UNIVERSITÀ DI PISA

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Elettronica

Anno accademico 2002/2003

TESI DI LAUREA

Caratterizzazione dello Spazio Architetturale di un Amplificatore Transconduttivo

Relatori:

Prof. Pierangelo Terreni

Ing. Fernando De Bernardinis

Candidato:

Pierluigi Nuzzo

Ai miei nonni
Margherita e Rocco, Donata e Luigi

Ringrazio i miei relatori, il Prof. Pierangelo Terreni e l'Ing. Fernando De Bernardinis per avermi fornito l'occasione di venire a contatto con un ambito di ricerca molto attuale e stimolante, e per la costanza dimostrata nel seguirmi.

Un grazie sincero va all'Ing. Giovanni Cesura e all'Ing. Michele Fedeli, del Centro di Ricerca STMicroelectronics *di Pavia, per avermi messo a disposizione gran parte del materiale di studio necessario per questa tesi, oltre alla loro grande esperienza. Grazie, per le illuminanti discussioni avute e per l'interesse dimostrato nei confronti di questo lavoro.*

Un grazie particolare va ai miei genitori e a quanti, parenti e amici, hanno sempre creduto in me e non hanno mai smesso di spronarmi a continuare deciso sulla mia strada, qualunque cosa accadesse.

Da ultimo un grazie di cuore ai miei amici, conosciuti a Scuola e in Facoltà, per i bei ricordi, che serbo, delle esperienze che con loro ho condiviso durante tutto il periodo dei miei studi universitari.

Pierluigi

Architectural Space Characterization of a Transconductance Amplifier

Contents

Introduzione

> "In the telecommunication era, *analog design* is not just a matter of isolated self-satisfying circuit tweaking, it *is above all a system planning.*" (da M. Gustavsson, J. J. Wikner, N. N. Tan, *"CMOS Data Converters for Communications"*, Kluwer Academic Publishers)

Nell'era delle telecomunicazioni, *la progettazione analogica* non consiste solo in un isolato e fine a se stesso ritocco di un circuito, *è soprattutto pianificazione a livello di sistema.*

La continua evoluzione delle telecomunicazioni ha accelerato ulteriormente lo sviluppo delle tecnologie microelettroniche; la velocità d'elaborazione cresce insieme con le capacità di memorizzazione e l'ampia diffusione d'accessori portatili spinge verso sistemi ad alto livello d'integrazione (*System-on-a-Chip*) e con consumi minimi di potenza. I circuiti digitali sfruttano meglio i benefici delle tecnologie più avanzate e continuano a migliorare per prestazioni e consumo di potenza. Per questo oggi si ricorre sempre più spesso a circuiti digitali per svolgere le più importanti operazioni sui segnali, anche quelle che in passato erano svolte da circuiti analogici. Peraltro, i cosiddetti *"sistemi embedded"* hanno bisogno di interfacciarsi col mondo esterno per acquisire informazioni dalla realtà fisica ed interagire di conseguenza con l'ambiente. Poiché il mondo esterno è prevalentemente "analogico", in questi sistemi è immancabile una parte analogica. Essa, pur interessando una percentuale modesta dei dispositivi di tutto il circuito, costituisce il collo di bottiglia, in termini di tempo, per chi lo progetta e lo mette in commercio.

Un esempio di un circuito analogico largamente utilizzato è il convertitore analogico-digitale. I sistemi di comunicazione digitali hanno generato una grande richiesta di convertitori ad elevate prestazioni. Per essi la convenzionale modellizzazione come semplici blocchi di conversione numerica non è sufficiente, date le applicazioni. Per esempio, dal punto di vista di un

sistema di comunicazione la valutazione della distorsione e dei prodotti d'intermodulazione acquista sempre più rilevanza e si affianca alle tradizionali caratterizzazioni basate sul rapporto segnale-rumore.

La sfida offerta dalla progettazione di un tale sistema può essere efficacemente raccolta se supportata da metodologie di progettazione strutturate, intese a ridurre i tempi di progetto, aumentarne l'efficienza e trasferirne la fase di valutazione dei compromessi (*trade-offs*) al livello di sistema.

Il rischio che corre un progettista analogico è quello di dover progettare circuiti le cui specifiche sono state stabilite da un ingegnere di sistema, senza avere un'idea abbastanza definita dell'effettivo funzionamento del sistema. Al contrario, un ingegnere di sistema è portato a trascurare molti effetti del secondo ordine caratterizzanti il mondo analogico. Questa situazione generalmente porta a soluzioni fattibili ma non ottimali. Il tutto accade perché, ancora oggi, nella progettazione analogica, una *esplorazione dello spazio di progetto* è molto difficile da compiere: un progettista ha a che fare con una gran quantità d'architetture realizzabili. Per ognuna di queste, le possibili prestazioni sono controllate da molte variabili continue per mezzo di sistemi di equazioni differenziali non lineari. Pertanto è veramente arduo -ammesso che sia possibile- stimare le prestazioni da una prospettiva di sistema e valutare quantitativamente l'entità dei compromessi di progetto.

Il presente lavoro di tesi affronta il problema della progettazione analogica a livello di sistema studiando un convertitore analogico/digitale di tipo pipeline ad elevate prestazioni in tecnologia CMOS a 0.13 μm. Il progetto di partenza del presente lavoro è stato sviluppato dal centro di ricerca della *STMicroelectronics* di Pavia. Più specificamente, nella tesi, è studiato l'amplificatore interstadio del convertitore al fine di valutare l'ottimalità delle specifiche richieste nel progetto originale. Tale amplificatore è realizzato con una topologia a condensatori commutati in cui l'operazionale è un *folded cascode* a doppia uscita.

In questa tesi viene applicata una metodologia di progetto basata sull'esplorazione e caratterizzazione dello spazio architetturale d'interesse, volta alla creazione di una libreria (Piattaforma Analogica) che racchiuda sia modelli di prestazioni dell'amplificatore sia modelli comportamentali dello stesso da utilizzarsi per la progettazione ad alto livello. Una tale libreria può essere efficace se, pur nascondendo al progettista di sistema i dettagli implementativi del componente in esame, è in grado di riprodurre con un certo grado di fedeltà le non idealità insite in esso.

La tesi si articola in quattro capitoli. Nel primo capitolo vengono affronta-

te le problematiche che sorgono in seguito all'introduzione di metodologie di progetto sistematiche o di sintesi automatica ad un sistema analogico. Viene, inoltre, presentata una panoramica dei risultati raggiunti in tema di sintesi automatica di celle analogiche e, successivamente vengono esposti i concetti-base della metodologia basata sulle Piattaforme Analogiche, adoperata in questa tesi.

Nel secondo capitolo viene presentato il sistema convertitore nella sua totalità e sono messi in evidenza i compromessi che il progettista è chiamato a valutare di fronte al problema del progetto di un convertitore ad alta risoluzione (14 bit), elevato *throughput rate* ($80 \cdot 10^6$ campioni al secondo) e basso consumo di potenza. Il sistema in esame è particolarmente complesso in quanto aggiunge alla nota architettura di un convertitore pipeline l'implementazione di tecniche di calibrazione digitale e correzione delle principali componenti dell'errore interne a ciascun blocco.

Nel terzo capitolo si passa all'analisi del primo stadio del convertitore pipeline volta a ricavare le specifiche del blocco amplificatore. La metodologia prevede un campionamento dello spazio delle prestazioni (guadagno, banda, slew rate, rumore, distorsione, ...) attraverso simulazione di configurazioni generate perturbando il progetto originale. Al fine di specificare lo spazio di campionamento, vengono ricavate delle relazioni che vincolano le dimensioni dei singoli dispositivi imponendo condizioni di polarizzazione, minimo guadagno e minima banda.

Tali relazioni sono state manipolate al fine di ottenere uno schema valutativo in grado di generare configurazioni casuali del circuito che le rispettano. Un insieme di indici di prestazione viene ricavato dai dati delle simulazioni cui si ricorre dato lo scarso potere predittivo dei modelli analitici. Infatti, con le moderne tecnologie CMOS i parametri di merito sono legati alle dimensioni dei dispositivi attraverso equazioni non esprimibili in forma analitica. L'implementazione di tale schema valutativo, al fine di effettuare la caratterizzazione dello spazio architetturale, è oggetto del quarto capitolo.

La soluzione proposta nella tesi si basa su un insieme di funzioni realizzate in linguaggio MATLAB® per la generazione delle configurazioni e su uno script `Ocean` con cui si prende il controllo del simulatore di circuito (*Spectre* della Cadence®) da linea di comando. MATLAB mette a disposizione un articolato insieme di funzioni matematiche e di routine già definite, utili per maneggiare le equazioni a cui le configurazioni devono sottostare mentre Ocean permette di ricavare in maniera efficiente i valori degli indici di prestazione sfruttando dei comandi predefiniti. Tali indici vengono utilizzati per la

creazione di un modello di prestazione il cui scopo è di vincolare i parametri del modello comportamentale corrispondente, a valori effettivamente ottenibili dall'architettura prescelta. Due modelli comportamentali, uno tempo discreto e l'altro tempo continuo, sono proposti nel capitolo quarto. Sono implementati utilizzando il pacchetto *Simulink* di MATLAB e sfruttando la sua flessibilità. Il cuore dei modelli consiste in alcune Funzioni di Sistema (*S-Functions*) chiamate durante la simulazione da Simulink.

I modelli comportamentali, adeguatamente vincolati dal modello di prestazione, possono essere utilizzati al fine di selezionare, tramite ottimizzazione a livello di sistema, un insieme di specifiche ottime per l'amplificatore in esame.

Introduction

> "In the telecommunication era, *analog design* is not just a matter of isolated self-satisfying circuit tweaking, it *is above all a system planning.*" [1]

Continuous evolutions of telecommunications lead to technology development in Microelectronics; computation speed increases with data storage capabilities and the wide diffusion of portable equipment pushes towards highly integrated systems (*System-on-a-Chip*) with minimum power consumption. Digital circuits can easily exploit technology scaling and keep improving performances and power consumption. As a consequence, nowadays more and more operations are performed by means of digital circuits and when the same functionality can be implemented in both digital and analog domains, the digital circuits are generally preferred to their analog counterparts. However, embedded systems need to interface with the external world to acquire information from the physical world and consequently interact with the environment. Since the external world is mostly "analog", in these systems analog components are always present. Even if they are typically a small fraction of the overall design size, made up of some thousands of analog transistors, the analog components in these designs are often the bottleneck for designers and vendors.

An example of a widespread analog circuit is provided by analog-to-digital converters. Digital communication systems have created a great demand in high performance data converters. The conventional view of data converters as numerical conversion blocks does not suffice for this kind of communication applications. From a communication system perspective, together with the signal-to-noise ratio requirements, distortion and intermodulation are of great importance as well.

The challenge of designing this kind of systems can be effectively faced if it is adequately supported by design methodologies, so as to reduce design

efforts, increase design efficiency and perform design trade-off evaluation at the system level.

Most analog designers can only design circuits specified by system engineers without much understanding of the system, and most system engineers have not much knowledge about analog issues. This combination usually results in feasible but non-optimal system solutions. This happens because nowadays, in analog design domain, a *design space exploration* is very difficult to perform. Designers have to deal with a large number of feasible architectures. For each architecture, the range of possible performances is controlled by several continuous variables by means of systems of nonlinear differential equations. Therefore, it is very hard -if possible at all- to estimate performances from the system level and quantitatively evaluate trade-offs.

This thesis faces the problem of system level analog design by examining a CMOS, 0.13 μm, high performance analog-to-digital pipeline converter. The case study comes from the *STMicroelectronics* research center in Pavia. We focus our attention on the interstage residue amplifier of the converter because this is the most critical block as for linearity and power trade-offs. The subsystem is a switched capacitor circuit and the operational amplifier used is a fully differential folded cascode. The architecture is studied in order to furnish an amount of data, which could help evaluating the optimality of the specifications required for the amplifier, given a set of optimal system level performances.

A design methodology based on exploration and characterization of the architectural space is applied with the purpose of building a library (Analog Platform) which includes both performance models for the operational amplifier (op amp) and behavioral models to be used in high level design. Such a library may be useful if it is capable to faithfully reproduce built-in non-idealities even by hiding the implementation details to the system designer.

The thesis is organized in four chapters. In chapter 1 the problems are discussed, which originate from the introduction in the analog world of systematic design methodologies and automatic synthesis. Moreover, a panorama is given of the results reached as for automatic synthesis of analog blocks and, later, the basic concepts of Analog Platform Based design methodology are summarized.

In chapter 2 the converter is described and most of the design trade-offs are pointed out. The 14 bit, 80 Ms/s converter is quite a complex system since it adds to the well-known pipeline converter architecture the implementation

of digital calibration and correction techniques in order to minimize the errors introduced in almost each subsystem.

In chapter 3 the first stage of the converter is analysed and the specifications for the amplifier are found. According to the methodology applied in this thesis, the op amp performance space (gain, bandwidth, slew rate, noise, distortion, ...) has to be sampled through simulation of configurations obtained by means of perturbations impressed to the original design from STMicroelectronics. In order to limit the sampling space some relations are found, which impose bias conditions, minimum gain and minimum bandwidth thus constraining sizes for each transistor.

These relations have been manipulated in order to obtain an evaluation scheme capable of generating random circuit configurations that satisfy the relations. A set of performance figures is extracted from simulation data. Simulation is necessary because of the lack in accuracy of our analytical models, when trying to predict circuit performances. In fact, in deep-submicron processes the performance figures are linked to the device sizes by means of equations that have not an analytical formulation. The implementation of this evaluation scheme useful to perform the architectural space characterization is described in chapter 4.

The solution proposed here is based on a set of functions written in MATLAB® language to generate configurations and on an `Ocean` script to control the circuit simulator (*Spectre* from Cadence®) in batch mode. MATLAB offers a set of routines and mathematic functions already implemented, useful to manipulate constraint equations; Ocean allows extracting easily, through a predefined set of commands, the values of the performance figures. These data are exploited to create a performance model which constraints the behavioral model parameters to assume the values reachable for the selected architecture. Two behavioral models, the first in the discrete time domain and the second in the continuous time domain, are proposed in chapter 4. They are implemented using the *Simulink* package of MATLAB and exploiting its flexibility and expansion capabilities. The core of the models consists in some *System Functions* (S-Functions) called by the Simulink simulator.

The behavioral models constrained by the performance model can be used to select, by means of system level optimization algorithms, a set of optimal specifications for the first stage residue amplifier of the converter.

Chapter 1

Analog Circuit Synthesis and Platform Based Design

.

The background in which this thesis is inserted is that one of the research of *design methodologies for analog integrated circuits* that could provide rigorous foundations and guarantee correctness of the analog design process by a set of powerful synthesis and verification tools. In this chapter some relevant results in this field are exposed and the basic concepts of analog platform based design are introduced. In next chapters we will show how this particular design methodology can be usefully applied to a concrete case, an amplifier for an Analog-to-Digital (A/D) converter.

1.1 Introduction

The Integrated Circuit (IC) market has been traditionally dominated by digital circuits. However in the next few years a large percentage of all chips will contain some analog circuitry. Mixed-signal designs are increasing in number as a large fraction of new integrated circuits require an interface to the external, continuous-valued world. Cellular telephones, magnetic disc drives and compact disc players are just a few such examples of those technologies that most people consider hallmarks of the digital revolution, but they actually rely on a core of analog circuitry.

The digital portion of these designs can be attacked with modern cell-based tools for synthesis, mapping and physical design. The analog portion, however, is still routinely designed by hand. Although it is typically a small

fraction of the overall design size, made up of some thousands of analog transistors, the analog partition in these designs is often the bottleneck because of the lack of automation tools [2]. To give an idea, the analog section of a modern Application Specific Integrated Circuit (ASIC) may consume 90 % of the overall design time, while consuming only 10 % of the ASIC's die area.

Abstraction levels, design methodologies already consolidated and hardware/software partitioning in the digital arena allow also the integration and reuse of pre-designed blocks and customization of systems by means of software. Not so in the analog world, which relies for basic functionality on tight-control of low-level device and circuit properties that vary from technology to technology. Analog design intrinsically deals with continuous time and continuous valued waveforms dominated by partial differential equations and design in the upper level also depend on a number of parasitic effects related to bottom-level device and circuit properties. The situation appears to be worsening as we head into the era of System-on-Chip (SoC) designs. To manage complexity and time-to-market, SoC designs require a high level of reuse, and cell-based techniques lend themselves well to a variety of strategies for capturing and reusing digital intellectual property (IP). However, the analog portions of these circuits are still designed by hand today. They are even routinely ported by hand as a given IC migrates from one fabrication process to another.

For these causes analog design has been traditionally the most difficult discipline of IC design. This difficulty stems from the effects that physical implementations have on the functionality of analog circuits. Over the years, a number of attempts have been carried out to ease analog design, but the practical results are still not satisfying in terms of productivity, design quality and time-to-market. Much worse, the analog and digital sections are defined very early in the design process, using feasibility rather than any optimality criterion to decouple analog design from the rest of the system [3].

From the above discussion, it is clear how the design of next generation integrated systems requires that "Art", a mix of knowledge, experience, intuition and creativeness, be supported by "Science", i.e. a structured methodology that theoretically limits the space of exploration and yet still achieves superior results in the fixed time constraints of the design. This is a necessary condition to create an economically feasible electronic design flow [4].

At first, the knowledge of the tradeoffs between the different competing circuit performances at the design space boundaries would be of considerable use to circuit designers. The designer then has all the information available

to choose which performance to promote at the cost of suboptimal values for the other performances. Secondly, to formulate a methodology is useful not only at the circuit level but more favorably at the system level in order to promote a *top down design flow*. Top down methodologies to analog design has been proposed more than ten years ago. It has been shown how this approach could be effective but the lack of general supporting tools has made its application quite cumbersome in real cases.

The methodology is made up of the following phases [5](see Fig. 1.1):

System-level Design—Starting from the system specifications, the system designer first needs to compose a system architecture consisting of a number of building blocks. In a second step, specifications for these blocks need to be assigned. In many cases performances of consecutive blocks can be traded off: for instance in a cascade of amplifier stages how can be partitioned the total gain? If a large amount of gain is assigned to the first stage, later stage can be sized easily regarding noise, but linearity violation could occur. If the designer could know the boundaries of the performance space for a given block, independently from the architecture, he would easily verify whether the system topology is feasible for the given specifications. Moreover, when assigning upper or lower bounds on functional performances of a specific building block one needs to understand the implications. Is this set of specifications located within the design space boundaries of the selected topology? How much power will be dissipated?

Topology Selection— In the next step different topologies are examined for each functional block. For a given block, the topology with the lowest implementation effort and capable of respecting the performance constraints is chosen. A library should be available of various different topologies adequately characterized so that performances attainable from each of them are known and can be compared.

Circuit Design—Now the circuit designer's task is to find the values of the selected topology's design variables such that all the specifications of the block are met. In most case the set of specifications does not uniquely determine the solution, some degree of freedom are left for optimization. Some kinds of trade-off curves should be derived: they are vital info for the designer to size the circuit properly, so as to achieve optimal performance.

The issue with all this procedure comes from the complexity of estimating analog performances and costs for partially specified blocks. The interactions among different components in analog systems are so complex that it is virtually impossible to evaluate performances and trade-offs without generating

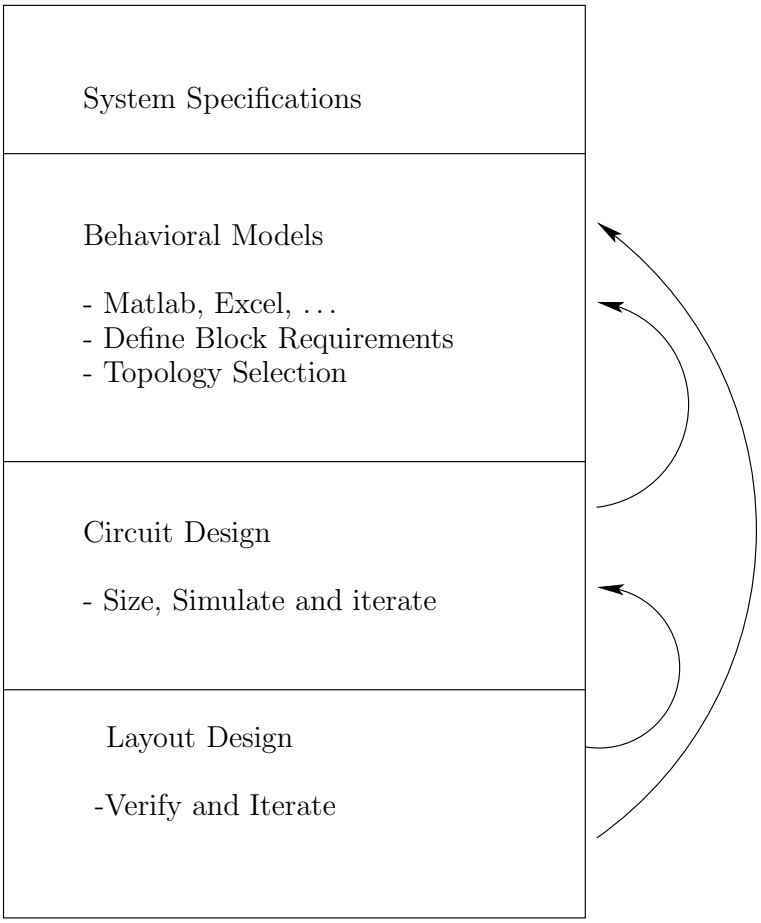


Figure 1.1: Traditional analog design flow—In the system level design phase, environments like Matlab and Excel are used to define block requirements. The arrows symbolize the frequent iterations in the design phase.

and simulating a complete cell. This fact severely limits the generality and the accuracy of the approach when proceeding from high-level specifications down to silicon implementation with successive optimizations and constraints propagation.

1.2 Relevant Results in the CAD Research Field

Let us make some terminological considerations in order to better understand what we are going to expose. For *analog circuit synthesis* we mean translating performance specifications into a circuit schematic with sized devices, thereby automating part of the analog design process. *Circuit optimization* also gives a circuit schematic with sized devices as a result but it requires a good initial starting point to find an excellent final answer. Synthesis requires no special starting point information. When we say a "good" initial starting point we mean that the optimization may converge to a local minimum significantly worse than the circuit's best capability; as a consequence it must be biased through a starting point not so far from that performance space region we are interested in. The ability to synthesize complete circuits rapidly, as demonstrated in the previous section, also brings with it the ability to explore design trade-offs rapidly. Design methodologies are largely exploited for digital projects. However analog and digital design problems are very different. In order to understand the problems encountered when trying to extend these procedures to the analog world it is advisable to point out the differences between analog and digital design problems. These differences can be partitioned into four categories: size, hierarchy, process and performance constraints [6].

1. As for the *size*, analog circuits tend to have fewer transistors than digital circuits, but these individual ones are more difficult to design. Analog circuits often exploit the full spectrum of capabilities inherent in the physics of individual devices, so a few transistors often suffice to perform complex functions.

2. Analog and digital circuits each employ *hierarchy*, but in substantial different ways. In digital circuits, there is tacit agreement on the abstraction levels through which a design must pass, and these abstractions play a central role in the organization of synthesis. The use of

hierarchy usually helps translate downward, level to level, the design constraints. There is not such a well-developed hierarchy for analog circuits, but hierarchy is important nevertheless. Each designer conceives a complex analog cell as a composition of more elementary blocks (e.g. current mirrors, active loads, differential pairs, bias chains ...), which at first he sizes singularly. What is difficult is the propagation of those constraints from one level to another, as we already pointed out in section 1.1. This happens because the expressions at our disposal are often complicated to manage and never so accurate. Besides they do not take into account the interactions among the various blocks so that if you find the optimum sizing for each block, the overall design will not generally coincides with the global optimum. What's more two blocks connected together surely will not behave as expected when they are sized singularly. For instance once you have sized an amplifier for a given low-frequency voltage gain, its gain will reduce to one half of the expected value if it is cascaded with a load whose impedance is comparable with the amplifier output load. This trivial example allows us to touch by hand how communication issues are important in a hierarchical design flow. In practise they are solved through cycles of simulations on the entire system so that the initial dimensions can be adequately fixed. An expert analog designer knows at what level in the hierarchy these effects must be considered. He is able to figure in his mind the global system both hierarchical and flat. This is a skill acquired with experience, a sort of "knowledge" that is very difficult to encapsulate into an automatic synthesis tool.

3. As for the *fabrication process*, at the higher levels of digital synthesis, process constraints appear in highly simplified forms (e.g. attainable clock frequency or drive capabilities). In analog circuit synthesis, such process constraints appear in far greater detail, even during high level design (e.g. matching).

4. *Performance constraints* on the behavior of analog circuits also differ radically from those of digital circuits. Digital circuits are often specified using a behavioral language which can capture the dataflow for digital quantities moving through functional blocks and storage elements. For the common analog circuits, the qualitative behavior is often known implicitly (e.g. an A/D converter digitizes continuous signals) and the specifications may take the form of a set of performance parameters that must be achieved, such as gain, bandwidth, input noise, or phase

margin.

Practical analog synthesis tools must deal with all these concerns and they must handle performance parameters that constrain continuous quantities, such as voltage and current. These parameters depend intimately on the fabrication process, and depend on the careful design of several mutually interacting devices at potentially different levels of the analog hierarchy.

As for the design methodology for sizing analog and RF integrated circuits, a transition has emerged from a "pencil and paper" approach, through a simulation-based methodology towards an optimization based approach [5]. In the early years, analog circuit designers derived design equations based on their experience, and solved these equations for transistor sizes to meet the given set of specifications. Numerical simulation techniques were then applied to verify the sized circuit behavior. Later on, the circuit simulator became a crucial tool in the analog design flow, as the device model had got more elaborate and specifications had become harder to evaluate (e.g. mismatch, intermodulation, second order and third order distortion coefficients in characteristic). Starting from an initial design, the designer interactively modifies the design variables using circuit simulation as a performance evaluation tool. This methodology is currently applied nowadays.

In a later phase, and this is the perspective for the future, the design of certain types of block is automated to meet stringent time-to-market request. As a consequence, in the area of analog design automation, optimization-based sizing techniques and tools have gained acceptance over the last few years [6, 7, 2]. Most of them operate on fixed topology/architecture and attempt to address the sizing problem with respect of a given set of objective performances. All these tools are composed by an *evaluation engine* and an *optimization engine* (Fig. 1.2).

For a given sets of design variables, the evaluation engine calculates a set of circuit performances and then combines these into an overall cost function of the implementation. The optimization algorithm is steered by this implementation cost to generate a new set of design variables. A successful sizing session through optimization finishes when the optimizer finds the set of design variables for which the design satisfies all performance constraints and the implementation cost is minimal. The existing sizing techniques differ from each other in two respects: the selection of the evaluation tool and the choice of optimization algorithms.

As for the selection of the evaluation tool, two main approaches are followed. The first one which dates back to the mid-eighties, relies on circuit

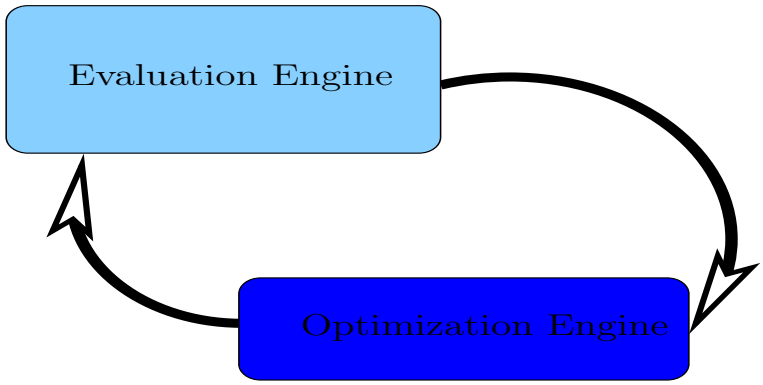


Figure 1.2: General representation of a synthesis tool. An optimazition engine interacts continuously with an evaluation engine.

simulation to evaluate the performances of a given circuit instance. The optimization problem, however, is extremely complex to solve and usually presents multiple local minima. Furthermore, performance evaluation is very expensive and poses practical limits on the design space exploration. Some industrial-size case studies have been reported but the feasibility of the approach has still to be confirmed for general applications areas and for RF applications. The second approach relies on equation based models for the circuit to be sized. Recently, several results have been obtained exploiting geometric programming to cast and solve the design problem [8]. Industrial efforts are being carried out in this direction meant to get IP generator tools. However, none of the existing tools allows performing efficient system level design.

1.2.1 Metrics for Analog Synthesis Tools

In this section we discuss some criteria in order to evaluate performances for an analog synthesis tool. In [7] five critical metrics have been listed for analog synthesis tools: accuracy, generality, complexity, synthesis time and preparatory effort. We propose to introduce another factor to verify the effectiveness of a system level design methodology embodied into a tool, i.e. the *abstraction level*. We define:

1. **Accuracy:** the discrepancy between the synthesis tool's internal performance prediction mechanisms and those of a detailed circuit simulator that uses realistic device models;

2. **Generality:** the breadth of the circuits and performance specifications that can be successfully handled by the synthesis tool;

3. **Complexity:** the largest circuit synthesis task that can be successfully completed by the synthesis tool;

4. **Synthesis time:** the CPU time required by the synthesis tool;

5. **Preparatory effort:** the designer-time/effort required to render a new circuit design in a form suitable to the tool to complete;

6. **Abstraction level:** the capability of providing models of the cells, which can be used at a system level to evaluate the trade-offs.

Of course, an ideal tool maximizes accuracy, generality, complexity and abstraction level, while minimizing synthesis time and preparatory effort.

Another problem is the role in the tool of process corners and circuit working conditions. The solution proposed in this thesis can provide design space exploration in the nominal case. It is also possible to deal with matching and process problems in the future by finding some adequate models for these phenomena. For now we are sure that a methodology that supports the designer even for an initial solution is already a great contribute in order to reduce design time and effort.

In next section we shall expose the analog platform based design methodology that will be illustrated in this thesis and applied to a concrete case. At the end we will discuss the advantages of this methodology with respect to the above metrics.

1.3 Analog Platform Based Design Methodology

Analog Platform based Design Methodology, proposed in [4], permits system level exploration through optimization on *behavioral models* of a given circuit, adequately constrained by *performance models*. This is a novel approach to system level analog design [3]. *Platform* is a new abstraction level introduced to separate circuit design from design space exploration. An *Analog Platform* (AP) encapsulates analog components concurrently modelling their behavior and their achievable performances. Therefore, once the designer gets the AP

for a given functional block, he can perform system level exploration guaranteeing implementability of analog components. Performance models are obtained through statistical sampling of circuit configurations in order to estimate the *evaluation function*. This function gives the circuit performances starting from a certain set of configurations of initial parameters of the circuit. In this context the design configuration space is specified with *Analog Constraints Graphs* (ACGs) so that the sampling space is significantly reduced. Actual approximation of performance models is achieved through *statistical learning theory* while system level optimization can be achieved through *simulated annealing* based algorithms.

1.3.1 The Concept of Platform

The concept of platform is so general that can be applied to design, not only in electronics, both analog and digital, but also to communication network design [4]. A platform can be defined as a set of architectural resources (hardware and software or communication), that allows mapping and implementation of system functionalities. The greatest benefit that platforms introduce is the improved design efficiency that is gained by restricting the design space to a family of configurable architectures and by providing accurate and efficient estimation mechanisms and mapping tools. Platforms provide a library of architectural resources that has been characterized *bottom-up* This library provides ad hoc estimation and mapping methods to support efficient *top-down* flows. Therefore, Platforms allow an effective *meet-in-the-middle* design approach (Fig. 1.3).

