Built-in self-test and self-calibration for analog and mixed signal circuits

Tao Chen

Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

Part of the Electrical and Electronics Commons

Recommended Citation
https://lib.dr.iastate.edu/etd/17161

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Built-in self-test and self-calibration for analog and mixed signal circuits

by

Tao Chen

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 Geiger
Nathan Neihart
Chris Chu
Philip Jones

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University
Ames, Iowa
2019

Copyright © Tao Chen, 2019. All rights reserved.
# TABLE OF CONTENTS

LIST OF TABLES .................................................. vii
LIST OF FIGURES .................................................. viii
ACKNOWLEDGMENTS ................................................. ix
ABSTRACT .......................................................... x

CHAPTER 1. INTRODUCTION ............................................. 1
  1.1 Background ................................................... 1
  1.2 Proposed Methods ............................................. 3
    1.2.1 Algorithm Developments ................................. 4
    1.2.2 BIST Implementations ................................. 5
  1.3 Dissertation Organization ................................... 5

CHAPTER 2. ULTRAFAST STIMULUS ERROR REMOVAL ALGORITHM FOR ADC
  LINEARITY TEST .................................................. 7
  2.1 Introduction .................................................. 7
  2.2 Background ................................................... 10
    2.2.1 SEIR .................................................... 10
    2.2.2 uSMILE .................................................. 11
  2.3 USER-SMILE ................................................... 12
    2.3.1 Segmented Non-parametric Model ...................... 14
    2.3.2 Stimulus Requirement .................................. 15
    2.3.3 Constant Offset ....................................... 15
  2.4 Simulations .................................................. 16
  2.5 Conclusion ................................................... 18
# CHAPTER 3. USER-SMILE: ULTRAFAST STIMULUS ERROR REMOVAL AND SEGMENTED MODEL IDENTIFICATION OF LINEARITY ERRORS FOR ADC BUILT-IN SELF-TEST

## 3.1 Introduction

## 3.2 USER-SMILE

### 3.2.1 Modeling of ADC Linearity Errors

### 3.2.2 Stimulus Error Removal

### 3.2.3 Error Identifications and INL Construction

## 3.3 Error Analysis

### 3.3.1 Effects of Unmodeled Error in the INL Model

### 3.3.2 Additive Noise in the Input Signals

### 3.3.3 Effects of the Voltage Shift Between Two Signals

### 3.3.4 Effects of Quantization Error

## 3.4 Simulation Results

### 3.4.1 Modeling Error

### 3.4.2 Noise Effect

### 3.4.3 Voltage Shift Effect

### 3.4.4 Missing Codes Identifications

## 3.5 Discussion

## 3.6 Conclusion

# CHAPTER 4. AN ULTRAFAST MULTI-BIT/STAGE PIPELINED ADC TESTING AND CALIBRATION METHOD

## 4.1 Introduction

## 4.2 Pipelined ADC Architecture Review

## 4.3 Pipelined ADC Nonlinearity Modeling

### 4.3.1 Nonlinearity Modeling

### 4.3.2 INL Construction
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4 ADC Testing and Calibration</td>
<td>55</td>
</tr>
<tr>
<td>4.5 Simulation Results</td>
<td>59</td>
</tr>
<tr>
<td>4.6 Measurement</td>
<td>62</td>
</tr>
<tr>
<td>4.7 Conclusion</td>
<td>64</td>
</tr>
<tr>
<td>CHAPTER 5. A LOW-COST ON-CHIP BUILT-IN SELF-TEST SOLUTION FOR ADC</td>
<td></td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>66</td>
</tr>
<tr>
<td>5.2 USER-SMILE Algorithm Review</td>
<td>68</td>
</tr>
<tr>
<td>5.3 R2R DAC Design</td>
<td>70</td>
</tr>
<tr>
<td>5.3.1 Signal Generator Requirement</td>
<td>70</td>
</tr>
<tr>
<td>5.3.2 Subradix-2 Architecture</td>
<td>72</td>
</tr>
<tr>
<td>5.3.3 Voltage Shift Generation</td>
<td>75</td>
</tr>
<tr>
<td>5.4 Simulations</td>
<td>80</td>
</tr>
<tr>
<td>5.5 Measurement Results</td>
<td>83</td>
</tr>
<tr>
<td>5.5.1 Test Setup</td>
<td>83</td>
</tr>
<tr>
<td>5.5.2 R2R DAC Performance</td>
<td>84</td>
</tr>
<tr>
<td>5.5.3 ADC Testing Results</td>
<td>85</td>
</tr>
<tr>
<td>5.6 Conclusion</td>
<td>86</td>
</tr>
<tr>
<td>CHAPTER 6. AN ON-CHIP ADC BIST SOLUTION AND THE BIST-ENABLED CALIBRATION SCHEME</td>
<td></td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>88</td>
</tr>
<tr>
<td>6.2 Review of USER-SMILE Algorithm</td>
<td>91</td>
</tr>
<tr>
<td>6.3 Algorithm Implementation</td>
<td>95</td>
</tr>
<tr>
<td>6.3.1 Design Choices - Hardware vs. Software</td>
<td>95</td>
</tr>
<tr>
<td>6.3.2 Adaptation of USER-SMILE for Hardware Implementation</td>
<td>96</td>
</tr>
<tr>
<td>6.4 ADC BIST Subsystem Design</td>
<td>99</td>
</tr>
<tr>
<td>6.4.1 12-bit Resistive DAC</td>
<td>100</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>ADC performance comparison</td>
<td>64</td>
</tr>
<tr>
<td>6.1</td>
<td>ADC performance comparison</td>
<td>112</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Single ADC test results with USER-SMILE</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>100-ADC test results with USER-SMILE</td>
<td>18</td>
</tr>
<tr>
<td>3.1</td>
<td>Segmented INL model</td>
<td>23</td>
</tr>
<tr>
<td>3.2</td>
<td>3-bit ADC Transfer Function</td>
<td>25</td>
</tr>
<tr>
<td>3.3</td>
<td>Algorithm Implementation</td>
<td>26</td>
</tr>
<tr>
<td>3.4</td>
<td>INL Modeling Error</td>
<td>38</td>
</tr>
<tr>
<td>3.5</td>
<td>16-bit ADC INL Estimation (1 LSB noise)</td>
<td>39</td>
</tr>
<tr>
<td>3.6</td>
<td>1000-time 16-bit ADC INL Estimation Error(1 LSB noise)</td>
<td>40</td>
</tr>
<tr>
<td>3.7</td>
<td>INL Estimation Error with 3ppm nonconstancy</td>
<td>41</td>
</tr>
<tr>
<td>3.8</td>
<td>Missing codes effects</td>
<td>42</td>
</tr>
<tr>
<td>4.1</td>
<td>Pipelined ADC architecture</td>
<td>49</td>
</tr>
<tr>
<td>4.2</td>
<td>Sub-ADC architecture for stage 1 and stage 2</td>
<td>49</td>
</tr>
<tr>
<td>4.3</td>
<td>1st stage residue voltage transfer function in ideal case</td>
<td>52</td>
</tr>
<tr>
<td>4.4</td>
<td>1st stage residue voltage transfer function with transient errors</td>
<td>53</td>
</tr>
<tr>
<td>4.5</td>
<td>Comparison between redundant INL and the histogram-based INL</td>
<td>53</td>
</tr>
<tr>
<td>4.6</td>
<td>Block diagram of the proposed algorithm</td>
<td>56</td>
</tr>
<tr>
<td>4.7</td>
<td>Testing and calibration flowchart</td>
<td>57</td>
</tr>
<tr>
<td>4.8</td>
<td>INL testing comparison</td>
<td>60</td>
</tr>
<tr>
<td>4.9</td>
<td>INL major transition comparison: no transient noise (left), with transient error (right)</td>
<td>61</td>
</tr>
<tr>
<td>4.10</td>
<td>INL and DNL comparison</td>
<td>62</td>
</tr>
</tbody>
</table>
Figure 4.11 100 simulated ADC before and after calibration .......................... 63
Figure 4.12 Performance comparison before and after calibration ....................... 64

Figure 5.1 USER-SMILE implementation .......................................................... 70
Figure 5.2 Standard R2R DAC ................................................................. 72
Figure 5.3 Subradix-2 DAC transfer function (subradix-2 for the first 2 MSB bits) . . 72
Figure 5.4 R2R DAC Voltage Shift Generation ................................................. 76
Figure 5.5 Equivalent circuit for voltage shift generation .............................. 76
Figure 5.6 Differential R2R DAC Voltage Shift Generation ............................ 78
Figure 5.7 maximum DNL over 1000 random simulations .............................. 81
Figure 5.8 Shift constancy VS PMOS/NMOS on-resistance difference ................ 81
Figure 5.9 Shift constancy VS on-resistance local variations ......................... 82
Figure 5.10 Shift constancy VS R2R resistor variations ................................. 82
Figure 5.11 R2R DAC die photo ............................................................. 83
Figure 5.12 Testbench .............................................................................. 84
Figure 5.13 DAC output voltages ................................................................. 84
Figure 5.14 DNL of the R2R DAC ............................................................... 85
Figure 5.15 Ramp-to-ramp shifting ............................................................... 86
Figure 5.16 Ping-pong shifting .................................................................. 87

Figure 6.1 ADC BIST Subsystem Block Diagram .............................................. 90
Figure 6.2 CDAC Array of SARADC ......................................................... 101
Figure 6.3 Algorithm Unit Design ............................................................... 105
Figure 6.4 ADC output ............................................................................. 107
Figure 6.5 DNL/INL correlation ................................................................. 108
Figure 6.6 Zoom in on one spike .............................................................. 109
Figure 6.7 INL correlations for histogram and BIST ..................................... 110
Figure 6.8 Performance Comparison .......................................................... 111
ACKNOWLEDGMENTS

First of all, I would like to express my deepest gratitude to my major professor Dr. Degang Chen for his guidance and support throughout my research work.

I would also like to thank my committee members Dr. Randall Geiger, Dr. Nathan Neihart, Dr. Chris Chu, Dr. Philip Jones for their contributions to this work.

I would like to thank Dr. Jiming Song for his support, advice and encouragement.

I would like to thank Xiankun Jin, Dr. Doug Garrity, Dr. Colin McAndrew from NXP (former Freescale) for their collaboration, support and inspiration. I would like to thank Dr. Silva Jose-Martinze from Texas A&M University for the collaboration and support.

I would like to thank my friends and colleagues Chulhyun Park, Dadian Zhou, Hao Meng, Shravan Chaganti, Siva Kumar Sudani, Tao Zeng, Chongli Cai, Ailing Mei, Nanqi Liu, Qianqian Wang, Yuming Zhuang and anyone who helped and supported me in various aspects.

I would like to thank Semiconductor Research Corporation (SRC), NXP Semiconductors (including Freescale semiconductor), National Science Foundation (NSF), Texas Instruments, TSMC, Texas A&M University and Iowa State University for the financial and technical support of my research work.

Finally, I would like to thank my wife Hongjuan Zhang for her love and support. I would like to thank my parents Anning Chen, Li Li and my sister Fei Chen for their support and sacrifices.
Analog-to-digital converters (ADC) are one of the most important components in modern electronic systems. In the mission-critical applications such as automotive, the reliability of the ADC is critical as the ADC impacts the system level performance. Due to the aging effect and environmental changes, the performance of the ADC may degrade and even fail to meet the accuracy requirement over time. Built-in self-test (BIST) and self-calibration are becoming the ultimate solution to achieve lifetime reliability. This dissertation introduces two ADC testing algorithms and two ADC built-in self-test circuit implementations to test the ADC integral nonlinearity (INL) and differential nonlinearity (DNL) on-chip.

In the first testing algorithm, the ultrafast stimulus error removal and segmented model identification of linearity errors (USER-SMILE) is developed for ADC built-in self-test, which eliminates the need for precision stimulus and reduces the overall test time. In this algorithm, the ADC is tested twice with a nonlinear ramp, instead of using a linear ramp signal. Therefore, the stimulus can be easily generated on-chip in a low-cost way. For the two ramps, there is a constant voltage shift in between. As the input stimulus linearity is completely relaxed, there is no requirement on the waveform of the input stimulus as long as it covers the ADC input range. In the meantime, the high-resolution ADC linearity is modeled with segmented parameters, which reduces the number of samples required for achieving high-precision test, thus saving the test time. As a result, the USER-SMILE algorithm is able to use less than 1 sample/code nonlinear stimulus to test high resolution ADCs with less than 0.5 least significant bit (LSB) INL estimation error, achieving more than 10-time test time reduction. This algorithm is validated with both board-level implementation and on-chip silicon implementation.

The second testing algorithm is proposed to test the INL/DNL for multi-bit-per-stages pipelined ADCs with reduced test time and better test coverage. Due to the redundancy characteristics of
multi-bit-per-stages pipelined ADC, the conventional histogram test cannot estimate and calibrate
the static linearity accurately. The proposed method models the pipelined ADC nonlinearity as
segmented parameters with inter-stage gain errors using the raw codes instead of the final output
codes. During the test phase, a pure sine wave is sent to the ADC as the input and the model
parameters are estimated from the output data with the system identification method. The modeled
errors are then removed from the digital output codes during the calibration phase. A high-speed
12-bit pipelined ADC is tested and calibrated with the proposed method. With only 4000 samples,
the 12-bit ADC is accurately tested and calibrated to achieve less than 1 LSB INL. The ADC
effective number of bits (ENOB) is improved from 9.7 bits to 10.84 bits and the spurious-free
dynamic range (SFDR) is improved by more than 20dB after calibration.

In the first circuit implementation, a low-cost on-chip built-in self-test solution is developed
using an R2R digital-to-analog converter (DAC) structure as the signal generator and the voltage
shift generator for ADC linearity test. The proposed DAC is a subradix-2 R2R DAC with a
constant voltage shift generation capability. The subradix-2 architecture avoids positive voltage
gaps caused by mismatches, which relaxes the DAC matching requirements and reduces the design
area. The R2R DAC based BIST circuit is fabricated in TSMC 40nm technology with a small area
of 0.02mm². Measurement results show that the BIST circuit is capable of testing a 15-bit ADC
INL accurately with less than 0.5 LSB INL estimation error.

In the second circuit implementation, a complete SAR ADC built-in self-test solution using the
USER-SMILE is developed and implemented in a 28nm automotive microcontroller. A low-cost 12-
bit resistive DAC with less than 12-bit linearity is used as the signal generator to test and calibrate
a SAR ADC with a target linearity of 12 bits. The voltage shift generation is created inside the
ADC with capacitor switching. The entire algorithm processing unit for USER-SMILE algorithm
is also implemented on chip. The final testing results are saved in the memory for further digital
calibration. Both the total harmonic distortion (THD) and the SFDR are improved by 20dB after
calibration, achieving -84.5dB and 86.5dB respectively. More than 700 parts are tested to verify
the robustness of the BIST solution.
CHAPTER 1. INTRODUCTION

Analog and mixed signal (AMS) circuits have been one of the most important components in modern electronic systems. There is a growing demand for chips containing digital-to-analog converters (DACs) and analog-to-digital converters (ADCs) in the area of automotive, 5G communication, Internet of things (IoT), etc.

In the automotive industry, advanced driver-assistance systems are being implemented to improve vehicle and road safety. Additionally, autonomous driving systems are being developed. With increasingly more electronic systems being integrated into vehicles, the reliability of the integrated circuits is becoming an important concern. The International Organization for Standardization (ISO) 26262 standard has therefore been developed to address the functional safety of automotive electronic systems (1). However, the hardware implementations required to meet the standard are missing. Each vehicle usually lasts more than 15 years. Aging effects, environmental changes and mechanical vibrations have significant impacts on the electronic systems performance. Therefore, the electronic systems need to be periodically tested, calibrated, and even replaced. In each vehicle, more and more ADCs are used for sensors, radar, engine control and many other applications. Realizing efficient built-in self-test, self-diagnosis and self-calibration for high performance ADCs is urgently needed.

1.1 Background

There are mainly two categories of specifications for testing the ADC. The first category is the static testing using DC or nearly DC input voltages. Static testing involves testing the static or pseudo-static performance of the ADC, which includes the integral nonlinearity (INL), differential nonlinearity (DNL), offset, gain error, etc. The other category is the spectral testing (or dynamic testing). In this category, the ADC is tested with a pure sine wave as the input stimulus for
the signal-to-noise ratio (SNR), total harmonic distortion (THD) and spurious free dynamic range (SFDR). Among these tests, the INL/DNL testing is the most time-consuming and the most expensive one. In the industrial standard histogram test, a slow linear ramp or a very low-frequency pure sine wave is generated from a high-precision tester as the input signal to the ADC (2; 3; 4; 5). The test stimulus has to be at least 2 or 3 bits more accurate than the ADC itself to achieve acceptable test results. In addition, it usually requires more than 20 hits (samples) per code to average out the testing noise. Therefore, the test time is very long. As the resolution of the ADC increases, the accurate signal generation becomes more and more difficult. The number of samples needed increases exponentially with the number of bits of the ADC. In addition, the sampling rate usually reduces as the resolution increases. All of these result in very long test times, and correspondingly, high test costs. As a result, realizing built-in self-test for ADCs is becoming an urgent need but it faces the bottleneck challenge of generating very accurate input signals in a small chip area. The on-chip stimulus generation and test time reduction have to be achieved for low-cost efficient BIST solutions. Many methods have been proposed to address the testing challenges for ADCs.

A straightforward way of achieving BIST is to implement the linear ramp generator on-chip. In (6), a signal generator based on delta-sigma modulation is proposed to test both the ADC and DAC on-chip. Very linear ramp generators based on capacitor charging are investigated and developed in (7; 8; 9; 10; 11; 12; 13) with different architectures. Fairly good linearity can be achieved with simple circuits. In the most recent publication, a 14.5-bit ENOB ramp generator is achieved, which can be used to test a 11-bit ADC with 0.3 LSB INL estimation accuracy (13). It is very difficult to test ADCs with more than 12-bit resolution using conventional histogram methods on-chip.

To achieve practical on-chip implementation, the hardware cost has to be addressed. In (14; 15; 16), methods are proposed to simplify the operations, reduce the digital processing time and reduce the memory size in the histogram test. Simple RC circuits are used to test the ADC in (17). An oscillation method is developed in (18) to test the ADC. But the accuracy is limited. Random noise is used as the input to the ADC to predicate the nonlinearity based on statistical analysis (19; 20; 21). It usually leads to a large number of samples, which significantly increases the
The stimulus error identification and removal (SEIR) algorithm is proposed in (22; 23). The ADC is tested with two nonlinear ramps but there is a constant voltage shift between the two ramps. The input nonlinearity can be identified and removed so that the ADC nonlinearity can be accurately estimated. But the test time is still very long since it is based on the histogram test. In addition, the SEIR algorithm requires the input signal to have smooth nonlinearity with respect to the voltage so that the nonlinearity can be modeled with a linear model using a small number of basis functions. A segmented DAC usually has discontinuities in the major transitions. Therefore, the DAC-generated nonlinear signals cannot be used in the SEIR algorithm. A polynomial fitting is used to model the ADC nonlinearity in (24) to reduce the test time with a low-resolution DAC as the signal generator. High resolution ADCs are tested with low-resolution dynamic element matched DAC in (25). All these methods enable on-chip signal generation, but the test time is still significant.

Many methods are proposed to reduce the test time (26; 27; 28; 29; 30; 31; 32; 33; 34; 35). In (26; 27), the ADC nonlinearity are estimated based on the dynamic test results. (28; 29) explore the pipelined ADC architecture and develop methods to reduce the test time. Spectral test and local histogram test are combined in (30). Kalman filtering is used in (32) to achieve more than 10 times test time reduction. An ultrafast linearity test method is developed in (35) with a non-parametric segmented model for high-resolution ADCs. These methods reduce the test time with certain assumptions or requirements on the ADCs, but an accurate signal source is still needed.

Despite these methods, efficient on-chip testing is still a challenge. There is a strong need to achieve low-cost signal generation and significant test time reduction at the same time.

### 1.2 Proposed Methods

In this work, several enabling technologies are developed to realize the built-in self-test on-chip for ADC static linearity. The thesis can be divided into two categories. The first category is the algorithmic theoretical development and analysis, which includes chapter 2, chapter 3 and chapter
4. The second category is the on-chip circuit adaptation and implementations, which includes chapter 5 and 6.

1.2.1 Algorithm Developments

Two ADC testing algorithms are developed. The ultrafast stimulus error removal and segmented model identification of linearity errors (USER-SMILE) is first introduced. The stimulus linearity requirement is completely relaxed. Instead of using a linear ramp or a pure sine wave, the proposed method takes a nonlinear signal as the input but the ADC samples it twice. In the two sampling, there is a constant voltage shift in between. In the USER-SMILE algorithm, the input nonlinearity is directly canceled without the need for modeling the input nonlinearity. Therefore, the USER-SMILE algorithm overcomes the fundamental limitations of using signal generators with smooth linearity in the SEIR algorithm. Any waveform including DAC-generated nonlinear signals can be used in the USER-SMILE algorithm as long as the signal can cover the ADC input range. In addition, the ADC nonlinearity is estimated using a segmented model. Much fewer number of parameters can be used to represent the ADC nonlinearity. As a result, the number of total samples is significantly reduced during the entire testing. The relaxed signal generator and reduced test time enables the built-in self-test.

The second algorithm is targeted for multi-bit-per-stage pipelined ADC testing and calibration. The pipelined ADC has a segmented architecture but the USER-SMILE algorithm cannot be directly applied to the pipelined ADC due to its redundancy. The conventional histogram test cannot accurately estimate the INL/DNL in the redundant regions of the transfer function. As a result, the histogram test results are unable to calibrate the ADC correctly. A new method is developed to model and correct the internal architecture errors of the pipelined ADC by using the raw codes in each stage. It significantly reduces the test time and test results can be used to correctly calibrate the ADC linearity errors.
1.2.2 BIST Implementations

Two ADC BIST circuits implementations are developed and validated. In the first circuit, a BIST solution based on subradix-2 R2R DAC is developed. In the USER-SMILE algorithm, the most critical part is the constant voltage shift generation. In the proposed circuit, an additional resistor is added at the output node to add the constant voltage shift to the DAC output. And the subradix-2 architecture avoids large positive gaps, which relaxes the DAC matching requirement while still covering the ADC input range. The proposed DAC is capable of testing a 15-bit ADC with less than 0.5 LSB estimation errors with the USER-SMILE algorithm.

In the second circuit, a complete SAR ADC on-chip BIST implementation is developed. An untested resistor DAC is used as the signal generator. The 12-bit SAR ADC is to be tested. The constant voltage shift is created inside the SAR ADC with capacitor switching. The digital control, the computation engine and the memory are all implemented on-chip as a self-contained solution. The final results and the calibration data are all stored in the memory. It achieves a self-contained ADC full-parametric on-chip testing and accurate test-based self-calibration.

1.3 Dissertation Organization

The rest of the thesis is composed of a collection of papers with editorial modifications. Chapter 2 is a paper published in IEEE VLSI Test Symposium (VTS) (36). The USER-SMILE method is introduced. The proposed method is verified in statistical simulations. Chapter 3 is a paper published in IEEE Transactions on Circuits and Systems I (TCAS-I) (37). The USER-SMILE algorithm is further improved. A complete analysis and error derivation are presented in this chapter. Chapter 4 is a manuscript submitted to IEEE Transactions on Instrumentation and Measurement (TIM). It introduces a testing method for multi-bit-per-stage pipelined ADC. Both simulation and measurement results are provided to validate the proposed method. Chapter 5 is a manuscript submitted to IEEE Transactions on Instrumentation and Measurement. A novel R2R DAC architecture is developed. A prototype circuit is designed and fabricated in TSMC 40nm technology as a proof of concept. Chapter 6 is a paper published in IEEE International Test
Conference (ITC)(38), in which a complete ADC BIST solution for SAR ADCs is presented. In this work, I did the theoretical analysis, the algorithm adaptation, the algorithm engine design and the calibration logic on-chip implementation. Xiankun Jin was in charge of project management and sub-system integration, finite state machine, etc. Chapter 7 is the general conclusion.
CHAPTER 2. ULTRAFAST STIMULUS ERROR REMOVAL ALGORITHM FOR ADC LINEARITY TEST

Linearity test of an analog-to-digital converter (ADC) can be very challenging because it requires a signal generator substantially more linear than the ADC under test. For high performance ADCs, the overall manufacturing cost could be dominated by the long test time and the high-precision test instruments. This chapter introduces the ultrafast stimulus error removal and segmented model identification of linearity errors (USER-SMILE) method for high resolution ADC linearity test, allowing the stimulus signal’s linearity requirement to be significantly relaxed and the test time to be reduced by orders of magnitude compared to the state-of-art histogram method. The USER-SMILE algorithm uses two nonlinear but functionally related input signals as ADC excitations and uses a stimulus error removal technique to recover test accuracy. The USER-SMILE algorithm also uses the ultrafast segmented model identification of linearity errors (uSMILE) approach to dramatically reduce test time while achieving test accuracy and coverage superior to the histogram method. The USER-SMILE algorithm is validated by extensive simulation with different types of ADCs, different resolution levels, and different types of input signals including nonlinear ramps, nonlinear sine waves and even random input signals. Statistical simulation results show that for a 16-bit SAR ADC, with two 1 hit/code nonlinear ramp signals, the INL test error is within +/- 0.4LSB.

2.1 Introduction

The analog-to-digital converter (ADC) is one of the most important analog and mixed signal (AMS) products (39). Accurate linearity test of ADC can be very challenging, especially for high resolution ADCs (5). As the manufacturing cost goes down, the test cost becomes more and more dominant in the overall cost. The ADC test cost is mainly due to the test equipment cost and the
test time. To test the ADC nonlinearity, the state-of-art histogram method uses a highly linear signal generated from the high-precision automated test equipment (ATE) (40; 4; 2; 3). The signal source is required to be substantially more linear than the device under test (DUT). It becomes more and more difficult to generate linear source as the ADC resolution goes high. Furthermore, the histogram method requires much more samples than the number of transitions in the ADC. As the industry standard, the histogram test usually uses tens or even hundreds hits per code to accurately test the ADC nonlinearity, which results in a very long data acquisition time. For high resolution ADC (higher than 16-bit), it is usually not practical to fully test the ADC linearity in production test due to the extremely long test time.

The stringent requirement on the input signal linearity and the extremely long time test become the challenges in the AMS test. Significant works have been done to overcome these challenges. Recently, researchers have developed different ways to address the stringent linearity requirement of the input signal. In (6), the author employed the delta-sigma modulation technique to generate the highly linear input signal. However, it is not easy to design such a signal generator as the ADC’s resolution or speed goes high. The design complexity often increases the cost. In the contrast, some researchers have put efforts on algorithms to relax the stimulus linearity requirement. In (41; 22; 23), stimulus error identification and removal (SEIR) algorithm is proposed to test precision ADC using nonlinear stimulus. It has been proved that 7-bit linear ramp signal can be used to test high resolution ADC and achieve more than 16 bits accuracy. A constant offset is required to identify the nonlinear components in the signal source. As the requirement on the input signal linearity is relaxed, built-in-self-test for ADC full code INL/DNL test becomes practical. However, SEIR is based on the histogram method, which means the data acquisition time is still very long.

