm, 65 nm, TSMC[88] 40 nm), diverse transistor models (e.g. BSIM3, BSIM4 [5]).

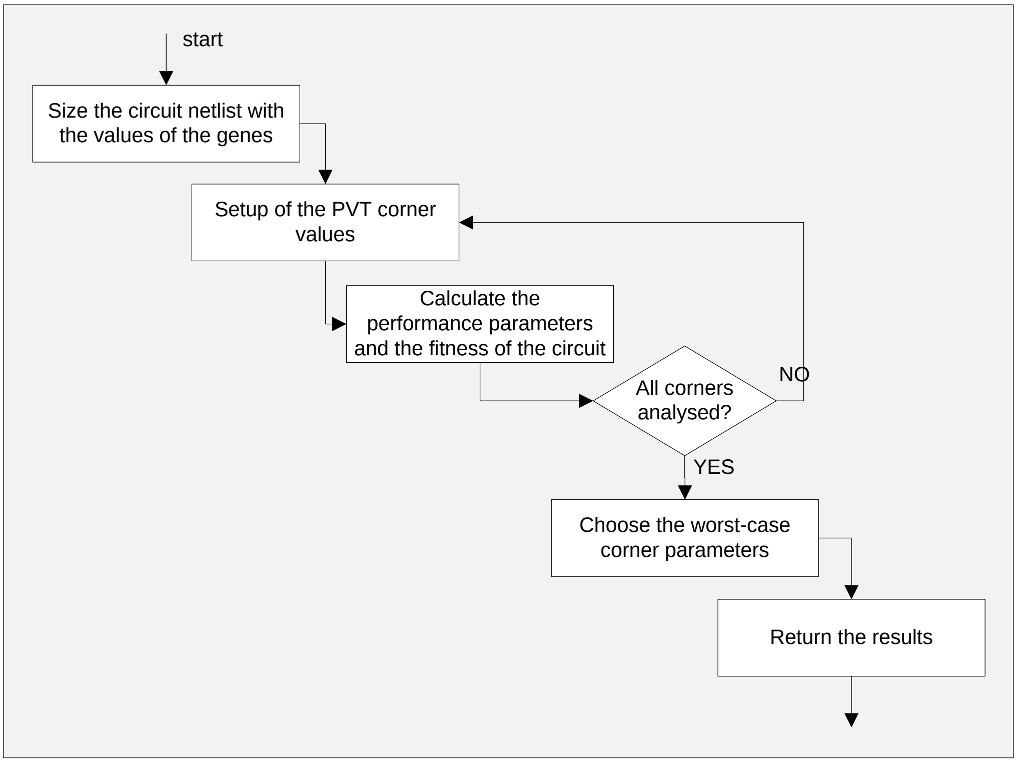

start
Size the circuit netlist with the values of the genes
Setup of the PVT corner values
Calculate the performance parameters and the fitness of the circuit
All corners analysed?
NO
YES
Choose the worst-case corner parameters
Return the results

Figure 4-15 Flowchart of the calculation of the an individual fitness

The main steps of this task are illustrated in Figure 4-15 and Figure 4-16. Figure 4-15 represents the manufacturing process, supply voltage and temperature (PVT) corners cycle evaluation, which requires the same number of circuit performance parameters evaluations (Figure 4-16).

The evaluation process starts with the correct circuit's netlist sizing. Depending on the circuit example, the sizing process is based on the individual's gene values that provide the size of the elements (e.g. *W* and *L* of the transistors) and all remaining circuit variables (e.g. currents and compensation capacitances) directly, or as a function of them. Then, for each PVT corner evaluation, the source code of the model of the elements is filled with the respective technology parameters, the voltage sources values are readjusted, the operating temperature is regulated, and, finally, the circuit performance parameters are computed.

At the end, the returned performance parameters are the corresponding ones that have originated the lowest (worst-case) fitness value, of all the corner evaluations.

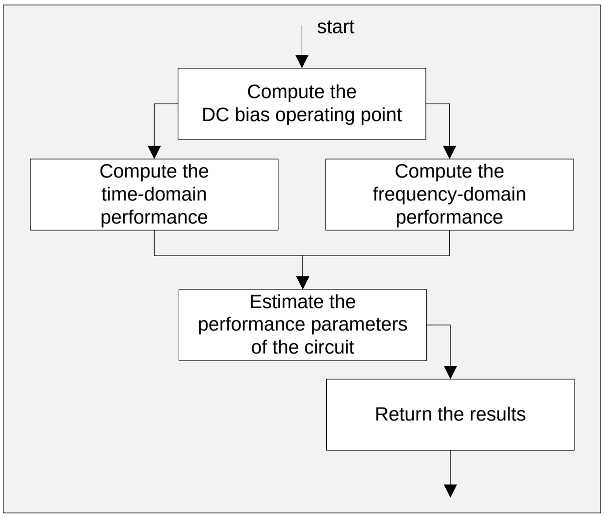

start
Compute the DC bias operating point
Compute the time-domain performance
Compute the frequency-domain performance
Estimate the performance parameters of the circuit
Return the results

Figure 4-16 Flowchart of a circuit evaluation

As depicted in the Figure 4-16, the calculation of the circuit performance parameters is preceded by the circuit analysis. The analysis starts with the DC bias operating point and small-signal parameters estimation for each device. These parameters include, for example, the drain current I_D, the threshold voltage, V_{TN} or V_{TP}, the saturation voltage V_{dsat}, the drain-source conductance g_{ds}, the transconductances *gm* and *gmb*, and all the parasitic capacitances (C_{db}, C_{gs}, C_{gd}, C_{sb}), based on the selected process corner level. Then, it continues with either the time-domain or frequency-domain analysis of the circuit. The circuit performance parameters are estimated using either an equation-based or an electrical simulation-based approach and considering also the selected PVT corner setup values. The results from the lowest fitness value PVT corner (worst-case corner) are returned to the main function of the search algorithm part of the optimization platform.

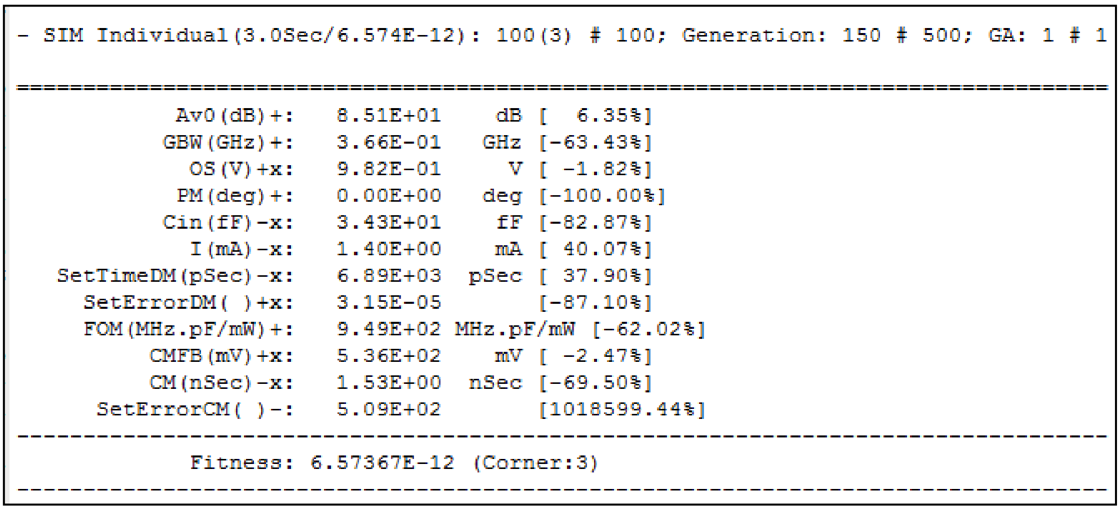
The main problem associated with this methodology is the large time required for the different circuit analyses. In order to minimize this problem, a distributed computing version of the search algorithm is considered and it will be described, later on, in section 4.6 of this chapter.

4.4 Highly Accurate Device Models

The evolution of circuit design is the increasing capacity of transistors integration in a single die, which means decreasing channel size (*W*, *L*) of the transistors. However, this originates that second and higher order short-channel effects that must be taken into account during the optimization process of circuits. As an example, it is difficult to determinate the r_{ds} value of the MOS transistor as a function of the transistor's drain-source voltage (v_{ds}). Therefore, it is almost mandatory to use advanced device simulation tools and accurate models in order to obtain acceptable results. As it will be explained later, it is possible to integrate any highly accurate device model, in the developed platform. For testing purposes in this work it was decided to integrate the model BSIM3v3 [5].

4.5 Exported Statistics and Results

The developed platform exports data during the optimization process and, at the end of the optimization process, it provides the optimum sized netlist of the circuit.



```
- SIM Individual(3.0Sec/6.574E-12): 100(3) # 100; Generation: 150 # 500; GA: 1 # 1

==================================================================================
            Av0(dB)+:   8.51E+01    dB [  6.35%]
           GBW(GHz)+:   3.66E-01   GHz [-63.43%]
             OS(V)+x:   9.82E-01     V [ -1.82%]
            PM(deg)+:   0.00E+00   deg [-100.00%]
           Cin(fF)-x:   3.43E+01    fF [-82.87%]
             I(mA)-x:   1.40E+00    mA [ 40.07%]
   SetTimeDM(pSec)-x:   6.89E+03  pSec [ 37.90%]
     SetErrorDM( )+x:   3.15E-05       [-87.10%]
     FOM(MHz.pF/mW)+:   9.49E+02 MHz.pF/mW [-62.02%]
          CMFB(mV)+x:   5.36E+02    mV [ -2.47%]
          CM(nSec)-x:   1.53E+00  nSec [-69.50%]
      SetErrorCM( )-:   5.09E+02       [1018599.44%]
----------------------------------------------------------------------------------
             Fitness: 6.57367E-12 (Corner:3)
----------------------------------------------------------------------------------
```

Figure 4-17 Example of an intermediate results printout

Throughout the sizing progression, at the end of each generation analysis, the platform displays the intermediate values achieved for each circuit performance parameter. Figure 4-17 illustrates an example of an intermediate results printout. Actually, the first line shown in Figure 4-17 is printed right after each individual evaluation. From the left to the right, the data printed is: the elapsed time during the individual processing;

the fitness achieved by the present individual; the number of the present individual, the worst-case PVT corner number; the total number of individuals; the number of the present generation; the total number of generations; the optimization run number; and the total number of runs to finish.

Next, there is a table that each line contains information of the performance parameters of the best classified circuit. The first column displays the name and units of the performance parameters. The center column shows displays the achieved values. The right column presents the percentage value of the difference between the achieved value and the desired value, of each performance parameter.

The last output line of information contains two values: the fitness value computed with the previous table performance's values; and the worst-case PVT corner selected.

These intermediate results are persisted in a text file, a Tab-Separated Values (TSV) formatted file[89]. Appendix A shows an example of the persisted data in a TSV file format. The persisted data permits the post-analysis of the optimization evolution and corresponding convergence.

At the end of the optimization process, the platform prints out the last intermediate results and also the optimum circuit netlist file compliant with the SPICE-like format. Appendix B shows an example of an exported circuit netlist.

4.6 Distributed Computing Version

As previously referred, one of the key improvements is the reduction of the processing time of the developed platform. This led to include distributed processing in order to efficiently evaluate the large number of individuals/circuits. Moreover, the concept behind the GAs is suitable to distributed/parallel processing. Hence, a distributed/parallel version of the platform was experimented, based on a standard Message Passing Interface (MPI), as proposed in [90].

The concept of the distributed genetic algorithm used in this work is based in a simple topology in which a master central computer controls the progress and execution of the GA, and the evaluation of the fitness of the population is realized in several remote slave computers, as shown in Figure 4-18. When compared to the centralized processing approach, described in the previous sections, this improvements of the architecture proposed achieves several new advantages, such as:

- the circuit evaluations are executed independently in parallel and in separate computers;
- in order to speed-up the optimization time, more computers can be added. These can have different hardware configurations and processing capabilities;
- it allows to use computers that are not 100% dedicated to the optimization engine, but still able to help the task, e.g., desktop computers;
- the hardware costs of a single multiprocessor machine, capable of running the optimization procedure in the same time are much higher than adopting this approach;
- due to the reduction of the optimization time, a larger population can be used in the GA, thus increasing the search capability within the design space by the algorithm and, therefore, maximizing the probability of finding a better final solution.

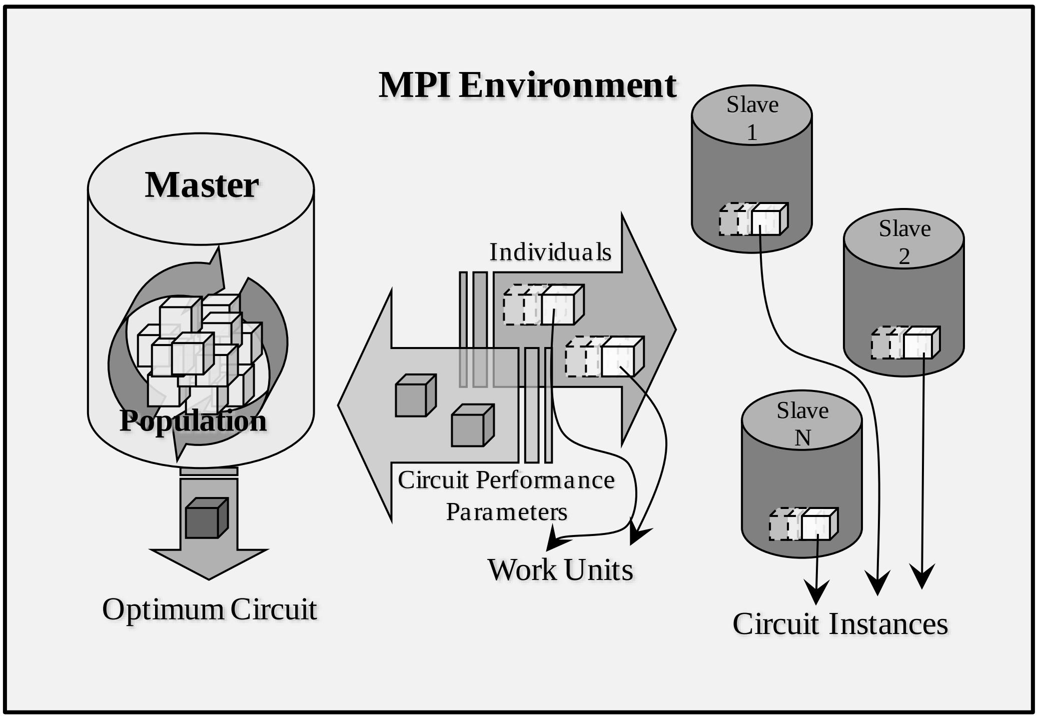


Figure 4-18 MPI Implementation of the distributed/parallel system

4.6.1 Classification, selection, crossover and mutation

Similarity to the previous version, the centralized processing, the algorithm starts by randomly generating a new population of circuits. Then it sends a set of new chromosomes to the slave computers, in which, an evaluation and classification of the individual is performed. This includes both, the circuit performance parameters and fitness computation, using the same approach as the centralized version. This information is returned back to the master computer that uses mutation and crossover operators randomly applied to selected elements of the current population, similarity to the centralized version. Moreover, a new population is then created for the next generation.

After finishing the set of circuit analysis (which comprises at least one circuit), the slave computer receives, from the master computer, a new set of circuits (one or more) to process. This procedure is repeated until the fitness of all the elements in the population is computed, until the last generation.

4.6.2 The master computer process

The master process, depicted in Figure 4-19 manages the genetic algorithm itself: it creates, distributes and receives all the data to and from the slave machines.

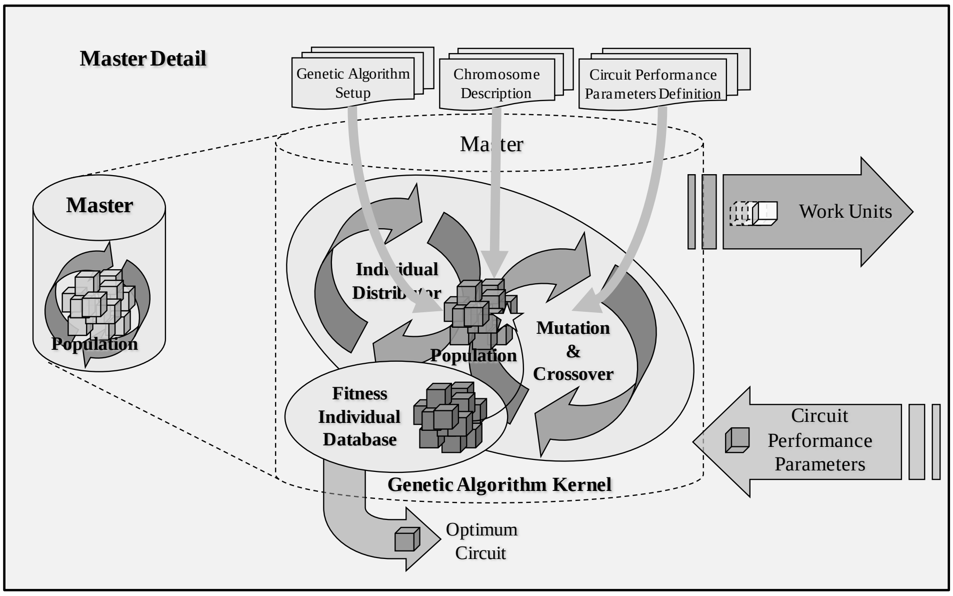


Figure 4-19 Master computer process of MPI implementation of the distributed/parallel system

Figure 4-20 gives an overview of the master process tasks. Based on the MPI configuration, it sets up a number of *nsla* slave machines and sends them work units to be processed. After, is the generation of the population of circuits, according to the chromosome description file (explained in 4.1.1). Then, for each generation, the following steps are repeated:

1. Distribution of one or more chromosomes through the slave machines as packages, work units, which are mapped to circuit netlists for analysis;
2. Evaluation of circuits, in the slaves;
3. Gathering the circuit results: the fitness and the circuit performance indicator's measures;
4. Apply selection, crossover and mutation operators to generate a new population;

The previous cycle continues until the stop criteria are reached, as in the centralized version. At the end, the netlist of the element of the last population with the best fitness is provided as an output.

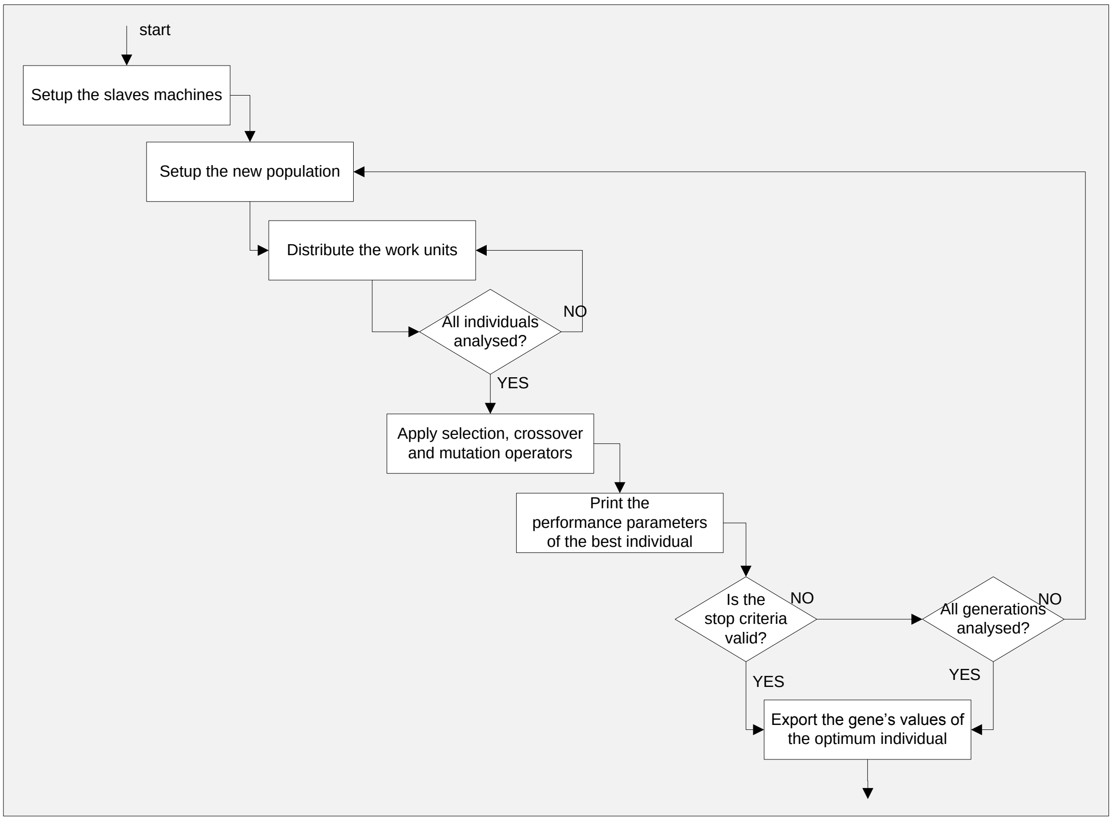

start
Setup the slaves machines
Setup the new population
Distribute the work units
All individuals analysed?
NO
YES
Apply selection, crossover and mutation operators
Print the performance parameters of the best individual
Is the stop criteria valid?
NO
YES
All generations analysed?
NO
YES
Export the gene's values of the optimum individual

Figure 4-20 Flowchart of the master process

4.6.3 The slave computer process

The slave process is, basically, the same software source-code package that implements the circuit analysis, as described in the centralized version as the "Circuit Library". A detailed overview of the slave process is show in Figure 4-21. As depicted in the Figure 4-21, a set of individuals is received within a package, work unit, and afterwards, each individual is processed.

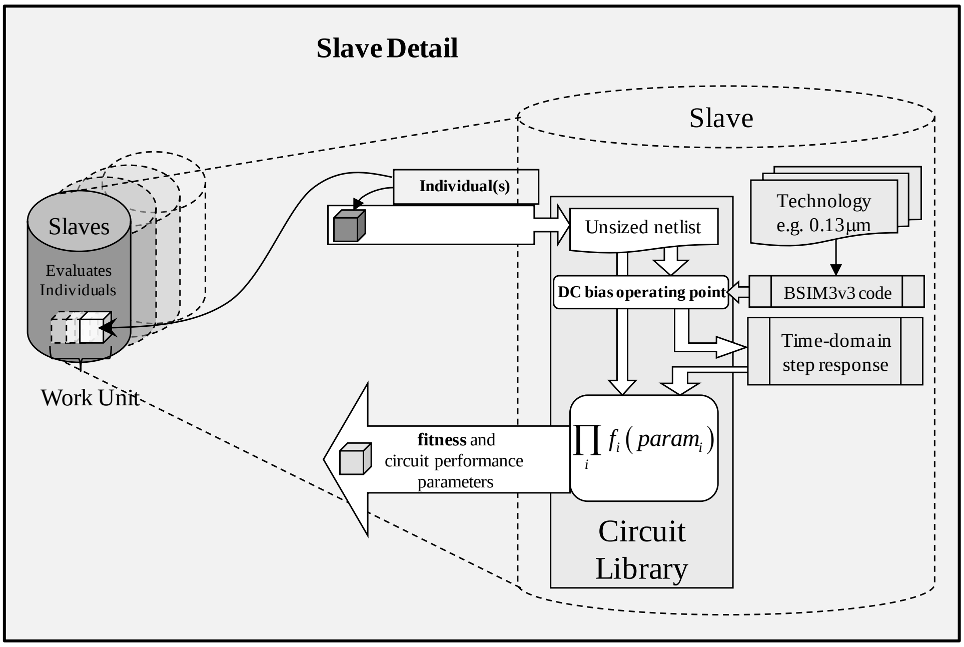


Figure 4-21 Slave processes from MPI implementation of the distributed/parallel system

The main steps of the slave process are illustrated in Figure 4-22. The first operation consists of unpacking the individuals sent as work units. These work units contain one or more individuals to be evaluated. Then, for each individual, it sizes the circuit netlist, computes the performance parameters values, in the various PVT corners, and the respective fitness, as in the centralized version. After the analysis of all individuals, packs the results and returns them to the master process.

In the same manner as in the centralized version, the calculation of the circuit performance parameters is preceded by the circuit analysis. The analysis starts with the DC bias operating point and small-signal parameters estimation for each device. These parameters include, for instance, the drain current I_D, the threshold voltage, V_{TN} or V_{TP}, the saturation voltage V_{dsat}, the drain-source conductance g_{ds}, the transconductances gm and gmb, and all the parasitic capacitances (C_{db}, C_{gs}, C_{gd}, C_{sb}), based on the selected process corner. Then, it continues with the time-domain or frequency-domain analysis of the circuit. Finally, the circuit performance parameters are estimated from the selected corner analyses results. The results from the lower fitness value corner are then returned.

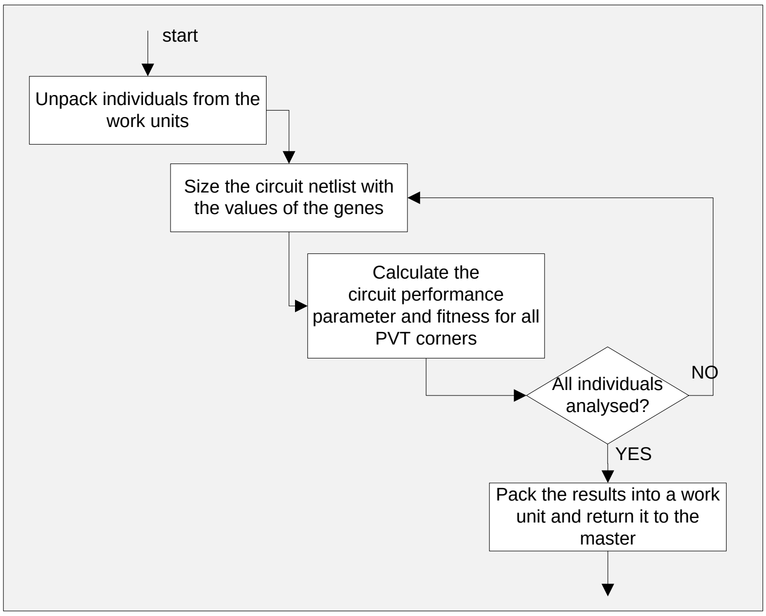

start
Unpack individuals from the work units
Size the circuit netlist with the values of the genes
Calculate the circuit performance parameter and fitness for all PVT corners
All individuals analysed?
NO
YES
Pack the results into a work unit and return it to the master

Figure 4-22 Flowchart of the slave process

In this work, each slave process can only evaluate one individual at a time because a single thread application is implemented. If using a multiprocessor slave machine several processes could run in parallel. In this case, the process parallelization would be handled by the MPI framework.

