2001

# High-speed high-resolution low-power self-calibrated digital-to-analog converters

Weibiao Zhang
*Iowa State University*

# INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

**The quality of this reproduction is dependent upon the quality of the copy submitted.** Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

# High-speed high-resolution low-power self-calibrated digital-to-analog converters

by

Weibiao Zhang

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Electrical Engineering (Microelectronics)

Major Professors: Marwan M. Hassoun and William B. Black

Iowa State University

Ames, Iowa

2001

Graduate College

Iowa State University

This is to certify that the Doctoral dissertation of

Weibiao Zhang

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Co-major Professor

Signature was redacted for privacy.

Co-major Professor

Signature was redacted for privacy.

For the Major Program

Signature was redacted for privacy.

For the Graduate College

To my family

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENTS

I would like to express my appreciation to those who helped me with various aspects of performing research and the writing of this dissertation. First and foremost, I thank Dr. Marwan Hassoun for his guidance, patience and support throughout the research. His insights and words of encouragement have often inspired me and renewed my hopes for completing my Ph.D education. I would like to thank Dr. William Black for his close guidance during the design and testing process. I like his teaching style and the interesting courses. I would also like to thank my committee members for their useful guidance and suggestions to not only this work but also many other aspects of academic and personal life. I would additionally like to thank Dr. Edward Lee, as my previous co-major professor, who had helped me a lot during the project.

I got a lot of help from Dr. Weber for testing the chip. His profound knowledge and experience on RF and analog circuits are something that I always dreamed of. Dr. Arun Somani and Dr. Yuhong Yang are so kind and generous to offer their help whenever needed. I thank them for reading my dissertation with great patience and helpful suggestions.

Without the help from managers and many engineers from RocketChips, Inc. and Texas Instruments, Inc., the manufacturing and testing of the prototypes are not possible. I would like to thank specially Dr. Scott Irwin, Mr. Matthew Shafer, Dr. Yvette Lee, Mr. Justin Gaither and Mr. Khaldoun Bataineh from RocketChips for their great help during the tapeout process and testing. I would like to thank Dr. Richard Hester and Dr. Glenn Westphal from TI for their help on testing of the prototypes and valu-

able discussions of the dissertation work. I thank Dr. Jim Hellums for showing me some useful Latex tricks and formating of the dissertation.

I would like to thank many classmates and friends at Iowa State University, they always offer generous help whenenve needed. We shared a great memory of that beautiful little campus town, Ames, Iowa. I thank Huimin Xia, Yonghui Tang, Huanzhang Huang, Huawen Jin, Lin Wu, David Han, Jiandong Jiang, Irene Deng, Sherli Chew, Jin Ye, Xiaoyu Xi, Baiying Yu, and Mathew Bibee for their help on research projects and courses.

I would also like to take this chance to thank my wife, Jun Li, who is always with me, sharing my frustration, pain, and success during these years. I greatly appreciate my parents and parents-in-law for their love and support.

# ABSTRACT

High-speed and high-resolution low-power digital-to-analog converters (DACs) are basic design blocks in many applications. Several obvious conflicting requirements such as high-speed, high-resolution, low-power, and small-area have to be satisfied. In this dissertation, a modular architecture for continuous self-calibrating DACs is proposed to satisfy the above requirements. This includes a redundant-cell-relay continuous self-calibration scheme. Several prototype DACs were implemented with self-calibration schemes. Also a DAC synthesis algorithm using a direct-mapping method and the modular structure was developed and implemented in the Cadence SKILL programming language.

One of the prototypes is a 250MS/s 8-bit continuous self-calibrated DAC that has been implemented in TSMC's 0.25$\mu$ single poly five metal logic CMOS process. The structure of the self-calibrated current cell has high impedance and low sensitivity to output node voltage fluctuations. The chip has achieved +0.15/-0.1 LSB DNL, -0.6/+0.4 LSB INL, and 55dB SFDR with a lower input frequency at a conversion rate of 250MS/s. It consumes 8 mW of power in a 0.13 mm$^2$ die area.

Glitches caused by switching of the calibration clock degrade the SFDR especially in high-speed applications. A new redundant-cell-relay continuous self-calibration scheme was proposed to reduce the glitches. Simulation results showed that the glitch energy is reduced 10 fold over existing schemes. A 10-bit DAC was implemented in the 0.25$\mu$ CMOS process mentioned above. +/-0.5 LSB INL and -0.45/+0.2 LSB DNL were measured and 70dB SFDR was achieved with a lower input frequency at a 250MS/s

conversion rate. Up to the Nyquist rate, the SFDR is above 53dB at a conversion rate of 200MS/s. The DAC dissipates 8mW in a 0.3mm$^2$ die area. The testing results verified the redundant-cell-relay continuous self-calibration for high-speed high-resolution low-power and low-cost DACs.

Additionally, a DAC synthesis algorithm was developed based on a direct mapping method. Given the specifications such as the DAC's resolution, full range scale and technology, the synthesizer will map them directly into pre-existing functional blocks implemented in the DAC synthesis libraries. The program will then synthesize the schematic and layout that closely meet the given specifications.

# CHAPTER 1. INTRODUCTION

## 1.1 Background

With the fast development of communication and networking technology, digital signal processing and digital computers, the interface speed between the analog domain and digital domain is becoming a bottleneck. High-speed and high-resolution analog-to-digital or digital-to-analog converters are in great demand in high-speed communication applications. Recent interests in using digital signal processing in wireless personal communication and video signal processing have created a demand for fast digital-to-analog converters (D/A converters or DACs) with accuracy of 10-14b [1]. Gigabit LAN applications push the speed of DACs to be equal or above some hundreds of MHz.

On the other hand, digital signal processing drives the process technology toward deep sub-micron dimensions and high-speed performance. Many companies use $0.25\mu$ and $0.18\mu$ CMOS processes for their very fast digital circuits. Figure 1.1 shows the CMOS technology trend in recent years [2]. Although the power supply is scaled down, the threshold voltage is not scaled down by the same factor, see Figure 1.2 [2]. Matching accuracy is also not improved with the scaling down of feature sizes. The lower power supply, high threshold voltages and bad matching (for small feature sizes, the matching can be even worse since matching accuracy is proportional to the area) generated a big challenge for analog or mixed-signal design of high accuracy systems.

In DAC design, although many implementations still use voltage mode, current mode structures are becoming more important especially for high-speed applications. Current-

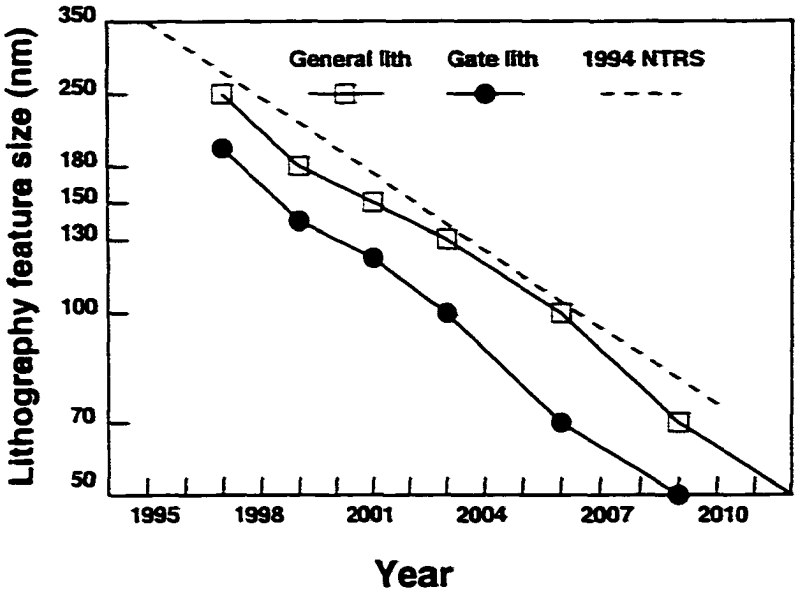


Figure 1.1 CMOS technology trend

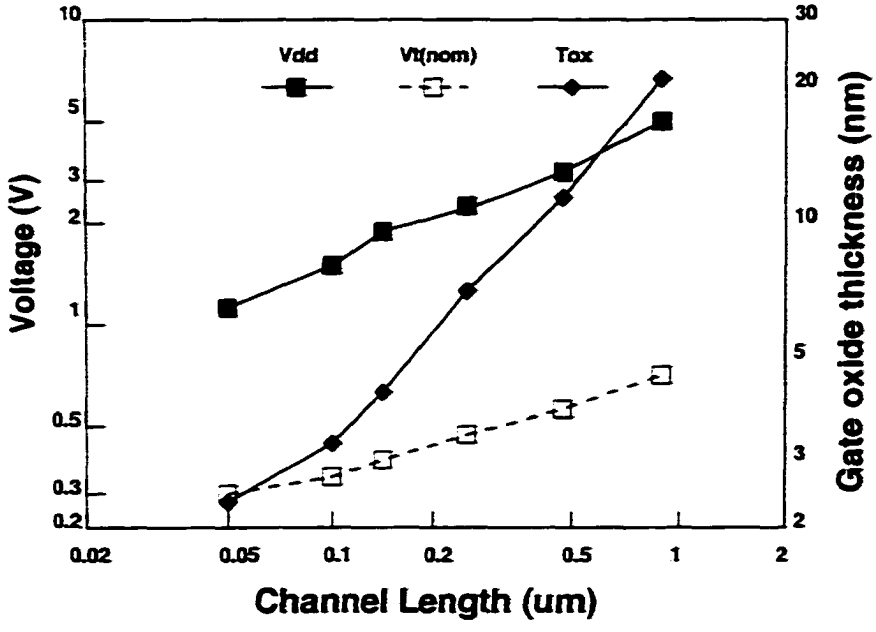


Figure 1.2 Typical CMOS design parameters.

steering DACs are quite popular since they offer faster conversion speed. In current steering DACs, accurate current cells are very critical. Matching is a limiting factor to accuracy. Normally matching accuracy is about 10-bits [3], with some analog processes, approximately 11 bits may be achieved. In order to achieve high accuracy, various calibration techniques have to be applied. Some calibration techniques like laser trimming and external adjustment [4] cost time and money on testing facilities and have aging and temperature sensitivity problems. Some self-calibration techniques were developed to overcome the disadvantages of trimming. These techniques need a special calibration period [5] during which no data conversions are performed. Some applications do not allow such conversion stoppages. A continuous self-calibrating technique [6] is very attractive for achieving high accuracy without the above disadvantages. It adopts a new current calibrating technique that does not need very accurate matching. The continuous self-calibration current-steering technique has been implemented in audio applications where the sampling speed is 44.1KS/s.

## 1.2 Motivation

In this dissertation, we will try to realize high-speed and high-resolution low-power small-area DACs. In high-speed situations, the accuracy is degraded since the time for settling is much less than in audio applications. We will employ the speed advantage of deep-submicron processes and overcome matching limitations by self-calibration. Some techniques are also needed for improving the accuracy of the current cells.

Synthesis of analog and mixed-signal circuits is a dream of many people, due to various advantages such as low cost, fast time-to-market, efficiency, etc. However, analog and mixed-signal synthesis lags behind its digital counterpart. This is partly due to the fact that analog circuits operation is continuous and specific knowledge is needed for a specific circuit. Digital synthesis has been very successful, many commercial tools are

available. Digital circuits have good portability. For analog and mixed-signal designs, modularity is also very important for synthesis. In this work, we propose a modular structure for high-speed high-resolution DACs. Based on that, a synthesis algorithm can be developed to generate a DAC given its specifications and modular selection.

## 1.3 Summary of Contributions

The contributions of this dissertation can be summarized in five parts. First, the dissertation has an extensive overview of DAC architectures and synthesis tools. This review can serve as a reference for DAC design. The review laid a basis for deriving the theoretical equations and choosing modular architectures for the design and synthesis of high-speed high-resolution low-power self-calibrated DACs. Secondly, the dissertation extends the concept of continuous self-calibration from the audio frequency range to conversion speeds of hundreds of MHz. This is verified with a working 8-bit 250MS/s prototype. High accuracy self-calibrated current cells that include charge injection compensation and clock feedthrough reduction techniques were proposed and implemented. Fast settling techniques were also included. The 8-bit prototype also clearly depicted the glitches that happen during the transition from the calibration mode to the supplying mode. The glitches clearly degrade the DAC accuracy as well as its SFDR. The third contribution is the invention of the redundant-cell-relay continuous self-calibration method for current-steering DACs. The method greatly reduces the glitch energy inherit in the previous self-calibration method. A patent application has been submitted based on the method [7]. The fourth contribution is the proof of the redundant-cell-relay method in silicon. A 200Ms/s 10-bit DAC prototype was fabricated and tested. Its combination of high-speed, high-resolution, low-power and small area has great potential to many embedded applications. A paper [8] based on the prototype has been accepted to the 27th European Solid-State Circuits Conference to be held in September,

2001. The fifth contribution is the development of a DAC synthesis tool based on the above prototypes' modules and the direct mapping method published by the author of the dissertation in 1998 [9]. The synthesis tool is good for users with little knowledge of details of DAC design. This is useful for achieving fast time-to-market turnaround of critical DAC designs. In summary, this dissertation provides a package of methodology and tools for the design, implementation and synthesis of high-speed high-resolution low-power small-area DACs.

## 1.4 Dissertation Organization

Starting from a literature review, we will investigate design techniques for DACs and propose a modular architecture for high-speed high-resolution DACs. The architecture includes the current-steering structure and continuous self-calibrating techniques. Then we will establish some prototype modules and chips. A simple synthesis algorithm of modular DACs will be designed based on the prototypes. The dissertation is organized as follows. In Chapter 2 literature on the DAC design techniques and synthesis methodology are reviewed. In Chapter 3 modular DAC architectures and some considerations about the realization and calibration methods to achieve high accuracy are described. In Chapter 4 the circuit design, implementation and testing results of a prototype 8-bit 250MS/s DAC in TSMC's 0.25$\mu$ single poly five metal logic CMOS process with continuous self-calibration are illustrated. Some improvement considerations for the future are proposed. Based on the analysis in Chapter 5, a new continuous self-calibration method called redundant-cell-relay is proposed. In Chapter 6, some other prototypes are designed and implemented in TSMC's 0.25$\mu$ single poly five mental logic CMOS process. The redundant-cell-relay method is implemented in 10-bit and 12-bit prototypes. Chapter 7 discusses the synthesis algorithm and a DAC synthesizer based on the modular architecture and a direct mapping method is implemented in SKILL language.

Chapter 8 concludes the dissertation.

# CHAPTER 2. DIGITAL-TO-ANALOG CONVERTER DESIGN AND SYNTHESIS

In this chapter, various architectures for digital-to-analog converters and the synthesis of digital and mixed-signal circuits are to be reviewed. A digital-to-analog converter (DAC) is a device which receives digital information in a form of n-bit word and transforms it into an analog signal [10]-[13]. DACs appear in many application fields, such as instrumentation and automatic testing, arithmetical and trigonometrical operations, communications and signal analysis, visual displays, and industrial automation. The list is definitely incomplete, anyone from a special field would think of several applications where DACs are needed. In addition, reviewing all the DACs is beyond the scope of the thesis. In this chapter, some commonly used high-speed and/or high-resolution DAC architectures and techniques are covered.

Automatic synthesis is the translation of a higher level description of a system into a lower level description in a design hierarchy. According to the description level, synthesis can be layout synthesis, structural synthesis, high level (behavioral) synthesis or system level synthesis. In this chapter, various synthesis tools and algorithms for digital and mixed-signal circuits are investigated.

## 2.1 Overview of DACs

Figure 2.1 shows a block diagram of a DAC. Digital signals are applied to the converter, they are converted into an analog output signal, which is either a voltage or

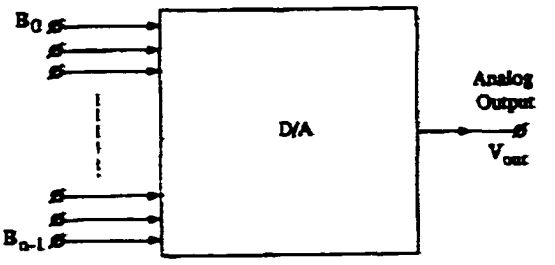


Figure 2.1 DAC diagram

current signal. The transfer function can be represented by equation 2.1:

$$\sum_{k=0}^{n-1} B_k 2^k R_{ref} = V_{out} \tag{2.1}$$

Here n is the width of the digital input word and the resolution of the DAC. $B_k$ is a digital bit for the word $B_{n-1}B_{n-2}...B_1B_0$, and $R_{ref}$ is a unit reference corresponding to a Least Significant Bit(LSB). It can be a voltage, a current or a charge. A reference voltage or current is normally used in practice. An example of a 3-bit DAC transfer function can be seen from Figure 2.2. There are $2^3$ staircases in the transfer curve.

It is not realistic to review all the DAC structures in one chapter. Due to the importance of DACs much research effort has been spent and many kinds of DAC structures exist. A simple classification of DACs can be seen from Figure 2.3. Generally, we can classify DACs into two big classes: Conventional DAC and oversampling DAC. Oversampling DAC adopts the noise-shaping techniques [14], [15]. Normally they are used in audio or voice-band applications. We are more interested in high-speed converters, so conventional converters are the major focus. Conventional DACs can be classified into indirect and direct types. Indirect DACs convert the digital input into some intermediate parameter, then using some other analog parts such as filter to convert the intermediate parameter to the analog output. Pulse Width Modulation (PWM) DACs [16], Stochastic DACs [17] and Integrating DACs [18] belong to this type. The accuracy of this kind

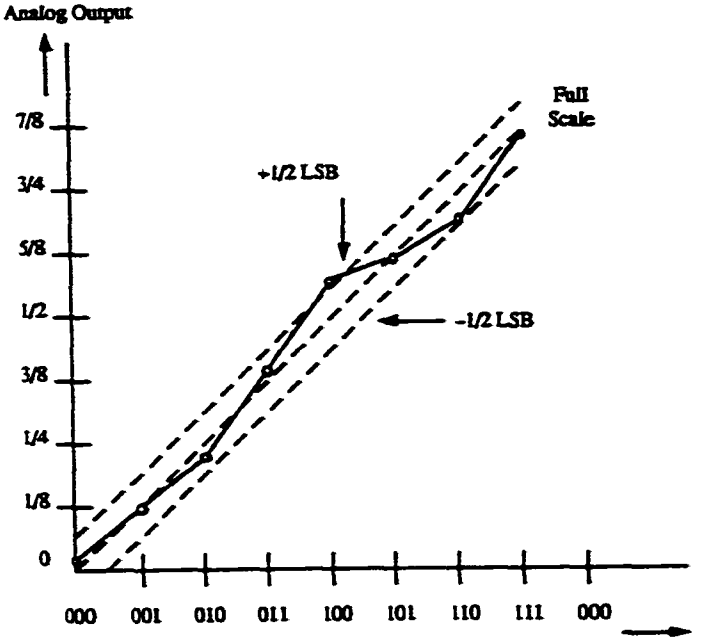


Figure 2.2 Transfer function of a DAC

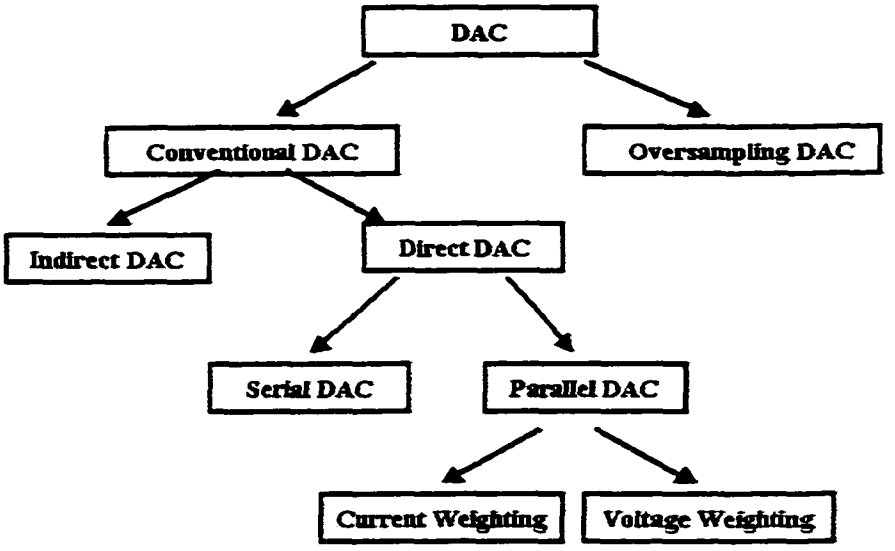


Figure 2.3 Classification of DACs

of converter can be very high, but the speed is normally low, so it is more widely used in audio applications and instrumentation than in high speed applications. In the direct converter family, serial direct converters such as the serial hold converter, need multiple clock conversion cycles for one binary word, so they tend to be slow. On the other hand, parallel direct DACs tend to be faster in nature, so we are focused on parallel direct DACs.

Parallel DACs may be deduced directly from transfer function 2.1. They have the following elements: a reference quantity; a weighting system; a digital control; and summation of these various signals [19]. Various parallel DACs differ in the way of realizing these elements. Generally there are two types of weighting system: current weighting and voltage weighting (division).

### 2.1.1 Current Weighting DAC

The Current weighting method generates weighted currents and sums them together. The summation is controlled by the digital words. Due to the easy summation of currents, this method is very popular. There are lots of variations in this type. Many of them use resistor network, such as R-2R network, resistor string, etc.. Some of the current weighting use device scaling; either bipolar or MOS devices are used to generate weighted currents, while others just use a number of unit current cells and add them according to the digital words.

#### 2.1.1.1 R-2R ladder DAC

Figure 2.4 shows a R-2R ladder DAC [20]. The R-2R resistor network provides binary weighted currents, the ON resistance of the MOS bit switches is scaled to obtain a good accuracy of the converter. The output currents of the ladder network are fed to the I-V conversion amplifier. The ladder system is simple to implement, but resistor matching is critical. In practical IC design, thin film resistors are best but need additional process

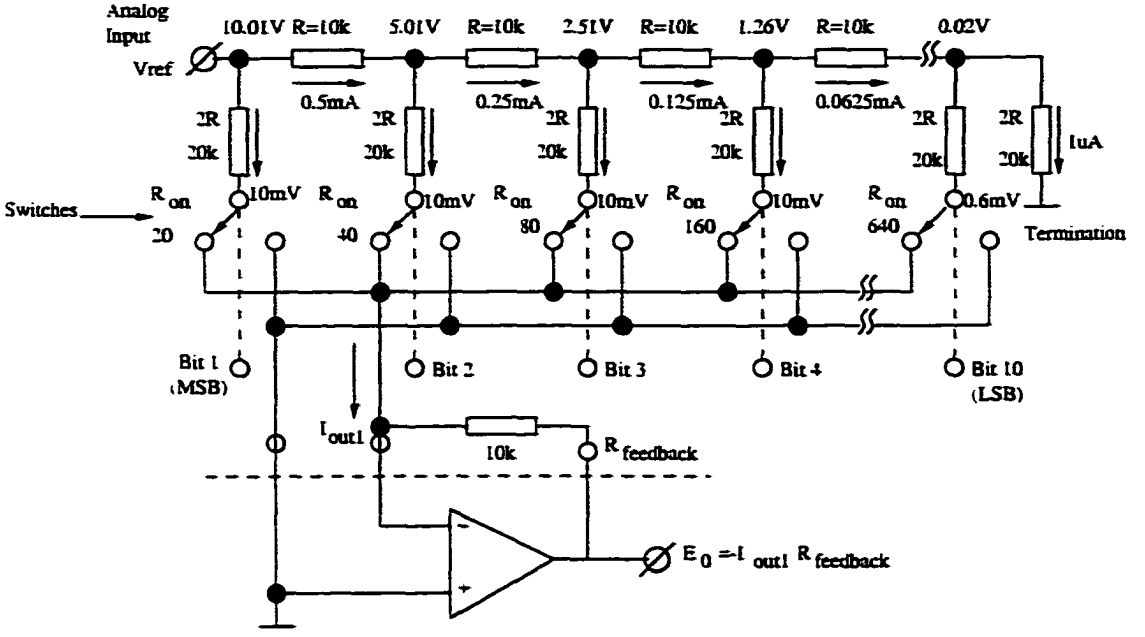


Figure 2.4 A R-2R ladder DAC in CMOS technology

steps to incorporate these elements on the same die. Laser trimming can be used to get accurate matched resistors and 12 bit accuracy can be achieved. But trimming is expensive, and the system becomes sensitive to material stresses. The speed of this kind of amplifier is also limited by the RC constant at the output node and the operational amplifier's settling speed.

### 2.1.1.2 Binary Resistor Weighting Current

Figure 2.5 is a DAC using a binary weighted resister Network. It maintains equal potential on the emitter nodes of the transistors. The resistance of the lower bits increase exponentially, this makes the die area unpractical for high resolution. Because of the large total resistor needed this architecture it is not suitable for IC circuits.

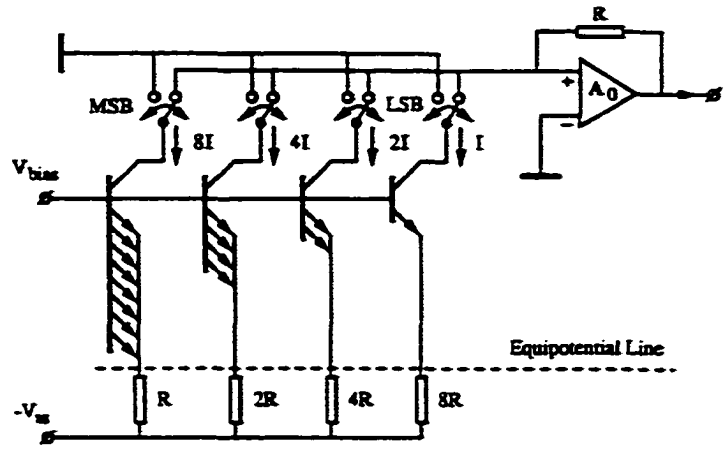


Figure 2.5 A binary weighted resister DAC

#### 2.1.1.3 Equal Currents Binary Weighted Resistor Network

Figure 2.6 shows an equal current binary weighted resistor network [20] . The reference current I is formed by a feedback loop that controls the base voltages of the transistors such that $I = \frac{V_{ref}}{R}$. The R-2R resistor network on the top gives the binary weights to the currents through the bottom resistor array. The speed of this structure is high since the output node has only one transistor capacitance load and the equivalent resistance at the node is normally designed for 75 Ohm or 50 Ohm. The disadvantage is the number of accurately matched elements is increased. This has an influence on the yield, especially when the resolution is increased to 10 bit or above.

#### 2.1.1.4 Device Scaling

Figure 2.7 depicts a current weighting scheme using device scaling [3]. The weights of the currents are determined by the ratio of the number of unit sized transistors. Large numbers of unit sized transistors are used to generate a binary-weighted current division. Analysis and measurement results show that a 10-bit binary-weighted MOS network can be designed having $\frac{1}{2}$ LSB INL over a tail from 1uA to 20mA. The MSB

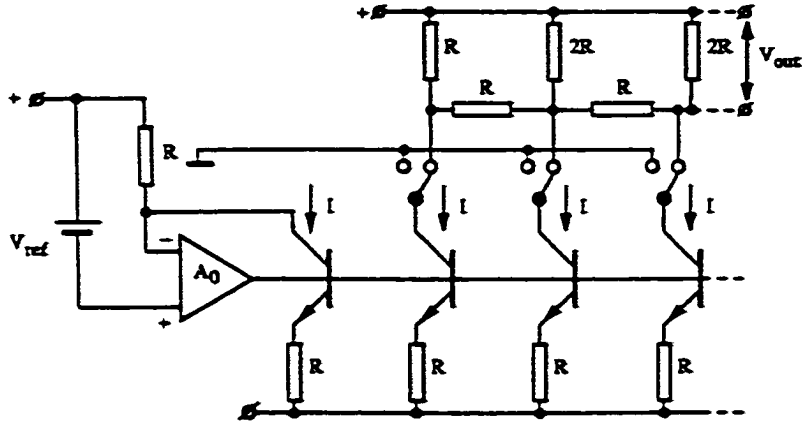


Figure 2.6 Equal currents resistor array DAC

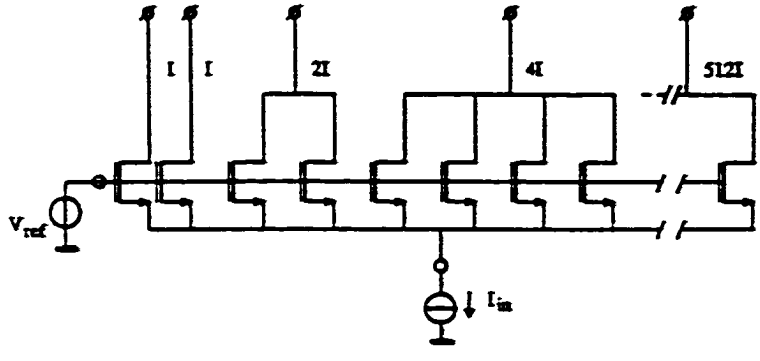


Figure 2.7 Current weighting by MOS device scaling

current value is made by adding randomly distributed unit currents, this gives a better matching result [3]. Channel length modulation effects can impact the ratio. When the Vds voltages of these transistors are not all equal, the currents are not scaled perfectly. Mismatches due to the unequal Vds's and device sizes impact the linearity.

#### 2.1.1.5 Two-step current division

Two step current division is based on device scaling [20]. Figure 2.8 shows one example. The reference current I is formed by a feedback loop. An R-2R current network and emitter scaling are used to form the most significant bit current. The current value $\frac{I}{8}$ is repeated and then scaled further through emitter scaling. Although the figure shows only bipolar transistors, it is also applicable to CMOS. Using the two-step structure, we can use a lower number of components that require high accuracy and relax the requirements on the remaining ones. Hence, the die size and amount of elements can be optimized.

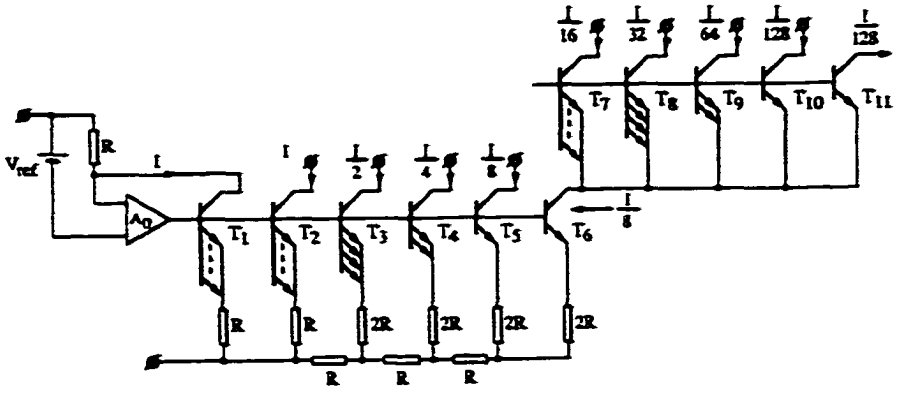


Figure 2.8 A two step DAC

#### 2.1.1.6 Current Steering

In recent years, current steering DACs have become quite popular [21]-[33]. They are used in video and broadband applications due to their high-speed merits. They are based on an array of matched current cells organized in unary or binary weighted elements. Figure 2.9 shows a typical current steering DAC architecture. The current cell array is the central part of the DAC. Row and column decoders are used to select the current cells. Latches are used to switch input data synchronously to control the current cell contribution to the output. This synchronization reduced the harmonic distortion due to the difference between the time delay of different input data bus to current cells. Matching errors are a major concern in current steering DACs. Calibration and compensation are needed for high accuracy design. The speed limitation factors are from switches, output load RC constants and digital decoding logic delays. In a deep sub-micron CMOS process, digital logic can go easily to 1GHz, while the RC load is normally determined by the application. In embedded applications, on-chip RC load can be so small that the high speed is easily obtained.

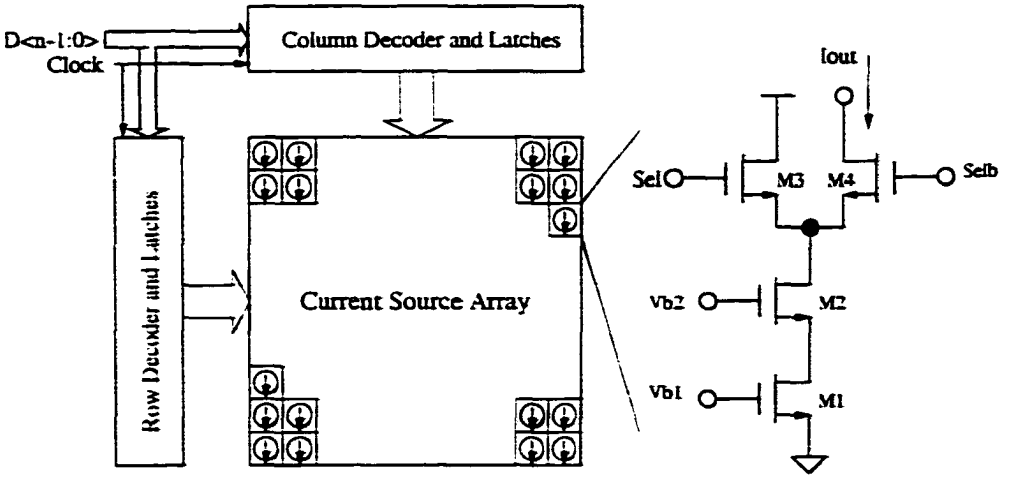


Figure 2.9 A current steering DAC

### 2.1.2 Voltage Weighting DAC

Voltage weighting can be done through weighted capacitors and the charge redistribution principle. It can also be realized by voltage division through a resistor string or several strings.

#### 2.1.2.1 Weighted Capacitor Array DAC

Figure 2.10 shows an example of weighted capacitor arracy DAC using binary capacitor array [34]. Clock signals are used to control the switches. When the clock is low, all the terminals of capacitors are shorted to the ground, when the clock is high, the voltage reference is applied to the capacitors depend on the digital inputs. The transfer function of the example DAC is written as: $V_{out} = \frac{Din}{16} V_{ref}$. These techniques can be applied to 10 to 12 bit DACs in a CMOS technology. The speed of the architecture is limited by the operational amplifier's settling performance and the parasitic capacitances.

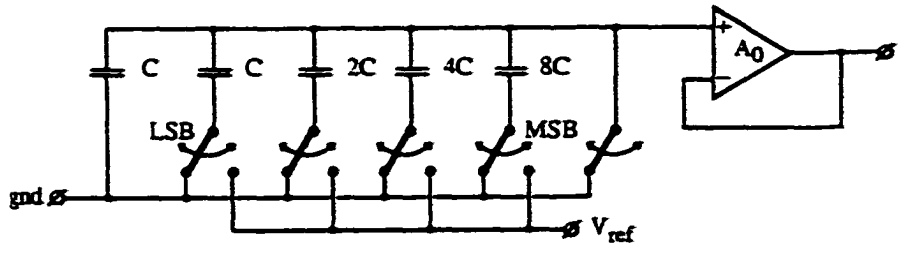


Figure 2.10 Binary weighted capacitor array DAC

#### 2.1.2.2 Voltage Division

A 16-bit DAC [35] using voltage division is shown in Figure 2.11. R1 to R255 construct a coarse divider. R256 to R511 construct the fine divider part. Two sets of switches in the MSB segmentation are driven by the MSB segment decoder and select a coarse

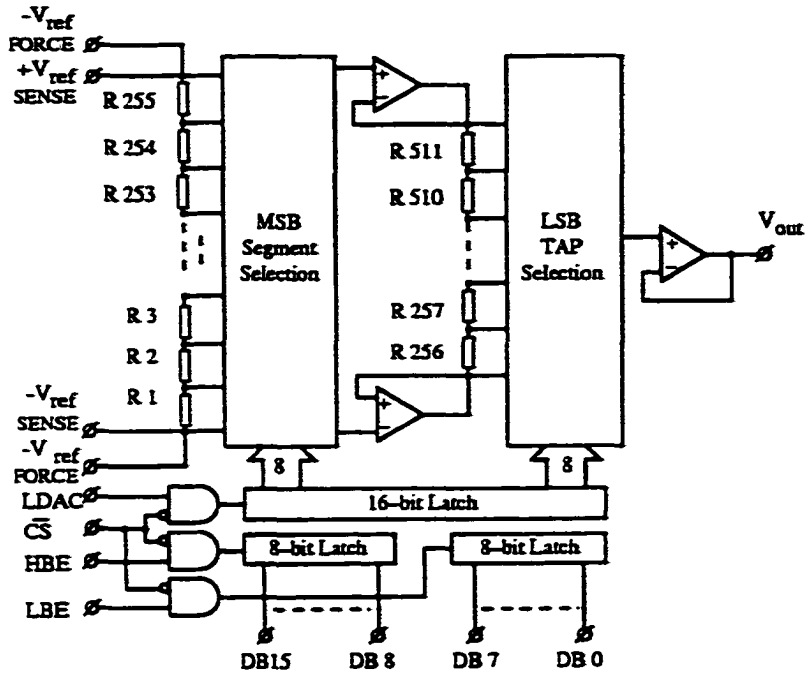


Figure 2.11 Voltage division DAC

voltage for the two source follower amplifiers, then the set of switches in the LSB part select the fine tap according to the LSB decoder control. Although two strings are used here, there are also one-string applications. The architecture also guarantees monotonic conversion. Buffers can create a nonmonoton output unless decoded to prevent it. The speed of the architecture is limited by the parasitic capacitance and the resister values, speeds less than 100MS/s have been reported [36]. If we reduce the resistor values, power consumption would be increased, so a trade-off has to be made between speed and power.