Other than relaxing the input signal requirement, lots of efforts have been made to reduce the test time. In (27; 42), a method was proposed to use fast Fourier transform (FFT) test to estimate the ADC’s INL. In (30), the INL can be estimated with a combined spectral and histogram method. A system identification approach is proposed in (28) to evaluate the nonlinearity of a pipeline ADC. In (24), the author uses the polynomial fitting method with low resolution input signal to test the
ADC nonlinearity. In (31), Goyal, et al introduced a selective code measurement method to reduce the test time of SAR ADCs. However, all above methods or similar ones reduce the test time by sacrificing other test aspects, so that they cannot achieve similar coverage or test accuracy than the histogram method. Therefore, the application is very limited. An ultrafast segmented model identification of linearity errors (uSMILE) (35) algorithm was proposed recently to take a system identification approach to capture both linear and nonlinear errors in the ADC. With the concept of the segmented non-parametric model, the algorithm can reduce the test data by a factor of over 100 and achieve a test accuracy superior to the histogram method. However, it still requires highly linear input signal source.

In summary, the existing solutions have at least one of the following issues: long test time, highly linear stimulus, low accuracy or coverage. Currently there is no valid solution to resolve all these issues at the same time. Test time is an important factor in AMS test and accurate input signal cannot be easily implemented on chip. There is a strong need to test the ADC using an easy-to-implement signal generator with much less test time. In this chapter, a new algorithm combing the concept of SEIR and uSMILE is proposed for accurate linearity test with dramatically reduced test time and also relaxes the requirement on source linearity. Two nonlinear input signals with constant offset between them are applied to the ADC. Two sets of ADC output codes will be generated. Segmented non-parametric model is used to represent the final INL. Rather than directly finding the INL of the ADC, the INL is indirectly evaluated from the difference of the segmented INL by subtracting the two sets of the output codes. The test accuracy and coverage is superior to the state-of-art histogram method.

The following of this chapter is organized as follows. Section 2.2 reviews two fundamental algorithms: the SEIR and the uSMILE algorithm. Section 2.3 presents the proposed algorithm. Section 2.4 shows the simulation results. And section 2.5 concludes this chapter.
2.2 Background

The SEIR algorithm relaxes the linearity requirement on the input signal by injecting a constant offset in the input signal. The uSMILE algorithm significantly reduces the test time with the segmented non-parametric model. These two algorithms are the basis of the USER-SMILE algorithm and will be reviewed below.

2.2.1 SEIR

Define the nonlinear ramp signal to be \( x(t) \) and normalize the time so that \( t_0 = 0 \ t_{N-2} = 1 \) and \( T_k \) is the transition level for output from code \( k-1 \) to code \( k \). Then it can be expressed as

\[ x(t) = T_0 + (T_{N-1} - T_0)t + F(t), \tag{2.1} \]

where \( F(t) \) is the nonlinear component of the ramp signal. The nonlinear component can be estimated using a set of basis function \( F(t) = \sum d_j F_j(t) \). The INL measured by nonlinear ramp will be:

\[ INL'_k = (N - 2)t_k - k = INL_k - F(t_k), \tag{2.2} \]

where \( INL'_k \) is the estimated INL using nonlinear ramp and \( INL_k \) is the corrected INL after removing the nonlinear component in the input signal.

\[ INL_k = (N - 2)t_k + F(t_k) - k. \tag{2.3} \]

For two ramp signal, \( x_1(t) \) and \( x_2(t) \) have a constant offset \( \alpha \).

\[ x_2(t) = x_1(t) - \alpha. \tag{2.4} \]

Then, \( x_1(t) \) and \( x_2(t) \) can be expressed with transition points and the nonlinear components as:

\[ x_1(t) = T_0 + (T_{N-2} - T_0)t + F(t), \tag{2.5} \]
\[ x_2(t) = T_0 + (T_{N-2} - T_0)t + F(t) - \alpha. \]  

(2.6)

For the transition level from code \( k - 1 \) to code \( k \), we can get \( T_k = x_1(t_{k,1}) = x_2(t_{k,2}) \). Replace \( x_1(t_{k,1}) \) and \( x_2(t_{k,2}) \) with equation (2.1), equation (2.7) is obtained.

\[ T_k = T_0 + (T_{N-2} - T_0)t_{k,1} + F(t_{k,1}) \]
\[ = T_0 + (T_{N-2} - T_0)t_{k,2} + F(t_{k,1}) - \alpha. \]  

(2.7)

Since the number of equations is much larger than the number of the unknowns, least square can be used to estimate the unknowns. The coefficients of \( F \) and the constant offset \( \alpha \) can be obtained from equation (2.8).

\[
\{\hat{a}_1, \hat{a}_2, \hat{a}_3, \cdots, \hat{a}_M, \hat{\alpha}\} = \arg \min \left\{ \sum \left( (N - 2)(t_{k,2} - t_{k,1}) - \left[ \sum a_j (F_j(t_{k,1}) - F_j(t_{k,2})) + \alpha \right] \right)^2 \right\}. 
\]  

(2.8)

Then, the nonlinear component of the input ramp signal has been identified. The INL can be reconstructed by removing the nonlinear component from input signal. Either \( INL_k^{(1)} \) or \( INL_k^{(2)} \) can be used for the evaluated INL.

\[ INL_k^{(1)} = (N - 2)t_k^{(1)} + \sum_{j=1}^{M} \hat{a}_j F_j(t_k^{(1)}) - k, \]  

(2.9)

\[ INL_k^{(2)} = (N - 2)t_k^{(2)} + \sum_{j=1}^{M} \hat{a}_j F_j(t_k^{(2)}) - k - \hat{\alpha}. \]  

(2.10)

2.2.2 uSMILE

Different from the SEIR, the uSMILE algorithm was proposed to significantly reduce the test time as well as achieve better accuracy. By a system identification approach with a segmented non-parametric model, the algorithm is able to capture the nonlinearity of the ADC with much less test data.
The segmented non-parametric model in the INL curve is to break down the INL into different segments. For example, an INL curve can be broken into 64 segments if 6 MSB bits are used. For each MSB segment, this short INL curve can be further broken into smaller segments (for example, 5 ISB bits). Similarly, the ISB can be broken into LSB (for example, 5 LSB bits). For each segment, there is a corresponding error term. Define the MSB error term to be $E_M(C_{MSB})$, where $C_{MSB}$ is the code of the MSB bits. Then, the errors for 64 segments are $E_M(0)$, $E_M(1)$, $\cdots$, $E_M(63)$ corresponding to the MSB code. Similarly, $E_I$ and $E_L$ are defined for ISB and LSB errors respectively and they are also called “segmented INL” in the following of this chapter. The final INL value for code $C$ will be:

$$INL(C) = E_M(C_{MSB}) + E_I(C_{ISB}) + E_L(C_{LSB}).$$

(2.11)

For an input signal, there will be an ideal expected output $C_{exp}$. Due to the ADC nonlinearity, the actual output code becomes $C$. Then, the input output relationship can be created:

$$C_{exp} - C + q = E_M(C_{MSB}) + E_I(C_{ISB}) + E_L(C_{LSB}),$$

(2.12)

where $q$ is the noise.

In order to estimate the INL with the segmented non-parametric model, the linear input signal information is used. With a pure sine wave as the ADC’s input signal, a linear ADC will get a linear sine wave in the output. However, due to the nonlinearity existing in the ADC, the output will have harmonics and other components. From the actual ADC’s output code, the DC and fundamental components can be extracted in the frequency domain and an ideal ADC is constructed using the DC and fundamental only. In other words, after removing the DC and fundamental in the frequency domain, everything else are just noise and nonlinear components in the actual ADC. After identifying the MSB, ISB and LSB errors, the final full-code INL can be constructed.

### 2.3 USER-SMILE

This section proposes the USER-SMILE algorithm to identify the INL/DNL using nonlinear input signal and with much less test data, and achieve better test coverage and accuracy than the
histogram test. Two identical input signals with constant offset between them are applied to the ADC. In the USER-SMILE, by subtracting the two sets of output data, the input signal information is no longer needed. Any error or nonlinearity in the stimulus is completely removed. At this point, there’s no assumption on the input signal linearity. Some other restrictions on the input signal will be discussed later.

Apply two input signals $V_{in}^{(1)}$ and $V_{in}^{(2)}$ to the ADC with a constant offset $\alpha$.

$$V_{in}^{(1)} = V_{in}^{(2)} + \alpha. \quad (2.13)$$

The converted output codes from ADC are $C^{(1)}$ and $C^{(2)}$. Then, equation (2.14) and (2.15) are obtained:

$$V_{in}^{(1)} + w^{(1)} = T^{(1)} + q^{(1)} = C^{(1)} \cdot V_{LSB} + INL^{(1)} + q^{(1)}, \quad (2.14)$$

$$V_{in}^{(2)} + w^{(2)} = T^{(2)} + q^{(2)} = C^{(2)} \cdot V_{LSB} + INL^{(2)} + q^{(2)}, \quad (2.15)$$

where the noise $w^{(1)}$ and $w^{(2)}$ are the input-referred noise and $T$ is the transition voltage for output code from $C$ to $C + 1$. And $q$ in the equation is the quantization noise. With the segmented non-parametric model, the INL can be broken into MSB segments, ISB segments and LSB segments. The total nonlinearity error for code $C$ can be written into the same format in equation (2.11).

Then, the equation (2.14) and (2.15) can be expressed as:

$$V_{in}^{(1)} + w^{(1)} = C^{(1)} \cdot V_{LSB} + E_{M}(C_{MSB}^{(1)}) \cdot V_{LSB} + E_{I}(C_{ISB}^{(1)}) \cdot V_{LSB} + E_{L}(C_{LSB}^{(1)}) \cdot V_{LSB} + q^{(1)}, \quad (2.16)$$

$$V_{in}^{(2)} + w^{(2)} = C^{(2)} \cdot V_{LSB} + E_{M}(C_{MSB}^{(2)}) \cdot V_{LSB} + E_{I}(C_{ISB}^{(2)}) \cdot V_{LSB} + E_{L}(C_{LSB}^{(2)}) \cdot V_{LSB} + q^{(2)}. \quad (2.17)$$
By subtracting the two equations (2.16) and (2.17), we can get equation (2.18):

\[
\begin{align*}
V_{in}^{(1)} - V_{in}^{(2)} + w^{(1)} - w^{(2)} &= V_{LSB} \{ C^{(1)} - C^{(2)} + E_M(C_{MSB}^{(1)}) + E_L(C_{LSB}^{(1)}) \\
&+ E_L(C_{LSB}^{(1)}) - E_M(C_{ISB}^{(2)}) - E_I(C_{LSB}^{(2)}) \\
&- E_L(C_{LSB}^{(1)}) \} + q^{(1)} - q^{(2)}. 
\end{align*}
\]  

(2.18)

Replace the \(V_{in}^{(1)} - V_{in}^{(2)}\) with \(\alpha\) and re-arrange the equation:

\[
\begin{align*}
C^{(1)} - C^{(2)} - \frac{\alpha}{V_{LSB}} &= -\{ E_M(C_{MSB}^{(1)}) + E_I(C_{ISB}^{(1)}) + E_L(C_{LSB}^{(1)}) \\
&- E_M(C_{MSB}^{(2)}) - E_I(C_{ISB}^{(2)}) - E_L(C_{LSB}^{(2)}) \} \\
&+ (q^{(2)} - q^{(1)} + w^{(1)} - w^{(2)})/V_{LSB}. 
\end{align*}
\]  

(2.19)

Assume that the input-referred noise is at a certain level and the quantization noise will be “whitened”. So the term \(q^{(2)} - q^{(1)} + w^{(1)} - w^{(2)}\) can be considered as one random noise. For this overdetermined system, the least square algorithm can be used to find the unknowns \(E_M, E_I\) and \(E_L\). With least square method, the noise term will be effectively averaged out. Then, the full code INL can be constructed.

Some crucial parts in the algorithm are discussed below:

### 2.3.1 Segmented Non-parametric Model

USER-SMILE leverages the segmented non-parametric model in the uSMILE algorithm. It treats the ADC itself as a black box and accurately models the actual INL curve. Any linear errors (mismatch and gain) and nonlinear error (voltage coefficients or code dependent parasitics) can be captured. Any advantages and restrictions in the segmented non-parametric model are also applied to the USER-SMILE algorithm. Therefore, with the segmented non-parametric model, the USER-SMILE algorithm can significantly reduce the test time. For a 16-bit ADC with 6-5-5 segmentation (6 MSB bits, 5 ISB bit and 5 LSB bits), only 128 unknowns need to be solved and the results can accurately reflect the actual INL. However, there are also some limitations. The
segmented non-parametric model is intended for high resolution ADCs whose architecture facilitates a segmented structure of the INL curve. So, the USER-SMILE method is not intended for flash ADC or delta sigma ADC. For other types of ADCs such as SAR ADC, Cyclic ADC, Pipeline ADC, the USER-SMILE algorithm works well.

2.3.2 Stimulus Requirement

It is usually difficult to design a fast and highly linear signal generator on chip. If we can relax the requirement on the stimulus, the signal generator design complexity and cost can be significantly reduced. By subtracting the two equations (2.16) and (2.17), the information of input signal is no longer needed. And there is no assumption on the signal linearity or the signal shape.

However, there’s still some constraints to the stimulus. Take an extreme case: if we use a fixed voltage to test the ADC, the ADC is always producing a similar code. There is no way to get the information of other codes. So, to achieve good estimation accuracy, the input signal should cover as most codes. Each segment (in MSB, ISB and LSB) needs to have sufficient coverage.

The signal generator design is simplified with the above consideration. It can be a very nonlinear signal generator but needs to cover most of the ADC input range. A low cost, nonlinear, ramp generator or sine wave generator can be easily built on chip as the stimulus with minimal area overhead.

2.3.3 Constant Offset

As showed in the previous derivation, a constant offset is required. Different methods have been proposed before for offset injections and good constancy can be achieved on chip.

The algorithm needs to know the exact value of the offset. Due to process variations, the actual offset value may be different with the simulated value. In this case, the offset $\alpha$ can be simply estimated by the average difference between the output codes $C^{(1)}$ and $C^{(2)}$.

Usually, we want to minimize the amount of offset due to the offset generator design. Larger offset will increase the design cost or complexity. To make the USER-SMILE algorithm work, the
offset cannot be too small. If we have a very small offset (for example, a few LSB), the MSB segments will hardly change after applying this offset. The $E_M$ in this set of data are the same and they cancel each other so that there is no information from these data. Ideally we expect all segments to be changed after applying the offset and all data are fully used. If we use 6-5-5 segmentation (6 MSB bits, 5 ISB bit and 5 LSB bits), the ideal offset value is 1 MSB + 1 ISB + 1 LSB, which is 1057 LSB. Due to the nonlinearity of the ADC or the variation of the offset, for each set of data, the segments may not be all different. But the amount of such data is small and the effect on the estimation accuracy is neglectable. So, making the offset value to be slightly larger than 1 MSB is the best choice.

2.4 Simulations

To verify the algorithm, extensive simulations have been done on different ADC architectures (SAR, Pipeline, and Cyclic) with various resolutions. The simulation results show that the algorithm works well with different architectures. SAR ADC is particularly studied due to its wide usage, high resolution and low power. A 16-bit SAR ADC is modeled with random capacitor mismatches. The true INL is constructed from the transition voltage. In all the following simulations, 0.5 LSB input-referred noise is added.

With two 1 hit/code nonlinear ramp signals, the INL (end point fitting) estimation is shown in Figure 2.1(a). The true INL of the ADC is plotted with the red line. In this ADC, the INL is about 1 LSB. The estimated INL from the USER-SMILE is plotted in the blue line. From the plot, we can see that the blue line matches the red line very well. The estimation error is defined to be the difference between the estimated full-code INL with the true INL. Figure 2.1(b) shows the estimation error in USER-SMILE. The maximum estimation error is about +/- 0.15 LSB. It shows that the USER-SMILE method produces good estimation accuracy over all the codes.

To further verify the algorithm with different ADCs, a large number of simulations have been done. The test uses two 1 hit/code nonlinear ramp signals to test different ADCs and 100 test results are randomly selected. For each INL curve, the maximum estimation error and the minimum
estimation error are recorded over all codes and shown in Figure 2.2(a). From the figure, the average maximum estimation error in USER-SMILE is around 0.2 LSB with few of them over 0.3 LSB. Figure 2.2(b) shows a different view of the estimation accuracy. The x-axis is the true INL and the y-axis is the estimated INL from the USER-SMILE. Ideally, if the estimated INL is the same as the true INL, this point will lie on the y=x line (the black line). Due to the estimation error, these points will be away from this line. From the figure, the red points (estimated INL from the USER-SMILE) is very close to the y=x line, which means the USER-SMILE has very good accuracy. For ADCs with different performance (INL from 0.6 LSB to 4.6 LSB), the test accuracy stays the same. From the production test point of view, the USER-SMILE method will guarantee less yield loss.

With two 1 hit/code ramp signals for a 16-bit ADC, the overall test data are only 131k points. The statistical study shows that the USER-SMILE algorithm is robust over different ADCs (including good and bad ADCs). And from all tests, the maximum estimation error for INL is within +/-0.4 LSB.
2.5 Conclusion

A fast and cost-effective method for ADC linearity test is presented in this chapter. The USER-SMILE algorithm allows the stimulus signal’s linearity requirement to be significantly relaxed and the test time to be reduced by orders of magnitude compared to the state-of-art histogram method, thus greatly reducing the test cost. The simulation demonstrates that the USER-SMILE can achieve superior test coverage and accuracy. With the USER-SMILE algorithm, a new BIST solution can be practical, which doesn’t require highly accurate and expensive ATE as the signal generator. Furthermore, it simplifies the test board and interface design.
CHAPTER 3. USER-SMILE: ULTRAFAST STIMULUS ERROR REMOVAL AND SEGMENTED MODEL IDENTIFICATION OF LINEARITY ERRORS FOR ADC BUILT-IN SELF-TEST

Linearity testing of analog-to-digital converters (ADCs) is very challenging and expensive due to the stringent linearity requirement on the stimulus and the extremely long test time. This chapter introduces a novel method for ADC static linearity testing, allowing the stimulus linearity requirement to be significantly relaxed and the test time to be significantly reduced compared to the state-of-art histogram method. Two nonlinear but functionally related input signals are used as the ADC’s excitation and a stimulus error removal technique is used to recover test accuracy. With a segmented non-parametric integral nonlinearity model, this method requires much fewer parameters to accurately represent the nonlinearity. The proposed algorithm has been extensively verified and correlated in simulations. This method not only enables low-cost production testing but can also be used for low-cost on-chip built-in self-test. This method is limited to ADCs with segmented architecture such as SAR ADCs, Pipeline ADCs, and Cyclic ADCs.

3.1 Introduction

The analog-to-digital converter (ADC) is one of the most important analog and mixed signal (AMS) components. The ADCs have been deeply embedded in the modern system-on-chip (SoC). With ever increasing applications in Internet of things (IoT) and automotive, the ADCs’ volume has grown significantly. Testing these ADCs is necessary to guarantee the performance before shipment. However, it is challenging and expensive to fully test the ADC’s performance due to various reasons. With such a large volume, reducing the test cost of ADCs becomes significant and necessary. In addition, for some critical applications such as automotive, aerospace and medical areas where reliability and safety requirement are extremely high, one-time test may not be enough since such
applications usually last for decades and the degradation of performance or environmental changes may cause a severe influence on the system. There is a strong need to achieve built-in self-test (BIST) capability to not only reduce the test cost, but also guarantee the reliability of the ADCs. The testing of digital circuits has been well addressed with automatic test pattern generation (ATPG) and logic BIST. However, it remains a challenge to test the ADC on chip (39).

The ADC testing involves static linearity tests, such as integral nonlinearity (INL) and differential nonlinearity (DNL) and dynamic linearity tests, such as signal to noise ratio (SNR), total harmonic distortion (THD) and spurious free dynamic range (SFDR) (3; 4; 5). Among these tests, the static linearity test is the most time-consuming one. To test the ADC static nonlinearity, the conventional histogram method uses a highly linear ramp or sine wave generated from the precision automated test equipment (ATE) (3; 4). The signal source is required to be 3 to 4 bits more linear than the device under test (DUT) to accurately test the ADC since any error in the input signal will be treated as part of the ADC’s linearity error in the conventional histogram test. To test ADCs beyond 16 bits, the signal generator has to be 19-bit linear or better, which is expensive and difficult to achieve. In addition, the ADC usually takes tens of samples per code to reduce the noise effect in a histogram test. For 16-bit ADC, there are $2^{16} - 1 = 65,535$ transitions to be tested and the number of samples will be close to or even more than millions, which requires seconds or more test time depending on the ADC’s sampling rate. Therefore, the cost associated with the test time is very high.

All these requirements pose significant challenges to achieve BIST. First of all, generating a highly linear signal on chip for ADC testing is difficult or even impossible for a high resolution ADC. The cost of building such a signal generator could be much more than the ADC itself. In addition, the test time is not saved if the conventional histogram test is still used. To achieve BIST in an efficient way, these two challenges have to be solved: the test stimulus linearity and long test time. Significant work has been done to overcome these challenges in the past decades. Researchers have achieved some limited BIST features (36; 43; 44; 9; 45; 46; 47; 48; 49; 6; 22; 11; 50; 51).
Some are addressing the stringent linearity requirement of the input signal. In (6), the delta-sigma modulation technique is applied to generate the highly linear input signal. However, it is not easy to design such a signal generator as the ADC’s resolution or speed increases. The design complexity often increases the cost. A low-cost linear ramp generator is proposed in (11). However, it takes multiple calibrations to achieve better precision and the linearity is still limited. In the contrast, some researchers have focused on algorithms to relax the stimulus linearity requirement. In (22; 23), the stimulus error identification and removal (SEIR) algorithm is proposed to test precision ADCs using nonlinear stimulus. It has been proven that a 7-bit linear ramp signal can be used to test a high resolution ADC and achieve more than 16 bits accuracy with this method. Rather than using a linear ramp, it uses two nonlinear ramps with a constant voltage shift in between. The nonlinearity in the two ramps will be identified by the algorithm and removed. Therefore, accurate linearity test can be achieved with nonlinear signals. Many practical applications have been presented in (51; 52; 53; 54) for SEIR. However, SEIR is based on the histogram method, which means that the data acquisition time is still very long.

In addition to the input signal requirement, the test time reduction is another challenge (29; 27; 42; 55; 35). Fast Fourier transform (FFT) test is used to estimate the ADC’s INL (27; 42). A system identification approach is proposed in (28) to evaluate the nonlinearity of a pipeline ADC. Design for test (DfT) methods are introduced in (55) to reduce the test time of calibrating the pipeline ADCs. Model-based testing for ADCs are developed in (56). Spectral and histogram methods are combined in (30) to reduce the test time. Segmented polynomial fitting method is developed in (24) with low resolution input signal to test the ADC nonlinearity. In (31), a selective code measurement method is introduced to reduce the test time of SAR ADCs. In (50), a ramp generator based on servo-loop method is developed to test the pipeline ADC with reduced test time by the reduced-code linearity test technique. However, these methods reduce the test time by sacrificing other test aspects such as test accuracy or test coverage compared to histogram test. An ultrafast segmented model identification of linearity errors (35) algorithm is proposed to take a system identification approach to capture both linear and nonlinear errors in the ADC. With the
segmented non-parametric model, the algorithm can reduce the test time by a factor of over 100 and still achieves a test accuracy superior to the histogram method. However, this method still requires highly linear input signal.

In this chapter, the ultrafast stimulus error removal and segmented model identification of linearity errors (USER-SMILE) algorithm is presented. This chapter is an expansion based on a previous work (36). The contribution of this chapter includes a complete derivation of the algorithm, improvement of algorithm, error analysis and extensive simulations and correlations. The USER-SMILE algorithm uses two nonlinear input signals for the ADC under test. One signal is shifted by a constant voltage with respect to the other nonlinear signal. By subtracting the output codes of the ADC for the two signals, the exact value of the input signal is canceled. The difference between the two output codes represents the nonlinearity difference of these two codes. This nonlinearity of ADC will be represented by a segmented non-parametric INL model. The model parameters will be identified with least square (LS) method. After the identification of the model parameters, the full-code INL/DNL can be constructed. This method is targeted for the built-in self-test which can be used both in production testing to save the test time and the test cost and in the field testing to realize self test in the entire life time of the ADC. The proposed method still has some limitations. Due to the segmented model, this method will work for the ADCs with segmented architectures such as SAR ADCs, Pipeline ADCs and Cyclic ADCs. It will not work for flash ADC or Delta Sigma ADCs.

The remainder of this chapter is organized as follows: Section 3.2 presents the proposed algorithm with mathematical equations. Section 3.3 provides the error analysis. Section 3.4 gives the simulation results and compare them with error analysis. Section 3.5 discusses the limitation and the practical implementation of the proposed method. Conclusion is drawn in section 3.6.

### 3.2 USER-SMILE

In this section, the details of the USER-SMILE algorithm will be presented. The modeling of the ADC nonlinearity errors will be explained first. Then, it will be shown that the input stimulus
error will be removed by two functionally related stimuli. At the end, the parameters of the ADC INL model will be identified and used for constructing the INL.

3.2.1 Modeling of ADC Linearity Errors

To test the nonlinearity of the ADC, all transition levels need to be identified to obtain the DNL and INL. Industry standard histogram method uses a sine wave or a ramp signal which is sufficiently linear (usually more than 3 bits) than the device under test (DUT). And tens of samples per code are used in order to average the noise. The histogram method shows significant inefficiency. For high resolution ADCs, the number of transitions increases exponentially. And the sampling rate is usually slower for higher resolution ADCs.

However, for high resolution ADCs, the number of components used to build the ADC is usually small. And these small number of components determine the entire ADC’s performance. The “segmented non-parametric” model is proved to be an efficient way of modeling the ADC linearity errors (35) with such characteristics. Instead of modeling the circuits inside the ADC, it models the ADC’s INL with a segmented non-parametric model. For an N-bit ADC, it uses a small number of parameters instead of the $2^N - 1$ transition levels to represent the INL.

![Figure 3.1 Segmented INL model](image.png)
The top plot of Figure 3.1 shows a typical INL plot of a binary-weighted SAR ADC. The black vertical lines break the INL into multiple same-width segments, defined as MSB (most-significant bits) segment. For example, if the first 4 bits are used to determine the segments, there will be $2^4 = 16$ MSB segments. Each MSB segment has an average INL value which is defined as $E_M(k)$ for the $k$-th segment. These MSB segmented errors, $E_M$, are from linear errors or nonlinear errors of the ADC or combination of them. With $N_M$ bits used for MSB segments, the ADC is treated like a $N_M$-bit flash ADC.

