2017

Low-cost, high-precision DAC design based on ordered element matching and verification against undesired operating points for analog circuits

You Li
Iowa State University

Low-cost, high-precision DAC design based on ordered element matching and verification against undesired operating points for analog circuits

by

You Li

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Electrical Engineering

Program of Study Committee:

Degang Chen, Major Professor

Nathan M. Neihart

Chris Chong-Nuen Chu

Zhengdao Wang

Jaeyoun (Jay) Kim

Iowa State University

Ames, Iowa

2017

DEDICATION

To my family

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

"Folk in those stories had lots of chances of turning back, only they didn't. They kept going, because they were holding on to something."

-J.R.R. Tolkien, *The Two Towers*.

It was a really hard journey for Frodo and Sam to finish their task of destroying the Ring at Mt Doom; similarly, it was the most difficult trip of my life to pursue my PhD degree at Iowa State University. As Frodo and Sam could not arrive at Mt Doom without the assistance of others, I would also like to attribute the completion of my PhD degree and this dissertation to those who helped me with various aspects of research and life in the past five years.

Foremost, I want to express my sincere gratitude to my advisor, Dr. Degang Chen, who offered me the opportunity to pursue my PhD degree, and guided me throughout my program. His rigorous scholarship and serious attitude to science deeply affected and always motivated me. His extensive knowledge and great insights often inspired me to see new ideas in research. I received many benefits from following his systematic and efficient work style. His guidance helped me in all my research and the writing of this thesis. I could not imagine finishing my PhD work without his mentoring and support.

I would also like to thank the rest of my thesis committee: Drs. Nathan M. Neihart, Chris Chong-Nuen Chu, Zhengdao Wang and Jaeyoun (Jay) Kim, for their encouragement, insightful comments, and stimulating questions. It has been a great experience to present my work to them and discuss it with them.

In the past five years, I made many good friends at Iowa State University; and I sincerely thank them for their help and companionship. I will never forget the classes we attended together, and the way they worked closely with me on struggling through projects; I also cannot forget the parties where we played together, and the funny moments I enjoyed with them. I will always remember those who supported me, Ames and Iowa State University.

There is a Chinese saying: A mother always worries about her traveling child. No matter how far away I am from home, my parents and sister always care for me and support me spiritually. I owe them a lot, and thank them for their love and support.

At the end of the story, Frodo and Sam saved their world after completing their journey to destroy the Ring. In my life, my five-year journey to pursue the PhD has greatly changed me, enriching my professional knowledge, building my life skills and strengthening my ability to face any difficulties. I feel confident as I start the next challenging journey in my life now.

We must keep running and hold on to our dreams!

ABSTRACT

Over the past 50 years, the integrated circuit (IC) industry has grown rapidly, following the famous "Moore's law." The process feature size keeps shrinking, whereby the performance of digital circuits is constantly enhanced and their cost constantly decreases. However, with the system integration and the development of system on chip (SoC), nearly all of today's ICs contain analog/mixed-Signal circuits. Although a mixed-signal SoC is primarily digital, the analog circuit design and verification consume most of the resources, and the dominant source of IC breakdowns is attributable to the analog circuits.

One important reason for the high cost and risk of breakdowns of analog circuits is that the technology advancement does not benefit many analog and mixed-signal circuits, and in fact imposes higher requirements on their performance. With process scaling, many important parameters of integrated circuit components degrade, which cause a drop in many key aspects of performance of analog circuits. Many analog circuits rely on matched circuit components (transistors, resistors, or capacitors) to achieve the required linearity performance; examples are amplifiers, digital-to-analog converters (DACs), etc. However, shrinking of the feature sizes increases the circuit components mismatch, thereby making it more difficult to maintain circuit accuracy.

Therefore, to reduce the cost of analog circuit design, designers should propose new structures whose key performance can be improved by the technology scaling. In this dissertation, we propose a low-cost, high-precision DAC structure based on ordered element matching (OEM) theory. High matching accuracy can be achieved by applying OEM calibration to the resistors in unary weighted segments and calibrating the gain error between different segments by calibration DAC (CalDAC). As a design example to verify the proposed structure, a high-precision DAC is designed in a 130 nm Global Foundry (GF) CMOS process. The 130 nm GF process features high-density digital circuits and is a typical process which is constantly enhanced by

the scaling of device dimensions and voltage supply; implementation of a high-precision DAC in such process is important to decreasing the costs of high-precision DAC designs. As a result, our proposed DAC structure is demonstrated to be able to significantly lower the cost of high-precision DAC design.

Another reason for the high cost and risk of breakdowns of analog circuits arises from the complexity of analog circuit working states. Most digital circuits serve as logic functions, so that digital transistors work in only two states, either low or high. In contrast, analog circuits have much more complicated functions; they may work in multiple operating points, since various feedback approaches are applied in analog circuits to enhance their performance. Circuits with undetected operating points can be devastating, particularly when they are employed in critical applications such as automotive, health care, and military products. However, since the existing circuit simulators provide only a single operating point, recognizing the existence of undesired operating points depends largely on the experiences of designers. In some circuits, even the most experienced designers may not be aware that a circuit they designed has undesired operating points, which often go undetected in the standard simulations in the design process.

To identify undesired operating points in an analog circuit and reduce its risk of breakdowns, a systematic verification method against undesired operating points in analog circuits is proposed in this dissertation. Unlike traditional methods of finding all operating points, this method targets only searches for voltage intervals containing undesired operating points. To achieve this, our method first converts the circuit into a corresponding graph and locates the break point to break all the positive feedback loops (PFLs). For one dimensional verification, divide and contraction algorithms could be applied to identify undesired operating points. Two dimensional vector field methods are used to solve the two dimensional verifications. Based on the proposed verification methods against undesired operating points, an EDA tool called "ITV" is developed to identify undesired operating points in analog and mixed-signal circuits. Simulation results show ITV to be effective and efficient in identifying undesired operating points in a class of commonly used benchmark circuits that includes bias generators, voltage references, temperature sensors, and op-amp circuits.

CHAPTER 1. INTRODUCTION

1.1 Background

With the development of the IC industry, the mixed-signal SoC, which integrates all components of a computer or other electronic systems, has become more and more widely used. It may contain digital circuits, analog, mixed-signal, and other function blocks, all on a single chip. Compared with a combination of individual chips, an SoC has a great cost advantage. First, the SoC is much smaller because of its high integration. This allows the complete computer to be put into smart phones and tablets and still leave plenty of room for batteries. In addition, because of its very high level of integration and much less wiring, an SoC uses considerably less power, which is a big advantage for mobile computing. Moreover, decreasing the number of physical chips makes it much cheaper to build a computer or electronic system using an SoC [1]. Although an SoC is composed mainly of digital circuits, the analog circuit design and verification consumes more than 75 percent of the total resources [2]. Moreover, more than 78 percent of electronic breakdowns are due to the analog circuits [3]. The high cost and risks of breakdowns in analog circuits are due to the great differences in design and verification between analog and digital circuits.

Although the performance of digital circuits is constantly enhanced by the scaling of device dimensions and voltage supply, this technology advancement does not benefit many analog and mixed-signal circuits and in fact imposes higher requirements on their performance. With process scaling, many important parameters of the integrated circuit components degrade, which causes decreases in key performance of analog circuits. One important specification of analog circuits is their linearity, which is dependent on matched circuit components such as transistors, resistors, or capacitors. For example, linearity performance of amplifiers mainly relies

on the matching of their input transistor pairs. Another example is seen in the high-precision digital-to-analog converter (DAC), which relies on matched resistor arrays to perform its data conversion tasks. However, the shrinking of feature sizes increases the circuit components mismatch [4], thereby requiring greater effort to maintain circuit accuracy. With process scaling, some analog circuits may maintain or improve their linearity performance by use of a larger die area or more complicated calibration circuits. However, other analog circuits, such as high-precision DACs can be implemented only by high-precision analog process, and can barely be compatible with the process features high-density digital circuits. Thus, it is impossible to integrate analog circuits such as high-precision DACs into an SoC system, and the cost of implementing such circuits is high and hardly benefits from the process scaling.

The high risk of the analog circuit arises from the complexity of its working states. Most digital circuits serve as logic functions, so that digital transistors work in only two states, either low or high, which makes tests for digital circuits easily standardized. Standard test technologies such as scan, JTAG, LBIST, MBIST are all widely used to address the digital circuit tests problem, which explains the low failure rate of digital circuits. In contrast, analog circuits have much more complicated functions; they amplify signals, stabilize the power supply, or convert digital signals to analog signals, among others. Because of the complicated functions of analog circuits, there is no standard way to test and verify them. Most of the analog tests depend only on the experience of test engineers, who have to write as many tests as possible to cover the specification in the time available and to maintain the test cost in a reasonable range relative to the selling price [3]. Since the analog test is by far the dominant test cost, it has even changed the role of analog circuit design engineers. Formerly, they focused only on circuit designs, but now they must be involved in numerous analog circuit designs and verification to reduce the test costs. However, even with the help of circuit designers, some analog circuit test and verification problems still cannot be detected, resulting in circuit failures.

The existence of multiple operating points is one of the most important, although often ignored, problems in analog circuit test and verification. While various feedback approaches, e.g., self-biasing [5], bootstrapping [6], and digitally-assisted-analog [7] have been applied to analog circuits, these may make a system vulnerable to multiple operating points. The exis-

tence of undetected operating points can be devastating, particularly in circuits employed in critical systems such as automotive, health care, and military products. However, since the existing circuit simulators provide only a single operating point [8], recognizing the existence of undesired operating points largely depends on the experience of designers. In some circuits, even the most experienced designers are not aware that a circuit they designed has undesired operating points, which often go undetected in the standard simulations in the design process. Since most analog circuits are not kept in constant working environment now, and because of Process/Voltage supply/Temperature (PVT) variations, the existence of undesired operating point may be detected. It can be very costly if an undesired operating point in the circuit is first detected in the field by a customer. On the other hand, more and more basic circuits are being designed by new graduates or inexperienced analog engineers and circuit designers who continuously adding "smart components" in their design utilizing feedback. Consequently, undesired operating points are becoming an increasingly widespread and insidious problem plaguing the circuit design industry.

1.2 Contribution of this Dissertation

To reduce the cost of analog circuit design, designers should propose new structures whose key performance can be improved by the technology scaling. In this dissertation, we use the high-precision DAC circuit as a low-cost design example and propose a low-cost, high-precision DAC structure based on OEM theory. The DAC is one of the circuits for which demands for high-accuracy requirements are increasing in precision medical, instrumentation, and test and measurement applications. The existing high-precision DACs require a large die area, high-precision analog process, advanced resistor trimming technique, complicated calibration circuits and additional test costs. As a result, their cost is high and difficult to reduce, since their implementation can hardly benefit from the scaling of digital circuits.

Random mismatch errors in resistor networks are dominant nonlinearity sources in high resolution and high accuracy resistor DACs. It has been rigorously proven and verified that ordered element matching (OEM) technology could significantly reduce the random mismatch errors and improve the linearity performance of DACs. More importantly, implementation

of this technique relies on digital circuits only, and its test method is consistent with the traditional INL test, making it a very prosperous technique of implementing a low-cost, high-precision DAC. In this dissertation, a low-cost, high-precision DAC structure based on OEM theory is proposed. It can achieve high matching accuracy by applying the OEM calibration to the resistors in unary weighted segments and calibrating the gain error between different segments by calibration DAC (CalDAC).

As a design example to verify the proposed structure, a high-precision DAC is designed in a 130 nm Global Foundry (GF) CMOS process. The 130 nm GF process features high-density digital circuits but lacks high-precision resistors or any resistor trimming techniques, making it generally unsuitable for any high-precision DAC design. However, we implemented our design in such process from a behavioral model to a schematic and a layout design. The simulation and measurement results show the proposed DAC structure can greatly reduce the area requirement and make it possible to implement a high-precision DAC without use of a high-precision fabrication process. Since the 130 nm GF process is a typical process that is constantly enhanced by the scaling of the device dimensions and voltage supply, implementation of a high-precision DAC in such a process is an important means of decreasing the cost of high-precision DAC design. As a result, our proposed DAC structure is demonstrated to be able to significantly lower the cost of high-precision DAC design.

To reduce the cost and risk of breakdowns of analog circuits, identifying and removing undesired operating points is one of the most important problems. In this dissertation, a divide and contraction verification method against undesired operating points in analog circuits is proposed. Unlike traditional methods to find all operating points, this method searches for only the voltage intervals containing undesired operating points. To achieve this, a systematic approach to automatically identifying all positive and negative feedback loops in circuits is introduced. A positive feedback loop breaking method and selection of breaking nodes are discussed to determine whether a monotonic return function can be obtained. Depending on the monotonicity of the return function, two types of divide and contraction algorithms are proposed to efficiently search for voltage intervals containing operating points.

In practice, designers also need to verify circuits with transistor sizing, PVT variations; or identify the existence of undesired operating point in complicated circuits (B2P circuits). This type of problems is called the two dimensional analog verification against undesired operating points. For this type of verification, a two dimensional vector field method is proposed that can effectively identify the existence of undesired operating points by visualizing the return functions in the circuits.

Based on the proposed verification methods used against undesired operating points, an EDA tool called "ITV" is developed to identify undesired operating points in analog and mixed-signal circuits. It can accomplish this on the basis of the break-loop Homotopy method. It first converts the circuit into a corresponding graph and locates the break point to break all of the positive feedback loops (PFLs). It then searches the voltage intervals that contain undesired operating points by use of divide and contraction algorithms or the two dimensional vector field method. Simulation results show ITV to be effective and efficient in identifying undesired operating points in a class of commonly used benchmark circuits, including bias generators, voltage references, temperature sensors, and op-amp circuits.

1.3 Organization of this Dissertation

This dissertation is organized as follows: Chapter 2 provides the proposed low-cost, high-precision DAC structure and its design methodology. In Chapter 3, design and measurement of a high-precision DAC based on the proposed structure is presented. Chapter 4 then develops a high efficient divide and contraction method to verify the existence of undesired operating points in analog circuits. To accomplish the two dimensional analog verifications, Chapter 5 proposes a two dimensional vector field method. In Chapter 6, an EDA tool called "ITV" is introduced to implement the proposed methods of identifying the undesired operating points in analog circuits. Finally, conclusions are presented in Chapter 7.

CHAPTER 2. LOW-COST, HIGH-PRECISION DAC DESIGN BASED ON ORDERED ELEMENT MATCHING

2.1 Introduction

Until now, the performance of digital circuits has been constantly enhanced by the scaling of device dimensions and voltage supply. However, the technology advancement does not benefit many analog and mixed-signal circuits and in fact it imposes higher requirements on their performance. The digital-to-analog converter (DAC) is one of the circuits seeing a demand for increased high accuracy in precision medical, instrumentation, and test and measurement applications [9]. High-precision 12-bit DACs were once considered to be difficult to implement; however, 16-bit accuracy is now widely used in high-precision applications. Recently, even a 20-bit, 1-ppm-accurate DAC integrated circuit was proposed to meet the needs of the precision instrumentation market [10].

Most high-precision DACs rely on accurate resistor arrays to perform data conversion tasks, so their accuracy is very sensitive to the matching performance of the resistor networks. However, the integrated-circuit (IC) fabrication technology cannot produce perfectly matched resistors, and with the process scaling, the random mismatch errors increase significantly [11, 12]. Therefore, implementing a high-precision DAC is very costly and depends on many different techniques.

The most important cost in achieving a high-precision DAC is the resistor area on the chip. In general, 1-bit linearity enhancement leads to quadrupling of the circuit area [13]. Nevertheless, the maximum allowed area is limited by the available die size as growing numbers of circuits and systems are integrated into a single chip. Therefore, the implementation of a high-precision DAC requires high-precision analog processes with high-precision resistors, such

as silicon-chromium thin-film resistors [10]. Unfortunately, such processes are usually associated with large transistor feature size and do not scale.

Aside from the resistor area, resistor trimming is often applied to achieve a high-precision DAC, which also demands high cost. Popular trimming techniques are divided into two main categories: laser trimming and fuse trimming [14, 15]. Laser trimming employs laser beams to accurately adjust the resistor parameters at the wafer level, while fuse trimming utilizes a fuse or anti-fuse for opening or closing the interconnections of a network of resistive elements intended to minimize mismatch errors. However, trimming techniques usually require high expenses such as extra layers or more die area for trim pads, and the achieved accuracy is reduced by temperature and aging effects [14].

High-accuracy calibration circuits are usually employed in high-precision DACs. They decrease the mismatch errors by either digital or analog feed-back signals from error-measuring circuits, such as a high-resolution high-accuracy analog-to-digital converter (ADC) [16, 17] or DAC [18, 19, 20]. However, the price of implementing those calibration circuits is high. It requires accurate measurement and complicated feedback circuits, which occupy a large silicon area, especially for high-precision analog processes whose feature sizes are large. In addition, specific calibration tests, other than the general DAC tests, are required to implement the calibration, making the expense of calibration circuits even higher.

Dynamic element matching (DEM) is another popular means of implementing a high-precision DAC. It dynamically changes the positions of mismatched elements at different times so that the equivalent component at each position is nearly matched on a time average. Several popular DEM algorithms are available, such as butterfly randomization [21], individual level averaging [22], and data weighted averaging [23]. Unlike the static random mismatch compensation techniques, DEM translates mismatch errors into noise. However, the translated noise is only partially shaped where the in-band residuals could possibly affect the data converters signal-to-noise ratio (SNR) [24]. Furthermore, the output will be inaccurate at one instant of time, since DEM guarantees matching only on average, so that its applications are limited to sigma delta modulators.

In summary, existing high-precision DACs require a large die area, high-precision analog processes, advanced resistor trimming techniques, complicated calibration circuits and additional test costs. As a result, their cost is high and difficult to reduce, because their implementation benefits very little from the scaling of digital circuits.

A different approach, called ordered element matching (OEM), was proposed in [25, 13, 26, 27] to implement high-precision DACs. The OEM technique can significantly reduce random mismatch errors and improve the linearity performance of DACs. More importantly, its implementation relies only on digital circuits and the test method is consistent with traditional INL tests [27], making it a very promising technique of implementing a low-cost, high-precision DAC. Based on the OEM technique, a high-precision segmented DAC structure has been introduced in [28] and has achieved accuracy of about 20-bit from the MATLAB simulation. Nevertheless, its implementation needs additional switch circuits to calibrate the gain errors between different segments, which requires extra circuits area and may cause leakage problems.

In this dissertation, a low-cost, high-precision DAC structure based on the OEM technique is presented. It consists of three segments: binary weighted least significant bits (LSB), unary weighted intermediate significant bits (ISB) and most significant bits (MSB). The optimized OEM process is applied to the unary weighted ISB and MSB segments. A sub-radix-2 calibration DAC (CalDAC) is implemented to calibrate the gain errors between ISB and MSB segments. On the basis of a 130 nm CMOS technique, a high-precision DAC is implemented to demonstrate the design methodology of the proposed DAC structure.

This chapter is organized as follows. In section 2, the proposed high-precision DAC structure based on OEM is introduced; section 3 discusses the design of the gain calibration pseudo DAC included in the proposed DAC structure; section 4 illustrates the OEM calibration process and implementation; behavioral simulation results are provided in section 5, and conclusions are stated in section 6.

2.2 Low-cost, High-precision DAC Structure Based on Ordered Element Matching

In this section, a low-cost, high-precision DAC structure based on OEM is illustrated. First, the OEM technology used for reducing the mismatch error is introduced. Then a segmented high-precision DAC structure is proposed and analyzed. To minimize the gain error between different segments, a gain calibration DAC (CalDAC) is integrated into the proposed DAC structure and its design is discussed. The process of applying OEM calibration and its implementation in digital circuits are then interpreted.

2.2.1 OEM Technology

Fig. 2.1 shows the process of OEM technology. Each rectangle denotes a component with random mismatch errors in the unary-weighted segment. Component can be a resistance, conductance or any other amplitude whose random mismatch errors are targeted to be minimized [29], with X_{AVG} as the average amplitude. First, all components are measured and sorted according to their amplitudes. The second step is to choose the component with amplitude closest to X_{AVG}. Third, complementary ordered components are paired, which is called one "folding". The original 3-bit unary coded component array is converted into a 2-bit unary-weighted and 1-bit binary-weighted array. In detail, the amplitude of each 2-bit unary weighted array is nearly twice of the X_{AVG}, and the random variations in the components are reduced. Mismatch errors consistently diminish after each choosing and folding operation. As shown in Fig. 2.1, if the choosing and single folding operations are repeated until the 3-bit unary weighted array is converted into a 3-bit binary weighted array, mismatch errors are further reduced. This process is called "OEM binarization."

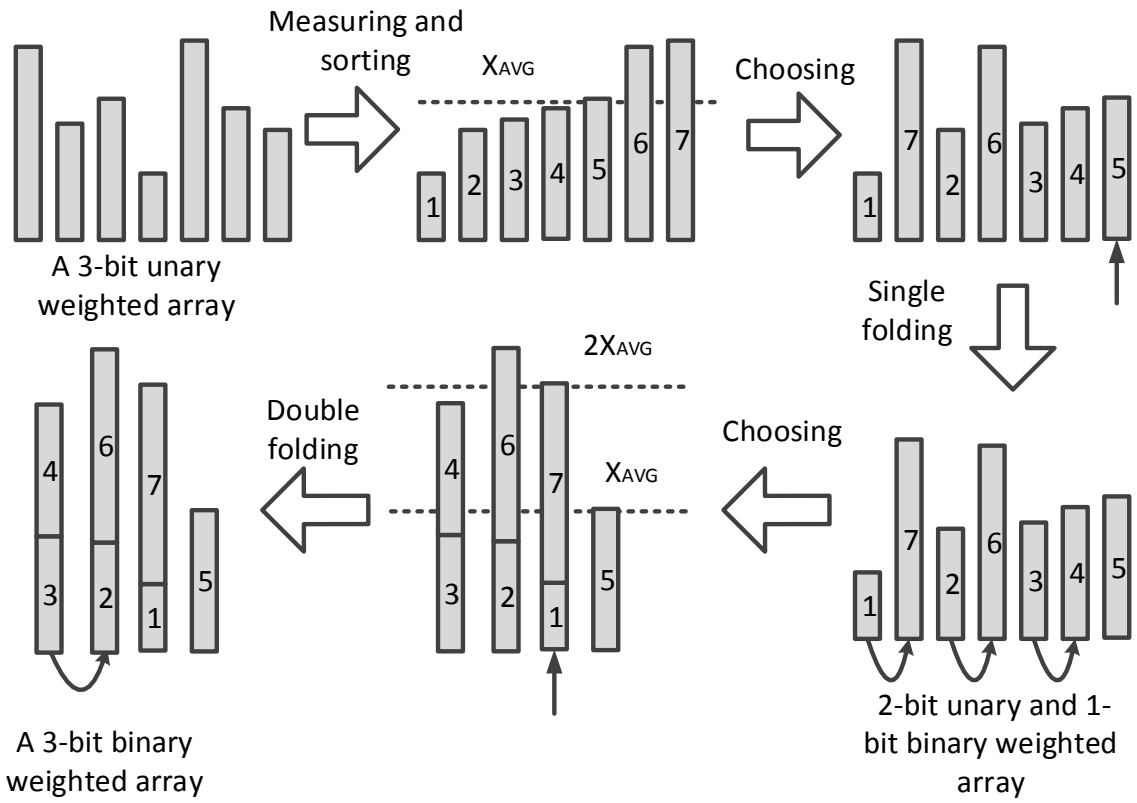


Figure 2.1: OEM Binarization

2.2.2 High-precision DAC Structure

Fig. 2.2 shows the proposed high-precision DAC structure based on OEM. Although the design requires an external feedback resistor that creates noise and endpoint errors [10], the commonly called current-mode R-2R ladder network DAC is selected, for the following reasons. First, the current-mode structure avoids the resistor non-linearity caused by self-heating, which is the dominant contributor to resistor non-linearity [30]. In the voltage mode R-2R ladder network, different currents flow through different resistors and the currents vary with the DAC codes, causing the non-linearity problem. In contrast, all resistors in the same segment of the current mode R-2R ladder network have the same current, which do not change with the DAC codes. Thus, self-heating will not cause non-linearity errors in the current-mode structure, saving the cost of calibrating the resistor non-linearity that is due to self-heating. The second reason to use the current-mode structure is to minimize the INL error caused by unequal re-

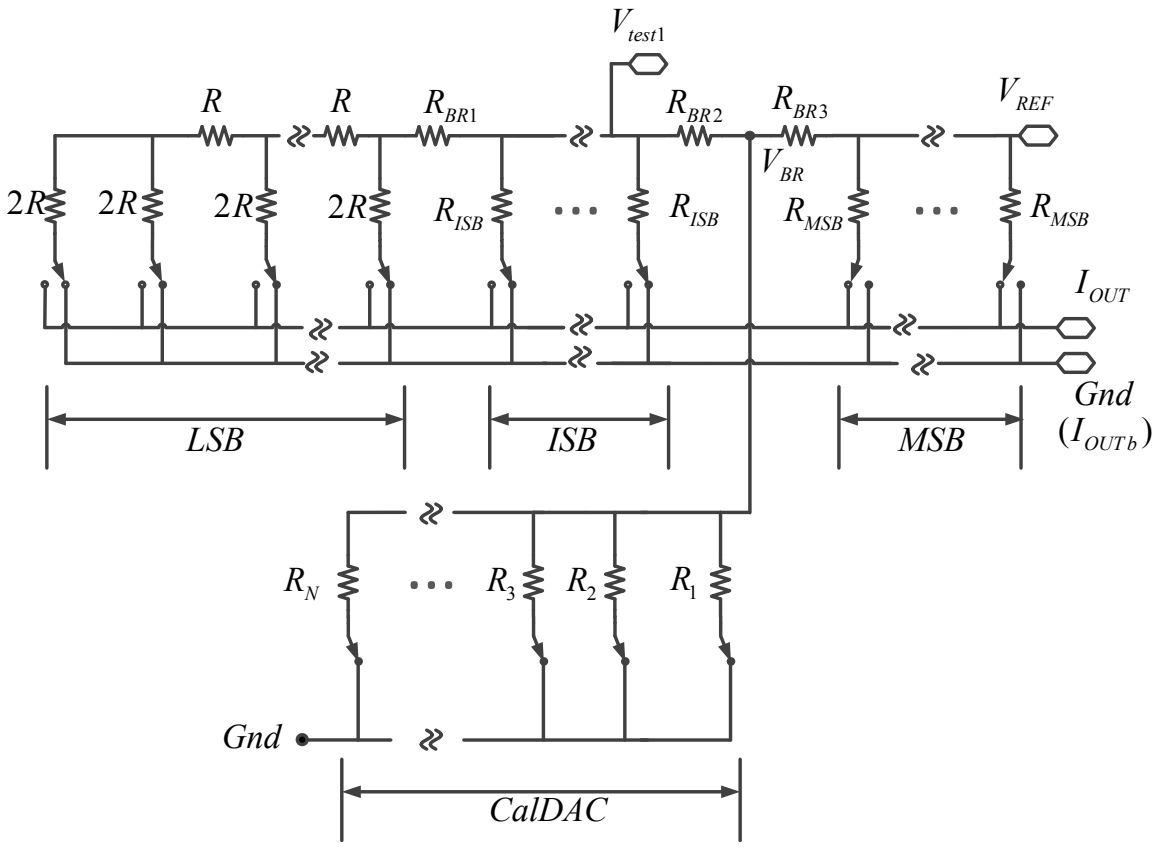


Figure 2.2: Proposed OEM Based High-precision DAC Structure

sistance in switch pairs. In a voltage-mode structure, pairs of PMOS and NMOS switches are usually used. To minimize the INL error due to the different on-resistance between PMOS and NMOS, techniques [31] and[10] are needed, requiring high-performance amplifiers and other circuitry on the chip for high-precision DACs. However, the current-mode structure uses pairs of NMOS switches, whose on-resistance difference can be minimized by using common-centroid layout techniques and reasonable switch areas. As a result, the current-mode structure is preferable for a low-cost, high-precision DAC design. Endpoint errors associated with the external feedback resistor can be minimized by using high-accuracy feedback resistors. Assuming the high-precision DACs are mostly used in pseudo DC applications, the noise problem caused by the external feedback resistor is negligible, since the bandwidths of high-precision DACs are narrow.

To meet the linearity target of DACs, the area requirements of resistors are often calculated by the resistors matching level, as shown in the following subsection. The matching level of the resistance of a resistor is the same as its corresponding conductance, as shown in

$$G = \frac{1}{R} \Rightarrow \Delta G = \frac{-\Delta R}{R^2} \Rightarrow \frac{\Delta G}{G} = -\frac{\Delta R}{R} \tag{2.1}$$

where resistance R is the resistance of a resistor and G is its corresponding conductance. Yet, the mismatching level of resistors is easily found in a process design manual instead of their corresponding conductance mismatching level. As a result, resistors in ISB and MSB segments are marked R_{ISB} and R_{MSB} in Fig. 2.2. However, because the proposed DAC is a current output DAC, its linearity performance is determined by the output current relationship with the DAC input codes. In the unary weighted segments, the resulting output current of each resistor is determined by its conductance. For instance, the output current generated by the ith MSB resistor is $I_{out(i)} = V_{REF}/R_{MSB(i)} = V_{REF} \times G_{MSB(i)}$. As a result, OEM binarization should be applied to conductances rather than resistances of ISB and MSB segments to improve their linearity performance.

The proposed DAC structure consists of three segments: the binary weighted LSB, unary weighted ISB and MSB. Compared with the traditional two-segment DAC structure [10], the proposed DAC structure achieves higher linearity performance by application of OEM binarization to both the ISB and MSB segments. A unary weighted array with more than 7 bits requires digital circuits that are too complicated for OEM binarization to be applied to. OEM binarization cannot be utilized in binary weighted array, and LSB segment accomplishes its targeted accuracy through the resistor's intrinsic matching. For a 16 $\sim$ 20 bit high-precision two-segment DAC structure, the intrinsic matching of the resistors in the binary weighted LSB should be above the 9 to 13 bit level, assuming the MSB segment is 7-bit and calibrated by OEM binarization. However, resistors with a 9 to 13 bit matching level require large areas if high-precision analog processes are not being used. Thus another unary weighted ISB segment is needed in high-precision DAC design. By applying OEM binarization to both the ISB and MSB segments, the matching level of resistors in the two segments can be significantly improved [28]. Moreover, the requirement of intrinsic matching level of resistors in the LSB segment is

greatly relaxed, thereby reducing the area of LSB resistors. For example, a 20-bit DAC with 5-bit ISB and 7-bit MSB requires only 8-bit matching resistors in the LSB segment, which significantly decreases the area of the LSB segment and the total area of the DAC. However, adding the ISB segment into the DAC produces a gain error problem between the ISB and MSB segments, which can be minimized by a gain calibration method discussed in the next section.

2.2.3 Segmentation Choices

For the proposed DAC structures, the segmentation choice should be carefully considered. First, resistors in different segments do not have the same effect on DAC linearity performance. In INL, the contribution of LSB resistor non-linearity could be approximated as

$$INL_{(R_{LSB})} = 2^{nlsb-0.5} \times \sigma_{R_{LSB}} \tag{2.2}$$

where $\sigma_{R_{LSB}}$ is the standard deviation of the LSB resistor random mismatch error, and $nlsb$ is the number of LSB segment bits. Although the current-mode structure is applied in the design, the variation of the output current depends mainly on the difference between resistors.

The contribution of the ISB and MSB resistors non-linearity could also be separately estimated as

$$INL_{(R_{ISB})} = 2^{nlsb+nisb/2+0.5} \times \sigma_{R_{ISB}} \tag{2.3}$$

$$INL_{(R_{MSB})} = 2^{nlsb+nisb+nmsb/2+0.5} \times \sigma_{R_{MSB}} \tag{2.4}$$

where $\sigma_{R_{ISB}}$ is the standard deviation of the ISB resistor random mismatch error, $\sigma_{R_{MSB}}$ is the standard deviation of the MSB resistor random mismatch error, $nisb$ is the number of ISB bits and $nmsb$ is the number of MSB bits.

After applying the OEM process as described in sub-section 1.2.2, the resulting INL caused by random mismatch errors in ISB/MSB segments is modeled as

$$INL'_{(R_{ISB})} = 2^{nlsb+nisb/2+0.5-X1} \times \sigma_{R_{ISB}} \tag{2.5}$$

$$INL'_{(R_{MSB})} = 2^{nlsb+nisb+nmsb/2+0.5-X2} \times \sigma_{R_{MSB}} \tag{2.6}$$

Table 2.1: OEM Reduction Factor

Bit number	5	6	7
x1	3	4	4.5
x2 (w/20% outlier)	4.5	5.5	7

where $X1$ and $X2$ are called OEM INL reduction factors. For the MSB segment, an outlier elimination technique is used to further improve the matching performance [13], and about 20% being the best percentage of outlier number. The OEM INL reduction factors vary with process and resistor areas, but their approximate values can be obtained as shown in TABLE 2.1.

The required area of resistors is estimated from the standard deviation of their mismatch error

$$A_{req} = A_u(\frac{\sigma_u}{\sigma_{req}})^2 \tag{2.7}$$

where σ_u is the standard deviation of unit resistor, σ_{req} is the required standard deviation of the mismatch errors, and A_u is the area of unit resistor. From equations (2.2) (2.5) (2.6) (2.7), the total area can be estimated as follows:

$$\begin{aligned} A_{total} =& (A_u(\frac{\sigma_u}{\log_2(INL_{MSB}/2^{nlsb+nisb+nmsb/2+0.5-X2})})^2 + A_{Digital} + A_{SW_{MSB}}) \\ & \times (2^{nmsb}-1) \times 1.2 + (A_u(\frac{\sigma_u}{\log_2(INL_{ISB}/2^{nlsb+nisb/2+0.5-X1})})^2 + A_{Digital} + A_{SW_{ISB}}) \\ & \times (2^{nisb}-1) + (A_u(\frac{\sigma_u}{\log_2(INL_{ISB}/2^{nlsb-0.5})})^2 + A_{SW_{LSB}}) \times (3nlsb+1) \end{aligned} \tag{2.8}$$

where $A_{Digital}$ is the OEM digital circuit for each resistor (illustrated in section 4), $A_{SW_{MSB}}$, $A_{SW_{ISB}}$ and $A_{SW_{LSB}}$ are the single switch areas for the MSB, ISB and LSB segments, respectively.

Equation (2.8) shows more unary weighted bits (more bits in the ISB and MSB segments) the linearity performance further improves, decreasing the area of resistors. However, with more unary-weighted bits, more digital circuits are required to implement OEM. Thus a trade-off exists between the digital circuit and the resistor area when segmentation is performed.

Another important consideration related to segmentation is power consumption. The power consumption of a DAC is

$$P = V_{ref}^2 / R_{output} \tag{2.9}$$

where $R_{output} = R_{msb}/2^{nmsb}$ is the output impedance of the DAC. Thus, the unit resistor value and the number of bits in the MSB segment are limited by power consumption. In most general analog processes, resistors with higher matching performance usually have lower resistance density. The large R_{output} needed to reduce power consumption may cause large area requirements for resistors with high matching performance but low resistance density. On the other hand, with one more bit in the MSB segment, the required resistance of the MSB resistor doubles for the same power consumption. Therefore, the number of bits in the MSB segment should be carefully selected, not only on the basis of the matching level to be achieved, but also on the basis of power consumption limitations.

2.3 Gain Calibration Pseudo DAC Design

2.3.1 Gain Error between Different Segments

Aside from mismatch errors in each segment, an important non-linearity error in the segmented DAC is the gain error between different segments. For example, in an ideal matched three-segment DAC, the following relation exists between the ISB and MSB segments:

$$V_{ref} G_{ILSB}(2^{N-nmsb}) = V_{ref} G_{MSB}(1) \tag{2.10}$$

where N is the total bits of the DAC, $nmsb$ is the bits of MSB segment, $G_{MSB}(1) = (R_{MSB(1)})^{-1}$ is the conductance of the lowest bit of MSB segment, and $G_{ILSB}(2^{N-N_{msb}})$ is the total conductance of the LSB and ISB segments. The gain error between the ISB segment and the MSB segment from (2.10) generates the following output current:

$$\Delta I_{Gain} = V_{ref}(G_{ILSB}(2^{N-nmsb}) - G_{MSB}(1)) = V_{ref}\Delta G. \tag{2.11}$$

Since the lowest bit of MSB stands for 2^{N-nmsb} LSB of the whole DAC, the following equation is required to make the total INL error less than 1 LSB:

$$|\Delta I_{Gain}| < \frac{I_{MSB}(1)}{2^{N-nmsb}}. \tag{2.12}$$

Similar analysis is applied to the gain error between the ISB and LSB segments. Gain error between different segments must be minimized in a high-precision DAC.

In the structure proposed in [28], gain calibration is achieved by changing different bridge resistors, and both the gain error between the LSB and ISB segments and the gain error between ISB and MSB segments are calibrated. Nevertheless, its implementation needs additional switch circuits to calibrate the gain errors between different segments, causing additional circuit area and potential leakage problems. Moreover, the gain error between the LSB and ISB segments can be minimized by increasing the area of the R_{br1}, and the gain error between the ISB and MSB segments is the dominant gain error. As a result, in the proposed DAC structure shown in Fig. 2.2, no additional switches in the unary weighted segments to calibrate the gain errors between different segments are used. Instead, the gain error between LSB and ISB is minimized by enlarging the area of the R_{br1}, and a calibration pseudo DAC (CalDAC) is inserted between the ISB and MSB segments for calibrating the gain errors.

2.3.2 Gain Calibration DAC

The results of (2.11) and (2.12) are

$$|\Delta G| = |G_{ILSB}(2^{N-nmsb}) - G_{MSB}(1)| < \frac{G_{MSB}(1)}{2^{N-nmsb}}, \tag{2.13}$$

creating,

$$\begin{aligned} &G_{MSB}(1)(1 - \frac{1}{2^{N-nmsb}}) < G_{ILSB}(2^{N-nmsb}) < G_{MSB}(1)(1 + \frac{1}{2^{N-nmsb}}) \Rightarrow \\ &- \frac{1}{2^{N-nmsb}+1} R_{MSB(1)} < R_{ILSB}(2^{N-nmsb}) - R_{MSB}(1) < \frac{1}{2^{N-nmsb}-1} R_{MSB}(1) \end{aligned} \tag{2.14}$$

where $R_{ILSB}(2^{N-nmsb}) = (G_{ILSB}(2^{N-nmsb}))^{-1}$ and $R_{MSB}(1) = (G_{MSB}(1))^{-1}$.

Assuming $2^{N-nmsb} >> 1$, (2.14) is simplified to

$$|R_{ILSB}(2^{N-nmsb}) - R_{MSB}(1)| = |\Delta R| < \frac{1}{2^{N-nmsb}} R_{MSB}(1) \tag{2.15}$$

From (2.15), the gain error is reduced to meet the DAC accuracy requirement if $|\Delta R|$ is calibrated to the required level.

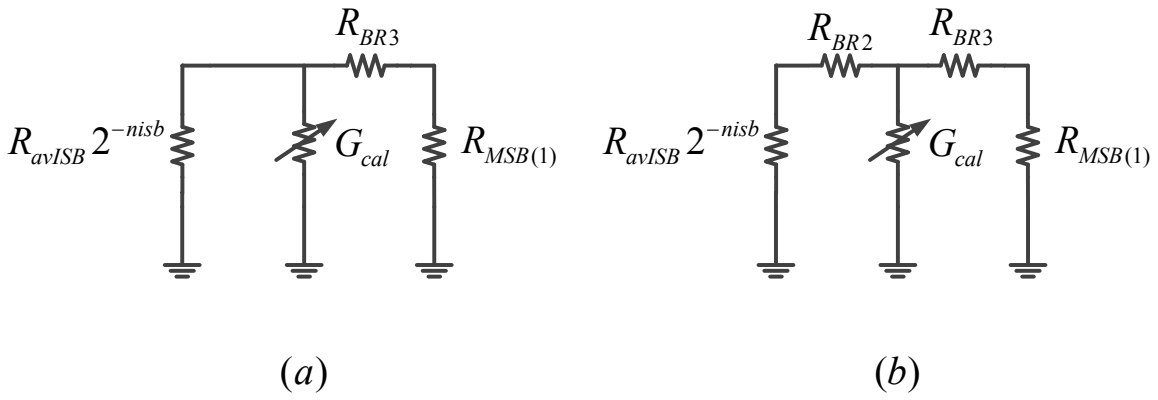


Figure 2.3: (a) First Simplified CalDAC Structure; (b) Second Simplified CalDAC Structure.

Fig. 2.3 shows a comparison of two types of pseudo calibration DAC structures. In the first simplified pseudo calibration DAC structure, shown in Fig. 2.3(a), R_{avISB} is the average value of all the resistors and switch on-resistances in the ISB segment, $nisb$ is the bits of the ISB segment, G_{cal} is the conductance of the CalDAC, and $R_{MSB(1)}$ is the lowest bit resistance of the MSB segment. In this structure,

$$\begin{aligned} R_{ILSB}(2^{N-nmsb}) &= R_{BR3}G_{cal}2^{-nisb}R_{avISB} + 2^{-nisb}R_{avISB} + R_{BR3} \\ &= R_{BR3}G_{cal}2^{-nisb}R_{avISB} + R_{T1} \end{aligned} \tag{2.16}$$

where $R_{T1} = 2^{-nisb}R_{avISB} + R_{BR3}$.

