2008

Fully digital-compatible built-in self-test solutions to linearity testing of embedded mixed-signal functions

Hanqing Xing
Iowa State University

Fully digital-compatible built-in self-test solutions to linearity testing of embedded mixed-signal functions

by

Hanqing Xing

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Electrical Engineering

Program of Study Committee:
Degang Chen, Major Professor
Randall L. Geiger
Chris Chong-Nuen Chu
Aleksandar Dogandzic
Stephen B. Vardeman

Iowa State University

Ames, Iowa

2008

Table of Contents

List of Figures

List of Tables

Abstract

Mixed-signal circuits, especially analog-to-digital and digital-to-analog converters, are the most widely used circuitry in electronic systems. In the most of the cases, mixed-signal circuits form the interface between the analog and digital worlds and enable the processing and recovering of the real-world information. Performance of mixed-signal circuits, such as linearity and noise, are then critical to any applications. Conventionally, mixed-signal circuits are tested by mixed-signal automatic test equipment (ATE). However, along with the continuous performance improvement, using conventionally methods increases test costs significantly since it takes much more time to test high-performance parts than low-performance ones and mixed-signal ATE testers could be extremely expensive depending on the test precision they provide. Another factor that makes mixed-signal testing more and more challenging is the advance of the integration level. In the popular system-on-chip applications, mixed-signal circuits are deeply embedded in the systems. With less observability and accessibility, conventionally external test methods can not guarantee the precision of the source signals and evaluations. Test performance is then degraded.

This work investigates new methods using digital testers incorporated with on-chip, built-in self-test circuits to test the linearity performance of data converters with less test cost and better test performance. Digital testers are cheap to use since they only offer logic signals with direct connections. The analog sourcing and evaluation capabilities have to be absorbed by the on-chip BIST circuits, which, meanwhile, could benefit the test performance with access to the internal circuit nodes. The main challenge of the digital-compatible BIST methods is to implement the BIST circuits with enough high test performance but with low

design complexity and cost. High-resolution data converter testing needs much higher-precision analog source signals and evaluation circuits. However, high-precision analog circuits are conventionally hard to design and costly, and their performance is subject to mismatch errors and process variations and cannot be guaranteed without careful testing. On the digital side, BIST circuits usually conduct procedure control and data processing. To make the BIST solution more universal, the control and processing performed by the digital BIST circuits should be simple and not rely on any complex microcontroller and DSP block. Therefore, the major tasks of this dissertation are 1) performance-robust analog BIST circuit design and 2) test procedure development. Analog BIST circuits in this work consist of only low-accuracy analog components, which are usually easy to design and cost effective. The precision is then obtained by applying the so-called deterministic dynamic element matching technique to the low-accuracy analog cells. The test procedure and data processing designed for the BIST system are simple and can be implemented by small logic circuits.

In this dissertation, we discuss the proposed BIST solutions to ADC and DAC linearity testing in chapter 3 and chapter 5, respectively. In each case, the structure of the test system, the test procedure, and the theoretical analysis of the test performance are presented. Simulation results are shown to verify the efficacy of the methods. The ADC BIST system is also verified experimentally. In addition, chapter 4 introduces a system-identification based reduced-code testing method for pipeline ADCs. This method is able to reduce test time by more than 95%. And it is compatible with the proposed BIST method discussed in chapter 3.

Chapter 1. Introduction

Along with the advance of the integration level and in order to meet the demands of the new applications, integrated circuits incorporating both digital and analog functions have become increasingly prevalent in the semiconductor industry.

VLSI technology scaling is driven by Moore's law, which states that the number of transistors on a given chip will double every two years [1]. For example, today's 65nm Pentium D CPU carries 376 millions transistors, which is about 50 times as many as the Pentium II in 350nm process 10 years ago. Meantime, working 45nm CPU products using the new technology have been manufactured with doubled transistor density.

The continuous advance of the integration level enables explosive growth of new applications in wireless communications, digital signal processing, multimedia, instrumentation, and so on. Portable devices are also becoming more and more popular and indispensable in our daily lives. As a result, the trend of integrated circuits moves towards adding more functionality, such as analog and mixed-signal circuits, microprocessor, memory, I/O and RF, to a single silicon chip [2]. The so-called system-on-a-chip (SOC) and system-in-package (SIP) techniques are investigated for system implementation with higher integrated level and lower cost. However, this trend causes challenges not only in design, integration, and packaging, but also in testing.

1.1 Mixed-Signal Circuits and Testing

Analog circuits are those circuits processing continuously variable analog signals, including operational amplifiers, active, or passive filters, comparators, voltage regulators, analog mixers, and voltage or current references, and so on. In contrast, in digital circuits,

such as microprocessors and logic blocks, signals usually take only two different levels. A mixed-signal circuit can be defined as a circuit consisting of both digital and analog elements. One type of the most common mixed-signal devices are analog-to-digital converters (ADCs) and digital-to-analog converters (DACs). An ADC is a circuit that samples a continuous analog signal at specific points in time and converts the sampled voltages (or currents) into a digital representation. Conversely, a DAC is a circuit that converts finite-precision numbers into an analog voltage (or current). ADCs and DACs are the most common mixed-signal components in complex mixed-signal systems, since they form the interface between the physical world and the world of digital logic. Other common mixed-signal circuits include the phase locked loop, the programmable gain amplifier, etc.

Mixed-signal circuits usually incorporate different analog blocks. As we know, the real-world IC fabrication process is subject to imperfections. Unfortunately, analog circuits are often extremely sensitive to process imperfections, which can cause catastrophic failure or variations in performance of the circuits. As the scale of process shrinks and the speed increases, performance sensitivities become more exaggerated. On the other hand, different applications have different performance requirements. For example, DACs in audio applications require a high dynamic range with little distortion, while for video systems the DAC linearity is the crucial parameter to ensure a good picture quality. Hence, to select the right mixed-signal circuits for a particular application, the identification of the performance parameters of the AMS circuits is absolutely necessary. Consequently, analog and mixed-signal circuits are often tested exhaustively to guard performance variations due to the imperfections.

AMS testing is becoming increasingly challenging for two reasons. The first is the increasing integration level. In AMS testing, analog source signals need to be fed into circuits under test and the responses of the circuit are traced out for measuring. Both in and out signals need to be of sufficient accuracy for accurate testing. Keeping the accuracy is a challenging job, especially when AMS circuits are more and more deeply embedded with other core semiconductor technologies, such as logic, memory, I/O, and RF. Furthermore, circuit nodes cannot be easily accessed because of the limited number of available pins for testing. The consequence is the loss of the accessibility and observability of the nodes in AMS circuits and, consequently, the signal integrity of the analog stimuli and outputs.

The second reason AMS testing is becoming more difficult is that the performances of AMS circuits keep improving to satisfy the demands of new applications. For example, data converters push performance towards higher resolution, speed, and dynamic range, but with lower power dissipation. The latest announced high-speed and high-resolution analog to digital converters have up to 16 bits resolution and more than 100M sample/second sampling rate. These state-of-the-art products are already close to the performance limitation of process. Thus, testing of those components is exceptionally challenging since source signals or digitizers used for testing need to be much more accurate than the devices under test (DUT). The ubiquitous belief in the IC test community is that to test ADCs, stimulus accuracy must be at least 10 times or 3 bits better than that of ADCs under test. Therefore, to test a 16-bit ADC, the stimuli need about 19-bit accuracy, which is very difficult to achieve even with extremely expensive analog and mixed-signal automated test equipments (ATEs). In addition, the speed of testing is normally very slow at high accuracy levels.

As a result of the challenges mentioned above, test costs keep increasing. Most conventional test methods involve expensive high-performance mixed-signal ATEs. Meanwhile, test time tends to increase since more circuits are integrated on-chip and under test. The difficulties in testing also increase the time to market and reduce the profit margin of the products. Thus, a lot of efforts have been dedicated to developing cost-saving testing solutions with equivalent testing accuracy.

AMS circuits can be tested in different ways to achieve satisfactory test performance with low test cost, and almost all the solutions incorporate automated test equipment (ATE) and device interface boards (DIB). There are several possible solutions for verifying AMS blocks. The first is designing a specific testing system for particular devices under test. In this case, the testing system includes a specific interface for connection, high-performance analog blocks, stimulus source generation and analog signal acquisition, memory for data storage, and a digital part for digital acquisition and analysis. This approach saves the cost of using automated test equipment. However, designing a testing system for high-performance DUTs is not a trivial task. The system has to be good enough to make sure that the test failure is from the DUT but not the testing system itself. The difficulties in designing those systems could increase the time to market of the product and, consequently, reduce profit.

The second solution, and also the most common way, is using a general-purpose analog and mixed-signal ATE tester, like Teradyne A575. The tester needs to have much better performance than the circuits under test (CUT) in accuracy, speed, noise, and so on. In fact, for high-resolution testing, the eligible AMS ATE will be prohibitively expensive because of its extremely high performance. A device interface board (DIB) should be carefully designed for each type of DUT according to their unique electrical and mechanical

testing requirements, but DIBs in AMS testing are much more complicated than those in digital testing. With a good DIB design, the testing cost is determined mainly by the AMS ATE, which is selected to provide necessary testing performance for DUTs. For that reason, testing high-performance DUTs is usually costly and the testing time for each DUT has to be as short as possible. With the CUTs deeply embedded in the system, this method has difficulties in maintaining signal integrity due to the loss of the node accessibility and observability. As a result, test accuracy is reduced.

The third solution is making use of a digital ATE tester with a complex DIB [3], and to keep costs low, using cheap and low performance testers is preferred. By adding more functionality on the DIB, requirements on the tester are dramatically reduced and existing low-cost digital testers can be reused for analog and mixed-signal testing. Cost savings produced by the tester are substantial. However, all the analog processing blocks are integrated in the DIBs. Designing such a DIB is quite elaborate and time consuming. More importantly, signal integrity still remains an issue.

The fourth possible solution for AMS testing is using a digital tester with a built-in self-test (BIST) technique. The BIST technique is also a possible solution to the signal integrity problem, which offers the on-chip stimulus and response verification capabilities for testing by adding some functionality circuits on-chip with the CUT. Therefore, no analog signal needs to be applied or processed off-chip, and only digital testers and digital DIBs are necessary for testing. Digital DIBs are used to simply provide point-to-point connectivity between the DUT pins and the tester, so that the testing cost can be further reduced. In addition, BIST could reduce the number of connections required for testing. Therefore, more parts could be tested in parallel to reduce total testing time.

1.2 DFT and BIST

As the integration level increases and products become more complex, applying conventional external testing methods with satisfactory testing performance and cost becomes more difficult. Therefore, people have been investigating how to add certain testability features to circuit design: “Any design methodology or circuit that results in a more easily or thoroughly testable product can be categorized as design-for-test (DFT). DFT, when properly implemented, can offer lower production costs and higher product quality [4].” Built-in self-test (BIST) is a type of DFT and is probably the ultimate solution to testing integrated circuits. BIST circuits provide the stimulus and response verification capabilities for testing on-chip and allow the DUT to evaluate its own quality without elaborate automated test equipment support. In this dissertation, we focus on built-in self-test techniques of analog and mixed-signal circuits.

One of the advantages of using BIST is reduced testing costs. BIST techniques could enable much faster testing and simpler test procedures, and shorter testing time directly reduces testing costs. Moreover, BIST can reduce testing costs by reducing the requirements of the ATE tester. A high-performance AMS ATE tester is generally far more expensive than one that is only capable of doing low-performance tests. The critical performance specifications of ATE testers include operation frequency, measuring accuracy, and the number of channels, etc. Therefore, if the IC design engineer can find a way to test high-frequency signals using low-frequency stimulus and measurement hardware, to test high-accuracy analog signals using low-accuracy circuits, or to reduce digital channel count, then the test cost can be reduced significantly.

Another important advantage of using BIST is better testing performance. As we have mentioned, BIST techniques could improve the signal integrity. With source signals generated or evaluation circuits built on-chip close to the circuits under test, the signal integrity of analog signals can be greatly improved compared to conventional external test methods. The circuits under test can then be tested in a much more clean testing environment with less external interference. Signals that need to be externally sent into or traced out of the circuits can be either digital control signals or analog signals with minimal accuracy requirements. Therefore, testing performance is degraded by external interference. But, the BIST technique reduces harmful external interference.

Unlike digital BIST techniques, which have been extensively applied and standardized, "analog/mixed-signal BIST techniques are lagging. No proven alternative to performance-based analog testing exists and more research in this area is needed [2]." The reason analog and mixed-signal BIST techniques have lagged is, although BIST simplifies the test setup, it increases the circuit complexity and silicon area. Therefore, to provide satisfactory testing performance for high-resolution AMS circuits, BIST circuits tend to be very complicated and large. Furthermore, conventional high-performance analog circuits are subject to mismatch and process variation errors. The use of on-chip stimulus and verification circuits throws doubt to the accuracy of measurements, since there is a question about the quality of the signals generated and measured on a given circuit under test. Therefore, BIST circuits have to be tested and calibrated. The efforts of testing the BIST circuits often make the entire prospect of using BIST unrealistic. As a result, practical BIST circuits should have high performance but be small, easy to design, and insensitive to mismatch and process variation.

1.3 Dissertation Organization

This dissertation investigates BIST solutions to linearity testing of embedded data converters using low-cost, low-accuracy on-chip BIST circuits. It comprises six chapters. The rest of the dissertation is organized as follows.

In chapter 2, ADC and DAC static linearity parameters and the normally used testing methods are introduced. The concept and challenges of BIST are discussed. Two examples of data converter BIST applications are presented and the drawbacks are detailed.

Chapter 3 introduces the proposed BIST solution to ADC linearity testing. The structure of the BIST system, the test procedure, and the theoretical analysis of the test performance are presented. Design of the prototype chip is discussed. Simulation and experimental results are shown for verification.

In chapter 4, a new system-identification-based, reduced-code testing method for pipeline ADCs' static linearity is investigated. The algorithm is described and theoretically analyzed. Simulation and experimental results are show to validate the efficacy of the method.

In chapter 5, a BIST solution to high-resolution DACs' linearity testing is presented. Similar to chapter 3, the structure of the BIST system and the theoretical analysis of the test performance are included. The effects of the circuit non-idealities are discussed and the solutions to remove those effects are proposed. Simulation results are listed to support the method.

Chapter 6 concludes the dissertation.

Chapter 2. Data Converter Linearity Test

As the interface between the analog and digital signals, analog-to-digital converters (ADC) and digital-to-analog converters (DAC) are widely used in a variety of applications. Figure 2.1 shows a block diagram of a DSP system as an example. An ADC quantizes the received analog signal and generates the digital outputs for later digital signal processing. A DAC converts the digital outputs after processing into analog voltage levels to reconstruct the continuous analog output signal. The performance of ADCs and DACs, then, directly determines the accuracy and speed of the whole system. In particular, sufficient linearity performance of ADCs is critically important to many applications in signal processing, communications, instrumentation, and other areas. Therefore, accurate testing of linearity is indispensable for almost all ADCs to validate the design and to reduce rejected parts

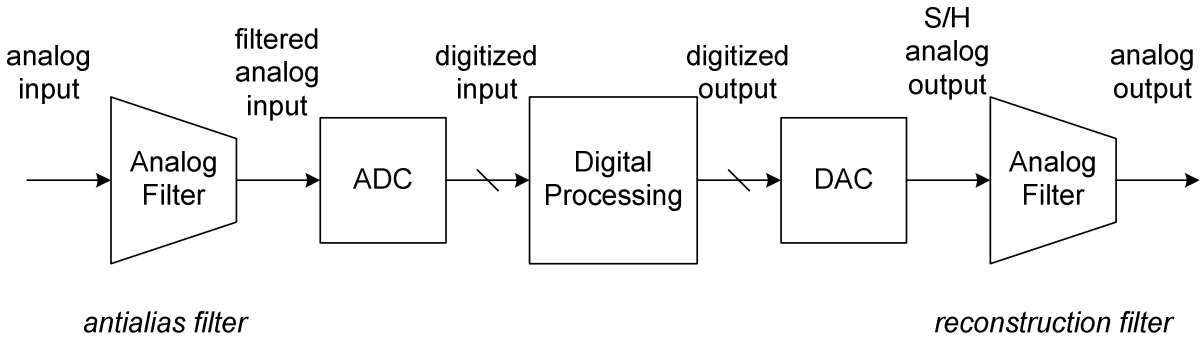


Figure 2.1. Block diagram of a DSP system

Data converters' performance can be represented by their static specifications, such as offset, gain error, integral nonlinearity error (*INL*), differential nonlinearity error (*DNL*) and equivalent RMS input noise, and dynamic specifications, such as signal-to-noise ratio (SNR), total harmonic distortion (THD), and spurious-free dynamic range (SFDR). Static linearity of data converters is mainly characterized by *INL* and *DNL*. In this chapter, we first introduce

the definitions of these two parameters for ADCs and DACs, respectively, and then briefly discuss the common methods of *INL* and *DNL* testing in each case.

2.1 ADC Linearity Test

2.1.1 Static Linearity

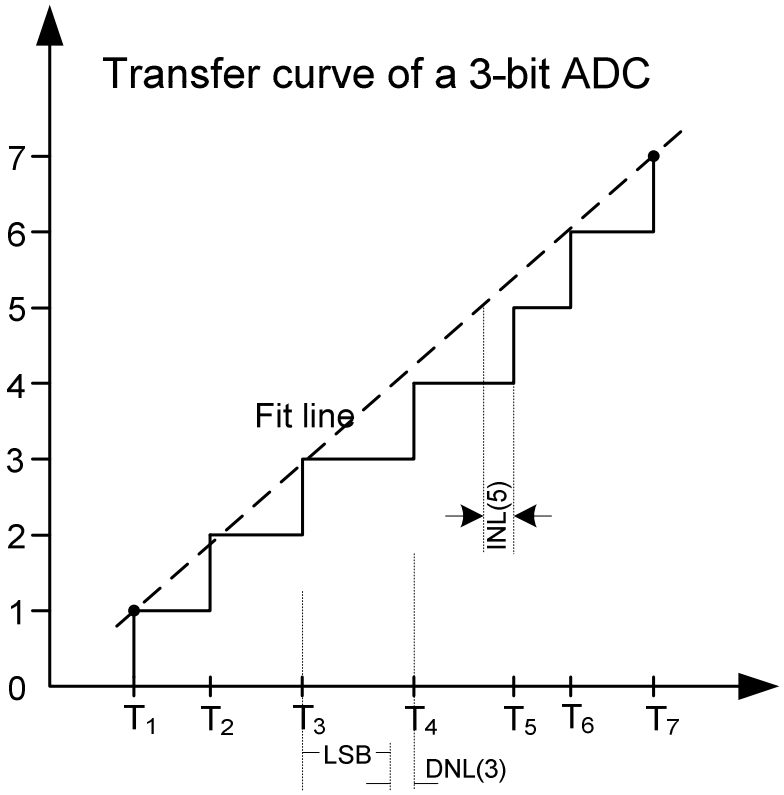


Figure 2.2. Transfer curve of an ADC

ADCs usually sample and quantize continuous analog input signals and generate outputs with discrete values at discrete times. The transfer curves of ADCs are staircase-like as shown in Figure 2.2. The *x* axis represents the input voltage, and the *y* axis shows the discrete digital outputs. The dotted line illustrates the transfer curve of an ideal 3-bit ADC. The solid line shows an example of a non-ideal ADC. The most important information in the transfer curve is the code transition levels, $T(1)$, $T(2)$, …, $T(2^n\text{-}1)$, where n is the resolution of

the ADC in bits. Quantitatively, a code transition level is the value of the converter input parameter, which causes half of the digital output codes to be greater than or equal to and half less than a given output code [5]. The uncertainty in the transition levels is because of the random effects in the ADC quantization procedure, such as noise, clock jitter, etc.

Static errors usually result from non-ideal spacing of the code transition levels. For example, offset and gain errors can be determined from the code transfer curve easily by looking at the deviations of the first and last transition levels of the actual curve from the ideal ones (the terminal-based definitions for gain and offset are adopted [5]). In most applications, these two errors do not affect the linearity performance of the system and can be simply compensated. The more important static specifications are the integral nonlinearity (*INL*) and the differential nonlinearity (*DNL*).

The integral nonlinearity is defined as the maximum difference between the ideal and actual code transition levels after correcting for gain and offset, divided by the average code-bin width. The differential nonlinearity is defined as the difference between a specified code-bin width and the average code-bin width, divided by the average code-bin width [5]. It is noted that the definitions of *INL* and *DNL* exclude the impact of offset and gain error. One simple way of correction for gain and offset is to define the "ideal" transfer curve using the end-point fit line of the actual curve. Thus, the average code-bin width, which is also the least significant bit (*LSB*), is calculated by

$$LSB = \frac{T(N-1) - T(1)}{N-2}, \tag{2.1}$$

where N is the total number of ADC digital output codes and equal to 2^n, and $T(1)$ and $T(N\text{-}1)$ are the first and last transition levels of the actual transition curve. Then, the ideal transition level at code k, $T_{id}(k)$, k = 1, 2, …, N-1, is calculated as in

$$T_{id}(k) = T(1) + (k-1)\cdot LSB. \tag{2.2}$$

The integral nonlinearity and differential nonlinearity at code k in *LSB*s, $INL(k)$ and $DNL(k)$ respectively, can be calculated by the follow equations:

$$INL(k) = \frac{T(k) - T_{id}(k)}{LSB} = \frac{T(k) - T(1)}{LSB} - (k-1) \text{ and} \tag{2.3}$$

$$DNL(k) = \frac{T(k+1) - T(k)}{LSB} - 1. \tag{2.4}$$

For the differential nonlinearity, $DNL(N\text{-}1)$ is not defined since there is no more transition levels for extra codes and the code-bin width for output code N-1 is unbounded. The *INL* and *DNL* of the ADC are the maximum values of the magnitudes of $INL(k)s$ and $DNL(k)s$, respectively.

From the equations, we can clearly see that ADCs' linearity characteristics are calculated from ADCs' transition levels. Thus, the major task of testing is to accurately estimate each transition level of ADCs.

2.1.2 Linearity Test

ADC linearity testing usually requires analog source signals to the input of ADC under test. The digital outputs are then monitored or post processed by some digital circuits to locate the transition levels. There are two test methods in common use: the histogram and the feedback loop.

2.1.2.1 Histogram test

The basic block diagram of the histogram test is illustrated in Figure 2.3. In this approach, a histogram of code occurrences is generated in response to input signals with known distributions over ADC input range. After a sufficiently large number of samples, the distribution of the output histogram should approach the known distribution of the input signals if an ideal ADC is under test. Then, for actual ADCs under test, the differences between the distributions of the histogram and the input signals can be used to compute the linearity performance of the ADCs. As shown in Figure 2.3, the source generator is applied to the device under test, the data generated by the excitation is collected and analyzed, and the analysis generates the code transition levels. From these levels, the key ADC parameters, *INL*, *DNL*, and gain and offset can be calculated.

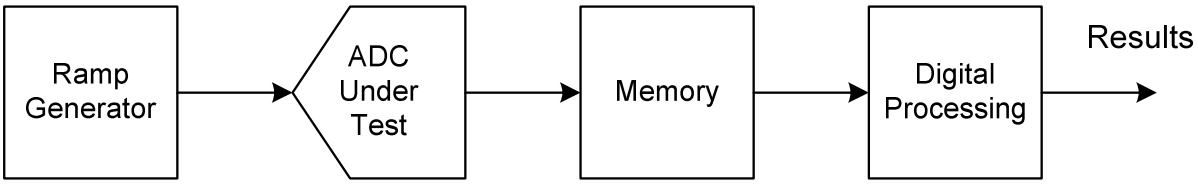


Figure 2.3. Diagram of ADC histogram test topology

A histogram test can either be done with ramp or sine-wave excitation. Ramp (or equivalent triangle wave) histogram testing is significantly faster than sine-wave histogram testing because a much smaller number of samples are measured in ramp histogram testing for a desired accuracy. Ramp histogram testing requires a very accurate and linear ramp to function correctly. If the ramp has uneven quantization, or non-linearity, they are translated into equivalent errors in the measured ADC transfer function. For example, the testing of a 16-bit converter to an accuracy of 1/8 of an LSB requires a ramp with 19 bits of resolution

and linearity. The input ramp is usually generated by a high-resolution DAC or arbitrary wave from a generator with suitable linearity.

The location of the code transition for code *k*, *T*(*k*), can be extracted by manipulating the data that is collected in a histogram test with a ramp input. The code transitions are given by

$$T(k) = C + A \cdot H_C(k-1) \ \ for \ k = 1, 2, ..., N-1 \tag{2.5}$$

where *A* and *C* are a gain factor and an offset factor, respectively, and $H_C(j)$ is the cumulated sum of the collected histogram up to code *j* and equal to $\sum_{i=0}^{i=j} H(i)$. *H*(*i*) is the number of histogram samples received in code bin *i*. The values of *C* and *A* will not affect the calculation of the nonlinearity based on the definitions. Therefore, the *INL*(*k*) and *DNL*(*k*) of ADC under test in *LSB* can be expressed by the histogram data as in

$$INL(k) = \frac{H_C(k) - H(0)}{H_C(N-2) - H(0)} \times (N-2) - k\,, \tag{2.6}$$

$$DNL(k) = \frac{H(k)}{H_C(N-2) - H(0)} \times (N-2) - 1\,. \tag{2.7}$$

The total number of histogram samples measured is very important to the precision of the measured values of the code transition levels. Increasing the number of samples decreases the uncertainty in the measurements and the effects of noise.

Another widely used excitation in histogram test is a sine wave, which is usually easier to obtain than a ramp signal. During the test, a pure sine wave of amplitude sufficient to slightly overdrive the ADC is input to the ADC under test. The frequency of the sine wave and the record length of the data collected must be carefully selected based on the sampling

frequency of the ADC to achieve coherent sampling. The DC offset of the sine wave is set to be equal to the center of the ADC full-scale input range. The transition levels can then be calculated from the histogram data and the known distribution of the sine-wave samples. The detailed equations and analysis are shown in [5]. Since the distribution of the sine-wave samples are not uniform, the calculation is a little more complex than with ramp signals. Also, to make sure there are enough number samples at the center of the input range, where the density of samples is lower, a larger number of samples are needed for testing than with ramp signals.

2.1.2.2 Feedback loop test

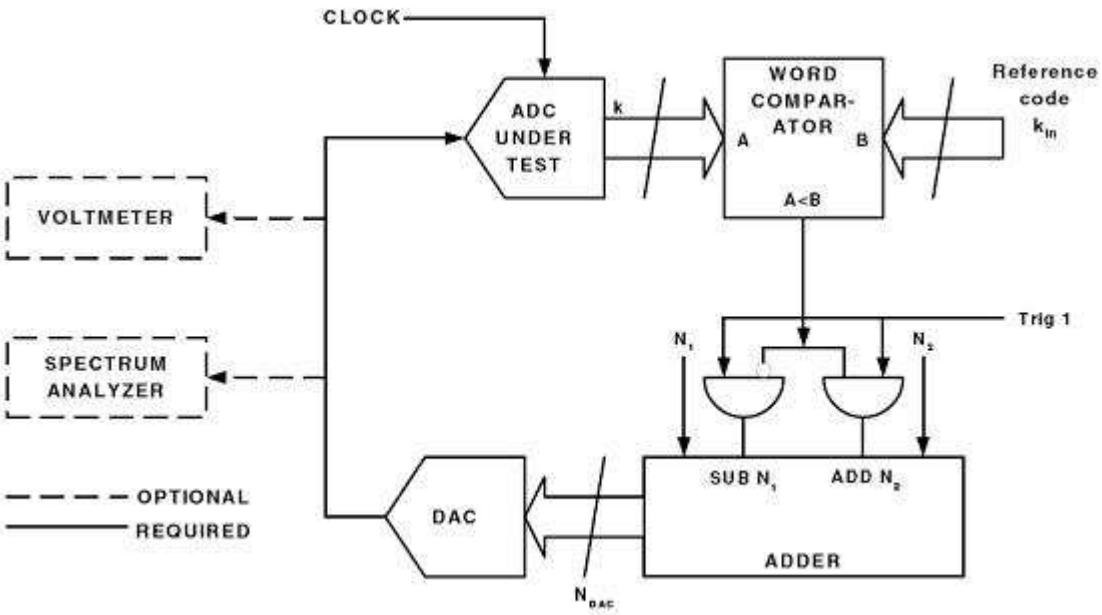


Figure 2.4. Block diagram of the feedback loop method

The block diagram of the feedback loop test method is depicted in Figure 2.4. In this approach, an input is applied to the ADC under test and the ADC outputs are compared to a preset value *k*, which specifies the code transition level *T*(*k*) for testing. If the ADC output is

below the preset value, the input is raised by a certain amount. If the ADC output is equal to or above the desired value, the input is reduced by a certain amount. This process is repeated until the ADC input has settled to a stable average value [5]. After the loop has settled, the input is regarded to be equal to the code transition level $T(k)$. Its value can then be either measured by a high-precision digitizer or computed from the known transfer function of the input source.

The analog input to the ADC can be generated by either a high-resolution DAC or an analog integrator. When using a DAC as source generator, the resolution of the DAC should be higher than that of the ADC under test because the fixed amount of the level adjustment is smaller than one LSB. The value of the adjustment is dependent on the noise level of the ADC. Detailed analysis can be found in [5].

2.2 DAC Linearity Test

2.2.1 Static Linearity

Similar to ADC static linearity, a DAC's static linearity is mainly measured by its integral nonlinearity and differential nonlinearity. Figure 2.5 presents an example of DAC transfer characteristics. The *x* axis represents the digital inputs, and the *y* axis shows the analog output levels. Different from the many-to-one transfer characteristics of an ADC, DACs' transfer curve is a one-to-one mapping function. The cycles illustrate the ideal transfer curve. The dots show an example of an actual DAC.

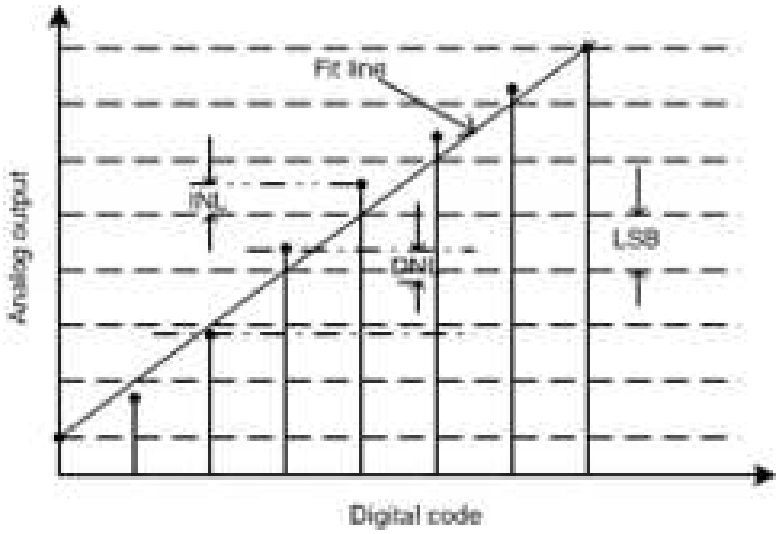


Figure 2.5. DAC transfer function

The static specifications of an actual DAC can be determined by comparing its transfer curve with the ideal one. Several different calculation methods for DACs' static specifications exist: best-fit, end-point, and absolute [6] [7]. In this dissertation, the end-point method is applied. In this case, the gain and the *LSB* are directly calculated from the slope of the end-point fit line of the actual transfer curve. The differential nonlinearity is then defined as the difference, after correcting for static gain, between a specified code and the next code, divided by the ideal code-bin width. *DNL* is expressed in *LSB*. The integral nonlinearity is the different between the ideal and measured code transition levels after correcting for static gain and offset. Integral nonlinearity is usually expressed as a percentage of full scale of in units of *LSB*s. When the differential nonlinearity or integral nonlinearity is given as one number without code specification, it is the maximum differential nonlinearity or integral nonlinearity of the entire range.

The *LSB* calculated by the end-point fit line is

$$LSB = \frac{V(N-1) - V(0)}{N-1} \tag{2.8}$$

where $V(0)$ and $V(N\text{-}1)$ are the DAC output levels at terminals, N is the total number of DAC output levels and equal to 2^n-1, and n is the resolution of the DAC in bits. Then, the integral nonlinearity and differential nonlinearity at input code k, $INL(k)$ and $DNL(k)$ k = 0, 1, 2, …, N-1, in LSB can be expressed by

$$INL(k) = \frac{V(k) - V(0)}{LSB} - k \tag{2.9}$$

$$DNL(k) = \frac{V(k+1) - V(k)}{LSB} - 1. \tag{2.10}$$

For the differential nonlinearity, $DNL(N\text{-}1)$ is not defined.

2.2.2 Linearity test

The equations make clear that in order to characterize the static linearity of a DAC, its output levels need to be accurately measured. The straightforward way of measuring DACs' outputs is using a high-precision digitizer. The resolution and the accuracy of the digitizer should be much higher than the DAC under test depending on the desired accuracy level of test.

2.3 BIST of Static Linearity

As described in chapter 1, a built-in self-test has been long proposed as a testing solution that lowers costs and improves testing accuracy. Although digital BIST has been widely used in general, its use in analog and mixed-signal BIST is very limited. AMS BIST techniques are mostly ad-hoc and hardly standardized. The available applications of AMS

BIST are typically confined to gross functional testing, and no widely accepted specification-based AMS BIST methods have been developed. In this section, I talk about the challenges of data converter linearity BIST and give several previous implementations as examples.

2.3.1 Challenges of Data Converter Linearity BIST

Conventional linearity testing methods requires either high-performance source signals or high-precision digitizers for accurate measurements. Moreover, the test procedure may not be easily applied to BIST because of either the volume of data storage or the complexity of processing.

For ADC linearity testing, the most ubiquitous approach is the histogram method, which provides an effective way of full-code testing. However, the accuracy of the test strongly depends on the distribution of the input stimuli, which is affected by both the linearity performance of the source signals and all the concerned random effects, such as noise and clock jitter. Source stimuli with better linearity are necessary for this method to obtain accurate measurements. In addition, for an *n*-bit ADC under test, it needs 2^n memory cells, which will be large when high-resolution ADCs are under test, to save the histogram counts during the linearity characterization processes. Both of these two factors make the histogram test difficult to use in BIST applications. Another commonly used method is the servo-loop feedback method, which is able to measure any specific ADC transition level using an additional precise digitizer. However, this technique is quite slow and significantly limits the total number of ADC transition levels that can be tested. In some applications, especially for high-resolution ADCs, the reduced-code testing instead of the full-code testing has to be applied to cut the testing time. In the case of built-in applications, the precise

digitizer will dramatically increase the area overhead and the design efforts. Therefore, this technique is often not a feasible solution either.