4.6.4 The message passing interface

The implementation of the mater-slave topology, as it was described in Figure 4-18, is based on the message passing interface (MPI)[90]. The MPI framework is based on the open-source concept and enables the communication on homogeneous and/or heterogeneous environments, between distinct machines. This way it is possible to extend the processing capacity over a network of processing machines, even if they are, technologically, different. Moreover, the adoption of MPI was based on its simplicity to build a simple time-domain optimization distributed environment, capable of launching and controlling multiple processes of intensive computation. Figure 4-23 presents the basic source functions used to parallelize an application.

Basically, the master process packs a number of individuals, as work units, using the *MPI_pack* function and sends this data to a slave process throughout the *MPI_send* function. After sending the work units to the slaves, it waits to receive the results from the slaves, with the *MPI_recv* function. Then, unpack the results, using the function *MPI_unpack*, and process the results. At the end of the optimization process, the function *MPI_finalize* is invoked to pop out of the distributed processing environment.

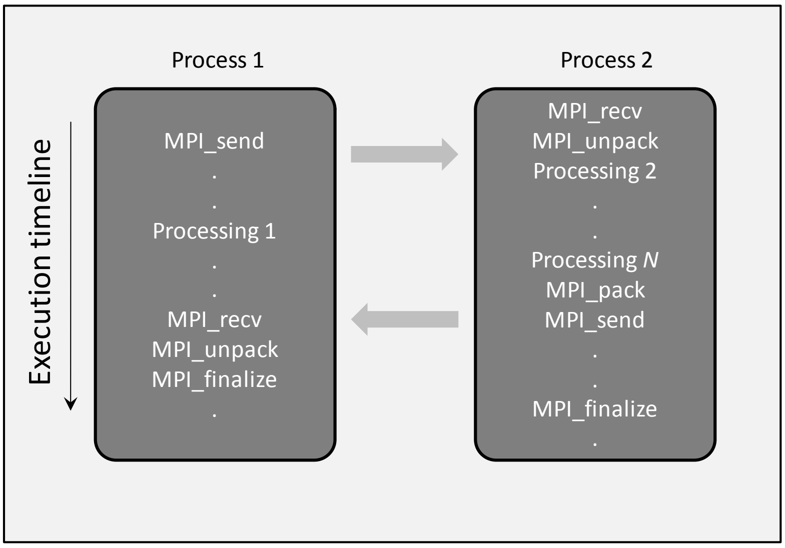
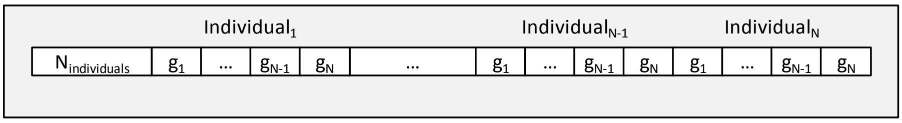


Figure 4-23 MPI basic functions

The slave, in the reverse order, invokes the same functions. It starts receiving the data throughout the function *MPI_recv* and unpacks the set of one or more individuals to be processed. After analyzing the set of one or more individuals, the slave packs the results data, *MPI_pack* and sends it back to the master using the function *MPI_send*. To end the collaborative work on the distributed processing environment, the slave invokes the function *MPI_finalize*. Usually, this last operation occurs at the end of the optimization process.

The two typical MPI messages format used in the implementation are:

1. the work unit sent from the master to the slaves, as shown in Figure 4-24, which contains the individuals to be evaluated,
2. the results from the slave to the master, as shown in Figure 4-25, which contains the maximum fitness (and the individual identification), the total execution time for the work unit sent, and the resulting values of the performance parameters of each individual sent, in the same order.



	Individual$_1$					Individual$_{N-1}$			Individual$_N$			
N$_{individuals}$	g_1	…	g_{N-1}	g_N	…	g_1	…	g_{N-1}	g_N	g_1	…	g_{N-1}	g_N

Figure 4-24 MPI message format: master to slaves

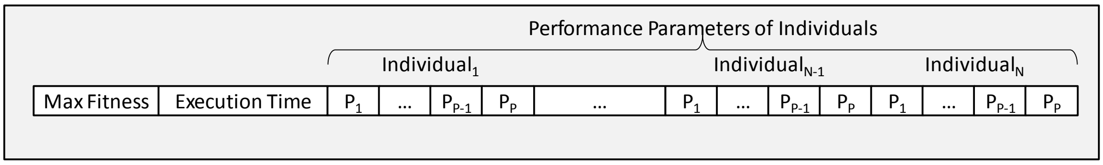


Figure 4-25 MPI message format: slave to master

4.6.5 Load distribution

MPI framework handles work load distribution, according to the slave computers performance. Initially, it distributes a preconfigured set of circuits, work unit (one or more individuals), to all slave computers, as shown in Figure 4-18. Then, the faster machines will continue to receive more work units to process while the slowest machine is still processing the previous distributed work unit. The evaluation finishes when the slowest slave machine concludes its last individual evaluation and has sent its results to the master.

4.6.6 Distributed/Parallel environment performance

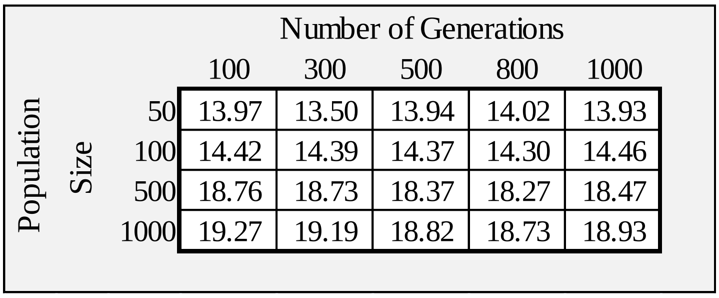
The algorithm performance tests were carried out using a cluster of computers with different hardware configurations: five Pentium 4@1.7 GHz, with one CPU logical core, named: pvm6, pvm7, pvm8, pvm9, pvm10; four Pentium 4@3.0 GHz, with two CPU logical cores, named: pvm1, pvm2, pvm3, pvm4 and one AMD Semptron @ 2.8 GHz (the master), with one CPU logical core, named pvm5.

To assess the performance of the parallel implementation *versus* the centralized version of the circuit optimizer, a speedup factor was defined (4.9)

$$speedup = \frac{T_{centralized}}{T_{parallel}} \qquad (4.9)$$

where $T_{centralized}$ is the time necessary to execute the optimization on a single machine (P4@3.0 GHz), with a centralized/sequential version of the genetic algorithm. The $T_{parallel}$ is the time necessary to execute the optimization by the distributed/parallel version.

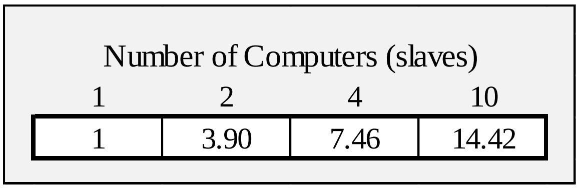
Figure 4-26 shows the speedup factor values, (4.9), for different combinations of the number of generations and population sizes. The results show that the speedup factor does not change considerably with the number of generations and it increases with the population size. This is expected because in each generation the individuals of the population are evaluated in parallel and the population of each generation is evaluated after the previous (in a centralized way). These examples were obtained using all the machines (10 computers) available in the cluster.



		Number of Generations				
		100	300	500	800	1000
Population Size	50	13.97	13.50	13.94	14.02	13.93
	100	14.42	14.39	14.37	14.30	14.46
	500	18.76	18.73	18.37	18.27	18.47
	1000	19.27	19.19	18.82	18.73	18.93

Figure 4-26 Speedup factor *versus* nr. of generations *versus* nr. of individuals

The speedup as function of the number of slave computers is shown next in Figure 4-27. In this test the following computers were used: 1 slave, pvm5; 2 slaves, pvm5, pvm1; 4 slaves, pvm5, pvm1, pvm2, pvm3. The 10 slaves test was conducted with all machines. These tests were executed with a population size of 100 individuals and for 100 generations.



Number of Computers (slaves)			
1	2	4	10
1	3.90	7.46	14.42

Figure 4-27 speedup factor *versus* nr. of computers (slaves)

The previous tests show that the parallel implementation of the genetic algorithm is much faster than the centralized implementation. Depending on the population size it can be up to 19 times faster if a set of 10 slave computers are used.

The speedup factor scales almost linearly with the increase of the number of slave computers, except when two computers are used, if compared to the case when used a single computer. The justification for that last observation is the fact that having only a single computer, it manages the GA and the Individuals evaluation. As opposed to the other scenarios, where there is a dedicated master to manage the GA, and one or more slaves to execute the evaluation.

Furthermore, a super-linear behavioral is observed in the speedup factor values. This can be justified by the fact that some individuals are not fully analyzed/evaluated.

The evaluation of each individual is abandoned, for example, if the DC bias operating calculation fails. Given this, not all individuals required the same amount of computation resources. The number of individuals, which the evaluation is early abandoned, is random. Also, the different processing frequencies of the set of the computers used and the availability of hyper-threading technology[2] might influence the results.

4.7 Conclusions

This chapter presented the developed software platform that permits demonstrate the efficiency of the proposed methodology for optimization of amplifiers, based in the time-domain (or, optionally, in the frequency-domain as well).

The source-code implementation in several separated software modules facilitates future integration of other software modules, such as different transistor model souce-code, other type of circuit evaluation library, or a distinct search algorithm. This genetic algorithm includes a stop criteria based on the variation of the circuit performance parameters. This prevents the algorithm to continue to run, generation after generation, and little or no better result is reached. Moreover, classification process of the algorithm is based on exponential functions which are continuous as well as the respective derivatives. As it is claimed, this feature might be an important factor to help the optimization convergence.

To enhance the platform performance, the processing paradigm was upgraded to employ distributed processing, based on the MPI. This framework achieves a considerable reduction in the optimization time and the increasing processing capacity allows searching within a larger design space using complex transistors models, e.g. BSIM3v3, consequently, yielding more accurate results. The optimization, based on transient simulations, was only possible due to the integration of both, the genetic algorithm kernel and the open-source source-code simulator NGSPICE. The distributed version also permits to reuse hardware such as old desktop computers.

[2] Intel® Hyper-Threading Technology (Intel® HT Technology)¹ uses processor resources more efficiently, enabling multiple threads to run on each core

5 Practical Design Examples and Silicon Results

This chapter illustrates the practical usability of the optimization methodology and platform developed within this work. Four different amplifier topologies were analyzed and then optimized in order to meet certain specifications, and their respective results are shown. Each one of these examples evidences the strengths of this work. The complexity of each example increased as the platform was being improved and tested. The last example is the most complete and complex. In this case a designer helped inventing a novel amplifier topology and made a frequency-domain analysis, which was then optimized using the developed platform. Furthermore, the same final example was also optimized on time-domain, for comparison purposes.

The first example presents the optimization and results of a low-voltage two-stage cascode-compensated opamp with enhanced performance. At circuit level, it is shown how to add an additional degree-of-freedom to the conventional topology, which allow to obtain, simultaneously, high open-loop gains and fast settling responses without increasing the power dissipation.

The second example explores the optimum sizing and compensation of two-stage amplifiers based on a time-domain approach. The selected topology includes three compensation capacitors, increasing the complexity of the transfer function a higher complex level of analysis. It is demonstrated, with consistent simulated results, that the optimum step-response is achieved using hybrid cascode-compensation comprising two unequal sized capacitors.

Further increasing the complexity of the topology of the second example, the 3rd example shows the design and optimization results of a two-stage amplifier employing gain-boosting techniques, with the transfer function order augmented to 8th order. This example demonstrates the optimum sizing and capacitor compensation schema.

Finally, the last case presents (to the best of the authors' knowledge) a novel two-stage fully-differential CMOS amplifier completely self-biased. It comprises two self-biased inverter stages with optimum compensation and high efficiency. Although it relies on a class A topology, it is shown through simulations, that it achieves the highest efficiency of its class and comparable to the best class AB amplifiers. Due to the self-biasing, a low variability in the DC gain over process, temperature, and supply is achieved. A prototype in a standard CMOS technology was fabricated and the experimental results show that a good energy-efficiency is achievable. A comparison among state-of-the-art amplifiers is then presented at the end.

5.1 Cascode Amplifier with Active-Biasing

This example presents a methodology, optimization and simulation results of a low-voltage two-stage cascode-compensated amplifier with enhanced performance.

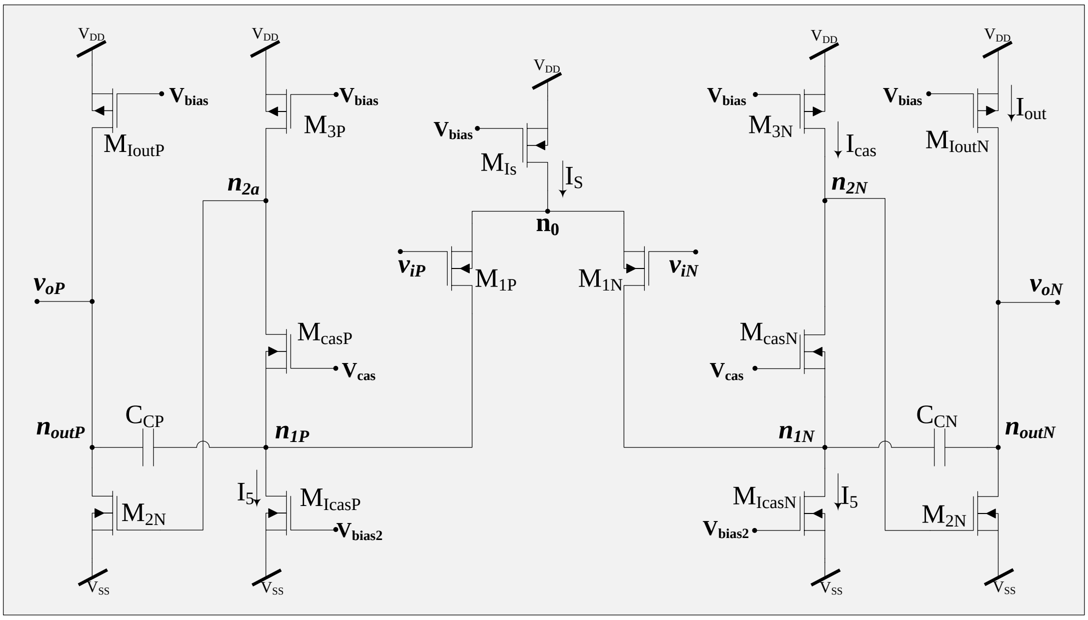


Figure 5-1 Schematic of a conventional low-voltage two-stage amplifier

At circuit level, it is shown how to add an additional degree-of-freedom to the topology that allowed reaching, simultaneously, high open-loop gains and fast settling responses without increasing the power dissipation. Figure 5-1 illustrates the basic topology of a low-voltage two-stage cascode-compensated amplifier.

5.1.1 Circuit insight

As explained in [65], the compensating capacitors ($C_{C\{N,P\}}$) are connected to the sources of the additional cascode devices (nodes $n_{1\{N,P\}}$) decoupling the gates of the transistors of the output-stage ($M_{2\{N,P\}}$). This technique (Ahuja)[65] can significantly improve the bandwidth over conventional Miller compensation, improve the high-frequency power-supply-rejection-ratio and moves the right-half plane (RHP) zero resulting from the Miller compensation into high frequencies.

Using the methodology described in chapter 3, the 3rd order transfer function of the amplifier is extracted. First, the linearization procedure (section 3.4.5) applied to half of the amplifier circuit results into the small signal equivalent (differential-mode, DM) of the amplifier. Next, the theory of the Y-parameters (section 3.4.6) allowed to isolate the nodes of the small signal equivalent. Then, the behavioral signal path model (BSP) [70] (section 3.3) of the circuit shown in Figure 5-2 is extracted. From the BSP model results the following transfer function of the amplifier:

$$H(s) = \frac{N_3 \cdot s^3 + N_2 \cdot s^2 + N_1 \cdot s^1 + N_0}{D_3 \cdot s^3 + D_2 \cdot s^2 + D_1 \cdot s^1 + D_0} \tag{5.1}$$

The complete transfer function of the amplifier is several pages long and for simplicity reasons it is decided not to show it here. Independently of its size, of the expression code, it is copied to the source code of the respective circuit library (section 4.3), to be integrated in the optimization platform.

From (5.1) the overall DC gain, A_{OL}, was determined, and can be approximated by:

$$A_{OL} \approx \frac{g_{m1}}{(g_{ds3} + \left(g_{dsIcas} + g_{ds1}\right) \cdot (g_{dsCAS} / g_{mCAS}))} \cdot \frac{g_{m2}}{\left(g_{dsIout} + g_{ds2}\right)} \tag{5.2}$$

Basically, the A_{OL} is defined by the product of the gain of both stages Note that the gain of the folded-input stage is primarily limited by the drain-source conductance of the PMOS current-sources M_3, g_{ds3}.

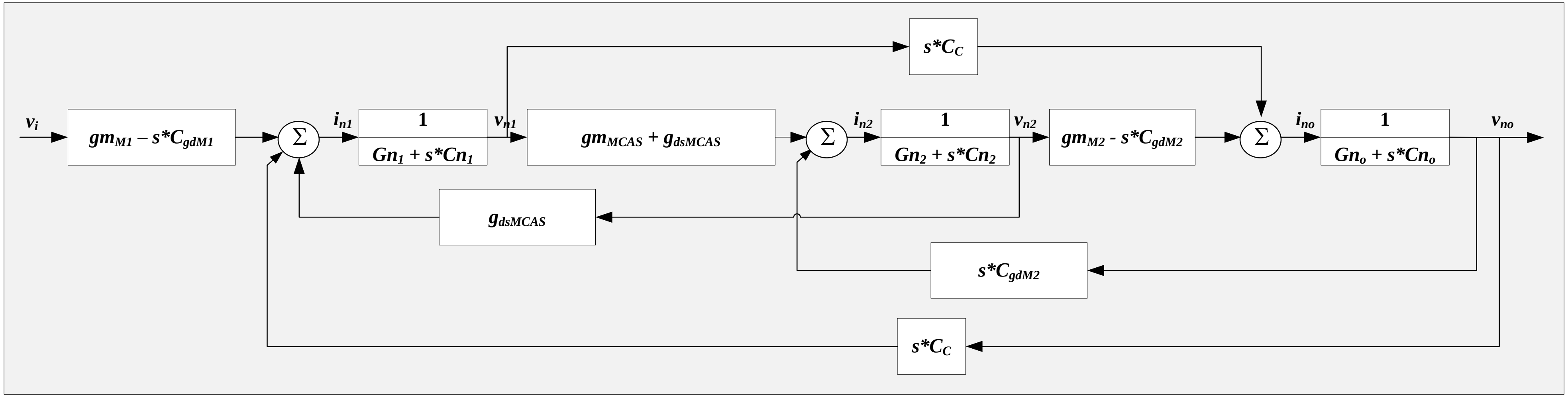


Figure 5-2 Behavioral Signal Path of half of the two-stage cascode amplifier shown in Figure 5-1

$$Gn_1 = g_{dsM1} + g_{dsMICAS} + g_{dsMCAS} + gm_{MCAS} \tag{5.3}$$

$$Cn_1 = C_{dbM1} + C_{gdM1} + C_{gdMICAS} + C_{dbMICAS} + C_{gsMCAS} + C_C \tag{5.4}$$

$$Gn_2 = g_{dsMCAS} + g_{dsM3} \tag{5.5}$$

$$Cn_2 = C_{dbMCAS} + C_{gdMCAS} + C_{dbM3} + C_{gdM3} + C_{gsM2} + C_{gdM2} \tag{5.6}$$

$$Gn_o = g_{dsM2} + g_{dsMIout} \tag{5.7}$$

$$Cn_o = C_{dbM2} + C_{gdM2} + C_{dbMIout} + C_{gdMIout} + C_c \tag{5.8}$$

Concerning the closed-loop transfer function of the circuit, this topology has a third-order transfer function with two zeros (at both left, LHP, and right half-plane, RHP, at high frequencies) and three poles. The poles follow the polynomial characteristic, *D(s)*:

$$D(s) = \beta \cdot (s + \omega_{cl}) \cdot \left(s^2 + 2\zeta\omega_n \cdot s + \omega_n^2\right) \tag{5.9}$$

where ω_{cl}, ω_{n} and ζ represent, respectively, the closed-loop real-pole frequency (which depends on the feedback factor, β), the natural frequency and the damping factor of the conjugated pole pair.

The remaining performance parameter needed to compute the fitness of the circuit are obtained using the definitions in section 3.5 of chapter 3. The output swing (OS) of this topology is limited only by the drain-source saturation voltages of the output transistors and it is defined by:

$$\mathrm{OS} \approx V_{DD} - V_{dsatM2} - V_{dsatMIout} - V_{m\arg in} \tag{5.10}$$

where V_{DD} and V_{margin} are, respectively, the positive supply-voltage and some additional safety margin to guarantee proper saturation of the output devices (~100 mV). The total power dissipated by this topology, P_{total}, is given by (excluding the CMFB and the biasing circuitry):

$$P_{total} \approx V_{DD} \cdot I_{total} = V_{DD} \cdot \left(I_S + 2 \cdot I_{OUT} + 2 \cdot I_{CAS}\right) \tag{5.11}$$

5.1.2 Adding a degree-of-freedom in a two-stage amplifier

Observing (5.2), one can conclude that the most efficient way of increasing the DC gain (A_{OL}) of the conventional topology is to decrease conductance g_{ds3}, which can be achieved simply by reducing the biasing current I_{CAS}. For example, reducing this current by a factor of 2 or 4 a gain increase of 6 dB (or 12 dB) can be obtained. However, reducing this current also results in a reduction in the current passing through cascode devices (M_{CAS}), degrading the frequency of the poles associated with node n_1 and, consequently, degrading the frequency-response of the amplifier. To avoid this degradation, an active biasing technique can be applied to the cascode devices as illustrated in Figure 5-3. A similar technique was previously proposed in [91] but applied to single-stage folded-cascode topologies. Adding a single fully-differential auxiliary Operational Transconductance Amplifier (OTA), the frequency of the pole associated with the source of M_{CAS} can be increased due to the local negative-feedback mechanism, since the impedance seen at the source of the cascode transistors is reduced by a factor of (1+A), where A represents the gain of the OTA. Remembering that current I_{CAS} was reduced by a certain factor for gain enhancement, the saved current can be used in the design of the auxiliary OTA without adding extra power dissipation. The auxiliary OTA, outlined in Figure 5-4, can be realized by a single-stage folded-input amplifier with transistor M_{1Z} implementing the common-mode feedback circuit as proposed in [92]. For achieving low-gains, A, and wide bandwidths, transistors $M_{5[x,y]}$ of the auxiliary OTA, can be biased in the boundary of the triode region.

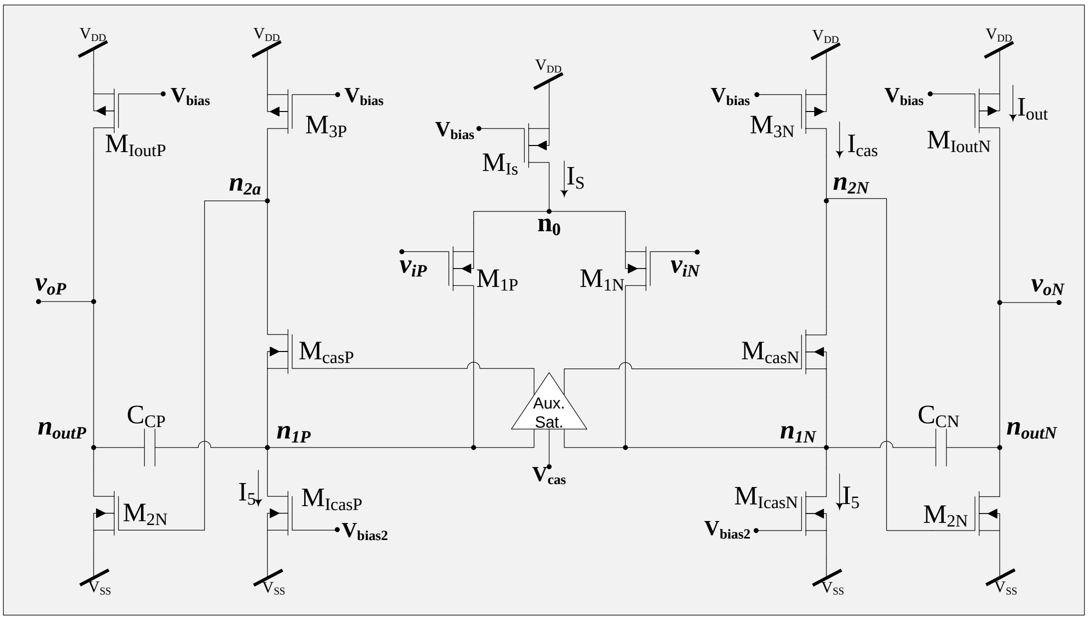


Figure 5-3 Schematic of a two-stage cascode amplifier with regulated active-biasing

Active-cascode biasing was previously employed for gain-enhancement [92][93][49] but with a different perspective. In these references, the idea is to use the local negative-feedback mechanism to enhance the DC gain of folded-cascode OTAs by boosting the output impedance of the cascode devices by the same factor (1+A). In the low voltage amplifier of Figure 5-3 this would not work because device M_3 is not being used together with a PMOS cacode device thus limiting the gain of the input stage. It is known that the auxiliary OTA introduces a closely spaced pole and zero (doublet), which can seriously degrade the settling behavior due to an additional slow-settling component[49].