### 2.1.3 Calibration Methods

The earliest calibration method was laser trimming. In a thin film resistor process, the laser trimming option is available. Trimming can get accurate resistor values and accurate matching, but there are several drawbacks associated with trimming. Trimming

is sensitive to mechanical stress, temperature, and there are aging effects. In addition, trimming is expensive. So some other calibration alternatives are much more attractive.

#### 2.1.3.1 Self-Calibration

If a DAC is not continuously in use, a self-calibration procedure can be used to correct the error caused by inaccurate elements. Figure 2.12 [37] shows an example DAC with self-calibration. During the calibration cycle, an accurate ramp is generated and the main DAC output is compared with the ramp signal. The output of the comparator triggers the error correction logic to generate a digital correction signal that is stored in the RAM. The RAM data controls the sub-DAC to make up for the difference between the main DAC and the ideal output. During normal conversion, when a code is applied to the DAC, the correction value in the RAM is called and applied to the sub DAC. Note that during calibration, the converter cannot convert . In some applications, this is not allowed. Another drawback is that the accurate ramp function is difficult to generate.

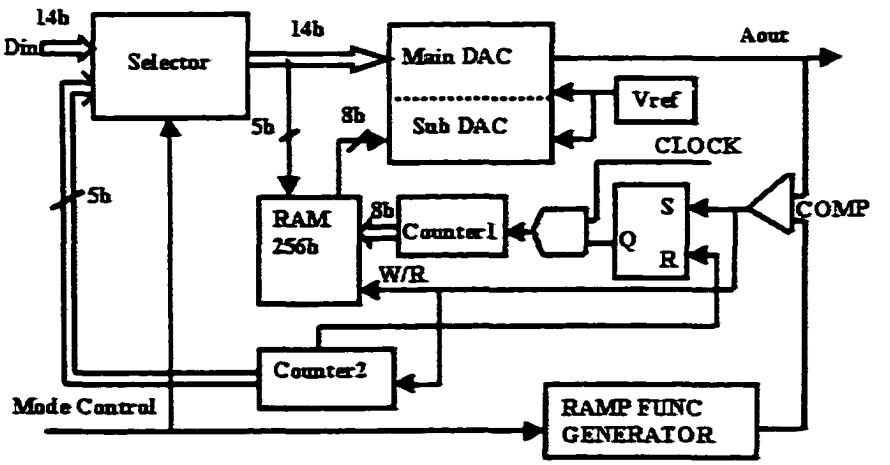


Figure 2.12 Self-compensation DAC

### 2.1.3.2 Continuous Self-Calibration

Figure 2.13 [6] shows a continuous self-calibration scheme for current steering DACs. In the continuous self-calibration scheme, a current source can be in one of the two modes: calibration mode and supplying mode. When a current source or cell is being calibrated to a reference current, it is in the calibration mode. When the current source or cell supplys current to the output load, it is in the supplying mode. The system consists of N+1 shift registers and N+1 current sources where N sources are needed for conversion and one source is a "spare" or "redundant" source. The shift register determines which current source is calibrated. When a cell enters its calibration mode, the "spare" source will take over and supply current for it. Current cells are calibrated to the reference current successively. The switching network performs the necessary switching between the calibrating and the supplying mode. By using N+1 current sources, the system can convert and calibrate at the same time. The shift register is a looped register so that continuous calibration is enabled. A small latency exists for the first calibration of all the N+1 cells.

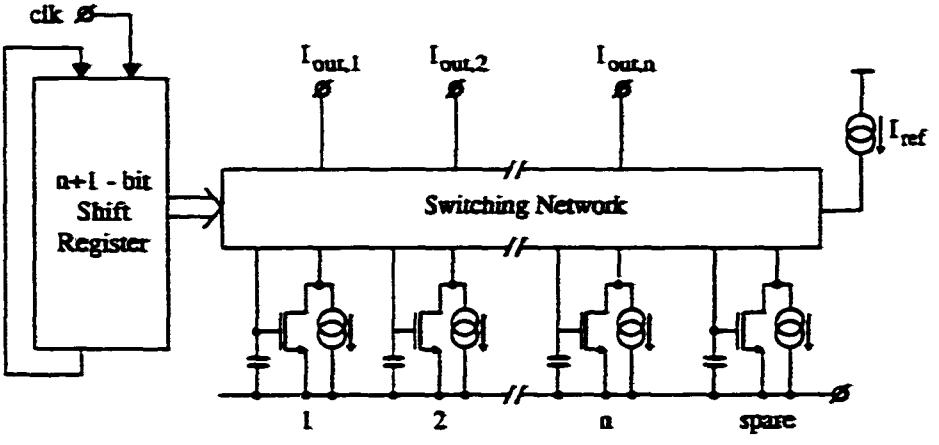


Figure 2.13 Continuous self-calibrating DAC

### 2.1.4 Other Error Compensation Techniques

In addition to calibration methods, some other techniques are also useful for achieving high accuracy. A layout technique using random distribution of current cells can be used to reduce the error effect. Threshold compensation technique may also be used to compensate the spatial gradients of threshold across the wafer.

#### 2.1.4.1 Matching Layout Techniques

In [22], a 12-bit high yield DAC is implemented by using centroid layout techniques. The binary weighted current sources are built through centroid distributed unit current sources over the array area. Figure 2.14 shows the distribution for the four most significant bits. A "new" unit current source with a value of 16 "old" unit current sources is the unit shown in Figure 2.14. The 16*16 matrix of "new" unit current sources is defined for the 8 most significant bits. The four least significant bits are constructed by connecting in series the necessary number of "new" unit current sources. Bit11 is constructed by connecting all the unit current sources and marked as "*". The unit current sources for Bit10, Bit9, and Bit8 are marked as "o", "÷", and "x" respectively.

In [29], a 14-b 150MS/s update rate DAC is obtained by using a switching scheme called "$Q^2$ Random Walk". The scheme used layout techniques to control the switching sequence that allows compensation of spatial errors. Figure 2.15 shows the switching sequence. Every unary current source is split into 16 units, which are spread across the current source array: four (quad) units in every quadrant compose one current source, hence the name Quad Quadrant ($Q^2$). The linear systematic and graded errors are compensated. Both of the techniques need careful layout skills, and the routing is complicated.

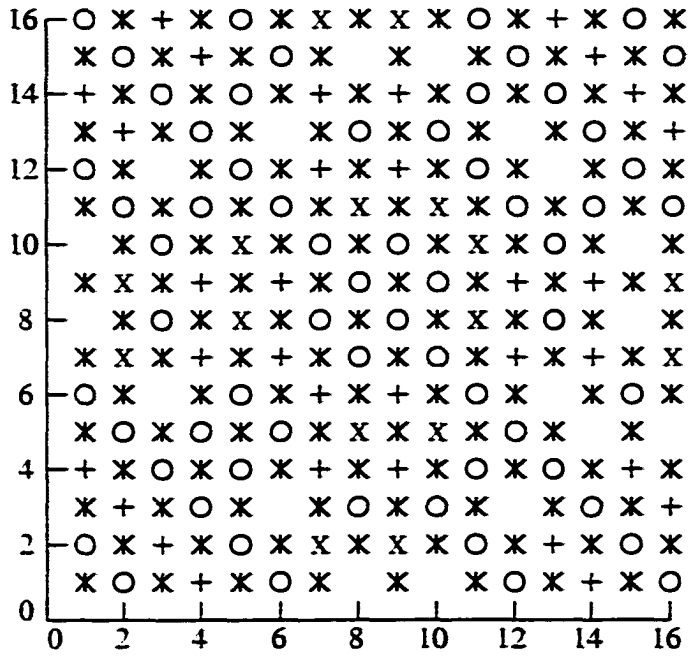


Figure 2.14 Binary weighted current source distribution

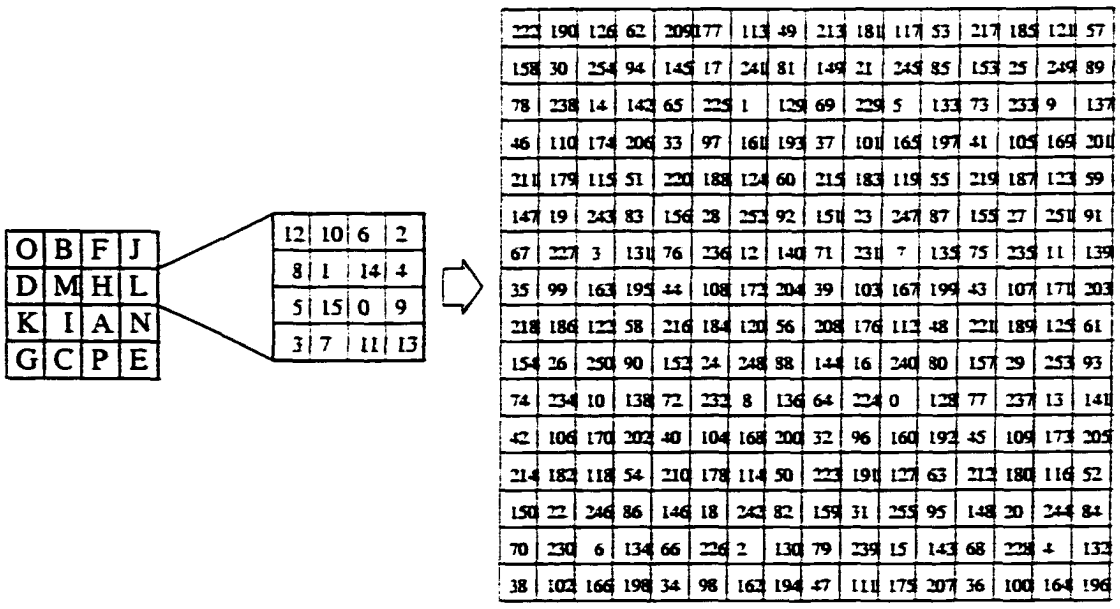


Figure 2.15 $Q^2$ random walk switching sequence

### 2.1.4.2 Threshold Compensation

Although current-steering DACs allow fast and accurate settling, parameter gradients over a wafer and mutual mismatches of current sources result in large variations of performance of the fabricated DACs [21]. Special layout arrangement and switching sequences can be used to improve the linearity of the current matrix, but the interconnection is complex and the silicon area is increased significantly for a high resolution DAC, and large area makes the mismatching and threshold variations worse. So reducing the number of current sources and layout area is necessary. A threshold-voltage compensated current source is shown in Figure 2.16.

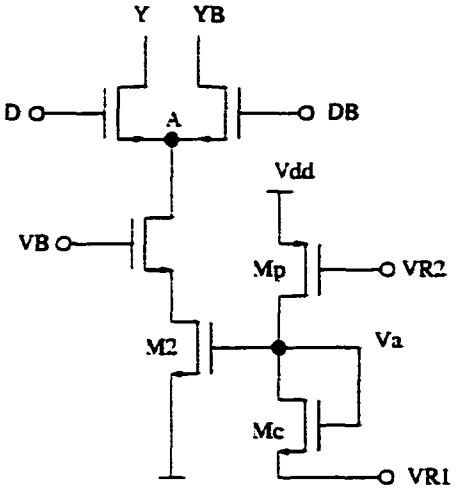


Figure 2.16 Switched current source with threshold-voltage compensation

Neglecting the early effect, the current through M2 is represented in equation 2.2.

$$I_2 = K\frac{W_2}{L_2}(V_a - V_{t2})^2 = K\frac{W_2}{L_2}(VR_1 + \delta V + V_{tc} - V_{t2})^2 \qquad (2.2)$$

where $\delta V = V_a - VR_1 - V_{tc}$. As long as the transistors M2 and Mc are locally matched, $V_{tc} - V_{t2}$ is very small. Even if the threshold-voltage mismatch exists, the

variation of $\delta V$ in 2.2 is dependent upon $I_c$. By choosing small W/L ratio of $M_p$ and high $VR_2$, we get $I_c << I_2$, the variation of $I_c$ to $I_2$ can be very small. Simulation shows 20 times variation reduction can be achieved.

The above overview shows that designing a very high-speed high-resolution DAC ( Hundreds of MS/s) is not a trivial task. Current-steering DAC with continuous calibration is a promising architecture for this design task. Other techniques such as layout matching, threshold compensation, segmentation or two-step and current-scaling are also beneficial if used properly. Detailed techniques and modifications will be considered in Chapter 4.

## 2.2 Overview of Synthesis

Automatic synthesis is the translation of a higher level description of a system into a lower level description in a design hierarchy. According to the description level, synthesis can be layout synthesis, structural synthesis, high level (behavioral) synthesis or system level synthesis. Market forces are demanding higher and higher levels of synthesis. Due to the advantage of eliminating the burden of coding the algorithm by the designers, system level synthesis is most desirable.

Digital synthesis is quite successful, many commercial tools (such as SYNOPSIS$^{TM}$) covering all levels of design hierarchy are used widely. On the other hand, analog synthesis lags behind its digital counterparts by a large step. Although there are some analog synthesis tools reported, design automation tools are traditionally sparse. This is partly due to the complexity of analog circuits and the dependence of the performance on many factors such as parasitic, layout, sizes, process, etc. Usually a synthesis tool cannot live without design knowledge of the domain, so developing analog or mixed-signal synthesis tools that apply to all processes and all kinds of analog circuits is unrealistically difficult. That's why many analog synthesis tools are limited to one specific structure or circuits

[38]-[42]. Hierarchical synthesis methodologies are used to synthesize and simulate complex circuits [39]. There are many kinds of synthesis methodologies, according to their synthesis procedure, we can simply classify them into two kinds: indirect and direct methods which are overviewed below.

### 2.2.1 Indirect Methods

Generally speaking, digital high level synthesis or behavioral synthesis, proceeds from a high level behavioral description into a technology-independent data flow/control graph representation. This graph is then subjected to a set of complex operations, resulting in a data path and controller design represented as a set of register-transfer level components. The RTL representation is bound to modules in an implementation technology, such as standard cell libraries. A variety of such systems have been developed. These systems can be classified according to their different representation of the control and data-flow graph and the scheduling and allocation algorithms.

Based on the representation of the control and data-flow graph, several different implementations exist: CMU's Value Trace [43], USC's Design Data Structure [44], The Yorktown Internal Format [45], Irvine's Behavioral Intermediate Format [46], IBM's Sequential Synthesis In-core Model [47], Stanford's Sequential Intermediate Form [48], etc. A list of tools is shown in table 2.1.

In analog and mixed-signal circuits synthesis, there are also similar indirect methods. It is obvious that trade-offs exist between the spectrum of application domain and the efficiency and effectiveness of the algorithms. The more the domain covers, the less the efficiency and effectiveness of the algorithms. This is due to the fact that the design knowledge of each domain differs drastically. That's why many synthesis systems are focused on a specific analog domain, like Opamps, Conerters, Filters etc. It's more practical to develop synthesis tools for a specific analog or mixed-signal architecture. Signal Flow Graph (SFG) is an indirect way reported in literature ( [49], [50],[51]). The

Table 2.1 Some digital synthesis tools

| Tools/Research | Input Desc. | Intermediate Representation | Synthesis Algorithm |
|---|---|---|---|
| CATHEDRAL(Leuven Belgium) | SILAGE | DSFG(Behavior), ANL(Structure), LIB(Knowledge) | |
| IBM HIS | VHDL | BDL/CS, CFG, DFG | ASAP Scheduling; Path-based Allocation : clique and coloring approach |
| MICON ( CMU) | User Spec | Control Graph Condition Vectors | SFF, SFS. |
| Cyber (NEC) | C, BDL | Control Data Flow Graph and Structured Control Graph | CVLS Scheduling |
| Hercules and Hebe (Stanford) | HardwareC | SIF(Sequential Intermediate Form) | Resource Binding; Scheduling. |
| PARTHENON ( NTT) | SFL | Module, stages, Messenger. | |
| HAL | C-like | CDFG | Force-directed Scheduling |
| THEDA ( Taiwan Tsinghua) | ABDL | Internal Data & Control Flow | Constrained Scheduling (ASAP, LIST, ALAP, ILP) |
| System Architect's Workbench | ISPS, Verilog, VHDL | Value Trace | List Scheduling |
| Unified System Construction Project(USC) | Task Data Flow Graph | Mathematical Programming | 3 D Scheduling ( Time and X-Y plane ) |
| Princeton PUBSS | VHDL | BPSM | New Scheduling and Allocation |

algorithm description is translated into an intermediate SFG, then a pattern recognition is applied to the SFG and select function blocks from a library. The topology is generated according to the function blocks. Some simplification steps are applied to the intermediate SFG in order for the pattern recognition. Figure 2.17 shows a procedure that employs SFG. In indirect methods, algorithms have to be written by designers who are familiar with the circuit structures and know the details of the circuits. The writing of the algorithms is time consuming and need expertise. In addition, the algorithm itself has already fixed the circuit structure.

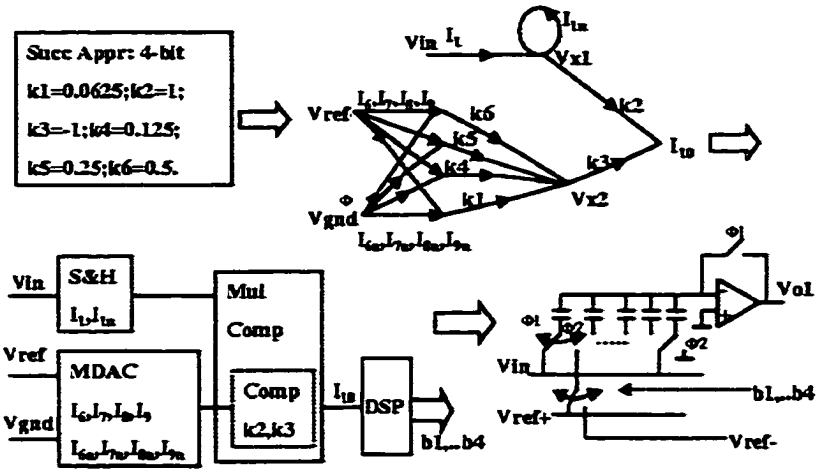


Figure 2.17 Synthesis by manipulating signal flow graph

### 2.2.2 Direct Methods

Direct methods are more open to users, by allowing users to specify the system level requirements to the circuits. Users can be relieved from the burden of writing the detail algorithmic descriptions. The system level specifications are mapped directly into a satisfying structure with proper implementations [52],[9]. Figure 2.18 shows a functional graph that is used to synthesize an ADC directly from the specifications. In a direct mapping method, an ADC can be represented by a functional graph. The synthesis algorithm will explore the functional graph according to the specifications. At first, all the necessary function sub-blocks are selected. Then for implementation of each sub-block, one of several candidate components is selected. Determination of the best choice is according to specifications and their derivations. After the components are selected, the associated structural information is used to construct topology. This looks straightforward, but every step of exploration need some design knowledge support.

Direct mapping methods need the functional topology graph and the design knowledge incorporated into the selection procedure. These can be accumulated when the

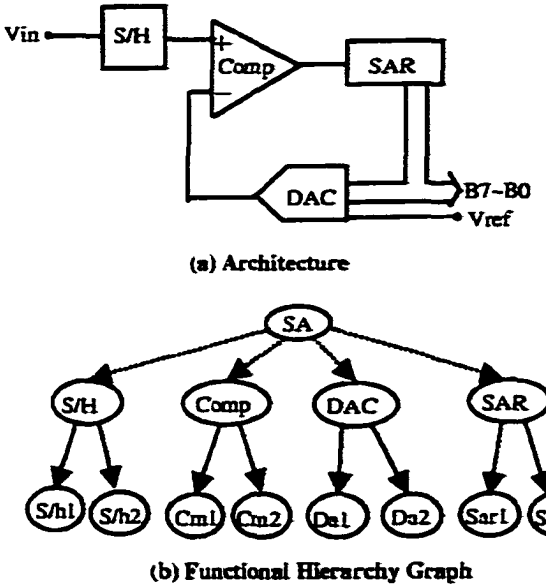


Figure 2.18 Direct mapping method

building blocks are created. The system can also be extended gradually. For a commercial company with many years of experience, the accumulation result could be a very comprehensive system. Direct mapping is very effective for modular structures since concerns about the coordination problems between function blocks are considerably reduced.

## 2.3 Summary of Review

In this chapter, many DAC architectures and synthesis methods for digital, analog and mixed-signal circuits were reviewed. We are pursuing high-speed, high-resolution DACs, so the current steering structure is the candidate of choice. For high resolution, calibration is a must. Modular design allows easiness of reconfigurability, low power and small area. These lead us to modular current-steering DAC architectures with self-calibration. Direct-mapping methodology is a good candidate for a DAC synthesis algorithm after the module library has been created and the knowledge for the DAC

configurations have been derived.

# CHAPTER 3. A RECONFIGURABLE DAC ARCHITECTURE

In this chapter, the DAC architecture will be selected. The major objectives are choosing DAC architectures with high-speed, high-resolution, low-power using digital CMOS processes and easy to synthesize. Speed and accuracy limitation factors are to be studied. Theoretical derivations will be given for various design parameters.

Current steering architecture is a good candidate for DACs with several hundreds of MS/s conversion rates. A traditional current steering DAC architecture is shown in Figure 3.1. It uses $2^n - 1$ unary current cells for an n-bit DAC. Large numbers of row and column decoders are needed when n increases. Although layout techniques can be used to achieve good matching and high accuracy, die area is increased due to complex routing and power dissipation is big due to complex decoders logic.

In order to achieve the above goals, we propose a reconfigurable DAC architecture shown in Figure 3.2. The architecture includes several sub-DAC blocks, multipliers and summations. If the sub-DAC blocks A, B and C are the same 4-bit DAC and the multipliers have the same ratio $\frac{1}{16}$ then they can be combined together to give a 12-bit DAC. This structure is scalable. More bits can be added by simply adding blocks assuming the accuracy of the blocks is good enough. When more blocks are available, the synthesis of new DACs is straight forward. Figure 3.3 shows another structure assuming each block uses the same unit current cell. One advantage of the architecture is that the segmentation simplifies the decoding logic circuits and reduces the power

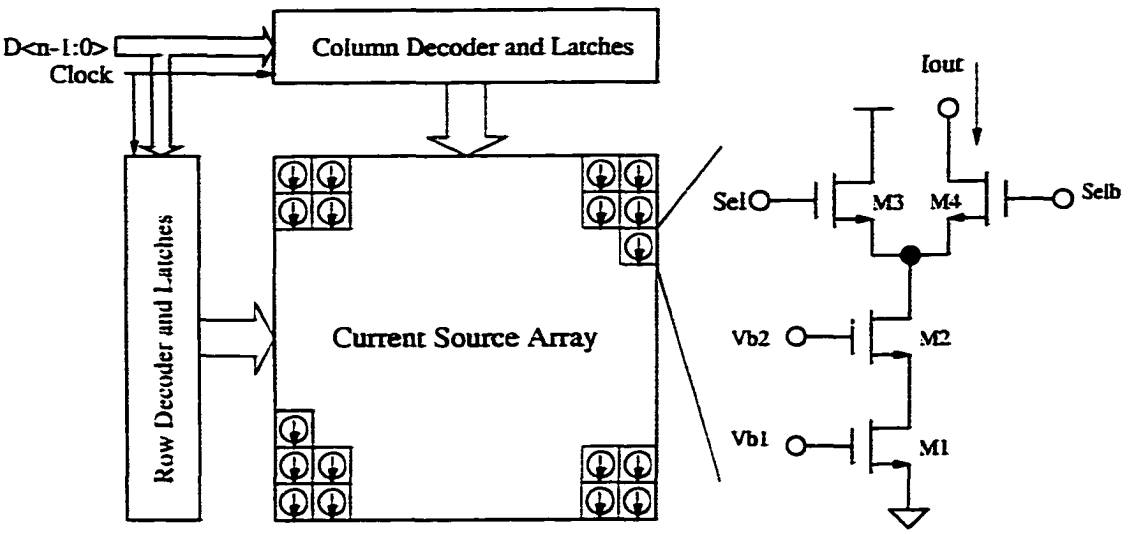


Figure 3.1 A current-steering DAC

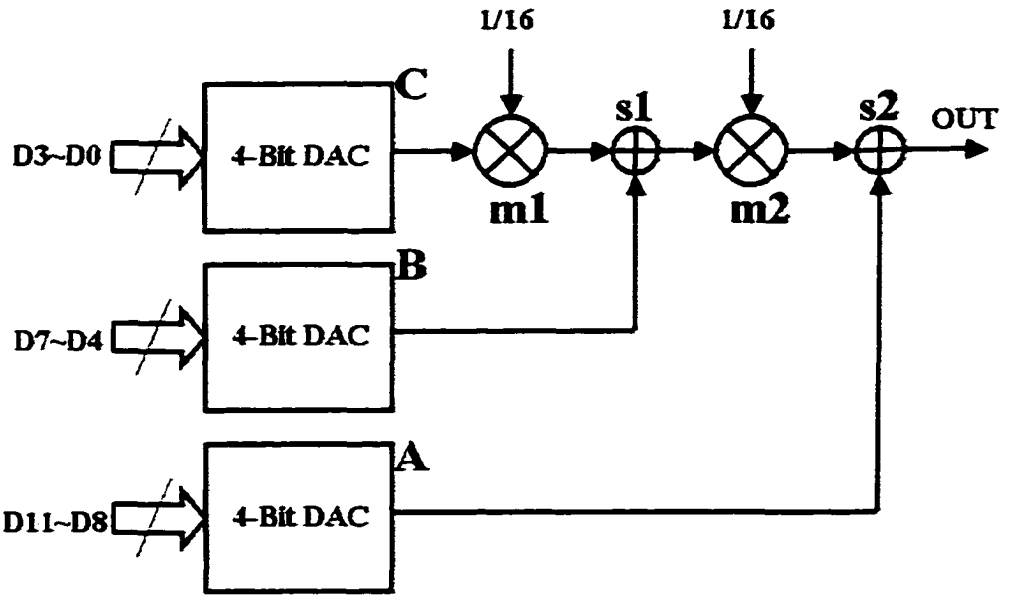


Figure 3.2 DAC architecture that allows reconfigurability

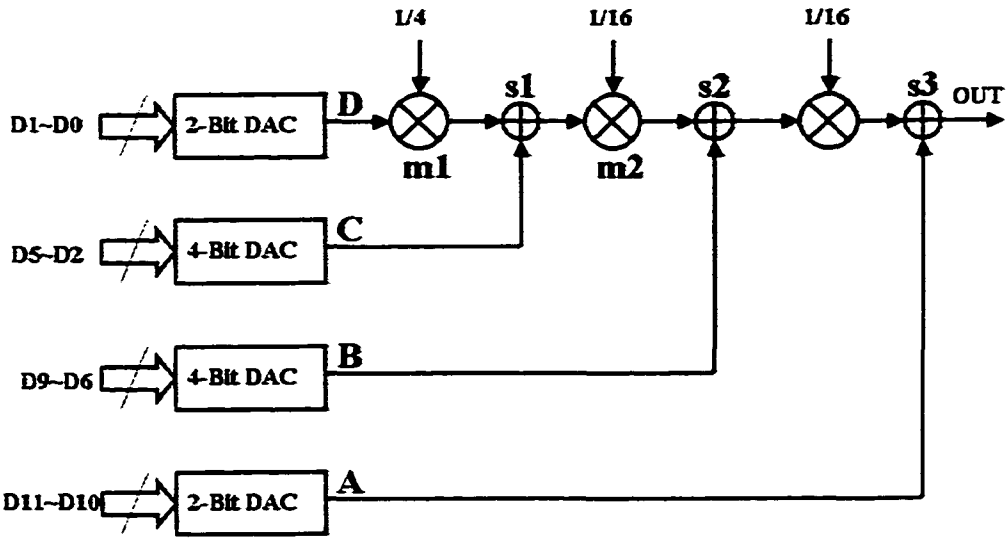


Figure 3.3 Reconfigurable DAC with more flexibility

dissipation, another advantage is easy to reconfigure and synthesize. If accurate sub-DAC blocks and ideal multipliers with zero delay and accurate scale factors of $\frac{1}{2^m}$ are easy to construct, then we could extend this architecture to as many bits as we want. In reality, limitation of accuracy in each sub-DAC block and timing cooperation of the multipliers and summation blocks imposes various problems. The delay of the multiplier circuits make the time from different sub-DAC blocks to the final output different, this will introduce glitches. To resolve this problem, we could introduce some delay components to the faster path, but it is not a trivial task to exactly control the delay. Even if we could introduce some delay components and give every path the same delay, when we cascade many stages, the settling time will become too long to be acceptable.

Multiplier function can be incorporated into sub-DAC blocks by arranging the weight of their unit current cells. This idea is preferred since the problem caused by the multiplier blocks mentioned above can be solved. The architecture is shown in Figure 3.4. For current mode DACs, the summation function of currents can be realized by just connecting all the current outputs together to the output load resister. This architecture

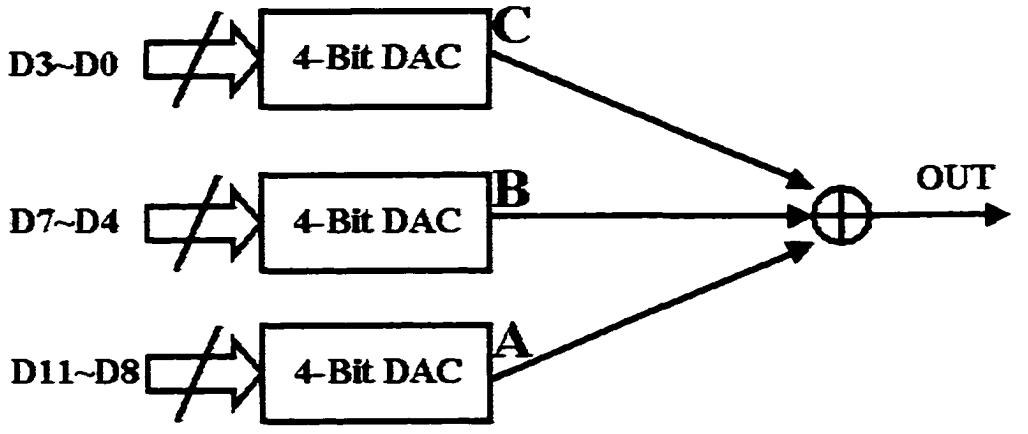


Figure 3.4 A practical reconfigurable DAC architecture

also allows flexible synthesis of DACs with different requirements.

## 3.1 Accuracy of Sub-DAC Blocks

For a modular architecture, accuracy requirements on different blocks are different. This can be shown by a simple example. For simplicity, consider an 8-bit DAC constructed by two 4-bit blocks A and B. Without losing generallity, assume: (1) blocks are independent and each block is a unary weighted 4-bit block. (2) block A is the 4-bit MSB block and block B is the 4-bit LSB block. (3) The nominal value of the unit current in block B is 1. It is obvious that both block A and block B have 15 unit current souces. Each unit current source for block A has a nominal value of 16. Assume the relative error of the current sources in block A and B are $e_A$ and $e_B$ respectively. The worst case is all the errors are in the same sign ( all are positive or negative). Then we have the total error represented by equation 3.1.

$$E_{total} = 15 * 16 * |e_A| + 15 * |e_B| \tag{3.1}$$

$E_{total}$ needs to be less than $\frac{1}{2}LSB$, which is half of the unit current in Block A. Based

on assumption (3), $E_{total} < \frac{1}{2}$. If only block A has errors, $e_A$ has to satisfy eqution 3.2. So the accuracy of block A needs to be at least 8-bits.

$$|e_A| < \frac{1}{2 * 15 * 16} \approx 2^{-9} \tag{3.2}$$

Similarly if only block B has errors, $e_B$ has to satisfy eqution 3.3. The accuracy of block B needs to be at least 4-bits.

$$|e_B| < \frac{1}{2 * 15} \approx 2^{-5} \tag{3.3}$$

In reality, both blocks have errors, the error budget for each block is less than $\frac{1}{2}LSB$. If the error budget is equally shared, then the accuracy of blocks A and B needs to be 9-bits and 5-bits respectively. If the MSB block is very accurate, then the LSB block can be relaxed. Due to the nominal value of the unit current in block A is 16 times that that in block B, so the error contribution of block A is larger and its accuracy needs to be higher.

In Figure 3.4, block A requires the highest accuracy, while block B and C require moderate and lowest accuracy correspondingly. Self-calibration is needed for 10-bits or higher accuracy, so block A is a self-calibrated block. 4-Bit block A is shown in Figure 3.5. One reference current is used to calibrate 16 current cells, one of which is a redundant cell. This is controlled by the shift-register loop. A binary to thermometer code (B2T) decoder is used to reduce the glitch caused by major bit switching ( eg. 0111 to 1000). The input 4-bit binary code is converted to thermometer code and select at most 15 unit current cells through the switching network. The reference currents block is constructed by an array of the RGC mirrors to pursue high output impedance and accuracy.

Block B needs moderate accuracy, a thermometer code decoder is used without self-calibration. The accuracy relys on matching of the unit current cells. Block B is shown in

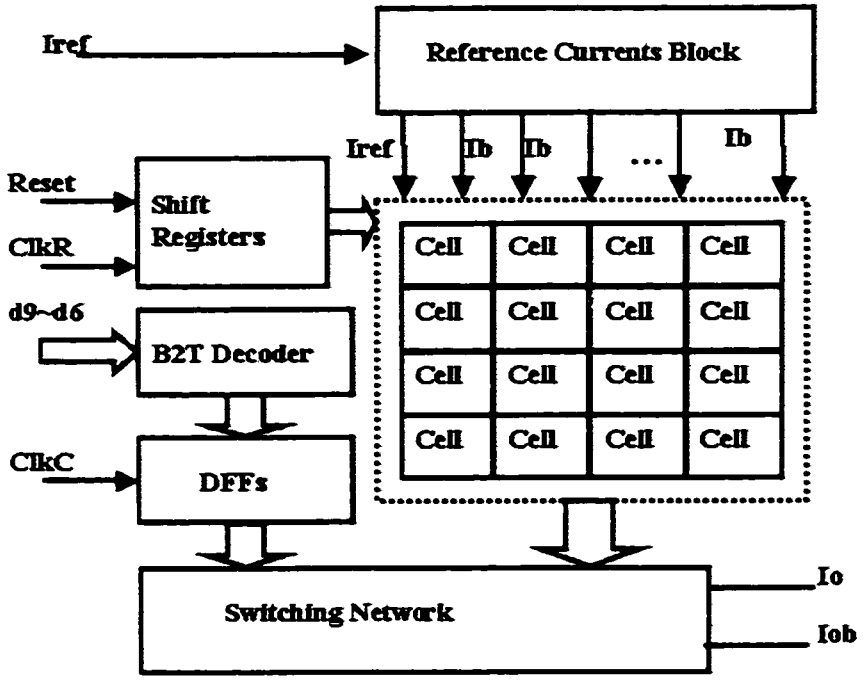


Figure 3.5 4-bit sub-DAC with self-calibration

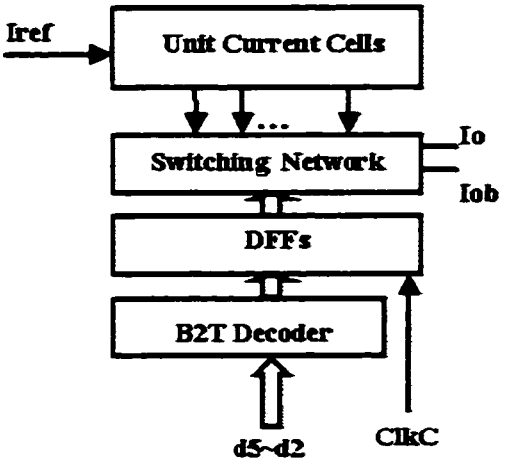


Figure 3.6 4-bit sub-DAC block without self-calibration

Figure 3.6. Block C is the lowest bit block, so simple decoding binary-weighted decoding logic is used. Block C can be either designed to have 2-bit or 4-bit to support different configurations. Figure 3.7. shows the 2-bit version of block C. By having these cells, we can easily configure a 6-bit, 8-bit, 10-bit, and 12-bit DAC.