For each MSB segment, there is a smaller INL curve (Figure 3.1). Within this segment, the MSB code remains the same and only lower bits are converting. For most ADCs except flash ADC and sigma-delta ADC, the lower-bit code are determined with the same sub-ADC or the same lower-bit circuits. Therefore, the errors contributed from the lower bits will repeat for each MSB segment. As shown in Figure 3.1 top, all 16 segments have the same shape. There is an assumption that the errors from lower bits are not affected by the MSB code.

Similarly, the smaller segment, defined as ISB (intermediate significant bits), can be further broken into smaller LSB (least-significant bits) segments. The ISB segments and LSB segments are shown in the middle and the bottom of Figure 3.1. The INL for code $C$ can be then defined as

$$INL(C) = E_M(C_{MSB}) + E_I(C_{ISB}) + E_L(C_{LSB})$$

where $C$ is the final ADC output code; $E_M$, $E_I$ and $E_L$ are the MSB, ISB and LSB segmented errors respectively; $C_{MSB}$, $C_{ISB}$ and $C_{LSB}$ are the MSB, ISB and LSB segment codes. In (3.1), the segment codes are decimal representations of the corresponding binary codes. In the rest of this chapter, the ADC used in simulations and measurements are all binary-weighted ADCs.

This model enables fewer number of parameters to represent the full-code DNL/INL. In most cases, the ADC itself has some nonlinear errors coming from parasitics and secondary effects of transistors or capacitors. If these errors are large enough, it will affect the INL shape. The MSB segmented INL is treating the ADC as a flash. So the MSB INL model can capture these errors.
3.2.2 Stimulus Error Removal

In Figure 3.2, a 3-bit ADC transfer curve is used to explain the notation and the relation among the transition voltages, the input signal and the quantization error. The red curve is the actual ADC transfer curve and the black dotted line is the ideal transfer curve with same initial and end point. Suppose that there is an input signal $x$ and the ADC output code is $C = 4$. The INL for code 4 is $[T(3) - T_i(3)]/V_{LSB}$, where $T$ is the actual transition level, $T_i$ is the ideal transition level and $V_{LSB}$ is ideal 1 LSB voltage. For convenience, $T(C)$ stands for transition voltage from code $C$ to code $C + 1$. So, $T(0)$ is the transition voltage from code 0 to code 1 and $T(0) = T_i(0)$ in this case. The middle for code $C$ is defined as $V_{mid}(C)$. And the quantization error is therefore defined as $n_q = V_{mid}(C) - x$ when the sampled voltage is $x$ and the output code is $C$. We can then have the relation:

$$ x + n_q = V_{mid}(C) = T(C - 1) + V_{LSB} \cdot \frac{1 + DNL(C)}{2} $$

$$ = T_i(C - 1) + V_{LSB} \cdot \{C - 1 + INL(C) + \frac{1 + DNL(C)}{2}\} $$

$$ = T_i(0) + V_{LSB} \cdot \{C + \frac{INL(C) + INL(C + 1) - 1}{2}\} $$

where $DNL(C) = INL(C + 1) - INL(C)$.
In the precision ADC testing, the input linearity requirement is very high thus making the test cost very high. For built-in self-test purpose, building such highly linear input source is challenging or not practical. Instead, functionally related excitation can be used to relax the linearity requirement.

![Signal Generator](image1.png)

Figure 3.3 Algorithm Implementation

The implementation of the algorithm is shown in Figure 3.3. The signal generator generates an output signal $V_{\text{sig}}$. In the first time, the switch $s_1$ is turned on and switch $s_2$ is turned off. The signal passes through an adder and the ADC input $x_1 = V_{\text{sig}} + \alpha$. In the second time, the switch $s_2$ is turned on and the switch $s_1$ is turned off. The signal generator is directly connected to the ADC input so that $x_2 = V_{\text{sig}}$. In these two samples, $x_1$ and $x_2$ are unknown but there is a constant voltage shift $\alpha$ between them.

$$x_1 - x_2 = \alpha. \quad (3.3)$$

Considering the additive noise, the two input signals can be expressed as

$$x_1 + n_{a1} + n_{q1} = T(0) + V_{\text{LSB}} \cdot \left\{ C_1 + \frac{\text{INL}(C_1) + \text{INL}(C_1 + 1) - 1}{2} \right\}, \quad (3.4)$$

$$x_2 + n_{a2} + n_{q2} = T(0) + V_{\text{LSB}} \cdot \left\{ C_2 + \frac{\text{INL}(C_2) + \text{INL}(C_2 + 1) - 1}{2} \right\} \quad (3.5)$$

where $n_{a1}$ and $n_{a2}$ are the additive noise for $x_1$ and $x_2$. $C_1$ and $C_2$ are the corresponding output codes in the two samples.
The amount of the voltage shift is also unknown and will be identified in the proposed method. Subtract (3.5) from (3.4), (3.6) can be obtained:

\[
\frac{x_1 - x_2 + n_a1 - n_a2 + n_q1 - n_q2}{V_{LSB}} + C_2 - C_1 = \frac{INL(C_1 + 1) + INL(C_1) - INL(C_2 + 1) - INL(C_2)}{2}.
\] (3.6)

In (3.6), both \(n_a1\) and \(n_a2\) are random additive noise. The subtraction of two independent random variables with same variance will also give an random term with doubled variance. \(n_q1\) and \(n_q2\) are the quantization error for these two conversions. However, the quantization error is not white. Assuming that the additive noise is large enough (more than 0.3 LSB rms), the entire term \(n_a1 - n_a2 + n_q1 - n_q2\) can be treated as one random noise because the quantization error can be effectively “whitened” by the additive noise (57). So, these four terms can be replaced by a single variable \(n_{all}\). Combining (3.3) and (3.6) and replacing the noise terms, we can obtain

\[
\frac{\alpha}{V_{LSB}} + n_{all} + C_2 - C_1 = \frac{INL(C_1 + 1) + INL(C_1) - INL(C_2 + 1) - INL(C_2)}{2}.
\] (3.7)

In this equation, we can notice that the input information is no longer needed. The input linearity requirement is completely relaxed. The voltage shift \(\alpha\) will be identified; the output codes are already available; and the INL are the unknowns to be solved.

### 3.2.3 Error Identifications and INL Construction

For an \(N\)-bit ADC, there are \(2^N - 1\) transition levels and there are \(2^N - 3\) INL values (with end-point fitting, the first and last transitions are both 0s by definition). With the “segmented non-parametric” INL model, the number of unknowns to represent the full-code INL can be significantly reduced as introduced in previous sections. Replacing the INL in (3.7) with the INL model (3.1),
\( (3.8) \) can be obtained:

\[
\begin{align*}
\alpha/V_{\text{LSB}} + n_{\text{all}} &= C_1 - C_2 + \frac{\text{INL}(C_1 + 1) + \text{INL}(C_1) - \text{INL}(C_2 + 1) - \text{INL}(C_2)}{2} \\
&= C_1 - C_2 \\
&\quad + \frac{E_M((C_1 + 1)_{\text{MSB}}) + E_I((C_1 + 1)_{\text{ISB}}) + E_L((C_1 + 1)_{\text{LSB}})}{2} \\
&\quad - \frac{E_M((C_2 + 1)_{\text{MSB}}) + E_I((C_2 + 1)_{\text{ISB}}) + E_L((C_2 + 1)_{\text{LSB}})}{2} \\
&\quad - \frac{E_M(C_{1\text{MSB}}) + E_I(C_{1\text{ISB}}) + E_L(C_{1\text{LSB}})}{2} \\
&\quad - \frac{E_M(C_{2\text{MSB}}) + E_I(C_{2\text{ISB}}) + E_L(C_{2\text{LSB}})}{2} \\
&= \text{INL}(0) + \text{INL}(2N - 1) = 0 \quad \text{for an N-bit ADC.}
\end{align*}
\]

where \( C_{1\text{MSB}} \) stands for the MSB code of code \( C_1 \) and \( (C_1 + 1)_{\text{MSB}} \) stands for the MSB code of code \( C_1 + 1 \). ISB and LSB codes are defined in the same way.

For each pair of input signals, one such equation can be obtained. With \( M \) pairs of input signals, \( M \) equations will be formed and \( M \) is much larger than the number of unknowns to be solved. \( C_1 \) and \( C_2 \) are then two vectors of output codes. For \( N_M \)-bit MSB, \( N_I \)-bit ISB and \( N_L \)-bit LSB segmentation, there are total \( 2^{N_M} + 2^{N_I} + 2^{N_L} \) unknowns. Take a 12-bit ADC as an example, 4-4-4 (MSB-ISB-LSB bits) segmentation has only 48 unknowns.

The amount of voltage shift \( \alpha \) is still unknown. With a large set of output codes, the value of \( \alpha \) can be estimated by the average value of \( C_1 - C_2 \) for quick estimation. A better estimation is to make the voltage shift as an unknown and identify it in the least square method together with the segmented model parameters. The LS solutions for these unknowns can be expressed as

\[
\begin{align*}
\{ \hat{E}_M(0), \hat{E}_M(1), \ldots, \hat{E}_I(0), \hat{E}_I(1), \ldots \hat{E}_L(0), \hat{E}_L(1), \ldots, \hat{\alpha} \} \\
= \arg \min \left\{ \sum_{k=1}^{M} \left[ \frac{\alpha}{V_{\text{LSB}}} + C_2(k) - C_1(k) + \frac{\text{INL}(C_2(k) + 1) + \text{INL}(C_2(k))}{2} - \frac{\text{INL}(C_1(k) + 1) + \text{INL}(C_1(k))}{2} \right]^2 \right\}
\end{align*}
\]
With all these model parameters identified, the full-code INL can be constructed using (3.1):

$$I\hat{NL}(C) = \hat{E}_M(C_{MSB}) + \hat{E}_I(C_{ISB}) + \hat{E}_L(C_{LSB}).$$ (3.10)

Since the full-code INL has been obtained, the full-code DNL can be derived too.

$$D\hat{NL}(C) = I\hat{NL}(C + 1) - I\hat{NL}(C)$$ (3.11)

With this method, the input doesn’t need to hit all the codes. With the hit codes and the ADC segmented architecture, those codes that are not hit can be predicted with the INL segmented models. Therefore, even missing codes can be identified.

3.3 Error Analysis

There are several factors that affect the effectiveness or the performance of the USER-SMILE method. Although the linearity of the input signal is significantly relaxed, there are certain requirements on the input signals. All segments have to be hit. Otherwise, there will be no information on that segment and least square method will fail to solve the equation. Each segment should get sufficient hits to average the noise effect which will be analyzed in the subsection. In addition to the input signal requirement, four most significant error sources are analyzed including the modeling error, the additive noise, the voltage shift nonconstancy and the quantization error. These error sources will affect the estimation accuracy of the algorithm. The USER-SMILE directly models and identifies the INL of the ADC. Therefore, the maximum DNL estimation error can be twice as the maximum INL estimation error.

3.3.1 Effects of Unmodeled Error in the INL Model

In the segmented non-parametric model, the INL curve for the lower bits is assumed to be identical in each MSB segment. However, this may not be true. If the ADC has large nonlinear errors, the lower bits will also be affected. This subsection evaluates the estimation error caused by these unmodeled nonlinear errors.
Define the INL of the ADC as two components: the linear component \( INL_{lk} \) and the nonlinear component \( INL_{nlk} \). The full-code INL can be expressed as

\[
INL_k = INL_{lk} + INL_{nlk}. \tag{3.12}
\]

For the linear component, it is from the capacitor mismatches, which results in a segmented INL shape. For the nonlinear component, it is mainly from the sampling capacitor voltage coefficient, nonlinear parasitics and other voltage dependent effects, which results in a smooth INL shape. The segmented linear errors are modeled by the proposed method. But the nonlinear error is not directly modeled. Define the nonlinear \( f_{nl}(x) \) as a function of the input voltage \( x \) and the unit is LSB. When the input voltage \( x \in [T_{k-1}, T_k) \) and the output code is \( k \), the amount of nonlinearity at code \( k \) is \( INL_{nlk} = f_{nl}(T_{k-1}) \). For the \( j \)-th MSB segment, define the middle point of this segment as \( x_{\text{mid}}(j) \). The Taylor series of the nonlinear function at this segment can be expressed as

\[
f_{nl}(x) = f_{nl}(x_{\text{mid}}(j)) + f'_{nl}(x_{\text{mid}}(j))[x - x_{\text{mid}}(j)] + \sum_{n=2}^{\infty} \frac{f^{(n)}_{nl}(x_{\text{mid}}(j))}{n!}[x - x_{\text{mid}}(j)]^n \approx f_{nl}(x_{\text{mid}}(j)) + f'_{nl}(x_{\text{mid}}(j))[x - x_{\text{mid}}(j)] \tag{3.13}
\]

where the higher order terms are ignored for approximation purpose. In this equation, the first part of the equation is the constant part which will be captured by the segmented INL model. The value of the second part changes as the slope changes, which is not modeled by the proposed method. Within one MSB segment, the difference between the maximum and minimum nonlinear component can be as large as \( f'_{nl}(x_{\text{mid}}(j)) \cdot V_{\text{MSB}} \), where \( V_{\text{MSB}} \) is voltage range of one MSB segment. Therefore, the absolute error is as large as \( f'_{nl}(x_{\text{mid}}(j)) \cdot V_{\text{MSB}}/2 \) with respect to the middle of this segment. With INL end-point fitting, the maximum unmodeled error can be as large as \( f'_{nl}(x_{\text{mid}}(j)) \cdot V_{\text{MSB}} \). Over the entire input range, the absolute value of first derivative of each segment is bounded by the maximum absolute derivative of the \( f_{nl} \) over all \( j \):

\[
\text{abs}[f'_{nl}(x_{\text{mid}}(j))] \leq \max \{ \text{abs}[f'_{nl}(x)] \}. \tag{3.14}
\]

Therefore, the maximum absolute unmodeled error is less than \( \max \{ \text{abs}[f'_{nl}(x)] \} \cdot V_{\text{MSB}} \) with end-point fitting, which can be used to evaluate how much the nonlinear error contributes to the final INL estimation error.
For a 16-bit binary-weighted ADC with 6-5-5 segmentation, there are \(2^6 = 64\) segments for MSB codes and each MSB segment corresponds to the voltage range of \(1/64\) (input range is normalized to 1). Suppose that the nonlinear function \(f_{nl}(x) = 10x^2(x - 1)\), and the maximum absolute \(INL_k\) caused by the nonlinear component is about 1.5 LSB. The maximum slope of this function is around 10. Therefore, the unmodeled error is less than \(\max \{\abs{f'_{nl}(x)}\} \cdot V_{MSB}/2 = 10 \cdot 1/64/2 = 0.078\) LSB with best fitting and 0.156 LSB with end-point fitting. This unmodeled error is very small and can be further reduced by increasing the number of MSB segments.

### 3.3.2 Additive Noise in the Input Signals

To analyze the additive noise effect, rewrite (3.8) into a complete matrix form:

\[
\alpha/V_{LSB} + n_{all} - C_1 + C_2 = H \begin{bmatrix} E_M \\ E_I \\ E_L \end{bmatrix}
\]  

(3.15)

where \(H\) is a \(M \times K\) matrix. \(M\) is the total number of samples in each ramp. \(K\) is the total number of unknown for \(E_M\), \(E_I\) and \(E_L\). In the \(H\) matrix, each row has three locations of \(+1/2\) corresponding to the code \(C_1 + 1\)’s MSB, ISB and LSB codes, three locations of \(+1/2\) corresponding to the code \(C_1\)’s MSB, ISB and LSB codes, three locations of \(-1/2\) corresponding to the code \(C_2 + 1\)’s MSB, ISB and LSB codes, and three locations of \(-1/2\) corresponding to the code \(C_2\)’s MSB, ISB and LSB codes. Then, (3.15) is the matrix expression for (3.8) with \(M\) sets of data. With the least square method, the segmented model parameters are estimated as:

\[
\begin{bmatrix}
\hat{E}_M \\
\hat{E}_I \\
\hat{E}_L 
\end{bmatrix} = (H^T H)^{-1} H^T (\alpha/V_{LSB} - C_1 + C_2).
\]  

(3.16)
In this equation, the noise is effectively averaged. In order to evaluate the exact effect of noise on the final estimation, the noise term \( n_{all} \) needs to be included. Define the estimation error vector to be \( e_M \), \( e_I \) and \( e_L \) for MSB, ISB and LSB segmented INL respectively.

\[
\begin{bmatrix}
\hat{E}_M \\
\hat{E}_I \\
\hat{E}_L
\end{bmatrix} = \begin{bmatrix}
E_M \\
E_I \\
E_L
\end{bmatrix} - \begin{bmatrix}
e_M \\
e_I \\
e_L
\end{bmatrix}
\]

\[
= (H^T H)^{-1} H^T (\alpha/V_{LSB} + n_{all} - C_1 + C_2) - (H^T H)^{-1} H^T n_{all}.
\]  

Therefore, the estimation error caused by the noise is expressed as:

\[
\begin{bmatrix}
e_M \\
e_I \\
e_L
\end{bmatrix} = (H^T H)^{-1} H^T n_{all}.
\]  

(3.18)

To evaluate the relation between the estimation error with the noise variance, some matrix transformations are performed. Multiply both sides by their transpose:

\[
\begin{bmatrix}
e_M \\
e_I \\
e_L
\end{bmatrix} \begin{bmatrix} e_M^T & e_I^T & e_L^T \end{bmatrix} = (H^T H)^{-1} H^T n_{all} n_{all}^T H ((H^T H)^{-1})^T.
\]  

(3.19)

In the left side of (3.19), the diagonal terms are the square of each estimation error.

\[
\begin{bmatrix}
e_M \\
e_I \\
e_L
\end{bmatrix} \begin{bmatrix} e_M^T & e_I^T & e_L^T \end{bmatrix} = \begin{bmatrix}
e_M(0)^2 & \cdots & e_M(0) \cdot e_L(2^{N_L} - 1) \\
\vdots & \ddots & \vdots \\
\end{bmatrix}.
\]  

(3.20)

Similarly, the noise vector multiplied by its transpose is

\[
n_{all}n_{all}^T = \begin{bmatrix}
n_{all}(1)^2 & \cdots & n_{all}(1) \cdot n_{all}(M) \\
\vdots & \ddots & \vdots \\
n_{all}(1) \cdot n_{all}(M) & \cdots & n_{all}(M)^2
\end{bmatrix}.
\]  

(3.21)
There are $M$ sets of noise terms and each noise term has a variance $\sigma_n^2$. Assume the noise is random with 0 mean and each noise term is independent of the other noise terms. The expected value of $n_{all}^T n_{all}^T$ matrix is

$$E[n_{all}n_{all}^T] = \begin{bmatrix}
n_{all}(1)^2 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & n_{all}(M)^2
\end{bmatrix} = \sigma_n^2 I_M. \quad (3.22)$$

where $I_M$ is the $M \times M$ identity matrix. The diagonal terms of $\sigma_n^2 I_M$ are $\sigma_n^2$ while other terms are all zeros.

In the right side of (3.19), $H$ matrix is a constant matrix for a given input signal. The expected values of both sides become:

$$E\left\{ \begin{bmatrix} e_M \\ e_I \\ e_L \end{bmatrix} \begin{bmatrix} e_M^T \\ e_I^T \\ e_L^T \end{bmatrix} \right\} = \sigma_n^2 (H^T H)^{-1} H^T H ((H^T H)^{-1})^T$$

$$= \sigma_n^2 ((H^T H)^{-1})^T$$

$$= \sigma_n^2 (H^T H)^{-1}. \quad (3.23)$$

Therefore, the expected value of the estimation error’s squares are the product of noise variance and the diagonal elements of the $(H^T H)^{-1}$ matrix.

$$E\left\{ \begin{bmatrix} e_M^2(0) \\ e_M^2(1) \\ \vdots \end{bmatrix} \right\} = \sigma_n^2 \cdot \text{diag}((H^T H)^{-1}). \quad (3.24)$$

The $H$ matrix actually depends on the input signal waveform and the voltage shift added to the input, as well as the segmentation in the INL model. To evaluate the sensitivity from the noise to the estimation error, the ADC and the input signal are assumed to be approximately linear for simplicity. Nonlinear ADC or nonlinear input will slightly change the sensitivity but the effect is very small. For a 12-bit ADC with 4-4-4 segmentation (4-bit MSB, 4-bit ISB and 4-bit LSB) and
two 1 hit per code ramps, the maximum value for the diagonal elements in the \((H^T H)^{-1}\) matrix is around 0.02. So, the variance of INL estimation is less than 0.06\(\sigma_n^2\) (worst case is when MSB, ISB and LSB have the same maximum variance). Therefore, the 3 sigma of the estimation error due to noise is less than 0.75\(\sigma_n\). For 16-bit ADC with 6-5-5 segmentation and two 1 hit per code ramps as inputs, the 3 sigma of the INL estimation error due to noise is less than 0.4\(\sigma_n\). For comparison, the histogram ramp test with \(h\) hits per code has a estimation uncertainty variance being \(\sigma_n^2/h\). The 3 sigma of 20 hits per code histogram ramp test is around 0.67\(\sigma_n\). So, the USER-SMILE algorithm produces a similar estimation error due to noise but with 10 times less data.

### 3.3.3 Effects of the Voltage Shift Between Two Signals

The constancy of the voltage shift is critical in the USER-SMILE algorithm. Recent researches have proposed various low-cost highly-constant shift generators (53; 52).

Define the voltage shift as \(\alpha = \bar{\alpha} + \alpha_e\), where \(\bar{\alpha}\) is the mean value of \(\alpha\) and the \(\alpha_e\) is the error part. Similar to (3.15), the estimation error can be obtained with only the error of the voltage shift considered

\[
\alpha/V_{\text{LSB}} - C_1 + C_2 = \bar{\alpha}/V_{\text{LSB}} + \alpha_e/V_{\text{LSB}} - C_1 + C_2
\]

\[
= H \begin{bmatrix} E_M \\ E_I \\ E_L \end{bmatrix} = H \begin{bmatrix} \hat{E}_M \\ \hat{E}_I \\ \hat{E}_L \end{bmatrix} + H \begin{bmatrix} \epsilon_M^{(\alpha)} \\ \epsilon_I^{(\alpha)} \\ \epsilon_L^{(\alpha)} \end{bmatrix},
\]

and

\[
\alpha_e/V_{\text{LSB}} = H \begin{bmatrix} \epsilon_M^{(\alpha)} \\ \epsilon_I^{(\alpha)} \\ \epsilon_L^{(\alpha)} \end{bmatrix}
\]  

(3.26)

where \(\epsilon_M^{(\alpha)}\), \(\epsilon_I^{(\alpha)}\) and \(\epsilon_L^{(\alpha)}\) are the estimation errors for \(E_M\), \(E_I\) and \(E_L\) respectively due to the nonconstancy part of the voltage shift.
As the input signal changes (such as sine wave or ramp), the LSB and ISB segments will change faster and the MSB segments will change slower. If there is an error in the voltage shift within a small input voltage range, it is likely to hit different ISB and LSB segments so that the error is evenly distributed in different segments. Therefore, the shift nonconstancy effect on ISB and LSB is small. To analyze the effect of shift nonconstancy on the MSB segments, the ISB and LSB estimation errors are assumed to 0 and (3.26) can be approximated as:

\[
\frac{\alpha_e}{V_{\text{LSB}}} \approx H_M \begin{bmatrix} e^{(\alpha)}_M \end{bmatrix}
\]

(3.27)

where \( H_M \) is the first \( 2^{N_M} \) columns of the \( H \) matrix. In most cases, output code \( C_1 \) and \( C_1 + 1 \) are in the same MSB segment. \( C_2 \) and \( C_2 + 1 \) are also in the same MSB segment. So, in the \( H_M \) matrix, each row has one “-1” and one “+1” in the \( C_{1\text{MSB}} \) and \( C_{2\text{MSB}} \) locations respectively. For different types of input signals, the MSB segments change differently. For a ramp input, \( C_{1\text{MSB}} \) and \( C_{2\text{MSB}} \) increase from 0 to the maximum. For a sine wave input, \( C_{1\text{MSB}} \) and \( C_{2\text{MSB}} \) go up and down periodically. Regardless of the input signal type, the \( H_M \) matrix can be sorted in ascending order according to the ADC’s output codes. The corresponding \( \alpha_e \) vector will be rearranged according to \( H_M \)’s sorting sequence. An example of \( H_M \) matrix after sorting is shown below.

\[
\begin{bmatrix}
-1 & 1 & 0 & 0 & \cdots & 0 \\
-1 & 0 & 1 & 0 & \cdots & 0 \\
0 & -1 & 1 & 0 & \cdots & 0 \\
0 & -1 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 1
\end{bmatrix}
\]

(3.28)

Each row has one “-1” and one “+1”. The column index of the “+1” location is the value of \( C_{1\text{MSB}} \). Group all rows with the same \( C_{1\text{MSB}} \) and take the average value. For example, the 2nd row and the 3rd row are summed and averaged since the locations of “+1” or the value of \( C_{1\text{MSB}} \) are the same. The average value for corresponding shift error with \( C_{1\text{MSB}} = k \) is defined as \( \bar{\alpha}_e(k) \). Repeat this averaging process for all the MSB codes. The smallest \( C_{1\text{MSB}} \) location is the second column.
since it will be canceled by the “-1” in the first column if \( C_{1\text{MSB}} \) is also in the first column. Since all the error differences are relative, \( e_M(0) \) can be defined to be 0. Therefore, the first column of the new matrix is removed. If \( e_M(0) \) is not 0, all the errors will add a constant value, which doesn’t change the INL. Then, a new lower triangle matrix can be formed with all ones in the diagonal.

\[
\begin{pmatrix}
1 & 0 & 0 & \cdots & 0 \\
-l_{2,1} & 1 & 0 & \cdots & 0 \\
-l_{3,1} & -l_{3,2} & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-l_{k,1} & -l_{k,2} & -l_{k,3} & \cdots & 0 \\
\vdots & \vdots & \vdots & \cdots & \vdots \\
\end{pmatrix}
\begin{pmatrix}
e_M(1) \\
e_M(2) \\
\vdots \\
e_M(k) \\
\end{pmatrix}
\approx
\begin{pmatrix}
\bar{\alpha}_e(1) \\
\bar{\alpha}_e(2) \\
\vdots \\
\bar{\alpha}_e(k) \\
\end{pmatrix}
/ V_{\text{LSB}}.
\]

(3.29)

All the elements above the diagonal are zeros. And for all the elements below the diagonal, they satisfy the following conditions

\[ 0 \leq l_{k,i} \leq 1 \]  
for all \( i < k \) and \( 1 < k < 2^N_M \).

\[ 0 \leq \sum_{i=1}^{k-1} l_{k,i} \leq 1 \]  
for all \( 1 < k < 2^N_M \).