By setting $R_{ILSB}(2^{N-nmsb})|_{nom} = R_{T1}$, the gain error can be reduced to our required accuracy by tuning G_{cal} to minimize ΔR; or we can make

$$R_{ILSB}(2^{N-nmsb})|_{nom} = R_{BR3}G_{cal}|_{nom}2^{-nisb}R_{avISB} + R_{T1} = R_{MSB(1)} \tag{2.17}$$

where $G_{cal}|_{nom}$ is the nominal value of the G_{cal}. Both cases require $R_{T1} < R_{MSB(1)}$.

The second simplified CalDAC structure is shown in Fig. 2.3(b); its $R_{ILSB}(2^{N-nmsb})$ is calculated as

$$\begin{aligned} R_{ILSB}(2^{N-nmsb}) =& R_{BR3}G_{cal}(2^{-nisb}R_{avISB} + R_{BR2}) + 2^{-nisb}R_{avISB} + R_{BR2} + R_{BR3} \\ =& R_{BR3}G_{cal}(2^{-nisb}R_{avISB} + R_{BR2}) + R_{T2}. \end{aligned} \tag{2.18}$$

where $R_{T2} = 2^{-nisb}R_{avISB} + R_{BR2} + R_{BR3}$. As is the case for the first structure, the gain error can be minimized by G_{cal}.

From (2.16) and (2.18), the ΔR is calibrated to the required accuracy by tuning G_{cal}, thereby reducing the gain errors in both CalDAC structures. However, the required calibration ranges for these two CalDAC structures differ greatly.

Assuming the standard deviation of $R_{MSB}(1)$ variation is $\sigma_{R_{MSB}(1)}$, the largest variation of $R_{MSB}(1)$ can be estimated as $5\sigma_{R_{MSB}(1)} = \Delta R_S$. With assignment of an area similar to $R_{MSB}(1)$, the largest variation of R_{T1} in the first structure and R_{T2} in the proposed structure are the same as ΔR_S. The calibration range of CalDAC is analyzed as shown in Fig.2.4. Meeting the condition $R_{T1} < R_{MSB(1)}$ and $R_{T2} < R_{MSB(1)}$ in the two structures respectively requires that

$$R_{MSB}(1)|_{nom} - R_{T1/2}|_{nom} \geq 2\Delta R_S. \tag{2.19}$$

where $R_{MSB}(1)|_{nom}$ is the nominal value of $R_{MSB}(1)$, and $R_{T1/2}|_{nom}$ is the nominal value of R_{T1} or R_{T2}. Conditions (2.15) and (2.19) require the tuning range of G_{CAL} to calibrate the range

$$\Delta R_{CAL} \geq 4\Delta R_S. \tag{2.20}$$

Together with (2.16) and (2.18), the calibration range for the first CalDAC structure is:

$$\Delta G_{CAL1} \geq \frac{4\Delta R_S}{R_{BR3}2^{-nisb}R_{avISB}}, \tag{2.21}$$

and the calibration range for the second CalDAC is:

$$\Delta G_{CAL2} \geq \frac{4\Delta R_S}{R_{BR3}(2^{-nisb}R_{avISB} + R_{BR2})}. \tag{2.22}$$

When (2.21) is compared with (2.22), the calibration range $\Delta G_{CAL2} < \Delta G_{CAL1}$, which results in a smaller CalDAC design; this is why the calibration DAC structure in Fig. 2.3(b) is used in the proposed DAC structure.

In addition to the calibration range, the calibration step of the CalDAC, i.e., the bits of CalDAC, should also be taken into consideration. Assuming $\Delta R_S = \frac{1}{2^{N_k}}R_{MSB}(1)$, to meet

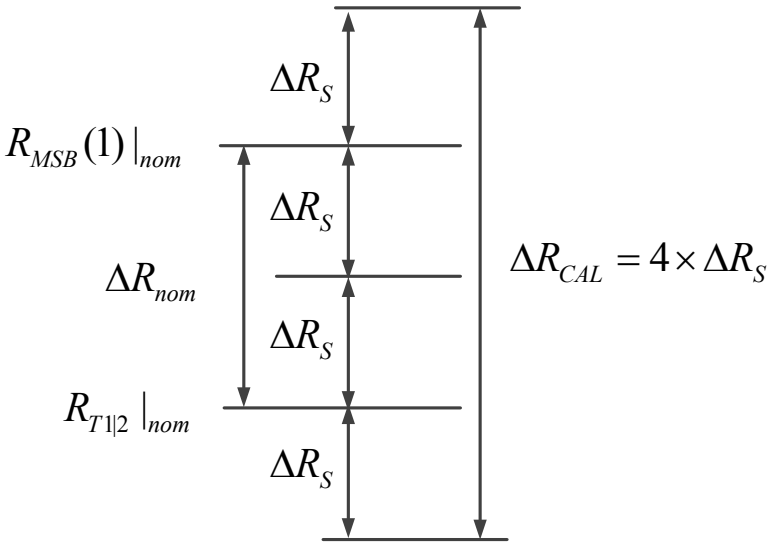


Figure 2.4: CalDAC Range

(2.15), requires the number of codes in the CalDAC

$$N_G \geq \frac{4\Delta R_S}{\frac{1}{2^{N-nmsb}} R_{MSB}(1)} = 2^{N-nmsb-N_k}, \tag{2.23}$$

and thus requires at least $N-nmsb-N_k$ bits CalDAC to calibrate the gain error to the expected accuracy. In practice, extra bits should be added to compensate for the resistor tolerance and variations of processes and temperature.

2.3.3 Implementation of Gain Calibration

In sub-section 1.3.2, gain calibration was discussed; however, its implementation needs to be further investigated. From (2.22) and (2.23), the required calibration range and number of bits can be calculated, respectively. However, there is no linearity requirement for the CalDAC, because it works as a pseudo DAC and its input code is fixed after the best gain calibration. Thus, a sufficiently large calibration range with enough number of calibration steps are enough for implementation the required CalDAC. The most efficient way to implement the CalDAC is to employ the sub-radix-2 DAC, as shown in Fig. 2.5. The output of the CalDAC is connected to node V_{BR}, as shown in Fig. 2.2.

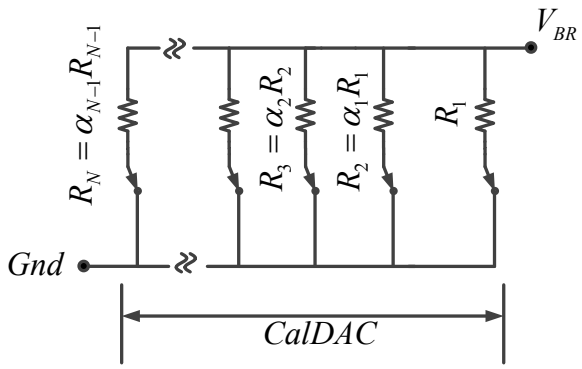


Figure 2.5: CalDAC Implemented As a Sub-radix-2 DAC

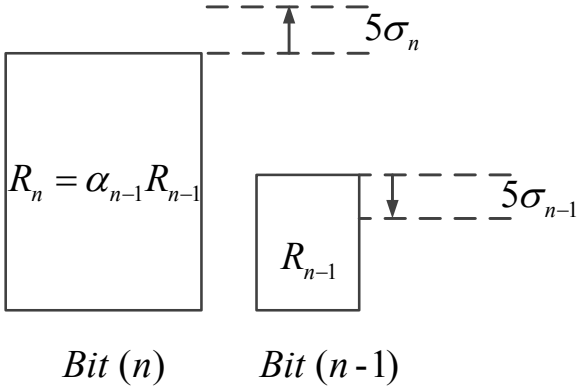


Figure 2.6: CalDAC Ratio

In CalDAC, each branch has a resistor and a switch. By connecting the switch to ground (Gnd), the corresponding bit contributes a conductance $C_i = \frac{1}{R_i}$. The ratio of the resistors between adjacent bits is $\alpha_i < 2$, e.g. $R_2 = \alpha_1 R_1$. The ratio α_i between different bits is determined by the variations of resistance, as shown in Fig. 2.6. To make the ratio between adjacent bits less than two, the following relationship can be used:

$$\alpha_i R_{n-1} + 5\sigma_n < 2 \times (R_{n-1} - 5\sigma_{n-1}) \Rightarrow \alpha_i < 2 - (5\sigma_n + 10\sigma_{n-1})/R_{n-1} \tag{2.24}$$

where R_{n-1} is resistance of $n-1$ bit of CalDAC, α_{n-1} is the resistance between R_n and R_{n-1}, σ_n and σ_{n-1} are the standard deviation of resistance of n and $n-1$ bits of CalDAC, respectively.

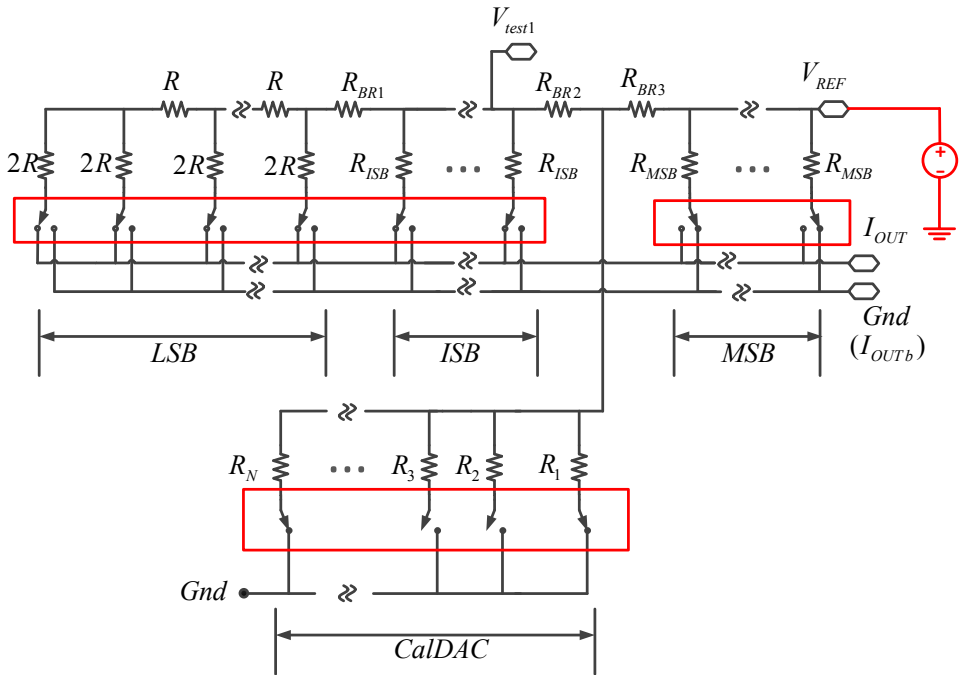


Figure 2.7: The Gain Calibration Process

The gain calibration process is shown in Fig. 2.7. All the LSB and ISB resistors are connected to I_{out} and MSB resistors to Gnd. The CalDAC codes are swept from the smallest to the largest, and the total conductances of LSB and ISB segments for each CalDAC code i are obtained as $G_{ILSB}(2^{N-nmsb})_{CALcode(i)}$. The best CalDAC code which has the minimal value of $|G_{ILSB}(2^{N-nmsb})_{CALcode(i)} - G_{MSB}(1)|$ among the different CalDAC codes, is selected. Therefore, the gain error can be calibrated by the CalDAC.

The gain calibration range (ΔG_{CAL}) of the CalDAC is determined by (2.22). Assuming that ΔR_S, $2^{-nisb}R_{avISB}$ are defined from the segmentation choice, ΔG_{CAL} is modified by changing the R_{BR2} and R_{BR3}. The number of bits for CalDAC ($ncsb$) is determined by (2.23). On this basis, the area of the CalDAC is analyzed as the following:

Without loss of generosity, we assume $ncsb$ is a odd number, and the middle bit of the CalDAC is bit $(ncsb+1)/2$. To simplify the analysis, resistance ratios between adjacent bits α_i are the same and equal to α. Since CalDAC works as a pseudo DAC and there is no linearity requirement for the CalDAC, all resistors in the CalDAC can have the minimum width. As a result, the area of each bit depends on resistance density $R_{\square}$, ΔG_{CAL} and $ncsb$. For example,

Table 2.2: Area of a 9-bit CalDAC

Bit number	Bit-9	Bit-8	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1
Resistance ratio	α^4	α^3	α^2	α	1	$1/\alpha$	$1/\alpha^2$	$1/\alpha^3$	$1/\alpha^4$
Area ratio	α^4	α^3	α^2	α	1	$4/\alpha$	$16/\alpha^2$	$64/\alpha^3$	$256/\alpha^4$

the resistance of the middle bit is

$$R_{(ncsb+1)/2} = \frac{\alpha^{(ncsb+1)/2}}{\Delta G}, \tag{2.25}$$

so its area requirement is calculated as

$$A_{(ncsb+1)/2} = \frac{\alpha^{(ncsb+1)/2}}{\Delta G R_{\square}}. \tag{2.26}$$

The resistance of bit $(ncsb+1)/2+1$ should be $\alpha R_{(ncsb+1)/2}$, which can be implemented by combing two $\alpha R_{(ncsb+1)/2}/2$ resistors series. Similarly, bit $(ncsb+1)/2+1$ is composed of four $\alpha^2 R_{(ncsb+1)/2}/4$ resistors series, etc. The resistance of bit $(ncsb+1)/2-1$ is implemented as two $2R_{(ncsb+1)/2}/\alpha$ resistors in parallel, and bit $(ncsb+1)/2-2$ implemented as four $4R_{(ncsb+1)/2}/\alpha^2$ resistors in parallel. For example, the resistor and area ratio between different bits in a 9-bit CalDAC is summarized in TABLE 2.2. From (2.26), the total area of the CalDAC is calculated as

$$A_{CalDAC} = [\frac{1-\alpha^5}{1-\alpha} + \frac{4(\alpha^4-256)}{\alpha^5-4\alpha^4}]\frac{\alpha^{(ncsb+1)/2}}{\Delta G R_{\square}}. \tag{2.27}$$

Assuming that $\alpha = 1.8$, $ncsb = 9$, and $\Delta G = 1mS$, changing the ratio between R_{BR2} and R_{BR3} from (2.22) results in an A_{CalDAC} of about $1.226E6/R_{\square}$. In a general analog processes, the $R_{\square}$ of the poly resistor is $1000 \sim 2000\Omega/\square$. Thus the A_{CalDAC} for a nine bit CalDAC is only $613 \sim 1226R_{\square}$, which is small enough to be regarded as negligible compared with the area of the resistor array in the DAC.

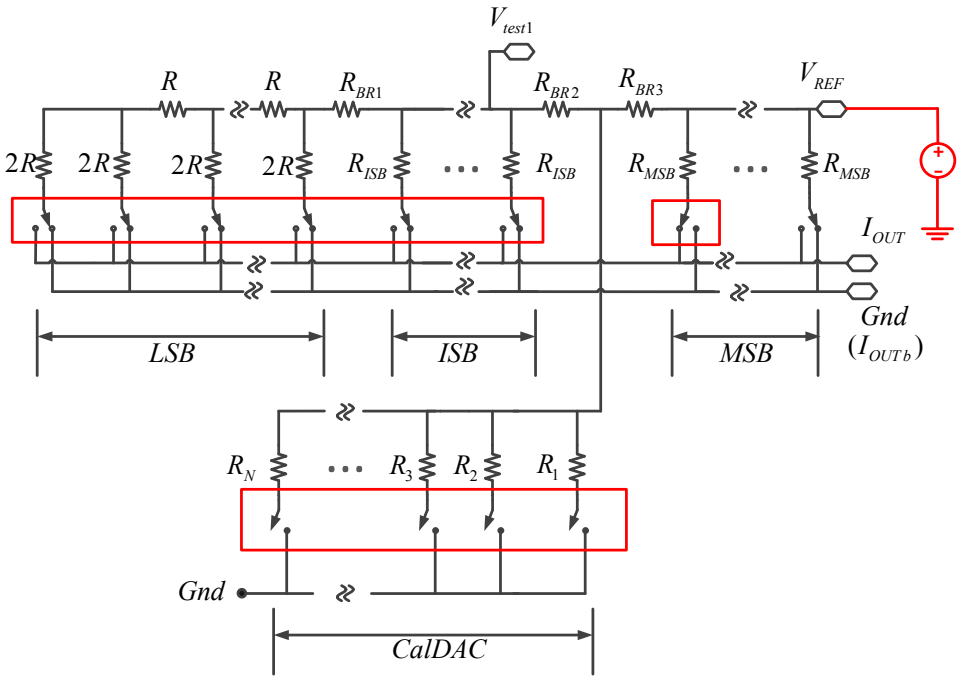
2.4 OEM Calibration

2.4.1 OEM Calibration

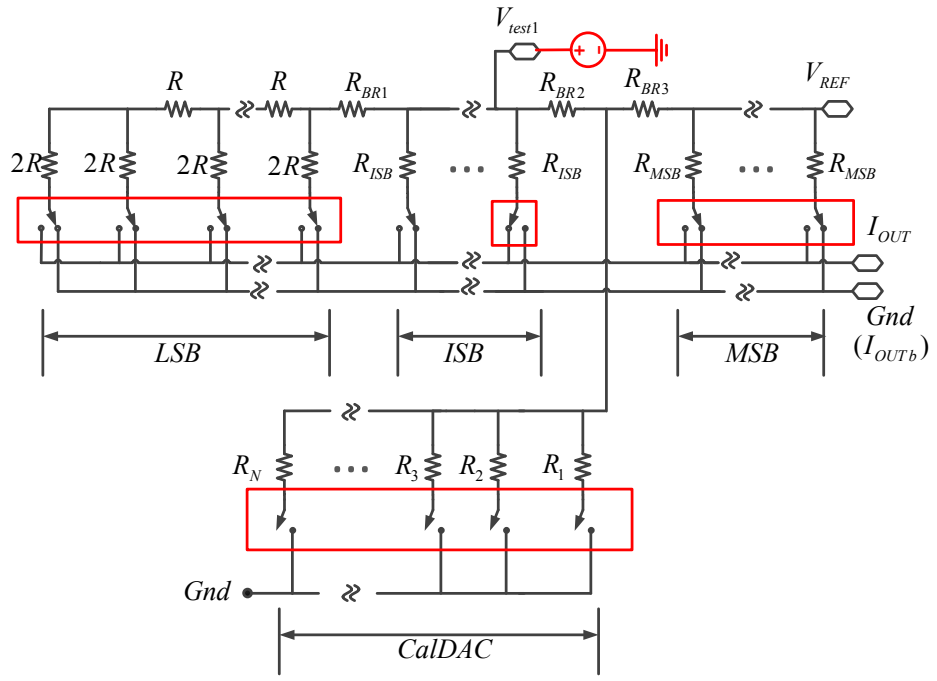
OEM calibration is applied to the resistances of the ISB and MSB segments in [28]. The calibration can be made more accurate by applying OEM calibration to the conductances of the ISB and MSB segments, because the proposed structure is that of a current output DAC. Moreover, since the CalDAC is employed to calibrate the gain error, the OEM calibration applied to the proposed DAC after fabrication is greatly simplified by following the steps shown in Fig.2.8.

Step 1 applies OEM binarization to the MSB segment. In this step, all resistors in LSB and ISB segments are switched to *Gnd* and all switches in the CalDAC are opened. Resistors in the MSB segment are sequentially switched to I_{out}, and the corresponding values of conductance $G_{MSB(i)} = 1/R_{MSB(i)}$ are obtained one by one. From the measurement results of $G_{MSB(i)}$, OEM binarization can be applied to MSB conductances.

Step 2 is similar to *Step 1* but applies the OEM binarization to the ISB segment. In this step, all resistors in LSB and MSB segments are switched to *Gnd* and all switches in the CalDAC are opened. Resistors in the ISB segment are sequentially switched to I_{out}, and the corresponding conductance values $G_{ISB(i)} = 1/R_{ISB(i)}$ are obtained one by one. OEM binarization is then applied to the ISB conductances.



(a)



(b)

Figure 2.8: OEM Calibration Process: (a) Application of OEM Binarization to the MSB Segment; (b) Application of OEM Binarization to the ISB Segment

Table 2.3: Digital Coding for OEM Binarization (7-bit)

Address Code	Number of components	Digital input
001	1	D[0]
010	2	D[1]
011	4	D[2]
100	8	D[3]
101	16	D[4]
110	32	D[5]
111	64	D[6]

2.4.2 Digital Circuits to Implement OEM Calibration

To implement OEM binarization, each resistor in a N-bit unary weighted segment needs to be connected to one of the N-bit lines. This requires an N-to-1 digital mux and a $log_2 N$ bit number of memories cells to store the mux address code for each element, similar to [26].

The calibration circuit for a 7-bit unary weighted segment is shown in Fig.2.9, where the OEM binarization is realized by the 3-bit memory cells and 7-bit muxes. The memory cells, which are arranged as serial-in parallel-out connections, can be either registers or one time programming (OTP) cells, depending on the application. TABLE 2.3 illustrates the corresponding address code and number of elements for different bit lines during the normal conversion phase. For example, if 010 is assigned to two resistors, D[1] will be selected to control them. With these arrangements, the 7-bit unary weighted segment operates in a binary weighted manner.

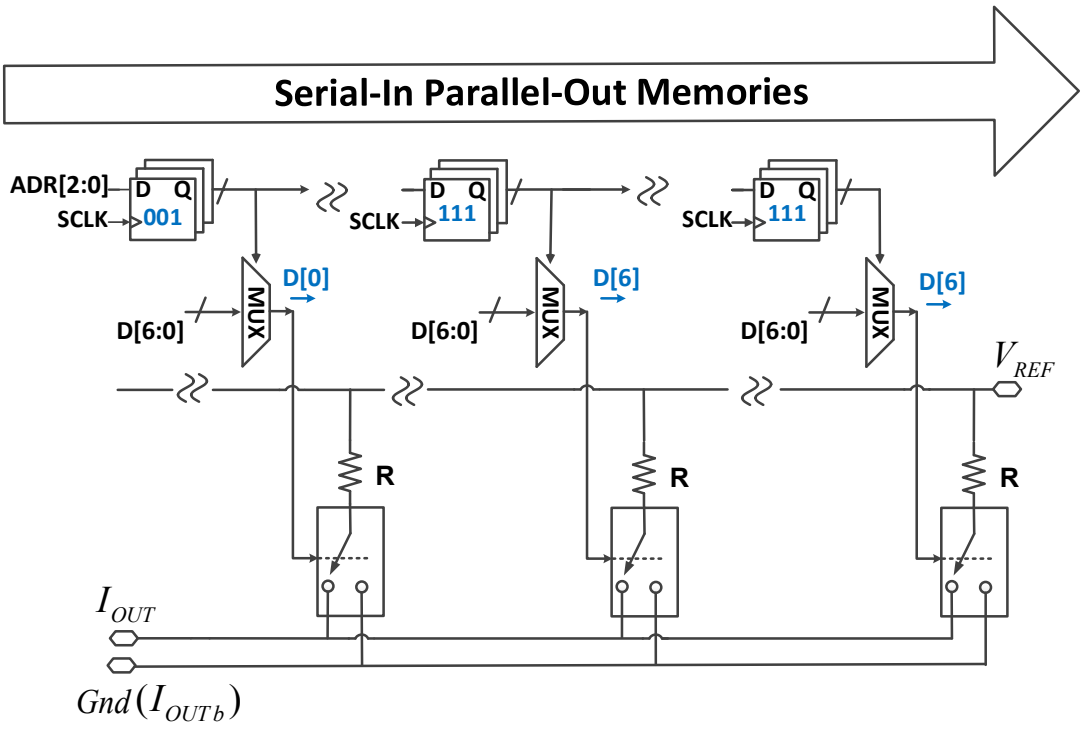


Figure 2.9: OEM Calibration Circuits

2.5 Behavioral Simulation Results

A MATLAB model of a 20-bit R-2R DAC with 8-5-7 segmentation was built on the basis of the proposed structure to verify its matching performance. We chose $R_{msb} = 2R_{mid}$, $R_{lsb} = R_{mid}$ and the same width for all resistors. To compare the model with state of the art models, the R-2R DAC model in [10] was also included, because it is the first and only one existing 20-bit R-2R DAC reported in the literature. It has a two-segment structure with 14-6 segments and could correct up to ±16 LSBs of INL by CALDAC.

First, the unit LSB in n bit level (LSB_n) was defined as

$$LSB_n = FS/2^n \tag{2.28}$$

where FS is the full scale of the DAC. If the matching level is the n bit, the standard deviation of the mismatch error $\sigma \approx 1/2^n$.

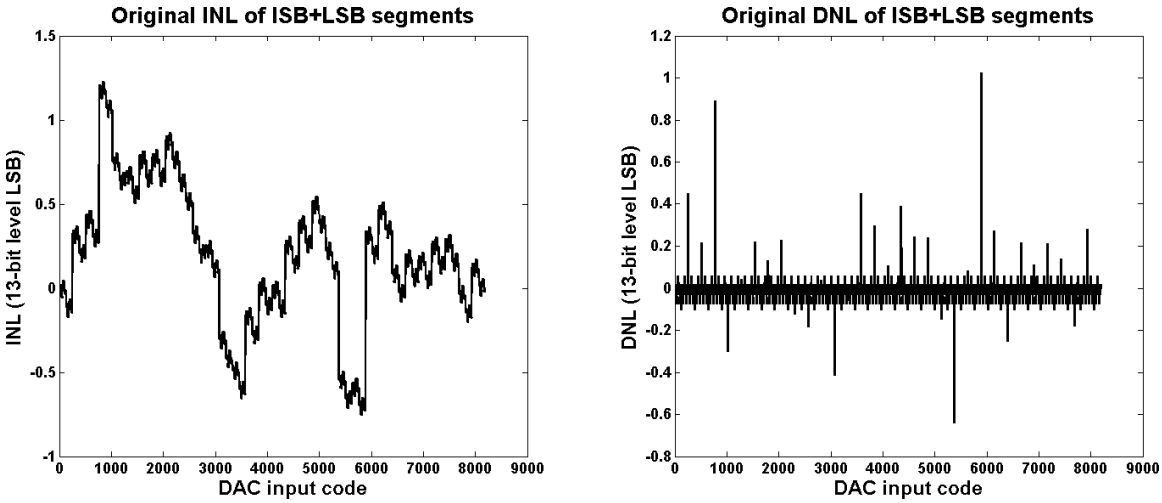


Figure 2.10: (a) INL Plot of the Original 13-bit LSB and ISB Segments; (b) DNL Plot of the Original 13-bit LSB and ISB Segments

Fig. 2.10 shows the INL and DNL of a 13 bit DAC, which consists of only the 8 bit LSB and 5 bit ISB segments. The intrinsic matching level of the unit resistors in the ISB segment is about 8 bit, and the unit $2R$ resistor in the LSB segment has about 8 bit matching accuracy. As shown in Fig. 2.10 (a), the INL of the 13 bit DAC is about 1.2 LSB and the DNL is 1.1 LSB. It should be noted the LSB used here is in the 13 bit level. After OEM binarization is applied to the ISB segment, the INL was improved to about 0.18 LSB and the DNL improved to about 0.19 LSB, as shown in Fig. 2.11.

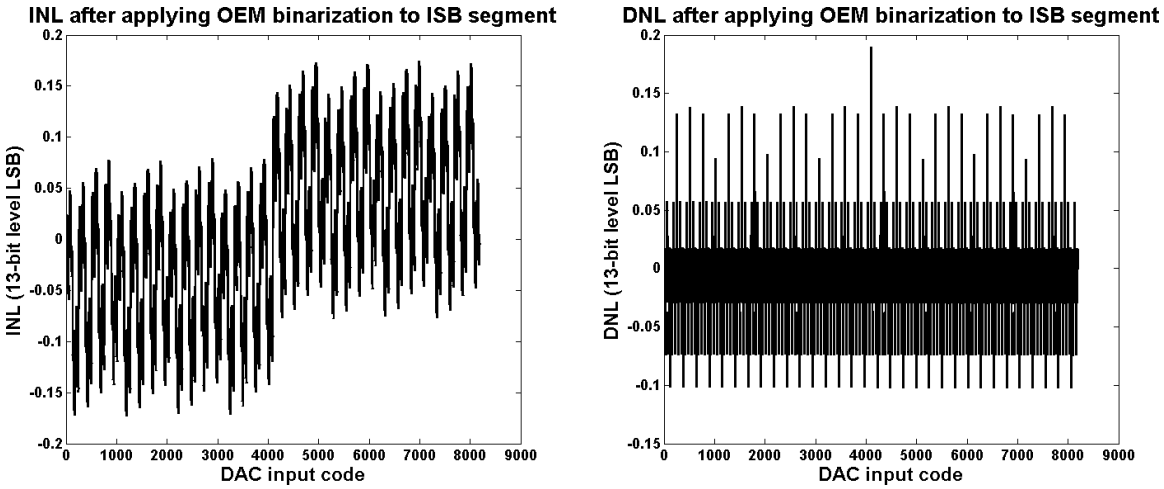


Figure 2.11: (a) INL Plot of the 13-bit LSB and ISB Segments After Applying OEM Binarization to the ISB Segment; (b) DNL Plot of the 13-bit LSB and ISB Segments After Applying OEM Binarization to the ISB Segment

To implement a 20-bit DAC, a 7-bit MSB segment was added, and the matching level of the unit resistor in MSB segment was 11 bit. If there was no gain error between the ISB and MSB segments, the linearity performance of the 20-bit DAC was dominated by the matching performance of the MSB segment. As shown in Fig. 2.12, the INL of the 20-bit DAC was about 45 LSB and DNL was 7 LSB. After applying OEM binarization to the MSB segment, the INL decreased to a 0.45 LSB and the DNL decreased to 0.46 LSB, as shown in Fig. 2.13. Results showed OEM binarization significantly reduced the mismatch error and improved linearity performance.

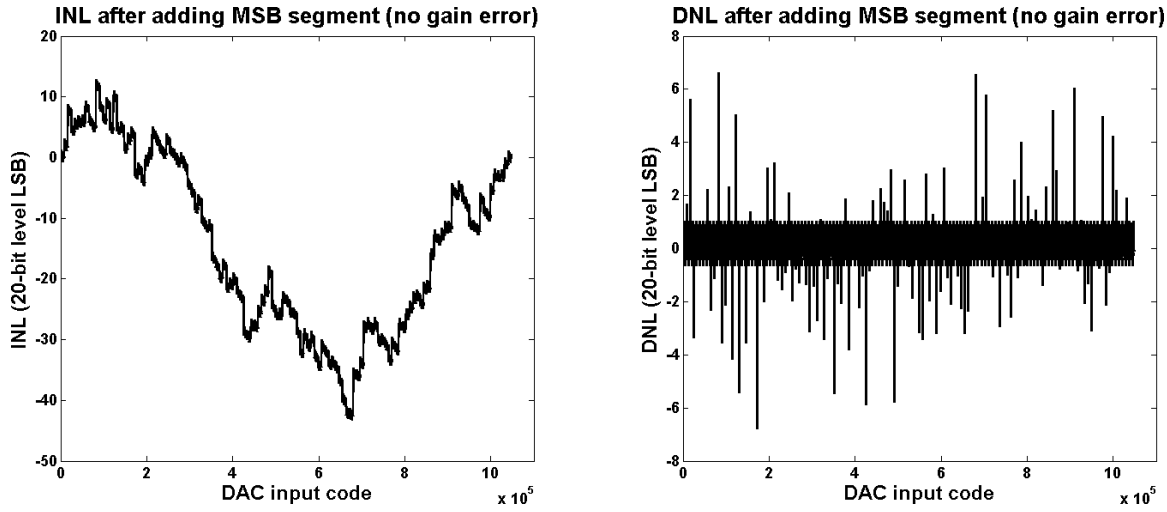


Figure 2.12: (a) INL Plot of the 20-bit DAC After Adding the MSB Segment Without Gain Error; (b) DNL Plot of the 20-bit DAC After Adding the MSB Segment Without Gain Error

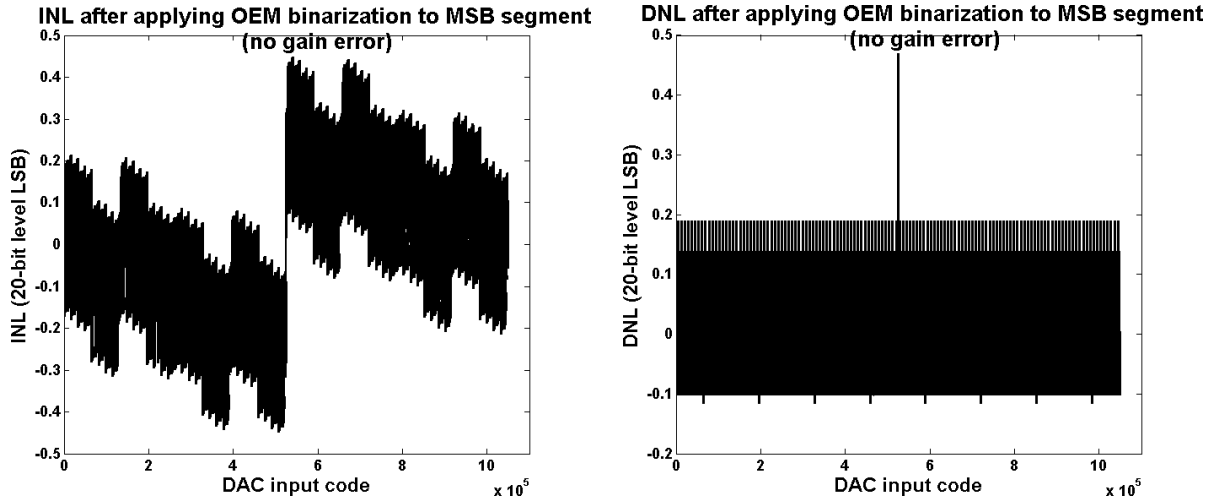


Figure 2.13: (a) INL Plot of the 20-bit DAC After Applying OEM Binarization to the MSB Segment Without Gain Error; (b) DNL Plot of the 20-bit DAC After Applying OEM Binarization to the MSB Segment Without Gain Error

However, there is another important source of nonlinearity error: the gain error between the ISB and MSB segments. In this model, the bridge resistor R_{BR2} and R_{BR3} was matched to the 11 bit level. Without CalDAC, there were two types of gain error. The first is

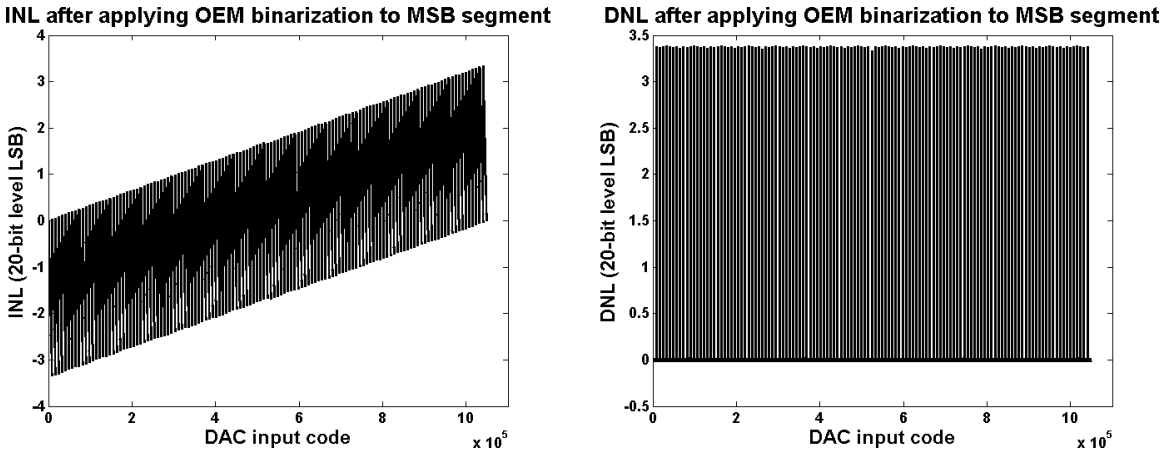


Figure 2.14: (a) INL Plot of the 20-bit DAC After Applying OEM Binarization to the MSB Segment With Positive Gain Error; (b) DNL Plot of the 20-bit DAC After Applying OEM Binarization to the MSB Segment With Positive Gain Error

$G_{ILSB}(2^{N-nmsb}) < G_{MSB}(1)$ and is called the positive gain error, as shown in Fig. 2.14. The negative gain error is that with $G_{ILSB}(2^{N-nmsb}) > G_{MSB}(1)$. If positive gain error existed, the INL and DNL plot after OEM binarization is applied to the MSB segment is shown in Fig. 2.14, and results for a negative gain error case are shown in Fig. 2.15. In both cases, the gain error dominated the INL and DNL errors. A 9-bit CalDAC was modeled, and after the gain calibration is applied to the gain error shown in Fig. 2.15, the INL and DNL plots are as shown in Fig. 2.16. After gain calibration, the INL dropped from about 14.9 LSB to 0.22 LSB and the DNL was reduced from 12.9 LSB to 0.21 LSB.

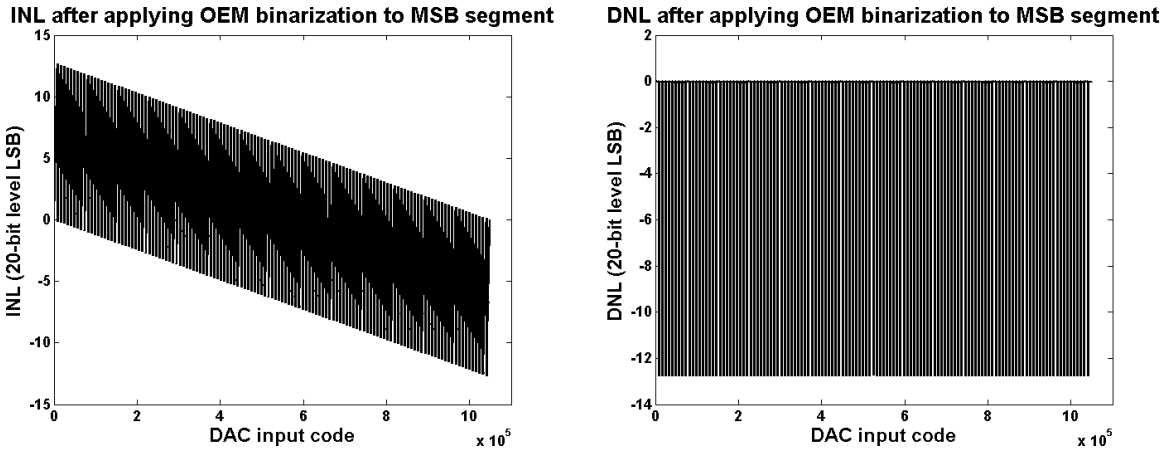


Figure 2.15: (a) INL Plot of the 20-bit DAC After Applying OEM Binarization to the MSB Segment With Negative Gain Error; (b) DNL Plot of the 20-bit DAC After Applying OEM Binarization to the MSB Segment With Negative Gain Error

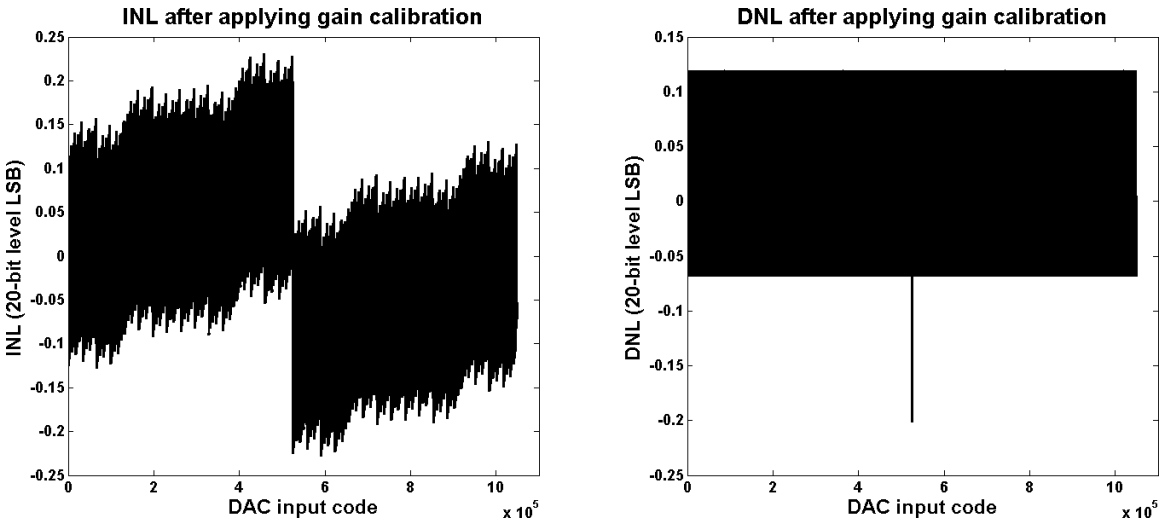


Figure 2.16: (a) INL Plot of the 20-bit DAC After Gain Calibration; (b) DNL Plot of the 20-bit DAC After Gain Calibration

Fig.2.17 shows DNL and INL distributions of 1,000 randomly generated 20-bit DACs with the proposed structure, with a standard deviation $\sigma_{R_{msb}} = 4.7E-4$. After OEM operation, the DNL was reduced from 43.3 LSB to 0.75 LSB and INL decreased from 69.9 LSB to 0.59 LSB! Results showed that the linearity performance improvement of the proposed structure

is much higher than that of [10].