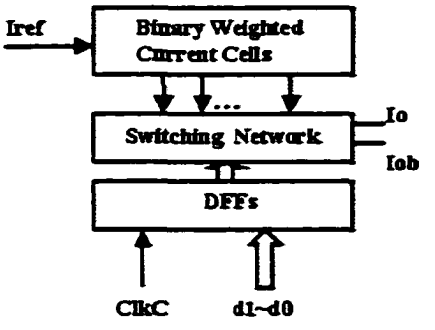


Figure 3.7 Least significant 2-bit block

The realization of the architecture relys on solutions to two issues. One is the accurate unit current cell within a sub-DAC block, the other is unit currents ratio between different blocks. Self-calibration is needed for achieving high accuracy current cells. The literature show that 16-bit accuracy can be obtained in audio frequency (44.1KHz) [6]. The accuracy is degraded when the sampling speed is increased to video frequency or several hundreds of MHz, because the settling time is greatly reduced. The settling is limited by many factors such as the RC constant of the output load, nonidealities of MOS switches, parasitic capacitors, charge injection etc. The consideration to the realization of accurate current cells and settling limitation factors are discussed next. Of course, the unit current sources of the sub-DAC blocks need to be matched. Current bias blocks should be designed carefully to satisfy matching requirements. In addition to matching, limitation factors to speed and accuracy need to be addressed properly.

## 3.2 Current Cells

For a current steering DAC, the output impedance of a current cell needs to be as high as possible. This can be shown from the simplified circuit model of the output state. In Figure 3.8, assume the current of the cell is $I_0$ and the output impedance of one current cell is $R_0$, and m cells are connected to a load resistor $R_L$. The output voltage $V_0$ is represented by equation 3.4.

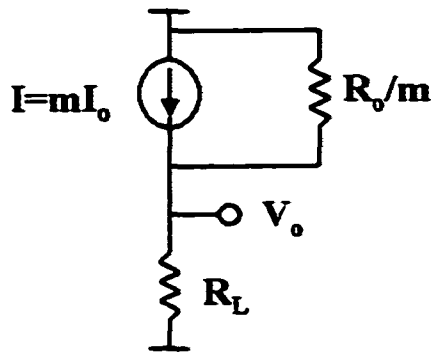


Figure 3.8 Output state current cells with limited output impedance

$$V_o = mI_0R_L\frac{R_0}{R_0 + mR_L} \tag{3.4}$$

If the current sources are ideal, $R_0$ is infinity, the ideal output voltage is represented by equation 3.5.

$$V_{oideal} = mI_0R_L \tag{3.5}$$

The relative error $\epsilon$ is defined as:

$$\epsilon = \frac{V_o - V_{oideal}}{V_{oideal}} \tag{3.6}$$

Substitution of equations 3.4 and 3.5 into 3.6 yields equation 3.7.

$$\epsilon = \frac{-mR_L}{R_0 + mR_L} \tag{3.7}$$

Assuming $R_L = 50$ Ohm and the 4-bit block uses 15 unit current cells (m=15). In order to achieve 12-bit accuracy, the absolute value of the relative error must be less than $2^{-13}$. This requires $R_0$ to be larger than $(2^{13}-1)*m*R_L = 6.14Mohm$. If we use a 2-bit block, which has 3 unit current cells (m=3), the requirement on output impedance is $R_0 = 1.23Mohm$. So in modular design, using less bits in first block relaxes to the current cell output impedance requirements.

A simple current mirror as a current cell is shown in Figure 3.9. When N1 and N2 are both in saturation and equally sized, $I_{out} = I_{ref}$. The output impedance of this mirror is approximately $R_o = r_{ds2}$. $r_{ds2}$ can be represented by equation 3.8, where $\lambda$ is inversely proportional to the length of transistor N2 [53]. So in order to get high output impedance, long channel transistors are needed.

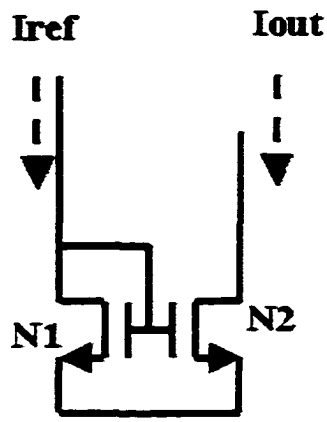


Figure 3.9 Simple current mirror

$$r_{ds2} = \frac{1}{\lambda I_{ref}} \tag{3.8}$$

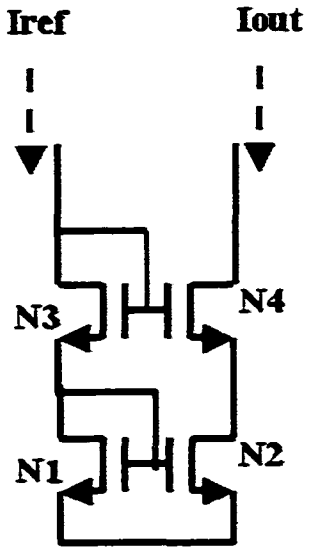


Figure 3.10 Cascode current mirror

A cascode current mirror is shown in Figure 3.10. The output impedance is increased to $R_o = g_{m4} r_{ds2} r_{ds4}$. The output voltage that keeps transistors N2 and N4 in saturation is equal to $2V_{dsat}$, this will also impact the signal swing of the output load.

As calculated above, the higher the output impedance of a current cell, the less the accuracy loss. At the same time, the output voltage that keeps the current mirror in saturation should be as low as possible. So far, the current mirror with the highest output impedance is a regulated cascode (RGC) mirror shown in Figure 3.11. N3-N5 constructs the RGC circuits [54]. N1, N2, P1 and P2 are just simple current mirrors for biasing the RGC circuits. The operating principle of the RGC circuits is described below.

N4 converts its gate voltage into a current through N5. In order to get a high output impedance, the drain-source voltage of N4 should be kept stable. A negative feedback loop is constructed by N3 and N5. When the drain voltage of N4 increases, then amplifier N3 will cause its drain voltage to decrease, then the source following N5 will cause its source voltage to decrease, which is same node as the drain of N4. By this

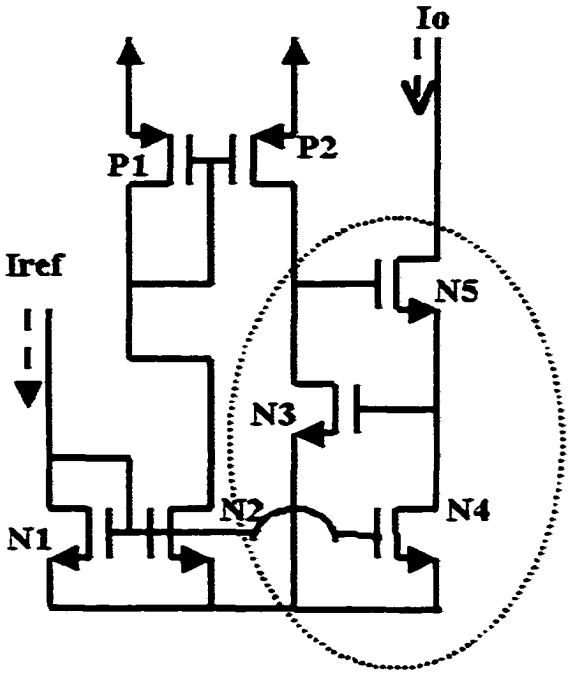


Figure 3.11 Regulated cascode current mirror

regulating function, the drain voltage of N4 is fixed to the same value of drain voltage N1. Adjusting the size of N3 can make this happen. The mirror's output impedance is represented by equation 3.9. It is on the order of $g_m^2 r_o^3$, while normal cascode mirrors only have $g_m r_o^2$.

$$r_{out} = \frac{1}{g_{o4}} \frac{g_{m3} g_{m5}}{g_{o5}(g_{o3} + g_{o_{p2}})} \tag{3.9}$$

Even if transistor N5 goes into the ohmic region, the regulation function still works, so the minimum output voltage of the RGC can be a little bit above one transistor's Vdsat. In summary, the regulated current mirror has the following characteristics: high output impedance and high output swing. Minimum length transistors can be used to reduce area and parasitic capacitances associated with the transistors. But for better matching, N4, N1 and N2 should avoid minimum length transistors, this also applys to

P1 and P2.

## 3.3 Limitation Factors to High-Speed and High-Resolution

As discussed in the previous section, the output impedance of a current mirror is a limiting factor to the output accuracy. For self-calibrated current cells, charge injection and leakaging are also important limiting factors to the accuracy. Speed limitation factors include the digital circuit decoding speed, flip-flop delays, the settling of the current cells, switches and their parasitic capacitance, load resistors and parasitic capacitances. In submicron processes, digital circuit speeds can easily go up to GHz ranges, so as long as overly complex decoding logic schemes and critical long paths are avoided, the digital circuitry should have no problem handling the speeds needed here. Settling of the current output is critical for achieving the high-speed and high-resolution. These factors are discussed as follows.

### 3.3.1 Charge Injection and Leakaging

As discussed in section 3.2, the accuracy of the output current is limited by the output impedance. For self-calibrated current cell, this limitatation is extended to the output impedance, charge injection and leakaging. Figure 3.12 depicts a self-calibrated current cell with clock-feedthrough charge compensation.

Cal and Sup are a apir of complementary clock signals. When Cal=1 and Sup=0, N1 and P3 are on, N2 is OFF, the current cell is calibrated to the reference current Iref so it is in calibration mode. During the calibration mode, the current Iref-Ib flows through the transistor P1 and determines its gate voltage. When Cal=0 and Sup=1, N1 and P3 are OFF, N2 is ON, the current cell is in supplying mode and supply its current equal to Iref to the output. During the calibration mode, the charge in the conducting channel of P3 is represented by equation 3.10. When the current cell switches from the

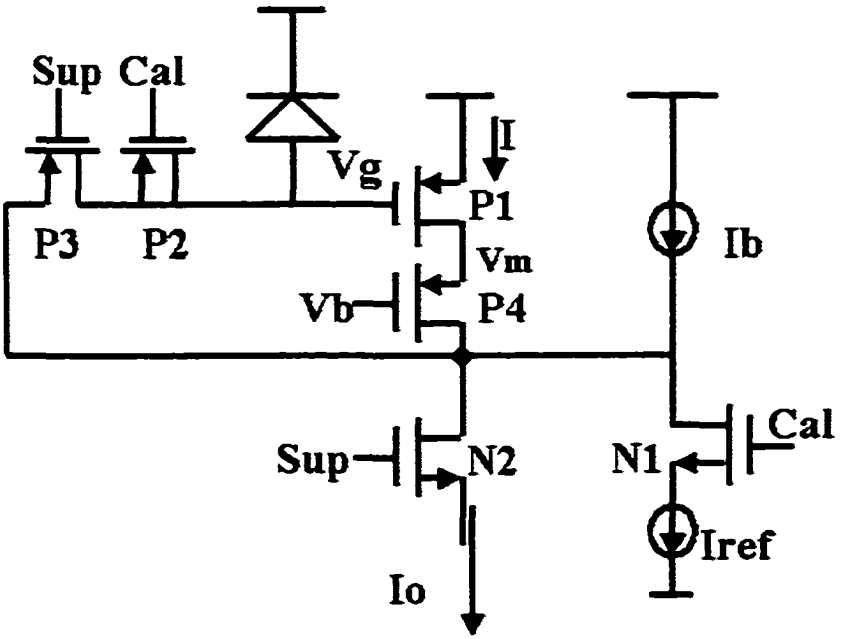


Figure 3.12 Self-calibrated current cell with charge compensation

calibration mode to the supplying mode. part of the charge in the P3 channel will be released to the capacitor of the gate node of P1. this will cause a change of gate voltage of P1 and to the current through drain of the P1. Several measures could be used to reduce this error: Selecting a small area of P3 to make $Q_{inj}$ small: using a half-sized dummy transistor P2 switched on a complementary clock cycle to absorb most of the charge released from P3: Increasing the area or gate node capacitance of transistor P1 to reduce the error impact to the gate voltage change: and reducing the current I through P1 to a small percent (normally 10%) of the total reference current Iref to reduce the charge injection error impact on the total output current.

$$Q_{inj} = W_3 L_3 (V_{sg3} - |V_{tp}|) \tag{3.10}$$

During the supplying mode. P3 is OFF but the leakage current through the reverse-biased diode between the drain of P3 and power supply will change the gate voltage

of P1 continuously as shown in Figure 3.13. Assume a continuous self-calibrated DAC block includes m+1 curent cells( one cell is redundant). Assuming Each cell repeats its calibration mode in m+1 calibration clock cycles. m is the maximum number of supplying cells. T is the refreshing clock cycle. In calibration mode, a current cell is calibrated to the reference current Iref. In supplying mode, the current cell's output current can be represented by equation 3.11.

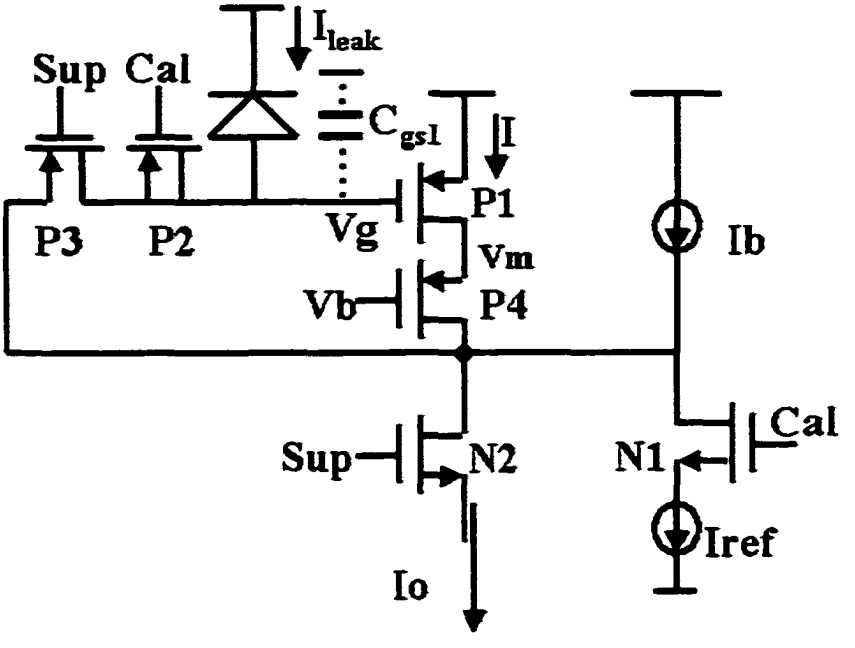


Figure 3.13 Self-calibrated current cell with leakage

$$I_o(t) = I_{ref} - g_{m1}\frac{I_{leak}}{C_{gs1}}(t - ((m+1)n+1)T) \tag{3.11}$$

Where, $I_o(t)$ is the supplying current of the cell. $((m+1)n+1)T < t < (m+1)(n+1)T$, n=0,1,2,.... . This is shown in Figure 3.14.

The leakage current $I_{leak}$ can be approximated as in [53]:

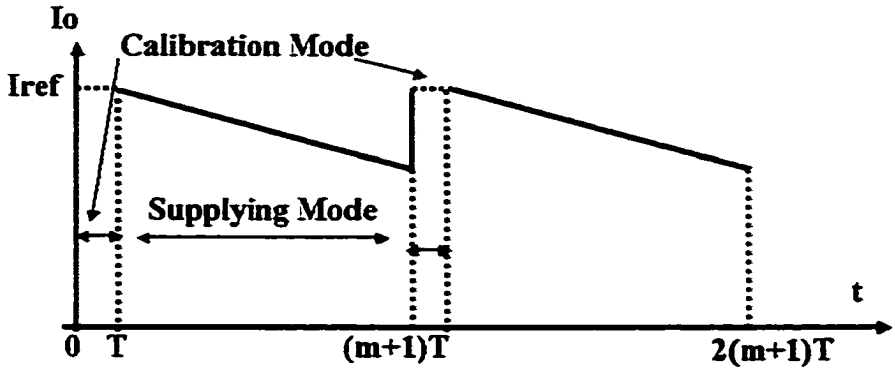


Figure 3.14 Leakage effect

$$I_{leak} = \frac{qA_j n_j}{2t_0} x_d \tag{3.12}$$

Where $A_j$ is the junction area of the diode, $n_j$ is the intrinsic concentration of carriers in undoped silicon, $t_0$ is the effective minority carrier lifetime, and $x_d$ is the thickness of the depletion region.

In its saturation region, P1's transconductance $g_{m1}$ and its gate-source capacitance $C_{gs1}$ are represented by equations 3.13 and 3.14.

$$g_{m1} = \sqrt{2\mu C_{ox} \frac{W_1}{L_1} I} \tag{3.13}$$

Gate-source capacitance Cgs1 is:

$$C_{gs1} = \frac{2}{3} W_1 L_1 C_{ox} \tag{3.14}$$

Substitution of equations 3.12, 3.13, 3.14 into 3.11 yields:

$$I_o(t) = I_{ref} - \frac{3}{4}\sqrt{\frac{2\mu I}{C_{ox} W_1 L_1^3}} \frac{qA_j n_j}{t_0} (t - ((m+1)n+1)T) \tag{3.15}$$

Where $((m+1)n+1)T < t < (m+1)(n+1)T$, n=0,1,2,.... In order to reduce the

leakage error, we can choose a small value of I and Aj and a large size for P1 (W1 and L1, especially L1). Other parameters are process related. A PMOS is preferred due to its lower mobility value $\mu$. Equation 3.15 provides some guideline to selecting W1 and L1.

Consider a 4-bit MSB block with continuous self-calibration, m=15. In the worst case, up to 15 current cells can be supplying output at the same time. Since the current cells are calibrated one-by-one, at any time t ( $(16n+1)T < t < 16(n+1)T$ ), there would be one supplying cell at the time just between (16(n+1)-1)T and 16(n+1)T, another supplying cell is at the time just between [16(n+1)-2]T and (16(n+1)-1)T, a third supplying cell is at the time between [16(n+1)-3]T to (16(n+1)-2)T , and so on. So the maximum error of output current $I_{emax}$ due to the leakage is:

$$I_{emax} = 90\sqrt{\frac{2\mu I}{C_{ox}W_1L_1^3}}\frac{qA_jn_j}{t_0}T \tag{3.16}$$

If we define a parameter Relative Leakage Rate(RLR) as in equation 3.17, then $I_{emax}$ can be represented as in equation 3.18. In order to make the leakage small enough to be negligable, equation 3.19 needs to be satisfied. For a 10-bit segmented structure with self-calibrated 4-bit MSB block, the LSB current is equal to $\frac{I_{ref}}{2^6}$.

$$RLR = \frac{g_{m1}I_{leak}}{C_{gs1}I_{ref}} \tag{3.17}$$

$$I_{emax} = 120 * RLR * T * I_{ref} \tag{3.18}$$

$$I_{emax} = \frac{1}{2}LSB \tag{3.19}$$

The minimum frequency of the calibration clock that limits the maximim leakage error within half LSB is derived in equation 3.20. In general, for a n-bit DAC with a

self-calibrated m-bit MSB, the minimum frequency of the calibration clock is shown in equation 3.21.

$$F_{min} = 15 * RLR * 2^{10} \tag{3.20}$$

$$F_{min} = (2^m - 1) * RLR * 2^n \tag{3.21}$$

Care should be taken to satisfy the saturation conditions of P1 and P4. The equations 3.22, 3.23, 3.24, 3.25 need to be satisfied at the same time.

$$0 < V_{dd} - V_g - \mid V_{tp} \mid < V_{dd} - V_m \tag{3.22}$$

$$\frac{\beta_1}{2}((V_{dd} - V_g) - \mid V_{tp} \mid)^2 = I \tag{3.23}$$

$$\frac{\beta_4}{2}((V_m - V_b - \mid V_{tp} \mid)^2 = I \tag{3.24}$$

$$0 < V_m - V_b - \mid V_{tp} \mid \leq V_m - V_g \tag{3.25}$$

In equations 3.22 to 3.25, $\beta_i = \mu C_{ox} \frac{W_i}{L_i}$, i=1,4, and $V_{tp}$ is the threshold voltage of PMOS transistors. I is the drain current of P1 and P4. From 3.23, given I, the W1/L1 ratio of P1 can be adjusted to obtain $V_g$. According to 3.22 and 3.25, the range of $V_m$ and $V_b$ can be obtained. 3.24 can be satisfied by adjusting W4/L4. $V_b$ should be set as high as possible to have the maximum value of the drain voltage of P4. This will set the highest voltage at the drain node of P4 that maintains P1 and P4 in saturation during supplying mode as well as the highest voltage swing at the output.

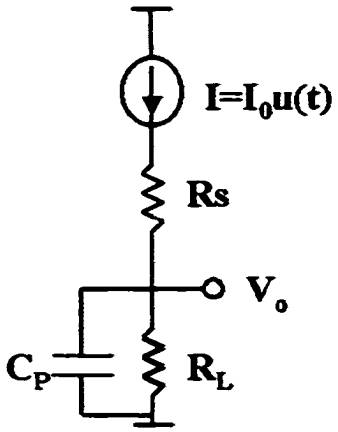


Figure 3.15 Simplified circuit model for DAC output

### 3.3.2 Settling Speed

Resistor load and parasitic capacitances at output nodes and switching are major limiting factors for settling. A simplified circuit model is shown in Figure 3.15.

A step response for an RC circuit is represented by equation 3.26. Generally speaking a pad capacitance is in the pF range. Without losing generality, assume a load capacitance of 10pF. If the output load resistance is 50 Ohm, then the time constant is $\tau = R_L * C_P = 0.5ns$. Within 4ns, a step response can settle to $1 - e^{-8} = 0.9996645$, the error is within $\frac{1}{2}$ LSB of 10-bit. If we want to achieve 12-Bit accuracy within 4 ns, the parasitic capacitance should be less than 8.87pF. To be conservative, settling to 12-Bit accuracy needs to be within 2ns, that will require less than 4pF $C_P$. This can be easily satisfied in embedded applications.

$$V_o(t) = I_0 R_L (1 - e^{\frac{-t}{R_L C_P}}) \tag{3.26}$$

In addition to the resistor load and parasitic capacitances, care should be taken to design the switches that steer the currents from the current cells to the output load smoothly. PMOS transistors are widely used for output switches as shown in Figure 3.16.

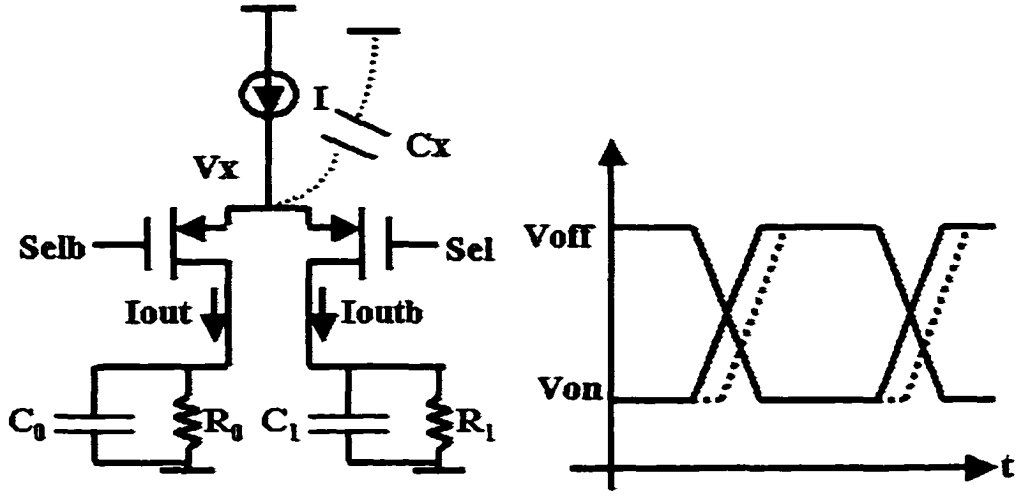


Figure 3.16 PMOS switches and switching signals

It was found that lowering the cross-point of the selection control signals can reduce the fluctuations of Vx and cause settling faster than conventional simultaneous switching ([? ],[56]). However, charge injection still limits the settling, since the charge injection has a sign that opposes the desired transition of the output [57]. By using NMOS transistors as output switches, charge injection can be utilized to speed-up settling. Figure 3.17 shows the NMOS switches used in the prototype to be described in next chapter. Assume N0 is to be switched, the charge injected by clock feed-through is represented by equation 3.27.

$$Q_{inj} = \frac{1}{2} C_{ox} W L (V_{gs} - V_t) \tag{3.27}$$

A charge needed for one LSB is represented by equation 3.28.

$$Q_{lsb} = I R_L C_P \tag{3.28}$$

If $Q_{inj} = Q_{lsb}$, then the output will quickly settle to the expected value. So proper sizing of NMOS switches and the level of the switching signals can achieve quick transition and optimal settling. Simulations were performed for comparison. The current cells

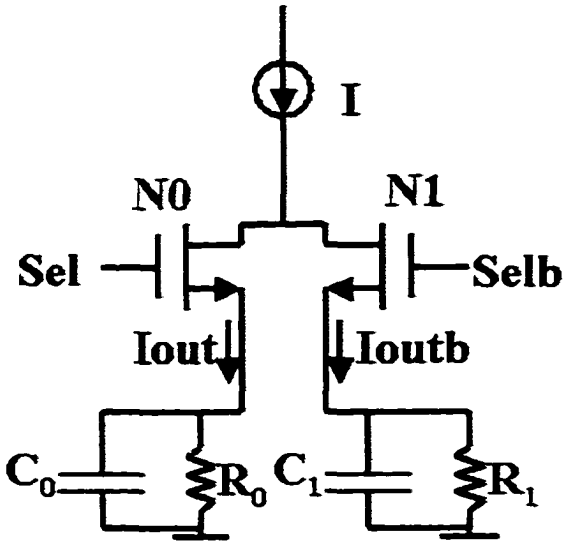


Figure 3.17 NMOS switches

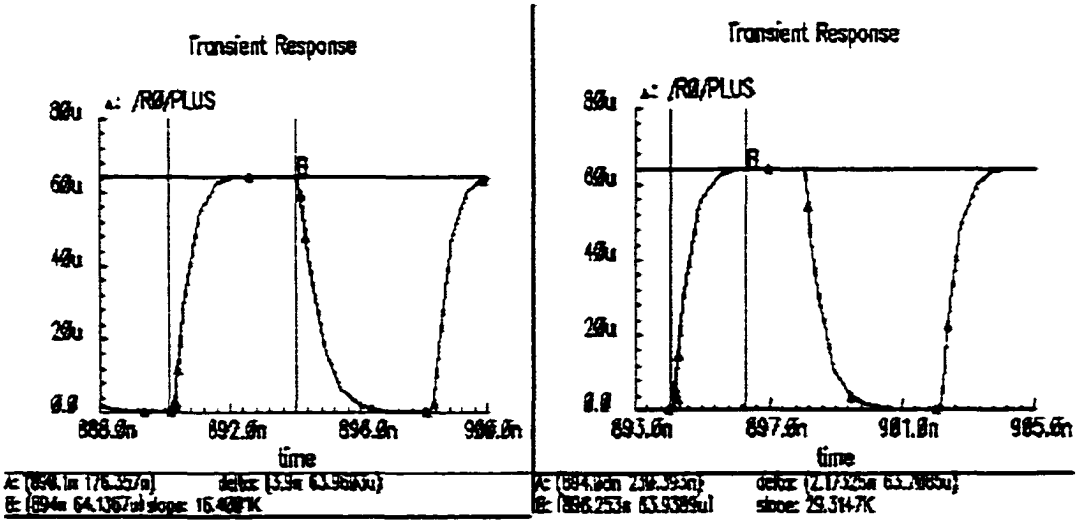


Figure 3.18 Simulation results for different switches

in Figures 3.16 and 3.17 are the same self-calibrated current cells with current of 64uA. The output loads are R0=R1=50, C0=C1=10pF. Figure 3.18 shows the simulation results. For the PMOS case (left), the output current settles to 0.1% in more than 3.9ns, while for the NMOS case (right), it settles within 2.17ns. This shows the advantage of utilizing charge feedthrough and NMOS switches over PMOS switches for speeding up the settling.

## 3.4 Summary

In this chapter we have considered some structural issues for the reconfigurable DAC design. Speed and accuracy limitation factors were also discussed and formulae for design considerations were derived. The chapter has set up a skeleton for the design guidlines and implementation strategy. The derivation of the parameters for the design will guide the selection of the circuits and detail implementations. The detailed implementation of a prototype 8-bit 250MS/s DAC will be illustrated in the next chapter.

# CHAPTER 4. AN 8-BIT 250MS/S CONTINUOUS SELF-CALIBRATED CURRENT-STEERING CMOS DAC

In this chapter, a prototype 8-bit 250MS/s continuous self-calibrated current-steering DAC with low power and small area is described. Our objectives are high-speed, high-resolution, low-power and small-area. High-accuracy current cells are needed for achieving the above goals. The prototype used the modular DAC architectures mentioned in Chapter 3. The prototype was designed and simulated in HSPICE and fabricated in TSMC's 0.25$\mu$ single poly five metal logic CMOS process. Verifying this prototype laid a good basis for going further to other prototypes and synthesis.

## 4.1 Architecture

The goal of the design was to realize an 8-bit 250MS/s current output DAC with higher than 10-bit accuracy. The cascaded architecture is shown in Figure 4.1. In order to obtain high speed, current steering DAC is adopted. There are two 4-bit blocks in Figure 4.1, a 4-bit MSB and a 4-bit LSB block. The LSB block output current will be divided by 16, then summed to the output of the MSB. Since the output is a current, the summation is implemented simply by connecting the MSB and LSB output to the output load that includes a resistor and parasitic capacitances. The output load RC constant should be small enough in order that the output current settles to the designed accuracy within the conversion period. Based on the derivation in Chapter 3, the error in the MSB block plays a big role in the whole error budget, so the continuous self-

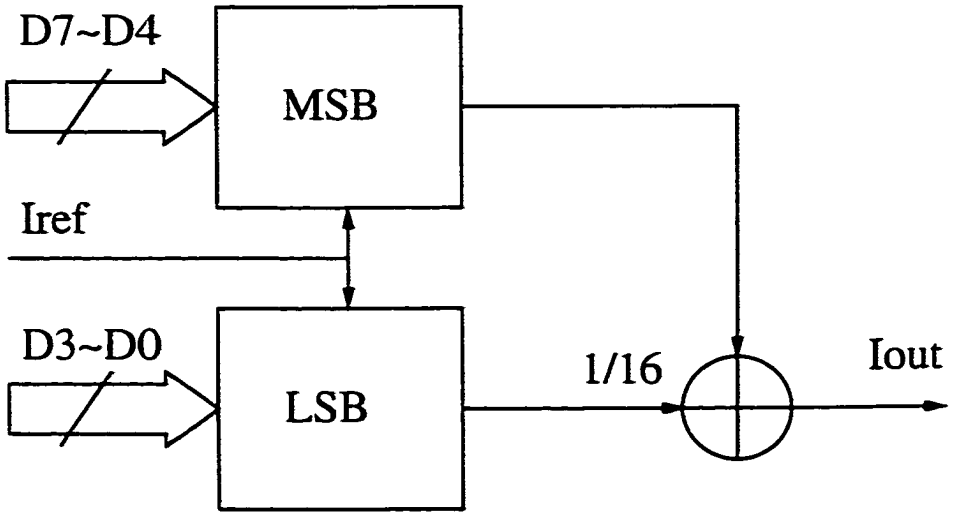


Figure 4.1 8-Bit DAC architecture

calibration technique is applied to the MSB block. We did not use self-calibration for the LSB block, since the overall error contribution there is small enough for this 8-bit DAC.

## 4.2 High-Accuracy Current Cell Structure

In current-steering DAC design, the most important issue is to achieve accurate unit current sources that are not affected by matching errors caused by process variations. Combining the dynamic element matching with a spare unit current source enables continuous current calibration. In audio speed, this accuracy can be as high as 16-bit [6]. In the literature [6], the conversion rate reported is 44.1KHz and thus the settling time allowed for each conversion is 22.67us. In high frequency applications, the accuracy will be degraded due to the much shorter settling time requirements. For example, when the conversion rate is 250MS/s, the settling time for each conversion is only 4ns. As stated in equation 3.26, the settling time must be long enough to achieve specified accuracy. Given the time constant of 0.5ns, within 4ns, the output current or voltage can only

settle up to 10-bit accuracy. In order to achieve higher than 10-bit accuracy within 4ns, smaller time constant must be guaranteed.

In order to achieve high-accuracy, current cells have to have high output impedance and the charge injection from the clock signals should be small enough to be negligible. The modification to the current cell used in literature [6] is shown in Figure 4.2. The left part shows the cell used in literature [6]. The right part is the modified cell used in the prototype here.

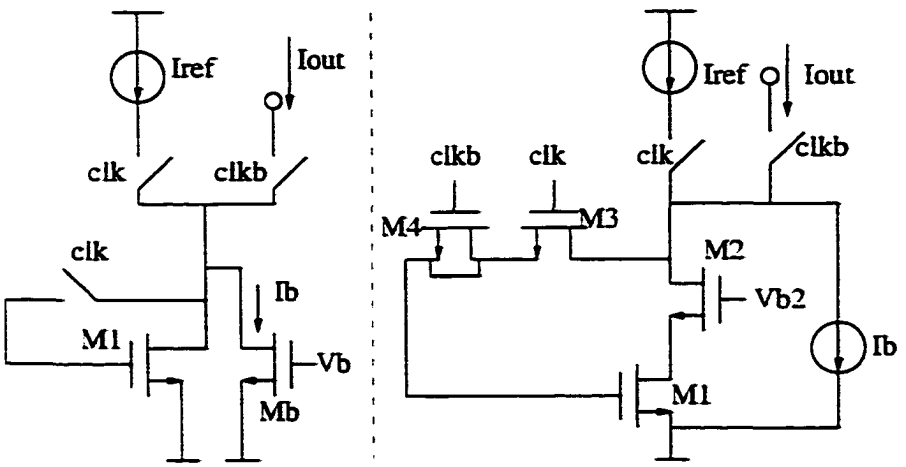


Figure 4.2 Self-calibrated current cells

It can be shown that the output impedance of the modified cell is represented by equation 4.1, where $g_{m2}$ is the transconductance of M2, $R_{o1}$ and $R_{o2}$ are output impedance of M1 and M2 respectively.

$$R_o \approx g_{m2} R_{o1} R_{o2} \tag{4.1}$$

In equation 4.1, $g_{m2}R_{o2}$ is much bigger than 1. $R_o$ is much bigger than $R_{o1}$. By increasing the output impedance, bigger output voltage fluctuation and resistor load changes can be tolerated. In Figure 4.2 (a) and (b), the charge stored on the gate of

M1 will leak when the cell goes into its supplying mode, this will cause some errors on the output. These errors can be reduced by adding a biasing current Ib which is most part of Iref (normally Ib is about 90 percent of Iref). In the worst case, the total leakage current needs to be within $\frac{1}{2}LSB$ of 10-bit (see equation 3.19 ). The biasing current Ib is realized by a regulated cascode current mirror which has very high output impedance. From equation 3.9, the output impedance of the regulated cascode current source is on the order of $g_m^2 R_o^3$.

## 4.3 Design and Implementation

In this prototype, the target current output range is about 1024uA. So the unit current cell for the 4-bit MSB is 64uA and for the LSB unit current is 4uA. The following cells and blocks are based on these numbers.

### 4.3.1 Self-Calibrated Current Cell

Although NMOS transistors are used for depicting the modification of current cell structure, PMOS transistors are used in the implementation. This is due to the fact that PMOS transistors have a lower mobility value $\mu$ than that of NMOS transistors, the leakage error is proportional to the square root of $\mu$ (see equation 3.15). The current cell used is shown in Figure 4.3. Two complementary signals Cal (representing calibrating) and Sup (representing supplying) determine the operation mode.

