A Computer-Aided Design Tool for Analog Integrated Circuit Building Blocks Realization

Seyed-Abbas Sajjadi

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Electrical Engineering) at
Concordia University
Montreal, Quebec, Canada

March 2005

Your file Votre référence
ISBN: 0-494-04391-1
Our file Notre référence
ISBN: 0-494-04391-1

ABSTRACT

A Computer-Aided Design Tool for Analog Integrated Circuit Building Blocks Realization

Seyed-Abbas Sajjadi

With the advent of submicron CMOS process technologies, application of current-voltage model equation, which is good for long channel devices, is becoming futile. Shortage of analog integrated circuits design automation tools results in relying on engineering experience and time-consuming trial and error simulation runs to design an analog integrated circuit (IC). Many designers are faced with the large design time devoted to the design of analog circuits in comparing with digital counterparts in mixed-signal IC designs.

The research presented in this thesis aims to improve the efficiency of analog IC design process with a new design methodology and computer-aided design program for automating the realization of analog IC building blocks. The new design methodology is based on small-signal analysis, and DC simulation of NMOS and PMOS transistors which is predicted by sophisticated circuit analysis program i.e. HSPICE. The CAD tool is intended to be a design assistant and comprises of a number of modules including setup, single stage amplifier design, current source/mirror design, voltage divider design, differential amplifier design, operational amplifier and operational transconductance amplifier design modules. To demonstrate the usefulness and reliability of the new methodology and the tool, some design examples are presented.

This thesis also studies the analog design automation methods and tools that have been reported in the literature.

ACKNOWLEDGEMENTS

I would like to express my appreciation to my supervisors Dr. R. Raut and Dr. P. Sinha. Their support and advices were very helpful and are reflected throughout this thesis. The support and comments by my colleagues in the Electrical and Computer Engineering Department are acknowledged with thanks. I am very grateful to all my teachers, my father and mother for helping me fulfill my goals.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AHDL	Analog high-level Description Language
ASIC	Application specific Integrated Circuit
AWE	Asymptotic Waveform Evaluation
BJT	Bipolar Junction Transistor
BSIM	Berkeley Short-Channel IGFET Model
CAD	Computer-Aided Design
CADT	Concordia Analog Design Tool
CMOS	Complementary Metal Oxide Semiconductor
CD	Common-Drain
CG	Common-Gate
CS	Common-Source
EDA	Electronic Design Automation
g_{ds}	Drain-Source (Output) Conductance
g_m	MOS Transistor Transconductance
g_{mb}	Bulk Effect Transconductance
GUI	Graphic User Interface
HDL	Hardware Description Language
IC	Integrated Circuit
MOS	Metal-Oxide Semiconductor
MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor
NMOS	N-Channel MOS Transistor

OPAMP	Operational Amplifier
OTA	Operational Transconductance Amplifier
PMOS	P-Channel MOS Transistor
SOC	System on a Chip
SPICE	Simulation Program with Integrated Circuit Emphasis
V	Volt
μA	Micro Ampere
μm	Micro Meter
μW	Micro Watt

Chapter 1

Introduction

1.1 General

Invention of computers has changed our life and the way we have been doing tasks. With the help of computers, data processing and design tasks can be performed much faster than using hand analysis and traditional methods. Computer based design methodologies and Computer-Aided Design (CAD) tools have opened a new door of possibility to all engineers, especially digital and analog integrated circuits (ICs) designers.

There are many reasons for using a CAD tool in Microelectronics. The most potent driving force is competition in the semiconductor and electronics industry. In order to win business, companies use CAD tools to produce better designs more quickly and more cheaply than their competitors. Productivity is much improved by a CAD program enabling one to easily draw schematics and layouts, and run advanced features such as sensitivity analysis, mixed-mode simulation and so on.

In digital IC design, the design procedure is almost fully automated enabling semiconductor companies to produce advanced digital integrated circuit designs very

fast. But in analog IC design, the CAD tools are not well developed and analog designers are still doing the design tasks traditionally. In recent years, after a huge progress in developing digital design CAD tools, the academic and industry researchers are trying to develop more efficient and advanced design methodologies and CAD tools in analog IC design domain as well. Ultimately, this can lead to short time-to-market for a mixed-signal and analog custom-design projects.

1.2 Motivations

As the demand for mixed signal and system on a chip (SOC) integrated circuits increase, the design of CMOS analog integrated circuits becomes more critical. Many designers are faced with the large design time devoted to the design of analog circuits in comparison with digital counterparts.

The shortage of analog integrated circuits design automation tools results in relying on engineering experience and time-consuming trial and error method to design an analog integrated circuit. The more complex the circuit is and the less experience the designer has, the more is the time spent on trial and error simulation runs to find an acceptable solution. In many cases, because of the non-linear nature of analog circuit design, if initial guesses are far away from the true solution, at some point the designer may face situations that the trial and error method stops improving the performance and the required specifications can not be met.

With the advent of submicron CMOS technologies, application of voltage-current model equation, which is good for long channel devices, is becoming inaccurate. The behavior estimated using these formulae appear different from that predicted by sophisticated circuit analysis programs such as HSPICE. Using of circuit simulators in

analog CAD systems could be an effective design approach. Circuit simulators can accommodate application of more sophisticated set of formulae involving myriads of model parameters (i.e. BSIM models) and hence the results of simulations using these software programs are considered to be more realistic than those predicted with the approximate design equations.

As well, reviewing the literature of analog automation design tools appears that one common disadvantage of all developed tools is inefficient use of circuit simulators and the information that can be obtained from them. Most systems do not use a simulator at all during device sizing phase (equation-based systems) and only after the completion of circuit design. Other tools employ circuit simulators in a loop to incrementally design a circuit. In the first case, since the approximate design equations are not accurate enough, the simulation result of designed circuit usually differs from the required specifications. Some tools try to avoid (not to solve) the problem of equations inaccuracies by introducing fitting or over-design factors. In the second case, tools use circuit simulators to incrementally size the various devices with the help of numerical algorithms or expert systems. This approach increases the design time, because of using a large number of iterations.

A new design methodology based on simulation and knowledge-based approach that simplifies and speeds up the process of designing analog integrated circuits is expected to help analog designers tackle challenges as the supply voltage and transistor channel lengths scale down with each generation of CMOS technologies.

1.3 Objectives

The primary purpose of this work is to create and examine a new computer based design methodology, and to develop a CAD tool that automates the process of designing submicron analog integrated circuit building blocks.

The specific objectives are to contribute to the design of analog integrated circuit as follows:

1- To provide analog circuit designers with a tool so that they can concentrate on designing the more complex parts of a circuit, knowing that the CAD tool will enable them to design the some basic parts of their circuits automatically.

2 - To reduce the design cycle time of analog integrated circuits by providing accurate design solutions so that analog designers need not to run time consuming and trial and error simulations.

1.4 Organization of the Thesis

Chapter 2 includes a brief literature review of analog integrated circuit automation and developed CAD tools.

Chapter 3 introduces the basic physics and characteristics of Metal Oxide Semiconductor (MOS) devices and the concept of modeling of these transistors. Also, HSPICE as a simulation tool is explained briefly.

Chapter 4 explains the new design methodology and the CAD tool that have been developed based on this method for designing of analog integrated circuit building blocks.

Chapter 5 illustrates the design of some analog small scale building blocks, which contain two or three transistors, such as simple current sources/mirrors, voltage dividers, and single stage amplifiers.

Chapter 6 presents the design of several medium scale analog building blocks, which contain more than three and less than six transistors, such as cascode and differential amplifiers.

Chapter 7 deals with the design of two large scale analog building blocks, which contain more than six and less than twenty transistors, such as an Operational Amplifier (OPAMP) and an Operational Transconductance Amplifier (OTA) using the proposed design methodology and the new tool.

The simulation results are also presented for all design examples in chapter 5 through 7 to prove the accuracy and reliability of the new design methodology.

Chapter 8 presents conclusion and recommend actions for future work to improve the proposed design methodology.

Chapter 2

Literature Review

In this chapter, some issues regarding analog integrated circuit design, analog integrated circuit design procedure and its phases are presented. As well, analog IC design automation approaches that have been published in literature are reviewed.

2.1 A brief review of analog integrated circuit design and automation

The idea of building integrated circuit (IC) was introduced in 1960. In the following 43 years, the technology has progressed from producing simple chips containing a small number of devices to fabricating processors and memories with more than one billion transistors. The dimension of transistors has dropped from about 25 micrometers in 1960 to about 90 nanometers in the year 2003, resulting in a tremendous improvement in the speed of integrated circuits.

The Bipolar Junction Transistor (BJT) technology was the first to be used for small and medium sized integrated circuits and systems. Since the late 1970s, a particular type of transistor, Metal-Oxide Semiconductor Field-Effect Transistor (MOSFET) has been used in IC design. Later, Complementary Metal Oxide Semiconductor (CMOS) devices (using both n-type and p-type MOS transistors) were introduced for integrated circuits implementation. CMOS technologies rapidly captured the market. The principal

reasons were lower power consumption and higher packing density compared to BJT devices. Furthermore, CMOS circuits have a relatively simple manufacturing process and a lower fabrication cost. The possibility of placing both analog and digital circuits on the same chip to improve the overall performance and to reduce the cost of packaging made CMOS technology even more attractive.

In the past two decades, signal processing has moved from analog to digital domain. The growth of digital signal processing algorithms has resulted in a huge potential for signal processing power. Advances in integrated circuit technology and computer aided design (CAD) tools have provided fast and efficient implementation of these algorithms in MOS (Metal-Oxide Semiconductor) technology. While, digital signal processing is very powerful, there are certain applications such as processing of natural signals, wireless communication systems, biomedical instrumentation and robot sensing, where it is impossible to replace analog circuits with their digital counterparts. So, the demand for analog integrated circuits design still prevailed.

Economic and technical factors are encouraging designers to put analog and digital integrated circuits onto the same die and build mixed-signal chips. Application specific integrated circuits (ASIC), which are custom design chips, are moving towards the integration of complete systems onto a single chip (SOC). The growing demand for CMOS mixed-signal chip designs with the trends towards smaller feature sizes and higher scale of integration have brought new challenges in integrated circuit design and automation tools [1].

While the design of digital circuit is almost fully automated, analog CAD tools are far away from automation. In digital domain, electronic design automation (EDA)

tools allow designers describe circuit function in terms of hardware description language (HDL) codes and then compile these codes into a set of standard logical gates (synthesis). The software also generates the physical layout automatically, resulting in reduced design cycle time. But, in analog domain designers carry out design tasks traditionally. They design circuits by performing small-signal hand analysis, draw schematics, and connect circuit blocks so that the design specifications can be achieved. Analog design automation tools have not yet progressed to the stage of producing a design by accepting coded specifications or to convert circuit schematic into a physical layout. Therefore, there is a huge difference in the design time of analog circuits versus digital counterparts. The lack of advanced analog integrated circuits design CAD tools results in relying on engineering experience and time-consuming trial and error method to design an analog integrated circuit. The more complex the circuit is and the less experience the designer has, the more is the time spent on trial and error simulation runs to find an acceptable solution. In many cases, because of the non-linear nature of analog circuit design, if initial guesses are far away from the true solution, at some point the designer may face situations that the trial and error method stops improving the performance and the required specifications can not be met [2].

In response to this issue, academic and industry sectors are moving forward to developing new design methodologies and tools for analog integrated circuits design. The analog design CAD tools are able to improve analog IC design process in the following ways:

1) By decreasing design cycle time
2) By simplifying the design process

3) By improving the likelihood of error-free designs

4) By reducing design and production cost

5) By improving manufacturing yield

6) By allowing easier track of fabrication process

7) By retaining expert design knowledge

To develop a CAD tool or design methodology to automate analog IC design process, it is essential to first study that process in its real form. Analog circuit design is a knowledge-intensive, multiphase, iterative task that takes a significant period of time. Fig. 2.1 shows various phase and iterative nature of analog IC design process. In the following, we will describe each phase in more detail.

1) Circuit synthesis

Analog circuit synthesis involves, selecting the right circuit architecture to fulfill the proposed application. First in this step, designers choose a circuit architecture that could meet the required specifications. Then, they use simple analytic design equations to predict circuit behavior and to relate specification to appropriate circuit design parameters. Special techniques have been developed, such as small-signal modeling and z-parameter analysis method that are used by circuit designers to drive design equation quickly and without significant loss of accuracy.

Analog designers may also benefit from algorithmic knowledge in the form of proven, step-by-step design strategies that they have developed in previous design sessions. For instance, the order of designing of circuit blocks should be selected so that minimizes possible performance conflicts.

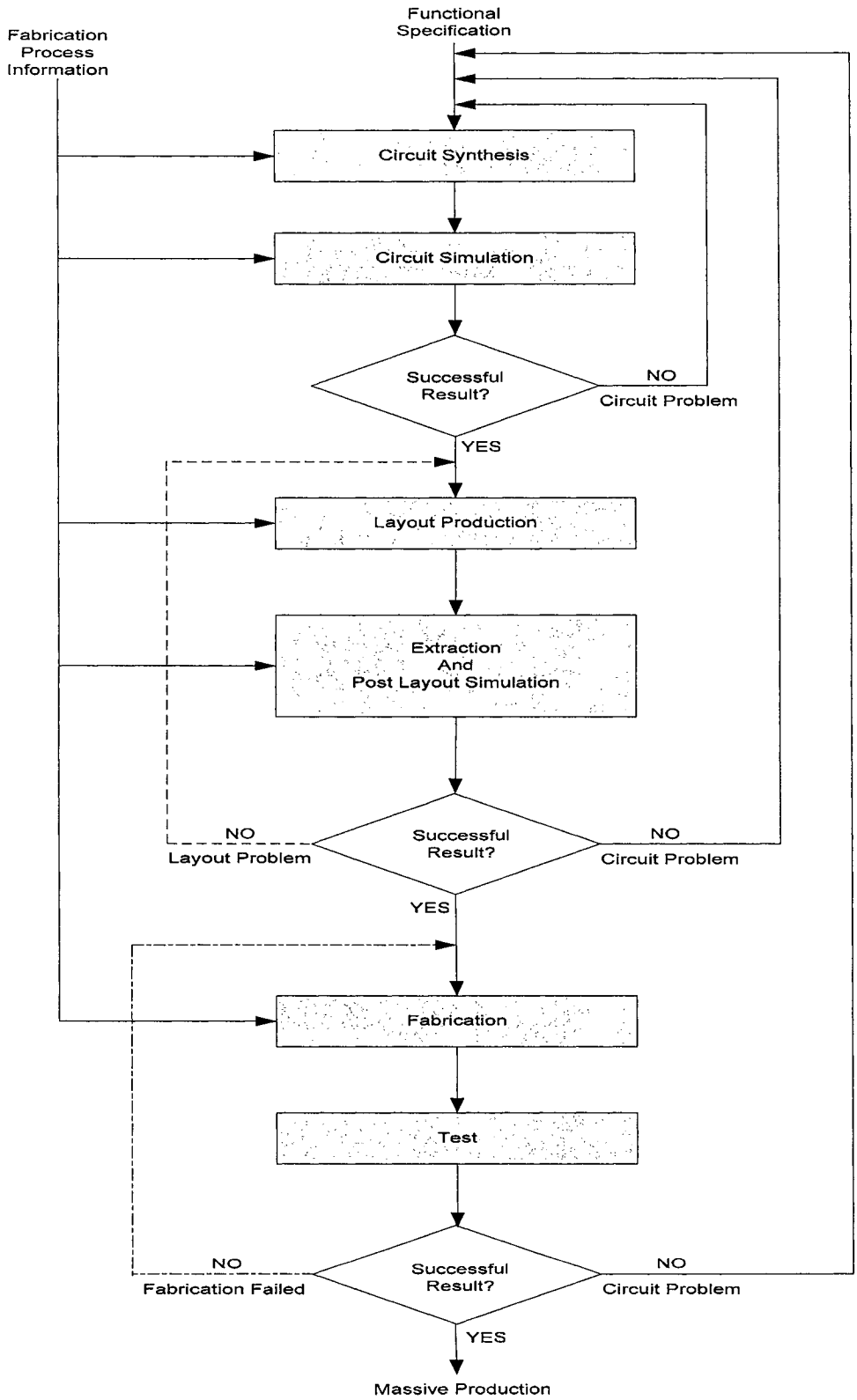


Figure 2.1: Analog IC design process

The other important aid to the design synthesis process is knowledge of circuit heuristics. For example, knowing that the small-signal gain of MOS transistor decreases by increasing its bias current is one typical circuit heuristics applicable to analog CMOS circuit design.

As well, the design of large circuit structures can be performed by dividing the circuit into smaller blocks and then assign specifications to each one of these blocks. The partitioning of the circuit into smaller blocks is called hierarchical design. For instance, an operational amplifier can be decomposed to smaller blocks including differential amplifier, common-source amplifier, and bias circuits.

2) Circuit simulation

Simulation is a kind of verification. Circuit simulation, performed by simulation tools such as HSPICE, is an essential part of the design process. It is the only way to predict accurately the behavior of analog integrated circuits prior to fabrication. Since a numerical simulator contains more sophisticated models of device behavior, it is likely to predict a circuit performance that differs from the initially expected performance. At this point, designers spend a considerable amount of time running multiple simulations and adjusting circuits design variables in order to improve the performance of the circuit and to converge on the desired specification. This trial and error process is lengthy and depends on designer's expertise.

3) Layout production

After successful simulation of circuit, layout (placement and routing) generation is the next major stage in analog IC design process. Layout production refers to physical and geometrical description of a circuit onto silicon die. Analog circuits demand careful

layout procedure to take care of issues such as mismatches, noise, and crosstalk. These can considerably affect the circuit performance.

4) Extraction and post layout simulation

Extraction is the process of extracting circuit information and parasitic components from the circuit's layout that can be done by CAD tools. Since parasitic components such as undesirable capacitances and resistances introduced by layout can deteriorate the performance of the analog circuit, always a performance verification phase precedes fabrication, which is called post layout simulation. Designer can modify layout to reduce the effects of parasitic components. If this can not resolve the errors then the designer may have to modify the circuit itself.

5) Fabrication

After circuit and layout have been designed, the next step is fabricating the proposed IC. A simplified fabrication process is a combination of the following operations: 1. wafer processing to provide the proper substrate, 2. photolithography to precisely define each region, 3. oxidation, deposition, and ion implantation to add materials to the wafer, 4. etching to remove materials from the wafer. Many of these steps require "heat treatment" i.e. the wafer must undergo a thermal cycle inside a furnace [3].

6) Test

Testing is an important operation before releasing the chip for mass production or returning it back to the design team for more improvements of the layout or its circuit. In analog test, every transistor or each subsystem on the chip needs to be tested for behavior and performance. Currently, the requirements of testing mixed-signal and system-on-chip designs are challenging research areas for industrial and academic researchers.

2.2 A Review of analog IC design automation approaches

We shall start with the distinction between a CAD tool and a design automation tool. A CAD tool is a computer-based system that provides assistance to a design task. This assistance can range from releasing a designer from long, frustrating and/or error prone tasks of the design process to performing complete designs with minimal human intervention. These latter classes of CAD tools that automate part or the whole of the design process are referred to as design automation tools [1].

The design automation problem can be divided into two major parts: verification and synthesis. Verification is just simulation that can be done by analog, digital or mixed-signal simulators. Synthesis includes both circuit design and physical layout generation. So far, many approaches have been reported in literature for simulating and synthesis of analog integrated circuits [4]-[10]. A classification of these approaches is shown in Fig. 2.2.

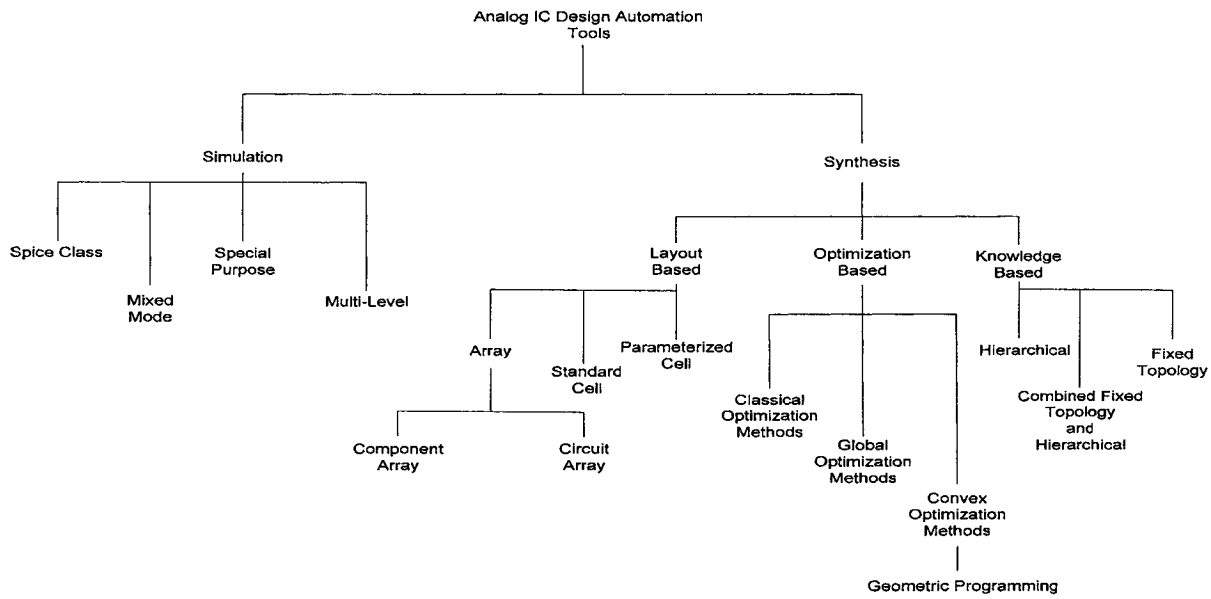


Figure 2.2: Classification of analog IC design automation tool approaches [1]

2.2.1 Verification by simulation

Simulation is a kind of verification and an essential part of analog integrated circuit design process. Both commercial and academic simulators are available. We can roughly classify simulators into four types as follows:

Spice-class: The first type of analog simulators developed for analog design problems. Recent research works are focused on improving capacity (netlist size), device model accuracy, and numerical robustness of these tools. HSPICE and PSPICE are two well-known commercial tools in this area.

Mixed-mode: These commercial simulation tools are based on multiple-engine designs, in which an analog and a digital simulator communicate over a backplane that connects them, converts digital and analog values, and controls which simulator to run, and when. Current research works are aimed at improvements of performance and accuracy, and integration of user-supplied simulators into the backplane.

Special-purpose: The special type of simulation tools employs efficient simulation algorithms for specific applications such as switched-capacitor systems and over-sampled data converters. Asymptotic Waveform Evaluation (AWE) is a typical technique that has been used in many verification and synthesis tools [11].

Multi-level: Sometimes it is desirable to model one part of a system with full electrical detail and another part with a simple behavioral model. The idea here is using behavioral simulation that is, replacing a part with its abstract behavioral equivalent. There is some progress in analog high-level description language (AHDL) for certain analog applications. Saber [12] and iMACSIM [13] are two simulators in this area. We

expect to see substantial progress on adapting high-level description languages to analog domain in the near future.

2.2.2 Synthesis

Synthesis can be referred to as circuit or physical layout synthesis. Circuit synthesis is known as creating a circuit topology, and finding the sizes and biasing points of its devices so that required specifications can be met. Layout synthesis is, transforming the synthesized circuit into silicon die. So far, three distinct synthesis approaches are being used in CAD tools, including optimization-based, layout-based, and knowledge-based methods.

2.2.2.1 Optimization-based design approach

The first developed analog CAD tools were optimization-based. These tools consider the device sizing of an analog circuit as an optimization problem. These accept user constraints, goals, and starting points, and then employ optimization algorithms to iteratively find device sizes in order to meet the entered specifications. In simulation-based optimization approaches, a simulator is used within the optimization loop to evaluate circuit performance.

This optimization algorithms fall under three groups: classical, global, and convex optimization.

Classical optimization methods, such as steepest descent, sequential quadratic programming and Lagrange multiplier approaches, have been used in different CAD tools, namely DELIGHT-SPICE [14], ECSTACY [15], and ADOPT [16]. One disadvantage of classical methods is that they only find locally optimal designs. This

means that we may find other set of design parameters that result in a better performance. The other disadvantage is that the classical optimization approach may fail to find a feasible design, even if one exists.

Global optimization methods, such as branch and bound, and simulation annealing have also been used in different CAD tools. The main disadvantage of these methods is that they are very slow. The ASTR/OBLX [17] and OPTIMAN [18] have used simulation annealing method to find a global optimal solution.

In convex optimization method, we minimize a convex objective function subject to linear equality constraints, and inequality constraints that are expressed as upper bounds on convex functions [19]. The special type of convex optimization methods, geometric programming, has been used in more recent CAD tool, GPCAD [20]. The advantage of convex optimization methods is that they can solve large problems, with thousands variables and constraints, very fast and efficiently. The other important features of these methods are that the global solution is always found, regardless of the starting point and the infeasible problem is proved by producing a certificate.

2.2.2.2 Layout-based design approach

CAD tools have followed this design approach to provide a direct path to layout generation. Layout-based design approach is based on standard cells, gate arrays and parameterized cell methods that are widely used in digital domain [21].

Analog arrays consist of different pre-designed and laid-out building blocks, ranging from single-component arrays to circuit arrays. The required functions can be obtained by proper programming of one or more levels of interconnect. SLIDE and

CAPSIZE , developed by Silicon Systems Inc., have used switched capacitor filter arrays for the design of complex filters with more than 50 poles without requiring any external component [4]. Using analog arrays has several drawbacks. The most important one is that they do not provide the required design flexibility for high performance analog integrated circuit designs. They also restrict the designer to choose from available active and passive component arrays with a limited range of component values. Furthermore, they are not cost-effective in terms of silicon usage.

Analog standard cells approach is the same as analog arrays, but more efficient in terms of silicon usage, since the necessary cells are only used to implement a specific function. Using standard cells in analog design is not as easy as digital domain, because it is very difficult to keep and provide a large number of analog standard cells in the library of analog CAD tools so as to achieve a wide range of possible specifications. Several CAD tools that have used this approach are reported in [22]-[24].

Analog parameterized cells approach is another layout-based design method. Analog parameterized cells are similar to analog standard cells, but they can provide more design flexibility by allowing the modification of cells according to the design requirements. The degree of flexibility directly depends on the complexity of module generator. Module generator is a program that generates the layout of the cells with given a set of input parameters. So the components can have a wider range of values that is an important feature not offered by other layout-based approaches. AIDE2 [5] and CONCORD [25] have been developed based on this approach.

It is important to note that all of layout-based design approaches suffer from tracking fabrication process.

2.2.2.3 Knowledge-based design approach

During recent years, there have been intensive interests in knowledge-based approaches to overcome the various deficiencies of the previous methods. Knowledge-based methods benefit from domain knowledge (design equations and heuristics) to design analog integrated circuits, and address the design task in a full-custom way. So, they provide maximum design flexibility for achieving better performances.

So far, hierarchical, fixed-topology, and combination of hierarchical and fixed-topology knowledge-based design approaches have been introduced.

The hierarchical design approach has been used to automate digital circuit design successfully. The same idea has come to the analog domain. The hierarchical design involves breaking of a circuit or system into smaller distinct parts or blocks. Each of these parts or blocks is assigned a set of specifications so that overall circuit or system specification can be met. The same procedure can be repeated for each part or block as far as possible. The number of hierarchical levels depends on the complexity of the proposed circuit or system. To be able to partition a circuit to smaller parts and to assign new specifications to each part, a great deal of domain knowledge in the form of design equations and heuristics are required. Hierarchical design systems are easier to expand and provide a better use of existing design knowledge and design flexibility. Several systems such as PROSAIC [6], BLADES [7], and OASYS [26] have been developed based on hierarchical design approach.

The fixed-topology approach is another knowledge-based design philosophy that differs widely from the hierarchical approach. In this design method, fixed, unsized,

device level circuit topologies are kept in the system library in terms of knowledge base, together with the necessary domain information for computing the device sizes. The nature of domain information depends on the method of computing the size of devices. Some of the systems that follow this approach are IDAC [27], OPASYN [28], and OAC [29]. The fixed topology design method has the least design flexibility among knowledge-based approaches.

The combined hierarchical and fixed topology approach combines features of both hierarchical and fixed topology methods. This design method provides design flexibility less than hierarchical approach and more than fixed topology method. ASAIC [30] and CAMP [31] are two design systems that fit into this class. ASAIC builds a circuit topology in a hierarchical method and then design the topology in a manner similar to fixed topology systems. CAMP is a system that designs a circuit by applying fixed topology method and then by modifying its building blocks to meet the desired specifications.

More recent analog IC design CAD system is a new Procedural Analog Design (PAD) tool that presents a new interactive knowledge-based design methodology for transistor sizing and layout generation. The present version of PAD covers the procedural design of transconductance amplifier (OTA) and operational amplifier (OPAMP) topologies [32].

2.3 Summary

Analog CMOS integrated circuit design procedure and analog IC design automation approaches have been discussed in this chapter. We found that there are three distinguished circuit synthesis approaches: layout-based, optimization-based and

knowledge-based. Advantages and disadvantages of these methods have been discussed. Our study showed that analog IC design methodologies and tools need designer intervention.

As a result, we expect to see more emphasis on complete cell-level synthesis process, coupling both circuit design and layout in the future.

Our design methodology is a mixed knowledge-based and simulation-based approach for the design of analog building blocks at the transistor-level. This design methodology is described in chapter 4.

Chapter 3

Some Basic Concepts Related to Our Work

This chapter gives the reader a brief review of basic concepts of analog IC design including MOS devices physics, structure, models, and current-voltage characteristics.

3.1 Introduction

In analog integrated circuit design, a solid understanding of semiconductor devices is necessary because transistors are not considered as simple switches and many of second-order effects directly impact the performance. As the device sizes scale down generation by generation, these effects become more important. To address this issue, researchers have developed advanced models to represent the behavior of short-channel devices in simulation tools, such as HSPICE.

Here, we present a brief review of basic concepts in the design of analog CMOS integrated circuits. These concepts provide the reader with required background and include the structure of MOS transistors, the current-voltage (I/V) characteristics and different operating regions of the device. As well, second-order effects including body effect, channel-length modulation, and sub-threshold conduction are explained. The MOS small-signal and SPICE models, HSPICE simulation software, a sample HSPICE input (net-list) and output files are also presented.

3.2 Metal-Oxide Semiconductor (MOS) device physics and structure

Fig. 3.1 shows the physical structure of an n-channel MOS transistor (NMOS). The device is fabricated on a p-type substrate called the Body or Bulk (B) that provides physical support for the device. There are two heavily doped n^+ regions forming the Source (S) and Drain (D) terminals, a heavily doped conductive piece of poly-silicon operating as the Gate (G), and a thin layer of silicon dioxide (SiO_2) insulating the gate from substrate. The structure is symmetric with respect to Source and Drain. The distance between the Drain and Source is called the channel length, L, and that perpendicular to the channel length is called the channel width, W. Note that L and W are effective channel length and width respectively in Fig. 3.1 and in all equations presented here.

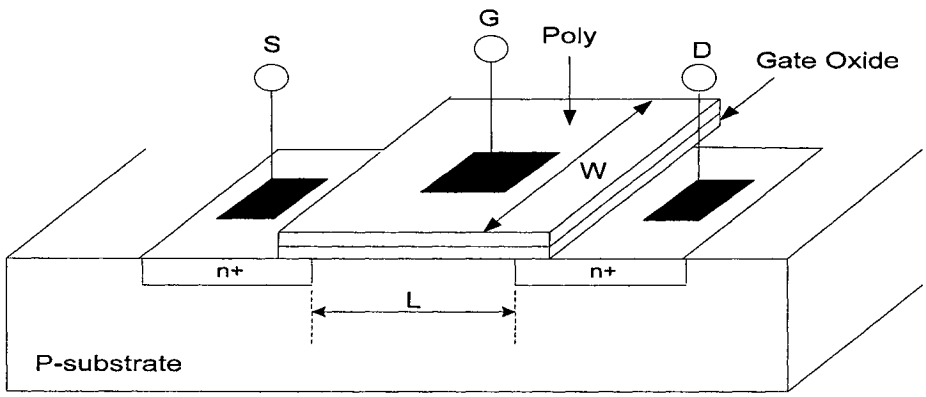


Figure 3.1: Structure of a MOS transistor (NMOS)

In practice, the MOS transistor is a four-terminal device and the substrate potential influences the device characteristics. In CMOS technology, both PMOS and NMOS transistors are used. PMOS and NMOS transistor must be fabricated on the same substrate. Thus, one device type can be placed in a local substrate, called "well".

In today's CMOS processes, the PMOS device is placed in an n-well, as shown in Fig. 3.2.

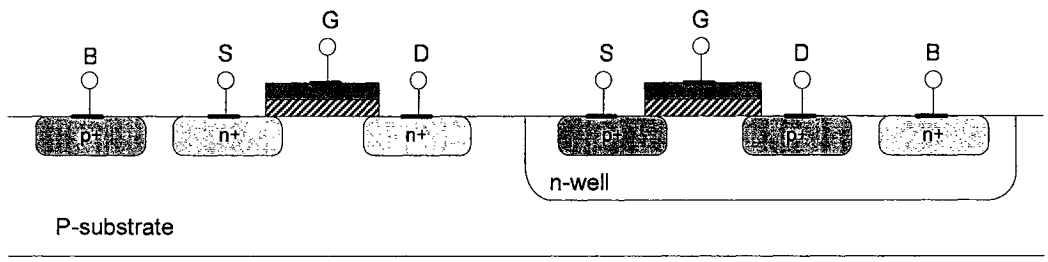


Figure 3.2: NMOS and PMOS transistor on the same substrate

The following circuit symbols are used to represent NMOS and PMOS transistors in this thesis. In many applications, the Body (B) terminal of NMOS and PMOS transistors are tied to the most negative and positive supply voltages respectively.