For a lower triangle matrix, forward substitution can be used to solve for all the unknowns.

\[ e_M(1) \approx \bar{\alpha}_e(1), \]
\[ e_M(2) \approx \bar{\alpha}_e(2) + l_{2,1} e_M(1), \]
\[ \vdots \]
\[ e_M(k) \approx \bar{\alpha}_e(k) + \sum_{i=1}^{k-1} [l_{k,i} e_M(i)]. \]

(3.32)

The maximum absolute value for \( \alpha_e \) is \( \max\{|\alpha_e|\} \). Taking the absolute value for both sides, the inequation can be obtained:

\[ |e_M(k)| \leq \max\{|\alpha_e|\} + \sum_{i=1}^{k-1} |l_{k,i}| |e_M(i)|, \]

(3.33)
with the initial condition

\[ |e_M(1)| \leq \max\{|\alpha_e|\}. \]  

(3.34)

Therefore, the bound for the \( k \)-th MSB’s estimation error is

\[ |e_M(k)| \leq k \cdot \max\{|\alpha_e|\}. \]  

(3.35)

Define that the constancy of the voltage shift as the ratio of the voltage shift error over the average shift value. Take a 12-bit ADC as an example. With 4-bit as the MSB segment and the voltage shift is around 1 MSB, the maximum error is less than \( 16 \times \max\{|\alpha_e|\} \). To achieve 0.2 LSB error from the USER-SMILE algorithm, the voltage shift error should be less than 0.0125 LSB in 12-bit level, which is 50ppm for the shift constancy. For a 16-bit ADC with 6-bit as the MSB segment, the constancy requirement is 3 ppm. The previous work has demonstrated that voltage shift constancy can achieve below 1 ppm (52).

### 3.3.4 Effects of Quantization Error

The USER-SMILE algorithm enables less data than histogram test. In all previous assumptions and derivations, we assume that the total noise is random with 0 mean. However, with 1 hit per code or less than 1 hit per code input signal, the quantization error is not random. With more hits per code like histogram, the average quantization error is less.

If the standard deviation of the additive noise is comparable to 1 LSB or fraction of LSB, the effect of the quantization error will be similar to random noise, which means it is effectively whitened. If the test environment is ultra low-noise and the ADC is also designed to have a very low noise, additional dithering is needed to whiten the quantization error (57; 58).

### 3.4 Simulation Results

In this section, extensive simulations have been done to verify the effectiveness and accuracy of the proposed algorithm. The estimation errors are compared with the analysis in section 3.3. The SAR ADC is particularly modeled due to its wide usage, high resolution and low power
features. The behavior model simulation has the advantage of statistical analysis as well as the control over ADC performance. Since everything is mathematical model, the ADC INL/DNL can be theoretically derived. In all the following simulations, 16-bit ADC with 6-5-5 segmentation is modeled unless specified. The input signals are two 1 hit/code nonlinear ramp with around 8-bit linearity performance.

In the SAR ADC, multiple non-idealities are modeled including capacitor random mismatches, voltage dependent coefficient in switch and capacitors, input-referred noise and voltage shift non-constancy. When analyzing one error source, the other error sources will be removed or minimized. In addition, the capability of identifying missing codes is explained at the end of this section.

### 3.4.1 Modeling Error

![INL Modeling Error](image)

In the USER-SMILE algorithm, it is assumed that the ISB or LSB error terms are identical across the entire full codes for the same ISB or LSB code. However, this may not be the case if the ADC nonlinearity has voltage dependency such as capacitor voltage coefficient or voltage
dependent parasitics. A high order polynomial function models the voltage dependency effect. In this test, the noise is set to 0 and the voltage shift is ideal.

In the previous section, the function of \( f_{nl}(x) = 10x^2(x - 1) \) is used. In order to exaggerate the modeling error effect, a 5 time larger nonlinear function (more than 7.5 LSB) is used to represent the smooth nonlinearity of the ADC. Figure 3.4 shows the INL comparison, the introduced nonlinear error and the estimation error. Based on the analysis, the maximum unmodeled error from USER-SMILE is 0.78 LSB with end-point fitting. In the simulation result, a boundary of +/- 0.78 LSB is shown and the estimation error is within the error boundary.

### 3.4.2 Noise Effect

![INL Estimation Comparison](figure3.5a)

![Estimation Error](figure3.5b)

Figure 3.5 16-bit ADC INL Estimation (1 LSB noise).

Random noise has a direct impact on the USER-SMILE estimation accuracy. In the previous analysis, the INL estimation error has a 3-sigma of \( 0.4\sigma_n \). For comparison, the ADC is also tested with a 20 hits/code linear ramp by histogram test. The 3 sigma estimation error for histogram is \( 0.67\sigma_n \). Random noise with 1 LSB sigma (\( \sigma_n \)) is added to the input. The voltage dependency and voltage shift constancy are set to 0. Both USER-SMILE and histogram INL estimations are shown in Figure 3.5(a) together with the theoretical true INL. Both histogram test and USER-SMILE align well with the theoretical INL. But the USER-SMILE result shows better noise averaging.
The INL estimation errors for this test case are plotted in Figure 3.5(b). The USER-SMILE INL estimation errors are all within the 3-sigma estimation boundary (+/- 0.4\(\sigma_n\)). For the histogram test, there are a few codes beyond the 3-sigma boundary (+/- 0.67\(\sigma_n\)), which is reasonable considering around 65k codes for a 16-bit ADC.

![Figure 3.6 1000-time 16-bit ADC INL Estimation Error(1 LSB noise)](image)

1000 SAR ADCs are randomly generated to further verify the noise effect. For each ADC, the worst INL estimation across all 65,536 codes is selected. Therefore, 1000 worst INL estimation errors (absolute value) can be plotted in Figure 3.6 for both USER-SMILE and histogram test. In this plot, the 6-sigma lines are drawn (0.8\(\sigma_n\) for USER-SMILE, 1.34\(\sigma_n\) for histogram). All the 1000 simulations are below the 6-sigma lines for both USER-SMILE and histogram.

### 3.4.3 Voltage Shift Effect

The constancy of the voltage shift is critical in the USER-SMILE algorithm. As analyzed in the earlier section, to achieve 0.2 LSB estimation error, the shift constancy needs to be less than 3ppm for a 16-bit ADC. In this example, we generate a 3ppm step function as the nonconstancy of the voltage shift while other error sources are all set to be 0. This voltage shift error results in a
“bell”-shaped estimation error as shown in Figure 3.7. From the simulation, the estimation errors are all within the $+/-.2$ LSB error limits.

![Figure 3.7 INL Estimation Error with 3ppm nonconstancy](image)

### 3.4.4 Missing Codes Identifications

The USER-SMILE algorithm can estimate the full-code INL with even less than 1 hit/code input signal, which means some codes may not be hit in the test. Thanks to the segmented INL model, even some codes are not hit, their linearity can still be accurately predicted from other codes. Missing codes in the ADC is usually a very important specification. Some special cases are generated to illustrate the missing codes identification in the USER-SMILE. In this example, 12-bit ADCs with 4-4-4 segmentation are used for simplicity. For SAR ADCs, define the most significant capacitor as $C_1$ and the least significant capacitor as $C_{12}$. 
In the first case, suppose the 4-th MSB capacitor \((C_4)\) is smaller than expected, causing missing codes at major transitions (Figure 3.8(a)). Therefore, code 255 will be missing. As a result, all the transition voltages for the second MSB segment (from code 256 to code 511) will be lower, causing the segmented error of this MSB segment \((e_M(1))\) to be lower. From Figure 3.8(a), the INL from code 256 to 511 is around 1 LSB lower than the INL from code 0 to 255. The missing codes occur every time when \(C_4\) is selected (every 512 codes). Therefore, the next missing code is 767 (255+512). For the USER-SMILE algorithm, although code 255 is never hit, the INL(255) can be estimated by:

\[
\hat{INL}(255) = E_M(0) + E_I(15) + E_L(15) \quad (3.36)
\]

where \(E_M(0)\), \(E_I(15)\) and \(E_L(15)\) can be estimated from other codes. When constructing the INL, if the beginning of the MSB segment is more than 1 LSB lower than the end of the previous MSB segment, a missing code is identified. The DNL for that code will be set to -1 and the INL will be updated accordingly.

In the second case, the missing codes occur in the middle of a MSB segment rather than the MSB major transitions. In Figure 3.8(b), the 5-th capacitor \((C_5)\) is smaller than normal and causes a missing code at code 127. It will be identified in the ISB segmented errors. This missing code
will repeat in every MSB segment regardless of the MSB errors. The next appearance will be 383 (127+256).

3.5 Discussion

The segmented INL model can accurately represent the INL for ADCs with segmented architecture. However, for low-resolution high-speed flash ADCs, the INL is not segmented. Therefore, the segmented INL model cannot be used to estimate the linearity for flash ADCs. The INL for delta sigma ADCs is also not segmented. But high resolution delta sigma ADCs are never tested for full-code INL/DNL (35).

In this chapter, the segmented model is targeted for binary-weighted ADCs. For other segmented ADCs such as Pipeline ADCs, Cyclic ADCs and subradix-2 SAR ADCs, some modifications are needed to the segmented model. For these ADCs, the segmentation of the INL is not determined by the final output codes due to calibration or redundancy. Instead, the raw code directly from the comparators or shift registers should be used to determine the segmentation. In (3.8), $C_1$ and $C_2$ are the final codes but the MSB/ISB/LSB codes should be obtained from the raw code.

The proposed method can be implemented as on-chip BIST in a low-cost way. The linearity requirement for the signal generator has been significantly relaxed. Therefore, the design of such signal generators is simplified. DACs such as R2R DACs or simple ramp generators can be chosen as the signal generators (54). For embedded ADCs, there is usually a DAC in the same SoC, which can be reused as the signal generator. The key of the implementation is the voltage shift generation. It has been shown in many literature that generation of highly constant voltage shift can be achieved on-chip(51; 52; 53; 54).

3.6 Conclusion

A fast and cost-effective method for ADC linearity test is presented in this chapter. The USER-SMILE algorithm allows the stimulus signal’s linearity requirement to be significantly relaxed and the test time to be reduced by orders of magnitude compared to the state-of-art histogram method,
thus greatly reducing the test cost. The simulation results demonstrate that the USER-SMILE can achieve superior test coverage and accuracy. With the USER-SMILE algorithm, a new BIST solution can be practical, which doesn’t require highly accurate and expensive ATE as the signal generator. Furthermore, it saves the test time and simplifies the test board and interface design.

To actually implement this method as on-chip hardware, the signal generator design is simplified. The voltage shift generator can be implemented in a cost-effective way. The algorithm part can be processed in the ATE for production testing. Or it can be implemented as software in the CPU of a system-on-chip (SoC) or designed as a hardware block which doesn’t consume CPU resource. The extra cost introduced is well compensated by the test cost reduction.

With on-chip signal generator and on-chip computation in SoC, it not only saves the test cost, but also enables the field testing. For example, the BIST can be performed every time the chip is powered on. For critical applications such as automotive, this self-test feature is very important to guarantee the functionality and performance of the electronic system over the product’s life time. In addition, the self-test results can be further used for calibration purposes which can repair and improve the ADC performance.
A novel ultrafast and low-cost pipelined ADC testing and calibration method is proposed. The ADC nonlinearities are modeled as segmented parameters with inter-stage gain errors. During the test phase, a pure sine wave is sent as input and the model parameters are estimated from the output data with the system identification method. Significantly fewer samples are required when compared to traditional histogram testing. The modeled errors are then removed from the digital output codes during the calibration phase. Extensive simulations have been run to verify the correctness and robustness of the proposed method. With just 4000 samples, a 12-bit ADC can be accurately tested and calibrated to achieve less than 1 LSB INL. Measurement results show that the ADC ENOB is improved from 9.7 bits to 10.84 bits and the SFDR is improved by 20dB after calibration. The chip is fabricated in 40nm technology and consumes 10.71mW at a sampling rate of 125MS/s.

4.1 Introduction

The analog-to-digital converter (ADC) is widely used in modern electronic systems. There are different types of ADCs for different applications. The successive approximation register (SAR) ADC is mainly used for data acquisition due to its high resolution and excellent power efficiency. The flash ADC is used for high speed sampling, radar and flash memory. The pipelined ADC has both high resolution and high speed characteristics, being widely used in high speed instrumentation, video, radio and many communication systems. In recent studies, the multi-bit per stage pipelined ADC with open-loop amplifier is becoming popular because it has better power efficiency than conventional pipelined ADCs. However, the linearity errors in the sub-ADCs and the open-loop amplifier gain error usually significantly affect the overall ADC performance. The ADC has to
be tested and calibrated to achieve excellent power efficiency while achieving high resolution and high speed at the same time.

Products used in mission critical areas such as automotive are tested with even more stringent requirements across different temperatures and different voltages supplies. However, testing the ADCs is very challenging for various reasons (3; 4; 5). There are different specifications of ADCs that need to be tested, mainly divided into two categories: static testing and dynamic testing. For static linearity testing, the ADC is tested for integral nonlinearity (INL), differential nonlinearity (DNL), offset and gain error. The ADC is tested with a linear ramp or sine wave to obtain the INL and DNL (2; 40). For dynamic linearity testing, the ADC is tested with a pure sine wave at a certain frequency to obtain the signal to noise ratio (SNR), total harmonic distortion (THD), spurious free dynamic range (SFDR) etc. The challenges of ADC testing are mainly from three requirements: data acquisition time, linear stimulus generation and precision clocking (22; 35). The most dominant cost of testing is from the INL/DNL tests. Various methods have been proposed to reduce the test cost. In (22), the proposed stimulus error identification and removal (SEIR) method relaxes the input stimulus linearity requirement and is able to test the ADC with nonlinear signals. This method can be used in all types of ADCs. But the test time is still very long. In (28), a pipelined ADC testing method is proposed to reduce the test time. It models the pipelined ADC linearity errors as 1-bit per stage. However, for multi-bit per stage pipelined ADCs, the linearity errors in each stage may not be a linear combination of each bit error. Especially for resistor-string DACs, the linearity error is determined by the full-code in each stage, rather than single bit value. Therefore, it will not work effectively in multi-bit per stage pipelined ADCs. In (35), an ultrafast ADC testing method is developed with significantly reduced test time. It models the ADC INL with non-parametric segmented parameters, which works very well in segmented high resolution ADCs such as SAR ADCs. But it cannot be directly applied to test the redundant pipelined ADCs. In (36; 37), the authors developed an on-chip built-in self-test (BIST) method with a reduced test time and relaxed stimulus linearity requirement. Similar to (35), it cannot be directly applied to the redundant pipelined ADCs.
After the testing of the ADC, bad parts are usually screened out, which results in a yield loss. However, the very accurate testing results are wasted. With minimum modifications to the ADC, calibration can be implemented to improve the performance and meet the test limits. There are many calibration methods for ADCs, especially for pipelined ADCs. Calibrations methods broadly fall under two categories: analog method and digital method. Process scaling and synthesizability make digital calibration favorable in digital processes. In (59), the authors calibrate the residue range and comparator offset. However, the linearity of the DAC is not calibrated, which could cause large errors in high resolution ADCs. The pseudorandom noise (PN) injection method is developed and applied in (60; 61; 62). In the PN injection method, capacitor mismatches are calibrated, while some other errors are not calibrated such as smooth nonlinearity due to capacitor voltage coefficients. In addition, the PN method is architecture dependent and it usually takes too many samples to achieve a decent estimation, which results in a long test and calibration time.

The dynamic element matching (DEM) technique in (63) reduces the digital-to-analog converter (DAC) nonlinearity with dynamic switching. The DEM is widely used in sigma delta ADCs and many temperature sensor units. The order element matching (OEM) technique measures the unit element value and groups them to achieve better matching (64), which is suitable for segmented DACs or ADCs. The least mean square (LMS) algorithm is used in (65; 66). It achieves background calibration with a slow but accurate ADC so that the fast ADC can approach the linearity of the linear ADC. However, the slow ADC still needs to be tested and calibrated to guarantee the performance. The method in (67) exploits the redundant ADC INL characteristics and takes an INL-based calibration scheme for pipelined ADC or other redundant ADCs. However, the number of samples needed to achieve good calibration results is very large, which results in a long test and calibration time.

Among all these prior arts, there is no method that can test and calibrate multi-bit per stage pipelined ADCs within a significantly reduced test time. In this chapter, a novel low-cost method for error identification and calibration of multi-bit/stage pipelined ADCs is proposed. It takes an accurate sine wave signal as the input. With the system identification method, the modeled ADC
errors are obtained. This method allows 10-time less samples than the conventional histogram test. The estimated errors are then used in the digital post-processing to do calibration. The proposed method is applied to a 12-bit pipeline ADC. The measurement results show that the ADC INL is reduced to less than 1 LSB. The effective number of bits (ENOB) based on signal-to-noise-and-distortion ratio (SNDR) is improved from 9.7 bits to 10.84 bits and the SFDR is improved by more than 20dB after calibration. It not only tests and calibrates the ADC, but also changes the way the ADC is designed. With the testing and calibration method, the design requirements for the DAC and the residue amplifier is relaxed. The DAC can use less area and the amplifier can consume less power. Therefore, the ADC area and overall power efficiency is significantly improved. The contribution of this chapter is to adopt the system identification approach in (35) and introduce a redundant ADC model so that a pipelined ADC can be accurately tested and calibrated with a significantly reduced number of samples. This model reveals the errors of the ADC internal architecture. It enables the calibration which cannot be achieved in conventional histogram test. The entire test and calibration flow is described, analyzed, simulated and validated in the experimental measurement.

This chapter is organized as follows. Section 4.2 reviews the architecture of the ADC tested and calibrated by the propose method. Section 4.3 describes the modeling of the pipeline ADC errors. Section 4.4 introduces the proposed testing and calibration method. The simulation results are shown in section 4.5 and the measurement results are shown in section 4.6. The conclusion is drawn in section 4.7.

4.2 Pipelined ADC Architecture Review

This section reviews the design for a high-speed multi-bit per stage pipelined ADC where the proposed method is implemented. The proposed method is applied to this specific ADC as a proof of concept. Both the simulation and the measurement results are based on this ADC architecture. It can also be applied to other types of pipelined ADCs for testing and calibration. This ADC has 3 stages and each stage has 5 bits, as shown in Figure 4.1, and the sub-ADC including the residue
amplifier is shown in Figure 4.2. Each stage has a 5 bit ADC, which is a 3 bit/cycle redundant SAR ADC to improve the speed, relax the comparator offset and reduce power consumption at the same time. The first two sub-ADCs are followed by a residue amplifier as shown in Figure 4.2. The last stage is just a 5 bit SAR ADC, same as the ADC used in the first two stages.

![Figure 4.1 Pipelined ADC architecture](image1)

![Figure 4.2 Sub-ADC architecture for stage 1 and stage 2](image2)

A conventional $N_S$-bit flash ADC, if used in the sub-ADC requires $2^{N_S} - 1$ comparators, consuming a large area and large power. A conventional SAR ADC uses a single comparator but takes $N_S$ clock cycles to finish the conversion. Therefore, the SAR ADC has good power efficiency but the entire ADC speed is slowed down. This ADC combines the advantages of speed of the flash ADC and the power efficiency of the SAR ADC. With 6 comparators, the sub-ADC can resolve 3 bits in one clock cycle. To further improve the speed and relax the settling error, one redundant bit is added between two conversion cycles. Therefore, after two conversion cycles, 5 bit results can be obtained in the sub-ADC. Benefiting from the redundancy in the pipelined ADC, the comparator offset and settling errors are significantly relaxed. The needed linearity for the sub-ADC is only at
the 4-bit level. For 3 stages, there are two redundant bits between stages, and the last bit of the last stage is not used. Therefore, the overall resolution of the 3-stage ADC is 12 bits.

In the sub-ADC architecture, the 5-bit DAC is made from a resistor string. The resistor string has relatively good matching with small area and the settling speed is fast. The same resistor string is also used for the comparator reference in the SAR ADC conversion phase. The residue amplifier usually suffers from large power consumption in conventional pipelined ADC design due to the high gain and high bandwidth requirement for a closed-loop amplifier. To overcome the large power consumption issue, open-loop amplifiers have been designed to replace the closed-loop amplifier in recent research. In this ADC, the open-loop amplifier is designed. However, open-loop amplifiers have large gain errors which need to be calibrated.

### 4.3 Pipelined ADC Nonlinearity Modeling

#### 4.3.1 Nonlinearity Modeling

In the conventional histogram test, either a pure sine wave or a linear ramp input signal is used. Around 20 or more samples per ADC code are used to average out the measurement and internal noise in the ADC. The sine wave histogram test requires $\pi/2$ times the total number of samples required by the histogram ramp test (35). For high resolution ADCs, the total number of samples for histogram is significant and it increases exponentially with the resolution of the ADC. The corresponding test time and test cost is extremely high. For a 16-bit ADC with 32 samples per code, the total number of samples will be 2 million. However, the total number of components is usually small for high resolution ADC architectures such as SAR, cyclic and pipelined ADCs. For pipelined ADC, the same sub-ADC is repeatedly used for different first stage codes. The same error in the sub-ADC will repeat in the final INL.

Instead of estimating the linearity error for each code, the proposed method uses a non-parametric segmented model for the pipelined ADC. The pipelined ADC has multiple stages and residue amplifiers between stages. The DAC nonlinearity in each stage and the inter-stage gain errors are usually the dominant error sources. The proposed method takes a black-box approach,
and models the INL contribution from the nonlinearity in each stage and the residue amplifier gain error. The exact error sources are not important but only their impacts on the INL matter. In each stage, the error is modeled as $2^{N_S}$ error terms for $N_S$ bits in each stage. If the first stage has 5 bits, there will be $2^5 = 32$ error terms. It is like treating each stage as a flash ADC even though the actual architecture is a binary-weighted one. Such a segmented model turns out to be very accurate. Even for many segmented ADCs, such as a capacitive SAR ADC, the INL cannot be simply modeled as $N_S$ terms since capacitor mismatch is usually not the only error source. Instead of modeling the capacitor mismatches, the proposed model is estimating the final INL from all the error sources.

The last stage of a pipelined ADC can easily achieve its linearity requirement. Normally, the last stage can be assumed to be linear. However, if the ADC is indeed very nonlinear, the last stage can still be included in the model. For the inter-stage gain error, it is modeled as a constant gain error. It is possible that the residue amplifier has both constant gain error and nonlinearity. Such nonlinearity in the residue amplifier can be treated as part of the next stage’s nonlinearity model. Then, the full INL model equation for a 3-stage pipelined ADC described in section 4.2 is:

$$INL_{D_1,D_2,D_3} = E_1(D_1) + E_2(D_2) + E_3(D_3) + G_1 \cdot D_2 + G_2 \cdot D_3,$$

(4.1)

where $INL_{D_1,D_2,D_3}$ is the modeled INL for code $D_1$, $D_2$ and $D_3$ in the three stages; $E_1$, $E_2$ and $E_3$ are the modeled nonlinearity in each stage; $G_1$ and $G_2$ are the modeled first and second residue amplifier gain errors. If the last stage and the second amplifier nonlinearity is negligible, the model can be simplified as:

$$INL_{D_1,D_2,D_3} = E_1(D_1) + E_2(D_2) + G_1 \cdot D_2 + G_2 \cdot D_3,$$

(4.2)

where the $E_3$ term is dropped.

### 4.3.2 INL Construction

Due to the redundancy characteristics of the pipelined ADCs, the final code is not a one-to-one mapping from the raw codes in each stage. The same final code may correspond to two different
raw codes combinations. A simple 12-bit pipelined ADC with 3 bits in the first stage and 10 bits in the second stage is used as an example to explain the relation and how the INL is affected.

Figure 4.3 shows the residue voltage transfer function from the first stage. The input is normalized with respect to the $V_{\text{ref}}$. Ideally, the sub-ADC in the first stage has the transition at $-5/8, -3/8, ..., 5/8$ of $V_{\text{ref}}$ to properly place the residue voltage in the middle of the next stage’s input range. Therefore, it can tolerate the error from settling and comparator offset in the first stage sub-ADC. If there is no error from the sub-ADC, the transfer function will be the black lines in Figure 4.3. However, with the sub-ADC, it can still tolerate a range shown in the grey line so that the residue voltage is still within $+/-1$ LSB of the first stage.

![Residue Voltage Transfer Function in Ideal Case](image)

Figure 4.3 1st stage residue voltage transfer function in ideal case

When the sub-ADC has errors, the residue voltage transfer function will be as shown in Figure 4.4. The transition levels for the sub-ADC codes change the locations. And the width of each black line also changes. If there is conversion noise and insufficient settling during the sub-ADC evaluation phase, then the transition level will not always be the same. Suppose the first vertical dashed line in Figure 4.4 is the noise-free and settling error free case transition level. When there is a large positive noise or a positive settling error, the transition level will move to the right and vice versa. Therefore, in the actual sub-ADC operation, the transition level is a region as shown in the shadowed area, not a fixed boundary. And the black lines for code 0 and code 1 have an extended length with an overlapped range in this region. And this is the same for all other transitions. The
Figure 4.4 1st stage residue voltage transfer function with transient errors

Figure 4.5 Comparison between redundant INL and the histogram-based INL

This overlapped region will have an impact on the final INL results which are different from normal non-redundant ADCs. For the same final output code, if the sub-ADC in the first stage output is 0, the DAC code in the first stage will be 0 too. The error contribution from the first stage is the error of the DAC with code equal to 0. Similarly, if the sub-ADC in the first stage output is 1, the error contribution from the first stage is the error of the DAC with code equal to
However, the normal INL will only have a single value at the same final output code. With conventional histogram test, the final INL is obtained with multiple hits per code. It is possible that the same output codes are from different DAC codes in the first stage. The final INL for this code will have a value in the middle of the errors of these two different DAC codes. The blue line in Figure 4.5 shows an example of histogram-based INL test result. Between code 512 and code 1023, the same output codes could come from either DAC code 0 or DAC code 1 in the first stage. The INL in this region has a relatively “smooth” transition rather than a sharp line.

For modeling, it is therefore not possible to accurately find the sub-ADC transition boundaries since the noise and settling error are not predictable. However, if the boundaries are assumed to be anywhere where the residue is in the valid input range, the INL can still be obtained for any boundary. To accurately represent the INL, an extended version of INL is shown as the black lines in Figure 4.5. The same output code corresponds to two different INL values. For a three stage pipelined ADC, there is another level of redundancy between the second and third stage. So, the same output code could have 4 different possible sub-ADC code combinations. Using the model parameters described in the previous subsection, the INL for code $D$ can be represented as:

$$INL_{D_1^{(1)},D_2^{(1)},D_3^{(1)}} = E_1(D_1^{(1)}) + E_2(D_2^{(1)}) + G_1 \cdot D_2^{(1)} + G_2 \cdot D_3^{(1)},$$

(4.3)

if the sub-ADC codes are $D_1^{(1)}, D_2^{(1)}, D_3^{(1)}$. For the same output code, it could have another sub-ADC code combination $D_1^{(2)}, D_2^{(2)}, D_3^{(2)}$. Then, the INL for this combination will be:

$$INL_{D_1^{(2)},D_2^{(2)},D_3^{(2)}} = E_1(D_1^{(2)}) + E_2(D_2^{(2)}) + G_1 \cdot D_2^{(2)} + G_2 \cdot D_3^{(2)}.$$

(4.4)

They have the same final output code but different INL values. In production testing, the worst case INL should be used since it is possible that the user might achieve that INL value in certain conditions.