An *Analog Platform* is a pre-characterized library of components that can be used to implement analog functionalities. The main purpose of introducing APs is to generate a *new abstraction level* in analog design, so that an effective decoupling is achieved between system level (analog) design and circuit design and synthesis. In the analog framework APs can enable quantitative exploration of the design space at the system level. As we have seen, in analog design no systematic approach to design space exploration is adopted and the specifications of system components are set by conservativeness and feasibility arguments rather than global optimality criteria. APs while theoretically limiting the design space, in real cases allow very effective explorations to be performed. However module generators approaches may work for CAD or IP vendors, but it is not likely to become a general way to approach analog design. In fact, no methodology is likely to be adopted in the analog world if it limits the creativity of the designers, which uniquely

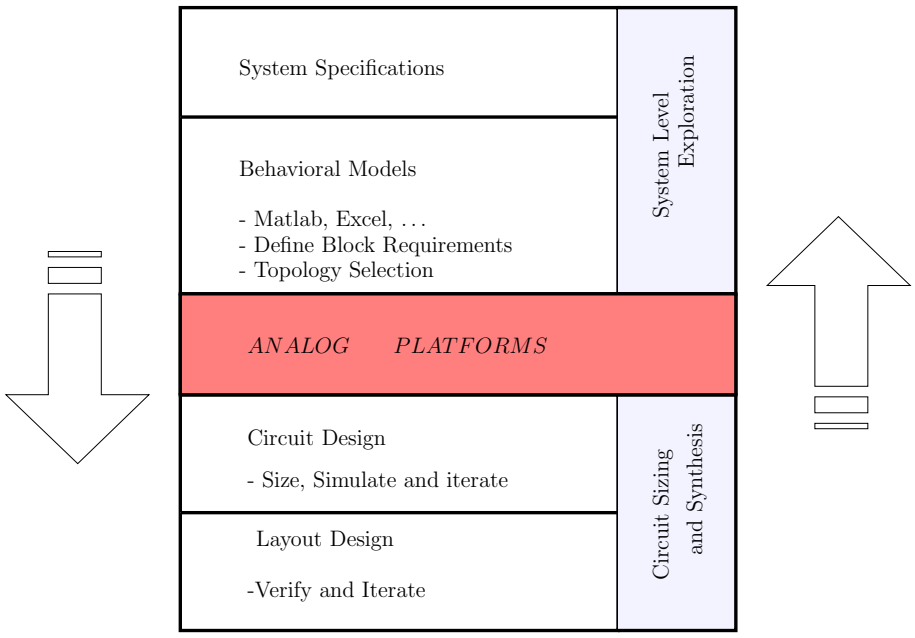


Figure 1.3: Introduction of APs into the traditional analog design flow—APs are another abstraction level in the design flow.

allow creating new architectures or reaching performances out of each technology generation. APs consist of hierarchies of *behavioral* models for the platform considered, and of matching *performance* models, as described in the following subsection.

1.3.2 Behavioral Models and Performance Models

A behavioral model $\mu(in, out, \zeta)$ is a parameterized executable model that introduce at the functional level a number of non-idealities due to the actual circuit implementation. ζ is a vector of parameters controlling the actual behavior of the model (e.g. specific gain and noise values) which practically coincides with the set of performance figures in which we are interested and that encapsulate the component from our viewpoint. In order to hide the details of the implementation, non-idealities are modelled in terms of the effects they introduce (e.g. distorsion, gain error,...) rather than in term of the causes (e.g. transistor behavior, topology property,...). Depending on the level of abstraction and the modelled effects, behavioral models may be expressed in the time or in the frequency domain, using continuous time

or discrete time blocks. They can be implemented in hardware description languages, such as Verilog AMS o VHDL AMS, or more generic languages, such as MATLAB/Simulink. Note that a behavioral model is capable to model only some specific effects of a particular system, that ones we need given the particular application. They cannot be totally self-contained models, since in this case simulation time would increase and there would not be any improvement in using these models rather than the complete transistor-level one. Anyway they remain *functional* models that do not contain any information about the actual architecture with which that functionality is implemented. An example of a behavioral model is shown in Fig. 1.4 for a generic operational amplifier.

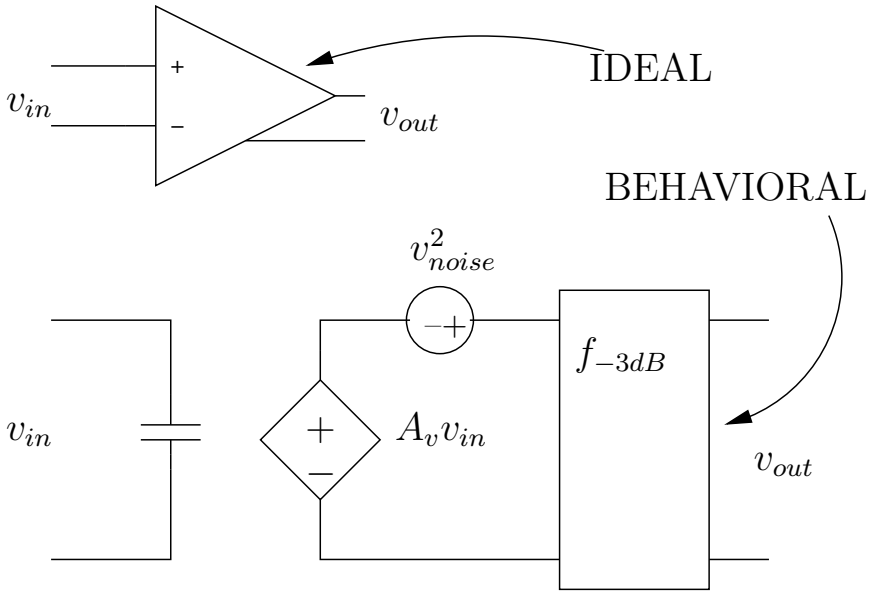


Figure 1.4: Example of a behavioral model for an amplifier

A performance model constrains behavioral models to behave consistently with the implementation architecture, introducing relations among behavioral model parameters that have to hold for the behavior to be consistent. The performance parameters are measured from simulation data.

Given a certain circuit topology $\mathcal{C}$, we consider the set of parameters κ (e.g. an array of m parameters) controlling circuit configurations. We say that κ belongs to the *Input space* $\mathcal{I}_\mathcal{C}$ of the *performance model* $\mathcal{P}$ for that implementation. We next consider a set performance figures ζ (suppose an array of n elements) completely characterizing its behavioral model. We state

that ζ belongs to the *Output space* $\mathcal{O}_\mathcal{C}$ of the *performance model*. $\mathcal{I}_\mathcal{C}$ is the set of vectors κ over which we want to characterize $\mathcal{C}$, $\mathcal{O}_\mathcal{C}$ is the set of vectors ζ that are achievable by $\mathcal{C}$. The ideal situation is that one in which a designer perfectly knows that *evaluation function* $\phi_\mathcal{C} : \mathcal{I}_\mathcal{C} \rightarrow \mathcal{O}_\mathcal{C}$, mapping a parameter set into a performance set, which is obviously impossible to occur since the number of variables and the complexity of the relations makes it untreatable using simple analytical models. To solve these problems we can only rely on designers' experience and cleverness and on simulation CAD tools. Only a deep understanding into working principles of electronic devices, circuit theory and simulation tools together with trial and error based design techniques are almost uniquely the designers' instruments. In this context we define the performance relation of $\mathcal{C}$, giving its input and output spaces and an evaluation function, to be $\mathcal{P}_\mathcal{C}$ on $\mathbb{R}^n$ that hold only for points $o \in \mathcal{O}_\mathcal{C}$. Sometimes also the performance relation characteristic function is denoted with $\mathcal{P}_\mathcal{C}$ so that we can say that given a behavioral model μ , a performance model $\mathcal{P}$ constrains μ to the values of ζ for which $\mathcal{P}_\mathcal{C}(\zeta) = 1$, i.e. for that value of ζ that can be achieved by that architecture. The situation described here is represented in Fig. 1.5

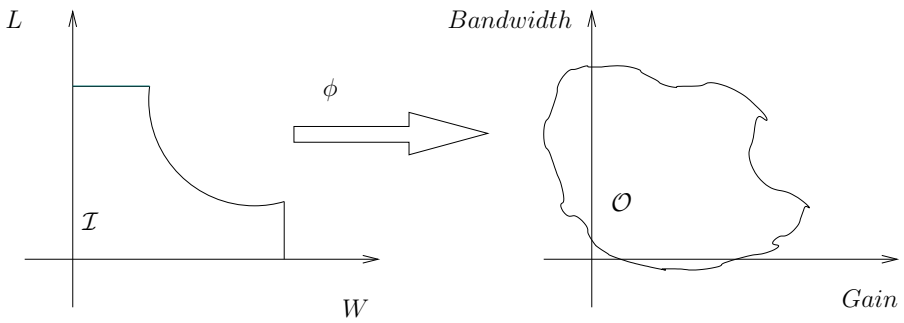


Figure 1.5: A point in the input parameter space $\mathcal{I}$ is mapped into a point in the output performance space $\mathcal{O}$. The definition of the performance relation $\mathcal{P}$ is also shown.

Once a representation of $\mathcal{P}_\mathcal{C}$ has been found (see section 1.3.4), the architecture parameters are hidden and design flow is allowed to proceed top-down because now different architectures can be compared basing on the effects they introduce.

1.3.3 Analog Constraint Graphs

Each variable of the parameter space $\mathcal{I}$ of a circuit is given a range of minimum and maximum values it can assume. For instance the various Ws and Ls are limited by the minimum feature size as well as by constraints on the occupied area. Usually the parameter space of the circuit is much smaller than its bounding hypercube; in fact there are also a quantity of relations among the various κ_is that come from basic circuit operating conditions and that are necessary for circuit to operate properly (i.e. to be biased correctly, to have a gain larger than a certain value,...). This means that each architecture imposes to various transistors some equality and inequality constraints we have to consider while building a platform. In chapter 3 some of these constraints will be given for a particular architecture of amplifiers. The problem is always underconstrained since the equations are less than the variables and there are many degrees of freedom in the hands of the designer.

Once these constraints are set the problem of representing the performance space can be faced in this manner:

1. Random configurations κs are generated, constrained by equations and inequalities we cited above;

2. These configurations are simulated to get the performance vectors ζs.

The exploitation of the constraints is a critical step as we shall discuss in section 4.6.

In this section a graphical method is described based on Analog Constraint Graphs (ACGs) useful to easily capture constraints present in an analog design. Simplification stems from the fact that circuit configuration to be simulated are not completely chosen in a random manner, but have to respect determined constraints. This reduces the input configuration space and the number of samples of the performance relation $\mathcal{P}$ we get, by excluding those solutions that give surely non feasible circuits or circuits that produce performances we are not interested in. The Analog Constraints Graph (ACG) is a practical representation mean to derive the constraints underlying a project. An ACG is no more than an ideal map which a designer can exploit to start every exploration process since it contributes to confine the architectural space that must be characterized. Actually all the constraints included in a graph are also in the mind of the designer; but, this is an important element to point out, the process of creating a graph is quite different with respect to the mental design process; this happens

because a graph offers a static and absolutely general vision of the circuit complexity, while a designer moves dynamically along the constraints he has in mind taking some short cuts, i.e. fixing some parameters and consequently determining the others. In this way he is compelled to make more than one walk along this virtual graph using heuristic observations to find a design compliant with specifications. As a consequence, if a graph is a static map of the available ways, if we were able to exploit an algorithm which quickly and automatically makes, for many times, even if in a silly manner, what the designer more consciously usually does, we could try to sample the evaluation function ϕ thus guiding circuit exploration. This operation carried out by this hypothetical algorithm can be defined *graph scheduling*.

Definitions

An AGC is a bipartite undirected graph (Ξ, Ψ, Υ) where Ξ is the set of design variables, Ψ is the set of constraints on ξs and $\Upsilon \subseteq \Xi \times \Psi \cup \Psi \times \Xi$ is the set of edges that link design variable ξ_i to constraint ψ_j. The design variables in Ξ include circuit configuration parameters we can call *primary variables* and a set of secondary variables that make the graph easier to represent but are not required to specify a configuration (*ancillary variables*). Clearly, ancillary variables can be expressed as a function of primary variables, but the resulting constraints ψ_i would be much more complex to handle. Since ψs are usually analytic approximations of the underlying circuit relations, some attention has to be paid on the *conservativeness* of constraints. A set of constraints is conservative when each configuration that does not respect them corresponds to an incorrect circuit, even though some incorrect circuit may be generated also for some configurations that respect the constraints. This is an important element to consider given the inaccuracy of equations expressing the performance constraints: the risk can occur of rejecting good candidates. An fragment of graph is shown in Fig. 1.6. The constraint is the current equation for a MOS device following the well-known squared-law model.

Sampling ACGs

A node ξ in an analog constraints graph is said *marked* if a value or a random variable and consequent probability density function (pdf) is assigned to it. Given an analog constraint graph and a set of marked nodes, the problem of generating κ vectors can be seen as a graph *scheduling problem* that provides

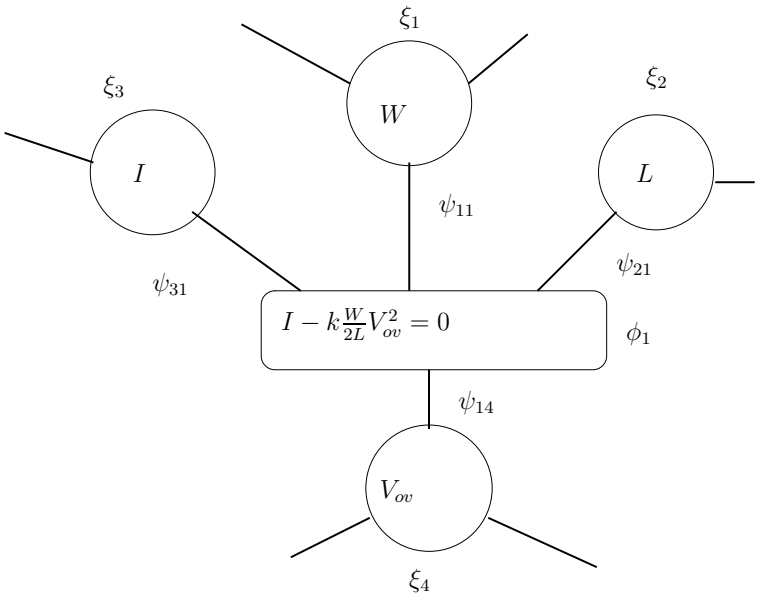


Figure 1.6: A fragment of graph showing some of the variables ξs, one of the constraints ϕs and some of the edges ψs.

a marking for *all* the primary variables. Usually initial markings consist of pdfs so that the overall graph can be seen as a graphical model for the pdf of κs. By choosing different variables or marking different node sets as initial marking, the overall pdf of κs can be altered thus biasing sampling in different ways. A (unmarked) node ξ_i is *markable* if there exists a constraint ψ_j and an edge v_{ij} such that all nodes ξ_m for which there exists an edge v_{mj} are marked, that is if ξ_i is the only unmarked node for the relation ψ_j. This condition allows to uniquely determine a marking as long as the constraint is invertible in ξ_i. If the constraint ψ_j is an inequality, there are two possibilities:

- a random slack variable may be inserted and an equality equation is then used to derive ξ_i;
- a random configuration may be found on the basis of the equality constraints, that are fewer than the variables to be marked, and if for this configuration the inequality also holds then we accept that configuration; otherwise we reject it and generate another one.

In our case the second method has been adopted as explained in section 4.7. Since in general the scheduling problem is underconstrained [1], the generation

[1] the equations are fewer than the variables

of the configurations is not a deterministic process; there are many solutions (sometimes infinite) and biasing become difficult. On the other hand when there is a number of constraints equal to the number of nodes or slightly inferior, since the starting marked nodes are generally chosen randomly it is possible that one constraint at a certain point is not respected (this is a scheduling failure) and the generation must restart. This can cause inefficiency as for the duration of the characterization process.

1.3.4 Performance Model Generation

The performance relation practically coincides with the region of the performance space reachable with a given architecture and with a given set of input variables we have decided to change. The problem of finding a representation for this region can be cast as an optimization problem. Unfortunately, very little is known a priori about $\phi_{\mathcal{C}}$; circuits are nonlinear systems so complex that it is difficult to derive by strong property concerning the output space or the performance relation. As a consequence, statistical sampling of $\mathcal{I}$ is adopted to generate a set of performance vectors ζs and statistical learning theory is exploited to generate an approximation to $\mathcal{P}_{\mathcal{C}}$ [9]. The bottom up characterization process is potentially expansive in terms of number of simulation required. However using some expedients, such as to limit $\mathcal{I}$, this cost can be adequately reduced.

1.3.5 Design Flow

In this section we describe the design flow whith the new methodology when trying to design a new analog circuit block. The rest of the thesis is an example of the procedure described here [3].

Exploration Phase

The platform paradigm can be effectively used to drive exploration and eventually optimization at a system level. We can define now *exploration* as the process of defining the set of optimal specifications compatible with the considered implementation architecture and the available performance models. The design process starts with the collection and generation of a suitable platform library for system implementation. For example, schematics coming from previous designs, module generators, analog IPs and eventually new

solutions can be included in the platform library. A number of test vectors have to be defined in order to get system performance under meaningful conditions. A behavioral model of the system is then built, and exploration is performed having platform performance models as constraints to enforce feasibility. Since no mathematical properties are generally available for the optimization problem, global optimization schemes are potentially very expansive, the number of optimization parameters available at the system level is usually low (a few tens) and behavioral simulation are expected to be very efficient.

Architecture Selection

Different implementation architectures may actually be *merged* at proper levels of abstraction since they are only view by the designer as a set of feasible functionalities. By enabling this level of abstraction, platform based analog design allows to explore a wide architectural space (depending on the platform library) while decomposing system specifications into block requirements. As for the selection of platform performances ζ, two facts may take place:

- the selected ζ is achievable by only one architecture: then the architecture has been selected;
- the selected performances are achievable by more than one architecture (which indicates that at the current abstraction level and with the current metrics the selected platforms are equivalent).

In the latter case, going to more detailed models or refining the metrics selects a proper architecture. Also, platforms may be locally refined to expand the design space or to explore interesting design corners with improved accuracy.

Actual Sizing

The result of the exploration phase and architecture selection is a set of specifications for each platform in the system. The exploration process can then be iterated over single subsystems. Eventually, the platform evaluation function $\phi_{\mathcal{C}}$ can be inverted so that specifications get converted in suitable configuration parameters. In practice, this task can be accomplished with some regression scheme on the performance data used to approximate $\mathcal{P}$. Once found the configurations whose performances are closest to the requested

specifications (nearest neighbors), this set of potential solutions could then be used as initial points of a local detailed optimization process, such as finalizing circuit sizing and layout.

1.4 Conclusions

In this chapter a brief introduction has been given to researches about design methodologies and some considerations have been exposed about the necessity of a structured methodology in support of analog design. The analog platform based design strategy has been described in particular. The rest of the thesis is but an application of this methodology to an industrial case study, the interstage residue amplifier of a pipeline analog to digital converter.

Chapter 2

The Pipelined Analog to Digital Converter

The converter we are going to describe is an $STMicroelectronics^{®}$ 80 MS/s, 14 bit ADC, digitally calibrated, in a 0.13 μm CMOS technology. We shall see how most of the power is dissipated into the interstage op amps, which become the most critical blocks of the converter to be designed. Our intention is to build an AP encapsulating the op amps with all the implication seen in chapter 1. This will permit us to explore trade-offs between linearity, noise and power. The following sections will give an overview of the most relevant characteristic of the converter system. However, it is advisable to spend first some words in order to precise the meaning of those terms we shall frequently use in the future.

2.1 Principles of Analog to Digital Conversion

An analog-to-digital converter (ADC or A/D converter) viewed as a black box ([1]) takes an analog input signal, a voltage in our case, and converts it to a digital output signal. The digital signal is a binary coded representation of the analog signal using n bits. The leftmost bit of the digital word is usually called the most significant bit (MSB) and the rightmost bit is called the least significant bit (LSB). The size of the LSB compared to the total code range is referred as the *resolution* of the converter. Hence, the resolution of a 14 bit converter is $1/2^{14}$ but, for simplicity, we refer to the resolution as being

14 bits. The transfer function of the data converter is a staircase function with saturation for large input values: therefore an input signal should never be in the saturated range since this gives rise to large conversion errors. The step size Δ is equal to the analog value of the LSB and the conversion range without causing saturation is referred as the Full Scale (FS) range of the converter. Hence the step size is

$$\Delta = \frac{FS}{2^n}. \tag{2.1}$$

Consequently we can affirm, as a general idea that an ADC is a system able to perform two distinct functions: sampling and then digitalizing an analog signal, so that the signal itself assumes discretized values during discretized time steps. To digitalize a signal means to quantize it and then to encode the samples obtained, according to a particular encoding. It is useful to introduce also some other quantities, such as the *analog transition levels*, i.e. the analog values where the output signal changes between two digital codes, and the *number of output levels* Q. A uniform quantizer is *midtreader* if Q is an odd number; otherwise it is said *midriser*. In Fig. 2.1 the two transfer characteristics are shown and compared for a 3-bit quantizer, in order to give a better idea about these definitions. The input-output characteristics have been drawn under the assumption that the analog signal range is [-V_{IN}, +V_{IN}] where in this case $V_{IN} = V_{REF} = FS/2$. In the first example (Fig. 2.1-*a*) the number of transition levels, apart from -FS/2 and FS/2, is 2^n and $Q = 2^n + 1$, while in the second case (Fig. 2.1-*b*) we have $2^n - 1$ transition levels and $Q = 2^n$. In both case we get $\Delta = \frac{FS}{Q-1}$ and $n \approx \log_2 Q$.

It is also evident that, since there are only a limited number of analog amplitude levels, there is an inherent conversion error, the *quantization error*. The quantization noise is a fundamental limitation in data converters that represents a lower limit on how small the error power[1] can be. Due to circuit imperfections the total error will always be larger in an actual implementation. We know that that the amplitude of the error is $\pm\Delta/2$, except the case of input in the saturated region. We define the ADC *input no-overload range* as the range of input values for which the quantization error magnitude never exceeds half of the step size. This is also known as the input dynamic range of the converter. It is also known that it is convenient to use the full input dynamic range of an ADC in order to minimize the effect of the quantization noise.

[1]We talk about error power and quantization *noise* since we consider the error as a random signal when the input signal is not known.

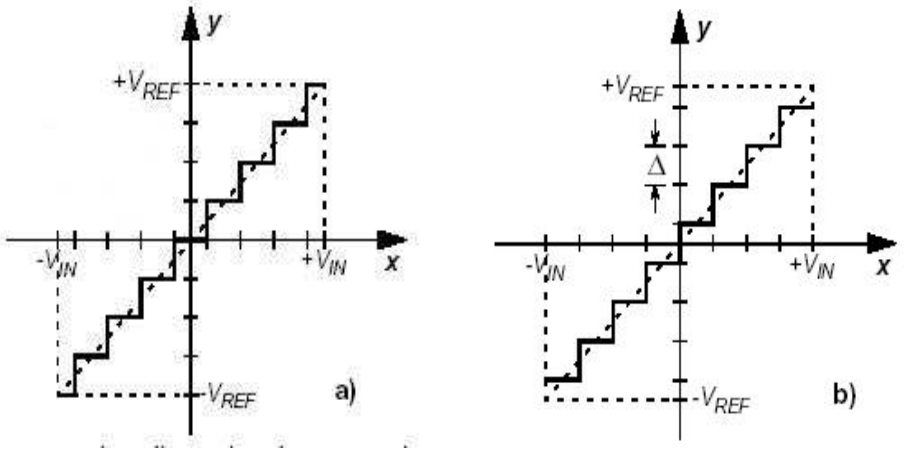


Figure 2.1: Transfer characteristic for a midtreader quantizer (a) and a midriser quantizer (b)—The step size Δ and the transition levels are also clearly represented. The figure is taken from [10].

To evaluate the performance of a data converter it is convenient to introduce a number of performance measures. The performance measures can be divided into two groups, static and dynamic measures. The differential nonlinearity (DNL) and integral nonlinearity (INL) are often used as static performance measures. The signal-to-noise ratio (SNR), total harmonic distortion (THD), spurious-free dynamic range (SFDR) and signal-to-noise and distortion ratio (SINAD) are commonly used as dynamic performance measures.

2.1.1 DNL, INL and Monotonicity

Due to non-ideal circuit elements in the actual implementation of ADCs the code transitions point in the characteristic will be moved. The step size in the non-ideal data converter deviates from the ideal size Δ and this error is called the differential nonlinearity error. We can define DNL as the deviation from one least significant bit of the range of input voltages that give the same output code ([11]). We define Integral nonlinearity (INL) as the deviation of the transfer curve from ideality.

The converter is *monotonic* if its analog amplitude level increases with increasing digital code. Non-monotonicity results in missing output codes that never appear for any analog input signal. If we number the digital

codes by an index $k = 0, \ldots, 2^n - 1$, so that $k = 0$ corresponds to the code representing the smallest analog value and $k = 2^n - 1$ corresponds to the largest analog value, then monotonicity is guaranteed if the deviation from the best-fit straight line is less than half a LSB, i.e.:

$$|INL_k| \leq \frac{1}{2} LSB \qquad \text{for all } k \tag{2.2}$$

where INL_k is the INL computed for each level. This implies that the DNL errors are less than one LSB, i.e.

$$|DNL_k| \leq LSB \qquad \text{for all } k. \tag{2.3}$$

It should be noted that the above relations are sufficient to guarantee monotonicity, but they are not necessary. In addition to the static errors caused by mismatch in the data converter components, several other error sources will appear when the input signal changes rapidly. These dynamic errors are often signal and frequency dependent and increase with signal amplitude and frequency.

2.1.2 Frequency Domain Measures

Consider a typical FFT spectrum of a non-ideal ADC when the input signal is single-tone sinusoidal. The input signal appears as the fundamental in the FFT spectrum and the quantization error generates a white noise floor. The nonlinearities in the ADC cause harmonic tones to appear above the noise floor where some of the harmonics may be folded from higher frequencies due to the sampling process. We remind that the signal-to-noise ratio (SNR) of an ideal ADC with a sinusoidal input signal is expressed in decibels as [1]:

$$SNR = 6.02 \cdot n + 1.76 \quad \text{dB}, \tag{2.4}$$

, which only holds if the input is a full scale sine wave. The SNR of an actual ADC including non ideal effects can be determined by measuring the output signal[2] and computing the ratio of the power of the fundamental and the total noise power within a certain frequency band, excluding the harmonic components, i.e.

$$SNR = 10 \log \left(\frac{\text{Signal Power}}{\text{Total Noisefloor Power}} \right). \tag{2.5}$$

[2]Measures are usually determined by using a single-tone sinusoidal input signal, but sometimes dual-tone or multi-tone measurements are more informative.

The spurious free dynamic range is the ratio of the power of the signal and the power of the largest spurious within a certain frequency band. SFDR is usually expressed in dBc (dB-carrier) as

$$SFDR = 10 \log \left(\frac{X_1^2}{X_S^2} \right) \tag{2.6}$$

where X_1 is the rms value of the fundamental and X_S the rms value of the largest spurious.

The total harmonic distortion (THD) is the ratio of the total harmonic distortion power and the power of the fundamental in a certain frequency band, i.e.

$$THD = 10 \log \left(\frac{\sum_{k=2}^{\infty} X_k^2}{X_1^2} \right) \tag{2.7}$$

where X_k is the rms value of the k-th harmonic component. Since there is an infinite number of harmonics the THD is usually calculated using the first 10-20 harmonics or until the harmonics can not be distinguished from the noise floor.

The signal-to-noise and distortion ratio (SINAD) is the ratio of the power of the fundamental and the total noise and distortion power within a certain frequency band, while the effective number of bits (ENOB) is the resolution of an ideal equivalent ADC with SNR (given by the only quantization noise) equal to the SINAD of the actual (non-ideal) ADC. The ENOB is determined by $ENOB = (SINAD - 1.76)/6.02$.

There are several well-known ADC architectures with different properties making them more or less suitable for a certain specification. As for a thorough investigation of their advantages and limitation see [1] and [12]. Here, we focus our attention on the architecture of a particular converter, a pipelined ADC.

2.2 The Pipelined ADC

The system we are going to examine has been studied, discussed and projected in [10]. For our purposes it is sufficient to give a general overview of its properties. Anyway, let us give first a general idea about the basic elements of this particular architecture.

2.2.1 The Pipeline Architecture

The pipelined ADC is the most popular architecture for high speed, high resolution applications. It consists in general of several pipelined stages, each containing an S/H circuit, a low-resolution A/D subconverter (sub ADC), a low-resolution reconstruction DAC, a subtractor and a residue amplifier. The structure of a pipelined converter stage is shown in Fig. 2.2. Here the S/H circuit is included into the sub ADC.

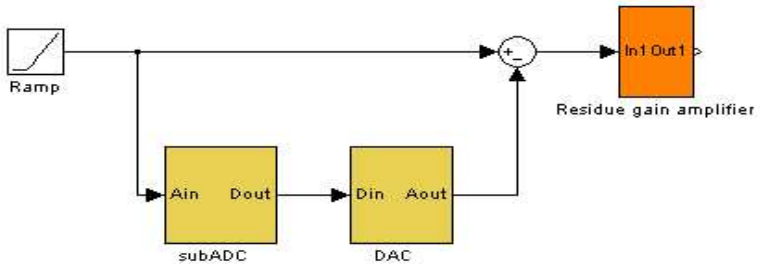


Figure 2.2: Structure of a pipelined converter stage.

For example, a 4-stage pipelined converter is represented in Fig. 2.3. In operation, each stage initially samples and holds the output from the previous stage. Each stage then does a low-resolution A/D conversion on the held input, and the code produced is converted back into an analog signal by a D/A converter. Finally the D/A converter output is subtracted from the held input, producing a residue that is amplified and sent to the next stage. All the digital outputs emerging from the quantizers present in each stage are combined as a final code by using the proper number of delay registers, combinational logic and digital error correction logic, as we shall see later.

Although this operation produces a latency corresponding to the sub-conversion stages before generating a valid output code, the conversion rate depends only on the speed of each stage and the acquisition time of the next sampler, which is dependant on the reconstruction DAC and residue amplifier settling time. As a consequence, the primary potential advantages of the pipelined architecture are high throughput rate and low hardware cost. The high throughput rate of the pipelined architecture stems from concurrent

operation of the stages. At any time, the first stage operates on the most recent sample, while the next stage operates on the residue from the previous sample, and so forth. In other words, thanks to the S/H block interposed between the subtractor and the following stage, the residue is stored before the finer conversion begins. Thus, the front-end stage can start processing the next sample while the following ADC operates on the previous one and so on [12].

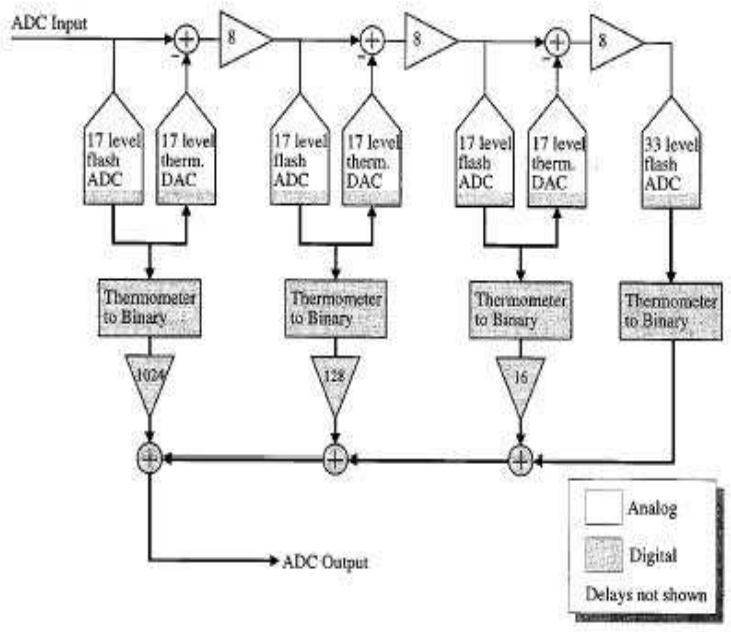


Figure 2.3: Functional diagram of the 14-bit four-stage pipelined ADC examined in this thesis (Figure from [13]).

The area and consequent manufacturing cost of pipelined converters is small compared to those of flash converters because, for instance, pipelined converters require fewer comparators than flash converters [14]. Not only is the area small for pipelined converters, but also it is linearly related to the resolution. In fact, the resolution can be increased by adding stages to the end of the pipeline without increasing the number of clock phases required per conversion.

In the functional block diagram in Fig. 2.3, there are 4 stages. The last one needs only a sub ADC. Sometimes an additional sample and hold (S/H) circuit is used at the input to avoid delay skew errors in the two

input signals to the subtractor. The first three stages each incorporate a 17-level flash ADC and a 17-level switched capacitor DAC, and the fourth stage consists of a 33-level flash ADC. The three 17-level ADCs are nominally identical; each consists of 16 voltage comparators that compare the voltage at the input of the ADC to a set of 16 reference voltages. The 33-level ADC is similar except that it consists of 32 comparators.

The output of each 17-level ADC is the set of its 16 1-bit comparators outputs, and that of the 33-level ADC is the set of its 1-bit comparators outputs. This type of digital encoding is referred to as *thermometer encoding*, and the digital value of each ADC output is interpreted as the number of its comparator outputs that are high.