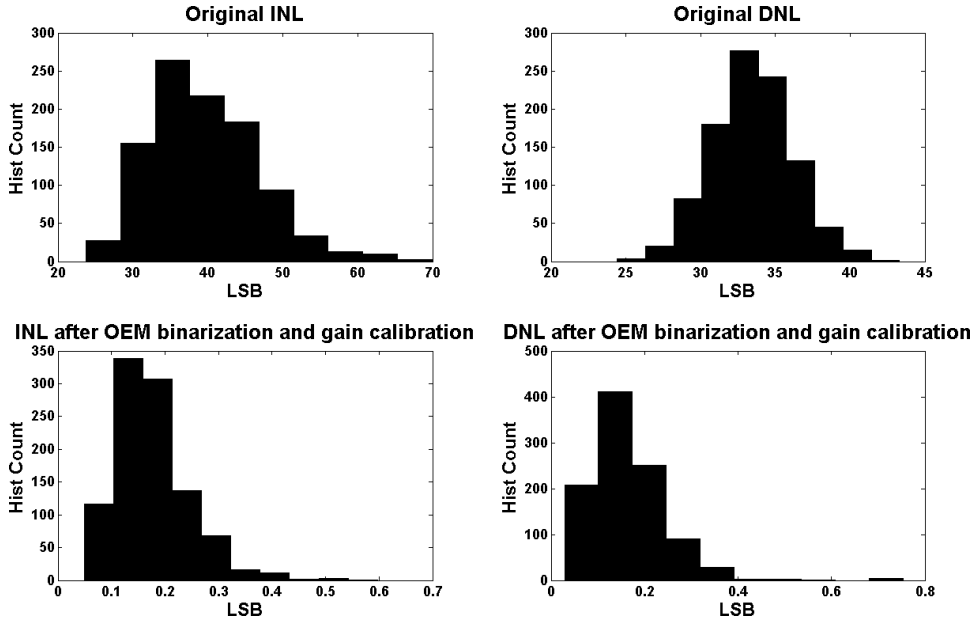


Figure 2.17: DNL and INL Distribution Comparison of 1,000 Randomly Generated Resistor Arrays in a 20-bit R-2R DAC With $\sigma_{R_{msb}} = 4.7e-4$

Table 2.4: Simulation Results of Area Comparison

Required Performance	DAC from [10]	Proposed DAC
$INL < 1LSB$	4.632e6	0.636e5

The resistor area requirement of two types of 20-bit R-2R DAC are compared in Table.2.4. Yield estimations by the Monte Carlo simulation showed the required resistors standard deviation of each type of DAC model. The area calculation was based on assuming $\sigma^2 = \frac{k}{Area}$ and $k = 1e-4$. To achieve yield $> 99.7\%$ with INL $< 1\ LSB$, TABLE 2.4 shows the required total resistor area of the proposed DAC is less than 1/72 of DAC calibrated by CALDAC in [10]. Thus the proposed DAC dramatically improved linearity performance and decreased the total resistor area.

2.6 Conclusion

Until now, performance of digital circuits was constantly enhanced by the scaling of the device dimensions and of the voltage supply. However, such technology advances do not benefit many analog and mixed-signal circuits, and in fact imposes higher requirements on their performance. The digital-to-analog converter (DAC) is one of the circuits for which high-accuracy requirements are increasing in precision medical, instrumentation, and test and measurement applications. Existing high-precision DACs require large die areas, high-precision analog processes, advanced resistor trimming techniques, complicated calibration circuits and additional test costs. As a result, costs are high and difficult to reduce because implementation benefits very little from the scaling of digital circuits. In this chapter, a low-cost, high-precision DAC structure based on OEM theory was proposed and its design methodology discussed. It achieves high matching accuracy by applying OEM calibration to the resistors in unary weighted segments and calibrating the gain error between different segments by calibration DAC (CalDAC). A MATLAB behavioral model was created, and the simulation results show that the proposed DAC structure can achieve the same accuracy as that of state-of-the-art model, in a much smaller resistor area. Moreover, with the scaling of digital devices, its advantage could be constantly enhanced.

CHAPTER 3. DESIGN AND MEASUREMENT OF A HIGH-PRECISION DAC IN 130 NM CMOS TECHNOLOGY

3.1 Introduction

To verify the advantages of the proposed DAC structure, we designed a high-precision DAC in the 130 nm Global Foundry (GF) CMOS fabrication process. This process features high-density 130 nm CMOS logic intended for RF and analog and mixed signal applications. However, it is not suitable for designing high-precision DACs, as it is not a high-precision analog process.

Most high-precision DACs rely on precision resistor arrays to perform data conversion tasks, so their accuracy is very sensitive to the matching performance of the resistor networks. Nevertheless, high-precision resistors are not part of the 130 nm GF process. The most accurate type of resistor in the process is the thin-film metal resistor, which requires a large resistor area and considerable space to achieve moderate matching performance (0.1%). Moreover, the thin-film metal resistor has very low sheet resistance, causing high power consumption and large switch transistor size.

Resistor trimming is often applied to achieve a high-precision DAC with improved accuracy. However, no resistor trimming techniques were available to us in this process. As discussed in last chapter, trimming techniques usually involve high expense, such as the expenses for extra layers or more die area for trim-pads, and the achieved accuracy is reduced by temperature and aging effects [14].

The high-accuracy calibration circuits usually employed in high-precision DACs decrease the mismatch errors by either digital or analog feedback signals from error measuring circuits, such as a high-resolution high-accuracy analog-to-digital converter (ADC) [16, 17] or DAC

[18, 19, 20]. However, accurate measurement and complicated feedback circuits are required, which may occupies a large silicon area using the 130 nm GF process, as it is not intended for use in high-precision analog design.

Therefore, this fabrication process is not intended for use in high-precision DAC designs. However, this process features high-density CMOS digital logic, and is a typical process that can be constantly enhanced by the scaling of the device dimensions and the voltage supply. As a result, implementation of a high-precision DAC in such a process has importance as a means of reducing the costs of high-precision DAC design.

In this chapter, a high-precision DAC based on the proposed structure is designed in the GF 130 nm process. The design process is described in detail from a behavioral model to a schematic and layout design. The test scheme and PCB board designs are also illustrated. Simulation and measurement results show that the proposed DAC structure can greatly reduce the area requirement and make it possible to implement a high-precision DAC without use of a high-precision analog process.

This chapter is organized as follows: In section 2, a high-precision DAC design in GF 130 nm process is introduced; section 3 discusses the test scheme and test board design; section 4 illustrates measurement results, and conclusions are stated in section 5.

3.2 DAC Design in GF 130 nm Process

We implemented our design using the GF 130 nm process from a behavioral model to a schematic and layout design. These will be illustrated in detail in this section.

3.2.1 Behavioral Model and Schematic Design

On the basis of the proposed DAC structure and its related design methodology, both a MATLAB model and a schematic design of a 21-bit R-2R DAC were built in GF 130 nm CMOS process.

3.2.1.1 Resistor Array Design and Segmentation Choice

The type of resistor used in the resistor array was chosen on the basis of the fabrication process. In the GF 130 nm process, the only available precision resistor is the thin-film metal resistor. Although its resistance density is low, its matching performance is at least 3 times better than that of any other resistor in this process, which means that it requires less area than other type of resistors in this process to achieve the same matching performance. As a result, the thin-film metal resistor was used in our resistor array.

After the resistor type was chosen, segmentation was decided upon, based on the design methodology discussed in the previous chapter. Since the segmentation is mainly dependent on the ratio between the resistor area and the OEM digital area, the OEM digital circuit area was estimated first. Based on the estimation, the total area using equation (2.8) can be calculated and the best segmentation choice to achieve minimum area can be obtained by using MATLAB model. The limitation of power consumption should also be taken into consideration. In our design, the target of power consumption is about 100 mW.

For this GF 130 nm process, the segmentation was chosen as $nmsb = 7, nisb = 5$ and $nlsb = 9$. The MSB resistor value was chosen as $16K\Omega$, ISB resistor was $8K\Omega$, $2R = 8K\Omega$ and $R = 4K\Omega$ in the LSB segment.

3.2.1.2 CalDAC Design

To calibrate gain error between different segments, the CalDAC was designed. As discussed in the previous chapter, the CalDAC is designed as a sub-radix-2 DAC and does not require high linearity performance, although a sufficiently large calibrating range is necessary. Thus, high resistor density is needed for the CalDAC design. The poly resistors were applied to the CalDAC since their resistance density is large.

From the segmentation decided on in the previous step, the calibration range and number of bits can be calculated by (2.22) and (2.23). The standard deviation of poly resistance was obtained by performing Monte Carlo simulations. The resistor ratio between the different bits

Table 3.1: The Calibration DAC (CalDAC) Design

Bits	0	1	2	3	4	5	6	7	8
Resistor value ($K\Omega$)	1330.7	723.6	390.9	210.0	112.3	57.8	31.7	17.2	9.7
Resistor width (μm)	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
Resistor length (μm)	13.2	14.3	15.4	16.5	17.6	18.7	19.8	11	6.6
Resistor multiplier	64	32	16	8	4	2	1	1	1
Resistor ratio	768	416	224	120	64	34	18	10	6

in CalDAC was then calculated using (2.24). The total number of calibration bits selected was 9, and the related resistors for each bit are shown in TABLE 3.1.

3.2.1.3 Switch Design

Unequal resistance of the PMOS and NMOS switch pairs causes important INL errors in the voltage mode DAC [10]; complicated force and sense techniques are required to reduce the INL caused by switch resistance. In the proposed DAC structure, pairs of NMOS switches are used to connect each resistor to either I_{out} or Gnd, since I_{out} is forced to Gnd by the output buffer. Because of this, on-resistance mismatches between the switches in the switch pairs are greatly reduced. To further reduce INL errors caused by switch resistance, switch sizes are increased to minimize its on-resistance, especially if resistance of the resistor is not large. As a general rule, in the resistor array, if one resistor with resistance equal to R and its weight is n LSB, on-resistance of the switch connecting to its resistor should be less than $R/2^n$. Moreover, in the LSB segment, since each bit has a different weight, it is important to ratio the size of the switches according to their weights, as shown in TABLE 3.2. In the lower-weight bits, different numbers of fingers can be used to ratio switches of different bits, but in the higher-weight bits, applying different multipliers together with a common centroid layout is necessary to achieve more accurate ratios between different bits.

Table 3.2: The Switch Design

Bits	LSB Bit-1	LSB Bit-2	LSB Bit-3	LSB Bit-4	LSB Bit-5	LSB Bit-6	LSB Bit-7	LSB Bit-8	LSB Bit-9	ISB Bits	MSB Bits
W(μm)	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
L(μm)	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
Number of Fingers	1	2	4	8	16	16	16	16	16	16	16
Multiplier	1	1	1	1	1	2	4	8	16	32	16

3.2.1.4 Digital Circuits Design

To implement the OEM calibration, digital circuits should be added into the DAC. As discussed in previous chapter, the series-in parallel-out memory cells can be implemented as the shift registers, as no one time programming (OTP) cells are available in the GF 130 nm process. The block diagram of the DAC is shown in Fig. 3.1. For the ISB and MSB, shift registers with mux circuits are needed, where the $ADRI$ and $ISBCLK$ are the data and clock signal for shift registers for the ISB segment. The $ISBO$ signals are output signals used to verify the functions of the shift registers. The situation is similar for the $ADRM$, $MSBCLK$ and $MSBO$ signals. The $D[0:21]$ is the input digital code and $Dc[0:8]$ is the input code for the CalDAC.

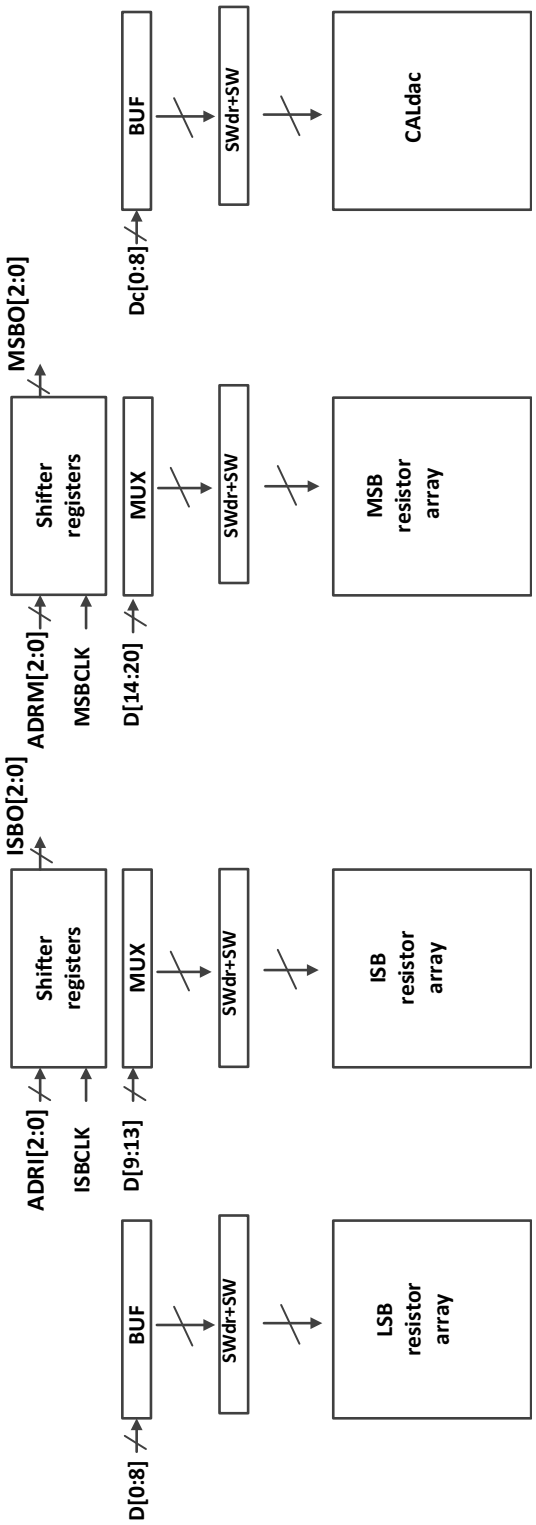


Figure 3.1: Block Diagram of the DAC

Table 3.3: MATLAB and Schematic Simulation Result

	Total INL	LSB INL	ISB INL	MSB INL	Total DNL	LSB DNL	ISB DNL	MSB DNL
MATLAB Simulation (LSB)	0.11	0.025	0.005	0.044	0.065	0.029	0.005	0.071
Schematic Simulation (LSB)	0.16	0.036	0.005	0.095	0.087	0.036	0.005	0.071

3.2.1.5 Simulation Results

To verify the proposed DAC structure, both a MATLAB model and a schematic design of a 21-bit R-2R DAC were built based on the GF 130 nm CMOS process.

In the MATLAB model, all the resistance parameters in the DAC are modeled as random variables, which include the variation of the resistor array, the on resistance of the switches and the CalDAC. The standard deviation of the MSB resistor is $\sigma_{R_{MSB}} = 5 \times 10^{-4}$. 10,000 randomly generated 21-bit DACs with the proposed structure were simulated, and the worst-case results are shown in TABLE 3.3. The total INL of the simulation results is only 0.11 LSB and the total DNL is only 0.065 LSB after OEM calibration. The INL and DNL of each segment are also shown in TABLE 3.3.

From the MATLAB simulation results, the resistance parameters in the DAC for the worst-case linearity performance can be obtained. Those parameters are applied to the schematic as variable components of the circuits. For example, the variations of the unit resistors in the MSB segment can be modeled as resistors with positive or negative resistance in series with the nominal MSB resistors. With addition of these variable components to the DAC circuits, the variations of the resistance parameters in DAC can be modeled. The schematic simulation results of INL and DNL simulation are shown in TABLE 3.3. The results are similar to those obtained with the MATLAB simulation, with total INL=0.16 LSB and total DNL = 0.087 LSB.

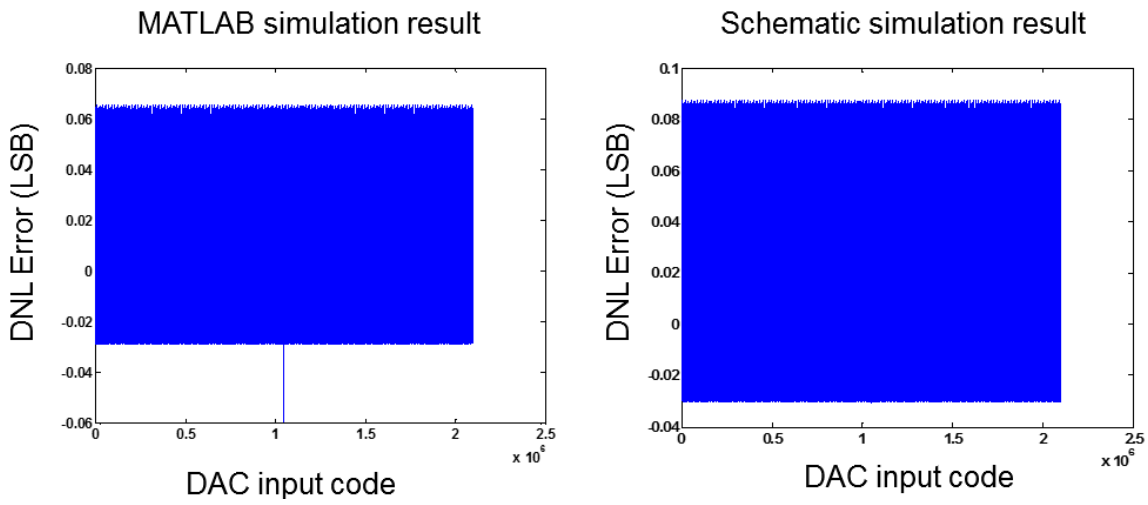


Figure 3.2: MATLAB and Schematic DNL Simulation Results

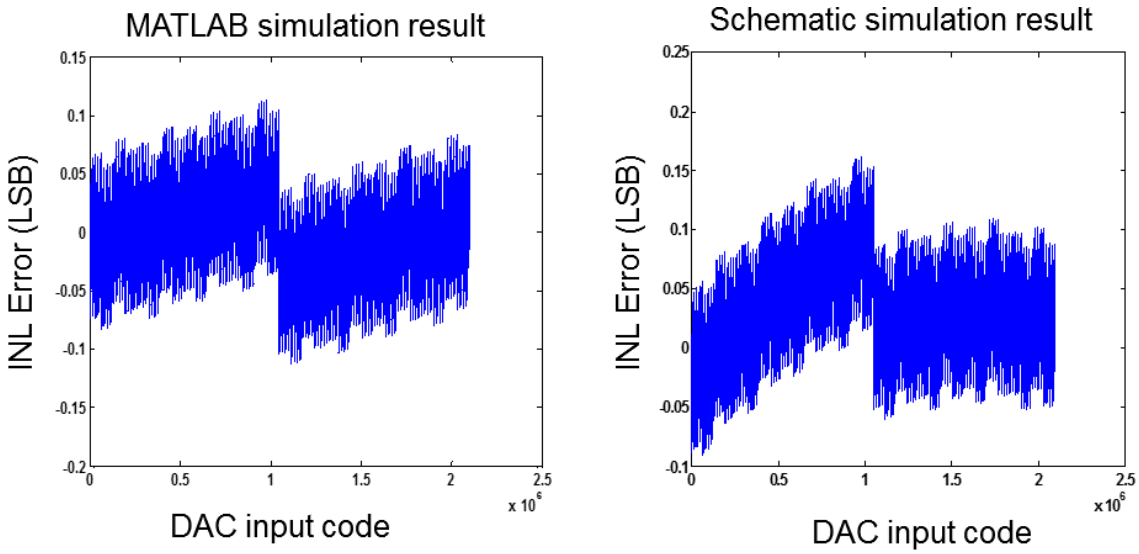


Figure 3.3: MATLAB and Schematic INL Simulation Results

The plot of INL and DNL for the MATLAB and schematic simulations are shown in Fig. 3.2 and Fig. 3.3, respectively. The results of both are similar, and the slightly difference between the two is due to the leakage and parasitic effect, which are ignored in the MATLAB simulations.

3.2.2 Layout Design

Such a large design requires a great deal of layout work, the key points of which are summarized in this section.

3.2.2.1 Top Level Layout

The top level layout is shown in Fig. 3.4; the total area included in the pads is about $2600\mu m \times 4000\mu m$. The total area is dominated by the resistor arrays, especially the MSB segment. By employing poly resistors, the CalDAC can be made to occupy a much smaller area. To implement the OEM calibration, digital circuits are integrated to the MSB and ISB segments. To balance the digital signals, digital buffer trees are also designed for the OEM digital circuits. Moreover, a large switch area is adopted to minimize on-resistances of the switches. However, the total area of the digital circuits and switches to implement the OEM calibration is still very small compared with the resistor area, as shown in Fig. 3.4.

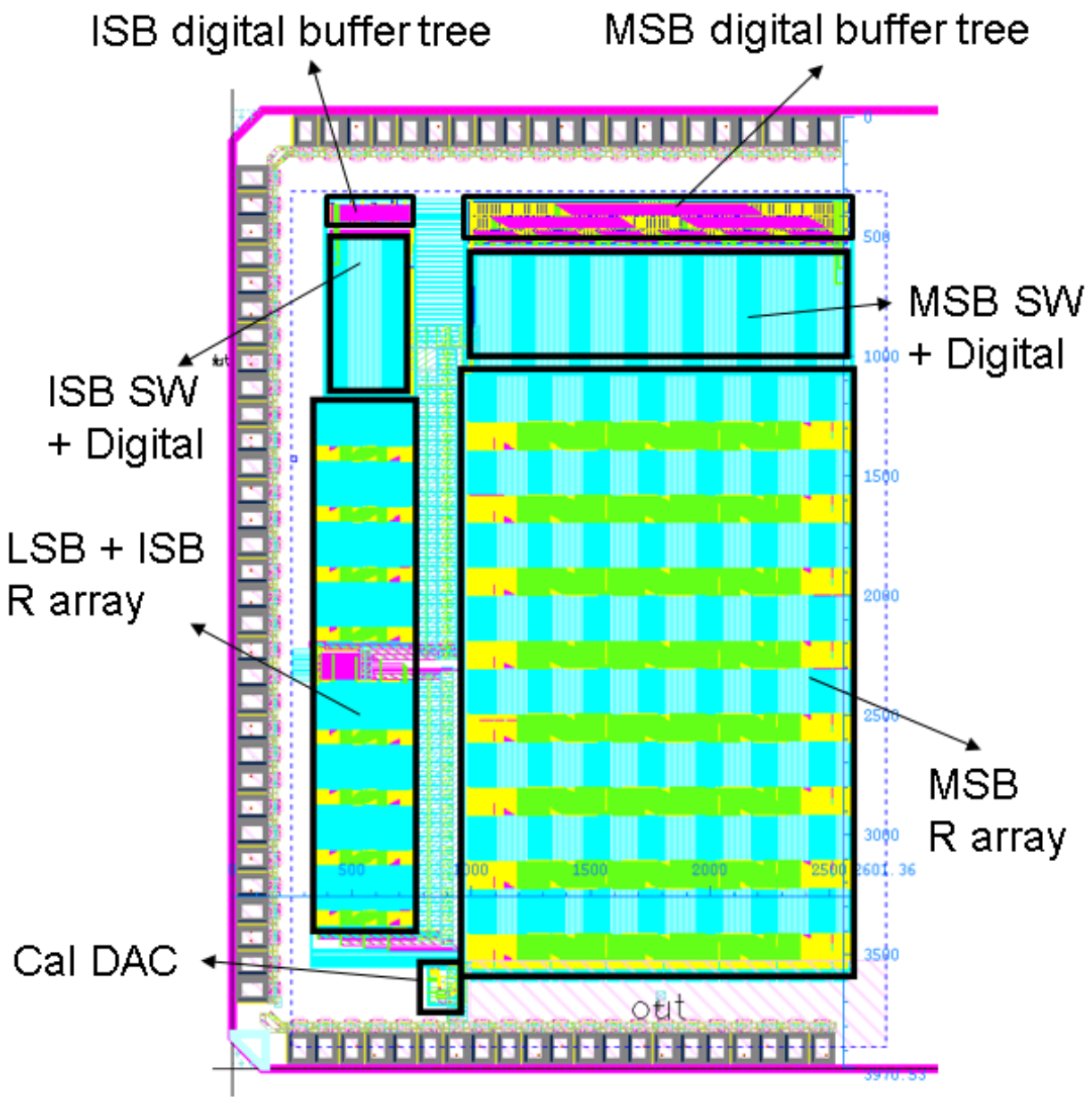


Figure 3.4: The Top Level Layout

Since our DAC design shares the same chip with other designs, only three sides of the chips are utilized for placement of the pins. The whole chip's size is $4000\mu m \times 4000\mu m$, and the die photograph is shown in Fig.3.5. The part in the blue block is our DAC circuit.

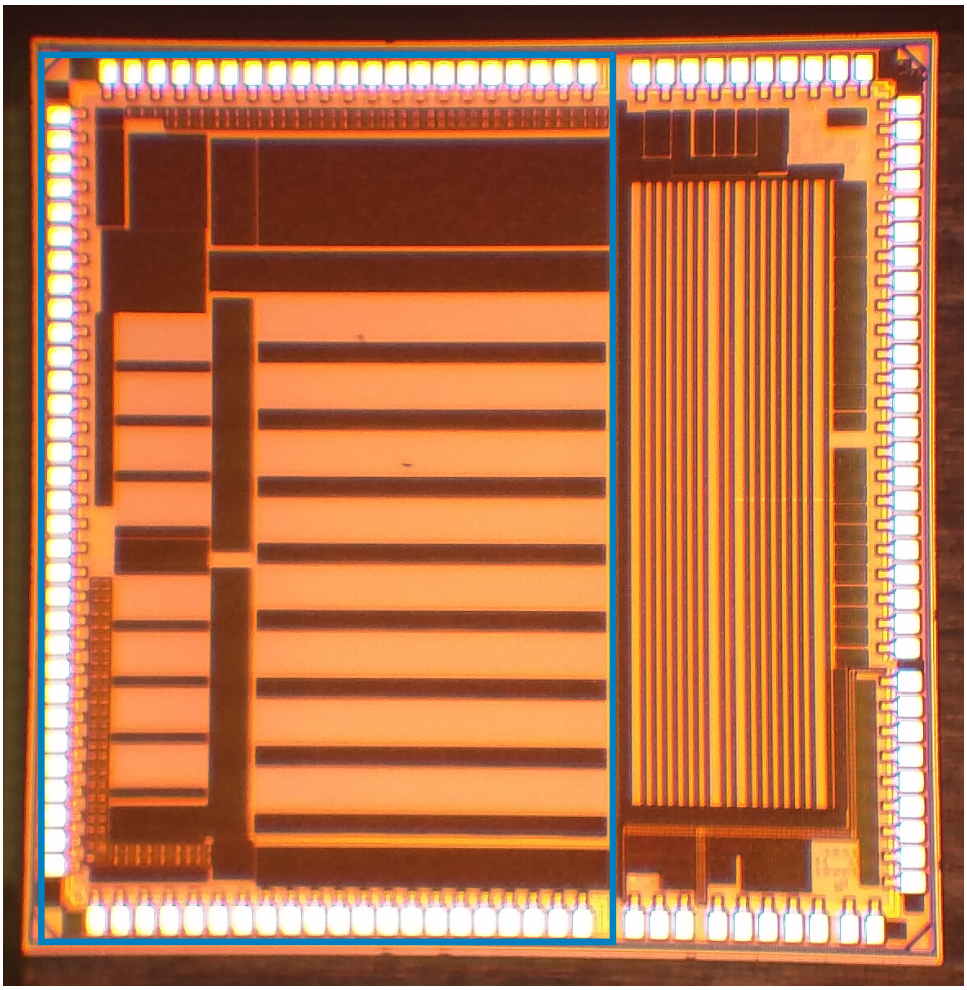


Figure 3.5: The Die Photograph of the Whole Chip

3.2.2.2 Common Centroid Layout for Resistor Array

The common centroid layout technique is often applied to minimize the gradient errors in a layout design [32]. Typically, a n_{th} order layout pattern can cancel n_{th} order gradient effects [33, 34, 35]. A typical 2nd order common centroid layout, shown in Fig. 3.6 (a), can reduce the 1st and 2nd order gradient error. Although higher-order layout patterns can reduce higher-order gradient effects, they require a more complicated wiring connection and larger die area. Considering such trade-off, a 2nd order common centroid layout is employed in our design, as shown in Fig. 3.6 (b). The R_{BR3} has the similar size of MSB resistors, so R_{BR3} is grouped with 143 resistors into a resistor array with size of 144. Those 144 resistors are separated into 8

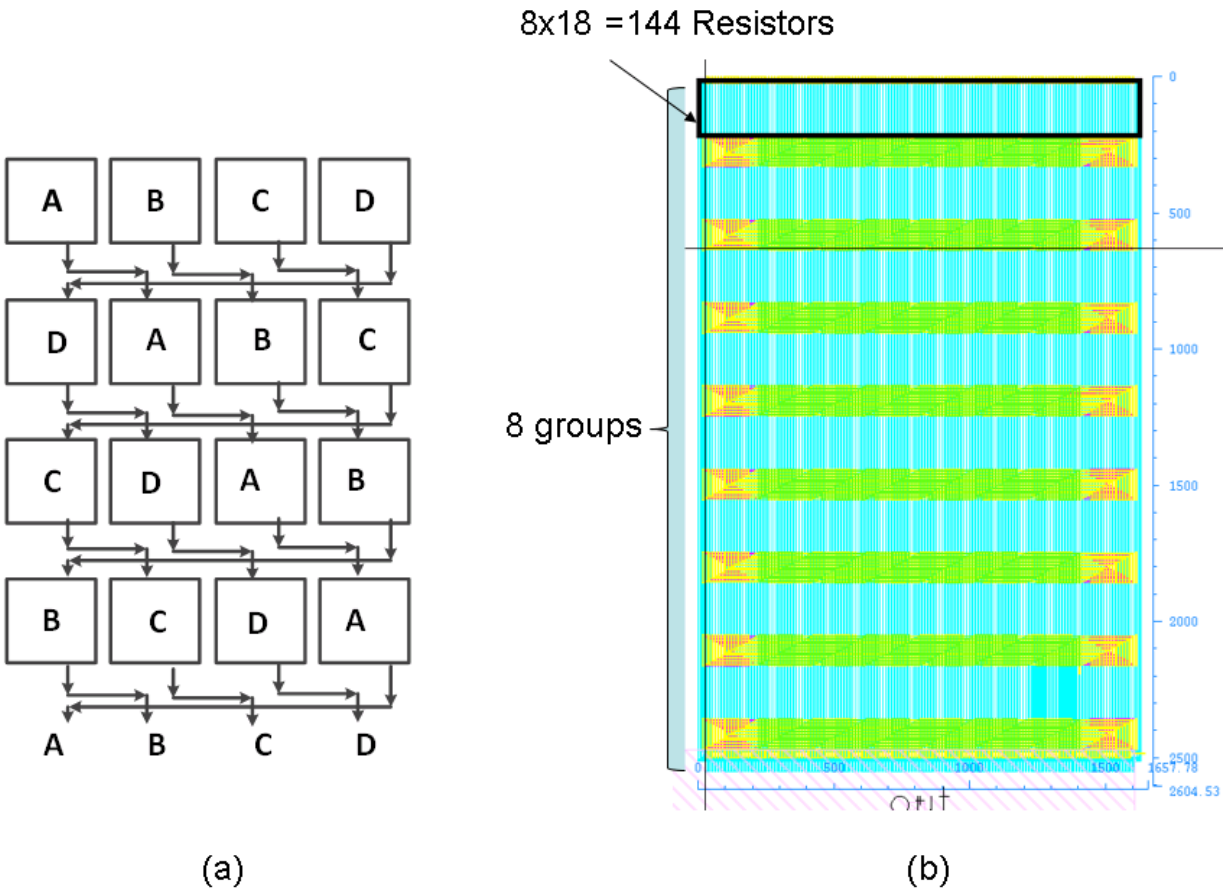


Figure 3.6: (a)2nd Order Common Centroid Layout Example; (b)Layout of the MSB Segment

sub-groups, and the 2nd order common centroid pattern is applied to them as shown in Fig.3.6 (b). The total area of the MSB segment is about $2600\mu m \times 1600\mu m$. A similar 2nd order common centroid layout is applied to the ISB and LSB segments.

3.2.2.3 Layout for Switch Pair and Digital Circuits

Mismatches in on-resistance of switch pairs is an important reason for the degradation of the linearity performance of the DAC. Since large switch pairs are employed in the MSB segment, common centroid layout techniques are also applied to the switch pair design, as shown in Fig. 3.7(a). The size of the MSB switch is $W = 1.2\mu m, L = 800nm$, $number\ of\ fingers = 16$ and $Multiplier = 16$, so each switch has 16 different sub-cells. By applying a 2nd order common centroid layout, the 1st and 2nd order gradient errors can be minimized.

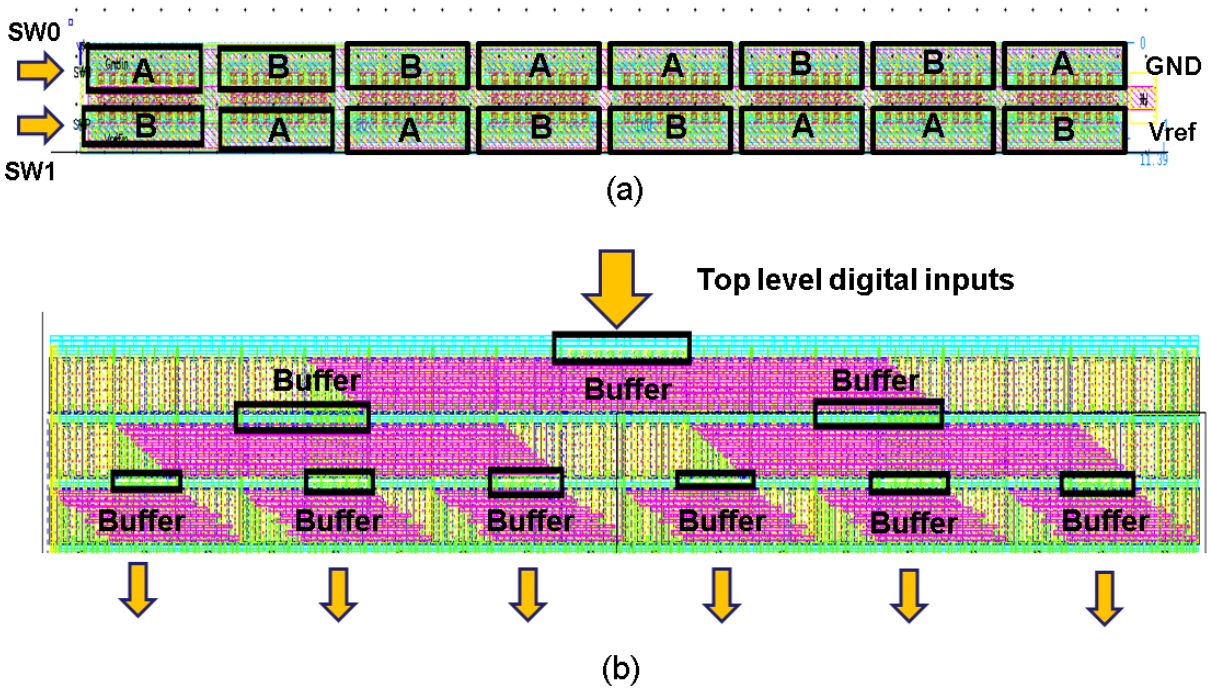


Figure 3.7: (a)Common Centroid Layout for a Switch Pair; (b)A Digital Signal Buffer Tree

To perform the OEM calibration, digital circuits must be implemented for each resistor in the unary-weighted arrays. However, the area of the MSB and ISB resistor arrays are so large that a large timing delay exists between different resistors in the same resistor array. This may cause serious timing problems, since we use the series-in and series-out method to change the data in the memory cells for each resistor. To balance the digital control signals to each resistor, a digital signal buffer tree is designed, the layout of which is shown in Fig. 3.7(b). Adding the buffer tree makes the delay from the top level input to each resistor almost the same, so the timing issue can be avoided for OEM calibration.

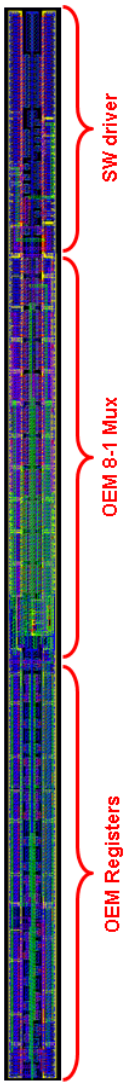


Figure 3.8: Layout of One-resistor OEM Logic

Furthermore, the width of the MSB resistor is chosen as 5.4 μm, with 5.6 μm space between resistors. To align the OEM digital circuits with the resistors, the layout of 1 bit OEM logic is 11 μm wide, as shown in Fig. 3.8. For each resistor in the MSB segment, 3-bit OEM registers, a 8-1 OEM mux and a switch driver are integrated in the one-resistor OEM logic layout. There are 143 resistors in the MSB segment, so there are 143 OEM logic and switch pairs. As mentioned previously, the buffer tree is inserted to balance the signals from the top to each resistor as shown in Fig. 3.9. Similar digital circuits and buffer trees are designed for the ISB segment.

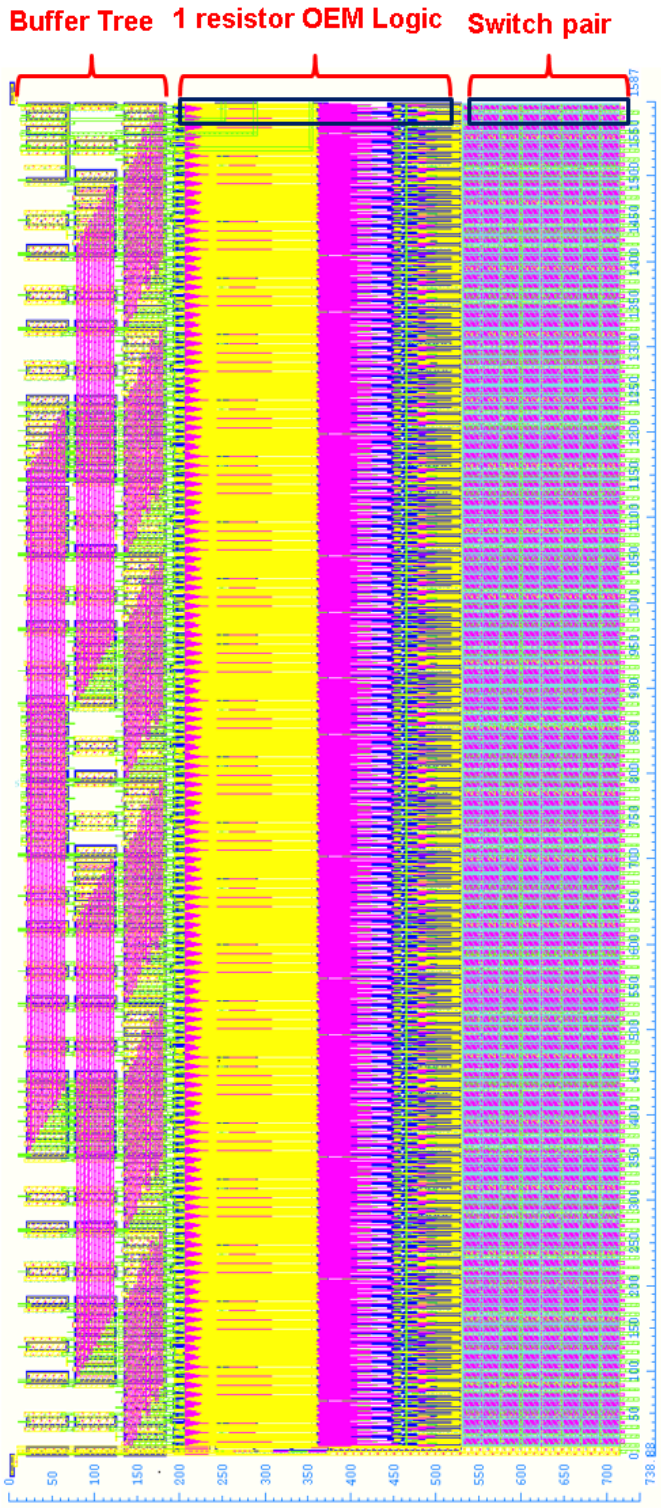


Figure 3.9: Layout of MSB Buffer Tree, Digital Circuits and Switches

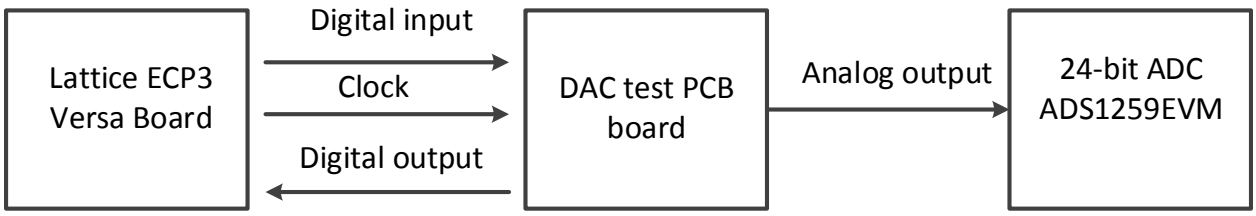


Figure 3.10: The Top Level Test Scheme

3.3 Test Scheme and Test Board Design

In this section, the test scheme and PCB board design are described in detail.