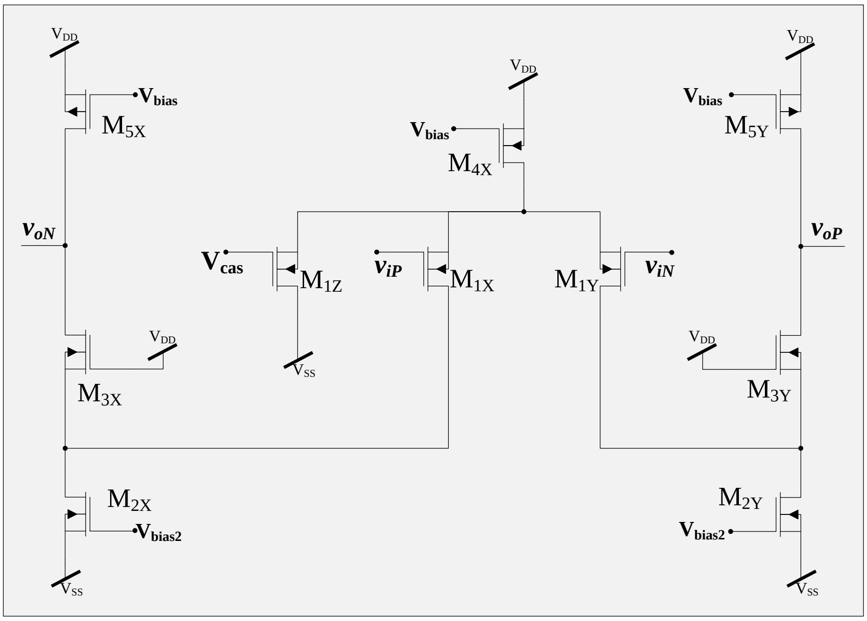


Figure 5-4 Schematic of an auxiliary folded-input OTA with CMFB transistor, *M_{1Z}*

Similarly, as in the case of the conventional topology (Figure 5-1), the BSP model of the proposed topology (Figure 5-3), with the auxiliary active-biasing amplifier (Figure 5-4), is created and is shown in the Figure 5-5. From this BSP model the amplifier the complete 5rd order transfer function can be calculated, alike the one presented next:

$$H(s)=\frac{N_5\cdot s^5+N_4\cdot s^4+N_3\cdot s^3+N_2\cdot s^2+N_1\cdot s^1+N_0}{D_5\cdot s^5+D_4\cdot s^4+D_3\cdot s^3+D_2\cdot s^2+D_1\cdot s^1+D_0} \qquad (5.12)$$

Again, the complete transfer function of the amplifier is several pages long and again, for simplicity reasons, it was decided not to present it here. Once more, the expression code is directly inserted into the source code of the respective circuit library (section 4.3), and, automatically, integrated in the optimization platform.

5.1.3 Design procedure and circuit optimization

Regarding the equation (5.9) some important questions remain:

a) which are the optimal values for ω_{cl}, ω_n and ζ?

b) where should the zeros of the transfer function be located in the s-plane?

c) how to treat transfer functions with doublets and/or with more than three poles?

The explicit answers to these questions can be avoided if the optimization of the amplifier is performed in the time-domain.

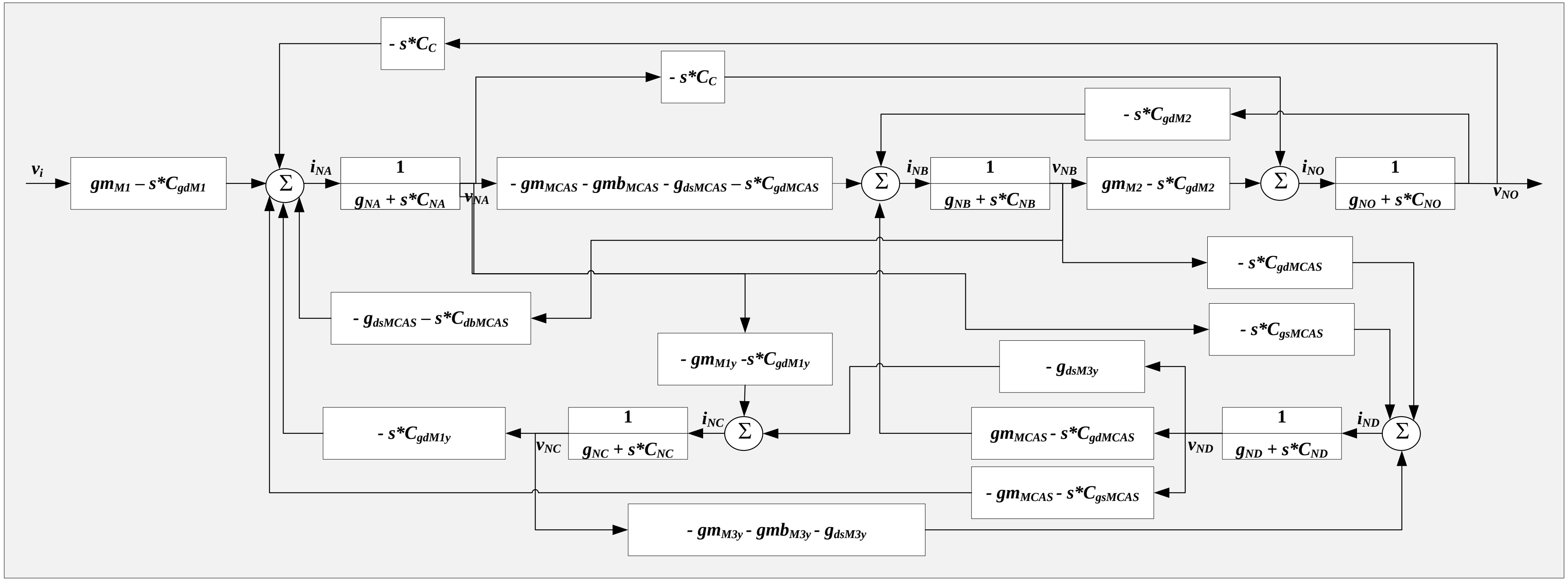

v_i
$gm_{M1} - s{*}C_{gdM1}$
Σ
i_{NA}
$\frac{1}{g_{NA} + s{*}C_{NA}}$
v_{NA}
$- gm_{MCAS} - gmb_{MCAS} - g_{dsMCAS} - s{*}C_{gdMCAS}$
Σ
i_{NB}
$\frac{1}{g_{NB} + s{*}C_{NB}}$
v_{NB}
$gm_{M2} - s{*}C_{gdM2}$
Σ
i_{NO}
$\frac{1}{g_{NO} + s{*}C_{NO}}$
v_{NO}
$- s{*}C_C$
$- s{*}C_C$
$- s{*}C_{gdM2}$
$- s{*}C_{gdMCAS}$
$- s{*}C_{gsMCAS}$
$- g_{dsMCAS} - s{*}C_{dbMCAS}$
$- gm_{M1y} -s{*}C_{gdM1y}$
$- g_{dsM3y}$
i_{NC}
Σ
$\frac{1}{g_{NC} + s{*}C_{NC}}$
v_{NC}
$- s{*}C_{gdM1y}$
$gm_{MCAS} - s{*}C_{gdMCAS}$
v_{ND}
$\frac{1}{g_{ND} + s{*}C_{ND}}$
i_{ND}
Σ
$- gm_{MCAS} - s{*}C_{gsMCAS}$
$- gm_{M3y} - gmb_{M3y} - g_{dsM3y}$

Figure 5-5 Behavioral Signal Path of half of the circuit cascode amplifier with active-biasing shown in Figure 5-3

$$g_{NA} = g_{dsM1} + g_{dsMICAS} + g_{dsMCAS} + gm_{MCAS} + gmb_{MCAS} \tag{5.13}$$

$$C_{NA} = C_{dbM1} + C_{gdM1} + C_{gdMICAS} + C_{dbMICAS} + C_{dbMCAS} + C_{gsMCAS} + C_{sbMCAS} + C_{gdM1y} + C_{gsM1y} + C_C \tag{5.14}$$

$$g_{NB} = g_{dsMCAS} + g_{dsM3} \tag{5.15}$$

$$C_{NB} = C_{dbMCAS} + C_{gdMCAS} + C_{dbM3} + C_{gdM3} + C_{gsM2} + C_{gdM2} \tag{5.16}$$

$$g_{NC} = g_{dsM1y} + g_{dsM2y} + g_{dsM3y} + gm_{M3y} + gmb_{M3y} \tag{5.17}$$

$$C_{NC} = C_{dbM1y} + C_{gdM2y} + C_{dbM2y} + C_{gsM3y} + C_{sbM3y} + C_{gdM1y} \tag{5.18}$$

$$g_{ND} = g_{dsM5y} + g_{dsM3y} \tag{5.19}$$

$$C_{ND} = C_{dbM3y} + C_{gdM3y} + C_{dbM5y} + C_{gdM5y} + C_{gsMCAS} + C_{gdMCAS} \tag{5.20}$$

$$g_{NO} = g_{dsM2} + g_{dsMIout} \tag{5.21}$$

$$C_{NO} = C_{dbM2} + C_{gdM2} + C_{dbMIout} + C_{gdMIout} + C_C \tag{5.22}$$

The main advantage of the time-domain optimization is that the only specification to consider is the settling-time for a given accuracy. Recalling chapter 4, when a given settling-error is reached within a desired settling-time it is automatically guaranteed that the amplifier has enough open-loop gain, A_{OL}, closed loop bandwidth and closed loop stability.

Focusing on the time-domain optimization, the chromosome used in the genetic algorithm (GA) is depicted in Figure 5-6. The genes of each chromosome are, basically: the biasing current, I_S; the biasing currents mirroring factors, mI_{CAS} and mI_{OUT}; the saturation voltages, V_{dsati}, and the channel lengths L_{Mi}, for each transistor; and the compensating capacitor, C_C.

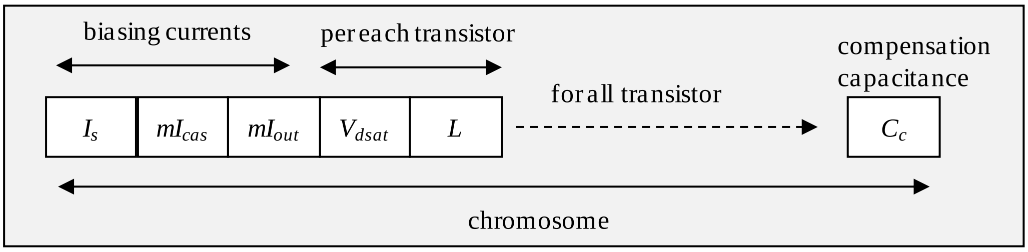


Figure 5-6 Format of the cascode amplifier with active-biasing chromossome

The channel width, W, of the transistors of the main circuit (Figure 5-3), are then computed straightforwardly, using the level 2 equation of the drain current of the transistor (3.17) in which the values of I_D, V_{dsat} and L are selected from each individual's chromosome and the $K_{N,P}$ is a constant derived from the technology parameter. The N and P distinguish the NMOS and PMOS channel-type constants, respectively, K_N and K_P. The values of the drain currents of the M_{CAS} and M_{Iout} transistors are derived from the I_S current, using the *mIcas* and *mIout* constants, respectively. The sizes of biasing transistors are computed, analogously as the main amplifier transistors, considering current mirroring factor of 1/10. This current consumption reduction still provides the correct biasing for the circuit. The sizes of the transistors of the auxiliary amplifiers are a direct mapping of the ones in the main amplifier, but scaled down by a factor of 4. Yet again, this ensures a power dissipation reduction and still keeping the design in the stability region.

The fitness function comprises the evaluation of the settling-time (ST), output-swing (OS) and of the total current consumption (I_{total}) according to:

$$f(\mathrm{ST},\mathrm{OS},I_{total}) = \left(1-e^{-\frac{\mathrm{ST\_desired}}{\mathrm{ST\_achieved}}}\right)\cdot\left(1-e^{-\frac{\mathrm{OS\_achieved}}{\mathrm{OS\_desired}}}\right)\cdot\left(1-e^{-\frac{I_{total}\_desired}{I_{total}\_achieved}}\right) \tag{5.23}$$

Equation (5.23) is maximized when ST and I_{total} are minimized and OS is maximized. OS is computed with (5.11) and I_{total} value is given by (5.10). The ST is computed based on the step response derived from the transfer function of the amplifier in (5.12), according to the method described on chapter 4. Previously, the DC bias operating point is estimated by a single (.op) simulation analysis. Moreover, the circuits that have transistors (main circuit and auxiliary amplifiers) out of the saturation operating region ($V_{DS} < V_{dsat}$ + 50 mV) are classified with a very low fitness (i.e. 1.0-e12). This way, a low probability of being selected to participate in the next population is given to the individual. Still, the genetic material of the individual is not, directly, discarded and it may be used in the generation of the new population process.

The target fabrication technology is a 350 nm CMOS technology (L_{min} = 350 nm). The mobility and threshold parameters level 2, K_N, K_P, V_{TN} and V_{TP} parameters of the devices are, respectively, 155 μAV^{-2}, 50 μAV^{-2}, 0.52 V and -0.65 V. This circuit was designed to operate with a supply voltage of 1.5 V and to be used in a front-end Sample-and-Hold (S&H) of a 10-bit 20 MS/s Pipeline ADC (with a unity feedback and a total loading capacitance, C_{LOAD}, of about 1.5 pF). The desired ST is less than 25 ns for an accuracy of about 0.1% (corresponding to an error smaller than 2 mV assuming a differential reference voltage of 2 V).

5.1.4 Post-optimization and simulation results

The optimum sizing of the topologies (optimum netlists) achieved by the platform are then verified by electrical verification, using HSPICE.

Table 5-1 provides a comparison of the results for the conventional topology (Figure 5-1). It shows the desired specifications used in the fitness function, the achieved values provided by the optimizer and the simulated results of the conventional circuit. The conventional topology (Figure 5-1) did not meet the specification for the settling-accuracy. In fact, it settles in 21 ns but with an error of more than 0.25% due to the insufficient gain.

Table 5-2 shows the specifications and the results values of the performance parameters of the proposed topology. Moreover, during the optimization process, the values of these parameters were being compared, in the fitness computation, with the desired specifications. The simulated results allow validating the optimum result obtained. With the proposed topology it is possible to reach, simultaneously, high DC gain (66.7 dB) and a good settling-time response (19 ns@0.1%) for about the same current consumption (3.15 mA).

Table 5-1 Optimized and post-simulated results for the conventional topology shown in Figure 5-1

	Desired specifications	Optimized Results	Simulated Results
A_{OL}	-	-	51.2 dB
OS	1.1 V	1 V	1 V
I_{total}	3 mA	3.15 mA	3.15 mA
ST	25 ns@0.1%	20 ns@0.25%[(a)]	21 ns@0.25%[(a)]

(a) the conventional topology never reached the 0.1% relative error due to insufficient gain

Table 5-2 Optimized and post-simulated results for the proposed topology shown in Figure 5-3.

	Desired specifications	Optimized Results	Simulated results
A_{OL}	-	-	66.7 dB
OS	1 V	1 V	1 V
I_{total}	3 mA	3.15 mA	3.15 mA
ST	25 ns@0.1%	18 ns@0.1%	19 ns@0.1%

The electrical simulation plots are shown. Figure 5-7 shows the zoom of a transient simulation of the complete S&H circuit when an input-step of 1 V is applied (step starts at 550 ns) and performed for three different cases. The 1st case corresponds to the simulation of the conventional topology (Figure 5-1) after optimization. As can be observed, the simulated settling-time is about 21 ns[(3)] but the accuracy is worse than 5 mV (0.25%). As previously explained, when the specifications are very stringent, this topology is not capable of reach, simultaneously, high DC gain and reduced ST. The lack of DC gain can be observed in Figure 5-8 (1st case) in which a frequency-domain simulation is shown.

[3] Although the level of the signal lowers, it does not decrease beyond the error margin. The settling, for the given error, is marked correctly.

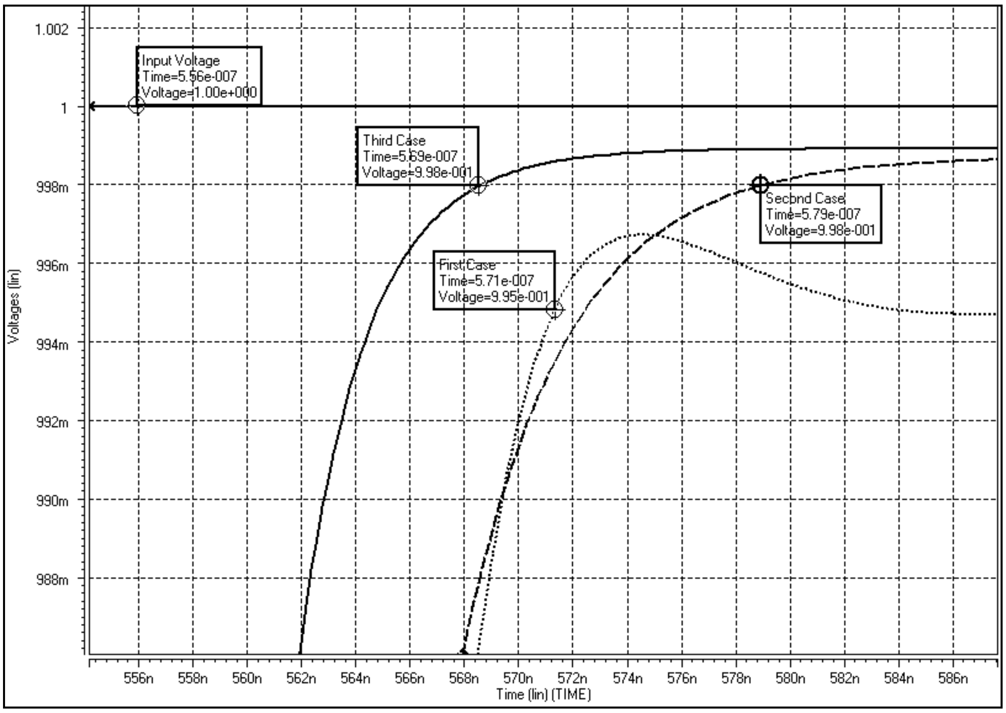


Figure 5-7 Zoom of the simulated settling-response of the conventional topology, Figure 5-1 (1st case), of the conventional topology with I_{CAS} reduced by a factor of 4 (2nd case) and of the proposed topology with I_{CAS} reduced by 4 plus the auxiliary OTA, Figure 5-3 (3rd case).

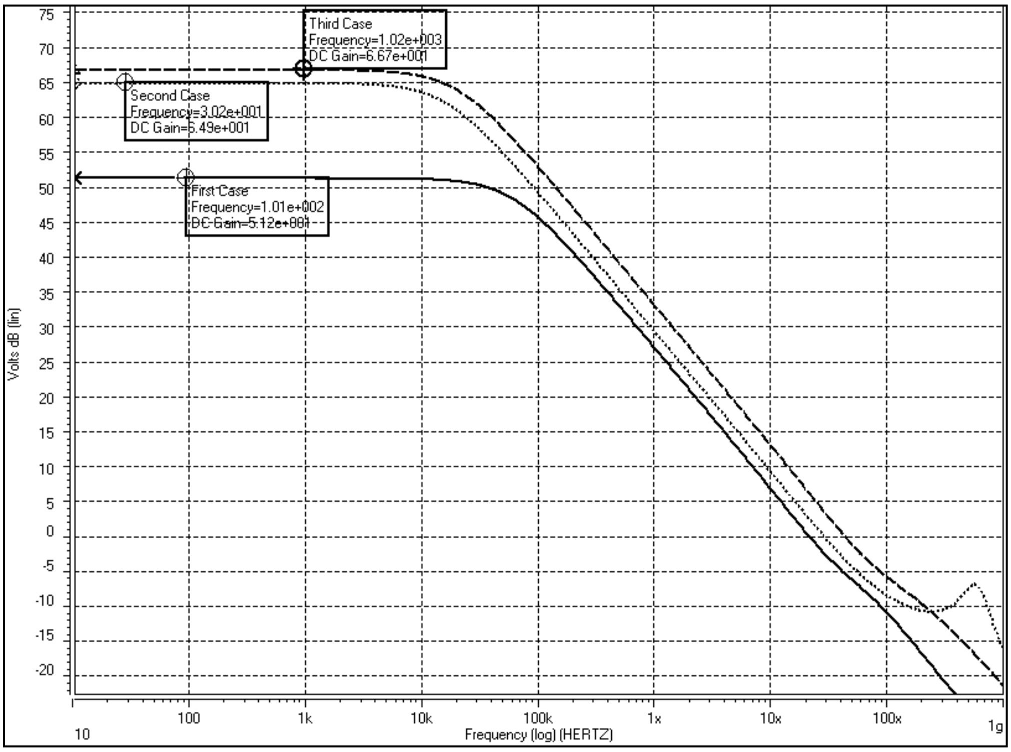


Figure 5-8 AC simulations (amplitude Bode diagrams) of the conventional topology (1st case), of the conventional topology with I_{CAS} reduced by a factor of 4 (2nd case) and of the proposed topology with I_{CAS} reduced by 4 plus auxiliary OTA (3rd case).

If the current I_{CAS} is reduced by a factor of 4 (and the aspect-ratio of devices M_3, M_{CAS} and M_{ICAS} are proportionally re-sized), a gain enhancement of 12 dB can be achieved, improving the accuracy to about 1.2 mV but the settling-time is degraded to about 29 ns (2nd case). This effect of the degradation of the speed is very well illustrated in the Bode diagram of Figure 5-8 (2nd case) for frequencies higher than 100 MHz. Finally, if the auxiliary OTA is used, according to the proposed topology (Figure 5-3) both, specifications for settling-time and accuracy are met, as illustrated in Figure 5-7 and Figure 5-8 (3rd case).

5.2 Optimum Compensation and Sizing

This example explores the optimum sizing and compensation of two-stage amplifiers based on a time-domain approach described in chapter 3. The main idea is to find the best sizing and compensation schema for the optimum efficiency. The selected CMOS amplifier topology is illustrated in Figure 5-9; it uses three compensation capacitors, C_A, C_B and C_M, which are connected between the two stages. The common-source second-stage is needed due to output range requirements. To achieve high DC gain, a differential folded-cascode structure is normally used for the input-stage.

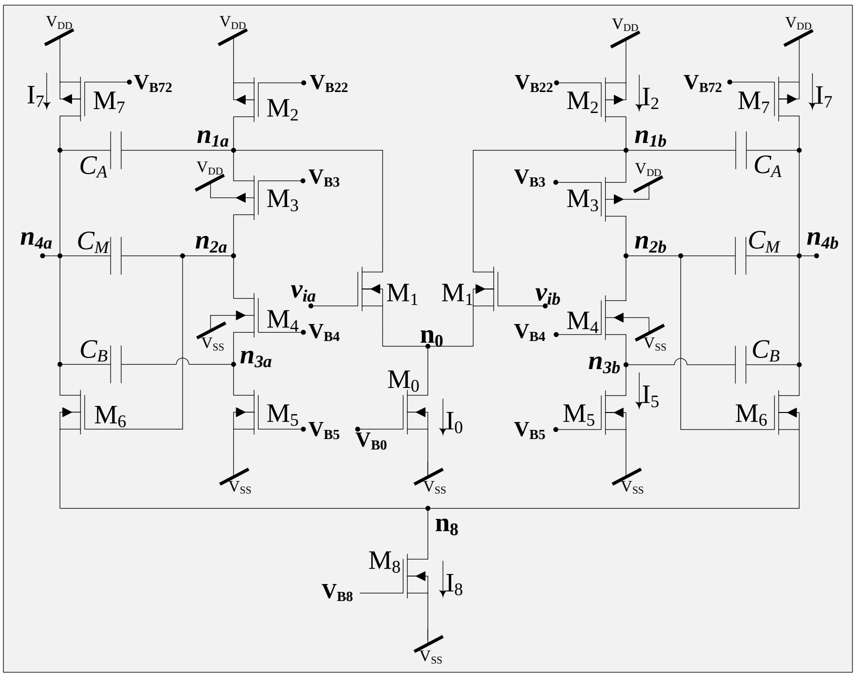


Figure 5-9 Schematic of a low-voltage two-stage cascode-compensated amplifier with a folded-cascode first-stage

5.2.1 Circuit insight

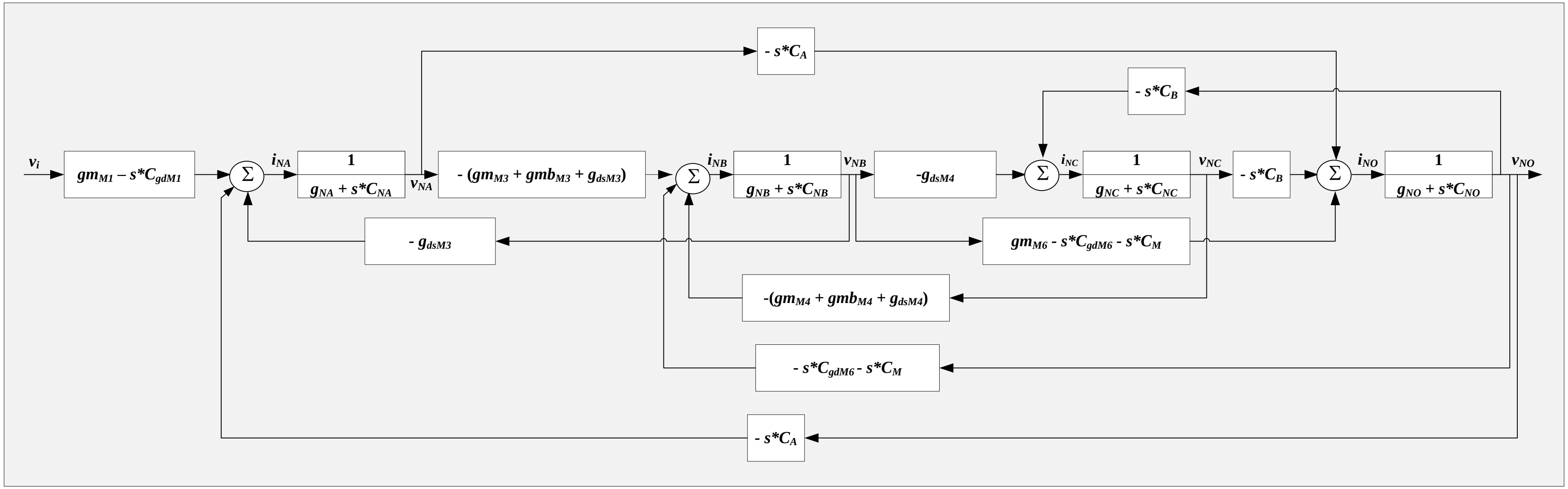
Usually only one of the three capacitors is used to compensate the amplifier. If only C_M is used, a pole splitting effect is achieved by a standard Miller compensation [94] which improves the stability of the amplifier but it puts a zero in right-half plane. To increase the PSRR, noise and bandwidth performance, an alternative proposed by Ahuja[65], consist of only using capacitor C_A, between a low impedance input-stage node and the amplifier output. In[66], an Improved-Ahuja style configuration is discussed which can be achieved by only using capacitor C_B. This technique reaches the same compensation effects but with lower power dissipation due to the fact that for a given transconductance a NMOS transistor needs less current than a PMOS transistor. For the circuit of Figure 5-9, the relevant transistors are the cascode transistor M_4. An hybrid combination of the Ahuja and Improved-Ahuja compensation techniques is proposed in[67]. This is obtained when C_A and C_B are used simultaneously and has the main advantage of increasing the amplifier unity-gain bandwidth when compared with other cascode-compensation schema. However, in[67], the system had to be reduced to 3rd order, by considering $C_A = C_B$. This is not the case in the example presented here. The three compensating capacitances have independent variables to be set by the optimization platform (chapter 4).