It is also possible that the sub-ADC in the first stage is really bad or the noise is significant enough that the transition boundary is beyond the tolerable range. Then, the residue voltage generated from the first stage will saturate in the following stage and the output code will not be
valid. And this ADC will be uncalibratable unless the sub-ADC has extra calibration or trimming capability, which is beyond the scope of this chapter.

The relation between the INL and DNL for code $D$ is normally defined as:

$$DNL(D) = INL(D + 1) - INL(D).$$  \hfill (4.5)$$

Unlike the normal INL/DNL, the DNL in the extended INL model is not well defined since the same output code corresponds to multiple INL values. As the two adjacent codes have many different INL combinations and all combinations can happen, the worst case combination can be used as the DNL value.

### 4.4 ADC Testing and Calibration

Suppose there is an ideal linear ADC with the same end point fit line as the ADC under test. For an input signal $V_{in}$, define the actual ADC output code as $D$. For the same input signal $V_{in}$, the output from the ideal ADC is defined as $D_{exp}$. The input/output relationship can be expressed as:

$$D_{exp} - D + noise = E_1(D_1) + E_2(D_2) + G_1 \cdot D_2 + G_2 \cdot D_3,$$

(4.6)

where $noise$ is the total noise sources including thermal noise and quantization noise. As long as the expected code $D_{exp}$ is known, any identification method can be used for estimation. This can be done with a similar procedure as the uSMILE algorithm (35). The implementation is shown in Figure 4.6. If the input signal is a pure sine wave, the ideal ADC and actual ADC will both have the same offset and fundamental. Let the ideal ADC have no quantization noise and no input referred noise. Then, the expected codes for the linear ADC can be obtained by constructing the time domain codes from only the DC and fundamental components in the actual ADC’s FFT results (35). Similarly, for a linear ramp input, a linear best fit line can be used to obtain the expected codes for the linear ADC. The sine wave testing needs to be done with coherent sampling to obtain the DC and fundamental accurately.
Figure 4.6  Block diagram of the proposed algorithm

Define the error for each sample as $D_{err} = D_{exp} - D$. For $k$-th input voltage, one such equation can be obtained:

$$D_{err}^{(k)} + \text{noise}^{(k)} = E_1(D_1^{(k)}) + E_2(D_2^{(k)}) + G_1 \cdot D_2^{(k)} + G_2 \cdot D_3^{(k)}.$$  \hspace{1cm} (4.7)

The values of $D_{err}$ are obtained by doing inverse FFT after removing the DC and fundamental of the FFT of the ADC output codes. With M samples, M such equations can be formed. The least square method can be used to identify the unknown parameters, i.e., $E_1$, $E_2$, $G_1$ and $G_2$. The noise term will be effectively averaged out in the least square method. To estimate all model parameters, all the codes in each stage need to appear at least once in the ADC output in the identification. In this ADC architecture, there are 66 unknown parameters (32 errors in the first stage, 32 errors in the second stage and 2 residue amplifier gain errors). Theoretically, 66 samples are enough to estimate the unknowns. In the actual implementation, usually more samples are needed to accurately obtain the FFT DC and fundamental, and effectively average out the effect of noise. Then, with known $D_{exp}$ and $D$, the unknown model parameters can be obtained. Once all the model parameters are identified, the redundant INL can be constructed.

These parameters can be further used for calibration to improve the linearity of the ADC. Digital calibration has the advantage of small area, being process-scaling friendly, speed and integration. Therefore, digital calibration is implemented to calibrate this ADC. From equation (4.6),
the expected output code will be:

\[ D_{\text{exp}} = D + E_1(D_1) + E_2(D_2) + G_1 \cdot D_2 + G_2 \cdot D_3 - \text{noise}. \]  

(4.8)

Since the noise term is unpredictable and random, the code after calibration is:

\[ D_{\text{cali}} = D + \tilde{E}_1(D_1) + \tilde{E}_2(D_2) + \tilde{G}_1 \cdot D_2 + \tilde{G}_2 \cdot D_3, \]  

(4.9)

where \( D_{\text{cali}} \) is the calibrated output code. The error in the calibrated output code will only have the noise (quantization noise and random noise), model parameters estimation error, the unmodeled errors and additional rounding error if the calibrated output code is rounded to the same resolution.

The entire testing and calibration flow is summarized in the flowchart in Figure 4.7 (35). The proposed method can be used just for testing to obtain the worst case nonlinearity. Or it can also be used to calibrate the ADC to improve the nonlinearity without computing the INL/DNL. In addition, this method can be used to estimate the INL/DNL again after calibration.

The total number of error terms is only 66 in this example and roughly 100 to 200 for different segmentation and resolutions. If we define the number of error terms as \( K \), the test results uncertainly will have a variance of \( \pi \cdot K \cdot \sigma^2/(2M) \), where \( M \) is the total number of measurements and
\( \sigma^2 \) is the total noise variance in (4.6). For sine wave histogram test, the test results uncertainty variance is \( \pi \cdot \sigma^2 / (2H) \), where \( H \) is the number of hits per code (35). With the same test result uncertainty, we can obtain \( M = H \cdot K \). Therefore, for an N-bit ADC, the histogram requires \( H \cdot 2^N \) samples to achieve the same test uncertainty as the proposed method with only \( H \cdot K \) samples. The proposed method can achieve a time saving factor of \( 2^N / K \), which is around 60 for this ADC (68).

For this specific ADC, there are 3 stages and each stage has 5 bits. Therefore, for each ADC conversion, the number of bits to be saved is 15. With 4096 samples, about 8k bytes memory is sufficient. Only the segmented INL model parameters need to be saved in the memory for calibration. The total number of the model parameters is \( 2^5 + 2^5 + 1 + 1 = 66 \). If each model parameter takes 1 byte memory (4-bit for integer part and 4-bit for fractional part), the size for calibration data is only 66 bytes. The intermediate computation also involves FFT and least square, which will require temporary memory spaces. The FFT function can use 4k bytes to calculate and store the final results. For the least square method, it requires a 66 \( \times \) 66 memory location and a 1 \( \times \) 66 memory location (38). With 2 bytes for each memory location in the least square, the memory for least square is around 8k bytes. Therefore, the total memory needed is roughly 20k bytes. This is the most relaxed requirement for the memory. Certain optimization and reuse can be achieved to reduce the memory requirement.

Compared to the histogram method, the proposed method takes advantage of the most accurate information of the pure sine wave input. In the histogram method, only the bin counts are used for DNL/INL calculation, which disregards the most accurate input information. And the number of hits in one bin is only used for estimating the DNL for this specific code, and provides no useful information for other codes DNL estimation (35). In the proposed method, the ADC output codes are used to estimate the input sine wave information. By removing the DC and fundamental in the FFT results, only the nonlinearity of the ADC, the noise and quantization error left. And for each error term in (4.3), multiple ADC output codes are used to estimate this term. For example, for the MSB error term \( E_1(0) \), there are roughly 100 ADC output codes with first stage code being
0 in a 4096-sample sine wave test. For the histogram test, there is less than 1 hit per code in a 4096-sample sine wave test.

This method can be used for virtually all ADCs with redundancy such as 1.5 bit per stage pipelined ADC, cyclic ADC, redundant SAR ADC and so on. The model needs to be adjusted to represent the corresponding segmentation. It needs to know the ADC segmentation and have access to the internal codes for each stage. Since the number of samples are significantly reduced, the quantization error will be significant if the environmental and thermal noise is very low. In this case, proper dithering is necessary to whiten the quantization error (58; 69). In this model, it is assumed that the following stages’ nonlinearity remain the same regardless of the previous stages’ codes so that the superposition can be used. However, if there is any dependency of the nonlinearity in the following stages on the previous stage’s codes, this model will not be accurate. In this case, a more complicated model needs to be developed.

4.5 Simulation Results

To demonstrate the correctness and effectiveness of the proposed method, both simulation and measurement results are presented. The simulation has some advantages over the measurement results. First, the true INL cannot be obtained in the measurement, while the simulation can accurately provide the INL. Second, the performance of the ADC is under control in the simulation, where ADCs with various performances can be compared under the proposed method. Third, a large number of ADCs can be simulated for statistical analysis, which is not possible for a test chip ADC. In this section, the simulation results are presented, and the measurement results will be shown in the next section.

A 12 bit ADC with the same architecture described in section 4.2 is modeled in Matlab. In the behavioral model, the comparator offset, resistor mismatches and the amplifier gain errors are all modeled. During the sampling phase, around 0.3 LSB (12-bit level) input-referred additive noise is added to the input of the ADC, which corresponds to the noise during sampling phase in the actual ADC design. During the conversion phase, the random noise is also applied to the comparator.
The ADC is first tested (in Matlab) with the standard histogram ramp test. This ADC is randomly generated. In the histogram test, around 20 hits per code ramp is applied. The INL estimated from histogram with end point fitting is shown in red lines in Figure 4.8. The same ADC, with the same matching and gain errors, is tested again with the proposed method. The input signal is a sine wave with frequency around 5% of the Nyquist rate. The total number of samples is 4096 (2^{12}). The same amount of noise variance in the sampling and conversion phase is added. The result from the proposed method is shown in blue in Figure 4.8. The INL from the proposed method shows an extended INL range which is described in section 4.3. The INL from histogram matches the blue line for most codes but has transitions in the middle of the overlapped region of adjacent segments of blue lines.

To compare the INL effect in the major transitions in the histogram test, two different cases are simulated. In the first case, the same ADC is tested with histogram without conversion errors/comparator noise. The first major transition in the first stage is shown in the left plot of Figure 4.9. The histogram results (the red plot) has a sharp transition at around code 190, which is the level where the sub-ADC makes a transition from 0 to 1 in the first stage. In the second case, a significant amount of conversion errors and comparator noise are added. The estimated INL

Figure 4.8 INL testing comparison
using histogram is shown in the right plot in Figure 4.9. It can be clearly seen that the transition is a smooth curve rather than a sharp one. When the input voltage is close to the transition, the noise or transient error causes the sub-ADC to have an output of either 0 or 1, depending on the amount of noise at that particular comparison time instance. With a large number of samples at this region, the averaged error depends on the percentage of code 0 and code 1. This effect also applies to the transition in the second stage.

In the proposed method, only 4096 samples are used while the histogram method uses more than 80,000 samples. In addition, the proposed method shows extended INL results which cannot be obtained from the conventional histogram test. It achieves 20-time test time reduction while achieving better test coverage.

The calibration is also verified with the same ADC in simulation. The calibrated output codes are based on equation (4.9). The same histogram test is used for comparison before and after calibration as an independent verification. Figure 4.10(a) shows the INL before and after calibration with histogram test. The INL after calibration across all codes is well below +/- 1 LSB. It shows significant linearity improvement with the proposed testing and calibration method. As shown
in Figure 4.10(b), there are many missing codes at major transitions before calibration. After calibration, the missing codes are gone and the DNL is around +/- 0.5 LSB.

To verify the robustness over different ADCs, 100-run simulations are performed. In each simulation, the ADC is randomly generated with random errors. In each test, the same amount of noise (0.3 LSB input-referred noise) is applied. And the proposed method takes 4096 samples. The 100 results are shown in Figure 4.11. The red lines show the maximum absolute INL of each ADC in the 100 simulations before calibration and the blue lines show the maximum absolute INL of each ADC after calibration. The maximum INL after calibration are all less than 1 LSB while the INL before calibration can be as large as 10 LSB and as small as 3 LSB. It shows that the proposed method can work robustly and effectively across different ADC performances.

4.6 Measurement

The prototype ADC was fabricated in TSMC 40nm technology and it is operated at 125MS/s sampling rate. The input signal is generated from Agilent N5171B and the clock signal is generated from Agilent E8267D. Both equipment can provide high quality signals to the ADC. The output of the ADC is collected from Tektronix MSO58 oscilloscope. On the PCB, the input signal is bandpass-
filtered and the voltage reference is buffered and also filtered to remove unwanted noise. The 5-bit raw codes for each stage are sent out as pulse-amplitude modulation (PAM) to reduce the number of pads. The proposed method including calibration, is implemented in Matlab for demonstration. With proper biasing and filtering, the ADC can achieve relatively decent performance. However, some harmonic bins are still quite high. After careful tuning to avoid uncalibratable errors, the best performance we can achieve is 9.7 bit ENOB. The ADC is first tested with the histogram sine wave test. The total number of samples is $2^{15}$. The ADC is then tested and calibrated with the proposed method with only $2^{12}$ samples. The end-point fitting INL comparison results are shown in Figure 4.12(a). The INL is improved from 2.5 LSB to 0.8 LSB. The spectral test results are shown in Figure 4.12(b) with an input frequency of 2 MHz. The SNDR is improved from 60 dB to 67 dB. And the SFDR is improved from 64 dB to more than 84 dB. All the harmonics are significantly reduced. Both low frequency and high spectral performance are summarized in Table 4.1. Before calibration, the spectral performance for both low frequency and high frequency are limited by the static errors. Therefore, the THD/SNDR/SFDR are very similar for both low frequency and high frequency tests. After calibration, the static errors are greatly reduced. In the low frequency test, the harmonics are mostly removed. In the high frequency test, the errors are mostly from the
dynamic errors, such as settling errors and clock jitter, which cannot be corrected with INL/DNL calibration. Nevertheless, the static error calibration can still provide a significant improvement in the spectral performance.

![INL Comparison](image1)

(a) INL comparison in measurement.

![FFT comparison](image2)

(b) FFT comparison in measurement with 2.1MHz input.

Figure 4.12 Performance comparison before and after calibration

<table>
<thead>
<tr>
<th></th>
<th>Low Frequency (2.1MHz)</th>
<th>High Frequency (61MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Before Cal</td>
<td>After Cal</td>
</tr>
<tr>
<td>THD (dB)</td>
<td>63.75</td>
<td>87.77</td>
</tr>
<tr>
<td>SNDR (dB)</td>
<td>60.25</td>
<td>67.04</td>
</tr>
<tr>
<td>SFDR (dB)</td>
<td>64.18</td>
<td>84.78</td>
</tr>
<tr>
<td>ENOB (bit)</td>
<td>9.7</td>
<td>10.8</td>
</tr>
</tbody>
</table>

**4.7 Conclusion**

In the conventional histogram test, large number of samples are needed to average out the noise, which results in a long test time and increases the test cost. In addition, the code density based INL results cannot be directly used for calibration in the redundant ADCs. The proposed testing
and calibration method significantly reduces the test time for multi-bit per stage pipelined ADCs. The
non-parametric INL model uses a very small number of parameters to accurately represent the
nonlinearity of the redundant ADC. Thus the noise averaging is greatly improved. It achieves more
than 10 times test time reduction compared to histogram test while achieving a better estimation
result. The test results can be directly used for calibration to improve the ADC performance with
digital post-processing.

In simulations, the estimated INL accurately represents the nonlinearity of the ADC with an
extended range in the redundant region. The statistical simulation shows that the ADC INL can
be calibrated well below 1 LSB under different variations and performances, which demonstrates
the robustness of the proposed method. In the measurement, the 12-bit pipelined ADC has an INL
less than 1 LSB and achieves 10.84 bit ENOB after calibration. The SFDR is improved by more
than 20dB.

The proposed method can be used either in production testing to reduce test time and test
cost or for calibration to improve the ADC linearity performance. From a design perspective, by
using the proposed calibration method, the design requirement can actually be relaxed and the
power/area/speed can all be improved. This method can be easily applied to existing industrial
pipelined ADCs to achieve low-cost testing and low-cost calibration to improve performance with
just extra digital calibration circuits.
State-of-the-art ADC built-in self-test methods relax the test stimulus linearity but require a constant voltage shift during testing. A low-cost on-chip built-in self-test solution with a modified R2R DAC structure is developed as a signal generator and a voltage shift generator for ADC linearity test. The proposed DAC is a subradix-2 R2R DAC with a constant voltage shift generation capability. The subradix-2 architecture avoids positive voltage gaps caused by random mismatches, which relaxes the DAC matching requirements and reduces the design area. The DAC is fabricated in TSMC 40nm technology with a small area of 0.02mm$^2$. The matching of the 14-bit DAC is only at 7-bit level. Measurement results show that it is capable of testing a 15-bit ADC accurately with 0.5 LSB estimation error.

5.1 Introduction

The analog-to-digital converters (ADC) are the bridge between the analog world and the digital processing domain. The ADCs have been widely used in modern electronic systems. In recent years, there is an increasing demand of ADCs in the Internet-of-things (IoT) and automotive areas. In the IoT applications, the devices should be small and low power. But testing the ADC is usually expensive. With billions of IoT devices, small test cost saving will result in a huge total cost saving. In the automotive application, especially the autonomous driving, the reliability and the functional safety of the ADC are extremely important. The ADC has to be fully tested before the shipment. However, due to the environmental variations and the aging effects, the performance of the ADC may degrade and fail to meet the expected specifications. In both scenarios, the built-in self-test (BIST) solution for ADC will be the ultimate solution from both test cost reduction and reliability points of view.
Industrial standard histogram methods are the conventional ADC testing method (3; 2). In the histogram test, a linear ramp or a pure sine wave is applied as the input of the ADC. With many samples per ADC code, the integral nonlinearity (INL) and differential nonlinearity (DNL) are determined based on the output code density. However, there are two challenges in the conventional histogram method: (a) the input signal has to be much more linear than the ADC under test; (b) for an N-bit ADC, the test data should be at least $10 \times 2^N$ (10 hits per code) to reduce the quantization noise and random noise effect. Therefore, the ADC testing cost is very high due to the linearity requirement of the signal generator and the long test time. These two challenges make low-cost ADC BIST difficult to implement on chip.

Many researchers are attempting to generate the high-precision ramp or sine wave signal on chip to implement the histogram test (11; 7; 9; 12; 13). First of all, to test an N-bit ADC, the conventional histogram method requires the linearity of the stimulus to be $N+3$ bits or higher. To test a 12-bit ADC, the signal generator needs to have less than +/- 0.5 LSB error in the 15-bit level, which is possible but not cost-effective. For higher resolution ADCs, such as 15 bits or 18 bits, the linear ramp generators become nearly impossible for certain speed. The most recent study demonstrates a step-wise ramp generator with near 14.5-bit ENOB, which can only be used to test a 11-bit ADC with an accuracy of +/- 0.3 LSB (13).

In the contrast, methods have been proposed to relax the stringent requirements (22; 24; 36; 38; 37). In (22), the stimulus error identification and removal (SEIR) algorithm is proposed to relax the stimulus linearity requirement. Instead of testing the ADC with a linear ramp, two nonlinear ramp signals are applied to the ADC, but these two ramp signals have a constant voltage shift or a constant attenuation in between. With digital processing, the nonlinearity of the signals can be identified and removed. However, the SEIR method is based on the histogram test, which still requires a long test time. The long test time will still dominate the test cost. In addition, for automotive application, especially from functional safety point of view, the BIST needs to be executed within a short time interval to meet functional safety requirement. In (24), similar to SEIR, a voltage shift is applied to the nonlinear input signal. A polynomial fitting method is used
to model the ADC nonlinearity to reduce the test time. It has demonstrated that the test time can be reduced by more than 50%. In (36; 37), the ultra-fast stimulus error removal segmented model identification of linearity errors (USER-SMILE) algorithm is proposed to combine the (22) and (35) to use the non-parametric segmented model to represent the ADC nonlinearity. It can achieve more than 10-time test time reduction. Similar to SEIR algorithm, the key of such ADC BIST algorithm is to have a constant voltage shift applied to the nonlinear input signal.

This chapter proposes a novel low-cost signal generator with voltage shift generation for ADC built-in self-test. The proposed signal generator is based on the R2R digital-to-analog converter (DAC) but with a reduced radix. An additional resistor is added to create the constant voltage shift at the DAC output. The prototype of the DAC is fabricated in the TSMC 40nm technology. The DAC has a 14-bit resolution and the voltage shift is around 3% of the DAC output full range. The USER-SMILE algorithm is used to test the ADC with the proposed circuit. The fabricated DAC only has 7-bit linearity as a normal DAC. But it is capable of testing a 15-bit ADC with less than 0.5 LSB estimation error in the measurement.

This chapter is organized as follows. Section 5.2 reviews the fundamental the ADC built-in self-test algorithm. Section 5.3 presents the proposed R2R DAC design and analyzes the design requirements. Detailed equations are shown in the appendix. The design requirements and equations are validated with simulations in section 5.4. The measurement results are described in section 5.5 and the conclusion is drawn in section 5.6.

### 5.2 USER-SMILE Algorithm Review

This section reviews the USER-SMILE algorithm for the ADC BIST. The USER-SMILE algorithm is proposed to achieve built-in self-test for high resolution ADCs with segmented architectures. It relaxes the input signal linearity requirement and reduces the test time significantly. It takes a nonlinear input, but samples it twice: one with a constant voltage shift and one without the voltage shift. With a stimulus error removal technique, it no longer requires a linear input for the ADC testing. In the meantime, it models the ADC INL with a non-parametric segmented
model, which enables a small number of parameters to accurately represent the ADC nonlinearity and dramatically reduces the test time and improves the noise averaging capability (36; 37).

If the ADC with $N$-bit resolution is divided into 3-level segmentation, the output code can be defined as the most-significant bit (MSB), the intermediate significant bit (ISB) and the least-significant bits (LSB). Then, the INL for code ADC code $C$ can be modeled as:

$$\text{INL}(C) = E_M(C_{\text{MSB}}) + E_I(C_{\text{ISB}}) + E_L(C_{\text{LSB}})$$  \hspace{1cm} (5.1)

where $C_{\text{MSB}}$, $C_{\text{ISB}}$ and $C_{\text{LSB}}$ are the MSB, ISB and LSB segment codes respectively. And $E_M$, $E_I$ and $E_L$ are the modeled segmented error parameters in MSB, ISB, and LSB segments respectively. For a 15-bit ADC with 5-5-5 segmentation, the total number of model parameters is 96 ($2^5 + 2^5 + 2^5$) while the conventional histogram method will have 32,768 ($2^{15}$) INL parameters.

The block diagram of USER-SMILE is shown in Figure 5.1. The signal generator produces a nonlinear ramp signal. The ADC takes two samples for the same signal $V_{\text{sig}}$. In the first sampling, a constant voltage $\alpha$ is added to this signal. In the second sampling, the ADC directly samples the $V_{\text{sig}}$.

Define the two output codes of ADC as $C_1$ and $C_2$ respectively. The relation between the output and input is:

$$x_1 + \text{noise}_1 = V_{\text{LSB}} \cdot C_1 + V_{\text{LSB}} \cdot \text{INL}(C_1) + V_{\text{offset}},$$  \hspace{1cm} (5.2)

$$x_2 + \text{noise}_2 = V_{\text{LSB}} \cdot C_2 + V_{\text{LSB}} \cdot \text{INL}(C_2) + V_{\text{offset}}.$$  \hspace{1cm} (5.3)

where $V_{\text{offset}}$ is the offset of the ADC, $V_{\text{LSB}}$ is the LSB voltage of the ADC, and $\text{noise}_1$ and $\text{noise}_2$ are the additive noise (input-referred) in the two sampling. By subtracting these two equations and replacing the input difference with a constant value $\alpha$, a new equation can be obtained:

$$\alpha + \text{noise}_1 - \text{noise}_2 = V_{\text{LSB}} \cdot (C_1 - C_2) + V_{\text{LSB}} \cdot (\text{INL}(C_1) - \text{INL}(C_2)).$$  \hspace{1cm} (5.4)
The input signal is now completely removed with only the output codes and the INL left. Therefore, it doesn’t require an accurate input source. With many sets of equations, the INL model parameters in (5.1) as well as the voltage shift $\alpha$ can be estimated with the least square method or other system identification methods.

The key of the USER-SMILE algorithm is to have a signal generator with a voltage shift generation. Designing a highly linear signal generator is difficult. But it is easier and more cost-effective to have a nonlinear signal generator with a constant voltage shift generation. Therefore, the efforts have been moved to the shift generator design.

![Figure 5.1 USER-SMILE implementation](image)

**5.3 R2R DAC Design**

**5.3.1 Signal Generator Requirement**

The linearity of the signal generator for ADC BIST is significantly relaxed benefiting from the ADC BIST algorithm. But there are still other requirements: voltage coverage, resolution, repeatability and voltage shift. First, the voltage range of the signal generator should cover the entire ADC input range. It not only requires the minimum and maximum voltages of the signal generator to exceed or be approximately equal to the ADC input range, but also requires that there is no big gap of the signal generator output in the entire ADC input range. It is usually fine with small gaps as the segmented models can estimate the error in the small gaps from the model parameters estimated by other voltage ranges. But if the gap is too large, some model parameters, or even the MSB model parameter cannot be hit. In this case, the missing model parameters cannot be estimated. Second, the signal generator resolution should be as high as possible. Although lower
resolution signal generator can be used to test the ADC, the signal generator resolution cannot be too low such that the ADC quantization error will finally dominate the estimation error. It is recommended that the signal generator has similar resolution or 1~2 bits lower resolution than the ADC under test. Third, for the USER-SMILE algorithm, the signal generator voltage should be repeatable in order for the ADC to sample the same voltage twice. Lastly and most importantly, the signal generator is able to generate a constant voltage shift in the output voltage. Although methods have been proposed to generate the voltage shift inside the capacitive SAR ADC itself (53), it is always better to have an independent and generic shift and signal generator for all types of ADCs. Such an independent shift generator should be also robust to input MUX nonlinearity and capacitor voltage dependency (52).

In the SEIR algorithm, the nonlinearity of the stimulus is modeled and identified with high order polynomials or sinusoidal functions. Therefore, the signal generator for SEIR should have a smooth nonlinearity. And SEIR requires more samples per codes to test the ADC since it is still based on the histogram test. Therefore, the signal generator for SEIR should have a few bits more resolution than the ADC under test. But the USER-SMILE doesn’t require the input to be a smooth ramp. It can be any types of input signals even for some arbitrary voltages that satisfy the above requirements (voltage coverage, resolution, repeatability and voltage shift).

There are many types of signal generators. Charge-pump based signal generator is small and low cost. But it is difficult to control a small voltage increment. The resistor string DAC is monotonic and the DNL is usually very small. However, the area increases exponentially as the resolution increases. Capacitive binary weight DAC requires timing circuits and driving circuit. The capacitor area is also usually large. An ideal candidate is the R2R DAC, which can be easily built for a high resolution without precise timing control.
5.3.2 Subradix-2 Architecture

The architecture for the R2R DAC is shown in Figure 5.2, where $b[0]$ is the MSB bit for an N-bit DAC. A fully-differential R2R DAC is implemented in the actual design. The plot shows the single-ended one for simplicity.