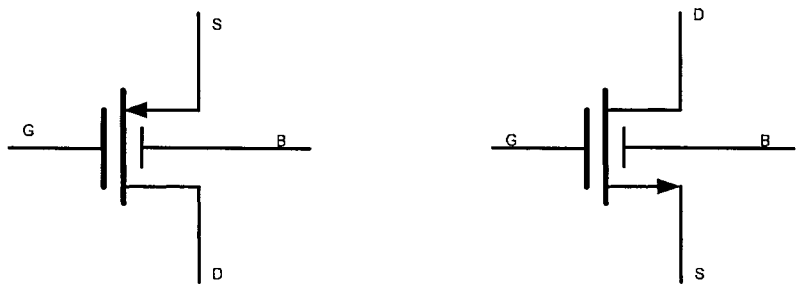


Figure 3.3: The symbol of MOS transistors

3.3 MOS transistor current-voltage (I/V) characteristics

MOS transistor is a voltage-controlled device. The voltage between the Gate (G) and Source (S) terminals (V_{GS}) controls the current flow in the third terminal (I_D). Here, we ignore the effect of the Body (B) on device operation and consider the MOSFET as a three-terminal component.

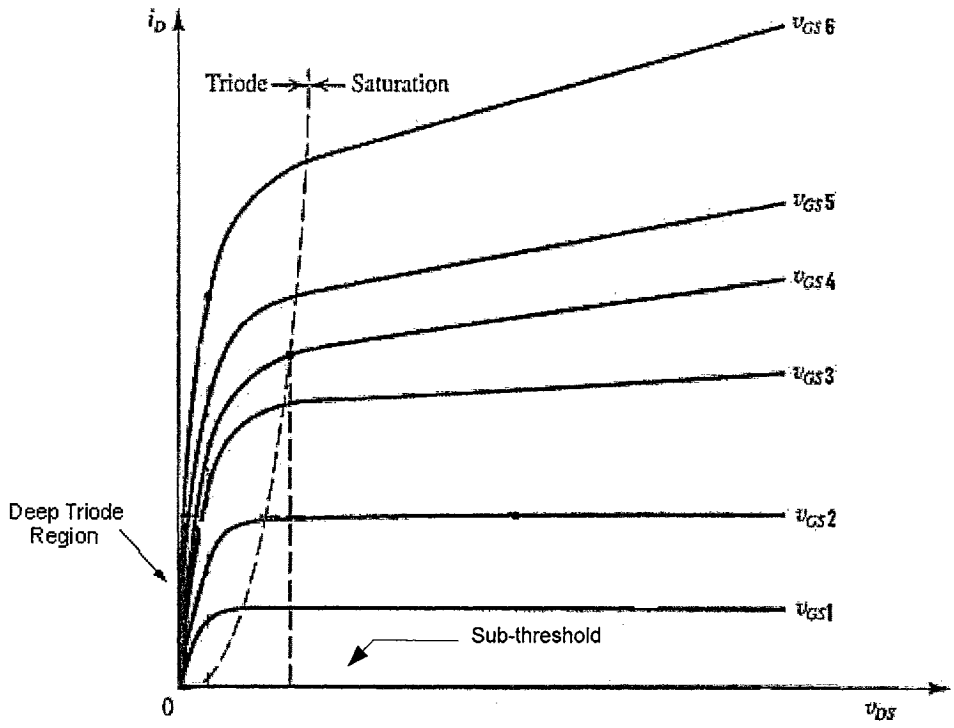


Figure 3.4: A typical current-voltage characteristics of a MOS device

Fig. 3.4, shows the output I-V characteristics of a typical MOS transistor. There are three distinct regions of operation that can be described as follows:

1- Sub-threshold Conduction Region ($V_{GS}<V_{TH}$)

$$I_D \cong 0 \tag{3.1}$$

2- Triode Region ($V_{DS}<V_{GS}-V_{TH}$ and $V_{GS}> V_{TH}$)

$$I_D=\mu_n c_{ox} \frac{W}{L}[(V_{GS}-V_t)V_{DS}-\frac{1}{2}V_{DS}^2] \tag{3.2}$$

3- Saturation Region ($V_{DS}>V_{GS}-V_{TH}$ and $V_{GS}>V_{TH}$)

$$I_D=\frac{\mu_n C_{ox}}{2}\times\frac{W}{L}(V_{GS}-V_{TH})^2(1+\lambda V_{DS}) \tag{3.3}$$

We call V_{TH} the "threshold voltage", $V_{GS} - V_{TH}$ the "overdrive voltage", W/L the "aspect ratio", λ the "channel-length modulation factor", and $\mu_n C_{ox}$ or K_n (K_p for PMOS) the "transconductance parameter". V_{TH}, λ, and K_n (K_p) are characteristic of the process. Also, μ_n is the mobility of electrons and C_{ox} is the gate oxide capacitance per unit area. The above equations are known as level-one model (square-law) equations in SPICE simulation. The square-law model is most common since it contains most features without being complex.

It is clear from the plot of Fig. 3.4 that for small $V_{DS} << 2(V_{GS}-V_{TH})$ i.e. deep triode region that MOSFET operates as a linear resistor whose value is controlled by the overdrive voltage.

$$R_{on} = \frac{1}{\mu_n C_{ox} \frac{W}{L}(V_{GS}-V_{TH})} \tag{3.4}$$

Also, the gain factor of NMOS and PMOS transistors are defined as follows:

$$\beta_n = K_n \times \frac{W}{L} \tag{3.5}$$

$$\beta_p = K_p \times \frac{W}{L} \tag{3.6}$$

Since a MOSFET operating in saturation produces a current controlled by its gate-source overdrive voltage, one may introduce a factor of merit that indicates how well a device converts a voltage to a current. We define the figure of merit as the change in the drain current divided by the change in the gate-source voltage and call it "Trans-conductance". This quantity, denoted by gm, is expressed as follows:

$$g_m = \frac{\partial I_D}{\partial V_{GS}}\Big|_{V_{DS}=\text{const.}} = \frac{I_{D_2} - I_{D_1}}{V_{GS_2} - V_{GS_1}}\Big|_{V_{DS}=\text{const.}} \tag{3.7}$$

In the saturation region, g_m can be defined as follows:

$$g_m = \mu_n C_{ox} \frac{W}{L}(V_{GS} - V_{TH})(1 + \lambda V_{DS}) \tag{3.8}$$

$$= \sqrt{\frac{2\mu_n C_{ox}(W/L) I_D}{1+\lambda V_{DS}}} \tag{3.9}$$

g_m represents the sensitivity of the device: for a high g_m, a small change in V_{gs} results in a large change in I_D. The g_m can also expressed as

$$g_m = \frac{2I_D}{V_{GS} - V_{TH}} \tag{3.10}$$

The concept of transconductance can also be applied to a device operating in the triode region as:

$$g_m = \mu_n C_{ox} \frac{W}{L} V_{DS} \tag{3.11}$$

Fig. 3.5 shows the real current-voltage characteristics of a 0.18-μm NMOS transistor with L = W = 1 μm. It is instructive to note how the drain-source current changes with changes in V_{GS} and V_{DS} values in the saturation and triode regions.

The transconductance drops if the device enters the triode region. Thus, we should bias MOS transistors in the saturation region for amplification.

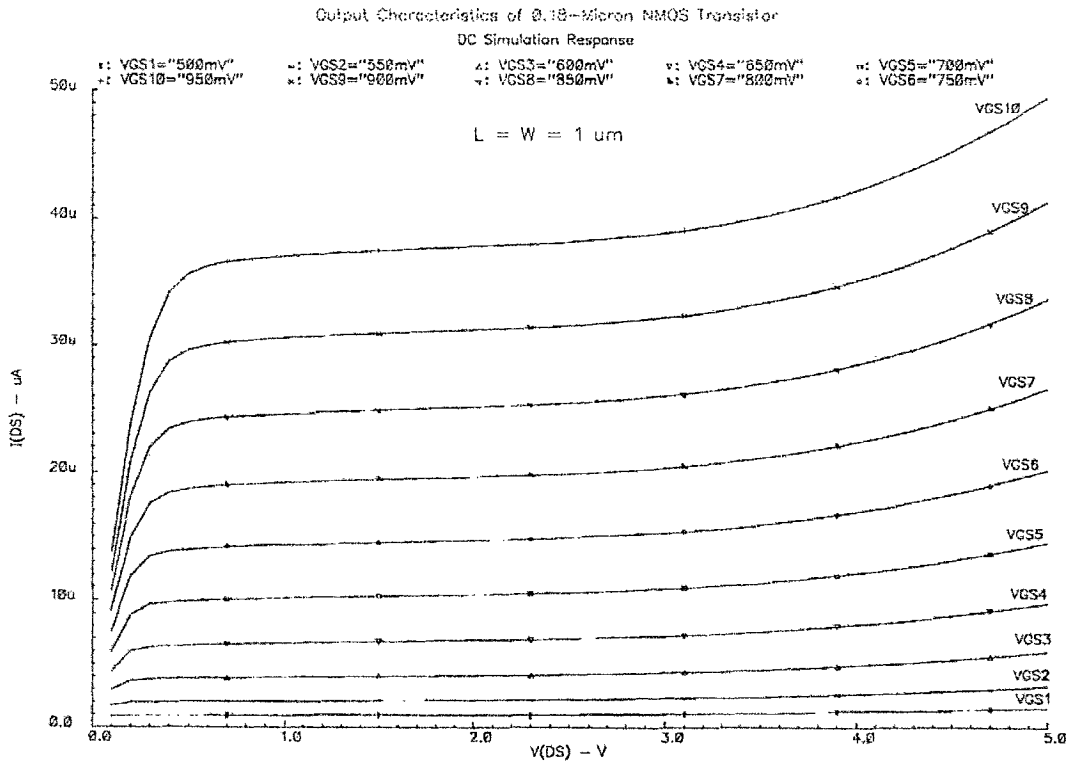


Figure 3.5: The current-voltage characteristics of a 0.18 μm NMOS transistor

3.4 Second-order effects

In this section, we describe three second-order effects that are important in analog integrated circuit analysis.

a) Body Effect

In many applications, the bulk and the source are not tied to each other. As the potential difference between them, V_{BS}, increases, V_{TH} increases for both the NMOS and PMOS transistors as expressed in equation (3.12). This is called "body effect".

$$V_{TH} = V_{TH0} + \gamma \left(\sqrt{|2\varphi_F - V_{BS}|} - \sqrt{|2\varphi_F|}\right) \tag{3.12}$$

Where V_{TH0} is the threshold voltage when $V_{BS}=0$, $\gamma=\sqrt{2q\varepsilon_{si}N_{sub}}/C_{ox}$ is the body effect coefficient, $\varphi_F=(kT/q)\ln(N_{sub}/n_i)$ and V_{BS} is the body-source potential difference [33].

b) Channel-Length Modulation

The actual length of channel gradually decreases as the potential difference between the gate and the drain increases. This effect is called "channel-modulation effect". Assuming a first-order relationship between $\frac{\Delta L}{L}$ and V_{DS} such as $\frac{\Delta L}{L}=\lambda V_{DS}$, we obtain the equation (3.3) as mentioned above, in the saturation region,

$$I_D=\frac{\mu_n C_{ox}}{2}\times\frac{W}{L}(V_{GS}-V_{TH})^2(1+\lambda V_{DS}) \qquad (3.13)$$

Where, λ is the channel-length modulation coefficient. This phenomenon results in a nonzero slope in I/V characteristics and hence a non-ideal current source between D and S in saturation as shown in Fig. 3.4. For longer channels, λ is smaller.

c) Sub-threshold V_{DS} Conduction

In practice, the MOSFET does not turn off abruptly as V_{GS} drops below V_{TH}. A "weak" inversion layer exists in the channel and some currents flows from the drain to the source. This phenomenon is called "sub-threshold conduction" or "weak inversion".

3.5 MOS small-signal models

A small-signal model is an approximation of the large-signal model around the DC operating point. This model is usually used to simplify the calculations and to analyze a circuit at low or high frequency. To obtain a small-signal low-frequency model for MOS transistors, we consider the effect of changes of one bias parameter in other bias

parameters. For instance, the drain current is a function of the gate-source voltage; we can model this relation with a voltage-dependent current source equal to $g_m v_{gs}$. Due to channel-length modulation, the drain current also varies with the drain-source voltage linearly. This effect can be modeled by a linear resistor equal to r_{ds} or by a linear conductance equal to g_{ds}, connected between D and S, and expressed as:

$$g_{ds} = \frac{\partial I_D}{\partial V_{DS}}|_{V_{GS}=const.} = \frac{I_{D_2} - I_{D_1}}{V_{DS_2} - V_{DS_1}}|_{V_{GS}=const.} \tag{3.14}$$

The output conductance, g_{ds} decreases with increasing incremental drain-source voltage and approaches zero as the device is operated in saturation.

The low-frequency impedance between G and S is very high and represented as an open circuit. In addition, the bulk potential influences the threshold voltage and hence the gate-source overdrive voltage (body effect). The bulk behaves as a second gate terminal. We can model this dependence with a current source connected between D and S, and equal to $g_{mb} v_{bs}$. g_{mb} Can be expressed as follows:

$$g_{mb} = \frac{\partial I_D}{\partial V_{BS}}|_{V_{DS}, V_{GS}=const.} = \frac{I_{D_2} - I_{D_1}}{V_{BS_2} - V_{BS_1}}|_{V_{DS}, V_{GS}=const.} \tag{3.15}$$

In the saturation region, we have:

$$g_{ds} = \frac{1}{2}\mu_n C_{ox} \frac{W}{L}(V_{GS} - V_{TH})^2.\lambda = \lambda.I_D \tag{3.16}$$

$$g_{mb} = \mu_n C_{ox} \frac{W}{L}(V_{GS} - V_{TH})(1 + \lambda V_{DS})(\frac{\partial V_{TH}}{\partial V_{BS}}) \tag{3.17}$$

Using equation (3.8) and (3.12), we obtain:

$$g_{mb} = g_m \times \frac{\gamma}{2\sqrt{2\varphi_F + V_{BS}}} \tag{3.18}$$

$$= g_m \times \eta \tag{3.19}$$

According to equation (3.18), g_{mb} is proportional to γ as we expect.

The MOS low-frequency small-signal equivalent circuit can be modeled as depicted in Fig. 3.6. This model is adequate for most low-frequency small-signal analysis.

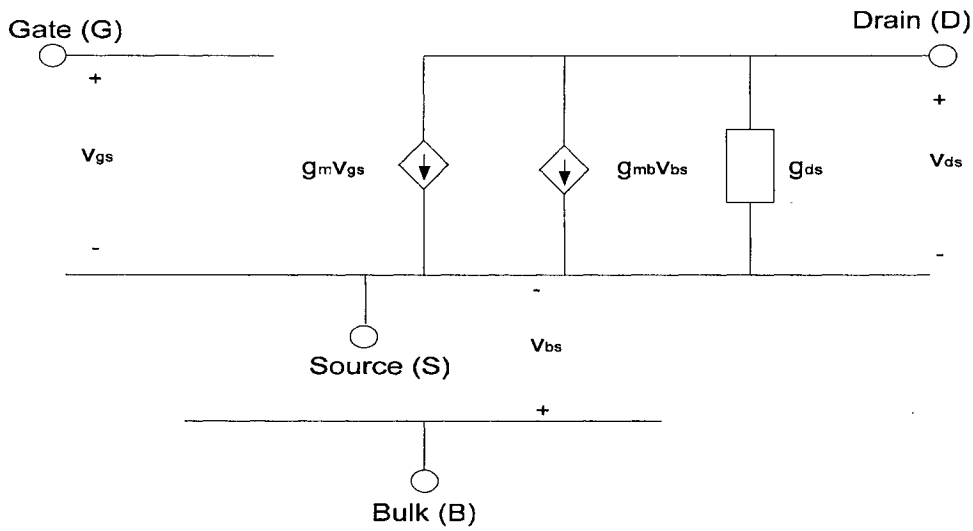


Figure 3.6: The low-frequency small-signal model

This is instructive to note that $g_m v_{gs}$ and $g_{mb} v_{bs}$ have the same polarity, i.e. raising the gate voltage has the same effect as raising the bulk potential. For high frequencies the parasitic capacitances that exist between every two of the four terminals of a MOS transistor need be considered. Therefore, a more complete MOS small-signal model of the MOS device becomes as shown in Fig. 3.7.

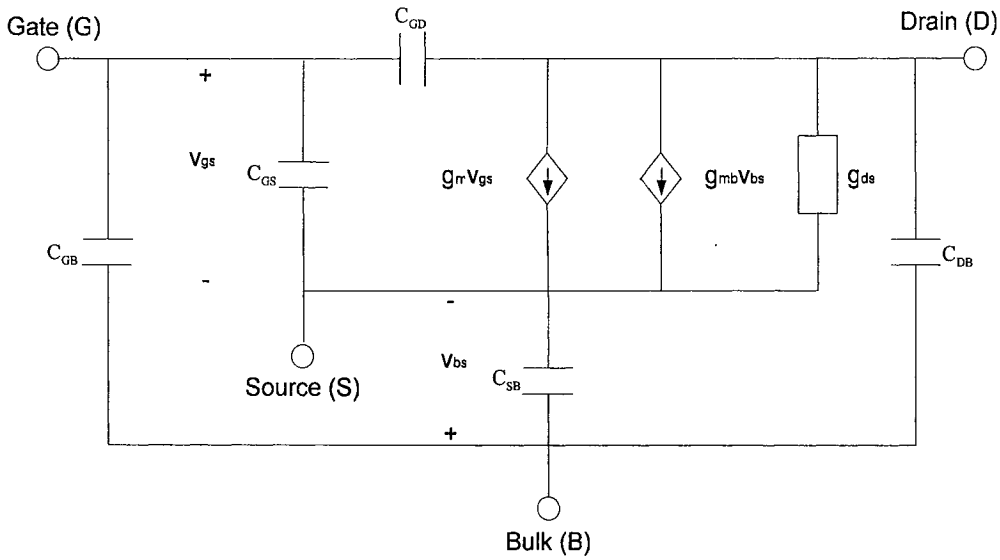


Figure 3.7: The complete small-signal model

3.6 Scaling theory and short-channel effects

One of the major advantages of CMOS technology is the tremendous potential of scaling of MOS transistors. The ideal scaling theory follows three rules: (1) reduce all lateral and vertical dimensions by α (α>1); (2) reduce the threshold voltage and the supply voltage by α; (3) increase all of doping levels by α. The ideal scaling of MOS transistor has the following effects on the device and analog circuits:

- All the electric field in the transistor remain constant
- Current capability of the transistor drops by a factor of α
- The total channel capacitance decrease by α
- The supply voltage reduces by α
- The maximum allowable voltage swings decrease by a factor of α
- The power dissipation drops by α^2

In practice, technology scaling has deviated from the ideal scaling considerably. The supply voltage and threshold voltage have not scaled as rapidly as device dimensions. For example, V_{DD} has decreased from 5 V to 2.5 V and V_{TH} from 0.8 V to 0.4 V as minimum channel length has dropped from 1 μm to 0.25 μm. Furthermore, many short-channel effects have influenced the transistors, making it difficult to obtain all of the benefits that could be achieved with ideal scaling.

The square-law model of the MOS device, mentioned in section 3.3, is simple to use for hand calculations but ignores many important second-order effects. Most of these second–order effects are due to short-channel dimensions (less than 3 μm). Therefore, we should consider a more complex model that is suitable for representing short-channel devices. For example, the BSIM3v3 model provides good performance when applied to analog circuit simulation and addresses the following important effects seen in deep-submicron MOSFET operation [3]:

- Threshold voltage reduction
- Mobility degradation due to a vertical field
- Velocity saturation effects
- Drain-induced barrier lowering
- Channel length modulation
- Sub-threshold (weak inversion) conduction
- Parasitic resistance in the source and drain
- Hot-electron effects on output resistance

Incorporation of the short-channel effects in MOS models makes those models so complicated that cannot be used for hand analysis by analog designers. On the other hand, simple models are not accurate enough to determine the design parameters of the short-channel designs properly. Therefore, a new design method to resolve this problem could be very desirable. This new methodology will be explained in the next chapter.

3.7 MOS SPICE models

MOS modeling has made tremendous progress on improving the accuracy of models as device dimensions scale down to submicron. In order to predict the behavior of an analog circuit properly, SPICE based simulators require an accurate model of the MOS transistors in circuit simulations. So far, many MOS models with different features and accuracies for long and short channel devices have been introduced. Selection of a model for NMOS and PMOS transistors depends on the process technology. The model itself has to be provided by the IC manufacturer. In the following, we review some of these models and their features.

The simplest MOS transistor model is level 1 that includes 16 parameters. Level 1 model does not include sub-threshold conduction or any short channel effects, and is based on equations (3.2) and (3.3). This level maintains reasonable I/V accuracy for channel lengths as small as $4\,\mu m$. But it still predicts the output impedance of transistor in saturation quite poorly [34].

The level 2 model was developed to predict many high-order effects as channel lengths fell below approximately $4\,\mu m$. Measured data show that the level 2 model provides reasonable I/V accuracy for wide, short devices in the saturation region with

$L \approx 0.7$ μm but it suffers from substantial error in representing the output impedance and the transition point between saturation and triode regions. For narrow or long devices, the model is quite inaccurate [34].

The level 3 model realization is almost similar to the level 2 model, with some equations simplified and many empirical constants introduced to improve the accuracy for channel lengths as small as $1\,\mu m$. An important drawback of this model is the discontinuity of the derivative of I_D with respect to V_{DS} at the edge of the triode region, leading to large errors in the calculation of the output impedance [34].

Another well-known MOS model, used by chip designers more than any other model, is the Berkeley Short-Channel IGFET Model (BSIM) [3]. Over the past decade, BSIM models have been established themselves as the de facto standard MOSFET SPICE model for circuit simulation and CMOS technology development.

The philosophy behind the level 1 to 3 models was to express the device behavior by means of equations that originated from the physical operation. BSIM model adopted a new approach. It introduced numerous empirical parameters to simplify the equations. One interesting feature of BSIM is the addition of a simple equation to represent the geometry dependence of many of the device parameters. The general form of this equation is as follows:

$$P = P_0 + \frac{\alpha_P}{L_{eff}} + \frac{\beta_P}{W_{eff}} \tag{3.20}$$

Where, P_0 is the value of the parameter for a long, wide transistor, and α_P and β_P are fitting factors. For example, the mobility can be calculated by:

$$\mu = \mu_0 + \frac{\alpha_\mu}{L_{eff}} + \frac{\beta_\mu}{W_{eff}} \tag{3.21}$$

BSIM model uses approximately 50 parameters. Its accuracy for narrow, short transistors is somewhat poor. Also, BSIM model predicts a negative output resistance for saturated MOSFETs at large drain-source voltages. Furthermore, in deep triode region, BSIM still exhibits slight discontinuities in the drain current [34].

The next model in BSIM series is BSIM2. It includes approximately 70 parameters. This version introduces new expressions for mobility, drain current, and sub-threshold conduction. It also presents the output impedance more accurately than BSIM model. Measured data shows that the overall accuracy of the model is only marginally higher than that of BSIM. For short, narrow transistors, BSIM2 suffers from large errors in the triode region and even substantial kinks in the saturation region [34].

Modeling the device behavior by means of empirical equations without enough attention to the physical phenomena, eventually, created difficulties in modeling of short-channel devices. For this reason, the next generation in BSIM series, BSIM3 has returned to the physical principals of device operation while maintaining many of the useful features of BSIM and BSIM2. BSIM3 has several versions and approximately includes 180 parameters in the third one. For channel lengths as low as 0.25 μm, BSIM3 is accurate for sub-threshold and strong inversion operation while still suffering from large errors in predicting the output impedance.

In 1995, the Compact Model Council (CMC) selected BSIM3 as a standard MOS model and BSIM3 and the next generation of it, BSIM4, have served many circuit designers. In recent years, the state of the art of compact modeling has advanced and the use of CMOS technology for analog and RF applications has increased. The CMC is going forward to introduce a new standard MOSFET model, one that is inherently symmetric, continuous and physically correct for charges and currents in the near future (2005).

The coverage of all MOS models is beyond the scope of this thesis. The reader is referred to [34] for more studies.

3.8 HSPICE – analog simulation tool

HSPICE is the SPICE based simulation software [35]. The acronym SPICE stands for Simulation Program with Integrated Circuit Emphasis. If the circuit of interest is described in a special syntax (HSPICE net-list syntax) with the excitation and supply sources properly defined, then the numerical routine can perform the required analysis and produce numerical and/or graphical results as requested by user. A HSPICE simulation has three steps:

1) Generating the circuit net-list (input) file
2) Running the simulation
3) Displaying, printing, and analyzing the simulation results

The net-list file can be generated by any text editor software with a special extension (i.e. filename.sp) and is made up of statements including circuit components, circuit interconnections, device models, input signals, voltage and current sources, type

of required analysis (i.e. DC, AC, Transient) and output data format. In general, the net-list file structure is as follows:

Title (required)

Circuit description

Sub-circuit description (if any)

Device models

Analysis

Output Format

.end (required)

The HSPICE net-list file which is used in our research to obtain the drain current (I_D) for various values of design parameters including W, L, V_{DS}, V_{GS} and V_{BS} is presented below.

```
HSPICE net-list for DC analysis of a single NMOS transistor
.lib '/CMC/kits/cmosp18/models/hspice/mm018.l' TT
vds 1 0 dc
vgs 2 0 dc vgs
vbs 3 0 dc vbs
m1 1 2 0 3  nch   l=l w=w
.options   nomod  post
.dc  vds 0.60  0.60  0.10  sweep  data=devsize
.print     dc   i(vds)
.param     w= 0.6u  l= 0.8u  vgs= 0.7v  vbs=0.5v
.data      devsize   w      l      vgs    vbs
                     1.00u  1.00u  0.60v  0.00v
                     2.00u  1.00u  0.60v  0.00v
                     1.00u  2.00u  0.60v  0.00v
                     2.00u  2.00u  0.60v  0.00v
                     1.00u  1.00u  0.60v  -1.00v
                     2.00u  1.00u  0.60v  -1.00v
                     1.00u  2.00u  0.60v  -1.00v
                     2.00u  2.00u  0.60v  -1.00v
.enddata
.end
```

The circuit schematic of a DC-biased NMOS transistor that is described in the above HSPICE net-list file is shown in Fig. 3.8.

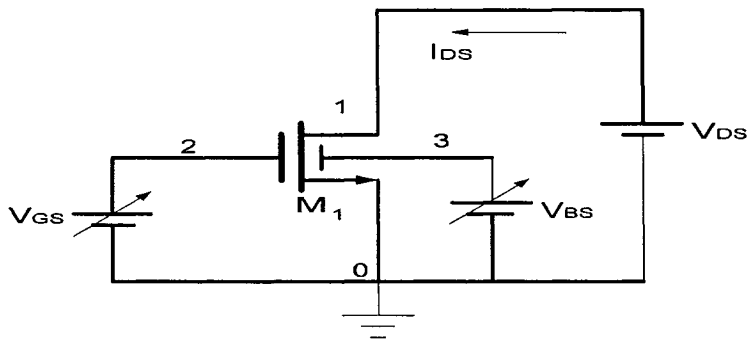


Figure 3.8: The DC-biased NMOS transistor

Some of the HSPICE commands used in the above net-list file are explained in the following:

(1) .options: This statement is used to store the simulation results and is needed to view graphical results using Awaves.

(2) .data : This statement allows the user to modify any number of parameter values during each simulation, in our case W, L, V_{GS} and V_{BS}.

(3) .param : This statement lets user create a parameter and assign algebraic mathematical expression to it.

(4) .lib : This statement introduces the component model and its parameters. In this case, a mixed-mode 0.18 μm CMOS process technology model is introduced.

(5) .print : This statement records the result of desired analysis (i.e. DC analysis) as output data in a file with "lis" extension.

(6) .dc : This statement specifies the type of required analysis (i.e. DC analysis).

The HSPICE output (.lis) file resulted from the above HSPICE input file is as follows:

```
******  Star-HSPICE  --    98.4   (990128)   12:13:46   05/11/2004
solaris
```

```
 Copyright (C) 1985-1997 by Avant! Corporation.
 Unpublished-rights reserved under US copyright laws.
 This program is protected by law and is subject to the
 terms and conditions of the license agreement found in:
 /CMC/tools/hspice1998.4/98.4/license.txt
 Use of this program is your acceptance to be bound by this
 license agreement. Star-HSPICE is the trademark of
 Avant! Corporation.
 Input File: ntest.sp
lic:
lic: FLEXlm:v5.12  USER:sa_sajja  HOSTNAME:townshend  HOSTID:830ff55a
PID:8682
lic: Using FLEXlm license file:
lic: /usr/local/etc/license/hspice/license.dat
lic: Checkout hspice;  Encryption code: 6CAEDA5D22AA0968537F
lic: License/Maintenance for hspice will expire on 12-jun-2004/2003.03
lic: 2(in_use)/50 FLOATING license(s) on SERVER mountain
lic:
Init: read install configuration file:
/CMC/tools/hspice1998.4/98.4/meta.cfg
Init: hspice initialization file:
/CMC/tools/hspice1998.4/98.4/hspice.ini
******
hspice net-list for dc analysis of a single nmos transistor

 Opening plot unit= 15
 file=ntest.sw0
******
hspice net-list for dc analysis of a single nmos transistor
 ******  dc transfer curves               tnom=  25.000 temp=  25.000
******
  *** d)ata name = devsize          , index =     1 ***
x
       volt     current
                    vds
 600.00000m      -3.8506u
y
******
hspice net-list for dc analysis of a single nmos transistor
 ******  dc transfer curves               tnom=  25.000 temp=  25.000
******
  *** d)ata name = devsize          , index =     2 ***
x
       volt     current
                   vds
 600.00000m      -7.8906u
y
******
hspice net-list for dc analysis of a single nmos transistor
 ******  dc transfer curves               tnom=  25.000 temp=  25.000
******
  *** d)ata name = devsize          , index =     3 ***
x
       volt     current
                    vds
 600.00000m      -2.2783u
y
```

```
******
hspice net-list for dc analysis of a single nmos transistor
 ******  dc transfer curves                tnom=  25.000 temp=  25.000
******
  *** d)ata name = devsize          , index =     4 ***
x
       volt    current
                   vds
 600.00000m     -4.7297u
y
******
hspice net-list for dc analysis of a single nmos transistor
 ******  dc transfer curves                tnom=  25.000 temp=  25.000
******
  *** d)ata name = devsize          , index =     5 ***
x
       volt    current
                   vds
 600.00000m    -53.6964n
y
******
hspice net-list for dc analysis of a single nmos transistor
 ******  dc transfer curves                tnom=  25.000 temp=  25.000
******
  *** d)ata name = devsize          , index =     6 ***
x
       volt    current
                   vds
 600.00000m   -106.3317n
y
******
hspice net-list for dc analysis of a single nmos transistor
 ******  dc transfer curves                tnom=  25.000 temp=  25.000
******
  *** d)ata name = devsize          , index =     7 ***
x
       volt    current
                   vds
 600.00000m    -37.0547n
y
******
hspice net-list for dc analysis of a single nmos transistor
 ******  dc transfer curves                tnom=  25.000 temp=  25.000
******
  *** d)ata name = devsize          , index =     8 ***
x
       volt    current
                   vds
 600.00000m    -75.2045n
y
         ***** job concluded
 ******  Star-HSPICE  --    98.4  (990128)  12:13:46  05/11/2004
solaris
******
hspice net-list for dc analysis of a single nmos transistor
 ******  job statistics summary            tnom=  25.000 temp=  25.000
******
```

```
          total memory used        458 kbytes

 # nodes =      6 # elements=      4
 # diodes=      0 # bjts    =      0 # jfets   =      0 # mosfets =      1
    analysis      time      # points  tot. iter  conv.iter
    op point          0.00          1          0
    dc sweep          0.00          8         25
    readin            0.40
    errchk            0.02
    setup             0.00
    output            0.00
          total cpu time          0.44 seconds
              job started at  12:13:46  05/11/2004
              job ended   at  12:13:47  05/11/2004
lic: Release hspice token(s)
```

In HSPICE output file, the results of simulation are presented between x and y letters. The output data, drain-source currents corresponding to certain values of W, L, V_{GS}, V_{DS}, and V_{BS}, can be used to calculate g_m, g_{ds}, and g_{mb} using equations (3.7), (3.14), and (3.15) respectively.

3.9 Summary

In this chapter, the physical structure, current-voltage characteristics, small-signal and SPICE models of MOS transistor were presented. The study of MOS transistor and its models reveal that as the device dimensions shrinks, the models for representing the device behavior become more complicated. Complex models of MOS transistors with lots of device parameters can not be used in hand analysis. On the other hand, a simple model such as level 1 is not accurate enough to predict the behavior of short-channel devices in today's short-channel analog IC design. Therefore, employing a new computer-based method for using advanced models that guarantees design accuracy and simplifies the design process will be quite helpful. This methodology is described in the next chapter.

Chapter 4

The Design Methodology

In this chapter, a new computer based design methodology for the design of basic building blocks of the analog integrated circuit (IC) is described. This method is intended to improve the design process of an analog IC by reducing design cycle time of basic building blocks.

4.1 Introduction

In digital IC design, the design procedure is almost fully automated. It consists of circuit specification, behavioral modeling, circuit synthesis, layout generation, layout extraction and post layout simulation. The digital design tools are able to automate digital functionalities from behavioral modeling to physical implementation without any transistor-level manipulation, facilitating cells retargeting and technology migration.

In analog IC design, the design procedure depends on the type of circuit that has to be designed and needs a lot of knowledge at transistor-level. CAD tools for analog IC design cannot provide circuit synthesis from the behavioral description. The problem of topology selection and circuit synthesis leads to transistor-level sizing. This makes the analog design procedure very complicated and automation without user optimization can be very restrictive. Therefore, retargeting and reuse of circuit topologies is not always possible and needs designer's intervention [32].

The design of an analog integrated circuit is a trade-off between performance parameters such as noise, gain, power dissipation, speed, supply voltage, voltage swings, input/output impedances and linearity that makes the design a multi-dimensional and difficult task. The design approach depends on the designer, type and complexity of the circuit. A typical design approach for designing an analog building block starts with choosing a suitable circuit topology that fulfills a given design specifications such as supply voltage, power dissipation, voltage/current gain, and output swing. Designer needs to determine dimensions, DC bias voltages and currents of all transistors in the circuit for correct operation. The drain-source voltage of each transistor is initially estimated by dividing rail-to-rail voltage (V_{DD}-V_{SS}) between NMOS and PMOS transistors in each branch. Note that PMOS transistors need more drain-source voltage, since they suffer from low mobility. The DC bias current of each branch can also be calculated using quiescent power equation as given below.

$$P_{diss} = (V_{DD} - V_{SS}) \times I_{branch} \tag{4.1}$$

By knowing branch current and drain-source voltage of each transistor in a branch, the next step is to determine the rest of design parameters including transistor dimensions (channel width and length), and gate-source voltage of all transistors so that performance specifications of the circuit are met. The small-signal analysis is carried out for formulating small-signal performance specifications, such as the gain of an amplifier. Then, the design parameters of the circuit are determined through small-signal model equations so that the design specifications are satisfied.