The optimization setup begins with the extraction of the transfer function of the topology, assuming the three independent capacitances. Analogous to the previous example, the methodology described in chapter 3 is used. In this case, the BSP model of this circuit shown in Figure 5-10. From the BSP model results a 4th order transfer function of the amplifier, similar to:

$$H(s) = \frac{N_4 \cdot s^4 + N_3 \cdot s^3 + N_2 \cdot s^2 + N_1 \cdot s^1 + N_0}{D_4 \cdot s^4 + D_3 \cdot s^3 + D_2 \cdot s^2 + D_1 \cdot s^1 + D_0} \tag{5.24}$$

Once more, the complete transfer function of this amplifier is several pages long and hence it will not be presented here. Bear in mind that the expression code, is, directly, copied into the source code of the circuit library (section 4.3), and integrated in the optimization platform.

From section 3.5, the performance parameters, used to compute the fitness of this circuit, were estimated.



v_i
$gm_{M1} - s*C_{gdM1}$
Σ
i_{NA}
$\frac{1}{g_{NA} + s*C_{NA}}$
v_{NA}
$- (gm_{M3} + gmb_{M3} + g_{dsM3})$
Σ
i_{NB}
$\frac{1}{g_{NB} + s*C_{NB}}$
v_{NB}
$-g_{dsM4}$
Σ
i_{NC}
$\frac{1}{g_{NC} + s*C_{NC}}$
v_{NC}
$- s*C_B$
Σ
i_{NO}
$\frac{1}{g_{NO} + s*C_{NO}}$
v_{NO}
$- s*C_A$
$- s*C_B$
$- g_{dsM3}$
$gm_{M6} - s*C_{gdM6} - s*C_M$
$-(gm_{M4} + gmb_{M4} + g_{dsM4})$
$- s*C_{gdM6} - s*C_M$
$- s*C_A$

Figure 5-10 Behavioral Signal Path of half of the amplifier with multiple compensation capacitors shown in Figure 5-9

$$g_{NA} = g_{dsM1} + g_{dsM2} + g_{dsM3} + gm_{M3} + gmb_{M3} \tag{5.25}$$

$$C_{NA} = C_{dbM1} + C_{dbM2} + C_{gdM2} + C_{gsM3} + C_{sbM3} + C_{gdM1} + C_{gsM1} + C_A \tag{5.26}$$

$$g_{NB} = g_{dsM3} + g_{dsM4} \tag{5.27}$$

$$C_{NB} = C_{dbM3} + C_{gdM3} + C_{dbM4} + C_{gdM4} + C_{gsM6} + C_{gdM6} + C_M \tag{5.28}$$

$$g_{NC} = g_{dsM5} + gm_{M4} + gmb_{M4} \tag{5.29}$$

$$C_{NC} = C_{gsM4} + C_{sbM4} + C_{gdM5} + C_{dbM5} + C_B \tag{5.30}$$

$$g_{NO} = g_{dsM6} + g_{dsM7} \tag{5.31}$$

$$C_{NO} = C_{dbM6} + C_{gdM6} + C_{dbM7} + C_{gdM7} + C_B + C_A + C_M + C_L \tag{5.32}$$

The OS of this topology is limited only by the drain-source saturation voltages of the output stage, as shown below:

$$\mathrm{OS} \approx V_{DD} - V_{dsatM7} - V_{dsatM6} - V_{dsatM8} - V_{margin} \tag{5.33}$$

where V_{DD} and V_{margin} are, respectively, the positive supply-voltage and the safety margin to guarantee proper saturation of the output devices (~100 mV). The total current consumption of this amplifier, I_{total}, is expressed by (excluding the common-mode feedback (CMFB) and the biasing circuitry):

$$I_{total} = 2 \cdot \left(I_{M2} + I_{M7}\right) \tag{5.34}$$

and consequently, the total power dissipation, P_{total}, is given by:

$$P_{total} = V_{DD} \cdot 2 \cdot \left(I_{M2} + I_{M7}\right) \tag{5.35}$$

5.2.2 Design procedure and circuit optimization

The optimization procedure assumes that all the compensation capacitors are included and, no fixed relations between the capacitance values are initially set. The addition of the Miller capacitor, C_M, in the hybrid configuration, enforces the effect associated with the parasitic capacitance of M_6 which cannot be ignored in the open-loop transfer function. Not only its impact on the circuit's step response must be analyzed but also a correct capacitor and transistor sizing optimization process has to be performed. In fact, this optimization process will determine how many compensation capacitors will be needed for the amplifier, as well as their proper sizing in order to reach a given settling-time.

The genetic algorithm optimization is configured with a set of genes for each chromosome, as shown in Figure 5-11. These include the biasing current, I_D, and the channel dimensions, L and W, for each transistor. The chromosome also includes the three compensation capacitors, C_A, C_B, C_M.

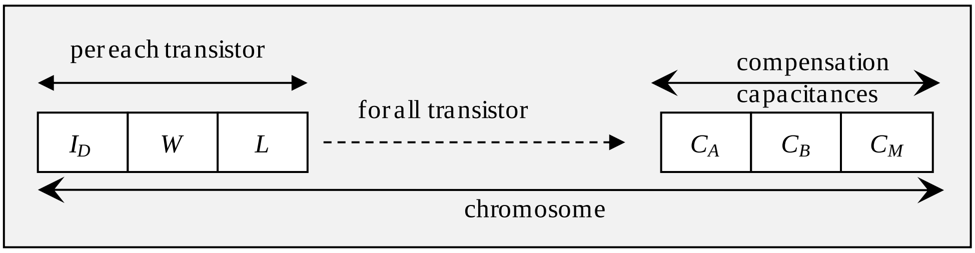


Figure 5-11 Format of the chromosome of the amplifier with multiple compensation capacitors

In this example, the channel width, *W*, and length, *L*, of each transistor, of the amplifier shown in Figure 5-9, are a direct mapping from the chromosome values. The sizes of biasing transistors are defined, analogously as the main amplifier transistors, considering a down-sizing factor of 1/10. This current consumption reduction still provides the correct biasing for the circuit.

During the algorithm evolution, (5.36) determines the fitness of each chromosome, which assess the circuit performance when compared to the desired specifications.

$$f(\mathrm{ST}, I_{total}, Cc_{total}) = \left(1 - e^{-\frac{\mathrm{ST\_desired}}{\mathrm{ST\_achieved}}}\right) \cdot \left(1 - e^{-\frac{I_{total}\_desired}{I_{total}\_achieved}}\right) \cdot \left(1 - e^{-\frac{Cc_{total}\_desired}{Cc_{total}\_achieved}}\right) \quad (5.36)$$

This fitness function evaluates the settling-time, ST, the total current consumption, I_{total}, and total compensation capacitance, Cc_{total}. All these three performance parameters have to be minimized in order to maximize the fitness. Cc_{total} is the sum of the three capacitances suggested by the optimization process: C_A; C_B and C_M. I_{total} is computed with (5.34). By means of the time-domain methodology, described in chapter 3, the ST is computed based on the step response derived from the transfer function of the amplifier in (5.24). Previously, the DC bias operating point is estimated by a single (.op) simulation analysis. Based on the DC bias operating point values, all the transistors in the main amplifier are checked to unsure that are operating in the saturation region. If one or more transistors fail the condition $V_{DS} < V_{dsat} + 50$ mV, a very low fitness (i.e. 1.0-e12) is given to the circuit/individual. This procedure prevents the lost of genetic material, which in the actual configuration (chromosome values) is not usable, but it is kept in the process of the generation of the new population. Furthermore, this poorly classified individual, with crossover and mutation operators, may lead to a better (or optimum) new individual.

To validate the proposed methodology, an amplifier was optimized for a 130 nm HS (high-speed) CMOS technology (L_{min} = 120 nm) with $V_{TP} \approx$ -0.33 V and $V_{TN} \approx$ 0.38 V. The circuit was designed to operate at a supply voltage of 1.2 V and to be used in a front-end Sample-and-Hold (S&H) of a 10-bit 240 MS/s Pipeline ADC (with unity feedback factor, β, and normalized sampling loading capacitances of about 1 pF). The settling-time specification is less than 2 ns for an accuracy better than 0.1% (corresponding to an error smaller than 0.5 mV assuming a differential reference voltage of 500 mV). For higher or lower load capacitances, the optimized amplifier scales linearly (*W's* and I_D*'s*), up and down, respectively.

5.2.3 Post-optimization and simulation results

Figure 5-12 shows the evolution of the amplifier performance parameters during the optimization process. At the beginning of optimization, we are able to see significant variations on the amplifier's parameters. That is due to the large design-space that is available for the chromosome variables. As the process runs, the algorithm is trying to search the right path to the best result, heading to a stable set of parameters. Little or no variations at all are present at the end of optimization process, meaning that the result is the optimal set of variables for our objective.

Figure 5-13 depicts the optimal set of compensating capacitances and its values during the overall optimization process. The obtained results demonstrate that the Miller capacitance, C_M, has a small or negligible contribution to the optimum time-domain step response. In fact, the process converged to a hybrid compensation type, i.e., a mixture of the Ahuja, C_A, and Improved-Ahuja, C_B, as suggested in[67].

However, an important observation is that optimum C_A and C_B values are not equal, as assumed in[67], but the total compensation capacitance is asymmetrically distributed. In this case, the compensation capacitances were 40% (C_A) and 60% (C_B), approximately, of the total compensation capacitance.

Table 5-3 shows the desired specifications used in the fitness function, the achieved values provided by the optimizer and the simulated results of the final netlist with the optimum values of the transistors sizes (*W*, *L*) and compensation capacitor values.

Figure 5-14 plots the output differential response of the amplifier for a differential input step with 500 mV, centered at 800 mV (input common-mode voltage). As shown in Figure 5-14 the differential output reaches the amplitude of, approximately, 500 mV. The zoomed area depicts the point where the signal enters the error margin range (higher than 499.5 mV) in approximately 1.63 ns after the step rising.

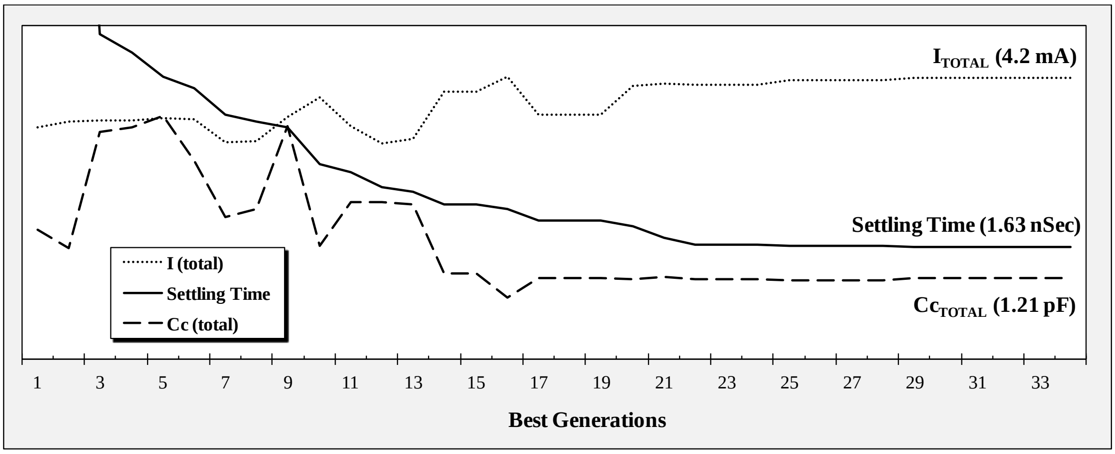


Figure 5-12 Variation of the performance parameters of the amplifier

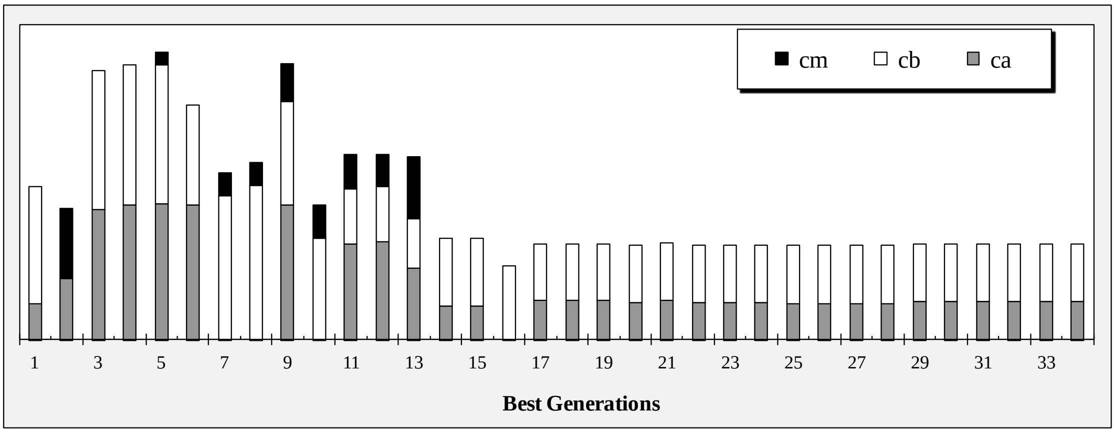


Figure 5-13 Evolution of the values of the compensation capacitances

Table 5-3 Optimized and post-simulated results for the proposed topology shown in Figure 5-9

	Desired specifications	Optimized Results	Simulated Results
A_{OL}	-	-	73.5 dB
OS	600 mV	706 mV	756 mV
C_{Ctotal}	2 pF	1.21 pF	1.21 pF
I_{total}	4 mA	4.2 mA	4.2 mA
ST	5 ns@0.1%	1.68 ns@0.1%	1.63 ns@0.1%

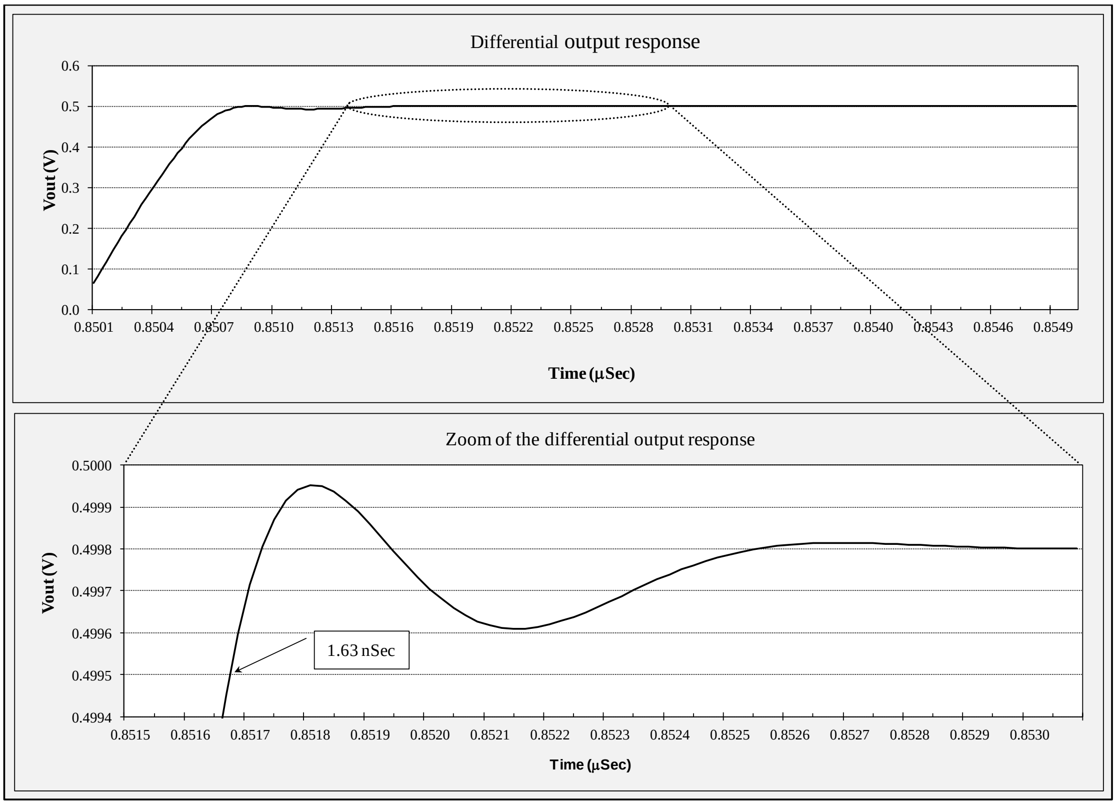


Figure 5-14 Simulated settling-response of the topology with the selected compensation schema.

5.3 Two-Stage Amplifier Employing Gain-Boosting Techniques

In this example, auxiliary amplifiers are used to boost the DC gain of the main amplifier, which also increased the analysis complexity with a transfer function of 8th order. It uses the hybrid compensation schema presented in the previous example. The amplifier topology is illustrated in Figure 5-15.

5.3.1 Circuit insight

A differential folded-cascode input stage followed by a differential common-source output stage is presented. The common-source second-stage is needed in order to obtain a larger output voltage swing. To achieve high gain, the input differential folded-cascode structure is normally used and the two auxiliary fully-differential single-stage folded-cascoded amplifiers, *SatN* and *SatP* are used to boost the output impedance of the first-stage in order to increase its finite DC gain[49]. As described in the previous example, of the previous section where the same topology without gain-boosting was used, it was demonstrated that, in fact, the best results are obtained using a hybrid compensation type, i.e., a mixture of the Ahuja, C_A, and Improved-Ahuja, C_B. It was also concluded that C_A and C_B should not be equally sized but, rather, the total compensation capacitance should be asymmetrically distributed (C_B larger than C_A), as shown the previous example, in section 5.2. Moreover, in the current example, two more capacitors are added, respectively, C_{sN} and C_{sP}. These two capacitors load the auxiliary amplifiers adding an extra degree of freedom to control the frequency of the doublets (pole-zero pair) added by the gain-boosting loops. All these capacitance effects are included in the open-loop transfer function. Two independent passive switched capacitor CMFB circuits are used to adjust the common-mode voltages of the two stages.

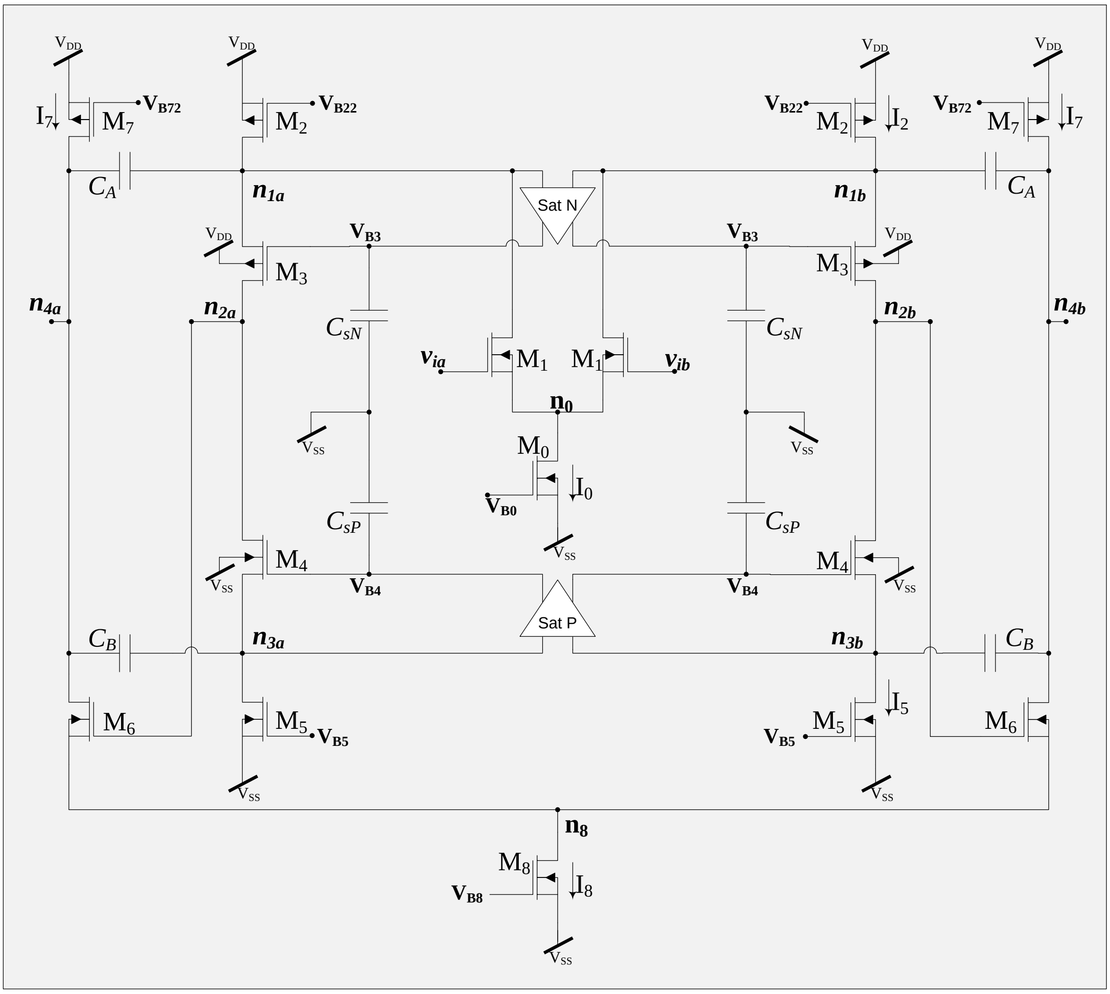


Figure 5-15 Schematic of the two-stage fully-differential gain-boosted OTA (biasing and CMFB circuitry not shown).

For the optimization, the first task is to obtain the transfer function, using the methodology derived in chapter 3. From the small signal equivalent, results the BSP model illustrated in Figure 5-16.

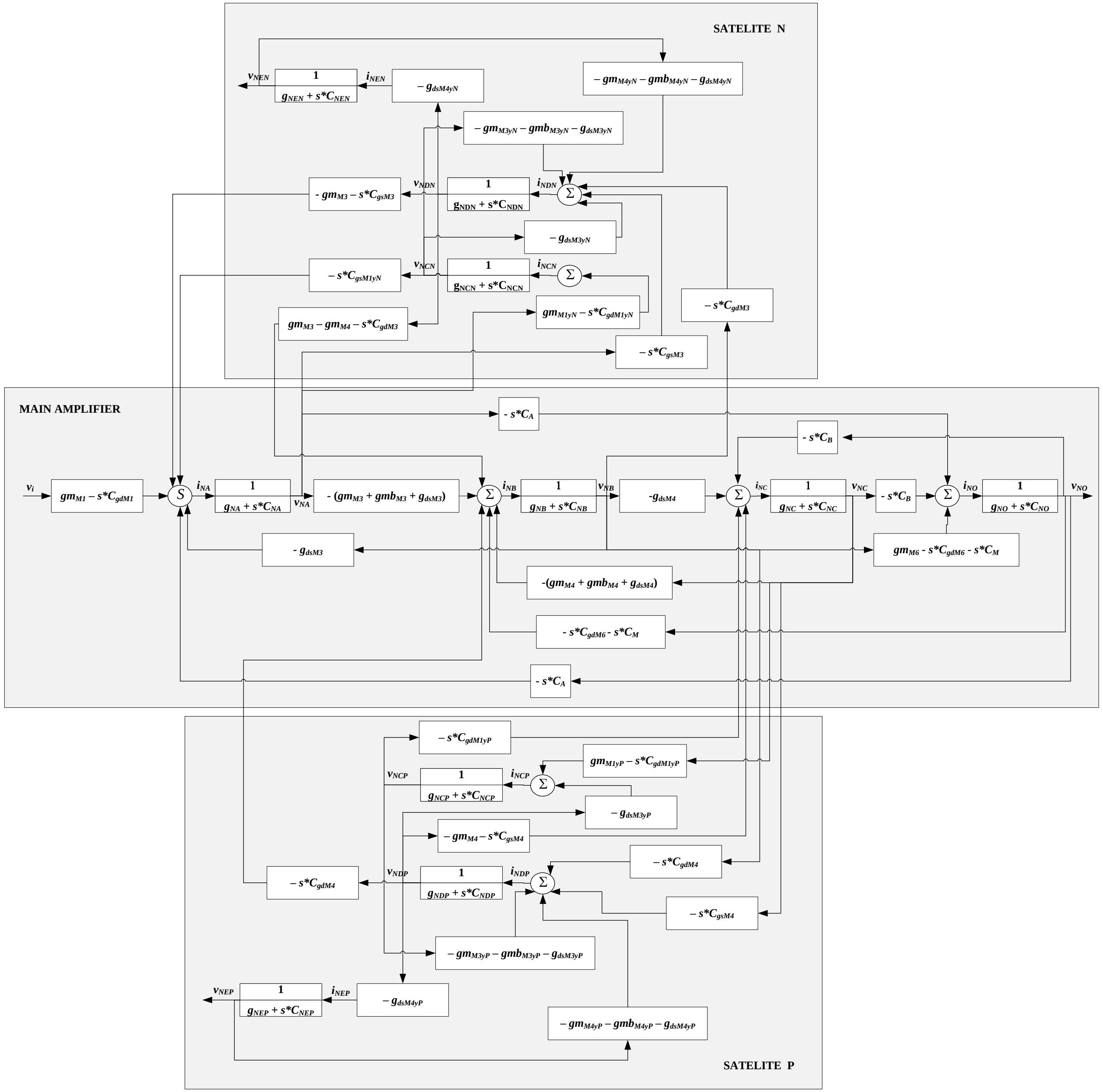

SATELITE N
v_{NEN}
$\frac{1}{g_{NEN} + s*C_{NEN}}$
i_{NEN}
$-g_{dsM4yN}$
$-gm_{M4yN} - gmb_{M4yN} - g_{dsM4yN}$
$-gm_{M3yN} - gmb_{M3yN} - g_{dsM3yN}$
$-gm_{M3} - s*C_{gsM3}$
v_{NDN}
$\frac{1}{g_{NDN} + s*C_{NDN}}$
i_{NDN}
Σ
$-g_{dsM3yN}$
$-s*C_{gsM1yN}$
v_{NCN}
$\frac{1}{g_{NCN} + s*C_{NCN}}$
i_{NCN}
Σ
$gm_{M1yN} - s*C_{gdM1yN}$
$-s*C_{gdM3}$
$gm_{M3} - gm_{M4} - s*C_{gdM3}$
$-s*C_{gsM3}$
MAIN AMPLIFIER
$-s*C_A$
$-s*C_B$
v_i
$gm_{M1} - s*C_{gdM1}$
S
i_{NA}
$\frac{1}{g_{NA} + s*C_{NA}}$
v_{NA}
$-(gm_{M3} + gmb_{M3} + g_{dsM3})$
Σ
i_{NB}
$\frac{1}{g_{NB} + s*C_{NB}}$
v_{NB}
$-g_{dsM4}$
Σ
i_{NC}
$\frac{1}{g_{NC} + s*C_{NC}}$
v_{NC}
$-s*C_B$
Σ
i_{NO}
$\frac{1}{g_{NO} + s*C_{NO}}$
v_{NO}
$-g_{dsM3}$
$gm_{M6} - s*C_{gdM6} - s*C_M$
$-(gm_{M4} + gmb_{M4} + g_{dsM4})$
$-s*C_{gdM6} - s*C_M$
$-s*C_A$
$-s*C_{gdM1yP}$
$gm_{M1yP} - s*C_{gdM1yP}$
v_{NCP}
$\frac{1}{g_{NCP} + s*C_{NCP}}$
i_{NCP}
Σ
$-g_{dsM3yP}$
$-gm_{M4} - s*C_{gsM4}$
$-s*C_{gdM4}$
$-s*C_{gdM4}$
v_{NDP}
$\frac{1}{g_{NDP} + s*C_{NDP}}$
i_{NDP}
Σ
$-s*C_{gsM4}$
$-gm_{M3yP} - gmb_{M3yP} - g_{dsM3yP}$
v_{NEP}
$\frac{1}{g_{NEP} + s*C_{NEP}}$
i_{NEP}
$-g_{dsM4yP}$
$-gm_{M4yP} - gmb_{M4yP} - g_{dsM4yP}$
SATELITE P