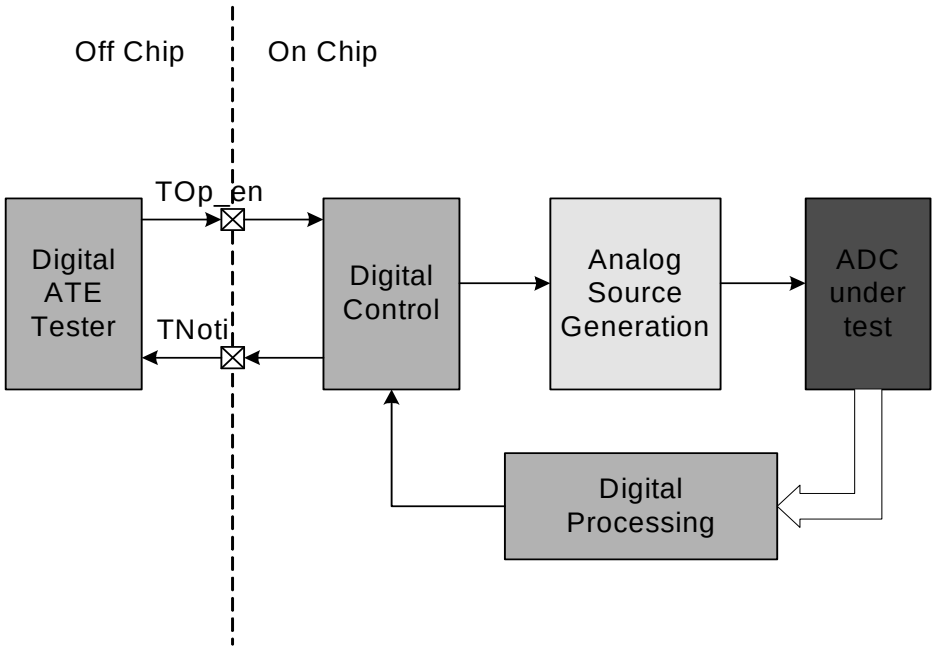


Figure 2.6. Block diagram of ADC BIST structure.

Usually, BIST solutions to ADC linearity testing should be able to provide source generation and response verification abilities for on-chip testing. Figure 2.6 shows a general block diagram of an ADC BIST solution. A digital control block controls the BIST procedure and communicates with external test equipment, and an analog source generator is built on-chip to produce source stimuli for testing. At the output of the ADC under test, a digital processing block captures the output data and computes the specifications of the ADC. The results are then sent out to the tester through the digital control block. With the help of on-chip BIST circuits, very inexpensive digital testing equipment can be used by simply sending an enable signal for start and receiving the test results in digital format after the test procedure is finished. Compared with the traditional methods, a successful BIST

implementation for ADC linearity testing tends to satisfy the following conditions. First, it includes a very low-cost, on-chip stimulus generator, which is more accurate than the ADCs under test and is able to provide stimulus signals for at-speed testing. Second, no complex digital signal processing and microprocessor are needed to obtain the measurements and the linearity performance of the ADC transition levels. Finally, the BIST strategy should be capable of characterizing the transition levels with small memory and digital circuit overhead.

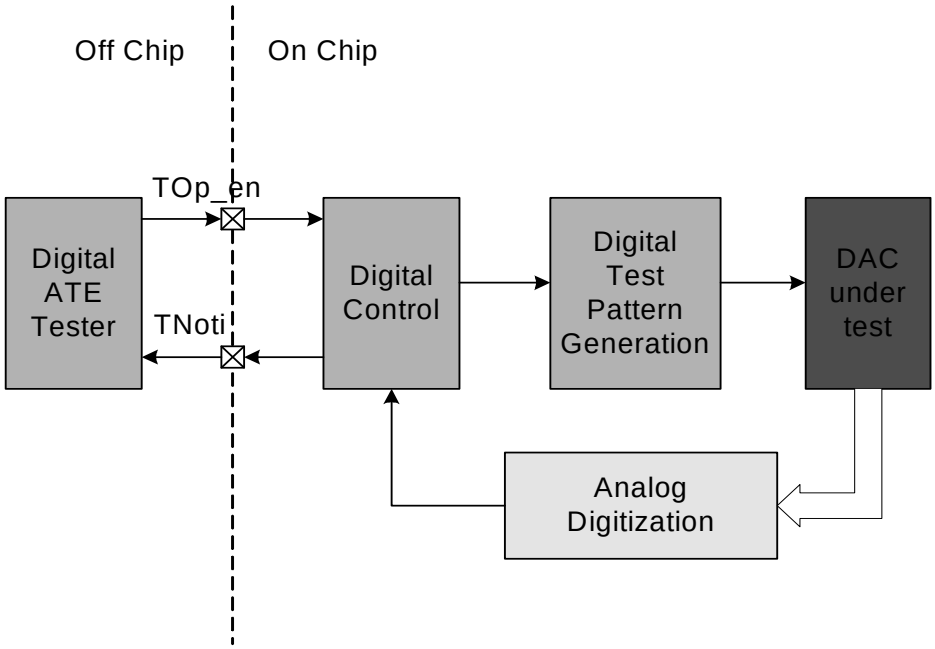


Figure 2.7. Block diagram of DAC BIST structure.

DAC testing is more challenging, especially for those high-resolution and high-speed parts recently developed for new applications. It is well known that the performance of the latest commercial DACs is much better than the best ADCs available in industry in terms of resolution and speed. Unfortunately, the performance of a DAC has to be evaluated by measuring its output levels, which is usually done by using a much more accurate ADC. In this case, the testing has to be at a low speed, which is limited by the ADC. For some cases in

which the resolution of DACs under test is state of the art, ADCs used for testing are not even available. Built-in self-test of DACs involves on-chip implementation of digitizers. Figure 2.7 illustrates a general block diagram of a DAC BIST solution. A digital control block controls the test procedure and communicates with external test equipment. A digital test pattern generator produces a sequence of digital codes to the input of a DAC under test, and the DAC's outputs are measured by an on-chip digitizer. *INL* and *DNL* information are calculated by digital circuits from the measurements and then sent out to the external digital tester. Similar to conventional external testing of DACs, the digitizer has to be much more accurate than the DAC under test. When a high-resolution DAC is under test, building such a high-precision, on-chip digitizer is extremely difficult and usually overwhelming.

2.3.2 Data Converter BIST Examples

2.3.2.1 ADC/DAC loop-back testing

Many mixed-signal systems contain both an ADC and a DAC, as shown in Figure 2.1. In this case, instead of testing the ADC and the DAC directly as the conventional method for discrete parts, this ADC/DAC pair enables an all-digital embedded testing of the on-chip circuits by reconfiguring them (i.e. by connecting the analog output of the DAC to the analog input of the ADC, possibly via some analog circuit under test) as shown in Figure 2.8.

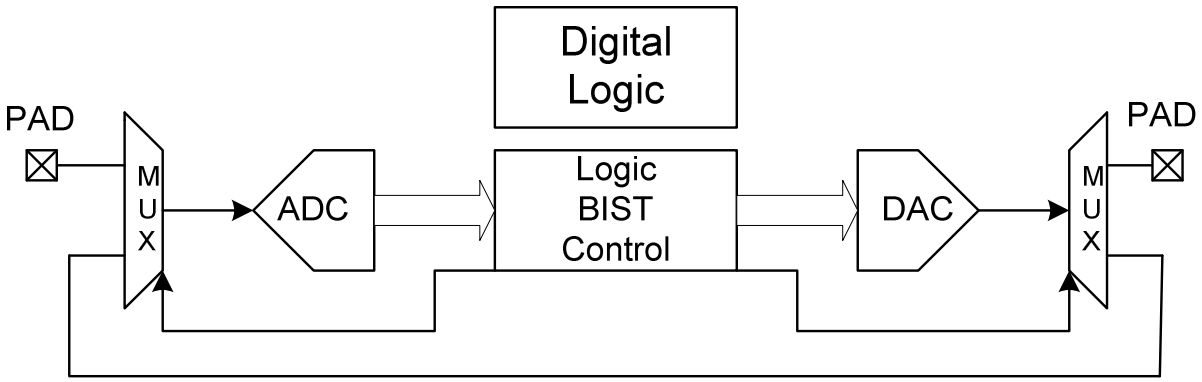


Figure 2.8. Example scheme for BIST of DAC and ADC

BIST then can be implemented in digital domains by providing a digital stimulus to the DAC using a digital source generator and monitoring the digital output of the ADC with a digital response verification circuit. This is usually referred to as the ADC/DAC loop-back BIST. As shown in Figure 2.8, the analog multiplexers are the only circuits associated with the BIST architecture to be inserted in the analog domain. This strategy minimizes the impact of the BIST circuitry on the operation and performance of the analog circuitry. The target circuitry under test is the analog system circuits, including the DAC and ADC as well as any analog circuits between them. The analog multiplexers are used to select different analog channels for test.

This ADC/DAC loop-feedback method has been shown to be effective in detecting catastrophic faults. In this case, an on-chip Linear Feedback Shift Register (LFSR) is usually used as the source generator to create pseudo-random inputs. The generated digital signals would be converted into analog form by the DAC circuit where it would then excite the ADC circuit. Then, the digital ADC output would be collected and processed by a simple digital-response analyzer [8] [9] [10]. The result would be compared to a digital signature stored on-chip in memory. Unfortunately, in this approach, the tests being performed are not the same

as those performed in production testing, but some other applications of the method are able to do specification testing with certain stimulus patterns and more complex digital processing. For example, in [11], the transfer function of the ADC/DAC pair in the digital domain is first obtained. Then, the proposed efficient polynomial fitting is applied to the transfer curve to characterize offset, gain, and harmonic distortions of the circuit.

The limitations of the ADC/DAC loop-feedback method are primarily due to the configuration and the characteristics of on-chip converters. First, this method is limited to applications where there is an ADC/DAC pair on the same IC. Second, since the whole analog signal path is tested as a black box in the all-digital test environment, the detected faults can not be easily isolated. Third, the loop-feedback test configuration cannot detect masked faults (i.e. those which occur in the DAC and in a complementary way in the ADC). However, in real operation, masked errors do degrade system performance, so providing additional tests to address masked faults is necessary. Fourth, the test accuracy is restricted by the resolution of the converters. If one of the data converters in the loop has already been calibrated, the other's characteristics can be measured. Furthermore, if the ADC and the DAC are calibrated, the analog circuits between them can be tested. For example, the outputs of the DAC and analog circuits can be measured by the calibrated ADC. The test accuracy is then limited by the quantization errors of the ADC. Usually the ADC (or DAC) used to test the DAC (or DAC) should have at least two bits of resolution more than the DAC (or ADC) under test. Finally, the test accuracy could be limited by the modeling error. For example, in the polynomial fitting method, the transfer function is modeled as a 3^{rd}-order polynomial and, therefore, it is more suitable for devices with insignificant 4^{th}- and higher-order harmonics.

2.3.2.2 Oscillation-test BIST

Another promising technique to test mixed-signal circuits is the oscillation-test strategy, which is very practical for designing effective BIST circuits [12] [13] [14]. Figure 2.9 shows the block of the oscillation-test method used to apply BIST to ADCs. At the input of the ADC under test, a capacitor is charged or discharged by a constant current. With the aid of some logic control in the feedback loop, the system oscillates between two pre-established codes. Then, functional specifications, such as offset, gain error, *DNL*, and *INL*, are evaluated by measuring the oscillation frequency of the circuit under test. Digital signatures can be generated by the control logic block and sent to external digital ATE.

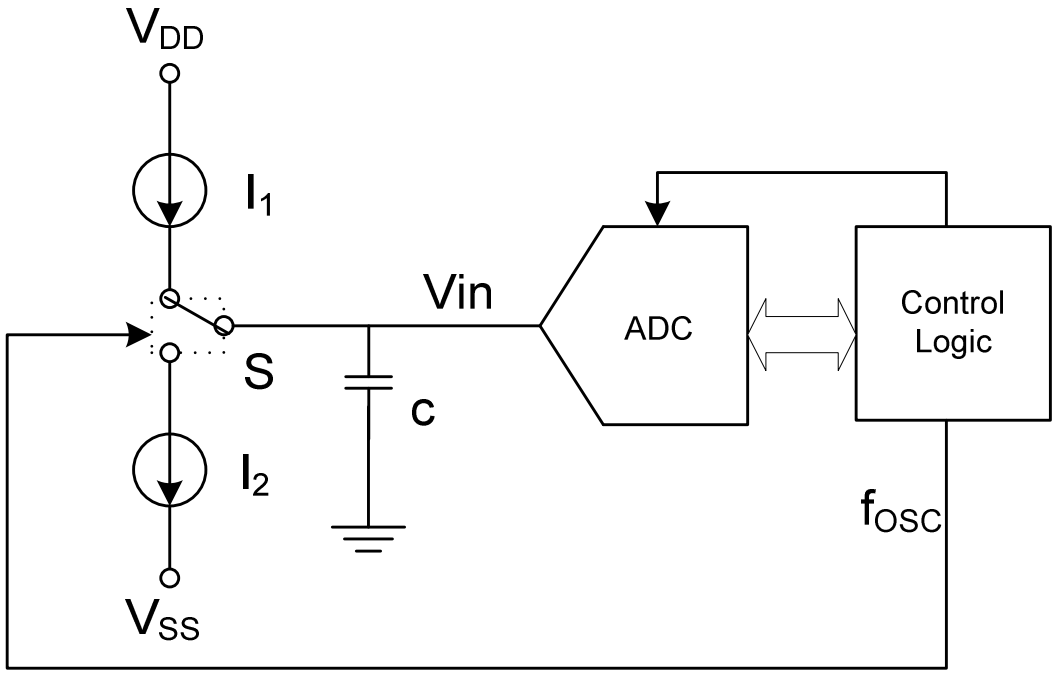


Figure 2.9. Block diagram of the oscillation-test method used in ADC BIST

Built-in self-test for DAC is difficult since DACs' outputs are analog signals and the measurements require high-resolution yet area efficient analog digitizers. The oscillation-test method deals with this problem by putting the DAC under test in a closed-loop in which the

output of the DAC under test oscillates between two analog values. Figure 2.10 illustrates the block diagram of the oscillation-test method used in DAC BIST. The loop is basically a 1st-order, delta-sigma modulator. The DAC under test is put in the feedback loop and used as a 1-bit DAC by presetting two input codes. The switching between the two available codes is controlled by the outputs of the comparator. Because of the infinite DC gain of the integrator, the average of the DAC outputs is forced to be equal to the input DC level. The digital signature of the DAC outputs can then be generated by the up/down counter at the output.

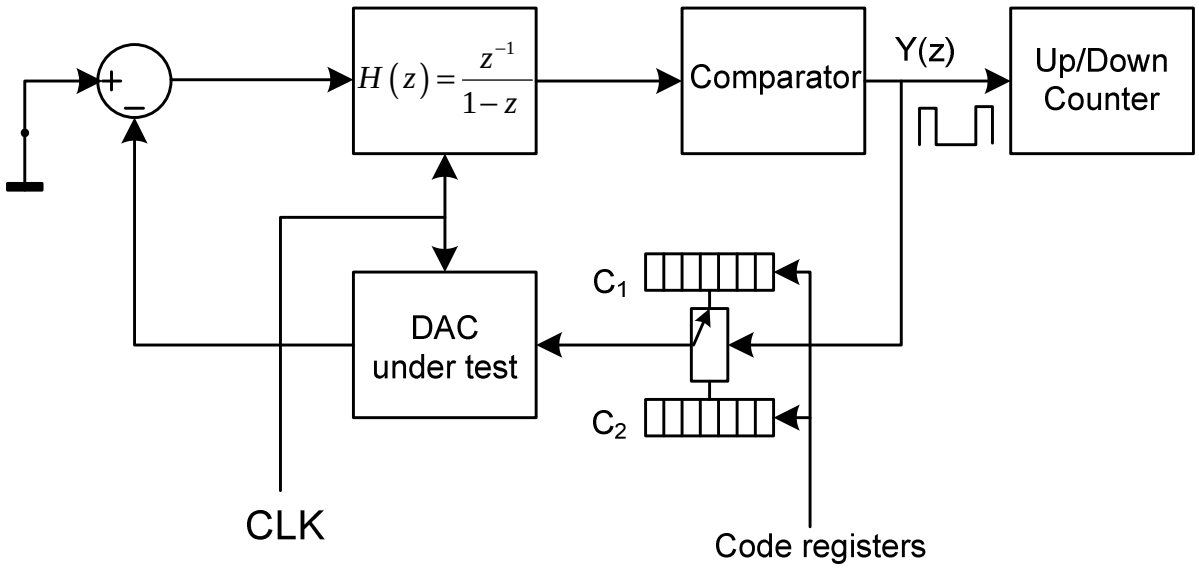


Figure 2.10. Block diagram of oscillation-test method used in DAC BIST

The limitations of the oscillation-test methods are primarily caused by the on-chip analog circuitry. For example, in Figure 2.9, the linearity of the capacitor charging and discharging limits the accuracy of ADC testing. In addition, building high-performance analog BIST circuitry with reasonable area is usually extremely challenges. Another limitation of the oscillation-based methods is the difficulty in accurately measuring the oscillation frequency. Frequency can be easily measured in a digital domain, but measuring resolution depends on many issues in both analog and digital systems, such as the currents I_1

and I_2, the sampling frequency of ADC under test, noise, and so on. The frequency measuring usually takes a long time to minimize the noise effect and achieve desired test resolution, especially for a high-resolution converters under test.

Chapter 3. High-resolution ADC Linearity Testing Using a Fully Digital-Compatible BIST Strategy

This chapter discusses a digital-compatible built-in self-test strategy for high-resolution ADC linearity testing using only digital testing environments. The on-chip, analog-stimulus generator consists of three low-resolution and low-accuracy current steering DACs, which are area efficient and easy to design. The linearity of the stimuli is improved by the proposed reconfiguration technique. ADCs' outputs are evaluated by simple digital logic circuits to characterize the nonlinearities, and the proposed BIST strategy is capable of characterizing ADC transition levels one by one with small hardware overhead. Furthermore, the testing performance is not sensitive to mismatches and process variations so that the analog BIST circuits can be easily reused without complex self-calibration. Simulation and experimental results verify the test performance of the proposed circuitry and BIST strategy.

3.1 Introduction

In the past few years, many papers have been published discussing approaches to the linearity BIST techniques for ADC. They can be divided based on the test methods that they rely on into different groups, such as the oscillation based [15], the FFT-based related [16], and the histogram related [17]-[20], and so on. Most of those methods require on-chip generation of test stimuli in test mode.

As discussed in chapter 2, the histogram test cannot be easily used in BIST applications because of two limitations: the linearity of on-chip produced test stimuli and the requirements on on-chip memory and DSP. Different methods of building on-chip source generator have been proposed to provide high-linearity source signals. In [17], on-chip

generation of sine waves was described and used in the histogram test; however, this kind of signal is difficult to produce on-chip with the required spectral purity. More common source signals for on-chip histogram test are ramps or triangle waves. A common way to generate a voltage ramp is to charge a capacitor by a constant current [18] [19], and to reduce the effect of the finite output impedance of the current source, a differential amplifier can be used in a feedback configuration [20]. The linearity is then still affected by the nonlinear capacitor's leakage current and the amplifier nonlinearities. Thus, the analog circuits need to be carefully designed. Another important issue in the ramp signal is the slope, which should be small enough for high-resolution testing and well controlled. However, when the desired charging current is quite small, the slope is very sensitive to the process variations. Moreover, adaptive ramp generators has been investigated to self calibrate the slope [19] [20] [21] [22] [23]. The best linearity reported in the papers is about 14-15 bit linear, with any ability to test ADCs with up to 12-13 bits resolution [19] [22] [23]. Histogram-based ADC testing requires access to a memory to collect code-bin hit information. If the memory is not already built on-chip, hardware overhead of additional memory for BIST could be excessive, especially for high-resolution ADCs under test. In [24], the authors suggested collecting the histogram of each code in a sequential manner as solution to the need of large memory. However, this method increases the test time exponentially with the ADC resolution. In [25], the authors proposed an alternative analysis technique for ADC BIST that does not increase the test time appreciably in the absence of a large memory. Furthermore, the processing of data requires on-chip digital processing ability, which limits the applications to the chips with DSP capabilities.

Instead of building high-performance source generators on-chip, some other research has been investigating using low-accuracy source stimuli, for simple implementation and less cost, with error compensation techniques. In [26] and [27], the authors proposed a linearity test method using low-linearity ramps for pipeline ADCs. The method first identifies the nonlinearity errors in the stimuli using the redundant information from the two correlated sets of collected data and then removes the stimulus error from the ADC histogram test data, allowing the ADC nonlinearity to be accurately measured. This is a promising solution to BIST source signal generation; however, the post digital processing needs a complex DSP block. Another approach to reduce the complexity of on-chip source generator is to use low-accuracy, current-steering DACs incorporated with a so-called deterministic dynamic element matching method (DDEM) [28] [29]. It has been proven that the resolution and linearity of overall output distribution can be improved by the reconfiguration method and that the proposed source generator can then be used in ADC histogram BIST for *INL* and *DNL* specifications. The latest experimental results show that 16-bit linear discrete source stimuli can be achieved using a low-accuracy, 12-bit thermometer-coded current steering DAC with 9-bit reconfiguration control [30]. Although low accuracy requirements on the original DAC relaxes design efforts and enables use of small-area current cells, designing a 12-bit thermometer-coded current steering DAC is still area consuming due to its 2^{12}-unit current cells and complicated logic control circuits. In addition, the conventional histogram-based testing still needs a large memory and complex digital processing.

In this chapter, we introduce a BIST solution to ADC linearity testing with an on-chip source generator and digital response verification ability. The method is fully compatible with digital testing environments using only digital testers and straightforward connections.

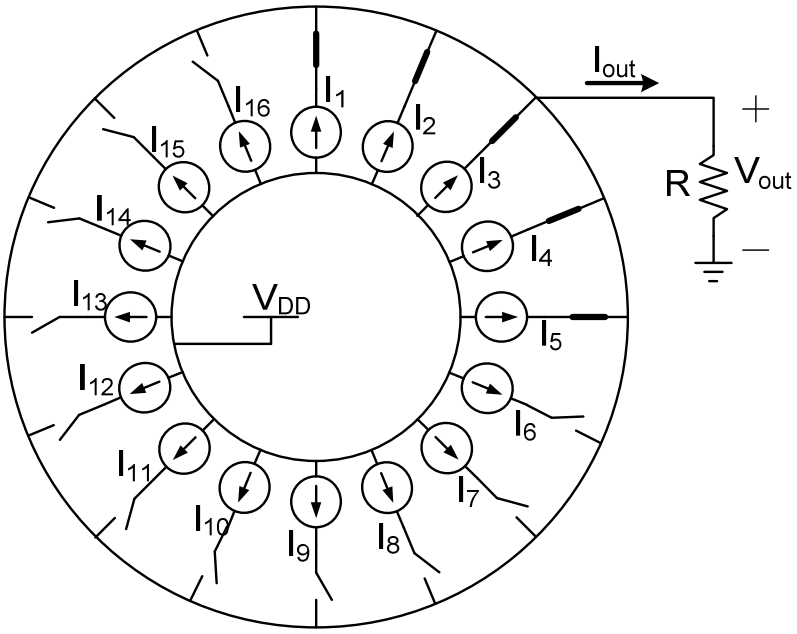
Furthermore, on-chip source generators based on the DDEM method are able to provide high testing performance under the process variation and mismatch with small silicon area and minimal design effort. No complex self calibration is required for BIST analog circuitry. The data processing for characterizing ADC transition levels can be done by simple digital logic circuits with a small memory overhead. The rest of this chapter is organized as follows. The second part briefly discusses the deterministic dynamic element matching technique being applied to thermometer-coded current steering DACs. Section 3.3 describes the structure and the performance of the proposed on-chip source generator. In section 3.4, the proposed test structure and BIST strategy are presented, and simulation results are shown in section 3.5. The design of the prototype BIST system and the Spectre simulation results are described in section 3.6, while section 3.7 shows the experimental results measured from the fabricated chips. Finally, section 3.8 concludes the chapter.

3.2 Deterministic Dynamic Element Matching

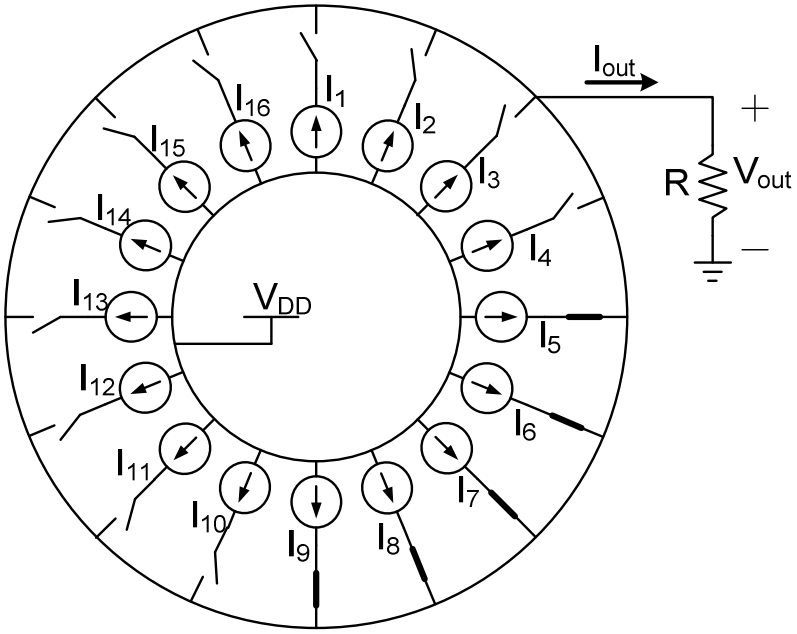
Mismatch errors are inevitable due to process variations, and although special layout techniques, such as special processes and laser trimming, can be used to reduce matching errors, these methods lead to significant cost increases. The dynamic element matching (DEM) technique accepts matching errors as inevitable and dynamically rearranges the interconnections of the mismatched elements so that, on the average, all element values are nearly equal. The DEM method has been used by many researchers to improve the performance of DACs, especially for the DACs in Delta-Sigma converters. From the spectral point of view, the randomizing effect of DEM spreads the mismatch errors in the DAC over a wide spectrum so that higher SFDR becomes possible [31]-[34]. There are different

approaches to applying dynamic element matching to a DAC, and they can be divided into two groups, random DEM and non-random DEM, based on the way of doing rearrangement.

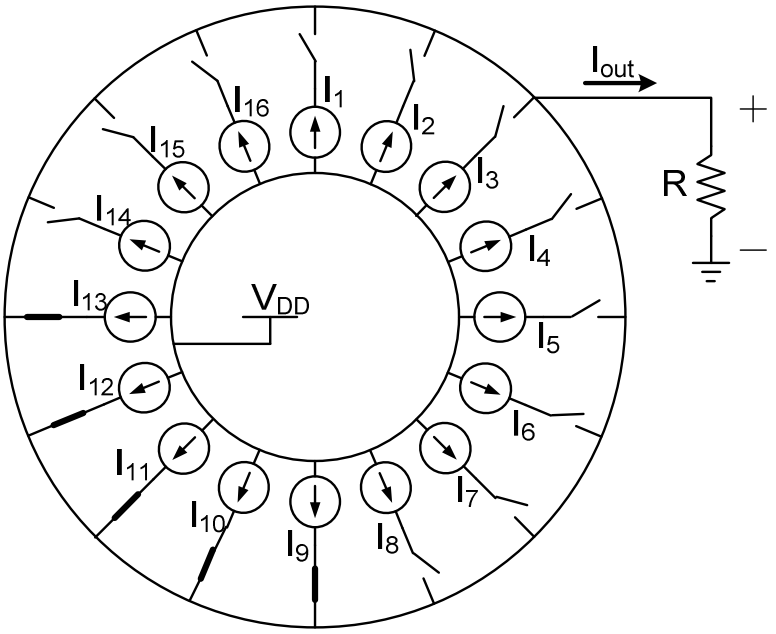
The so-called deterministic dynamic element matching belongs to the latter of the two groups. It was first described in [35], and the basic idea is that instead of generating the desired signal with a costly high-resolution and high-accuracy DAC, we create a set of cheap and "poor" DACs by reconfiguration the mappings between the current cells and the input digital codes. Each of these DACs generates a series of low-resolution and low-accuracy output samples. If all these samples, which are distributed in a common range, follow a nearly uniform distribution, the equivalent output linearity of the DAC will be improved. Assume a *n*-bit thermometer-coded, current-steering DAC with DDEM has $N=2^n$ current cells, denoted as i_1, i_2, ..., i_N, which is one current cell more than a normal *n*-bit current steering DAC. The DDEM technique creates *P* (a submultiple of N_M and an exponent of 2) different configurations, which is controlled by a $\log_2 P$-bit control code C_{DDEM}. For simplicity, we conceptually put all the current elements on a circle clockwise from i_1 to i_N then back to i_1, as shown in Figure 3.1. To generate an output voltage for input code *d*, we may start from i_1 and clockwise turn on *d* consecutive current cells. Then, *P* analog outputs can be generated for one digital input code by choosing *P* different current cells as the start points. In the DDEM algorithm, those *P* start points are evenly distributed on the circle. A simple 4-bit DDEM-incorporated DAC example, with $P = 4$ and $d = 5$, is illustrated in Figure 3.1. The four start points selected are i_1, i_5, i_9, and i_{13}. Figures 3.1.a, 3.1.b, 3.1.c, and 3.1.d show the cases that the selected five current cells start from i_1, i_5, i_9, and i_{13}, respectively.



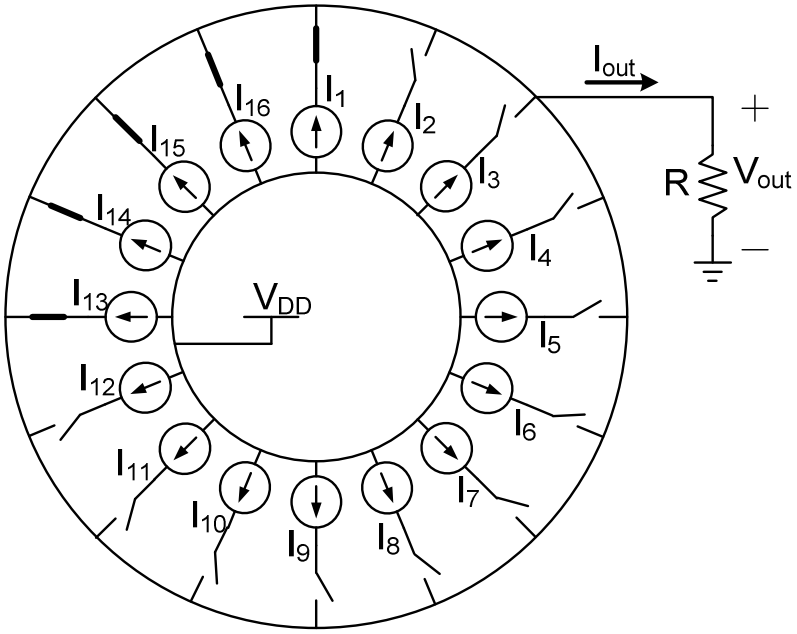
a) 1st output sample with i_1~i_5 on



b) 2nd output sample with i_5~i_9 on



c) 3rd output sample with i_9~i_{13} on



d) 4th output sample with i_{13}~i_{16} and i_1 on

Figure 3.1. 4-bit DDEM DAC with *P*=4 and input code *d*=5

In the previous research, the DDEM-incorporated DACs are used as source generators, producing discrete stimulus samples for ADC histogram testing [28] [29] [35].

The distribution of all possible output samples from a DDEM DAC is, therefore, important to the test performance. Theoretical analysis focuses on the overall output distribution and indicates that in a certain parametric range of *P*, the test performance of a DDEM DAC as a source generator is improved by $\log_2 P$ bits, where *P* is the number of the reconfigurations as well as the number of the DAC outputs associated with one input code. However, out of the range, the test performance will not be improved as fast as shown in the analysis. The previous works fail to explain why the test performance improvement is limited when *P* becomes large.

Design of the DDEM DAC as a source generator is explained in [30]. The logic circuit is simple since the ramp signals, which are enough for histogram testing, can be easily generated using a shift-register loop with each unit current source controlled by one shift register. The ramps can be generated for different configurations by starting the shift register loop at different points. However, the proposed design loses the control of the DAC output levels. Furthermore, the memory and DSP issues of the histogram testing need to be addressed before we use the DDEM DAC in a BIST application.

In this work, the DDEM technique is utilized in another way, in which the DDEM DAC is used not to simply generate ramp signals but as a calibrated high-performance DAC. This DAC is then put into an ADC feedback loop test system to create stimuli to the input of ADC under test for characterizing its transition levels. It is shown in the following sections that no large memory or complex DSP is needed in the proposed BIST solution.

3.3 On-chip Source Generator

This section describes the structure of the source generator used in the proposed BIST solution. The source generator consists of only small and low-accuracy DACs, which are cost

efficient and easy to design. The test performance is ensured by the deterministic DEM technique and the dithering method, as later shown in the theoretical analysis.

3.3.1 Segmented DDEM DAC

The source generator is primarily a reconfigurable, low-accuracy segmented current steering (SCS) DAC. Segmented structure achieves a good combination of high resolution and small area. As shown in Figure 3.2, an n-bit, segmented-current steering DAC usually consists of an n_M-bit thermometer-coded MSB array and an n_L-bit binary-coded LSB array, where $n=n_M+n_L$. The MSB array and LSB array generate currents according to their input digital codes, D_M and D_L respectively. The total current generated is then forced to flow through a resistor to produce the output voltage. In the normal applications, the MSB array's linearity dominates the whole DAC's performance and needs to meet the specification of the whole ADC. For example, a 10-bit segmented DAC, which has a 5-bit MSB part and a 5-bit LSB part, specifies 10-bit linearity performance. Thus, the 5-bit MSB array has to be no less than 10-bit linear. As a result, the MSB array requires considerable consumption of area and power. To reduce the cost, this work uses a low-accuracy, thermometer-coded MSB array with its original linearity much lower than the required test performance. It will be shown later that the nonlinear error of the MSB array is reduced by the reconfigurations controlled by the DDEM technique. The LSB array is a normal binary-coded current DAC with simple control logic and small area. The linearity of the LSB array is much less critical to the test performance since its full-scale output range is small.

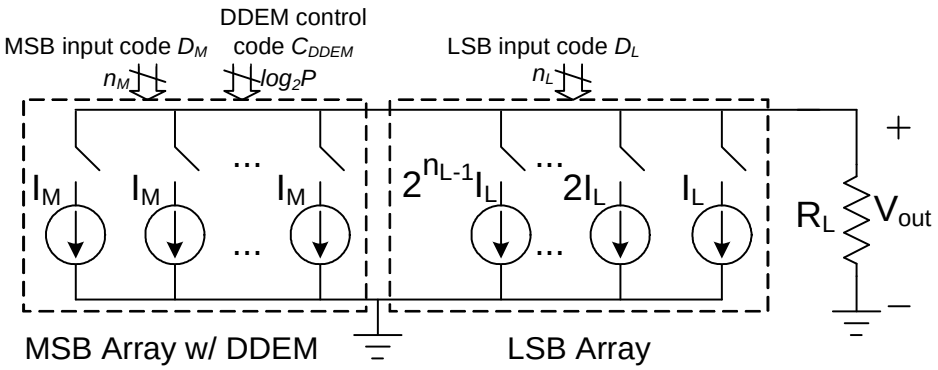


Figure 3.2. *n*-bit segmented current steering DAC

3.3.2 Test Performance Analysis

In this section, we will analytically show the linearity improvement of the DAC and evaluate the overall testing performance. In order to separate the contributions of DDEM and the LSB array, we first consider only the MSB array with DDEM as the source generator. With the estimation error identified, we then add the effect of the LSB array into the analysis to see how it further improves the accuracy.