In this research, our goal is to develop a new design methodology and a computer-aided design tool based on above-mentioned approach. This new method takes

advantage of the transistor model equations (current-voltage model equations) that exist in the library of SPICE-based simulation tools and allows us to design a circuit without dealing with complicated model equations of a submicron process technology.

4.2 Description of the new design methodology

In this section, we shall describe the new design methodology for realization of basic building blocks realization of analog integrated circuits. Fig. 4.1 shows the design flow diagram of this method. The first step starts with selecting a popular building block topology according to design specifications. The second step is to perform small-signal analysis and find the small-signal model equations (design equations) in terms of the transconductances g_m, g_{ds} and g_{mb}. Next, a computer program (subroutine) is developed to automate the design of the selected circuit based on its topology and design equations. These steps have to be done once for every building block. The set of all programs developed for different basic building blocks forms an interactive, knowledge-based computer-aided design tool that can be used to design those analog building blocks automatically. Also, the designer has to create the HSPICE Net-list files and run HSPICE DC simulation of single transistors (NMOS and PMOS) for a specified MOS model (process technology) that provides the values of DC bias currents (I_{DS}) as output data for a certain range of DC bias voltages (V_{DS}, V_{GS}, V_{BS}) and device dimensions (W, L) as input variables. In other words, the output current-voltage characteristics of NMOS and PMOS transistors are obtained in this phase for a certain range of W, L and bias voltages (V_{DS}, V_{GS}, V_{BS}). This part has been automated by embedding a small setup program inside the CAD tool.

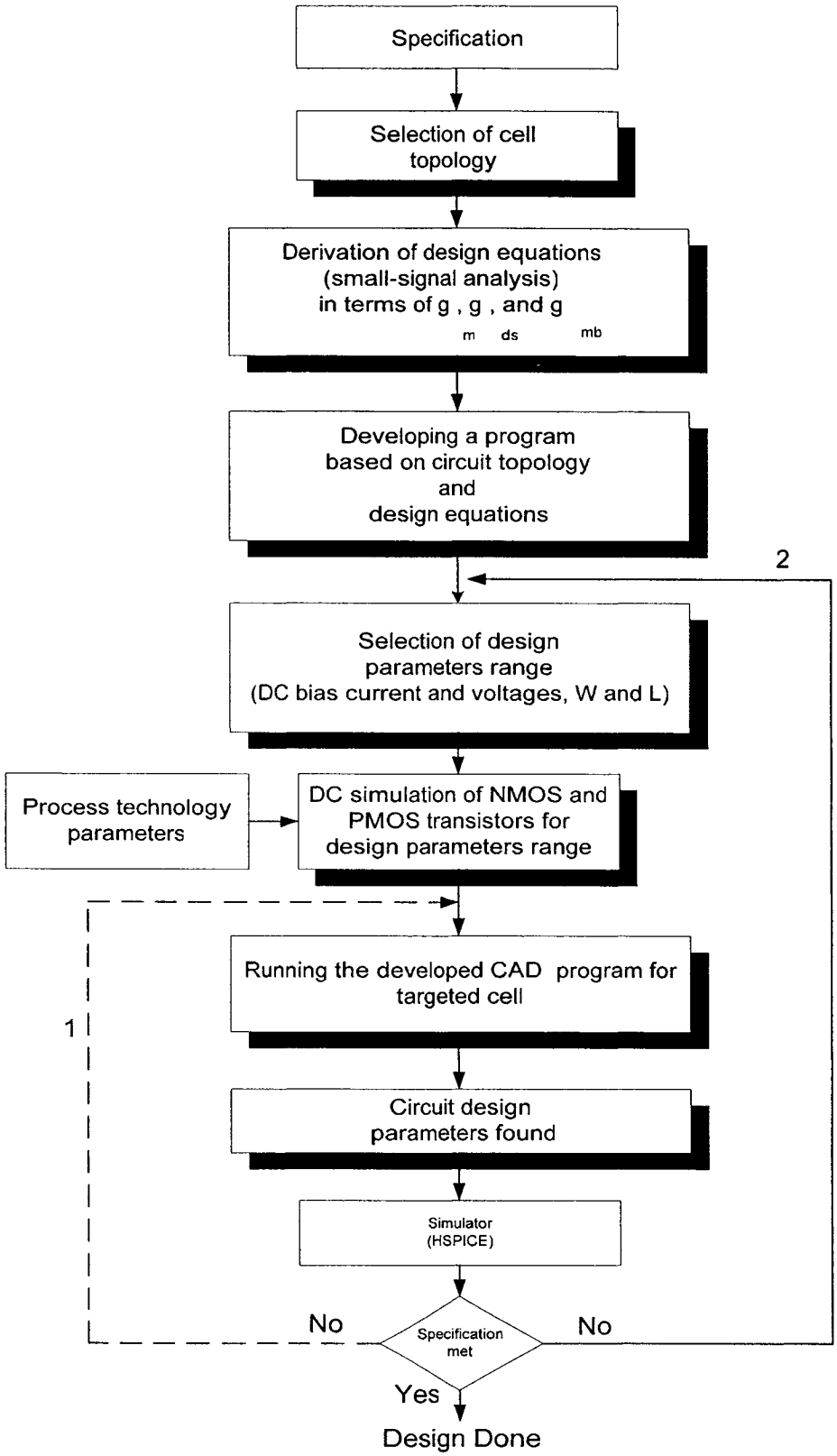


Figure 4.1: Flow diagram of new design methodology

All of the design variables including transistor dimensions, the values of bias currents and bias voltages are recorded in a file on computer hard disk. The CAD tool uses these data to calculate the values of g_m, g_{ds} and g_{mb} which are needed to estimate design specifications through its small-signal design equations for each building block. In order to design a building block, the designer has to run the subprogram related to that building block and enters the DC bias conditions. The values of g_m, g_{ds}, and g_{mb} for all transistors that exist in the building block of interest are calculated through the design equations for a number of points on the output current-voltage characteristics of the NMOS and the PMOS transistors based-on design variables range. If the design specifications are met, which is tested by HSPICE simulation, all corresponding design parameters are saved in a file to be displayed or printed by the designer. But, if the design specifications are not met then the designer has to rerun the subprogram for new bias conditions (path 1 on Fig. 4.1) or start from running HSPICE DC simulation for a new range of design variables (path 2 on Fig. 4.1). It is important to note that there could be one, more than one or no solution for a design problem.

4.3 Calculation of g_m, g_{ds} and g_{mb}

The calculation of g_m, g_{ds} and g_{mb} contributes significantly to the accuracy of the proposed design methodology. Therefore, the values of these parameters must be calculated accurately. To illustrate how the values of g_m, g_{ds} and g_{mb} are calculated by the CAD tool and how accurate these values are, several examples are presented by using output current-voltage characteristics of an NMOS transistor. All calculations and concepts are applicable to PMOS transistor as well. We consider two process

technologies, 0.5 μm (BSIM1) as an educational, and 0.18 μm (BSIM3) as a professional design process technology to illustrate the g_m, g_{ds} and g_{mb} calculations.

4.3.1 Calculation of g_m

To calculate the value of g_m, we shall consider the output current-voltage characteristics of the NMOS transistor for the following range of design parameters. Two process technologies, i.e. 0.5 μm and 0.18 μm, and two different resolutions (i.e. step values) of V_{GS} (1 Volt and 0.25 Volt), as shown in Figs. 4.2 to 4.5 are considered. Note that the body-source voltage (V_{BS}) and the dimensions of the NMOS transistor are fixed.

V_{DS}: 0.01 to 5 V with step value of 0.1 V

V_{GS}: (a) 2 to 4 V with step value of 1 V, and (b) 2.75 to 3.25 V with step value of 0.25 V

V_{BS} = 0 V and W = L = 2 μm

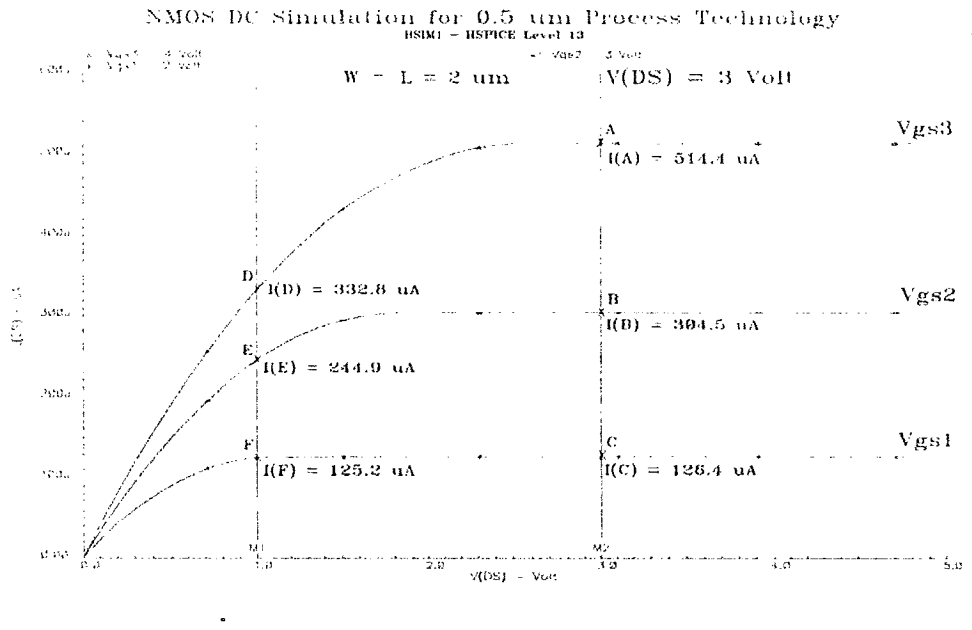


Figure 4.2: I / V characteristics of NMOS with V_{GS} step value of 1 Volt

We choose two points B and E on each I / V characteristics, one in saturation region (V_{DS} = 3 Volt and V_{GS} = 3 Volt) and another one in the triode region (V_{DS} = 1 Volt and V_{GS} = 3 Volt), because we usually bias NMOS and PMOS transistors in these two regions. In Fig. 4.2, the values of g_m at points B and E can be determined using equation 3.7 as follows:

$$g_m(B) = \left(\frac{I(A)-I(B)}{V_{GS_3}-V_{GS_2}} + \frac{I(B)-I(C)}{V_{GS_2}-V_{GS_1}}\right) / 2 \tag{4.2}$$

$$= \left(\frac{514.4-304.5}{3-2} + \frac{304.5-126.4}{2-1}\right) / 2$$

$$= 194 \ \frac{\mu A}{V}$$

$$g_m(E) = \left(\frac{I(D)-I(E)}{V_{GS_3}-V_{GS_2}} + \frac{I(E)-I(F)}{V_{GS_2}-V_{GS_1}}\right) / 2 \tag{4.3}$$

$$= \left(\frac{332.8-244.9}{1} + \frac{244.9-125.2}{1}\right) / 2$$

$$= 103.8 \ \frac{\mu A}{V}$$

Where I(A), I(B), I(C), I(D), I(E), and I(F) are measured currents by Cadence DC simulation at point A, B, C, D, E and F respectively. Also, $g_m(B)$ and $g_m(E)$ are values of transconductances at points B, and E.

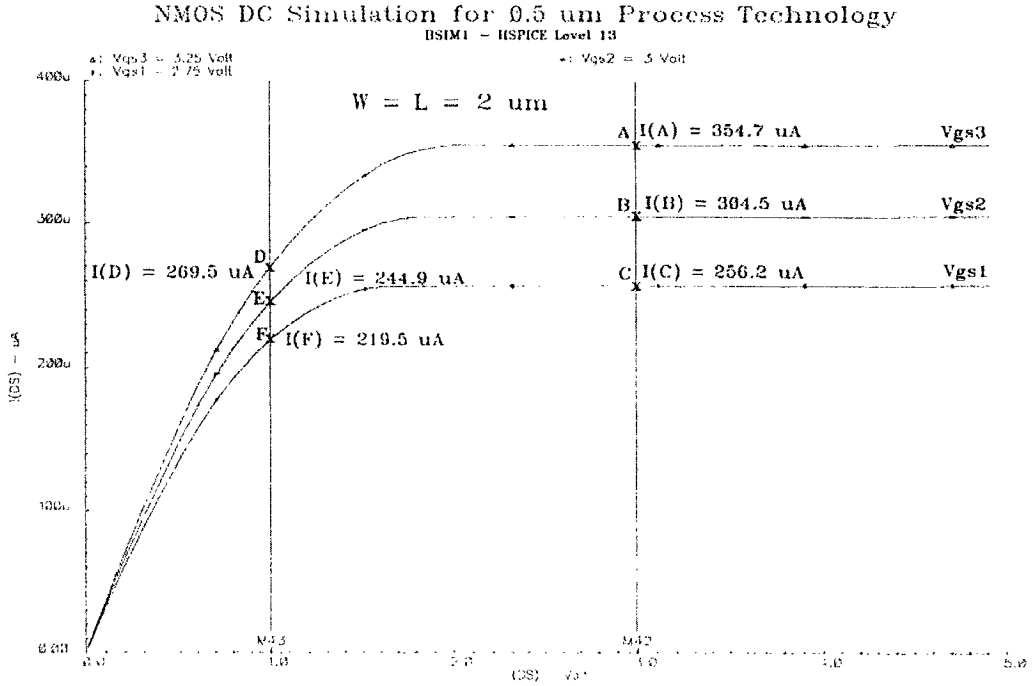


Figure 4.3: I / V characteristics of NMOS with V_{GS} step value of 0.25 V

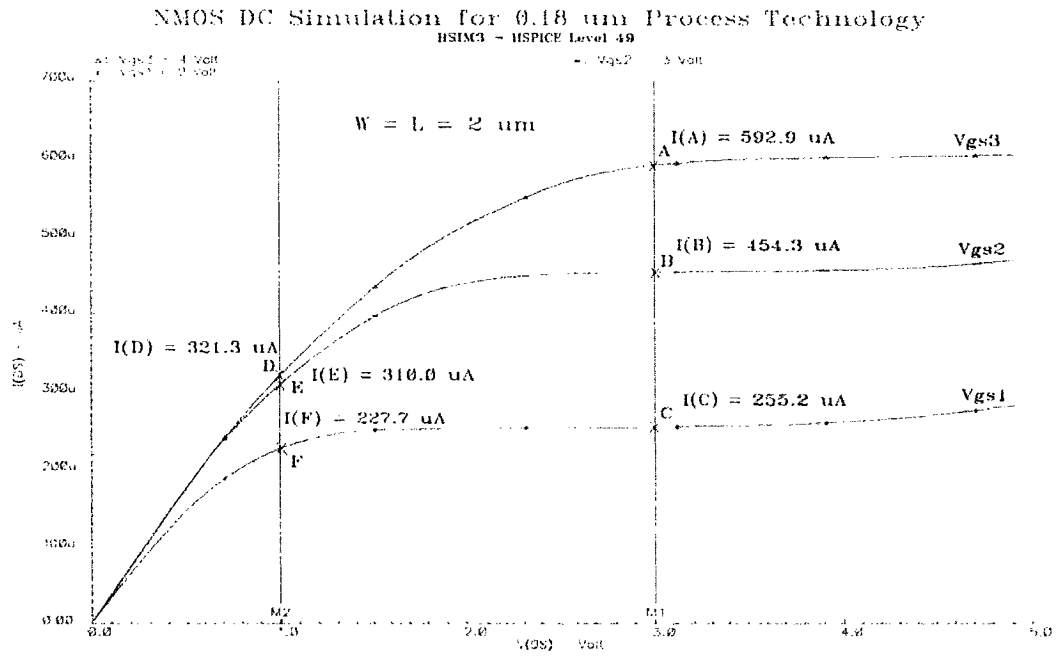


Figure 4.4: I / V characteristics of NMOS with V_{GS} step value of 1 Volt

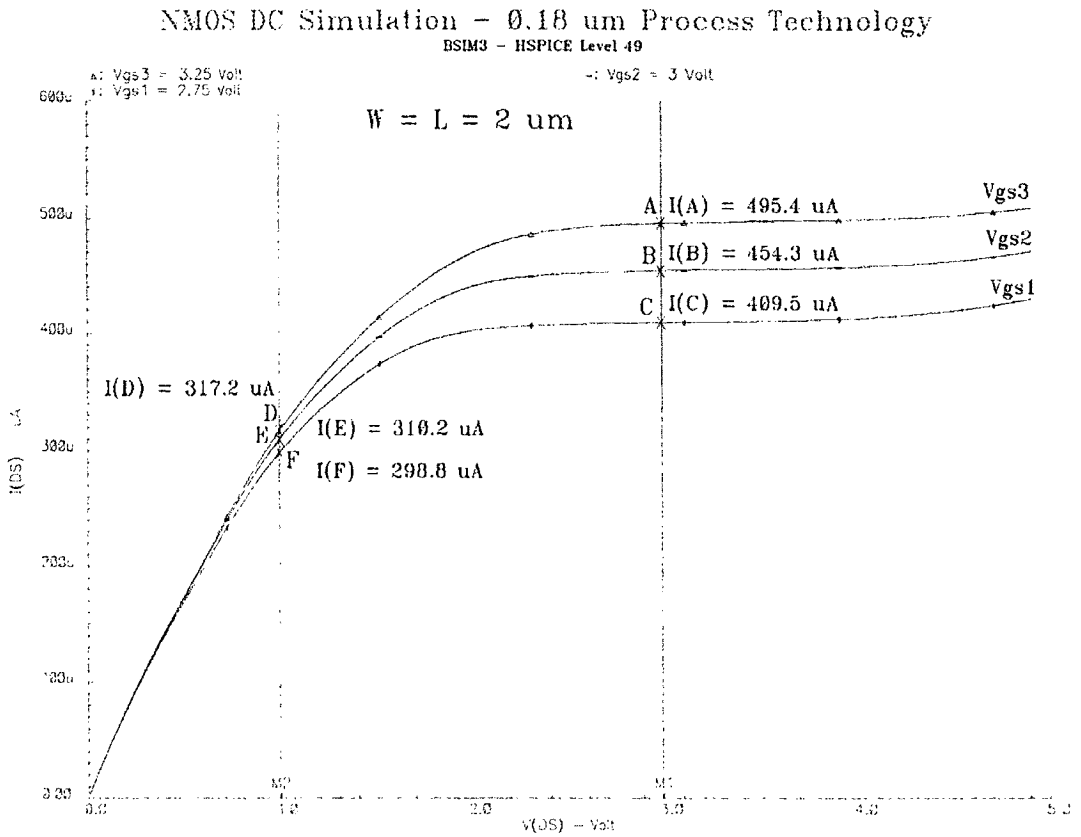


Figure 4.5: I/V characteristics of NMOS with V_{GS} step value of 0.25 V

For all other cases, shown in Fig. 4.3 to 4.5, the values of g_m at points B and E can be calculated in the same way as above. Table 4.1 lists the g_m values that are calculated through plots and obtained from HSPICE DC simulations.

Table 4.1: g_m values obtained from calculation and HSPICE simulation

Process Technology	V_{GS} Step Value (Volt)	Point	g_m Calculated from Plots $(\frac{\mu A}{V})$	g_m Obtained from Simulation $(\frac{\mu A}{V})$	Difference of g_m Values (%)
0.5 μm	1	B	194	198	2
		E	103.8	101.9	1.8
	0.25	B	197	198	0.5
		E	100	101.9	1.9
0.18 μm	1	B	168.85	171.9	1.8
		E	46.8	36.57	22
	0.25	B	171.8	171.9	0.06
		E	36.8	36.57	0.625

From table 4.1, we conclude that the error of g_m calculation can be kept in a reasonable range if designer choose the step value of bias voltage V_{GS} precisely.

4.3.2 Calculation of g_{ds}

In order to calculate the value of g_{ds}, we consider two points A and D on the output current-voltage characteristics of the NMOS transistor as shown in Fig. 4.6 to 4.10. We calculate the g_{ds} values for two different step values of V_{DS} and two types of process technologies (0.5 μm and 0.18 μm), and compare those values with HSPICE DC simulation results. The design parameters for obtaining the output I/V characteristics, as shown in Fig. 4.6 to 4.10, are listed below.

V_{DS}: 0.01 to 5 V with step value of 0.1 V

V_{GS}: (a) 2 to 4 V with step value of 1 V, and (b) 2.75 to 3.25 V with step value of 0.25 V

V_{BS} = 0 V and W = L = 2 μm

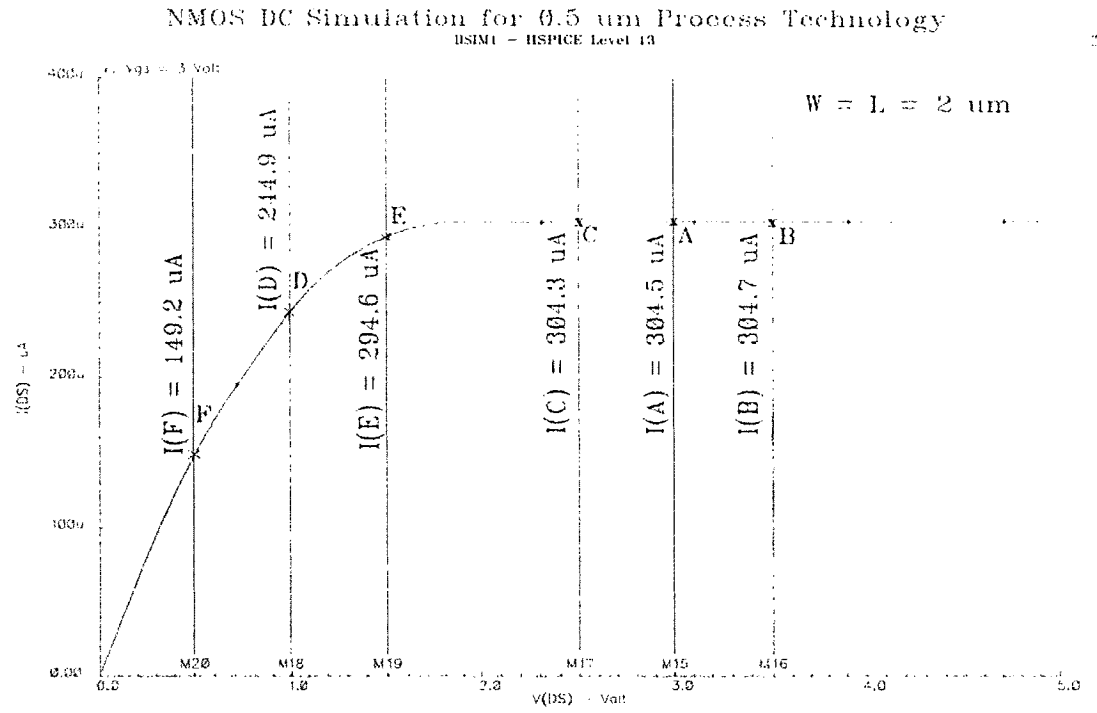


Figure 4.6: I/V characteristics of NMOS with V_{DS} step value of 0.5 V

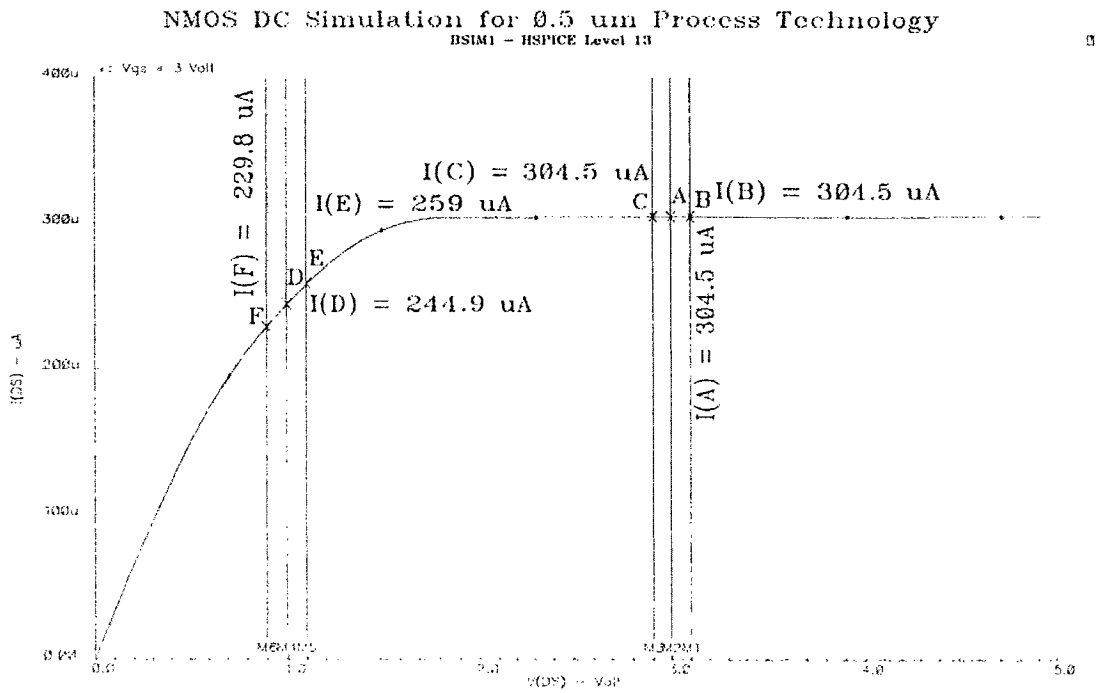


Figure 4.7: I/V characteristics of NMOS with V_{DS} step value of 0.1 V

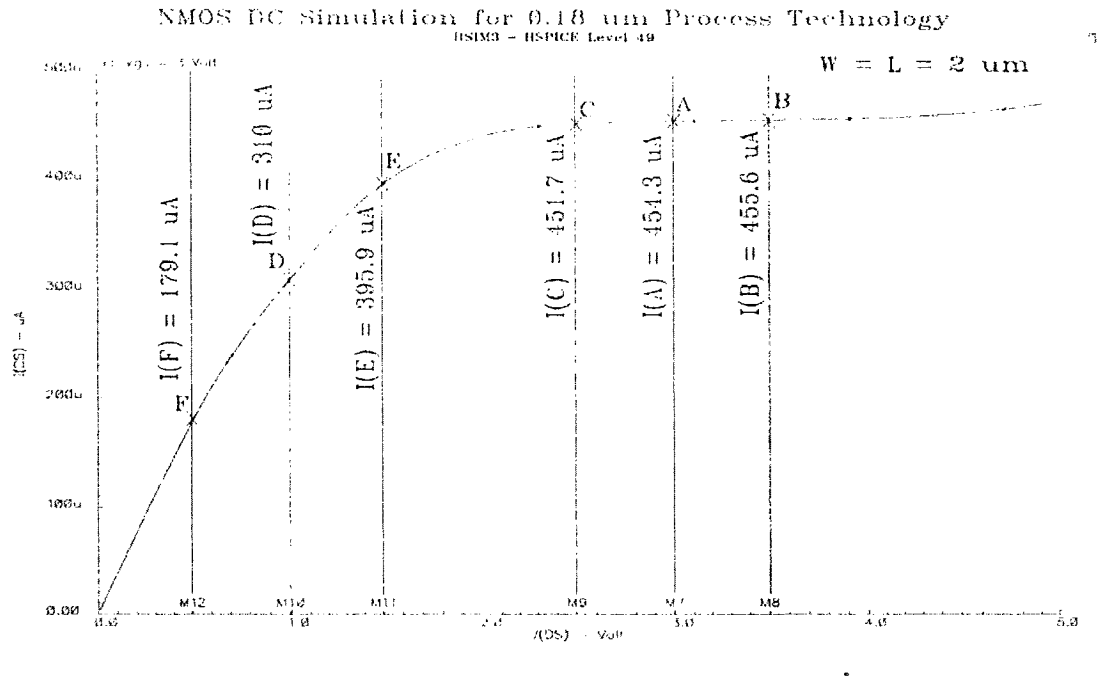


Figure 4.8: I/V characteristics of NMOS with V_{DS} step value of 0.5 V

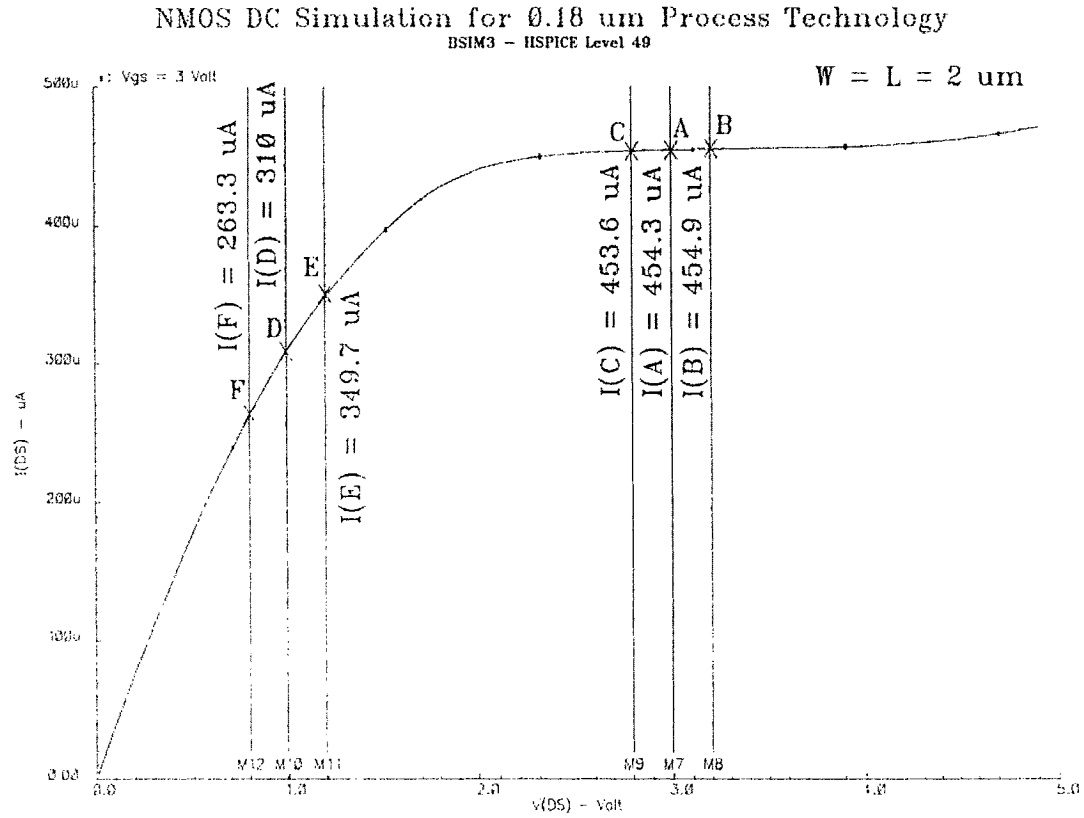


Figure 4.9: I/V characteristics of NMOS with V_{DS} step value of 0.2 V

As an example, we calculate the g_{ds} values at points A and D, using equation 3.14 derived in previous chapter as follows:

$$g_{ds}(A) = \left(\frac{I(C)-I(A)}{V(C)-V(A)} + \frac{I(A)-I(B)}{V(A)-V(B)}\right) / 2 \qquad (4.4)$$

$$= \left(\frac{454.9-454.3}{0.2} + \frac{454.3-453.6}{0.2}\right) / 2$$

$$= 3.25 \ \frac{\mu A}{V}$$

$$g_{ds}(D) = \left(\frac{I(E)-I(D)}{V(E)-V(D)} + \frac{I(D)-I(F)}{V(D)-V(F)}\right) / 2 \qquad (4.5)$$

$$= \left(\frac{349.7-310}{0.2} + \frac{310-263.3}{0.2}\right) / 2$$

$$= 216 \ \frac{\mu A}{V}$$

Where I(A), V(A), I(B), V(B), I(C), V(C), I(D), V(D), I(E), V(E), I(F), and V(F) are measured currents and voltages by Cadence DC simulation at points A, B, C, D, E and F respectively. Also, $g_{ds}(A)$ and $g_{ds}(D)$ are values of output conductance at points A and D.

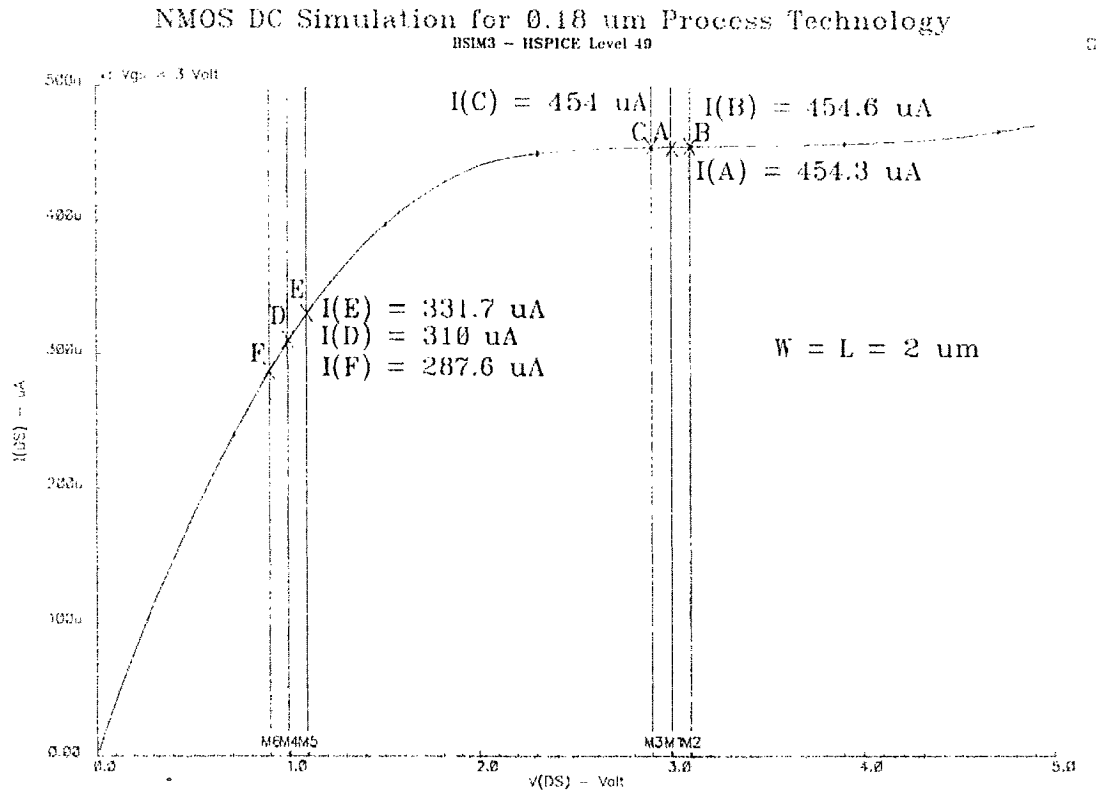


Figure 4.10: I/V characteristics of NMOS with V_{DS} step value of 0.1 V

All the g_{ds} values obtained by calculation and HSPICE DC simulation for different I / V characteristics are listed in table 4.2.