![Figure 5.2 Standard R2R DAC](image)

Since the linearity of the R2R DAC is not important in this application, the resistor area can be sized very small. Normally, $R_2$ is twice as $R_1$ for a binary weighted R2R DAC. However, the poor matching of resistor could cause large positive gaps in DAC output voltage, which needs to be avoided in the USER-SMILE algorithm as described in the previous subsection. To resolve this issue, the radix is designed to be less than 2. In this case, all major transitions will have less than

![Figure 5.3 Subradix-2 DAC transfer function (subradix-2 for the first 2 MSB bits)](image)
1-LSB jumps that are translated into large negative DNL. An example is shown in Figure 5.3. Only the first two MSB bits are in subradix-2. There are negative jumps in the 1/2, 1/4 and 3/4 locations. Compared to the normal radix-2 DAC, the slope of all other codes in a subradix-2 DAC transfer function is slightly higher. Therefore, the DNL for all other codes are slightly positive. In the actual design, all bits are set to be subradix-2. For the USER-SMILE algorithm, such large negative DNL is completely fine since the USER-SMILE algorithm will cancel any input signal nonlinearity as long as the DAC voltage can cover the ADC input range without gaps. Due to mismatches, these major transitions can have different values. The radix should be determined to have always-negative DNL at major transitions regardless of mismatches.

Define that the impedance looking into the left of the node \( V_k \) is \( R_{\text{imp}}[k] \) as shown in Figure 5.2.

So, for the last bit (bit N-1), the impedance looking into the left of \( V_{N-1} \) is \( R_{\text{imp}}[N - 1] = (R_1[N] + R_1[N - 1])||R_2[N - 1] \), assuming the switch on-resistance is neglectable. For the \( k \)-th bit, there will be the following equation:

\[
R_{\text{imp}}[k] = (R_{\text{imp}}[k + 1] + R_1[k])||R_2[k].
\] (5.5)

If there is an infinite number of bits, \( R_{\text{imp}}[k] \) and \( R_{\text{imp}}[k - 1] \) will be equal. Then, the value of \( R_{\text{imp}}[k] \) can be solved:

\[
R_{\text{imp}}[k] = \sqrt{R_1[k] \cdot R_2[k] + \frac{R_1[k]^2}{4} - \frac{R_1[k]}{2}}.
\] (5.6)

The same result can be applied to all other bits. When \( R_2 \) is twice of \( R_1 \), \( R_{\text{imp}}[k] \) will be equal to \( R_1[k] \), which is the conventional binary weighted R2R DAC.

In the actual manufacturing, the resistor will have random mismatches. Define the actual resistance of \( R_2 \) as \( R_2^i + \Delta R_2 \) where \( R_2^i \) is the ideal resistance and \( \Delta R_2 \) is the variation in \( R_2 \). Assume \( \Delta R_2 \) follows Gaussian distribution and the standard deviation is \( \sigma_{R_2} \). The mismatch in every resistor is assumed to be independent of each other and the variance of the mismatches is inverse proportional to the area of the resistor.

For the first MSB bit, the output voltage for MSB equal to 1 and all other bits equal to 0 is:

\[
V_{\text{MSB}} = V_{\text{refl}} + (V_{\text{refh}} - V_{\text{refl}}) \cdot \frac{R_{\text{imp}}[1] + R_1[0]}{R_{\text{imp}}[1] + R_1[0] + R_2[0]}.
\] (5.7)
To ensure no big positive gap in the MSB bit transition, $V_{\text{MSB}}$ must be less than or equal to $1 - V_{\text{MSB}}$. Therefore, we have the following relation:

\[
V_{\text{refl}} + (V_{\text{refh}} - V_{\text{refl}}) \cdot \frac{R_{\text{imp}}[1] + R_1[0]}{R_{\text{imp}}[1] + R_1[0] + R_2[0]} < V_{\text{refl}} + (V_{\text{refh}} - V_{\text{refl}}) \cdot \frac{R_2[0]}{R_{\text{imp}}[1] + R_1[0] + R_2[0]}.
\] (5.8)

After simplification, the following relation is obtained:

\[
R_{\text{imp}}[1] + R_1[0] < R_2[0]
\] (5.9)

Such a relation is also valid for any following bits if the positive gap needs to be avoided in the transition. So, for the $k$-th bit, we have:

\[
R_{\text{imp}}[k] + R_1[k - 1] < R_2[k - 1]
\] (5.10)

Replace the actual resistance with the ideal resistance and its mismatches:

\[
R_{\text{imp}}^i[k] + R_1^i[k - 1] + \Delta R_{\text{imp}}[k] + \Delta R_1[k - 1] < R_2^i[k - 1],
\] (5.11)

where $R_{\text{imp}}^i$ is the ideal value of $R_{\text{imp}}[k]$ without mismatches; $\Delta R_{\text{imp}}[k]$ is the mismatch for $R_{\text{imp}}[k]$; and similar definition applies to $R_1[k - 1]$ and $R_2[k - 1]$. Replacing $R_{\text{imp}}[k]$ with (5.6), the following inequality is obtained:

\[
\Delta R_{\text{imp}}[k] + \Delta R_1[k - 1] - \Delta R_2[k - 1] < R_2^i - R_1^i - \sqrt{R_1^i \cdot R_2^i + \left(\frac{R_1^i}{2}\right)^2}
\]

\[
= \frac{R_2^i (R_2^i - 2R_1^i)}{R_2^i - \frac{R_1^i}{2} + \sqrt{R_1^i R_2^i + \left(\frac{R_1^i}{2}\right)^2}} < \frac{2(R_2^i - 2R_1^i)}{3},
\] (5.12)

where $R_2^i$ is slightly greater than $2R_1^i$. Otherwise, the left side of the inequality can be positive and this inequality will not be valid. $R_{\text{imp}}$ has a similar value as $R_1$ but is obtained with multiple $R_2$ and $R_1$ parallel/series combinations. Therefore, the variance of $\Delta R_{\text{imp}}$ should be less than the
variance of $\Delta R_1$. For estimation, the variance of $R_1$ ($\sigma_{R_1}^2$) is used for $\Delta R_{imp}$. The variance for $R_2$ can be estimated as $2\sigma_{R_1}^2$. Therefore, the variance of $\Delta R_{imp}[k] + \Delta R_1[k-1] - \Delta R_2[k-1]$ is estimated as $4\sigma_{R_1}^2$. If the R2R DAC is designed to meet the $3\sigma$ bound, the relation between the $R_2^i$ and $R_1^i$ is:

$$R_2^i - 2R_1^i > 9\sigma_{R_1}. \quad (5.13)$$

For example, if $\sigma$ for $R_1$ is 100Ω and $R_1 = 10K\Omega$. Then, $R_2$ should be sized more than 20.9KΩ to avoid positive jumps in the transfer curve.

In all the previous derivations, it is assumed that the resolution is infinite. For a finite-resolution DAC, the same conclusion can be drawn if $R_1^i[N]$ is equal to $R_{imp}^i$. If $R_1^i[N]$ is equal to $R_1^i$ as other $R_1$, $R_{imp}^i[k]$ will be slightly smaller but the conclusion in (5.12) and (5.13) are still valid.

### 5.3.3 Voltage Shift Generation

Many ADC built-in self-test algorithms require the offset injection or voltage shift generation capability (24; 36; 38; 37; 23). And many voltage shift generators are proposed (53; 52; 54; 44; 70). Among these methods, some are built-in as part of the signal generators and some are built inside the SAR ADC with capacitor switching. In (53) and (52), two different methods of generating a constant offset for SAR ADC are proposed. Both of two methods can achieve high constancy by either injecting the charge through the additional capacitor or using IR drop to create a constant voltage shift. In (54), a stand-alone shift generator is integrated with an on-chip signal generator. And in (70), a modified version of the shift generator is designed to achieve true rail-to-rail performance with the high constancy of the offset. However, there are some limitations in the methods mentioned above. For the methods in (53) and (52), the design of the shift generator is dependent on the ADC structure, which is not suitable for different types of ADCs in SoC. On the other hand, in (54) and (70), the shift generators are independent, but the structure of the BIST is relatively more complicated, consuming more power and requiring more design efforts. To eliminate the dependency on the ADC architecture, an independent low-power shift generator with simple structure is a better choice for on-chip implementation. For the proposed R2R architecture,
a simple modification can be made to generate a voltage shift to meet these requirements. As shown in Figure 5.4, an extra resistor $R_s$ is added.

![Figure 5.4 R2R DAC Voltage Shift Generation](image)

The equivalent circuit is shown in Figure 5.5. $V_{eq}$ is the equivalent output voltage at the original output node before adding the $R_s$. When $b[s]$ is 0, the resistor $R_s$ is connected to $V_{refl}$. Then, the output voltage is:

$$V_{out|b[s]=0} = \frac{R_s V_{eq} + R_{imp}[0] V_{refl}}{(R_s + R_{imp}[0])}.$$  \hspace{1cm} (5.14)
When \( b[s] \) is 1, the resistor \( R_s \) is connected to \( V_{\text{refh}} \). Then, the output voltage is:

\[
V_{\text{out}}|_{b[s]=1} = \frac{R_s V_{\text{eq}} + R_{\text{imp}}[0] V_{\text{refh}}}{(R_s + R_{\text{imp}}[0])}.
\] (5.15)

The difference between the two is:

\[
V_{\text{shift}} = V_{\text{out}}|_{b[s]=1} - V_{\text{out}}|_{b[s]=0} = \frac{R_{\text{imp}}[0] (V_{\text{refh}} - V_{\text{refl}})}{(R_s + R_{\text{imp}}[0])}.
\] (5.16)

It can be seen that the voltage shift generated by the \( s \) bit is independent of the codes of the DAC. So, this voltage shift is expected to be constant over all the DAC codes. The value of the voltage shift is determined by the ratio of \( R_s \) and \( R_{\text{imp}}[0] \). If 10% voltage shift is expected, then, the \( R_s \) should be around 90k\( \Omega \) if \( R_1 \) is around 10k\( \Omega \) (\( R_{\text{imp}}[0] \) is slightly greater than \( R_1 \)). The resistance of \( R_s \) seems to be very large. But the exact value of \( R_s \) is not important in the ADC BIST algorithm. Therefore, it can be designed with small area to achieve an inaccurate large resistance.

However, in the actual implementation, the variations of switch on-resistance affect both the DAC output voltage and the output impedance, which affects the voltage shift constancy. To have a complete analysis, the fully-differential version DAC is used, as shown in Figure 5.6. The positive side and the negative side are identical in terms of the structure. But the reference voltages connected to each bit are reversed. Those who are connected to \( V_{\text{refh}} \) in the positive side is now connected to \( V_{\text{refl}} \) in the negative side. And the last resistor \( R_1[N] \) is connected to \( V_{\text{refh}} \). To distinguish the two sides, all the notations are adding a superscript (p) in the positive side and a superscript (n) in the negative side. For example, the \( R_s^{(p)} \) stands for the \( R_s \) in the positive side. The differential output voltage is defined as

\[
V_{\text{diff}} = V_{\text{out}}^{(p)} - V_{\text{out}}^{(n)}.
\] (5.17)

The differential output voltage when the \( b[s] \) equal to 0 is:

\[
V_{\text{diff}}|_{b[s]=0} = V_{\text{out}}^{(p)}|_{b[s]=0} - V_{\text{out}}^{(n)}|_{b[s]=0} = \frac{R_s^{(p)} V_{\text{eq}}^{(p)} + R_{\text{imp}}^{(p)}[0] V_{\text{refl}}}{(R_s^{(p)} + R_{\text{imp}}^{(p)}[0])} - \frac{R_s^{(n)} V_{\text{eq}}^{(n)} + R_{\text{imp}}^{(n)}[0] V_{\text{refh}}}{(R_s^{(n)} + R_{\text{imp}}^{(n)}[0])}.
\] (5.18)
When $b[s]$ is 1, the differential output voltage is:

$$V_{\text{diff}b[s]=1} = V_{\text{out}b[s]=1} - V_{\text{out}b[s]=1}$$

$$= \frac{R_{s}(p) + R_{\text{imp}[0]}^{(p)}}{(R_{s}(p) + R_{\text{imp}[0]}^{(p)})} - \frac{R_{s}(n) V_{\text{eq}}^{(n)} + R_{\text{imp}[0]}^{(n)} V_{\text{refh}}}{(R_{s}(n) + R_{\text{imp}[0]}^{(n)})}.$$  

(5.19)

The difference between the two differential output voltages is the differential voltage shift:

$$V_{\text{shift}} = V_{\text{diff}b[s]=1} - V_{\text{diff}b[s]=0}$$

$$= \frac{R_{\text{imp}[0]}^{(p)}(V_{\text{refh}} - V_{\text{refl}})}{(R_{s}(p) + R_{\text{imp}[0]}^{(p)})} + \frac{R_{\text{imp}[0]}^{(n)}(V_{\text{refh}} - V_{\text{refl}})}{(R_{s}(n) + R_{\text{imp}[0]}^{(n)})}.$$  

(5.20)

From this equation, the voltage shift of the differential outputs is still constant as long as the impedance of $R_{\text{imp}[0]}^{(p)}$ and $R_{\text{imp}[0]}^{(n)}$ is constant for all DAC input codes. However, the constant impedance requirement is not true if the PMOS and NMOS switch on-resistance are different.
addition, the PMOS/NMOS switches also have local mismatches. Define the k-th bit PMOS switch on-resistance in the positive side as:

\[ R_{onp}^{(p)}[k] = R_{onp} + \Delta_{Ronp}^{(p)}[k], \]  

(5.21)

where \( R_{onp}^{i} \) is the ideal on-resistance (for all PMOS switches) with the same gate voltage and \( \Delta_{Ronp}^{(p)}[k] \) is the local mismatch for the k-th bit PMOS on-resistance in the positive side. All other switches on-resistance is defined in the same way. Since all PMOS or NMOS switches are biased with the same voltage, the ideal on-resistance is the same for all PMOS or all NMOS switches without mismatches. Define the difference between the ideal PMOS and the ideal NMOS on-resistance as:

\[ \Delta R_{on} = R_{onp}^i - R_{onn}^i. \]

(5.22)

Including the switch on-resistance, (5.20) is no longer valid since the denominator changes for PMOS and NMOS switches. Therefore, combining (5.18) and (5.19) with the switch on-resistance, the \( V_{shift} \) becomes:

\[
V_{shift} = \frac{(R_s^{(p)} + R_{onp}[s])V_{eq}^{(p)} + (R_{imp}^{(p)}[0])V_{refh}}{R_s^{(p)} + R_{onp}[s] + R_{imp}^{(p)}[0]} - \frac{(R_s^{(n)} + R_{onn}[s])V_{eq}^{(n)} + (R_{imp}^{(n)}[0])V_{refl}}{R_s^{(n)} + R_{onn}[s] + R_{imp}^{(n)}[0]}
\]

(5.23)

\[
\max(V_{shift}) - \min(V_{shift}) = \frac{\text{mean}(V_{shift})}{4R_{imp}^i + \sqrt{2\sigma_{Ron}^2} + \frac{\sqrt{2\sigma_{Ron}^2}}{R_s^i}}
\]

(5.24)

where \( \sigma_{Ron} \) is the switch on-resistance variation standard deviation; \( R_s^i \) is the ideal resistance for shift resistor \( R_s \); and \( R_{imp}^i \) is the ideal impedance without mismatches before adding the shift
resistor. Therefore, in a differential architecture, only the local random mismatches ($\sigma_{Ron}$) for PMOS and NMOS switched are important. The difference between ideal PMOS and NMOS on-resistance ($\Delta R_{on}$) causes errors in voltage shift of the single-ended output. But in the differential output, the voltage shift errors in positive side and negative side cancel each other.

To test a 15-bit ADC with 0.2 LSB estimation error due to the shift constancy, the most conservative constancy requirement should be less than 12.2ppm. With $R_1 = 10K\Omega$, $R_2 = 20.9K\Omega$ and $R_s = 250K\Omega$, the value of $\sigma_{Ron}$ should be less than 0.5\Omega. However, this is the most conservative estimation. In most cases, a few Ohms variation can achieve less than 10ppm constancy. For a 12-bit ADC testing, the on-resistance mismatch can be relaxed to 4\Omega, which is much easier to realize.

5.4 Simulations

The proposed R2R DAC differential architecture is modeled in Matlab to verify the subradix-2 architecture and the shift constancy of the DAC. A 14-bit R2R DAC is modeled, which is the same as the fabricated one. In the ideal case, the value of $R_1$ is 10k\Omega and the value of $R_s$ is 200k\Omega. The value of $R_2$ is determined by the variation of the resistor based on equation (5.13).

The target of the subradix-2 architecture is to avoid large positive gaps in the major transitions. In the DNL plots, all the major transitions should have large negative DNL values. As a result, the remaining codes will have very small positive values. The DAC is randomly simulated for 1000 times with different mismatches and the most positive DNL for each simulation is saved. 100\Omega $\sigma_{R1}$ is used in the simulation and the value of $R_2$ is therefore designed to be 20.9k\Omega. The 1000 maximum DNL values are shown in Figure 5.7. The maximum value over the 1000 simulations is less than 0.3 LSB. Therefore, the R2R DAC has no large positive jumps over the entire transfer curve.

To analyze the voltage shift constancy, a large number of simulations are done to obtain the averaged value and the standard deviation of the voltage shift constancy. The non-idealities include the PMOS and NMOS on-resistance difference in ideal case (no mismatch), the PMOS/NMOS on-
resistance mismatches and the resistor mismatches. The voltage shift constancy is defined as the worst case voltage shift difference over the averaged voltage shift value (expressed in ppm). To test a 15-bit ADC, the most conservative voltage shift constancy is 12.2ppm to achieve less than 0.2 LSB estimation error in the ADC linearity test.

The on-resistance of PMOS switch and NMOS switch are determined by the switch sizes and the bias condition of the transistors. Therefore, for different voltage reference, the on-resistance will not be the same. Even without mismatches, there will be a difference between PMOS and NMOS switches. All the PMOS switches will have the same on-resistance and all the NMOS switches will have the same on-resistance. The relation between the voltage shift constancy and the PMOS/NMOS on-resistance difference is shown in Figure 5.8. The on-resistance difference between PMOS and NMOS have a very small impact on the voltage shift constancy. Even with 10Ω on-resistance difference, the voltage shift constancy is around 2ppm.

Figure 5.7  maximum DNL over 1000 random simulations

Figure 5.8  Shift constancy VS PMOS/NMOS on-resistance difference
In the second case, the switches on-resistance local mismatches are modeled. In this case, both PMOS and NMOS have local mismatches and they are all independent. Figure 5.9 shows that the on-resistance mismatches have a significant impact on the voltage shift constancy. Therefore, the switches need to be sized large enough to minimize the local mismatches. And the dominant errors are from the MSB bit switches.

In the third case, the mismatches of the resistors are modeled. In the previous section, it has been shown that the voltage shift is constant if the impedance of the R2R DAC is constant with different DAC codes. Therefore, simulation with only resistors mismatches will have 0ppm voltage shift constancy. So, in addition to the resistor mismatches, 0.5Ω on-resistance local variation and 10Ω PMOS/NMOS on-resistance difference are added. The relation between the voltage shift constancy and the resistor variations is shown in Figure 5.10. There is very little change in the voltage shift constancy with an increasing resistor variation from 50Ω to 500Ω.
As a conclusion of the voltage shift constancy, the dominant error is from the switch local mismatches. The PMOS/NMOS on-resistance difference has a small impact on the shift constancy but it is better to minimize the difference. And the resistor variations have almost no impact on the shift constancy.

5.5 Measurement Results

In the final DAC design, the MSB switches are sized to have less than 0.5Ω on-resistance mismatch. The PMOS on-resistance and NMOS on-resistance are sized to be approximately the same. The exact value is not important as derived above. The resistors in the main R2R DAC is sized to have around 1% mismatches. The overall design target is to be able to test a 15-bit ADC with less than 0.5 LSB estimation error (0.2 LSB from shift non-constancy and 0.3 LSB from the USER-SMILE estimation error due to noise and modeling error). The prototype of the R2R is designed and fabricated in TSMC 40nm technology. The die photo is shown in Figure 5.11. The entire DAC occupies approximately 100µm × 200µm.

Figure 5.11  R2R DAC die photo

5.5.1 Test Setup

The testbench is shown in Figure 5.12. In the test board, a high resolution ADC (TI ADS8881) is used to capture the R2R DAC output voltage. The communication of the R2R DAC is through the SPI interface. The DAC input codes are sent from the FPGA board. The ADC samples the DAC output voltage at 400KSPS sampling rate. The ADC output codes are sent back to the
FPGA and stored in the FPGA board memory. Once all the samples are collected, the memory is transferred to PC for data analysis.

5.5.2 R2R DAC Performance

![Figure 5.13 DAC output voltages](image)

In Figure 5.13(a), the transfer curves for the shifted and non-shifted DAC clearly show the non-monotonicity as a subradix-2 DAC. The reference voltage for DAC is 1.1V. Therefore, the differential output voltage has a range from -1.1V to 1.1V. The DNL (Figure 5.14) at major transitions are all negative. The most positive DNL values is nearly 0. Therefore, the DAC has no big positive gap in the transfer curve, which meets our design target. The most negative DNL is close to -700 LSB, which is expected in a subradix-2 DAC and is completely fine for testing purpose with USER-SMILE algorithm.
The difference between the two transfer curve shows a constant but slightly noisy voltage of 71.5mV (Figure 5.13(b)). With around 0.2mV noise band, it is not obvious how constant the voltage shift is. Therefore, the DAC output is used to test an ADC model to indirectly verify the constancy.

5.5.3 ADC Testing Results

In the first test, the DAC is controlled to sequentially increase the code without shift for the entire ramp. Right after this ramp, the DAC codes start from 0 again but with voltage shift bit enabled. Figure 5.15(a) shows the captured output voltages. However, with such ramp-to-ramp shifting, it shows significant inconsistency in the estimation error. One of the estimation results is shown in Figure 5.15(b). Compared to the true INL, the estimated INL using the DAC output has 1 LSB error. These errors are largely due to the 1/f noise. In the first ramp, the 1/f noise causes the ramp to drift for the entire ramp period. For the second ramp, the drift occurs again. When subtracting the two ramps, the voltage difference will have a component of the difference of two drifted noise. As the drift is at low frequency, it usually accumulates into a large error.

In order to overcome the 1/f noise drift issue, a code-to-code ping-pong shifting is used (Figure 5.16(a)). In this scheme, for a DAC code, the shift is disabled and the ADC will sample it once. For the next sample, the same DAC codes is used but the voltage shift is enabled. In the next ADC sampling, the DAC code is increased by one. This process continues until the DAC
reaches the maximum code. In this case, the voltage shift for each code will have a very small noise difference instead of the drifted noise over the entire DAC ramp, as the drift within two samples is very small. The same ADC is tested with the ping-pong shifting and the INL estimation results are shown in Figure 5.16(b). The estimation errors are well within +/- 0.5 LSB, which is a significant improvement over the ramp-to-ramp shifting. The same ADC is tested for 32 times and all the estimation errors are less than 0.5 LSB.

5.6 Conclusion

In this chapter, a low-cost ADC built-in self-test solution is proposed. With a subradix-2 R2R design architecture, large positive gaps in the transfer curve are avoided. With an extra resistor, a constant voltage shift generation is realized. The proposed circuit achieves an integrated solution for both signal generation and voltage shift generation. In the differential architecture, the voltage shift error due to the PMOS/NMOS on-resistance difference is canceled. Therefore, the proposed signal generator can achieve excellent constancy. The prototype of the DAC is designed and fabricated in TSMC 40nm technology. The linearity of the DAC is only at 6~7 bit level. When testing a 15-bit ADC, the INL estimation result shows a less than 0.5 LSB error compared to the true INL.
With the relaxed requirements on the DAC linearity and PMOS/NMOS on-resistance mismatch, the DAC can be designed with a small area. A systematic design flow is described in this chapter to achieve certain constancy requirement. The DAC can be easily integrated into modern system-on-chips. The test results can be further used for ADC calibration to improve the ADC linearity. Therefore, the life-time performance and reliability of the ADC can be guaranteed, which is suitable for applications in IoT and automotive.

Figure 5.16 Ping-pong shifting
CHAPTER 6. AN ON-CHIP ADC BIST SOLUTION AND THE BIST-ENABLED CALIBRATION SCHEME

This chapter presents a complete on-chip ADC BIST solution based on a segmented stimulus error identification algorithm known as USER-SMILE. By adapting the algorithm for efficient hardware realization, the solution is implemented towards a 1Msps 12-bit SAR ADC on a 28nm CMOS automotive microcontroller. While sufficient test accuracy is demonstrated, the solution is further extended to correct linearity errors of ADC. The entire BIST and calibration circuitry occupies 0.028mm$^2$ silicon area while enabling more than 10 times tester time reduction and >10dB THD/SFDR performance improvement over an existing structural capacitor-weight-identification calibration scheme. The added die cost is estimated to be 1/8 of the saved test cost from tester time reduction alone.

6.1 Introduction

Analog and mixed signal (AMS) circuits are widely used in system(s)-on-chip (SoCs). Among them, the analog to digital converter (ADC) has consistently been one of the world’s largest volume mixed-signal products (39). At the same time, testing ADCs has become ever more challenging in the semiconductor industry. As manufacturing cost goes down, the test cost could become dominant for SOCs that include ADCs. Therefore, there is a need to develop a low-cost way to test the ADC within SoCs. Test cost reduction of SoCs is critical due to the SoC’s high volume, large digital content and wide range of applications.

Testing ADCs is challenging due to the stringent requirements on the input stimulus and the long test time (5; 3). To test an N-bit ADC, the rule of thumb is that the input stimulus has to be at least 2 bits more linear than the ADC under test and there are $2^N - 1$ code transitions that need to be tested. In a traditional histogram test, a ramp or sine wave with sufficient resolution
and linearity/spectral purity to provide tens of hits per code (HPC) or higher is used as the input for the ADC static linearity test. This results in an extremely long test time with correspondingly high test and equipment cost.

Testing deeply embedded analog/mixed-signal blocks (such as ADCs) in an SoC is also becoming very challenging and expensive due to the lack of access to the internal nodes as well as the difficulty of maintaining adequate signal integrity while driving an accurate signal on and off chip (44). A built-in self-test (BIST) capability is thus highly desirable since it doesn’t require an external signal generator and the data can be processed using on-chip resources. Researchers have developed several methods to implement such a ramp or sine wave generator on-chip (9; 11; 71). However, building such a highly linear signal generator on-chip is not trivial and is often not practical due to the extra cost and limited performance. The stimulus error identification and removal (SEIR) (22; 23) algorithm was proposed to relax the stimulus linearity requirement. In this method, two nonlinear signals with a constant voltage shift between them are used as the stimulus. The method can distinguish between the non-linearity of the ADC and the signal generator thus accurately identifying the ADC’s integral nonlinearity (INL) and differential nonlinearity (DNL). Results shown in (22) that a 7-bit linear signal generator can be used with the SEIR technique to test a 16-bit ADC accurately. However, the SEIR is still based on the histogram method which requires a long test time.