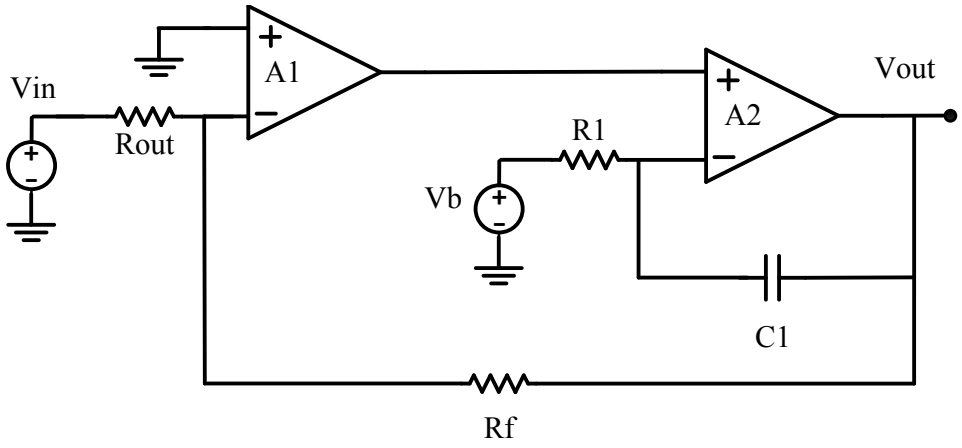
3.3.1 Test Scheme

The top level test scheme is shown in Fig.3.10. A Lattice FPGA board is used to generate the digital input and the clock signals to the DAC test PCB board. Those input signals to the DAC test PCB board are employed to measure and perform OEM calibration. From the digital output signal of the DAC, the FPGA board can verify the function of the OEM calibration digital circuits. The analog output signals of the DAC are sent to a 24-bit sigma-delta evaluation board (ADS1259EVM) for measurement, as this is the ADC with the best linearity performance we can find.

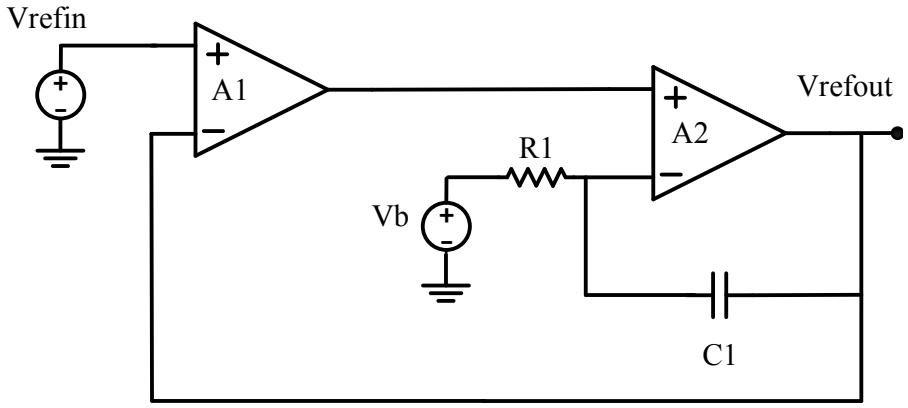
3.3.2 Test Board Design

3.3.2.1 Output Buffer Design

The output buffer should have high gain, low noise, low offset and large driving capacity for our application. Therefore, two-amplifier combined structure was employed, as shown in Fig. 3.11(a), in which A1 is ADI ADA4528 and A2 is ADI ADA4898. Their key specifications are shown in TABLE 3.4.



(a)



(b)

Figure 3.11: (a)Output Buffer Design for DAC Tests; (b)Reference Buffer Design for DAC Tests

ADA4528 has a very low offset voltage (0.3 μV(typical), 2.5 μV(max)), low noise and high gain. ADA4898 has high gain together with a large drive ability (40 mA linear output current). The output buffer could achieve a dc gain of about 257dB, PM=58 degree, and its total output voltage noise is 0.35 μV (0.1m$\sim$1KHz).

Table 3.4: Key Specifications of A1 and A2

	A1(ADA4528)	A2(ADA4898)
$V_{OS}(\mu V)$	0.3(TYP)/2.5(MAX)	20(TYP)/125(MAX)
-3dB BW(MHz)	6.5	65
Aol(dB)	139(TYP)/127(MIN)	103(TYP)/99(MIN)
I_{SC}(mA)	±40	±150
Input noise density nV/$\sqrt{Hz}$	5.6	0.9
Voltage supply range (V)	2.2∼ 5.5	10∼ 32

3.3.2.2 Reference Buffer Design

The reference buffer has a structure similar to that of the output buffer, as shown in Fig. 3.11(b). The only difference is that there is no feedback resistor Rf in the reference buffer.

3.3.2.3 Reference Design

4 LTC 6655 [36] reference circuits are paralleled together to generate the required 3.3V reference voltage. It can achieve a temperature coefficient less than 2 ppm/C.

3.3.2.4 Digital Buffer and Switch Implementation

Because digital buffers are necessary to synchronize the input digital and clock signals from the FPGA board, several digital buffer circuits are implemented in the PCB board.

As discussed in the previous chapter, applying OEM binarization to the ISB segment requires switching the reference voltage from the V_{REF} to V_{test1} node, so a switch circuit is needed in the test board. A relay circuit is used as the switching circuit in our design because of its low on-resistance.

3.3.2.5 The Whole PCB Design

The whole test PCB board is shown in Fig. 3.12. The digital signals are input from the FPGA header (HD). The chip is seated in a 128-pin chip socket and connected to the test board

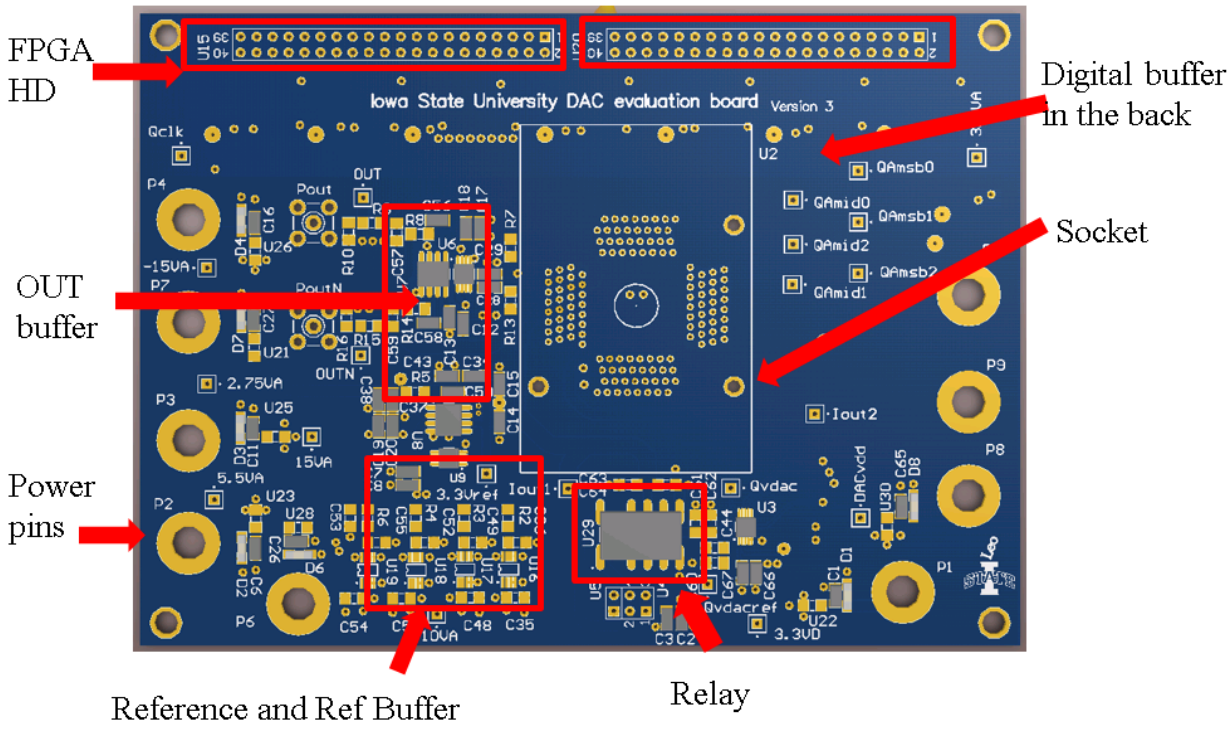


Figure 3.12: The Whole Test PCB Board

by the socket. The Output buffers are made as close as possible to the socket to minimize the parasitic resistance in the output path. The reference circuit, reference buffer and relay are all shown in Fig. 3.12.

3.4 Measurement Results

After fabrication, the DAC chip is measured by a designed test scheme, and the OEM and gain calibration are applied to it. The result is shown in Fig. 3.13 and summarized in TABLE 3.5. From the measurement results, the INL of the DAC is seen to be 1.625 LSB (LSB in 18 bit level) and the DNL is seen to be about 1.08 LSB. As a result, the linearity performance is in 17 bit level. The total area of the chip is about 10 mm^2. For comparison, based on the intrinsic matching performance of the GF 130 nm process, the required area can be calculated to be about 309.6 mm^2, as shown in TABLE 3.6; this area is almost impossible to be implemented as a single chip. Therefore, the proposed DAC structure can significantly reduce the required

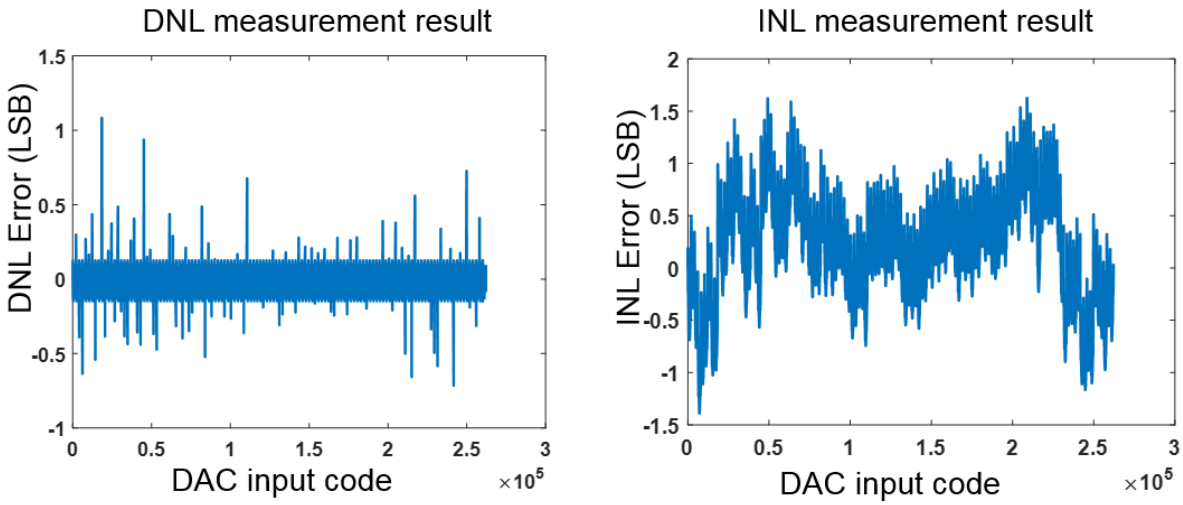


Figure 3.13: The Measurement Results

Table 3.5: Measurement Results

	Total INL	LSB INL	ISB INL	MSB INL	Total DNL	LSB DNL	ISB DNL	MSB DNL
MATLAB Simulation (LSB)	1.625	0.205	1.032	1.491	1.084	0.147	0.925	0.956

area for high-precision DAC design and implement a high-precision DAC in a low-cost process.

Compared with the schematic simulation results shown in section 2, there is a large difference of the linearity performance in the measurement result, which is due to the parasitic wire resistance and the noise effect. The first of these is the most important cause of performance degradation; wire resistance is crucial to the linearity performance of a high accuracy DAC. As shown in Fig .3.14, it is the MSB segment in the proposed DAC structure. The effect of wire resistance can be analyzed as follows: The wire resistances R_{w0} and R_{w3} connecting to I_{outb} bring no errors, as they have almost no impact on the output current. The current flowing through wire resistance (R_{w4}) connecting to V_{REF} does not change with input code, because the $I_{out} + I_{outb}$ is always constant. Thus, R_{w4} does not create nonlinearity errors with input code variations. The wire resistance R_{w2} generates nonlinear errors, but R_{w2} is independent for each branch. This kind of error can be calibrated by OEM calibration. However, wire resis-

Table 3.6: Area Comparison

Chip area (mm^2)	Estimated area to achieve 17-bit accuracy by intrinsic matching(mm^2)
10	309.6

tance R_{w1} connecting to I_{out} causes interaction of different branches and results in nonlinearity errors with changing in the input code. To illustrate its effect on the linearity performance of the R-DAC, the INL with and without the R_{w1} wire resistance is compared in Fig. 3.15, using results from schematic simulations. Accuracy of about 21 bit can be achieved without wire resistance. In our design, the resistor in the MSB segment have a large area and space, so a three-layer-parallel very wide (more than 100 μm width) wire is used to connect all resistors in the MSB segment. It can achieve $R_{w1} = 0.6\ m\Omega$, but such small wire resistance still can degrade the INL performance to less than the 18 bit level, as shown in Fig. 3.15. To achieve 20-bit INL performance, the largest tolerable wire resistance is as low as $R_{w1} = 20\ \mu\Omega$, which requires about 3 mm width if a three-layer-paralleled wire connection is used. Therefore, the required wire resistance is impossible to realize by wide wiring or multiple layers in the GF 130 nm process, and it is the dominant factor limiting the linearity performance of our proposed structure.

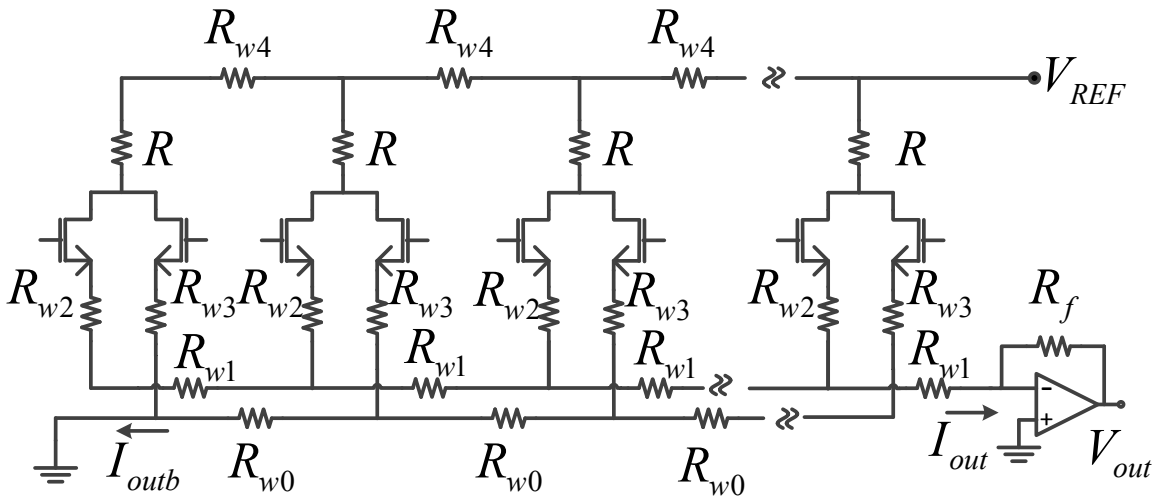


Figure 3.14: Simplified DAC Structure With Wire Resistance

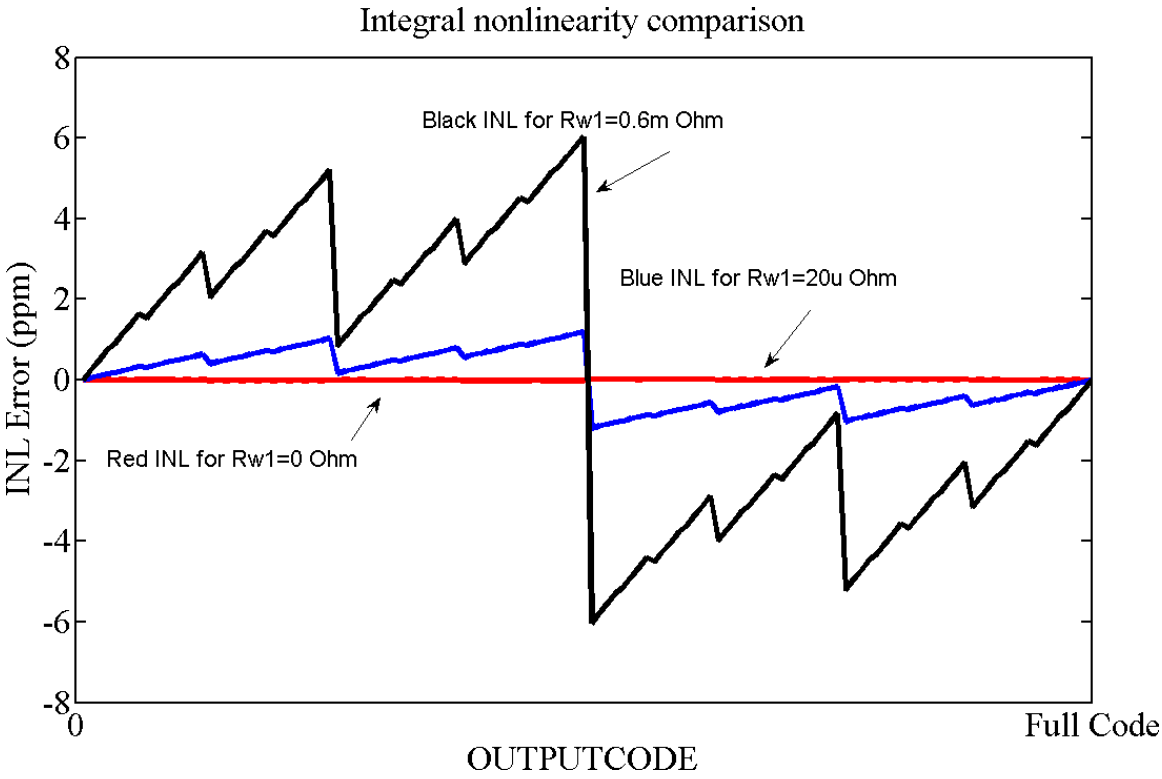


Figure 3.15: INL Performance Comparison for DAC With and Without Wire Resistance

The second important factor that limits linearity performance is noise. Although the testing ADC: ADS1259EVM is the ADC with the best linearity performance that we could find, its

Table 3.7: Noise Performance of the ADS1259EVM

Data Rate (SPS)	Noise free bits
10	21.4
50	20.4
400	19
1200	18.2

measurement accuracy is limited by its noise performance, as shown in TABLE 3.7. Other important components on the test board contribute additional noise, such as the reference circuit, reference buffer and output buffer.

3.5 Conclusion

As a design example to verify the proposed structure, a high-precision DAC was designed in a 130 nm Global Foundry (GF) CMOS process. The 130 nm GF process features high-density digital circuits but lacks high-precision resistors or any resistor trimming techniques, making it generally unsuitable for any high-precision DAC design. However, we implemented our design in such a process from behavioral model to schematic and layout design. A complicated test scheme and PCB design were also performed. The simulation and measurement results show that the proposed DAC structure can greatly reduce the area requirement and make it possible to implement a 17-bit DAC without use of a high-precision fabrication process. As a result, it is demonstrated that our proposed DAC structure can significantly lower the costs of high-precision DAC design.

CHAPTER 4. EFFICIENT VERIFICATION AGAINST UNDESIRED OPERATING POINTS FOR MOS ANALOG CIRCUITS

4.1 Introduction

Feedback is a very powerful circuit design technique that has found wide application in analog circuits. While various feedback approaches, e.g. self-biasing [5], bootstrapping [6], and digitally-assisted-analog [7] have been applied to analog circuits, these may make a system vulnerable to multiple operating points. The existence of undesired operating points is an important problem in many analog circuits such as bias generators, current/voltage references, temperature sensors, supply regulators, and frequency generators [37, 38, 39, 40]. Such analog circuits are present in virtually all non-trivial circuits and the integration of analog with mixed-signal circuits has grown rapidly with emerging IC applications. Once a circuit is locked into an undesired operating point, it may experience dramatic performance loss and system failure may result, possibly damaging system components or requiring a power recycle for unlocking. Circuits with undetected operating points can be devastating, particularly when they are used in critical systems such as automotive, health care, and military applications. As a result, identifying the presence/absence of undesired operating points is one of the most critical problems in circuit design; undesired operating points must be identified and eliminated for proper circuit operation.

However, since existing circuit simulators provide only a single operating point[8], recognizing the existence of undesired operating point largely depends on the designer's experience. The most popular technique for removing undesired operating point is to use start-up circuits, but sometimes, because of unanticipated transients and variations in operating conditions, a circuit equipped with start-up circuits may still enter into an unknown operating point. In some

circuits, even most experienced designers are not aware that a circuit they have designed has undesired operating point potential and such operating points often go undetected in standard simulations used in the design process.

Various approaches for finding operating points of circuits can be found in the literature, and they could specifically be used to detect undesired operating points. Topological methods identify multiple operating points using graphical representation and certain topological criteria [41, 42, 43, 44, 45]. However, even though topological methods may simply identify the existence of multiple operating points by examining the structure of a circuit, they never take into account device and environmental parameters, so their validity is limited by certain assumptions of the circuits.

Aside from topology methods, many researchers prefer to regard a circuit as a matrix of equations and try to seek a mathematical solution. This, however, is equivalent to solving simultaneous non-linear equations and the effort becomes formidable as the size of circuits grows. As a result, several approximation methods for obtaining circuit solution have been developed. Piecewise-linear methods can discover all operating points of a circuit based on piecewise-linear approximations of non-linear devices [46, 47, 48, 49]. Contraction methods can frame non-linear functions appearing in the mathematical description of the circuit and exploit formulas using matrix theory to contract a hyper-rectangular region that includes the solutions [50, 51]. Interval methods seek a set of solutions based on interval analysis theory [52, 53]. However, all these methods rely on simplifying device models or replacing non-linear equations by linear ones, so they are often not applicable to large-scale circuits and many practical applications.

The continuous/homotopy approach is another popular method for identifying a circuit's operating point. In general, this type of method entails embedding one or two continuation parameters into a set of non-linear equations, thereby causing a continuous deformation from a known or easily-computed solution to the required solution. In [54], Goldgeisser and Green presented an algorithm that realizes a continuous deformation ranging from a short circuit to an open circuit by adding parametrically-represented resistors at each transistor. Roychowdhury and Melville demonstrated that certain common circuit structures can lead to extreme

inefficiency or failure and developed a new homotopy MOS model featuring decoupled continuation parameters in [55]. An efficient deflation technique was developed by Tadeusiewicz and Halgas in [56] and combined with the homotopy method to find multiple DC operating points of CMOS circuits. In [57], Inoue, et al., proposed a Newton homotopy method that chooses one of the node voltages in each positive feedback loop (PFL) as variable and implemented this method using SPICE transient analysis. These methods could trace DC solutions, but they could not guarantee that all operating points would be identified; their implementation is also tedious.

Because a positive feedback structure is the cause of multiple operating points in the circuit [42], break-loop homotopy methods were proposed to find multiple operating points. This often involves introduction of a voltage or current source that can be swept to trace operating points of a circuit [58, 6], but it relies on linear ramp-up of inserted sources and generally has very low efficiency, and achieving an accurate result can require very long simulation time because a small sweeping step size is needed. It also assumes that the PFL are already known and does not provide a systematic approach for how to break the PFLs. In [59], Premoli, et al., proposed an approach that identifies the loops in circuits by topology partitioning and adds independent voltage sources to open the loop. Even though this approach does not guarantee that all of the solutions will be found, it is numerically efficient and able to deal with medium-size circuits. However, it doesn't present a method to automatically identify the feedback loops in circuits and distinguish between PFLs and negative feedback loops (NFLs).

In summary, all these methods attempt to find all the operating points of a circuit, a very complicated problem. Identification of all operating points can be very difficult even for simple circuits, e.g., two-transistor circuits [60, 61, 62]. Therefore, simple and efficient verification methods that are guaranteed to find all operating points in even rather simple circuits do not exist.

Recently a divide and contraction verification method for discovering the existence of undesired operating points was proposed in [63]. Contrary to traditional methods used to deal with discovery of operating points, this method does not try to find all operating points or even any operating point at all. It only targets finding voltage intervals that contain undesired operat-

ing points and thereby verifying existence of undesired operating points. If such an interval is detected, it proves that there is at least one undesired operating point in this circuit and notifies the designer to deal with it. This essentially replaces a difficult root-finding problem with simply identifying the existence of test intervals that include undesired operating points. For this reason, the method is efficient and can dramatically reduce computational requirements to determine the presence/absence of undesired operating point. However, implementation of this method is only briefly mentioned in the paper and its resultant algorithm can only apply to MOS circuits having a node where all the PFLs can be broken without cutting off any NFL.

In this chapter, the divide and contraction verification method to identify the existence of undesired operating points is systematically proposed and its application is extended to more general MOS analog circuits. The method is based on an approach that automatically converts a circuit netlist to a Directed Dependency Graph (DDG) for MOS circuits. From the definition of signs of dependencies for MOSFETs, all PFLs and NFLs are found and distinguished. The PFL breaking method and selection of breaking node that determines whether a monotonic return function of PFLs can be obtained are also discussed. Based on the monotonicity of the return function, two types of divide and contraction algorithms are proposed to efficiently search voltage intervals containing operating points. The proposed method has been implemented in standard circuit simulators such as Spectre and Ultrasim, and simulation results show that this method is effective and efficient in identifying undesired operating points in a set of commonly used benchmark circuits, including bias generators, current/voltage references, temperature sensors and op-amps [64].

This chapter is organized as follows. Section 2 introduces the systematic method for finding feedback loops and determining their signs; section 3 discusses the break-loop homotopy method and the return function characteristics; section 4 and 5 illustrate the proposed divide and contraction algorithms for monotonic and non-monotonic return functions respectively; simulation results are provided in section 6. Discussion is given in section 7, and conclusions are stated in section 8.

4.2 Feedback Loop Finding and Sign Determination

In this section, a systematic method that automatically identifies feedback loops and determines their signs is introduced. The approach to automatically convert a circuit netlist into a DDG is studied, and all the PFLs and NFLs in the circuit are determined based on the definition of signs of dependencies for MOSFETs.

4.2.1 Convert Circuit Netlist to Graph

A systematic approach to automatically convert the MOS circuit into a DDG was proposed in [65]. For a MOSFET, the drain-source is considered as a channel that conducts current flowing in a branch from Vdd to gnd (which is defined as a "branch-current" in [65]). The voltage of any net that controls the gate-source voltage of any MOSFET is defined as a "controlling voltage". Following that, dependencies between the controlling voltages and branch-currents are identified. For example, there is a dependency from gate voltage to the branch-current that flows through the drain-source channel of a MOSFET. On the other hand, the branch-current will affect the drain or source voltage (if drain or source node is not connected to a power supply or an independent source) that creates a current to voltage dependency. As a result, the controlling voltages and branch-currents form vertices of the DDG and the dependencies between them form edges of the DDG.

For example, in the inverse Widlar Bias Generator (Inv-Widlar) shown in Fig. 4.1(a), two independent branches are identified; the left branch can be written as V_{DD}, $M_4(V_{DD}, V_1)$, $M_1(V_1, gnd)$, gnd and the right branch can be written as V_{DD}, $M_5(V_{DD}, V_2)$, $M_2(V_2, V_3)$, $M_3(V_3, gnd)$, gnd. The current flowing in these two branches are labeled I_1 and I_2, respectively. V_2 is a controlling voltage from M_1 which controls the current I_1 and we thus obtain one voltage to current dependency: $V_2 \rightarrow I_1$. Similarly, V_1 is a controlling voltage from M_5 which controls the current I_2 and we obtain another voltage to current dependency: $V_1 \rightarrow I_2$. Note that diode-connected transistors such as M_2, M_3 and M_4 are treated as resistors. Finally, two current to voltage dependencies are found: $I_1 \rightarrow V_1$ and $I_2 \rightarrow V_2$. The DDG of this circuit is shown as Fig. 4.1(b).

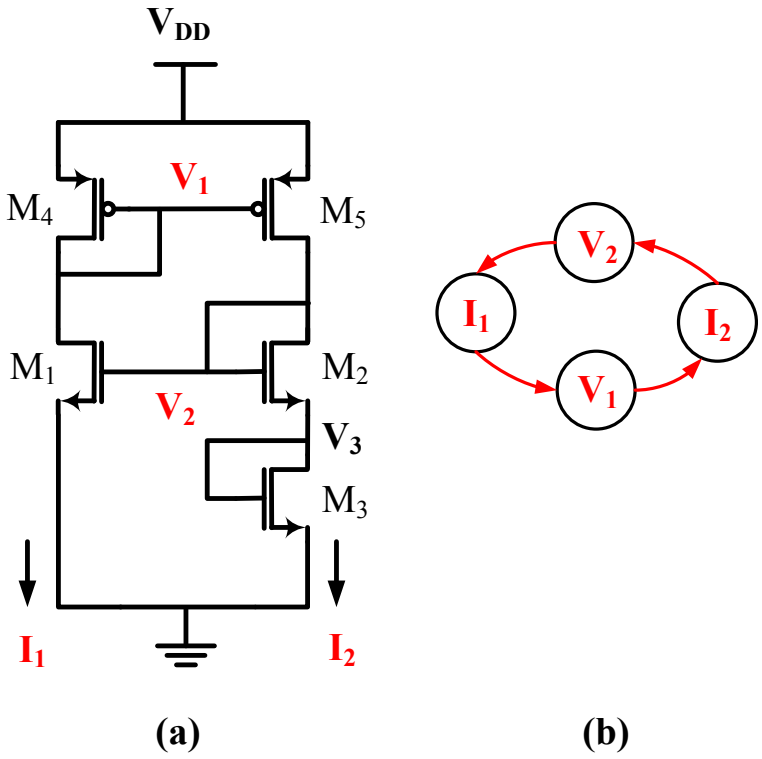


Figure 4.1: (a) Inv-Widlar Circuit; (b) Circuit Graph for Inv-Widlar Circuit

Table 4.1: The Signs of Different Dependencies for MOSFETs

Voltage to current dependency	Sign	Current to voltage dependency	Sign
$V_{G,PMOS}$[a]$\rightarrow I_{SD}$	$-$	$I_{SD} \rightarrow V_{S,PMOS}$	$-$
$V_{G,NMOS} \rightarrow I_{DS}$	$+$	$I_{DS} \rightarrow V_{S,NMOS}$	$+$
$V_{S,PMOS} \rightarrow I_{SD}$	$+$	$I_{SD} \rightarrow V_{D,PMOS}$	$+$
$V_{S,NMOS} \rightarrow I_{DS}$	$-$	$I_{DS} \rightarrow V_{D,NMOS}$	$-$

[a] $V_{G,PMOS}$ stands for the gate voltage of PMOS; S stands for source and D stands for drain

4.2.2 Determine the Signs of Feedback Loops

A feedback loop is composed of alternating $I \rightarrow V$,$V \rightarrow I$ dependencies with any vertex being visited at most once and the initial and terminal vertices are the same [66]. Thus, when determining the sign of a feedback loop, we choose one arbitrary vertex in it as the starting

point and follow the dependency direction to assign a sign for each edge. Each $I \rightarrow V$ or $V \rightarrow I$ dependency is determined by the static I/V characteristics of the related devices in the loop. Given a dependency $Y \rightarrow X$, its sign is defined to be the sign of $\frac{\partial Y}{\partial X}$. The signs of possible dependencies for a MOSFET are summarized in TABLE 4.1.

Taking a NMOS as an example, we could analyze the sign of dependencies between its terminals and drain-source current. Assuming other input variables constant, the increase of gate voltage gives rise to the growth of drain-source current for a NMOS, i.e., $\frac{\partial I_{DS}}{\partial V_{G,NMOS}} > 0$ that establishes a positive dependency for $V_{G,NMOS} \rightarrow I_{DS}$. Likewise, the negative dependency of $V_{S,NMOS} \rightarrow I_{DS}$ is due to $\frac{\partial I_{DS}}{\partial V_{S,NMOS}} < 0$. As far as the signs of dependency $I_{DS} \rightarrow V_{D,NMOS}$ and $I_{DS} \rightarrow V_{S,NMOS}$ are concerned, they could be analyzed as follows.

Since any vertex is visited at most once in each feedback loop, there is only one MOSFET in each involved branch, whose drain-source current is strongly regulated by a controlling voltage generated in other branches. Such a MOSFET is called a "controlling MOSFET" that adjusts a branch-current in a feedback loop. Therefore, only controlling MOSFETs should be included to decide the signs of dependencies in a feedback loop, and they can be regarded as current sources regulated by controlling voltages generated in other branches. As an example, a NMOS controlling MOSFET is shown in Fig. 4.2. Other devices in the same branch are defined as "resistive devices", referring to two-terminal devices that have the following relationship,

$$\frac{\partial V}{\partial I} > 0 \tag{4.1}$$

where V is the voltage across the device and I is the current flowing through it. In addition, V and I have the same signs as shown in Fig. 4.2, so the voltage across the resistive devices increases with the rise of flowing through current.

For a resistor or a diode-connected transistor (treated as a resistor), the relationship $\frac{\partial V}{\partial I} > 0$ is valid, so they are resistive devices. Another type of resistive device is the drain-source channels of MOSFETs other than the controlling MOSFET in a branch. Their gate voltages are constant because the feedback loop only alters their drain and source voltages by changing the branch-current. So they can be regarded as two-terminal components with relationships

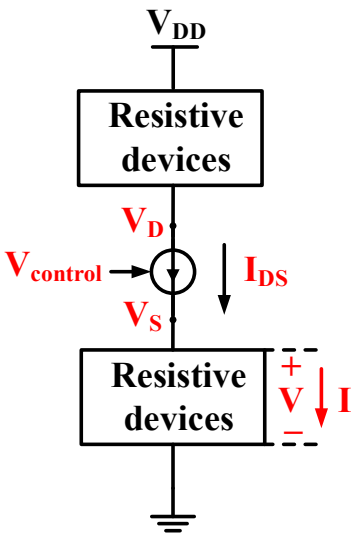


Figure 4.2: Analyzing the Sign of Dependency for Drain-source Current to Source/Drain Voltage in a NMOS

given by $\frac{\partial V_{DS}}{\partial I_{DS}} > 0$ for a NMOS and $\frac{\partial V_{SD}}{\partial I_{SD}} > 0$ for a PMOS. Thus a drain-source MOSFET channel other than that of the controlling MOSFET is also a resistive device. As far as an ordinary diode is concerned, it can be proven that its $\frac{\partial V}{\partial I} > 0$ so it is also a resistive device. For most analog circuits, only these three types of devices are involved in the DC analysis and all of them are resistive devices. As a result, with the rise of current I_{DS}, the voltage across the resistive devices between the source and *gnd* increases, i.e., $\frac{\partial V_{S,NMOS}}{\partial I_{DS}} > 0$. Thus, the sign of the dependency $V_{S,NMOS} \rightarrow I_{DS}$ is positive. Similarly a negative dependency for $I_{DS} \rightarrow V_{D,NMOS}$ is found since V_D is equal to V_{DD} minus the voltage across certain resistive devices. The signs of dependencies for a PMOS can be acquired in a similar way. Now a PFL can be defined as a feedback loop that contains a positive even number of negative dependencies, and a NFL is defined as a feedback loop that contains an odd number of negative dependencies [65]. As an example, by applying the signs of dependencies to the DDG of the Inv-widlar shown in Fig. 4.1(b), it can be seen that there is one PFL $I_1 \rightarrow V_1 \rightarrow I_2 \rightarrow V_2$ in the Inv-widlar circuit.

It's worth pointing out that the sign of a feedback loop does not change with different operating points, assuming that the drain/source of any MOSFET in the loop are well-defined. If the drain and source nodes do not exchange, the signs of dependencies involved in each MOSFET will not be altered with variation of operating points. Such design criteria is met for

most existing well-designed analog circuits and should be checked by designers from the circuit topology. In our method, the drain/source is determined by following a branch from V_{DD} to *gnd*, e.g., for a NMOS, the terminal that is closer to V_{DD} is the drain and the other side is the source.

4.3 PFLs Breaking and Return Function

After all the feedback loops and their signs are identified in the circuit, a method for breaking all the PFLs must be found. In this section, methods to break all the PFLs is discussed and the return function of a PFL is defined. The breaking method and selection of breaking node to generate monotonic return function will then be studied. After that, several important definitions and theorems regarding the return function are advanced.

4.3.1 Break PFLs and Obtain a Return Function

After all PFLs are identified in the circuit, they can be broken by different circuit-level homotopy methods. These methods often involve introduction of a testing source that can be swept to trace a returning signal. A valid operating point of a circuit is identified when the returning signal satisfies certain conditions. They can be categorized into four different types according to the voltage/current testing source and returning current or voltage signals[67]. If a current source is included for sweeping, determination of the range over which the current should be swept may require considerable effort, so voltage test sources are generally more preferable. Moreover, the relationship between a voltage testing source and a returning current signal is usually unknown as they can be found by solving complex I-V equations. However, if a certain breaking method and break node selection are used, there is a monotonic relationship between a voltage testing signal and a returning voltage signal, as will be illustrated in the following section.

An example of breaking the PFLs in a MOS circuit is shown in Fig. 4.3, in which V_2 in the inverse Widlar Bias Generator (Inv-Widlar) broken into V_2' and V_2''. Since the break is made at the gate of a MOS device where the input impedance is ideally infinite, loop loading is not affected by this break. The loop is then driven by a voltage source (V_{IN}) inserted at

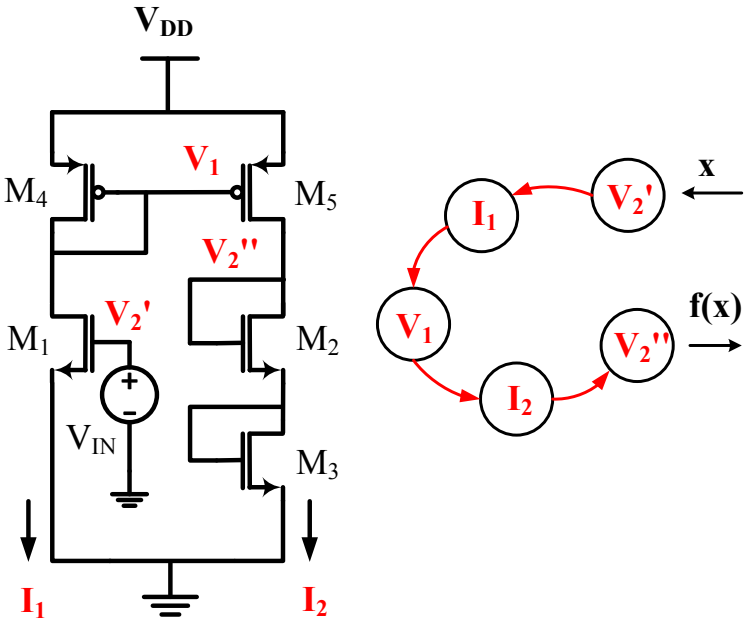


Figure 4.3: Circuit Graph for Inv-Widlar Showing the Break-loop Homotopy.

the break. With a predetermined sweep range, any voltage at which $V_2'' = V_{IN}$ is an operating point. Since all the PFLs are broken, each input test signal $x = V_{IN}$ is related to exactly one output $V_2'' = f(x)$, so $f(x)$ is called the *return function* of the PFL. The return function is determined by the static I/V characteristics of the devices involved in each PFL. Besides, the I/V characteristics of real circuit devices associated with DC analysis are generally continuous. Thereby, a return function can be regarded as a continuous function.