Figure 5-16 Behavioral Signal Path of half of the gain-boosted amplifier circuit shown in Figure 5-15

$$g_{NA} = g_{dsM1} + g_{dsM2} + g_{dsM3} + gm_{M3} + gmb_{M3} \tag{5.37}$$

$$C_{NA} = C_{dbM1} + C_{dbM2} + C_{gdM2} + C_{gsM3} + C_{sbM3} + C_{gdM1} + C_{gsM1} + C_{A} + C_{gsM1yN} + C_{gdM1yN} \tag{5.38}$$

$$g_{NB} = g_{dsM3} + g_{dsM4} \tag{5.39}$$

$$C_{NB} = C_{dbM3} + C_{gdM3} + C_{dbM4} + C_{gdM4} + C_{gsM6} + C_{gdM6} \tag{5.40}$$

$$g_{NC} = g_{dsM5} + gm_{M4} + gmb_{M4} \tag{5.41}$$

$$C_{NC} = C_{gsM4} + C_{sbM4} + C_{gdM5} + C_{dbM5} + C_{B} + C_{gdM4} + C_{gdM1yP} + C_{gsM1yP} \tag{5.42}$$

$$g_{NO} = g_{dsM6} + g_{dsM7} \tag{5.43}$$

$$C_{NO} = C_{dbM6} + C_{gdM6} + C_{dbM7} + C_{gdM7} + C_{B} + C_{A} + C_{L} \tag{5.44}$$

$$g_{NCN} = g_{dsM1yN} + g_{dsM2yN} + g_{dsM3yN} + gm_{M3yN} + gmb_{M3yN} \tag{5.45}$$

$$C_{NCN} = C_{dbM1yN} + C_{gdM1yN} + C_{dbM2yN} + C_{gdM2yN} + C_{gsM3yN} + C_{sbM3yN} \tag{5.46}$$

$$g_{NDN} = g_{dsM3yN} + g_{dsM4yN} \tag{5.47}$$

$$C_{NDN} = C_{dbM3yN} + C_{gdM3yN} + C_{dbM4yN} + C_{gdM4yN} + C_{gdM3} + C_{gsM3} + C_{sN} \tag{5.48}$$

$$g_{NEN} = gm_{M4yN} + gmb_{M4yN} + g_{dsM4yN} + g_{dsM5yN} \tag{5.49}$$

$$C_{NEN} = C_{gsM4yN} + C_{sbM4yN} + C_{gdM5yN} + C_{dbM5yN} \tag{5.50}$$

$$g_{NCP} = g_{dsM1yP} + g_{dsM2yP} + g_{dsM3yP} + gm_{M3yP} + gmb_{M3yP} \tag{5.51}$$

$$C_{NCP} = C_{dbM1yP} + C_{gdM1yP} + C_{dbM2yP} + C_{gdM2yP} + C_{gsM3yP} + C_{sbM3yP} \tag{5.52}$$

$$g_{NDP} = g_{dsM3yP} + g_{dsM4yP} \tag{5.53}$$

$$C_{NDP} = C_{dbM3yP} + C_{gdM3yP} + C_{dbM4yP} + C_{gdM4yP} + C_{gdM4} + C_{gsM4} + C_{sP} \tag{5.54}$$

$$g_{NEP} = gm_{M4yP} + gmb_{M4yP} + g_{dsM4yP} + g_{dsM5yP} \tag{5.55}$$

$$C_{NEP} = C_{gsM4yP} + C_{sbM4yP} + C_{gdM5yP} + C_{dbM5yP} \tag{5.56}$$

The complexity of this BSP model originates a 10th order open-loop transfer function of the amplifier, similar to:

$$H(s)=\frac{N_{10}\cdot s^{10}+N_9\cdot s^9+...+N_1\cdot s^1+N_0}{D_{10}\cdot s^{10}+D_9\cdot s^9+...+D_1\cdot s^1+D_0} \tag{5.57}$$

For the sake of simplicity, as in the previous examples, the complete transfer function is not presented here. Anyhow, the expression code is, directly integrated in the optimization platform.

The performance parameter needed to compute the fitness of the circuit, during optimization, are obtained using the definitions in section 3.5 of chapter 3. The OS of this topology is limited only by the drain-source saturation voltages of the output transistors and it is defined by:

$$\text{OS}\approx V_{DD}-V_{dsatM7}-V_{dsatM6}-V_{dsatM8}-V_{m\arg in} \tag{5.58}$$

where V_{DD} and V_{margin} are, respectively, the positive supply-voltage and some safety margin to guarantee proper saturation of the output devices (~100 mV). The total current used by this topology, I_{total}, is given by (excluding the CMFB and the biasing circuitry):

$$I_{total}=2\cdot\left(I_{M2}+I_{M7}\right) \tag{5.59}$$

5.3.2 Design procedure and circuit optimization

The presented amplifier topology, with the proposed compensation capacitances (C_A and C_B) plus C_{sN} and C_{sP}, is an eight-order system. As claimed in chapter 4, the time-domain optimization methodology can significantly simplify the calculus needed for circuit optimization of superior order topologies and still provide accurate results verified by electrical simulation. This means that considering the settling-time for a given accuracy, it is guaranteed that the amplifier has enough open-loop gain, A_{OL}, output-swing, OS, closed loop bandwidth and closed loop stability.

The genetic algorithm optimization is configured with a set of genes for each chromosome, as shown in Figure 5-17. These include the biasing current, I_D, and the channel dimensions, L and W, for each transistor in the main circuit of the amplifier. The chromosome also includes the compensation capacitors, C_A, C_B, and the two extra capacitances C_{sP} and C_{sN} described earlier. The sizes of the transistors of the biasing circuitry are computed to ensure the correct DC polarization of the amplifier. In order to reduce the current dissipation, again, the biasing transistors sizes are made 1/10 of the main circuit amplifier.

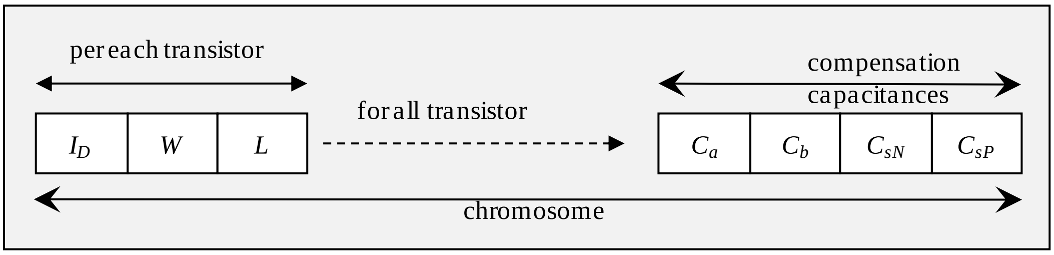


Figure 5-17 Format of the two-stage gain-boosted amplifier chromossome

During the algorithm evolution, expression (5.60) evaluates the fitness of each chromosome, quantifying the circuit performance when compared to the desired specifications.

$$f(\mathrm{ST}, I_{total}, Cc_{total}) = \left(1 - e^{-\frac{\mathrm{ST\_desired}}{\mathrm{ST\_achieved}}}\right) \cdot \left(1 - e^{-\frac{I_{total}\_desired}{I_{total}\_achieved}}\right) \cdot \left(1 - e^{-\frac{Cc_{total}\_desired}{Cc_{total}\_achieved}}\right) \qquad (5.60)$$

The adopted fitness function applied to circuit evaluates the settling-time, ST, the total current dissipation, I_{total}, and total compensation capacitance, Cc_{total}. All these three performance parameters have to be minimized. Cc_{total} is the sum of the three compensation capacitances: C_A and C_B. I_{total} is computed with (5.59). The ST is computed based on the step response derived according to the method described on chapter 4, using the transfer function of the amplifier in (5.57). Once more, the fitness value is also dependent of the DC bias operating point values of the transistors of the main circuitry. Again, the circuit instances that have transistors (main circuit and auxiliary amplifiers) with $V_{DS} < V_{dsat}$ + 50 mV, are classified with a very low fitness (i.e. 1.0-e12). As already mentioned, the genetic material of these individuals are not (completely) discarded and it may be used in the generation of the mew population process, perhaps, providing a new individual with an optimum result.

The target fabrication technology is a 130 nm HS (high-speed) 1.2 V CMOS technology (L_{min} = 120 nm). The mobility and threshold parameters (level 2), K_N, K_P, V_{TN} and V_{TP} parameters of the devices are, respectively, 525 μAV^{-2}, 145 μAV^{-2}, 0.38 V and -0.33 V.

5.3.3 Post-optimization and simulation results

Figure 5-18 shows the simulated step response in a closed-loop gain of 2 configuration of the amplifier.

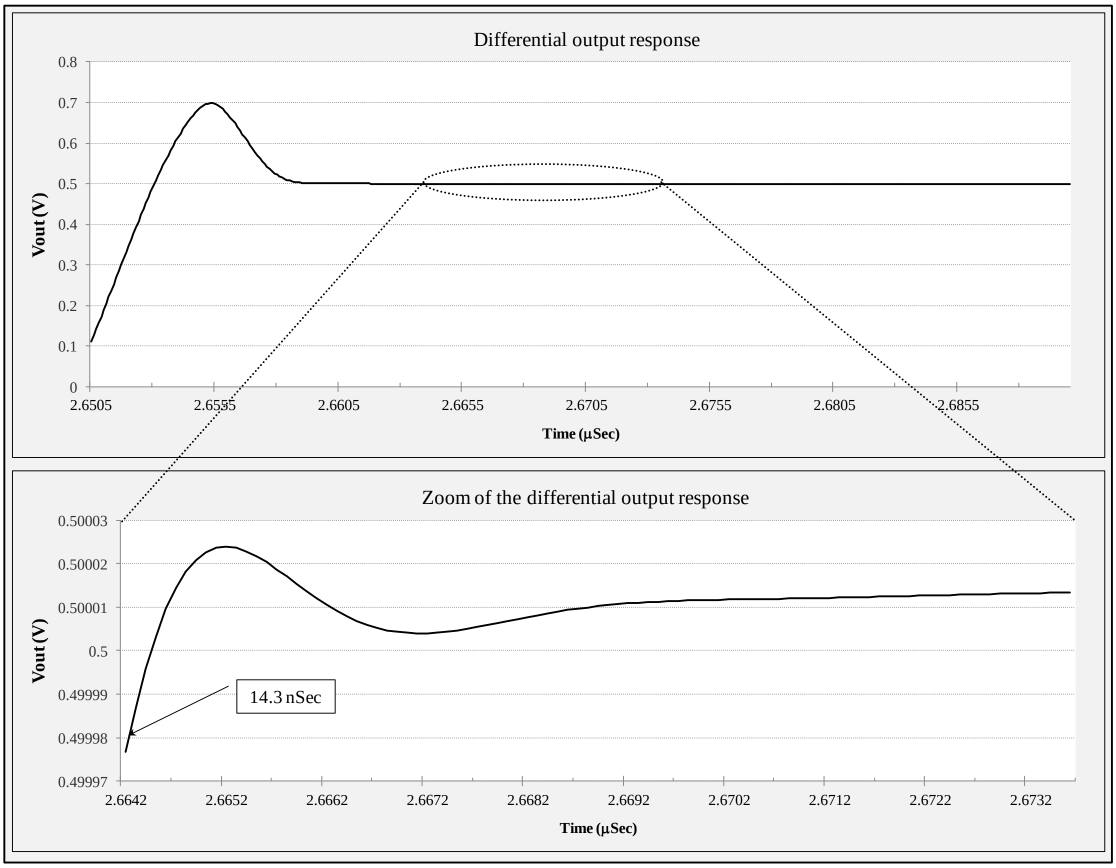


Figure 5-18 Simulated differential output response of the OTA, employing gain-boosting techniques.

Table 5-4 Optimized and post-simulated results for the amplifier topology shown in Figure 5-15

	Desired specifications	Optimized Results	Simulated Results
A_{OL}	-	-	101.8 dB
OS	600 mV	823 mV	810 mV
C_{Ctotal}	10 pF	1.88 pF	1.88 pF
I_{total}	10 mA	8 mA	9 mA
ST	25 ns@0.004%	15.8 ns@0.004%	14.3 ns@0.004%

The differential input step voltage is 250 mV, centered at 800 mV (input common-mode voltage). The differential output voltage reaches the amplitude of, approximately, 500 mV. The zoomed area depicts the point where the signal enters the error margin range (higher than 499.98 mV). This occurs, approximately 15 ns after the step rising within the 20 μV settling accuracy.

Table 5-4 resumes the results of the HSPICE electrical simulation of the topology for the resulting optimum transistors sizes (*W*, *L*) and compensation capacitor values.

5.4 A Novel Two-Stage Self-Biased Inverter-Based Amplifier

Finally, a novel two-stage fully-differential CMOS amplifier completely self-biased is optimized by the proposed methodology and developed software platform.

The novel topology consists of two cascaded inverter stages with the topology depicted in Figure 5-19. The input stage consists of an inverter differential pair (M_{12} and M_{13}) connected to a current source, M_{11}, and to a voltage controlled resistor M_{14}. The output stage, has an identical topology, except for nodes n_{BA} and n_{BB}, which are connected together into node n_{B2}. The differential-pair is formed by transistors M_{22} and M_{23}, the current source is transistor M_{21} and the voltage controlled resistor is transistor M_{24}. At the input stage, node n_{BA} and n_{BB} have been separated to connect the compensation capacitors, C_C, thus making it unnecessary to use the inefficient Miller compensation (nodes v_{on1} e v_{op1}). In general, and as stated before, the main drawbacks of Miller compensation are poor power efficiency [95], low PSRR and require a large value of compensation capacitor [96]. Also, the feed-forward current that flows through the C_C to the output is also another issue of Miller-compensated amplifiers [97]. The current introduces a right-half-plane (RHP) zero which significantly reduces the closed-loop stability. The deteriorating effect of this current is originated from the fact that it tries to pass the signal to the output by directly bypassing the second stage. Hence the 180° phase shift introduced by this stage is nullified and the output polarity is reversed at lower frequencies. A nullifying resistor is applied in series with the compensation capacitor to avoid this [79]. The resistor increases the impedance of the path which equivalently moves the RHP zero to higher frequencies. In practice however, the resistor is affected by temperature and device fabrication which results in variation of stability from die to die [96].

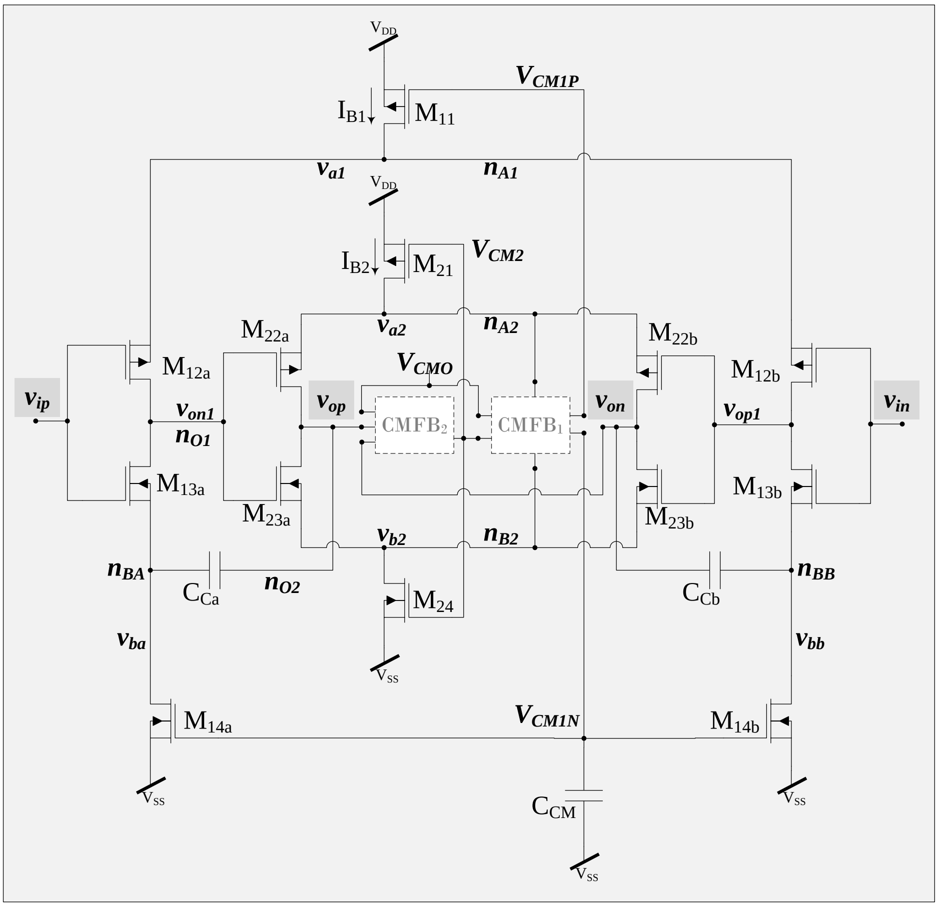


Figure 5-19 Schematic of the new two-stage self-biased inverter-based amplifier

The CMFB circuits presented in Figure 5-20 source the control voltages of both stages and, simultaneously, bias of the two stages of the amplifier. The output common-mode (CM) level is adjusted through a dedicated circuit, CMFB$_2$, as illustrated by Figure 5-20-a), $V_{CM2} = (v_{op} + v_{on})/2$[98]. This control voltage also biases transistor M_{21}, controls the resistance value of M_{24}, and is used to generate the biasing control voltage of stage one. The CMFB$_1$ circuit, as illustrated by Figure 5-20-b), is an inverter-based differential pair which compares voltage V_{CM2} with a reference voltage, V_{CMO}, and generates control voltages, V_{CM1P} and V_{CM1N}, to bias the first stage of the amplifier. It should be noticed that the CMFB$_1$ circuit is connected to nodes n_{A2} and n_{B2}, thus avoiding the use of extra biasing transistors. Transistors M_{32} and M_{33} are down-scaled versions of M_{22} and M_{23}, respectively.

The PVT effects are reduced by using completely complementary (half PMOS and half NMOS) circuit implementation, having a negative-feedback loop, and also the fact that the self-biasing voltages V_{CM1P}, V_{CM1N} and V_{CM2} are connected to the main circuit of the amplifier through a negative feedback. To illustrate this last effect, consider that the voltages on nodes v_{op1} and v_{on1} have already been stabilized. If V_{DD} increases, the source-gate voltage, V_{SGP21}, in device M_{21} (PMOS) also increases, raising the bias current I_{B2}. This effect will change, proportionally, the current in the two output inverters, augmenting the output common-mode voltage. As a consequence, CMFB$_2$ circuit will produce a higher V_{CM2} output control voltage forcing V_{SGP21} to remain constant, thus compensating the V_{DD} variation.

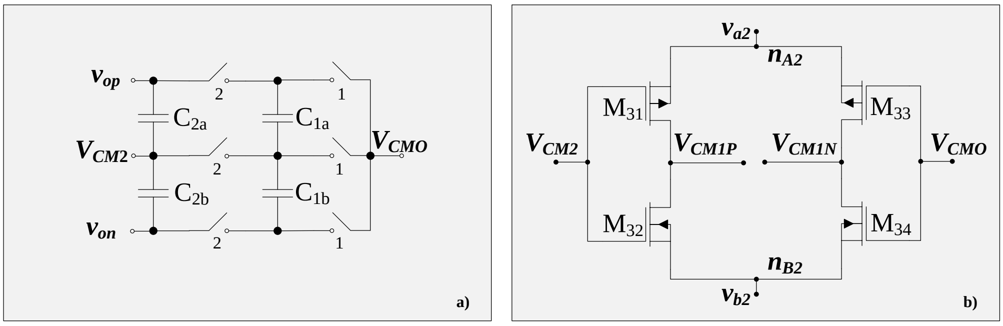


Figure 5-20 Schematics of the common-mode feedback circuits: a) SC network for 2nd stage (CMFB2); b) Continuous time CMFB circuit for input stage (CMFB1).

As will be shown in the next sub-section, the DC gain (A_{V0}) of the proposed amplifier is proportional to $(gm/g_{ds})^2$. Constant gain is achieved through the negative feedback self-biasing loop which adjusts gm and g_{ds} in the same way, through the biasing current, since both, gm and g_{ds} are, directly proportional to the biasing current.

Using the linearization techniques described in section 3.4.5 of chapter 3, the small signal equivalent (differential-mode, DM) of the amplifier is obtained. From small signal equivalent, the BSP model [70] is extracted and illustrated in Figure 5-21. As claimed in chapter 3, this model permits a better insight of the small-signal behavior of the amplifier, in particular:

- The feedback loop is created by the compensation capacitor, C_C, and by the finite output conductance of transistor M_{13};
- The Miller effect through parasitic capacitance C_{gd2};
- Feed-forward paths through C_{gs13} and Cc; and the poles and zeros, in other words, the order of the transfer function (in this case, 3rd order).

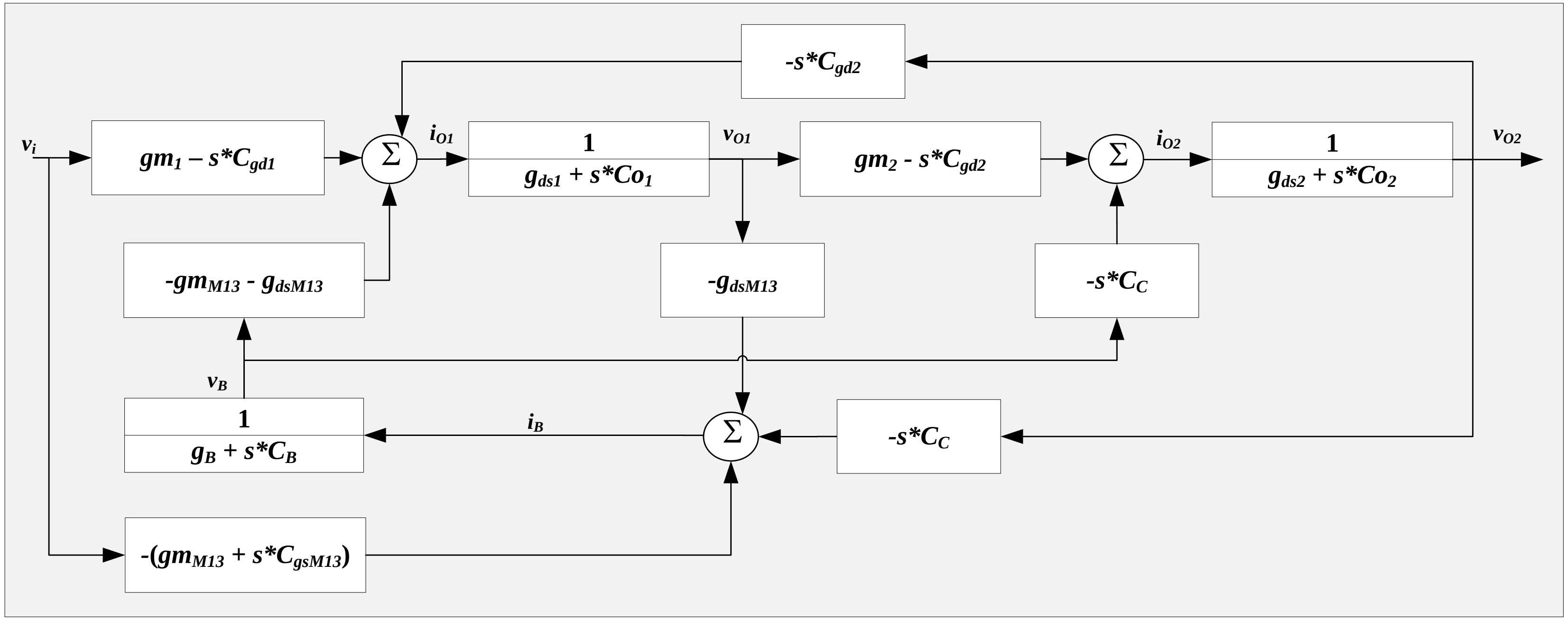


Figure 5-21 Behavioral signal path model of the two-stage self-biased inverter-based amplifier (for simplicity only half the circuit is shown)

$$gm_1 = gm_{M12} + gm_{M13} \tag{5.61}$$

$$g_{ds1} = g_{dsM12} + g_{dsM13} \tag{5.62}$$

$$C_{gd1} = C_{gd\,M12} + C_{gd\,M13} \tag{5.63}$$

$$gm_2 = gm_{M22} + gm_{M23} \tag{5.64}$$

$$g_{ds2} = g_{dsM22} + g_{dsM23} \tag{5.65}$$

$$C_{gd2} = C_{gd\,M22} + C_{gd\,M23} \tag{5.66}$$

$$g_B = g_{dsM13} + g_{dsM14} + gm_{M13} \tag{5.67}$$

$$C_B = C_{gd\,M14} + C_{dbM14} + C_{gsM13} + C_{sbM13} + C_C \tag{5.68}$$

$$Co_1 = C_{gd\,M12} + C_{gd\,M13} + C_{dbM12} + C_{dbM13} + C_{gsM22} + C_{gsM23} + C_{gd\,M22} + C_{gd\,M23} \tag{5.69}$$

$$Co_2 = C_{gd\,M22} + C_{gd\,M23} + C_{dbM22} + C_{dbM23} + C_C + C_L \tag{5.70}$$