The three 17-level DACs are each implemented using 16 switched capacitor 1-bit DACs that share a common summing node. In each case, the 16 1-bit DACs are driven directly by the 16 comparator outputs from the corresponding 17-level ADC. In each stage except for the last, the difference between the ADC input and DAC output is amplified by an *interstage gain* of 8. The choice of this factor is discussed in section 2.2.2; now we only point out that in the absence of ADC and DAC errors, the interstage gains are such that just under half of the no-overload ranges of the ADCs in the second through last stages are ever used. Therefore, the choice of 8 for the interstage gains ensures slightly more than a 100 % margin for signals to overrange (i.e. to exceed their ideal ranges), as a result of non-ideal analog circuit behavior, without overloading the ADCs.

Prior to arithmetic processing, the output of each ADC is converted from thermometer encoded data to *two's-complement binary encoded* data $D_{out}(i), \quad i \in \{1, 2, 3, 4\}$. As a consequence, binary conversion of 17 and 33 levels leads to the exploitation of respectively 5 and 6 bits in $D_{out}(i)$. After the conversion, the digital signals from the ADCs are scaled and added as shown in Fig. 2.3. It can be verified that in the absence of non-ideal circuit behaviour the pipelined ADC of Fig. 2.3 performs uniform quantization with 14 bits of precision. The way in which the digital output of the converter is obtained depends on the value of the interstage gain and will be explained in the following section.

2.2.2 The Digital Correction and the Digital Output of the ADC

The difference between the analog input of the pipelined stage, which is the input of the sub ADC, and the DAC output of the same stage is referred as the *residue*. The transfer function from the input of the stage to the output of the subtractor is sawtooth shaped with the amplitude

$$FS/2(Q_i - 1) \quad i \in \{1, 2, 3\}$$

where we assume that the input signal of each sub ADC range from $-FS/2$ and $+FS/2$ as for the total pipelined ADC. If we define n_i the number of bit in stage i then Q_i is the number of output codes of the sub ADC in stage i, in our case equal to 17. The swing of the residue is thus 2^{n_i} times smaller than the swing of the input signal to the stage and it is necessary to amplify the residue in order to utilize the entire swing of the following stage. Choosing the gain in the stage as $G_i = 2^{n_i}$ makes the swing equal to the FS of the following stage.

The digital outputs $D_{out}(i)$ of the stages must be combined to generate the total output code of the pipeline converter. The output signal of the first stage is the input signal to the second stage. Hence each segment of the sawtooth shaped output signal of the first stage will be quantized by the second stage. It can be shown ([1]) that the total number of bits for the m stages is given by

$$n = \sum_{i=1}^{m} n_i. \tag{2.8}$$

while D_{out}, the total digital output is

$$D_{out} = \sum_{i=1}^{m} \left(D_{out}(i) \cdot \prod_{k=i+1}^{m} 2^{n_k} \right). \tag{2.9}$$

We will illustrate how to use (2.8) and (2.9) by an example. Assume that we have a four stage pipelined converter with the stage resolutions $n_1 = n_2 = n_3 = 4$ and $n_4 = 5$ bits. If we choose $G_1 = G_2 = G_3 = 16$, then we get $n = 17$ and

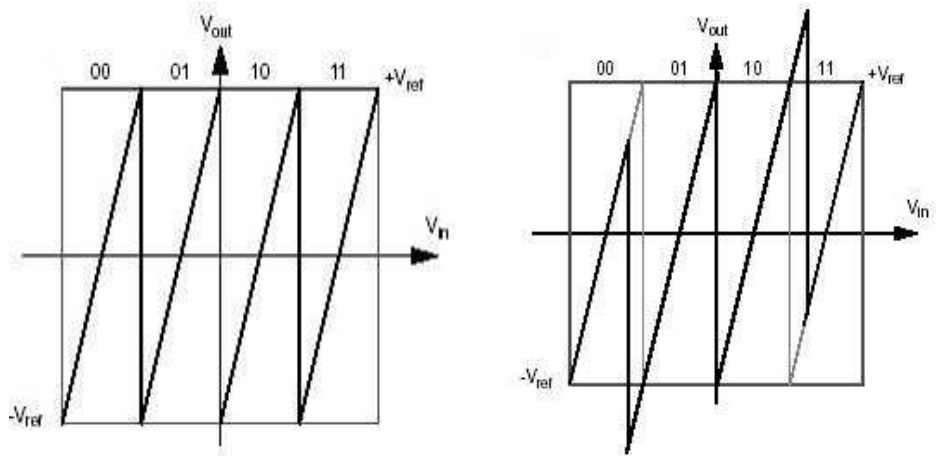
$$\begin{aligned} D_{out} &= 2^{n_2} \cdot 2^{n_3} \cdot 2^{n_4} \cdot D_{out}(1) + 2^{n_3} \cdot 2^{n_4} \cdot D_{out}(2) + 2^{n_4} \cdot D_{out}(3) + D_{out}(4) \\ &= 8192 \cdot D_{out}(1) + 256 \cdot D_{out}(2) + 32 \cdot D_{out}(3) + D_{out}(4). \end{aligned}$$

Since multiplication by 2 is easily performed, in the digital domain, by shifting, the total output code in this example is calculated with no digital processing except shifting. This is not the case of our converter, where *digital error correction (DEC)* [1] is extensively applied to relax the requirements on the comparators. In fact, in a pipelined converter where $G_i = 2^{n_i}$, the decision levels in the sub ADCs, and thereby also the comparators, must be very accurate if the total resolution of the converter is high. If a decision level in a sub-ADC is moved the stage output signal swing becomes larger than FS and the following sub-ADC will be overloaded and will saturate. Hence there is a large conversion error and the effective resolution of the converter is reduced.

To give a more impressive idea about the necessity of the digital correction, consider Fig. 2.4. Here V_{in} is the input of a 2-bit stage, V_{out} is the output of the amplifier and represents the amplified residue. We assume once more that $FS/2 = V_{ref}$ so that input signal of the sub ADC ranges from $-V_{ref}$ and $+V_{ref}$. The transfer function from the input of the stage to the output of the subtractor is sawtooth shaped with the amplitude $V_{ref}/2^2$. In order to utilize the entire swing of the following stage, which is equal to the first one, it is necessary to amplify the residue. Choosing the gain in the stage as $G = 4$ makes the swing equal to $2 \cdot V_{ref}$, i.e. the FS of the following stage. This reasoning could be easily applied if the system was ideal and, in particular if there were no errors affecting sub-DAC decision levels. On the contrary, in a flash converter, the decision levels depend on the thresholds of the comparators, which are subject to fluctuations because of non-idealities. Fig. 2.4-*b* shows what happens when decision levels move: the residue output swing becomes larger than $V_{ref}/4$ and, if amplification is not reduced, the sub-ADC of the next stage will be overloaded.

To overcome this problem, in a pipelined ADC with digital correction, the residue gain is reduced and the output of the following stages is digitally corrected to compensate the reduced gain. For instance, in our case the residue gain has been reduced from 16 to 8, the output signal swing of the first, second and third stages is now only one half the input range of the following stages: this means that some of the codes in the second, third and fourth ADC will never be used. This is why the total resolution is 14 bits and not 17 as we calculated before without taking into account the digital correction. All the codes in the first stage are used but only

$$\left[(Q_2 - 1) \cdot \frac{G_1}{(Q_1 - 1)}\right] + 1$$



a) *b)*

Figure 2.4: Transfer characteristic of a 2-bit stage from the input to the amplifier output in the ideal case (*a*) and with errors affecting the decision levels of the sub-ADC (*b*)— The figure shows also the digital output with respect to the input voltage V_{in} (Figure from [10]).

codes are used in the second stage and only

$$\left[(Q_i - 1) \cdot \frac{G_{i-1}}{(Q_{i-1} - 1)}\right] + 1$$

are used in the i-th stage [3]. In other words, when all stages, apart from the last, have their output digitally corrected, the residue gain of all stages, apart from the last, for which it is not necessary to generate the residue, are reduced and redundancy is introduced in all stages, apart from the first. If x_i is the number of bits used by the i-th stage for digital correction we define the effective resolution of the stage $n_{eff,i} = n_i - x_i$ and now equation (2.8) must be rewritten like this:

$$n = \sum_{i=1}^{m} n_{eff,i}. \tag{2.10}$$

It is convenient to introduce r_i as the resolution remaining to be determined

[3] Remenber that all the converter are supposed to be midtreader, as explained in section 2.1.

from stage $i+1$ to stage m

$$r_i = n - \sum_{k=1}^{i} n_{eff,k}. \tag{2.11}$$

The situation for our converter is illustrated in Table 2.1.

Parameter	1^{st} **Stage**	2^{nd} **Stage**	3^{rd} **Stage**	4^{th} **Stage**
n_i	4	4	4	5
x_i	0	1	1	1
$n_{eff,i}$	4	3	3	4
Q_i	17	17	17	33
G_i	8	8	8	/

Table 2.1: Prospect of the parameters for the Pipelined ADC.

G_i is usually reduced by a factor 2 but can in principle be chosen arbitrarily. However it is usually desirable to have the same step size for all the codes in the following sub-ADC and the gain is then restricted to values that give the correct step size at the decision levels in the first stage. In this case it can be seen that moving the decision levels will not cause saturation in the following stage since there are now redundant codes in it.

The digital correction can correct errors in the comparators as long as the residue is within the FS range of the following stage. The smaller the residue gain, the larger errors can be accepted. The maximum decision line deviation allowed without large conversion errors is given by

$$\Delta V = \pm \frac{FS}{2}\left(\frac{1}{G_i} - \frac{1}{2^{n_i}}\right) \tag{2.12}$$

and if the gain factor is reduced by a factor 2 we get $\Delta V = \pm\frac{FS}{2\cdot 2^{n_i}}$. Hence the maximum error allowed in ADC_i can be $\pm LSB_i/2$. Equation (2.9) must be generalized like this:

$$D_{out} = \sum_{i=1}^{m}\left(D_{out}(i)\cdot\frac{2^{n_m}}{2^{n_i}}\prod_{k=i}^{m-1}G_k\right). \tag{2.13}$$

For example, let us consider our converter shown in Table 2.1; by applying (2.13) we get:

$$\begin{aligned} D_{out} &= \frac{2^{n_4}}{2^{n_1}}G_1G_2G_3D_{out}(1) + \frac{2^{n_4}}{2^{n_2}}G_2G_3D_{out}(2) + \frac{2^{n_4}}{2^{n_3}}G_3D_{out}(3) + D_{out}(4) \\ &= 1024\cdot D_{out}(1) + 128\cdot D_{out}(2) + 16\cdot D_{out}(3) + D_{out}(4). \end{aligned}$$

It is easy to verify that in this case the digital codes from the stages overlap when they are added and digital logic must handle carry propagation in the addition. In this way we have explained why the result has a 14 bit resolution. Because of the conversion from thermometer to binary encoding all output digital data are encoded in 5 bits, apart from the last stage, for which we need 6 bits. Anyway, even if all input dynamic range is exploited, only the first stage MSB is used, while the others remain equal to zero in ideal conditions. Moreover, it should be pointed out that the architecture of the sub ADC must be adequately adapted in order to perform the DEC through the use of *redundant signed digit* converters [1]. The DEC technique allows commensurating the sub ADC nonidealities with the stage resolution instead of with the resolution of the entire ADC. For instance, the digital correction block detects overrange in the outputs of the second stage and "changes" the output of the first stage by 1 LSB at an n_1 bit level if overrange occurs. Of course, this reasoning cannot be repeated for the DAC of the first stage. The DAC of the first stage must therefore have a linearity which corresponds to the full resolution of the pipelined ADC. Hence, with digital correction, nonlinearities in the A/D subconverters can be corrected only if the D/A converters are ideal. The linearity of the DACs is influenced by component matching; for high resolutions some calibration algorithm is needed. This problem will be discussed in sections 2.2.3 and 2.2.4.

2.2.3 Errors and Non-idealities

In this section we show how different error sources affect the performance of the pipelined ADC ([15]). Let the error in stage i be e_i, the input referred error e_{in} can be expressed like this

$$e_{in} = e_1 + \sum_{i=1}^{3} \frac{e_{i+1}}{G^i}; \tag{2.14}$$

then to keep the errors smaller than LSB/2 (which guarantees no missing codes) we have

$$e_i \leq \frac{FS}{2^{n+1}} G^{i-1} \tag{2.15}$$

where $G = G_1 = G_2 = G_3 = 2^{n_i - x_i}, \quad i \in \{1, 2, 3\}$. It is evident that larger relative errors are acceptable for the LSB stages of the converter. Error contributions of the second through the fourth stage are diminished if a large interstage gain is chosen. This indicates that large resolutions in the stages

are preferable, but leads to larger power dissipation and to difficulties in designing a high-speed amplifier with a large gain factor (see section 2.2.5).

The error in one stage of the pipelined converter can in principle appear at four different places:

- at the input of the sub ADC;
- in series with the DAC;
- at the input of the subtractor;
- at the input of the amplifier or in series with the output of the stage.

For most practical situations the DEC technique is sufficient to correct errors in the sub-ADCs as seen in section 2.2.2; therefore any static errors in the sub-ADCs are of minor concern. The third category of error will have the same effect as a DAC error and we need not consider the error source separately. DAC errors and amplifier errors remain. DAC errors are examined in [1], [15] and [10]. Some types of these errors, such as offset errors and gain errors can be treated as sub-ADC errors or amplifier errors, while linearity errors are more serious, dependent on the implementation and, in our case, overcome by digital calibration discussed in section 2.2.4. Our purpose is to optimize the amplifier design in order to minimize gain and linearity errors without too raising power consumption. Therefore we will focus our attention on the errors introduced by the amplifier. In general we know that

- errors appearing at the output of the subtractor in the first stage must be within the resolution of the remainder of the pipeline
- errors at the input of the subtractor must be within the total resolution of the converter.

Whether an error should be modelled as appearing at the input or the output of the subtractor depends on the circuit solution.

S/H Amplifier Errors

We have seen that the S/H function is necessary in each stage of the converter in order to enable concurrency, so that the input signal of the second stage sub ADC remains stable and can be easily processed by later stages even if

the input signal of the first stage sub ADC has changed in compliance with the specified throughput rate.

The S/H function can be obtained free if a switched capacitor amplifier is used in CMOS technology. When this is the case the interstage residue amplifier is also referred as S/H amplifier (SHA) because the same block, working in the sampled signal domain, performs also the sampling and holding functions, together with the subtraction and multiplication operations. In most of SC implementations, as in our case, the same circuit performs the functions of the DAC and the SHA and in this case it is usually referred as multiplying DAC (MDAC). This means that actual implementation combines all the above functions in one circuit.

The relative gain errors, ϵ_G must meet the relation [15]:

$$\epsilon_{Gi} \leq \frac{1}{2^{r_i}} \tag{2.16}$$

where r_i is the resolution remaining in the stages after stage i. This shows again that the MSB stages must have a more accurate gain compared to the LSB stages and that a high stage resolution is needed to reduce the effect of such errors. This condition will be translated into specifications for the amplifier in section 3.2.2 where two types of gain errors will be discussed, the actual amplification error ϵ_A due to the finite op amp gain, and the limited bandwidth error ϵ_{BW} due to the finite settling time of the op amp. Once the DAC errors in the first two stages have been compensated using DNC (see next section), most of the remaining performance degradation relative to the ideal case arises from the interstage gain error between the first and the second stages. This is a source of converter nonlinearity because it causes vertical jumps (missing codes) in the overall converter transfer characteristic when the interstage gain is too low, and horizontal jumps when the interstage gain is too high [16]. In addition to this, there is also the intrinsic nonlinearity of the amplifier characteristic.

Nonlinear distortion introduced by the amplifier becomes more and more important when trying to minimize power consumption of the stage, which is around $150mW$, according to the simulations. This power dissipation is dominated by the precision amplifier necessary to respect the conflicting low noise and high bandwidth requirements to achieve the desired accuracy [4]. Our proposal is to relax constraints on the amplifier gain and bandwidth, which translates into constraints on bias current, even if it will deteriorate

[4]See also section 2.2.5

the first stage linearity. The idea is to correct that distortion by means of a digital block able to compensate it. Recently, for this purpose, a pipelined ADC using open-loop residue amplification has been reported, and a digital background calibration technique has been proposed to replace precision amplifiers [17]. By examining the amplifier open loop characteristic, deviations of the residues from ideality can be decomposed into two components. First, an error in the linear gain term of the amplifier results in a non-ideal transition height around comparator decision levels. Secondly, nonlinearities in the amplifier cause curvature in the residue segments. By appropriately design the gate overdrive V_{ov} of the differential pair, cubic gain compression becomes the dominant odd order non-linearity error. Even order nonlinearities are also present, caused by mismatch, but we initially neglect them since, as we said in section 1.2.1, we provide a platform to support nominal design. Only non-idealities of the MOS behavior are considered.

If we want to correct these non linearities the first step is to characterize the amplifier static non-linear behavior and to introduce some performance figures in the performance model which could adequately represent that behavior. We are interested into the static behavior since, as shown in chapter 3, we work in the sampled signal domain and if the amplifier is adequately sized so that it settles within one half of the clock period, when the output signal is sampled, all transients may be considered reasonably negligible. The characterization process will be useful for two orders of motives:

- First of all, by applying platform based methodology to the op amp, we could obtain accurate system level models. Using these models in an optimization loop it is possible to estimate quantitatively how much power can be actually saved without too compromising the converter linearity and accuracy. The op amp design can be consequently optimized with respect to power consumption.

- Secondly, the exploration phase can furnish an amount of data that make easier to figure out novel strategies of calibrations to compensate the op amp distortion.

2.2.4 DAC Noise Cancellation

Switched capacitor based pipelined ADCs tend to be highly sensitive to errors arising from component mismatches in their internal DACs (Digital-to-analog converters). Unlike other types of noise in a conventional pipelined

ADC, this error, referred to as *DAC noise*, introduced by the first stage DAC is not attenuated or cancelled along the pipeline. With present VLSI circuit technology it is difficult to match capacitors to better to 0.1 %. This translates into an A/D conversion limit of about 11 bits in pipelined ADC architectures without some form of cancellation. An all-digital technique, referred to as DAC Noise Cancellation has been applied to the 14 bit converter with the purpose of mitigating the problem, as shown in [13]. The technique continuously measures and cancels the portion of the ADC error arising from DAC noise during normal operation of the ADC, so no special calibration signal or auto-calibration phase is required.

In [13] is shown that a significant performance improvement can be obtained by digitally removing from the pipelined ADC output the error components resulting from the noise introduced by the 17-level DACs in the first two stages. The ADC architecture modified to apply the DNC technique is shown in Fig. 2.5. It differs from the conventional version in two respects. First, the 17-level DACs in the first two stages have been replaced by modified DACs referred to as *dynamic element matching* (DEM) DACs. The DEM DACs differ from the thermometer encoded DACs described in section 2.2.1 in that each contains a *digital encoder* that randomly permutes the connections between the 16 thermometer encoded input bits and the 16 switched capacitor 1-bit DACs. Second, two blocks, labelled *DNC logic* in Fig. 2.5, have been included. Each of them generates estimates of the error components in the ADC output arising from the noise introduced by the associated DEM DAC. These estimated error sequences are subtracted from what would otherwise be the output sequence in a conventional pipelined ADC. The working principles of the new blocks are explained in [13], while an attempt to design them, exploiting a top-down methodology, is present in [10].

Here we want to spend some words about the dynamic randomization technique. The DAC capacitors are ideally identical but in practice random inequalities cause conversion errors since the charge stored depends linearly on capacitance values. Moreover, errors depend on signal level if we think that in order to convert a digital value a from the sub-ADC the same a capacitances are always charged. Thus, let only one capacitor be different from the others, every time a signal solicits that capacitor, the output will be distorted because that reference capacitor is associated with specific input codes [1].

The solution adopted in the DEM DACs allows selecting in a pseudo-random manner the capacitor to charge; it can be seen that if the algorithm,

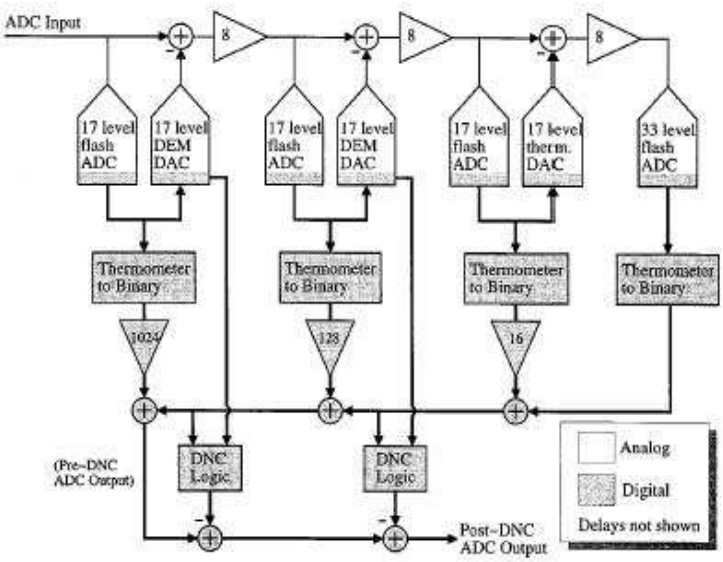


Figure 2.5: Functional diagram of the 14-bit four-stage pipelined ADC with DNC applied to the first two stages (Figure from [13]).

referred as "scrambling algorithm", can use on average all the possible permutations we have on average only a gain error, which gives less problem than non-linearity. In other words we can introduce a circuitry, called *randomizer* or *scrambler* so that, at different times, different references are chosen to represent a particular code. If we choose the references in a way that is uncorrelated with the signal, the matching errors will no longer be signal dependent and hence *the error becomes noise*. In real implementation a pseudo-random binary sequence generator, which does not give a completely uncorrelated error signal, but the improvement can be significant. The DEM technique is not able to correct the error but it simply gives to the error a random statistic distribution so that the mean value of the error is zero. The actual error cancellation in the architecture that we have considered is due to the DNC logic blocks that act directly on the digital output of the converter.

2.2.5 Power Reduction Techniques and Trade-offs

The pipeline architecture allows reducing the resolution of each stage by increasing the number of stages in order to obtain the desired resolution. The most basic architectural decision is to choose the resolution per stage.

This is complicated by the fact that resolution is not obliged to be equal for all stages. Moreover, for efficient use of the conversion range of each stage, this choice determines the corresponding value of the interstage gain.

The first trade-off to consider is the famous power-speed trade-off which reflects in choosing per-stage resolution. Diminishing this factor leads to increase the number of stages to reach total converter resolution thus increasing also power dissipation; however this maximize the bandwidth of the SH/Gain SC circuit which limits the overall conversion rate ([18]). In fact, in order to perform fast interstage signal processing, the output of the operational amplifier in the SC circuit has to settle in half the clock period to the given accuracy of each stage prior to the next stage sampling instance. Since the bandwidth of the SC interstage amplifier depends on its interstage gain, we should choose the per-stage resolution which allows a low closed loop gain configuration, essential for fast settling. Low per-stage resolutions lead to configurations with low load capacitances (composed of sampling capacitors of the next stage and input capacitances of comparators in the flash A/D section) and large feedback factors. As a result a large interstage amplifier bandwidth is achieved compared to that of larger per-stage resolution.

Conversely, large resolution and corresponding gain per stage are desirable to achieve high linearity because, as we have seen in section 2.2.3, the contributions of nonidealities in all stages after the first are reduced by the combined interstage gain preceding the nonideality. Thus the second trade-off is between the speed and linearity requirements, that conflict in determining the optimum resolution per stage ([14]).

It is difficult to find this optimum resolution since it depends on both the specifications and the circuit technique used for the implementation. For instance, it is known ([10]) that a 1-bit DAC has no non-linearity errors, and since we would like to avoid nonlinearity errors, we are induced to use DAC with resolution as low as possible. However, a low resolution DAC means a low resolution stage, in conflict with the high resolution required to minimize the effect of SHA gain errors. This is a third trade-off which can be pointed out between DAC linearity and SHA errors, i.e. the most important error sources in modern pipelined ADC, which are also the most difficult to correct.

On the other hand, in pipeline A/D converters a major portion of the total power dissipation is from the static power dissipated in analog circuit components that require dc bias currents, such as precision comparators and op amps. The charging and discharging of sampling capacitors, clock drivers

and digital circuits contribute a relatively small amount to the overall power dissipation. Thanks to digital correction we need no more precision comparators and so static power dissipation of them can be eliminated ([18]). Also, a substantial power reduction can be achieved by using the minimum possible size of sampling capacitors at each point in the pipeline, as dictated by thermal noise considerations.

As we shall see in section 3.2.1, a fundamental noise source present in A/D converters is thermal noise, and the magnitude of this noise is a function of the sampling capacitor size ($\sigma^2_{thermal} \sim kT/C$). Therefore, in the ideal case, the minimum achievable power dissipation in a MOS sample and hold circuit is set by the maximum allowable value of this kT/C noise to achieve the required SNR before quantization. This sets the minimum sampling capacitor value, which in turn sets the minimum power dissipation for a given sample rate assuming the capacitor must be completely discharged on each sample period. In practice the dissipation achieved is much more because the S/H power is dominated by dissipation in the operational amplifier that drives the sampling capacitor. As a practical matter, power minimization in the overall A/D converter translates to minimizing the power in the active circuitry driving the sampling capacitor whose kT/C noise limits the SNR of the converter.

In the pipeline architecture this again translates into minimizing the SC circuit power in each stage. In order to do so, the minimum allowable value of sampling capacitor must be used at each point of pipeline, since it becomes the load capacitance of the previous stage and the size of the amplifier is proportional to that of the capacitor for given speed. Thus, optimization of the power dissipation of each of the operational amplifier in the pipeline can be performed taking into account the source, load, and feedback capacitors seen by each one. Since the stage requirements on the speed and accuracy become less stringent as the stage resolution decreases down the pipeline, stages in the later part of the pipeline can be scaled down by using smaller sampling capacitors and op amps. In other words, if the first stages have high resolutions the design of the following stages becomes easier since they can tolerate more noise and the accuracy required is less than for the first stage, the amplifiers have small capacitances and dissipate less power. In this case, the sizes of sampling capacitors and op amps near the front end are determined by the noise floor, while towards the end of the pipeline, parasitic capacitances begin to dominate and settling time requirements determine the size of each stage.

Due to this "scaling" possibilities, in general a good rule of thumb is

to use a low stage resolution of 2-3 bits for converters up to 10 bits. For higher resolutions the first stages should have a higher resolution of 4-5 bits; otherwise low resolution first stages would require a precision too high with respect to the resolution of the single stage.

2.3 Conclusions

In this chapter an overview is given of the global A/D-converter system. An accurate analysis of the causes of error is given. Most of these errors are corrected using standard procedures (e.g. Digital Error Correction) or calibration algorithms (e.g. DAC noise cancellation). Another error cause is distortion due to the amplifier non-linearity. This is not generally treated when designing ADCs. In next chapters we shall concentrate our attention on the first stage residue gain amplifier. With the purpose of evaluating trade-offs between power and linearity we have stated the importance of considering also the op amp weak-distortion behavior during the amplifier characterization phase. The objective is to build an op amp analog platform, which encapsulates also these aspects. The future phase of characterization of the op amp will furnish models for system level simulation and op amp design optimization with respect of power consumption. This exploration phase will also help us to figure out novel strategies of calibrations to compensate the distortion effects.

Chapter 3

Design Considerations for the Interstage Amplifier

In this chapter the most important steps are described, followed while trying to re-design the interstage amplifier used in the converter. Performances derived by system level specifications and equations derived by circuit analysis are later exploited to generate an Analog Constraints Graph (ACG) for the amplifier architecture. The graph will be useful for guiding the operational amplifier characterization phase.

3.1 Introduction

It is well-known that the design of MOS operational amplifiers (op amps) is not an exact scientific process. Up to now, analog system level design has been carried out only through experience and trial and error. Typically the circuit must satisfy many requirements, often including conflicting ones. In chapter 2 Analog Platforms are introduced to overcome some of these problems. We must precise that it is almost impossible to design a modern IC analog system (not only a converter, but also a CMOS SC filter or a radiofrequency receiver) starting from the simplified model equations, since the errors induced are too large. First of all, in deep submicron processes it is impossible to usefully exploit the squared-law model for a CMOS process because many second order physical effects not included in the simple model have a significant impact on MOS characteristics which becomes less and less ideal. It is sufficient to cite short-channel effect, subthreshold current and all other parasitic effects, which are not negligible. Moreover, complexity of

analog circuits, even if smaller than that of digital circuits, can translate in some cases into a huge quantity of equality and inequality constraints that the designer has to consider in order to reach some specifications. In general it is impossible to cope with them using hand calculations. Eventually, the behavior of an analog block is different on the bases of the circuit loading that block. Analog circuits cannot be designed separately since their interrelations must be taken into account (see chapter 1).

Consequently the only instrument a designer can rely on is the simulator which exploits accurate model very complicated but quite accurate. The traditional analog design is carried-out at the transistor level (*full custom design*) and design space exploration is quite limited because of the time needed to evaluate alternative solutions with circuit simulators. This is why today system level analog design is a design process dominated by heuristics. Usually what is achieved is just a feasible point at the system level, while optimality is sought locally at the circuit level.

Analog platform based design methodology allows an effective exploration of the design space. In order to do this, it is not possible to use simplified equations for the motives we have already exposed. One alternative should be to size randomly a circuit, to simulate that particular configuration of parameters, then to register the results obtained as for feasibility issues and performance figures. But, do we really need to simulate all the random configurations possible or the exploration can be biased by some required specifications we are looking for and that permit to limit the set of variables? The characterization of a platform over its input space is exponentially complex with the dimensionality of $\mathcal{I}$. This problem requires some hints from the analog designer to limit the dimensionality of $\mathcal{I}$, i.e. we need to know which "knobs" are most meaningful for defining the platform and which constraints on the value of each parameter can be exploited to prune the characterization space.

Thus, we want to simulate a set of configurations which should at least verify some general constraints, beyond which we are already sure that the circuit is not feasible or it is not suitable for our particular application. In order to obtain these constraints we are compelled to turn once more to the old simple equations or to model more complicated, but also easily manageable at least by a computer. This allows us to carry out a coarse characterization of the system and to decide which parameters are really important for a particular result since they influence a particular performance figure we are interested in. This is why a coarse re-design of the circuit has been necessary.

Most of the following considerations are based on an existing design carried out in Pavia, at the *STMicroelectronics® Studio di Microelettronica*[19] so that some design iterations were already done when we started building an Analog Platform (AP) of the amplifier. First simulations, extended to the entire residue first stage of the converter, allowed finding the amplifier specifications. S a consequence, we will not build a generic purpose AP; in this case, the platform design space can be made more and more biased toward a particular application since the designer understanding of the system has improved.

Starting from specifications, we first calculate the specified design parameters for our particular architecture, using the simplified square-law model for MOS transistor; in next chapter we map the design effort, in a more "scientific fashion", in an ACG, exploiting more accurate, although simplified, device models. We remind that the amplifier has to be designed in a 0.13 μm CMOS process.

3.2 Specifications

The interstage amplifier is a *Switched Capacitor (SC)* gain stage that multiplies the difference between the analog input signal and the digital converted one by 8. The stage uses a *fully differential operational amplifier*, which have a differential input and produce a differential output. Fully differential op amps are widely used in modern integrated circuits because they have some advantages over their single-ended counterparts:

- they provide a larger output voltage swing;
- they are less susceptible to common mode noise or interferences;
- since they are *balanced circuits*, i.e. symmetric with perfectly matched elements on either side of an axis of symmetry, even-order non linearities are not present in the differential output (only in the ideal case);
- they eliminate systematic offset voltage, so that only the offset component due to mismatch remains.

A disadvantage of fully differential op amps is that they require two matched feedback networks and a common-mode feedback circuit to control the common-mode output voltage. This makes the circuit more complicated;

the chip area required is 50–100 % larger than for single ended realization of the same network.

Introducing such architecture becomes compulsory in *Mixed Signals Integrated Circuits (IC)* with V_{DD} *bounces* and *ground* or *substrate* voltage fluctuations to guarantee a high *Power Supply Rejection Ratio (PSRR)* and *Common Mode Rejection Ratio (CMRR)* at high frequencies notwithstanding parasitic capacitances. In addition, it can be shown ([20], [21]) that the effective output voltage swing is doubled by the balanced op amp configuration, while the input circuit (and hence most of the noise) remains the same as for the single ended output op amps. Thus, the dynamic range is greater by nearly 3 dB than for single ended op amps.

Table 3.1 gives a short summary of specifications required for the amplifier. Next section describes how these specifications have been derived from system level performance constraints. Later, we shall describe how they can be satisfied.