First, assume the outputs of the DDEM MSB array are used to measure a specific ADC transition level, T_k. The procedure is that under each DDEM configuration (for example the j^{th} configuration, j=1, 2, …, P) we search for a digital input code d_j that satisfies $V_j(d_j) \leq T_k < V_j(d_j+1)$, where $V_j(d_j)$ and $V_j(d_j+1)$ are the MSB array outputs associated with input codes d_j and d_j+1, respectively. In this method, the measurement of T_k is represented by the average of d_1, d_2,…, d_P under different configurations. Used in measuring, the codes d_j, j=1, 2, …, P actually represent analog voltage levels at code d_j defined by an ideal n_M-bit DAC, and the equivalent analog measurement of T_k is expressed by

$$\hat{T}_k = \frac{1}{P}\sum_{j=1}^{P} V_{id}(d_j), \tag{3.1}$$

where $V_{id}(d_j)$ is the output voltage at code d_j if an ideal n_M-bit DAC is used as the MSB array. Thus, the measuring error is

$$e_k = \hat{T}_k - T_k = \frac{1}{P}\sum_{j=1}^{P} V_{id}(d_j) - T_k \quad . \tag{3.2}$$

Under the j^{th} configuration, T_k, can be expressed by the sum of the DAC output at code d_j and a residue voltage as in

$$T_k = V_j(d_j) + r_j(k), j = 1,2,\cdots,P \quad , \tag{3.3}$$

where $r_j(k)$ is denoted as the residue voltage between the transition level and the MSB array output at code d_j. Thus, considering all the configurations, we have

$$T_k = \frac{1}{P}\sum_{j=1}^{P}\left[V_j(d_j) + r_j(k)\right]. \tag{3.4}$$

Substituting equation (3.4) into equation (3.3) leads to the further expression of the measuring error as in

$$\begin{aligned} e_k &= \frac{1}{P}\sum_{j=1}^{P} V_{id}(d_j) - \frac{1}{P}\sum_{j=1}^{P}\left[V_j(d_j) + r_j(k)\right] \\ &= \frac{1}{P}\sum_{j=1}^{P} r_j(k) - \frac{1}{P}\sum_{j=1}^{P}\left[V_j(d_j) - V_{id}(d_j)\right] \end{aligned} \quad . \tag{3.5}$$

It is noted that $V_j(d_j)–V_{id}(d_j)$ in (3.5) is the integral nonlinearity of the j^{th}-configuration MSB array at code d_j, designated as $INL_j(d_j)$. From the definitions of DAC's integral and differential nonlinearities, $INL_j(d_j)$ can be rewritten as the sum of a set of the differential nonlinear errors as in

$$\frac{V_j(d_j)-V_{id}(d_j)}{LSB}=INL_j(d_j)=\sum_{m=0}^{d_j-1}DNL_j(m), \tag{3.6}$$

where *LSB* is the least significant bit of the MSB array and defined by the average size of all the N_M steps, and $DNL_j(m)$, m=0, 1, …, d_j-1, are the j^{th}-configuration MSB array's differential nonlinearity errors. $INL_j(\cdot)$ and $DNL_j(\cdot)$ in equation (3.6) are in *LSB*s. Therefore, the measuring error is further given by

$$e_k=\frac{1}{P}\sum_{j=1}^{P}r_j(k)-\frac{1}{P}\sum_{j=1}^{P}\sum_{m=0}^{d_j-1}DNL_j(m). \tag{3.7}$$

The definitions based on the end-point fit line provides that the differential nonlinear errors satisfies

$$\sum_{i=0}^{N_M-1}DNL(i)=0. \tag{3.8}$$

In the DDEM technique, the differential nonlinearity errors of the MSB array are cyclically shifted with the current cells for different configurations. This fact combined with the definition of the DAC's differential nonlinearity gives us

$$\sum_{j=1}^{P}\sum_{t=0}^{s*q-1}DNL_j(t)=s\times\sum_{i=0}^{N_M-1}DNL_1(i)=0, \tag{3.9}$$

since the expression in (3.9) exactly covers the differential nonlinearity errors of all the N_M current sources for s times, where $q = N_M/P$ and s is a number in 1, 2, …, P. Assume s is the largest number that satisfies d_j-$sq \geq 0$ for j = 1, 2, …, P. We can rewrite the measuring error in (3.7) as

$$e_k = \frac{1}{P}\sum_{j=1}^{P} r_j(k) - \frac{LSB_M}{P}\sum_{j=1}^{P}\sum_{m=0}^{d_j-sq-1} DNL_j(m), \tag{3.10}$$

where LSB_M is the ideal MSB array least significant bit.

To evaluate the testing performance, we start with the first term in (3.10). The residue voltages $r_j(k)$s are originally at n_M-bit level because of the MSB array's resolution. That means the MSB array outputs cannot approach the transition level under test very accurately due to its step sizes. With the help of the segmented structure, the resolution can be increased by adding the LSB array. Then, the transition level T_k can be further approximated by the LSB array outputs by adding its output levels to $V_j(d_j)$. The new residue voltages will be at $n_{DAC}=n_M+n_L$ bits level because they are the difference between T_k and the segmented DAC's outputs that are the closest to but less than it. Furthermore, for any transition level T_k the residue voltages are randomized by DDEM because of the mismatch errors. Assuming the residue voltages are randomly uniform-distributed, the variation of the first term in (3.10) is approximately at $n_{DAC}+0.5\log_2 P$ bits level. Here, we ignore the effects of LSB array nonlinearity for two reasons: first, the full-scale output range is very small and that makes the nonlinear error of the LSB array at high-resolution level, and second, the averaging of the residue voltages can further reduce the nonlinear error effect (in the fully random case, it is reduced by $0.5\log_2 P$ bits). Then, the overall effect of the LSB array nonlinearity can be easily controlled to be no larger than the error induced by the DAC's resolution, which is at $n_{DAC}+0.5\log_2 P$ bits level.

The second term in (3.10) is induced by the nonlinearity of the original MSB array. If we assume the linearity performance of the MSB array is not worse than its resolution, n_M bits, the second term can be simplified to the sum of a set of non-repeating $DNL_1(k)$s divided

by P. The maximum value of the sum is comparable to the INL of the original MSB array, INL_M. Then, the nonlinearity of the original MSB array is reduced by $\log_2 P$ bits.

On the whole, the equivalent testing performance of the segmented DDEM DAC is expressed by

$$n_{eq} \approx \min\{n_{DAC} + 0.5\log_2 P, ENOB_M + \log_2 P\}\ bits, \tag{3.11}$$

where

$$ENOB_M = n_M - \log_2\left(\frac{INL_M}{0.5}\right) = nM - \log_2 INL_M - 1 \tag{3.12}$$

is the effective number of bits of the original MSB array. Therefore, if we assume the segmented structure provides enough resolution so that the first term in (3.11) does not limit the testing performance (that means it is much larger than the second term), the linearity of the testing stimuli is improved by $\log_2 P$ bits.

3.3.3 Structure of the Source Generator

The target of this work is to test high-resolution ADCs, which requires highly linear source signals. The low-accuracy segmented DDEM DAC is built on-chip as the source generator. Considering the number of the current cells and the complexity of the digital control block, we still want the thermometer-coded MSB array as low resolution as possible. In this case, the second term in (3.11) will probably limit the test performance, since when the MSB array is low resolution, the number of DDEM configurations, P, has to be small (we have $P \leq N$ by the rearranging algorithm). A solution to this problem is to incorporate another low-resolution DAC to generate extra linear dither steps at the output. As shown in

Figure 3.3, these small dithers are added to the outputs of the DDEM DAC. Each output of the DDEM DAC is spread by N_d dither levels, where N_d is the resolution of the dither DAC in decimal. Then the measurement of the transition level T_k, m_k, is expressed as

$$m_k = \frac{1}{P}\sum_{j=1}^{P}\sum_{i=1}^{N_d} d_{j,i}, \tag{3.13}$$

where $d_{j,i}$ is the obtained DDEM DAC input code when the j^{th} DDEM configuration and the i^{th} dither output are applied. $d_{j,i}$ and i satisfy $V_j(d_{j,i})+V_d(i) \le T_k < V_j(d_{j,i}+1)+V_d(i)$, where $V_j(\bullet)$ and $V_d(\bullet)$ are the transfer functions of the j^{th}-configuration DDEM DAC and the dither DAC respectively. The output range of the dither DAC is set to be $q=N_M/P$ LSB_Ms, since it is noticed that the second term in (3.10) has a repeating form for different ADC transition levels with a period of q LSB_Ms. To verify this statement, we can simply consider two transition levels, T_k and T_k+qLSB_M. Assume the estimation error of the transition level T_k from the nonlinear error is e_k. Then, based on the cyclic shifting of DDEM, the estimation error of the transition level T_k+qLSB_M, e_k' is about

$$e_k^{'} \approx e_k + \sum_{i=0}^{N_M-1} DNL_1(i) = e_k, \tag{3.14}$$

approximately the same as that of the transition level q LSB_Ms away from it. Thus, linearly spreading the error distribution over one period range and getting the average will effectively reduce the error variation and improve the testing performance. Adding dither DAC also increases the number of residue voltages averaged so that it will help reduce the error from the resolution limitation as well. Theoretically, the estimation error with dither DAC incorporated can be expressed by

$$e_k = \frac{1}{N_d P}\sum_{i=0}^{N_d-1}\sum_{j=1}^{P} r_{j,i}(k) - \frac{LSB_M}{N_d P}\sum_{i=0}^{N_d-1}\left(\sum_{j=1}^{P}\sum_{m=0}^{d_{j,i}-sq-1} DNL_j(m)\right), \\ \approx \frac{1}{N_d P}\sum_{i=0}^{N_d-1}\sum_{j=1}^{P} r_{j,i}(k) - \frac{LSB_M}{N_d P} INL_M + C \tag{3.15}$$

where $r_{j,i}$ is the residue voltage when the j^{th} DDEM configuration and the i^{th} dither output are applied, and C is a constant induced by the average dither level. Then, it can be shown that the effect of the dither DAC on the testing performance is very similar to that of the DDEM. The equivalent test performance of the segmented DDEM DAC with dithering can be represented by

$$n_{eq} \approx \min\left\{n_{DAC} + 0.5\left(\log_2 P + n_d\right), ENOB_M + \log_2 P + n_d\right\} bits, \tag{3.16}$$

where n_d is the resolution of the dither DAC in bits. Here, we assume the nonlinear errors of the dither DAC do not limit the testing performance. This assumption usually holds since the full range of the dither DAC is much smaller than the DAC's output range. For effective implementation, the parameters, like n_M, n_L, P, and n_d, need to be optimized so that both terms in (3.16) are reduced to the same level.

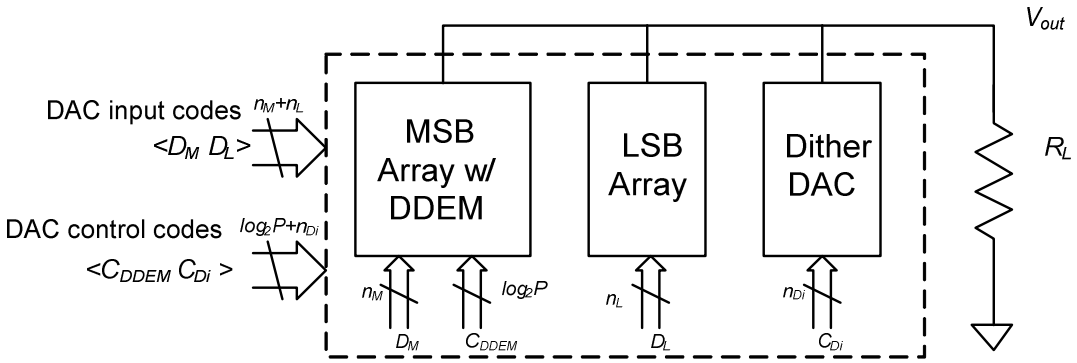


Figure 3.3. Proposed ADC BIST structure

3.4 ADC BIST Strategy

In this section, we discuss the structure of the testing system and the BIST procedure using the proposed source generator.

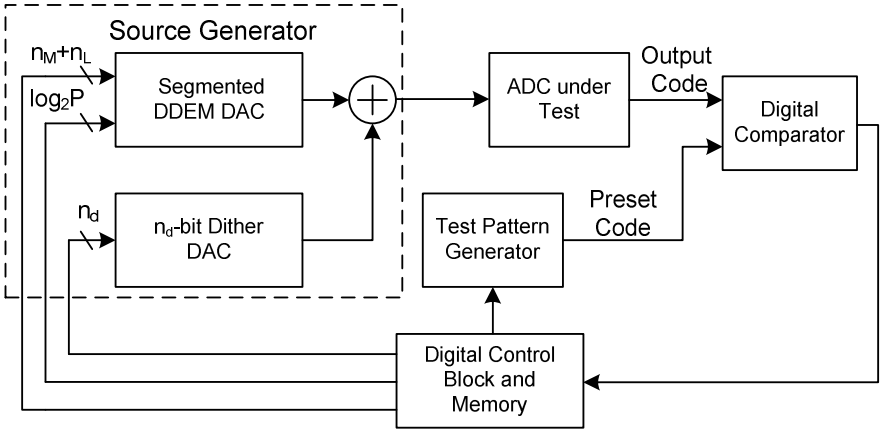


Figure 3.4. Proposed ADC BIST structure

3.4.1 Testing Structure

The proposed testing structure is illustrated in Figure 3.4. Stimulus signals to the ADC under test are generated by adding together the outputs of the dither DAC and the segmented DDEM DAC. Several digital codes, which include the input codes for the MSB array and the LSB array, the control code for DDEM configuration, and the dither DAC input, are generated by a digital control block. This block simply consists of a state machine and a small number of memory cells. A preset code, k, is set by a test-pattern generator for measuring the ADCs' k^{th} transition level T_k. The digital comparator will compare the ADC output code with the preset code k and send the result back to the control block. In addition, the comparator and the control block form a digital feedback loop. During measurement, under each DDEM configuration and dither input, the feedback loop will help find the desired input codes, d_M and d_L, for the MSB array and LSB array. These codes generate the

stimulus sample that is the closest to but less than the transition level T_k. The codes will be recorded to get the measurement of T_k. Binary search is applied to find those codes with fewer iteration cycles. The detailed procedure for measuring transition level T_k with the proposed structure is as follows.

1. Select a control code pair (j, i) for DDEM configuration and the dither DAC input, where $j=1, 2, \ldots, P$ and $i=0, 1, \ldots, N_d$-1.
2. Set k as the preset code for comparison. Do binary search for the input codes, d_M and d_L, with the following steps:

$$
\begin{aligned}
&set\ d_M = 0,\ d_L = 0\\
&for\ v = n_M : 1\\
&\quad a_M = d_M + 2\wedge(v-1)\\
&\quad a_L = d_L\\
&\quad Set\ a_M\ and\ a_L\ as\ input\ codes\ for\ SCS\ DAC\\
&\quad if\ ADC's\ output < k\\
&\qquad d_M = a_M\\
&\quad end\\
&end\\
&for\ u = n_L : 1\\
&\quad a_L = d_L + 2\wedge(u-1)\\
&\quad a_M = d_M\\
&\quad Set\ a_M\ and\ a_L\ as\ input\ codes\ for\ SCS\ DAC\\
&\quad if\ ADC's\ output < k\\
&\qquad d_L = a_L\\
&\quad end\\
&end\\
&d = d_M \times 2^{n_L} + d_L
\end{aligned}
$$

3. Add the obtained code d into a register. Go back to step 1 if there is an unused control code pair left.

After finishing the binary search for all the control code pairs, the algorithm uses the average (or the sum equivalently) of the obtained codes as the measurement of T_k and saves it for later use. The processing needs only one memory cell and a digital adder. The total memory size is then mainly determined by how many transition levels we need to record for testing at one time. The testing time will be of great concern when high test performance is desired, since the total times of binary search is $N_D P$.

In the implementation of the DDEM DAC, we introduce one bit overlapping between the MSB array and the LSB array to compensate for the considerable *DNL* errors in the MSB part and to make sure that all the residue voltages in (3.10) and (3.15) are covered by the LSB array. In this case, the DAC's resolution will be $n_{DAC}=n_M+n_L-1$ and the equivalent input code will be $d=d_M*2\wedge(n_L-1)+d_L$. A little change needs to be made in the second step of the procedure.

3.4.2. BIST Procedure

There are different approaches to verifying ADCs' linearity performance, but usually the full-code INL_k testing is preferable for complete performance identifications. However, for high-resolution ADCs, which are typically slow, the data acquisition time may be prohibitively long. In order to cut down the test time, sometimes reduced-code testing is applied. In this case, only a small subset of the ADC output codes are guaranteed. On the other hand, for the production test, we may only need to know whether an ADC meets the specification or not. This is noted as pass/fail testing. It means as long as we can find a transition level that is out of the error bound, the test is finished and characteristics of other transition levels are not important anymore. Based on the test procedure described in the

previous subsection, different BIST strategies for different kinds of ADC testing could be developed.

The flow chart of a pass/fail BIST procedure for ADC linearity testing is shown in Figure 3.5. The procedure starts after a testing-enable signal is sent by the external digital tester. The digital control block in Figure 3.4 controls the testing using a state machine. The first step is to characterize the end-point fit line of the ADC under test. The first and the last transition levels, T_1 and T_{N-1}, will be measured by setting the preset code k equal to 1 and N-1, respectively. The measuring process for each transition level is described in the previous

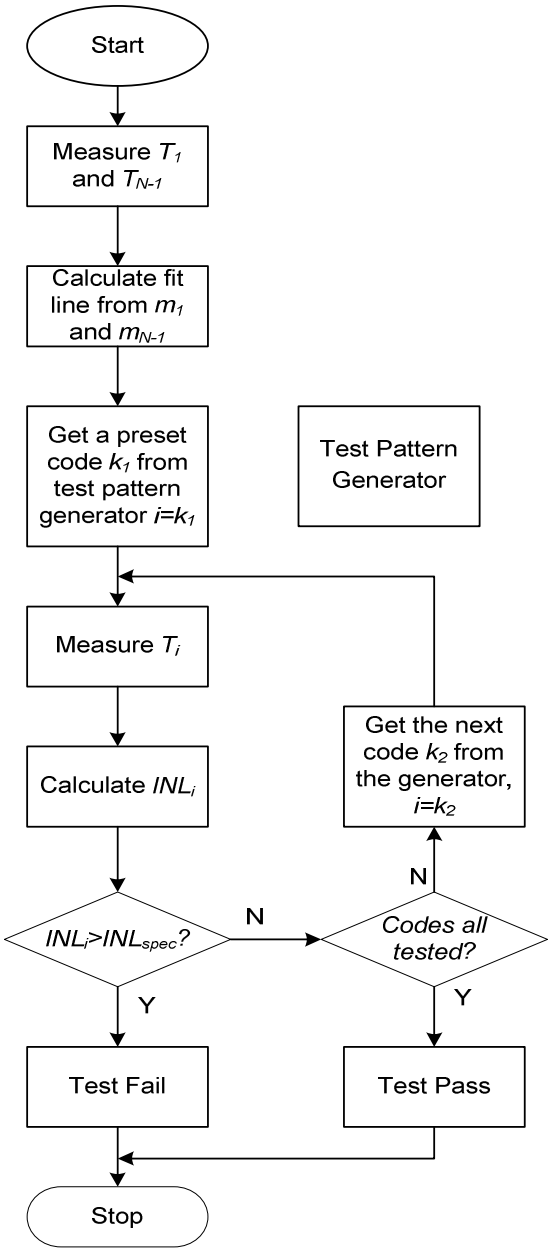


Figure 3.5. Flow chart of a pass/fail BIST procedure

subsection. The measurements then are recorded in the memory cells as the references for the ideal transfer curve. For any preset code k between 1 and N-1, the measurement of T_k can be obtained and compared with the ideal transfer curve to get the information about INL_k. The procedure requires only simple linear calculations, which can be done by digital logic

circuits. This INL_k then is compared with the specification, and if it does not meet the specification, the ADC fails the test. Otherwise, the next transition level will be tested. If all the codes that need to be tested meet the specification, the ADC passes the test. After the test is finished, a notification signal and the testing results will be sent back to the digital tester.

The test pattern generator is built on-chip to create a list of codes for testing, and a simple and general way of doing that is using a counter. After measuring the fit line, the transition levels are tested sequentially from T_2 to T_{N-2}. However, that may not be an efficient way in terms of less test time if we can find some transition levels that are more likely to have large INL_k errors than others. In this case, it makes sense to put those transition levels at the front of the list to reduce the average testing time for bad parts. Usually that information can be determined from the ADC structures. As an example, assume *n*-bit pipeline ADCs using 1bit/stage structure are under test. The gain error and the comparator offset error of each stage will cause nonlinearity. It can be shown that for a good design the largest INL_k error happens the most often around the position where the code *k* has its MSB bit just change from 0 to 1, which is 2^{n-1} in decimal form. Thus, in the testing, several codes around 2^{n-1} can be set by the test pattern generator first to get a local maximal INL_k. If it is within the error bound, the next code for test should be at the transition of the second MSB bit, which happens at two positions, 2^{n-2} and $2^{n-1}+2^{n-2}$ and so on. The code list can then be generated by a state machine based on this information. For other ADC structures like SAR ADCs, cyclic ADCs, and pipeline ADCs with different number of bits per stage, some modifications need to be made according to their individual characteristics.

3.5 BIST circuit design

The prototype test chip has been designed to verify the test performance of the proposed source generator, the test system, and procedure.

3.5.1 General Structure

The complete structure of the test system has been shown in Figure 3.4. However, in the prototype test chip, no high-resolution ADC is incorporated for testing since we do not have a design-completed ADC ready for testing, and designing such a high-resolution ADC from scratch requires a lot of effort. Instead, we include an analog comparator on-chip, which can work with an accurate reference voltage to form a 1-bit ADC as the on-chip analog circuit under test. Therefore, the proposed method can be applied to evaluate the threshold voltage of the comparator, which is equal to the reference voltage plus the offset voltage of the comparator. Assume for different reference voltages, the offset voltage of the comparator is constant. Then, we can evaluate the test performance of the proposed method and design by mapping the measurements of the threshold voltages with the known accurate reference voltages. The block diagram of the test system is illustrated in Figure 3.6. As we have described, the segmented DDEM DAC combined with the dither DAC operates as the high-performance source generator. The sum of the DDEM DAC and the dither DAC outputs is sent to the on-chip analog comparator to compare with a known accurate reference voltage. The result of comparison is fed back to the digital control block. The control block in the feedback loop adjusts the input codes to the segmented DDEM DAC according to the comparison results during the binary search process. The control code pairs, which include the dither DAC input codes and the DDEM configuration codes, are set externally by test equipment. As described in the previous sections, with each control code pair, the test system

will send out the final input code to the DDEM DAC after binary search. The final measurement of the analog reference voltage is the average of final DDEM DAC input codes under different configurations. The detailed test procedure can be found in section 3.4.

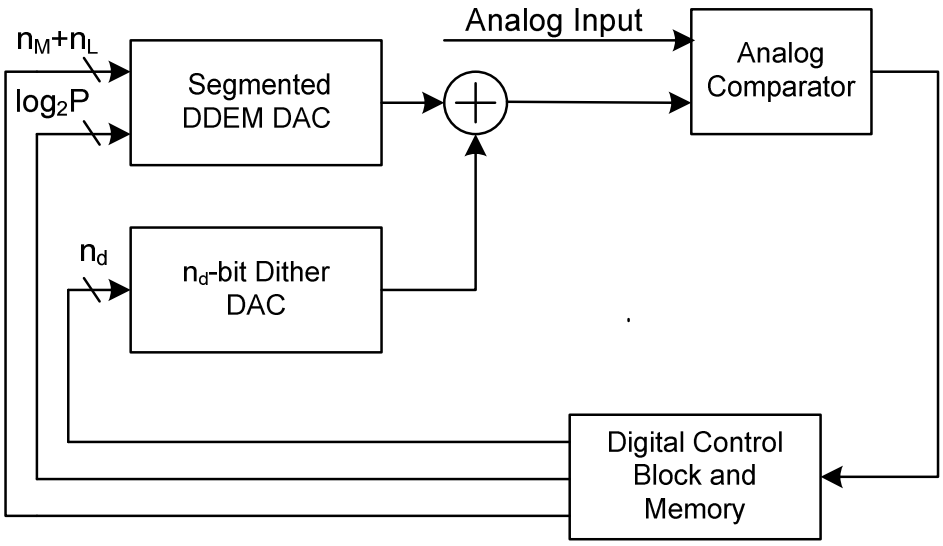


Figure 3.6. Block diagram of the test system with an analog comparator

The detailed structure and signal connections of the prototype chip are shown in Figure 3.7. Besides the three main blocks, which are the control block, the source generator and the analog comparator, the chip also includes a clock generator, which generates clock signals for the three main blocks. Main input signals to the chip incorporate the DDEM DAC control code pair, the mode select signal, start signal for binary search operation, bias voltages of the DAC current sources, clock signal, analog reference voltage to be measured, enable signal of the comparator, and power supply signals. The main output signals consist of a valid signal indicating the end of the binary search, output code signals containing the final DAC input codes after binary search, and the instantaneous DAC output voltages. The detailed signal declarations are listed in Table 3.1.

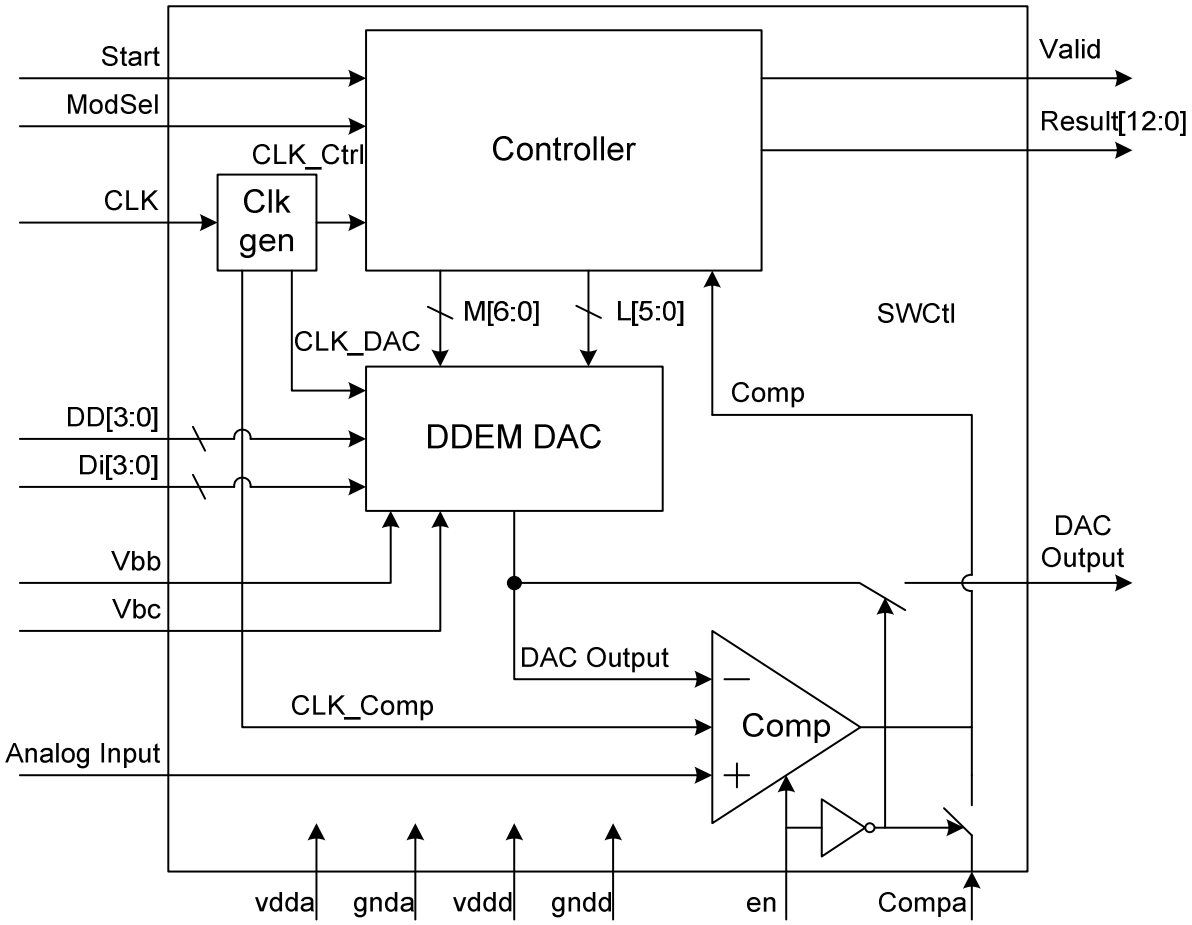
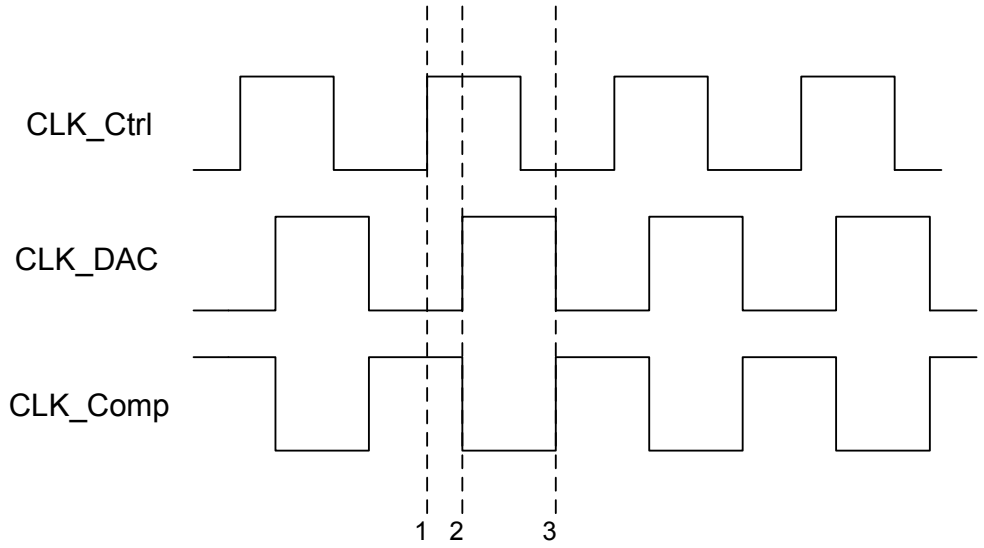


Figure 3.7. Block diagram of the prototype chip

Table 3.1. Description of external and internal signals

Pin Name	Num	I/O	Typ	Description
Start	1	in	D	Enable binary search for evaluation
ModSel	2	in	D	Select operation mode, 0: binary search, 1: ramp output
CLK	3	in	D	Clock input
DD[3:0]	4-7	in	D	4-bit DDEM control code
Di[3:0]	8-11	in	D	4-bit dither control code
Vbb	12	in	A	Bias for current source transistor
Vbc	13	in	A	Bias for cascode transistor
Vinp	14	in	A	Positive input reference voltage
Vinn	15	in	A	Negative input reference voltage
Voutp	16	out	A	Positive DAC output voltage
Voutn	17	out	A	Negative DAC output voltage
Result[12:0]	18-30	out	D	13-bit evaluation result

Valid	31	out	D	Indicate that output result is ready
en	32	in	D	Comparator enable
Compa	33	in	D	External comparison result input
Vdda	34	in/out	P	Analog power supply
Gnda	35	in/out	P	Analog ground
Vddd	36	in/out	P	Digital power supply
Gndd	37	in/out	P	Digital ground



1 Control block obtains comparison result and starts to adjust DAC input code accordingly
2 DAC input code valid and DAC switching network starts to operate
3 Comparator latch starts to operate

Figure 3.8. Block diagram of the prototype chip

The timing of the clock signals is illustrated in Figure 3.8. Three main clocks are the clock of the control block, denoted as CLK_Ctrl; the clock of the DAC switching network, denoted as CLK_DAC; and the clock of the comparator, denoted as CLK_Comp. All the clocks' signals are valid at rising edges or logic level "H." At the rising edges of CLK_Ctrl, the control block obtains the comparison results and starts to adjust the DAC input codes accordingly. When CLK_DAC is "H," DAC input codes will be valid and the switching network starts to operate and generate analog outputs, and when CLK_Comp goes high, the comparator latch starts to operate and output the comparison results of comparing the

updated DAC outputs with the reference voltage. Then, the new comparison result will be read into the logic for the next binary search cycle.

3.5.2 Operation Modes

The prototype chip has three operation modes: ramp generation mode, digitizer mode, and the external ADC BIST mode. The detailed description of the different operation modes are as follows.

Ramp generation mode

Ramp generation mode is used to evaluate the linearity performance of the DDEM outputs generated by MSB array. This operation mode is selected by setting ModSel at low and disabling the comparator. As the name indicates, in the ramp generation mode, the chip generates a step-case ramp signal by MSB array for each DDEM DAC control code pair and sends it out through DAC output pins. Figure 3.8 shows the plot explanation of this operation mode. With 4-bit dither DAC and 4-bit DDEM control, there are a total of 256 ramps generated.

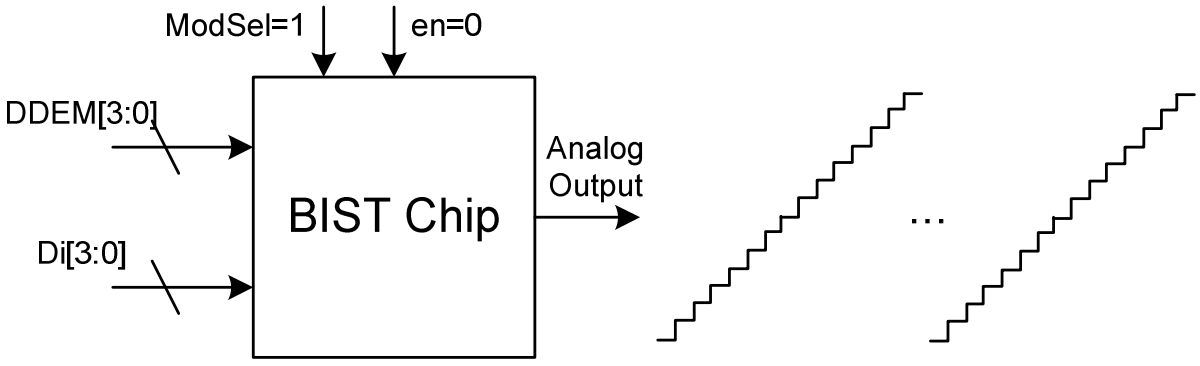


Figure 3.9. Ramp generation mode

Digitizer mode

Digitizer mode is the primary operation mode for this design, and the procedure of the mode has been discussed in the previous part in this section. The Modsel signal is set to zero and the comparator enable signal is valid, as shown in Figure 3.10. It is called digitizer mode because the external reference voltage sent to the comparator will be equivalently measured by the system shown in Figure 3.7. When the valid signal goes high, the result signal shows the final DDEM DAC input codes under certain control code pair sets by the external tester.