5.4.1 Circuit insight

Using the behavioral signal path model presented in Figure 5-21 and by writing down the equations for I_{O1}, V_{O1}, I_{O2}, V_{O2}, I_b and V_b, it becomes possible to extract the transfer function of the amplifier. For the sake of simplicity, minor simplifications were used in the derived equations and can be understood through the following example: $C_{gd\mathbf{x}} = C_{gd\mathbf{x}2} + C_{gd\mathbf{x}3}$; $\mathbf{x} \in \{1, 2\}$. This is valid for all parasitic capacitances: C_{gd}; C_{db}; C_{gs}; output conductance: g_{ds}; and transconductances: *gm*. Body effect: *gmb;* of transistors M_{12}, M_{13}, M_{22}, and M_{23} were neglected, but can be easily included into the equations. The following equations represent the capacitances on nodes n_{O1}, n_{O2}, and n_B, and the admittance g_B on node n_B. C_L represents the load capacitance at the output node, n_{O2}.

$$C_{O1} = C_{gd1} + C_{db1} + C_{gs2} + C_{gd2} \tag{5.71}$$

$$C_{O2} = C_{gd2} + C_{db2} + C_C + C_L \tag{5.72}$$

$$C_B = C_{gd14} + C_{db14} + C_{gs13} + C_C + C_{sb13} \tag{5.73}$$

$$g_B = g_{ds13} + g_{ds14} + gm_{13} \approx g_{ds14} + gm_{13} \tag{5.74}$$

From the transfer function it is possible to obtain the low-frequency open-loop gain (DC gain), A_{V0},

$$A_{V0} = \frac{gm_2 \cdot \left(gm_{13}^{\ 2} - gm_1 \cdot g_B\right)}{g_{ds2} \cdot \left(g_{ds13} \cdot gm_{13} - g_{ds1} \cdot g_B\right)} \tag{5.75}$$

It is curious that this gain is not the cascaded gain of each inverter stage, which is explained, because nodes n_{Ba} and n_{Bb} have been separated to create two independent low-impedance nodes for the compensation capacitors. To a good approximation, A_{V0} can be given by the cascaded gain of each inverter stage,

$$A_{V0} \approx \frac{gm_2 \cdot gm_1}{g_{ds2} \cdot g_{ds1}} \tag{5.76}$$

From the pole/zero analysis of the amplifier it is possible to verify that there are three poles and three zeros. There are two positive high frequency zeros which do not influence the stability of the amplifier (so they will not be considered), and one negative zero (as long as $(gm_{13}^2/gm_1) < g_B$), (5.77), which should be taken into account. As for the poles, there is a dominant one and a pair of complex conjugated poles. Equation (5.78) represents the dominant pole, while (5.79) and (5.80) respectively represent the natural frequency and quality factor of the complex conjugated poles.

$$w_Z = \frac{\frac{gm_{13}^{\;2}}{gm_1} - g_B}{C_B} \tag{5.77}$$

$$w_{P1} = \frac{g_{ds2} \cdot \left(g_{ds13} \cdot gm_{13} - g_{ds1} \cdot g_B\right)}{\left(C_{O2} \cdot g_{ds1} + C_{O1} \cdot g_{ds2} + C_{gd2} \cdot gm_2\right) \cdot g_B + C_C \cdot gm_{13} \cdot gm_2} \tag{5.78}$$

$$w_{n2,3} = \sqrt{\frac{\left(C_{O2} \cdot g_{ds1} + C_{O1} \cdot g_{ds2} + C_{gd2} \cdot gm_2\right) \cdot g_B + C_C \cdot gm_{13} \cdot gm_2}{C_{O1} \cdot C_B \cdot C_{O2}}} \tag{5.79}$$

$$Q_{P2,3} = \frac{\sqrt{C_{O1} \cdot C_B \cdot C_{O2} \cdot \left(\left(C_{O2} \cdot g_{ds1} + C_{gd2} \cdot gm_2\right) \cdot g_B + C_C \cdot gm_{13} \cdot gm_2\right)}}{\left(g_B \cdot C_{O1} + g_{ds1} \cdot C_B\right) \cdot C_{O2} + C_{gd2} \cdot C_B \cdot gm_2} \tag{5.80}$$

From (5.75) and (5.78) it is possible to arrive at the expression for the gain-bandwidth product, GBW,

$$\text{GBW} = \frac{gm_2 \cdot \left(gm_1 \cdot g_B - gm_{13}^{\;2}\right)}{\left(\left(C_{O2} \cdot g_{ds1} + C_{O1} \cdot g_{ds2} + C_{gd2} \cdot gm_2\right) \cdot g_B + C_C \cdot gm_{13} \cdot gm_2\right)} \tag{5.81}$$

The OS is given by the minimum of two values: OS$^+$ or OS$^-$, defined by

$$\text{OS}^+ \approx V_{DD} - V_{CMO} - V_{dsat21} - V_{dsat22} - V_{margin} \tag{5.82}$$

$$\text{OS}^- \approx V_{CMO} - V_{dsat23} - V_{dsat24} - V_{margin} \tag{5.83}$$

where V_{dsat} are the saturation voltage of the transistors of the output stage, V_{CMO} is the common-mode output voltage, V_{DD} and V_{margin} are, respectively, the positive supply-voltage and some additional safety margin to guarantee proper saturation of the output devices (~100 mV). The total current used by this topology, I_{total}, is given by (excluding the CMFB and the biasing circuitry):

$$I_{total} = \left(I_{D_{11}} + I_{D_{21}}\right) \tag{5.84}$$

5.4.2 Design procedure and circuit optimization

It is not an easy task, determining guidelines for a good and successful sizing of this amplifier, due to the complicated expressions that were obtained for the poles and zeros from the small signal analysis. The many feedback loops and feed-forward paths present in the behavioral signal path model, Figure 5-21, illustrate the degree of complexity in achieving an accurate qualitative analysis of the proposed amplifier. Although there is not a clear design procedure, some precautions and considerations may be mentioned for a good starting point. Another option, probably the best choice due to the complexity of the circuit, is to use the proposed optimization platform setup with the equations for sizing the circuit. The proposed design constraints for our amplifier are as follows:

- The minimum value of C_C is mainly imposed by the kT/C thermal noise constraints (which is set by the application where the amplifier is being used). The value of C_C is a compromise between the pole quality factor (large C_C) and the zero and dominant pole (small C_C).
- M_{12} is designed to have a large L for high DC gain and a low V_{dsat} for high gm to move the zero to higher frequencies. The width should not be too wide to decrease C_{O1}.
- Transistor M_{13} should have a large L for high DC gain. The transconductance (gm) value is a compromise between higher bandwidth (small gm) and lower $Q_{P2,3}$ (large gm). The V_{dsat} of this transistor should be chosen with care to keep C_{O1} and C_{gs13} small.
- M_{14} should be designed to have a very large output conductance. For this, the channel length and V_{dsat} of this transistor should be small. Care should be taken when choosing V_{dsat}, not to load node n_B, keeping C_B small.

- Transistors M_{22} and M_{23} should have low V_{dsat} to increase output swing. They should have large L for high DC gain but should not be large transistors to keep C_{O1} small. The *gm* of these transistors are a trade-off between bandwidth and $Q_{P2,3}$ (small *gm*) and phase margin (large *gm*).
- Transistor M_{11} should be biased in the triode/saturation boundary region with a V_{DS} that keeps M_{12} saturated. This transistor should be sized to guarantee the current necessary for the *gm* of transistors M_{12} and M_{13}.
- Transistors M_{21} and M_{24} should be biased in the triode/saturation boundary region, with low V_{dsat} to guarantee highest possible OS. These transistors should be sized to guarantee the current for the *gm* of M_{22} and M_{23}.
- Transistors M_{32} and M_{33} are simply down-scaled (D = 4) versions of M_{22} and M_{23}, respectively.
- Optimum channel length of 1.3-to-1.5×L_{min} should be used to maintain good insensitivity to PVT variations, avoiding short channel-length effects and, at the same time, maximizing speed.

The previous enumerated constrains were considered in order to limit the search design space available to the optimizer. The Figure 5-22 depicts the genetic individual configuration for the optimizer. It contains the range variation of design space for the widths, *W*, and lengths, *L*, of all the transistors and the compensation capacitance value of the circuit.

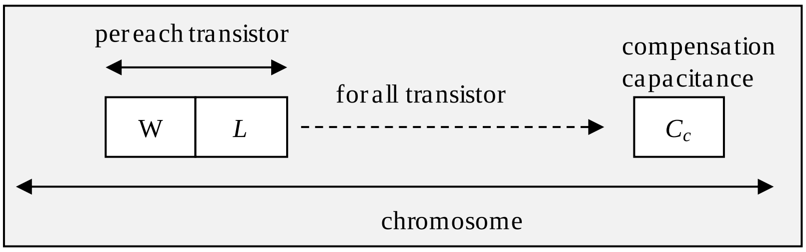


Figure 5-22 Format of the chromossome of the two-stage self-biased inverter-based amplifier.

The target fabrication technology is a 130 nm HS (high-speed) 1.2 V CMOS technology (L_{min} = 120 nm). The mobility and threshold parameters (level 2), K_N, K_P, V_{TN} and V_{TP} parameters of the devices are, respectively, 525 μAV^{-2}, 145 μAV^{-2}, 0.38 V and -0.33 V. The common-mode input voltage, V_{CMI}, was established at 550 mV.

5.4.2.1 Frequency-domain optimization

Using the frequency-domain performance parameters defined in section 5.4.1, the amplifier was optimized using the frequency-domain optimization and it was defined as a possible application for the amplifier, a 12-bit pipeline analog-to-digital converter (ADC). This requires the amplifier to have a DC gain larger than 80 dB and a GBW as large as possible for a 4 pF load, while minimizing power dissipation to be used in the front-end stage of the ADC. Due to thermal noise, the value for C_C is left with a minimal variable range from 0.5 pF to 0.6 pF.

During optimization, the fitness value of each individual is computed according to:

$$fitness = f_{Av0} \cdot f_{\mathrm{GBW}} \cdot f_{wz} \cdot f_{Qp2,3} \cdot f_{Wn2,3} \cdot f_{\mathrm{OS}} \cdot f_{\mathrm{PM}} \cdot f_{Itotal} \qquad (5.85)$$

where f_{x} represents the partial fitness of each circuit performance parameter to be considered in the final fitness of each individual, defined as:

- f_{Av0} the DC gain, to be maximized;
- f_{GBW} the GBW, to be maximized;
- f_{wz} the first pole frequency, to be maximized;
- $f_{\mathrm{Qp2,3}}$ the *Qp2,3*, to be maximized;
- $f_{\mathrm{Wn2,3}}$ the *Wn2,3*, to be maximized;
- f_{OS} the OS available for the output, to be maximized;
- f_{PM}, the PM, to be maximized;
- f_{Itotal} the total current drawn from V_{DD}, to be minimized.

In section 4.2.2 of chapter 4, the calculation of the partial fitness was presented, according the requirements: maximization, minimization or target value.

Table 5-5 shows the desired specifications, the reached performance parameters, obtained using the optimization platform, and the electric simulated corresponding values, which are obtained from electrical simulation using SPECTRE.

Table 5-5 Optimized and post-simulated results of the circuit performance parameters, in the frequency-domain.

	Desired specifications	Optimized Results	Simulated results
A_{V0}	80 dB	86.6 dB	84 dB
GBW	300 MHz	203.3 MHz	319 MHz
F_{p1}	10 kHz	10.73 kHz	15.73 kHz
PM	60º	56.17º	60.5º
I_{total}	1000 µA	548.1 µA	545 µA
OS	900 mV	916.5 mV	995.9 mV

The optimum size value for the compensation capacitance, C_C, is 500.86 fF.

5.4.3 Time-domain optimization

In this optimization version, the main performance parameter, as discussed in chapter 4, is the settling time (ST), for a given settling error, in the closed-loop step response of the circuit. As opposed to the previously examples, in this case, the ST is obtained by transient simulation, based on the NGSPICE (SPICE-like open-source) simulator source-code that is integrated in the platform. Since the distributed processing is included, the increased quantity of computing resources makes the transient simulation a suitable option, since the optimization processing time is overcome with the multiple, distributed (in parallel) processing units. Moreover, the accuracy is guaranteed by the complex and complete models of, both, transient simulation and devices models (e.g. BSIM3v3), used by the electrical simulator. The amplifier was also optimized using the time-domain where the following goals where:

- A ST of approximately 150 ns within an error of 24 µV, for an output step response of 100 mV.
- The minimum power dissipation, I_{total};
- The maximum output voltage swing, OS;
- A compensation capacitance, *Cc*, around 0.5~0.6 pF, due to thermal noise constraints.

The fitness value of each individual is computed according to:

$$fitness = f_{\mathrm{ST}} \cdot f_{\mathrm{OS}} \cdot f_{Itotal} \tag{5.86}$$

The partial fitness's are defined as:

- f_{ST} the ST @ 24 μV error for an output voltage of 100 mV, to be minimized;
- f_{OS} the voltage range available for the output, to be maximized;
- f_{Itotal} the total current drawn from V_{DD}, to be minimized.

Table 5-6 shows the desired specifications, the results obtained using the optimization platform, and the electrical simulation verification results. The frequency-domain specifications: A_{v0}, and GBW; are computed the same way, as for the frequency-domain optimization.

Table 5-6 Optimized and post-simulated results of the circuit performance parameters, in the time-domain

	Desired specifications	**Optimized Results**	**Simulated results**
A_{V0}	-	96.9 dB	88.4 dB
GBW	-	68.41 MHz	86.5 MHz
I_{total}	50 μA	100.1 μA	99.57 μA
OS	800 mV	739 mV	850 mV
ST	150 ns@0.024%	83.7 ns@0.024%	86.7 ns@0.024%

The optimum size value for the compensation capacitance, C_C, is 510 fF.

Comparing the results of the tables Table 5-5 and Table 5-6 by computing the figure-of-merit (FoM)[99], using:

$$\mathrm{FoM} = \frac{\mathrm{GBW} \cdot C_L}{P_{total}} \quad [\mathrm{MHz \cdot pF/mW}] \tag{5.87}$$

one can conclude the time-domain optimization reached a better FoM, 2278 MHz.pF/mW, than frequency-domain, 1236 MHz.pF/mW, and, consequently, one can say that time-domain approach is a better approach for optimization.

5.4.4 Simulation results

Figure 5-23 shows the Bode diagrams for gain and phase, obtained through SPECTRE electrical simulations, for the case of the amplifier instance optimized in the frequency-domain. The simulated amplifier achieved a DC gain of 84 dB, a GBW of 319 MHz, 60 ° PM (for the 4 pF load), an OS of 995.9 mV and a power dissipation of 654 μW (@ 1.2 V).

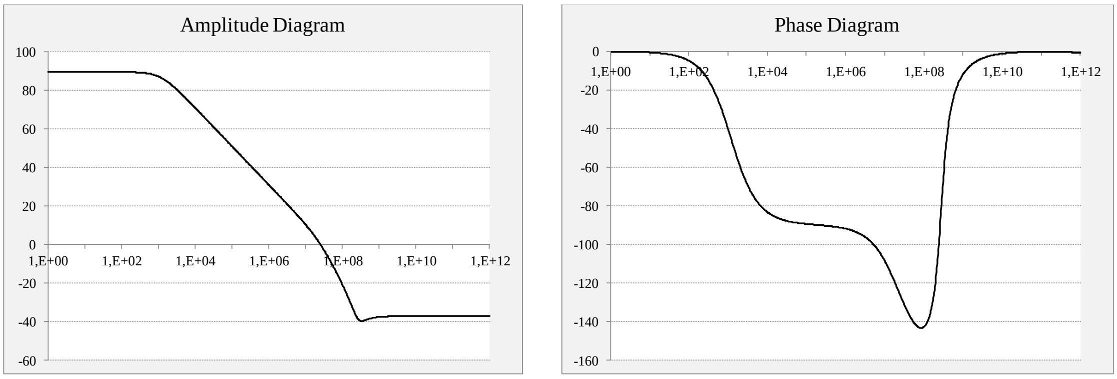


Figure 5-23 Simulated Bode diagrams of the two-stage self-biased inverter-based amplifier

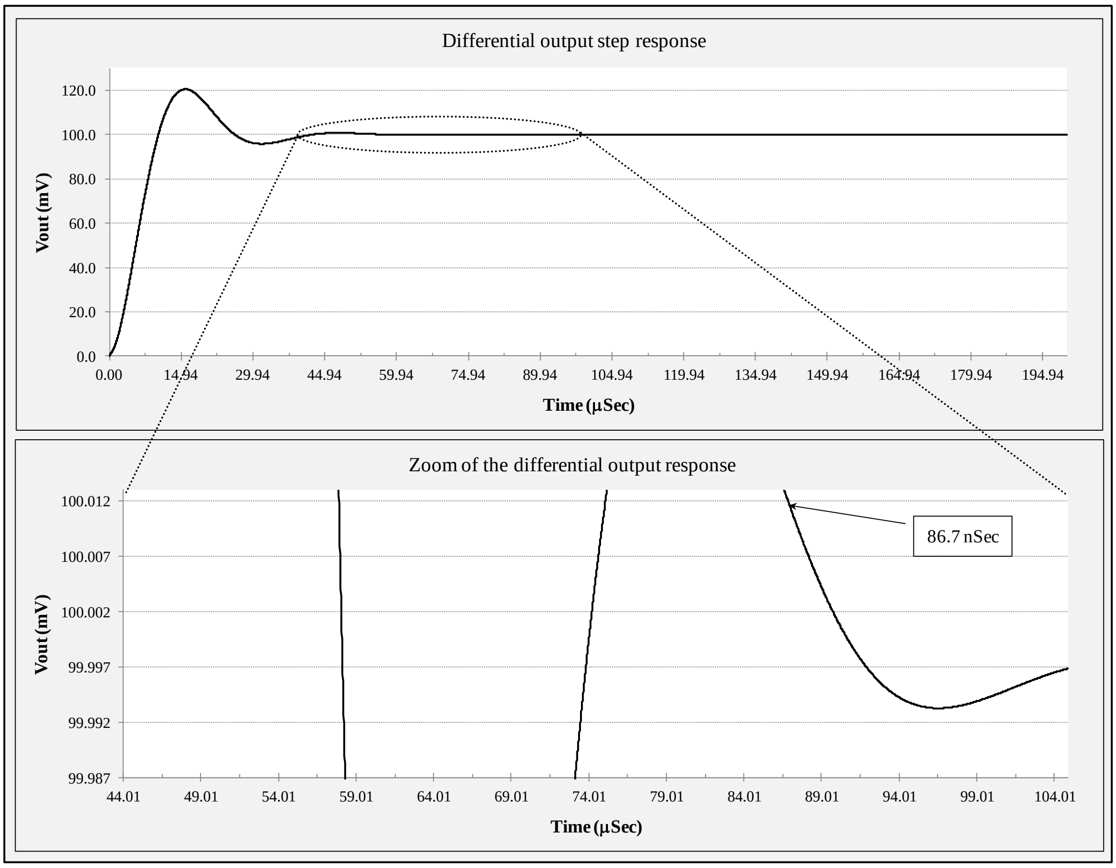


Figure 5-24 Simulated step response of the two-stage self-biased inverter-based amplifier

For the case of the circuit optimized in the time-domain, the differential response to an input differential step of 100 mV is shown in Figure 5-24. It also illustrates that the response quickly converge to the final voltage, and an over-shoot with a maximum of, approximately, 20 mV. The "86.7" mark demonstrates the point where the response enters the settling error, 86.7 ns after step rise.

5.4.5 Layout Design

In order to have experimental measurements to support the electrical simulated results, a prototype integrated circuit was fabricated. Figure 5-25 shows the amplifier layout with the PADs, and superimpose the floorplan diagram blocks of the amplifier circuit are marked (white). It contains the two differential inputs and the voltage source, V_{DD}, at the top of the floorplan. The outputs and the V_{SS} are located at the bottom. The bias voltage control of the common-mode is placed at the right side. Each signal input and the V_{CMI} have a circuit for ESD protection, and bidirectional. The output signals also have ESD protection, but unidirectional. The power lines have a larger width to reduce resistance.

Figure 5-26 presents the amplifier layout, and superimpose the floorplan diagram blocks marked (white). Inside amplifier block the PMOS transistors are placed on top of the NMOS transistors. The compensation capacitors are located at each side and for symmetry purposes, e.g. mismatch, the capacitor C_{CX} is divided into blocks, at each side of the amplifier block. Each of the two CMFB circuits also placed apart on both sides. As show in the amplifier layout, Figure 5-26, the largest blocks are the compensation capacitances, C_C and C_{CX}. Total silicon area occupied by the circuit, including PADS, is: 331 μm x 291 μm. The amplifier occupies an area of: 179 μm x 66 μm.

Table 5-7 presents a performance comparison for various single-stage and multi-stage class A and A/B amplifiers. Notice that, although [99] and [100] achieve a better efficiency, FoM (5.87), they were designed targeting heavy loads and very low GBW. Hence, these amplifiers were biased with very low biasing and quiescent currents. If a GBW above 100 MHz is required, the reported efficiencies cannot be reached. Moreover, the amplifier reported in [100] has a very low DC gain. Using time-domain optimization technique, a higher FoM is achieved. Although less GBW is achieved, also less power dissipation and faster settling response is obtained.

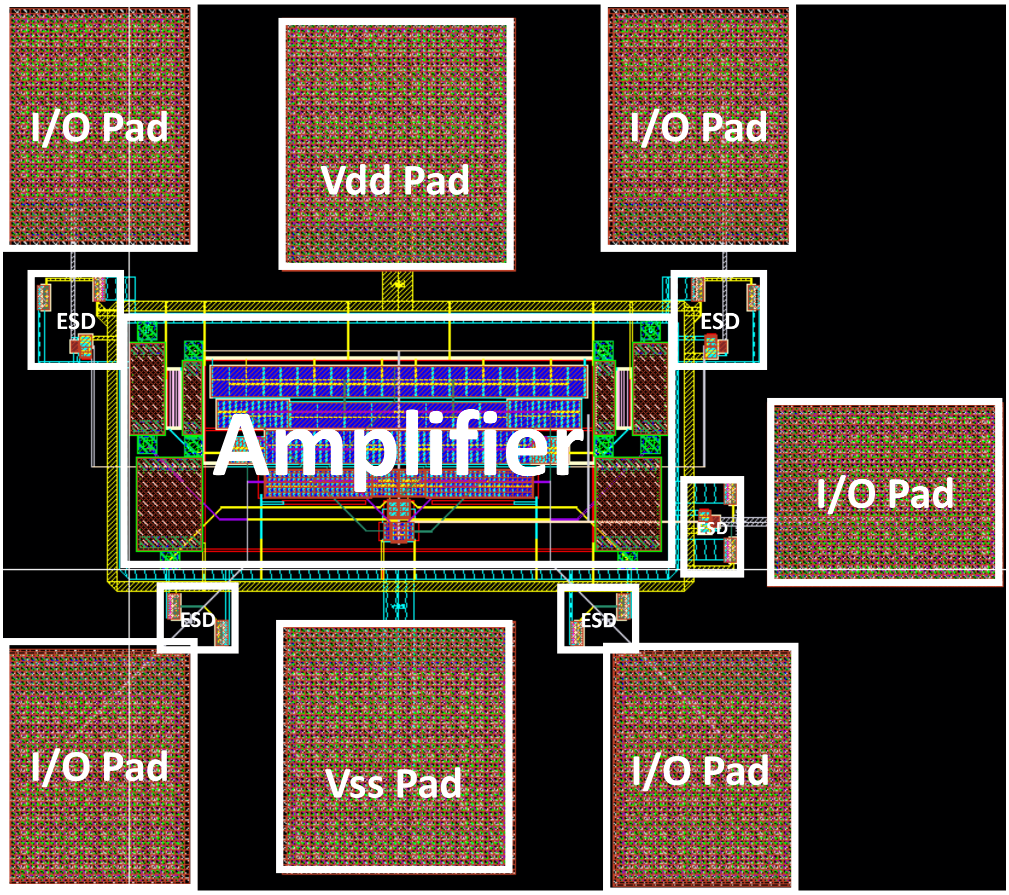


Figure 5-25 Complete circuit floorplan layout

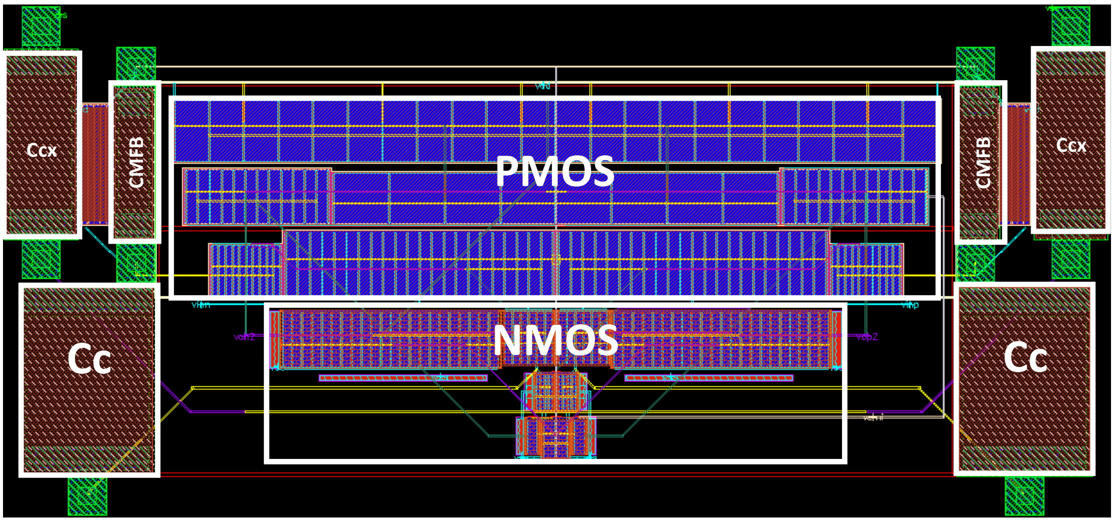


Figure 5-26 Layout of the proposed new two-stage self-biased inverter-based amplifier

Table 5-7 Performance comparisons of the simulated results

Ref.	**Nº Stages**	**Class**	**Tech.** (µm)	***C_L*** (pF)	**GBW** (MHz)	**PM** (º)	***A_{v0}*** (dB)	**Power** (mW@V_{DD})	**FoM** (MHz·pF/mW)
[99]	3	A/B	0.35	500	1.4	75	113	0.225@1.5	3111
[101]	2	A/B	0.5	50	12	N/A	14	1.05@1.5	571
[102]	2	A/B	0.25	4	165	65	68.5	5.8@1.2	114
[103]	2	A/B	0.35	5	7.3	44	99	0.123@0.8	297
[104]	2	A	0.25	4	500.4	62.6	88.4	2.6@2.5	770
[100]	2	A/B	0.5	25	11	N/A	45	0.06@2	4853
[105]	1	A	0.18	5.6	134.2	70.6	60.9	1.44@1.8	522
Frequency-optimization	**2**	**A**	**0.13**	**4**	**319**	**60**	**84**	**0.654@1.2**	**1950**
Time-optimization	**2**	**A**	**0.13**	**4**	**87**	**-**	**88**	**0.121@1.2**	**2876**

The two last (bottom) row's values are obtained through SPECTRE simulations.