When Cal=1, and Sup=0, the current cell works in calibrating mode. N1 and P3 are ON and N2 and P2 (dummy) are OFF. The reference sink current Iref=64uA and source bias current Ib=57.6uA will be connected to the current cell at the same time. The current through P1 plus Ib will be equal to 64uA, which will determine the gate voltage of P1, and a proportional amount of charge will be stored on the gate of P1.

When Cal=0 and Sup=1, The cell works in supplying mode. N1 and P3 are OFF,

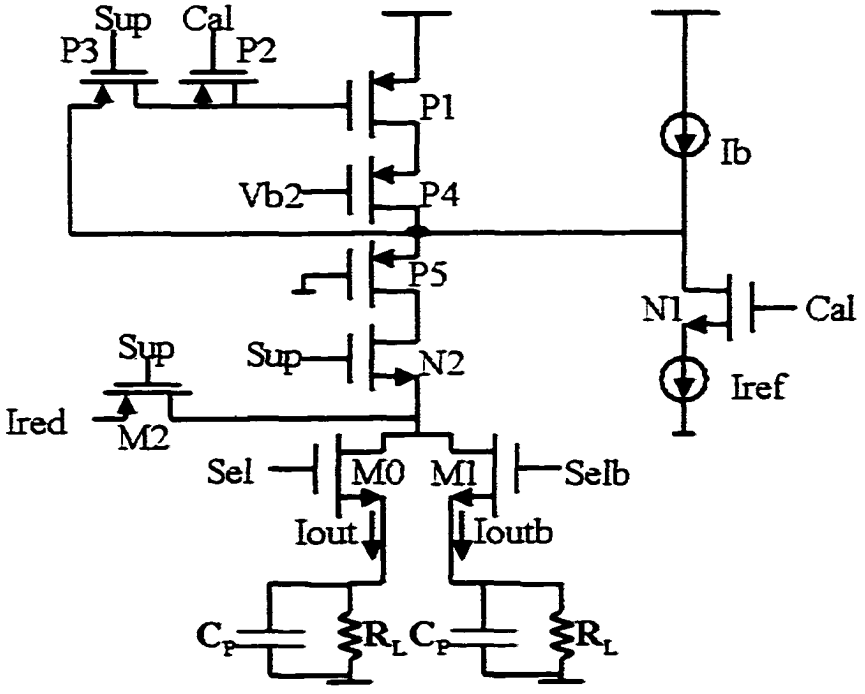


Figure 4.3 Self-calibrated current cell

N2 and P2 are on, the drain current of P1 plus Ib will be supplied to the output load connected to the source of N2. This summation is equal to Iref if we neglect the charge leakage and charge feed-through during the switching. Both factors are reduced by the modified current cell as explained below.

When P3 is switched from ON to OFF, its channel charge will be released to its drain and source. The charge released to the gate of P1 will cause error in its gate voltage and drain current. When the switching is very fast, the charge is equally partitioned into drain and source side as described in [58]. A half-sized dummy transistor P2 controlled by Cal is introduced to cancel the charge released from P3 to the gate node of P1. Although there are many charge compensation methods, the half-sized dummy transistor is simple to implement.

P5 is introduced to reduce the clock feed-through from Sup to the drain node of P4.

Without P5, the switching of clock signal Sup will cause fluctuations of drain node of P4 by coupling through gate-to-drain capacitance of N2. When Sup goes from low to high, while N2 is forming a channel, P5 is OFF, so the signal path from the drain of P5 to the drain of P4 is open, the clock feed-through due to coupling is therefore avoided. When Sup goes from high to low, some coupling exists at the beginning, but since N2 is quickly cut off, the voltage at the drain of P5 goes up, turning it off, and isolating the drain node of P4 for the remaining transition of Sup.

In Figure 4.3, M2 connects to a redundant current cell for continuous conversion. Assume one cell is supplying current just before it is calibrated. When this current cell is switching to the calibration mode, Cal=1 and Sup=0, the current it supplied will be cut off. During this time, M2 is ON and a redundant cell will take over and supply the same amount of current. Care should be taken in selecting the switch sizes of M2 in order to reduce the glitches during the transition time.

In order to output currents to the resistor load, output switches are introduced as shown in Figure 4.3. Two complementary selecting signals Sel and Selb control the transistors M0 and M1 to switch the output currents to complementary output currents Iout and Ioutb. The output switches here are NMOS switches rather than PMOS. The trick is to employ charge injection to speed-up settling of output [57]. The comparison between PMOS and NMOS switching is made in Chapter 3. Figure 3.16 and Figure 3.17 show the different switching schemes. For comparison, simulation results are shown in Figure 3.18. For the PMOS case (left), the output current settles to 0.1% in more than 3.9ns, while for the NMOS case (right), it settles within 2.17ns. This shows the advantage of utilizing charge feedthrough and NMOS switches over PMOS switches for speeding up the settling.

In order to have high output impedance to tolerate drain node variations, P1 is cascoded with P4. Vb2 is set to a value such that P4 is in saturation even if there are some power-supply fluctuations. Vb2 is a bias voltage supplied by a series of a PMOS

and an NMOS that are diode connected shown in Figure 4.4. The nominal value of Vb2 is 0.9V, which corresponds to power supply value of 2.5V. +/-20% power supply variation causes Vb2 to change from 0.9V to 1.1V and 0.7V, respectively. In order to maintain P1 and P4 in saturation, equation 3.22 to 3.25 need to be satisfied. Proper sizing of transistor P1 and P4 is necessary.

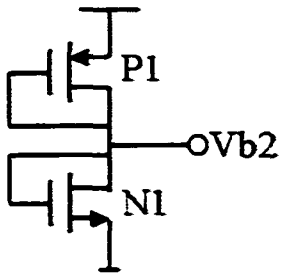


Figure 4.4 A simple voltage divider

For illustration purposes, a 1-bit DAC schematic with a redundant cell is shown in Figure 4.5. Simulation results of the 1-bit DAC are shown in Figure 4.6. As marked in the figure, the current output of the cell settles to 63.99uA. The relative error to the ideal 64uA is 0.01/64=0.015%, which is less than $\frac{1}{16}$th of an 8-bit LSB.

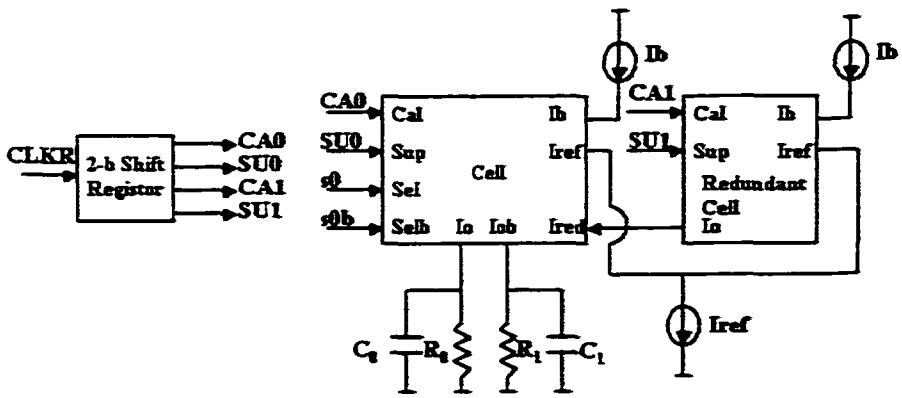


Figure 4.5 Schematic of 1-bit DAC with redundant cell

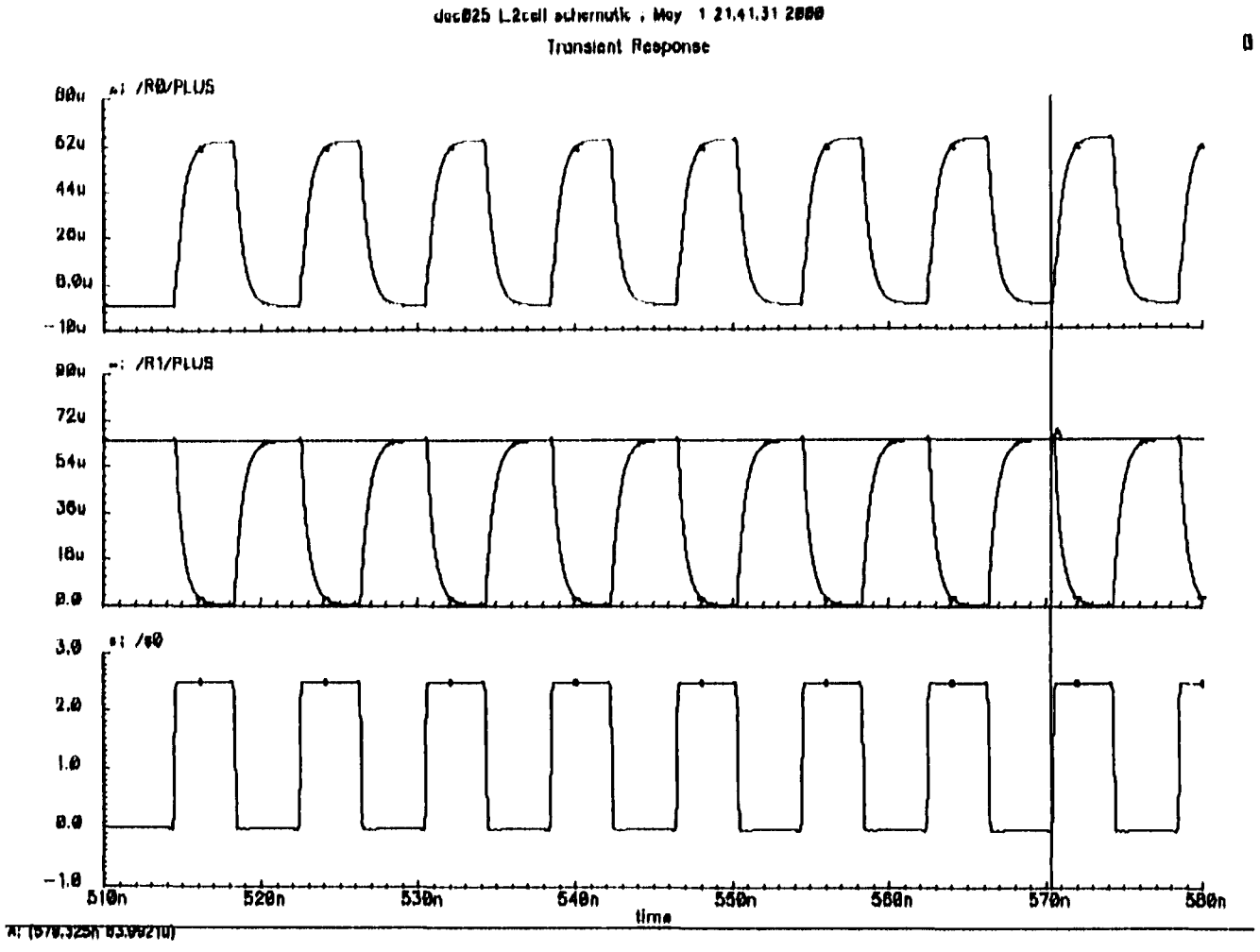


Figure 4.6 Simulation of self-calibrated current cell

Figure 4.7 shows 16 continuous self-calibrated cells for a 4-bit DAC. Ci and Si (i=1,2, ... 16) are complementary clock signals for controlling the current cells to be in calibrating or supplying modes. When Ci=1 and Si=0, Celli will be calibrated. At any one time, only one cell is being calibrated, the others are supplying. These signals are generated by shift registers explained in the next section. The redundant cell's output connects to all the other 15 current cells through 15 transistors (all are the same size as M2 in Figure 4.3).

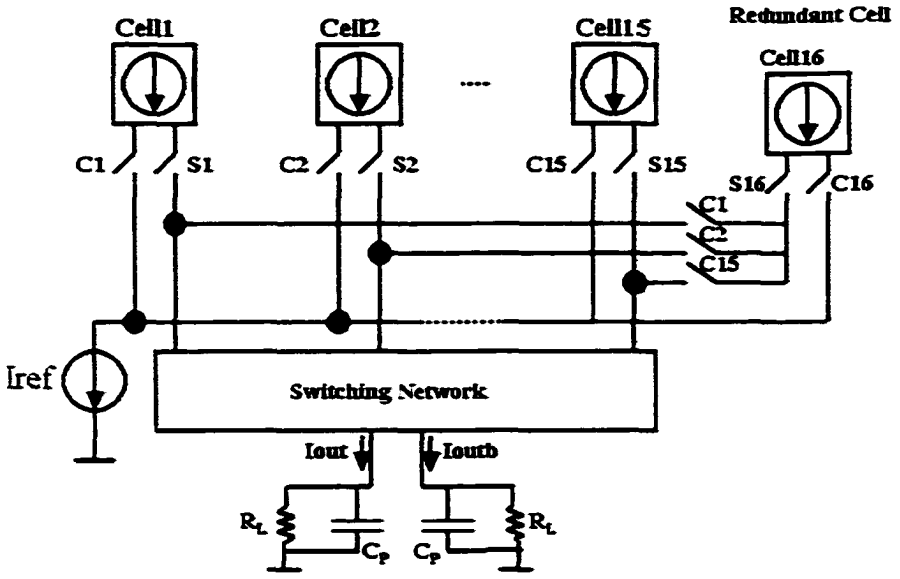


Figure 4.7 4-bit MSB block current cell array

### 4.3.2 Regulated Biasing Current Mirrors

In order to get all the source and sink currents for continuous current calibration, current mirrors are used. For good performance, we need each current mirror to have high output impedance and small minimum output voltage Vmin that maintains the mirror in normal operation. Generally speaking, cascode mirrors and Wilson mirrors

have high output impedance, but cascode mirrors need Vmin=2Vds(sat) and Wilson mirror need Vgs+Vds(sat). Compared with a simple mirror, the value of Vmin is larger while the output impedance is higher. So a better structure is needed that has a small Vmin requirement. The best candidate is regulated cascode mirror. It has very high output impedance and the same minimum Vmin=Vds(sat) requirement at the output node as a simple mirror. A simplified regulated cascode current mirror is shown in Figure 3.11. In Figure 4.8 some extensions for applying it to this implementation are shown. The regulated cascode current mirror consists of N3, N4 and N5. N1 and N2 form a simple mirror, N3 has the regulation function and makes the Vds4 about equal to Vds1. Given the size of N4 is equal to N1, then the drain current of N4 is equal to that of N1. From equation 3.9, the output impedance of this regulated cascode mirror is approximately given by $g_m^2 R_o^3$ and is much higher than both a Wilson mirror and a cascode mirror. In Figure 4.8, N1 to N5 and P1 to P2 form a mirror to generate a unit current equal to Iref. P3 to P8 and N6 to N10 generate the Ib=0.9Iref. By repeating the circuits, multiple biasing currents are obtained.

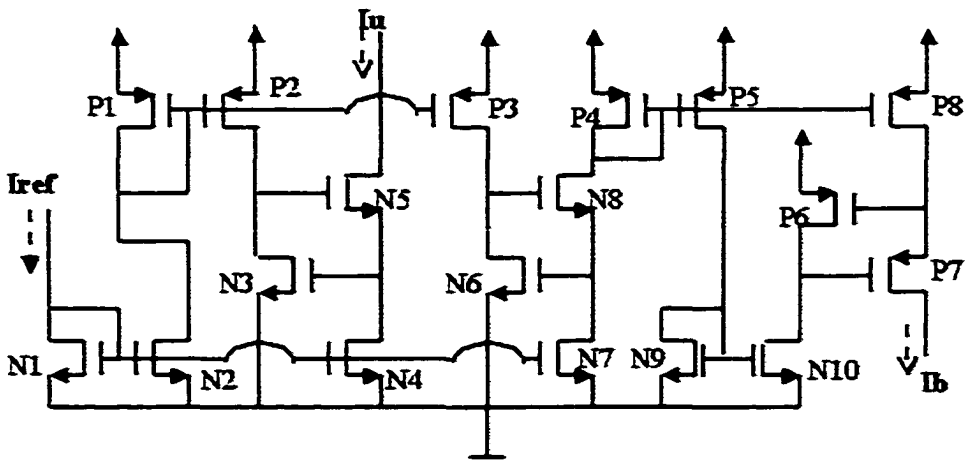


Figure 4.8 Regulated biasing current mirrors

This technique is also applied to current scaling needed for the LSB block. By repeating P2, N3, N4 and N5, another Iref is generated, which can be scaled down by a factor of 16. Figure 4.9 shows the unit currents obtained through this method. By repeating transistors P6-P8, we can get multiple unit currents for the 4-bit LSB block. The issue here is the matching, we need to match transistors N1, N4 and N7, and P4 and P8. Due to the modular design, the matching accuracy here needed is relaxed.

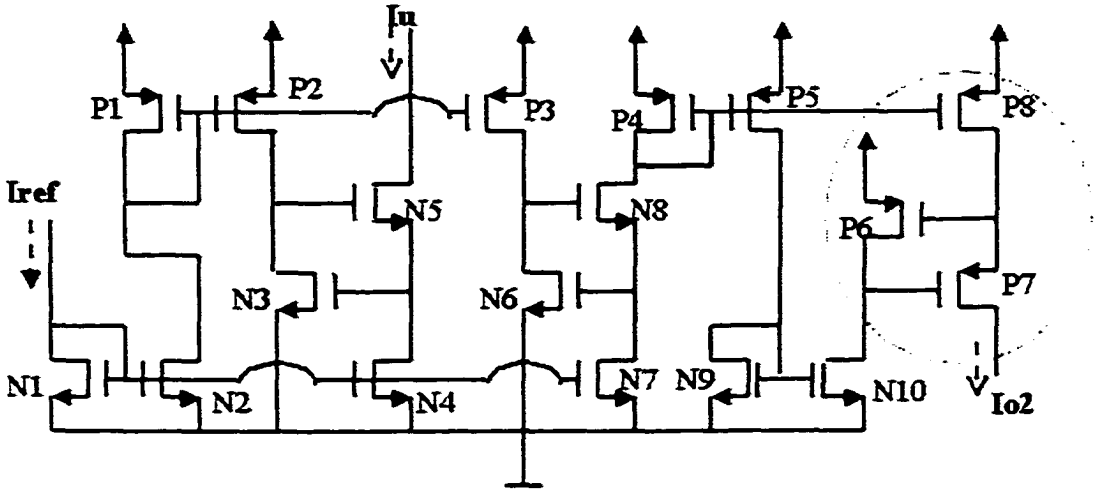


Figure 4.9 Current cell realized by regulated current mirror

### 4.3.3 4-Bit MSB

A 4-Bit MSB DAC block schematic is shown in Figure 4.10. It consists of a 16-bit shift register loop, a binary-to-thermometer decoder, a 15-bit register and 16 current cells (including the redundant cell). The reference current Iref=64uA and 16 current sources of Ib=0.9Iref are supplied from the bias current block. The transistors of the biasing current block need to be laid out as close as possible to achieve good matching.

Figure 4.11 shows the timing sequence of the continuous self-calibration. The RESET signal will set all the DFFs to initial states. CLKR is the self-calibration clock. When

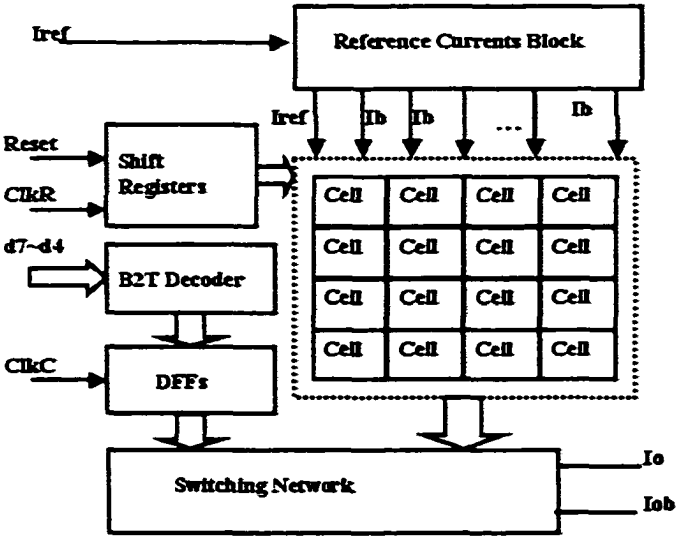


Figure 4.10 4-bit MSB block

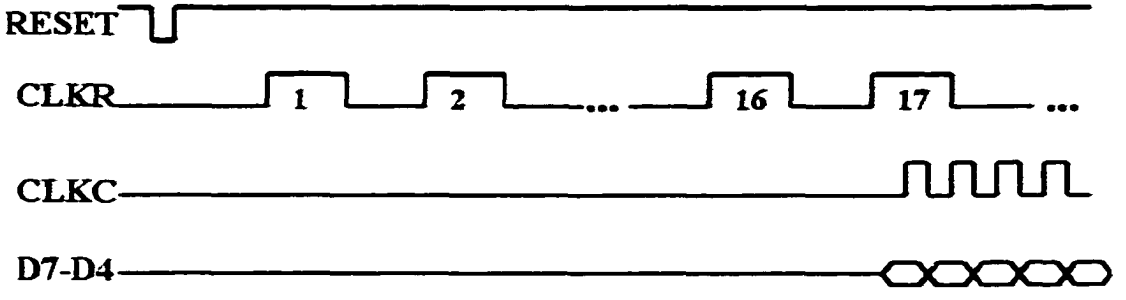


Figure 4.11 Timing sequence for the 4-bit MSB block

CLKR's rising edge comes, one of the 16 current cells will be calibrated, after all 16 cells are calibrated once, the conversion clock CLKC and data D7-D4 are fed in synchronously. The calibration cycles continue and current cells are calibrated continuously.

Normally when one cell is taken over by a redundant cell, a glitch will appear in the output. Three current cells are involved in switching, one cell that is going to be calibrated ( I1), one cell that is going to supply ( I2), and the redundant cell (Ired). So I1 will stop supplying, I2 will start supplying, and Ired will stop supplying for I2 and start supplying for I1. These switching actions will inevitably introduce the current glitches even if the sizes of the switches are optimized.

In order to reduce the glitch effect to the output spectrum, a random number generator can be used to select the cells. A simple way is to randomly assign the decoder output order and optimize the size of the transistor M2 and N2 in Figure 4.3. By increasing the size of transistor M2, the switching speed of the redundant cell can be matched to the cell that it will replace, and the glitch will be reduced. Figure 4.12 shows one simulation result for the 4-bit MSB DAC. The transfer curve shown here is very close to the ideal transfer curve. From the measured points, we can see that the full range current is about 959.9uA and the absolute error is 0.1uA. If the fabricated DAC can achieve this accuracy, we can get more than 12-bit accuracy.

### 4.3.4 4-Bit LSB

There are two methods to construct the 4-bit LSB. The first way is to use the same 4-bit MSB DAC and divide its output current by 16. This is shown in Figure 4.13. The divider implemented by a simple mirror is shown in Figure 4.14. This mirror will suffer the modulation effect caused by different Vds of the mirrored transistors. The self-calibration control logic hardware is also needed.

The second way is to divide the current at the front end. By switching 1/16th of the unit current of MSB block, the same result is obtained. By incorporating the unit

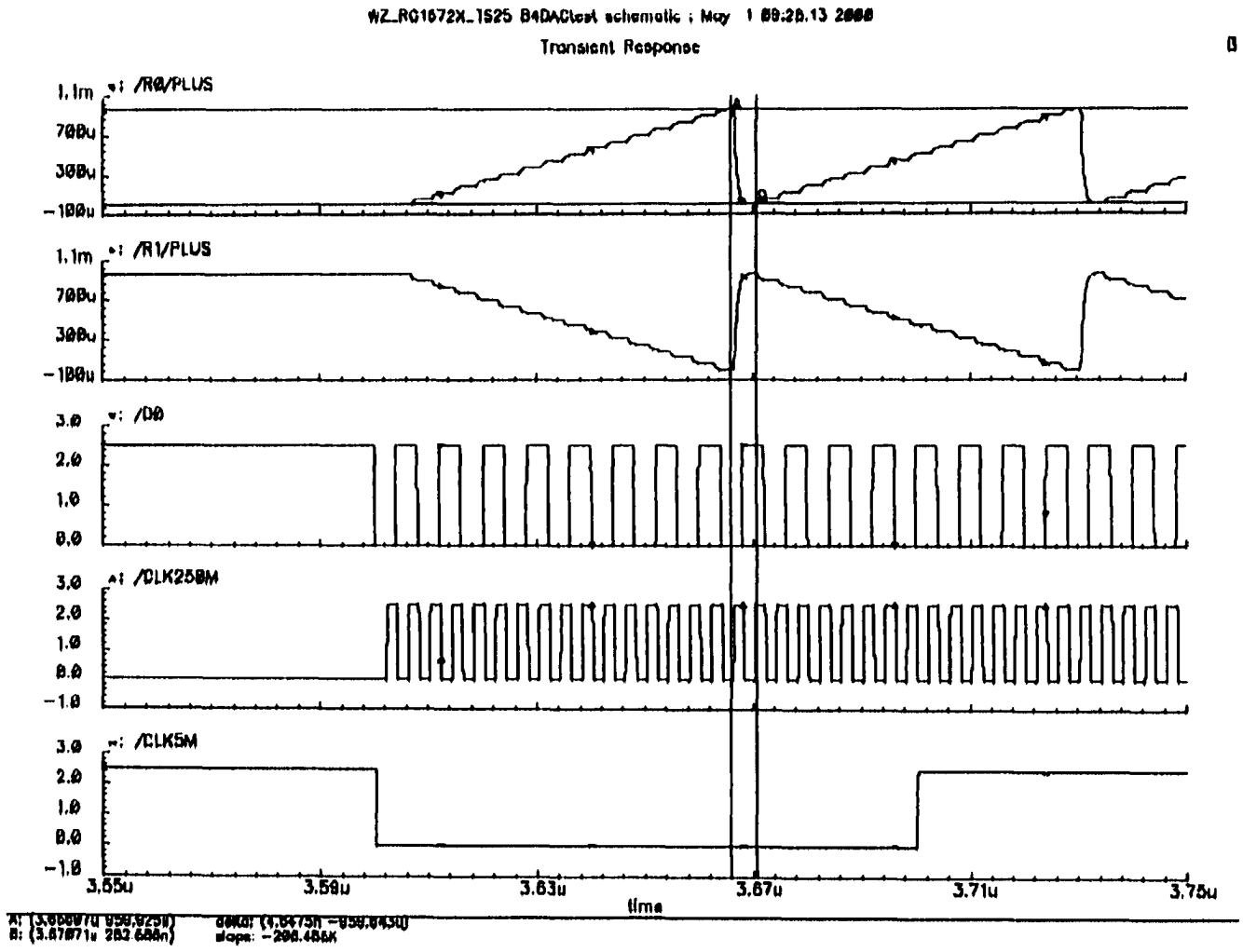


Figure 4.12 Simulation result of the 4-bit MSB block

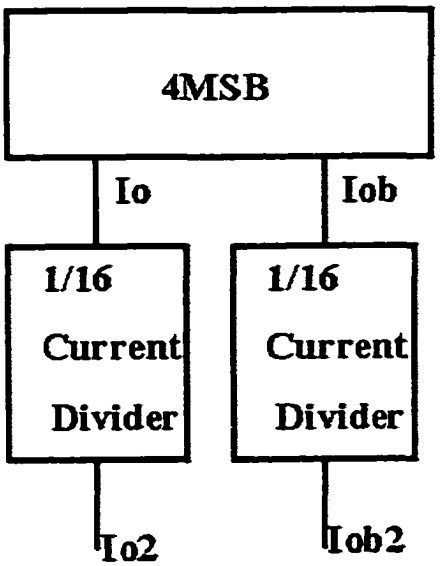


Figure 4.13 Realization for 4-bit LSB by current divider

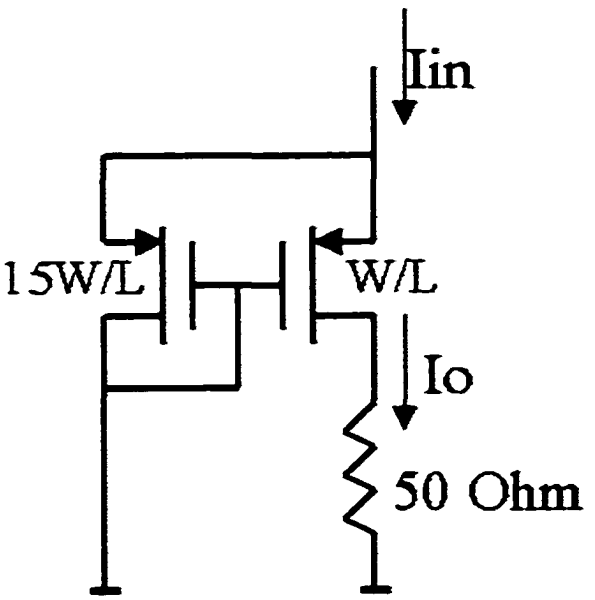


Figure 4.14 1/16 current divider

current sources into the biasing block for both 4-Bit MSB and LSB, the ratio between the two blocks can be maintained by local matching. Doing it this way, the self-calibration control logic circuits are not necessary and the hardware for the flip-flops is reduced. This method is adopted in the prototype. The block diagram for the 4-bit LSB is shown in Figure 4.15.

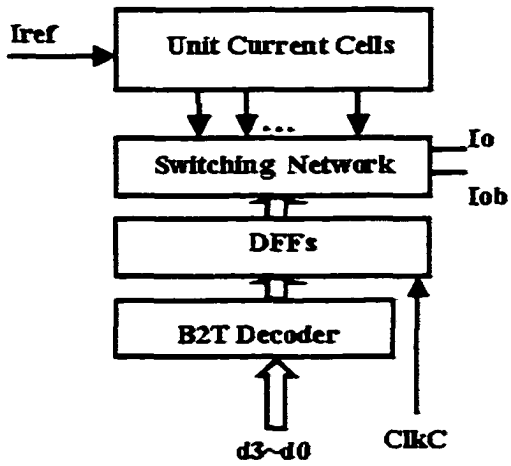


Figure 4.15 4-bit LSB

### 4.3.5 8-Bit DAC

The 8-bit DAC diagram is shown in Figure 4.16. When two 4-bit blocks work together, one problem is that during the transitions from xxx01111 to xxx10000 there is a time difference between the switching of the MSB block and the LSB block. This timing difference will generate a glitch. Although the output settles to its designed value, this glitch will introduce some noise and lower the signal to noise ratio. A solution to reduce the glitch error is the addition of a so-called deglitcher circuit at the output of a DAC. Such a deglitcher circuit with a distortion level below the LSB level is difficult to design [20]. In practice, As long as the glitch energy is less than 1/2 LSB energy, it is acceptable. By optimizing the latches and adjusting the timing of the selecting switches

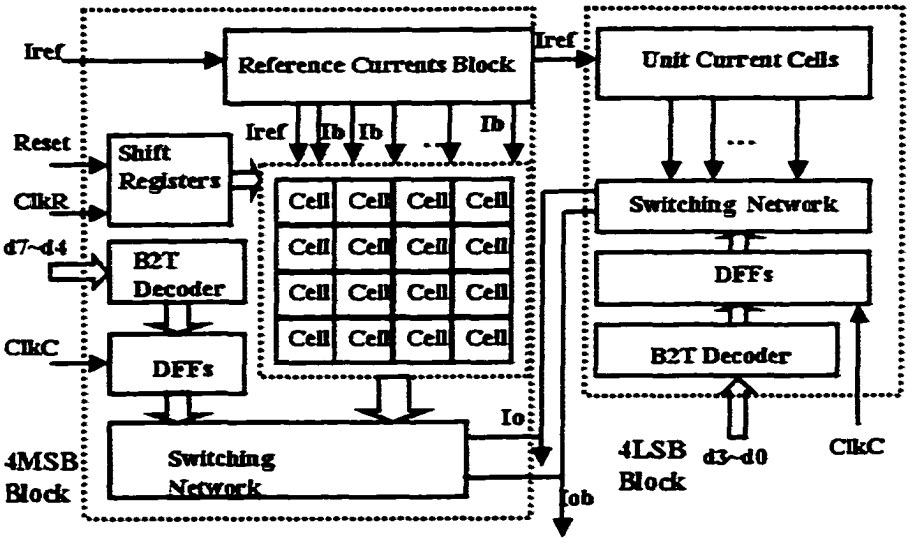


Figure 4.16 8-bit DAC prototype diagram

to the MSB and LSB current sources the glitch can be greatly reduced. This can be seen from the top level simulation results in Figure 4.17.

## 4.4 Top Level Simulation

The simulation is done for different process corners and temperature corners. The circuit works fine in different environmental temperatures (-40, 25, 100 °C) with Slow-Slow, Typical-Typical and Fast-Fast models. Slow-Slow models at 100 °C is the worst case. Figure 4.17 shows a simulation result of 8-bit conversion with an update rate of 250MS/s. It is clear that the maximum error of the DAC is within 0.5uA, which is less than $2^{-11}$ of the full scale, which results in an absolute accuracy of more than 10-bits. The power dissipation is about 8mW.

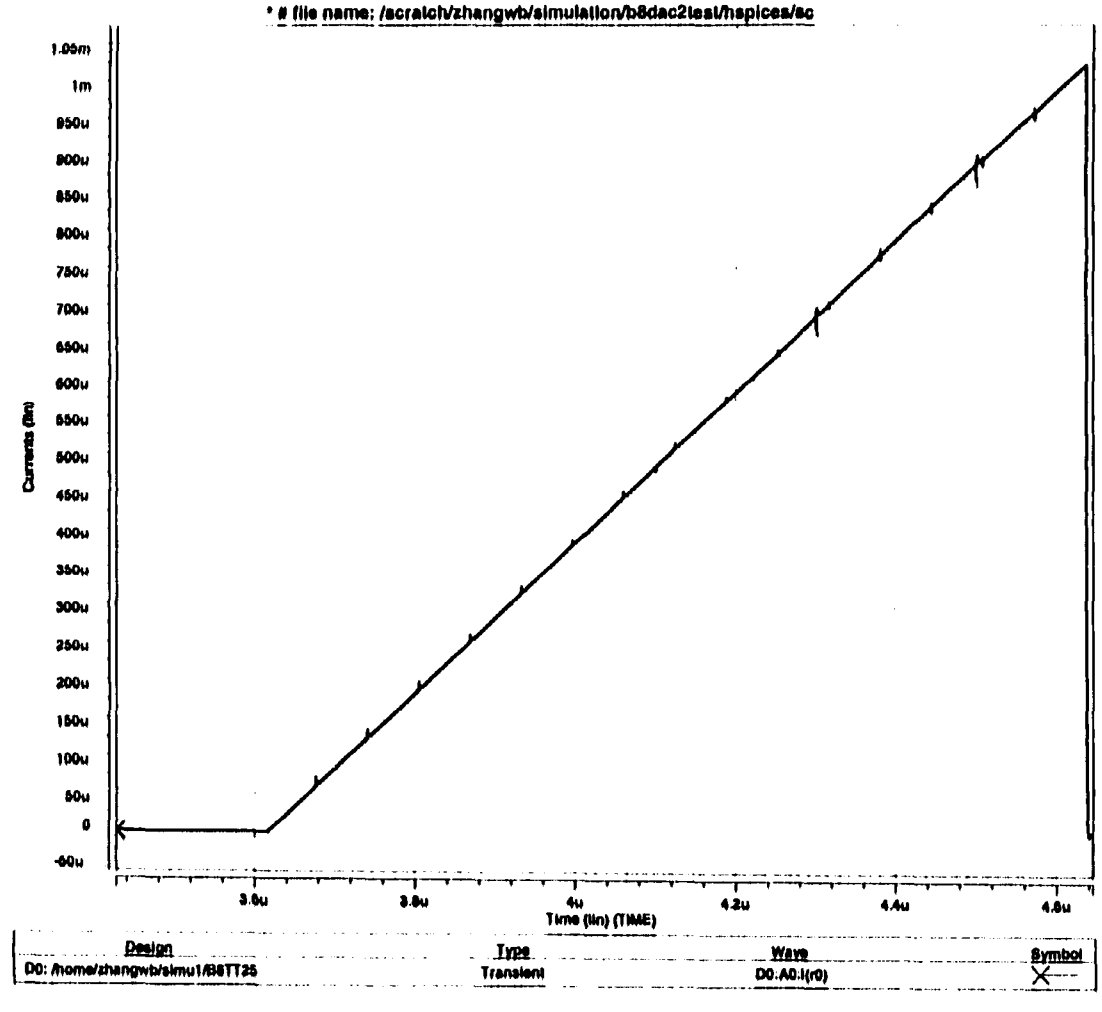


Figure 4.17 A simulation result plot of transfer function

## 4.5 Layout

The layout is shown in Figure 4.18. Symmetry is applied to match the transistors especially those in the bias-current sources. In the bias current block layout, interdigitized and commoncentroid layout techniques are applied to the critical transistors that require matching. Digital circuits and analog circuits are separated by a guard-ring which prevents the digital noise propagating to the analog cells. Analog power and digital power lines are also separated for this purpose. Routing from different flip-flops to the switching network is carefully done in order to achieve approximately equal wiring lengths for all the digital buses. Current cells are arranged randomly to reduce further spatial errors. The layout is implemented in TSMC's 0.25$\mu$ single poly five metal logic CMOS process. The total die area for the DAC is 0.44mm*0.3mm, which is 8% of that of a recently reported 8-bit current steering DAC [55]. This small area shows the advantages of continuous self-calibration method versus conventional methods that rely on matching.