Parameter	**Symbol**	**Value**
Input Capacitance	C_S	32 pF
Feedback Capacitance	C_F	4 pF
Load Capacitance	C_L	4 pF
Supply Voltage	V_{DD}	2.5 V
Differential Input Signal (peak-to-peak)	v_{di}	2 Vpp
Input Common Mode Signal	V_{IC}	1.25 V
Input Signal-to-Noise-Ratio	SNR_i	$>$ 14 bit
Clock Frequency	f_c	80 MHz
Closed Loop Low Frequency Gain	A_F	$>$ 7.9
Open Loop Low Frequency Gain	A_{vol0}	$>$ 55 dB
Op Amp Transconductance	G_m	$>$ 76 m℧
Phase Margin	PM	$>$ 45°

Table 3.1: Specified values for the amplifier circuit parameters.

3.2.1 Input, Feedback and Load Capacitances

The closed loop circuit can be studied using a differential-mode (DM) half-circuit technique to model the small-signal behavior of our OTA [20]: the half circuit is shown in Fig. 3.1 where the fully differential amplifier has been

substituted with an equivalent single-ended op amp with the same transconductance as the original amplifier. Each switch in the schematic is controlled by one of two *nonoverlapping* clock phases ϕ_1 and ϕ_2: we assume that each switch is closed when its controlling clock signal is high and open when its clock signal is low. ϕ_1 and ϕ_2 are never both high at the same time; therefore the switches controlled by one clock phase are never closed at the same time as the switches controlled by the other clock phase.

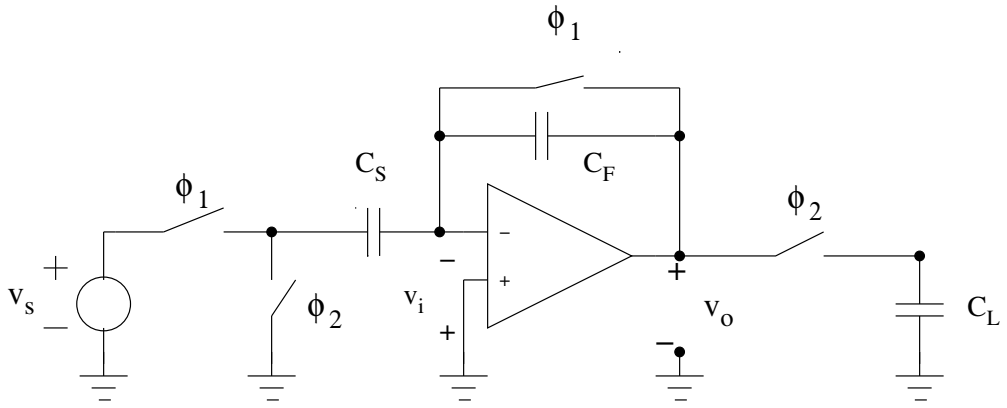


Figure 3.1: Single ended equivalent circuit for the SC interstage amplifier.

We define the DM source voltage as the difference of the source voltages applied to the amplifier inputs, $v_{sd} = v_{s1} - v_{s2}$, the DM op amp input voltage as the difference of the amplifier input voltages, $v_{id} = v_{i1} - v_{i2}$, and the DM output voltage as the difference of the output voltages $v_{od} = v_{o1} - v_{o2}$. It can be demonstrated that it is sufficient to analyze the circuit in Fig. 3.1 to obtain the original circuit transfer function and the time domain behavior. In fact, if we impose for the voltages in Fig. 3.1 $v_s = v_{sd}/2$, $v_i = v_{id}/2$ and $v_o = v_{od}/2$, then the single-ended version of the circuit is equivalent to each one of the DM half-circuit deriving from the original configuration. Since $\frac{v_o}{v_s} = \frac{v_{od}}{v_{sd}}$ the system in Fig. 3.1 can be referred as the single-end circuit equivalent to the fully differential one. Thus, if the op amp is ideal (infinite input impedance, zero output impedance and infinite gain) we obtain in the z-transform domain:

$$\frac{v_{od}}{v_{sd}} = \frac{C_S}{C_F} z^{-1}. \tag{3.1}$$

The input sampling capacitors have been sized to match the specification of maximum input noise. We suppose that the equivalent input noise is lim-

ited only by thermal noise ($\approx kT/C$ [1]); this is equivalent to consider only thermal noise injected by the input switch. Actually both thermal and flicker ($1/f$) noise are generated in the switches and op amp, but in [21] is shown how the effect of the wide-band thermal noise in SC circuits, is often much more important. Due to the internal sampling and holding performed by the switches and capacitors both types of noise are replicated in the frequency domain. However, for usual parameter values, this does not lead to appreciable aliasing of the $1/f$ noise, since the flicker noise corner frequency f_{cr} (usually in the range of $1 \sim 50$ kHz, with 10 kHz as a typical value) is usually well below $f_c/2$ while the thermal noise will get seriously undersampled, and hence aliased replicated so that it becomes the most important noise component. As regards the op amp noise, it is not in general negligible and will be discussed later in section 3.3.2. What practically happens is that when a capacitor is connected to the input signal through a switch, the aliasing due to the sampling of the noise concentrates the full noise power of the switch resistor, kT/C, into the baseband. In order to avoid the large impact of the thermal noise on the resolution of the converter, the total input noise power, as a function of C_S should be smaller than the quantization noise. In this case, when the thermal noise power is equal to the quantization noise power the SNR decreases by 3 dB. Therefore the input capacitor in the actual implementation should be larger to give some margin for other error sources. The obtained value for C_S is computed supposing a 15 bit equivalent resolution, that is 1 bit margin. From 3.1 we find that a 4 pF feedback capacitor is necessary to reach a gain equal to 8. The input sampling capacitors of the second stage (second residue stage and flash converter) realize the load for the first one and its value present in table 3.1 has been estimated.

Note that in the other stages thermal noise is attenuated when referred to the input; hence the input capacitor sizes will be smaller than for the first stage.

3.2.2 Gain and Bandwidth

The small signal simplified circuit in Fig. 3.2 derives from circuit in Fig. 3.1 with the exception that the small signal equivalent circuit of the op amp has been introduced. A Norton equivalent configuration has been adopted and input impedance has been assumed to be infinite. Because of the op amp

[1] In this expression k is the ubiquitous Boltzmann's constant, T is the absolute temperature and C the sampling capacitance. At room temperature $kT = 4.16 \cdot 10^{-21}\,\mathrm{V \cdot C}$.

symmetry, only one half of the circuit has been represented and all nodes intersecting the axis of symmetry have been connected to ac ground. In fact, we also imagine the op amp being driven in a balanced way [2], as is typical for a fully differential architecture. We shall analyze this circuit when ϕ_2 is high and ϕ_1 is low in order to evaluate the most important parameters of an op amp in that configuration, such as transconductance, output resistance and gain-bandwidth product.

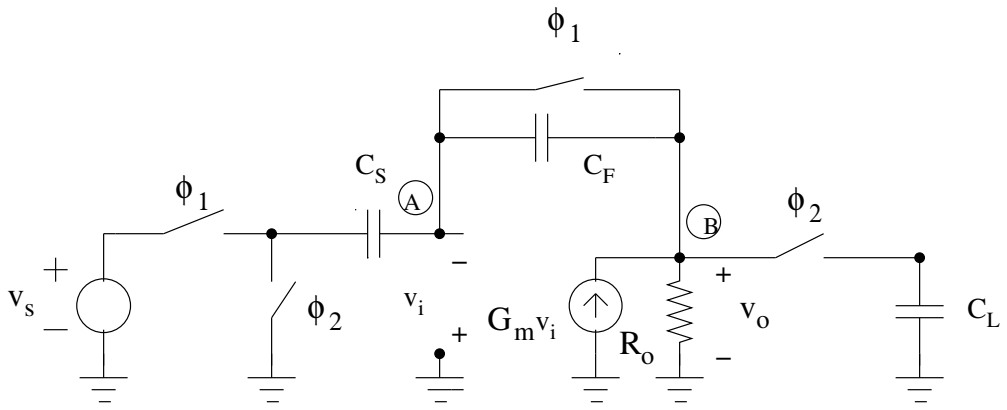


Figure 3.2: Norton small-signal equivalent circuit for the SC single-ended amplifier in Fig. 3.1.

At first, supposing a clock sampling frequency f_c, or a period T_c, we try to estimate the amplifier transconductance G_m in order to reach a particular precision. For this purpose a transient analysis in the time domain can be done trying to determine what happens in the first part of the period since we already know that in the second half of each sampling period op amp output goes to zero. During the first half-period (ϕ_2 high) the circuit behaves as a continuous time system and an ordinary linear differential equation can be used for transient evaluation. As we have already explained in section 2.2.5, the output of the op amp has to settle in half the clock period to the given accuracy of each stage prior to the next stage sampling instance. From this condition we can derive bounds for minimum G_m and minimum bandwidth to be achieved.

Applying Kirchoff's current law (KCL) at the inverting input (node Ⓐ) and at the output node (node Ⓑ), we obtain the following differential equa-

[2] with *differential* input voltages equal in amplitude and opposite in phase

tions:

$$(C_F + C_S)\frac{\mathrm{d}v^-}{\mathrm{d}t} = C_F\frac{\mathrm{d}v_o}{\mathrm{d}t} \tag{3.2}$$

$$C_F\frac{\mathrm{d}v^-}{\mathrm{d}t} = (C_F + C_L)\frac{\mathrm{d}v_o}{\mathrm{d}t} + \frac{v_o}{R_o} + G_m v^- \tag{3.3}$$

where v^- is the inverting input voltage, v_o is the output voltage of the op amp in Fig. 3.1, R_o is the output resistance. We have already stated the correspondences of these parameters with those of the original amplifier. Initial conditions for these equations can be derived by imposing charge conservation and supposing instantaneous charge redistribution. We obtain:

$$v^-(0^+) = -v_S\frac{C_F + C_L}{C_F + C_L + \frac{C_F C_L}{C_S}} \tag{3.4}$$

$$v_o(0^+) = -v_S\frac{C_F}{C_F + C_L + \frac{C_F C_L}{C_S}} \tag{3.5}$$

where v_S is the value of the input signal previously sampled, after ϕ_1 has become low. Solving these equations, this result holds for $0 \le t \le T_c/2$:

$$v_o(t) = v_S\,\frac{C_S}{C_F}\,\frac{-\beta A_0}{1-\beta A_0}(1 - \mathrm{e}^{-\frac{t}{\tau}}) - v_S\,\frac{C_F}{C_L + C_F + \frac{C_L C_F}{C_S}}\,\mathrm{e}^{-\frac{t}{\tau}} \tag{3.6}$$

where is $\beta = \frac{C_F}{C_F + C_S}$, $A_0 = -G_m R_o$ while time constant τ comes from this equation:

$$\tau = \frac{C_L + C_S + \frac{C_L C_S}{C_F}}{G_m}\,\frac{-\beta A_0}{1-\beta A_0} = \frac{C_L + C_S + \frac{C_L C_S}{C_F}}{G_m}\,\frac{G_m R_o \frac{C_F}{C_F + C_S}}{1 + G_m R_o \frac{C_F}{C_F + C_S}}. \tag{3.7}$$

Symbols chosen in preceding expressions will become meaningful later when we shall find the same results in the frequency domain. We would like $v_o(T_c/2)$, the output voltage at time $T_c/2$, be equal to $v_S\frac{C_S}{C_F}$. In practice we have an error because of finite amplifier gain and finite bandwidth. Since gain is finite, output ideal value cannot be reached even for time $t \to +\infty$; since bandwidth is also finite the final value $v_{od}(+\infty)$, even if it was coinciding with the ideal value, cannot be reached in finite time. What is then the maximum absolute error we should tolerate? As shown in [15] and in [1] the error must be smaller than LSB/2 referred to the required resolution of the remaining stages of the pipeline, in our case $r = 10$ bit (see sections 2.2.2

and 2.2.3). Let us simplify the problem by considered the two error sources separately. At first we suppose infinite bandwidth and find minimum low frequency amplifier gain $G_m R_o$ necessary to achieve desired precision. By imposing the following condition on the relative gain error ϵ_A:

$$\epsilon_A = \frac{v_{o_{id}}(+\infty) - v_o(+\infty)}{v_{o_{id}}(+\infty)} \leq \frac{1}{2^r}. \tag{3.8}$$

we get:

$$-\beta A_0 = G_m R_o \frac{C_F}{C_F + C_S} \geq 2^r - 1. \tag{3.9}$$

Now we consider the low frequency amplifier gain infinite, i.e. we neglect gain error and evaluate only error deriving from finite op amp settling time ϵ_{BW}. By imposing:

$$\epsilon_{BW} = \frac{v_{o_{id}}(T_c/2) - v_o(T_c/2)}{v_{o_{id}}(T_c/2)} \leq \frac{1}{2^r} \tag{3.10}$$

we find minimum G_m permitted:

$$G_m \geq 2 f_c \left(C_L + C_S + \frac{C_L C_S}{C_F} \right) \ln \left[2^r \left(1 + \frac{C_F}{C_L + C_S + \frac{C_L C_S}{C_F}} \right) \right]. \tag{3.11}$$

Results found in time domain have been confirmed by an AC analysis of the circuit exploiting feedback theory. Given a proper network decomposition, a cut at the input terminals of the op amp have been done in order to calculate the open loop and closed loop parameters, on the bases of the approach proposed in [22] and [23]. Now, if we define βA the loop gain or return ratio transfer function, where β, introduced above, is the feedback factor transfer function then $-\beta A_0$ is the low frequency loop gain. In fact, the equivalent loading capacitance seen by the output node is C_{LOAD}, evaluated in the open loop network after the cut, and is given by:

$$C_{LOAD} = C_L + \frac{C_F C_S}{C_F + C_S} \tag{3.12}$$

where the op amp input capacitance has been neglected.[3] Performing calculations in the s-transform domain, since the system behaves as continuous time system in the time interval considered, we obtain for the loop gain:

$$\beta A = -\frac{C_F}{C_F + C_S} \frac{G_m R_o}{1 + R_o C_{LOAD}\, \mathrm{s}} \tag{3.13}$$

[3] A more accurate expression of C_{LOAD} is given in section 3.3.2.

It is now evident how equation (3.9) turns into a condition on low frequency loop gain. However we are not interested in the return ratio poles, but in the return ratio gain-bandwidth product ω_G, which approximately gives the –3-dB bandwidth of the feedback circuit and the already cited τ. If $|\beta A| \gg 1$, we obtain:

$$\omega_G = \frac{1}{\tau} = \frac{1-\beta A}{|\beta A|}\frac{G_m}{C_{eq}} \approx \frac{G_m}{C_{eq}} \tag{3.14}$$

where C_{eq} can be calculated as:

$$C_{eq} = \frac{C_{LOAD}}{\beta} = C_L + C_S + \frac{C_L C_S}{C_F}. \tag{3.15}$$

These equations show how also (3.11) turns into a condition on the gain-bandwidth product of the loop gain.

Note that the expressions above were derived on clock phase ϕ_2; what happens on phase ϕ_1? When designing SC circuits it is important to consider both clock phases with respect to speed and stability ([1]). In clock phase ϕ_1 the total load capacitance at the output is

$$C_{LOAD,1} \approx C_S \tag{3.16}$$

while the feedback factor is $\beta_1 = 1$; therefore $C_{eq,1}$ is smaller and $\omega_{G,1}$ is larger. Then the speed is lowest in phase ϕ_2 during which the circuit has been correctly analysed. This fact is most likely to occur especially with large gain factors.

3.3 Equation Based Design Methodology

The amplifier can be considered as composed by three main blocks: a bias generator, the amplifier itself and the common mode controller. Applying a band gap voltage across the off chip reference resistor, the bias generator create a temperature and technology process independent current. This current is replicated and adjusted for the different stages of the converter. Common mode controller operation principle will be explained in section 3.3.5. Next sections deal with the amplifier, on which we concentrate in this thesis, since the non ideal frequency behavior of this block determines the non ideal behavior of the whole stage.

3.3.1 Architecture Selection

We need an op amp with an high small signal low frequency gain together with a high gain-bandwidth product; in CMOS technology this is obtainable through a cascode topology by increasing the output resistance without modifying input stage g_m. Actually, a two stage amplifier is also a good choice in order to increase gain; however in this case it is impracticable since we also want to diminish power consumption and to avoid compensation problems. In fact, in a cascode amplifier the only high impedance node is the output one, responsible, as explained later, of the dominant pole; therefore a capacitive load accurately chosen provides compensation for the stage. Moreover, a cascode amplifier is able to give a larger gain than a two-stage architecture, with transistors properly sized. A folded cascode has been eventually chosen, since folded cascode amplifiers offer the following advantages over simple telescopic cascode amplifiers:

- input common mode dynamic range and output swing are independent and both high;
- there are less stacked transistors and the stage is more suitable for low supply voltage application.

Thus, the folded cascode configuration is particularly suitable for achieving wide and stable closed-loop bandwidths with large capacitive load, such as required in high frequency SC circuits.

In addition, the compensation in this circuit is achieved without coupling high-frequency noise from the power supply to the output, as for a multistage op amp. Hence, the high frequency PSRR can be high. A disadvantage remains the reduced output voltage swing due to the many cascaded devices and to the fact that the added devices contribute to the output noise and hence reduce the dynamic range. To keep low noise the input differential pair is realized by PMOS transistors. The amplifier is mainly composed of three parts: the core (differential pair and folding branches), the biasing circuit for the cascode mirrors and the mirror controlled.

3.3.2 The Folded Cascode Operational Amplifier

Fig. 3.3 represents the schematic of the circuit. V_{i1} and V_{i2} mark the input nodes while V_{o1} and V_{o1} mark the output nodes. M_1 and M_2 are the input differential pair, while M_5 and M_6 are the folded transistors (in common gate

configuration). Transistors M_3, M_4, M_{11}, M_{12} provide bias currents for the stage, while M_7, M_8, M_9 and M_{10} forms the cascode active loads for high output resistance. V_{B1}, V_{B2}, V_{B3}, V_{B4} and V_{POL} are all DC bias voltages and they have to be provided by a bias circuit, adequately sized as will be explained in section 3.3.4. V_{cmc} comes from the common mode control circuit, analyzed in section 3.3.5, and allows to regulate the output common mode by controlling the current flowing through M_{11}. In this section the circuit will be completely studied using simplified models, in order to get the equations representing its behavior, regarding:

- bias point;
- AC performances, i.e. gain, poles, bandwidth;
- large signal performances, i.e. output swing and common mode input swing;
- noise and offset considerations.

Bias Point

To correctly bias the amplifier are necessary two currents: one sinked from the biasing circuit and one sinked from the mirror controlled (both are provided by the bias generator block). As a first approach to the problem, we make these simplifying assumptions:

- the bias network can be separated from the amplifier and seen as a couple of ideal current generators, one providing I_{B1}, which flows in M_1 and another providing I_{B2}, which flows in M_9;
- the effect of common mode controller can be neglected with respect to C_{LOAD};
- symmetric devices are perfectly matched and so the variables' number reduces to a half;[4]

[4] Actually it is not correct to neglect matching errors, especially for a fully differential amplifier. However diminishing matching error sensibility simply limits minimum geometry for input differential pair transistors; this fact will be included in our constraints later as a static bound. See section 3.3.2.

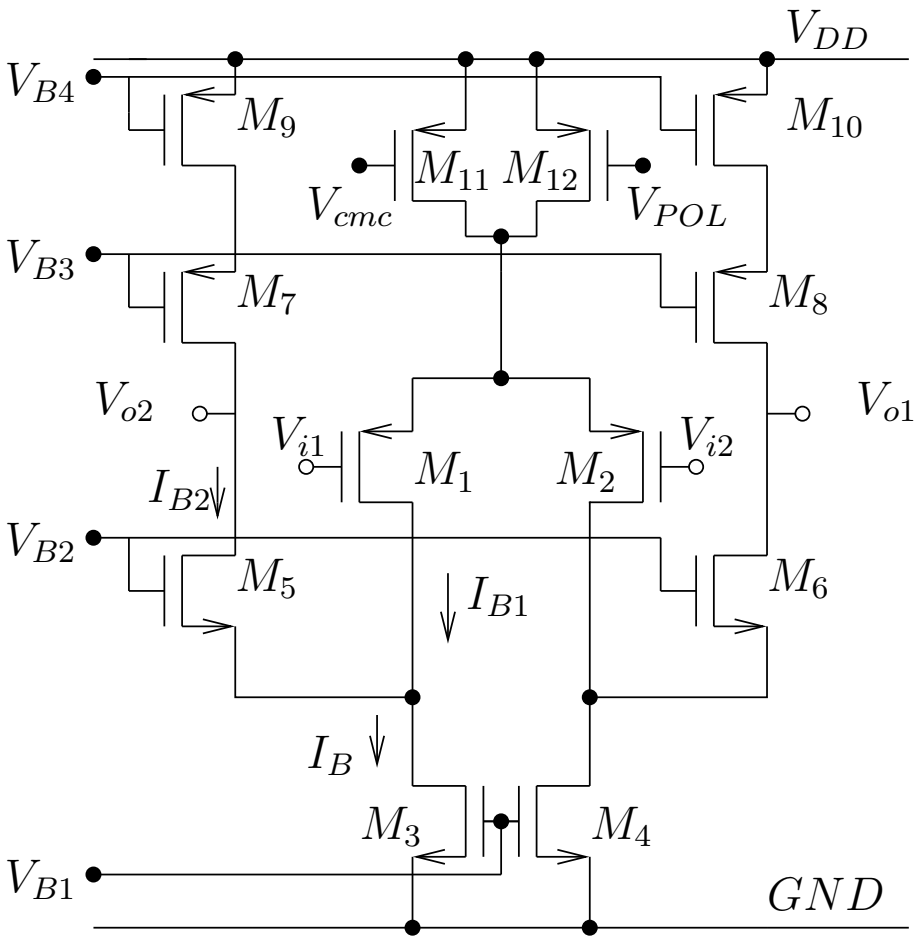


Figure 3.3: Schematic of the folded cascode fully differential op amp.

- body effect is ignored, except on including it in threshold voltage estimations;
- channel length modulation effect is also ignored to simplify calculations.

We assume that drain current for transistors M_{12}, M_1, M_3, M_5, M_7, M_9, can be expressed as a function of W, L, V_{GS} using the well-known parabolic equation. This is only to give an insight into circuit behavior from a designer perspective. Later graph will be generated using more sophisticated models including effects as mobility degradation and velocity saturation, relevant in submicron processes. Therefore this equation holds:

$$I_D = \frac{\mu_n C_{ox}}{2} \frac{W}{L} (V_{GS} - Vt)^2 \tag{3.17}$$

where C_{ox} is the gate oxide capacitance per unit area, μ_n is the average electron mobility in the channel, V_t is the threshold voltage. All above parameters depend on technological process and can be estimated from device models and simulations. Let I_B be the drain current of M_3, V_{DD} the power supply voltage and $V_{ov} = V_{GS} - V_t$ the *overdrive* of a transistor, the following relations stand for currents and voltages:

$$I_B = I_{B1} + I_{B2} \tag{3.18}$$

$$|V_{ov12}| + |V_{ov1}| + V_{ov3} - V_{DD} < 0 \tag{3.19}$$

$$|V_{ov9}| + |V_{ov7}| + V_{ov5} + V_{ov3} - V_{DD} < 0. \tag{3.20}$$

Equations (3.19) and (3.20) must be stated to operate all transistors in their active region.

Power consumption can be easily evaluated as $2 \cdot V_{DD} \cdot I_B$.

AC Performances

Basing on fully differential amplifier symmetry properties, a differential-mode (DM) half equivalent circuit of the op amp is represented in Fig. 3.4.

The voltage gain at low frequencies is $A_{v0} = -g_{m1} R_o$ where g_{m1} coincides with the overall stage transconductance and R_o is the output resistance. By inspection,

$$R_o = (g_{m7} r_{ds7} r_{ds9}) \| [g_{m5} r_{ds5} (r_{ds1} \| r_{ds3})] \tag{3.21}$$

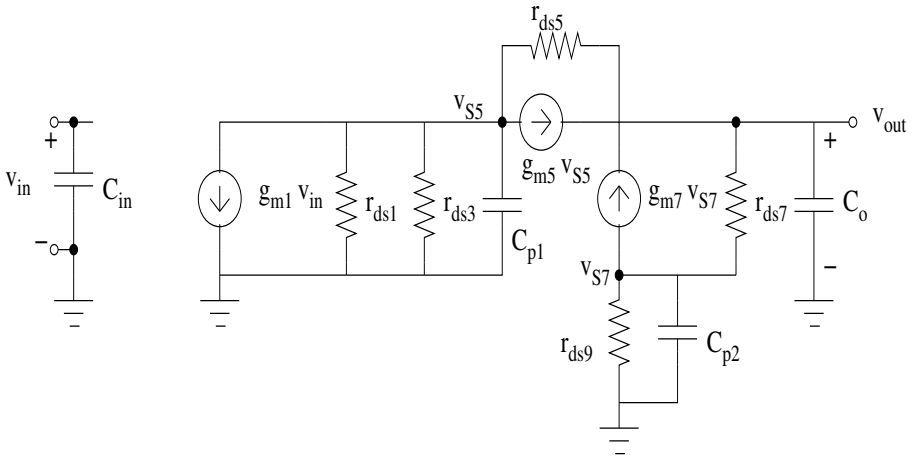


Figure 3.4: High frequency equivalent half circuit of the folded cascode amplifier.

Here, for each transistor, g_m is related to bias current I_D, gate overdrive and aspect ratio (W/L) through the equations below, valid for an NMOS transistor:

$$g_m = K_n \frac{W}{L}(V_{GS} - V_t) \tag{3.22}$$

and

$$g_m = \sqrt{2K_n \frac{W}{L} I_D} \tag{3.23}$$

where $K_n = \mu_n C_{ox}$ is the *transconductance parameter*. Output resistance is roughly estimated, for a n-channel device, like this:

$$r_{ds} = \frac{1}{\lambda I_D}. \tag{3.24}$$

Parameter λ is given by

$$\lambda = \frac{1}{L_{eff}} \left(\frac{\mathrm{d}X_d}{\mathrm{d}V_{DS}} \right) \tag{3.25}$$

where X_d is the width of the depletion layer between the physical pinch-off point in the channel at the drain end in saturation region, and the drain region itself; $L_{eff} = L - X_d$ is the *effective* channel length. Since direct calculation of λ from the device structure is quite difficult, in general it is estimated from experimental data. In our case these data coincide with *BSIM3v3* model parameters at our disposal. However, these models become

more and more complicated and λ, even if present in previous SPICE models (level 1), becomes a function of V_{DS} difficult to treat in hand calculations. The only thing the designer can consider is that λ is inversely proportional to channel length. We understand now the necessity of a bit more accurate models since op amp gain is strictly affected from output resistance, which is strictly dependent on λ. An equation will be introduced in section 4.3 which includes, without complicating formulas excessively, λ's dependence from V_{DS}.

In order to model the circuit high frequency behavior, we have considered for each transistor the presence of two principal intrinsic capacitances: the *Gate-to-Drain Capacitance* and the *Gate-to-Source Capacitance* C_{gs}; capacitances towards substrate are negligible. C_{gd} is only due to the overlap of the gate and the drain diffusion when MOS is in saturation region, since no channel is present on the drain side because of pinch-off. This capacitance can be estimated like this:

$$C_{gd} = W L_{ov} C_{ox} \tag{3.26}$$

where L_{ov} is the lateral diffusion length of both source and drain. The *Gate-to-Source Capacitance* C_{gs} has two components: the gate-to-source thin-oxide overlap capacitance and the gate-to-channel capacitance. In the saturation region the latter prevails and so we can assume that coincides with all the C_{gs}, expressed by:

$$C_{gs} = \frac{2}{3} W L C_{ox}. \tag{3.27}$$

A more accurate analysis of circuit in Fig. 3.4 using node equations is present in [24]. We are not interested in rigorously determining circuit poles (which is time consuming but can be done easily through Grabel's method for estimating characteristic polynomial coefficients, as shown in [25]). We need only to estimate the dominant pole and to grant its existence. Phase margin (PM) will also be estimated. We know that cascode op amps have high open-loop output resistance and the dominant pole is associated with the output node. Giving the determinant role of the total loading capacitance, it is impossible to furnish a reasonable estimation of the dominant pole without considering feedback network. Intrinsic capacitances have been lumped into three capacitors that are not independent. Miller effect due to C_{gd1} has been neglected since it is less important when dealing with cascode amplifier. Capacitances in Fig. 3.4 are related to the device capacitances by the following approximated equation:

$$C_{in} = C_{gs1} \tag{3.28}$$

$$C_o = C_{gd5} + C_{gd7} + C_{LOAD} \tag{3.29}$$

$$C_{p1} = C_{gd3} + C_{gs5} \approx C_{gs5} \tag{3.30}$$

$$C_{p2} = C_{gd9} + C_{gs7} \approx C_{gs7} \tag{3.31}$$

where gate-to channel capacitances prevail on gate-to-drain capacitances. The dominant pole is set by the zero value time constant for C_o, which is computed with C_{p1} and C_{p2} open. The nondominant poles can be approximated using the short circuit time constant for C_{p1} and C_{p2}, computed with C_o shorted so that C_{p1} and C_{p2} become independent. We obtain:

$$p_H \approx -\frac{1}{R_o C_o} \tag{3.32}$$

while

$$p_1 \approx -\frac{1}{\left(r_{ds3} \| r_{ds1} \| \frac{1}{g_{m5}}\right) C_{p1}} \tag{3.33}$$

and

$$p_2 \approx -\frac{1}{\left(r_{ds9} \| \frac{1}{g_{m7}}\right) C_{p2}}. \tag{3.34}$$

For our purposes it is sufficient to grant that p_1 and p_2 are distant one decade at least from p_H, the dominant pole. Since there is a zero at approximately $-g_{m7}/C_{p2}$, the effect of p_2 is cancelled out ([24]). Under this hypothesis, the phase margin depends on reciprocal positions between $|p_1|$ and the unity gain frequency of the amplifier in the open loop configuration, ω_0, given by

$$\omega_0 = \frac{g_{m1}}{C_o} \approx \frac{g_{m1}}{C_{LOAD}}. \tag{3.35}$$

Therefore PM is a strong function of the parasitic capacitance at the cascode node and can be evaluated as

$$PM = \arctan \frac{|p_1|}{\omega_0}. \tag{3.36}$$

In general ([21]) $|p_1| = 3\omega_0$ gives a margin greater than 60° respected in order to assure stability and a good frequency response.

Another important point is the role of the input capacitance C_{in}, which was neglected in the preliminary analysis made in section 3.2.2. In fact when studying return ratio unity gain frequency, this capacitance comes in parallel with C_S and contributes to determine the value of C_{LOAD} and β, which could be relevant. For instance, as regards C_{LOAD}, we obtain:

$$C_{LOAD} = C_L + \frac{C_F(C_S + C_{in})}{C_F + C_S + C_{in}}. \tag{3.37}$$

Large Signal Behavior

The output swing is defined to be the range of output voltages for which all transistors operate in the active region so that the gain is approximately constant. By inspection of Fig. 3.3 we obtain:

$$V_{o1} \leq V_{B3} + |V_{t8}| = V_{DD} - |V_{DS10}| - |V_{ov8}| \tag{3.38}$$

$$V_{o1} \geq V_{DS4} + V_{ov6}. \tag{3.39}$$

To maximize the output swing, transistors M_8 and M_6 should be biased at the verge of triode region, so that the output voltage can swing within two overdrives of each supply. In practice, this is not possible. MOS transistors display an indistinct transition from the triode region to the saturation region and to obtain high incremental drain impedances, r_{ds8} and r_{ds4}, they must be biased by a few hundred mVs (100 or 200 mV) into the saturation region. V_{DS4} and $|V_{DS10}|$ can be regulated by V_{B3} and V_{B2} as shown in section 3.3.4.

The common-mode input range is the range of dc common-mode input voltages for which all transistors of the op amp operate in the active region. The common mode range limits can be written as:

$$V_{IC} < V_{DD} - |V_{ov12}| - |V_{ov1}| - |V_{t1}| \tag{3.40}$$

$$V_{IC} > V_{DS4} - |V_{t1}|\,. \tag{3.41}$$

However, the correct working of the amplifier is granted for an input common mode of 1.25 V with an estimated variation of -500 mV and +200 mV [19].