Table 4.2: g_{ds} values obtained from calculation and HSPICE simulation

Process Technology	V_{DS} Step Value (Volt)	Point	g_{ds} Calculated from Plots $(\frac{\mu A}{V})$	g_{ds} Obtained from Simulation $(\frac{\mu A}{V})$	Difference of g_{ds} Values (%)
0.5 μm	0.5	A	0.4	0.3462	13.45
		D	145.4	145.1	0.2
	0.1	A	0	0.3462	100
		D	146	145.1	0.6
0.18 μm	0.5	A	3.9	3.227	17.25
		D	216.8	216	0.37
	0.2	A	3.25	3.227	0.7
		D	216	216	0
	0.1	A	3	3.227	7
		D	220.5	216	2

From the table 4.2, we conclude that the error of g_{ds} calculation can be kept in a reasonable range if designer choose the step value of bias voltage V_{DS} precisely.

4.3.3 Calculation of g_{mb}

For calculation of g_{mb}, we consider an NMOS transistor with fixed W, L, V_{DS}, and V_{GS}, and plot the drain-source current versus body-source voltage (V_{BS} or V_{SB}) variations as shown in Figs. 4.11 and 4.12 for both 0.5 μm and 0.18 μm process technologies respectively. The design parameters are as follows:

V_{SB} : 0.01 to 5 V with step value = 0.1 V

$V_{DS} = V_{GS} = 3$ V and W = L = 2 μm

For two different step value of V_{SB}, we calculate g_{mb} at point A, using equation 3.15 derived in the previous chapter as below.

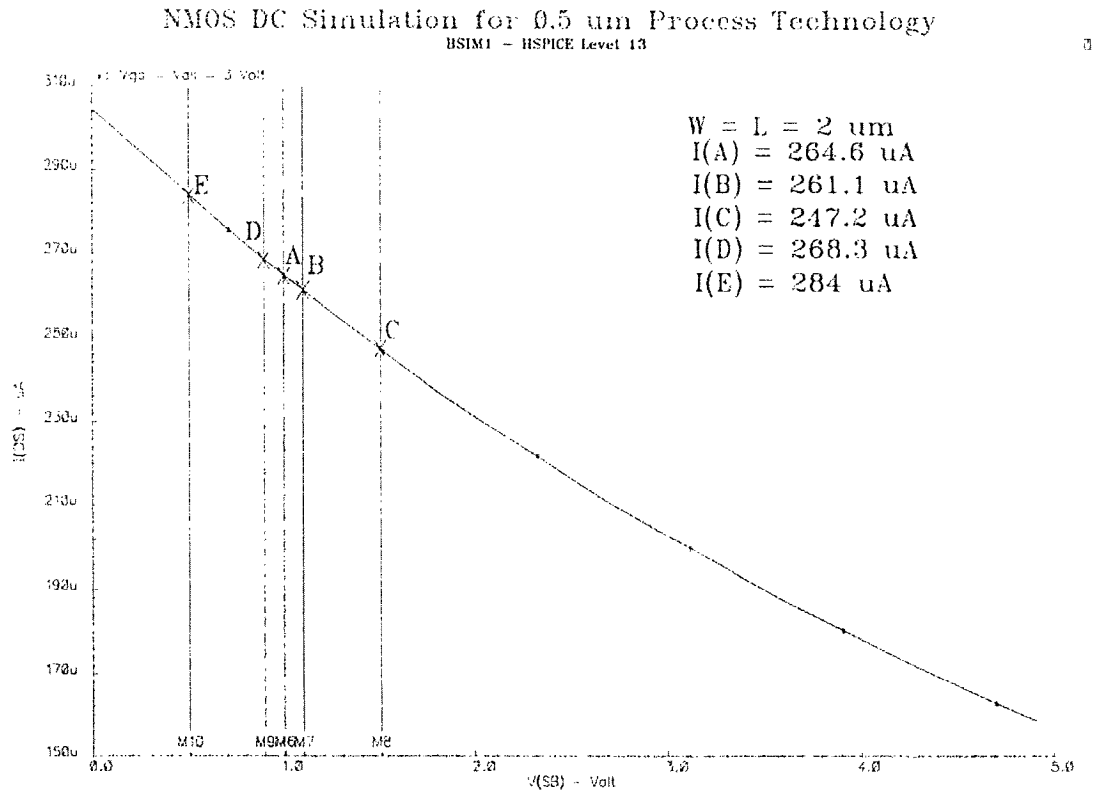


Figure 4.11: I/V characteristics of NMOS with V_{SB} step value of 0.1 V and 0.5 V

For a step value of 0.1 volt, at point A we have

$$g_{mb}(A) = \left(\frac{I(D)-I(A)}{V(D)-V(A)} + \frac{I(A)-I(B)}{V(A)-V(B)}\right) / 2 \tag{4.6}$$

$$= \left(\frac{380.1-373.9}{0.1} + \frac{373.9-367.6}{0.1}\right) / 2$$

$$= 62.5 \ \frac{\mu A}{V}$$

and for a step value of 0.5 Volt,

$$g_{mb}(A) = \left(\frac{I(E)-I(A)}{V(E)-V(A)} + \frac{I(A)-I(C)}{V(A)-V(C)}\right) / 2 \tag{4.7}$$

$$= \left(\frac{409.1-373.9}{0.5} + \frac{373.9-344.9}{0.5}\right) / 2$$

$$= 64.2 \ \frac{\mu A}{V}$$

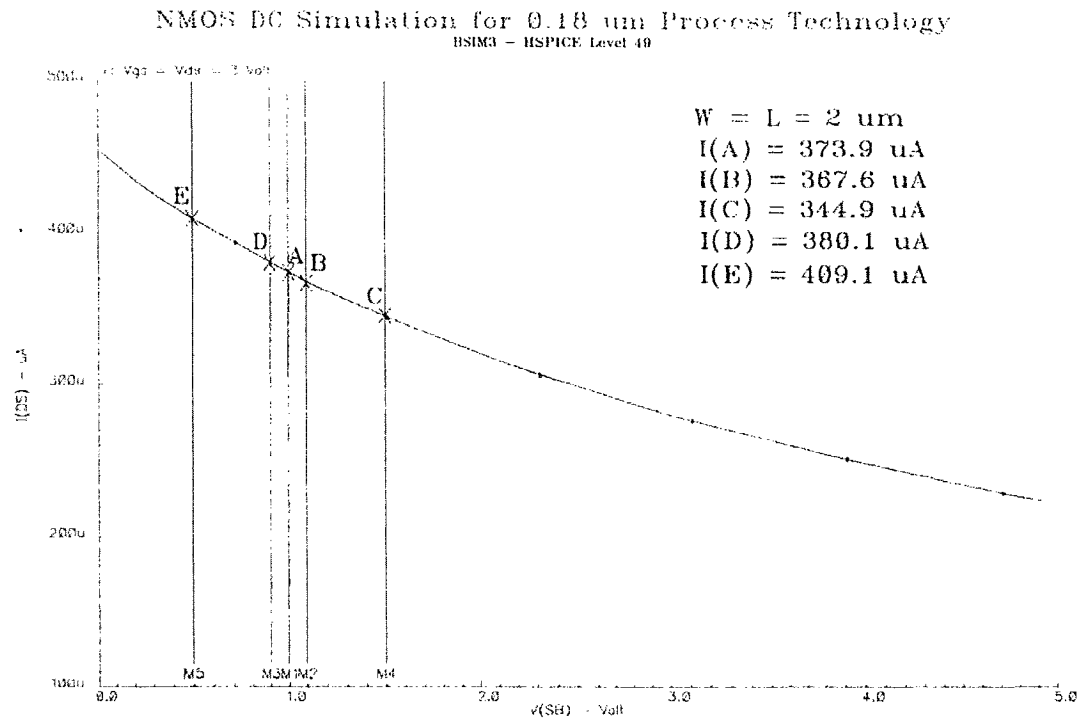


Figure 4.12: I/V characteristics of NMOS with V_{SB} step value of 0.1 V and 0.5 V

Where I(A), V(A), I(B), V(B), I(C), V(C), I(D), V(D), I(E) and V(E) are measured currents and voltages by Cadence DC simulation at points A, B, C, D and E respectively. $g_{mb}(A)$ is the value of transconductance at point A.

All the values of g_{mb} obtained by calculation and HSPICE DC simulation for different cases are listed in table 4.3.

Table 4.3: g_{mb} values obtained from calculation and HSPICE simulation

Process Technology	V_{SB} Step Value (Volt)	Point	g_{mb} Calculated from Plots $(\frac{\mu A}{V})$	g_{mb} Obtained from Simulation $(\frac{\mu A}{V})$	Difference of g_{mb} Values (%)
0.5 μm	0.1	A	36	36.87	2.35
	0.5		36.8		0.19
0.18 μm	0.1		62.5	63.35	1.34
	0.5		64.2		1.32

It is instructive to note that as the source-body potential difference increases, the drain-source current decreases.

Finally, the comparison of calculated values from the following plots with the values obtained from HSPICE DC simulations reveal that calculation accuracy of these parameters depend on the following factors.

1- The resolution (step value) of input bias voltages range i.e. V_{GS}, V_{DS}, and V_{BS}

2- The location of a point on current-voltage characteristics

4.4 The structure and menu options of developed CAD tool

Fig. 4.13 shows the structure of the software that implements the new design methodology described in section 4.2. It has three parts including setup module, library of basic building blocks modules, and HSPICE simulation tool.

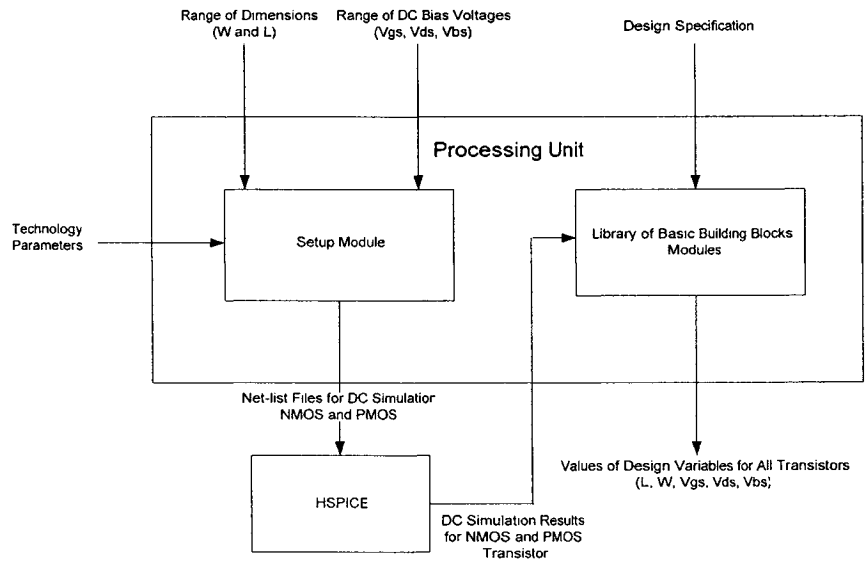


Figure 4.13: The CAD tool structure

The first version of the software was developed in MATLAB using MATLAB programming language. The menus of this version are presented in appendix A. The second version of the software was developed in C because a fast program is needed, and MATLAB-based programs are not good enough in terms of speed. The second version of software is called CADT (Concordia Analog Design Tool). It has a multiple choice menu and is modular so that can be modified or expanded easily. A sample source code of this program is presented in appendix B. The tool calls the HSPICE to perform DC simulation and provide required data for further processing. The processing unit consists of two distinguished parts i.e. the setup module and the library of modules. The setup module

accepts five major parameters (transistor dimensions and DC bias voltages) in a specific range and generates Net-list files to be simulated by HSPICE. The output of HSPICE DC analysis is imported back into the CAD tool for creating a database of design variables. This database is processed by library modules. The library modules are a set of modules that describe the behavior of a basic building block in terms of design equations. Each module in the library is used to design one basic building block. The final results, output data of each module, are the values of design variables such as V_{GS}, V_{DS}, V_{BS}, L, and W for all transistors in the circuit that satisfy the design specifications. Since, the final results depend on initial design variables range, it is possible that the tool will fail to find a feasible design for certain set of input specifications.

The main menu of the CAD tool, shown in Fig. 4.14, consists of five different options of analog circuit building blocks, including current source or mirror, single stage amplifiers, differential amplifiers, voltage dividers, and other building blocks (OPAMP, OTA, etc.).

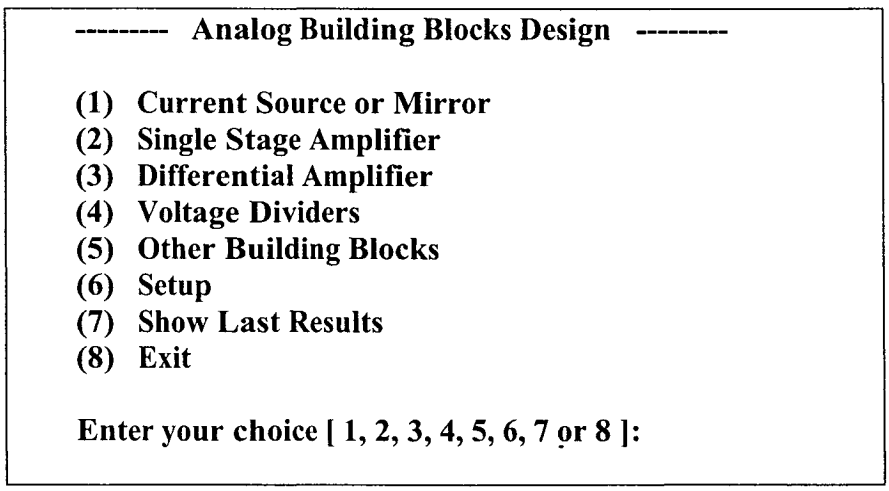


Figure 4.14: Main menu of the CAD tool

Each main menu option has several sub-menus that allow the design of different versions of analog cells in each category. The designer needs to set up the tool for both a specific process technology and a suitable range of design parameters before using it. This has to be done through the setup menu option (choice no. 6) and by modifying the HSPICE Net-list file named "source.sp". The setup option allows the user to enter a desirable range and step value for design parameters including channel width (W), channel length (L), drain-source voltage (V_{DS}), gate-source voltage (V_{GS}), and body-source voltage (V_{BS}), a sample case of entered data is shown in Fig. 4.15.

Setup Option Creates Database Files for Analog Building Blocks Realization
Press Ctrl+C to Return to the Main Menu

Enter Channel Length Range [Min. Max. Step] (u): 1 5 0.5
Enter Channel Width Range [Min. Max. Step] (u): 1 20 0.5
Enter Drain-Source Voltage Range [Min. Max. Step] (V): 0.1 3 0.1
Enter Gate-Source Voltage Range [Min. Max. Step] (V): 0.1 3 0.1

Notice: Enter 0 for Min. of V(BS) Voltage Always
Enter Body - Source Voltage Range [Min. Max. Step] (V): 0 3 0.1

Figure 4.15: A sample of entered data through the setup menu

To select the desirable process technology model parameters, the second line of "source.sp" file has to be modified so that the suitable MOS models for a specific process technology is called from the HSPICE library. The "source.sp" file is listed below.

```
* HSPICE DC ANALYSIS NET-LIST FOR NMOS/PMOS
.lib '/CMC/kits/cmosp18/models/hspice/mm018.1' TT   <--- second line
vds 1 0 dc
vgs 2 0 dc vgs
vbs 3 0 dc vbs
m1 1 2 0 3 nch pch  l=l  w=w
.options   nomod  post
```

```
.dc        vds  vds1  vds2  stepvds  sweep   data=devsize
.print     dc   i(vds)
.param     w= 0.6u  l= 0.8u  vgs= 0.7v  vbs=0.5v
.data      devsize   w  l  vgs  vbs
.enddata
.end
```

The "source.sp" file is used by the setup module to create two HSPICE Net-list files called "nmos.sp" and "pmos.sp" for running HSPICE DC simulation for both NMOS and PMOS transistors. The results of HSPICE DC simulation of NMOS and PMOS transistors are stored in "nmos.lis" and "pmos.lis" files respectively. These files contain the necessary data to be processed by the software modules.

For designing a specific building block, one shall select the suitable option of the main menu and then the submenu related to the design of that building block. Other menu options and a set of design examples that illustrate the applications and abilities of the tool in the design of analog integrated circuit basic building blocks are presented in the next chapters.

4.5 Comparison of the developed tool with already available tools

There are many CAD tools for the design automation of analog integrated circuits. Some of these tools have been discussed along with their advantages and disadvantages in the chapter 2. Briefly, some of them cannot provide optimal design solutions and the others can only provide an optimal design solution for a specific topology. Also, many of the analog IC automation methods are very slow. There is a commercial IC design tool called "Cadence". This tool is the most popular in the semiconductor industry and seems to be the best, but expensive. Therefore many companies are using alternative tools that are much cheaper. One of the Cadence tools (parametric analysis) provides the possibility of entering design goal(s) and a range of related design variables with a specific step value for designing a specific circuit. A

simulation based method is used by this tool i.e. running a number of simulations by changing the values of design parameters so that the design goals can be reached. As tested by the author, when the numbers of design variables are increased this tool becomes very slow in finding a design solution. In comparison with Cadence, the advantages of our new design methodology are speed and ease of usage. It is also less expensive and accurate enough to be developed by analog designers especially for basic building blocks, such as DC biasing and single stage amplifiers.

4.6 Summary

A new method of the design of analog integrated circuit is described in this chapter. It is illustrated how to calculate the accurate values of g_m, g_{ds} and g_{mb} parameters using current-voltage output characteristics of MOS transistors obtained from DC simulation of both NMOS and PMOS transistors. These parameters appear in the design equations derived from small-signal analysis and can be used to estimate the values of small-signal design specifications.

Analog designers need a PC or workstation, HSPICE simulator, and C/C++ programming language compiler to implement this new design methodology. The structure and main menu of the CAD tool which is developed based on our new design methodology are also presented in this chapter. Other menus and the way of using the tool to design a set of analog building blocks will be explained in the next chapters.

Chapter 5

Small Scale Building Blocks

Design Examples

This chapter presents the design of small sized analog integrated circuit building blocks using the new design methodology and the CAD tool which are described in the previous chapter.

5.1 Introduction

To study the design of analog integrated building blocks through our new design methodology, we divide the sample design examples into three groups including small, medium, and large scale building blocks. The schematic of small building blocks along with their design procedures and related submenus of the CAD tool are presented in this chapter.

Design parameters obtained from our CAD tool are tested by HSPICE DC and/or AC analysis to prove the accuracy and reliability of the developed CAD tool. Our CAD tool is also referred to as CADT (Concordia Analog Design Tool).

5.2 Current source and mirror design examples

Fig. 5.1 shows the submenu of current source/mirror design of the CADT. It provides the possibility of the design of several kinds of current source and mirrors.

current source and mirror have different topologies such as basic current source, basic current mirror, active current mirror and cascode current mirror that have wide applications in modern analog systems design. We discuss the design of these circuits using the CADT in the following sections.

```
--------- Current Source or Mirror Design ---------

          (1) NMOS Current Source/Mirror
          (2) PMOS Current Source/Mirror
          (3) Cascode Current Mirror
          (4) Return to Main Menu

Enter your choice [ 1, 2 or 3 ]:
```

Figure 5.1: Current source/mirror design submenu of the tool

5.2.1 Basic current source

Current sources have a wide usage in analog circuits such as single-stage and differential amplifiers. A single MOS transistor (NMOS or PMOS) operating in saturation can act as a current source. To design a stable current source, we usually apply a DC bias voltage to the gate of MOS transistor as shown in Fig. 5.2.

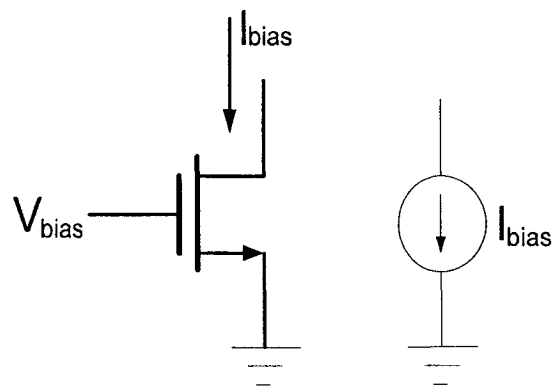


Figure 5.2: NMOS basic current source

In this case, the design problem is to find the dimensions of MOS transistor while the drain-source current and at least two of DC bias voltages (V_{GS} or V_{DS} and V_{BS}) are known.

The current source design module of the software receives the required bias voltages and drain-source current from the designer and finds the corresponding dimensions of the transistor. As a design example, suppose on being prompted by the CADT, the user provides the following data, V_{GS} = 1 V, V_{BS} = 0 V and I_{DS} = 10 μA for a 0.18 μm process technology as the input to the current source design subprogram of the CADT. Then, the software will process the database files and deliver its results for both NMOS and PMOS transistors as shown in table 5.1. There could be more than one solution when the exact drain-source current does not exist in the database files of the CADT and the designer has to choose the best (the nearest) values of design parameters.

Table 5.1: NMOS and PMOS current source design parameters

Type of Device	V_{GS} (Volt) Input	V_{DS} (Volt) Output	V_{BS} (Volt) Input	W (μA) Output	L (μA) Output	I_{DS} (μA) Received by Tool	I_{DS} (μA) Obtained from Simulation
NMOS	1	1	0	0.8	3.7	10.0449	10.0449
PMOS	1	1	0	3.5	3	10.0334	10.0334

The results show that PMOS transistor needs a larger size (area) than NMOS transistor in the same bias conditions. Also, HSPICE DC simulation results with above design parameters for both transistors lead to the same drain-source currents that show the accuracy and reliability of the CADT.

5.2.2 Basic current mirror

Current mirrors are important building blocks in the modern current mode IC design where signal is processed primarily as a current rather than a voltage. In a current mirror, the current produced by a reference transistor, also called the source, is copied to another device by employing the principle of equal gate to source control voltage (V_{GS}) and ratio of transistor dimensions. The basic MOS current mirror consisting of two MOS devices is shown below.

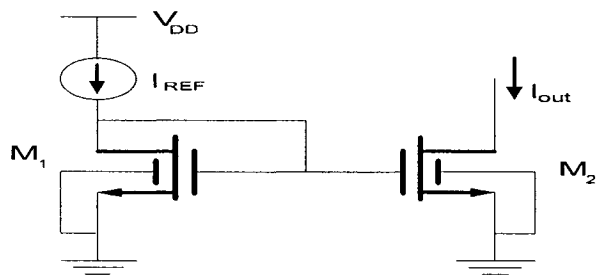


Figure 5.3: NMOS basic current mirror

Transistor M_1 is always in saturation because its gate and drain are connected together. Assuming the drain-source voltage of M_2 is large enough to keep M_2 in the saturation region, we can write

$$\frac{I_{out}}{I_{REF}} = \frac{I_{DS_2}}{I_{DS_1}} = \frac{K_{n_2} \times \frac{W_2}{L_2}(V_{GS_2} - V_{th_2})(1 + \lambda V_{DS2})}{K_{n_1} \times \frac{W_1}{L_1}(V_{GS_1} - V_{th_1})(1 + \lambda V_{DS1})} \tag{5.1}$$

Neglecting channel-length modulation for simplicity and assume that both transistors have the same type, then we have

$$I_{out} = \frac{(\frac{W_2}{L_2})}{(\frac{W_1}{L_1})} \times I_{REF} \tag{5.2}$$

The ratio of $\frac{(\frac{W_2}{L_2})}{(\frac{W_1}{L_1})}$ can be used to obtain relative amplification or attenuation.

Equation 5.2 reveals that this topology allows precise copying of the current with no dependence on process and temperature.

To design a current mirror stage, the current mirror design subprogram of the tool receives the bias current (I_{REF} or I_{out}) and at least two of the three DC bias voltages (V_{GS} or V_{DS} and V_{BS}) of each transistor and provides us with its dimensions.

As a design example, suppose on being prompted by the CADT, the user provides the following data, V_{GS}=1V, V_{BS}= 0V, $V_{DS_{M1}}$ =1V, $V_{DS_{M2}}$ =1 V, $I_{REF} = 20\,\mu A$ and $I_{out} = 40\,\mu A$ for a $0.18\,\mu m$ process technology. Then, the software will process the database files and deliver its results for both NMOS transistors as shown in table 5.2. There could be more than one solution and the designer has to choose the best values of design parameters. As well, when there are not exact values of design parameters in database files the software provides the nearest ones.

Table 5.2: Basic NMOS current mirror design parameters

Device	V_{GS} (Volt) Input	V_{DS} (Volt) Output	V_{BS} (Volt) Input	W (μA) Output	L (μA) Output	$I_{DS}(\mu A)$ Obtained by Tool	$I_{DS}(\mu A)$ Obtained from HSPICE
M_1	1	1	0	4	10	20.4419	20.4419
M_2	1	1	0	10	13	40.1614	40.1614

HSPICE DC simulation results with above design parameters of the current mirror lead to the same drain-source currents that show the accuracy and reliability of the CADT. Also, the results prove equation 5.2.

5.2.3 Cascode current mirror

In the discussion of basic current mirror, we have neglected channel length modulation factor. This effect results in significant errors in copying currents, especially if minimum length transistors are used to minimize the width and the output capacitance of the current source. In order to reduce the effect of channel length modulation, a cascode current source can be used. As shown in Fig. 5.4, if V_N is chosen such that $V_X = V_Y$ then I_{out} closely tracks I_{REF}. This is because the cascode device shields the bottom transistors from variations in V_P at node P. Therefore, we can conclude that V_Y remains close to V_X and hence $I_{D2} \approx I_{D1}$ with high accuracy. Such accuracy is obtained at the cost of the voltage headroom consumed by M_4. Note that while L_1 must be equal to L_2, the length of M_3 need not to be equal to L_1 and L_2.

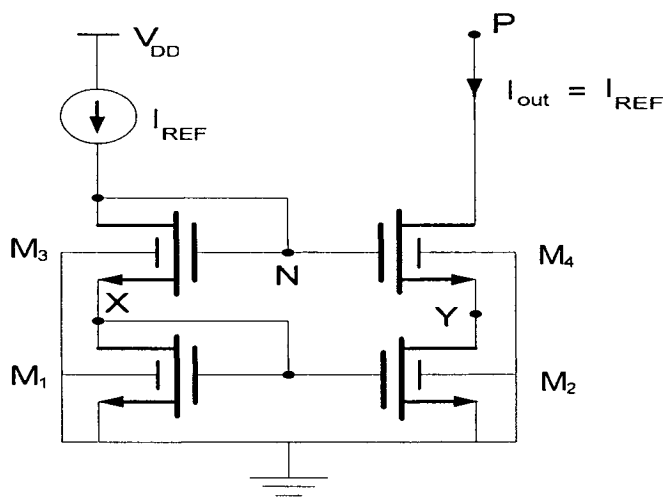


Figure 5.4: Cascode current mirror

From Fig. 5.4, we have $V_{GS3} + V_X = V_{GS4} + V_Y$. Thus, If $(W/L)_4/(W/L)_3 = (W/L)_2/(W/L)_1$, then $V_{GS3} = V_{GS4}$ and hence $V_X = V_Y$. Note that this result holds if M_3 and M_4 even suffer from body effect. For simplicity, we ignore the body effect and assume all of the transistors are identical. Then, minimum allowable voltage at node P is equal to

$$V_P = V_N - V_{th} = V_{GS3} + V_{GS1} - V_{th} = (V_{GS3} - V_{th}) + (V_{GS1} - V_{th}) + V_{th} \qquad (5.3)$$

i.e. two overdrive voltages plus one threshold voltage. The small-signal equivalent circuit of cascode current mirror is shown in Fig. 5.5.

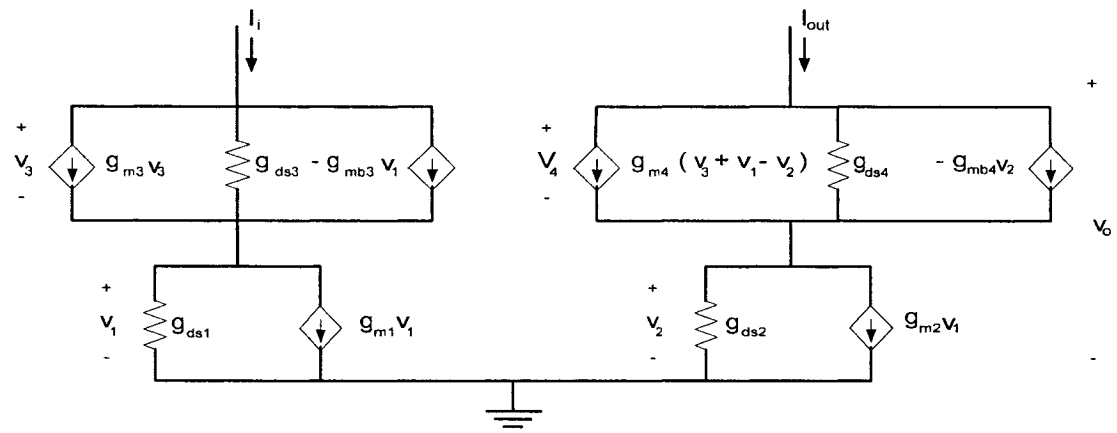


Figure 5.5: Small-signal equivalent circuit of cascode current mirror

In order to find the output impedance of cascode current mirror, one has to set $i_i = 0$, this makes $v_1 = v_3 = 0$. Then R_o can be derived as follows:

$$R_o = \frac{1}{g_{ds4}}[1 + \frac{1}{g_{ds2}}(g_{m4} + g_{mb4} + g_{ds4})] \qquad (5.4)$$

To design a cascode current mirror, suppose on being prompted by the CADT, the user provides the following data, V_P=2 V, V_{DD}=2 V, $V_X = V_Y$ = 1 V, and $I_{REF} = I_{out}$ = 30 μA for a 0.18 μm process technology to the cascode current mirror design subprogram of the CADT. Then, the software will process the database files and deliver its results for all

of NMOS transistors as shown in table 5.3. There could be more than one solution and the designer has to choose the best values of design parameters. As well, there could be no solution because design parameters entered by the designer may not find suitable match in the database.

Table 5.3: Design parameters of cascode current mirror

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{REF} (μA) Received by Tool	I_O (μA) Obtained from Simulation
NMOS (M_1 & M_2)	1	1	0	0.8	1.2	30	30.28
NMOS (M_3 & M_4)	1	1	-1	0.8	0.4		

5.3 CMOS voltage divider

The voltage divider submenu of the CADT is shown in Fig. 5.6. Four different series combinations of NMOS and PMOS transistors are supported as listed below.

```
--------- Voltage Divider Series Combination Design ---------

          (1) PMOS on Top and NMOS on Bottom
          (2) PMOS on Top and PMOS on Bottom
          (3) NMOS on Top and PMOS on Bottom
          (4) NMOS on Top and NMOS on Bottom
          (5) Return to Main Menu

Enter your choice [ 1, 2 , 3, 4 or 5 ]:
```

Figure 5.6: Voltage divider design submenu of the tool

The most common application of voltage dividers is in DC biasing. Fig. 5.7 shows a voltage divider consisting of two diode-connected transistors. These transistors behave as a series connection of two resistors in a potential divider.

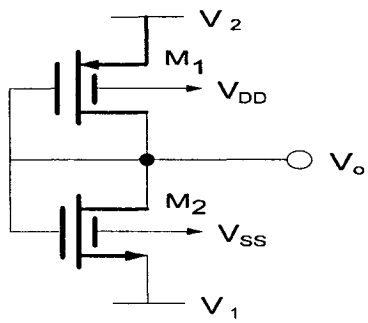


Figure 5.7: The basic voltage divider

Transistors M_1 and M_2 are always in saturation because their gate and drain are connected together. To design a voltage divider, the related subprogram of the tool receives a range of DC bias current, top and bottom voltages of the branch (V_2 and V_1) and DC level of the output voltage, and provides us with the transistor dimensions and other DC bias conditions. Note that both transistors are in series, thus they carry the same current. As a design example, suppose on being prompted by the CADT tool, the user provides the following data, $V_{DD}=V_2=$ 1.8 V, $V_{SS}=V_1=$ -1.8 V, $V_o=$ 0.5 V and I_{DS} between 40 μA to 50 μA (power dissipation less than 180 μW) to the voltage divider design subprogram of the tool for a 0.18 μm process technology. Then, the software will process the database files and deliver its results for both NMOS and PMOS transistors as shown in table 4.7. There could be more than one solution and the designer has to choose the best values of design parameters, for instance select a suitable case to minimize the area. As well, if exact values of the design parameters are absent in the database the software provides the nearest possible values.

Table 5.4: CMOS voltage divider design parameters

Device	V_{GS} (Volt) Input	V_{DS} (Volt) Output	V_{BS} (Volt) Input	W (μA) Output	L (μA) Output	I_{DS} (μA) Obtained by Tool	I_{DS} (μA) and V_o (Volt) Obtained from HSPICE
M_1	-1.3	-1.3	0	22	10	42.3749	42.3761 μA 0.5 Volt
M_2	2.3	2.3	0	2	16	42.3793	

HSPICE DC simulation results with above design parameters of the current mirror lead to almost the same drain-source currents. This shows the accuracy and reliability of the CADT.