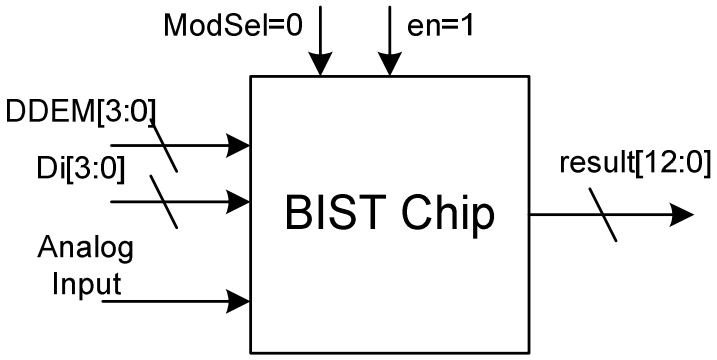


Figure 3.10. Digitizer mode setup

ADC testing mode

ADC testing mode enables the prototype working with external ADCs and digital testers to form a linearity testing system. The block diagram of the testing setup is illustrated in Figure 3.11. The ModSel signal is set to low and the comparator is disabled. Analog output from the DDEM DAC is connected to the input of ADC under test, and ADC output codes are sent to a digital comparator and compared with the preset code to be tested. The comparison results are then fed back to the chip through the Compr pin. The digital

comparison and the preset codes can be easily implemented by the digital tester with software programming. The detailed description of the test procedure is in section 3.4.

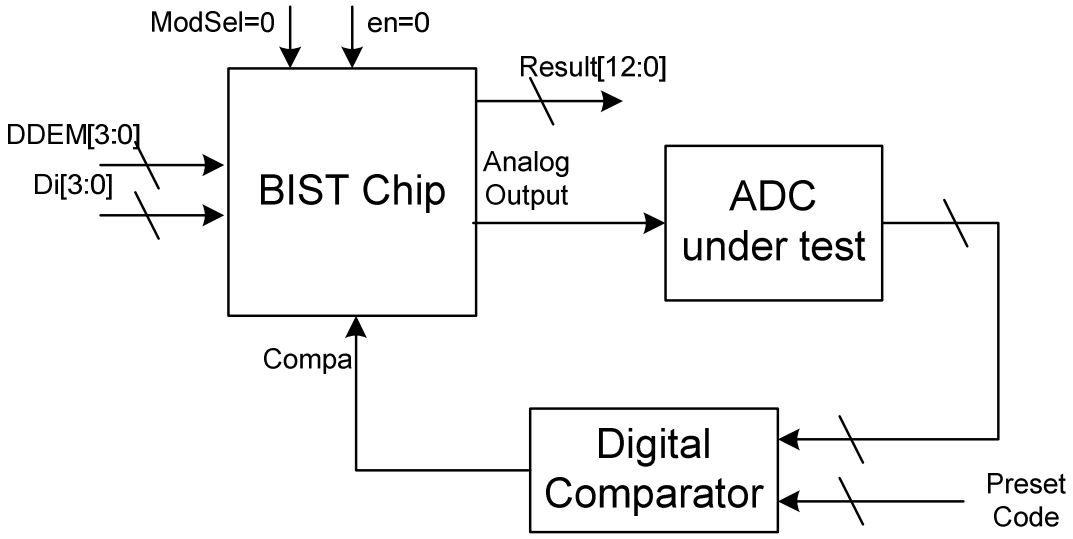


Figure 3.11. Digitizer mode setup

3.5.3 Logic Circuit Design

To implement the different operation modes introduced above, we designed the state machine for the control logic block as shown in Figure 3.12. The control block in this design is implemented with Verilog and digital synthesization. The Verilog code of the logic control block is listed in Appendix A.

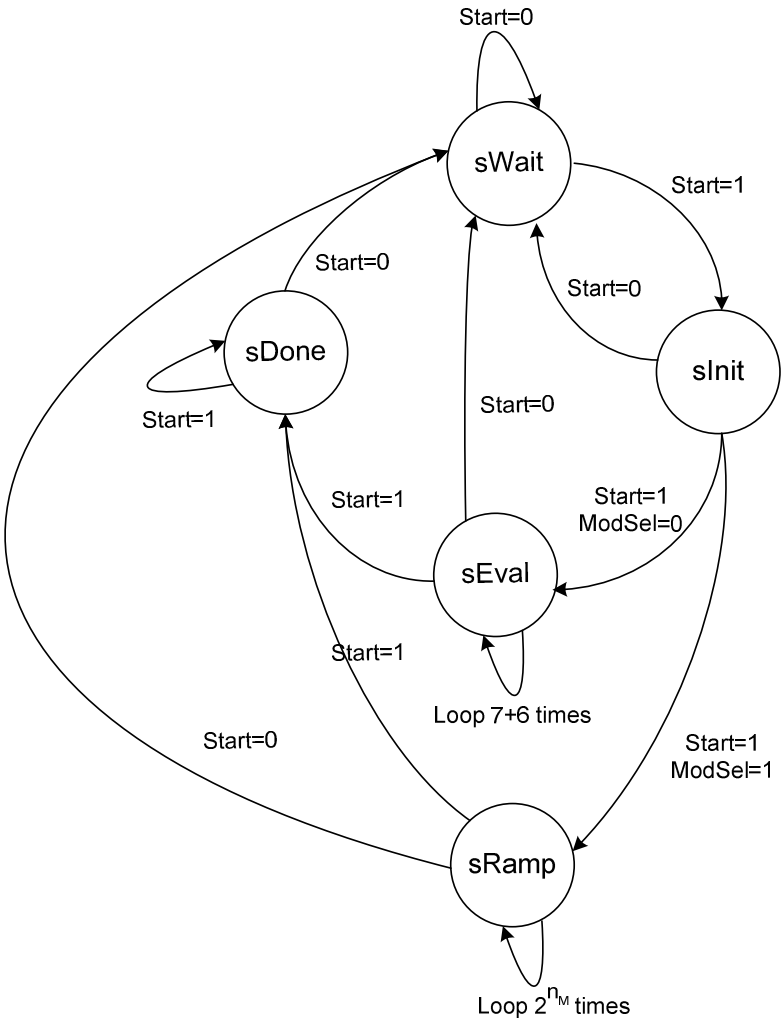


Figure 3.12. State machine design of control block

3.5.4 DDEM DAC and Dither DAC Design

The segmented DDEM DAC has a 7-bit MSB array with 4-bit DDEM control and a 6-bit LSB array. The dither DAC is 4-bit resolution. The block diagram of the MSB array with DDEM control is shown in Figure 3.13. A 3-bit row decoder and a 4-bit column decoder convert the binary MSB input codes to the thermometer codes for row and column selection. The DDEM mux block cyclically shifts the 4-bit thermometer codes according to the 4-bit

DDEM control codes. The 4-bit column decoder and the 4-bit DDEM mux block are implemented with Verilog and digital synthesization together in this design.

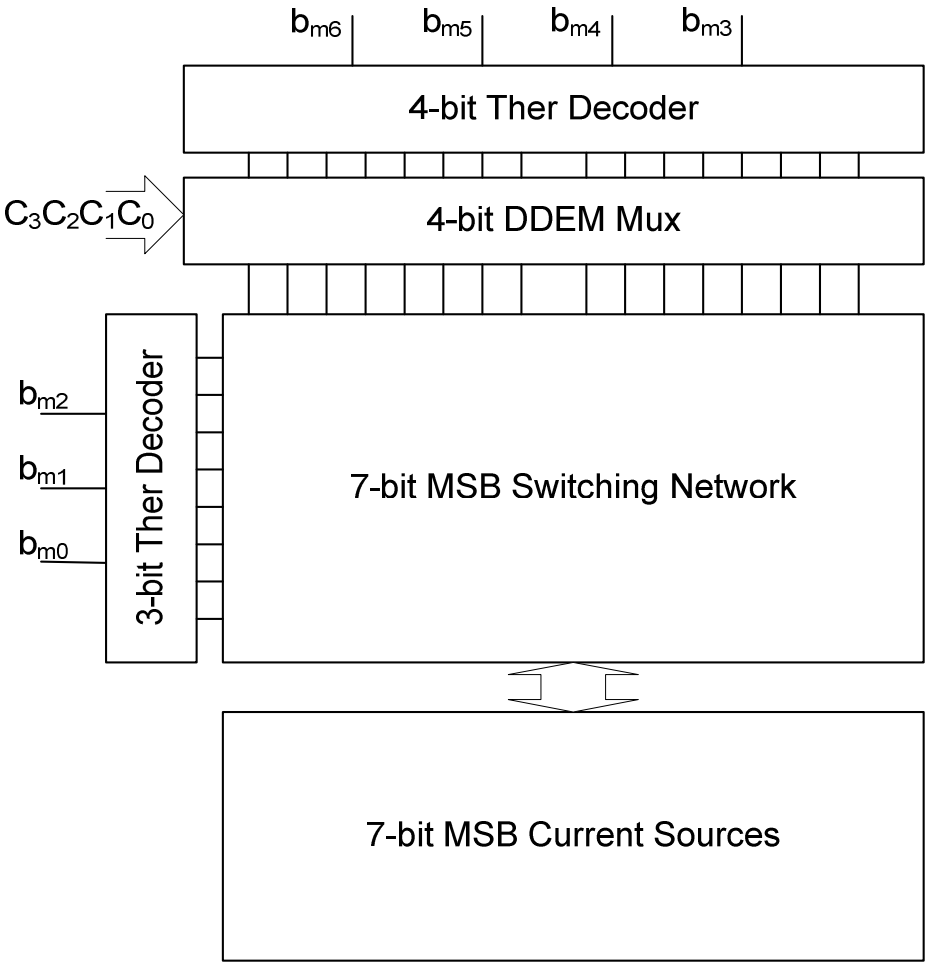


Figure 3.13. Floor plan of segmented DDEM DAC

The MSB array includes 128 unit current cells. With a DDEM algorithm, MSB array current cells allow considerable mismatch errors. The required original linearity performance of MSB array is low and at 7-bit level, as shown in the following section. Based on the matching property of the process obtained from previous fabrication, the size of the current source transistors are determined as in Figure 3.14 to achieve approximately 7-bit linearity. The current cells are cascoded to increase their output impedance, so that the output

nonlinearity due to finite output impedance can be reduced. Figure 3.15 plots the nonlinearity error due to the output impedance simulated in Spectre. The nonlinear error due to output impedance is under 15-bit level, which is negligible in this testing application.

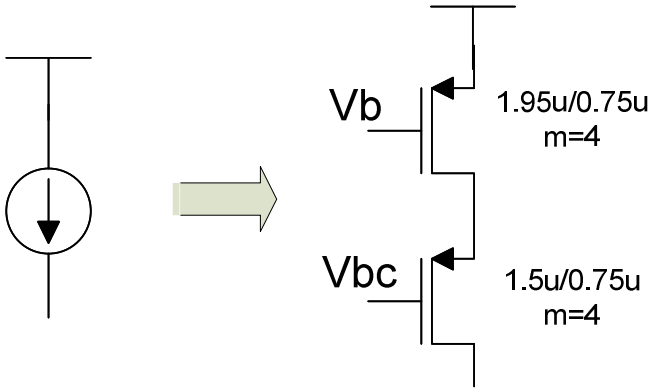


Figure 3.14. MSB array unit current source design

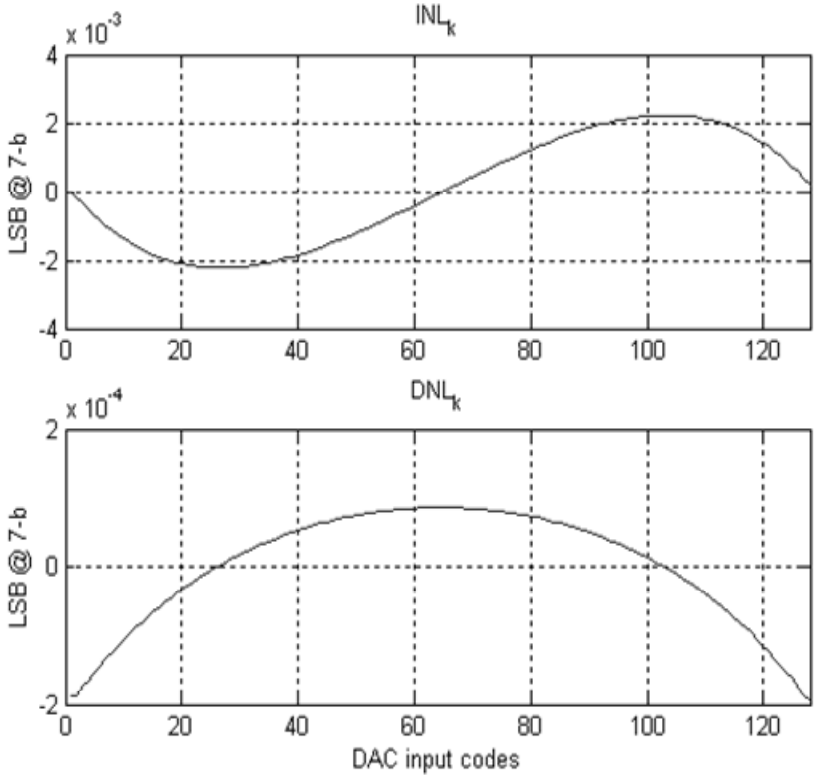


Figure 3.15. Spectre simulated MSB array nonlinear error due to finite output impedance of unit current sources

The binary-coded LSB array and the dither DAC can also be designed using the determined unit current sources with different connections. The current sources of the LSB array and the dither DAC for different bits are shown in Figure 3.16 and 3.17, respectively. For simplicity and clarity, in Figure 3.16 and Figure 3.17, only current source transistors are drawn and the cascode transistors are not shown. The full range of the LSB array is $2LSB_M$. Therefore, the current source for the first bit of the LSB array, b_5, is the same as one unit current source of MSB array. The full range of the dither DAC is $8LSB_M$. So, the current source for the first bit of the dither DAC, b_3, consists of 4 unit current sources of MSB array. The binary-coded LSB array and dither DAC do not need a decoder for input code conversion.

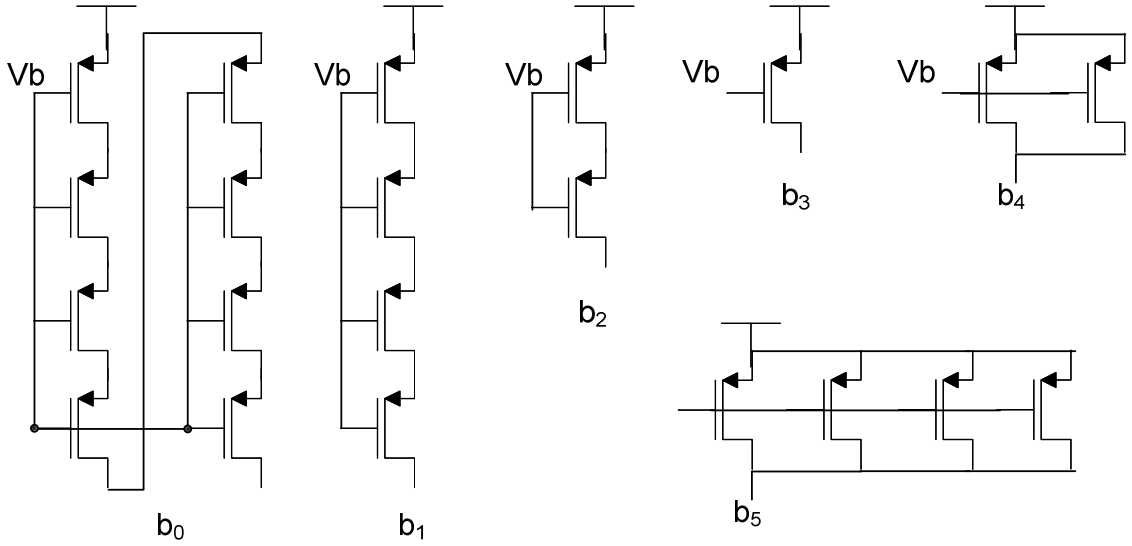


Figure 3.16. LSB array current source design

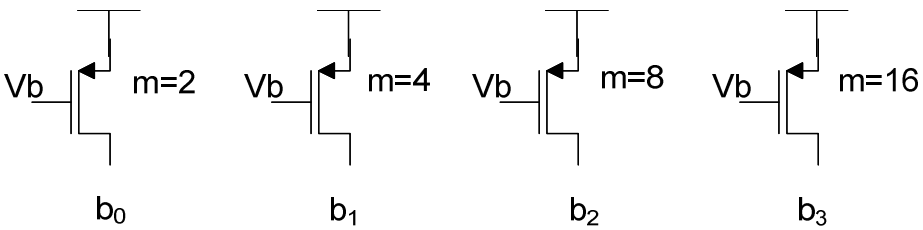


Figure 3.17. Dither DAC current source design

The simple switching circuits without latches, as shown in Figure 3.18, are used in this design. Two inverters are connected in serial at the bit input to reduce the clock feedthrough. When *clk* signal is high, the digital code D and Db control the switching of the current between branch Iop and branch Ion. Vh and Vl are the gate voltages of the switches, and proper selection of the voltage levels of Vh and Vl can improve the switching speed and the DAC settling. In this design, Vh and Vl are selected to be positive and negative power supplies respectively. Spectre simulation shows satisfactory speed performance.

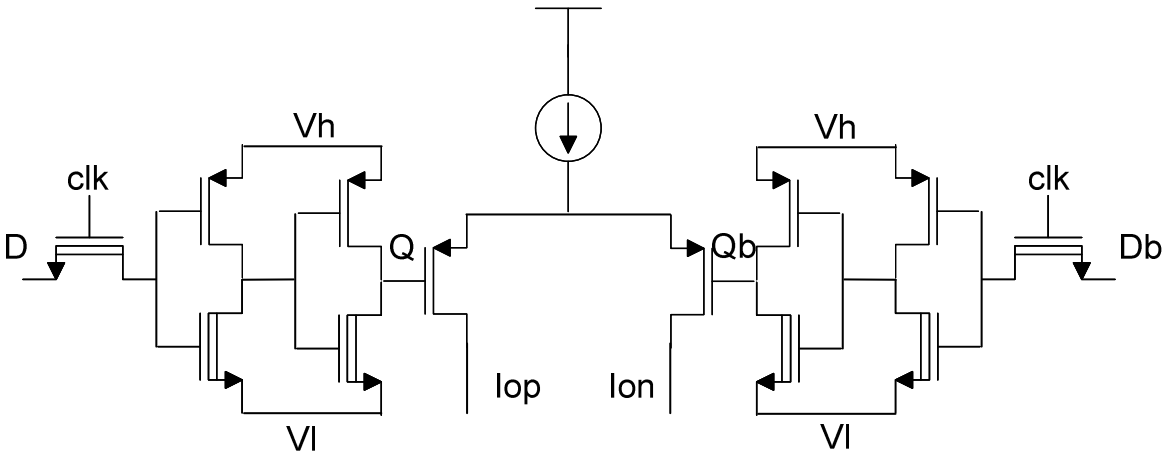


Figure 3.18. LSB array current source design

Comparator design

The structure of the on-chip comparator is illustrated in Figure 3.19. The upper half of the figure shows the bias and the pre-amp, and the lower half shows the latch structure and the output buffers. The accuracy of the comparator is not very critical to the test performance. For example, the input offset of the comparator will induce a constant offset in the digital measurements, which does not affect the linearity performance testing. The input-referred noise of the comparator will affect the testing. However, measuring one external reference

voltage requires repeating the comparison many times. The effective noise effect is then easily averaged down to a negligible level.

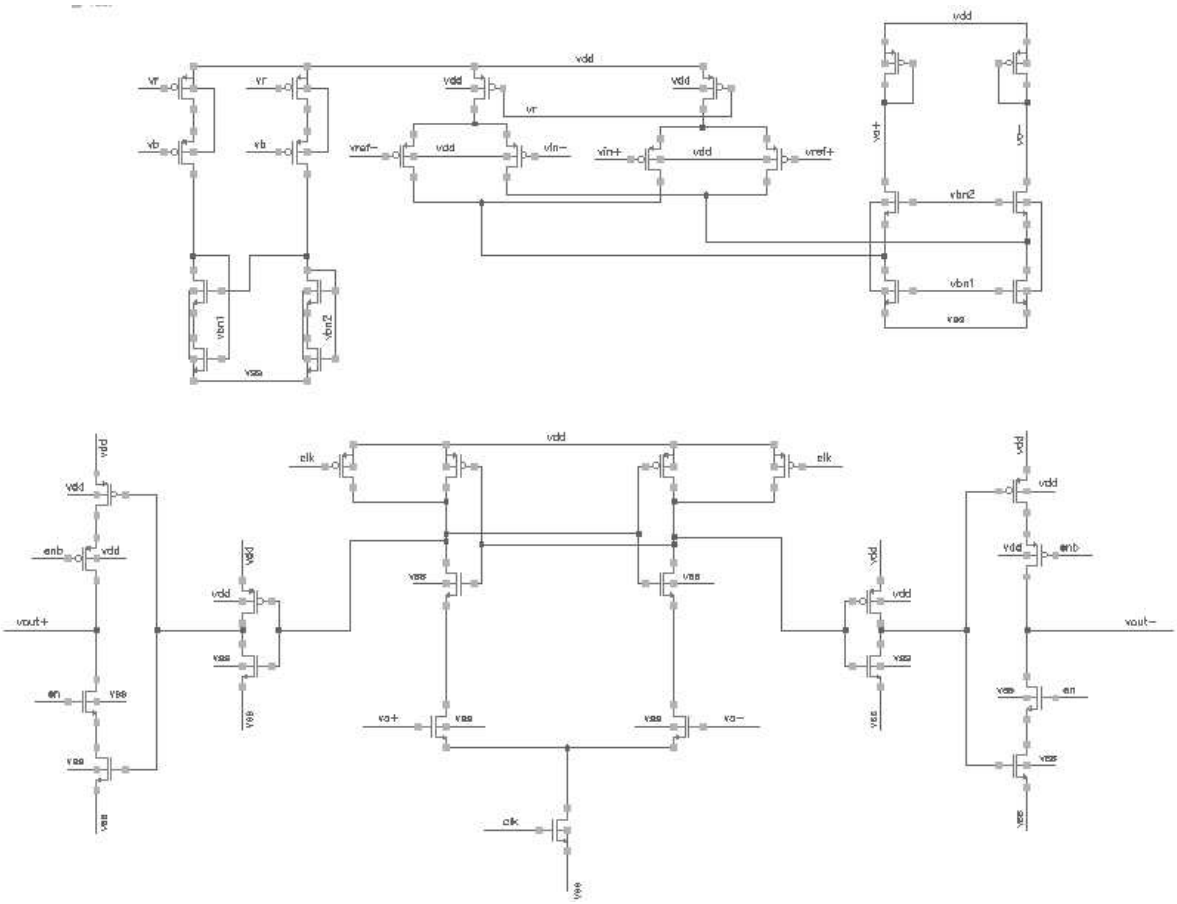


Figure 3.19. LSB array current source design

The comparator makes decision at the rising edges of signal *clk.* When clk is low, the latch is reset. The comparator output can be disabled by setting signal en to low.

Operation shown in Spectre simulation

The output ramp signal in the ramp generation mode is plot in Figure 20. The clock frequency in simulation is 100MHz. The binary search procedure in the digitizer mode is shown in Figure 21. The simulated clock frequency is 50MHz.

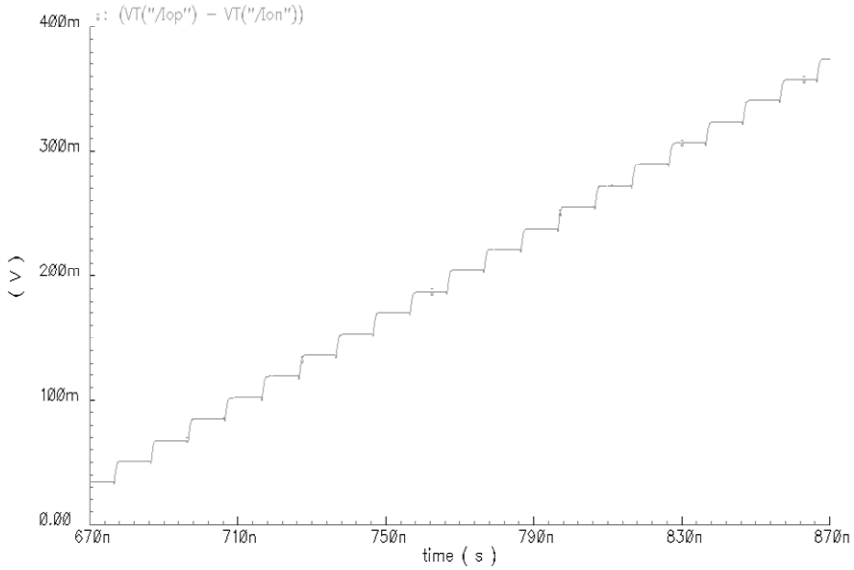


Figure 3.20. Output ramp signal in ramp generation mode

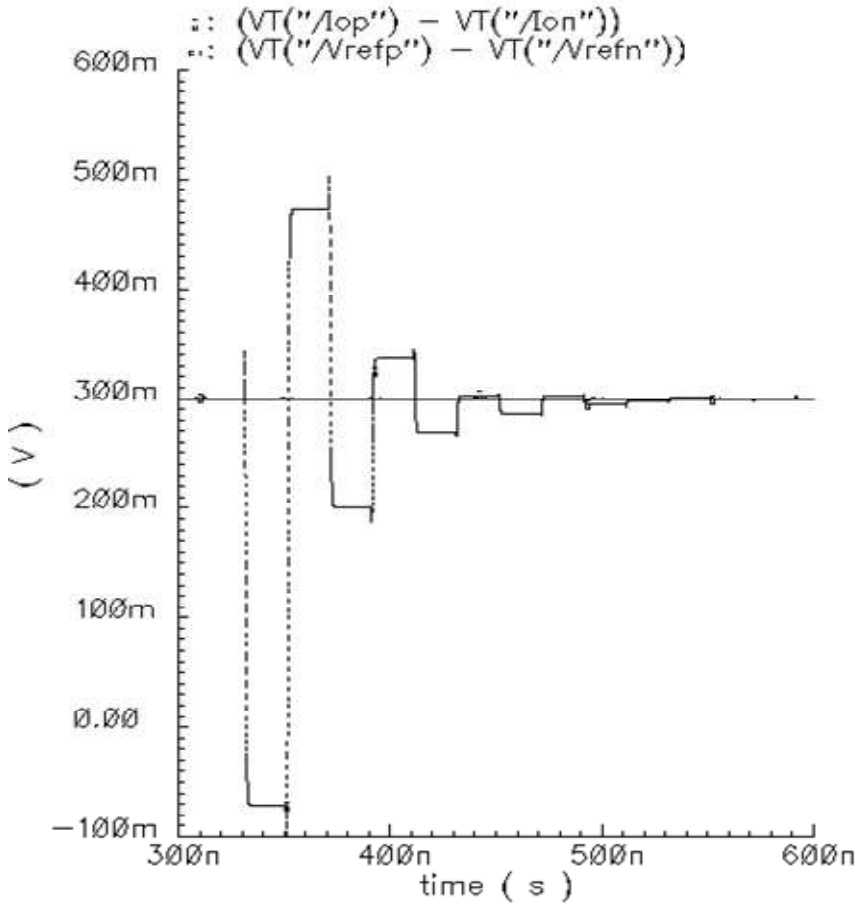


Figure 3.21. Binary search procedure in digitizer mode

Layout

Figure 2.2 shows the layout of the chip. All the current sources are put together and surrounded by guarding rings to isolate substrate noise from logic and switching circuits. In the analysis, we mentioned that the full-scale range of the LSB array is not important, but the full-scale range of the dither DAC is. To make sure the accuracy of the dither DAC output range, we put the current sources of the MSB array and the current sources of the dither DAC together following a common-centroid pattern.

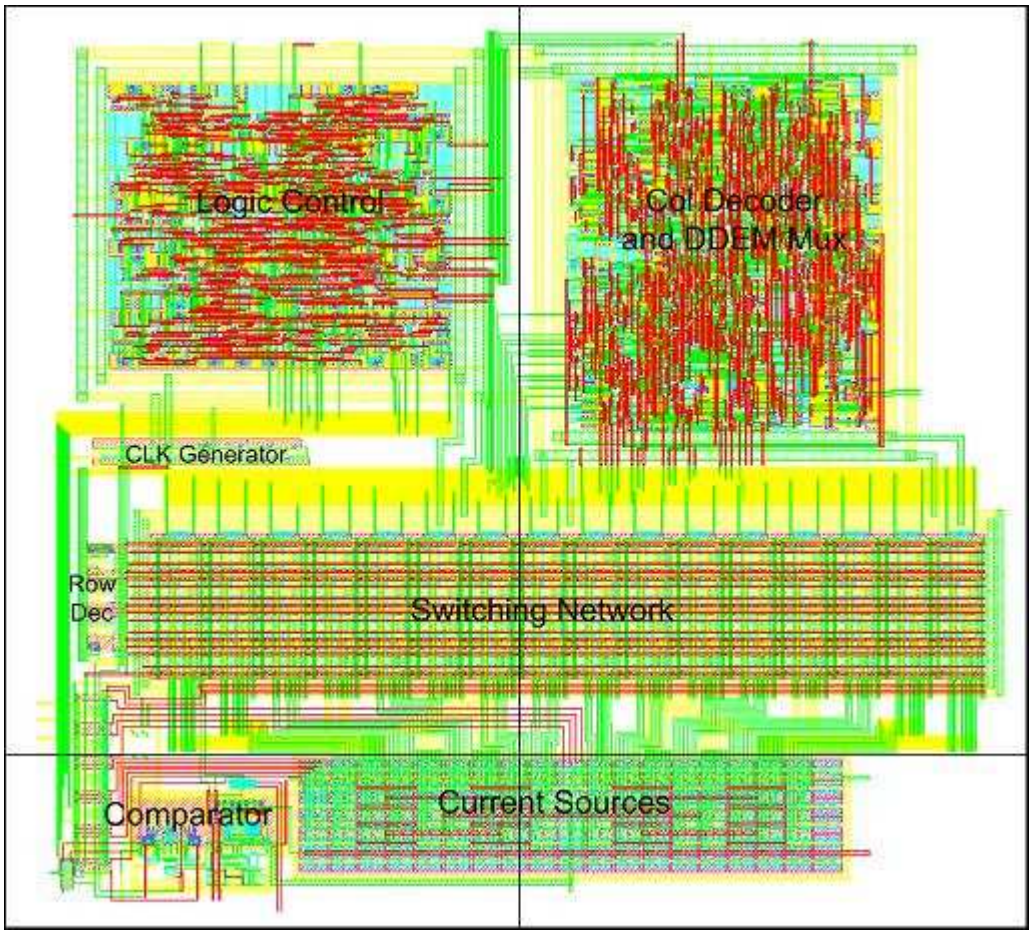


Figure 3.22. Layout of the prototype chip

The goal of this source generator design is to provide the high test performance of the analog BIST circuit without doing any measurement and calibration. The DDEM algorithm

and the proposed test algorithm guarantee this part, which has been theoretically proved. In reality, the test performance of some implementations will possibly be not enough for testing. As long as only a very small portion of the parts have this problem, this situation is still tolerable. That is because it is almost unlikely that we can test a bad part as a good one using a bad test system. So, we will more likely to have false bad parts but not false good parts that cause more severe consequence. In the other case, it is possible that the DAC has some failure deficiencies in the digital part that cause the source generator not functional. Some simple functional tests can be done with the external digital tester to make sure this situation is taken care of.

3.6 Simulation Results

Numerical simulations validate the testing performance of the stimulus generator and the proposed testing procedure. In the simulation, the segmented DDEM DAC has a 7-bit MSB array and a 6-bit LSB array. The full-scale range of the LSB array is equivalent to $2LSB_M$s for error compensation, where LSB_M is the ideal MSB array's least significant bit. The number of the configurations, P, is 16. Thus, we have a 4-bit DDEM control code for configuration selection. The dither DAC has 4-bit resolution and a full-scale range of $8LSB_M$s, which is equal to N_M/P, as explained in section 3. The current sources in the simulation are modeled by a nominal current with a random Gaussian-distributed mismatch error. Figure 3.23 shows the linearity performance of the original MSB array. The *INL* is about $0.52LSB$, and, consequently, the linearity of the MSB array is at only 7-bit level. The LSB array and the dither DAC in the simulation are both approximately 6-bit level linear.

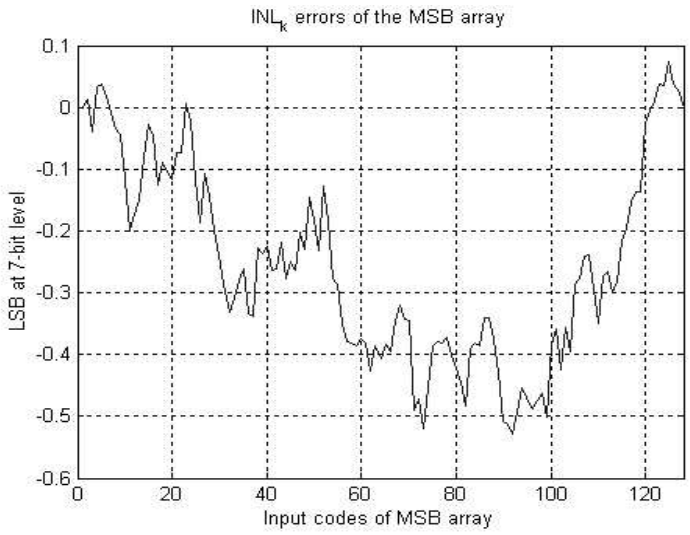


Figure 3.23. *INL$_k$* error of the MSB array

From the analysis of the testing performance, we can calculate the equivalent test performance in bits as

$$\begin{aligned} n_{eq} &\approx \min\{n_{DAC}+0.5(\log_2 P+n_d), ENOB_{DAC}+\log_2 P+n_d\} \\ &\approx \min\{12+0.5\times(4+4), 7+4+4\} \\ &= \min\{16,15\}=15 bits \end{aligned} \quad . \qquad (3.17)$$

The equivalent testing performance of the stimulus generator is at about 15-bit level. In the simulation, a 12-bit ADC is under test. The integral nonlinearity errors of the simulated ADC are shown at the top of Figure 3.24. A white-noise signal is added to the input samples of the ADC to create the noise effect of a practical ADC, and the standard deviation of the additive noise is set to be 0.25LSB at 12-bit level. The measuring of the transition levels follows the test procedure described in section 3.4. The INL_k information of each transition level is then calculated from the measurements. Figure 3.24 shows the true and the estimated INL_k errors of the ADC under test along with the estimation errors, which is the difference between the first two plots. From the simulation, the INL_k estimation errors are bounded by about

±0.15LSB. Therefore, the testing performance of the stimulus source is evaluated to be at about 15-bit level, which verifies the theoretical analysis in (3.17).