Noise and Offset Considerations

The input offset voltage of a fully differential amplifier is defined as the differential input voltage for which the differential output voltage is zero. In fully differential configuration only the random offset resulting from mismatches is relevant. A straightforward analysis for the offset voltage of our circuit, analogous to that found for the first stage of the two stage op amp in [20], gives the following result:

$$V_{OS} = \Delta V_{t(1-2)} + \Delta V_{t(3-4)} \left(\frac{g_{m3}}{g_{m1}}\right) + \Delta V_{t(9-10)} \left(\frac{g_{m9}}{g_{m1}}\right) + \\ + \frac{V_{ov(1-2)}}{2} \left[\alpha \frac{\Delta\rho_{(3-4)}}{\rho_{(3-4)}} + (\alpha - 1)\frac{\Delta\rho_{(9-10)}}{\rho_{(9-10)}} - \frac{\Delta\rho_{(1-2)}}{\rho_{(1-2)}}\right] \tag{3.42}$$

where ρ is the aspect ratio W/L of each transistor and $\alpha = I_B/I_{B1}$.[5] The first term represents the threshold mismatch of the input pair; the second and the third are the threshold mismatches of the current-mirror-load devices (M_3, M_4, M_9, M_{10}) and are minimized by choosing the W/L ratio of the load devices so that their overdrive is large compared to that of the input transistors. For this reason a longer channel length for M_3, M_4, M_9, M_{10} reduces the random input offset voltage. The forth term represents the effects of mismatches of various ρs in the input transistors and loads and is generally minimized by operating the input transistors at low values of overdrive, typically on the order of 50 to 200 mV.

Similar considerations can be done about noise. In section 3.2.1 we neglected the noise from the op amp. In a more detailed analysis this component should be taken into account since it may be as large as or even larger than the switch noise. Noise generated by each device can be represented symbolically by an equivalent voltage source v_n connected to its gate. The impedance at the sources of cascode transistors is low and the noise contributions from the cascode transistors can be neglected, as is suggested also in [1]. Since all noise sources are uncorrelated, by examining low-frequency equivalent circuit, the equivalent input noise voltage mean-square value can be estimated like this:

$$\overline{v_{nd}^2} = \overline{v_{n1}^2} + \overline{v_{n2}^2} + \left(\frac{g_{m3}}{g_{m1}}\right)^2 (\overline{v_{n3}^2} + \overline{v_{n4}^2}) + \left(\frac{g_{m9}}{g_{m1}}\right)^2 (\overline{v_{n9}^2} + \overline{v_{n10}^2}). \tag{3.43}$$

[5] Even if transistors in each pair must be matched, only mismatch in the differential pair and in the load mirrors has been taken into account in order to compute input offset voltage: cascode transistors have negligible influence in determining errors on currents.

Hence, to minimize $\overline{v_{nd}^2}$, v_{n1} and v_{n2} should be small and g_{m3}, $g_{m9} \ll g_{m1}$. The former requires ([20],[21]) that the area $(W \cdot L)$ and transconductance g_m of M_1 and M_2 be large. To obtain large g_m, the bias current and (W/L) ratio should be large; this, however, requires large devices and high power dissipation. Anyway, in our case it is not so important to increase the input g_m since, in this way, even if we minimize v_{n1} and v_{n2} power spectral density, we increase the circuit equivalent noise bandwidth, so that the total noise power is less dependent on g_m of M_1 and M_2 [1]. Therefore, it is more useful to reduce the noise contribution of the load devices. This can be done, as section 3.43 shows, by making their transconductances as *small* as their biasing conditions permit. This can be achieved by increasing their length L by a factor of two or more. Thus, assuming that the areas of the input and load devices are given, the W/L ratios of the input devices should be as *large*, while those of the load devices as *small* as other considerations permit. Also, it has been found experimentally that the root mean square (RMS) equivalent $1/f$ noise is about three times larger for an n-channel device than for a p-channel one. It is hence advantageous to use PMOS devices for the differential stage.

3.3.3 Principal Design Steps

In this section we show how all the equations listed above are sufficient to derive from specifications the sizes of the transistors in the circuit, following an *equation based design methodology*. All reasoning carried out in this section constitutes the starting point to extract an efficient algorithm for successive graph scheduling (see section 4.7).

The op amp parameters have been already specified in Table 3.1: other important design criteria include dynamic range, output impedance, area occupied on the chip, slew rate common-mode rejection, dc power consumption. The specific steps followed in the design depend on the application, the circuit chosen and the relative importance of the various criteria. In some situations, when trying to respect a specification, it is possible to conflict with others; in some other case other specifications are likely to be automatically respected. For instance, gain, bandwidth and noise considerations lead to bias currents so high that no relevant slew rate problems are present for typical input step sizes. It will be assumed that the transconductance factor K_n for a NMOS is 198 μA/V^2, K_p for a PMOS is 66 μA/V^2, C_{ox} is 6.35 mF/m^2 and the threshold voltages are assumed to be $V_{tn} = 0.36$ V and $V_{tp} = -0.5$ V.

Short-channel effects become important in MOS transistors at channel lengths of about 1 μm or less and require modifications to the square-law model given previously; however many of the MOS transistors in an analog circuit can be deliberately designed to have channel lengths larger than the minimum and may be well approximated by the square-law model. This fact occurs also because noise and matching considerations require larger transistor areas $W \cdot L$ even if this leads to larger parasitic capacitances and smaller values of the transition frequency for the transistors, thus deteriorating high frequency performances. In our case following the conclusions in section 3.3.2, the smallest channel length $L_1 = 0.25\,\mu$m (approximately two times the minimum allowed length) has been selected for the input transistors M_1 and M_2, while the larger one $L_3 = L_9 = 0.7\mu\text{m} \approx 3L_1$ has been selected for the transistors in the current mirrors in order to increase output resistance and to diminish their contributes to input offset voltage and input referred noise. For M_5 and M_7 a smaller length has been selected: $L_5 = L_7 = 0.5\mu\text{m} \approx 2L_1$. Anyway, we deal with submicron lengths and this means that our calculations will be affected by errors, sometimes on the order of 40 percent, especially in evaluating parameters like R_o, since we have not accurate values for λ available. As a consequence, real sizing can be done only with the help of simulations.

At first the differential stage has been sized. Considering that ([20]) input stage characteristic remains linear if

$$|v_{id}| \leq \sqrt{2} \cdot |V_{ov1}|\,,$$

remembering that $|V_{ov1}|$ cannot be done too large because of matching errors (see section 3.3.2) and given the most typical values used in similar applications $|V_{ov1}| = 200$ mV has been used. As for g_{m1} we must notice that short channel effects cause degradations in the output resistance of the amplifier and a minimum small signal gain of 55 dB cannot be reached, unless g_{m1} is fixed at least at 150 m$\mho$, that is two times the minimum allowed value, present in Table 3.1. This conclusion derives from some tests performed using the transistors of the design kit and estimating their output resistances with the simulator. Introducing these values in equations (3.22) and (3.17), adequately adapted to PMOS transistors, we obtain:

$$W_1 = 2950\,\mu m$$

and

$$I_{B1} = 15.6\,mA.$$

Now from equations (3.28), (3.27), (3.37) and (3.35) we can estimate:

$$C_{in} = 3.1\,\text{pF},$$

$$C_{LOAD} = 7.6\,\text{pF},$$

and

$$\omega_0 = 19.7\,\text{GHz}.$$

From condition:

$$\frac{g_{m5}}{C_{p1}} = \frac{3\,K_n V_{ov5}}{2\,C_{ox}\,L_5^2} = 2\omega_0$$

to obtain $PM \approx 60°$we find:

$$V_{ov5} = 200\,\text{mV}.$$

At this point, in order to simplify the design process, we make some considerations based on symmetry:

$$V_{ov3} = |V_{ov9}|$$

and,

$$V_{ov5} = |V_{ov7}|\,.$$

We must point out that they are totally arbitrary and they are used only in the absence of other conditions. To minimize noise factor given by M_9 and M_3 we should select a large g_m for these transistors. By manipulating (3.17) and (3.22) the well-known relation is obtained:

$$\frac{g_m}{I_D} = \frac{2}{V_{ov}} \tag{3.44}$$

that we can apply to M_3. If g_{m3} has to be smaller than g_{m1} but the drain current $I_{D3} = I_B$ has to be larger than $I_{D1} = I_{B1}$, then V_{ov3} has to be much larger than V_{ov1} [6]. However, if V_{ov3} is too large, output swing decreases; as a compromise we state:

$$V_{ov3} = V_{ov9} = 300\,\text{mV}.$$

These observations point out an important *trade-off between the input referred noise and output voltage swing* for the cascode architecture, shown also in [12]: the contribution of M_3, M_4, M_9 and M_{10} to the input-referred

[6] This is also a good choice to diminish mismatch errors.

rms noise increases if their g_m increases, and from (3.44) if their V_{ov} is minimized to allow a large output swing.

g_{m9} can be easily done equal to $0.5 \cdot g_{m1}$. Once more from (3.22) and (3.17) we obtain:

$$W_9 = 2750\,\mu m$$

and

$$I_{B2} = 11.7\,mA.$$

Then from (3.18) and (3.17) we find:

$$I_B = 27.3\,mA$$

and

$$W_3 = 2100\,\mu m.$$

Now, using once more (3.17) M_5 and M_7 are sized:

$$W_5 = 1450\,\mu m,$$

$$W_7 = 4400\,\mu m.$$

The source voltage of M_1 is:

$$V_{S1} = V_{IC} + |V_{ov1}| + |V_{tp}| = 1.95\,\mathrm{V}$$

and thus $|V_{DS12}| = 0.55\,\mathrm{V}$. To allow a 200 mV margin, as stated in section 3.3.2, $|V_{ov12}| = 300\,mV$ and $L_{12} = 0.7\,\mu m$ can be introduced into (3.17), which gives:

$$W_{12} = 2600\,\mu m.$$

Using the model in section 4.3 a gain of 56 dB has been estimated. Notice that no care has been taken of power consumption.

We conclude by selecting the values for the bias voltages. By allowing V_{DS3} and $|V_{DS9}|$ to be 100 mV larger than their overdrives, as exposed in section 3.3.2 we obtain:

$$V_{B1} = V_{ov3} + V_{tn} = 0.66\,\mathrm{V},$$

$$V_{B2} = V_{DS3} + V_{ov5} + V_{tn} = 0.96\,\mathrm{V},$$

$$V_{B3} = V_{DD} + V_{DS9} + V_{ov7} + V_{tp} = 1.4\,\mathrm{V},$$

$$V_{B4} = V_{DD} + V_{ov9} + V_{tp} = 1.7\,\mathrm{V}.$$

3.3.4 The Biasing Circuit

This circuit must provide the bias voltages and currents calculated before. The schematic is represented in Fig. 3.5 and can be sized by means of the following considerations. It is not difficult to size the bias network, once the amplifier itself has been sized; therefore we shall only expose the steps followed, without performing all calculations. Of course once more a certain number of simulations are necessary to verify the amplifier behavior in conjunction with the biasing network. This is why the hypothesis that the circuit could be sized as separate blocks is not rigorously valid, although this is the procedure followed by a designer.

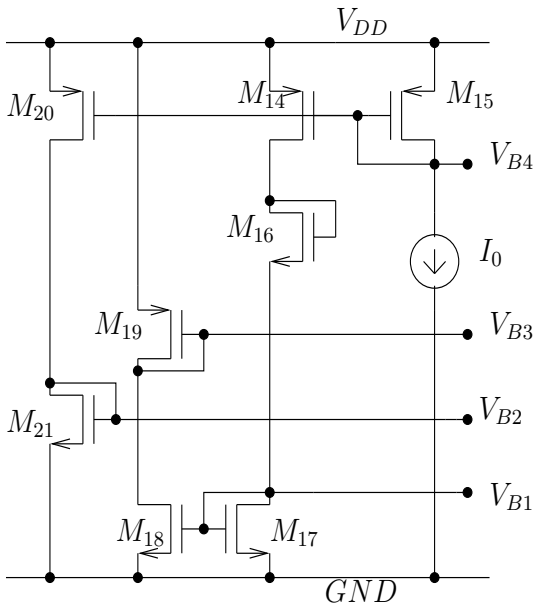


Figure 3.5: Biasing circuit for the folded cascode op amp.

Current I_0 in M_{15} is known since it comes from the band gap reference voltage, L can be set for all transistors equal to 0.7 μm like for M_9 and M_3. Using relations valid for current mirrors we get:

$$(W/L)_{15} = \frac{I_0}{I_{B2}}(W/L)_9 \tag{3.45}$$

and

$$(W/L)_{17} = \frac{I_{D17}}{I_B}(W/L)_3 \tag{3.46}$$

when I_{D17} is the drain current of M_{17}.

One choice could be to use the same current in all the branches, thus obtaining:

$$(W/L)_{14} = (W/L)_{20} = (W/L)_{15},$$
$$(W/L)_{17} = (W/L)_{18} = \frac{I_0}{I_B}(W/L)_3.$$

Anyway, even if we decide not to maintain the same currents, from equations (3.45) and (3.46) derives a relation between the aspect ratio of M_{14} and M_{17} as a function of the current ratio $\chi = I_{B1}/I_{B2}$. In our case, with all Ls equal, we obtain:

$$\frac{W_{14}}{W_{17}} \cdot \frac{W_3}{W_9} = \chi + 1. \tag{3.47}$$

M_{19} and M_{21} can be sized from equation (3.17) knowing that

$$V_{GS21} = V_{B2}$$

and

$$V_{GS19} = V_{B3} - V_{DD}.$$

Transistor M_{16} can be sized by selecting $V_{GS16} = V_{B4} - V_{B1}$: this fixes the drain voltage of M_{14} to be equal to that of M_{15} and makes the current mirror composed by M_{14} and M_{15} more accurate.

3.3.5 Common Mode Controller

This circuit works in sampled time domain. A switched capacitor common mode feedback circuit has been used because it has some advantages with respect to other techniques since it imposes less limitations to the op amp output swing and allows to avoid resistive output loading of the op amp. Sensing the output common mode value of the amplifier and comparing it with a reference voltage, it stores the voltage result in a capacitor. The stored value is utilized by the mirror controlled in the amplifier to fix its output common mode value. The circuit is implemented by 8 pass gate switches sized to minimize the charge injection and two capacitors.

A simplified version of the circuit is represented in Fig. 3.6. When ϕ_1 is high total charge on capacitors $C_X = C_Y$ is $Q_1 = (C_X + C_Y)(V_{POL} - V_{REF})$. When ϕ_2 is high the total charge is $Q_2 = (V_{cmc} - V_{o1})C_X + (V_{cmc} - V_{o2})C_Y$. By imposing charge conservation, i.e. $Q_1 = Q_2$ we obtain:

$$V_{cmc} = (V_{oc} - V_{REF}) + V_{POL} \tag{3.48}$$

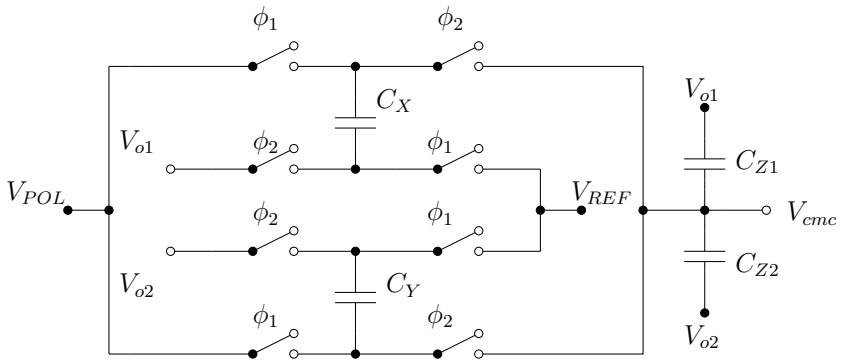


Figure 3.6: Simplified diagram of the SC common mode controller circuit.

where $V_{oc} = \frac{V_{o1}+V_{o2}}{2}$ is the common mode output voltage. The common mode loop is closed by the amplifier through the mirror controlled and V_{cmc} is the gate voltage of M_{11}, completely matched to M_{12}. It can be demonstrated ([20]) that the loop gain of the common mode feedback loop is high (thanks to the presence of the amplifier) and this means that V_{oc}, for a sort of virtual short circuit, becomes equal to V_{REF}, fixed at 1.25 V. In this case V_{cmc} becomes equal to V_{POL}, that is that voltage necessary to bias transistors in the mirror controlled in order to grant a common mode output voltage of 1.25 V.

In order to understand the working principle of the common mode feedback circuit, let us suppose that V_{oc} rises, then V_{cmc}, from (3.48), also rises and current in each input transistor I_{B1} decreases. Since I_B remains fixed by V_{B1}, I_{D6} and I_{D5} increase and consequently their drain voltages, that is the output nodes, are lowered, compensating the initial effect on V_{oc}.

Note that in the analysis made above, equation (3.48) was obtained neglecting the capacitors C_{Z1} and C_{Z2}, which are not switched capacitors since their connections do not change in the two different phases.

3.3.6 Mirror Controlled

Last block included in the amplifier is the mirror which provides biasing current to transistors M_1 and M_2. In Fig. 3.7 the schematic view of the subcircuit is shown.

I_{bias1} is the input bias current [7], which is mirrored through two match-

[7] It derives from a band gap voltage reference.

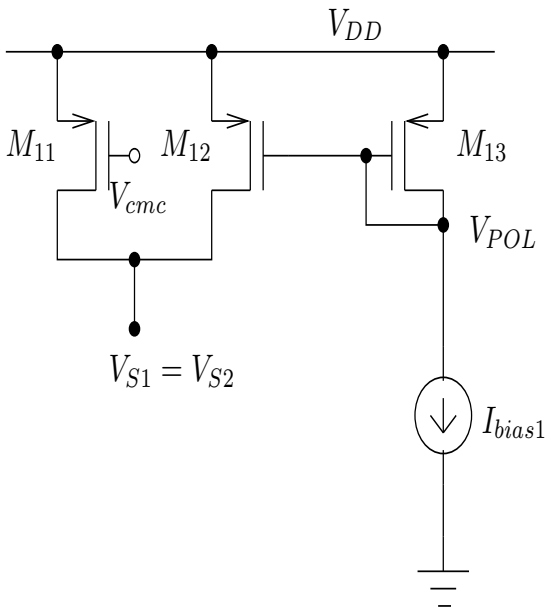


Figure 3.7: Schematic of the mirror controlled.

ing transistors M_{11} and M_{12}. Only one half of the current is controlled by the common mode feedback circuit through the voltage V_{cmc}. In fact, the remaining part of the current is granted by transistor M_{12} with its gate voltage fixed to V_{POL}. The working principle of common mode control has been shown in section 3.3.6.

3.4 Conclusions

In this chapter all the preliminary study has been performed to obtain those performance constraints we shall exploit in next chapter in order to characterize the amplifier. We use the simulator to approximate the evaluation function which maps the input space of circuit configurations into the output space of performance figures. In order to increase the efficiency of the simulations and to focus exploration on a more limited and interesting design space, configurations must not be totally random, but they should respect these constraints.

Chapter 4

The Architectural Space Characterization

With the methodology exploited in this thesis two forms of exploration are possible. One option is simply to explore trade-offs among performance specifications by examining performance variations across the range of circuits resulting from simulations. Another style of exploration is *threshold hunting*, in which we seek to determine the actual surface that separates feasible and infeasible designs. This is the approach followed also by [5]. This chapter describes the procedure followed to perform the characterization of the architectural design space for the folded cascode amplifier.

4.1 Introduction

The development of an equation based procedure in chapter 3 provided a quick and effective mechanism for directly estimating the MOS circuit parameters (sizes, bias currents and voltages) of the op amp from the performance requirements. In this phase the evaluation function $\phi(.)$ used to get performances is made up of a system of equations derived from a very simplified model for the MOS device, that one which is in the mind of an analog base-band IC designer. However, circuits sized through this procedure are unfaithful because of the lack in accuracy of the model equations. Actual design requires many simulator iteration and exploits empirical considerations and estimations that only an expert designer can intuitively conceive.

Up to now we have applied the traditional top-down design approach since we passed from circuit performances to transistor parameters. The in-

accuracy of the evaluation function in this procedure leads to a large number of simulator runs and design iterations. Another approach could be that one which uses a bottom-up exploration of the architectural space using the simulator as the evaluation function $\phi(.)$ and giving to the simulator a great number of vectors κs containing several (e.g. 10 000 or more) configuration parameters (samples of the $\mathcal{I}$ space) randomly chosen. As a result we should receive vectors ζs of performance parameters which samples the $\mathcal{O}$ output space. Performance data are in this case faithful since they derive, in an experimental fashion, from the simulator. In a certain sense to build an analog synthesis tool is to find methodologies able to solve complex optimization problems. Analog synthesis tools have traditionally traded quality for speed, substituting simplified circuit evaluation methods for full simulation in order to accelerate the numerical search for solution candidates [2]. As a result these tools have failed to migrate into mainstream use primarily because of difficulties in reconciling the simplified models required for synthesis with the industrial-strength simulation environments required for validation. We think that for synthesis to be practical, it is essential to synthesize a circuit using the same simulation environment created to validate the circuit.

However some questions arise. Does a random configuration of input parameters really translate into a physically reasonable circuit? If we already know we are not interested in an amplifier with a gain under 55 dB, is there a method to exclude simulations of parameters which do not respect this constraints so that we can limit the sample space and the simulation time? The design equations already obtained and all the considerations made in preceding sections come and help us. They highlight the principal factors affecting the performance specifications, which can make it very easy to reduce the sample space on the bases of the objectives we want to achieve in the design. As a consequence, we propose a strategy organized into two main phases (see Fig. 4.1):

1. **Preliminary Design**, during which simplified model equations are used to constrain the generation of random configurations. To easily encapsulate information contained in all the design constraints we have seen, an ACG has been extracted for the amplifier and it will be explained in section 4.6.

2. **Configuration Validation**, during which the random configurations respecting all the imposed constraints are given to the simulator and are validated on the bases of the extracted performances.

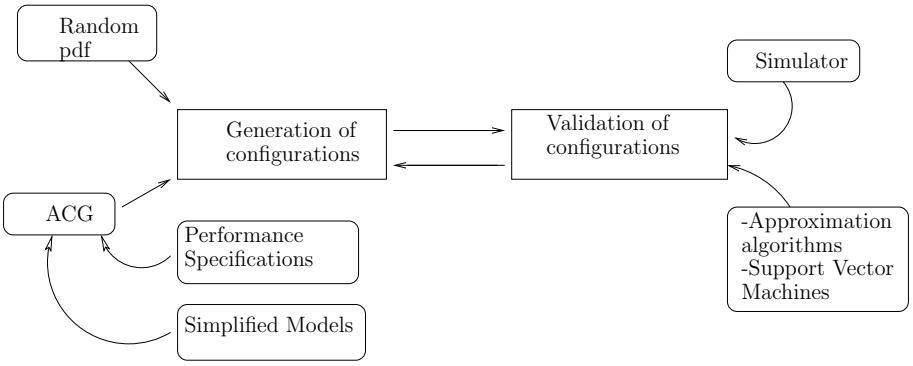


Figure 4.1: Main components of the two phase evaluation scheme.

In the preliminary design phase random probability density functions (pdfs) are also necessary as shown in the figure. The validation phase has to decide which set of performances are reachable from the particular architecture and to reject the unfeasible or unwanted regions. Results from simulations are classified and the feasible performance output space is approximated using statistical learning machinery and Support Vector Machines, as we shall describe later. In Fig. 4.1 is also represented the interaction between the two main blocks. In fact, on the bases of the results of the approximation the number of configurations to be generated can change and the process of generation stops when there are enough data to build a reasonable approximation of the feasible performance region.

4.2 Introduction to the ACG

As defined in section 1.3.3 an AGC is a bipartite undirected graph (Ξ, Ψ, Υ) where Ξ is the set of design variables, Ψ is the set of constraints on ξs and $\Upsilon \subseteq \Xi \times \Psi \cup \Psi \times \Xi$ is the set of edges that link design variable ξ_i to constraint ψ_j. The design variables in Ξ include circuit configuration parameters we can call *primary variables* and a set of secondary variables that make the graph easier to represent but are not required to specify a configuration (*ancillary variables*). In our case, primary variables include only Ws, Ls, Is, C_{LOAD}, which models the amplifier loading capacitance, and β, which refers to the feedback network transfer function, as exposed in sections 3.2.1, 3.2.2, 3.3.2; ancillary variables include the others, such as g_m, r_{ds} and so on. Clearly, ancillary variables can be expressed as a function of primary variables, but

the resulting constraints ψ_i would be much more complex to handle.

Graph has been intended in a hierarchical way: a first group of nodes refers to DC bias constraints, a second group refers to the AC gain of the amplifier, the third one is related to bandwidth. Obviously we use hierarchy only to better organize ideas: no priority is given at present to anyone node or group of nodes.

Simulink® *Stateflow* has been used as a graph editor. The graph is composed by more than 30 constraints and more than 40 variables. Only two fragments are shown in Figures 4.2 and 4.3.

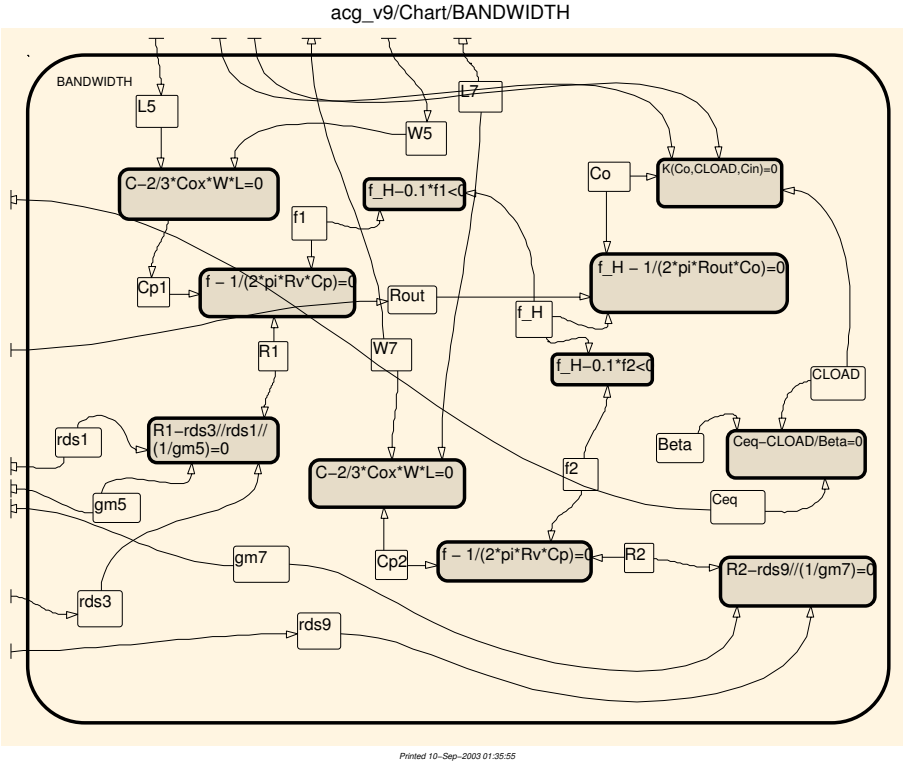


Figure 4.2: Analog constraint graph for bandwidth constraints.

All the constraints deriving from the circuit architecture in itself or from closed loop performances have been basically treated in chapter 3 and they will be listed once more later. Now we point out only those novel elements introduced while extracting the graph. The first one is the new model used to write these constraints, which is discussed in next section.

We pointed out that analytical models are not so accurate, thus it is

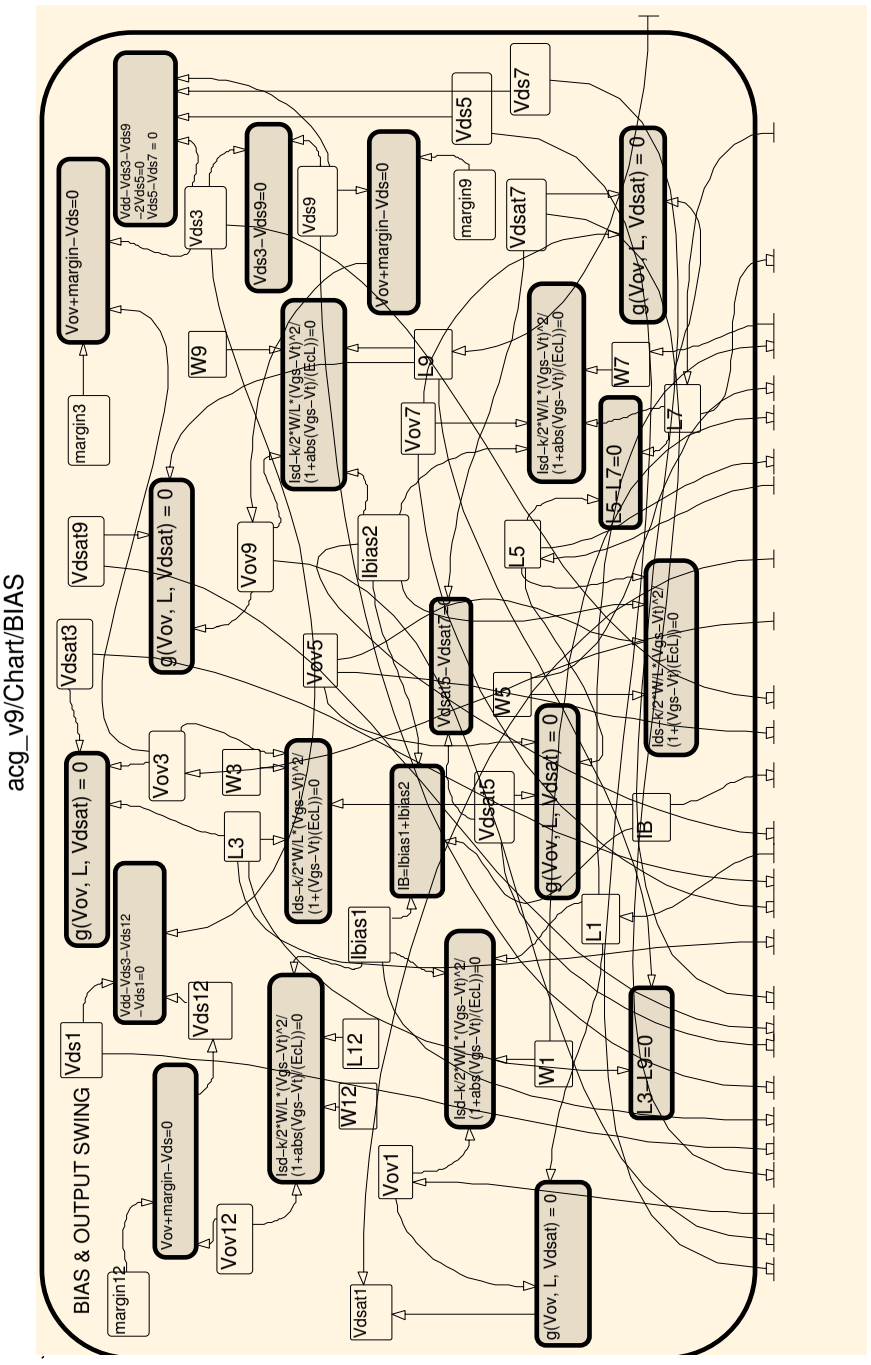


Figure 4.3: Analog constraint graph for bias constraints.

not advisable to impose too many constraints when generating the ACG. In fact, in this case we limit the architectural exploration space to a region which won't surely respect the performance specifications since the constraints were imposed through equations too far from real transistor behavior. On the contrary, if we allow more random generations of configurations and if the configurations generated are a relevant number, the probability of finding a good candidate increases.

The disadvantage is that increasing the necessary samples means increasing the number of simulations to perform and this is not always feasible especially for very complex analog cells. The harsh truth is that analog circuits are difficult and time-consuming to evaluate properly. Even a small cell requires a mix of ac, dc, and transient analyses to correctly validate. Especially in later phases of the design, when the designer has a better idea of the circuit behavior and non-idealities, the necessity rises to refine the platform and to bias the design space exploration towards some directions. At a certain level of the design some trade-offs becomes more important than before. In this cases it is not easy to reach particular transistor operating regions without the use of more accurate models. This becomes the critical points of any tool in the analog CAD arena. A more complicate model turns into more CPU computation time and does not guarantee necessarily better accuracy. This is why in modern design environments, there is enormous investments in simulators, device models, process characterization, and "cell sign-off" validation methodologies. Moreover, equation-based design tools require a great preparatory time to derive the circuit equations from complicated models. This typically requires a user who is a programmer, an analog designer and an expert intimate with the internal architecture of the tool. Some attempts have been reported of tools which incorporated complete BSIM device models, thus complicating solving also for dc operating point in an evolving circuit because the models cannot be inverted analytically. Equation-based synthesis tools use only minutes of CPU time but require the designer to spend months deriving, coding and testing equations and models. Even if many models are reported (see for example [26]) also capable to describe MOS behavior in almost all possible operation regions, it is necessary to test them with respect to the particular process in use. In next section a description of the model exploited is given.