5.4 Single stage amplifiers

Fig. 5.8 shows the single stage amplifier design submenu of the CADT. From this menu, the designer can choose to design popular kinds of single stage amplifiers. Each of the menu options also has several submenus that provide the possibility of the design of other versions of a single stage amplifier.

--------- Single Stage Amplifier Design ---------

(1) Common Source
(2) Common Drain
(3) Common Gate
(4) Cascode
(5) Push Pull
(6) Return to Main Menu

Enter your choice [1, 2, 3, 4, 5, or 6]:

Figure 5.8: Single stage amplifier design menu

In the following, we discuss the design of the most popular single stage amplifiers using the CADT.

5.4.1 Common-source amplifier with diode-connected load

The NMOS or PMOS transistor can operate as a small-signal resistor if its gate and drain are connected together. In many CMOS technologies, it is desirable to replace a resistor with a MOS transistor as shown in Fig. 5.9 (a).

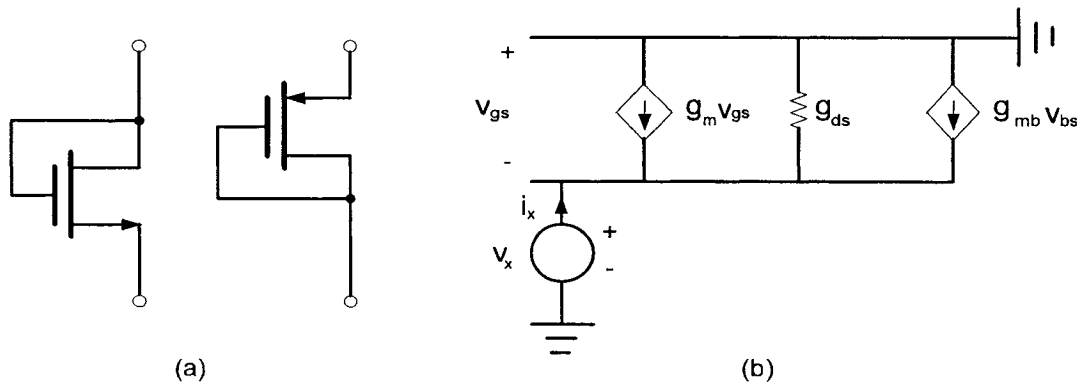


Figure 5.9: (a) Diode-connected NMOS and PMOS transistors, (b) small-signal equivalent circuit of NMOS transistor with body effect

From Fig. 5.9 (b), we can write:

$$\frac{v_X}{i_X} = \frac{1}{g_m + g_{ds} + g_{mb}} \tag{5.5}$$

If we ignore the body effect, then the impedance seen between drain and source of MOS device is higher and we have:

$$\frac{v_X}{i_X} = \frac{1}{g_m + g_{ds}} \approx \frac{1}{g_m} \tag{5.6}$$

A typical common-source (CS) stage with diode-connected load is shown in Fig. 5.10. An NMOS transistor (M_1) converts the variations of gate-source voltage to a small-

signal drain current which can pass through a diode-connected PMOS transistor (M_2) as a resistor to generate an output voltage.

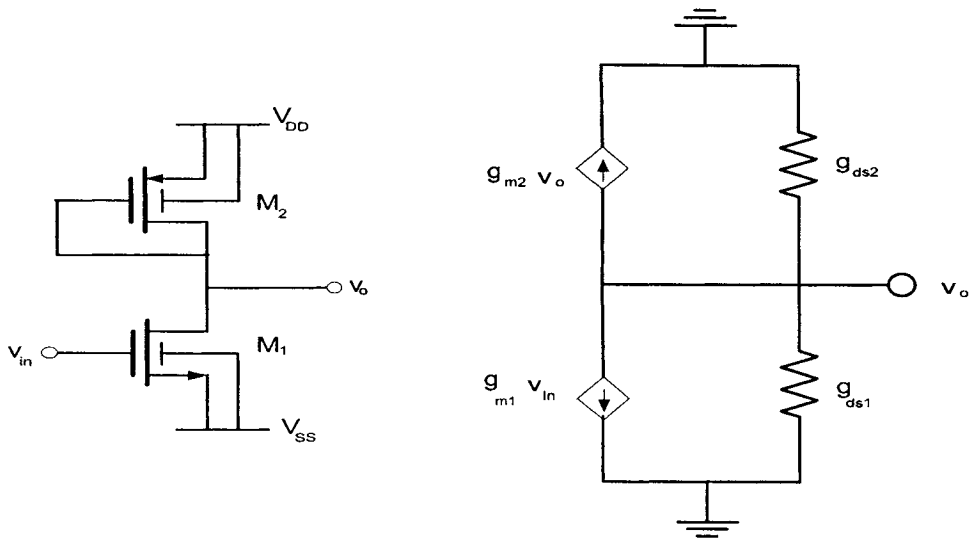


Figure 5.10: Common-source (CS) amplifier and its small-signal equivalent

The small-signal gain of CS amplifier with diode-connected load can be derived as follows:

$$\frac{v_o}{v_{in}} = \frac{g_{m1}}{g_{ds1} + g_{ds2} + g_{m2}} \tag{5.7}$$

When the supply voltages (V_{DD} and V_{SS}), power dissipation and the required gain are given, then the design problem is to find the transistors sizes and DC bias voltages so that the given specification can be met. The tool receives the supply voltages, DC level of output voltage and a certain range of DC bias current, and delivers dimensions and DC bias voltages of both transistors. We also use this fact that two or more transistors carry the same current when in series connection.

For example, suppose on being prompted by the CADT, the user provides the following data, V_{DD} = 1 V, V_{SS} = -1 V, voltage gain = 2.5 V/V, I_{DS} between 95 μA to 100 μA (power dissipation less than 200 μW) for a 0.5 μm process technology. The design parameters obtained from the CADT and the simulation results of the circuit with these parameters are listed in table 5.5. The simulation results confirm the capability of the software program to arrive at design values that produce results close to given specifications.

Table 5.5: CS amplifier design parameters with diode-connected load

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain from Simulation
NMOS (M_1)	0.65	1	0	1	0.6	97.8	2.4951 (V/V)	2.2428 (V/V)
PMOS (M_2)	-1	-1	0	1	3	97.1		

5.4.2 Common-source amplifier with current source load

In applications that we need a large voltage gain, a practical approach is to replace the diode-connected load with a current source as shown in Fig. 5.11. In this case, both transistors are operating in saturation.

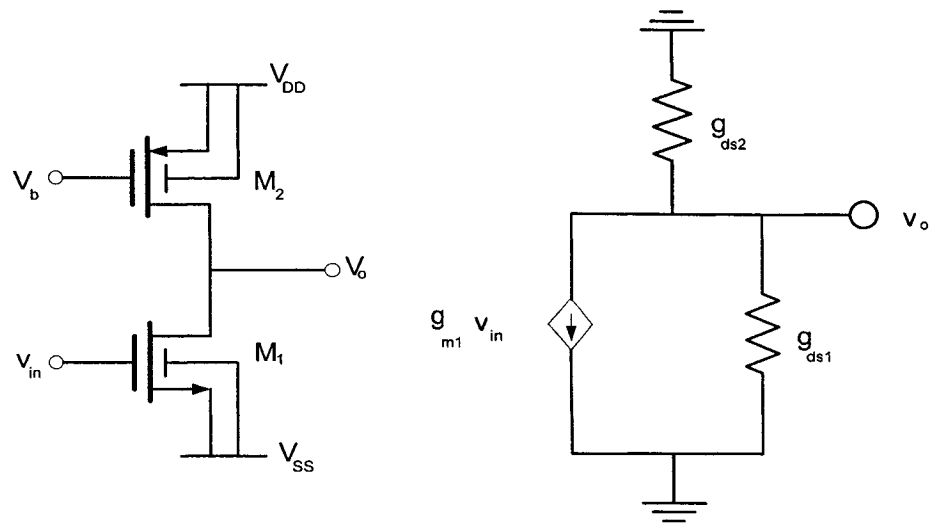


Figure 5.11: Common-source with current-source load and its small-signal equivalent

The small-signal voltage gain is given by:

$$A_v = \frac{v_o}{v_{in}} = -\frac{g_{m1}}{g_{ds1} + g_{ds2}} \tag{5.8}$$

We apply the voltage gain equation as a design equation to design a CS amplifier with current-source load through the CADT. The tool receives the supply voltages, DC level of output voltage and a certain range of DC bias current, and delivers dimensions and DC bias voltages of both transistors. We also use this fact that two transistors carry the same current when in series connection. Here, we present 3 examples as follows:

Example (1): Suppose on being prompted by the CADT, the user provides the following data, V_{DD} = 1 V, V_{SS} = -1 V, V_{out} = 0 V, voltage gain = 33 V/V, I_{DS} between 40 μA to 50 μA (power dissipation less than 100 μW) for a 0.5 μm process technology. The design parameters achieved from the CADT and the results of simulations with these parameters that prove accuracy of the tool are listed in table 5.6.

Table 5.6: CS amplifier design parameters with current source load of example 1

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain from Simulation
NMOS (M_1)	1.15	1	0	11	3	46.4145	32.9560 (V/V)	32.96 (V/V)
PMOS (M2)	-1.6	-1	0	6	1.4	46.4148		

Example (2): Suppose on being prompted by the CADT, the user provides the following data, V_{DD} = 3 V, V_{SS} = 0 V, V_{out} = 1 V, voltage gain = 50±20% V/V, I_{DS} between 25μA to 30μA (power dissipation less than 90μW) for a 0.18μm process technology. The design parameters achieved from the CADT and the results of simulations with these parameters that prove accuracy of the tool are listed in table 5.7.

Table 5.7: CS amplifier design parameters with current source load of example 2

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain from Simulation
NMOS (M_1)	0.5	1	0	6	0.2	27.7764	52.8160 (V/V)	50.3134 (V/V)
PMOS (M2)	-1	-2	0	7	2.2	27.7757		

Example (3): Suppose on being prompted by the CADT, the user provides the following data, $V_{DD} = 3$ V, $V_{SS} = 0$ V, $V_{out} = 1$ V, voltage gain = 100 $\pm 20\%$ V/V, I_{DS} between 10 μA to 15 μA (power dissipation less than 45 μW) for a 0.18 μm process technology. The design parameters achieved from the CADT and the results of simulations with these parameters that prove accuracy of the tool are listed in table 5.8.

Table 5.8: CS amplifier design parameters with current source load of example 3

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain from Simulation
NMOS (M_1)	0.55	1	0	3	0.4	14.1035	118.0361 (V/V)	116.2935 (V/V)
PMOS (M2)	-1.0	-2	0	3	1.8	14.1041		

5.4.3 Push-pull amplifier

In a push-pull topology as shown in Fig. 5.12, the gate of both NMOS (M_1) and PMOS (M2) are connected together. In comparing the common-source and push-pull amplifiers, it is seen that push-pull amplifier has a higher voltage gain.

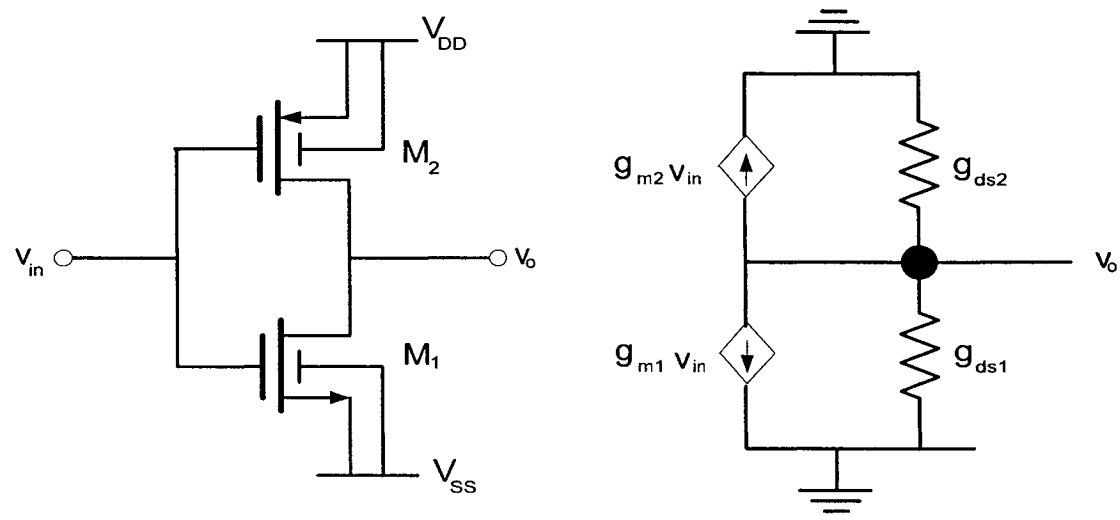


Figure 5.12: Push-pull amplifier and its small-signal equivalent

From the small-signal circuit, the voltage gain of a push-pull amplifier is derived as follows:

$$A_v = \frac{v_o}{v_{in}} = -\frac{g_{m1} + g_{m2}}{g_{ds1} + g_{ds2}} \tag{5.9}$$

The equation 5.9 is used to design a push-pull amplifier through the CAD tool. The software receives the supply voltages, DC level of output voltage and a certain range of DC bias current, and delivers dimensions and DC bias voltages of both transistors. We also use this fact that two transistors carry the same current when in series connection.

For example, suppose on being prompted by the CADT, the user provides the following data, V_{DD} = 1 V, V_{SS} = -1 V, gain = 235 V/V, I_{DS} between 10 μA to 15 μA (power dissipation less than 30 μW) for a 0.5 μm process technology. The design parameters obtained from the CADT and simulations of the circuit with these parameters that prove accuracy of the tool are listed in table 5.9.

Table 5.9: Push-pull amplifier design parameters

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain from Simulation
NMOS (M_1)	1	1	0	11	4.6	14.527	236.5888 (V/V)	234.05 (V/V)
PMOS (M_2)	-1	-1	0	1	36	14.527		

5.4.4 Common-drain amplifier

Common-drain amplifier also called the source follower can operate as a voltage buffer. This stage is placed after a high gain amplifier to drive low-impedance loads without considerable loss of signal level. Also, it can be used to perform voltage level shift. As shown in Fig. 5.13 the input signal applies to the gate and drives the load at the source, allowing the source voltage to follow the gate voltage.

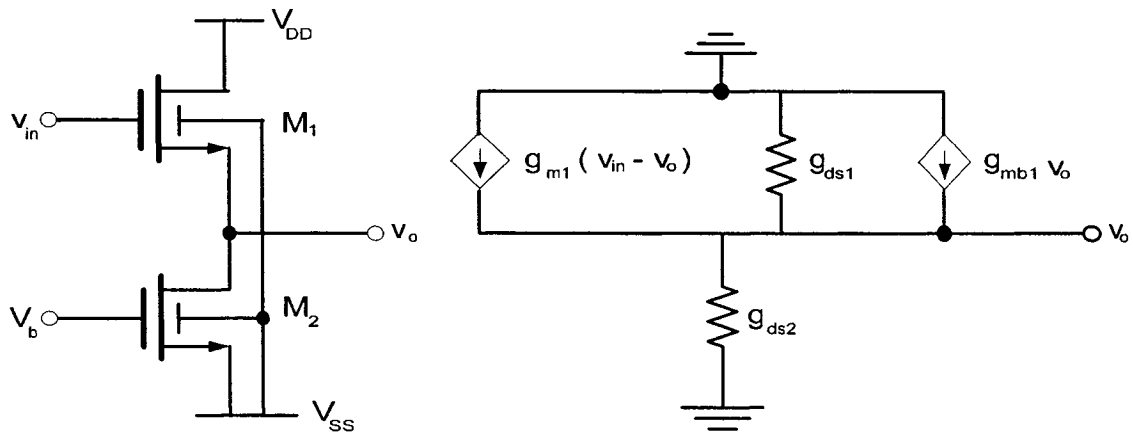


Figure 5.13: Common-drain amplifier and its small-signal equivalent

From the low-frequency small-signal equivalent circuit, the voltage gain can be derived as below:

$$\frac{v_o}{v_{in}} = -\frac{g_{m1}}{g_{ds1} + g_{ds2} + g_{m1} + g_{mb1}} \tag{5.10}$$

Equation 5.10 shows that the gain of common-drain amplifier is always less than unity. This design equation is used to design a common-drain amplifier through the CADT. The software receives the supply voltages, DC level of output voltage and a certain range of DC bias current and the required voltage gain, and provides us with dimensions and DC bias voltages of NMOS and PMOS transistors. We also use this fact that two transistors carry the same current when in series connection.

For example, suppose on being prompted by the CADT, the user provides the following data, $V_{DD} = 3$ V, $V_{SS} = 0$ V, gain $\approx$ 1 V/V, I_{DS} between 50 μA to 60 μA (power dissipation less than 180 μW) for a 0.18 μm process technology. The design parameters obtained from the CADT and the results of circuit simulation are listed in table 5.10.

Table 5.10: Common-drain amplifier design parameters

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain from Simulation
NMOS (M_1)	2	1	-1	4	13	56.2599	0.9059 (V/V)	0.7781 (V/V) at 55.8673 μA
NMOS (M_2)	0.8	2	0	14	6	56.2592		

We observe an error of 10 percent between the gain calculated by the tool and gain estimated by HSPICE which is due to errors in calculation of g_m, g_{ds} and g_{mb} as discussed in chapter 4.

5.4.5 Common-gate amplifier

In a common-gate (CG) amplifier the signal applies to the source and the output produces at the drain. The gate is connected to a DC bias voltage to establish proper operating conditions. The schematic and the small-signal equivalent circuit are shown in Fig. 5.14.

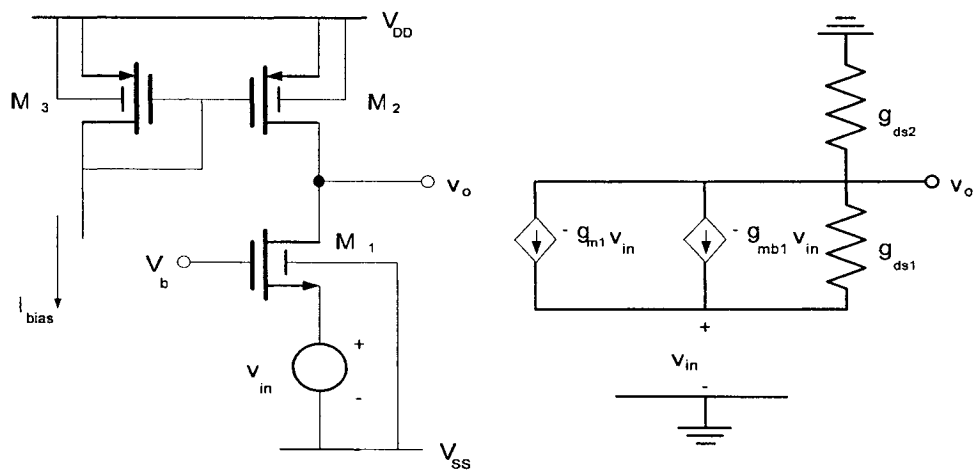


Figure 5.14: Common-gate amplifier and its small-signal equivalent circuit

From the small-signal equivalent circuit, we get the voltage gain as:

$$\frac{v_o}{v_{in}} = \frac{g_{m1} + g_{ds1} + g_{mb1}}{g_{ds1} + g_{ds2}} \tag{5.11}$$

Note that the gain is positive (no phase shift). The body effect increases the gain and decreases the input impedance of the common-gate stage. Equation 5.11 is used to design a common-gate amplifier through the CADT. The software receives the supply voltages, DC level of output voltage and a certain range of DC bias current and the

required voltage gain, and provides us with dimensions and DC bias voltages of NMOS and PMOS transistors. In the software, we also use this fact that two transistors carry the same current when in series connection.

For example, suppose on being prompted by the CADT, the user provides the following data, V_{DD} = 1.8 V, V_{SS} = 0 V, gain = 200 V/V , I_{DS} between 20 μA to 25 μA (power dissipation less than 45 μW) for a 0.18 μm process technology. The design parameters obtained from the CADT and simulation results of the circuit with these parameters are listed in table 5.11.

Table 5.11: Common-gate amplifier design parameters

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain from Simulation
NMOS (M_1)	0.8	1	0	10	10	23.8121	203.6065 (V/V)	288.5949 (V/V)
PMOS (M_2)	-1.1	-0.8	0	12	6	23.8126		

The HSPICE simulation shows that there is a 30 percent error in voltage gain calculation using the tool that occurs because of errors in calculation of the g_m, g_{ds} and g_{mb} as discussed in chapter 4. Note that the voltage gain obtained from AC simulation is bigger than the voltage gain predicted by the CADT.

5.4.6 CMOS common-drain amplifier

A CMOS common-drain amplifier and its small-signal equivalent circuit are shown in Fig. 5.15.

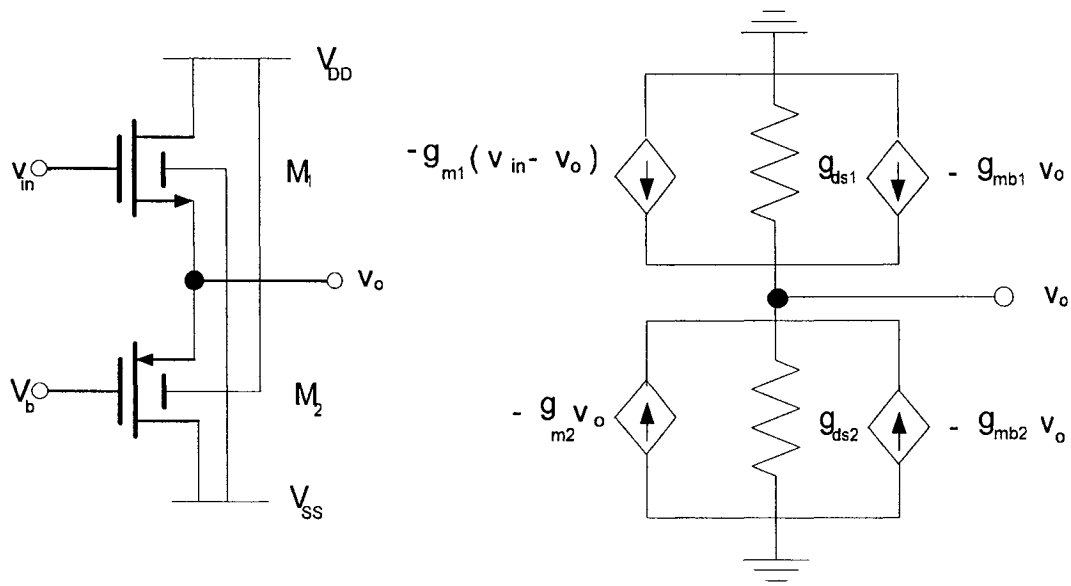


Figure 5.15: Common-drain amplifier and its small-signal equivalent circuit

From the small-signal equivalent circuit, we get the voltage gain as:

$$\frac{v_o}{v_{in}} = \frac{g_{m1}}{g_{m1} + g_{m2} + g_{ds1} + g_{ds2} + g_{mb1} + g_{mb2}} \tag{5.12}$$

Note that the gain is positive (no phase shift). Equation 5.12 is used to design a common-drain amplifier through the CADT. The software receives the supply voltages, DC level of output voltage and a certain range of DC bias current and the required voltage gain, and provides us with dimensions and DC bias voltages of NMOS and

PMOS transistors. We also use this fact that two transistors carry the same current when in series connection.

For example, suppose on being prompted by the CADT, the user provides the following data, $V_{DD} = 1$ V, $V_{SS} = -1$ V, a gain close to 0.6 V/V, DC output voltage equal to 0 Volt, I_{DS} between $70\,\mu A$ to $80\,\mu A$ (power dissipation less than $160\,\mu W$) for a $0.18\,\mu m$ process technology. After data processing, the design parameters obtained from the CADT and simulation results of the circuit with these parameters that confirm reliability of the tool are listed in table 5.12.

Table 5.12: CMOS common-drain amplifier design parameters

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain Tool	Voltage Gain from Simulation
NMOS (M_1)	1	1	-1	2.5	0.5	74.8723	0.6407 (V/V)	0.6421 (V/V)
PMOS (M_2)	-1.8	-1	1	2	0.6	74.9512		

To show the accuracy of our new design method in more detail, we calculate gain of the common-drain amplifier (Fig. 5.15) using equation 5.12, and g_{m1}, g_{ds1}, g_{mb1}, g_{m2},

g_{ds2}, and g_{mb2} values obtained from the CADT and HSPICE simulation as listed in table 5.13.

Table 5.13: Parameters obtained from CADT and HSPICE

Tool	g_{m1} $(\frac{\mu A}{V})$	g_{ds1} $(\frac{\mu A}{V})$	g_{mb1} $(\frac{\mu A}{V})$	g_{m2} $(\frac{\mu A}{V})$	g_{ds2} $(\frac{\mu A}{V})$	g_{mb2} $(\frac{\mu A}{V})$	Voltage Gain (V/V)
CADT	435.8807	3.8548	87.1318	109.7703	8.9001	34.7543	0.6407
HSPICE	436.1983	3.8564	86.0230	109.7800	8.8953	34.5090	0.6421

The values of parameters obtained from CADT and HSPICE simulation are very close to each other. This confirms the reliability of small-signal analysis (proposed model of MOS transistors) and our new design method as well.

5.5 Summary

In this chapter, we have shown how the new design methodology and our CAD tool (CADT) can be used to design a variety of small scale analog IC building blocks for different process technologies i.e. 0.5 μm and 0.18 μm. The presented HSPICE simulation results reveal the usefulness and the reliability of the CADT.

Chapter 6

Medium Scale Building Blocks

Design Examples

This chapter presents the design of medium sized analog integrated circuit building blocks using the new design methodology and the CAD tool which are described in chapter 4.

6.1 Introduction

We study the design of medium scale analog IC building blocks in this chapter. The schematic of medium sized building blocks along with their design procedures using our CAD tool (CADT) are presented.

Design parameters obtained from the CADT are tested by HSPICE DC and/or AC analysis to prove the accuracy and reliability of the tool.

6.2. Triple cascode stage

The cascade of a common-source stage and common-gate stage is called a cascode topology. Fig. 6.1 shows the basic configuration of a cascode amplifier. M_1 generates a small-signal drain current proportional to v_{in} and M_2 routes this current to M_3 as a current source load. Note that in this example, M_1, M_2 and M_3 carry equal currents and also M_2 reduces the output voltage swing by at least the overdrive voltage of M_2 compared with a common-source stage.

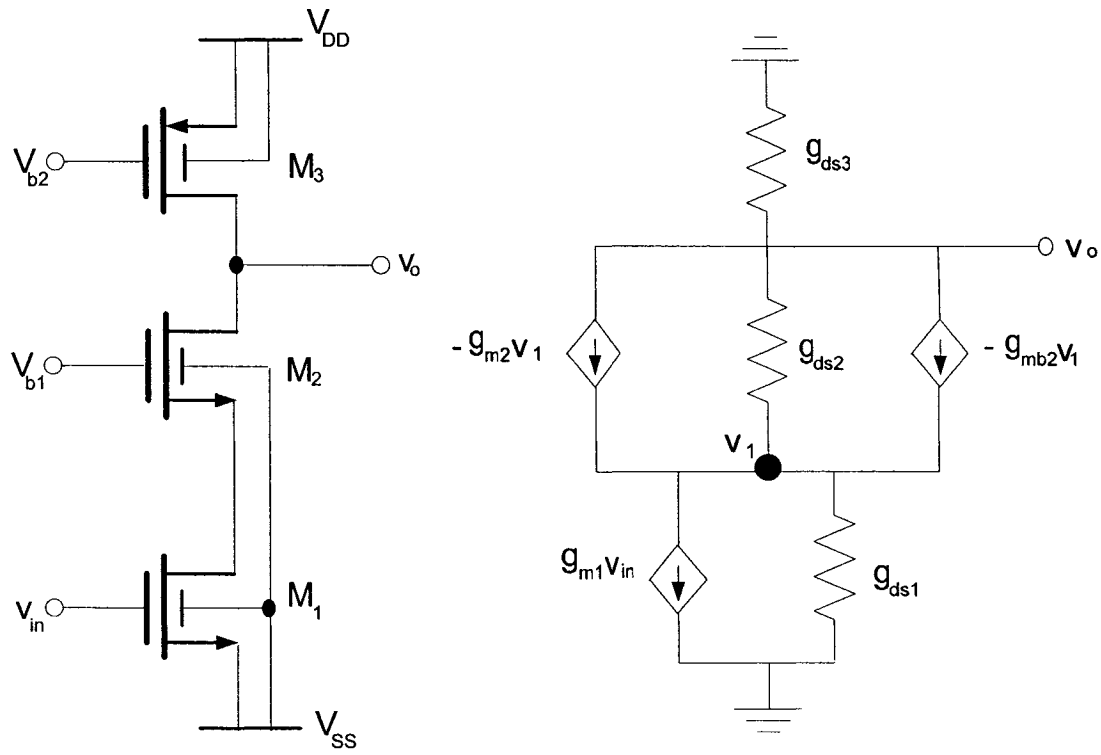


Figure 6.1: The triple Cascode stage and its small-signal equivalent

From the small-signal equivalent circuit, we get the voltage gain as:

$$\frac{v_o}{v_{in}} = -\frac{g_{m1}(g_{m2}+g_{ds2}+g_{mb2})}{(g_{ds1}+g_{m2}+g_{mb2})(g_{ds2}+g_{ds3})+g_{ds2}(g_{mb2}+g_{m2}-g_{ds3})} \qquad (6.1)$$

Equation 6.1 is used to design a triple cascode amplifier through the CADT. The software is provided with data about the supply voltages, DC level of output voltage and a certain range of DC bias current and the required voltage gain, and provides us with dimensions and DC bias voltages of NMOS and PMOS transistors. Here, we present two examples of the triple cascode amplifier design using the CADT.

Example (1): Suppose on being prompted by the CADT, the user provides the following data, V_{DD} = 1.5 V, V_{SS} = -1.5 V, gain = 50 ± 20%, V_{out} = 0.5 V, I_{DS} between 18.5 μA to 19 μA (power dissipation less than 60 μW) for a 0.18 μm process technology. The design parameters obtained from the CADT and simulation results of the circuit with these parameters that show usefulness of the tool are listed in table 6.1.

Table 6.1: Triple cascode amplifier design parameters

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain from Simulation
NMOS (M_1)	0.8	1	0	3	3.6	18.776	55.3937 (V/V)	55.01 (V/V) at 18.78 μA
NMOS (M_2)	0.8	1	-1	8	1.2	18.847		
PMOS (M_3)	-0.65	-1	0	5.5	0.4	18.814		

Example (2): Suppose on being prompted by the CADT, the user provides the following data, V_{DD} = 1.5 V, V_{SS} = -1.5 V, gain = 100 ± 20%, V_{out} = 0.5 V, I_{DS} between 20 μA to 21 μA (power dissipation less than 63 μW) for a 0.18 μm process technology. The design parameters obtained from the CADT and simulation results of the circuit with these parameters that show usefulness of the tool are listed in table 6.2.

Table 6.2: Triple cascode amplifier design parameters

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain from Simulation
NMOS (M_1)	0.8	1	0	3.5	3.8	20.945	106.5788 (V/V)	175.20 (V/V) at 20.95 μA
NMOS (M_2)	0.8	1	-1	6	0.8	20.577		
PMOS (M_3)	-1.55	-1	0	3.5	4.8	20.903		

6.3. Common-source amplifier with cascode load stage

Fig. 6.2 shows a common-source amplifier with cascode load stage and the associated small-signal equivalent circuit. This is used for class AB operation in the output stage of some amplifiers. The two series connected transistors M_2 and M_3, which are configured as active resistors, act as a load for input stage (M_1) and provide DC bias for the next stage. From the small-signal equivalent circuit, we get the voltage gain as:

$$\frac{v_o}{v_{in}} = -\frac{(g_{m2}+g_{ds2})(g_{m3}+g_{ds3}+g_{ds4})+g_{ds4}(g_{m3}+g_{ds3})}{g_{ds1}(g_{m2}+g_{ds2})(g_{m3}+g_{ds3}+g_{ds4})+g_{ds4}(g_{m3}+g_{ds3})(g_{ds1}+g_{ds2}+g_{m2})} \qquad (6.2)$$

Equation 6.2 is used to design this amplifier through the CADT. The software receives the supply voltages, DC level of output voltage, V_1 and V_2 DC bias voltages and a certain range of DC bias current and the required voltage gain, and provides us with

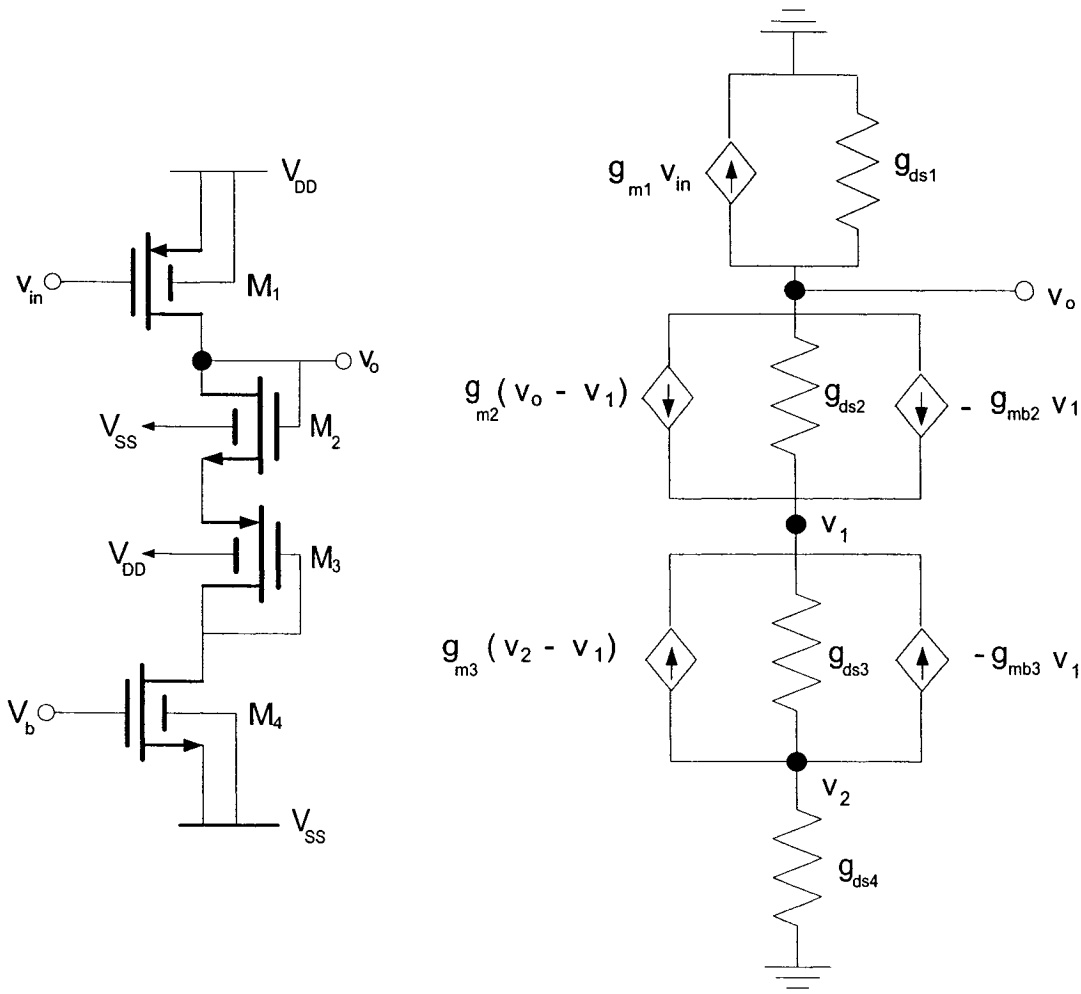


Figure 6.2: Common-source with cascode load and its small-signal equivalent

dimensions and DC bias voltages of NMOS and PMOS transistors. Note that all transistors carry the same current.