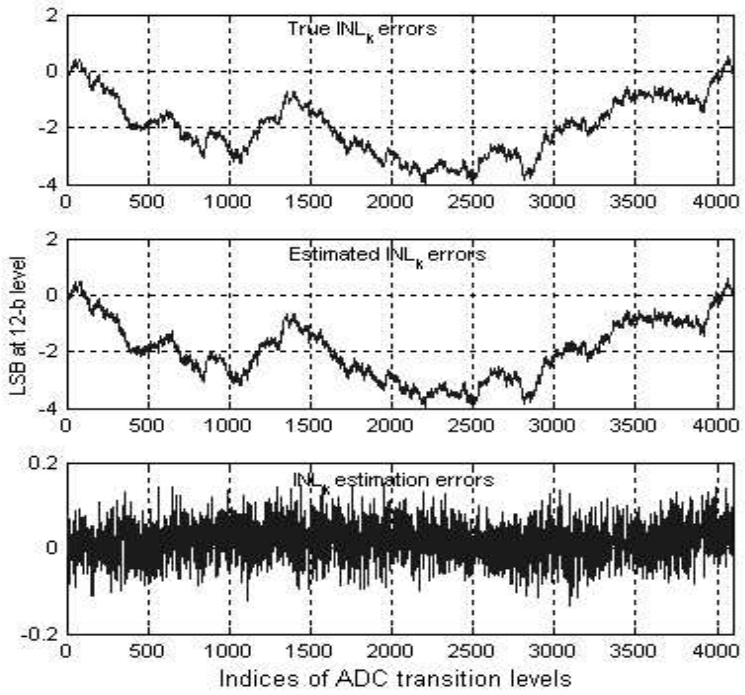
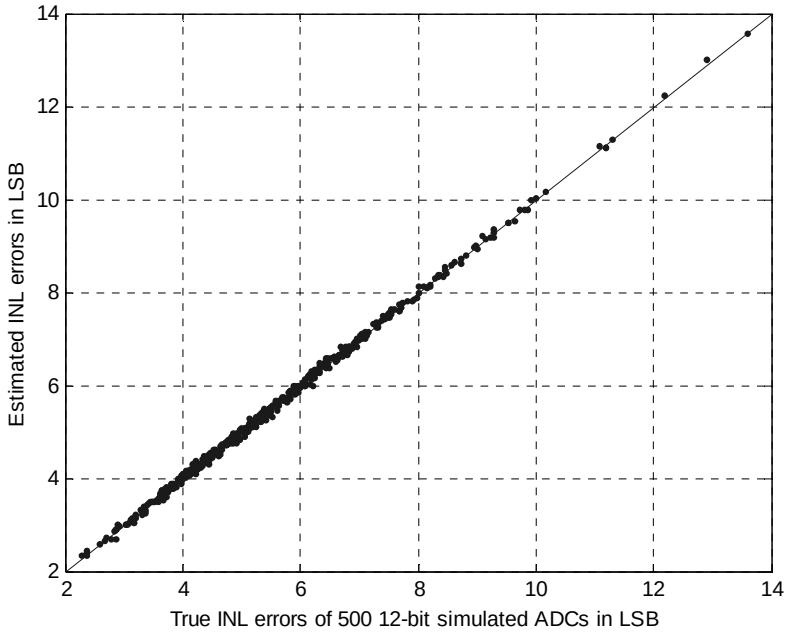
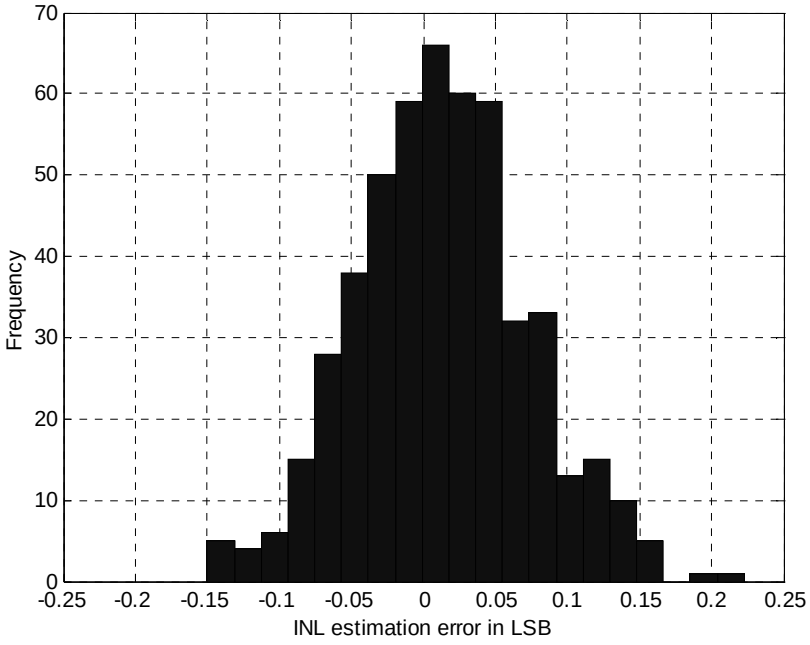


Figure 3.24. INL_k estimation errors of ADC under test

To validate the robustness of the method to the random mismatch errors, 500 different 12-bit ADCs are tested by 500 test systems with different mismatch errors in the simulation. The configurations and accuracy levels of the test systems are the same as those in the previous simulation. The results are shown in Figure 3.25. In Figure 3.25(a), each dot in the figure represents one testing result and the true *INL* and the estimated *INL* of the ADCs are represented by the coordinates of the dot. The results show that the *INL* of the 500 ADCs varies from 2*LSB* to about 14*LSB* and the *INL* estimation errors are from about -0.15*LSB* to 0.2*LSB*. The testing performance is very robust for the different implementations with different errors.



(a)



(b)

Figure 3.25. (a) Estimated and true INL errors of 00 12-b ADCs; (b) INL estimation error histogram

Since the mismatch errors in fabrication can be modeled as random variables in analysis, the INL_k and INL estimation errors of the BIST system are random too. It is important to theoretically express the distribution of INL estimation errors since the distribution gives confidence information about the estimation accuracy of the system. However, it is extremely difficult to theoretically obtain the expression of the distribution. Even the distribution of the INL itself is hard to express. There is no accepted solution in both industry and academy. Fortunately, the distribution can be numerically approached by the histogram shown in Figure 3.25(b). Assume the distribution of the INL estimation error follows normal distribution, it is straightforward to approximate the distribution of the INL estimation errors by getting the mean and the standard deviation of the histogram, which are $0.0130LSB$ and $0.0597LSB$ respectively. Furthermore, since the BIST system estimates the individual transition levels, for each ADC estimation done in this section, we actually have about 4000 measurements. The histogram of the INL_k estimation error can be obtained. This histogram shows the approximated distribution of the INL_k estimation errors for one specific implementation.

Analysis shows that the test performance can be improved by increasing several parameters, such as the number of configurations P and the resolution of the dither DAC n_d. In another simulation, P is increased while all the other system setups are the same as in the previous simulations. The same MSB array as that shown in Figure 3.23 is used. Figure 3.26 illustrates the reduction in the INL_k estimation errors of a 14-bit ADC with increasing P. The standard deviation of the noise in this simulation is set to be $0.25LSB$ at 14-bit level. When P is increased, the test performance is improved along with the cost of test time. However, we should also notice that the test time is proportional to the number of the reconfigurations.

When testing high-resolution ADCs, the number of the configurations needs to be large enough to guarantee the desired test performance. In this case, the test time could be significantly long. Fortunately, the proposed test system and procedure can measure ADCs' transition levels one by one. That enables the use of the reduced-code testing method, which characterizes the linearity performance by measuring a small set of ADCs' transition levels so that the test time can be greatly reduced. More will be discussed in chapter 5.

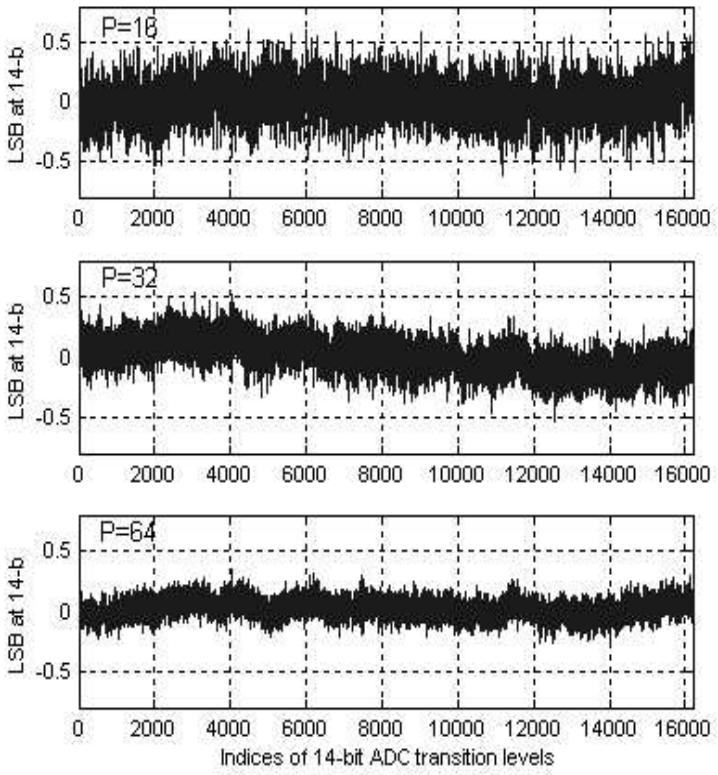


Figure 3.26. INL_k estimation errors of a 14-b ADC with P=16, 32 and 64

3.7 Experimental Results

3.7.1 Test Performance Verification

The experimental results shown in this section are measured from the fabricated thermometer-coded DDEM DAC in a 0.5um CMOS process [30]. The chip includes 4096 unit current cells designed with all minimal-size transistors and the DDEM logic control circuits. The DAC's resolution and the DDEM control parameter *P* are both programmable and can be up to 12 bits and 512. Although the Spectre simulation shows that the stimulus frequency is up to 100MHz, the data are measured at 1MHz because of the speed limitation of the high-precision digitizer used in the data acquisition. The differential outputs of a 7-bit DDEM MSB array with *P*=16 are measured, as shown in Figure 3.27. Each ramp in the figure shows the output characteristics of the MSB array under one specific DDEM control code. Linearity of the original MSB array is at about 9-bit level as shown in Figure 3.28.

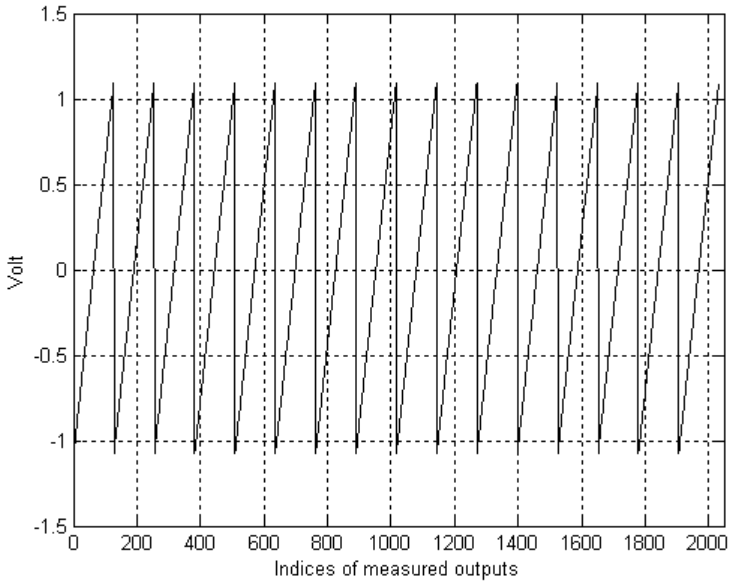


Figure 3.27. Measured differential outputs of the DDEM MSB array

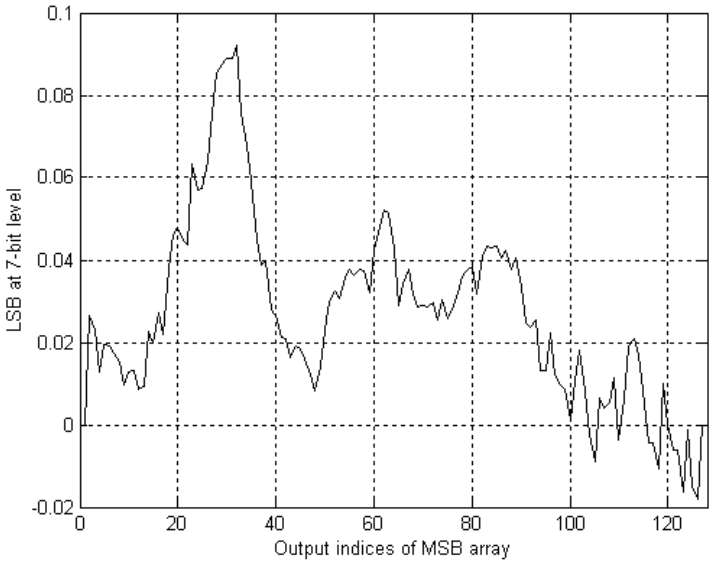


Figure 3.28. Measured INL_k errors of the DDEM MSB array

Similarly, output levels of the 6-bit LSB array and the 4-bit dither DAC are measured from the chip. Both of them are tested to be less than 8-bit linear. A simulated 12-bit ADC is also tested using the measurements of the DAC outputs to verify the testing performance of the measured data. Figure 3.29 shows the true and the estimated INL_k errors along with the estimation errors. The results prove the testing performance of the proposed source generator and the testing procedure. It is noted that although the linearity of the MSB array is 2-bit better than in the simulation section, the testing performance is still at 15-16 bits level because of the error from the resolution limitation, as in (3.16).

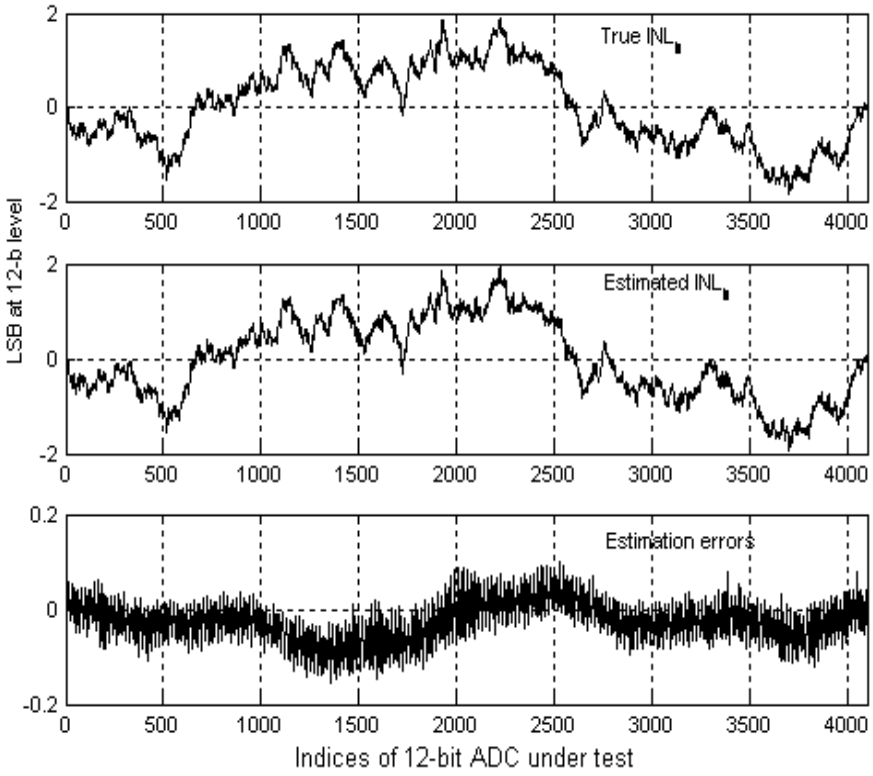


Figure 3.29. Testing results of the simulated 12-b ADC using measured data

3.7.2 BIST Structure and Procedure Verification

Experimental results shown in this section are measured from the latest fabricated prototype chip described in section 3.5. The chip is measured by mixed-signal ATE tester, Teradyne J750 test system. All data is measured at low speed, which is 50 kHz, for accurate characterization.

Firstly, the linearity characteristics of the MSB array, the LSB array and the dither DAC can be measured from the chip operating in the ramp generation mode. Figure 3.30 shows the linearity of the original MSB array without DDEM configuration. The *INL* of the original MSB array is 0.5*LSB*.Therefore, it is proved that the original MSB array is about 7-bit linear as targeted during the design phase.

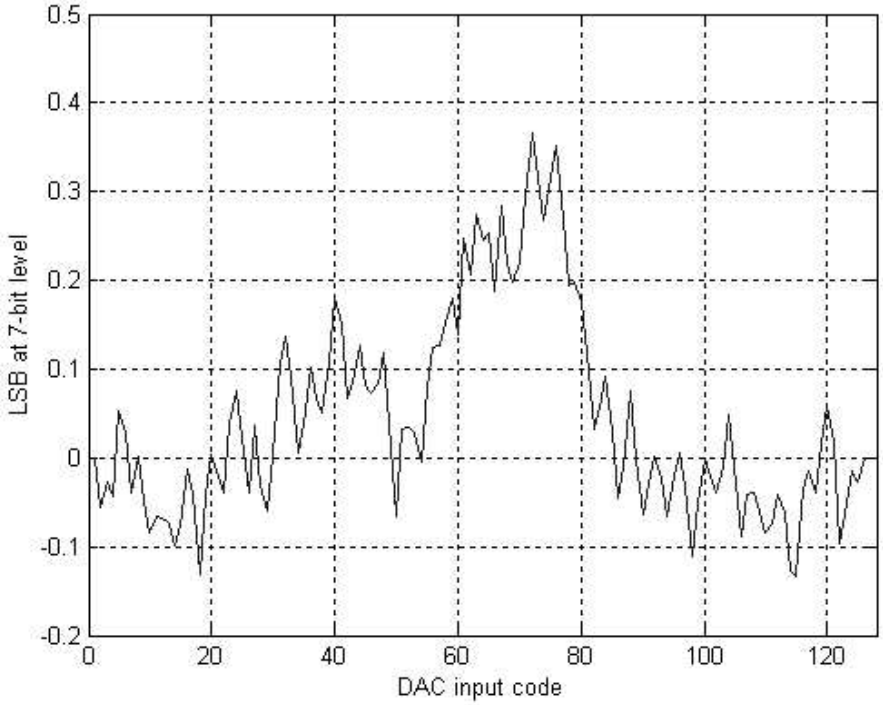


Figure 3.30 INL_k of the original MSB array

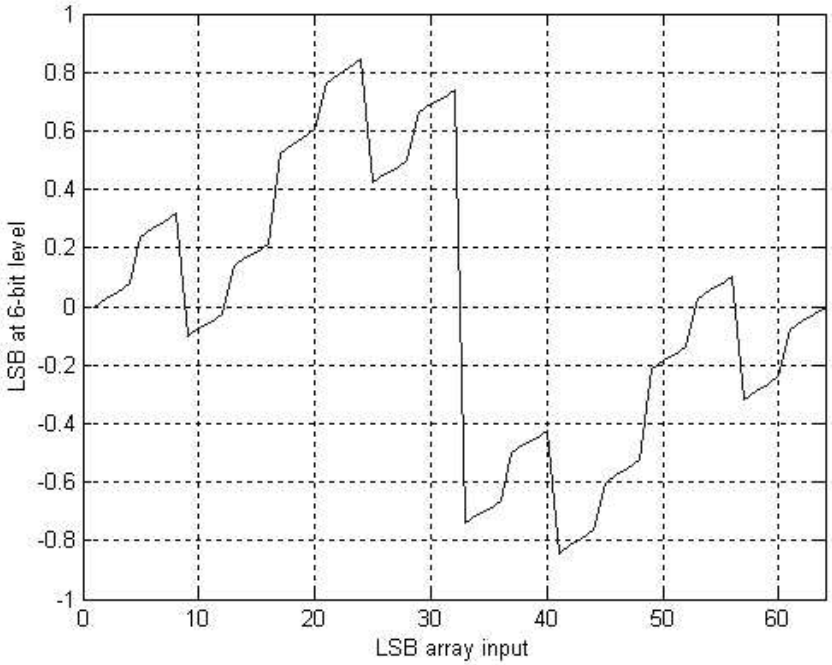


Figure 3.31 INL_k of the LSB array

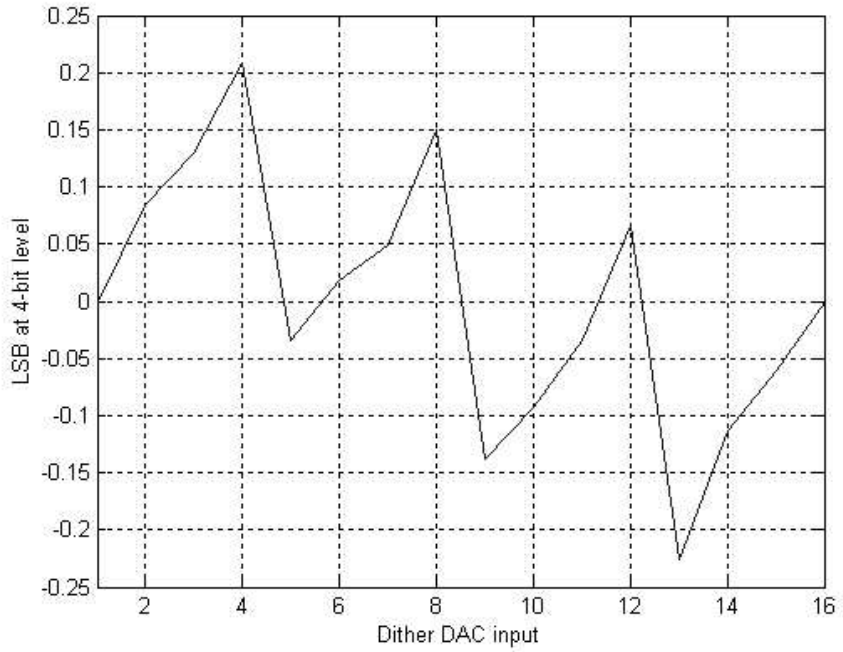


Figure 3.32 INL_k of the dither DAC

The ramps under different configurations, which have different DDEM control codes or different dither DAC inputs, can be measured. Results verify that the DDEM shift and the dither DAC work properly. The output integral nonlinearities of the LSB array and the dither DAC are measured and shown in Figure 3.31 and Figure 3.32 respectively. Both of them have about 5-bit output linearity. As we discussed in the previous sections, the full-scale range (FSR) of the dither DAC is critical to the test performance. The desired FSR of the dither DAC is 8 LSB_M. The real FSR of the dither DAC is measured to be 2 percent off, which is proved to be tolerable in the numerical simulation. The FSR of the LSB array is desired to be 2LSBM. The accuracy is not important to the test performance. The measurement shows that the FSR of the LSB array is about 10 percent larger than the desired value. All the dither DAC and LSB array outputs are measured in the case that the differential output of the MSB array is close to zero. The FSRs of the dither DAC and the

LSB array change with the MSB array differential outputs slightly. It has been proved in the simulation that these small changes do not degrade the test performance.

Like in section 3.7.1, we can verify the test performance of the prototype chip with an external emulated ideal ADC, which is virtually implemented by the high-precision tester. In addition, we can easily use the resources in the test to close a feedback loop to execute binary searches for desired DAC input codes under total 256 configurations. Assume the ADC under test has 12-bit resolution and ideal transition levels with certain noise level. The test error of the system can then be evaluated. Figure 3.33 shows the estimation error of the test system used to measure 12-bit ADCs. The noise level is set to be one fourth of *LSB* at 12-bit level.

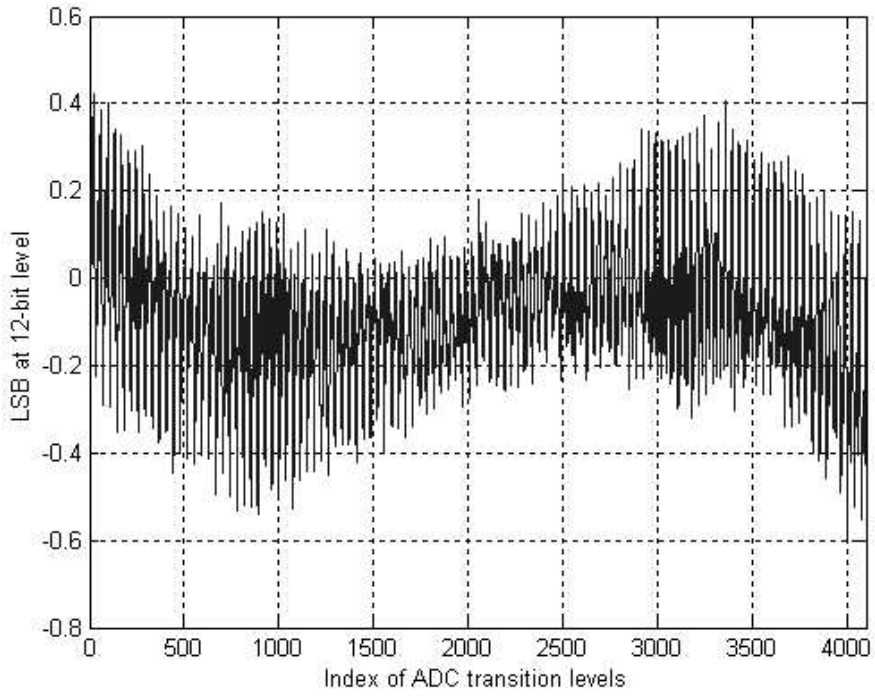


Figure 3.33 INL_k estimation errors of the measured test system, 12-bit ADC, 0.25*LSB* noise power

As shown in Figure 3.33, the INL_k estimation errors are under 0.5 *LSB* at 12 bit level. Thus, the test system designed is capable of testing 12-bit ADCs to 0.5 *LSB* accuracy. It is noticed that there is a low-frequency wave-like shape in the estimation errors. This error shape can be verified to be due to the finite output impedance of unit current sources. The same effect seen in the Spectre simulation is shown in Figure 3.15. The estimation errors shown in the experimental results are about one bit larger than that shown in the simulation results. The reason of it is possibly the FSR error of the dither DAC.

3.8 Conclusions

Using a digital tester with BIST for analog and mixed-signal testing is an ideal way of reducing testing costs and improving test quality. However, the traditional high-performance circuits and testing solutions are too costly and complicated to be built on-chip just for testing. Instead, a BIST strategy for ADC linearity testing may be used, which is fully compatible with digital test environments using a low-cost digital tester and a simple digital DIB. Low-resolution and low-accuracy DACs (which are cost efficient) are built on-chip as source generators. The testing performance is guaranteed by the DDEM reconfiguration technique and the testing procedure. Design of the on-chip testing circuits could be as easy as digital design because of the low accuracy requirements on the analog blocks. Simulation and experimental results demonstrate that the proposed strategy is able to test the INL_k error of 12-bit ADCs to ±0.15*LSB* accuracy level using very low-accuracy DACs. In addition, the BIST strategy can be easily adopted for DAC testing if the digital comparator is replaced by an analog one, which compares the outputs of the DAC under test with the outputs of the source generator.

Chapter 4. System Identification-Based Full-Code Linearity Characterization of Pipeline ADCs Using Reduced-Code Testing Method

This chapter presents a system identification based test method for characterizing pipeline ADCs' full-code static linearity performance, which can be a simple application of the BIST system discussed in the previous chapter. To reduce test time, only a small set of codes are measured to identify ADCs' characteristics. In this work, the pipeline ADC under test is identified by characterizing the two most critical parameters in each stage, the stage gain and the comparator offset. The transfer function is investigated to obtain the effects of the gain error and comparator offset on ADC's linearity performance. With the measurements of a small set of specific transition levels or code-bin widths, the system parameters of interest can be calculated using only straightforward linear calculations. The identified model is then used to compute the ADC's full-code linearity information. Compared to standard histogram-based, full-code linearity test methods, the proposed method can reduce the data capture time by more than 90% without appreciably degrading the testing accuracy. Both simulation results and experimental results are included to demonstrate the efficacy of the proposed method.

4.1 Introduction

As the interface between analog and digital signals, analog-to-digital converters (ADC) are one of the most extensively used analog and mixed-signal (AMS) circuitries [36]. The performance of ADCs directly determines the accuracy performance of the whole system, so ADC testing is indispensable for almost all systems to validate the design and to reduce the wasted parts. In particular, linearity testing is critically important to many

applications in signal processing, communications, instrumentation, and other areas. To satisfy the demands of new applications, the performance of ADCs improves continuously, and testing of ADCs' linearity performance becomes increasingly challenging [37].

Full-code testing of high-resolution ADCs is not only challenging but expensive. Conventional testing methods like the histogram test need highly linear source signals, which are usually generated by expensive mixed-signal automated test equipment (ATE) [38]. In this case, testing costs are mainly from the use of ATE testers and proportional to the test time. The full-code test time of high-resolution ADCs is significantly long, since the number of transition levels under test increase exponentially with ADCs' resolution, and, furthermore, such ADCs usually operate at low sampling frequencies. As a result, the cost of testing ADCs' full-code linearity performance also increases rapidly with resolution and becomes more and more prohibitive. To cut down the test time, sometimes reduced-code testing is used instead of full-code testing to determine ADCs' performance by measuring only a small subset of the ADC output codes [39]. Commonly used reduced-code testing methods select a small group of transition levels for testing based on experience. In this case, only the small set of transition levels are guaranteed, and no full-code performance information is available. The selection of the transition levels for testing is very difficult and there is no universally-accepted method of selection. One alternative method to reduced-code testing is the system-identification based approach. In this approach, by measuring a small set of transition levels, the ADC parameters of interest are identified. Then, the full-code nonlinearity performance can be constructed by those parameters and the predefined model. This work presents a system identification-based reduced-code testing method for testing the linearity of pipeline ADCs.

In this chapter, we present a system identification-based test method for testing static linearity performance of pipeline ADCs, which are widely used because of their attractive combination of speed, resolution, low power dissipation, and low cost. Unlike the method proposed in [39] using nonlinear modeling and iterations, the proposed method uses only straightforward linear calculations in the system identification. The rest of the chapter is organized as follows. Section 4.2 discusses the pipeline ADC characteristics, such as the structure and the transfer functions. In section 4.3, the proposed method is presented and theoretically analyzed, and the simulation results are shown in section 4.4 to verify the correctness of the algorithm. Section 4.5 discusses the test system setup for measuring the critical transition levels or code-bin widths, while the experimental results are illustrated in section 4.6 to demonstrate the efficacy of the proposed method. Finally, section 4.7 concludes the chapter.

4.2 Characteristics of Pipeline ADC

The general architecture of a pipeline ADC is shown in Figure 4.1. A pipeline converter usually includes a front-end sample-and-hold amplifier followed by a cascade of identical stages, each of which executes one or more bits conversion from MSB to LSB sequentially and generates a residue voltage for the following stages. Some digital logic circuits are included for output time alignment and digital correction. In each stage, the input voltage from either the SHA or the previous stage is quantized by a low-resolution sub-ADC to generate the stage digital output while being sampled. The difference between the sampled input and the sub-DAC output is amplified by a gain stage to generate the residue for the next stage. In this work, we will focus on 1-bit/stage pipeline ADC, which uses a comparator as

the sub-ADC, a 1-bit sub-DAC, which generates its output according to the comparator output, and a gain-of-two amplification stage. Then, the digital output of stage *k* is given by

$$d_k = \begin{cases} 0 & ,V_{k-1} \le V_{thr,k} \\ 1 & ,V_{k-1} > V_{thr,k} \end{cases}, \tag{4.1}$$

where V_{k-1} is the input signal from the previous stage and $V_{thr,k}$ is the comparator threshold voltage of stage *k*-1. For the 1-bit/stage pipeline ADC, the ideal comparator threshold voltage of each stage is 0.

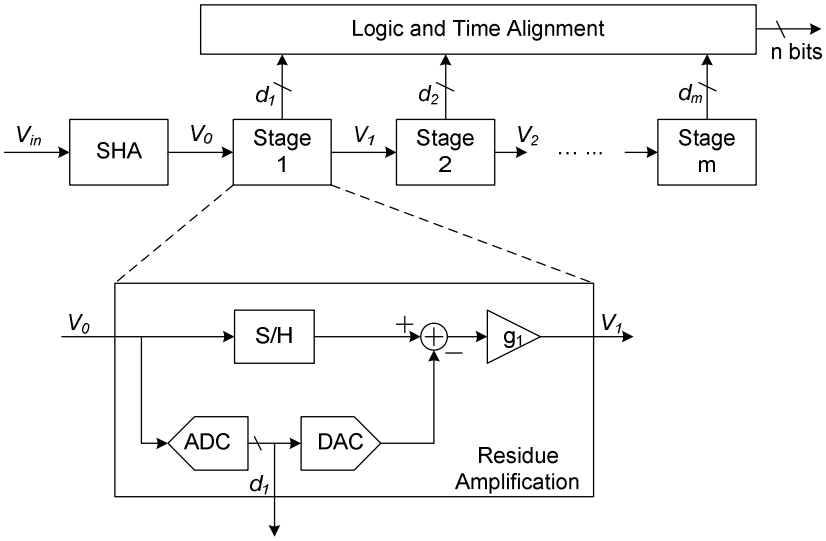


Figure 4.1. Block diagram of an *n*-bit Pipeline ADC

The sample and hold, the subtractor, and the gain stage are usually implemented together using a switched-capacitor amplifier. Figure 4.2 shows the switched capacitor model of the gain stage *k*. The differential signals V_{k-1} and V_k are the input signal from stage *k*-1 and the output signal to stage *k*+1, respectively. Moreover, the differential DAC output V_{DAC} is equal to $-V_{ref}$ when d_k=0 or V_{ref} when d_k=1. $\pm V_{ref}$ are the differential reference voltages of the ADC, which define the valid input range of the pipeline stages. The two non-overlapping clocks, Φ_1 and Φ_2, control the capacitor switching, and the clock Φ_1' leads Φ_1 a little bit to

reduce the signal-dependent charge injection. It is easy to prove that the ideal transfer function of the gain stage is given by

$$V_k = \begin{cases} V_{k-1}\left(1+\dfrac{C_s}{C_f}\right)+V_{ref}\dfrac{C_s}{C_f}, & d_k = 0 \\ V_{k-1}\left(1+\dfrac{C_s}{C_f}\right)-V_{ref}\dfrac{C_s}{C_f}, & d_k = 1 \end{cases}. \tag{4.2}$$

With C_s=C_f in 1-bit/stage structure, the switched-capacitor amplifier has a nominal gain of 2 from the input to the output and a nominal gain of -1 from the DAC to the output.