4.3.2 Monotonic Return Function

Definition 1: If $\forall x_1, x_2 \in [a, b]$ such that $x_1 > x_2$ one has $f(x_1) > f(x_2)$, the return function $f(x)$ is called monotonic return function.

The return function in a circuit is composed of multiple dependency relationships, and each of these relationships is either a current to voltage or a voltage to current dependency resulting from the static I/V characteristics of a device in the feedback loop. Although the numerical I/V relationship is nonlinear and difficult to calculate from the related static I/V functions, the signs of these dependencies are fixed for a given type of device and do not change with operating points as discussed in section 4.2.2. The sign of derivative of the return function

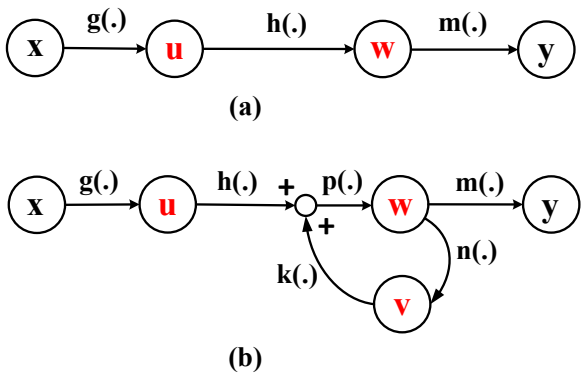


Figure 4.4: (a) Signal Map of a Return Function Without NFL; (b) Signal Map of a Return Function With NFL.

is therefore the product of each dependency relationship. For example, a "signal map" which illustrates the dependency relationships between a test signal and its return function when a PFL is broken is shown in Fig. 4.4(a) where x is the test signal, y is the return function, u, w stands for a controlling voltage or branch-current in the DDG and g, h, m are the static I/V functions related by each dependency relationship between vertices. It should be noted that the numerical values of these dependency relationships vary with the operating points but their fixed signs result from the static I/V functions of a certain device in the feedback loop. Since y is a composition of related static I/V functions, i.e., $y = m(h(g(x)))$, we can calculate the derivative of the return function from the chain rule as follows.

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial w} \cdot \frac{\partial w}{\partial u} \cdot \frac{\partial u}{\partial x} = \frac{\partial m}{\partial w} \cdot \frac{\partial h}{\partial u} \cdot \frac{\partial g}{\partial x} \tag{4.2}$$

So the sign of derivative of the return function is,

$$sgn(\frac{\partial y}{\partial x}) = sgn(\frac{\partial m}{\partial w}) \cdot sgn(\frac{\partial h}{\partial u}) \cdot sgn(\frac{\partial g}{\partial x}) \tag{4.3}$$

Suppose g, h and m form a PFL, $sgn(\frac{\partial m}{\partial w}) \cdot sgn(\frac{\partial h}{\partial u}) \cdot sgn(\frac{\partial g}{\partial x}) > 0$. Then the derivative of return function $\frac{\partial y}{\partial x} > 0$ from (4.3).

However, if only the PFLs are broken, there may exist NFLs in the signal path from test signal x to the return function y. Moreover, in most practical circuits, only simple NFL structures exist and complex nested NFLs are seldom used, so only a simple example is shown here as in Fig. 4.4 (b). Assuming $w \to v \to w$ is a NFL, it has $\frac{\partial k}{\partial v} \cdot \frac{\partial n}{\partial w} \cdot \frac{\partial p}{\partial z} < 0$ where $z = h(u) + k(n(w))$.

Since $w = p(z)$, it gets,

$$\frac{\partial w}{\partial u} = \frac{\partial p}{\partial z} \cdot (\frac{\partial h}{\partial u} + \frac{\partial k}{\partial v} \cdot \frac{\partial n}{\partial w} \cdot \frac{\partial w}{\partial u})$$
$$\Rightarrow \frac{\partial w}{\partial u} = \frac{\frac{\partial p}{\partial z} \cdot \frac{\partial h}{\partial u}}{1 - \frac{\partial k}{\partial v} \cdot \frac{\partial n}{\partial w} \cdot \frac{\partial p}{\partial z}} \tag{4.4}$$

From (4.4), it can be concluded that the sign of the $\frac{\partial w}{\partial u}$ is equal to the sign of $\frac{\partial p}{\partial z} \cdot \frac{\partial h}{\partial u}$ as $\frac{\partial k}{\partial v} \cdot \frac{\partial n}{\partial w} \cdot \frac{\partial p}{\partial z} < 0$. Similar to the previous example, $\frac{\partial y}{\partial x} = \frac{\partial y}{\partial w} \cdot \frac{\partial w}{\partial u} \cdot \frac{\partial u}{\partial x}$. As a result, it gets,

$$sgn(\frac{\partial y}{\partial x}) = sgn(\frac{\partial m}{\partial w}) \cdot sgn(\frac{\partial p}{\partial z}) \cdot sgn(\frac{\partial h}{\partial u}) \cdot sgn(\frac{\partial g}{\partial x}) \tag{4.5}$$

If g, h, p and m form a PFL, $sgn(\frac{\partial m}{\partial w}) \cdot sgn(\frac{\partial p}{\partial z}) \cdot sgn(\frac{\partial h}{\partial u}) \cdot sgn(\frac{\partial g}{\partial x}) > 0$, so the derivative of return function $\frac{\partial y}{\partial x} > 0$ from (4.5).

Therefore, in practical circuits, if only PFLs are broken, the return function is monotonic.

For traditional break-loop methods, a linear ramp up voltage source is inserted and swept over the whole power supply range to search for operating points. However, this method has low efficiency if high accuracy or large circuit scale is required. By analyzing the properties of the return function and voltage intervals, we can quickly detect the existence of undesired operating points in a circuit.

4.3.3 Definitions and Theorems for General Return Function

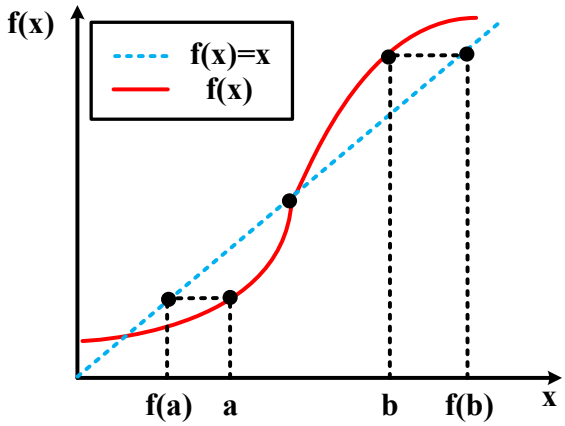
First, one type of interval could be defined as a Sign-Change Interval(SCI); its return function crosses the input interval.

Definition 2: An interval $[a, b]$ is a Sign-Change Interval (SCI) of the PFL if $(f(a) - a) \times (f(b) - b) < 0$, where $f(x)$ is the return function of the PFL, continuous on $[a, b]$.

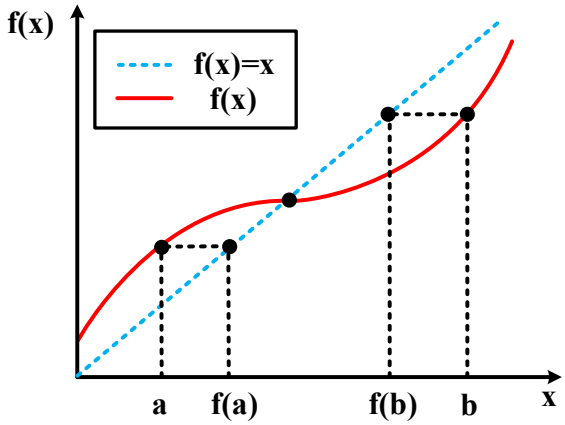
Fig. 4.5 gives two examples of $SCIs$. For an SCI, the return function of the PFL is continuous in $[a, b]$, and combined with the condition $(f(a) - a) \times (f(b) - b) < 0$, there is at least one operating point $\varepsilon \in (a, b)$ such that i.e. $f(\varepsilon) = \varepsilon$ from the intermediate value theorem. The following is theorem 1.

Theorem 1: There is at least one operating point in an SCI.

The second type of interval to be defined is the Sign-Definite Interval (SDI) whose return function has no intersection with the input interval.



(a)



(b)

Figure 4.5: Examples of Sign Change Interval (SCI) With Monotonic Return Function.

Definition 3: An interval $[a, b]$ is a Sign-Definite Interval (SDI) of the PFL if $f(x) > x, \forall x \in [a, b]$ or $f(x) < x, \forall x \in [a, b]$, where $f(x)$ is the return function of the PFL.

In Fig. 4.5, two examples of SDI are given.

It is obvious that in an SDI there is no operating point where $f(x) = x$, so we can state:

Theorem 2: There is no operating point in an SDI.

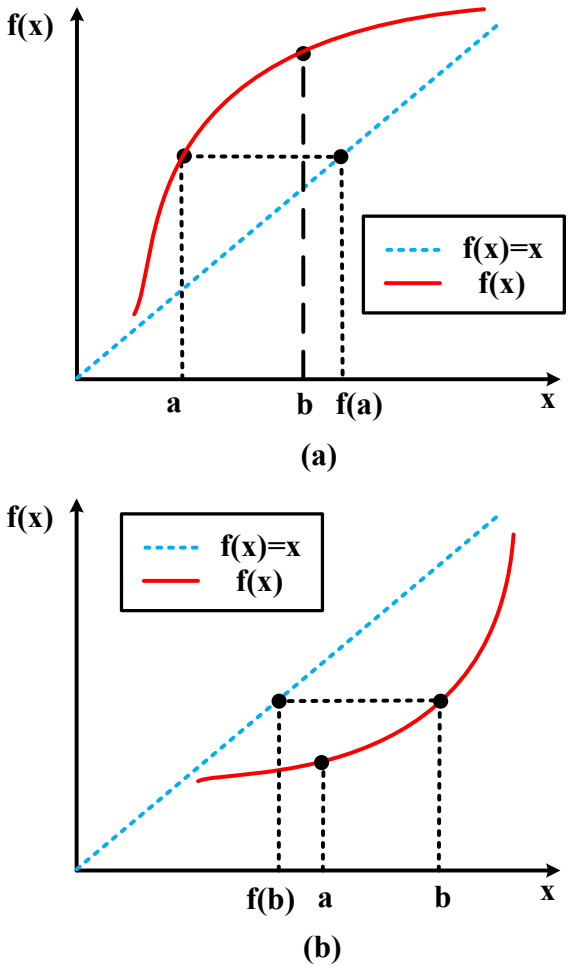


Figure 4.6: Examples of Sign Definite Interval (SDI) With Monotonic Return Function.

4.3.4 Definitions and Theorems for Monotonic Return Function

In this subsection, the monotonic return function is discussed.

From its definition, an SDI is difficult to verify. However, for a monotonic return function, an SDI can easily be identified using *Lemma 1* and *Corollary 1*.

Lemma 1: Assuming $f(x)$ to be a monotonic return function of the PFL, if $f(a) > b$ or $f(b) < a$, then $[a, b]$ is an SDI.

Proof. As shown in Fig. 4.6 (a), if $f(a) > b$, it follows that $f(a) > b \geq a$. Since $f(x)$ is a monotonic return function, we obtain $f(b) \geq f(a) > b$.

Assuming that there is one $c \in (a, b)$ such that $f(c) \leq c$. Then, because $f(a) > b$ and $c < b$, we obtain $f(a) > f(c)$, contradicting the assumption that $f(x)$ is a monotonic return function of the PFL.

This proves that $f(x) > x, \forall x \in [a, b]$, i.e., $[a, b]$ is an SDI.

Similarly, as shown in Fig. 4.6 (b), it could be proven that if $f(b) < a$, then $f(x) < x, \forall x \in [a, b]$, i.e., $[a, b]$ is an SDI. □

As shown in Fig. 4.5 (b), for a monotonic return function $f(x)$, if $f(a) > a$, then $[a, f(a)]$ is an SDI because there is no $f(x) = x, \forall x \in [a, f(a)]$. Similarly, $[f(b), b]$ is also an SDI if $f(b) < b$. We therefore obtain corollary 1 and its proof as follows.

Corollary 1: Assuming that $f(x)$ is a monotonic return function of the PFL, if $f(a) > a$, then $[a, f(a)]$ is an SDI; If $f(b) < b$, then $[f(b), b]$ is an SDI.

Proof. If $f(a) > a$, assume that there is one $c \in (a, f(a)]$ that $f(c) \leq c$.

Then, because $f(a) \geq c$ and $c > a$, we obtain $f(a) \geq c \geq f(c)$ contradicting the assumption that $f(x)$ is a monotonic return function. This proves that if $f(a) > a$, then $f(x) > x, \forall x \in [a, f(a)]$, i.e., $[a, f(a)]$ is an SDI.

Similarly, it could be proven that, if $f(b) < b$, then $[f(b), b]$ is an SDI. □

4.3.5 Verification for Different Types of Return Functions

As in the previous discussion, if the return function is monotonic, the SDI can be detected by the *Lemma 1* and *Corollary 1*, resulting in higher verification efficiency as illustrated in a later section. We also known the monotonic return function could be obtained by breaking only PFLs without cutting off any NFL. So from the DDG of the analog circuit, a graph theory technique will be employed to search the break node where all the PFLs can be broken without interrupting any NFL[65]. The existence of such break node determines the monotonicity of the corresponding return function. For monotonic and non-monotonic return function, two types of Divide and Contraction algorithms are proposed and illustrated in the following two sections.

4.4 Divide and Contraction Algorithms for Monotonic Return Function

Two types of divide and contraction algorithms applied to monotonic return function are illustrated in this section. We first introduce the Monotonic Divide and Contraction (MDC) algorithm to be applied to the monotonic return function. Since the desired operating point is usually specified by the circuit designer, an improved algorithm called the User-defined Monotonic Divide and Contraction (UMDC) algorithm is discussed in the second subsection.

4.4.1 Monotonic Divide and Contraction (MDC) Algorithm

From Theorem 1, $SCIs$ could be verified by evaluating return functions of test interval boundaries. The return functions $f(a)$ and $f(b)$ are obtained by DC simulation for the test voltage interval $[a, b]$. Then, the test interval could be proven to be an SCI if $(f(a)-a)\times(f(b)-b) < 0$. If two separate $SCIs$ are identified, it can be concluded that more than one operating point exists in the circuit, i.e., there is at least one undesired operating point. For a monotonic return function, SDI could be easily identified by *Lemma 1* and *Corollary 1*. Dropping all the $SDIs$ in the test interval could significantly improve the verification efficiency.

We therefore will develop our Monotonic Divide and Contraction (MDC) algorithm, whose flow chart is shown in Fig. 4.7.

Step 0. Two queues should be initialized at this step: the Test Interval Queue (TIQ), a queue containing all the input testing intervals; and an SCI queue ($SCIQ$), the queue including all the $SCIs$ found. The TIQ is initialized to the power supply range of the circuit and the return functions of the test interval boundaries are obtained by circuit simulation. From the definition of SCI, $SCIQ$ is also initialized to the power supply range.

Step 1. In this step, the first interval in the TIQ is popped out for testing. Because TIQ is sorted from large width to small width (this will be explained in step 7), the first test interval in the TIQ is always the largest and denoted as $TIQ_{(1)}$. Consequently, width of $TIQ_{(1)}$ less than our setting error tolerance ε indicates that all the intervals in the TIQ are smaller than ε and no existence of undesired operating point has been found in the circuit under the given error-tolerance setting, and the flow exits and reports this result.

Step 2. The $TIQ_{(1)}$ is divided into two parts with equal width, e.g., the interval $[a, b]$ is divided into $[a, c]$ and $[c, b]$, where $c = (a + b)/2$. Moreover, $f(c)$ is obtained by simulation and recorded.

Step 3. From previous steps, the $TIQ_{(1)}$ is divided into two sub-intervals, and this step assesses whether they are SCI. If any of them is an SCI, the flow goes to step 4; if not, it goes to step 6.

Step 4. Since one or two $SCIs$ are found from step 3, we need to check whether any of them is an interval separated from the existing SCI in the $SCIQ$. If the SCI in the $SCIQ$ contains the new SCI identified in step 3, the former is replaced by the latter; otherwise the new SCI is inserted into the $SCIQ$.

Step 5. The size of the $SCIQ$ is examined in this step, and if larger than or equal to two, it denotes existence of more than one operating points, i.e., there is at least one undesired operating point. In that case the algorithm exits and gives a report; otherwise proceed to step 6.

Step 6. This step identifies and drops all the $SDIs$ in the two sub-intervals. It worth mentioning that, the whole of one sub-interval may be an SDI according to *Lemma 1.* In that case, there is only one sub-interval needed to be inserted back into TIQ in following step 7.

Step 7. Sub-intervals are inserted back into the TIQ. Because an interval with larger width usually implies greater likelihood of an operating point, the sub-intervals are inserted into the TIQ according to their width. This step is accomplished by a binary search in the TIQ and an insert operation. After finishing this step, the TIQ is refreshed and the flow moves back to step 1 to check the new $TIQ_{(1)}$.

4.4.2 User-defined Monotonic Divide and Contraction (UMDC) Algorithm

In practice, the desired operating point is known by the circuit designers and the desired operating point with a tolerable range could be defined as one SCI by users. In such a case, finding another SCI separated from the one that is user-defined indicates existence of an undesired operating point. Based on this consideration, the MDC algorithm could be modified into a User-defined Monotonic Divide and Contraction (UMDC) algorithm and efficiency in

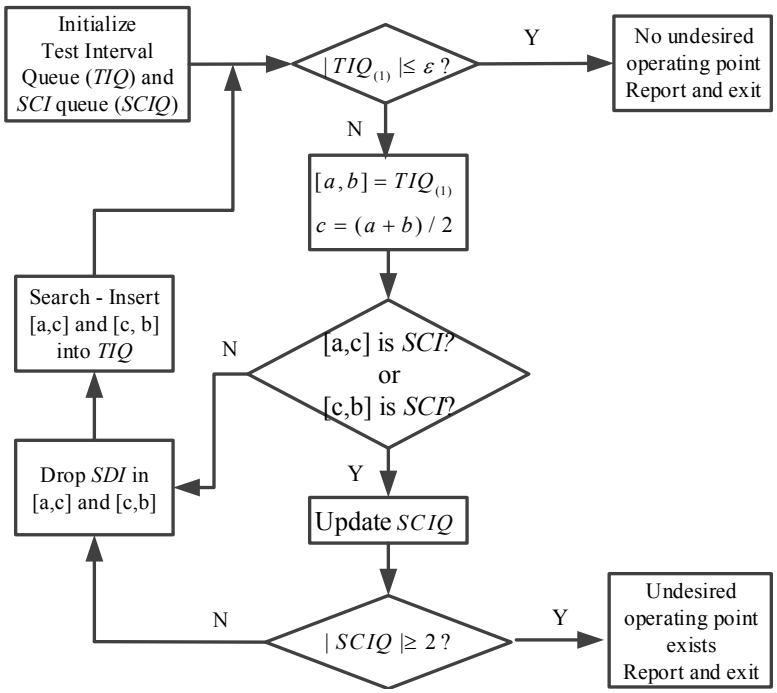


Figure 4.7: Flow Chart of the MDC Algorithm.

discovering the existence of undesired operating points could be further improved. The UMDC algorithm can be illustrated as follows:

Step 0. Similar to the MDC algorithm, TIQ and $SCIQ$ are initialized at this step. The only difference is in the initialization of $SCIQ$. For the UMDC, $SCIQ$ is initialized as containing only one interval, the user-defined interval including the desired operating point.

Step 1. This step is the same as MDC step 1.

Step 2. If the width of $TIQ_{(1)}$ is larger than ε, this step checks to see if the test interval $TIQ_{(1)}$ is embraced by the user-defined SCI in the $SCIQ$. If so, the $TIQ_{(1)}$ is ignored and the flow proceeds to step 1, otherwise proceed to step 3.

Step 3. This step is the same as MDC step 3.

Step 4. Similar to MDC step 4 except that if any of the sub-intervals is verified to be an SCI, it reports that the existence of undesired operating points has been found and the flow exits. Since step 2 excludes all test intervals embraced by the user-defined SCI, finding one SCI denotes that at least one undesired operating point exists.

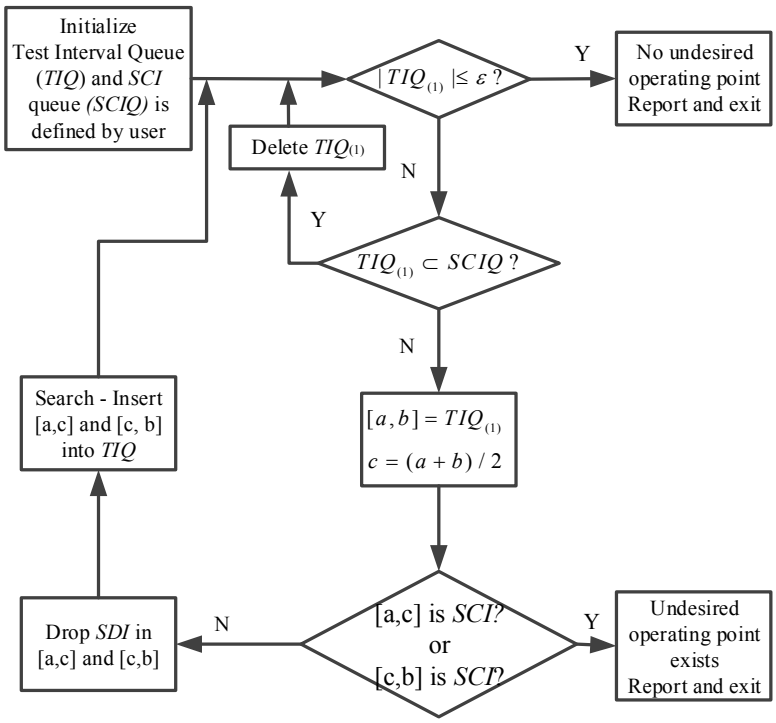


Figure 4.8: Flow Chart of the UMDC Algorithm.

Step 5 and *Step 6* are the same as MDC step 6 and step 7, respectively.

The flow chart of the UMDC algorithm is shown in Fig. 4.8. Compared to the MDC, the UMDC algorithm is simplified and the computational time for verifying the undesired operating point is reduced.

4.5 Divide and Contraction Algorithms for Non-monotonic Return Function

PFLs in circuit may share some parts with NFL so PFLs cannot be broken without cutting off the NFLs. In this case, the return function is not guaranteed to be monotonic and the MDC/UMDC algorithms cannot be applied. For such circuits, we proposed a General Divide and Contraction (GDC) algorithm and a User-defined General Divide and Contraction (UGDC) algorithm.

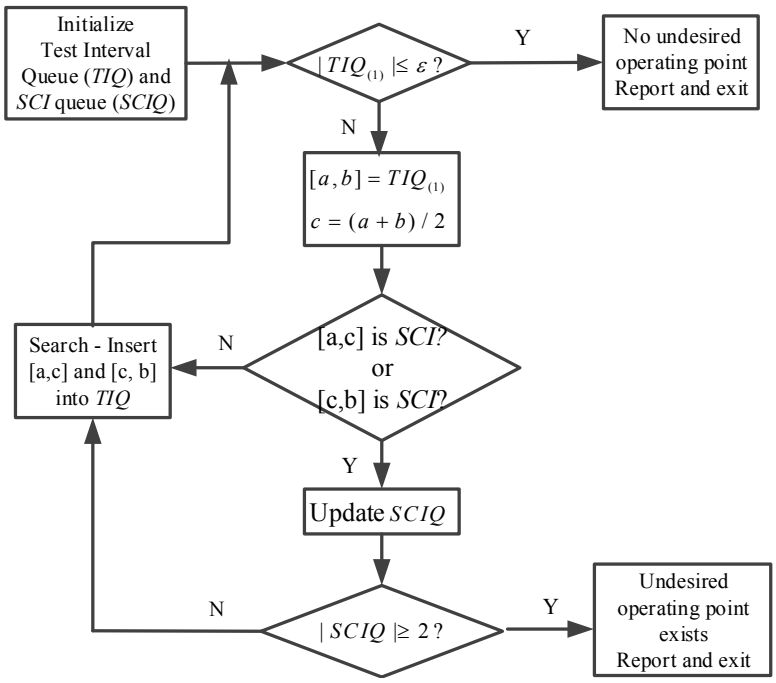


Figure 4.9: Flow Chart of the GDC Algorithm.

4.5.1 General Divide and Contraction (GDC) Algorithm

The flow chart of the GDC algorithm is shown in Fig. 4.9. Compared with the flow chart of the MDC, the GDC cannot drop $SDIs$ in the test interval due to its return function is not monotonic. Although it usually has less efficiency than the MDC algorithm, the GDC algorithm can be applied to more general circuits for verification.

4.5.2 User-defined General Divide and Contraction (UGDC) Algorithm

Similar to the UMDC, the GDC algorithm could be modified into a User-defined General Divide and Contraction (UGDC) algorithm if the desired operating point is defined by the user. The flow chart of the UDC algorithm is shown in Fig. 4.10.

Compared with the GDC, the UGDC algorithm reduces computational time and improves verification efficiency.

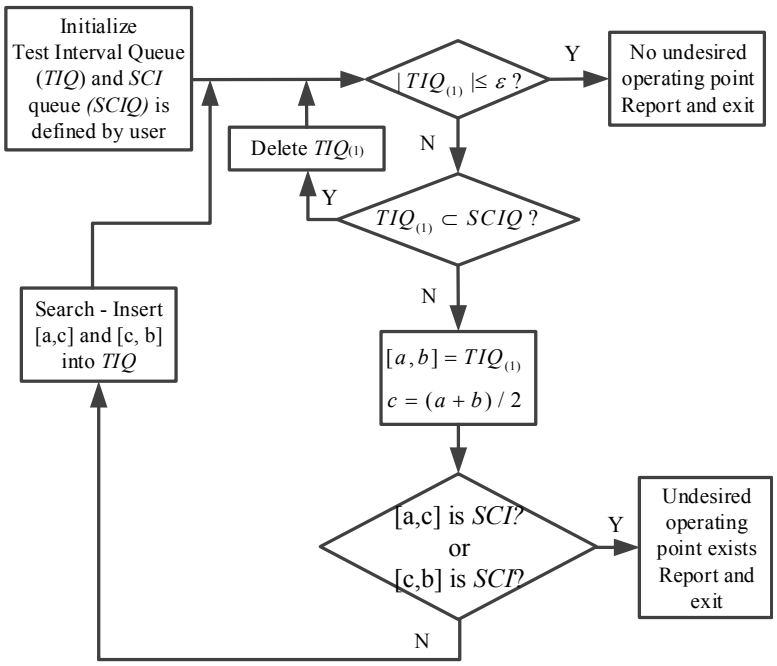


Figure 4.10: Flow Chart of the UGDC Algorithm.

4.6 Simulation Results

The proposed divide and contraction verification method has been implemented in standard circuit simulators such as Spectre and Ultrasim. It can automatically find all the PFLs and NFLs in the circuit and search for break nodes. Then it employs the proposed divide and contraction algorithms to identify the existence of undesired operating point. According to [68], the computational complexity for converting a circuit to a DDG and identifying and breaking loops is approximately linear with circuit size; this is expected to be similar to a single DC simulation. Thus the computational requirement of our method is dominated by the number of DC simulations, and that is why we just compare the number of DC simulations among various methods. A set of widely used benchmark circuits designed by $0.6\mu m$ CMOS techniques including bias generators, current/voltage references, temperature sensors, and op-amps [64] has been verified. Some have been selected as examples to demonstrate the concept and efficiency of our method, as shown in this section.

Example 1: Consider one of the benchmark circuits—the bootstrapped Vt reference circuit and its circuit graph (as shown in the Fig. 4.11). There is one PFL ($I_2 \rightarrow V_1 \rightarrow I_1 \rightarrow V_2 \rightarrow I_2$)

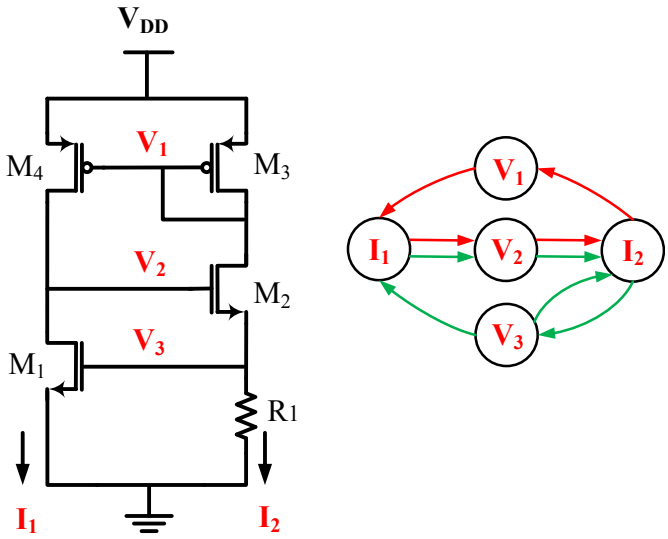


Figure 4.11: Circuit Graph for the Bootstrapped Vt Reference Circuit.

and two NFLs ($I_2 \to V_3 \to I_1 \to V_2 \to I_2$ and $I_2 \to V_3 \to I_2$)in this circuit. If V_1 is broken, then only the PFL is broken, resulting in a monotonic return function. A voltage source is inserted at the gate of M_4 to drive the loop.

The proposed MDC algorithm can be applied to this circuit to verify if it has multiple operating points; a traditional linear-sweeping method (linearly ramp up the voltage source to get the operating point) is provided for comparison. TABLE 4.2 shows the simulation results: with temperature at 75°C and V_{DD} at $5V$, the designed bootstrapped Vt reference circuit has three operating points according to the result of the linear sweeping method using a $0.1mV$ step size. The operating point $3.5885V$ is our desired operating point, while $4.5658V$ and $4.6301V$ are undesired operating points. On the other hand, MDC provides one operating point $4.6301V$ and an interval in which lies at least one operating point, $[3.4002V, 4.0154V]$. From the MDC algorithm, the existence of undesired operating points in this circuit is verified. More importantly, although the error tolerance of MDC is the same as the step size of the linear sweeping method, the efficiencies of these two methods are significantly different; it took 50000 DC simulations for the linear sweeping method to find the three operating points, while the MDC algorithm found the existence of undesired operating point in only 8 DC simulations! The proposed divide and contraction algorithm dramatically reduces computational requirements

Table 4.2: Simulation Results of Bootstrapped Vt Reference Circuit

VDD=5V Temperature=75 °C	Linear Sweeping	MDC	GDC
Number of simulation	50000	8	63
Step size/Error tolerance	$0.1mV$	$0.1mV$	$0.1mV$
Operating points	3 OP: $3.5885V$ (desired one), $4.5658V$ (undesired) and $4.6301V$ (undesired)	$[3.4002V, 4.0154V]$ and $4.6301V$[a]	$[3.5156V, 3.5938V]$ and $[4.6094V, 4.6875V]$

[a] When the width of the SCI is less than the error tolerance, it is regarded as one operating point.

and identifies the existence of undesired operating points with much higher efficiency.

For comparison, it took 63 DC simulations for the GDC algorithm to find two test intervals that included operating points as shown in TABLE 4.2. Although this is still very fast compared with the linear sweeping method, GDC is usually less efficient than the MDC, so for the general analog circuit, the proposed method will automatically check whether there exists a break node to obtain a monotonic return function [68]. With such a break node, it applys the MDC/UMDC algorithm; otherwise it uses the GDC/UGDC algorithm.

Example 2: To remove the undesired operating point, a certain start-up circuit should be added to the circuit. Fig. 4.12 illustrates a bootstrapped Vt reference where M_5, M_6 and R_2 form a start-up circuit. Comparison of results with those of linear-sweeping method is shown in TABLE 4.3. The step size of the linear sweep is also the same as the error tolerance of the MDC algorithm. Although the two methods provide the same desired operating point, the linear-sweep method took 50000 simulations compared to only 17 simulations for the MDC algorithm. The verification efficiency can be improved even further by employing the UMDC algorithm. Given a user-defined interval $[3.5635V, 3.6135V]$ and $0.1mV$ error tolerance, the UMDC concluded there were no other operating points beyond the user-defined SCI in only 11 DC simulations. The proposed verification method could thus efficiently verify the efficacy of the start-up circuit.

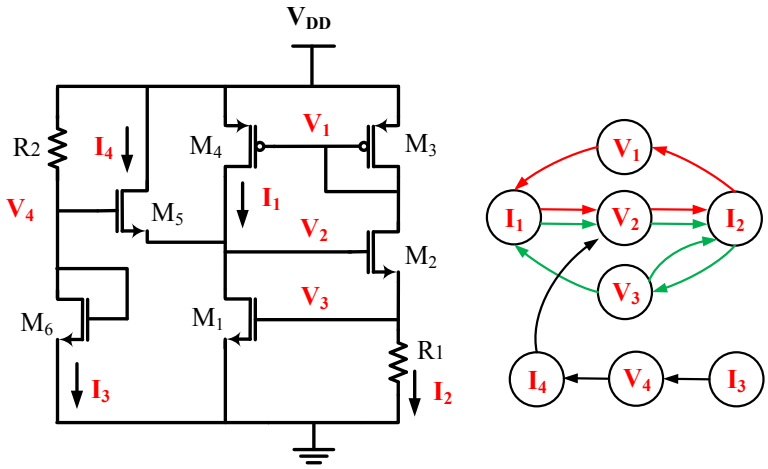


Figure 4.12: Circuit Graph for the Bootstrapped Vt Reference Circuit With Start-up Circuit.

Table 4.3: Simulation Results of Bootstrapped Vt Reference With Start-up Circuit

VDD=5V Temperature=75°C	Linear Sweeping	MDC	UMDC
User-define SCI	N/A	N/A	$[3.5635V, 3.6135V]$
Number of simulation	50000	17	11
Step size/Error tolerance	$0.1mV$	$0.1mV$	$0.1mV$
Operating points	$1OP$: $3.5885V$	$1OP$: $3.5885V$	*No other OP*

Example 3: Let's look at another example circuit-the self-biased Banba bandgap reference [69] shown in Fig. 4.13. From its circuit graph (Fig. 4.14), there are three PFLs and five NFLs:

PFL 1: $I_5 \rightarrow V_A \rightarrow I_3 \rightarrow V_1 \rightarrow I_2 \rightarrow V_2 \rightarrow I_1 \rightarrow V_o \rightarrow I_5$

PFL 2: $I_2 \rightarrow V_2 \rightarrow I_1 \rightarrow V_o \rightarrow I_7 \rightarrow V_3 \rightarrow I_6 \rightarrow V_{bias} \rightarrow I_2$

PFL 3: $I_5 \rightarrow V_A \rightarrow I_3 \rightarrow V_4 \rightarrow I_2 \rightarrow V_2 \rightarrow I_1 \rightarrow V_o \rightarrow I_5$

NFL 1: $I_2 \rightarrow V_2 \rightarrow I_1 \rightarrow V_o \rightarrow I_4 \rightarrow V_B \rightarrow I_2$

NFL 2: $I_2 \rightarrow V_2 \rightarrow I_1 \rightarrow V_o \rightarrow I_7 \rightarrow V_3 \rightarrow I_6 \rightarrow V_{bias} \rightarrow I_3 \rightarrow V_1 \rightarrow I_2$

NFL 3: $I_1 \rightarrow V_o \rightarrow I_7 \rightarrow V_3 \rightarrow I_6 \rightarrow V_{bias} \rightarrow I_1$

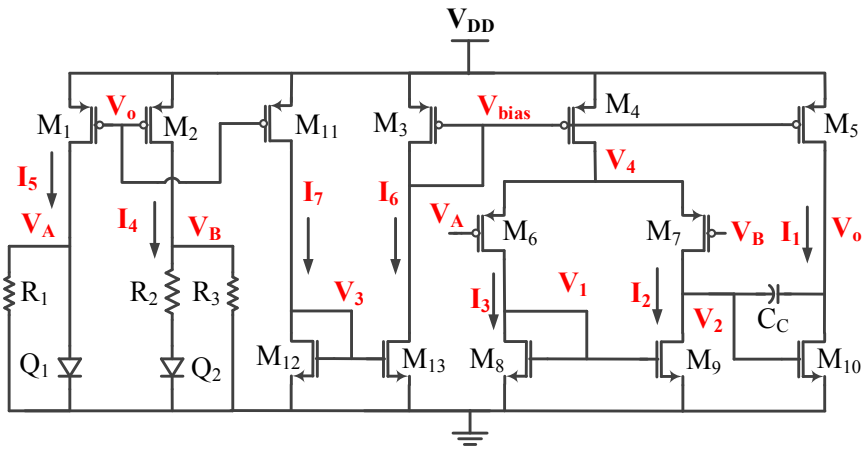


Figure 4.13: Self-biased Banba Bandgap Reference Circuit.

NFL 4: $I_2 \to V_2 \to I_1 \to V_o \to I_7 \to V_3 \to I_6 \to V_{bias} \to I_3 \to V_4 \to I_2$

NFL 5: $I_2 \to V_4 \to I_3 \to V_1 \to I_2$

From the circuit graph, there is no node in this circuit where all the PFLs could be broken without interrupting any NFL, so only GDC/UGDC can be applied to identify the existence of undesired operating point. V_o is chosen to be broken into V_o' and V_o''. A voltage source is inserted at node V_o'. The circuit implementation is shown in Fig. 4.15. TABLE 4.4 shows the results: with temperature at 0°C and V_{DD} at $2V$, the linear-sweep method requires 20000 simulations with step size $0.1mV$ to find three operating points: $0.7383V$ is the desired operating point, while $0.8857V$ and $1.1305V$ are the undesired operating points. In contrast, with $0.1mV$ error tolerance, the GDC finds the two intervals $[0.5000V, 0.7500V]$ and $[1.0000V, 1.5000V]$ in 7 DC simulations. Furthermore, given a user-defined interval $[0.7133V, 0.7633V]$, the UGDC finds the interval $[0.9750V, 2.0000V]$ beyond the user-defined SCI in only 5 DC simulations. This demonstrated that the GDC/UGDC algorithm can efficiently identify the existence of undesired operating point in more general analog circuits.

4.7 Discussion

In most MOS analog circuits, there are various sub-circuits with feedback structures which are susceptible to the existence of undesired operating points, such as bias generators, current/voltage references, temperature sensors. However, existing circuit simulators provide only a

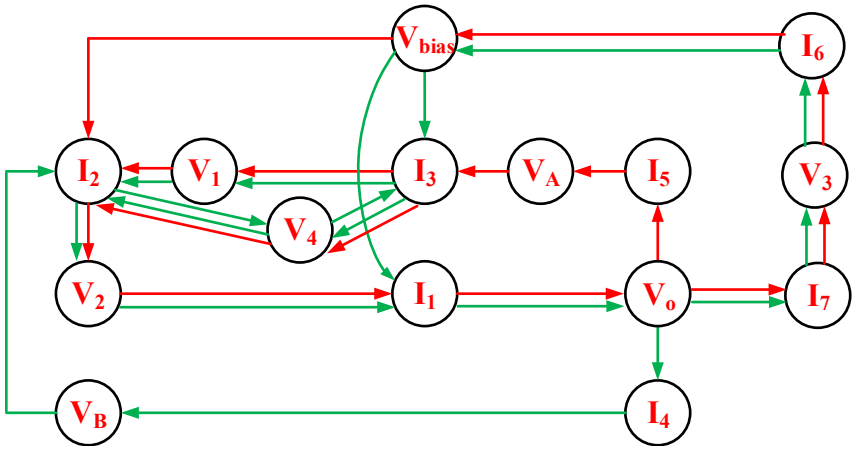


Figure 4.14: Circuit Graph of the Self-biased Banba Bandgap Reference Circuit

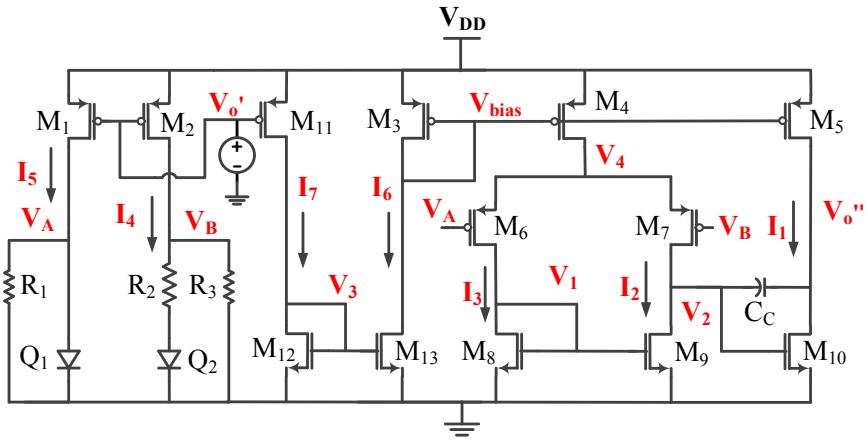


Figure 4.15: Implementation of Breaking Loop of the Self-biased Banba Bandgap Reference Circuit

single operating point, and general methods to solve all operating points in even rather simple circuits do not exist. The proposed method targets to efficiently identify the existence of undesired operating points in those widely used analog circuits. It can be an important tool for the designers indicating the existence of undesired operating points.