## 4.6 Testing

### 4.6.1 Testing Setup

Testing of the high-speed DAC is as difficult as designing it. Many factors need to be considered. Among them, clean and stable clock signals, clean and stable VDDs and VSSs power planes, accurate and stable voltage and current references are critical. Impedance matching between instrument sources and the loads on the PCB is also very important, because the unwanted reflections due to the bad matching will create unwanted harmonics and noise. In high frequency testing, parasitic inductances of the PCB wires play a big role. Several nH inductance could easily reduce the settling of the current output and incomplete settling causes harmonics. Further research in this topic

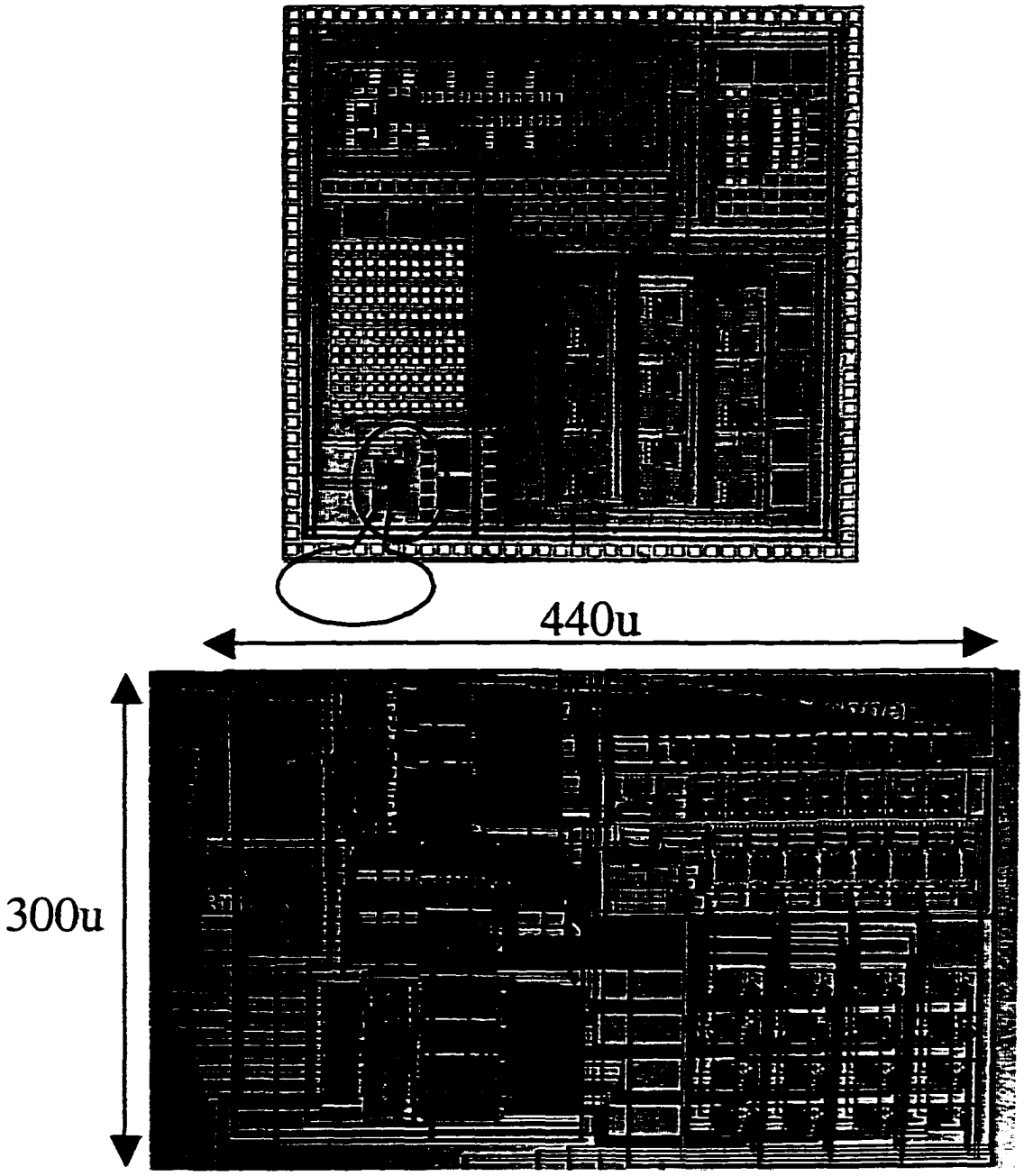


Figure 4.18 Die photograph of the chip and the 8-bit DAC

is interesting but beyond our scope. Care should be taken to reduce the inductance effects in both the chip design and PCB design.

The testing-setup used to measure DC performance as well as dynamic performance is shown in Figure 4.19. We used an HP 8133A pulse generator to generate a low jitter reference clock. The pulse generator has complementary outputs CLK and CLKN. CLK is used as a reference clock for the Sony/Tecktronix DG2020A data pattern generator to provide digital sine wave ($D7 \sim D0$) and the self-calibration clock (f=Fr). CLKN is used as a low jitter sampling clock source, it is filtered by a low pass filter to get a clean sine wave source (f=Fc) which is used as the conversion clock of the DAC. Since the DAC has input buffers following the digital input pads, the sine wave is converted to a pulse clock internally. By doing so, a lot of coupling noise triggered by high frequency transitions are avoided. The outputs are observed by a spectrum analyzer and an oscilloscope.

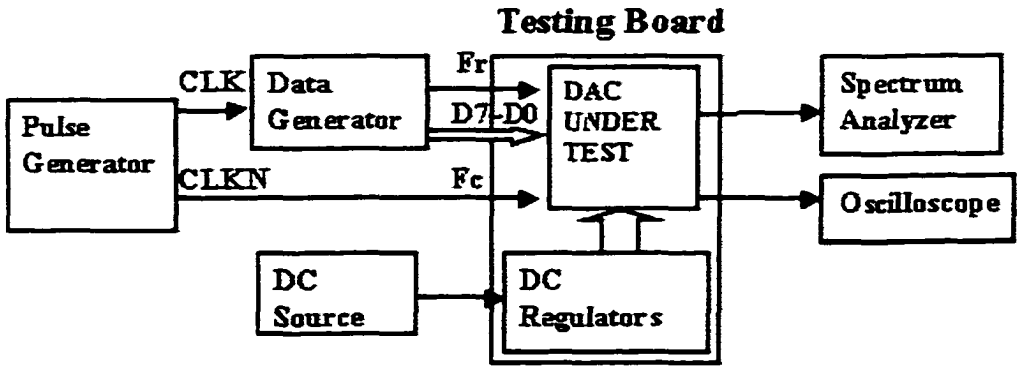


Figure 4.19 Testing setup for the DAC

Since the maximum speed of the data generator is 200MBPS, the maximum frequency of the input digital sine wave is less than 100MHz. In the Nyquist testing, a 99MHz input digital sine wave is tested. For sampling rate of 250MS/s or more, an asynchronous method is used. This is done by letting the data pattern generator use its internal

clock reference, and the pulse generator supplies a 250MHz clock. Asynchronous testing may introduce some noise since invalid data during switching will be converted. A better result would be obtained if an on-chip digital synthesizer is used, since an on-chip synthesizer can easily go to 250MHz, less parasitics are present and the signal buses can be managed to have equal delay. This will reduce the unwanted reflections due to imperfect impedance matching between instrument source and the load on the PCB board.

In the prototype testing, a two layer PCB board was used. The ground plane and shielding were not designed optimally. A lot of interference sources exist, so many manual fixes were applied to reduce the noise and interferences from the board itself. Better results could have been obtained if a multilayer PCB were used and the interference signals separated. If the coupling signals to the analog power supply can be reduced, the results would be better as well. The PCB is shown in Figure 4.20.

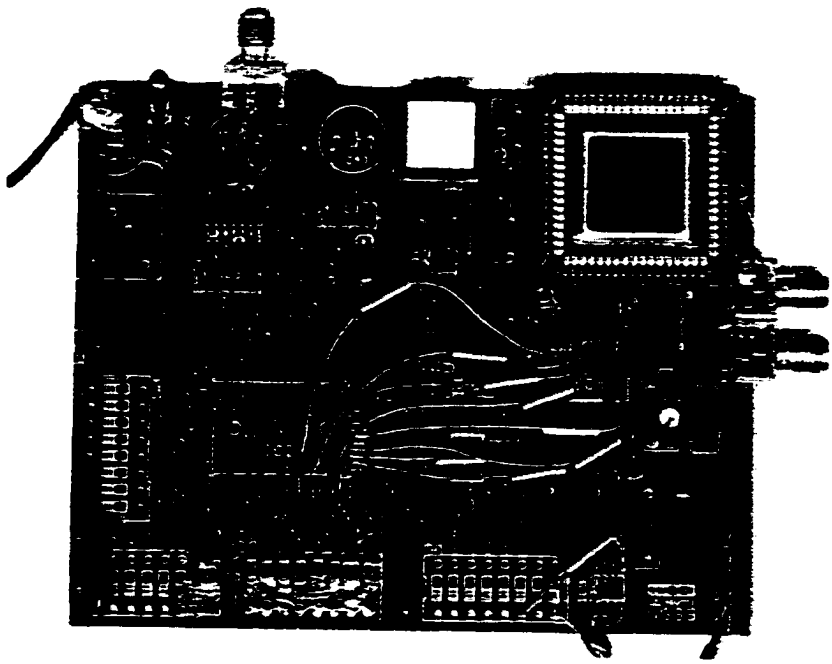

Figure 4.20 PCB for the 8-bit DAC

Several techniques are applied to the testing board to reduce parasitic effects and coupling effects from unwanted signal. Power regulators have been used to reduce the fluctuation of the DC source supply to the chip, and de-coupling capacitors are distributed on the board to remove the unwanted high frequency signals coupled in the power line and signal lines. Some ferrite bead cores are used to absorb the high frequency noise in the digital input bus lines and power lines. A current reference circuit is used to supply a dc current source that has low tempco and immune to voltage source fluctuation.

### 4.6.2 DC Results

The DNL and INL plots are obtained after the transfer curve (Figure 4.21) is obtained. The transfer curve is obtained when a digital ramp is applied with the conversion clock of 1MHz. Eight chips were tested, and all of them were functional. Figure 4.22 shows the INL and DNL. It can be seen that the maximum INL is less than 0.6 LSB. The die photograph is shown in Figure 4.18.

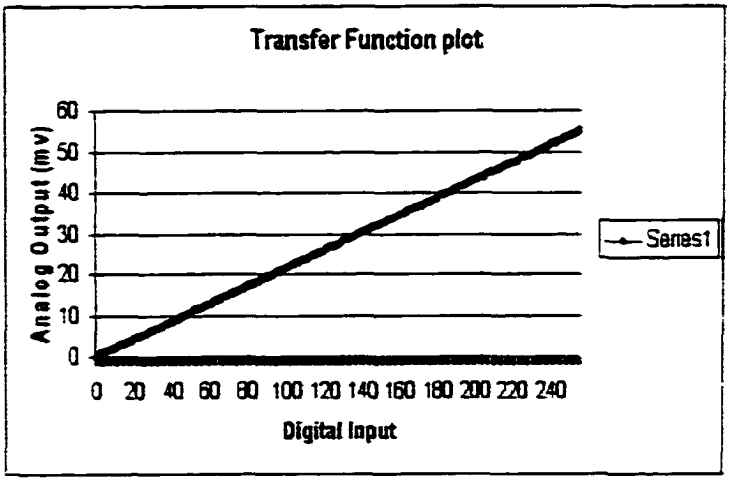


Figure 4.21 Tested transfer curve

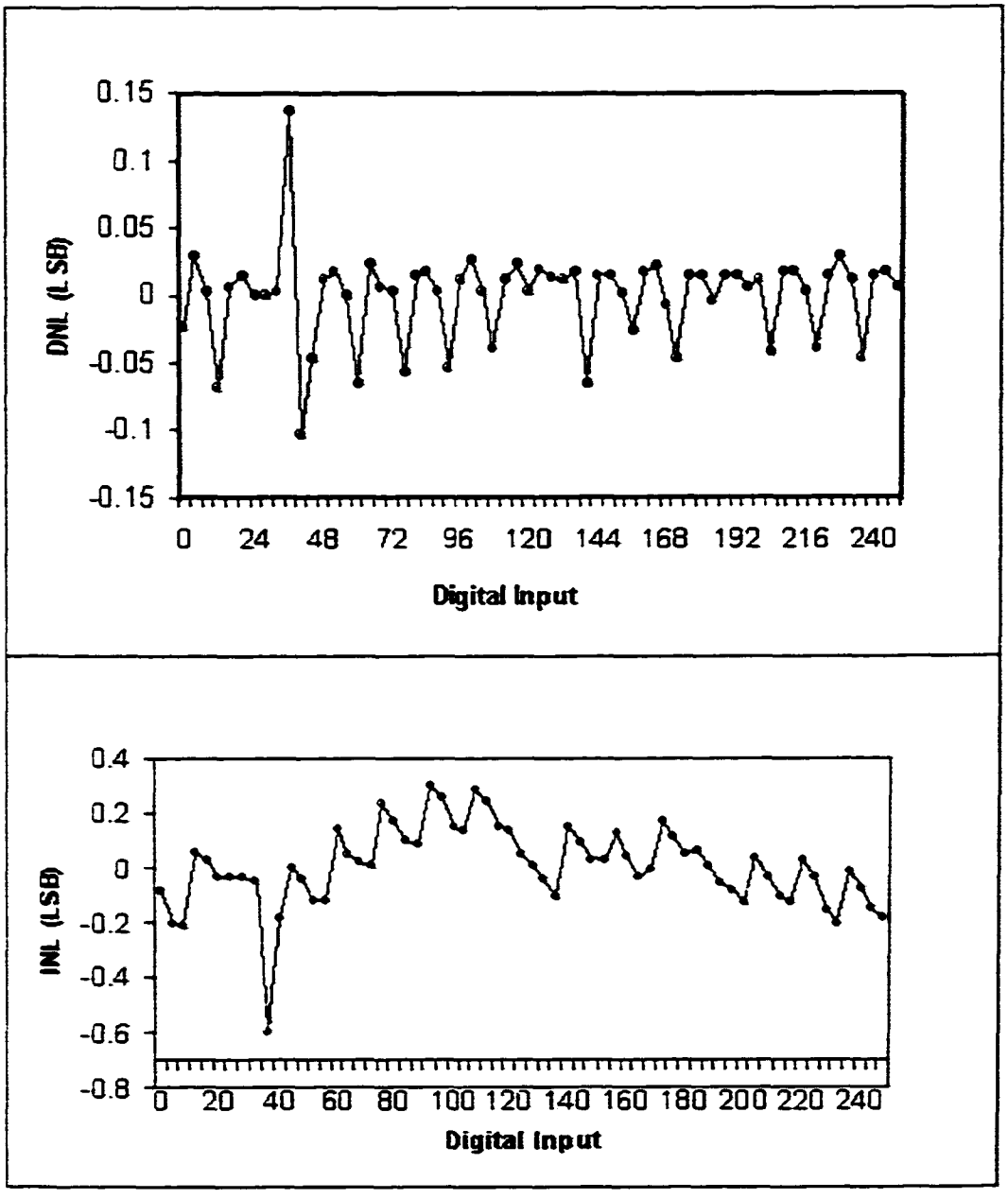


Figure 4.22 DNL and INL plots

### 4.6.3 Dynamic Performance Results

Output spectrums under different input conditions were obtained. Figure 4.23 shows the output spectrum when the input digital sine wave is 15.625KHz (8MHz/512) and the sampling rate is 250MS/s. Figure 4.24 shows the transient waveform of the output voltage corresponding to a digital sine wave input.

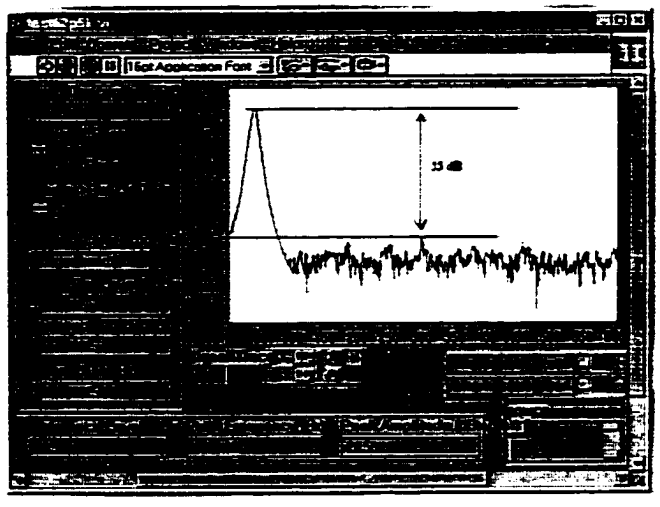

Figure 4.23 Output spectrum of a digital sine wave

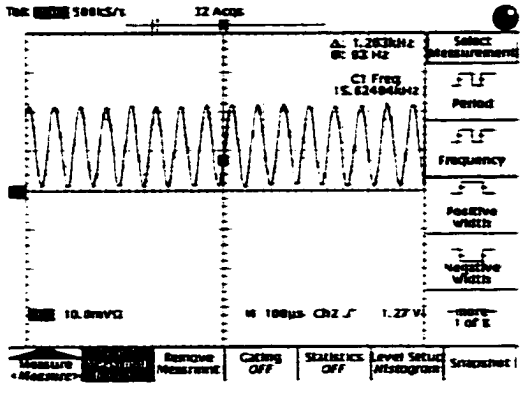

Figure 4.24 Output waveform of a digital sine wave

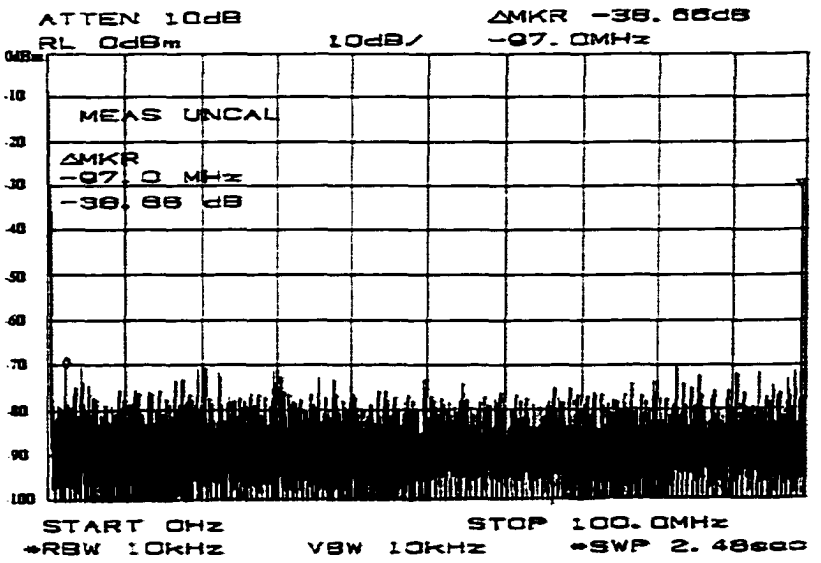


Figure 4.25 SFDR for high frequency input digital sine wave

Nyquist rate testing was done when the input frequency is 99MHz, and the sampling frequency is 200MHz ( the maximum speed of the pattern generator). Figure 4.25 shows a typical testing result with SFDR equal 39dB. Lots of harmonic signals present. This is due to bigger glitches caused by bigger step sizes and the different time delay between digital input lines. The 99MHz digital sine wave for this test consists of a series of digital samples by 200MS/s sampling rate. There are lots of step changes from the positive peak to the negative peak and vise versa. Big glitches happened due to these step changes lead to big harmonic signals. Reflections due to the imperfect impedance matching between the spectrum analyzer and the DAC output node also cause harmonic distortion.

The SFDR could be worse if the input codes change faster with bigger step sizes and the coupling signals are bigger. The coupling signals can deteriorate the input signal, calibration clock, sampling clock, power line and ground. SFDR (Spurious Free Dynamic Range) for several synchronous tests are summarized in Figure 4.26. Notice that with

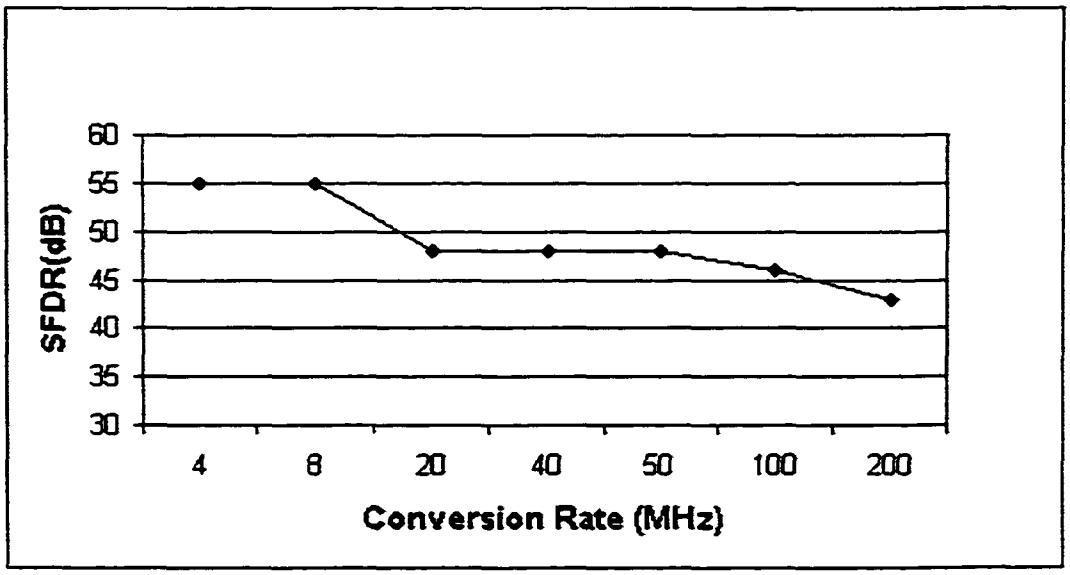


Figure 4.26 SFDR for several synchronous conversion

the increase of the conversion rate, the digital sine wave frequency is also increasing, and the SFDR is decreasing. The measured DAC characteristics are summarized in Table 4.1.

It is interesting to note that the DAC can work even when the conversion clock increases to around 1GHz. This shows that the digital circuits can work at much higher frequencies than 250MHz. In embedded applications, if we change the output load to have smaller RC value, a faster DAC is obtained. A simulation of 500MS/s with an output load of 50 Ohm in parallel with 1pF capacitor is shown in Figure 4.27.

## 4.7 Summary

In this chapter, we have demonstrated the extension of continuous self-calibration from audio frequency to 250MS/s. A new self-calibrated current cell was proposed and implemented in the 8-bit 250MS/s DAC prototype in TSMC's 0.25$\mu$ single poly

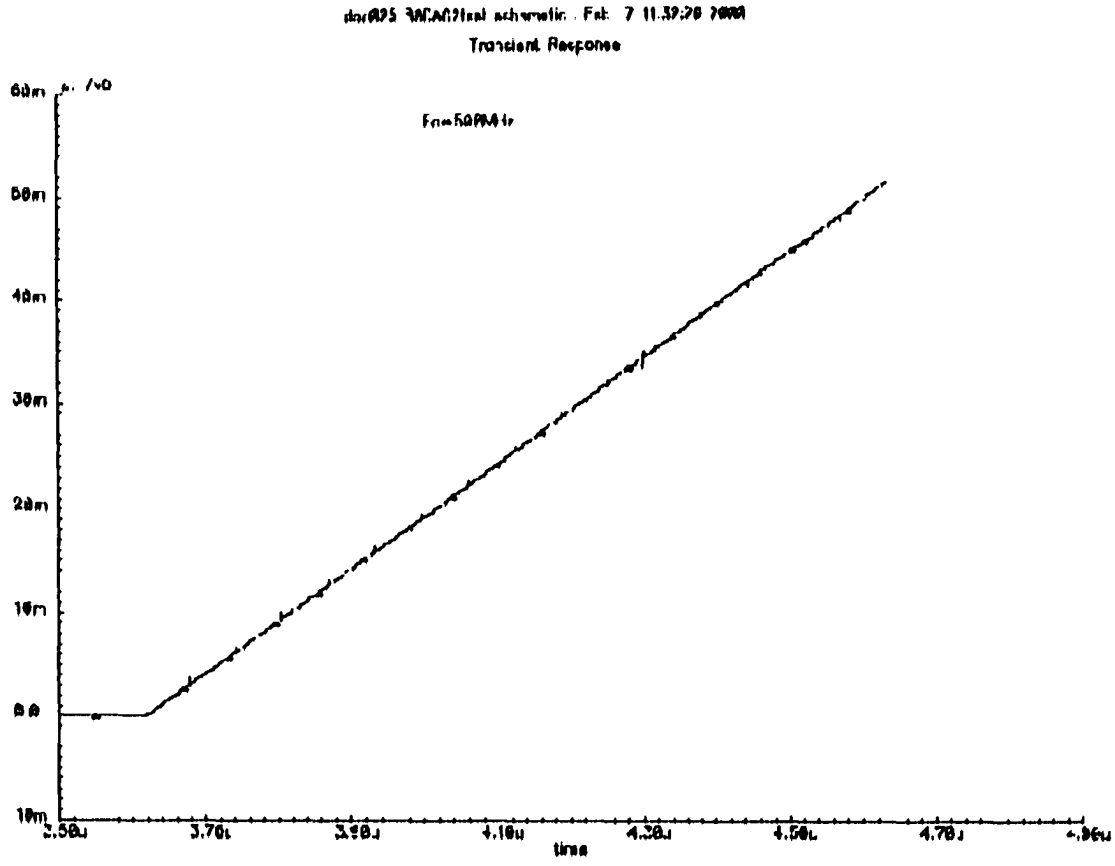


Figure 4.27 A simulated transfer function when conversion rate is 500MS/s with output load of 50 Ohm in parallel with 1pF

Table 4.1 Measured 8-bit DAC characteristics

| | |
|---|---|
| Technology | 0.25$\mu$ CMOS |
| INL | < 0.6 LSB |
| DNL | < 0.15 LSB |
| Update Rate | 250MS/s |
| Full Scale Output Range (RL=50 Ohm) | 1mA |
| SFDR (Fin=16.6KHz) | 55dB |
| SFDR (Fin=99MHz, Fs=200MHz) | 39dB |
| Power Consumption | 8mW |
| Supply (Analog/Digital) | 2.5V/2V |
| Die Area | 0.13mm$^2$ |

five mental logic CMOS process. Charge injection and clock feedthrough reduction are featured in the new cell. This DAC with its 8mW power consumption and 0.13mm$^2$ die area is very potentially suitable to high-resolution and low-power applications.

From the testing, we can see that the prototype 8-bit 250MS/s DAC works, but harmonic effects are large. Although this is primarily due to a sub-optimal test board, the glitches during transition of calibration mode to supplying mode are also responsible. A way to reduce the glitches further is necessary for achieving high accuracy and high speed.

# CHAPTER 5. A REDUNDANT-CELL-RELAY METHOD FOR CONTINUOUS SELF-CALIBRATED CURRENT-STEERING DACS

In order to get high SFDR in high-speed, the glitches during the input code transitions should be as small as possible. As described in Chapter 4, the glitches during transition of calibration mode to supplying mode degrade the SFDR. A method to reduce the glitches should be very useful if developed. This chapter proposes a redundant-cell-relay method for continuous self-calibration of current-steering DACs. Compared with the prior-art continuous self-calibration scheme [6] extended in the 8-bit prototype described in Chapter 4, this scheme has the advantages of lower glitch energy, lower distortion and higher accuracy. The method facilitates high-speed and high-resolution current steering DAC design. A patent application has been submitted to US Patent Office [7]. A prototype 10-bit DAC implementing this method will be described in Chapter 6.

## 5.1 Continuous Self-Calibration and Transition Glitch

As discussed in Chapter 2, although there are many calibration methods in DAC design, such as laser trimming and self-compensation techniques, continuous self-calibration has several interesting advantages such as adaptability to temperature variations, insensitivity to process variations and of course continuous conversion after the first calibration cycle. Matching is not a major concern due to the dynamic element matching

principle [6]. While glitches occur during the switching from calibration into output mode of the calibrated cells, the glitches can be ignored if the energy is less than $\frac{1}{2}$ LSB. This is the case in audio frequency applications where settling time is long enough as in [6]. However, in high speed applications, the settling time for the converted analog signal is short($\frac{1}{F_s}$). These glitches not only can be higher than $\frac{1}{2}$LSB energy, but also cause unsettled results and thus distortion. Thus periodic errors with a frequency of the calibration clock exist. This will increase the noise floor and deteriorate the SFDR of the DAC. To minimize the glitches, a deglitcher can be implemented before the total output current is applied to the current to voltage converting operational amplifier [20]. However the operational amplifier is a new bottleneck for very high speed applications. Another choice is to randomize the current cells in order to distribute the glitches into the two complementary outputs, by arranging this way, the glitch effect is reduced by half. In the 8-bit prototype described in Chapter 4, spacial randomization is used.

The glitches are caused by the fact that the redundant cell is fixed as shown in Figure 5.1. Glitches occur during the switching from calibration-mode to supplying-mode of the calibrated cells. For example, at time t1, Cell 1 is being calibrated (connects to Iref), Cell 16 (the fixed redundant cell) supplies current for Cell 1. At time t2, when Cell 1 is changing from calibration-mode to supplying-mode and Cell 2 is changing from supplying-mode to calibration-mode at the same time, Cell 16 will take over for Cell 2. Cell 16 has to be disconnected from the operation for Cell 1 and then supply for Cell 2. So during the switching, three current cells and six switches are involved, a transient glitch appears on the output current. If the input updating rate high and the time for settling is short, then complete settling of the output cannot be achieved. This is shown by the simulation in Figure 5.2. If the redundant cell can be dynamically assigned and "warmed-up" before it is supplying current to the output, the take-over action would be more smooth and the glitch would be much less if not removed. The working mechanism is similar to the relay in sports, so the method is called "the redundant-cell-

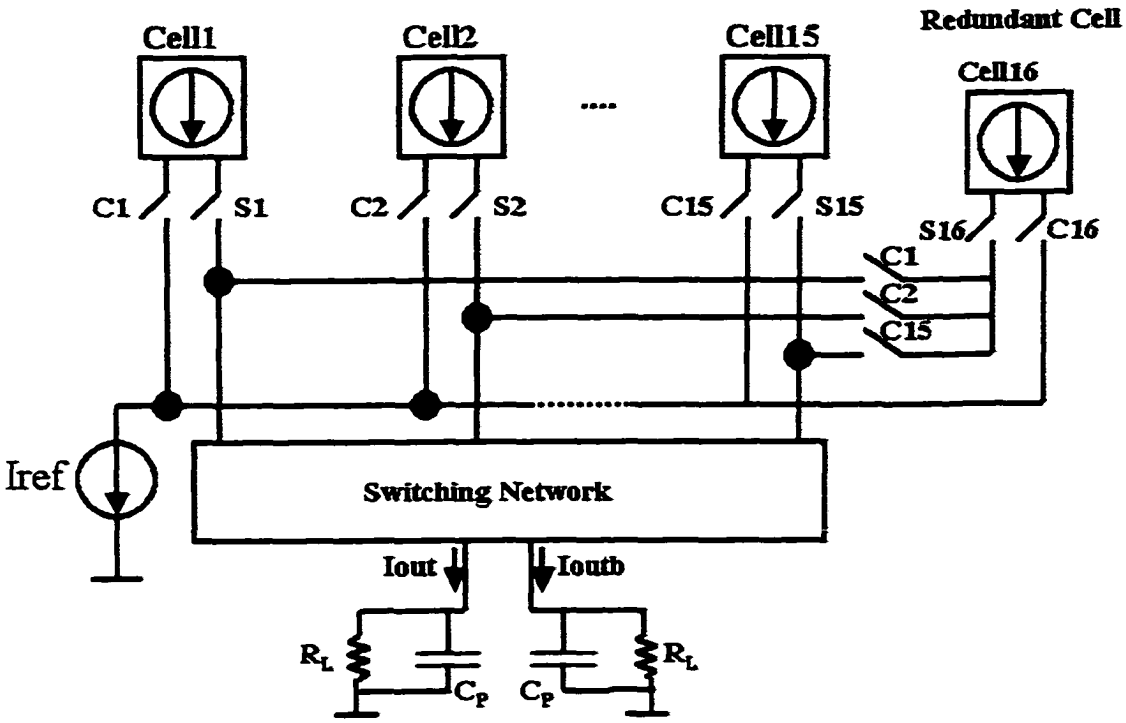


Figure 5.1 Fixed redundant cell in prior-art continuous self-calibration

relay" continuous self-calibration. The details are in next section.

## 5.2 Redundant-Cell-Relay Continuous Self-Calibration

In order to obtain smooth switching from the calibration mode to the supplying mode, the current cells should have the capability to warm-up before entering the supplying mode. Figure 5.3 shows the schematic of a self-calibrated current cell.

The current cell can be in one of the following working modes: calibration, supplying or dumping. The switches are used to control its working modes. In Figure 5.3, the switches Si and SiN (i = 1,2,3) are controlled by a pair of complementary signals. When switches marked S1 are ON, and those marked S1N are OFF, the cell is in calibration mode, the reference current Iref and current source Ib are applied to the current cell.

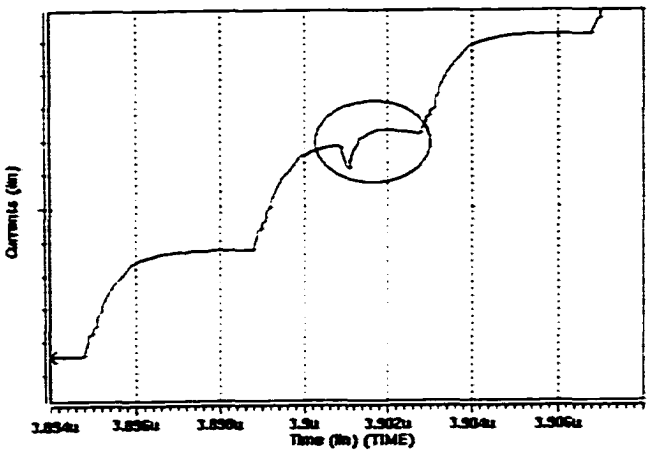

Figure 5.2 A glitch occurs during transition time

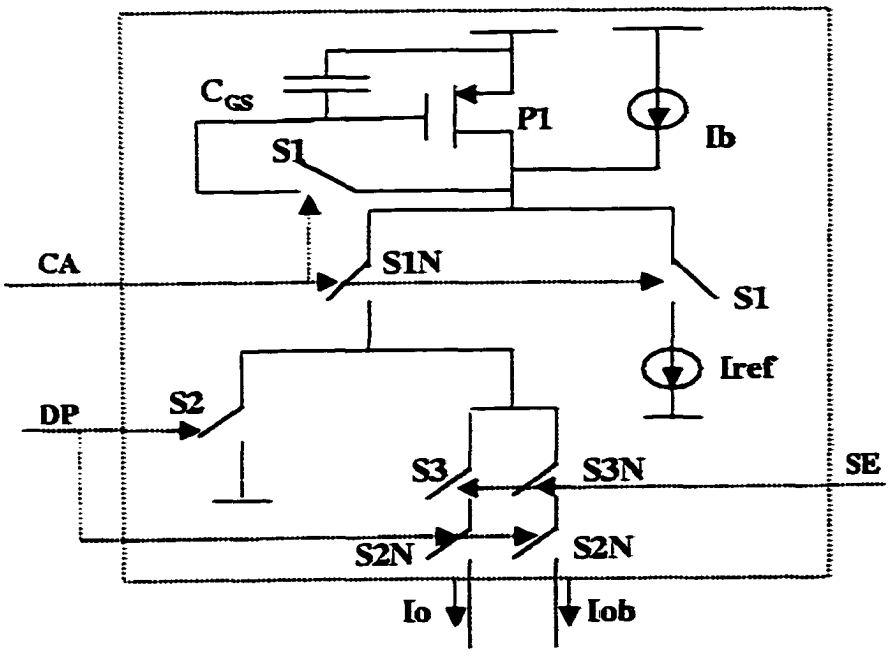


Figure 5.3 A self-calibrated current cell with relay control

the current of Iref-Ib flows through the transistor P1 and a charge is stored in capacitor Cgs. When S1s are OFF and S1N is ON, the cell operates in either dumping mode or supplying mode depending on S2. If S2 is ON, and S2Ns are OFF, the current cell works in dumping mode, and the current is dumped to ground. If S2 is OFF, and S2Ns are ON, the cell current will be steered to output Io or complementary output Iob. This is controlled by the selecting switches S3 and S3N. If the dumping mode happens after calibration mode and before supplying mode, it works as if warming up. If the dumping happens after the supplying and before the calibration, we can call it resting. The controlling signals to the switches are listed in Table 5.1. In order to realize a smooth continuous self-calibration, the relay style timing diagram should be designed. For simplicity, the timing diagram for two consecutive current cells in continuous self-calibration is shown in Figure 5.4.