Test time is also critically important in ADC testing. Many methods have been proposed to reduce the test time (27; 42; 30; 28; 32; 34; 24; 43; 31; 55; 33). In (35), the ultrafast segmented model identification of linearity errors (yu2012algorithm) algorithm was proposed. It takes a fundamentally different approach to modeling the ADC’s INL by applying a “segmented non-parametric model”. It achieves more than 100x test time reduction while maintaining similar or better test precision compared to a conventional histogram test. However, it still requires a highly linear input source.

In (36), adopted from SEIR and uSMILE, the ultrafast stimulus error removal and segmented model identification of linearity errors (USER-SMILE) algorithm was proposed to relax the input
linearity requirement as well as reduce the test time. In this algorithm, two nonlinear input signals are generated with a constant voltage shift in between thereby easing the design of the signal generator. The segmented INL model reduces the number of samples needed for the INL estimation. With this combination, a low-cost ADC BIST with significantly reduced test time becomes practical.

Figure 6.1  ADC BIST Subsystem Block Diagram

In this chapter, a completely on-chip ADC BIST circuit is developed based on USER-SMILE algorithm and demonstrated on a 28nm CMOS automotive microcontroller. The ADC test subsystem as shown in Figure 6.1 includes a 12-bit digital-to-analog converter (DAC), a 12-bit, 1Ms/s single-ended successive-approximation-register (SAR) ADC with a built-in voltage shift generator,
a BIST computation engine and dedicated memory cells. The silicon measurement results show a
good correlation of test results between ADC BIST and traditional histogram test.

Another aspect of this chapter is to leverage the test results to correct the 12-bit SAR ADC’s
linearity error. Although the switch capacitor based ADCs like charge redistribution SAR ADC
appearing in this chapter are popular due to their simple structure and power efficiency, the ca-
pacitor mismatches as well as other error sources limit their linearity performance as the resolution
goes high. Therefore, a method of reliably calibrating the ADC is needed. Different calibration or
trimming methods for these types of ADC have been proposed in (72; 73; 74). Capacitor switching
sequence, perturbation method, and split-ADC architecture are used in these studies to identify
and calibrate the capacitor mismatches. With full code nonlinearity error of ADC known to us by
the BIST solution, this chapter proposes a simple digital calibration which demonstrates >10dB
THD/SFDR improvement over an existing structural capacitor-weight-identification (CWI) cali-
bration method.

The remainder of the chapter is organized as follows. Section 6.2 reviews the fundamentals of
USER-SMILE algorithm. Section 6.3 discusses the design choices of the USER-SMILE algorithm
and adapts it for an efficient hardware implementation. In section 6.4, the details of the ADC
test subsystem and its operation are described. Section 6.5 presents the silicon measurement and
calibration results. Section 6.6 concludes this chapter.

6.2 Review of USER-SMILE Algorithm

In the USER-SMILE algorithm, segmented INL model is used. For an N-bit 3-level segmenta-
tion, ADC is divided into $N_{MSB}$-bit most significant bits (MSB), $N_{ISB}$-bit intermediate significant
bits (ISB) and $N_{LSB}$-bit least significant bits (LSB). And $N = N_{MSB} + N_{ISB} + N_{LSB}$. Each MSB
code has a corresponding error terms $E_M$. There are $2^{N_{MSB}}$ different codes in MSB segments and
there are $2^{N_{MSB}}$ different error terms correspondingly. Define $eE_M$ as a row vector:

$$E_M = \begin{bmatrix} E_M(0) \\ E_M(1) \\ \vdots \\ E_M(2^{N_{MSB}} - 1) \end{bmatrix}$$  \hspace{1cm} (6.1)

$E_I$ and $E_L$ are defined in a similar way for ISB and LSB. Therefore, for MSB code $C_{MSB}$, the value of its $E_M$ can be written as a vector multiplication of a column matrix and $eE_M$:

$$E_M(C_{MSB}) = \begin{bmatrix} 0 & 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} E_M.$$  \hspace{1cm} (6.2)

There is only one 1 in the $C_{MSB} + 1$ location and all the other locations are all 0s in the column matrix.

The INL for ADC output code $C$ is modeled as:

$$INL(C) = E_M(C_{MSB}) + E_I(C_{ISB}) + E_L(C_{LSB})$$

$$= \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} E_M + \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} E_I + \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} E_L$$

$$= \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \begin{bmatrix} E_M \\ E_I \\ E_L \end{bmatrix}$$ \hspace{1cm} (6.3)

where $C_{MSB}$, $C_{ISB}$ and $C_{LSB}$ are the code values for MSB, ISB and LSB bits respectively. In the column matrix, the first 1 appear at the $C_{MSB} + 1$ location; the second 1 appear at the $C_{ISB} + 1 + 2^{N_{MSB}}$ location and the last one appear at the $C_{LSB} + 1 + 2^{N_{MSB}} + 2^{N_{ISB}}$ location.

There are $K = 2^{N_{MSB}} + 2^{N_{ISB}} + 2^{N_{LSB}}$ unknowns to be solved in order to model the full-code INL. For a 16-bit ADC with 6-5-5 segmentation ($N_{MSB} = 6$, $N_{ISB} = 5$ and $N_{LSB} = 5$), only $K = 128$ unknowns need to be solved and the results can accurately reflect the actual INL.

In the USER-SMILE algorithm, two identical input signals with a constant offset $a$ between them ($V_{in}^{(1)} = V_{in}^{(2)} + a$) are applied to the ADC in two separate samples. The ADC output codes after two conversions from the two input signals are $C^{(1)}$ and $C^{(2)}$. Then, the two input voltages
can be expressed in terms of the INL:

\[ V_{in}^{(1)} + n^{(1)} = V_{lsb} \cdot [C^{(1)} + INL(C^{(1)})] + q^{(1)} \]  \hspace{1cm} (6.4)

\[ V_{in}^{(2)} + n^{(2)} = V_{lsb} \cdot [C^{(2)} + INL(C^{(2)})] + q^{(2)} \]  \hspace{1cm} (6.5)

where the noise \( n^{(1)} \) and \( n^{(2)} \) are the input-referred noise; \( q \) is the quantization noise; and \( V_{lsb} \) is 1 LSB expressed as a voltage.

By subtracting (6.5) from (6.4) and replacing the INL with the segmented model described in (6.3), (6.6) can be obtained. The term \( q^{(2)} \), \( q^{(1)} \), \( n^{(1)} \) and \( n^{(2)} \) can be considered as one noise term \( \text{noise} \). \( V_{in}^{(1)} \) and \( V_{in}^{(2)} \) are removed without knowing any information of them. The only thing left is the difference of the two input signals \( a \).

\[ C^{(1)} - C^{(2)} - a/V_{LSB} \]

\[ = - E_M(C_{MSB}^{(1)}) - E_I(C_{ISB}^{(1)}) - E_L(C_{LSB}^{(1)}) \]

\[ + E_M(C_{MSB}^{(2)}) + E_I(C_{ISB}^{(2)}) + E_L(C_{LSB}^{(2)}) + \text{noise} \]  \hspace{1cm} (6.6)

Each set of output codes has one equation. With \( M \) sets of output codes, the over-determined system can be formed in (6.7).

\[ y_d = H \begin{bmatrix} E_M \\ E_I \\ E_L \end{bmatrix} + \text{noise} \]  \hspace{1cm} (6.7)

where \( y_d \) is a column matrix of \( M \) sets of \( C^{(1)} - C^{(2)} - a/V_{LSB} \); \( H \) is an \( M \times K \) matrix where there are three +1s and three -1s in each row and all the other locations are all 0s. Each row represents one set of (6.6). The over-determined system can be solved with least square method (6.8) and the
noise term will be effectively averaged out. After estimating all the unknowns, the full code INL can be constructed using (6.3).

\[
\begin{bmatrix}
E_M \\
E_I \\
E_L
\end{bmatrix}
\approx (H^T H)^{-1} (H^T y_{dk})
\]  \hspace{1cm} (6.8)

As a summary, the USER-SMILE algorithm has the following characteristics and requirements:

### 6.2.0.1 Segmented INL Model

The segmented non-parametric INL model is intended for high resolution ADCs whose architecture facilitates a segmented structure of the INL curve. It assumes that the lower-bit errors are contributed from the same sub-circuit of the ADC, which is true for most high resolution ADCs. Take SAR ADC as an example. Each bit corresponds each capacitor in the capacitor array. If the last 4 bits are the same for different conversions, the connection for last 4 capacitors are the same and the errors contributed from these capacitors will be the same. However, for flash ADC, different codes will correspond to different nodes in the resistor string. The lower bits error will not repeat. So, the USER-SMILE method is not intended for a flash ADC or a delta sigma ADC. For other types of Nyquist-rate ADCs such as SAR ADC, Cyclic ADC, Pipeline ADC, the USER-SMILE algorithm works well.

### 6.2.0.2 Stimulus Requirement

As seen in (6.6), the linearity of test signal is no longer part of the final equations. Thus, the linearity requirement of test signal is completely dropped. This is one of the distinct advantages of USER-SMILE algorithm over SEIR algorithm which significantly eases the required tests of the signal source itself. However, the input signal needs to cover all the MSB, ISB and LSB segments to solve (6.8) providing sufficient number of codes for each segment to ensure estimation accuracy of the error terms. The required resolution of signal source is established through extensive simulation.
A 12-bit resolution of signal source is found sufficient for testing 12-bit ADC through USER-SMILE algorithm.

6.2.0.3 Constant Offset

The constant voltage offset is the most critical part of USER-SMILE algorithm. Any error in the voltage offset will cause estimation errors. The exact value of the voltage offset is also needed during the estimation steps. The offset \(a\) can be simply estimated by the average difference between the output codes \(C^{(1)}\) and \(C^{(2)}\). Although the constancy is not directly measurable in most cases, the voltage offset enabled by capacitor charge sharing scheme in this chapter is believed to be the best possible solution for on-chip implementation.

6.3 Algorithm Implementation

6.3.1 Design Choices - Hardware vs. Software

If a digital processor is available on chip e.g. a microcontroller, the BIST algorithm can be executed in the form of software routines by the processor. The software approach is more cost effective than the hardware approach (i.e. hardening the algorithm on chip as a co-processor) since no dedicated computing hardware is needed. However, the hardware approach remains attractive because of a few critical advantages that it provides.

6.3.1.1 No Dependency on On-chip Processor

One obvious advantage the hardware approach offers is that it still works when an on-chip processor is not available. Even with the presence of on-chip processor, the hardware approach offers the opportunity to test the ADC in a non-intrusive and non-competing manner. This advantage is critical for in-field testing when the on-chip processor is busy handling multiple user applications.
6.3.1.2 Memory Efficiency

In the software approach, the software implementation of the algorithm needs to be stored in on-chip memory (static RAM or non-volatile memory). This takes up system memory space which could cause issues especially when the system memory is tiny such as on some low-end microcontrollers. Contrary to the software approach, other than a small static RAM being used to hold ADC conversion results as well as intermediate data, no system memory is required for hardware approach.

6.3.1.3 Fast Execution Time

A dedicated computing resource realized by the hardware approach also speeds up the execution time of the algorithm significantly. Given same clock rate, it can be shown that the hardware approach is 38 times faster than the software approach based on ARM Cortex-M7 core. In the case of field-test, the hardware approach also means that the self-test can be completed much faster than the software approach. It makes a big difference in determining whether the self-test can be included as part of the power-on self-check sequence in an automotive application.

With all above being said, software approach still has its own advantages such as cost efficiency and flexibility. The decision regarding which approach to choose is driven by the availability of an on-chip processor, the technology node being used, the system memory size and the user application. In this chapter, the hardware approach is chosen due to its independence from an on-chip processor and reduced implementation complexity. As demonstrated later, the algorithm adaptation for hardware implementation and the advanced technology node being used help deliver a very efficient algorithm hardware.

6.3.2 Adaptation of USER-SMILE for Hardware Implementation

The fundamentals of the USER-SMILE algorithm are presented in above section. However, since $H$ is an $M \times K$ matrix and (6.8) requires the matrix inversion of an $K \times K$ matrix, a prohibitively long computation time and large data storage are required which is not suitable for an on-chip
implementation. In the above example, for a 16-bit ADC with 6-5-5 segmentation using 1 hit/code as the input, \( M \) will be 65536 and \( K \) is 128. To facilitate the ADC BIST on-chip computation and achieve fast and effective estimation, some modifications to the original equations must be made.

In matrix \( H \), three sub-matrices \( H_M, H_I \) and \( H_L \) can be defined, where \( H_M \) is column 1 to \( 2^{N_{MSB}} \) in \( H \), \( H_I \) is column \( 2^{N_{MSB}} + 1 \) to \( 2^{N_{MSB}} + 2^{N_{ISB}} \) in \( H \) and \( H_L \) is column \( 2^{N_{MSB}} + 2^{N_{ISB}} + 1 \) to \( 2^{N_{MSB}} + 2^{N_{ISB}} + 2^{N_{LSB}} \) in \( H \). Then, (6.7) can be expressed as:

\[
y_{dk} = [H_M \; H_I \; H_L] \cdot \begin{bmatrix} E_M \\ E_I \\ E_L \end{bmatrix} + \text{noise} \quad (6.9)
\]

Multiply both sizes by \( H^T \), which is \( [H_M \; H_I \; H_L]^T \).

\[
H^T \cdot y_{dk} = H^T \cdot H \cdot \begin{bmatrix} E_M \\ E_I \\ E_L \end{bmatrix} + \text{noise} \quad (6.10)
\]

In (6.10), \( H^T \cdot y_{dk} \) is a column matrix after multiplication and \( H^T \cdot H \) is a \( K \times K \) matrix.

\[
H^T \cdot H = \begin{bmatrix} H^T_M \\ H^T_I \\ H^T_L \end{bmatrix} \begin{bmatrix} H_M & H_I & H_L \\ H_M^T & H_I^T & H_L^T \end{bmatrix} = \begin{bmatrix} H^T_M H_M & H^T_M H_I & H^T_M H_L \\ H^T_I H_M & H^T_I H_I & H^T_I H_L \\ H^T_L H_M & H^T_L H_I & H^T_L H_L \end{bmatrix} \quad (6.11)
\]

Note that in this equation, most elements are 0s in \( H^T_M H_I, H^T_M H_L, H^T_I H_M, H^T_I H_L, H^T_I H_M, H^T_L H_I \). Then, these submatrices can be replaced with 0s and \( H^T \cdot H \) can be estimated as:

\[
H^T \cdot H \approx \begin{bmatrix} H^T_M H_M & \ldots & 0 \\ \vdots & H^T_I H_I & \vdots \\ 0 & \ldots & H^T_L H_L \end{bmatrix} \quad (6.12)
\]
With the estimated $H^T \cdot H$, (6.10) can be expressed as:

$$
\begin{bmatrix}
H^T_M \\
H^T_I \\
H^T_L
\end{bmatrix} \cdot y_{dk} \approx
\begin{bmatrix}
H^T_M H_M & \ldots & 0 \\
\vdots & H^T_I H_I & \vdots \\
0 & \ldots & H^T_L H_L
\end{bmatrix}
\begin{bmatrix}
E_M \\
E_I \\
E_L
\end{bmatrix} + \text{noise}
\quad (6.13)
$$

Next, define $H_{ydM} = H^T_M y_{dk}$ and $H_{HM} = H^T_M H_M$. Then $H_{ydI}, H_{ydL}$ and $H_{HI}, H_{HL}$ are defined in a similar way. The matrix can then be divided into three smaller matrices as shown in (6.14) which are now independent of each other so that they can be solved sequentially.

$$
H_{ydM} \approx H_{HM} \cdot E_M
$$

$$
H_{ydI} \approx H_{HI} \cdot E_I
$$

$$
H_{ydL} \approx H_{HL} \cdot E_L
\quad (6.14)
$$

Using the least-square method, $E_M, E_I$ and $E_L$ can be solved in (6.15).

$$
E_M \approx (H_{HM})^{-1} H_{ydM}
$$

$$
E_I \approx (H_{HI})^{-1} H_{ydI}
$$

$$
E_L \approx (H_{HL})^{-1} H_{ydL}
\quad (6.15)
$$

For every two samples, one set of ADC output codes will be obtained $C^{(1)}$ and $C^{(2)}$: one with offset and one without offset. Then, the matrix $H_{HM}$ and $H_{ydM}$ will be updated by the following equations:

$$
H_{HM}(C^{(1)}_{MSB}, C^{(1)}_{MSB}) = H_{HM}(C^{(1)}_{MSB}, C^{(1)}_{MSB}) + 1
$$

$$
H_{HM}(C^{(2)}_{MSB}, C^{(2)}_{MSB}) = H_{HM}(C^{(2)}_{MSB}, C^{(2)}_{MSB}) + 1
$$

$$
H_{HM}(C^{(1)}_{MSB}, C^{(2)}_{MSB}) = H_{HM}(C^{(1)}_{MSB}, C^{(2)}_{MSB}) - 1
\quad (6.16)
$$

$$
H_{ydM}(C^{(1)}_{MSB}) = H_{ydM}(C^{(1)}_{MSB}) + y_{dk}
$$

$$
H_{ydM}(C^{(2)}_{MSB}) = H_{ydM}(C^{(2)}_{MSB}) - y_{dk}
$$

The corresponding matrices for ISB and LSB will be also updated in a similar fashion. This operation only requires simple additions and increments which can be finished before the next set of
output codes are ready. After all M sets of codes have been obtained, the complete matrix can then be formed and the MSB, ISB and LSB error terms can then be evaluated one by one. Since $H_{HM}$, $H_{HI}$ and $H_{HL}$ are positive definite matrices, Cholesky decomposition can be used to evaluate the least square. The actual hardware implementation will be explained in section 6.4.

In all the equations above, we assume that the value of the offset between the pair of input signals is known. However, in practice, it is difficult to measure the offset directly so the offset value is estimated by calculating the average difference between the two sets of ADC output codes. During the data acquisition time, the average difference is also unknown. In order to calculate it, a default offset value $a^{(def)}$ is preset at the beginning. There will be a difference between the default offset $a^{(def)}$ and the actual offset $a$ which is summed and stored in a variable which is then used to store the summation of the output code difference to compute the average offset value. The $H_{ydM}$, $H_{ydI}$ and $H_{ydL}$ values will be updated after all the data is collected and the average output code difference is computed. During this process, the ADC output codes will not be stored in memory and only $H_{HM}$, $H_{HI}$, $H_{HL}$ and $H_{ydM}$, $H_{ydI}$, $H_{ydL}$ are stored. The memory required for the algorithm is thus significantly reduced and the subsequent computation time is also reduced since the matrix is already formed. This adaption from the USER-SMILE enables the on-chip ADC BIST implementation.

6.4 ADC BIST Subsystem Design

The block diagram of the ADC BIST subsystem is shown in Figure 6.1. The 12-bit redundant SAR ADC is the test object. An 12-bit resistive DAC is used here as test signal generator which does not have sufficient resolution and linearity accuracy to test 12-bit ADC using traditional histogram method. The ADC BIST and calibration logic is designed as a standalone digital block that performs all required control, computing and calibration tasks. A dedicated test mode is defined for the ADC BIST block to take over the essential control over ADC and DAC when the mode is activated. Each block is reviewed in details below.
6.4.1 12-bit Resistive DAC

A signal generator for this BIST solution needs to cover the majority of the input range of ADC to ensure most of the codes are hit. Although its linearity performance requirement is greatly reduced, it needs to provide sufficient resolution to drive ADC to generate sufficient number of code pairs (with and without offset) for accurate INL approximation by the algorithm. 1 hit per code is demonstrated to be sufficient as seen in simulation, therefore a 12-bit DAC is required. The chosen 28nm automotive microcontroller happens to carry a 12-bit DAC using resistor-ladder architecture which meets the requirements. When the BIST test mode is enabled, a 12-bit counter inside the BIST controller is sending control word to the DAC which output is sampled by the ADC. Every one DAC output after settling is sampled by ADC twice, one without offset and one with offset. This is to make sure that the ADC is converting same input voltage where the offset is applied within ADC as detailed in next section. Benefiting from the USER-SMILE algorithm, the DAC doesn’t need to be tested on its linearity performance. Therefore, the checking of a dead DAC (ADC output code does not change), missing input range (no code found in specific segments) and invalid offset are implemented in ADC BIST sub-block to screen out catastrophic DAC failures only.

6.4.2 12-bit Redundant SAR ADC and Voltage Offset Creation

The ADC under test is a single-ended charge redistribution SAR ADC. As shown in Figure 6.2, the p-side CDAC array (CDACP) is the main array being used for conversion. The m-side CDAC array (CDACM) maintains common mode voltage on its top plate to provide reference voltage and capacitor matching to the comparator. A split-capacitor structure is implemented to limit the die size of CDACP. Therefore, the CDACP is partitioned into 3 binary weighted sub-DACs based on unit size capacitors $C_u$. The MSB capacitor array is controlled by 5 bits (bit 12 to 9 and bit s), the ISB capacitor array is controlled by 4 bits (bit 8 to 5), and the LSB capacitor array is controlled by 6 bits (bit 4 to 0 and bit terminal). To correct the capacitor mismatches, a calibration scheme is implemented. To distinguish this with later proposed BIST enabled calibration, this
The accuracy of linearity approximation provided by the USER-SMILE algorithm is critically dependent on the constancy of the voltage offset (54). Previous publications have focused on creating
the voltage offset within a signal generator e.g. (75). This significantly complicates the design of the signal generator and may still not provide the required constancy across the entire input range of the ADC. As presented by (53), an additional capacitor is inserted into the existing SAR capacitor array to create the voltage offset which is only determined by the capacitor ratio and a constant reference voltage supply. This is believed to provide superior constancy and simplicity to prior arts.

In this work, the creation of constant voltage offset shares the same principle as (53). Instead of adding an additional capacitor, the CDACM array is leveraged to create a different reference voltage for the comparator. When a voltage offset is wanted, the bottom plate of capacitor $C_{rt}$ which is controlled by bit $brt$ is toggled to $V_{refh}$ during evaluation phase where the bottom plates of other capacitors in CDACM remain connected to $V_{refl}$. A few disadvantages that were encountered in (53) have been avoided. No additional capacitor is required in this implementation. It does not reduce the step size seen by the comparator. The top plate voltage of CDACM does not surpass safeguard voltage which avoided a potential device reliability risk. This essentially created a voltage offset shown in (6.17).

$$V_{\text{Offset}} = (V_{\text{refh}} - V_{\text{refl}}) \cdot \frac{C_{rt}}{C_M} \cdot (2^N + W(C_s) - C_{\text{cal}}) \quad (6.17)$$

In the above equation, $V_{\text{refh}}$ and $V_{\text{refl}}$ are the high reference voltage and low reference voltage respectively. $C_M$ is the sum of total capacitors in CDACM array. $N$ is the resolution of ADC. $W(C_s)$ is the weight of $C_s$ where $C_{\text{cal}}$ is the total sum of CWI calibration codes of MSB capacitors. The constancy of the offset across all possible codes is determined by the constancy of capacitor ratio $C_{rt}/C_M$, the stability of reference voltages $V_{\text{refh}}$ and $V_{\text{refl}}$, and the constancy of offset voltage of the comparator. CDACM does not participate the bit evaluation while merely providing reference voltage to the comparator. The constancy of the capacitor ratio should hold regardless of the code being converted. The reference voltages are supplied from dedicated and precise tester instrument which short term drift performance is guaranteed. A large amount of decoupling capacitors are added on the reference voltage supplies to reduce noise and maintain voltage level. The voltage offset of the comparator is small due to the high gain of comparator and fixed due to fixed inputs.
of the comparator. Substituting true design values into the equation and representing $V_{\text{Offset}}$ in LSB, the expected amount of offset is between 263 and 264 LSB depending on actual value of $C_{\text{cal}}$. Although there is no direct way to measure the constancy of the offset, we use the expected value and the silicon correlation results in section 6.5 to measure it indirectly.

### 6.4.3 BIST and BIST Enabled Calibration

The BIST block is consisted of a finite sate machine (FSM) sub-block, the hardened USER-SMILE algorithm unit, a small memory to hold intermediate and final test data, a small test MUX sub-block, and a calibration sub-block enabled by BIST. The FSM sub-block is designed control the operation sequence of the subsystem. When the operation kicks off by activating BIST test mode, the FSM logic starts a 12-bit counter that sends control words to the 12bit DAC. It also toggles the offset enable control signal to the ADC for voltage offset creation. Once valid ADC conversion results are available, the FSM logic kicks off the USER-SMILE algorithm unit for post processing. The entire operation continues until all possible control words of 12-bit DAC have been covered and valid BIST pass/fail status have been reported. The test MUX is a simple logic block that is designed to bypass the normal control and data path of the DAC and the ADC for the BIST block to take over. The algorithm unit is the hardened block implementation of the USER-SMILE algorithm. A small dedicated memory is paired with the algorithm unit to hold intermediate and final data. It would be possible to share a small portion of the system memory within the SoC for this purpose to reduce the die size of the solution. We choose to implement a dedicated memory due to its small size and simplicity of integration. Below sections highlight the design of algorithm unit and BIST enabled calibration logic.
6.4.3.1 High Level Synthesis Assisted Algorithm Unit Design

Per the USER-SMILE algorithm, the nonlinearity of SAR ADC is modeled as coefficients in MSB, ISB and LSB segments which are represented as $INL_M$, $INL_I$ and $INL_L$. After running through all ADC output codes, a matrix equation is constructed in the form of the equation below.

$$H_{Hx} \cdot INL_x = H_{yd}x, \ x \in [M, I, L]. \quad (6.18)$$

$H_{Hx}$ and $H_{yd}x$ are obtained through (6.16). The hardened algorithm unit is implemented as shown in Figure 6.3. The matrix construction block receives the pair of ADC output codes (with and without voltage offset applied): $C^{(1)}$ and $C^{(2)}$, and constructs a matrix $HH_x$ and $H_{yd}x$, and then saves the matrix into memory until all valid ADC conversions have been completed. The least square block then uses the linear least square method to solve this over-determined system to get $INL_x$. Once all of the INL coefficients have been identified, they will be used to reconstruct final INL, DNL and TUE parameters. A high level synthesis (HLS) technique was used to realize this design. HLS significantly simplifies the design, speeds up the design process while yielding satisfactory results.