Compared with other methods for finding operating points in literature, the proposed analog verification method offers several valuable advantages:

First, it significantly reduces computational requirements to identify the existence of undesired operating point in analog circuits. It can automatically identify all the feedback loops from a circuit netlist and determine their signs. Moreover, the number of DC simulations re-

Table 4.4: Simulation Results of Self-biased Banba Bandgap Reference Circuit

VDD=2V	Linear Sweeping	GDC	UGDC
Temperature(°C)	0	0	0
User-define SCI	N/A	N/A	$[0.7133V, 0.7633V]$
Number of simulation	20000	7	5
Step size/Error tolerance	$0.1mV$	$0.1mV$	$0.1mV$
Operating points	3 OP: $0.7383V$ (desired), $0.8857V$ (undesired) and $1.1305V$ (undesired)	$[0.5000V, 0.7500V]$ and $[1.0000V, 1.5000V]$	$[0.9750V, 2.0000V]$ is found

quired to find the presence/absence of undesired operating points is dramatically reduced by the divide and contraction algorithms, so it can be applied to larger size analog circuits and is therefore appropriate for use in more practical applications.

Second, this method does not rely on simplifying the device models or linearizing the circuit. As a result, unlike topological or piecewise-linear methods, it takes into account the device and environmental parameters and provides an accurate verification result under practical simulation circumstances.

Third, it can easily be implemented in standard circuit simulators such as Spectre or Ultrasim. The implementation of this method does not require building any new MOS model or other complicated methodology to form continuous deformation as in other homotopy methods. In comparison, the whole flow of our method has been realized as an automatic tool with much less effort.

This method does, however, have some limitations: For large scale circuits, graph theory techniques can be employed to partition the DDG into strongly connected components (SCCs),

which is equivalent to partitioning the original circuit into several sub-circuits [68] that can be separately verified. There may exist one or multiple PFLs in an SCC. If all PFLs can be broken at a single node for each SCC, the circuit is called Single-Input Single-Output (SISO) circuit. In its current form, the proposed divide and contraction method only applies to SISO circuits that can have a single input and a single output when all the PFLs are broken in each SCC. For Multiple-Input Multiple-Output (MIMO) circuits, that is, circuits can only have multiple inputs and multiple outputs when all the PFLs are broken in any SCC, an extended version of the divide and contraction algorithms needs to be developed.

The applications of the proposed method are still limited to MOS circuits and whether it gives correct result for BJT circuits is not guaranteed. This is due to the fact that the impedance at the base of a BJT could not be generally regarded as infinite as in the gate of a MOSFET. The loop loading effect of breaking the BJT circuits thus could not be ignored. To solve this problem, a method for breaking a BJT circuit is also under study.

Furthermore, the robustness of proposed verification method is worth discussion. For both linear sweeping and the proposed method, adjacent operating points may be undetectable if they lie in an interval with width less than the error tolerance ε. Besides only reducing the error tolerance as linear sweeping method, the proposed method can also improve its robustness as follows.

In practice, this method not only checks the width of a test interval $[a, b]$ but also evaluates the values of $f(a) - a$ and $f(b) - b$. If no undesired operating point is identified under the existing error tolerance setting, it also gives warnings if there exists suspicious intervals that may include undesired operating point and designers need to check them with smaller error tolerance setting. Specifically, for a monotonic return function $f(x)$, there are two types of suspicious intervals. The first meets the following conditions: $b - a \leq \varepsilon$, $0 < f(a) - a$, $0 < f(b) - b$, and $f(a) - a \leq \varepsilon$. This is due to the fact that $\forall x \in (a, b)$ if $b - a \leq \varepsilon$, $0 < f(a) - a$, $0 < f(b) - b$ and $\varepsilon < f(a) - a$, then $f(x) > x$. It can be proven as follows. Since $f(x)$ is a monotonic return function, $f(x) > f(a) > a + \varepsilon$. Also $x < a + \varepsilon$ due to that $b - a \leq \varepsilon$ and $x \in (a, b)$. Thus, it gets $f(x) - x > a + \varepsilon - x > 0$. The other type of suspicious interval is $b - a \leq \varepsilon$, $0 < a - f(a)$, $0 < b - f(b)$ and $b - f(b) \leq \varepsilon$ and similar proof can be applied to.

Moreover, running the verifications at different PVT (power supply, process and temperature) settings can obtain a more robust result. The proposed method can efficiently identify the existence of undesired operating point in each PVT setting. With PVT variations, return functions may have large changes and the existence of undesired operating points could be found.

4.8 Conclusion

In this chapter, an efficient and systematic analog verification method for identifying the existence of undesired operating points for MOS analog circuits has been proposed. Unlike traditional approaches which find all operating points of the circuit, the method searches voltage intervals that contain undesired operating point based on the break-loop homotopy method.

First, an approach that automatically converts a MOS analog circuit into a directed dependency graph is introduced. Then, all the feedback loops in the circuit are found and their signs determined. Positive feedback loops breaking method and selection of breaking nodes are studied to decide if a monotonic return function can be obtained. Based on the monotonicity of the return function, two types of divide and contraction algorithms that can efficiently identify the existence of undesired operating point in analog circuits are presented.

Compared with other approaches in literature, the proposed method provides several valuable advantages. Since this method is based on a circuit-level break-loop homotopy method, it can take into account device and environment parameters and provide an accurate verification result. The realization of proposed method requires much less effort and the method can be easily implemented in standard circuit simulators such as Spectre or Ultrasim. Finally, since this method significantly cuts down on computational requirements, it can be applied to larger size analog circuits and is therefore eligible for more practical applications.

Simulation results show the proposed method to be effective and efficient in identifying undesired operating points in a set of commonly used benchmark circuits, including bias generators, voltage references, temperature sensors, and op-amps.

CHAPTER 5. TWO DIMENSIONAL ANALOG VERIFICATIONS AGAINST UNDESIRED OPERATING POINTS FOR MOS ANALOG CIRCUITS

5.1 Introduction

The existence of undesired operating points is an important problem in many analog circuits such as bias generators, current references, temperature sensors, and Bandgap references. In addition, various feedback approaches, e.g., self-biasing, bootstrapping, self-stabilization, and digitally-assisted-analog are applied in analog circuits to enhance their performance. Inevitably, these methods result in structures with one or more feedback loops that make them vulnerable to the presence of undesired operating points. The effect of undetected operating points can be devastating, particularly in critical systems such as automotive, health care, and military products. As a result, identifying the presence/absence of undesired operating points is a critical problem in circuit design, and undesired operating points should be identified and eliminated for proper circuit operation.

However, since the existing circuit simulators provide only a single operating point [8], recognizing the existence of undesired operating points largely depends on the experience of designers. To remove undesired operating points, the most popular technique uses start-up circuits. When power is supplied, a start-up circuit is supposed to circumvent entrance of the circuit into the undesired operating points. Nonetheless, because of unanticipated transients and variations of working conditions, the circuit with start-up circuits may still enter an unknown operating point and fail [63]. With some circuits, even the most experienced designers are not aware that a circuit they designed has undesired operating points, which often go undetected in the standard simulations used in the design process. It can be very costly if

the undesired operating point in the circuit is first detected in field by a customer. However, more and more basic circuits are designed by new graduates or inexperienced analog engineers and circuit designers continuously add "smart components" in their design utilizing feedback. Consequently, the undesired operating points may increasingly be a widespread and insidious problem plaguing the circuit design industry.

Various specific approaches in the literature could be used to detect undesired operating points in circuits. Topological methods identify multiple operating points, using graphical representation and certain topological criteria [41, 42, 43, 44, 45]. However, although topological methods may identify the existence of multiple operating points by examining the structure of a circuit, they never take into account device and environmental parameters, so their validity is limited by certain assumptions regarding the circuits.

In additional to topological methods, many approximation approaches for obtaining circuit solution have been developed, since formidable effort would be necessary to find all the mathematical solutions of the circuit non-linear equations matrix . Piecewise-linear methods can discover all operating points of a circuit based on piecewise-linear approximations of non-linear devices [46, 47, 48, 49]. Contraction methods can frame non-linear functions appearing in the mathematical description of the circuit and exploit formulas using matrix theory to contract a hyper-rectangular region that includes the solutions [50, 51]. Interval methods seek a set of solutions based on interval analysis theory [52, 53]. However, all these methods rely on simplifying device models or replacing non-linear equations by linear ones; consequently, they are often not applicable to large-scale circuits or to many practical applications.

The homotopy/continuation approach is another popular method of identifying a circuit's operating points. In general, this type of method entails embedding one or two continuation parameters into a set of non-linear equations, thereby causing a continuous deformation from a known or easily-computed solution to the required solution [54, 55, 56, 57]. The homotopy/-continuation methods could trace DC solutions, but they could not guarantee that all operating points would be identified; also their implementation is tedious.

Because a positive feedback structure is the cause of multiple operating points in the circuit [42], break-loop homotopy methods have been proposed to find multiple operating points. This

often involves introduction of a voltage or current source that can be swept to trace operating points of a circuit [58, 6, 59], but it relies on linear ramp-up of inserted sources and generally has very low efficiency; also, achieving accurate results can require a very long simulation time because a small sweeping step is needed. It also assumes that the PFLs are already known and does not provide a method to automatically identify the feedback loops in circuits and distinguish between PFLs and negative feedback loops (NFLs).

In summary, all these methods attempt to find all the operating points of a circuit, a very complicated problem, which can be very difficult even with simple circuits, e.g., two-transistor circuits [60, 61, 62]. Therefore, simple and efficient verification methods that are guaranteed to find all operating points in even rather simple circuits do not exist.

Recently, a divide and contraction verification method for discovering the existence of undesired operating points was systematically proposed [70]. In contrast to traditional methods of dealing with discovery of operating points, this method does not try to find all operating points or even any operating point at all. It only targets finding voltage intervals that contain undesired operating points, thereby verifying the existence of undesired operating points. This essentially replaces a difficult root-finding problem with simply identifying the existence of test intervals that include undesired operating points. To achieve this, the method automatically converts a circuit netlist to a Directed Dependency Graph (DDG) and finds all the PFLs and NFLs in the circuit. By breaking all the PFLs and applying the proposed divide and contraction algorithm, the method can dramatically reduce computational requirements to determine the presence/absence of undesired operating point. However, the method focuses mainly on the one dimensional analog verification problems, i.e., the verification of fixed transistor size, fixed Process/Voltage supply/Temperature (PVT) setting and one input voltage source to break all the PFLs (B1P circuits).

In practice, designers also need to verify circuits with the transistor sizing, PVT variations or identify the existence of undesired operating points in complicated circuits whose PFLs cannot be broken at only one point (B2P circuits). This type of problems is called the two dimensional analog verification against undesired operating points, which will be discussed in this chapter. For this type of verification, a two dimensional vector field method is proposed,

which can effective identify the existence of undesired operating points by visualizing the return functions in the circuits.

This chapter is organized as follows: Section 2 introduces the two dimensional analog verification problems; section 3 discusses the two dimensional vector field characteristics; section 4 illustrates the proposed two dimensional vector field method. Application examples are given in section 5, and conclusions are stated in section 6.

5.2 Two Dimensional Analog Verification Problems

The verification method proposed in [70] mainly focuses on the one dimensional problems such as exists with a fixed PVT setting and one input voltage source to break all the PFLs. However, two dimensional analog verification also needs to be implemented, such as the PVT variation, transistor sizing or complicated circuits that need more than one point to break all the PFLs. These two dimensional analog verification problems are illustrated in this section.

5.2.1 B1P and B2P circuits

For large scale circuits, graph theory techniques can be employed to partition the DDG into strongly connected components (SCCs), which is equivalent to partitioning the original circuit into several sub-circuits [68] that can be separately verified. There may exist one or multiple PFLs in an SCC, and a break point set (BPS) of an SCC is used to represent a minimum subset of controlling-voltage nodes that will break all the PFLs if removed from the SCC. With use of existing algorithms in graph theory [71, 72], the BPS of an SCC can be found with high efficiency. The size of the BPS ($|BPS|$) stands for the minimum number of controlling-voltage nodes to break all the PFLs in an SCC.

If all the SCCs in the circuit have $|BPS| = 1$, i.e., all PFLs can be broken at a single point for each SCC, the circuit is called a break-one-point (B1P) circuit. Most of the commonly used analog circuits, including bias generators, current/voltage references, temperature sensors and op-amps are B1P circuits. If any SCC has $|BPS| = 2$ and all other SCCs have $|BPS| \leq 2$, the circuit is called a break-two-point (B2P) circuit. For SCCs with $|BPS| > 2$, the circuit

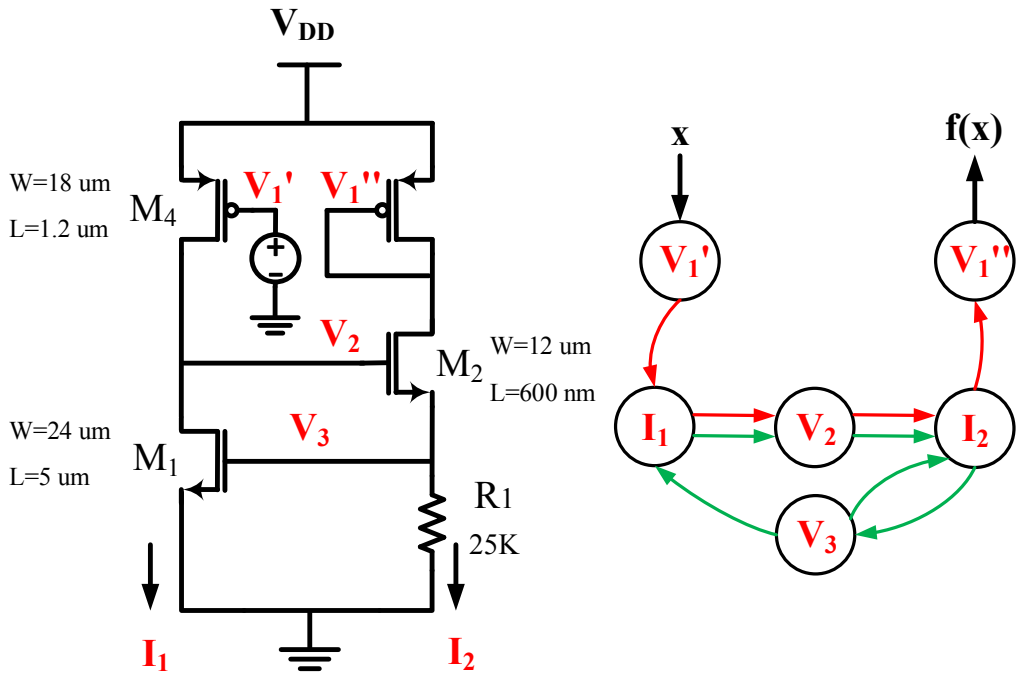


Figure 5.1: Break the Bootstrapped Vt Reference Circuit

has more than two PFLs coupled with each other, and at least three voltage sources have to be inserted and swept simultaneously, which is rare for practical circuits.

The verification of B1P circuits in a fixed setting (such as specified PVT, etc.) is a one dimensional problem. For example, the bootstrapped Vt reference circuit (as shown in Fig. 5.1) is a B1P circuit with one SCC and has multiple operating points at high temperatures. By breaking at the V_1 node and sweeping the inserted voltage source x, the output return function can be obtained, as shown in Fig. 5.1. From the break-loop homotopy method, DC operating points exist where the input $x = f(x)$. Except for the one dimensional problems, verification of B1P circuits with PVT variations or the B2P circuits are two dimensional problems.

5.2.2 Existence of Undesired Operating Points with Temperature Variation

Analog circuits should be designed to have only one operating point at a fixed working temperature, which can be guaranteed by the method proposed in [70]. However, the circuits may be locked into an undesired operating point with temperature variations. Fig. 5.2 shows the return functions of the bootstrapped Vt reference circuit at different temperatures. At

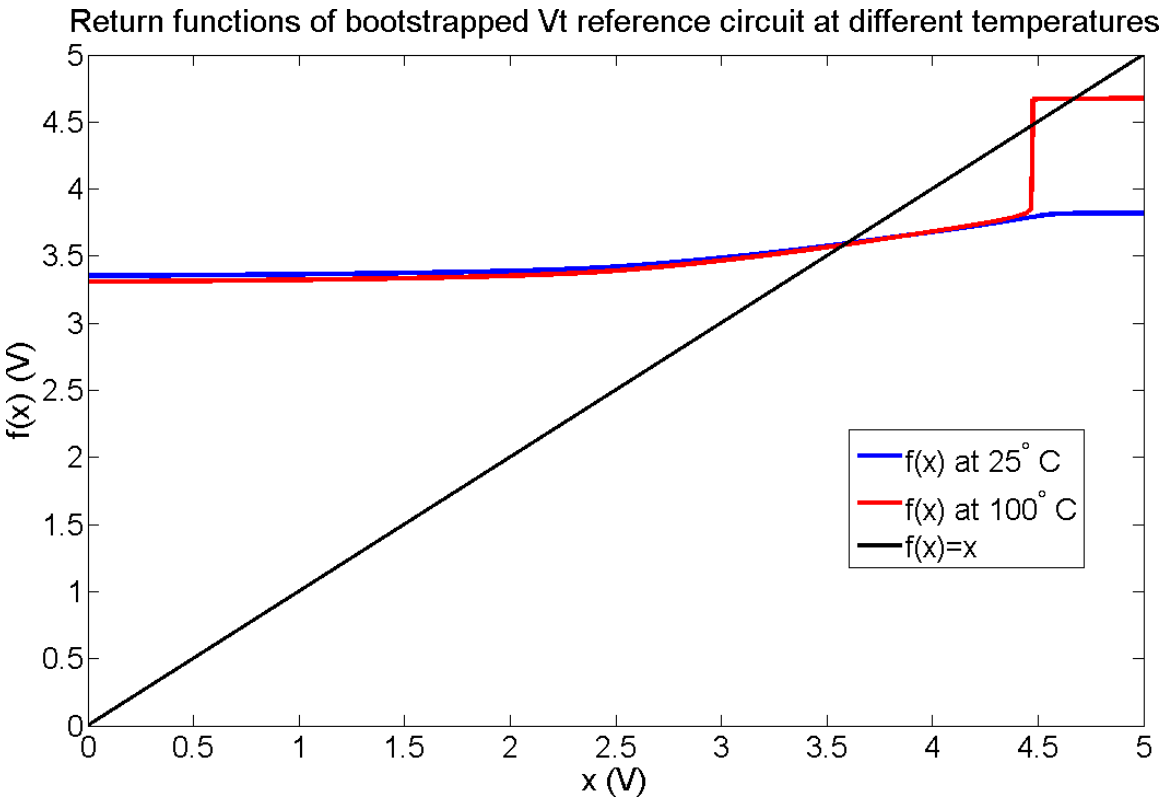


Figure 5.2: Return Functions of the Bootstrapped Vt Reference Circuit at Different Temperatures

a temperature of $25°C$, there is only one operating point in this circuit. However, there are multiple operating points when the temperature is $100°C$.

Therefore, methods are required to verify the existence of undesired operating points in analog circuits with temperature variations.

5.2.3 Existence of Undesired Operating Points with Voltage Supply Variation

In general, the voltage supply of circuits varies somewhat because of the load or environmental changes, and undesired operating points may exist with variation of the voltage supply. Using the bootstrapped Vt reference circuit as an example, its return functions at voltage supply 5V and 3V are shown in Fig. 5.3. In the circuit, undesired operating points do not exist at $VDD = 5V$, but they appear when the voltage supply drops to 3V. Therefore, an effective method of verifying the existence of undesired operating points with variation of the voltage supply is needed.

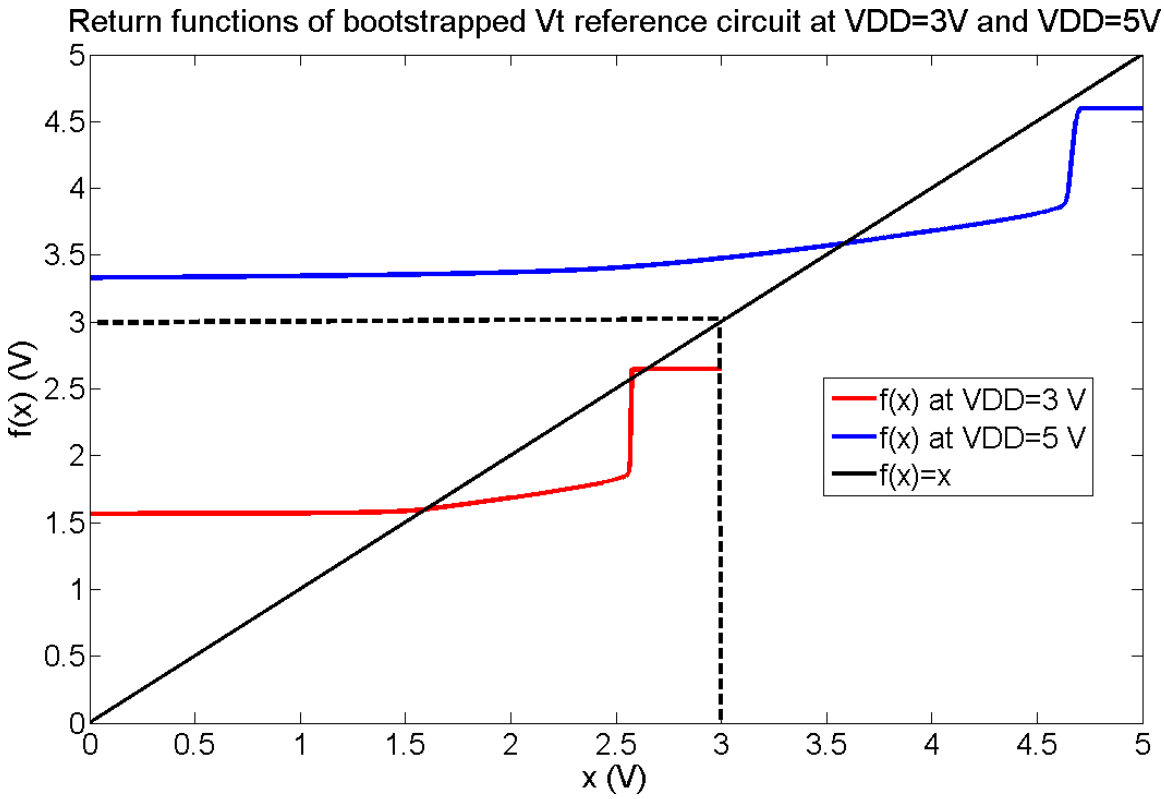


Figure 5.3: Return Functions of the Bootstrapped Vt Reference Circuit at VDD=3V and VDD=5V

5.2.4 Existence of Undesired Operating Points with Process Variations

Process variations are also important factors that affect the existence of undesired operating points. With a different process corner, the model parameters of the circuit components are changed; this alters the static I-V relationships in the circuits, which results in different operating points. Consequently, the existence of undesired operating points as a result of the process variations should be verified.

5.2.5 Existence of Undesired Operating Points with Transistor Sizing

In many designs, undesired operating points of a circuit can be eliminated by changing the transistor size. For example, by sizing the M_1 in the bootstrapped Vt reference circuit (as shown in the Fig. 5.4), the undesired operating point can be eliminated. As shown in Fig. 5.4,

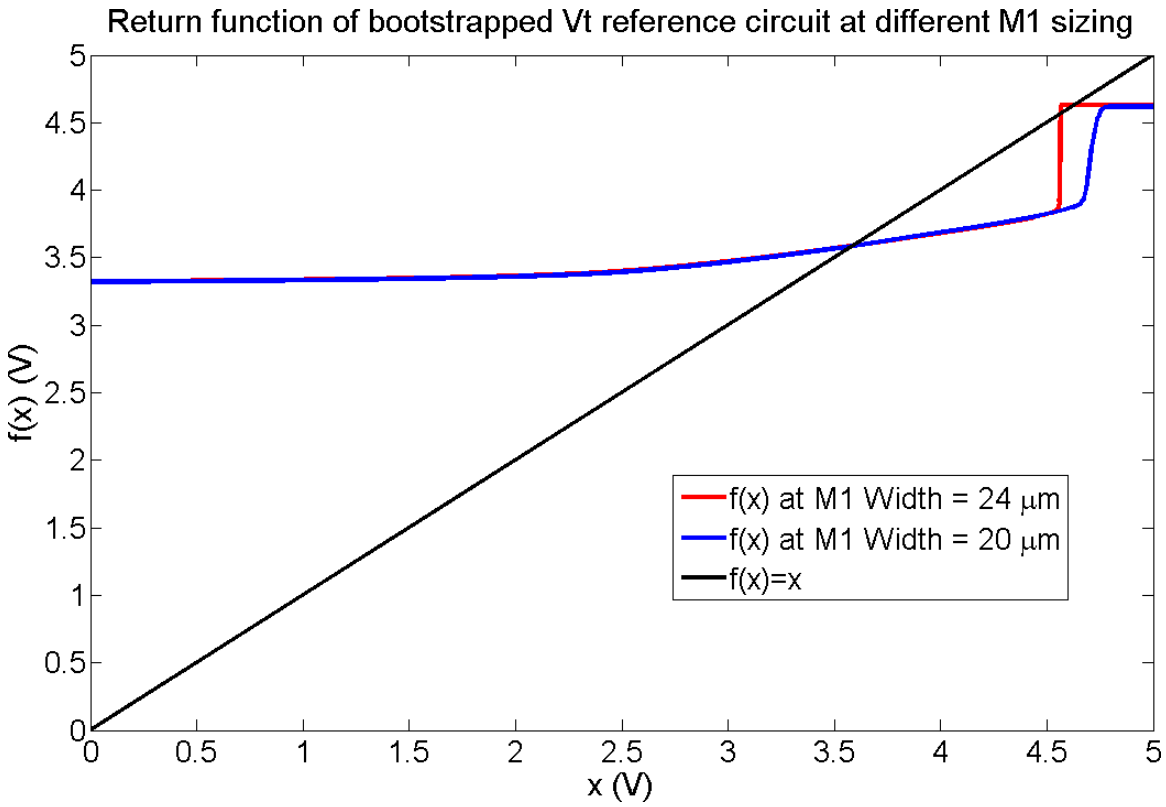


Figure 5.4: Return Functions of the Bootstrapped Vt Reference Circuit at Different M1 Sizing

if the width of the M_1 is reduced by half, there is only one operating point in the circuit when the temperature is 70°C and $VDD = 5V$. Therefore, verification of undesired operating points by changing transistor size provides great convenience to designers.

5.2.6 Existence of Undesired Operating Points in B2P Circuits

To verify a B2P circuit, two voltage sources should be inserted and swept simultaneously, so it is essentially a two dimensional problem. A typical B2P circuit is the Van Kessel-Banba circuit [73, 69] as shown in Fig. 5.5. In this circuit, the Van Kessel circuit is used to generate the bias current for the Banba bandgap circuit, which in turn provides the bias voltage for the Van Kessel circuit. Its corresponding circuit graph (DDG) seen in Fig. 5.6, shows that there are 8 PFLs and 6 NFLs in the circuit. To break all the PFLs, at least two break nodes should be broken, such as V_o and V_3. To verify such a type of circuit, a new verification method should be proposed.

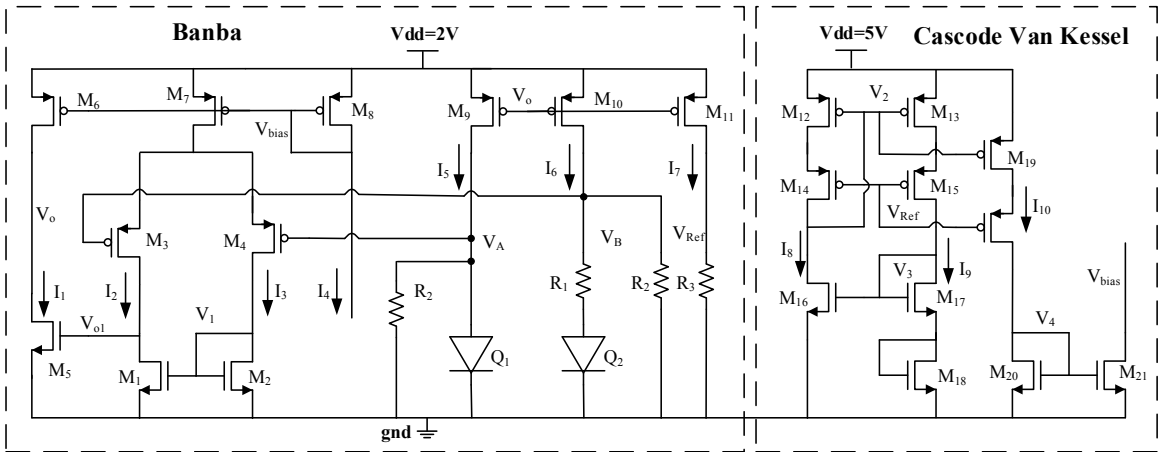


Figure 5.5: Van Kessel-Banba Circuit

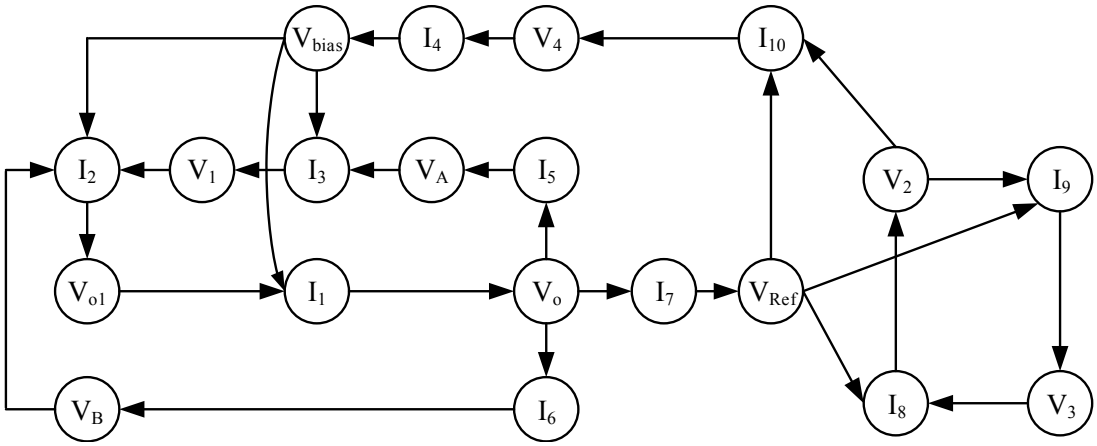


Figure 5.6: Circuit Graph of Van Kessel-Banba Circuit

5.3 Two Dimensional Vector Fields

In this section, the definition of a two dimensional vector field is given. After that, the two dimensional vector fields for continuous dynamic systems and discrete dynamic systems are discussed respectively.

In vector calculus, a two dimensional vector field is defined as a function whose input is a point in $\mathbb{R}^2$ and whose output is a vector (also in $\mathbb{R}^2$) emanating from the point [74]. For

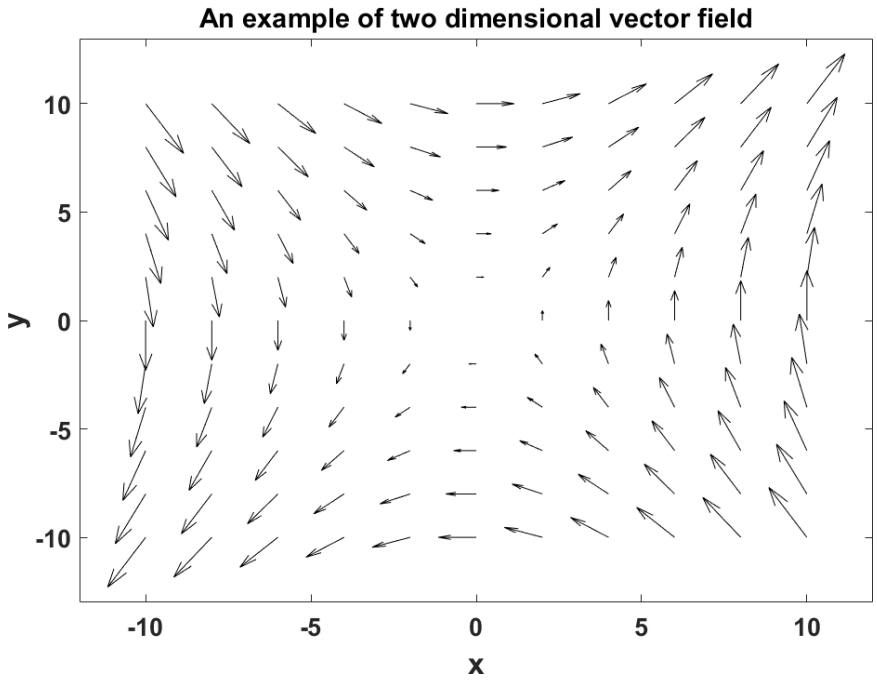


Figure 5.7: An Example of a Two Dimensional Vector Field

example, a two dimensional vector field can be written as,

$$\vec{F}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} x^2 + y \\ y^2 + 2x \end{bmatrix} \tag{5.1}$$

and it can be plotted as shown in Fig. 5.7.

A vector field shows a collection of arrows with a given magnitude and direction with each point in the plane. For example, each vector in the plane shown in Fig. 5.7 can stand for the speed and direction of a moving fluid through space, or the strength and direction of a magnetic or gravitational force, etc. A two dimensional vector field visualizes a moving fluid, or a magnetic or gravitational force.

Vector fields are applied in the analysis of dynamic systems, especially when the analytic expressions of the systems are unknown. For example, it is always difficult to obtain an expression of the speed and direction of every point in a moving fluid, but the speed and direction of each point in the fluid can be measured, so the moving fluid can be visualized by the model of a vector field. For each point in the vector field, the direction of its vector shows the direction of movement and the length of the vector indicates the speed. An equilibrium point exists where

the speed of the system equals zero, i.e., the length of the vector at that point is zero. In this way, the whole dynamic system is visualized by the vector field without solving its analytic expression.

For a two dimensional discrete dynamic system, it is defined as follows:

$$\begin{aligned} x_{k+1} &= x_k + g(x_k, y_k) \\ y_{k+1} &= y_k + h(x_k, y_k) \end{aligned} \tag{5.2}$$

where k stands for the kth iteration, and h and g are the iteration functions in the x and y directions, respectively. Thus, the relationship is as follows:

$$\vec{F}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} g(x,y) \\ h(x,y) \end{bmatrix} \tag{5.3}$$

An equilibrium point is where $g(x, y) = 0$ and $h(x, y) = 0$. Since $g(x, y)$ and $h(x, y)$ are smooth continuous functions, the directions and lengths of vectors gradually change from one point in the system to another. On that basis, equilibrium points can be approximately identified.

A simple example as shown in Fig. 5.8 can be used to intuitively illustrate how to find the equilibrium points in a two dimensional vector field of a discrete dynamic system. In the two dimensional interval $x = [-1, 1] \times y = [-1, 1]$, all the vectors point to the center ($x = 0$, $y = 0$). This indicates that the system will eventually move toward the center, which is an equilibrium point, regardless of the location of the initial start point in the two dimensional interval.

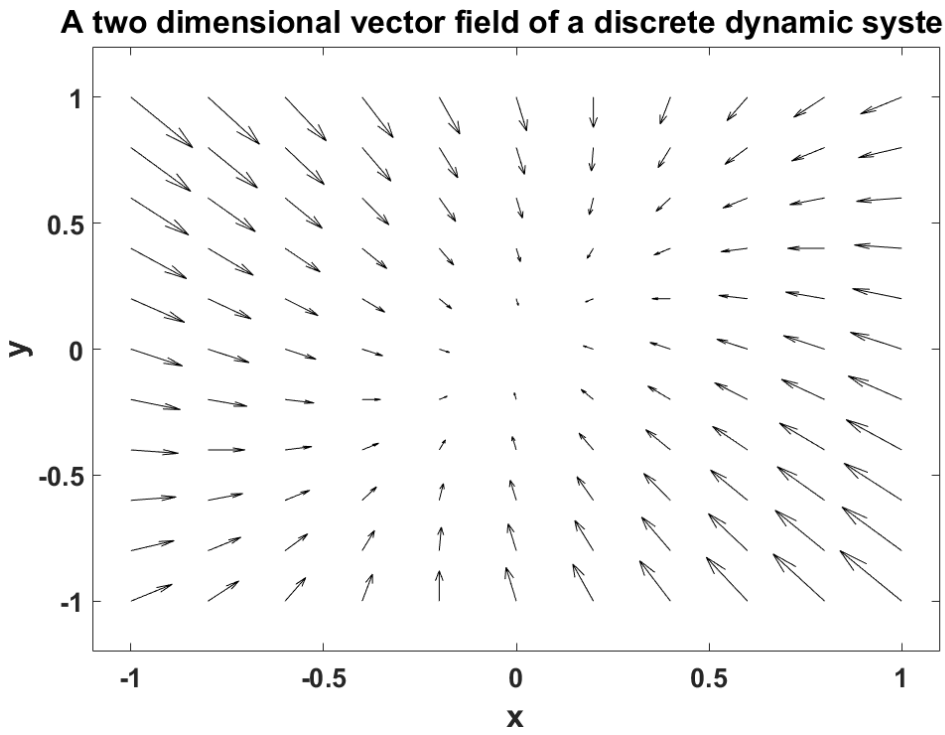


Figure 5.8: A Two Dimensional Vector Field of a Discrete Dynamic System

A two dimensional continuous dynamic system has the following relationship:

$$\dot{x} = g(x, y)$$
$$\dot{y} = h(x, y) \tag{5.4}$$

where $\dot{x}$ is the derivative of x, and h and g are vector functions in the x and y directions, respectively. Thus, a two dimensional vector field of a continuous dynamic system can be defined as,

$$\vec{F}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} g(x, y) \\ h(x, y) \end{bmatrix} \tag{5.5}$$

Similarly, $g(x, y) = 0$ and $h(x, y) = 0$ indicate an equilibrium point. The directions and lengths of vectors gradually change between two points in the system, since $g(x, y)$ and $h(x, y)$ are smooth continuous functions. Thus, equilibrium points can be approximately identified by observing the length and directions of vectors.

Fig. 5.9 shows a two dimensional vector field of a continuous dynamic system, where all the vector converge to the center ($x = 0$, $y = 0$). Thus, a point in the two dimensional interval

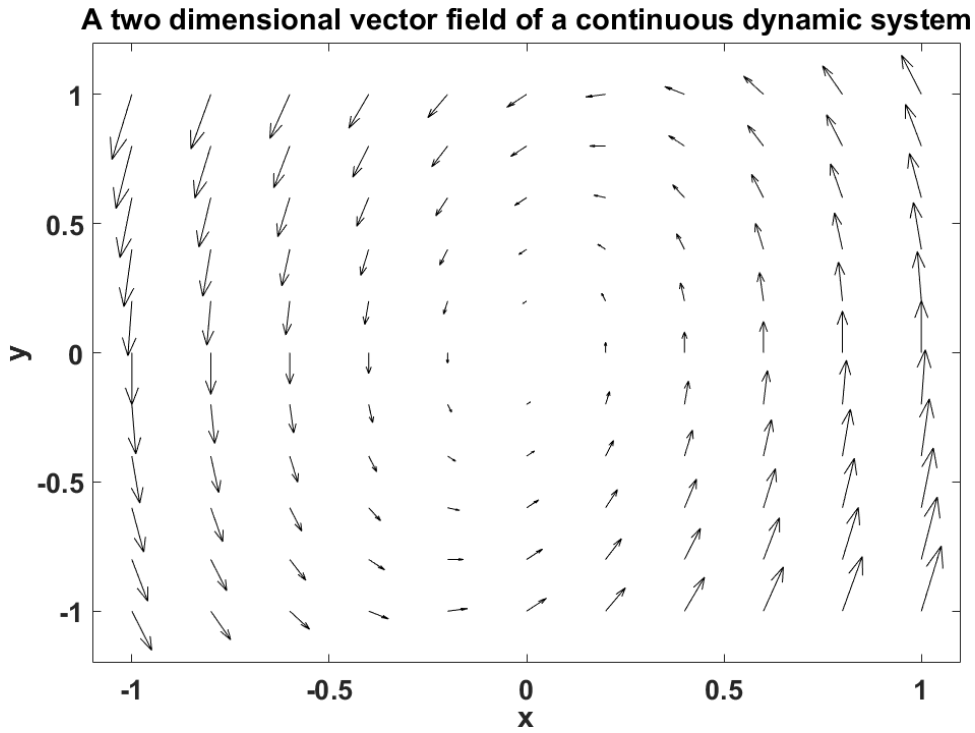


Figure 5.9: A Two Dimensional Vector Field of a Continuous Dynamic System

$x = [-1, 1] \times y = [-1, 1]$ cannot move to another point but eventually goes to the center, which is an equilibrium point.