4.3 Small Signal Model of the MOS Transistors

In order to perform MATLAB® calculations a more accurate model has been chosen based on equations derived from *BSIM3v3* model for short channel devices. The advantages of such a model consist in the fact that it takes into account some important effects, such as mobility degradation and velocity saturation, without complicating excessively mathematics behind formulas. Actually, it can be shown ([20]) that velocity-saturation effects are insignificant in hand calculations if:

$$(V_{GS} - V_t) < 0.1(\mathscr{E}_c L) \tag{4.1}$$

where $\mathscr{E}_c$ is the critical field value for velocity saturation. This situation is verified in most of our cases, but a problem rises for the designer especially in evaluating output resistances of transistors. First of all, for submicron processes it is difficult to give an accurate estimation of parameter λ. Then once you have estimated it according to certain criteria, you realize you are overestimating r_{ds} of the devices. This is not useful in the design process since we generally need conservative estimation when we state, for instance, that gain should be larger than a minimum value.

The model proposed by the researchers at the *University of California, Berkeley*, uses the equations of the BSIM3v3 model reported in [27], that one used by many device level simulators. This model has been adequately simplified for rough calculations so that it takes into account only a limited number of non ideal effects. The result is a set of equations very similar to those of the well known short channel model for MOS transistors present also in [20]. The difference is that parameters in formulas can be readily extracted by model files present in the design kit libraries for a particular process and the designer can extract them very quickly. All constraints obtained in section 3.3.2 are basically represented in the graph and they have all been expressed through these more accurate equations we are going to list.

In Table 4.1 are listed the parameters required in order to model each transistor, with their typical values for processes like ours.

Two additional parameters must be defined before discussing the expression of graph constraints. The first one is the *mobility degradation coefficient*:

[1] It refers to depth of source and drain doped regions.

Name	Significance	NMOS	PMOS
TOX	Gate Oxide Thikness	5.33 nm	5.53 nm
U0	Mobility (ideal case)	314.32 cm^2/Vs	103.22 cm^2/Vs
UA	Mobility Degradation	$-4.87 \cdot 10^{-10}$ m/V	$4.8 \cdot 10^{-10}$ m/V
VSAT	Saturation Velocity	$1.5 \cdot 10^5$ m/s	$2.08 \cdot 10^6$ m/s
VTH0	Threshold Voltage (ideal)	0.362 V	-0.506 V
PCLM	Channel Modulation	1.766	0.965
XJ	Junction Depth[1]	$1.7 \cdot 10^{-7}$ m	$1.7 \cdot 10^{-7}$ m

Table 4.1: Parameters similar to those extracted by model files.

$$u_d = \frac{UA}{t_{ox}}, \tag{4.2}$$

and the second is the already cited *critical electric field* for velocity saturation, which can be computed from the following equation:

$$\mathscr{E}_c = \frac{2v_{sat}}{U0}. \tag{4.3}$$

For process like ours $\mathscr{E}_c = 9.63 \cdot 10^6$ V/m for a NMOS transistor and $403.88 \cdot 10^6$ V/m for a PMOS transistor can be assumed.

In next sections, it will be shown how the above parameters can be used to model the transistors.

4.4 Expression of the Constraints Through the New Model

As for bias constraints we require something similar to that found in section 3.3.2. The equation equivalent to (3.17), which expresses drain current for transistors M_{12}, M_1, M_3, M_5, M_7, M_9 as a function of W, L, V_{GS}, written for a short-channel NMOS transistor is the following:

$$I_D = \mu_0 C_{ox} \frac{W}{2L} \left[\frac{(V_{GS} - Vt)^2}{1 + \left(u_d + \frac{1}{\mathscr{E}_c L} \right) (V_{GS} - V_t)} \right] \tag{4.4}$$

were all symbols are known except for μ_0 which stands for U0 and is the average electron mobility in the channel in the ideal case, when all factors

(such as vertical field, horizontal field, velocity saturation) are neglected. At present we don't include the effect of V_{DS} on bias since, given the small supply voltage value (2.5 V), V_{DS} for each transistor is on the order of hundreds of mVs. Therefore we suppose the product λV_{DS}, negligible with respect to unity; we are neglecting in saturation region the effect of channel length modulation, but we shall recover it when we deal with r_{ds}. Note that, according to the model, the parameter VTH0 should be used as an estimation of V_t.

Relations (3.19) and (3.20) can be rewritten using in place of V_{ov} for each transistor, $V_{DS(sat)}$, the minimum value of V_{DS} for which the transistor operates in the active region, that can be calculated like this:

$$V_{DS(sat)} = (V_{GS} - Vt) \left[\frac{1 + u_d(V_{GS} - Vt)}{1 + \left(u_d + \frac{1}{\mathscr{E}_c L} \right) (V_{GS} - V_t)} \right]. \tag{4.5}$$

It is shown in ([20]) that when velocity saturation is relevant $V_{DS(sat)}$ is less than the overdrive. However this substitution is not necessary since we are overestimating $V_{DS(sat)}$ and equations (3.19) and (3.20) remain valid even more so.

In all constraints, g_m has been expressed as a function of the other variables through the following equation:

$$g_m = \frac{I_D}{(V_{GS} - V_t)} \left[1 + \frac{1}{1 + \left(u_d + \frac{1}{\mathscr{E}_c L} \right) (V_{GS} - V_t)} \right] \tag{4.6}$$

where I_D is the drain current in saturation region calculated in (4.4). For the output resistance the following relation has been used, able to model dependence of r_{ds} from L and from I_{Dlong}, that is the drain current for a long channel device already shown in (3.17):

$$r_{ds} = \frac{\{(V_{DS} - V_{DSsat}) + [1 + u_d(V_{GS} - V_t)] (V_{GS} - V_t)\} L}{I_{Dlong} l P_{CLM} [1 + u_d(V_{GS} - V_t)]} \tag{4.7}$$

where we define $l = \sqrt{3 t_{ox} x_j}$. Note that (4.7) has been used only to give an estimation of the output resistance of the amplifier and of its low frequency gain so that, after having sized the circuit, we can have an initial validation of our choice.

4.5 Validation of the Model

The model we have used for the constraints is made up of all the above equations with the addition of another one for the MOS current in the triode region I_{Dlin}. We report that equation even if we do not use it for the graph:

$$I_{Dlin} = \mu_0 C_{ox} \frac{W}{L} \left(V_{GS} - V_t - \frac{V_D}{2} \right) \left[\frac{V_{DS}}{1 + u_d(V_{GS} - V_t) + \left(\frac{V_{DS}}{\mathscr{E}_c L} \right)} \right]. \quad (4.8)$$

First of all some simulations have been performed to verify if the error introduced by this simplified model is acceptable. The Cadence® CAD tools *Virtuoso Schematic* and *Analog Artist (Spectre)* has been used for schematic capture and for simulation [28]. In this phase a relative error within 50 % in estimating the parameters is acceptable; in fact, we do not use these equation to directly design the circuit but only to obtain reasonable configurations to simulate later. On the other hand, as we have already seen, the simple squared-law model gives errors of the order of 100 % and cannot model effectively the transistor output resistance. To have an idea about the accuracy of the model adopted two transistors have been chosen from the design kit at our disposal, an NMOS ($W = 1400\mu$m, $L = 0.5\mu$m) and a PMOS ($W = 7000\mu$m, $L = 0.5\mu$m). These dimensions have been chosen because from our calculations (see section 3.3.3) they are very likely to be used in our case, where, apart from the input pair, all transistors have their channel length quite far from the minimum allowed value. In this way we want to make a comparison between the full and the simplified model in a typical situation.

As for the output resistances, many simulations have been performed with the transistors in different bias conditions. From the file containing the small signal model parameters in a given bias point [2], g_{ds} has been extracted and compared with the value deriving from (4.7) [3]. A relative error of the 40 % has been estimated; this is acceptable given the limited use we do of equation (4.7).

As for the accuracy of the other equations, next sections report the fitting results.

[2] This is an output file produced by Spectre.

[3] The value of V_{DSsat} can be taken from the same Spectre output file as g_{ds}.

4.5.1 Fitting Results for an NMOS

In Fig. 4.4 drain current is represented with respect to the gate-source voltage for a transistor with a V_{DS} fixed to a reasonable value of 850 mV [4]. Error evaluation has to be done considering the range of V_{GS} in which the transistor actually operates and the model is actually used. A reasonable choice is $V_{GS} \in [0.6; 1]$ V since the transistors has to work in strong inversion and with an overdrive that rarely exceeds 400 mV.

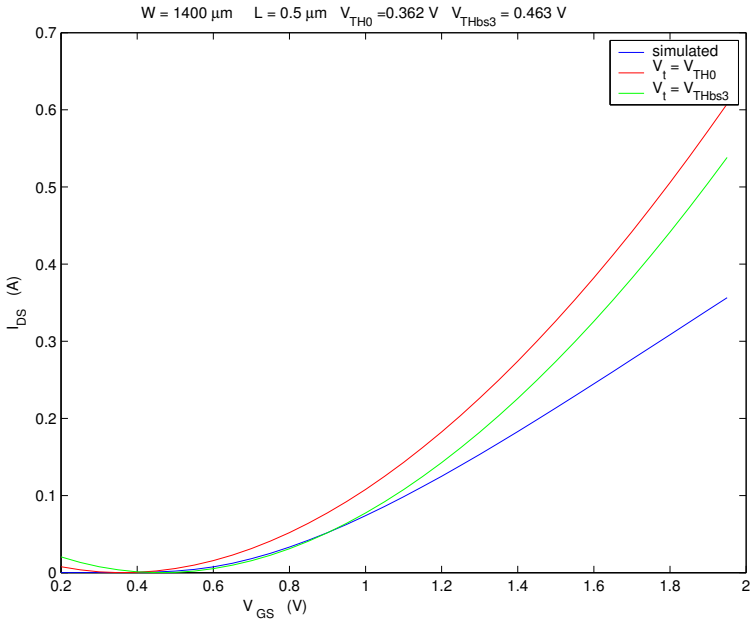


Figure 4.4: I_{DS} versus V_{GS} for a NMOS of the design kit ($V_{DS} = 0.850$ V).

Equation (4.4) has been plotted using at first the parameter VTH0 for V_t as suggested by the simplified model: the result is the red curve which gives error within 40 % in that region. The cause of this large error is the estimation of the threshold voltage: it is not so accurate to substitute the parameter VTH0. In the full $BSIM3v3$ model V_t is a very complicated expression which includes, in addition with the body effect [5], the vertical non-uniform doping effect, the lateral non uniform doping effect, the narrow channel and short channel effect and the drain induced barrier lowering. The

[4] For instance, transistors like M_5 or M_7 in Fig. 3.3 are likely to operate with such a drain-source voltage amplitude.

[5] not present in our case since the source and the substrate have been shortened

actual threshold voltage computed according to the model has been extracted from the simulator and resulted to be $V_{THbs3} = 0.463$ V, 100 mV larger than VTH0. Introducing this new value into (4.4) we obtain the green curve, which guarantees an error within 8 %. This suggest a possible correction of this simplified model in order to adapt it to our particular process. We have shown that it may be not sufficient to use directly VTH0 but sometimes we may need to calculate according to the model the actual value of the threshold voltage or to recover that value from a certain amount of simulations. In our case the second alternative has been followed, which gave us a satisfying result in less time.

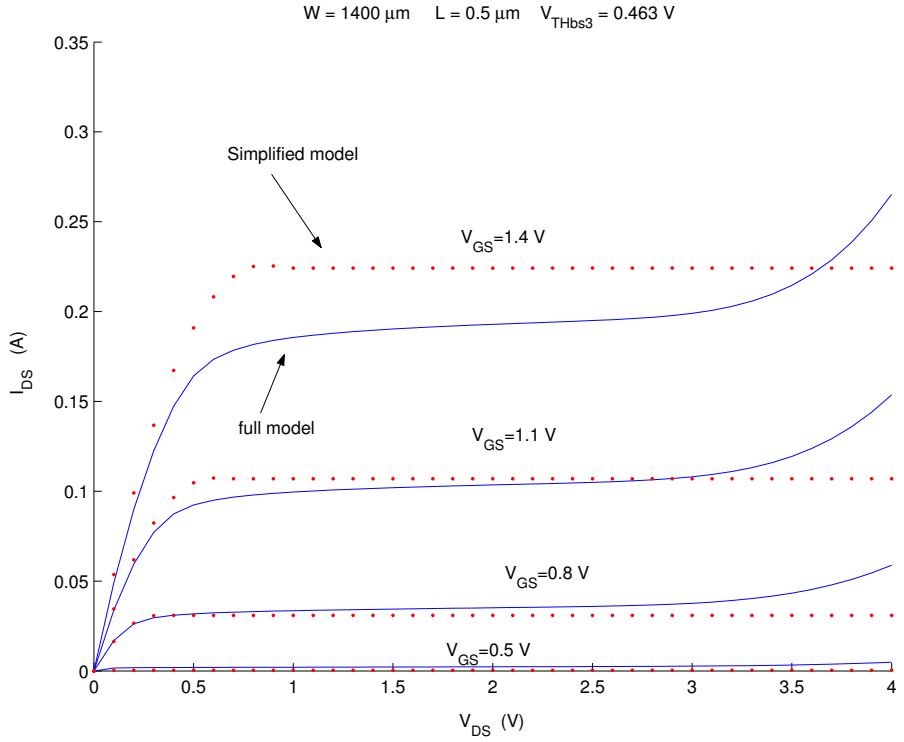


Figure 4.5: I_{DS} versus V_{DS} for various V_{GS} for an NMOS of the design kit. For $V_{GS} = 0.8$ V, and $V_{DS} \sim 1$ V the relative error with respect to the full model is ~ 8 %.

In Fig. 4.5 the MOS characteristics are plotted and once more it is evident that, apart from channel length modulation which becomes relevant only for large V_{DS}, fitting is very good in the typical operating ranges ($V_{GS} = 0.8$ V or 1.1 V) and $V_{DS} \in [0.2; 1.5]$ V (remember that all V_{DS}s in the circuit must

range from ground to $V_{DD} = 2.5$ V). The error becomes larger in the critical region of transition between triode and saturation regions.

In Fig. 4.6 the g_m is represented for $V_{DS} = 0.85$ V. Once more the green plot gives the best fitting within the operating range we are interested in.

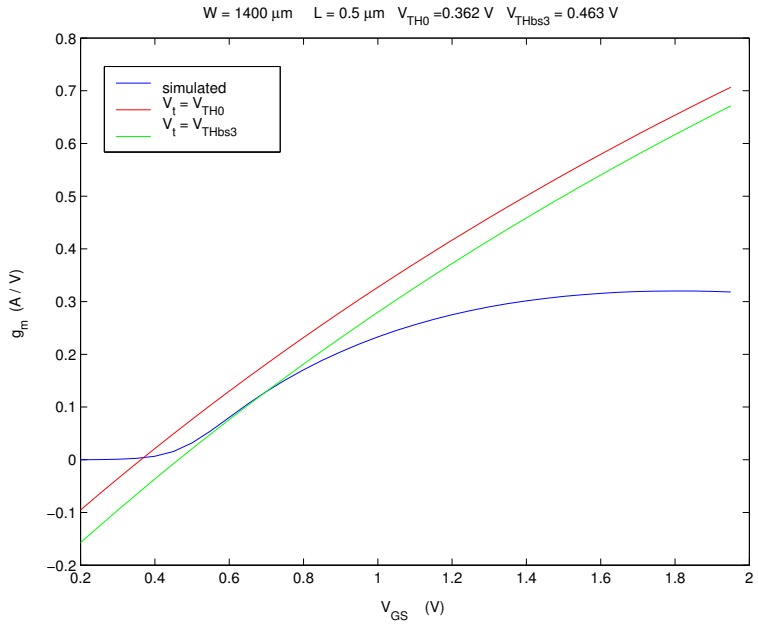


Figure 4.6: g_m versus V_{GS} for $V_{DS} = 0.85$ V. For $V_{GS} = 0.7$ V the red curve gives an error of 28 % while the green and the blue curves practically coincide.

4.5.2 Fitting Results for a PMOS

The same procedure has been followed for the PMOS transistor. Model validation has to be done for $V_{GS} \in [-0.8; -0.6]$ V and $V_{DS} \in [-1.5; -0.2]$ V. The PMOS device proved to be more ideal since the actual value of the threshold voltage is next to the VT0 parameter. In Fig. 4.8 is evident how both the red and green curves give a good fitting result, with a relative error within 5 % with respect to the full model.

In Fig. 4.7 the PMOS I_{DS}–V_{DS} characteristics are shown with the absolute value of the voltages and the currents on the axes so that comparison with Fig. 4.5 becomes easier. The error is ~ 3 % at $V_{GS} = -0.7$ V and $V_{DS} = -1$ V, no matter what value is chosen for V_t.

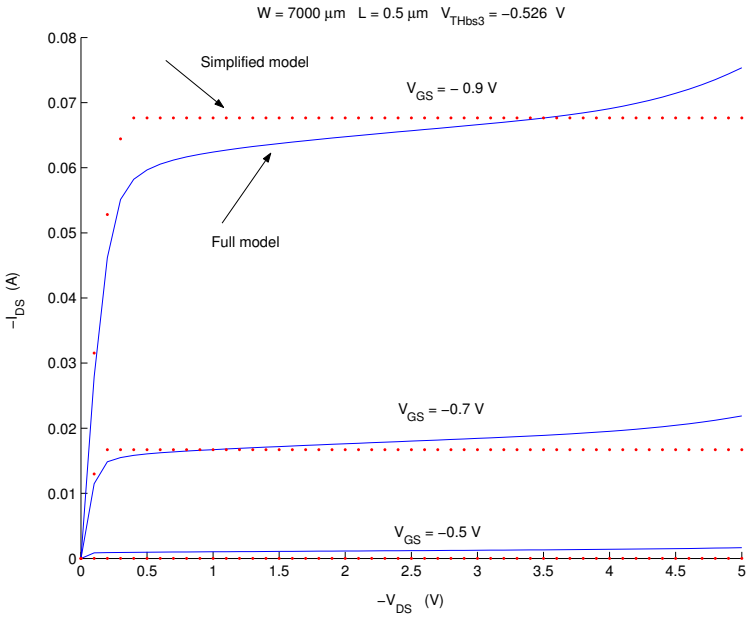


Figure 4.7: $-I_{DS}$ versus $-V_{DS}$ for various V_{GS} for a PMOS of the design kit.

4.6 ACG Generation

All the constraints used in the graph can be divided into four groups. They will be listed here to give an idea of the complexity of the scheduling problem summarized in section 4.7. In fact, the scheduling problem consists in finding a set of random configurations satisfying these constraints. The problem can be modelled in a mathematical way once the constraints have been clearly expressed. In Fig. 4.9 the schematic of the amplifier is repeated so that it is easier to associate the following constraints.

All the constraints are related to one of the branches of the op amp: we suppose perfect matching so that it is sufficient to study only one half of the overall circuit.

4.6.1 Biasing Constraints

A set of relations on configuration parameters is derived to specify necessary conditions for correct biasing, using the new model. All the equations are taken from section 3.3.2 and listed in section 4.4. This constraints are conservative, in the sense that all the circuits that do not respect them are not

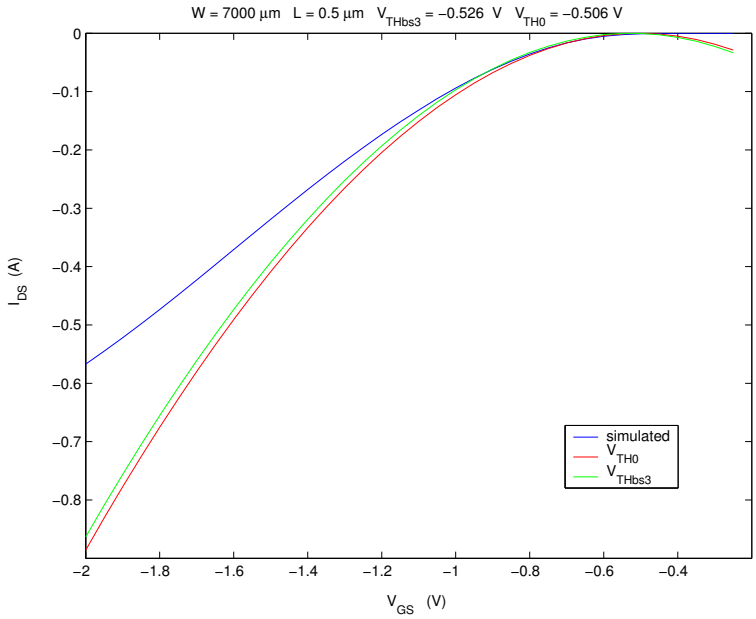


Figure 4.8: I_{DS} versus V_{GS} (a) and g_m versus V_{GS} (b) for a PMOS in the design kit, with $V_{DS} = 0.85$ V.

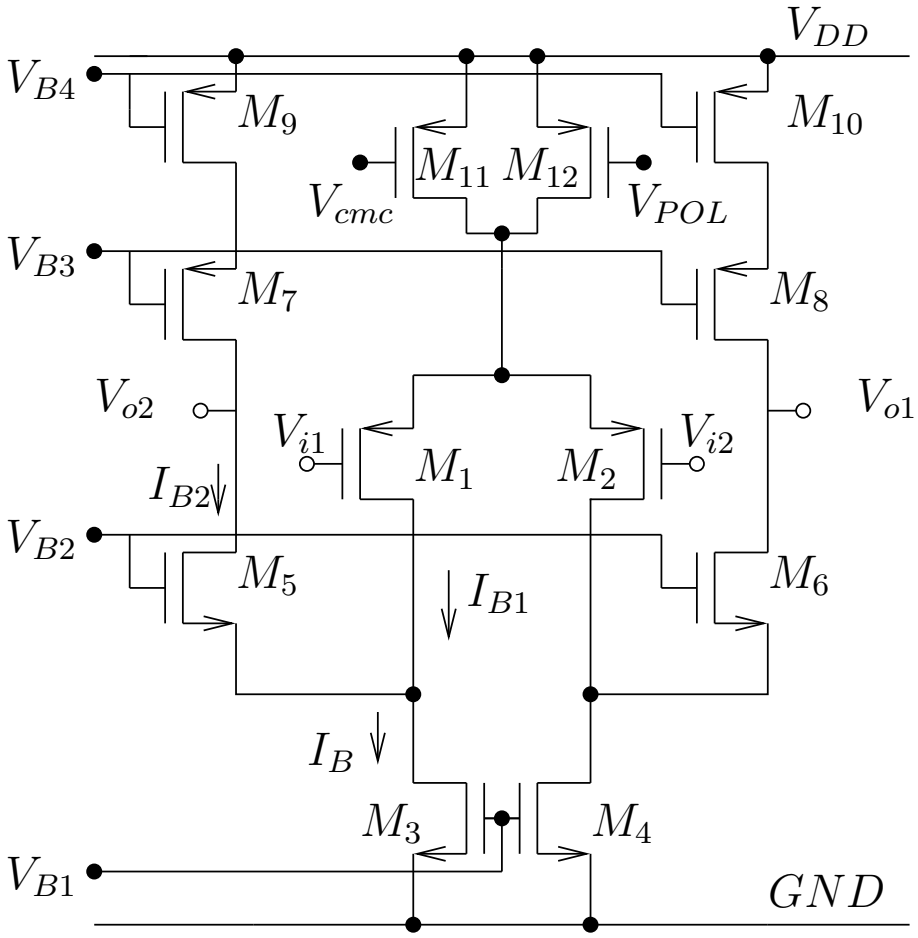


Figure 4.9: Schematic of the folded cascode fully differential op amp.

correctly biased and thus do not work properly. Here is the set of equalities and inequalities written in a general form, valid both for NMOS and PMOS:

$$g_{mi} = \frac{|I_{Di}|}{|V_{GSi} - V_{ti}|}\left[1 + \frac{1}{1 + \left(u_{di} + \frac{1}{\mathscr{E}_{ci}L_i}\right)|V_{GSi} - V_{ti}|}\right] \quad i \in \{1\}, \tag{4.9}$$

$$|I_{Di}| = \mu_{0i}C_{oxi}\frac{W_i}{2L_i}\left[\frac{|V_{GSi} - V_{ti}|^2}{1 + \left(u_{di} + \frac{1}{\mathscr{E}_{ci}L_i}\right)|V_{GSi} - V_{ti}|}\right] \quad i \in \{1, 3, 5, 7, 9, 12\}, \tag{4.10}$$

$$\left|V_{DS(sat)i}\right| = |V_{GSi} - V_{ti}|\left[\frac{1 + u_{di}|V_{GSi} - V_{ti}|}{1 + \left(u_{di} + \frac{1}{\mathscr{E}_{ci}L_i}\right)|V_{GSi} - V_{ti}|}\right] \quad i \in \{1, 3, 5, 7, 9, 12\}, \tag{4.11}$$

$$I_B = I_{B1} + I_{B2}, \tag{4.12}$$

$$0 \leq V_{Bi} \leq V_{DD} \quad i \in \{1, 2, 3, 4\}. \tag{4.13}$$

were all symbols are known.

4.6.2 Conditions on Gain and Bandwidth

Inequality conditions on gain and bandwidth are included to further refine the sampling space $\mathcal{I}$ so that we can reject a large number of configurations which give performances quite far from our specifications. The conditions used derive from (3.9) and (3.11). They can be expressed like this:

$$|A_0| = G_m R_o \geq (2^r - 1)\frac{C_F + C_S}{C_F}, \tag{4.14}$$

and

$$G_m \geq 2f_c\left(C_L + C_S + \frac{C_L C_S}{C_F}\right)\ln\left[2^r\left(1 + \frac{C_F}{C_L + C_S + \frac{C_L C_S}{C_F}}\right)\right]. \tag{4.15}$$

In practice the minimum open loop gain, in constraint (4.14), has been relaxed as shown in Table 3.1 to 55 dB thanks to the benefits of digital correction explained in section 2.2.2. We accept values of gain and bandwidth

greater than some fixed thresholds. Since we only have approximate equations to compute these parameters we have decided to lower the threshold values so that we are sure we are not losing good candidate circuits, with the risk of including bad candidates.

As for R_o, equation (3.21) is used:

$$R_o = (g_{m7} r_{ds7} r_{ds9}) \| [g_{m5} r_{ds5} (r_{ds1} \| r_{ds3})] \tag{4.16}$$

where the various r_{dsi} are computed using:

$$r_{dsi} = \frac{\left\{ (|V_{DSi}| - \left|V_{DS(sat)i}\right|) + [1 + u_{di} |V_{GSi} - V_{ti}|] |V_{GSi} - V_{ti}| \right\} L_i}{I_{Dlongi} l_i P_{CLMi} [1 + u_{di} |V_{GSi} - V_{ti}|]}. \tag{4.17}$$

Another inequality constraint is that one on the phase margin which is imposed greater than 45°. The expression of PM is taken from (3.36). We use the following constraints:

$$C_{in} = C_{gs1} = \frac{2}{3} W_1 L_1 C_{ox1}, \tag{4.18}$$

$$C_o = C_L + \frac{C_F (C_S + C_{in})}{C_F + C_S + C_{in}}, \tag{4.19}$$

$$C_{p1} \approx C_{gs5} = \frac{2}{3} W_5 L_5 C_{ox5} \tag{4.20}$$

,

$$f_H = \frac{1}{2\pi R_o C_o}, \tag{4.21}$$

$$f_1 = \frac{1}{2\pi \left(r_{ds3} \| r_{ds1} \| \frac{1}{g_{m5}} \right) C_{p1}}. \tag{4.22}$$

The following constraint is used to express the unity gain frequency

$$f_0 = \frac{g_{m1}}{2\pi C_o} \tag{4.23}$$

while the condition on phase margin can be expressed by:

$$f_1 > f_0. \tag{4.24}$$

4.6.3 Large Signal Specifications Constraints

Constraints deriving from large signal performance specifications found in section 3.3.2 are:

$$\begin{cases} V_{DS3} = V_{ov3} + \epsilon_3 \\ |V_{DS9}| = |V_{ov9}| + \epsilon_9 \\ |V_{DS12}| = |V_{ov12}| + \epsilon_{12} \end{cases} \tag{4.25}$$

Here ϵ_3 and ϵ_9 are chosen as random variable in the interval $[0.1\,;0.2]$ V in order to bias M_3 and M_9 quite far from the triode region, where the output resistance is deteriorated [6]. ϵ_{12} is a uniform random variable in $[0.15\,;0.25]$ V, selected on the bases of the specifications on the common mode input range, imposing that:

$$|V_{DS12}| = V_{DD} - (V_{IC} + |V_{GS1}|) \tag{4.26}$$

where V_{IC} is the input common mode voltage, which varies within the interval $[0.75\,;1.45]$ V.

4.6.4 Symmetry constraints

Constraints derived from symmetry conditions, such as

$$L_3 = L_9$$

and

$$L_5 = L_7$$

serve to diminish the number of free variables and help scheduling convergence.

Another important constraint is related to the output dynamic range. From the symmetry condition $v_{o1max} = |v_{o1min}|$ we get:

$$V_{DS5} - V_{DSsat5} = |V_{DS7}| - |V_{DSsat7}| \tag{4.27}$$

or, given that the output common mode range has been fixed at 1.25 V, in the middle between GND and V_{DD}, this is equivalent to:

$$V_{DS3} + V_{DSsat5} = |V_{DS9}| + |V_{DSsat7}|\,. \tag{4.28}$$

[6] as explained in section 3.3.2

Given our model it's not possible to impose a value for V_{DS} and in general all equalities in which the various V_{DS}s are involved will not occur in reality. However, V_{DS3} and V_{DS9} are fixed by selecting V_{B4} and V_{B1} and finally the bias network. However from equations (4.27) and (4.28), we infer that even though V_{DS3} and V_{DS9} are not forced to be equals, they will not be too different in module. This is because the output swing will be compromised if one of these V_{DS}s is much smaller or greater than the other. The same concept can be stated for V_{ov3} and V_{ov9}. In order to simplify graph scheduling and to infer reasonable values for these overdrives in a correlated manner it is advisable that their difference be less than 100 mV, that is:

$$|V_{ov3} - |V_{ov9}|| \leq 100\,\text{mV}. \tag{4.29}$$

This is a "trick" in order to generate configurations with parameters correlated in a reasonable manner. A reasonable correlation helps scheduling convergence.

4.6.5 Bounding Boxes

All the primary variables have implicit range constraints attached to them and these also contribute to limit the sample space by embracing it into an hypercube or bounding box. We started from the initial ST design at our disposal and perturbed the parameter values in order to enlarge the scope of exploration. The various Ls are allowed to vary within 20% from their initial values, while we let the Ws vary within 50%. On the contrary I_Bs vary from -75% to $+45\%$ of their original value. The interval is not symmetric with respect to the initial value since our objective is to explore those regions of the design space characterized by smaller values of bias currents and less dissipation of power in order to get more insight in the linearity-power trade-off. Since there are some conditions also involving ancillary variables, it is also necessary to give them a bounding box. For instance, the minimum value for V_{ov} is chosen to assure that each transistor is in strong inversion. However, attention must be paid in this case to the fact that these variables can be expressed as a function of the primary variables and consequently they are already constrained. It is useless to require that V_{ov} of a transistor is included into a particular range if we already know that it will never be in that interval in consequence of the bounds given to I_{DS}, W and L of the same transistor. In this case if we want to prevent failures in scheduling we must return to the primary variables' bounds and try to enlarge them.

It is evident that the exploitation of the constraints is a critical step since if they are too loose the design space will be less biased; in this case it is likely that many useless configurations are simulated and, for those systems that need expensive simulations to be characterized [7], this can be a problem in terms of efficiency. On the contrary if too tight constraints are found the design space become more and more biased; in this case the characterization process may become more efficient but the generation of κs may become more complex. Moreover, there is the risk we have already exposed to bias the design space towards a direction we are not interested in because of the inaccuracy in the formulas used to express the constraints. The delicate choice must be done in each case on the bases of the particular exigencies we have.

4.7 Graph Scheduling

The scheduling algorithm in order to obtain configurations has been implemented through a method `ACG_FDOTA(.)` in MATLAB®. All the code is shown in appendix A; here we illustrate the main idea behind the code. A brief passage extracted from the function is in Fig. 4.10. `ACG_FDOTA(.)` accepts as input parameter `numSim`, i.e. the number of configurations to be produced and gives a vector containing the sizes of each transistors, the bias currents and the sizes of some of the bias network transistors. The function `biasize(.)` computes the sizes of the transistors in the bias network on the bases of the values obtained for $V_{B1}, \ldots, V_{B4}$, during the scheduling phase.