Table 5.1 Control signals for the self-calibrated current cells

| Switches | Control Signals | Notes |
|---|---|---|
| S1, S1N | Calibration Control | If CA=1, S1 ON, S1N OFF, calibrating. |
| S2, S2N | Dump Control | If CA=0, DP=1, S2 ON, S2N OFF, dumping. |
| S3, S3N | Output Selection | When CA=0, DP=0, Output mode:<br>If SE=1, S3 ON and S3N OFF, outputs to Io.<br>If SE=0, S3 OFF and S3N ON, outputs to Iob. |

In Figure 5.4, DPi and DPi+1 are dumping control signals for cell i and i+1 respectively. CAi and CAi+1 are calibration control signals for cell i and i+1 respectively. After cell i has some rest, is calibrated and warmed-up, it will supply current. Cell i+1 starts its rest when cell i starts to output, then repeats the states of cell i. For an n-bit DAC, $N + 1 = 2^n$ cells need to work in the relay style, one of which is a dynamically redundant cell. The cell i is in redundant mode during the time t3-t0, the cell does not supply its current into Io nor Iob outputs. In the remaining time, cell i supplys its current to Io or Iob, thus is in supplying mode. The supplying mode is N times longer

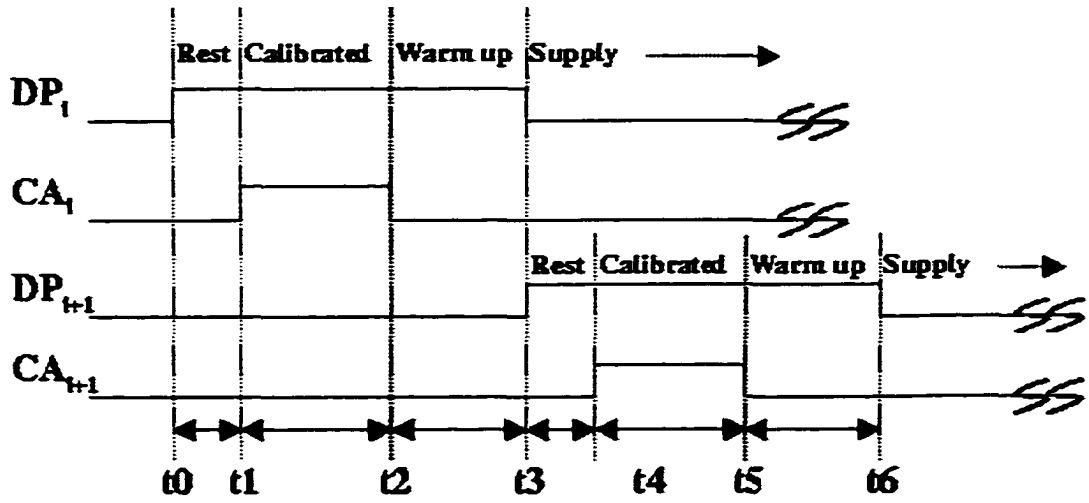


Figure 5.4 A timing diagram for two cells in continuous self-calibration

than the redundant mode. After every $(N+1)(t3-t0)$, a cell repeats its redundant mode.

The scheme minimizes the switching glitch, because only two current cells are involved in the switching comparing with three in the prior-art continuous self-calibration. In addition, the redundant cell is "warmed up" before entering its supplying mode. The warm-up time can be designed long enough for the cell to "practice" the output function by dumping the current to ground.

Figure 5.5 shows a block diagram for the new continuous self-calibration scheme. For simplicity, a 4-bit block is depicted, although more than a 4-bit block can be constructed with the same method. The 4-bit DAC includes 16 (here N=15) self-calibrated current cells, each of which can be a redundant cell dynamically. Each of the 16 cells is the same as the one shown in Figure 3 with their outputs Io and Iob tied together. A Single Pulse Generator, 4-bit Counter, 16-bit Shifter and Data Selector Array work together to control the self-calibration and output synchronization. The Single-Pulse Generator generates a negative pulse whenever there is a falling edge of the CLKR ( Calibration

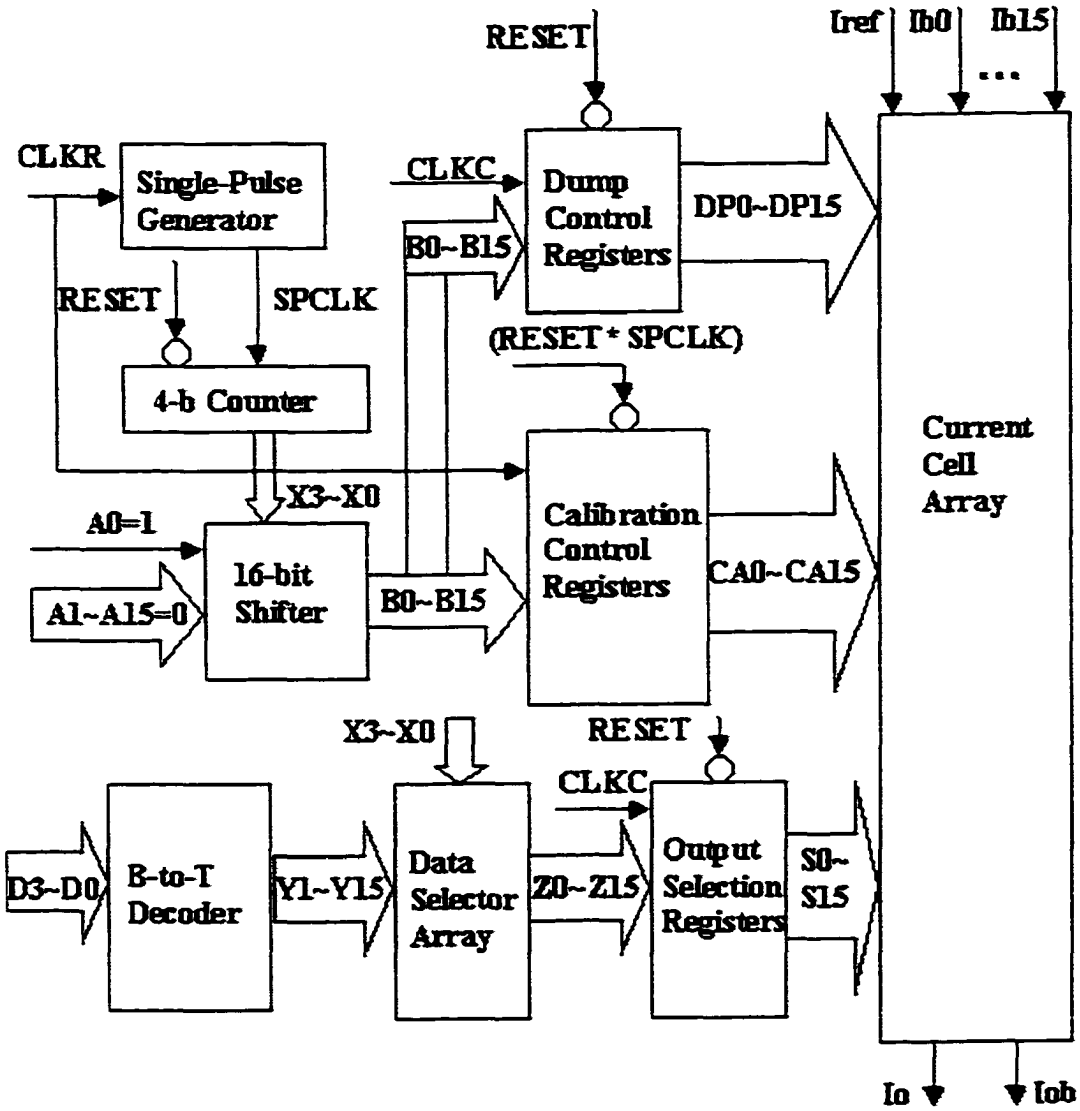


Figure 5.5 Redundant-cell-relay continuous self-calibration diagram

clock). The SPCLK is then applied to the 4-bit binary counter. The outputs of the counter X3~X0 are used to control which cell enters the redundant mode. This is done through the 16-bit Shifter and the Data Selector Array. With only one input bit equal to 1, the shifted-by-M ( M is equal to the decimal value of X3~X0) version of A15~A0, B15~B0 are fed to the Dump Control Registers and Calibration Control Registers. The position of the "1" in DP15~DP0 and CA15~CA0 determine which cell will enter the redundant mode. X3~X0 also controls the Data Selector Array to select those cells in supplying mode. A Binary to Thermometer Decoder is used to reduce the glitch caused by the MSB digital input data changes such as 0111 to 1000. The thermometer code Y15~Y1 is fed into the Data Selector Array controlled by X3~X0. Z15~Z0 is the code that guarantees which cells in supplying mode are selected. The coding function is shown in Table 5.2.

The timing diagram of the 4-bit block is shown in Figure 5.6. In the very beginning, the asynchronous RESET signal clears the 4-bit Counter, the Dump Control Registers, the Calibration Control Registers and the Output Selection Registers. Then the conversion clock CLKC and calibration clock CLKR are applied to the system. The first current cell enters into the redundant mode. The Single Pulse Generator generates its output signal SPCLK whose rising edge triggers the counter and increases the value of X3~X0 by 1. This new X3~X0 will change B15~B0. Thus the next cell enters into the redundant mode and is calibrated. After all the 16 cells have experienced at least one time the redundant mode, the input data can be converted. The 4-bit Counter output X3~X0 also synchronizes the calibration with the conversion by making the right selection of those current cells that are in the supplying mode. "Don't care" states can be utilized to simplify the decoding logic and increase the speed of logic circuits. All the bus signals are represented as single ended for simplicity, but in the implementation, complementary signals are used.

Simulation results using TSMC's 0.25$\mu$ single poly five metal logic CMOS process

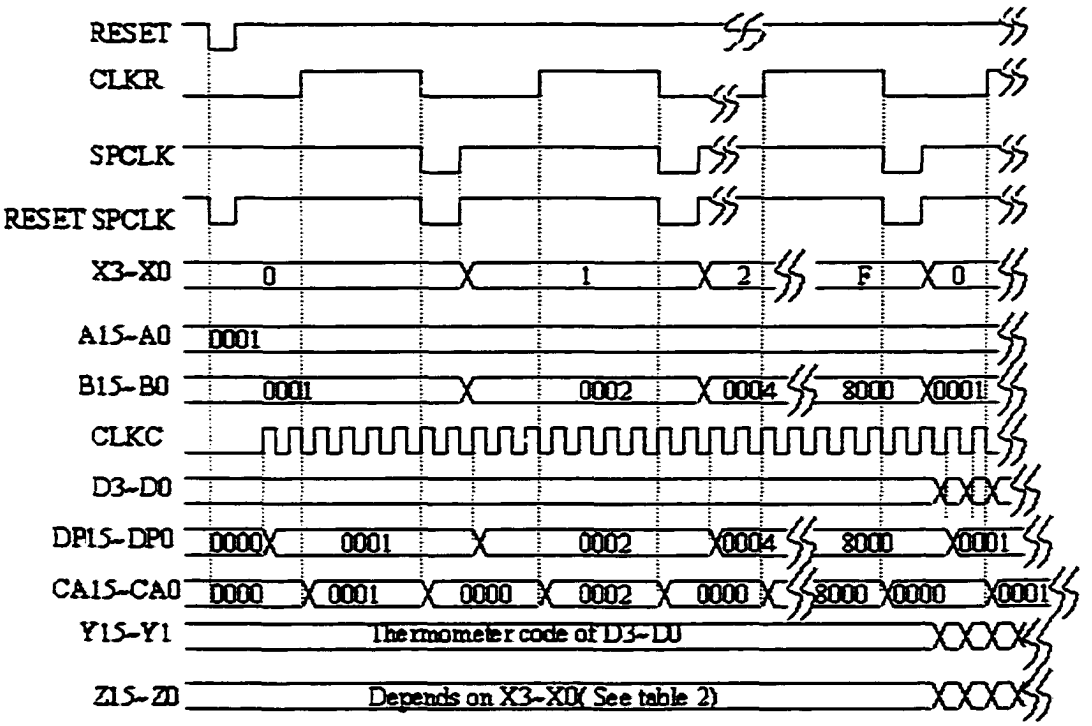


Figure 5.6 Timing diagram for 4-bit block

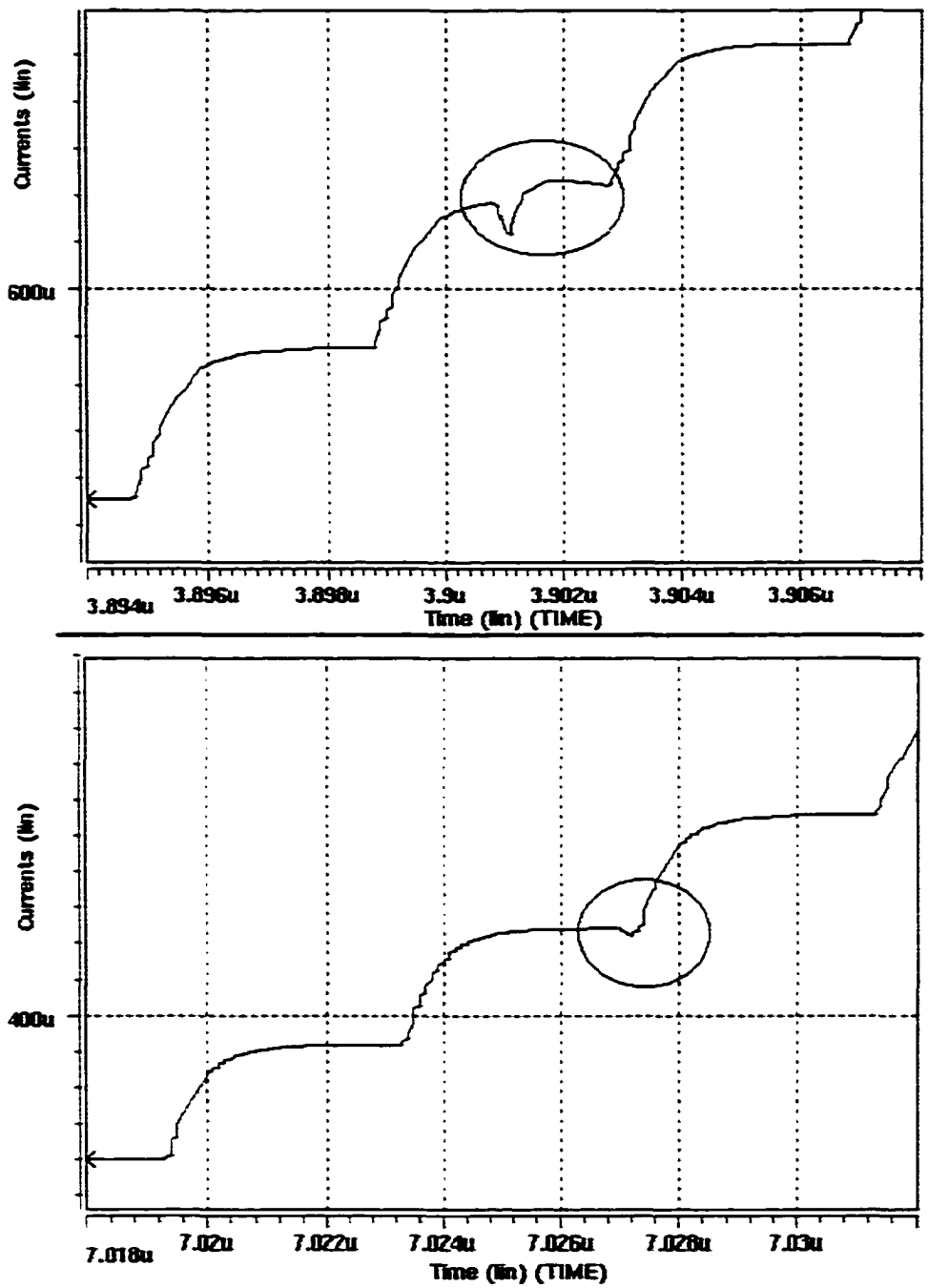


Figure 5.7 Comparison of prior-art and the new continuous self-calibration

and HSPICE are obtained. For comparison, a 4-bit DAC with prior-art continuous self-calibration [6] is also constructed with the same output load and the same conversion clock frequency (250MS/s). A digital ramp signal is converted into a staircase. In Figure 5.7, the top part is part of the output of the 4-bit DAC constructed using the prior-art continuous self-calibration. The bottom part is the result of the new redundant-cell-relay method. Both outputs show the glitches (inside the circles) that occurred during the switching from calibration to supplying. It is clear that the redundant-cell-relay continuous self-calibration glitches has much less ($\frac{1}{10}$ or less) energy than that in the prior-art. Furthermore, by using the new scheme, the output waveform settles to the expected value after the glitch happens, while the prior-art waveform cannot settle if the glitch happens, and as a result the residue value is the source of distortion and worse SFDR.

Table 5.2 Data selector array function table

| X3~ X0 | Z15~ Z0 |
|---|---|
| 0 | Zi=Yi, when i=1, 2, ...15. Z0=x( here x means don't care) |
| 1 | Zi=Yi, when i=2, 3, ...15. Z1=x, Z0=Y1 |
| 2 | Zi=Yi, when i=3, 4, ...15, Z2=x, Z1=Y2, Z0=Y1 |
| 3 | Zi=Yi, when i=4, 5, ...15, Z3=x, Zj=Yj+1 when j=0, 1,2 |
| 4 | Zi=Yi, when i=5, 6, ...15, Z4=x, Zj=Yj+1, when j=0,1, ...3. |
| 5 | Zi=Yi, when i=6, 7, ...15, Z5=x, Zj=Yj+1, when j=0,1, ...4. |
| 6 | Zi=Yi, when i=7, 8, ...15, Z6=x, Zj=Yj+1, when j=0,1, ...5. |
| 7 | Zi=Yi, when i=8, 9, ...15, Z7=x, Zj=Yj+1, when j=0,1, ...6. |
| 8 | Zi=Yi, when i=9,10, ...15, Z8=x, Zj=Yj+1, when j=0,1, ...7. |
| 9 | Zi=Yi, when i=10,11, ...15, Z9=x, Zj=Yj+1, when j=0,1, ...8. |
| 10 | Zi=Yi, when i=11,12, ...15, Z10=x, Zj=Yj+1, when j=0,1, ...9. |
| 11 | Zi=Yi, when i=12,13, ...15, Z11=x, Zj=Yj+1, when j=0,1, ...10. |
| 12 | Zi=Yi, when i=13,14,15, Z12=x, Zj=Yj+1, when j=0,1, ...11. |
| 13 | Zi=Yi, when i=14,15, Z13=x, Zj=Yj+1, when j=0,1, ...12. |
| 14 | Z15=Y15, Z14=x, Zj=Yj+1, when j=0,1, ...13. |
| 15 | Z15=x, Zj=Yj+1, when j=0,1, ...14. |

## 5.3 Summary

In this chapter, a new redundant-cell-relay (RCR) continuous self-calibration method is described. The contributions here include a self-calibrated current cell structure with relay control, the timing for each cell, the functional architecture to realize a DAC and the timing diagram for all the involved blocks. Simulations of the constructed 4-bit DACs proved that the RCR continuous self-calibration greatly improved the glitch performance. The key in this invention is the relay style mechanism of the continuous self-calibration method which reduces drastically the glitch during switching from calibration to supplying mode. This allows the achievement of high accuracy without compromising the high-speed operation and the achievement of high dynamic performance.

# CHAPTER 6. HIGH-SPEED AND HIGH-RESOLUTION DAC PROTOTYPES

In this chapter, high-speed and high-resolution DAC prototypes will be presented. The objectives are to build workable prototypes as well as optimal configurations for modular structures. The basic building blocks are put into a library for later synthesis usage. The experience on the building of the module library can be used to guide the synthesis process.

In Chapter 4, an 8-bit prototype and the experimental results were shown. So in this chapter, we will construct the 6-bit, 10-bit and 12-bit DACs. The 10-bit and 12-bit prototypes are based on the redundant-cell-relay continuous self-calibration because of the high-resolution potential without compromising the high-speed. A 10-bit DAC prototype was fabricated in TSMC's 0.25$\mu$ single poly five metal logic CMOS process to prove the effectiveness of the RCR method. The design, implementation, simulation and testing results for the 10-bit DAC prototype will be presented. Simulation results for the 12-bit DAC are also shown.

## 6.1 6-Bit DAC

A 6-bit DAC can be constructed by a 4-bit block plus a 2-bit block. Figure 6.1 shows the functional diagram of a 6-bit DAC. The 4-bit LSB block in the 8-bit prototype described in Chapter 4 is reused here. The 2-bit DAC block is constructed using binary weighted current cells because of the relaxed accuracy requirement. The 6-bit block is

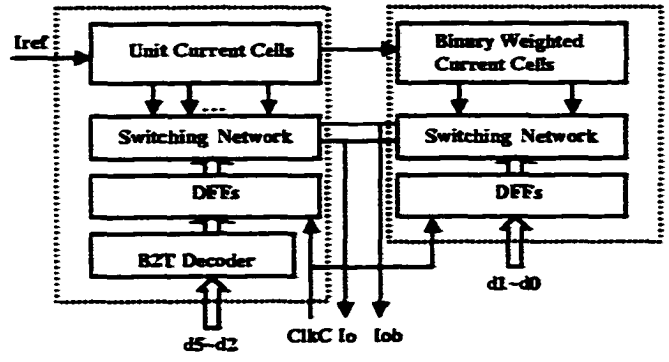


Figure 6.1 6-bit DAC

also a building block for the 10-bit DAC.

## 6.2 10-Bit DAC

### 6.2.1 Block Diagram

Due to the modular architecture, a 10-bit DAC can be constructed by a 4-bit MSB plus a 6-bit LSB. Figure 6.2 shows the functional diagram of a 10-bit DAC. The dashed block L6 is a simplified representation of the 6-bit DAC shown in Figure 6.1.

### 6.2.2 Implementation

The 10-bit prototype is implemented in TSMC's 0.25$\mu$ single poly five metal logic CMOS process. Here the redundant-cell-relay continuous self-calibration method described in Chapter 5 is used in the 4-bit MSB in order to remove transition glitches. The 4-bit MSB is shown in Figure 5.5. Two reference current inputs Iref1 and Iref2 are introduced to bias separate blocks. Their nominal values are the same. In Figure 6.2 the 1:1 mirror of Iref1 is used to calibrate the current cells of the 4-bit MSB. The Iref2 is mirrored to generate the bias currents for the self-calibrated cells and the unit currents for the 6-bit DAC. The matching error between the different blocks can be adjusted by

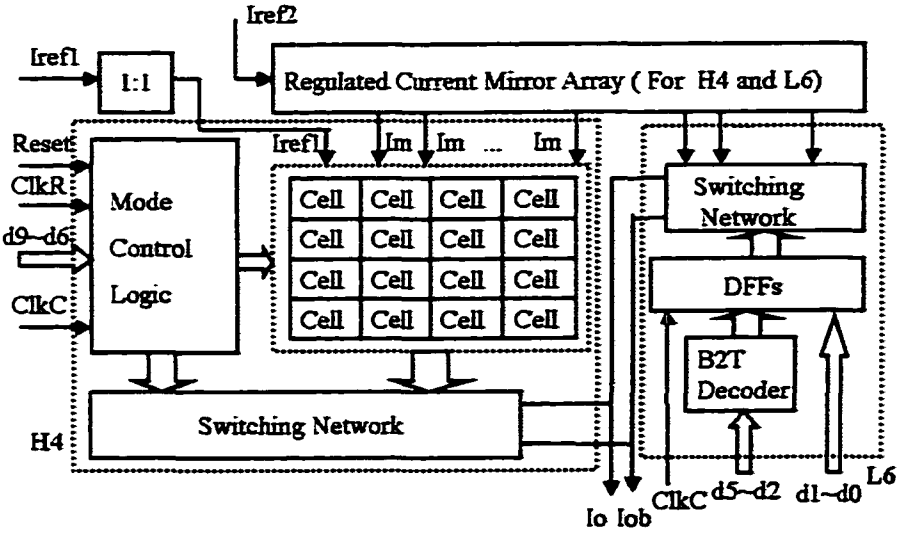


Figure 6.2 10-bit 250MS/s DAC

changing the ratio of Iref1:Iref2. This change is limited to the extent that the current mirrors are still working in saturation.

Simulations were performed for different process and temperature corners. The circuit works fine in different environmental temperatures (-40, 25, 100°C) with Slow-Slow, Typical-Typical and Fast-Fast models. Slow-Slow models at 100°C is the worst case. The output load for the simulation is a 50 Ohm resistor in parallel with a 10pF capacitor. The timing diagram is similar to Figure 5.6. At first, 16 current cells in the 4-bit MSB block are calibrated sequentially, then a series of digital ramp input codes are applied at the conversion rate of 250MS/s while the calibration continues in background. The simulation of the 10-bit DAC with a ramp digital input is shown in Figure 6.3. The maximum error is less than 0.5uA ( 1uA is one LSB here). The glitch energy caused during the transitions of the current cells' operation modes is $\frac{1}{10}$ of that in the 8-bit DAC prototype described in Chapter 4.

The layout is implemented in TSMC's 0.25$\mu$ single poly five metal logic CMOS process and shown in Figure 6.4.

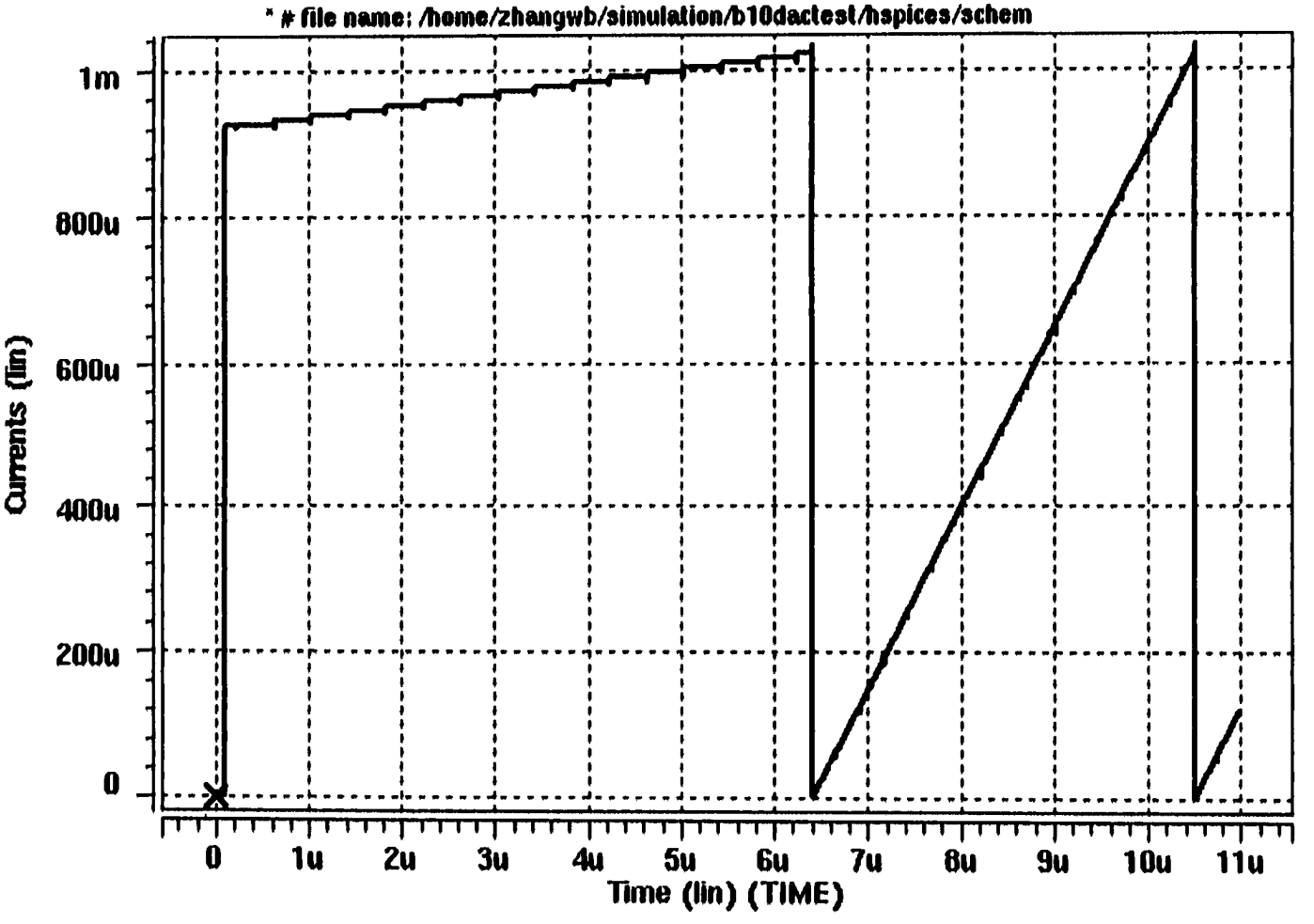


Figure 6.3 10-bit 250MS/s DAC Simulation

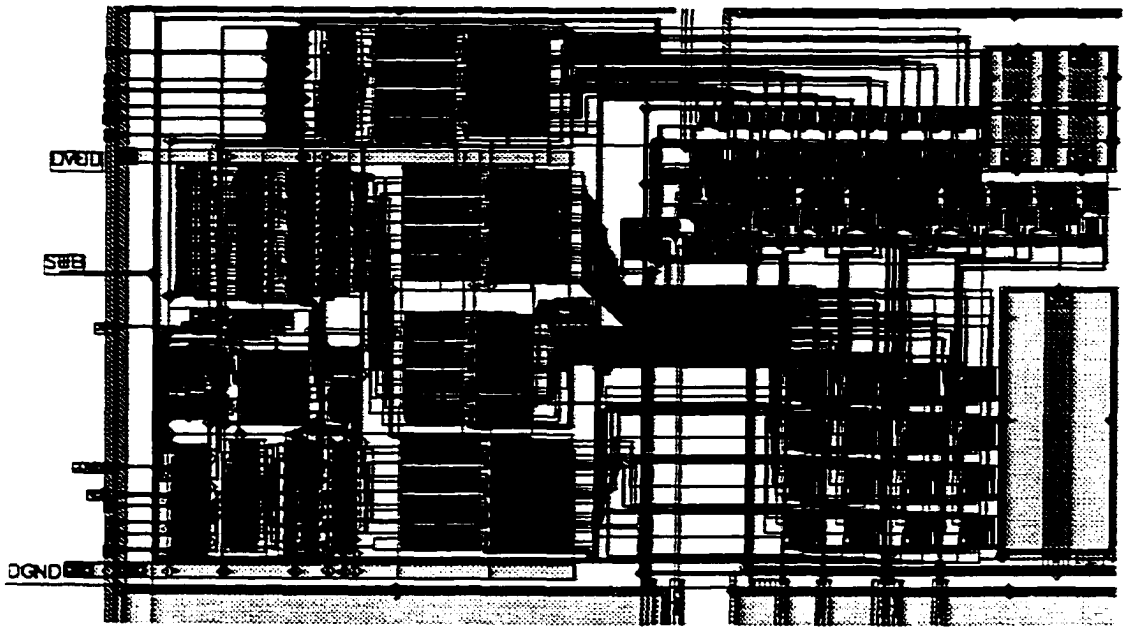


Figure 6.4 10-bit 250MS/s DAC Layout

Symmetry is applied to match the transistors especially those that need matching in the bias-current sources. In the bias current block layout, interdigitized and common-centroid layout techniques are applied to the critical transistors that require matching. Digital circuits and analog circuits are separated by a guard-ring which prevents the digital noise propagating to the analog cells. Analog power and digital power lines are also separated to minimize cross coupling. Routing from different flip-flops to the switching network is carefully done in order to achieve approximately equal wiring lengths for all the digital buses. As in Chapter 4, current cells are arranged randomly to reduce spatial distribution errors. In addition, on-chip decoupling capacitors are introduced in order to get a clean DC power supply to the DAC core. Without the decoupling capacitor, the die size is 0.3mm$^2$.

## 6.3 12-Bit DAC

Figure 6.5 shows the functional diagram of the 12-bit DAC. A 12-bit DAC is constructed by reusing the self-calibrated 4-bit MSB H4 plus an 8-bit DAC L8. The 8-bit DAC reuses the 4 bit LSB from the prototype 8-bit DAC as its MSB. A 4-bit binary weighted DAC is used as the least significant bits block for the 12-bit DAC. By doing so, the accuracy requirement for the 4-bit MSB is addressed by the continuous self-calibration. The remaining blocks need to be scaled properly to match with the 4-bit MSB.

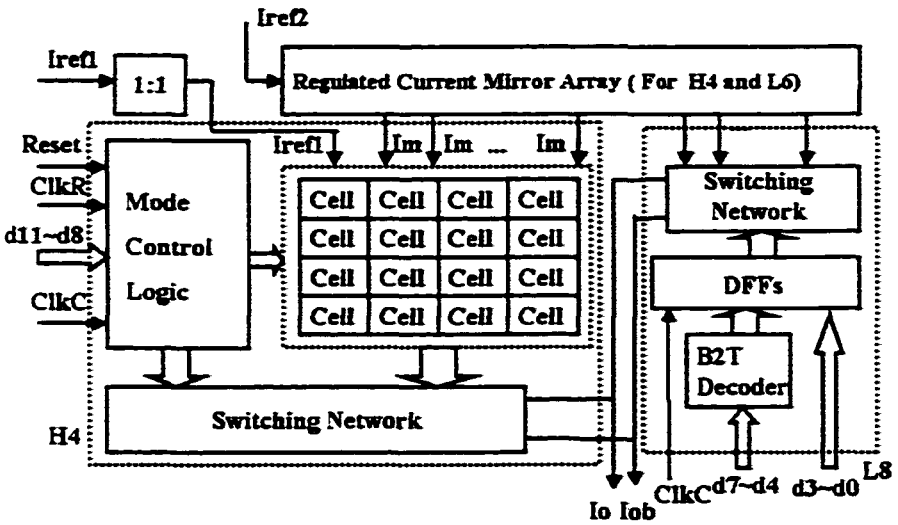


Figure 6.5 12-bit 250MS/s DAC

The output load for the simulation is a 50 Ohm resistor in parallel with a 5pF capacitor, because the settling to $\frac{1}{2}$ LSB of 12-bit need 9 RC constant given equation 3.26. For 250MS/s, the settling time is less than 4ns, so the RC constant need to be less than 0.444 ns. Given RL=50 Ohm, C need to be less than 8.88 pF. For a safety margin, C=5pF is used. The simulation of the 12-bit DAC with a ramp digital input is shown in Figure 6.6.