5.4 Application of Two Dimensional Vector Fields to Analog Verification

The two dimensional verification problems mentioned in section 3 can be solved by use of two dimensional vector fields. The applied two dimensional vector fields are introduced in this section.

5.4.1 Two Dimensional Vector Fields for Temperature Verification

The break-loop homotopy method breaks all PFLs in the circuit and inserts a voltage source that can be swept to trace the operating points of a circuit [58, 6]. Since all the PFLs are broken, each input test signal x is related to exactly one output $f(x)$, so $f(x)$ is called the *return function* of the PFL [70]. Because seeking the operating point is equivalent to obtaining mathematical solutions of the circuit non-linear equations matrix, obtaining an analytic expression of the

return function $f(x)$ is formidable. However, for each test signal x, $f(x)$ can be evaluated by one DC simulation. Therefore, vector fields can be applied to find the characteristic of the return function and identify the existence of undesired operating points.

An operating point is where $f(x) - x = 0$, and an interval $[a, b]$ is a Sign-Change Interval (SCI) of the PFL if $(f(a) - a) \times (f(b) - b) < 0$ [70]. Since there is at least one operating point in an SCI [70], identification of two SCIs is equivalent to finding the existence of undesired operating points.

Therefore, to verify the existence of undesired operating points in circuits with temperature variation, a two dimensional vector field can be defined as follows:

$$\vec{F}\begin{pmatrix} x \\ T \end{pmatrix} = \begin{bmatrix} f(x) - x \\ 0 \end{bmatrix} \tag{5.6}$$

where x is the voltage of input voltage source, $f(x)$ is the corresponding return function, and T is the temperature. Since the y dimension of the vector is zero, all vectors are parallel to the x axis. SCIs can be identified by observing the vectors in the x direction. For example, with the temperature T_0 and input voltages a and b, if the direction of vector $\vec{F}\binom{a}{T_0}$ is different from $\vec{F}\binom{b}{T_0}$, this indicates that $(f(a) - a) \times (f(b) - b) < 0$, whereby an SCI is identified. Finding two SCIs at a certain temperature proves the existence of undesired operating points at that temperature. Therefore, the existence of undesired operating points with temperature variations is verified.

One example of a two dimensional vector field for temperature verification is shown in Fig. 5.10. The x axis is the input voltage and the y axis is the temperature. In Fig. 5.10, if the vector length is close to zero, it is marked as red or cyan; otherwise, it is shown as pink. For each temperature, observing two red/cyan regions separated by pink vectors indicates the existence of more than two operating points. Therefore, Fig. 5.10 clearly indicates the existence of more than two operating points in the temperature range from 84°C to 128°C.

5.4.2 Two Dimensional Vector Fields for Voltage Supply Variation

Similarly, another type of two dimensional vector field can be used to verify the existence of undesired operating points in circuits with voltage supply variation, and can be defined as

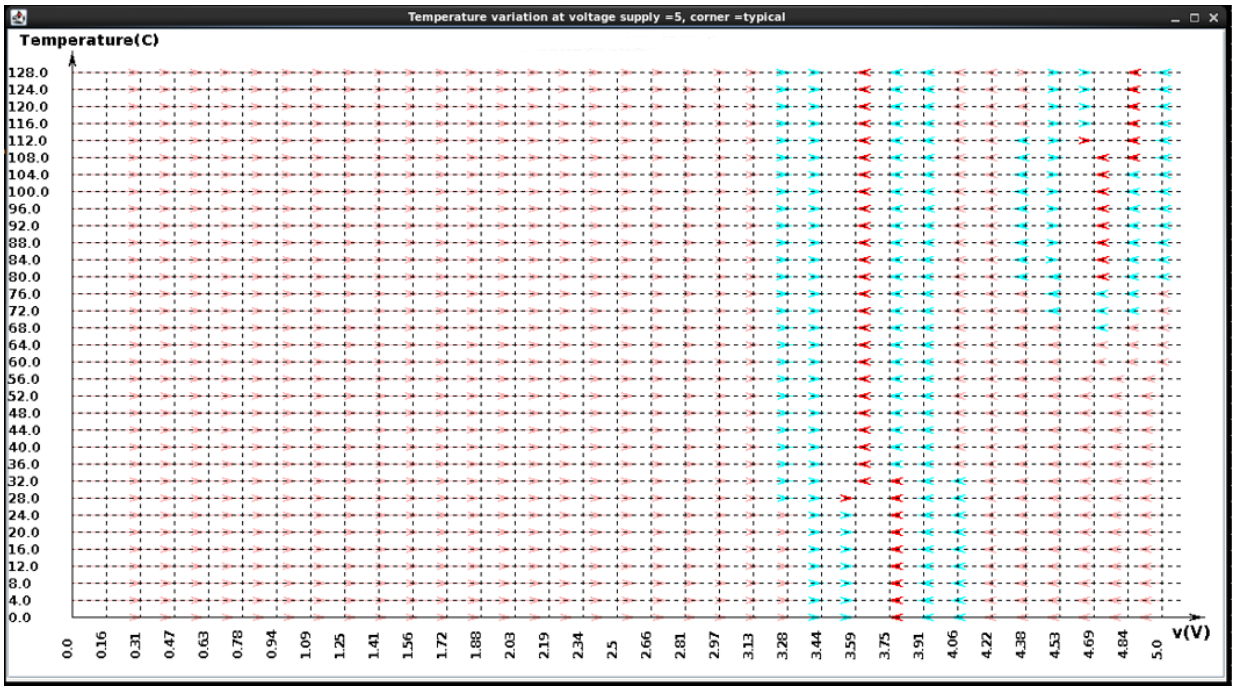


Figure 5.10: Bootstrapped Vt Reference Circuit's Two Dimensional Vector Field for Temperature Verification

following,

$$\vec{F}\begin{pmatrix} x \\ V \end{pmatrix} = \begin{bmatrix} f(x) - x \\ 0 \end{bmatrix} \tag{5.7}$$

where x is the voltage of input voltage source, $f(x)$ is the corresponding return function, and V is the voltage supply. Similarly, the y dimension of the vector is also zero, so all the vectors are parallel to the x axis and SCIs can be identified by observing the vectors in the x direction. With the voltage supply V_0 and input voltages a and b, if the direction of vector $\vec{F}\binom{a}{V_0}$ is different from that $\vec{F}\binom{b}{V_0}$, it shows that $(f(a) - a) \times (f(b) - b) < 0$, and an SCI is found. The existence of two SCIs in a voltage supply identifies the existence of undesired operating points. Therefore, the existence of undesired operating points with voltage supply variations can be verified.

5.4.3 Two Dimensional Vector Fields for Transistor Sizing

As mentioned earlier, great convenience is provided to designers if circuits can be verified according to transistor sizing. Thus, a third type of two dimensional vector field to identify the existence of undesired operating points in circuits with transistor sizing can be defined as:

$$\vec{F}\begin{pmatrix} x \\ S \end{pmatrix} = \begin{bmatrix} f(x) - x \\ 0 \end{bmatrix} \tag{5.8}$$

where x is the voltage of the input voltage source, $f(x)$ is the corresponding return function, and S is the size of the tuning transistor. Similarly, all the vectors are parallel to the x axis and SCIs can be identified by observing the vectors in the x direction. With the transistor size S_0 and input voltages a and b, if the direction of vector $\vec{F}\binom{a}{S_0}$ is different from that of $\vec{F}\binom{b}{V_0}$, $(f(a) - a) \times (f(b) - b) < 0$ and an SCI is found. In this way, the existence of undesired operating points with transistor sizing can be verified.

5.4.4 Two Dimensional Vector Fields for B2P Circuits

To verify B2P circuits, the fourth type of two dimensional vector field can be defined as follows:

$$\vec{F}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} f_1(x_1) - x_1 \\ f_2(x_2) - x_2 \end{bmatrix} \tag{5.9}$$

where x_1 is the voltage of the input voltage source at the first break node, $f_1(x_1)$ is the corresponding return function, x_2 is the voltage of the input voltage source at the second break node, and $f_2(x_2)$ is the corresponding return function. The vector has components in both the x and the y direction because of simultaneously sweeping two voltage sources.

An equilibrium point is the point at which the vector length equals zero. Since $f_1(x_1) - x_1$ and $f_2(x_2) - x_2$ are smooth continuous functions, the directions and lengths of the vectors gradually change from one point in the system to another. On that basis, equilibrium points of dynamic systems can be approximately identified from the vector directions. An equilibrium exists where the vector lengths equal zero, since this indicates that system will stay at that point.

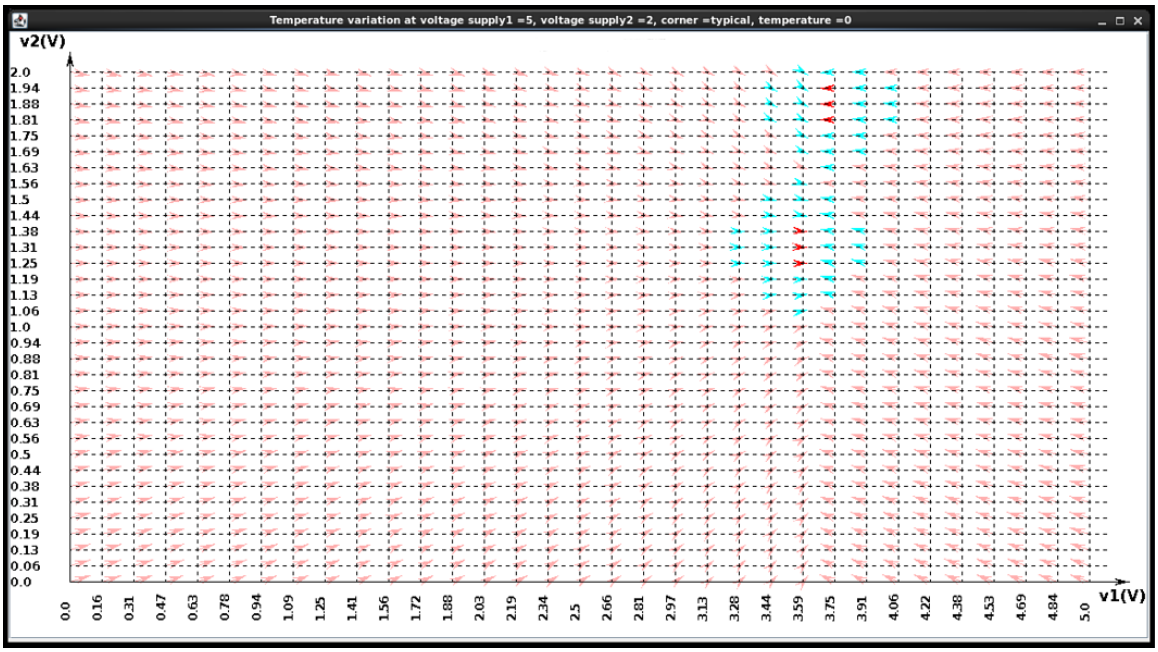


Figure 5.11: Van Kessel-Banba Circuit's Two Dimensional Vector Field

An example of a two dimensional vector field for a B2P circuit is shown in Fig. 5.11. The x axis is one inserted voltage source $V1$ and the y axis is the voltage source $V2$. Similar to Fig. 5.10, all the vector lengths are greatly reduced but their directions are unchanged. Those vectors with lengths close to zero are marked as red or cyan; other vectors are pink. If there are two disjoint red/cyan regions, this indicates the existence of undesired operating points. Fig. 5.10 shows that there are two operating points in $[3.59V, 3.91V] \times [1.81V, 1.94V]$ and $[3.44V, 3.75V] \times [1.25V, 1.38V]$.

5.5 Application Examples

A set of widely used benchmark circuits designed by use of $0.6\mu m$ CMOS technology, including bias generators, current/voltage references, temperature sensors, and op-amps [64], has been verified. Some have been selected as examples to demonstrate the proposed two dimensional vector field method shown in this section.

5.5.1 Analog Verification with PVT Variation

Example 1: Consider one of the benchmark circuits—the bootstrapped Vt reference circuit and its circuit graph (as shown in the Fig. 4.11.

To verify this B1P circuit with PVT variation, we applied the two dimensional vector field method. When the circuit works at voltage supply = 5V and typical corner, its two dimensional vector field is as shown in Fig. 5.10, which clearly indicates the existence of more than two DC operating points in the temperature range from 84°C to 128°C. The PVT verification results are shown in TABLE 5.1. The temperature range to be verified is from 0°C to 128°C, and several voltage supplies and process corners are verified. With the variation of process corners and voltage supplies, the temperature ranges associated with undesired operating points change. Therefore, the two dimensional vector fields can tell the designers if undesired operating points exist in the circuits as PVT variations occur.

Table 5.1: Simulation Results of Bootstrapped Vt Reference With PVT Variations

Temperature range with undesired OP	VDD=4V	VDD=5V	VDD=6V
typical corner	92°C ∼ 128°C	84°C ∼ 128°C	76°C ∼ 128°C
fast corner	84°C ∼ 128°C	92°C ∼ 128°C	72°C ∼ 128°C
slow corner	76°C ∼ 128°C	88°C ∼ 128°C	70°C ∼ 128°C

5.5.2 Analog Verification with Two Dimensional Circuits

Example 2: The Van Kessel-Banba circuit [73, 69] is a typical B2P circuit, as shown in Fig. 5.5. By breaking at both node V_o and V_3, all the PFLs in the circuit can be cut off, as shown in Fig. 5.13. Its circuit implementation is shown in Fig. 5.12. We can apply our two dimensional vector field to it. The simulation result in the typical corner is shown in Fig. 5.11; V1 is the voltage of the V_o node and V2 is the voltage of the V_3 node. This indicates that two

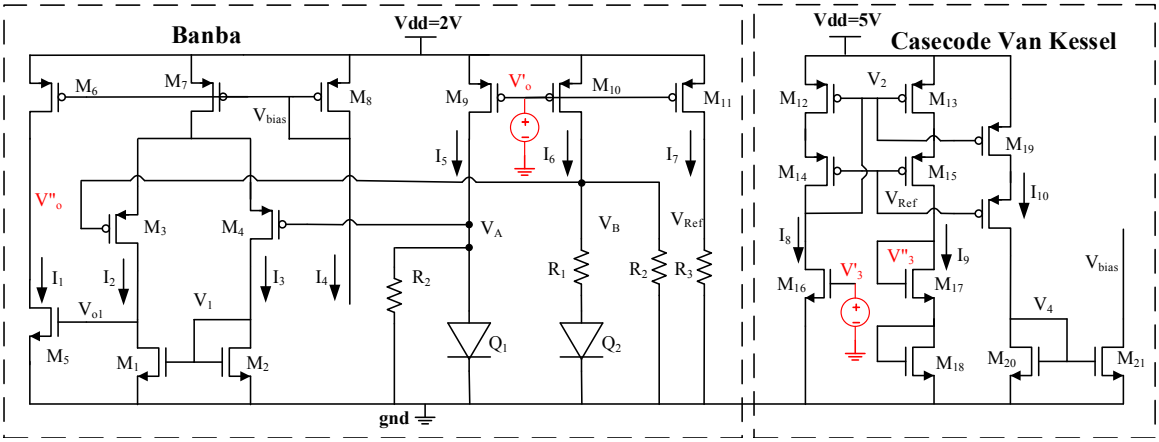


Figure 5.12: Implementation of Breaking Loop of the Van Kessel-Banba Circuit

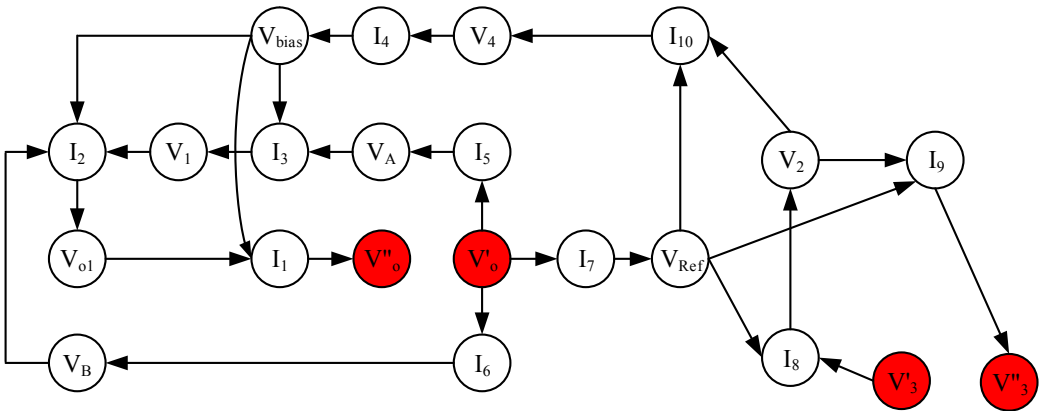


Figure 5.13: Circuit Graph of Breaking Loop of the Van Kessel-Banba Circuit

operating points exist, at $[3.59V, 3.91V] \times [1.81V, 1.94V]$ and $[3.44V, 3.75V] \times [1.25V, 1.38V]$. The simulation results are summarized in TABLE 5.2.

5.6 Conclusion

The existence of undesired operating points is an important problem in the design of many analog circuits, such as bias generators, current references, temperature sensors, and Bandgap references. In this chapter, we focus on the two dimensional analog verification against undesired operating points, i.e., to verify circuits with the PVT variations, transistor sizing, or B2P circuits. For this type of verification, a two dimensional vector field method is proposed that can effectively identify the existence of undesired operating point by visualizing the vector field

Table 5.2: Simulation Results of the Van Kessel-Banba Circuit

VDD1	VDD2	temperature	have undesired OP?	Solution Intervals for Nodes V_o and V_3
5V	2V	0°C	YES	$[3.59V, 3.91V] \times [1.81V, 1.94V]$ $[3.44V, 3.75V] \times [1.25V, 1.38V]$

of return functions. Simulation results show the proposed method to be effective in identifying undesired operating points in a set of commonly used benchmark circuits, including bias generators, voltage references, temperature sensors, and op-amps.

CHAPTER 6. ITV: A NEW VERIFICATION TOOL TO IDENTIFY UNDESIRED OPERATING POINTS IN ANALOG AND MIXED-SIGNAL CIRCUITS

6.1 Introduction

It is well known that undesired operating points exist in many analog circuits, such as bias generators, current references, temperature sensors, and Bandgap references. In addition, various feedback approaches, e.g., self-biasing, bootstrapping, self-stabilization, and digitally-assisted-analog, are applied in designing analog circuits to enhance their performance. Inevitably, these methods result in structures with one or more feedback loops that make them vulnerable to the presence of undesired operating points. Circuits with undetected operating points can have devastating results, particularly when employed in critical systems such as automotive, health care, and military products. However, there is no reliable approach usable with currently available commercial simulators to find unexpected multiple operating points [8]. Recognizing the existence of undesired operating points largely depends on the designer's experience. The most popular technique for removing undesired operating points is to use start-up circuits, but sometimes, because of unanticipated transient and variations in operating conditions, a circuit equipped with start-up circuits may still enter an unknown operating point. In some circuits, even the most experienced designers are not aware that a circuit they have designed has undesired operating point potential, because such operating points often go undetected in the standard simulations used in the design process.

In chapter 4, an efficient analog verification method against undesired operating points is systematically proposed. In contrast to traditional methods used to discover operating points, this method does not try to find all operating points or even any operating point at all. It

only targets finding voltage intervals that contain undesired operating points, thereby verifying existence of such operating points. This essentially replaces a difficult root-finding problem with simply identifying the existence of test intervals that include undesired operating points, which is why it is efficient and dramatically reduces the computational requirements. However, the method mainly focuses on the one dimensional analog verification problems, i.e., the verification in specific transistor size, fixed process/voltage supply/temperature (PVT) setting and one input voltage test source to break all the PFLs (B1P circuits).

To solve the problem of two dimensional analog verification against undesired operating points, i.e., to verify circuits with the PVT variations, transistor sizing, or complicated circuits whose PFLs cannot be broken at only one node (B2P circuits), a two dimensional vector field method is proposed in chapter 5. It can effective identify the existence of undesired operating points by visualizing vector fields of return functions.

Based on these two verification methods, a verification tool called "ITV" is proposed in this chapter to identify undesired operating points in analog and mixed-signal circuits. ITV automatically converts a circuit netlist to a Directed Dependency Graph (DDG), partitions the DDG into Strong Connected Components (SCCs) and finds all the PFLs and NFLs in each SCC. It then identifies the break points to break all the PFLs in each SCC. For one dimensional verification, it applies the proposed divide and contraction algorithms to determine the presence/absence of undesired operating points. For a two dimensional verification, the two dimensional vector field method is applied to identify the existence of undesired operating points.

This chapter is organized as follows. In section 2, the verification flow is illustrated; the tool implementation is provided in section 3. The ITV usage is discussed in section 4; section 5 introduces ITV installation, program files and demo circuits; Finally, conclusions are drawn in section 6.

6.2 Proposed Verification Flow

In this section, the proposed verification flow used in ITV is introduced. Since ITV partitions a circuit graph (DDG) into SCCs, the SCC is introduced first. According to the dimension

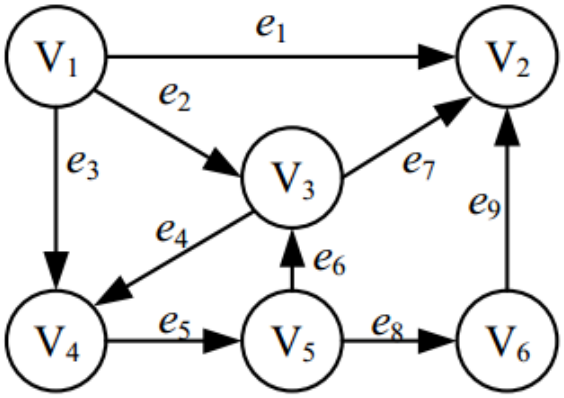


Figure 6.1: An Example of SCC & BPSet

of the verification problems, different verification methods are applied, so classification of one and second dimensional problems are discussed later. Finally, the whole verification flow is proposed.

6.2.1 Strongly Connected Component (SCC)

A Strongly Connected Component (SCC) H of a digraph G, is a directed subgraph of G such that for every pair of vertices u and v in H, there is a directed $u - v$ path and also a directed $v - u$ path in H [66]. Any digraph can be partitioned into a set of disjoint SCCs. For example, the digraph corresponding to the node-link diagram in Fig.6.1 has four disjoint SCCs: $H_1 : (V_1, \varnothing), H_2 : (V_2, \varnothing), H_3 : (V_6, \varnothing), H_4 : (V_3, V_4, V_5, e_4, e_5, e_6)$.

The motivation to partition a DDG into SCCs is to partition the original circuit into several sub-circuits that can be separately verified. This significantly enhances the efficiency of the break-loop homotopy method for large-scale circuits, since the break-loop method can be applied to each SCC independently [68].

One or multiple PFLs may exist in an SCC, and a break point set (BPS) of an SCC is used to represent a minimum subset of controlling voltage nodes that will break all the PFLs if removed from the SCC. With existing algorithms in graph theory [71, 72], the BPS of an SCC can be found with high efficiency.

It is assumed that the size of BPS F for each SCC satisfies $|F| \leq 2$ where $|F|$ is the size of BPS. This assumption is reasonable, because $|F| > 2$ means that a circuit has more than two

PFLs coupled to each other, which is rare for practical circuits. Furthermore, $|F| > 2$ means that, for the break-loop continuation method, at least three voltage sources have to be inserted and swept simultaneously, which is inefficient.

6.2.2 Dimensions of the Verification Problems

As mentioned in Chapter 5, the divide and contraction verification method proposed in [70] mainly focuses on a one dimensional problem, such as a fixed PVT setting for B1P circuits. In practice, designers also need to verify circuits with the transistor sizing, PVT variations or identify the existence of undesired operating points in B2P circuits. This type of problem is called two dimensional analog verification against undesired operating points. For this type of verification, the two dimensional vector field method is applied, which can effectively identify the existence of undesired operating points by visualizing the return functions in the circuits.

6.2.3 Verification Flow

In summary, the ITV verification is shown in Fig.6.2 as consisting of the following steps: first, a graphical representation of the circuit (DDG) is obtained that can be used to identify all structural PFLs of a circuit. From this graphical representation, all the SCCs are identified. For each SCC, a minimal set of break points (BPS) is determined that can be used to break all PFLs in the circuit. If the size of BPS is greater than two, i.e., $|F| > 2$, ITV reports that the circuit is too complicated to verify and exits; otherwise, it starts the verification process. For one dimensional verification, the divide and contraction algorithms could be applied to identify the undesired operating point. The two dimensional vector field methods are used to for the two dimensional verification.

Each SCC is verified, one by one, until any of them is proven to have undesired operating points; otherwise, ITV reports that the circuit has only one operating point.

6.3 Tool Implementation

To implement the proposed verification flow, a new EDA tool was developed, the implementation of which is introduced in this section.

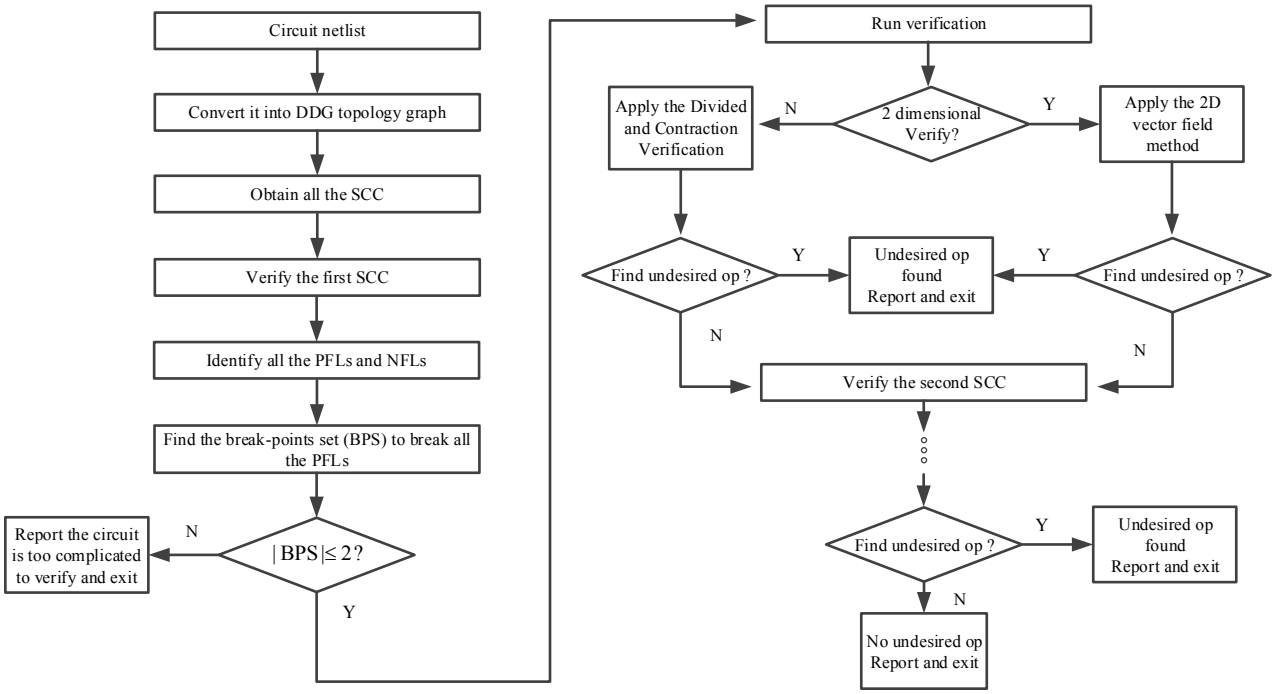


Figure 6.2: Flow of Proposed Verification Method

6.3.1 ITV Flow

The ITV flow, shown in Fig. 6.3, starts from user interface, which is built into the Cadence schematic editor. From the user interface, ITV can export the netlist of the schematic to be verified. ITV converts the generated netlist to DDG and identifies the resulting SCCs. Then, general graphic algorithms are employed to find BPS in the selected SCC. In the found BPS, users can choose breaking point(s) to break all the PFLs in an SCC. After that, ITV automatically generates a new netlist, which breaks all the PFLs at the selected break point(s) and inserts the test voltage source(s). With the new netlist and verification settings, ITV decides if it is a one dimensional verification. If so, it verifies the circuits by the divide and contraction algorithm program; Otherwise, the two dimensional vector field method is applied to identify the existence of undesired operating points. If undesired operating points are discovered or all SCCs have been verified, it reports the verification results and exits. If there are more SCCs to be verified and no undesired operating points are detected in the current SCC, ITV goes to the second SCC and repeats the previous steps in that SCC. The iteration continues until all the

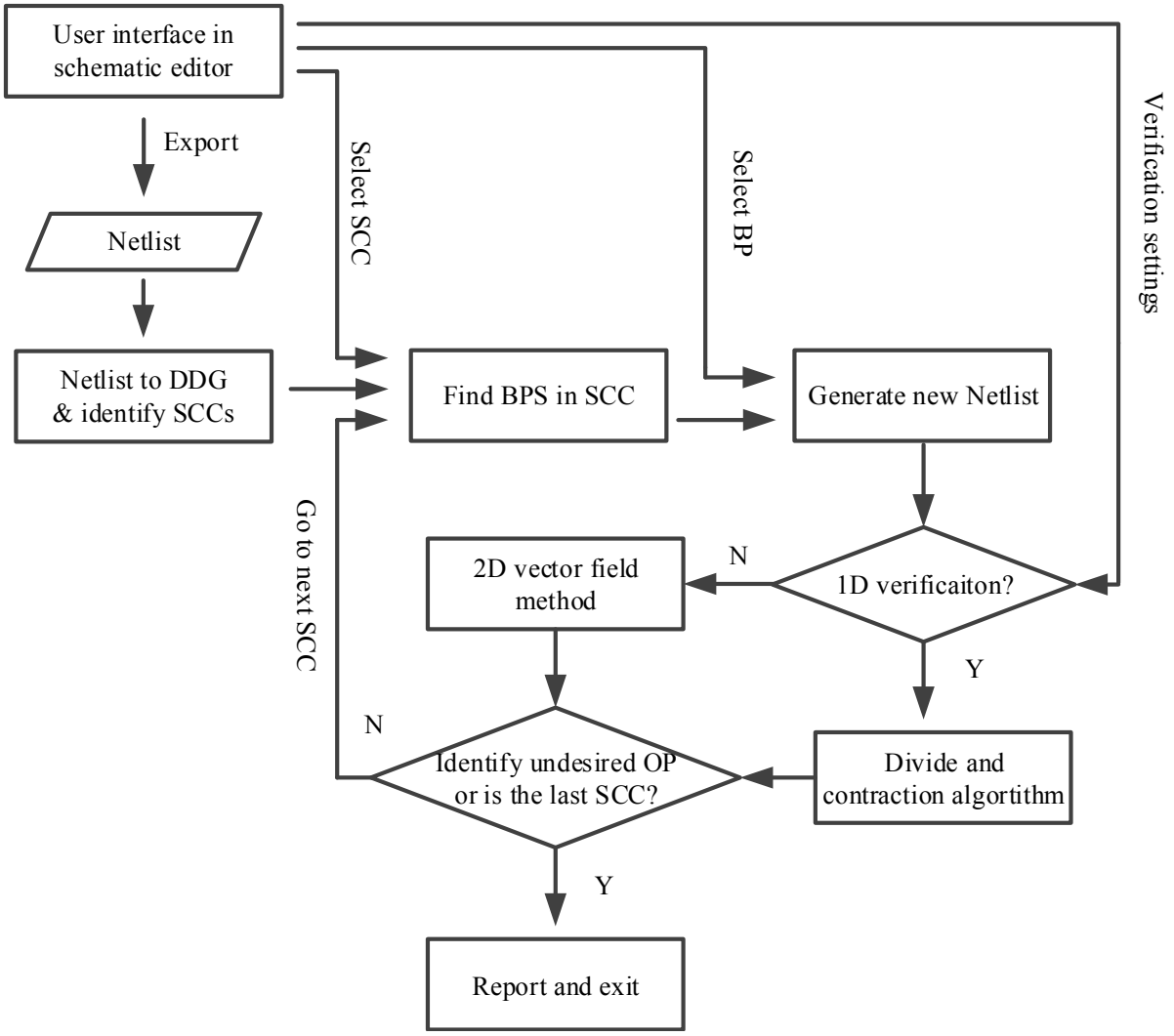


Figure 6.3: ITV Verification Flow

SCCs are verified without detecting any undesired operating points, or exits when undesired operating points found in any SCC.

6.3.2 User Interface

To make ITV a convenient tool, a user interface has been developed in the widely used Cadence analog circuit design environment, using the SKILL language. It is integrated to the menu of the schematic editor, as shown in Fig. 6.4. ITV can be launched by clicking the "check → ITV Verification" button. More details about the user interface are given in section 6.

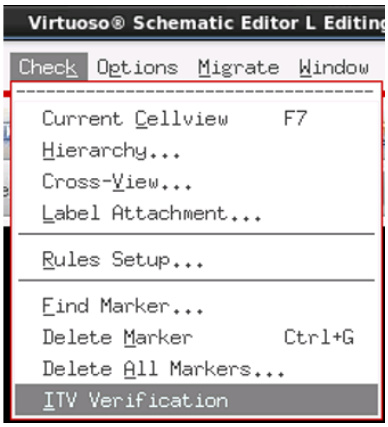


Figure 6.4: ITV Menu Item in Cadence Virtuoso Schematic Editor

6.3.3 Implementation of Loop Identification and Break

ITV automatically converts the export netlist to a DDG. First, it searches all the branch-current between power supply and ground nodes, which is similar to finding all paths between two nodes in a graph [75]. Therefore, the depth-first search algorithm [76] can be directly applied. Then, the controlling voltages and the dependencies between the branch-current and controlling voltage can be identified by analyzing the relationships between the current flowing in each branch and the gate-source connections of each transistor [68]. Thus, the DDG of the circuit can be obtained from the netlist.

To decompose a DDG into strongly connected components, ITV uses Tarjan's algorithm [77], which has a $O(n+e)$ running time, where there are n vertices, e edges in the input graph.

Detecting feedback loops for SCC can also be achieved by utilizing standard graph theory techniques. ITV adopted Johnson's method, described in [78], which can find all the feedback loops of a graph in time bounded by $O((n+e)(c+1)$ and space bounded by $O(n+e)$, where c is the number of feedback loops. The sign of the feedback loops can be automatically identified by the method described in [63].

The next step is to find BPS to break all the PFLs in each SCC, which is a classic NP-complete problem [71]. However, it is relatively easy to solve with our application, for the following reasons. First, the DDG has been partitioned into SCCs such that each SCC is small. Moreover, it is assumed that the size of BPS for each SCC satisfy $|F| \leq 2$, where $|F|$ is the size of BPS. Therefore, ITV applies the algorithm in [72] to find the BPS for each SCC.

After selecting the break points, ITV applies a script to modify the original netlist. It automatically breaks at the break point(s), and inserts the test voltage source(s) into the netlist.

6.3.4 Implementation of Divide and Contraction Algorithms

The divide and contraction algorithms have been implemented in a script program. Each iteration of the algorithms calls for the circuit simulator to complete the DC simulations, utilizing the generated netlist from the previous step as an input to the simulator. It also configures the simulation settings from the user interface, such as settings for the process corner, voltage supply, temperature, etc. After the simulation, it collects the simulation results and analyzes them for the existence of undesired operating points. Based on the analysis, it continues the iteration or reports the existence of undesired operating points.

6.3.5 Implementation of the Two Dimensional Vector Field Method

Similarly, the two dimensional vector field method is implemented in a script program file. Based on the netlist generated in the previous step and the simulation settings from the user interface, it calls for the circuit simulator to run the DC simulations. It records all the DC simulation results of the return functions. A plot program developed by Java is then utilized to plot the two dimensional vector fields. Depending on the verification settings, it may plot one or multiple two dimensional vector fields. By analyzing them, the user can conclude whether or not undesired operating points exist in the circuit.

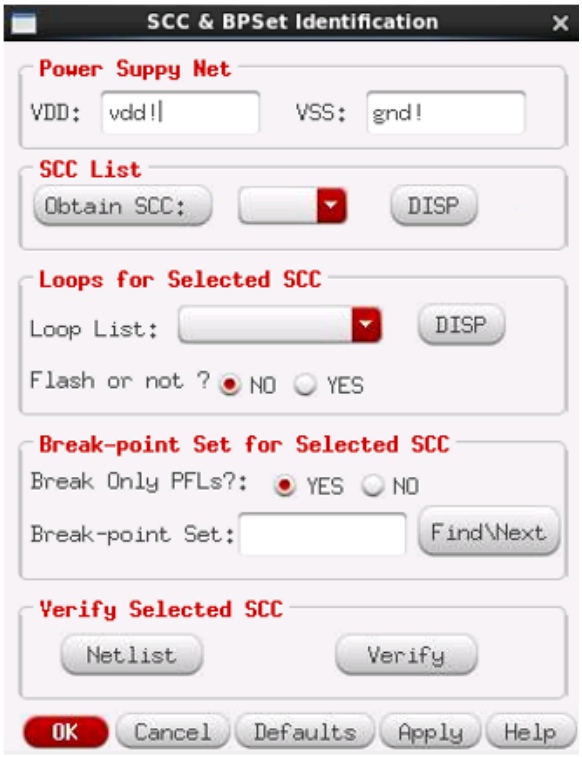


Figure 6.5: SCC & BPSet Identification Form

6.4 ITV Usage

In this section, the usage of ITV is introduced in detail. The loop identification and break functions are first illustrated, after which the run verification settings are given. By reading this section, readers can have the basic knowledge necessary to use this tool for their analog circuits verification.

6.4.1 Loop Identification and Break

The first step in ITV verification flow is to identify all the PFLs and determine the break point. Those functions are implemented as an "SCC & BPSet Identification" form, shown in Fig. 6.5, which is implemented by SKILL.

In this subsection, each function included in the "SCC & BPSet Identification" form is illustrated. After knowing these functions, readers are able to use the identify and break loops functions in ITV.

6.4.1.1 Construction of DDG and Obtaining SCC

First, ITV automatically converts a MOS circuit into a DDG, using the systematic approach proposed in [65, 70]. For a MOSFET, the drain-source is considered as a channel that conducts current flowing in a branch from Vdd to gnd (which is defined as a "branch-current" in [65]). The voltage of any net that controls the gate-source voltage of any MOSFET is defined as a "controlling voltage." Following that, dependencies between the controlling voltages and branch-currents are identified. As a result, the controlling voltages and branch-currents form vertices of the DDG and the dependencies between them form edges of the DDG.

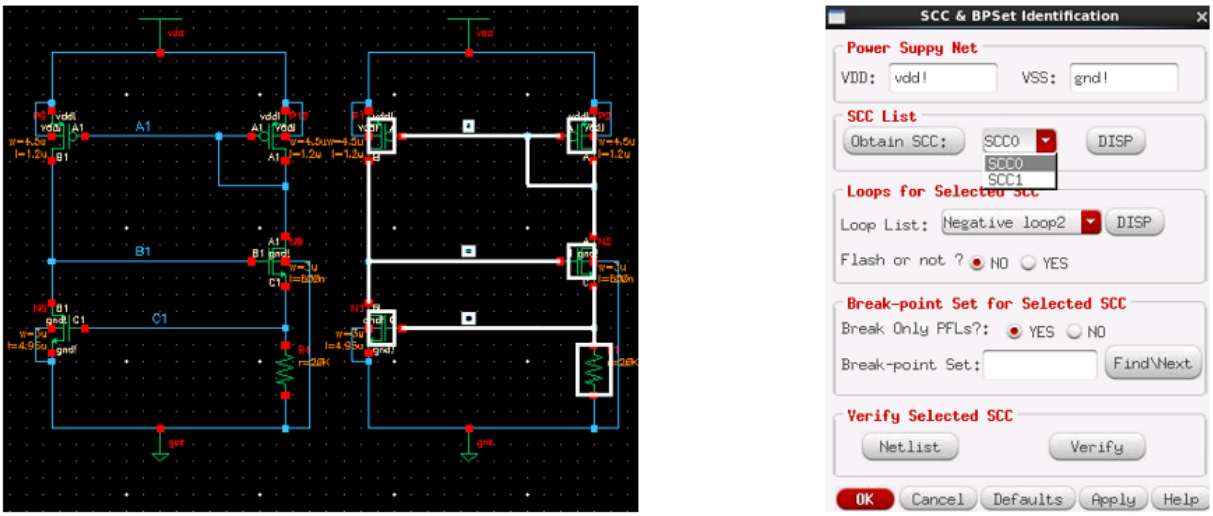
To identify the branch current in a circuit, the node names of Vdd to gnd should be given to ITV. That is why the user needs to input the VDD and VSS node names in the "Power Supply Net" frame, as shown in Fig. 6.5. ITV supports multiple power supplies, separating their names with spaces. With the name of the power supply, ITV automatically analyzes the circuit's netlist, and identifies all the branch-currents, controlling voltages and their dependency relationships.