A scheduling procedure has been planned in order to give, although using these simplified models, configurations matching specifications in Table 3.1. The nodes L_1 has selected as fixed, while L_3, L_5, L_{12} have been marked as random variables by assigning them a double triangular probability density function (pdf) centred on their initial values. Nodes I_{B1} and I_{B2} are random variable with uniform pdf. Other ancillary variables have been marked giving a uniform pdf, such as V_{ov3}, V_{ov5} and g_{m1}, the maximum value of which can be selected on the basis of the portion of the space we want to explore. The various node are seen as class objects `SchedNode(.)` and a series of methods have been defined thus implementing all the operations allowed on nodes (e.g. give a value, set as random variable, set constraints,...). All the

[7] This is the case of RF systems that need complex analyses such as PSS or PAC based on shooting methods, or harmonic balance techniques [29].

inequality constraints have been used at the end of the scheduling as control conditions.

Another set of control conditions is given by the bounding box inequalities that hold for the primary variables. It may happen that a parameter obtained by inverting a constraint is not in the admitted variation range: this is a *scheduling failure*. Since this phenomenon is strictly related to the particular architecture analysed, we cannot easily foresee failures and try to diminish them a priori. We simply admit their existence and allow scheduling to restart after each failure. As a consequence, the scheduling function is cyclic and stops when `nvalid`, the number of valid configurations is equal to `numSim`. In the first part of the fragment in Fig. 4.10, there is the node instantiation.

```
function s=SchedNode(flag, min_range, max_offset, varargin)
```

is the class constructor. When `flag` is 0 a new object is created: `max_offset` and `min_range` are read as the maximum and minimum values that the object can assume (bounding boxes). When flag is 1 `max_offset` and `min_range` are read as the an offset and a range parameter and an initial value (a positive number) has to be given as a forth argument to the function. The minimum and maximum allowed values are selected according to the following relations:

```
min = inital_value*(1-0.5*min_range+max_offset);
max = initial_value*(1+0.5*min_range+max_offset).
```

In this way all Ls, Ws, V_{ov}s have their bounds fixed. The method `function y=SetRand(s, fun)` acts on a node `s` by marking it with a random pdf. The node can assume a random value generated through the function `fun` with a particular pdf. For instance `function y=unifGen(mi, ma)` gives a random number uniformly distributed in the interval `[mi; ma]` and can be passed as second argument to `SetRand(.)`. The `function y=Value(s)` extracts the value assumed by the object `s`: this value can be deterministic or random. The method

```
function [y, error]=SetValue(s,value)
```

sets the value of node `s` to `value`. `error` is 0 if `value` is included in the range prescribed for `s`; otherwise `error` is 1 and a scheduling failure occurred.

In Fig. 4.10 two steps of the scheduling are shown. At first, given g_{m1}, L_1 and I_{B1}, a value for $|V_{ov1}|$ is found by inverting constraint (4.9). Then W_1 is

```
nvalid = 0; % counts number of valid configurations
% eps(1) regulates maximum and minimum tolerances on bounds for gm
for i=1:6
    L(i) = SchedNode(1, 0.4, 0, l_in(i));                                    L
    % l are allowed to vary within 20% from initial values
    W(i) = SchedNode(1, 1, 0, weff_in(i));
    % w are allowed to vary within 50% from initial values
    Vov(i) = SchedNode(0, vov_min(i), vov_max(i)); % sets bounds for Vov
end
Ib(1) = SchedNode(1, 1.2, -0.05, ib_in(1));                                  10
% Ibs are allowed to vary from -65% to +55% of initial values
Ib(1) = SetRand(Ib(1), @unifGen);
Gm1=SchedNode(0, gm1_in*(1-eps(1)), 200e-3*(1+eps(1)), gm1_in);
% 200e-3 is maximum gm1 allowed
Gm1=SetRand(Gm1, @unifGen);

while (nvalid < numSim)
 % given gm1, l(1), ibias(2) marked -> vov(1)
 gm1 = Value(Gm1);
 ibias(1) = Value(Ib(1));                                                    20
 l(1) = Value(L(1));     % f(gm, Idsat, L, Vov)=0
 sf_vov=sprintf('%g-%g/x*(1+1/(1+(%g+1/(%g*%g))*x))', gm1, ibias(1), ud_p, ...
    Ec_p, l(1));
 f_vov = inline(sf_vov, 'x');
 vov(1) = fzero(f_vov, vov_in(1));
 [Vov(1), error] = SetValue(Vov(1), vov(1));
 if error == 1
   continue
 end
                                                                             30
 % given ibias(1), l(1), vov(1) -> w(1)
 w(1) = 2*ibias(1)*l(1)/(u0_p*Cox_p)/(vov(1)^2/(1+vov(1)*(ud_p+1/(Ec_p*l(1)))));
 [W(1), error] = SetValue(W(1), w(1));
 if error == 1
   continue
 end
```

Figure 4.10: A fragment of the scheduling code.

computed by inverting constraint (4.10). The inversion is not immediate in certain cases since the model is not so simple as the squared-law model. In these cases the MATLAB built-in function `fzero` has been usefully exploited. The fragment of the code corresponds to a graph fragment, which is shown in Fig. 4.11. Here the original graph has been manipulated and ordered so that it can represent a flow chart for the code in Fig. 4.10.

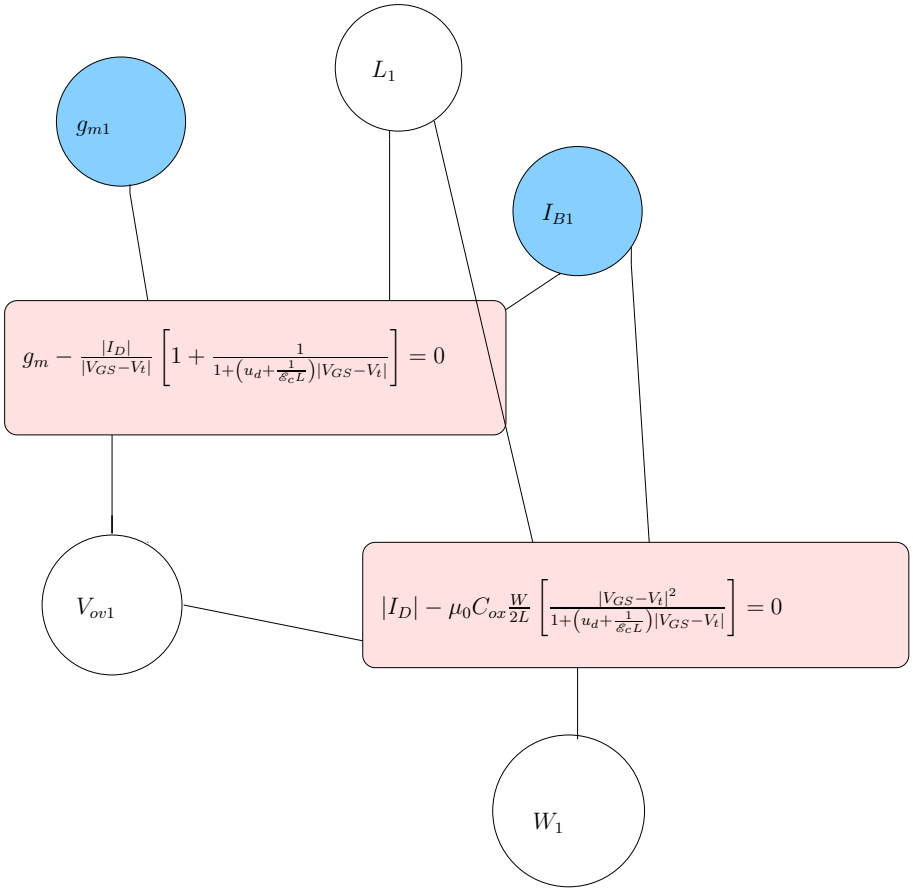


Figure 4.11: Fragment of the ACG corresponding to code in Fig. 4.10—The random marked node are in a blue circle while the constraints are in a pink rectangle.

Some of the configurations produced by the function ($\approx 1,000$) have been represented in Fig. 4.12, projected in a three dimensional space. It is evident how the space $\mathcal{I}$ of the op amp is much smaller than its bounding hypercube. This is the effect produced by all the non-linear constraints imposed to the

variables of each configuration.

The complexity of a graph is strictly dependent on either the complexity of the architecture (e.g. number of transistors, number of bias sources,...), or the number of conditions on performances. The first factor is well measured by the dimensionality of the input space $\mathcal{I}$, while the second one is given by the dimensionality of $\mathcal{O}$ as well as the degree of refinement we want to reach. In our case, as we shall see later, scheduling required approximately 20 minutes to give 100 configurations. However, in an advanced design phase graph refinement may translate into a high number of equations and inequalities (e.g. 30 or 40) so that failure probability in graph scheduling increases and the generation of κ becomes the bottleneck of the characterization process.

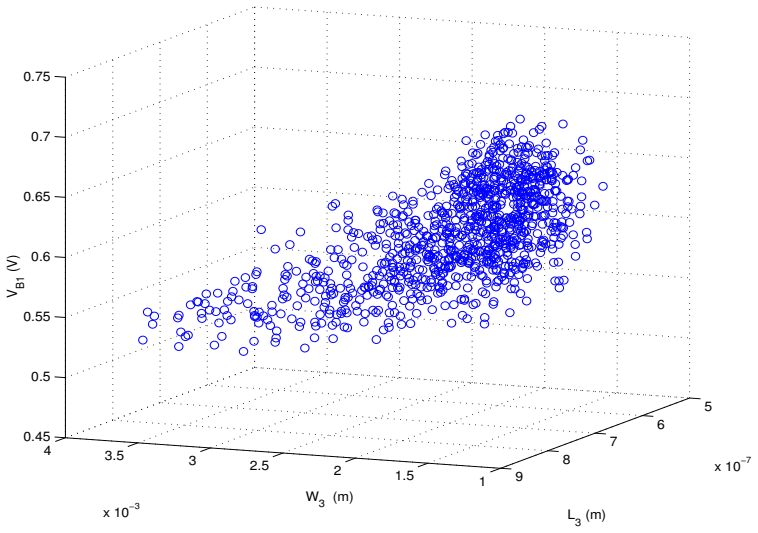


Figure 4.12: A 3-D projection of $1,000$ samples of the configuration space $\mathcal{I}$.

4.8 The Tool for Performance Model Generation

4.8.1 Introduction

In this brief introduction some questions are pointed out, which rise when trying to introduce the simulator into the exploration process. The *encapsulation* ([2]) of the simulator in the exploration and optimization loop is a

problem in itself because of the "unfriendly" behavior in the simulator. Most simulators are designed either for batch oriented operation, or for interactive schematic-update-simulate operation. In the latter the time scales are optimized for humans and overheads of a few seconds for simulation invocation are negligible. But inside an optimizer that seeks to run perhaps 10 000 or more simulations, these overheads are magnified. Our ideal is a simulator which can be invoked once and, remaining live, can interpret quickly a stream of requests to modify circuit values and re-simulate. This means that even the perspective from which we look at simulators is quite different from that of traditional interactive design methodology. Few simulators approach this ideal. For example, some flush all internal state or drop myriad temporary files in the local file system. Of course the maximally difficult behavior exhibited by a simulator is a crash, an event far from rare even in commercial offerings. This is especially problematic in synthesis, in general, and in the design methodology applied here, in particular. Random generation may often visit circuit candidates with highly non physical parameter values, which occasionally cause simulator failure. A good encapsulation of the simulator in an optimization loop or in a characterization loop should not only detects the crash but also restart and reinitializes the simulator.

On the other hand, CPU time should not be a great problem in perspective, since it is destined to diminish thanks to the progress in technology. What's more, CPU time could be negligible with respect to human time, even for an expert designer, to furnish an initial rough project of a complex analog system. This is one of the objectives of the platform based methodology: spending some hours of a computer's time can save the months of designer's time required to complete the design manually or with other analog synthesis tools.

4.8.2 The Client/Server System

The solution adopted in this thesis ([3], [30]) transfers the optimization phase at a system level on the bases of the behavioral models and the performance models we build. Therefore the first phase consists in building these models through a characterization process we are going to describe.

The characterization phase is based on a client-server system which exploits console-based circuit simulation offered by `Ocean` [31, 32], which is the implementation platform for controlling simulation. It has also been used to extract performance figures because of the many functions offered, capable of extracting many parameters (e.g. bandwidth, gain, unity gain frequency, to-

tal output noise) giving to the program simple commands. The most suitable environment to implement schedules is MATLAB, as we have demonstrated before, because a huge mathematical library is available to speed-up implementation of sophisticated heuristics. The problem then is integration with simulations. In the proposed solution:

1. a MATLAB client controls Spectre simulation run on an Ocean server. Matlab generates configuration files that are simulated. Performances are extracted and returned back;

2. Ocean commands cannot be invoked from command-line and so a server solution has be adopted for it. *Synchronization* is achieved by proper file semaphores (Fig. 4.13);

3. since MATLAB run on Windows, *ssh communication* has been set up.

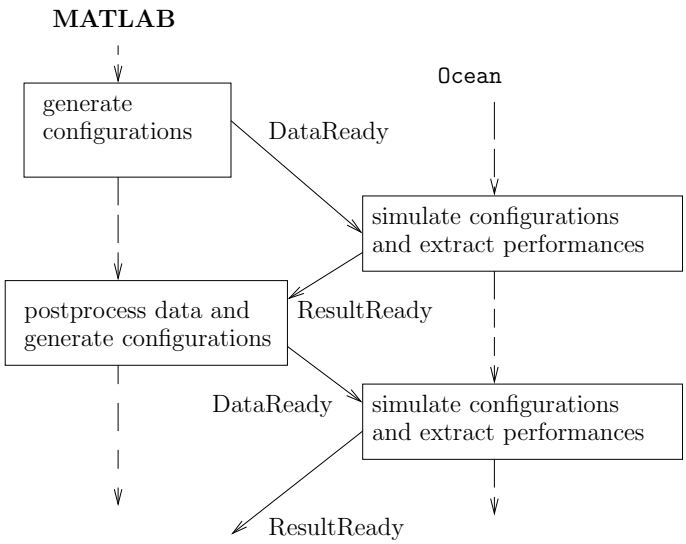


Figure 4.13: Message flow and synchronization for a the client(Matlab)/server(Ocean) system used for performance model generation.

In practice the initial synchronization scheme has been modified to make the characterization process more efficient. In fact, in our case the effort required to the simulator for characterization is not too large. Therefore, the bottleneck of the exploration phase proved to be the configuration generator

because of the high number of constraints and variables, which increase the failure probability. A good idea is to start generating configurations while simulating the preceding ones. In this way the two operations of generation and simulation are executed in pipeline and a gain in efficiency, although not so high, has been registered. In next sections the client and the server will be briefly described.

4.8.3 The Matlab Client

The Matlab client is practically made up of three parts:

1. A group of simulation setup functions in order to transfer files, to upload to the server the data files containing parameter configurations, to download from the server the files with the results from simulations.

2. The functions which implement the scheduling algorithm discussed above.

3. A series of functions for the post-processing of the data obtained, classification of the samples and approximation of the performance relation $\mathcal{P}$ using *Support Vector Machines* (SVMs).

The approximation method used is described in [9] and it will no deeper explained here. We only give some element to help the reader understand what follows. The problem is to build an approximation for the performance relation $\mathcal{P}$. This means to individuate the output performance space reached by all the input configurations satisfying the constraints. Each configuration is a sample of the input space $\mathcal{I}$ and, after having performed the simulation, gives a sample of the output space $\mathcal{O}$. We assume performance evaluation to define a continuous function of the input variables. This is a reasonable assumption while exploring "working" circuits. If we assume $\mathcal{I}$ to be a connected set then $\mathcal{P}$ is a connected set in $\mathbb{R}^m$ (see chapter 1). Given $x_1 \in \mathbb{R}^m$ and its nearest neighbor x_{1N}, if they are "close enough" then all the points in the segment x_1—x_{1N} satisfy $\mathcal{P}$. Therefore, $\mathcal{P}$ can be approximated with the smallest connected set in a given family containing the sample points. Statistical learning machinery is used to infer $\mathcal{P}$ from data. SVMs belong to the class of Large Margin Classifiers pioneered by Vapnik and Chervonenkis in the Sixties. The approximation method through SVMs is based on a simple conceptual model: samples are mapped into a high dimensional space through non-linear functions and hyperplanes are used to separate samples

which should satisfy $\mathcal{P}$ from the other points corresponding to unreachable performances.

4.8.4 The Ocean Server and the Simulation Setup

The Ocean script `serverFDOTA.ocn` must take the configurations from the client and change the circuit parameters accordingly. Successively the script performs simulations required and write results on a set of files. In order to characterize the op amp the following analyses have been chosen:

- A DC sweep of the input differential voltage of the amplifier in order to get and print the *output static characteristic* of the amplifier. This give us some information about the gain of the op amp (slope of the characteristic for $v_{di} \rightarrow 0$) and the linearity (see section 4.8.5). From the bias point data we also can estimate *power consumption*.
- AC frequency sweep to get *bandwidth* (or *gain-bandwidth product*), which is approximately equivalent, once the gain is known.
- A noise analysis which gives the *output rms voltage noise* after integration on the full op amp bandwidth.
- A transient analysis with a high value step voltage (e.g. 1 V [8]) to compute *slew rate*.

The choice of the performance figures must be taken on the bases of the behavioral model of the circuit we want to build. Since this behavioral model will be included in higher level simulations and optimization cycles, the system level specifications will eventually help in selecting those parameters that best model our circuit for our purposes. Useless or redundant figures should be avoided if we don't want to increase the dimensionality of the output space and the simulation time. The server code is in appendix B. The file is included to give an idea of the analyses selected and of the way through which files are used for synchronization.

4.8.5 Post-processing of the Characteristics

Once the output file with the characteristics derived by the simulation of each configuration has been loaded into MATLAB, a function `postProcess(.)` is

[8] also a smaller voltage is sufficient since we simulate the open loop op amp slewing performances

called, which extracts from data two non-linear fitting parameters used to model the op amp input output characteristic. If linear (i.e. polynomial) fitting is adopted the problem becomes bad conditioned: for instance, the third order coefficient is of the order of 10^8 and errors, when trying to approximate the characteristic for a large interval of the input differential voltage, become large. The secret is to chose a base of functions which takes into account more properly saturation phenomena (i.e. $\tanh(x)$, $\arctan(x)$, ...). Non linear fitting is much more accurate than in polynomial case, even if it is more complicated to manage. In Fig. 4.14 all the combinations of functions used to test fitting are visualized.

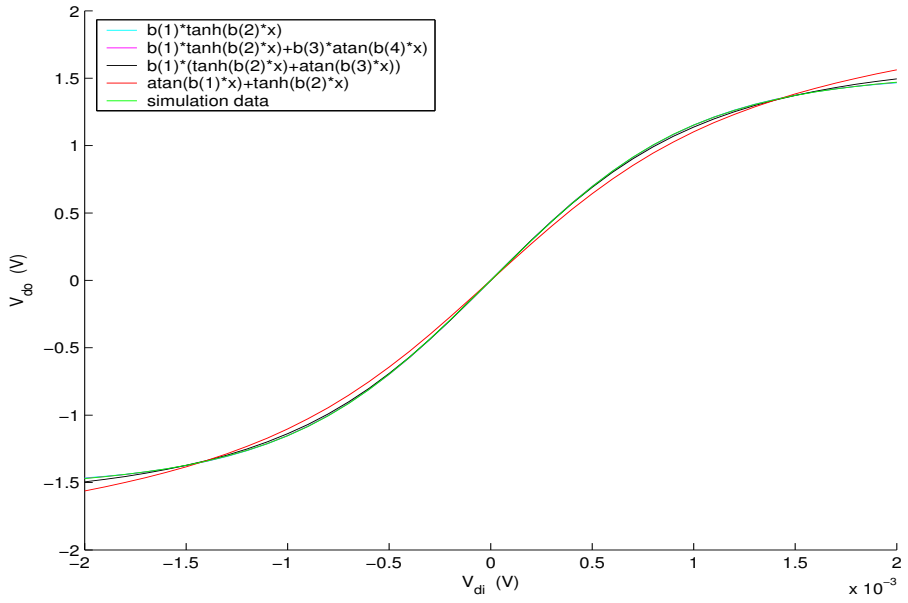


Figure 4.14: Candidate functions for the best fitting.

There are only tanh-based functions, only arctan-based functions, or both. $\mathcal{B} = (\tanh(ax), \arctan(bx))$ proved to be the best base, although it models only odd order non-linearities [9]. Of course, the greater number of fitting parameters we admit the more accurate is fitting but the model becomes more complex. In Fig. 4.15 we can see how a combination of tanh and arctan offers the best fitting in a large range, for V_{di} from -2 to +2 mV.

[9]Mismatch can introduce even order non linearities that can be modelled only by refining the base chosen.

Fig. 4.16 shows the relative error with respect to the maximum value of the output voltage. In the best case it is less than $2 \cdot 10^{-3}$.

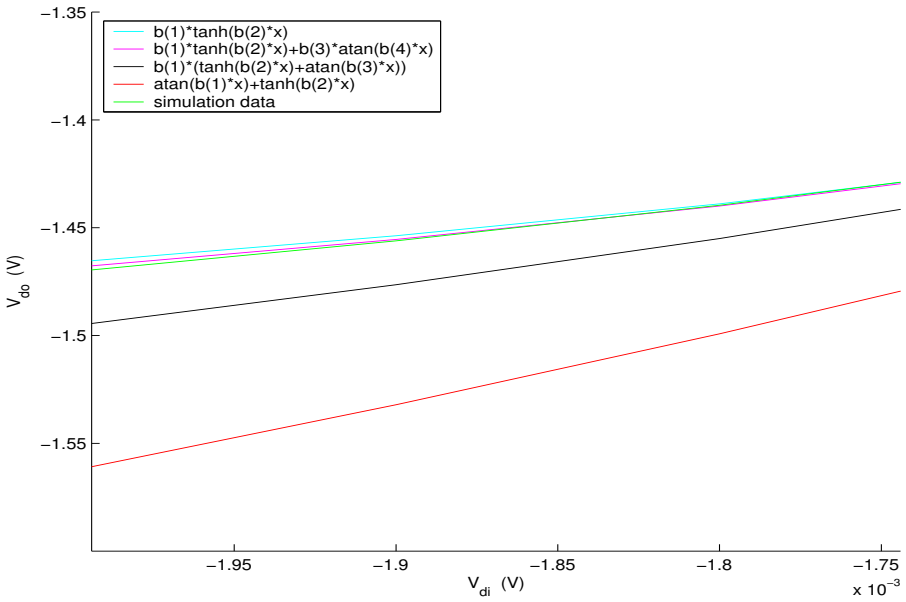


Figure 4.15: Comparison between different nonlinear functions chosen for fitting the op amp characteristic.

We get a good result also by selecting a function of the form $a\tanh(bx)$ (error less than $2 \cdot 10^{-3}$) with the advantage that only two parameters, a and b need to be found. This is the solution adopted for data post-processing. Once the non-linear function has been found, using Taylor series expansion we immediately recover information about gain and third order distortion coefficient, using these formulas:

$$\begin{cases} g = a \cdot b \\ t = -a\frac{b^3}{3} \\ v_{do} = g \cdot v_{din} + t \cdot v_{din}^3 + o(v_{din}^4) \end{cases} \tag{4.30}$$

The above expressions give an idea of the meaning of the two fitting parameters, but they contain much more information since they also can give the higher order odd polynomial coefficients. Moreover the parameters proved to be more easy to handle in computations.

The two fitting parameters linked to gain and third order distortion coefficients are included into the vector ζ of performance figures and used to

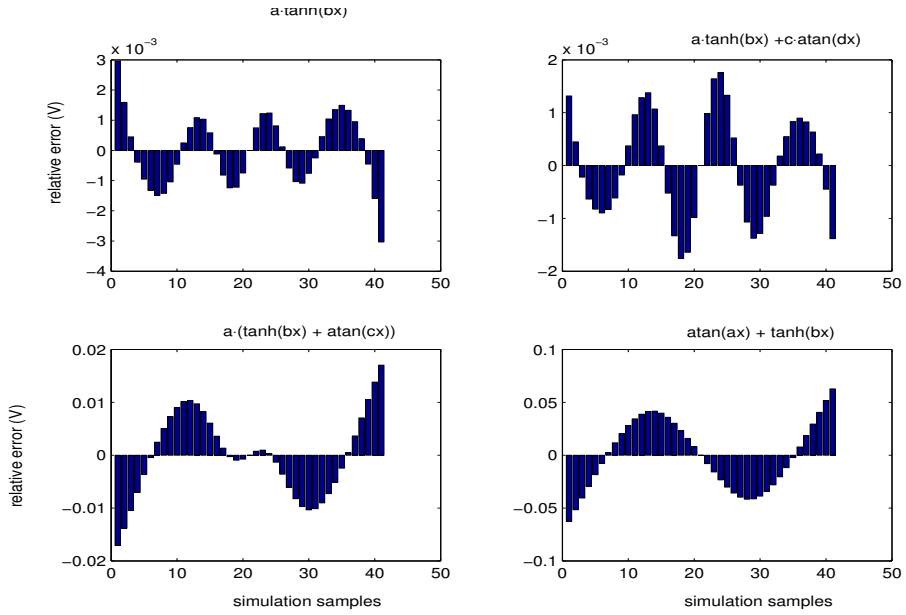


Figure 4.16: Relative error normalized to the maximum output voltage value.

build the performance model.

4.9 The Op Amp Performance Model

It took approximately 14 hours on a Sun Blade 1000 Workstation [10] to perform the overall characterization process. The total number of simulations performed was $2,222$. The stop criterion is decided on the bases of the data from the approximation algorithms. In this section a short description is given of the resulting op amp performance model $\mathcal{P}_{opamp}$. The configuration parameter $\kappa \in \mathcal{I}$ include $\{W_1, W_3, W_5, W_7, W_9, W_{12}, L_1, L_3, L_5, L_7, L_9, L_{12}, I_{bias1}, I_{bias2}, W_{16}, W_{19}, W_{21}\}$. These are the parameters related to the MOS transistors in Fig. 4.9 and in Fig. 3.5, which has been repeated in Fig. 4.17. All the Ws and Ls are degrees of freedom for the amplifier. I_{bias1} and I_{bias2} are the currents of the ideal biasing generators that provide to the amplifier the already cited I_{B1} and I_{B2}. I_{bias1} is the input generator of the mirror controlled while I_{bias2} is the input generator of the bias network studied in section 3.3.4 and coincides with I_0 in Fig. 4.17. These currents are related

[10] with two 750 MHz processors, 1.5 GB of RAM

to that of the two branches of the amplifier through these equations:

$$I_{B1} = K_{s1} I_{bias1}, \tag{4.31}$$

$$I_{B2} = K_{s2} I_{bias2} \tag{4.32}$$

where K_{s1} and K_{s2} are the two ideal mirror gain which have a fixed value chosen so that currents from generators be of the order of a few mAs or less, which is reasonable for this circuit. W_{16}, W_{19} and W_{21} are used to fix the various V_Bs needed by the amplifier and shown in Fig. 4.9. The various Ls of the bias network have been left fixed for instance to 0.7μm, since the influence of their actual value on the amplifier performances is negligible. It can be seen that once this variable has been fixed and the gain of the current mirrors has been decided all the other transistors, whose sizes are not included in κ, are easily sized.

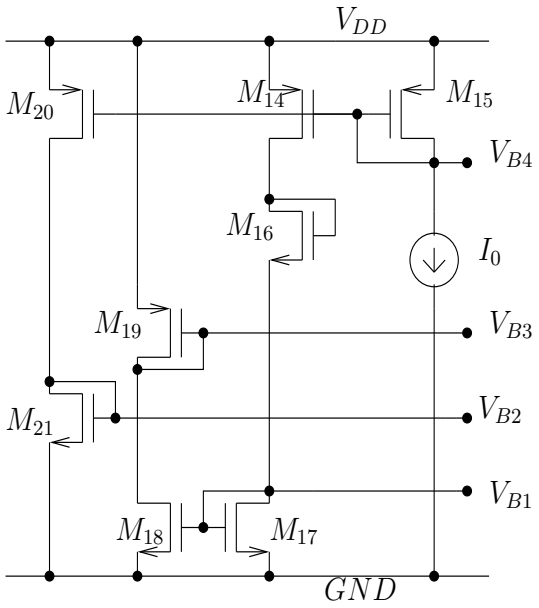


Figure 4.17: Biasing circuit for the folded cascode op amp.

The considered performance figures $\zeta \in \mathcal{O}$ include {Bandwidth, Slew Rate, Noise, A_1, A_3} where noise is the rms value of the total output noise and A_1 and A_3 are the two fitting coefficients derived from the input-output static characteristic of the amplifier through a non linear fitting (see section

4.8.5). They give us information about the amplifier linearity performances: they approximate the first order and the third order coefficients of the polynomial expansion of the characteristic of the circuit considered as a nonlinear memoryless system [11]. The evaluation function $\phi(.)$ is a Spectre simulator.

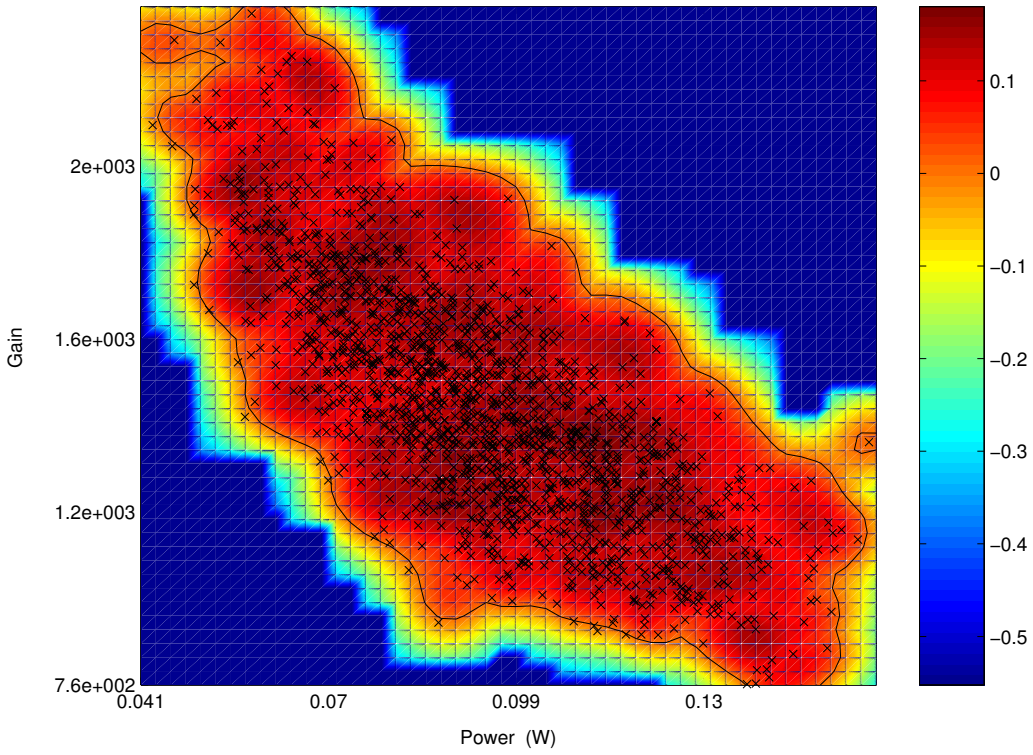


Figure 4.18: Example of 2-D projection of performance relation $\mathcal{P}$—The black cross-marked points are the performances obtained from simulation of configuration samples.

In Fig. 4.18 a two-dimensional projection of the output performance space $\mathcal{O} \subset \mathbb{R}^6$ is shown. The red zone is the one of the feasible designs, the blue one coincides with the unfeasible region. The feasible region includes those performances achieved by the configurations generated and by other configurations supposed to produce feasible performances according to the approximation algorithm. The color gradation indicate the level of confidence according to which a region belongs to the feasible space or not. Image like

[11] Note that if mismatch is neglected even order non-linearity are absent in a fully differential amplifier.

this are important in our methodology for two motives at least. First of all they are the graphic representation of the feasible performances and hence can be used to constrain the system level optimization process. For instance, now we could look for an amplifier with a gain higher than a certain value g_0, power consumption lower than p_0, but these values cannot be chosen arbitrarily. They must be in the red region of Fig. 4.18, which gives the feasible pairs $\{g_0, p_0\}$. Secondly, these figures give a powerful visualization of the trade-offs involved in the design process and may constitute a useful map for the designer. In Fig. 4.19 *a*) the projection on the power-noise plan is shown, while in Fig. 4.19 *b*) the projection is done on the gain-noise plan. The qualitative trend was already predictable, but now more quantitative information taken from simulated, and hence "experimental", data is available.