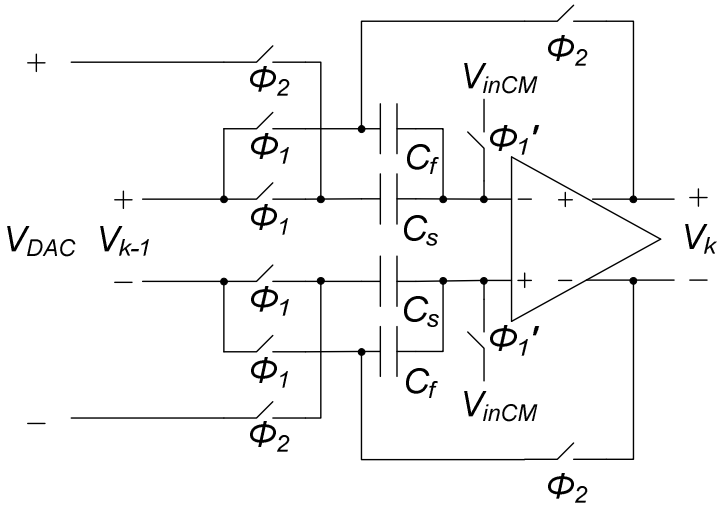


Figure 4.2. Switched capacitor model of the gain stage

The ideal transfer curve of stage *k* is shown as the dashed line in Figure 4.3. However, the real transfer curve, the solid line in Figure 4.3 as an example, is affected by the circuit non-idealities. Three major errors in the transfer curve are present. The first is the error in the comparison threshold that is induced by the comparator offset, $V_{os,k}$, and usually modeled by a random variable with zero mean. The second is the gain error shown as the slope difference between the ideal and non-ideal transfer curves in Figure 4.3, which is generated by the capacitor mismatch and the finite gain of the op amp. The third error is the

nonlinearity in the transfer curve induced by the op amp and capacitor nonlinearities. Some other non-idealities, such as the switch charge injections and the op amp input offset, introduce negligible errors in the output and are not critical to the ADC's linearity performance.

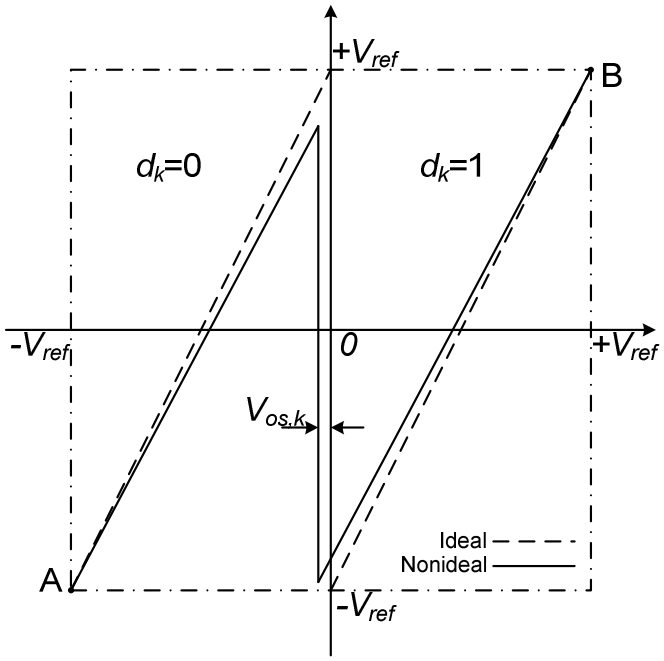


Figure 4.3, Transfer curve of the stage *k* residue amplification

4.3 System Identification-Based Linearity Testing

ADC's linearity performance is usually characterized by the integral nonlinearity (*INL*) and the differential nonlinearity (*DNL*) of the transition levels. For transition level $T(k)$, we can calculate the nonlinearities as in

$$INL(k) = \frac{T(k) + V_{ref}}{LSB} - k \text{, and} \tag{4.3}$$

$$DNL(k) = \frac{T(k) - T(k-1)}{LSB} - 1, \tag{4.4}$$

where the least significant bit (*LSB*), which is the ideal code-bin width and defined by the ideal transfer curve as

$$LSB = \frac{V_{ref} - \left(-V_{ref}\right)}{2^n} = \frac{V_{ref}}{2^{n-1}}. \tag{4.5}$$

In the system modeling, it is assumed that among the three major errors mentioned in section 4.2, the ADCs' nonlinearity is dominated by the comparator offsets and the gain errors in the pipeline stages, while the nonlinear errors in the transfer curve are negligible compared to the other two. This assumption is reasonable when the op amps are carefully designed and the high-linearity capacitors are adopted. Also, we assume that the reference voltages for different stages are the same, which is reasonable as well since all the reference voltages can be generated from one standard reference voltage. Under these two assumptions, it is easy to prove that the points A and B in Figure 4.3 are fixed regardless of the existence of the capacitor mismatch and the comparator offset. Thus, the nonlinearity induced by the stage can be evaluated by using A and B as reference points.

Assume a 1-bit/stage pipeline ADC with *n*-bit resolution, which consists of *n* stages, is under test. $V_{os,i}$ and g_i are the comparator offset and the gain of stage *i*, respectively, and *i*=1, 2, …, *n*. To study the effect of $V_{os,i}$ and g_i on the ADC linearity performance, we start from stage 1 and regard the following *n*-1 stages as an (*n*-1)-bit ADC with the transition level distribution $T_{n-1}(j)$, *j*=1, 2, …, 2^{n-1}-1. Based on the transfer curve of stage *i*, we can express the transition level $T_n(k)$ as a function of $V_{os,1}$, Δg_1 and $T_{n-1}(j)$, *j*=1, 2, …, 2^{n-1}-1 as in

$$T_n(k)=\begin{cases}\dfrac{T_{n-1}(k)+V_{ref}}{g_1}-V_{ref} & ,1\le k\le 2^{n-1}-1\\ V_{os,1} & ,k=2^{n-1}\\ V_{ref}-\dfrac{V_{ref}-T_{n-1}\left(k-2^{n-1}\right)}{g_1} & ,2^{n-1}+1\le k\le 2^{n}-1\end{cases}. \tag{4.6}$$

In (4.6), the expressions of $T_n(k)$s for $1\le k\le 2^{n-1}-1$ and $2^{n-1}+1\le k\le 2^{n}-1$ are calculated from the fixed points A and B, respectively. From (4.6), it is clear that the errors $V_{os,1}$ and g_1 can be expressed by transition levels $T_n(2^{n-1}-1)$, $T_n(2^{n-1})$ and $T_n(2^{n-1}+1)$ as in

$$V_{os,1}=T_n\left(2^{n-1}\right) \text{ and} \tag{4.7}$$

$$g_1=\frac{2V_{ref}+\left[T_{n-1}\left(2^{n-1}-1\right)-T_{n-1}(1)\right]}{2V_{ref}-\left[T_n\left(2^{n-1}+1\right)-T_n\left(2^{n-1}-1\right)\right]}. \tag{4.8}$$

In (4.8), $T_{n-1}(2^{n-1}-1)-T_{n-1}(1)$ is approximately equal to $2V_{ref}\times(2^{n-1}-2)/2^{n-1}$. Consequently, (4.8) can be rewritten as

$$g_1\approx\frac{\dfrac{2^{n-1}-1}{2^{n-3}}V_{ref}}{2V_{ref}-\left[T_n\left(2^{n-1}+1\right)-T_n\left(2^{n-1}-1\right)\right]}. \tag{4.9}$$

Therefore, with measured transition levels $T_n(2^{n-1}-1)$, $T_n(2^{n-1})$, and $T_n(2^{n-1}+1)$, the first stage comparator offset and gain can be characterized. In some cases, it is much easier and cheaper to measure the code-bin widths instead of the absolute transition levels. Assume the width of code k, $W_n(k)=T_n(k+1)-T_n(k)$, and the $V_{os,i}$ and g_i can be expressed by

$$V_{os,1}=\frac{W_n\left(2^{n-1}-1\right)-W_n\left(2^{n-1}\right)}{2} \text{ and} \tag{4.10}$$

$$g_1\approx\frac{\dfrac{2^{n-1}-1}{2^{n-3}}V_{ref}}{2V_{ref}-\left[W_n\left(2^{n-1}-1\right)+W_n\left(2^{n-1}\right)\right]}, \tag{4.11}$$

by noting that

$$T_n\left(2^{n-1}-1\right)+T_n\left(2^{n-1}+1\right)\approx 0\,. \tag{4.12}$$

After stage 1 is characterized, the values of $V_{os,1}$ and g_1 can be used with the measured the transition levels T_ns or the code-bin widths W_ns to solve T_{n-1}s or W_{n-1}s of the following n-1-bit ADC by (4.6). Similar procedures can then be applied to solve the $V_{os,2}$ and g_2 after getting $T_{n-1}(2^{n-2}-1)$, $T_{n-1}(2^{n-2})$, and $T_{n-1}(2^{n-2}+1)$ or the corresponding $W_{n-1}(2^{n-2}-1)$ and $W_{n-1}(2^{n-2})$. Thus, all the stage parameters comparator offsets and the gains can be obtained sequentially. To get the linearity performance of the ADC, we first calculate all the transition levels from the obtained stage parameters $V_{os,i}$ and g_i, i=1, 2, …, n. Then, the *INL* and *DNL* can be computed by their definitions.

Evident in the analysis, only three transition levels or two code-bin widths are required to characterize each stage. Thus, for testing an n-bit ADC, no more than $3n$ transition levels or $2n$ code-bin widths are required. For example, if a 14-bit ADC is under test, we only need to measure $T_n(2^m-1)$, $T_n(2^m)$, and $T_n(2^m+1)$, m=0, 1, …, 13, for a total of 38 transition levels, or $W_n(2^m-1)$ and $W_n(2^m)$, m=1, …, 13, for a total of 26 code-bin widths. As a result, the size of the data set for testing is reduced by a factor of several hundreds.

In the analysis, it is assumed that there is no missing code for ADCs under test. This is just for simplicity of the explanation. The missing codes can be easily detected by measuring the selected transition levels or code bin-widths, and the algorithm will still be useful with a minor change.

4.4 Simulation Results

In this section, numerical simulation results will be shown to verify the correctness of the derivation and the procedure of the system identification-based testing.

The comparator offset and the stage gain in simulation are modeled as random variables following normal distribution as $V_{os,i} \sim N(0, \sigma_{os}^2)$ and $g_i \sim N(g_0, \sigma_g^2)$, where σ_{os} is the standard deviation of the offset, and g_0 and σ_g are the nominal value and the standard deviation of the stage gain, respectively. The pipeline stage is modeled based on the transfer curve shown in Figure 4.3. To verify the derivation, first we test the simulated ADC using the conventional histogram method with 16 samples per code to get the measured transition levels and the *INL*(*k*)s as the reference. Then, the transition levels, which are necessary for system identification, are extracted and $V_{os,i}$ and g_i can be calculated. Finally, the *INL*(*k*)s from the histogram test and from the proposed method are compared. Figure 4.4 and figure 4.5 illustrate the measured *INL*(*k*)s and the result difference for a 12-bit ADC and a 14-bit ADC, respectively.

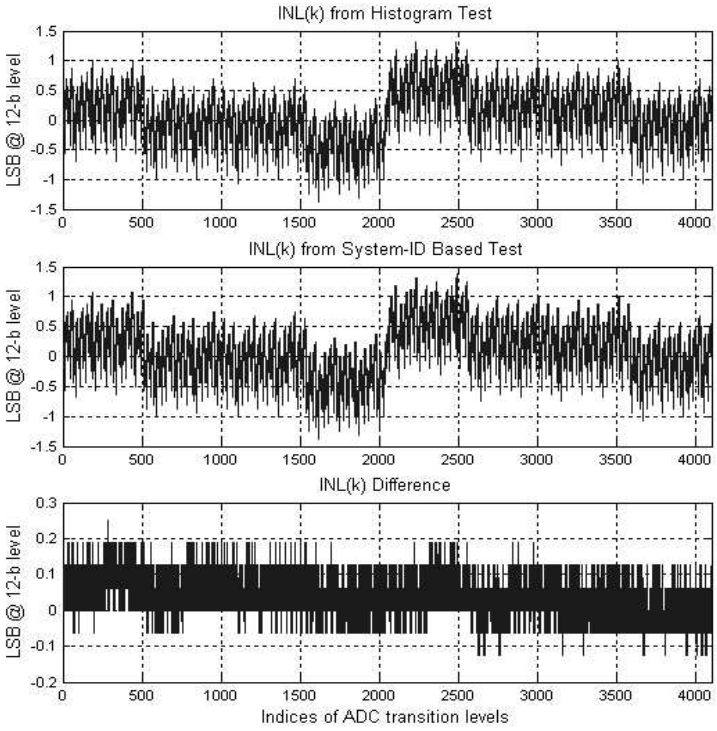


Figure 4.4. Simulated *INL*(*k*)s of a 12-bit ADC

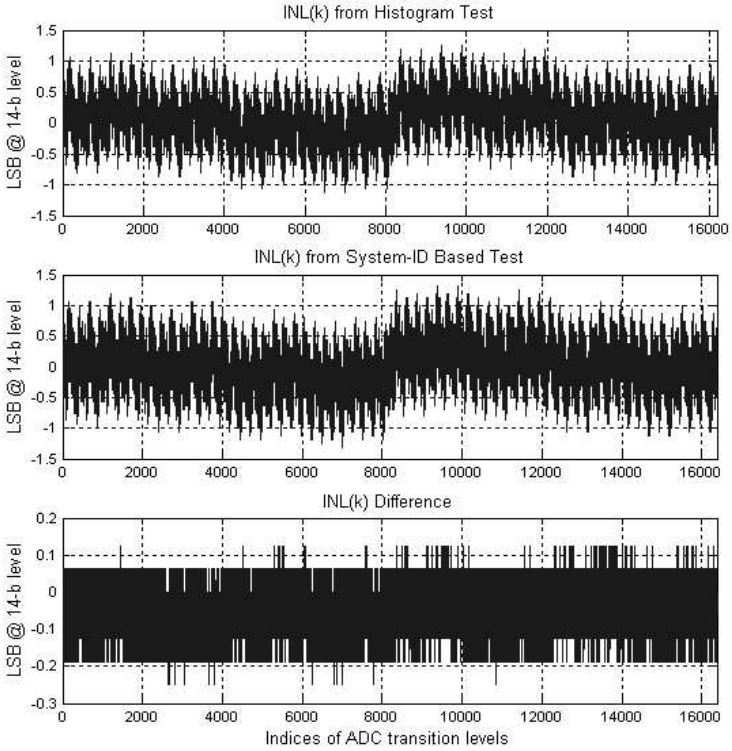


Figure 4.5. Simulated *INL*(*k*) of a 14-bit ADC

Simulation shows that the proposed method is able to offer satisfactory testing performance. However, it has a systematic error in the identification, which is generated by the approximation in the derivation. From the simulation, the error, which is evaluated by the *INL*(*k*) difference, is bounded by a quarter LSB for both 12-bit and 14-bit ADCs, which is usually small enough in most cases. Simulation shows that the identification using code-bin widths can provide similar testing performance.

4.5 Measurement Setup

This section talks about the measurement setup for testing the selected transition levels or code-bin widths. As explained in the previous sections, the full-code linearity performance of the pipeline ADCs can be identified by the measurements of a small number of transition levels or code-bin widths. The commonly used method of testing ADC transition

levels one by one is the servo loop feedback method [36]. As has been discussed in Section 2.1.2.2, in this approach an input is applied to the ADC under test, and the ADC outputs are compared to a preset value *k*, which specifies that the transition level at code *k* is under test. If the ADC output is below the preset value, the input is raised by a certain amount. If the ADC output is equal to or above the desired value, the input is reduced by a certain amount. This process is repeated until the ADC input has settled to a stable average value. After the loop has settled, the input is regarded to be equal to the code transition level, $T(k)$. Its value can then be either measured by a high-precision digitizer or computed from the known transfer function of the input source. However, this method requires a considerable amount of time for the test loop to be settled every time. Even for a very small number of transition levels, the test time may be still prohibitive, because, overall, using high-precision test equipment with long test times results in high test costs. Another method that can be used to measure the transition levels one by one is the BIST solution proposed in chapter 3. In this case, an on-chip source generator and evaluation circuits eliminate the need for external high-precision, mixed-signal ATE testers. Therefore, test costs can possibly be reduced. In addition, embedded testing helps with keeping the signal integrity, and so test performance may be improved as well. However, this approach involves on-chip BIST circuit design and extra die area, so it may not be easily applied in some cases. Furthermore, BIST is obviously not a choice for ADCs that are already implemented on silicon.

Sometimes code-bin widths can be more easily measured than the absolute ADC transition levels. For example, the histogram test using ramp-source signals can easily measure each code-bin width from the histogram information. The difference in this method from the full code histogram test is that in this case only a small number of code-bin widths

need to be measured. Therefore, input ramps do not need to cover the whole ADC input range, and, thus, test time can be reduced. As explained in the previous sections, to test an *n*-bit pipeline ADC, only $W_n(2^m-1)$ and $W_n(2^m)$, m=1, …, n-1, for a total of 2(*n*-1) code-bin widths are enough to characterize the gains and offsets of the pipeline stages. Then, the ramps for testing should fully cover those critical code bins.

The measurement system basically includes a ramp generator and a DAC. The ramp generator generates small-range ramp signals with enough linearity performance, which cover the desired code bins, and the DAC is used to generate proper offsets additions to the ramps to shift them to appropriate positions. Figure 4.6 shows the block diagram of the test system, while figure 4.7 shows the coverage of the source signals. Considering the noise effects and the nonlinearity on the ADC transition levels, the ramps' signals should cover extra code bins overall. For example, according to the analysis, there are two adjacent code bins that need to be measured at each critical position. In the test setup, the small ramps generated by the ramp generator cover eight ideal code bins. The extra cover range at each location not only accommodates the noise effect and the nonlinearity of ADC, but also relaxes the accuracy requirement of the DAC for offset generation. Furthermore, in the previous analysis, all the critical locations are on the lower half of the ADC input range. It is easy to see that the same information can also be generated from the code-bin widths on the upper half of the input range except for the first stage. In the measurement, we can measure those locations on both lower- and upper-input ranges so that the measurement accuracy can be improved. With all these considerations, we can calculate the total code bins that are actually covered by the input ramps, which are about 8×[2(*n*-1)-1], where *n* is the resolution of ADCs under test. In this case, 14-bit ADCs are under test, and it can be calculated that the

total number of code bins covered by the ramps is less that 2% of the full code bins, assuming the ramp slope and the ADC sampling rate are the same as the full code histogram test for the similar test accuracy. As a result, the testing time can be reduced by more than 98% by using this system identification-based reduced-code testing method.

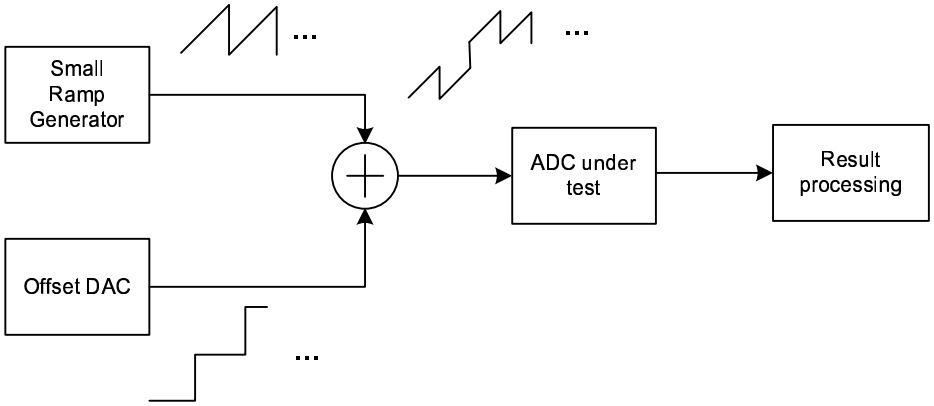


Figure 4.6 Block diagram of test system for measuring critical code-bin widths

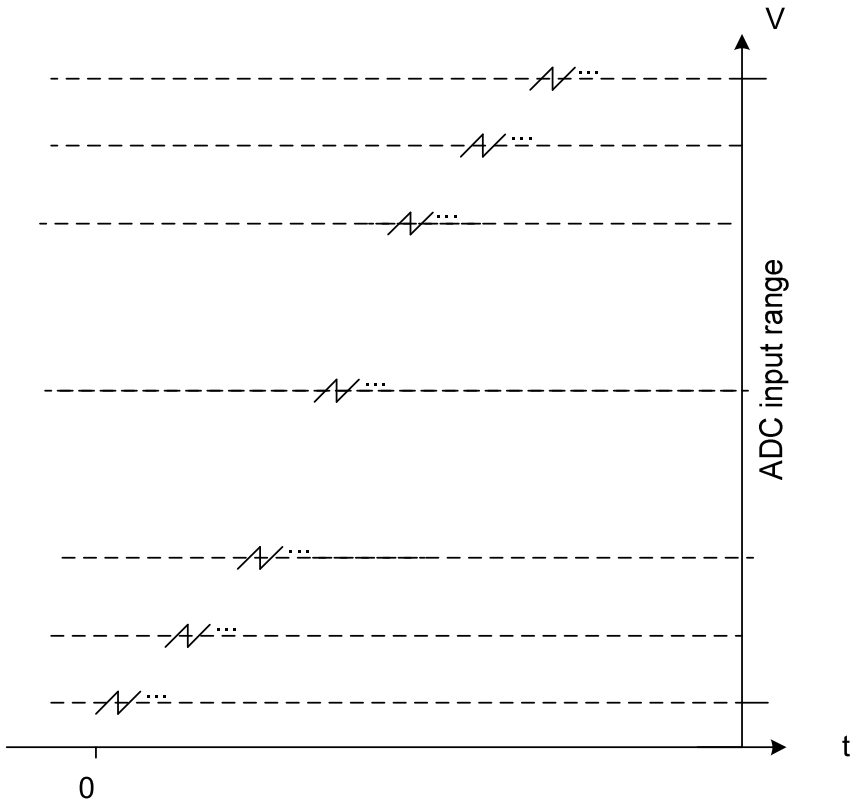


Figure 4.7 Coverage of input ramp signals for measuring critical code-bin widths

4.6 Experimental Results

Experimental testing has been conducted to further validate the rationality of the assumptions and the efficacy of the algorithm. First, the 14-bit commercial ADCs are carefully characterized using the histogram test with about 240 samples per code. The transition levels or the code-bin widths information are available from the histogram test. Then, the proposed method can be applied to estimate the linearity performance of the ADCs. Figure 4.8 shows the reduced-code testing of the *INL*(*k*)s using the transition levels, and figure 4.9 shows the reduced-code testing of another ADC's *INL*(*k*)s using the code-bin widths. The experimental results prove that the proposed method is able to test real 14-bit ADCs to a quarter LSB accuracy level by testing a very small set of the transition levels or the code-bin widths. 10 samples have been tested for the verification. They all show the same estimation error level.

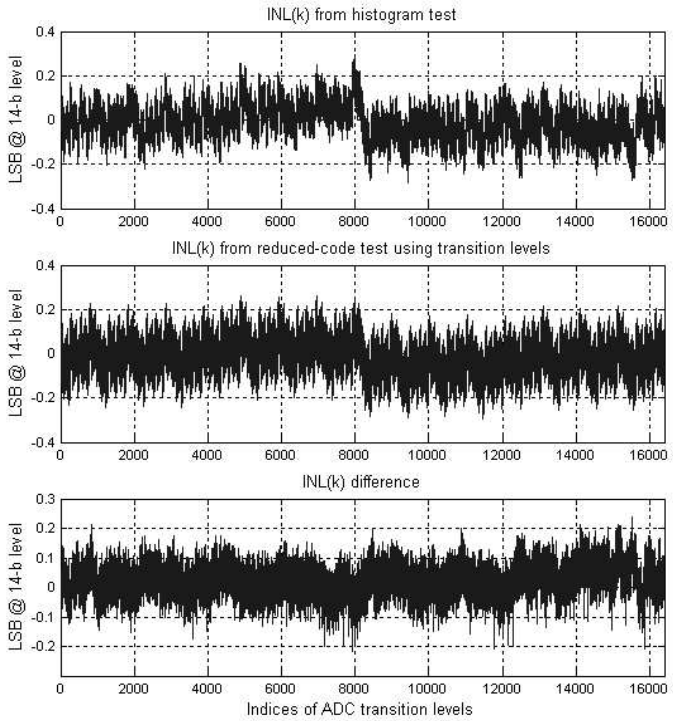


Figure 4.8. *INL*(*k*)s of a 14-bit ADC using the transition levels

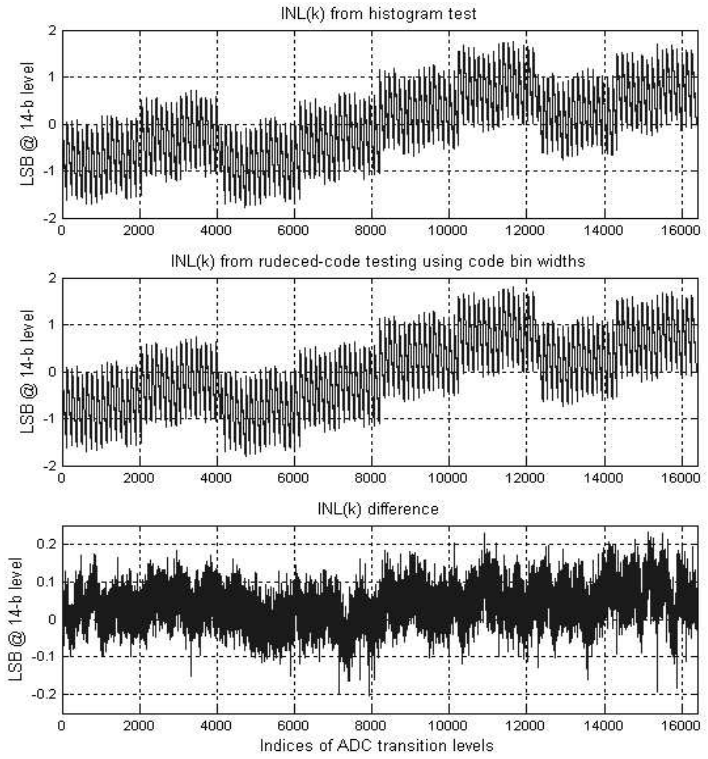


Figure 4.9. *INL*(*k*)s of a 14-bit ADC using the code-bin widths

4.7 Conclusions

This work investigates a reduced-code testing method for pipeline ADC linearity testing aimed at greatly reducing testing time and shows one example of using the BIST system proposed in Chapter 3. System parameters, such as the comparator offsets and the stage gains, are identified based on a very small set of measurements. The full-code linearity performance is then reconstructed using the identified model. Both simulation and experimental results show that the new method cuts down the data acquisition time by a factor of several hundreds and still is able to achieve satisfactory testing accuracy. Although this work examines a 1-bit/stage pipeline ADC, the method can be extended to multi-bit per stage pipeline ADCs as well. This method is only one simplistic way of using reduced-code testing to obtain full-code test performance. As we discussed, the nonlinear errors in the

system are not specifically considered in this method. However, in many cases that the designs are not quite linear, this method can not predict the nonlinear errors accurately. We have been trying to get the nonlinear information of the SHA and each stage. However, it is unlikely to do that from a small set of transition levels. As a result, this method is more suitable to the parts that are well designed to characterize their full-code nonlinear performance with less test time.

Chapter 5. Static linearity Built-in Self-Test of High-resolution High-Speed DACs Using Low-accuracy on-chip circuitry

On-chip testing of high-resolution high-speed DACs is extremely challenging because of the stringent requirements on the accuracy, speed, and cost of the embedded measurement circuits. In this chapter, we introduce a new on-chip strategy for DAC linearity testing applying the proposed deterministic dynamic element matching (DDEM) technique. Low-accuracy, two-step flash ADCs are used as test devices. The speed advantage of flash structure enables at-speed testing, while its accuracy and resolution are improved by the DDEM algorithm, the second stage, and the dithering technique. The architecture of the DDEM flash ADC and DDEM algorithm are described, and the design considerations of the major circuit blocks are examined. Finally, the test performance is analyzed theoretically and verified by simulation.

5.1 Introduction

As an interface between digital processing and the analog world, the digital-to-analog converter (DAC) is one of the most widely used mixed-signal integrated circuits. In the recent years, along with the development of new applications in wireless communications and multimedia signal processing, digital to analog conversion performance has become increasingly important [40]. High-speed, high-resolution DACs are widely manufactured and used, and the best co

mmercial parts, such as AD9771 from ADI and DAC5678 from TI, have 16-bit resolutions and more than 500MSPS update rates. Consequently, testing of those high-

performance DACs has become one of the most challenging problems in the area of AMS testing.

On-chip built-in self-test (BIST) enables improvements in test efficiency and test speed. Meanwhile, it avoids the high cost of using of external automatic test equipment (ATE). However, BIST needs additional circuitry, which sometimes is difficult to implement. This problem is more obvious when BIST is applied to the linearity test of high-speed and high-resolution DACs, since that testing usually requires measurement circuits with much better linearity performance and faster speed than others DACs. Developing measurement circuits with high performance but low cost has been regarded as the bottleneck of on-chip DAC testing.

This chapter discusses a solution to high-performance DAC on-chip testing by using low-accuracy but high-test performance circuits as on-chip evaluation circuits. Low-accuracy circuits are usually fast, easy to build and cost effective, but in order to use them as measurement devices, we need to improve their linearity/ resolution performance. The so-called deterministic dynamic element matching (DDEM) technique, which was introduced in chapter 3 for on-chip linearity testing of high-resolution ADCs, is applied to improve the accuracy of the low-cost, on-chip analog circuitry by rearranging the connections of analog unit cells with considerable mismatch errors.

In the proposed solution, the low-resolution and low-accuracy Flash ADCs incorporating the deterministic DEM method are used as on-chip measurement circuits, and the analog unit cells that are rearranged by DDEM are resistors in the resistor strings of the flash ADCs. The rest of chapter 5 is organized as follows. Section 5.2 discusses the basic idea of statistic testing of data converters and the distribution characteristics of the DDEM

algorithm, while section 5.3 describes the proposed BIST structure and DDEM test algorithm. Section 5.4 theoretically evaluates the test performance, section 5.5 talks about some design considerations of the DDEM ADC, and section 5.6 shows the simulation results. Section 5.7 concludes the chapter.

5.2 Statistical Test of Data Converters and DDEM

5.2.1 Statistical Test of Data Converters

Statistical approaches are widely used in analog-to-digital converter testing. A typical example is the histogram test method. As described in chapter 2, one of the most commonly reported stimuli used in the histogram test is the linear ramp, and an ADC under test takes lots of samples on a linear ramp when it is used as test signals. In this case, the sampled voltages need to be uniformly distributed over the input range of the ADC. The width of the ADC's k^{th} code bin can be represented by the number of the hits $H[k]$ at output code k, where k=1, 2… 2^n-2 and n is the resolution of the ADC. The relative position of ADC transition point T_k with respect to the first transition point T_1 can be characterized by the number of output codes that belong to [1, k-1] as in

$$T_k - T_1 \propto \sum_{j=1}^{k-1} H_o(j). \tag{5.1}$$

Then, the estimation of the nonlinearity errors of the ADC under test can be expressed by the histogram counts, but the accuracy of this estimation is limited by the linearity of the test ramp signal and some random issues such as noise, sampling uncertainty, and clock jitter. The ideal case is that the sampled voltages are uniformly distributed over the input range of ADC under test with enough resolution. The nonlinearity errors of realistic ramp signals will affect the histogram test accuracy.

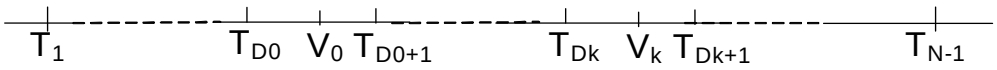


Figure 5.1. Relative position of V_k with respect to ADC transition points

A similar method can be applied to the DAC nonlinearity test. The differential nonlinearity (*DNL*) and the integral nonlinearity (*INL*) of a DAC describe its nonlinearity errors. They are defined as following:

$$DNL(k) = \frac{V_{k+1} - V_k}{LSB} - 1 \tag{5.2}$$

$$INL(k) = \frac{V_k - V_0}{LSB} - k \tag{5.3}$$

$$LSB = \frac{V_{2^n-1} - V_0}{2^n - 1}, \tag{5.4}$$

where V_k is the analog output of the DAC for input code *k*, and *LSB* is the ideal voltage increment defined by the end point fit line. As shown in (5.2) and (5.3), the *DNL* and *INL* of the DAC can be characterized by estimations of V_k-V_0. Usually, a high-resolution/accuracy ADC is used as a test device to make the estimations. Assume the ADC digital output codes for inputs V_0 and V_k are D_0 and D_k, respectively, as shown in Figure 5.1. The estimation of V_k-V_0 is D_k-D_0, which also indicates the number of ADC transition points between V_0 and V_k. This process is the same as when we use the number of samples between two adjacent transition points to estimate the width of a code bin in ADC testing. In order to reduce the estimation error, transition points of the ADC need to have high resolution and be uniformly distributed, but, due to component mismatches, this requirement is very challenging, especially when the DAC under test has a high resolution. An alternative approach is to use a bunch of ADCs with low resolution and accuracy, but the overall transition points conform to

the requirement: with high resolution and uniformly distributed. As will be explained in the following section, DDEM can provide ADCs with properly distributed transition points.

5.2.2 Deterministic Dynamic Element Matching in Flash ADCs

Due to process variation, mismatch errors are inevitable in integrated circuits. Although special layout techniques, special processes, and laser trimming can be used to reduce mismatch errors, these methods lead to significant cost increases and are difficult to use in BIST environment. The dynamic element matching (DEM) technique accepts matching errors as inevitable, but by dynamically arranging the interconnections of matching-sensitive elements, these elements can make great contributions to generating high-precision output signals in the existence of mismatch errors.