Then, ITV employs the algorithm from [77] to compose the generated DDG into SCCs. The result is shown in the "SCC List" frame field. By clicking the button "Obtain SCC", the SCC list will appear, as shown in Fig. 6.6. Clicking "DISP" after choosing the SCC causes all the SCC associated circuit components to be highlighted, and all loops in the SCC are shown in the "Loops for Selected SCC" frame. For example, the SCC0 is highlighted in Fig. 6.6.

6.4.1.2 Identify Feedback Loops in Circuit

Since DDG is given in the previous step, all feedback loops in SCCs can be automatically identified by ITV. Based on the sign definitions of dependency relationship, the sign of each feedback loop can be determined.

The "Loops for Selected SCC" frame allows the user to highlight the loops (for the selected SCC in previous step). Positive feedback loops and negative feedback loops will be highlighted in different colors as shown in Fig.6.7.



(a) (b)

Figure 6.6: Obtain SCC: (a)Schematic with SCC0 Highlighted; (b)Highlight SCC0 From the "SCC & BPSet Identification" Form

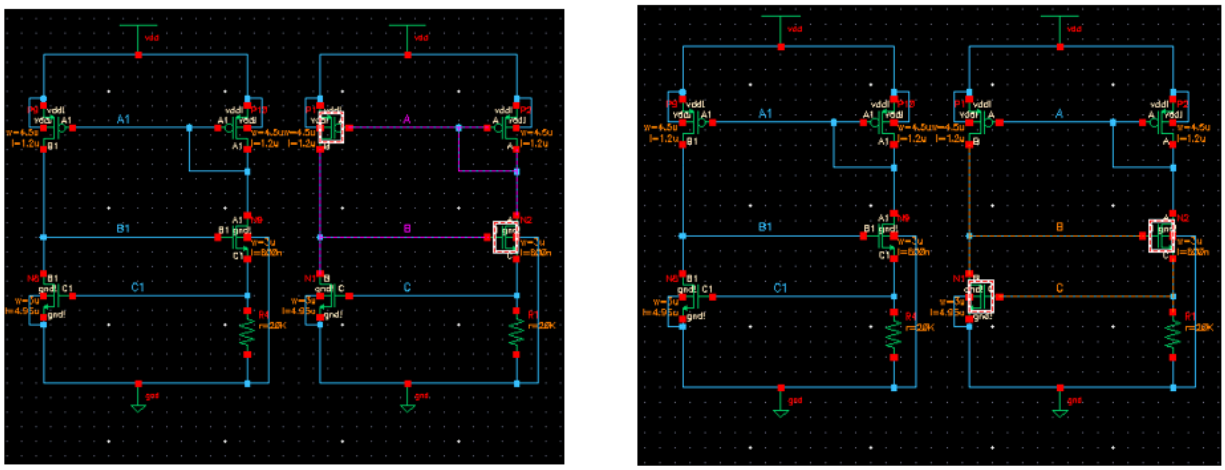
In the second row of the frame, there is a click button "Flash or not?". If "No" is chosen, the selected loop is highlighted after the "DISP" button is clicked. If "YES" is chosen, each circuit component in the loop is highlighted in sequence to show the loop more clearly.

6.4.1.3 Select Break Points

After the previous steps have been followed, all the feedback loops in each SCC and their signs have been identified, and the algorithm in [72] is applied to find the BPS for each SCC.

In the "Break-point Set for Selected SCC" frame, there is an option "Break Only PFLs?" as shown in Fig.6.8 (a). The default value is "Yes" and it means that ITV will try to obtain a break point set (BPS) to break all PFLs and avoid breaking any NFLs. Clicking "Find\Next" button will produce two results:

❑ If it can find such BPS, the BPS will be shown in the "Break-point Set" field. If more than two such BPS are found, the user could choose either of them by clicking the "Find\Next" button. For one dimensional verification, such kinds of BPS result in monotonic return function of PFLs, which could be verified with higher efficient divide and contraction algorithms.



(a) (b)

Figure 6.7: Highlight Feedback Loops: (a) PFL of SCC0; (b) NFL of SCC0

❑ If no such BPS is found, ITV gives a warning as shown in Fig.6.8 (b). Then the user should choose "No" as shown in Fig.6.8 (c). By clicking "Find\Next" button, the user can selects break point(s) from the BPS to break all PFLs but also break some NFLs. In this case, the return function is non-monotonic.

6.4.1.4 Generate Netlist

In the "Verify Selected SCC" frame as shown in Fig.6.8(c), there are two buttons: "Netlist" and "Verify," which should be clicked in sequence. First, clicking the button "Netlist", a new netlist with PFLs break at selected points is generated, which is used as an input file for verification. Then, clicking the "Verify" button brings the "Run Verification Setting" form, which will be discussed in the next sub-session.

6.4.2 Run Verification Settings

After breaking all the PFLs in each SCC, ITV applies the divide and contraction algorithms for one dimensional analog verification to identify the undesired operating point. For two dimensional verification, two dimensional vector field methods are used. Their usage and functions are illustrated in this sub-section.

(a) (b) (c)

Figure 6.8: Select Break Points: (a)Only Break PFLs; (b)The Warning for "Can't Break Only PFLs"; (c)Break Both PFL and NFL

6.4.2.1 One Dimensional Verification

For one dimensional verification, its "Run Verification Settings" form is given in ITV, as shown in Fig.6.9. Its settings and choices are explained as follows:

In the *"Basic Options"* frame, the user could choose the error tolerance for algorithm, and the specific environment settings.

✓ *"Temperature"*: set environment temperature in Celsius.

✓ *"VDD"*: set the voltage supply for the break point. It is worth mentioning that there may be different VDDs in the circuit and this field only specifies the VDD for the break point in the circuit.

✓ *"Process Corner"*: set the process corner name for verification.

✓ *"Error Tolerance"*: set the error tolerance for our divide and contraction algorithm. A smaller error tolerance could provide a more accurate solution but would require more time. It is only useful in one dimensional verification.

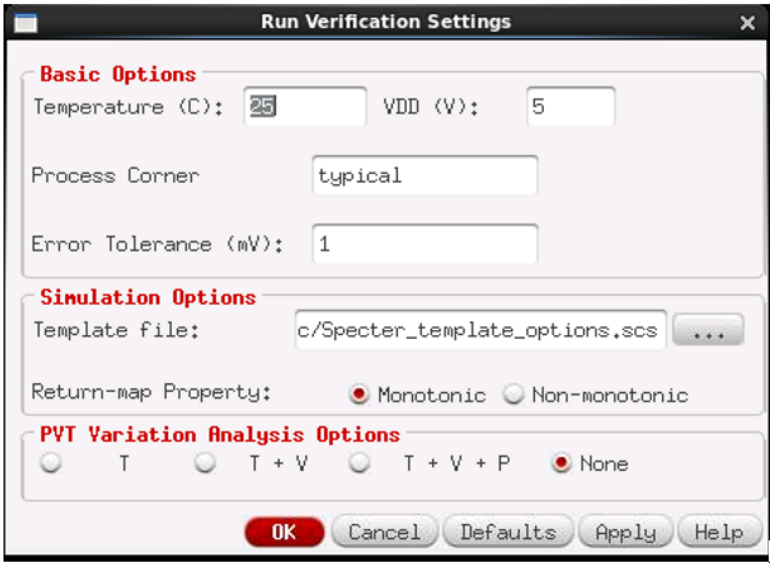


Figure 6.9: ITV "Run Verification Settings" Form for B1P Circuits Stand-alone Verification

With these settings, ITV applies the divide and contraction verification methods to verify the existence of undesired operating points in the circuit. After verification, results are reported to the user.

6.4.2.2 PVT Variation Verification for B1P circuits

By use of the two dimensional vector field method, ITV can find the presence/absence of undesired operating points in B1P circuits with PVT variations. The setting for B1P circuits PVT variation is shown in Fig.6.10.

In addition to the previous setting, there is a "PVT Variation Analysis Options" frame that defines the verification settings. First, the type of PVT variations should be chosen: temperature variation ("T"), or both temperature and voltage supply variations ("T + V"), or even temperature, voltage power supply and process variations ("T + V + P").

✓ *"Vector Scale ratio"*: set the ratio to scale the length of vectors in the two dimensional vector fields.

✓ *"Resolution"*: set the resolution for the two dimensional vector field. To verify the B1P circuits with PVT variation, ITV draws a two dimensional vector field, with the first dimension is the swept voltage source at the break node and the second dimension as the temperature

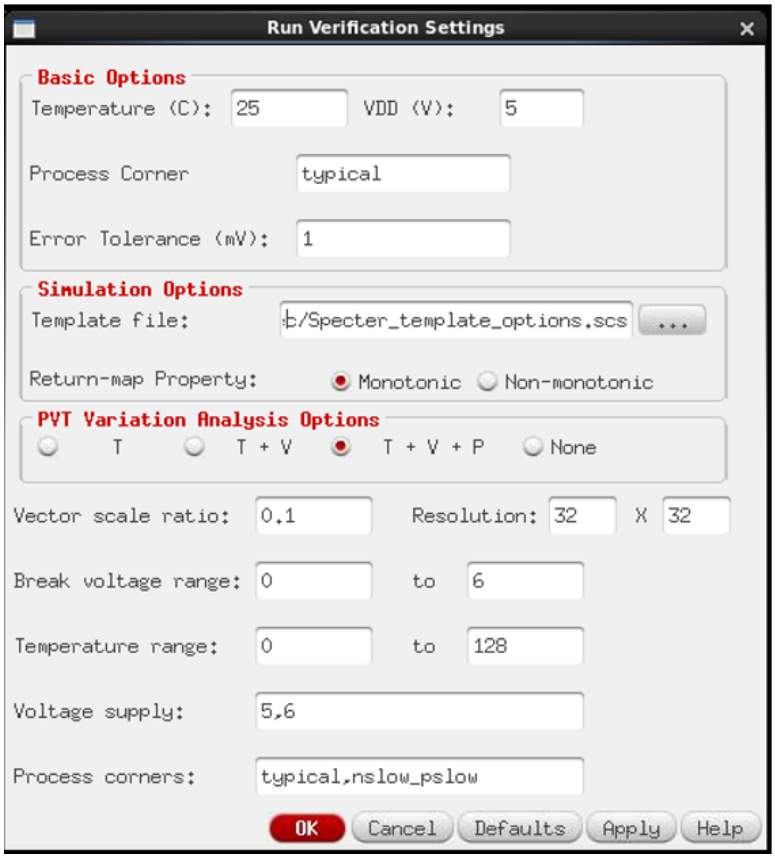


Figure 6.10: ITV "Run Verification Settings" Form for B1P circuits PVT Verification

sweep. Thus, there are two dimensional resolutions: the resolution of "Break voltage range" and the resolution of the "Temperature range." With higher resolution, more accurate verification can be achieved, but at the cost of more simulation time.

✓ *"Break voltage range"*: the sweep voltage range for the inserted voltage source at the selected break point.

✓ *"Temperature range"*: the user can choose the temperature range for verification.

✓ *"Voltage supply"*: this specifies the variation of the voltage supply to be verified. For each voltage supply to be verified, a two dimensional vector field is given to identify the existence of the undesired operating points.

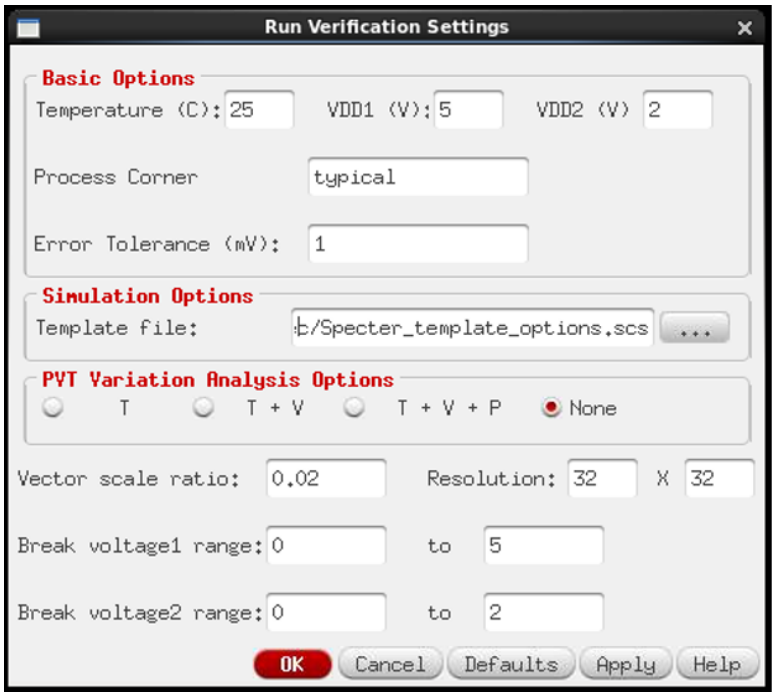


Figure 6.11: ITV "Run Verification Settings" Form for B2P Circuits Stand-alone Verification

✓ *"Process corners"*: this specifies the process corners to be verified. For each corner, ITV plots a two dimensional vector field for each specified voltage supply. For example, if there are m different voltage supplies and n process corners to be verified, ITV draws $m \times n$ two dimensional vector fields.

6.4.2.3 Verification for B2P circuits

For B2P circuits, ITV uses the two dimensional vector field method for both stand-alone and PVT variation verification.

To verify the existence of undesired operating points in B2P circuits, two independent voltage sources need to be swept according to the break-loop homotopy method. Thus, ITV uses the two dimensional vector field method to analyze the existence of undesired operating points in the B2P circuits. Its verification setting is shown in Fig.6.11.

The setting form is similar to that used for B1P circuits but some additional settings should be given in the stand-alone verification for B2P circuits.

✓ *"Vector Scale ratio"*: set the ratio to scale the length of vectors in the two dimensional vector field.

✓ *"Resolution"*: set the resolution for the two dimensional vector field. To verify the B2P circuits, ITV automatically inserts two independent voltage sources at two break points, and their resolutions are chosen here.

✓ *"Break voltage1 range"*: the sweep voltage range for the inserted voltage source at the first break point.

✓ *"Break voltage2 range"*: the sweep voltage range for the inserted voltage source at the second break point.

With these settings, ITV employs the two dimensional vector field to identify the existence of undesired operating points in circuits.

6.4.2.4 PVT Verification for 2D circuits

As with B1P circuits, ITV can verify B2P circuits with PVT variations. The process is implemented by obtaining a series of two dimensional vector fields at different process/voltage supply/temperature settings.

As with PVT variation verification for B1P circuits, the type of PVT variations should be chosen. It can be verification with temperature variation ("T") only, or with both temperature and voltage power supply variations ("T + V"), or even with temperature, voltage power supply and process variations ("T + V + P").

✓ *"Vector Scale ratio"*: set the ratio to scale the length of the vector in the two dimensional vector field.

✓ *"Resolution"*: set the resolution for the two dimensional vector field.

✓ *"Temperature"*: choose the specific temperatures for verification. The difference from the PVT variation for a B1P circuit is that a two dimensional vector field is given for each temperature to be verified.

✓ *"Voltage supply 1"*: this specifies the variation of voltage supply for the first inserted voltage source. For each voltage supply, ITV draws a two dimensional vector field for each specified temperature. For example, if there are m different voltage supplies and n temperatures

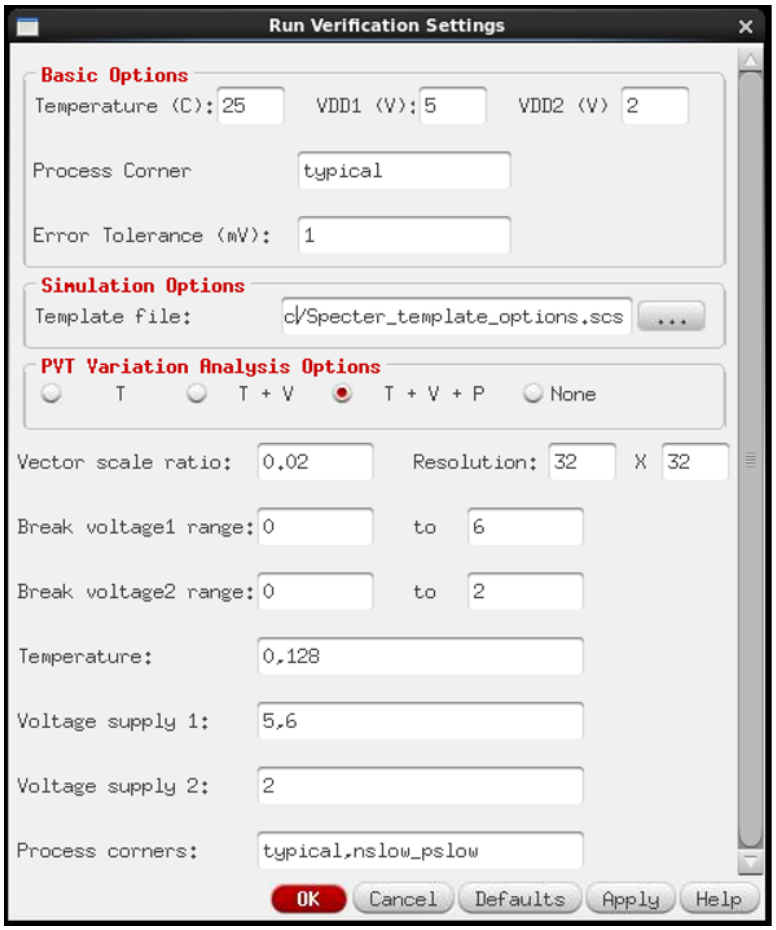


Figure 6.12: ITV "Run Verification Settings" Form for B2P circuits PVT Verification

to be verified, ITV draws $m \times n$ two dimensional vector fields to verify whether undesired operating points exist in the circuit.

✓ *"Voltage supply 2"*: similar to the previous setting, but specify the variation of voltage supplies for the second inserted voltage source.

✓ *"Process corners"*: it specifies the variation of process corners to be verified. For example, if there are m different temperatures, n voltage supply 1, l voltage supply 2 and t process corners to be verified, ITV draws $m \times n \times l \times t$ two dimensional vector fields.

6.5 ITV Installation, Program Files and Demo Circuits

In this section, installation, each file used in the ITV program and benchmark circuits for demo are briefly introduced. All of the files are under the directory:

$\backslash\backslash rs.ece.iastate.edu\backslash Project\backslash Multi\_state\_Verification\backslash ITV\backslash Release\,version\backslash Latest$

This directory is referred to as the "ITV directory" in this chapter.

6.5.1 ITV Installation

The tool installation is very easy and can be finished in 2 steps:

Step1: copy the ITV directory to your virtuoso launch directory.

Step2: Add the following code to the end of your cadence .cdsinit file:

```
ITVMenufile=simplifyFilename("./ITV/customizedVerifyMenuList.il")
procedure( TrUserPostInstallTrigger( args)
;;; creating the first menu item for the pulldown menu
unless( boundp('TrMenuItemTwo)
TrMenuItemTwo = hiCreateMenuItem(
?name 'TrMenuItemTwo
?itemText "&ITV Verification"
?callback "load(ITVMenufile)"
)
);unless
when(
boundp('schEditMenu) && !schEditMenu-¿TrMenuItemTwo
hiAddMenuItem(schCheckMenu TrMenuItemTwo)
)
);procedure
```

After it is installed, in Cadence Virtuoso Schematic Editor, in the check menu, there is an item called "ITV verification, shown in Fig.6.4.

6.5.2 Program Files in ITV

6.5.2.1 Scripts to Generate the Menu in Cadence

The script "customizedVerifyMenuList.il" is a Cadence SKILL script for generating the interface in schematic editor, as shown in Fig. 6.4. It will generate the "SCC & BPSet Identification form shown in Fig. 6.5. It calls other program files to implement the verification tasks.

6.5.2.2 Program Files for Loop Identification and Break

All of these files are included in the "breakLoop" directory.

The top level script is the "createNetlist_hier.il"; it is a SKILL script to generate the interface between ITV program and Cadence Virtuoso environment.

"main.pl" is the top level perl script to convert a circuit netlist to DDG.

"findPath.pl" is the perl script that defines a function to identify all the branch-currents in the netlist and is used by the "main.pl".

"flattenNetlist.pl" is the perl script to flatten the hierarchy netlist , and it is used by the "main.pl".

On the basis of the DDG generated by "main.pl", there are two SKILL scripts to partition the DDG into SCC and identify the BPS in each SCC.

"identifySCC.il" is a SKILL script for finding all the SCCs and highlight them in the Cadence.

"identifyBPS.il" is a SKILL script for finding the BPS for each SCC, and it calls the "searchBreakPoints.pl" perl script to identify the BPS.

With the break points chosen in the previous step, the "autoGenerateNetlist.il" (in the "$breakLoop\autoModifyNetlist$" under the ITV directory) automatically breaks the circuits and adds the voltage source to sweep. It generates a netlist to be verified.

6.5.2.3 Program Files for Run Verification Settings

After the loop is broken, ITV programs uses the files in the "runVerification" directory to run the verification.

For the B1P circuits, it calls the "runVerification1D.il" to implement the verification.

For the B2P circuits, the "runVerification2D.il" would be applied.

In both cases, if it is a two dimensional verification, the two dimensional vector field method is adopted, which is implemented in the "GNUGDCplot.pl"; otherwise, the divide and contraction method is used as defined in the "GNUGDC.pl".

6.5.3 Demo Circuits in ITV

Several benchmark circuits are used as demo. They are in the "*benchmark_circuit\ITV demo*" directory.

"Wilson" is the so-called bootstrapped Vt reference circuit. It can be used as a basic example of identifying undesired operating points.

"WilsonWstartup" is the bootstrapped Vt reference circuit with a startup circuit. It can be used as an example of a circuit with one operating point.

"fixedbias_banba_widlar" is the Van Kessel-Banba circuit, which is the typical 2D circuit.

"Banba_Widlar2SCC" is an example of a circuit with two SCCs.

6.6 Conclusion

In this chapter, an EDA tool called "ITV" is proposed to identify undesired operating points in analog and mixed-signal circuits. ITV automatically converts a circuit netlist to a Directed Dependency Graph (DDG), partitioning the DDG into Strong Connected Components, and finds all the PFLs and NFLs in each SCC. It then automatically identifies the break points to break all the PFLs in each SCC. For a one dimensional verification, it applies the proposed divide and contraction algorithms to determine the presence/absence of undesired operating point. For a two dimensional verification, the two dimensional vector field method is applied

to identify the existence of undesired operating points. Moreover, implementation and usage of ITV have been discussed in detail in this chapter.

CHAPTER 7. CONCLUSION

In this dissertation, two research topics are covered and each is illustrated in detail. First, a low-cost, high-precision DAC structure based on OEM theory is proposed and its design methodology is discussed. It can achieve high matching accuracy by application of the OEM calibration to the resistors in unary weighted segments and calibration of the gain error between different segments by use of calibration DAC (Cal DAC).

As a design example to verify the proposed structure, a high-precision DAC is designed in a 130 nm Global Foundry (GF) CMOS process. The 130 nm GF process features high-density digital circuits but lacks of high-precision resistors or any resistor trimming techniques, so it is generally not suitable for any high-precision DAC design. However, we implemented our design in such process from behavioral model to schematic and layout design. Complicated test scheme and PCB design were also performed. The simulation and measurement results show that the proposed DAC structure can greatly reduce the area requirement and make it possible to implement a 17-bit DAC without using high-precision fabrication process. As a result, it is shown that our proposed DAC structure can significantly lower the costs of high-precision DAC design.

The second topic targeted in this dissertation is identification of the existence of undesired operating points in analog circuits. In this dissertation, a divide and contraction verification method against undesired operating points in analog circuits is proposed. Unlike traditional methods of finding all operating points, this method targets only searches of those voltage intervals containing undesired operating points. To achieve this, a systematic approach for automatically identifying all positive and negative feedback loops in circuits is introduced. A positive feedback loop breaking method and selection of breaking nodes are utilized to determine whether a monotonic return function can be obtained. Depending on the monotonicity of the

return function, two types of divide and contraction algorithms are proposed for use in efficiently searching voltage intervals containing operating points.

In practice, designers also need to verify circuits with transistor sizing and PVT variations or identify the existence of undesired operating point in complicated circuits whose PFLs cannot be broken at only one node (B2P circuits). This type of problem is called the two dimensional analog verification against undesired operating points. For this type of verification, a two dimensional vector field method is proposed that can identify the existence of undesired operating points effectively by visualizing the return functions in the circuits.

On the basis of the proposed verification methods against undesired operating points, an EDA tool called "ITV" was developed for identifying undesired operating points in analog and mixed-signal circuits, on the basis of the break-loop homotopy method. It first converts the circuit into a corresponding graph and locates the break point to break all the positive feedback loops (PFLs). Then, it searches the voltage intervals that contain undesired operating points by divide and contraction algorithms or the two dimensional vector field method. Simulation results show ITV to be effective and efficient in identifying undesired operating points in a class of commonly used benchmark circuits, including bias generators, voltage references, temperature sensors, and op-amp circuits.

BIBLIOGRAPHY

[1] Sebastian Anthony. Soc vs. cpu-the battle for the future of computing, extremetech. https://www.extremetech.com/computing/126235-soc-vs-cpu-the-battle-for-the-future-of-computing. accessed on Jul. 24, 2017. [Online].

[2] Ahmed Eisawy. Improving analog/mixed-signal verification productivity, mentor graphics. https://verificationacademy.com/verification-horizons/october-2012-volume-8-issue-3/improving-analog-mixed-signal-verification-productivity. accessed on Jul. 2, 2017. [Online].

[3] Brian Bailey. Devices threatened by analog content?, semiconductor engineering. https://semiengineering.com/devices-threatened-by-analog-content/. accessed on Jul. 3, 2017. [Online].

[4] A. Ripp, M. Buhler, J. Koehl, J. Bickford, J. Hibbeler, U. Schlichtmann, R. Sommer, and M. Pronath. Date 2006 special session: DFM/DFY design for manufacturability and yield-influence of process variations in digital, analog and mixed-signal circuit design. In *Design, Automation and Test in Europe, 2006. DATE'06. Proceedings*, volume 1, pages 1–6. IEEE, 2006.

[5] Adriana Becker-Gomez, T. Lakshmi Viswanathan, and Thayamkulangara R. Viswanathan. A low-supply-voltage CMOS sub-bandgap reference. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 55(7):609–613, 2008.

[6] Wen Hou et al. Use of a continuation method for analyzing startup circuits. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 1527–1530, 2010.

[7] Chi Yat Leung, Philip K.T. Mok, and Ka Nang Leung. A 1-v integrated current-mode boost converter in standard 3.3/5-v CMOS technologies. *Solid-State Circuits, IEEE Journal of*, 40(11):2265–2274, 2005.

[8] Kenneth S. Kundert and Paul Foreword By-Gray. *The designer's guide to SPICE and Spectre*, pages 42–44. Kluwer Academic Publishers, 1995.

[9] Maurice Egan. The 20-bit DAC is the easiest part of a 1-ppm-accurate precision voltage source. *Analog Dialogue*, 44(04), 2010.

[10] Roddy C. McLachlan, Alan Gillespie, Michael C.W. Coln, Douglas Chisholm, and Denise T. Lee. A 20b Clockless DAC With Sub-ppm INL, 7.5 nV/$\sqrt{Hz}$ Noise and 0.05 ppm/C° Stability. *Solid-State Circuits, IEEE Journal of*, 48(12):3028–3037, 2013.

[11] A.-J. Annema, Bram Nauta, Ronald van Langevelde, and Hans Tuinhout. Analog circuits in ultra-deep-submicron CMOS. *Solid-State Circuits, IEEE Journal of*, 40(1):132–143, 2005.

[12] Peter R. Kinget. Device mismatch and tradeoffs in the design of analog circuits. *Solid-State Circuits, IEEE Journal of*, 40(6):1212–1224, 2005.

[13] Tao Zeng and Degang Chen. An order-statistics based matching strategy for circuit components in data converters. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 60(1):11–24, 2013.

[14] Franco Maloberti, Franco Maloberti, and Franco Maloberti. *Data converters*, volume 1. Springer, 2007.

[15] David Marche, Yvon Savaria, and Yves Gagnon. Laser fine-tuneable deep-submicrometer CMOS 14-bit DAC. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 55(8):2157–2165, 2008.

[16] Alex R. Bugeja and Bang-Sup Song. A self-trimming 14-b 100-MS/s CMOS DAC. *Solid-State Circuits, IEEE Journal of*, 35(12):1841–1852, 2000.

[17] Yonghua Cong and Randall L. Geiger. A 1.5-V 14-bit 100-MS/s self-calibrated DAC. *Solid-State Circuits, IEEE Journal of*, 38(12):2051–2060, 2003.

[18] Douglas A. Mercer. Low-Power Approaches to High-Speed Current-Steering Digital-to-Analog Converters in 0.18-μm CMOS. *IEEE Journal of Solid-State Circuits*, 42(8):1688–1698, 2007.

[19] Gerald A. Miller, Michael C.W. Coln, Lawrence A. Singer, and P.R. Oaklander. A true 16 b self-calibrating BiCMOS DAC. In *Solid-State Circuits Conference, 1993. Digest of Technical Papers. 40th ISSCC., 1993 IEEE International*, pages 58–59. IEEE, 1993.

[20] Roderick McLachlan and Samuel Blackburn. Voltage mode DAC with calibration circuit using current mode DAC and ROM lookup, 2012. US Patent 8,089,380.

[21] L. Richard Carley. A noise-shaping coder topology for 15+ bit converters. *IEEE Journal of Solid-State Circuits*, 24(2):267–273, 1989.

[22] Feng Chen and Bosco H. Leung. A high resolution multibit sigma-delta modulator with individual level averaging. *IEICE Transactions on Electronics*, 78(6):701–708, 1995.

[23] Russ E. Radke, Aria Eshraghi, and Terri S. Fiez. A 14-bit current-mode $\Sigma - \Delta$ DAC based upon rotated data weighted averaging. *Solid-State Circuits, IEEE Journal of*, 35(8):1074–1084, 2000.

[24] Jiang Yu and Franco Maloberti. A low-power multi-bit/spl sigma//spl delta/modulator in 90-nm digital cmos without dem. *IEEE Journal of Solid-State Circuits*, 40(12):2428–2436, 2005.

[25] Tao Zeng and Degang Chen. New calibration technique for current-steering DACs. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 573–576. IEEE, 2010.

[26] Tao Zeng, Kevin Townsend, Jingbo Duan, and Degang Chen. A 15-bit binary-weighted current-steering DAC with ordered element matching. In *Custom Integrated Circuits Conference (CICC), 2013 IEEE*, pages 1–4. IEEE, 2013.

[27] You Li, Tao Zeng, and Degang Chen. A high resolution and high accuracy R-2R DAC based on ordered element matching. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 1974–1977. IEEE, 2013.

[28] You Li and Degang Chen. A novel 20-bit r-2r dac structure based on ordered element matching. In *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, pages 1030–1033. IEEE, 2015.

[29] Hans Van de Vel, Joost Briaire, Corne Bastiaansen, Pieter van Beek, Govert Geelen, Harrie Gunnink, Yongjie Jin, Mustafa Kaba, Kerong Luo, Edward Paulus, et al. 11.7 A 240mW 16b 3.2 GS/s DAC in 65nm CMOS with<-80dBc IM3 up to 600MHz. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, pages 206–207. IEEE, 2014.

[30] Philipp Steinmann, Eric Beach, Wally Meinel, Amitava Chatterjee, Doug Weiser, Roland V Bucksch, and Weidong Tian. Simple analytical model of the thermal resistance of resistors in integrated circuits. *IEEE Transactions on Electron Devices*, 57(5):1029–1036, 2010.

[31] Richard A. Meaney and Raymond J. Speer. Voltage-switching d/a converter using p-and n-channel mosfets, December 24 1991. US Patent 5,075,677.

[32] Chengming He, Xin Dai, Hanqing Xing, and Degang Chen. New layout strategies with improved matching performance. *Analog Integrated Circuits and Signal Processing*, 49(3):281–289, 2006.

[33] Chengming He, Kuangming Yap, Degang Chen, and Randall L Geiger. Nth order circular symmetry pattern and hexagonal tesselation: two new layout techniques cancelling nonlinear gradient. In *ISCAS (1)*, pages 237–240, 2004.

[34] Alan Hastings. *The art of analog layout*. Prentice Hall, 2006.

[35] Xin Dai, Chengming He, Hanqing Xing, Degang Chen, and Randall Geiger. An n/sup th/order central symmetrical layout pattern for nonlinear gradients cancellation. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 4835–4838. IEEE, 2005.

[36] Linear technology. *Precision References*, 9 2014. Rev. E.

[37] Behzad Razavi. *Design of Analog CMOS Integrated Circuits*, pages 380–381. Tata McGraw-Hill Education, 2002.

[38] Ka Nang Leung and Philip K.T. Mok. A CMOS voltage reference based on weighted δv gs for CMOS low-dropout linear regulators. *Solid-State Circuits, IEEE Journal of*, 38(1):146–150, 2003.

[39] Phillip E. Allen and Douglas R. Holberg. *CMOS analog circuit design*, pages 147–150. Oxford Univ. Press, 2002.

[40] Hui Zhang, Hai-gang Yang, Fei Liu, Yuan-feng Wei, and Jia Zhang. Start-up analysis for differential ring oscillator with even number of stages. In *Circuits and Systems (APCCAS), 2010 IEEE Asia Pacific Conference on*, pages 636–639. IEEE, 2010.

[41] Richard O. Nielsen and A.N. Willson Jr. A fundamental result concerning the topology of transistor circuits with multiple equilibria. *Proceedings of the IEEE*, 68(2):196–208, 1980.

[42] Tetsuo Nishi and Leon O. Chua. Topological criteria for nonlinear resistive circuits containing controlled sources to have a unique solution. *Circuits and Systems, IEEE Transactions on*, 31(8):722–741, 1984.

[43] A. Sarmiento-Reyes, L. Hernandez-Martinez, and H. Vazquez-Leal. A topological approach for determining the uniqueness of the dc solutions in mos-transistor circuits. In *Electronics, Circuits and Systems, 2001. ICECS 2001. The 8th IEEE International Conference on*, volume 1, pages 205–208. IEEE, 2001.

[44] Arturo Sarmiento-Reyes, Jan Davidse, E Kleihorst, and A. Van Roermund. A partitioning-based method to determine the uniqueness of the dc operating points of transistor circuits.

In *Circuits and Systems, 1994. ISCAS'94., 1994 IEEE International Symposium on*, volume 6, pages 193–196. IEEE, 1994.

[45] Arturo Sarmiento-Reyes. A novel method to predict both the upper bound on the number and the stability of dc operating points of transistor circuits. In *Circuits and Systems, 1995. ISCAS'95., 1995 IEEE International Symposium on*, volume 1, pages 101–104. IEEE, 1995.

[46] Leon O. Chua and Robin L.P. Ying. Finding all solutions of piecewise-linear circuits. *International Journal of Circuit Theory and Applications*, 10(3):201–229, 1982.

[47] Kiyotaka Yamamura and T. Ohshimar. Finding all solutions of piecewise-linear resistive circuits using linear programming. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 45(4):434–445, 1998.

[48] Stefano Pastore. Fast and efficient search for all dc solutions of pwl circuits by means of oversized polyhedra. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 56(10):2270–2279, 2009.

[49] Stefano Pastore and Amedeo Premoli. A unified set-theoretic approach to the analysis of pwl resistive circuits and composite n-ports. *International Journal of Circuit Theory and Applications*, 39(8):801–822, 2011.

[50] Michał Tadeusiewicz and Stanisław Hałgas. A contraction method for locating all the dc solutions of circuits containing bipolar transistors. *Circuits, Systems, and Signal Processing*, 31(3):1159–1166, 2012.

[51] Michał Tadeusiewicz and Stanisław Hałgas. Some contraction methods for locating and finding all the dc operating points of diode-transistor circuits. *International Journal of Electronics and Telecommunications*, 56(4):331–338, 2010.

[52] L.V. Kolev. An efficient interval method for global analysis of non-linear resistive circuits. *International journal of circuit theory and applications*, 26(1):81–92, 1998.

[53] Kiyotaka Yamamura and Naoya Igarashi. An interval algorithm for finding all solutions of non-linear resistive circuits. *International journal of circuit theory and applications*, 32(1):47–55, 2004.

[54] Leonid B. Goldgeisser and Michael M. Green. A method for automatically finding multiple operating points in nonlinear circuits. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 52(4):776–784, 2005.

[55] Jaijeet Roychowdhury and Robert Melville. Delivering global dc convergence for large mixed-signal circuits via homotopy/continuation methods. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 25(1):66–78, 2006.

[56] Michał Tadeusiewicz and Stanisław Hałgas. A method for finding multiple dc operating points of short channel CMOS circuits. *Circuits, Systems, and Signal Processing*, 32(5):2457–2468, 2013.

[57] Akio Ushida, Yoshihiro Yamagami, Yoshifumi Nishio, Ikkei Kinouchi, and Yasuaki Inoue. An efficient algorithm for finding multiple dc solutions based on the spice-oriented newton homotopy method. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(3):337–348, 2002.

[58] Yen-Ting Wang, Degang Chen, and Randall L. Geiger. Practical methods for verifying removal of trojan stable operating points. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 2658–2661. IEEE, 2013.

[59] Giancarlo Storti Gajani, Angelo Brambilla, and Amedeo Premoli. Numerical determination of possible multiple dc solutions of nonlinear circuits. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 55(4):1074–1083, 2008.

[60] Byeong Gi Lee and Alan N. Willson Jr. All two-transistor circuits possess at most three dc equilibrium points. In *Proc. 26th Midwest Symp. Circuits and Systems, Puebla, Mexico*, pages 504–507, 1983.

[61] Leonid B. Goldgeisser and Michael M. Green. Some two-transistor circuits possess more than three operating points. In *Circuits and Systems, 1999. ISCAS'99. Proceedings of the 1999 IEEE International Symposium on*, volume 5, pages 302–305. IEEE, 1999.

[62] Yusuke Nakaya, Shinichi Oishi, Tetsuo Nishi, and M. Claws. Numerical verification of five solutions in two–transistor circuits. *NOLTA2006*, pages 11–14, 2006.

[63] You Li and Degang Chen. Efficient analog verification against trojan states using divide and contraction method. In *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, pages 281–284. IEEE, 2014.

[64] Chongli Cai and Degang Chen. Performance enhancement induced trojan states in op-amps, their detection and removal. In *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, pages 3020–3023. IEEE, 2015.

[65] Zhiqiang Liu, You Li, Randall L Geiger, and Degang Chen. Auto-identification of positive feedback loops in multi-state vulnerable circuits. In *VLSI Test Symposium (VTS), 2014 IEEE 32nd*, pages 1–5. IEEE, 2014.

[66] K. Ruohonen. Graph theory, graafiteoria lecture notes, tut. http://math.tut.fi/~ruohonen/GT_English.pdf, 2013. accessed on Feb. 2, 2017. [Online].

[67] Yen-Ting Wang, Degang J Chen, and Randall L Geiger. Effectiveness of circuit-level continuation methods for trojan state elimination verification. In *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1043–1046. IEEE, 2013.

[68] Zhiqiang Liu, You Li, Yan Duan, Randall L Geiger, and Degang Chen. Identification and break of positive feedback loops in trojan states vulnerable circuits. In *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, pages 289–292. IEEE, 2014.

[69] Hironori Banba, Hitoshi Shiga, Akira Umezawa, Takeshi Miyaba, Toru Tanzawa, Shigeru Atsumi, and Koji Sakui. A CMOS bandgap reference circuit with sub-1-v operation. *Solid-State Circuits, IEEE Journal of*, 34(5):670–674, 1999.

[70] You Li, Zhiqiang Liu, and Degang Chen. Efficient verification against undesired operating points for mos analog circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2017.

[71] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[72] Jianer Chen, Yang Liu, Songjian Lu, Barry O'sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *Journal of the ACM (JACM)*, 55(5):21, 2008.

[73] T.J. Van Kessel and R.J. Van Der Plaasche. integrated linear basic concepts. *Phillips Tech Review*, 32:1–12, 1971.

[74] John H. Hubbard and Barbara Burke Hubbard. *Vector calculus, linear algebra, and differential forms: a unified approach*. Matrix Editions, 2015.

[75] M. Migliore, V. Martorana, and F. Sciortino. An algorithm to find all paths between two nodes in a graph. *Journal of Computational Physics*, 87(1):231–236, 1990.

[76] Thomas H. Cormen. *Introduction to algorithms*. MIT press, 2009.

[77] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[78] Donald B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84, 1975.