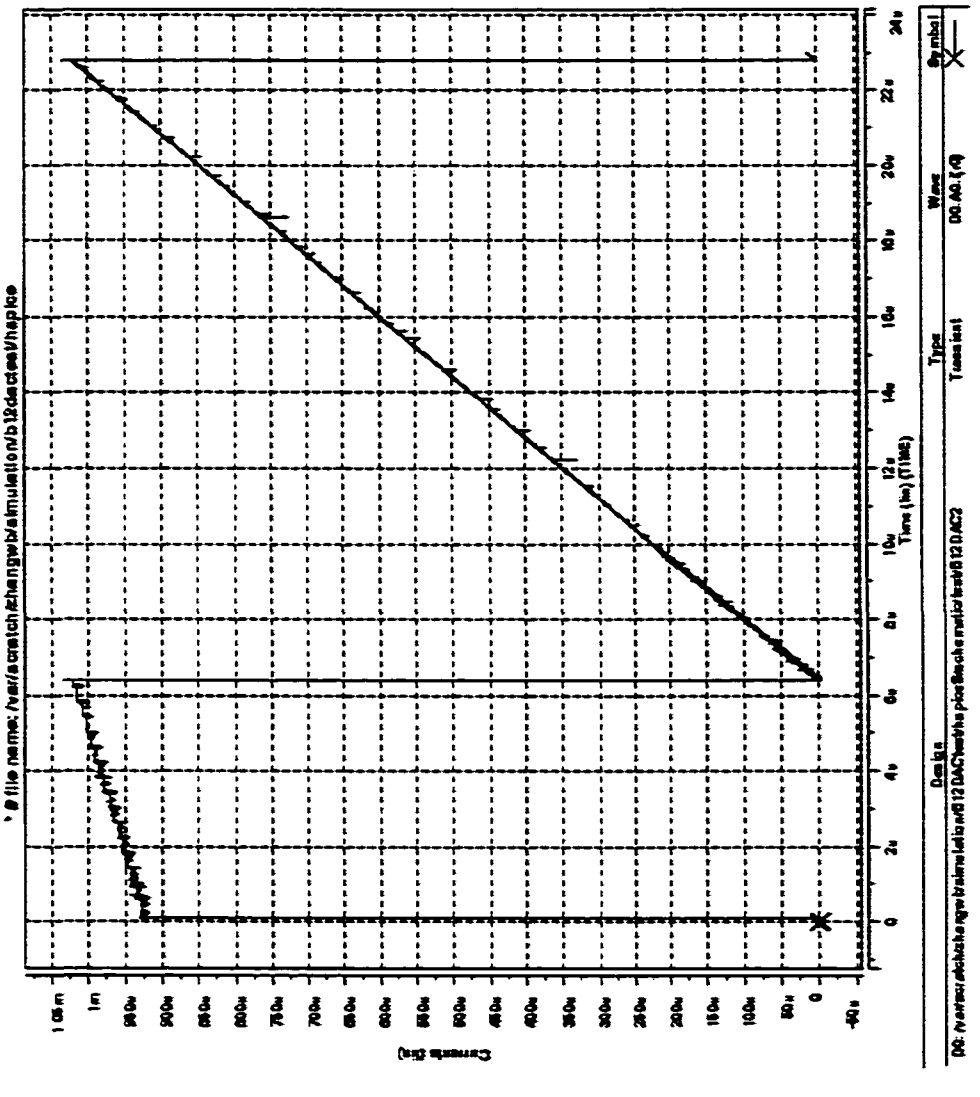

Figure 6.6 12-bit 250MS/s DAC Simulation

## 6.4 10-Bit DAC Testing Results

The micrograph of the multisystem chip is shown in Figure 6.7. The DAC is located in the upper left corner. It occupies only $0.3mm^2$. The zoomed micrograph of the DAC itself is shown in Figure 6.8. Smaller area could be achieved if the spare spaces between the sub-blocks are reduced further.

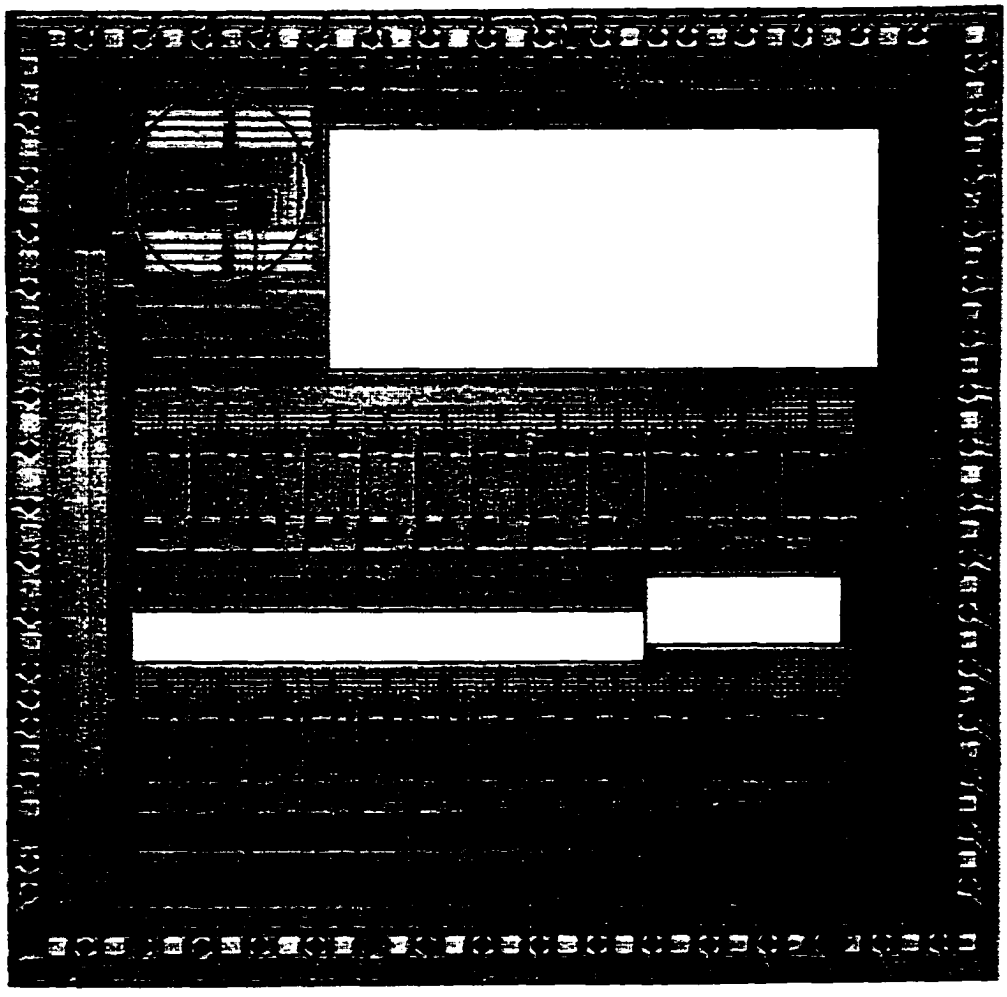

Figure 6.7 Micrograph of the DAC in a multisystem chip

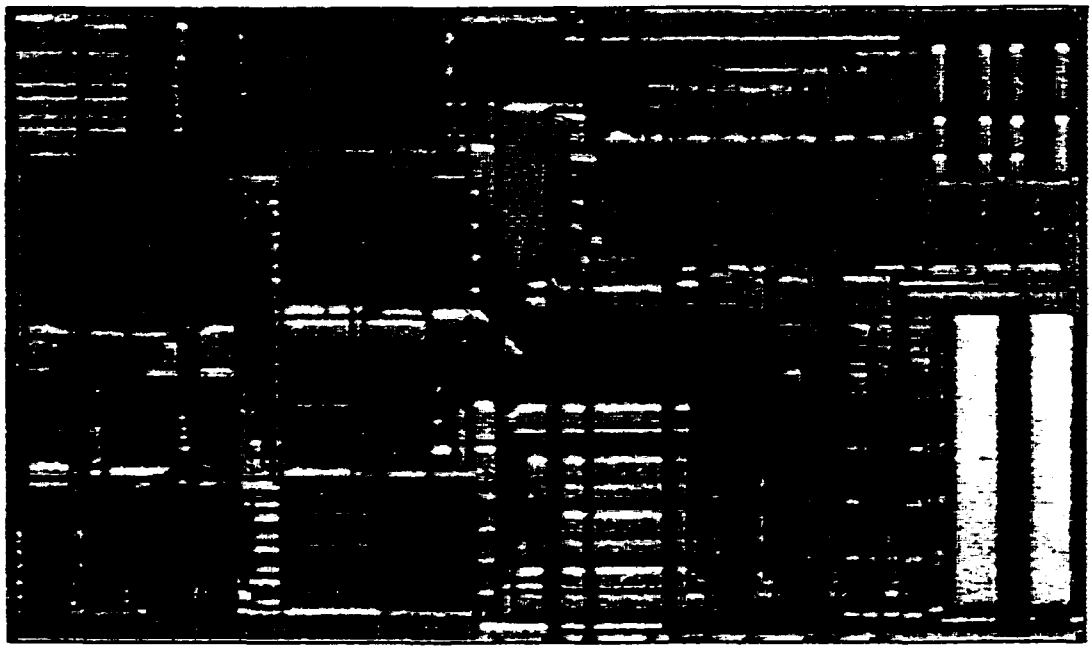

Figure 6.8 Zoomed micrograph of the 10-b DAC

### 6.4.1 Testing Setup

The testing-setup used to measure DC performance as well as dynamic performance is similar to Figure 4.19. There are some small differences. One is the calibration clock is supplied from an independent clock generator (HP 3326A two channel synthesizer). An Agilent 8664A 0.1-3000MHz synthesized signal generator triggers an HP 8131A 300MHz pulse generator that generates low glitch complementary reference clocks CLK and CLKN. CLK is used as a reference clock for the Sony/Tecktronix DG2020A data pattern generator to provide digital sine wave ($D9 \sim D0$). CLKN is used as a low jitter sampling clock source, it is filtered by a low pass filter to get a clean sine wave source (f=Fc) which is used as the conversion clock of the DAC. Since the DAC has input buffers following the digital input pads, the sine wave is converted to a pulse clock internally. By doing so, a lot of coupling noise triggered by high frequency transitions

are avoided. The outputs are observed by Rohde&Shwarz FSEM30 spectrum analyzer.

Since the maximum speed of the data pattern generator is 200MBPS, the maximum frequency of the input digital sine wave is less than 100MHz. In the Nyquist rate testing, a 99MHz input digital sine wave is tested. For sampling rate of 250MS/s or more, an asynchronous method is used. This is the same as the 8-bit DAC testing.

The PCB is shown in Figure 6.9. Several techniques are applied to the testing board to reduce parasitic effects and coupling effects from unwanted signal. Four-layer board is designed in order to have clean and separated power supplies. De-coupling capacitors are distributed on the board to remove the unwanted high frequency signals coupled in the power plane and signal lines. Digital buses on the board are laid out carefully to have equal lengths.

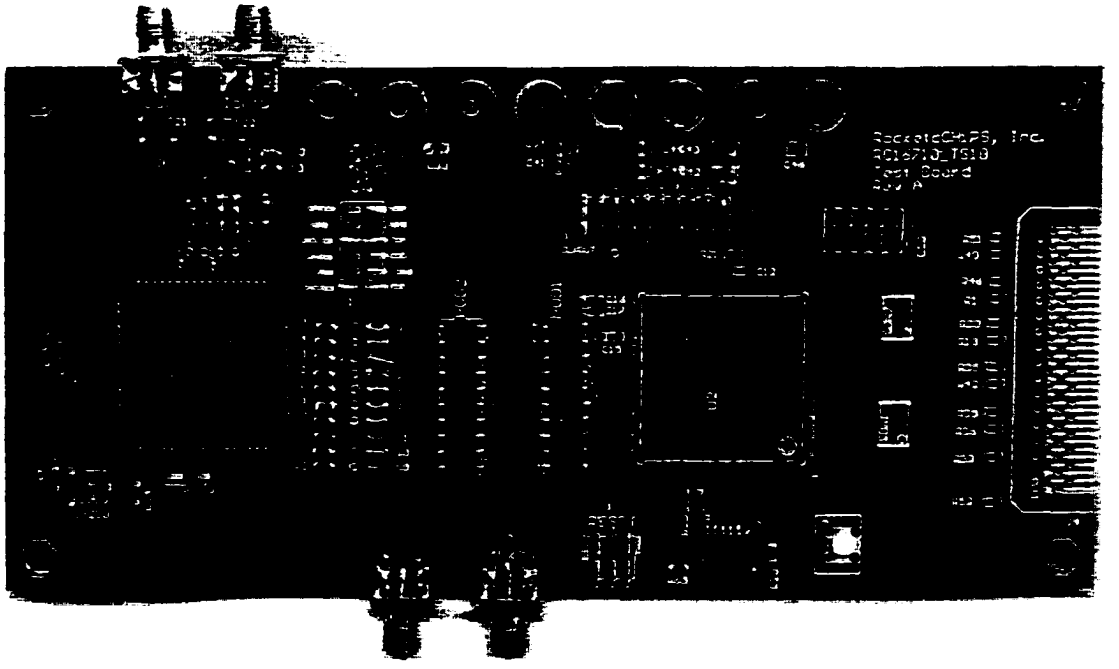

Figure 6.9 PCB for the 10-b DAC

### 6.4.2 DC Results

The INL and DNL plots are obtained after the transfer curve is measured. The transfer curve is got by applying a digital ramp codes to the DAC at the conversion rate of 50MS/s. Eight chips were tested, and all of them were functional. A typical INL and DNL measurement is shown in Figure 6.10. The segmented behavior is obvious, this is due to the matching error between the 4-bit MSB block and the 6-bit LSB block. From Figure 6.10, less than +/-0.5 LSB INL and +0.2/-0.45 LSB DNL are achieved.

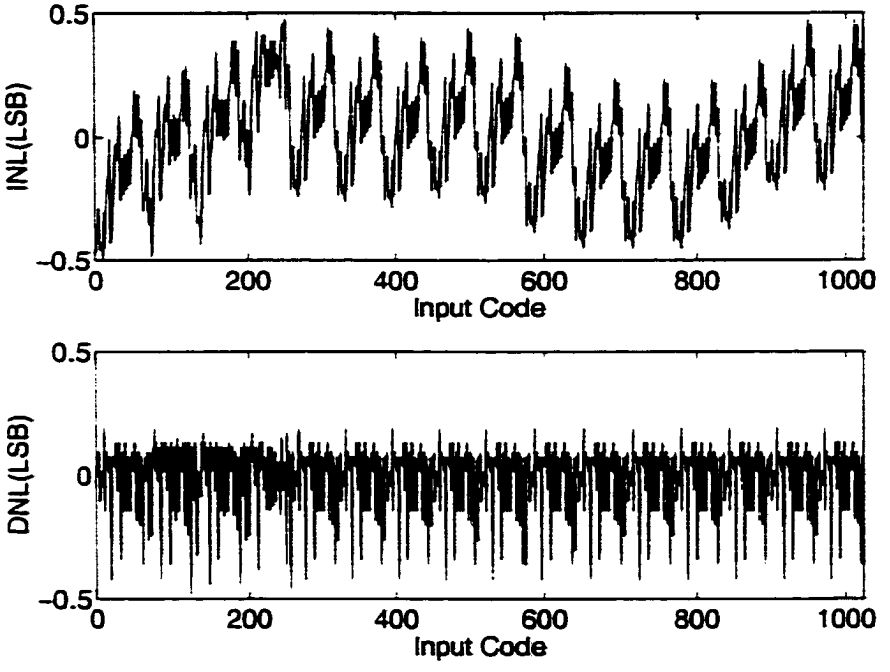


Figure 6.10 Measured INL and DNL of the 10-b DAC

### 6.4.3 Leakage Testing

In continuous self-calibration, leakage is an important factor for determing the calibration clock. Table 6.1 shows the measurement results. Row 1 shows the voltage on the output node when all the 15 self-calibrated current cells are supplying current. Row

2 shows the voltage on the same node 10 seconds after the conversion clock (50MS/s) and calibration clock (1.1MHz) are turned off. The above measurement is based on the assumption of the leakage behavior derived in equation 3.15 and leakage time for each cell is equal ( the period of 1.1MHz clock is negligible comparing to 10 seconds). Row 3 showed the leakage rate. The formula for the lowest calibration clock frequency is given in equation 3.21. Here, the 4-bit MSB is self-calibrated, and 10-bit accuracy needs to be maintained. Given the testing results, the minimum frequency of calibration clock is 176 Hz.

Table 6.1 Relative Leakage Ratio testing result ( 10 seconds)

| Start(mv) | 48.13 | 48.12 | 48.14 | 48.11 | 48.11 | 48.11 | 48.11 |
|---|---|---|---|---|---|---|---|
| Stop(mv) | 42.61 | 42.57 | 42.60 | 42.56 | 42.55 | 42.57 | 42.60 |
| RLR( %/s) | 1.147 | 1.153 | 1.151 | 1.154 | 1.156 | 1.152 | 1.145 |
| Average RLR (%/s) | | | | | | | 1.151 |

### 6.4.4 Dynamic Performance Results

Figure 6.11 shows when the input digital sine wave is 24.4KHz (50M/2048) and the conversion rate is 250MS/s. An SFDR of 70dB is measured. The noise floor is lower than the 8-bit DAC. The digital and analog power planes are separated. An SNR of 59dB is measured using the noise density test.

Nyquist rate testing is done when the input frequency is 99MHz, and the conversion rate is 200MS/s. Figure 6.12 shows a typical test result with the SFDR=53dB. It should be noted that the true performance of the DAC is difficult to measure because of the non-ideal PCB and equipment limitations. Noise clock signals and power supplies can deteriorate the SFDR. Different signals from the input digital bus with different delay times are causing the skewed input data and harmonic distortion. It would be better if shorter and matched traces are drawn on the PCB. Another factor is the unmatched

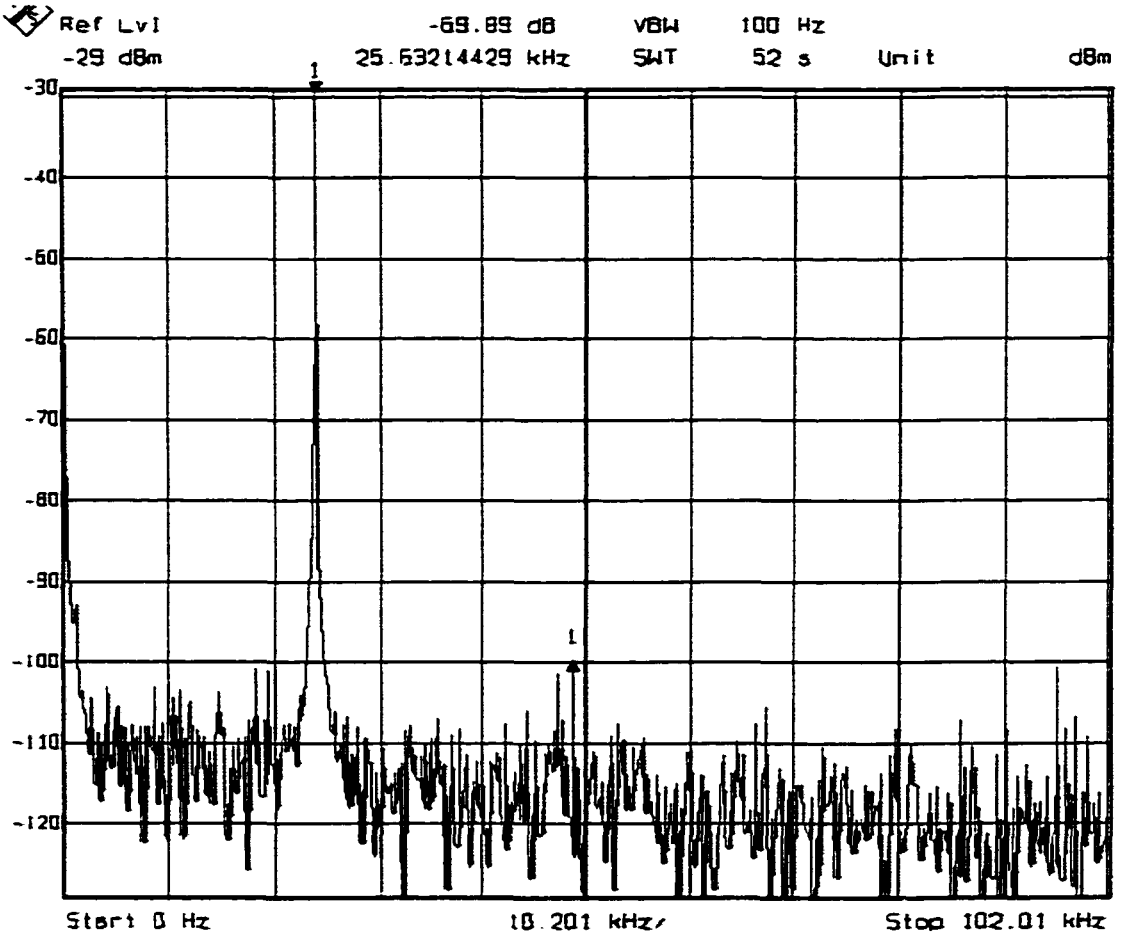


Figure 6.11 SFDR when Fin=24.4K, Fs=250MHz

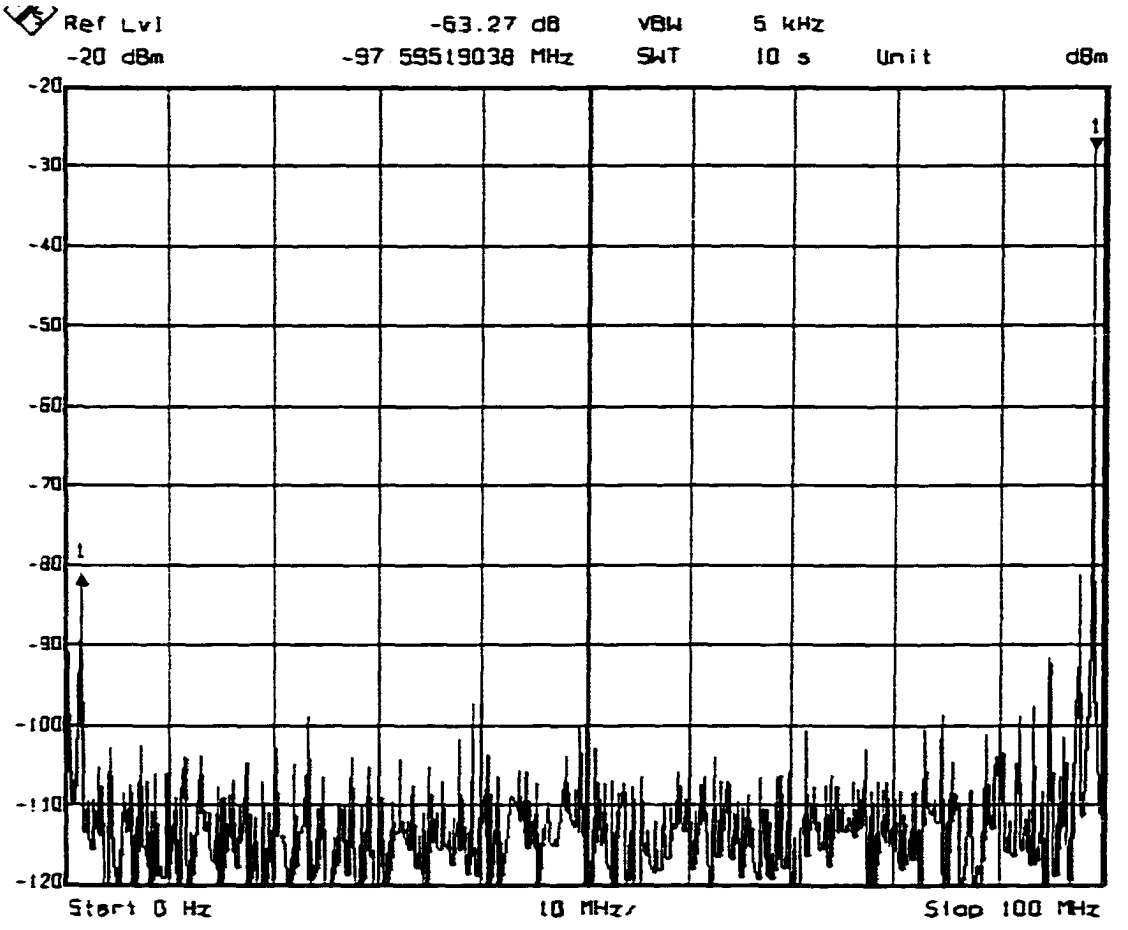


Figure 6.12 SFDR when Fin=99MHz, Fs=200MHz

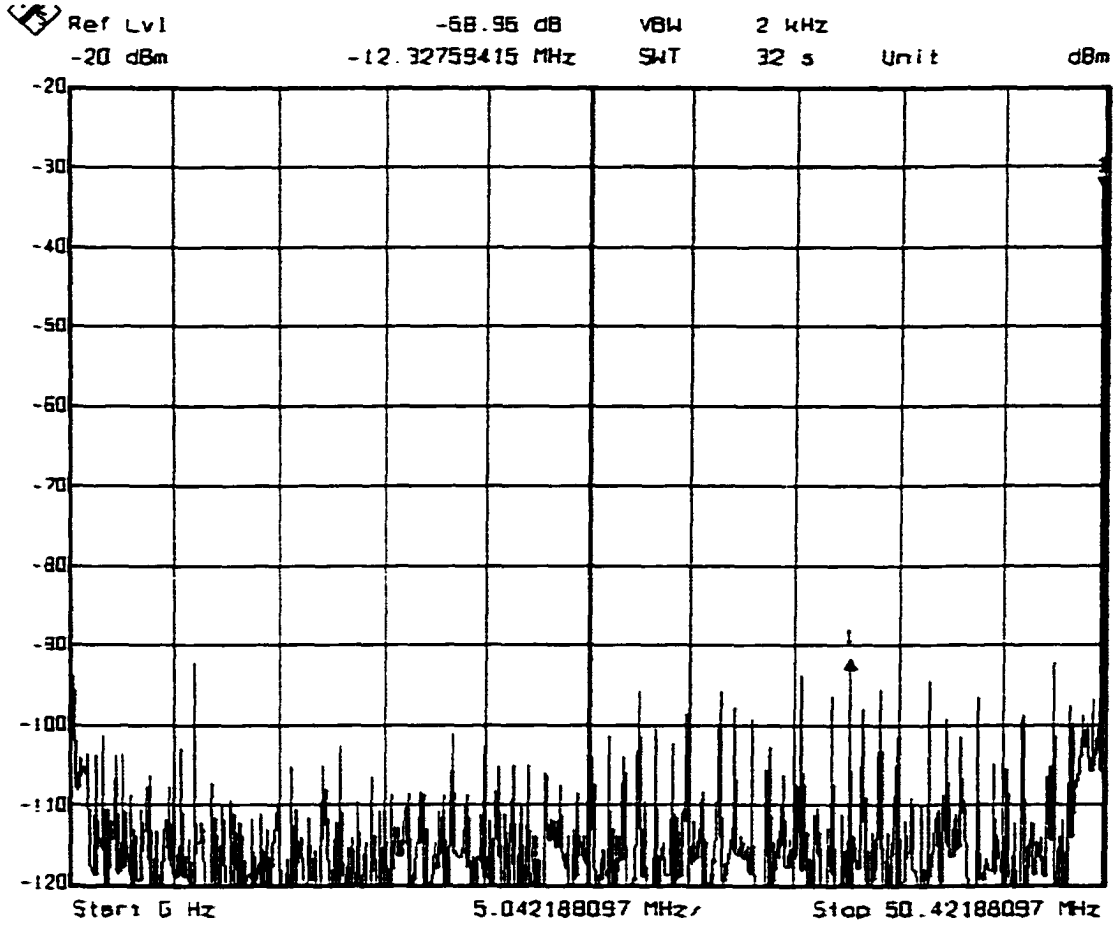


Figure 6.13 SFDR when Fin=49MHz. Fs=100MHz

impedance and the reflections caused from it. These factors are less harmful when the conversion speed and the input frequency are lower. Figure 6.13 shows an SFDR of 59dB when the input frequency is 49MHz and the conversion rate is 100MS/s. The measured DAC characteristics are summarized in Table 6.2.

Table 6.2 Measured 10-bit DAC characteristics

| | |
|---|---|
| Technology | 0.25$\mu$ CMOS |
| INL | < 0.5 LSB |
| DNL | < 0.45 LSB |
| Update Rate | 200MS/s |
| Full Scale Output Range (RL=50 Ohm) | 1mA |
| SFDR (Fin=24.4KHz) | 70dB |
| SFDR (Fin=99MHz) | 53dB |
| SFDR( Fin=49MHz, Fs=100MHz) | 59dB |
| Power Consumption | 8mW |
| Supply (Analog/Digital) | 2.5V/2V |
| Die Area | 0.3mm$^2$ |

Compared with recently reported high-performance DACs [27], [33], [32], this DAC shows the advantages of low-power and small area (see Table 6.3). The top two DACs are extremely high-speed, with very high power consumption. The third DAC consumes more power and more area at a lower conversion rate than this work. In addition, the third DAC's Nyquist SFDR drops to less than 45dB.

Table 6.3 Comparison of different high-performance DACs

| Results | Power | Area | Speed | Resolution | Nyquist SFDR |
|---|---|---|---|---|---|
| JSSC9812 | 125mW | 0.6mm$^2$ | 500MS/s | 10b | 51dB |
| JSSC0103 | 110mW | 0.35mm$^2$ | 1GS/s | 10b | 61dB |
| ESSC00 | 20mW | 1mm$^2$ | 100MS/s | 14b | < 45dB |
| This work | 8mW | 0.30mm$^2$ | 200MS/s | 10b | 53dB |

## 6.5 Summary

In this chapter, the modules and prototypes for high-speed high-resolution DACs were designed. A 10-b DAC using the redundant-cell-relay continuous self-calibration method was fabricated in TSMC's 0.25$\mu$ single poly five metal logic CMOS process. The testing results proved the effectiveness of the redundant-cell-relay method. Compared with recent published high performance DACs, this DAC shows a great potential for many embedded applications because of its high-speed, high-resolution, low-power and small area. During the design process, 6-bit to 12-bit DAC modules were built. By constructing workable prototypes, mixed-signal synthesis according to modular design can be implemented easily. Although the prototypes only represented part of the exploration space for synthesis, they laid down a basis for future expansion.

# CHAPTER 7. DAC SYNTHESIS ALGORITHM

In this chapter, a DAC synthesis tool is developed. Based on the modular architectures and the building modules, a simplified version of a direct-mapping algorithm discussed in [9] can be used to synthesize the DACs. Function mapping and layout generation can be done at the same time and synthesis is performed quickly. When a user supplys the specifications such as number of bits, full range of outputs, and technology the synthesis algorithm will map the requirements to the closest solution existing in the library. The tool and the libraries can be extended concurrently whenever new design and new modules are ready.

## 7.1 DAC Functional Graph

Considering the modular architecture proposed in Chapter 3 and the direct mapping synthesis algorithm, a functional graph of a DAC system is shown in Figure 7.1. Here the graph represents the modular DAC architecture. The graph is defined as a pair of sets G=(N,E), where N is a set of nodes, and E is a set of pairs of different nodes called edges. N and E are defined as follows:

N= { $n_i \| n_i$ is a node that represents a component }

E= { $(n_i, n_j) \| n_i, n_j \in$N. the component represented by $n_i$ contains one or more or is realized by $n_j$ }

Note that the edges could have two types. Type 1 edge represents a "contains one or more" relation between $n_i$ and $n_j$. We use dashed arrow to denote a type 1 edge. Type

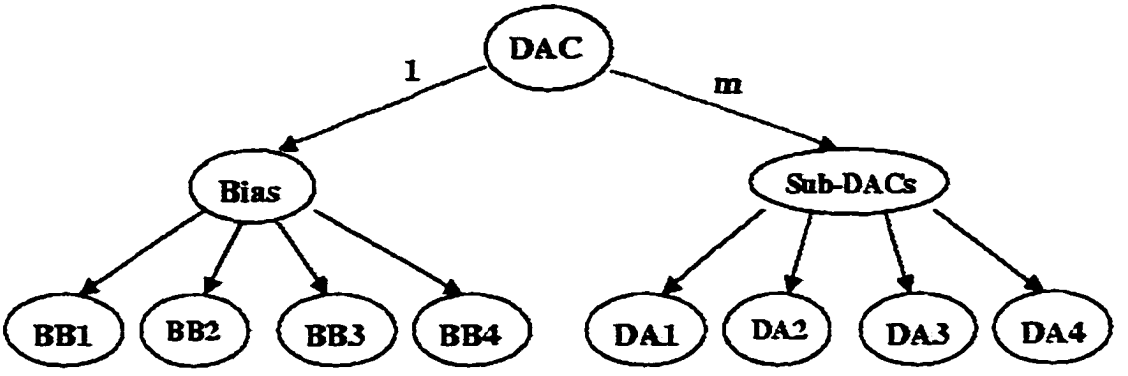


Figure 7.1 A functional diagram of DAC

2 edge represents a "realized by" relationship between $n_i$ and $n_j$ which is denoted by a solid arrow. Propagation through the function graph is guided by user requirements, design knowledge and specifications of each component. In Figure 7.1, each of four BBi (i=1 to 4) represents a biasing block, and each of four DAi ( i=1 to 4) represents a sub-DAC block. The edges from node Bias to nodes BBi (i=1 to 4) are type 2 edges. This means Bias is realized by one of the four BB's. Similarly a sub-DAC is realized by one of the four DA's. The root node DAC has type 1 edges to nodes Bias and sub-DACs. On the type 1 edges, the numbers 1 and m means a DAC contains one Bias block and multiple sub-DACs.

## 7.2 Synthesis Algorithm

Direct mapping synthesis algorithm will propagate the functional graph and select proper components needed for a given specification. The propagation through the graph needs some design knowledge, but the modular design simplifies the selection procedure. A simple algorithm is shown in Figure 7.2. The input is the user specifications. The program will read the specifications and map the specifications to corresponding blocks in the graph. The topology is naturally generated according to the blocks and their con-

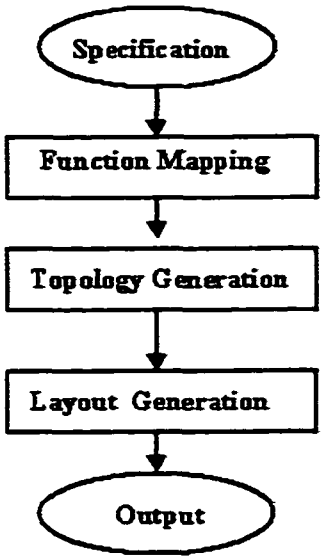


Figure 7.2 A flow diagram for synthesizing DAC

nections. In the implementation, the topology is represented by a hierarchical schematic in CADENCE. According to the schematic blocks, the corresponding layout blocks can be loaded in and the layout representation is obtained.

## 7.3 Implementation

The modules in the library should have good portability. It is not necessary to modify a module in order to work with other modules. They could be easily combined to form a whole DAC. This is good for a simple synthesis algorithm. Different kinds of blocks should have no functional overlap. Careful consideration of the parameters are needed. For example, when we want to use a 4-bit block A plus a 2-bit block B to construct a 6-bit DAC, the current scale of the two blocks should be matched, that is, the unit current of block A is 4 times that of block B. Biasing block for the 6-bit DAC is needed in order to guranttee the scale between the two blocks.

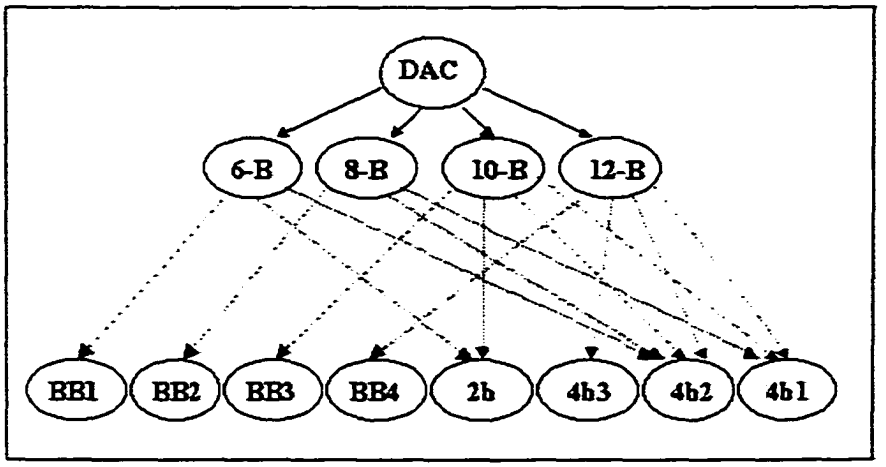


Figure 7.3 A functional mapping graph for synthesis of DAC

Some functional specification parameters of these blocks are needed to associate with the block in the library. For example, a 4-bit cell should have the following specification: Maximum operation speed, resolution, unit current value, inputs and outputs, references, etc.