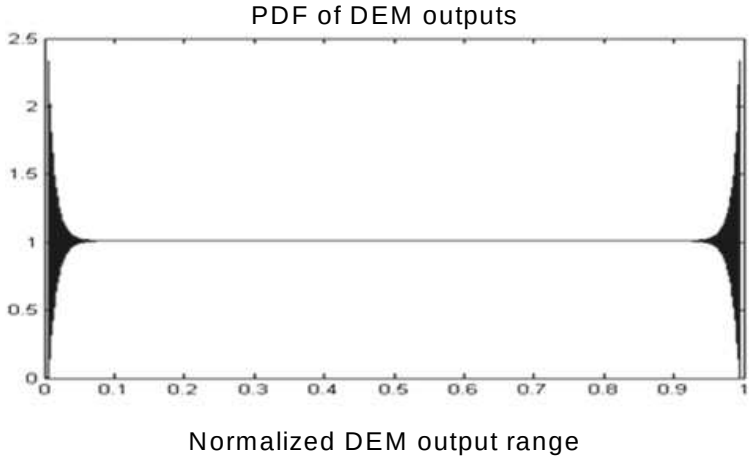


Figure 5.2. Probability density function of DEM output

We have been discussing applications of the DEM method in current-steering DAC design for high-resolution ADC testing. In that case, multiple-output voltages were obtained for one digital input by rearranging the connections of unit current sources. Different from other studies on DEM, which usually use a time averaged value of those outputs for a given

code and randomize the effect of nonlinear errors in frequency domain, DEM in this BIST application focuses on the distribution of all possible output voltages. Assume mismatch errors in resistors are from a normal distribution with a zero mean and a standard deviation $\sigma*R_0$, where R_0 is the desired resistance value. Theoretically, it can be proved that these voltages are almost uniformly distributed over the output range of the DAC except at the two ends, as illustrated in Figure 5.2 [41]. This statement is valid when there are enough analog outputs available for each input code and the unit cells with mismatch can be rearranged arbitrarily. But, those two conditions usually require complex control logic and long operation time. The so-called deterministic DEM, which deterministically rearranges the components to get reasonable number of outputs and a satisfactory output distribution, provides a solution to reduce logic complexity and limit the operation time. It has been proven to be a good way of generating stimuli for ADC linearity test, and in addition, the implementation is very simple.

In this work, we will apply DDEM to a flash ADC for testing DAC nonlinearity. The basic idea of DDEM is that instead of building one high-performance ADC satisfying the resolution, linearity, and speed requirements, we use DDEM to generate a set of ADCs with low resolution and low linearity, but with fast speed, conditions which are much easier to implement. Although the transition levels of each ADC are inaccurate, those of all the ADCs are distributed nearly evenly in their common input range, and, as a result, the overall resolution and linearity performances are greatly improved. In this case, the matching-sensitive components are resistors in the resistor-string (R-string) of the flash ADC. Rearrangements are applied to resistors to form different R-strings and get different sets of ADC transition levels. The overall distribution of all the possible transition points is nearly

uniform, similar to what was shown in Figure 5.2, which is the desired distribution of ADC transition levels to be used in DAC testing. In testing, the DAC's output range should be fully covered by the middle of ADC input range, where the distribution of the transition levels is flat.

Flash structure is adopted for this application because of its speed advantage. Assume we have an *n*-bit flash ADC with DDEM reconfiguration capability, and figure 5.3 shows the structure of an *n*-bit DDEM flash ADC. Similar to a typical flash ADC, an R-string with *N* resistors of R forms a voltage divider that provides reference voltages, where $N=2^n$. Each comparator gives out a "1" when the analog input voltage is higher than the reference voltage applied to it. Otherwise, the comparator output is "0". The decoder converts thermometer codes to binary codes. Different from the conventional flash structure, resistors are physically connected as a loop via switches in the DDEM ADC. The loop can be broken at different positions by opening specific switches to build different R-strings and, consequently, different ADCs. *N* comparators are needed to make comparators rearrange with resistors easily. The proposed DDEM strategy dynamically generates R-strings in such a way that all the resistors are almost equally used. Each time, one of *P* switches, S_i for $i=(j-1)*q+1$, $j=1, 2\ldots P$, is open, where these *P* switches are uniformly distributed on the resistor loop. The parameter *P* is the number of different R-strings and selected so that $q=N/P$ is an integer. By connecting the two nodes of the open switch to external reference voltages, a set of internal reference voltages is generated. Therefore, *P* digital outputs are available for one analog input quantized by the DDEM ADC with different sets of reference voltages.

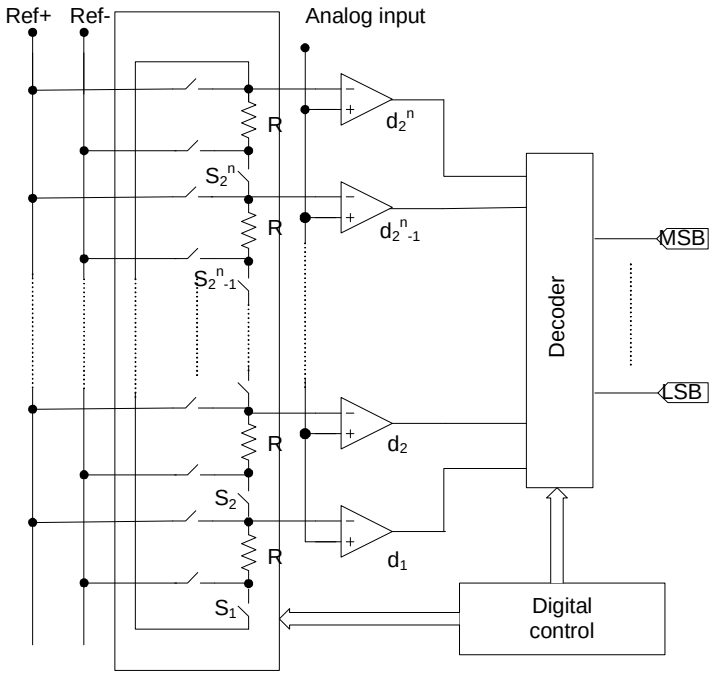


Figure 5.3. Structure of an *n*-bit DDEM flash ADC

Figure 5.4 shows an example of the reconfiguration of a 4-bit resistor string controlled by DDEM. There are 16 resistors connected in a loop through switches. To build a resistor string, we can break one switch, for example the switch S_1, and connect the two ends to the positive and negative references, respectively, to generate a set of transition levels. Assume there are P different configurations totally, and we can just pick different P switches in the loop as the broken points. In DDEM algorithm, as we have mentioned, these selected P switches are equally spaced on the loop. For example, if P is equal to 4, the first configuration can be generated by open S_1 as shown in Figure 5.4(a), and for the second configuration, the switch S_5 will be open as in Figure 5.4(b). The switch, S_9,then will be open for the third configuration, and the switch S_{13} will be open for the last configuration, as depicted in Figure 5.4(c) and Figure 5.4(d), respectively. Therefore, for each digital output

code, this DDEM flash ADC generated four corresponding transition levels. For example, the four transition points for output code D=5 will be:

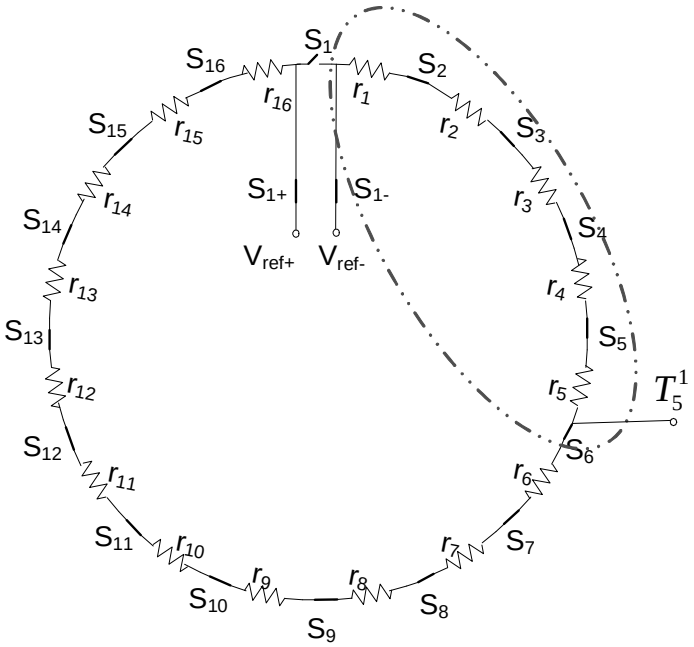
$$T_5^1 = \frac{(V_{ref+} - V_{ref-})}{R_{total}}(r_1 + r_2 + r_3 + r_4 + r_5), \tag{5.5}$$

$$T_5^2 = \frac{(V_{ref+} - V_{ref-})}{R_{total}}(r_5 + r_6 + r_7 + r_8 + r_9), \tag{5.6}$$

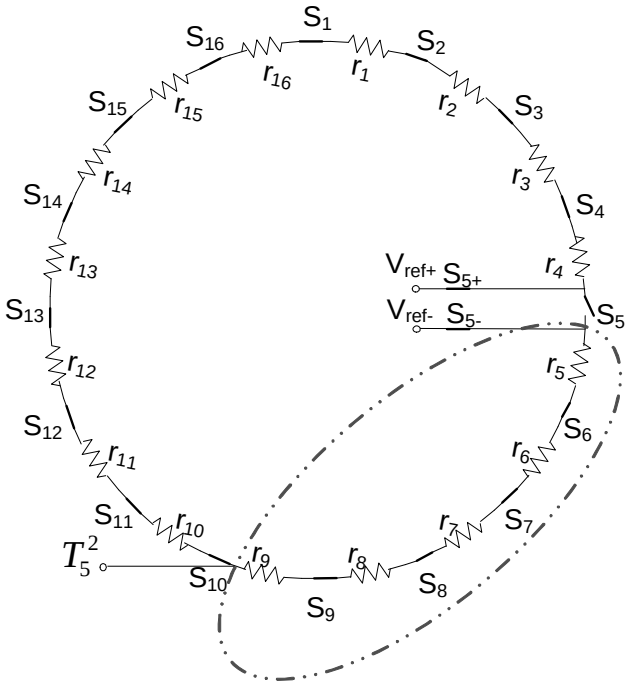
$$T_5^3 = \frac{(V_{ref+} - V_{ref-})}{R_{total}}(r_9 + r_{10} + r_{11} + r_{12} + r_{13}), \tag{5.7}$$

$$T_5^4 = \frac{(V_{ref+} - V_{ref-})}{R_{total}}(r_{13} + r_{14} + r_{15} + r_{16} + r_1), \tag{5.8}$$

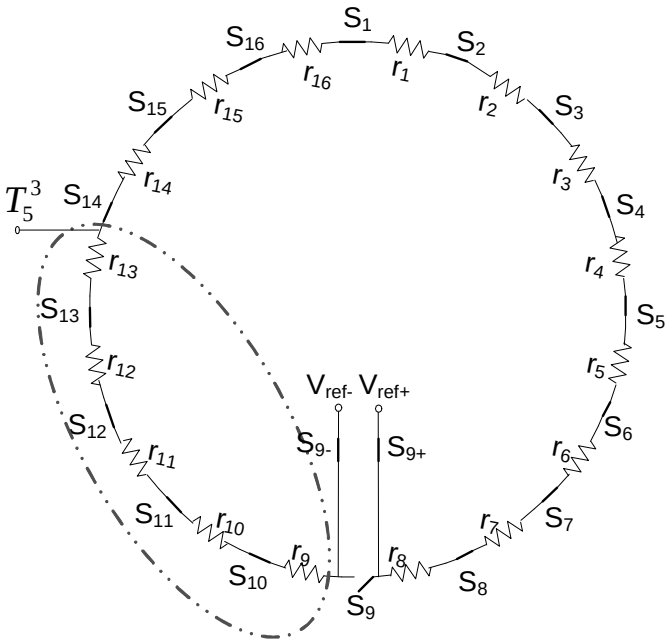
where V_{ref+} and V_{ref-} are reference voltages and R_{total} is the total resistance of the R-string, which is assumed to be a constant for all the four R-strings.



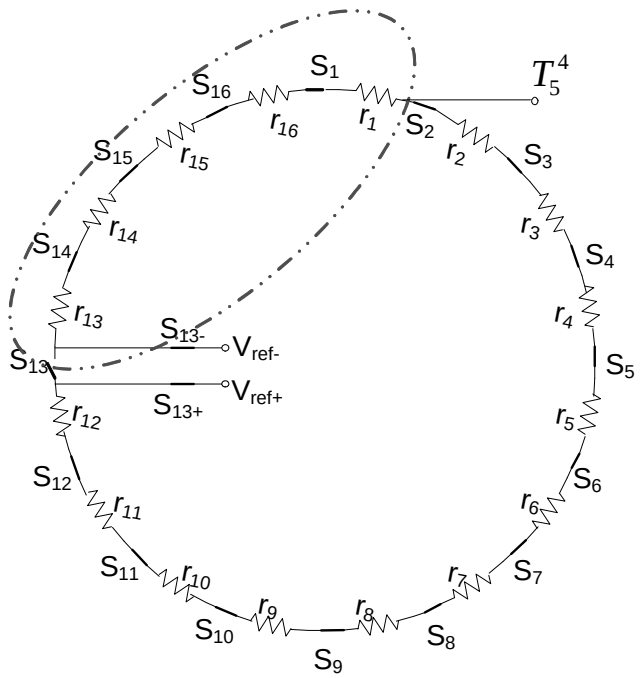
(a) 1st R-string, switch S_1 is open



(b) 2nd R-string, switch S_5 is open



(c) 3rd R-string, switch S_9 is open



(d) 4th R-string, switch S_{13} is open

Figure 5.4 Switching of a 4-bit DDEM flash ADC with *P*=4

As we have pointed out, the distribution of all the transition points are nearly uniform in the ADC input range and can be used to estimate DACs' output voltages. For each analog input V_i, which is from the DAC under test, P digital output codes, $d_{i,1}$, $d_{i,2}$… $d_{i,P}$, are obtained from the ADC. Assume the j^{th} reconfiguration of the resistor string provides a transition level set $\{T_{j,1}, T_{j,2}\ldots T_{j,N-1}\}$. Output code $d_{i,j}$ means the first $d_{i,j}$ transition levels in the j^{th} transition level set are less than the input voltage V_i. To get the measurement of V_i, which is denoted as m_i, we add up the P digital outputs $d_{i,1}$, $d_{i,2}$… $d_{i,P}$ under different configurations together as in

$$\begin{aligned} \sum_{j=1}^{P} d_{i,j} &= \sum_{j=1}^{P} \#\left\{T_{j,k} : T_{j,k} < V_i, k = 1,2,\cdots,N-1\right\} \\ &= \#\left\{T_{j,k} : T_{j,k} < V_i, k = 1,2,\cdots,N-1,\ j = 1,2,\cdots,P\right\} \end{aligned} \tag{5.9}$$

If we plot all the transition points of the P ADCs obtained from DDEM on one axis as in Figure 5.1, they are interleaved with each other and m_i is the number of points on the left to V_i as shown in (5.9). The more uniform the distribution of transition points, the more accuracy estimation we can get. This process is very similar to the code density of ADC linearity testing by using a DDEM DAC as source generator. Because transition levels are not uniform at the ends of the DDEM ADC's input range, the input signals to the ADC, which are the output voltages of the DAC under test, should be limited to the middle part of the whole input range. In the analysis, we will use the average of the P output codes as the measurement of the input. Because we will implement the output codes processing in the digital domain, as long as we have good computation accuracy, it will give the same estimation result.

4.3 DAC Linearity BIST Structure

The flash ADC provides the fastest conversion from an analog signal to a digital code and is ideal for applications requiring a large bandwidth. However, the resolution of single-stage flash ADCs is limited by their large number of comparators with usually no more than 8 bits. To reduce the quantization noise, we add a second stage and a dithering input DAC.

The second stage also uses flash structure so that the speed performance is not dramatically degraded. Figure 5.5 shows the block diagram of the two-step DDEM flash ADC. The two-step structure comprises an n_1-bit, coarse-stage DDEM flash ADC, an n_2-bit, fine-stage flash ADC, a residual voltage generator, a gain stage, a digital adder, and output latches. The coarse-stage ADC does the conversion for the first n_1 bits. A residual voltage is generated by subtracting from the analog input the reference voltage right smaller than it, determined by the coarse ADC output, and the residual voltages are amplified by the gain

stage to provide the input signals for the fine stage. Because of the significant mismatch errors in the coarse stage, the residual voltages are possibly larger than 1 LSB of the coarse stage. In order to avoid missing codes, the full-scale range of the fine stage is set to be equivalent to two coarse-stage LSBs, and a constant offset voltage is added to the residual voltages to move them up to the middle of the fine ADC's input range. This shift operation can compensate for the errors in residual voltages introduced by comparator offsets and, therefore, reducing the offset requirement for the comparators and simplifying the design. The coarse outputs and the fine-stage outputs are combined to get the final codes for the analog inputs to the DAC. In this DDEM structure, mismatches in the coarse resistor strings are desired to spread out distribution of transition levels after DDEM. This low matching requirement dramatically reduces the area consumption of the R-string. Because the full-scale range of the fine stage is only equivalent to 2 LSBs of the coarse stage, the fine stage can effectively increase the resolution of the test structure, and accuracy and linearity of the fine stage are not critical to the test performance.

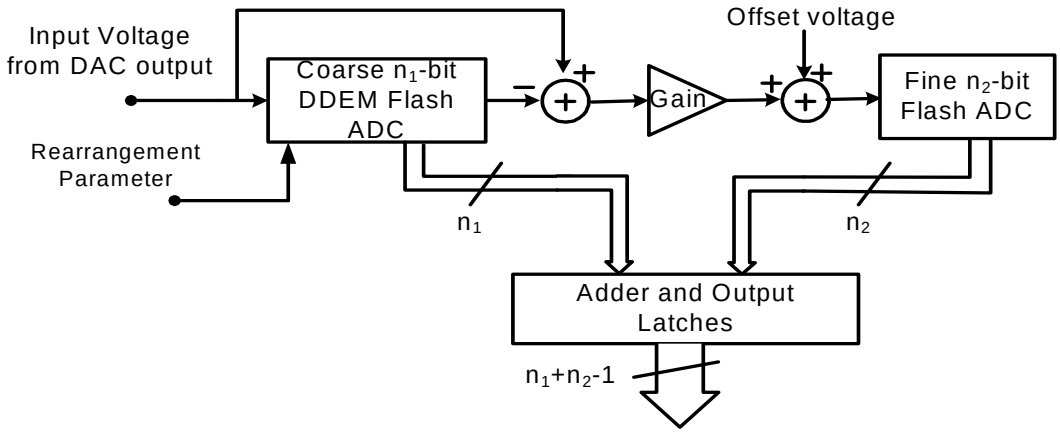


Figure 5.5. Block diagram of the two-step DDEM flash ADC

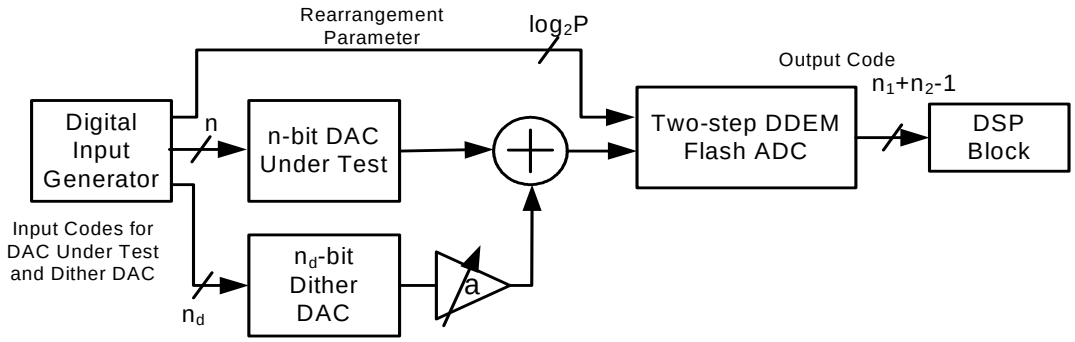


Figure 5.6. Block diagram of the proposed BIST scheme

As we have discussed, the resolution of the flash structure is limited by the number of comparators and is usually no more than 8 bits. To reduce the cost of the BIST circuit, the resolutions of the coarse stage and the fine stage should be as low as possible. But, as we will show in the analysis and simulation, the test performance of the two-step DDEM ADC cannot satisfy the requirement for high-resolution DAC testing. One solution to improving the test performance with little cost overhead is to incorporate another low-resolution dithering DAC to provide small and linear dither steps at the ADC input. Figure 5.6 illustrates the structure of the proposed BIST system. The output of the dithering DAC is added to the output of the DAC under test, and the sum is taken as the input to the DDEM ADC. The full-scale output range of the dithering DAC is adjustable and very small relative to the ADC input range (e.g. several LSBs of the original first stage flash ADC), which ensures the shifted DAC output signal is still covered by the middle linear part of DDEM ADC transition points. For each output of the dithering DAC, output voltages of the DAC under test are shifted up by a small offset. It is equivalent to where all the transition points of the ADC are shifted to the opposite direction by an equal amount. Assume the resolution of the dithering DAC is n_d, and the DDEM ADC's transition points are shifted 2^{n_d} times. The

overall distribution of transition levels with both DDEM and dithering is then the sum of 2^{n_d} shifted distributions of the DDEM transition points, and becomes more uniformly distributed and with better resolution. The nonlinearity error in the dithering DAC introduced by component mismatches can be neglected because of its small output range.

It is easy to see that the fine-stage quantization reduces the quantization errors in the estimation. For the dithering DAC, we will show in the next subsection that it can reduce the estimation errors introduced by the DDEM algorithm.

4.4 Test Performance Analysis

In this section, we theoretically evaluate the linearity performance of the proposed scheme. We will look at the DDEM ADC itself first and then add on the effect of dithering.

For a specific input voltage without dithering, V_{in}, the test system generates P digital outputs, d_1, d_2, …, d_j. The average of those P digital outputs is used as the measurement of V_{in}, m_{in}, as expressed in

$$m_{in} = \frac{1}{P}\sum_{j=1}^{P} d_j \ . \tag{5.10}$$

Considering the two-step structure, we can further express m_{in} as

$$m_{in} = \frac{1}{P}\sum_{j=1}^{P}\left(d_j^c + d_j^f\right) = \frac{1}{P}\sum_{j=1}^{P}\left(T_{id}\left(d_j^c\right) + r_j + \varepsilon_j\right), \tag{5.11}$$

where d^c_j and d^f_j are the first-stage (also known as the coarse-stage) and second-stage (also known as the fine-stage) outputs with proper scaling, respectively. Digital output codes can be equivalently represented by the ideal analog levels. For example, the coarse output d^c_j can be represented by the ideal coarse-stage transition level at code d^c_j, which is $T_{id}(d^c_j)$. As we have discussed, the residue amplifier generates residual voltages for the fine stage, which is

the difference between the input V_{in} and the actual transition level at code d^c_j from the j^{th} reconfiguration R-string expressed by $T_j(d^c_j)$, as in

$$V_{in} = T_j(d^c_j) + r_j,\ j = 1, \cdots, P. \tag{5.12}$$

Then, the fine stage output code can be represented by a residue voltage, r_j, in addition with a quantization error ε_j at n_{ADC}-bit level, where n_{ADC} is the resolution of the two-step ADC. The input voltage can be expressed by

$$m_{in} = \frac{1}{P}\sum_{j=1}^{P}\left(T_{id}\left(d^c_j\right) + r_j + \varepsilon_j\right). \tag{5.13}$$

On the other hand, the input V_{in} can be also accurately expressed by

$$V_{in} = T_{id}(k) + r_{id}, \tag{5.14}$$

where $T_{id}(k)$ is the nearest ideal coarse-stage transition level smaller than V_{in}, and r_{dd} is the residue voltage calculated by subtracting $T_{id}(k)$ from V_{in}.

We can further write equation (5.13) as

$$\begin{aligned} m_{in} &= \frac{1}{P}\sum_{j=1}^{P}(T_{id}\left(d^c_j\right) + r_j + \varepsilon_j) \\ &= T_{id}\left(k\right) + \frac{1}{P}\sum_{j=1}^{P}(T_{id}\left(d^c_j\right) - T_{id}\left(k\right) + r_j + \varepsilon_j) \end{aligned}. \tag{5.15}$$

With the expressions of V_{in} in (5.12) and (5.13), we can easily obtain that

$$r_{id} - T_j\left(d^c_j\right) = r_j - T_{id}\left(k\right),\ j = 1,\ 2,\ \cdots,\ P. \tag{5.16}$$

Then (5.15) can be rewritten as

$$
\begin{aligned}
m_{in} &= T_{id}(k) + \frac{1}{P}\sum_{j=1}^{P}(r_{id} + T_{id}(d_j^c) - T_j(d_j^c) + \varepsilon_j) \\
&= T_{id}(k) + r_{id} + \frac{1}{P}\sum_{j=1}^{P}\left[T_{id}(d_j^c) - T_j(d_j^c) + \varepsilon_j\right] \quad . \\
&= V_{in} - \frac{1}{P}\sum_{j=1}^{P} INL_j(d_j^c) + \frac{1}{P}\sum_{j=1}^{P}\varepsilon_j
\end{aligned}
\tag{5.17}
$$

Therefore, the estimation error is

$$
e_{in} = m_{in} - V_{in} = -\frac{1}{P}\sum_{j=1}^{P} INL_j(d_j^c) + \frac{1}{P}\sum_{j=1}^{P}\varepsilon_j \,, \tag{5.18}
$$

where $INL_j(d^c_j)=T_j(d^c_j)-T_{id}(d^c_j)$ is the coarse-stage integral nonlinearity error at code d^c_j of the j^{th} reconfiguration resistor string. The definitions of the differential and integral nonlinearity errors provide the following relationships:

$$
INL(k) = \sum_{i=1}^{k} DNL(i) \text{ and} \tag{5.19}
$$

$$
\sum_{i=1}^{N} DNL(i) = 0 \,, \tag{5.20}
$$

where N is the number of resistors in the string. In the DDEM algorithm, the differential nonlinearity errors are shifted by switching. Then, equation (5.20) can induce the following relationship:

$$
\sum_{j=1}^{P}\sum_{t=1}^{s*q} DNL_j(t) = s \times \sum_{i=1}^{N} DNL_1(i) = 0 \,, \tag{5.21}
$$

where $q=N/P$ and s could be any integer, and here we assume it to be a number in 1, 2, …, P and satisfy $d^c_j - sq \cong 0$ for all j=1, 2, …, P. With the three equations of *DNL* and *INL* above, the estimation error can be further expressed as

$$e_{in} = m_{in} - V_{in} = -\frac{1}{P}\sum_{j=1}^{P}\sum_{t=1}^{d_j^C - s*q} DNL_j(t) + \frac{1}{P}\sum_{j=1}^{P}\varepsilon_j \ . \tag{5.22}$$

It is noted that the first term in (5.22) is reduced to the summation of a set of *DNL(k)*s. With reasonable nonlinear errors in the coarse stage resistor string (less than 1 *LSB* of the coarse stage noted as LSB_C), those *DNL(k)*s are non-repeating. Thus, the value of first term in (5.22) should be comparable to the *INL* of the original coarse stage, INL_C. It is noted that this term is approximately periodic over the ADC input range with a period of q coarse-stage LSB_Cs. For the second term in (5.22), the quantization errors are at the n_{ADC}-bit level. With enough resolution in the fine stage, we can make the second term negligible in comparison to the first term. In this case, the estimation error becomes

$$e_{in} \approx -\frac{1}{P} INL_C \ . \tag{5.23}$$

As expressed in (5.23), the estimation error is at the effective number of bits of the coarse stage resistor string in addition to $\log_2 P$ and then the linearity of the coarse stage is improved by $\log_2 P$ bits.

Now, the effect of the input dithering DAC is considered. Its full-scale input range is taken as one period of the estimation error, which is q LSB_Cs of the coarse stage. The similar analysis of DDEM shows that this setup further reduces the nonlinear error after DDEM. The effect of the dithering is similar to that of the DDEM, and the estimation error with the dithering DAC can be finally expressed as

$$e_{tot} \approx \frac{1}{2^{n_d}} \frac{1}{P} INL_C \, , \tag{5.24}$$

where n_d is the resolution in bit of the dithering DAC and the quantization error part is neglected. As a result, the test performance of the whole system can be calculated as

$$n_{test} \approx ENOB_C + \log_2 P + n_d, \tag{5.25}$$

where $ENOB_C = n_1 - \log_2 INL_C - 1$ is the effective number of bits of the coarse stage.

4.5 Considerations for Circuit Implementation

In this section, we focus on the implementation of the two-step DDEM flash ADC, which is the main part of the system.

The detailed structure of the two-step DDEM ADC is shown in Figure 5.7. The DDEM ADC is composed of a 6-bit DDEM first stage, a residue amplifier, and a 6-bit second-stage flash ADC. A sample and hold stage is not necessary since DACs under test themselves will compensate for the time delay in the coarse and fine quantization. The first stage does the conversion for the first 6 MSB bits. After that, a residual voltage is created by subtracting the differential reference, which is the transition level of the coarse stage at code of the coarse-stage digital output, from the analog input. In order to avoid missing codes, the full-scale range of the second stage ADC is set to be equivalent to 2 *LSB*s of the first stage. To achieve that, the residue amplifier amplifies the generated residue voltage by a gain of 32.

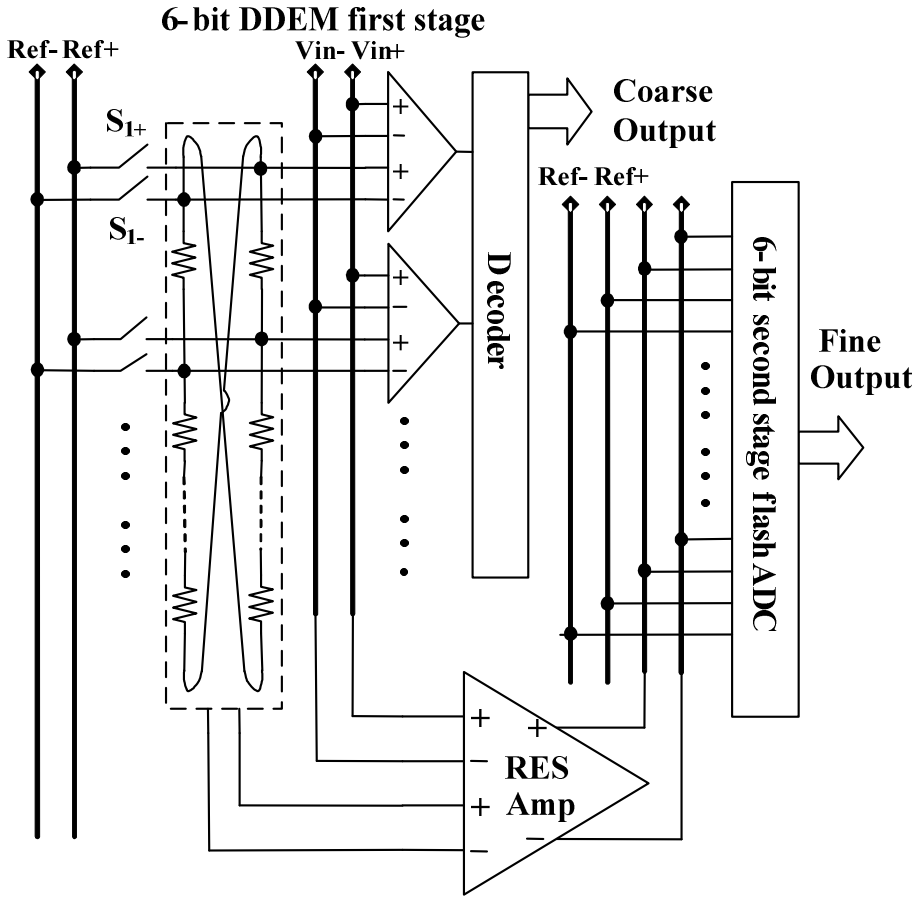


Figure 5.7. Structure of the proposed two-step DDEM ADC

Since the basic concept behind our on-chip testing solution is using low-accuracy circuits, the proposed scheme should be able to accommodate a considerable level of errors. In the rest of this section, three major circuit non-idealities, which include the on-resistance of switches, the offsets, and the gain error of the residue amplifier, will be discussed. Some will be approved by theoretical analysis to be non-critical. Others will degrade the test performance and need to be addressed.

5.5.1 On-Resistance of the Switches in Resistor Loop

Switches in the resistor loop have their own resistance when conducting. As shown in Figure 5.4, those resistances are added to the resistors in the loop and affect the transition levels. Bu, unlike the resistors, which have only voltage-independent random mismatches, the resistances of MOS switches are voltage-dependent and vary with their positions in the R-string because of their different gate-source voltages. In the DDEM algorithm, resistors in the R-string will be cyclically shifted to different voltage levels. For P different configurations, there are P different transition levels, $T_1(k)$, $T_2(k)$ … $T_P(k)$, generated for a specific digital output code k. In the ideal case, those transition levels are generated by resistances with only random mismatches so that the average of $T_1(k)$, $T_2(k)$ … $T_P(k)$ is close to the ideal transition level. However, if a voltage-dependent part is included in the resistance, all the transition level will exhibit the same level of error and so does their average. It can be shown that this error cannot be canceled out by the DDEM algorithm.

To solve this problem, the switches need to be moved out of the resistor loop, while the shifting ability of DDEM remains. That can be achieved in fully differential structure. In Figure 5.7, all the resistors are in a fixed loop, and switches are used to connect different nodes to the references. With one pair of switches on (like S_{1+} and S_{1-}), there are two resistor strings formed and connected to V_{ref+} and V_{ref-}, and differential transition levels can be generated for comparison. In this case, there are only two switches in the resistor strings and they are always connected to V_{ref+} and V_{ref-}. Therefore, voltage-dependent resistance is removed.

5.5.2 Offset Voltages

Two kinds of offset voltages are discussed in this subsection, which are comparator offset voltages and the residue amplifier offset voltage.

Comparator offset voltages

Comparator offset voltages degrade the accuracy of flash ADCs since they directly sum to the reference voltages generated from the resistor strings. The offset voltages are random and characterized by the standard deviation, so to design for high yield, the standard deviation is usually less than 0.2 *LSB* of the ADC. Large transistors or some offset cancellation techniques sometimes are necessary to reduce the offsets.

Two-step ADCs usually have extra input range in their second stage to compensate for the comparator offsets in the first stage. In the DDEM ADC described here, the full-scale input range of the second stage is equivalent to 2*LSB* of the first stage. Thus, $\pm 0.5LSB$ of the first stage is available for error compensation. In addition, the averaging effect of DDEM and dithering further relaxes the requirement on the first-stage comparator offsets. Simulation results show that a 6-bit first stage can have comparator offsets voltages with a standard deviation of 0.3*LSB* of the first stage without degrading the test performance significantly. Thus, the first-stage comparators can be very low-accuracy and area-efficient.

The second stage will not affect the linearity; instead, it affects the quantization error of the system. Therefore, the second-stage comparator offsets are not critical as long as the quantization error is small enough. Small input transistors can be used in the second-stage comparator to reduce the load capacitance of the residue amplifier and, therefore, improve the speed.

The offset voltage of the residue amplifier

Offset voltage errors from the residue amplifier changes the residue signal by a fixed value. For any analog input of the ADC, this error induces a constant offset in the final measurement. The good thing is the constant offset will not cause any error in *DNL* and *INL* estimation since they are calculated from relative DAC output levels, not absolute ones.