For example, suppose on being prompted by the CADT, the user provides the following data, V_{DD} = 1.8 V, V_{SS} = -1.8 V, V_{out} = 1 V, gain = 45 V/V, I_{DS} between 40 μA to 50 μA (power dissipation less than 180 μW) for a 0.18 μm process technology. The design parameters obtained from the CADT and simulation results of the circuit with these parameters that show reliability of the tool are listed in table 6.3.

Table 6.3: Design parameters of common-source stage with cascode load

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain From Simulation
PMOS (M_1)	-0.8	-0.8	0	0.5	0.5	3.2556	45.6001 (V/V)	47.9301 (V/V) at 3.2622 μA
NMOS (M_2)	1	1	-1.8	1.5	2.5	3.2304		
PMOS (M_3)	-1	-1	1.8	4	0.5	3.2337		
NMOS (M_4)	0.5	0.8	0	5	2	3.2654		

6.4 Differential amplifiers

The differential amplifiers are one of the most important cells in today's high performance analog circuits. We present two design examples of differential pairs with current mirror and current-source loads in this section. Fig. 6.3 shows the differential amplifier design submenu of the tool.

```
--------- Differential Amplifier Design ---------

(1) Diff. Amp. with Current Source/Mirror Load
(2) Diff. Amp. with Active Load
(3) Return to Main Menu

Enter your choice [ 1, 2 or 3 ]:
```

Figure 6.3: Differential amplifier design menu

6.4.1 Single-ended differential amplifier

Fig. 6.4 shows a differential amplifier with active current mirror load.

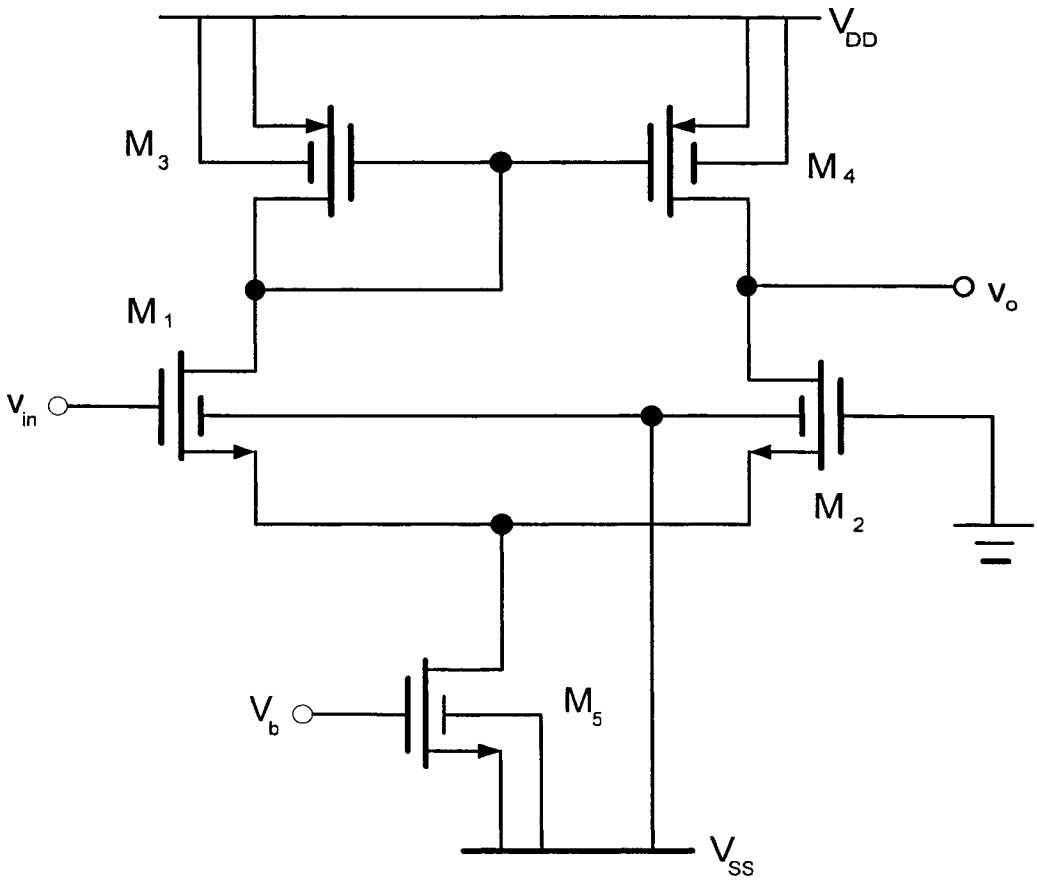


Figure 6.4: Differential amplifier with active current mirror load

To design this circuit through our new method, the first step is to calculate the small-signal, low frequency voltage gain. The simplified small-signal model as shown in Fig. 6.5 is used for this purpose. Note that we consider the body effect of both M_1 and M_2 transistors.

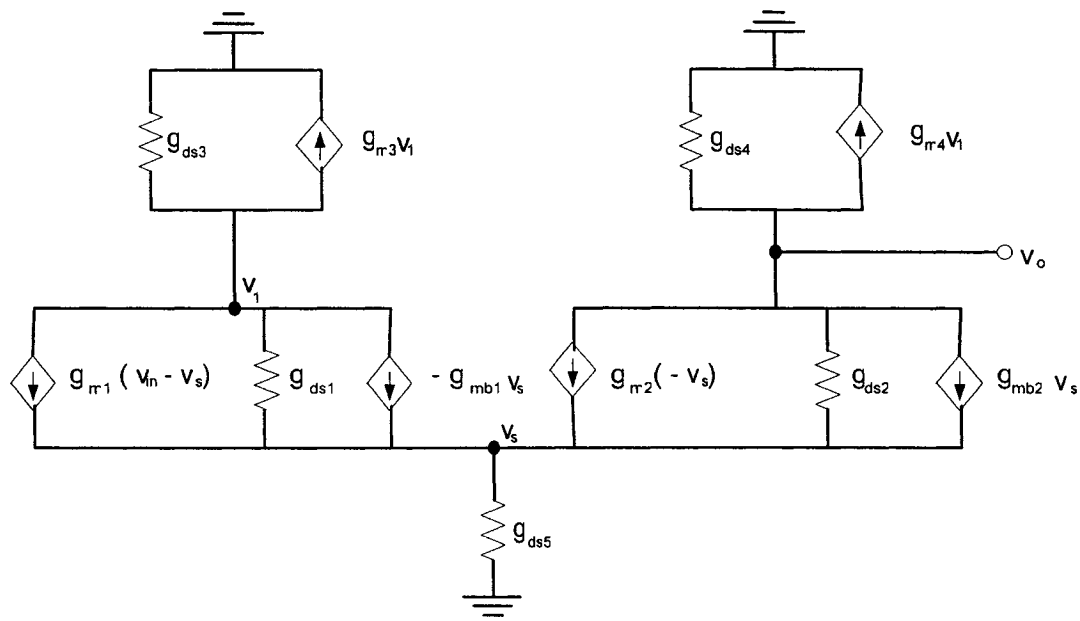


Figure 6.5: Differential amplifier small-signal equivalent circuit

From Fig 6.5, we get:

$$v_1(g_{ds1} + g_{m3} + g_{ds3}) - v_s(g_{m1} + g_{ds1} + g_{mb1}) + v_{in}g_{m1} = 0 \tag{6.3}$$

$$v_o(g_{ds4} + g_{ds2}) - v_s(g_{m2} + g_{ds2} + g_{mb2}) + v_1 g_{m4} = 0 \tag{6.4}$$

$$v_o g_{ds2} + v_{in} g_{m1} - v_s(g_{m1} + g_{ds1} + g_{mb1} + g_{m2} + g_{ds2} + g_{mb2} + g_{ds5}) + v_1 g_{ds1} = 0 \tag{6.5}$$

Since M_1 and M_2, and M_3 and M_4 are matched transistors, we have:

$g_{m1} = g_{m2}$ $\quad$ $g_{ds1} = g_{ds2}$ $\quad$ $g_{mb1} = g_{mb2}$

$g_{ds3} = g_{ds4}$ $\quad$ $g_{m3} = g_{m4}$

From equations 6.3, 6.4, and 6.5, after simplifying we obtain the small-signal gain as follows:

$$A_v = \frac{v_o}{v_{in}} = \frac{g_{m2}(Ag_{ds2} - Ag_{m4} - AB - g_{ds5}g_{m4})}{Ag_{ds2}(B + C - g_{m4}) - BC(2A + g_{ds5})} \tag{6.6}$$

Where

$A = g_{m1}+g_{ds1}+g_{mb1} = g_{m2}+g_{ds2}+g_{mb2}$

$B = g_{m4}+g_{ds4}+g_{ds2}$

$C = g_{ds2}+g_{ds4}$

We use equation 6.6 as a design equation in the CADT to calculate the voltage gain of the differential amplifier with active current mirror and to find the design variables. The software receives the supply voltages, DC level of output voltage, V_S (DC bias voltage at source of M_1), and a certain range of DC bias current and the required voltage gain. The CADT provides us with dimensions and DC bias voltages of NMOS and PMOS transistors. Note that M_1, M_2, M_3 and M_4 transistors carry the same current and M_5 carries a DC bias current twice of them. As well, input transistors ($M_1 \& M_2$), and load transistors ($M_3 \& M_4$) are matched. These constraints must be met to determine the design parameters through the CADT.

For example, suppose on being prompted by the CADT, the user provides the following data, V_{DD} = 1.5 V, V_{SS} = -1.5 V, gain = 350 V/V, V_{out} = 0.5 V, I_{DS} between 5 μA to 10 μA (power dissipation less than 60 μW), V_S = -0.7 V for a 0.18 μm process technology. The design parameters obtained from the CADT and simulation results of the circuit with these parameters that show accuracy of the tool are listed in table 6.4.

Table 6.4: Design parameters of single-ended differential amplifier

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain from Simulation
NMOS (M_1 & M_2)	0.7	1.2	-0.8	10	1.9	7.4252	357.9681 (V/V)	322.9500 (V/V)
PMOS (M_3 & M_4)	-1	-1	0	1.2	1.7	7.4304		
NMOS (M_5)	0.7	0.8	0	1.6	1.1	14.8606		

6.4.2 Differential amplifier with current-source load

As common-source stages, differential pairs can also employ current-source load. Fig.6.6 shows a differential amplifier with current-source load. To design a differential amplifier with current-source load using our new method, the first step is to calculate the small-signal, low-frequency voltage gain. The simplified small-signal model, as shown in Fig. 6.7, is used for this purpose. Note that we consider the body effect of both M_1 and M_2.

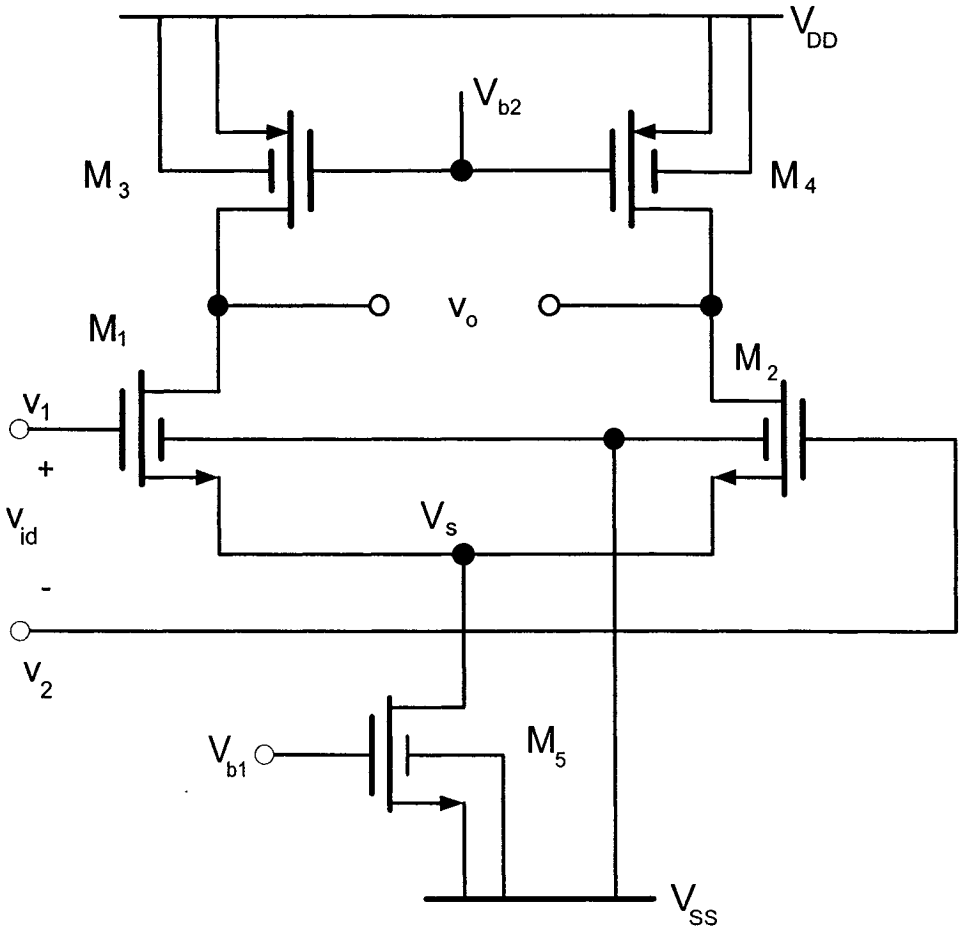


Figure 6.6: Differential amplifier with current-source load

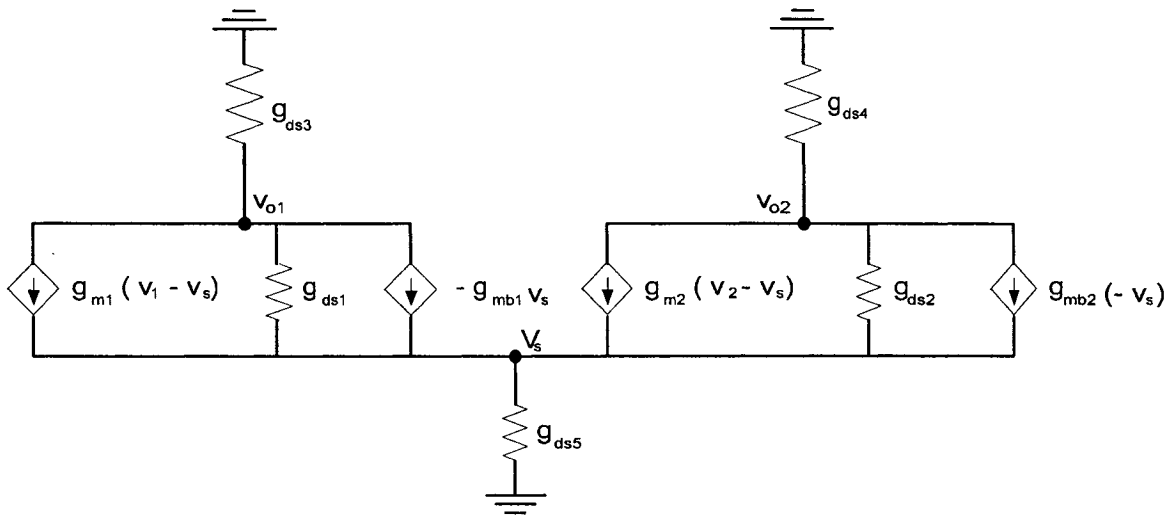


Figure 6.7: Small-signal equivalent circuit of differential amplifier

From Fig. 6.7, we get:

$$v_{o1}(g_{ds1}+g_{ds3})-v_s(g_{m1}+g_{ds1}+g_{mb1})+v_1 g_{m1}=0 \tag{6.7}$$

$$v_{o2}(g_{ds2}+g_{ds4})-v_s(g_{m2}+g_{ds2}+g_{mb2})+v_2 g_{m2}=0 \tag{6.8}$$

Since M_1 and M_2, and M_3 and M_4 are matched transistors, we have:

$g_{m1}=g_{m2}$ $\quad$ $g_{ds1}=g_{ds2}$ $\quad$ $g_{mb1}=g_{mb2}$

$g_{ds3}=g_{ds4}$ $\quad$ $g_{m3}=g_{m4}$

By subtracting equations 6.7 from 6.8 and considering matching conditions, after simplifying we obtain the small-signal gain as follows:

$$A_v=\frac{v_{od}}{v_{id}}=\frac{v_{o2}-v_{o1}}{v_2-v_1}=-\frac{g_{m2}}{(g_{ds2}+g_{ds4})} \tag{6.9}$$

We use equation 6.9 in the CADT to calculate the voltage gain of the differential amplifier with active current-source load and to find the design variables. The software receives the supply voltages, DC level of output voltage, a certain range of DC bias current, the drain voltage of M_5 and the required voltage gain, and provides us with dimensions and DC bias voltages of NMOS and PMOS transistors in this topology.

For instance, suppose on being prompted by the CADT, the user provides the following data, V_{DD} = 1 V, V_{SS} = -1 V, V_S = -0.5 V, gain = 100 V/V, V_{out} = 0.5 V, I_{DS} between 40 μA to 50 μA for tail current source (power dissipation less than 100 μW) for a 0.18 μm process technology. The design parameters obtained from the CADT and

simulation results of the circuit with these parameters that show usefulness of the tool are listed in table 6.5.

Table 6.5: Design parameters of differential amplifier with current-source loads

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain from Simulation
NMOS (M_1 & M_2)	0.8	1	-0.5	9.5	4.5	22.2897	117.1501 (V/V)	101.6959 (V/V) at 44.6968 μA
PMOS (M_3 & M_4)	-0.9	-0.5	0	7.5	2.0	22.2903		
NMOS (M_5)	0.6	0.5	0	6	0.5	44.5606		

6.5 Summary

In this chapter, the new design methodology and our CAD tool (CADT) are used to design a variety of medium scale analog IC building blocks for 0.18 μm process technology. The HSPICE simulation results prove the usefulness and the reliability of the CADT for the design of medium scale analog building blocks.

Chapter 7

Large Scale Building Blocks

Design Examples

This chapter presents the design of large sized analog integrated circuit building blocks using the new design methodology and the CAD tool which are described in chapter 4.

7.1 Introduction

Here, we study the design of two large scale analog IC building blocks including an Operational Amplifier (OPAMP) and Operational Transconductance Amplifier (OTA). The schematic of two large scale building blocks along with their design procedures using our CAD tool (CADT) are presented.

Design parameters obtained from the CADT are tested by HSPICE DC and/or AC analysis to prove the accuracy and reliability of the tool.

7.2 Large sized design examples

In this section, we present the design of an OPAMP and OTA as the large sized building block design examples. Fig. 7.1 shows the corresponding submenu of the CADT.

-------- Other Building Blocks --------

(1) OPAMP
(2) OTA
(3) Return to Main Menu

Enter your choice [1, 2 or 3]:

Figure 7.1: Large sized building block submenu of the CADT

7.2.1 Operational amplifier

An operational amplifier (OPAMP) is used in many analog circuits and systems and provides many signal processing functions. The design of OPAMP continues to be a challenging task as the supply voltage and transistor channel lengths scale down with each generation of CMOS technologies. OPAMP usually has four major sections including a differential amplifier, an intermediate high gain amplifier, the output buffer stage and the biasing section. Fig. 7.2 shows a sample OPAMP as a large sized building block design example [34].

In Fig. 7.2, the first stage, differential amplifier is an essential part and provides most of the open loop gain of the OPAMP. The second stage, an intermediate common-source amplifier boost up the gain and performs DC level shift so that the DC level at the output of OPAMP is nearly zero volt. The third stage is a power amplifier that is required to drive low resistance loads.

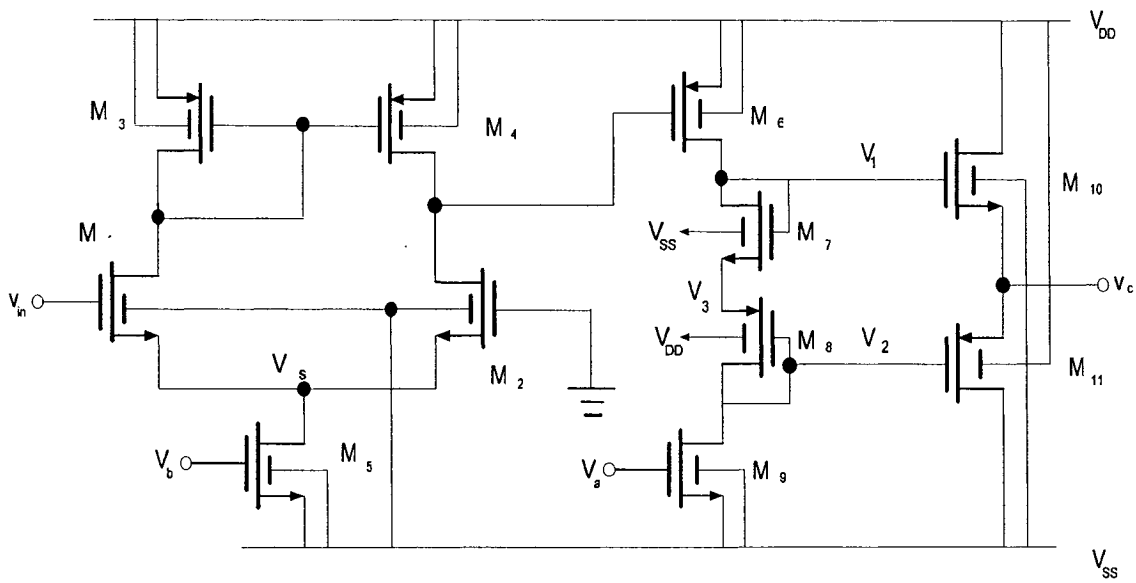


Figure 7.2: A three-stage operational amplifier

All three stages in Fig. 7.2 are discussed in sections, 6.4.1, 6.3 and 5.4.6 respectively. Therefore, we employ the procedures that have been discussed before to design each stage of OPAMP separately and then connect them together to build the full circuit. To do so, we first need to derive design equations of the OPAMP using the small-signal analysis. The overall open-loop voltage gain of the proposed OPAMP can be obtained by multiplication of the gain of all three stages as formulated in equation 7.1.

$$A_v = \frac{v_o}{v_{in}} = A_1 \times A_2 \times A_3 \tag{7.1}$$

Where,

$$A_1 = -\frac{g_{m2}(Ag_{ds2} - Ag_{m4} - AB - g_{ds5}g_{m4})}{Ag_{ds1}(B + C - g_{m4}) - BC(2A + g_{ds5})} \tag{7.2}$$

$$A_2 = \frac{D \times E + g_{ds4} \times F}{g_{ds1} \times D \times E + g_{ds4} \times F \times G} \tag{7.3}$$

$$A_3 = \frac{v_o}{v_{in}} = \frac{g_{m1}}{g_{m1} + g_{m2} + g_{ds1} + g_{ds2} + g_{mb1} + g_{mb2}} \tag{7.4}$$

$A = g_{m1}+g_{ds1}+g_{mb1} = g_{m2}+g_{ds2}+g_{mb2}$, $B = g_{m4}+g_{ds4}+g_{ds2}$, $C = g_{ds2}+g_{ds4}$

$D = (g_{m2} + g_{ds2})$, $E = (g_{m3} + g_{ds3} + g_{ds4})$, $F = (g_{m3} + g_{ds3})$, and $G = (g_{ds1} + g_{ds2} + g_{m2})$

Here, our goals are to design an OPAMP with the following specifications:

a) Power dissipation of less than 306 μW

b) DC gain between 65 dB and 70 dB

c) DC output voltage very close to zero

To design the first stage of the OPAMP, suppose on being prompted by the CADT, the user provides the following data, V_{DD}= 1.8 V, V_{SS}= -1.8 V, V_S= -0.8 V, V_{out} = 1 V, gain = 95 to 100 V/V, I_{DS} of each branch between 10 μA and 12 μA (power dissipation less than 90 μW) for a 0.18 μm process technology. The design parameters obtained from the CADT and simulation results for first stage of the OPAMP are presented in table 7.1.

Table 7.1: Design parameters of differential amplifier (OPAMP first stage)

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain From Tool	Voltage Gain from Simulation
NMOS (M_1 & M_2)	0.8	1.8	-1	2.5	0.6	10.818	98.0454 (V/V)	97.84 (V/V) at 10.84 μA
PMOS (M_3 & M_4)	-0.8	-0.8	0	1.6	0.5	10.819		
NMOS (M_5)	0.8	1	0	0.6	0.5	21.675		

To design the second stage of the OPAMP, suppose on being prompted by the CADT, the user provides the following data, V_{DD}= 1.8 V, V_{SS}= -1.8 V, V_1 = 1.12 V, V_2 = -0.85 V, gain = 30 - 35 V/V, I_{DS} between 5 μA to 10 μA (power dissipation less than 36 μW) for a 0.18 μm process technology. The design parameters obtained from the CADT and simulation results for the second stage of OPAMP with these parameters are listed in table 7.2.

Table 7.2: Design parameters of the second stage of the OPAMP

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain From Tool	Voltage Gain From Simulation
PMOS (M_6)	-.81	-.68	0	1	0.4	8.778	31.8119 (V/V)	33.7135 (V/V) at 8.783 μA
NMOS (M_7)	1.02	1.02	-1.8	1	0.65	8.827		
PMOS (M_8)	-.95	-.95	1.8	13	0.3	8.779		
NMOS (M_9)	0.75	0.95	0	0.5	0.85	8.783		

To design the third stage of the OPAMP, suppose on being prompted by the CADT, the user provides the following data, $V_{DD} = 1.8$ V, $V_{SS} = -1.8$ V, $V_{out} \approx 0$V , gain = 0.8 - 1 V/V, I_{DS} between 30 μA to 50 μA (power dissipation less than 180 μW) for a 0.18 μm process technology. The design parameters obtained from the CADT and simulation results for the second stage of OPAMP with these parameters are listed in table 7.3.

Table 7.3: Design parameters of the third stage of the OPAMP

Transistor	V_{GS} (V)	V_{DS} (V)	V_{BS} (V)	W (μm)	L (μm)	I_{DS} (μA)	Voltage Gain from Tool	Voltage Gain from Simulation
NMOS (M_{10})	0.78	1.8	-1.8	9.4	0.2	31.603	0.8851 (V/V)	0.8587 (V/V) at 31.8 μA
PMOS (M_{11})	-1.19	-1.8	1.8	9.2	0.7	31.595		

The overall OPAMP performance parameters obtained by HSPICE simulation show that our goals are met. These parameters are listed in table 7.4.

Table 7.4: Overall simulation results of OPAMP

Parameter	Value	Unit
Power Supply	±1.8	V
Power Dissipation	223.7	μW
DC Gain	69.45	dB
Output DC Voltage	-0.13	mV

7.2.2 Operational transconductance amplifier (OTA)

A linear, fully balanced, voltage-tunable CMOS OTA with large DC gain and wide bandwidth is shown in Fig. 7.3. This OTA is presented by Stanislaw Szczepanski [35] and uses two differential-pair transconductor with a cross-coupled input stage together with a negative resistance load for compensating the parasitic output resistance of the OTA.

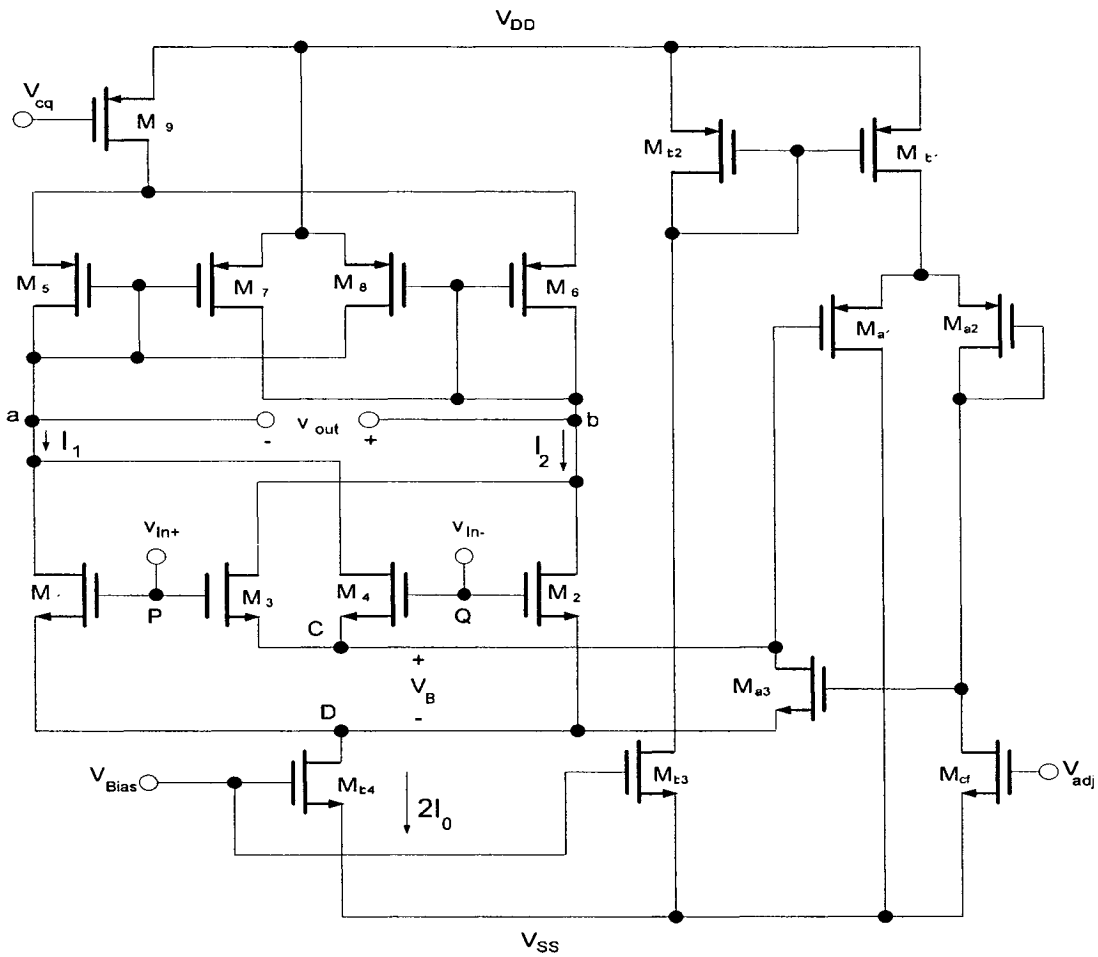


Figure 7.3: The schematic of Szczepanski's OTA

In Fig. 7.3, two cross-coupled differential pairs with identical MOS devices M_1, M_2, M_3, and M_4 operate in the saturation region. Both pairs are biased by a DC current $2I_0$ in combination with an adjustable floating voltage source V_B with low output resistance, r_{outB}. V_B is connected between the common-source nodes C and D. Thus, this topology of transconductor circuit (ideally with $r_{outB} = 0$) does not introduce additional internal nodes, resulting in improved high-frequency performance. In the range of operation $2I_0$ is assumed constant and tuning is achieved by adjusting V_B.