6.4.3.2 BIST Enabled Calibration

The SAR ADC has its own structural CWI calibration based on capacitor mismatch measurement as detailed in section 6.4.2. However, calibrating capacitor mismatch may not provide best DNL/INL results as the total errors contributing to INL is not only from the capacitor mismatches. All these structure-based test methods have such limitations as they cannot remove other error sources. The goal for ADC static linearity is to achieve minimum INL. With the full-code INL/DNL information, we can calibrate it correspondingly. Unlike other structural test which have some assumptions, full-code INL doesn’t assume any structure-specific errors. Therefore, it is preferred to obtain the full-code INL information for calibration purpose. As the BIST solution readily provides the full-code INL information, it is only natural to extend the BIST results to further correct ADC linearity errors.
INL is defined as the deviation of code transition from its ideal location. The INL at code $C$ is:

$$INL(C) = \frac{[T(C) - T^i(C)]}{V_{\text{lsb}}}$$ \hspace{1cm} (6.19)

where $T(C)$ is the actual transition voltage from code $C - 1$ to $C$; $T^i(C)$ is the ideal transition voltage; and $V_{\text{lsb}}$ is the ideal 1 LSB in voltage.

For digital calibration, the final output code is:

$$C_{\text{final}} = C_{\text{adc}} + C_{\text{cal}}$$ \hspace{1cm} (6.20)

where $C_{\text{adc}}$ is the adc codes from shift registers and $C_{\text{cal}}$ is the calibration code we want to generate.

The INL after calibration is expressed as:

$$INL(C_{\text{final}}) = \frac{[T(C_{\text{adc}}) - T^i(C_{\text{final}})]}{V_{\text{lsb}}} = \frac{[T(C_{\text{adc}}) - [T^i(C_{\text{adc}}) + C_{\text{cal}} \cdot V_{\text{lsb}}]]}{V_{\text{lsb}}} = INL(C_{\text{adc}}) - C_{\text{cal}}$$ \hspace{1cm} (6.21)

As indicated in (6.21), to minimize $INL(C_{\text{final}})$, $C_{\text{cal}}$ should be equal to $INL(C_{\text{adc}})$. Considering the most significant error is from the MSB segment, in the current implementation, only
MSB errors are calibrated. Therefore, the calibration code is:

\[ C^{cal} = \text{round}(E_M(C_{\text{adc}MSB}^{cal})) \]  

(6.22)

where \( C_{\text{MSB}}^{cal} \) is the MSB segment codes of \( C^{cal} \) and the \text{round} function is to round the calibration code to the ADC’s minimum quantization bit. Assuming the error from ISB and LSB segment is small, the INL after calibration is the rounding error due to finite resolution.

6.5 Measurement Results

The ADC BIST and BIST enabled calibration logic consumes 18K gates after logic synthesis. The 4KB memory array occupies 0.011\( mm^2 \). With 75% utilization rate, the total BIST solution occupies 0.028\( mm^2 \) on 28nm CMOS technology. Compared to a minimum 20 hits per code that is required by traditional histogram ramp test method in production, this BIST solution only requires 2 measurements per code. The computation time of the BIST takes 1ms with a 80MHz clock which is negligible in the overall test time. A minimum 10X tester time reduction is achieved which significantly reduces the ADC’s test cost. Although cost metric (test cost per second, die cost per \( mm^2 \) and etc.) varies and deemed business confidential, the cost of increased die size is estimated to be only 1/8 of the saved cost from tester time alone.

6.5.1 BIST Test Results

In order to verify the test accuracy of the ADC BIST solution, a 200 HPC histogram ramp test is developed for comparison. The ramp signal is delivered by a high-performance tester instrument which is specified to be sufficiently linear. The ADC BIST only uses two 1 hit per code ramps generated using the on-chip DAC. 5-4-3 segmentation is chosen for the definition of INL error coefficients to mimic the segmentation of SARADC capacitor array, thus only 56 \( (2^5 + 2^4 + 2^3) \) error coefficients need to be identified. After INL or DNL performance data for 4096 codes is derived, they are saved into on-chip memory for correlation or characterization.

Raw ADC data with and without offset are captured and shown in Figure 6.4. The amount of offset is indeed centering around 263 LSB as calculated in Eq. 6.17. As opposed to a clean and
constant value where only an ideal noise-free and error-free ADC could achieve, the offset clearly incorporates the noise and nonlinearity errors which are the exact info needed by the USER-SMILE algorithm. A minor issue is exposed by Figure 6.4 where the ramp with offset saturates at code index 3700. This reduces the effective number of code pairs from 4096 to 3700 to construct the matrix which may negatively impacts the test accuracy of BIST solution. The root cause of the issue is found out to be the $C_s$ induced gain error which is not properly compensated in ADC BIST test mode. An fix to the issue is either to remove the gain error completely or to introduce a negative gain error which would maximize the available number of code pairs. However, it has to be pointed out that all segments of the ADC are covered from 0 to 4095 despite this issue as shown clearly in Figure 6.4.

The DNL and INL curves measured on one unit using the 200 HPC histogram method and ADC BIST method are plotted in Figure 6.5. Figure 6.5.c showed the $\Delta$ of INL between histogram method and BIST solution. Despite spikes at a few locations, most differences are within +/-0.5LSB which indicates the level of BIST test accuracy. This level of test accuracy is deemed sufficient as
Figure 6.5  DNL/INL correlation

demonstrated in following correlation exercises. However, a few things are believed to attribute to the differences:

- In USER-SMILE algorithm, the noise term including input referred noise and quantization noise is assumed to be averaged to 0 when certain criteria is met. In reality, this may not be the case which causes estimation error. Proper dithering is needed.

- The amount of offset in this design is more than 2 MSB segments (<128 code for each MSB segment) which causes fewer correlated code pairs between adjacent even and odd MSB segments. In Figure 6.5.c, the “toggling” of the differences in even and odd MSB segments can be attributed to this. Choosing appropriate offset is required to address this issue.
- The aforementioned $C_s$ induced gain error not being compensated reduces the available number of code pairs for algorithm data processing, thus reducing the estimation accuracy.

Figure 6.6 zoomed in on one of the code locations where a spike is noticed. As the graph clearly indicates, the error is due to the misalignment of transition edge between two MSB segments. This happens during reconstruction of the full code INL when the transition edges of MSB segments is approximated without knowing their exact code location. Those edges are determined by the CWI calibration codes, however the codes are not made directly available to the BIST sub-block where approximation has to be made. The fix is straightforward which is to make the CWI calibration codes accessible to BIST block. As this is a misalignment issue, it does not impact the maximum DNL and INL errors which are reported to determine the pass/fail status of the ADC test.

To further verify the correlation between histogram and ADC BIST and to establish the error tolerance of the BIST solution, a large number (> 700) of parts from different corner processes are tested across maximum and minimum operating voltages and temperatures. The maximum INL and minimum INL are obtained for comparison. Figure 6.7(a) plots the INL data obtained from
histogram test with different number of hits per code. It shows good correlation between the two with +/-0.5LSB as a clear boundary of the variance.

Figure 6.7 plots the correlation of INL between the histogram method using 200 HPC and the BIST solution. The same +/-0.5LSB boundary is applied for comparison. Although the correlation between those two sets of data is slightly worse than the correlation in Figure 6.7(a) with a few data points falling out of the boundary, it clearly indicates similar level of test accuracy of BIST solution when compared to that of traditional histogram method. There appeared to be 4 outlier points, but since the tester does not save intermediate test data to save tester time, attempting to identify the outlier parts or causes would require long test time and high cost that are not warranted since the errors are small. With the ADC INL specification limits set at +/-3LSB, the BIST accuracy is deemed to be sufficient with no good unit being rejected or bad unit escaping. One unit that failed on both histogram tests is correctly identified and rejected by BIST solution.

### 6.5.2 Calibration Results

To prove the effectiveness of this BIST enabled calibration scheme, the ADC is tested on a high end mixed signal tester. Traditional histogram based test is performed to find its static
The aforementioned code misalignment issue does not impact the calibration accuracy here since the MSB bits are monitored and used to decode and retrieve calibration codes from BIST memory where CWI calibration codes need not be known. Since the low frequency dynamic performance of a ADC is closely tied to its static linearity, a dynamic test is performed to obtain the dynamic performance. Figure 6.8(a) and 6.8(b) showed a typical device’s DNL and INL under different calibration conditions. Both structural CWI calibration and BIST enabled calibration can bring the ADC static performance to a satisfactory level. However, noticeable performance enhancement of DNL and INL is achieved by the BIST enabled calibration. As only MSB errors are calibrated, the INL error is now dominated by the gain error in each MSB segment and noise.

Figure 6.8 Performance Comparison
spectrum as shown in Figure 6.8(c). The detailed performance data of one device is shown in Table 6.1.

<table>
<thead>
<tr>
<th></th>
<th>No Cal</th>
<th>Structural CWI Cal</th>
<th>BIST Cal</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNL (LSB)</td>
<td>1.1253/-1</td>
<td>0.6726/-0.9531</td>
<td>0.4066/-0.6718</td>
</tr>
<tr>
<td>INL (LSB)</td>
<td>2.4324/-3.4909</td>
<td>0.9636/-1.1818</td>
<td>0.6633/-0.7572</td>
</tr>
<tr>
<td>SNR (dB)</td>
<td>62.3533</td>
<td>68.6602</td>
<td>69.7948</td>
</tr>
<tr>
<td>THD (dB)</td>
<td>-64.8619</td>
<td>-74.937</td>
<td>-84.4755</td>
</tr>
<tr>
<td>SNDR (dB)</td>
<td>60.4186</td>
<td>67.7412</td>
<td>69.6494</td>
</tr>
<tr>
<td>SFDR (dB)</td>
<td>66.2537</td>
<td>75.4594</td>
<td>86.5137</td>
</tr>
<tr>
<td>ENOB (bit)</td>
<td>9.744</td>
<td>10.9603</td>
<td>11.2773</td>
</tr>
</tbody>
</table>

A >0.3bit ENOB improvement is achieved over structural CWI calibration with the THD and SFDR contributing the most with each parameter gaining >10dB improvement. The performance improvement is significant even not considering this calibration being essentially free. Applying this BIST enabled calibration scheme does not exclude the BIST solution from testing the ADC again as the fundamentals of the BIST algorithm are not changed by the calibration.

6.6 Conclusion

An ADC BIST solution and calibration scheme has been presented and reviewed. It removes the dependency to a high-performance external test instrument, and also significantly reduces test time. Extensive correlation has been performed to demonstrate the test accuracy of the BIST solution. The presented technique is extended to calibrate an ADC for superior static and dynamic linearity performance. It opens the door for in-field ADC performance testing that may be required to meet stringent functional safety requirement in certain application e.g. autonomous driving. The BIST circuit’s die size cost is negligible compared to the test cost savings being realized, both recurring and non-recurring.
CHAPTER 7. CONCLUSION

This work focuses on low-cost built-in self-test, self-calibration algorithms and the hardware implementations for ADC linearity test. The challenges associated with linearity testing of high performance ADCs are addressed. Two testing algorithms are proposed with rigorous analysis and verification. Two BIST circuits are developed, designed and fabricated. These results demonstrate the feasibility, effectiveness and efficiency of implementing ADC BIST on-chip.

The USER-SMILE algorithm is proposed to realize the ADC on-chip INL/DNL testing. It uses two nonlinear signals with a constant voltage shift in between. With a stimulus error removal technique, the stimulus linearity requirement is eliminated. It overcomes the fundamental limitation of requiring the input signal to have smooth nonlinearity in the SEIR algorithm. Therefore, any waveform including DAC-generated nonlinear signals can be used in the USER-SMILE algorithm as long as the input signal covers the ADC input range. With a segmented INL model, the number of parameters needed to represent the full-code INL is reduced, and thus, the total test time is significantly reduced. With the two techniques, USER-SMILE can achieve low-cost on-chip but accurate testing for high resolution ADCs with an inaccurate signal generator. In the conventional histogram test, the signal generator linearity must be much better than the ADC itself and the test time is significantly long. The USER-SMILE algorithm overcomes both challenges, which makes low-cost and efficient ADC BIST feasible.

The pipelined ADC testing and calibration algorithm explores the redundancy characteristics of the multi-bit/stage pipelined ADC. It applies a segmented linearity model to represent the linearity error of the pipelined ADC using the raw codes in each stage. It enables fewer number of samples to test the ADC INL/DNL. And the model parameters can be used to reconstruct an extended version of the INL, which cannot be achieved with the conventional histogram test. The model parameters can be further used for ADC linearity calibration. Measurement results have shown that a 12-bit
pipelined ADC can be tested and calibrated to achieve 10.84 bits ENOB and 85 dB SFDR. The same technique can be used for on-chip BIST for pipelined ADC using the modified USER-SMILE algorithm. It significantly reduces the test time while achieving better testing coverage than the conventional histogram test.

The proposed R2R DAC is developed as a low-cost on-chip signal generator for the USER-SMILE algorithm. By using a subradix-2 architecture, the linearity requirement for the R2R DAC is significantly relaxed. With an extra resistor, the R2R DAC can achieve the constant voltage shift generation. A detailed error analysis is presented. The R2R DAC can be designed in a systematic way with a small area. A 14-bit R2R DAC fabricated in 40nm technology was used to test a 15-bit ADC accurately with less than 0.5 LSB INL estimation errors.

A complete SAR ADC BIST solution is developed. The signal generator is an untested DAC and the voltage shift generator is implemented with a switched-capacitor inside the SAR ADC. The USER-SMILE algorithm is implemented as a hardware block on-chip. All the digital control logic, memory and calibration circuits are also implemented on-chip. The measurement results show that the ADC can be accurately tested with only 8000 samples and the estimation error is less than 0.5 LSB. The ADC is calibrated on-chip with the test results. The ADC INL is improved from 3.5 LSB to 0.75 LSB and the SFDR is improved by 20dB.

The proposed algorithms and circuits not only reduce the test cost, but also enable on-chip testing and calibration, which is extremely important for mission-critical applications such as automotive where reliability and safety are the top priority. With the power-on and periodic self-test and self-calibration, the ADC performance is guaranteed during its entire lifetime. The accurate ADC after calibration can also be used to monitor other important signals in the chip to ensure the functionality of the entire chip and the functional safety in automotive electrical and electronic systems.

Benefiting from the self-testing and calibration capability, the design requirements can also be relaxed. The ADC can be sized with relaxed matching, a smaller area, and less power. Researchers
and companies can apply the proposed methods to existing and future analog and mixed signal products to achieve better performance and better reliability with less design efforts and test cost.
BIBLIOGRAPHY


APPENDIX . DERIVATION OF EQUATIONS

Derivation of Voltage Shift Non-constancy

This section derives the constancy requirement for the voltage shift. Equation (5.23) can be re-written into:

\[
V_{\text{shift}} = \frac{R_{\text{imp}}^{(p)}[0](V_{\text{refh}} - V_{\text{refl}})}{R_s^{(p)} + R_{\text{omp}}[s] + R_{\text{imp}}^{(p)}[0]} + \frac{R_{\text{imp}}^{(n)}[0](V_{\text{refh}} - V_{\text{refl}})}{R_s^{(n)} + R_{\text{omp}}[s] + R_{\text{imp}}^{(n)}[0]}
\]

\[
+ \frac{R_{\text{imp}}^{(p)}[0](R_{\text{omp}}^{(p)}[s] - R_{\text{omp}}^{(p)}[0])}{(R_s^{(p)} + R_{\text{omp}}^{(p)}[s] + R_{\text{imp}}^{(p)}[0])(R_s^{(p)} + R_{\text{omp}}^{(p)}[s] + R_{\text{imp}}^{(p)}[0])}
\]

\[
+ \frac{R_{\text{imp}}^{(n)}[0](R_{\text{omp}}^{(n)}[s] - R_{\text{omp}}^{(n)}[0])}{(R_s^{(n)} + R_{\text{omp}}^{(n)}[s] + R_{\text{imp}}^{(n)}[0])(R_s^{(n)} + R_{\text{omp}}^{(n)}[s] + R_{\text{imp}}^{(n)}[0])}
\]  

(1)

The first two terms only contain the impedance values. The last two terms have both the impedance values and the equivalent output voltages. To simplify the analysis of this complicated equation, the first two terms \(V_{\text{shift-12}}\) and last two terms \(V_{\text{shift-34}}\) are evaluated separately.

The first two terms can be decomposed into:

\[
V_{\text{shift-12}} = \frac{(R_{\text{imp}}^{(p)}[0] + \Delta R_{\text{imp}}^{(n)}[0])(V_{\text{refh}} - V_{\text{refl}})}{R_s^{(p)} + R_{\text{omp}}^{(p)}[s] + R_{\text{imp}}^{(p)}[0]} + \frac{(R_{\text{imp}}^{(p)} + \Delta R_{\text{imp}}^{(n)}[0])(V_{\text{refh}} - V_{\text{refl}})}{R_s^{(n)} + R_{\text{omp}}^{(n)}[s] + R_{\text{imp}}^{(n)}[0]}
\]

\[
\approx \frac{R_{\text{imp}}^{(p)}(V_{\text{refh}} - V_{\text{refl}})}{R_s^{(p)} + R_{\text{omp}}^{(p)}[s] + R_{\text{imp}}^{(p)}[0]} + \frac{R_{\text{imp}}^{(n)}(V_{\text{refh}} - V_{\text{refl}})}{R_s^{(n)} + R_{\text{omp}}^{(n)}[s] + R_{\text{imp}}^{(n)}[0]}
\]

\[
+ \frac{\Delta R_{\text{imp}}^{(p)}[0]}{R_s^{(p)} + R_{\text{omp}}^{(p)}[s] + R_{\text{imp}}^{(p)}[0]} \frac{R_{\text{imp}}^{(n)}[0](V_{\text{refh}} - V_{\text{refl}})}{R_s^{(p)} + R_{\text{omp}}^{(p)}[s] + R_{\text{imp}}^{(p)}[0]}
\]

\[
+ \frac{\Delta R_{\text{imp}}^{(n)}[0]}{R_s^{(n)} + R_{\text{omp}}^{(n)}[s] + R_{\text{imp}}^{(n)}[0]} \frac{R_{\text{imp}}^{(p)}[0](V_{\text{refh}} - V_{\text{refl}})}{R_s^{(n)} + R_{\text{omp}}^{(n)}[s] + R_{\text{imp}}^{(n)}[0]}
\]  

(2)

where \(\Delta R_{\text{imp}}\) is the impedance difference caused by the mismatches. In (2), the first two terms are constant over all DAC codes. Define these two terms as the ideal constant shift \(\alpha^i\):

\[
\alpha^i = \frac{R_{\text{imp}}^{(p)}[0](V_{\text{refh}} - V_{\text{refl}})}{R_s^{(p)} + R_{\text{omp}}^{(p)}[s] + R_{\text{imp}}^{(p)}[0]} + \frac{R_{\text{imp}}^{(n)}[0](V_{\text{refh}} - V_{\text{refl}})}{R_s^{(n)} + R_{\text{omp}}^{(n)}[s] + R_{\text{imp}}^{(n)}[0]}
\]

(3)

Only the last two terms in (2) are changing with DAC codes. The MSB bit is evaluated first to see its impact on \(R_{\text{imp}}^{(p)}[0]\) and \(R_{\text{imp}}^{(n)}[0]\). For bit 0, the impedance \(R_{\text{imp}}^{(p)}[0]\) can be rewritten with
the switch-on resistance:

\[ R_{\text{imp}}^{(p)}[0]|_{b[0]=1} = (R_{\text{imp}}^{(p)}[1] + R_{1}^{(p)}[0])||(R_2^{(p)}[0] + R_{\text{onp}}^{(p)}[0]), \tag{4.4} \]

when the \( b[0] = 1 \) and PMOS switch is turned on in the positive side. The negative side and the \( b[0] = 0 \) case can also be obtained in the same way. Then, we can compare the voltage shift difference between the \( b[0] = 1 \) case and the \( b[0] = 0 \) case.

\[
\frac{R_{\text{imp}}^{(p)}[0]|_{b[0]=1} - R_{\text{imp}}^{(p)}[0]|_{b[0]=0}}{R_{\text{imp}}^{(p)}[1] + R_{1}^{(p)}[0] + R_2^{(p)}[0] + R_{\text{onp}}^{(p)}[0]} = \frac{(R_{\text{imp}}^{(p)}[1] + R_1^{(p)}[0])(R_2^{(p)}[0] + R_{\text{onp}}^{(p)}[0])}{R_{\text{imp}}^{(p)}[1] + R_1^{(p)}[0] + R_2^{(p)}[0] + R_{\text{onp}}^{(p)}[0]} \tag{5.5}.
\]

Define the resistor ratio \( \beta^{(p)} \) as

\[
\beta^{(p)} = \frac{R_{\text{imp}}^{(p)}[1] + R_1^{(p)}[0]}{R_{\text{imp}}^{(p)}[1] + R_1^{(p)}[0] + R_2^{(p)}[0]}. \tag{6.6}
\]

\( \beta^{(p)} \) is slightly less than 1/2 due to the subradix-2 architecture even with resistor mismatches. For approximation, we can use 1/2 to obtain the relative quantity.

Then, we can obtain the impedance difference due to the switch on-resistance mismatches for positive and negative sides.

\[
\Delta R_{\text{imp}}^{(p)}[0]|_{b[0]=1} - \Delta R_{\text{imp}}^{(p)}[0]|_{b[0]=0} \approx (\beta^{(p)})^2(\Delta R_{\text{on}} + \Delta R_{\text{onp}}^{(p)}[0] - \Delta R_{\text{onn}}^{(p)}[0]) \tag{7.7}
\]

\[
\Delta R_{\text{on}} + \Delta R_{\text{onp}}^{(p)}[0] - \Delta R_{\text{onn}}^{(p)}[0]) / 4.
\]

Similarly,

\[
\Delta R_{\text{imp}}^{(n)}[0]|_{b[0]=1} - \Delta R_{\text{imp}}^{(n)}[0]|_{b[0]=0} \approx (-\Delta R_{\text{on}} - \Delta R_{\text{onp}}^{(n)}[0] + \Delta R_{\text{onn}}^{(n)}[0]) / 4. \tag{8.8}
\]

The on-resistance difference between the MSB bit NMOS and PMOS switches is approximately scaled down by a factor of 4 at the output node. With equation \((.7)\) and \((.8)\), the difference of
equation (2) at \( b[0] = 1 \) and \( b[0] = 1 \) can be approximated as:

\[
V_{\text{shift-12}}|_{b[0]=1} - V_{\text{shift-12}}|_{b[0]=0} \\
\approx \alpha^i \frac{\Delta R_{\text{onp}}^{(p)}[0] - \Delta R_{\text{onp}}^{(p)}[0]}{8R_{\text{imp}}^i} + \alpha^i \frac{\Delta R_{\text{onn}}^{(n)}[0] - \Delta R_{\text{onp}}^{(n)}[0]}{8R_{\text{imp}}^i}.
\]

The on-resistance difference \( \Delta R_{\text{on}} \) between NMOS and PMOS switch is canceled in the differential architecture. Only the local mismatches of PMOS and NMOS switches contribute to the final non-constancy. Similarly, the second MSB bit on-resistance mismatch will contribute 1/4 of errors to the second bit node impedance, which is 1/4 of the impact from first MSB contribution. Considering all the following bits, the worst case total error from the second to the last bit is 1/3 of the error from the first MSB bit. Therefore, the dominant error is from the first MSB bit.

The last two terms of (1) can be further decomposed into:

\[
V_{\text{shift-34}}
\]

\[
= \frac{R_{\text{imp}}^{(p)}[0](R_{\text{onp}}^{(p)}[s] - R_{\text{onp}}^{(p)}[s])}{(R_s^{(p)} + R_{\text{onp}}^{(p)}[s] + R_{\text{imp}}^{(p)}[0])(R_s^{(p)} + R_{\text{onp}}^{(p)}[s] + R_{\text{imp}}^{(p)}[0])} (V_{\text{eq}}^{(p)} - V_{\text{refl}})
\]

\[
+ \frac{R_{\text{imp}}^{(n)}[0](R_{\text{onp}}^{(n)}[s] - R_{\text{onn}}^{(n)}[s])}{(R_s^{(n)} + R_{\text{onp}}^{(n)}[s] + R_{\text{imp}}^{(n)}[0])(R_s^{(n)} + R_{\text{onp}}^{(n)}[s] + R_{\text{imp}}^{(n)}[0])} (V_{\text{eq}}^{(n)} - V_{\text{refl}})
\]

\[
+ R_{\text{imp}}^i (\Delta R_{\text{onp}}^{(n)}[s] - \Delta R_{\text{onn}}^{(n)}[s])(V_{\text{eq}}^{(n)} - V_{\text{refl}})
\]

\[
\approx \alpha^i \frac{\Delta R_{\text{on}}}{R_s^i + R_{\text{imp}}^i} + \alpha^i \frac{(\Delta R_{\text{onp}}^{(p)}[s] - \Delta R_{\text{onp}}^{(p)}[s])(V_{\text{eq}}^{(p)} - V_{\text{refl}})}{(R_s^i + R_{\text{imp}}^i)(V_{\text{vefh}} - V_{\text{refl}})}
\]

\[
+ \alpha^i \frac{(\Delta R_{\text{onp}}^{(n)}[s] - \Delta R_{\text{onn}}^{(n)}[s])(V_{\text{eq}}^{(n)} - V_{\text{refl}})}{(R_s^i + R_{\text{imp}}^i)(V_{\text{vefh}} - V_{\text{refl}})}
\]

where the first term is constant and the last two terms change with DAC codes. The worst case value of \( V_{\text{shift-34}} \) depends on the on-resistance difference. Since the value of \( V_{\text{eq}}^{(p)} \) and \( V_{\text{eq}}^{(n)} \) are approximately linear to the DAC code, the non-constant part from \( V_{\text{shift-34}} \) is approximately a straight line over the DAC output voltage. Therefore, the worst case non-constant value is always at the two ends of the DAC output voltages.
Combining (2) and (10), the final voltage shift is approximately as:

\[
V_{\text{shift}} = V_{\text{shift-12}} + V_{\text{shift-34}}
\]

\[
\approx \alpha i + \alpha i \frac{\Delta R_{on}}{R_s + R_{\text{imp}}} + \alpha i \frac{\Delta R_{\text{imp}}^{(p)}[0] + \Delta R_{\text{imp}}^{(n)}[0]}{R_{\text{imp}}}
\]

\[
+ \alpha i \frac{(\Delta R_{onp}^{(p)}[s] - \Delta R_{onp}^{(n)}[s])(V_{eq}^{(p)} - V_{\text{refl}})}{(R_s + R_{\text{imp}})(V_{\text{vefh}} - V_{\text{refl}})}
\]

\[
+ \alpha i \frac{(\Delta R_{onp}^{(n)}[s] - \Delta R_{onp}^{(p)}[s])(V_{eq}^{(n)} - V_{\text{refl}})}{(R_s + R_{\text{imp}})(V_{\text{vefh}} - V_{\text{refl}})}.
\]  

With the value in (9), the voltage shift constancy is:

\[
\frac{\max(V_{\text{shift}}) - \min(V_{\text{shift}})}{\text{mean}(V_{\text{shift}})} \approx \frac{\sigma_{Ron}}{4R_{\text{imp}}} + \sqrt{2}\frac{\sigma_{Ron}}{R_s + R_{\text{imp}}}
\]  

(.12)