5.5 Experimental results

The amplifier optimized in the time domain (with the lower GBW) was chosen to be integrated in a prototype which was fabricated in a 130 nm HS (high-speed) 1.2 V 1P-8M standard CMOS technology (L_{min} = 120 nm), all capacitors are implemented as Metal-Insulator-Metal (MIM) capacitors. Figure 5-27 shows the chip photograph highlighting the amplifier core and the on-chip continuous-time CMFB circuit (half on each side of the amplifier), which substitutes the SC CMFB circuit (Figure 5-20 (a)) of the output stage. This circuit is basically a continuous-time version of the circuit depicted in Figure 5-20 (a) and it comprises, on chip, two capacitors of 100 fF and two resistors of 50 kΩ which inevitably reduced the DC gain of the amplifier. The area of the amplifier including CMFB, C_C, and C_{CM} is approximately 179 × 69 (µm^2).

The test setup used to characterize the performance of the amplifier is shown in Figure 5-28. This test setup was replicated from a Texas Instruments fully-differential amplifier THS4521D evaluation module[106]. In Figure 5-28, the solid lines represent the circuit used for transient analyses, the dashed lines correspond to the circuit used for AC analyses, and the dot-dash line indicates how the noise was measured. To drive the feedback resistors to allow a closed-loop testing schema, it is necessary to employ two buffers (one for each output). The AD8000 were used for the buffers. The input CM voltage and V_{CMO} are set to 550 mV.

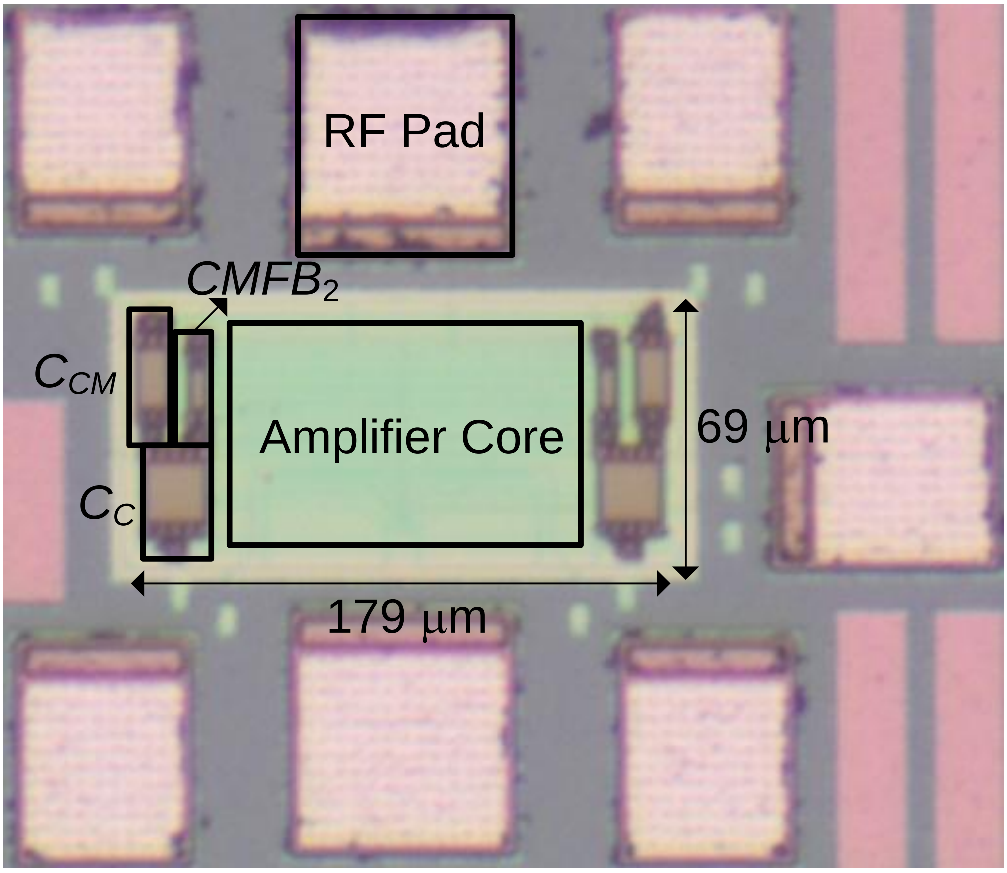


Figure 5-27 Chip photograph with amplifier core area, C_C, C_{CM}, and CMFB2

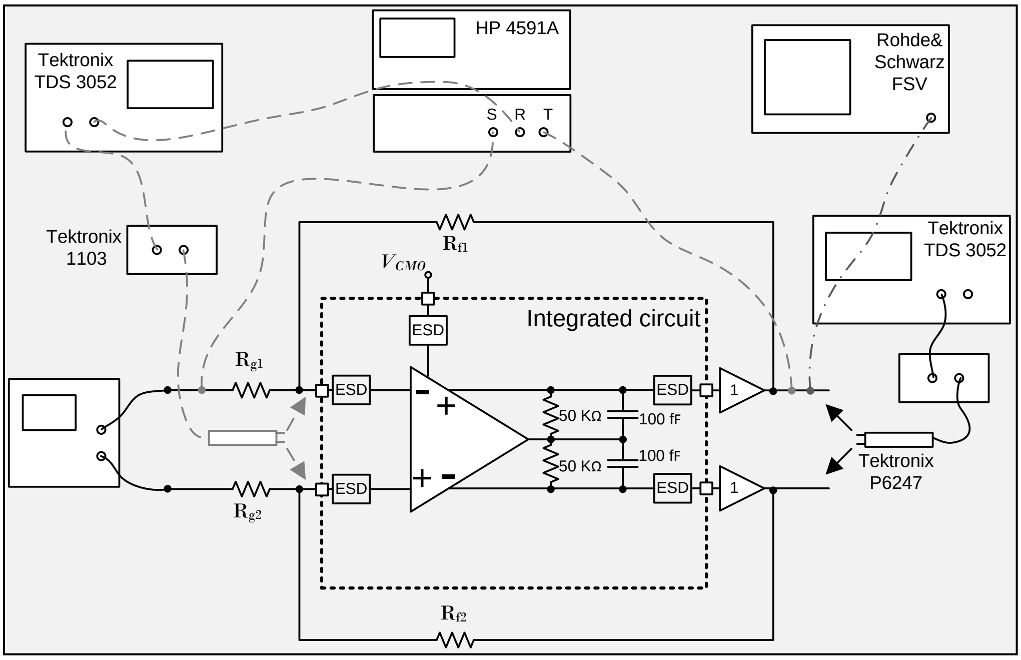


Figure 5-28 Measurement setup used for the amplifier characterization

The test equipment used is described as follows: for frequency response measurements a HP 4195A Network Analyser and a Tektronix P6247 Differential Active Probe were used; regarding step response measurements, the input signals were produced with a Tektronix AWG510 and the output signals were read with the mentioned active probe and a Tektronix TDS3052 oscilloscope; finally, a Rohde&Schwarz FSV Signal Analyzer was used for noise measurements.

The measured open-loop gain and phase Bode diagrams of the amplifier are shown in Figure 5-29. For these measurements the amplifier was in unity gain configuration and the active probe was connected to the amplifier's inputs. The AC response was then measured between the output of the setup and amplifier's inputs. Due to this setup, the amplifier's inputs were loaded with an extra 200 kΩ || 1 pF impedance, while the amplifier's outputs were loaded with the RF pad, the PCB trace and the input impedance of the AD8000 (≈ 2 MΩ || 3.6 pF). These extra loads degraded the AC response, especially the phase margin and the unity gain frequency, which were measured to be less than 45° and 30.4 MHz, respectively. The gain-bandwidth product was extrapolated to be around 35 MHz. Given this phase margin, it is highly probable that the step response show some ringing. Regarding the DC gain of the amplifier, over 71 dB was measured. The large gain of the amplifier made it even more difficult to measure. This also explains the inaccuracy of Figure 5-29 at low frequencies, especially in the phase diagram.

To measure the small signal step response, the loop of the amplifier was closed with a gain of two and a square wave signal with 50 mVpp (100 mVpp at output) at 1 MHz was applied to the amplifier's input, with the result shown in Figure 5-30. Even with a closed-loop gain of two, the amplifier denotes some oscillation (mainly due to the unexpected larger output capacitance). Measuring the ST proved to be a difficult task given the limited (8-to-9 bits) vertical resolution of the oscilloscope, but at 1 % error, the ST was measured to be approximately 154 ns.

Table 5-8 presents a summary of the key measured parameters, as well as, a performance comparison for various single-stage and multi-stage amplifiers. The criteria chosen for the amplifiers were GBWs above 30 MHz and DC gains higher than 60 dB.

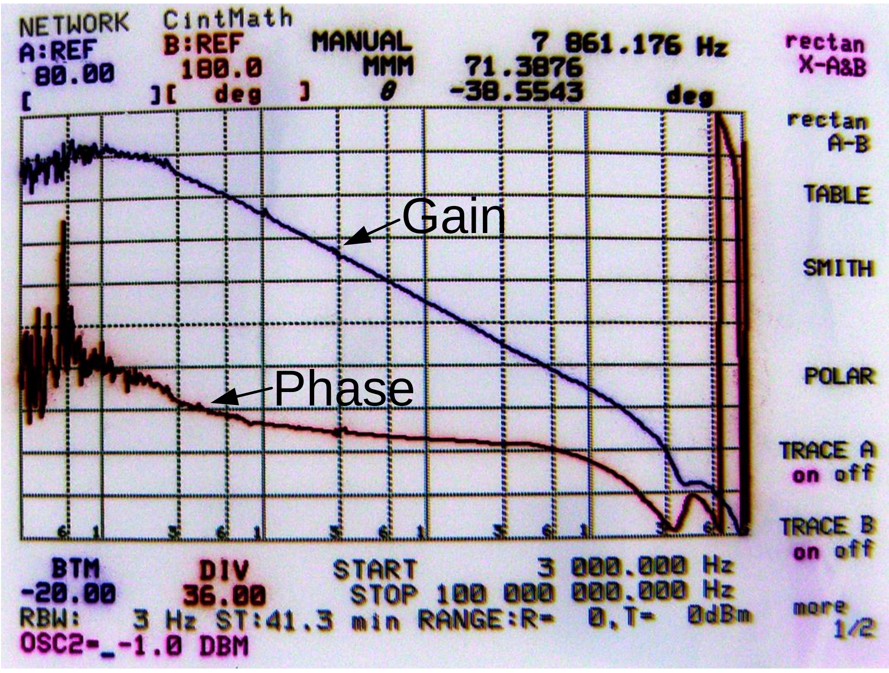


Figure 5-29 Amplifier open-loop gain and phase Bode diagrams

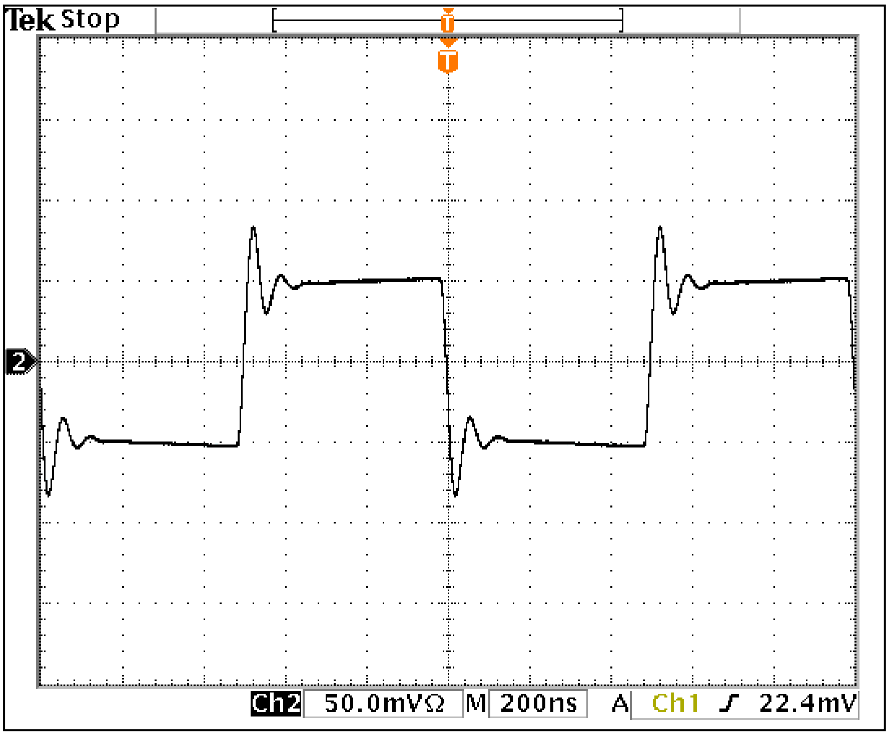


Figure 5-30 Small signal step response

Table 5-8 Performance comparisons and key performance summary of the amplifier

	[102]	[107]	[105][b)]	[108]	[16]	**This work** [b)]
Tech. (μm)	0.25	0.18	0.18	0.18	0.13	**0.13**
C_L (pF)	5	1.75	5.6	1	4	**>5.6**
GBW (MHz)	165	160	134.2	660	319	**35**
PM (º)	65	N/A	70.6	73	60.5	**>45**[a)]
A_{V0} (dB)	68.5	74	60.9	80	84	**>70**
T_{Serror} (%)	0.01	N/A	1	0.01	N/A	**1**
T_S (ns@Vpp.diff)	11@0.8	N/A	11.2@0.1	2.2@0.1	N/A	**134@0.1**
Power (mW@V_{DD})	5.8@1.2	0.362@1.8	1.44@1.8	3.8@1.8	0.654@1.2	**0.11@1.2**
FoM (MHz·pF/mW)	114	772	522	173	1951	**1750**

a) assuming a closed-loop gain of two b) measured results

5.6 Conclusions

This chapter presented four practical examples, and respective results, that validate the time-domain methodology as a new and efficient method to design and optimize topologies of amplifiers. As shown, the proposed approach and tools developed are suitable to handle different and complex amplifier topologies, with any number of elements and an unlimited number of poles and zeros.

The first example described the method for designing and optimizing, in the time-domain, low-voltage amplifiers with enhanced performance. The complexity of topology is augmented by adding an auxiliary amplifier for active biasing purposes. This added feature intends to make possible an amplifier to reach, simultaneously, high open-loop gains and fast settling responses without increasing the power dissipation. Although the extra degree of freedom increased the circuit analysis, the circuit optimization process was not affected.

Next, a new and optimized compensation schema for two-stage amplifiers was shown. It enforced the compensation theory first described by Ahuja [65] and Yao (Improved-Ahuja) [66], and illustrates that the best results are achieved by using a (new) hybrid compensation type, i.e. an unbalanced mixture of both. The two compensation capacitances should be, therefore, asymmetrically distributed: 40% (Ahuja) - 60% (Improved-Ahuja).

The third example conjugated the two previous concepts on a single amplifier/optimization circuit. The number of compensation paths and the gain-boosting techniques employed increased the transfer function order, and increases the number of zeros and poles. The total number of elements to size also is augmented.

The last section presented a novel two-stage fully-differential CMOS amplifier comprising two self-biased inverter stages. The amplifier is completely self-biased, precluding any biasing circuitry. Although the amplifier relies on a quasi-class-A topology, the optimization sizing reached a high efficiency and optimum compensation circuit, comparable with class AB. The two optimizations presented: frequency-domain and time-domain approaches; permitted to asses that a time-domain approach reaches an optimum circuit sizing. A prototype designed in a 130 nm HS (high-speed), 1.2 V, CMOS technology (L_{min} = 120 nm) was fully designed. Although many difficulties were encountered during the measurement phase, due to the employed setup, the experimental results showed that a good energy-efficiency is achievable.

6 Conclusions and Future Work

This thesis discussed the problem of optimization and automatic sizing of analog circuits, focusing in particular in CMOS amplifier design. A novel methodology was introduced based on time-domain analysis of amplifiers. This optimization design methodology was implemented in an optimization platform, using genetic algorithms, and based on distributed computing.

It was demonstrated that the presented optimization methodology is able to handle the high complexity demanded by high performance circuits. Furthermore, the main advantage of this new time-domain methodology is that, when a given settling-error is reached within the desired settling-time, it is automatically guaranteed that the amplifier has enough open-loop gain, A_{OL}, output-swing (OS), slew-rate (SR), closed loop bandwidth and closed loop stability. The described procedure to extract the time-domain step response, based on the open-loop transfer function of the circuit, is relatively straightforward. Moreover, it was demonstrated, throughout several practical examples (chapter 5) that, this method can handle complex circuits, with complex transfer functions, with an unlimited number of zeros and poles.

The flexibility of the platform allows working on different levels of abstraction. This means it can either choose the best compensation schema in a multi-stage amplifier, or, for example, find the optimum specifications for system blocks.

Two options are available to compute the time-domain step response of the circuit: a) based on the inverse Laplace transform applied to the transfer function of the circuit, multiplied, symbolically, by $1/s$ (unitary input step) and the DC bias operating point computed by means of accurate device models; b) based on the transient response of the circuit, estimated by the open-source source-code of the electrical simulator NGSPICE, which was successfully integrated into the developed platform. The equation-based approach (option a)) runs faster and accurately, but the initial setup requires the extraction of the expression of the closed-loop step response. On the other hand, the simulation-based approach (option b)) is straightforward to setup and accurate, but the run-time is higher than the previous option, since a set of several transient points need to be computed. Moreover, it covers wider range of amplifiers.

In order to improve the circuits yield, process, voltage source and operating temperature (PVT) variations were addressed during the optimization process. Inside each circuit performance parameter evaluation, a PVT evaluation loop is executed.

Distributed/parallel processing is one key concept of this platform. The genetic algorithm is well suited for distribute processing work. This feature was explored allowing the individuals/circuits to be analyzed on different processors. This allows a substantial reduction of the processing time. Since more processing capacity is available, it permits a large design space to be examined. Another advantage is the computing hardware reuse. The distributed platform is able to optimize circuits independently of the type of computing hardware available. The slaves can run on simple, out-dated configuration computer, desktop computer, or on a state-of-art multi-processor, multi-core computer. This means it is a low-cost solution, considering hardware issues, which can delivery optimum results.

The proposed time-domain methodology and the implemented platform have been assessed, experimentally, through silicon results of an integrated IC prototype (*two-stage fully-differential inverter-based self-biased CMOS amplifier with high efficiency*). Moreover, several amplifiers prototypes were optimized as building blocks of analog-digital-analog converters, e.g. "*A 14-bit 1.5 Msample/s two-stage algorithmic ADC with a power-and-area efficiency better than 0.5 pJmm2 per conversion*", which were fully verified through electrical simulation, using HSPICE or CADENCE SPECTRE.

The practical examples demonstrated that the platform and methodology are extremely useful to assist, and even replace the manual analog design flow. The main focus is to facilitate the analog design flow, at circuit level, such as circuit sizing and design trade-offs. It is intended to liberate the designer from error-prone and repetitive tasks.

6.1 Future Work

Only a small portion of research and development has been reaching the analog circuit designers community. The presented work focuses only a small part of the design automation of the analog circuit design flow: sizing optimization. Improvements should be considered, and new developments, in different areas, are necessary.

An improvement to consider is to bring into the sizing stage, the knowledge of the layout techniques to be applied. There some layout techniques that helps to get improved circuit layouts in order to reduce area and parasitic capacitances. If in an early stage, the engineer, or the platform, knows which of those techniques will be applied to the transistors of the circuit, it can result in a more accurate design in order to obtain enhanced circuit performances. For instance, using the knowledge of the multi-fingered transistors some areas and perimeters can be reduced, which will attenuate the parasitic capacitances associated to the device. In the optimization process, the reducing of the parasitic capacitances can lead to improved optimization results.

Another issue to consider in the future is the improvement of the distributed processing management. Having more autonomy among the different processing units, and exchange more data, in between the different processing units, during the optimization process, instead only to a centralized master. Moreover, implement the local parallel processing, instantiating multiple concurrent threads (Multithread) to evaluate more individuals, locally.

Regarding other areas, other than sizing, one could extend this work into optimization on system level and selection of the most appropriated topology for the specifications. Considering some sort of co-optimization: system and circuit level.

Extend the knowledge of genetic algorithm optimization to the layout placement, using predefined layout cells of elementary circuit blocks, e.g. pair-differential.

An essential item, on the near future, is the integration of this work within the major commercial tool's frameworks, through standard interfaces

Appendix A. Example of the Persisted Optimization Data

This appendix presents the format of the optimization progress data file. During the optimization process, at the end of each generation, the platform persist some progress data. This persisted data permits the post-analysis of the search evolution.

For simplicity and compatibility, for example, with Microsoft Excel, the file format selected is according to the Tab-Seprated-Values (TSV).

Each row in the file saves the data for each completed generation. The number of columns is variable according to the design being optimized, e.g. number of circuit performance parameters.

Generally, the first *n* columns contain the values of the indicators, of the best individual, that are considered to the fitness classification. The next two columns hold the fitness value computed for the best individual, and the median value of all individual's fitness of the present generation. The following column indicates if the individuals evaluations is equation-based or electrical simulation-based. Next, there is the column of the total sum of the elapsed time used by individual evaluation, and the median value. After that, appears the column with the flag that specifies the stop criteria. Finally, the last set of columns holds the values of the chromosome of the best individual.

The next page shows an example of the persisted data in a TSV file format.

```
#           Itotal       OS           SetTime      Vout         Gain(DB)     NoiseExcess          Fitness Max          Fitness Median
            Eq/Sim       IndTimeTotal              IndTimeMedian             GAStop       Chromossome...

1           6.26914E+03               8.220E-01    2.4343E+01                5.99948E-01          7.381E+01 3.340E+00 2.05330E-06
            1.61000E-06               SIM          1.190E+02 1.190E+00 NORMAL  1.32484E+03        1.33810E+03
            1.35714E+03               7.62222E+01               6.75873E-01               2.81197E+02               2.39065E-01
            5.20757E+02               5.67863E-01               7.15507E+02               7.41343E-01               9.10256E+01
            4.35165E-01               9.44689E+01               2.00317E-01               2.67460E+02               1.03330E+00
            2.05165E+03               2.17988E-01               1.02711E+03               4.17778E-01               2.55922E+00
            7.36313E+00               1.17568E+01               1.16806E+01               1.48806E+01               2.93548E+00
            2.61290E+00               1.71685E+01               7.98864E+02               3.89634E+02

2           6.26914E+03               8.220E-01    2.4343E+01                5.99948E-01          7.381E+01 3.340E+00 2.05330E-06
            1.58737E-06               SIM          2.370E+02 2.370E+00 NORMAL  1.32484E+03        1.33810E+03
            1.35714E+03               7.62222E+01               6.75873E-01               2.81197E+02               2.39065E-01
            5.20757E+02               5.67863E-01               7.15507E+02               7.41343E-01               9.10256E+01
            4.35165E-01               9.44689E+01               2.00317E-01               2.67460E+02               1.03330E+00
            2.05165E+03               2.17988E-01               1.02711E+03               4.17778E-01               2.55922E+00
            7.36313E+00               1.17568E+01               1.16806E+01               1.48806E+01               2.93548E+00
            2.61290E+00               1.71685E+01               7.98864E+02               3.89634E+02

3           6.26914E+03               8.220E-01    2.4343E+01                5.99948E-01          7.381E+01 3.340E+00 2.05330E-06
            1.59618E-06               SIM          3.560E+02 3.560E+00 NORMAL  1.32484E+03        1.33810E+03
            1.35714E+03               7.62222E+01               6.75873E-01               2.81197E+02               2.39065E-01
            5.20757E+02               5.67863E-01               7.15507E+02               7.41343E-01               9.10256E+01
            4.35165E-01               9.44689E+01               2.00317E-01               2.67460E+02               1.03330E+00
            2.05165E+03               2.17988E-01               1.02711E+03               4.17778E-01               2.55922E+00
            7.36313E+00               1.17568E+01               1.16806E+01               1.48806E+01               2.93548E+00
            2.61290E+00               1.71685E+01               7.98864E+02               3.89634E+02

(….)

47          6.40735E+03               7.031E-01    1.4900E+01                6.54303E-01          8.787E+01 4.197E+00 3.35433E-06
            1.42791E-06               SIM          5.574E+03 5.574E+01 NORMAL  1.33275E+03        5.71453E+02
            2.36923E+03               1.32557E+02               1.26571E+00               1.95726E+02               1.35480E+00
            6.45788E+02               3.33626E-01               7.32723E+02               8.56606E-01               6.77778E+01
            5.81099E-01               3.46557E+02               1.22000E+00               2.45910E+02               3.55824E-01
            1.69377E+03               1.25310E+00               1.07692E+03               7.65397E-01               4.75116E+00
            7.68742E+00               9.43590E-01               7.60733E+00               7.07985E+00               3.00000E+00
            1.51613E+00               9.43956E+00               2.98926E+02               3.71770E+02

48          6.40735E+03               7.031E-01    1.4900E+01                6.54303E-01          8.787E+01 4.197E+00 3.35433E-06
            1.37837E-06               SIM          5.691E+03 5.691E+01 NORMAL  1.33275E+03        5.71453E+02
            2.36923E+03               1.32557E+02               1.26571E+00               1.95726E+02               1.35480E+00
            6.45788E+02               3.33626E-01               7.32723E+02               8.56606E-01               6.77778E+01
            5.81099E-01               3.46557E+02               1.22000E+00               2.45910E+02               3.55824E-01
            1.69377E+03               1.25310E+00               1.07692E+03               7.65397E-01               4.75116E+00
            7.68742E+00               9.43590E-01               7.60733E+00               7.07985E+00               3.00000E+00
            1.51613E+00               9.43956E+00               2.98926E+02               3.71770E+02



Global (Sec)         Global (Tics)        NG total     NG median            GA total     GA median

5.77500E+03          1.47300E+03          5.77200E+03          1.17796E+00          5.77300E+03          5.77300E+03
```

Appendix B. Example of the Optimized SPICE-like Netlist

At the end of the optimization process, the platform persist the optimum circuit netlist file compliant with the SPICE-like format. The following is an example of a folded cascade amplifier.

```
 cmb n4b n2b 0.000fF
cma n4a n2a 0.000fF
cbb n4b n3b 4000.000fF
cba n4a n3a 4000.000fF
cab n4b n1b 4000.000fF
caa n4a n1a 4000.000fF
clb n4b vss 2000.000fF
cla n4a vss 2000.000fF
ib2 vb21 vss 0.124mA
ib1 vb3 vss 0.124mA
m8 n8 vb8 vss vss n_12_hsl130e w=1076.923u l=0.765u
m6b n4b n2b n8 vss n_12_hsl130e w=245.910u l=0.356u
m6a n4a n2a n8 vss n_12_hsl130e w=245.910u l=0.356u
m5b n3b vb5 vss vss n_12_hsl130e w=346.557u l=1.220u
m5a n3a vb5 vss vss n_12_hsl130e w=346.557u l=1.220u
m4b n2b vb4 n3b vss n_12_hsl130e w=67.778u l=0.581u
m4a n2a vb4 n3a vss n_12_hsl130e w=67.778u l=0.581u
m1b n1b vib n0 vss n_12_hsl130e w=195.726u l=1.355u
m1a n1a via n0 vss n_12_hsl130e w=195.726u l=1.355u
m0 n0 vb0 vss vss n_12_hsl130e w=132.557u l=1.266u
mb19 vb0 vb0 vss vss n_12_hsl130e w=13.256u l=1.266u
mb16 np5 vb5 vss vss n_12_hsl130e w=143.682u l=1.220u
mb15 vb71 vb4 np5 vss n_12_hsl130e w=28.101u l=0.581u
mb13 np6 vb5 vss vss n_12_hsl130e w=34.656u l=1.220u
mb12 vb5 vb4 np6 vss n_12_hsl130e w=6.778u l=0.581u
mb9 vb4 vb4 vss vss n_12_hsl130e w=1.709u l=0.581u
mb6 vb8 vb8 vss vss n_12_hsl130e w=107.692u l=0.765u
m7b n4b vb72 vdd vdd p_12_hsl130e w=1693.773u l=1.253u
m7a n4a vb72 vdd vdd p_12_hsl130e w=1693.773u l=1.253u
m3b n2b vb3 n1b vdd p_12_hsl130e w=732.723u l=0.857u
m3a n2a vb3 n1a vdd p_12_hsl130e w=732.723u l=0.857u
m2b n1b vb22 vdd vdd p_12_hsl130e w=645.788u l=0.334u
m2a n1a vb22 vdd vdd p_12_hsl130e w=645.788u l=0.334u
mb18 vb0 vb3 np7 vdd p_12_hsl130e w=170.886u l=0.857u
mb17 np7 vb21 vdd vdd p_12_hsl130e w=69.531u l=0.334u
mb14 vb71 vb71 vdd vdd p_12_hsl130e w=169.377u l=1.253u
mb11 vb5 vb3 np4 vdd p_12_hsl130e w=73.272u l=0.857u
mb10 np4 vb21 vdd vdd p_12_hsl130e w=29.813u l=0.334u
mb8 vb4 vb3 np3 vdd p_12_hsl130e w=73.272u l=0.857u
mb7 np3 vb21 vdd vdd p_12_hsl130e w=29.813u l=0.334u
mb5 vb8 vb3 np2 vdd p_12_hsl130e w=607.570u l=0.857u
mb4 np2 vb21 vdd vdd p_12_hsl130e w=247.211u l=0.334u
mb3 vb21 vb3 np1 vdd p_12_hsl130e w=158.715u l=0.857u
mb2 np1 vb21 vdd vdd p_12_hsl130e w=64.579u l=0.334u
mb1 vb3 vb3 vdd vdd p_12_hsl130e w=25.677u l=0.857u
```

7 Bibliography

[1] K. Kundert, H. Chang, D. Jefferies, G. Lamant, E. Malavasi, F. Sendig; "Design of mixed-signal systems-on-a-chip", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 19, pp. 1561-1571, 2000.