Using the standard square-law model for MOS devices, the currents I_1 and I_2 can be derived as follows:

$$I_1 = I_0 + \frac{I_{out}}{2} = k_n (V_P - V_{th})^2 + k_n (V_Q - V_B - V_{th})^2 \tag{7.5}$$

$$I_2 = I_0 - \frac{I_{out}}{2} = k_n (V_Q - V_{th})^2 + k_n (V_P - V_B - V_{th})^2 \tag{7.6}$$

Where, $k_n = 0.5\mu_n C_{ox} \frac{W}{L}$ is the transconductance parameter, and μ_n, C_{ox}, W, and L are the mobility, oxide capacitance per unit area, channel width and channel length respectively. V_{th} is the threshold voltage, I_{out} is the small-signal differential output current, and V_P and V_Q are the gate-source voltages of M_1 and M_2, respectively. Using equations 7.5 and 7.6, then I_{out} can be expressed as follows:

$$I_{out} = I_1 - I_2 = 2k_n V_B V_{id} \tag{7.7}$$

$V_{id} = V_{in^+} - V_{in^-}$ is the differential input voltage. Thus, the presented circuit exhibits a linear transconductance of value $G_m = 2k_n V_B$ which is tunable with DC voltage V_B. In practice, r_{outB} is never equal to zero, thus I_{out} will become nonlinear as:

$$I_{out} = 2k_n [V_B + r_{outB}(I_{d3} + I_{d4})]V_{id} \tag{7.8}$$

I_{ds3} and I_{ds4} are the drain currents of M_3 and M_4 devices, respectively. The sum of theses currents is given by:

$$I_{ds3} + I_{ds4} = I_0 - k_n V_B \sqrt{\frac{2I_0}{k_n} - V_B^2 - V_{id}^2} \tag{7.9}$$

It is assumed that the input signal is fully balanced around a common-mode value V_{ic}. It can be proved that for M_1-M_4 to operate in saturation the linear range of V_{id} is limited by

$$|V_{id}| \le \sqrt{\frac{I_0}{k_n} - \frac{3V_B^2}{4}} - \frac{V_B}{2} \tag{7.10}$$

The current source $2I_0$ operates in saturation if, $V_{ic} \ge V_{th} + V_B + V_{DSat}$, where V_{DSat} is the minimum voltage required for the current source $2I_0$ to operate in saturation. M_5, M_6, M_7, and M_8 are act as the voltage-controlled negative resistance load (NRL) to the transconductor. The NRL cancels out the small-signal drain-source resistance of M_1, M_2, M_3, and M_4, and boosts the output resistance of the transconductor to infinite theoretically.

The complete half-circuit small-signal equivalent circuit and its simplified are shown in Fig. 7.4 and Fig. 7.5, respectively.

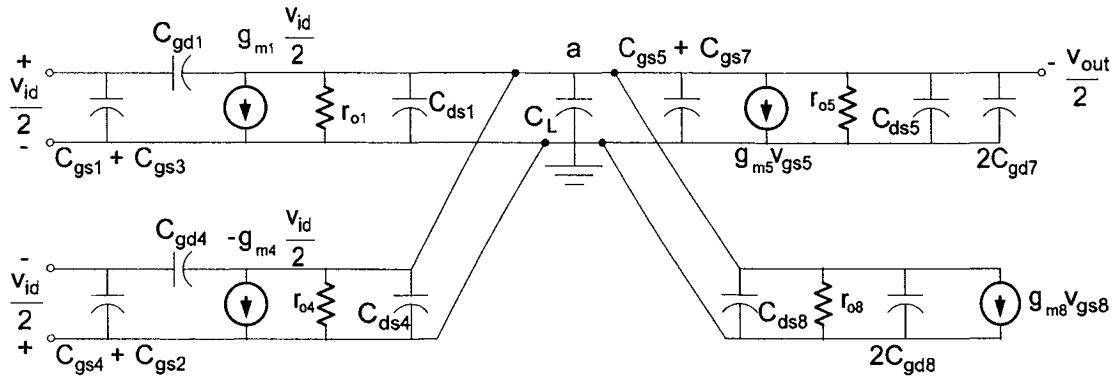


Figure 7.4: Small-signal equivalent circuit of the half-circuit of Szczepanski's OTA

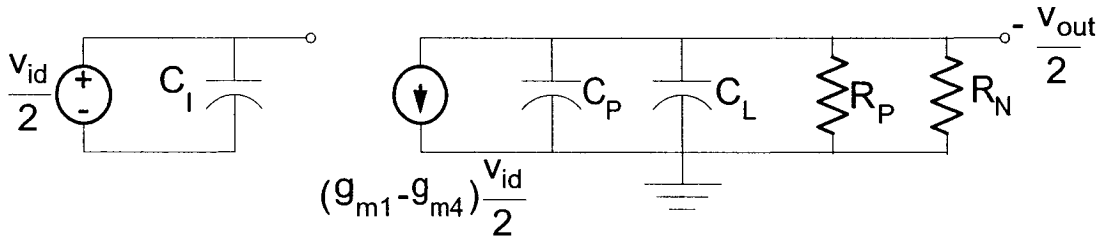


Figure 7.5: Simplified small-signal model of the half-circuit of Szczepanski's OTA

In Fig. 7.4, we have

$$R_N = \frac{1}{g_{m5} - g_{m8}} = \frac{1}{2k_p(V_{DD} - V_A)} \tag{7.11}$$

$$R_P = \frac{1}{g_{o1} + g_{o4} + g_{o5} + g_{o8}} \tag{7.12}$$

$$C_I = C_{gs1} + C_{gs3} + 2C_{gd} \tag{7.13}$$

$$C_I = C_{ds1} + C_{ds4} + C_{ds5} + C_{ds8} + C_{gs5} + C_{gs7} + 2C_{gd} + 2C_{gd7} + 2C_{gd8} \tag{7.14}$$

Note that $g_{oi} = \frac{1}{r_{oi}}$, applying the model in Fig. 5.27, the transfer function is given by,

$$A(s) = \frac{V_{out}}{V_{id}} = -\frac{G_m}{s(C_L + C_P) + \frac{1}{R_P} + \frac{1}{R_N}} \tag{7.15}$$

Also, assuming an input source resistance $R_s = 0$, then the short-circuit transconductance is derived as

$$\frac{I_{out}}{V_{id}} = G_m + S(C_{gd4} - C_{gd1}) \tag{7.16}$$

Where $G_m = g_{m1} - g_{m4} = 2k_n V_B$. So, if we make the gate-drain capacitances, C_{gd} of M_1, M_2, M_3, and M_4 equal, the OTA transconductance frequency dependent component caused by the gate-drain capacitances of MOS devices is canceled out, resulting a very wideband frequency response. This can be achieved by selecting $W_1 = W_2 = W_3 = W_4$.

As a design example, we design the OTA shown in Fig. 7.3 with considering $V_{DD} = 1.5$ V, $V_{SS} = -1.5$ V, $V_S = -0.5$ V, transconductance (g_m) between 350 to 360 $\frac{\mu A}{V}$, power dissipation less than 3 mW, DC output voltage close to zero, and a 0.18 μm process technology. To design the OTA, we divide the circuit to three parts including driver stage (gain stage), negative resistance load (NRL) and biasing stage. For designing the driver and NRL stages, we use equations 7.11, 7.12 and 7.16 in our CAD tool (CADT). The design parameters obtained from the CADT for each stage of the OTA are presented in the following tables.

Table 7.5: Transistor sizes and bias conditions for OTA driver stage

Transistor	$\frac{W}{L}(\frac{\mu m}{\mu m})$	V_{GS} (Volt)	V_{DS} (Volt)	V_{BS} (Volt)	$I_{DS}(\mu A)$	$g_m(\frac{\mu A}{V})$	$g_{ds}(\frac{\mu A}{V})$
M_1 and M_2	$\frac{2.08}{0.5}$	1.2	1.2	0.3	228.2992	548.5720	8.3351
M_3 and M_4	$\frac{2.08}{1.35}$	1.0	1.0	0.5	44.0520	189.1942	0.9750

From table 7.5, we obtain the gain for the proposed OTA as follows:

$$G_m = g_{m1} - g_{m4} = 359.3779 \ \frac{\mu A}{V} \tag{7.17}$$

Table 7.6: Transistor sizes and bias conditions for OTA NRL stage

Transistor	$\frac{W}{L}(\frac{\mu m}{\mu m})$	V_{GS} (Volt)	V_{DS} (Volt)	V_{BS} (Volt)	$I_{DS}(\mu A)$	$g_m(\frac{\mu A}{V})$	$g_{ds}(\frac{\mu A}{V})$
M_5 and M_6	$\frac{3.02}{0.5}$	1.4	1.4	0.1	121.3244	221.4872	5.5061
M_7 and M_8	$\frac{3.02}{0.5}$	1.5	1.5	0.0	153.4176	239.9158	6.5620

Table 7.7: Transistor sizes and bias conditions for OTA biasing section

Transistor	$\frac{W}{L}(\frac{\mu m}{\mu m})$	V_{GS} (Volt)	V_{DS} (Volt)	V_{BS} (Volt)	$I_{DS}(\mu A)$
M_9	$\frac{48.9}{0.6}$	-1.0	-0.1	0	242.6801
M_{a1}	$\frac{0.6}{0.5}$	-1.73	-2.23	0.77	27.7418
M_{a2}	$\frac{0.6}{0.5}$	-1.29	-1.29	0.77	10.7632
M_{a3}	$\frac{9.7}{0.5}$	0.7	0.2	-0.3	88.3622
M_{b1}	$\frac{4.5}{0.5}$	-0.84	-0.76	0	39.5699
M_{b2}	$\frac{0.7}{0.5}$	-0.84	-0.84	0	5.6008
M_{b3}	$\frac{0.5}{1.0}$	0.7	2.15	0	5.6529
M_{b4}	$\frac{28.6}{0.5}$	0.7	0.3	0	544.2668
M_{cf}	$\frac{1.1}{1.0}$	0.7	0.93	0	10.9794

The overall OTA performance parameters obtained by Cadence simulation are listed in table 7.8. These results show that our design goals are almost met.

Table 7.8: Simulation results of OTA

Parameter	Value	Unit
Power Supply	±1.5	V
Power Dissipation	1.765	mW
Transconductance	371	$\frac{\mu A}{V}$
DC Output Voltage	7.715	mV

7.3 Summary

We have shown how the new design methodology and our CAD tool (CADT) can be used to design large scale analog IC building blocks. The benefit and usefulness of the CADT for reducing the transistor level design cycle time was illustrated by thesis design examples as well.

Chapter 8

Conclusion

This chapter summarizes achievements and shortcomings of this research work and provides the recommendations for the future actions.

8.1 Conclusion and discussion

A new computer based method for the design of analog IC building blocks has been presented in this thesis. As described, it is based on both a knowledge-based and simulation-based design approaches. This design methodology was implemented in a CAD tool, called CADT, that integrates building blocks design subprograms and the databases. The usefulness and effectiveness of the new design methodology and the CADT in automating the design of some analog IC building blocks has been proved by presenting design examples. The design examples confirm that this new design methodology and the CADT can help analog designers to design many analog building blocks with acceptable accuracy faster and easier. Therefore, we achieved our goals in improving the design cycle time and facilitating the design process by avoiding using hand analysis calculations and running frustrating simulations that result in faster designs. The shortcomings and limitations of the new method and the CADT be summarized as follows:

- The method is based on calculating of g_m, g_{ds} and g_{bs} using output current-voltage characteristic of NMOS and PMOS devices that is prone to errors and it could result in obtaining inaccurate design parameters in some cases. We can minimize the errors in the CADT by using smaller step values for DC bias voltages (0.01 Volt) while calculating these parameters.
- The output results provided by the CADT depend on the input data. Therefore, if the user does not choose a suitable range for design variables (input data); the possibility of finding a design solution by the CADT degrades.
- Our CAD tool can only provide a solution in terms of gain and power dissipation for an analog IC building block.

8.2 Future work

The design method and the CAD tool that has been presented in this thesis is the first version of the program and therefore, it is not perfect and mature. The following future actions are suggested to improve the quality and to overcome the limitation and the shortcomings of the design methodology and the CADT reported in this thesis.

- Enhance the quality of the user interface of the CAD tool up to the standard of commercial tools and provide the possibility of visual displaying of circuit schematics as well.
- Expand the tool to support more building blocks by allowing the users to develop new modules for their circuits and add them to the tool.
- Improve the quality of the methodology and the tool by considering more design performance measures and constraints, for example, area as the objective to be minimized and bandwidth as the objective to be maximized.

- Continue to update the methodology as required for deep sub-micron IC technology of the future (i.e., Nano-technology).
- Introduce new algorithms which would optimize the usage of the computer memory for application of the CAD tool.

References

[1] Christofer Toumazou and Costas A. Makris, "Automated Circuit Generation: New Concepts and Methods," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 14, No. 2, February 1995

[2] M. G. Guvench, M. Miske and E. Crain, "Design, Fabrication and Testing of CMOS Operational Amplifiers as Training Tool in Analog Integrated Circuit Design," IEEE, 2001

[3] Behzad Razavi, "Design of Analog CMOS Integrated Circuits," McGraw-Hill, 2001

[4] G. Kelson, "Design automation techniques for analog VLSI," VLSI Design, PP. 78-82, January 1985

[5] P. E. Allen, E. R. Macaluso, S. F. Bily, "AIDE2: An Automated Analog IC Design System," in Proc. IEEE Custom Integ. Circ. Conf. (CICC), PP. 498-501, 1985

[6] R. J. B. Bowman and D. J. L. Lane, "A Knowledge-Based System for Analog Integrated Circuit Design ." in Proc. IEEE Int. Conf. Computer-Aided Design (ICCAD), PP. 210-212, 1985

[7] F. El-Turky and E. E. Perry, "BLADES: An Artificial Intelligence Approach to Analog Circuit Design," IEEE Trans. Computer-aided Design, Vol. 8, No. 6, PP. 680- 692, 1989

[8] R. Harjani, R. A. Rutenbar, and L. R. Carley, "A Prototype framework for Knowledge-Based Analog Circuit Synthesis," in Proc. ACM/IEEE Design Automation Conf. (DAC), PP. 42-49, 1987

[9] G. Gielen, K. Swings, and W. Sansen, "An Intelligent Design System for Analog Integrated Circuits" in Proc. IEEE Euro. Design Automation Conf. (ECAD), PP. 169-173, 1990

[10] H. Y. Koh, C. H. Sequin, and P. R. Gray, "Automatic Synthesis of Operational Amplifiers Based on Analytic Circuit Models," in Proc. IEEE Int. Conf. Computer-Aided Design (ICCAD), PP. 502-505, 1987

[11] L. Pillage and R. Rohrer, "Asymptotic Waveform Evaluation for Tuning Analysis," IEEE Trans. On CAD, Vol. 9, No. 4, PP. 352-366, April 1990

[12] I. Getreu, "Behavioral Modeling of Analog Blocks Using the Saber Simulator," Proc. MWCAS, PP. 977-980, Aug. 1989

[13] J. Singh and R. Saleh, "iMACSIM: A Program for Multi-Level Analog Circuit Simulation," Proc. IEEE ICCAD, PP. 16-19, Nov. 1991

[14] W. Nye Et Al, "DELIGTH.SPICE: An Optimization-Based System for the Design of Integrated Circuits," IEEE Transactions on Computer_Aided Design, 7:501-518, April 1988

[15] J. M. Shyu and A. Sangiovanni-Vincentelli, "ECSTASY: A New Environment for IC Design Optimization," in Proc. IEEE Int. Conf. Computer-Aided Design (ICCAD), PP. 484-487, 1988

[16] J. C. Lai, J. S. Kueng, H. J. Chen, and F. J. Fermandez, "ADOPT: A CAD System for Analog Circuit Design ," in Proc. IEEE Custom Integ. Circ. Conf. (CICC), PP. 3.2.1-3.2.4, 1988

[17] E. S. Ochotta, R.A. Rutenbar, and L. R. Carley, "Synthesis of High-Performance Analog Circuits in ASTRX/OBLX," IEEE Trans. On Computer-Aided Design, PP. 273-293, March 1996

[18] G. G. E. Gielen, H. C. C. Walscharts, and W. M. C. Sansen, "Analog Circuit Design Optimization Based on Symbolic Simulation and simulated Annealing," IEEE Journal of Solid-State Circuits, PP. 707-713, June 1990

[19] Y. Nesterov and A. Nemirovsky, "Interior-Point Polynomial Methods in Convex Programming," Vol. 13 of Studies in Applied Mathematics, SIAM, Philadelphia, PA, 1994

[20] Maria Del Mar Hershenson, Stephen P. Boyd, Thomas H. Lee, "GPCAD: A Tool for CMOS OP-Amp Synthesis," ICCAD, 1998

[21] P. E. Allen, "A tutorial-Computer-Aided Design of Analog Integrated Circuits," in Proc. IEEE Custom Integr. Circ. Conf. (CICC), PP. 608-616, 1986

[22] D. C. Sussman and R. M. Stallman, "Heuristic techniques in Computer-Aided Circuit Analysis," IEEE Trans. Circuits Syst., Vol. CAS-22, No. 11, PP. 857-865, Nov. 1975

[23] J. De Kleer, "How Circuits Work," in Qualitative Reasoning About Physical Systems, D. G. Bobrow, Ed. Cambridge, MA, MIT Press, PP. 205-280, 1985

[24] M. J. S. Smith Et Al, "Cell Libraries and Assembly Tools for Analog/Digital CMOS and BiCMOS Application-Specific Integrated Circuit Design," IEEE Journal Solid-State Circ., Vol. 24, No. 5, PP. 1419-1432, 1989

[25] J. Kuhn, "Analog Module Generators for Silicon Compilation," VLSI Syst., PP. 74-80, May 1987

[26] R. Harjani, R. A. Rutenbar, and L. R. Carley, "OASYS: A Framework for Analog Circuit Synthesis," IEEE Trans. Computer-Aided Design, Vol. 8, No. 12, PP. 1247-1265, 1989

[27] M. Degrauwe Et Al, "IDAC: An Interactive Design Tool for Analog CMOS Circuits," IEEE Journal Solid-State Circuits, Vol. 22, No. 6, PP. 1106-1115, 1987

[28] H. Y. Koh, C. H. Sequin and P. R. Gray, "OPASYN: A Compiler for CMOS Operational Amplifiers," IEEE Trans. Computer-Aided Design, Vol. 9, No. 2, PP. 113-125, 1990

[29] H. Onodera, H. Kanbara, and K. Tamaru, "Operational Amplifier Compilation with Performance Optimization," IEEE Journal Solid-State Circuits, Vol. 25, PP. 466-473, Apr. 1990

[30] G. Gielen and H. C. C. Walscharts, and W. M. C. Sansen, "Analog Circuit Design Optimization Based on Symbolic Simulation and Simulated Annealing," IEEE Journal Solid-State Circ., Vol. 25, No. 3, PP. 707-713, June 1990

[31] B. J. Sheu, J. C. Lee, and A. H. Fung, "Flexibles Architecture approach to Knowledge-Based Analog IC Design," IEE Porc. Part G., Vol. 137, No. 4, PP. 266-274, Aug. 1990

[32] Danica Stefanovic, Maher Kayal, Marc Pastre, and Vanco B. Litovski, "PAD: A Procedural Analog Design Tool," IEEE Proc. Fourth Int. Symposium on Quality Electronic Design (ISQED), 2003

[33] R.S. Muller and T.I. Kamins, "Device Electronics for Integrated Circuits," Second Edition, Wiley, 1986

[34] D.P. Foty, "MOSFET Modeling with SPICE," Prentice- Hall, 1997

[35] Meta-Software, HSPICE User's Manual, 1992

[34] R. Raut, "Introduction to Analog VLSI," Lecture Notes, Concordia University Publication, 2002

[35] Stanislaw Szczepanski, Jacek Jakusz, and Rolf Schaumann, "A Linear Fully Balanced CMOS OTA for VHF Filtering Applications," IEEE Trans. on Circuits and Systems, Vol. 44, No. 3, March 1997