At the same time, the structural information of each block is needed too. Since the topology will be generated, the interconnection relationship between different function blocks should be incorporated somewhere. PIN names, input/output properties etc are the least required information.

Knowledge based realization is suggested since the design of the mixed-signal circuits involves a lot of knowledge in different domains. Due to the CADENCE hierarchical representation, all of the above requirements could be realized in a hierarchical schematic cell view. By building up the schematic and the layout modules, the synthesis process is easily realized . The functional parameters and the topology information are easily realized by the CADENCE database. This simplifies the knowledge collection.

Based on the modules built in the current library, the DAC functional graph is constructed as in Figure 7.3. Users specification could be mapped into one node realized

by corresponding blocks. Since structural and layout information is in the library, the task of the synthesis tool is just loading in all blocks needed and displaying the combined cellview to the user.

A simple prototype synthesis program is coded in the SKILL language, which is the script language used to develop the CADENCE. The advantage of using SKILL is that the program is compatible with all CADENCE versions. The program can be upgraded easily when the synthesis library grows. The synthesis menu can be added by simply loading the SKILL files as shown in Figure 7.4.

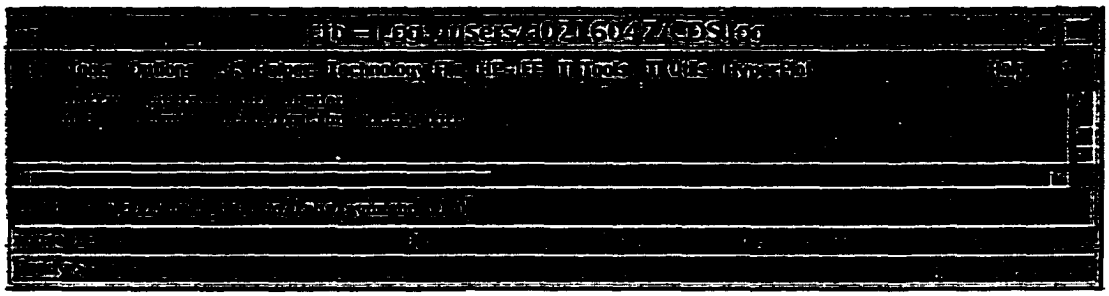

Figure 7.4 Loading of the synthesis program

The synthesis menu item is added in the left of the icfb window as shown in Figure 7.5. If clicked, the pull-down menu, "Synthesis" will show two simple menu items for selection: DAC Synthesis or Quit. Figure 7.6 shows the specification input form when DAC Synthesis is clicked. Three fields need to be filled by the user: Technology, Number of Bits and Full Scale Range. A user can select one of several process technologies. After the specifications are given, the program will synthesize a DAC and shows the schematic as in Figure 7.7 and the layout is generated in Figure 7.8. The routing between the blocks can be done manually or automatically by existing commercial tools. If the top level layout is proved by the prototype silicon, the layout can be loaded directly.

Appendix shows the SKILL codes for the synthesis program.

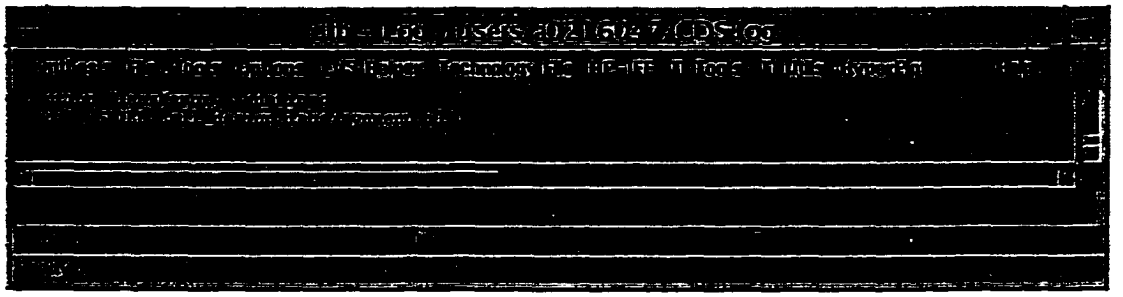

Figure 7.5 Synthesis menu added after loading

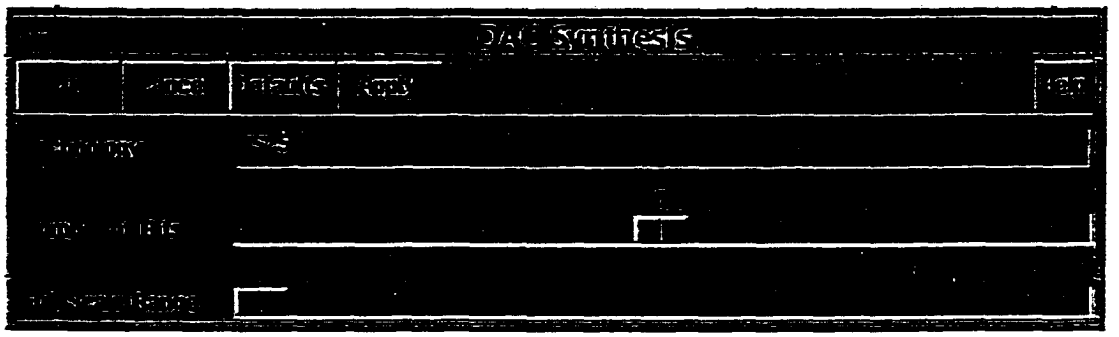

Figure 7.6 DAC synthesis specification input form

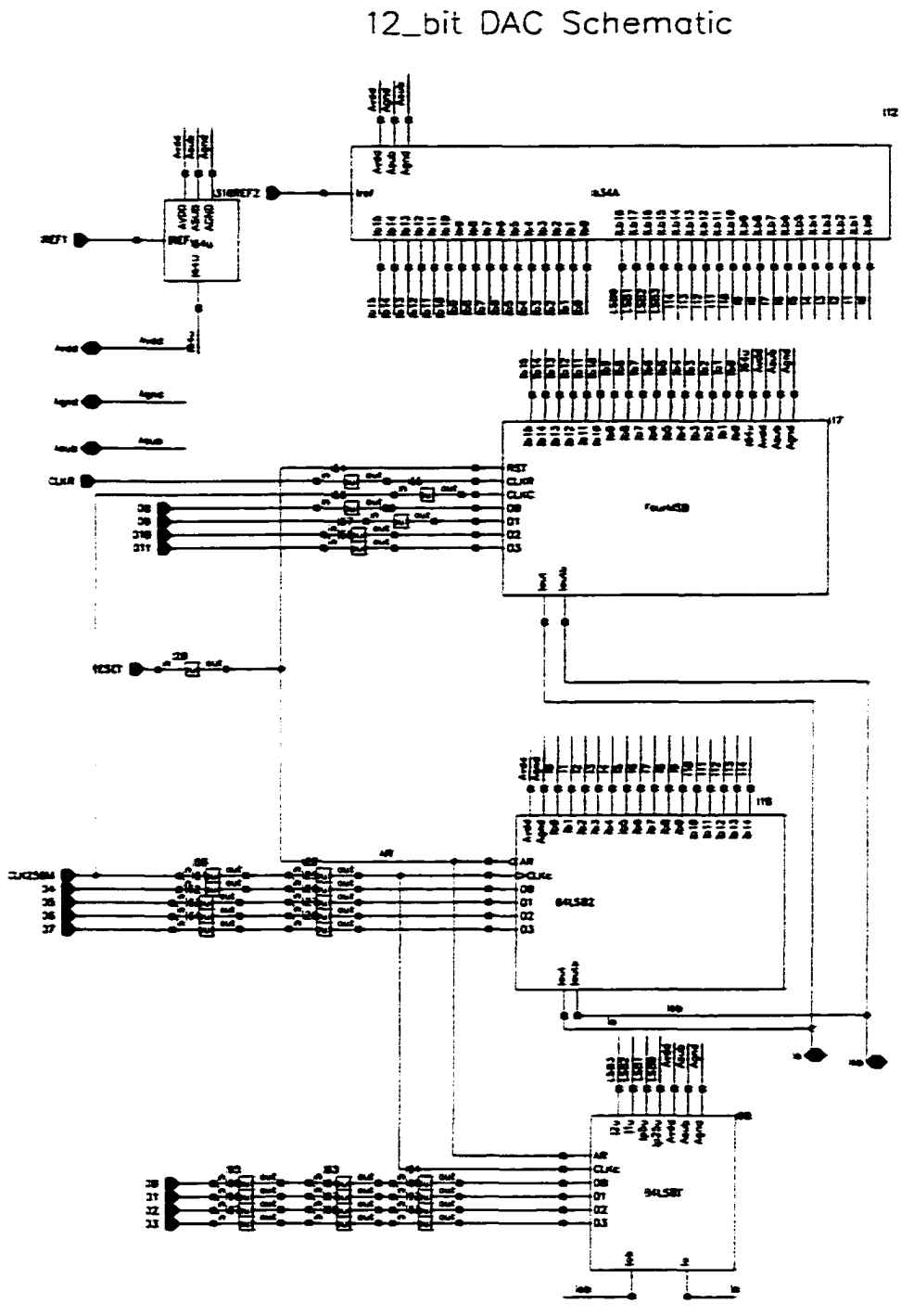


Figure 7.7 Schematic for a 12-bit DAC

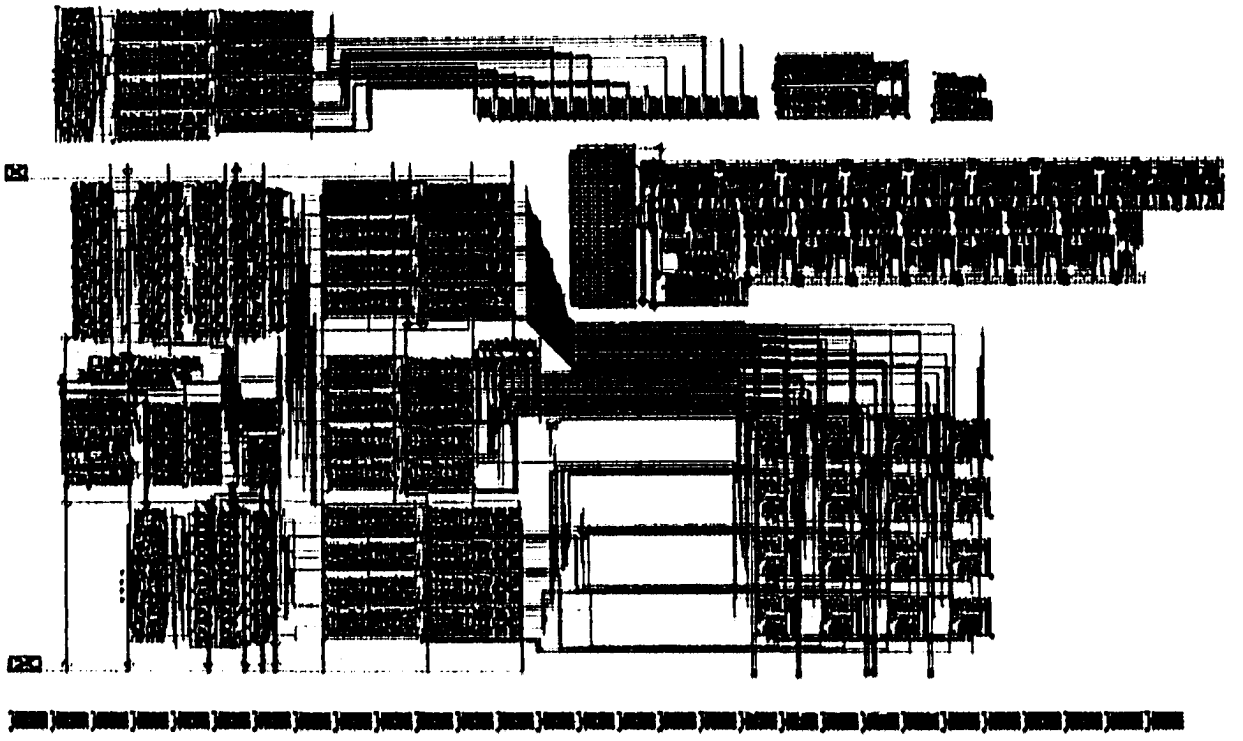

Figure 7.8 Synthesized layout for a 12-bit DAC

## 7.4 Summary

The chapter shows the direct mapping synthesis algorithm for modular DAC architectures and the implementation results. The contributions here are the organization of the library in the manner of direct mapping synthesis, development of the synthesis algorithm in the SKILL language in order to be integrated in CADENCE environments. By incorporating the SKILL codes with the synthesis library, it would be useful for users who want to obtain DAC core without knowing much detail of the DAC design. The synthesis library and the synthesis program can grow with more and more DAC prototypes.

# CHAPTER 8. CONCLUSIONS

## 8.1 Discussion of Results

The major goals of the dissertation include two parts. The first is to develop high-speed, high-resolution low-power, and small-area DACs, including self-calibration algorithms, that are appropriate for deep submicron digital CMOS technology. The other is to develop a synthesis tool for high-speed, high-resolution, low-power, and small-area DACs.

At first an extensive overview for high-speed and high-resolution DAC design is presented. It did not cover all the architectures due to the focus on Nyquist-rate, high-speed and high-resolution DACs. Based on the review of the different DAC architectures and their merits, a continuous self-calibrated current-steering architecture was used for the construction of high-speed and high-resolution DACs due to its insensitivity to process variation, matching and environmental changes. This dissertation extended the performance of the previous continuous self-calibration technique from audio frequency to conversion rates as high as hundreds of MS/s. The modular structures developed also help achieve low-power and small-area. Speed and accuracy limitation factors were carefully modelled. According to the model derivations, structural modifications to the self-calibrated current cell were made in order to achieve high-resolution and fast settling. The testing results of the prototype 8-bit 250MS/s DAC fabricated in TSMC's $0.25\mu$ single poly five metal logic CMOS process verified the success of the idea.

One of the major issues that arose is that glitches occur during the operation mode

transition of self-calibrated cells. Although glitch effects can be reduced by randomization, they impact the settling accuracy and dynamic performance in high-speed operation. This dissertation further proposes a redundant-cell-relay method (patent applied for in January 2001) to reduce these transition glitches. The glitch energy can be reduced by more than 10 times what previously published methods have demonstrated. The reduction of the glitches help to achieve the dynamic performance in high-speed and high-resolution DACs. A 10-bit DAC using the redundant-cell-relay with continuous self-calibration is simulated and implemented in TSMC's 0.25$\mu$ single poly five metal logic CMOS process. The testing results proved the method and very low-power high-speed high-resolution and small-area was achieved using the new relay method. A 12-bit DAC was also designed and simulated.

Furthermore, a direct mapping methodology and a synthesis tool were developed in this dissertation. The synthesis program is coded in CADENCE's SKILL language. The synthesis algorithm relys on the the DAC prototype building blocks, highlighted above, that were made part of the synthesis library. The tool developed takes user specifications from the CADENCE environment input form and then configures the corresponding DAC modules to, as closely as possible, create a DAC that meets the given specifications. The tool is also useful for users with little knowledge of the details of DAC design. With all these prototypes and algorithm developed, we have a silicon-proven DAC synthesizer. The tool can cover more and more mixed-signal blocks as long as the library is extended. The program is compatible with all versions of CADENCE. Design knowledge can be easily input and accumulated through the CADENCE tools.

## 8.2 Summary of Contributions

The contributions of this dissertation can be summarized in five parts. First, the dissertation has an extensive overview of DAC architectures and synthesis tools. This

review can serve as a reference for DAC design. The review laid a basis for deriving the theoretical equations and choosing modular architectures for the design and synthesis of high-speed high-resolution low-power self-calibrated DACs. Secondly, the dissertation extends the concept of continuous self-calibration from the audio frequency range to conversion rate of hundreds of MS/s. This is verified with a working 8-bit 250Ms/s prototype. High accuracy self-calibrated current cells which include charge injection compensation and clock feedthrough reduction techniques were proposed and implemented. Fast settling techniques were also included. The 8-bit prototype also clearly depicted the glitches that happen during the transition from the calibration mode to the supplying mode. The glitches clearly degrades the DAC accuracy as well as its SFDR. The third contribution is the invention of the redundant-cell-relay continuous self-calibration method for current-steering DACs. The method greatly reduces the glitch energy inherit in the previous self-calibration method. A patent application has been submitted based on the method [7]. The fourth contribution is the proof of the redundant-cell-relay method in silicon. A 200Ms/s 10-Bit DAC prototype was fabricated and tested. Its combination of high-speed, high-resolution, low-power and small area has great potential to many embedded applications. A paper [8] based on the prototype has been accepted to the 27th European Solid-State Circuits Conference to be held in September, 2001. The fifth contribution is the development of a DAC synthesis tool based on the above prototypes' modules and the direct mapping method published by the author of the dissertation in 1998 [9]. The synthesis tool is good for users with little knowledge of details of DAC design. This is useful for achieving fast time-to-market turnaround of critical DAC designs. In summary, this dissertation provides a package of methodology and tools for the design, implementation and synthesis of high-speed high-resolution low-power small-area DACs.

# APPENDIX. SYNTHESIS PROGRAM

This appendix includes the procedures written in the SKILL scripting language to synthesize the modular DACs. The procedure represents the flow shown in Chapter 7. Although the program is developed in CADENCE version 4.4.3, the SKILL language is compatible with any CADENCE version.

## Main Program

The main program is loaded from the command line of the ICFB window by issuing the following command: load("the_path_name\synmenu.il") The skill script for the synmenu.il is shown below. After loading the main program, the synthesis window shown in Figure 7.5 pops up. If you click the "Synthesis" menu and then DAC Synthesis, the input form for specifications shown in Figure 7.6 will be loaded.

```
trMenuItemOne = hiCreateMenuItem(
        ?name    'trMenuItemOne
        ?itemText "DAC Synthesis"
        ?callback "load("~ /SKILL/cell_design/labs/DACSYN3.il")"
        )
    trMenuItemTwo = hiCreateMenuItem(
        ?name 'trMenuItemTwo
        ?itemText "Quit"
```

```
    ?callback "ClearMemory()"
    )
  hiCreatePulldownMenu(
    'trPulldownMenu;;;the menu variable
    "synthesis";;;Menu title
    list(trMenuItemOne trMenuItemTwo)
    )
  hiInsertBannerMenu(window(1)trPulldownMenu0)
```

## Procedures

As shown in the above program, the synthesis will load a program DACSYN3.il from the path ~ /SKILL/cell_design/labs.Theprogramislistedhere.

```
procedure(StartSynthesis()
  let((NumberofBitsFullScaletheFormSymboltheForm)
    theFormSymbol = gensym('WindowForm)
  TechField = hiCreateStringField(
      ?name 'TechField
      ?prompt "Technology"
      ?value "TS25"
      ?defValue "TS25"
      ?editable t
      ?callback "println('TechField)"
      )
    NumberofBits = hiCreateScaleField(
      ?name 'NumberofBits
```

```
    ?prompt "NumberofBits"
    ?defValue 8
    ?value 8
    ?range list(4 12);;;4to12bit
    ?callback "println('NumberofBits)"
    )
FullScale = hiCreateScaleField(
    ?name 'FullScale
    ?prompt "FullScaleRange"
    ?value 1
    ?defValue 1
    ?range list(1 4);;;1 to 4mA
    ?callback "println('FuleScale)"
    )
theForm = hiCreateAppForm(
    ?name theFormSymbol
    ?formTitle sprintf(nil"%15s" "DACSynthesis")
    ?dontBlock t
    ?callback "trWindowFormCB(hiGetCurrentForm())"
    ?fields list(TechField NumberofBits FullScale)
    ?help ""
    ?unmapAfterCB t
    )
wid = hiOpenWindow(
    ?bBox list(100 : 100 1000 : 300);;;thisworksforopeningawindow
    ?type "form"
    ?appType "formdow"
```

```
            ?form theForm
            )
        theForm → wid = wid
        theForm;;;returntheForm
        );let
    );procedure
procedure(trWindowFormCB(theForm)
        let((wid Technology NumBits FullScale)
            wid = theForm → wid
            Technology = theForm → TechField → value
            NumBits = theForm → NumberofBits → value
            FullScale = theForm → FullScale → value
            printf("Technology : %sNumBits :
                %dFullScale : %dn.4\n" Technology NumBits FullScale )
            DacSynthesis(wid Technology NumBits FullScale)
            ;;; Call synthesis procedure according to the input values
        t
    ) ; let
    ) ; procedure
procedure( DacSynthesis(wid Tech NumBits FullScale)
    let( ( Libname )
        case( Tech (
            "TS25"
            Libname="dacsyn25"
            case( FullScale (
                1
                case( NumBits
```

```
(4
geOpen( ?lib Libname ?cell "FourMSB" ?view "schematic" ?mode "r")
geOpen( ?lib Libname ?cell "FourMSB" ?view "layout" ?mode "r")
)
(5
printf( "Sorry, 5 Bit is not supported now.\n" )
)
( 6
CreateInstFromSch( Libname "B6DAC" Libname "B6DAC_syn")
)
( 7
geOpen( ?lib Libname ?cell "B7DAC" ?view "schematic" ?mode "r")
printf(" Need some top block layout. \n" )
)
( 8
geOpen( ?lib Libname ?cell "B8DAC" ?view "schematic" ?mode "r")
geOpen( ?lib Libname ?cell "B8DAC" ?view "layout" ?mode "r")
)
( 9
printf( "Sorry, 9 Bit is not supported now.\n" )
)
( 10
CreateInstFromSch( Libname "B10DAC" Libname "B10DAC_syn")
)
( 11
printf(" Sorry 11 Bit not supported now. \n" )
)
```

```
        ( 12
        CreateInstFromSch( Libname "B12DAC" Libname "B12DAC_syn")
        )
        ) : case
    )
    (4
    case( NumBits
        ( 6
        geOpen( ?lib Libname ?cell "B6DAC4" ?view "schematic" ?mode "r")
        )
        ( 10
        geOpen( ?lib Libname ?cell "B10DAC4" ?view "schematic" ?mode "r")
        )
    ) : case
    )
    (2 printf("Sorry, Full scale current=2mA is not supported now.\n") )
    (3 printf("Sorry, Full scale current=3mA is not supported now.\n") )
) : case
) : TS25
( "TS18"
Libname="dacsyn18"
printf("Sorry, %s Libraries not supported now.\n" Libname )
) : TS18
( t
printf(" Sorry %s Technology not supported yet.\n" Tech )
) : Other Technologies
) : case
```

```
    ) ; let
    ) :procedure
procedure( CreateInstFromSch( Lib1Name Cell1Name Lib2Name Cell2Name)
    wid1=geOpen( ?lib Lib1Name ?cell Cell1Name ?view "schematic" ?mode "r")
    cv1=geGetWindowCellView(wid1)
    ;;; get current window

    wid2=geOpen( ?lib Lib2Name ?cell Cell2Name ?view "layout"
    ?viewType "maskLayout" ?mode "w")
    cv2=geGetWindowCellView(wid2)
    PosList=list(0  0)
    foreach( Inst cv1→ instances
        CellName = Inst → cellName
        LibName = Inst → libName
        printf("CellName : %sLibName : %s\n" CellName LibName)
        if( CellName=="ipin" || CellName=="opin" || CellName=="iopin"
            then
            printf(" Input or output or inout pin: %s \n" CellName)
            else
            PosList=LoadCellLayout( cv2 LibName CellName "layout" PosList)
        ) : if
    ) : foreach
    printf(" Congratulations! Layout cell %s in Library %s generated. \n"
        Cell2Name Lib2Name )
    dbSave(cv2)
    dbClose(cv2)
    ) : procedure
```

```
procedure( LoadCellLayout( cv libName cellName viewName position)
    let( (BBox1 ll ur xdif x)
        dinst1=dbCreateInstByMasterName( cv libName cellName viewName nil position "R0")
        BBox1=dinst1→ bBox
        ll = lowerLeft(BBox1)
        ur = upperRight(BBox1)
        xdif = xCoord(ur) − xCoord(ll) + 1
        x = xdif + car(position)
        NewPos = list(x 0)
        NewPos
    ); let
    ); procedure
procedure(ClearMemory()
    hiDeleteForm(theForm)
    theForm = nil
    hiDeleteBannerMenu(window(1) 0)
    wid = nil
    ); procedure
```

# BIBLIOGRAPHY

[1] R.J. Romanczyk and B.H. Leung, "BiCMOS circuits for high speed current mode D/A converters," *IEEE J. Solid-State Circuits*, Vol. 30 8 , Aug. 1995, pp. 923-934.

[2] Tak H. Ning, "CMOS performance and density trends as we approach 0.1 um," *Proceedings of 5th International Conference on Solid-State and Integrated Circuit Technology*, pp. 6–9, 1998.

[3] M.J.M Pelgrom, A.C.J. Duinmaijer et al., "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, Vol. 24, pp. 1433–1440, Oct. 1989.

[4] J.R. Naylor, "A complete high-speed voltage output 16-bit monolithic DAC," *IEEE J. Solid-State Circuits*, Vol. SC-18, pp. 729–735, Dec. 1983.

[5] H.S. Lee, D.A. Hodges, and P.R. Gray , "A self-calibrating 15b CMOS A/D converter," *IEEE J. Solid-State Circuits*, Vol. SC-19, pp. 813-819, Dec. 1984.

[6] D.W.J. Groeneveld, H.J. Schouwenaars, H.A.H. Termeer, and C.A.A. Bastiaansen, "A self-calibration technique for monolithic high resolution D/A converters," *IEEE J. Solid-State Circuits*, Vol. 24, No. 6, pp. 1517-1522, Dec. 1989.

[7] W. Zhang and M.M. Hassoun, "Apparatus for and method of performing a conversion operation". US patent application, Application No. 09/820,801, Reference ISURF #02718, 2001

[8] W. Zhang and M.M. Hassoun, "A redundant-cell-relay continuous self-calibration method for current-steering DACs", *IEEE 2001 ESSCIRC*, to be held in Sept. 2001.

[9] W. Zhang, H. Xia, R. Al-Omari, and M. Hassoun, "Symbolic synthesis of Analog-to-Digital conversion architectures using direct-mapping techniques," *1998 IEEE International Conference on Electronics, Circuits and Systems*, Vol. 3, pp. 215–218, Sept. 1998.

[10] H. Schmid, *Electronic analog/digital conversions*, Van Nostrand, Reinhold, New York, 1970.

[11] D.F. Hoeschele, *AD/DA Conversion Technique*, John Wiley, New York, 1968.

[12] D. Sheingold, *Analog/Digital Conversion Handbook*, Analog Devices Inc., Norwood, 1972.

[13] D.B. Bruck, *Data Conversion Handbook*, Hybrid System Corporation, Billerica, Massachusetts, 1972.

[14] G.C. Temes, and J.C. Candy, "A tutorial discussion of the oversampling method for A/D and D/A converters," *IEEE Symposium on Circuits and Systems*, Vol. 2, pp. 910-913.

[15] H.J. Schouwenaars, D.W. Groeneveld, C.A.A. Bastiaansen and H.A. Termeer, "An oversampled multibit CMOS D/A converter for digital audio with 115-dB dynamic range," *IEEE J. Solid-State Circuits*, Vol. 26, pp. 1775-1780, Dec. 1991.

[16] A. Paul and M. Sandler, "Issues for a 20-bit D/A converter based on pulse width modulation and noise shaping," *IEEE Colloquium on Advanced A-D and D-A Conversion Techniques and Applications*, 1993 , pp. 4/1–4/4.

[17] M. Corradetti and I. Olivia,"MOS A/D and D/A converter circuits based on the stochastic principle", *International Conference Mikroelektronik*, Munich, Nov 27-29, 1972.

[18] W.D. Mack, M. Horowitz and R.A. Blauschild, "A 14b PCM DAC", *ISSCC Digest of Technical Papers*, pp. 85-86,Feb. 1982.

[19] B. Loriferne, *Analog-digital and digital-analog conversion*, Heyden & Son Ltd, 1982.

[20] R.V.D. Plassche, *Integrated analog-to-digital and digital-to-analog converters*, Kluwer Academic Publishers, 1994.

[21] S.Y. Chin and C.Y. Wu, "A 10-b 125MHz CMOS digital-to-analog converter(DAC) with threshold-voltage compensated current sources", *IEEE J. Solid-State Circuits*, pp1374-1380, 1994.

[22] J. Bastos, M. Steyaert and W. Sansen, "A high yield 12-bit 250MS/s CMOS D/A converter". *IEEE 1996 CICC*, pp. 431-434, 1996.

[23] B.J. Tesch and J.C. Garcia, "A low glitch 14-bit 100 MHz D/A converter," *Proceedings of the 1996 Bipolar/BiCMOS Circuits and Technology Meeting*, pp. 204-207.

[24] D. Chang, J.G. Fossum, S.K. Reynolds, and M.M. Pelella, "Kink-free analog circuit design with floating-body NFD/SOI CMOS: a current-steering D-A converter," *1997 IEEE International SOI Conference*, pp. 158-159.

[25] J. Bastos, A.M. Marques, M.S.J. Steyaert, and W. Sansen, "A 12-bit intrinsic accuracy high-speed CMOS DAC". *IEEE J. Solid-State Circuits*, Vol. 33 12 , Dec. 1998 . pp. 1959-1969

[26] A. Van den Bosch, et al., "A 12 bit 200 MHz low glitch CMOS D/A converter," *Proceedings of the IEEE 1998 Custom Integrated Circuits Conference*, pp. 249 -252, 1998.

[27] C. Lin and K. Bult, "A 10-b 500-MSample/s CMOS DAC in 0.6 $mm^2$," *IEEE J. Solid-State Circuits*, Vol. 33, No. 12, December 1998, pp. 1948–1958.

[28] A. Marques, J. Bastos, M. Steyaert, and W. Sansen, "A current steering architecture for 12-bit high-speed D/A converters," *1998 IEEE International Conference on Electronics, Circuits and Systems*, Vol. 1, pp. 23-26, 1998.

[29] G.V. Vandenbussche, A. Van den Bosch, W. Daems, G. Gielen, M. Steyaert, and W. Sansen, "A 14b 150MSample/s update rate $Q^2$ random walk CMOS DAC", *IEEE 1999 ISSCC*, pp146-147, 1999.

[30] K. Khanoyan, F. Behbahani, and A.A. Abidi, "A 10 b, 400 MS/s glitch-free CMOS D/A converter," *1999 Symposium on VLSI Circuits*, Digest of Technical Papers, pp. 73-76

[31] A. Van den Bosch, M. Steyaert, and W. Sansen, "SFDR-bandwidth limitations for high speed high resolution current steering CMOS D/A converters," *The 6th IEEE International Conference on Electronics, Circuits and System*, Vol. 3, pp. 1193-1196, 1999.

[32] M. Tiilikainen, "A 1.8V 20mW 1$mm^2$ 14b 100MSample/s CMOS DAC," *2000 IEEE ESSCIRC*, Sept. 2000.

[33] A. Bosch, M. Borrenmans, M. Steyart, and W. Sanson, "A 10-bit 1-GSample/s Nyquist Current-Steering CMOS D/A Converter," *IEEE J. Solid-State Circuits*, Vol. 36, No. 3, March 2001, pp. 315-324.

[34] J.L. McCreary and P.R. Gray, "All-MOS Charge redistribution analog-to-digital conversion techniques," *IEEE J. Solid-State Circuits*, Vol. SC-10, pp. 371-379, Dec. 1975.

[35] P. Hollooway,"A trimless 16-bit digital potentiometer," *ISSCC Digest of Technical Papers*, pp. 66-67, February 1984.

[36] M.J.M. Pelgrom, "A 10-b 50-MHz CMOS D/A converter with 75-Ω buffer," *IEEE J. Solid-State Circuits*, Vol. 25, pp. 1347–1352, Dec. 1990.

[37] K. Maio, M. Hotta, N. Yokozawa, M. Nagata, K. Kaneko, and T. Iwasaki, "An untrimmed D/A converter with 14-bit resolusion," *IEEE J. Solid-State Circuits*, Vol. 16, pp. 616-621, Dec. 1981.

[38] R. Harjani, R.A. Rutenbar, and L.R. Carley, "Analog circuit synthesis and exploration in OASYS" *Proceedings of the 1988 IEEE International Conference on Computer Design: VLSI in Computers and Processors*, 1988. pp. 44–47.

[39] E.S. Ochotta, L.R. Carley, and R.A. Rutenbar, "Analog circuit synthesis for large, realistic cells: designing a pipelined A/D converter with ASTRX/OBLX" *Proceedings of the IEEE 1994 Custom Integrated Circuits Conference*, pp. 365 -368

[40] E.S. Ochotta, R.A. Rutenbar, and L.R. Carley, "Synthesis of high-performance analog circuits in ASTRX/OBLX" *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 15 3, March 1996 , pp. 273-294

[41] B.G. Song, I.K. Chang, O.J. Kwon, J.G. Park, and K.D. Kwack, "A simulation-based analog cell synthesis with improved simulation efficiency", *Proceedings of the IEEE Region 10 Conference TENCON 99*, Vol. 2, pp. 1018-1021, 1999.

[42] R. Phelps, M. Krasnicki, R.A. Rutenbar, L.R. Carley, and J.R.Hellums, "ANACONDA: robust synthesis of analog circuits via stochastic pattern search" *Proceedings of the IEEE 1999 Custom Integrated Circuits*, 1999 , pp. 567-570.

[43] M.C. McFarland, "The value trace: a data base for automated digital design", *Design Research Center, Carnegie-Mellon University, Report DRC-01-4-80*, Dec. 1978.

[44] D.W. Knapp and A.C. Parker, "A unified representation for design information," *7th International Symposium on Computer Hardware Description Languages and Their Applications*, pp. 337-353, Tokyo, August 1985.

[45] R.K. Brayton, N.L. Brenner, C.L. Chen, G. DeMicheli, C.T. McMullen, and R.H.J.M. Otten, "The Yorktown silicon compiler system," *Silicon Compilation*, Addison-Wesley, 1988.

[46] N.D. Dutt, T. Haddley, and D.D. Gajski, "An intermediate representation for behavioral synthesis," *Proceedings of 27th ACM/IEEE DAC*, pp. 14-19, Orlando, FL, June 1990.

[47] M. Monachino, "Design verification system for large scale LSI design," *IBM J. Res. Develop.* vol 26, No 1, January 1982.

[48] D.C. Ku, and G.De. Micheli. "Optimal synthesis of control logic from behavioral specifications," *J. VLSI Integration*, 1991.

[49] G. Gielen, and W. Sansan "Symbolic analysis for automated design of analog integrated synthesis", *Kluwer Academic Publisher*, 1991.

[50] M.M. Hassoun and J.E. McCarville, *Symbolic analysis of large-scale networks using a hierarchical signal flow graph approach in analog integrated circuits and signal processing*, Kluwer Academic Publishers, 1993.

[51] N.C. Horta and J.E. Franca,"Algorithm-driven synthesis of data conversion architectures," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 16, No. 10, Oct. 1997.

[52] W.P. Birmingham, A.P. Gupta, and D.P. Siewiorek, "The MICON system for computer design." In Proc of 26th Design Automation Conference. *IEEE Computer Society and ACM-SIGA*, 1989.

[53] D.A. Johns, K. Martin, *Analog Integrated Circuit Design*, John Wiley & Son, Inc, New York, 1997.

[54] E. Sackinger and W. Guggenbuhl, "A high swing, high-impedance MOS cascode circuit". *IEEE J. Solid-State Circuits*, Vol. 25, No. 1.pp. 289-298, Feb. 1990.

[55] H. Kohno, et al., "A 350-MS/s 3.3-V 8-bit CMOS D/A converter using a delayed driving scheme," *IEEE 1995 CICC*, pp. 211-214.

[56] N.V. Bavel, "A 325 MHz 3.3V 10-bit CMOS D/A converter core with novel latching driver circuit." *IEEE 1998 CICC*, pp245-248.

[57] S. Boiocchi, S. Brigati, G. Caiulo, and F. Maloberti, "Self-calibration in high speed current steering CMOS D/A converters," *Second International Conference on Advanced A-D and D-A Conversion Techniques and Their Applications*, pp. 148-152, 1994.

[58] G. WEGMANN, E.A. VITTOZ, and F. RAHALI, "Charge injection in analog MOS switches", *IEEE J. Solid-State Circuits*, Vol. SC-22, No. 6, pp. 1091-1097, Dec. 1987.