[2] C. Toumazou, G. Moschytz, B. Gilbert; "Trade-Offs in Analog Circuit Design", Springer, ISBN 1402080468, 2004.

[3] J. Williams; "The art and science of analog circuit design", Elsevier Science Publishers, ISBN 0750670622, 1998.

[4] IBS Corporarion, "Industry Reports", 2002.

[5] Y. Cheng, C. Hu; "MOSFET Modeling and BSIM3 User's Guide", Kluwer Academic Publishers, ISBN 0792385756, 1999.

[6] X. Xi, K. M. Cao, H. Wan, M. Chan, C. Hu; "BSIM4.2.1 MOSFET Model - Users' Manual", The Regents of the University of California, 2001. Available: http://www-device.eecs.berkeley.edu/~bsim3/bsim4.html

[7] G. Gildenblat, X. Li, H. Wang, W. Wu, R. van Langevelde, A. J. Scholten, G. D. J. Smit, D. B. M. Klaassen; "Introduction to PSP MOSFET Model", Technical Proceedings of the 2005 Workshop on Compact Modeling, 2005. Available: http://pspmodel.asu.edu/

[8] G. E. Moore; "Cramming more components onto integrated circuits", *Proc. of the IEEE*, pp. 82-85, 1965.

[9] N. Mokhoff; "Chips fall where they may in the 'More than Moore' era", EE Times, 2008.

[10] B. Vaz, R. Costa, N. Paulino, J. Goes, **R. Tavares**, A. Steiger-Garção; "A general-purpose kernel based on genetic algorithms for optimization of complex analog circuits", *Proc. of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems*, Vol. 1, pp. 83-86, Dayton, OH , USA, 2001.

[11] **R. Santos-Tavares**, N. Paulino, J. Higino, J. Goes, J. P. Oliveira; "Optimization of multi-stage amplifiers in deep-submicron CMOS using a distributed/parallel genetic algorithm",

Proc. of the 2008 IEEE International Symposium on Circuits and Systems, pp. 724-727, Seattle, USA, 2008.

[12] *Many Authors*, “Ngspice mixed-mode/mixed-level circuit simulator User’s Manual”, The Regents of the University of California, 2004. Available: http://ngspice.sourceforge.net/

[13] **R. Tavares**, B. Vaz, J. Goes, N. Paulino, A. Steiger-Garção; "Design and optimization of low-voltage two-stage CMOS amplifiers with enhanced performance", *Proc. of the 2003 IEEE International Symposium on Circuits and Systems*, Vol. 1, pp. 197-200, Bankok, Thailand, 2003.

[14] **R. Santos-Tavares**, N. Paulino, J. Goes, J. P. Oliveira; "Optimum Sizing and Compensation of Two-Stage CMOS Amplifiers Based On a Time-Domain Approach", *Proc. of the 13th IEEE International Conference on Electronics, Circuits and Systems*, pp. 533-536, Nice, France, 2006.

[15] B. Esperança, J. Goes, **R. Tavares**, A. Galhardo, N. Paulino, M. M. Silva; "Power-and-area efficient 14-bit 1.5 MSample/s two-stage algorithmic ADC based on a mismatch-insensitive MDAC", *Proc. of the 2008 IEEE International Symposium on Circuits and Systems*, pp. 220-223, Seattle, USA, 2008.

[16] M. Figueiredo, E. Santin, J. Goes, **R. Santos-Tavares**, G. Evans; "Two-Stage Fully-Differential Inverter-based Self-Biased CMOS Amplifier with High Efficiency", *Proc. of the 2010 IEEE International Symposium on Circuits and Systems*, pp. 2828-2831, Paris, France, 2010.

[17] M. Figueiredo, **R. Santos-Tavares**, E. Santin, J. Ferreira, G. Evans, J. Goes., "A Two-Stage Fully-Differential Inverter-based Self-Biased CMOS Amplifier with High Efficiency" *submitted by invitation to IEEE Transactions on Circuits and Systems - Part I: Regular papers*, October 2010.

[18] B. Vaz, N. Paulino, J. Goes, R. Costa, **R. Tavares**, A. Steiger-Garção; "Design of low-voltage CMOS pipelined ADCs using 1 pico-Joule of energy per conversion", *Proc. of the 2002 IEEE International Symposium on Circuits and Systems*, Vol. 1, pp. 921-924, Arizona, USA, 2002.

[19] B. Vaz, J. Goes, N. Paulino; "A 1.5-V 10-b 50 MS/s time-interleaved switched-opamp pipeline CMOS ADC with high energy efficiency", *Proc. of the 2004 Symposium on VLSI Circuits*, pp. 432-435, 2004.

[20] N. Paulino, J. Goes, A. Steiger-Garção; "Design methodology for optimization of analog building blocks using genetic algorithms", *Proc. of the 2001 IEEE International Symposium on Circuits and Systems*, Vol. 5, pp. 435-438, Sydney, Australia, 2001.

[21] M. G. R. Degrauwe, O. Nys, E. Dijkstra, J. Rijmenants, S. Bitz, B. L. A. G. Goffart, E. A. Vittoz, S. Cserveny, C. Meixenberger, G. van der Stappen, H. J. Oguey; "IDAC: an interactive design tool for analog CMOS circuits", *IEEE Journal of Solid-State Circuits*, Vol. 22, pp. 1106-1116, 1987.

[22] F. El-Turky, E. E. Perry; "BLADES: an artificial intelligence approach to analog circuit design", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 8, pp. 680-692, 1989.

[23] R. Harjani, R. A. Rutenbar, L. R. Carley; "OASYS: a framework for analog circuit synthesis", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 8, pp. 1247-1266, 1989.

[24] Mentor Graphics Corporation, 1981.

[25] C. Toumazou, C. A. Makris, C. M. Berrah; "ISAID - A methodology for automated analog IC design", *Proc. of the 1990 IEEE International Symposium on Circuits and Systems*, Vol. 1, pp. 531-535, New Orleans, USA, 1990.

[26] H. Y. Koh, C. H. Sequin, P. R. Gray; "OPASYN: a compiler for CMOS operational amplifiers", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 9, pp. 113-125, 1990.

[27] F. Medeiro, F. V. Fernández, R. Domínguez-Castro, A. Rodríguez-Vázquez; "A Statistical Optimization-based Approach For Automated Sizing Of Analog Cells", *Proc. of the 1994 IEEE/ACM International Conference on Computer-Aided Design*, pp. 594-597, 1994.

[28] R. Phelps, M. Krasnicki, R. A. Rutenbar, L. R. Carley, J. R. Hellums; "Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 19, pp. 703-717, 2000.

[29] J. Rijmenants, T. Schwarz. J. Litsios, R. Zinszner; "ILAC: an automated layout tool for analog CMOS circuits", *IEEE Journal of Solid-State Circuits*, Vol. 24, pp. 417-425, 1989.

[30] N. Lourenço; "LAYGEN: Automatic Layout Generation of Analog ICs, From a System to Device Level Using both Hierarchical Template Descriptions and Intelligent Computing Techniques", Master Thesis, Dept. Electrical and Computer Engineering, Instituto Superior Técnico, Portugal, 2007.

[31] J. P. Harvey, M. I. Elmasry, B. Leung; "STAIC: an interactive framework for synthesizing CMOS and BiCMOS analog circuits", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 11, pp. 1402-1417, 1992.

[32] P. C. Maulik, L. R. Carley; "Automating analog circuit design using constrained optimization techniques", *Proc. of the 1991 IEEE International Conference on Computer-Aided Design*, pp. 390-393, Santa Clara, CA, USA, 1991.

[33] E. S. Ochotta, R. A. Rutenbar, L. R. Carley; "Synthesis of high-performance analog circuits in ASTRX/OBLX", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 15, pp. 273-294, 1996.

[34] G. Gielen, G. Debyser, K. Lampaert, F. Leyn, K. Swings, G. Van der Plas, W. Sansen, D. Leenaerts, P. Veselinovic, W. van Bokhoven; "An Analog Module Generator for Mixed

Analog/Digital ASIC Design", I*nternational Journal of Circuit Theory and Applications*, Vol. 23, pp. 269-283, 1995.

[35] M. del Mar, S. P. Boyd, T. H. Lee; "GPCAD: a tool for CMOS op-amp synthesis", *Proc. of the 1998 IEEE/ACM International Conference on Computer-Aided Design*, pp. 296-303, 1998.

[36] F. V. Fernandez, A. Rodriguez-Vazquez; "Symbolic analysis tools-the state of the art", *Proc. of the 1996 IEEE International Symposium on Circuits and Systems*, Vol. 4, pp. 798-801, Atlanta, USA, 1996.

[37] B. Johnson, T. Quarles, A.R. Newton, D.O. Pederson and A. Sangiovanni-Vincentelli; “SPICE3 Version 3f User's Manual”, University of California, Berkeley, 1992.

[38] Barcelona Design, 2004. Available: http://www.barcelonadesign.com.

[39] W. Nye, D. C. Riley, A. Sangiovanni-Vincentelli, A. L. Tits; "DELIGHT.SPICE: an optimization-based system for the design of integrated circuits", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 7, pp. 501-519, 1988.

[40] E. S. Ochotta, T. Mukherjee, R. A. Rutenbar, L. R. Carley; “Practical Synthesis of High-Performance Analog Circuits”, ISBN 0-7923-8237-4, Kluwer Academic Publishers, Norwell, USA, 1998.

[41] F. Medeiro, B. Perez-Verdu, A. Rodriguez-Vazquez, J. L. Huertas; "A vertically integrated tool for automated design of modulators", *IEEE Journal of Solid-State Circuits*, Vol. 30, pp. 762-772, 1995.

[42] A. Torralba, J. Chavez, L. G. Franquelo; "FASY: a fuzzy-logic based tool for analog synthesis", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 15, pp. 705-715, 1996.

[43] M. Krasnicki, R. Phelps, R. A. Rutenbar, L. R. Carley; “MAELSTROM: efficient simulation-based synthesis for custom analog cells”, *Proc. of the 36th Annual ACM/IEEE Design Automation Conference*, pp. 945-950, New Orleans, United States, 1999.

[44] Cadence Inc. Products: NeoCircuit, NeoCell. Available: http://www.cadence.com.

[45] G. Alpaydin, S. Balkir, G. Dundar; "An evolutionary approach to automatic synthesis of high-performance analog integrated circuits", *IEEE Transactions on Evolutionary Computation*, Vol. 7, pp. 240-252, 2003.

[46] M. Barros, J. Guilherme, N. Horta; "An evolutionary optimization kernel using a dynamic GA-SVM model applied to analog IC design", *Proc. of the 18th European Conference on Circuit Theory and Design*, pp. 32-35, Seville, Spain, 2007.

[47] Marti A. Hearst; "Support Vector Machines", *IEEE Intelligent Systems*, Vol. 13, pp. 18-28, 1998.

[48] L. T. Pillage, R. A. Rohrer; "Asymptotic waveform evaluation for timing analysis", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 9, pp. 352-366, 1990.

[49] K. Bult, G. J. G. M. Geelen; "A fast-settling CMOS op amp for SC circuits with 90-dB DC gain", *IEEE Journal of Solid-State Circuits*, Vol. 25, pp. 1379-1384, 1990.

[50] G. Beenker, J. Conway, G. Schrooten, A. Slenter; "Analog CAD for consumer IC's" in *Analog Circuit Design*, Huijsing, van der Plassche, and Sansen Eds., Kluwer Academic, Ch. 15, pp. 347-368, Boston, USA, 1993.

[51] T. Edwards; “Magic - A VLSI Layout system”. Available: http://vlsi.cornell.edu/magic/

[52] Cadence Virtuoso Layout Suite. Available: http://www.cadence.com/products/rf/layout_suite/pages/default.aspx

[53] Tanner Tools EDA, Analog Layout Accelaration - HiPer DevGen. Available: http://www.tannereda.com/devgenoffering

[54] J. Kuhn; "Analog Module Generators for Silicon Compilation", *VLSI System Design*, pp. 75-80, 1987.

[55] M. Wolf, U. Kleine, B. J. Hosticka; "A Novel Analog Module Generator Environment", *Proc. of the 1996 European conference on Design and Test*, pp. 388-392, Paris, France, 1996.

[56] L. R. Carley, G. Gielen, R. A. Rutenbar, W. Sansen; "Synthesis Tools for Mixed-Signal ICs: Progress on Frontend and Backend Strategies"; *Proc. of the 33rd Design Automation Conference*, pp. 298-303, Las Vegas, NV , USA, 1996.

[57] J. M. Cohn, D. J. Garrod, R. A. Rutenbar, L. R. Carley; "KOAN/ANAGRAM II: new tools for device-level analog placement and routing", *IEEE Journal of Solid-State Circuits*, Vol. 26, pp. 330-342, 1991.

[58] P. Khademsameni, M. Syrzycki; "A tool for automated analog CMOS layout module generation and placement", *Proc. of the 2002 IEEE Canadian Conference on Electrical and Computer Engineering*, Vol. 1, pp. 416-421, 2002.

[59] K. Lampaert, G. Gielen, W. Sansen; "A performance-driven placement tool for analog integrated circuits", *IEEE Journal of Solid-State Circuits*, Vol. 30, pp. 773-780, 1995.

[60] Silicon Integration Initiative, “OpenAccess Standard”. Available: http://www.si2.org/?page=977

[61] Steven M. Rubin; “Electric VLSI Design System”. Available: http://www.staticfreesoft.com/index.html

[62] Alliance - A free VLSI CAD system. Available: http://www-asim.lip6.fr/recherche/alliance/

[63] gEDA Project's Homepage. Available: http://www.gpleda.org/index.html

[64] W. Sansen, M. Steyaert, V. Peluso, E. Peeters; "Toward sub 1 V analog integrated circuits in submicron standard CMOS technologies", *Proc. of the 1998 IEEE International on Solid-State Circuits Conference*, pp. 186-187,435, San Francisco, CA, USA, 1998.

[65] Ahuja, B. K., "An improved frequency compensation technique for CMOS operational amplifiers", *IEEE Journal of Solid-State Circuits*, Vol. 18, pp. 629-633, 1983.

[66] L. Yao, M. Steyaert, W. Sansen; "Fast-settling CMOS two-stage operational transconductance amplifiers and their systematic design", *Proc. of the 2002 IEEE International Symposium on Circuits and Systems*, pp. 839-842, Phoenix-Scottsdale, AZ , USA, 2002.

[67] M. Yavari, O. Shoaei, F. Svelto; "Hybrid cascode compensation for two-stage CMOS operational amplifiers", *Proc. of the 2005 IEEE International Symposium on Circuits and Systems*, Vol. 2, pp. 1565-1568, Kobe, Japan, 2005.

[68] G. Gielen, R. A. Rutenbar; "Computer-aided design of analog and mixed-signal integrated circuits", *Proc. of the IEEE*, Vol. 88, pp. 1825-1854, 2000.

[69] Edinei Santin, Flávio Gil; "MSAAC - MATLAB Tool for Symbolic Analysis of Analog Circuits", *Internal Report*, Departamento de Engenharia Electrotécnica da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Portugal.

[70] F. Leyn, W. Daems, G. Gielen, W. Sansen; "A behavioral signal path modeling methodology for qualitative insight in and efficient sizing of CMOS opamps", *Proc. of the 1997 IEEE/ACM International Conference on Computer-Aided Design*, pp. 374-381, San Jose, CA , USA, 1997.

[71] Chung-Wen Ho, A. Ruehli, P. Brennan; "The modified nodal approach to network analysis", *IEEE Transactions on Circuits and Systems*, Vol. 22, pp. 504-509, 1975.

[72] G. Gielen, P. Wambacq, W. Sansen; "Symbolic analysis methods and applications for analog circuits: a tutorial overview", *Proc. of the IEEE*, Vol. 82, pp. 287-304, 1994.

[73] Y. Tsividis; "Operation and Modeling of the MOS Transistor", Oxford University Press, 2003.

[74] Christian C. Enz, Eric A. Vittoz; "Charge-Based MOS Transistor Modeling: The EKV Model for Low-Power and RF IC Design", Willey and Sons, 2006.

[75] D. M. Binkley, C. E. Hopper, S. D. Tucker, B. C. Moss, J. M. Rochelle, and D. P. Foty; "A CAD methodology for optimizing transistor current sizing in analog CMOS", *IEEE Transactions Computer-Aided Design of Integrated Circuits and Systems*, Vol. 22, pp. 225-237, 2003.

[76] David J. Comer, Donald T. Comer; "Using the Weak Inversion Region to Optimize Input Stage Design of CMOS Op Amps", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 51, pp. 8-14, 2004.

[77] Kenneth L. Ashley; "Analog electronics with LabView", Prentice Hall, 2003.

[78] Philip E. Allen, Douglas R. Holberg; “CMOS Analog Circuit Design”, Oxford University Press Press, 2002.

[79] P. R. Gray, P. J. Hurst, S. H. Lewis and R. G. Meyer; “Analysis and Design of Analog Integrated circuits. 4th Ed.”, John Wiley & Sons, 2010.

[80] Gabor C. Temes, Roubik Gregorian; “Analog MOS Integrated Circuits for Singal Processing”, John Wiley and Sons, 1986.

[81] D. Johns, K. Martin; “Analog Integrated Circuit Design”, John Wiley and Sons, 1997.

[82] A. Fettweis; "Realization of General Networks Functions Using the Resonant-Transfer Principle", *Proc. of the 4th Asilomar Conference on Circuits and Systems*, pp. 663-666, Pacific Grove, CA, USA, 1970.

[83] D. L. Fried; "Analog Sample-Data Filters", *IEEE Journal of Solid-State Circuits*, Vol. 7, pp. 302-304, 1972.

[84] C. Enz, F. Krummenacher, E. A. Vittoz; "An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications", *Special issue of the Analog Integrated Circuits and Signal Processing journal on Low-Voltage and Low-Power Design*, Vol. 8, pp. 83-114, 1995.

[85] Mads Haahr; "RANDOM.ORG - The true random number service". Available: http://www.random.org

[86] “Initialization Files and the Registry”, Windows NT Workstation Resource Kit, Microsoft TechNet. Available: http://en.wikipedia.org/wiki/INI_file

[87] UMC - United Microelectronics Corporation Semiconductor Foundry. Available: http://www.umc.com/english/index.asp

[88] TSMC, Taiwan Semiconductor Manufacturing Company. Available: http://www.tsmc.com/english/default.htm

[89] Korpela, Jukka; "Tab Separated Values (TSV)". Available: http://www.cs.tut.fi/~jkorpela/TSV.html

[90] R. L. Graham, G. M. Shipman, B. W. Barrett, R. H. Castain, G. Bosilca, A. Lumsdaine; “Open MPI: A High-Performance, Heterogeneous MPI”, *Proc. of the 2006 IEEE International Conference on Cluster Computing*, pp. 1-9, Barcelona, Spain, 2006.

[91] W. Sansen, Z. Y. Chang; "Feedforward compensation techniques for high-frequency CMOS amplifiers", *IEEE Journal of Solid-State Circuits*, Vol. 25, pp. 1590-1595, 1990.

[92] J. Lloyd, Hae-Seung. Lee; "A CMOS op amp with fully-differential gain-enhancement", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 41, pp. 241-243 1994.

[93] E. Sackinger, W. Guggenbuhl; "A high-swing, high-impedance MOS cascode circuit", *IEEE Journal of Solid-State Circuits*, Vol. 25, pp. 289-298, 1990.

[94] B. Razavi; “Design of Analog CMOS Integrated Circuits”, McGraw-Hill, 2000.

[95] R. Lotfi, M. Taherzadeh-Sani, M . Yaser Azizi, Omid Shoaei, "Low-power design techniques for low-voltage fast-settling operational amplifiers in switched-capacitor applications", *Integration, the VLSI journal*, Vol. 36, pp. 175-189, 2003.

[96] Hamed Aminzadeh, Khalil Mafinezhad; "On the power efficiency of cascode compensation over Miller compensation in two-stage operational amplifiers", *Proc. of the 2008 ACM/IEEE International Symposium on Low Power Electronics and Design*, pp. 283-288, Bangalore, India, 2008.

[97] P. J. Hurst, S. H. Lewis, J. P. Keane, F. Aram, K. C. Dyer; "Miller compensation using current buffers in fully differential CMOS two-stage operational amplifiers", *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 51, pp. 275-285, 2004.

[98] O. Choksi, L. R. Carley; "Analysis of switched-capacitor common-mode feedback circuit", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 50, pp. 906-917, 2003.

[99] A. D. Grasso, G. Palumbo, S. Pennisi; "Three-Stage CMOS OTA for Large Capacitive Loads With Efficient Frequency Compensation Scheme", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 53, pp. 1044-1048, 2006.

[100] J. Ramírez-Angulo, R. G. Carvajal, J. A. Galán, and A. López-Martín; "A Free But Efficient Low-Voltage Class-AB Two-Stage Operational Amplifier", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 53, pp. 568-571, 2006.

[101] J. Ramírez-Angulo, M. S. Sawant, S. Thoutam, A. J. López-Martín, and R. G. Carvajal., "New Low-Voltage Class AB/AB CMOS Opamp With Rail-to-Rail Input/Output Swing", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 53, pp. 289-293, 2006.

[102] M. Yavari, O. Shoaei; "Very Low-Voltage, Low-Power and Fast-Settling OTA for Switched-Capacitor Applications", *Proc. of the 14th International Conference on Microelectronics*, pp. 10-13, 2002.

[103] K. D. Layton, D. T. Comer, D. J. Comer; "Bulk-Driven Gain-Enhanced Fully-Differential Amplifier for VT+2Vdsat Operation", *Proc. of the 2008 IEEE International Symposium on Circuits and Systems*, pp. 77-80, Seattle, USA, 2008.

[104] L. Shang-Quan, Y. Yong-Sheng, D. Hong-Hui, S. Yu-Kun and G. Ming-Lun; "A Low Power Consumption, High Speed Op-amp for a 10-bit 100MSPS Parallel Pipeline ADC", *Proc. of the 2008 IEEE Asia Pacific Conference on Circuits and Systems*, pp. 818-821, Macao, 2008.

[105] R.S. Assaad, J. Silva-Martinez; "The Recycling Folded Cascode: A General Enhancement of the Folded Cascode Amplifier", *IEEE Journal of Solid-State Circuits*, Vol. 44, pp. 2535-2542, 2009.

[106] “Very low power, negative rail inut, rail-to-rail output, fully differential amplifier”, Texas Instruments. Available: http://focus.ti.com/docs/prod/folders/print/ths4521.html

[107] A.P. Perez, K.Y.B. Nithin, E. Bonizzoni, and F. Maloberti., "Slew-rate and gain enhancement in two stage operational amplifiers", *Proc. of the 2009 IEEE International Symposium on Circuits and Systems*, pp. 2485-2488, Taipei, Japan, 2009.

[108] M. M. Ahmadi; "A New Modeling and Optimization of Gain-Boosted Cascode Amplifier for High-Speed and Low-Voltage Applications", *IEEE Transactions on Circuits and Systems II: Expression Briefs*, Vol. 53, pp. 169-173, 2006.

[109] M. del M. Hershenson, S. P. Boyd, T. H. Lee; "Optimal design of a CMOS op-amp via geometric programming", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 20, pp. 1-21, 2001.

[110] F. You, S. H. K. Embabi, E. Sanchez-Sinencio; "A multistage amplifier topology with nested Gm-C compensation for low-voltage application", *Proc. of the 43rd IEEE International Solid-State Circuits Conference*, pp. 348-349, San Francisco, CA, USA, 1997.

[111] Matthias Köfferlein; “KLayout - High performance Layout Viewer and Editor”. Available: http://www.klayout.de/index.html