Appendixes

Appendix – A

The Menus of First Version of the CADT Based on MATLAB Programming

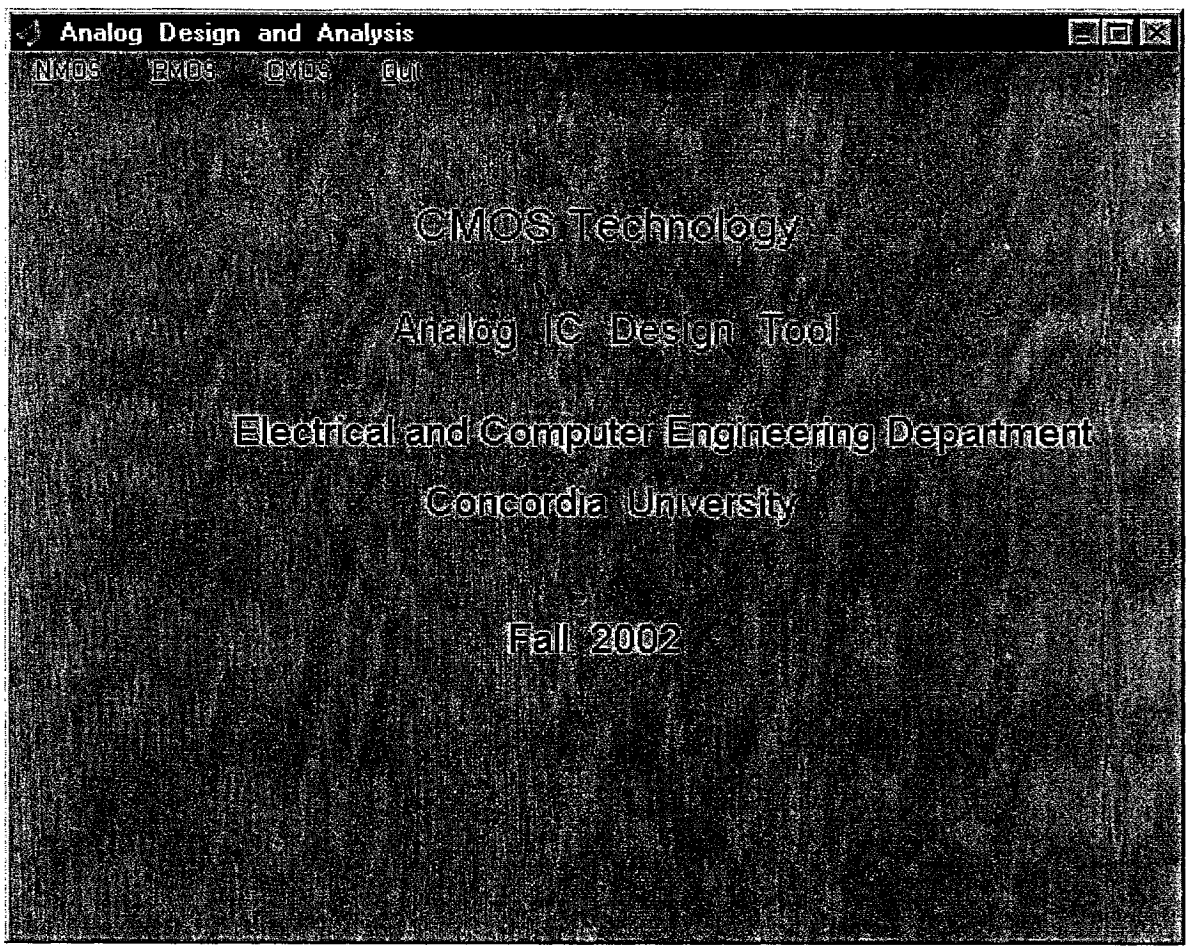


Figure A.1: Main Menu

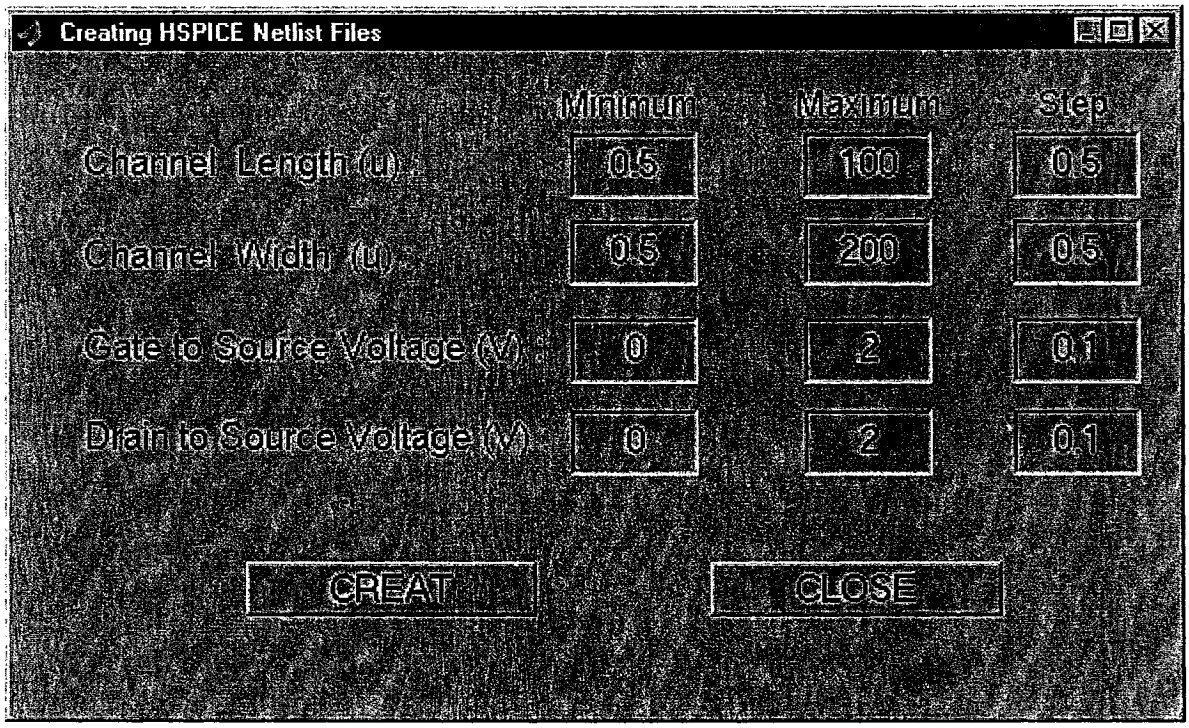


Figure A.2: Setup Menu of the CAD tool

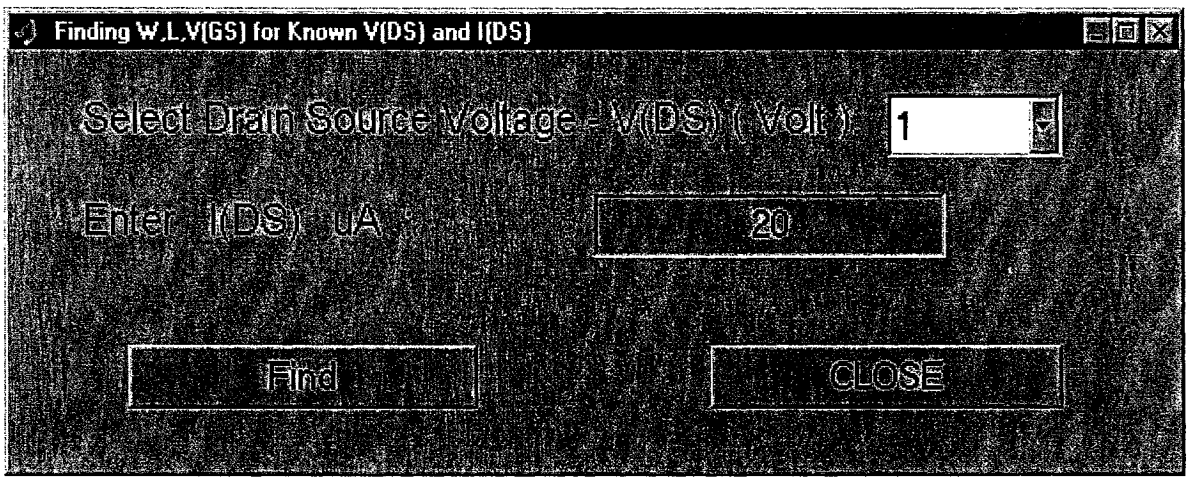


Figure A.3: Search Window Based on I_{DS} and V_{DS}

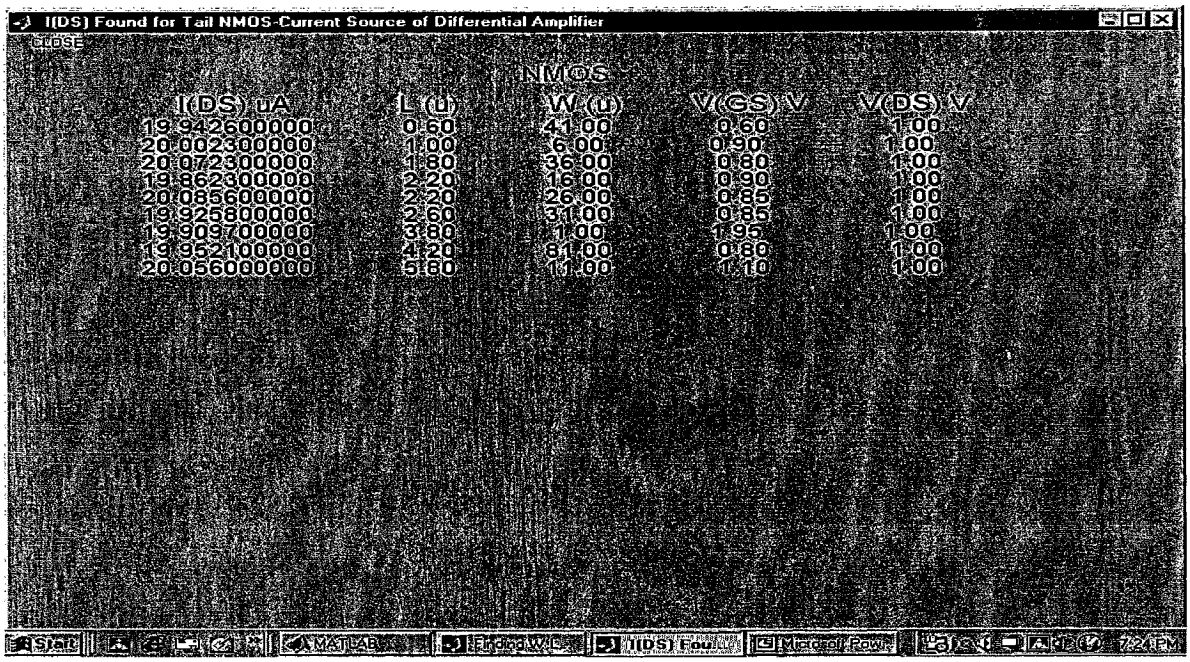


Figure A.4: Window of Output Results

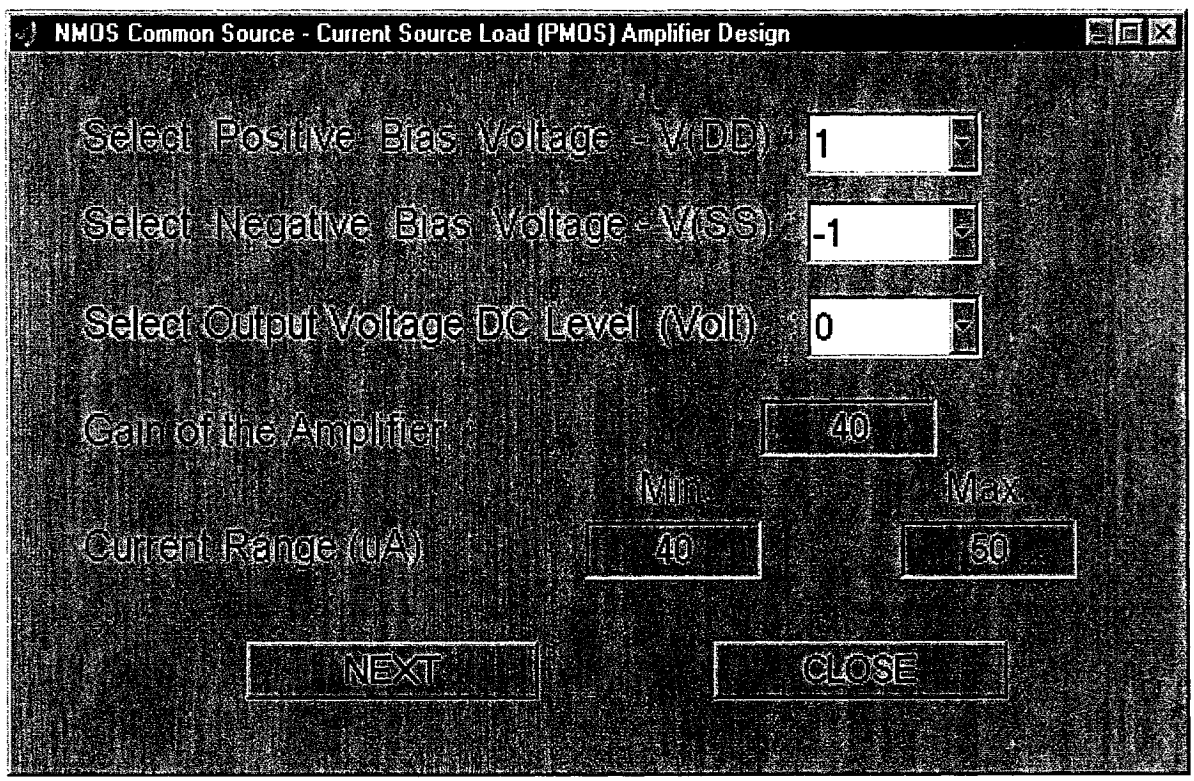


Figure A.5: Design Menu of Common-Source Amplifier

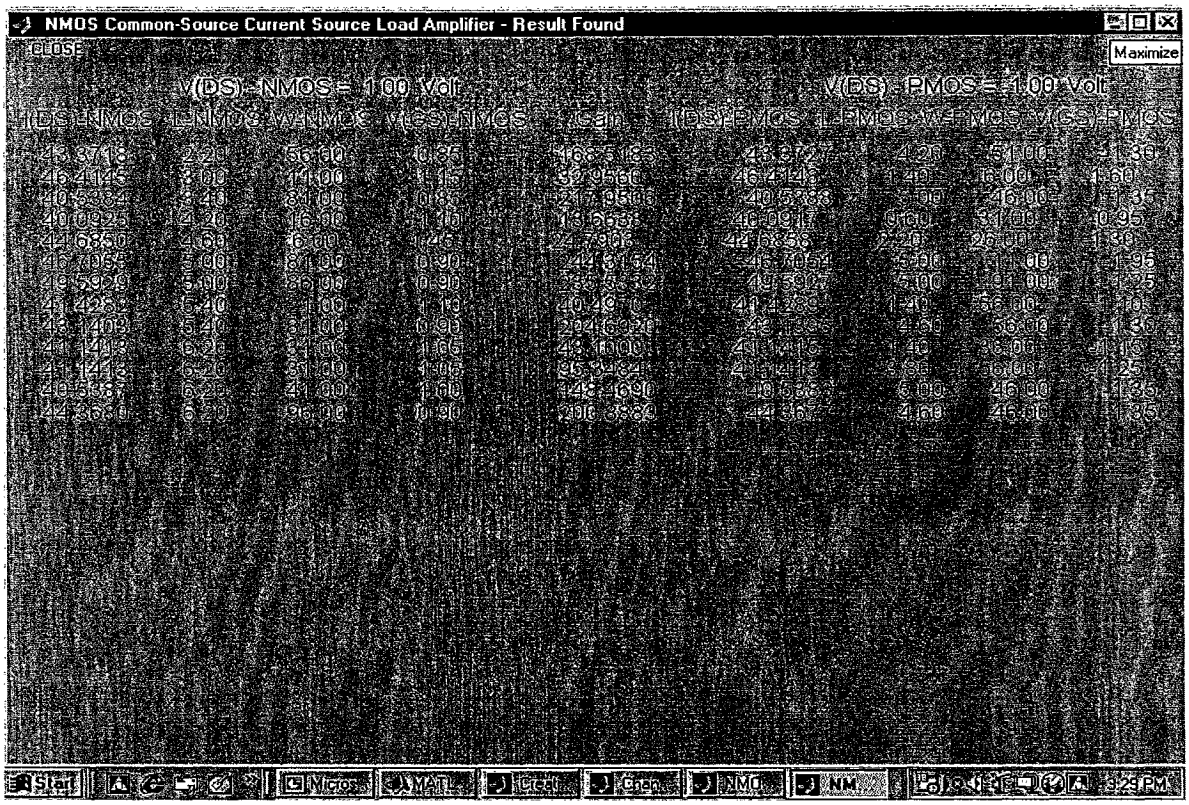


Figure A.6: Output Results of Common Source Amplifier Design

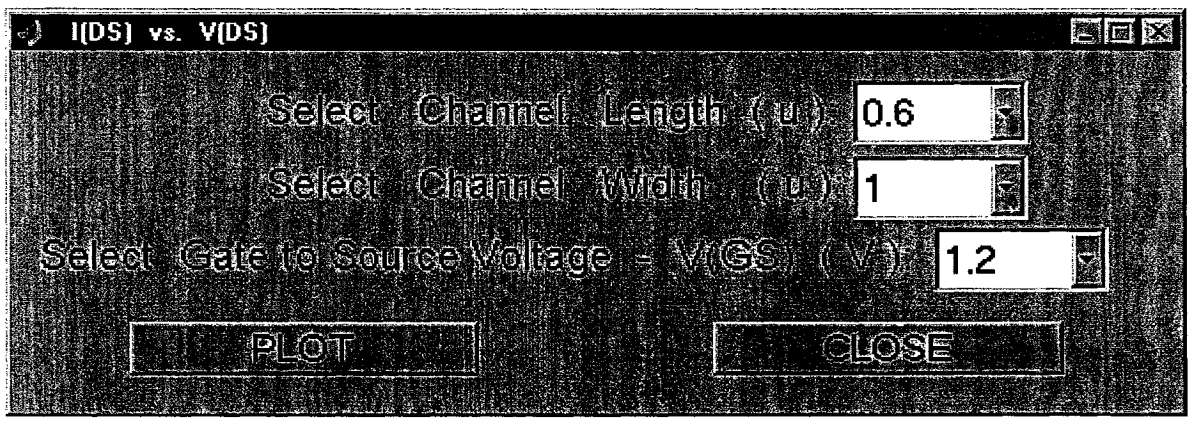


Figure A.7: Setup Menu for Plotting Output Characteristics of MOS Device

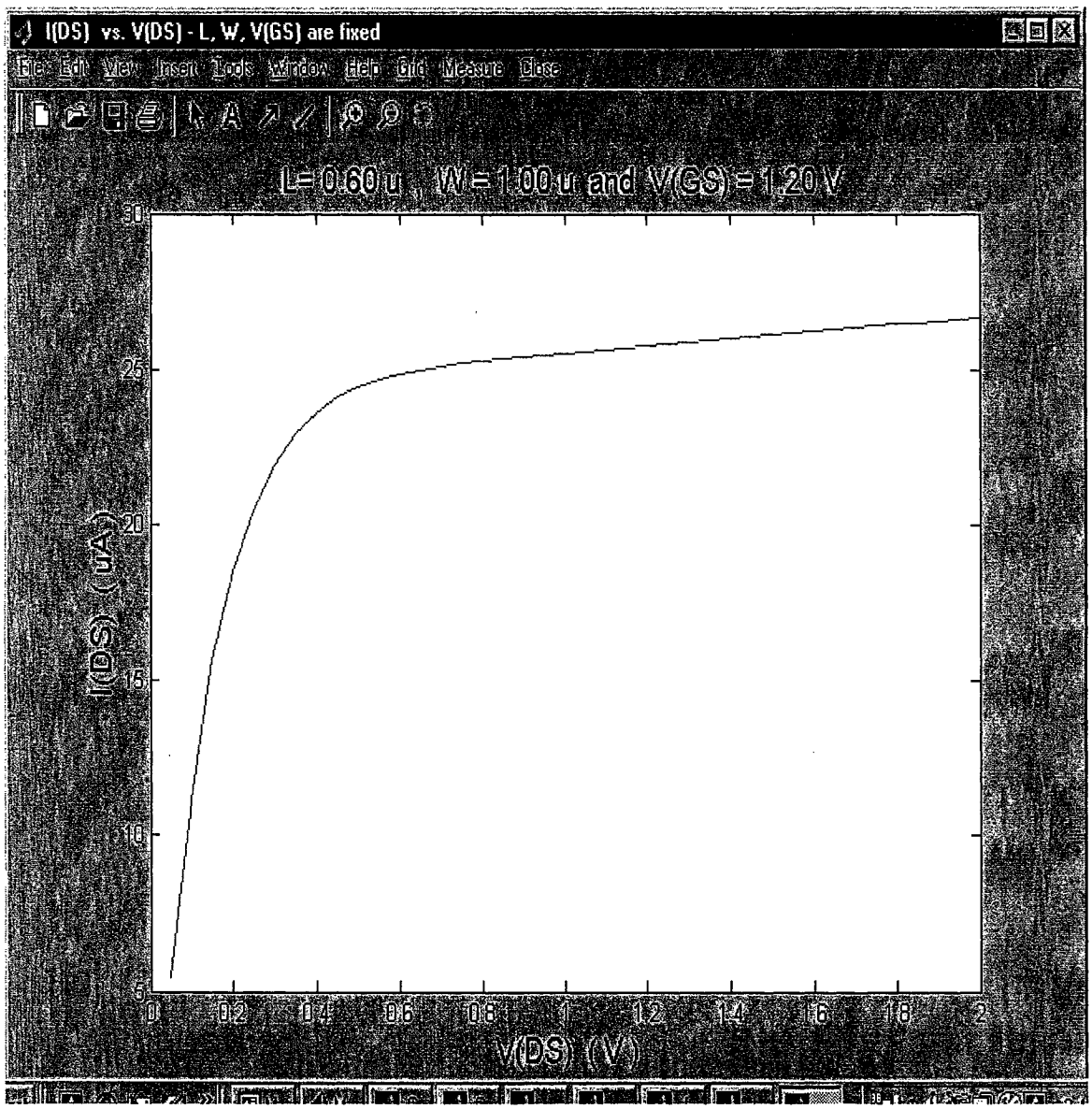


Figure A.8: Plot of Output Characteristic of MOS Device

Appendix – B

Sample Source Code of C Program for Common-Source Amplifier Design

```
/*****************************************************************/
/*                                                               */
/*                                                               */
/*    Filename       :     csa.c                                 */
/*    File Type      :     C Programing Language for UNIX OS     */
/*    Date           :     Nov. 25, 2003                         */
/*    Description    :     Common Source Amplifier Design        */
/*    Dept.          :     ELEC. & COMP.  ENG.                   */
/*    University     :     ECE Concordia University              */
/*                                                               */
/*                                                               */
/*****************************************************************/

#include "stdio.h"
#include "stdlib.h"
#include "string.h"

void main(void)

{

char  ch1,myline[100]=" ",str1[10]=" ",str2[10]=" ",ch,fnn[10],fnp[10];
int nrvbs=10000,prvbs=10000,nrvds=10000,prvds=10000,q,d,tmp=0,
    rowcount=0,rowcountp=0,i,icount,j,k,m,l,cntr1,cntr2,cntr3,
    cntr4,cntr5;
int   *index_l,*index_w,*index_vgs,*index_idsnmos,*index_lp,*index_wp,
      *index_vgsp;
float vbsn,vbsp,num,den,res1,res2,*csgain,*gm,*gds,*gdsp,
      *myidsnmos,*myidspmos;
float *np_l,*np_w,*np_vgs,*np_vbs,*np_vds,str3,k111,k112;
float tempcurr=0,vdd,vss,vout,gain=0,idsmin=0,idsmax=0,vds1,vds2,
      *idsn,*idsp;
float min,minl,maxl,stepl,minvgs,maxvgs,stepvgs,minvds,maxvds,
      stepvds,minw,maxw,stepw,minvbs,maxvbs,stepvbs;
FILE  *data,*dcnmos,*dcpmos,*npdata;
index_l       = (int *) malloc(1000000*sizeof(int));
index_w       = (int *) malloc(1000000*sizeof(int));
index_vgs     = (int *) malloc(1000000*sizeof(int));
index_idsnmos = (int *) malloc(1000000*sizeof(int));
index_lp      = (int *) malloc(1000000*sizeof(int));
index_wp      = (int *) malloc(1000000*sizeof(int));
index_vgsp    = (int *) malloc(1000000*sizeof(int));
csgain        = (float *) malloc(1000000*sizeof(float));
gm            = (float *) malloc(1000000*sizeof(float));
gds           = (float *) malloc(1000000*sizeof(float));
gdsp          = (float *) malloc(1000000*sizeof(float));
```

```
myidsnmos      = (float *) malloc(1000000*sizeof(float));
myidspmos      = (float *) malloc(1000000*sizeof(float));
np_l           = (float *) malloc(10000*sizeof(float));
np_w           = (float *) malloc(10000*sizeof(float));
np_vgs         = (float *) malloc(10000*sizeof(float));
np_vbs         = (float *) malloc(10000*sizeof(float));
np_vds         = (float *) malloc(10000*sizeof(float));
idsn            = (float *) malloc(10000000*sizeof(float));
idsp            = (float *) malloc(10000000*sizeof(float));
system("clear");
printf("\nPress Ctrl+C to Return to Main Menu");
printf("\n\nEnter NMOS DC Analysis HSPICE Filename: ");
gets(fnn);
if((dcnmos=fopen(fnn,"r")) == NULL ){
printf(" Can not Open File\n");
                  exit(1);
                                        }
printf("\n\n");
printf("Enter PMOS DC Analysis HSPICE Filename: ");
gets(fnp);
if((dcpmos=fopen(fnp,"r")) == NULL ){
      printf(" Can not Open File\n");
      exit(1);
      }
if((npdata=fopen("setup.txt","r")) == NULL ){
      printf(" Can not Open File\n");
      exit(1);
      }
fgets(myline,100,npdata);
fgets(myline,100,npdata);
sscanf(myline," %f %f %f", &minl, &maxl, &stepl);
fgets(myline,100,npdata);
sscanf(myline," %f %f %f", &minw, &maxw, &stepw);
fgets(myline,100,npdata);
sscanf(myline," %f %f %f", &minvds, &maxvds, &stepvds);
fgets(myline,100,npdata);
sscanf(myline," %f %f %f", &minvgs, &maxvgs, &stepvgs);
fgets(myline,100,npdata);
sscanf(myline," %f %f %f", &minvbs, &maxvbs, &stepvbs);
fgets(myline,100,npdata);
sscanf(myline," %i %i %i %i %i ", &cntr1, &cntr2 , &cntr3 , &cntr4 ,
&cntr5);
min=minl;
*np_l=min;
for(k=1; k < cntr1;k++){ min = min + stepl;
                  *(np_l+k)=min;
                  }
min=minw;
*np_w=min;
for(l=1; l < cntr2;l++){ min = min + stepw;
                  *(np_w+l)=min;
                  }
min=minvgs;
*np_vgs=min;
for( j=1 ; j < cntr3 ; j++ ){ min = min + stepvgs;
                        *(np_vgs+j)=min;
                     }
```

```
min=minvbs;
*np_vbs=min;
for(i=1; i < cntr4;i++){ min = min + stepvbs;
                    *(np_vbs+i)=min;
                    }
min=minvds;
*np_vds=min;
for(m=1; m < cntr5;m++){ min = min + stepvds;
                    *(np_vds+m)=min;
                    }
fclose(npdata);
top:;
printf("\n\nEnter Positive DC Bias Voltage [ V(DD) ] (V): ");
scanf("%f", &vdd);
printf("\n\nEnter Negative DC Bias Voltage [ V(SS) ] (V): ");
scanf("%f",&vss);
printf("\n\nEnter Output Voltage DC Level [ V(OUT) ] (V): ");
scanf("%f",&vout);
printf("\n\nEnter NMOS Body-Source DC Bias Voltage [ V(BS) ] (V): ");
scanf("%f",&vbsn);
printf("\n\nEnter PMOS Body-Source DC Bias Voltage [ V(BS) ] (V): ");
scanf("%f",&vbsp);
printf("\n\nEnter the Voltage Gain of Common Source Amplifier [A(v)
with 20 Percent Tolerance]: ");
scanf("%f",&gain);
printf("\n\nEnter Drain-Source Current Range [ I(DS): Min.  Max. ]
(uA): ");
scanf("%f%f",&idsmin,&idsmax);
if(gain == 0 || vdd== 0 || idsmax==0) goto top;
vds1 = vout - vss;
vds2 = vout - vdd;
for( i=0 ; i < cntr5 ; i++){
                      if(  fabs( *(np_vds+i) - vds1 ) < 0.0001 ) {
nrvds=i;
                                                          }
                    }
for( i=0 ; i < cntr5 ; i++){
                      if(  fabs( *(np_vds+i) + vds2 ) < 0.0001 ){
prvds=i;
                                                          }
                    }
for( i=0 ; i < cntr4 ; i++){
                      if(  fabs( *(np_vbs+i) - vbsn ) < 0.0001 ){

nrvbs=i;
                                                          }
                    }
for( i=0 ; i < cntr4 ; i++){
                      if(  fabs( *(np_vbs+i) - vbsp ) < 0.0001 ){
prvbs=i;
                                                          }
                    }
if( nrvds == 0 || prvds == 0 || nrvbs == 10000 || prvbs == 10000 ||
nrvds == 10000 || prvds == 10000 ){
printf("\n\nThere is no data ...!!!\n\n");
        ch1 = getchar();
        printf("Press any key to continue...");
```

```
                                        ch1 = getchar();
                                        printf("\n\n\n");
                                        fclose(dcnmos);
                                        fclose(dcpmos);
                                        exit(1);

                                             }

fgets(myline,100,dcnmos);

while( !feof(dcnmos) ){
if( !strncmp( myline,"x",1 ) ){
                              fgets(myline,100,dcnmos);
                              while( strncmp( myline,"y",1) ){
                                      fgets(myline,100,dcnmos);
                                      if( !strncmp(myline,"y",1))break;
                                      if( !(strstr(myline,"volt") ||
strstr(myline,"vds")) ){
                sscanf(myline,"%s %s",str1,str2);
                if ( strstr(str2,"m") ){
sscanf(str2,"%f%c",&str3,&ch);
*( idsn + rowcount ) = str3*1000;
      }
         else if ( strstr(str2,"u") ){
sscanf(str2,"%f%c",&str3,&ch);
*( idsn + rowcount ) = str3;
            }
         else if ( strstr(str2,"n") ){
sscanf(str2,"%f%c",&str3,&ch);
*( idsn + rowcount ) = str3/1000;
                  }
         else if ( strstr(str2,"p") ){
sscanf(str2,"%f%c",&str3,&ch);
*( idsn + rowcount ) = str3/1000000;
                  }
         else if ( strstr(str2,"f") ){
sscanf(str2,"%f%c",&str3,&ch);
*( idsn + rowcount ) = str3/1000000000;
                  }
         else {
         /*  *ids = str/1000;*/
         /*  fprintf(idsnmos,"%s\n", str2); */
                                          }
*( idsn + rowcount ) = - ( *( idsn + rowcount ) );
                     rowcount = rowcount + 1;
                                          }
                                          }
                                       }
fgets(myline,100,dcnmos);
}
fclose(dcnmos);
icount=0;
                  for(k=0;k<cntr1;k++){
                        for(l=0;l<cntr2;l++){
                              for(j=1;j<cntr3-1;j++){
```

```
tempcurr =
*(idsn+((nrvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*cntr5)+(j*cntr2*
cntr5)+(l*cntr5)+ nrvds));

if( tempcurr <= idsmax && tempcurr >= idsmin && nrvds < cntr5-1 ){

*(gm+icount)=(((*(idsn+((nrvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*
cntr5)+((j+1)*cntr2*cntr5)+(l*cntr5)+nrvds))-
*(idsn+((nrvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*cntr5)+(j*cntr2*
cntr5)+(l*cntr5)+nrvds)))/(*(np_vgs+j+1)-
*(np_vgs+j)))+((*(idsn+((nrvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*
cntr5)+(j*cntr2*cntr5)+(l*cntr5)+nrvds))-
*(idsn+((nrvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*cntr5)+((j-
1)*cntr2*cntr5)+(l*cntr5)+ nrvds)))/(*(np_vgs+j)-*(np_vgs+j-1))))/2;

*(gds+icount)=(((*(idsn+((nrvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2
*cntr5)+(j*cntr2*cntr5)+(l*cntr5)+nrvds+1))-
*(idsn+((nrvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*cntr5)+(j*cntr2*
cntr5)+(l*cntr5)+nrvds)))/(*(np_vds+nrvds+1)-
*(np_vds+nrvds)))+((*(idsn+((nrvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cn
tr2*cntr5)+(j*cntr2*cntr5)+(l*cntr5)+nrvds))-
*(idsn+((nrvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*cntr5)+(j*cntr2*
cntr5)+(l*cntr5)+ nrvds-1)))/(*(np_vds+nrvds)-*(np_vds+nrvds-1))))/2;
/* k111= *(gm+icount); */
/* k112= *(gds+icount); */
      *(index_l+icount)    = k;
      *(index_w+icount)    = l;
      *(index_vgs+icount) = j;
*(myidsnmos+icount) =
*(idsn+((nrvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*cntr5)+(j*cntr2*
cntr5)+(l*cntr5)+ nrvds));
                  icount  = icount+1;
                                    }
                                                                  }
                                                                  }
                                                                  }
fgets(myline,100,dcpmos);
while( !feof(dcpmos) ){
if( !strncmp( myline,"x",1 ) ){
                              fgets(myline,100,dcpmos);
                              while( strncmp( myline,"y",1) ){
                               fgets(myline,100,dcpmos);
                               if( !strncmp(myline,"y",1))break;
                               if( !(strstr(myline,"volt") ||
strstr(myline,"vds")) ){
sscanf(myline,"%s %s",str1,str2);
if ( strstr(str2,"m") ){
sscanf(str2,"%f%c",&str3,&ch);
*( idsp+rowcountp ) = str3*1000;
            }
      else if ( strstr(str2,"u") ){
sscanf(str2,"%f%c",&str3,&ch);
*( idsp+rowcountp ) = str3;          }
      else if ( strstr(str2,"n") ){
sscanf(str2,"%f%c",&str3,&ch);
*( idsp+rowcountp ) = str3/1000;}
      else if ( strstr(str2,"p") ){
```

```
sscanf(str2,"%f%c",&str3,&ch);
          *( idsp+rowcountp ) = str3/1000000;
                    }
      else if ( strstr(str2,"f") ){
sscanf(str2,"%f%c",&str3,&ch);
*( idsp+rowcountp ) = str3/1000000000;
                    }
      else {
         /* ids[rowcountp] = str/1000;*/
        /*  fprintf(idspmos,"%s\n", str2); */
                                                    }
                                                rowcountp = rowcountp + 1;
                                                   }
                                              }
                                         }
fgets(myline,100,dcpmos);
}
fclose(dcpmos);
q=0;
for( d=0 ; d < icount ; d++ ){
res2 = *( myidsnmos + d );
                    for(k=0;k<cntr1;k++){
                          for(l=0;l<cntr2;l++){
                                for(j=1;j<cntr3-1;j++){
res1 =
*(idsp+((prvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*cntr5)+(j*cntr2*
cntr5)+(l*cntr5)+ prvds));
if(  fabs(res1-res2) < 0.001 && prvds < cntr5-1 ){
*(gdsp+q)=(((*(idsp+((prvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*cnt
r5)+(j*cntr2*cntr5)+(l*cntr5)+prvds))-
*(idsp+((prvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*cntr5)+(j*cntr2*
cntr5)+(l*cntr5)+prvds-1))) /(*(np_vds+prvds) -*(np_vds+prvds-
1)))+((*(idsp+((prvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*cntr5)+(j
*cntr2*cntr5)+(l*cntr5)+ prvds+1))-
*(idsp+((prvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*cntr5)+(j*cntr2*
cntr5)+(l*cntr5)+ prvds)))/(*(np_vds+prvds+1)- *(np_vds+prvds) )))/2;

      *( myidspmos + q ) =
*(idsp+((prvbs*cntr1*cntr3*cntr2*cntr5)+(k*cntr3*cntr2*cntr5)+(j*cntr2*
cntr5)+(l*cntr5)+ prvds));
      *( index_idsnmos+q ) = d;
      *( index_lp+q ) = k;
      *( index_wp+q ) = l;
      *( index_vgsp+q ) = j;
      num = *(gm+d);
      den = *(gds+d) + *(gdsp+q);
      *(csgain+q) = ( num / den );
      q = q + 1;
                          }
                                                                  }
                                                                  }
                                                                  }

                          }
data = fopen("data.tmp","w");
fprintf(data,"  Common Source Amplifier with NMOS driver stage and PMOS
current source stage                                          \n\n");
```

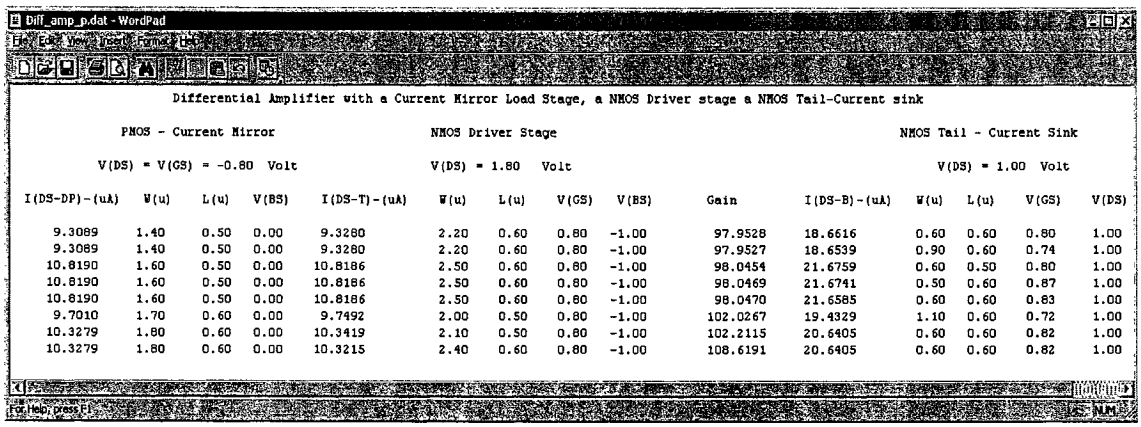
```
fprintf(data," NMOS - Driver PMOS - Current Source Load\n\n");
fprintf(data," V(DS)= %.2f  Volt V(DS)= %.2f  Volt \n\n", vds1, vds2);
fprintf(data,"I(DS)-(uA)\tW(u)\tL(u)\tV(GS)\tV(BS)\t\tGain\t\t\tI(DS)-
(uA)\t\tW(u)\tL(u)\tV(GS)\tV(BS)\n\n");
for( tmp = 0 ; tmp < q ; tmp++ ){
/*k111= *(myidsnmos+*(index_idsnmos+tmp));*/
/*k112= *(gdsp+tmp);*/
/*printf("\n%f\t%f\t%i\t",k111,k112,tmp);*/
if( *(csgain+tmp) >= gain*0.8  &&  *(csgain+tmp) <= gain*1.2
)fprintf(data,"%10.4f
\t%.2f\t%.2f\t%.2f\t%.2f\t\t%10.4f\t\t%10.4f\t\t%.2f\t%.2f\t%.2f\t%.2f\
n",*(myidsnmos+*(index_idsnmos+tmp)),*(np_w+*(index_w+*(index_idsnmos+t
mp))),*(np_l+*(index_l+*(index_idsnmos+tmp))),*(np_vgs+*(index_vgs+*(in
dex_idsnmos+tmp))),-
*(np_vbs+nrvbs),*(csgain+tmp),*(myidspmos+tmp),*(np_w+*(index_wp+tmp)),
*(np_l+*(index_lp+tmp)),-*(np_vgs+*(index_vgsp+tmp)),*(np_vbs+prvbs));
/*fprintf(data,"%10.4f
\t%.2f\t%.2f\t%.2f\t%.2f\t\t%10.4f\t\t%10.4f\t\t%.2f\t%.2f\t%.2f\t%.2f\
n",*(myidsnmos+*(index_idsnmos+tmp)),*(np_w+*(index_w+*(index_idsnmos+t
mp))),*(np_l+*(index_l+*(index_idsnmos+tmp))),*(np_vgs+*(index_vgs+*(in
dex_idsnmos+tmp))),-
*(np_vbs+nrvbs),*(csgain+tmp),*(myidspmos+tmp),*(np_w+*(index_wp+tmp)),
*(np_l+*(index_lp+tmp)),-
*(np_vgs+*(index_vgsp+tmp)),*(np_vbs+prvbs));*/

                        }
fclose(data);
system("nedit data.tmp  &");

}
```

Appendix – C

Sample Output Screens of The CADT



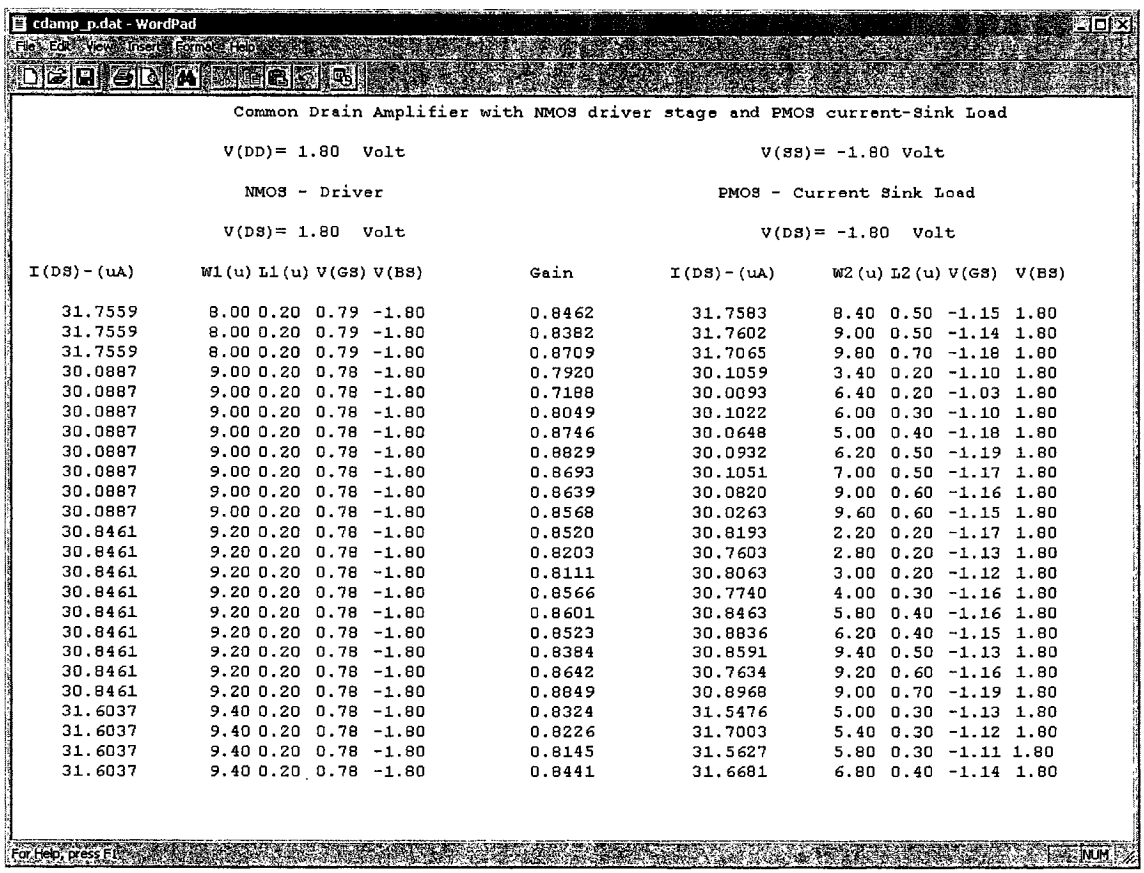
Diff_amp_p.dat - WordPad

Differential Amplifier with a Current Mirror Load Stage, a NMOS Driver stage a NMOS Tail-Current sink

PMOS - Current Mirror V(DS) = V(GS) = -0.80 Volt				NMOS Driver Stage V(DS) = 1.80 Volt						NMOS Tail - Current Sink V(DS) = 1.00 Volt				
I(DS-DP)-(uA)	W(u)	L(u)	V(BS)	I(DS-T)-(uA)	W(u)	L(u)	V(GS)	V(BS)	Gain	I(DS-B)-(uA)	W(u)	L(u)	V(GS)	V(DS)
9.3089	1.40	0.50	0.00	9.3280	2.20	0.60	0.80	-1.00	97.9528	18.6616	0.60	0.60	0.80	1.00
9.3089	1.40	0.50	0.00	9.3280	2.20	0.60	0.80	-1.00	97.9527	18.6539	0.90	0.60	0.74	1.00
10.8190	1.60	0.50	0.00	10.8186	2.50	0.60	0.80	-1.00	98.0454	21.6759	0.60	0.50	0.80	1.00
10.8190	1.60	0.50	0.00	10.8186	2.50	0.60	0.80	-1.00	98.0469	21.6741	0.50	0.60	0.87	1.00
10.8190	1.60	0.50	0.00	10.8186	2.50	0.60	0.80	-1.00	98.0470	21.6585	0.60	0.60	0.83	1.00
9.7010	1.70	0.60	0.00	9.7492	2.00	0.50	0.80	-1.00	102.0267	19.4329	1.10	0.60	0.72	1.00
10.3279	1.80	0.60	0.00	10.3419	2.10	0.50	0.80	-1.00	102.2115	20.6405	0.60	0.60	0.82	1.00
10.3279	1.80	0.60	0.00	10.3215	2.40	0.60	0.80	-1.00	108.6191	20.6405	0.60	0.60	0.82	1.00

For Help, press F1 NUM

Figure C.1: Differential Amplifier Design Data Obtained by the CADT



cdamp_p.dat - WordPad

Common Drain Amplifier with NMOS driver stage and PMOS current-Sink Load

V(DD)= 1.80 Volt V(SS)= -1.80 Volt

NMOS - Driver V(DS)= 1.80 Volt					PMOS - Current Sink Load V(DS)= -1.80 Volt				
I(DS)-(uA)	W1(u)	L1(u)	V(GS)	V(BS)	Gain	I(DS)-(uA)	W2(u)	L2(u)	V(GS)	V(BS)
31.7559	8.00	0.20	0.79	-1.80	0.8462	31.7583	8.40	0.50	-1.15	1.80
31.7559	8.00	0.20	0.79	-1.80	0.8382	31.7602	9.00	0.50	-1.14	1.80
31.7559	8.00	0.20	0.79	-1.80	0.8709	31.7065	9.80	0.70	-1.18	1.80
30.0887	9.00	0.20	0.78	-1.80	0.7920	30.1059	3.40	0.20	-1.10	1.80
30.0887	9.00	0.20	0.78	-1.80	0.7188	30.0093	6.40	0.20	-1.03	1.80
30.0887	9.00	0.20	0.78	-1.80	0.8049	30.1022	6.00	0.30	-1.10	1.80
30.0887	9.00	0.20	0.78	-1.80	0.8746	30.0648	5.00	0.40	-1.18	1.80
30.0887	9.00	0.20	0.78	-1.80	0.8829	30.0932	6.20	0.50	-1.19	1.80
30.0887	9.00	0.20	0.78	-1.80	0.8693	30.1051	7.00	0.50	-1.17	1.80
30.0887	9.00	0.20	0.78	-1.80	0.8639	30.0820	9.00	0.60	-1.16	1.80
30.0887	9.00	0.20	0.78	-1.80	0.8568	30.0263	9.60	0.60	-1.15	1.80
30.8461	9.20	0.20	0.78	-1.80	0.8520	30.8193	2.20	0.20	-1.17	1.80
30.8461	9.20	0.20	0.78	-1.80	0.8203	30.7603	2.80	0.20	-1.13	1.80
30.8461	9.20	0.20	0.78	-1.80	0.8111	30.8063	3.00	0.20	-1.12	1.80
30.8461	9.20	0.20	0.78	-1.80	0.8566	30.7740	4.00	0.30	-1.16	1.80
30.8461	9.20	0.20	0.78	-1.80	0.8601	30.8463	5.80	0.40	-1.16	1.80
30.8461	9.20	0.20	0.78	-1.80	0.8523	30.8836	6.20	0.40	-1.15	1.80
30.8461	9.20	0.20	0.78	-1.80	0.8384	30.8591	9.40	0.50	-1.13	1.80
30.8461	9.20	0.20	0.78	-1.80	0.8642	30.7634	9.20	0.60	-1.16	1.80
30.8461	9.20	0.20	0.78	-1.80	0.8849	30.8968	9.00	0.70	-1.19	1.80
31.6037	9.40	0.20	0.78	-1.80	0.8324	31.5476	5.00	0.30	-1.13	1.80
31.6037	9.40	0.20	0.78	-1.80	0.8226	31.7003	5.40	0.30	-1.12	1.80
31.6037	9.40	0.20	0.78	-1.80	0.8145	31.5627	5.80	0.30	-1.11	1.80
31.6037	9.40	0.20	0.78	-1.80	0.8441	31.6681	6.80	0.40	-1.14	1.80

For Help, press F1 NUM

Figure C.2: Common-Drain Amplifier Design Data Obtained by the CADT