Note that in Fig. 4.19 *b*) the total output noise is represented with respect to the amplifier gain: a reasonable almost linear trend is evident. Another example of the performance model projection is in Fig. 4.20. We can appreciate the gain represented versus bandwidth and, what could be very interesting for us, an estimation of the third order non-linearity coefficient is represented versus power. If one would evaluate the possibility of finding a feasible circuit with less power consumption, he could have an approximation of the amount of linearity he has to trade-off.

4.10 Communication Issues and Model Composition

Behavioral models do not have any intrinsic loading notion, which may have a huge impact on circuit performances. Communication between analog platform has to be explicitly modelled. In our case it is not so immediate to model the interconnection: it is not useful to express them in terms of input and output impedances, since the interface is quite complex. As we discussed in section 3.2.1 input impedance can be considered to be infinite, while the load is made up by the capacitors of the next interstage amplifier, and the capacitors of the second stage sub-ADC. Moreover, as we explained in section 4.8.4, the amplifier has been characterized loaded by the estimated capacitance C_{LOAD} which also include the input parasitic capacitance C_{in} of the amplifier, function in the dimensions of M_1. This is the configuration useful to get the closed loop bandwidth which is one of the most relevant performance parameters of our amplifier. In fact, we remember that closed

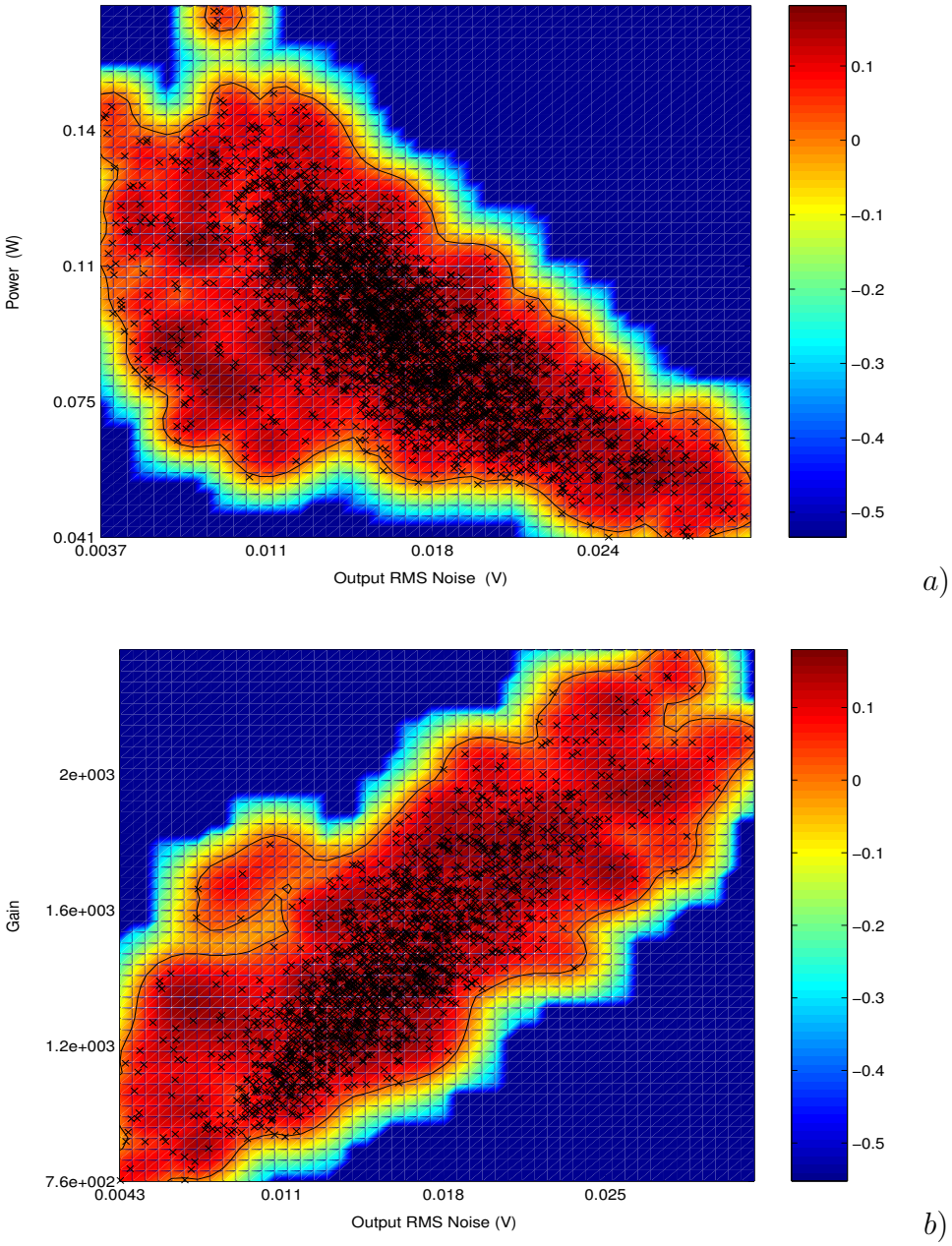


Figure 4.19: 2-D Projections of performance relation $\mathcal{P}$—It is now evident how the feasible space can be approximated with the smallest connected set in a given family, containing the sample points.

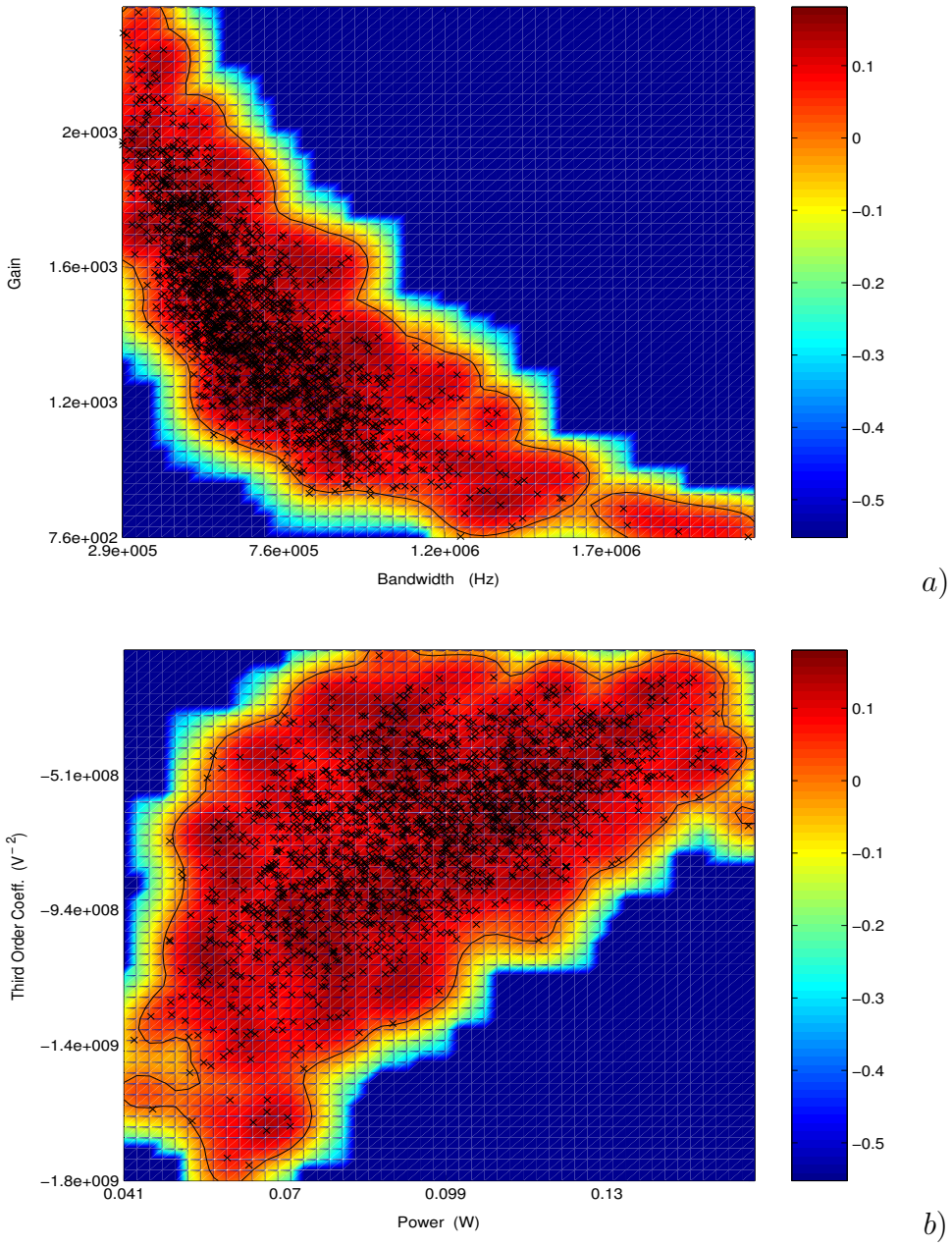


Figure 4.20: 2-D Projections of performance relation $\mathcal{P}$— Gain-bandwidth trade-off is visualized in a) while linearity-power trade-off is visualized in b).

loop bandwidth coincides with the open loop gain-bandwidth product under these hypothesis and this is the parameter the designer can directly control.

In our case the load capacitance has been kept fixed. This means that loading effects have been explicitly included in the performance model so that our characterization is not the *generic* characterization of a folded cascode fully differential amplifier, but a *particular* platform of a particular folded cascode in some specific charging conditions. This means that our platform has been built during an advanced phase of the design process, when the architectural space has been already refined.

In other words, our platform has been intended for system level optimization of the first stage residue amplifier. Anyway, this does not limit its future re-use in sizing the other stages' amplifier, if load capacitances are properly parameterized. This is easily done introducing in the Ocean code one more design variable, and to give it a bounding box, so that characterization can be performed in different loading conditions.

Through this approach all the composition effects are modelled in the output port of the driving block (the amplifier) so that all the performance figures include the loading effect of the following blocks. This approach implies that at least a simplified input stage of the loading block has to be evaluated at the same time as the driving one. This is the disadvantage of the platforms approach in terms of loss of abstraction: a actual general characterization cannot be done since composition is limited to well determined blocks and topologies. Anyway, this not only a disadvantage of this methodology, but is an inevitable obstacles of the analog world, i.e. the reciprocal interrelations between the different blocks.

4.11 The Op Amp Behavioral Models

In this sections two behavioral models are shown: the first is a continuous time model, the second is a discrete time one. Both models can be used for system level simulation. The second one is more convenient to simulate a switched capacitor system because it already works in the discrete time domain and requires less time for simulation. The models proved to accurately represent the behavior of the amplifier for typical input voltage ranges. The new library blocks have been implemented in the popular MATLAB *Simulink* environment. For each block a description of the considered effect as well as all the implementative details are provided. Once these models have been validated in different operating conditions, they can constitute, constrained

by the performance models, a simulation environment for system level optimization.

4.11.1 Continuous Time Model

The first system models the open loop operational amplifier and it is represented in Fig. 4.21. The first block models the op amp linearity and saturation behavior. This is a non-linear memoryless block which implements the function $f(u) = a\tanh(b \cdot u)$ and consequently the input-output amplifier transfer function. Validity is assured for that input range used to extract, through non-linear fitting, the parameters a and b. A white noise block has been added to models the total output noise power. The filtering and slewing properties of the amplifier have been treated together through the S-Function (System Function) `csfOpamp(.)`.

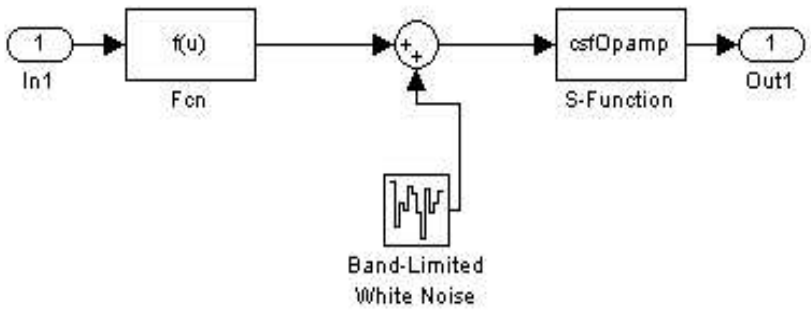


Figure 4.21: Continuous time model of the open loop operational amplifier.

The S-function implements the continuous time equations typical of a dynamic system:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \tag{4.33}$$

where x is the status of the system, u is the input, y is the output and A, B, C, D are the matrices of the state space representation. For systems with a single pole at frequency f_p and unitary low-frequency gain we get:

$$A = -2\pi f_p, \quad B = 1, \quad C = 2\pi f_p, \quad D = 0. \tag{4.34}$$

The second equation has been adequately modified to include slewing effects. From the relation:

$$|\dot{y}| \leq \mathrm{SR} \tag{4.35}$$

we infer the following relation for the state derivative:

$$\dot{x} = \mathrm{sign}(Ax + Bu)\min(|Ax + Bu|, \mathrm{SR}/C). \tag{4.36}$$

This amplifier has been used to simulate the closed loop circuit settling behavior during the phase of operation when ϕ_2 is high. We imagine that a step of amplitude v_S is given at the input, then the output voltage v_o of the amplifier is given by the solution of the following system:

$$\begin{cases} v_o = \mathrm{f}(v_i, a, b, v_n, f_p, \mathrm{SR}) \\ v_i = \frac{C_S}{C_S+C_F} v_S + \frac{C_F}{C_S+C_F} v_o \end{cases} \tag{4.37}$$

where v_i is the voltage at the input port of the op amp, v_n is the rms output noise voltage, f is the open loop op amp model function and the other symbols are already known. The only hypothesis here is that current absorbed by the amplifier is negligible. The system has been given to Simulink as shown in Fig. 4.22. Simulink solves the system with the same techniques used for algebraic loops even if the first equation of the system is a differential equation, not an algebraic one. It has been verified that if v_S is in the linear input range of the op amp, then the closed loop system is also linear and evolves as a single pole system. The new frequency is equal to $\beta \cdot \omega_0/(2\pi)$ where ω_0 is the op amp gain-bandwidth frequency (in rad/s) and β is the feedback factor.

In Fig. 4.22 the coefficients $\alpha = \frac{C_S}{C_S+C_F}$ and $\beta = \frac{C_F}{C_S+C_F}$ have been implemented using two gain blocks. Values for C_S and C_F are in Table 3.1.

Some problems have been met during simulations. These were overcome by diminishing to 10^{-10} the absolute and relative error tolerances in setting the simulation parameters.

4.11.2 Discrete Time Model

The discrete time model represents the behavior of the closed loop amplifier in the discrete domain, so that the output is given at each sampling instant. The differential equation $v_o = \mathrm{f}'(v_i,\ a,\ b,\ f_p,\ ,\mathrm{SR})$ once the sampling time T_s has been given, becomes a simple function

$$v_o = \mathrm{g}(v_i,\ a,\ b,\ f_p,\ \mathrm{SR},\ T_s) \tag{4.38}$$

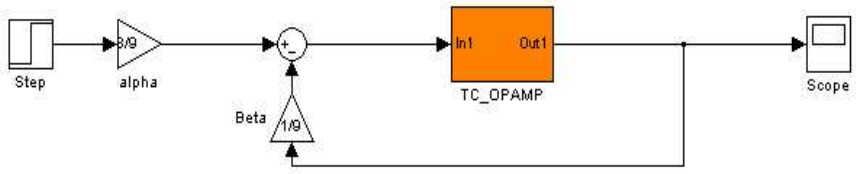


Figure 4.22: Continuous time model of the closed loop interstage amplifier.

which, given the input voltage v_i and the other parameters, gives one real value for v_o [12], i.e. the value assumed by the output at the end of the half-period of the clock signal, when the output is actually sampled. This final value is a function of the pole, the slew rate and the distortion introduced by the op amp. Note that to simplify the problem, noise has been neglected in this phase [13].

It can be shown that the output voltage v_o of the amplifier with slew rate limited settling can be calculated as [1]:

$$v_o = \begin{cases} v_f(1 - \mathrm{e}^{-\frac{T_s}{2\tau}}), & |v_f| \leq v_{lin} \\ v_f - \mathrm{sgn}(v_f) v_{lin} \mathrm{e}^{\left(\frac{|v_f|}{v_{lin}} - \frac{T_s}{2\tau} - 1\right)}, & v_{lin} < |v_f| \leq v_{lin} + T_s \cdot \mathrm{SR}/2 \\ \mathrm{sgn}(v_f) \cdot T_s \cdot \mathrm{SR}/2, & v_{lin} + T_s \cdot \mathrm{SR}/2 < |v_f| \end{cases} \tag{4.39}$$

where v_f is the step size of the output voltage at $t \to \infty$, SR is the slew rate, τ is the closed loop time constant and v_{lin} is the maximum step size that gives linear settling. The maximum step size for linear settling can be calculated as

$$v_{lin} = \mathrm{SR} \cdot \tau. \tag{4.40}$$

The function `g(.)` with the observations given above has been implemented through a MATLAB function. If we substitute the expression for

[12] not a waveform

[13] It could be included with the insertion of an equivalent noise source at the input of the amplifier.

v_i from the second equation of the system (4.37) into the equation (4.38), we obtain $v_o = \mathrm{h}(v_o, v_s, a, b, f_p, \mathrm{SR}, T_s)$: v_o is a fixed point of h. The non-linear equation can be solved numerically by MATLAB, using, for instance, the *Newton* method. The block capable of solving this equation is implemented as an S-function shown in Fig. 4.23 and called `td_opamp`. Distortion is taken into account by computing the final value for the step response using the tanh-approximation.

Figure 4.23: Discrete time model of the closed loop interstage amplifier.

In this way, given a sampled input, the block gives the output of the amplifier at each sampling instant.

4.12 Conclusions

In this chapter all the process of op amp characterization has been described. A new simplified model of transistors has been validated and used to express the constraints that the amplifier has to satisfy. The constraints are related to bias conditions, small signal and large signal behavior and symmetry properties. An ACG has been used as a graphic representation of the constraints. A scheduling algorithm allows generating random circuit configurations respecting the constraints. These configurations are simulated and performances manipulated to get an approximation of the feasible performance space for the op amp. The resulting performance model can be used to constrain the behavioral models proposed in this chapter when performing a system level optimization process. The overall characterization process

($\sim$ 2,000 simulations) proved to be quite efficient; it lasted approximately fourteen hours on a Sun Blade 1000 workstation. It is possible to evaluate the proposed methodology on the bases of the performance metrics we have listed in chapter 1. As for *accuracy*, the new methodology uses the simulator as an instrument of performance prediction, thus exploiting the realistic and very accurate device models; the evaluation scheme proposed is *absolutely general*: the only parts that should be customized are the scheduling algorithm and the simulation setup. Using platforms at different levels of abstractions, by propagating constraints, it should be possible to design and optimize very complex systems, such as the converter of this thesis. *Time required* and *preparatory effort* strongly depend on the complexity of the architecture to be encapsulated (number of devices, performance variables,...). In any case time and effort are incomparable to those required for a completely new design. However, the strength of the methodology is the fact that it offers an *abstraction* level between circuit design and system design thanks to the capability of providing system level models based on accurate experimental data.

Conclusions

This thesis gave a little but important contribute to research in the field of the design methodologies for analog integrated circuits: a new methodology has been applied, based on Analog Platforms, to an industrial project, offered for study by *STMicroelectronics*. The system is a fully differential folded cascode used as the first stage residue amplifier of a high performance analog-to-digital pipeline converter implemented in 0.13 μm CMOS technology. The objective was to perform the architectural space characterization for this amplifier. This objective has been successfully achieved and the results obtained proved to be very interesting.

First of all, a preliminary analysis of the overall system has been performed in order to find the specifications of the amplifier. Secondly, the op amp has been analyzed to get the equations and the inequalities that link the architecture performance figures to each transistor's sizes. These constraints have been introduced into a MATLAB function which generates random configuration sets obeying those constraints. In fact, the problem of generation of configurations has been modelled through a set of variables and a number of constraints smaller that the number of variables. This is an underdetermined problem which gives infinite solutions, even if many of them must be rejected. The constraints were expressed by using a simplified model deriving from the BSIM3v3. This model includes the short channel effects and mobility degradation effects. This model has been validated and proved to approximate full model data with acceptable errors, within the operating conditions of the various transistors.

An Ocean script has been also adequately adapted to extract in an efficient way the performance figures for the selected architecture. This script controlled in batch mode the circuit simulator so that we got the performances of each configuration. Both the MATLAB method and the Ocean script worked properly. The huge quantity of information from simulation served to build a performance model to constrain the op amp behavioral

model.

Two behavioral models have been proposed, in the continuous time domain and in the discrete time domain. They simulate the amplifier open loop and closed loop behavior. Next objective is to exploit these models for high level simulations. The models will help finding optimal specifications for the amplifier and evaluating quantitatively how much power can be actually saved without excessively compromising the converter linearity and accuracy. The op amp design can be consequently optimized with respect to power consumption and the amount of data of the exploration phase will make easier to figure out novel strategies of calibrations to compensate the distortion component due to the op amp non-idealities.

Each step while building the platform has been diffusely explained. The problems met and the solutions found have been exposed. It is advisable to give a proper explanation of the proposed strategies, when dealing with design methodologies. The purpose must be to simplify the designer's life, by offering efficient procedures meant to reduce the design time, but without limiting his creativity. Consequently, future efforts should be directed to make the methodology more flexible. For instance we remind that, for the generation of the configurations, an analog constraints graph have been exploited, so that generation became a scheduling problem. It has been pointed out how, especially when trying to refine the graph, some considerations must be done depending on the particular architecture. It is advisable to find how the scheduling algorithm could be generalized or how to include the platform related elements into the graph paradigm.

This enables the implementation of a completely automated characterization tool. This tool, if provided with a user-friendly interface, could become an extremely useful support for the analog designer.

Conclusioni

La tesi ha fornito un piccolo ma importante contributo alla ricerca svolta nel campo delle metodologie di progetto per circuiti integrati analogici: è stata applicata una metodologia sperimentale, basata sulle Piattaforme Analogiche, ad un progetto industriale, gentilmente messo a disposizione dalla *STMicroelectronics*. Il sistema oggetto di studio è un amplificatore folded cascode, a doppia uscita, utilizzato come amplificatore del residuo del primo stadio di un convertitore analogico-digitale pipeline. Il convertitore, ad elevate prestazioni, deve essere realizzato in tecnologia CMOS 0.13 μm. L'obiettivo proposto era quello di condurre la caratterizzazione dello spazio architetturale per l'operazionale. Tale obiettivo è stato pienamente raggiunto e i risultati ottenuti si sono rivelati molto interessanti.

E' stato effettuato uno studio preliminare del sistema nella sua globalità per ricavare le specifiche del blocco amplificatore. In seguito si è passati all'analisi dell'amplificatore per arrivare alle equazioni e alle disuguaglianze che legano i parametri di merito della cella con le dimensioni dei transistori. Tali vincoli sono stati introdotti nel corpo di una funzione MATLAB in grado di generare insiemi di configurazioni casuali di circuiti che rispettano comunque quei vincoli. Infatti, il problema di generazione di configurazioni circuitali è stato modellato attraverso un insieme di variabili e un insieme di vincoli non lineari che sono in numero minore rispetto alle variabili. Si tratta di un problema sottodeterminato in grado di produrre infinite soluzioni, non tutte valide. Per esprimere le equazioni da introdurre nella funzione MATLAB si è adoperato un modello semplificato derivato dal BSIM3v3, che tiene conto anche degli effetti di canale corto e della degradazione della mobilità. Tale modello è stato preliminarmente validato e si è dimostrato capace di approssimare i dati del modello completo, nelle condizioni di lavoro dei transistori, con errori più che accettabili.

E' stato anche predisposto uno script Ocean opportunamente adattato affinché potesse estrarre in modo efficiente i parametri di merito dell'archi-

tettura in esame. Tale script controlla, da riga di comando, il simulatore di circuito ottenendo così le prestazioni d'ogni configurazione simulata. Il codice messo a punto per la caratterizzazione ha adempiuto pienamente la funzione per cui era stato ideato. La notevole mole d'informazioni derivate dalle simulazioni è servita a creare un modello di prestazioni concernenti la particolare architettura, per vincolare il modello comportamentale dell'operazionale.

Sono stati proposti due modelli comportamentali, uno tempo continuo e l'altro tempo discreto, dell'amplificatore, capaci di simulare il suo comportamento sia ad anello aperto che ad anello chiuso. Il prossimo obiettivo è quello di sfruttare tali modelli per simulazioni ad alto livello, che consentano di individuare un insieme di specifiche ottime per l'amplificatore. Attraverso simulazioni ad alto livello si potrebbe valutare, ad esempio, quanta potenza si possa risparmiare senza eccessivamente compromettere la linearità e l'accuratezza del convertitore, cercando così di ottimizzare il progetto dell'amplificatore riguardo al consumo di potenza. I dati raccolti nella fase di esplorazione potranno essere d'aiuto per escogitare delle strategie di calibrazione per il convertitore volte a compensare gli effetti della distorsione dovuta alle non idealità dell'operazionale.

Ogni passo compiuto verso la messa a punto della piattaforma è stato ampiamente discusso. Tutti i problemi incontrati sono stati elencati e si è spiegato come si sono superati. E' doveroso, per chi si occupa di metodologie di progetto, curare anche l'aspetto "didascalico". Quando si propone o si applica una metodologia, il fine deve essere quello di semplificare la vita al progettista, mettendogli a disposizione delle procedure il più possibile semplici ed efficienti, nel senso che riducono i tempi di progetto e gli sforzi necessari. Tali procedure mai devono essere tanto rigide e codificate da limitare la sua creatività o da risultare di laboriosa applicazione. Un impegno maggiore pertanto sarà richiesto, in futuro, per rendere ancora più flessibile la metodologia utilizzata in questa tesi. Per esempio, si ricorda che, per agevolare la scrittura del codice per la generazione delle configurazioni, sono state proficuamente adottate le potenzialità della rappresentazione grafica offerta dal grafo dei vincoli. Generare configurazioni diventa allora un problema di schedulazione del grafo. Questa tesi ha evidenziato come questo passo sia il più delicato della metodologia giacché le valutazioni di fattibilità si basano sullo spazio delle prestazioni esplorato che dipende strettamente dalle configurazioni prodotte. Si è visto come, soprattutto in una fase di rifinitura del grafo, sia necessario, per uno "sbroglio" più efficiente, tenere conto di alcune considerazioni dipendenti dalla particolare architettura. Sarebbe auspicabile escogitare dei modi per svincolare lo sbroglio dalla particolare piattaforma

oppure per includere entro il paradigma del grafo degli espedienti che ne avessero tenuto conto.

Questo faciliterebbe l'implementazione di un tool completamente automatico, che, se corredato da un'interfaccia amichevole, potrebbe costituire un validissimo supporto per l'attività del progettista.

Bibliography

[1] M. Gustavsson, J. J. Wikner, and N. N. Tan. *CMOS Data Converters for Communications*. Kluwer Academic Publishers, 2000.

[2] R. Phelps, M. Krasniki, R. A. Rutenbar, L. R. Carley, and 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(no. 6):pp. 703–717, June 2000.

[3] F. De Bernardinis. "A Methodology for System-Level Analog Design Space Exploration". Report, April 2003.

[4] L. Carloni, F. De Bernardinis, A. Sangiovanni Vincentelli, and M. Sgroi. "The art and science of integrated system designing". In *Proceedings of ESSCIRC'02*, pages pp. 25–36, September 2002.

[5] B. De Smedt and G. G. E. Gielen. "WATSON: Design Space Boundary Exploration and Model Generation for Analog and RF IC Design". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22(no. 2):pp. 213–224, February 2003.

[6] R. Harjani, R. A. Rutenbar, and L. R. Carley. "OASYS: A Framework for Analog Circuit Synthesis". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 8(no. 12):pp. 1247–1266, December 1989.

[7] E. S. Ochotta, R. A. Rutenbar, and 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(no. 3):pp. 273–294, March 1996.

[8] M. Hershenson, S. Boyd, and 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, January 2001.

[9] F. De Bernardinis, M. I. Jordan, and A. Sangiovanni Vincentelli. "Support Vector Machines for Analog Circuit Performance Representation". In *Proceedings of DAC*, June 2003.

[10] E. Biorci. "Progettazione Top-Down di un A/D a 14 bit e 80 Msps con correzione digitale delle non linearità". Tesi di Laurea, Università degli Studi di Pavia, 2001?

[11] J. Doernberg, H. Lee, and D. A. Hodges. "Full-Speed Testing of A/D Converters". *IEEE Journal of Solid-State Circuits*, vol. sc-19(no. 6):pp. 820–827, December 1984.

[12] B. Razavi. *Priciples of Data Conversion System Design*. IEEE Press, 1995.

[13] I. Galton. "Digital Cancellation of D/A Converter Noise in Pipelined A/D Converters". *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, vol. 47(no. 3):pp. 185–196, March 2000.

[14] S. H. Lewis and P. R. Gray. "A Pipelined 5-Msample/s 9-bit Analog-to-Digital Converter". *IEEE Journal of Solid-State Circuits*, vol. sc-22(no. 6):pp. 954–961, December 1987.

[15] S. H. Lewis. "Optimizing the Stage Resolution in Pipelined, Multistage, Analog-to-Digital Converters for Video-Rate Applications". *IEEE Trans. on Circuits and Systems-II*, vol. 39(no. 8):pp. 516–523, August 1992.

[16] S. Sutarja and P. R. Gray. "A Pipelined 13-bit, 250-ks/s, 5-V Analog-to-Digital Converter". *IEEE Journal of Solid-State Circuits*, vol. 23(no. 6):pp. 1316–1323, December 1988.

[17] B. Murmann and B. E. Boser. "A 12b 75MS/s Pipelined ADC using Open-Loop Residue Amplification". In *Proceedings of ISSCC*, 2003.

[18] T. B. Cho and P. R. Gray. "A 10 b, 20 Msample/s, 35 mW Pipeline A/D Converter". *IEEE Journal of Solid-State Circuits*, vol. 30(no. 3):pp. 166–172, March 1995.

[19] M. Fedeli. "OPAMP_RE_FIRST Functional Specification". Technical report, *STMicroelectronics®*, January 2003.

[20] P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer. *Analysis and Design of Analog Integrated Circuits*. John Wiley & Sons, Inc., forth edition, 2001.

[21] R. Gregorian and G. C. Temes. *Analog MOS Integrated Circuits for Signal Processing*. Wiley Series on Filters: Design, Manufacturing, and Applications. John Wiley & Sons, Inc., 1986.

[22] B. Pellegrini. "Considerations on the Feedback Theory". *Alta Frequenza*, vol. 41(no. 11):pp. 825–829, 1972.

[23] B. Pellegrini and P. Terreni. "Nuova teoria della reazione". *Alta Frequenza*, vol. 55(no. 2):pp. 95–105, 1986.

[24] S. M. Mallya and J. H. Nevin. "Design Procedures for a Fully Differential Folded-Cascode CMOS Operational Amplifier". *IEEE Journal of Solid-State Circuits*, vol. 24(no. 6):pp. 1737–1740, December 1989.

[25] J. Millman and A. Grabel. *Microelectronics*. McGraw-Hill, Inc., 1987.

[26] A. I. A. Cunha, M. C. Schneider, and C. Galup-Montoro. "An MOS Transistor Model for Analog Circuit Design". *IEEE Journal of Solid-State Circuits*, vol. 33(no. 10):pp. 1510–1519, October 1998.

[27] W. Liu, X. Jin, M. Chen, et al. *BSIM3v3.2.2 MOSFET Model-Users' Manual*. Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, 1999.

[28] Cadence Design Systems, Inc. *Affirma Spectre Circuit Simulator Reference*, June 2000.

[29] K. S. Kundert. "Introduction to RF Simulation and Its Application". *IEEE Journal of Solid-State Circuits*, vol. 34(no. 9):pp. 1298–1319, September 1999.

[30] F. De Bernardinis. "Heuristics for Performance Model Generation". Report, April 2003.

[31] H. Reid. "Console-Based Circuit Simulation in CADENCE Analog Artist: A Tutorial Introduction to `Ocean`". http://electron.mit.edu/∼ homer/misc/ocean/.

[32] Cadence Design Systems, Inc. *OCEAN Reference*, October 2001.