5.5.3 Gain Error of the Residue Amplifier

The residue amplifier amplifies the difference between the input signal and the reference by a nominal gain of 32. Gain error of the amplifier will induce missing codes or cause the two-step ADC to be non-monotonic. In this subsection, we will analyze the effect of the gain error in the DDEM algorithm. Gain error of the residue amplifier can be divided into two parts: the gain difference error and the gain variation error. The gain difference is the static error between the nominal gain and the actual gain after fabrication. It does not change during the ADC operation. The gain variation error represents the dynamic error of the gain, which changes with the input common mode of the residue amplifier.

Gain difference error

In this design, open-loop low-gain amplifiers are used for residue amplification because of their high bandwidth. The whole amplifier consists of three gain stages, and each stage has a nominal gain of about 3.2 to achieve the desired gain of 32. However, the gain of the open-loop amplifiers can not be accurately controlled as a result of process variation. Assume we have an analog input V_{in}. Then, the measurement of V_{in} obtained from the dithering-incorporated DDEM ADC is expressed as

$$m_{in} = \frac{1}{N_D P} \sum_{i=1}^{N_D} \sum_{j=1}^{P} C_{i,j} \,, \tag{5.26}$$

where N_D is the number of available dither levels, P is the iteration parameter, and $C_{i,j}$ is the digital output of the DDEM ADC associated with the i^{th} dither level and j^{th} R-string configuration. Assume m_{in0} is the measurement when the gain of the residue amplifier is exactly 32, and it can be shown that with an actual gain of $32+\Delta G$ the measurement can be expressed as

$$m_{in,\Delta G} = m_{in,0} + \frac{\Delta G}{32} \cdot \frac{1}{N_D P} \sum_{i=1}^{N_D} \sum_{j=1}^{P} r_{i,j} \,, \tag{5.27}$$

where $r_{i,j}$ is the residue voltage for i^{th} dither level and j^{th} R-string configuration. The second item in the right portion of (5.27) is the additional error induced by the gain difference. It is noted that the dithering DAC almost uniformly spreads the analog input in q LSB_Cs of the first stage. At the same time, the DDEM algorithm also randomizes the residue voltages associated with a specific analog input uniformly into a 1-LSB_C range. Although each of the residue voltages associated with a specific input cannot be accurately expressed, the average of them is nearly constant, especially for a large P. Thus, the error term in (5.27) is almost constant for all the input signals and will not affect the estimation of *INL*. This fact greatly reduces the sensitivity of the testing accuracy with respect to the gain difference error.

Gain variation error

The inputs of the residue amplifier are two differential signals. Their common mode signal depends on the input signal level and is different for different first-stage digital outputs. The disadvantage of this fact is that changing the common mode will cause the gain of the amplifier to vary and make it input signal dependent. As described in the subsection 5.4.1,

this signal dependence generates test error that cannot be cancelled by DDEM and dithering, since this error is almost constant for a specific analog input despite of DDEM and dithering but different for different input levels. It is noted from the analysis that the induced estimation error directly copies the gain variation. Therefore, the gain variation can be calibrated out if the gain variation is tested. This information can be obtained from the digital outputs of the second stage. The second stage quantizes the amplified residue voltages, and for the same first-stage output code, *k*, the largest of all the second-stage output codes shows the characteristics of the gain error as well as *DNL(k)* of the first stage. If the quantization error is ignored, the maximum second-stage output at code *k*, $C^{f}_{Max,k}$, can be expressed as

$$C^{f}_{Max,k} = G \cdot (1 + g_k) \cdot (1LSB_C + DNL(k)), \tag{5.28}$$

where *G* is the static gain of the amplifier, and g_k is the normalized dynamic gain variation at code *k*. Considering the DDEM algorithm, we have the maximum outputs under different configurations as $C^{f}_{Max,k,1}$, $C^{f}_{Max,k,2}$, …, $C^{f}_{Max,k,P}$. Then, the average can be calculated as

$$\bar{C}^{f}_{Max,k} = G \cdot (1 + g_k) \cdot \left(1LSB_C + \frac{1}{P} \sum_{j=1}^{P} DNLj(k) \right), \tag{5.29}$$

where $DNL_j(k)$ represents the *DNL(k)* error under the j^{th} DDEM configuration. From there, it is easy to show that with $P=2^{n_1}$, where n_1 is the first-stage resolution in bits, we have

$$\frac{1}{P} \sum_{j=1}^{P} DNL_j(k) = 0. \tag{5.30}$$

Thus, the gain variation error can be accurately estimated from the second-stage output codes.

5.6 Simulation Results

The proposed DDEM algorithm and test scheme are verified by numerical simulation. In simulation, a 14-bit DAC is modeled as the device under test. Its *INL(k)*s are shown in the top plot of Figure 5.8. The test system has a 6-bit DDEM first stage, a 6-bit second stage, and a 5-bit dithering DAC. The linearity of the coarse stage is less than 7 bits with the INL of $0.38LSB_C$, and the fine stage and the dithering DAC are nearly 6-bit linear. The standard deviations of comparator offsets in two stages are *0.3LSB* of the first and second stages, respectively. The gain of the residue amplifier is 32, and noise is added to the input of the DDEM ADC with a standard deviation equal to 1 *LSB* at the 14-bit level. With $P=32$, the quantization error is calculated to be less than 16-bit level. The test performance of the specified system is roughly equivalent to $n_{test}=6+5+5=16$ *bits*. Therefore, it should be capable of testing a 14-bit DAC. The *INL*(*k*) estimation errors in the bottom plot of Figure 5.8 show that, with the above configuration, the maximum *INL(k)* estimation error is about *0.5LSB* at 14-bit level and the *INL* estimation error is *0.0729LSB*.

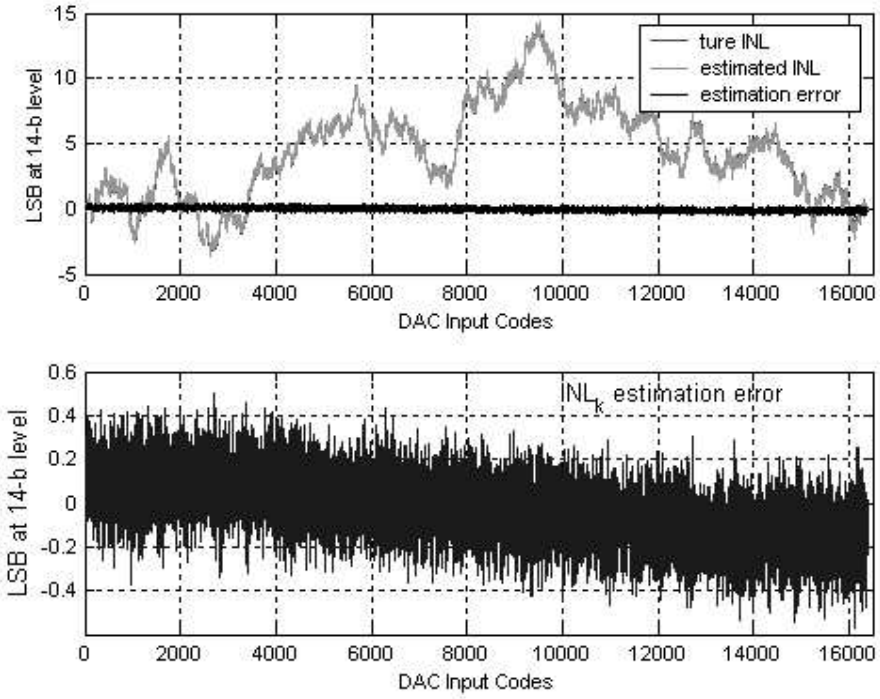


Figure 5.8. *INL(k)* estimation error with *P*=32 and 5-bit dithering DAC, 14-bit DAC under test

In order to validate the robustness of the algorithm, different DDEM ADCs are implemented. In this simulation, we use 100 different DDEM ADCs, which have the same configuration and accuracy as in the former simulation, to test 100 different 14-bit DACs. Figure 5.9 shows the relationship between the estimated *INL* values of different DACs and the true values, where the estimation errors are in the range from *-0.2425LSB* to *0.3893LSB* and the *INLs* of the DACs are in the range from *4LSB* to *18LSB*. The results show that with *P* equal to 32 and a 5-bit dithering DAC, the proposed two-step DDEM ADC is capable of testing 14-bit DACs.

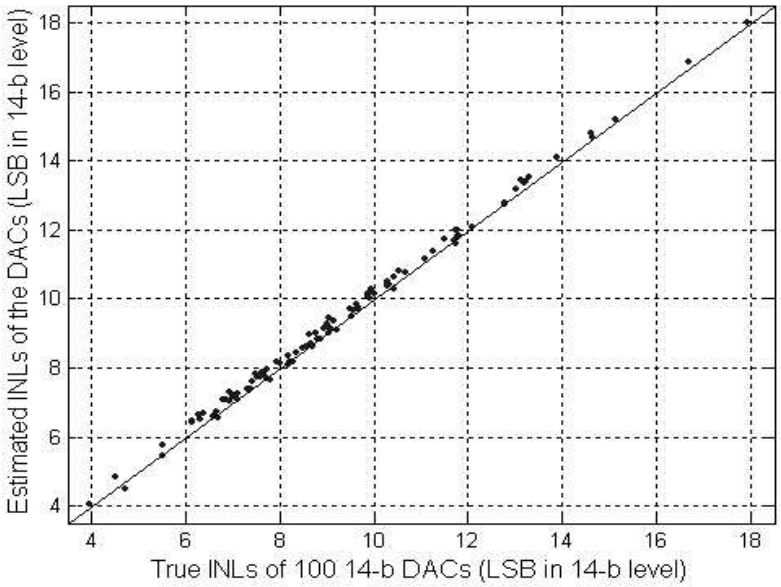


Figure 5.8. Estimated INLs vs. true INLs of 100 14-bit DACs tested by 100 DDEM ADC transition points

The effect of voltage-dependent resistance is simulated. The same system parameters are used in the test of a 14-bit DAC. The estimation errors—the difference between estimated *INL(k)* and true *INL(k)*—with and without voltage-dependent resistance are plotted in Figure 5.9 where the voltage-dependent resistance variation in the simulation is set to be 1% of the nominal value. Simulation result shows that voltage-dependent resistance in the first-stage R-string will cause considerable estimation error. Those resistances, such as switch on-resistance, need to be removed from the R-string.

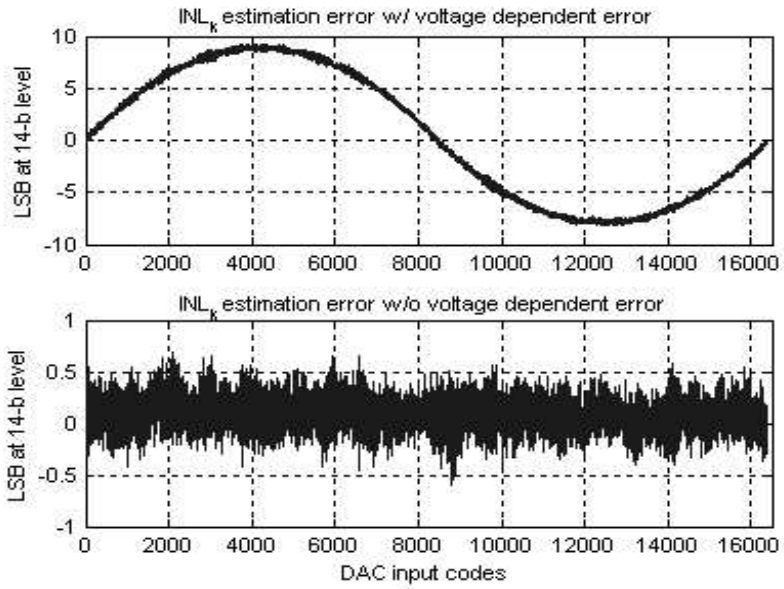


Figure 5.9. INL$_k$ estimation error with and withoug voltage-dependent errors

Figure 5.10 shows the *INL(k)* estimation errors in the cases of the actual gain of the residue amplifier equal to 32, 37, and 27, respectively. The system configurations are the same as the previous simulation. Simulation results verify that the DDEM ADC can accurately test 14-bit DACs with a large gain difference error.

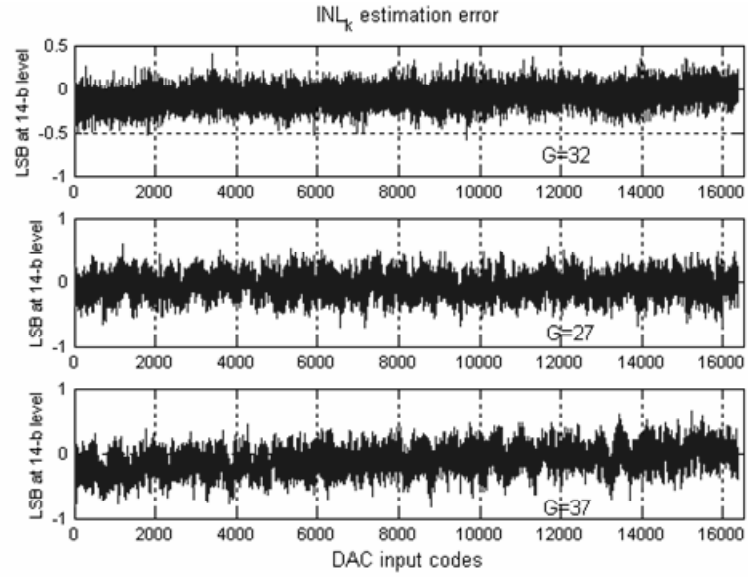


Figure 5.10. INLk estimation error when *G=32,27, and 37*

Next, the calibration of the gain variation error is simulated. Figure 5.11(a) shows the normalized gain variation against first-stage outputs simulated in Spectre. The induced test error is shown in Figure 5.11(b) where $P=2^{n_1}=64$ is set for the calibration. Figure 5.11(c) shows the estimated gain variation, and the estimation errors after calibration is shown in Figure 5.11(d). Simulation shows that the system's test ability is not degraded by the variation error after calibration.

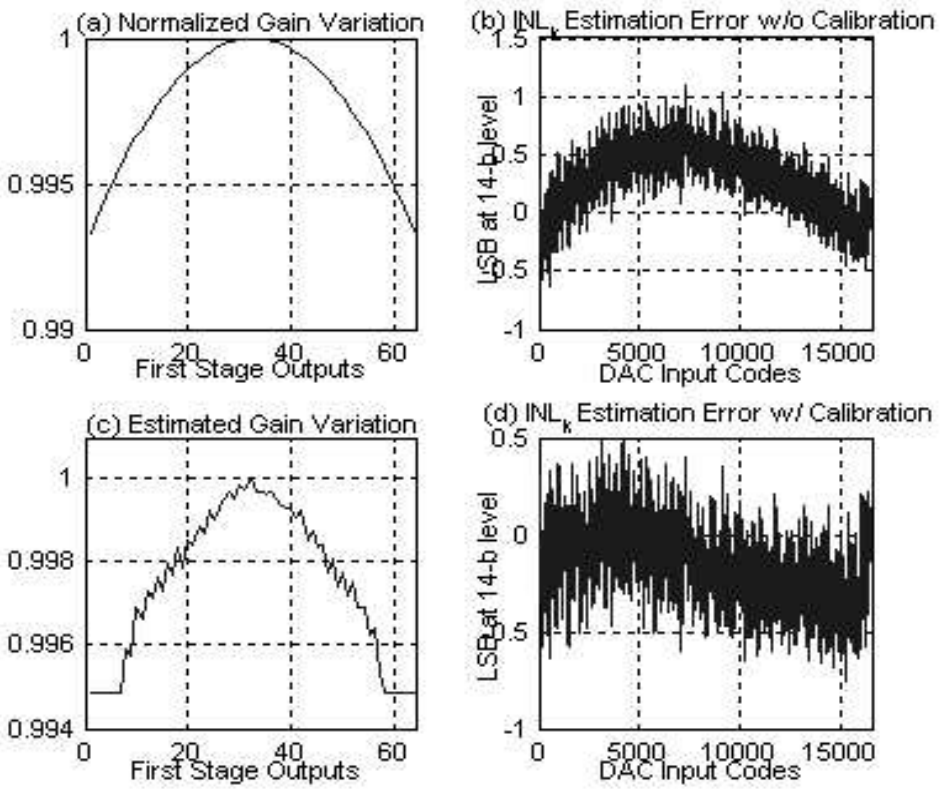


Figure 5.11. *INL_k* estimation error when *ΔG=0, 5. and -5*

5.7 Conclusions

This chapter described an on-chip BIST method for the linearity testing of high-performance DACs. This method overcomes the difficulty of DACs' on-chip testing in requiring costly high-accuracy measurement devices. Here, low-accuracy but high-speed

flash ADCs are used with the proposed DDEM algorithm. Both theoretical analysis and simulation show that this method is capable of providing high-resolution test results. In addition, it is also shown that the proposed algorithm and circuit structure can accommodate considerable circuit non-idealities. These characteristics make this method a promising solution to on-chip testing of high-precision DACs.

Chapter 6. Summary

This dissertation described a series of methods for testing mixed-signal circuits, especially linearity performance of analog-to-digital and digital-to-analog converters, with less test cost and better test performance. Among different setups for mixed-signal testing, this work investigates fully digital-compatible built-in self-test solutions. This setup uses only simple and inexpensive digital test environments incorporated with on-chip BIST circuits, which provide high-precision analog sourcing and evaluation on-chip. To make the solutions practical, the analog BIST circuits must be small, easy to design, and robust to mismatch errors and process variations. Another requirement is that the test procedure needs to be simple, so that the digital BIST circuits are simple and cost effective as well.

In chapter 3, the author propose a BIST strategy for ADC linearity testing, which is fully compatible with digital test environments using a low-cost digital tester and a simple digital DIB. Low-resolution and low-accuracy DACs (which are cost efficient) are built on-chip as source generators, and their testing performance is guaranteed by the DDEM reconfiguration technique and the testing procedure. Design of the on-chip testing circuits could be as easy as digital design because of the low accuracy requirements on the analog blocks. Simulation and experimental results demonstrate that the proposed strategy is able to test the INL_k error of 12-bit ADCs to $\pm 0.15LSB$ accuracy level using very low-accuracy DACs. In addition, the BIST strategy can be easily adopted for DAC testing if the digital comparator is replaced by an analog one, which compares the outputs of the DAC under test with the outputs of the source generator.

Chapter 5 described an on-chip BIST method for the linearity testing of high-performance DACs. This method overcomes the difficulty of DACs' on-chip testing requiring costly high-accuracy measurement devices. Instead, here low-accuracy but high-speed flash ADCs were used with the proposed DDEM algorithm. Both theoretical analysis and simulation showed that this method is capable of providing high-resolution test results. In addition, it is also shown that the proposed algorithm and circuit structure can accommodate considerable circuit non-idealities. These characteristics make this method another promising solution to on-chip testing of high-precision DACs.

In addition, a reduced-code testing method for pipeline ADC linearity testing aimed at greatly reducing the test time was investigated in chapter 4. System parameters, such as the comparator offsets and the stage gains, were identified based on a very small set of measurements. The full-code linearity performance was then reconstructed using the identified model. Both simulation and experimental results showed that the new method cuts down the data acquisition time by a factor of several hundreds while still being able to achieve satisfactory testing accuracy. Although this research examined 1-bit/stage pipeline ADC, the method can be extended to multi-bit per stage pipeline ADCs as well.

Overall, cost-effective, performance-robust analog circuit design and efficient test algorithm development are two critical requirements for high-performance mixed-signal BIST. With the proposed low-cost, high-performance circuit design and test algorithms, the built-in self-test solutions described in this dissertation are promising.

Appendix A: Verilog code for logic control block

```
// Verilog HDL for "DDEMDAC_hx", "LogControl3" "functional"

module LogControl3(start, modsel, clk, comp, valid, result, value) ;

input start; // start=1 to perform conversion

input modsel; // modsel=0: binary search mode, modsel=1: ramp output mode

input clk; // clock input for the controller

input comp; // feedback signal from comparator

output valid; // valid=1 when conversion finished

output [12:0] result; // 7-bit MSB plus 6-bit LSB result to output

output [12:0] value; // 7-bit MSB plus 6-bit LSB to DDEM DAC


reg  [2:0] state; // current state in state machine

reg  [12:0] mask; // bit to test in binary search

reg  [12:0] result; //hold partally converted


// state assignment

parameter sWait=0, sInit=1, sEval=2, sRamp=3, sDone=4;


// synchronous design

always @(posedge clk) begin

        if (!start) state<=sWait; // stop and reset if start=0

        else case (state)
```

```
sWait: begin
        state<= sInit;
        mask <= 13'b0000000000000;
        result <= 13'b0000000000000;
        end
sInit: begin
        if (modsel) begin
                state<= sRamp; // enter convert state next
                mask <= 13'b0000000000000; // reset mask to all zero
        end
        else begin
                state<= sEval;
                mask <= 13'b1000000000000; // reset mask to MSB only
        end
        result <= 13'b0000000000000; // clear result
end
sEval: begin
        if (!comp) result <= result | mask;
        mask <= mask>>1;
        if (mask[0]) state <= sDone;
end
sRamp: begin
        result[12:6] <= result[12:6]+1;
```

```
                    if (result[12:6]==7'b1111111) state <= sDone;
                end
                sDone: ;
            endcase
    end

    assign value = result | mask; // (result so far) OR (bit to try)
    assign valid = state==sDone;   // indicate when finished

    endmodule
```

Appendix B: Verilog code for column decoder and DDEM mux

```
// Verilog HDL for "DDEMDAC_hx", "LogMSB4bdec" "functional"

module LogMSB4bdec(inputcode, control, C1, C2) ;
input [3:0] inputcode;
input [3:0] control;
output [15:0] C1;
output [15:0] C2;

wire [15:0] thcod1;
wire d;
reg [15:0] thcod2;


assign thcod1[0] = 1'b1;
assign thcod1[1] = inputcode[0] | inputcode[1] | inputcode[2] | inputcode[3];
assign thcod1[2] = inputcode[1] | inputcode[2] | inputcode[3];
assign thcod1[3] = (inputcode[0] & inputcode[1]) | inputcode[2] | inputcode[3];
assign thcod1[4] = inputcode[2] | inputcode[3];
assign thcod1[5] = ((inputcode[0] | inputcode[1]) &  inputcode[2]) | inputcode[3];
assign thcod1[6] = (inputcode[1] & inputcode[2]) | inputcode[3];
```

```
assign thcod1[7] = (inputcode[0] & inputcode[1] & inputcode[2]) | inputcode[3];
assign thcod1[8] = inputcode[3];
assign thcod1[9] = (inputcode[0] | inputcode[1] | inputcode[2]) & inputcode[3];
assign thcod1[10] = (inputcode[1] | inputcode[2]) & inputcode[3];
assign thcod1[11] = ((inputcode[0] & inputcode[1]) | inputcode[2]) & inputcode[3];
assign thcod1[12] = inputcode[2] & inputcode[3];
assign thcod1[13] = (inputcode[0] | inputcode[1]) & inputcode[2] & inputcode[3];
assign thcod1[14] = inputcode[1] & inputcode[2] & inputcode[3];
assign thcod1[15] = 1'b0;
assign d = inputcode[0] & inputcode[1] & inputcode[2] & inputcode[3];
always @( thcod1 or control)
begin
case (control)
        4'b0000: thcod2 = thcod1;
        4'b0001: thcod2 = {thcod1[0], thcod1[15:1]};
        4'b0010: thcod2 = {thcod1[1:0], thcod1[15:2]};
        4'b0011: thcod2 = {thcod1[2:0], thcod1[15:3]};
        4'b0100: thcod2 = {thcod1[3:0], thcod1[15:4]};
        4'b0101: thcod2 = {thcod1[4:0], thcod1[15:5]};
        4'b0110: thcod2 = {thcod1[5:0], thcod1[15:6]};
        4'b0111: thcod2 = {thcod1[6:0], thcod1[15:7]};
        4'b1000: thcod2 = {thcod1[7:0], thcod1[15:8]};
        4'b1001: thcod2 = {thcod1[8:0], thcod1[15:9]};
```

```
        4'b1010: thcod2 = {thcod1[9:0], thcod1[15:10]};
        4'b1011: thcod2 = {thcod1[10:0], thcod1[15:11]};
        4'b1100: thcod2 = {thcod1[11:0], thcod1[15:12]};
        4'b1101: thcod2 = {thcod1[12:0], thcod1[15:13]};
        4'b1110: thcod2 = {thcod1[13:0], thcod1[15:14]};
        4'b1111: thcod2 = {thcod1[14:0], thcod1[15]};
endcase
end
assign C1[0] = thcod2[0] & (thcod2[1] | d);
assign C2[0] = thcod2[0] | d;
assign C1[1] = thcod2[1] & (thcod2[2] | d);
assign C2[1] = thcod2[1] | d;
assign C1[2] = thcod2[2] & (thcod2[3] | d);
assign C2[2] = thcod2[2] | d;
assign C1[3] = thcod2[3] & (thcod2[4] | d);
assign C2[3] = thcod2[3] | d;
assign C1[4] = thcod2[4] & (thcod2[5] | d);
assign C2[4] = thcod2[4] | d;
assign C1[5] = thcod2[5] & (thcod2[6] | d);
assign C2[5] = thcod2[5] | d;
assign C1[6] = thcod2[6] & (thcod2[7] | d);
assign C2[6] = thcod2[6] | d;
assign C1[7] = thcod2[7] & (thcod2[8] | d);
```

```
assign C2[7] = thcod2[7] | d;

assign C1[8] = thcod2[8] & (thcod2[9] | d);

assign C2[8] = thcod2[8] | d;

assign C1[9] = thcod2[9] & (thcod2[10] | d);

assign C2[9] = thcod2[9] | d;

assign C1[10] = thcod2[10] & (thcod2[11] | d);

assign C2[10] = thcod2[10] | d;

assign C1[11] = thcod2[11] & (thcod2[12] | d);

assign C2[11] = thcod2[11] | d;

assign C1[12] = thcod2[12] & (thcod2[13] | d);

assign C2[12] = thcod2[12] | d;

assign C1[13] = thcod2[13] & (thcod2[14] | d);

assign C2[13] = thcod2[13] | d;

assign C1[14] = thcod2[14] & (thcod2[15] | d);

assign C2[14] = thcod2[14] | d;

assign C1[15] = thcod2[15] & (thcod2[0] | d);

assign C2[15] = thcod2[15] | d;


endmodule
```

References

[1] Moore, Gordon E., "Cramming more components onto integrated circuits," Electronics, vol. 38, no. 8, pp. 114-117, April 19, 1965.

[2] "International Technology Roadmap for Semiconductors", 2005 Edition, http://public.itrs.net/

[3] John Sweeney, Alan Tsefrekas, "Reducing test cost through the use of digital testers for analog tests," Proceedings IEEE Int. Test Conf., 2005, pp. 9-17.

[4] Mark Burns and Gordon W. Roberts, "An Introduction to Mixed-Signal IC Test and Measurement," Oxford University Press 2001.

[5] IEEE Std. 1241-2000, "IEEE Standard for Terminology and Test Methods for Analog-to Digital Converters."

[6] IEEE Std. 1658 Draft, "IEEE Standard for terminology and test methods for digital-to-analog converters," 2005.

[7] E. Balestriere, S. Moisa, S. Rapuano, "DAC static parameter specifications – some critical notes," P1658 meeting archive.

[8] M. J. Ohletz, "Hybrid Built In Self Test (HBIST) for Mixed Analog/Digital Integrated Circuits," Proc. European Test Conference, 1991, pp.307-16.

[9] N. Nagi, A. Chatterjee, J. Abraham, "A Sigature Analyzer for Analog and Mixed-Signal Circuits," Proc. ICCD, 1994, pp. 284-87.

[10] K. Damm, W. Anheier, "HBIST of nonlinear analog building blocks in mixed-signal circuits," International Mixed Signal Testing Workshop, June 1995, pp. 257-62.

[11] S. Sunter & N. Nagi, "A Simplified Polynomial-Fitting Algorithm for DAC and ADC BIST", Proceedings of ITC, pp.389-395, November 1997

[12] K. Arabi, and B. Kaminska, "Oscillation Built-In Self-Test Scheme for Functional and Structural Testing of Analog and Mixed-Signal Integrated Circuits," *IEEE International Conference*, November 1997, pp. 786-795.

[13] K. Arabi & B. Kaminska, "Parametric and catastrophic fault coverage of analog circuits in Oscillation-test methodology," 15th IEEE VLSI Test Symposium. Monterey CA, pp. 166-17 (1997).

[14] D. Vazquez, G. Huertas, G. Leger, A. Rueda, J. L. Huertas, "On-Chip Evaluaion of Oscillation-Based test," Analog IC and Signal Proc., Kluwer Pub., Vol 33, 2002

[15] K. Arabi, and B. Kaminska, "Oscillation Built-In Self-Test Scheme for Functional and Structural Testing of Analog and Mixed-Signal Integrated Circuits," *IEEE International Conference*, November 1997, pp. 786-795.

[16] M.F. Toner and G.W. Roberts, "A Frequency Response, Harmonic Distortion, and Intermodulation Distortion Test for BIST of a Sigma-Delta ADC", *IEEE Trans. Circuits & Systems II*, Vol. 43, No. 8, pp. 608-613, 1996.

[17] J. Blair, "Histogram Measurement of ADC Nonlinearities Using Sine Waves", *IEEE Transactions on Instrumentation and Measurement*, Vol. 43, No. 3, pp. 373-383, June 1994.

[18] Yun-Che Wen and Kuen-Jong Lee, "An on-chip ADC Test Structure," Design, Automation and Test Conference and Exihibition in Europe (DAS'00), 2000, pp. 221-225.

[19] S. Bernard, F. Azais, Y. Bertrand and M. Renovell, "Efficient on-chip generator for linear histogram BIST of ADCs", *Proc. International Mixed-Signal Testing Workshop*, 2001.

[20] Benoit Provost, Edgar Sanchez-Sinencio, "On-chip Ramp Generators for Mixed-Signal BIST and ADC Self-Test", IEEE Journal of Solid-State Circuits, Vol. 38, No. 2, pp. 263-273, Feb. 2003.

[21] F. Azais, S. Bernard, Y. Bertrand, X. Michel and M. Renovell, "A low-cost adaptive ramp generator for analog BIST applications", *Proc. VLSI Test Symposium*, 2001.

[22] .S. Bernard, F. Azais, Y. Bertrand and M. Renovell, "A High Accuracy Triangle-Wave Signal Generator for On-Chip ADC Testing", *Proc. Of the Seventh IEEE European Test Workshop*, 2002.

[23] Y. C. Wen, "Novel Step-Ramp Signal for Testing ADCs and DACs", *Proc. of the 14th International Conference on Mixed Design of Integrated Circuits and Systems*, 2007.

[24] M. Renovell, F. Azais., S. Bernard, and Y. Bertrand, "Hardware Resource Minimization for a Histogram-based ADC BIST", *IEEE VLST Test Symposium*, Apr. 2000.

[25] E. S. Erdogan & S. Ozev, "An ADC-BiST Scheme Using Sequential Code Analysis", *Proceeding of the Conference on Design, Automation and Test*, 2007.

[26] L. Jin, K. Parthasarathy, T. Kuyel, D. Chen and R. Geiger, "Accurate testing of analog-to-digital converters using low linearity signals with stimulus error identification and removal," *IEEE Trans. Instrum. Meas.*, vol. 54, pp. 1188 – 1199, June 2005.

[27] L. Jin, D. J. Chen, R. L. Geiger, "Code-Density Test of Analog-to-Digital Converters Using Single Low-Linearity Stimulus Signal," *Proc. of 25th IEEE VLSI Test Symposium*, pp. 303-310, 2007

[28] B. Olleta, H. Jiang, D. Chen and R. L. Geiger, "A deterministic dynamic element matching approach for testing high-resolution ADCs with low-accuracy excitations," *IEEE Trans. Instrum. Meas.*, vol. 55, pp. 902 – 915, June 2006.

[29] H. Jiang, B. Olleta, D. Chen and R. L. Geiger, "Testing High Resolution ADCs with Low Resolution/ Accuracy Deterministic Dynamic Element Matched DACs," *IEEE Trans. Instrum. Meas.*, vol. 56, pp. 1753 – 1762, Oct. 2007.

[30] H. Jiang, H. Xing, D. Chen and R. Geiger, "A low-cost on-chip stimulus source with 16-bit equivalent linearity for ADC built-in self-test," TECHCON 2005.

[31] R. J. Van De Plassche and H. H. Schouwenaars, "A Monolithic 14 Bist A/D Converter", *IEEE JSSC*, Vol. SC-17, No. 6, Dec. 1982.

[32] H. T. Jensen and I. Galton, "A Low-Complexity Dynamic Element Matching DAC for Direct Digital Synthesis", *IEEE Transactions on Circuits and Systems*, Vol. 45, January 1998.

[33] R. Adams, K. Q. Nguyen and K. Sweetland, "A 113-dB SNR Oversampling DAC with Segmented Noise-Shaped Scrambling", *IEEE JSSC*, Vol. 33, No. 12, December 1998.

[34] R. E. Radke, A. Eshraghi and T. S. Fiez, "A 14-Bit Current Mode ΣΔ DAC Based Upon Rotated Data Weighted Averaging", *IEEE JSSC*, Vol. 35, August 2000.

[35] B. Olleta, H. Jiang, D. Chen, R. Geiger, "Testing high resolution ADCs using deterministic dynamic element matching", *Proceedings of International Symposium on Circuits and Systems*, Vol. 1, pp. 920—923, May 2004.

[36] International Technology Roadmap for Semiconductors, 2007 Edition, http://public.itrs.net/

[37] T. Kuyel, "Linearity Testing Issues of Analog-to-Digital Converters," in *Proc. 1999 International Test Conference*, pp. 747-756, Sept 1999.

[38] IEEE Std 1241-2000, IEEE Standard for Terminology and Test Methods for Analog-to Digital Converters.

[39] Zhongjun Yu, Degang Chen, and Randall Geiger, "Pipeline ADC Linearity Testing with Dramatically Reduced Data Capture Time" *IEEE International Symposium on Circuits and Systems*, Vol. 1, pp. 792-795, May 2005.

[40] *International Technology Roadmap for Semiconductors*, 2005 edition, available.

[41] H. Jiang, B. Olleta, D. Chen and R. L. Geiger, "Testing High Resolution ADCs with Low Resolution/ Accuracy Deterministic Dynamic Element Matched DACs", 2004 *International Test Conference*, Charlotte, NC, Oct 2004.

Acknowledgements

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research during my Ph.D. study and the writing of this dissertation, in particular to the following persons and agencies:

- Dr. Degang Chen and Dr. Randall Geiger for their guidance and instructions throughout my research work and the writing of this dissertation.
- Dr. Chris Chu, Dr. Aleksandar Dogandzic, and Dr. Stephen Vardeman for their support and suggestions on my work.
- All my student colleagues for their collaborations while I completed my Ph.D.
- Semiconductor Research Corporation, National Science Foundation, Texas Instruments, and Iowa State University for their partial financial support of my doctorate work.